Design Time Body Bias Selection for Parametric Yield Improvement

Cheng Zhuo, Yung-Hsu Chang, Dennis Sylvester, David Blaauw
EECS Department, University of Michigan, Ann Arbor, MI 48109
{czhuo, yunghsuc, dennis, blaauw}@eecs.umich.edu

Abstract— Circuits designed in aggressively scaled technologies face both stringent power constraints and increased process variability. Achieving high parametric yield is a key design objective, but is complicated by the correlation between power and performance. This paper proposes a novel *design time* body bias selection framework for parametric yield optimization while reducing testing costs. The framework considers both inter- and intra-die variations as well as power-performance correlations. This approach uses a feature extraction technique to explore the underlying similarity between the gates for effective clustering. Once the gates are clustered, a Gaussian quadrature based model is applied for fast yield analysis and optimization. This work also introduces an incremental method for statistical power computation to further reduce the optimization complexity. The proposed framework improves parametric yield from 39% to 80% on average for 11 benchmark circuits while runtime is linear with circuit size and on the order of minutes for designs with up to 15K gates.

I. Introduction

Semiconductor technologies are characterized by trends of ever shrinking feature dimensions. As a result, process variability has become more prominent in sub-nanometer regime designs and poses a major challenge to improving circuit performance and reducing leakage [1]. Given the large contribution of leakage power to total power in recent technology nodes, delay and power are now negatively correlated across process corners [2]. In such a scenario, high speed parts are also very high leakage, imposing a two-sided constraint on the feasible region of delay and leakage for parametric yield optimization. This ultimately causes a significant yield loss of manufactured dies in modern integrated circuits [2].

To address this issue, numerous pre- and post-silicon statistical optimization methods have been proposed to mitigate yield loss due to process variability [2]–[14]. However, several of these approaches neglect the power-performance correlation by treating the impact of delay and power separately [8], [9]. Works in [10], [11] investigated nonlinear optimization by assuming that gate sizes are continuous. Thus, they could either apply a simplified power yield model or transform yield maximization to slack minimization. These techniques only mitigate the variation indirectly, rather than performing true yield maximization due to the approximated formulation or neglect of power-performance correlation. Moreover, growing circuit size further restricts the efficacy of traditional pre-silicon techniques (gate sizing or dual-threshold voltage assignment) in guaranteeing reliable circuit operation with desired parametric yield [2], [10], [11].

Beyond these pre-silicon approaches, several post-silicon techniques have been proposed for design optimization [12]–[14]. Among these, adaptive body biasing (ABB) is a promising post-silicon technique due to its flexibility [15]. Traditionally ABB is used to tune each chip individually after chip fabrication and testing, as shown in Figure 1. Thus, conventional post-silicon ABB is limited by (1) routing/control overhead to adjust devices/gates at a very fine grained level

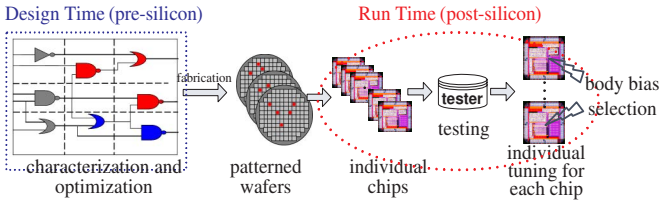


Fig. 1. Flow of post-silicon ABB tuning employed by [12], [13]

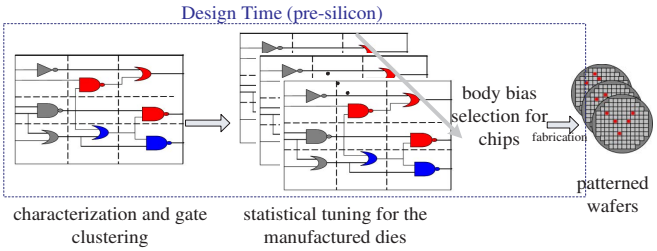


Fig. 2. Flow of the proposed pre-silicon framework for yield maximization

[12], [15] and (2) increased post-fabrication testing costs to determine the optimal body voltage. To reduce overhead to a feasible level, [12] presented a heuristic clustering method, in which gates are grouped at design time into a small set of clusters and controlled by one body bias within the cluster. Mani, *et.al.*, suggested the coordination of pre-silicon (gate sizing) and post-silicon (ABB) techniques, and formulated this as a robust programming problem. Similar to [10], [11], both methods [12], [13] separate the correlation between power and delay and do not evaluate the true parametric yield (joint yield of power and delay). Above all, tuning in [12], [13] is carried out entirely as a post-silicon step. Clearly, such a strategy incurs large post-silicon testing costs.

In order to reduce testing overhead, this paper presents a low-cost *pre-silicon* body bias selection framework to maximize parametric yield. This framework considers both inter- and intra-die variation as well as power-performance correlation. The major difference between the proposed framework and traditional ABB is that our work does not require individual tuning of each chip during post-silicon testing to select the body bias to be applied. Instead, as shown in Figure 2, our framework optimizes and fixes body bias during *design time* to improve the yield of manufactured dies. Once the bias levels are chosen, simple and compact circuits can be readily designed to provide the chosen reference voltages [16], [17].

Unlike post-silicon ABB, in which bias voltages for each chip are chosen in a deterministic way (since measurement results for a particular manufactured die are known and deterministic), pre-silicon body bias selection framework must statistically incorporate the variability to tune the ensemble of all chips simultaneously. This provides the following advantages:

- *Low testing time and cost* Pre-silicon body bias selection statistically optimizes the body bias for the manufactured dies, eliminating the testing cost increases associated with the post-silicon approaches.

978-1-4244-5767-0/10/$26.00 2010 IEEE

- *High flexibility* Design-time body bias selection can be easily implemented using on-chip reference voltages [16], [17] and hence has continuous-domain design variables. This appealing feature is in contrast to most traditional pre-silicon techniques such as gate sizing or dual V_{th}, which have discrete domain design variables.
- *Good scalability* Our pre-silicon body bias selection uses a small number of gate clusters (where each cluster is assigned to a different bias voltage), enabling a theoretically rigorous formulation of parametric yield as well as scalability to large circuits.

The proposed framework consists of two phases. We first determine the body bias profiles for each gate, which reflects the preferred body biases across an expected representative set of dies based on process variability models. Then a feature extraction technique is applied to those profiles to efficiently cluster the gates. The general concept behind gate clustering is to group gates with statistically similar behavior. The heuristic approach in [12] uses an affine function of mean, standard deviation, and correlation coefficients to determine similarity. However, this method requires large runtime and memory consumption for the greedy search and correlation matrix construction. Also, the chosen weights of the affine function may not be globally applicable across all circuit topologies. In addition, this approach discards most information from the original body bias profiles and therefore is not robust with respect to outliers. As a result, in our framework we propose a general and scalable clustering method based on feature extraction, without any dependence on empirical parameters. The feature extraction technique projects the original body bias profiles of the gates to a reduced set of features (feature vector) [18]. The feature vectors contain the general characteristics of the profiles and can be computed efficiently for body bias profile similarity comparison. In particular, the comparison is made by computing the distance of two feature vectors and grouping together the gates with closer distance.

After clustering the gates, the second phase formulates the body bias selection problem as a small-sized unconstrained nonlinear programming (NLP) problem. The NLP is solved by a large-scale optimizer (Lancelot [19]) with a fast yield evaluation scheme called Gaussian quadrature to compute the objective yield. An incremental method is also introduced to quickly compute the probability density function (PDF) of leakage power. Experiments show that the proposed framework can optimize a circuit with 14592 gates within 20 minutes to achieve 52 point yield improvement. For eleven circuits of different sizes, yield is improved from 39% to 80% on average. The key contributions of this paper are listed below:

(1) We present a low-cost *pre-silicon* framework to select body bias at design time for direct parametric yield optimization. We show that the proposed pre-silicon approach retains the majority of the yield benefits of more complex die-specific post-silicon ABB approach as well as a higher flexibility than traditional pre-silicon methods like dual-V_{th} and gate sizing. The framework considers both process variation and the correlation between performance and power.

(2) To effectively cluster the gates, a feature extraction-based technique is employed. We apply a Haar wavelet transform to extract the features from the statistical body bias profile of each gate. Then a k-median-like algorithm is presented to optimally cluster the gates with similar features.

(3) In the optimization framework, the yield objective is repeatedly computed. We present a fast and accurate method using Gaussian quadrature to compute the yield in the form of a bi-variate normal integral. An incremental technique for statistical power computation is also introduced to further reduce gradient computation complexity.

II. Feature Extraction-Based Gate Clustering

Gate clustering is a critical step in practical ABB approaches. Once the clustering is performed, the body voltage of the cluster is determined so that its most timing critical gates meet the overall circuit delay constraint, indicating that most gates in a cluster will end up requiring a larger (more forward) body bias than necessary. It is therefore vital to cluster gates with similar body bias characteristics to minimize optimality loss. This section discusses a new feature extraction-based technique for gate clustering.

A. Leakage Power and Delay Modeling

Body biasing uses the body effect to modulate the threshold voltage of a MOSFET. Since the analytical expressions that govern the impact of body bias on delay and leakage at the gate level are fairly complex, we adopt the quadratic leakage model and linear delay model from [12]. SPICE simulation validates that a quadratic model for leakage and linear model for delay achieve an average error of 5.9% and 1.5% respectively across a 90nm standard cell library. Change of leakage and delay with body bias can then be expressed by [12]:

$$\Delta L_i(v_{b,i}) = L_{0,i}(p_{0,i} + p_{1,i}v_{b,i} + p_{2,i}v_{b,i}^2) \quad (1)$$

$$\Delta D_i(v_{b,i}) = D_{0,i}(d_{0,i} + d_{1,i}v_{b,i}) \quad (2)$$

where $\Delta L_i(v_{b,i})$ and $\Delta D_i(v_{b,i})$ are the leakage and delay change, $L_{0,i}$ and $D_{0,i}$ are the nominal leakage and delay value with zero body bias, $v_{b,i}$ is the body voltage for gate i, $p_{j,i}$ (j=0,1,2) are the fitting parameters for the quadratic leakage model, and $d_{j,i}$ (j=0,1) are the fitting parameters for the linear delay model.

B. Design Space Exploration

We assume that each circuit constitutes its own unique design space subject to certain parameter variations. Our variation formulation incorporates both inter- and intra-die variations [3], [4]. To identify the difference in gates, we first generate multiple "die samples" following certain variations for the given circuit in a Monte Carlo fashion. For each sample circuit we assume each gate can be tuned individually and construct the deterministic quadratic programming (QP) to find the optimal body bias of each gate for leakage minimization [12]:

$$\textbf{Minimize} \quad \sum_j \Delta L_j(v_{b,j}) \quad (3)$$

$$\textbf{Subject to} \quad AT_{PI} = 0, AT_{PO} < Target \quad (4)$$

$$AT_{i,j} + D_j^{ABB} < AT_{o,j} \qquad for \ \forall j \quad (5)$$

$$D_j^{ABB} = D_{0,j} - \Delta D_j(v_{bj}) \qquad for \ \forall j \quad (6)$$

where AT is the arrival time of the signal on a wire, subscripts "i" and "o" denote input and output, and D_j^{ABB} denotes the delay of a biased gate. The first constraint limits the arrival times at primary input (PI) to be zero and the arrival times at primary output (PO) to be less than the design target. The second and third constraints indicate that the delay at the output of each gate should be at least equal to the arrival

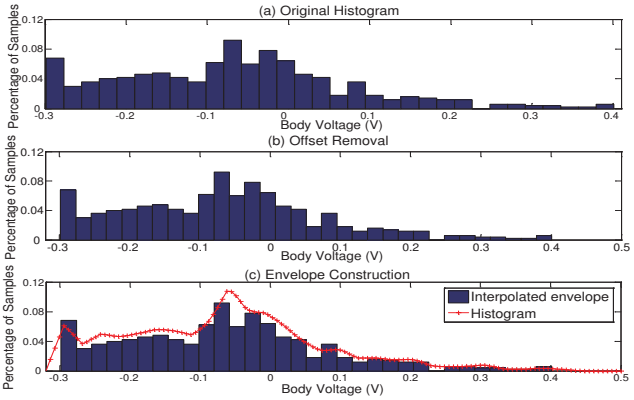


Fig. 3. Pre-processing procedure: (a) original histogram (b) offset removal for body voltage (x-scale) (c) envelope construction

time at each of its inputs plus the delay of the gate D_j^{ABB}. This QP can be efficiently solved by CPLEX [20] to obtain the optimal body voltage for a particular sample (die) in the design space. The histogram of the optimal body bias for a gate sheds insight on the statistical behavior of the gate in this design space. We can then distill information from this histogram in determining which gates should share a common body potential. This is the critical clustering step (described below). The design space exploration is executed only once for each unique design before the optimization stage.

C. Feature Extraction

A straightforward approach to clustering is to group together the gates with similar body bias profiles. However, it is difficult to define "similarity" quantitatively. It is impractical and inefficient to simply use the complete profiles to cluster the gates because of their large sizes and the resulting noise sensitivity in the distributions. As previously stated, [12] suggested a weighted affine function of mean, deviation and correlation to judge the similarity between the gates. However, construction of the correlation matrix between the gates leads to a memory complexity of $O(N_g^2)$ for N_g gates and limits its applicability. Beyond these runtime concerns, the greedy search is heavily dependent on the carefully chosen weights and the order of the gates to be visited. This makes the method sensitive to outliers and allows gates to be misgrouped. Furthermore, since the affinity of the non-grouped gate to the cluster is computed by taking the average of the weights, highly deviated data may have a disproportionate impact on the average and lead to poor selections. We therefore present a faster and more robust clustering strategy in our framework.

We employ a pattern recognition technique called feature extraction to obtain the main features of the profile while filtering out noise and redundant information. The concept behind feature extraction is to extract the general characteristics of the profiles, maintaining the most common information and discarding outliers. The body bias profile for each gate is then uniquely identified by a feature vector, $\mathbf{v_i} = [x_1, x_2, ..., x_n]^T$ with n features, which are used to measure similarity. To apply the technique across all gates and preserve important information, some pre-processing is performed to build a unified and suitable system. The pre-processing includes two stages:

(1) Offset removal. This stage simply aligns the histograms to the same body voltage intervals, so that voltage ranges are unified for all the gates.

(2) Envelope construction. The original histogram is based on a coarse grid and cannot be directly used. In this stage,

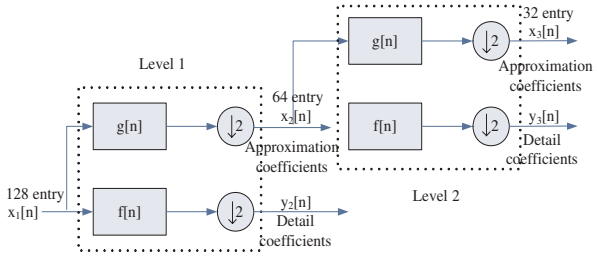


Fig. 4. An example of a two-level Haar wavelet transform. A 128-entry input $x_1[n]$ is reduced to a 32-entry vector.

we apply linear interpolation to map the histogram data to a finer grid and construct the basic shape of the profile envelope. Figure 3(a)-(c) shows the pre-processing procedure for a randomly selected gate in circuit c6288.

Once the body bias profiles are available in the form of unified envelopes, we apply the feature extraction technique to determine the underlying characteristics of each gate. The proposed feature extraction is achieved by Haar wavelet transform. With a one-level Haar transform, the original body bias waveform with n-sample points can be transformed to two n/2-entry vectors (approximation coefficients x_i and detail coefficients y_i):

$$x_i[n] = (x_{i-1}[2n] + x_{i-1}[2n+1])/\sqrt{2} \quad (7)$$

$$y_i[n] = (x_{i-1}[2n] - x_{i-1}[2n+1])/\sqrt{2} \quad (8)$$

The detail coefficients representing the local characteristics are easily disturbed by outliers and hence discarded. The approximation coefficients preserving the general characteristics are then decomposed repeatedly until a feature vector with a required number of features (n/4, n/8, etc.) is obtained. In our work, an eight-entry feature vector is extracted from the body bias profile for each gate. Figure 4 shows a simple example of a two-level Haar transform architecture, where g[n] and f[n] represent (7) and (8), respectively.

Since the approximation coefficients indicate the accumulated activities, the feature vectors naturally embody the mean and variance information of the profiles. Moreover, as two highly correlated gates should exhibit similar body bias profiles and hence similar feature vectors, the correlation between gates is well modeled by the proposed method. Thus, the feature vector preserves more information than the method in [12].

D. Gate Clustering

Gate clustering groups gates with similar behaviors. We here propose the following definition to quantify the similarity of feature vectors.

Definition: The similarity of two feature vectors $\mathbf{v_1}$, $\mathbf{v_2}$ is the cosine of the angle between them:

$$S_{\mathbf{v_1},\mathbf{v_2}} = \cos(\alpha) = \frac{|\mathbf{v_1}^T \mathbf{v_2}|}{\|\mathbf{v_1}\|\|\mathbf{v_2}\|} \quad (9)$$

where $\|\cdot\|$ denotes the Euclidean norm. The use of the angle between vectors provides two main advantages: (1) It correctly measures the distance between two vectors. Since any entry in a feature vector is always non-negative, a larger Euclidean distance is equivalent to a larger angle and hence a smaller $S_{\mathbf{v_1},\mathbf{v_2}}$ (2) The value is normalized and does not depend on any amplitude gains or empirically chosen weights.

Here we present the simplest example of two clusters. N-cluster decomposition will be an extension of the two-cluster case and is discussed in section II-E. In this example we need to classify the gates into two clusters based on their feature

Procedure: *2-Cluster Gate Clustering*
Input: feature vectors for all the gates
Output: clustered circuit
1: Choose the initial seed for each cluster;
2: **For** each gate i with feature vector $\mathbf{v_i}$ **do**
3: Measure the similarity to the centroid of each cluster;
4: Find cluster $j = arg\mathbf{max}(S_{\mathbf{v_i},\mathbf{u_j}})$, j=1 or 2;
5: Put gate i into cluster j;
6: Update the centroid of cluster j;
7: **End for**

Fig. 5. Algorithm for 2-cluster gate clustering

vectors. The initial seed gates for each cluster may be easily assigned, namely the most forward-biased and most reverse-biased gates, which are determined by sorting the mean of the body bias profiles. These two gates should clearly be in separate clusters. The seeds become the initial $centroids$ of the clusters. A *centroid* is defined as a vector that maximizes the sum of similarities of all other points within the cluster to itself. After initial seeds are selected, gates are visited in sequence and their similarities to the centroid of each cluster are computed by (9). Each gate is then placed in the cluster with the highest similarity after which the centroid of the corresponding cluster is updated. This procedure is described in Figure 5 and carried out repeatedly until all gates are classified.

Since computing the centroid using the arithmetic mean is not robust to outliers or noise, we therefore propose a low-cost k-median-like algorithm in this paper to compute the centroid. This strategy circumvents the potential problem of highly deviated data skewing the arithmetic average in [12].

For example, if m gates are contained in the cluster, the centroid is:

$$\mathbf{u} = arg\mathbf{max} \sum_{\mathbf{v_i} \in cluster} S_{\mathbf{u},\mathbf{v_i}}, \quad \mathbf{u} \in \{\mathbf{v_1}, \mathbf{v_2}, \ldots \mathbf{v_m}\} \quad (10)$$

This is a nonlinear discrete optimization problem that is difficult to solve. We therefore employ a two-phase relaxation scheme to tackle this problem. The first phase relaxes the problem to an unconstrained continuous optimization and finds the optimal condition, which is:

$$\mathbf{max} \sum_{\mathbf{v_i} \in cluster} \frac{\mathbf{v_i}^T \mathbf{u}}{\|\mathbf{v_i}\| \|\mathbf{u}\|}, \quad \mathbf{u} \in R^{8\times 1} \quad (11)$$

This can be further simplified to:

$$\mathbf{max} \quad \mathbf{w}^T \mathbf{x} \quad (12)$$

where $\mathbf{w}$ is $\sum_i \mathbf{v_i}/\|\mathbf{v_i}\|$ and $\mathbf{x}$ is a normalized vector $\mathbf{u}/\|\mathbf{u}\|$. The inner product of vector **a** and a normalized vector **b** is the length of projection of **a** on **b**. Thus, the maximum of (12) is reached when vectors $\mathbf{x}$ and $\mathbf{w}$ lie in the same direction, $i.e.$, $\mathbf{x} = \mathbf{w}/\|\mathbf{w}\|$. This is denoted as the optimal condition for centroid selection.

The second phase consists of a local search among gates in the cluster to find the closest match to the optimal centroid found above. This is achieved by computing the similarity between each normalized feature vector and the optimal centroid (using (9)). The vector with the largest similarity is then chosen to be the centroid of the cluster. The algorithm for centroid update is shown in Figure 6. Since a correlation matrix is not required in our algorithm, the memory complexity is reduced from $O(N_g^2)$ to $O(N_g)$.

E. Extension to N Clusters

The 2-cluster gate clustering algorithm in Figure 5 can be extended to an efficient successive clustering algorithm for

Procedure: *Centroid Update*
Input: feature vectors for all the gates in the cluster
Output: centroid of the cluster
1: Compute the optimal condition $\mathbf{v_{opt}}$ of the centroid using (12);
2: **For** each gate i with feature vector $\mathbf{v_i}$ **do**
3: Compute $S_{\mathbf{v_i},\mathbf{v_{opt}}}$ using (9);
4: **End for**
5: Set the gate with the largest $S_{\mathbf{v_i},\mathbf{v_{opt}}}$ as the centroid;

Fig. 6. Algorithm for centroid update

Procedure: *N-Cluster Gate Clustering*
Input: feature vectors for all the gates, number of clusters N
Output: clustered circuit
1: Set n as the number of the cluster;
2: Perform *2-Cluster Gate Clustering* recursively till $n \geq N$;
3: **If** $n > N$
4: Perform recombination repeatedly till n==N;
5: **End if**

Fig. 7. Algorithm for N-cluster gate clustering

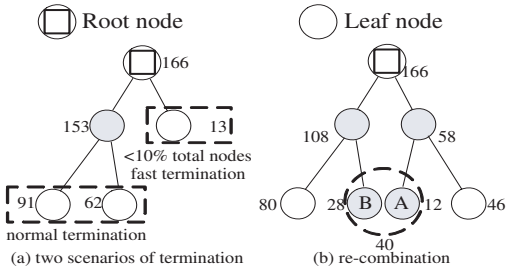


Fig. 8. (a) Two possible scenarios to achieve leaf nodes: fast termination and normal termination; (b) Recombination of the non-leaf nodes. The number beside each node denotes the number of gates in the node

N-clusters. The cluster is recursively bi-partitioned until the number of clusters reaches or exceeds N. The algorithm is outlined in Figure 7. We use a binary-tree data structure to model the successive clustering of the gates. The root node of the tree contains all the gates in the circuit whereas a leaf node represents the resulting cluster without any children nodes.

There are two possible scenarios for creating a leaf node: (1) Normal termination. When the total number of leaf nodes and non-leaf nodes reaches the required N, all the nodes at the lowest level become leaf nodes; (2) Fast termination. For a node with no more than 10% of the total gates[1], we consider this to be a leaf node without further decomposition. A typical example is shown in Figure 8(a) in which N=3. On the second level, the node on the right contains fewer than 10% of the total gates and is immediately considered to be a leaf without further decomposition (fast termination). The node on the left with 153 gates is further decomposed to two non-leaf nodes on the third level. Since the total number of non-leaf nodes and leaf nodes has reached the required number N (=3), the two non-leaf nodes are then considered as leaves and terminated (normal termination).

If the number of the nodes (including leafs and non-leafs on the bottom level) exceeds "N", which commonly occurs, a recombination stage is employed. In this case the node with the fewest gates (node A in Figure 8(b)) is recombined to either node A's sibling node or the node whose parent is the sibling of node A's parent. The candidate with fewer gates will be chosen (node B in the figure). Since the number of clusters is limited in practice, the algorithm in general is terminated within 3-4 iterations.

[1]In practice the number of the clusters will not exceed 10

III. Design-Time Body Bias Selection

A. Statistical Delay and Leakage Models for Biased Gates

This section describes the statistical gate-level models for the parametric yield optimization framework. Following the examples of [2]–[4], each process parameter can be transformed to a linear combination of m independent gaussian random variables (z_j) and the random residual R from principal component analysis (PCA). Both delay and log of leakage can then be canonically expressed by two gaussian random variables:

$$\begin{aligned} D &= D_0 + \sum_{i=1}^{m} a_i z_i + a_{m+1} R \\ \ln(L) &= V_0 + \sum_{i=1}^{m} b_i z_i + b_{m+1} R \end{aligned} \tag{13}$$

where a_i and b_i are the corresponding coefficients obtained from PCA [3], [4]. Assuming the gate is biased at a particular body voltage v_b, with the models in (1)-(2), gate delay and log of leakage are:

$$D^{ABB} = \Delta k \times D, \quad \ln(L^{ABB}) = \ln(L) + \Delta V \tag{14}$$

where $\Delta k = 1 - d_0 - d_1 v_b$ and $\Delta V = \ln(1 + p_0 + p_1 v_b + p_2 v_b^2)$.

For certain body bias v_b, the framework performs timing analysis by propagating the delay from gate to gate as in [3], [4] using (14). Meanwhile we can maintain the node delay in a canonical form with different coefficients. Leakage power analysis is achieved by summing lognormal random variables using Wilkinson's method as in [2]. The efficiency of statistical power analysis is further improved with an incremental approach (discussed in section III-C).

Since the principal component z_i is an independent standard Gaussian random variable (RV), the correlation between D^{ABB} and $\ln(L^{ABB})$ can be easily evaluated as:

$$\mathrm{Cov}(D^{ABB}, \ln(L^{ABB})) = \sum_{i=1}^{m+1} \Delta k a_i b_i \tag{15}$$

B. Yield Analysis and Optimization

Based on the biased gate models, we can perform statistical timing and power analysis and compute the correlation between delay and leakage power. Parametric yield of the circuit is defined as in [2]:

$$\mathrm{Y} = \Pr(D < D_{con}, \ln(P_L) < \ln(P_{con} - P_D)) \tag{16}$$

where D_{con} and P_{con} are constraints for delay and power, respectively, P_L is the leakage power and P_D is the dynamic power of the circuit. Both circuit delay and log of leakage are two Gaussian random variables. The underlying problem in (16) is then the integral of a bi-variate normal distribution over a rectangular region. The five parameters μ_D, σ_D, μ_L, σ_L and ρ (the mean and standard deviation of circuit delay and log of leakage power, and their correlation coefficient) are used to define the bi-variate normal distribution.

For simplicity, (16) can normalized as:

$$\begin{aligned} \mathrm{Y} &= \Pr(x < a, y < b) \\ &= \frac{1}{2\pi\sqrt{(1-\rho^2)}} \int_{-\infty}^{a}\int_{-\infty}^{b} e^{-\frac{x^2 - 2\rho x y + y^2}{2(1-\rho^2)}} dx dy \end{aligned} \tag{17}$$

where $x = \frac{D-\mu_D}{\sigma_D}$ and $y = \frac{\ln(P_L)-\mu_L}{\sigma_L}$ are normalized random variables, $a = \frac{D_0-u_D}{\sigma_D}$ and $b = \frac{\ln(P_0-P_d)-\mu_L}{\sigma_L}$ are the normalized constraints on delay and log of leakage power, and ρ is the correlation coefficient between the circuit delay and log of leakage. To evaluate this integral, [2] transformed the original rectangular region to a triangular region. The new region is then partitioned into several sub-domains and computed in sequence. However, this method may suffer from a high complexity of transformation and partitioning. To avoid these problems we propose the use of the Gaussian quadrature technique [21]–[23]. Gaussian quadrature is an efficient approach to compute integrals by a weighted sum of function values at specified abscissae within the domain of integration, and can reach analytical accuracy by a suitable choice of abscissae and weights. Reference [21] suggests a Gaussian quadrature model to compute the integral $\int_0^\infty \exp(-x^2) f(x) dx$:

$$\int_0^\infty \exp(-x^2) f(x) dx \approx \sum_{i=1}^{15} w_i f(x_i) \tag{18}$$

where x_i and w_i are abscissae and weights that are fixed for the integral of the form above without any dependence on $f(x)$.

With the substitutions of $u = \frac{(a-x)}{\sqrt{2(1-\rho^2)}}$, $v = \frac{(a-y)}{\sqrt{2(1-\rho^2)}}$, $a_1 = \frac{a}{\sqrt{2(1-\rho^2)}}$ and $b_1 = \frac{b}{\sqrt{2(1-\rho^2)}}$, (17) can be simplified to:

$$\mathrm{Y} = \frac{\sqrt{(1-\rho^2)}}{\pi} \int_0^\infty \int_0^\infty \exp(-u^2 - v^2) Y(u,v) du dv \tag{19}$$

where

$$Y(u,v) = e^{[a_1(2u-a_1) + b_1(2v-b_1) + 2\rho(u-a_1)(v-b_1)]} \tag{20}$$

By applying the model in (18) to u and v separately, we obtain:

$$\mathrm{Y} = f(a_1, b_1, \rho) = \sum_{i=1}^{15}\sum_{j=1}^{15} w_i w_j Y(x_i, x_j) \tag{21}$$

Since x_i and w_i in (21) are fixed for any arbitrary function $Y(u,v)$ [21], the computation time of (21) is independent of the problem size.

Based on the proposed yield analysis[2], our yield optimization problem can be formulated as an unconstrained optimization problem where the objective function is (16) and the design variables are the body voltage of each cluster, as shown below:

$$\mathbf{max} \quad \Pr(D < D_{con}, \ln(P_L) < \ln(P_{con} - P_D)) \tag{22}$$

This problem is then solved by the optimizer Lancelot [19]. Lancelot numerically evaluates the objective function and gradient of the yield. Thus, the optimization formulation in this section can use high-order models or even table-look-up models for computing the intrinsic gate delay and leakage to guarantee the accuracy in optimization.

C. Gradient Computation and Complexity

Lancelot [19] requires the computation of the gradient of the yield with respect to the body voltage of each cluster. This can be estimated by increasing or decreasing the body voltage of a cluster by a small amount and then computing the yield difference due to the body voltage change. To improve the efficiency of this step, we suggest a power perturbation scheme instead of a full-circuit statistical power analysis.

Assuming the body voltage for a cluster k is changed by a small amount Δv, the change in leakage power can then be written as:

[2]The model in (18) requires that when $\rho < 0$, which is the typical case for log of leakage and delay, the constraints should be $a \le 0$ and $b \le 0$. The other constraint cases, $\{a \ge 0, b \ge 0\}$, $\{a \le 0, b \ge 0\}$ and $\{a \ge 0, b \le 0\}$, can be easily transformed to $\{a \le 0, b \le 0\}$ by exploiting the underlying characteristics of bi-variate normal PDF [21]–[23].

$$\Delta P_L = \sum_{i\in k}\{P_{i,0}[1 + p_{i,0} + p_{1,i}(v_{b,k} + \Delta v) + p_{2,i}\times (v_{b,k} + \Delta v)^2] - P_{i,0}(1 + p_{i,0} + p_{1,i}v_{b,k} + p_{2,i}v_{b,k}^2)\} \quad (23)$$

where $P_{i,0}$ is the leakage with zero-body bias for gate i, and $p_{1,i}$ and $p_{2,i}$ are the coefficients for the leakage model in (1). This can be further simplified to:

$$\Delta P_L = \sum_{i\in k} P_{i,0}(p_{1,i}\Delta v + p_{2,i}\Delta v^2) + v_{b,k}\sum_{i\in k} 2P_{i,0}p_{2,i}\Delta v \quad (24)$$

Since the body voltage increment Δv can be fixed in gradient analysis, $v_{b,k}$ is the only variable in (24), which indicates that the coefficients $\sum_{i\in k} P_{i,0}(p_{1,i}\Delta v + p_{2,i}\Delta v^2)$ and $\sum_{i\in k} 2P_{i,0}p_{2,i}\Delta v$ can be computed in advance and used throughout the whole optimization process. We just need to perform N summations to compute the change in the leakage PDF for N clusters in gradient computation. The complexity is reduced from $O(NN_g)$ for N full statistical power analysis to $O(N)$, where N_g is the number of gates.

Timing perturbation is performed by a full statistical static timing analysis (SSTA). Once we obtain the delay and leakage-power PDFs of the perturbed circuit (the body voltage of the k_{th} cluster is changed from $v_{b,k}$ to $v_{b,k} + \Delta v$), the yield of the perturbed circuit can be calculated by (21), and the change in yield is used to define the particular component of the yield gradient. Since yield analysis has a constant complexity, the overall algorithm complexity of this optimization framework is dominated by SSTA, the complexity of which is $O(N(N_g+E))$, where E is the number of edges of the timing graph. The number of the clusters is limited in real designs and is negligible compared to N_g and thus the framework maintains a linear complexity[3]. Experimental results in section IV also validate that yield optimization takes only seconds even for a circuit with tens of thousands of gates.

IV. Experimental Results

The proposed framework discussed in sections II and III were implemented in C and tested on ISCAS85 benchmark circuits and a Viterbi Decoder circuit (Vit1) that vary in size from 166 to 14539 gates. The circuits were synthesized using an industrial 1.2V 90nm triple-well dual-V_{th} technology. The two V_{th} values are 0.32V (0.33V) and 0.22V (0.24V) for NMOS (PMOS). Body voltage is varied between -0.5 and 0.5V. All standard cells in the library were characterized (using SPICE) at both the high- and low-V_{th} values. Only channel length variation is considered for simplicity. However, the overall approach can be extended to consider other sources of variability. We considered inter-die, spatially correlated intra-die, and random components of variation. Total $3\sigma/\mu$ for channel length variability is set to 15% and then split evenly among the three variation components.

A. Efficacy of Feature Extraction-Based Clustering

Reference [12] proposed a clustering algorithm based on an empirical affine weighting function. Table I compares the proposed feature extraction-based clustering algorithm (Feat.) with the work of [12] (Empir.) in terms of both resulting leakage and runtime. Column 2 lists the number of gates for each circuit, varying from 166 to 14539 gates. Columns 3-6 compare the mean/standard deviation and 95$^{\text{th}}$ percentile leakage of the proposed method and the method from [12], respectively. The proposed approach improves upon the prior work in all measures and achieves a 10% and 17% reduction in the mean and standard deviation of the leakage, respectively. The last two columns of Table I compare the gate clustering runtime for the two methods. The runtime for the proposed method shows linear dependence on circuit size with a small slope, whereas the runtime for the empirical function-based method [12] increases exponentially. On average, the proposed method is 5.1× faster than the method in [12]. For the largest circuit Vit1, the proposed method achieves 18× speed-up.

[3]The number of Lancelot iterations (around 30) is limited due to the small problem size.

TABLE I
Clustering efficiency comparison between the proposed feature extraction-based clustering method and the empirical affine weighting function-based clustering method from [12]

CKT.	#gates	Leakage comparison (μW)				Time (sec.) for clustering	
		Empir. [12]		Feat.			
		u/σ	95%	u/σ	95%	Empir.	Feat.
c432	166	3.8/1.3	6.2	3.5/1.2	5.6	0.8	5.9
c499	519	18.2/7.1	31.7	17.8/6.3	28.3	5.3	6.5
c880	390	4.2/1.6	7.0	3.9/1.5	6.4	4.2	6.3
c1355	558	15.7/5.2	25.0	14.8/4.5	22.1	8.6	6.7
c1908	432	8.9/3.1	14.2	7.8/2.3	12.1	7.5	6.5
c2670	964	8.5/3.3	14.7	7.5/2.8	12.7	26.0	7.4
c3540	962	14.9/6.1	26.7	14.2/5.7	24.5	25.7	7.4
c5315	1750	19.7/7.6	35.6	17.7/7.1	31.4	84.3	9.2
c6288	2502	89/35	155	82/30	134	179	11
c7552	2102	23/10	42	20/8	35	122	10
Vit1	14539	246/110	396	210/80	348	901	52
Average improvement (%)				10/17	13.4		

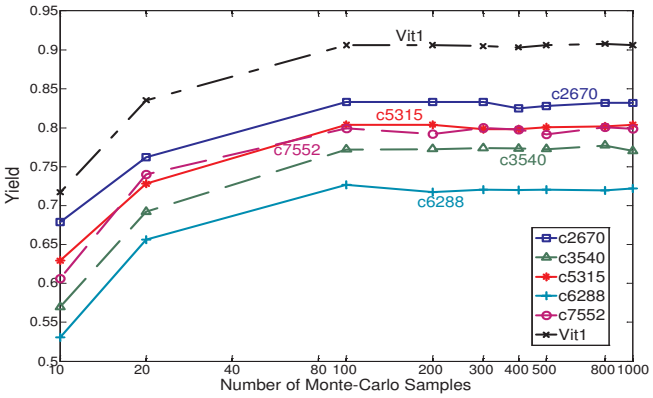


Fig. 9. Monte Carlo convergence

B. Monte Carlo Convergence

The design space exploration step described in section II-B is executed only once in the framework but still involves solving a large number of QP problems to determine the body bias profile across process variability. To speed up this step, we employ the smart sampling approach in [24], which captures the importance of the samples to reduce the number of samples. Figure 9 shows the dependence of yield optimization results on the number of Monte Carlo samples for the six largest circuits in our set of benchmarks. The quality of the yield optimization results with 100 samples is similar to the results with 1000 samples. We therefore use 100 samples in the exploration step, as design space exploration is only required to outline general features instead of local details. Moreover, since the QP optimization for a

TABLE II
COMPARISON OF YIELD OPTIMIZATION RESULTS USING THE PROPOSED PRE-SILICON BODY BIAS SELECTION FRAMEWORK AND POST-SILICON ABB WITH DIFFERENT NUMBER OF CLUSTERS (TWO AND THREE CLUSTERS)

CKT.	Initial yield (%)	Optimized yield (%)/Yield point improvement							
		pre-silicon BB selection				post-silicon ABB			
		2 cluster		3 cluster		2 cluster		3 cluster	
c2670	38.7	73.3	35	83.1	44	83.8	45	90.7	52
c3540	38.7	65.7	27	77.7	39	75.8	37	84.1	45
c5315	39.2	69.4	30	80.7	42	78.6	39	88.2	39
c6288	38.4	63.4	25	72.1	34	71.1	33	77.8	39
c7552	38.8	69.4	31	80.1	41	76.7	38	86.9	48
Vit1	39.1	78.6	40	90.6	52	83.3	44	91.5	52
Ave. improv.		31		42		40		48	

TABLE III
COMPARISON OF YIELD OPTIMIZATION RESULTS USING THE PROPOSED PRE-SILICON BODY BIAS SELECTION FRAMEWORK AND TRADITIONAL PRE-SILICON APPROACHES (DUAL-V_{th} [8] AND GATE SIZING [2])

CKT.	Initial yield (%)	Optimized yield (%)/Yield point improvement							
		pre-silicon BB selection				dual-V_{th} [8]		sizing [2]	
		2 cluster		3 cluster					
c2670	38.7	73.3	35	83.1	44	39.5	0.8	46.3	7.6
c3540	38.7	65.7	27	77.7	39	41.4	2.7	44.1	5.4
c5315	39.2	69.4	30	80.7	42	40.3	1.1	45.0	5.8
c6288	38.4	63.4	25	72.1	34	38.8	0.4	43.2	4.8
c7552	38.8	69.4	31	80.1	41	40.2	1.4	47.9	9.1
Vit1	39.1	78.6	40	90.6	52	50.7	12	52.8	14
Ave. improv.		31		42		3.0		7.7	

given circuit sample is independent of the QP optimizations for other samples of the same circuit, this step can be easily parallelized to achieve further speedup.

C. Pre-Silicon Body Bias Selection Framework vs. Post-Silicon ABB

The proposed pre-silicon body bias selection framework chooses statistically optimal body voltages for the full ensemble of chips, while post-silicon ABB uses measurement results for a particular manufactured chip and deterministically selects the bias voltage for each cluster of that chip. It is clear that post-silicon ABB should provide higher yields at the cost of higher testing times and costs. This section quantifies the yield loss when using the proposed pre-silicon approach compared to a post-silicon ABB with the same clustering method described in section II.

Given a clustering, the yield of post-silicon ABB is computed by first generating 1000 chip samples which are then individually tuned to minimize leakage subject to a delay constraint. The number of chips that fail to simultaneously meet the leakage and delay targets is then calculated. The yield optimization results of our pre-silicon approach and post-silicon ABB are summarized in Table II. Column 2 lists the initial pre-optimized yield of each circuit for the target constraint $\{\text{Delay}<\mu_D+\sigma_D,\ \text{Leakage}<\mu_L\}$. Columns 3-10 display the yield optimization results and yield point improvement using our pre-silicon framework and post-silicon ABB for two- and three-cluster scenarios. Although post-silicon ABB achieves slightly higher yield than the proposed pre-silicon body bias selection framework, the difference degrades for larger number of clusters and larger circuits.

D. Pre-Silicon Body Bias Selection Framework vs. Traditional Pre-Silicon Approaches

We further evaluate the efficacy of our pre-silicon framework in Table III when compared to traditional pre-silicon methods (a statistical dual-V_{th} assignment approach [8] and a yield maximization approach using gate sizing [2]). Columns 3-10 list the yield optimization results and yield point improvement for the constraint $\{\text{Delay}<\mu_D+\sigma_D,\ \text{Leakage}<\mu_L\}$ using our pre-silicon framework and two traditional pre-silicon statistical optimization methods (dual-V_{th} and gate sizing [2], [8]). The proposed approach with either two or three clusters potentially doubles the original yield of 39% (the optimized yield is 70% for two clusters and 81% for three clusters on average). Meanwhile the yield improvement is limited to 3.0 point on average for the statistical dual-V_{th} approach [8] and 7.7 point on average for gate sizing [2]. This further validates the statement in section I that the proposed pre-silicon body bias selection has continuous domain design variables and hence higher flexibility than the traditional pre-silicon approaches like gate sizing or dual-V_{th}.

E. Yield Analysis and Optimization

This section discusses the accuracy and optimality of the proposed pre-silicon framework, as shown in Table IV when compared to Monte Carlo simulation. Columns 2-5 list the mean and standard deviation of the delay and leakage for the initial designs. The target constraint is set to $\{\text{Delay}<\mu_D+\sigma_D,\ \text{Leakage}<\mu_L\}$. Given this constraint we compute the original yield and compare it with a Monte Carlo model using 2000 samples, shown in columns 6-7. The absolute errors of the proposed yield analysis in section III-B vary from 0.9% to 5.7%, which is due to the computation approximation in SSTA, $e.g.$, statistical maximization operation.

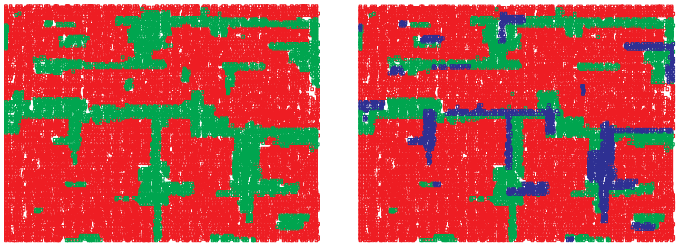
The optimized yield results and the yield point improvements are shown in Columns 8-11 for 11 designs. The optimized yield almost doubles the original yield with 69% for two clusters and 80% for three clusters on average. The improvements are consistent among all the benchmarks studied. We also perform a Monte Carlo sweep (MC-sweep) to determine whether the optimized yield obtained by the proposed framework is globally optimal. MC-sweep performs Monte Carlo simulations on all possible combinations of body voltages for a three-cluster configuration. The sweep increment is set to 0.1V for the sweep space $v_{b,1} \times v_{b,2} \times v_{b,3}$. Columns 12-13 of Table IV show the maximum yield found by MC-sweep and the relative deviation of the proposed approach with respect to MC-sweep. The maximum deviation is 6.7%, which is due to the relatively coarse grid used for sweep. Columns 14-17 summarize the runtime for the critical stages of the proposed framework (including design space exploration (explo.), clustering (clust.) and yield optimization (optim.)) as well as the total runtime. The last column lists the ratio of the total runtime to the circuit size, indicating a linear relationship. Runtime is dominated for larger circuits by the exploration stage, which can be parallelized across machines as mentioned above.

F. Implications for Physical Design

Adaptive body bias incurs physical design overheads, including generation/distribution of the body voltages and extra well spacing. There are limited numbers of clusters (2-3 in our experiments) and as a result, these overheads can be reasonably bounded. The major impact of gate clustering on placement is then the extra well spacing between adjacent cells having different biases, which is imposed by triple-well-layout rules. As stated in section II, the proposed clustering method naturally captures the spatial correlation in the feature

TABLE IV
YIELD ANALYSIS/OPTIMIZATION RESULTS AND SUMMARY OF RUNTIME (FOR 3-CLUSTER SCENARIO) UNDER THE CONSTRAINT {DELAY<$\mu_D+\sigma_D$, LEAKAGE<μ_L} USING THE PROPOSED METHOD WITH DIFFERENT NUMBER OF CLUSTERS (2 CLUSTERS AND 3 CLUSTERS) AND MONTE CARLO APPROACHES

CKT.	Initial design				Init. yield (%)		Optim. yield(%)/Impro.				MC-sweep(%)		Time for critical stages(sec.)			Total (sec.)	Ratio
	μ_D	σ_D	μ_L	σ_L	Prop.	MC	2 cluster		3 cluster		yield	err.	explo.	clust.	optim.		
c432	0.74	0.05	0.97	0.26	38.4	43.6	70.5	32	79.2	41	82.5	3.9	0.68	5.89	2.35	36.1	0.22
c499	0.68	0.04	3.80	1.02	39.2	41.9	65.8	27	74.7	36	76.2	1.9	5.87	6.48	2.40	47.3	0.09
c880	0.77	0.05	1.17	0.32	38.6	42.6	68.7	30	79.1	41	78.8	0.3	3.29	6.32	2.48	41.1	0.11
c1355	0.89	0.05	3.57	0.96	39.3	43.8	68.4	29	82.7	43	79.4	4.1	6.43	6.64	2.68	46.7	0.08
c1908	1.15	0.07	2.16	0.58	39.0	42.1	68.1	29	80.6	42	79.2	1.8	4.13	6.46	3.28	48.8	0.11
c2670	0.77	0.05	2.06	0.56	38.7	43.4	73.3	35	83.1	44	84.3	1.4	18.4	7.38	3.13	64.8	0.07
c3540	1.22	0.07	3.15	0.84	38.7	44.5	65.7	27	77.7	39	83.3	6.7	23.5	7.36	3.37	73.5	0.08
c5315	1.12	0.07	4.14	1.11	39.2	42.2	69.4	30	80.7	42	79.4	1.7	68.2	9.21	4.03	134	0.08
c6288	3.52	0.21	14.77	3.91	38.4	41.1	63.4	25	72.1	34	76.4	5.6	185	10.5	5.0	258	0.10
c7552	1.28	0.08	4.14	1.10	38.8	39.7	69.4	31	80.1	41	75.8	5.6	110	9.8	4.3	173	0.08
Vit1	2.41	0.14	149.44	39.54	39.1	41.8	78.6	40	90.6	52	91.6	1.0	747	51.9	12.2	1177	0.08
Average yield point improvement							30		41								



(a) (b)

Fig. 10. Vit1 placement with physically contiguous cluster regions by CAPO [25] (different clusters are shown with different colors): (a) two clusters; (b) three clusters

vectors. That is, most gates are intrinsically clustered within the physically continuous regions, which helps reduce the well spacing overheads. Moreover, we employ the incremental placer CAPO [25] to minimize the gate displacement and area overhead, following a similar flow as in [12]. CAPO works in its Engineering Change Order (ECO) mode to make limited changes to the initial placement based on certain constraints [25]. Figure 10 demonstrates the resulting layout for the Vit1 circuit with both two and three clusters after applying CAPO to the initial placement. Most gates in the layout are clearly clustered in the physically continuous regions. In particular, the average gate displacement is 2.2%-2.3% and half perimeter wire-length increase is 2.7%-3.9% compared with the initial designs, for two- and three-cluster configurations.

V. CONCLUSION

In this paper we presented a gate-level parametric yield optimization framework using *design time* body bias selection. The approach considers the power and performance constraints as well as their correlation. A feature extraction-based clustering approach is proposed that achieves speedups of 5.1× on average and up to 18× for 11 benchmark circuits compared to a recently reported clustering strategy, with leakage savings of more than 10%. In addition the framework employs a fast yield analysis calculation method and an efficient power perturbation technique for optimization and achieves 41% yield improvement on average across 11 benchmark circuits.

VI. ACKNOWLEDGEMENT

The authors gratefully acknowledge the Semiconductor Research Corporation (SRC), Semiconductor Technology Academic Research Center of Japan (STARC) and National Science Foundation (NSF) for supporting this work.

REFERENCES

[1] D. Frank, *et.al*. Design and CAD challenges in 45nm CMOS and beyond. In *Proc. ICCAD*, pages 329–333, 2006.
[2] A. Srivastava, *et.al*. A novel approach to perform gate-level yield analysis and optimization considering correlated variations in power and performance. *IEEE TCAD*, vol. 27, no. 1:272–285, 2008.
[3] H. Chang and S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single pert-like traversal. In *Proc.ICCAD*, pages 621–625, 2003.
[4] C. Visweswariah, *et.al*. First order incremental block based statistical timing analysis. In *Proc.DAC*, pages 331–336, 2004.
[5] X. Bai, *et.al*. Uncertainty aware circuit optimization. In *Proc. DAC*, pages 58–63, 2002.
[6] R. Gandikota, *et.al*. Victim alignment in crosstalk aware timing analysis. In *Proc. ICCAD*, pages 698–704, 2007.
[7] R. Gandikota, *et.al*. Modeling crosstalk in statistical static timing analysis. In *Proc. DAC*, pages 947–979, 2008.
[8] A. Srivastava, *et.al*. Statistical optimization of leakage power considering process variations using dual-vth and sizing. In *Proc. DAC*, pages 773–778, 2004.
[9] D. Sinha, *et.al*. Statistical gate sizing for timing yield optimization. In *Proc. ICCAD*, pages 1037–1041, 2005.
[10] S. Bhardwaj and S. Vrudhula. Leakage minimization of nano-scale circuits in the presence of systematic and random variations. In *Proc. DAC*, pages 541–546, 2005.
[11] M. Mani, *et.al*. An efficient algorithm for statistical minimization of total power under timing yield constraints. In *Proc. DAC*, pages 309–314, 2005.
[12] S. Kulkarni, *et.al*. A statistical framework for post-silicon tuning through body bias clustering. In *Proc. ICCAD*, pages 39–46, 2006.
[13] M. Mani, *et.al*. Joint design-time and postsilicon minimization of parametric yield loss using adjustable robust optimization. In *Proc. ICCAD*, pages 19–26, 2006.
[14] V. Khandelwal and A. Srivastava. Variability-driven formulation for simultaneous gate sizing and post-silicon tunability allocation. In *Proc. ISPD*, pages 17–25, 2006.
[15] J. Tschanz, *et.al*. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. *IEEE JSSC*, vol. 37, issue 11:1396–1401, Nov. 2002.
[16] K. Leung and P. Mok, A capacitor-free CMOS low-dropout regulator with damping-factor-control frequency compensation. *IEEE JSSC*, vol. 38, no. 10:1691–1702, Oct 2003.
[17] H. Tanaka, *et.al*. A precise on-chip voltage generator for a gigascale DRAM with anegative word-line scheme. *IEEE JSSC*, vol. 34, issue 8:1084–1090, 1999.
[18] I. Guyon, *et.al*., *Feature Extraction, Foundations and Applications*. Springer, 2006.
[19] Lancelot, http://www.numerical.rl.ac.uk/lancelot/blurb.html.
[20] CPLEX, http://www.ilog.com/products/cplex/.
[21] N. Steen, *et.al*. Gaussian quadratures for the integrals. *Math. of Comp.*, vol. 23, no.107:661–671, 1969.
[22] Z. Drezner, *Computation of the multivariate normal integral*. *ACM TOMS*, vol. 18, issue 4:470–480, 1992.
[23] S. Kotz, *et.al*. *Continuous Multivariate Distributions*. Wiley, 2000.
[24] V. Veetil, *et.al*. Efficient monte carlo based incremental statistical timing analysis. In *Proc. DAC*, pages 676–681, 2008.
[25] J. Roy and I. Markov, ECO-system: embracing the change in placement. University of Michigan, Ann Arbor, MI, Tech. Rep. CSETR-519-06, 2006.