Variability-Driven Formulation for Simultaneous Gate Sizing and Post-Silicon Tunability Allocation

Vishal Khandelwal
Department of Electrical and Computer Engineering
University of Maryland College Park
vishalk@glue.umd.edu

Ankur Srivastava
Department of Electrical and Computer Engineering
University of Maryland College Park
ankurs@glue.umd.edu

ABSTRACT

Process variations cause design performance to become unpredictable in deep sub-micron technologies. Several statistical techniques (timing analysis, gate-sizing) have been proposed to counter these variations during design optimization. Another interesting approach to improve timing yield is post-silicon tunable (PST) clock-tree. In this work, we propose an integrated framework that performs simultaneous statistical gate-sizing in presence of PST clock-tree buffers for minimizing binning-yield loss (BYL) and tunability costs by determining the ranges of tuning to be provided at each buffer. The simultaneous gate-sizing and PST buffer range determination problem is proved to be a convex stochastic programming formulation under longest path delay constraints and hence solved optimally. We further extend the formulation into a heuristic to additionally consider shortest path delay constraints. We make experimental comparisons using nominal gate sizing followed by PST buffer management using [12] as a base-case. We take the solution obtained from this approach and perform 1) Sensitivity-based statistical gate-sizing while retaining the PST clock tree 2) Simultaneous gate sizing and PST buffer range determination as proposed in this work. On an average, the BYL obtained from our approach is 98% *lower than the base-case ([12]) and* 95% *lower than the sensitivity-based algorithm. On an average the base-case approach [12] gave* 22% *timing yield loss (YL), the sensitivity approach gave* 19% *YL, where as our proposed algorithm gave only* 3% *YL. The total PST tuning buffer range that is allocated through the proposed algorithm is comparable to that obtained from [12]. The proposed algorithm had a* $2.2x$ *runtime speedup compared to the sensitivity-based algorithm.*

Categories and Subject Descriptors: B.6.3 [Design Aids]: Optimization

General Terms: Algorithms, Design

Keywords: Post-Silicon Tunability, Timing Optimization, Gate Sizing, Stochastic Optimization, Variability

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ISPD'07, March 18–21, 2007, Austin, Texas, USA.
Copyright 2007 ACM 978-1-59593-613-4/07/0003 ...$5.00.

1. INTRODUCTION

Process variations are posing a major challenge to IC designers in the nanometer regime. They cause a significant spread in the performance distribution of designs, making traditional design and analysis techniques to become inaccurate. There has been a distinct shift in VLSI design paradigm to try and develop variability aware methodologies.

In high performance designs, process variations result in a spread in the achievable frequency, thereby causing some chips to fail from meeting the nominal target frequency. In [18], the authors have mentioned that as much as 30% frequency variation can be observed in high-performance designs. Chips can be *binned* according to their operating frequency. Each speed bin has a corresponding penalty cost that is proportional to its slowdown from the target frequency. Thus, there exists a binning-yield loss with each design depending on the spread in its operating frequency due to process variations. In this work, we use binning yield loss (BYL) as an optimization objective in our formulation.

A lot of recent work has focused on statistical techniques for considering process variability during analysis and optimization. One such direction of research has been timing analysis in presence of variability. Statistical Timing Analysis has emerged as a powerful tool to predict the timing distribution of designs [6, 24, 22, 7]. Other recent approaches have tried to utilize this available statistical information about the design to perform statistical optimizations like gate sizing [13, 17, 1, 3, 10, 14]. Essentially, these are analysis and optimization techniques that can be used to counter variability at design time.

Post-silicon tunability is another technique to improve timing yield in circuits. This would allow the manufacturer to tune each chip individually to try and meet the required performance constraints. Recently, post-silicon tunable (PST) clock-tree synthesis [12, 21, 15, 4] has been proposed as one such approach that can be applied to high performance designs to correct timing violations. It can be noted that having PST in the design incurs a cost overhead both in terms of hardware (area) and power, which is termed as the cost of tunability in the design.

There is no existing work that tries to integrate both post-silicon and pre-silicon optimization paradigms into one flow. While performing design time optimization (say gate sizing) one can leverage the information about the available post-silicon tunability and vice-versa. The work in [12] determines the locations of the PST buffers and also their

ranges. In this work, we do not decide the location of the PST buffers. The PST clock tree structure as determined by [12] is taken as an input to our algorithm. We retain the PST buffer locations and clock tree structure but perform simultaneous gate sizing and PST buffer range determination for improved BYL.

The problem that we address in this work can be formally stated as: **Given a sequential design with a synthesized PST clock-tree (with known tunable buffer locations), we perform simultaneous gate sizing of the combinational logic gates and tuning range determination of each PST buffer, such that the Binning Yield Loss and Tunability Cost is minimized.**

We formulate this problem as a two-step stochastic program [16]. We will first develop a formulation considering only longest path constraints. We will prove that it is a convex formulation and hence can be solved optimally. We extend this formulation further into a heuristic considering shortest path constraints (which are inherently non-convex). We use the Kelley's Cutting Plane Method [16] to solve the formulation. We do not make any assumption about the nature of the underlying process variations as well as the correlation modeling strategy.

We make experimental comparisons using nominal gate sizing followed by PST buffer management using [12] as a base-case. We take the solution obtained from this approach and perform 1) Sensitivity-based statistical gate-sizing (similar to [3]) while retaining the PST buffer locations and ranges as determined in the base-case [12] in an effort to re-optimize the design. 2) Simultaneous gate sizing and PST buffer range determination as proposed in this work. On an average, the BYL obtained from our approach is 98% lower than the base-case ([12]) and 95% lower than the sensitivity-based algorithm. On an average the base-case approach ([12] gave 22% timing yield loss (YL), the sensitivity approach gave 19% YL, where as our proposed algorithm gave only 3% YL. The total PST tuning buffer range that is allocated through the proposed algorithm is comparable to that obtained from [12]. The proposed algorithm had a $2.2x$ runtime speedup compared to the sensitivity-based algorithm.

The rest of the paper is organized as follow: section 2 presents the relevant background information and definitions, section 3 presents the convex problem formulation, section 4 extends the formulation into a heuristic considering the shortest path constraints, section 5 presents the proposed algorithm that is used to solve the formulation, section 6 and section 7 discuss the experimental results and conclusions from this work respectively.

2. BACKGROUND AND DEFINITIONS

In this section, we will discuss the relevant background information that is needed to understand this work.

2.1 Binning-Yield Loss

In high performance designs, process variations result in a spread in the achievable frequency, thereby causing some chips to fail from meeting the nominal target frequency. In [18], the authors have mentioned that as much as 30% frequency variation can be observed in high-performance designs. Chips can be *binned* according to their operating frequency. The penalty that the chips in a speed bin have to incur is proportional to the slowdown from the target timing

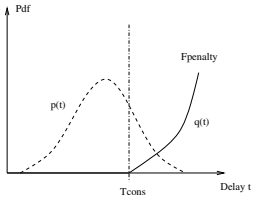

Figure 1: Binning Yield Loss with a Convex Penalty Function

constraint (T_{cons}). Let us suppose that the timing delay of the chip is t. We define a BYL penalty function $F_{penalty}(t)$ as follows:

$$F_{penalty}(t) = \begin{cases} q(t - T_{cons}); & t \geq T_{cons} \\ 0; & otherwise \end{cases} \quad (1)$$

where $q(t-T_{cons})$ is assumed to be a convex function. Let us suppose the probability density function (pdf) of circuit delay is $p(t)$ as shown in figure 1. Hence for longest path constraints, we can define the BYL for the design as:

$$BYL = \int_{-\infty}^{\infty} F_{penalty}(t)p(t)dt = \int_{T_{cons}}^{\infty} q(t - T_{cons})p(t)dt \quad (2)$$

BYL for shortest path constraints can also be defined similarly. In the optimization framework proposed in this work, we will use the above definition for BYL.

2.2 Traditional Gate Sizing

The traditional gate sizing problem tries to minimize the cumulative sum of gate sizes while assigning a size to each gate in the circuit such that the timing constraint T_{cons} at the primary outputs are met. Let x_i denote the size of gate i. The delay of the gate d_i is a function of its size and the sizes of all its fanout gates and hence is denoted as $d_i(\vec{x})$. In general, we perform sizing by varying the channel widths of each transistor in the gate (hence gate size x_i is proportional to the channel width), while the channel lengths are kept constant. If we denote the arrival time at gate i as t_i. The traditional gate sizing problem can be written as:

$$Minimize \quad \sum_{\forall gate\ i} c_i \times x_i$$
$$Subject \quad to: \begin{cases} t_j + d_i(\vec{x}) \leq t_i & \forall j \in fanin(i); \forall gate\ i \\ t_i \leq T_{cons} & \forall i \in PO \\ x_{min}^i \leq x_i \leq x_{max}^i & \forall gate \quad i \end{cases} \quad (3)$$

where c_i is a positive weighting constant for each gate. In this simple formulation, we propose to optimize the total area of the gates which is the most common optimization objective [10, 14]. Additionally, one gate can perform gate sizing to minimize the power [13, 17] or yield-loss [1, 3].

2.3 Convex Gate Delay Modeling

As shown in [9, 20], the elmore delay of a gate can be modeled as a posynomial function of the transistor sizes $\vec{x}$. We can model each transistor as an equivalent resistor and capacitor whose magnitudes are proportional to the channel width w of each transistor. Elmore delay of gate i can be written as a posynomial functions of these resistors and capacitors of gate i and the capacitors of its fanout gates. As shown in [20], gate delay can be written as a function of its size x_i (since it is proportional to the channel width w). Hence, the posynomial gate delay can be expression as:

$$d_i(\vec{x}) = a_{0i} + a_{1i}\frac{\sum_{\forall j} x_j}{x_i} \qquad j \in fanout(i) \quad (4)$$

where a_{0i} and a_{1i} are positive constants that depend on circuit parameters such as threshold voltage, effective channel length, supply voltage and oxide thickness. This posynomial gate delay representation can be changed into a convex

form but making a change of variables $x_i = e^{y_i}$. Each arrival time variable t_i in the gate sizing formulation can be represented as $t_i = e^{z_i}$. Hence, the gate sizing formulation can be presented as:

$$Minimize \quad \sum_{\forall gate\ i} c_i \times e^{y_i}$$
$$Subject \quad to: \begin{cases} t_j(z_j) + d_i(\vec{y}) \leq t_i(z_i) & \forall j \in fanin(i) \\ t_i(z_i) \leq T_{cons} & \forall i \in PO \\ x^i_{min} \leq e^{y_i} \leq x^i_{max} & \forall gate \quad i \end{cases} \quad (5)$$

All variables have an exponential representation which makes the above gate sizing formulation convex in $\vec{y}$ [19].

2.4 Post-Silicon Tunable Clock Tree

Several recent work [12, 21, 15, 4] have proposed that PST clock tree can improve the timing yield for designs in presence of process variations. The central idea is to insert post-silicon tunable buffers into the clock tree that can be used to introduce extra slack into the critical paths in order to correct the timing violations by adjusting the clock skews. In [12], the authors have proposed an approach for PST clock-tree synthesis that tries to minimize the total number of candidate PST clock buffer locations and also reduce the hardware cost of each PST buffer by computing its required tuning range. It is important to note here that inserting redundant PST buffers into the clock tree may results in significant overhead in chip area. Moreover, since the clock buffer also have some capacitance, they also increase the power consumption of the clock tree.

Let us try to understand how a PST clock tree can help improve timing yield. Given a sequential design, we can represent it as a graph $G = (V, E)$, where V is a set of flip-flops (FFs) and E is a set of edges representing timing arcs between the FFs. An edge e_{ij} would represent a combinational logic path between flop i and j. Let us suppose that T_i and T_j are the clock arrival times at flops i and j respectively (they may not be the same due to clock skew). In this work, we look to satisfy the longest path constraint in sequential design for BYL optimization. Let the maximum delay between all combinational logic paths between FFs i and j be D_{ij}. Let the setup time for flip-flop (FF) j be T^j_{set} and T_{clk} be the nominal clock period. In order to meet the longest path timing constraint, the circuit needs to satisfy the following inequality:

$$T_i + D_{ij} \leq T_{clk} + T_j - T^j_{set} \quad (6)$$

Now, as shown in figure 2 let us suppose that we have a PST clock tree with tunable buffers $B1 - B7$ as shown. Each of these tunable buffers k has a tuning delay T^{Buf}_k that can be in the range of 0 to R^{max}_k which has been decided during the design stage (pre-fabrication):

$$0 \leq T^{Buf}_k \leq R^{max}_k \quad (7)$$

Now, as is evident from figure 2, each FF i can have its clock arrival time T_i adjusted by tuning appropriate buffers that lie on the path between the clock tree source and itself. For example, FF 1 can be affected by PST buffers $B1$, $B2$ and $B4$. Hence, if a path starting at FF 1 violates the timing constraint (equation (6)) post-fabrication due to process variability, we can adjust the tuning of the corresponding buffers to try to bring the path back into feasibility region. Each FF i is affected by a subset C_i of PST tunable buffers and hence this technique can be used to redistribute timing slack between critical and non-critical paths such that maximum timing violations can be mitigated. Also, it is easy to

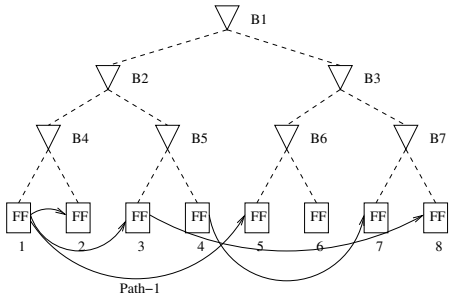

Figure 2: Sequential Design with a PST Clock Tree

note that since many FFs share the same PST buffer, this tuning needs to be done carefully to ensure that no other path violates its timing constraint. In essence, we can rewrite equation (6) considering PST tunability as:

$$(T_i + \sum_{k \in C_i} T^{Buf}_k) + D_{ij} \leq T_{clk} + (T_j + \sum_{k \in C_j} T^{Buf}_k) - T^j_{set} \quad (8)$$

In [21, 11], the authors have proposed a design for PST buffers using passive loads and inverters. The final tuning can be done by connecting the required number of passive banks through a programmable pass bit at each bank. This design provides tuning proportional to its RC delay which in turn corresponds to its bank size and silicon area. There is a hardware and power overhead that is associated with implementing a PST clock tree. The hardware cost is reflective of the silicon area overhead which is proportional to both the number of tunable buffers and their respective tuning ranges (which decides the passive load bank and inverters that are used). There is also a cost associated with the actual tuning delay used at each buffer (indicative of the clock tree power overhead). Thus it is important to compute the tuning range at each buffer such that the maximum timing yield improvements can be achieved without having wasted passive load banks at the PST buffers. *We define tunability cost (TC) as a metric of the overhead of having these passive load banks and inverters in the PST tree. As explained, this overhead is in terms of both silicon area and power and is proportional to the range of the tuning buffers. In this work, TC is also an optimization objective.*

$$TC = \sum_{k \in PST-buffers} R^{max}_k \quad (9)$$

where R^{max}_k is the tuning range allocated to PST buffer k.

3. SIMULTANEOUS GATE SIZING AND PST BUFFER RANGE DETERMINATION FOR MINIMIZING BYL AND TC

In this work, we address the following problem:

Given a sequential design with a synthesized PST clock-tree (with known tunable buffer locations), we perform simultaneous gate sizing of the combinational logic gates and tuning range determination of each PST buffer, such that a combined objective function of the binning yield loss and tunability cost is minimized.

In this section, we first develop the formulation considering only longest path constraints and prove it to be optimally solvable. Later, we will extend the formulation to consider shortest path constraints as well.

3.1 Effect of Variability on Gate Sizing

Process variations cause significant spread in circuit parameters like L_{eff}, t_{ox} and V_{th}. These in turn make the gate

delays unpredictable. Typically, the circuit parameters that are affected by variations can be treated as random variables, making gate delay a function of these random variables. Let us denote the random vector denoting all the variable circuit parameters be $\vec{\Omega}$ where each parameter can have its own density function and these can be correlated in arbitrary ways. Thus, the coefficients in the gate delay model presented in equation (4) would now become a function of the underlying random field. In this light, the delay of gate i in the convex gate sizing formulation (as presented in equation (5)) also becomes a random variable and can be denoted as:

$$d_i(\vec{y},\vec{\Omega}) = a_{0i}(\vec{\Omega}) + a_{1i}(\vec{\Omega})\frac{\sum_{\forall j} e^{y_j}}{e^{y_i}} \qquad j \in fanout(i) \tag{10}$$

In presence of process variability, we can therefore redefine the objective of gate sizing to be BYL minimization. Let us suppose that $\vec{\omega}$ represents the nominal values of each of the varying parameters. We attempt to perform gate sizing at these nominal parameter values such that the BYL is minimized. This problem can be formulated as:

$$Minimize \quad BYL(\vec{y})$$
$$Subject \quad to: \begin{cases} t_j(z_j) + d_i(\vec{y}) \le t_i(z_i) & \forall j \in fanin(i) \\ t_i(z_i) \le T_{cons} & \forall i \in PO \\ x^i_{min} \le e^{y_i} \le x^i_{max} & \forall gate \quad i \end{cases} \tag{11}$$

We try to meet the timing constraint T_{cons} at these nominal parameter values. Additionally, in order to control the total sizing area, we could add a constraint $\sum c_i \times e^{y_i} \le Area_{max}$ to the formulation above. The gate sizing formulation used in this work is similar to that proposed in [2].

3.2 PST Clock Tree Structure and Assumptions

In this work, we assume that we are given a synthesized PST clock tree as well as the location of the tunable clock buffers. We do not make any assumption about the structure of the clock tree (it can be balanced or unbalanced), clock skews or the location of the tunable buffers. We use the PST clock tree alongwith the buffer locations obtained from [12] as an input to our algorithm.

3.3 Problem Formulation

The simultaneous gate sizing and PST buffer range determination problem can be formulated as a Two-Stage *Stochastic Program* [16]. The sequential design can be viewed as a set of FFs and logic gates. Each pair of FFs can share a combinational logic path between them. Each such path needs to meet the timing constraint in order to make the design feasible.

For every pair of FFs i, j that are connected through combinational logic, we define a variable D_{ij} that represents the delay of the longest path between them. We can compute D_{ij} using the inequalities similar to that in the gate sizing formulation on the combinational logic between these two FFs:

$$\begin{array}{c} t_p(z_p) + d_q(\vec{y}) \le t_q(z_q) \quad \forall p \in fanin(q) \\ t_q(z_q) \le D_{ij} \quad q \ is \ fanin \ of \ FF:j \\ x^q_{min} \le e^{y_q} \le x^q_{max} \qquad \forall gate \quad q \end{array} \tag{12}$$

For each pair of FFs i,j, we can write the constraints mentioned above through inequalities (12) and compute the longest path delay D_{ij}.

3.3.1 Variables of Interest

There are three sets of variables in the problem formulation. The first set are the gate-size variables represented by $\vec{y}$, where the size of gate i is given by e^{y_i}. The second set of variables represented by $\vec{r}$, where the tuning buffer ranges for each PST buffer i is given by e^{r_i}. The third set of variables are represented by $\vec{z_i}$, where the arrival time at each gate i is given by e^{z_i}.

3.3.2 Objective of Interest

A general objective function can be to minimize a combination of BYL, TC (which is representative of the area and power overhead incurred in PST clock tree) and also the total gate-size (similar to traditional gate-sizing problem). Since the tuning range at each PST buffer is proportional to the area and power overhead, TC can be represented by the sum of the total range of all PST tuning buffers. Hence, a general objective function of interest could be written as:

$$Minimize \quad (BYL(\vec{y},\vec{r})) + TC(\vec{r}) + \sum_i Gate - Sizes) \tag{13}$$

$$Minimize \quad (BYL(\vec{y},\vec{r})) + \sum_k \alpha_k e^{r_k} + \sum_i \beta_i e^{y_i}) \tag{14}$$

BYL is a function of both $(\vec{y},\vec{r})$ as explained later. This objective function allows to explore the trade-off between BYL, TC and the total gate-size area by appropriately scaling the constants $\vec{\alpha}$ and $\vec{\beta}$.

3.3.3 Two-Stage Stochastic Program

The first stage of the problem formulation can be written in general form as:

$$Minimize \quad (BYL(\vec{y},\vec{r}) + \textstyle\sum_k \alpha_k e^{r_k} + \sum_i \beta_i e^{y_i})$$

Subject to:

$$\begin{cases} T_i + D_{ij} \le T_{clk} + T_j - T^j_{set} \quad \forall FFs(i,j) \\ \left.\begin{array}{l} t_p(z_p) + d_q(\vec{y}) \le t_q(z_q) \quad \forall p \in fanin(q) \\ t_q(z_q) \le D_{ij} \qquad q \ is \ fanin \ of \ FF:j \end{array}\right\} \forall FFs(i,j) \\ x^q_{min} \le e^{y_q} \le x^q_{max} \qquad \forall gate \quad q \\ \sum_k e^{y_k} \le X_{max} \qquad \forall gate \quad k \\ 0 \le e^{r_m} \le R^{max}_m \qquad \forall m \in PST \ Buffer \\ \sum_m e^{r_m} \le Range^{max} \qquad \forall m \in PST \ Buffer \end{cases} \tag{15}$$

Let us try to understand the constraints in the above formulation. The first constraint in inequalities of (15) represents the longest-path constraint (equation (6)) between each pair of FFs that share a path between them. Here, T_i, T_j, T_{clk} and T^j_{set} are known constants that correspond to clock arrival times. The longest path delay D_{ij} can be determined from the next three inequalities that represent the gate sizing formulation for the logic paths between FFs i and j. We note that a gate can show on multiple paths, hence there would be several such sizing constraints on each gate. But since the first stage problem considers all these constraints together, there is no discrepancy that can come in. The total sum of gate sizes for the design can be bounded to be less than a constant X_{max} using inequality 5 above. Each PST buffer m can be bound to have a maximum allowed tuning range R^{max}_m. In order to limit the total tunability cost, we can also have a bound on the total cumulative tuning range given by $Range^{max}$. These are represented by the last two inequalities. This is the most general form of the first stage problem.

In presence of process variability, the delay between each pair of FFs i-j that have a combinational logic path between

them, becomes a random variable that can be represented as $D_{ij}(\vec{y},\vec{r},\vec{\Omega})$ that depends on the gate-sizes $\vec{y}$, the tuning buffer range $\vec{r}$ and the random field due to process variations $\vec{\Omega}$ (that may have some correlation between its components). Let us define a random variable P that denotes the penalty of violating the timing constraint (T_{clk}) as:

$$P(\vec{y},\vec{r},\vec{\Omega}) = \begin{cases} q(D_{ij}(\vec{y},\vec{r},\vec{\Omega}) - T_{cons}); & D_{ij} \geq T_{cons} \\ 0; & otherwise \end{cases} \quad (16)$$

where $q(.)$ is the convex penalty function that was defined in equation (1).

In equation (2), BYL was defined as the expected value of the timing-violation penalty. For a given $(\vec{y},\vec{r})$ and a sample ω of the random field Ω, let $p(\vec{y},\vec{r},\vec{\omega})$ be the value of the random variable P. By definition, $p(\vec{y},\vec{r},\vec{\omega})$ denotes the timing-violation penalty for a given $(\vec{y},\vec{r})$ at that variability sample ω. Hence, BYL would be the average timing-violation penalty over all such samples ω which is the expected value of the random variable P for a given $(\vec{y},\vec{r})$. Therefore:

$$BYL(\vec{y},\vec{r}) = E[P(\vec{y},\vec{r},\vec{\Omega})] \quad (17)$$

We can evaluate the timing-violation penalty $p(\vec{y},\vec{r},\vec{\omega})$ given a fixed $\vec{y}$, $\vec{r}$ and a variability sample $\vec{\omega}$ through another convex formulation that can be written as:

$$p(\vec{y},\vec{r},\vec{\omega}) = Minimize \quad \textstyle\sum_{FF(i,j)} q(T_{ij}^{viol_s})$$

Subject to:

$$\begin{cases} (T_i + \sum_{k\in C_i} T_k^{Buf}) + D_{ij}(\vec{y},\vec{r},\vec{\omega}) \leq T_{clk} + \\ (T_j + \sum_{k\in C_j} T_k^{Buf}) - T_{set}^j + T_{ij}^{viol_s} \quad \forall FFs(i,j) \\ t_p + d_q(\vec{y},\vec{\omega}) \leq t_q \quad \forall p \in fanin(q) \\ t_q \leq D_{ij}(\vec{y},\vec{r},\vec{\omega}) \quad q \ is \ fanin \ of \ FF:j \end{cases} \forall FFs(i,j)$$
$$T_{ij}^{viol_s} \geq 0 \ \forall FFs(i,j)$$
$$0 \leq T_k^{Buf} \leq e^{r_k} \ \forall k \in PST \ Buffer \quad (18)$$

Let us try to understand this formulation. Given a value of $\vec{y}$, $\vec{r}$ and a variability sample $\vec{\omega}$ implies that the delay of each gate i ($d_i(\vec{y},\vec{\omega})$) is known. Also, since $\vec{r}$ is given, the range of each tuning buffer k is will be e^{r_k}. As mentioned before in subsection 2.4, T_i and T_j are the clock arrival times at FFs i and j respectively and are known values. For each FF i, we know the set of tuning buffers C_i that can affect the clock arrival time at this FF. In the above formulation, the problem variables are T_k^{Buf} which is the actual tuning at PST buffer k that is used to reduce the timing violation. The longest path delay D_{ij} for each pair of FFs (i,j) is a variable and the arrival time t_i at each gate i is a variable. Additionally, we define a variable T_{ij}^{viol} for each pair of FFs (i,j) that represents the timing violation along the longest path between those FFs. The timing-violation penalty at each FF pair (i,j) can be computed as $q(T_{ij}^{viol_s})$. The objective of this problem is to minimize the sum of timing-violation penalty across all pairs of FFs (i,j) by appropriately assigning delay tuning to each PST buffer within the range given by the variables $\vec{r}$. Essentially, this formulation tries to determine the best combination of tuning set ($\vec{T^{Buf}}$) that should be applied at the PST buffers such that the total timing-violation penalty for the design is minimized.

For a given value of $\vec{y}$, $\vec{r}$, the optimal objective to this formulation gives us $p(\vec{y},\vec{r},\vec{\omega})$ which is the desired quantity to compute BYL$(\vec{y},\vec{r})$.

The two formulations defined by inequalities (15) and (18) form a classic Two-Stage Stochastic Programming formulation [16], where the former is called the first-stage problem and the latter second-stage problem. We would like to point out that even though the proposed formulation considers clock arrival times (T_i, T_j) to be constant, our formulation can be extended to consider uncertainty in clock tree as well. In that case, the second stage formulation would consider the clock arrival times $(T_i(\vec{\omega}), T_j(\vec{\omega}))$ to be dependent on the randomness (Ω).

3.4 The problem formulation is convex in ($\vec{y}$,$\vec{r}$)

Theorem: The proposed two-stage stochastic programming formulation is convex.
Proof: Detailed proof omitted for brevity.

4. SHORTEST PATH DELAY CONSTRAINTS

The formulation discussed in the earlier sections presents a provably optimal technique considering only longest path (setup time) constraints. However, for a pair of FFs i and j, we also need to satisfy the shortest path (hold time) constraints. Given the shortest path delay D_{ij}^{short} between the two FFs, we can write the shortest path delay constraint as:

$$T_i + D_{ij}^{short} \geq T_j + T_{hold}^j \quad \forall FFs(i,j) \quad (19)$$

where T_{hold}^j is a constant denoting the hold-time for FF j, T_i and T_j are clock arrival times. As can be seen, this is a non-convex constraint considering the convex gate delay models given by equation 10. Hence, considering shortest path constraints in the formulation proposed in the earlier section would break the convex nature of the problem. We will now present an efficient heuristic to consider the shortest path constraints in our formulation while preserving its convexity.

Let us suppose that we are given p paths which are candidates for shortest path delay violation (can be determined from static timing analysis). The cumulative delay of the gates on each of these paths would give us the delay of the path. We will make a linear approximation on the gate delay model for these gates wrt the gate sizing variable. Given a gate m (with size e^{y_m}) and its fanout gate n (with size e^{y_n}), we can approximate its gate delay as a linear function of the sizing variables (y). This model is constructed such that it is a lower bound to the convex gate delay model given by equation 4. Therefore, the shortest path delay is under-predicted by our linear gate delay model approximation and any valid solution will always satisfy the shortest path delay constraint. Let us suppose that the path delay of the pth shortest path is denoted by $D_{ij}^{short_p}$, we can compute the linear gate delay and the shortest path delay as:

$$D_{ij}^{short_p} = \sum_m d_m^{lin} \quad \forall gates \ m \ on \ path \ p \quad (20)$$

$$d_m^{lin} = a_{0m} + a_{1m} y_m + \sum_{\forall fanout-n} b_n y_n \quad (21)$$

where a_0, a_1 and b_n are constants. Under these assumptions, it can be seen that the shortest path constraint as given by equation 19 is now convex and can be added to our proposed formulation without breaking the convex nature of the problem.

Let us now understand, how we can extend the two-stage stochastic programming formulation to also consider shortest path delay constraints. Given the p paths which are

candidates for shortest path delay violation, the first stage formulation as given by equations 15 can be modified to additionally consider the constraint:

$$T_i + D_{ij}^{short_p} \geq T_j + T_{hold}^j \quad \forall paths\ p \quad \forall FFs(i,j) \qquad (22)$$

where $D_{ij}^{short_p}$ is defined using equations 20 and 21.

The BYL will now consists of both longest path delay violation and shortest path delay violation. The second stage problem given by equations 18 can be modified to consider the BYL due to shortest path delay violation. The timing violation penalty can now be computed as:

$$p(\vec{y},\vec{r},\vec{\omega}) = Minimize \quad \sum_{FF(i,j)} q(T_{ij}^{viol_s}, T_{ij}^{viol_h}) \qquad (23)$$

where $T_{ij}^{viol_s}$ represents the timing violation in the longest path constraints and $T_{ij}^{viol_h}$ represents the timing violation in the shortest path constraints. The second stage formulation can consider additional constraints for shortest path delay violation as given by:

$$\begin{aligned}(T_i + \sum_{k\in C_i} T_k^{Buf}) + D_{ij}^{short_p}(\vec{y},\vec{r},\vec{\omega}) + T_{ij}^{viol_h} \geq (T_j + \sum_{k\in C_j} T_k^{Buf}) \\ + T_{hold}^j \quad \forall paths\ p \quad \forall FFs(i,j) \\ T_{ij}^{viol_h} \geq 0 \quad \forall FFs(i,j)\end{aligned} \qquad (24)$$

where T_k^{Buf} is the tunable delay introduced due to PST buffer k. This constraint gives us the timing violation in the shortest path constraint $T_{ij}^{viol_h}$ for path p.

This completes the extension of the two-stage stochastic programming formulation to consider the shortest path constraints in addition to the longest path constraints. Although, we preserve the convex nature of the formulation, the error introduced due to the lower bounding linear approximation on the gate delay models for shortest path constraints makes this a heuristic technique.

5. SOLVING THE TWO-STAGE STOCHASTIC PROGRAM

In this work, we have used Kelley's Cutting Plane Method [19] to solve the two-stage stochastic programming formulation. We would like to point out that this is just one technique that can be applied to solve this convex formulation. Any other convex optimization scheme can be used as well. *For the sake of brevity, we will not include details of the method in this paper.*

This method relies on the computation of a lower bound to BYL($\vec{x}$) and is the most critical step in Kelley's Algorithm. We will briefly discuss some details regarding this step of the method. At a given solution of the first stage problem, i.e. ($\vec{y}$,$\vec{r}$), computing the BYL($\vec{y}$,$\vec{r}$) amounts to estimating the expected value of the timing-violation penalty $P(\vec{y},\vec{r},\vec{\Omega})$. In the scenario when there are no PST clock buffers in the design, the problem of computing the timing-violation penalty would amount to computing the timing pdf that can be done using STA technique ([6, 24, 22, 7]). But in our case, we also have PST clock buffers, where the amount of tuning required at each buffer for best timing yield would vary depending on each variability sample ω. To our best knowledge, there are no current STA techniques that can handle timing analysis in presence of PST clock buffers.

Consequently, in this work we resort to using a Monte-Carlo based STA technique were for each sample ω of the random field, we formulate the second-stage problem as proposed using inequalities (18) and compute the actual timing-violation penalty $p(\vec{y},\vec{r},\vec{\omega})$. This is repeated for every variability sample ω such that the expected value of timing-violation penalty which equals BYL($\vec{y}$,$\vec{r}$) is eventually computed. We note here that since we need to generate each β_i once at a time, this STA process is repeated for every variable $\vec{y}$ and $\vec{r}$. It is easy to note that this step becomes a major bottleneck in the performance of our algorithm and makes the entire computation slow.

However, the proposed algorithm is free to use any efficient STA technique that can predict timing pdf in presence of PST clock buffers. In the future, when such a STA technique has been developed, it can be plugged into the proposed algorithm. In our results section, we will show that almost all the computational time for our algorithm goes into this Monte-Carlo based STA computation.

6. EXPERIMENTAL RESULTS

The overall formulation considering shortest and longest path constraints was implemented in SIS [5]. We performed experiments on the ISCAS benchmark suite. We generated a valid placement for each benchmark using *CAPO*. The correlation information between gates was generated using the model proposed in [6]. We assumed that process variability caused threshold voltage to have a Gaussian distribution with a mean value of $0.2V$ and a standard deviation of 15% from the mean. We used 90nm technology parameters (from [23]) to compute the coefficients of the convex gate delay expression (as a function of its size) as given by equation (4). The PST clock tree structure used in our experiments is obtained using the algorithm proposed in [12]. Each PST buffer was allowed to have a maximum tuning delay of 5 psec.

In order to solve the first-stage convex formulation, we integrated MOSEK [8] with SIS. The formulation proposed in section 5 was also implemented in SIS. As mentioned in that section, we implemented a Monte-Carlo based STA scheme to compute the BYL during each iteration of the cutting plane algorithm. In figure 3, we can see that the upper bound (objective) representing the BYL at the current solution improves in each iteration and quickly converges to the lower bound.

There is no scheme in the literature that does simultaneous gate sizing and PST buffer management. We have run three set of experiments to evaluate our algorithm:

1. A nominal gate sizing scheme followed by PST buffer management as proposed in [12]: We first run gate sizing assuming nominal process parameter values. On this solution, we perform PST buffer management (location and tuning range determination of each PST buffer) using the algorithm proposed in [12].

2. Taking the solution from experiment 1 ([12]), we retain the PST clock buffer structure (location and ranges) but try to re-optimize the design using a sensitivity-based statistical gate-sizing approach similar in spirit to that proposed in [3]: This approach is an iterative

bench	T_{cons}	[12]			[12] + Sensitivity			[12] + Convex Stochastic		
name	(*psec*)	*BYL*	*Area*	*Buf.Range*	*BYL*	*Area*	*Buf.Range*	*BYL*	*Area*	*Buf.Range*
s27	450	4165	402	9	3293	403	9	4	418	16
s298	700	30854	4135	4	28477	4187	4	414	4146	5
s344	1000	38850	3822	3	14289	4006	3	377	3838	3
s382	700	54916	5073	6	95364	5273	6	116	5162	7
s400	850	71823	5370	3	61400	5441	3	1638	5418	8
s499	1350	232523	6614	15	260749	6706	15	8766	6714	17
s526	900	58750	8091	8	2210	8152	8	568	8133	10
s635	2500	253551	7730	3	111368	7853	3	1308	7784	1

Table 1: Comparison of Binning Yield-Loss, Area and Total PST Buffer Range in (psec)

bench	T_{cons}	[12]	[12] + Senstivity	[12] + Convex Stochastic
s27	450	0.23	0.18	0.03
s298	700	0.16	0.11	0.02
s344	1000	0.24	0.15	0.03
s382	700	0.16	0.11	0.02
s400	850	0.26	0.18	0.05
s499	1350	0.24	0.26	0.03
s526	900	0.26	0.09	0.02
s635	2500	0.17	0.12	0.05
Average		0.22	0.19	0.03

Table 2: Comparison of Yield-Loss

scheme where at each step, we evaluate the BYL improvements that can be achieved per unit size increase for each gate. The most sensitive gate is chosen as the next gate to be upsized.

3. Taking the solution from experiment 1 ([12]), we retain only the locations of the PST clock buffers and run our simultaneous gate sizing and PST buffer range determination algorithm: The PST clock tree obtained in experiment 1 is taken as an input, though we reallocate the range of each of the PST buffers while performing gate sizing as proposed in this work.

The aim of these experiments is to show that our proposed algorithm can provide significant improvements over the design obtained from [12]. Furthermore, comparison with experiment 2 shows that the simultaneous gate sizing and PST buffer range determination algorithm proposed in this work is significantly more effective than performing a statistical resizing of the design.

In order to compute the BYL for each experiment, we impose the process parameter variations (Ω) on the final design solution through monte-carlo simulation and compute the minimal timing violation considering tunability for each sample (ω). The average BYL over all ω was taken as the BYL for the design.

Table 1 compares the three approaches in terms of the BYL, the total area after gate sizing and the tuning buffer range. We can see that our proposed convex-stochastic approach resulted in significantly lower BYL compared to the other two cases. Since the nominal gate sizing is not variability-aware, experiment 1 resulted in the highest BYL. On an average, the BYL obtained from our approach is 98% lower than the solution from experiment 1 ([12]) and 95% lower than experiment 2, the sensitivity-based algorithm.

The final gate-size area obtained for our approach is on an average 1.25% lower than that obtained from experiment 1 (nominal gate sizing followed by [12]) and 0.62% higher than experiment 2, the sensitivity approach. Hence, the convex-stochastic algorithm gives better BYL for similar total gate-size area. From figure 4, we can see that our approach gives much lower BYL for the same total gate-size area as compared to the sensitivity-based algorithm.

bench	Sensitivity		Convex Stochastic		Speedup
	#itera.	time	#itera.	time	
s27	14	0.5	10	0.9	0.6
s298	16	13.7	9	11.6	1.2
s344	24	24.3	7	14.6	1.7
s382	40	53.9	19	41.3	1.3
s400	18	28.3	13	19.5	1.5
s499	35	87.1	19	72.2	1.2
s526	15	52.0	14	40.1	1.3
s635	109	378.0	7	43.3	8.7
Average					2.2x

Table 3: Comparison of Total Run-Time (min) and Number of Iterations

bench	Avg. Iter. Time	Avg. STA time / Iter	%
s27	5	4	80.0
s298	77	74	96.1
s344	125	117	93.6
s382	130	127	97.7
s400	90	85	94.4
s499	228	225	98.7
s526	172	165	95.9
s635	371	351	94.6
Average			93.8

Table 4: Contribution of Monte-Carlo Based STA time to Iteration Time (sec)

The total PST tuning buffer range that is allocated through the proposed algorithm is comparable to that obtained from [12]. Hence, our algorithm is able to identify PST buffer ranges that result in BYL reduction without putting any additional overhead in terms of PST buffer cost while performing simultaneous gate sizing.

Table 2 reports the traditional timing YL that were obtained for the solutions from all three approaches. It can be seen that on an average the nominal-sizing followed by [12] gave 22% yield-loss, while the sensitivity approach gave 19% yield-loss whereas our proposed algorithm gave only 3% yield-loss. These results show that even though we do not directly optimize for timing yield loss (we optimize BYL), we get better and more robust design solutions.

From figure 5, it is evident that the convex stochastic algorithm has a much faster rate of convergence than the sensitivity-based algorithm. The runtimes for each benchmark are reported in table 3 alongwith the number of iterations. It can be observed that our approach converges to a better solution in fewer iterations and on an average is $2.2x$ faster than the sensitivity-based algorithm.

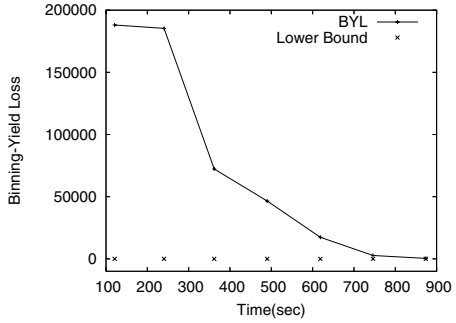

Figure 3: Convergence of BYL to its lower bound with time for s344

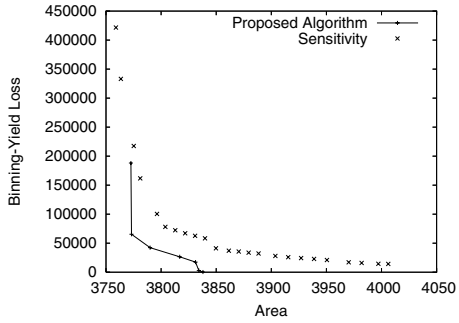

Figure 4: BYL vs. Area Generated at Different Iterations of Kelley's and Sensitivity-Based Algorithms

As pointed out earlier in this paper, the maximum runtime in our approach is taken in computing the BYL using Monte-Carlo based STA. This is due to the fact that none of the current STA techniques are able to perform timing analysis considering tunability. Our proposed algorithm is independent of the STA algorithm used and can be used in combination with an efficient PST aware STA scheme developed in future. From table 4 it can be seen that almost 93% of the computational runtime goes into the STA process.

7. CONCLUSION AND FUTURE WORK

In this work, we have proposed a novel framework that performs simultaneous gate sizing and PST clock tree buffer range determination in order to get lower BYL and tunability costs. An extension to this work would be to include power in the optimization formulation as well as develop an STA scheme that considers post-silicon tunability during timing analysis.

8. REFERENCES

[1] A. Agrawal, K. Chopra, D. Blaauw, and V. Zolotov. "Circuit Optimization Using Statistical Static Timing Analysis". In *DAC*, pages 338–342, 2005.

[2] A. Davoodi and A. Srivastava. "Variability-Driven Gate Sizing for Binning Yield Optimization". In *Procs of DAC*, 2006.

[3] D. Sinha, N. V. Shenoy, and H. Zhou. "Statistical Gate Sizing for Timing Yield Optimization". In *ICCAD*, Nov. 2005.

[4] E. Takahashi, Y. Kasai, M. Murakawa, and T. Higuchi . "A post-silicon clock timing adjustment using genetic algorithms ". In *Digest of technical papers of the 2003 symposium on VLSI circuits*, 2003.

[5] E.M. Sentovich, K.J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P.R. Stephan, R.K. Brayton, A.L. Sangiovanni-Vincentelli. *SIS: A System for Sequential Circuit Synthesis*. Memorandum No. UCB/ERL M92/41, Department of EECS. UC Berkeley, May 1992.

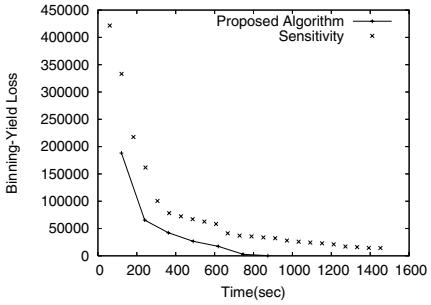

Figure 5: BYL vs. Time Generated at Different Iterations of Kelley's and Sensitivity-Based Algorithm

[6] H. Chang and S. Sapatnekar. "Statistical Timing Analysis Considering Spatial Correlations Using a Single Pert-Like Traversal". In *Procs of ICCAD*, 2003.

[7] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah. "Parameterized block-based statistical timing analysis with non-gaussian parameters". In *Procs of DAC*, 2005.

[8] http://www.mosek.com. .

[9] J. Fishburn and A. Dunlop. "TILOS: A Posynomial Programming Approach to Transistor Sizing". In *ICCAD*, pages 326–328, 1985.

[10] J. Singh, V. Nookala, Z. Luo, and S. Sapatnekar. "Robust Gate Sizing by Geometric Programming". In *DAC*, pages 315–320, July 2005.

[11] Jeng-Liang Tsai, DongHyun Baik, Charlie Chung-Ping Chen, and Kewal K. Saluja. "A yield improvement methodology using pre- and post-silicon statistical clock scheduling". In *Procs. of ICCAD*, 2004.

[12] Jeng-Liang Tsai, Lizheng Zhang and Charlie Chung-Ping Chen. "Statistical Timing Analysis Driven Post-Silicon-Tunable Clock-Tree Synthesis". In *Procs. of ICCAD*, 2005.

[13] M. Mani, A. Devgan, and M. Orshansky. "An Efficient Algorithm for Statistical Minimization of Total Power under Timing Yield Constraints". In *DAC*, pages 309–314, July 2005.

[14] M. R. Guthaus, N. Venkateswaran, C. Visweswariah, and V. Zolotov. "Gate Sizing Using Incremental Parameterized Statistical Timing Analysis". In *ICCAD*, Nov. 2005.

[15] Patrick Mahoney, Eric Fetzer, Bruce Doyle, and Sam Naffziger. "Clock distribution on a dual-core multi-threaded Itanium-family processor ". In *Digest of technical papers of the 2005 international solid-state circuits conference*, 2005.

[16] R. J-B Wets. Stochastic Programs with Fixed Recourse: The Equivalent Deterministic Program. In *SIAM Review*, pages 309–339, July 1974.

[17] S. Bhardwaj, S. B.. K. Vrdhula. "Leakage Minimization of Nano-scale Circuits in the Presence of Systematic and Random Variations". In *ICCAD*, Nov. 2005.

[18] S. Borkar et al. "Parameter Variations and Impact on Circuits and Microarchitecture". In *Proc. Design Automation Conference*, June 2003.

[19] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge 2004.

[20] S. Sapatnekar, V. B. Rao, P.M. Vaidya, and S. M. Kang. "An Exact Solution to the Transistor Sizing Problem for CMOS Circuits Using Convex Optimization". In *IEEE Transactions on CAD*, pages 1621–1634, Nov. 1993.

[21] Simon Tam, Stefan Rusu, Utpal Nagarji Desai, Robert Kim, Ji Zhang, and Ian Young. "Clock generation and distribution for the first IA-64 microprocessor". In *IEEE Journal of Solid-State Circuits*, pages 35(11):1545–1552, Nov 2000.

[22] V. Khandelwal and A. Srivastava. "A General Framework for Accurate Statistical Timing Analysis Considering Correlations". In *Procs of DAC*, 2005.

[23] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu. "New paradigm of predictive MOSFET and interconnect modeling for early circuit design". In *Proc. of CICC*, pages 201–204, 2000.

[24] Y. Zhan, A. J. Strojwas, X. Li, L. T. Pileggi, D. Newmark, and M. Sharma. "Correlation-aware Inboxstatistical timing analysis with nongaussian delay distributions". In *Procs of DAC*, 2005.