A Fault Tolerance Technique for Combinational Circuits Based on Selective-Transistor Redundancy

Ahmad T. Sheikh, Aiman H. El-Maleh, Muhammad E. S. Elrabaa, and Sadiq M. Sait

***Abstract*— With fabrication technology reaching nanolevels, systems are becoming more prone to manufacturing defects with higher susceptibility to soft errors. This paper is focused on designing combinational circuits for soft error tolerance with minimal area overhead. The idea is based on analyzing random pattern testability of faults in a circuit and protecting sensitive transistors, whose soft error detection probability is relatively high, until desired circuit reliability is achieved or a given area overhead constraint is met. Transistors are protected based on duplicating and sizing a subset of transistors necessary for providing the protection. In addition to that, a novel gate-level reliability evaluation technique is proposed that provides similar results to reliability evaluation at the transistor level (using SPICE) with the orders of magnitude reduction in CPU time. LGSynth'91 benchmark circuits are used to evaluate the proposed algorithm. Simulation results show that the proposed algorithm achieves better reliability than other transistor sizing-based techniques and the triple modular redundancy technique with significantly lower area overhead for 130-nm process technology at a ground level.**

***Index Terms*— Fault tolerance, logic synthesis, radiation hardening, single event multiple upsets, single event transient (SET), single event upset (SEU), soft error tolerance.**

I. Introduction

DUE to advancements in CMOS technology and shrinking of feature size to the nanometer scale, studies have indicated that high-density chips will not only be increasingly accompanied by manufacturing defects but also susceptible to dynamic faults during chip operation [1], [2]. Nanoscale devices are limited by several characteristics; most dominant are the device's higher defect rates and the increased susceptibility to soft errors. Both of these types of errors affect the operations of a circuit if they are not addressed. Reliability of a circuit can be defined as its ability to function properly despite the existence of such errors.

Transient (soft) errors can arise due to multiple sources. These include high-energy particles, coupling, power supply noise, leakage, and temporal circuit variations. A soft error leads to transient error(s), which can last for one or several clock cycles. A single event transient (SET) occurs when a charged particle hits the combinational logic resulting in a transient current pulse. If this transient has enough width and magnitude, it can result in an erroneous value at the gate output. If the erroneous value is latched at a memory element, an SET becomes a single event upset (SEU). A single SET can produce multiple transient current pulses at the output [3]. This is due to the logic fan-out in the circuit.

Ziegler *et al.* [4] presented intensive experimental study over the period of 15 years to evaluate the radiation-induced soft fails in large scale integrated electronics at different terrestrial altitudes. Baumann [5] highlighted the dominant sources responsible for the creation of soft errors in terrestrial applications. Shivakumar *et al.* [6] modeled the effects of soft errors in memory devices and logic devices and demonstrated that with each technology generation, soft errors will increase by orders of magnitude in logic devices and projected that soft errors in logic devices will be comparable to that of memory devices. The minimum charge required to create a soft error in a transistor is referred to as Q_{crit}. It has been shown that Q_{crit} is going to be reduced with technology improvement and with the advent of low-power devices [6], [7].

In this paper, we propose a selective-transistor scaling method that protects individual sensitive transistors of a circuit. A sensitive transistor is a transistor whose soft error detection probability is relatively high. This is in contrast to previous approaches where all transistors, series transistors, or those transistors connected to the output of a sensitive gate, whose soft error detection probability is relatively high, are protected. Transistor duplication and asymmetric transistor sizing are applied to protect the most sensitive transistors of the circuit. In asymmetric sizing, nMOS and pMOS transistors are sized independently. Reliability is evaluated for different protection thresholds and area overhead constraints. Finally, a novel gate-level soft error reliability evaluation technique for combinational circuits is proposed that produces similar results as produced by transistor-level simulations (using SPICE), but with orders of magnitude reduction in CPU time.

The rest of this paper is organized as follows. Section II reviews some of the contemporary fault tolerance techniques. Section III highlights the motivation and rationale behind the proposed approach. Section IV presents the proposed selective-transistor-redundancy (STR) algorithm. Reliability evaluation technique is discussed in Section V. Simulation results are elaborated in Section VI. Finally, this paper is concluded in Section VII.

Manuscript received November 29, 2015; revised April 1, 2016; accepted May 10, 2016. Date of publication May 30, 2016; date of current version December 26, 2016. This work was supported by the Deanship of Scientific Research at the King Fahd University of Petroleum and Minerals under Grant IN131014. *(Corresponding author: Aiman H. El-Maleh.)*

A. T. Sheikh is with the College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia (e-mail: atsheikh@kfupm.edu.sa).

A. H. El-Maleh, M. E. S. Elrabaa, and S. M. Sait are with the Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia (e-mail: aimane@kfupm.edu.sa; elrabaa@kfupm.edu.sa; sadiq@kfupm.edu.sa).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2016.2569532

II. Related Literature

Reliability in systems can be achieved by redundancy. Redundancy can be added at the module level, gate level, transistor level [8], or even at the software level [9]. Design of reliable systems by using redundant unreliable components was proposed in [10]. Since then, plethora of research has been done to rectify soft errors in combinational and sequential circuits by applying hardware redundancy [11], [12]. Triple modular redundancy (TMR), a popular and widely used technique, creates three identical copies of the system and combines their outputs using a majority voter [13], [14]. The generalized modular redundancy [15] scheme considers the probability of occurrence of each combination at the output of a circuit. The redundancy is then added to only protect those combinations that have high probability of occurrence, while the remaining combinations are left unprotected to save area. El-Maleh and Al-Qahtani [16] proposed a fault tolerance technique for sequential circuits that enhances the reliability of sequential circuits by introducing redundant equivalent states for states with high probability of occurrence.

Mohanram and Touba [17] proposed a partial error masking scheme based on TMR, which targets the nodes with the highest soft error susceptibility. Two reduction heuristics are used to reduce soft error failure rate, namely, cluster sharing reduction and dominant value reduction. Instead of triplicating the whole logic as in TMR, only those nodes with high soft error susceptibility are triplicated; the rest of the nodes are clustered and shared among the triplicated logic. In [18] and [19], sensitive gates are duplicated and their outputs are connected together. Physically placing the two gates with a sufficient distance reduces the probability of having the two gates hit by a particle strike simultaneously and, therefore, reduces the soft error rate (SER). Another technique based on TMR maintains a history index of correct computation module to select the correct result [20]. Teifel [21] proposed a double/dual modular redundancy (DMR) scheme that utilizes voting and self-voting circuits to mitigate the effects of SETs in digital integrated circuits. The Bayesian detection technique from communication theory has been applied to the voter in DMR, called soft NMR [22]. In most cases, it is able to identify the correct output even if all redundant modules are in error, but at the expense of very high area overhead cost of the voters.

Another class of techniques enhances fault tolerance by increasing soft error masking based on modifying the structure of the circuit by addition and/or removal of redundant wires or by resynthesizing parts of the circuit. In [23], SER is reduced based on redundancy addition and removal of wires. In [24], redundant wires are added based on the existing implications between a pair of nodes in the circuit. In [25], two-level circuits are synthesized by assigning do not care conditions to improve input error resilience, which minimizes the propagation of fault effects. In [26], an algorithm is proposed to synthesize two-level circuits to maximize logical masking utilizing input pattern probabilities.

Soft error protection of combinational logic can also be achieved by adding redundancy at the transistor level. Nicolaidis [27] proposed a scheme where a circuit is duplicated containing all but the last stage gate where the last stage gate is implemented as a state preserving gate. This last stage gate is the NOT, NAND, or NOR gate with each transistor duplicated and connected in series to preserve the fault-free state that the output had before the transient fault occurred. More recently, El-Maleh *et al.* [28] proposed a technique to mask defects in combinational circuits by quadrupling every transistor in a circuit, making the area overhead four times the original circuit. A quadded transistor guarantees the tolerance of all single-transistor (1T) defects and many multiple defects. In the quadded-transistor structure, each transistor A is replaced by a structure that implements the logic function $(A + A)(A + A)$. Techniques that combine the gate-level redundancy and quadded-transistor redundancy are proposed in [29] and [30].

Selective hardening techniques, as the name suggests, protect only the most sensitive gates of the circuit [31]. Zhou and Mohanram [32] proposed a gate sizing method that protects the sensitive gates by symmetrically sizing their nMOS and pMOS transistors. Lazzari *et al.* [33] proposed an asymmetric transistor sizing technique, i.e., nMOS and pMOS transistors are sized independently of each other for the most sensitive gates of the circuit, but they considered that incident particles strike only the transistors connected to the output of a gate. Sizing parallel transistors according to the sensitivity of their corresponding series transistors can significantly improve the fault tolerance of combinational circuits [34], [35]. Variable sizing among all transistors in a gate is a viable option if particle strikes of varying charge are considered. To further improve the fault tolerance, more up sizing quota is given to the most sensitive gates [36].

III. Effect of Energetic Particle Strike

When an energetic particle strikes a semiconductor, it ionizes the region around it, resulting in the generation of electron–hole pairs. The charge due to the particle strike is then transported by drift and diffusion, resulting in the establishment of transient electric field, i.e., SET. The change in voltage observed at the output due to SET depends on the energy and angle of incidence of energetic particle. Source and drain regions are the most sensitive nodes to such events due to the large field around the junction regions, which sweeps in the generated electron–holes and result in large currents. If the energy of a striking particle is high enough, it will flip the output of a gate resulting in an SET [3], [5].

To explain the STR principle, we first consider the effect of an energetic particle striking a CMOS inverter. When the inverter input is LOW and the energetic particle strikes the drain of an nMOS transistor, the output voltage is temporarily lowered. Whereas, when the inverter input is HIGH and the energetic particle strikes the drain of a pMOS transistor, the output voltage is temporarily raised. In both the cases, the output logic value of the inverter can be changed to a wrong value if enough charge is collected. This is shown in Fig. 1, using 130-nm predictive technology model [37]. Fig. 1(a) shows the fault injection mechanism employed in this paper. The output load is assumed to be equal to an inverter

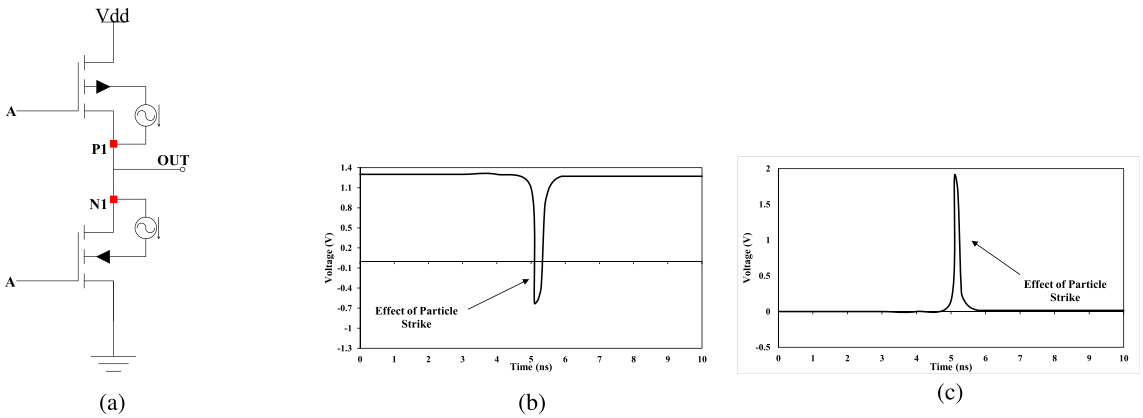


Fig. 1. Effect of energetic particle strike on CMOS inverter at $t = 5$ ns. (a) Particle strike model. (b) Effect of particle strike at nMOS drain. (c) Effect of particle strike at pMOS drain.

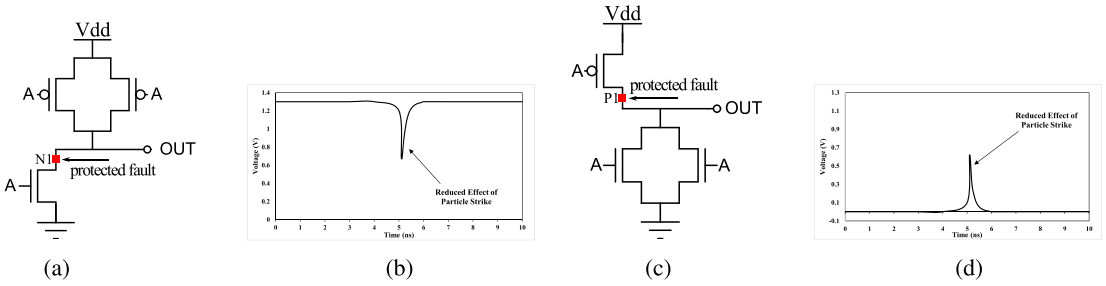


Fig. 2. Proposed protection schemes and their effect. (a) Particle hit at nMOS drain, OUT = HIGH. (b) Reduced effect of particle strike at nMOS drain. (c) Particle hit at pMOS drain, OUT = LOW. (d) Reduced effect of particle strike at pMOS drain.

load. The soft error is modeled by injecting a current I of charge Q at the drain of a transistor. The direction of injected current is from drain-to-body (bulk) in the nMOS transistor and from body (bulk)-to-drain in the pMOS transistor. The double exponential current pulse I is used to model the charge deposited due to a particle strike at the drain of nMOS or pMOS transistor [38], [39] and is depicted as

$$I(t) = \frac{Q}{(\tau_f - \tau_r)}\left(e^{-\frac{t}{\tau_f}} - e^{-\frac{t}{\tau_r}}\right) \tag{1}$$

where Q is the charge deposited by a particle strike, τ_f denotes the falling time of the pulse, and τ_r denotes the rising time of injected current pulse and varies for each process technology. The value of τ_f is greater than τ_r. The supply rail Vdd is connected to 1.3 V. We will be taking 130-nm technology as our case study in this paper; however, the technique is general and applicable to any process technology.

Fig. 1(b) shows the effect of a particle strike on the drain of an nMOS transistor when the true output of an inverter is HIGH. The particle strike at N1 will cause a sudden drop in the output voltage (approximately -0.7 V) of an inverter. This type of soft error will be modeled as a stuck-at-0 (sa0) fault at the output of the gate. To protect from this fault, the pMOS transistors of an inverter must be scaled enough, so that the output voltage becomes $>$Vdd/2. Fig. 1(c) shows the effect of a particle strike on the drain of a pMOS transistor when the true output of an inverter is LOW. The particle strike at P1 will cause a sudden rise in the output voltage ($\approx$1.9 V) of an inverter. This type of soft error will be modeled as a stuck-at-1 (sa1) fault at the output of the gate. To protect from this fault, the nMOS transistor of an inverter must be scaled enough, so that the output voltage becomes $<$Vdd/2.

Now, consider the transistor arrangement shown in Fig. 2(a) where duplicate pMOS transistors are connected in parallel. The width of the redundant transistors must also be increased to allow dissipation (sinking) of the deposited charge as quickly as it is deposited, so that the transient does not achieve sufficient magnitude and duration to propagate to the output. If the output is currently high and an energetic particle hits the drain N1 of the nMOS transistor (with the same current source used in the simulations shown in Fig. 1), this should result in a lowered voltage observed at the output. But, due to the employed transistor configuration, the net negative voltage effect will be compensated, as evident from Fig. 2(b), resulting in a spike that has lesser magnitude as compared with the one shown in Fig. 1(b). The spike magnitude is reduced due to increased output capacitance and reduced resistance between the Vdd and the output.

Consider another arrangement of transistors in Fig. 2(c) where redundant nMOS transistors are connected in parallel. If the output is low and the incident energetic particle strikes the drain P1 of pMOS transistor, then the raised voltage effect at the output shown in Fig. 1(c) will be reduced, as shown in Fig. 2(d). This reduction in the spike magnitude is due to the same reasons mentioned for the nMOS transistor.

Similarly, to protect from both sa0 and sa1 faults, the transistor structures in Fig. 2(a) and (b) can be combined to fully protect the NOT gate. A fully protected NOT gate offers

the best hardening by design, but at the cost of higher area overhead and power. It must be noted that the optimal size of the transistor for SEU immunity depends on the charge Q of the incident energetic particle.

Due to aggressive nodes and voltage scaling, the effect of transient fault is no more constrained to a node where the incident particle strikes. This could result in the possibility of deposited charge being simultaneously shared by multiple circuit nodes in the circuit [40]–[42], leading to the single event multiple upsets, also referred to as multiple-bit upsets [3]. Considering the inverter example in Fig. 1, if two particles strike at the drain of nMOS and pMOS transistors simultaneously, then the charge collection at the nMOS and pMOS transistors will offset each other, resulting in an insignificant change in voltage at the output. Therefore, by the duplication of transistors, it is intended to increase the probability of multiple fault hits at the same gate, so that the victim transistors could cancel the effect of each other. For that matter, LEAP [43] placement technique can be utilized. This scheme places the drain contact nodes of nMOS and pMOS transistors in an interleaved fashion, so that multiple drain contact nodes can act together to fully or partially suppress the SETs. Another advantage of using parallel duplicate transistors is the defect tolerance of transistor stuck-open faults for protected transistors.

IV. Proposed Algorithm

In this section, the proposed STR algorithm is presented. The algorithm protects sensitive transistors whose probability of failure (POF) is relatively high. The proposed algorithm can be utilized in two capacities: 1) apply protection until the POF of circuit reaches a certain threshold and 2) apply protection until certain area overhead constraint is met. We will first discuss different relations that realize the circuit POF. These relations are then used in the proposed algorithm.

A. Circuit Probability of Failure

Let us first define the POF of a transistor. In all discussions, subscripts i and j refer to gate i and transistor j, respectively. The POF_{ij} of the jth transistor of gate i is defined as the probability of circuit failure due to a fault hitting the transistor. It is computed using the following relation:

$$\mathrm{POF}_{ij} = P_{\mathrm{DET}_{ij}} \times P_{\mathrm{HIT}_{ij}} \tag{2}$$

where $P_{\mathrm{DET}_{ij}}$ is the probability of detecting a fault hitting transistor j of gate i at a primary output, and $P_{\mathrm{HIT}_{ij}}$ is the probability that transistor j of gate i is hit by a fault. The greater the transistor width/area is, the greater its hit probability is.

$P_{\mathrm{HIT}_{ij}}$ is computed separately for nMOS and pMOS transistors as they have different drain widths. Let NW_{ij} and PW_{ij} be the width of nMOS and pMOS transistors, respectively, and Area be the total circuit area; then, the probability of a transistor j of gate i to be hit by a fault, $P_{\mathrm{HIT}_{ij}}$, is computed using the following relation:

$$P_{\mathrm{HIT}_{ij}} = \frac{W_{ij}}{\mathrm{Area}} \quad W_{ij} \in \{\mathrm{NW}_{ij}, \mathrm{PW}_{ij}\}. \tag{3}$$

$P_{\mathrm{DET}_{ij}}$, as defined before, depends on two factors: 1) probability of input patterns for which a fault that hits the transistor is propagated to the output of a gate, i.e., controllability conditions to excite the fault and 2) stuck-at fault observability probability of the gate at one of the primary outputs of a circuit, i.e., observability probability. $P_{\mathrm{DET}_{ij}}$ is computed using the following relation:

$$P_{\mathrm{DET}_{ij}} = P_{\mathrm{Excitation}_{ij}} \times P_{\mathrm{Propagation}_{ij}} \tag{4}$$

where $P_{\mathrm{Excitation}_{ij}}$ denotes the probability that the fault is excited at gate i output due to a fault hit at transistor j. $P_{\mathrm{Propagation}_{ij}}$ denotes the probability that an error that is excited at the gate's output is observable at one of the primary outputs. Let $\mathbf{S}$ be a set of patterns for which an error that strikes transistor j is propagated to the output of gate i; then, $P_{\mathrm{Excitation}_{ij}}$ is computed as

$$P_{\mathrm{Excitation}_{ij}} = \sum_{k=1}^{|\mathbf{S}|} \mathrm{Prob.}\ \mathbf{S}_k \tag{5}$$

where Prob. $\mathbf{S}_k$ denotes the probability of occurrence of the kth input pattern. SPICE tool is used to find the input patterns for which a transistor fault is excited and observed at the gate output.

Similarly, $P_{\mathrm{Propagation}_{ij}}$ can be computed using the following relation:

$$P_{\mathrm{Propagation}_{ij}} = \frac{\mathrm{stuck-at-detection-prob_i}}{\mathrm{PC}_i} \tag{6}$$

where stuck-at-detection-prob$_i$ defines stuck-at fault detection probability of gate i and PC_i is the controllability probability to produce logic value opposite to the fault effect at the gate output. The fault simulator tool Hope [44] is used to compute the stuck-at fault detection probability and PC_i of a gate i.

Finally, the circuit POF $\mathrm{POF}_{\mathbf{C}}$ for a single fault is simply the summation of POFs of all transistors n over all gates m of a circuit

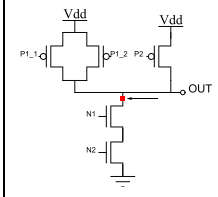
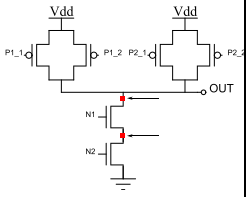
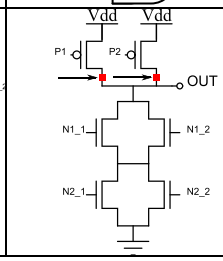
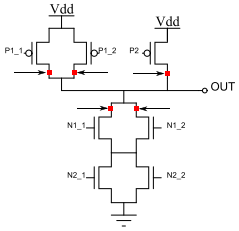
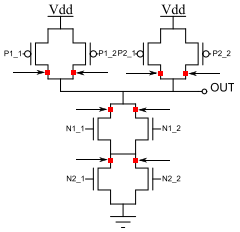
$$\mathrm{POF}_{\mathbf{C}} = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathrm{POF}_{ij}. \tag{7}$$

B. Selective-Transistor-Redundancy-Based Design

The selective redundancy technique is applied to protect the transistors of a circuit that have relatively high POF_{ij}. Sensitive transistors that have relatively high POF are identified based on fault simulation of random input patterns. Different arrangements of nMOS and pMOS transistors are proposed for each gate for various transistor protection scenarios.

Algorithm 1 highlights the steps of the proposed method. Initially, the POF of circuit under test is computed using (7) by first computing the POF of each transistor using (2). The proposed algorithm applies transistor protection until the circuit POF reaches a predefined protection threshold, or a certain area overhead constraint is met. Each time, the algorithm selects a transistor with the highest POF. The effect of a transient fault on the selected nMOS (pMOS) transistor is suppressed or reduced by duplicating and scaling the widths of a subset of transistors necessary for providing the protection. For example, in a two-input NAND gate,

TABLE I
PROPOSED CMOS IMPLEMENTATIONS OF TWO-INPUT NAND GATE

Gate	NAND21	NAND22	NAND23	NAND24	NAND25
CMOS [1]					

[1] Arrows indicate fault hit at the transistor that is protected.

Algorithm 1 STR Algorithm

Require: Gate level circuit, Th or $OverHead$
Th : Required circuit reliability in %
$OverHead$: Required area overhead in %
POF_{ij} : Circuit POF due to fault hit at j^{th} transistor of Gate i
$\mathbf{POF_C}$: Circuit probability of failure

Compute random pattern fault detection probability of each gate g_i using fault simulator
For all transistors compute POF_{ij} using Eqn. (2)
Compute $\mathbf{POF_C}$ using Eqn. (7)
TargetArea = CircuitArea + (CircuitArea $\times$ OverHead)
while $((\mathbf{POF_C} \geq (1-Th))or(CircuiArea < TargetArea))$ **do**
    Pick a transistor $trans_{ij}$ with the highest POF_{ij}
    Protect $trans_{ij}$
    Update CircuitArea
    Update POF_{ij} of transistors using Eqn. (2)
    Update $\mathbf{POF_C}$ using Eqn. (7)
end while

protecting an nMOS transistor requires duplicating and scaling its corresponding pMOS transistor connected to the same input. However, protecting a pMOS transistor requires duplicating and scaling both of the nMOS transistors in the gate. Once a transistor is protected, the POF of all transistors in the circuit is updated. Protecting a transistor in a gate g_i affects the selection/hit probability of all transistors in the circuit. Therefore, after protecting a transistor in a gate, the POF of the selected transistor is reduced significantly, while the POF of the remaining transistors may increase or reduce slightly. The circuit area, POF of all transistors, and $\text{POF}_\mathbf{C}$ are updated after each transistor protection is applied. The transistor with maximum POF_{ij} is selected for protection in the next iteration. The process is repeated until the desired protection threshold is reached or the maximum area overhead constraint is met.

The protection threshold Th takes the value between [0%, 100%] and represents the reliability of the circuit required to be achieved. For example, a protection threshold of 99% implies applying the protection until the POF of circuit is less than or equal to $(1 - 99\%) = 0.01$. Increasing Th will result in more transistors being protected and vice versa.

C. Redundancy Models

In light of Algorithm 1, let us now explain the proposed CMOS implementations of a two-input NAND gate in Table I. In this paper, a library consisting of two-, three-, and four-input NAND/NOR gates and an inverter is considered. For brevity, the case of a two-input NAND gate will be discussed.

The proposed CMOS implementations of a two-input NAND gate are shown in Table I. The first row shows the names of different implementations of a two-input NAND gate, while the second row shows their corresponding implementations at the transistor level. The first numeric value 2 in the gate name (e.g., NAND21) denotes the number of inputs of a gate and the second numeric value, which ranges from 1 to 5, is used to select the proper transistor-level implementation of a two-input NAND gate.

In CMOS implementation of NAND21, pMOS transistor P1 is duplicated, scaled, and connected in parallel to protect a fault that hits the drain of nMOS transistor N1. Similarly, to protect nMOS transistors N1 and N2, pMOS transistors P1 and P2 are duplicated and scaled to protect from a fault that can occur at the drain of any of the nMOS transistors. Hence, the protection type for that gate will be NAND22.

To protect from faults hitting the drain of pMOS transistors, all the nMOS transistors are required to be scaled and duplicated. This is due to the fact that pMOS transistors P1 and P2 are in parallel, which makes them equally sensitive to a fault. This implementation is called NAND23. NAND24 provides protection from faults that can occur at any of the pMOS transistors or at the nMOS transistor N1. Finally, to fully protect the two-input NAND gate, all the transistors are duplicated along with their widths increased. This type of protection is called NAND25. So, for a two-input NAND gate, there are five distinct redundancy models.

For a two-input NOR gate, similar arrangements can be used to create its redundancy model. For three- and four-input NAND/NOR gates, we created seven and nine redundancy models, respectively. The necessity to create a variety of redundancy models for every possible scenario is to achieve as much area savings as possible. The number of redundancy models is technology-independent and allows the proposed algorithm to improve the reliability of circuit by applying fine grained protection (protecting one transistor at a time) instead of protecting the whole gate at once as has been proposed by other techniques [32], [33]. It will be shown that due to this fine granularity of protection, the area overhead can be significantly reduced.

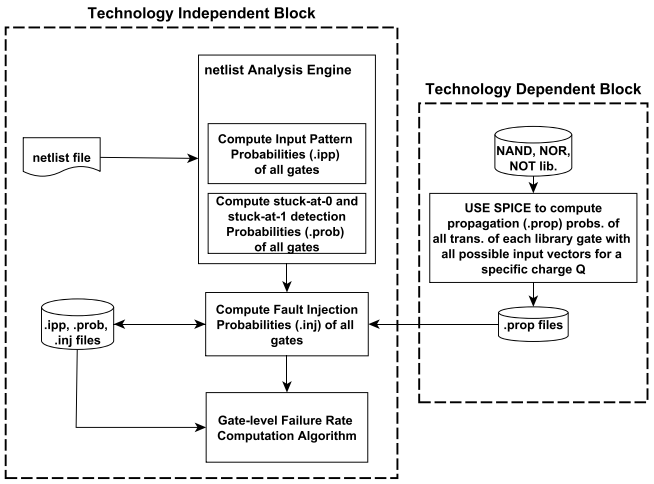


Fig. 3. Reliability evaluation framework.

V. Reliability Evaluation

A novel method to compute the reliability of a circuit at the gate level is proposed in this section. The proposed technique bridges the gap between circuit-level simulations performed at the transistor level using SPICE and gate-level simulations, which could be done using any gate-level simulator. In this realm, we propose the probability of fault injection, which quantifies the probability with which a fault must be injected at the gate level, so that SPICE-level and gate-level simulation results are highly matched.

The reliability evaluation framework, shown in Fig. 3, consists of two major blocks: 1) technology-independent block and 2) technology-dependent block. The purpose of the technology-independent block is to analyze a given benchmark circuit to compute three important parameters for all gates: 1) input pattern probability (`.ipp`); 2) stuck-at detection probability (`.prob`); and 3) fault injection probability (`.inj`). The input patterns observable at the input of each gate along with their probability of occurrence and stuck-at fault detection probabilities are computed by performing the simulation of random test vectors using the parallel fault simulator Hope [44]. The fault injection probability denotes the probability with which a fault must be injected at the gate level as a stuck-at fault. All of these parameters are saved in a database for later usage.

The technology-dependent part mainly consists of the library gates comprising NAND/NOR gates with varying input configurations and an inverter. The purpose of this block is to observe the behavior of different process technologies, e.g., 130, 45, 32 nm, and so on, against a specific charge value. This block computes the effect of an induced current of charge (Q) for every transistor of the gate in the library. The input patterns that result in a gate value flip when a transistor is hit are then saved in the propagation (`.prop`) file. In fact, we can compute and save the behavior of different technologies against different charge (Q) values, and this has to be done only once.

Now, the fault injection probability of a gate in a circuit can be computed for any process technology. It must be noted that the initial analysis of a circuit has to be done only once. Sections V-A–V-C contain the detailed elaboration of the reliability evaluation framework.

A. Probability of Fault Injection

The fault injection probabilities of a gate depend on the conditional fault excitation probability (CFEP_{ij}) and probability of hit/selection. A general relation to compute CFEP_{ij} of the jth transistor of a gate i can be derived as follows. Let $\mathbf{S}$ be a set of patterns for which an error is excited to the output of a gate and PC_i be the controllability probability to produce a logic value opposite to the fault effect at the gate output. Then, CFEP_{ij} can be defined as

$$\text{CFEP}_{ij} = \frac{\sum_{k=1}^{|\mathbf{S}|} \text{Prob. } \mathbf{S_k}}{\text{PC}_i} = \frac{P_{\text{Excitation}_{ij}}}{\text{PC}_i}. \tag{8}$$

CFEP_{ij} of any MOS transistor depends on the process technology and the charge of the incident particle. Therefore, in order to get the exact CFEP_{ij} probability for each MOS transistor, transistor-level simulations are performed using SPICE.

Now, the sa0 fault injection probability of gate G_i is computed using the following equation:

$$G_i \text{ sa0 inj. Prob } = \sum_{j=1}^{n} \left(\text{CFEP}_{N_{ij}} \times \frac{\text{NW}_{ij}}{\sum_{k=1}^{n} \text{NW}_{ik}} \right) \tag{9}$$

where n is the total number of nMOS transistors in gate G_i, NW_{ij} is the width of the drain of the jth nMOS transistor, and $\text{CFEP}_{N_{ij}}$ is the CFEP due to a fault hit at the jth nMOS transistor of gate i.

Similarly, the sa1 fault injection probability of gate G_i is computed as follows:

$$G_i \text{ sa1 in}j\text{. Prob } = \sum_{j=1}^{p} \left(\text{CFEP}_{P_{ij}} \times \frac{\text{PW}_{ij}}{\sum_{k=1}^{p} \text{PW}_{ik}} \right) \tag{10}$$

where p is the total number of pMOS transistors in gate G_i, PW_{ij} is the width of the drain of the jth pMOS transistor, and $\text{CFEP}_{P_{ij}}$ is the CFEP due to a fault hit at the jth pMOS transistor of gate i.

B. Fault Injection Mechanisms

Two fault injection mechanisms are applied in this paper. The first method performs fault injection at the transistor level and measures the magnitude of voltage V_{out} at the output. The second method deals with injecting the fault at the gate level by injecting the sa0 or sa1 fault at the gate output.

1) Transistor Level: The current I of charge Q is injected at the drain of a transistor. The magnitude and the pulsewidth of injected current are modeled using (1). Algorithm 2 highlights the steps of failure rate/reliability computation at the transistor level. In this algorithm, a set of **m** transistors are selected for fault injection using roulette wheel (RW) algorithm [45]. The RW algorithm selects the transistors that have higher area with high probability. For each random input vector, the outputs are saved before and after the fault injection and are then compared to check for correctness. The failure rate and the reliability of circuit are then computed after SIM simulations are performed.

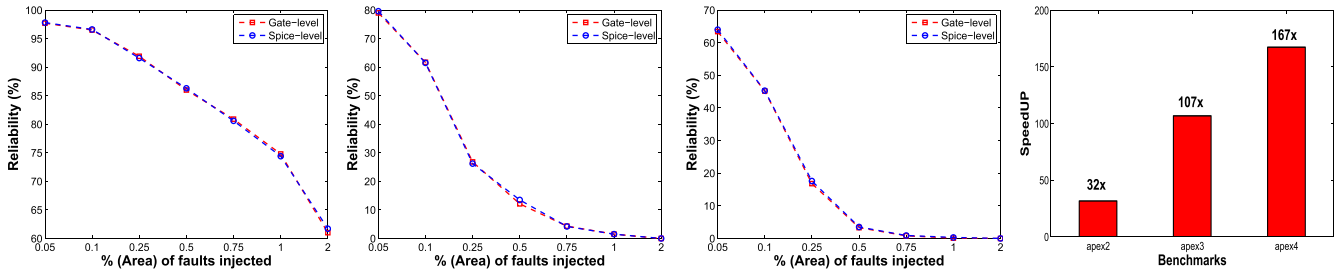


Fig. 4. SPICE and gate-level simulation accuracy comparisons and speedup. (a) $apex2$. (b) $apex3$. (c) $apex4$. (d) Speedup.

Algorithm 2 Transistor-Level Failure Rate Computation

Require: Transistor-level netlist
SIM : Simulation Count
F_m : Failure rate of the circuit with m faults
Rel_m : Reliability (%) of the circuit with m faults
K : Failure Count
R : Output of circuit with no fault injection
R_f : Output of circuit after fault injection
RW : Roulette Wheel algorithm
for $(i = 1 \rightarrow SIM)$ **do**
  $K \leftarrow 0$
  Generate a random test vector $\mathbf{V}$
  Apply $\mathbf{V}$ to the circuit
  Simulate the circuit and save the output in R
  RW($\mathbf{m}$) ▷ Select $\mathbf{m}$ transistors using Roulette Wheel Algorithm
  Inject faults in selected $\mathbf{m}$ transistors
  Apply $\mathbf{V}$ to the circuit with faults injected
  Simulate and save the output in R_f
  if $(R \neq R_f)$ **then** increment K
end for
$F_m = \left(\frac{K}{SIM}\right)$
$Rel_m(\%) = (1 - F_m) \times 100$

Algorithm 3 Gate-Level Failure Rate Computation

Require: Gate level netlist
SIM : Simulation Count
$sa0_{G_i}$: Stuck-at-0 injection probability of Gate G_i
$sa1_{G_i}$: Stuck-at-1 injection probability of Gate G_i
R : Output of circuit with no fault injection
R_f : Output of circuit after fault injection
RW : Roulette Wheel algorithm
F_m : Failure rate of circuit with m faults
Rel_m : Reliability (%) of circuit with m faults
K : Failure Count
$rand1(\cdot)$: Uniformly distributed random number $\sim$ (0,1)
$rand2(\cdot)$: Uniformly distributed random number $\sim$ (0,1)

for $(i = 1 \rightarrow SIM)$ **do**
  $K \leftarrow 0$
  Generate a random test vector $\mathbf{V}$
  Apply $\mathbf{V}$ to the circuit
  Simulate the circuit and save the output in R
  G$\leftarrow$RW($\mathbf{m}$) ▷ Select $\mathbf{m}$ gates
  for $(j = 1 \rightarrow G)$ **do** ▷ Iterate through G gates
    if $((sa0_{G_j} + sa1_{G_j}) == 0)$ **then**
      Don't inject any fault ▷ Gates protected
    else if $\left(rand1(\cdot) \leq \frac{sa0_{G_j}}{sa0_{G_j} + sa1_{G_j}}\right)$ **then**
      if $(rand2(\cdot) \leq sa0_{G_j})$ **then**
        Inject sa0 fault at the gate G_j output
      end if
    else
      if $(rand2(\cdot) \leq sa1_{G_j})$ **then**
        Inject sa1 fault at the gate G_j output
      end if
    end if
  end for
  Apply $\mathbf{V}$ to the circuit with faults injected
  Simulate and save the output in R_f
  if $(R \neq R_f)$ **then** increment K
end for
$F_m = \frac{K}{SIM}$
$Rel_m(\%) = (1 - F_m) \times 100$

2) Gate Level: Faults injected at the gate level assume a stuck-at fault model. When a fault is injected at a gate output, it can be either the sa1 fault (i.e., connected to Vdd) or the sa0 fault (i.e., connected to ground). Algorithm 3 is used to compute the circuit failure rate/reliability at the gate level. To inject $\mathbf{m}$ faults in a circuit, $\mathbf{m}$ gates are selected randomly using an RW algorithm. For each gate selected for fault injection, the following is performed. First, if both the sa0 and sa1 fault inject probabilities are 0, then no fault will be injected as the gate will be fully protected. Otherwise, a selection is made between injecting the sa0 fault or the sa1 fault according to the ratio of their fault injection probabilities. The selected fault will be injected based on its fault injection probability.

C. Transistor-Level Versus Gate-Level Simulations

To illustrate the accuracy of gate-level simulations, a comparison between transistor-level and gate-level simulations is shown in this section for few benchmark circuits. The transistor-level simulations are performed using SPICE. Fig. 4 shows the close match between the reliability obtained by SPICE and the gate-level simulations for the three compared benchmark circuits: $apex2$, $apex3$, and $apex4$. The circuits reliabilities are evaluated after performing 1000 iterations for each fault injection case. In Fig. 4(c), 0.10% on the x-axis corresponds to $0.1\% \times 12190 \approx 12$ faults injected in the $apex4$ circuit with a reliability of ≈45% achieved after performing 1000 iterations for both SPICE and gate-level simulations.

Time is another factor that must be considered while evaluating a circuit for reliability. The time taken by SPICE

TABLE II
RELIABILITY OF ORIGINAL BENCHMARK CIRCUITS

Circuit (Inputs, Outputs)	True Phase Synthesis				Majority Phase Synthesis				ΔArea[2]
	Area1 (μ)[1]	1 Fault	2 Faults	5 Faults	Original Area	1 Fault	2 Faults	5 Faults	
alu4 (14, 8)	1831.44	96.78%	92.86%	83.44%	1429.74	97.89%	95.86%	87.44%	–21.93%
apex1 (45, 45)	4282.2	96.80%	92.72%	82.64%	4602.00	96.72%	94.20%	86.40%	7.47%
apex2 (39, 3)	804.18	99.10%	97.38%	94.54%	609.96	99.20%	98.04%	95.42%	–24.15%
apex3 (54, 50)	3091.92	96.00%	93.28%	84.12%	3025.62	96.88%	94.76%	85.66%	–2.14%
apex4 (9, 19)	4754.1	95.78%	92.32%	82.30%	4575.48	96.20%	92.74%	84.16%	–3.76%
b12 (15, 9)	130.26	88.36%	78.18%	54.28%	121.68	89.10%	78.22%	55.30%	–6.59%
clip (9, 5)	372.84	93.24%	86.40%	71.44%	372.84	93.24%	86.40%	71.44%	0 %
cordic (23, 2)	238.68	98.02%	96.24%	90.54%	241.02	98.10%	96.28%	92.14%	0.98%
ex5 (8, 63)	948.48	93.28%	88.64%	70.70%	977.34	93.50%	88.26%	71.34%	3.04%
misex1 (8, 7)	139.62	83.40%	68.92%	41.74%	152.88	84.34%	71.70%	48.44%	9.50%
misex2 (25, 18)	225.42	93.36%	88.02%	71.80%	230.88	93.20%	88.32%	69.48%	2.42%
misex3 (14, 14)	2410.98	97.48%	95.56%	88.58%	1886.82	97.64%	95.40%	88.90%	–21.74%
rd84 (8, 4)	368.94	91.36%	84.94%	65.18%	496.08	93.38%	87.22%	71.40%	34.46%
seq (41, 35)	4693.26	98.92%	98.26%	95.96%	4970.94	99.05%	98.22%	95.74%	5.92%
squar5 (5, 8)	111.54	82.50%	70.04%	41.04%	101.40	83.44%	69.46%	42.68%	–9.09%
table3 (14, 14)	3073.2	98.12%	95.52%	88.54%	3475.68	98.68%	97.78%	94.28%	13.10%
table5 (17, 15)	3230.76	97.48%	96.00%	88.52%	3535.74	98.70%	97.72%	94.52%	9.44%
z5xp1 (7, 10)	248.82	85.96%	75.38%	49.36%	251.16	87.96%	75.38%	52.36%	0.94%
Avg.		**93.66%**	**88.37%**	**74.71%**		**94.29%**	**89.22%**	**77.06%**	**-0.12%**

[1] Summation of nmos and pmos drain widths
[2] $\Delta\text{Area} = \left(\frac{Original\ Area - Area1}{Area1}\right) \times 100$

simulations becomes exorbitantly high as the number of transistors is increased. The $apex4$ benchmark took around four days for SPICE simulations, while it took 30 min of gate-level simulations, hence achieving a speedup of $\approx 167\times$. It can be observed from Fig. 4(d) that as the number of transistors is increased, the speedup achieved by gate-level simulations also increases significantly.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the impact of the proposed algorithm on the area and reliability of LGSynth'91 benchmarks [46], which consist of circuits with varying complexity in terms of area, number of inputs, and outputs. Sensitive nodes (transistors/gates) in a circuit are identified based on the fault simulation of random input vectors using the parallel fault simulator Hope [44]. The input patterns are applied until stuck-at fault coverage of 95% is achieved. It was found that 1 million random input patterns achieved more than 95% stuck-at fault coverage for all benchmarks in this paper.

Whenever cell hardening against soft errors is considered, the first step is to select a range of particles energy against which the tolerance is sought. In this paper, it is assumed that energy of the incident particle will always result in the maximum deposition of charge. For that matter and to compare with other sizing techniques, the values of $Q = 0.3$ pC, $\tau_f = 0.2$ ns, and $\tau_r = 0.05$ ns are used for all the simulations in this paper. The value of charge $Q = 0.3$ pC is the maximum charge that could be collected by the 130-nm process technology [32]. The simulations are performed for varying protection thresholds to find the best tradeoff between area and reliability for each circuit. The number of faults injected in a protected circuit is prorated according to its area overhead. Algorithm 3 is used to compute the reliability of each circuit. The number of simulations count SIM is 5000 iterations for each fault injection scenario.

The LGSynth'91 benchmark circuits used in this paper are represented in two-level pla formats; therefore, they are synthesized with single output optimization using Espresso [47] tool and then mapped to 130-nm technology using SIS [48] to get the proper gate-level representation of the circuit. The library used for mapping consists of an inverter and two-, three-, and four-input NAND and NOR gates. The parameter phase in the logic synthesis process defines whether the output function should be synthesized as an ON-set (phase = 1) or an OFF-set (phase = 0). By default, each output is synthesized as an ON-set by the Espresso tool. We synthesize each output by synthesizing the phase with higher probability, i.e., if the output probability of 1 is higher than the probability of 0, then the value of (phase = 1) is set; otherwise, (phase = 0) is set. This produces circuits with higher reliability, as shown in Table II.

In Table II, the first column denotes the circuit names along with the number of inputs and outputs in each circuit. The second and the third major columns report the reliability of circuits using default synthesis settings and the proposed majority phase synthesis mechanism. The reliability of circuits is evaluated against one, two, and five faults. The Area of a benchmark is computed by summing the drain area of all the nMOS and pMOS transistors.

Table II highlights the increase in reliability when the circuits are synthesized with respect to the majority phase. This is due to the increase in fault masking that may occur as if the final gate is an OR gate and has a value of 1, any fault propagating through any of the other inputs will be masked. If the OR gate has at least two inputs with a value of logic 1, all the faults propagating through any of the inputs will be masked. Thus, synthesizing the circuit to maximize the probability of getting a logic 1 at the final gate by synthesizing the majority phase will maximize the probability of fault masking and, hence, improve reliability. Therefore, circuits

TABLE III

RELIABILITY OF CIRCUITS BASED ON THE PROPOSED STR TECHNIQUE WITH VARYING PROTECTION THRESHOLDS AGAINST A SINGLE FAULT

Circuit	95%		98%		99%	
	OH[1]	Rel	OH	Rel	OH	Rel
alu4	0%	97.89%	20.04%	98.44%	51.84%	99.02%
apex1	0%	96.72%	12.64%	98.10%	48.81%	99.10%
apex2	0%	99.20%	0%	99.20%	0%	99.20%
apex3	0%	96.88%	13.93%	98.05%	51.57%	99.08%
apex4	0%	96.20%	25.88%	98.30%	71.88%	99.02%
b12	86.50%	95.01%	149.02%	98.03%	197.69%	99.05%
clip	14.64%	95.00%	68.61%	98.11%	119.87%	99.08%
cordic	0%	98.10%	0%	98.10%	19.53%	99.00%
ex5	19.53%	95.62%	81.73%	98.12%	139.89%	99.02%
misex1	85.68%	95.07%	175.48%	98.01%	249.35%	99.02%
misex2	14.12%	95.24%	50.29%	98.36%	92.48%	99.20%
misex3	0%	97.64%	3.33%	98.10%	34.27%	99.18%
rd84	14.16%	95.10%	56.49%	98.02%	101.21%	99.24%
seq	0%	99.05%	0%	99.05%	0%	99.05%
squar5	107.79%	95.10%	206.15%	98.06%	286.51%	99.00%
table3	0%	98.68%	0%	98.68%	0.34%	99.24%
table5	0%	98.70%	0%	98.70%	1.70%	99.32%
z5xp1	68.22%	95.06%	143.69%	98.11%	202.88%	99.06%
Avg.	**22.81%**	**96.68%**	**55.96%**	**98.31%**	**92.77%**	**99.11%**

[1] Area overhead (OH) $= \left(\frac{Area\ After\ Protection}{Original\ Area} - 1\right) \times 100$

synthesized with the majority phase are the baseline circuits used in all our simulations in this paper. It can be observed that for a few benchmarks, reliability is above 90% for all fault injection scenarios. These benchmarks promise great reliability improvement with slight area overhead.

Table III shows the results of applying Algorithm 1 on the benchmark circuits and highlights the reliability of circuits for varying protection thresholds. A protection threshold of 98% implies that the circuit POF must be less than or equal to $(1 - 98\%) = 0.02$. Therefore, the applied protection threshold highly correlates with the reliability achieved by the circuit for a single fault. In Table III, under the 99% column, the minimum area overhead required for the $ex5$ circuit to achieve a reliability greater than or equal to 99% against a single fault is $\approx$ 140%. For $alu4$, $apex1$, $apex2$, $apex3$, $apex4$, $cordic$, $misex3$, seq, $table3$, and $table5$ benchmarks under the 95% column in Table III, zero area overhead implies that these benchmarks achieve 95% reliability against single fault without any area overhead. Similarly, $apex2$, seq, $table3$, and $table5$ benchmarks also achieve 98% reliability against single fault without any protection/area overhead. Only the seq circuit achieves 99% reliability without any overhead. The average area overhead required by the proposed algorithm to achieve 95%, 98%, and 99% reliability is 22.81%, 55.96%, and 92.77%, respectively.

Next, we compare the proposed technique with the asymmetric transistor sizing technique of sensitive gates proposed in [33]. The technique asymmetrically sizes the transistors connected to the output of a gate, i.e., nMOS and pMOS networks are sized independently. We have implemented the technique proposed in [33] as follows. The sensitivity of a gate is measured by considering sa0 and sa1 fault detection probabilities independently. Gates are then sorted according to their detection probabilities. Algorithm 1 is then applied, but now the possible protections that can be applied to a gate are restricted to transistors connected to the output of a gate. For example, for a two-input NAND gate, possible protections are NAND21, NAND23, and NAND24 only. After each protection is applied to a gate, the $\text{POF}_{\mathbf{C}}$ of circuit is updated using (7). The process is repeated until the reliability/area overhead requirement is met or all possible protections are applied to all the gates.

The results of this technique are shown in Table IV. It can be observed that by selectively protecting the transistors connected to the output of a gate, benchmarks, such as $b12$, $clip$, $ex5$, $misex1$, $misex2$, $rd84$, $squar5$, and $z5xp1$ are unable to achieve 99% reliability against a single fault. Benchmarks $b12$, $misex1$, $squar5$, and $z5xp1$ are also unable to achieve 98% reliability. In addition to that, the area overhead becomes significantly higher in comparison to the proposed STR technique even if the required reliability measure of 99% is achieved against a single fault.

The technique in [32] protects all sensitive gates symmetrically, i.e., all transistors in a sensitive gate are protected and are equally scaled. We also compared with the technique similar to [32] based on fully protecting sensitive gates but with protecting transistors asymmetrically. Protecting transistors asymmetrically have an advantage over symmetric protection due to the difference in the characteristics of nMOS and pMOS transistors. The sensitivity of a gate is measured as the sum of sa0 and sa1 fault detection probabilities. Gates are then sorted according to their detection probabilities. Algorithm 1 is then applied by fully protecting the gate with the highest detection probability. For example, a two-input NAND gate will be implemented as NAND25 in Table I. After each protection is applied to a gate, the $\text{POF}_{\mathbf{C}}$ of circuit is updated using (7). The process is repeated until the reliability/area overhead requirement is met or all gates are fully protected.

Table IV also highlights the area overhead incurred by fully protecting sensitive gates asymmetrically against a single fault. It can be observed from Tables III and IV that the proposed technique offers less area overhead as compared with the asymmetric technique for all protection threshold scenarios. Also, under the 99% column header in Table III and under Asymmetric column header in Table IV, it is evident that the proposed technique achieves significant area savings for 13 out of 18 benchmark circuits with similar reliability measures.

The simulations are further extended to analyze circuit reliability against multiple faults. The number of faults injected is correlated with the area of a circuit. Table V shows the reliability achieved by prorating the one, two, and five faults for each circuit according to its area. For example, if the area overhead is 131%, then the actual area is increased by a factor of 2.31. So, one, two, and five faults in the original circuit will prorate to 2.31, 4.62, and 11.55 faults in the protected circuit. For each prorated fault, the circuit is simulated twice. For example, if the prorated faults to be injected are 4.62, then the circuit is simulated twice, once by injecting four faults and another by injecting five faults. The failure rate achieved by both fault injection scenarios is then computed based on a weighted average to compute the final failure rate/reliability. It is interesting to observe that with the prorated faults, the average reliability achieved by the proposed method with 99% protection is above 96% for one and two prorated faults.

TABLE IV

RELIABILITY OF CIRCUITS BASED ON LAZZARI *et al.* [33] AND ASYMMETRIC GATE SIZING TECHNIQUE AGAINST A SINGLE FAULT

Circuit	Lazzari [33]				Asymmetric			
	98%		99%		98%		99%	
	OH	Rel	OH	Rel	OH	Rel	OH	Rel
alu4	70.33%	98.01%	218.64%	99.01%	24.06%	98.01%	60.48%	99.18%
apex1	55.42%	98.01%	312.95%	99.00%	19.54%	98.20%	64.64%	99.28%
apex2	0%	99.20%	0%	99.20%	0%	99.20%	0%	99.20%
apex3	43.37%	98.00%	333.11%	99.01%	28.04%	98.02%	72.03%	99.18%
apex4	108.59%	98.00%	483.00%	99.34%	32.92%	98.05%	94.94%	99.12%
b12	358.85%	96.78%	358.84%	96.78%	173.50%	98.20%	264.83%	99.20%
clip	203.47%	98.00%	321.62%	98.47%	79.34%	98.11%	144.09%	99.19%
cordic	0%	98.10%	122.59%	99.00%	0%	98.10%	21.17%	99.06%
ex5	349.78%	98.00%	429.32%	98.22%	117.69%	98.19%	195.46%	99.10%
misex1	322.86%	97.06%	322.86%	97.06%	209.66%	98.13%	287.38%	99.10%
misex2	394.03%	98.00%	504.89%	98.36%	118.00%	98.06%	156.35%	99.11%
misex3	4.18%	98.02%	211.21%	99.10%	6.15%	98.10%	43.36%	99.20%
rd84	237.37%	98.07%	456.35%	98.64%	66.67%	98.11%	133.41%	99.02%
seq	0%	99.05%	0%	99.05%	0%	99.05%	0%	99.05%
squar5	321.85%	97.13%	321.85%	97.13%	231.08%	98.21%	327.28%	99.21%
table3	0%	98.68%	0.82%	99.01%	0%	98.68%	3.99%	99.24%
table5	0%	98.70%	1.30%	99.01%	0%	98.70%	4.41%	99.28%
z5xp1	323.83%	97.62%	323.83%	97.62%	156.79%	98.20%	235.18%	99.26%
Avg.	**155.22%**	**98.02%**	**262.40%**	**98.50%**	**70.19%**	**98.30%**	**117.17%**	**99.17%**

TABLE V

RELIABILITY OF CIRCUITS BASED ON THE PROPOSED STR TECHNIQUE AGAINST PRORATED FAULTS. (a) ONE PRORATED FAULT. (b) TWO PRORATED FAULTS. (c) FIVE PRORATED FAULTS

(a)

Circuit	95%	98%	99%
alu4	97.89%	98.11%	98.63%
apex1	96.72%	98.16%	98.66%
apex2	99.20%	99.20%	99.20%
apex3	96.88%	98.21%	98.43%
apex4	96.20%	97.85%	98.41%
b12	92.50%	95.74%	98.18%
clip	95.19%	97.39%	97.60%
cordic	98.10%	98.10%	98.98%
ex5	93.78%	96.25%	97.64%
misex1	92.62%	95.50%	97.15%
misex2	95.57%	97.33%	98.32%
misex3	97.64%	98.28%	98.61%
rd84	94.68%	97.96%	98.29%
seq	99.05%	99.05%	99.05%
squar5	91.50%	95.81%	97.10%
table3	98.68%	98.68%	99.20%
table5	98.70%	98.70%	99.08%
z5xp1	92.17%	96.22%	96.83%
Avg.	**95.95%**	**97.59%**	**98.30%**

(b)

Circuit	95%	98%	99%
alu4	95.86%	96.46%	97.69%
apex1	94.20%	96.12%	97.07%
apex2	98.04%	98.04%	98.04%
apex3	94.76%	95.67%	97.30%
apex4	92.74%	95.21%	96.71%
b12	82.69%	91.39%	95.16%
clip	94.00%	94.50%	95.86%
cordic	96.28%	96.28%	97.73%
ex5	89.15%	93.49%	96.11%
misex1	83.51%	90.87%	94.57%
misex2	90.75%	94.40%	96.75%
misex3	95.40%	95.86%	97.53%
rd84	90.70%	93.83%	95.95%
seq	98.22%	98.22%	98.22%
squar5	81.17%	89.63%	94.73%
table3	97.78%	97.78%	98.05%
table5	97.72%	97.72%	97.58%
z5xp1	84.41%	91.08%	95.01%
Avg.	**92.08%**	**94.81%**	**96.67%**

(c)

Circuit	95%	98%	99%
alu4	87.44%	90.26%	93.92%
apex1	86.40%	89.98%	93.47%
apex2	95.42%	95.42%	95.42%
apex3	85.66%	90.20%	93.52%
apex4	84.16%	88.12%	92.40%
b12	60.89%	79.48%	88.80%
clip	76.29%	85.16%	91.40%
cordic	92.14%	92.14%	95.25%
ex5	77.95%	83.77%	89.60%
misex1	62.12%	76.53%	86.88%
misex2	77.93%	88.25%	92.05%
misex3	88.90%	90.34%	94.37%
rd84	77.46%	86.07%	90.99%
seq	95.74%	95.74%	95.74%
squar5	58.77%	72.58%	84.07%
table3	94.28%	94.28%	95.15%
table5	94.52%	94.52%	95.99%
z5xp1	65.57%	80.45%	86.19%
Avg.	**81.20%**	**87.40%**	**91.96%**

The reliability measures achieved by the asymmetric sizing technique against the prorated faults are shown in Table VI. It can be observed that the average reliability achieved by the proposed scheme under all fault injection scenarios and for all protection thresholds is better/close to the asymmetric gate sizing technique.

To further illustrate the advantage of our proposed STR technique against techniques that fully protect sensitive gates, Table VII shows the percentage distribution of gates that have been protected with 1T protection, e.g., NAND21 from Table I, full protection (FP), e.g., NAND25 from Table I, and no protection (NP) for each circuit when Algorithm 1 is applied for target reliability of 98% and 99%. It is clear from Table VII that for some circuits, the percentage of protected gates without FP is significant. This percentage is even higher than the percentage of fully protected gates, such as $apex2$, $apex3$, $cordic$, $misex2$, $table3$, and $table5$.

Table VIII shows the reliability achieved by TMR algorithm. The TMR algorithm is evaluated under the same conditions as for Algorithm 1. The average area incurred by TMR is always more than three times the original area. Compared with TMR, it can be observed that the average reliability achieved by the proposed scheme under all fault injection scenarios and for all protection thresholds is far better. With 95% protection threshold and an area overhead of just 22.81%, better reliability is achieved by the proposed algorithm than TMR. This is due to the fact that voters in the TMR technique are not protected.

TABLE VI

RELIABILITY OF CIRCUITS BASED ON ASYMMETRIC GATE SIZING TECHNIQUE AGAINST PRORATED FAULTS. (a) ONE PRORATED FAULT. (b) TWO PRORATED FAULTS. (c) FIVE PRORATED FAULTS

(a)

Circuit	95%	98%	99%
alu4	97.89%	97.93%	98.72%
apex1	96.72%	97.92%	98.58%
apex2	99.20%	99.20%	99.20%
apex3	96.88%	97.46%	98.59%
apex4	96.20%	97.62%	98.10%
b12	92.81%	96.13%	98.19%
clip	94.77%	97.31%	98.26%
cordic	98.10%	98.10%	99.07%
ex5	92.90%	96.70%	98.15%
misex1	92.02%	95.40%	97.28%
misex2	93.46%	97.00%	98.72%
misex3	97.64%	98.40%	98.49%
rd84	94.45%	97.51%	98.70%
seq	99.05%	99.05%	99.05%
squar5	91.59%	95.56%	97.93%
table3	98.68%	98.68%	98.93%
table5	98.70%	98.70%	99.06%
z5xp1	92.03%	95.37%	97.77%
Avg.	**95.73%**	**97.45%**	**98.49%**

(b)

Circuit	95%	98%	99%
alu4	95.86%	95.71%	98.18%
apex1	94.20%	95.18%	97.36%
apex2	98.04%	98.04%	98.04%
apex3	94.76%	95.64%	97.04%
apex4	92.74%	95.43%	96.27%
b12	84.23%	91.48%	95.76%
clip	88.73%	93.99%	96.38%
cordic	96.28%	96.28%	97.92%
ex5	88.39%	92.77%	96.18%
misex1	82.55%	91.37%	95.28%
misex2	87.84%	93.75%	96.51%
misex3	95.40%	96.14%	97.60%
rd84	90.22%	93.86%	96.64%
seq	98.22%	98.22%	98.22%
squar5	82.49%	90.35%	95.40%
table3	97.78%	97.78%	98.07%
table5	97.72%	97.72%	98.25%
z5xp1	85.25%	91.48%	94.73%
Avg.	**91.71%**	**94.73%**	**96.88%**

(c)

Circuit	95%	98%	99%
alu4	87.44%	90.69%	94.00%
apex1	86.40%	90.05%	93.84%
apex2	95.42%	95.42%	95.42%
apex3	85.66%	88.26%	93.16%
apex4	84.16%	88.26%	91.94%
b12	62.81%	80.45%	90.68%
clip	76.06%	85.34%	91.15%
cordic	92.14%	92.14%	94.63%
ex5	73.93%	83.93%	90.25%
misex1	63.54%	77.23%	88.88%
misex2	71.17%	83.51%	90.92%
misex3	88.90%	91.27%	93.78%
rd84	76.22%	86.37%	91.58%
seq	95.74%	95.74%	95.74%
squar5	59.87%	74.52%	86.39%
table3	94.28%	94.28%	94.89%
table5	94.52%	94.52%	95.10%
z5xp1	66.50%	81.27%	88.09%
Avg.	**80.82%**	**87.40%**	**92.25%**

TABLE VII

DISTRIBUTION OF PROTECTION SCHEMES

Circuit	# of Gates	98%			99%		
		1T[1]	Full[2]	NP[3]	1T	Full	NP
alu4	832	0.36%	3.25%	95.91%	0.96%	9.74%	87.98%
apex1	2723	2.75%	1.54%	95.41%	5.25%	7.27%	86.49%
apex2	372	0%	0%	100%	0.54%	0%	99.46%
apex3	1791	6.09%	1.34%	92.29%	7.26%	6.98%	83.53%
apex4	2539	3.54%	4.37%	91.77%	8.82%	12.60%	77.79%
b12	88	1.14%	22.73%	73.86%	1.14%	36.36%	60.23%
clip	228	1.32%	14.04%	83.77%	0.88%	24.12%	70.61%
cordic	163	0%	0%	100%	1.23%	1.23%	96.32%
ex5	648	10.49%	7.10%	80.09%	12.35%	17.59%	67.75%
misex1	108	0.93%	35.19%	62.04%	0%	52.78%	44.44%
misex2	151	17.22%	8.61%	70.86%	17.22%	14.57%	64.24%
misex3	1100	2.18%	0.09%	97.64%	1.73%	3.91%	93.64%
rd84	296	1.69%	14.86%	82.77%	3.38%	22.97%	72.30%
squar5	71	0%	42.25%	53.52%	0%	54.93%	40.85%
table3	1953	0%	0%	100%	0.77%	0%	99.23%
table5	2020	0%	0%	100%	1.39%	0.05%	98.51%
z5xp1	176	1.70%	33.52%	64.20%	1.70%	46.02%	49.43%

[1] % of gates with single transistor protection
[2] % of gates with full protection
[3] % of gates with no protection

To improve the reliability of the TMR technique, the majority voters are protected by fully protecting the voters using our proposed scheme. The results for TMR with voter protection are shown in Table VIII(b). It can be observed that the average reliability results have significantly improved for different fault injection scenarios as compared with TMR without voter protection at the expense of additional average area overhead of ≈28.5%. In comparison to TMR with voter protection, our proposed technique with 99% protection threshold achieves comparable reliability with a significantly lower area overhead.

For further evaluation, the proposed scheme is then compared with the simulation-based synthesis technique [26]. The technique is based on maximizing the probability of logical masking when a soft error occurs. This is done by extracting subcircuits from an original multilevel circuit, and then resynthesizing each extracted subcircuit to increase fault masking against a single fault, taking advantage of input probabilities and don't care conditions. Table IX shows the reliabilities obtained based on the original circuit, the circuits synthesized by [26], and by the application of our proposed STR technique for the same area overhead obtained by [26]. From Table IX, it is clear that the final synthesized circuits from [26] are unable to achieve 95% reliability against single fault except for $ex1010$ and $apex4$. This is a limitation of the technique in [26] as it improves reliability but cannot achieve the given target reliability. The proposed STR technique achieves slightly better results for all fault injection scenarios in comparison to the circuit synthesized by the technique in [26].

TABLE VIII

RELIABILITY OF CIRCUITS BASED ON TMR TECHNIQUE WITH PRORATED FAULTS. (a) TMR WITHOUT VOTER PROTECTION. (b) TMR WITH VOTER PROTECTION

(a)

Circuit	OH	1 Fault	2 Faults	5 Faults
alu4	203.93%	99.24%	98.69%	94.67%
apex1	206.56%	98.23%	95.54%	89.50%
apex2	203.45%	98.84%	98.17%	94.09%
apex3	211.60%	96.18%	93.22%	81.57%
apex4	202.76%	99.49%	98.15%	95.67%
b12	251.92%	90.69%	80.89%	54.49%
clip	209.41%	98.11%	95.78%	84.77%
cordic	205.83%	98.83%	97.21%	95.60%
ex5	245.25%	91.15%	83.74%	62.89%
misex1	232.14%	90.81%	79.86%	51.91%
misex2	254.73%	81.55%	67.68%	36.96%
misex3	205.21%	98.78%	96.78%	90.70%
rd84	205.66%	98.75%	96.08%	84.78%
seq	204.94%	98.24%	97.28%	92.79%
squar5	255.38%	85.50%	73.56%	36.54%
table3	202.83%	99.00%	98.36%	95.33%
table5	202.98%	99.03%	98.26%	94.76%
z5xp1	225.16%	93.91%	87.78%	65.90%
Avg.	**218.32%**	**95.35%**	**90.94%**	**77.94%**

(b)

Circuit	OH	1 Fault	2 Faults	5 Faults
alu4	207.53%	99.78%	99.30%	97.07%
apex1	215.41%	99.72%	99.59%	98.05%
apex2	207.51%	99.93%	99.37%	98.44%
apex3	228.91%	99.59%	99.36%	98.22%
apex4	205.21%	99.60%	99.16%	97.91%
b12	329.44%	98.64%	95.30%	76.26%
clip	222.78%	99.75%	98.10%	90.49%
cordic	211.43%	99.79%	98.65%	95.97%
ex5	305.03%	99.67%	98.30%	96.34%
misex1	288.88%	98.53%	94.35%	73.16%
misex2	347.30%	99.37%	98.02%	92.85%
misex3	212.74%	99.73%	99.15%	97.47%
rd84	212.58%	99.47%	98.05%	89.84%
seq	210.86%	99.76%	99.63%	98.77%
squar5	356.21%	98.16%	91.33%	60.67%
table3	205.66%	99.84%	99.38%	98.53%
table5	206.14%	99.84%	99.44%	98.61%
z5xp1	268.10%	98.91%	95.87%	81.39%
Avg.	**246.76%**	**99.45%**	**97.91%**	**91.11%**

TABLE IX

COMPARISON OF CIRCUIT RELIABILITY FOR THE PROPOSED STR TECHNIQUE WITH THE TECHNIQUE IN [26]

Circuit	Original			Synthesized by [26]				STR applied to Original[3]		
	Area (μ)	S[1]	2	OH	S	1P[2]	2P	S	1P	2P
apex3	1994.46	84.16%	68.96%	31.68%	92.60%	89.18%	78.52%	93.15%	91.45%	82.78%
apex4	2532.66	87.06%	76.70%	50.79%	95.74%	92.63%	86.37%	96.20%	93.95%	87.60%
bench1	1313.52	82.32%	67.98%	34.98%	92.86%	91.18%	83.17%	93.46%	91.69%	84.10%
cps	1452.36	78.14%	59.78%	46.35%	91.64%	88.51%	77.85%	92.82%	89.33%	81.43%
duke2	535.86	79.22%	64.10%	30.86%	91.06%	87.83%	77.43%	91.90%	89.55%	78.42%
ex1010	4219.02	87.64%	79.18%	42.17%	95.52%	94.85%	89.88%	96.22%	95.75%	90.41%
exp	363.48	75.34%	56.72%	27.90%	89.48%	85.86%	74.10%	89.38%	86.98%	75.53%
misex3	883.74	87.36%	76.18%	29.30%	94.30%	93.44%	86.15%	95.12%	94.35%	85.88%
spla	475.8	81.08%	65.36%	18.85%	87.62%	86.02%	74.83%	89.92%	87.61%	75.82%
table3	991.38	85.74%	73.18%	29.19%	93.28%	92.08%	83.93%	94.12%	92.34%	85.47%
table5	1106.04	82.64%	68.56%	38.22%	94.30%	92.45%	85.69%	94.78%	92.26%	86.78%
test1	1040.52	82.00%	68.18%	37.18%	92.82%	89.95%	82.01%	93.78%	90.09%	83.60%
Avg.		**82.73%**	**68.74%**	**34.79%**	**92.60%**	**90.33%**	**81.66%**	**93.40%**	**91.28%**	**83.15%**

[1] Single fault
[2] 1 prorated faults
[3] STR applied to Original with area overhead constraint mentioned in column header "OH"

However, our proposed STR technique has the advantage that it can be applied to achieve any given target reliability or under any given area overhead constraint.

It may be noted that the technique proposed in [26] and our proposed technique are complementary to each other. This is because the technique in [26] is based on enhancing logical masking and is applied at the gate level while our proposed technique is based on protecting sensitive transistors at the transistor level through transistor sizing. Hence, applying both the techniques could produce better results than applying any of the techniques separately. To illustrate this, Algorithm 1 is applied on both the original circuits and the synthesized circuits obtained by [26] with a target reliability of 99%. From Table X, it is clear that the proposed technique applied on top of the synthesized circuits obtained by [26] results in significant area savings as compared with applying STR alone on the original circuits. This clearly indicates that the proposed method is scalable and can be used to further improve other techniques.

TABLE X

RELIABILITIES OF CIRCUITS BASED ON APPLYING THE PROPOSED STR TECHNIQUE TO CIRCUITS OBTAINED BY THE TECHNIQUE IN [26]

Circuit	STR Applied to Original with 99%		STR Applied to Syntheized by [26] with 99%	
	OH	Rel	OH	Rel
apex3	170.27%	99.10%	141.93%	99.05%
apex4	109.29%	99.02%	96.83%	99.10%
bench1	148.65%	99.01%	100.04%	99.09%
cps	164.66%	99.00%	160.33%	99.01%
duke2	140.69%	99.06%	135.10%	99.03%
ex1010	90.39%	99.10%	60.54%	99.01%
exp	168.03%	99.04%	149.10%	99.02%
misex3	81.65%	99.05%	75.92%	99.12%
spla	115.83%	99.05%	112.21%	99.00%
table3	89.77%	99.10%	67.00%	99.12%
table5	116.56%	99.10%	71.00%	99.13%
test1	165.14%	99.08%	115.49%	99.02%
Avg.	**130.08%**	**99.06%**	**107.12%**	**99.06%**

VII. CONCLUSION

In this paper, we have proposed an STR-based fault tolerance technique for combinational circuits. The technique can

be applied to achieve a given circuit reliability or enhance the reliability of a circuit under a given area constraint. The technique is based on estimating the POF of each transistor and iteratively protecting transistors with the highest POF until the desired objective is achieved. Transistors are protected based on duplicating and scaling a subset of transistors necessary for providing the protection. Experimental results on LGSynth91 benchmarks demonstrate the effectiveness of the proposed technique. Compared with the existing transistor sizing techniques, the proposed algorithm incurs significantly less area overhead with similar reliability measures. Better reliability results are also achieved in comparison to TMR with lower area overhead. Unlike TMR, which has an area overhead of at least three times the area overhead of the original circuit, the area overhead of the proposed technique varies depending on the reliability of the original circuit. For some circuits, high reliability ($>99\%$) is achieved with small area overhead ($<10\%$). In addition, the reliability of the TMR technique has been enhanced significantly by protecting the voters based on applying the proposed technique. In addition, comparison with the simulation-based synthesis technique further highlights the merit of the proposed method.

A novel gate-level reliability evaluation technique has also been proposed that achieves reliability values similar to those obtained by simulation at the transistor level with the orders of magnitude less CPU time.

REFERENCES

[1] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A defect-tolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, no. 5370, pp. 1716–1721, 1998.

[2] N. Cohen, T. S. Sriram, N. Leland, D. Moyer, S. Butler, and R. Flatley, "Soft error considerations for deep-submicron CMOS circuit applications," in *Proc. Int. Electron Devices Meeting (IEDM)*, Dec. 1999, pp. 315–318.

[3] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 3, pp. 583–602, Jun. 2003.

[4] J. F. Ziegler *et al.*, "IBM experiments in soft fails in computer electronics (1978–1994)," *IBM J. Res. Develop.*, vol. 40, no. 1, pp. 3–18, Jan. 1996.

[5] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 305–316, Sep. 2005.

[6] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proc. Int. Conf. Dependable Syst. Netw. (DSN)*, 2002, pp. 389–398.

[7] T. Karnik and P. Hazucha, "Characterization of soft errors caused by single event upsets in CMOS processes," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 2, pp. 128–143, Apr./Jun. 2004.

[8] J. Henkel *et al.*, "Reliable on-chip systems in the nano-era: Lessons learnt and future trends," in *Proc. 50th ACM/EDAC/IEEE Annu. Design Autom. Conf. (DAC)*, May/Jun. 2013, pp. 1–10.

[9] S. Rehman, F. Kriebel, M. Shafique, and J. Henkel, "Reliability-driven software transformations for unreliable hardware," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 11, pp. 1597–1610, Nov. 2014.

[10] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Autom. Stud.*, vol. 34, pp. 43–98, 1956.

[11] J. Han, "Fault-tolerant architectures for nanoelectronic and quantum devices," Ph.D. dissertation, Dept. Appl. Sci., Delft Univ. Technol., Delft, The Netherlands, 2004.

[12] D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems: Design and Evaluation*, 3rd ed. Natick, MA, USA: A. K. Peters, Ltd., 1998.

[13] A. Namazi and M. Nourani, "Reliability analysis and distributed voting for NMR nanoscale systems," in *Proc. 2nd Int. Design Test Workshop (IDT)*, Dec. 2007, pp. 130–135.

[14] M. Hamamatsu, T. Tsuchiya, and T. Kikuno, "On the reliability of cascaded TMR systems," in *Proc. IEEE 16th Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Dec. 2010, pp. 184–190.

[15] A. H. El-Maleh and F. C. Oughali, "A generalized modular redundancy scheme for enhancing fault tolerance of combinational circuits," *Microelectron. Rel.*, vol. 54, no. 1, pp. 316–326, 2014.

[16] A. H. El-Maleh and A. S. Al-Qahtani, "A finite state machine based fault tolerance technique for sequential circuits," *Microelectron. Rel.*, vol. 54, no. 3, pp. 654–661, 2014.

[17] K. Mohanram and N. A. Touba, "Partial error masking to reduce soft error failure rate in logic circuits," in *Proc. 18th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Nov. 2003, pp. 433–440.

[18] A. K. Nieuwland, S. Jasarevic, and G. Jerin, "Combinational logic soft error analysis and protection," in *Proc. 12th IEEE Int. On-Line Test. Symp. (IOLTS)*, Jul. 2006, p. 6.

[19] C. Zoellin, H. Wunderlich, I. Polian, and B. Becker, "Selective hardening in early design steps," in *Proc. 13th Eur. Test Symp.*, May 2008, pp. 185–190.

[20] Y. Dotan, N. Levison, and D. Lilja, "Fault tolerance for nanotechnology devices at the bit and module levels with history index of correct computation," *IET Comput. Digit. Techn.*, vol. 5, no. 4, pp. 221–230, Jul. 2011.

[21] J. Teifel, "Self-voting dual-modular-redundancy circuits for single-event-transient mitigation," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 6, pp. 3435–3439, Dec. 2008.

[22] E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," *IEEE Trans. Comput.*, vol. 61, no. 3, pp. 323–336, Mar. 2012.

[23] K.-C. Wu and D. Marculescu, "Soft error rate reduction using redundancy addition and removal," in *Proc. Asia South Pacific Design Autom. Conf. (ASPDAC)*, Mar. 2008, pp. 559–564.

[24] S. Almukhaizim and Y. Makris, "Soft error mitigation through selective addition of functionally redundant wires," *IEEE Trans. Rel.*, vol. 57, no. 1, pp. 23–31, Mar. 2008.

[25] A. Zukoski, M. R. Choudhury, and K. Mohanram, "Reliability-driven don't care assignment for logic synthesis," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2011, pp. 1–6.

[26] A. H. El-Maleh and K. A. K. Daud, "Simulation-based method for synthesizing soft error tolerant combinational circuits," *IEEE Trans. Rel.*, vol. 64, no. 3, pp. 935–948, Sep. 2015.

[27] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *Proc. 17th IEEE VLSI Test Symp. (VTS)*, Apr. 1999, pp. 86–94.

[28] A. H. El-Maleh, B. M. Al-Hashimi, A. Melouki, and F. Khan, "Defect-tolerant n^2-transistor structure for reliable nanoelectronic designs," *IET Comput. Digit. Techn.*, vol. 3, no. 6, pp. 570–580, Nov. 2009.

[29] J. Han, E. Leung, L. Liu, and F. Lombardi, "A fault-tolerant technique using quadded logic and quadded transistors," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 8, pp. 1562–1566, Aug. 2015.

[30] A. Mukherjee and A. S. Dhar, "Fault tolerant architecture design using quad-gate-transistor redundancy," *IET Circuits, Devices Syst.*, vol. 9, pp. 152–160, May 2015. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-cds.2014%.0106

[31] S. N. Pagliarini, L. A. de B. Naviner, and J.-F. Naviner, "Selective hardening methodology for combinational logic," in *Proc. 13th Latin Amer. Test Workshop (LATW)*, Apr. 2012, pp. 1–6.

[32] Q. Zhou and K. Mohanram, "Gate sizing to radiation harden combinational logic," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 25, no. 1, pp. 155–166, Jan. 2006.

[33] C. Lazzari, G. Wirth, F. L. Kastensmidt, L. Anghel, and R. A. da Luz Reis, "Asymmetric transistor sizing targeting radiation-hardened circuits," *Elect. Eng.*, vol. 94, no. 1, pp. 11–18, 2012.

[34] W. Sootkaneung and K. K. Saluja, "Sizing techniques for improving soft error immunity in digital circuits," in *Proc. ISCAS*, vol. 232. 2010.

[35] W. Sootkaneung and K. K. Saluja, "On techniques for handling soft errors in digital circuits," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2010, pp. 1–9.

[36] W. Sootkaneung and K. K. Saluja, "Soft error reduction through gate input dependent weighted sizing in combinational circuits," in *Proc. 12th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2011, pp. 1–8.

[37] *Predictive Technology Model for Spice*, accessed on May 31, 2016. [Online]. Available: http://ptm.asu.edu/

[38] A. Dharchoudhury, S. M. Kang, H. Cha, and J. H. Patel, "Fast timing simulation of transient faults in digital circuits," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 1994, pp. 719–726.

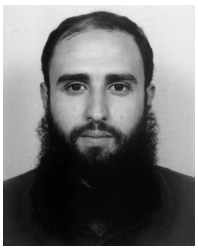
[39] G. C. Messenger, "Collection of charge on junction nodes from ion tracks," *IEEE Trans. Nucl. Sci.*, vol. 29, no. 6, pp. 2024–2031, Dec. 1982.
[40] B. D. Olson *et al.*, "Simultaneous single event charge sharing and parasitic bipolar conduction in a highly-scaled SRAM design," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2132–2136, Dec. 2005.
[41] N. Seifert, B. Gill, V. Zia, M. Zhang, and V. Ambrose, "On the scalability of redundancy based SER mitigation schemes," in *Proc. IEEE Int. Conf. Integr. Circuit Design Technol. (ICICDT)*, May 2007, pp. 1–9.
[42] O. A. Amusan *et al.*, "Mitigation techniques for single-event-induced charge sharing in a 90-nm bulk CMOS process," *IEEE Trans. Device Mater. Rel.*, vol. 9, no. 2, pp. 311–317, Jun. 2009.
[43] H.-H. K. Lee *et al.*, "LEAP: Layout design through error-aware transistor positioning for soft-error resilient sequential cell design," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, May 2010, pp. 203–212.
[44] H. K. Lee and D. S. Ha, "HOPE: An efficient parallel fault simulator for synchronous sequential circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 15, no. 9, pp. 1048–1058, Sep. 1996.
[45] S. M. Sait and H. Youssef, *Iterative Computer Algorithms With Applications in Engineering: Solving Combinatorial Optimization Problems*, 1st ed. Los Alamitos, CA, USA: IEEE Computer Society Press, 1999.
[46] *Lgsynth'91 Benchmark Circuits*, accessed on Jun. 24, 2014. [Online]. Available: http://ddd.fit.cvut.cz/prj/benchmarks/
[47] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*. Norwell, MA, USA: Kluwer, 1984.
[48] E. M. Sentovich *et al.*, "SIS: A system for sequential circuit synthesis," Dept. EECS, Univ. California, Berkeley, Berkeley, CA, USA: Tech. Rep. UCB/ERL M92/41, 1992.

Ahmad T. Sheikh received the B.S. degree in computer science and engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2005, the M.S. degree in electrical engineering from the College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Rawalpindi, Pakistan, in 2008, and the Ph.D. degree in computer science and engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2016.

His current research interests include fault-tolerant design, CAD, synthesis and design of digital systems, and nondeterministic heuristic algorithms.

Aiman H. El-Maleh is currently an Associate Professor with the Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He holds three U.S. patents. His current research interests include synthesis, testing, and verification of digital systems, defect-tolerant design, VLSI design, and design automation.

Dr. El-Maleh is the winner of the best paper award for the most outstanding contribution in the field of test at the European Design and Test Conference in 1995.



Muhammad E. S. Elrabaa received the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 1991 and 1995, respectively.

He is currently an Associate Professor with the Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He has authored or co-authored numerous papers and a book, and holds two U.S. patents. His current research interests include networks-on-chip, defect tolerant circuit techniques, and reconfigurable computing.

Sadiq M. Sait received the bachelor's degree in electronics from Bangalore University, Bengaluru, India, in 1981, and the master's and Ph.D. degrees in electrical engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 1983 and 1987, respectively.

He has been with the Department of Computer Engineering, King Fahd University of Petroleum and Minerals, since 1987, where he is currently a Professor. He has authored over 200 research papers, contributed chapters to technical books, and lectured in over 25 countries. He is the Principle Author of the books entitled *VLSI Physical Design Automation: Theory and Practice* (Europe: McGraw-Hill Book Company, 1995) (and also co-published by IEEE Press), and *Iterative Computer Algorithms with Applications in Engineering (Solving Combinatorial Optimization Problems)*, (CA, USA: IEEE Computer Society Press, 1999). His current research interests include digital design automation, VLSI system design, high level synthesis, and iterative algorithms.