Statistical Clock Skew Analysis Considering Intradie-Process Variations

Aseem Agarwal, *Student Member, IEEE*, Vladimir Zolotov, *Member, IEEE*, and David T. Blaauw, *Member, IEEE*

***Abstract*—With shrinking cycle times, clock skew has become an increasingly difficult and important problem for high performance designs. Traditionally, clock skew has been analyzed using case-files which cannot model intradie-process variations and hence result in a very optimistic skew analysis. In this paper, we present a statistical skew analysis method to model intradie process variations. We first present a formal model of the statistical clock-skew problem and then propose an algorithm based on propagation of joint probability density functions in a bottom-up fashion in a clock tree. The analysis accounts for topological correlations between path delays and has linear runtime with the size of the clock tree. The proposed method was tested on several large clock-tree circuits, including a clock tree from a large industrial high-performance microprocessor. The results are compared with Monte Carlo simulation for accuracy comparison and demonstrate the need for statistical analysis of clock skew.**

***Index Terms*—Clock skew, probability, process variation, statistical analysis.**

I. Introduction

CLOCK SKEW results from the unequal propagation delay of clock paths from the source of the clock tree to the various sink nodes at the latch points and directly impacts the performance of a design. With rapidly increasing clock frequencies, the allowable clock skew is increasingly constrained, making clock skew a critical concern for high-performance processors. Clock skew can be introduced either at design time, during fabrication of the design, or during its operation. During the design phase, clock skew can arise due to unbalanced clock-path delays resulting from unexpected changes in the capacitive loading at the clock sinks and routing constraints. To address this, extensive work has been performed on automatic sizing and routing of clock trees to minimize skew during design time [1]–[8]. However, even if clock-skew constraints are met at design time, process variations can introduce unwanted clock skew during the fabrication of the chip, thereby compromising the obtainable performance. Also, environmental fluctuations, such as power-supply variations and coupling noise can introduce clock skew during the operation of the design and a number of methods for analyzing such sources of clock skew have been presented in [9] and [10].

Manuscript received August 1, 2003; revised December 10, 2003. This work was supported in part by the SRC under Contract 2001–HJ-959 and in part by the NSF under Grant CCR-0205227. This paper was recommended by Associate Editor F. N. Najm.

A. Agarwal and D. T. Blaauw are with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48105 USA (e-mail: abagarwa@umich.edu).

V. Zolotov is with Motorola, Inc., Austin, TX 78759 USA.

Digital Object Identifier 10.1109/TCAD.2004.831573

In this paper, we propose a statistical method to analyze the impact of process variations on clock skew. Process variations result in uncertainty in the device and interconnect characteristics, such as effective gate length, doping concentrations, oxide thickness, and ILD thickness, and are a source of significant clock skew. In general, process variations can be divided into *interdie* and *intradie* variations. Interdie variations represent differences in device characteristics from one die to the next, while intradie variations represent differences in device characteristics within a single die. Intradie variations can either be systematic or random. The systematic component is deterministic in nature due to topological dependencies of device processing, such as CMP effects and optical proximity effects [11]–[13]. In some cases, such topological dependencies are directly accounted for in the analysis of clock skew, thereby reducing the statistical variations [14]–[16], whereas in other cases, such variations are treated as random. Furthermore, random variations can either be spatially correlated, meaning that devices close to each other are more likely to have similar characteristics than those spaced far apart, or completely independent. Causes of spatially correlated variations are equipment-related effects, such as lens aberration and exposure time, whereas doping fluctuations cause independent random variations.

Traditionally, clock skew is computed using case analysis, where all devices are assumed to have identical best-case, nominal, or worst-case characteristics. Such analysis is appropriate for interdie process variations. However, it cannot model intradie variations where devices have different characteristics on the same die. Case analysis, therefore, results in an *optimistic* skew estimate, as the mismatch between the devices in a clock tree is ignored. With continuous shrinking of process dimensions, intradie variations are becoming increasingly prominent and case-analysis is no longer valid. It is, therefore, critical that a statistical analysis of the clock skew is performed to determine the expected distribution of the skew across the manufactured die. Once the skew distribution is computed, the expected number of die meeting a specific skew can be determined. Statistical analysis of clock skew is also useful during the design of a clock tree to reduce its sensitivity to process variations and increase its robustness. Hence, the target application for a statistical analysis could either be in the synthesis flow or during physical verification.

Recently, a method for statistical clock-skew analysis based on Monte Carlo simulation was proposed [17]. However, Monte Carlo-based approaches have very high runtimes, especially for large clock designs. A probabilistic approach to clock-skew analysis was proposed in [18] and [19], and has an efficient runtime. However, the proposed analysis is restricted to binary

clock trees and also uses a Gaussian distribution to approximate the maximum and minimum of two Gaussian random variables, which may compromise the accuracy of the analysis.

In this paper, we therefore propose a new approach to clock-skew analysis, which accurately models intradie-process variations and has a linear runtime complexity with circuit size. Our analysis is focused on random variations, meaning that topological dependencies are either removed prior to the analysis or are treated as random variations. We provide a formal definition of the statistical clock skew problem from which we derive our proposed analysis method. Statistical clock skew analysis is complicated by the correlation between the minimum and maximum path delays in a clock tree. The approach proposed in this paper uses joint probability density functions (JPDFs) that preserve this correlation between minimum and maximum delays in an efficient manner. The JPDFs are propagated in a bottom up fashion along the clock tree in a single pass, and we present efficient methods for merging and propagating JPDFs during the traversal. The proposed method computes the skew distribution for the entire clock tree as well as the skew distribution of all subtrees simultaneously and therefore allows the designer to identify which portions of the clock tree are most prone to process variations. The presented methods were implemented and tested on a number of clock tree circuits, including a large clock structure from an industrial high-performance microprocessor design. Comparison of results with Monte Carlo simulation confirms the correctness of the approach and demonstrates its efficiency. A comparison with traditional case analysis shows the importance of statistical clock-skew analysis.

The remainder of this paper is organized as follows. In Section II, we present the problem definition and modeling assumptions. In Section III, we discuss our approach and implementation for statistical clock skew computation. In Section IV, we show experimental results and comparisons with Monte Carlo simulation. Finally, in Section V, we draw our conclusions.

II. Problem Definition and Modeling Assumptions

In this section, we define the statistical clock-skew problem and discuss our modeling assumptions. We consider clock networks as composed of driver gates, such as buffers, inverters, and distributed resistance–inductance–capacitance interconnects. In this paper, we restrict our analysis to clock networks that have a tree topology, meaning that the circuit does not have reconvergent fanout. Some very high-performance clock-tree designs are constructed using multidriven meshes and can only be represented by directed acyclic graphs (DAGs). While such DAG clock networks cannot be modeled with our proposed approach, a number of clock networks, especially those in application specific integrated circuit design, have a tree topology. Also, in most cases, DAG clock networks are composed of clock-tree-driven meshes. In these cases, we can analyze the clock tree up to driven meshes. It would require an extension of our approach to analyze the mesh networks themselves.

We represent a clock tree with a so-called *timing tree*, which is similar to the well-known timing graph, except that its topology is restricted to a tree. The root of the tree is the primary clock driver and the lowest level gates of the clock hierarchy are the sink nodes that drive the latches in the design. An example of a clock tree and its corresponding timing-tree representation is shown in Fig. 1. Each edge in the timing tree represents the delay from a driver-gate input to an interconnect sink point and, therefore, represents the sum of gate and interconnect delays. For most clock trees, the dominant factor in delay variation is the driver delay uncertainty, due to variability of process parameters, such as gate length [20]. In this paper, we therefore focus on driver-delay variation, although the analysis can be easily extended to incorporate interconnect-delay variability, using variational interconnect modeling methods such as those discussed in [16]. Also, as shown in [22], the impact of these variations can be assumed to be linear, for small variabilities. Hence, these variabilities can be handled in the presented framework by representing them as additional edges in the timing tree.

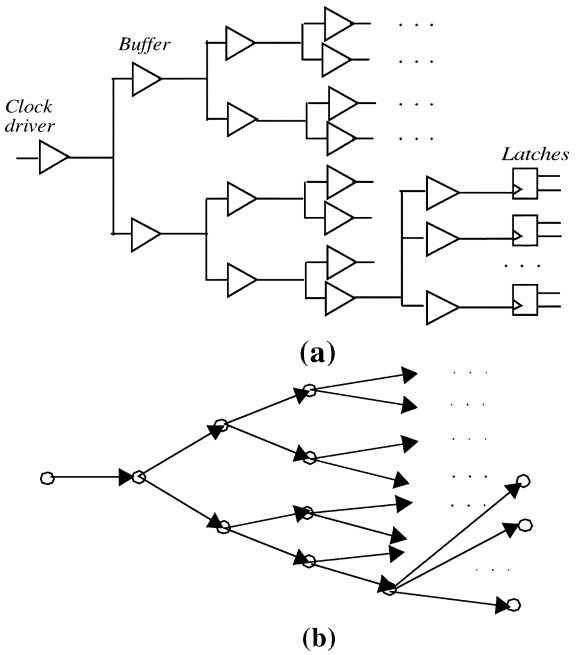

Fig. 1. Clock tree and its timing tree representation.

Since timing trees are a special case of timing graphs, they inherit all common attributes of timing graphs, such as the definition of path delay, arrival times, critical paths, etc.

A *deterministic timing tree* (DTT) is defined as a timing tree where each edge has a fixed delay. The skew of a DTT is defined in terms of the minimum and maximum path delay in the timing tree, as stated below:

Definition 1: The minimum delay of a DTT is the minimum of all path delays $d_{p,i}$ from the root node to any of the sink nodes $\{S_1, S_2, \ldots, S_n\}$

$$d_{\min} = \min(d_{p,1}, d_{p,2} \ldots, d_{p,n}). \tag{1}$$

Definition 2: The maximum delay of a DTT is the maximum of all path delays $d_{p,i}$ from the root node to any of the sink nodes

$$d_{\max} = \max(d_{p,1}, d_{p,2} \ldots, d_{p,n}). \tag{2}$$

Definition 3: Clock skew for a DTT can be defined as the difference between the maximum delay and minimum delay of a DTT

$$s = d_{\max} - d_{\min}. \tag{3}$$

Clock skew, therefore, is the maximum arrival time difference between any pair of sink nodes.

Typically, the aim of the designer is to create a clock tree with zero skew, referred to as a *zero-skew* clock tree. However, in high-performance design, clock skew is sometimes intentionally introduced by the designer to accommodate unbalanced combinational logic delays in the circuit, referred to as a *nonzero skew* clock tree. By setting nonzero clock skew targets the designer effectively enables cycle stealing or time borrowing which can improve the performance of the design. However, any deviation of the skew from their intended targets in a nonzero skew clock tree will degrade the performance of the design in the same manner as it does for a zero-skew clock tree. For clarity, we derive our analysis in this paper for a zero-skew clock tree, noting that the analysis can be easily extended to nonzero skew clock trees with intentional skew targets. Also, we define clock skew as the maximum arrival time difference between any pair of sink nodes. As skew is meaningful only between sink nodes corresponding to adjacent pairs of latches, considering skew between any pair of sink nodes is conservative. If necessary, however, it is straightforward to restrict our analysis to only a particular set of sink node pairs to reduce this pessimism.

At design time, process variations create uncertainty in the gate delays of the clock tree. Hence, we define a so-called *probabilistic timing tree* (PTT), T_p where the delay of edge e is modeled with random variable D_e. Each random variable D_e is characterized by its probability density function (PDF) $p_e(D_e)$. Although we formulate the clock skew problem using continuous PDFs, we use discretized versions of these functions in our implementation, similar to those discussed in [21].

For the purpose of our analysis, we assume that edge delays are independent random variables. However, certain device parameters, such as gate length, will exhibit spatial correlation, meaning that drivers that are closely spaced together are more likely to have similar device parameters than those spaced further apart. Such spatial correlations will introduce dependencies between the edge delay random variables in the PTT. However, in typical process technologies, spatial correlation is reported to drop off sharply for distances greater than 100–300 μm [31]. The driver gates in a clock tree are typically spaced relatively far apart, as they are distributed evenly in the die, with separation typically greater than 300 μm. This, therefore, diminishes the impact of spatial correlation for typical clock-tree designs. However, for situations where spatial correlation does impact driver-delay variability, spatial correlations must be incorporated in the presented framework. A possible extension to handle correlated effects would be to express the correlation as a sum of two random variables, one which is perfectly correlated and the other which is independent. Then, the independent component can be handled by our current approach, while the perfectly correlated part can be handled separately by enumeration, and then combined together. Also, systematic variations can be handled in our methodology by changing the mean of the distribution for edge delay random variables, as a preprocessing step and, hence, is orthogonal to the methodology explained in this paper.

Since all edge delays take a deterministic value on a manufactured die, the sample space consists of all possible dies with different edge delay combinations. Note that the edge delays are deterministic only in the context of process variations, but may still vary due to environmental variations. The probability that a manufactured die has a driver with a delay in interval $[d_1, d_2]$ is

$$\Pr\{d_1 \le D_e \le d_2\} = \int_{d_1}^{d_2} f(D_e) dD_e. \tag{4}$$

Furthermore, since the edge delays are independent random variables, the probability of the occurrence of a particular combination of edge delays is simply the product of the probabilities of the occurrence of each individual edge delay [32]. Finally, since all clock-tree characteristics, such as minimum and maximum path delay and skew are defined over the sample space, they are also random variables. In statistical clock-skew analysis, the goal is to obtain the PDF or the cumulative distribution function (CDF) of the clock skew, based on the PDF or CDF of the edge delays in the PTT.

III. Proposed Approach for Statistical Skew Computation

We start with a formal definition of the CDF $F(S)$ of clock skew over the sample space of manufactured dies. The probability of skew s being equal or less than value S can be expressed as the integral over the sample space of the DTTs which satisfies $s \le S$. As mentioned, the probability of occurrence of a DTT is the product of the probabilities of occurrence of its individual edge delays $p_{e,i}(t)$, which leads to the following expression for clock skew CDF:

$$\begin{aligned} F(S) &= \Pr\{s \le S\} \\ &= \int_{\mathrm{d_{max}}-\mathrm{d_{min}} \le S} \cdots \int p_{e,1}(t_1) p_{e,2}(t_2) \ldots p_{e,n}(t_n)\, dt_1 \ldots dt_n \end{aligned} \tag{5}$$

where $d_{\min}$ and $d_{\max}$ are defined for a DTT in (1) and (2).

The brute-force approach for computing the CDF of clock skew would involve a complete enumeration of the sample space consisting of all possible DTTs, computing the likelihood of their occurrence, and determining if the $d_{\min}$ and $d_{\max}$ associated with each satisfies $d_{\max} - d_{\min} \le S$. This approach has exponential complexity with respect to the number of edges in the graph and, hence, is not practical.

A more intuitive approach would be to implement a statistical timing analysis method that mirrors the approach for computing skew in a DTT, according to (1)–(3). Using one of several statistical timing-analysis methods presented in [21]–[30], we can easily compute the earliest (minimum) and latest (maximum) arrival time distributions of each clock sink in the tree. We can then define the maximum and minimum delay of the clock tree as random variables $D_{\max}$ and $D_{\min}$, similar to that

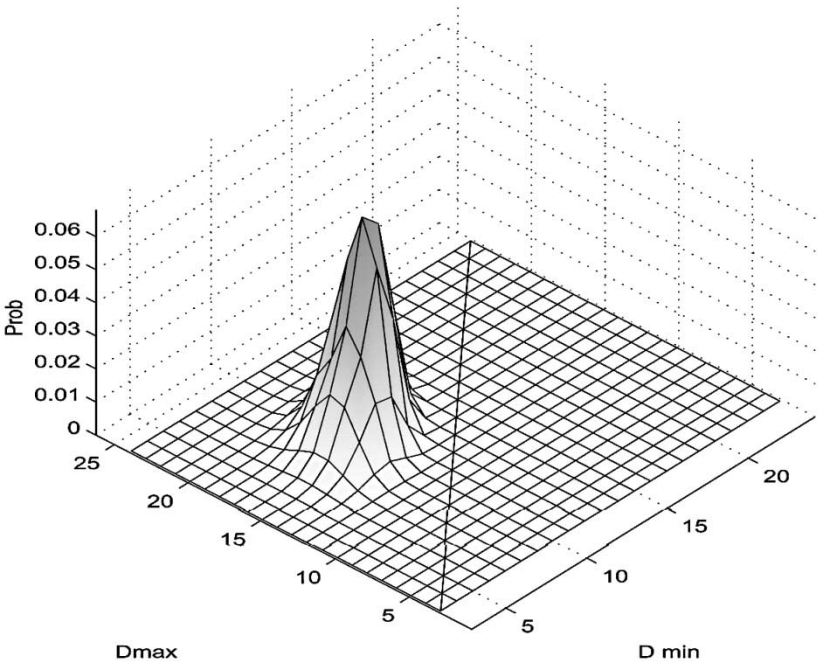

Fig. 2. Joint distribution of $D_{\max}/D_{\min}$ among a group of sinks.

for the DTT in (1) and (2), and attempt to compute their probability distributions by taking a statistical maximum and minimum over all sink nodes. From the difference of these two random variables, the skew is then obtained. Unfortunately, this approach is complicated by the correlations that must be accounted for during the computation. First, the arrival times at sink nodes are correlated since timing paths to sink nodes typically share multiple edges in the timing tree. This correlation must be explicitly expressed when computing the maximum and minimum clock tree delay distributions, which is computationally difficult. Second, the minimum and maximum clock tree delays themselves are correlated. This is immediately obvious from the fact that minimum delay can never exceed the maximum delay and vice versa. Therefore, the correlation between the minimum PTT delay $D_{\min}$ and the maximum PTT delay $D_{\max}$ must be determined in order to correctly compute the distribution of skew, again complicating the analysis.

We, therefore, propose an alternate approach to statistical skew analysis as detailed in the next section. The key idea is to avoid separate computation of the minimum and maximum PTT delays and instead compute their joint probability density function (JPDF) which preserves their correlation information. Furthermore, we show that by propagating the JPDF of minimum and maximum PTT delay in a bottom up traversal of the clock tree, the JPDFs that are merged during the traversal are independent, which simplifies the analysis. From the JPDF of $D_{\min}$ and $D_{\max}$, the distribution of the clock skew is then computed in a straightforward manner. The propagation and merging of JPDFs during the bottom-up traversal of the clock tree is performed using discretized distributions. We propose an efficient method for merging of JPDFs during the traversal which reduces the worst-case runtime complexity for merging from $O(n^4)$ to $O(n^2)$, where n is the number of discretization in each dimension of the JPDF. Note that the runtime complexity in terms of circuit size is linear in all cases.

A. Computation of Clock-Skew Distribution

We now define JPDF and the joint CDF (JCDF) of minimum and maximum PTT delays and then show how the skew, as defined in (5) can be computed using such a JPDF. The JCDF of $D_{\min}$ and $D_{\max}$ is defined over the sample space as follows:

$$F(D_{\min}, D_{\max}) = \Pr\{\min(d_{p,i}) \leq D_{\min}, \max(d_{p,i}) \leq D_{\max}\} \tag{6}$$

where $d_{p,i}$ are clock-tree path delays and minimum and maximum operations are taken over all clock net sinks. The JPDF can be obtained from the JCDF through differentiation

$$f(D_{\min}, D_{\max}) = \frac{\partial^2}{\partial D_{\min} \partial D_{\max}} F(D_{\min}, D_{\max}). \tag{7}$$

For numerical computation, it is often more convenient to discretize the JPDF. An example of the discretized JPDF for $D_{\min}$ and $D_{\max}$ is shown in Fig. 2, as a mesh plot. Here, the darker regions indicate the areas with higher probability. The entire distribution lies above the $D_{\max} = D_{\min}$ line. This follows from the obvious property that $D_{\min}$ can never be greater than $D_{\max}$.

Using the JPDF of $D_{\min}$ and $D_{\max}$, we can compute the probability of manufacturing a chip with minimum and maximum delays within the intervals $[D_{\min,1}, D_{\min,2}]$ and $[D_{\max,1}, D_{\max,2}]$ as follows:

$$\begin{aligned}&\Pr\{D_{\min,1} \leq D_{\min} \leq D_{\min,2}, D_{\max,1} \leq D_{\max} \leq D_{\max,2}\} \\ &= \int_{D_{\min,1}}^{D_{\min,2}} \int_{D_{\max,1}}^{D_{\max,2}} f(D_{\min}, D_{\max}) dD_{\max} dD_{\min}.\end{aligned} \tag{8}$$

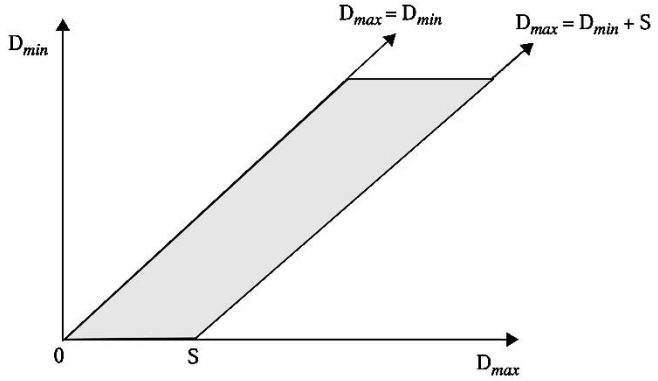

Fig. 3. Graphical representation of integration region in (10).

We now write the expression of clock skew CDF in (5) in terms of the JPDF of minimum and maximum PTT delays as follows:

$$F(S) = \Pr\{s \le S\} = \int\limits_{D_{\max} - D_{\min} \le S} \int f(D_{\min}, D_{\max}) dD_{\max} dD_{\min} \tag{9}$$

which follows directly from the definition of the JPDF in (8) and the definition of clock skew in (5). Finally, we rewrite the integral in (9) above using simple manipulation of the integral limits as follows (as illustrated in Fig. 3):

$$F(S) = \Pr\{s \le S\} = \int\limits_{0}^{\infty} \int\limits_{D_{\min}}^{D_{\min}+s} f(D_{\min}, D_{\max})\, dD_{\max} dD_{\min}. \tag{10}$$

Using the above expression of clock skew, and a given discretized JPDF of $D_{\min}$ and $D_{\max}$, the computation of the clock-skew PDF can be accomplished through simple integration. We now show how the JPDF of $D_{\min}$ and $D_{\max}$ for a PTT can be efficiently computed using a single bottom-up traversal and how the final clock-skew distribution is computed.

B. Joint Probability Distribution Computation

We compute the JPDF $f(D_{\min}, D_{\max})$ for the minimum and maximum path delays in a PTT in a bottom up fashion. The JPDF $f_i(D_{\min}, D_{\max})$ for an internal node n_i in the PTT represents the joint probability distribution of minimum and maximum path delays from node n_i to any of the leaf node of n_i. The JPDF $f_i(D_{\min}, D_{\max})$ at node n_i is defined in terms of the JPDFs $f_j(D_{\min}, D_{\max})$ at the children n_j of node n_i. The JPDFs for all nodes in the PTT are, therefore, computed using a single topological traversal of the clock tree starting at the leaf nodes of the tree. After the JPDF of $D_{\min}$ and $D_{\max}$ is computed for the root of the tree, we compute the skew distribution using the integral in (10). Below, we first discuss how the JPDFs are computed in a PTT tree and then how the final clock skew distribution is computed.

1) Computing the JDPF of $D_{\min}$ and $D_{\max}$: We consider a parent node n_i with two children n_{j1} and n_{j2}, and edges e_1 and e_2, as shown in Fig. 4. Given the JPDF $f_{j1}(D_{\min}, D_{\max})$ at node n_{j1} and $f_{j2}(D_{\min}, D_{\max})$ at n_{j2}, and the edge delay PDFs $f_{e1}(D_{e1})$ of edge e_1 and $f_{e2}(D_{e2})$ of edge e_2, we compute the JPDF $f_i(D_{\min}, D_{\max})$ at the parent node n_i using the following two operations.

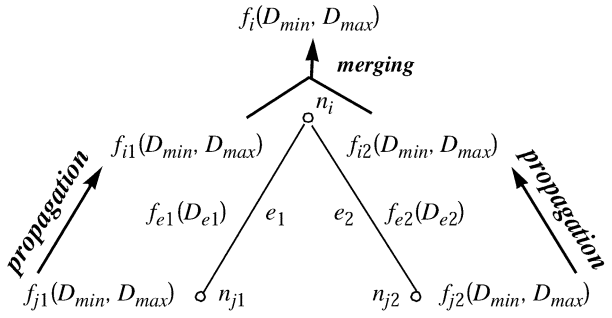

Fig. 4. Propagation and merging of JPDFs in a PTT.

a) Propagation: Propagation computes JPDF $f_{\text{prop}}(D_{\min}, D_{\max})$ of minimum and maximum delays of signals from the parent node to all its successors. The JPDF is propagated through edge between a child and the parent node. JCDF $F_{\text{prop}}(D_{\min,p}, D_{\max,p})$ can be expressed as the following integral:

$$\begin{aligned} &F_{\text{prop}}(D_{\min,p} D_{\max,p}) \\ &\quad = f_{\text{prop}}(D_{\min} \le D_{\min,p}, D_{\max} \le D_{\max,p}) \\ &\quad = \iint\limits_{D_{\min,j}+D_e \le D_{\min,p}, D_{\max,j}+D_e \le D\max,p} \int f_j(D_{\min}, D_{\max}) \\ &\qquad \cdot f_e(D_e) dD_{\min,j} dD_{\max,j} dD_e \end{aligned} \tag{11}$$

where $f_j(D_{\min}, D_{\max})$ is the JPDF of minimum and maximum delays at the child node and $f_e(D_e)$ is PDF of the edge delay. The JPDF $f_{\text{prop}}(D_{\min}, D_{\max})$ can be computed by differentiating this formula as shown in (7).

We compute the JPDFs using discretized functions. Each of the JPDFs $f_{j1}(D_{\min}, D_{\max})$ and $f_{j2}(D_{\min}, D_{\max})$ computed at a child node n_{j1} and n_{j2} is propagated to parent node n_i along the respective edges e_1 and e_2 to obtain the JPDFs $f_{i1}(D_{\min}, D_{\max})$ and $f_{i2}(D_{\min}, D_{\max})$. For node n_{j1} this is performed by enumeration of all possible triples $(D_{\min,j1}, D_{\max,j1}, D_{e1})$ of minimum and maximum path delays and edge delay, corresponding to JPDF $f_{j1}(D_{\min}, D_{\max})$ at node n_{j1} and the delay PDF $f_{e1}(D_{e1})$ of edge e_1. Initially, the JPDF $f_{i1}(D_{\min}, D_{\max})$ at node n_i is initialized with zero for all combinations of $D_{\min}, D_{\max}$. Then, for each enumerated triplet, we compute the minimum and maximum path delay at node n_i by adding the edge delay to the path delay at node n_{j1} : $D_{\min,i1} = D_{\min,j1} + D_{e1}$ and $D_{\max,i1} = D_{\max,j1} + D_{e1}$.

From our assumption that all edge delays are independent random variables, it follows that the edge delay random variable D_{e1} is independent from random variables $D_{\min}$ and $D_{\max}$ at node n_{j1}. Therefore, the probability of occurrence of triplet $(D_{\min,j1}, D_{\max,j1}, D_{e1})$ is

$$\Pr\{D_{\min,j1}, D_{\max,j1}, D_{e1}\} = \Pr\{D_{\min,j1}, D_{\max,j1}\} \cdot \Pr\{D_{e1}\} \tag{12}$$

```
1. Initialize f_i1(D_min, D_max) to zero for all D_min, D_max
2. For each D_min,j1 from {0,1,2,....j} {
3.   For each D_max,j1 from {0,1,2,....k} {
4.    For each D_e1 from {0,1,2,....l} {
5.       D_min,i1 = D_min,j1 + D_e1
6.       D_max,i1 = D_max,j1 + D_e1
7.       f_i1(D_min, D_max) = f_i1(D_min, D_max) +
         f_j1(D_min, D_max) * f_e1(D_e1)
8.    }
9.   }
10. }
```

Fig. 5. JPDF Propagation Algorithm.

```
1. Initialize f_i(D_min, D_max) to zero for all D_min, D_max
2. For each D_min,i1 from {0,1,2,....j} {
3.   For each D_max,i1 from {0,1,2,....k} {
4.    For each D_min,i2 from {0,1,2,.....l} {
5.     For each D_max,i2 from {0,1,2,....m} {
6.          D_min,i = min (D_min,i1, D_min,i2)
6.          D_max,i = max (D_max,i1, D_max,i2)
7.          f_i(D_min, D_max) = f_i(D_min, D_max) +
            f_i1(D_min, D_max) * f_i2(D_min, D_max)
8.    }
9.   }
10.  }
11. }
```

Fig. 6. Algorithm for Merging JPDFs.

which can be computed directly from the JPDF $f_{j1}(D_{\min}, D_{\max})$ and PDF $f_{e1}(D_{e1})$. The probability value of the $f_{i1}(D_{\min}, D_{\max})$ of the JPDF at node n_i is then incremented with the probability of the occurrence of the triplet $(D_{\min,j1}, D_{\max,j1}, D_{e1})$. The same calculation is performed for node n_{j2} to compute the JPDF $f_{i2}(D_{\min}, D_{\max})$. The computation is shown in pseudocode in Fig. 5. This can also be represented as a discrete equation, as follows:

$$f_{i1}(D_{\min,j1} + D_{e1}, D_{\max,j1} + D_{e1}) = \sum_{D_{\min,j1}=0}^{j} \sum_{D_{\max,j1}=0}^{k} \sum_{D_{e1}=0}^{l} \Pr\{D_{\min,j1}, D_{\max,j1}\} \cdot \Pr\{D_{e1}\}. \quad (13)$$

b) Merging: Using the two propagated JPDFs $f_{i1}(D_{\min}, D_{\max})$ and $f_{i2}(D_{\min}, D_{\max})$, we compute the JPDF $f_i(D_{\min}, D_{\max})$ at node n_i. The JCDF $F_i(D_{\min,i}, D_{\max,i})$ can be expressed by the following integral. (See the equation at the bottom of the page.) JPDF $f_i(D_{\min}, D_{\max})$ can be obtained from the above formula by differentiating it according to (7). We compute the JPDF using discretized functions. This is performed by enumerating all possible quadruplets $(D_{\min,i1}, D_{\max,i1}, D_{\min,i2}, D_{\max,i2})$ of minimum and maximum path delays to n_i corresponding $f_{i1}(D_{\min}, D_{\max})$ and $f_{i2}(D_{\min}, D_{\max})$. Again, we first initialize the JPDF $f_i(D_{\min}, D_{\max})$ at node n_i with zero for all combinations of $D_{\min}, D_{\max}$. For each quadruplet, we then compute the minimum and maximum path delays at node n_i: $D_{\min,i} = \min(D_{\min,i1}, D_{\min,i2})$ and $D_{\max,i} = \max(D_{\max,i1}, D_{\max,i2})$. Since the two JPDFs at node n_{j1} and n_{j2} are computed in a bottom up fashion, they are completely determined by the delays of the subtrees rooted at the nodes n_{j1} and n_{j2}. Since these two subtrees are by definition disjoint, (meaning they do not share any edges) it is clear that the random variables $D_{\min}$ and $D_{\max}$ at n_{j1} are independent with respect to random variables $D_{\min}$ and $D_{\max}$ at n_{j2}. Also, the two edge delays D_{e1} and D_{e2} are independent random variables. Therefore, the probability of occurrence of quadruplet $(D_{\min,i1}, D_{\max,i1}, D_{\min,i2}, D_{\max,i2})$ is

$$\Pr\{D_{\min,i1}, D_{\max,i1}, D_{\min,i2}, D_{\max,i2}\} = \Pr\{D_{\min,i1}, D_{\max,i1}\} \cdot \Pr\{D_{\min,i2}, D_{\max,i2}\} \quad (14)$$

which can be obtained directly from the JPDF $f_{i1}(D_{\min}, D_{\max})$ and JPDF $f_{i2}(D_{\min}, D_{\max})$. The value of the JPDF $f_i(D_{\min}, D_{\max})$ at node n_i is then incremented with the probability of the occurrence of quadruplet $(D_{\min,i1}, D_{\max,i1}, D_{\min,i2}, D_{\max,i2})$. The computation is shown in pseudocode in Fig. 6. This can also be represented as a discrete equation as follows:

$$f_i\big(\min(D_{\min,i1}, D_{\min,i2}), \max(D_{\max,i1}, D_{\max,i2})\big) = \sum_{D_{\min,i1}=0}^{j} \sum_{D_{\max,i1}=0}^{k} \sum_{D_{\min,i2}=0}^{l} \sum_{D_{\max,i2}=0}^{m} Pr\{D_{\min,i1}, D_{\max,i1}\} \cdot \Pr\{D_{\min,i2}, D_{\max,i2}\}. \quad (15)$$

Note that if node n_i has more than two children, the merging procedure is iteratively repeated, each time merging a propagated JPDF from a new child node with the JPDF resulting from the merging operation of already processed children.

By repeating the propagation and merging operations in a hierarchical fashion during a bottom-up traversal of the PTT, the JPDFs of $D_{\min}$ and $D_{\max}$ are computed for all nodes in the tree. The complexity of the algorithm is linear with the number of edges in the clock tree, since each edge in the tree requires exactly one propagation and merge operation. In terms of the dis-

$$F_i(D_{\min,i}, D_{\max,i}) = f_i(D_{\min} \le D_{\min,i}, D_{\max} \le D_{\max,i}) = \int\int_{\min(D_{\min,i1}, D_{\min,i2}) \le D_{\min}, \max(D_{\max,i1}, D_{\max,i2}) \le D_{\max}} \int f_{i1}(D_{\min}, D_{\max}) \times f_{i2}(D_{\min}, D_{\max}) dD_{\min,i1} dD_{\max,i1} dD_{\min,i2} dD_{\max,i2}$$

cretization, the complexity of the analysis is $O(n^3)$ for the propagation operation and $O(n^4)$ for the merging operation, where n is the number of discretizations of the edge delay PDF in each of the two dimensions of the JPDF. Since the merging operation has the highest computational complexity in terms of the number of discretization, we propose a more efficient method for merging two JPDFs in Section IV, which reduces the complexity of the merging operation to $O(n^2)$.

It is important to note that the size of n increases as we propagate JPDFs up the tree. Therefore, the JPDFs must be pruned as they are propagated. However, in our benchmark testing presented in Section IV, it was not necessary to perform pruning, as the clock trees in consideration had a small number of levels of logic. Also, the efficiency of the algorithm can be improved by exploiting the fact that all the arrays of JPDFs have nonzero values only above their diagonals. This allows reduction in memory consumption by a factor of two. The constant of proportionality for the runtime complexity is reduced by a factor of two for the propagation procedure and by a factor of four for the merging procedure.

Also, the merging operation is simplified for nodes in the tree whose children are leaf nodes. The JPDF $f_{i1}(D_{\min}, D_{\max})$ propagated from a leaf node is equal to the edge-delay probability $f_{e1}(D_{e1})$ of the leaf edge for values $D_{\min} = D_{\max} = D_{e1}$, and is zero for all values $D_{\min} \neq D_{\max}$. This allows the enumeration for the merge operation to be simplified from enumerating quadruplets to enumerating triplets $(D_{e1}, D_{\min,i2}, D_{\max,i2})$ if child node n_{j1} is a leaf node, or enumerating only pairs (D_{e1}, D_{e2}) if both children of n_i are leaf nodes. The complexity of merging, therefore, reduces to $O(n^3)$ or $O(n^2)$ for processing leaf nodes of the PTT. In practice, most nodes of a clock tree are leaf nodes, which improves the runtime of the algorithm.

2) Efficient Merging Procedure: Since the merge operation has the highest complexity in terms of the number of discretizations, we introduce a new procedure based on precomputation of JCDFs and marginal JCDFs to improve the computational complexity.

We consider the computation of JPDF $f_i(D_{\min}, D_{\max})$ at node n_i by merging two JPDFs $f_{i1}(D_{\min}, D_{\max})$ and $f_{i2}(D_{\min}, D_{\max})$. From the merging procedure presented in the previous section, it follows that for each possible quadruplet of minimum/maximum path lengths $(D_{\min,i1}, D_{\max,i1}, D_{\min,i2}, D_{\max,i2})$ the resulting path delay values $D_{\min}$ and $D_{\max}$ at n_i are as follows:

$$D_{\min} = \min(D_{\min,i1}, D_{\min,i2}) \tag{16}$$

$$D_{\max} = \max(D_{\max,i1}, D_{\max,i2}). \tag{17}$$

From this, it follows that $D_{\min} \leq D_{\min,i1}, D_{\min} \leq D_{\min,i2}$ and similarly that $D_{\max} \geq D_{\max,i1}, D_{\max} \geq D_{\max,i2}$. In addition, we have the following inequalities: $D_{\min} \leq D_{\max}, D_{\min,i1} \leq D_{\max,i1}, D_{\min,i2} \leq D_{\max,i2}$. From (16) and (17), it is clear that either $D_{\min} = D_{\min,i1}$ or $D_{\min} = D_{\min,i2}$ and $D_{\max} = D_{\max,i1}$ or $D_{\max} = D_{\max,i2}$. Also, we consider that $D_{\min} = D_{\min,i2}$ if $D_{\min,i1} = D_{\min,i2}$ and similarly, $D_{\max} = D_{\max,i2}$ if $D_{\max,i1} = D_{\max,i2}$. The resulting JPDF $f_i(D_{\min}, D_{\max})$ can be computed by considering the following four mutually exclusive cases for $D_{\min,i1}, D_{\max,i1}, D_{\min,i2}, D_{\max,i2}$, and their probabilities:

$$\begin{aligned} \text{case I}: D_{\min} &= D_{\min,i1} \\ D_{\max} &= D_{\max,i1} \\ D_{\min} &< D_{\min,i2} < D_{\max,i2} < D_{\max} \end{aligned} \tag{18}$$

$$\begin{aligned} \mathrm{P}_1 = {} & f_{i1}(D_{\min}, D_{\max}) \\ & \cdot f_{i2}(D_{\min} < D_{\min,i2}, D_{\max,i2} < D_{\max}) \end{aligned} \tag{19}$$

$$\begin{aligned} \text{case II}: D_{\min} &= D_{\min,i1} \\ D_{\max} &= D_{\max,i2} \\ D_{\max,i1} &\leq D_{\max} \\ D_{\min} &< D_{\min,i2} \end{aligned} \tag{20}$$

$$\begin{aligned} \mathrm{P}_2 = {} & f_{i1}(D_{\min}, D_{\max,i1} \leq D_{\max}) \\ & \cdot f_{i2}(D_{\min} < D_{\min,i2}, D_{\max}) \end{aligned} \tag{21}$$

$$\begin{aligned} \text{case III}: D_{\min} &= D_{\min,i2} \\ D_{\max} &= D_{\max,i2} \\ D_{\min} &\leq D_{\min,i1} \leq D_{\max,i1} \leq D_{\max} \end{aligned} \tag{22}$$

$$\mathrm{P}_3 = f_{i1}(D_{\min} \leq D_{\min,i1} \leq D_{\max}) \cdot f_{i2}(D_{\min}, D_{\max}) \tag{23}$$

$$\begin{aligned} \text{case IV}: D_{\min} &= D_{\min,i2} \\ D_{\max} &= D_{\max,i1} \\ D_{\max,i2} &< D_{\max} \\ D_{\min} &\leq D_{\min,i1} \end{aligned} \tag{24}$$

$$\begin{aligned} \mathrm{P}_4 = {} & f_{i1}(D_{\min} \leq D_{\min,i1}, D_{\max}) \\ & \cdot f_{i2}(D_{\min}, D_{\max,i2} < D_{\max}). \end{aligned} \tag{25}$$

Based on the four mutually exclusive cases identified above, we can obtain the following expression for JPDF $f_i(D_{\min}, D_{\max})$:

$$\begin{aligned} f_i(D_{\min}, D_{\max}) &= P_1 + P_2 + P_3 + P_4 \\ &= f_{i1}(D_{\min}, D_{\max}) \\ &\quad \cdot f_{i2}(D_{\min} < D_{\min,i2}, D_{\max,i2} < D_{\max}) \\ &\quad + f_{i1}(D_{\min}, D_{\max,i1} \leq D_{\max}) \\ &\quad \cdot f_{i2}(D_{\min} < D_{\min,i2}, D_{\max}) \\ &\quad + f_{i1}(D_{\min} \leq D_{\min,i1}, D_{\max,i1} \leq D_{\max}) \\ &\quad \cdot f_{i2}(D_{\min}, D_{\max}) \\ &\quad + f_{i1}(D_{\min} \leq D_{\min,i1}, D_{\max}) \\ &\quad \cdot f_{i2}(D_{\min}, D_{\max,i2} < D_{\max}) \end{aligned} \tag{26}$$

where each term corresponds to each of the cases I–IV in (18)–(24). Each of the nontrivial probability expressions in the first two terms of (26) can be expressed with the following summations over the discretized JPDFs, with a discretization unit of Δ_d:

$$\begin{aligned} & f_{i2}(D_{\min} < D_{\min,i2}, D_{\max,i2} < D_{\max}) \\ &= \sum_{D_{\max,i2,k}=0}^{(D_{\max}-\Delta_d)/\Delta_d} \sum_{D_{\min,i2,k}=(D_{\min}+\Delta_d)/\Delta_d}^{\infty} \\ &\quad f_{i2}(D_{\min,i2,k} \cdot \Delta_d, D_{\max,i2,k} \cdot \Delta_d) \end{aligned} \tag{27}$$

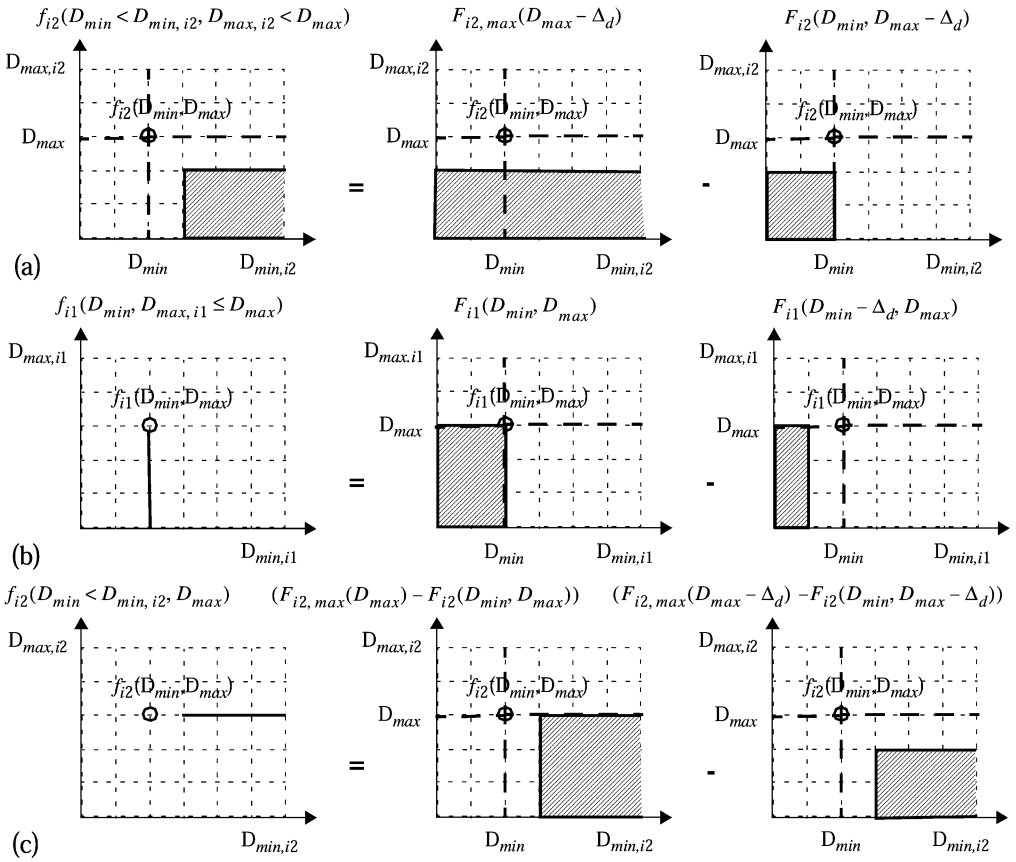

Fig. 7. JPDF computation using JCDFs and marginal CDFs (a) (30), (b) (31), (c) (32).

$$f_{i1}(D_{\min}, D_{\max,i1} \le D_{\max}) = \sum_{D_{\max,i1,k}=0}^{D_{\max}/\Delta_d} f_{i1}(D_{\min}, D_{\max,i1,k} \cdot \Delta_d) \quad (28)$$

$$f_{i2}(D_{\min} < D_{\min,i2}, D_{\max}) = \sum_{D_{\min,i2,k}=(D_{\min}+\Delta)/\Delta_d}^{\infty} f_{i2}(D_{\min,i2,k} \cdot \Delta_d, D_{\max}) \quad (29)$$

with similar expressions for the nontrivial probability terms in the last two terms of (26). Note that since the expressions necessary to compute (26) involve at most double summations, (26) can be computed with $O(n^2)$ computational complexity, where n is the number of discretizations of the JPDFs f_{i1} and f_{i2}. However, the computation can be further improved in efficiency by precomputing the following probability functions:

1) The JCDFs F_{i1} and F_{i2}, where

$$F_{i1}(D_{\min}, D_{\max}) = f_{i1}(t_{\min} \le D_{\min}, t_{\max} \le D_{\max}) = \sum_{t_{\min,k}=0}^{D_{\min}/\Delta_d} \sum_{t_{\max,k}=0}^{D_{\max}/\Delta_d} f_{i1}(t_{\min,k} \cdot \Delta_d, t_{\max,k} \cdot \Delta_d) \quad (30)$$

and F_{i2} is expressed similarly.

```
1. Initialize f(s) to zero for all s
2. For each D_min from {0,1,2,....j} {
3.    For each D_max from {0,1,2,....k} {
6.       s = D_max - D_min
7.       f(s) = f(s) + f_i(D_min, D_max)
8.    }
9. }
```

Fig. 8. Skew computation algorithm.

2) The marginal PDFs $f_{i1,\min}, f_{i1,\max}, f_{i2,\min}$, and $f_{i2,\max}$, where

$$f_{i1,\min}(D_{\min}) = \sum_{t_{\max,k}=0}^{\infty} f_{i1}(D_{\min}, t_{\max,k} \cdot \Delta_d) \quad (31)$$

and $f_{i1,\max}, f_{i2,\min}$, and $f_{i2,\max}$ computed similarly.

3) The marginal CDFs $F_{i1,\min}, F_{i1,\max}, F_{i2,\min}$, and $F_{i2,\max}$ where

$$f_{i1,\min}(D_{\min}) = f_{i1,\min}(t_{\min} \le D_{\min}) = \sum_{t_{\min,k}=0}^{D_{\min}/\Delta_d} f_{i1,\min}(t_{\min,k} \cdot \Delta_d) \quad (32)$$

and $F_{i1,\max}, F_{i2,\min}$, and $F_{i2,\max}$ can be computed similarly.

TABLE I
RESULTS OF OUR ALGORITHM AND MONTE CARLO

Circuit					Monte Carlo		Our Algorithm		% error		run time (s)		
tree no.	#sink nodes	# levels	avg # fanouts	max # fanouts	mean (ps)	99% pt. (ps)	mean (ps)	99% pt. (ps)	mean (ps)	99% pt. (ps)	Our Approach	Efficient Method	Imprv. factor
T1	16	6	1.4	2	35.98	59.69	35.91	59.47	-0.194	-0.368	5	1	5
T2	120	7	1.7	6	57.59	81.02	57.47	79.99	-0.208	-1.250	18	3	6
T3	1200	5	1.9	50	61.64	78.22	61.61	78.07	-0.048	-0.191	20	4	5
T4	2400	5	1.9	100	63.43	78.91	63.24	78.72	-0.315	-0.240	42	7	6
T5	4800	5	1.9	200	66.98	82.98	66.75	82.69	-0.343	-0.349	80	14	5.7
T6	6000	5	2	250	66.47	83.07	66.33	81.63	-0.210	-1.733	90	18	5
T7	12000	5	2	500	69.90	85.38	69.46	84.40	-0.629	-1.140	180	30	6

We can now express $f_{i2}(D_{\min} < D_{\min,i2}, D_{\max,i2} < D_{\max})$ in (27) in terms of JCDF F_{i2} and marginal CDF $F_{i2,\max}$ as illustrated in Fig. 7 as follows:

$$f_{i2}(D_{\min} < D_{\min,i2}, D_{\max,i2} < D_{\max}) = F_{i2,\max}(D_{\max} - \Delta_d) - F_{i2}(D_{\min}, D_{\max} - \Delta_d). \quad (33)$$

Also, (28) and (29) can be computed as follows:

$$f_{i1}(D_{\min}, D_{\max,i1} \le D_{\max}) = F_{i1}(D_{\min}, D_{\max}) - F_{i1}(D_{\min} - \Delta_d, D_{\max}) \quad (34)$$

$$f_{i2}(D_{\min} < D_{\min,i2}, D_{\max}) = \big(F_{i2,\max}(D_{\max}) - F_{i2}(D_{\min}, D_{\max})\big) - \big(F_{i2,\max}(D_{\max} - \Delta_d) - F_{i2}(D_{\min}, D_{\max} - \Delta)\big) \quad (35)$$

where the other summations necessary to compute (26) can be expressed similarly.

Hence, we precompute JCDFs and marginal CDFs from the JPDF $f_{i1}(D_{\min}, D_{\max})$ and $f_{i2}(D_{\min}, D_{\max})$ once, and then compute all terms in (26) directly from these JCDFs and marginal CDFs avoiding repeated numerical summation and improving the computational efficiency. By following the above method, the computational complexity of the merging operation is reduced to $O(n^2)$, where n is the number of discretizations of the JPDFs f_{i1} and f_{i2}. However, the computational complexity of propagation remains $O(n^3)$, and hence the overall computational complexity in terms of the number of discretizations is reduced from $O(n^4)$ to $O(n^3)$. The runtime complexity in terms of the number of edges in the PTT remains linear.

3) Computing the Skew-Probability Distribution: Once the JPDF of $D_{\min}$ and $D_{\max}$ is computed at the root node of the PTT, it can be used for computing the probability distribution of clock skew. We simply enumerate all possible pairs $(D_{\min}, D_{\max})$ of the JPDF and for each pair compute the associated skew $s = D_{\max} - D_{\min}$. We then update the probability of occurrence of this skew with the probability of occurrence of the pair $(D_{\min}, D_{\max})$. The algorithm is shown in pseudocode in Fig. 8. The complexity of the algorithm is $O(n^2)$ where n is the number of discretizations.

Since a JPDF of $D_{\min}$ and $D_{\max}$ is obtained for all nodes in the PTT during the bottom-up traversal, it is possible to compute the skew distribution for individual subtrees in the PTT with minimal runtime overhead. This allows the designer to compare the variability of different parts of a clock tree, which can be helpful to determine which parts are most prone to process variations. The proposed algorithms, therefore, provide not only a way for predicting the expected clock skew in manufactured die, but also a means to guide the designer in improving the robustness of the clock tree to process variations.

IV. EXPERIMENTAL RESULTS

The proposed method for statistical clock-skew computation was implemented and tested on a number of clock-tree benchmark circuits, including a large industrial clock tree from an industrial high-performance microprocessor in 130-nm technology. The other clock-tree benchmark circuits were synthesized with varying numbers of levels and sinks to examine the operation of the algorithm under different configurations. Gate-delay PDFs with standard deviation of 10%–15% of the mean delay were used. Gaussian distributions truncated at their 3σ points were used for the PDFs. The number of discretizations to represent the delay PDFs was ten for the performed experiments. We also implemented Monte Carlo simulation to obtain the skew distribution for comparison with our proposed method.

The results for the proposed algorithm and Monte Carlo simulation are shown in Table I. Columns 2 and 3 show the number of sink nodes and the number logic levels for the tree, respectively. Columns 4 and 5 show the average and maximum number of fanouts for the tree. The industrial test case is circuit T7 with 12 000 sink nodes and a maximum fanout of 500. Columns 6 and 7 show the mean and 99% confidence point of the computed skew using Monte Carlo simulations and columns 8 and 9 show these values using our proposed algorithm. The 99% confidence point is the skew value corresponding to the 99% yield point on the CDF of clock skew, and signifies the maximum skew for the best 99% of the manufactured dies. Columns 10 and 11 show the percent error for the mean and 99% confidence points obtained by our approach and Monte Carlo simulation. Approximately 10 000 simulations were used to achieve a good accuracy with Monte Carlo simulation at the 99% confidence point. The maximum error is negligible, demonstrating the correctness of the proposed approach. In column 12, the runtime in seconds for our algorithm is shown, which includes parsing the benchmarks, generating PDFs for the edge delays, bottom up propagation of JPDFs and skew computation. Column 13 shows the improved

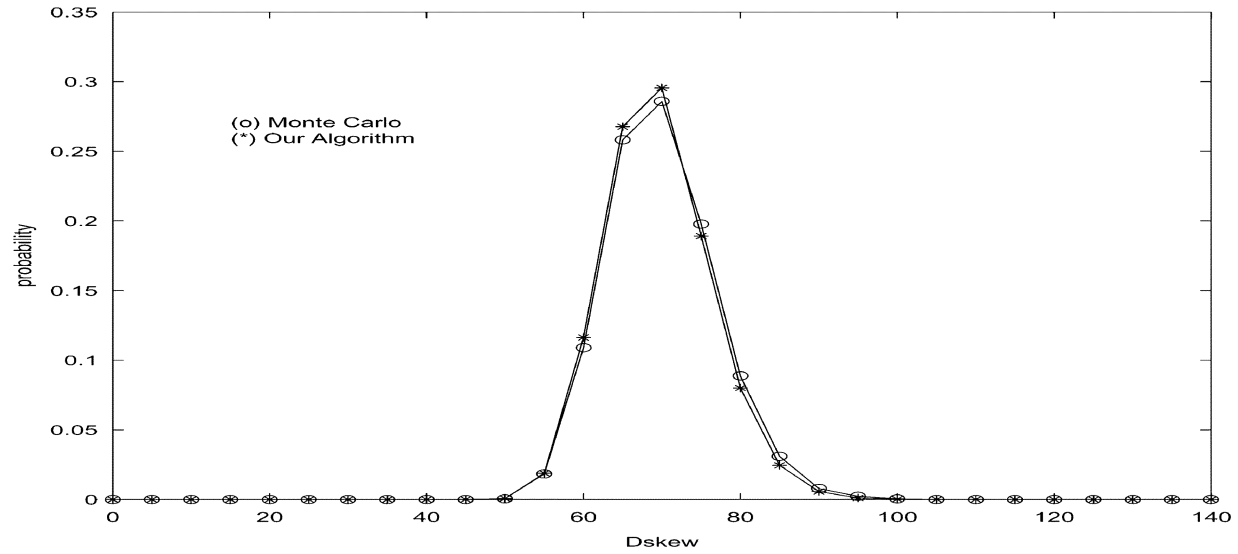

Fig. 9. PDF of skew for clock tree T7.

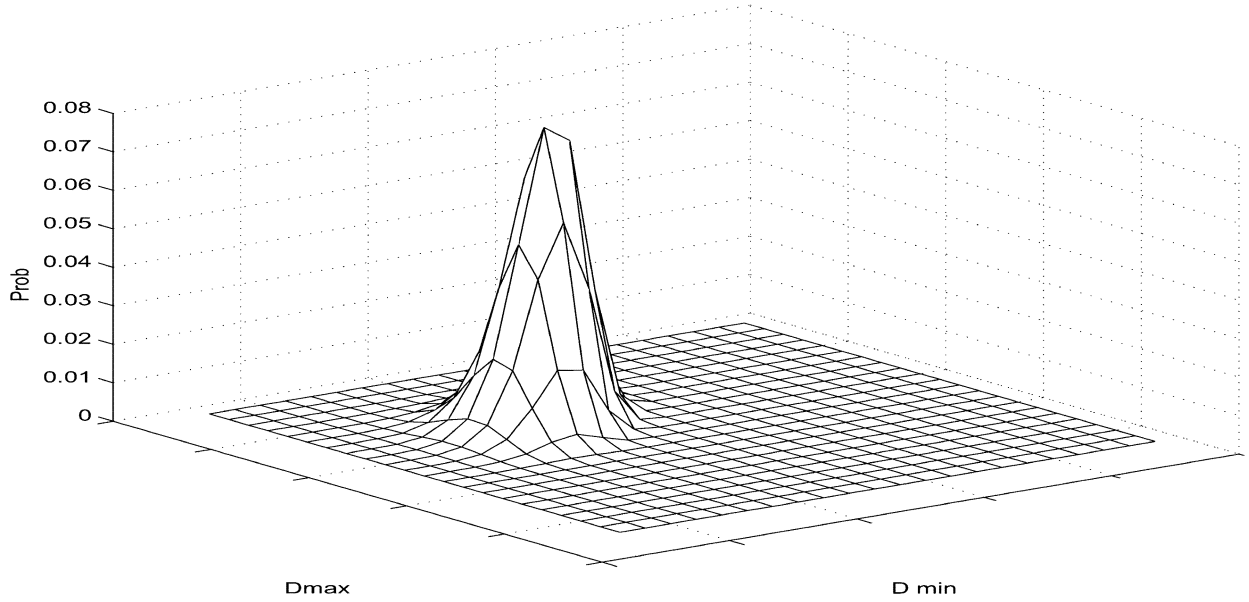

Fig. 10. JPDF of $D_{\min}$ and $D_{\max}$ for clock tree T7.

runtime using our efficient merging procedure, and column 14 shows the improvement factor. Fig. 9 shows a plot of the skew PDF, while Fig. 10 shows a three-dimensional representation of the JPDF of $D_{\min}$ and $D_{\max}$ at the root of clock tree of T7. We also performed Monte Carlo simulations for the same amount of time as our algorithm, to see how accurate Monte Carlo results are as compared to our approach. For the industrial test-case clock tree T7, the mean of the skew PDF obtained by Monte Carlo had a small error of 0.5%, although the 99% confidence point had an error of 4.7%. The errors were computed with reference to Monte Carlo simulation results which were allowed to converge, requiring 10 000 simulations typically. This confirms the usefulness of our approach as compared to Monte Carlo simulation.

In Table II, we show a comparison between our algorithm, worst-case skew analysis and traditional case analysis. In worst-case skew analysis, a deterministic delay is assigned to each gate within its $\pm 90\%$ or $\pm 99\%$ confidence point range. The delay of each gate is independently chosen from this range, such that the total skew of the clock tree is maximized. The results shown in columns 3 and 8 demonstrate that worst-case skew analysis can significantly overestimate the likelihood of skew, with overestimates ranging from 5% to over 100%. In traditional case analysis, we again perform a deterministic analysis but this time we use case-files and all gates are set at their 90% or 99% delay values. The results shown in columns 5 and 10 demonstrate that traditional case analysis is highly optimistic since it ignores the mismatch between drivers due to intradie-process variations.

V. Conclusion

In conclusion, we have presented a method for modeling the effects of process variations on clock skew. We have shown

TABLE II
RESULTS OF OUR ALGORITHM, WORST-CASE AND TRADITIONAL CASE ANALYSIS

Ckt	Analysis for 90% confidence point (ps)					Analysis for 99% confidence point (ps)				
tree no.	Our algorithm	Worst-Case skew analysis	%error	Traditional case analysis	%error	Our algorithm	Worst-Case skew analysis	%error	Traditional case analysis	%error
T1	46.38	105	126.39	1.604	-96.54	59.47	130	118.59	5.108	-91.41
T2	67.74	130	91.91	2.470	-96.35	79.99	160	100.02	8.106	-89.86
T3	68.67	85	23.78	1.733	-97.47	78.07	100	28.09	5.769	-92.61
T4	69.58	75	7.78	1.668	-97.60	78.72	100	27.03	5.382	-93.16
T5	73.59	85	15.50	1.634	-97.77	82.69	100	20.93	5.407	-93.46
T6	73.01	80	9.57	1.615	-97.78	81.63	100	22.50	5.565	-93.18
T7	75.72	80	5.65	1.619	-97.86	84.40	95	12.55	5.034	-94.03

how the distribution of the clock skew can be efficiently obtained from the JPDF of minimum and maximum clock-tree delay. We proposed an algorithm which is linear with circuit size, and demonstrated efficiency of the algorithm. We verified the correctness of our algorithm by comparing with Monte Carlo simulations. We also compared our statistical approach with worst-case skew analysis and traditional case analysis and demonstrated the importance of statistical analysis of clock-tree skew.

REFERENCES

[1] J. Cong, A. B. Kahng, C. K. Koh, and C.-W. A. Tsao, "Bounded-skew clock and Steiner routing," in *ACM Trans. Design Automation Electron. Syst.*, vol. 3, 1998, pp. 341–388.

[2] T. H. Chao, Y. C. Hsu, J. M. Ho, K. D. Boese, and A. B. Kahng, "Zero skew clock routing with minimum wirelength," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 799–814, Nov. 1992.

[3] B. Lu, J. Hu, G. Ellis, and H. Su, "Process variation aware clock tree routing," in *Proc. IEEE/ACM Int. Symp. Phys. Design*, 2003, pp. 174–181.

[4] I. M. Liu, T. L. Chou, A. Aziz, and D. F. Wong, "Zero-skew clock tree construction by simultaneous routing, wire sizing and buffer insertion," in *Proc. IEEE/ACM Int. Symp. Phys. Design*, Apr. 2000, pp. 33–38.

[5] S. Pullela, N. Menezes, and L. T. Pillage, "Reliable nonzero skew clock trees using wire width optimization," in *Proc. Design Automation Conf.*, 1993, pp. 165–170.

[6] R. S. Tsay, "An exact zero-skew clock routing algorithm," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 242–249, Feb. 1993.

[7] S. Lin and C. K. Wong, "Process-variation-tolerant clock skew minimization," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1994, pp. 284–288.

[8] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," in *Proc. ACM/IEEE Design Automation Conf.*, June 1993, pp. 612–616.

[9] G. Bai, S. Bobba, and I. N. Hajj, "Static timing analysis including power supply noise effect on propagation delay in VLSI circuits," in *Proc. IEEE/ACM Design Automation Conf.*, 2001, pp. 295–300.

[10] M. Zhao, K. Gala, V. Zolotov, Y. Fu, R. Panda, R. Ramkumar, and B. Agarwal, "Worst case clock skew under power supply variations," *Proc. TAU*, pp. 22–28, Dec. 2002.

[11] S. Nassif, "Delay variability: Sources, impacts and trends," in *Proc. ISSCC*, 2000.

[12] A. Kahng and Y. Pati, "Subwavelength optical lithography: Challenges and impacts on physical design," in *Proc. Int. Symp. Phys. Design*, 1999, pp. 112–119.

[13] D. Boning and J. Chung, "Statistical metrology-tools for understanding spatial variation," in *Proc. SPIE Symp. Microelectron. Manufact.*, Oct. 1996, pp. 16–26.

[14] M. Orshansky, L. Milor, P. Chen, K. Keutzer, and C. Hu, "Impact of systematic spatial intra-chip gate length variability on performance of high-speed digital circuits," in *Proc. Int. Conf. Computer-Aided Design*, 2000, pp. 62–67.

[15] V. Mehrotra, S. L. Sam, D. Boning, A. Chandrakasan, R. Vallishayee, and S. Nassif, "A methodology for modeling the effects of systematic within-die interconnect and device variation on circuit performance," in *Proc. Design Automation Conf.*, 2000, pp. 172–175.

[16] Y. Liu, S. Nassif, L. T. Pileggi, and A. J. Strojwas, "Impact of interconnect variations on the clock skew of a gigahertz microprocessor," in *Proc. Design Automation Conf.*, 2000, pp. 168–171.

[17] E. Malavasi, S. Zanella, M. Cao, J. Uschersohn, M. Misheloff, and C. Guardiani, "Impact analysis of process variability on clock skew," in *Proc. Int. Symp. Quality Electron. Design*, 2002, pp. 129–132.

[18] X. Jiang and S. Horiguchi, "A probabilistic approach to modeling skews and the largest delays of general clock distribution networks," in *Proc. ACM/IEEE Int. Workshop Timing Issues Specification Synthesis Dig. Syst. (TAU) 2000*, Dec. 2000, pp. 21–26.

[19] X. Jiang and S. Horiguchi, "Statistical skew modeling for general clock distribution networks in presence of process variations," *IEEE Trans. VLSI Systems*, vol. 9, pp. 704–717, Oct. 2001.

[20] D. Harris and S. Naffziger, "Statistical clock skew modeling with data delay variations," *IEEE Trans. VLSI Systems*, vol. 9, pp. 888–898, Dec. 2001.

[21] J. J. Liou, K. T. Cheng, S. Kundu, and A. Krstic, "Fast statistical timing analysis by probabilistic even propagation," in *Proc. Design Automation Conf.*, 2001, pp. 661–666.

[22] A. Gattiker, S. Nassif, R. Dinakar, and C. Long, "Timing yield estimation from static timing analysis," in *Proc. ISQED*, 2001, pp. 437–442.

[23] S. Devadas, H. F. Jyu, K. Keutzer, and S. Malik, "Statistical timing analysis of combinational circuits," in *Proc. ICCD*, 1992, pp. 38–43.

[24] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst-case timing analysis," in *Proc. Design Automation Conf.*, 2002, pp. 556–561.

[25] M. Berkelaar, "Statistical delay calculation, a linear time method," in *Proc. TAU*, 1997, pp. 15–24.

[26] A. Agarwal, D. Blaauw, S. Sundareswaran, V. Zolotov, M. Zhou, K. Gala, and R. Panda, "Statistical delay computation considering spatial correlations," in *Proc. Asia South Pacific Design Automation Conf.*, 2003, pp. 271–276.

[27] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computation and refinement of statistical bounds on circuit delay," in *Proc. Design Automation Conf.*, 2003, pp. 348–353.

[28] X. Bai, C. Visweswariah, P. N. Strenski, and D. J. Hathaway, "Uncertainty-aware circuit optimization," in *Proc. Design Automation Conf.*, 2002, pp. 58–63.

[29] J. A. G. Jess, K. Kalafala, S. R. Naidu, C. Visweswariah, and R. H. J. M. Otten, "Statistical timing for parametric yield prediction of digital integrated circuits," in *Proc. Design Automation Conf.*, 2003.

[30] L. Scheffer, "Explicit computation of performance as a function of process variation," in *Proc. TAU*, 2002.

[31] private communication Kerry Bernstein, IBM Corp., Burlington, VT, Personal Communication.

[32] W. P. Feller, *An Introduction to Probability Theory and Its Applications*. New York: Wiley, 1970, vol. 1.

Aseem Agarwal (S'03) received the B.E. degree in electronics and communication from Gujarat University, Ahmedabad, India, in 2001. Since August 2001, he is a Doctoral Student in Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor.

He is a Research Assistant in the Advanced Computer Architecture Lab, University of Michigan, working with Prof. D. T. Blaauw. His research focuses on timing analysis models and algorithms that consider process variability.

Vladimir Zolotov (M'97) received the Engineer degree from the Moscow Institute of Electronics, Moscow, Russia, and the Ph.D. degree from the Scientific Research Institute of Micro Devices, Moscow, Russia, both in electrical engineering.

He has been with the Advanced Tools Group, Motorola, Inc., Austin, TX, since 1998. He is involved in the development of EDA tools and methodology for high-performance VLSI designs. Previously, he was with the Moscow Research Laboratory, Motorola, Inc., Moscow, Russia. His research interests include signal integrity, reliability, on-chip inductance, timing analysis, and optimization of VLSI.

David T. Blaauw (M'01) received the B.S. degree in physics and computer science from Duke University, Durham, NC, in 1986 and the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana–Champaign, in 1988 and 1991, respectively.

He was with the Engineering Accelerator Technology Division, IBM Corporation, Endicott, NY, as a Development Staff Member, until August 1993. From 1993 to August 2001, he was with Motorola, Inc., Austin, TX, where he was the Manager of the High Performance Design Technology Group. Since August 2001, he has been a member of the faculty at the University of Michigan, Ann Arbor, as an Associate Professor. His work has focused on VLSI design and CAD with particular emphasis on circuit analysis and optimization problems for high-performance and low-power designs.

Prof. Blaauw was the Technical Program Chair and General Chair for the International Symposium on Low Power Electronics and Design, in 1999 and 2000, respectively, and was the Technical Program Co-Chair and member of the Executive Committee of the ACM/IEEE Design Automation Conference in 2000 and 2001.