Abstract—High energy particles from cosmic rays or packaging materials can generate a glitch or a current transient (single event transient or SET) in a logic circuit. This SET can eventually get captured in a register resulting in a flip of the register content, which is known as soft error or single-event upset (SEU). A soft error is typically modeled as a probabilistic single bit-flip model. In developing such abstract fault models, an important issue to consider is the likelihood of multiple bit errors caused by particle strikes. In an earlier related work, we performed a characterization study of the impact of an SET on a logic circuit to quantify the extent to which this model is accurate. We found that, a substantial fraction of SEU outcomes had multiple register flips. In this paper, we perform a deeper analysis to understand the individual contributions of, ‘strike on a register’ and ‘strike on a logic gate’, on multiple flips. We use post-layout circuit simulations and Monte Carlo sampling scheme to study the impact of an SET. We perform our simulations on ISCAS’85, ISCAS’89 and ITC’99 benchmarks in 180nm and 65nm technologies. We find that, amongst the erroneous outcomes, the probability of multiple bit-flips for ‘gate-strike’ cases was substantial and went up to 50%, where as those for ‘register-strike’ cases was just about 2%. This implies that, if we were to do hardened flip-flop designs to eliminate the flips due to register strikes, in such designs, out of the remaining flips which are going to be due to ‘gate strikes’, a large fraction will be multiple flips. Hardening the flip-flops is not going to help in this scenario. Thus, gate strikes are problematic and complex error correction schemes at the circuit level are needed to eliminate the multiple errors due to ‘gate strikes’.

Index Terms—soft error, gate strike, multiple bit flips, fault model, logic circuits

I. INTRODUCTION

Soft errors have a significant impact on the circuit reliability at nanoscale technologies. High energy particles from either cosmic rays or packaging materials are known to be the major contributors towards soft errors. When such particles strike a semiconductor substrate, they generate charge which in turn results in a glitch or a transient current in a circuit. Such a glitch is called the single event transient (SET). The SET can occur on a register or on a logic gate and can eventually get captured in a register, altering the stored bit (0 → 1 or 1 → 0). Such an error is known as a single event upset (SEU) or soft errors. The cosmic rays interact with earth’s atmosphere creating secondary particles, mainly neutrons, protons, muons and pions, as they penetrate down to the sea level. The energy or flux of these particles increases at a rate of 5x per 5000 feet, reaching a maximum intensity at about 12-15 km from sea level [1], [2]. Thus, the errors caused due to these particles were of specific interest in satellite, aircraft and space applications. For example, in-flight measurements reveal an upset rate of approximately 5x10⁻³ upsets per hour per memory device [3].

Several studies were then conducted by IBM and also summarized by Boeing Defense and Space group, which reveal that a significant amount of neutron flux is found even at sea level [1], [4]. The average low energy neutron flux (around 10MeV) at sea level is reported to be nearly 100000 neutrons/sq.cm per year [1]. They report that thousands of upsets happen every year at the ground level, which are mainly recorded by computer systems having error detection and correction logs. For instance, the average error rate in memories observed in several computers were reported to be nearly 1.5e-12 upsets/bit-hr and most of the upsets are reported to be due to atmospheric neutrons [4]. Thus, SEUs caused due to neutrons gained significant importance. Most early studies focused on SEUs only in memories. SEUs in logic circuits were not considered because such circuits exhibit inherent masking phenomena, which prevented the SETs from getting captured in a register/flip-flop. However, as technology scales, the impact of masking phenomena tends to reduce [5], [6], [7], [8] and it is important to study SEUs in logic circuits.

At the architectural level, soft errors are commonly modeled by a probabilistic single bit-flip fault model [9], [10], [11], [12]. In developing such abstract fault models, an important issue to consider is the likelihood of multiple bit errors caused by particle strikes. This likelihood has been studied to a great extent in memories, but has not been understood to the same extent in logic circuits. This model has been challenged in [13], [14], [15], [16], which report that multiple bit flips do occur in logic circuits. However, the state-of-art fault model continues to be a single bit-flip, single cycle model for soft error in logic circuits. Reliability estimates (such as mean time to failure/MTTF) and reliability enhancement techniques (such as error correction codes etc) are also based on the assumption that a single bit flip occurs due to a particle strike. However, in reality, if multiple errors occur, these MTTF estimates are likely to be optimistic and the error correction methods are likely to be insufficient. Most of the existing techniques that estimate soft error rate (SER) use approximate modeling techniques to arrive at these conclusions which reduces the accuracy of their results. Therefore it is essential that our estimation technique be as accurate as possible, in order to increase the confidence in the conclusions on the multiple bit flip probability in logic circuits. So, we propose to perform a detailed characterization of the impact of an SET using post-
layout circuit (SPICE) simulations in order to get accurate bit-flip statistics. Our goal is to evaluate the accuracy of the single bit-flip, single cycle fault model. In other words, we attempt to quantify the likelihood that an SET can cause multiple bit errors in logic circuits consisting of combinational gates and flip-flops.

We performed a circuit level study to this extent, earlier [17]. We had established that the impact of an SET in a circuit can be understood as a two-cycle phenomenon, that is, the SEU outcomes need to be observed across two clock cycles (the cycle which had the SET injected and the following clock cycle) in order to accurately capture the phenomenon. This leads us to the fact that there are several possible SEU outcomes. We estimated the multiple flip probability given that there was at least one flip in that particular clock cycle. We found that a substantial fraction (24%) of such outcomes were multiple register flips. In this paper, our focus is still on evaluating the multiple flip probability. However, we revisit the previous analysis and estimate the multiple flip probability given that there is at least one erroneous outcome in either of the two clock cycles. In other words, given that an SET propagates and causes an error in a flip-flop, what is the probability that it can flip multiple flip-flops? We also delve a bit more deeper and evaluate this probability separately for strikes on logic gates and strikes on registers to understand their individual contributions. We run our experiments on the ISCAS’85, ISCAS’89 and ITC’99 benchmark circuits in 180nm and 65nm technologies. We evaluate the bit-flip statistics by comparing the SET-induced circuit simulation with a fault-free register-transfer-level (RTL) reference simulation. In our simulations, we assume that an SET affects a single transistor [18], [19].

We find that, overall, up to 8% of the erroneous outcomes result in multiple bit-flips. Although this probability is low, it can have a significant impact on error-detection or correction schemes. It means that a single bit error correction scheme can go wrong in as much as 8% of the cases, and these errors will go down as silent undetected errors. The probability of multiple errors also increases as technology is scaled (based on 180nm and 65nm data). So, the traditional single bit-flip soft error model certainly needs to be reexamined to get more realistic circuit reliability estimates. A key observation is that, amongst the erroneous outcomes, the probability of multiple bit-flips for ‘gate-strike’ cases was substantial and went up to 50%, that is, these errors are caused due to the propagation of the SET from the logic gate to the flip-flop. On the other hand, out of the erroneous outcomes, the likelihood of multiple flips for ‘register-strike’ cases was just about 2%. This implies that, if we were to do hardened flip-flop designs to eliminate the flips due to register strikes, in such designs, out of the remaining flips which are going to be due to ‘gate strikes’, a large fraction will be multiple flips. So, although the traditional circuit designs with hardened flip-flops will solve one problem, they will uncover a different problem. Since, strikes on gates can be problematic, different types of error correction schemes such as delay-capture flip-flop designs or uneven paths with padded delays might be needed at the circuit level to eliminate such multiple errors due to ‘gate strikes’.

We organize the rest of the paper as follows. In Section II, we provide a brief introduction to the two-cycle phenomenon and describe the possible SEU outcomes in a logic circuit. We describe the experimental setup in Section III. We present our simulation results on ISCAS’85, ISCAS’89 and ITC’99 benchmarks in Section IV. In Section V, we summarize our paper.

II. A BRIEF INTRODUCTION TO TWO-CYCLE PHENOMENON AND THE POSSIBLE SEU OUTCOMES

In a logic circuit, a high-energy particle can strike a logic gate or a register, resulting in an SET. We call this the ‘gate-strike’ and ‘register-strike’ respectively. We model the SET as a current injection at the drain of a transistor in a particular clock cycle ‘k’. This SET can eventually propagate and get captured in a register or a flip-flop in the same clock cycle ‘k’ or in the subsequent clock cycle ‘k+1’, depending on the time instant at which the SET occurs in clock cycle ‘k’. If the SET flips a register content early in the clock cycle ‘k’, it will have enough time to propagate and flip some register in clock cycle ‘k+1’. However, if the SET flips a register later in the clock cycle ‘k’, the error will not have enough time to propagate and flip some other register in the subsequent clock cycle ‘k+1’. Thus, the impact of an SET, in reality, needs to be viewed across two clock cycles to accurately capture the phenomenon. We call this the ‘two-cycle phenomenon’. This time dependence of the strike is currently missing in the single bit flip, single cycle fault model. This phenomenon is described in more detail in our earlier work [17]. We can thus come up with a systematic notation for the possible SEU outcomes across two clock cycles. This is shown in Figure 1. In the figure, ‘N’ stands for no-flip, ‘F’ stands for flip and ‘Fm’ stands for multiple flips.

![Figure 1. A systematic notation for the possible error outcomes caused due to an SET in a logic circuit](image_url)
register flips. The ones caused directly as a result of the SET are denoted by ‘NF\textsubscript{m}’, ‘F\textsubscript{m}F’ and ‘F\textsubscript{m}N’. The multiple flip caused due to the propagation of the previous flip is denoted by ‘F\textsubscript{m}m’, which will be anyway captured by modeling the first flip using the traditional single bit-flip model. So, the multiple flips in this case are not a direct result of the SET and we do not consider this in our experiments.

III. SIMULATION SETUP

We model the SET as a current injection at the drain of a transistor [18], [19]. We use a fixed glitch in our experiments and we assume that the probability that an SET affects a drain is proportional to its area. We use the scaling trends presented in [20] to arrive at the glitch height and width in our experiments.

We perform our simulations on the ISCAS’85, ISCAS’89 and ITC’99 benchmark circuits in 180nm and 65nm technologies. We add flip-flops to the inputs and outputs of the combinational circuits of the ISCAS’85 benchmarks. Each circuit that we perform our experiment on, is described as a Verilog or VHDL netlist (RTL). We first perform a Verilog/VHDL simulation with a representative test-bench and store the input/output values of all the flip-flops (registers). The circuit is then implemented to layout using synthesis and placement and route tools. The post-layout circuit netlist is then extracted. From this netlist, we generate ‘sample circuit simulation decks’ by picking a random clock cycle (t) for simulation, selecting a random drain (d) to inject the SET and selecting a random time instant (k) in the clock cycle to inject the SET. The sample circuit is simulated and the register outputs are recorded at two clock instants: ‘k’ and ‘k+1’. The register values from the SET injected circuit simulation are compared with the corresponding reference values from the fault-free RTL simulation. Differences between the sampled values in the circuit simulation and the reference trace are recorded as bit-flips.

We generate several sample simulation decks with different (d,k,t) values and use the Ngspice or Hspice circuit simulators [21] to perform our simulations. The average number of sample circuit simulations run for each circuit was about 4000. We ran these simulations in parallel using GNU Parallel [22] on a high performance computing cluster in CDAC, Pune. The time taken to run these many simulations for each circuit was on an average about 1 to 2 hours. We then classify these flips in to one of the SEU outcomes described in Figure 1. The experimental setup is shown in Figure 2. This entire process is automated using a set of python and perl scripts. We continue generating circuit samples until the standard error is reduced to less than 10% of the value of the estimate. The probabilities obtained from this Monte Carlo sampling experiment fall within the 95% confidence interval. A detailed description of the setup can be found in our earlier work [17].

IV. RESULTS

We perform our experiments on ISCAS’85 (c432 etc), ISCAS’89 (s344 etc) and ITC’99 (b01 etc) benchmark circuits in 180nm and 65nm technologies. Flip-flops (registers) are added to the inputs and outputs of the combinational circuits of the ISCAS’85 benchmarks. Clock frequency for each circuit is set to the maximum operable frequency of the post-layout netlist, which is determined by post-layout timing analysis. We calculate the probability of a ‘gate-strike’ and ‘register-strike’ depending on whether the SET was injected on a gate or a register. Further, we classify the bit-flips into one of the SEU outcomes described in Figure 1. In this paper, we focus on the multiple bit flip probability independently due to ‘gate-strike’ and ‘register strike’ scenarios.

A. Probability of multiple register flips given an erroneous outcome

Given that an error occurred due to an SET (that is, one of these cases occurred- NF\textsuperscript{m}, F\textsuperscript{m}F or F\textsuperscript{m}N), what is the likelihood that the first error event caused by the SET affects multiple registers? There are two possibilities:

- The first error event occurs at cycle k + 1 and consists of multiple registers being in error, that is, \( P_m = P(NF_m|\text{atleast one flip}) \).
- The first error event occurs at cycle k and consists of multiple registers being in error, that is, \( P(F_m|N|\text{atleast one flip}) \).

We do not count the \( P(FF_m|\text{atleast one flip}) \), because the multiple flips in cycle k + 1 are caused due to the propagation of the first flip, which will be anyway captured by modeling the first flip using the traditional single bit-flip model. So, the multiple flips in this case are not a direct result of the SET.

Amongst the two cases mentioned above, we did not come across the scenario of \( P(F_m|N|\text{atleast one flip}) \) or \( P(F_m|F|\text{atleast one flip}) \) in our experiments, hence we only report the first probability: \( P_m = P(NF_m|\text{atleast one flip}) \).

This is calculated as follows for gate-strike and register-strike (denoted by reg-strike in the equation) cases:

\[
\begin{align*}
P_m &= P(NF_m|\text{atleast one flip}) \\
&= P(NF_m) - P(NF_m|F_m) \\
&= \frac{P(NF_m)}{P(F_m)} \\
&= \frac{1 - \frac{P(F_m|F_m|\text{atleast one flip})}{P(F_m)}}{P(F_m)} \\
&= \frac{1 - \frac{P(F_m|F_m|\text{atleast one flip})}{P(F_m)}}{P(F_m)}.
\end{align*}
\]
\[
P_m = \frac{No. \ of \ (NF_m)_{\text{gate-strike}} + No. \ of \ (NF_m)_{\text{reg-strike}}}{(No. \ of \ cases \ with \ at \ least \ one \ flip)_{\text{gate+reg-strike}}}
\]

In other words, we calculate

\[
P_m = \frac{No. \ of \ (NF_m)_{\text{gate-strike}} + No. \ of \ (NF_m)_{\text{reg-strike}}}{(No. \ of \ NF, \ FN \ or \ FF)_{\text{gate+reg-strike}}}
\]

For instance, if we ran 5000 simulations, out of which there were 1000 cases in which at least one error occurred (this can include NF, FN or FF across gate and register strikes), out of which, say 100 cases had multiple errors \((NF_m)\), we calculate \(P_m\) as 100/1000. This probability is shown in Table II for the benchmark circuits, across both register and gate strikes.

| Example | \(P_m\) - gate and register strike | \(P_m\) - gate and register strike |
|---------|----------------------------------|----------------------------------|
| c125    | 0.006                            | 0.035                            |
| c409    | 0.003                            | 0.016                            |
| c880    | -                                | 0.001                            |
| c1355   | 0.003                            | 0.007                            |
| c1908   | 0.003                            | 0.037                            |
| b01     | -                                | 0.02                             |
| b03     | 0.005                            | 0.007                            |
| b04     | -                                | 0.03                             |
| b06     | 0.042                            | 0.05                             |
| b09     | 0.005                            | 0.044                            |
| b10     | 0.007                            | 0.06                             |
| b11     | -                                | 0.036                            |
| b13     | -                                | 0.038                            |
| s344    | -                                | 0.02                             |
| s820    | -                                | 0.081                            |
| s1190   | -                                | 0.014                            |
| s1238   | -                                | 0.022                            |
| s1423   | -                                | 0.023                            |
| 3:8 decoder | 0.006                         | 0.014                            |
| 8-bit LFSR | 0.007                          | 0.018                            |

From Table II, we see that the maximum probability of multiple flips \(P_m\) that was observed in the circuits we simulated, across both register and gate strikes was 8%. This probability is considerably higher in most of the 65nm implementations as compared to the 180nm implementations. For a particular circuit, it is likely to depend on various factors such as the structure of the circuit itself, the presence of balanced paths, logic depth, the kind of logic gates used and the input combinations. Although 8% seems low, it can still have a significant impact on error detection and correction schemes. This means that, a single error correction scheme can go wrong 8% of the times and the errors can go down as silent undetected errors.

B. The contribution of gate strikes and register strikes to multiple register flips

In this section, we analyze our results further to understand the role of gate-strikes and register-strikes independently on the multiple flip probability. In other words, we address the following question. Given that an error occurred due to a strike on a logic gate, what is the probability that it was a multiple flip? This is denoted by \(P_{GM}\). Similarly, we calculate the multiple flip probability \(P_{RM}\) given that an error was caused due to a register strike. We calculate \(P_{GM}\) and \(P_{RM}\) respectively as follows:

\[
P_{GM} = \frac{\text{Number of } (NF_m)_{\text{gate-strike}}}{\text{Number of } NF, FN \ or \ FF}_{\text{gate-strike}}
\]

\[
P_{RM} = \frac{\text{Number of } (NF_m)_{\text{register-strike}}}{\text{Number of } NF, FN \ or \ FF}_{\text{register-strike}}
\]

Here, we should note that, the denominator in both the fractions is the total number of cases which had erroneous outcomes (flips). For the gate strike scenario, a major contributor of these erroneous outcomes is the ‘NF’ scenario, whereas for the register strike case, errors are mainly contributed by the ‘FN’ and ‘FF’ scenarios as already observed in [17]. This is because, when an SET occurs at a register, it is more likely to cause a flip in the same clock cycle (clock cycle ‘k’). So, \(P_{FN}\) and \(P_{FF}\) are dominant as compared to \(P_{NF}\) for register strike scenarios. On the other hand, when an SET occurs on a logic gate in clock cycle ‘k’, the register flip is most likely to occur due to the propagation of the SET and hence, the flip is likely to occur in the subsequent clock cycle ‘k+1’. So, \(P_{FN}\) and \(P_{FF}\) are extremely small as compared to \(P_{NF}\) for gate-strike scenarios. This difference is seen to have a significant impact on the probabilities \(P_{GM}\) and \(P_{RM}\). We present these probabilities in Table II. We find that \(P_{GM}\) is significantly greater than \(P_{RM}\) and its value is found to be a maximum of 50%. This is mainly because of two reasons:

1) When an SET occurs on a register,

a) The probability of a flip or an error occurring is high (high probabilities of an FF or an FN) as compared to the SET occurring on a gate [17]. The flip probability for register-strike cases is found to be nearly 10x greater than that for the gate-strike cases [17]. This increases the denominator in equation 2.5 and results in a small value for \(P_{RM}\).

b) The probability of an NF scenario is very less, as compared to the FN and FF scenarios for a register strike as already observed in [17]. Hence NF\(_m\) scenarios are also less leading to a small value in the numerator in equation 2.5. This again leads to low value of \(P_{RM}\).

2) When an SET occurs on a logic gate,

a) The probability of a flip is low (low probabilities of an FF or an FN) as compared to the SET occurring on a register as already observed in [17]. This decreases the denominator in equation 2.4.
Table II: Probability of multiple bit flips given that at least one error occurred. Gate strikes and register strikes are presented separately. (‘-’ indicates no multiple error events were observed)

| Example   | \(P_{GM} - 'gate strike'\) 180nm | \(P_{GM} - 'gate strike'\) 65nm | \(P_{RM} - 'register strike'\) 180nm | \(P_{RM} - 'register strike'\) 65nm |
|-----------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| 452       | 0.4                             | 0.5                             | -                               | -                               |
| c499      | 0.11                            | 0.2                             | -                               | -                               |
| 8806      | -                               | -                               | -                               | -                               |
| c1355     | 0.3                             | 0.2                             | -                               | -                               |
| c9016     | 0.066                           | 0.25                            | -                               | 0.008                           |
| 801       | -                               | 0.035                           | -                               | 0.016                           |
| 104       | 0.12                            | 0.042                           | -                               | 0.004                           |
| 904       | -                               | 0.172                           | -                               | -                               |
| 960       | 0.288                           | 0.35                            | -                               | 0.003                           |
| 909       | 0.08                            | 0.36                            | -                               | 0.003                           |
| 100       | 0.058                           | 0.4                             | -                               | -                               |
| 112       | -                               | 0.12                            | -                               | -                               |
| 123       | -                               | 0.34                            | -                               | -                               |
| 344       | -                               | 0.07                            | -                               | 0.012                           |
| 820       | -                               | 0.12                            | -                               | -                               |
| 1196      | -                               | 0.045                           | -                               | -                               |
| 1238      | -                               | 0.08                            | -                               | -                               |
| 1242      | -                               | 0.27                            | -                               | 0.003                           |
| 3:8 decoder | 0.107                      | 0.27                            | 0.001                            | -                               |
| 8-bit LFSR | 0.073                       | 0.17                            | -                               | -                               |

b) The major contributor of flips for gate strikes is \(NF\) [17]. Hence \(NF_{gm}\) scenarios are also high, as compared to the register strike cases, leading to a large value in the numerator in equation 2.4. These lead to a high value of \(P_{GM}\).

Overall, the summary from this section is described below:
- The probability that a flip or an error occurs is high in the register-strike cases. Robust flip-flop designs can prevent a flip due to a strike on a register and hence can reduce the overall flip probability.
- Given that there is an error, the probability of multiple errors is less (about 2%) in register strike cases, while they are significant in the gate-strike cases (up to 50%). Thus, in a circuit with robust flip-flops, we can eliminate the flips due to ‘register strike’ cases, but out of the remaining flips which are going to be due to ‘gate strikes’, multiple flips are extremely likely. This is depicted in Figure 3. Thus, gate strikes are problematic. Hardening the flip-flops is not going to help in this scenario. Alternate methods such as delay capture flip-flop designs or uneven path delays might be needed at the circuit level to eliminate such multiple errors due to ‘gate strikes’. The broad conclusion is that, although there is a need to re-examine the traditional single-bit-flip soft error model, there is a need to focus on circuit techniques to eliminate multiple bit errors due to gate strikes.

V. Conclusions

In this paper, we have extended our previous work, in which we performed a detailed characterization of the impact of an SET on a logic circuit. Our focus in this paper is on quantifying the extent to which the single bit-flip fault model for soft errors is accurate, by understanding the individual contributions of strikes on logic gates and strikes on registers.

We find that, amongst the erroneous outcomes, the probability of multiple bit-flips for ‘gate-strike’ cases was substantial and went up to 50% whereas those for ‘register-strike’ cases was just about 2%. This implies that, if we were to do hardened flip-flop designs to eliminate the flips due to register strikes, in such designs, out of the remaining flips which are going to be due to ‘gate strikes’, a large fraction will be multiple flips. So, although the traditional circuit designs with hardened flip-flops will solve one problem, they will uncover a different problem. Since, strikes on gates can be problematic, different approaches such as delay-capture flip-flop designs or uneven paths with padded delays might be needed at the circuit level to eliminate such multiple errors due to ‘gate strikes’. The broad conclusion is that, although there is a need to re-examine the traditional single-bit-flip soft error model, there is a need to focus on circuit techniques to eliminate multiple bit errors due to gate strikes.

REFERENCES

[1] J. Ziegler, “Terrestrial cosmic ray intensities,” IBM Journal of Research and Development, vol. 42, pp. 117–140, Jan 1998.
[2] E. Normand and T. J. Baker, “Altitude and latitude variations in avionics seu and atmospheric neutron flux,” IEEE Transactions on Nuclear Science, vol. 40, pp. 1484–1490, Dec 1993.
[3] K. Johansson, P. Dyreklev, B. Granbom, M. C. Calvet, S. Fourtine, and O. Feuillatre, “In-flight and ground testing of single event upset sensitivity in static rams,” in Radiation and Its Effects on Components and Systems, 1997. RADEC 97. Fourth European Conference on, pp. 584–588, Sep 1997.
[4] E. Normand, “Single event upset at ground level,” Nuclear Science, IEEE Transactions on, vol. 43, no. 6, pp. 2742–2750, 1996.
[5] R. Ramanarayan, V. Degalalahal, R. Krishnan, J. Kim, V. Narayanan, Y. Xie, M. Irwin, and K. Unlu. “Modeling soft errors at the device and logic levels for combinational circuits,” Dependable and Secure Computing, IEEE Transactions on, vol. 6, no. 3, pp. 202–216, 2009.
[6] M. Fazeli, S. Ahmadian, S. Miremadi, H. Asadi, and M. Tahoori, “Soft error rate estimation of digital circuits in the presence of multiple event transients (mets),” in Design, Automation Test in Europe Conference Exhibition (DATE), 2011, pp. 1–6, 2011.
[7] L. Hayes, I. Polian, and B. Becker, “An analysis framework for transient-error tolerance,” in VLSI Test Symposium, 2007. 25th IEEE, pp. 249–255, 2007.
[8] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, “Modeling the effect of technology trends on the soft error rate of combinational logic,” in Dependable Systems and Networks, 2002. DSN 2002. Proceedings, International Conference on, pp. 389–398, 2002.
[9] S. Mukherjee, J. Emer, and S. Reinhardt, “The soft error problem: an architectural perspective,” in *High-Performance Computer Architecture, 2005. HPCA-11. 11th International Symposium on*, pp. 243–247, 2005.

[10] N. Wang, J. Quek, T. Rafacz, and S. Patel, “Characterizing the effects of transient faults on a high-performance processor pipeline,” in *Dependable Systems and Networks, 2004 International Conference on*, pp. 61–70, 2004.

[11] N. J. Wang, A. Mahesri, and S. J. Patel, “Examining ace analysis reliability estimates using fault-injection,” *SIGARCH Comput. Archit. News*, vol. 35, pp. 460–469, June 2007.

[12] N. Wang and S. Patel, “Restore: Symptom-based soft error detection in microprocessors,” *Dependable and Secure Computing, IEEE Transactions on*, vol. 3, pp. 188–201, July 2006.

[13] L. Chen, M. Ebrahimi, and M. Tahoori, “Cep: Correlated error propagation for hierarchical soft error analysis,” *Journal of Electronic Testing, vol. 29, no. 2*, pp. 143–158, 2013.

[14] J. Blome, S. Mahlke, D. Bradley, and K. Flautner, “A microarchitectural analysis of soft error propagation in a production-level embedded microprocessor,” in *Proceedings of the First Workshop on Architecture Reliability*, 2005.

[15] H. Cha, E. Rudnick, J. Patel, R. Iyer, and G. Choi, “A gate-level simulation environment for alpha-particle-induced transient faults,” *Computers, IEEE Transactions on*, vol. 45, no. 11, pp. 1248–1256, 1996.

[16] G. Saggese, N. Wang, Z. Kalbarczyk, S. Patel, and R. Iyer, “An experimental study of soft errors in microprocessors,” *Micro, IEEE*, vol. 25, no. 6, pp. 30–39, 2005.

[17] N. P. Rao and M. P. Desai, “A detailed characterization of errors in logic circuits due to single-event transients,” in *Digital System Design (DSD), 2015 Euromicro Conference on*, pp. 714–721, Aug 2015.

[18] Q. Zhou and K. Mohanram, “Gate sizing to radiation harden combinational logic,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 155–166, 2006.

[19] E. Neto, I. Ribeiro, M. Vieira, G. Wirth, and F. Kastensmidt, “Using bulk built-in current sensors to detect soft errors,” *Micro, IEEE*, vol. 26, no. 5, pp. 10–18, 2006.

[20] P. Dodd, M. Shaneyfelt, J. Felix, and J. Schwank, “Production and propagation of single-event transients in high-speed digital logic ics,” *Nuclear Science, IEEE Transactions on*, vol. 51, pp. 3278–3284, Dec 2004.

[21] “Ngspice.” http://ngspice.sourceforge.net/, 2013.

[22] O. Tange, “Gnu parallel - the command-line power tool,” *;login: The USENIX Magazine*, vol. 36, pp. 42–47, Feb 2011. http://www.gnu.org/s/parallel.