Design as You See FIT: System-Level Soft Error Analysis of Sequential Circuits

Daniel Holcomb   Wenchao Li   Sanjit A. Seshia
EECS Department, UC Berkeley
{holcomb,wenchaol,sseshia}@eecs.berkeley.edu

Abstract

Soft errors in combinational and sequential elements of digital circuits are an increasing concern as a result of technology scaling. Several techniques for gate and latch hardening have been proposed to synthesize circuits that are tolerant to soft errors. However, each such technique has associated overheads of power, area, and performance. In this paper, we present a new methodology to compute the failures in time (FIT) rate of a sequential circuit where the failures are at the system-level. System-level failures are detected by monitors derived from functional specifications. Our approach includes efficient methods to compute the FIT rate of combinational circuits (CFIT), incorporating effects of logical, timing, and electrical masking. The contribution of circuit components to the FIT rate of the overall circuit can be computed from the CFIT and probabilities of system-level failure due to soft errors in those elements. Designers can use this information to perform Pareto-optimal hardening of selected sequential and combinational components against soft errors. We present experimental results demonstrating that our analysis is efficient, accurate, and provides data that can be used to synthesize a low-overhead, low-FIT sequential circuit.

1. Introduction

As technology scales and node capacitances decrease, soft errors due to atmospheric neutrons are a concern even for commodity hardware. Particle strikes can cause errors either by striking state elements directly, or by striking combinational logic and propagating into downstream state elements. Although strikes to state elements have historically caused the most errors, strikes to combinational logic are expected to cause comparably many errors in sub-100nm technologies [19, 5]. In either case, the soft error is observed as an upset in one or more state bits. However, not all combinational nodes or state elements are affected equally. Prior work shows that the properties of combinational logic can result in the soft error rate differing by more than two orders of magnitude across state bits of a single design [16, 22].

While several design techniques for hardening circuits exist (e.g., [23, 21]), they can incur substantial overheads, especially a power overhead, and should be applied judiciously. At the same time, designers must work toward a system-level FIT goal.[1] There is therefore a need for analysis tools that can provide the designer the trade-off between the overheads of circuit mechanisms for error resilience and the system-level FIT value of the circuit. In particular, since not all circuit components are affected equally, as noted above, a tool that pin-points components of the circuit to protect in order to achieve a certain FIT goal would be very useful.

In this paper, we present a novel methodology that performs this kind of analysis for sequential circuits. A failure in this context is at the system level, as detected by monitors derived from formal specifications of functionality or a reference model. The starting point of our methodology is an arbitrary sequential circuit, a set of simulation vectors derived from the circuit's workload, and the specification that the circuit must satisfy. Our analysis first runs the simulation vectors in order to determine the sequences of states that appear while running the circuit's workload. A combinational logic soft error tool called BFIT is used to determine the FIT rate of each of the combinational nodes as well as of the latches in the circuit. The system-level FIT contribution (SFIT) of a latch is then defined as the product of its FIT rate and the probability that an error in the latch causes a system-level failure. The latter is computed by simulating a faulty circuit with monitors constructed from the set of specifications given. The output of our methodology is a list of nodes and latches, ranked according to their contribution to the SFIT rate of the circuit. We show that this information allows designers to make a Pareto-optimal selection of nodes and latches to harden.

The significant contributions of this paper are:

- A novel methodology for assessing the *system-level* failure (FIT) contribution of *individual* logic gates and latches of a *sequential circuit*;
- A new approach to the soft error analysis of combinational logic that addresses *logical, electrical, and timing masking* efficiently and accurately; and
- A demonstration of how the FIT contributions of individual elements (latches and combinational nodes) can be useful in determining which parts of the circuit to harden in order to *efficiently achieve a system-level FIT goal.*

Our system-level analysis can be applied to circuits that are not just traditional processor core circuits such as pipelines and functional units (whose characteristics are well-understood); we can also handle implementations of communication protocols and on-chip interconnection networks. We demonstrate our approach on a chip multiprocessor router [15] and the largest ISCAS'89 benchmarks.

2. Background and Related Work

The methods presented in this paper assess the system level failure contribution of individual logic gates and latches. The relevant related work thus spans the areas of system-level soft error sensitivity and circuit-level soft error analysis.

System-Level Analysis. The probability of an upset in state bits causing a system-level failure depends on the measure of system-level correctness. The upset bits may not impact system outputs at all, or may do so only after significant latency. This makes it difficult to measure system-level impact through simulation and output equivalence.

Miskov-Zivanov and Marculescu model sequential system behavior using Markov chain theory, and evaluate system-level failure rate as the steady-state probability of output nonequivalence [2]. In small circuits, steady-state is shown to be well approximated after a small bound k of cycles subsequent

[1] FIT, standing for "failures in time," is the number of failures encountered in 10^9 hours of operation.

978-3-9810801-5-5/DATE09 © 2009 EDAA

to the bit flip. The runtime of this approach grows with k; the two circuits with the longest runtimes are the only two for which k can exceed 10, in spite of each having less than 100 gates. The largest circuit for which the Markov chain approach is demonstrated is 540 gates, and required only 4 cycles to reach steady state.

A metric to evaluate system-level impact of soft errors in microprocessor cores is the architectural vulnerability factor (AVF) [17, 9]. AVF estimates the probability that a bit flip in various functional blocks of a system will lead to incorrect future execution. Operating at the architectural level, AVF is well suited to very large systems. AVF estimation requires an evaluation of the fraction of time a unit or resource holds a value (instruction or data) that will affect program behavior, which in turn requires detailed models of processor logic and data structures such as pipelines, integer units, and register files. For designs that are not processor cores, such as on-chip interconnection networks, there is little work on AVF estimation.

Analysis of general sequential circuit designs can be done by the verification-guided approach proposed by Seshia et al. [18]. The key insight of this work is that formal assertions can be used as a measure of correctness for soft errors at the system level, without requiring a detailed model of the entire design. Defining failure according to formal specifications instead of block-level output equivalence can be less constraining when appropriate, which can be useful in systems with higher-level resiliency to correct such errors. A drawback to this prior work is that the output is binary, indicating only whether a flip to some bit is capable of violating an assertion, and does not provide the probability of it doing so. In this work, we will combine formal assertions with probabilistic analysis by replacing formal verification with simulation and hardware monitors.

Combinational Circuit Analysis. A soft error can be initiated when a neutron strikes a chip in the vicinity of a reverse-biased p-n junction of a circuit node. The strike creates electron-hole pairs in the substrate, and some of these carriers are collected in the circuit node, causing a transient voltage glitch. If the glitch occurs in a state holding element, its feedback structure can flip the stored bit. If the glitch occurs in a combinational node, it can propagate and be latched in one or more downstream state elements. Depending upon the input vector applied to the circuit when the strike occurs, three masking factors can prevent the glitch from causing an upset in state [11]: *1*) Logical Masking occurs if a struck node does not have a logically sensitized path to a downstream state element; *2*) Timing Masking occurs if a strike propagates to a downstream state element, but arrives while it is not open to a change in state; and *3*) Electrical Masking occurs if a strike is not of sufficient magnitude to upset any downstream state elements.

Timing masking of a single path can be resolved using an analytical approximation instead of exhaustive simulation [19, 16, 20]. The work of Krishnaswamy et al. [10] and of Asadi and Tahoori [1] apply accurate unified analysis of timing and logical masking including multiple sensitized paths, but neglect electrical masking entirely.

Electrical masking is partly due to the attenuation of transient glitches as they propagate through logic towards sequential elements. This attenuation is often modeled using a parameterized representation of a glitch, and using transfer functions to describe its transformation as it propagates through logic [19]. Examples of parameterized glitch representations are pulse height and width [20], trapezoidal shaped waveforms [14], and Weibull functions [16]. While glitch attenuation is diminished in sub-100nm technologies [13, 5], electrical effects are still important, as the FIT contribution of a gate depends strongly on its input states [16, 7, 20].

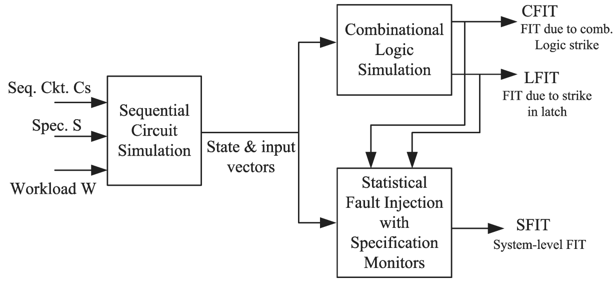

Seq. Ckt. Cs
Spec. S
Workload W
Sequential Circuit Simulation
State & input vectors
Combinational Logic Simulation
CFIT
FIT due to comb. Logic strike
LFIT
FIT due to strike in latch
Statistical Fault Injection with Specification Monitors
SFIT
System-level FIT

Figure 1: Overview of Tool Flow

To properly account for all masking factors, logical, timing, and electrical masking must be considered via a unified analysis. The work of Miskov-Zivanov and Marculescu [3] shows that symbolic techniques can be combined with simplified glitch models to analyze of all masking factors. A similar symbolic technique is used in the FASER tool of Zhang et al. [20]. While the two aforementioned symbolic approaches propagate transient glitches, the SERA tool developed by Zhang et al. [22] avoids propagating glitches entirely. In SERA, logically sensitized paths are extracted, and FIT estimation of each path is performed by approximating the path as an inverter chain for which an analytical expression is derived.

In this paper, we present a unified approach that seeks to find, *for each input vector and logic gate, each possible timestep, and each possible magnitude of collected charge, what set of latch errors will be induced.* This approach is much in the spirit of SERA, with two notable exceptions; we consider the impact of different input states on the struck gate, and we attribute all FIT to the gates that initiate it, and to exact sets of latches that are upset.

3. Overview of Approach

A sequential circuit C_s is formally modeled as a tuple $\langle I, O, L, \delta, \rho, \theta \rangle$, where I is the set of input signals, O is the set of output signals, L is the set of state variables (latches) that induce the state space 2^L, $\delta : 2^I \times 2^L \rightarrow 2^L$ is the deterministic next state function of C_s, ρ is the output function, and θ describes the initial state of the circuit.

Our approach is outlined in Figure 1. There are three inputs to our tool: the sequential circuit C_s, its specification S (which is either a set of assertions or a reference model), and its workload W, where circuit C_s satisfies specification S on workload W. There are three steps of operation: *(Step 1)* Simulate W on C_s. The output is the set of state and input vectors that are generated during this simulation. This output is fed into the two subsequent steps; *(Step 2)* Estimate the FIT rate of combinational circuit nodes, denoted CFIT, and the FIT rate due to a strike in a latch, denoted LFIT. CFIT can upset one or more latches, and LFIT upsets only the struck latch (state-holding element); and *(Step 3)* Estimate the system-level FIT rate of the sequential circuit (SFIT) by performing statistical fault injection and determining the probability that a system-level specification (e.g., an assertion of system-level behavior) fails due to a strike in a combinational node or latch.

To do this, we construct a combinational circuit C_c, where $L \cup I$ is the set of inputs to C_c and L' is the set of outputs of C_c, such that for a set of assignments l and i, $l' = \delta(l, i)$. The state vector l and input vector i are obtained by recording states and inputs generated from the workload of the circuit C_s.

Let G be the set of gates in C_c. A soft error in a gate $g_i \in G$ creates a faulty circuit C_c' such that for input assignments l and

i, $g_i = \overline{f(l,i)}$ where f is the boolean function that computes the value g_i in the original circuit C_c.

To determine the probability that an exact set of upset latches (denoted E) causes a system failure, we construct a monitor circuit M from the given specification S, such that M outputs 1 iff C_s does not satisfy S. We simulate the combined circuit $C = C_s || M$ with E flipped in a randomly chosen cycle to compute the probability that a soft error in E causes M to output 1. Because the number of different multiple-latch upsets is potentially exponential in the number of latches, we simulate only sets containing a single latch, denoted l_i, and call the probability of a flip in l_i causing an assertion to fail P_{l_i}. Hence, the system-level FIT contribution of l_i, SFIT(l_i) is $FIT_{l_i} \times P_{l_i}$. We conservatively assume that all multiple-latch CFIT will lead to system failure; hence the system-level FIT contribution due to multi-latch upsets is simply the CFIT.

In the following section, we describe how we compute the FIT contribution of individual circuit elements accounting for logical, electrical, and timing masking.

4. Circuit-Level Soft Error Analysis

When a particle strikes a combinational logic gate, a soft error will only occur when the resulting glitch is captured by one or more downstream sequential elements. We use an efficient simulation methodology to determine the FIT rate of each gate, and to break down the FIT rate according to what set of 1 or more latches captures the glitch. Using g_j to denote a struck gate, and E to denote the exact set of flipped latches, $CFIT_{g_j \to E}$ describes the FIT captured in E and originating in g_j. CFIT depends on electrical, logic, and timing masking, and thus depends on the input vector applied to the circuit. Representative results are obtained by using vectors (denoted L_s) sampled from sequential simulation (Eq. 1).

An event of particular interest is that in which set E is a single latch l_i is in error. In such a case, the error can be caused by either a direct strike to l_i (denoted $LFIT_{l_i}$) or by CFIT that is captured in latch l_i and no others. A majority of all FIT falls into this category of single-latch FIT [4].

$$CFIT_{g_j \to E} = \frac{1}{|L_s|} \sum_{v \in L_s} \left(CFIT^v_{g_j \to E} \right) \tag{1}$$

$$FIT_{l_i} = LFIT_{l_i} + \sum_{\forall g_j} \left(CFIT_{g_j \to l_i} \right) \tag{2}$$

To determine what set of latches, if any, will capture a glitch originating from a strike to a combinational logic gate, the set of sensitized paths must be considered. The methodology of this paper shows that the soft error susceptibility of paths can be characterized independently, and that logical operations can subsequently be applied to these characterizations to determine the FIT caused by each gate, broken down according to the exact sets of latches capturing the error. Because the sets are disjoint, any quantity of interest can be derived by summing over the relevant gates and latch sets.

Every possible neutron strike is described by a pair q, t representing the magnitude of collected charge and the time within the clock cycle that the strike occurs, with 0 defined to be the arrival of the latching edge of the system clock. For a sensitized path π leading from combinational gate g_j to latch l_i, a boolean valued function $N^\pi_{g_j \to l_i}(q,t)$ is used to describe the outcome of each possible strike, with this function taking the value 1 to indicate the strikes that become latched in l_i, and taking the value 0 to indicate that a glitch is not latched.

Before describing how $N^\pi_{g_j \to l_i}(q,t)$ can be obtained, we describe how it is used at the core of combinational logic SER analysis. For any event of interest, the key to finding the FIT is to determine the $N(q,t)$ function; in other words, to determine exactly which charges and times of strikes can lead to that event. For the event of a collected charge in g_j being captured in latch l_i via any path, the appropriate function is the disjunction over all sensitized paths from g_j to l_i (Eq. 3). Our analysis neglects situations where two or more propagating glitches recoverge and mask each other; Zhang et al. observe that the impact of this assumption is minimal [20]. Note that the function $N(q,t)$ subsumes all masking factors. Timing masked and electrically masked strikes are those (q,t) described by $N(q,t) = 0$. Logically masked strikes have $N(q,t) = 0$ for all q, t, as no strikes to a masked node can cause an error.

For any fully specified set of latch errors, the relevant event is a minterm of the $N(q,t)$ functions of the latches, with polarity depending on whether or not the latch is included in the set. For example, to determine the FIT caused by g_j and captured in exactly latch set E, the relevant event is described by the minterm of Eq. 4. Although the number of possible latch sets is exponential in the number of latches, errors are only observed in a small subset of the possible sets.

$$N^v_{g_j \to l_i}(q,t) = \bigvee_{\forall \pi : g_j \to l_i} N^\pi_{g_j \to l_i}(q,t) \tag{3}$$

$$N^v_{g_j \to E}(q,t) = \bigwedge_{\forall l_i \in E} N^v_{g_j \to l_i}(q,t) \wedge \bigwedge_{\forall l_i \in L-E} \overline{N^v_{g_j \to l_i}(q,t)} \tag{4}$$

Based on the work of Hazucha et al. [8] (and further adopted in [22, 16]), the number of occurrences per 10^9 hours of a collected charge exceeding q, occurring at time t within the clock cycle is given by Eq. 5. For brevity, we omit a description of all parameters, which can be found in the BFIT manual [4].

$$R(q,t) = \frac{1}{t_{cycle}} \times F \times K \times A \times \exp\left(\frac{q}{Q_s}\right) \tag{5}$$

Using the function $N(q,t)$ to specify which strikes will cause a specific error, and the function $R(q,t)$ to specify the rate of occurrence of all such strikes, the FIT of a specified failure event is obtained by integrating the rate of occurrence over all strikes that cause that failure event. For the event of a collected charge in gate g_j being latched by in set E after when input vector v is applied, the FIT is given by Eq. 6. Note that a boolean-valued function $N(q,t)$ merely selects which q, t pairs should be included in the integral. The integral is evaluated numerically, using a timestep of 10ps. Values of q exceeding 150fC have probabilities of practically 0, and are not considered.

$$CFIT^v_{g_j \to E} = \int_{q=0}^{\infty} \int_{t=0}^{t_{cycle}} R(q,t) * N^v_{g_j \to E}(q,t) dt dq \tag{6}$$

4.1. Determining $N(q,t)$ for a single path

As technologies scale, the switching time of logic gates is reduced, lessening the tendency of logic to attenuate transient glitches. Studies have shown that soft error-induced transient glitches can propagate through logic in sub 100nm technologies without being significantly attenuated [13, 5]. This is in sharp contrast to older technologies, where soft error-induced glitches typically could not propagate through more than a few gates before being attenuated away [5]. Accordingly, the importance of accurately modeling glitch generation is increasing, relative to the importance of modeling path attenuation. The novel combinational logic soft error methodology of this paper demonstrates that the reduction in attenuation can greatly simplify soft error analysis of combinational logic. This methodology allows for large circuits to be analyzed, relying heaving on pre-characterization of each gate type.

4.1.1 Pre-characterizing gates for glitch generation

The two factors that determine the CFIT of a gate are the magnitudes of collected charges required to cause a flip, and the rate-of-occurrence of such charges within that gate; both of these factors depend on the input state of the gate. An NMOS diffusion will only collect electrons (creating a negative voltage glitch), and the P-type diffusion of PMOS devices will collect holes (creating a positive voltage glitch). In either case, the neutron strike is modeled as a current of the form of Eq. 7 injected into a circuit node, as proposed by Freeman [6] and adopted in subsequent work [8, 16, 22]. Technology-dependent parameter τ is the time constant of the injected current; τ is set to 20ps for both PMOS and NMOS devices in this work, as is shown to be appropriate for 90nm technology by Hazucha et al. [8]. Strikes of various magnitudes are modeled by changing the proportionality constant of the injected current. The total collected charge q of a strike is the integral of the injected current.

$$I\left(t\right) \propto \frac{1}{\tau}\sqrt{\frac{t}{\tau}}\exp\left(\frac{-t}{\tau}\right) \tag{7}$$

For a collected charge to cause an error, it must be able to charge a capacitance to create and propagate a voltage glitch. In CMOS logic, each gate output is always connected to ground or supply by a pull-up or pull-down path; the neutron-induced charge collection must be of sufficient magnitude to overcome that path in creating an erroneous voltage. The strength of this pull-up or pull-down path is a function of the gate input [12, 7], hence $N(q,t)$ is a function of the gate input. Additionally, the rate of occurrence of each magnitude of strike is a function of the type and area of the reverse-biased diffusions, which is also a function of gate input.

The various ways in which FIT depends on gate input state are illustrated using a NAND2 gate, with drive strengths and sensitized diffusions for each input state annotated (Fig. 2). Figure 4 gives the $N(q,t)$ function for each input state. Note that the 01 and 10 input combinations have similar $N(q,t)$ functions, since these two states have identical drive strength (each via a single PMOS device), and that $N(q,t)$ of the 00 input state requires a much larger charge collection to cause an error, on account of having a higher drive strength (via both PMOS devices). Now consider how the type and area of sensitive diffusion and the $N(q,t)$ function impact the FIT rate. Although the 01 and 10 state are susceptible to roughly the same set of strike magnitudes and times, the 10 state has double the FIT of the 01 state on account of having twice as much area of sensitive N-type diffusion. Also note that the 11 state contributes the least FIT, despite being susceptible to smaller charges than the 00 state. This is due to charge collections in P-type diffusions being less frequent than those in N-type diffusions, on account of PMOS devices sitting inside a well [8]; in our work, this is captured in the different collection efficiency parameter (Q_s) used for N and P devices (Eq. 5).

4.1.2 Path-Based Analysis

Under the assumption that paths do not significantly attenuate glitches, the FIT of a gate depends on its input, capacitance, and delays of sensitized paths. Consider a gate of gate type g, input state s, and capacitive load c, and a single sensitized path π. Assume that we have already performed a full characterization of another gate of same type, input state, and fanout load, but for a different path length π'. We can then construct the $N(q,t)$ function for this gate by referencing this existing $N(q,t)$ function, and adjusting the time axis. Extending this logic, it is only necessary to analyze each combination of g, s, c once. This pre-characterization is done for all cells in the library. Observed capacitances are rounded to their closest match among a set of 20 pre-characterized capacitances, representing a fanout of 1 minimum sized inverter through fanout of 20 minimum sized inverters.

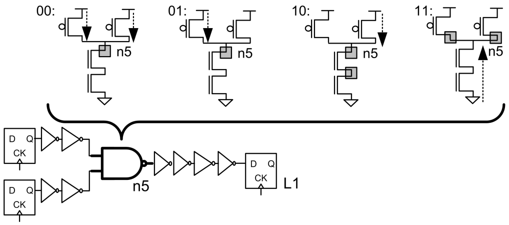
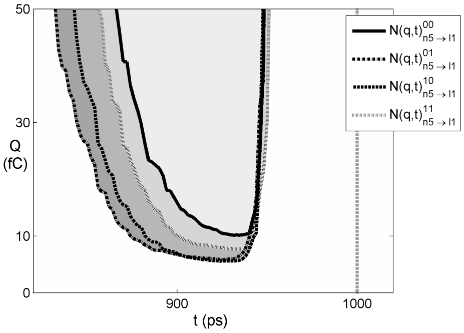


Figure 2: Drive strength and sensitive areas for NAND2



Input	A	FIT
00	2N	2.8E-5
01	2N	5.4E-5
10	4N	10E-5
11	2P	1.4E-5

Figure 3: Sensitive area and FIT

Figure 4: FIT depends on $N(q,t)$ and area/type of sensitized diffusion. Clock edge occurs at 1ns.

For each combination of $g \times c$, a characterization is performed of the $N(q,t)$ function, and of the timing arcs. To characterize $N(q,t)$, a test circuit is created connecting the desired gate to a latch through a single path For each gate input, SPICE simulation is used on this to find $N(q,t)$ at each timestep using a binary search to resolve the boundary between $N(q,t) = 1$ and $N(q,t) = 0$ to within 0.01fC; this boundary can be thought of as a time varying representation of Qcrit. Let a gate g_j have a single sensitized path π (with delay d_π) to latch l_i. Let g_j be of type g, with capacitive load c and input state s. The appropriate $N(q,t)$ function is then described by Eq. 8, where $\hat{N}$ is the pre-characterized gate.

$$N^{\pi}_{g_j \to l_i}(q,t) = \hat{N}^{s}_{g}(q, t - d_\pi) \tag{8}$$

4.1.3 Sources of Error

To demonstrate the error introduced by neglecting attenuation, the FIT rates produced by our methodology are compared SPICE simulation of path of NAND2 gates with all side inputs set to 1; the experiment is repeated with each NAND2 gate driving a fanout of 1, 2, and 3. Gates that are an odd number of gates from the output are in the logical 1 state and sensitized to an NMOS strike, while even numbered gates are in the 0 state and sensitized to a PMOS strike; all similar gates have the same input and load capacitance, and thus the same FIT rate. Table 1 demonstrates that the assumption of no attenuation does not induce significant error for the gates with single fanout, regardless of path length. With higher fanout, attenuation induces error, reflected in an overestimation of CFIT. In the worst case, we observe a 25% error. This error can be potentially reduced by using a pre-characterized lookup table for propagating the $N(q,t)$ function. Note that the 25% error is relatively minor compared to the 200-700% error (see Fig 3)

		Number of gates between strike and latch							
FO	**Method**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
1	SPICE	1.4	4.9	1.4	4.8	1.4	4.9	1.4	4.9
	CFIT	1.4	4.8	1.4	4.8	1.4	4.8	1.4	4.8
2	SPICE	1.3	5.0	1.3	5.1	1.4	5.1	1.4	4.9
	CFIT	1.4	5.0	1.4	5.0	1.4	5.0	1.4	4.8
3	SPICE	1.1	5.2	1.2	5.3	1.3	5.4	1.4	4.9
	CFIT	1.4	5.3	1.4	5.3	1.4	5.3	1.4	4.8

Table 1: Attenuation does cause some error on long paths when fanout is high. All FIT rates in table are multiples of 10^{-5}.

that would result from neglecting gate input states.

4.2. Implementation of BFIT

The Berkeley FIT Estimation tool (BFIT) implements the combinational logic soft error methodology described in the preceding subsections [4]. BFIT reads in the netlist and a pre-characterized gate library, creates and levelizes a DAG from the netlist, and outputs an executable simulation for the circuit by way of a C++ model. An important circuit modification made by BFIT is decomposing each non-inverting CMOS gate into inverting CMOS gates, ensuring a one-to-one correspondence between gates and strikeable nodes, allowing simulation to be performed at the gate level of abstraction.

Input vectors to the BFIT simulation are sampled from the sequential simulation of the same circuit. FIT rates are averaged over all vectors. Each vector is processed as described by the following three steps:

1. **Propagate vector.** A sampled state l from L_S, and random assignment i to circuit inputs I, collectively represent an input vector v to the combinational logic. The input v is propagated through the levelized combinational logic DAG. Once v is propagated, the state of each gate is $g_j = f_j(v)$, and the next state of each latch is $l'_k = \delta_k(v)$.
2. **Find sensitized delays.** Once the input vector has been fully propagated, backtracing through the levelized DAG is used to determine the set of sensitized path delays from each gate g_j to each latch l_i. Denote this list D_{g_j}. To allow for the delays to be obtained using an efficient dynamic programming on the levelized DAG, only statically sensitized paths are included in D_{g_j}.
3. **Find CFIT.** For each gate g_j in the circuit, the list of sensitized path delays and their terminating latches is known. For each sensitized path π with delay d_π, $N(q,t)$ is generated by time shifting the pre-characterized $N(q,t)$ for a gate of the same type, input, and fanout, as in Eq. 8. The $N(q,t)$ function is then obtained for g_j with respect to each latch using Eq. 3. From this, $N(q,t)$ of all possible latch sets is determined using Eq. 4, and the FIT of each set is determined using Eq. 6.

5. Results

We evaluated our approach with respect to its efficiency and usefulness in pin-pointing circuit components to harden. For the evaluation, we used the seven largest ISCAS'89 benchmarks (listed in Table 2) and a chip multiprocessor (CMP) router design [15]. For each circuit, we assume that latches are already hardened against direct strikes (LFIT=0), and show that our methodology can be used to guide efficient and flexible hardening against combinational logic soft errors.

Circuit	FIT	TIME (s)	TYPE	Fractional CFIT reduction		
				p=10%	p=50%	p=90%
s5378	1.80E-02	272	GATE (2102)	0.324	0.873	0.999
			LATCH (151)	0.520	0.807	0.935
s9234	6.82E-02	778	GATE (4569)	0.330	0.840	0.999
			LATCH (229)	0.212	0.708	0.956
s13207	1.29E-01	1381	GATE (6795)	0.282	0.709	0.995
			LATCH (594)	0.246	0.728	0.953
s15850	1.57E-01	2117	GATE (8400)	0.342	0.784	0.998
			LATCH (584)	0.231	0.666	0.954
s35932	2.25E-01	1274	GATE (14913)	0.393	0.818	0.985
			LATCH (1729)	0.271	0.746	0.955
s38417	3.84E-01	5076	GATE (20608)	0.298	0.773	0.997
			LATCH (1419)	0.155	0.615	0.956
s38584	3.01E-01	14216	GATE (16253)	0.311	0.863	0.997
			LATCH (1299)	0.192	0.582	0.931

Table 2: The reduction in FIT, as a fraction of the original FIT, that can be obtained by hardening the top p % latch or gates contributing the most error. The total number of gates and latches that contribute FIT to each circuit are shown in parentheses.

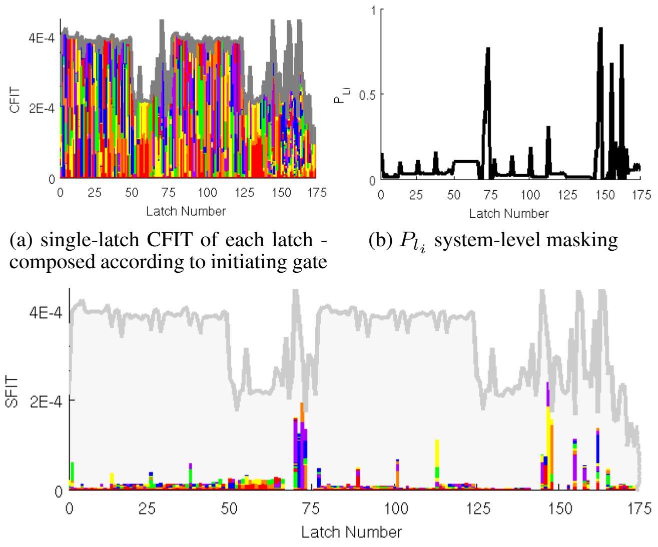
Hardening Methodology. Each combinational logic soft error is initiated at a gate, and captured at one or more downstream latches. Soft-error hardening can be applied at the initial combinational gate (e.g. through upsizing), or else at the capturing latch (e.g. through time-filtering). Amenable to both hardening approaches, our methodology determines the FIT contribution of each logic gate, and further breaks down the FIT of each gate according to the exact set of latches that will be upset. Because we specify exact (disjoint) sets of upset latches, the FIT rate of any event can be determined by summing over the appropriate gates and sets of latches. Determining even something as simple as the FIT of an entire circuit cannot be accomplished without disjoint events; approaches that give only the FIT rate of each latch will multiple-count strikes that lead to multiple latch failures. While our experiments show that single-latch FIT accounts for over 95% of all FIT, multiple-latch errors cannot be neglected, as they can be more difficult to protect against at the system level.

Given any definition of system failure and a FIT target T, the FIT breakdown gives the designer a Pareto-optimal point for gate hardening (x_{gate}, T) where the value of x indicates gates that must be protected to reduce FIT to T. It also gives an efficient point for latch hardening (x_{latch}, T) by using a greedy algorithm to always choose to harden the latch that will most reduce failure; in cases where single-latch errors dominate, the latch-hardening is Pareto-optimal. Given the two options of hardening, the designer can choose one or the other based on their own constraints.

ISCAS'89 Benchmarks. Lacking formal assertions for the ISCAS benchmarks, we assume that system failure is caused by any error captured in one or more latches ($P_{l_i} = 1$ for all latches). From Table 2, we see that for the largest ISCAS'89 benchmarks, hardening the top 50% of gates results in a FIT reduction of 71% or more. Similarly, hardening the top 50% of latches results in a FIT reduction of 58% or more.

CMP Router. The CMP router was chosen because it is representative of on-chip interconnection networks with readily available system-level specifications. The router's functionality is easy to state: it must correctly direct each of its input packets to the output port specified by the packet header within a specified number of cycles. The original CMP router is simplified to a 2-port version with 997 gates and 174 latches. A 3-to-1 workload for the two input ports is used for both state sampling and simulation with soft error injection. The rate of logic-soft-error-induced latch flips (CFIT) is determined by simulating 10,000 sampled states. The total CFIT of the router circuit is 0.0609, with 94.2% being single-latch CFIT, 3.6% being two-latch CFIT, 0.5% being 3-latch CFIT, and the remaining 1.6% coming from an assortment of larger sets.

Figure 5a shows the sources of the single-latch CFIT of each latch. Each gate is assigned a color in the figure (the colors repeat), and the single-latch CFIT of each latch is shown by stacking up the relevant portions of CFIT across all gates. Each latch has between 1.5e-4 and 4.5e-4 single-latch CFIT.



(a) single-latch CFIT of each latch - composed according to initiating gate

(b) P_{l_i} system-level masking

(c) single-latch SFIT of each latch, broken down by initiating gate.

Figure 5: Single-latch CFIT, P_{l_i}, and single-latch $SFIT$ of CMP router.

To resolve system-level masking for single-latch flips, 100 sequential simulations are performed, each with a soft error injected at a random cycle. System-level failure probability P_{l_i} of a latch l_i is then estimated to be the fraction of runs in which the monitors flag an error. Adding system-level masking (P_{l_i}) to consideration reveals an asymmetry in the failure contribution of various gates and latches. It is observed that in some latches, a majority of bit flips will lead to errors (see latches with $P_{l_i} > 0.5$ in Fig. 5b); in other latches, few (if any) bit flips will lead to errors ($P_{l_i} \approx 0$). Overall, only 6 percent of single-latch CFIT will lead to system failures. The product of single-latch CFIT and P_{l_i} contributes to the system level failure rate (SFIT); observe that the P_{l_i} line closely tracks single-latch SFIT (Fig. 5c), indicating that P_{l_i} is the dominant factor. The other contributor to SFIT is multi-latch CFIT, which we conservatively assume always leads to a failure. Because of the significant system-level masking to single-latch flips, the single-latch contribution to SFIT is comparable to that of multiple-latch FIT. SFIT of the entire circuit is 7.8e-3; with 3.2e-3 coming from single-latch flips, and the other 3.5e-3 coming from multiple-latch flips.

We demonstrate our approach to hardening the CMP router, as shown in Figure 6. If hardening gates, the Pareto-optimal gates to harden are identified. If hardening latches, the greedy selection algorithm starts out by optimally eliminating single-latch SFIT; after the single-latch SFIT is eliminated, other latches are hardened until all multiple-latch SFIT is eventually protected.

The analysis we present is efficient. Running all experiments on a 64-bit Linux workstation with 3 GHz Intel Xeon processors and 4 GB RAM, it takes no more than 240 minutes to run 10,000 vectors on any ISCAS'89 circuit. This compares favorably with SERA, which has a runtime of 593 minutes for 10,000 vectors on a 32x32 bit multiplier using a 2.8 GHz Intel Xeon with 1 GB RAM [22]. The analysis for the CMP router was also efficient: it took only 311 seconds to simulate 10,000 sampled states and to obtain the FIT contribution of each gate (broken down by exact latch set), and on average 58.2 seconds to compute P_{l_i} for each l_i using Icarus Verilog.

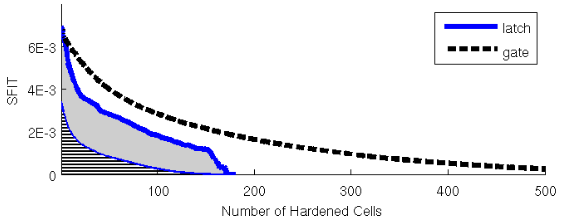


Figure 6: SFIT reduction through hardening of either latch cells or gate cells. For latch hardening, the lined section is single-latch FIT, while the gray area is multi-latch FIT.

6. Conclusions

We have presented a new methodology for performing system-level soft error analysis of arbitrary sequential circuit designs. The approach handles all kinds of error masking in combinational logic, and is shown to be efficient and accurate. Experimental results show that a system-level analysis is necessary for accurately estimating circuit vulnerability.

Acknowledgments. This research was supported in part by the Hellman Family Faculty Fund and the Gigascale Systems Research Center, one of five research centers funded under the Focus Center Research Program. The authors also thank Radu Zlatanovici for providing access to, and supporting, Cadence RTL Compiler.

References

[1] H. Asadi and M. B. Tahoori. Soft error derating computation in sequential circuits. In *ICCAD*, pages 497–501, 2006.

[2] N. Miskov-Zivanov and D. Marculescu. Modeling and Optimization for Soft-Error Reliability of Sequential Circuits. *IEEE Trans. on CAD of Integ. Circ. and Sys.*, pages 803–816, May 2008.

[3] N. Miskov-Zivanov and D. Marculescu. Circuit Reliability Analysis Using Symbolic Techniques. *IEEE Trans. on CAD of Integ. Circ. and Sys.*, pages 2638–2649, Dec. 2006.

[4] D. Holcomb et al. Berkeley FIT estimation tool (BFIT). http://www.eecs.berkeley.edu/~holcomb/BFIT.htm

[5] R. Baumann. Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Trans. Device and Materials Reliability*, 5(3):305–316, Sept. 2005.

[6] L. B. Freeman. Critical charge calculations for a bipolar SRAM array. *IBM J. Res. Dev.*, 40(1):119–129, 1996.

[7] R. Garg et al. A fast, analytical estimator for the SEU-induced pulse width in combinational designs. *DAC'08*, pages 918–923.

[8] P. Hazucha and C. Svensson. Impact of CMOS technology scaling on the atmospheric neutron soft error rate. *IEEE Trans. Nuclear Science*, 47(6):2586–2594, Dec 2000.

[9] J. A. Rivers et al. Phaser: Phased methodology for modeling the system-level effects of soft errors. *IBM Journal of Research and Development*, 52(3):293–306, May 2008.

[10] S. Krishnaswamy, et al. On the role of timing masking in reliable logic circuit design. *DAC 2008*, pages 924–929.

[11] P. Liden, et al. On latching probability of particle induced transients in combinational networks. *FTCS 1994*, pages 340–349.

[12] L. Massengill, et al. Analysis of single-event effects in combinational logic-simulation of the AM2901 bitslice processor. *IEEE Trans. Nuclear Science*, 47(6):2609–2615, Dec 2000.

[13] D. Mavis and P. Eaton. Soft error rate mitigation techniques for modern microcircuits. *Rel. Phy. Symp.*, pages 216–225, 2002.

[14] M. Omana, et al. A model for transient fault propagation in combinatorial logic. *IOLTS 2003*, pages 111–115, July 2003.

[15] L.-S. Peh. *Flow Control and Micro-Architectural Mechanisms for Extending the Performance of Interconnection Networks*. PhD thesis, Stanford University, August 2001.

[16] R. R. Rao, et al. Computing the soft error rate of a combinational logic circuit using parameterized descriptors. *IEEE Trans. on CAD of Integ. Circ. and Sys.*, 26(3):468–479, 2007.

[17] S. S. Mukherjee et al. A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor. In *MICRO 2003*, pages 29–40.

[18] S. A. Seshia, et al. Verification-guided soft error resilience. In *DATE 2007*, pages 1442–1447.

[19] P. Shivakumar, et al. Modeling the effect of technology trends on soft error rate of combinational logic, *DSN'02*, pp. 389-398.

[20] B. Zhang, et al. FASER: fast analysis of soft error susceptibility for cell-based designs. *ISQED 2006*, pages 755-760.

[21] M. Zhang, et al. Sequential element design with built-in soft error resilience. *IEEE Transactions on VLSI*, Dec. 2006.

[22] M. Zhang and N. R. Shanbhag. Soft-error-rate-analysis (SERA) methodology. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 25(10):2140–2155, 2006.

[23] Q. Zhou and K. Mohanram. Gate sizing to radiation harden combinational logic. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 25(1):155–166, 2006.

[24] J. Ziegler. Terrestrial cosmic rays. *IBM J. Res. Develop*, 40(1):pp19–39, January 1996.