A Model for Transient Faults in Logic Circuits

John P. Hayes[*] Ilia Polian[§] and Bernd Becker[§]

[*] University of Michigan
2260 Hayward Street
Ann Arbor, MI 48109, USA
jhayes@eecs.umich.edu

[§] Albert-Ludwigs-University
Georges-Köhler-Allee 51
79110 Freiburg im Breisgau, Germany
{polian|becker}@informatik.uni-freiburg.de

Abstract: **Transient (soft) faults due to particle strikes and other environmental and manufacturing effects are a frequent cause of failure in ICs. We propose a general, technology-independent model called single transient fault (STF) model to represent transient faults and errors in logic circuits. It is defined in terms of a temporary stuck-at fault and its associated circuit state. STFs can be used to estimate the transient error probability p_{err} of a circuit's nodes, as well as various measures of reliability and error tolerance. We demonstrate the use of STFs with combinational and sequential logic circuits, including several types of adders. Some other applications of STFs are also briefly considered.**

Keywords: Fault modeling, error probability, failure analysis, soft errors, transient faults.

1. Introduction

As IC-based systems are scaled down to the very deep submicron range, they are increasingly sensitive to radiation strikes and similar single-event upsets (SEUs), which produce transient (soft) errors, whose impact ranges from a minor glitch to a major crash [DM03, HN06]. The erroneous behavior depends on many interacting physical factors, such as where the SEU occurs, how much energy it has, the circuit's current state, and the strike time relative to the system clock cycle. Hence previous studies of transient errors have relied heavily on electrical or probabilistic models that are technology- or application-dependent and computationally complex [MM06, ZDB05].

Our aim is to develop a general, technology-independent model for transient faults in logic circuits, which is useful for analyzing and simulating transient errors, as well as their impact on circuit reliability. Such models are also useful for synthesizing circuits that are resistant to, or tolerant of, soft errors [AT05, Bre04, MT03]. The target circuits are assumed to be synchronous digital circuits composed of logic gates, flip-flops, register-transfer-level (RTL) elements, and the like.

2. Fault Model

Let $C = (I,O,S,\delta,\lambda,S_0)$ be a sequential circuit with k logic lines. 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 $p = 0$ or 1, for 1 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 $2k|I||S|$. While this number is large, STFs are by no means intractable. There is no cause-effect relationship between an STF and its associated state; $f(l/p,x,s)$ is a fault that happens to occur in state x [BH03].

The STF model is clearly related to the standard stuck-at fault (SAF) model. Unlike an STF, an SAF l/p persists 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 test requirements.

The STF model is static and deterministic, which are key to its simplicity and tractability. Nevertheless, complex probabilistic effects can be derived from STFs. Consider the question: What is the probability of an erroneous output from C within t cycles of an STF's occurrence? Such faults tend to occur at random times and are likely to affect all states equally. Hence we can treat all STFs as equiprobable. By fault-simulating C for t cycles with an initial state defined by each possible STF, we can answer the foregoing question. For circuits with regular structures, it is possible, as we demonstrate here, to determine error probabilities analytically.

3. Combinational Circuits

In this case, an STF reduces to the form $f(l/p,x)$ and an STF-induced error corresponds to an SAF l/p and a test x for l/p. Let C have k lines, n inputs, and a single output z, and assume that all STFs are equiprobable. The *STF error probability* $p_{\mathrm{err}}(z)$ is the total number of possible errors produced at z by STFs, divided by the total number of possible STFs:

$$p_{err}(z) = \left(\sum_{\ell} \text{No. of tests for line } \ell\right) \Big/ k2^{n+1} \qquad (1)$$

Suppose that C is an n-input "elementary" gate G of the (N)AND, (N)OR or NOT type. Equation (1) implies that the probability of an STF error in G is

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

Here $p_{\mathrm{err}}(z)$ approaches $1/(2(n + 1)) = 1/(2k)$ as n increases, which implies that gates with greater fan-in are less sensitive to STFs, that is, are more likely to tolerate STF errors. In the case of a "linear "gate of the XOR or XNOR type, $p_{\mathrm{err}}(z) = 1/2$. Hence for any k-line single-output combinational circuit

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

Adder example: Consider the n-bit ripple-carry adder *RCA* shown in Fig. 1. It consists of n copies of the full adder FA_i, which realizes the sum $z_i = x_i \oplus y_i \oplus c_{i\text{-}1}$ and the carry-out $c_i = x_iy_i + x_ic_{i\text{-}1} + y_ic_{i\text{-}1}$. *RCA* has n full-adder stages, $2n + 1$ inputs (including an external carry-in line), and $n + 1$ outputs (including a carry-out line). There are $4n + 1$ lines that can be faulty, so the total number of possible STFs is $(4n + 1)2^{2n+2}$.

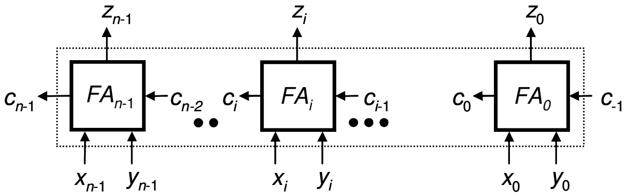

Figure 1: An n-bit ripple-carry (RC) adder *RCA*.

A 1-bit *RCA* is just a single copy of FA_i with $i = 0$, $p_{\mathrm{err}}(z_0) = 0.4$ and $p_{\mathrm{err}}(c_0) = 0.25$. Every n-bit *RCA* has z_0 as an output line but the corresponding $p_{\mathrm{err}}(z_0)$ values vary with n and k. We can compute $p_{\mathrm{err}}(z_i)$ in the general n-bit case by counting the errors produced at the adder outputs by all possible STFs. The errors on z_i fall into two groups: local errors due to faults in FA_i itself, and remote errors that originate in the preceding i stages $0,1,\ldots,i-1$ and enter FA_i via its carry-in line $c_{i\text{-}1}$. All remote errors propagate to z_i and some of them propagate to c_i. The local error count at z_i is 2^{2n+3}. The corresponding remote error count is

$$e(c_i) = 2^{2n+2} + 2e(c_{i\text{-}1}) \qquad (4)$$

leading to the following formula for the total number of STF errors transmitted to z_i.

$$e(z_i) = 2^{2n+3} + 2^{2(n-i)}\,[e(c_{i\text{-}1}) - 2^{2i+1}]$$

On dividing $e(z_i)$ by the total number of STFs, we get the STF error probability on z_i.

$$p_{\mathrm{err}}(z_i) = [2^{2n+3} + 2^{2(n\text{-}\mathrm{i})}\,(e(c_{i\text{-}1}) - 2^{2i+1})]/(4n + 1)2^{2n+2} \quad (5)$$

Size n	**Carry $c_{n\text{-}1}$**	**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 2: STF error probabilities for various RC adders.

Figure 2 shows some p_{err} values derived from this analysis (and confirmed independently by circuit simulation). This kind of data provides useful information about a circuit's error propagation or masking properties. For example, the $p_{\mathrm{err}}(z_i)$'s of the RC adder increase slowly with i, eventually leveling off. The error probability $p_{\mathrm{err}}(c_{n\text{-}1})$ at the carry-out is always less than that of the z_i (sum) outputs.

Fanout-free circuits: An interesting special case for p_{err} calculation is an n-input fanout-free circuit C. Let C_i be the subcircuit of C whose primary output is i, and let X_i denote C_i's set of n_i primary inputs. X_i is therefore i's support set and n_i is the fan-in of i. We also need the controllability numbers c_i^0 and c_i^1, defined as the numbers of vectors X_i that apply 0 and 1, respectively, to i. Obviously, $c_i^0 + c_i^1 = 2^{ni}$. All these numbers can be easily computed for every i by a single input-to-output scan of C.

We use n_i, c_i^0 and c_i^1 to compute an STF error count e_i for each line i as was done in Equation (2). Again e_i is the number of errors applied to i by all possible tests for SAFs in C_i. To calculate the e_i's, we scan C once from primary inputs to primary output and for each gate's output, we calculate a new error count from the error counts on its inputs. The final error count e_z on the primary output z of C leads to $p_{\text{err}}(z) = e_z/k2^{n+1}$.. The key steps to compute the e_i's are as follows:

1. If i is a primary input, then $e_i = 2$.

2. Let i be the output of a gate G whose inputs are $j_1, j_2, \ldots, j_g$. Then

$$e_i = (\sum_{h=1:g} e_{jh}(\prod_{k \neq h} c_{jk}^{p})) + 2^{n_i} \tag{6}$$

where $p = 1$ for an AND or NAND gate and $p = 0$ for an OR or NOR gate.

If (6) is divided by $k/2^{n_i}$, then we get an algorithm for propagating error probabilities, or calculating p_{err} on the fly.

For example, if G is a 3-input NAND gate in C with inputs 1,2,3, output i, and fan-in 7, then by (6)

$$e_i = (e_1 c_2^1 c_3^1 + e_2 c_1^1 c_3^1 + e_3 c_1^1 c_2^1) + 2^7 \tag{7}$$

The term in parentheses represents all the ways that errors occurring up to G can be propagated through G by input vectors (tests) of the 7-input circuit C_i. Observe that each input h requires every other input $k \neq h$ to be set to 1; this can be done in c_k^1 ways because C_i is fanout-free. The 2^7 term in (7) is the number of errors contributed by line i itself, since every input vector of C_i detects either the stuck-at-0 or stuck-at-1 fault on i.

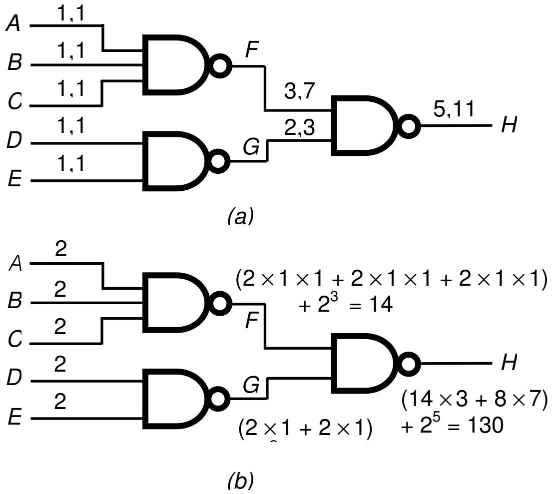

Figure 3: (a) Fanout-free circuit with lines labeled n_i, c_i^1. (b) Calculation of the error count e_i on each line.

Figure 3 shows a small fanout-free circuit C composed of NAND gates to which the foregoing error-counting algorithm has been applied. The final result is $e_H = 130$, which implies that the circuit's STF error probability is $p_{\text{err}} = 130/(8 \times 64) = 0.25390625$.

General combinational circuits: Let C be any n-input combinational circuit with m output functions $Z = z_1, z_2, \ldots, z_m$. The STF error probability at output z_i an be expressed as

$$p_{err}(z_i) = \sum_{f \in SAF} \left| z_i \oplus z_i^f \right| \Big/ |SAF| \cdot 2^n \tag{8}$$

where SAF denotes the set of all the stuck-at faults (not STFs) in C, z_i is the function at the i-th output, z_i^f is the same function with fault f present, and |...| indicates set cardinality. Equation (8) can be evaluated efficiently using computer simulation with BDDs representing Z. It can also be approximated using random-pattern simulation or techniques such as those in [AT05]. The circuit's error probability $p_{\text{err}}(Z)$ considering all m outputs is expressed as follows:

$$p_{err}(Z) = \sum_{f \in SAF} \left| \bigcup_{i=1}^{m} z_i \oplus z_i^f \right| \Big/ |SAF| \cdot 2^n \tag{9}$$

5. Sequential Circuits

To illustrate the application of STFs to sequential logic, we use the RTL model for a serial adder SA shown in Fig. 4. This circuit adds two binary numbers X and Y serially (bit by bit) to produce the sum Z. It consists of a combinational full adder FA and a D flip-flop DFF that is used to store the carry bit c. It contains a total of 80 STFs.

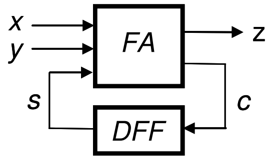

Figure 4: A serial adder SA.

Consider the effect on z and c of an STF f in SA that occurs in the initial clock cycle 0. Figure 5 below gives a 4-way fault classification based on whether or not *f* produces an erroneous value of *z* and/or *c* in clock cycle 0. As expected from Inequality (3), half the faults in Class 0 are undetectable since the normal and faulty values on the faulted line are the same. Class 1 represents the case where *SA's* output, but not its internal state, is erroneous. If an error of duration 1 cycle is acceptable at *SA*'s outputs, then all Class 0 and 1 errors, and thus 75% of all STFs, are tolerable.

STF class	**Class definition**	**Number 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 STFs affecting *SA*.

Class 3 includes a few STFs that leave errors lurking indefinitely in *SA*'s internal state. These errors may be eliminated by suitable design methods, or they may be flushed out automatically by normal input sequences that take the circuit to a correct state. As we show next, the probability of such automatic recovery can be analyzed by Markov methods.

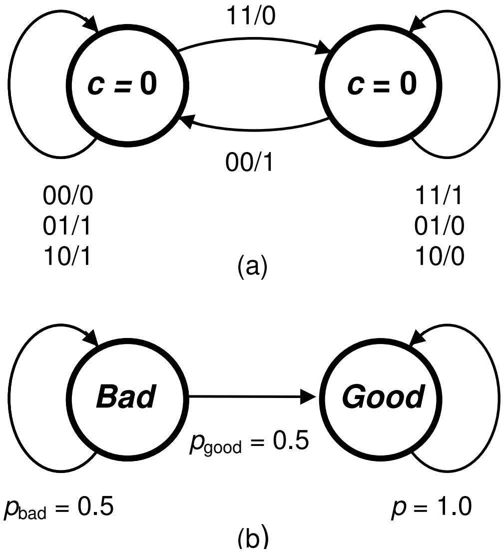

Figure 6: (a) State transition graph, and (b) Markov model for *SA*.

The state transition graph for *SA* is shown in Fig. 6*a*. For half the input combinations *xy* the next internal state *c* (but not *z*) is independent of the initial state: *xy* = 00 always sets *c* to 0, while *xy* = 11 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 the probability of an error-correcting transition from either of *SA*'s two internal states is 1/2. Once it has returned to a good state, *SA* operates correctly until a new fault occurs. This leads to the simple Markov model shown in Fig. 6*b*. 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.

Figure 7 shows how the probability $p_{\text{err}}(t)$ of an error lurking in *SA* decreases exponentially with time. Thus, in this case, we can derive an analytic formula for $p_{\text{err}}(t)$ that can be used to determine error tolerance with respect to given thresholds on p_{err} or *t*.

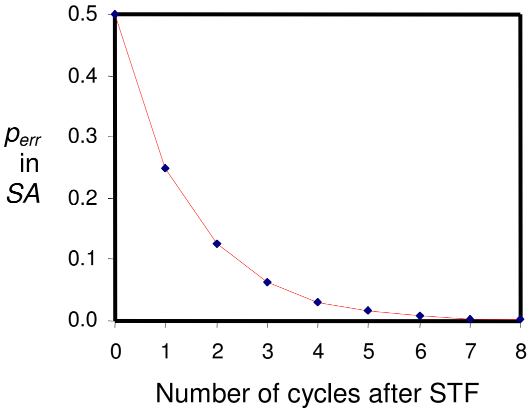

Figure 7: Probability of an error in *SA*'s (total) state after a single transient fault.

6. Discussion

The STF model appears to be a powerful and computationally tractable way to represent and analyze transient faults and errors in logic circuits. Its key advantages are its generality and technology independence. We have demonstrated its use for computing output error probabilities in both combinational and sequential circuits.

While the basic STF model presented here deliberately does not incorporate low-level (physical) information, such information can be easily integrated into the modeling framework. To illustrate this, we outline how the STF model can be extended to account for pattern sensitivity.

Consider the two-input CMOS gate NAND2 in Fig. 8. A radiation strike can upset one or more of its transistors, causing output signal z to temporarily undergo a flip-to-0 or flip-to-1 error. The specific transient error depends on the input pattern x_1x_2 at the time of the strike (and other factors we do not consider). Input $x_1x_2 = 11$ flips z from 0 to 1 if one of the gate's p-transistors is upset by a radiation strike, as depicted in Fig. 8. Under input patterns 10 and 01, only one p-transistor is susceptible to the strike. Under 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 and methods introduced here can readily be extended to STFs weighted by physical upset probabilities of this kind. However, such extensions, as well as the methods needed to determine appropriate p values (see, for example, [PHKB05]) are beyond the scope of this paper.

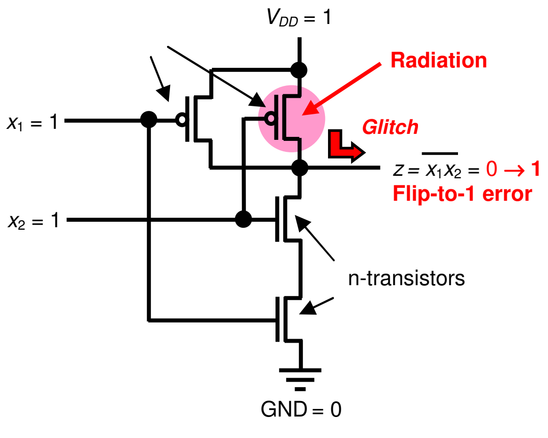

Figure 8: NAND gate with a transient flip-to-0 fault.

Acknowledgements: This work was supported in part by the Alexander von Humboldt Foundation and in part by DFG Project *RealTest* under Grant BE 1176/15–1.

References

[AT05] H. Asadi and M. Tahoori. “Soft error modeling and protection for sequential elements.” *Proc. Symp. DFT*, pp.463–471, 2005.

[BH03] R.D. Blanton and J.P. Hayes: “On the properties of the input pattern fault model.” *ACM Trans. Design Automation Dig. Sys.,* vol. 8, pp.108–124, Jan. 2003.

[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 and 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 and A. Nieuwland: “Soft-error rate testing of deep-submicron integrated circuits.” *Proc. European Test Symp*., pp.247–252, 2006.

[MM06] N. Miskov-Zhivanov and D. Marculescu: “MARS-C: modeling and reduction of soft errors in combinational circuits.” *Proc. Des. Autom. Conf*., pp.767–772, 2006.

[MT03] K. Mohanram and N.A. Touba: “Cost-effective approach for reducing soft error failure rate in logic circuits.” *Proc. Int’l Test Conf*., pp.893–901, 2003.

[PHKB05] I. Polian, J.P. Hayes, S. Kundu and B. Becker: “Transient fault characterization in dynamic noisy environments.” *Proc. Int’l Test Conf*., pp.1039–1048, 2005.

[ZDB05] C. Zhao, S. Dey and X. Bai: “Soft-spot analysis: targeting compound noise effects in nanometer circuits.” *IEEE Design & Test*, vol. 22, pp.362–375, July 2005.