

# An Analysis Framework for Transient-Error Tolerance

John P. Hayes\* Ilia Polian<sup>§</sup> and Bernd Becker<sup>§</sup>

\* University of Michigan  
2260 Hayward Street  
Ann Arbor, MI 48109-2121, USA

<sup>§</sup> Albert-Ludwigs-University  
Georges-Köhler-Allee 51  
79110 Freiburg im Breisgau, Germany

**Abstract:** *Transient or soft errors are an increasing problem in mainstream microelectronics. We propose a framework for modeling transient-error tolerance (TET) in logic circuits. We classify transient errors as critical or non-critical according to their impact on circuit behavior, such as their ability to disturb the internal state for specified periods of time. We introduce a metric called the critical soft-error rate (CSER) as an alternative to conventional SER, and present some analysis strategies based on CSER. This approach employs a new single transient fault (STF) model, which is defined in terms of a temporary stuck-at fault and its associated circuit state. Although basically technology-independent, STFs can be extended with low-level physical attributes. With STFs, we can estimate the transient error probability  $p_{err}$  of a circuit's nodes, as well as various measures of error susceptibility and TET. We demonstrate the use of STFs with combinational and sequential circuits, including several types of adders. We also present a systematic hardening strategy that uses  $p_{err}$  as a guide to improving TET.*

## 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 digital ICs [HN06]. 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 some applications, transient errors are acceptable as long as the correct behavior is restored quickly. Suppose 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 corrupt the system's output for a few clock cycles, after which it recovers automatically. Another example is a video system that tolerates a few

missing pixels [KKK06]. In systems with a human end-user, brief deviations of the output data from their correct values may not be perceptible [Bre04].

A basic unresolved question is how to model transient faults and errors affecting general logic circuits. Such models should be useful for analysis tasks such as computing soft error rate (SER), and synthesis tasks like error-tolerant design. Early research in this field addressed various behavioral and statistical aspects of intermittent and transient faults without defining explicit fault models for them [Bre73, Sav80]. Other work considered the computation of signal probabilities in fault-free circuits to estimate fault-detection difficulty with random testing [PM75, BT89]. Random testing has also been analyzed probabilistically in terms of the delay (error latency) between a fault's occurrence and the appearance of an output error [SM76]. Recent papers have focused primarily on physical models of transient faults and their error propagation probabilities [AT05, ZDB05]. Hardening a selected subset of nodes (gates) is a promising way to achieve protection against transient errors at acceptable cost [MT03, Bau05, HN06]. In [MT03], for example, the susceptibility of individual nodes to soft errors is calculated via electrical analysis. The reduction in SER by hardening the most susceptible nodes can then be calculated. It is also possible to harden only the flip-flops of the circuit, or a subset of flip-flops [JR+06, ZM+07].

We are investigating a very general type of *transient-error tolerance* (TET) based on the observation that not all soft errors at a circuit's output are critical. A soft error is considered *non-critical* or *tolerable* if it disappears within a specified time, the *non-critical error period* (NEP), with some specified probability. Non-critical 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 non-critical errors, thus reducing design costs. In this work, we explore the modeling of transient faults and errors, computing their probability of occurrence, and the notion of NEP. We also introduce a technique for selective hardening to maximize the probability of error-tolerant circuit operation, and compare its overhead to that of other design objectives.

## 2. Transient Fault Modeling

We first introduce a general, technology-independent model for transient faults and errors. 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 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,  $\delta$  is the next-state function and  $\lambda$  is the output function. A *single transient fault (STF)* in  $C$ , denoted  $f(l/p, x, s)$  is defined by the following properties: (i) it causes line  $l$  to be stuck-at- $p$ , where  $p$  is 0 or 1, for one clock cycle; and (ii) 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 by no means intractable. 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 happens to occur 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. Like SAFs, STFs need not precisely mimic physical defects to provide useful information about the defects' behavior and detection requirements.

The basic STF model is relatively simple. Nevertheless, complex non-deterministic effects can be deduced from STFs. Consider the question: What is the probability of an SEU producing an erroneous output from  $C$  within  $t \geq 0$  cycles of the fault's occurrence? Assume that the faults of interest appear and disappear within one clock cycle. Such faults tend to occur at random times and are likely to affect all (total) states of  $C$  equally. Hence we can reasonably treat all STFs as equiprobable. By fault-simulating  $C$  for  $t$  cycles with an initial state defined by each possible STF  $f$ , we can answer the above question. For some circuits, it is possible (as we show later) to determine error probabilities analytically.

Erroneous behavior depends on many interacting physical factors [DM03]. For this reason, studies of transient errors rely heavily on electrical or probabilistic models that are technology- or application-dependent. While the STF model as defined above intentionally excludes low-level (physical) information, such information can be integrated into the modeling framework by attaching to a fault or a group of faults, a weight denoting a factor like susceptibility to radiation strikes. The STF

model can also be made to account for pattern sensitivity as described below.

Consider the two-input CMOS gate NAND2 in Fig. 1. A radiation strike can upset one or more of its transistors, causing 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. Thus if only one transistor is upset at a time, and the upset probability is identical for all four transistors, the soft error susceptibility or *upset probability* of NAND2 can be set to some value  $p$  under input 10 or 01,  $2p$  under input 11, and zero under input 00. The concepts introduced here can readily be extended to STFs weighted by physical upset probabilities of this kind [MT03, PHKB05].

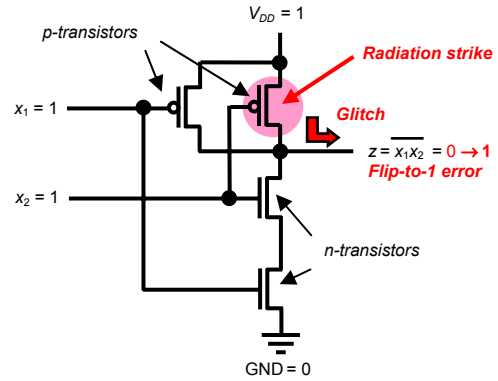


Figure 1: NAND2 gate illustrating a transient flip-to-1 error

## 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 STF set  $F$  with *non-critical error period (NEP)*  $k$  and *self-recovery probability*  $p_{sr}$ , denoted  $(F, k, p_{sr})$ -TET, if the internal states of the erroneous and error-free circuits are the same after  $k$  cycles with probability 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  $p_{sr} = 1.0$ , 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  $F$ . It has been

shown [PB+06] that a motion estimation circuit is TET with  $p_{sr} = 1.0$  for over 70% of its faults with period 96. A combinational circuit is thus (1,1.0)-TET since it recovers from a 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.

The frequency with which errors occur is referred to as the soft error rate SER.. We define the *critical soft error rate* (CSER) as the frequency of critical errors. We assume that the system specifications include parameters  $k$  and  $p_{sr}$ . Errors for which the circuit is  $(k, p_{sr})$ -TET are excluded from the CSER metric. CSER can be calculated as the SER multiplied by the probability that an error is critical.

#### 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 an 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:

$$p_{err}(z) = \left( \sum_{\ell} \text{No. of tests for faulty line } \ell \right) / k 2^{n+1} \quad (1)$$

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

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

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 “linear” gate of the X(N)OR type,  $p_{err}(z) = 1/2$ . Elementary and linear gates are 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

$$1/(2k) \leq p_{err}(z) \leq 1/2 \quad (3)$$

Hence STFs capture our intuitive notions of transient error propagation and masking quite well.

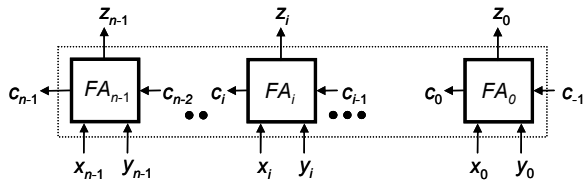


Figure 2: An  $n$ -bit ripple-carry (RC) adder.

#### 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: local errors due to faults in  $FA_i$  itself, and remote errors that originate in preceding stages and enter  $FA_i$  via  $c_{i-1}$ . 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  $z_i$ .

$$p_{err}(z_i) = [2^{2n+3} + 2^{2(n-i)}(e(c_{i-1}) - 2^{2i+1})] / (4n + 1) 2^{2n+2}$$

Size $n$	Carry $c_{n-1}$	$z_0$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$
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

Figure 3: STF error probabilities in RC adders.

Figure 3 shows some  $p_{err}$  values derived from this analysis. Such data provides 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 carry-out is always less than that of the  $z_i$  (sum) outputs.

#### 4.2 Calculation of $p_{err}$

Consider a general,  $n$ -input combinational circuit  $C$  with  $m$  output functions  $Z = z_1, z_2, \dots, z_m$ . The STF error probability of output  $z_i$  can be expressed as

$$p_{err}(z_i) = \sum_{f \in SAF} |z_i \oplus z_i^f| / |SAF| \cdot 2^n$$

where  $SAF$  is the set of all the stuck-at faults (not the STFs) in  $C$ ,  $z_i$  is the function at the  $i^{\text{th}}$  output,  $z_i^f$  is the same function with fault  $f$  present, and  $|\dots|$  indicates set cardinality. This equation can be evaluated efficiently using symbolic simulation with BDDs representing  $Z$ . It can also be approximated using random-pattern simulation or techniques from [AT05]. The circuit’s error probability  $p_{err}(Z)$  considering all  $m$  outputs is expressed by

$$p_{err}(Z) = \sum_{f \in SAF} \left| \bigcup_{i=1}^m (z_i \oplus z_i^f) \right| / |SAF| \cdot 2^n \quad (4)$$

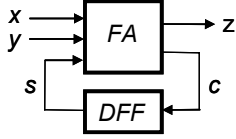


Figure 4: RTL model of a serial adder  $SA$ .

## 5. Errors in Sequential Circuits

We illustrate our TET concepts and formalisms using a serial adder as an example, and then propose a general method to calculate TET properties.

### 5.1 Serial Adder

To illustrate TET and STF-induced error behavior in sequential logic, consider the serial adder  $SA$  in Fig. 4. This 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 a total of 80 STFs.

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. Figure 5 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 Eq. (3), half the possible STFs are undetectable, so  $SA$  is  $(F_0, 0, 1.0)$ -TET. Class  $F_1$  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% 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  $0.75 + 0.5 \cdot 0.25 = 0.875$ .

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

Figure 5: Classification of all STFs affecting  $SA$  of Fig. 4.

It is easily seen that the STFs in Fig. 5 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 [Bre73]. We use them to compute the probability  $p_{\text{good}}$  of the circuit going from erroneous states induced by STFs to correct states within  $k$  clock cycles. We can then say the circuit is  $(k, p_{\text{good}})$ -TET.

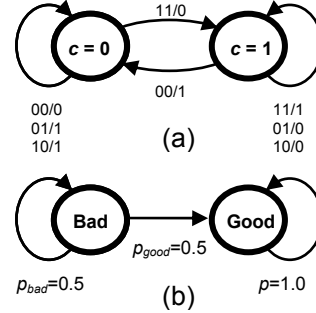


Figure 6: (a) State transition graph, and (b) Markov model for the serial adder  $SA$ .

Considering  $SA$  again, its state transition graph is in Fig. 6a. 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. 6b. 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 probability of self-recovery  $1 - 0.5^k$ . Figure 7 shows how the probability  $p_{\text{err}}(k)$  of an error lurking in  $SA$  decreases exponentially with time. Thus we can, in cases like this, derive an analytic formula for  $p_{\text{err}}(k)$  that can be used to determine error tolerance with respect to given thresholds on  $p_{\text{err}}(k)$  or  $k$ .

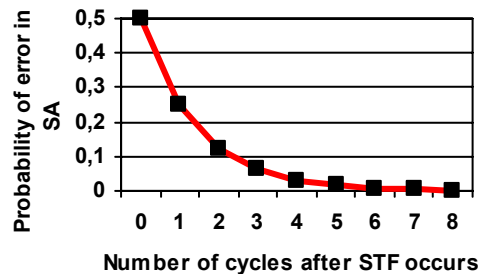


Figure 7: Probability  $p_{\text{err}}(k)$  of an error in  $SA$ 's state  $k$  cycles after an STF.

For large or unstructured sequential circuits, computer simulation can be used to determine  $p_{\text{err}}(k)$  numerically. The probability that the circuit's state is still erroneous after  $k$  clock cycles can be found by constructing a combinational  $k$ -time-frame expansion of the circuit and calculating  $p_{\text{err}}$  via Eq. (4). The primary inputs of the expanded circuit are the primary inputs of time frames 0 through  $k$  and the flip-flops of time frame 0. The primary outputs are those of the flip-flops in time frame  $k$ . The probability of self-recovery  $p_{\text{sr}}$  is obtained as  $1 - p_{\text{err}}$ .

## 6. Selective Hardening

Suppose that a circuit's specification requires it to be  $(k, p_{\text{target}})$ -TET. It is possible to check whether this specification is met by calculating  $p_{\text{err}}$  for the  $k$ -time-frame circuit as described in Sec. 4. If the computed  $(1 - p_{\text{err}})$  is greater than or equal to  $p_{\text{target}}$ , then the specification is met. Otherwise, it requires redesign using, for example, hardening techniques such as those mentioned in the introduction.

We assume that hardening is applied to nodes (gates) rather than to individual STFs. If a node is hardened, all its associated STFs are assumed to be hardened irrespective of the input pattern. We assume it is possible to selectively harden a node against flip-to-1, against flip-to-0, against both flip-to-1 and flip-to-0, or not to harden it all. We also assume that the cost of hardening a node against flip-to-1 or flip-to-0 is one cost unit  $d$  for all nodes, and that the cost of hardening against both flip-to-1 and flip-to-0 is  $2d$ . We further assume that the probability of a soft error on a hardened node is 0.

Suppose the goal is to make the circuit meet some TET specification at the lowest possible cost. For this purpose, a minimal number of nodes is hardened such that the probability of self-recovery  $p_{\text{sr}}(k)$  is  $p_{\text{target}}$  or more. To achieve this, we first observe that hardening a node against a flip-to-0 (flip-to-1) has the same effect as excluding the test set size  $\left| \bigcup_{i=1}^m (z_i \oplus z_i^f) \right|$  for the corresponding stuck-at-0 (stuck-at-1) fault from Eq. (4). Then we sort all SAFs according to their test set size and harden the faults from the sorted list until  $p_{\text{sr}}(k) \geq p_{\text{target}}$  holds.

For example, suppose that  $SA$  must be TET with  $k = 2$  and  $p_{\text{target}} = 93\%$ , i.e., the adder must self-recover after two cycles with probability 0.93 or more. From Fig. 7, we see that  $p_{\text{err}}(2) = 0.125$  (i.e.,  $p_{\text{sr}} = 0.875$ ), which means that  $SA$  does not meet the specification and requires hardening. The calculation of the test set sizes for SAFs is illustrated by the stuck-at-0 fault on line  $x$ , denoted  $x/0$ . 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 is four. Similarly, the test set size is also four for faults  $x/1, y/0, y/1, s/0$  and  $s/1$ ; it is eight for faults  $c_{in}/0$  and  $c_{in}/1$ ; and it is zero for faults  $z/0$  and  $z/1$ .

A fault with the largest test set size is selected first, e.g.,  $c_{in}/0$ . This reduces  $p_{\text{err}}(2)$  from 0.125 to 0.1, which still exceeds  $1 - p_{\text{target}} = 0.07$ . Fault  $c_{in}/1$  is selected next, resulting in  $p_{\text{err}}(2) = 0.075$ . Selecting a third fault such as  $x/0$ , results in  $p_{\text{err}}(2) = 0.0625$ , which is below  $1 - p_{\text{target}}$ . The specification has been met ( $p_{\text{sr}} = 0.9375$ ) by hardening node  $c$  against both flip-to-0 and flip-to-1, and hardening node  $x$  against flip-to-0 only. The hardening cost is  $3d$  or 30% of the  $10d$  cost of hardening all nodes. Note that by selecting only the most critical nodes to harden, the achieved reduction of 50% for  $p_{\text{err}}(2)$  exceeds the proportion of hardened nodes (30%).

Cct.	$p_{\text{err}}$	$p_{\text{target}} = 0.9$		$p_{\text{target}} = 0.99$		$p_{\text{target}} = 0.999$		$p_{\text{target}} = 0.9999$	
		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

**Figure 8:**  $p_{\text{err}}(k)$  and cost of selective hardening for the ISCAS-85 combinational benchmarks.

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  $SA$ ,  $p_{\text{err}}(2) = p_{\text{err}}(0) \cdot (0.5)^2$  holds, it suffices to select faults considering only the probability that the circuit enters state  $Bad$  in the beginning. To meet the specification, this probability must be below  $p_{\text{target}}/(0.5)^2 = 0.28$  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. Experimental Results

Using symbolic simulation, we calculated the effect of selective hardening on  $p_{\text{err}}$  for nine combinational ISCAS-85 benchmark circuits for which BDD-based analysis was feasible. Figure 8 gives  $p_{\text{err}}$  (with  $k = 0$ ) for a circuit without any hardening and, for four values of  $p_{\text{target}}$ , the cost of selective hardening to achieve  $(1 - p_{\text{err}}) \geq p_{\text{target}}$  (which is the number of stuck-at faults excluded from Eq. (4)) and the

Circuit	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$	$k = 10$
s27	2,41E-01	8,07E-02	3,57E-02	1,55E-02	6,70E-03	2,88E-03	1,23E-03	5,27E-04	2,25E-04	9,62E-05
s298	1,46E-01	6,11E-02	2,96E-02	1,37E-02	6,76E-03	3,32E-03	1,67E-03	8,63E-04	4,34E-04	2,19E-04
s208	1,18E-01	1,71E-02	4,14E-03	1,03E-03	2,61E-04	6,47E-05	1,61E-05	4,03E-06	1,17E-06	2,54E-07
s344	1,92E-01	1,03E-01	5,16E-02	2,76E-02	1,42E-02	7,22E-03	3,61E-03	1,81E-03	9,03E-04	4,51E-04
s349	1,91E-01	1,03E-01	5,17E-02	2,77E-02	1,43E-02	7,22E-03	3,61E-03	1,81E-03	9,03E-04	4,51E-04
s382	1,50E-01	5,37E-02	2,65E-02	1,31E-02	6,53E-03	3,25E-03	1,62E-03	8,09E-04	4,04E-04	2,01E-04
s386	4,78E-02	2,02E-02	7,17E-03	2,38E-03	8,28E-04	2,96E-04				
s400	1,48E-01	5,23E-02	2,58E-02	1,28E-02	6,37E-03	3,17E-03				
s420	1,14E-01	8,94E-03	2,05E-03	5,12E-04	1,29E-04	3,21E-05				
s444	1,43E-01	5,19E-02	2,56E-02	1,27E-02	6,31E-03	3,14E-03				
s510	1,14E-01	1,12E-01	1,04E-01	1,00E-01	9,70E-02	9,46E-02				
s526	1,22E-01	5,51E-02	2,81E-02	1,41E-02	7,12E-03	3,57E-03				
s641	1,87E-01	7,74E-02	4,22E-02	2,40E-02						
s713	1,81E-01	7,46E-02	4,08E-02	2,33E-02						
s820	2,28E-02	9,17E-03	3,98E-03	1,79E-03						
s832	2,24E-02	9,02E-03	3,91E-03	1,76E-03						
s953	1,92E-01	7,91E-02	5,06E-02	4,69E-02						
s1196	6,36E-02	3,43E-03	2,37E-04	0,00E+00						
s1488	2,23E-02	1,02E-02	5,00E-03	2,47E-03						
s1494	2,22E-02	1,01E-02	4,99E-03	2,46E-03						
s1238	6,16E-02	3,29E-03	2,28E-04							
s1423	1,66E-01	8,33E-02								
s5378	1,67E-01	7,05E-02								

Figure 9:  $p_{\text{err}}(k)$  for ISCAS-89 sequential benchmarks

percentage of these faults among all faults. Equation (4) has been implemented by BDD operations using the CUDD package and arbitrary-precision arithmetic to handle large numbers.

The probability  $p_{\text{err}}$  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. The cost of selective hardening with  $p_{\text{target}} = 0.1$  is generally quite low. In contrast, lower values of  $p_{\text{target}}$  require overheads which are probably unacceptable. Recall that  $p_{\text{target}}$  could assume higher values for combinational circuits as the faulty effect will definitely last only for one clock cycle. Hence, selective hardening is useful for combinational circuits if  $p_{\text{target}}$  is not much larger than  $p_{\text{err}}$ , which is likely to be the case.

Figure 9 shows  $p_{\text{err}}(k)$  for the sequential ISCAS-89 circuits and various values of  $k$ . Note that  $p_{\text{err}}(k)$  is the probability that the error still affects the circuit state after  $k$  cycles and that it decreases with  $k$ . Clearly, this decrease is significant. In contrast, the probability that an error will show up in any of cycles 1 through  $k$  is estimated in [AT05], and that metric increases with  $k$  as erroneous effects in cycles 1 through  $k$  do not count towards our metric but do count towards the metric in [AT05].

The probability  $p_{\text{bad}}(k)$  that the circuit which is in an erroneous state after  $k$  cycles stays in an erroneous state after  $k + 1$  cycles can be calculated as  $p_{\text{bad}}(k) = p_{\text{err}}(k + 1)/p_{\text{err}}(k)$ . If  $p_{\text{bad}}(k)$  is independent of  $k$ , then the circuit can be represented by a Markov model such as the one in Fig. 6b, with  $p_{\text{bad}}$  being the probability that the system stays in state *Bad* and  $(1 -$

$p_{\text{bad}}$ ) being the transition probability from state *Bad* to state *Good*. (In the example, both these probabilities equal 0.5). A Markov model significantly reduces the computational effort needed for analysis.

Figure 10 shows the values of  $p_{\text{bad}}(k)$  calculated from the data in Fig. 9. It can be seen that  $p_{\text{bad}}(k)$  can vary significantly for different circuits, the extreme cases being s420 and s510. For  $k > 1$  the value of  $p_{\text{bad}}(k)$  tends to remain nearly constant. It seems that knowing  $p_{\text{err}}(k)$  for  $k = 1, 2$  and  $3$  is sufficient for an accurate analysis in most cases, and it is possible to approximate  $p_{\text{err}}(k)$  for larger  $k$ 's with high accuracy.

Cct.	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$
s27	0.34	0.44	0.43	0.43	0.43	0.43	0.43	0.43	0.43
s298	0.42	0.48	0.46	0.49	0.49	0.50	0.52	0.50	0.51
s208	0.14	0.24	0.25	0.25	0.25	0.25	0.25	0.29	0.22
s344	0.54	0.50	0.54	0.52	0.51	0.50	0.50	0.50	0.50
s349	0.54	0.50	0.54	0.51	0.51	0.50	0.50	0.50	0.50
s382	0.36	0.49	0.50	0.50	0.50	0.50	0.50	0.50	0.50
s386	0.42	0.35	0.33	0.35	0.36				
s400	0.35	0.49	0.50	0.50	0.50				
s420	0.08	0.23	0.25	0.25	0.25				
s444	0.36	0.49	0.50	0.50	0.50				
s510	0.98	0.93	0.97	0.97	0.97				
s526	0.45	0.51	0.50	0.50	0.50				
s641	0.41	0.55	0.57						
s713	0.41	0.55	0.57						
s820	0.40	0.43	0.45						
s832	0.40	0.43	0.45						
s953	0.41	0.64	0.93						
s1488	0.46	0.49	0.49						
s1494	0.46	0.49	0.49						

Figure 10:  $p_{\text{bad}}$  for ISCAS-89 sequential benchmarks.

TET and selective hardening is evaluated in Fig. 11 for the benchmark circuit s298. It can be seen that this

kind of hardening is indeed a low-cost way to achieve a given  $p_{\text{target}}$  if non-reference behavior is acceptable for a few clock cycles.

NEP $k$	$p_{\text{target}} = 0.9$		$p_{\text{target}} = 0.99$		$p_{\text{target}} = 0.999$		$p_{\text{target}} = 0.9999$	
	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	24.5	252	42.28
6	0	0	0	0	102	17.11	208	34.9
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

Figure 11: Selective hardening of sequential circuit s298

## 8. Conclusions

We have introduced the concept of transient-error tolerance (TET) which grades soft errors according to their impact on circuit functionality, in particular, their ability to disturb the circuit state for extended periods of time. We analyzed combinational and sequential behavior by means of a new single transient fault (STF) model, which is technology-independent but can be easily extended to account for low-level (electrical) information. We defined the STF error probability  $p_{\text{err}}$  and the probability of self-recovery  $p_{\text{sr}}$ , which can serve as metrics for soft error susceptibility/tolerance of a design during logic synthesis. We also obtained analytical and experimental data using the ISCAS benchmarks. A synthesis procedure optimizing  $p_{\text{err}}$  will tend to use more elementary gates of larger size, which is consistent with our soft error analysis methodology. The implications of imposing such a requirement during synthesis on area, delay and power consumption form an interesting research question [AMYV06].

We also studied circuits that may deviate from their correct behavior for a number of cycles (the non-critical error period NEP) and determined the probability that this specification is violated. We defined the critical SER (CSER) metric, which excludes tolerable errors. It provides more realistic error-tolerance estimates and may reduce hardware overhead. For both combinational and sequential circuits, we introduced a selective hardening strategy to improve this metric. The proposed technique can reduce the soft error rate when limited resources are available for hardening. Besides an exact BDD-based analysis method, we outlined the construction of approximate Markov models and discussed ways to achieve a specified level of accuracy.

## Acknowledgements

This work was supported in part by the Alexander von Humboldt Foundation, the DFG Project *RealTest* under Grant BE 1176/15–1, and the U.S. Air Force Research Laboratory under Agreement No. FA8750-05-1-0282.

## References

- [AMYV06] S. Almkhaizim et al. “Seamless integration of SER in rewiring-based design space exploration.” *Proc. ITC*, 2006.
- [AT05] H. Asadi & M. Tahoori: “Soft error modeling and protection for sequential elements.” *Proc. Symp. DFT*, pp.463–471, 2005.
- [BT89] B. Krishnamurty & I.G. Tollis: “Improved techniques for estimating signal probabilities.” *IEEE Trans. Computers*, vol. 38, pp. 1041-1045, 1989.
- [Bau05] R. Baumann. “Radiation-induced soft errors in advanced semiconductor technologies.” *IEEE Trans. Device & Materials Reliability*, vol. 5, pp. 305-316, 2005.
- [Bre73] M.A. Breuer. “Testing for intermittent faults in digital circuits.” *IEEE Trans. Computers*, vol. C-22, pp. 241-146, 1973.
- [Bre04] M.A. Breuer. “Determining error rate in error tolerant VLSI chips.” *Proc. DELTA Wkshp. Electronic Design, Test and Apps.*, pp. 321–326, 2004.
- [DM03] P.E. Dodd & L.W. Massengill. ” Basic mechanisms and modeling of single-event upset in digital microelectronics.” *IEEE Trans. Nuclear Sci.*, vol. 50, pp.583-602, 2003.
- [HN06] T. Heijmen & A. Nieuwland. “Soft-error rate testing of deep-submicron integrated circuits.” *Proc. ETS*, pp. 247-252, 2006.
- [JR+06] V. Joshi et al. Logic SER Reduction through Flipflop Redesign. *Proc. ISQED*. 2006.
- [KKK06] W.Y.Kung et al. “Spatial and temporal error concealment techniques for video transmission over noisy channels.” *IEEE Trans. CAS Video Tech.*, vol. 16, pp.789-802, 2006.
- [MT03] K. Mohanram & N.A. Touba. “Cost-effective approach for reducing soft error failure rate in logic circuits.” *Proc. ITC*, pp.893-901, 2003.
- [PM75] K.P. Parker & E.J. McCluskey. “Probabilistic treatment of general combinational circuits.” *IEEE Trans. Computers*, vol. C-24, pp. 668-670, 1975.
- [PHKB05] I. Polian et al. “Transient fault characterization in dynamic noisy environments.” *Proc. ITC*, pp.1039-1048, 2005.
- [PB+06] I. Polian et al. “Low-cost hardening of image processing applications against soft errors.” *Proc. DFT*, pp.274-279, 2006.
- [Sav80] J. Savir. “Testing for single intermittent failures in combinational circuits by maximizing the probability of fault detection.” *IEEE Trans. Computers*, vol. C-29, pp. 410-416, 1980.
- [SM76] J.J. Shedtetsky & E.J. McCluskey. “The error latency of a fault in a sequential digital circuit.” *IEEE Trans. Computers*, vol. C-25, pp.655-659, 1976.
- [ZDB05] C. Zhao et al. “Soft-spot analysis: targeting compound noise effects in nanometer circuits.” *IEEE Design & Test*, vol. 22, pp.362–375, July 2005.
- [ZM+07] M. Zhang, S. Mitra et al. Sequential Element Design with Built-In Soft Error Resilience. *IEEE Trans. VLSI*, to appear