Statistical Fault Injection: Quantified Error and Confidence

R. Leveugle, A. Calvez, P. Maistri, P. Vanhauwaert
TIMA Laboratory (Grenoble INP, UJF, CNRS)
46 Avenue Félix Viallet - 38031 Grenoble Cedex - France
Regis.Leveugle@imag.fr

Abstract— **Fault injection has become a very classical method to determine the dependability of an integrated system with respect to soft errors. Due to the huge number of possible error configurations in complex circuits, a random selection of a subset of potential errors is usual in practical experiments. The main limitation of such a selection is the confidence in the outcomes that is never quantified in the articles. This paper proposes an approach to quantify both the error on the presented results and the confidence on the presented interval. The computation of the required number of faults to inject in order to achieve a given confidence and error interval is also discussed. Experimental results are shown and fully support the presented approach.**

Keywords-dependability analysis, statistical fault injection

I. INTRODUCTION

Dependability analysis is a growing concern for integrated circuit designers and manufacturers due to the increasing sensitivity of the circuits manufactured with aggressive technologies. Particle strikes or electromagnetic interferences can induce spurious currents in the circuit core, resulting in erroneous logical behaviors and potentially catastrophic application failures [1]. The probability of such events has become non negligible even at sea level due to alpha particles and atmospheric neutrons and protons It is thus an increasing concern in most applications. In addition to such natural perturbations, deliberate fault-based attacks, using for example a laser, can be mounted to hack critical data such as secret cryptographic keys stored in circuits designed for security applications [2]. No matter the physical origin of the fault, an application failure results either from an erroneous value directly induced on one circuit output, or from an erroneous sequential behavior due to one or several incorrect bits in some internal register(s). Such internal errors, occurring without damaging the circuit, are called "soft errors".

The robustness of a given circuit with respect to perturbations can be qualified by various means after manufacturing (particle accelerators, alpha particle sources, laser, etc.). However, it is mandatory to analyze the possible consequences of soft errors during the design phase to avoid discovering unacceptable behaviors only during qualification.

Fault injection has become a classical approach for design-time dependability analysis. A fault injection campaign consists in comparing the reference behavior of the circuit for a given workload (i.e., the correct behavior validated by the designer) with the behavior obtained in presence of each fault in a predetermined set. Many approaches have been developed and the comparison can be based on simulations [3, 4, 5, 6, 7] or emulation [8, 9, 10]. Even with fault pruning techniques [7] and the acceleration obtained using emulation hardware, the main limitation of fault injection is the impossibility to completely analyze large circuits running complex workloads. As a matter of fact, it is often impossible to inject in a reasonable time all possible errors in all locations and at each clock cycle. In consequence, most of the results published in the literature are based on Statistical Fault Injection (SFI). During a SFI campaign, only a subset of the possible errors is injected. This subset is selected randomly with respect to the injection target and with respect to the injection cycle. This process allows a designer to tune the number of experiments according to the time available for the evaluation. Most often, the authors report that they have injected "a large number of faults" that gives "a good confidence" in the results. Unfortunately, neither this confidence nor the error on the results is quantified.

Up to our knowledge, this paper is the first attempt to propose a rigorous evaluation of the margin of error and confidence interval for SFI campaigns, without assumptions about the expected results. Also, the presented equations allow a designer to compute the required number of experiments to achieve given margin of error and confidence level. The focus is on synchronous digital circuits. Experimental results on a medium-size circuit and with a complex error model are shown, supporting the proposed approach.

The statistical background is summarized in section 2 with a sensitivity analysis of the various parameters. The coprocessor used as case study is presented in section 3. The experimental results are discussed in section 4. Some comments on previous work are finally summarized in section 5.

II. STATISTICAL BACKGROUND

Sampling is used in many domains. We propose to apply to fault injection the mathematical framework used in other fields where sampling is necessary, for example surveys [11, 12]. In this section, we will summarize the hypotheses and the formula that can be used to compute either the sample size (i.e. the number of faults to inject) or the confidence and error interval on the results.

978-3-9810801-5-5/DATE09 © 2009 EDAA

A. Hypotheses

It is assumed in the computations that the characteristics of the population (in our case, all the possible soft errors at any clock cycle) follow a normal distribution.

Each individual (i.e. a given error configuration at a given cycle) in the initial population must have the same probability to be selected in the sample. So a uniform distribution must be used during the random sampling.

In fault injection, the initial population is always finite, although the total number of individuals N can be huge. N depends on the circuit (number of memory elements that can be modified by the perturbations), on the error model (expected multiplicity and distribution of the erroneous bits for a soft error at a given time), and on the application (number of cycles of the workload).

In practice, the population can be considered as infinitely large if the sampling fraction is less than 5%. However, in what follows, we will use the equations related to finite populations (results would be the same in most cases assuming an infinite population, with simplified equations, but this is a conservative and more general approach).

In the initial population, sampling is done without replacement. Otherwise, the population can be considered as infinite but of course bias can be introduced if the same individual is chosen several times in the sample.

Finally, the advantage in the case of fault injections (as opposed to polls) is that the margin of error must only account for random sampling error. There is no systematic error induced by non-response, lies, and so on …

B. Number of Faults to Inject

With the previous hypotheses, the sample size n, or number of faults to (randomly) select for injection, can be computed with (1) with respect to:

- the initial population size N.
- the estimated proportion p of individuals in the population having a given characteristic (e.g. the estimated probability of faults resulting in a failure). This parameter defines the standard error.
- the margin of error e. This is the error on the result Peval obtained during the campaign using the sample. The exact probability that individuals have the desired characteristic should be in the interval [Peval – e ; Peval + e], or [Peval – e*100 ; Peval + e*100] if Peval is given as a percentage.
- the cut-off point (or critical value) t corresponding to the confidence level. This level is the probability that the exact value is actually within the error interval. A 90%, 95% or 99% confidence level is usually chosen (typically 95%). The cut-off point is computed with respect to the Normal distribution (quantile table).

$$n = \frac{N}{1 + e^2 \times \frac{N-1}{t^2 \times p \times (1-p)}} \quad (1)$$

For a given sample size n and a given confidence level, the margin of error can be conversely computed with (2).

$$e = t \times \sqrt{\frac{p \times (1-p)}{n} \times \frac{N-n}{N-1}} \quad (2)$$

C. Sensitivity Analysis: p Parameter

The parameter p basically corresponds to an estimate of the true value being searched (e.g. percentage of errors resulting in a failure). Since this value is a priori unknown (but between 0 and 1), a conservative approach is to use the value that will maximize the sample size. In other words, the sample size will be chosen so that it will be sufficient to ensure the expected margin of error with the expected confidence level, no matter the actual value of the proportion. It has been demonstrated that this is achieved for p=0.5; it is therefore sufficient to use this value in all cases. If the expected proportion is very small or very large, refining this estimate would lead to a reduced sample size for a given margin of error but our goal is to avoid any a priori assumption on the results.

D. Sensitivity Analysis: N Parameter

When N is large, its increase has little influence on the sample size for a given margin of error and a given confidence. This is illustrated by the curve in Figure 1. Such an asymptotic behavior is very interesting since it means that the number of fault injection experiments can remain reasonable when the error multiplicity (or number of simultaneous erroneous bits) increases over a given threshold value. Of course, the sample has to be obtained uniformly from the total population of possible errors up to the selected multiplicity.

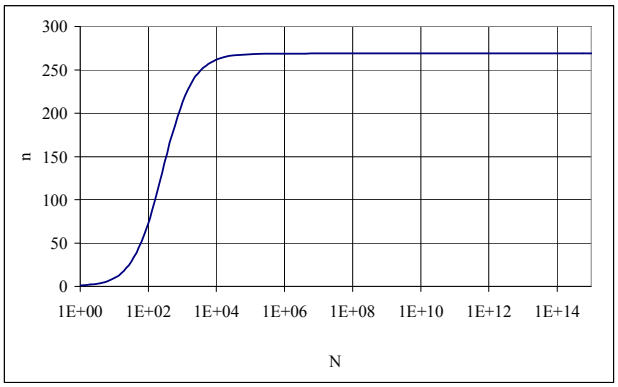


Figure 1. n as a function of N (p=0.5, margin of error equal to 5%, 90% confidence level).

E. Sensitivity Analysis: e Parameter

The definition of the margin of error is probably the most sensitive one. Reducing this margin quickly increases the required sample size (example in Figure 2).

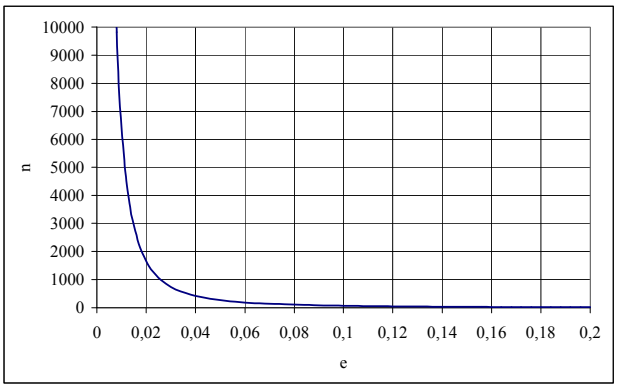


Figure 2. n as a function of e (p=0.5, N=150,000, 90% confidence level).

F. *Sensitivity Analysis: t Parameter*

Increasing the confidence level (and thus the t parameter) has less impact on n than reducing e. This will be shown hereafter on the example of the case study.

G. *Practical Implementation*

In order to correctly implement the approach, it is important to ensure that the sampling process is minimally biased, so that the sample is actually representative of the total population. As far as we know, the Mersenne-Twister algorithm [13] is one the most powerful to ensure the best uniform distribution. The results presented hereafter have therefore been obtained using samples selected with this algorithm.

III. CASE STUDY: A ROBUST AES COPROCESSOR

Experiments have been performed on several circuits to confirm the validity of the approach. The most complex case study was performed on a cryptographic coprocessor performing AES computations with continuous on-line detection of errors [14], assuming soft errors with unknown multiplicity. The circuit complexity is around 13K equivalent gates. Results presented in this paper are limited to this case study but the other experiments have also confirmed the validity of the proposed approach.

A. *Algorithm and Coprocessor Architecture*

The AES algorithm [15] is a round-iterated cipher which encrypts 128-bit input blocks with 128-, 192- or 256-bit keys. All iterations are virtually identical and are made of four operations: SubBytes, ShiftRows, MixColumns, AddRoundKey. SubBytes is a nonlinear byte substitution. This is the most complex operation of AES and ensures that a significant nonlinearity is introduced during the encryption. The elements implementing this function are referred to as S-Boxes. They are often described as substitution tables, synthesized as ROMs or multilevel combinatorial logic. However, an algebraic definition also exists and can be used to implement smaller and more power-efficient S-Boxes. The structure can then be easily pipelined into two or three stages. Figure 3 shows an inner pipeline stage made of three 4-bit registers.

The pipeline registers can be a target for fault-based attacks: the secret key can be easily found by computing few faulty encryptions and comparing the correct and the faulty results [17]. For this reason, several countermeasures have been proposed in the literature. Some of them are based on temporal redundancy: the same process is computed twice and the results compared, in order to detect any transient fault. The Double-Data-Rate technique [14] uses this approach in order to detect transient faults during the computation. However, since data must be processed and saved at each clock edge, the number of registers within the design increases. In particular, all the data registers are shadowed by another register of the same size, but triggered by the opposite clock edge. For instance, the 2-stage DDR implementation of the S-Box is depicted in Figure 4.

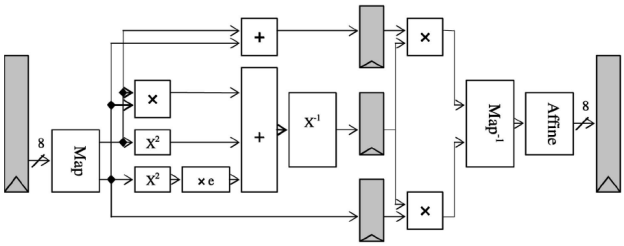


Figure 3. 2-stage pipelined implementation of AES S-Box in GF($(2^4)^2$) [16].

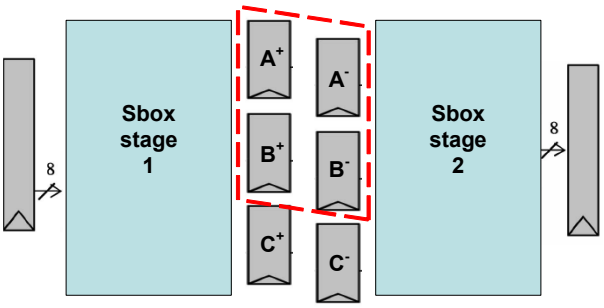


Figure 4. 2-stage pipelined implementation of AES S-Box with DDR-based on-line error detection.

B. *Campaign Specifications*

The fault injection campaign used as a case study is one of those aiming at evaluating the robustness of the DDR implementation of the S-Box with respect to errors with large multiplicity. In this campaign example, the injections were performed on two 4-bit registers and their DDR counterpart. This means that globally 16 bits were targeted. No assumption was made about the attacker, leading to assume 65535 possible error configurations. With the studied architecture, a single block encryption with a 128-bit key involves loading the data register during the first 4 clock cycles and then computing 10 round iterations: thus, a block can be encrypted or decrypted in 64 clock cycles. Almost every clock cycle of the computation process was considered, except the first and last ones, not meaningful for a successful attack. In consequence, the total number of errors is N = 65,535*62 = 4,063,170 since we only considered the case of attack durations inferior to 1 clock cycle.

A first campaign was performed, actually injecting all those errors. The errors were classified in four classes: detected,

undetected, silent, or false positive. The undetected errors correspond in fact to the cases where a register and its shadow have exactly the same error. False positives are obtained when the error detection is triggered but the encryption result is correct. Errors are silent if they are not detected but have no consequence on the application result.

Then, we computed the classification results from only a subset of these experiments. As shown in Table 1, as few as 385 experiments should be sufficient to achieve a 5% margin of error with a 95% confidence! This corresponds to only 0.009% of the possible errors. Increasing the confidence is not very costly since a 99.8% confidence would require 955 experiments (0.024% of the possible errors). Reducing the margin of error makes the sample size increase much more rapidly, as previously discussed. For the comparisons presented in this paper, we have performed the classification for ten different samples generated for margins of error equal to 5% and 1%, and confidence levels of 95% and 99%. The sample sizes are therefore 384, 663, 9581 and 16519. The largest samples correspond to only 4% of the possible errors.

TABLE I. VALUE OF THE SAMPLE SIZE (N) WITH RESPECT TO THE MARGIN OF ERROR E AND THE CONFIDENCE LEVEL (OR T PARAMETER) FOR N=4,063,170 AND P=0.5.

	t = 1.96 (95% conf.)	t = 2.5758 (99% conf.)	t = 3.0902 (99.8% conf.)
e = 5%	**385**	**663**	955
e = 1%	**9581**	**16519**	23734
e = 0.1%	776792	1177857	1503781

IV. RESULT ANALYSIS

The goal here is not to discuss the robustness of the AES implementation, but the conclusions obtained with either a complete fault injection campaign or a SFI campaign with the four selected sample sizes.

A. *Full Campaign Results*

The results of the complete fault injection campaign are summarized in Table 2. These values correspond to the real probability for one of the considered errors to be in one of the four classes. They have been used as a reference in the other experiments.

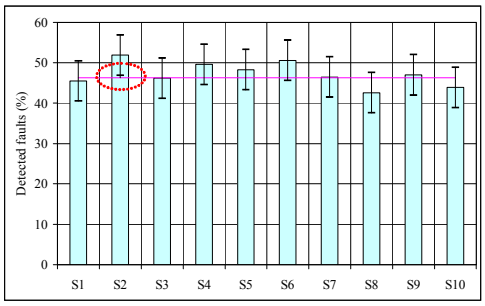
TABLE II. CLASSIFICATION OF ERROR EFFECTS AFTER A COMPLETE FAULT INJECTION CAMPAIGN.

Class	detected	undetected	silent	false pos.
Percentage	46.3	1.9	5.5	46.3

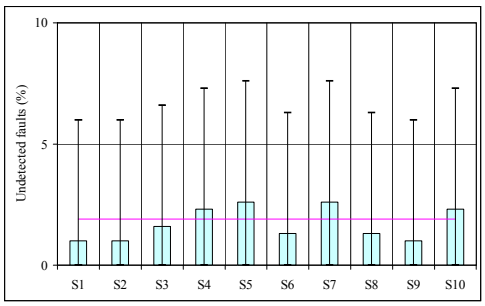
B. *SFI Campaign Results*

The results obtained with the ten smallest samples (n=385) are shown in Figure 5. In 38 cases out of 40, the real value is within the error margin. This corresponds to the 95% confidence level specified for this value of n. The two values that are out of the error margin are identified in Figure 5: percentage of detected faults for sample S2 and percentage of false positives for sample S8. In the first case, the error margin indicates a value between 46.9 and 56.9 while the correct value is 46.3. In the second case, the error margin indicates a value between 46.4 and 56.4 while the correct value is 46.3. In both cases, the real value is therefore not so far from the error interval.

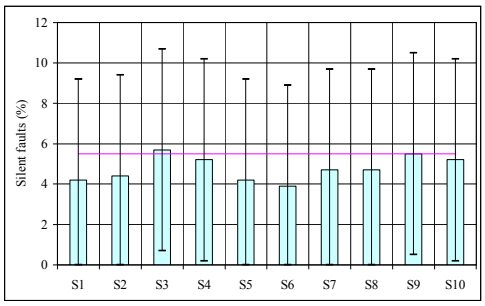
In all the other cases, with either an increased confidence level or a decreased margin of error, the real value was always within the error interval for the four classes.



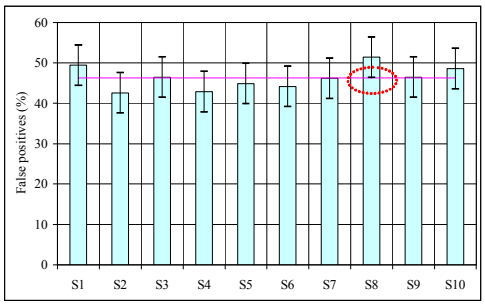
(a) Percentage of faults classified as detected



(b) Percentage of faults classified as undetected



(c) Percentage of faults classified as silent



(d) Percentage of faults classified as false positive

Figure 5. Results and computed margins of error obtained for the four classes with 10 different samples (n=385, N=4,063,170, p=0.5, e=5%, 95% confidence). The horizontal lines show the real values, evaluated with the full campaign and indicated in Table 2.

V. PREVIOUS WORK

As mentioned in the introduction, up to our knowledge no paper has reported similar experiments or has discussed the significance of a given number of injected faults without assumptions about the results.

The discussion in [18] is limited to the cases in which the studied property must hold for almost 100% of the injected faults, and no comparison with exhaustive injections is made.

In [19], a formula is very briefly introduced to justify a drastic reduction in the number of target flip-flops used during the experiments (300 instead of 150,000). However, the numerical applications in this paper do not match the mentioned formula (that does not exactly correspond to the one proposed here). Also, time sampling was done independently, 100 uniform sampling points being chosen without any justification (it is mentioned it was "chosen somewhat arbitrarily"). Nothing was said about the global margin of error and the global confidence level, while 30,000 RTL simulations were performed instead of the 150 trillion possible flips. Finally, once again, no comparison with exhaustive injections was provided.

Applying our formula to their specific case (with N=150 trillion) we find that 30,000 injections (i.e. an experimental time reduced by 7 orders of magnitude) lead to a 0.57% margin of error with 95% confidence level, or to a 0.1% margin of error with t=0.3464, i.e. a 27% confidence level, or to a 1% margin of error with t=3.4641, i.e. a confidence level of almost 100%. Depending on the exact goals of the experiment, a smaller number of injection experiments may therefore have been used with a reasonable margin of error and a reasonable confidence level.

VI. CONCLUSION

This paper presents a rigorous approach to determine either the sample size necessary during a SFI campaign, or the margin of error with a given confidence level. The experimental results show that it is possible to obtain very precise results about the robustness of a circuit while injecting only a very small portion of the possible errors. The mathematics presented here are not new, but up to our knowledge it is the first time such a study is reported in the context of SFI. The impact is very strong since the approach allows a designer to actually know the margin of error for a given SFI campaign while restricting the experimental time to the minimum. Two major limitations of current practice are therefore addressed at the same time. As illustrated by the presented case study, this controlled sampling process is particularly important when errors with large multiplicity have to be considered since the total number of potential errors becomes rapidly huge. This process is also very important in order to correctly quantify the robustness of large circuits running complex workloads.

ACKNOWLEDGMENT

This work has been partially supported by the French Research Ministry in the frame of the ASTER project (Minalogic).

REFERENCES

[1] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies", IEEE transactions of Device and Materials Reliability, vol. 5, no. 3, September 2005, pp. 305-316

[2] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, C. Whelan, "The sorcerer's apprentice guide to fault attacks", Proceedings of the IEEE, vol. 94, no. 2, February 2006, pp. 370-382

[3] R. Leveugle, K. Hadjiat, "Multi-level fault injections in VHDL descriptions: alternative approaches and experiments", Journal of Electronic Testing: Theory and Applications (JETTA), Kluwer, vol. 19, no. 5, October 2003, pp. 559-575

[4] A. Ammari, R. Leveugle, M. Sonza-Reorda, M. Violante, "Detailed comparison of dependability analyses performed at RT and gate levels", IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems, 2003, pp. 336-343

[5] G. C. Cardarilli, F. Kaddour, A. Leandri, M. Ottavi, S. Pontarelli, R. Velazco, "Bit-flip injection in processor-based architectures: a case study", 8th IEEE International On-Line Testing workshop, Isle of Bendor, France, July 8-10, 2002, pp. 117-127

[6] J. Gracia, J. C. Baraza, D. Gil, P. J. Gil, "Comparison and application of different VHDL-based fault injection techniques", IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems, 2001, pp. 233-241

[7] L. Berrojo, I. Gonzalez, F. Corno, M. Sonza-Reorda, G. Squillero, L. Entrena, C. Lopez, "New techniques for speeding up fault-injection campaigns", Design, Automation and Test in Europe Conference (DATE), March 4-8, 2002, pp. 847-852

[8] R. Leveugle, "Fault injection in VHDL descriptions and emulation", IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems, 2000, pp. 414-419

[9] P. Civera, L. Macchiarulo, M. Rebaudengo, M. Sonza Reorda, A. Violante, "Exploiting FPGA-based techniques for fault injection campaigns on VLSI circuits", IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems, 2001, pp. 250-258

[10] P. Vanhauwaert, R. Leveugle, P. Roche, "A flexible SoPC-based fault injection environment", 9th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS), Prague, Czech Republic, April 18-21, 2006, pp. 192-197

[11] Berenson Student CD-ROM – 10th edition, http://courses.wcupa.edu/rbove/Berenson/10th%20ed%20CD-ROM%20topics/section8_7.pdf

[12] R. Johnson, I. Miller, J. Freund, "Probability and Statistics for Engineers (7th Edition)", Prentice Hall

[13] M. Matsumoto, T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator", ACM Transactions on Modeling and Computer Simulation (TOMACS), vol. 8, no. 1, 2008, pp. 3-30

[14] P. Maistri, R. Leveugle, "Double-Data-Rate computation as a countermeasure against fault analysis", IEEE Transactions on Computers, vol. 57, no. 11, November 2008, pp. 1528-1539

[15] National Institute for Standards and Technology (NIST), "FIPS-197: Advanced Encryption Standard (AES)", Federal Information Processing Standards Publications, November 2001

[16] A. Hodjat, I. Verbauwhede, "Area-Throughput Trade-Offs for Fully Pipelined 30 to 70 Gbits/s AES Processors", IEEE Transactions on Computers, vol. 55, issue 4, 2006, pp. 366-372

[17] G. Piret, J.-J. Quisquater, "A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD", Workshop on Cryptographic Hardware and Embedded Systems (CHES 2003), Springer-Verlag, 2003, pp. 77-88

[18] F. M. Gonçalves, M. B. Santos, I. C. Teixeira, J. P. Teixeira, "Self-checking and fault tolerance quality assessment using fault sampling", IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems, 2002, pp. 216-224

[19] H. T. Nguyen, Y. Yagil, N. Seifert, M. Reitsma, "Chip-level soft error estimation method", IEEE Transactions of Device and Materials Reliability, vol. 5, no. 3, September 2005, pp. 365-381