On Statistical Timing Analysis with Inter- and Intra-die Variations

Hratch Mangassarian
ECE Department, University of Waterloo
Waterloo, ON N2L 3G1, Canada
hrm@vlsi.uwaterloo.ca

Mohab Anis
ECE Department, University of Waterloo
Waterloo, ON N2L 3G1, Canada
manis@vlsi.uwaterloo.ca

Abstract

In this paper, we highlight a fast, effective and practical statistical approach that deals with inter and intra-die variations in VLSI chips. Our methodology is applied to a number of random variables while accounting for spatial correlations. Our methodology sorts the Probability Density Functions (PDFs) of the critical paths of a circuit based on a confidence-point. We show the mathematical accuracy of our method as well as implement a typical program to test it on various benchmarks. We find that worst-case analysis overestimates path delays by more than 50% and that a path's probabilistic rank with respect to delay is very different from its deterministic rank.

1 Introduction

Variability is posing an increasing challenge to timing analysis of VLSI designs implemented in the nanometer regime. Variations can be either *environmental* or *physical*. Environmental variations are due to unpredictable operating conditions such as the power supply voltage V_{dd} and the chip's temperature, whereas physical variations arise during the fabrication of the chip and include permanent discrepancies in the oxide thickness t_{ox}, effective channel length L_{eff} and interconnect dimensions. These variations appear at different levels of the manufacturing process: *inter-lot*, *inter-wafer*, *inter-die* and *intra-die*. Traditionally, discrepancies in VLSI chip parameters have been accounted for using several cases (best-case, nominal and worst-case) in *Static Timing Analysis*. Worst-case analysis is a deterministic timing analysis that was a fair assumption to make until a few years ago, when intra-die parameter mismatches were considered to be negligible in the presence of the more significant inter-die variations. Today, intra-die variations must be accounted for, as worst-case timing analysis is highly potential-cutting [1].

Physical parameters are susceptible to random variations from their nominal values, and therefore should be modeled as Random Variables RVs. If the circuit to be analyzed contains thousands of gates, and because of the existing spatial correlations between RVs of the same kind, thousand-variable Joint PDFs (JPDFs) should be used for each RV kind. Getting all these joint distributions is computationally impossible for some reasonable discretization accuracy, let alone the exponential run-time needed for computing any path PDF using such JPDFs. If the number of discretization points of the JPDFs per dimension is equal to QUALITY, then an N-variable JPDF will have $\mathsf{QUALITY}^N$ discretization points, making the computational complexity for computing the delay PDF of that path $\bigcirc(\mathsf{QUALITY}^N)$. Hence, the exact numerical solution for finding the PDF of the critical path is unrealistically complex.

Recently, a number of techniques have been developed to perform statistical timing analysis. They can be broadly classified into *full-chip* analysis and *path-based* analysis. Full-chip analysis maps the whole block to an acyclic graph and strives to propagate and merge the PDFs of the gate delays, in order to get the PDF of the critical path of the circuit [2-9]. This method has an inherently exponential run-time with circuit size. Therefore, in the pursuit of reducing the run-time, advocators of full-chip analysis have to use primitive delay models, assume that gate delay PDFs are given [3,4], neglect parameter correlations [2,3,8] and draw on inaccurate approximations [7]. And even with this, they sometimes give *bounds* for the delay PDF and not the PDF itself [2,8,9]. In the path-based approach, the PDF of one path is calculated at a time [10-11]. First, Deterministic Timing Analysis is performed on the circuit in order to find some upper percentage of nominally critical paths. Then, each *candidate* critical path is analyzed statistically, with the aim of computing its delay PDF. The *probabilistic critical path* is decided by comparing some confidence-point on the PDFs, such as the 3σ points. This method allows for more complex delay and interconnect models [11]. Some common assumptions and drawbacks, both in full-chip and path-based statistical analysis, include the consideration of only 1 RV (usually effective channel length) [9,10], the use of 1 correlation-space at most (usually spatial correlations) [9,10], the restriction to a certain kind of input PDF (usually Gaussian) [10,12], and the exponential run-time needed to get accurate results [2,9,12].

In this work, we propose a path-based methodology which relaxes some these assumptions while maintaining a run-time of polynomial complexity. Our methodology includes *all the steps* that lead to finding the probabilistic critical path, from the initial deterministic analysis algorithms to the ranking of all probabilistic critical paths of the circuit. The methodology accounts for several RVs which affect gate delay variations. Spatial correlations for these RVs are taken into consideration. Our findings emphasize the critical path delay overestimation that arises due to worst-case analysis.

2 Modeling and Assumptions

2.1 Gate Delay Using Elmore's Model

In order to perform probabilistic timing analysis later in the paper, we first derive the delay of any path which is composed of gates by using Elmore's model. Based on Elmore's delay model, [13] showed that the propagation delay of an n-input NAND gate t_p is formalized as

$$t_p = 0.345[R_N C_{dN} F_I (F_I - 1) + F_I R_N C_n + R_P C_n] \quad (1)$$

1530-1591/05 $20.00 © 2005 IEEE

where F_I is the number of fan-ins, C_{dN} is the drain capacitance, C_n is the sum of the drain capacitances of this gate at the output node, plus the gate capacitances of all fan-out gates, as well as the wire capacitance, while R_N and R_P are the NMOS and PMOS on-resistances, respectively. For $short\ channel\ models$ for R_N and R_P, t_p can be expressed as

$$t_p = 0.345\frac{t_{ox}L_{eff}}{\epsilon_{ox}}[\alpha(\frac{V_{dd}}{(V_{dd}-V_{Tn})^{1.3}}+\frac{1}{1.5V_{dd}-2V_{Tn}}) + \beta(\frac{V_{dd}}{(V_{dd}-|V_{Tp}|)^{1.3}}+\frac{1}{1.5V_{dd}-2|V_{Tp}|})] \quad (2)$$

where

$$\alpha = \frac{C_{dN}F_I(F_I-1)+F_IC_n}{\mu_nW_n} \quad (3)$$

$$\beta = \frac{C_n}{\mu_pW_p} \quad (4)$$

where ϵ_{ox} is the oxide permittivity, W_n/W_p are the NMOS/PMOS channel widths, μ_n/μ_p are the NMOS/PMOS mobilities and V_{Tn}/V_{Tp} are the NMOS/PMOS threshold voltages. Similar delay equations can be found for an inverter, an n-input NOR gate and a 2-input XNOR gate. They all have the *same form* as (2) with different values for α and β. These are the gates under investigation which constitute all ISCAS85 benchmarks. Thus, a path consisting of N such gates will have a delay equation of the form

$$t_{PATH} = \sum_{i=1}^{N}\{0.345\frac{t_{ox_i}L_{eff_i}}{\epsilon_{ox}}[\alpha_i(\frac{V_{dd_i}}{(V_{dd_i}-V_{Tn_i})^{1.3}}+\frac{1}{1.5V_{dd_i}-2V_{Tn_i}}) + \beta_i(\frac{V_{dd_i}}{(V_{dd_i}-|V_{Tp_i}|)^{1.3}}+\frac{1}{1.5V_{dd_i}-2|V_{Tp_i}|})]\} \quad (5)$$

2.2 Sensitivity Analysis

In order to investigate how much each parameter's variability affects the delay, we performed a first-degree sensitivity analysis for the delay of a 2-input NAND gate, an inverter, a 2-input NOR gate and a 2-input XNOR gate, all with fan-outs of 2, at the nominal values of all parameters. To avoid complicated calculations, and to focus the reader's attention to the approach, we assumed the parameters to be independent and all capacitances to be constant. We compared the values of $\left|\frac{\partial t_p}{\partial x_i}|_{\mathbf{X}_{nom}}\cdot\sigma_{x_i}\right|$ for each parameter x_i, where $\mathbf{X}$ is the vector of parameters and σ_{x_i} is the standard deviation of x_i. $130nm$ CMOS technology values were used, and the typical variances were obtained from [15]. The parameters that had the most impact on the delay were t_{ox}, L_{eff}, V_{dd}, while V_{Tn} and V_{Tp} had less effect. Table 1 shows the linear approximations of gate delay variations as a result of varying those 5 RVs by one σ_{x_i}.

2.3 Layering of Correlation-Spaces

In this section, we show a simple yet efficient method developed in [9],[10] that divides the spatial correlation-space into several levels. We use this technique to get rid of the correlated RVs while keeping the correlation information. The model first replicates the die on several layers, then divides each layer i of the die into 4^i rectangle regions. Using variabilities in L_{eff} for each layer's partition-size, the model replaces the L_{eff} of each gate by a sum of RVs, such

	2-NAND	2-NOR	INV	2-XNOR
t_{ox}	0.587ps	0.369ps	0.225ps	0.529ps
L_{eff}	2.061ps	1.296ps	0.792ps	1.859ps
V_{dd}	0.360ps	0.227ps	0.136ps	0.324ps
V_{Tn}	0.071ps	0.046ps	0.030ps	0.070ps
$\|V_{Tp}\|$	0.088ps	0.025ps	0.078ps	0.066ps

Table 1. Sensitivity Analysis of the Elmore Delay Model ($\left|\frac{\partial t_p}{\partial x_i}|_{\mathbf{X}_{nom}}\cdot\sigma_{x_i}\right|$ for each x_i) ($\sigma_{t_{ox}}$=0.15nm,$\sigma_{L_{eff}}$=15nm,$\sigma_{V_{dd}}$=40mV,$\sigma_{V_{Tn}}$=13mV,$\sigma_{V_{Tp}}$=14mV)

that the correlation information between two L_{eff}'s is in number of common RVs they have and their variances. This method was used only for spatial correlations in L_{eff}. We include other RVs to the technique proposed in [9], to account for any other chip variations.

Since we are accounting for more than 1 RV, we will first provide a general formulation to the spatial correlation problem. Let χ refer to a parameter (RV) with a known marginal probability distribution, χ_i refers to a RV of type χ of any partition in a certain level i, and $\chi_{i,j}$ refers to the RV of type χ and of partition j in level i. χ_0 is assigned a PDF whose mean is $\chi_{nominal}$. *Each* subsequent χ_i is assigned another PDF, with a mean of $zero$, such that the PDF of $\sum_{i=0}^{L-1}\chi_i$ is equal to the PDF of χ, which is given. If χ has a Gaussian PDF, this simplifies to assigning a Gaussian PDF to each χ_i, such that

$$\sigma_\chi^2 = \sum_{i=0}^{L-1}\sigma_{\chi_i}^2 \quad (6)$$

where L is the number of hierarchical layers, and layer 0 represents the whole correlation space. The RV of a certain gate, χ_{gate} for instance, will be set to the sum over all layers of the partition RVs that it belongs to, yielding

$$\chi_{gate} = \sum_{i=0}^{L-1}\sum_{j=0}^{4^i-1}\xi(i,j,gate)\chi_{i,j} \quad (7)$$

where $\xi = 1$ *only if* j *is the partition the* $gate$ *belongs to* on level i, and $\xi = 0$ otherwise. It is the inter-die variations in some χ that decide the chip-mean of χ. The remaining layers correspond to different levels of intra-die variations. Intra-die variations are only deviations *around* the chip-mean. This is why all layers except layer 0 were assigned zero-mean PDFs. As for the mean of the inter-PDF of some χ (the mean of the PDF of χ_0), it is equal to $\chi_{nominal}$, because the average of the means of χ, in all chips of all wafers of all lots, is $\chi_{nominal}$.

Using the above spatial correlation description, the path delay (5) can be re-written as

$$t_{PATH} = \sum_{i=1}^{N}\{0.345\frac{\sum_{i,u,w}t_{ox_{u,w}}\sum_{i,u,w}L_{eff_{u,w}}}{\epsilon_{ox}}\times [\alpha_i(\frac{\sum_{i,u,w}V_{dd_{u,w}}}{(\sum_{i,u,w}V_{dd_{u,w}}-\sum_{i,u,w}V_{Tn_{u,w}})^{1.3}}+ \frac{1}{1.5\sum_{i,u,w}V_{dd_{u,w}}-2\sum_{i,u,w}V_{Tn_{u,w}}}) + \beta_i(\frac{\sum_{i,u,w}V_{dd_{u,w}}}{(\sum_{i,u,w}V_{dd_{u,w}}-\sum_{i,u,w}|V_{Tp_{u,w}}|)^{1.3}}+ \frac{1}{1.5\sum_{i,u,w}V_{dd_{u,w}}-2\sum_{i,u,w}|V_{Tp_{u,w}}|})]\} \quad (8)$$

where u and w represent the layer number and the partition number, respectively. Even though the RVs in the equation of a path are considered as independent, we still cannot use this to our advantage in order to find the PDF of any t_{PATH}. This is because the same

RV, for instance, say, $L_{eff_{1,2}}$, can belong to the t_p of many gates in the same path. Therefore, we cannot separately calculate the delay PDF of each t_{p_i} along a path, since the same RVs may re-occur in the path, causing correlation between t_{p_i}'s. In the following section, we will use the Taylor series first-order approximation for the gate delay, in order to linearize part of t_{p_i}, and therefore transform (8) into a suitable form for PDF calculation in low run-time complexity.

2.4 Taylor Series First-Order Approximation for Path Delay

Let $\mathbf{X}_i$ represent the vector of RVs of gate i, $\mathbf{X}_{inter}$ the vector of inter-RVs, $\Delta\mathbf{X}_{intra_i}$ the vector of intra-RVs of gate i, and for any gate i: $\mathbf{X}_i = \mathbf{X}_{inter} + \Delta\mathbf{X}_{intra_i}$. Using the Taylor-series first order approximation of t_{p_i} taken at the bias point $\mathbf{X}_{inter}$, the gate delay equation is formulated in vector and scalar forms as follows

$$t_{p_i}(\mathbf{X}_{inter} + \Delta\mathbf{X}_{intra_i}) \approx t_{p_i}(\mathbf{X}_{inter}) + \nabla t_{p_i}|_{\mathbf{X}_{inter}} \cdot \Delta\mathbf{X}_{intra_i} \quad (9)$$

It should be noted that the mean of $\mathbf{X}_{inter}$ is $\mathbf{X}_{nominal}$, the mean of $\Delta\mathbf{X}_{intra_i}$ is zero, and that this approximation is accurate if $\Delta\mathbf{X}_{intra_i} \ll \mathbf{X}_{inter}$. Since the intra-die RV standard deviations are much smaller than the nominal values of those RVs, the approximation is valid. Replacing the intra and inter RVs of (9) by the layer RVs described in the previous subsection, we will get

$$t_{p_i}(\mathbf{X}_{inter} + \Delta\mathbf{X}_{intra_i}) \approx t_{p_i}(\mathbf{X}_{0,0}) + \nabla t_{p_i}|_{\mathbf{X}_{0,0}} \cdot \sum_{u,w} \mathbf{X}_{u,w} \quad (10)$$

where $\mathbf{X}_{0,0}$ is $\mathbf{X}_{inter}$.

All summation RVs are assumed to be independent, both inside each sum and across sums. This is very helpful because, *if their coefficients were constants*, we would have an intra-part that is simply a *linear combination of independent RVs*. Finding the intra-delay PDF would then be very easy. However, their coefficients are the partial derivatives of delay *evaluated at $\mathbf{X}_{inter}$, which is a random vector*. Hence, we will take a final zeroth-order approximation that evaluates the partial derivatives at nominal value [10], and thus making them constants:

$$\frac{\partial t_{p_i}}{\partial \chi}|_{\mathbf{X}_{inter}} \approx \frac{\partial t_{p_i}}{\partial \chi}|_{\mathbf{X}_{nominal}} \quad (11)$$

yielding

$$t_{p_i}(\mathbf{X}_i) \approx t_{p_i}(\mathbf{X}_{0,0}) + a_i \sum_{u,w} t_{ox_{u,w}} + b_i \sum_{u,w} L_{eff_{u,w}} + c_i \sum_{u,w} V_{dd_{u,w}} + d_i \sum_{u,w} V_{Tn_{u,w}} + e_i \sum_{u,w} |V_{Tp_{u,w}}| \quad (12)$$

where a_i, b_i, c_i, d_i and e_i are constants equal to the derivatives of the delay of gate i at nominal value. Thus, the path-delay equation in (8) changes when we use this approximation. Using (12), for a path of N gates, (8) yields

$$t_{PATH} = \sum_{i=1}^{N} t_{p_i}(t_{ox_{0,0}}, L_{eff_{0,0}}, V_{dd_{0,0}}, V_{Tn_{0,0}}, |V_{Tp_{0,0}}|) + \sum_{u,w} a_{u,w} t_{ox_{u,w}} + \sum_{u,w} b_{u,w} L_{eff_{u,w}} + \sum_{u,w} c_{u,w} V_{dd_{u,w}} + \sum_{u,w} d_{u,w} V_{Tn_{u,w}} + \sum_{u,w} e_{u,w} |V_{Tp_{u,w}}| \quad (13)$$

where $a_{u,w}$, $b_{u,w}$, $c_{u,w}$, $d_{u,w}$ and $e_{u,w}$ are constants corresponding to layer u and partition w. The first summation is the $t_{PATH_{inter}}$, while the following terms are the $t_{PATH_{intra}}$. Therefore, the intra-delay of a path is simply a linear combination of RVs. If the input RVs were Gaussian, then finding the intra-PDF would be equivalent to finding its variance

$$\sigma^2_{t_{PATH_{intra}}} = \sum_{u,w} a^2_{u,w} \sigma^2_{t_{ox_{u,w}}} + \sum_{u,w} b^2_{u,w} \sigma^2_{L_{eff_{u,w}}} + \sum_{u,w} c^2_{u,w} \sigma^2_{V_{dd_{u,w}}} + \sum_{u,w} d^2_{u,w} \sigma^2_{V_{Tn_{u,w}}} + \sum_{u,w} e^2_{u,w} \sigma^2_{|V_{Tp_{u,w}}|} \quad (14)$$

2.5 Convexity Analysis

For the delay equation for the gates used, we calculated the values of $\left|\frac{\partial^2 t_p}{\partial {x_i}^2}|_{\mathbf{x}_{nom}} \cdot \sigma_{x_i}\right|$ for each parameter x_i. This represents the change of the derivative of delay with respect to that parameter for a one standard deviation change of the parameter. The first derivative of gate delay with respect to those RVs is in the order of tens of pico-seconds per Volts, which is significantly higher that the values of $\left|\frac{\partial^2 t_p}{\partial {x_i}^2}|_{\mathbf{x}_{nom}} \cdot \sigma_{x_i}\right|$ for all parameter x_i. Therefore, we can conclude that even a worst case *change in the derivative*, for a 3σ change in those RVs, would still be an order of magnitude less than the actual value of the derivative. Therefore, the convexity is small enough to ensure an acceptable accuracy for this approximation.

Doing the numerical computation of the inter-PDF at once would require a complexity of $\bigcirc(\mathsf{QUALITYinter}^R)$, where QUALITYinter is the discretization of the PDFs of the inter-RVs and does not have to be equal to QUALITYintra (the discretization of the PDFs of the intra-RVs), and R is the number of different parameters being varied. One should try to separate as many variables, in order to reduce the complexity of the PDF-computation. It should be noted however, that with more complex models, inter-delay equation will tend to become inseparable in almost all its parameters. In such a case, one should identify the parameters with the *least* inter-variabilities, and when using the Taylor-series approximation, instead of using their inter-RVs, one should use their nominal values. Another possible compromise of accuracy for faster run-time could be to reduce QUALITYinter.

In the next section, we explain our full methodology, which makes use of the two assumptions analyzed in this section, in order to find and rank some upper percentage of probabilistic critical paths.

3 Methodology

A high-level description of our methodology flowchart is the following. First, we set the number of RVs (R), the number of layers (L) for the spatial correlation-space, as well as the variabilities of each layer in each RV. The circuit is then mapped to a *timing graph*, and we evaluate all gate deterministic delays as well as derivatives with respect to all RVs that are being considered, at their nominal values. The inputs to the methodology are a description of the circuit connections, gates, inputs, outputs as well as any necessary information related to correlations, like (x, y) coordinates to compute the spatial correlations. These are one time calculations. Next, the *deterministic* critical path is found, using Bellman-Ford. We perform the probabilistic timing analysis of the deterministic critical path delay: We find its intra-delay PDF and its inter-PDF, using the techniques described in the previous section, and finally its total delay PDF (accounting for inter and intra variations). From its delay PDF, its standard deviation σ_C is extracted in order to have an idea about the variability of our circuit. Using that standard deviation and some arbitrary *confidence* constant C that the user inputs, we find all the next deterministic critical paths that are within $C \cdot \sigma_C$ of the first

deterministic critical path delay. We perform the same probabilistic timing analysis to each of those paths sequentially. In the end, we can compare any confidence point, such as the 3σ's, on the critical path PDFs, in order to rank them and obtain the *probabilistic critical path*. Fig. 1 shows the graphical flowchart of our methodology.

3.1 Deterministic Delay Computation

The Bellman-Ford algorithm was used to find the deterministic critical path of a circuit. The weight of each edge in the algorithm is to be the nominal delay of the node before the edge, since the graph is simple, directed and acyclic. The worst-case complexity of Bellman-Ford is $\bigcirc(|N| \times |E|)$, where $|N|$ is the number of nodes in the graph and $|E|$ is the number of edges. This can be quite big if the graph is very dense, however, the fact that our graph is simple and acyclic makes it highly unlikely to reach that worst-case.

3.2 Probabilistic Timing Analysis to Deterministic Critical Path

Probabilistic timing analysis is applied to the deterministic critical path to find its standard deviation. The probabilistic analysis is separated into intra- and inter-delay calculations. The variance of the intra-die variations are formulated using (14), and the PDF of the intra-delay is computed (assuming it's Gaussian). The complexity of such a PDF computation is simply $\bigcirc(\mathsf{QUALITYintra})$. For the inter-delay PDF calculation, the first term in (13) is used. Finally, the intra- and inter- PDFs are convolved to evaluate the total delay PDF. Supposing that both PDFs have a discretization of QUALITY, the complexity of the convolution is obviously $\bigcirc(\mathsf{QUALITY}^2)$.

We use the standard deviation (σ_C) of the deterministic critical path total delay PDF as an indicator of variability in the circuit. We will choose and analyze all the paths whose delays are larger than $D - C \cdot \sigma_C$, where D is the deterministic delay of the critical path and C is a constant the user specifies. The larger this constant, the more confident we get in that there is no other path in the circuit that is probabilistically longer than the one we got, but the more run-time is needed. In order to find all the next critical paths within $C\sigma_C$ of the deterministic critical path, we used the recursive algorithm shown in Fig. 2, where W_i is the weight of edge i and LABEL_{n_i} is the delay label of node n_i, equal to the maximum arrival time to n_i from n_s, that was calculated using Bellman-Ford. In broad terms, the algorithm starts from a root node, which is n_f, and checks for

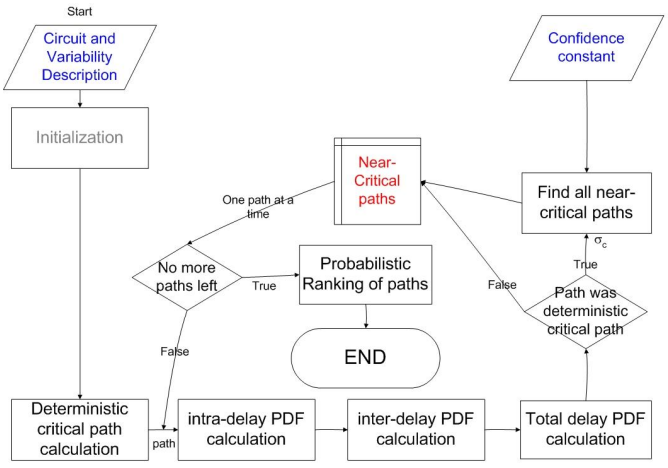


Figure 1. Methodology Flowchart

1. **Initialization**
   $\mathsf{ROOT} = n_f$
   $\mathsf{LABEL}_{ROOT} = D - C\sigma_C$
   $\mathsf{FIRST}_{ROOT} = true$
2. **Find Near-Critical Paths** (ROOT, LABEL_{ROOT},FIRST_{ROOT})
   For all fan-in nodes n_i of ROOT
   If ($\mathsf{LABEL}_{n_i} >= \mathsf{LABEL}_{ROOT} - W_i$)
   If (FIRST_{ROOT} = true)
   Add n_i to the path
   Update delay and node count of path
   Find Near-Critical Paths(n_i, $\mathsf{LABEL}_{ROOT} - W_i, true$)
   Else
   Create new path
   Find all previous nodes to ROOT in previous path
   Add previous nodes, ROOT and n_i to new path
   Set delay and node count of new path
   Find Near-Critical Paths(n_i, $\mathsf{LABEL}_{ROOT} - W_i, true$)
   End If
   End If
   End For

Figure 2. Finding near-critical paths

fan-in nodes that have labels larger than the label of the root minus the weight of the edge between them. If there is such a node, than the same thing is repeated for that node as the root. This is repeated until n_s is reached for each path. The worst-case complexity of this algorithm is $\bigcirc(\kappa \times |E|)$, where κ is the number of near-critical paths and $|E|$ is the number of edges in the graph.

Finally, probabilistic analysis is done for all the near-critical paths, one path at a time, to find the delay PDF of each. In the end, we rank the paths based on their PDFs by some confidence point. We determine the probabilistic critical path and visualize the change in rank of paths, going from deterministic to statistical analysis. It is important to note that the means of the delay PDFs of the paths are **not** the same as their deterministic delays, because the inter-delay is not a linear function of RVs and therefore *the expected value of the delay, is not the delay of the expected values.* The intra-delay calculations for all the paths yield a complexity of $\bigcirc(\kappa \times \Omega \times \mathsf{QUALITYintra}^2)$, where Ω is the number of layer-RVs in the path. However, the bottleneck complexity of the methodology comes from the inter-delay PDFs, which have a worst-case complexity of $\bigcirc(\kappa \times \mathsf{QUALITYinter}^R)$.

4 Results and Discussion

We implemented a program that reads the circuit-description as a Design Exchange Format (DEF) file and gets the variability information from the user, in order to generate and rank the delay-PDFs of critical paths. It applies all the steps of the methodology described in the previous section and depicted in Fig. 1. The considered RVs were: t_{ox}, L_{eff}, V_{dd}, V_{Tn} and V_{Tp}. Their PDFs were assumed to be Gaussian, truncated at their 6σ points. Typical standard-deviations were taken from [15]. The (x, y) coordinates of the gates were extracted from the DEF files in order to account for spatial correlations. We used a 4 layer model along with a fifth random layer, and we divided the total variances equally over all layers. We tested our program on the ISCAS85 benchmark circuits, using $130nm$ technology nominal values. The computations were performed using PDF discretizations of $\mathsf{QUALITYintra} = 100$ and $\mathsf{QUALITYinter} = 50$. We shall illustrate how these values were chosen as an optimal trade-off between run-time and solution accuracy. Table 2 summarizes the results.

The confidence constant C provides a mean for controlling the number of near-critical paths to consider. We started with a minimum of $C = 0.05$ (except for $c6288$) and we increased C until we

converged to a critical path that is not changing. In Table 2, we used the minimum value of C that found the correct probabilistic critical path. However, for benchmark $c6288$, even for a $C = 0.005$, the number of near-critical paths was more than a hundred thousand, which is unacceptable both in terms of run-time and memory. So we used $C = 0.001$, which still yielded about 900 paths. Column

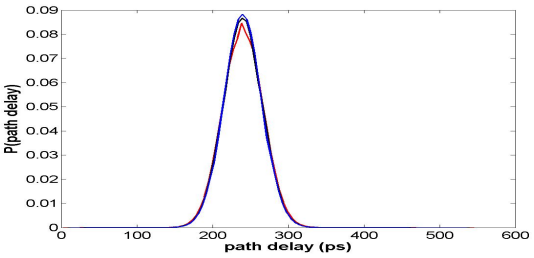


Figure 3. Delay PDFs of the 1^{st}, 798^{th} and 1596^{th} paths in $c1355$

7 shows the number of critical paths for each circuit. This number depends heavily on the choice of the confidence C, because all paths with a delay within $C\sigma_C$ of the nominal critical delay are considered near-critical, where σ_C is the deterministic critical path's standard deviation. In addition, the type of benchmark (structure of the graph), as well as the variability and correlations of parameters, affect σ_C, and therefore the number of critical paths. To see this, one can observe the large difference in critical path numbers, between circuits for which we used the same C, such as benchmarks $c880$ and $c1355$: For $C = 0.05$, the former had 3 near-critical paths, while the latter had 1596. Fig. 3 shows how close the delay PDFs of the 1^{st}, 798^{th} and 1596^{th} paths are in $c1355$. It should be noted that Column 8 shows the mean of the probabilistic critical path PDF which is very close but *not* equal to the nominal critical path delay, because the delay is non-linear. Columns 9 and 5 depict the 3σ points for the critical path delay PDFs and the % overestimation of the worst-case analysis over the 3σ points of the probabilistic critical path delay. On average, an overestimation of 55% is found, which demonstrates how unacceptable and persistent the conservatism resulting from worst-case analysis can get. Fig. 4 shows the critical path's intra- and inter- delay PDFs for $c432$, as well as the convolution of the two, which is the total delay PDF. In addition, the 3-sigma point of the total probabilistic (statistical) PDF compared to the worst-case deterministic analysis.

In order to characterize the different effects of inter- and intra- variations on the delay, we performed probabilistic timing analysis of $c432$, for 3 different scenarios of inter- and intra- variances, for the same total RV variabilities. Table 3 demonstrates the results. One can clearly see that the larger the inter-variability is, the larger will be the path-delay standard-deviation. In fact, large inter-variability increases the possibility that all the gates in a path are subject to worst-case variations. Even the number of near-critical paths increases because of the increase of σ_C.

c432	critical path mean (ps)	total σ (ps)	inter σ (ps)	intra σ (ps)	# of critical paths
Only intra-die variations	265.891	19.950	0	19.950	20
50% inter-die, 50% intra-die	267.074	35.577	32.674	14.076	54
75% inter-die, 25% intra-die	266.889	41.388	39.960	10.778	76

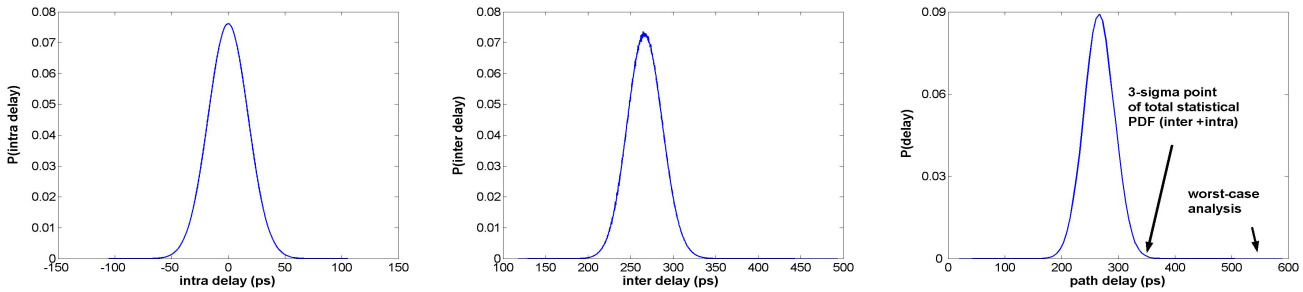
Table 3. inter- and intra- variations

Column 11 in Table 2 demonstrates the new rank of deterministic critical path after applying probabilistic timing analysis. Some paths remained unchanged (such as $c432$, $c880$, $c7552$), while others were nominally faster paths (notably in $c1355$ - what used to be the 40th slowest deterministic path is now the critical path in the probabilistic analysis). This is because of the spatial correlations' impact on $c1355$'s topology, increasing the variability in a path, causing their 3σ delays to become very big. The larger the variances and correlations, the more deterministic and probabilistic ranks will tend to differ from one another. Fig. 5 draws the probabilistic rank of the first 100 paths of benchmark $c1355$ versus their deterministic ranks, when around 1600 near-critical paths were analyzed. Fig. 6 illustrates the same plot for the first 100 paths of benchmark $c7552$ with the same number of analyzed near-critical paths. We observe that the first plot is considerably far from the first bisector, which means that distances between paths are very small for $c1355$ compared to the existing amount of variability. On the other hand, in the plot for $c7552$, we can see that the ranks' changes are minor. This can be explained by the fact that the graph of $c1355$ is more bushy than that of $c7552$, therefore paths are much closer in terms of their delays. Thus, path delays are very vulnerable to change ranks for the $c1355$ case (as depicted in Fig. 5). This is not the case for $c7552$ where the delay of the paths are more distinctive. Moreover, spatial correlations can be accounted for larger delay variances in $c1355$, due to its DEF circuit description. In conclusion, it is the topology and placement of the circuit that usually determine changes in critical path ranks.

Finally, the last column in Table 2 shows the run-times of the

Circuit		**Deterministic analysis**			**Probabilistic analysis**						
name	# of gates	critical path delay (ps)	worst-case delay (ps)	% diff. from 3σ point	C	# of critical paths	critical path mean (ps)	critical path 3σ point (ps)	# of gates	det. rank	run-time (s)
$c432$	160	266.771	545.009	56.61	0.05	32	266.640	347.996	16	1	0.2
$c499$	202	180.004	358.336	49.94	0.05	58	179.183	238.979	11	40	0.6
$c880$	383	205.999	421.535	58.68	0.05	3	206.036	265.655	23	1	$<$0.1
$c1355$	546	241.245	486.283	52.46	0.05	1596	240.180	318.963	24	902	27
$c1908$	880	326.109	675.068	58.07	0.05	5	324.403	427.082	40	5	$<$0.1
$c2670$	1269	375.465	762.627	57.26	0.1	74	373.216	484.960	32	18	1.5
$c3540$	1669	459.501	903.289	48.32	0.05	32	458.431	609.015	41	8	0.5
$c5315$	2307	381.292	775.375	50.69	0.05	5	381.177	514.552	48	1	0.4
$c6288$	2416	1033.433	2163.213	62.22	0.001	896	1033.531	1333.470	124	1	15
$c7552$	3513	383.688	754.628	51.57	0.05	5	383.557	497.886	21	1	0.4

Table 2. Results



(a) Intra-delay PDF (b) Inter-delay PDF (c) Total path delay PDF for $c432$

Figure 4. Inter-, Intra-, and Total PDF for $c432$

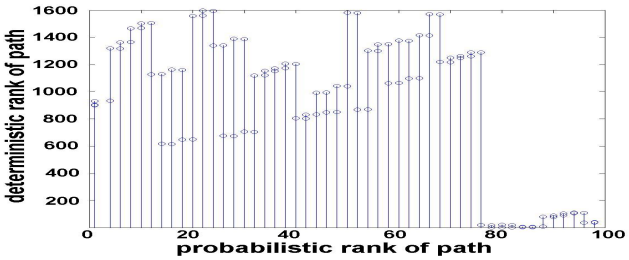


Figure 5. Probabilistic rank vs. Deterministic rank for $c1355$ (Large change in ranks)

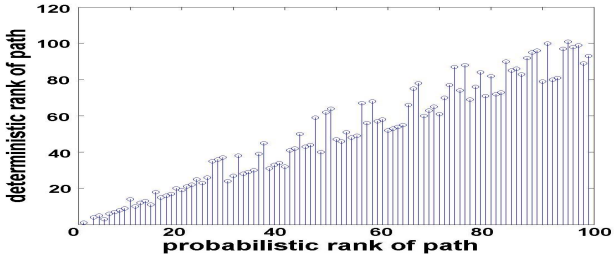


Figure 6. Probabilistic rank vs. Deterministic rank for $c7552$ (Limited change in ranks)

whole program, from the input-file parsing, to finding the deterministic critical path, to computing their delay PDFs and ranking them. For a path-based approach such as ours, the path delay complexity is directly proportional to the number of near-critical paths to consider, hence those run-times are very strong functions of C. Moreover, the run-time also varies with the graph structure: very close deterministic delays will imply a considerable number of near-critical paths. Lastly, run-time is a strong function of the discretization points QUALITYintra and QUALITYinter.

In order to measure the trade-off between accuracy and run-time, as far as QUALITYintra and QUALITYinter are concerned, we calculated the delay PDF of $c499$, using a series of discretization combinations. The most accurate would be the one done with the most discretization points. As an optimal trade-off between accuracy and run-time, we chose the point (QUALITYintra $= 100$, QUALITYinter $= 50$) because it yielded an accuracy within 0.009% of the 3σ point with the highest discretization, with a run-time of 0.4 seconds. Thus, the work presented in this paper uses QUALITYintra $= 100$, QUALITYinter $= 50$.

One last observation involves the typical minimum value of C. Looking at Table 2, we can see that the biggest C we needed, to find the absolute critical path, was 0.1. This means that, for typical circuits, the probabilistic critical path is within 10% of a typical path-delay standard-deviation from the deterministic critical path. This is a vital advantage for path-based statistical analysis, because it means that typically the number of near-critical paths to consider is acceptable.

5 Conclusion

We presented a framework for performing statistical timing analysis. Five random variables have been accounted for in spacial correlations. The work presented has a polynomial run time. Our findings confirmed that the worst-case analysis overestimates path delays by more than 50%. Moreover, depending on the circuit's topology and placement information, a path's probabilistic rank with respect to delay could be very different from its deterministic rank.

References

[1] D. Boning and S. Nassif, Design of High-Performance Microprocessor Circuits, Wiley-IEEE, 2001. Chapter 6

[2] A. Agarwal et al., "Statistical Timing Analysis Using Bounds and Selective Enumeration," *IEEE Trans. CAD*, Sept. 2003, pp. 1243 - 1260.

[3] S. Devadas et al., "Statistical Timing Analysis of Combinational Circuits," *ICCD* 1992, pp. 38 - 43.

[4] H. Jyu et al., "Statistical Timing Optimization of Combinational Logic Circuits," *ICCD* 1993, pp. 77 - 80.

[5] F. Najm et al., "Statistical Timing Analysis Based on a Timing Yield Model," *DAC* 2004, pp. 460 - 465.

[6] S. Tongsima et al., "Optimizing Circuits with Confidence Probability Using Probabilistic Timing," *ISCAS* 1998, pp. 270 - 273.

[7] L. Jing-Jia et al., "Fast Statistical Timing Analysis by Probabilistic Event Propagation," *DAC* 2001, pp. 661 - 666.

[8] A. Agarwal et al., "Computation and Refinement of Statistical Bounds on Circuit Delay," *DAC* 2003, pp. 348 - 353.

[9] A. Agarwal et al., "Statistical Timing Analysis for Intra-die Process Variations with Spacial Correlations," *ICCAD* 2003, pp. 900 - 907.

[10] A. Agarwal et al., "Statistical Delay Computation Considering Spacial Correlations," *ASP-DAC* 2003, pp. 271 - 276.

[11] A. Gattiker et al., "Timing Yield Estimation from Static Timing Analysis," *ISQED* 2001, pp. 437 - 442.

[12] M. Orshansky et al., "A General Probabilistic Framework for Worst-Case Timing Analysis," *DAC* 2002, pp. 556 - 569.

[13] L. Wei et al., "Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications," *IEEE Trans. VLSI*, March 1999, pp. 16 - 24.

[14] J. Rabaey, Digital Integrated Circuits. Englewood Cliffs, Prentice-Hall, 1996.

[15] S. Nassif, "Delay Variability: Sources, Impacts and Trends," *ISSCC* 2000, pp. 368 - 369.