A Methodology to Improve Timing Yield in the Presence of Process Variations

Sreeja Raj, Sarma B. K. Vrudhula, Janet Wang
NSF Center for Low Power Electronics, ECE Dept., University of Arizona.
sreeja@ece.arizona.edu, sarma@ece.arizona.edu, wml@ece.arizona.edu

ABSTRACT

The ability to control the variations in IC fabrication process is rapidly diminishing as feature sizes continue towards the sub-100 nm regime. As a result, there is an increasing uncertainty in the performance of CMOS circuits. Accounting for the worst case values of all parameters will result in an unacceptably low *timing yield*. *Design for Variability*, which involves designing to achieve a given level of confidence in the performance of ICs, is fast becoming an indispensable part of IC design methodology. This paper[1] describes a method to identify certain paths in the circuit that are responsible for the spread of timing performance. The method is based on defining a *disutility* function of the gate and path delays, which includes both the means and variances of the delay random variables. Based on the moments of this disutility function, an algorithm is presented which selects a subset of paths (called *undominated* paths) as being most responsible for the variation in timing performance. Next, a statistical gate sizing algorithm is presented, which is aimed at minimizing the delay variability of the nodes in the selected paths subject to constraints on the critical path delay and the area penalty. Monte-Carlo simulations with ISCAS '85 benchmark circuits show that our statistical optimization approach results in significant improvements in timing yield over traditional deterministic sizing methods.

Categories and Subject Descriptors: B.8.2 Performance Analysis and Design Aids.

General Terms: Algorithms, Performance, Design.

Keywords: Timing Analysis, Timing Yield, Gate Sizing.

1. INTRODUCTION

The performance of CMOS ICs is becoming increasingly unpredictable due to a significant increase in the variability of the process parameters [3]. The sources of uncertainty that cause deviations from nominal performance values are the variations in the fabrication process. These disturbances can cause a large spread in the various performance measures such as speed, power, etc. The parametric yield is the number of fabricated chips whose performance indices lie within a specified acceptable range. Assuming worst-case values of all the key device parameters can often result in an unacceptably low parametric yield.

There is a need to modify the existing worst-case design methodology to account for process variations [19]. However, this requires incorporation of statistical techniques to predict the parametric yield [9], as well as methods to optimize the design so that its performance falls within the acceptable region with a specified probability. When the performance metric is circuit delay, the fraction of chips whose delay is at or below a specified value is referred to as the *timing yield*. Some of the more recent works on timing yield estimation appear in [8, 9, 19]. The approach described in [8] assumes that path delays are Gaussian random variables. The means and variances of the delays of critical paths are computed as functions of the variations of the node parameters on each path. This is used to obtain a conservative estimate of the timing yield of the critical paths of the circuit. In [9], timing yield is estimated by numerically integrating the joint *pdf* of the process parameters over the feasible region.

Probabilistic delay analysis on directed graphs has a long history [6]. Much of the earlier work was done in the context of project completion times in PERT networks, where completion times of individual jobs are modeled as random variables. One of the earliest works on computing the probability distribution function of the circuit delay appears in [13]. This is an extremely difficult problem as it involves the maxima of a large number of dependent random variables. The solution presented in [13] is a method to compute a bound on the upper tail probability of circuit delay. The method requires solving an optimization problem that is not practical even for moderate size circuits.

The recent works [1, 2, 4, 10, 11] involve computation of circuit delay probabilities. In [10], the authors use symbolic simulation by representing the potentially longest paths as symbolic delay expressions. Simulation based approaches are presented in [4, 11]. In [4], the probability distribution of longest path delay is obtained by performing timing simulation on the k-most dominant deterministic critical paths. In [11], a new delay model and sensitization criterion are proposed, and the probability distribution of circuit delay is obtained by performing Monte Carlo simulations on the sensitizable longest paths. More recently, in [1], an efficient method for propagation of bounds on the circuit delay *pdf* is presented. Another approach to do the same using Bayesian networks is described in [2].

In this paper, we present a different approach to identify certain paths (called *undominated* paths) that are responsible for the spread of timing performance. The method is based on defining a *disutil-*

[1]The work was supported in part by the NSF Center for Low Power Electronics (Grant #EEC-9523338), by NSF ITR Grant (Grant #CCR-0205227), and by the NSF Center Connection One (Grant #EEC-0333046).

DAC 2004, June 7–11, 2004, San Diego, California, USA.

ity function of gate and path delays, which includes both means and variances of the delay random variables. Using the disutility function, an algorithm is presented to identify the undominated paths which are most responsible for the variation in timing performance. Next, a statistical gate sizing algorithm is presented, which is aimed at minimizing simultaneously, the means and variances of the delays of nodes in the selected paths subject to constraints on delay and area penalty.

The rest of the paper is organized as follows. Notation and terminology used in the paper appears in Section 2. An explanation of utility functions is given in Section 3. The method to identify the *undominated* paths is described in Section 4. In Section 5, the formulation of the statistical gate sizing problem and the method of solution is presented. Experimental results are presented in Section 6. Concluding remarks are in Section 7.

2. NOTATION & TERMINOLOGY

A circuit is modeled as a PERT network which is a directed acyclic graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes representing the gates of the circuit, and $\mathcal{A}$ is the set of arcs representing the interconnection between the gates. The start node is denoted by s and the sink node is denoted by t. The weight on each node i is its delay, which is denoted by $d_i \sim N(\eta_i, \sigma_i^2)$. A path p is an ordered set of adjacent arcs in the network, and $|p|$ denotes the cardinality of this set. The delay of a path p is denoted as D_p.

$\mathcal{P}$**:** The set of all paths from node s to node t.

$\mathcal{P}_{ij}$**:** The set of all paths from node i to node j. $\mathcal{P}_{i*}$ denote the set of all paths starting at i, and $\mathcal{P}_{*i}$ denote the set of all paths terminating at i.

C_{ij}**:** Covariance between delays of nodes d_i, d_j.

τ_i**:** Intrinsic delay of node i.

s_i**:** Sizing factor of the gate represented by node i, given by the ratio of aspect-ratio of the transistors in the gate to the aspect-ratio of unit sized transistor.

l_i, u_i**:** Lower and upper bounds of the size of node i.

$E(\cdot)$**:** Expected value.

U_p**:** Disutility function of delay of path p. Also, U_i denotes disutility of delay of node i .

$p + q$**:** Concatenation of paths p and q.

$fanins_i$**:** Set of all nodes j such that $a_{ji} \in A$.

$fanouts_i$**:** Set of all nodes j such that $a_{ij} \in A$.

$FI(i)$**:** Set of all nodes in the fanin cone of node i.

$FO(i)$**:** Set of all nodes in the fanout cone of node i.

$\mathcal{P}_U^i$**:** Set of all undominated paths terminating at i.

γ_{ij}**:** $\min_{p_{ij} \in \mathcal{P}_{ij}} \mid p_{ij} \mid$.

3. MOTIVATION FOR UTILITY THEORY

The limitation associated with statistical timing analysis is the complexity involved in computing the exact distribution of longest path delay. Also, in order to find the critical nodes in the circuit that are responsible for yield loss, the *stochastically longest* paths have to be identified. This requires ordering of random variables. The strongest ordering is *stochastic ordering* [18], which is based on the comparison of the probability distribution functions (*pdf*'s). For example, if X and Y are two random variables, then X is stochastically greater than Y, denoted by $X \geq_{\text{st}} Y$, if $P\{X > u\} \geq P\{Y > u\}, \forall u \in (-\infty, \infty)$. Figure (1a) depicts stochastic ordering, where $X \geq_{\text{st}} Y$. For even simple graphs, it is practically impossible to order paths using this ordering due to dependencies among the paths. There are many other (weaker) orderings of random variables, such as *convex ordering*, *hazard-rate ordering*, and *likelihood ratio ordering* [18]. Figure (1b) describes *convex ordering*. If X and Y are two random variables, then X is convexly larger than Y if $\int_u^\infty P\{X > u\} \geq \int_u^\infty P\{Y > u\}$, $\forall u \in (-\infty, \infty)$. However, for any practical circuit graph, none of these can be efficiently established. It appears that it is not practical to use any of the ordering schemes of random variables to compare the random delays of the paths of a circuit. For this reason we investigated simpler techniques to analyze the spread/narrowness of circuit performance.

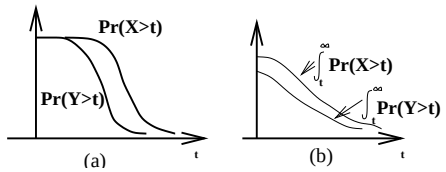

Figure 1: Stochastic and Convex Ordering.

Consider the *pdfs* shown in Figure (2). Intuitively, $f_2(X)$ is *narrower* or more concentrated at the mean as compared to $f_1(X)$. To formalize this notion and provide a framework that will allow us to compare designs based on variability or yield loss, we define a suitable function of the underlying random variables and use their moments to compare the random variables. These functions are called *utility* functions. The fundamental drawback is of course the difficulty in making precise probabilistic comparisons of the random variables. The intuitive justification for our approach is a set of axioms that imply the existence of utilities with the property that the expected utility is an appropriate measure for consistent decision-making [16]. One of the basic axioms is stated below.

AXIOM 1. *If an appropriate utility is assigned to each possible consequence and the expected utility of each alternative is calculated, then the best course of action is the alternative with the highest expected utility.*

With respect to the problem of choosing the path that causes high yield-loss, the choices for a utility function are continuous random variables (delay of the path), each characterized by a specific probability density function. Because yield-loss is an undesirable quality, we use the term *disutility* associated with each path.

DEFINITION 1. *The Disutility of a path is a function U_p of the delay along the path p. If each path is assigned the expected value of its disutilty function ($E(U_p)$), then the path that corresponds to maximum expected disutility is responsible for maximum timing yield loss.*

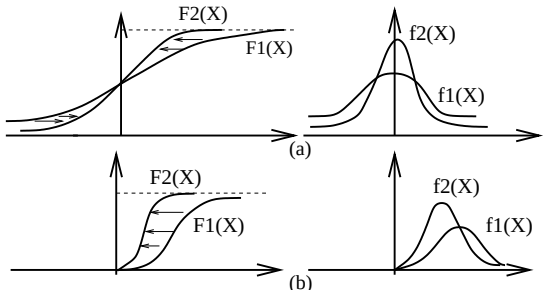

Figure 2: Comparison of spread/narrowness of random variables.

Two important properties of utility functions are monotonicity and risk.

Monotonicity: The disutility for the delay of a path should increase with delay. Therefore, we are interested in a monotonically increasing disutility function. That is, $[d_1 > d_2] \Leftrightarrow [U_1 > U_2]$.

Risk: The shape and the functional form of a utility function reveals a great deal about how much *risk* (*uncertainty*) is tolerated. A decision maker may be classified as being *Risk Averse*, *Risk Prone*, or *Risk Neutral*. A circuit designer trying to identify the paths that are responsible for yield-loss is bound to choose the paths that result in maximum uncertainty. Therefore, the decision maker in this scenario would be *risk prone* [16].

THEOREM 1. *For increasing utility functions, a decision maker is risk prone if and only if his utility function is convex [16].*

Based on the above axioms, we choose a quadratic function to assess the disutility of delay as the expected value of this function will involve both mean and variance of path delay. Higher order convex functions can be used as well, with a possible significant increase in computational cost.

DEFINITION 2. *Given a path p, with delay D_p, its disutility U_p is given by, $U_p = D_p^2 + D_p$.*

4. IDENTIFYING UNDOMINATED PATHS

Identifying the probabilistic shortest paths in PERT networks using concave quadratic utility functions was first discussed in [12]. Node delays are assumed to be uncorrelated. This problem is analogous to that of identifying the stochastic longest paths in a PERT network representing the circuit using a convex quadratic disutility, under the assumption that the gate delays are uncorrelated.

However, the assumption of independent gate delays is untenable. The delay of a gate depends on the capacitive load provided by the gates it drives. Also process variations lead to spatial/structural correlations. So we derive the expressions for expected disutility of path delay, given an arbitrary $N \times N$ covariance matrix describing the correlation between node delays.

Let $d_{1,p}, d_{2,p}, \ldots, d_{n,p}$ be the random variables that denote delay of the nodes on path p. Let $\eta_{i,p} = E(d_{i,p})$. The delay of a path p is the sum of delays of the nodes that appears in the path, i.e. $D_p = \sum_i d_{i,p}$. The expected value of D_p is $E(D_p) = \sum_i \eta_{i,p}$.

DEFINITION 3. *For two paths p and q, p is said to dominate q if $E(U_p) > E(U_q)$.*

DEFINITION 4. *A subset of paths $P_U \subseteq \mathcal{P}$ is called the set of undominated paths, if for every path $p \in P_U$ $E(U_p) > E(U_q)$ $\forall q \notin P_U$.*

The expected disutility of a path p is given by:

$$E(U_p) = E(D_p^2) + E(D_p). \tag{1}$$

The second moment of the delay of path p is given by

$$E(D_p^2) = \left(\sum_{i \in p} \eta_i\right)^2 + \sum_{i \in p} \sigma_i^2 + 2 \sum_{i,j \in p} C_{ij}. \tag{2}$$

Therefore, the expected disutility of a path can be expressed as a function of means, standard deviations and covariances of the delays of nodes that appear in the path.

$$E(U_p) = \left(\sum_{i \epsilon p} \eta_i\right)^2 + \sum_{i \epsilon p} \sigma_i^2 + 2 \sum_{i,j \in p} C_{ij} + \sum_{i \epsilon p} \eta_i. \tag{3}$$

In the deterministic longest path problem, at each intermediate node i, the delays along the paths in $\mathcal{P}_{si}$ are compared, and the critical paths responsible for maximum delay are propagated forward to the nodes $n \in fanouts_i$.

This procedure does not work in the statistical scenario. At any intermediate node, if a path is dominated by another path, it is not guaranteed that this dominance relation between the two remains valid until the node t is reached. For this reason, the notions of *temporary preference* and *permanent preference* are used [12].

DEFINITION 5. *If $p_1, p_2 \in \mathcal{P}_{si}$, then p_1 is said to be temporarily preferred over p_2 if $E(U_{p_1}) > E(U_{p_2})$.*

DEFINITION 6. *If $p_1, p_2 \in \mathcal{P}_{si}$, then p_1 is said to be permanently preferred over p_2 if $E(U_{p_1+p}) > E(U_{p_2+p})$, $\forall p \in \mathcal{P}_{it}$*

LEMMA 1. *Given two paths $p_1, p_2 \in \mathcal{P}_{si}$ in a PERT network, where the nodes are correlated, p_1 is permanently preferred over p_2, if*

$$\begin{aligned} &E(U_{p_1}) - E(U_{p_2}) + 2E(D_p)(E(D_{p_1}) - E(D_{p_2})) \\ &\quad +2\Big(\sum_{i \in p_1, j \in p} C_{i,j} - \sum_{i \in p_2, j \in p} C_{i,j}\Big) > 0 \ \ \forall p \in \mathcal{P}_{it}. \end{aligned} \tag{4}$$

Proof: From Definition (6), the conditions for permanent preference can be derived as follows:

$$\begin{aligned} &E\left((D_{p_1} + D_p)^2\right) + E\left(D_{p_1} + D_p\right) \\ &\quad > E\left((D_{p_2} + D_p)^2\right) + E\left(D_{p_2} + D_p\right) \ \ \forall p \in \mathcal{P}_{it}. \end{aligned} \tag{5}$$

This gives,

$$\begin{aligned} &\big(E(D_{p_1}^2) + E(D_p^2) + 2E(D_{p_1} D_p)\big) \\ &\quad + \big(E(D_{p_1}) + E(D_p)\big) \\ &\quad > \big(E(D_{p_2}^2) + E(D_p^2) + 2E(D_{p_2} D_p)\big) \\ &\quad + \big(E(D_{p_2}) + E(D_p)\big). \end{aligned} \tag{6}$$

In Equation (6), the expectations of products are given by

$$E(D_{p_1} D_p) = \sum_{i \in p_1, j \in p} C_{i,j} + E(D_{p_1}) E(D_p), \tag{7}$$

$E(D_{p_2} D_p)$ can be expressed in a similar manner. The condition for permanent preference given by Equation (4) follows from the above. ■

The brute force approach to establish permanent preference requires enumeration of all the paths $p \in \mathcal{P}_{it}$. This is not necessary if we prune the space by studying the circuit correlation structure. Various attributes of nodes are generally correlated. This correlation might be due to logical dependencies, or caused by the fabrication process. For example, the nominal delay of two otherwise independent gates might be correlated if they are placed close to each other. For this reason we identify a set of nodes that might be correlated with a given node. This is done so that propagation of path information to determine permanent preference need not proceed beyond the set of correlated nodes.

DEFINITION 7. *The correlation front of node i, denoted by $F_i(\alpha)$ is the set of nodes in the fanout cone of i that are farthest from i and have a covariance with any node in the fanin cone of i greater than α.*

The front identifies the set of nodes in any path that emanates out of node i and is strongly correlated to atleast one node in the paths that terminate at node i.

LEMMA 2. *Given two paths $p_1, p_2 \in \mathcal{P}_{si}$ in a PERT network, where the nodes are correlated, p_1 is permanently preferred over p_2, if*

$$\begin{aligned} E(U_{p_1+p_{il}}) &> E(U_{p_2+p_{il}}), \text{ and} \\ E(D_{p_2+p_{il}}) - E(D_{p_1+p_{il}}) &< \frac{E(U_{p_1+p_{il}}) - E(U_{p_2+p_{il}})}{2(\max_{p\in\mathcal{P}_{lt}} E(D_p))} \\ & \forall p_{il}, l \in F_i(\alpha). \end{aligned} \quad (8)$$

Proof: Any path $p \in \mathcal{P}_{it}$ will contain at least one node $l \in F_i(\alpha)$. Each path $p \in \mathcal{P}_{it}$ is partitioned into two sections, p_{il} and p_{lt}, such that the node $l \in p$ and $l \in F_i(\alpha)$, then paths p_1 and p_{lt} are uncorrelated, and so are paths p_2 and p_{lt}. To establish preference, it is to be shown that, $E(U_{p_1+p_{il}+p_{lt}}) > E(U_{p_2+p_{il}+p_{lt}})$. This can also be expressed as,

$$\begin{aligned} &E(U_{p_1+p_{il}}) - E(U_{p_2+p_{il}}) + 2E(D_{p_{lt}}) \\ &\quad (E(D_{p_1+p_{il}}) - E(D_{p_2+p_{il}})) > 0. \end{aligned} \quad (9)$$

Equation (8) follows from the above. ■

The algorithm begins at node s, and at each intermediate node i, computes $\mathcal{P}_{si}$ and $F_i(\alpha)$. Each pair of paths in $\mathcal{P}_{si}$ are compared to each other and if the conditions of Equation (8) are satisfied for each $p \in \mathcal{P}_{il}$, $\forall l \in F_i(\alpha)$, then permanent preference is established. The set of undominated paths are propagated forward to $fanouts_i$.

Algorithm (1) is the pseudo code for this procedure. A dynamic programming approach can be used for propagating the set of undominated paths from the start node to the sink node. Starting at node s, at each intermediate node i, we compare all the paths $p \in \mathcal{P}_{si}$ for permanent preference. Any dominated path is dropped from further consideration. Paths constituting $\mathcal{P}_U^i$ are propagated forward to the all $j \in fanouts_i$. At the node t, we are left with the set of *undominated* paths in the circuit. In the worst-case, the algorithm is exponential in complexity (if all the paths are retained as undominated). However, the ISCAS benchmark circuits were solved very efficiently. (See Table(1)).

Algorithm 1: Undominated Paths

```
UNDOMINATEDPATHS(G(N, A))
(1)  LIST ← LIST ∪{s};
(2)  while LIST!=NULL
(3)      i = Extract a node from LIST;
(4)      Paths(i) = ∅;
(5)      foreach (node j ∈ fanins_i)
(6)          foreach (path q ∈ P_U^j)
(7)              p ← q;
(8)              Append i to p;
(9)              Compute E(U_p);
(10)             Paths(i) ← Paths(i) ∪ p;
(11)     Compute F_i(α);
(12)     foreach (p ∈ Paths(i))
(13)         foreach (q ∈ Paths(i), q ≠ p)
(14)             if (p >_P q)
(15)                 ∀p_il, l ∈ F_i(α);
(16)             P_U^i ← P_U^i ∪ p;
(17)     foreach j ∈ fanouts_i
(18)         LIST← LIST∪j;
(19) end;
```

5. STATISTICAL GATE SIZING TO IMPROVE TIMING YIELD

Gate sizing involves optimal assignment of drive strength to individual gates of a circuit for a given objective function and constraints. The deterministic gate sizing problem is generally formulated to *minimize delay* subject to *area constraints* [7, 17]. The statistical gate sizing algorithm is aimed at minimizing the timing yield loss subject to constraints on the mean delay of each path in the circuit, and the total area consumed. Typically, deterministic gate sizing speeds up the nodes on the timing-critical paths in order to meet the timing constraints. In our algorithm, we minimize the disutility of the nodes that appear in the undominated paths that are responsible for timing yield loss of the circuit. To minimize the area overhead and to assign the hardware resources judiciously, a criticality index is assigned to each node, based on the frequency of its occurrence in the set of undominated paths. The criticality index of node i, CI_i, is the number of undominated paths that contain node i.

In the statistical gate delay model, the delay of a node is assumed to be a Gaussian random variable $d_i \sim N(\eta_i, \sigma_i^2)$. The mean delay η_i of node i is taken to be equal to its nominal delay. The expression for the nominal delay of a node i as a function of its size is given by Equation (10), in which τ_i is the intrinsic delay of the gate i, s_i is the sizing factor of node i, $\{s_j : j \in fanouts_i\}$ are the sizing factors of nodes in the fanout of node i, and constants c and c_j's relates the propagation delay to sizes of transistors in the gates. Therefore from Equation (10), $\eta_i = f_{\eta_i}(\tau_i, s_i, \{s_j : j \in fanouts_i\})$. This is to emphasize the fact that η_i is a function of τ_i, s_i, and $\{s_j : j \in fanouts_i\}$.

$$\eta_i = \tau_i + c\,\frac{\sum_{j\in fanouts(i)} c_j s_j}{s_i}. \quad (10)$$

Parametric variations are also functions of designable circuit parameters such as width and length of the transistors. A widely accepted model, known as *Pelgrom's model* [14, 15], relates parametric variations to circuit parameters. Generally, the standard deviation of a device parameter varies inversely with the size of the gate. Perlgrom's model states that $\sigma^2_{V_{TH}} = \frac{A_{V_{TH}}}{s}, \sigma^2_{T_{OX}} = \frac{A_{T_{OX}}}{s}, \sigma^2_W = \frac{A_W}{s}, \sigma^2_L = \frac{A_L}{s}$, where constants $A_{V_{TH}}$, $A_{T_{OX}}$, A_W and A_L are empirically determined and are specific to the fabrication process. Since internal and load capacitances are functions of the model parameters, the standard deviation of delay can be expressed as a function of the sizing factors of the driving gate and the gates being driven by the gate.

We can estimate the standard deviation of the Gaussian random variable corresponding to the gate delay [8] as

$$\begin{aligned} \sigma_i^2 &= \left(\frac{\partial d_i}{\partial L_i}\right)^2 \sigma^2_{L_i} + \left(\frac{\partial d_i}{\partial V_{TH_i}}\right)^2 \sigma^2_{V_{TH_i}} + \cdots \\ &\quad \sum_{j\in fanouts(i)} \left(\frac{\partial d_i}{\partial L_j}\right)^2 \sigma^2_{L_j} + \cdots \end{aligned} \quad (11)$$

The partial differential coefficients in Equation (11) are the sensitivities of delay to the variations in the model parameters. Therefore, from Pelgrom's model and (11), $\sigma_i^2 = f_{\sigma_i}(s_i, \{s_j : j \in fanouts_i\})$, showing the fact that σ_i^2 is a function of s_i and $\{s_j : j \in fanouts_i\}$.

The statistical sizing problem can now be expressed as a non-linear program given by Equation (12). The objective function is the sum of the expected disutilities of the nodes, each scaled by their criticality index. The criticality index for each node is computed from the set of undominated paths identified from the algorithm in Section (4). The mean delay of every path is constrained to a required time T_{req}. The expected utility, mean and variance of the delay of a node are expressed as non-linear functions of the sizes of the nodes. In the deterministic sizing algorithm, the objective is to minimize the critical delay of the circuit. But in the statistical case, the mean critical path delay and area penalty is constrained within imposed limits. The mean critical path delay is constrained by the optimal delay obtained by deterministic sizing, denoted by T_{req}. If the area used up in the deterministic solution is denoted by A, then the area penalty in statistical sizing is constrained to ΔA, given by the last constraint in Equation (12) which limits the total area used to $A + \Delta A$. The sizing factors of the nodes are constrained to be between 1 and 4 (l_i, u_i).

$$\begin{aligned} \textit{Minimize} \quad & \sum_{i \in N} CI_i E(U_i) \\ \textit{Subject To} \quad & \max_{p \in \mathcal{P}} E(D_p) \leq T_{req} \\ & E(U_i) = (\eta_i^2 + \sigma_i^2) + \eta_i \qquad \forall i \in N \\ & \eta_i = f_{\eta_i}(\tau_i, s_i, \{s_j : j \in fanouts_i\}) \\ & \sigma_i = f_{\sigma_i}(s_i, \{s_j : j \in fanouts_i\}) \\ & l_i \leq s_i \leq u_i \qquad \forall i \in N \\ & \text{Area} \ \leq A + \Delta A. \end{aligned} \tag{12}$$

We solve the deterministic and statistical sizing problems using a large scale non linear program solver called LANCELOT [5], to obtain the sizes of all the nodes in the network. The circuit yield from both the designs are compared using Monte-Carlo Analysis.

6. EXPERIMENTAL RESULTS

We tested the proposed sizing technique on the ISCAS '85 benchmark circuits. The nominal delays of the gates were computed using the standard cell library for the MOSIS TSMC CMOS 0.25μ process. The interconnect delays are not included in the analysis. However, the algorithm can be easily extended to incorporate interconnect delays by adding a node corresponding to each interconnect net in the PERT network. The standard deviation was computed using the predictions from [3], which presents the data extracted from ITRS'97 augmented with data from IBM processes. The constants of the Pelgrom's model and sensitivity of delay to the process parameters were assumed such that, the percentage of total variation of delay of gates is **15%** of its nominal value.

6.1 Obtaining Covariance Matrix

It is observed that spatial correlations due to process variations are higher for devices that are close to each other and lower for those far apart. Due to absence of actual process data describing the spatial/structural correlations, we assume that the correlation between two nodes i, j diminishes as the minimum number of edges connecting i and j, (γ_{ij}) increases. The PERT is assumed to have a k-degree correlation, i.e., $C_{ij} = 0$ for any two nodes $\{i, j \in N\} \ni \mathcal{P}_{ij} \neq \emptyset$ and $\gamma_{ij} > k$. Then, the correlation front of node i is the set of nodes farthest from i, such that $\gamma_{i,j} \leq k$. This gives the k-Front of node i, denoted by $F_i(k)$(See Figure(3)). Then, $F_i(\alpha)$ is replaced by $F_i(k)$ in Equation (8).

In our experiments we used a covariance matrix, where the values of C_{ij}'s are inversely proportional to the degree of separation

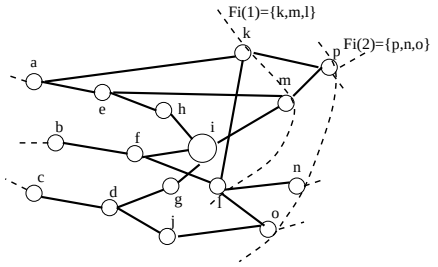

Figure 3: k-Front of node i, $F_i(1)$ and $F_i(2)$

γ_{ij}'s. In the experimental results, we present the analysis results for $k = 1$ case. The cases with higher values of k are not included as they did not significantly impact the yield improvements. Also, the complexity of the algorithm to identify the undominated paths increases with k. The circuit C6288 can be analyzed successfully only if it is assumed that the gate delays are not correlated i.e., $k = 0$.

6.2 Sizing using LANCELOT

LANCELOT was used to solve the non-linear programs corresponding to deterministic and statistical sizing methods for each circuit. In each case, the mean and standard deviation of the delay of every node was computed as a function of the gate sizes. Ten thousand runs of Monte Carlo simulations were performed using normal random number generators to randomly assign delays to each node in the circuit, and for each sample, the critical path delay of the circuit was computed. The timing yield at a delay value T^* is the number of samples that have a critical delay $\leq T^*$. The timing yield obtained from deterministic and statistical sizing are shown in the Table (2). Starting with minimal size devices, the deterministic sizing problem is solved to minimize the critical delay. If the solution to the deterministic sizing is denoted as T_{req} and area A, then three runs of statistical sizing is performed, with critical path delay constrained by T_{req} and area penalty constrained by A, $1.2A$ and $1.3A$. The value of delay at which the yield is 95 % is observed in both cases and reported as $t_{d,0.95}$ and $t_{s,0.95}$ for the deterministic and statistical cases respectively. The deterministic yield Y_d at $t_{s,0.95}$ is compared with statistical yield at $t_{s,0.95}$, $Y_s = 95\%$, to obtain the yield improvement. (see Figure 4).

From Table (2) we observe that a significant improvement in timing yield can be achieved by applying the statistical method of sizing as opposed to deterministic sizing. The method is robust, and gives consistent increase in yield improvement with increase in area penalty. We also observed that in most cases, the critical path delay distribution of the statistically sized circuit is convexly smaller (see Section(3)) than that of the deterministically sized circuit(see Figure (5)).

7. CONCLUSIONS

This paper presents a method based on the concept of utility theory to logically analyze the timing performance of the paths in the circuit. We developed an algorithm to identify the statistical critical paths responsible for timing yield loss using expected disutility of path delay as a measure of spread/narrowness of the delay random variable. We also proposed a yield optimization approach by sizing the gates in the circuit with the objective of minimizing the disutility of the critical paths with constraints on delay and area.

It has been shown using Monte Carlo simulations that there is a significant improvement in timing yield of the benchmark circuits when sized using the statistical method and constrained by

Table 1: Number of Undominated Paths

Circuit	C432	C499	C880	C1355	C1908	C2670	C3540	C5315	C7552
No: of undominated paths ($\|\mathcal{P}_\mathcal{U}\|$)	729	3072	190	196608	254	364	344	337	81
Total no: of paths ($\|\mathcal{P}\|$)	83926	9440	8642	4173216	729057	374270	28676671	1341305	726494

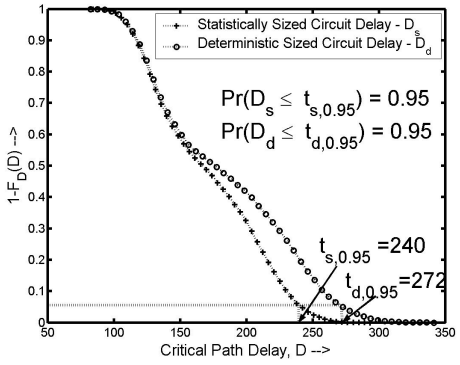

Figure 4: Comparison of Yield of C3540 for $\Delta A = 0.3$.

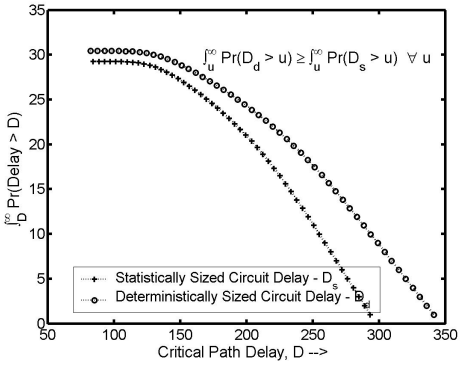

Figure 5: Convex Ordering shown in C3540

the delay and area penalty of the deterministic solution. Also, it is observed that additional improvement can be obtained by allowing a further area penalty of 20% or 30% of the deterministic solution.

This work paves the way for the application of more rigorous statistical methods for design in the nanometer regime, where design for variability is indispensable. Our future work will explore other performance issues such as power that are susceptible to variability, and yield optimization with multiple objective utility functions.

Table 2: Comparison of Yield

ISCAS circuit	$t_{d,0.95}$	$Y_d@t_{s,0.95}$ (%)			Time Improvement (%)	Yield Improvement (%)
		ΔA (%)	$t_{s,0.95}$	Y_d (%)		
C432	103	0	93	86.49	9.71	9.72
		20	92	85.33	10.68	12.5
		30	91	82.10	11.65	15.7
C499	99	0	88	88.56	11.11	7.24
		20	84	84.65	15.15	12.36
		30	83	83.42	16.16	13.86
C880	168	0	138	83.58	17.86	13.65
		20	135	82.20	19.64	15.61
		30	132	80.71	21.42	17.92
C1355	141	0	136	81.44	3.55	16.55
		20	131	76.73	7.09	23.78
		30	120	73.60	14.89	28.99
C1908	199	0	178	81.12	10.55	15.74
		20	172	75.75	13.57	25.49
		30	168	72.89	15.58	30.32
C2670	123	0	110	77.57	10.57	22.57
		20	100	67.04	18.69	41.42
		30	97	63.72	21.14	48.93
C3540	302	0	271	85.85	10.26	10.72
		20	252	82.84	16.56	14.65
		30	246	80.55	18.54	18.03
C5315	265	0	248	88.63	6.42	6.86
		20	242	85.27	8.68	11.22
		30	238	82.89	10.19	14.01
C7552	232	0	225	85.37	3.02	11.28
		20	212	82.65	8.62	15.37
		30	207	80.09	10.78	18.87

8. REFERENCES

[1] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula. *Statistical timing analysis using boounds and selective enumeration.* Workshop TAU pages 332–337, Dec 2002.

[2] S. Bhardwaj, S. Vrudhula, and D. Blaauw. *Tau: Timing analysis under uncertainty.* Proc. Int'l Conf. on Computer-Aided Design, 2003.

[3] D. Boning and S. Nassif. *Models of Process Variations in Device and Interconnect in Design of High-Performance Microprocessor Circuits.* A. Chandrakasan, 2000.

[4] R. B. Brashear, N. Menezes C. Oh, L.T. Pillage, M.R. Mercer. *Predicting circuit performance using circuit-level statistical timing analysis.* European Design and Test Conf., Mar 2002.

[5] A. R. Conn and N. I. M. Gould. *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization.* Springer-Verlag, 1992.

[6] S E. Elmaghraby. *Activity Networks: Project Planning and Control by Network Models*. John Wiley & Sons, 1977.

[7] J. Fishburn, A. Dunlop. *TILOS: A Posynomial Programming Approach to Transistor Sizigng*. Proc. Int'l Conf. on Computer-Aided Design, 1985.

[8] A. Gattiker, S. Nassif, R. Dinakar, and C. Long. *Timing yield estimation from static timing analysis.* Proc. Int'l. Symposium on Quality Electronic Design (ISQED), 2001.

[9] J. A. G. Jess, K. Kalafala, S. R. Naidu, R. H. J. M Otten, and C. Vishweswariah. *Statistical timing for parametric yield prediction of digital integrated circuits.* Proc. IEEE Design Automation Conf., 2002.

[10] H-F Jyu, S. Malik, S. Devdas, and K. Keutzer. *Statistical timing analysis of combinational logic circuits.* IEEE Trans. on VLSI, 1(2):126–137, June 1993.

[11] R-B Lin and M-C Wu. *A new statistical approach to timing analysis of VLSI circuits.* Proc. 11th Int'l Conf. on VLSI Design, pp. 507–513, Jan 1998.

[12] P. B. Mirchandani and H. Soroush. *Optimal paths in probabilistic networks: A case with temporary preferences.* Operations Research, 1985.

[13] A. Nadas. *Probabilistic PERT.* IBM Journal Res. Develop, pages 339–347, May 1979.

[14] K. Okada and K. Yamaoka and H. Onodera. *A Statistical Gate Delay Model for Intra-chip and Inter-chip Variabilities.* Proc. Asia-Pacific Design Automation Conf., 2003.

[15] M. Pelgrom and A. Duinmaijer and A. Welbers. *Matching Properties of MOS Transistors.* IEEE Trans. on Solid-State Circuits, Oct 1989.

[16] R. L. Keeny and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* John Wiley & Sons, 1976.

[17] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, S. M. Kang. *An Exact Solution to Transistor Sizing Problem for CMOS Circuits using Convex Optimization.* IEEE Transactions on CAD, 1993.

[18] M. Shaked and J.G. Shanthikumar. *Stochastic Orders and their Applications*. Academic Press, 1994.

[19] C. Vishweswariah. *Death, taxes and failing chips.* Proc. Design Automation Conf., 2003.