

Fast Statistical Timing Analysis Handling Arbitrary Delay Correlations

Michael Orshansky and Arnab Bandyopadhyay

Department of Electrical and Computer Engineering

The University of Texas at Austin

1 ABSTRACT

An efficient statistical timing analysis algorithm that can handle arbitrary (spatial and structural) causes of delay correlation is described. The algorithm derives the entire cumulative distribution function of the circuit delay using a new mathematical formulation. Spatial as well as structural correlations between gate and wire delays can be taken into account. The algorithm can handle node delays described by non-Gaussian distributions. Because the analytical computation of an exact cumulative distribution function for a probabilistic graph with arbitrary distributions is infeasible, we find tight upper and lower bounds on the true cumulative distribution. An efficient algorithm to compute the bounds is based on a PERT-like single traversal of the sub-graph containing the set of N deterministically longest paths. The efficiency and accuracy of the algorithm is demonstrated on a set of ISCAS'85 benchmarks. Across all the benchmarks, the average *rms* error between the exact distribution and lower bound is 0.7%, and the average maximum error at 95th percentile is 0.6%. The computation of bounds for the largest benchmark takes 39 seconds.

Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance analysis

General Terms

Algorithms, performance, reliability

2 INTRODUCTION

Statistical timing analysis is needed because the traditionally formulated worst-case timing analysis is no longer sufficient for timing verification of advanced chips. Multiple approaches have been proposed over the years to deal with the statistical timing problem. These methods rely either on the performance-space Monte-Carlo STA [1,2], the parameter-space integration methods [8], or the analytical approaches [3,4,6,10]. Monte-Carlo based and integration techniques are very accurate but computationally expensive and become infeasible in the presence of a large number of independent sources of variation and for large circuits. The central challenge for the analytical techniques is the computation of the probabilistic maximum of circuit delay. The proposed approaches for dealing with the challenge rely either on approximating the true distribution by a more convenient one [3,9,10], or on producing the bounds on the true distribution [5,6].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Dac 2004, June 7–11, 2004, San Diego, California, USA

Copyright 2004 ACM 1-58113-828-8/04/0006...\$5.00.

Several features seem to be required of a real statistical timing analysis tool. First, it must handle path delay correlation due to all possible sources, including reconvergence, spatial, and structural (non-spatial) correlation. Second, it should allow the analysis of timing graphs with arbitrarily distributed node delays (non-Gaussian). Third, the statistical computation must be based on parametric statistical techniques which represent distributions in a compact parametric form (by moments), not histograms. Storing a full distribution (histogram) in memory for every nominal delay point is likely to be too expensive. Finally, given the designer's overwhelming reliance on the standard STA tools such as PrimeTimeTM, using the statistical engine as the post-processor to the standard STA is an important feature.

In this paper we describe a statistical static timing analysis tool that addresses the above requirements. The exact distribution of the longest path through a probabilistic timing graph appears intractable regardless of the assumed dependence structure [11].

In contrast to the approaches based on the approximations [8][9], we believe that computing accurate bounds on the exact distribution is more appropriate. It gives greater flexibility in modeling complex sources of uncertainty, and is essentially consistent with the existing timing paradigm, i.e. the worst-case timing is also a bound, albeit a poor one. We introduce several key mathematical and algorithmic advances. In Section 3, the new mathematical formulation for deriving the entire *cdf* of the longest circuit path is described. Arbitrary correlations between the delay of the nodes in the probabilistic timing graph, and arbitrary node delay distributions can be handled. In Section 4, an efficient algorithm for computing the bounds based on a PERT-like one pass traversal of the probabilistic timing graph is described. The runtime of the algorithm is $O(V+E)$ where V, E are the number of nodes and edges in the sub-graph containing the N deterministically longest paths to be analyzed. In Section 5, the implementation details and the result of benchmark testing of the algorithm are presented.

3 MATHEMATICAL BASIS FOR PROBABILISTIC BOUNDING

3.1 Problem formulation

The clock cycle of a chip is constrained by the maximum path delay, $\max\{D_1 \dots D_N\} \leq T_{clock}$, where D_i is the delay of the i_{th} path in the circuit. The delay of each path is a random variable, described by a probability distribution. Because $\{D_1 \dots D_N\}$ is a random vector, the value of $\max\{D_1 \dots D_N\}$ is also a random variable. Then, in order to estimate the statistical properties of the chip's timing, we must find the distribution of $\max\{D_1 \dots D_N\}$.

The cumulative probability function of $\max\{D_1 \dots D_N\}$ is given by $F(t) = P\{\max\{D_1 \dots D_N\} \leq t\}$, or equivalently:

$$F(t) = P\{D_1 \leq t, D_2 \leq t, \dots, D_N \leq t\} \quad (1)$$

where $F(t)$ is the cumulative probability function defined over the path delay probability space. The contribution of this work is a way to efficiently derive bounds on the *cdf* of circuit delay for *any* probabilistic timing graph, which is simply a timing graph with random node delays.

In contrast to other approaches, we assume that path delays, not node delays, are Gaussian. This assumption is not, however, as restrictive as it appears because of the distributional convergence under the Central Limit Theorem (CLT) [12]. CLT ensures that a sum of arbitrarily distributed random variables converges to a normal distribution. It is well known that in practice convergence occurs quite rapidly, for $N=8-10$ random variables, especially for distributions that are nearly Gaussian. Also, while the standard formulation of CLT is for independent random variables, the sum of correlated random variables is also guaranteed to converge to a normal distribution for most practical models of correlation [15].

Under the above model of path delay, we can handle node delay variation described by an arbitrary probability distribution (Gaussian or non-Gaussian). Not being limited to Gaussian distributions would permit us to handle non-linear dependencies of delay to process and environmental parameter variations. The statistical model of delay can then be found by a higher-order Taylor expansion of the gate and interconnect delay expressions, as compared to the typically used linear expansions [5][6][9]. This will improve the accuracy of statistical gate and interconnect delay models.

3.2 Bounding the Distribution of Circuit Delay

We now show how we can derive a bound on the *cdf* of the longest path delay propagating through the probabilistic timing graph that has the properties defined in the previous section. Specifically, we derive an upper and a lower bounds on

$$P(\max\{D_1 \dots D_N\} \leq t)$$

where D_i is the delay of the i_{th} path. We can re-write this equation as a cumulative probability:

$$P(\max\{D_1 \dots D_N\} \leq t) = P(\bigcap\{D_i \leq t\}) \quad (5)$$

Assuming that the path delay vector is a Gaussian vector, the following theorem can be proven.

Theorem 1. For any multi-normal random vector with a correlation matrix given by Σ

$$P_{\Sigma}(\bigcap\{D_i \leq t\}) = P_{\Sigma}(\bigcap\{Z_i \leq t'_i\}) \quad (6)$$

where $t'_i = (t - \mu_{D_i}) / \sigma_{D_i}$ and $Z_i \sim N(0,1)$.

This theorem expresses the sought cumulative probability in terms of the distribution of a standard multivariate normal vector with an arbitrary correlation matrix. Note that the vector t' that determines the set over which the probability content is evaluated is not equi-coordinate, i.e. $t'_i \neq const$. The first step is to re-express the cumulative probability of Eq. 6 by a cumulative

probability of a vector with a well structured correlation matrix. This is needed because it will allow to later use accurate numerical methods to evaluate the cumulative probability. The following theorem can be used to do that [13]:

Theorem 2: Let X be distributed according to $N(0, \Sigma)$,

where Σ is a correlation matrix. Let $R=(\rho_{ij})$ and $T=(\tau_{ij})$ be

two positive semidefinite correlation matrices. If $\rho_{ij} \geq \tau_{ij}$ holds

for all i, j , then

$$P_{\Sigma=R}(\bigcap_{i=1}^N \{X_i \leq a_i\}) \geq P_{\Sigma=T}(\bigcap_{i=1}^N \{X_i \leq a_i\})$$

holds for all $a = (a_1, \dots, a_N)'$.

Corollary 1: Using Theorem 2, the cumulative distribution of Eq. 6, can be bounded from below and from above by:

$$P_{\Sigma}(\bigcap\{Z_i \leq t'_i\}) \geq P_{\Sigma_{\min}}(\bigcap\{Z_i \leq t'_i\}) \quad (7)$$

$$P_{\Sigma}(\bigcap\{Z_i \leq t'_i\}) \leq P_{\Sigma_{\max}}(\bigcap\{Z_i \leq t'_i\}) \quad (8)$$

where Σ_{\min} (and Σ_{\max}) are generated by setting all their off-diagonal elements to $\Sigma_{\min}^{ij} = \min\{\Sigma^{ij}\}$ (and $\Sigma_{\max}^{ij} = \max\{\Sigma^{ij}\}$) for all $i \neq j$.

If $\Sigma_{\min}^{ij} = \min\{\Sigma^{ij}\}$ is small, the bound produced by Equations 7 and 8 may be further improved by the following transformation. Let C be a subset of $\{1 \dots N\}$, such that $\rho_{ij} \geq q$ for all $i \in C$ and $j \notin C$, then:

$$P_{\Sigma}(\bigcap\{Z_i \leq t'_i\}) \geq P\left[\bigcap_{i \in C} \{Z_i \leq t'_i\}\right] P\left[\bigcap_{i \notin C} \{Z_i \leq t'_i\}\right]$$

Then, to get the tightest bound, we will use the largest bound:

$$\max\left\{P_{\Sigma_{\min}}(\bigcap\{Z_i \leq t'_i\}), P\left[\bigcap_{i \in C} \{Z_i \leq t'_i\}\right] P\left[\bigcap_{i \notin C} \{Z_i \leq t'_i\}\right]\right\}$$

The cumulative probabilities (7) and (8) are now described by correlation matrices with identical off-diagonal elements. Still, they require evaluating the probability content of a standard multi-normal vector over the non-equicoordinate set, which is numerically expensive. To enable a more efficient numerical evaluation of these cumulative probabilities (c.p.), we now express them in terms of the equicoordinate probability.

3.2.1 Bounding Cumulative Probability through Stochastic Majorization

In this section we will show how we can bound the cumulative probabilities (7) and (8) by a partial ordering of the mean and variance vectors, using the techniques of the theory of majorization [14]. In order to derive the bounds, however, we will need to ensure that several criteria are met, since the bounding is predicated on several properties of the probability distributions.

The notions of strong and weak majorization can be used to compare random variables and their distributions [13]. First, we introduce a set of formal definitions that allow comparing random variables and their distributions. The theorems in this section are shown without the proofs, which can be found in [13].

Let $a = (a_1, \dots, a_N)'$ and $b = (b_1, \dots, b_N)'$ be two real vectors. Let $a_{[r]} \geq \dots \geq a_{[N]}$ and $b_{[r]} \geq \dots \geq b_{[N]}$ be the ordered values of the components of a and b .

Definition : a is said to majorize b , in symbols $a \succ b$, if

$$\sum_{i=1}^N a_i = \sum_{i=1}^N b_i \text{ and } \sum_{i=1}^r a_{[i]} \geq \sum_{i=1}^r b_{[i]} \text{ for } r = 1, \dots, N-1.$$

Definition : a is said to weakly majorize b , in symbols $a \succ\text{-} b$, if

$$\sum_{i=1}^r a_{[i]} \geq \sum_{i=1}^r b_{[i]} \text{ for } r = 1, \dots, N-1.$$

The notions of strong and weak majorization can be extended to enable comparing random variables and their distributions.

Definition (Strong Stochastic Majorization) : Let X and Y be two N -dimensional random variables. X is said to stochastically majorize Y , in symbols $X \stackrel{st}{\succ} Y$, if

$$P[X \in A] \geq (\leq) P[Y \in A]$$

for every Borel-measurable Schur-convex (Schur-concave) set A .

Definition (Weak Stochastic Majorization) : X is said to weakly stochastically majorize Y , in symbols $X \succ\text{-} Y$, if

$$P[X \in A] \geq (\leq) P[Y \in A]$$

for every Borel-measurable increasing Schur-convex (decreasing Schur-concave) set A .

The key idea that we are exploiting in deriving bounds on the cumulative probabilities (7) and (8) is that for certain distributions stochastic inequalities can be established on the basis of a partial ordering of the parameter vectors using ordinary (deterministic) majorization.

The class of random vectors which can be ordered via the ordering of their parameter vectors is limited to distributions that are Schur-convex (Schur-concave). The following theorems formalize this fact [13]:

Theorem 3 : Let the random variable X_θ have a density $f(x - \theta)$ for $x \in \mathfrak{R}^N$, $\theta \in \mathfrak{R}^N$ (a location parameter vector). If $f(x)$ is Schur-convave in x , then $\xi \succ \theta$ implies that $X_\xi \stackrel{st}{\succ} X_\theta$.

Theorem 4 : Let the random variable X_θ have a density $f(x - \theta)$ for $x \in \mathfrak{R}^N$, $\theta \in \mathfrak{R}^N$. If $f(x)$ is Schur-convave in x , then $\xi \succ\text{-} \theta$ implies $X_\xi \succ\text{-} X_\theta$.

Thus, if the probability density function of random vector X_θ satisfies the properties of Theorem 3 and 4, then we can find a random vector X_ξ that will stochastically majorize, or bound the distribution of, X_θ . This is equivalent to saying that (by the definition of stochastic majorization) the probability content of X_ξ over the appropriate set will bound the probability content of X_θ over this set. This set must satisfy two properties. First, this set must enable computing the probability content that corresponds to our original purpose: the cumulative distribution of

the path delay random vector. Second, by definition, it must be Borel-measurable and Schur-convex.

The next theorem provides final answers:

Theorem 5 : (1) Let X_θ be the multivariate normal random variable with the mean vector θ . The density function $f(x - \theta)$ is Schur-concave.

(2) Let A denote the set $A = \{x \mid x_i \leq a, i = 1, \dots, N\}$. This set is Schur-concave.

We can now point out that for A defined above, $P_\theta[X \in A] = P(\bigcap \{X_i + \theta_i \leq a\})$, and putting everything together:

Theorem 6 : If $\xi \succ \theta$, then

$$P_\xi[X \in A] \leq P_\theta[X \in A], \text{ or } P(\bigcap \{X_i + \xi_i \leq a\}) \leq P(\bigcap \{X_i + \theta_i \leq a\})$$

The similar theorems can be stated for the case of weak stochastic majorization:

Theorem 7 : (1) Let X_θ be the multivariate normal random variable with the mean vector θ . The density function $f(x - \theta)$ is Schur-concave.

(2) Let A denote the set $A = \{x \mid x_i \leq a, i = 1, \dots, N\}$. This set is decreasing and Schur-concave.

Theorem 8 : If $\xi \succ\text{-} \theta$, then

$$P_\xi[X \in A] \leq P_\theta[X \in A], \text{ or } P(\bigcap \{X_i + \xi_i \leq a\}) \leq P(\bigcap \{X_i + \theta_i \leq a\})$$

In the next section, we use the above results to bound the cumulative distribution of the path delay vector. In doing that, it will also become clear why the parallel notions of strong and weak stochastic majorization had to be developed.

We can now easily find an *equi-coordinate* parameter vector that is majorized by the mean path delay vector t . Using the definition of majorization, one may check

Fact 1. If $\underline{t} = (t_1, \dots, t_N)$ and $\bar{t} = (\bar{t}, \dots, \bar{t})$, where $\bar{t} = \frac{1}{N} \sum_{i=1}^N t_i$, then

$$\underline{t} \succ \bar{t} \longrightarrow P(\bigcap \{Z_i \leq t_i\}) \leq P(\bigcap \{Z_i \leq \bar{t}\})$$

It is impossible, however, to find an *equi-coordinate* parameter vector that (strongly) majorizes t . Here we need to resort to *weak majorization*. Using Theorem 8, we can show that:

Fact 2. If $\underline{t} = (t_1, \dots, t_k)$ and $t_{\min} = (\min\{t\}, \dots, \min\{t\})$, then

$$\underline{t} \succ\text{-} t_{\min} \longrightarrow P(\bigcap \{Z_i \leq t_{\min}\}) \leq P(\bigcap \{Z_i \leq t_i\})$$

Indeed, let $\Delta t = t_{\min} - \underline{t} \leq 0$, $\underline{0} = \{0, \dots, 0\}$. Then, $\underline{0} \succ\text{-} \Delta t$ and

$$P(\bigcap \{Z_i \leq t_i\}) = P(\bigcap \{Z_i + \Delta t_i \leq t_{\min}\}) \geq P(\bigcap \{Z_i + \underline{0} \leq t_{\min}\})$$

We have finally bounded the original cumulative probability by cumulative probabilities expressed in terms of an equi-coordinate vector, a correlation matrix with identical off-diagonal elements, and the standard multivariate normal $N(0, I)$ vector.

3.2.2 Evaluating the Cumulative Probability

The prior transformations reduced the problem to estimating the cumulative probabilities of a well-structured object whose probability content is amenable to numerical evaluation. This can be done by using pre-generated look up tables for a wide range of dimensionalities, coordinates, and correlation coefficients. The tables have been generated via the Monte-Carlo integration of the cumulative probability of a multivariate standard normal vector for a range of: (1) vector cardinalities (e.g. number of paths, N); (2) the off-diagonal correlation coefficients. To reduce the data storage requirements, interpolation is used to find the cumulative probability values.

4 ALGORITHM FOR COMPUTING CIRCUIT DELAY DISTRIBUTION

4.1 Linear Additive Statistical Delay Model

The exposition above assumed that the path-to-path covariance matrix is given. In reality, there will be a significant cost to computing this matrix since this operation is $O(N^2)$ with the number of paths N . In this section, we derive a set of results that allow computing the correlation matrix and the vector of variances in an algorithmically efficient manner.

The efficient algorithm to compute the bounds is based on the realization that for *linear additive* models of statistical dependence between the delays of nodes in the probabilistic timing graph, the computation of bounds is greatly simplified. It turns out that under this model, the binary relation of correlation can be converted into a unary relation that reflects the degree of correlation of a node delay variation to the common level of variation among the nodes. This transformation allows key computational improvements. The linear additive statistical model is a natural result of ascribing the total node delay variation to two components: inter-chip (or chip-to-chip) and intra-chip (or within-chip) components of variation [16]:

$$d = d^{inter} + d^{intra}$$

The first is due to the chip-to-chip, wafer-to-wafer, and lot-to-lot variations in the processing conditions. The second is due to local variations on the same chip. Note that the environmental variations can also be captured by the intra-chip source of variation.

We will now work in the space of node delays. For the purpose of notational simplicity, in this section we refer to delay by x . For reasons of greater analytical suitability, we write the statistical delay model as a sum of two *independent* weighted random variables:

$$x_i = b_i z + a_i y_i$$

where x_i is the overall delay of node i , y_i is the node-specific component of variation, and z is the chip-level component of variation. The y_i and z can be modeled as independent random variables with $E\{z\} = E\{y_i\} = 0$ and $Var\{y_i\} = Var\{z\} = 1$. Together, the coefficients a and b determine the variance of x and the correlation of x to the chip-level component of variation:

$$Var\{x_i\} = a_i^2 Var\{y_i\} + b_i^2 Var\{z\} = V_i$$

Most importantly, the above formulation allows us to parsimoniously represent the relation of correlation between nodes. It's easy to show by writing out the covariance expression term by term that:

$$cor(x_i, x_j) = \frac{cov(x_i, x_j)}{Var\{x_i\}^{1/2} Var\{x_j\}^{1/2}} = \frac{b_i b_j}{\sqrt{a_i^2 + b_i^2} \sqrt{a_j^2 + b_j^2}}$$

We now introduce a concept of *node-to-chip* correlation. This is the correlation of any particular node delay to the common, chip-to-chip delay variation. It is this transformation that breaks the computational bottleneck. The transformation is achieved by introducing the pure chip-level variation $x_o = z$, defined by $a_o = 0$ and $b_o = 1$. Then, $cor(x_i, x_o) = b_i / \sqrt{a_i^2 + b_i^2}$.

Obviously, the binary relationship of correlation between the delay of node i , and the chip-level variation is expressed exclusively in terms of the attributes that describe the statistical structure of delay of node i . To further drive this point home, we will use the following notation to highlight the unary nature of the node-to-chip correlation: $cor(x_i, x_o) = c_i$. Now we can express the relationship of correlation between the two node delays as:

$$cor(x_i, x_j) = c_i \cdot c_j.$$

This is the key result because it allows representing a binary operator as a product of two unary operators. Note, however, that this is simply an algorithmically convenient way of capturing the fact that random variables described by linear additive statistical models have a specially simple structure of correlation.

It is easy to see that the above definition can be extended to compute the correlation between two sums:

$$cor\left(\sum_{i=1}^{k_1} x_i, \sum_{j=1}^{k_2} x_j\right) = \frac{\left(\sum_{i=1}^{k_1} b_i\right) \cdot \left(\sum_{j=1}^{k_2} b_j\right)}{\sqrt{\sum_{i=1}^{k_1} a_i^2 + \left(\sum_{i=1}^{k_1} b_i\right)^2} \sqrt{\sum_{j=1}^{k_2} a_j^2 + \left(\sum_{j=1}^{k_2} b_j\right)^2}}$$

where a_i , b_i and a_j , b_j are the statistical coefficients of the nodes from the first and second groups of summands.

The node-to-chip correlation can be directly computed by setting the second sum to be the chip-level variation, z . This simply requires setting $k_2=1$, $a_o = 0$ and $b_o=1$. Then, setting $k=k_1$,

$$f_{k,0} \triangleq cor\left(\sum_{i=1}^k x_i, z\right) = \frac{\sum_{i=1}^k b_i}{\sqrt{\sum_{i=1}^k a_i^2 + \left(\sum_{i=1}^k b_i\right)^2}} \quad (9)$$

Finally, note that using the above notation, the correlation between two sums of node delays can be expressed succinctly by:

$$f_{k_1, k_2} \triangleq cor\left(\sum_{i=1}^{k_1} x_i, \sum_{j=1}^{k_2} x_j\right) = f_{k_1,0} f_{k_2,0}$$

Again, we can represent a binary operator that describes the correlation between sums of nodes, as a product of two unary operators.

4.2 Efficient Computation of the Bounds

From the full probabilistic timing graph G , we first extract a sub-graph G' that contains N deterministically (with respect to the mean delay value) longest paths using an efficient path-extraction algorithm, e.g. [17]. Assuming the longest path is known (D_{\max}) and the timing threshold (ΔD) is set, the N longest paths with delays in the interval $\{D_{\max} - \Delta D, D_{\max}\}$ can be extracted in a very reasonable time. We now use the unary correlation to describe an efficient algorithm for computing the lower and upper bounds on the circuit delay *cdf*. This algorithm has the computational complexity of $O(V+E)$ where V, E are the number of nodes and edges in G' . The algorithm computes all the variables needed for establishing the bounds in a single PERT-like pass of the graph.

The first key ingredient is the update formula for computing the correlation of the sum of $(k+1)$ variables based on the node-to-chip correlation of k variables and the node-to-chip correlation of the $(k+1)^{th}$ node:

Theorem 10. The update formula is $f_{k+1,0} = \sqrt{\frac{1+\alpha}{f_{k,0}^{-2} + \alpha + \beta}}$

where $\alpha \triangleq 2 \frac{b_{k+1}}{h(k)} + \frac{b_{k+1}^2}{h(k)^2}$, $\beta \triangleq \frac{a_{k+1}^2}{h(k)^2}$, $h(k) \triangleq \sum_{i=1}^k b_i$, and $g(k) \triangleq \sum_{i=1}^k a_i^2$.

This expression would allow an exact computation of path-to-path delay correlation, and path variance, for all paths through the timing graph. It is clearly impossible to compute the full correlation matrix in a one-pass manner. The second ingredient is the realization that the computation of the entire path-to-path correlation matrix is unnecessary, since the bounds rely only on the minimum and maximum values of the correlation matrix. With that in mind, we can use an efficient graph-traversal algorithm to compute only the smallest (largest) path-to-path correlation, rather than every path-to-path correlation. The algorithm is based on the following two theorems. By directly analyzing the unary definition of path-to-path correlations ($f_{k_1, k_2} = f_{k_1, 0} f_{k_2, 0}$) we can see that:

Theorem 11: The minimum (maximum) path-to-path correlation in a timing graph is given by the product of two smallest (largest) path-to-chip correlations in the graph.

At the same time, we are guaranteed to find the two smallest (largest) path-to-chip correlation values by propagating them in a PERT-like fashion from the primary inputs to the primary outputs:

Theorem 12. Computing two smallest (largest) path-to-chip correlations in the probabilistic timing graph can be done in a single traversal of a graph.

Similarly, the maximum and minimum path variance can be computed in one PERT-like traversal of the graph. We can show that, defining the variance update function, VU :

$$VU(k+1) \triangleq a_{k+1}^2 + 2h(k)b_{k+1} + b_{k+1}^2$$

$$Var\{\sum_{i=1}^{k+1} x_i\} = Var\{\sum_{i=1}^k x_i\} + VU(k+1)$$

In summary, all the variables needed for computing the bounds on the true *cdf* can be found in a single traversal of the timing graph.

5 RESULTS

The algorithms described in Sections 3 and 4 have been implemented in C, and have been run on Windows PC with CPU 2.4 GHz and 512 MB memory. Below is our flow's pseudo-code:

1. Pre-generate cumulative probability tables for different sample size and correlations.
2. Using standard STA based on mean delay values, extract N longest paths with delays in $\{D_{\max} - \Delta D, D_{\max}\}$
3. In a single traversal of the timing graph:
 - Compute max and min path-to-path correlations (Σ_{\min}^{ij} and Σ_{\max}^{ij})
 - Compute min and max path variance among M paths
4. Compute from the tables lower and upper bounds on circuit delay *cdf*.

The algorithms have been tested on a set of combinational ISCAS85 benchmark circuits. The 3σ values of node delays were set at 15-20% of the mean values. The node delays were described by a correlation matrix with entries ranging from 0.8-0.95. The exact cumulative distribution function was computed via Monte-Carlo runs of the deterministic static timing analysis algorithm with samples taken from the relevant correlated gate delay distributions. For each simulation, 1000 iterations of Monte Carlo were run.

The bounds generated by the algorithm follow the Monte Carlo distribution very closely (Figure 1 and 2a). Table 1 shows the errors of both upper and lower bounds with respect to the exact distribution. The bounds were computed for $N=20$ and 50 longest paths. Across the benchmarks, the lower bound rms error is 0.3%-1.4% while the upper bound rms error is 0.7%-2.1%. Note that we are specifically interested in the lower bound since it gives us a conservative value of circuit delay at any confidence level. Importantly, the lower bound becomes tighter at higher confidence levels, giving a more reliable estimate of parametric yield: at the 95th percentile the maximum error across all the benchmark circuits is 0.7%. Figure 2b demonstrates that accurately accounting for node delay correlations is crucial in predicting the shape of *cdf*: the mean value of an uncorrelated case is larger than that of the correlated one, while the spread is much smaller.

Table 1 also contains the evaluation of run time of the algorithm. The Monte Carlo (for 1000 samples) is substantially slower than our algorithm. The bounding algorithm is very fast: the runtime is

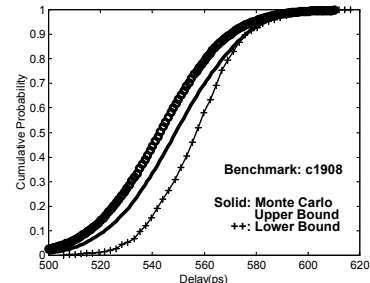


Figure 1. Comparison of Monte Carlo *cdf* and bounds for the benchmark circuit c1908. The lower bound *rms* error is 0.6%.

under 20s for N=20 and under 40s for N=50 longest paths, over all the benchmarks. (At the same time, the longest path extraction takes up to 90% of total runtime - 188s and 210s for N=20,50). Figure 3 shows a close-to-linear growth in the runtime of the algorithm as a function of circuit size, which makes practical the use of the algorithm for significantly larger circuits.

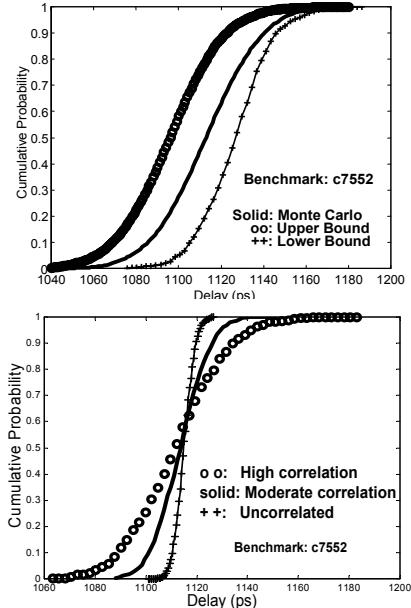


Figure 2. a) Comparison of cdfs for c7552. The lower bound rms error is 1.0%; b) Change in cdf for c7552 depending on the level of node delay correlations.

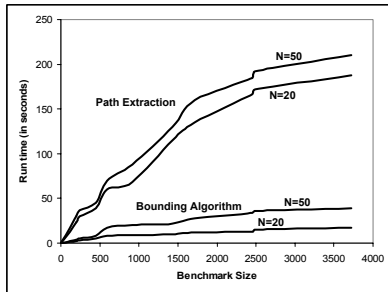


Figure 3. Run time of the algorithm for different circuit sizes.

Benchmark	Number of nodes	Bounding Error RMS (%)		95th percentile error (%)	Run-time (sec) (Monte Carlo)	Run-time (sec)	
		Lower Bound	Upper Bound			Path Extraction	Bounding Algorithm
c17	13	1.0/0.8	1.2/1.5	0.7/0.6	8	0.994/1.2	0.006/0.010
c432	245	0.4/0.3	1.2/1.4	0.02/0.02	435	23/30	2/4
c499	198	1.2/1.0	2.1/2.4	0.4/0.3	810	30/37	3/5
c880	445	1.1/0.9	1.7/1.9	0.5/0.5	1258	39/45	5/8
c1355	589	1.3/1.1	1.8/2.1	.001/.001	1567	54/60	8/17
c1908	915	0.6/0.4	0.8/0.9	.001/.001	1815	65/82	9/20
c2670	1428	0.7/0.5	0.7/1.1	0.02/0.02	2084	78/97	10/22
c3540	1721	0.8/0.7	0.8/0.8	0.1/0.1	2250	89/102	12/28
c5315	2487	0.8/0.6	1.1/1.2	0.2/0.2	2480	110/138	13/34
c6288	2450	1.1/0.9	1.2/1.6	0.6/0.5	2615	153/179	15/36
c7552	3721	1.0/0.8	1.4/1.5	0.2/0.2	2819	188/210	17/39

Table 1. Prediction errors and run-time for Monte-Carlo and for bounding algorithm (double entries are values for 20/50 paths).

6 REFERENCES

- [1] R. Hitchcock, "Timing Verification and the Timing Analysis Program," *Proc. of DAC*, 1982.
- [2] H.-F. Jyu *et al*, "Statistical timing analysis of combinational logic circuits," *Trans. on VLSI Systems*, vol.1, (no.2), pp. 126-37, 1993.
- [3] M. Berkelaar, "Statistical Delay Calculation", *International Workshop on Logic Synthesis*, pp. 15-24, 1997.
- [4] A. Gattiker *et al*, "Timing yield estimation from static timing analysis," *Proc. of ISQED*, pp. 437-442, 2001.
- [5] M. Orshansky and K. Keutzer, "General probabilistic framework for worst-case timing analysis," *Proc of DAC*, pp. 556-561, 2002.
- [6] A. Agarwal *et al*, "Computation and Refinement of Statistical Bounds on Circuit Delay," *Proc of DAC*, pp. 348-353, 2003.
- [7] A. Agarwal *et al*, "Path-Based Statistical Timing Analysis Considering Inter- and Intra-Die Correlations," *TAU*, pp. 16-21, 2002.
- [8] J.A.G. Jess *et al*, "Statistical timing for parametric yield prediction of digital integrated circuits," *Proc of DAC*, pp. 932-937, 2003.
- [9] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single pert-like traversal," *Proc. of ICCAD*, pp. 621-625, 2003.
- [10] A. Devgan, C. Kashyap, "Block-based static timing analysis with uncertainty," *Proc. of ICCAD*, pp. 607-614, 2003.
- [11] A. Nadas, "Probabilistic PERT," *IBM J. of Research and Development*, vol. 23, no. 3, pp. 339-347, 1979.
- [12] W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley and Sons, 3rd Edition, 1968.
- [13] Y. L. Tong, *Probability Inequalities in Multivariate Distributions*, Academic Press, 1980.
- [14] A. W. Marshall, I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, 1979.
- [15] S. Louhichi, "Rates of Convergence in the CLT for Some Weakly Dependent Random Variables," *Theory Probab. Appl.* Vol. 46, No. 2.
- [16] S. Nassif, "Delay Variability: Sources, Impact and Trends," *Proc. of International Solid-State Circuits Conference*, 2000.
- [17] S. Yen, D. Du, S. Ghanta, "Efficient algorithms for extracting the k most critical paths in timing analysis," *Proc. of DAC*, pp. 649-652, 1989.