General Methodology for Soft-Error-Aware Power Optimization Using Gate Sizing

Foad Dabiri, *Student Member, IEEE*, Ani Nahapetian, *Member, IEEE*, Tammara Massey, *Member, IEEE*, Miodrag Potkonjak, *Member, IEEE*, and Majid Sarrafzadeh, *Fellow, IEEE*

***Abstract*—Power consumption has emerged as the premier and most constraining aspect in modern microprocessor and application-specific designs. Gate sizing has been shown to be one of the most effective methods for power (and area) reduction in CMOS digital circuits. Recently, as the feature size of logic gates (and transistors) is becoming smaller and smaller, the effect of soft-error rates caused by single-event upsets (SEUs) is becoming exponentially greater. As a consequence of technology feature size reduction, the SEU rate for typical microprocessor logic at sea level will go from one in hundred years to one every minute. Unfortunately, the gate sizing requirements of power reduction and resiliency against SEU can be contradictory. 1) We consider the effects of gate sizing on SEU and incorporate the relationship between power reduction and SEU resiliency to develop a new method for power optimization under SEU constraints. 2) Although a nonlinear programming approach is a more obvious solution, we propose a convex programming formulation that can be solved efficiently. 3) Many of the optimal existing techniques for gate sizing deal with an exponential number of paths in the circuit. We prove that it is sufficient to consider a linear number of constraints. 4) We generalize our methodology to include nonlinear delay models and leakage power as well. As an important preprocessing step, we apply statistical modeling and validation techniques to quantify the impact of fault masking on the SEU rate. Furthermore, we adapt our method to incorporate process variation and evaluate our gate sizing technique under uncertainty. We evaluate the effectiveness of our methodology on ISCAS benchmarks and show that error rates can be reduced by a factor of 100%–200% while, on average, the power reduction is simultaneously decreased by less than 6%–10%, respectively, compared to the optimal power saving with no error rate constraints.**

***Index Terms*—Gate sizing, logic synthesis, optimization, power, soft error (SE).**

I. Introduction

SINGLE-EVENT upsets (SEUs) from transient faults have emerged as a key challenge in logic circuitry design [26]. Recent studies indicate that from 1992 to 2011, the SEU rate for logic will increase by more than a billion times and will surpass the soft error rates (SERs) of unprotected memory. As a consequence of technology feature size reduction, the SEU rate for typical microprocessor logic at sea level will go from one in hundred years to one every minute [26], resulting in a clear need for addressing the problem in systematic way. SEU faults arise from energetic particles such as neutrons from cosmic rays and alpha particles from packaging material, generating electron–hole pairs as they pass through a semiconductor device [32]. During the ICs' normal operation, these faults can be caused by electromagnetic interference. Transistor source and diffusion nodes can collect these charges, and a sufficient amount of accumulated charge may invert the state of a logic device, such as an SRAM cell, a latch, or a gate, thereby introducing a logical fault into the circuit's operation. Because this type of fault does not reflect a permanent failure of the device, it is termed soft error (SE) or transient fault (TF).

Advances in microelectronic technology, which shrink IC size to the nanometer range while also reducing the power supply, are making electronic circuits increasingly susceptible to TFs. In fact, the reduction of the charge stored on circuit nodes, along with the decrease in noise margins, greatly increases the probability of voltage glitches temporarily altering nodes' voltage values [4]. Meanwhile, the continuous increase in ICs' operating frequencies makes the sampling of such glitches increasingly probable. Consequently, TFs will become a frequent cause of failure in many applications as the technology advances.

Power consumption has been recognized as the critical constraint in modern microprocessor and application-specific designs [5], [17]. In addition, gate sizing has been one of the most effective methods for power minimization in CMOS digital circuits. Unfortunately, gate sizing requirements for power reduction and resiliency against SEU are contradictory. The possible tradeoffs between gate sizing and power consumption have been studied in [7]. We consider the effects of gate sizing on SEU and incorporate the relationship between power reduction and SEU resiliency, and we have developed a new method of power optimization under SEU constraints, which leverages convex programming to obtain provably optimal solutions. As an important preprocessing step and consideration, we apply statistical modeling and validation techniques.

Gate sizing is a timing optimization process in high-performance very large scale integration (VLSI) circuit design. In this design process, the size of each gate in a combinational circuit is properly tuned so that circuit area and/or overall power dissipation is minimized under specified timing constraints. Gate sizing or the similar problem of transistor sizing has been an active research topic in recent years. Many approaches have been proposed before [2], [18], [21], [24], [25], [30]. Previous approaches that have taken power considerations into account

Manuscript received October 14, 2007; revised February 27, 2008. Current version published September 19, 2008. This paper was recommended by Associate Editor S. Vrudhula.

The authors are with the Computer Science Department, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: dabiri@cs.ucla.edu; ani@cs.ucla.edu; tmassey@cs.ucla.edu; miodrag@cs.ucla.edu; majid@cs.ucla.edu).

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/TCAD.2008.2003268

during transistor sizing include those in [1], [16], and [29]. The approach in [16] utilizes linear dependence between power and gate sizes; however, since it optimizes one path at a time, the approach may lead to suboptimal solutions.

A linear programming approach for exploring the power–delay–area tradeoff for a CMOS circuit is presented in [1]; we use more accurate nonlinear logical effort delay models in this work. Another linear-programming-based approach is presented in [29]. Power optimization with convex programming is proposed by Menezes *et al.* [21]. In their work, timing constraints are constructed for every path in the circuit which can potentially generate a very large (exponential) number of constraints. In [22], a piecewise convex programming has been proposed, which only targets power with no consideration for SEU and transient errors. In order to capture the effects of logic fault masking, we introduce resubstitution-based statistical methodology and techniques [10], [9] for quantifying error propagation through logic circuitry. We also introduce a new formulation for gate sizing problem using convex programming. Our approach is different from previous ones because the number of constraints in our formulation is linear with respect to circuit size as opposed to the exponential size of constraints in previous work. At the same time, we impose a bound on SERs and evaluate the performance of gate sizing considering SEU. This paper is an extended version of our preliminary results published in [8].

The rest of this paper is organized in the following way. First, in Section II, we go over the preliminaries and cover the models that we have used for delay and SEs. In Section III, we introduce the statistical methodology that we have incorporated to calculate logical masking probabilities. Our formulation and problem transformation are presented in Section IV. Sections V and VI include generalization of our method and incorporation of process variation. Finally, Section IV illustrates the simulation results on ISCAS '85 benchmarks.

II. Preliminaries

A. Power and Delay Models

Power dissipation of gates in digital CMOS circuits is composed of dynamic and static components. In this section, we consider dynamic power consumption, and furthermore, in Section V, we will study leakage power as well. Dynamic power corresponds to the power dissipated in charging and recharging internal capacitors in every gate, which is given by

$$P_{\text{dynamic}} = \sum_{i=1}^{N} C_i f_{\text{clock}} V_{\text{DD}}^2 = \sum_{i=1}^{N} \Phi_i . W_i \tag{1}$$

where C_i is the effective switching capacitance of the gate i, which is a linear function of the size of the gate; f_{clock} is the clock frequency; and V_{DD} is the power supply voltage. Φ_i simply represents the linear dependence of power on size since the capacitance of a gate is a linear function of size (width of the gate). The aforementioned sum is taken over all the gates in the circuit.

Gate delay can be represented as a function of the internal capacitors of logic gates. We use the logical effort method to model the delay [28], which is a linear delay model. Furthermore, in Section V, we generalize our methodology for nonlinear delay modes as well.

The delay d_i of gate i can be written as

$$d_i = p_i + g_i h_i \tag{2}$$

where p_i is the parasitic delay of the gate and is independent of size; g_i is the logical effort of the gate, which is intuitively the driving capability of the gate; and h_i is the electrical effort (gain). h_i is the size-dependent term in the delay

$$h_i = \frac{\sum_j C_j}{C_i}. \tag{3}$$

The aforesaid sum is taken over all the loads that gate i drives. As stated previously, gate capacitance is linearly dependent on the size of the gate, and therefore, it is a function of size. To represent the dependence of delay on size, we rewrite (2) using a new function κ

$$d_i = \kappa_i(W_i, W_j, \ldots) = p_i + g_i \frac{\sum_j k_j W_j}{W_i} \tag{4}$$

where $(C_j/C_i) = (k_j W_j)/W_i$.

The logical effort model is a simplified gate delay model which may not be very accurate for current circuit technologies. We have chosen this model to illustrate the concept of SER impact on power optimization and how we can address this issue. In Section V, we generalize our formulation using accurate power and delay models.

B. SEU

An SEU is an event that occurs when a charged particle deposits some of its charge in a microelectronic device, such as a CPU, a memory chip, or a power transistor. This happens when cosmic particles collide with atoms in the atmosphere, creating cascades or showers of neutrons and protons. At deep-submicrometer geometries, this affects semiconductor devices at sea level. In space, the problem is worse in terms of higher energies. Similar energies are possible on a terrestrial flight over the poles or at high altitude. Trace amounts of radioactive elements in chip packages also lead to SEUs. Frequently, SEUs are referred to as bit flips.

A method for estimating SER in CMOS SRAM circuits was recently developed by Hazucha *et al.* [15]. This model estimates SER due to atmospheric neutrons (neutrons with energies > 1 MeV) for a range of submicrometer feature sizes. It is based on a verified empirical model for 600-nm technology, which is then scaled to other technology generations. The basic form of this model is

$$\text{SER} = F \times A \times e^{-\frac{Q_{\text{crit}}}{Q_S}} \tag{5}$$

where F is the neutron flux with energy ≥ 1 MeV (in particles per square centimeter per second), A is the area of the circuit that is sensitive to particle strikes (the sensitive area is the area of the source of the transistors, which is a function of gate size) (in square centimeters), Q_{crit} is the critical charge (in femtocoulombs), and Q_S is the charge collection efficiency of the device (in femtocoulombs).[1]

Cazeaux *et al.* [4] presents a very accurate model for Q_{crit} and its dependence on gate sizes. In the following model, Q_{crit} for the gate i is dependent on gate sizes as stated in

$$Q_{\mathrm{crit}_i} = Q_{\mathrm{crit}_{\min}} + a'_i(W_i - W_{i_{\min}}) + \sum_j b'_j(W_j - W_{j_{\min}}) \quad (6)$$

where $Q_{\mathrm{crit}_{\min}}$ is the critical charge for minimum driver conductance, minimum diffusion capacitances, and minimum fanout gate input capacitance. Coefficients a'_i and b'_j are constant parameters that weigh the contribution to Q_{crit}. The sum is taken over all gates driven by gate i.

As seen in (5), gate sizing has an effect on Q_{crit}; therefore, we use a function, Θ, to represent this dependence

$$Q_{\mathrm{crit}_i} = \Theta_i(W_i, W_j, \ldots). \quad (7)$$

Furthermore, the sensitive area to SEU, A, is linearly dependent on size

$$A_i = \alpha_i W_i. \quad (8)$$

Substituting Q_{crit} and A_i in (5) gives us a nonlinear relationship between error rate and gate sizes for a given logic gate. It is important to notice that even if an SE is generated in a logic gate, it does not necessarily propagate to the output. SEs can be masked due to the following factors.

1) Logical masking occurs when the output is not affected by the error in a logic gate due to subsequent gates whose outputs only depend on other inputs.
2) Temporal masking (latching-window masking) occurs in sequential circuits when the pulse generated from particle hit reaches a latch but not at the clock transition; therefore, the wrong value is not latched.
3) Electrical masking occurs when the pulse resulting from SEU attenuates as it travels through logic gates and wires. In addition, pulses outside the cutoff frequency of CMOS elements will be faded out [11], [26].

Therefore, we assign a probability ρ to each logic gate to indicate how likely a pulse resulting from SEU can survive to the end and cause an error in the output. The final error rate (λ) assigned to each gate i would be $\lambda_i = \mathrm{SER}_i \cdot \rho_i$. In Section III, we introduce a methodology for statistically computing these probabilities.

C. System Lifetime and MTTF

In this paper, we are considering SERs as a measure for system failure. If the error rate in a system is λ, the mean time to failure (MTTF) is

$$\mathrm{MTTF} = \frac{1}{\lambda}. \quad (9)$$

Therefore, if an MTTF that is greater than a given value, such as Υ, is desired, it implies that $\lambda \leq (1/\Upsilon)$.

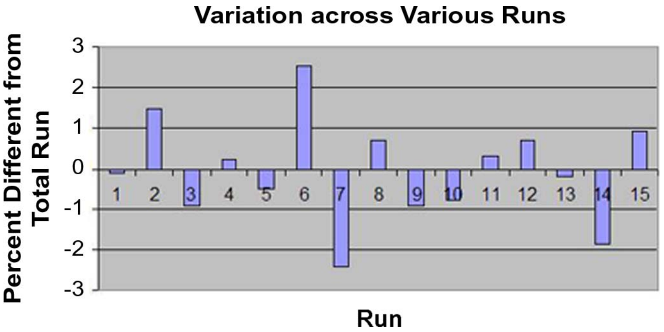


Fig. 1. Shown here for a single benchmark, c432, across 15 different runs, we see a less than 3% variation from the values obtained using the total iterations across all 15 runs. This provides evidence that our results are statistically close to the actual circuit characteristics.

In digital circuits, since all gates are potentially prone to SEs, the total error rate of the circuit (Λ) is

$$\Lambda = \sum_{\forall \mathrm{gate}} \lambda_i. \quad (10)$$

Using (5), we can derive the following equation for the total error rate of a digital circuit:

$$\Lambda = \sum_i \rho_i \mathrm{SER}_i = \sum_i \rho_i \cdot F \cdot A_i e^{-\frac{Q_{\mathrm{crit}_i}}{Q_{S_i}}}. \quad (11)$$

III. Statistical Analysis of Gate Masking

Extensive statistical analysis was done on the circuits from the ISCAS '85 and '89 benchmarks to determine the impact that gate masking can have on the circuit-level SER. The first approach was to observe statistically what the impact of an error in a specific gate would have on the observed error in the circuit. In other words, we compared the global output that we observed with and without SEs in gates. From this analysis, we were able to determine the probability that an error in a specific gate could result in an error in the circuit.

The analysis was conducted by simulating the output values of the circuits for randomly generated input values. First, we simulated the function of the circuit for a statistically large number of times, using random independently generated input values for all the inputs. Then, we compared the output results of the proper functioning circuit with that of the circuit where an SEU had occurred or, in other words, where a bit had been flipped. As would be expected, because of gate masking, the effect of the flipped bit was not always realized at any of the global outputs. We carried out this simulation for every gate in the circuit for all the benchmark circuits.

Specifically, in our experimentation, we considered 2000 independently random input values. Of course, this is a small fraction of the actual number of possible input values, but experimentation over various runs with different input instances revealed a large correlation among runs. We verified our results by running various instances of the experiments to verify that, indeed, the results that we were obtaining were consistent. One such instance, benchmark c432, is shown in the graph in Fig. 1. The graph shows the 15 different runs on the same benchmark

[1] The term "gate" is used to represent both "logic gates" and "gate terminal" of a CMOS transistor, which can be misleading.

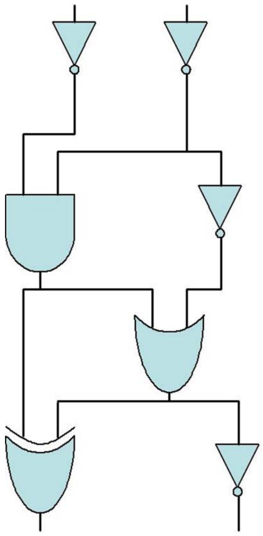


Fig. 2. Example of a digital circuit.

for a statistically significant number of different randomly selected input values. The results obtained are compared with the total sum of all of these 15 runs. The graph shows that each of the 15 runs deviates by less than 3% from the values obtained with the 15 times larger test case. The results help to demonstrate the reliability of the statistical analysis conducted. Furthermore, they give statistical evidence of the correlation between the results obtained and the actual characteristics of the circuit.

On the other hand, we are not taking into account electrical masking. Electrical masking evaluation requires postlayout information, including interconnection characteristics. Gate sizing under study, in this paper, is a prelayout process. An interesting future work will be including layout (placement and routing) information and will utilize it to calculate electrical masking.

IV. Problem Formulation

Intuitively, gate sizing problem can be stated as a timing management problem in such a way that, given a digital circuit with distinct constituting gates, the maximum tolerable power reduction of the circuit is desired while the timing constraints of the system is not violated and, at the same time, the SER of the circuit is bounded by a given value. Although these gates are often modeled as nodes in a directed acyclic graph (DAG), the problem of timing management on nodes is a special case of a more general budgeting on edges. In this section, we illustrate how the delay budgeting on nodes is transformed to delay budgeting on edges. Therefore, we focus on delay budgeting on edges throughout this paper.

In logic synthesis, circuits are usually modeled as a DAG $G = (V, E)$ (see Figs. 2 and 3). In this model, nodes represent the logic gates, and edges stand for the precedence relation between them. We transform the given graph G into G' in such a way that each node v in G is split into two nodes v_1 and v_2 and that an edge connects v_1 to v_2. In the transformed graph, the new edges are basically the logic gates from the original graphs. Fig. 3 shows an example of such transformation. In order to have a single input and a single output, nodes s and t have been added to the graph; s is connected to all primary inputs, and all primary outputs are connected to t.

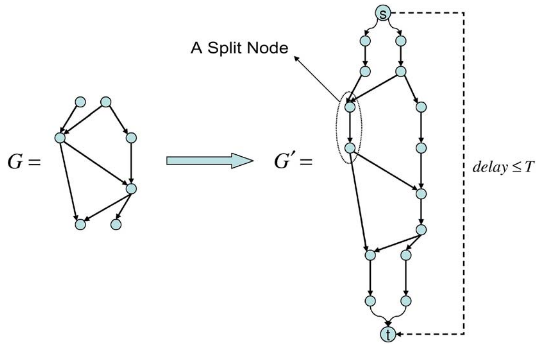


Fig. 3. DAG representation of the circuit in Fig. 2 and its transformation. Each node (gate) in the original DAG is replaced by an edge in G'.

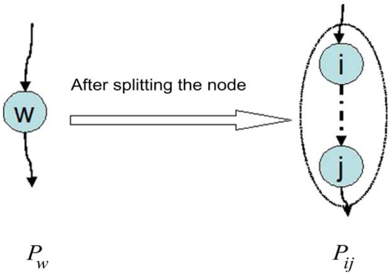


Fig. 4. Since each node is replaced by an edge in the DAG transformation, individual unit parameters such as delay and power are represented with two indices instead of one.

The delay of a path $p = \langle s, v_1, v_2, \ldots, t\rangle$ from node s to node t is equal to the summation of the delays of each edge along the path. We use the terms "delay of a path" and "the distance between nodes s and t" interchangeably; here, the sum is taken over all the edges in path p.

From this point on, we use double indices since we are dealing with edges instead of nodes. In other words, since each node is replaced by an edge, indices represent the edges. For instance, assume that node w is split and that the new edge e_{ij} is representing the node (see Fig. 4). In this case, instead of P_w for the power consumption of this node, we use P_{ij} to represent its power consumption.

The problem is defined as follows. Given a DAG $G = (V, E)$, a timing constraint T, and an error rate constraint Υ

$$\text{minimize} \sum_{\forall e_{ij} \in E} P_{ij} \tag{12}$$

such that the delay of every path from s to t is less than or equal to T and that the error rate (caused by SEU) is less than Υ. P_{ij} is the power consumption of the ijth edge in the DAG, which is a function of the capacitance of the gate.[2] The timing constraint can be stated as $\sum_{e_{ij} \in p_k} d_{ij} \le T$ for every path p_k from s to t. Note that the number of paths in a DAG is exponential in terms of the number of edges in the graph, so this formulation is not

[2] Indices for gate parameters such as power (P) and delay (d) are changed starting from this section of the paper because every gate is represented by an edge in the transformed graph (see Fig. 3). For example, instead of P_i, we use P_{ij} for the power consumption of a gate.

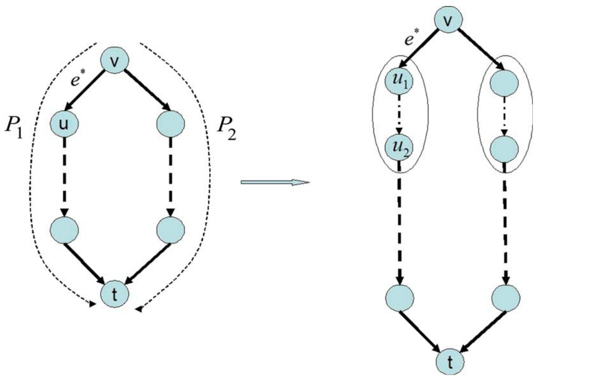


Fig. 5. Figure for Theorem 1. e^* is an edge representing dependence between v and u, and therefore, any positive "dummy" delay can be assigned to it to equalize delays of paths P_1 and P_2.

efficient. Throughout the rest of this section, we will convert it to a formulation with the same objective function that has a linear number of constraints.

Theorem 1: There is an optimal gate sizing solution on a DAG such that the distance between any node u and the output t is independent of the choice of the path taken between them and that this distance is unique.

Proof: Suppose that the claim is not true, i.e., there exists a node v where its distance to t through path P_1 is less than P_2 (see Fig. 5). Before getting to the proof, it is important to note that the edge $e_{u_1u_2}$ is a split node. In other words, it represents the gate u in the original graph and, therefore, the incoming edge to u_1, i.e., e^* is representing an actual edge from the original graph, not any gates.

Without loss of generality, we can assume that P_1 is shorter than P_2. We claim that there exists an edge (e^*) in P_1 that can be slowed down and still not violate the timing constraint because P_2 is on the critical path from v to t. One immediate candidate for e^* is the first edge in P_1. Increasing the delay of e^* by $d_{P_2} - d_{P_1}$ will not cause a timing violation, and since it does not contribute in the cost function nor error rate constraint, the total power dissipation and error rate remain constant. This increase is made by assigning a "dummy" delay to e^*. e^* is on the outgoing edge from the gate v, which represents dependence and does not contribute to the total power consumption. Therefore, we can maintain the same objectives by equalizing the delay of P_1 and P_2, and since this optimization problem has only one global minimum, there should exist an optimal solution in which the statement in Theorem 1 holds.

Theorem 1 still holds for more complex delay–power model. As long as the power dissipation of a gate is a nondecreasing function of gate size (which indeed is), the theorem holds. Therefore, more exact timing models can be used in our proposed formulation. The following observation is immediately inferred from Theorem 1.

Corollary: There is an optimal solution in which the delay of each path in the optimal solution from the primary input node s to the primary output node t is equal to T.

An optimal solution is which the objective (power) is minimized while all the constraints (timing and SER) are met. Now that the delay between every node to the destination in the optimal solution is independent of the path taken, let t_i be a variable assigned to each node v_i that represents its distance to t. A similar technique was proposed in [13], which has resulted in an efficient integer delay budgeting algorithm. We call t_i the distance variable of node v_i. In other words, t_i is the delay of the system from node v_i to the output. Therefore, the delay and power consumption of each edge (node in the original graph) are represented, respectively, by

$$d_{ij} = t_i - t_j = p_{ij} + g_{ij}h_{ij}$$
$$P_{ij} = \phi_{ij}W_{ij}, \qquad \forall e_{ij} \in E.$$

Thus, instead of having a constraint for each path from s to t, we construct the following constraints:

$$t_i - t_j \geq p_i + g_i h_i, \qquad \forall e_{ij} \in E(G') - E(G) \tag{13}$$
$$t_i - t_j \geq 0, \qquad \forall e_{ij} \in E(G') \cap E(G) \tag{14}$$
$$t_s - t_t \leq T \tag{15}$$
$$W_{ij} \geq W_{\min}. \tag{16}$$

Equation (13) enforces that the delay assigned to each gate is greater or equal to its minimum delay (parasitic delay), while (14) assigns a nonnegative delay to those edges in the original graph that represent connection between gates. Equation (15) guarantees that the distance from s to t is less than or equal to the timing constraint T. Equation (15) can be interpreted as a minimum delay required for the virtual edge between s and t. This edge is also shown in Fig. 3. The constraint in (16) is simply the lower bound on transistor width. Error rate can be bounded by the following constraint:

$$\sum_i \rho_i \cdot F \cdot \alpha_i W_i e^{-\frac{\Theta_i(W_i, W_j, \ldots)}{Q_{S_i}}} \leq \Upsilon \tag{17}$$

in which Υ is the desired upper bound on SER. All the timing constraints on paths are reformulated as edge constraints, and the optimization problem can be restated as

$$\text{minimize } f(\vec{W}) = \sum_{\forall e_{ij} \in E} \phi_{ij} W_{ij} \tag{18}$$

subject to the constraints in (13)–(17). The objective in (18), along with the constraints stated in (13)–(17), forms a nonlinear optimization problem with linear number of constraints in terms of the size of the graph. In the next section, we modify and solve this problem using the convex programming method.

Theorem 1 is also valid for circuits with reconvergent fanouts. Although there are dependences of edges across many paths, there exists a delay assignment on an edge to equalize delays on all paths passing through it.

A. Convexity of the Optimization Problem

Global optimization is the task of finding the absolutely best set of parameters to optimize an objective function. In general, there exist solutions that are locally but not globally optimal. Consequently, global optimization problems are typically quite difficult to solve; in the context of combinatorial problems, they are often NP-hard. In convex optimization problems, a locally optimal solution is also globally optimal. These include

LP and QP problems where the objective is positive definite if minimizing (and negative definite if maximizing). Furthermore, NLP problems belong to the same class where the objective is a convex function if minimizing (or a concave function if maximizing), and the constraints form a convex set [3]. In this section, we will show that our proposed formulation yields a convex optimization problem.

Convex optimization problems are far more general than linear programming problems, but they share the desirable properties of LP problems. They can be solved quickly and reliably even in very large size. A convex optimization problem is a problem where all of the constraints are convex functions and the objective is a convex function while minimizing or a concave function while maximizing. With a convex objective and a convex feasible region, there can only be one optimal solution, which is globally optimal. Several proposed methods—notably interior-point methods—can either find the globally optimal solution or prove that there is no feasible solution to the problem.

Geometrically, a function is convex if a line segment drawn from any point $(\vec{x}, f(\vec{x}))$ to another point $(\vec{y}, f(\vec{y}))$—called the chord from $\vec{x}$ to $\vec{y}$—lies on or above the graph of f. Algebraically, f is convex if, for any $\vec{x}$ and $\vec{y}$, and any α between zero and one

$$f\left(\vec{x} + (1-\alpha)\vec{y}\right) \leq f(\vec{x}) + (1-\alpha) f(\vec{y}). \tag{19}$$

The objective defined in (18) is a linear function of the variable W_i and is therefore convex. To state the convexity of the proposed formulation, it is required to show that the feasible solution space created by the constraints is, in fact, convex. The constraints in (13) can be rewritten as

$$\frac{p_i}{t_i} + \frac{g_i h_i}{t_i} + \frac{t_j}{t_i} \leq 1. \tag{20}$$

Since all the variables in (20) are positive, the constraint imposed by (20) is a posynomial expression. Posynomials are not convex in this format, but with a change of variables, they can be mapped to a convex space. If each variable x in a posynomial expression is substituted with e^z, the resulting expression becomes an exponential convex function [3]. The constraint presented in (17) is a nonlinear function, which is generally very hard to handle. This function has both linear and exponential dependences on a variable, which results in a non-monotone function. Each variable W_i is linearly contributing in only one term of (17) (through the sensitive area corresponding to A_i) but is exponentially included in a few terms of same equation (Q_{crit} for gate i and all fan-in gates). Therefore, not only are there more terms that include W_i in the exponent, but their effect compared to the linear dependence of A on W_i is much more significant. On the other hand, shrinking a gate size reduces both the power consumption and the sensitive area to SEU. Therefore, the exponential contributions of gate sizes and power dissipation are two contradictory factors, not the sensitive area.

This observation led us to modify (17) such that we assume an average value for each A_i and change the constraint to exponential form which is convex. This modification and exponential transformation in the posynomial constraints keep the objective and other constraints convex, and the solution space, which is the intersection of all subspaces created by convex constraints, is itself convex. In the simulation section, after calculating gate sizes, we use the actual resulting gate sizes to recompute total error rate, and report the correct numbers.

The significance of our optimization methodology lies within two facts. First, the problem size is linear in terms of the circuit size, which makes the problem generation (constraints and objective) a linear-time process. Second, since the problem is convex (both in simple and accurate delay–power model), finding the optimal point in solution space can be done efficiently.

B. Simulation Flow

In order to analytically evaluate our methodology, we first compute the delay and power of each benchmark and use the delay value T as the timing constraint for the optimization process. The outputs of the optimization process are gate sizes and distance variables. We use gate sizes to compute the total power consumption for comparison with the initial circuit. As discussed in Section IV-A, to evaluate SEU resiliency, we recompute SER and use this value in our report. Further information is provided in Section VII.

V. Generalized Delay and Power Model

In this section, we expand our results for the cases in which more accurate delay and power models are presented. Generally speaking, power dissipation of VLSI circuits is composed of dynamic and leakage power

$$P = P_{\text{dynamic}} + P_{\text{leakage}}. \tag{21}$$

We did study dynamic power before in this paper. In this section, we study leakage power and include it in our gate sizing methodology. Leakage power can be modeled according to

$$P_{\text{leakage}} = V_{\text{DD}} \cdot I_{\text{sub}} \tag{22}$$

where I_{sub} is the subthreshold current which is accurately described in [12]

$$I_{\text{sub}} = A \cdot e^{\frac{V_{\text{gs}} - V_{\text{th}_0} - \gamma V_s + \eta V_{\text{ds}}}{nkT/q}} \times \left[1 - e^{\frac{-V_{\text{ds}}}{kT/q}}\right] \tag{23}$$

where A is the size-dependent term

$$A = \mu_0 c_{\text{ox}} \frac{W}{L} \left(\frac{kT}{q}\right)^2 \times e^{1.8} \tag{24}$$

where c_{ox} is the gate oxide capacitance, μ is the bias mobility, W and L are the width and length of the gate, respectively, k is the Boltzmann constant, T is the absolute temperature, and q is the charge on the electron. Now, we merge the size-independent terms of subthreshold current and V_{DD} into a single function Ψ and restate (22)

$$P_{\text{leakage}} = \Psi \cdot W. \tag{25}$$

As seen in (25), using this model, there is linear dependence between leakage power and size. The generalized objective of our formulation can be restated as

$$\begin{aligned}\text{minimize} \sum_{\forall e_{ij} \in E} P_{ij} &= \sum_{\forall e_{ij} \in E} (\Phi_{ij} \cdot W_{ij} + \Psi_{ij} \cdot W_{ij}) \\ &= \sum_{\forall e_{ij} \in E} (\Phi_{ij} + \Psi_{ij}) \cdot W_{ij}. \qquad (26)\end{aligned}$$

Remember that double indices are used since, in the transformed graph, each gate is represented by an edge.

As for the delay model, we consider the nonlinear delay model as opposed to the linear Elmore delay. A very accurate nonlinear delay model can be found in [31] in which gate delay can be represented as

$$d = \sum_{z=1}^{2} a_z \tau^{b_z} w_n^{c_z} w_p^{d_z} L^{e_z} \qquad (27)$$

where L represents the output load which is the sum of output wiring and gate loading

$$L = \sum kC_j + \sum w_j. \qquad (28)$$

All a_z, b_z, c_z, d_z, and e_z are real-value constants. Using the aforementioned model, timing constraints can be rewritten as

$$t_i - t_j \geq \sum_{z=1}^{2} a_z \tau^{b_z} w_n^{c_z} w_p^{d_z} L^{e_z}, \qquad \forall e_{ij} \in E(G') - E(G) \qquad (29)$$

which is equivalent to

$$\frac{\sum_{z=1}^{2} a_z \tau^{b_z} w_n^{c_z} w_p^{d_z} L^{e_z}}{t_i} + \frac{t_j}{t_i} \leq 1. \qquad (30)$$

In the presence of the more general delay model, SER constraints remain intact. Constraints in the form of (30) are also in posynomial form and, therefore, with the same change of variable, can be transformed into standard convex format. Since the objective in (26) is a linear function of size, the optimization problem will be a convex optimization which maintains the same properties as discussed in Section IV-A.

A. Complexity of Optimization Problem Using General Models

An immediate concern using general delay and power models is its effect of the optimization complexity. Fortunately, this model does not effect the complexity of the problem for the following reasons: the number of constraints in the general formulation is the same as before. In other words, more accurate models do not enforce new constraints. Furthermore, the objective has the same number of terms as seen (26). Therefore, asymptotically, the generalized problem formulation has the same running time.

VI. Process Variation: Statistical Optimization Refinement

Process variation has become an inevitable characteristic of modern CMOS circuits, particularly in sub-100-nm processes [19], [20]. Process parameter variations may result in large variability in circuit delay affecting the yield [14], [27]. Statistical timing analysis and gate sizing methods try to address optimization techniques considering process variations. Many of these techniques are based on adapting a known deterministic method to consider variability [6]. Process variation is not the primary focus of this paper, but in order to signify the problem and show the effectiveness of our method, we incorporated a similar approach as in [6] to address variability issues. The idea behind this approach is to dynamically change the timing constraints on the circuit and reassign gate sizes with the new constraints to meet timing and/or yield constraints.

In this approach, new timing constraint is presented in statistical way. We rewrite (15) as

$$\text{Prob}(t_s - t_t \leq T) \geq \theta \qquad (31)$$

where θ is the tunable bound on timing violation probability, which is set according to the desired yield. $d_c = t_s - t_t$ is the circuit delay which is a random variable. d_c can be modeled as a normally distributed random variable [6]. Equation (31) enforces the critical path delay to be less than T with probability larger than θ. Algorithm 1 summarizes the high-level approach in the SEU-aware gate sizing with yield consideration.

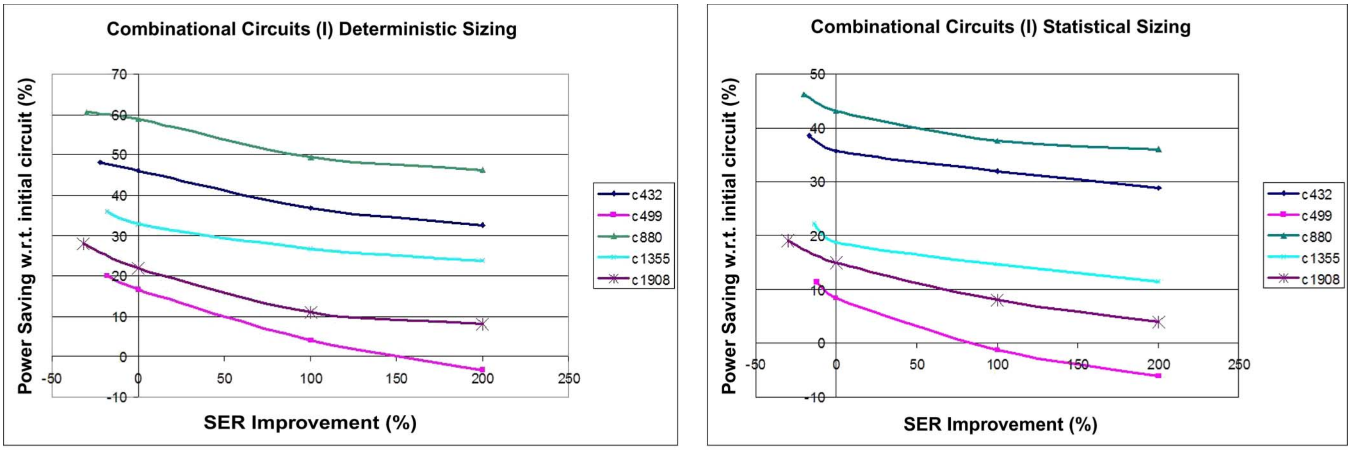
```
Adapting SEU-aware gate sizing to process variation and yield
Solve optimal gate sizes with timing constraint $= T$, initialize $T' = T''$
Extract circuit delay distribution (pdf) through statistical timing analysis
**while** $\text{Prob}(t_s - t_t \leq T) < \theta$ **do**
    Refine the timing constraint $T'$, $T' \leftarrow T' - \delta$
    Reassign deterministic gate sizes using the proposed convex programming method
    Recalculate the statistical characteristics of the circuit delay, i.e., $\mu$ and $\sigma$
**end while**
Compute the power consumption and SER for the final assignment
```

In Algorithm 1, δ is the reduction step in the new timing constraint. The new timing constraint, $T' - \delta$, is set such that if the delay of the resulting circuit has a mean that is equal to $T' - \delta$, then $\text{Prob}(t_s - t_t \leq T) \geq \theta$. The new timing constraint can be calculated as

$$T' \leftarrow T' - s \cdot \left(\phi^{-1}(\theta) - \mu\right), \qquad \theta \geq 0.5 \qquad (32)$$

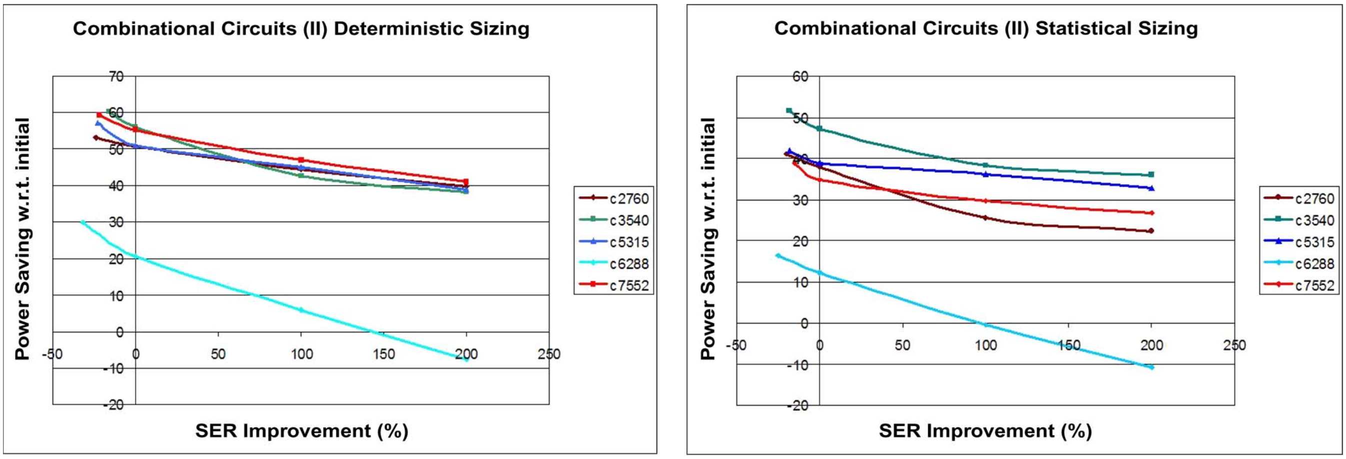
where s is a normalization factor and $\phi^{-1}(z)$ is the inverse of the normal distribution function

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \int_{0}^{z} e^{-z_2/2} dz \qquad (33)$$



(a) (b)

Fig. 6. (a) Simulation results for five combinational circuits. The initial flatness of these curves shows that SER can be reduced by a huge factor without significantly compromising the power saving. (b) Same benchmarks have been resized under process variation. For each SER bound, gate sizing guarantees that the circuit delay is less that the timing constraint with probability of more than 97%.



(a) (b)

Fig. 7. (a) Simulation results for five larger combinational circuits. It shows that power saving in the c6288 benchmark is more dependent to bounds on error rates compared with other benchmarks. (b) Same benchmarks have been resized under process variation. One interesting observation is that, under process variation, the graphs have smaller slopes.

with $z \equiv (x-\mu)/\sigma$ and $dz = dx/\sigma$ for arbitrary distribution. If the standard deviation of a circuit after gate sizing remained intact, $s = 1$ would be the correct choice for the timing constraint adjustment. However, as the circuit goes under sizing, the standard deviation also changes, and setting $s = 1$ may overconstrain the problem. For evaluation purposes, we set $s = 0.7$, which was obtained by multiple simulation runs. It is common to set $\theta \simeq 84\%$ which concludes that the mean is smaller than the timing constraint by 1σ or $\theta \simeq 97\%$ which concludes that the mean is smaller than the timing constraint by 2σ.

VII. Simulation Results

We used the MOSEK convex optimization tool [23] to solve the proposed formulation on ISCAS benchmarks. We applied our technique on 65-nm technology with the assumption that the process variation has Gaussian distribution with $\sigma/\mu = 5\%$. For each benchmark, we calculate the total delay of the circuit, T, with initial size values and then use that delay as the timing constraint for the optimization problem. Each benchmark has a total error rate, Λ, associated with its initial size which we used to impose constraints on the error rate. For each circuit, we minimize the power consumption with four different bounds on the error rate, Υ, starting with the no bound on error rate, error rate of the initial circuit, i.e., Λ, reduced rate by a factor of 100%, and, finally, reduced error rate by a factor of 200%. Figs. 6(a) and 7(a) show the power consumption reduction versus the bound on error rate for combinational circuits. A point (x, y) in these graphs means that power dissipation has been reduced by $y\%$, while the total error rate has been decreased by at least a factor of $x\%$ compared with the error rate in the initial circuit.

It can be observed that an average of 47% power saving can be achieved without any constraint on the error rate for these benchmarks. Obviously, power saving percentage depends on the initial design which we compare our results with. If an original design is far from optimal, then the power saving ratio becomes larger. Therefore, the significance of the results is that

TABLE I
RUNNING TIME COMPARISON FOR DIFFERENT ISCAS BENCHMARKS. THIS TABLE ALSO COMPARES THE RUNNING TIME OF THE OPTIMIZATION PROBLEM BETWEEN THE CASE WITH NO BOUND ON SER AND WHEN WE IMPOSE AN UPPER BOUND ON SER

Benchmark	Number of Gates	CPU time (s) (without SER)	CPU time (s) (with SER)
c432	120	0.28	0.32
c499	162	0.42	0.66
c880	320	2.55	3.78
c1355	506	4.50	6.49
c1908	880	0.25	14.77
c2760	872	8.10	12.22
c3540	1179	17.45	28.73
c5315	1726	30.10	51.56
c6288	2384	98.71	130.54
c7552	2636	124.65	199.04

we can achieve *optimal* power reduction in the presence of SER constraints. This is the reason why we have not compared our results with any other power optimization method since power reduction through gate sizing is a well-known fact, and it is important to know how bounding SERs can be integrated in gate sizing.

We ran the same optimization with process variation. Figs. 6(b) and 7(b) summarize these results. The timing constraint is set such that the circuit delay is less than the constraint with probability $\geq 97\%$, which means that $d_c \leq T + 2\sigma_T$. The average power reduction is about 34% for these benchmarks under process variation.

Furthermore, in order to get a feeling of the running time of the proposed methodology, Table I summarizes the running time of the optimization problem for the set of benchmarks. The first column is the benchmark name, and the second column contains the number of gates in the circuit (each DFF is counted as one gate). The third column is the CPU time in seconds for the solver to find the optimal solution for the case with no bounds on SER, while the fourth column is the CPU time for the case when we impose an upper bound on the error rate. Timing results also validate the fact that our linear size model and the efficient convex programming method are a fast process, and runtime grows almost polynomially with the circuit size.

VIII. CONCLUSION

In this paper, we have introduced a new formulation for gate sizing that targets power optimization, resiliency against SEUs, and timing constraints simultaneously. As a preprocessing step for optimization, we developed a statistical modeling and validation technique that quantifies the impact of fault masking in combinational logic. We formulated the problem as a convex optimization problem of linear size as opposed to the previous convex programming approaches which could potentially be exponential in size. We further generalized our methodology to include nonlinear delay models and leakage power as well. The MOSEK convex optimization tool was used to evaluate the proposed approach on ISCAS benchmarks. We were able to minimize power dissipation for a given timing constraint and various upper bounds on error rates caused by an SEU. An important practical result was that convex-programming-based gate sizing can simultaneously reduce power consumption and improve SEU resiliency. Our simulation showed that different circuits from various benchmarks behave differently when carrying out power optimization while considering SEs. To further generalize the method, we used our method in combination with statistical techniques to consider process variations. As future work, we propose the examination of the question of designing circuits such that, through gate sizing, optimum power savings can be achieved while enforcing low error rates.

REFERENCES

[1] M. R. C. M. Berkelaar and J. A. G. Jess, "Gate sizing in MOS digital circuits with linear programming," in *Proc. EURO-DAC*, 1990, pp. 217–221.

[2] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor sizing for minimizing power consumption of CMOS circuits under delay constraint," in *Proc. ISLPED*, 1995, pp. 167–172.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge Univ. Press, 2004.

[4] J. M. Cazeaux, D. Rossi, M. Omana, C. Metra, and A. Chatterjee, "On transistor level gate sizing for increased robustness to transient faults," in *Proc. IOLTS*, 2005, pp. 23–28.

[5] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. Brodersen, "Optimizing power using transformations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 1, pp. 12–31, Jan. 1995.

[6] S. H. Choi, B. C. Paul, and K. Roy, "Novel sizing algorithm for yield improvement under process variation in nanometer technology," in *Proc. 41st Annu. DAC*, 2004, pp. 454–459.

[7] M. R. Choudhury, Q. Zhou, and K. Mohanram, "Design optimization for single-event upset robustness using simultaneous dual-VDD and sizing techniques," in *Proc. IEEE/ACM ICCAD*, 2006, pp. 204–209.

[8] F. Dabiri, A. Nahapetian, M. Potkonjak, and M. Sarrafzadeh, "Soft error-aware power optimization using gate sizing," in *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation*. New York: Springer-Verlag, 2007, pp. 255–267.

[9] B. Efron, *The Jackknife, the Bootstrap, and Other Resampling Plans*. Philadelphia, PA: SIAM, 1982.

[10] R. J. Tibshirani and B. Efron, *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1994.

[11] S. Mitra, T. Karnik, N. Seifert, and M. Zhang, "Logic soft errors in sub-65 nm technologies design and CAD challenges," in *Proc. DAC*, 2005, pp. 2–4.

[12] F. Farbiz, M. Farazian, M. Emadi, and K. Sadeghi, "Sizing consideration for leakage control transistor," in *Proc. 17th VLSID*, 2004, p. 639.

[13] S. Ghiasi, E. Bozorgzadeh, S. Choudhuri, and M. Sarrafzadeh, "A unified theory of timing budget management," in *Proc. ICCAD*, 2004, pp. 653–659.

[14] M. R. Guthaus, N. Venkateswarant, C. Visweswariaht, and V. Zolotov, "Gate sizing using incremental parameterized statistical timing analysis," in *Proc. ICCAD*, 2005, pp. 1029–1036.

[15] P. Hazucha, C. Svensson, and S. A. Wender, "Cosmic-ray soft error rate characterization of a standard 0.6-μm CMOS process," *IEEE J. Solid-State Circuits*, vol. 35, no. 10, pp. 1422–1429, Oct. 2000.

[16] K. S. Hedlund, "Aesop: A tool for automated transistor sizing," in *Proc. 24th ACM/IEEE DAC*, 1987, pp. 114–120.

[17] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, "Power optimization of variable-voltage core-based systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 12, pp. 1702–1714, Dec. 1999.

[18] K. S. Lowe and P. G. Gulak, "A unified discrete gate sizing/cell library optimization method for design and analysis of delay minimized CMOS and BiCMOS circuits," in *Proc. EURO-DAC*, 1994, pp. 42–47.

[19] M. Mani, A. Devgan, and M. Orshansky, "An efficient algorithm for statistical minimization of total power under timing yield constraints," in *Proc. 42nd Annu. DAC*, 2005, pp. 309–314.

[20] M. Mani and M. Orshansky, "A new statistical optimization algorithm for gate sizing," in *Proc. IEEE ICCD*, 2004, pp. 272–277.

[21] N. Menezes, R. Baldick, and L. T. Pileggi, "A sequential quadratic programming approach to concurrent gate and wire sizing," in *Proc. IEEE/ACM ICCAD*, 1995, pp. 144–151.

[22] N. Miura, N. Kato, and T. Kuroda, "Practical methodology of post-layout gate sizing for 15% more power saving," in *Proc. ASP-DAC*, 2004, pp. 434–437.

[23] *The MOSEK Optimization Tools Manual*, MOSEK ApS, Copenhagen, Denmark, 2002. [Online]. Available: http://www.mosek.com

[24] S. S. Sapatnekar and W. Chuang, "Power vs. delay in gate sizing: Conflicting objectives?" in *Proc. IEEE/ACM ICCAD*, 1995, pp. 463–466.

[25] S. S. Sapatnekar and W. Chuang, "Power–delay optimizations in gate sizing," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 5, no. 1, pp. 98–114, Jan. 2000.

[26] 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. DSN*, 2002, pp. 389–398.

[27] J. Singh, V. Nookala, Z.-Q. Luo, and S. Sapatnekar, "Robust gate sizing by geometric programming," in *Proc. 42nd Annu. DAC*, 2005, pp. 315–320.

[28] I. E. Sutherland and R. F. Sproull, "Logical effort: Designing for speed on the back of an envelope," in *Proc. Univ. California/Santa Cruz Conf. Advanced Res. VLSI*, 1991, pp. 1–16.

[29] Y. Tamiya, Y. Matsunaga, and M. Fujita, "LP based cell selection with constraints of timing, area, and power consumption," in *Proc. ICCAD*, 1994, pp. 378–381.

[30] H. Tennakoon and C. Sechen, "Efficient and accurate gate sizing with piecewise convex delay models," in *Proc. DAC*, 2005, pp. 807–812.

[31] H. Tennakoon and C. Sechen, "Efficient and accurate gate sizing with piecewise convex delay models," in *Proc. 42nd Annu. DAC*, 2005, pp. 807–812.

[32] C. Weaver, J. Emer, S. S. Mukherjee, and S. K. Reinhardt, "Techniques to reduce the soft error rate of a high-performance microprocessor," in *Proc. 31st Annu. ISCA*, 2004, p. 264.

Foad Dabiri (S'03) received the B.S. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, and the M.S. and Ph.D. degrees from the University of California at Los Angeles (UCLA), in 2005 and 2008, respectively.

Since 2003, he has been with the Embedded and Reconfigurable Systems Group, Computer Science Department, and is currently a PostDoc Researcher with the Wireless Health Institute, UCLA. His research area mainly includes wireless health, emphasizing on infrastructure design, applications, and algorithm design for networked embedded systems. He is particularly focusing on hybrid optimization algorithms (power and reliability) for embedded systems.

Ani Nahapetian (S'03–M'07) received the B.S. degree in computer science and engineering and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles (UCLA).

She is currently with the Computer Science Department, UCLA. Her research interests include hardware security, reconfigurable computing, and embedded systems.

Tammara Massey (M'04) received the M.S. degree in computer science from the Georgia Institute of Technology, Atlanta, in 2004. She is currently working towards her Ph.D. degree in computer science at the Embedded and Reconfigurable Computing Laboratory, Computer Science Department, University of California, Los Angeles.

In 2006, she was a visiting researcher and hardware lead of the Advanced Health and Disaster Aid Network at the John Hopkins University Applied Physics Laboratory. Her current research interests include embedded medical system design and development, systems security, and statistical modelling. She is a Student Member of NSBE and SWE.

Miodrag Potkonjak (M'02) received the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 1991.

In 1991, he was with C&C Research Laboratories, NEC USA, Princeton, NJ. Since 1995, he has been with the University of California, Los Angeles, where he is currently a Professor with the Computer Science Department. He has published one book and more than 250 papers. His research interests include applied statistical and optimization techniques, embedded systems, computational sensing, computer-aided design, and computational and system security.

Majid Sarrafzadeh (M'87–SM'92–F'96) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Illinois, Urbana, in 1982, 1984, and 1987, respectively.

He was with Northwestern University, Evanston, IL, as an Assistant Professor in 1987. Since 2000, he has been with the Computer Science Department, University of California, Los Angeles. His recent research interests include embedded and reconfigurable computing, very large scale integration (VLSI) computer-aided design, and the design and analysis of algorithms. He has contributed to "Theory and Practice of VLSI Design."