A Novel Approach to Perform Gate-Level Yield Analysis and Optimization Considering Correlated Variations in Power and Performance

Ashish Srivastava, Kaviraj Chopra, *Student Member, IEEE*, Saumil Shah, *Student Member, IEEE*, Dennis Sylvester, *Senior Member, IEEE*, and David Blaauw, *Member, IEEE*

***Abstract*—Increasing levels of process variation in current technologies have a major impact on power and performance and result in parametric yield loss. In this paper, we develop an efficient gate-level approach to accurately estimate and optimize the parametric yield, defined by leakage power and delay limits, by finding their joint probability distribution function. We consider inter-die variations, as well as intra-die variations, with correlated and random components. The correlation between power and performance arises due to their dependence on common process parameters and is shown to have a significant impact on the yield, particularly in high-frequency bins. We then propose a new heuristic approach to incrementally compute the gradient of yield with respect to gate sizing and gate-length biasing in the circuit with high efficiency and accuracy. We show how this gradient information can be effectively used by a nonlinear optimizer to perform yield optimization. The proposed yield-analysis approach is compared with Monte Carlo simulations and shows high accuracy, with the yield estimates achieving an average error of 2%. The proposed optimization approach is implemented and tested, and we demonstrate an average yield increase of 40% using gate sizing (as compared to a deterministically optimized circuit). Even higher improvements are demonstrated when both gate sizing and gate-length-biasing techniques are used.**

***Index Terms*—Correlation, leakage, variability, yield optimization.**

I. Introduction and Prior Work

CONTINUED process scaling has resulted in a large increase in process variability that leads to large fluctuations in process parameters from their nominal values. These variations have grown, due, in part, to aggressive lithographic techniques that are used to pattern dimensions smaller than the wavelength of light. In addition, smaller device dimensions and fewer doping atoms increase the influence of phenomena, such as line-edge roughness and random-dopant effects. These variation sources translate into wide spreads in performance metrics of current designs. In particular, leakage current, which is extremely sensitive to a number of key process parameters [1], has shown huge fluctuations, with [2] showing a 20× variation in leakage power for a 30% variation in performance across 1000 samples of a design manufactured in a 180-nm technology. In addition, as the contribution of leakage power has grown, the fluctuation in power dissipation is now dominated by leakage power. This results in a negative correlation between the power dissipation and the delay of a design. Thus, high-performance samples of a design are also expected to have higher power dissipation and vice versa. This leads to a two-sided constraint on the feasible region of parametric yield defined by delay and power limits [26] and causes significant yield loss under process variation.

This yield loss will worsen in future technologies due to increasing process variation and the continued significance of leakage power. Another growing design concern is that increased variation results in both a larger spread of leakage power and higher average leakage power. Most traditional optimization approaches do not consider process variations and are unaware of their impact on yield. Reference [3] was the first to consider the impact of variability on circuit optimization. The authors proposed a heuristic approach to prevent the build-up of a large number of paths near the critical delay of the circuit, which reduces the susceptibility of the design to process variations. However, the approach was deterministic in nature and did not use any statistical information during optimization.

Recently, several approaches have been proposed to perform statistical timing or power optimization [4]–[6], [28]–[30], [36]. However, some of these approaches (such as [28]) consider only timing yield optimization, which results in yield loss due to the power constraint. In addition, all current approaches neglect the correlation of power and performance. Furthermore, most of these approaches suffer from large computational complexity and runtimes [4], [6], [29] or are dependent on significantly simplified delay and power models [5], [30], [36]. Thus, there is a critical need to develop approaches that perform accurate and efficient parametric yield optimization, where yield is defined using both power and timing limits. To support these optimization tools, there is an underlying requirement for accurate and computationally efficient yield-estimation approaches.

Previous work in yield estimation has been limited to predicting either timing [7]–[12] or (leakage) power yield [13],

Manuscript received October 9, 2006; revised April 9, 2007 and July 1, 2007. This work was supported in part by grants from the National Science Foundation (NSF), Semiconductor Research Corporation (SRC), Gigascale Systems Research Center (GSRC), Intel, and IBM. This paper was recommended by Associate Editor F. N. Najm.

A. Srivastava was with the University of Michigan, Ann Arbor, MI 48105 USA. He is now with Magma Design Automation, Austin, TX 78758 USA (e-mail: ashishs@magma-da.com; ansrivas@umich.edu).

K. Chopra, S. Shah, D. Sylvester, and D. Blaauw are with the University of Michigan, Ann Arbor, MI 48105 USA.

Digital Object Identifier 10.1109/TCAD.2007.907227

0278-0070/$25.00 © 2008 IEEE

TABLE I
ESTIMATED YIELD FOR DIFFERENT VALUES OF CORRELATION COEFFICIENTS. POWER CONSTRAINT IS SET AT 1.5× NOMINAL POWER. $\mu(D)$ AND $\sigma(D)$ REPRESENT THE MEAN AND STANDARD DEVIATION OF DELAY (D), RESPECTIVELY

Timing Constraint	Estimated Yield		
	Corr = -1.0	Corr = 0.0	Corr = 1.0
$D \leq \mu(D)-\sigma(D)$	0.000	0.095	0.159
$D \leq \mu(D)$	0.100	0.300	0.500
$D \leq \mu(D)+\sigma(D)$	0.441	0.505	0.600
$D \leq \mu(D)+2*\sigma(D)$	0.577	0.586	0.600
$D \leq \mu(D)+3*\sigma(D)$	0.599	0.599	0.600

[27]. Recently, Rao *et al.* [14] presented a chip-level approach to estimate the yield in separate frequency bins given a power constraint. This high-level approach is based on global circuit parameters, such as total device width on a chip. Since it does not use circuit-specific information from a gate-level netlist, it is difficult to use this approach for optimization of gate-level parameters, such as sizes and threshold voltages of individual gates. Another important requirement for an accurate yield-estimation approach is to consider all classes of variations, which impact delay [15] and power [13] differently. Process variations are typically classified into inter-die and intra-die components. Intra-die variations are further classified as having correlated and random components. Traditionally, inter-die variations have been the dominant source of variations, but with process scaling, the random and correlated components of intra-die variations can now exceed inter-die variations [16]. The relative magnitude of these components of variation also depends on the process parameter being considered. For example, gate-length variations are generally considered to have roughly comparable random and correlated components, whereas gate-length-independent threshold voltage is commonly assumed to vary randomly due to random dopant fluctuations [17].

In this paper, we propose a novel approach to compute the parametric yield of a circuit with high efficiency and accuracy, given leakage power and delay limits. This is the first such gate-level approach in estimating parametric yield to the best of our knowledge. Central to the approach is maintaining the correlation between leakage power and delay by expressing both metrics in a canonical form with the same underlying process variations. To demonstrate the importance of this power/delay correlation, Table I shows yield for three different correlation-coefficient values. The yields are calculated for different points on the delay distribution (listed in Column 1) and at a fixed power constraint of 1.5× the nominal power. The data in Table I clearly shows that the correlation of power and delay has a strong impact on parametric yield, particularly for mid- to high-performance speed bins.

The yield-analysis engine is then used to build an optimization approach. The yield optimization is formulated as an unconstrained optimization problem where the objective is to maximize the parametric yield of a design. The optimization is performed using a gradient-based nonlinear optimizer. A gradient-based optimization is not guaranteed to find a global optimum and will eventually converge to one of the local minima. However, the choice of the nonlinear-optimization technique is based on the nonlinear relationships that exist between device lengths and widths and their associated delays, particularly with strong short-channel effects in the nanometer region, and leakage power. Due to their timing accuracy, accurate nonlinear-optimization techniques [39]–[41] have enjoyed significant popularity in deterministic gate sizing over linear-[42] and geometric-programming-based [43] techniques.

Moreover, in the statistical case, it is much harder to find efficient formulations that can exploit the structure offered by simpler delay relationships. The linear relationship is a special case and allows an efficient formulation, which was investigated in [36]. However, this formulation currently suffers from several drawbacks. Most importantly, the node-based formulation in [36] is based on the assumption that all variations are perfectly correlated, and thus, it cannot exploit the statistical averaging of delay variations along a path, which is crucial for true "statistical" optimization of a circuit. In addition, extending the approach to handle more complicated delay models does not seem feasible. Reference [44] provides a good overview of robust formulations of various categories of convex problems and shows that efficient robust formulations exist only in few special cases.

On the other hand, a brute-force gradient-descent approach based on iterative yield analysis leads to large computational overheads. The novelty of our proposed optimization approach lies in an efficient heuristic technique to perform yield gradient computation. This gradient computation technique is then integrated with LANCELOT [31], a large-scale nonlinear optimizer, to improve the parametric yield of the design and is found to provide an 8× improvement in runtime with an average error of 0.1% when compared to a brute-force approach.

The remainder of this paper is organized as follows. Section II reviews the principal-component-based approach to model process variations. Section III presents the core statistical power- and timing-analysis approaches. Section IV develops an approach to estimate the yield, given power and delay constraints. Section V presents the incremental timing- and power-analysis techniques that are used to compute the yield gradient. Section VI provides details regarding the implementation of our yield-optimization approach and presents results including a comparison of our approach to deterministic optimization. We provide our conclusions in Section VII.

II. MODELING PROCESS VARIATION

This section details the variability-modeling infrastructure used in this paper. Much of this framework is similar in spirit to [7] for statistical timing analysis—we also use the same models to consider leakage variability such that the correlation between power and delay is preserved for yield estimation.

In this paper, we consider process variations in gate-length and gate-length-independent threshold voltage (Vth_0), although the approach can be easily extended to consider other sources of variations. The process parameters are expressed as a sum of correlated and random components, and the sum of variances of both these components provides the overall variation in the process parameter. The correlated variations are handled by partitioning the die area using a grid, and a single

random variable (RV) is used to represent the variation in each square of the grid. To simplify the problem, this set of correlated RVs is replaced by another set of mutually independent RVs with zero mean and unit variance by performing principal-component analysis (PCA) of the set of correlated RVs. Note that, generally, the correlation structure of process parameters is defined using a distance-dependent function. Using such a function to define the correlation matrix may result in correlation matrices that are not positive-definite. Simple techniques, such as discarding the negative eigenvalues of such a correlation matrix, may be used to address such issues [34].

We express the delay and leakage power of an individual gate, as shown in the following:

$$\text{Delay} = d_{\text{nom}} + \sum_{i=1}^{p} \alpha_p(\Delta P_p)$$

$$\text{Leakage} = \exp\left(V_{\text{nom}} + \sum_{i=1}^{p} \beta_p(\Delta P_p)\right) \tag{1}$$

where d_{nom} and $\exp(V_{\text{nom}})$ are the nominal values of delay and leakage power, respectively, and the α's and β's represent the sensitivities of delay and the log of leakage to the process parameters under consideration. The RV ΔP_p represents the change in the process parameters from their nominal value. The values of d_{nom} and the sensitivities α_p's are characterized for each timing arc associated with the gate and stored as a 2-D table indexed by the input transition time and the output load capacitance. Similarly, V_{nom} and β_p's are characterized for each input state of the gate. Since variations in dynamic power are typically much smaller than those observed in static power, we focused on statistical leakage-power analysis for yield-estimation purposes. We will later discuss the technique used to consider the impact of dynamic power dissipation while performing circuit optimization.

After the overall circuit is partitioned using a grid, the delay of individual gates is expressed as a function of the RVs defined in the grid. Using the principal-component approach, the delay in (1) is then expressed as

$$\text{Delay} = d_{\text{nom}} + \sum_{i=1}^{p}\left(\alpha_p \sum_{j=1}^{n} \gamma_{ji} z_j\right) + \eta_d R \tag{2a}$$

where z_j's are the principal components of the correlated RVs ΔP_p's in (1), and the γ's can be obtained from PCA. $R \sim N(0,1)$ in the above equation represents the random component of variations of all the process parameters lumped into a single term that contributes a total variance of η_d^2 to the overall variance of delay. Similarly, the leakage power for an individual gate can be expressed as

$$\text{Leakage} = \exp\left(V_{\text{nom}} + \sum_{i=1}^{p}\left(\beta_p \sum_{j=1}^{n} \gamma_{ji} z_j\right) + \eta_l R\right). \tag{2b}$$

The next section shows that these representations of delay and power allow for significant simplification in the joint timing and power analysis, which otherwise becomes computationally inefficient if the spatial correlation were maintained without such a unified approach.

III. Statistical Analysis

In this section, we first provide an overview of the statistical timing analysis in [7], which has been extended to consider both correlated and random components of variations. We then provide details of the approach for performing statistical leakage-power analysis. During timing analysis, the arrival time (AT) at each node is maintained in the same canonical form as the delay of the individual gates, which enables an efficient approach for the traversal of the timing graph. Similarly, during power analysis, the sum of leakage power is maintained in a canonical form as the leakage of different gates is summed.

A. Timing Analysis

The delay of each gate (for example, a) can be expressed as follows using the expression developed in the previous section:

$$D_a = a_o + \sum_{i=1}^{n} a_i z_i + a_{n+1} R. \tag{3}$$

This serves as the canonical expression for delay. The mean delay is simply the nominal delay (a_0). Since the principal components (z_i's) are uncorrelated $N(0,1)$ RVs, the variance of the delay can be expressed as

$$\text{Var}(D_a) = \sum_{i=1}^{n} \left(a_i^2\right) + a_{n+1}^2 \tag{4}$$

and the covariance of the delay with one of the principal components can be obtained as

$$\begin{aligned}\text{Cov}(D_a, z_i) &= E(D_a z_i) - E(D_a)E(z_i) \\ &= a_i^2 \qquad \forall i \in \{1, 2, \ldots, n\}.\end{aligned} \tag{5}$$

In [9], it is shown that delay distributions arising due to correlated reconvergent fanouts can be tightly upper bounded by assuming them to be independent. Since the random components are uncorrelated and do not contribute to the covariance of the delay at the two nodes at the inputs of a gate (e.g., "a" and "b"), the covariance can be obtained as

$$\text{Cov}(D_a, D_b) = \sum_{i=1}^{n} a_i b_i. \tag{6}$$

In deterministic timing analysis, the delay of the circuit is found by applying two functions to the delay of individual gates: sum and max. Similar functions for the canonical delay expressions (3) are defined as

$$\text{Sum}(D_a, D_b) = (a_o + b_o) + \sum_{i=1}^{n} (a_i + b_i) z_i + \sqrt{a_{n+1}^2 + b_{n+1}^2}\, R. \tag{7}$$

The max function of normally distributed RVs is not a strict Gaussian. References [7] and [19] have shown that the maximum of two Gaussian RVs can be closely approximated by another Gaussian. If

$$c = \max(a, b) \tag{8}$$

where a and b are Gaussian RVs, then the parameters of c, which is assumed to be Gaussian, can be obtained using expressions developed in [20]. This approach provides the mean and variance of c in terms of the mean and variance of a and b and their correlation coefficient. Reference [20] also develops expressions to evaluate the correlation of c with any other RV in terms of the correlation of the RV with a and b. In the spirit of [7] and [8], we assume that c can again be expressed in the same canonical form as a and b. To find the coefficients in the expression for c in canonical form, the mean, variances, and the correlation of c with the principal components are matched, giving

$$\begin{aligned} c_0 &= E\left(\max(a, b)\right) \\ c_i &= \mathrm{Cov}(c, z_i) = \mathrm{Cov}\left(\max(a, b), z_i\right) \qquad \forall i \in \{1, \ldots, n\} \\ c_{n+1} &= \left(\mathrm{Var}\left(\max(a, b)\right) - \sum_{i=1}^{n} c_i^2\right)^{1/2}. \end{aligned} \tag{9}$$

By modeling the random component, we can preserve the mean, variance, and correlations and avoid the need to scale the coefficients of the principal components to match the variance, which loses their correlation [7]. This technique was also used in [8] to consider random variations in timing analysis. To compute the max of more than two variables, the above technique is applied recursively.

Using the timing-analysis approach previously outlined, we can develop an expression for the delay of a circuit in terms of the RVs associated with process-parameter variations. In the next section, we develop an approach for gate-level statistical leakage-power analysis. The key in this step is to preserve the correlation in delay and power, which is achieved by using a similar principal-component-based approach with the same underlying RVs.

B. Power Analysis

We express the leakage power of each gate as a lognormal (exponential of a Gaussian) RV based on the power model discussed in Section II. The leakage power of the total circuit can then be expressed as a sum of correlated lognormal variables. This sum can be accurately approximated as another lognormal RV [22]. Reference [21] shows that the approximation performed using an extension of Wilkinson's method [22], based on matching the first two moments, provides good accuracy. The leakage power of an individual gate a is expressed as

$$P_{\mathrm{leak}}^{a} = \exp\left(a_0 + \sum_{i=1}^{n} a_i z_i + a_{n+1} R\right) \tag{10}$$

where the z_i's are principal components of the RVs, and the a_i's are the coefficients obtained using (2b). The mean and variance of the RV in (10) can then be computed as

$$E\left(P_{\mathrm{leak}}^{a}\right) = \exp\left(a_0 + \frac{1}{2}\sum_{i=1}^{n+1} a_i^2\right) \tag{11}$$

$$\mathrm{Var}(P_{\mathrm{leak}}^{a}) = \exp\left(2a_0 + \sum_{i=1}^{n+1} a_i^2\right) - \exp\left(2a_0 + \frac{1}{2}\sum_{i=1}^{n+1} a_i^2\right). \tag{12}$$

The correlation of the leakage of gate a with the lognormal RV associated with z_j is found by evaluating

$$E\left(P_{\mathrm{leak}}^{a} e^{z_j}\right) = \exp\left(a_0 + \frac{1}{2}\sum_{i=1, i\neq j}^{n+1} a_i^2 + (a_j + 1)^2\right) \qquad \forall j \in \{1, 2, \ldots, n\}. \tag{13}$$

Similarly, the covariance of the leakage of two gates (a and b) can be obtained by using

$$E\left(P_{\mathrm{leak}}^{a} P_{\mathrm{leak}}^{b}\right) = \exp\left((a_0 + b_0) + \frac{1}{2}\left(\sum_{i=1}^{n}(a_i + b_i)^2 + a_{n+1}^2 + b_{n+1}^2\right)\right). \tag{14}$$

We assume that the sum of leakage power can be expressed in the same canonical form as (10). If the RVs associated with all the gates in the circuit are summed in a single step, then the overall complexity of the approach is $O(n^2)$ [37] due to the size of the correlation matrix. Since the sum of two lognormal RVs is assumed to be a lognormal variable in the same form, we can use a recursive technique to estimate the sum of more than two lognormal RVs.

In each recursive step, we sum two RVs of the form in (10) to obtain another RV in the same canonical form. To find the coefficients in the expression for the sum of the RVs, we match the first two moments, as in Wilkinson's method, and the correlations with the lognormal RVs associated with each of the Gaussian principal components. We outline the steps in one of the recursive steps where we sum P_{leak}^{b} and P_{leak}^{c} to obtain P_{leak}^{a}

$$\begin{aligned} P_{\mathrm{leak}}^{a} &= \exp\left(a_0 + \sum_{i=1}^{n} a_i z_i + a_{n+1}\right) \\ &= \exp\left(b_0 + \sum_{i=1}^{n} b_i z_i + b_{n+1}\right) \\ &\quad + \exp\left(c_0 + \sum_{i=1}^{n} c_i z_i + c_{n+1}\right) \\ &= P_{\mathrm{leak}}^{b} + P_{\mathrm{leak}}^{c}. \end{aligned} \tag{15}$$

The coefficients associated with the principal components can be found using (11)–(14) and expressing the coefficients associated with the principal components as

$$a_i = \log\left(\frac{E\left(P^a_{\text{leak}} e^{z_i}\right)}{E\left(P^a_{\text{leak}}\right) E(e^{z_i})}\right) = \log\left(\frac{E\left(P^b_{\text{leak}} e^{z_i}\right) + E\left(P^c_{\text{leak}} e^{z_i}\right)}{\left(E\left(P^b_{\text{leak}}\right) + E\left(P^c_{\text{leak}}\right)\right) E(e^{z_i})}\right). \quad (16)$$

Using the expressions developed in [13], the remaining two coefficients in the expression for P^a_{leak} can be expressed as (17) and (18), shown at the bottom of this page. Having obtained the sum of two lognormals in the original canonical form, the process can be recursively repeated to compute the expression for the total leakage power of the circuit.

The timing- and power-analysis techniques outlined in this section can be used to efficiently estimate the individual probability distribution functions (pdfs) of delay and power. The correlation in delay and leakage power arising from the correlated components of variation can be easily estimated, since the correlated variations are expressed in terms of the principal components used to develop the expressions for both delay and power.

As will be shown in Section VI, the dependence of the variance of leakage power on the random component is very weak. This arises due to the fact that the random component associated with each gate is independent, and hence, the ratio of standard deviation to mean for the sum of these independent RVs is inversely proportional to the square root of the number of RVs summed [18]. This ratio does not reduce for strongly correlated RVs—therefore, if a large number of RVs are summed with both correlated and random components, then the overall variance is dominated by the variance of the correlated component. Hence, the correlation due to the random component, which is difficult to compute efficiently, is also insignificant and can be safely neglected.

IV. Yield Estimation

The parametric yield of a circuit, given the delay and power constraints, can be expressed as

$$Y = P\left(D \le D_0, P \le P_0\right) \quad (19)$$

which is the probability of the circuit delay and power being less than D_0 and P_0, respectively. Since delay and power are correlated, the yield cannot be simply computed by multiplying the probabilities of the two events separately. To express the yield as the probability of a bivariate Gaussian RV, we take the logarithm of the leakage-power constraint. Note that the joint distribution of the delay and the log of leakage power has a multinormal distribution, which follows from Theorem 1.

Theorem 1 [32]: Let the p-dimensional random vector X be distributed according to a multinormal distribution with mean vector Δ and covariance matrix Σ of rank p. If A is any $m \times p$ matrix of real numbers with rank $m \le p$, then the m-component vector $Y = AX$ is a multinormal random vector with mean vector $A\Delta$ and covariance matrix $A\Sigma A^{\mathrm{T}}$.

The correlation coefficient of the two Gaussian RVs in the yield equation can now be obtained using (6). We express the yield in terms of two standard $N(0,1)$ RVs N_0 and N_1 as

$$Y = P\left(N_0 \le \frac{D_0 - \mu_D}{\sigma_D}, N_1 \le \frac{\log\left(P_0\right) - \mu_{\log(P)}}{\sigma_{\log(P)}}\right). \quad (20)$$

Since correlation does not change under an affine transformation with positive multiplicative coefficients, the correlation between N_0 and N_1 remains the same as the correlation between delay and the log of leakage. One approach to evaluate this expression is to perform numerical integration of the joint pdf (jpdf) over the feasible region, but this makes the approach computationally inefficient. A lookup-table-based approach, although efficient, involves substantial inaccuracy due to the required interpolation, as noted in [23]. Hence, we adopt an analytical approach to estimate the yield, which makes the approach efficient and practical within a yield-optimization framework.

The feasible region defined using two correlated RVs (20) is transformed to a set of two uncorrelated RVs using the following transformation:

$$R_0 = N_0; \quad R_1 = \left(\frac{N_1 - \rho N_0}{(1-\rho^2)^{\frac{1}{2}}}\right). \quad (21)$$

This transformation maps the feasible region from a rectangle [Fig. 1(a)] to a triangle [Fig. 1(b)] when $\rho < 0$, which is the case of interest. Note that a set of uncorrelated RVs can also be obtained using PCA. However, PCA also attempts to capture the maximum variance of the RVs in the first component, which is not needed for our purposes. Therefore, we simply subtract the correlated component of N_0 from N_1 to obtain two uncorrelated RVs (R_0 and R_1). Theorem 1 implies that R_0 and R_1 (N_0 and N_1) have a joint bivariate normal distribution. Independence of R_0 and R_1 follows from the fact that uncorrelated components of a multinormal distribution are independent.

$$a_0 = \frac{1}{2}\log\left(\frac{\left(E\left(P^b_{\text{leak}}\right) + E\left(P^c_{\text{leak}}\right)\right)^4}{\left(E\left(P^b_{\text{leak}}\right) + E\left(P^c_{\text{leak}}\right)\right)^2 + \text{Var}\left(P^b_{\text{leak}}\right) + \text{Var}\left(P^c_{\text{leak}}\right) + 2\text{Cov}\left(P^b_{\text{leak}} P^c_{\text{leak}}\right)}\right) \quad (17)$$

$$a_{n+1} = \left[\log\left(1 + \frac{\text{Var}\left(P^b_{\text{leak}}\right) + \text{Var}\left(P^c_{\text{leak}}\right) + 2\text{Cov}\left(P^b_{\text{leak}} P^c_{\text{leak}}\right)}{\left(E\left(P^b_{\text{leak}}\right) + E\left(P^c_{\text{leak}}\right)\right)^2}\right) - \sum_{i=1}^{n} a_i^2\right]^{0.5} \quad (18)$$

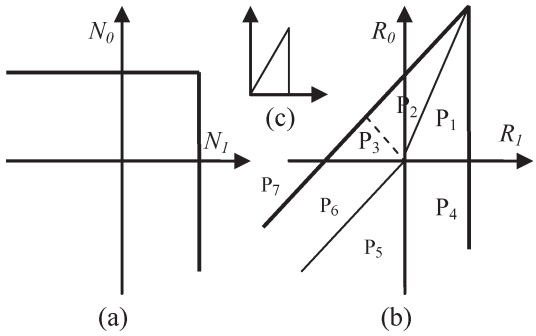

Fig. 1. Transformation of the feasible region from (a) to (b) using (21) for negative values of correlation.

Note that both the RVs of the bivariate distribution now have zero mean and unit variance. In addition, the pdf contours are circles rather than ellipses, simplifying the computation of the integral (which becomes circularly symmetric). The desired probability can now be obtained by using approximate expressions developed in [23] for evaluating probabilities of uncorrelated standard bivariate Gaussian RVs (unit variances and zero means) in regions of the form shown in Fig. 1(c), which can be expressed as

$$\frac{1}{2\pi}\int_0^a\int_0^{mx}\exp\left(-\frac{x^2+y^2}{2}\right)dydx \tag{22}$$

where a represents the base of the triangle, and m represents the slope of the inclined line forming the triangle. Note that a is allowed to be infinite in the above expression.

To evaluate the probability of the region shown in Fig. 1(b), we partition the figure as shown. The desired probability can then be expressed as a sum of the probabilities in Regions P_1–P_6, which can be evaluated as follows.

Region P_1: Already in the form required in [23].

Region P_2: Since the integral of the region is circularly symmetric, if the axes are rotated such that the dotted line, as shown in Fig. 1(b), lies along the x-axis then Region 2 is again in the same form as Fig. 1(c).

Region P_4: The probability in this region is

$$P(R_0 \le 0, 0 \le R_1 \le X) \tag{23}$$

where X is the point where the vertical line cuts the R_1 axis. Since R_0 and R_1 are statistically independent, this probability can be simply expressed as

$$P(R_0 \le 0)P(0 \le R_1 \le X) = 0.5\Phi(X) \tag{24}$$

where Φ is the normal integral from zero to x.

Regions P_3, P_5, and P_6: The probability for this region can be expressed as

$$P(R_0 \le 0, R_1 \le 0) + P(P_3 + P_6) - P(P_6 + P_7). \tag{25}$$

The first term can again be evaluated using the independence of R_0 and R_1. The second term in (25) corresponds to a region that has the same form as Region P_4 (after a rotation about the origin), and the region for the third term has the same form as Region P_1 (with an infinite base). Thus, the desired yield expressed in (19) can be efficiently estimated using closed-form expressions.

In terms of computational complexity, the proposed approach differs from [7] in the computation of an extra term associated with the random component. Thus, the overall complexity of the timing analysis remains $O(nN_g)$, where N_g is the number of terms in the delay expression that corresponds to the number of partitions into which the circuit is divided and n is the number of gates in the circuit. The power analysis is similar and requires an additional $O(nN_g)$ steps. The correlation computation requires an additional $O(N_g)$ steps, and the yield estimation runs in constant time. The computation of the principal components requires $O(pN_g^3)$ steps, where p is the number of correlated-process parameters considered in the analysis. The cubic dependence results from the eigenvector computation required during PCA. Since the principal components need to be calculated only once for each design being optimized, it does not impact the overall complexity, and hence, we do not include it in the overall complexity, which becomes $O(nN_g)$. The cost of PCA is amortized over the number of calls to the yield analysis engine during the optimization. However, if the standalone analysis engine is used for a new design partitioning each time, then the PCA overhead should be considered in the overall complexity.

Based on this yield-analysis engine, a brute-force approach to perform yield optimization using gate sizing and gate-length biasing can be developed. This involves computing the gradient of yield to the size of each gate, which can be estimated by perturbing the gate and performing yield analysis and setting the gate back to its original dimensions. After computing the gradient, we use a large-scale nonlinear optimizer to improve the yield of the circuit. We now consider the computational complexity of a single iteration of this approach. Each gradient computation requires $n+1$ yield-analysis runs and, thus, has an overall complexity of $O(n^2N_g)$. Since the size of the partitions is fixed, the number of partitions N_g can also be expected to increase with the size of the design. Thus, the overall computational requirements soon become untenable for large designs. In addition, note that the brute-force approach spends most of the time recalculating the same information for most of the circuit, motivating the need for an efficient gradient-computation approach.

V. Yield Optimization and Gradient Computation

Our yield-optimization problem is formulated as an unconstrained optimization problem where the objective function is the circuit yield, as defined in (19). The optimization is performed using a gradient-based nonlinear-optimization engine. However, a direct brute-force computation of yield gradient using a finite-difference approximation is computationally expensive and inefficient.

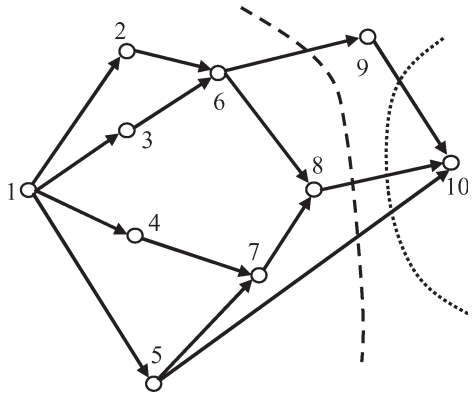

Fig. 2. Timing graph showing an LTO for the nodes and cutsets for nodes eight and nine.

In this section, we discuss our new gradient-computation approach that calculates the updated timing and delay pdfs based on a change in gate size or gate length. Both the timing- and power-perturbation analysis techniques update the coefficients of the delay and leakage pdf expression based on the change in gate size/length. These updated delay and leakage-power pdfs are then used to compute the yield of the perturbed design. The change in yield is used to estimate the gradient of yield to the gate size/length for each gate in the design.

A. Timing-Perturbation Computation

We first explain our timing-perturbation-computation approach based on cutsets, using the following graph representation for our circuits.

Definition 1: A timing graph is a directed acyclic graph having exactly one source and one sink: $G = \{N, E, n_s, n_f\}$, where $N = \{n_1, n_2, \ldots, n_k\}$ is a set of nodes; $E = \{e_1, e_2, \ldots, e_l\}$ is a set of edges; n_s is the source node; n_f is the sink node; each edge e is an ordered pair of nodes $e = (n_i, n_j)$; and each node is associated with a delay for each fanin edge, which depends on the characteristics of the fanout nodes.

The nodes in the timing graph correspond to gates, and the edges correspond to nets in a circuit. A probabilistic timing graph is defined as a timing graph where each node is associated with an RV for the delay for each fanin edge. Fig. 2 shows an example timing graph with ten nodes, eight of which represent actual gates and nodes, one and ten, which represent the source and sink nodes, respectively. The latest AT and required-AT (RAT) pdfs for each node in the timing graph are now defined as follows.

Definition 2: The latest AT at an edge e in the probabilistic timing graph is an RV whose cumulative distribution function (CDF) $A_e(t)$ gives the probability that a deterministic sample of this timing graph has an AT of less than t.

Definition 3: The earliest RAT at an edge e in the probabilistic timing graph is an RV whose CDF $R_e(t)$ gives the probability that the deterministic sample meets the timing constraint T_{crit} if the deterministic AT at the node is less than t.

Note that the sum of the AT and RAT at a node represents the partial pdf of delay, since it does not take into account the influence of the edges that are not present in either the fanin or the fanout cone of the node on the pdf of circuit delay. To express the dependence of circuit delay on the delay of one of the nodes, let us define the following terms.

Definition 4: A linear topological ordering (LTO) of the nodes in a timing graph is a total order based on the relationship that the order of any node $\boldsymbol{x}$ that lies in the fanout cone of a node n is strictly larger than the order of node n and that no two nodes in a timing graph have the same order.

An LTO of a timing graph can be easily determined by performing a topological traversal of the timing graph. Although a given timing graph can have many LTOs, finding the optimal LTO is not the focus of this paper. Fig. 2 illustrates a timing graph with nodes labeled according to an LTO of the timing graph. Note that swapping nodes eight and nine will still maintain a valid LTO of the nodes.

Definition 5: A cutset of a timing graph with a given LTO of a node n is defined to be the set of edges (n_i, n_j) of the timing graph that satisfy $\mathrm{LTO}(n_i) \leq \mathrm{LTO}(n)$ and $\mathrm{LTO}(n_j) > \mathrm{LTO}(n)$.

Definition 6: A node x of the timing graph belongs to the cutset source of node n if there exists an edge $(x, *)$ which belongs to the cutset of node n.

Definition 7: The fanin set of a node n of a timing graph is the set of immediate predecessor nodes of node n.

Definition 8: The AT set, or ATSet, of a node n is the union of the fanin set of node n and the nodes in the fanout cone of the fanin set of node n that have order less than or equal to the order of node n.

Definition 9: The convolution set, or ConvSet, of a node n is the intersection of the ATSet and cutset source of node n.

Any cutset of the timing graph divides the timing graph into two disconnected components, and the statistical maximum of the sum of the AT and RAT of all edges in the cutset gives the complete pdf of circuit delay. Now, if we perturb the delay characteristics of a node n (e.g., by gate sizing), we also change the capacitive loading of the fanin gates, affecting also their delay characteristics. To compute the new circuit delay, we note that the RAT of the edges in the cutset does not change, since all the gates in their fanout cone include gates that have an order strictly greater than the order of node n and have unchanged delay characteristics.

However, the AT for all edges that are in the fanout cone of the fanin set of node n changes. We are only interested in AT changes for edges that are driven by nodes that have an order less than the order of gate n, since we need to compute the AT for the edges in the cutset only. This is exactly the set of nodes defined by the ATSet of node n. If the AT of an edge in the cutset changes, we need to recompute the convolution of the AT and RAT at that edge. These edges are driven by the nodes in the intersection of the ATSet and the cutset, which is defined as the ConvSet of node n.

Let us revisit the example timing graph in Fig. 2 and consider node eight. The cutset for this node is the set of edges (6, 9), (8, 10), and (5, 10), as shown by the dashed line. The ATSet for the node can be identified as the set of nodes six, seven, and eight, as shown in Fig. 3. The intersection of the cutset source and ATSet defines the ConvSet and is the set of nodes six and eight. The ConvSet identifies that the AT and RAT have not

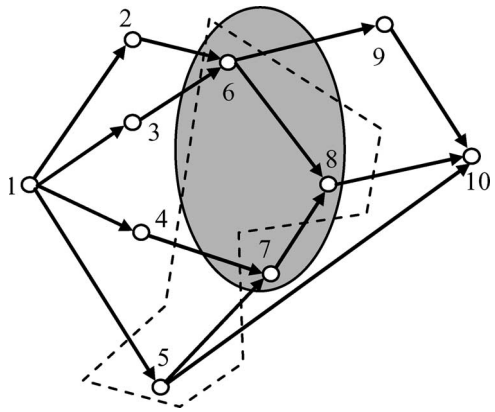

Fig. 3. Timing graph showing the ATSet (nodes within the shaded ellipse) and cutset-source set (nodes within the dashed shape) for node eight.

```
CUTSETSSTA (n)
  for each node (x ∈ ATSET (n))
    Compute AT(x);
  for each node (x ∈ CONVSET (n))
    CT(x) ← convolution (AT(x), RT(x))
  for each edge(x ∈ CUTSET (n))
      T_n ← maximum (T_n, CT(x))
  return T_n
```

Fig. 4. Pseudocode for the CutSetSSTA routine.

changed on the edge (5, 10), and we do not need to recompute the convolution of the AT and RAT for this edge. However, if we consider node nine, the cutset is defined by the edges from nodes five, eight, and nine to node ten, as shown as the dotted line in Fig. 3. The pseudocode of the "CutSetSSTA" routine to calculate the delay pdf of the perturbed circuit is shown in Fig. 4, where we refer to the edge by the name of the driving node. The pseudocode involves the computation of the AT for all nodes in the ATSet, convolution of the AT and RAT for all nodes in the ConvSet, and the statistical maximum of the convolution for all edges in the cutset.

All the computations in "CutSetSSTA" are performed using the same canonical expression for the delay pdf. Thus, the final delay pdf of the perturbed circuit is also expressed in the same form. Although the approach, as described, seems exact, it is heuristic. This results from the fact that the computation of the max function of delay pdfs is not exact, and forward and backward traversals of the graph result in timing delays that are not identical. However, this error is generally very small, as will be shown later in Section VI. The error depends on the relative difference in the mean, the variance, and the correlation of the RVs [45].

B. Power-Perturbation Computation

The statistical power computation is performed by summing the power dissipation of each gate in a circuit to compute the complete pdf of leakage power. To perform power analysis of a circuit with perturbations in the size of a gate, we first perform statistical power analysis of the unperturbed circuit, as described in Section III-B. Now, the leakage power after the size of gate i has been perturbed is expressed as

$$\begin{aligned} P_{\text{circ}}^{\text{pert}} &= P_{\text{circ}}^{\text{unpert}} - P_{\text{gate},i}^{\text{unpert}} + P_{\text{gate},i}^{\text{pert}} \\ &= P_{\text{circ}\setminus i}^{\text{unpert}} + P_{\text{gate},i}^{\text{pert}} \end{aligned} \tag{26}$$

where P_{pert} and P_{unpert} refer to the perturbed and unperturbed power, respectively, and the subscript indicates whether the power refers to the circuit or to the gate. Since the leakage power is expressed as a lognormal (exponential of a Gaussian) RV, we can approximate their sum using another lognormal. In general, we discussed in Section III-B that when we sum two lognormal RVs in canonical form, then the coefficients of the resulting expression can be obtained by matching the mean, variance, and the correlation coefficient with the exponential of the principal components (z_i's) leading to a set of $n+2$ equations in $n+2$ variables that can be analytically solved. To compute the expression in (26), we need to use (16)–(18) twice to calculate the final perturbed leakage. However, when one of the lognormals is subtracted, the signs associated with its expected value and covariance terms in the expressions are reversed.

C. Yield Gradient

To this point, we have developed efficient approaches to perform statistical timing and power-perturbation computations. Now, we will use these techniques to perform the computation of the gradient of yield in an efficient manner.

The computation of yield gradient involves the computation of the perturbation in yield for small changes in the size of gates in the design. The pseudocode for the computation of the yield gradient (FastYieldGradient) is shown in Fig. 5.

After each perturbation, the nonlinear optimizer calls the yield-gradient-computation routine "FastYieldGradient." The first step is to initialize the circuit so that all nodes are assigned the correct load capacitance based on the loading capacitance of the gates in its immediate fanout and the correct leakage power. Based on the load capacitance of the node, each input of a node is assigned to a delay pdf, which represents the delay of the timing arc from that particular input to the output of the gate. After the initialization step, the next step involves the propagation of the AT from the source node to the sink node in the timing graph using the timing-analysis approach described in Section III-A. This is represented as "ForwardSSTA" in the pseudocode. The next step is to perform statistical power analysis and generate the leakage-power pdf using the power-analysis steps from Section III-B. This is represented as "StatPowerAnalysis" in the pseudocode in Fig. 5. The "Yield" subroutine is then used to compute the yield based on the timing and leakage-power pdfs, given a leakage-power constraint P and a delay constraint D, as outlined in Section IV.

To perform the yield-gradient computation, we first propagate the RAT from the sink node to the source node using "ReverseSSTA." Details regarding the propagation of RATs from the sink nodes to the source nodes in a topological fashion

```
FastYieldGradient (Circuit, Size )
  for each gate (g ∈ Circuit)
     update load cap and size of (g) using
  Size;
  for each gate (g ∈ Circuit)
     compute new gate delay & power of (g)
  T ← ForwardSSTA (Circuit)
  P ← StatPowerAnalysis (Circuit)
  Y ← Yield (P, T)
  do ReverseSSTA (Circuit)
  for each gate(g ∈ Circuit)
     save current state of Circuit
     s+ ← Size(g)+ ΔSize(g)
     compute new gate delay & power of g
     for each gate(i ∈ FanIn(g))
        update load cap and delay of i
     T+ ← CutSetSSTA (Circuit)
     P+ ← IncremSPA (Circuit, P)
     Y+ ← Yield (P+, T+)
     ∇Y(g) ← (Y+ -Y)/ ΔSize(g)
     restore the original state of the Circuit
  return ∇Y
```

Fig. 5. Pseudocode for the yield-gradient-computation routine.

can be found in [7] and [9]. Then, we go through each node in the circuit iteratively and perturb the size or gate length of each gate by a small amount. Note that this step is repeated for each node in the timing graph. The load capacitance of the nodes in the fanin set of the node and the delay pdf assigned to each timing arc of this node and the nodes in the fanin set are updated. Then, using the statistical timing- and power-perturbation computation techniques discussed in this section, we compute the delay and leakage-power pdfs of this perturbed circuit. The yield corresponding to the perturbed circuit is then calculated, and the change in yield is used to define the particular component of the yield gradient. Note that the yield-gradient computation implicitly considers the impact of gate sizing or length biasing on both timing and power. The consideration of the correlation between timing and power in this paper effectively captures the power–performance relationship and guarantees overall yield of the design rather than providing separate guarantees for timing and power yields, as in prior work.

Let us consider the computational complexity of our proposed approach and compare it to the brute-force approach, where each iteration had a complexity of $O(n^2 N_g)$. Each iteration in our proposed approach involves a single run of the complete yield-analysis approach, as aforementioned, which has a complexity of $O(nN_g)$. The timing- and power-perturbation computation is repeated $O(n)$ times. The complexity of the incremental power analysis is $O(N_g)$, since we require two sum operations of the lognormal RVs. For the statistical timing-perturbation computation, most of the $\max$ computations in the cutset can be reused by storing the information as a binary-search tree. Thus, the timing-perturbation computation has a complexity of $O(N_g \log(n))$. The overall complexity of the approach is then $O(nN_g \log(n))$, which is a significant improvement over the brute-force approach.

VI. Results and Implementation Details

The proposed approach was implemented in C++, and we compared our analysis results to Monte Carlo (MC)-based simulations. The benchmark circuits were synthesized using an industrial 0.13-μm technology. We considered channel-length and gate-length-independent threshold-voltage variations for our experiments with 3 σ/μ of 20%. These variation levels are consistent with values in the literature [46], [47]; however, we note that the absolute value of variability is not critical in validating the proposed techniques. All variation in Vth_0 was assumed to be random, due to random-dopant effects, whereas half the variation in channel length was considered to be correlated. The gates in the library were characterized for delay and leakage power using SPICE simulations for different values of channel length (for gate-length biasing), which were fit to expressions of the form in (1) using linear regression. The circuits were placed using Cadence Silicon Ensemble and partitioned such that each square on the grid had a maximum dimension of 40 $\times$ 40 μm. The correlation coefficient among different squares on the grid was assumed to be inversely proportional to the distance between the centers of their grids. MC simulations were performed by generating correlated and random Gaussian RVs for the process parameters and performing timing and power analysis. For yield optimization, we compared our yield improvements to a deterministic circuit-optimization technique. All experiments were performed on an Intel 2.8-GHz Xeon processor with 3 GB of RAM.

A. Yield Analysis

Table II shows results for the ISCAS85 [24] and Microelectronics Center of North Carolina (MCNC) [25] benchmark circuits (i1 and c17 are excluded due to their very small sizes). The table compares the means and standard deviations of delay and power obtained using the proposed approach and MC-based simulations. The table also compares the coefficient of correlation of delay and the log of leakage power, which is required for yield estimation, as discussed in Section IV. The results show that the estimates obtained using the proposed approach for the values of the mean delay and leakage power are very accurate with an average error of 1.2% and 1.8%, respectively. The standard deviations show an average error of 7.6% and 13.7% for power and delay, respectively. A comparison of the direct $(O(n^2))$ leakage-analysis approach (described in Section III-B) and our recursive approximation technique was presented in [38]. Based on the results, the authors conclude that the accuracy of the two approaches is essentially the same.

We observe that circuits with smaller logic depth show larger error in delay as compared to circuits with larger logic depths. This results from the fact that the correlations in the random component are neglected, which can result in an error in estimated variance. The contribution of the random component

TABLE II
COMPARISON OF OUR PROPOSED APPROACH AND MC-BASED SIMULATION RESULT. SD = STANDARD DEVIATION

		Analytical					Error compared to Monte Carlo simulations						
Benchmark Circuit	Number of gates	Mean Delay (ns)	SD of delay (ns)	Mean Leakage (μW)	SD of Leakage (μW)	Correlation	Mean Delay	SD of delay	Mean Leakage	SD of Leakage	Correlation	Logic Depth	Runtime (sec)
c432	256	0.91	0.04	12.20	4.05	-0.91	0.5%	8.4%	0.1%	11.1%	2.0%	24	0.02
c499	544	0.89	0.03	36.14	10.29	-0.95	0.4%	11.4%	0.9%	8.8%	4.9%	24	0.12
c880	500	0.82	0.04	30.00	8.64	-0.88	1.1%	11.1%	1.1%	8.3%	4.9%	19	0.12
c1908	603	1.22	0.04	19.03	5.38	-0.95	2.1%	11.9%	1.0%	9.2%	4.0%	33	0.12
c2670	780	0.91	0.04	7.47	2.34	-0.93	1.7%	9.8%	1.7%	10.3%	0.7%	23	0.18
c3540	1163	1.43	0.06	57.54	14.70	-0.74	2.0%	10.4%	1.3%	6.4%	0.9%	40	0.56
c5315	1692	1.23	0.04	88.41	20.95	-0.87	1.4%	12.0%	2.7%	5.2%	4.1%	33	1.92
c6288	3834	3.32	0.11	116.73	25.38	-0.79	2.6%	13.4%	3.6%	5.3%	3.2%	94	15.43
c7552	2152	1.12	0.04	85.39	20.27	-0.86	0.6%	16.2%	2.5%	5.4%	8.8%	30	3.79
i2	192	0.47	0.02	4.53	1.46	-0.91	5.0%	20.1%	0.2%	10.1%	4.9%	10	0.02
i3	120	0.27	0.01	0.83	0.26	-0.89	1.0%	16.8%	1.0%	3.9%	3.6%	6	0.02
i4	264	0.38	0.02	11.34	3.73	-0.87	2.6%	20.6%	0.9%	8.1%	1.7%	10	0.04
i5	423	0.34	0.01	20.63	5.98	-0.88	1.0%	17.8%	0.0%	7.3%	8.7%	8	0.10
i6	461	0.31	0.01	13.90	4.58	-0.88	1.5%	10.4%	0.4%	9.0%	6.4%	8	0.08
i7	769	0.34	0.02	22.71	6.69	-0.65	3.1%	14.4%	0.6%	10.5%	3.6%	9	0.20
i8	1013	0.52	0.02	26.71	7.66	-0.94	1.5%	18.0%	1.4%	7.9%	6.6%	13	0.33
i9	723	0.53	0.02	24.47	7.91	-0.83	1.8%	13.4%	0.6%	3.0%	0.7%	14	0.16
i10	2482	1.41	0.05	49.55	12.59	-0.86	2.4%	11.1%	2.3%	7.4%	5.5%	39	5.00
Average		0.91	0.04	34.86	9.05	-0.87	1.8%	13.7%	1.2%	7.6%	4.2%		

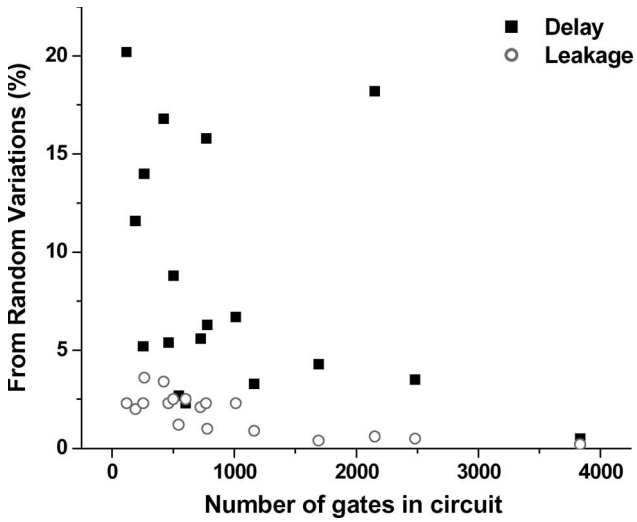

Fig. 6. Contribution of random variation to the total variation in delay and power.

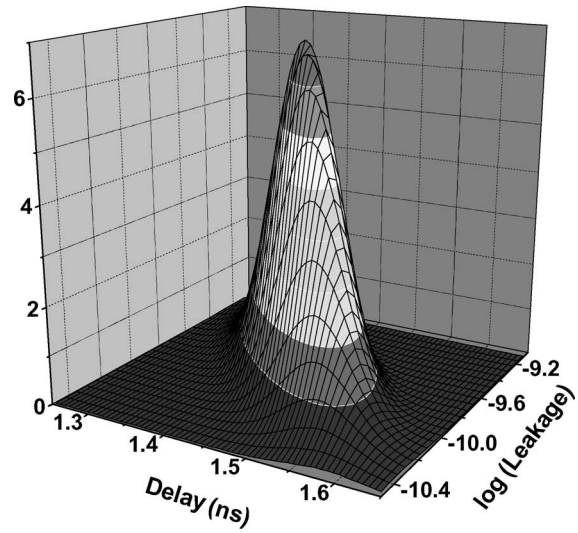

Fig. 7. JPDF for the bivariate Gaussian distribution for c3540.

is inversely proportional to the depth of the critical path [15] and can generally be expected to be small for larger circuits. The coefficient of correlation between the log of leakage power and delay shows a very good match to MC results with an average error of 4.2%.

Fig. 6 shows the contribution of the random component to the variance of delay and leakage power. Circuit-delay variance shows an average contribution of 8.4% with a maximum of 20.2%. On the other hand, variance in leakage power shows a much smaller average contribution of 1.8% with a maximum of 3.6%. This confirms our assumption that the impact of the random component of variation is negligible when estimating the correlation in power and performance. In addition, it is interesting to note that this random contribution reduces with increasing circuit size for leakage power; however, there is no clear trend in the case of circuit delay since it strongly depends on circuit topology.

Fig. 7 shows a representative jpdf of the log of leakage and delay, which is a bivariate Gaussian jpdf. The contours of the jpdf are ellipses with center at the mean of delay and the log of leakage.

Table III compares the yield estimates achieved using the proposed approach of Sections II–IV and those obtained using MC-based simulations for all benchmark circuits at two performance bins. For both bins, the leakage power is constrained to be less than $1.1\times$ the mean leakage value. Two different performance bins are constructed with delay being less than $1\times$ and between 1 and $1.1\times$ the mean delay. The proposed

TABLE III
YIELD ESTIMATES FOR DIFFERENT FREQUENCY BINS USING THE PROPOSED APPROACH AND MC-BASED SIMULATIONS. $D\mu$ REPRESENTS THE MEAN OF DELAY (D)

Benchmark Circuit	Monte Carlo		Analytical Yield		Yield Neglecting Correlation	
	D < Dμ	Dμ < D < 1.1 * Dμ	D < Dμ	Dμ < D < 1.1 * Dμ	D < Dμ	Dμ < D < 1.1 * Dμ
c432	0.17	0.43	0.14	0.46	0.31	0.30
c499	0.17	0.46	0.15	0.49	0.32	0.32
c880	0.20	0.43	0.16	0.46	0.32	0.31
c1908	0.18	0.48	0.14	0.49	0.32	0.32
c2670	0.16	0.44	0.14	0.47	0.31	0.30
c3540	0.22	0.43	0.20	0.44	0.33	0.32
c5315	0.19	0.48	0.19	0.48	0.33	0.33
c6288	0.22	0.47	0.21	0.46	0.34	0.34
c7552	0.20	0.47	0.19	0.47	0.33	0.33
i2	0.17	0.40	0.14	0.45	0.31	0.30
i3	0.17	0.40	0.15	0.45	0.31	0.30
i4	0.19	0.40	0.15	0.46	0.31	0.30
i5	0.18	0.45	0.16	0.47	0.32	0.31
i6	0.20	0.39	0.15	0.45	0.31	0.30
i7	0.22	0.38	0.21	0.40	0.32	0.30
i8	0.18	0.46	0.15	0.49	0.32	0.32
i9	0.19	0.44	0.17	0.45	0.31	0.31
i10	0.20	0.46	0.18	0.47	0.33	0.32
Average	0.19	0.44	0.17	0.46	0.32	0.31

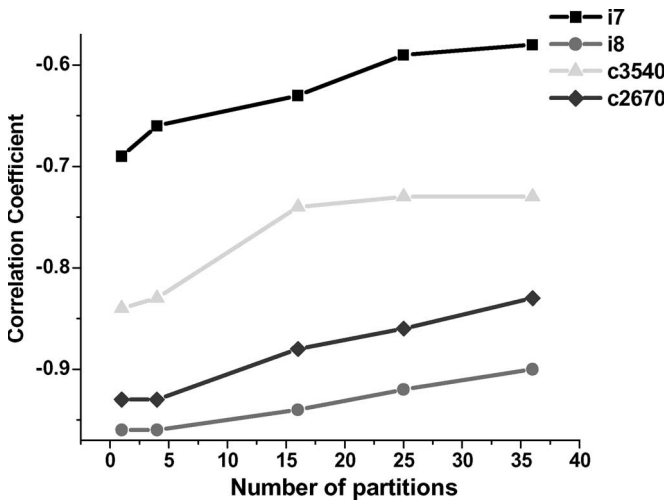

Fig. 8. Dependence of the correlation coefficient between delay and log (leakage) on the correlation structure of process parameters.

approach is seen to provide good estimates of the yield for the different frequency bins with an average misprediction in yield of 2%. If the correlation in power and delay is ignored, the yield in different bins can be both significantly overestimated (up to 15% in the high-performance bin) and underestimated (up to 16% in the low-performance bin), as shown in the last two columns of the table.

Fig. 8 shows the dependence of the correlation coefficient of delay and the log of leakage on the correlation structure. We selected two benchmark circuits from each of the ISCAS85 and MCNC benchmark suites and varied the correlation structure. The correlation structure was adjusted by changing the number of partitions while maintaining a fixed correlation coefficient

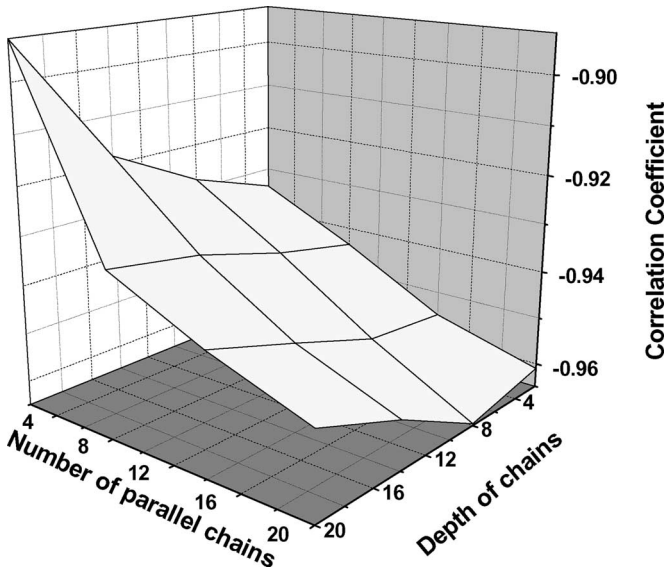

Fig. 9. Dependence of the correlation coefficient of delay and log (leakage) on the depth and the number of paths.

TABLE IV
COMPARISON OF YIELD-GRADIENT COMPUTATION USING FASTYIELDGRADIENT AND BRUTE-FORCE APPROACH. ALL RUNTIMES ARE IN SECONDS

Bench. Circuit	Average Cut-Size	Brute-Force	Fast Gradient	Speed-up	Max. Error (%)	Avg. Error (%)
c432	46.8	0.6	0.1	7.0	1.2	6.7E-03
c499	96.0	5.2	0.7	7.1	0.3	3.4E-03
c880	103.8	4.8	0.6	7.9	0.2	1.2E-02
c1908	84.3	6.5	0.7	9.8	2.6	1.7E-02
c2670	248.2	5.7	1.1	5.3	5.2	1.0E-03
c3540	140.6	41.9	3.5	12.1	1.0	1.5E-03
c5315	295.1	130.3	13.1	9.9	7.4	2.4E-03
c6288	297.9	991.3	51.0	19.5	1.8	1.4E-03
c7552	317.8	979.4	51.1	19.2	1.8	1.4E-03
i2	105.0	214.8	20.3	10.6	2.0	1.5E-03
i4	105.8	0.4	0.1	2.9	1.3	1.8E-02
i5	137.5	0.6	0.2	2.8	8.0	9.4E-02
i6	177.8	3.2	0.7	4.5	0.1	2.3E-02
i7	252.3	2.0	0.7	3.0	1.8	6.0E-02
i8	243.7	10.4	2.2	4.7	1.0	4.0E-02
i10	413.0	19.6	3.8	5.2	2.3	2.3E-03

across partitions. Thus, increasing the number of partitions results in a reduction in the correlation distance (the distance at which correlation falls to a given value). Note that the case of a single partition is equivalent to the case where all variations are inter-die variations. The results show that the correlation reduces in absolute value with increasing number of partitions. However, the various circuits behave differently when the number of partitions is increased from one value to another. To understand the dependence of the correlation coefficient on circuit topology, we created artificial circuits consisting of varying numbers of parallel inverter chains of varying depth. Results in Fig. 9 show that, as the depth of the circuits is increased, the absolute value of correlation reduces due to the increasing contribution of random variations to delay variance. As the number of parallel chains increases, the absolute value of correlation increases since the delay variance reduces [15].

TABLE V
COMPARISON OF THE PARAMETRIC YIELD ACHIEVED USING DETERMINISTIC AND STATISTICAL OPTIMIZATION

Bench. Circuits	Initial solution				D < Mean delay, P < Mean Power							Runtime (s)
	Delay (ps)		Power (μW)		Yield (%)			Delay (ps)		Power (μW)		
	Mean	Sigma	Mean	Sigma	Det.	Stat. (No Lgate)	Stat. (Lgate)	Mean	Sigma	Mean	Sigma	
c432	1040	33.03	7.66	0.495	3.40	45.44	80.18	960	30	3.6	0.33	14.0
c499	1112	34.4	7.86	1.002	3..56	39.21	59.03	1110	34.8	4.77	0.812	50.5
c880	1049	30.3	8.28	0.895	4.27	49.33	83.18	982	27.2	7.65	0.62	59.3
c1908	1314	45.5	3.43	1.099	2.78	47.85	82.77	1228	33.9	2.61	0.747	69.0
c2670	1112	37.2	11.6	1.49	6.11	51.07	85.31	1017	30.9	10.54	0.99	75.5
c3540	1438	48.8	18.6	2.05	8.62	51.18	87.1	1312	40.3	17.01	1.39	302.8
c5315	1530	53.01	26.17	2.94	7.91	50	87.27	1411	45.7	23.78	1.97	848.5
c6288	2683	81	59.18	6.5	6.59	50.29	86.49	2497	72.5	54.11	4.46	6280.0
c7552	1518	47.7	32.64	3.74	8.36	51.23	80.81	1383	41.87	30.25	2.74	1558.3
i2	725	25.71	3.5	0.367	8.64	51.47	82.78	659	23.28	3.254	0.261	7.8
i3	646	16.9	2.21	0.235	8.33	47.79	74.22	613.3	16.67	2.08	0.178	6.8
i4	749	27.17	3.92	0.504	8.21	48.66	84.33	673	24.2	3.56	0.336	18.0
i5	777	31.52	6.294	0.768	11.30	49.22	82	727	22.49	5.769	0.539	27.5
i6	925	30.94	7.236	0.873	6.24	31.36	59.85	885	31.2	6.89	0.704	21.8
i7	947	30.26	11.33	1.38	6.42	37.82	63.86	908	30.25	10.64	1.08	78.3
i8	1070	33.11	15.94	1.85	5.67	42.11	69.13	1022	33.27	14.91	1.38	171.3
i9	1020	34.65	11.2	1.33	8.37	50.55	84.52	942	26.53	10.26	0.92	71.0
i10	1434	43.68	38.11	4.4	4.75	51.34	86.81	1323	38.02	34.69	2.99	1793.5

B. Yield Optimization

Our proposed approach for the computation of the yield gradient is written as a subroutine that the optimizer uses to calculate the gradient of the objective function. The yield-analysis engine serves as the subroutine to calculate the objective function itself. If we assume that ReverseSSTA and ForwardSSTA give exact timing distributions at each node, then the procedure would be also exact. However, as noted earlier, the Gaussian approximation considered, while computing the maximum, introduces an inaccuracy while performing ForwardSSTA and ReverseSSTA. In practice, we find that the error introduced due to this inaccuracy is small. Table IV shows the runtime comparison and accuracy results of the proposed gradient-computation procedure as compared to the naïve brute-force approach. The average cut-width over all nodes in the circuit is reported in the second column. Runtime per gradient-vector computation (i.e., the runtime for gradient computation per descent step of the optimization) using the brute-force approach and the proposed procedure are given in Columns 3 and 4, respectively. The speed up of the proposed method over the brute-force approach is given in Column 5 and ranges between 3× to 20× and is found to be larger for bigger circuits. The maximum error over all gates, found using gradient computation normalized with respect to the brute-force method, is given in Column 6 and is found to be small in most cases with an average of 2.4%. As aforementioned, the error results from the fact that the computation of the max function of delay pdfs is not exact; forward and backward traversals of the timing graph result in timing delays that are not identical. The error averaged over all gates in the circuit is given in the last columns of Table V and is extremely small. The gates in our standard-cell library are characterized for a set of sizes and lengths that range from minimum to maximum sizes and the delay and leakage power for intermediate gate sizes is obtained using linear interpolation. All designs are then deterministically optimized for power under delay constraints using LANCELOT. We use our statistical yield-maximization approach to improve the yield of this already traditionally optimized design for a set of various power and timing constraints. Our results indicate that performing statistical optimization can significantly improve the timing yield of the design. We compare our results based on the ISCAS85 and MCNC benchmarks synthesized in a 0.13-μm technology.

The yield-optimization results are given in Table V. The first subsection, including Columns 2, 3, 4, and 5, report the initial timing and power statistics of the benchmark circuits. The next three columns report the parametric yield achieved using a deterministic (with gate sizing and gate-length biasing), statistical with no gate-length biasing, and statistical optimization with gate-length biasing. The deterministic optimization was performed using nominal delay and power models and minimizing the power dissipation while sweeping the delay constraint. The yield of the final optimized design is then analyzed, and the delay constraint that provides the maximum parametric yield is assumed to represent deterministic optimization. As a deterministic optimizer is unaware of the variation in power and timing and their correlation, it results in small yields. Statistical optimization using gate sizing provides significant improvement with an average yield increase of approximately 40%. With gate-length biasing, an additional yield increase of approximately 30% on average is obtained. By varying gate length within just a 10-nm range around the nominal length (125–135 nm), yield improvements are significant. As

discussed in [35], leakage has a strong dependence on gate length. We find that across the entire 10-nm range of our allowed lengths, the leakage varies by $2\times$ while the delay spread is only 10%. The large yield improvements due to length biasing can be attributed to the fact that, in addition to reducing the nominal leakage, increasing gate length greatly suppresses leakage variability [35]. This in turn reduces the susceptibility of the design to process variations. The last three columns list the delay and power statistics of the final optimized design using statistical optimization with gate-length biasing.

VII. Conclusion

To the best of our knowledge, we have presented the first approach to gate-level parametric yield analysis and optimization considering correlation between power and performance resulting from assorted components of process variability. The analysis approach is shown to be computationally efficient and forms the core of our yield-optimization technique, which relies on efficient yield-gradient computation.

The yield-gradient-computation technique exhibits reasonable computational complexity and is shown to provide an $8\times$ improvement in runtime on average as compared to a brute-force gradient-computation approach. The proposed analysis approach matches well with MC-based simulations with an average error of 4.2% in estimating power and delay correlation. Yield estimated using this approach is within 2% of MC results on average, and we demonstrated that neglecting the correlation in power and performance leads to gross mispredictions in yield. The parametric yield-optimization approach provides significant improvements in yield, and we show that the use of gate-length biasing is a strong lever in improving yield due to both its favorable leakage/delay tradeoff and its positive impact on process-induced leakage spread.

One possible avenue for future work involves extending the approach to handle fast path constraints. This would require the analysis of a trivariate normal distribution in contrast to the bivariate normal distribution addressed in this paper.

References

[1] A. Chandrakasan, W. J. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. Piscataway, NJ: IEEE Press, 2001.

[2] T. Karnik, S. Borkar, and V. De, "Sub-90 nm technologies challenges and opportunities for CAD," in *Proc. ACM/IEEE ICCAD*, 2002, pp. 203–206.

[3] X. Bai *et al.*, "Uncertainty aware circuit optimization," in *Proc. ACM/IEEE DAC*, 2002, pp. 58–63.

[4] S. Raj, S. Vrudhula, and J. Wang, "A methodology to improve timing yield in the presence of process variations," in *Proc. ACM/IEEE DAC*, 2004, pp. 448–453.

[5] S. Choi, B. Paul, and K. Roy, "Novel sizing algorithm for yield improvement under process variation in nanometer technology," in *Proc. IEEE/ACM DAC*, 2004, pp. 454–459.

[6] A. Srivastava, D. Sylvester, and D. Blaauw, "Statistical optimization of leakage power considering process variations using dual-Vth and sizing," in *Proc. ACM/IEEE DAC*, 2004, pp. 773–778.

[7] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *Proc. ACM/IEEE ICCAD*, 2003, pp. 621–625.

[8] C. Visweswariah *et al.*, "First-order incremental block-based statistical timing analysis," in *Proc. ACM/IEEE DAC*, 2004, pp. 331–336.

[9] A. Agarwal *et al.*, "Statistical timing analysis using bounds and selective enumeration," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 9, pp. 1243–1260, Sep. 2003.

[10] S. Bhardwaj, S. Vrudhula, and D. Blaauw, "TAU: Timing analysis under uncertainty," in *Proc. ACM/IEEE ICCAD*, Nov. 2003, pp. 615–620.

[11] A. Devgan and C. Kashyap, "Block-based statistical timing analysis with uncertainty," in *Proc. ACM/IEEE ICCAD*, 2003, pp. 607–614.

[12] M. Orshansky and A. Bandyopadhyay, "Fast statistical timing analysis handling arbitrary delay correlations," in *Proc. ACM/IEEE DAC*, 2004, pp. 337–342.

[13] R. R. Rao *et al.*, "Statistical analysis of subthreshold leakage current for VLSI circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 2, pp. 131–139, Feb. 2004.

[14] R. R. Rao *et al.*, "Parametric yield estimation considering leakage variability," in *Proc. ACM/IEEE DAC*, 2004, pp. 442–447.

[15] K. Bowman *et al.*, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 183–190, Feb. 2002.

[16] S. Duvall, "Statistical circuit modeling and optimization," in *Proc. IEEE Workshop Statistical Metrology*, 2000, pp. 56–63.

[17] S. D. Samaan, "The impact of device parameter variation on the frequency and performance of VLSI chips," in *Proc. ACM/IEEE ICCAD*, 2004, pp. 343–346.

[18] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1991.

[19] S. Tsukiyama, M. Tanaka, and M. Fukui, "A new statistical static timing analyzer considering correlations between delays," in *Proc. TAU*, Dec. 2002, pp. 27–33.

[20] C. Clark, "The greatest of a finite set of random variables," *Oper. Res.*, vol. 9, pp. 85–91, 1961.

[21] A. Abu-Dayya and N. Beaulieu, "Outage probabilities in the presence of correlated lognormal interferers," *IEEE Trans. Veh. Technol.*, vol. 43, no. 1, pp. 164–173, Feb. 1994.

[22] S. C. Schwartz and Y. S. Yeh, "On the distribution function and moments of power sums with lognormal components," *Bell Syst. Tech. J.*, vol. 61, pp. 1441–1462, Sep. 1982.

[23] J. H. Cadwell, "The bivariate normal integral" *Biometrika*, vol. 38, no. 3-4, pp. 31–35, Dec. 1951.

[24] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran," in *Proc. ISCAS*, May 1989, pp. 695–698.

[25] [Online]. Available: http://www.cbl.ncsu.edu

[26] R. Rao *et al.*, "Parametric yield analysis and constrained-based supply voltage optimization," in *Proc. ACM/IEEE ISQED*, 2005, pp. 284–290.

[27] S. Narendra, "Full-chip sub-threshold leakage power prediction model for sub-0.18 μm CMOS," in *Proc. ACM/IEEE ISLPED*, 2002, pp. 19–23.

[28] A. Agarwal *et al.*, "Statistical timing based optimization using gate sizing," in *Proc. ACM/IEEE DAC*, 2005, pp. 321–324.

[29] A. Davoodi, V. Khandelwal, and A. Srivastava, "Variability inspired implementation selection problem," in *Proc. ACM/IEEE ICCAD*, 2004, pp. 423–427.

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

[31] A. R. Conn, N. I. M. Gould, and P. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimizer (Release A)*. New York: Springer-Verlag, 1992.

[32] D. F. Morrison, *Multivariate Statistical Methods*. New York: McGraw-Hill, 1967.

[33] D. Lee, V. Zolotov, and D. Blaauw, "Static timing analysis using backward propagation," in *Proc. ACM/IEEE DAC*, 2004, pp. 664–669.

[34] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," in *Proc. IEEE ISPD*, 2006, pp. 2–9.

[35] P. Gupta *et al.*, "Selective gate-length biasing for cost-effective runtime leakage control," in *Proc. ACM/IEEE DAC*, 2004, pp. 327–330.

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

[37] H. Chang and S. Sapatnekar, "Full-chip analysis of leakage power under process variations, including spatial correlations," in *Proc. ACM/IEEE DAC*, 2005, pp. 523–528.

[38] H. Chang, "Circuit timing and leakage power analysis under process variation," Ph.D. dissertation, Univ. Minnesota, Minneapolis, MN, 2005.

[39] J. Fishburn and A. Dunlop, "TILOS: A polynomial programming approach to transistor sizing," in *Proc. IEEE/ACM ICCAD*, 1985, pp. 326–328.

[40] A. Wächter, C. Visweswariah, and A. R. Conn, "Large-scale nonlinear optimization in circuit tuning," *Future Generation Computer*

Systems—Special Issue Speedup/PARS Workshop in Basel, Germany, vol. 21, no. 8, pp. 1251–1262, Oct. 2005.
[41] A. R. Conn, I. M. Elfadel, W. W. Molzen, Jr., P. R. O'Brien, P. N. Strenski, C. Visweswariah, and C. B. Whan, "Gradient-based optimization of custom circuits using a static-timing formulation," in *Proc. ACM/IEEE DAC*, 1999, pp. 452–459.
[42] M. Berkelaar and J. Jess, "Gate sizing in MOS digital circuits with linear programming," in *Proc. Eur. Des. Autom. Conf.*, 1990, pp. 217–221.
[43] S. S. Sapatnekar, V. B. Rao, and P. M. Vaidya, "A convex optimization approach to transistor sizing for CMOS circuits," in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, 1991, pp. 482–485.
[44] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Math. Oper. Res.*, vol. 23, no. 4, pp. 769–805, Nov. 1998.
[45] H. Damerdji, A. Dasdan, and S. Kolay, "On the assumption of normality in statistical timing analysis," in *Proc. ACM/IEEE Int. Workshop Timing Issues*, 2005, pp. 2–7.
[46] K. Bernstein *et al.*, "High-performance CMOS variability in the 65nm regime and beyond," *IBM J. Res. Develop.*, vol. 50, no. 4/5, pp. 433–449, Jul./Sep. 2006.
[47] Y. Cao *et al.*, "Design sensitivities to variability: Extrapolations and assessments in nanometer VLSI," in *Proc. IEEE Int. ASIC/SOC Conf.*, 2002, pp. 411–415.

Ashish Srivastava received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 2001 and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 2003 and 2005, respectively.

He was with the TCAD Division, Intel Corporation, Hillsboro, OR. He is currently with Magma Design Automation, Austin, TX, where he is a Senior Member of Technical Staff. His research interests include optimization and circuit-analysis techniques applied to computer-aided design issues in very large scale integration design.

Dr. Srivastava was the recipient of the Intel Ph.D. Foundation Fellowship.

Kaviraj Chopra (S'99) received the B.E. degree in instrumentation and control from Gujarat University, Ahmedabad, India, in 2001 and the M.S. degree in electrical and computer engineering from the University of Arizona, Tucson, in 2004. He is currently working toward the Ph.D. degree in the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor.

He was with the National Optical and Astronomical Observatory, Tucson, during the winter and summer of 2003. He was a Summer Intern with IBM Corporation, Austin, TX, in 2005, and with Synopsys, Mountain View, CA, in 2006.

Saumil Shah (S'03) received the B.E. degree (with honors) in electrical engineering from Birla Institute of Technology and Science, Pilani, India, in 2003 and the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 2005, where he is currently working toward the Ph.D. degree.

In fall 2005, he was with Blaze DFM, Inc., Sunnyvale, CA, working as a Research Intern on leakage-power optimization and as a member of the Technical Staff. His research interests include leakage and variability analysis and reduction techniques for yield improvement.

Dennis Sylvester (S'95–M'00–SM'04) received the B.S. degree (*summa cum laude*) in electrical engineering from the University of Michigan, Ann Arbor, and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 1999, where his dissertation research was recognized with the David J. Sakrison Memorial Prize as the most outstanding research in the Electrical Engineering and Computer Science Department.

He has held research staff positions with the Advanced Technology Group, Synopsys, Mountain View, CA, and Hewlett-Packard Laboratories, Palo Alto, CA, and he was a visiting Professor of electrical and computer engineering with the National University of Singapore, Singapore. He is currently an Associate Professor of electrical engineering and computer science with the University of Michigan. He has published numerous articles along with one book and several book chapters in his field of research, which includes low-power circuit design and design-automation techniques, design-for-manufacturability, and interconnect modeling. He also serves as a Consultant and Technical Advisory Board Member for several electronic-design-automation firms in these areas.

Dr. Sylvester was the recipient of the National Science Foundation CAREER award, the Beatrice Winner Award at International Solid State Circuits Conference (ISSCC), the IBM Faculty Award, the Semiconductor Research Corporation Inventor Recognition Award, and several Best Paper Awards and nominations. He was also the recipient of the Association for Computing Machinery (ACM) Special Interest Group on Design Automation Outstanding New Faculty Award and the University of Michigan Henry Russel Award for distinguished scholarship. He has served on the technical program committee of numerous design-automation and circuit-design conferences, the steering committee of the ACM/IEEE International Symposium on Physical Design, and he was the General Chair for the 2005 ACM/IEEE Workshop on Timing Issues in the Synthesis and Specification of Digital Systems. He is currently an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN and previously served as Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He is a member of ACM, American Society of Engineering Education, and Eta Kappa Nu.

David Blaauw (M'01) received the B.S. degree in physics and computer science from Duke University, Durham, NC, in 1986 and the Ph.D. degree in computer science from the University of Illinois, Urbana, in 1991.

Until August 2001, he was with Motorola, Inc., Austin, TX, where he was the Manager of the High Performance Design Technology Group. Since August 2001, he has been on the Faculty with the University of Michigan, Ann Arbor, as an Associate Professor. His work is focused on very large scale integration design and computer-aided design with particular emphasis on circuit design and optimization for high-performance and low-power applications.

Dr. Blaauw was the Technical Program Chair and General Chair for the International Symposium on Low Power Electronic and Design and was the Technical Program Cochair and member of the Executive Committee at the Association for Computing Machinery/IEEE Design Automation Conference. He is currently a member of the ISSCC Technical Program Committee.