A Statistical Framework for Post-Silicon Tuning through Body Bias Clustering

Sarvesh H Kulkarni Dennis Sylvester David Blaauw
EECS Department, University of Michigan, Ann Arbor, MI 48109, USA
{ shkulkar, dennis, blaauw } @eecs.umich.edu

ABSTRACT

Adaptive body biasing (ABB) 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 to a fixed design time based dual threshold voltage assignment method, we improve leakage power by 38-71% while simultaneously reducing the standard deviation of delay by 2-9X.

1. 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 [7]. 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 threshold voltages) to maximize yield [6,10]. 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 [3,12] that allow 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 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. 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 at the same time limiting overhead incurred by the added hardware. To effectively make the trade-off 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) [12] as the method for post-silicon tuning, but note that the methodology is generally applicable to other post-silicon tuning approaches as well.

Many issues arise while implementing an ABB scheme in practice. Although it is desirable to bias each gate in a design independently, supplying this many separate voltages inside a die is not viable due to well-spacing related 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 sub-optimal 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 later show, 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 [12], relatively little work has been performed in the area of design time optimization for ABB. In [5], a framework for assigning tuning voltages is cast as an integer linear program (ILP). However, the body voltages are fixed at design time using a deterministic formulation and post-silicon tuning is not considered. In [4], results for two small ABB enabled designs are presented. This work 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 overhead to practical levels, is not available in literature.

ICCAD'06, November 5-9, 2006, San Jose, CA

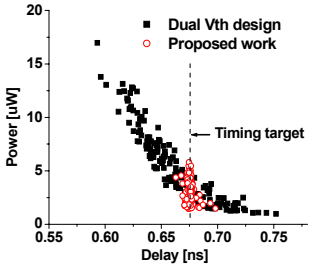


Figure 1. Power/delay scatter plots (dual Vth and ABB).

In this paper, we present a novel three-phase approach to gate-level body bias clustering considering variability. In the first phase, we compute for each gate the probability distribution functions 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 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 power and delay constraints. In addition, we present a methodology for obtaining compact layouts for ABB enabled circuits. By limiting the number of clusters to just a few, the overhead is already drastically reduced compared to approaches that use individual gate-level ABB control. We show that modern placers [1,8] can be used to incrementally perturb an initial placement leading to only small increases in area and wirelength and that the gains of the method far outweigh these small penalties.

We compare our approach to fixed dual threshold voltage (Vth) assignment [11] on a set of benchmark circuits. We show that with only 2-3 ABB clusters, the proposed approach yields significant improvements over dual Vth design. For instance, Figure 1 shows a scatter plot of leakage and delay for the c432 circuit for a traditional dual Vth design and for a design tuned using our work using three ABB clusters. The delay spread as well as the mean power is significantly reduced resulting in higher yield.

In summary, the key contributions of this paper are:

- This work presents the first gate-level optimization method for circuits enabled with ABB while taking process variations into account. We also present a physical design methodology which delivers tight control on placer overheads.
- A new gate-level framework for the optimization of post-silicon tunable circuits is presented. Although results in this paper focus on ABB as the underlying post-silicon tuning mechanism, the ideas are applicable to other tuning methods such as the tuning of Adaptive Supply Voltage (ASV) [3] domains.
- We show that it is important to consider post-silicon tunability during the *pre-silicon* design cycle in order to truly leverage the available post-silicon adaptivity.

Our paper is organized as follows. Section 2 provides background and describes our power/delay models and simulation setup. Section 3 describes our QP formulation for body bias assignment. In Section 4, we present the new variation-aware body bias clustering methodology for optimized post-silicon tuning. Section 5 presents results including an analysis of physical design implications. Finally, Section 6 summarizes our findings.

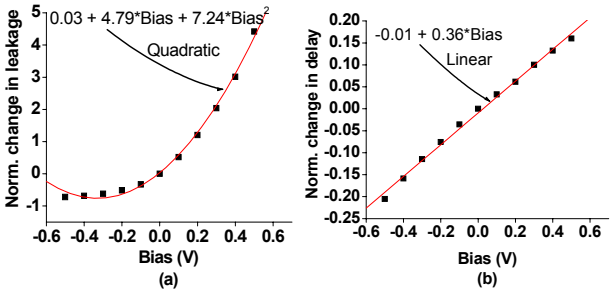


Figure 2. Power and delay modeling.

2. BACKGROUND

2.1 Power and delay models

Body biasing relies on the body effect phenomenon to modulate the Vth of a MOSFET. Eq. 1 gives the dependence of Vth of an NMOS transistor on the body-source (Vbs) voltage. V_{th0} is the nominal Vth at zero body bias, γ is the body coefficient, and Φ_F is the Fermi potential.

$$V_{th} = V_{th0} + \gamma\left(\sqrt{2\phi_F + V_{sb}} - \sqrt{2\phi_F}\right) \qquad (1)$$

Forward biasing (FBB) the body with respect to the source reduces Vth, increasing speed. However, because of the exponential dependence of leakage on Vth, it also leads to a large increase in power. Similarly, reverse body bias (RBB) reduces leakage at the cost of increased delay. This power-delay tradeoff enabled by 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 Vth gates which can be tuned using forward and reverse body bias. In comparison, in traditional dual Vth schemes the process needs to provide two different Vths. The lower Vth provides higher speed at the cost of power and is used on critical paths to meet timing in such dual Vth schemes.

Our work is based upon an industrial 1.2V 90nm triple-well dual Vth process. The two Vth values that are available are 0.32V (−0.33V) and 0.22V (−0.24V) for NMOS (PMOS). Body bias is varied between ±0.5V in our analysis for measuring delay and power changes (accounting for all components of leakage such as subthreshold leakage, body-source/drain junction diode leakage and band-to-band tunneling [9]).

Figure 2 shows our power and delay models. Figure 2a and 2b plot the change in leakage power (averaged across input states) and delay as body bias is varied between ±0.5V (normalized to the zero body bias). The exact relationship between the leakage and delay as body bias is varied is a complex non-linear function. However, we see that the change in leakage and delay can be modeled with good accuracy using quadratic and linear functions of the body voltage. A +0.5V forward bias can provide a speedup of 16% with a leakage increase of 4.4X, while -0.3V reverse bias reduces leakage by 38% while slowing the gate down by 11%.

2.2 Simulation setup

The standard cell library for the target 90nm process contains 2- and 3-input NOR and NAND gates and inverters, and the process provides a triple-well option which allows for body biasing. Cells are characterized using SPICE to quantify their delay and leakage

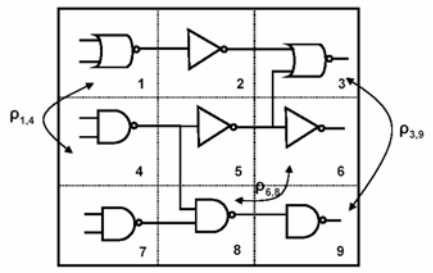


Figure 3. Grid for modeling spatial correlations.

across different Vth and body bias values. Cells are also characterized for their delay and leakage as channel length varies. In this work, we consider channel length as the source of variability. The implicit dependence of Vth variation induced by channel length variation is automatically captured in SPICE.

Our variability modeling is similar to [2]. Spatial correlations between gates are modeled by storing them in a grid-based correlation matrix (Figure 3). The correlation coefficients among the different quadrants of the grid are taken to be inversely proportional to the distance between them. We consider inter-die, spatially correlated intra-die as well as random components of variation. The 3σ/μ ratio for channel length was set to be 15%.

Test circuits taken from the ISCAS85 benchmark set are first sized up using a TILOS-based gate sizer using only high Vth gates. ABB clustering or low Vth assignment is then used to speed up the circuit. We consider speedups of 5% and 10% beyond the initially sized design. Our implementation of dual Vth assignment is based on a sensitivity-driven method presented in [11], which inserts low Vth gates on only those timing arcs that most improve timing.

We also present results for a DSP circuit ('Viterbi') with approximately 15000 gates to demonstrate the effectiveness of our work on larger designs.

3. QP-BASED 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 Monte Carlo simulations to obtain the distribution of body bias voltage across process variations. Consider the c17 circuit shown in Figure 4. *AT* represents the arrival time of the signal on a wire. All primary inputs (PI) and outputs (PO) are tied to supernodes '*s*' and '*t*'.

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 Vth gates (described in Section 2.2). These delay values will now be optimally reduced using a Quadratic Program such that the circuit meets the timing target. The constraints can be written as in Eq. 2:

$$\left.\begin{aligned} & AT_s = 0 \\ & AT_t \le Target \\ & AT_{ip} + d_{gate}^{ABB} \le AT_{op} && \forall\ input\ 'ip', output\ 'op', gate\ 'gate' \\ & d_{gate}^{ABB} = \left(1 - s_{gate}\right) \cdot d_{gate}^{HighVth} && \forall\ gate\ 'gate' \\ & s_{gate} = d_{0,gate} + d_{1,gate} \cdot b_{gate} && \forall\ gate\ 'gate' \\ & -0.5 \le b_{gate} \le +0.5 && \forall\ gate\ 'gate' \end{aligned}\right\} \quad (2)$$

The first and second constraints in (2) 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 + 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 plotted in Figure 2b). 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 Figure 2b these coefficients are -0.01 and 0.36 respectively. The last constraint sets the bounds on the bias voltage to ±0.5V.

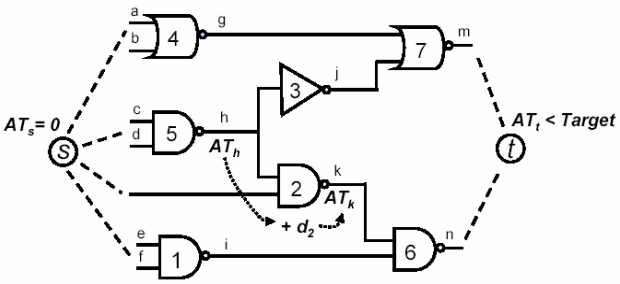


Figure 4. Setting up the QP for body bias assignment.

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

$$minimize \sum_{\forall gate} \left[l_{gate}^{HighVth} + \left(p_{0,gate} + p_{1,gate} \cdot b_{gate} + p_{2,gate} \cdot b_{gate}^2 \right) \cdot l_{gate}^{HighVth} \right] \quad (3)$$

Here, the coefficients p_0, p_1 and p_2 correspond to the degree-0, degree-1 and degree-2 coefficients of the quadratic function relating leakage and bias. For instance, these coefficients are 0.03, 4.79 and 7.24 for the example gate in Figure 2a.

The optimization problem has thus been cast using linear constraints and a quadratic objective. Also, the objective function is separable and convex since it is the sum of convex functions for each gate as seen in Figure 2a. 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_{gate} value, leading to minimum leakage cost. The reasons for allowing this freedom in this formulation along with our clustering algorithm are described next.

4. PROPOSED FRAMEWORK

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_{eff}) distribution, leading to variation in delay and leakage. In the QP formulation of the previous section, this translates to distributions of $d_{gate}^{HighVth}$ and $l_{gate}^{HighVth}$ rather than single deterministic values for these terms. Hence, the optimal solution found in the deterministic QP run of Section 3 will be non-optimal for a general die. Ideally, 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 3, solving the quadratic program leads to each

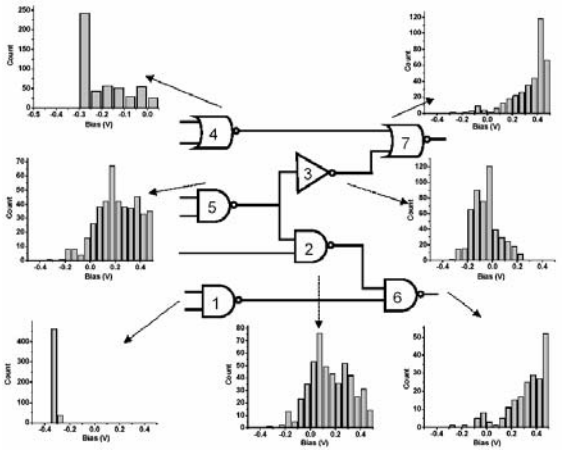


Figure 5. Gate body voltage distributions for 500 QP runs.

gate having its own body bias, which is infeasible in practice. In general, only a handful of different biases will be allowable 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 above discussed 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 delay. We now detail these phases using the simple seven gate c17 circuit for illustration.

4.1 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_{eff} distribution for a given circuit in a Monte Carlo fashion and then solving each scenario optimally using the described QP. Since each die differs from 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). Figure 5 shows the frequency histograms of body biases for each gate in c17. 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) one would take post-silicon for each unique die. These *probabilities of tuning actions* will now be used to form the ABB clusters.

4.2 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,

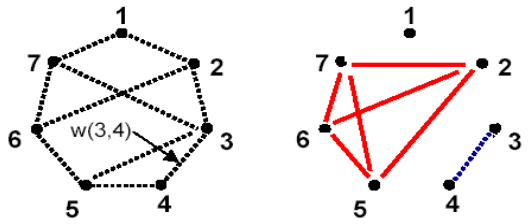


Figure 6. (a) Adjacency graph, (b) Sample partitioning.

Table 1. Properties of the body bias PDFs.

(a) Mean, Sigma

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) Correlation matrix

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

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 non-optimality (thereby accommodating the subtlety in the optimization of post-silicon tunable circuits as described in Section 1 Paragraph 3). Information contained in the distributions such as those shown in Figure 5 is useful for this purpose.

In Figure 5, we can see that some distributions are very similar to others. Properties of these PDFs such as the mean, standard deviation and correlations can be used to guide clustering. Table 1 summarizes the properties of the probability distributions in Figure 5. Table 1a reports the mean and standard deviations of the body bias PDFs while Table 1b is the correlation matrix for these PDFs.

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 as their means are very different and their correlation 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 Figure 6a. Every vertex corresponds to a gate and every pair of vertices is connected by an edge in this graph. Some edges are shown in Figure 6a. Next, we assign a weight to every edge where the weight is given by an affinity function defined in Eq. 4. In this equation, *'i'* and *'j'* can be any two vertices.

$$w(i,j) = k_1 M_{ij} + k_2\left(1 - \left|\mu_i - \mu_j\right|\right) + k_3\left(1 - \left|\sigma_i - \sigma_j\right|\right) \quad (4)$$

M_{ij} is the correlation coefficient between the body bias PDFs of Gates i and j. μ_i, μ_j, σ_i and σ_j are the respective means and standard deviations. k_1, k_2 and k_3 are weight factors assigned to the correlation coefficient, the difference between means, and the difference between standard deviations for Gates i and j.

Table 2. Power and delay comparisons between dual Vth and ABB with 1-4 clusters.

(a) Power

POWER	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%
c17	0.2	0.2	0.5	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1	0.2
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
Avg. % Improv. vs. Dual Vth				**-12**		**21**	**28**		**51**	**35**		**56**	**38**		**59**

(b) Delay

DELAY	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%
c17	0.06	0.00	0.07	0.07	0.00	0.07	0.07	0.00	0.07	0.07	0.00	0.07	0.07	0.00	0.07
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
Avg. % Improv. vs. Dual Vth				**0**		**5**	**0**		**5**	**0**		**5**	**0**		**5**

We can see from Eq. 4 that gates that have *body bias PDFs* that are more '*like*' each other (i.e., highly correlated body bias PDFs 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 non-covered vertex 'v',
   Calculate Affinity(v) = $\sum_{\forall y \text{ in non-empty bin}} w(v,y)$.
   Put v^* in bin X, where v^* has the minimum Affinity(v).
   Flag v^* as covered.
   }
4. For each of the remaining non-covered vertices (say 'v'), {
   For each bin (say 'X'),
   Calculate Affinity(v:X) = $\sum_{\forall x \in X} w(v,x) \Big/ \#\,vertices\ in\ X$.
   Put v in bin X^*, where X^* has the maximum Affinity(v:X). Flag v as covered.
   }
5. Vertices in each of the 'N' bins form the 'N' desired clusters.

}

4.3 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 similar to the one described in Section 3. The only difference here is that all gates in a cluster will be constrained to have the same b_{gate}. In practice, this step would be done by high-speed automated testing equipment.

5. RESULTS

5.1 ABB clustering power and delay analysis

5.1.1 Optimization with 1-4 clusters

Table 2 summarizes the main results of the proposed approach for leakage power and delay on circuits from the ISCAS85 benchmark set and a DSP circuit ('Viterbi') with roughly 15000 gates. Tables 2a and 2b report the mean, standard deviation and 95th percentile of power and delay. The delay target in this set of experiments is 10% faster than the original all high Vth design.

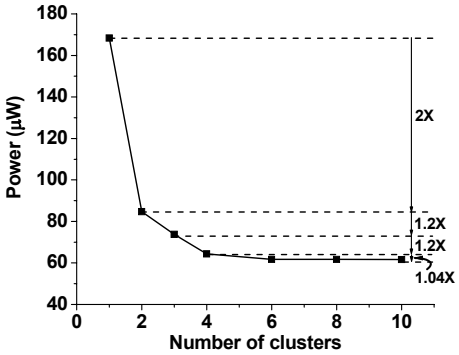


Figure 7. Power reduction with more clusters for Viterbi.

Comparing the one cluster ABB design and the dual Vth design in Table 2a, we find that the mean power with the ABB design is in fact 12% worse (on average) than dual Vth. 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.

Moving to only two clusters, Table 2 shows that the resulting power/performance of the ABB designs significantly outperforms that of the dual Vth design. In particular, considering the dual Vth design and the ABB design with the optimized 2 clusters we find that the ABB designs reduce power by 38−63% (95th percentile) and 12-42% (mean) while tightening delay spread (σ) by 7X (average). These improvements grow when more clusters are allowed in the ABB designs.

5.1.2 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 quadratic program 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). Figure 7 quantifies this effect by showing the results of average power for the Viterbi circuit as the number of allowed body bias clusters is varied from 1 to 10. We find that power shows further improvements as more and more tunable clusters are provided. As the number of clusters increases from 1 to 2, power reduces by a factor of 2X. On adding two more clusters, power goes down further by factors of 1.2X. Diminishing additional power reduction is found beyond 4 clusters with only a 1.04X improvement between 4 and 10 clusters.

This slowed rate continues beyond 10 clusters leading to a total improvement of only 1.8X going from 10 clusters to 14539 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 14539 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 4 clusters instead of 14539.

Table 3. Importance of considering tuning at design time.

(a) Power

POWER (μW)	Dual Vth Clustering			Proposed Clustering		
	μ	σ	95%	μ	σ	95%
c432	4.7	1.5	7.3	3.4	1.1	5.3
c499	23.1	8.9	39.0	16.1	6.3	26.5
c880	5.7	2.0	9.4	4.8	1.9	8.3
c1355	18.8	6.3	29.2	15.5	5.9	25.1
c1908	11.4	4.0	18.1	9.1	3.3	14.3

(b) Delay

DELAY (ns)	Dual Vth Clustering			Proposed Clustering		
	μ	σ	95%	μ	σ	95%
c432	0.67	0.00	0.68	0.68	0.00	0.68
c499	0.56	0.01	0.58	0.57	0.01	0.58
c880	0.69	0.00	0.70	0.69	0.00	0.70
c1355	0.74	0.01	0.74	0.74	0.01	0.75
c1908	1.01	0.01	1.02	1.01	0.01	1.02

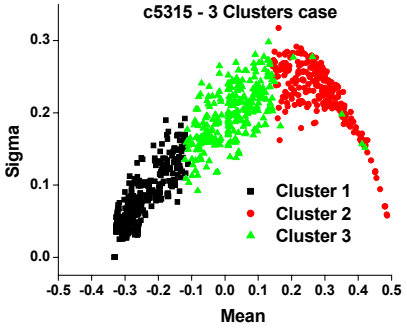


Figure 8. Effectiveness of the proposed clustering algorithm.

5.1.3 Importance of optimizing the formation of clusters at design time (pre-silicon)

This paper focuses on correctly identifying gates to group 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 configuration (because of the lack in literature of a deterministic body biasing algorithm with scalable and well-controlled ABB overheads) we used clusters found using the dual Vth algorithm [11] and employ ABB to tune the design. In the second configuration, we used the clustering produced by our proposed approach. The corresponding results for the dual Vth clustering and our clustering are given in Table 3 for five representative circuits. The mean power using our method is 18-43% lower than the straightforward approach using the dual Vth groups for similar delays, thus underlining the importance of proper selection of gates in biasing bins.

We next examine the effectiveness of the proposed greedy clustering algorithm *GREEDY_CLUSTER( )*. Figure 8 is a scatter plot of the sigma and mean values of the body bias PDFs (in Volts) for each gate in c5315 with 3 clusters. Gates in different clusters are shown by different symbols and colors. From the figure, we find that the clustering algorithm is successful in clustering similar gates.

5.1.4 Comparisons at relaxed timing constraint

Results in Table 2 are for a stringent timing constraint, which was 10% faster than the original high Vth design. In order to examine the efficacy at a relaxed timing target, we report simulation results

Table 4. Delay and power comparisons at relaxed target timing (5% faster than initial high Vth design).

(a) Power

POWER	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) Delay

DELAY	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 5. Runtime.

	Gate Count	Runtime (s)
c17	7	0.3
c432	166	6.8
c499	519	41.4
c880	390	24.0
c1355	558	52.1
c1908	432	33.7
c2670	964	130.1
c3540	962	191.9
c5315	1750	489.0
c6288	2502	1226.7
c7552	2102	628.1
Viterbi	14539	8640.1

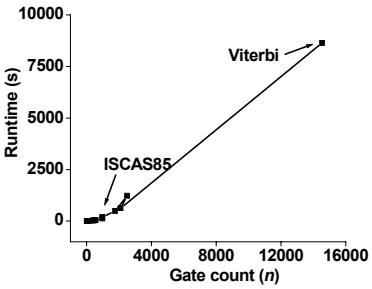


Figure 9. Runtime vs. gate count.

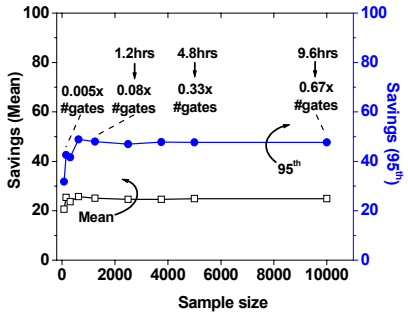


Figure 10. Power savings with varied sample size for Viterbi.

for five circuits in Table 4 where the timing constraint is 5% faster than the high Vth design. We find stronger improvements in power and delay (as compared to Table 2) in this case.

5.1.5 Sensitivity of clustering to L_{eff} distribution models

Like several statistical approaches in literature, our work operates on *models* of the underlying silicon variation. It is pertinent to ask whether the clustering shows fidelity, given the inaccuracies such models may possess. We analyzed this and found our conclusions to hold; details have however been left out due to space limitations.

5.2 Runtime

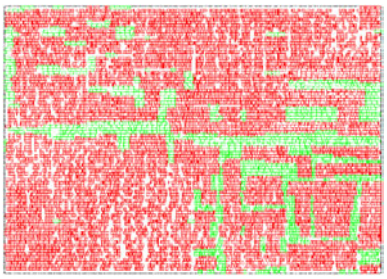
Runtime for our approach is reported in Table 5 and Figure 9. The reported time includes the time required for running the quadratic program to generate the body bias PDFs and the time required by the clustering algorithm. In reporting the results in this paper, the quadratic solver was invoked as many times as needed for results to converge. The acceptable runtime of our approach is a direct result of the speed with which the proposed quadratic formulation can be solved (Section 3). Figure 10 presents the dependence of our results on the number of times the QP solver is invoked. Here we compare the mean and 95th percentile power savings for the Viterbi design with two clusters (compared to the conventional dual Vth design) when the sample size is varied from 0.005n to 0.67n (where n is the number of gates in the circuit = 14539 for Viterbi). From this figure, power savings are found to be quite insensitive to the sample size for such large circuits. The quality of results with only 2500 samples (= 16% of gate count) is similar to that for higher sample sizes indicating that the number of times the QP solver needs to be invoked increases very slowly with circuit size, bringing the runtime down to 2.4hrs. PDF generation can be further sped up by caching results from prior L_{eff} distribution scenario runs or using sampling methods such as importance sampling which serve as excellent alternatives to standard Monte Carlo. Even without such speed up techniques, the complexity of our approach is found to be between linear and quadratic in n (Figure 9).

5.3 Supporting physical design methodology

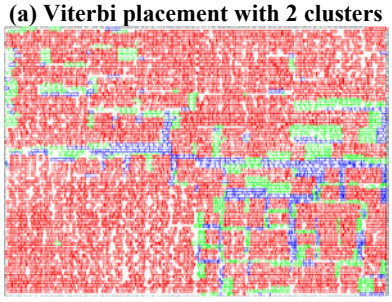
Physical design related issues arise when implementing designs with ABB due to bias control signal routing, well spacing between adjacent cells having different bias, and bias generation overhead. The bias generation overhead in our scheme is well controlled since we have demonstrated good results with only 2-4 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 [1,8] in an extension of the Engineering Change Order (ECO) placement algorithm described in [8]. 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 Figure 3) and can build contiguous regions of similarly clustered cells. As examples, Figure 11a and 11b show the resulting layouts after this step for the largest (Viterbi) circuit with 2 and 3 clusters (each cluster shown with a different color). Since Capo causes gates to move only by minimal distances, it was found that the layouts in Figure 11a and 11b have average and maximum gate displacements of about 1.7% and 12% (referenced to die length = 232μm) as compared to the original layout, respectively. Also, 96% of the gates do not leave their



(a) Viterbi placement with 2 clusters



(b) Viterbi placement with 3 clusters

Figure 11. Resulting layouts after running Capo to generate physically contiguous clusters for the Viterbi benchmark.

Table 6. Retuning with final placement.

(a) Power

POWER (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) Delay

DELAY (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

correlation grid quadrant (Figure 3) while the remaining gates (originally near quadrant borders) move by only one grid square (i.e., to the neighboring quadrant). Thus the initial placement and final placement are very similar. To further study the impact of the slightly perturbed layout, we reran the tuning part of our approach (Section 4.3) for the designs with these final placements. Table 6 presents these results showing that results change negligibly.

We also studied the increase in area and wirelength. Half perimeter wirelength for the placements in Figures 11a and 11b are only 2.3% and 3.1% higher than the original placements. From Figures 11a and 11b, 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 2-3μm in target 90nm 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.

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

6. CONCLUSIONS

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 (2-4). In comparison to the traditional technique of dual Vth assignment, we show that our physical design aware ABB approach can produce designs with 2-9X tighter delay distributions and power reductions of 38-71% while tightly controlling 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 post-silicon adaptive techniques require a fundamentally different optimization methodology which should be actively incorporated in the pre-silicon design cycle to enable high parametric yields.

7. ACKNOWLEDGMENTS

The authors sincerely thank Jarrod Roy and Prof. Igor Markov from the University of Michigan for their kind support and helpful advice with Section 5.3.

8. REFERENCES

[1] A. Caldwell *et al.*, "Can recursive bisection alone produce routable placements?," *Proc. DAC*, pp. 477-482, 2000.

[2] H. Chang *et al.*, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," *Proc. ICCAD*, pp. 621-625, 2003.

[3] T. Chen *et al.*, "Comparison of Adaptive Body Bias (ABB) and Adaptive Supply Voltage (ASV) for improving delay and leakage under process variations," *IEEE TVLSI*, pp. 888-899, 2003.

[4] J. Gregg *et al.*, "Optimization of individual well adaptive body biasing (IWABB) using a multiple objective evolutionary algorithm," *Proc. ISQED*, pp. 297-302, 2005.

[5] V. Khandelwal *et al.*, "Active mode leakage reduction using fine-grained forward body biasing strategy," *Proc. ISLPED*, pp. 150-155, 2004.

[6] M. Mani *et al.*, "An efficient algorithm for statistical minimization of total power under timing yield constraints," *Proc. DAC*, 309-314, 2005.

[7] S. Nassif, "Delay variability: sources, impacts and trends," *Proc. ISSCC*, pp. 368-369, 2000.

[8] J. Roy *et al.*, "ECO-System: embracing the change in placement," *Tech. Report CSE-TR-519-06*, Univ. of Michigan. web.eecs.umich.edu/techreports/cse/2006/CSE-TR-519-06.pdf

[9] K. Roy *et al.*, "Leakage current mechanisms and leakage reduction techniques in deep-submicron CMOS circuits," *Proc. IEEE*, pp. 305-327, 2003.

[10] J. Singh *et al.*, "Robust gate sizing by geometric programming," *Proc. DAC*, pp. 315-320, 2005.

[11] S. Sirichotiyakul *et al.*, "Duet: An accurate leakage estimation and optimization tool for dual-Vt circuits," *IEEE TVLSI*, pp. 79-90, 2002.

[12] 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*, pp. 1396-1402, 2002.