

# STATISTICAL TIMING ANALYSIS CONSIDERING SPATIAL CORRELATIONS USING A SINGLE PERT-LIKE TRAVERSAL

Hongliang Chang <sup>†</sup>

Sachin S. Sapatnekar <sup>‡</sup>

<sup>†</sup> Department of Computer Science and Engineering, University of Minnesota

<sup>‡</sup> Department of Electrical and Computer Engineering, University of Minnesota

## Abstract

We present an efficient statistical timing analysis algorithm that predicts the probability distribution of the circuit delay while incorporating the effects of spatial correlations of intra-die parameter variations, using a method based on principal component analysis. The method uses a PERT-like circuit graph traversal, and has a run-time that is linear in the number of gates and interconnects, as well as the number of grid partitions used to model spatial correlations. On average, the mean and standard deviation values computed by our method have errors of 0.2% and 0.9%, respectively, in comparison with a Monte Carlo simulation.

## 1. INTRODUCTION

Device and interconnect parameters in new technology generations show a significant amount of variability due to process variations, and the prediction of circuit performance is becoming a challenging task. Conventional static timing analysis (STA) handles variability by analyzing a circuit at multiple process corners, and it is generally accepted that such an approach is inadequate. An alternative approach that overcomes these problems is statistical STA, which treats delays not as fixed numbers, but as probability density functions (PDFs), taking the statistical distribution of parametric variations into consideration while analyzing the circuit.

Process variations can be classified as follows: *inter-die variations* are the variations from die to die, while *intra-die variations* correspond to variability within a single chip. Inter-die variations affect all the devices on the same chip similarly, while intra-die variations affect different devices differently on the same chip. It used to be the case that the inter-die variations dominated intra-die variations, so that the latter could be safely neglected. However, in modern technologies, intra-die variations are rapidly and steadily growing and their effects significantly affect the variability of performance parameters on a chip. A number of publications on statistical timing analysis have focused on circuit performance prediction considering intra-die variation [1–4]. However, most prior work has ignored intra-chip spatial correlations by simply assuming zero correlations among devices on the chip. The difficulty in considering spatial correlations between parameters is that it always results in complicated path correlation structures that are hard to deal with. The authors of [5] consider correlation between delays among the transistors inside a single gate, but not correlations between gates. The work in [6] uses a Monte Carlo sampling-based framework to analyze circuit timing on a set of selected sensitizable true paths. Another method in [7] computes path correlations on the basis of pair-wise gate delay covariances and used analytic method to derive lower and upper bounds of circuit delay. The statistical timing analyzer in [8] takes into account capacitive coupling and intra-die process variation to estimate the worst case delay of critical path. The approach in [9] proposes a model for spatial correlation and a method of statistical timing analysis to compute the delay distribution of a specific critical path. However, the PDF for a critical path may not be a good predictor of the distribution of the circuit delay (which is the maximum of all path

delays), as explained in Section 2. Moreover, any strictly path-based method will eventually be faced with an explosion in the number of critical paths.

We propose an algorithm for statistical STA that computes the distribution of circuit delay while considering correlations due to path reconvergence as well as spatial correlations. We model the circuit delay as a correlated multivariate normal distribution, considering both gate and wire delay variations. The complexity of the algorithm is  $O(n \times (N_g + N_I))$ , which is linear in the number of gates  $N_g$  and interconnects  $N_I$ , and also linear in the number of grids  $n$  that are used to model the variational regions. In other words, the cost is, at worst,  $n$  times the cost of a deterministic STA.

## 2. PROBLEM FORMULATION

Under process variations, parameter values such as the transistor gate length, width, the metal line width and the metal line height are random variables. Some of these variations are deterministic, while others are random: this work will focus on the effects of random variations, and will model these parameters as random variables. The gate and interconnect delays, as functions of these parameters, also become random variables. The task of statistical STA is to find the PDF of the circuit delay under these variations.

Since we will employ a PERT-like traversal to analyze the distribution of circuit delay, we will work on the *statistical timing graph* of a circuit. This is similar to the timing graph for deterministic STA, but each node (gate delay) or edge (interconnect delay) is weighted by a random variable instead of a deterministic delay, and these random variables may be uncorrelated or correlated.

**Definition 2.1** *The problem of statistical STA for a circuit is that of finding the probability distribution of  $\max(D_1, \dots, D_{n_{path}})$ , where  $D_i$  is the path delay distribution of a path from the source node to the sink node in the statistical timing graph of the circuit.*

For the same nominal design, the identity of the longest path may change, depending on the random values taken by the process parameters. Therefore, finding the delay distribution of one critical path at a time is not enough, and correlations between paths due to structural correlations (reconvergent fanouts) and spatial correlations must be considered in finding the PDF of the maximum of all path delays. Such an analysis is essential for finding the probability of failure of a circuit, which is available from the cumulative density function (CDF) of the circuit delay.

In our work, we model all process parameter values as normally distributed random variables, and justify the modeling of node and edge weights as normally distributed random variables that are functions of process parameters. Since we consider spatial correlations of parameters, some of the random variables are correlated. Furthermore, we model the circuit delay as a multivariate normal distribution as well, and show that the loss of accuracy under this approximation is not significant.

## 3. MODELING PARAMETER VARIATIONS

### 3.1. Components of Variations

We will focus on intra-die variations; however, the method can be easily extended to include inter-die variations. We model the pa-

This work was supported in part by the NSF under award CCR-0205227 and by the SRC under contract 2003-TJ-1092.

parameter variations as location-dependent normally distributed random variables as follows [10]:

$$p = \bar{p} + \delta_x \cdot x + \delta_y \cdot y + N(0, \Sigma) \quad (1)$$

where  $\bar{p}$  is the nominal design parameter value,  $(x, y)$  is its die location,  $\delta_x$  and  $\delta_y$  are the location-dependent gradient of parameter, and  $N(0, \Sigma)$  is the random component which is a multivariate normal distribution, where  $\Sigma$  is the covariance matrix.

In this work, for transistors, we consider the following geometry parameters as random variables: transistor length  $L_g$  and width  $W_g$ , gate oxide thickness  $T_{ox}$ , doping concentration density  $N_a$ ; for interconnect, at each metal layer, we consider the following parameters: metal width  $W_{int_l}$ , metal thickness  $T_{int_l}$  and ILD thickness  $H_{ILD_l}$ , where the subscript  $l$  represents that the random variable is of layer  $l$ , where  $l = 1 \dots n_{layers}$ . We believe that this framework is general enough that it can be applied to handle variations in other parameters as well.

### 3.2. Spatial Correlations

To model the intra-die spatial correlations of parameters, we partition the die into  $nrow \times ncol = n$  grids. Since devices [wires] close to each other are more likely to have more similar characteristics than those placed far away, we assume perfect correlations among those in nearby grids and low or zero correlations in far-away grids. For example, in Figure 1: gates  $a$  and  $b$  (whose sizes are shown to be exaggeratedly large) are located in the same grid square, and it is assumed that their parameter variations (such as the variations of their transistor gate length), are always identical. Gates  $a$  and  $c$  lie in neighboring grids, and their parameter variations are not identical but highly correlated due to their spatial proximity. On the other hand, gates  $a$  and  $d$  are far away from each other, their parameters are uncorrelated.

Under this model, a parameter variation in a single grid at location  $(x, y)$  can be modeled using a single random variable  $p(x, y)$ . For each type of parameter,  $n$  random variables are needed, each representing the value of a parameter in one of the  $n$  grids. In addition, we assume that correlation exists only among the same type of parameters and there is no correlation between different types of parameters. For example, the  $L_g$  values for transistors in a grid are correlated with those in nearby grids, but are uncorrelated with other parameters such as  $T_{ox}$  or  $W_{int_l}$  in any grid. (Note here that we consider interconnect parameters in different layers to be “different types of parameters,” e.g.,  $W_{int_1}$  and  $W_{int_2}$  are uncorrelated.) For each type of parameter, a (sparse) correlation matrix of size  $n \times n$  represents the spatial correlations of such a structure. In this work, we use the correlation matrix derived from the spatial correlation model used in [9].

## 4. STATISTICAL TIMING ANALYSIS ALGORITHM

### 4.1. Modeling Gate/Interconnect Delay PDFs

In this section, we will show how the variations in the process parameters are translated into probability density functions (PDFs) that describe the variations in the gate and interconnect delays that correspond to the weights on the nodes and edges, respectively, of the statistical timing graph. We will use first-order Taylor expansions to approximate the distributions of the gate or interconnect delays.

In this work, we use the Elmore delay model for simplicity to calculate the interconnect delays<sup>1</sup>. The interconnect delay can be expressed as a function of the process parameters of the interconnect and the receivers, such as  $W_{int_l}$ ,  $T_{int_l}$ ,  $H_{ILD_l}$ ,  $W_g$ ,  $L_g$  and  $T_{ox}$ . Recall that under our model, we divide the chip area into grids so that the parameter variations within a grid are identical,

<sup>1</sup>However, it should be emphasized that any delay model may be used, and all that is needed is the sensitivity of the delay to the process parameters, say, through a full circuit simulation and adjoint network analysis.

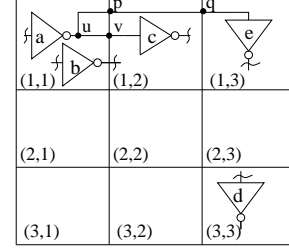


Figure 1: Grid model for spatial correlations

but those in different grids exhibit spatial correlations. Now consider an interconnect tree with several different segments that reside in different grids. The delay variations in the tree are affected by the parameter variations of wires in all grids that the tree traverses. For example, in Figure 1, consider the interconnect tree driven by gate  $a$  which passes through grid  $(1, 1)$  and grid  $(1, 2)$ , the distribution of the tree delay is actually a function of random variables of interconnect parameters in both grids and should incorporate any correlations between these random variables. Similarly, if the gates that the interconnect tree drives reside in different grid locations, the interconnect delay to any sink is also a function of random variables of gate parameters of all grids in which the receivers are located. Using a first order Taylor expansion, the distribution of interconnect delay can be approximated as a normal distribution by:

$$d_{int} = d_{int}^0 + \sum_{i \in \Gamma_g} \left[ \frac{\partial d}{\partial L_g^i} \right]_0 \Delta L_g^i + \sum_{i \in \Gamma_g} \left[ \frac{\partial d}{\partial W_g^i} \right]_0 \Delta W_g^i \quad (2)$$

$$+ \sum_{i \in \Gamma_g} \left[ \frac{\partial d}{\partial T_{ox}^i} \right]_0 \Delta T_{ox}^i + \sum_{l=1}^{n_{layers}} \left\{ \sum_{i \in \Gamma_{int_l}} \left[ \frac{\partial d}{\partial W_{int_l}^i} \right]_0 \Delta W_{int_l}^i \right.$$

$$\left. + \sum_{i \in \Gamma_{int_l}} \left[ \frac{\partial d}{\partial T_{int_l}^i} \right]_0 \Delta T_{int_l}^i + \sum_{i \in \Gamma_{int_l}} \left[ \frac{\partial d}{\partial H_{ILD_l}^i} \right]_0 \Delta H_{ILD_l}^i \right\}$$

where  $d_{int}^0$  is the interconnect delay value calculated with nominal values of parameters.  $\Gamma_g$  is the set of indices of grids that all the receivers reside in.  $\Gamma_{int}$  is the set of indices of grids that the interconnect tree traverses.  $\Delta L_g^i = L_g^i - \mu_{L_g^i}$  where  $L_g^i$  is the random variable representing transistor length in the  $i^{\text{th}}$  grid.  $\Delta W_g^i$ ,  $\Delta T_{ox}^i$ ,  $\Delta W_{int_l}^i$ ,  $\Delta T_{int_l}^i$  and  $\Delta H_{ILD_l}^i$  are similarly defined. The subscript “0” next to each sensitivity represents the fact that it is evaluated at the nominal value of the delay.

The gate delay can be similarly approximated as a function of parameters such as  $L_g$ ,  $W_g$ ,  $T_{ox}$  and  $N_a$  of the gate and its receivers, and  $W_{int_l}$ ,  $T_{int_l}$ ,  $H_{ILD_l}$  of the interconnect that it drives.

### 4.2. Orthogonal Transformation of Correlated Variables

Correlation due to reconvergent paths is known to be a problem for statistical timing analysis, even when spatial correlations are ignored. When these relationships among process parameters are taken into consideration, the correlation structure becomes even more complicated. To make the problem tractable, we use the Principal Component Analysis (PCA) technique [11] to transform the set of correlated parameters into an uncorrelated set.

Given a set of correlated random variables  $\vec{X}$  with a covariance matrix  $R$ , the PCA technique transforms the set into a set of mutually orthogonal random variables,  $\vec{X}'$ , such that each member of  $\vec{X}'$  has zero mean and unit variance. The set  $\vec{X}'$  is called the set of principal components and its size is no larger than the size of  $\vec{X}$ .

Any variable  $x_i \in \vec{X}$  can then be expressed as a linear function of the principal components in  $\vec{X}'$ :

$$x_i = \left( \sum_j \sqrt{\lambda_i} \cdot v_{ij} \cdot x'_j \right) \sigma_i + \mu_i \quad (3)$$

where  $x'_j \in \vec{X}'$ ,  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of the covariance matrix  $R$ ,  $v_{ij}$  is the  $i^{\text{th}}$  element of the  $j^{\text{th}}$  eigenvector of  $R$ , and  $\sigma_i$  and  $\mu_i$  are, respectively, the mean and standard deviation of  $x_i$ .

For instance, let  $\vec{L}_g$  be the set of random variables representing transistor gate length variations in all grids and the set of random variables is of multivariate normal distribution with covariance matrix  $R_{L_g}$ . Let  $\vec{L}'_g$  be the set of principal components computed by PCA. Then any  $L_g^i \in \vec{L}_g$  representing the variation of transistor gate length of the  $i^{\text{th}}$  grid can then be expressed as a linear function of the principal components:

$$L_g^i = \mu_{L_g^i} + a_{i1} \times l_g^{\prime 1} + \dots + a_{it} \times l_g^{\prime t} \quad (4)$$

where  $\mu_{L_g^i}$  is the mean of  $L_g^i$ ,  $l_g^{\prime i}$  is a principal component in  $\vec{L}'_g$ , all  $l_g^{\prime i}$  are independent with zero means and unit variances, and  $t$  is the total number of principal components in  $\vec{L}'_g$ .

In this way, any random variable in  $\vec{W}_g, \vec{T}_{ox}, \vec{N}_a, \vec{W}_{intl}, \vec{T}_{intl}$  and  $\vec{H}_{ILD1}$  can be expressed as a linear function of their corresponding principal component set  $\vec{W}'_g, \vec{T}'_{ox}, \vec{N}'_a, \vec{W}'_{intl}, \vec{T}'_{intl}$  and  $\vec{H}'_{ILD1}$ . Superposing the set of rotated random variables of parameters on the random variables in gate or interconnect delay as in equation (2), the expression of gate or interconnect delay is then changed to the linear combination of principal components of all parameters:

$$d = d_0 + k_1 \times p'_1 + \dots + k_m \times p'_m \quad (5)$$

where  $p'_i \in \vec{P}'$  and  $\vec{P}' = \vec{L}'_g \cup \vec{W}'_g \cup \vec{T}'_{ox} \cup \vec{N}'_a \cup \vec{W}'_{intl} \cup \vec{T}'_{intl} \cup \vec{H}'_{ILD1}$  and  $m$  is the size of  $\vec{P}'$ . Equation (5) has the following properties:

**Property 1** Since all  $p'_i$  are orthogonal, the variance of  $d$  can be simply computed as  $\sum_{i=1}^m k_i^2$ .

**Property 2** The covariance between  $d$  and any principal component  $p'_i$  is given by:  $cov(d, p'_i) = k_i \sigma_{p'_i}^2 = k_i$ , i.e., the coefficient of  $p'_i$  is exactly the covariance between  $d$  and  $p'_i$ .

**Property 3** Let  $d_i$  and  $d_j$  be two random variables:

$$d_i = d_i^0 + k_{i1} \times p'_1 + \dots + k_{im} \times p'_m \quad (6)$$

$$d_j = d_j^0 + k_{j1} \times p'_1 + \dots + k_{jm} \times p'_m \quad (7)$$

The covariance of  $d_i$  and  $d_j$ ,  $cov(d_i, d_j)$ , can be computed by  $\sum_{r=1}^m k_{ir} k_{jr}$ .

### 4.3. PERT-like Traversal of Statistical STA

Using the techniques discussed up to this point, all nodes and edges of the statistical timing graph may be modeled as normally distributed random variables. In this section, we will describe a procedure for finding the distribution of the statistical longest path in the graph.

In conventional deterministic STA, the PERT algorithm can be used to find the longest path in a graph by traversing it in topological order using two types of functions: the *sum* function, and the *max* function. To apply the PERT algorithm in our statistical timing analysis, we must find the probability distributions of the *sum* and *max* functions of a set of correlated Gaussian random variables:

1)  $d_{sum} = \sum_{i=1}^n d_i$ , and

2)  $d_{max} = \max(d_1, \dots, d_n)$ ,

where  $d_i$  is a random variable representing either gate delay or wire delay expressed in the form as equation (6).

### Computing the distribution of the *sum* function

The computation of the distribution of *sum* function is simple. Since  $d_{sum}$  is a linear combination of normally distributed random variables,  $d_{sum}$  is a normal distribution with the mean  $\mu_{d_{sum}} = \sum_{i=1}^n d_i^0$  and variance  $\sigma_{d_{sum}}^2 = \sum_{j=1}^m \sum_{i=1}^n k_{ij}^2$ .

### Computing the distribution of the *max* function

The *max* function of  $n$  normally distributed random variables,  $d_{max}$ , is, strictly speaking, not Gaussian. However, we have found that it can be approximated closely by a Gaussian. This idea is similar in spirit to Berkelaar's approach in [1], although it is more general since Berkelaar's work restricted its attention to delay random variables that were uncorrelated. In this work, we approximate  $d_{max} \sim N(\mu_{d_{max}}, \sigma_{d_{max}}^2)$  as a linear function of all the principal components  $p'_1 \dots p'_m$ :

$$d_{max} = \mu_{d_{max}} + a_1 p'_1 + \dots + a_m p'_m \quad (8)$$

From *Property 2* of Section 4.2, we know that the coefficient  $a_r$  equals  $cov(d_{max}, p_r)$ . Then the variance of the expression on the right hand side of equation (8) is computed as  $s_0^2 = \sum_{r=1}^m a_r^2 = \sum_{r=1}^m cov^2(d_{max}, p_r)$ . Since this is merely an approximation, there may be a difference between the value  $s_0^2$  and the actual variance  $\sigma_{d_{max}}^2$  of  $d_{max}$ . To diminish the difference, we normalize the value of  $a_r$  by  $cov(d_{max}, p_r) \cdot \frac{\sigma_{d_{max}}}{s_0}$ .

We can see now that to find the linear approximation for  $d_{max}$ , we need to compute  $\mu_{d_{max}}$ ,  $\sigma_{d_{max}}$  and  $cov(d_{max}, p_i)$ . If  $\xi$  and  $\eta$  are two random variables,  $\xi \sim (\mu_1, \sigma_1)$ ,  $\eta \sim (\mu_2, \sigma_2)$ , with a correlation coefficient of  $r(\xi, \eta) = \rho$ , then the mean  $\mu_t$  and the variance  $\sigma_t^2$  of  $t = \max(\xi, \eta)$  can be approximated using the closed-form formula provided in [12]. Moreover, if  $\gamma$  is another normally distributed random variable and  $r(\xi, \gamma) = \rho_1$ ,  $r(\eta, \gamma) = \rho_2$ , then the correlation between  $\gamma$  and  $t$  can also be obtained.

**Example** Let us consider the evaluation of  $d_{max}$  for a two-variable *max* function,  $d_{max} = \max(d_i, d_j)$ , using the results from [12] and the properties introduced in Section 4.2. The procedure is as follows:

1. Compute the means and standard deviations of  $d_i$  and  $d_j$ :  $\mu_{d_i}$ ,  $\sigma_{d_i}$  and  $\mu_{d_j}$ ,  $\sigma_{d_j}$ , respectively as described in *Property 1* of Section 4.2.
2. Find  $r(d_i, d_j)$ , the correlation coefficient between  $d_i$  and  $d_j$  by  $cov(d_i, d_j) \sigma_{d_i}^{-1} \sigma_{d_j}^{-1}$ , where  $cov(d_i, d_j)$ , the covariance of  $d_i$  and  $d_j$ , can be computed using *Property 3* in Section 4.2. Now if  $r(d_i, d_j) = 0$  and  $\sigma_{d_i} = \sigma_{d_j}$ , set  $d_{max}$  to be identical to  $d_i$  or  $d_j$ , whichever has larger mean value and we can stop here; otherwise, we will continue to the next step.
3. Calculate the mean  $\mu_{d_{max}}$  and variance  $\sigma_{d_{max}}^2$  of  $d_{max}$  using the results from reference [12].
4. Find all coefficients  $a_r$  of  $p'_r$ . According to *Property 2*,  $a_r = cov(d_{max}, p'_r)$ , also,  $cov(d_i, p'_r) = k_{ir}$  and  $cov(d_j, p'_r) = k_{jr}$ . Applying the results from [12], the values of  $cov(d_{max}, p'_r)$  and thus  $a_r$  can be calculated.
5. After all of the  $a_r$ 's have been calculated, determine  $s_0 = \sqrt{\sum_{r=1}^m a_r^2}$ . Reset each coefficient  $a_r = a_r \frac{\sigma_{d_{max}}}{s_0}$ .

The calculation of the two-variable *max* function can easily be extended for an  $n$ -variable *max* function by repeating the steps of the two-variable case recursively. During the computation of  $n$ -variable *max* function, some inaccuracy could be introduced since

we approximate the *max* function as normal even though it is not really normal, and proceed with further recursive calculations. We will show in Section 6 that such inaccuracies are not significant and the results match very well with the simulation results from a Monte Carlo analysis.

At this point, not only the nodes and edges, but also the results of *sum* and *max* functions are expressed as linear functions of the principal components, and thus all path delays in the statistical timing graph also become the linear functions of the principal components. Therefore, we can find the distribution of statistical longest path using a PERT traversal, incorporating the computation of *sum* and *max* functions described above. To further speed up the process, several techniques may be used:

1. Before we run the statistical timing analyzer, we perform one run of deterministic timing analysis to determine a loose bound on the best-case and worst-case delays for all paths. Any path whose worst-case delay is less than the best-case delay of the longest path will never be critical, we can safely remove the edges that lie only on such paths.
2. During the *max* operation of statistical STA, if the value of  $mean + 3 \cdot \sigma$  of one path has a lower delay than the value of  $mean - 3 \cdot \sigma$  of another path, we can simply calculate the *max* function ignoring the former path.

## 5. COMPUTATIONAL COMPLEXITY

We present a run time complexity analysis here to show which factors most greatly affect the CPU time of the algorithm. The PCA step requires computation of eigenvectors and eigenvalues of the covariance matrix. Its time complexity is  $O(p \cdot n^3)$ , where  $n$  is total number of grids divided and  $p$  is the number of parameters considered. Since the PCA step is to be performed just once for each process, we do not consider it here; in our experiments, it was performed in Matlab in less than 1 second. The run time of the algorithm can be divided into:

1. The time required to find the delay distribution of the gate and interconnect: This run time depends on how many different grids the interconnect passes through and how many grids the gates are located in, and in general these numbers are bounded by constant numbers. The run time is also proportional to the total number of principal components, since we perform orthogonal transformation at each wire segment of interconnect. For each random variable, the number of principal components is no more than the total number of grids  $n$  partitioned on the chip. Thus, the time required to find the distribution of a single gate or wire can be estimated as  $O(n)$ . If  $N_g$  is the total number of gates and  $N_I$  the number of net connections in the circuit, the time of this part can be estimated as  $O(n \cdot (N_g + N_I))$ .
2. The time required to evaluate the *max* function: The cost of this operation is proportional to the number of random variables involved in the max operation and the number of principal components of each random variable. The *max* operation is used at all multi-input gates and at the last level (sink node) where the maximum circuit delay is computed. This number can be upper bounded by the total number of net connections  $N_I$  in the circuit. Thus, the run time of this part is  $O(n \cdot N_I)$ .
3. The time required to evaluate the *sum* function: The *sum* operation must be performed at all gates and interconnects encountered during the PERT-like traversal. A single *sum* operation requires  $O(n)$ , and therefore, the total complexity for this part is  $O(n \cdot (N_g + N_I))$ .

Therefore, the run time complexity of the algorithm is  $O(n \cdot (N_g + N_I))$ , which is  $n$  times that of deterministic STA.

Table 1: Parameters used in the experiments

Parameters	$L_g$	$W_g$	$T_{ox}$	$N_a$ ( $\times 10^{17} cm^{-3}$ )	$W_{int}$	$T_{int}$	$H_{ILD}$
	(nm)	(nm)	(nm)	nmos/pmos	(nm)	(nm)	(nm)
Mean	180	270	4.1	2.3549/4.1589	180	320	70
Deviation (%)	25%	20%	10%	10%	25%	20%	35%

## 6. EXPERIMENTAL RESULTS

The proposed algorithm was implemented in C++ as the software package “*MinnSSTA*”, and tested on the edge-triggered ISCAS89 benchmark circuits. All experiments were run on a Linux PC with a 2.0GHz CPU and 256MB memory. As in [7,9], we use 180nm technology parameter and 2-metal layer interconnect model. The process parameters and their max percentages of deviations from the nominal values (Table 1) used here are based on predictions from [13, 14]. In the experiments, the parameter variations are set up so that spatial and random components each accounts for half of the deviations from the nominal value. Since the computation requires physical information about the locations of the gates and interconnects, all cells in the circuit were first placed using the placement tool, Capo [15]. Global routing was then performed to route all the nets in the circuits. Depending on the size of circuit, we divided the chip area into different sizes of grids, so that each grid contains  $< 100$  cells.

To verify the results of our method *MinnSSTA*, we used Monte Carlo (MC) simulation for comparison. To balance the accuracy and run time, we chose to run 10,000 iterations for the Monte Carlo simulation. A comparison of these results with those from *MinnSSTA* is shown in Table 2. For each test case, the mean and standard deviation (SD) values for both methods are listed. The results of *MinnSSTA* can be seen to be very close to the MC results: the average error is 0.2% for the mean value and 0.9% for the standard deviation. In Figure 2, for the largest test case *s38417*, we show the plots of the PDF and CDF of the circuit delay for both *MinnSSTA* and MC methods. It is observed that the curves almost perfectly match each other. In Table 2, we also provide the CPU times for both methods. To show the PCA steps require very little run time, the run time for this part is also listed. We can see that the CPU time of *MinnSSTA* on all test cases is very fast. The circuit with the longest run time, *s35932*, was analyzed in only 182 seconds, while the MC simulation required over 13 hours.

To show the importance of considering spatial correlations, we consider the difference between performing statistical timing analysis while considering spatial correlation and while ignoring it. Since this is a comparison to determine why spatial correlations are important, the CPU time is not a consideration. Therefore, we run another set of Monte Carlo simulations (*MCNoCorr*) on the same set of benchmarks, this time assuming zero correlations among the devices and wires on the chip. The comparison between

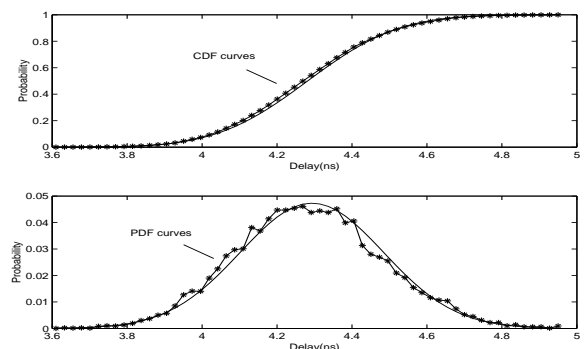


Figure 2: Comparison of *MinnSSTA* and MC methods. *MinnSSTA*: solid-line curve. MC: starred-line curve

Table 2: Comparison results of the proposed method and Monte-Carlo simulation method

Benchmark			Monte-Carlo simulation (MC)			MinnSSTA				$\frac{(MinnSSTA-MC)}{MC} \%$	
Name	#Cells	#Grids	Mean(ps)	SD(ps)	CPU-time(s)	Mean(ps)	SD(ps)	CPU-time(s)	PCA-time(s)	Mean	SD
s38417	23815	256	4285.6	192.7	15295	4281.7	191.4	130.32	0.15	-0.1%	-0.7%
s38584	20705	256	7678.7	271.3	19024	7668.4	268.8	132.08	0.15	-0.1%	-0.9%
s35932	17793	256	5655.9	228.6	48087	5641.1	220.3	182.31	0.15	-0.3%	-3.6%
s15850	10369	256	6441.0	248.1	9932	6430.6	247.2	56.00	0.15	-0.2%	-0.3%
s13207	8260	256	5258.5	206.4	5082	5254.4	208.0	50.48	0.15	-0.1%	0.8%
s9234	5825	64	2870.4	112.5	2952	2868.7	112.9	9.42	0.02	-0.1%	0.4%
s5378	2958	64	2369.0	83.7	1531	2362.8	83.2	5.27	0.02	-0.3%	-0.5%
s1196	547	16	2326.8	94.2	378	2320.9	93.9	0.41	0.01	-0.3%	-0.3%
s27	13	4	944.2	46.9	67	943.3	46.8	0.00	0.00	-0.1%	-0.3%

Table 3: Comparison of timing analysis with and without spatial correlations

Benchmark	Anal. w/ corr. (MC)		Anal. w/o corr. (MCNoCorr)		$\frac{(MC-MCNoCorr)}{MCNoCorr} \%$		Multi-Process-Corner (MPC)		$\frac{(MPC-MC)}{MC} \%$	
	Mean(ps)	SD(ps)	Mean(ps)	SD(ps)	Mean	SD	Mean(ps)	SD(ps)	Mean	SD
s38417	4285.6	192.7	4285.8	100.5	0.0%	91.7%	4324.4	428.7	0.9%	122.5%
s38584	7678.7	271.3	7660.4	178.8	0.2%	51.7%	7683.0	764.6	0.1%	181.8%
s35932	5655.9	228.6	5670.8	188.8	-0.3%	21.1%	5591.5	465.6	-1.1%	103.6%
s15850	6441.0	248.1	6445.2	78.3	-0.1%	216.8%	6508.5	625.7	1.0%	152.2%
s13207	5258.5	206.4	5264.7	73.5	-0.1%	180.9%	5293.9	526.2	0.7%	154.9%
s9234	2870.4	112.5	2881.8	41.7	-0.4%	169.6%	2881.9	280.5	0.4%	149.3%
s5378	2369.0	83.7	2371.2	33.3	-0.1%	151.3%	2349.5	232.9	-0.8%	178.3%
s1196	2326.8	94.2	2340.2	40.8	-0.6%	130.6%	2340.9	225.3	0.6%	139.3%
s27	944.2	46.9	945.2	36.7	-0.1%	27.8%	951.6	95.2	0.8%	102.8%

the data is shown in Table 3. It can be observed that although the mean values are close, the variances of the uncorrelated cases (*MCNoCorr*) are much smaller than the correlated cases (*MC*). On average, the standard deviation of the correlated case increases by 115.7%. Again, we plot the PDF and CDF curves of both simulations for circuit *s38417* in Figure 3. It is seen that the CDF and PDF curves of *MCNoCorr* deviate significantly from those of *MC*. In other words, statistical timing analysis without considering correlation may incorrectly predict the real performance of the circuit and could even overestimate the performance of the circuit. This underlines the importance of developing efficient statistical STA methods that can incorporate spatial correlations.

As an alternative, we consider the option of using multiple process corners (*MPC*) for these experiments, and these results are also displayed in Table 3. On average, such an approach overestimates the standard deviation by 142.8%. These results also emphasize the importance of considering spatial correlations during statistical STA, as is done by our algorithm.

## 7. REFERENCES

- [1] M. Berkelaar, "Statistical Delay Calculation, a Linear Time Method," in *Proc. TAU*, pp. 15-24, 1997.
- [2] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Statistical Timing Analysis Using Bounds and Selective Enumeration," in *Proc. TAU*, pp. 29-36, 2002.
- [3] J. J. Liou, K. T. Cheng, S. Kundu, and A. Krstic, "Fast Statistical Timing Analysis By Probabilistic Event Propagation," in *Proc. DAC*, pp. 661-666, 2001.
- [4] S. Naidu, "Timing Yield Calculation Using an Impulse-Train Approach," in *Proc. Int. Conf. on VLSI Design*, Bangalore, pp. 219-224, 2002.
- [5] S. Tsukiyama, M. Tanaka, and M. Fukui, "A New Statistical Static Timing Analyzer Considering Correlation Between Delays," in *Proc. TAU*, pp. 27-33, Dec 2000.
- [6] J. J. Liou, A. Krstic, L. C. Wang, and K. T. Cheng, "False-Path-Aware Statistical Timing Analysis and Efficient Path Selection for Delay Testing and Timing Validation," in *Proc. ICCAD*, pp. 556-569, 2002.
- [7] M. Orshansky and K. Keutzer, "A General Probabilistic Framework for Worst Case Timing Analysis," in *Proc. DAC*, pp. 556-561, 2002.
- [8] B. Choi and D. M. H. Walker, "Timing Analysis of Combinational Circuits Including Capacitive Coupling and Statistical Process Variation," in *Proc. VLSI Test Symp.*, pp. 49-54, 2000.
- [9] A. Agarwal, D. Blaauw, S. Sundareswaran, V. Zolotov, M. Zhou, K. Gala, and R. Panda, "Path-Based Statistical Timing Analysis Considering Inter- and Intra-die Correlations," in *Proc. TAU*, pp. 16-21, 2002.
- [10] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas, "Impact of Interconnect Variations on the Clock Skew of a Gigahertz Microprocessor," in *Proc. DAC*, pp. 168-171, 2000.
- [11] D. F. Morrison, "Multivariate Statistical Methods," New York: McGraw-Hill, 1976.
- [12] C. E. Clark, "The Greatest of a Finite Set of Random Variables," *Operations Research*, vol. 9, pp. 85-91, 1961.
- [13] J. Cong, "Challenges and Opportunities for Design Innovations in Nanometer Technologies," *SRC Design Science Concept Paper*, 1997.
- [14] S. Nassif, "Delay Variability: Sources, Impact and Trends," in *Proc. ISSCC*, pp. 368-369, 2000.
- [15] A. Caldwell, A. B. Kahng, and I. Markov, "Capo: a large-scale fixed-die placer," avail. at <http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement>.

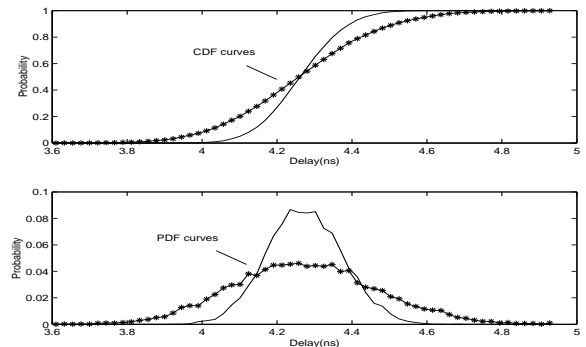


Figure 3: Comparison of statistical STA with and without considering spatial correlations. *MCNoCorr*: solid-line curve. *MC*: starred-line curve (same as in Figure 2)