Fast Monte Carlo Estimation of Timing Yield With Importance Sampling and Transistor-Level Circuit Simulation

Alp Arslan Bayrakci, Alper Demir, *Senior Member, IEEE,* and Serdar Tasiran, *Member, IEEE*

***Abstract*—Considerable effort has been expended in the electronic design automation community in trying to cope with the statistical timing problem. Most of this effort has been aimed at generalizing the static timing analyzers to the statistical case. On the other hand, detailed transistor-level simulations of the critical paths in a circuit are usually performed at the final stage of performance verification. We describe a transistor-level Monte Carlo (MC) technique which makes final transistor-level timing verification practically feasible. The MC method is used as a golden reference in assessing the accuracy of other timing yield estimation techniques. However, it is generally believed that it can not be used in practice as it requires too many costly transistor-level simulations. We present a novel approach to constructing an improved MC estimator for timing yield which provides the same accuracy as standard MC but at a cost of much fewer transistor-level simulations. This improved estimator is based on a unique combination of a variance reduction technique, importance sampling, and a cheap but approximate gate delay model. The results we present demonstrate that our improved yield estimator achieves the same accuracy as standard MC at a cost reduction reaching several orders of magnitude.**

***Index Terms*—Importance sampling (IS), Monte Carlo (MC) method, statistical timing analysis, statistically critical paths, transistor-level simulation, yield estimation.**

I. Introduction

IN VLSI DESIGN methodologies, gate-level static timing analysis (STA) techniques have been widely used due to their desirable features such as linear computational complexity with circuit size and static nature as a result of not reliance on input vectors [1]. With current fabrication technologies in the nano-meter regime, the impact of process variations, especially intra-die variations, have become much more significant. This necessitated the development of techniques for accurate and meaningful modeling of statistical process variations in timing analysis. After the turn of the millennium, we have witnessed an extensive amount of effort being expended in statistical timing analysis research [1]. Most of this effort has been aimed at the development of statistical static timing analysis (SSTA) techniques, as a direct generalization of the STA algorithms to the statistical case. A comprehensive review of the recent developments in this field that puts all relevant work into perspective is given in [1]. At this point, the SSTA problem and its key challenges are very well understood. Most of the approaches to SSTA are based on what is referred to as the block-based scheme [1], which follows the STA algorithms quite closely. In these block-based SSTA methods, random variables (or their probability distributions) for latest node arrival times are propagated on an abstract timing graph of the circuit, as opposed to deterministic times that are propagated in STA. Block-based methods have been preferred due to their runtime advantage when compared with other approaches to SSTA. Moreover, block-based SSTA can be performed in an incremental manner enabling its use in timing yield optimizations and for diagnostic purposes [1], [2]. On the other hand, spatial and topological correlations, non-Gaussian process parameters and non-linear dependence of gate delay on these parameters, approximation of the maximum of random variables (for latest arrival times) at every node of the timing graph are issues that need to be addressed in block-based SSTA methods. In most basic form, SSTA algorithms ignore correlations, assume that all statistical process parameters and gate delays have a Gaussian distribution and approximate the maximum of two Gaussian random variables as another Gaussian random variable. All of these assumptions and simplifications make it possible to obtain very efficient SSTA algorithms [2]. However, ignoring correlations and the Gaussian assumption have detrimental, and in some cases, unacceptable, effects on the accuracy and meaningfulness of the results obtained by SSTA [3]. As a result, several extensions of SSTA that take correlations into account, that use non-linear gate delay models and employ non-Gaussian approximations for the maximum of two random variables have been proposed [1]. These extensions indeed improve the accuracy of SSTA, but at the same time increase its computational complexity and may render it unusable in timing optimizations which require very efficient in-the-loop evaluations [3]. Nevertheless, block-based SSTA on an abstract timing graph is widely accepted as a useful tool and is becoming indispensable in current state-of-the-art statistical design methodologies.

In traditional very large scale integration (VLSI) design methodologies, designers usually choose to perform transistor-

Manuscript received June 15, 2009; revised October 10, 2009 and January 31, 2010. Date of current version August 20, 2010. This work was supported in part by the Turkish Academy of Sciences Distinguished Young Scientist Award Program and in part by the two Scientific and Technological Research Council of Turkey (TUBITAK) Career Awards, under Grants 104E057 and 104E058. This paper was recommended by Associate Editor F. N. Najm.

The authors are with Koc University, Istanbul 34450, Turkey (e-mail: abayrakci@ku.edu.tr; aldemir@ku.edu.tr; stasiran@ku.edu.tr).

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.2010.2049042

level circuit simulations on the critical path(s) of a circuit that have been identified with deterministic STA, as a final verification before timing sign-off. One would ideally like to perform a similar transistor-level, but statistical, timing verification on the *statistically critical* path(s) that are possibly identified with a block-based SSTA method. Taiwan Semiconductor Manufacturing Company, a leading chip manufacturer, has already announced the insertion of transistor-level path-based statistical timing analysis into its new reference design flow in order to enhance timing accuracy [4].

Transistor-level path-based statistical timing analysis can be simply performed by running Monte Carlo (MC) transistor-level circuit simulations on the statistically critical paths of a circuit based on the same statistical process variations model used in SSTA. In conventional MC yield estimation, a number of samples in the parameter probability space are generated. The overall maximum delay for the statistically critical paths at each sample point in the probability space is determined by performing transistor-level circuit simulations. An estimator for timing yield is obtained by considering the fraction of samples for which the timing constraint is satisfied. Because of the high-computational cost of transistor-level simulations for each sample, it is generally believed that MC analysis can not be used in practice for estimating timing yield, even though there are some arguments to the contrary [5]. In order for MC analysis to be affordable, the number of samples in probability space one has to work with needs to be limited. This, however, adversely affects the accuracy of the MC yield estimator, which has a large error for a small number of samples. This is a weakness of the conventional MC method and has prevented it from finding widespread use for practical yield estimation, even though it is widely used as a golden reference in assessing the accuracy of other timing yield estimation techniques.

In this paper, we address the problem of estimating timing yield for a circuit under statistical process parameter variations and environmental fluctuations by proposing a novel and improved MC method that is based on transistor-level circuit simulations that are run on the statistically critical paths of a circuit. The technique we propose aims to improve the accuracy of the yield estimates obtained from a given number of MC simulations. Alternatively, our improved MC estimator achieves the same accuracy as the standard MC estimator, but at a cost of much fewer number of transistor-level circuit simulations. This is made possible by using a variance reduction technique called importance sampling (IS) that we combine in a novel manner with a cheap-to-evaluate but approximate gate delay model. We use the cheap gate delay model to guide the generation and selection of sample points in the parameter probability space in a transistor-level simulation-based MC method for timing yield estimation.

Our paper is definitely not the first one in the literature that uses IS or other variance reduction techniques in order to increase the efficiency, or improve the accuracy, of MC analysis of statistical phenomena in electronic circuits. In fact, IS-based MC analysis has been used in order to estimate the yield of analog circuits [6], perform failure analysis for SRAM circuits [7]–[9], for statistical interconnect analysis [10], and even for the statistical timing analysis of digital circuits [11]. The use of simple, cheap-to-evaluate gate delay models (linear, quadratic or more sophisticated response surface models) in statistical analysis is also prevalent in the literature [12]–[15]. Moreover, the idea of using path-based transistor-level analysis for statistical performance verification has also been explored. However, the challenge and key in using IS to achieve significant variance reduction is the non-costly determination of a useful biasing distribution. The technique we propose in this paper is novel in the sense that a cheap-to-evaluate gate delay model and approximate path-based statistical timing analysis are used in a unique way to (in effect) construct an effective biasing distribution for IS that indeed results in significant variance reduction/speed-up. Furthermore, an adaptive/automated algorithm we propose makes it possible to apply this IS technique in practice with negligible overhead. In [6], the outline and a simple analysis for an IS-like technique (called sectional weighting) that resembles the technique we propose in this paper was given. In [6], the authors are not very encouraging regarding the use of this technique due to the insignificant speed-ups (over standard MC) predicted by their simple analysis and due to the potentially high-computational cost of forming the biasing distribution. The computational complexity of the construction of the biasing distribution we propose in this paper is not dependent directly on the dimension of the random parameter space, resulting in negligible overhead. Moreover, we achieve significant (two-orders of magnitude) speed-ups over standard MC.

The approach proposed in this paper is based on the premise that, given the magnitude of process parameter variations and the non-linear dependence of gate and circuit delay on these variations, the only sufficiently reliable and accurate method for final timing yield verification before sign-off is transistor-level circuit simulation. However, we realize that transistor-level MC estimation of timing yield will never become efficient enough for use in a loop for timing optimizations. As such, the MC timing yield estimation technique based on transistor-level simulations we propose in this paper is meant not as a replacement for fast block-based SSTA methods, but rather, as a complement to them. In fact, in the timing analysis methodology we describe in this paper, the statistically critical paths on which we perform transistor-level MC analysis are identified using a fast block-based SSTA technique.

MC timing yield analysis can also be performed at a higher-level, in a block-based fashion on the abstract timing graph, in contrast with the low-level scheme we propose in this paper that is based on running transistor-level circuit simulations on a set of statistically critical paths. This cheaper block-based, gate-level MC timing analysis scheme is in fact used to verify the accuracy of various block-based SSTA methods which employ approximations for the distribution of random parameters and arrival times, gate delay dependence on parameters and the maximum operation. Recently, variance reduction techniques such as Latin hypercube sampling [16] were used to improve the efficiency of block-based, gate-level MC statistical timing analysis techniques. We believe that sufficient accuracy and reliability in final timing yield estimation can not be obtained even by applying MC simulations at a high level using a block-based scheme. We believe that accurate final verification of

timing yield must have transistor-level circuit simulation as its basis, in line with the common practice in traditional VLSI design where critical paths are simulated at transistor-level in order to verify that the circuit indeed satisfies the timing constraints. We demonstrate in this paper that MC transistor-level simulation in conjunction with a novel variance reduction technique can serve as an accurate yet computationally viable timing yield estimation method, to be used for final verification before timing sign-off.

In Section II, we provide background information and preliminaries on basic definitions of path and circuit delay, timing yield and loss, and the MC method and IS. In Section III, we describe a comprehensive timing yield estimation methodology complete with a statistical model for process and transistor parameters, a cheap-to-evaluate but approximate gate delay model, a technique to identify statistically critical paths, and finally, the main novel contribution of this paper, an IS based, improved MC timing yield estimation technique based on transistor-level simulations. In Section IV, we provide a precise, comparative convergence (error) analysis that reveals the speed-up obtained by our proposed estimator over the standard MC estimator. Finally in Section V, we present experimental results on the ISCAS'85 benchmark suite which demonstrate that our proposed yield estimator offers several orders of magnitude cost reduction while achieving the same level of accuracy as the standard MC estimator.

II. Background and Preliminaries

A. Basic Definitions and Notation

1) *Random Variables:* The random variables that represent the statistical variations in the circuit are collected into an n-dimensional vector X, with a joint probability density function (PDF) denoted by $f(X)$ which is *not* assumed to be Gaussian. We note here that the number of random variables, n, is dictated by the particular inter and intra-die variations model used and is in general much larger than the number of statistical process and transistor parameters considered. While we consider only two random transistor parameters for the work described in this paper, we employ hundreds of random variables in modeling the statistical variations of the circuit. We describe the inter and intra-die variations model we use later in this paper.

2) *Path and Circuit Delay:* We use $d_\pi^M(X)$ to denote the *path delay* for a *path* π computed by *method* M. We will be using two different methods for computing path delays, one that is based on an approximate but cheap gate delay model and another based on expensive but accurate transistor-level circuit simulations. The path delay naturally depends on the random variables in X, and hence, it is also a random quantity. We then define the *circuit delay* $d_C^M(X)$ computed by method M as the *maximum* path delay with

$$d_C^M(X) = \max_{\pi \in \Pi_{\text{crit}}} d_\pi^M(X) \tag{1}$$

where the maximum is computed over the set of *statistically critical* paths Π_{crit}. We describe later how we identify a set of statistically critical paths.

3) *Loss and Yield:* We define an indicator random variable $\mathrm{I}^M(T_c, X)$ as follows:

$$\mathrm{I}^M(T_c, X) = \begin{cases} 1 & \text{if } d_C^M(X) > T_c \\ 0 & \text{if } d_C^M(X) \le T_c \end{cases} \tag{2}$$

where M is the method used for circuit delay computation and T_c is the maximum acceptable delay or timing constraint. This indicator variable "indicates" whether the delay of the circuit meets the timing constraint for a given realization of the random variables in X. We then define *Loss* computed with method M using

$$Loss^M = \int_\Omega \mathrm{I}^M(T_c, X) f(X) dX \tag{3}$$

as the fraction of the circuits that fail to satisfy the timing constraint. The integral in (3), the expectation of the indicator variable $\mathrm{I}^M(T_c, X)$, is computed over the domain Ω of the PDF $f(X)$ of X. Then, *Yield*, the fraction of the circuits that fulfill the timing constraint is simply given by

$$Yield = 1 - Loss. \tag{4}$$

One very effective method for computing expectation integrals of the form in (3) is the MC technique, which we describe below.

B. Monte Carlo Method

MC techniques can be used to compute expectation integrals of the form

$$G = \int_\Omega g(X) f(X) dX \tag{5}$$

where Ω is the domain of the PDF $f(X)$, with $f(X) \ge 0$ for all X and $\int_\Omega f(X)dX = 1$. MC estimation of G in (5) is accomplished by drawing a set of independent random samples $X_1, X_2, ..., X_N$ from $f(X)$ and by using

$$G_N = (1/N) \sum_{i=1}^{N} g(X_i). \tag{6}$$

The estimator G_N above is itself a random variable. Its mean is equal to the integral G that it is trying to estimate, i.e., $E(G_N) = G$, making it an unbiased estimator. The variance of G_N is $Var(G_N) = \sigma^2/N$, where σ^2 is the variance of the random variable $g(X)$ given by

$$\sigma^2 = \int_\Omega g^2(X) f(X) dX - G^2. \tag{7}$$

The standard deviation of G_N can be used to assess its accuracy in estimating G. If N is sufficiently large, due to the Central Limit Theorem, $\frac{G_N - G}{\sigma/\sqrt{N}}$ has an approximate standard normal $(N(0, 1))$ distribution. Hence

$$P\left(G - 1.96\frac{\sigma}{\sqrt{N}} \le G_N \le G + 1.96\frac{\sigma}{\sqrt{N}}\right) = 0.95 \tag{8}$$

where P is the probability measure. The equation above means that G_N will be in the interval $[G - 1.96\frac{\sigma}{\sqrt{N}}, G + 1.96\frac{\sigma}{\sqrt{N}}]$ with 95% confidence. Thus, one can use the error measure

$$|Error| \approx \frac{2\sigma}{\sqrt{N}} \tag{9}$$

in order to assess the accuracy of the estimator.

Several techniques exist for improving the accuracy of MC evaluation of expectation integrals. In these techniques, one tries to construct an estimator with a reduced variance for a given, fixed number of samples, or equivalently, the improved estimator provides the same accuracy as the standard MC estimator but with considerably fewer number of samples. This is desirable because computing the value of $g(X_i)$ is typically computationally or otherwise costly.

C. Importance Sampling

One MC variance reduction technique is IS [17], [18]. IS improves upon the standard MC approach described above by drawing samples for X from another distribution $\tilde{f}(X)$. G in (5) is first rewritten as below

$$G = \int_{\Omega} \left(\frac{g(X) f(X)}{\tilde{f}(X)} \right) \tilde{f}(X) dX. \tag{10}$$

If $X_1, X_2, ..., X_N$ are drawn from $\tilde{f}$ instead of f, the improved estimator $\tilde{G}_N$ takes the form

$$\tilde{G}_N = \frac{1}{N} \sum_{i=1}^{N} g(X_i) \frac{f(X_i)}{\tilde{f}(X_i)} \tag{11}$$

where the factor $f(X_i)/\tilde{f}(X_i)$ has been used in order to compensate for the use of samples drawn from the biasing distribution $\tilde{f}$. In order for the improved estimator above to be well-defined and unbiased, $\tilde{f}(X_i)$ must be nonzero for every X_i for which $f(X_i)g(X_i)$ is nonzero. We refer to this as the *safety requirement*. The ideal choice for the biasing distribution $\tilde{f}$ is

$$\tilde{f}_{\text{ideal}}(X) = \frac{g(X)\, f(X)}{G} \tag{12}$$

which results in an exact estimator with zero variance with a single sample! However, $\tilde{f}_{\text{ideal}}$ obviously can not be used in practice since the value of G is not known a priori. Instead, a practically realizable $\tilde{f}$ that resembles $\tilde{f}_{\text{ideal}}$ is used. The key (and also the challenge) in using IS in practical problems is the determination of an effective biasing distribution that results in significant variance reduction.

III. Timing Yield Analysis Methodology

The comprehensive timing yield estimation methodology we propose in this paper features the following.

1) [*Section III-A*]: Modeling of inter- and intra-die statistical variations based on a quad-tree model that captures spatial correlations.
2) [*Section III-B*]: An approximate, polynomial gate delay model that captures delay dependence on random transistor parameters, gate load and input slope.
3) [*Section III-C*]: Identification of a set of statistically critical paths for a circuit, based on a MC block-based SSTA analysis that uses the polynomial gate delay model above and a path sensitization test to identify false paths.
4) [*Section III-D*]: For comparison purposes, a description of transistor-level MC determination of timing yield without IS.
5) [*Section III-E*]: Fast, accelerated MC estimation of circuit timing yield based on the set of statistically critical paths identified above and accurate transistor-level circuit simulations, using an IS technique which also utilizes the polynomial gate delay model.
6) [*Section III-F*]: Our automated algorithm for IS-based MC determination of circuit. yield.

The main, novel contribution of the work described in this paper is in devising a unique IS scheme for accelerating timing yield computations based on transistor-level MC simulations, which fills a gap in statistical design methodologies and enables final transistor-level verification and timing sign-off. However, in order to demonstrate the effectiveness of our proposed technique, we have developed, and describe, a comprehensive timing yield estimation methodology. While some elements of this methodology are borrowed from previous paper (such as the quad-tree model for capturing spatial correlations in modeling intra-die variations) and not necessarily the most comprehensive implementations, our paper addresses some other important open problems (such as statistically critical path identification) and offers reasonable and practical solutions. We now describe the elements that make up our methodology in more detail.

A. Modeling Process and Transistor Parameter Variations

In this section, we present the statistical model we use for inter and intra-die variations in process and transistor parameters. The inter-die variations are perfectly spatially correlated throughout the circuit. In order to model intra-die variations and the resulting (partial) spatial correlations in the circuit, we use the quad-tree model that was proposed by Agarwal *et. al.* [19]. In this model, a statistical process or transistor parameter P such as channel length is expressed as follows:

$$P = P_{\text{inter}} + \sum_{q=1}^{Q-1} R_{\text{intra}\,q}(x, y) \tag{13}$$

where the random variable P_{inter} models the perfectly correlated inter-die variations, $R_{\text{intra}\,q}(x, y)$ are layout position (x, y) dependent random variables that are assigned to level q of the quad-tree model, and Q is the total number of levels in the model for both intra and inter-die variations. In most previous paper, P_{inter} and $R_{\text{intra}\,q}(x, y)$ are assumed to be independent random variables with a Gaussian distribution. In our approach, the basic statistical process and transistor parameters and the random variables in (13) can have arbitrary (joint) PDFs.

In a Q-level quad-tree model $\sum_{q=1}^{Q} 2^{2(q-1)} = \frac{4^Q-1}{3}$ random variables are needed for every basic process or transistor parameter. In this paper, we consider two basic statistical parameters: the channel length and the threshold voltage. We use four levels in the quad-tree model including a top level covering the whole area of the circuit with one grid rectangle. As a result, for every random process or transistor parameter we use $\frac{4^4-1}{3} = 85$ random variables.

It should be emphasized that the computational cost of the technique we propose is very weakly dependent on the dimension of the random variable vector X, as is the case with all MC integral computation techniques.

B. Gate Delay Model

In the timing yield estimation methodology being proposed in this paper, an approximate but cheap (in terms of evaluation cost) gate delay model is used as the key tool in devising an effective biasing distribution for IS in a unique manner to accelerate MC yield analysis. In previous paper, we have employed a stochastic version of the logical effort gate delay model for this purpose [20]. In this paper, we use a polynomial gate delay model (PDM) that uses third-degree polynomials to express the delay and the output slope as a function of the random process and transistor parameters, input slope and load (fanout) of the gate. This polynomial gate delay model requires more computational resources to construct (but still very cheap to evaluate), but it is more accurate than the logical effort delay model and results in a much more effective biasing distribution for IS.

If the channel length L and threshold voltage V_t are considered as the random transistor parameters, then the delay and the output slope of a gate r can be represented with

$$d_r^{PDM}(L_r, V_{tr}, F_r, InS_r) \tag{14}$$

and

$$OutS_r^{PDM}(L_r, V_{tr}, F_r, InS_r) \tag{15}$$

where L_r and V_{tr} are the random parameters for the transistors in gate r, F_r is the fanout, and InS_r is the input slope. $OutS_r^{PDM}$ is the output slope and d_r^{PDM} is the delay of gate r computed by PDM.

Using this model, the delay of a path π with k gates in a circuit can be easily computed as follows. First, given the input slope of the first gate in the path (dictated by a primary input), the input slopes of all the other gates are computed using (15) and

$$InS_{i+1} = OutS_i^{PDM}(L_i, V_{ti}, F_i, InS_i), \quad i = 1, \ldots, k-1. \tag{16}$$

Then, the delay of the path is computed with

$$d_\pi^{PDM}(X) = \sum_{i=1}^{k} d_i^{PDM}(L_i, V_{ti}, F_i, InS_i) \tag{17}$$

where X is the vector that collects all of the random variable realizations used in the quad-tree model. The transistor parameters L_i and V_{ti} are computed using X and (13).

The polynomial delay models need to be constructed for the standard cell library that is being used. Delay look-up models for gates similar to the ones described above are routinely constructed in standard cell characterizations. These delay models have traditionally been used for STA. The delay model extraction needs to be done only once for a standard cell library for a given fabrication process. In order to construct the gate delay and output slope models for the gates in our library, we run SPICE simulations at suitably chosen sample points and fit third-order polynomials to the simulation data using a least-squares technique. For the results presented in this paper, delay models were constructed with SPICE simulations run per gate at 1700 sample points in the parameter space. These 1700 sample points were generated as follows. For the two random parameters considered (L and V_t in this paper), 425 sample points were placed non-uniformly in the rectangle in the L-V_t plane bounded by $\mu - 3.\sigma$ and $\mu + 3.\sigma$ for each parameter, where μ is the mean and σ is the standard deviation of the parameter. The sampling frequency was three times higher in the center $\mu - \sigma$ to $\mu + \sigma$ interval. Only two samples (values) for both input slope and load were used due to almost linear dependence of delay on these parameters. As a result, we end up with $425 \times 2 \times 2 = 1700$ points at which SPICE simulations are run. We should point out that the parameter space sampling scheme described here for fitting and building the gate delay model is only rudimentary and was considered adequate for the results we present in this paper. If a larger number of random transistor parameters are included in the gate delay model, a more efficient sampling scheme that does not have exponential complexity, such as Latin hypercube sampling [21], needs to be employed. Efficient and effective design of experiments [21] (selection of sample points in the parameter space) in statistical model fitting is a well-studied problem in statistics and beyond the scope of this paper.

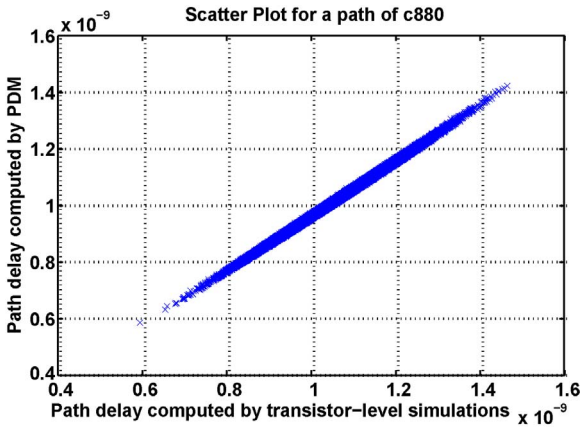


Fig. 1. Accuracy of polynomial delay model.

In Fig. 1, a scatter plot that shows the accuracy of the polynomial delay model against SPICE simulations is presented. In order to generate the graph in Fig. 1, the delay of a complete path in a circuit (c880 in the ISCAS'85 benchmark suite) was determined both by transistor-level circuit simulations and by evaluating the polynomial gate delay model at a number of sample points in the parameter space. This plot has 50 000 points that correspond to random realizations for the two transistor parameters, which do not include the set of 1700 points that were used for fitting the polynomial model. The polynomial delay models capture the trends and relative variations in delay as a function of the transistor parameters quite accurately. However, the delay model is not accurate enough to replace transistor-level simulation in predicting timing yield with sufficient accuracy. We use this model in order to construct an effective biasing distribution to be used in IS, but not as a replacement for transistor-level simulation in accurately determining the delay of a circuit.

Gate delay models are utilized in almost all statistical timing analysis methodologies. The nature of the algorithms used in statistical analysis may impose restrictions on the

complexity and form of these models. For instance, in block-based statistical timing analysis schemes based on PDF algebra/propagation, linear or at most quadratic models are used in order to make the PDF computations tractable and practical. In our methodology, the only requirement on the delay model is that it be cheap to evaluate. Otherwise, there is no restriction on the complexity (can use higher-order polynomials) or form (not restricted to polynomial models) of the delay model. A more complex delay model may result in a larger construction cost, but again, this is done only once for a gate library for a given process. The ability to use more accurate and complex gate delay models is one of the key benefits of our methodology.

C. Identifying Statistically Critical Paths

In non-statistical design flows, critical path(s) for the circuit are identified and simulated at transistor-level as a final verification that the circuit satisfies the timing constraint. When statistical variations are considered, the set of *statistically critical* paths, i.e., paths that are critical for some assignment to random parameters and for some primary input transition, must be considered.

In order to identify a set of statistically critical paths, we proceed as follows. We first run block-based, traditional STA on the abstract timing graph of the circuit. However, we perform this analysis many times in a MC manner by generating many samples for the random variables X from the joint PDF $f(X)$. For each sample, we collect the critical and nearly critical paths into a set of statistically critical path candidates. In performing STA on the circuit, we use the cheap polynomial gate delay model described in the previous section. For this purpose, block-based SSTA methods (e.g., in [22], [23]) could have been used as well. Next, we apply a static sensitization test [24] to eliminate false paths. Sensitizable paths are identified using a satisfiability solver, and included in the set Π_{crit}. We chose not to explore more elaborate techniques for identifying sensitizable paths, as this is not the focus of this paper.

D. Transistor-Level Monte Carlo Without Importance Sampling

As reviewed in Section II-B, the standard N-sample MC estimator for $Loss$ defined by (3) is given by

$$Loss_N^{TL} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{I}^{TL}(T_c, X_i) \tag{18}$$

where the superscript $(\cdot)^{TL}$ indicates that the value of indicator random variable defined by (2) is computed based on transistor-level (TL) simulations, that is, the path delays in (1) and hence, the circuit delay in (2) are computed with TL simulations. In (18) above, the N samples for the random variables, X_i, $i = 1, \ldots, N$, are drawn from the joint PDF $f(X)$. For every sample X_i, the random parameters for the transistors in the circuit are first computed according to the random variation model described in Section III-A, then a TL simulation with SPICE is performed for each path in the set of statistically critical paths Π_{crit} (obtained as described in Section III-C) to compute the path delays $d_\pi^{TL}(X)$, finally, (1), (2), and (18) are used to compute the loss estimate $Loss_N^{TL}$.

The MC loss estimator described above will result in accurate yield estimation results, because it is based on TL simulations as opposed to an approximate gate delay model, and the maximum operation in (1) is *not* approximated in any manner. However, the standard MC estimator typically requires too many samples (N) to converge. For each sample, one needs to perform TL simulations for all of the statistically critical paths, and hence, the computational cost of the standard estimator could become prohibitive for practical use.

E. Importance Sampling-Based Estimation of Timing Yield

We now describe the novel contribution of this paper: an improved loss estimator which is based on IS that significantly accelerates the convergence of the MC estimator *without* forfeiting accuracy and enables its use in practice. The IS-based MC estimator for $Loss$

$$Loss_N^{IS} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{I}^{TL}(T_c, X_i)\frac{f(X_i)}{\tilde{f}(X_i)} \tag{19}$$

draws the samples X_i from another, biasing distribution $\tilde{f}$. We propose the following biasing distribution to be used in the IS estimator above

$$\tilde{f}(X) = \frac{\mathrm{I}^{PDM}(T_c^\epsilon, X) f(X)}{Loss^{PDM,\epsilon}} \tag{20}$$

where the loss estimate $Loss^{PDM,\epsilon}$ and $\mathrm{I}^{PDM}(T_c^\epsilon, X)$ are computed based on the approximate but cheap gate delay model described in Section III-B, without performing any TL simulations. In (20) above, the target delay is set to $T_c^\epsilon = T_c - \epsilon$ where ϵ is a margin parameter. This margin parameter is introduced in order to guarantee that $\tilde{f}(X_i)$ is nonzero everywhere $\mathrm{I}^{TL}(T_c, X_i) f(X_i)$ is nonzero, i.e., $\mathrm{I}^{PDM}(T_c^\epsilon, X_i)$ must take the value 1 everywhere $\mathrm{I}^{TL}(T_c, X_i)$ is 1. The margin parameter ϵ must be large enough so that the indicator variables never assume the values $\mathrm{I}^{PDM}(T_c^\epsilon, X_i) = 0$ (the timing constraint T_c^ϵ is satisfied according to the PDM model) and $\mathrm{I}^{TL}(T_c, X_i) = 1$ (the actual circuit fails to satisfy the timing constraint according to TL simulations) for any of the sample points, X_i. This condition is called the *margin condition*. We note here that the safety requirement described in Section II-C dictates that the margin condition described above be satisfied. In the next section, we present an algorithm for computing $Loss_N^{IS}$. As this algorithm explores a set of sample points, it also gathers the data required for computing a value of ϵ that satisfies the margin condition. For ease of exposition, we continue the mathematical presentation of our method as if ϵ is determined first, before computing $Loss_N^{IS}$. In reality, the algorithm carries out the $Loss_N^{IS}$ computation and ϵ determination concurrently.

Substituting the biasing distribution $\tilde{f}$ in (20) into (19), and performing some simplifications based on the fact that $\mathrm{I}^{PDM}(T_c^\epsilon, X_i)$ takes the value 1 for all samples drawn from $\tilde{f}(X)$, we arrive at a simplified form of the IS estimator

$$Loss_N^{IS} = \frac{Loss^{PDM,\epsilon}}{N}\sum_{i=1}^{N} \mathrm{I}^{TL}(T_c, X_i) \tag{21}$$

where the samples X_i are drawn from $\tilde{f}(X)$ in (20).

In (21), the loss estimate

$$Loss^{PDM,\epsilon} = \int_{\Omega} \mathrm{I}^{PDM}(T_c^{\epsilon}, X) f(X) dX \tag{22}$$

is computed based on the approximate but cheap gate delay model described in Section III-B, without performing any TL simulations, but again using a MC estimator as follows:

$$Loss_K^{PDM,\epsilon} = \frac{1}{K} \sum_{i=1}^{K} \mathrm{I}^{PDM}(T_c^{\epsilon}, X_i) \tag{23}$$

for which one can afford to use a very large number of samples K, since the evaluation of $\mathrm{I}^{PDM}(T_c^{\epsilon}, X)$ for every sample is very cheap based on the approximate delay model.

In evaluating the IS estimator, in order to draw a sample from $\tilde{f}(X)$ in (20), we first draw a sample X_i from $f(X)$. We keep the sample if $\mathrm{I}^{PDM}(T_c^{\epsilon}, X_i)$ evaluates to 1 at the sample point and discard it otherwise. Again, the evaluation of $\mathrm{I}^{PDM}(T_c^{\epsilon}, X_i)$ is performed cheaply based on the delay model. Each kept sample constitutes one of the X_i in (21). $Loss_N^{IS}$ is then computed by determining whether $\mathrm{I}^{TL}(T_c, X_i) = 1$ for each such kept sample, i.e., by performing TL SPICE simulations.

F. CompLossMC-IS: Adaptive Detection Algorithm for Margin

This section presents CompLossMC-IS, the algorithm for determining $Loss_N^{IS}$ as described by (21). To accomplish this, CompLossMC-IS first generates a set of NS sample points $\mathcal{X} = \{X_1, X_2, ..., X_{NS}\}$ from the distribution f. The choice of NS will be discussed later in this section. Let Y_i be these sample points in decreasing order of d_C^{PDM}, i.e., $\mathcal{X} = \{Y_1, Y_2, ..., Y_{NS}\}$ such that $d_C^{PDM}(Y_i) \geq d_C^{PDM}(Y_j)$ if $i < j$. Using the sample set $\mathcal{X}$, CompLossMC-IS must compute:

1) the subset $\mathcal{W} = \{Y_1, Y_2, ..., Y_N\} \subseteq \mathcal{X}$ consisting of all sample points for which $\mathrm{I}^{PDM}(T_c^{\epsilon}, Y_i)$ evaluates to 1 (using the gate delay model);
2) the subset $\mathcal{Q} \subseteq \mathcal{W}$ of sample points for which $\mathrm{I}^{TL}(T_c, Y_i) = 0$ (by performing TL simulations);
3) the set $SafeMargin = \{Y_{N+1}, Y_{N+2}, ..., Y_{N+SM}\}$; (to be defined below) and the corresponding value of ϵ;
4) using ϵ above, the value of $Loss^{PDM,\epsilon}$ as in (23).

Then, the loss estimate $Loss_N^{IS}$ will be computed as

$$Loss_N^{IS} = Loss^{PDM,\epsilon} . \frac{|\mathcal{W} - \mathcal{Q}|}{|\mathcal{W}|}.$$

The first factor on the right hand side is the IS biasing factor, and the second factor is the fraction of points in $\mathcal{W}$ which result in a loss value.

The only non-straightforward task that the algorithm must carry out is the determination of the margin parameter ϵ. ϵ uniquely determines $\mathcal{W}$, $\mathcal{Q}$, $Loss^{PDM,\epsilon}$, and thus $Loss_N^{IS}$. The requirements on ϵ are discussed next.

Constraints on ϵ: For IS to provide an unbiased estimator in our approach, ϵ must be large enough to satisfy the safety requirement that for every value of X that $f(X)\mathrm{I}^{TL}(T_c, X)$ is non-zero, $\tilde{f}(X)$ is also non-zero. This translates to the requirement that $\mathrm{I}^{TL}(T_c, X_i) = 1 \Rightarrow \mathrm{I}^{PDM}(T_c^{\epsilon}, X_i) = 1$. Let us define ϵ_{abs} to be the smallest value of ϵ that theoretically guarantees the margin condition. ϵ_{abs} as a function of the timing constraint T_c is given by

$$\epsilon_{abs}(T_c) = max_{\text{over all } X \text{ such that } d_C^{TL}(X) \geq T_c} \ (T_c - d_C^{PDM}(X)).$$

However, the value of ϵ_{abs} is not known in practice because it requires knowledge of d_C^{TL} throughout the entire sample space. Therefore, the algorithm must try to heuristically provide a value of ϵ close enough to ϵ_{abs} in order to minimize the bias in the estimator.

On the other hand, as seen in (27), the closer $Loss^{PDM,\epsilon}$ is to $Loss^{TL}$, the more speedup the IS estimator achieves over standard MC. Making $Loss^{PDM,\epsilon}$ close to $Loss^{TL}$ requires that ϵ be kept close to a particular value ϵ^* that satisfies $Loss^{PDM,\epsilon^*} = Loss^{TL}$. Thus, to make IS efficient while preserving correctness, we must choose ϵ as close to ϵ^* as possible. Similarly to the case in the paragraph above, the value of ϵ^* is not known in practice, since it requires the entire sample space to be covered by TL simulations.

To summarize, the algorithm must pick a value of ϵ as close to ϵ^* as possible without going below ϵ_{abs}. However, since neither of these quantities are known a priori, we use the heuristic algorithm in this section to compute an ϵ that is a good compromise. In the experiments in Section V, we demonstrate that our heuristic strikes a good compromise between accuracy and efficiency in all benchmarks.

Heuristic criterion for ϵ: CompLossMC-IS explores the samples Y_i in increasing order of i, i.e., in decreasing order of their d_C^{PDM} values. For a given value of SM (short for "Safety

Algorithm 1 CompLossMC-IS (NS, SM, T_c)

1: Generate NS sample points $\{X_1, X_2, ..., X_{NS}\}$ from $f(X)$.
2: For each X_i, compute $d_C^{PDM}(X_i)$.
3: Let $\mathcal{X} = \{Y_1, Y_2, Y_3, ..., Y_{NS}\}$ be the NS samples in decreasing order of $d_C^{PDM}(Y_i)$.
4: $i = 1$, $\mathcal{Z} = \emptyset$, $SafeMargin = \emptyset$
5: **while** ($|SafeMargin| < SM$ and $i \leq NS$) **do**
6: $d_C = d_C^{TL}(Y_i)$
7: **if** ($d_C < T_c$) **then**
8: $\mathcal{Z} = \mathcal{Z} \cup \{Y_i\}$
9: **if** $SafeMargin == \emptyset$ **then**
10: $\epsilon = T_c - 0.5(d_C^{PDM}(Y_i) + d_C^{PDM}(Y_{i-1}))$
11: **end if**
12: $SafeMargin = SafeMargin \cup \{Y_i\}$
13: **else**
14: $SafeMargin = \emptyset$
15: **end if**
16: $i = i + 1$
17: **end while**
18: Let $N = i - SM - 1$ and $SafeMargin = \{Y_{N+1}, ..., Y_{N+SM}\}$.
19: Let $\mathcal{W} = \{Y_1, ..., Y_N\}$
20: $\mathcal{Q} = \mathcal{W} \cap \mathcal{Z}$
21: $Loss_N^{IS} = Loss^{PDM,\epsilon} . |\mathcal{W} - \mathcal{Q}| / |\mathcal{W}|$

Margin"), CompLossMC-ISs goal is to select ϵ satisfying the following property.

1) There is a sequence of *SM* sample points $\{Y_{N+1}, ..., Y_{N+SM}\}$ that constitute the safety margin (called *SafeMargin* in the algorithm). For each point Y in the margin

$$d_C^{TL}(Y) < T_c.$$

And ϵ satisfies

$$\epsilon = T_c - 0.5(d_C^{PDM}(Y_{N+1}) + d_C^{PDM}(Y_N)).$$

The safety margin (heuristically) provides confidence that the safety condition is satisfied for the remaining points for which a TL simulation has not been carried out. This is because all of these remaining samples have a value of d_C^{PDM} less than $T_c - \epsilon$.

In Section V, we show that, using a reasonably small *SM*, the heuristic criterion provides an estimator with negligible bias. We further confirm that none of the *NS* points that have a d_C^{TL} value greater than T_c are ever missed by the heuristic. CompLossMC-IS runs only *SM* additional TL simulations beyond those needed for $Loss_N^{IS}$.[1] The computational cost of $Loss^{PDM,\epsilon}$ determination is unavoidable with the IS estimator and is not due to the adaptive determination of ϵ.

Determining *NS*: Roughly speaking, the user provides the algorithm with a number *NS*, and s/he expects to carry out approximately *Loss.NS* TL simulations. Since the intended use of our proposed approach is accurate, late-stage yield determination, a rough estimate for *Loss* should be available. If not, $Loss^{PDM}$ can be used as a rough guide. It is important to note that the choice of *NS* is guided by how small one would like the variance of the $Loss^{IS}$ estimator to be. The purpose of *NS* is not to sample the parameter space in order to determine a safe value of ϵ. ϵ is determined heuristically and this heuristic is empirically justified separately. *NS* is chosen so that roughly *NS.Loss* TL simulations are affordable, and the variance of the IS estimator for *NS.Loss* samples is as small as desired.

G. Discussion

A key benefit of the IS approach is that TL simulations are avoided for discarded samples, i.e., when X_i results in a PDM circuit delay estimate smaller than T_c^ϵ ($I^{PDM}(T_c^\epsilon, X_i) = 0$). The improvement brought about by the IS estimator, however, goes significantly beyond this. For the same number of samples N, the IS estimator in (21) provides a much more accurate (with significantly reduced variance) loss estimate than the standard estimator in (18). Were it possible to use the ideal biasing function $\tilde{f}_{\text{ideal}}$, a zero-variance estimator would have been obtained with a single sample. IS approach makes it possible to explore the space between standard MC and this ideal. Using an $\tilde{f}$ that approximates $\tilde{f}_{\text{ideal}}$ as closely as possible, IS both reduces the number of TL circuit simulations required *and* improves upon the standard estimator accuracy achieved for the same number of TL simulations. The next section makes this discussion precise.

[1]The overhead due to the additional *SM* simulations is taken into account in the reported `Speedup` results in Section V.

IV. Convergence (Error) and Speedup Analysis

In this section, we present a precise analysis that quantifies the variance reduction and the speed-up obtained when we use the IS estimator instead of the standard MC estimator.

The error of an estimator is the deviance of the estimator's result from the actual loss as explained in Section II for a general estimator. Next, the errors of the standard and IS MC estimators are derived and the results are compared.

Theorem 1: The error of the standard estimator in (18) obtained with N samples is

$$Error^{TL}(N) = \frac{2\sqrt{Loss^{TL}.Yield^{TL}}}{\sqrt{N}} \tag{24}$$

with more than 95% confidence.

Proof: See [20]. ■

Theorem 2: The error of the IS estimator in (19) and (21) obtained with N samples is given by

$$Error^{IS}(N) = \frac{2\sqrt{Loss^{TL}.(Loss^{PDM,\epsilon} - Loss^{TL})}}{\sqrt{N}} \tag{25}$$

with more than 95% confidence.

Proof: See [20]. ■

In the derivation of the IS estimator error above, $Loss^{PDM,\epsilon}$ was assumed to be a known deterministic quantity. However, $Loss^{PDM,\epsilon}$ is estimated using the estimator in (23), and in fact, it is a random quantity with a nonzero variance that decreases proportionally to the number of samples K used in (23). In order for the error derivation for the IS estimator in (21) to be valid, the estimation of $Loss^{PDM,\epsilon}$ must be performed by using a large enough number of samples in (23) so that it has negligible variance. This would validate its treatment as a deterministic quantity in the derivation of the error equation for the IS estimator. The use of a large number of samples in (23) is easily affordable, because no TL simulations are performed, only simple evaluations of the cheap delay models are needed. The results we present later show that the theoretical error expressions derived here are in excellent agreement with experimental data.

The error equations (24) and (25) that have been derived with Theorem 1 and Theorem 2 for the standard and IS MC estimators can be used to compare them. If the same number of samples N is used for both methods (meaning an equal number of TL simulations), then the ratio of the errors of the estimators is given by

$$ErrorRatio(N) = \frac{Error^{TL}(N)}{Error^{IS}(N)} = \sqrt{\frac{Yield^{TL}}{(Loss^{PDM,\epsilon} - Loss^{TL})}}. \tag{26}$$

Alternatively, suppose a bound on the allowable estimation error is given. The ratio of the number of samples (TL circuit simulations) required by the two approaches to achieve this same error bound is given by

$$\texttt{Speedup} = \frac{N_{TL}}{N_{IS}} = \frac{Yield^{TL}}{(Loss^{PDM,\epsilon} - Loss^{TL})} \tag{27}$$

which is obtained by solving $Error^{TL}(N_{TL}) = Error^{IS}(N_{IS})$ for $\frac{N_{TL}}{N_{IS}}$, which we call `Speedup`, since the number of samples used

in the estimators determines the number of TL simulations that need to be performed on the statistically critical paths of the circuit. Based on (26) and (27) above, we note here that Speedup can alternatively be computed with

$$\texttt{Speedup} = \frac{Error^{TL}(N)^2}{Error^{IS}(N)^2} \tag{28}$$

as the ratio of the squared errors for the standard MC and IS estimators with the same number of samples $N = N_{TL} = N_{IS}$, i.e., the same number of TL simulations.

Finally, we address a question that may arise in the mind of an attentive reader. If Speedup in (27) is large, one might conclude that TL simulations are not needed and $Loss^{PDM,\epsilon}$ can simply be used as an accurate loss estimate. This conclusion would be based on the observation that $Loss^{PDM,\epsilon}$ has to be very close to $Loss^{TL}$ if one can attain a large Speedup in (27). However, this conclusion is not correct. $Loss^{PDM,\epsilon}$ is computed using (23), where $T_c^{\epsilon} = T_c - \epsilon$ with ϵ as the margin parameter. The margin parameter is determined by an adaptive algorithm which performs TL simulations in its search for the correct ϵ value. If $Loss^{PDM,\epsilon}$ is not computed based on the ϵ value found by the CompLossMC-IS algorithm described in Section III-F, then the resultant $Loss^{PDM,\epsilon}$ will not be close to $Loss^{TL}$. Therefore, attaining a large Speedup does not mean that the PDM model is by itself accurate enough for loss estimation. The PDM model needs to be in some sense "calibrated" or corrected with the TL simulations run at the critical samples in the parameter space selected using IS.

V. Results

A. Experimental Setup

We first describe our experimental setup in order to help interpret our results better:

We present results on the ISCAS'85 benchmark suite [25] in this paper. We use the 0.13μ standard cell library provided by Graham Petley [26] for the transistor-level implementations of the gates needed in order construct the benchmark circuits. We have added some missing 5-, 8-, and 9-input gates to this library, as they are needed for some of the circuits in the ISCAS'85 benchmark suite. The layout information for the circuits, i.e., relative locations of the gates on the layout for a particular benchmark circuit (needed for the intra-die variation model that captures spatial correlations), are extracted from the information provided on the VLSI computer aided design group web pages at Texas A&M University [27].

Two random transistor parameters, namely the transistor gate length L and the threshold voltage V_t, are considered. Both inter and intra-die variations for these parameters are taken into account and a statistical model as described in Section III-A is constructed. In this model, half of the variation is allocated to inter-die variations and the other half to intra-die variations [19], with a total $3\sigma/\mu$ ratio of 15% for both of the random parameters L and V_t. In the quad-tree model [19] that captures spatial correlations, we use four grid levels (layers). We allocate half of the variation to the top level that covers the whole area of the circuit with one grid rectangle in order to capture perfectly correlated inter-die variations. The other three levels in the model capture the spatially correlated intra-die variations and are allocated one sixth of the total variation each. These allocations are done by appropriately choosing the variances of the grid random variables in the quad-tree model. As described in Section III-A, we use 85 random variables in the quad-tree model per parameter. With two random transistor parameters, L and V_t, the random variable vector X defined in Section II-A has a dimension of 170 in all of our experiments.

For each of the circuits in the ISCAS85 benchmark suite, we determine a set of statistically critical paths using the method described in Section III-C. We experiment with two timing constraints for each circuit, $T_{c,\,\mathrm{low}}$ and $T_{c,\,\mathrm{high}}$ that result in roughly 10% and 5% loss, respectively. When we use our improved IS-based estimator for timing yield, the required margin parameter ϵ that was introduced in Section III-E is computed automatically using the algorithm described in Section III-F. It is important to note that, as it computes ϵ, this algorithm carries out all TL circuit simulations required for computing the IS estimator.

With the results that we present in this section, we compare the accuracy and the efficiency of our improved IS estimator against the standard MC estimator. In doing so, we empirically compute the error (variance) achieved for both of the loss estimators. In order to measure the error of an estimator, we perform M (to be quantified precisely) independent repetitions of the same experiment (evaluation of the estimator). In each independent run, we compute the loss estimates with the IS estimator by using R (to be quantified precisely) independently drawn samples from the PDF $f(X)$ in the parameter space. These M independent runs constitute the samples of the loss estimator, and the variance and error of the loss estimator is computed over these M samples. For the IS estimator, most of the R samples are discarded as explained in Section III-E based on the evaluation of the PDM equations, and a reduced number (N_{IS} on the average, showing negligible variation from run to run) of TL simulations are performed. All of these N_{IS} simulations are performed as part of the iterative algorithm for computing ϵ. In other words, N_{IS} includes all of the TL simulations required to compute ϵ and $Loss_N^{IS}$. In evaluating the standard MC estimator, we choose $N_{TL} = N_{IS}$ samples randomly among the R samples in every set. For the standard MC estimator, the results of TL circuit simulations performed at every one of the N_{TL} sample points are used.

The $Loss^{PDM,\epsilon}$ value that is needed for computing the IS estimator in (21) is computed using the PDM-based estimator in (23) using all of the $K = M \times R$ sample points generated during all of the M runs.

The Speedup that we report for the IS estimator over the standard MC estimator represents the ratio of the number of TL circuit simulations required by the standard MC and IS estimators to achieve the same error, as given by (27). Alternatively, Speedup is the ratio of the squared errors (variances) for the loss estimates obtained by the two estimators with the same number of samples (TL simulations), as given by (28).

B. Experiments

1) *Experiment A: Three Statistically Critical Paths:* In this experiment, for each benchmark circuit we choose three most statistically critical paths which are identified using the scheme described in Section III-C. We then perform the following:

We construct M sample sets, each with R samples drawn from the PDF $f(X)$, with a total of $K = M \times R$ samples in the random parameter space. In this experiment, we have used $M = 100$ and $R = 200$ for $T_{c,\,\mathrm{low}}$ and $M = 50$ and $R = 400$ for $T_{c,\,\mathrm{high}}$, for a total of $K = 20\,000$ samples in each case. For each set, we evaluate the IS estimator. The IS estimator eliminates most of the samples in the set without performing TL simulations, as explained in Section III-E. Loss estimates obtained with the IS estimator and the number of actual TL simulations run for each set, N_{IS}, are recorded. All of the sets have the same number of random samples in the parameter space, R. N_{IS}, corresponding to the number of non-discarded samples at which TL simulations are run, does not show much variation from set to set. For each experiment, up to SM extra TL simulations that are used in order to heuristically determine a safe value of ϵ are included in N_{IS}. This allows a fair comparison with standard MC. The variance of the loss estimates over the M sets is computed as VAR^{IS}. The standard MC estimator is evaluated for every set using N_{IS} number of samples chosen randomly among the R samples in each set. Thus, loss with the standard MC estimator is evaluated using the same number of TL simulations as the IS estimator, i.e., $N_{TL} = N_{IS}$. The variance of the M loss values computed with the standard MC estimator is computed as VAR^{TL}.

With the construction above, we have $N_{TL} = N_{IS} = N + SM$. Hence, we substitute VAR^{IS} and VAR^{TL} in (28) (in place of squared errors, as the ratio of the variances of the estimators is the same as the ratio of their squared errors) in order to quantify the Speedup of the IS estimator over the standard MC estimator

$$\mathtt{Speedup} = \frac{VAR^{TL}}{VAR^{IS}} \qquad \text{with} \quad N_{IS} = N_{TL}. \tag{29}$$

The Speedup results, computed as described above for $T_{c,\,\mathrm{low}}$ and $T_{c,\,\mathrm{high}}$, for the ISCAS85 benchmark suite are presented in Tables I and II. In these tables, the mean values for $Loss^{IS}$ and $Loss^{TL}$ using the same number of TL simulations ($N_{TL} = N_{IS} = N + SM$) are also shown. Furthermore, we also report the loss values (labeled as $Loss$) computed using TL simulations at all 20 000 samples, which can be regarded as the real loss value. The mean value of the $Loss^{IS}$ estimator was on the average within 0.56% of the loss value computed using TL simulations at all 20 000 samples. The mean value of the $Loss^{TL}$ estimator was on the average within 6.82% of the loss value computed using TL simulations at all 20 000 samples.

As discussed in Section IV, Speedup in Tables I and II represents the ratio of the number of TL simulations required by the standard MC estimator to the number of TL simulations needed by our proposed IS estimator in order to estimate the loss of the circuit with the same error (accuracy). Alternatively, if the same number of TL simulations are used for both of the estimators, the estimation variance for loss will be Speedup times less for our IS estimator. As seen in Tables I and II, our accelerated yield estimator achieves on the average *two orders of magnitude* Speedup over standard MC.

Contrasting absolute errors of estimators: In the method we propose, while computing the loss estimator $Loss^{IS}$, approximate delay computed using the PDM method (with an ϵ margin) is used by IS in order to achieve a low-variance estimator. The question naturally arises as to how good an estimator $Loss^{PDM}$ is, and whether it itself could be used for yield estimation. We contrast the absolute errors in the following four estimators.

1) $Loss^{TL}$*:* The MC estimator.
2) $Loss^{IS}$*:* The MC estimator with IS.
3) $Loss^{PDM}$*:* The PDM estimator with no adjustment.
4) $Loss^{PDM,\epsilon}$*:* The PDM estimator with the ϵ margin (i.e., with $T_c - \epsilon$ as the timing constraint).

We computed the loss estimated by each approach. The results for $T_{c,\,\mathrm{low}}$ and $T_{c,\,\mathrm{high}}$ are presented in Tables III and IV, respectively. Since it is computationally very costly to run TL to convergence, we have taken the $Loss^{TL}$ value computed from 20 000 samples as the reference in each case. These reference loss values are labeled as $Loss$ in Tables III and IV and correspond to the columns with the same label in Tables I and II. The $Loss^{IS}$, $Loss^{PDM}$ and $Loss^{PDM,\epsilon}$ values were also obtained from a single run on the same 20 000 samples. To compute $Loss^{PDM,\epsilon}$ we used the ϵ value found during the $Loss^{IS}$ computation. As seen in the tables, $Loss^{IS}$ is bias-free in all cases. This is to be expected, since $Loss^{IS}$ is an unbiased estimator in theory—a fact experimentally demonstrated further in Experiments B and C below.

The value of the uncorrected PDM estimator $Loss^{PDM}$ is too far from $Loss^{TL}$ to be acceptable for most benchmarks, resulting in errors of 39.4% at most and 16.3% on the average. It is important to note that, if one had carried out block-based MC statistical timing analysis using our polynomial gate delay models, this is the accuracy one would have obtained. While delay values as computed by the PDM model only correlate well with the actual values, in terms of the absolute value of delay, they are far off. Thus, while the PDM model may serve as a rough guide for timing and yield optimization, it is not accurate enough for the numerical prediction of yield. This is important as it justifies the use of our method as a final pass of yield estimation.

The ϵ-corrected $Loss^{PDM,\epsilon}$ is a better estimator for loss. The fact that $Loss^{PDM,\epsilon}$ is in many cases close to $Loss^{TL}$ might appear to suggest that yield prediction using PDM with the ϵ correction is a sufficiently accurate and cheap method. However, in order to compute the ϵ correction factor, all the TL simulations required for computing $Loss^{IS}$ have to be carried out. Furthermore, $Loss^{PDM,\epsilon}$ is actually not close enough to the actual loss value to be an accurate estimator in its own right. Thus, it makes more sense to use more accurate and provably unbiased estimator $Loss^{IS}$.

Validating heuristically-computed ϵ values: Recall that the ϵ margin involved in the computation of $Loss^{IS}$ is arrived at using a heuristic in the algorithm CompLossMC-IS. In order to validate the computed ϵ values, we performed the

TABLE I
SPEEDUP FOR $T_{c,\text{low}}$, $M = 100$ SETS, AND $R = 200$ SAMPLES EACH

Bench.	$N + SM$	*Loss*	Mean $Loss^{TL}$	Mean $Loss^{IS}$	$Error^{TL}$ (%)	$Error^{IS}$ (%)	Speedup
c432	30	0.1158	0.1073	0.1145	51.05	5.15	100
c499	30	0.1225	0.1197	0.1220	48.90	3.52	194
c880	29	0.1132	0.1172	0.1122	53.50	5.81	86
c1355	32	0.1311	0.1297	0.1308	46.47	3.26	204
c1908	29	0.1175	0.1186	0.1164	51.39	4.38	140
c2670	30	0.1195	0.1210	0.1190	46.87	3.74	159
c3540	28	0.1109	0.1132	0.1104	54.11	4.30	160
c5315	29	0.1170	0.1172	0.1169	52.00	3.76	192
c7552	27	0.1054	0.1074	0.1044	57.29	3.99	210

TABLE II
SPEEDUP FOR $T_{c,\text{high}}$, $M = 50$ SETS, AND $R = 400$ SAMPLES EACH

Bench.	$N + SM$	*Loss*	Mean $Loss^{TL}$	Mean $Loss^{IS}$	$Error^{TL}$ (%)	$Error^{IS}$ (%)	Speedup
c432	32	0.0632	0.0500	0.0630	70.51	5.94	142
c499	22	0.0411	0.0300	0.0408	100.02	5.70	312
c880	27	0.0528	0.0519	0.0525	79.97	5.63	204
c1355	31	0.0644	0.0477	0.0644	68.60	4.81	203
c1908	33	0.0677	0.0642	0.0670	65.21	4.13	255
c2670	37	0.0788	0.0751	0.0781	55.92	3.00	352
c3540	35	0.0733	0.0651	0.0730	58.74	3.98	220
c5315	38	0.0788	0.0779	0.0784	56.82	3.49	269
c7552	30	0.0613	0.0640	0.0614	68.43	3.19	459

following checks.

1) *Confirming that $Loss^{IS}$ is unbiased:* Results presented in Fig. 3 in Section V-B3 below, and Tables III and IV indicate that when a large number of samples NS is used, $Loss^{IS}$ is an unbiased estimator. Of more practical importance is the fact that when $Loss^{IS}$ is computed with about 30 samples, the mean of $Loss^{IS}$ is on the average within 0.56% of the *Loss* value computed from 20 000 samples.
2) *Exploring different values of ϵ:* Recall that $\epsilon_{abs}(T_c) = max_X$ such that $_{d_C^{TL}(X) \geq T_c}(T_c - d_C^{PDM}(X))$ is the smallest value of ϵ that guarantees the margin condition. As the closest practical approximation to the ideal ϵ_{abs}, we computed ϵ_{abs} using the equation above and letting X range over the 20 000 sample points we had for each benchmark. In order to investigate the sensitivity of the yield estimator to the safety margin, we carried out the following experiment. We forced our IS algorithm to use a fixed value of ϵ. We varied this value of ϵ in the range 0, 0.1 ϵ_{abs}, 0.2 ϵ_{abs}, ..., 0.9 ϵ_{abs}, ϵ_{abs}. We then computed the percentage bias error in the mean of the IS estimator for each of these values of ϵ, including the one our heuristic computes. The results are shown in Table V. The last column presents the results for the ϵ our heuristic finds. The first observation is that, for each benchmark, there is a value of ϵ below which the bias in the loss estimator is too high. This value of ϵ is different for each benchmark, but is in the 0.5–0.8 ϵ_{abs} range. Above this value of ϵ, the bias is not very sensitive to the particular value of ϵ. The second key observation is that the heuristically found ϵ value for each benchmark always results in an acceptable bias (error) in the loss computed. This bias is less than 1% of the absolute loss. Given that larger approximations are probably involved in parameter variation modeling, etc., a bias error of 1% of the loss is certainly negligible.
3) *Validating the value of SM used:* For every benchmark and every run of CompLossMC-IS in Experiments A and B, we confirmed that no run of CompLossMC-IS missed a sample point X with $d_C^{TL}(X) > T_c$ in any of the runs.

   For each benchmark, we explored values of *SM* from 1 to *NS*. We found that even very small values of SM result in an acceptable error in the loss computation. Larger values of SM result in values of ϵ closer to ϵ_{abs} but this results in only very small differences in the actual bias. In order to make sure that the ϵ value we choose provides a good compromise between accuracy and high speedup, and to keep the computational cost still low, we pick SM to be 20% of the number of points that we expect the IS approach to perform TL simulations on. This amounts to $SM = 4$ for the examples in which R=200 ($T_{c,\text{low}}$) and R=400 ($T_{c,\text{high}}$), and $SM = 20$ for $R = 1000$ ($T_{c,\text{low}}$). With this choice, the bias error in each benchmark is within approximately 1% of the absolute loss.

2) *Experiment B: Ten Statistically Critical Paths:* In this experiment, we randomly choose one of the ISCAS'85 benchmark circuits, and repeat the same experiment described above

TABLE III
LOSS FOR $T_{c,\text{low}}$, FOUR DIFFERENT ESTIMATORS

Benchmark	$Loss$	$Loss^{IS}$	$Loss^{PDM,\epsilon}$	$Loss^{PDM}$
c432	0.1158	0.1158	0.1418	0.0973
c499	0.1225	0.1225	0.1288	0.0803
c880	0.1132	0.1132	0.1288	0.0914
c1355	0.1311	0.1311	0.1379	0.1003
c1908	0.1175	0.1175	0.1280	0.1122
c2670	0.1195	0.1195	0.1254	0.1061
c3540	0.1109	0.1109	0.1196	0.0961
c5315	0.1170	0.1170	0.1271	0.1083
c7552	0.1054	0.1054	0.1134	0.1012

TABLE IV
LOSS FOR $T_{c,\text{high}}$, FOUR DIFFERENT ESTIMATORS

Benchmark	$Loss$	$Loss^{IS}$	$Loss^{PDM,\epsilon}$	$Loss^{PDM}$
c432	0.0632	0.0632	0.0798	0.0514
c499	0.0411	0.0411	0.0426	0.0229
c880	0.0528	0.0528	0.0633	0.0405
c1355	0.0644	0.0644	0.0687	0.0454
c1908	0.0677	0.0677	0.0741	0.0652
c2670	0.0788	0.0788	0.0823	0.0693
c3540	0.0733	0.0733	0.0803	0.0613
c5315	0.0788	0.0788	0.0862	0.0724
c7552	0.0613	0.0613	0.0665	0.0595

($T_{c,\text{low}}$), but with ten most statistically critical paths.[2] In this case, we obtain a Speedup of 147, which is almost the same as the one obtained for the same benchmark circuit with only three critical paths. The following values were obtained in this experiment: $N = 42$, $Loss = 0.1244$, mean $Loss^{TL} = 0.1306$, mean $Loss^{IS} = 0.1239$, $Error^{TL} = 47.91$ %, $Error^{IS} = 3.95$ %. These results confirm that the Speedup achieved by our IS estimator is not dependent on the number of statistically critical paths considered for a circuit. The efficiency of the IS estimator does not degrade if a large number of critical paths are included in yield estimation, because the maximum of the path delays in (1) for the overall circuit delay is computed exactly, without employing approximations. This is a key advantage of our technique. If an approximate maximum operation is employed in computing the circuit delay from path delays, the accuracy will degrade if a large number of paths are considered.

3) *Experiment C: Validating Theoretical Error Equations:* The purpose of this experiment is to empirically validate the theoretical convergence analysis conducted and the error estimation equations derived in Section IV for the standard MC and IS estimators.

The results we present in Fig. 2 experimentally confirm the error/convergence equations, (24) for the standard MC and (25) for the IS estimators, that were derived in Section IV. In this figure, a plot of loss error versus the number of TL circuit simulations is shown for both estimators. The smooth curves in this plot were obtained using the theoretical error formulas. The two other curves were generated by computing

[2]We were not able to run this experiment for all of the circuits in the benchmark suite due to the excessive computational resources required by the standard MC technique against which we compare our proposed estimator.

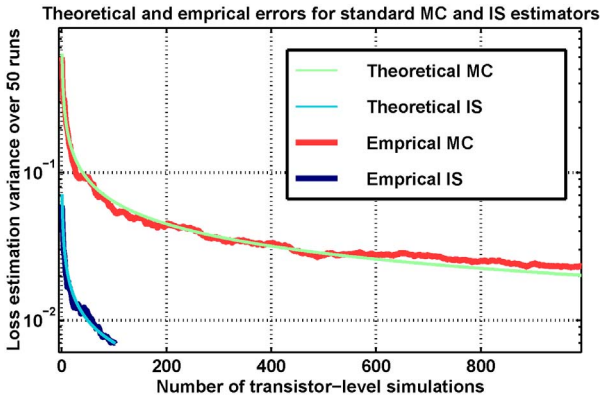


Fig. 2. Convergence of standard MC and IS estimators.

loss estimate errors over 50 independent runs, each of which explore a sample set size R (number of samples drawn from the PDF $f(X)$) ranging from 1 to 1000. As explained before, TL circuit simulations are performed at all of the sample points for the standard MC estimator, but a reduced number of simulations are performed for the IS estimator since most of the samples are discarded based on the evaluation of the PDM equations. We observe the excellent match between the theoretical and experimental error curves in this plot, validating the $1/\sqrt{N}$ dependency of error on the number of TL simulations for both of the estimators. The significant reduction in error that the IS estimator provides is also obvious in this graph. The results in Fig. 2 were generated with circuit **c3540** in the ISCAS'85 benchmark suite (with similar results for the other circuits). In this case, the $Loss^{PDM,\epsilon}$ value that is needed for computing the IS estimator in (21) is computed using the PDM-based estimator in (23) using all of the 50 000 sample points generated during all of the 50 runs. In empirically computing the variances of both of the estimators to generate the curves in Figure 2, we use the loss value computed based on the standard MC estimator with TL simulations at all of the 50 000 sample points in the parameter space. Since the number of samples used here is very large, we treat this loss value as the actual loss as if it was given to us by an oracle.

As discussed in Section II and Section III, both the standard MC and IS estimators are unbiased and their means converge to the actual loss if a large number of samples are used. We empirically confirm this with the plot in Fig. 3. The curves in this plot were generated using the same experiment described above that was used to generate the error curves in Fig. 2. In order to generate the plot in Fig. 3, we simply compute the means of the loss estimates obtained by the two estimators over the 50 independent runs with varying number of samples, whereas variances over these 50 runs were used for Fig. 2. We can clearly observe in Fig. 3 that the IS estimator converges to the actual loss value much earlier, with only a few number of TL simulations.

Finally, we report the Speedup values achieved by our IS estimator over the standard MC estimator in this experiment where 1000 samples in each of the 50 independent runs were used ($M = 50$ and $R = 1000$). The Speedup obtained for five

TABLE V

% IS ESTIMATOR BIAS ERROR VERSUS MARGIN PARAMETER ϵ ($T_{c,\text{low}}$, M = 100 SETS, R = 200 SAMPLES EACH)

	0	0.1 ϵ_{abs}	0.2 ϵ_{abs}	0.3 ϵ_{abs}	0.4 ϵ_{abs}	0.5 ϵ_{abs}	0.6 ϵ_{abs}	0.7 ϵ_{abs}	0.8 ϵ_{abs}	0.9 ϵ_{abs}	ϵ_{abs}	ϵ (Heur.)
c432	15.98	12.44	9.19	6.51	3.74	2.00	1.08	0.62	0.14	0.07	0.07	**1.10**
c499	34.49	31.22	27.59	23.84	19.67	16.16	11.88	8.45	4.26	1.33	0.04	**0.37**
c880	19.27	16.88	14.02	11.31	8.08	4.96	3.22	1.44	0.49	0.08	0.02	**0.85**
c1355	23.46	21.02	18.81	16.52	13.93	10.99	7.46	4.40	1.73	0.35	0.04	**0.20**
c1908	4.73	3.53	2.67	2.29	1.59	1.03	0.79	0.57	0.42	0.21	0.09	**0.93**
c2670	11.18	9.54	7.83	6.57	4.94	3.53	2.12	1.26	0.55	0.17	0.03	**0.40**
c3540	13.35	11.40	9.82	7.23	5.50	4.30	3.01	1.07	0.46	0.24	0.09	**0.43**
c5315	7.44	6.24	4.90	3.26	2.08	1.13	0.42	0.18	0.03	0.09	0.09	**0.11**
c7552	4.40	3.68	2.74	2.01	1.31	0.85	0.36	0.10	0.15	0.16	0.10	**0.87**

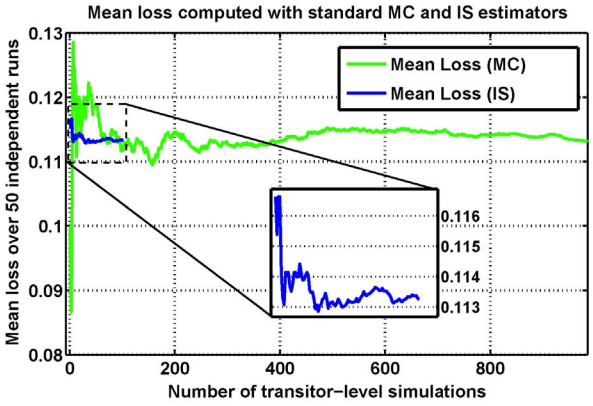


Fig. 3. Mean loss computed with standard MC and IS estimators.

TABLE VI

SPEEDUP OF THE IS ESTIMATOR OVER STANDARD MONTE CARLO

Experiment C	1) Three statistically critical paths 2) 1000 samples in 50 independent runs
Benchmark	Speedup
c880	80
c1908	120
c2670	124
c3540	142
c5315	195

of the circuits in the ISCAS'85 benchmark suite are given in Table VI. As seen, we obtain similar Speedup values here as for the ones in Table I which were obtained with an experiment where 200 samples in each of 100 independent runs were used. Thus, the two orders of magnitude Speedup performance that is achieved by the proposed IS estimator is confirmed again with the results obtained here from a much larger data set.

VI. CONCLUSION

We have described an overall methodology that enables the practical use of transistor-level MC simulations in estimating the timing yield of a digital circuit as a final verification before timing sign-off. The original, novel contribution of this paper is a unique IS scheme for accelerating timing yield computations based on transistor-level MC simulations. In the proposed scheme, a cheap-to-evaluate but approximate gate delay model is utilized in order to generate "important" samples in the random parameter space at which transistor-level simulations are run. The improved MC estimator based on these important samples achieves the same accuracy as the standard MC estimator but at a significantly reduced cost of much fewer number of transistor-level simulations. The results we present in this paper show that the achieved speed-up is two orders of magnitude over standard MC.

REFERENCES

[1] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 4, pp. 589–607, Apr. 2008.

[2] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *Proc. 41st DAC*, 2004, pp. 331–336.

[3] J. Singh and S. S. Sapatnekar, "A scalable statistical static timing analyzer incorporating correlated nongaussian and gaussian parameter variations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 1, pp. 160–173, Jan. 2008.

[4] Taiwan Semiconductor Manufacturing Company (TSMC). New TSMC reference flow 9.0 supports 40 nm process technology. *Reference Flow 9.0 Press Release* [Online]. Available: http://www.tsmc.com

[5] L. Scheffer, "The Count of Monte Carlo," in *Proc. ACM/IEEE TAU*, Feb. 2004.

[6] D. E. Hocevar, M. R. Lightner, and T. N. Trick, "A study of variance reduction techniques for estimating circuit yields," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 2, no. 3, pp. 180–192, Jul. 1983.

[7] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization," in *Proc. IEEE/ACM ICCAD*, Nov. 2008, pp. 322–329.

[8] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events," in *Proc. 43rd ACM/IEEE DAC*, 2006, pp. 69–72.

[9] A. Singhee and R. A. Rutenbar, "Statistical blockade: A novel method for very fast Monte Carlo simulation of rare circuit events, and its application," in *Proc. DATE Conf. Exhibition*, Apr. 2007, pp. 1–6.

[10] M. Zhang, M. Olbrich, H. Kinzelbach, D. Seider, and E. Barke, "A fast and accurate Monte Carlo method for interconnect variation," in *Proc. IEEE ICICDT*, Jan. 2006, pp. 1–4.

[11] G. Yu, W. Dong, Z. Feng, and P. Li, "A framework for accounting for process model uncertainty in statistical static timing analysis," in *Proc. 44th ACM/IEEE DAC*, Jun. 2007, pp. 829–834.

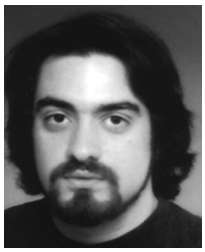
[12] V. Khandelwal and A. Srivastava, "A general framework for accurate statistical timing analysis considering correlations," in *Proc. 42nd DAC*, Jun. 2005, pp. 89–94.

[13] M. Chen and A. Orailoglu, "Circuit-level mismatch modelling and yield optimization for CMOS analog circuits," in *Proc. 25th ICCD*, Oct. 2007, pp. 526–532.

[14] S. Yaldiz, U. Arslan, X. Li, and L. Pileggi, "Efficient statistical analysis of read timing failures in SRAM circuits," in *Proc. ISQED*, Mar. 2009, pp. 617–621.

[15] B. Liu, "Gate level statistical simulation based on parameterized models for process and signal variations," in *Proc. ISQED*, Mar. 2007, pp. 257–262.

[16] V. Veetil, D. Sylvester, and D. Blaauw, "Efficient Monte Carlo-based incremental statistical timing analysis," in *Proc. 45th DAC*, 2008, pp. 676–681.
[17] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods, Volume 1, Basics*. New York: Wiley, 1986.
[18] P. Glasserman, P. Heidelberger, and P. Shahabuddin, "Importance sampling and stratification for value-at-risk," in *Proc. 6th Int. Conf. Comput. Finance*, May 1999, pp. 7–24.
[19] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," in *Proc. ACM/IEEE ICCAD*, 2003, pp. 900–907.
[20] A. A. Bayrakci, A. Demir, and S. Tasiran, "Fast Monte Carlo estimation of timing yield: Importance sampling with stochastic logical effort (ISLE)," Koc Univ., Istanbul, Turkey, Tech. Rep. arXiv:0805.2627, 2007 [Online]. Available: http://arxiv.org
[21] K-T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Comput. Experiments*. Boca Raton, FL: CRC Press, 2006.
[22] Y. Zhan, A. J. Strojwas, M. Sharma, and D. Newmark, "Statistical critical path analysis considering correlations," in *Proc. ACM/IEEE ICCAD*, Nov. 2005, pp. 699–704.
[23] F. Wang, Y. Xie, and H. Ju, "A novel criticality computation method in statistical timing analysis," in *Proc. Conf. DATE*, 2007, pp. 1–6.
[24] F. S. Marques, R. P. Ribas, S. Sapatnekar, and André I. Reis, "A new approach to the use of satisfiability in false path detection," in *Proc. 15th ACM GLSVLSI*, 2005, pp. 308–311.
[25] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1985, pp. 695–698.
[26] G. Petley's 0.13μ Cell Library, Release 8.5 [Online]. Available: http://www.vlsitechnology.org
[27] Texas A&M University. *VLSI Computer-Aided Design (CAD) and Test Research Group* [Online]. Available: http://dropzone.tamu.edu/∼cad
[28] A. D. Sokal, "Monte Carlo methods in statistical mechanics: Foundations and new algorithms," in *Functional Integration: Basics and Applications*, P. Cartier, C. DeWitt-Morette, and A. Folacci, Eds. New York: Plenum, 1997.



Alp Arslan Bayrakci received the B.S. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 2004. He is currently pursuing the Ph.D. degree in computer engineering from Koc University, Istanbul, Turkey.

His current research interests include statistical analysis and computer-aided design methodologies.

Alper Demir (S'93–M'96–SM'10) received the B.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 1991 and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1994 and 1997, respectively.

In 1995, he was with Motorola, Austin, TX; in 1996, with Cadence Design Systems, San Jose, CA; from 1997 to 2000, with Bell Laboratories Research, Murray Hill, NJ; from 2000 to 2002, with CeLight, Silver Spring, MD (start-up in optical communications); in 2002 and 2005, with the Research Laboratory for Electronics, Massachusetts Institute of Technology, Cambridge, MA; and from 2009 to 2010, with the University of California, Berkeley, as a Visiting Professor. From 2002 to 2007, he was with the Department of Electrical and Electronics Engineering, Koc University, Istanbul, Turkey, as an Assistant Professor and has been an Associate Professor since 2008. The work he did at Bell Laboratories and CeLight is the subject of six patents. He has co-authored two books in the areas of nonlinear-noise analysis and analog-design methodologies and published around 40 articles in journals and conferences. His current research interests include computational prototyping of electronic and opto-electronic systems, numerical modeling and analysis, stochastic dynamical systems, and noise in nonlinear electronic, optical, communication and biological systems.

Dr. Demir was the recipient of several best paper awards: the 2002 Best of International Conference on Computer-Aided Design Award, 20 years of Excellence in CAD Award, 2003 IEEE/Association for Computing Machinery William J. McCalla ICCAD Best Paper Award, and 2004 IEEE Circuits and Systems Society Guillemin-Cauer Best Paper Award. In 1991, he was the recipient of the Regents Fellowship and the Eugene-Mona Fay Gee Scholarship from the University of California, and was selected to be an Honorary Fellow of the Scientific and Technological Research Council of Turkey (TUBITAK). In 2003, he was selected by the Turkish Academy of Sciences to receive the Distinguished Young Scientist Award. He received a TUBITAK Career Award in 2005 and the TUBITAK Young Scientist Award in 2007. In 2009, he was awarded the 2219 Research Fellowship by TUBITAK for his sabbatical year at Berkeley.

Serdar Tasiran (S'92–M'98) received the B.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 1991 and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1995 and 1998, respectively.

From 1998 to 2000, he was a Research Scientist with the Gigascale Systems Research Center, San Jose, CA, a consortium of leading U.S. research universities and electronics companies. From 2000 to 2003, he was a Research Scientist with the Systems Research Center, Palo Alto, CA, which was part of Digital Equipment Corporation, Maynard, MA, Compaq, Palo Alto, and HP, Palo Alto.

Since 2003, he has been with the Department of Computer Engineering, Koc University, Istanbul, Turkey. His visiting appointments include internships with Bell Laboratories, Murray Hill, NJ, in 1995 and 1996, Visiting Professor positions with the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 2003, with the Microsoft Research, Redmond, WA, from 2003 to 2007, and with the Research Laboratory for Electronics, Palo Alto, CA, in 2005. He has published some 40 journal and conference papers on topics including circuit timing analysis and verification, optoelectronics, information visualization, haptics, simulation-based validation methods, validation coverage metrics, and formal verification of concurrent software and multiprocessor hardware. His research focus is the application of formal and algorithmic methods to system verification and validation, particularly to concurrent software and hardware. His broader current research interests include the synthesis, verification, and performance analysis and optimization of hardware and software systems.

Dr. Tasiran was an Honorary Fellow of the Scientific and Technological Research Council of Turkey (TUBITAK) and a recipient of the North Atlantic Treaty Organization Science Fellowship in 1991. In 1992, he received the Eugene-Mona Fay Gee Fellowship from the University of California, Berkeley. In 2005, he received a TUBITAK Career Award.