Statistical Clock Skew Analysis Considering Intra-Die Process Variations

Aseem Agarwal, David Blaauw, *Vladimir Zolotov

University of Michigan, Ann Arbor, MI
*Motorola, Inc., Austin, TX

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 intra-die process variations and hence result in a very optimistic skew analysis. In this paper, we present a statistical skew analysis method to model intra-die process variations. We first present a formal model of the statistical clock skew problem and then propose an algorithm which is based on propagation of joint probability distribution functions in a bottom up fashion in a clock tree. The analysis accounts for topological correlations between path delays and has linear run time 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.

1 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][2][3]. 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 [4][5]

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 *inter-die* and *intra-die* variations. Inter-die variations represent differences in device characteristics from one die to the next, while intra-die variations represent differences in device characteristics within a single die. Intra-die variations can have a deterministic component due to topologically dependencies of device processing, such as CMP effects and optical proximity effects [6][7][8]. In some cases, such topological dependencies are directly accounted for in the analysis of clock skew, thereby reducing the statistical variation [9][10][11], whereas in other cases, such variations are treated as random. Furthermore, intra-die process variations exhibit spatial correlation, meaning that devices close to each other are more likely to have similar characteristics than those spaced far apart.

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 inter-die process variations. However, it cannot model intra-die 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, intra-die variations are becoming increasingly prominent and case-analysis is no longer valid. It is therefore critical that a statistical analysis of the clock skew be 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.

Recently, a number of methods for statistical clock skew analysis based on Monte Carlo simulation have been proposed [12]. However, Monte Carlo based approaches can have very high run times, especially for large clock designs. A probabilistic approach to clock skew analysis was proposed in [13][14] having efficient run time. However, this 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 propose a new approach to clock skew analysis, which accurately models intra-die process variations and has a linear run-time complexity with circuit size. Our analysis is focussed 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

ICCAD'03, November 11-13, 2003, San Jose, California, USA.
Copyright 2003 ACM 1-58113-762-1/03/0011 ...$5.00.

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 sensitive 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. Comparison with traditional case-analysis shows the importance of statistical clock skew analysis.

The remainder of this paper is organized as follows. In Section 2, we present the problem definition and modeling assumptions. In Section 3, we discuss our approach and implementation for statistical clock skew computation. In Section 4, we show experimental results and comparisons with Monte Carlo simulation. Finally, in Section 5, we draw our conclusions.

2 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 RLC 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 multi-driven meshes and can only be represented by directed acyclic graphs (DAGs). While such DAG clock networks cannot be modeled with our proposed approach, most clock networks, especially those in ASIC design, have a tree topology. Also, the proposed formulation forms a basis on which statistical skew analysis for DAG clock networks can be built.

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 Figure 1.

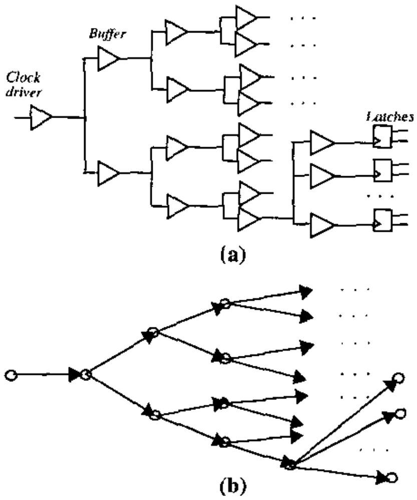

Figure 1. Clock tree and its timing tree representation

Each edge in the timing tree represents the delay from a driver gate input to an interconnect sink point and therefore represent 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 [15]. 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 [11]. Since timing trees are a special case of timing graphs, they inherent 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 $\{S_1,S_2,....S_n\}$.

$$d_{min} = min(d_{p,1}, d_{p,2} \,........., d_{p,n}) \qquad \text{(EQ 1)}$$

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.

$$d_{max} = max(d_{p,1}, d_{p,2} \,........., d_{p,n}) \qquad \text{(EQ 2)}$$

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

$$s = d_{max} - d_{min} \qquad \text{(EQ 3)}$$

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 *non-zero skew* clock tree. By setting non-zero 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 non-zero 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 non-zero skew clock trees with intentional skew targets.

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 $p_e(D_e)$. Although we formulate the clock skew problem using continuous probability density functions, we use discretized versions of these functions in our implementation, similar to those discussed in [16].

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 - 300μm [17]. 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 300μm. This therefore diminishes the impact of spatial correlation for typical clock tree designs. Also, spatial correlation of clock tree drivers tends to introduce positive correlation between the minimum and maximum path delays in a clock tree and hence, is expected to decrease the skew of a clock tree [14], making our analysis which ignores spatial correlation conservative. The current analysis also forms a frame work from which spatial correlations can be incorporated in future extensions.

Since all edge delays take a deterministic value on a manufactured die, the sample space consists of all possible die with different edge delay combinations. The probability that a manufactured die has a driver with a delay in interval $[d_1,d_2]$ is:

$$P(d_1 \le D_e \le d_2) = \int_{d1}^{d2} f(D_e)dD_e \qquad \text{(EQ 4)}$$

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 [20]. 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 probability density function (PDF) or the cumulative distribution function (CDF) of the clock skew, based on the PDF or CDF of the edge delays in the PTT.

3 Proposed approach for statistical skew computation

We start with a formal definition of the CDF of clock skew over the sample space of manufactured dice. 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 $p_{e,i}(t)$ for each edge i, which leads to the following expression for clock skew CDF:

$$P(S) = P(s \le S) = \qquad \text{(EQ 5)}$$

$$\int \cdots \int_{d_{max} - d_{min} \le S} p_{e,1}(t_1)\ p_{e,2}(t_2) \cdots\cdots\ p_{e,n}(t_n)\ dt_1 \ldots dt_n$$

where d_{min} and d_{max} are defined for a deterministic timing tree in EQ1 and EQ2.

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 deterministic timing tree, according to EQ1 through EQ3. Using one of several statistical timing analysis methods presented in [16][18][19][21], 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 for the DTT in EQ1 and EQ2, and attempt to compute their probability distributions by taking a statistical maximum and minimum over all path delays. 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 and maximum PTT delays 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 distributions function (JPDF) which preserves their correlation information. Furthermore, we show that by propagating the JPDF of minimum and maximum PTT delay in a single 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 run time complexity for merging from $O(n^4)$ to $O(n^2)$, where n is the number of discretizations in each dimension of the JPDF. Note that the run time complexity in terms of circuit size remains linear in all cases.

3.1 Computation of Clock Skew Distribution.

We now define the joint probability density function (JPDF) and the joint cumulative distribution function (JCDF) of minimum and maximum PTT delays and then show how the skew, as defined in EQ5 can be computed using such a JPDF. The JCDF of D_{min} and D_{max} is defined over the sample space as follows:

$$P(D_{min}, D_{max}) = P(min(d_{p,i}) \le D_{min}, max(d_{p,i}) \le D_{max}) \qquad \text{(EQ 6)}$$

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

$$p(D_{min}, D_{max}) = \frac{\partial^2}{\partial D_{min} \partial D_{max}} P(D_{min}, D_{max}) \qquad \text{(EQ 7)}$$

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 Figure 2, as a contour level 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:

$$P(D_{min,\,1} \le D_{min} \le D_{min,\,2}, D_{max,\,1} \le D_{max} \le D_{max,\,2}) \qquad \text{(EQ 8)}$$

$$= \int_{D_{min,\,1}}^{D_{min,\,2}} \int_{D_{max,\,1}}^{D_{max,\,2}} p(D_{min}, D_{max}) dD_{max} dD_{min}$$

We now write the expression of clock skew CDF in EQ5 in terms of the JPDF of minimum and maximum PTT delays as follows:

$$P(S) = P(s \le S) = \int \int_{D_{max} - D_{min} \le S} p(D_{min}, D_{max}) dD_{max} dD_{min}, \qquad \text{(EQ 9)}$$

EQ9 follows directly from the definition of the JPDF in EQ8 and the definition of clock skew in EQ5. Finally, we rewrite the integral in EQ9 above using simple manipulation of the integral limits as follows:

$$P(S) = P(s \le S) = \int_0^{\infty} \int_0^{D_{min} + s} p(D_{min}, D_{max}) dD_{max} dD_{min} \qquad \text{(EQ 10)}$$

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.

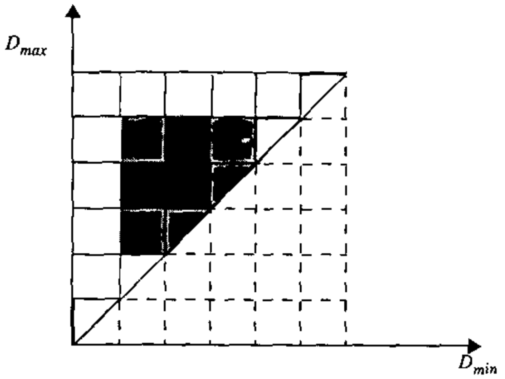

Figure 2. Joint distribution of D_{max}/D_{min} among a group of sinks

3.2 Joint Probability Distribution Computation.

We compute the JPDF $p(D_{min}, D_{max})$ for the minimum and maximum path delays in a PTT in a bottom up fashion. The JPDF $p_i(D_{min}, D_{max})$ 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 nodes of n_i. The JPDF $p_i(D_{min}, D_{max})$ at node n_i is defined in terms of the JPDFs $p_j(D_{min}, D_{max})$ 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 EQ10. Below, we first discuss how the JPDFs are computed in a PTT tree and then how the final clock skew distribution is computed.

Computing the JPDF 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 Figure 3. Given the JPDF $p_{j1}(D_{min}, D_{max})$ at node n_{j1} and $p_{j2}(D_{min}, D_{max})$ at n_{j2}, and the edge delay PDFs $p_{e1}(D_{e1})$ of edge e_1 and $p_{e2}(D_{e2})$ of edge e_2, we compute the JPDF $p_i(D_{min}, D_{max})$ at the parent node n_i using the following two operations:

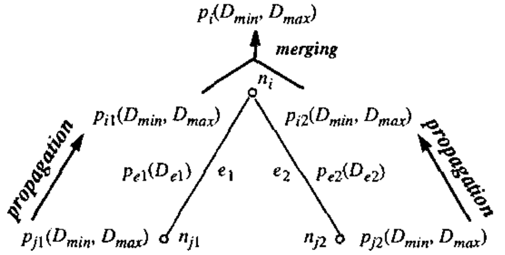

Figure 3. Propagation and merging of JPDFs in a PTT

1. **Propagation**: Each of the JPDFs $p_{j1}(D_{min}, D_{max})$ and $p_{j2}(D_{min}, D_{max})$ 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 $p_{i1}(D_{min}, D_{max})$ and $p_{i2}(D_{min}, D_{max})$. 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 $p_{j1}(D_{min}, D_{max})$ at node n_{j1} and the delay PDF $p_{e1}(D_{e1})$ of edge e_1. Initially, the JPDF $p_{i1}(D_{min}, D_{max})$ at node n_i is set to 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

$$P(D_{min,j1}, D_{max,j1}, D_{e1}) = P(D_{min,j1}, D_{max,j1}) \cdot P(D_{e1}) \quad \text{(EQ 11)}$$

which can be computed directly from the JPDF $p_{j1}(D_{min}, D_{max})$ and PDF $p_{e1}(D_{e1})$. The probability value of the $p_{i1}(D_{min,i1}, D_{max,i1})$ 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 $p_{i2}(D_{min}, D_{max})$. The computation is shown in pseudo code in Figure 4.

2. **Merging**: Using the two propagated JPDFs $p_{i1}(D_{min}, D_{max})$ and $p_{i2}(D_{min}, D_{max})$, we compute the JPDF $p_i(D_{min}, D_{max})$ at node $n_{i.}$ 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 to $p_{i1}(D_{min}, D_{max})$ and $p_{i2}(D_{min}, D_{max})$. Again, we first initialize the JPDF $p_i(D_{min}, D_{max})$ 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 JPDF 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:

$$P(D_{min,i1}, D_{max,i1}, D_{min,i2}, D_{max,i2}) = P(D_{min,i1}, D_{max,i1}) \cdot P(D_{min,i2}, D_{max,i2}) \quad \text{(EQ 12)}$$

which can be obtained directly from the JPDF $p_{i1}(D_{min}, D_{max})$ and JPDF $p_{i2}(D_{min}, D_{max})$. The value of the JPDF $p_i(D_{min,i}, D_{max,i})$ 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 pseudo code in Figure 5. 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 discretization, 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 the following Section 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, it is important to prune JPDFs as they are propagated. However, in our benchmark testing presented in Section 4 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 non-zero values only above their diagonals. This allows reduction in memory consumption by a factor of two. The constant of proportionality for the run time complexity is reduced by a factor of 2 for the propagation procedure by a factor of 4 for the merging procedure.

Also, the merging operation is simplified for nodes in the tree whose children are leaf nodes. The JPDF $p_{i1}(D_{min}, D_{max})$ propagated from a leaf node is equal to the edge delay probability $p_{e1}(D_{e1})$ 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 run time of the algorithm.

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 pre-computation of joint cumulative distribution functions (JCDFs) and marginal JCDFs to improve the computational complexity.

```
Initialize $p_{i1}(D_{min}, D_{max})$ to zero for all $D_{min}, D_{max}$
For each $D_{min,j1}$ from {0,1,2,.....m} {
  For each $D_{max,j1}$ from {0,1,2,....n} {
    For each $D_{e1}$ from {0,1,2,....p} {
      $D_{min,i1} = D_{min,j1} + D_{e1}$
      $D_{max,i1} = D_{max,j1} + D_{e1}$
      $p_{i1}(D_{min,i1}, D_{max,i1}) = p_{i1}(D_{min,i1}, D_{max,i1}) +$
      $p_{j1}(D_{min,j1}, D_{min,j1}) * p_{e1}(D_{e1})$
} } }
```

Figure 4. JPDF Propagation Algorithm

```
Initialize $p_i(D_{min}, D_{max})$ to zero for all $D_{min}, D_{max}$
For each $D_{min,i1}$ from {0,1,2,.....m} {
  For each $D_{max,i1}$ from {0,1,2,....n} {
    For each $D_{min,i2}$ from {0,1,2,.....p} {
      For each $D_{max,i2}$ from {0,1,2,....q} {
        $D_{min,i} = min(D_{min,i1}, D_{min,i2})$
        $D_{max,i} = max(D_{max,i1}, D_{max,i2})$
        $p_i(D_{min,i}, D_{max,i}) = p_i(D_{min,i}, D_{max,i}) +$
        $p_{i1}(D_{min,i1}, D_{max,i1}) * p_{i2}(D_{min,i2}, D_{max,i2})$
} } } }
```

Figure 5. Algorithm for Merging JPDFs

We consider the computation of JPDF $p_i(D_{min}, D_{max})$ at node n_i by merging two JPDFs $p_{j1}(D_{min}, D_{max})$ and $p_{j2}(D_{min}, D_{max})$. From the merging procedure presented in the previous section, it follows that for each possible quadruplet of minimum/maximum path lengths $(D_{min,1}, D_{max,1}, D_{min,2}, D_{max,2})$ the resulting path delay values D_{min} and D_{max} at n_i are as follows:

$$D_{min} = min(D_{min,1}, D_{min,2}), \qquad \text{(EQ 13)}$$

$$D_{max}=max(D_{max,1},D_{max,2}). \qquad \text{(EQ 14)}$$

From this it follows that $D_{min} \le D_{min,1}, D_{min} \le D_{min,2}$ and, similarly that $D_{max} \ge D_{max,1}, D_{max} \ge D_{max,2}$. In addition, we have the following inequalities: $D_{min} \le D_{max}$, $D_{min,1} \le D_{max,1}$, $D_{min,2} \le D_{max,2}$. From EQ13 and EQ14 it is clear that either $D_{min} = D_{min,1}$ or $D_{min} = D_{min,2}$ and $D_{max} = D_{max,1}$ or $D_{max} = D_{max,2}$. Thus, the resulted JPDF $p_i(D_{min}, D_{max})$ can be computed by considering the following four mutually exclusive cases for $D_{min,1}$, $D_{max,1}$, $D_{min,2}$, $D_{max,2}$ and their probabilities:

$$\text{case I: } D_{min} = D_{min,1}, D_{max} = D_{max,1}, \\ D_{min} < D_{min,2} < D_{max,2} < D_{max} \qquad \text{(EQ 15)}$$

$$P_1 = p_{j1}(D_{min}, D_{max}) \cdot p_{j2}(D_{min} < D_{min,2}, D_{max,2} < D_{max}) \qquad \text{(EQ 16)}$$

$$\text{case II: } D_{min} = D_{min,1}, D_{max} = D_{max,2}, \\ D_{max,1} \le D_{max}, D_{min} < D_{min,2} \qquad \text{(EQ 17)}$$

$$P_2 = p_{j1}(D_{min}, D_{max,1} \le D_{max}) \cdot p_{j2}(D_{min} < D_{min,2}, D_{max}) \qquad \text{(EQ 18)}$$

$$\text{case III: } D_{min} = D_{min,2}, D_{max} = D_{max,2}, \\ D_{min} \le D_{min,1} \le D_{max,1} \le D_{max} \qquad \text{(EQ 19)}$$

$$P_3 = p_{j1}(D_{min} \le D_{min,1}, D_{max,1} \le D_{max}) \cdot p_{j2}(D_{min}, D_{max}) \qquad \text{(EQ 20)}$$

$$\text{case IV: } D_{min} = D_{min,2}, D_{max} = D_{max,1}, \\ D_{max,2} < D_{max}, D_{min} \le D_{min,1} \qquad \text{(EQ 21)}$$

$$P_4 = p_{j1}(D_{min} \le D_{min,1}, D_{max}) \cdot p_{j2}(D_{min}, D_{max,2} < D_{max}) \qquad \text{(EQ 22)}$$

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

$$\begin{aligned} p_i(D_{min}, D_{max}) &= P_1 + P_2 + P_3 + P_4 = \\ &p_{j1}(D_{min}, D_{max}) \cdot p_{j2}(D_{min} < D_{min,2}, D_{max,2} < D_{max}) \\ &+ p_{j1}(D_{min}, D_{max,1} \le D_{max}) \cdot p_{j2}(D_{min} < D_{min,2}, D_{max}) \\ &+ p_{j1}(D_{min} \le D_{min,1}, D_{max,1} \le D_{max}) \cdot p_{j2}(D_{min}, D_{max}) \\ &+ p_{j1}(D_{min} \le D_{min,1}, D_{max}) \cdot p_{j2}(D_{min}, D_{max,2} < D_{max}) \end{aligned} \qquad \text{(EQ 23)}$$

where each term corresponds to each of the cases I through IV in EQ15 - EQ21. Each of the non-trivial probability expressions in the first two terms of EQ23 can be expressed with the following integrals:

$$p_{j2}(D_{min} < D_{min,2}, D_{max,2} < D_{max}) \qquad \text{(EQ 24)}$$

$$= \int_0^{D_{max}} \int_{D_{min}}^{\infty} p_{j2}(D_{min,2}, D_{max,2}) dD_{min,2} dD_{max,2}$$

$$p_{j1}(D_{min}, D_{max,1} \le D_{max}) = \int_{D_{min}}^{D_{max}} p_{j1}(D_{min}, D_{max,1}) dD_{max,1}, \qquad \text{(EQ 25)}$$

$$p_{j2}(D_{min} < D_{min,2}, D_{max}) = \int_{D_{min}}^{D_{max}} p_{j2}(D_{min,2}, D_{max}) dD_{min,2}, \qquad \text{(EQ 26)}$$

with similar integral expressions for the non-trivial probability terms in the last two terms of EQ23. Note that since the expressions necessary to compute EQ23 involve at most double integrals, EQ23 can be computed with $O(n^2)$ computational complexity, where n is the number of discretizations of the JPDFs p_{j1} and p_{j2}. However, the computation can be further improved in efficiency by precomputing the following probability functions:

1. The joint cumulative distribution functions (JCDFs) P_{j1} and P_{j2}, where:

$$P_{j1}(D_{min}, D_{max}) = P(t_{min} \le D_{min}, t_{max} \le D_{max}) = \qquad \text{(EQ 27)}$$

$$\int_0^{D_{max}} \int_0^{D_{min}} p_{j1}(t_{min}, t_{max}) dt_{min} dt_{max}$$

   and P_{j2} is expressed similarly.

2. The marginal PDFs $p_{j1,\text{min}}$, $p_{j1,\text{max}}$, $p_{j2,\text{min}}$, and $p_{j2,\text{max}}$, where:

$$p_{j1,min}(D_{min}) = \int_0^{\infty} p_{j1}(D_{min}, t_{max}) dt_{max}, \qquad \text{(EQ 28)}$$

   and $p_{j1,\text{max}}$, $p_{j2,\text{min}}$, and $p_{j2,\text{min}}$ computed similarly.

3. The marginal CDFs $P_{j1,\text{min}}$, $P_{j1,\text{max}}$, $P_{j2,\text{min}}$, and $P_{j2,\text{max}}$, where:

$$P_{j1,min}(D_{min}) = P(t_{min} \le D_{min}) = \int_0^{D_{min}} p_{j1,min}(t_{min}) dt_{min}, \qquad \text{(EQ 29)}$$

   and $P_{j1,\text{max}}$, $P_{j2,\text{min}}$, and $P_{j2,\text{max}}$ can be computed similarly.

We can now express $p_{j2}(D_{min} < D_{min,2}, D_{max,2} < D_{max})$ in EQ24 in terms of JCDF P_{j2} and marginal CDF $P_{j2,\text{max}}$ as follows:

$$p_{j2}(D_{min} < D_{min,2}, D_{max,2} < D_{max}) = \\ P_{j2,max}(D_{max}) - P_{j2}(D_{min}, D_{max}) \qquad \text{(EQ 30)}$$

where the other integrals necessary to compute EQ23 can be expressed similarly.

Hence, we precompute JCDFs and marginal CDFs from the JPDF $p_{j1}(D_{min,1}, D_{max,1})$ and $p_{j2}(D_{min,2}, D_{max,2})$ once, and then compute all terms in EQ23 directly from these JCDFs and marginal CDFs avoiding repeated numerical integration 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 p_{j1} and p_{j2}. 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 run time complexity in terms of the number of edges in the PTT remains linear.

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 pseudo code in Figure 6. 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 sub-trees in the PTT with minimal runtime overhead. This allows the designer to compare the expected 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.

4 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 number 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 point were used for the PDFs. We also implemented Monte Carlo simulation to obtain the skew distribution for comparison with our proposed method.

```
Initialize p(s) to zero for all s
For each $D_{min}$ from {0,1,2,.....m} {
    For each $D_{max}$ from {0,1,2,....n} {
        $s = D_{max} - D_{min}$
        $p(s) = p(s) + p_i(D_{min}, D_{max})$
} }
```

Figure 6. Skew Computation Algorithm

The results for the proposed algorithm and Monte Carlo simulation are shown in Table 1. Columns 2 and 3 show the number of sink nodes and the number logic levels for the tree, respectively. Column 4 shows 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 5 and 6 show the mean and 99% confidence value of the computed skew using Monte Carlo simulations and the proposed algorithm, while the last column shows the percent error between 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. The maximum error is negligible, demonstrating the correctness of the proposed approach. In column 7, the run-time 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. Figure 7 shows a plot of the skew CDF for clock tree T3 while Figure 8 shows a 3-D representation of the JPDF of D_{min} and D_{max} at the root of clock tree of T7.

In Table 2, 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 +/- 90% or +/- 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 are shown in column 3 and demonstrate that worst-case skew analysis can significantly overestimate the likelihood of skew, with overestimates ranging from 22 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 are shown in column 5 and demonstrate that case-analysis is highly optimistic since it ignores the mismatch between drivers due to intra-die process variation.

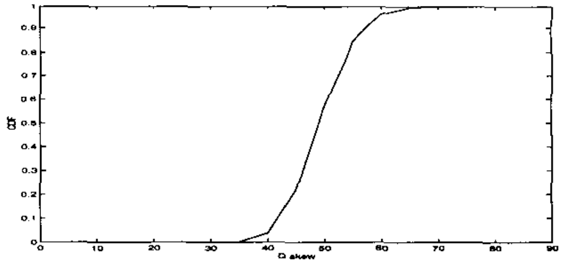

Figure 7. CDF of skew for clock tree T3

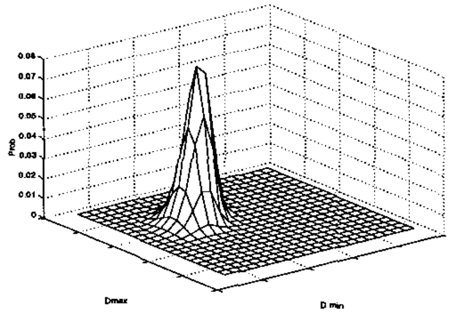

Figure 8. JPDF of D_{min} and D_{max} for clock tree T7

Table 1. Results of our algorithm and monte carlo

Circuit				Monte Carlo mean/ 99% pt. (ps)	Our Algorithm mean/ 99% pt. (ps)	run time (s)	% error of mean/ 99% pt.
tree no.	#sink nodes	# levels	avg/max fanouts				
T1	16	6	1.4/2	31.27/50.78	31.21/51.06	0.1	0.19/0.50
T2	120	7	1.7/6	49.82/69.29	49.76/68.92	1	0.12/0.53
T3	1200	5	1.9/50	51.86/64.16	51.93/64.29	3	0.13/0.20
T4	2400	5	1.9/100	52.45/64.19	52.43/64.15	6	0.04/0.07
T5	4800	5	1.9/200	53.96/64.99	53.90/65.25	8	0.11/0.39
T6	6000	5	2/250	53.20/64.54	53.25/64.67	14	0.09/0.20
T7	12000	5	2/500	54.45/65.39	54.49/65.38	28	0.08/0.02

Table 2. Results of our algorithm, worst-case and traditional case analysis

Ckt	Analysis for 90% / 99% confidence points (ps)				
tree no.	Our algorithm	Worst-Case skew analysis	%error	Traditional case analysis	%error
T1	39.73/51.06	100/105	152/106	1.70/2.22	-95.7/-95.6
T2	58.25/68.92	120/130	106/88.6	2.73/3.93	-95.3/-94.2
T3	57.44/64.29	75/85	30.5/32.2	2.05/3.39	-96.4/-94.7
T4	57.58/64.15	70/80	21.5/24.7	1.88/2.82	-96.7/-94.9
T5	59.02/65.25	75/85	27.0/30.2	1.95/3.26	-96.6/-95.0
T6	58.39/64.67	70/85	19.8/31.4	1.91/3.18	-96.7/-95.1
T7	59.27/65.38	70/80	18.1/22.3	1.84/2.82	-96.8/-95.6

5 Conclusions

In conclusion, we have presented a method for modelling the effects of process variations on clock skew. We have shown how the distribution of the clock skew can be efficiently obtained from the joint probability distribution function (JPDF) of minimum and maximum clock tree delay. We proposed an algorithm which is linear with circuit size, and demonstrated the 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 case-analysis and demonstrated the importance of statistical analysis of clock tree skew.

Acknowledgements

This research was supported by SRC, NSF, Intel and IBM.

References

[1] J. Cong, A. B. Kahng, C. K. Koh, C.-W. Albert Tsao, "Bounded-Skew Clock and Steiner Routing," ACM Trans. on Design Automation of Electronic Systems, vol. 3, pp. 341-388, 1998.

[2] T. H. Chao, Y. C. Hsu, J. M. Ho, K. D. Boese, A. B. Kahng, "Zero Skew Clock Routing With Minimum Wirelength", IEEE Trans. on Circuits and Systems 39(11), November 1992, pp. 799-814.

[3] B. Lu, J. Hu, G. Ellis, H. Su, "Process Variation Aware Clock Tree Routing," Proc. IEEE/ACM International Symposium on Physical Design, ISPD, 2003

[4] G. Bai, S. Bobba, I.N. Hajj, "Static timing analysis including power supply noise effect on propagation delay in VLSI circuits," Proceedings IEEE/ACM Design Automation Conference, 2001, pp. 295 -300

[5] M. Zhao, K. Gala, V. Zolotov, Y. Fu, R. Panda, R. Ramkumar, B. Agarwal, "Worst Case Clock Skew Under Power Supply Variations", TAU 2002, December 2002

[6] S. Nassif, "Delay Variability: Sources, Impacts and Trends," Proceedings of ISSCC, 2000.

[7] A. Kahng, Y. Pati, "Subwavelength optical lithography: challenges and impacts on physical design," Proceedings of ISPD, 1999.

[8] D. Boning et. al., "Manufacturing yield, reliability and failure analysis," SPIE Symposium on Microelectronic Manufacturing, Oct. 1996.

[9] M. Orshansky, L. Milor, P. Chen, K. Keutzer, C. Hu, "Impact of systematic spatial intra-chip gate length variability on performance of high-speed digital circuits", ICCAD 2000, pp. 62 -67.

[10] V.Mehrotra, S.L.Sam, D.Boning, A.Chandrakasan, R.Vallishayee, S.Nassif "A methodology for modelling the effects of systematic within-die interconnect and device variation on circuit performance. DAC 2000.

[11] Y.Liu, S.Nassif, L.T.Pileggi,A.J.Strojwas, "Impact of Interconnect Variations on the Clock Skew of a Gigahertz Microprocessor", DAC 2000.

[12] E. Malavasi, S. Zanella, M. Cao, J. Uschersohn, M. Misheloff, C. Guardiani, "Impact analysis of process variability on clock skew," Proceedings International Symposium on Quality Electronic Design, 2002, Page(s): 129 -132

[13] X. Jiang, S. Horiguchi, "A Probabilistic Approach to Modeling Skews and the Largest Delays of General Clock Distribution Networks," ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (Tau) 2000, Dec. 2000, Pages: 21-26

[14] X. Jiang, S. Horiguchi, "Statistical skew modeling for general clock distribution networks in presence of process variations," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 9 Issue: 5, Oct 2001. Page(s): 704 -717.

[15] D.Harris, S.Naffziger, "Statistical Clock Skew Modeling with Data Delay Variations", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 9 Issue: 6, Dec. 2001,Page(s): 888 -898

[16] J.J Liou, K.T. Cheng, S. Kundu, A. Krstic, "Fast Statistical Timing Analysis By Probabilistic Even Propagation", DAC 2001

[17] Personal communication, Kerry Bernstein, IBM Corp., Burlington, VT.

[18] M. Orshansky, K. Keutzer, "A general probabilistic framework for worst-case timing analysis", Proc. DAC 2002.

[19] M. Berkelaar, "Statistical Delay Calculation, a Linear Time Method," Proc. of TAU, 1997.

[20] Feller, W., P. "An Introduction to Probability Theory and its Applications", Vol. 1,2 John Wiley & Sons, New York, 1970.

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