

A Scalable Statistical Static Timing Analyzer Incorporating Correlated Non-Gaussian and Gaussian Parameter Variations

Jaskirat Singh Sachin S. Sapatnekar

Department of Electrical & Computer Engineering

University of Minnesota

Minneapolis, MN 55455

{jsingh,sachin}@ece.umn.edu

Abstract

We propose a scalable and efficient parameterized block-based statistical static timing analysis (SSTA) algorithm incorporating both Gaussian and non-Gaussian parameter distributions, capturing spatial correlations using a grid-based model. As a preprocessing step, we employ independent component analysis to transform the set of correlated non-Gaussian parameters to a basis set of parameters that are statistically independent, and principal components analysis to orthogonalize the Gaussian parameters. Given the moments of the variational parameters, we use a Padé approximation-based moment matching scheme to generate the distributions of the random variables representing the signal arrival times, and preserve correlation information by propagating arrival times in a canonical form. Our experiments reveal that for the cases, when the sensitivities of Gaussian parameters outweigh that of the non-Gaussian parameters, a Gaussian SSTA proves to be reasonably accurate. However, for the cases when the non-Gaussian parameter sensitivities dominate the Gaussians, modeling all parameters as normal leads to significant inaccuracies in the SSTA results. For both cases, our SSTA procedure is able to generate the circuit delay distributions with reasonably small prediction errors. For the ISCAS89 benchmark circuits, as compared to Monte Carlo simulations, we obtain average errors of 0.99%, 2.05%, 2.33% and 2.36%, respectively, in the mean, standard deviation, 5% and 95% quantile points of the circuit delay. Experimental results show that our procedure can handle as many as 256 correlated non-Gaussian variables in about 5 minutes of run time. For a circuit with $|G|$ gates and a layout with g spatial correlation grids, the complexity of our approach is $O(g|G|)$.

I. INTRODUCTION

As transistor and interconnect geometries shrink, the reduced level of control over the chip fabrication process results in significant levels of variation in process parameters such as the effective channel length, gate width, gate oxide thickness, dopant concentration, and interlayer dielectric thickness. These variations create randomness in the behavior of circuit-level electrical parameters, such as gate and interconnect capacitances, transistor on-resistances, threshold voltages and via resistances. The prediction of chip timing characteristics in the face of these process-driven random parameter uncertainties remains a challenging problem.

Traditionally, to safeguard against this variability, a static timing analysis (STA) procedure is employed at different process corners, and margins are introduced in the design based on the STA results. This worst case design, corresponding to the process corners, where the gate and wire delays are at their extreme levels, ensures that the design would work for any other values of gate and interconnect delays. However, with increasing levels of variations, corner-based method becomes impractical and computationally expensive. The number of process corners that must be considered grows exponentially as the number of uncertain parameters increase. Moreover, the corner-based method does not utilize any statistical information about the variations of parameters, such as the correlations between the process variables arising from the spatial proximity of the manufactured transistors on chip, or from the structural properties of the circuit such as path reconvergences, and hence can result in overly pessimistic and suboptimal designs. The results of variation-aware timing are eventually required to be used for a circuit optimization tool. Since, the multi-corner-based methodology produces overly pessimistic estimates of a circuit timing characteristics, any optimization tool using these results could lead to a design employing much more resources than necessary. This may adversely impact the other performance measures of the circuit, such as the circuit power.

As a result, the field of statistical static timing analysis (SSTA) has recently become an active area of research. An SSTA procedure aims at efficiently predicting the probability distribution function (PDF) and the cumulative distribution function (CDF) of the delay. In other words, SSTA evaluates the statistical distributions of the delay from the statistical information of sources of variation. A computationally efficient SSTA algorithm facilitates the easy prediction of timing yield, and can be used within an optimization engine to robustly optimize the circuit in the presence of parameter variations.

Existing SSTA algorithms have many flavors: they may be path-based or block-based; they may assume Gaussian or non-Gaussian distributions; they may be parameterized in expressing all delay variables in terms of underlying parameters or not; they may incorporate spatial correlations due to physical proximity or not; and so on. In [1], the authors provide a non-parameterized method to perform SSTA in a block-based manner. This method is based on performing statistical operations of the assumed independent arrival time and random variables, by piecewise-linear modeling of CDF of variables. The authors of [2] present another non-parameterized SSTA procedure to estimate the bounds on the circuit delay PDF and CDF. In contrast, parameterized methods for SSTA provide a convenient framework for analyzing the relationship between the statistical information of the sources of variation to that of the circuit delay distributions, and are more useful in practice. A parameterized model also enables efficient computation of the statistical sensitivities of the circuit delay with respect to the varying parameters [3]–[5].

Practical parameterized SSTA algorithms are block-based in nature, i.e., they propagate the distributions of the delay from the primary inputs to the primary outputs of a circuit using a PERT-like (Program Evaluation and Review Technique) [6] traversal of the circuit graph. One of the exceptions is a path-based SSTA method proposed in [7]. In this work, the authors provide a simple procedure to perform statistical timing analysis using a path-based scheme, as a post-processing step, after identifying a sufficiently large number of critical paths by a deterministic STA. The parameterized block-based SSTA algorithms [8]–[12] provide efficient methods for performing statistical timing analysis, under the assumption of normality of parameter distributions. In [8], a novel SSTA procedure is proposed by approximating all delay and arrival time random variables as linear functions of correlated parameters. By assuming that the random vector, comprising of the parameters of variations,

has all its components following a Gaussian distribution, a principal component analysis (PCA) transformation techniques is employed to generate another random vector comprising of components which are statistically independent Gaussian random variables. A similar work [9] assumes Gaussian modeling of parameters and linear delay representation to perform efficient SSTA. Both these works, [8] and [9], use Clark's closed-form formulae [13] to approximate the maximum of two Gaussian random variables as another Gaussian random variable. The authors of [10] also propose a linear Gaussian SSTA procedure by simplifying the computations involving a set of correlated normal variables, using the PCA method. The algorithms presented in [11] and [12] provide techniques for performing SSTA using quadratic delay models of Gaussian parameters.

For all of the abovementioned Gaussian SSTA algorithms, the assumption of normality of process variations lends itself rather well for generating closed-form expressions for the delay and arrival time PDFs. Although correlation and statistical dependence between random variables tends to increase the complexity of SSTA, recent work has presented efficient techniques for handling such correlations under Gaussian distributions, using PCA to perform a simple variable transformation. This transformation enables efficient SSTA, representing delays and arrival times as functions of a new set of orthogonal, statistically independent Gaussian random variables.

However, the normality assumption is not always valid [14], and it is well known that some process parameters deviate significantly from a Gaussian distribution. For example, via resistances exhibit an asymmetric probability distribution [15], and the dopant concentration density is also observed to be well modeled by a Poisson distribution: a normality assumption may lead to significant sources of errors in SSTA. Some recent works [15], [16] propose SSTA methods that do away with the assumptions of normality for the parameter distributions, but to the best of our knowledge, no prior approach is scalable to handle large number of non-Gaussian parameters, or has presented an efficient SSTA solution under correlated non-Gaussian parameter distributions. In [15], the solution to tackle uncorrelated non-Gaussian parameters employs a numerical integration technique. However, the method of numerical integration in higher dimensions has an exponential computational complexity with respect to the number of non-Gaussian parameters. Thus, the method can efficiently handle only a few non-Gaussian sources of variation, and the runtime does not scale well with the number of such sources. The SSTA framework of [16] is general enough to consider both Gaussian and non-Gaussian parameters of variations, as long as the non-Gaussian parameters are uncorrelated. However, the technique relies on a regression strategy that requires a Monte Carlo simulation in the inner loop of the SSTA procedure. Such a technique is unlikely to scale well for large circuits with numerous sources of variations.

From the discussion of the existing SSTA methods in this section, the procedures can be broadly classified into the following four categories:

- 1) *Linear, Gaussian SSTA*: These methods employ a linear delay representation and assume normality of parameter distributions. Some examples of the techniques that offer an efficient and an accurate solution within this class of SSTA algorithms are [7]–[10].
- 2) *Nonlinear, Gaussian SSTA*: These SSTA algorithms use a nonlinear delay model, in particular, a quadratic representation of all gate delay and arrival time variables, but still assume that all parameters as Gaussians. The works of [11] and [12] fall into this category of SSTA methods.

- 3) *Linear, non-Gaussian SSTA*: This class of SSTA procedures consists of techniques that do away with the Gaussian assumption for all parameters, but still employ a first order delay model. Our SSTA method, presented in this paper, is the only efficient and scalable known work for this class of algorithms.
- 4) *Nonlinear, non-Gaussian SSTA*: These SSTA methods are a superset of the other three classes, and cover the most general case for performing statistical timing analysis. Such SSTA procedures not only use a general nonlinear delay model, they also allow the parameters to be non-normally distributed. The methods of [15] and [16] are two such examples of these general SSTA algorithms. However, as mentioned before, these works rely on computationally expensive techniques, and are not scalable to a large number of variables. In fact, even the application of these methods to a simpler case of linear representation (the subset of class 3 SSTA methods, as described above) is just as inefficient. Thus, the quest for an efficient SSTA technique for a nonlinear delay form that includes non-Gaussian parameters of distribution, remains an unsolved research problem.

In this paper, we propose an efficient algorithm to perform a linear, non-Gaussian SSTA. To the best of our knowledge, this is the only work that can handle a large number of Gaussian and non-Gaussian process parameters with correlations. The correlations are described using a grid structure, similar to that used in [8], but also incorporates non-Gaussian distributions. For a circuit with $|G|$ gates and a layout with g spatial correlation grids, the complexity of our approach is $O(g|G|)$, similar to the Gaussian case in [8]. An early version of our work appeared in [17].

II. OUTLINE OF THE SSTA PROCEDURE

The main steps in our SSTA algorithm are:

- 1) **Preprocessing to obtain an independent set of basis variables**: We employ a technique known as independent component analysis (ICA) [18]–[21] as a *preprocessing step*, with the goal of transforming the random vector of correlated non-Gaussian components to a random vector whose components are statistically independent. We then compute moments of the independent components from the moments of the non-Gaussian parameters. We orthogonalize the Gaussian parameters separately, performing PCA as in [8]. Together, we refer to this set of independent variables as the *basis set*.
- 2) **Moment matching-based PDF evaluation**: Next, we represent the gate delays as a linear canonical function of the basis set. From the moments of the basis set, we compute the moments of the gate delay variables. Finally, we translate the moments into an approximating PDF for the delay variables, using a Padé approximation-based moment matching scheme, as proposed in [22].
- 3) **Correlation-preserving statistical operations**: We process the circuit in a block-based manner, in topological order, computing the statistical sum and max operations at every step to compute the extracted PDFs of the arrival time variables. These variables are stored in terms of the linear canonical form through a moment-matching procedure.

During our exposition of our procedure, it will become amply clear that the approach borrows some techniques from several existing algorithms from the literature. However, it is important to note that the overall algorithm is distinctly different from any existing method.

The rest of this paper is organized as follows. Section III describes the method of generating moments of the varying process parameters from the process data. These moments are used as inputs to our SSTA procedure. Section IV explains the effect on non-Gaussian parameters on the SSTA results. The linear delay model employed, to represent the gate delay and arrival time expressions, is presented in Section V. Independent component analysis and its applicability to our SSTA algorithm is described in Section VI. Section VII explains the method to derive the moments of the independent components. The moment matching-based PDF evaluation scheme is delineated in Section VIII. The computations of statistical “sum” and “max” operations are presented in Section IX. Section X contains the time complexity analysis of the proposed SSTA algorithm. Experimental results are presented in Section XI, and Section XIII concludes this paper.

III. GENERATING MOMENTS FROM PROCESS DATA

Interval i ($[lb, ub)$ nm)	$Pr(\hat{L}_e = \frac{L_e - \mu_{L_e}}{\sigma_{L_e}}) \in i$
[-4.0,-3.0)	0.000
[-3.0,-2.6)	0.000
[-2.6,-2.3)	0.006
[-2.3,-1.9)	0.022
[-1.9,-1.5)	0.072
[-1.5,-1.2)	0.092
[-1.2,-0.8)	0.128
[-0.8,-0.4)	0.106
[-0.4,-0.1)	0.120
[-0.1,0.3)	0.108
[0.3,0.7)	0.122
[0.7,1.1)	0.110
[1.1,1.4)	0.070
[1.4,1.8)	0.036
[1.8,2.2)	0.002
[2.2,2.5)	0.002
[2.5,2.9)	0.004
[2.9,3.3)	0.000
[3.3,3.6)	0.000
[3.6,4.0)	0.000

TABLE I

A CUMULATIVE FREQUENCY TABLE FOR 500 RANDOMLY GENERATED VALUES OF L_e WITH $\mu_{L_e} = 65$ nm AND $\sigma_{L_e} = 5.2$ nm.

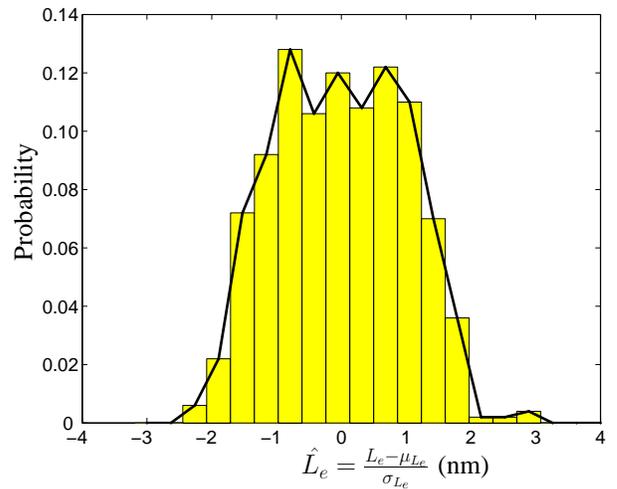


Fig. 1. A frequency histogram of the \hat{L}_e values listed in Table I.

It is important to note that our algorithm requires minimal input information: rather than relying on closed-form distribution of variational parameters, the knowledge of their moments is sufficient for our scheme to generate the circuit delay distribution. This is a desirable property for an SSTA method, as it is typically difficult to extract precise distributions from process data, and it is more realistic to obtain the moments of the parameter variations from a process engineer. For instance, given the measurements of a particular parameter X across N chips¹, k^{th} moment of X , denoted by $m_k(x)$, where x represents a sample point, can be easily computed as $m_k(x) = \sum_x x^k Pr(X = x)$. The probability $Pr(X = x)$ can be calculated by binning² all the measured values of X in some small discrete intervals $[lb, ub)$, and then dividing the frequency of values in each bin by

¹For simplicity, we ignore the intra-die variation for parameter X in this discussion.

²Binning sample points in intervals simplifies the computation by reducing the dimensionality of total number of sample points. Alternatively, it is also possible to use the raw process data to compute the moments by assigning a discrete probability, $Pr(X = x)$, to each sample point.

the total number of samples N . This process is much easier than trying to fit an accurate closed-form PDF expression for the measured values of parameter X across all N sample points, given by the value of X in each of the N chips.

k	$m_k(\hat{L}_e)$
1	0.0000
2	1.0000
3	0.0384
4	2.2733
5	0.6174
6	7.9839
7	6.2385
8	39.3220
9	56.2410
10	245.0898
11	485.8118
12	1.7515×10^3
13	4.1178×10^3
14	1.3447×10^4
15	3.4592×10^4
16	1.0711×10^5
17	2.8938×10^5
18	8.7014×10^5
19	2.4167×10^6
20	7.1479×10^6

TABLE II

A TABLE SHOWING THE FIRST TWENTY MOMENTS OF \hat{L}_e VALUES LISTED IN TABLE I.

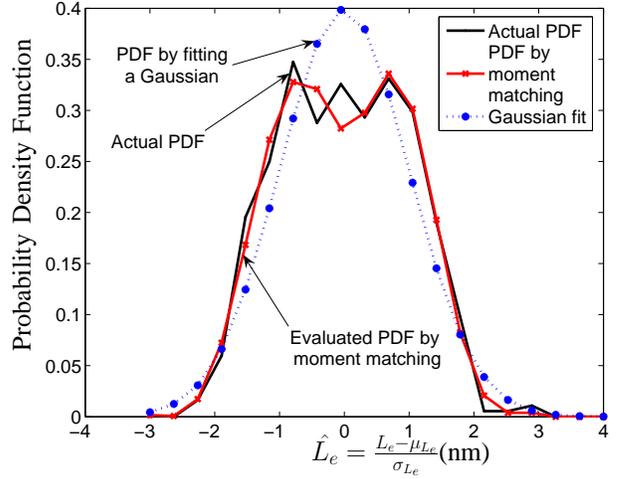


Fig. 2. PDF of \hat{L}_e values listed in Table I.

To understand the moment generation process, consider values of the effective channel length (L_e) as shown in Table I. The table contains 500 randomly generated values of L_e with a mean of 65 nm , and a standard deviation of 5.2 nm . These L_e values can be thought of as measurements across $N = 500$ chips, similar to the ones expected to be extracted from the real wafer data. Table I is the cumulative frequency table for the zero-mean, unit-variance variable \hat{L}_e values, derived by subtracting from L_e values, the sample mean (μ_{L_e}), and scaling the result by the reciprocal of the sample standard deviation (σ_{L_e}). The probabilities of occurrence of the random variable \hat{L}_e in each discrete interval or bin in the range $[-4, 4]$, shown in column one of Table I, is computed by simply dividing the frequency of \hat{L}_e in the particular bin by the total number of measured points, in this case $N = 500$. Figure 1, depicts the frequency histogram of the \hat{L}_e values listed in Table I. The solid dark line in Figure 1, corresponds to the PDF³ of \hat{L}_e . As seen in the figure, it is extremely difficult to fit a closed-form expression that would closely match this PDF.

However, the moments of the \hat{L}_e values can be easily computed by using the relation, $m_k(\hat{l}_e) = \sum_{\hat{l}_e} \hat{l}_e^k Pr(\hat{L}_e = \hat{l}_e)$, where the values of $Pr(\hat{L}_e = \hat{l}_e)$ are shown in the second column of Table I. The first twenty such moments are listed in Table II. The only inputs required by our SSTA procedure are these moments of the varying parameters. As will be explained in Section VIII, using the moments as input, the moment matching-based PDF evaluation method can generate closed-form

³It is trivial to derive the PDF of L_e from the PDF of \hat{L}_e , as will be discussed in Section VIII.

PDF expressions. Figure 2 shows the actual PDF of \hat{L}_e , the PDF corresponding to fitting a Gaussian distribution to the data of Table I, and the PDF obtained by using the moment matching-based PDF evaluation scheme. As seen from the figure, using the moments information, it is possible to derive the PDF of \hat{L}_e that matches closely with the actual PDF.

IV. NON-GAUSSIANITY IN SSTA

The circuit delay distribution depends on a number of parameters such as the effective channel length, transistor width, metal thickness, interlayer dielectric thickness, dopant density, and the oxide thickness. As pointed out in Section I, not all parameters of variations can be accurately modeled by a normally distributed random variable. Moreover, these non-Gaussian parameters may be correlated to each other due to the effect of spatial proximity. As a result, the approximation of parameters as normal distributions, followed by performing a Gaussian SSTA, may lead to significant inaccuracies in the PDF and CDF of the circuit delay.

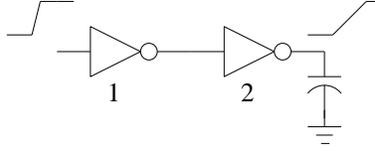


Fig. 3. A simple circuit example to illustrate the effect of non-Gaussian parameters on the PDF of the circuit delay.

To illustrate the effect of such non-Gaussian parameters on the delay distribution, we use a toy circuit, shown in Figure 3. We assume W_i and L_{e_i} for each inverter i to be the random parameters of variation. Using a first order Taylor series approximation, the delay of this circuit can be written as:

$$D = \mu + a_1.W_1 + a_2.W_2 + b_1.L_{e_1} + b_2.L_{e_2} \quad (1)$$

where a_1, a_2, b_1 , and b_2 are the sensitivities of the delay with respect to the zero-mean randomly varying parameters W_1, W_2, L_{e_1} , and L_{e_2} , respectively, and μ is the nominal delay of the circuit. Next, we perform a simple Monte Carlo simulation to evaluate the PDF of the circuit by considering the following four scenarios:

Case 1: $\{W_1, W_2\}$ are modeled as uniformly distributed random variables in $[-\sqrt{3}\sigma_W, \sqrt{3}\sigma_W]$, and $\{L_{e_1}, L_{e_2}\}$ are assumed to be Gaussian random variables with a normal distribution $N(0, \sigma_{L_e})$. Furthermore, all parameters are assumed to be statistically independent with respect to each other. Figure 4(a) illustrates the PDF of the circuit delay for this case.

Case 2: Employing the same model for the distributions of W and L_e parameters as above (Case 1), but assuming that W_1 is perfectly correlated with W_2 , and L_{e_1} is perfectly correlated with L_{e_2} . The circuit delay PDF for this case is shown in Figure 4(b).

Case 3: $\{L_{e_1}, L_{e_2}\}$ are modeled as uniformly distributed random variables in $[-\sqrt{3}\sigma_{L_e}, \sqrt{3}\sigma_{L_e}]$, and $\{W_1, W_2\}$ are assumed to be Gaussian random variables with a normal distribution $N(0, \sigma_W)$. Furthermore, all parameters are assumed to be statistically independent with respect to each other. Figure 5(a) shows the PDF of the circuit delay for this case.

Case 4: Employing the same model for the distributions of W and L_e parameters as above (Case 3), but assuming that W_1

is perfectly correlated with W_2 , and L_{e1} is perfectly correlated with L_{e2} . The circuit delay PDF for this case is illustrated in Figure 5(b).

The dashed curve in Figures 4 and 5, show the actual PDF of the circuit delay obtained by performing a Monte Carlo

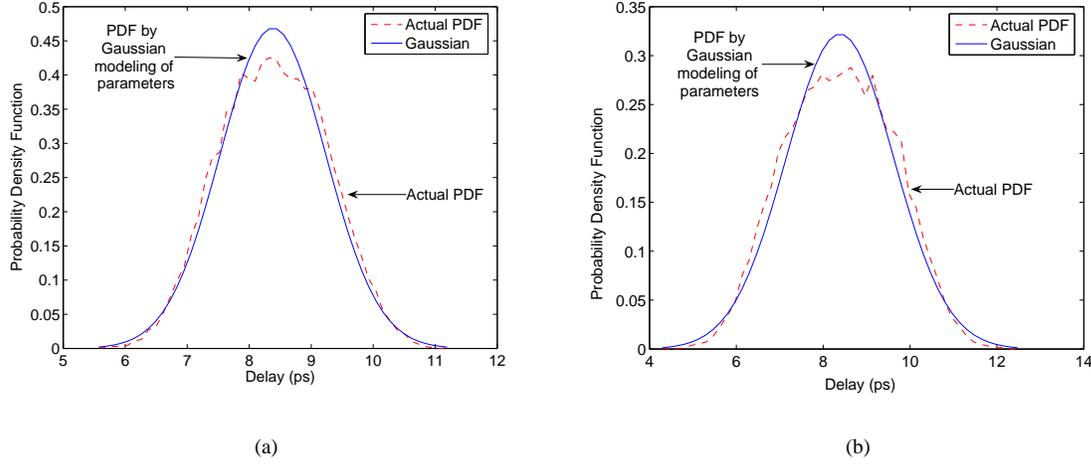


Fig. 4. PDF of the delay of the example circuit of Figure 3, when $\{W_1, W_2\}$ are modeled as uniformly distributed, and $\{L_{e1}, L_{e2}\}$ are modeled as normally distributed random variables for (a) uncorrelated and (b) correlated W and L_e process variables.

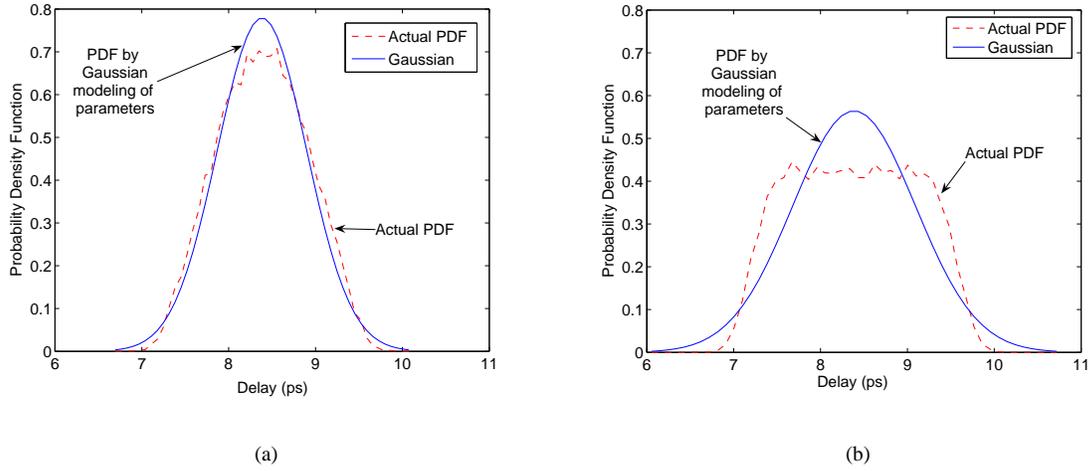


Fig. 5. PDF of the delay of the example circuit of Figure 3, when $\{L_{e1}, L_{e2}\}$ are modeled as uniformly distributed, and $\{W_1, W_2\}$ are modeled as normally distributed random variables for (a) uncorrelated and (b) correlated W and L_e process variables.

simulation, and correctly modeling W (for Cases 1 and 2) and L_e (for Cases 3 and 4) parameters, as uniformly distributed random variables, while the solid curve is the PDF obtained if the non-Gaussian variables were also modeled as Gaussian variables with the same mean and standard deviation as the uniformly distributed variables. Figures 4(a) and 5(a) show the PDFs for the cases where all of the parameters are considered to be statistically independent with respect to each other, while Figures 4(b) and 5(b) show the PDFs when W_1 is considered to perfectly correlated with W_2 , and L_{e1} is assumed to be perfectly correlated with L_{e2} . In each case, it is seen that the circuit delay PDF deviates from a Gaussian distribution due to

the presence of the non-Gaussian random variables. However, the deviation from a normal distribution is most significant in Figure 5(b). The following two reasons explain this significant non-Gaussian behavior of the circuit delay PDF:

- 1) The delay model used for the circuit of Figure 5 in these experiments, given by Equation (1), contains terms b_1 and b_2 , corresponding to the sensitivities of L_{e_1} and L_{e_2} , that outweigh the terms a_1 and a_2 , corresponding to the sensitivities of W_1 and W_2 . In particular, $|b_1| = 5.2|a_1|$, and $|b_2| = 9.8|a_2|$. Therefore, for the experiments for Cases 1 and 2, corresponding to the PDF curves of Figures 4(a) and 4(b), the effect of the Gaussian parameters $\{L_{e_1}, L_{e_2}\}$ dominates the effect of the non-Gaussian parameters $\{W_1, W_2\}$, and the circuit delay PDF does not significantly aberrate from a normal distribution.

For the experiment for Case 4, corresponding to the PDF curve in Figure 5(b), $\{L_{e_1}, L_{e_2}\}$ are modeled as uniformly distributed variables, therefore in this case, the non-Gaussian parameters dominate the normally distributed $\{W_1, W_2\}$ parameters, and the circuit delay PDF shows significant divergence from a Gaussian one.

- 2) For both Cases 3 and 4, $\{L_{e_1}, L_{e_2}\}$ are modeled as non-Gaussian variables. However the Monte Carlo PDF for Case 3, shown in Figure 5(a), assumes statistical independence of parameters. This PDF has a much closer match to a Gaussian distribution, compared to the one shown in Figure 5(b), that assumes perfect correlation between $W_1 [L_{e_1}]$ and $W_2 [L_{e_2}]$ parameters. The intuition for the significant change from a normal PDF, for the correlated case, can be arrived at by appealing to the Central Limit Theorem, according to which the addition of independent variables makes them “more Gaussian,” but this is not necessarily true for correlated random variables.

For real circuits, where many parameters are correlated due to the presence of the inherent spatial and structural correlations, the presence of non-Gaussian parameters, the sensitivities of which could potentially outweigh the Gaussian ones, implies that the circuit delay may deviate significantly from a normal distribution.

V. DELAY REPRESENTATION

To incorporate the effects of both Gaussian and non-Gaussian parameters of distribution in our SSTA framework, we represent all delay and arrival times in a linear form as:

$$D = \mu + \sum_{i=1}^n b_i \cdot x_i + \sum_{j=1}^m c_j \cdot y_j + e \cdot z = \mu + \mathbf{B}^T \mathbf{X} + \mathbf{C}^T \mathbf{Y} + e \cdot z \quad (2)$$

where D is the random variable corresponding to a gate delay or an arrival time at the input port of a gate, x_i is a non-Gaussian random variable corresponding to a physical parameter variation, b_i is the first order sensitivity of the delay with respect to the i^{th} non-Gaussian parameter, y_j is a parameter variation modeled as a Gaussian random variable, c_j is the linear sensitivity with respect to the j^{th} Gaussian parameter, z is the uncorrelated parameter which may be a Gaussian or a non-Gaussian random variable, e is the sensitivity with respect the uncorrelated variable, n is the number of correlated non-Gaussian variables, and m is the number of correlated Gaussian variables. In the vector form, \mathbf{B} and \mathbf{C} are the sensitivity vectors for \mathbf{X} , the random vector of non-Gaussian parameter variations, and \mathbf{Y} , the random vector of Gaussian random variables, respectively. Note that we assume statistical independence between the Gaussian and non-Gaussian parameters: this is a reasonable assumption as

parameters with dissimilar distributions are likely to represent different types of variables, and are unlikely to be correlated.

The value of the mean delay μ is adjusted so that the random vectors \mathbf{X} and \mathbf{Y} are centered, i.e., each component x_i and y_i is a zero-mean random variable. The uncorrelated random variable z is also centered. Note that in the representation of Equation (2), the random variables x_i are correlated with each other and may be of any underlying non-Gaussian distribution. Unlike the delay models of [8], [9], we do not constraint the parameter distributions to be Gaussian. The canonical model of equation (2) is similar to the model of [15] without the nonlinear terms. The slight difference is that the uncorrelated parameter z is not constrained to be a Gaussian variable.

VI. INDEPENDENT COMPONENT ANALYSIS

For reasons of computational and conceptual simplicity, it is useful to work with a set of statistically independent random variables in the SSTA framework. If the components of random vector \mathbf{X} were correlated Gaussian random variables with a covariance matrix Σ , a PCA transformation $\mathbf{R} = P_x \mathbf{X}$ would yield a random vector \mathbf{R} comprising of Gaussian uncorrelated random variables [8]. Since for a Gaussian distribution, uncorrelatedness implies statistical independence⁴, the components of \mathbf{R} are also statistically independent.

However, such a property does not hold for general non-Gaussian distributions. In Equation (2), the random vector \mathbf{X} consists of correlated non-Gaussian random variables, and a PCA transformation, $\mathbf{S} = P_x \mathbf{X}$, would not guarantee statistical independence for the components of the transformed vector \mathbf{S} . Since the PCA technique focuses only on second order statistics, it can only ensure uncorrelatedness, and not the much stronger requirement of statistical independence.

Independent component analysis [18]–[21] is a mathematical technique that precisely accomplishes the desired goal of transforming a set of non-Gaussian correlated random variables to a set of random variables that are statistically as independent as possible, via a linear transformation. ICA has been an active area of research in the area of signal processing, feature extraction and neural networks due to its ability to capture the essential structure of data in many applications.

A. The Cocktail Party Problem

The ICA principle can be explained by the *cocktail party problem* example illustrated in Figure 6. The set up shown in the figure, consists of n speakers, who can be regarded as independent sources, and n receivers, represented by the ears in Figure 6. The speakers or the independent sources emit independent speech signals, but their simultaneous speech results in interferences of the independent signals. As shown in Figure 6, due to the interference or mixing of the independent speech signals, the signals observed by the receivers are no longer independent. The amount of mixing of the independent speech signals may be derived from elements of a mixing matrix A , which could depend on metrics such as the distance of each speaker from the receiver. Mapping the cocktail party problem set up back to the ICA problem, the ICA set up consists of having a vector \mathbf{S} consisting of n statistically independent components, s_1, \dots, s_n , and observations of n linear mixtures, x_1, \dots, x_n , of the n

⁴Two random variables X and Y are uncorrelated if $E[XY] = E[X]E[Y]$, while they are independent if $E[f(X)g(Y)] = E[f(X)]E[g(Y)]$ for any functions f and g . For instance, if X and Y are independent, then $E[X^i Y^j] = E[X^i]E[Y^j]$. For Gaussian distributions, uncorrelatedness is identical to independence. For a general non-Gaussian distribution, independence implies uncorrelatedness, but not vice versa.

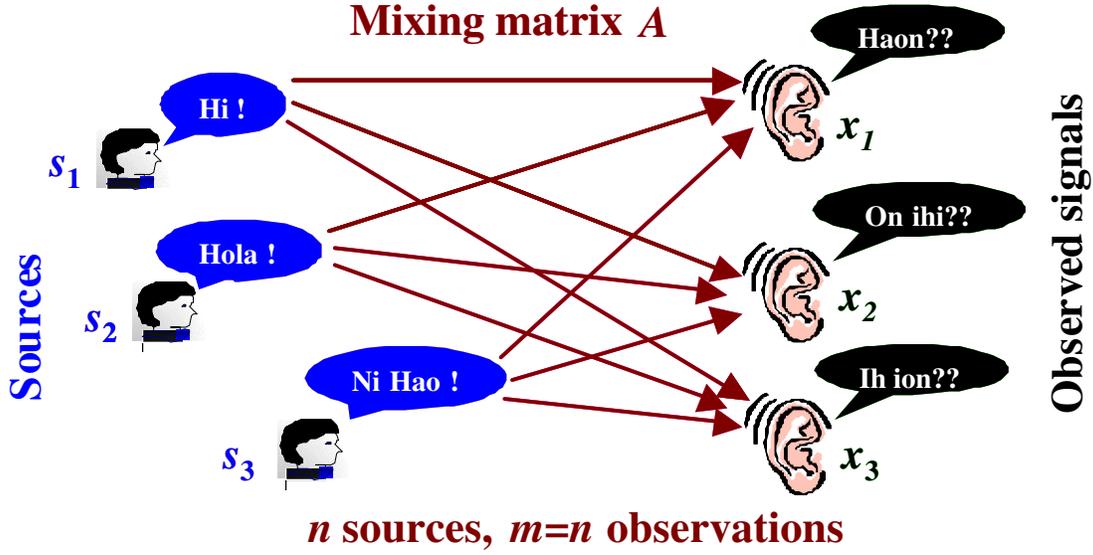


Fig. 6. The cocktail party problem to illustrate the independent component analysis set up.

independent components. The observed components can be thought of as the correlated non-Gaussian random variables \mathbf{X} in Equation (2), produced by a linear mixing of the elements of a vector \mathbf{S} of independent random variables, as follows:

$$\mathbf{X} = \mathbf{A}\mathbf{S} \quad (3)$$

where \mathbf{A} is the $n \times n$ mixing matrix.

The problem of ICA is to estimate the elements of the unknown mixing matrix \mathbf{A} , and the samples of statistically independent components s_1, \dots, s_n , as accurately as possible, given only the samples of the observed vector \mathbf{X} . Equation (3) can be alternatively written as:

$$\begin{aligned} \mathbf{S} &= \mathbf{W}\mathbf{X} \text{ where} \\ s_i &= \mathbf{W}_i^T \mathbf{X} = \sum_{j=1}^n w_{ij} x_j \quad \forall i = 1, \dots, n \end{aligned} \quad (4)$$

In the above equation, \mathbf{W} is the inverse of the unknown mixing matrix \mathbf{A} . Algorithms for ICA estimate the vectors \mathbf{W}_i that maximize the non-Gaussianity of $\mathbf{W}_i^T \mathbf{X}$ by solving a nonlinear optimization problem. Typical measures of non-Gaussianity are kurtosis, negentropy, and mutual information; for a comprehensive reference on ICA, see [18]–[21].

For our SSTA algorithm, we use ICA as a preprocessing step to transform the correlated set of non-Gaussian random variables x_1, \dots, x_n to a set of statistically independent variables s_1, \dots, s_n , by the relation $\mathbf{S} = \mathbf{W}\mathbf{X}$ of Equation (4). In practice, ICA estimates the mixing matrix \mathbf{A} and its inverse matrix \mathbf{W} , which yield the components, s_1, \dots, s_n , which are statistically as independent as possible. For the purposes of application of ICA transformation in our SSTA algorithm, we will consider the vector \mathbf{S} to consist of truly statistically independent components. Experimental results, presented in Section XI, validate this assumption.

Like principal components, the independent components of vector \mathbf{S} are mathematical abstractions that cannot be directly observed. Similar to the PCA procedure, which requires normalization of $N(\mu, \sigma)$ variables to $N(0,1)$ variables, the ICA methods also require centering and whitening of the components of vector \mathbf{X} , i.e., prescaling the variables to have zero mean and unit variance [20]. For a specific grid, the independent components of the non-Gaussian random variables must be computed just once, and this can be carried out as a precharacterization step. In other words, ICA need not be recomputed for different circuits or different placements of a circuit. *Thus, the ICA preprocessing step does not impact the runtime of the SSTA procedure.*

One of the requirements of the ICA technique is that all of the original source of independent sources, s_1, \dots, s_n , should be non-Gaussian. Therefore, in the delay model of Equation (2), we must treat the correlated non-Gaussian random variables \mathbf{X} , and the correlated Gaussian random variables \mathbf{Y} , separately. The ICA technique is applied to non-Gaussian parameters \mathbf{X} , and a PCA transformation is applied to Gaussian variables \mathbf{Y} , to obtain a set of statistically independent non-Gaussian variables \mathbf{S} , and a set of independent Gaussian variables \mathbf{R} . We then substitute the respective transformation matrices A and P_y in Equation (2) to arrive at the following *canonical delay model*:

$$\begin{aligned} D &= \mu + \mathbf{B}'^T \mathbf{S} + \mathbf{C}'^T \mathbf{R} + e.z \\ &= \mu + \sum_{i=1}^n b'_i \cdot s_i + \sum_{j=1}^m c'_j \cdot r_j + e.z \end{aligned} \quad (5)$$

where $\mathbf{B}'^T = \mathbf{B}^T A$, $[\mathbf{C}'^T = \mathbf{C}^T P_y^{-1}]$ is the new sensitivity vector with respect to the statistically independent non-Gaussian components, s_1, \dots, s_n [Gaussian principal components r_1, \dots, r_m].

B. Generating Samples of Correlated Non-Gaussian Variables

Algorithm 1 Generate Correlated Non-Gaussian Samples

```

1: /*Inputs: Correlation matrix  $Q$  ( $n \times n$ ), mean vector  $\mu_{\mathbf{X}}$  ( $n \times 1$ ), CDF of  $x_j$  parameter  $F_j(x_j), \forall j = 1, \dots, n$ */
2: /*Output: Matrix  $Corr(NUM\_SAMPLES \times n)$  as samples of correlated non-Gaussian variables*/
3: /*Step1 : Generate samples of multivariate normal distribution  $N(\mu, Q)$ */
4:  $i=1$ ;
5: while ( $i < NUM\_SAMPLES$ ) do
6:    $Z(i)=\text{mvnrnd}(\mu, Q)$ ;
7:    $i=i+1$ ;
8: end while
9: /*Step2: Map the multivariate normal samples to a multivariate uniform samples in  $[0,1]$ */
10:  $U=\text{normcdf}(Z)$ ;
11: /*Step3: Apply inverse CDF transformation to samples in each column of  $U$ */
12:  $j=1$ ;
13: while ( $j < n$ ) do
14:    $Corr(j)=\mathbf{F}_j^{-1}(U)$ ;
15:    $j=j+1$ ;
16: end while

```

The ICA method requires, as inputs, the samples of the correlated non-Gaussian parameters. If these samples are readily available from the process data, they can be directly provided to the ICA module to generate the estimates of the mixing matrix A , and the samples of the independent components, s_1, \dots, s_n . However, if instead of the samples of correlated parameters, the

closed-form PDFs of the non-Gaussian sources of variation are provided, we must first generate samples of the parameters from the given PDF expressions⁵. To model the correlation between the non-normal parameters, x_1, \dots, x_n , the chip area is first tiled into a grid, as in [8], and the correlation matrix, Q , associated with \mathbf{X} is determined. The matrix Q and the mean vector $\mu_{\mathbf{X}}$ is used to generate the samples of the correlated non-Gaussian variables by employing the method of normal copulas [23]. Algorithm 1 shows the pseudo-code of this method, which is based on performing a series of correlation preserving transforms on a set of random numbers.

The procedure consists of three main steps. In the first step, spanning lines 4–8, samples from a multivariate normal distribution, $N(\mu_{\mathbf{X}}, Q)$, are generated. As will become clear in the next steps, these set of Gaussian random numbers are used to generate the required non-normal numbers having a mean vector $\mu_{\mathbf{X}}$, and the correlation matrix Q . The function call **mvnrnd** generates these samples. In the next step, shown on line 10, the normal samples are mapped to a multivariate uniform distribution in the range [0,1]. The transformation function **normcdf** is simply the CDF of the standard normal distribution. The following relations prove that for a single standard normal random variable y , with a CDF denoted by $F_y(y)$, a transformation $u = F_y(y)$ results in a uniformly distributed variable u in the range [0,1].

$$F_u(u_0) = Pr(u \leq u_0) = Pr(F_y(y) \leq u_0) = Pr(y \leq F_y^{-1}(u_0)) = F_y(F_y^{-1}(u_0)) = u_0 \quad (6)$$

Thus, the CDF of u is $F_u(u_0) = u_0$, which is same as the CDF of a uniformly distributed random variable in the range [0,1]. In our case, Z comprises of samples of multivariate normal distribution. Thus, each component of random vector associated with Z , has a marginal distribution of a standard normal. Therefore, the function mapping $U = \mathbf{normcdf}(Z)$, maps each normally distributed component of the random vector associated with Z , into a uniformly distributed variable in the range [0,1]. The statistical dependence between the generated samples still remains after the transformation. The subroutines for generating samples of multivariate normal distribution (**mvnrnd**()), and the CDF of normal distribution (**normcdf**()) are commonly available in standard mathematical software packages, such as [24] and [25].

The last step in Algorithm 1, shown in lines 12–16, consists of transforming the multivariate uniform samples in U to the individual non-Gaussian marginal distributions. The transformation function is F_j^{-1} , which is the inverse of the CDF of the j^{th} non-Gaussian random variable. For example, if the j^{th} non-Gaussian parameter x_j is uniformly distributed in the range $[lb, ub]$, $F_j^{-1}(x) = lb + (ub - lb)x$. It is easy to prove that mapping uniformly distributed random numbers on interval [0,1], by a function which is an inverse CDF $F^{-1}(x)$ of a particular distribution, produces random numbers which have a distribution as given by CDF $F(x)$ [26]. Since samples in each column of the matrix U , are mapped by the required inverse CDF function F_j^{-1} , the correlation structure between the columns of U is preserved after the transformation. The output of the algorithm produces a matrix *Corr*, with *NUM_SAMPLES* rows and n columns. Each column of this matrix contains samples of a non-Gaussian parameter drawn from the required distribution. The columns are correlated with each other according to the original linear correlation matrix Q , and their sample mean is the same as the original mean vector $\mu_{\mathbf{X}}$.

⁵As will be explained in Section XI, we use the method of generating correlated non-Gaussian random numbers, described in this section, for our experimental set up that assumes, as inputs, well-known closed-form PDFs for parameters x_1, \dots, x_n .

Following the steps described in Algorithm 1, we generate samples of correlated non-Gaussian parameters. These samples are required as input to the ICA methods, which generate the ICA transformation matrix A in Equation (3).

VII. PREPROCESSING TO EVALUATE THE MOMENTS OF THE INDEPENDENT COMPONENTS

The inputs required for our SSTA technique correspond to the moments of parameters of variation. Consider a process parameter represented by a random variable x_i : let us denote its k^{th} moment by $m_k(x_i) = E[x_i^k]$. We consider three possible cases:

Case I: If the closed-form of the distribution of x_i is available, and it is of a standard form (e.g., Poisson or uniform), then $m_k(x_i) \forall k$ can be derived from the standard mathematical tables of these distributions.

Case II: If the distribution is not in a standard form, then $m_k(x_i) \forall k$ may be derived from the moment generating function (MGF), if a continuous closed-form PDF of the parameter is known. If the PDF of x_i is the function $f_{x_i}(x_i)$, then its moment generating function $M(t)$ is given by

$$M(t) = E[e^{tx_i}] = \int_{-\infty}^{\infty} e^{tx_i} f_{x_i}(x_i) dx_i \quad (7)$$

The k^{th} moment of x_i can then be calculated as the k^{th} order derivative of $M(t)$ with respect to t , evaluated at $t = 0$. Thus, $m_k(x_i) = \frac{d^k M(t)}{dt^k}$ at $t = 0$.

Case III: If a continuous closed-form PDF cannot be determined for a parameter, the moments can still be evaluated from the process data files as:

$$m_k(x_i) = \sum_x x^k Pr(X_i = x) \quad (8)$$

where $Pr(x_i = x)$ is the probability that the parameter x_i assumes a value x . This moment generation process is explained in Section III.

Given the underlying process variables and their moments, the next step after performing ICA is to determine the moments of the independent components, s_1, \dots, s_n , from the moments of the correlated non-Gaussian parameters x_1, \dots, x_n . The moments of the parameters, $E[x_i^k]$, are the inputs to the SSTA algorithm.

We now refer back to the ICA transformation of Equation (3), $\mathbf{X} = \mathbf{A}\mathbf{S}$ and rewrite the relationship by taking the expectation of both sides as:

$$\begin{aligned} E[x_1^k] &= E[(a_{11}s_1 + a_{12}s_2 + \dots + a_{1n}s_n)^k] \\ E[x_2^k] &= E[(a_{21}s_1 + a_{22}s_2 + \dots + a_{2n}s_n)^k] \\ &\vdots \\ E[x_n^k] &= E[(a_{n1}s_1 + a_{n2}s_2 + \dots + a_{nn}s_n)^k] \end{aligned} \quad (9)$$

where a_{ij} is an element of the mixing matrix A obtained via ICA. In the above equation, the left hand side, which is the

k^{th} moment of each component of \mathbf{X} , is known. The right hand side can be simplified by performing an efficient multinomial expansion using the idea of binomial moment evaluation presented in [22]. The moments are computed successively, starting from the first to the second to the third, and so on. For example, after all of the first moments have been computed, the second moment of each s_i can be computed by rewriting Equation (9) using $k = 2$ as

$$\begin{aligned}
E[x_1^2] &= \sum_{i=1}^n a_{1i}^2 E[s_i^2] + 2 \sum_{i=1}^n \sum_{j=i+1}^n a_{1i} a_{1j} E[s_i] E[s_j] \\
E[x_2^2] &= \sum_{i=1}^n a_{2i}^2 E[s_i^2] + 2 \sum_{i=1}^n \sum_{j=i+1}^n a_{2i} a_{2j} E[s_i] E[s_j] \\
&\vdots \\
E[x_n^2] &= \sum_{i=1}^n a_{ni}^2 E[s_i^2] + 2 \sum_{i=1}^n \sum_{j=i+1}^n a_{ni} a_{nj} E[s_i] E[s_j]
\end{aligned} \tag{10}$$

The only unknowns in the above equation are the second moments, $E[s_i^2]$, of each s_i , and these can be calculated easily.

In general, while solving for the k^{th} moment of s_i using Equation (9), all of the $(k-1)$ moments are known from previous computations. Moreover, since the components of \mathbf{S} are independent, we can perform the operation $E[s_i^a s_j^b] = E[s_i^a] E[s_j^b]$, and efficiently apply the binomial moment evaluation scheme. As indicated by Equation (10), the computation of the k^{th} moment of the independent components, s_1, \dots, s_n , requires the solution of an $n \times n$ system of linear equations. Thus, to compute $2M$ moments of the independent components, we must solve $2M$ systems of linear equations corresponding to (9) for $k = 1, \dots, 2M$. However, since this is a part of the preprocessing phase, it may be carried out off-line for a specific technology, and it does not contribute to the complexity of the SSTA algorithm.

Note that while ICA does provide the W matrix, it is not easily possible to use $\mathbf{S} = W\mathbf{X}$ to find the moments of the s_i variables. This is because the binomial moment evaluation procedure requires the random variables to be statistically independent, which is true for the s_i variables but not the x_i variables.

VIII. MOMENT MATCHING-BASED PDF EXTRACTION

To compute the PDF/CDF of the delay or arrival time random variable we adapt the probability extraction scheme, *APEX*, proposed in [22]. Given $2M$ moments of a random variable as inputs to the *APEX* algorithm, the scheme employs an asymptotic waveform evaluate (AWE) technique to match the $2M$ moments in order to generate an M^{th} order linear time invariant (LTI) system. The scheme then approximates the PDF [CDF] of a random variable by an impulse response $h(t)$ [step response $s(t)$] of the M^{th} order LTI system. The details of the *APEX* algorithm can be found in [22].

We return to the example of Figure 3 to explain moment matching-based PDF evaluation method. To compute the delay PDF for the example, we must first calculate $2M$ moments of D from Equation (1). Assuming (W_1, W_2) to be perfectly correlated identical Gaussian random variables, and (L_1, L_2) to be perfectly correlated, and uniformly distributed identical random variables (Case 4 of Section IV), we have:

$$\hat{D} = a.W + b.L_e \tag{11}$$

where $\hat{D} = D - \mu$, $a = a_1 + a_2$ and $b = b_1 + b_2$. Assuming W and L_e as statistically independent variables, the k^{th} moment of \hat{D} can be computed by using the binomial expansion formula as:

$$m_k[\hat{D}] = \sum_{i=0}^k \binom{k}{i} a^i b^{k-i} m_i(W) m_{k-i}(L_e) \quad (12)$$

where all of the k moments of W and L_e are known from the underlying normal and uniform distributions. Since the normal and uniform distributions used in this example are both well-studied, their moments can be obtained from mathematical tables. Having computed $2M$ moments of \hat{D} from Equation (12), we can now employ the AWE-based PDF evaluation scheme to approximate the PDF and CDF of \hat{D} by an impulse response as:

$$f_{\hat{D}}(\hat{d}) = \begin{cases} \sum_{i=1}^M \hat{r}_i \cdot e^{\hat{p}_i \cdot \hat{d}} & \hat{d} \geq 0 \\ 0 & \hat{d} < 0 \end{cases} \quad (13)$$

$$F_{\hat{D}}(\hat{d}) = \begin{cases} \sum_{i=1}^M \frac{\hat{r}_i}{\hat{p}_i} (e^{\hat{p}_i \cdot \hat{d}} - 1) & \hat{d} \geq 0 \\ 0 & \hat{d} < 0 \end{cases} \quad (14)$$

where \hat{r} [\hat{p}] are the residues [poles] of the LTI approximation.

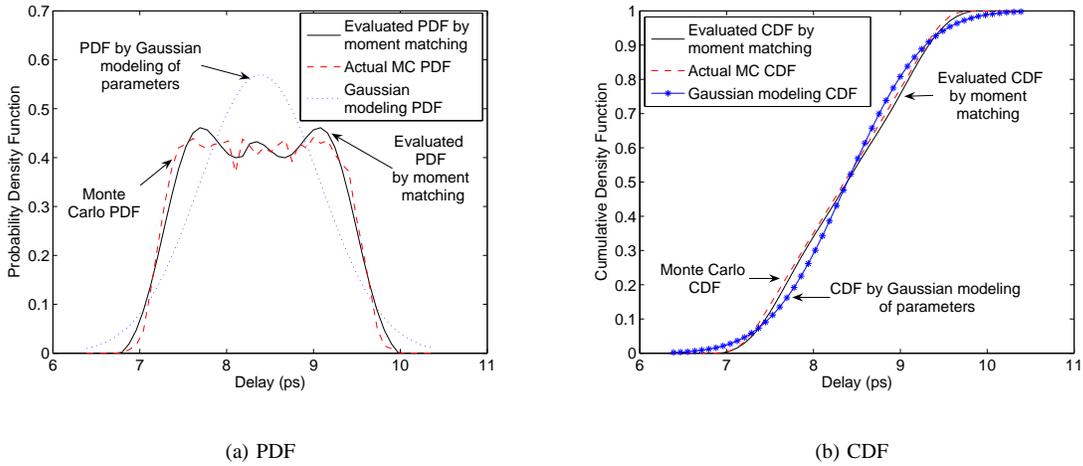


Fig. 7. Extracted PDF and CDF for the delay of the example circuit.

Figure 7 shows the evaluated delay PDF ($f_D(d) = f_{\hat{D}}(d + \mu)$) and CDF ($F_D(d) = F_{\hat{D}}(d + \mu)$) of the circuit of Figure 3 using $M = 10$ moments. The evaluated PDF matches closely with the Monte Carlo simulation; the match for the CDF is even better.

We can generalize the PDF evaluation idea, illustrated in the above example, to compute the PDF (CDF) of any random delay variable expressed in the canonical form of Equation (5). For such a delay variable with $l = m + n + 2$ terms, the binomial moment evaluation procedure can be employed to calculate the $2M$ moments, as long as all l variables in the delay expression are statistically independent. The canonical form expression of Equation (5) satisfies this independence requirement by construction.

We have enhanced the PDF evaluation algorithm in [22] for better numerical accuracy and stability. Instead of evaluating the PDF of a random variable D directly, we first prescale it by defining a new random variable $\hat{D} = \frac{D - \mu_D}{\sigma_D}$, and evaluate the PDF of \hat{D} . Without the prescaling step, the higher order moments of D can become extremely large (or extremely small) and affect the numerical accuracy of the moment computation. We compute the flipped PDF of $(-\hat{D})$, and reconstruct the final PDF from the flipped and the original PDF to avoid numerical errors due to the final value theorem, as in [22]. The PDF and CDF of D is retrieved from the PDF of \hat{D} by using the relationship:

$$\begin{aligned} f_D(d) &= \frac{1}{\sigma_D} f_{\hat{D}}\left(\frac{d - \mu_D}{\sigma_D}\right) \\ F_D(d) &= F_{\hat{D}}\left(\frac{d - \mu_D}{\sigma_D}\right) \end{aligned} \quad (15)$$

In general, given the moments of the independent components, precharacterized as in Section VII, we can compute the moments of the delay and arrival time random variables from Equation (5). The moments of an $N(0, 1)$ Gaussian distribution corresponding to each principal component, r_1, \dots, r_m , are well known as:

$$m_k(r_i) = \begin{cases} 1 & k = 0 \\ 0 & k = 1, 3, 5, \dots \\ 1 \cdot 3 \cdot 5 \dots (k - 1) & k = 2, 4, 6, \dots \end{cases} \quad (16)$$

The moments of the uncorrelated process parameter z can be easily computed using the techniques in Section VII. As we will see in Section IX, during the SSTA propagation, the role of z in the canonical form is to serve as a place holder for the moments of the uncorrelated part, and these moments will be propagated further. For each gate, given the moments of all random variables $s_1, \dots, s_n, r_1, \dots, r_m$, and z , which are all statistically independent with respect to each other, we may use the binomial evaluation method to compute the $2M$ moments of the gate delay; a similar procedure will be used to compute the arrival times in the canonical form in Section IX.

IX. SSTA PROCEDURE

From the theory explained in the previous sections, we now have the ability to evaluate the PDF and the CDF of the delay and the arrival time random variables, expressed in the linear canonical form, as a function of Gaussian and non-Gaussian parameters of variation. In this section, we describe our SSTA framework. It is well known that the arrival time propagation procedure, operating in topological order on the circuit graph, involves the atomic operations of “sum” and “max.” We will show how these atomic operations can be performed to produce a result that can be represented in the canonical form of Equation (5).

A. The “sum” Operation

The sum operation to add two arrival time or delay random variables, expressed in the linear canonical form of Equation (5), is mostly straightforward. Consider two random variables, D_1 and D_2 expressed as:

$$\begin{aligned} D_1 &= \mu_1 + \sum_{i=1}^n b'_{i_1} \cdot s_i + \sum_{j=1}^m c'_{j_1} \cdot r_j + e_1 \cdot z_1 \\ D_2 &= \mu_2 + \sum_{i=1}^n b'_{i_2} \cdot s_i + \sum_{j=1}^m c'_{j_2} \cdot r_j + e_2 \cdot z_2 \end{aligned} \quad (17)$$

The sum $D_3 = D_1 + D_2$ can be expressed in the same canonical form as:

$$D_3 = \mu_3 + \sum_{i=1}^n b'_{i_3} \cdot s_i + \sum_{j=1}^m c'_{j_3} \cdot r_j + e_3 \cdot z_3 \quad (18)$$

where $\mu_3 = \mu_1 + \mu_2$, $b'_{i_3} = b'_{i_1} + b'_{i_2}$, and $c'_{j_3} = c'_{j_1} + c'_{j_2}$.

The one difference here, as compared to the Gaussian case (e.g., in [8]), relates to the computation of the uncorrelated non-Gaussian parameter, $e_3 \cdot z_3$. The random variable $e_3 \cdot z_3 = e_1 \cdot z_1 + e_2 \cdot z_2$, serves as a place holder to store the moments of $(e_1 \cdot z_1 + e_2 \cdot z_2)$. In other words, rather than propagating an uncorrelated component z in the canonical form, we propagate its $2M$ moments.

B. The “max” Operation

The PDF of the maximum of the two *independent* random variables U and V , given by $T = \max(U, V)$, can be simply computed as:

$$f_T(t) = F_U(t)f_V(t) + F_V(t)f_U(t) \quad (19)$$

where f represents the PDF of each random variable, and F its CDF. If U, V are not only independent, but can also be expressed in the canonical form of Equation (5), then the PDF and CDF of T can be easily computed using the PDF evaluation technique described in Section IX, in a closed-form using Equation (19).

However, in general, two arrival time random variables A_1 and A_2 , expressed in the canonical form of Equation (5), *do not* satisfy the independence requirement above, as they may both have nonzero coefficients associated with an s_i and/or an r_i variable. Fortunately, it is possible to work around this by using a simple technique that permits the application of Equation (19) to compute the PDF of random variable $A_{max} = \max(A_1, A_2)$. Let us begin with the canonical expressions for A_1 and A_2 :

$$\begin{aligned} A_1 &= \mu_1 + \sum_{i=1}^n b'_{i_1} \cdot s_i + \sum_{j=1}^m c'_{j_1} \cdot r_j + e_1 \cdot z_1 \\ A_2 &= \mu_2 + \sum_{i=1}^n b'_{i_2} \cdot s_i + \sum_{j=1}^m c'_{j_2} \cdot r_j + e_2 \cdot z_2 \end{aligned} \quad (20)$$

The operation $A_{max} = \max(A_1, A_2)$ can be now simplified as:

$$A_{max} = W + \max(U, V) \quad (21)$$

where

$$\begin{aligned} W &= b'_{1_2} \cdot s_1 + c'_{1_2} \cdot r_1 + \sum_{i=2}^n b'_{i_1} \cdot s_i + \sum_{j=2}^m c'_{j_1} \cdot r_j \\ U &= \mu_1 + (b'_{1_1} - b'_{1_2}) \cdot s_1 + (c'_{1_1} - c'_{1_2}) \cdot r_1 + e_1 \cdot z_1 \\ V &= \mu_2 + \sum_{i=2}^n (b'_{i_2} - b'_{i_1}) \cdot s_i + \sum_{j=2}^m (c'_{j_2} - c'_{j_1}) \cdot r_j + e_2 \cdot z_2 \end{aligned} \quad (22)$$

The above representation of the max operation ensures that the random variables U and V involved in the max operation, $\max(U, V)$, are statistically independent as they do not share any variables⁶.

Therefore, from Equations (19) and (21), we can write $A_{max} = W + T$. Clearly, from Equation (22), W is available in the canonical form, and our next task is to express T in the form of Equation (5) as well, since this would permit us to write A_{max} in the canonical form.

To achieve this, we employ the idea of tightness probability [9], to express $T = \max(U, V)$ as:

$$T = \mu_T + \sum_{i=1}^n b'_{i_T} \cdot s_i + \sum_{j=1}^m c'_{j_T} \cdot r_j + e_T z_T \quad (23)$$

Our discussions in the previous sections provide us with all of the machinery required to efficiently compute the tightness probability, $p_{U>V} = Pr(U > V)$. We define a random variable $\hat{Q} = V - U$, and use the sum operation defined in Section IX-A to express the random variable \hat{Q} in the canonical form. Next, employing the technique described in Section VIII, we compute the $2M$ moments of random variable \hat{Q} , and evaluate the CDF, $F_{\hat{Q}}(\hat{q})$, as a step response of the approximated LTI system using the following relationship:

$$\begin{aligned} F_{\hat{Q}}(\hat{q}) &= \sum_{i=1}^M \frac{\hat{r}_i}{\hat{p}_i} (e^{\hat{p}_i \cdot \hat{q}} - 1) \quad (\hat{q} \geq 0) \\ &= 0 \quad (\hat{q} < 0) \end{aligned} \quad (24)$$

where \hat{r} and \hat{p} are the residues and poles of the approximated M^{th} order LTI system. The tightness probability $p_{U>V}$ is simply given by the CDF of \hat{Q} evaluated at $\hat{q} = 0$, since $Pr(U > V) = Pr(\hat{Q} \leq 0) = F_{\hat{Q}}(0)$.

Unlike [15], this method does not require the computationally expensive technique of numerical integration in high dimensions for non-Gaussian parameters. The ability to compute the tightness probability $p_{U>V}$ analytically, from the evaluated CDF of $(\hat{Q} = V - U)$, makes the SSTA procedure very efficient and allows us to process a large number non-Gaussian variables.

Having computed the tightness probability, $p_{U>V}$, the sensitivities b'_{i_T} , c'_{i_T} , and z_T of $T = \max(U, V)$ in Equation (23)

⁶Note that this is a sufficient condition for independence since all variables in the expressions of U and V , obtained from the ICA and the PCA transforms are statistically independent.

can be written in terms of the sensitivities of U and V . Specifically:

$$\begin{aligned} b'_{i_T} &= p_{U>V}.b'_{i_U} + (1 - p_{U>V}).b'_{i_V} \quad \forall i = 1, \dots, n \\ c'_{j_T} &= p_{U>V}.c'_{j_V} + (1 - p_{U>V}).c'_{j_U} \quad \forall j = 1, \dots, m \end{aligned} \quad (25)$$

Recall that the uncorrelated parameter term in Equation (23) is a place holder for the moments of the uncorrelated parameter: the moments of z_T can also be computed using the tightness probability: z_T assigned the moments of the random variable $(p_{U>V}.e_U.z_U + (1 - p_{U>V}).e_V.z_V)$. The adjustment of the sensitivity term e_T will be explained later in this section.

The use of tightness probabilities is only a heuristic and suffers from problems of accuracy. Therefore, to reduce the error in the heuristic, we compute the mean μ_T in Equation (23) and the variance of T , σ_T^2 , exactly from the PDF of T . In order to achieve this, we use Equation (19): note that this is applicable since U and V are independent by construction. Using the closed-form PDF, $f_T(t)$, we can compute μ_T from the first principles as $\mu_T = E[\max(U, V)] = \int_{-\infty}^{\infty} t f_T(t) dt$.

The last term left to compute is e_T , the coefficient term of the uncorrelated random variable z_T . We compute this term so that we match the variance of the closed-form PDF of T , $f_T(t)$, alluded to above, with the variance of canonical representation of Equation (23). The variance can be computed from $f_T(t)$ as:

$$\sigma_T^2 = \int_{-\infty}^{\infty} t^2 f_T(t) dt - (E[\max(U, V)])^2 \quad (26)$$

Having matched the variance term in Equation (26) to the variance in the expression Equation (23), all of the terms required to represent $T = \max(U, V)$ back to the canonical form are known. As a final step, referring back to Equation (21), we perform the sum operation between W and $T = \max(U, V)$ to complete the computation of $A_{max} = \max(A_1, A_2)$.

X. TIME COMPLEXITY ANALYSIS

The steps to generate the ICA mixing matrix A , the PCA transform, and the moments of the independent components s_i, \dots, s_n do not affect the online runtime of the procedure. These preprocessing steps have a one time precharacterization cost. Hence, the computational cost of the main steps in the SSTA procedure is comprised of the circuit graph traversal, and the sum and max operations.

The sum operation has a time complexity of $O(n + m)$, where n is the number of non-Gaussian independent components and m is the number of Gaussian principal components.

The main steps in the max operation consists of computing moments of the delay variables, PDF evaluation by the AWE-based method, and calculating the mean and the variance terms to express the result of max operation back to a canonical form. The cost of computing $2M$ moments using the binomial moment evaluation procedure is $O(M(n + m))$. The PDF evaluation involves the solution of a linear $M \times M$ system of linear equations, described by a Hankel matrix, is $O(M^3)$; in practice, M is upper-bounded by a small constant, and excellent solution are obtained for $M \leq 10$. The mean and the variance terms are computed by one dimensional numerical integration and can be calculated in constant time. Thus, the complexity of the max operation is $O(m + n)$. For a layout with g spatial correlation grids, $m + n = O(g)$. Therefore, both the sum and the max

operation have a complexity of $O(g)$.

In the PERT-like traversal of the circuit graph, for each gate we must change the delay representation of Equation (2) to that of Equation (5). In particular, we require the new sensitivity vectors $\mathbf{B}'^T = \mathbf{B}^T A$, $[\mathbf{C}'^T = \mathbf{C}^T P_y^{-1}]$. The dimensions of the ICA transformation matrix A is $n \times n$, and the PCA transformation matrix P_y is $m \times m$. However, the original sensitivity vectors \mathbf{B}^T and \mathbf{C}^T are typically sparse because a gate, in a particular grid, would fanout to other gates in not more than k different grids⁷, with $k \ll \text{Min}(m, n)$. Therefore, the cost of computing the new sensitivity vectors, \mathbf{B}'^T and \mathbf{C}'^T by the multiplication of a sparse vector and a dense matrix is $O(m + n) = O(g)$.

For a circuit graph with V nodes and E edges, the overall time complexity of the SSTA procedure is $O(g(V + E))$. Therefore, the time complexity for our SSTA procedure, incorporating both Gaussian and non-Gaussian parameters, is the same as that of SSTA techniques considering only Gaussian variables [8], [9]. However, the complexity constant for our procedure is higher due to the steps of moments evaluation and PDF extraction, and this is not surprising since [8], [9] can be reduced to special cases of our solution.

XI. EXPERIMENTAL RESULTS

The proposed SSTA algorithm was implemented in C++, using the *MinSSTA* code [8], and tested on edge-triggered ISCAS89 benchmark circuits. All experiments were performed on Pentium-4 Linux machines with a clock speed of 3.2GHz and 2GB of memory. The *FastICA* package [27] and the *Icasso* software [28], were used to obtain the ICA transform of Equation (3). To generate samples of correlated non-Gaussian parameters, required as inputs to the *FastICA* code, we use the method of *normal copula* [23], as described in Section VI-B. For all the experiments, we generate 5000 samples of each non-Gaussian parameter to feed to the ICA module. We use the Elmore delay model and the first order Taylor series terms to represent the canonical delay model of Equation (2). However, clearly this is not a restriction, as our canonical form is similar in form to that in [8], [9], and any analytical or numerical delay model may be used, as long as the sensitivities of the delay with respect to the varying parameters can be computed.

We consider the effective channel length, L_e , the transistor width W , and the dopant concentration, N_d as the sources of variation. The parameters L_e and W are modeled as correlated sources of variations, and the dopant concentration, N_d , is modeled as an independent source of variation. The same framework can be easily extended to include other parameters of variations. For simplicity, our current implementation ignores the effect of the input signal transition time on the delay at the output port of the gate. However, according to the technique described in [29], our SSTA procedure can also be extended to incorporate and propagate the distributions of the signal transition times. As described in [29], it is possible to express slope at the output pin of the gate as a probability weighted sum of distributions of the slope from all input pins to the output pin of the gate. In our SSTA framework, we can efficiently compute these weights as closed-form probabilities, using the AWE-based PDF extraction scheme.

We use the grid-based model of [8] to generate the spatial correlations for the W and L_e parameters. Due to the lack

⁷In the case of a gate driving a global wire which spans many grids, it is highly likely that the global wire would be buffered.

of access to any real wafer data and process data files, we do not have the required information to realistically model the parameter distributions. We consider the following two cases for modeling the W and L_e parameters:

Case 1: W of gates in each grid are modeled as non-Gaussian parameters, and L_e are modeled as Gaussian variables. Section XI-A discusses the SSTA results for this case.

Case 2: L_e of gates in each grid are modeled as non-Gaussian parameters, and W are assumed to be normally distributed variables. Section XI-B discusses the SSTA results for this case.

For both cases, the independent parameter N_d is assumed to follow a Poisson distribution. The μ and σ values of the parameters are based on the predictions from [30]. For 90nm technology, we use $\mu_W = 150nm$, $\mu_{L_e} = 60nm$, $\sigma_W = 7.5nm$ and $\sigma_{L_e} = 4nm$. For the independent parameter N_d modeled as a Poisson random variable, we use $\mu_{N_d} = 10 \times 10^{17}cm^{-3}$ for both nmos and pmos. We test our SSTA procedure by comparing our results for each benchmark with 10,000 Monte Carlo (MC) simulations based on the same grid model. The samples of correlated non-Gaussian parameters for Monte Carlo simulations are also generated using the method of normal copula, as described in Section VI-B.

A. SSTA results for Case 1

For these experiments, we model W of gates in each grid as non-Gaussian parameters, and L_e of gates in each grid as Gaussian parameters. For the correlated non-Gaussian W parameters, we randomly assign to W in each grid either a uniform distribution in $[\mu_W - \sqrt{3}\sigma_W, \mu_W + \sqrt{3}\sigma_W]$, or a symmetric triangular distribution in $[\mu_W - k\sigma_W, \mu_W + k\sigma_W]$, given by:

$$f_W(w) = \frac{2(w-a)}{(b-a)(c-a)} \quad a \leq w \leq c$$

$$f_W(w) = \frac{2(b-w)}{(b-a)(b-c)} \quad c < w \leq b \quad (27)$$

where $a = \mu_w - k\sigma_w$, $c = \mu_w$, and $b = \mu_w + k\sigma_w$. The number k is chosen so that the variance of the symmetric triangular distribution described in Equation (27) is the same as σ_w^2 .

Benchmark			Error ($\frac{SSTA-MC}{MC}\%$)				Error ($\frac{MC_{Gauss}-MC}{MC}\%$)			
Name	# Cells	# Grids	μ	σ	95% Pt	5% Pt	μ	σ	95% Pt	5% Pt
s27	13	4	0.13%	0.22%	0.13%	0.57%	0.26%	0.54%	0.24%	0.81%
s1196	547	16	0.29%	0.59%	0.97%	0.83%	0.66%	1.22%	1.57%	1.35%
s5378	2958	64	-0.53%	-1.32%	-1.34%	-1.56%	0.93%	2.03%	1.93%	2.05%
s9234	5825	64	0.91%	1.81%	1.29%	-1.31%	0.87%	1.95%	2.59%	2.61%
s13207	8260	256	1.77%	2.24%	2.39%	3.03%	2.26%	3.35%	3.55%	3.11%
s15850	10369	256	1.98%	2.51%	3.14%	3.79%	2.89%	3.82%	3.51%	3.09%
s35932	17793	256	1.15%	2.82%	3.78%	3.67%	1.56%	2.56%	4.12%	4.26%
s38584	20705	256	1.71%	3.29%	3.59%	3.87%	2.09%	3.89%	4.22%	4.17%
s38417	23815	256	1.51%	3.68%	3.50%	3.61%	2.05%	4.35%	4.93%	4.88%
Avg Abs Err	-	-	1.11%	2.05%	2.24%	2.47%	1.51%	2.63%	2.96%	2.93%

TABLE III

A COMPARISON OF RESULTS OF THE PROPOSED SSTA WITH MONTE CARLO SIMULATION. W PARAMETERS ARE MODELED AS NON-GAUSSIAN VARIABLES, AND L_e PARAMETERS ARE MODELED AS GAUSSIAN VARIABLES.

Table III shows a comparison of the results of the Monte Carlo (MC) simulations with our SSTA procedure for each benchmark circuit. We compare the mean (μ), the standard deviation (σ), the 95% and the 5% quantile points of the delay

distribution obtained from our SSTA scheme with those generated from the Monte Carlo simulations, as the metrics of accuracy. As seen in Table III, the results of the proposed SSTA scheme are quite close to that of Monte Carlo analysis. The average of the absolute errors, across the nine benchmark circuits, shown in the last row of Table III, is 1.11% for μ , 2.05 % for σ , 2.24% for the 95% point, and 2.47% for the 5% quantile point. We also compare the actual Monte Carlo results with the ones obtained by incorrectly modeling the non-normal W parameters as Gaussian variables, and then performing a Monte Carlo analysis, termed as MC_{Gauss} . Columns eight to eleven of Table IV report the errors for comparison between the actual Monte Carlo results, and the ones obtained by Gaussian modeling of all parameters. As seen in the table, the errors for assuming an incorrect Gaussian distribution for W parameters, *does not* result in significant errors, implying that the circuit delay PDF does not significantly deviate from a Gaussian distribution. It should be noted that for our gate delay models, the coefficients of the L_e terms are greater than the coefficients of the W terms by a factor of about $5\times$ to $12\times$. Since the sensitivities of the Gaussian L_e terms outweigh the sensitivities of the non-Gaussian W terms, the circuit delay PDF is dominated by the normal parameters, and does not significantly diverge a normal distribution.

B. SSTA results for Case 2

For these experiments, we model L_e of gates in each grid as non-Gaussian parameters, and W of gates in each grid as Gaussian parameters. For the correlated non-Gaussian L_e parameters, we randomly assign to L_e in each grid either a uniform distribution in $[\mu_{L_e} - \sqrt{3}\sigma_{L_e}, \mu_{L_e} + \sqrt{3}\sigma_{L_e}]$, or a symmetric triangular distribution, similar to the one described by Equation (27), but replacing W by L_e .

Benchmark			Error ($\frac{SSTA-MC}{MC}\%$)				Error ($\frac{MC_{Gauss}-MC}{MC}\%$)			
Name	# Cells	# Grids	μ	σ	95% Pt	5% Pt	μ	σ	95% Pt	5% Pt
s27	13	4	-0.09%	-0.34%	-0.75%	0.79%	0.56%	3.23%	8.56%	2.04%
s1196	547	16	-0.23%	-0.67%	-0.87%	-0.53%	0.84%	8.82%	11.27%	2.21%
s5378	2958	64	0.31%	1.12%	1.21%	1.28%	0.98%	10.23%	10.91%	1.21%
s9234	5825	64	0.82%	1.78%	1.32%	-1.48%	1.88%	15.32%	15.28%	-1.83%
s13207	8260	256	1.58%	2.34%	-2.54%	2.89%	2.96%	28.13%	18.34%	-2.13%
s15850	10369	256	1.85%	-2.12%	3.36%	3.61%	2.63%	22.12%	17.62%	3.16%
s35932	17793	256	-1.07%	2.78%	4.01%	3.57%	2.34%	26.71%	19.17%	3.31%
s38584	20705	256	1.65%	-3.56%	3.89%	3.91%	2.21%	25.67%	18.28%	2.95%
s38417	23815	256	1.34%	3.78%	3.37%	3.22%	2.81%	34.62%	21.63%	2.51%
Avg Abs Err	-	-	0.99%	2.05%	2.33%	2.36%	1.91%	19.42%	15.67%	2.37%

TABLE IV

A COMPARISON OF RESULTS OF THE PROPOSED SSTA WITH MONTE CARLO SIMULATION. L_e PARAMETERS ARE MODELED AS NON-GAUSSIAN VARIABLES, AND W PARAMETERS ARE MODELED AS GAUSSIAN VARIABLES.

Table IV shows a comparison of the results of the Monte Carlo simulations with our SSTA procedure for each benchmark circuit. As seen in Table IV, the results of the proposed SSTA scheme are quite close to that of Monte Carlo analysis. The average of the absolute errors, across the nine benchmark circuits, is 0.99% for μ , 2.05 % for σ , 2.33% for the 95% point, and 2.36% for the 5% quantile point. These errors are reasonably small as compared to the accuracy penalty paid by assuming the incorrect normal distribution modeling of L_e parameters. Columns eight to eleven of Table IV show the error incurred when modeling the non-Gaussian L_e parameters as normally distributed random variables and performing Monte Carlo simulations,

termed as MC_{Gauss} , for each benchmark circuit. For instance, for the largest benchmark circuit s38417, when assuming that the non-Gaussian L_e parameters follow Gaussian distributions, the error observed is 2.81% for μ , 34.62% for σ , 21.63 % for the 95% point and 2.51% for the 5% point. Unlike, the results in Section XI-A, modeling the non-Gaussian L_e parameters as normally distributed ones, leads to significant inaccuracy in the circuit delay PDF. Due to the fact that the sensitivities of the non-Gaussian L_e terms outweigh the sensitivities of the Gaussian W terms, the correlated non-Gaussian parameters have a dominating effect on the circuit delay distribution, causing it to significantly aberrate from a normal distribution.

Benchmark			CPU Time (sec)		
Name	# Cells	# Grids	$SSTA_{Gauss}$ [8]	SSTA	MC
s27	13	4	0.0	1.1	6.0
s1196	547	16	1.2	8.3	634.2
s5378	2958	64	17.1	41.6	3214.4
s9234	5825	64	20.3	137.9	4756.6
s13207	8260	256	108.6	303.6	8532.1
s15850	10369	256	110.8	410.8	9587.8
s35932	17793	256	315.2	761.4	10156.5
s38584	20705	256	322.4	910.6	18903.3
s38417	23815	256	377.3	1235.6	22398.5

TABLE V
A RUNTIME COMPARISON THE PROPOSED SSTA WITH GAUSSIAN SSTA AND MONTE CARLO SIMULATION

Table V compares the runtime performance of our proposed SSTA algorithm with that of a Gaussian SSTA procedure [8], and the Monte Carlo simulations. As expected, our SSTA procedure is considerably faster than the Monte Carlo simulations, but has a higher runtime cost as compared to a Gaussian SSTA [8], due to the additional feature of handling non-Gaussian variables. On an average our procedure is $33\times$ faster than Monte Carlo method, but about $3\times$ slower than the Gaussian SSTA algorithm. Our approach can handle a large number of correlated and independent non-Gaussian parameters. The number of grids chosen for each benchmark circuit, shown in the third column of Table V, is equal to the number of correlated Gaussian and non-Gaussian variables. The number of independent non-Gaussian variables is the same as the number of cells in a circuit. For instance, the SSTA procedure for the circuit s13207 processes 256 correlated Gaussian variables, 256 correlated non-Gaussian variables, and 8260 independent non-Gaussian variables in about 5 mins of online runtime. Thus, our procedure scales well with the number of non-Gaussian parameters. The runtime reported in Table V does not include the time spent for the preprocessing steps of Sections VI and VII, which are carried out only once for a process and a given discretization. For the largest benchmark s38417, the preprocessing time taken to generate the ICA matrix A , and to compute the moments of the independent components is 3.5 hours.

In Figures 8 and 9, the PDF and CDF plots for the benchmark circuits s13207 and s38417 are provided. As seen in the figures, the PDF and the CDF as predicted by the proposed SSTA scheme matches well with the Monte Carlo PDF and CDF. The dashed curves in Figures 8 and 9, represent the case when the L_e parameters are incorrectly modeled as Gaussian variables with the same μ_{L_e} and σ_{L_e} as the original non-Gaussian parameters. The plots in these figures show that in the presence of correlated non-Gaussian parameters, the real circuit delay distribution deviates significantly from the one obtained by assuming

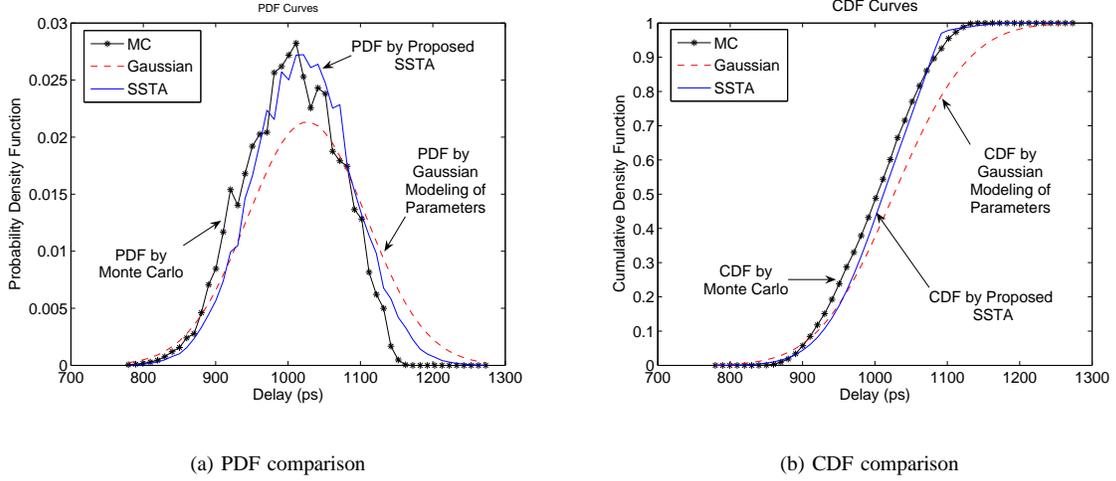


Fig. 8. A comparison of SSTA and Monte Carlo distribution for circuit s13207.

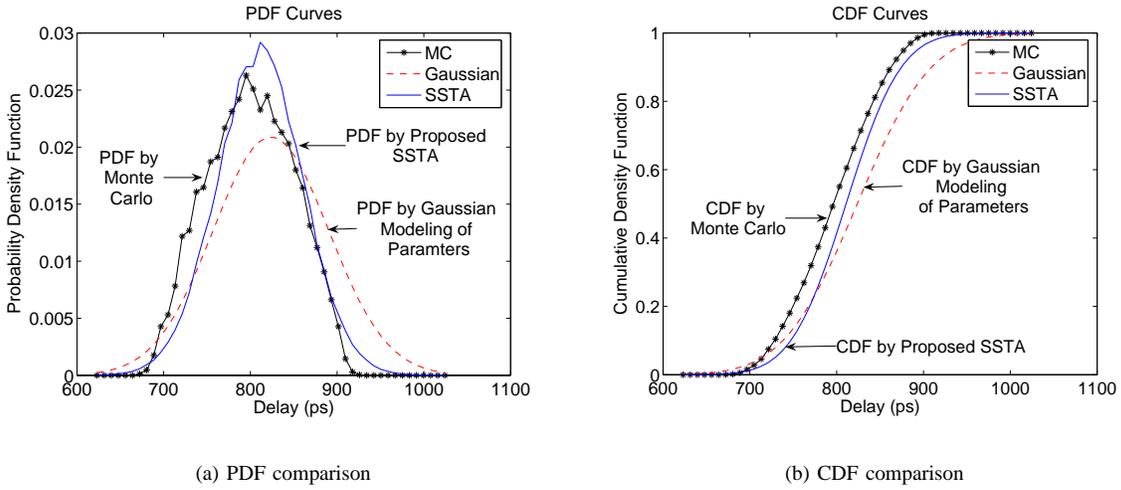


Fig. 9. A comparison of the results of SSTA and Monte Carlo for circuit s38417.

normality for parameters. The distribution functions evaluated by SSTA approach are able to match, within reasonably small errors, the real distribution functions.

XII. ACKNOWLEDGMENT

Our thanks to Hongliang Chang for providing the code of the *MinSSTA* software package and assistance with the benchmark circuits.

XIII. CONCLUSION

In this paper, we present a novel and an efficient statistical timing analysis algorithm that incorporates correlated parameters, both Gaussian and non-Gaussian. Our approach is based on PDF evaluation by matching the moments of the delay variables. We use the independent component analysis technique in our SSTA framework to handle correlations between the non-Gaussian

parameters. A time complexity analysis of our procedure shows that it is linear in the number of grids and the number of gates in the circuit. Hence, our scheme provides a scalable solution to the problem of performing SSTA in the presence of a large number of correlated non-Gaussian parameters. Experimental results validate our hypothesis that performing a Gaussian SSTA, in the presence of dominating non-Gaussian parameters of variation, could result in significant inaccuracies in estimating the PDF and CDF of the circuit delay. Our proposed SSTA procedure is able to match the real PDF and CDF of the delay much more closely, and produces the delay distributions with reasonably small errors compared to the Monte Carlo distributions, and is much faster than the Monte Carlo analysis.

REFERENCES

- [1] A. Devgan and C. Kashyap. Block-based Static Timing Analysis with Uncertainty. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 607–614, 2003.
- [2] M. Orshansky and A. Bandyopadhyay. Fast Statistical Timing Analysis Handling Arbitrary Delay Correlations. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 337–342, 2004.
- [3] 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 *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 844–852, 2005.
- [4] Y. Zhan, A. J. Strojwas, M. Sharma, and D. Newmark. Statistical Critical Path Analysis Considering Correlations. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 699–704, 2005.
- [5] J. Xiong, V. Zolotov, N. Venkateswaran, and C. Visweswariah. Criticality computation in Parameterized Statistical Timing. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 63–68, 2006.
- [6] T. Kirkpatrick and N. Clark. PERT as an Aid to Logic Design. *IBM Journal of Research and Development*, 10(2):135–141, June 1966.
- [7] C. Amin, N. Menezes, K. Killpack, F. Dartu, U. Choudhury, N. Hakim, and Y. I. Ismail. Statistical Static Timing Analysis: How simple can we get? In *Proceedings of ACM/IEEE Design Automation Conference*, pages 652–657, 2005.
- [8] H. Chang and S. S. Sapatnekar. Statistical Timing Analysis Considering Spatial Correlations Using a Single PERT-like Traversal. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 621–625, 2003.
- [9] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-Order Incremental Block-Based Statistical Timing Analysis. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 331–336, 2004.
- [10] J. Le, X. Li, and L. T. Pileggi. STAC: Statistical Timing Analysis with Correlation. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 343–348, 2004.
- [11] Y. Zhan, A. J. Strojwas, X. Li, and L. T. Pileggi. Correlation-Aware Statistical Timing Analysis with Non-Gaussian Delay Distributions. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 77–82, 2005.
- [12] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C.-P. Chen. Correlation-Preserved Non-Gaussian Statistical Timing Analysis with Quadratic Timing Model. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 83–88, 2005.
- [13] C. E. Clark. The Greatest of a Finite Set of Random Variables. *Operations Research*, 9:145–162, March-April 1961.
- [14] H. Damerdj, A. Dasdan, and S. Kolay. On the Assumption of Normality in Statistical Static Timing Analysis. In *Proceedings of TAU*, pages 2–7, 2005.
- [15] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah. Parameterized Block-Based Statistical Timing Analysis with Non-Gaussian Parameters, Nonlinear Delay Functions. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 71–76, 2005.
- [16] V. Khandelwal and A. Srivastava. A General Framework for Accurate Statistical Timing Analysis Considering Correlations. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 89–94, 2005.
- [17] J. Singh and S. Sapatnekar. Statistical Timing Analysis with Correlated Non-Gaussian Parameters using Independent Component Analysis. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 155–160, 2006.
- [18] T. Bell. An ICA page – papers, code, demos, links. Available at <http://www.cnl.salk.edu/tony/ica.html>.

- [19] A. Hyvärinen and E. Oja. Independent Component Analysis: A Tutorial. http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/, 1999.
- [20] A. Hyvärinen and E. Oja. Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13:411–430, 2000.
- [21] R. Manduchi and J. Portilla. Independent Component Analysis of Textures. In *Proceedings of IEEE Conference on Computer Vision*, volume 2, pages 1054–1060, 1999.
- [22] X. Li, J. Le, P. Gopalakrishnan, and L. Pileggi. Asymptotic Probability Extraction for non-Normal Distributions of Circuit Performance. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 2–9, 2004.
- [23] Simulating Dependent Random Variables Using Copulas. Available at <http://www.mathworks.com/products/statistics/>.
- [24] Matlab Reference Manual. Available at <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>.
- [25] Mathematica Reference Guide. Available at <http://documents.wolfram.com/v5/TheMathematicaBook/MathematicaReferenceGuide/index.htm>.
- [26] M. H. Degroot and M. J. Schervish. *Probability and Statistics*. Addison Wesley, Boston, MA, 2002.
- [27] A. Hyvärinen. Fast ICA. Available at <http://www.cis.hut.fi/projects/ica/fastica/>, 2005.
- [28] J. Himberg and A. Hyvärinen. Icaso: Software for Investigating the Reliability of ICA Estimates by Clustering and Visualization. In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pages 259–268, 2003.
- [29] H. Chang and S. S. Sapatnekar. Statistical Timing Analysis Considering Spatial Correlations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24:1467–1482, September 2005.
- [30] S. Nassif. Delay Variability: Sources, Impact and Trends. In *Proceedings of IEEE International Solid State Circuit Conference*, pages 368–369, 2000.