Design-Time Optimization of Post-Silicon Tuned Circuits Using Adaptive Body Bias

Sarvesh H. Kulkarni, Dennis M. Sylvester, *Senior Member, IEEE*, and David T. Blaauw

***Abstract*—Adaptive body biasing is a powerful technique that allows post-silicon tuning of individual manufactured dies such that each die optimally meets the delay and power constraints. Assigning individual bias control to each gate leads to severe overhead, rendering the method impractical. However, assigning a single bias control to all gates in the circuit prevents the method from compensating for intra-die variation and greatly reduces its effectiveness. In this paper, we propose a new variability-aware method that clusters gates at design time into a handful of carefully chosen independent body-bias groups, which are then individually tuned post-silicon for each die. We show that this allows us to obtain near-optimal performance and power characteristics with minimal overhead. For each gate, we generate the probability distribution of its post-silicon ideal body bias voltage using an efficient sampling method. We then use these distributions and their correlations to drive a statistically aware clustering technique. We study the physical design constraints and show how the area and wirelength overhead can be significantly limited using the proposed method. Compared with a fixed design-time based dual threshold voltage assignment method, we improve leakage power by 38%–68% while simultaneously reducing the standard deviation of delay by two to nine times.**

***Index Terms*—Adaptive body bias (ABB), post-silicon tuning, variability, very large scale integration.**

I. Introduction

MODERN CMOS circuits suffer from high parametric yield loss due to the strong dependence of leakage and delay on process parameters such as channel length and threshold voltage ($V_{\rm th}$) [1]. A number of approaches have been proposed to mitigate this using pre-silicon statistical optimization. These approaches optimize the selection of design-time variables (such as gate sizes and $V_{\rm th}$s) to maximize yield [2], [3]. By using statistical models of the underlying silicon variation, these techniques aim to maximize the number of chips that will meet power and delay constraints post-silicon. However, since the obtained optimization decisions apply to the entire set of manufactured die, it is inevitable that for some dies with badly skewed process parameters, delay or power constraints will not be met post-silicon.

On the other hand, post-silicon tuning techniques have been introduced [4], [5] that allow the adjustment of device characteristics after a die has been manufactured to compensate for the specific deviations that occurred on that particular die. Because post-silicon tuning allows each die to be adjusted independently, even dies with strongly skewed process conditions can be adjusted to meet the power and delay specifications. Hence, post-silicon adaptive techniques provide the opportunity for almost all manufactured chips to exactly meet their constraint, and it is well accepted that post-silicon adaptive techniques significantly outperform conventional pre-silicon statistical optimization.

This unique opportunity necessitates a fundamental shift in design-time optimization formulations. The conventional pre-silicon statistical optimization is akin to predicting the most likely process conditions and centering the design parameters to give a maximum yield within its vicinity. In contrast, post-silicon tunable methodologies leave the compensation for process variation to the post-silicon phase and aim to provide the maximum tuning flexibility while limiting overhead incurred by the added hardware. To effectively make the tradeoff between fine-grain control and low tuning overhead, the design optimization process should group or cluster gates based on predicted post-silicon tuning values of the individual gates. For an effective clustering, the tuned values of each gate must be computed and compared across a large set of possible die. While this process is statistical in nature, it is clear that this task is fundamentally different from the traditional statistical design optimization problems that have been formulated. In this paper, we therefore propose an entirely different optimization methodology to address this problem. We focus on adaptive body biasing (ABB) [4], [5] as the method for post-silicon tuning, but note that the methodology may be useful to other post-silicon tuning approaches as well.

ABB allows the tuning of the $V_{\rm th}$s of gates by controlling the transistor body-source voltage ($V_{\rm bs}$). A forward body bias ($V_{\rm bs} > 0$) reduces $V_{\rm th}$, thus increasing speed at the cost of increased leakage power. Alternatively, a reverse body bias (RBB, $V_{\rm bs} < 0$) reduces leakage while slowing the device. Thus, the impact of process variations can be mitigated by speeding up slow and less leaky devices while slowing down devices that are fast and highly leaky, or non-critical.

Many issues arise while implementing an ABB scheme in practice. As already noted, it is desirable to bias each gate in a design independently. However, supplying this many separate voltages inside a die is not viable due to well spacing related

Manuscript received September 10, 2006; revised April 22, 2007 and August 15, 2007. This work was supported in part by the National Science Foundation (NSF), by the Semiconductor Research Corporation (SRC), and by the Gigascale Systems Research Center (GSRC). This paper was recommended by Associate Editor S. Vrudhula.

S. H. Kulkarni was with the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109 USA. He is now with the Advanced Design Group, Portland Technology Development, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: sarvesh.h.kulkarni@intel.com).

D. M. Sylvester and D. T. Blaauw are with the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: dennis@eecs.umich.edu; blaauw@eecs.umich.edu).

Digital Object Identifier 10.1109/TCAD.2008.915529

layout rules as well as the high routing and bias generation overhead. On the other hand, using the same body bias for all devices limits the ability to compensate for intra-die variations and results in suboptimal power results. It is therefore necessary to cluster the gates in a design such that gates within a cluster share the same body bias. As we will show later, it is vital that the correct gates are clustered together. Clustering hence becomes a difficult problem and must be considered at design time while accounting for the expected levels of process variation.

While ABB as a tuning technique is well established [4], [5], relatively few research works have been performed in the area of design-time optimization for ABB. In [6], a framework for assigning tuning voltages is cast as an integer linear program. However, the body voltages are fixed at design time using a deterministic formulation, and post-silicon tuning is not considered. In [7], results for two small ABB-enabled designs are presented. This paper relies on a multiple objective evolutionary algorithm to determine ABB voltages for individual device wells post-silicon. However, a general scalable clustering approach to reduce the number of ABB control voltages, and hence reduce the overhead to practical levels, is not available in the literature.

In this paper, we present a novel three-phase approach to gate-level body-bias clustering considering variability. The proposed method accommodates all components of variations, including inter-die, spatially correlated intra-die, and random variations. In the first phase, we compute for each gate the probability distribution functions (PDFs) of its optimal post-silicon tuned body bias voltage. The underlying optimization problem in this phase relies on a quadratic-program (QP) formulation that can be solved very efficiently and is therefore embedded inside a Monte Carlo simulation. The second phase then performs a statistically aware clustering of the gates using these probability distributions and their correlation information. Gates can be partitioned into any given number of clusters, allowing us to explore the power/performance impact of the number of clusters in a design. Finally, in the third phase, we perform the post-silicon tuning of the ABB clusters by taking dies from a sample set and finding the best tuning configuration for each die such that it meets the power and delay constraints. We also study the related physical design issues. By limiting the number of clusters to just a few, the overhead is already drastically reduced compared with approaches that use individual gate-level ABB control. We show that modern placers [8]–[10] can be used to incrementally perturb an initial placement, yielding only small area and wirelength overheads given the proposed ABB clusters, and that the gains far outweigh these small penalties.

We compare our approach with fixed dual $V_{\rm th}$ assignment [11] on a set of benchmark circuits. We show that with only two to three ABB clusters, the proposed approach yields significant improvements over the dual $V_{\rm th}$ design. For instance, Fig. 1 shows a scatter plot of leakage and delay for the c432 circuit for a traditional dual $V_{\rm th}$ design and for a design tuned using our proposed work with three ABB clusters. The delay spread, as well as the mean power, is significantly reduced, resulting in higher parametric yield and improved performance.

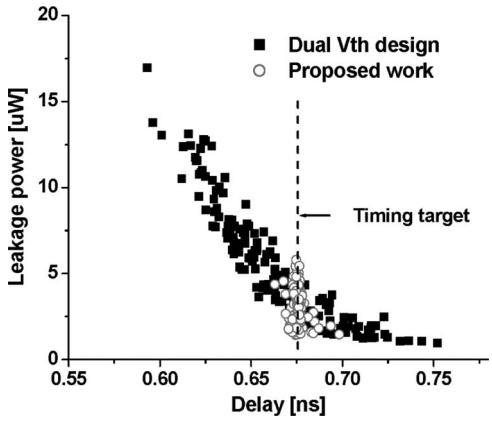


Fig. 1. Power/delay scatter plot for dual $V_{\rm th}$ and ABB designs.

In summary, the key contributions of this paper are the following.

1) This paper presents the first gate-level optimization method for circuits enabled with ABB while taking process variations into account. A physical design methodology is also presented which delivers tight control on placer overheads while using the proposed cluster formation approach.
2) We present a new gate-level optimization framework for post-silicon tunable circuits. Although results in this paper focus on ABB as the post-silicon tuning mechanism, we feel that the underlying philosophy could be useful for other post-silicon optimization methods such as adaptive supply voltage [12].
3) We show that it is important to actively consider the post-silicon tunability during the pre-silicon design cycle in order to truly leverage the post-silicon adaptivity.

This paper is organized as follows. Section II provides background and describes our power/delay models and simulation setup. Section III describes our QP formulation for body-bias assignment. In Section IV, we present the new variation-aware body-bias clustering methodology for optimized post-silicon tuning. Section V presents results, including an analysis of physical design implications. Finally, Section VI summarizes the findings of this paper.

II. Background

A. Power and Delay Models

Body biasing relies on the body-effect phenomenon to modulate the $V_{\rm th}$ of a MOSFET. Equation (1) gives the dependence of $V_{\rm th}$ of an NMOS transistor on the $V_{\rm bs}$. $V_{\rm th0}$ is the nominal $V_{\rm th}$ at zero body bias, γ is the body coefficient, and $\phi_{\rm F}$ is the Fermi potential.

$$V_{\rm th} = V_{\rm th0} + \gamma\left(\sqrt{2\phi_{\rm F} + V_{\rm bs}} - \sqrt{2\phi_{\rm F}}\right). \tag{1}$$

Forward biasing (FBB) the body with respect to the source reduces $V_{\rm th}$, increasing the speed. However, because of the exponential dependence of subthreshold leakage on $V_{\rm th}$, it also leads to a large increase in leakage power. Similarly, RBB reduces leakage at the cost of increased delay. This power–delay

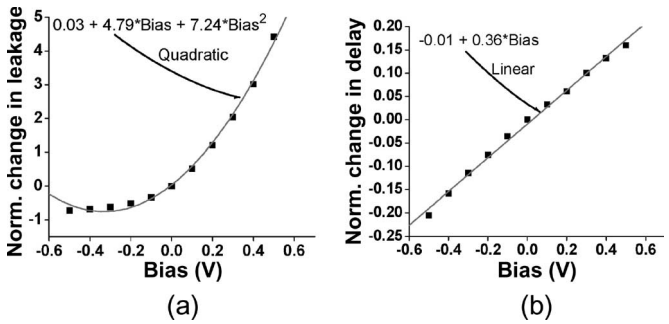


Fig. 2. Leakage power and delay modeling.

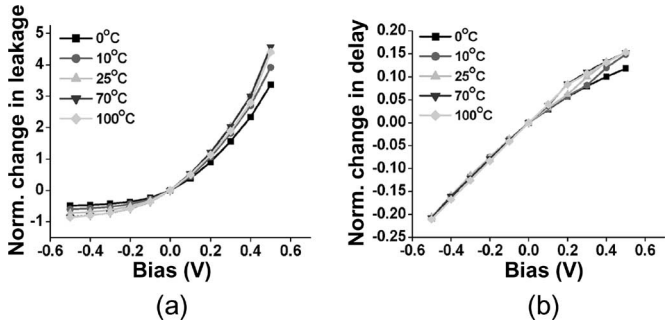


Fig. 3. Effect of temperature on the body effect.

tradeoff enabled by the body bias can be exploited by forward biasing gates on critical paths while reverse biasing gates on non-critical paths. Thus, the process needs to only provide high $V_{\rm th}$ gates which can be tuned using FBB and RBB. In comparison, in traditional dual $V_{\rm th}$ schemes, the process needs to provide two different $V_{\rm th}$s. The lower $V_{\rm th}$ provides higher speed at the cost of power and is used on critical paths to meet timing in such dual $V_{\rm th}$ schemes.

This paper is based upon an industrial 1.2-V 90-nm triple-well dual $V_{\rm th}$ process. The two $V_{\rm th}$ values that are available are 0.32 V (-0.33 V) and 0.22 V (-0.24 V) for NMOS (PMOS). The body bias is varied between ± 0.5 V in our analysis for measuring delay and power changes.

The power models in this paper include all leakage mechanisms affected by the $V_{\rm th}$ and the body bias, namely, subthreshold leakage, body-source/drain junction diode leakage [13], and band-to-band tunneling [14]. These leakage mechanisms have become serious concerns as they exhibit large variations and are also the dominant static power components at typical elevated operating temperatures. Other components of total power, such as gate leakage or dynamic power, are thus not included in our reported results and remain fairly constant under our optimizations.

The exact analytical equations governing the dependence of leakage (and delay) on the body bias will be very complex. This becomes evident when we couple (1) with (2) which shows the dependence of subthreshold leakage and delay on the $V_{\rm th}$ [15], [16]

$$I_{\rm subthreshold} = K_1 e^{\frac{(V_{\rm GS}-V_{\rm th})}{n\nu_{\rm T}}}, \quad Delay = \frac{K_2}{(V_{\rm DD}-V_{\rm th})^{\alpha}} \qquad (2)$$

where α is the velocity saturation index, $\nu_{\rm T}$ is the thermal voltage, and n is the subthreshold swing coefficient.

In order to make the problem mathematically tractable, we approximate this complex dependence using the power and delay models shown in Fig. 2. Fig. 2(a) and (b) shows the change in leakage power (averaged across input states) and delay as the body bias is varied between ± 0.5 V (normalized to the zero body-bias power and delay). From these figures, we see that the change in leakage and delay can be modeled with a reasonable accuracy using quadratic and linear functions of the body voltage. The average errors in leakage and delay are 5.9% and 1.5%, respectively. Note that these approximations are used only to set up a formulation which can be optimized with efficient runtimes (described in Section III). After optimization, the solution is reevaluated using a cubic polynomial for leakage which further reduces the average error to 0.8%.

From Fig. 2, a $+0.5$-V forward bias can provide a speedup of about 16% with a leakage increase of $4.4\times$, whereas a -0.3-V reverse bias reduces the leakage by 38% while slowing the gate down by 11%.

Fig. 3 shows the power and delay models as temperature is varied. From these figures, we find that the quadratic power and linear delay modeling method described in the previous paragraph continues to hold over a wide range of operating temperatures. This also suggests a weak dependence of the body effect on the temperature in the technology used.

B. Simulation Setup

The standard cell library for the target 90-nm process has two- and three-input NOR and NAND gates and inverters, and the process provides a triple-well option which allows for the ABB. Cells are characterized using SPICE to quantify their delay and leakage across different $V_{\rm th}$ and body-bias values. Cells are also characterized for their delay and leakage as the channel length varies. In this paper, we consider the channel length as the source of variability. The implicit dependence of $V_{\rm th}$ variation induced by channel-length variation is automatically captured in SPICE. We have not considered other sources of variation such as gate-oxide thickness or doping concentration.

Different correlation grids, correlation functions, and breakdowns between inter/intra/random components were studied in order to validate our findings against our assumed models of variability.

Spatial correlations between gates are modeled by storing them in a 3×3 grid-based correlation matrix (Grid 1 in Fig. 4). Results for a second 7×7 grid (Grid 2 in Fig. 4) are also presented in this paper (Section V-A-5). Layouts for all benchmarks were generated using Cadence Silicon Ensemble. The assumed correlation functions are shown in Fig. 4. Grid 1 assumes a linear falloff in the correlation coefficients (ρ), whereas Grid 2 is based on a correlation function from industry. ($\sigma_{\rm inter}$, $\sigma_{\rm intra}$, and $\sigma_{\rm radom}$) are (3%, 4%, and 1%) and (3%, 2%, and 1%) of μ for Grids 1 and 2, respectively.

Test circuits taken from the ISCAS85 benchmark set [17] are first sized up using a TImed LOgic Synthesizer (TILOS)-based gate-sizing tool [18] using only high $V_{\rm th}$ gates. ABB clustering or low $V_{\rm th}$ assignment (for comparing) is then used to speed up the design. We consider speedups of 5% and 10% beyond the initially sized design in this paper (Fig. 5). Our implementation

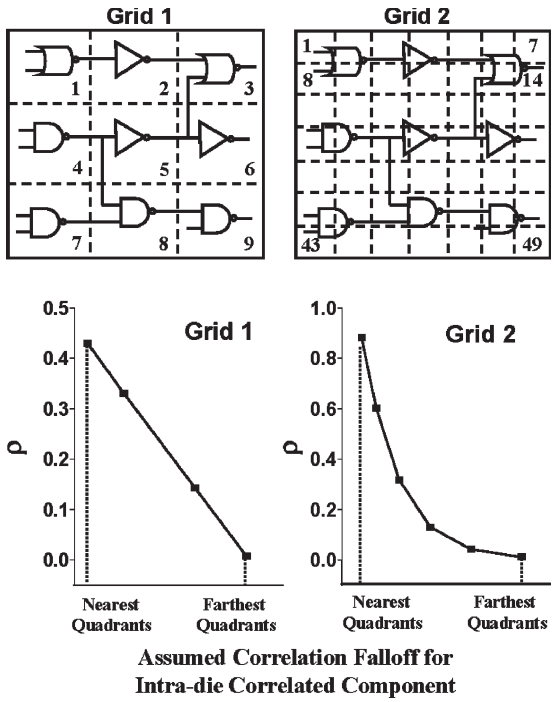


Fig. 4. Models for spatial correlations.

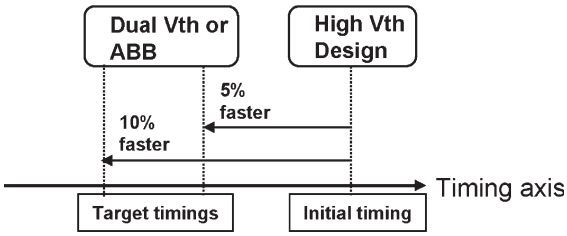


Fig. 5. General experimental setup.

of the dual $V_{\rm th}$ assignment is based on a sensitivity-driven method presented in [11], which inserts low $V_{\rm th}$ gates on only those timing arcs that most improve timing.

We also present results for two larger circuits ("Viterbi" and "SOVA2") with about 15 000 and 32 000 gates to demonstrate the effectiveness of this paper on larger designs. We do not consider gate sizing in this paper as it is an orthogonal knob that can be added to the body biased as well as the dual $V_{\rm th}$ design.

III. QP-Based Deterministic Body-Bias Assignment

This section describes our formulation of the optimization problem for body-bias voltage tuning in a deterministic scenario. We then use this optimization as the basis of the Monte

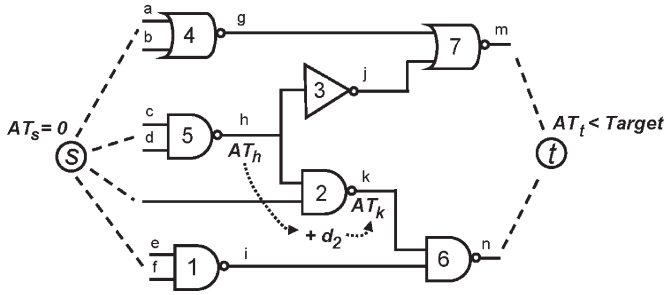


Fig. 6. Setting up the QP for body-bias assignment.

Carlo simulations to obtain the distribution of body bias voltage across process variations. Consider the c17 circuit shown in Fig. 6. Wires are represented as letters (a–m) and gates as numbers (1–7). AT represents the arrival time of the signal on a wire. All primary inputs (PIs) and primary outputs (POs) are tied to supernodes "s" and "t," respectively.

We now develop the constraints of the optimization problem. All gates initially have some delay values as obtained in the gate-sizing step using only high $V_{\rm th}$ gates (described in Section II-B). These delay values will now be optimally reduced using a QP such that the circuit meets the timing target. The constraints can be written as (3), shown at the bottom of the page.

The first and second constraints in (3) fix the arrival times at PIs to zero and limit the arrival time at POs to be less than the target time, respectively. The third constraint dictates that the arrival time at the output of each gate should be at least equal to the arrival time at each of its inputs plus the delay of the gate d_{gate}^{ABB} itself. The delay of a gate is expressed using the fourth and fifth constraints through the quantity s_{gate} which represents the amount of speedup [i.e., change in delay as shown in Fig. 2(b)]. Here, "d_0" and "d_1" are the degree-0 and degree-1 coefficients of the linear function between delay and gate bias (b_{gate}). As an example, for the gate shown in Fig. 2(b), these coefficients are -0.01 and 0.36, respectively. The last constraint sets the bounds on the bias voltage to ± 0.5 V.

We now develop the objective function. Fig. 2(a) showed our quadratic model for leakage as a function of the body bias. The total circuit leakage, which is the objective function to be minimized, then becomes

$$\text{minimize} \sum_{\forall gate} \Big[l_{gate}^{High\ V_{\rm th}} + (p_{0,gate} + p_{1,gate} \cdot b_{gate} + p_{2,gate} \cdot b_{gate}^2) \cdot l_{gate}^{High\ V_{\rm th}} \Big]. \quad (4)$$

Here, the coefficients p_0, p_1, and p_2 correspond to the degree-0, degree-1, and degree-2 coefficients, respectively, of

$$\left.\begin{array}{ll} AT_s = 0 & \\ AT_t \le Target & \\ AT_{ip} + d_{gate}^{ABB} \le AT_{op} & \forall\ input\ \text{"}ip\text{"}, output\ \text{"}op\text{"}, gate\ \text{"}gate\text{"} \\ d_{gate}^{ABB} = (1 - s_{gate}) \cdot d_{gate}^{High\ V_{\rm th}} & \forall\ gate\ \text{"}gate\text{"} \\ s_{gate} = d_{0,gate} + d_{1,gate} \cdot b_{gate} & \forall\ gate\ \text{"}gate\text{"} \\ -0.5 \le b_{gate} \le +0.5 & \forall\ gate\ \text{"}gate\text{"} \end{array}\right\} \quad (3)$$

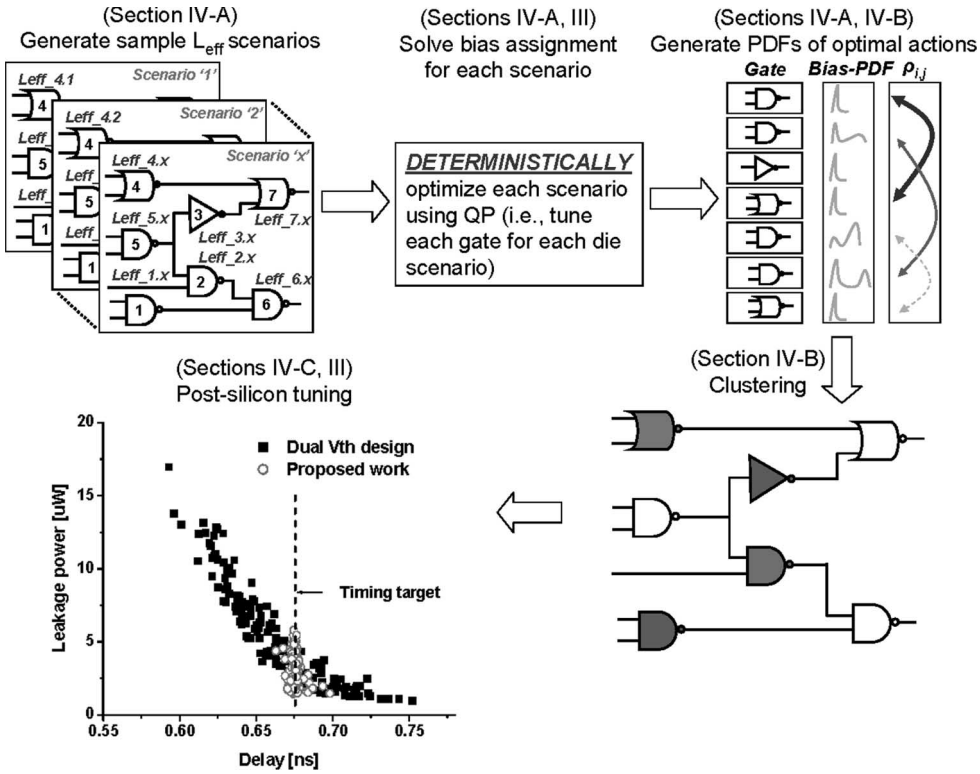


Fig. 7. Highlights of the proposed optimization framework.

the quadratic function relating leakage and bias. For instance, these coefficients are 0.03, 4.79, and 7.24 for the example gate in Fig. 2(a).

The optimization problem has thus been cast using linear constraints and a quadratic objective. In addition, the objective function is separable and convex since it is the sum of convex functions for each gate, as shown in Fig. 2(a). This type of optimization problem (separable convex quadratic objective subject to linear constraints) is amenable to very fast interior point algorithms.

There is no clustering of gates in this formulation—each gate is free to have its optimal post-silicon $b_{\rm gate}$ value, leading to a minimum leakage cost. The reasons for allowing this freedom in this formulation, along with our clustering algorithm, are described in Section IV.

IV. Proposed Variability-Aware Clustering

We now describe our variability-aware body-bias clustering methodology. Due to variability, each fabricated die exhibits a different on-die effective channel-length ($L_{\rm eff}$) distribution, leading to variation in delay and leakage. In the QP formulation of the previous section, this translates to the distributions of $d_{\rm gate}^{\rm High\ V_{th}}$ and $l_{\rm gate}^{\rm High\ V_{th}}$ rather than single deterministic values for these terms. Hence, the optimal solution found in the deterministic QP run of Section III will be nonoptimal for a general die (other than a perfectly nominal die). Ideally (assuming separate body control for every gate), we could solve the QP for each as-fabricated die and choose the optimal body biases for each gate on each die individually. This is exactly the opportunity that post-silicon tuning provides; however, as discussed in Section III, solving the QP leads to each gate having its own body bias, which is infeasible in practice. In general, only a handful of different biases will be allowable (the appropriate number represents a tradeoff between area/circuit overhead and improved yield), and hence, clustering the gates becomes critical. Once these clusters are determined at design time, each cluster can be separately tuned post-fabrication for each die. Our methodology achieves each of the aforementioned objectives in a three-phase process. The first phase obtains probability distributions of the body biases that would ideally be applied to each gate in the presence of variability. The second phase then clusters gates based upon these body-bias probability distributions and their correlations. Finally, after clustering the gates, the third phase tunes each cluster of each die to minimize power while meeting the delay. Fig. 7 shows this framework. We now describe each of these phases in detail using the simple seven gate c17 circuit to illustrate the concepts.

A. Body-Bias Probability Distributions

In this phase, we obtain the probability distributions of the body biases that would be applied to each gate to counteract the effects of variability. We begin by generating multiple "dies" drawing from the expected $L_{\rm eff}$ distribution for a given circuit in a Monte Carlo fashion and then by solving each scenario optimally using the described QP. Since each die differs from the others, we obtain distributions of body biases for each gate rather than single deterministic values. The quadratic formulation of the power–delay relationship helps us in this phase since by solving the QP for each scenario, we obtain the optimal body bias for each gate in that scenario (as each gate is free to choose its own body bias independently). Fig. 8 shows the frequency histograms of body biases for each gate in c17 (discretized

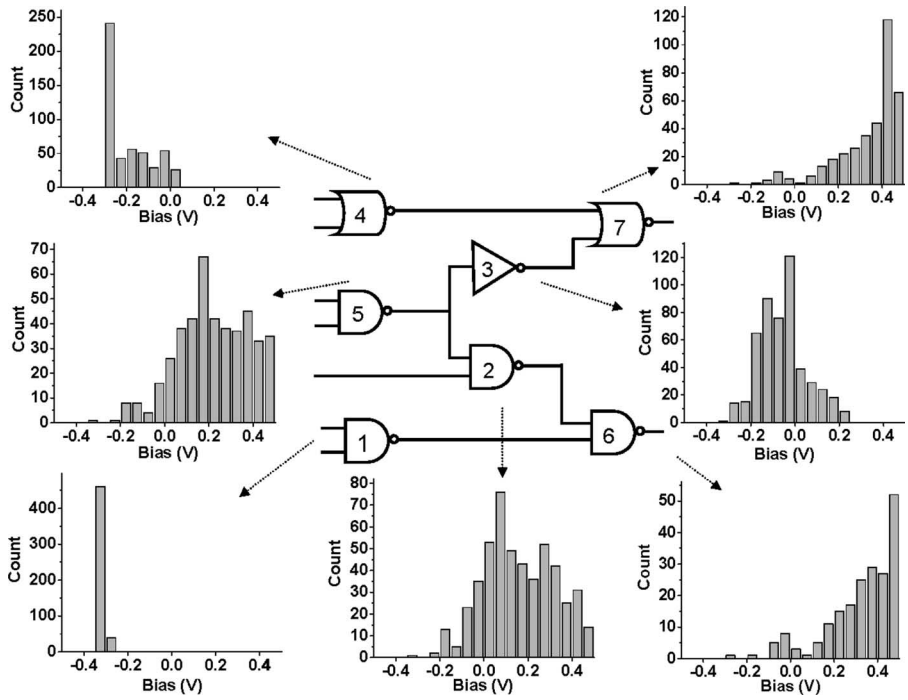


Fig. 8. Gate body-bias voltage distributions for 500 QP runs.

for drawing purposes only). The gate-level body-bias PDFs are then obtained from these frequency histograms.

In essence, the information stored in these PDFs is the optimal tuning action (i.e., amount of body bias voltage for each gate) that one would take post-silicon tuning for each unique die. These probabilities of tuning actions will now be used to form the ABB clusters.

B. Gate Clustering

The previous phase assumed that each gate has complete freedom for its body-bias value under all possible L_{eff} distribution scenarios. This freedom does not exist in practice since it is not possible for every gate to have its own separate body bias. Gates, hence, must be clustered, degrading the power/performance tradeoff and losing optimality. Once some gates are grouped into a cluster, they are constrained to have the same body bias. In order to meet timing, the body bias of each cluster is dictated by the timing critical gates in that cluster, implying that some gates may end up having more FBB (and, hence, more leakage) than in the ideal case. It is thus important to cluster the appropriate gates together that tend to have similar body-bias tuning assignments on a large number of dies to minimize the nonoptimality (thereby accommodating the subtlety in the optimization of post-silicon tunable circuits, as described in Section I, paragraph 3). Information contained in the probability distributions such as those shown in Fig. 8 is useful for this purpose.

In Fig. 8, we can see that some distributions are very similar to others. Properties of these PDFs, such as the mean, the standard deviation, and their correlations, can be used to guide clustering. Table I summarizes the properties of the probability distributions in Fig. 8. Table I(a) reports the mean and standard deviations of the body-bias PDFs, and Table I(b) is the corresponding correlation matrix.

TABLE I
PROPERTIES OF THE BODY-BIAS PDFS. (a) MEAN AND SIGMA. (b) CORRELATION MATRIX

(a)

Gate	Bias (V) μ	Bias (V) σ
1	-0.30	0.01
2	0.16	0.16
3	-0.06	0.11
4	-0.20	0.11
5	0.25	0.18
6	0.43	0.13
7	0.38	0.14

(b)

Gate	1	2	3	4	5	6	7
1	1.00	0.47	0.58	0.50	0.35	0.13	0.21
2	0.47	1.00	0.98	0.94	0.99	0.75	0.88
3	0.58	0.98	1.00	0.94	0.96	0.70	0.83
4	0.50	0.94	0.94	1.00	0.91	0.48	0.67
5	0.35	0.99	0.96	0.91	1.00	0.80	0.92
6	0.13	0.75	0.70	0.48	0.80	1.00	0.97
7	0.21	0.88	0.83	0.67	0.92	0.97	1.00

From this table, we find that Gates 2, 5, 6, and 7 are strongly correlated and also have similar PDF shapes (mean and sigma). It is therefore intuitive that these gates are good candidates to cluster together. Similarly, Gates 3 and 4 could be clustered together. On the other hand, Gates 1 and 7 are poor choices to cluster together since their means are very different and their correlation coefficient is also low.

When generalizing these ideas to larger circuits, we clearly need to develop a systematic procedure for clustering gates. To accomplish this, we first create an "adjacency graph" for the circuit. The adjacency graph for c17 is shown in Fig. 9(a). Every vertex corresponds to a gate, and every pair of vertices is connected by an edge in this graph. Some edges are shown in Fig. 9(a). Next, we assign a weight to every edge where the weight is given by an affinity function defined in (5). In this equation, "i" and "j" can be any two vertices

$$w(i,j) = k_1 M_{ij} + k_2\,(1-|\mu_i-\mu_j|) + k_3\,(1-|\sigma_i-\sigma_j|)\,. \quad (5)$$

M_{ij} is the correlation coefficient between the body biases of Gates i and j. μ_i, μ_j, σ_i, and σ_j are the respective means

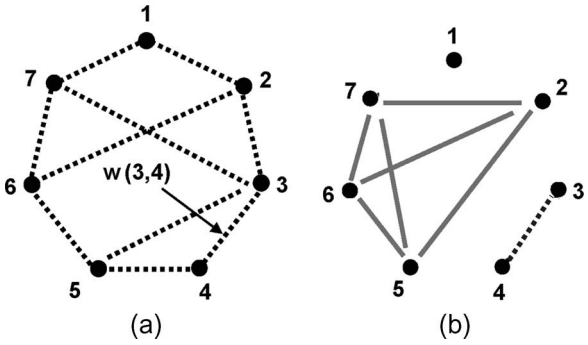


Fig. 9. (a) Adjacency graph. (b) Sample clustering with three clusters.

and standard deviations. k_1, k_2, and k_3 are the weight factors assigned to the correlation coefficient, the difference between means, and the difference between the standard deviations for Gates i and j. We empirically found that fixing k_1 and k_2 to 0.4 and k_3 to 0.2 gave good overall results; this skews the affinity more toward correlations and means. We also explored nonlinear affinity functions. However, since different gates have very different looking body-bias PDFs (Fig. 8), the results did not show a significant sensitivity to the exact form of the affinity function. The linear function previously shown was hence chosen largely for its simplicity.

We can see from (5) that gates that have body-bias PDFs that are more "like" each other (i.e., high correlations and similar body-bias means and standard deviations) have higher affinities and heavy edges between them. Since we seek to cluster similar gates, the problem of clustering reduces to the min-cut partitioning of the adjacency graph. The following greedy clustering algorithm "GREEDY_CLUSTER( )" which produces "N" clusters on completion is used to accomplish this.

GREEDY_CLUSTER() {

1. Create an empty bin for each of the "N" to-be-formed clusters.
2. Let $w(x^*, y^*) = \min_{\forall i,j}[w(i,j)]$.
   Put x^* in bin 1; y^* in bin 2. Flag x^* and y^* as covered.
3. While empty bins remain, {
   Choose an empty bin (say "X"),
   For every noncovered vertex "ν,"
   Calculate Affinity$(\nu) = \sum_{\forall y \text{ in nonempty bin}} w(\nu, y)$.
   Put ν^* in bin X, where ν^* has the minimum Affinity(ν).
   Flag ν^* as covered.
   }
4. For each of the remaining noncovered vertices (say "ν"), {
   For each bin (say "X"),
   Calculate Affinity$(\nu : X) = \sum_{\forall x \in X} w(\nu, x)/$ $\#vertices\ in\ X$.
   Put ν in bin X^*, where X^* has the maximum Affinity $(\nu : X)$.
   Flag ν as covered.
   }
5. Vertices in each of the "N" bins form the "N" desired clusters.

}

In Step 1, we first create empty bins for each cluster. Step 2 finds the pair of vertices with the minimum affinity and places them in separate bins, steering the algorithm toward pushing dissimilar gates apart. Step 3 continues populating bins until each contains exactly one vertex. In Step 4, we add each remaining noncovered vertex to the bin with which it has the maximum average affinity. By Step 5, the completed bins contain the desired clusters. Fig. 9(b) shows an example. The implemented partitioner includes multiple randomized initial starts in order to obtain a good min-cut solution.

The large-scale multiway partitioning tool METIS [19] could not be used here as it requires the user to provide the number of gates to be put in each bin. We, hence, implemented the aforementioned partitioner which begins with widely disparate gates (i.e., minimum edge weight) and then keeps growing clusters.

C. Post-Silicon Tuning

Once the clusters have been formed, the design-time optimization is complete. The adaptive nature of ABB which allows the tuning of each individual die can be modeled using a QP that is similar to the one described in Section III. The only difference between the formulation used here and that discussed earlier is that all gates in a given cluster are constrained to have the same bias (b_{gate}). We resorted to this mathematical emulation for the post-silicon step since this paper did not include fabrication. On silicon, this step can be accomplished using a methodology such as the one shown in Fig. 10 and described in the following paragraphs.

Under the proposed methodology, the body voltages of every cluster of every manufactured die would need to be swept in order to first obtain a power–delay envelope. The voltage sweep can either be exhaustive or a search method such as binary search. In order to ascertain whether the cluster body-voltage configurations meet the target timing, a conventional vector-based delay testing program may be used. Test programs routinely used for frequency binning can be reused for this purpose. Finally, predesignated fuses can be programmed to permanently store the optimal body-voltage solution for each chip (thereby permanently tuning the associated body bias generating analog circuits).

The body-voltage sweep phase can be significantly sped up if process-corner observability test structures (such as inexpensive ring oscillators) have been included in the layout at regular intervals. In this case, a simple lookup table can be set up that will directly identify the fuses to be programmed, based upon the variations sensed by the process monitors.

It is difficult to provide a generic solution for the post-silicon tuning step since it will have a strong dependence on top-level design decisions such as the availability of process-corner-indicating structures, the availability of electrically programmable fuses, etc. Literature related to this topic can also be found in references such as [4], [5], [20], and [21].

The post-silicon step will naturally result in a higher tester cost per die. This paper tries to mitigate this cost since the solution space is reduced to only a small number of clusters.

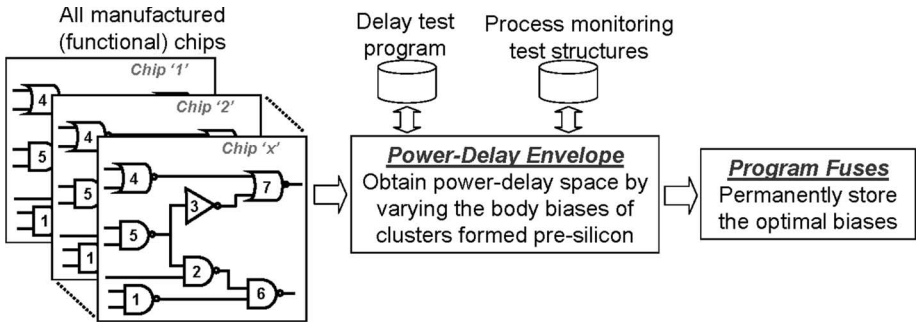


Fig. 10. Proposed tuning methodology.

TABLE II
LEAKAGE POWER AND DELAY COMPARISONS BETWEEN DUAL VTH AND ABB WITH ONE TO FOUR CLUSTERS. (a) LEAKAGE POWER. (b) DELAY

(a)

	Dual Vth			1 Cluster			2 Clusters			3 Clusters			4 Clusters		
(μW)	μ	σ	95%	μ	σ	95%	μ	σ	95%	μ	σ	95%	μ	σ	95%
c432	5.6	3.8	12.4	5.8	2.2	9.8	3.4	1.1	5.3	3.1	1.0	4.6	3.0	0.9	4.3
c499	26.7	20.1	63.7	25.5	10.7	42.9	16.1	6.3	26.5	14.8	5.6	24.4	14.6	5.4	23.6
c880	6.6	5.0	16.9	7.2	3.0	12.2	4.8	1.9	8.3	4.4	1.7	7.4	4.2	1.6	7.0
c1355	20.4	14.6	49.9	22.7	9.3	38.7	15.5	5.9	25.1	14.5	5.0	22.8	12.9	4.4	20.3
c1908	14.6	11.3	38.6	13.1	5.1	20.9	9.1	3.3	14.3	8.3	3.0	13.1	7.9	2.7	12.3
c2670	12.6	9.3	31.2	19.4	8.1	33.1	9.6	3.6	15.9	8.6	3.0	13.9	7.9	2.8	12.5
c3540	20.1	14.8	50.4	22.1	8.7	36.5	15.5	6.1	26.2	13.6	4.8	21.7	13.5	4.8	21.7
c5315	22.4	16.1	54.1	31.0	13.4	54.8	19.6	8.1	33.6	17.7	7.2	30.3	16.9	6.8	28.4
c6288	133.2	97.9	335.9	110.8	51.1	195.6	95.0	42.2	167.5	83.4	34.2	142.0	79.4	32.4	134.9
c7552	25.4	17.9	61.6	33.6	15.3	60.7	20.5	8.9	36.1	18.1	7.7	31.8	17.9	7.7	31.7
Viterbi	112.8	82.2	281.9	168.4	73.9	298.7	84.7	35.6	147.4	73.8	31.4	130.1	64.4	25.7	109.1
SOVA2	394.6	288.3	1031.1	520.9	243.3	928.7	280.4	118.6	488.3	244.7	101.7	402.4	236.2	94.5	392.2
Avg. % Improv. vs. Dual Vth				-17		18	27		51	35		57	38		59

(b)

	Dual Vth			1 Cluster			2 Clusters			3 Clusters			4 Clusters		
(ns)	μ	σ	95%	μ	σ	95%	μ	σ	95%	μ	σ	95%	μ	σ	95%
c432	0.66	0.03	0.72	0.67	0.00	0.68	0.68	0.00	0.68	0.68	0.00	0.68	0.68	0.00	0.68
c499	0.57	0.03	0.62	0.56	0.01	0.57	0.57	0.01	0.58	0.57	0.01	0.58	0.57	0.01	0.58
c880	0.69	0.03	0.74	0.69	0.00	0.69	0.69	0.00	0.70	0.69	0.01	0.70	0.69	0.00	0.70
c1355	0.73	0.04	0.80	0.74	0.01	0.74	0.74	0.01	0.75	0.74	0.01	0.75	0.74	0.01	0.75
c1908	0.99	0.05	1.08	1.00	0.01	1.02	1.01	0.01	1.02	1.01	0.01	1.02	1.01	0.01	1.02
c2670	0.68	0.04	0.73	0.67	0.00	0.68	0.67	0.00	0.68	0.67	0.00	0.68	0.67	0.00	0.68
c3540	1.08	0.06	1.18	1.08	0.01	1.09	1.08	0.01	1.09	1.08	0.01	1.09	1.08	0.01	1.09
c5315	1.00	0.05	1.08	0.99	0.01	1.02	0.99	0.01	1.02	0.99	0.01	1.02	0.99	0.01	1.02
c6288	2.95	0.15	3.18	3.00	0.14	3.39	2.99	0.06	3.12	2.99	0.06	3.11	2.99	0.07	3.13
c7552	1.20	0.06	1.30	1.20	0.02	1.23	1.20	0.02	1.23	1.20	0.02	1.24	1.20	0.02	1.24
Viterbi	3.70	0.21	4.05	3.69	0.05	3.79	3.70	0.05	3.79	3.70	0.04	3.79	3.70	0.05	3.80
SOVA2	4.13	0.22	4.48	4.07	0.07	4.19	4.07	0.06	4.20	4.07	0.05	4.19	4.08	0.05	4.19
Avg. % Improv. vs. Dual Vth				0		5	0		6	0		6	0		6

V. RESULTS

A. Body-Bias Clustering Leakage Power and Delay Analysis

1) Optimization With One to Four Clusters: Table II summarizes the main results of the proposed approach for leakage power and delay on circuits from the ISCAS85 benchmark set and two larger circuits ("Viterbi" and "SOVA2") with about 15 000 and 32 000 gates, respectively. Table II(a) and (b) reports the mean, the standard deviation, and the 95th percentile of leakage power and delay. The delay target in this set of experiments is 10% faster than the original all high $V_{\rm th}$ design.

Comparing the one cluster ABB design and the dual $V_{\rm th}$ design in Table II(a), we find that the mean power with the ABB design is, in fact, 17% worse (on average) than the dual $V_{\rm th}$. This is expected since non-critical gates are also supplied with the same forward bias now as required by the timing critical gates, leading to a large penalty in power. Thus, simply applying a single tunable body bias across the entire design is not viable, necessitating careful clustering.

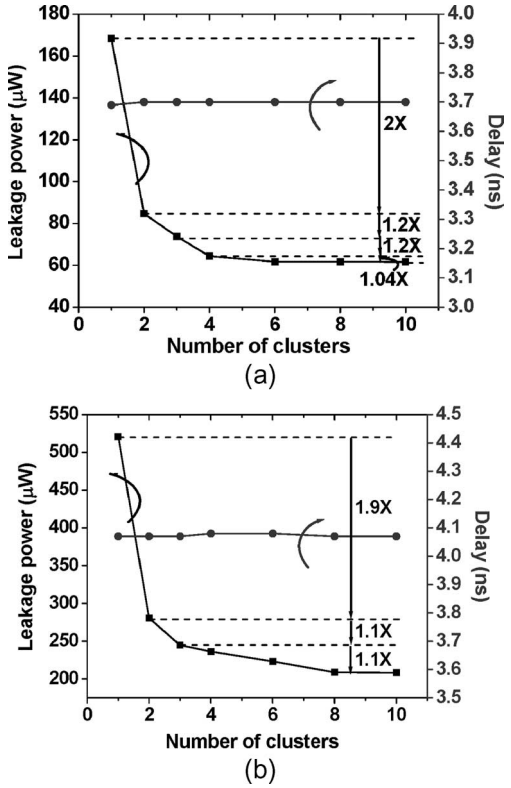


Fig. 11. Leakage and delay for additional clusters. (a) Viterbi. (b) SOVA2.

Moving to only two clusters, Table II shows that the resulting power/performance of the ABB designs significantly outperforms that of the dual $V_{\rm th}$ designs. In particular, considering the dual $V_{\rm th}$ design and the ABB design with the optimized two clusters, we find that the ABB designs reduce power by 38%–63% (95th percentile) and 12%–40% (mean) while tightening the delay spread (σ) by six times (average). These improvements grow when more clusters are allowed in the ABB designs.

2) Optimization With Additional Body Biases: Delay does not change significantly as the number of clusters in the ABB design is increased. This is expected since the QP solver can always find a solution of body-bias values that can make the circuit meet timing irrespective of the clustering. The major impact of fewer clusters is that dissimilar gates must be grouped together, leading to higher power levels (due to the same reason described for the one cluster ABB design). Fig. 11 quantifies this effect by showing the results of leakage and delay for the Viterbi and SOVA2 circuits as the number of allowed body-bias clusters is varied from one to ten. We find that power shows further improvements as more and more tunable clusters are provided. From Fig. 11(a), as the number of clusters increases from one to two, the power reduces by a factor of two times. On adding two more clusters, the power goes down further by factors of $1.2\times$. Diminishing additional power reduction is found beyond four clusters with only a $1.04\times$ improvement between four and ten clusters. This slowed rate continues beyond ten clusters, leading to a total improvement of only $1.8\times$ going from 10 to 14 539 clusters (i.e., number of clusters = number of gates, where each gate is allowed to optimally have its own independent body bias). Needless to say, it is completely impractical to realize the design with 14 539 clusters, and we have included this paragraph only to highlight the potential promise of adding more clusters and show the effectiveness of our clustering algorithm. Our clustering algorithm provides a significant fraction of the improvements of this best case design with only four clusters instead of 14 539. A similar trend is observed for the SOVA2 circuit in Fig. 11(b).

3) Importance of Optimizing the Formation of Clusters at Design Time (Pre-Silicon): This paper focuses on carefully identifying gates to be grouped together in an ABB scheme to limit overhead and maximize leakage savings. To quantify the importance of optimized clustering, we considered two possible alternative configurations of a design with two available body bias levels. In the first alternative configuration (because of the lack in the literature of a deterministic body-biasing algorithm with scalable and well-controlled ABB overheads), we used clusters found using the dual $V_{\rm th}$ algorithm [11] and employ ABB to tune the design. In the second alternative, the chip floorplan was partitioned into two halves, and all the gates in each partition formed the clusters (this alternative is akin to [20]). The results for these two simplistic clustering alternatives and our proposed clustering are compared in Table III for five representative circuits. The mean (95th percentile) power using our method is found to be 18%–43% (13%–47%) and 33%–59% (29%–68%) lower than the straightforward approaches for similar delays, thus underlining the importance of proper selection of gates in biasing bins.

We next examine the operation of the proposed greedy clustering algorithm GREEDY_CLUSTER. Fig. 12 is a scatter plot of the sigma and mean values of the body-bias PDFs for each gate in c2670 with three clusters. The correlation information is not seen directly in this figure; however, it becomes evident when one sees that the cluster boundaries overlap, suggesting that the clustering algorithm is inclined toward finding alike gates based not only on the standard deviations and means but also on the correlation coefficients.

4) Comparisons at Relaxed Timing Constraint: Results in Table II are for a stringent timing constraint, which was 10% faster than the original high $V_{\rm th}$ design. In order to examine the efficacy at a relaxed timing target, we report simulation results for five circuits in Table IV where the timing constraint is 5% faster than the high $V_{\rm th}$ design. We find stronger improvements in power and delay (as compared with Table II) in this case.

5) Sensitivity of Clustering to $L_{\rm eff}$ Distribution Models: Finally, Table V presents our analysis of the sensitivity of results to underlying models of $L_{\rm eff}$ distributions. As described earlier, our methodology operates on samples of $L_{\rm eff}$ distributions generated in the phase described in Section IV-A. Since these distributions rely on models of the underlying statistical silicon variations, it is pertinent to ask how sensitive the clustering results are to the accuracy of such models (since, for example, these models may be slightly different across different fabrication plants, or different runs at the same fabrication plant, spanning the development cycle of the product). We hence tested our clustering against such modeling uncertainties by: 1) forming clusters (Sections IV-A and B) based upon one set of $L_{\rm eff}$ distribution models, but 2) subjecting these sample dies to a second set of $L_{\rm eff}$ distribution models (obtained by

TABLE III
IMPORTANCE OF CONSIDERING TUNING AT DESIGN TIME. (a) LEAKAGE POWER. (b) DELAY

(a)

	Alternative #1			Alternative #2			Proposed Clustering		
(μW)	μ	σ	95%	μ	σ	95%	μ	σ	95%
c432	4.7	1.5	7.3	5.4	2.1	8.9	3.4	1.1	5.3
c499	23.1	8.9	39.0	23.7	9.4	38.5	16.1	6.3	26.5
c880	5.7	2.0	9.4	6.4	2.4	10.7	4.8	1.9	8.3
c1355	18.8	6.3	29.2	21.5	8.6	35.5	15.5	5.9	25.1
c1908	11.4	4.0	18.1	12.3	4.6	20.2	9.1	3.3	14.3

(b)

	Alternative #1			Alternative #2			Proposed Clustering		
(ns)	μ	σ	95%	μ	σ	95%	μ	σ	95%
c432	0.67	0.00	0.68	0.67	0.01	0.68	0.68	0.00	0.68
c499	0.56	0.01	0.58	0.56	0.01	0.57	0.57	0.01	0.58
c880	0.69	0.00	0.70	0.69	0.00	0.69	0.69	0.00	0.70
c1355	0.74	0.01	0.74	0.74	0.01	0.75	0.74	0.01	0.75
c1908	1.01	0.01	1.02	1.00	0.01	1.01	1.01	0.01	1.02

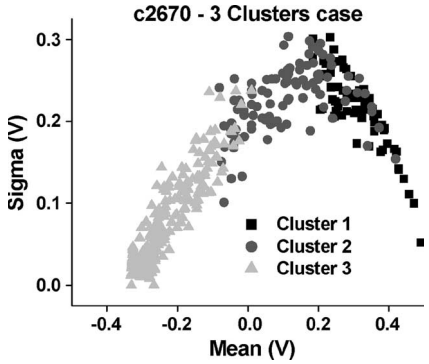


Fig. 12. Clusters using GREEDY_CLUSTER.

perturbing the correlation matrix by $+10\%$) during the tuning phase (Section IV-C). Results in Table V demonstrate that the ABB designs produced by our methodology continue to provide strong improvements in this eventuality as well (average 33% and 56% improvement in power μ and 95th percentile and average $6.3\times$ tighter delay σ).

To further study the sensitivity of our approach to the assumed models of variability, we applied a different set of variability parameters, as described by "Grid 2" in Section II-B. Results for this study are reported in Table VI. We find an average of 42% and 68% improvement in leakage μ and 95th percentile and an average of ten times tighter delay σ.

B. Runtime

Runtimes are reported in Table VII and Fig. 13. The reported time includes the time required for running the QP to generate the body-bias PDFs and the time required by the clustering algorithm. The reasonable runtimes of our approach are a direct result of the speed with which CPLEX can solve the quadratic formulation. As shown in Table VII, runtime is heavily dominated by the PDF generation step (column marked "QP"). From Fig. 13, CPLEX shows near-linear runtime, with the exception of SOVA2. Thus, the overall runtime for our framework can be made near linear (by parallelizing the QP runs if required for circuits such as SOVA2). The runtimes reported in Table VII are based on 600 samples on a single executing processor.

Fig. 14 shows the Monte Carlo convergence for the Viterbi circuit. Here, we compare the mean and 95th percentile power savings for the Viterbi design with two clusters (compared with the conventional dual $V_{\rm th}$ design) as the sample size is varied. This figure shows the results to have converged in about 600 samples. The PDF generation step can be further sped up by caching results from prior $L_{\rm eff}$ distribution-scenario runs and invoking the QP solver only if a newly encountered scenario differs from the cached ones significantly. Variance-reduction techniques, such as importance sampling [22], serve as excellent alternatives to the standard Monte Carlo.

C. Supporting Physical Design Methodology

We now describe the supporting physical design methodology. Physical design related issues arise when implementing designs with ABB due to bias control signal routing, well spacing between the adjacent cells having different bias and bias generation overhead. The bias generation overhead in our scheme is well controlled since we have demonstrated good results with only two to four clusters.

Since our clustering scheme is based on spatial correlations (which affect physically proximal cells similarly), clusters are inclined to be formed as contiguous regions naturally. However, it can certainly be the case that there are some instances where differently clustered gates (i.e., differently biased wells) are physically neighboring. Such gates need to be separated due to conditions imposed by triple-well layout rules and can lead to significant area and routing overheads.

To overcome this problem, we ran Capo [8]–[10] in an extension of the Engineering Change Order placement algorithm described in [10]. In this mode, Capo makes incremental changes to a given placement (which, in this case, is the initial placement used to form the correlation grid in Fig. 4) and can build contiguous regions of similarly clustered cells. As examples, Fig. 15(a) and (b) shows the resulting layouts after this step for the Viterbi circuit with two and three clusters

TABLE IV
DELAY AND LEAKAGE POWER COMPARISONS AT RELAXED TARGET TIMING (5% FASTER THAN THE INITIAL HIGH VTH DESIGN). (a) LEAKAGE POWER. (b) DELAY

(a)

	Dual Vth			1 Cluster			2 Clusters			3 Clusters			4 Clusters		
(μW)	μ	σ	95%	μ	σ	95%	μ	σ	95%	μ	σ	95%	μ	σ	95%
c432	4.0	2.8	9.1	3.9	1.3	5.9	2.4	0.8	3.7	2.1	0.6	3.1	2.0	0.7	3.0
c499	17.5	12.7	43.9	17.3	6.8	28.9	11.1	4.1	18.4	9.6	3.4	15.6	9.5	3.3	15.7
c880	4.6	3.3	11.4	5.0	2.1	8.6	3.3	1.3	5.5	3.2	1.2	5.1	2.8	1.1	4.6
c1355	16.3	11.7	39.8	15.2	5.9	25.3	10.1	3.7	16.1	9.0	3.1	14.1	8.0	2.7	12.5
c1908	10.5	8.0	27.1	8.7	2.7	13.2	6.0	1.9	9.0	5.5	1.7	8.3	5.1	1.5	7.6
Avg. % Improv. vs. Dual Vth				4		36	37		59	43		64	47		66

(b)

	Dual Vth			1 Cluster			2 Clusters			3 Clusters			4 Clusters		
(ns)	μ	σ	95%	μ	σ	95%	μ	σ	95%	μ	σ	95%	μ	σ	95%
c432	0.70	0.03	0.76	0.71	0.00	0.71	0.71	0.00	0.72	0.71	0.00	0.72	0.71	0.00	0.72
c499	0.61	0.03	0.66	0.59	0.00	0.60	0.60	0.00	0.60	0.60	0.00	0.60	0.60	0.00	0.60
c880	0.74	0.04	0.80	0.73	0.00	0.73	0.73	0.00	0.73	0.73	0.00	0.73	0.73	0.00	0.74
c1355	0.78	0.04	0.85	0.77	0.00	0.78	0.78	0.00	0.78	0.78	0.00	0.78	0.78	0.00	0.78
c1908	1.05	0.06	1.14	1.06	0.00	1.06	1.06	0.00	1.06	1.06	0.00	1.06	1.06	0.00	1.07
Avg. % Improv. vs. Dual Vth				1		8	0		8	0		8	0		8

TABLE V
SENSITIVITY OF RESULTS TO MODELS OF L_{eff} DISTRIBUTIONS. (a) LEAKAGE POWER. (b) DELAY

(a)

	Dual Vth			2 Clusters		
(μW)	μ	σ	95%	μ	σ	95%
c432	5.7	4.7	15.4	3.3	1.3	5.5
c499	25.3	19.2	66.5	15.8	6.4	26.6
c880	6.7	5.3	17.5	4.9	2.1	8.7
c1355	20.3	14.4	47.5	15.6	5.9	25.8
c1908	14.0	10.3	36.2	9.1	3.3	14.6

(b)

	Dual Vth			2 Clusters		
(ns)	μ	σ	95%	μ	σ	95%
c432	0.67	0.03	0.72	0.68	0.00	0.68
c499	0.57	0.03	0.62	0.57	0.01	0.58
c880	0.69	0.04	0.75	0.69	0.01	0.70
c1355	0.73	0.04	0.80	0.74	0.01	0.75
c1908	1.00	0.05	1.08	1.01	0.01	1.02

TABLE VI
RESULTS FOR GRID 2. (a) LEAKAGE POWER. (b) DELAY

(a)

	Dual Vth			2 Clusters		
(μW)	μ	σ	95%	μ	σ	95%
c432	5.5	5.3	17.5	2.7	1.1	4.7
c499	21.5	20.7	68.6	11.3	4.8	20.5
c880	5.5	5.2	17.7	3.7	1.6	6.6
c1355	17.2	16.6	54.8	10.8	4.4	18.4
c1908	12.0	11.5	38.1	6.9	2.8	12.2

(b)

	Dual Vth			2 Clusters		
(ns)	μ	σ	95%	μ	σ	95%
c432	0.66	0.03	0.71	0.67	0.00	0.68
c499	0.56	0.03	0.61	0.57	0.00	0.57
c880	0.68	0.03	0.73	0.69	0.00	0.69
c1355	0.72	0.04	0.78	0.74	0.00	0.74
c1908	0.99	0.05	1.07	1.01	0.00	1.01

(each cluster shown with a different color). Since Capo causes gates to move only by minimal distances, it was found that the layout in Fig. 15(a) and (b) has average and maximum gate displacements of about 1.7% and 12% (referenced to die length = 232 μm) as compared with the original layout, respectively. In addition, 96% of the gates do not leave their correlation-grid quadrant (Fig. 4) while the remaining gates (originally near quadrant borders) move by only one grid square (i.e., to the neighboring quadrant). Thus, the initial and final placements are very similar. To further study the impact of the slightly perturbed layout, we reran the tuning part of our approach (Section IV-C) for the designs with these final placements. Table VIII presents these results showing that results change negligibly.

We also studied the increase in area and wirelength. Half-perimeter wirelengths for the placements in Fig. 15(a) and (b)

TABLE VII
RUNTIME

	Gate count	Runtime (s)	
		Total	QP
c432	166	24	23
c499	519	48	48
c880	390	37	36
c1355	558	55	55
c1908	432	45	44
c2670	964	72	70
c3540	962	122	118
c5315	1750	168	163
c6288	2502	292	278
c7552	2102	179	170
Viterbi	14539	2160	1800
SOVA2	31931	16056	13200

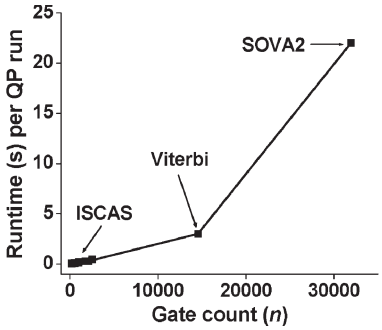


Fig. 13. CPLEX runtime per sample.

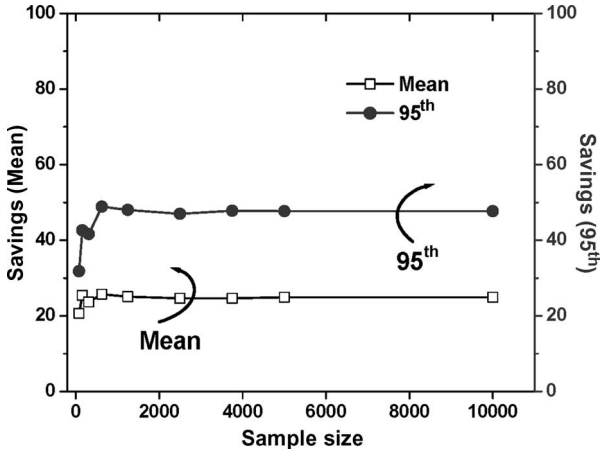


Fig. 14. Convergence of results for the Viterbi circuit.

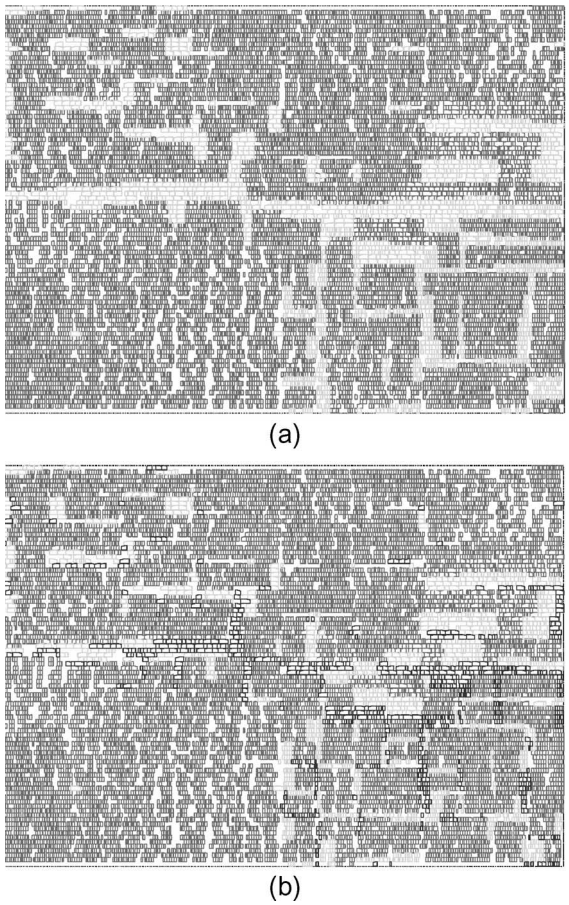


Fig. 15. Resulting layouts after running Capo to generate physically contiguous clusters for the Viterbi benchmark. (a) Viterbi placement with 2 clusters. (b) Viterbi placement with 3 clusters.

TABLE VIII
RETUNING WITH FINAL PLACEMENT. (A) LEAKAGE POWER. (B) DELAY

(a)

(μW)	Original Placement			Final Placement		
	μ	σ	95%	μ	σ	95%
Viterbi (2 clusters)	84.7	35.6	147.4	83.8	35.5	145.5
Viterbi (3 clusters)	73.8	31.4	130.1	73.1	31.0	128.7

(b)

(ns)	Original Placement			Final Placement		
	μ	σ	95%	μ	σ	95%
Viterbi (2 clusters)	3.70	0.05	3.79	3.70	0.05	3.80
Viterbi (3 clusters)	3.70	0.04	3.79	3.70	0.05	3.80

are only 2.3% and 3.1% higher than the original placements, respectively. From Fig. 15(a) and (b), instances where neighboring gates belong to different body-bias clusters and necessitate spacing are seen to have been greatly reduced by Capo. For well-separation rules of 2–3 μm in target 90-nm processes and given the white space in each standard cell row, the area overhead is about 5.2%–7.8%. These increases in wirelength and area are far outweighed by the improvements in power and delay demonstrated earlier. Note that our layout style will require some power grid rerouting for the bottommost metal layer. Finally, we believe that routing the bias control signals can be easily accomplished and is facilitated by this layout methodology as only a few contiguous regions need to be supplied with the bias voltages. In addition, circuit structures, such as the one described in [7], are potential techniques for biasing the connected wells of proximally clustered gates.

This physical design analysis demonstrates that it is indeed possible to implement the proposed clustering technique with well-controlled layout overheads.

VI. CONCLUSION

This paper proposed the first method that considers process variability for body-bias clustering to maximize yield using ABB. Our placement-aware work relies on the optimized clustering of gates to reduce the number of required on-die body biases to a small number (two to four). In comparison to the traditional technique of dual $V_{\rm th}$ assignment, we show that our physical design aware ABB approach can produce designs with two to nine times tighter delay distributions and leakage power reductions of 38%–68% while tightly controlling the area, wirelength, and bias routing overheads. We also demonstrated that adding more bias levels on the die provides rapidly diminishing returns on power reduction, suggesting that only a handful of biases are sufficient.

The general spirit underlying the work is that the post-silicon adaptive techniques require a fundamentally different optimization methodology which should be actively incorporated in the pre-silicon design cycle to enable high performance and parametric yield.

ACKNOWLEDGMENT

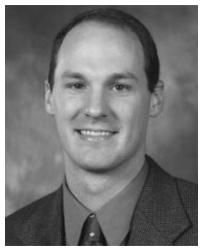
The authors would like to thank J. Roy and Prof. I. Markov of the University of Michigan for their helpful advice and kind support with Section V-C. S. H. Kulkarni would like to thank A. Agarwal of Intel for the helpful discussions on modeling process-parameter correlations. The authors would also like to thank the anonymous reviewers for some excellent feedback.

REFERENCES

[1] S. Nassif, "Delay variability: Sources, impacts and trends," in *Proc. ISSCC*, 2000, pp. 368–369.
[2] M. Mani, A. Devgan, and M. Orshansky, "An efficient algorithm for statistical minimization of total power under timing yield constraints," in *Proc. DAC*, 2005, pp. 309–314.
[3] J. Singh, V. Nookala, Z. Luo, and S. Sapatnekar, "Robust gate sizing by geometric programming," in *Proc. DAC*, 2005, pp. 315–320.
[4] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1396–1402, Nov. 2002.
[5] J. Tschanz, S. Narendra, R. Nair, and V. De, "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 38, no. 5, pp. 826–829, May 2003.
[6] V. Khandelwal and A. Srivastava, "Active mode leakage reduction using fine-grained forward body biasing strategy," in *Proc. ISLPED*, 2004, pp. 150–155.
[7] J. Gregg and T. Chen, "Optimization of individual well adaptive body biasing (IWABB) using a multiple objective evolutionary algorithm," in *Proc. ISQED*, 2005, pp. 297–302.
[8] A. Caldwell, A. Kahng, and I. Markov, "Can recursive bisection alone produce routable placements?" in *Proc. DAC*, 2000, pp. 477–482.
[9] J. Roy, S. Adya, D. Papa, and I. Markov, "Min-cut floorplacement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 7, pp. 1313–1326, Jul. 2006.
[10] J. Roy and I. Markov, "ECO-system: Embracing the change in placement," Univ. Michigan, Ann Arbor, MI, Tech. Rep. CSE-TR-519-06, 2006. [Online]. Available: http://web.eecs.umich.edu/techreports/cse/2006/CSE-TR-519-06.pdf
[11] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda, and D. Blaauw, "Duet: An accurate leakage estimation and optimization tool for dual-V_t circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 2, pp. 79–90, Apr. 2002.
[12] T. Chen and S. Naffziger, "Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 5, pp. 888–899, Oct. 2003.
[13] A. Keshavarzi, S. Narendra, S. Borkar, C. Hawkins, K. Roy, and V. De, "Technology scaling behavior of optimum reverse body bias for standby leakage power reduction in CMOS ICs," in *Proc. ISLPED*, 1999, pp. 252–254.
[14] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.
[15] A. Chandrakasan, W. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. New York: IEEE Press, 2001.
[16] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.
[17] F. Brglez and H. Fujiwara, "A neural netlist of 10 combinational benchmark circuits and a target translator in Fortran," in *Proc. ISCAS*, 1985, pp. 695–698.
[18] J. Fishburn and A. Dunlop, "TILOS: A polynomial programming approach to transistor sizing," in *Proc. ICCAD*, 1985, pp. 326–328.
[19] G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme for irregular graphs," *J. Parallel Distrib. Comput.*, vol. 48, no. 1, pp. 96–129, Jan. 1998.
[20] S. Kumar, C. Kim, and S. Sapatnekar, "Mathematically assisted adaptive body bias (ABB) for temperature compensation in gigascale LSI systems," in *Proc. ASP-DAC*, 2006, pp. 559–564.
[21] M. Olivieri, G. Scotti, and A. Trifiletti, "A novel yield optimization technique for digital CMOS circuits design by means of process parameters run-time estimation and body bias active control," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 5, pp. 630–638, May 2005.
[22] W. Cochran, *Sampling Techniques*, 3rd ed. New York: Wiley, 1977.

Sarvesh H. Kulkarni received the B.Tech. degree in electrical engineering and the M. Tech. degree in microelectronics from the Indian Institute of Technology, Bombay, India, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 2003 and 2006, respectively.

Since 2006, he has been a Senior Design Engineer with the Advanced Design Group, Portland Technology Development, Intel Corporation, Hillsboro, OR, where he has worked in the areas of programmable read-only memory and SRAM design. His research interests include design and design automation techniques for very large scale integration circuit optimization.



Dennis M. Sylvester (S'95–M'00–SM'04) received the B.S. degree in electrical engineering (*summa cum laude*) from the University of Michigan, Ann Arbor, and the Ph.D. degree in electrical engineering from the University of California, Berkeley (UC Berkeley), Berkeley, in 1999.

He previously held research staff positions with the Advanced Technology Group of Synopsys, Mountain View, CA, and the Hewlett–Packard Laboratories, Palo Alto, CA, and a visiting professorship in electrical and computer engineering at the National University of Singapore, Singapore. He is currently an Associate Professor with the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor. He has published numerous articles along with one book and several book chapters in his field of research, which includes low-power circuit design and design automation techniques, design for manufacturability, and interconnect modeling. He also serves as a Consultant and a Technical Advisory Board Member for several electronic design automation and semiconductor firms in these areas.

Dr. Sylvester is a member of the Association for Computing Machinery (ACM), the American Society of Engineering Education, and Eta Kappa Nu. He has served on the technical program committee of numerous design automation and circuit design conferences and the steering committee of the ACM/IEEE International Symposium on Physical Design. He was the General Chair for the 2005 ACM/IEEE Workshop on Timing Issues in the Synthesis and Specification of Digital Systems. He is currently an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN and previously served as an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS. His dissertation research was recognized with the David J. Sakrison Memorial Prize as the most outstanding research in the Electrical Engineering and Computer Science Department, UC Berkeley. He received an NSF CAREER Award, the Beatrice Winner Award at the International Solid-State Circuits Conference, an IBM Faculty Award, a Semiconductor Research Corporation Inventor Recognition Award, and several best paper awards and nominations. He was the recipient of the ACM Special Interest Group on Design Automation Outstanding New Faculty Award and the University of Michigan Henry Russel Award for distinguished scholarship.

David T. Blaauw received the B.S. degree in physics and computer science from Duke University, Durham, NC, in 1986, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, Urbana, in 1991.

Until August 2001, he was with Motorola, Inc., Austin, TX, where he was the Manager of the High Performance Design Technology Group. Since August 2001, he has been with the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, as a Professor. His work has focused on very large scale integration design and computer-aided design with a particular emphasis on circuit design and optimization for high-performance and low-power applications.

Dr. Blaauw was the Technical Program Chair and the General Chair for the International Symposium on Low Power Electronics and Design. He was the Technical Program Co-chair and a member of the Executive Committee for the ACM/IEEE Design Automation Conference. He is currently a member of the International Solid-State Circuits Conference Technical Program Committee.