Statistical Timing Analysis with Correlated Non-Gaussian Parameters using Independent Component Analysis *

Jaskirat Singh Sachin Sapatnekar
Department of Electrical and 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 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. The procedure requires minimal input information: 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. For the ISCAS89 benchmark circuits, as compared to Monte Carlo simulations, we obtain average errors of 0.99% and 2.05%, respectively, in the mean and standard deviation of the circuit delay. For a circuit with $|G|$ gates and a layout with g spatial correlation grids, the complexity of our approach is $O(g|G|)$.

1. INTRODUCTION

Technology scaling brings about increased process parameter variations, causing larger spreads in circuit timing characteristics. In the face of these variations, traditional corner-based static timing analysis is inadequate, and there has been much recent work on developing statistical static timing analysis (SSTA) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. SSTA predicts the probability distribution function (PDF) and the cumulative distribution function (CDF) of the delay, given the statistical distribution of the process parameters. Existing SSTA algorithms may be path-based or block-based, may assume Gaussian or non-Gaussian distributions, may incorporate spatial correlation effects or not, etc.

The assumption of normality of process variations lends itself rather well to 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 principal components analysis (PCA) to perform a simple variable transformation [1]. 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 [11], and it is well known that some process parameters deviate significantly from a Gaussian distribution. For example, via resistances exhibit an asymmetric probability distribution [3], 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. Moreover, 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.

Some recent works [3, 7] propose SSTA methods that do away with the assumptions of normality, but none of these is scalable to a large number of non-Gaussian parameters, and none has presented a solution in the presence of correlated non-Gaussian parameter distributions. The solution in [3] employs expensive numerical integration, while [7] relies on a highly computational regression strategy. These methods can efficiently handle only a few non-Gaussian sources.

In this work, we propose an efficient parameterized block-based SSTA algorithm that can handle the case where the underlying process parameters may be spatially correlated non-Gaussian as well as Gaussian distributions. The correlations are described using a grid structure, similar to that used in [1], which works for Gaussian distributions only. Our scheme is general enough to work even for the cases when the closed-form expression of the PDF of the sources of variation is not available, and it only requires the moments of the process parameter distributions as an input. These moments are relatively easier to calculate from the process data files than the actual PDFs, and our procedure uses a moment matching scheme to generate the PDFs of the arrival time and delay variables.

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, our SSTA procedure can be extended 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 [12]. In our SSTA framework, we can efficiently compute these weights as closed-form probabilities, using the moment matching PDF extraction scheme.

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) [13, 14, 15, 16] 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 [1]. Together, we refer to this set of independent variables as the *basis set*.

*This work was supported in part by the SRC under contract 2003-TJ-1084 and the NSF under award CCR-0205227.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

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 a PDF for the delay variables, using a Padé approximation-based moment matching scheme [17].

3. **Correlation-preserving statistical operations:** Our block-based circuit traversal employs statistical sum and max operations at every step to compute the extracted PDFs of the arrival time variables. These variables are stored as a linear canonical form, obtained through a moment-matching procedure.

During our exposition, it will become amply clear that we borrow some techniques from existing algorithms from the literature. However, it is important to note that the overall algorithm is distinctly different from any existing method. 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 solution for the Gaussian case in [1].

2. NON-GAUSSIANITY IN SSTA

We use a toy circuit, shown in Figure 1, to illustrate the effects of non-Gaussian parameters on the delay distribution. We assume the width W_i and the effective length L_{e_i} for each inverter i to be the random parameters of variation. Using a first order 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.

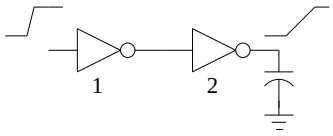
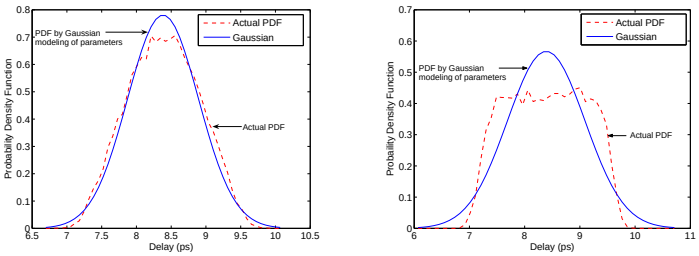


Figure 1: A simple circuit example to illustrate the effect of non-Gaussian parameters on the PDF of the circuit delay.



(a) (b)

Figure 2: PDF of the delay of the example circuit of Figure 1 for (a) uncorrelated and (b) correlated non-Gaussian and Gaussian process variables.

Next, modeling W_1, W_2 as Gaussians, and L_{e_1}, L_{e_2} as uniformly distributed random variables in $[\mu_{L_e} - \sqrt{3}\sigma_{L_e}, \mu_{L_e} + \sqrt{3}\sigma_{L_e}]$, we perform Monte Carlo simulations to evaluate the circuit delay PDF. The dashed curve shows the actual delay PDF obtained for the circuit by correctly modeling L_e as a uniform distribution, while the solid curve is the PDF obtained if the non-Gaussian L_e variables were modeled as Gaussian variables with the same mean and standard deviation as the uniform distribution. Figure 2(a) shows the PDFs for the case where all of the parameters are considered to be statistically independent, while Figure 2(b) shows the PDF when W_1 $[L_{e_1}]$ is considered to perfectly correlated with W_2 $[L_{e_2}]$. In each case, it is seen that the circuit delay PDF deviates from a Gaussian distribution due to the presence of the non-Gaussians.

Figure 2(b) suggests that the deviation from a normal distribution becomes more significant when the non-Gaussian random variables exhibit correlation. The intuition for this 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 distributions implies that the circuit delay may deviate significantly from a normal distribution.

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:

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

where D is the random variable corresponding to a gate delay or an arrival time at the input port of a gate, x_i $[y_j]$ is a non-Gaussian [Gaussian] random variable corresponding to a physical parameter variation, b_i $[c_j]$ is the first order sensitivity of the delay with respect to the i^{th} non-Gaussian [j^{th} Gaussian] parameter, z is the uncorrelated parameter which could be a Gaussian or a non-Gaussian random variable, e is the sensitivity with respect the uncorrelated variable, and n $[m]$ is the number of correlated non-Gaussian [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. We assume statistical independence between the Gaussian and non-Gaussian parameters.The mean μ is adjusted so $\mathbf{X}$ and $\mathbf{Y}$ are centered, i.e., each x_i, y_j, and z has zero-mean.

3. 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 $\sum$, a PCA transformation $\mathbf{R} = P_x\mathbf{X}$ would yield a random vector $\mathbf{R}$ comprising of Gaussian uncorrelated random variables [1]. Since for a Gaussian distribution, uncorrelatedness implies statistical independence[1], 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 [13, 14, 15, 16] 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. The ICA set up consists of having a vector $\mathbf{S}$

[1]Two 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 $E[X^iY^j] = E[X^i]E[Y^j]$. For Gaussian distributions, uncorrelatedness is identical to independence. For a general non-Gaussian distribution, independence implies uncorrelatedness, but not vice versa.

consisting of n statistically independent components, $s_1, \cdots, s_n$, and observations of n linear mixtures, $x_1, \cdots, x_n$, of the n independent components. The observed components here are 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:

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

where A is the $n \times n$ *mixing matrix*. 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(μ, σ) 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 [15]. The problem of ICA is to estimate the elements of the unknown mixing matrix A, and the samples of statistically independent components $s_1, \cdots, s_n$, given only the samples of the observed vector $\mathbf{X}$. Equation (3) can be alternatively written as:

$$\mathbf{S} = W\mathbf{X} \text{ where } s_i = \mathbf{W_i^T X} = \sum_{j=1}^{n} w_{ij} x_i \ \ \forall i = 1, \cdots, n \quad (4)$$

Here, W is the inverse of the unknown mixing matrix A. Algorithms for ICA estimate the vectors $\mathbf{W_i}$ that maximize the non-Gaussianity of $\mathbf{W_i^T X}$ by solving a nonlinear optimization problem. Typical measures of non-Gaussianity are kurtosis, negentropy, and mutual information; for a comprehensive reference on ICA, see [13, 14, 15, 16].

In the context of our SSTA algorithm, we use ICA as a preprocessing step to transform our correlated set of non-Gaussian random variables $x_i, \cdots, x_n$ to a set of statistically independent variables $s_i, \cdots, s_n$, by the relation $\mathbf{S} = W\mathbf{X}$ of Equation (4). As in [1], the chip area is first tiled into a grid, and the covariance matrix associated with $\mathbf{X}$ is determined. Using the covariance matrix, and the underlying probability distributions of the variables in $\mathbf{X}$, samples of the correlated non-Gaussian variables are generated and are given as input to the ICA module, which produces as output, the estimates of the matrices A and W. 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.*

ICA is applied to the non-Gaussian parameters $\mathbf{X}$ and PCA to the 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*:

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

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, $s_1, \cdots, s_n$ [Gaussian principal components $r_1, \cdots, r_m$].

4. 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 two possible cases:
Case I: If the closed form of the distribution of x_i is available, and it is in a standard form (e.g., Poisson or uniform), then $m_k(x_i)\ \forall\ k$ can be derived from standard mathematical tables and the parameters of the distribution. For a nonstandard distribution, $m_k(x_i)\ \forall\ k$ may be derived from the moment generating function (MGF) if a continuous closed-form PDF of the parameter is known. The moment generating function $M(t)$ of a PDF $f_{x_i}(x_i)$ is given by

$$M(t) = E[e^{tx_i}] = \int_{-\infty}^{\infty} e^{tx_i} f_{x_i}(x_i) dx_i \quad (6)$$

The k^{th} moment of x_i can then be calculated $m_k(x_i) = \frac{d^k M(t)}{dt^k} \mid_{t=0}$.
Case II: If a continuous closed-form PDF cannot be determined for a parameter, the moments can be evaluated from the process files as:

$$m_k(x_i) = \sum_x x^k \text{Probability}(x_i = x) \quad (7)$$

Given the underlying process variables and their moments, the next step after performing ICA is to determine the moments of the independent components, $s_i, \cdots, s_n$, from the moments of the correlated non-Gaussian parameters $x_i, \cdots, x_n$. These moments, $m_k(x_i) = E[x_i^k]$, are inputs to the SSTA algorithm.

Referring back to the ICA transformation of Equation (3), $\mathbf{X} = A\mathbf{S}$, we take the expectation of both sides to obtain:

$$\begin{aligned} E[x_1^k] &= E[(a_{11}s_1 + a_{12}s_2 + \cdots a_{1n}s_n)^k] \\ E[x_2^k] &= E[(a_{21}s_1 + a_{22}s_2 + \cdots a_{2n}s_n)^k] \\ &\vdots \quad\quad \vdots \\ E[x_n^k] &= E[(a_{n1}s_1 + a_{n2}s_2 + \cdots a_{nn}s_n)^k] \end{aligned} \quad (8)$$

where a_{ij} is an element of the mixing matrix A obtained via ICA. In the above equation, the LHS, which is the k^{th} moment of each component of $\mathbf{X}$, is known. The RHS can be simplified by performing an efficient multinomial expansion using binomial moment evaluation technique [17]. 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 computing by rewriting Equation (8) using $k = 2$ as

$$\begin{aligned} E[x_1^2] &= \textstyle\sum_{i=1}^{n} a_{1i}^2 E[s_i^2] + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} a_{1i}a_{1j}E[s_i]E[s_j] \\ E[x_2^2] &= \textstyle\sum_{i=1}^{n} a_{2i}^2 E[s_i^2] + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} a_{2i}a_{2j}E[s_i]E[s_j] \\ &\vdots \quad\quad \vdots \\ E[x_n^2] &= \textstyle\sum_{i=1}^{n} a_{ni}^2 E[s_i^2] + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} a_{ni}a_{nj}E[s_i]E[s_j] \end{aligned} \quad (9)$$

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 (8), 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 (9), the computation of the k^{th} moment of the independent components, $s_i, \cdots, s_n$, 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 (8) for $k = 1, \cdots, 2M$. 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 SSTA.

5. MOMENT MATCHING-BASED PDF EVALUATION

To compute the PDF/CDF of the delay or arrival time random variable we adapt the probability extraction scheme, *APEX*, proposed in [17]. Given $2M$ moments of a random variable as input

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. We refer the reader to [17] for details about the *APEX* algorithm.

We return to the example of Figure 1 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, we have:

$$\hat{D} \quad = \quad a.W + b.L_e \tag{10}$$

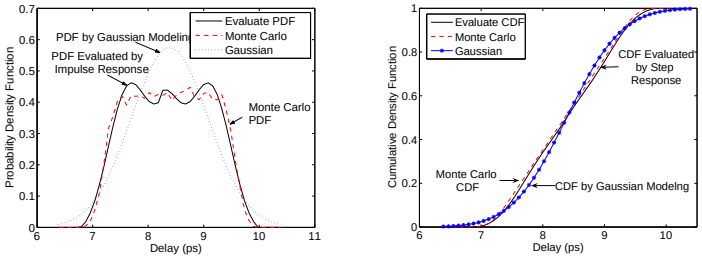
where $\hat{D} = D - \mu$, $a = a_1 + a_2$ and $b = b_1 + b_2$. Assuming W and L_e as statistically independent variables, the k^{th} moment of $\hat{D}$ can be computed by using the binomial expansion formula as:

$$m_k[\hat{D}] = \sum_{i=0}^{k} \binom{k}{i} a^i b^{k-i} m_i(W) m_{k-i}(L_e) \tag{11}$$

where all of the k moments of W and L are known from the underlying normal and uniform distributions. Having computed $2M$ moments of $\hat{D}$ from Equation (11), we can now employ the AWE-based PDF evaluation scheme to approximate the PDF of $\hat{D}$ by an impulse response as:

$$f_{\hat{D}}(\hat{d}) \quad = \quad \begin{cases} \sum_{i=1}^{M} \hat{r_i}.e^{\hat{p_i}.\hat{d}} & \hat{d} \geq 0 \\ 0 & \hat{d} < 0 \end{cases} \tag{12}$$

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



(a) PDF (b) CDF

Figure 3: Extracted PDF and CDF for the delay of the example circuit.

Figure 3 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 1 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 [17] 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 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 [17]. The PDF of D is retrieved from the PDF of $\hat{D}$ by using the relationship:

$$f_D(d) \quad = \quad \frac{1}{\sigma_D} f_{\hat{D}} \left(\frac{d - \mu_D}{\sigma_D} \right)$$

In general, given the moments of the independent components, precharacterized as in Section 4, we can compute the moments of the delay and arrival time random variables from Equation (5). For each gate, given the moments of all random variables $s_1, \cdots, s_n$, $r_1, \cdots, r_m$, 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 6.

6. 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 a block-based arrival time propagation procedure 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).

6.1 The "sum" Operation

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

$$\begin{aligned} D_1 \quad &= \quad \mu_1 + \textstyle\sum_{i=1}^{n} b'_{i_1}.s_i + \sum_{j=1}^{m} c'_{j_1}.r_j + e_1.z_1 \\ D_2 \quad &= \quad \mu_2 + \textstyle\sum_{i=1}^{n} b'_{i_2}.s_i + \sum_{j=1}^{m} c'_{j_2}.r_j + e_2.z_2 \end{aligned} \tag{13}$$

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

$$D_3 \quad = \quad \mu_3 + \textstyle\sum_{i=1}^{n} b'_{i_3}.s_i + \sum_{j=1}^{m} c'_{j_3}.r_j + e_3.z_3 \tag{14}$$

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

The one difference here, as compared to the Gaussian case (e.g., in [1]), relates to the computation of the uncorrelated non-Gaussian parameter, $e_3.z_3$. The random variable $e_3.z_3 = e_1.z_1 + e_2.z_2$, serves as a place holder to store the moments of $(e_1.z_1 + e_2.z_2)$. In other words, rather than propagating an uncorrelated component z in the canonical form, we propagate its $2M$ moments.

6.2 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:

$$f_T(t) = F_U(t) f_V(t) + F_V(t) f_u(t) \tag{15}$$

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 6, in a closed form using Equation (15).

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_j. Fortunately, it is possible to work around this by using a simple technique that permits the application of Equation (15) 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:

$$\begin{aligned} A_1 \quad &= \quad \mu_1 + \textstyle\sum_{i=1}^{n} b'_{i_1}.s_i + \sum_{j=1}^{m} c'_{j_1}.r_j + e_1.z_1 \\ A_2 \quad &= \quad \mu_2 + \textstyle\sum_{i=1}^{n} b'_{i_2}.s_i + \sum_{j=1}^{m} c'_{j_2}.r_j + e_2.z_2 \end{aligned} \tag{16}$$

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

$$A_{max} = W + \max(U, V) \quad (17)$$

where

$$\begin{aligned} W &= b'_{1_2}.s_1 + c'_{1_2}.r_1 + \textstyle\sum_{i=2}^{n} b'_{i_1}.s_i + \sum_{j=2}^{m} c'_{j_1}.r_j \quad (18)\\ U &= \mu_1 + (b'_{1_1} - b'_{1_2}).s_1 + (c'_{1_1} - c'_{1_2}).r_1 + e_1.z_1 \\ V &= \mu_2 + \textstyle\sum_{i=2}^{n}(b'_{i_2} - b'_{i_1}).s_i + \sum_{j=2}^{m}(c'_{i_2} - c'_{i_1}).r_i + e_2.z_2 \end{aligned}$$

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 variables.

Therefore, from Equations (15) and (17), we can write $A_{max} = W + T$. We begin by calculating the mean and standard deviation of A_{max}; next we will express A_{max} in the canonical form.

We use Equation (15) to obtain the distribution of A_{max}: note that this is applicable since U and V are independent by construction. Using this closed-form PDF, $f_T(t)$, we can compute μ_T from the first principles as $\mu_T = E[\max(U, V)] = \int_{-\infty}^{\infty} t f_T(t) dt$, and the variance as $\sigma_T^2 = \int_{-\infty}^{\infty} t^2 f_T(t) dt - (E[\max(U, V)])^2$.

However, Equation (15) does not provide T in the desired canonical form and it must be written in this manner for further propagation. Given that $A_{max} = W + T$, we use Equation (18) to observe that W is in canonical form. If we could express T in canonical form as well, A_{max} could be easily written in the canonical form. We employ the idea of tightness probability [2], to express $T = \max(U, V)$ as:

$$T = \mu_T + \textstyle\sum_{i=1}^{n} b'_{i_T}.s_i + \sum_{j=1}^{m} c'_{j_T}.r_j + e_T z_T \quad (19)$$

Our discussions in the previous sections provide us with all of the machinery required to efficiently compute the tightness probability, p = Probability$(U > V)$. We define a random variable $\hat{Q} = V - U$, and use the sum operation defined in Section 6.1 to express the $\hat{Q}$ in the canonical form. Next, employing the technique in Section 5, we compute the $2M$ moments of $\hat{Q}$, and evaluate the CDF, $F_{\hat{Q}}(\hat{q})$, as a step response of the approximated LTI system as:

$$F_{\hat{Q}}(\hat{q}) = \begin{cases} \sum_{i=1}^{M} \frac{\hat{r}_i}{\hat{p}_i}(e^{\hat{p}_i.\hat{q}} - 1) & \hat{q} \geq 0 \\ 0 & \hat{q} < 0 \end{cases} \quad (20)$$

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

Therefore, we see that unlike [3], we do not need to employ computationally expensive numerical integration methods in high dimensions for non-Gaussian parameters. The ability to compute the tightness probability p analytically, from the evaluated CDF of $(\hat{Q} = V - U)$, makes our SSTA procedure very efficient and allows us to process a much larger number of non-Gaussian variables.

Having computed the tightness probability, p, the sensitivities b'_{i_T}, c'_{i_T}, and z_T of $T = \max(U, V)$ in Equation (19) can be written in terms of the sensitivities of U and V. Specifically:

$$\begin{aligned} b'_{i_T} &= p.b'_{i_U} + (1-p).b'_{i_U} \ \ \forall i = 1, \cdots, n \\ c'_{j_T} &= p.c'_{j_V} + (1-p).c'_{j_V} \ \ \forall j = 1, \cdots, m \quad (21) \end{aligned}$$

Recall that the $e_T.z_T$ term in Equation (19) is a placeholder 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 $(p.e_U.z_U + (1-p).e_V.z_V)$.

The use of tightness probabilities is only a heuristic and suffers from problems of accuracy. Therefore, to reduce the error in the heuristic, we use the values of the mean, μ_T, and variance, σ_T^2, computed exactly earlier from $f_T(t)$. The coefficient e_T of the uncorrelated random variable z_T is determined so that we match the variance of the closed-form PDF of T, σ_T^2, with the variance of canonical representation of Equation (19). Thus, all of the terms required to represent $T = \max(U, V)$ back to the canonical form have been determined. As a final step, referring back to Equation (17), we perform the sum operation between W and $T = \max(U, V)$ to complete the computation of $A_{max} = \max(A_1, A_2)$.

7. COMPUTATIONAL COMPLEXITY ANALYSIS

Considering the steps to generate the ICA mixing matrix A, the PCA transform, and the moments of the independent components $s_i, \cdots, s_n$ as a one time precharacterization cost, the computational cost of the main steps in the SSTA procedure consists of performing the sum and max operations during the circuit graph traversal. 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 cost of performing the max operation is $O(M(n + m))$, where $2M$ is the number of moments evaluated for each random variable. In practice, M is upper-bounded by a small constant, and excellent solutions are obtained for $M \leq 10$. Thus, the complexity of the max operation is $O((m + n))$. For a circuit with $|G|$ gates, each with bounded fanin, the overall time complexity of the SSTA procedure is $O((m + n)|G|)$. Since m and n are both $O(g)$, where g is the number of grids, the time complexity for our SSTA procedure, incorporating both Gaussian and non-Gaussian parameters, is $O(g|G|)$, which is the same as that of SSTA techniques considering only Gaussian variables [1, 2].

8. EXPERIMENTAL RESULTS

The proposed SSTA algorithm was implemented in C++, using the *MinSSTA* code [1], and tested on edge-triggered ISCAS89 benchmark circuits. All experiments were performed on P-4 Linux machines with a clock speed of 3.2GHz and 2GB memory. The *FastICA* package [18] and the *Icasso* software [19], 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* [20]. 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 [1, 2].

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 [12], our SSTA procedure can also be extended to incorporate and propagate the distributions of the signal transition times. As described in [12], one can 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 [1] to generate the spatial correlations for the W and L_e parameters. Due to the lack 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 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 in $[\mu_{L_e} - 3.\sigma_{L_e}, \mu_{L_e} + 3.\sigma_{L_e}]$. The independent parameter N_d is assumed to follow a Poisson distribution. The μ and σ values of

Benchmark			Error ($\frac{SSTA-MC}{MC}$%)				Error ($\frac{MC_{Gauss}-MC}{MC}$%)				CPU Time (sec)		
Name	# Cells	# Grids	μ	σ	95% Pt	5% Pt	μ	σ	95% Pt	5% Pt	$SSTA_{Gauss}$[1]	SSTA	MC
s27	13	4	-0.09%	-0.34%	-0.75%	0.79%	0.56%	3.23%	8.56%	2.04%	0.0	1.1	6.0
s1196	547	16	-0.23%	-0.67%	-0.87%	-0.53%	0.84%	8.82%	11.27%	2.21%	1.2	8.3	634.2
s5378	2958	64	0.31%	1.12%	1.21%	1.28%	0.98%	10.23%	10.91%	1.21%	17.1	41.6	3214.4
s9234	5825	64	0.82%	1.78%	1.32%	-1.48%	1.88%	15.32%	15.28%	-1.83%	20.3	137.9	4756.6
s13207	8260	256	1.58%	2.34%	-2.54%	2.89%	2.96%	28.13%	18.34%	-2.13%	108.6	303.6	8532.1
s15850	10369	256	1.85%	-2.12%	3.36%	3.61%	2.63%	22.12%	17.62%	3.16%	110.8	410.8	9587.8
s35932	17793	256	-1.07%	2.78%	4.01%	3.57%	2.34%	26.71%	19.17%	3.31%	315.2	761.4	10156.5
s38584	20705	256	1.65%	-3.56%	3.89%	3.91%	2.21%	25.67%	18.28%	2.95%	322.4	910.6	18903.3
s38417	23815	256	1.34%	3.78%	3.37%	3.22%	2.81%	34.62%	21.63%	2.51%	377.3	1235.6	22398.5

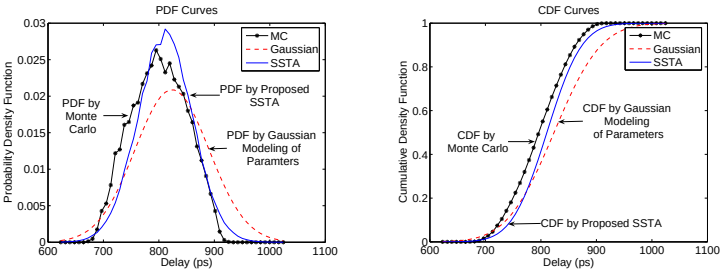
Table 1: Comparison results of the proposed SSTA with Monte Carlo simulation

the parameters are based on the predictions from [21]. 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.

Table 1 shows a comparison of the results of the Monte Carlo (MC) simulations with our SSTA procedure for each benchmark using 10,000 MC samples, based on the same grid model. The samples of correlated non-Gaussian parameters for MC simulations are also generated using the method of normal copula. 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 MC simulations as the metrics of accuracy. As seen in Table 1, the results of the proposed SSTA scheme are quite close to that of MC simulations. 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 1% quantile point. These errors are reasonably small as compared to the accuracy penalty paid by assuming the incorrect normal distribution modeling of parameters. Columns eight to eleven of Table 1 show the error incurred when modeling the non-Gaussian L_e parameters as normally distributed random variables and performing MC 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. Thus, modeling the non-Gaussian parameters as normally distributed ones leads to significant inaccuracy.

As expected, our SSTA procedure is considerably faster than the MC simulations, but has a higher runtime cost as compared to a Gaussian SSTA [1], due to the additional feature of handling non-Gaussian variables. 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 1, 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 Gaussians variables, 256 correlated non-Gaussian variables, and 8260 independent non-Gaussian variables in about 5 mins of online run time. Thus, our procedure scales well with the number of non-Gaussian parameters. The run time reported in Table 1 does not include the time spent for the preprocessing steps of Sections 3 and 4, 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 about 3.5 hours.

In Figure 4, the PDF and CDF plots for the largest benchmark circuit s38417 are provided. As seen in the Figure, 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 Figure 4 represents 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 Figure 4 show that in the presence of correlated non-Gaussian parameters, the real circuit delay distribution deviates significantly from the one obtained by assuming normality for parameters. The distribution functions evaluated by SSTA approach are able to match, within reasonably small errors, the real distribution functions.



(a) PDF comparison (b) CDF comparison

Figure 4: A comparison of SSTA and MC distribution for circuit s38417.

9. REFERENCES

[1] H. Chang and S. S. Sapatnekar. Statistical Tming Analysis Considering Spatial Correlations using a single pert-like traversal. In *Proc. IEEE/ACM ICCAD*, pages 621–625, 2003.

[2] C. Visweswariah *et al*. First-Order Incremental Block-Based Statistical Timing Analysis. In *Proc. ACM/IEEE DAC*, pages 331–336, 2004.

[3] H. Chang *et al*. Parameterized Block-Based Statistical Timing Analysis with Non-Gaussian Parameters, Nonlinear Delay Functions. In *Proc. ACM/IEEE DAC*, pages 71–76, 2005.

[4] A. Devgan and C. Kashyap. Block-based Static Timing Analysis with Uncertainty. In *Proc. IEEE/ACM ICCAD*, pages 607–614, 2003.

[5] C. Amin *et al*. Statistical Static Timing Analysis: How simple can we get? In *Proc. ACM/IEEE DAC*, pages 652–657, 2005.

[6] M. Orshansky and A. Bandyopadhyay. Fast Statistical Timing Analysis Handling Arbitrary Delay Correlations. In *Proc. ACM/IEEE DAC*, pages 337–342, 2004.

[7] V. Khandelwal and A. Srivastava. A General Framework for Accurate Statistical Timing Analysis Considering Correlations. In *Proc. ACM/IEEE DAC*, pages 89–94, 2005.

[8] Y. Zhan *et al*. Correlation-Aware Statistical Timing Analysis with Non-Gaussian Delay Distributions. In *Proc. ACM/IEEE DAC*, pages 77–82, 2005.

[9] L. Zhang *et al*. Correlation-Preserved Non-Gaussian Statistical Timing Analysis with Quadratic Timing Model. In *Proc. ACM/IEEE DAC*, pages 83–88, 2005.

[10] J. Le, X. Li, and L. T. Pileggi. STAC: Statistical Timing Analysis with Correleation. In *Proc. ACM/IEEE DAC*, pages 343–348, 2004.

[11] H. Damerdji, A. Dasdan, and S. Kolay. On the Assumption of Normality in Statistical Static Timing Analysis. In *Proc. TAU*, pages 2–7, 2005.

[12] H. Chang and S. S. Sapatnekar. Statistical Tming Analysis Considering Spatial Correlations. In *IEEE Trans. on CAD*, volume 24, pages 1467–1482, Sep 2005.

[13] Tony Bell. An ICA page-papers, code, demos, links. `http://www.cnl.salk.edu/ tony/ica.html`.

[14] A. Hyvärinen and E. Oja. Independent Component Analysis: A Tutorial. `http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/`.

[15] A. Hyvärinen and E. Oja. Independent component analysis:Algorithms and applications. In *Neural Networks*, volume 13, pages 411–430, 2000.

[16] R. Manduchi and J. Portilla. Independent component analysis of textures. In *Proc. IEEE Conference on Computer Vision*, volume 2, pages 1054–1060, 1999.

[17] X. Li *et al*. Asymptotic Probability Extraction for non-Normal Distributions of Circuit Performance. In *Proc. IEEE/ACM ICCAD*, pages 2–9, 2004.

[18] Available at `http://www.cis.hut.fi/projects/ica/fastica/`.

[19] Available at `http://www.cis.hut.fi/projects/ica/icasso/`.

[20] Simulating Dependent Random Variables Using Copulas. `http://www.mathworks.com/products/statistics/`.

[21] S. Nassif. Delay Variability: Sources, Impact and Trends. In *Proc. ISSCC*, pages 368–369, 2000.