

AN EFFICIENT AND ANALYTICAL APPROACH FOR SOFT ERROR RATE ESTIMATION IN DIGITAL CIRCUITS

1. VULAPU HIMA BINDU 2. R.RJESH KUMAR

1. M.Tech Student, Dept. of ECE, Malineni Lakshmaiah Engineering College, Singarayakonda, A.P

2. Associate Professor, Dept. of ECE, Malineni Lakshmaiah Engineering College, Singarayakonda, A.P

ABSTRACT:

Soft errors due to cosmic rays cause reliability problems during lifetime operation of digital systems, which increase exponentially with Moore's law. The first step in developing efficient soft error tolerant schemes is to analyze the effect of soft errors at the system level. This enhanced concept develops a systematic approach for soft error rate estimation. The proposed framework is divided in two stages. First, signal probabilities are computed via a hybrid approach combining heuristics and selective simulation of reconvergent subnets. In the second stage, signal probabilities are used to compute the vulnerability of all the gates in a combinational block using an incremental algorithm. Experiments on benchmark circuits and comparison of the results with random fault injection (previous work) show that proposed method is accurate while orders of magnitude faster.

INTRODUCTION:

Soft errors, also called transient errors, are intermittent malfunctions of the hardware that are not reproducible [1]. Soft errors arise from Single Event Upsets (SEU), which are caused by energetic particles (neutrons and alpha particles). Soft Error Rate (SER) for a device is defined as the error rate due to SEUs, which depends on both the particle flux and circuit characteristics. Device circuit parameters that influence the error rate include the amount of charge stored, the vulnerable cross-sectional area, and the charge collection efficiency. Device scaling significantly affects the susceptibility of integrated circuits to soft errors [2]. As the feature size shrinks, the amount of charge per device decreases enabling a particle strike to be much more likely to cause an error. As a result, particles of lower energy, which are far more plentiful, can generate sufficient charge to cause a soft error. Hence, in the absence of error correction schemes, the error rate of vulnerable parts will grow in direct proportion to the number of bits on the chip [3]. So far, memory elements have been more susceptible to soft errors than the combinational logic. However, analytical models predict that the soft error rate in the combinational logic will be comparable to that of memory elements by 2011 [2]. Soft error avoidance techniques such as shielding, Silicon-On-Insulator (SOI), and radiation-hardened can only reduce the effect of soft error while introducing significant amount of area and performance penalty. The first step

in developing soft error tolerant scheme with low cost and performance penalties is to estimate the system failure rate due to soft errors and the contribution of each component to the overall system failure rate. Previous work on SER estimation is based on fault injection using random simulations and hence inaccurate and very time-consuming [1][2][3].

SER ESTIMATION METHODS:

SER estimation methods of combinational circuits can generally be classified into two categories, depending on how they evaluate the effects of SETs and their reliance on simulation: 1) dynamic approaches and 2) static approaches. In dynamic approaches, a transient fault is injected into the circuit and the circuit operation is simulated for different test vectors to calculate the number of faults stored in the circuit leading to soft errors [9]. However, some of dynamic approaches use the pre-characterization of circuit elements for simulation [8]. On the other hand, static approaches are symbolic or analytical methods which are used for estimating circuit SER. Symbolic methods use specific data structures such as matrix or graph to maintain information about triple masking factors (logical, electrical, timing) [27]. Analytical methods exploit the mathematical relationships such as Boolean satisfaction [13], or probability relationships to estimate SER [14][20][29][30]. In Boolean satisfaction-based approaches, the SER

Copyright @ 2018 ijearst. All rights reserved.

**INTERNATIONAL JOURNAL OF ENGINEERING IN ADVANCED RESEARCH
SCIENCE AND TECHNOLOGY**

Volume.02, IssueNo.03, October -2018, Pages: 642-649

estimation issue is modeled as an equivalent Boolean satisfaction problem, and the exact value of SER can be found by solving that problem. In probability methods, probability of fault propagation toward main outputs and flip-flops is calculated using signals probability and some mathematical equations to model triple masking factors.

Today, reliability estimation plays an important role in the design flow and SER has become one of the most important reliability factors in VLSI circuits. Hence, the SER estimation problem is a significant challenge for circuit designers. During an optimization procedure, a change applies to the circuit to reduce one of the circuit parameters such as power, delay, or SER. Then, the circuit is re-analyzed to determine the improvement in the parameter of interest (here SER) obtained from the applied change.

ANALYTICAL SER ESTIMATION METHOD:

A typical synchronous circuit consists of combinational logic and flip-flops (Fig. 1). Primary Inputs (PIs) and the outputs of flip-flops (PIFF) are inputs of combinational logic (CL). Also, Primary Outputs (POs) and the inputs of the FFs (POFF) are outputs of CL. To compute the error rate of a node in a circuit, three probability factors are required to be computed: () () () SEU i latched i sensitized i $R_n \times P_n \times P_n$. These parameters are defined as follows: • RSEU(ni) is the occurrence rate of SEUs at node ni to cause a glitch at the output of the gate. This parameter depends on the energy of the particle, type and the size of the gate, and device characteristics. • Platched(ni) is the probability that an erroneous value reaching the flip-flop inputs is latched. • Psensitized(ni) is the probability that node ni is functionally sensitized by the input vectors to propagate the erroneous value from the error site to POs/FFs. RSEU(ni) can be obtained from layout information of library cells, technology parameters, and particle energy [1][4][5]. Platched(ni) estimation consists of logic and timing derating. Logic derating is the probability that an erroneous value is propagated to the input of a flip-flop. Timing derating is the probability that there is an overlap between the width of an error glitch and the latching window of a reachable flip-flop. In this work we focus on the estimation of the logic part of Platched(ni) and Psensitized(ni). This is based on the fact that the error propagation probability computation is the most timeconsuming part of SER estimation. The error sites considered in this paper are all circuit nodes (inputs and output of all gates and FFs). In the proposed approach, we first extract the structural paths from the error sites to all reachable outputs and then traverse these paths to compute the propagation probability of the erroneous value to the

reachable primary outputs or to the reachable flip-flops. Based on the error site, we categorize nets and gates in the circuit as follows. An on-path signal is a net on a path from the error site to a reachable output. Also, an on-path gate is defined as the gate with at least one on-path input. Finally, an off-path signal is a net that is not on-path and is an input of an on-path gate. These three are also shown in Fig. 2. For error propagation probability calculation, as we traverse the paths, we use signal probability for off-path signals and use our propagation rules for on-path signals. The signal probability (SP) of a line l indicates the probability of l having logic value 1 [6]. SP techniques have been presented in [7] [8]. The problem statement can be described as follows: Given the SEU probability in node ni calculate the probability of the propagation of this error to POs/FFs (i.e., system failure).

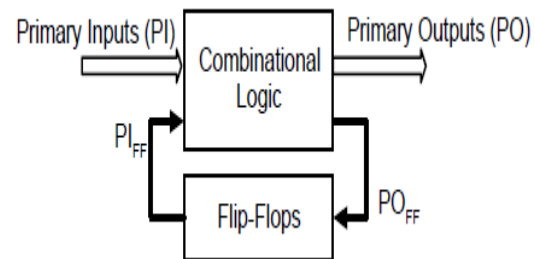


Fig. 1: A typical block diagram of synchronous sequential circuits

Errors can be directly propagated to a primary output and cause a system failure at the same clock cycle, or they can be propagated to flip-flops repeatedly, and finally manifest as errors at a primary output several clock cycles later. First, consider a simple case when there is only one path from the error site to an output. As we traverse this path gate by gate, the error propagation probability from an on-path input of a gate to its output depends on the type of the gate and the signal probability of other off-path signals. In the example shown in Fig. 3, the error propagation probability to the output of the gate C (AND gate) is the product of the probability of the output of gate A being 1 and the error probability at the PI ($1 \times 0.2 = 0.2$). Similarly, EPP at the output of the gate D (OR gate) is calculated as $0.2 \times (1 - \text{SP}_B) = 0.2 \times 0.7 = 0.14$. In the general case in which reconvergent paths might exist, the propagation probability from the error site to the output of the reconvergent gate depends on not only the type of the gate and the signal probabilities of the off-path signals, but also the polarities of the propagated error on the on-path signals. To address

this issue, we need propagation rules for reconvergent gates. First, we define the $Pa(U_i)$, $P\bar{a}(U_i)$, $P1(U_i)$, and $P0(U_i)$ as follow: • $Pa(U_i)$ and $P\bar{a}(U_i)$ are defined as the probability of the output of node U_i being a and $\bar{a}$, respectively, where $\bar{a}$ is inverted of a. In other words, $Pa(U_i)$ is the probability that the erroneous value is propagated from the error site to U_i with an even number of inversions, whereas $P\bar{a}(U_i)$ is the similar propagation probability with an odd number of inversions. • $P1(U_i)$ and $P0(U_i)$ are defined as the probability of the output of node U_i being 1 and 0, respectively. In these cases, the error is blocked and not propagated. Note that $Pa(U_i) + P\bar{a}(U_i) + P1(U_i) + P0(U_i) = 1$.

MASKING EFFECTS:

Modeling and analyzing the SER in logic is more complex than in memory elements, since there are some well-known masking effects that reduce the overall likelihood that a pulse caused by a particle strike is latched and results in an error. These masking effects are commonly classified as:

- Logical masking: Transient faults are masked by gates whose output is independent of the faulty input (e.g., an OR gate with an input set to 1). To model the logical masking, first we need to introduce the concepts of time interval and signal probability: To model the logical masking, first we need to introduce the concepts of time interval and signal probability: Time interval: In a set of hybrid pulses, each one consists of several single-pulses. If we sort the time of one (zero) to zero (one) transitions of all hybrid pulses in ascending order, the timing distance between two consecutive transitions is called a time interval. It is obvious that the leading and trailing transitions of a time interval do not necessarily have an opposite polarity and both may be from one type; a one (zero) to zero (one) transition. For example, all time intervals related to two single-pulses are shown in Figure 4. Signal probability: signal probability of node k is demonstrated by sp_k indicating the probability of that node k has logical value one ('1') [14]. Signal probability is considered for those gate inputs on which there is no transient pulse.

- Electrical masking: The pulse is attenuated (either its amplitude is reduced or rise/fall times are increased) by the electrical properties of the gates throughout the logic chain, and the resulting magnitude is insufficient to change the value that is latched. As the pulse traverses through the chain of logic gates in the combinational part, it may be attenuated and disappeared due to electrical properties of gates. To model the electrical masking, we use the mathematical equations presented in [10]. In this model, the falling

and rising time delays of all gate cells in the library are calculated. Then, the amplitude of the output pulse is computed based on the input pulse width (PW_i), the falling and the rising delays of the output pulse.

$$V_{o_{min}} = \begin{cases} VDD \left(1 - \frac{PW_i}{1.25 \times t_f}\right) & \text{if: } PW_i < t_f \times 1.25 \\ 0 & \text{if: } PW_i > t_f \times 1.25 \end{cases} \quad (1)$$

$$V_{o_{max}} = \begin{cases} VDD \left(\frac{PW_i}{1.25 \times t_r}\right) & \text{if: } PW_i < t_r \times 1.25 \\ VDD & \text{if: } PW_i > t_r \times 1.25 \end{cases} \quad (2)$$

Then, the pulse width is computed as [10]:

$$PW_o = (PW_i - T_{Delay1}) + T_{Delay2} \times \frac{V_{DD}/2 - V_{o_{min}}}{V_{DD}/2}, 0 \rightarrow 1 \quad (3)$$

$$PW_o = (PW_i - T_{Delay1}) + T_{Delay2} \times \frac{V_{o_{max}} - V_{DD}/2}{V_{DD}/2}, 1 \rightarrow 0$$

- Timing masking: The pulse arrives at a state-holding element out of its latching-time window. After a transient pulse propagates through the combinational circuit and arrives at a flip-flop, it is stored in the flip-flop and leads to a soft error only if it completely overlaps the latching window of the flip-flop. Otherwise, it is not stored in the flip-flop and a timing masking occurs. The probability that a transient pulse latches at a reachable flip-flop is called Latching Probability (LP). Since the pulse width changes as it is travelling through various gates, after computing the pulse width on the input of a flip-flop, the latching probability is calculated as: $LP = (S + H + W)/T$ where S is the setup time, H is the hold Time, W is the pulse width, and T is the clock period.

INCREMENTAL SOFT ERROR RATE ESIMATION

The circuit delay is one of the most important factors affecting the SER as it plays the key role in the logical, electrical, and time masking mechanisms:

- Logical masking: overlapping of two single pulses with the same polarity that has been arrived at separate inputs of a gate will increase the error propagation probability.

- Electrical masking: the gate delay has a significant impact on this masking. As the delay of a gate

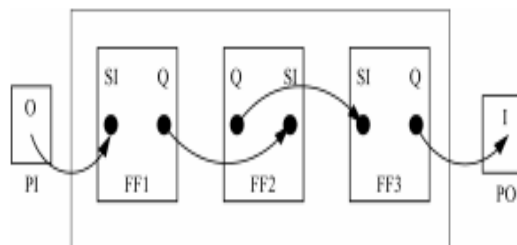
increases, the output pulse attenuation becomes greater.

□ Time masking: the latching probability of a transient pulse within a flip-flop relates to the time it reaches a flip-flop which is directly related to delay of gates through which the pulse has been traversed.

So, in this section, it is assumed that the changed parameter made by the designer is the gate delay. This change could be the result of changing the gate sizing, the threshold voltage, or other technology parameters [24][25][26]. In the following, we introduce a technique to re-estimate the SER resulting from this temporal change. Before describing the proposed algorithm, we explain some concepts.

PROPOSED TECHNIQUE: SCAN CHAIN REORDERING

In VLSI design for testability, a scan chain is commonly used to connect the shift registers that store the input and output vectors during the testing phase of manufacturing. Registers in the scan chain are connected as a single path with ends of the path connected to a primary input (PI) pad and a primary output (PO) pad. Test input values are shifted into the registers through the PI pad; then, a test is performed and the test output values are shifted out through the PO pad. Figure 1 depicts a simple example of a scan chain.



One of the primary objectives in design-for-testability is to minimize the impact of test circuitry on chip performance and cost. Thus, it is essential to minimize the wirelength of a scan chain: this decreases wiring congestion and/or reduces the chip area while, at the same time, increasing signal speed by reducing capacitive loading effects on nets that share register pins with the scan chain. Previous placement-based scan chain ordering approaches compute the cost of stitching one flip-flop to another as either cell-to-cell Manhattan distance [Hirech et al. 1998; Makar 1998; Barbagello et al. 1996] or pin-to-pin Manhattan distance [Boese et al. 1994; Kobayashi et al. 1999].

The former metric gives a symmetric TSP, while the latter gives rise to an almost symmetric TSP [Boese et al. 1994]. The fundamental assumption in all current work on layout-driven scan chain ordering is that the wirelength overhead due to scan insertion is equal to the Manhattan distance between the scan-in and scan-out pins of the flip-flops. However, this assumption is incorrect: the scan connection need only reach the output net, not the output pin. In this work, we propose a (trial) routing-based flow for scan chain ordering that uses the incremental routing cost (connecting to existing or anticipated routing, rather than to the output pin) as the cost measure for a scan connection. This is in contrast to existing placement-based methods which use simply the Manhattan distance from the flip-flop output pin to the scan-in pin of the other flip-flop as the cost measure. Under our formulation, the resulting Asymmetric Traveling Salesman Problem (ATSP) may be highly nonmetric. We give an efficient method to calculate the costs of the ATSP instance based on a trial routing of nonscan nets. Our work considers the possibility of using both Q and Q^- pins of the flip-flop to make any given scan connection, and it also extends to timing- and noise-driven scan chain ordering (in a more detailed routing-driven context).

Also assume that there is a single scan chain in the circuit. As there exists a high correlation between the switching activities in the internal nodes of the circuit with the transitions taking place in the scan cells we further assume that the primary inputs are directly controllable and all the switching activities in the circuit is due to transitions in the scan cells. Our objective is two-fold. They are the following.

- (i) To determine the order of interconnection between the scan cells such that the total power consumption due to toggling is minimized.
- (ii) To identify the input and output scan cells in the scan chain.

This will reduce the overall average power and also the peak power that may arise while scanning out a captured response. The problem of the capture power (peak power in the test cycle) will be solved by using a novel algorithm that will reorder some cells in the scan chain in such a way that minimizes the Hamming distance between the applied test vector and the captured response in the test cycle, hence reducing the test cycle peak power (capture power). In this scan-chain-ordering algorithm, some cells of the ordered scan chain using the algorithm will be reordered again in order to reduce the peak power which may result during the test cycle.

Copyright @ 2018 ijeerst. All rights reserved.

INTERNATIONAL JOURNAL OF ENGINEERING IN ADVANCED RESEARCH
SCIENCE AND TECHNOLOGY

Volume.02, IssueNo.03, October -2018, Pages: 642-649

The screenshot displays the Intel Quartus II software interface. The top menu bar includes File, Edit, View, Simulation, Window, Layout, and Help. The main workspace is divided into several panes. On the left, the 'Instances and Processes' pane shows a tree structure for the 'final' project, with 'final_top1/184' selected. The 'Simulation Objects for final' pane shows a list of objects including 'clk', 'rst', 'en0[0]', 'en1[0]', 'en2[0]', 'en3[0]', 'en4[0]', 'en5[0]', 'en6[0]', 'en7[0]', 'en8[0]', 'en9[0]', 'en10[0]', 'en11[0]', 'en12[0]', 'en13[0]', 'en14[0]', 'en15[0]', 'en16[0]', 'en17[0]', 'en18[0]', 'en19[0]', 'en20[0]', 'en21[0]', 'en22[0]', 'en23[0]', 'en24[0]', 'en25[0]', 'en26[0]', 'en27[0]', 'en28[0]', 'en29[0]', 'en30[0]', 'en31[0]', 'en32[0]', 'en33[0]', 'en34[0]', 'en35[0]', 'en36[0]', 'en37[0]', 'en38[0]', 'en39[0]', 'en40[0]', 'en41[0]', 'en42[0]', 'en43[0]', 'en44[0]', 'en45[0]', 'en46[0]', 'en47[0]', 'en48[0]', 'en49[0]', 'en50[0]', 'en51[0]', 'en52[0]', 'en53[0]', 'en54[0]', 'en55[0]', 'en56[0]', 'en57[0]', 'en58[0]', 'en59[0]', 'en60[0]', 'en61[0]', 'en62[0]', 'en63[0]', 'en64[0]', 'en65[0]', 'en66[0]', 'en67[0]', 'en68[0]', 'en69[0]', 'en70[0]', 'en71[0]', 'en72[0]', 'en73[0]', 'en74[0]', 'en75[0]', 'en76[0]', 'en77[0]', 'en78[0]', 'en79[0]', 'en80[0]', 'en81[0]', 'en82[0]', 'en83[0]', 'en84[0]', 'en85[0]', 'en86[0]', 'en87[0]', 'en88[0]', 'en89[0]', 'en90[0]', 'en91[0]', 'en92[0]', 'en93[0]', 'en94[0]', 'en95[0]', 'en96[0]', 'en97[0]', 'en98[0]', 'en99[0]', 'en100[0]', 'en101[0]', 'en102[0]', 'en103[0]', 'en104[0]', 'en105[0]', 'en106[0]', 'en107[0]', 'en108[0]', 'en109[0]', 'en110[0]', 'en111[0]', 'en112[0]', 'en113[0]', 'en114[0]', 'en115[0]', 'en116[0]', 'en117[0]', 'en118[0]', 'en119[0]', 'en120[0]', 'en121[0]', 'en122[0]', 'en123[0]', 'en124[0]', 'en125[0]', 'en126[0]', 'en127[0]', 'en128[0]', 'en129[0]', 'en130[0]', 'en131[0]', 'en132[0]', 'en133[0]', 'en134[0]', 'en135[0]', 'en136[0]', 'en137[0]', 'en138[0]', 'en139[0]', 'en140[0]', 'en141[0]', 'en142[0]', 'en143[0]', 'en144[0]', 'en145[0]', 'en146[0]', 'en147[0]', 'en148[0]', 'en149[0]', 'en150[0]', 'en151[0]', 'en152[0]', 'en153[0]', 'en154[0]', 'en155[0]', 'en156[0]', 'en157[0]', 'en158[0]', 'en159[0]', 'en160[0]', 'en161[0]', 'en162[0]', 'en163[0]', 'en164[0]', 'en165[0]', 'en166[0]', 'en167[0]', 'en168[0]', 'en169[0]', 'en170[0]', 'en171[0]', 'en172[0]', 'en173[0]', 'en174[0]', 'en175[0]', 'en176[0]', 'en177[0]', 'en178[0]', 'en179[0]', 'en180[0]', 'en181[0]', 'en182[0]', 'en183[0]', 'en184[0]', 'en185[0]', 'en186[0]', 'en187[0]', 'en188[0]', 'en189[0]', 'en190[0]', 'en191[0]', 'en192[0]', 'en193[0]', 'en194[0]', 'en195[0]', 'en196[0]', 'en197[0]', 'en198[0]', 'en199[0]', 'en200[0]', 'en201[0]', 'en202[0]', 'en203[0]', 'en204[0]', 'en205[0]', 'en206[0]', 'en207[0]', 'en208[0]', 'en209[0]', 'en210[0]', 'en211[0]', 'en212[0]', 'en213[0]', 'en214[0]', 'en215[0]', 'en216[0]', 'en217[0]', 'en218[0]', 'en219[0]', 'en220[0]', 'en221[0]', 'en222[0]', 'en223[0]', 'en224[0]', 'en225[0]', 'en226[0]', 'en227[0]', 'en228[0]', 'en229[0]', 'en230[0]', 'en231[0]', 'en232[0]', 'en233[0]', 'en234[0]', 'en235[0]', 'en236[0]', 'en237[0]', 'en238[0]', 'en239[0]', 'en240[0]', 'en241[0]', 'en242[0]', 'en243[0]', 'en244[0]', 'en245[0]', 'en246[0]', 'en247[0]', 'en248[0]', 'en249[0]', 'en250[0]', 'en251[0]', 'en252[0]', 'en253[0]', 'en254[0]', 'en255[0]', 'en256[0]', 'en257[0]', 'en258[0]', 'en259[0]', 'en260[0]', 'en261[0]', 'en262[0]', 'en263[0]', 'en264[0]', 'en265[0]', 'en266[0]', 'en267[0]', 'en268[0]', 'en269[0]', 'en270[0]', 'en271[0]', 'en272[0]', 'en273[0]', 'en274[0]', 'en275[0]', 'en276[0]', 'en277[0]', 'en278[0]', 'en279[0]', 'en280[0]', 'en281[0]', 'en282[0]', 'en283[0]', 'en284[0]', 'en285[0]', 'en286[0]', 'en287[0]', 'en288[0]', 'en289[0]', 'en290[0]', 'en291[0]', 'en292[0]', 'en293[0]', 'en294[0]', 'en295[0]', 'en296[0]', 'en297[0]', 'en298[0]', 'en299[0]', 'en300[0]', 'en301[0]', 'en302[0]', 'en303[0]', 'en304[0]', 'en305[0]', 'en306[0]', 'en307[0]', 'en308[0]', 'en309[0]', 'en310[0]', 'en311[0]', 'en312[0]', 'en313[0]', 'en314[0]', 'en315[0]', 'en316[0]', 'en317[0]', 'en318[0]', 'en319[0]', 'en320[0]', 'en321[0]', 'en322[0]', 'en323[0]', 'en324[0]', 'en325[0]', 'en326[0]', 'en327[0]', 'en328[0]', 'en329[0]', 'en330[0]', 'en331[0]', 'en332[0]', 'en333[0]', 'en334[0]', 'en335[0]', 'en336[0]', 'en337[0]', 'en338[0]', 'en339[0]', 'en340[0]', 'en341[0]', 'en342[0]', 'en343[0]', 'en344[0]', 'en345[0]', 'en346[0]', 'en347[0]', 'en348[0]', 'en349[0]', 'en350[0]', 'en351[0]', 'en352[0]', 'en353[0]', 'en354[0]', 'en355[0]', 'en356[0]', 'en357[0]', 'en358[0]', 'en359[0]', 'en360[0]', 'en361[0]', 'en362[0]', 'en363[0]', 'en364[0]', 'en365[0]', 'en366[0]', 'en367[0]', 'en368[0]', 'en369[0]', 'en370[0]', 'en371[0]', 'en372[0]', 'en373[0]', 'en374[0]', 'en375[0]', 'en376[0]', 'en377[0]', 'en378[0]', 'en379[0]', 'en380[0]', 'en381[0]', 'en382[0]', 'en383[0]', 'en384[0]', 'en385[0]', 'en386[0]', 'en387[0]', 'en388[0]', 'en389[0]', 'en390[0]', 'en391[0]', 'en392[0]', 'en393[0]', 'en394[0]', 'en395[0]', 'en396[0]', 'en397[0]', 'en398[0]', 'en399[0]', 'en400[0]', 'en401[0]', 'en402[0]', 'en403[0]', 'en404[0]', 'en405[0]', 'en406[0]', 'en407[0]', 'en408[0]', 'en409[0]', 'en410[0]', 'en411[0]', 'en412[0]', 'en413[0]', 'en414[0]', 'en415[0]', 'en416[0]', 'en417[0]', 'en418[0]', 'en419[0]', 'en420[0]', 'en421[0]', 'en422[0]', 'en423[0]', 'en424[0]', 'en425[0]', 'en426[0]', 'en427[0]', 'en428[0]', 'en429[0]', 'en430[0]', 'en431[0]', 'en432[0]', 'en433[0]', 'en434[0]', 'en435[0]', 'en436[0]', 'en437[0]', 'en438[0]', 'en439[0]', 'en440[0]', 'en441[0]', 'en442[0]', 'en443[0]', 'en444[0]', 'en445[0]', 'en446[0]', 'en447[0]', 'en448[0]', 'en449[0]', 'en450[0]', 'en451[0]', 'en452[0]', 'en453[0]', 'en454[0]', 'en455[0]', 'en456[0]', 'en457[0]', 'en458[0]', 'en459[0]', 'en460[0]', 'en461[0]', 'en462[0]', 'en463[0]', 'en464[0]', 'en465[0]', 'en466[0]', 'en467[0]', 'en468[0]', 'en469[0]', 'en470[0]', 'en471[0]', 'en472[0]', 'en473[0]', 'en474[0]', 'en475[0]', 'en476[0]', 'en477[0]', 'en478[0]', 'en479[0]', 'en480[0]', 'en481[0]', 'en482[0]', 'en483[0]', 'en484[0]', 'en485[0]', 'en486[0]', 'en487[0]', 'en488[0]', 'en489[0]', 'en490[0]', 'en491[0]', 'en492[0]', 'en493[0]', 'en494[0]', 'en495[0]', 'en496[0]', 'en497[0]', 'en498[0]', 'en499[0]', 'en500[0

[illegible]

CONCLUSION:

Soft errors due to single event upsets are the main reliability threat of digital systems. In particular, vulnerability of digital systems grows in direct proportion to the Moore's law. In this paper, an accurate propagation probability computation technique has been developed, which significantly reduces the SER estimation time. The proposed approach leverages the signal probability calculation, which is already used in other steps of the design flow, and computes the error propagation probability. Some efficient graph-based algorithms have been used for this computation. To improve the accuracy of our approach, we have considered the output dependencies. Experiments on benchmark circuits and comparison of the results with the random simulation technique show the effectiveness and the accuracy of the presented approach.

- [1] S. Almkukhaizim, Y. Makris, Y. S. Yang, and A. Veneris, "Seamless Integration of SER in Rewiring-Based Design Space Exploration," in Proc. International Test Conference, 2006, pp. 1–9.
- [2] G. Asadi and M. B. Tahoori, "An Accurate SER Estimation Method Based on Propagation Probability," Proc. Design Automation and Test in Europe Conf, pp. 306–307, 2005.
- [3] V. Carreno, G. Choi, and R. K. Iyer, "Analog-Digital Simulation of Transient-Induced Logic Errors and Upset Susceptibility of an Advanced Control System," in NASA Technical Memo 4241, 1990.
- [4] D. L. Chenette, J. Chen, E. Clayton, T. G. Guzik, J. P. Wefel, M. Garcia-Munoz, C. Lopate, K. R. Pyle, K. P. Ray, E. G. Mullen, and D. A. Hardy, "The CRRES/SPACERAD Heavy Ion Model of the Environment (CHIME) for Cosmic Ray and Solar Particle Effects on Electronic and Biological Systems in Space," IEEE Trans. on Nuclear Science, vol. 41, no. 6, pp. 2332–2339, 1994.
- [5] A. Dharchoudhury, S. M. Kang, H. Cha, and J. H. Patel, "Fast Timing Simulation of Transient Faults in Digital Circuits," in Proc. IEEE/ACM International Conference on Computer-Aided Design, 1994, pp. 719–722.
- [6] P. E. Dodd, F. W. Sexton, G. L. Hash, M. R. Shaneyfelt, B. L. Draper, A. J. Farino, and R. S. Flores, "Impact of Technology Trends on SEU in CMOS SRAMs," IEEE Trans. Nuclear Science, vol. 43, no. 6, pp. 2797–2804, Dec. 1996.
- [7] Graham, "Soft errors a problem as SRAM geometries shrink," <http://www.ebnews.com/story/OEG20020128S0079>, ebn, 28 Jan 2002.

- [8] K. J. Hass and J. W. Ambles, "Single Event Transients in Deep Submicron CMOS," 1999.
- [9] T. Heijmen and A. Nieuwland, "Soft-Error-Rate Testing of Deep-Submicron Integrated Circuits," in Proc. Eleventh IEEE European Test Symposium, 2006, pp. 247–252.
- [10] B. Ingols and A. Rambaud, "iRoC Releases Robust SPARC Test Report," <http://www.us.designreuse.com/news/news65.html>, 2002.