TECHNICAL RESEARCH REPORT

Variability Driven Gate Sizing for Binning Yield Optimization

by Azadeh Davoodi, Ankur Srivastava

TR 2006-2

Variability Driven Gate Sizing for Binning Yield Optimization

Azadeh Davoodi Ankur Srivastava
University of Maryland, College Park, MD
{azade,ankurs}@umd.edu

Abstract*— Process variations result in a considerable spread in the frequency of the fabricated chips. In high performance applications, those chips that fail to meet the nominal frequency after fabrication are either discarded or sold at a loss which is typically proportional to the degree of timing violation. The latter is called binning. In this paper we present a gate sizing-based algorithm that optimally minimizes the binning yield-loss. Specifically we make the following contributions: 1) prove the binning yield function to be convex, 2) the proof does not make any assumptions about the sources of variability, their distributions (Gaussian/Non-Gaussian) or correlation, 3) by using Kelley's cutting-plane method for convex programs, we integrate our strategy with statistical timing analysis tools (STA), without making any assumptions about how STA is done, 4) if the objective is to optimize the traditional yield (and not binning yield) our approach can still optimize the same to a very large extent. Comparison of our approach with sensitivity-based approaches under fabrication variability shows an improvement of on average 72% in the binning yield-loss with an area overhead of an average 6%, while achieving a 2.69 times speedup under a stringent timing constraint. Moreover we show that a worst-case deterministic approach fails to generate a solution for certain delay constraints. We also show that optimizing the binning yield-loss minimizes the traditional yield-loss (although it is not a direct objective) with a 61% improvement from a sensitivity-based approach.

I. Introduction

One of the major challenges of today's IC design is dealing with the variabilities caused by the sub-90nm fabrication process. Fabrication variability diverts the parameters of the devices and consequently moves the characteristics of the chips away from their nominal values. In high performance systems, fabrication variability results in a considerable spread in the frequency of the chips (about 30% according to [15]). In some cases, the chips that violate the timing constraint are simply discarded and in other cases they are sold at a loss. In the latter case, those chips that fail to meet the nominal frequency after fabrication are binned based on their speed. Some work such as [2] design hardware to do speed binning in microprocessor design. For each speed bin a loss value exists for selling the chips in that bin for a reduced price. Therefore, depending on the spread in the circuit delay, there exists a binning yield-loss. In this paper we propose a gate sizing approach to minimize this binning yield-loss.

Many researchers have investigated the gate sizing problem from a fabrication-variability perspective [1], [4], [8], [10], [11], [12], [14]. These approaches could be grouped into worst case approaches [8], sensitivity-based approaches [1], [4], [12], [10], and the ones based on a mathematical programming framework [11], [14]. These approaches try to addresses different objectives under variability. For example, [8] minimizes area while considering the worst case uncertainty ellipsoid of parameter variations in a convex formulation. Others minimize the yield-loss [1], [4] or leakage power [11], [14] or combination of both [10].

Firstly none of these approaches consider the binning yield-loss and focus on more traditional definitions of yield (where the chips are discarded if they fail the timing constraint). Secondly these approaches they do not guarantee convergence to the optimal solution in a general case, or at least not from a yield perspective. Some of these approaches may converge to the optimal for their own problem specification but that may not lead to the optimal solution from a yield perspective. For example, the worst case approaches like [8], although look promising do not guarantee optimality of the yield function.

The sensitivity-based approaches optimize the cost function in a neighborhood and do not guarantee convergence to the optimal. The mathematical programming approaches do consider optimality but make constraining assumptions like the Gaussian nature of uncertainty [11] or work with specific models of fabrication variability [14]. Also the approach of [14] approximates the yield percentiles by their upper bounds, and thereby it is not provably optimal.

In this paper, we present a gate sizing approach to optimize the binning yield. Our specific contributions are enumerated below:

1) We optimize the binning yield and propose an optimal algorithm to minimize the same using gate sizing. Our algorithm can be trivially extended to minimize the binning yield under area/power constraints as well. The core of our algorithm is based on the proof of convexity of the binning yield function w.r.t. gate sizes, which allows usage of various convex optimization schemes.
2) The proof of convexity and consequently the optimality of the algorithm is not constrained by any assumptions on the underlying nature of the fabrication variability and/or the model of correlation used.
3) We use Kelley's Cutting Plane algorithm [16] to optimize the binning yield function. Usage of this scheme allows the integration of our approach with any of the existing statistical timing analysis (STA) methods (Gaussian [3] or Non-Gaussian [5], [18]). In fact use of the analytical center approach for convex optimization [16] will allow us to minimize the total number of calls to the STA engine, thereby improving the speed of optimization (since STA is the slowest part of this framework).
4) In case the objective is optimizing the traditional yield, our binning yield-based approach could be used as a heuristic to optimize this objective. We prove that *if there exists a solution in which the traditional yield-loss is 0, our binning yield-loss approach will find such solution. Also, if the optimal value of the binning yield-loss is non zero, then there does not exist a solution to the traditional yield problem in which the yield loss is 0.*

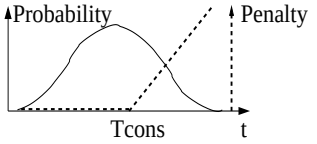

Fig. 1. Binning Yield-Loss Based on Linear Penalty Function

We have compared our approach to the sensitivity-based approaches and have shown an improvement of 72% in the binning yield-loss with a small area overhead of on average 6%, while achieving a 2.69 times speedup. We also show that optimizing the binning yield-loss minimizes the traditional yield-loss on average by 61% when compared to a sensitivity-based approach.

II. Objective: Minimizing the Binning Yield-Loss in High Performance Applications

In high performance systems, fabrication variability results in a considerable spread in the frequency of the chips (about 30% according to [15]). The chips that have a frequency lower than the nominal frequency can either be discarded, or be sold at a loss. For the latter case, the chips that violate the timing constraint are sorted (binned) according to their speed. [2] is a recent work which presents the hardware for doing this speed-binning. Depending on the degree of timing constraint violation for each bit, the chips are sold at a loss. This loss is defined by a penalty function; slower a chip, higher its penalty and loss. All the chips of at least the nominal speed will not have any penalty.

Let t denote the delay of a chip. Let us define a linear penalty function as follows:

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

where T_{cons} is the timing constraint (nominal delay) that the chips are designed for. The chips that have a delay larger than T_{cons} have a penalty equal to their delay-offsets from T_{cons}. This linearity assumption will be relaxed later.

Let $f_T(t)$ denote the probability density function (pdf) for the potential delay values of a design. For the above penalty function, the overall *binning yield-loss* (BYL) is defined as follows:

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

In this paper we will minimize the BYL based on the penalty function of equation 1. We propose an optimal and efficient algorithm to minimize the same. The optimality of our approach holds even if the penalty function is convex (and not necessarily linear).

The delay of a design and consequently our objective can be expressed in terms of the gate sizes, among other parameters:

$$BYL(\vec{s}) = \int_{T_{cons}}^{\infty} (t(\vec{s}) - T_{cons}) f_T(t(\vec{s})) dt \quad (3)$$

where $\vec{s}$ is a vector of the gate sizes in the design. In this paper optimization of $BYL(\vec{s})$ is done over $\vec{s}$ (using gate sizing).

Most of the exiting related work have focused on gate sizing to minimize the yield-loss (YL) under fabrication variability [4], [10], where the YL is given by:

$$YL = \int_{T_{cons}}^{\infty} f_T(t) dt \quad (4)$$

III. Conventional Gate Sizing Problem

A. Problem Formulation

Let s_i denote the size variable of gate i. The variable s_i is proportional to the channel width of the gates' transistors as the channel lengths are usually kept uniform. Let t_i denote the arrival time at the output of gate i from the primary inputs, and d_i denote the delay of gate i. The gate sizing problem is formulated as:

$$Minimize \quad \sum_{\forall gate\ i} c_i \times s_i$$
$$Subject \quad to: \begin{cases} t_j + d_i(\vec{s}) \leq t_i \quad \forall j \in fanin(i); \forall i \\ t_i \leq T_{cons} \quad \forall i \in PO \\ s_{min} \leq s_i \leq s_{max} \quad \forall gate \quad i \end{cases} \quad (5)$$

These constraints ensure that the delay of any path in the circuit is at most T_{cons}. The objective is minimizing the area of the circuit given as summation of s_i variables with a c_i proportionality factor. The solution is the set of gate sizes given as $\vec{s} = \{s_1; s_2; ...; s_n\}$.

Minimizing area while meet a timing constraint is a common gate sizing objective [8], [12]. Other works optimize the yield-loss [1], [4], [10], or power [11], [14] using gate sizing. The formulation could also be written so as to find the feasible solution for a given timing constraint. The delay of gate i depends on its size and of its fanouts sizes. In the above constraints, this dependence is shown as $d_i(\vec{s})$. Therefore the objective and the arrival times in the above formulation also depend on $\vec{s}$.

B. Computing Delay of A Gate As A Posynomial

The delay of a gate can be written as a posynomial function of its transistors sizes using the Elmore delay model [7], [17]. Each transistor is represented using an equivalent on-resistor (r), and a capacitor (c) given as a function of its channel width (w) as [9]:

$$\begin{array}{ll} r = k_r \frac{1}{w} & k_r = f_r(v_{th}, l_{eff}, v_{dd}, t_{ox}) \\ c = k_c w & k_c = f_c(l_{eff}, t_{ox}) \end{array} \quad (6)$$

where k_r and k_c are *positive constants* that are expressed as functions of parameters such as threshold voltage, effective channel length, supply voltage or oxide thickness as expressed above.

The delay of each gate is the time to charge/discharge the capacitors in the resistive path to vdd/ground. Using the Elmore model, this delay is written as a posynomial function of the resistors and capacitors in the gate and of the capacitors of the gates' fanouts. Given that s_i is proportional to the channel widths of the gates' transistors, the delay of a gate i is expressed as [17]:

$$d_i(\vec{s}) = a_{0i} + a_{1i} \frac{\sum_{\forall j} s_j}{s_i} \qquad j \in fanout(i) \quad (7)$$

In the above posynomial expression, a_{0i} and a_{1i} are *positive constants* that depend on k_r and k_c values of the transistors. The inequalities of 5 will therefore be a posynomial formulation.

C. Convex Representation

The presented posynomial formulation is translated into a convex one by the change of variables $s_i = e^{x_i}$ and $t_i = e^{y_i}$ [16]. Therefore $\vec{s} = \{e^{x_1}; e^{x_2}; ...\}$. The formulation in inequalities of 5 is then transformed to:

$$Minimize \quad \sum_{\forall gate\ i} c_i \times e^{x_i}$$
$$Subject \quad to: \begin{cases} t_j(y_j) + d_i(\vec{x}) \leq t_i(y_i) \quad \forall j \in fanin(i) \\ t_i(y_i) \leq T_{cons} \quad \forall i \in PO \\ s_{min} \leq e^{x_i} \leq s_{max} \quad \forall gate \quad i \end{cases} \quad (8)$$

The above formulation will consequently have an exponential optimization form, which is convex with respect to $\vec{x}$ [16].

IV. Gate Sizing for Minimizing the BYL

In this section, we will show the minimization of the BYL over the gate sizes can be *optimally* achieved. Initially we will discuss the effects of fabrication variability on the traditional formulation, and then present our approach, and prove its optimality.

A. Effects of Variability on the Traditional Formulation

Fabrication variability randomizes different device parameters such as L_{eff} or T_{ox} etc.. The resistance and capacitance of a device expressed in equation 6 will therefore be a random variable (r.v.), as they are expressed in terms of such varying parameters.

Assume $\vec{\Omega}$ is a random vector which represents a set of varying parameters in equation 6 such as $\{L_{eff}, T_{ox}\}$ etc.. Each sample vector $\vec{\omega} \in \vec{\Omega}$ represents a set of samples from the assumed field of uncertainty (which can have any associated density function and any correlation).

In equation 7, the coefficients of the delay expression of each gate become r.v.s, and are represented as $a_{0i}(\vec{\Omega})$ and $a_{1i}(\vec{\Omega})$. In equation 5 the delay of gate i also becomes a r.v.:

$$d_i(\vec{x}, \vec{\Omega}) = a_{0i}(\vec{\Omega}) + a_{1i}(\vec{\Omega}) \frac{\sum_{\forall j} e^{x_j}}{e^{x_i}}$$

B. Minimizing BYL: Mathematical Formulation

Under fabrication variability our objective to minimize the BYL can be formulated in terms of $\vec{x}$ (defined in section III-C) as:

$$Subject \quad to: \begin{cases} Minimize \quad BYL(\vec{x}) \\ t_j + d_i(\vec{x}, \vec{\omega_0}) \leq t_i \quad \forall j \in fanin(i); \forall i \\ t_i(y_i) \leq T_{cons} \quad \forall i \in PO \\ s_{min} \leq e^{x_i} \leq s_{max} \qquad \forall gate \quad i \end{cases} \tag{9}$$

In the above formulation $\vec{\omega_0}$ represents the nominal value of $\vec{\Omega}$ assuming no variations. The delay of each gate ($d_i(\vec{x}, \vec{\omega_0})$) is also at its nominal value. The above formulation therefore ensures that T_{cons} is satisfied in the nominal case.

If the goal is to also have a small area, an upper bound on the overall area can be added as a new constraint: $\sum c_i e^{x_i} \leq A_{max}$.

C. A Two-Stage Stochastic Programming Formulation

In the above formulation BYL is a function of $\vec{x}$. To elaborate the consideration for variability in the objective function, let us define the following r.v.:

$$V(\vec{x}, \vec{\Omega}) = \begin{cases} T(\vec{x}, \vec{\Omega}) - T_{cons}; & T \geq T_{cons} \\ 0; & else \end{cases} \tag{10}$$

where $T(\vec{x}, \vec{\Omega})$ is a r.v. that represents the delay of the design. This r.v. depends on both the gate size vector $\vec{x}$ and also the random field $\vec{\Omega}$. The r.v. $V(\vec{x}, \vec{\Omega})$ represents the degree of violating T_{cons}. For a given value of $\vec{x}$, the pdf of V can be written in terms of the pdf of the delay of the circuit $f_T(t)$:

$$f_V(v) = \begin{cases} f_T(t); & v > 0 \\ \int_{-\infty}^{T_{cons}} f_T(t)dt; & v = 0 \end{cases} \tag{11}$$

Note that both $f_T(t)$ and $f_V(v)$ are functions of $\vec{x}$. Now the objective in equation 2 can be expressed in terms of V as:

$$BYL(\vec{x}) = \int_{T_{cons}}^{\infty} (t - T_{cons}) f_T(t)dt = \int_{-\infty}^{\infty} v f_V(v)dv = E[V] \tag{12}$$

Since both $f_T(t)$ and $f_V(v)$ are functions of $\vec{x}$, so will BYL be. Also as illustrated, minimizing the BYL can be thought of minimizing the expected value of violating the timing constraint.

Now let $v(\vec{x}, \vec{\omega})$ be the value for V for a given $\vec{x}$ and a sample $\vec{\omega}$ from the field of uncertainty. Equation 12 can be written as:

$$BYL(\vec{x}) = \int_{-\infty}^{\infty} v(\vec{x}, \vec{\omega}) f_{\vec{\Omega}}(\vec{\omega}) d\vec{\omega} \tag{13}$$

where $f_{\vec{\Omega}}(\vec{\omega})$ is the pdf of $\vec{\Omega}$. Note that this is just another way of understanding BYL. No approximation has been done and no assumption has been made on the nature of the variabilities and their correlations. Therefore equation 13 states that for a known $\vec{x}$ the corresponding $BYL(\vec{x})$ can be found by finding the $E[V(\vec{x}, \vec{\Omega})]$ for all values $\vec{\omega}$ of $\vec{\Omega}$.

Conceptually $v(\vec{x}, \vec{\omega})$ is the degree of violating the delay constraint for a given choice of $\vec{x}$ and a sample $\vec{\omega}$. This itself can be written as a convex program as follows:

$$Subject \quad to: \begin{cases} v(\vec{x}, \vec{\omega}) = Minimize \quad q \\ t_j + d_i(\vec{x}, \vec{\omega}) \leq t_i \quad \forall j \in fanin(i); \forall i \\ t_i \leq T_{cons} + q \quad \forall i \in PO \\ q \geq 0 \end{cases} \tag{14}$$

Solving this formulation results in the arrival times of the gates t_i with the gate delays $d_i(\vec{x}, \vec{\omega})$. The optimal value of q denoted by q^* is the degree of delay violation for a fixed $\vec{x}$ and $\vec{\omega}$.

This falls within the classic formulation of *Two-Stage Stochastic Programming* [13]. The optimization problem given by 9 is called the first stage problem and the one given by equation 14 is called the second stage problem. The region of feasibility for the first stage problem is a convex set (since it simply comprises of a set of convex function constraints). The objective BYL is the expected value of a random variable V which depends on ($\vec{x}$ and $\vec{\omega}$) according to the optimization set of 14. In the next subsection we will prove that $E[V]$ is a convex function of $\vec{x}$. In doing so we will extend the classic Two-Stage Stochastic Programming theory to incorporate convex first and second stage problems. The traditional theory was valid only for linear programs [13].

Please note that our presented formulation does not make any specific assumptions about the distribution of $\vec{\Omega}$ and the correlation of components of $\vec{\Omega}$ (such as L_{eff} and T_{ox}).

D. Proof of Convexity of the Optimization Set

In this section we will prove that the formulation of the inequalities of 9 is convex. To do this it is sufficient to show the optimization's objective ($BYL(\vec{x})$) is convex, as the constraints in equation 9 can be represented in an exponential form similar to section III-C and therefore will be a convex set [16].

Theorem: $BYL(\vec{x})$ is convex.

Proof: As shown in equation 13, $BYL(\vec{x}) = E[V(\vec{x}, \vec{\Omega})]$. The $E[.]$ can be thought of the weighted summation of all the samples $v(\vec{x}, \vec{\omega})$ of V. The weights are the probability values $f_{\vec{\Omega}}(\vec{\omega})$ that are always positive. Therefore we will show that any $v(\vec{x}, \vec{\omega})$ is individually a convex function of $\vec{x}$ to conclude that the $BYL(\vec{x})$ is convex, because the summation of positively weighted convex functions is convex.

To show $v(\vec{x}, \vec{\omega})$ is a convex function we need to show for $\vec{x_1}$ and $\vec{x_2}$, the following inequality holds (for $0 \leq \theta \leq 1$) [16]:

$$v(\theta\vec{x_1} + (1-\theta)\vec{x_2}, \vec{\omega}) \leq \theta v(x_1, \vec{\omega}) + (1-\theta) v(\vec{x_2}, \vec{\omega}) \tag{15}$$

where $v(x_1, \vec{\omega})$ and $v(x_2, \vec{\omega})$ are the optimal solutions of the optimization set expressed by the inequalities of 14. The constraints in the inequalities of 14 are written for $\vec{x_1}$ and $\vec{x_2}$ as:

$$\begin{cases} t_j^{(1)} + d_i(\vec{x_1}, \vec{\omega}) \leq t_i^{(1)} \\ t_i^{(1)} \leq T_{cons} + q^{(1)} \\ q^{(1)} \geq 0 \end{cases} \quad \begin{cases} t_j^{(2)} + d_i(\vec{x_2}, \vec{\omega}) \leq t_i^{(2)} \\ t_i^{(2)} \leq T_{cons} + q^{(2)} \\ q^{(2)} \geq 0 \end{cases} \tag{16}$$

Let $\{\vec{t}^{*(1)}, q^{*(1)}\}$ and $\{\vec{t}^{*(1)}, q^{*(2)}\}$ be the optimal solutions of the left and right inequalities respectively. Multiplying the left inequalities by θ and the right ones by $(1-\theta)$ (for $0 \leq \theta \leq 1$) and adding the corresponding inequalities we get:

$$\begin{cases} (\theta t_j^{*(1)} + (1-\theta) t_j^{*(2)}) + (\theta d_i(\vec{x_1}, \vec{\omega}) + (1-\theta) d_i(\vec{x_2}, \vec{\omega})) \\ \leq \theta t_i^{*(1)} + (1-\theta) t_i^{*(2)} \\ \theta t_i^{*(1)} + (1-\theta) t_i^{*(2)} \leq T_{cons} + (\theta q^{*(1)} + (1-\theta) q^{*(2)}) \\ \theta q^{*(1)} + (1-\theta) q^{*(2)} \geq 0 \end{cases} \tag{17}$$

Since $d_i(\vec{x}, \vec{\omega})$ is convex in $\vec{x}$, we have:

$$d_i(\theta \vec{x_1} + (1-\theta)\vec{x_2}, \vec{\omega}) \leq \theta d_i(\vec{x_1}, \vec{\omega}) + (1-\theta) d_i(\vec{x_2}, \vec{\omega}) \tag{18}$$

Therefore inequalities of 17 can be written as:

$$\begin{cases} (\theta t_j^{*(1)} + (1-\theta) t_j^{*(2)}) + d_i(\theta \vec{x_1} + (1-\theta)\vec{x_2}, \vec{\omega}) \\ \leq \theta t_i^{*(1)} + (1-\theta) t_i^{*(2)} \\ \theta t_i^{*(1)} + (1-\theta) t_i^{*(2)} \leq T_{cons} + (\theta q^{*(1)} + (1-\theta) q^{*(2)}) \\ \theta q^{*(1)} + (1-\theta) q^{*(2)} \geq 0 \end{cases} \tag{19}$$

Let us introduce $\vec{x_3} = \theta \vec{x_1} + (1-\theta)\vec{x_2}$ and $\{\vec{t}^{(3)} = \theta \vec{t}^{*(1)} + (1-\theta)\vec{t}^{*(2)}, q^{(3)} = \theta q^{*(1)} + (1-\theta) q^{*(2)}\}$. By replacing these definitions in the inequalities of 19 we will obtain:

$$\begin{cases} t_j^{(3)} + d_i(\vec{x_3}, \vec{\omega}) \leq t_i^{(3)} \\ t_i^{(3)} \leq T_{cons} + q^{(3)} \\ q^{(3)} \geq 0 \end{cases} \tag{20}$$

This implies that for $\vec{x} = \vec{x_3}$, the following set: $\{\vec{t}^{(3)} = \theta \vec{t}^{*(1)} + (1-\theta)\vec{t}^{*(2)}, q^{(3)} = \theta q^{*(1)} + (1-\theta) q^{*(2)}\}$ is a feasible solution to the inequalities of 14. Therefore the optimal solution at $\vec{x} = \vec{x_3}$ must be smaller than (or equal to) $\theta q^{*(1)} + (1-\theta) q^{*(2)}$.

The optimal solution is nothing but $v(\theta \vec{x_1} + (1-\theta)\vec{x_2})$. Therefore, $v(\theta \vec{x_1} + (1-\theta)\vec{x_2}), \vec{\omega}) \leq \theta v(\vec{x_1}, \vec{\omega}) + (1-\theta) v(\vec{x_2}, \vec{\omega})$, and therefore v and consequently $E[V(\vec{x}, \vec{\Omega})]$ are convex in $\vec{x}$.

V. Some Generalizations

A. Generalized Penalty Function

The proof of convexity of our objective outlined in section IV-D, assumed that the penalty of violating the timing constraint is a linear function of the degree of violation (equation 1). If we redefine this penalty as follows:

$$penalty(t) = \begin{cases} f(t - T_{cons}); & t \geq T_{cons} \\ 0; & else \end{cases} \tag{21}$$

where f is any convex function, then the convexity of the new BYL still holds and optimality can still be achieved. We omit the proof for brevity.

B. Minimizing the Yield-Loss

The previous few sections discussed optimal minimization of BYL. From our experiments we found that there was a high degree of correlation between optimizing BYL and YL. In fact our approach could be used as a heuristic for optimizing YL. But there are some important results that can be proved about the optimality of YL as illustrated below:

Theorem: The optimal BYL will be 0 iff the optimal YL is 0.

Proof: Let us suppose we have a solution for which $BYL = 0$. Referring to equation 12, this can happen only if $f_V(v) = 0$ for all v greater than (not equal to) 0. This means that the pdf of the timing of the circuit (for the given gate sizes) lies entirely within the timing constraint. Thus $YL = 0$. Now let BYL be more than zero, therefore $f_V(v)$ must have a positive value for some v greater than 0. Therefore, some part of the timing pdf must be greater than T_{cons}. Thus YL cannot be zero.

This is an important result, since by optimizing BYL we can 1)achieve a solution for which $YL = 0$, 2) or by looking at the optimal value of BYL check if a solution with $YL = 0$ exists.

VI. Solving the Convex Formulation

In the previous sections we proved that our proposed formulation to minimize the BYL is convex. This means that our formulation is optimally solvable using the convex optimization techniques. We used the Kelley's Cutting Plane technique [16] among other possible methods, which is briefly explained below.

A. Kelley's Cutting Plane Algorithm

Kelley's algorithm is an iterative approach. At each iteration a linear lower bound of the convex objective is generated. This lower bound together with the lower bounds of the previous iterations develop a piecewise linear lower bound on the objective function. As the number of iterations increase, the linear lower bounds of the previous iterations converge to the accurate objective. At any iteration k, the objective function represented by the piecewise linear lower bounds is optimized while satisfying the feasibility criteria of the constraints. This gives us a solution vector x_k. At this point a new linear lower bound is computed for the true objective function and the entire process is repeated. These steps can be summarized in Algorithm 1:

Initially at Step 1 a feasible solution ($\vec{x_1}$) is found for the inequalities of 9. Kelley's algorithm follows an iterative approach: In the k^{th} iteration, the lower bound at $BYL(\vec{x}_{k-1})$ found in the previous iteration is used to generate a new solution $\vec{x}_k$. This lower bound is generated as follows: We find the sub-gradient $\alpha_k + \vec{\beta}_k.\vec{x}$ of the BYL function such that at $\vec{x} = \vec{x}_{k-1}$, $BYL(\vec{x}_{k-1}) = \alpha_k + \vec{\beta}_k.\vec{x_{k-1}}$ where $\vec{\beta}_k$ is conceptually the slope of the BYL function at $\vec{x} = \vec{x}_{k-1}$. By definition this sub-gradient is the linear lower bound of the BYL function. A new $\vec{x}_k$ is now chosen as follows: A new variable l_k is incorporated in the optimization framework which is constrained to be larger than all the lower bounds found so far. The actual objective is then replaced by l_k which approximates the BYL (Step 4 of Algorithm 1). This gives us a new value for $\vec{x}_k$ and the entire process is repeated till the lower bound approximation and the upper bound are within a user specified range of tolerance (note that each $\vec{x}_k$ corresponds to an upper bound $BYL(\vec{x_k})$). This approach *provably* reaches the optimal solution in convex optimization [16].

Next we will explain how the statistical timing analysis (STA) can be integrated as a useful tool in our formulation, and in the Kelley's Cutting-Plane algorithm to find the lower bound on BYL.

Please note that in case that the optimization of area and/or power is necessary, new constraints can be added to our formulation that bound the overall area or power. These can be expressed as convex constraints which allows the use of Kelley's Cutting-Plane algorithm to solve the new optimization formulation.

Algorithm 1: Kelley's Cutting Plane Algorithm

Step 1: *Initialize*
Let $\epsilon > 0$ and $\vec{x}_1$ be a feasible solution satisfying the constraints.
Let $k \leftarrow 0$ and define $l_0(\vec{x}) = -\infty$, $u_0(\vec{x}) = \infty$.

Step 2: *Set* $k \leftarrow k + 1$

Step 3: *Define the Lower Bound at* $\vec{x}_k$
Evaluate α_k and $\vec{\beta}_k$ such that $l_k \geq \alpha_k + < \vec{\beta}_k, \vec{x} >$:
$\alpha_k = BYL(\vec{x}_k) - \vec{\beta}_k \vec{x}_k \qquad \vec{\beta}_k = \frac{\partial BYL(\vec{x})}{\partial \vec{x}}|_{\vec{x}_{k-1}}$

Step 4: *Update the Optimization Set*
Add the following to the existing set of constraints:
$l_k \geq l_{k-1} \qquad l_k \geq \alpha_k + < \vec{\beta}_k, \vec{x} >$
Update the objective function to $Minimize\ l_k$.

Step 5: *Solve the Optimization to get* $\vec{x}_k$ *and Update the Bounds*
Let upper bound $u_k = Min\{u_{k-1}, BYL(\vec{x}_k)\}$ and lower bound l_k.

Step 6: *Stopping Rule*
Stop if $u_k - l_k \leq \epsilon$, otherwise go to Step 1.

B. Integration with STA

1) Computing the BYL: Given a gate-level circuit, statistical timing analysis can be used to efficiently compute the BYL. In section IV-C we explained how BYL can be computed parametrically over all samples $\vec{\omega}$ in $\vec{\Omega}$ and for a particular set of gate sizes using equation 13. It can also be equivalently obtained using equation 12. This is equivalent to doing an STA (for a given choice of gate sizes) on the circuit and then evaluating the expected value of violating the timing constraint in order to find the BYL (equation 12). Assuming variability in $\vec{\Omega}$, STA provides the spread of delay at the primary outputs (essentially the pdf $f_T(t)$) for a given $\vec{x}$. This STA can be done based on any possible approach such as [5] and [18].

2) Computing the Lower Bound in Kelley's Algorithm: The linear lower bound on BYL is expressed as $\alpha_k + < \vec{\beta}_k, \vec{x} >$ in Algorithm 1. As expressed in step 3 of the algorithm, $\vec{\beta}_k$ is found by evaluating the slope of the $BYL(\vec{x})$ at $\vec{x}_{k-1}$. The coefficient α_k is found such that $BYL(\vec{x}_{k-1}) = \alpha_k + < \vec{\beta}_k, \vec{x}_{k-1} >$. Therefore in order to find the lower bound, it is sufficient to show the computation of $\vec{\beta}_k$.

Finding the sub-gradient of a non-differentiable function is an important research problem. Many techniques have been proposed that can approximate the sub-gradient. In this paper we will be using the finite-difference method [16].

The vector $\vec{\beta}_k = \{\beta_1; \beta 2; ...; \beta_n\}$, where β_i is the projection of $\vec{\beta}_k$ with respect to component x_i (or $\beta_i = \frac{\partial BYL(\vec{x})}{\partial x_i}|_{\vec{x}_{k-1}}$). In other words β_i expresses the sensitivity of the objective function with respect to x_i. We approximate this sensitivity as below:

$$\beta_i = \frac{BYL(\{x_1; ...x_i; ...; x_n\}) - BYL(\{x_1; ...x_i + \Delta x_i; ...; x_n\})}{\Delta x_i}|_{\vec{x}_{k-1}} \tag{22}$$

Given an $\vec{x_{k-1}}$ vector, the sensitivity β_i is found using equation 22. Computation of BYL in the above equation can be done using STA as explained in the previous subsection. Therefore computation of β_i in the above equation requires doing two STAs for each component $\vec{x_{k-1}}$ vector, assuming x_i for gate i is slightly changed. The paper [10] proposes ways that allows the sensitivity to be more efficiently computed. Once $\vec{\beta}_k$ is found, α_k and consequently the lower bound are determined.

Note that the STA at any of these stages can be done using any of the proposed techniques in the literature such as [5] or [18], and can assume any distribution for $\vec{\Omega}$ *and any correlation model for its components*.

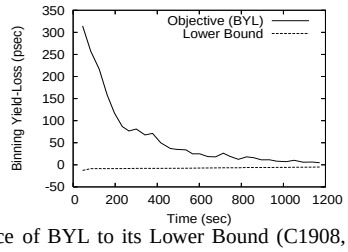

Fig. 2. Convergence of BYL to its Lower Bound (C1908, T_{cons} =3500 psec)

VII. Results

Our experiments were conducted on the ISCAS bench suite. We initially placed each benchmark and generated correlation data between different gates based on the model of [18]. We assumed a variability in the L_{eff} of each device with a Gaussian distribution with a mean equal to the nominal value and a 12% standard deviation from the mean. We determined the convex expression for the delay of each gate as a function of its size assuming a 90nm technology (for which we got the information from [19]).

We implemented our proposed method in the SIS framework and used the MOSEK [6] convex optimization tool. In the proposed method using the Kelley's algorithm, we integrated STA method of [3] to compute BYL as explained in section VI-B.

Figure 2 shows the values of our objective BYL and its lower bound as iterations progress. At each iteration the value of the objective corresponds to the upper bound of the optimal. Kelley's algorithm iteratively improves the lower bound till the lower and upper bounds converge. This algorithm guarantees optimality.

In order to make comparison with other methods, we implemented a sensitivity-based approach as well as a worst-case method. The sensitivity method had a framework as in [1] or [4]. In this method initially all the gates are set to their minimum size. The sensitivity-based method is a greedy iterative approach, in which at each iteration the most sensitive gate is determined and sized up. The most sensitive gate is the one that results in the maximum change in the objective due to a small change in its size. For comparison of this method with ours we set the objective of the sensitivity-based approach to be the BYL.

We also implemented a worst-case deterministic approach. The worst-case approach had a convex optimization framework similar to [17]. However the delay expression for each gate was computed assuming the value of L_{eff} is fixed at its worst ($\mu + 3\sigma$). In this approach we set the optimization objective to be minimization of the arrival time at the primary output nodes. We also added a new constraint to impose an upper bound on the maximum area of this approach. In order to make comparison with our proposed method, we set this maximum area of the worst-case approach to be the area of the optimal solution generated by our approach.

Table I compares the BYL and area of these three methods for two different timing constrains for each benchmark. One of these timing constraints is more stringent than the other one. For the stringent timing constraint, the deterministic approach could not generate any solution as it was too pessimistic in approximating the delay of each gate and consequently of the timing constraint. For the more relaxed timing, the worst-case approach however was able to generate solutions of good quality comparable to our method. Compared to sensitivity-based approach, we achieved an average of 72% improvement in the BYL with only a 6% area overhead given the stringent timing constraint. We also generated better solution when the timing constraint was relaxed.

bench	T_1							T_2						
	T_{cons}	Sensitivity		Worst-Case		Kelley Convex		T_{cons}	Sensitivity		Worst-Case		Kelley Convex	
		BYL	$Area$	BYL	$Area$	BYL	$Area$		BYL	$Area$	BYL	$Area$	BYL	$Area$
C17	210	21.60	353	N/A	N/A	6.83	369	300	0.00	365	0.00	321	0.00	342
C432	2500	252.47	10446	N/A	N/A	45.76	11504	3000	46.58	8908	1.73	8789	1.65	8789
C499	2300	32.73	15279	N/A	N/A	32.36	21869	2700	9.59	14408	1.46	13920	1.42	14684
C880	3150	226.05	13502	N/A	N/A	19.92	13336	3500	45.82	13935	1.80	13353	1.65	13485
C1355	2050	105.28	15821	N/A	N/A	17.79	21410	2300	32.73	15279	2.50	14977	1.59	14977
C1908	3000	327.94	18624	N/A	N/A	29.38	21812	3500	101.56	17139	1.95	18009	1.32	18009
C3540	4000	270.00	37547	N/A	N/A	76.67	37574	5500	8.66	36778	3.73	36728	3.72	36728
C5315	4000	105.92	50192	N/A	N/A	61.19	50138	5500	9.45	49661	8.59	49584	8.32	49596
C6288	15000	323.41	89201	N/A	N/A	181.65	88503	23000	8.77	87750	8.77	87750	8.77	87750
Ave.		185.04	27884	N/A	N/A	52.35	29613		29.24	27135	3.39	27047	3.16	27151

TABLE I

COMPARISON OF BINNING YIELD-LOSS (IN PSEC) AND AREA

bench	T_{cons}	Sensitivity		Kelley Convex	
		#itera.	time	#itera.	time
C17	210	73	0.06	4	1.33
C432	2500	1636	894.18	30	438.13
C499	2300	390	739.27	13	537.60
C880	3150	339	475.48	7	150.11
C1355	3000	390	772.82	7	216.99
C1908	3500	711	1585.47	31	1172.77
C3540	4000	120	2004.22	5	776.28
C5315	4000	164	5127.82	7	1870.41
C6288	15000	138	13616	4	1802.47

TABLE II

COMPARISON OF TOTAL RUN-TIME (SEC) AND NUMBER OF ITERATIONS

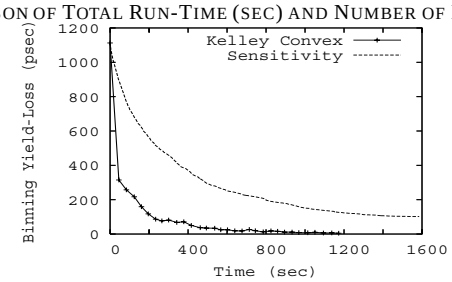

Fig. 3. Binning Yield-Loss vs. Time (C1908, T_{cons} =3500 psec)

Figure 3 shows the optimization of objective over time using our approach compared to the sensitivity-based method for C1908. It can be seen that our approach has clearly a faster convergence rate. In fact as the run-times are reported in table II, our method achieves an average of 2.69 speed up due to fewer number of iterations. Although each individual iteration takes longer in our method (as a convex optimization set needs to be solved at each iteration in our case), but due to the very few number of iterations, the overall run-time will be much smaller.

We also compared the traditional Yield-Loss of the solution generated by our approach to a sensitivity-based approach in which the most sensitive gate was defined as the one with maximum change in Yield-Loss due to the change in its size. Our method also improves the Yield-Loss on average by 61%.

Finally figure 4 shows the curve generated by our approach between the area and BYL. Each point corresponds to the solution of an iteration of Kelley's algorithm. It can be seen that as the iterations progress, increase in area results in a decrease in BYL.

VIII. CONCLUSIONS

In this paper we presented a convex formulation to optimally minimize the Binning Yield-Loss in high performance applications. This optimization is done using gate sizing under fabrication variability. Our optimal approach results in an average 72% improvement in the BYL with a small area over head of on average 6% for stringent timing constraints. The solutions generated by our approach also improves the traditional Yield-Loss by 61% compared to sensitivity-based methods.

bench	T_{cons}	Sensitivity	Worst-Case	Kelley Convex
C17	210	0.75	N/A	0.40
C432	2500	0.76	N/A	0.28
C499	2300	0.23	N/A	0.23
C880	3150	0.72	N/A	0.15
C1355	2050	0.53	N/A	0.15
C1908	3000	0.82	N/A	0.18
C3540	4000	0.71	N/A	0.24
C5315	4000	0.33	N/A	0.18
C6288	15000	0.64	N/A	0.39

TABLE III

COMPARISON OF YIELD-LOSS

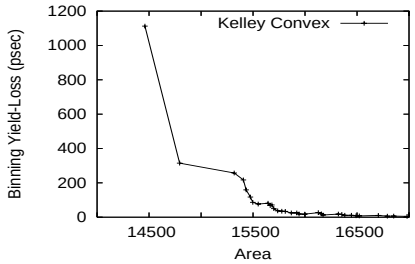

Fig. 4. BYL vs. Area Generated at Different Iterations of Kelley's Algorithm

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. Raychowdhury, S. Ghosh, and K. Roy. A Novel On-chip Delay Measurement Hardware for Efficient Speed-Binning. In *IOLTS*, July 2005.

[3] C. Visweswariah , K. Ravindran , K. Kalafala , S. G. Walker , and S. Narayan. First-Order Incremental Block-Based Statistical Timing Analysis. In *DAC*, 2004.

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

[5] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah. Parameterized Block-Based Statistical Timing Analysis with Non-Ga ussian Parameters and Nonlinear Delay Functions. In *DAC*, 2005.

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

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

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

[9] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. Digital Integrated Circuits. Prentice Hall.

[10] K. Chopra, S. Shah, A, Srivastava, David Blaauw, and D. Sylvester. Parameteric Yield Maximization using Gate Sizing based on Efficient Statistical Power and Delay Gradient Computation. In *ICCAD*, Nov. 2004.

[11] M. Mani, A. Devgana, and M. Orshansky. An Efficient Algorithm for Statistical Minimization of Total Power under Timing Yield Constraints. In *DAC*, July 2005.

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

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

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

[15] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, V. De. Parameter Variations and Impacts on Circuits and Microarchitecture. In *DAC*, 2003.

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

[17] 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.

[18] V. Khandelwal, A. Srivastava. A General Framework for Accurate Statistical Timing Analysis Cons idering Correlations. In *DAC*, 2005.

[19] 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*, 2000.