Advances in Computation of the Maximum of a Set of Gaussian Random Variables

Debjit Sinha, *Member, IEEE*, Hai Zhou, *Senior Member, IEEE*, and Narendra V. Shenoy, *Senior Member, IEEE*

***Abstract*—This paper quantifies the approximation error when results obtained by Clark (*Oper. Res.*, vol. 9, p. 145, 1961) are employed to compute the maximum (max) of Gaussian random variables, which is a fundamental operation in statistical timing. We show that a finite lookup table can be used to store these errors. Based on the error computations, approaches to different orderings for pairwise max operations on a set of Gaussians are proposed. Experimental results show accuracy improvements in the computation of the max of multiple Gaussians, in comparison to the traditional approach. In addition, we present an approach to compute the tightness probabilities of Gaussian random variables with dynamic runtime-accuracy tradeoff options. We replace required numerical computations for their estimations by closed form expressions based on Taylor series expansion that involve table lookup and a few fundamental arithmetic operations. Experimental results demonstrate an average speedup of 2× using our approach for computing the maximum of two Gaussians, in comparison to the traditional approach, without any accuracy penalty.**

***Index Terms*—Computer-aided design (CAD), Gaussian approximation, statistical timing, very large-scale integration (VLSI).**

I. Introduction

AN INCREASING significance of variability, as well as an increase in the number of sources of variation in modern very large-scale integrated (VLSI) circuits, necessitate statistical approaches to timing analysis and optimization. Analytical approaches to statistical static timing analysis have thus emerged as an active research topic [3]–[10]. Many recent works in the literature consider circuit component delays as Gaussian random variables since this facilitates fast analytical evaluation. Chang and Sapatnekar propose a statistical timing analysis approach under this assumption which considers spatial correlations [8]. A timing analysis algorithm that accounts for correlations and accommodates dominant interconnect coupling is proposed by Le *et al.* in [9]. A first-order incremental block-based statistical timing analyzer is presented by Visweswariah *et al.* in [3].

Propagation of Gaussian distributions in block-based statistical timing analysis involves operations like add and max. It is required that the output of these operations be a Gaussian for further propagation. An add operation on Gaussians yields another Gaussian and is accurate. However, the max of multiple Gaussians is not a Gaussian, and approximating its distribution with that of a Gaussian introduces inaccuracies. Results obtained by Clark [2] are used to approximate the max of two Gaussians with another Gaussian (by matching the first two moments of their distributions). For multiple Gaussians, the max operation is performed a pair at a time. Each of these pairwise operations introduces errors by approximating the resulting distribution with a Gaussian. These approximation errors can propagate and affect accuracy. We observe that the final loss in accuracy during the max of multiple Gaussians is dependent on the order in which pairwise max operations are performed. Prior work does not describe the impact of ordering on the inaccuracy of the process.

Our contributions in this paper are summarized as follows.

1) We quantify the error in the approximation of the max of two arbitrary Gaussians with a Gaussian. The closed form expression for the distribution of the true max is derived and is used to quantify the approximation error.
2) We present a transformation to obtain the max of two random variables from the max of a new pair of derived random variables, parameters of which can be bounded. In addition, we show that the approximation error in the max operation is an invariant of our transformation.
3) We introduce the idea of using a finite lookup table (LUT) to store quantified approximation errors in the max operation on any Gaussian pair.
4) We study the approximation errors as functions of the given Gaussians and propose good orderings for pairwise max operations on a given set of Gaussians. The orderings attempt to reduce the loss in accuracy, without significant increase in run times.

Due to the symmetry of max and min operations, the max operation is considered in this paper. The approaches presented are extensible for the case of min operations. Experimental results show timing estimation accuracy improvements in comparison to the traditional approach.

We also present an approach to compute the tightness probability of Gaussian pairs with dynamic runtime-accuracy tradeoff options. We replace traditionally required numerical computations for these estimations by closed form expressions

Manuscript received January 25, 2006; revised May 22, 2006 and September 20, 2006. This work was supported in part by the National Science Foundation under Grant CCR-0238484. This paper was presented in part at the Proceedings International Symposium on Quality Electronic Design, pp. 306–311, 2006. This paper was recommended by Associate Editor V. Bertacco.

D. Sinha was with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA. He is now with IBM Microelectronics, Hopewell Junction, NY 12533 USA.

H. Zhou was with the Advanced Technology Group, Synopsys, Inc., Mountain View, CA 94043 USA. He is now with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: haizhou@eecs.northwestern.edu).

N. V. Shenoy is with the Advanced Technology Group, Synopsys, Inc., Mountain View, CA 94043 USA.

Digital Object Identifier 10.1109/TCAD.2007.893544

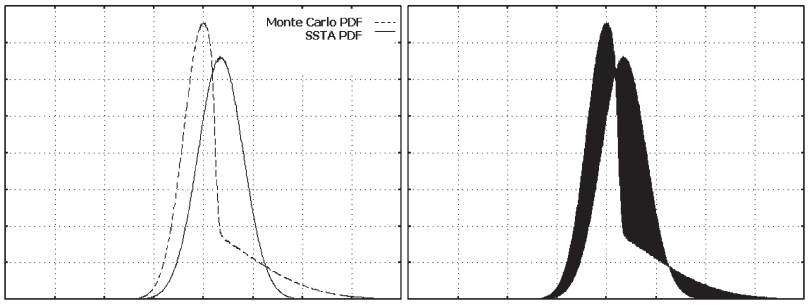

Fig. 1. Error Ξ between two random variables with given pdfs is represented by the area of the shaded region.

based on Taylor series expansion that involve table lookup and a few fundamental arithmetic operations. Experiments demonstrate an average speedup of $2\times$ using our approach for computing the max of two Gaussians, in comparison to the traditional approach, without any accuracy penalty.

The rest of this paper is organized as follows. Section II provides some preliminaries to this paper. The error in the approximation of the max of two Gaussians with another Gaussian is quantified in Section III. We present our error minimization problem in Section IV and propose smart orderings for pairwise max operations in Section V. Experimental results and conclusions are presented in Sections VI and VII, respectively.

II. Preliminaries

Based on prior work in [3], [8], and [9], we consider circuit delays as Gaussian random variables, having a joint normal distribution. A Gaussian random variable X is formally expressed as $N(\mu_X, \sigma_X^2)$, with mean μ_X and variance σ_X^2.

The add operation on Gaussians is trivial and yields another Gaussian. The max operation, on the other hand, is intricate, and for a given set of Gaussians, performed a pair at a time. We next show the moment matching approach (based on Clark [2]) to compute the max of two Gaussians X and Y. ρ is used to represent the correlation coefficient between X and Y. We define the following:

$$\phi(x) \triangleq \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \tag{1}$$

$$\Phi(y) \triangleq \int_{-\infty}^{y} \phi(x)dx \tag{2}$$

$$a \triangleq \left(\sigma_X^2 + \sigma_Y^2 - 2\rho\sigma_X\sigma_Y\right)^{1/2} \tag{3}$$

$$\alpha \triangleq \frac{\mu_X - \mu_Y}{a}. \tag{4}$$

The mean μ_Z and variance σ_Z^2 of $Z \triangleq \max(X, Y)$ is expressed analytically as follows:

$$\mu_Z = \mu_X\Phi(\alpha) + \mu_Y\Phi(-\alpha) + a\phi(\alpha) \tag{5}$$

$$\sigma_Z^2 = \left(\sigma_X^2 + \mu_X^2\right)\Phi(\alpha) + \left(\sigma_Y^2 + \mu_Y^2\right)\Phi(-\alpha) + (\mu_X + \mu_Y)a\phi(\alpha) - \mu_Z^2. \tag{6}$$

The tightness probability of X over Y is denoted by $\Phi(\alpha)$ [3]. Analytical evaluations of μ_Z and σ_Z involve computing the exponential $\phi(\alpha)$ and the definite integral $\Phi(\alpha)$, in addition to some fundamental arithmetic operations. Further, if we assume that X and Y are timing random variables at the fan-in edges of a node, the corresponding node and edge criticality estimations [3] require the value of $\Phi(\alpha)$. It is therefore evident that efficient approaches to compute the exponential $\phi(\alpha)$ and definite integral $\Phi(\alpha)$ can speedup the max computation and thereby improve performance of a statistical timer. Faster statistical timing analysis helps improving the run time for traditionally slow statistical timing optimization, even when the criticality concepts are not employed.

Z is traditionally approximated to a Gaussian variable $Z_{\rm G} \sim N(\mu_Z, \sigma_Z^2)$ for delay propagation. The first- and second-order moments of Z are matched to obtain $Z_{\rm G}$, while the higher order moments of Z are ignored. This is the first and foremost source of inaccuracy in the approach. The nonlinearity of the max operation causes Z to have an asymmetric density function. However, $Z_{\rm G}$ has a symmetric density function. We quantify the error introduced during the above approximation in the following section.

III. Approximation Errors in the max Operation

A. Error Definition

A formal comparison of two given distributions requires a metric that quantifies the dissimilarity (or similarity) between them. Given two random variables X and Y, and their probability density functions (pdfs) φ_X and φ_Y, respectively, we quantify the dissimilarity or the error Ξ_{XY} between the variables as the total area under the nonoverlapping region of their pdfs. This is formally expressed as

$$\Xi_{XY} \triangleq \int_{-\infty}^{\infty} |\varphi_X(t) - \varphi_Y(t)|\, dt. \tag{7}$$

Fig. 1 shows pdfs of two random variables, the dissimilarity between which we wish to quantify. The area of the shaded region represents the error between them. Since the area under each pdf is 1

$$0 \le \Xi_{XY} \le 2.$$

B. Error in Approximating the max of Two Gaussians

We consider the max of two jointly normal Gaussian distributions $Z \triangleq \max(X, Y)$, where $X \sim N(\mu_X, \sigma_X^2)$, $Y \sim N(\mu_Y, \sigma_Y^2)$, and ρ denotes their correlation coefficient. Z is approximated to a Gaussian $Z_{\rm G} \sim N(\mu_Z, \sigma_Z^2)$, after computing μ_Z and σ_Z from (5) and (6). Based on (7), we formally quantify the error introduced in this approximation as

$$\Xi_{(Z)(Z_{\rm G})} \triangleq \int_{-\infty}^{\infty} |\varphi_Z(t) - \varphi_{Z_{\rm G}}(t)|\, dt. \tag{8}$$

$\varphi_{Z_{\rm G}}$ denotes the pdf of the Gaussian $Z_{\rm G}$. Mathematically

$$\varphi_{Z_{\rm G}}(t) \triangleq \frac{1}{\sqrt{2\pi}\sigma_Z} e^{-\frac{(t-\mu_Z)^2}{2\sigma_Z{}^2}} = \frac{1}{\sigma_Z}\phi\left(\frac{t-\mu_Z}{\sigma_Z}\right). \tag{9}$$

We next present a closed form expression for $\varphi_Z(t)$ which denotes the true pdf of $\max(X, Y)$. The proof is presented in [1] (similar to [11]).

Lemma 1:

$$\varphi_Z(t) = \frac{1}{\sigma_Y}\phi\left(\frac{t-\mu_Y}{\sigma_Y}\right)\Phi\left[\frac{\left(\frac{t-\mu_X}{\sigma_X}\right) - \rho\left(\frac{t-\mu_Y}{\sigma_Y}\right)}{(1-\rho^2)^{1/2}}\right] + \frac{1}{\sigma_X}\phi\left(\frac{t-\mu_X}{\sigma_X}\right)\Phi\left[\frac{\left(\frac{t-\mu_Y}{\sigma_Y}\right) - \rho\left(\frac{t-\mu_X}{\sigma_X}\right)}{(1-\rho^2)^{1/2}}\right]. \tag{10}$$

$\Xi_{(Z)(Z_{\rm G})}$ can now be evaluated from (9) and (10) using numerical integration.

Since the correlation between Z and an arbitrary Gaussian X' is identical to the correlation between $Z_{\rm G}$ and X' [2], we do not explicitly consider correlation in $\Xi_{(Z)(Z_{\rm G})}$. In addition, $\Xi_{(Z)(Z_{\rm G})}$ is an upper bound on the absolute maximal difference between the cumulative distribution functions (CDFs) at any point of Z and $Z_{\rm G}$.

C. Error as a Function of Bounded Parameters

Given an arbitrary nonnegative real number λ, and another arbitrary real number θ, we consider the following two properties of a generic max operation.

1) Scaling property

$$\max(\lambda X, \lambda Y) = \lambda \cdot \max(X, Y).$$

2) Shift-invariance property

$$\max(X + \theta, Y + \theta) = \max(X, Y) + \theta.$$

We consider Gaussians $X \sim N(\mu_X, \sigma_X^2)$ and $Y \sim N(\mu_Y, \sigma_Y^2)$, with correlation coefficient ρ. Without any loss of generality, we assume $\sigma_X \geq \sigma_Y$. Application of the above properties on $\max(X, Y)$ (shifting by μ_X, and subsequent scaling by σ_X) results in the following:

$$\max(X, Y) = \mu_X + \sigma_X \cdot \max(X', Y') \tag{11}$$

where

$$X' \sim N\left(\mu_{X'}, \sigma_{X'}^2\right) = \frac{X - \mu_X}{\sigma_X} = N(0, 1)$$

$$Y' \sim N\left(\mu_{Y'}, \sigma_{Y'}^2\right) = \frac{Y - \mu_X}{\sigma_X} = N\left(\frac{\mu_Y - \mu_X}{\sigma_X}, \frac{\sigma_Y^2}{\sigma_X^2}\right).$$

ρ' denotes the correlation coefficient between X' and Y'. Since $\sigma_{Y'} = \sigma_Y/\sigma_X$, and $\sigma_X \geq \sigma_Y$, it is evident that

$$0 \leq \sigma_{Y'} \leq 1. \tag{12}$$

Lemma 2: $\rho' = \rho$.

Proof: The covariance (cov) of two Gaussians is independent of their means and is directly proportional to their standard deviations. Since $\sigma_{X'} = \sigma_X/\sigma_X$ and $\sigma_{Y'} = \sigma_Y/\sigma_X$, we have $\text{cov}(X', Y') = \text{cov}(X, Y)/\sigma_X^2$. From definition, $\rho = \text{cov}(X, Y)/\sigma_X\sigma_Y$

$$\rho' = \frac{\text{cov}(X', Y')}{\sigma_{X'}\sigma_{Y'}} = \frac{\frac{\text{cov}(X,Y)}{\sigma_X{}^2}}{1 \cdot \frac{\sigma_Y}{\sigma_X}} = \rho.$$

■

Based on the definitions in (1)–(4), we define the following:

$$a' \triangleq \left(\sigma_{X'}^2 + \sigma_{Y'}^2 - 2\rho'\sigma_{X'}\sigma_{Y'}\right)^{1/2} \tag{13}$$

$$\alpha' \triangleq \frac{\mu_{X'} - \mu_{Y'}}{a'}. \tag{14}$$

Lemma 3: $\alpha' = \alpha$.

Proof: From (3) and (13), we have $a' = a/\sigma_X$. Thus

$$\alpha' = \frac{\mu_{X'} - \mu_{Y'}}{a'} = \frac{-\mu_{Y'}}{a'} = \frac{-\frac{\mu_Y - \mu_X}{\sigma_X}}{\frac{a}{\sigma_X}} = \alpha.$$

■

We denote the error in approximating $Z' \triangleq \max(X', Y')$ with a Gaussian $Z'_{\rm G} \sim N(\mu_{Z'}, \sigma_{Z'}^2)$ to be $\Xi_{(Z')(Z'_{\rm G})}$ and prove that it is exactly equal to the error $\Xi_{(Z)(Z_{\rm G})}$ in approximating $Z = \max(X, Y)$ with Gaussian $Z_{\rm G} \sim N(\mu_Z, \sigma_Z^2)$. The proof of the following theorem is presented in Appendix I.

Theorem 1:

$$\Xi_{(Z')\left(Z'_{\rm G}\right)} = \Xi_{(Z)(Z_{\rm G})}. \tag{15}$$

The approximation error in the max of any two Gaussians can thus be estimated from the approximation error in the max of the derived Gaussians, one of which is the unit normal Gaussian. The error is therefore a function of $\mu_{Y'}$, $\sigma_{Y'}$, and $\rho'(= \rho)$. Since $\alpha'(= \alpha)$ is a function of $\mu_{Y'}$, the error can be expressed as a function of α, σ'_Y, and ρ. It is known that $\phi(\alpha) \approx 0$ for $|\alpha| \geq 4$, $\Phi(\alpha) \approx 0$ for $\alpha \leq 4$, and $\Phi(\alpha) \approx 1$ for $\alpha \geq 4$. Consequently, for $|\alpha| \geq 4$, $\max(X, Y)$ almost identically resembles the dominating Gaussian [3]. The approximation error in this case is negligible. Thus, the region of interest

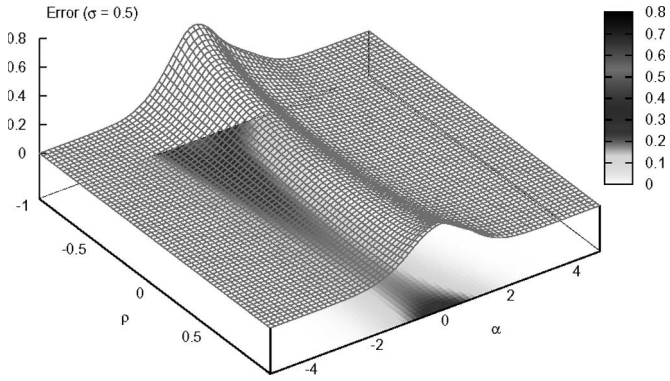

Fig. 2. $\Xi_{(Z')(Z'_{\mathrm{G}})}$ as a function of ρ and α ($\sigma_{Y'} = 0.5$).

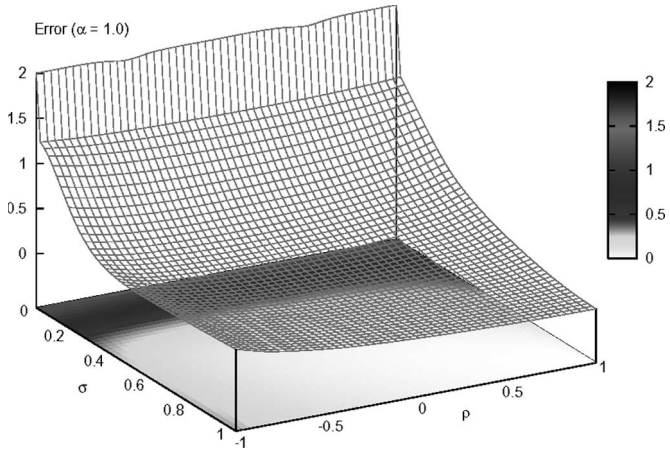

Fig. 3. $\Xi_{(Z')(Z'_{\mathrm{G}})}$ as a function of $\sigma_{Y'}$ and ρ ($\alpha = 1.0$).

for the parameters that affect the approximation error can be bounded as

$$\begin{aligned} -4 \leq\ & \alpha \leq 4 \\ 0 \leq\ & \sigma_{Y'} \leq 1 \quad \text{from (12)} \\ -1 \leq\ & \rho \leq 1 \quad \text{from definition.} \end{aligned}$$

Experiments are performed to study the dependence of the approximation error $\Xi_{(Z')(Z'_{\mathrm{G}})}$ on the above parameters. It is observed that $\Xi_{(Z')(Z'_{\mathrm{G}})}$ decreases when either of the Gaussians dominates the other, that is, $|\alpha| \geq 3$ and increases for Gaussians that contribute almost equally to the max, that is, for α in the neighborhood of 0. $\Xi_{(Z')(Z'_{\mathrm{G}})}$ is found to increase with decreasing $\sigma_{Y'}$ and is convex with respect to the correlation coefficient. Figs. 2–4 show the surface plots of $\Xi_{(Z')(Z'_{\mathrm{G}})}$ as functions of α, $\sigma_{Y'}$, and ρ. The presented plots reveal that while the $\max$ of two Gaussians can be very well approximated with a Gaussian in some cases, the approximation in other cases yields large errors.

Zhang *et al.* [12] present plots of the skewness of the maximum of two Gaussians for different Gaussian parameters. Their plots are similar to ours in the sense that they depict significant errors in approximating the max of two Gaussians with a Gaussian, when the two original Gaussians have similar means, but very different variances, and also when these two Gaussians have similar means and variances but have large negative correlation.

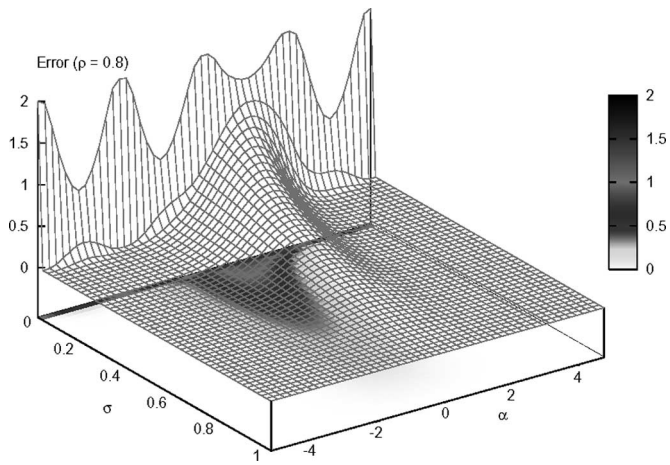

Fig. 4. $\Xi_{(Z')(Z'_{\mathrm{G}})}$ as a function of $\sigma_{Y'}$ and α ($\rho = 0.8$).

D. LUT for Error Storage

The computations for the mean and variance of the max of two Gaussians involve the evaluation of a definite integral $\Phi(\alpha)$ and an exponential $\phi(\alpha)$. Numerical computations for their accurate estimation are CPU expensive. We consider the infinite Taylor series expansion [13] about a point k of $\Phi(\alpha)$

$$\Phi(\alpha) = \Phi(k) + \Phi^{(1)}(k)(\alpha - k) + \frac{\Phi^{(2)}(k)}{2!}(\alpha - k)^2 + \cdots + \frac{\Phi^{(n)}(k)}{n!}(\alpha - k)^n + \cdots. \quad (16)$$

$\Phi^{(n)}(k)$ and $\phi^{(n)}(k)$ represent the nth derivatives of $\Phi(k)$ and $\phi(k)$, respectively. We observe that the nth derivative of $\Phi(\alpha)$ [and $\phi(\alpha)$], for any $n > 0$ is a product of $\phi(\alpha)$ and a polynomial in α

$$\begin{aligned} \Phi^{(1)}(\alpha) &= \phi(\alpha) \\ \Phi^{(2)}(\alpha) &= \phi^{(1)}(\alpha) = \phi(\alpha)(-\alpha) \\ \Phi^{(3)}(\alpha) &= \phi^{(2)}(\alpha) = \phi(\alpha)(\alpha^2 - 1) \\ \Phi^{(4)}(\alpha) &= \phi^{(3)}(\alpha) = \phi(\alpha)(-\alpha)(\alpha^2 - 3) \\ \Phi^{(5)}(\alpha) &= \phi^{(4)}(\alpha) = \phi(\alpha)(\alpha^4 - 6\alpha^2 + 3) \\ &\vdots \\ \Phi^{(n+1)}(\alpha) &= \phi^{(n)}(\alpha) = \phi(\alpha)P_n(\alpha) \end{aligned}$$

where $P_n(\alpha)$ is a polynomial of order n in α, $\forall n \geq 0$. It is observed that $P_n(\alpha) = (-1)^n(2)^{-(n/2)}H_n(\alpha/\sqrt{2})$, where H_n denotes the Hermite polynomial [14], [15]. A closed form expression for $\Phi^{n+1}(\alpha)$ is now expressed as follows (Proof in Appendix II):

$$\Phi^{(n+1)}(\alpha) = \phi^{(n)}(\alpha) = \phi(\alpha)(-1)^n n! \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^i \alpha^{n-2i}}{2^i i!(n-2i)!}. \quad (17)$$

We use the above Taylor series expansion to compute $\Phi(\alpha)$ and $\phi(\alpha)$ in the region $|\alpha| < 4$. We propose to precompute $\Phi(k)$ and $\phi(k)$ for multiple values of $k \in [0, 4)$[1] and store the values in two LUTs. Thereafter, $\Phi(\alpha)(\phi(\alpha))$ is computed by a table lookup on the closest k in the vicinity of α to obtain $\Phi(k)(\phi(k))$, followed by a finite Taylor series expansion about k. For a uniformly sampled LUT with step-size p, it is observed that a very high degree of accuracy can be obtained by expanding a few (typically 3 to 4) terms of the Taylor series expansion. We can obtain any desired accuracy in the computation of $\Phi(\alpha)$ and $\phi(\alpha)$ by either decreasing p while keeping n constant or by increasing n for a given p. A discussion on the Taylor series expansion truncation-error bound [in (24)] and an application is presented in Appendix III.

We extend this idea to using a LUT for storing quantified approximation errors next. Given that the approximation error for any two Gaussians is a function of three bounded parameters for all practical purposes, we accurately evaluate approximation errors at discrete points in the bounded space of these parameters. These evaluated errors are stored in a 3-D finite LUT. Since time is not a constraint in construction of the table, the LUT can be created with as much accuracy as desired. The error estimation for a given point is evaluated from (8)–(10). Approximation error estimation for a max operation is now performed very efficiently by a simple transformation to evaluate $\sigma'_Y = \sigma_Y/\sigma_X$ and a subsequent table lookup based on σ'_Y, α, and ρ. Our motivation toward constructing this LUT is that we could employ this constant time approximation error estimation for smarter pairwise max operations while evaluating the max of multiple Gaussians. We discuss further details in Section V.

IV. Error Minimization Problem

We consider the max operation on n given Gaussian random variables $X_1, X_2, \ldots, X_n$, such that

$$X_M \triangleq \max(X_1, X_2, \ldots, X_n).$$

Pairwise max operations are performed on the given Gaussians to yield a Gaussian $X_{\rm G} \sim N(\mu_{X_{\rm G}}, \sigma^2_{X_{\rm G}})$, which is used to approximate X_M. The loss in accuracy of the final result is dependent on the ordering of the pairwise max operations. This is because the introduced inaccuracy for each pairwise max operation is a function of the Gaussian parameters of the pair themselves and can accumulate or get reduced. The max operation on n Gaussians is analogous to the construction of a binary tree with n leaves such that each internal node computes the max of its two children. We refer to this tree as a Max Binary Tree (MBT) in the rest of this paper.

Given n Gaussians, the $\max\ (X_M)$ of which we want to estimate and approximate with $X_{\rm G}$, the Error minimization problem is to create an MBT that yields some $X_{\rm G}$ at its root such that $\Xi_{X_M X_{\rm G}}$ is minimized.

According to Knuth [16], the total number of different labeled oriented binary trees with n leaves is $\binom{2n-1}{n-1}(2n-2)!/2^{n-1}$. In an MBT, only the leaves are labeled. Therefore, the total number of different MBTs is

$$\frac{\binom{2n-1}{n-1}(2n-2)!}{2^{n-1}(n-1)!} > (2n-1)^{n-1}.$$

An exhaustive enumeration is thus prohibitive in solving this problem. Consequently, we consider good MBT construction approaches for error reduction.

- Algorithm: **Simple MBT**
- Input: Queue of n Gaussians $Q = [X_1, X_2, \ldots, X_n]$
- Output: $X_G^{Simple} \approx$ maximum of $(X_1, \ldots, X_n)$
- begin
  1) $X_G \leftarrow$ Dequeue(Q)
  2) while ($Q \neq$ NULL) {
  3) $\quad X_G \leftarrow max(X_G,$ Dequeue(Q))
  4) }
  5) return X_G
- end

Fig. 5. Algorithm for simple MBT construction.

V. Intelligent MBT Construction Approaches

In this section, we present novel approaches for constructing good MBTs based on the study in the previous sections. We assume that a max operation and other fundamental operations on Gaussian random variables take $\Theta(1)$ time in complexity analysis. Pseudocodes of all approaches are presented. We assume that the input to each MBT construction approach is a queue (or set) of n Gaussians, and the output is a Gaussian $X_{\rm G}$, denoting their maximum.

A. Simple MBT

This approach constructs the MBT as a skewed binary tree. A max is performed on two of the n given Gaussians to yield a new approximated Gaussian. Another max operation follows, which evaluates the max of the previously evaluated maximum and one of the remaining $(n-1)$ given Gaussians. This process is repeated $(n-1)$ times to obtain the max of the n given Gaussians. The complexity of this approach is $\Theta(n)$, and its pseudocode is presented in Fig. 5.

B. Partition MBT

The partition MBT approach reduces the depth of the approximation errors accumulated in $(n-1)$ stages of the previous approach to $\log\ n$, by constructing a balanced binary tree. The given set of Gaussians is randomly partitioned into two subsets. The subsets are then further bipartitioned, and the process is done recursively, until each subset contains no more than two Gaussians. A max operation is then performed on the Gaussians in each subset. The results are now propagated backward to evaluate the max of these values bottom up, until we reach the root of the MBT. Fig. 6 presents the pseudocode of a $\Theta(n)$ complexity algorithm that constructs a partition MBT.

[1] It is known that $\Phi(-k) = 1 - \Phi(k)$ and $\phi(-k) = \phi(k)$.

- Algorithm: **PARTITION MBT**
- Input: Queue of n Gaussians $Q = [X_1, X_2, \ldots, X_n]$
- Output: $X_G^{Partition} \approx$ maximum of $(X_1, \ldots, X_n)$
- begin
  1) $X_G \leftarrow$ DEQUEUE(Q)
  2) while ($Q \neq$ NULL) {
  3) ENQUEUE(Q, X_G)
  4) $X_i \leftarrow$ DEQUEUE(Q)
  5) $X_j \leftarrow$ DEQUEUE(Q)
  6) $X_G \leftarrow max(X_i, X_j)$
  7) }
  8) return X_G
- end

Fig. 6. Algorithm for partition MBT construction.

C. Sort MBT

This approach initially sorts the given set of Gaussians in a nondecreasing order of their mean values. A simple MBT construction approach is next employed on the sorted list to evaluate the max. The complexity of this approach is $\Theta(n(\log\ n))$.

D. Greedy MBT

This MBT construction approach involves a greedy heuristic to reduce approximation errors. Based on the study in the previous sections, the greedy MBT approach iteratively computes the max of two Gaussians from the given set such that the approximation error for that pair is the least in comparison to all other pairs. The computed max is then returned to the original set, and the process continues for $(n-1)$ similar iterations. The method is analogous to a graph reduction problem. We consider a fully connected graph with n nodes, each representing a given Gaussian. Edges of the graph contain weights that denote the approximation error in the max operation of the pair of nodes it joins. Adjacent nodes of the edge with the least weight are combined using a max operation into a single new node, and edge weights are incrementally recomputed. This process is repeated $(n-1)$ times. The graph now contains only a single node that denotes the approximated max of the n given Gaussians. An error LUT facilitates edge weight computations in constant time. We present the pseudocode of an efficient implementation of the greedy MBT construction approach in Fig. 7, which employs the heap data-structure [17], and has a complexity of $\Theta(n^2 \log\ n)$. The greedy MBT approach reduces given identical Gaussians (those having same means, variances and $\rho = 1$) to a single one.

The error in approximation of the max of adjacent nodes of a given edge can also be evaluated analytically from higher order moments of the max (in a way similar to that proposed in [18]). An alternate metric for any edge weight could be the skewness of the max of its adjacent nodes.

E. Cluster MBT

The cluster MBT is constructed as a combination of the partition MBT and the greedy MBT approaches. Following a greedy strategy, a max operation is performed on that Gaussian pair whose error table lookup yields the minimal approximation

- Algorithm: **GREEDY MBT**
- Input: Set of n Gaussians $S = \{X_1, X_2, \ldots, X_n\}$
- Output: $X_G^{Greedy} \approx$ maximum of $(X_1, \ldots, X_n)$
- begin
  1) $num \leftarrow index \leftarrow n$
  2) Lookup $\Xi_{X_iX_j}\ \forall (X_i, X_j) \in (S \times S) : j > i$, store tuples $\langle i, j \rangle$ in a MIN-HEAP H (using $\Xi_{X_iX_j}$ values as the comparison key)
  3) while($num \neq 1$) {
  4) $\langle i, j \rangle \leftarrow$ HEAP-EXTRACT-MIN(H)
  5) if ($X_i \in S\ \wedge X_j \in S$) {
  6) $index \leftarrow index + 1$
  7) $X_{index} \leftarrow max(X_i, X_j)$
  8) $S \leftarrow S - \{X_i, X_j\}$
  9) foreach $X_k \in S$ {
  10) Lookup $\Xi_{X_{index}X_k}$
  11) MIN-HEAP-INSERT$(H, \langle index, k \rangle)$ (using $\Xi_{X_{index}X_k}$ as comparison key)
  12) }
  13) $S \leftarrow S \bigcup \{X_{index}\}$
  14) $num \leftarrow num - 1$
  15) }
  16) }
  17) return $X_G \leftarrow$ only element in S
- end

Fig. 7. Algorithm for greedy MBT construction.

error among all other pairs. However, the computed max is not sent back to the set of the given Gaussians as in the previous approach. A new pair is selected from the original set for the max operation iteratively in a similar manner, until, no more than one Gaussian is left. The remaining Gaussian (if any) is added to the set of the computed max variables. We denote this process of reducing a set of n Gaussians to a set of $\lceil n/2 \rceil$ Gaussians as a REDUCE step.

The process restarts with the computed max distributions as the given set of Gaussians henceforth and continues until convergence. This approach ensures that the constructed MBT is balanced and tries to reduce accumulation of approximation errors by constraining the maximum depth of the tree to $(\log\ n)$. The cluster MBT construction approach reduces identical Gaussians to fewer ones, but is guaranteed to reduce them to one only when the number of identical Gaussians is 2^k for some positive integral value of k. A pseudocode of a cluster MBT construction implementation is presented in Fig. 8. The step 2 of the REDUCE operation shown involves sorting and can be done in $O(n^2 \log n^2)$ or $O(n^2 \log n)$ time. Since the sorting is done on the edges of the fully connected n node tree, the complexity expression involves the n^2 term instead of n. Multiple calls to REDUCE add up as: $(n^2 \log n^2) + ((n/2)^2 \log(n/2)^2) + ((n/4)^2 \log(n/4)^2) \ldots$ which is $\leq 2(n^2 \log n^2)$. The complexity of the shown implementation is therefore $O(n^2 \log n)$.

VI. EXPERIMENTAL RESULTS

We present experimental results of the proposed MBT construction approaches in this section. A LUT is constructed to store approximation errors as presented in Section III-D. Experimental results presented are for a LUT having 2×10^6 entries ($100 \times 100 \times 200$ for $\sigma_{Y'}$, ρ, and α, respectively). The

- Algorithm: **CLUSTER MBT**
- Input: Set of n Gaussians $S = \{X_1, X_2, \ldots, X_n\}$
- Output: $X_G^{Cluster} \approx$ maximum of $(X_1, \ldots, X_n)$
- begin
  1) $num \leftarrow n$
  2) while($num \neq 1$) {
  3) $S \leftarrow$ REDUCE(S)
  4) $num \leftarrow \lceil \frac{num}{2} \rceil$
  5) }
  6) return $X_G \leftarrow$ only element in S
- end

- Algorithm: **REDUCE**
- Input: Set of k Gaussians $S = \{X_1, X_2, \ldots, X_k\}$
- Output: Reduced set S' of $\lceil \frac{k}{2} \rceil$ Gaussians
- begin
  1) $S' \leftarrow \{\}$
  2) Lookup $\Xi_{X_iX_j}$ $\forall (X_i, X_j) \in (S \times S) : j > i$, sort them in a non-decreasing order, and store tuples $\langle i, j \rangle$ of sorted $\Xi_{X_iX_j}$ in queue Q'
  3) while ($Q' \neq$ NULL) {
  4) $\langle i, j \rangle \leftarrow$ DEQUEUE(Q')
  5) if ($X_i \in S \ \wedge X_j \in S$) {
  6) $S' \leftarrow S' \bigcup \{max(X_i, X_j)\}$
  7) $S \leftarrow S - \{X_i, X_j\}$
  8) }
  9) }
  10) return $S' \bigcup S$
- end

Fig. 8. Algorithm for cluster MBT construction.

simulations to generate the LUT take 4 h on a Pentium 2.4-GHz machine, with 1-GB RAM. The greedy and the cluster MBT approaches use this LUT to pick Gaussian pairs.

The max operation is considered on randomly generated sets of 3 to 100 Gaussians. For each set, we compare the distribution of the max obtained from different MBT construction approaches with the distribution obtained from 10^5 Monte Carlo (MC) simulations, which we assume golden. The mean value of each Gaussian is randomly selected within a range of [1.0, 3.0]. The standard deviation of each Gaussian is constrained within 20% of its mean value and is set to about 8% of the mean value on the average. Correlation coefficients between Gaussian pairs are set randomly within the range of [−1.0, 1.0].

In statistical timing, we are primarily concerned about the estimation accuracy of specific probability points in any distribution. We define a probability point $V_{\Pr=p}$ for a random variable X as

$$V_{\Pr=p} \triangleq x : (\Pr(X \leq x) = p), \qquad p \in [0, 1].$$

We next define the error in estimation of any probability point $V^A_{\Pr=p}$ employing MBT construction approach A (could be any of the MBT construction approaches from Section V) using results obtained from MC simulations for reference as

$$E^A \triangleq \frac{\left|V^A_{\Pr=p} - V^{\mathrm{MC}}_{\Pr=p}\right|}{V^{\mathrm{MC}}_{\Pr=p}} \times 100. \tag{18}$$

To quantify the improvements in estimation accuracy of a given probability point $V_{\Pr=p}$ obtained by different MBT construction approaches over the traditional simple MBT approach, we define the gain (Δ^A) for any MBT construction approach A as

$$\Delta^A \triangleq E^{\mathrm{Simple}} - E^A. \tag{19}$$

Δ^A denotes the value by which the MBT construction approach A reduces the error E^{Simple} for a given $V_{\Pr=p}$. Table I presents error and accuracy gain values for the probability point $p = 0.5$ (the median of the max distribution) while evaluating the max of different sets of Gaussians. Since we approximate the max with a Gaussian, we loosely refer to the $p = 0.5$ point as the point for error estimation in the mean. For each set, we present the number of Gaussians in that set (N), the average and maximal error in mean estimation of the max using the simple MBT construction approach ($E^{\mathrm{Simple}}_{\mathrm{Avg}}$ and $E^{\mathrm{Simple}}_{\mathrm{Max}}$, respectively) over 1000 runs on randomly generated data sets within bounds mentioned earlier, and the average and maximal gains obtained from various MBT construction approaches (Δ_{Avg} and Δ_{Max}, respectively). As an example, for the set of nine Gaussians, we observe from Table I that the simple MBT construction approach yields an average and a maximal absolute error of 0.208% and 1.941% in mean estimation, respectively. We further observe that the partition MBT construction approach reduces this error (or improves the accuracy) on the average by 0.028%, which implies that the average error in mean estimation for the partition MBT approach is $0.208\% - 0.028\% = 0.18\%$. We also observe that in the best case, the greedy MBT construction approach reduces the maximal error in simple MBT mean estimation by $\Delta^{\mathrm{Greedy}}_{\mathrm{Max}} = 1.712\%$. This implies that in at least one case where the error in the simple MBT approach or $E^{\mathrm{Simple}}_{\mathrm{Max}} \in [1.712, 1.941]\%$, the error in the greedy MBT construction approach on that same set is reduced to the range of $[(1.712 - 1.712), (1.914 - 1.712)]\% = [0, 0.202]\%$.

Tables II and III present similar results for the probability points $p = 0.95$ and $p = 0.998$, respectively. From these tables, we observe that the average error in estimation of the above probability points using the simple MBT approach is $\leq 2\%$ and usually $\leq 1\%$. It is thus natural that improvements obtained by the other MBT approaches on the average are $\leq 0.7\%$ and usually $\leq 0.4\%$. The partition MBT approach seems to yield negligible accuracy gains on the average. The sort MBT approach is found to increase average estimation errors in some cases (especially for $p = 0.95$ and $p = 0.998$). We expect better results when the sum of the mean and thrice the sigma are used as the metric for sorting in the sort MBT approach instead of sorting by the mean only. Both the cluster and greedy MBT approaches yield relatively significant accuracy gains. From the extreme cases, we observe that $E^{\mathrm{Simple}}_{\mathrm{Max}} \leq 6.914\%$ $(N = 12, p = 0.998)$. In such situations, the proposed MBT approaches seem to be very effective in reducing the high errors; a best case reduction by 6.832% is observed, which is a very significant gain.

Table IV presents similar error and accuracy gain results for standard deviation estimation. Average and maximal errors in standard deviation estimation using the simple MBT approach

TABLE I
PERCENTAGE ERROR AND ACCURACY GAIN RESULTS IN $V_{\mathrm{Pr}=0.5}$(Mean)

N	Simple		Partition		Sort		Cluster		Greedy	
	E_{Avg}	E_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}
3	0.123	1.024	0.001	0.878	-0.001	0.878	0.013	0.878	0.013	0.878
5	0.151	1.174	0.004	0.646	-0.009	0.721	0.029	0.885	0.030	0.825
7	0.162	1.184	0.003	0.695	-0.011	0.693	0.026	1.093	0.035	1.093
9	0.208	1.941	0.028	0.681	0.018	1.059	0.054	1.676	0.054	1.712
12	0.257	1.458	0.051	1.021	0.053	1.164	0.074	1.333	0.088	1.311
15	0.322	2.048	0.074	1.036	0.084	1.165	0.116	1.534	0.120	1.429
20	0.429	1.896	0.135	0.916	0.109	1.458	0.181	1.469	0.185	1.642
30	0.666	2.415	0.273	0.968	0.163	1.207	0.342	2.125	0.323	2.297
50	1.096	2.762	0.521	1.339	0.116	1.013	0.639	1.920	0.566	1.904
100	1.991	3.306	1.061	1.892	0.058	1.178	1.203	2.638	1.135	2.754

TABLE II
PERCENTAGE ERROR AND ACCURACY GAIN RESULTS IN $V_{\mathrm{Pr}=0.95}$

N	Simple		Partition		Sort		Cluster		Greedy	
	E_{Avg}	E_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}
3	0.209	2.438	-0.001	2.035	-0.04	2.035	0.046	2.035	0.046	2.035
5	0.296	2.595	0.012	2.480	-0.06	2.217	0.101	2.523	0.108	2.523
7	0.353	2.952	0.013	2.627	-0.07	2.835	0.048	2.627	0.142	2.730
9	0.439	3.221	0.034	2.983	-0.07	1.892	0.179	2.951	0.200	2.951
12	0.509	4.742	0.047	2.167	-0.03	4.626	0.151	4.543	0.228	4.637
15	0.538	4.141	0.054	2.092	-0.06	2.440	0.116	3.696	0.234	3.323
20	0.630	2.922	0.062	2.400	-0.01	2.698	0.186	2.485	0.282	2.912
30	0.711	3.674	0.111	3.048	0.065	2.500	0.197	3.102	0.317	3.405
50	0.875	2.915	0.248	2.597	0.119	2.308	0.337	2.606	0.434	2.461
100	1.173	3.411	0.559	2.912	0.064	3.120	0.563	2.859	0.681	3.250

TABLE III
PERCENTAGE ERROR AND ACCURACY GAIN RESULTS IN $V_{\mathrm{Pr}=0.998}$

N	Simple		Partition		Sort		Cluster		Greedy	
	E_{Avg}	E_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}
3	0.300	3.388	0.000	2.923	-0.07	2.923	0.074	3.095	0.074	3.095
5	0.439	4.128	0.019	3.933	-0.10	3.098	0.158	4.016	0.168	4.016
7	0.543	4.753	0.025	3.869	-0.11	4.153	0.079	3.869	0.234	4.225
9	0.676	5.540	0.054	5.130	-0.11	2.725	0.291	5.060	0.319	5.082
12	0.791	6.914	0.073	3.115	-0.06	6.821	0.245	6.709	0.358	6.832
15	0.841	6.262	0.082	3.230	-0.11	3.530	0.183	5.621	0.363	4.617
20	0.948	4.693	0.060	4.094	-0.06	4.300	0.264	3.450	0.406	4.164
30	1.034	5.344	0.077	4.114	0.058	3.677	0.229	4.584	0.412	5.070
50	1.134	5.141	0.128	3.983	0.146	4.464	0.344	3.638	0.416	4.681
100	1.139	4.173	0.109	3.689	0.201	3.973	0.302	3.791	0.314	3.898

TABLE IV
PERCENTAGE ERROR AND ACCURACY GAIN RESULTS IN STANDARD DEVIATION ESTIMATION

N	Simple		Partition		Sort		Cluster		Greedy	
	E_{Avg}	E_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}	Δ_{Avg}	Δ_{Max}
3	1.041	12.34	-0.02	9.967	-0.33	8.961	0.275	11.88	0.275	11.88
5	1.601	24.56	0.043	23.29	-0.50	11.72	0.589	23.92	0.618	23.92
7	2.095	19.42	0.094	14.73	-0.55	13.29	0.308	12.49	0.922	17.19
9	2.704	24.75	0.278	24.66	-0.46	15.50	1.203	24.24	1.274	24.24
12	3.162	26.16	0.330	12.95	-0.27	26.06	1.026	25.91	1.403	26.03
15	3.490	26.55	0.352	13.43	-0.49	15.98	0.849	24.04	1.448	17.13
20	3.867	18.58	0.253	15.58	-0.36	16.68	1.121	16.10	1.545	17.28
30	4.273	18.32	0.224	13.41	0.125	13.71	0.975	16.10	1.514	18.00
50	4.757	21.85	0.189	15.57	0.239	19.55	1.510	15.36	1.056	21.68
100	5.307	21.57	-0.38	11.15	0.835	17.17	1.644	17.79	0.478	14.87

are observed to be $\leq 5.307\%$ and $\leq 26.55\%$, respectively. The greedy MBT approach performs the best overall, with average and best case improvements of $\leq 1.545\%$ and $\leq 26.03\%$, respectively. Note that the comparatively cheap (in terms of run time) sort MBT approach is very effective toward error reduction in this case.

Run times are comparable for sets having up to 30 Gaussians. For the max of 50 Gaussians, run times for the simple, partition, sort, cluster, and greedy MBT approaches are found to be 0.00005, 0.00019, 0.00003, 0.002, and 0.016 s, respectively. Based on the above results, we conclude that the cluster and the greedy MBT approaches are effective in accurately evaluating the max of multiple Gaussians, and gains obtained are likely worth the performance tradeoff. Further, we demonstrate that the sort MBT construction approach is not always a better choice than the traditional simple MBT approach. In [19], we have shown that smart pairwise max operations yield accuracy gains in statistical timing analysis as well. All experiments are

performed on a Pentium 2.4-GHz 1-GB RAM machine, running Red Hat Linux 9.0.

VII. Conclusion

In this paper, we quantify the error in approximating the max of two Gaussians with another Gaussian. Based on these error computations, we propose approaches to smart orderings for pairwise max operations while evaluating the max of multiple Gaussians for improved accuracy, with application to statistical timing.

Prior research [20] has shown that the average errors in estimating the mean and standard deviation of a circuit's arrival time distribution in comparison to MC simulations are ≈1.8% and 13.7%, respectively. Similar numbers are presented in [8], which also reports average errors of ≈4.56% for $V_{\mathrm{Pr}=0.99}$ point estimation. The proposed approaches in this paper improve the accuracy in estimation of the critical probability points, as well as that of the variance, for the max of multiple Gaussians. Further, we employ smart orderings for pairwise max operations in [19], and observe accuracy gains in statistical timing analysis.

The proposed approach is amenable to being applied at one, multiple, or all levels of logic depth in timing analysis. When applied to just one level, no modifications to existing timing data-structures are required. Here, we perform a smart max of multiple Gaussians that denote timing information on the fan-ins of some gate, and then propagate the approximated result. Depending on the quantified error in intermediate stages, we could choose to propagate multiple Gaussians through a gate, if the error is above a certain threshold. This approach requires changes in the data-structures to accommodate multiple Gaussians as timing information on any circuit node. If we propagate all Gaussians, the approach falls back to path-based timing analysis. Smart pairwise max operations on a larger set of Gaussians improves the scope of accuracy gain. Consequently, we do not restrict this approach to just one level during timing analysis.

We believe that the proposed approaches would increase the accuracy in the estimation of node and edge criticalities [3], and would thereby guide statistical timing optimization better. In addition, expressions for the CDF and pdf of the true max of two Gaussians would help in accurate yield estimation, when considering both timing and power [20].

Finally, we do not advocate that the approximation error in a max operation is the only source of error in statistical timing analysis. We acknowledge that the inherent nonlinearity of a max operation could cause large errors, which our approach may not be able to reduce (for example, the error in the max of two Gaussians with similar means but having largely varying variance). In addition, inaccuracies in statistical delay modeling, and ignorance of spatial or fan-out-reconvergence correlations contribute to significant errors.

Appendix I

We consider approximating Z with a Gaussian $Z_{\mathrm{G}} \sim N(\mu_Z, \sigma_Z^2)$ and approximating $Z' \triangleq \max(X', Y')$ with a Gaussian $Z'_{\mathrm{G}} \sim N(\mu_{Z'}, \sigma_{Z'}^2)$. $\varphi_Z(t)$ and $\varphi_{Z_{\mathrm{G}}}(t)$ denote pdfs as defined earlier, and $\varphi_{Z'}(t)$ and $\varphi_{Z'_{\mathrm{G}}}(t)$ denote the pdfs of Z' and Z'_{G}, respectively. From (9) and (10), we have the following:

$$\varphi_{Z'_{\mathrm{G}}}(t) \triangleq \frac{1}{\sqrt{2\pi}\sigma_{Z'}} e^{\frac{-(t-\mu_{Z'})^2}{2{\sigma_{Z'}}^2}} \tag{20}$$

$$\varphi_{Z'}(t) \triangleq \frac{1}{\sigma_{Y'}}\phi\left(\frac{t-\mu_{Y'}}{\sigma_{Y'}}\right)\Phi\left[\frac{t-\rho\left(\frac{t-\mu_{Y'}}{\sigma_{Y'}}\right)}{(1-\rho^2)^{1/2}}\right] + \phi(t)\Phi\left[\frac{\left(\frac{t-\mu_{Y'}}{\sigma_{Y'}}\right)-\rho t}{(1-\rho^2)^{1/2}}\right]. \tag{21}$$

Lemma 4: $\mu_Z = \mu_X + \sigma_X\mu_{Z'}$ and $\sigma_Z = \sigma_X\sigma_{Z'}$.

Proof: The results are trivially derived from (5) and (6). ■

Lemma 5: $\varphi_{Z_{\mathrm{G}}}(t) = (1/\sigma_X)\varphi_{Z'_{\mathrm{G}}}(t - \mu_X/\sigma_X)$.

Proof: The result is immediate from a simple substitution in (21) and from Lemma 4. ■

Lemma 6: $\varphi_Z(t) = (1/\sigma_X)\varphi_{Z'}((t-\mu_X)/\sigma_X)$.

Proof: The result is immediate from a simple substitution in (21) which yields (10). ■

We next prove Theorem 1, that is, the error in approximating Z with Z_{G} is the same as the error in approximating Z with Z'_{G}. From our error definition in (8), and from the results obtained above, we have the following:

$$\begin{aligned}
&\Xi_{(Z')(Z'_{\mathrm{G}})} \\
&= \int_{-\infty}^{\infty} \left|\varphi_{Z'}(t) - \varphi_{Z'_{\mathrm{G}}}(t)\right| dt \\
&= \int_{-\infty}^{\infty} \left|\varphi_{Z'}\left(\frac{t-\mu_X}{\sigma_X}\right) - \varphi_{Z'_{\mathrm{G}}}\left(\frac{t-\mu_X}{\sigma_X}\right)\right| d\left(\frac{t-\mu_X}{\sigma_X}\right) \\
&= \sigma_X \int_{-\infty}^{\infty} \left|\varphi_Z(t) - \varphi_{Z_{\mathrm{G}}}(t)\right| d\left(\frac{t-\mu_X}{\sigma_X}\right) \\
&= \sigma_X \int_{-\infty}^{\infty} \left|\varphi_Z(t) - \varphi_{Z_{\mathrm{G}}}(t)\right| dt \frac{1}{\sigma_X} = \Xi_{(Z)(Z_{\mathrm{G}})}.
\end{aligned}$$

Appendix II

We prove the following using mathematical induction.

Let

$$P_n(\alpha) \triangleq (-1)^n n! \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^i \alpha^{n-2i}}{2^i i!(n-2i)!}$$

then

$$\Phi^{(n+1)}(\alpha) = \phi^{(n)}(\alpha) = \phi(\alpha)P_n(\alpha).$$

Proof: The above can be trivially verified for $n = 0$. We assume that (22) is true for a given $n = m$ and compute the coefficient of α^{m+1-2i} in $\phi^{(m+1)}(\alpha)/\phi(\alpha)$

$$\begin{aligned}
\phi^{(m)}(\alpha) &= (-1)^m \phi(\alpha) m! \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \frac{(-1)^i \alpha^{m-2i}}{2^i i!(m-2i)!} \\
&= (-1)^m \phi(\alpha) m! \\
&\quad \times \left[\frac{\alpha^n}{n!} + \cdots + \frac{(-1)^{i-1} \alpha^{m-2i+2}}{2^{i-1}(i-1)!(m-2i+2)!} \right. \\
&\qquad \left. + \frac{(-1)^i \alpha^{m-2i}}{2^i i!(m-2i)!} \cdots \right] \\
\phi^{(m+1)}(\alpha) &= \frac{d}{d\alpha} \phi^{(m)}(\alpha) \\
&= (-1)^m \phi(\alpha) m! \\
&\quad \times \left[\frac{\alpha^{n-1}}{(n-1)!} + \cdots + \frac{(-1)^{i-1} \alpha^{m-2i+1}}{2^{i-1}(i-1)!(m-2i+1)!} \right. \\
&\qquad \left. + \frac{(-1)^i \alpha^{m-2i-1}}{2^i i!(m-2i-1)!} \cdots \right] \\
&\quad + (-1)^{m+1} \phi(\alpha) m! \\
&\quad \times \left[\frac{\alpha^{n+1}}{n!} + \cdots + \frac{(-1)^{i-1} \alpha^{m-2i+3}}{2^{i-1}(i-1)!(m-2i+2)!} \right. \\
&\qquad \left. + \frac{(-1)^i \alpha^{m-2i+1}}{2^i i!(m-2i)!} \cdots \right].
\end{aligned}$$

The coefficient of α^{m+1-2i} in $\phi^{(m+1)}(\alpha)/\phi(\alpha)$ is found to be

$$\begin{aligned}
&(-1)^m m! \left[\frac{(-1)^{i-1}}{2^{i-1}(i-1)!(m-2i+1)!} \right] \\
&\quad + (-1)^{m+1} m! \left[\frac{(-1)^i}{2^i i!(m-2i)!} \right] \\
&\quad = (-1)^{m+1}(m+1)! \frac{(-1)^i}{2^i i!(m-2i+1)!}
\end{aligned}$$

and is identical to the coefficient of α^{m+1-2i} in $P_{m+1}(\alpha)$. ■

APPENDIX III

Statistical timing analysis is often used to guide timing optimization [19], [21]–[24]. Guthaus *et al.* [25] propose a gate-sizing algorithm to optimize circuit area while satisfying a given timing yield target. They employ a sensitivity metric based on slack distributions to select gates for resizing. Their sensitivity metric is computed from the node and edge criticalities [3] of the circuit. An approach to compute node and edge sensitivities considering correlations is proposed by Li *et al.* in [26]. Criticality (sensitivity) computations in both the approaches involve evaluating tightness probabilities or $\Phi(\alpha)$. We therefore realize that accurate computations of tightness probabilities are critical to statistical timing analysis and subsequently to statistical timing optimization. At the same time, it is important that these computations be efficient. Ideally, it is desirable to have an option of dynamically choosing a runtime-accuracy tradeoff point during any such computation in a timing optimization tool.

A well-defined closed form of $\phi^{(n)}(\alpha)$ from (17) facilitates the computation of $\Phi(\alpha)$ and $\phi(\alpha)$ using the Taylor series expansion about any point. For practical evaluation, the infinite series is truncated to a finite one. The sum of the terms lost is called the Lagrange remainder or the residual. When the first $n+1$ terms are retained in (16), (that is, terms $(\Phi^{(n+1)}(k)/(n+1)!)(\alpha-k)^{n+1} + \cdots$ are truncated), the Lagrange remainder R_n is expressed as the following (from [13]):

$$\begin{aligned}
R_n &\triangleq \underbrace{\int_k^\alpha \cdots \int_k^\alpha}_{n+1} \Phi^{(n+1)}(\alpha)(d\alpha)^{n+1} \\
&= \int_k^\alpha \Phi^{(n+1)}(t) \frac{(\alpha - t)^n}{n!} dt \\
&= \frac{(\alpha - k)^{n+1}}{(n+1)!} \Phi^{(n+1)}(\alpha^*) \qquad (22)
\end{aligned}$$

for some $\alpha^* \in [k, \alpha]$ (using the mean-value theorem). R_n is used as an accuracy estimate in the computation of $\Phi(\alpha)$ (and $\phi(\alpha)$). We next present the following bound.

Lemma 7:

$$\left| \Phi^{(2n+1)}(\alpha) \right| \leq \frac{(2n)!}{\sqrt{2\pi} 2^n} \qquad \forall \alpha \in (-\infty, \infty).$$

The derivative of $\Phi^{(2n+1)}(\alpha)$ is a multiple of α for any nonnegative n. It is observed that the local extremal at $\alpha = 0$ corresponds to the global maximal of $|\Phi^{(2n+1)}(\alpha)|$. Fig. 9 plots $\Phi^{(2n+1)}(\alpha)$ against α for $n = \{0, 1, 2, 3\}$. The above bound is obtained by setting $\alpha = 0$ in (22). From this result and (22), we have the following:

$$|R_{2n}| \leq \frac{(|\alpha - k|)^{2n+1}}{\sqrt{2\pi} 2^n (2n+1)}. \qquad (23)$$

The given bound is a good bound for any α in the neighborhood of 0. We observe that the bound can be severely tightened in the given region by choosing a k very close to α, such that

$$|\alpha - k| \ll 1.$$

This makes the residual decrease exponentially with n. For practical purposes, we make the following assumptions:

$$\begin{aligned}
\Phi(\alpha) &\approx 0 \qquad \forall \alpha \ \leq -7 \\
\Phi(\alpha) &\approx 1 \qquad \forall \alpha \ \geq 7 \\
\phi(\alpha) &\approx 0 \qquad \forall |\alpha| \leq 7.
\end{aligned}$$

It is known that $(0.5 - |0.5 - \Phi(\alpha)|)$ and $\phi(\alpha)$ are monotonically decreasing functions of α. Consequently, the approximation errors in the above assumptions for any $|\alpha| > 7$ must be

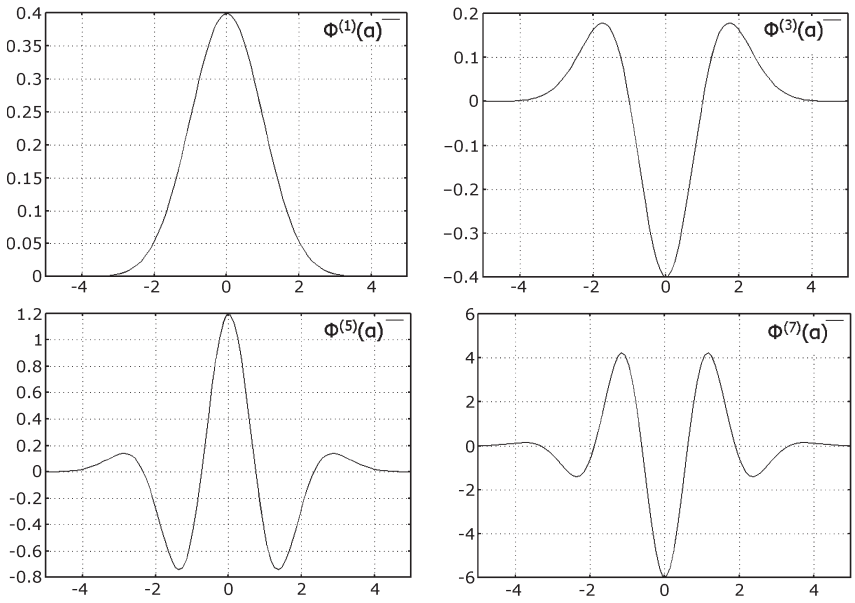

Fig. 9. Plots of $\Phi^{(2n+1)}(\alpha)$ against α for different values of n, showing global extremal at $\alpha = 0$.

TABLE V
MAX RESIDUALS R_n IN $\Phi(\alpha)$ FOR VARIOUS LUT STEP-SIZES ($p = 7.0/\#\text{Entries}$)

# Entries	p	Max Residual				
		n = 0	n = 2	n = 4	n = 6	n = 8
35	0.2000	3.98×10^{-02}	6.65×10^{-05}	1.99×10^{-07}	7.12×10^{-10}	2.77×10^{-12}
70	0.1000	1.99×10^{-02}	8.31×10^{-06}	6.23×10^{-09}	5.56×10^{-12}	5.41×10^{-15}
350	0.0200	3.98×10^{-03}	6.65×10^{-08}	1.99×10^{-12}	7.12×10^{-17}	2.77×10^{-21}
700	0.0100	1.99×10^{-03}	8.31×10^{-09}	6.23×10^{-14}	5.56×10^{-19}	5.41×10^{-24}
1050	0.0067	1.33×10^{-03}	2.46×10^{-09}	8.21×10^{-15}	3.26×10^{-20}	1.41×10^{-25}
1400	0.0050	9.97×10^{-04}	1.04×10^{-09}	1.95×10^{-15}	4.34×10^{-21}	1.06×10^{-26}
2100	0.0033	6.65×10^{-04}	3.08×10^{-10}	2.56×10^{-16}	2.54×10^{-22}	2.75×10^{-28}
2800	0.0025	4.99×10^{-04}	1.30×10^{-10}	6.09×10^{-17}	3.40×10^{-23}	2.06×10^{-29}
3500	0.0020	3.98×10^{-04}	6.65×10^{-11}	1.99×10^{-17}	7.12×10^{-24}	2.77×10^{-30}
7000	0.0010	1.99×10^{-04}	8.31×10^{-12}	6.23×10^{-19}	5.56×10^{-26}	5.41×10^{-33}

less than the approximation errors at $\alpha = 7$, which is less than 10^{-11}. Based on the desired level of accuracy, the boundary (here 7) can be increased or decreased.

For a uniformly sampled LUT with step-size p, an upper bound on the loss of accuracy in the computation of $\Phi(\alpha)$ with $2n + 1$ terms in the Taylor series expansion is given by the following:

$$|R_{2n}| \leq \frac{(\frac{p}{2})^{2n+1}}{\sqrt{2\pi}2^n(2n+1)}. \tag{24}$$

Table V presents an upper bound on the maximal residual for various number of LUT entries and number of finite Taylor series expansion terms $(n + 1)$. The choice of n in the Taylor expansion can be made dynamically based on the desired accuracy for a given LUT step-size. The table indicates that a very high level of accuracy can be reached with expanding extremely few terms in the Taylor expansion. In addition, it shows that obtaining a given accuracy result using naive table lookup $(n = 0)$ without Taylor series expansion requires a much larger table size. For example, it takes $\approx 1.7 \times 10^9$ entries to compute $\Phi(\alpha)$ with the same accuracy as with a table having 700 entries and a Taylor series expansion with only three terms.

Although we propose to perform Taylor expansion for the interval $|\alpha| < 7$, the expansion can theoretically be performed for any α outside the interval to obtain results with higher precision. However, for practical purposes, the approximation error at the boundary points is used as an accuracy estimate, and the table size is determined accordingly. This method of computing $\Phi(\alpha)$ and $\phi(\alpha)$ facilitates dynamic runtime-accuracy tradeoff options.

We implement this technique for estimating $\Phi(\alpha)$ and $\phi(\alpha)$ while evaluating the max of two Gaussians. We study average run times of these computations for over 10^6 random max operations. For fair comparison, we compute the average run time for the same set of max operations in a different implementation which uses the erf() and exp() functions from the standard UNIX math library to compute the definite integral $\Phi(\alpha)$ and the exponential $\phi(\alpha)$. We do not consider the time to create the LUTs in our comparison because they need to be computed only once. Run time comparisons demonstrate that our proposed method is faster by 8.2 times in computing $\Phi(\alpha)$ and $\phi(\alpha)$ on the average.

Table VI presents speedup results obtained in evaluating the max of two Gaussians for a different number of independent global sources of variations (N) in addition to a local independent component (similar to [3]). The run time ratio of the

TABLE VI
SPEEDUP RESULTS WHILE EVALUATING THE MAX OF TWO GAUSSIANS

N	Speed-up	N	Speed-up	N	Speed-up
1	3.21	6	2.20	12	1.86
2	3.12	8	2.12	18	1.66
4	2.82	10	2.00	24	1.50

approach employing calls to erf() and exp() to the proposed approach (speedup) is found to decrease from 3.2 for $N = 1$ to 1.86 for $N = 12$. This is expected as overheads for other computations gradually overshadow the $8.2\times$ speedup in the computation of $\Phi(\alpha)$ and $\phi(\alpha)$. To ensure the same level of accuracy, the implementation is performed for 3500 entries in the LUTs and Taylor expansion of three terms (accuracy $\geq 6.65 \times 10^{-11}$). A naive LUT would require $\approx 2 \times 10^{10}$ entries for the same accuracy.

Although the effective speedup of this technique for a single statistical timing analysis run may not seem lucrative enough, performance gains have been observed (in an industrial framework) when this approach is employed in statistical timing driven circuit optimization by gate sizing.

REFERENCES

[1] D. Sinha, H. Zhou, and N. V. Shenoy, "Advances in computation of the maximum of a set of random variables," in *Proc. Int. Symp. Quality Electron. Des.*, 2006, pp. 306–311.

[2] C. E. Clark, "The greatest of a finite set of random variables," *Oper. Res.*, vol. 9, no. 2, pp. 145–162, Mar./Apr. 1961.

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

[4] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," in *Proc. Int. Conf. Comput.-Aided Des.*, 2003, pp. 607–614.

[5] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," in *Proc. Int. Conf. Comput.-Aided Des.*, 2003, pp. 900–907.

[6] S. Bhardwaj, S. Vrudhula, and D. Blaauw, "TAU: Timing analysis under uncertainty," in *Proc. Int. Conf. Comput.-Aided Des.*, 2003, pp. 615–620.

[7] V. Khandelwal, A. Davoodi, and A. Srivastava, "Efficient statistical timing analysis through error budgeting," in *Proc. Int. Conf. Comput.-Aided Des.*, 2004, pp. 473–477.

[8] H. Chang and S. S. Sapatnekar, "Statistical timing analysis under spatial correlations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 9, pp. 1467–1482, Sep. 2005.

[9] J. Le, X. Li, and L. Pileggi, "STAC: Statistical timing with correlation," in *Proc. Des. Autom. Conf.*, 2004, pp. 343–348.

[10] D. Sinha and H. Zhou, "A unified framework for statistical timing analysis with coupling and multiple input switching," in *Proc. Int. Conf. Comput.-Aided Des.*, 2005, pp. 837–843.

[11] M. Cain, "The moment-generating function of the minimum of bivariate normal random variables," *Amer. Stat.*, vol. 48, no. 2, pp. 124–125, May 1994.

[12] L. Zhang, W. Chen, Y. Hu, and C. C. Chen, "Statistical static timing analysis with conditional linear MAX/MIN approximation and extended canonical timing model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 6, pp. 1183–1191, Jun. 2006.

[13] E. W. Weisstein, *Taylor series—Wolfram web resource.* [Online]. Available: http://mathworld.wolfram.com/TaylorSeries.html

[14] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1972.

[15] E. W. Weisstein, *Hermite polynomial—Wolfram web resource*. [Online]. Available: http://mathworld.wolfram.com/HermitePolynomial.html

[16] D. E. Knuth, "Fundamental algorithms," in *The Art of Computer Programming*, 2nd ed, vol. 1. Reading, MA: Addison-Wesley, 1973.

[17] T. H. Cormen, C. E. Leiserson, and R. H. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1989.

[18] X. Li, J. Le, P. Gopalakrishnan, and L. Pileggi, "Asymptotic probability extraction for non-normal distributions of circuit performance," in *Proc. Int. Conf. Comput.-Aided Des.*, 2004, pp. 2–9.

[19] D. Sinha, N. V. Shenoy, and H. Zhou, "Statistical timing yield optimization by gate sizing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 10, pp. 1140–1146, Oct. 2006.

[20] A. Srivastava, S. Shah, K. Agarwal, D. Sylvester, D. Blaauw, and S. Director, "Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance," in *Proc. Des. Autom. Conf.*, 2005, pp. 535–540.

[21] A. Agarwal, K. Chopra, and D. Blaauw, "Statistical timing based optimization using gate sizing," in *Proc. DATE*, 2005, pp. 400–405.

[22] A. Agarwal, K. Chopra, D. Blaauw, and V. Zolotov, "Circuit optimization using statistical static timing analysis," in *Proc. Des. Autom. Conf.*, 2005, pp. 321–324.

[23] J. Singh, V. Nookala, Z. Luo, and S. Sapatnekar, "Robust gate sizing by geometric programming," in *Proc. Des. Autom. Conf.*, 2005, pp. 315–320.

[24] M. Mani and M. Orshansky, "A new statistical optimization algorithm for gate sizing," in *Proc. Int. Conf. Comput. Des.*, 2004, pp. 272–277.

[25] M. Guthaus, N. Venkateswaran, C. Visweswariah, and V. Zolotov, "Gate sizing using incremental parameterized statistical timing analysis," in *Proc. Int. Conf. Comput.-Aided Des.*, 2005, pp. 1029–1036.

[26] X. Li, J. Le, M. Celik, and L. Pileggi, "Defining statistical sensitivity for timing optimization of logic circuits with large-scale process and environmental variations," in *Proc. Int. Conf. Comput.-Aided Des.*, 2005, pp. 844–851.

Debjit Sinha (M'06) received the B.Tech. degree (with honors) in electrical engineering from the Indian Institute of Technology, Kharagpur, India, in 2001, and the Ph.D. degree in electrical engineering and computer science from Northwestern University, Evanston, IL, in 2006.

He is currently an Advisory Engineer with IBM Microelectronics, Hopewell Junction, NY. His research interests include algorithms and computer-aided design (CAD) for very large-scale integrated (VLSI) circuits, especially related to crosstalk, and statistical timing analysis and optimization.

Hai Zhou (M'04–SM'04) received the B.S. and M.S. degrees in computer science and technology from Tsinghua University, Bejing, China, in 1992 and 1994, respectively, and the Ph.D. degree in computer sciences from the University of Texas, Austin, in 1999.

He is currently an Assistant Professor of electrical engineering and computer science with Northwestern University, Evanston, IL. Before he joined the faculty of Northwestern University, he was with the Advanced Technology Group, Synopsys, Inc. His research interests include VLSI CAD, algorithm design, and formal methods.

Dr. Zhou has served on the technical program committees of many conferences on VLSI circuits and CAD. He is a recipient of the CAREER Award from the National Science Foundation in 2003.

Narendra V. Shenoy (M'86–SM'99) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Mumbai, India, and the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley.

He is currently a Synopsys Scientist with the Advanced Technology Group at Synopsys, Inc., Mountain View, CA. He has pursued research in the areas of timing analysis, logic synthesis and optimization, retiming, physical synthesis, routing, signal integrity, extraction, and programmable fabrics. His current interests are in statistical timing analysis and optimization, and reducing variability in design.