Body Bias Voltage Computations for Process and Temperature Compensation

Sanjay V. Kumar, Chris H. Kim, and Sachin S. Sapatnekar
Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455
sanjay, chriskim, sachin@ece.umn.edu

***Abstract*— With continued scaling into the sub-90nm regime, the role of process, voltage and temperature (PVT) variations on the performance of VLSI circuits has become extremely important. These variations can cause the delay and the leakage of the chip to vary significantly from their expected values, thereby affecting the yield. Circuit designers have proposed the use of threshold voltage modulation techniques to pull back the chip to the nominal operational region. One such scheme, known as Adaptive Body Bias (ABB), has become extremely effective in ensuring optimal performance or leakage savings. Our work provides a means to efficiently compute the body bias voltages required for ensuring high performance operation in gigascale systems. We provide a CAD perspective for determining the exact amount of bias voltages that can compensate both temperature and process variations. Mathematical models for delay and leakage based on minimal tester measurements are built, and a nonlinear optimization problem is formulated to ensure highest frequency operation under all conditions, and thereby minimize the overall circuit leakage. Three different algorithms are presented and their accuracies and run-times are compared. The algorithms have been applied to a wide range of process and temperature corners, for a 65nm and a 45nm technology node based process. A suitable implementation mechanism has also been outlined.**

***Index Terms* : Delay, Leakage, Adaptive Body Bias (ABB), Process Variations, Temperature Variations, Circuit Optimization**

I. Introduction

With continued technology scaling, the effects of on-chip variations have caused the delay and leakage of present day circuits to vary significantly from their nominal values. Two main contributors to on-chip variability arise from changes in process parameters, and changes in operating temperatures. Process variations occur due to proximity effects in photolithography, non-uniform conditions during deposition, random dopant fluctuation, etc. [1]. These cause fluctuations in parameters such as channel length, width, oxide thickness, as well as dopant concentrations, and result in variations in the delay, and the leakage of the circuit.

Changes in the operating temperature occur due to power dissipation in the form of heat. On-chip thermal variations have a significant bearing on the mobilities of electrons and holes, as well as the threshold voltage of the devices. An increase in the operating temperature causes the mobilities to decrease, thereby decreasing the on-current I_{on}, which, in turn, can reduce the speed of the circuit. Further, elevated temperatures also lead to an increase in the leakage current. On-chip variations can be categorized as lot-to-lot (L2L), wafer-to-wafer (W2W), die-to-die (D2D), and within-die (WID) variations [2].

Thus, the effect of on-chip variations has resulted in a large number of dies failing to meet the frequency-leakage requirements during testing, thereby decreasing the yield significantly. This has heightened the need for post-silicon tuning in order to salvage the dies, and ensure that transistor scaling remains economically viable. While the effects of process parametric variations require a one-time compensation as soon as the chip is fabricated, thermal variations are dependent on the operating environment and hence require a run-time compensation. A typical means of achieving post-silicon tuning to compensate for variations in circuits is through threshold voltage modulation.

Body biasing, as a means of threshold voltage modulation provides an effective knob to alter the delay and leakage of the circuit. Traditionally, it has been used in two different operational scenarios [3]. The first, known as static body biasing uses reverse body biasing when the microprocessor is in a stand-by state. This procedure is aimed at reducing the subthreshold leakage current. Algorithms to determine the optimal configuration that achieves the lowest leakage in the presence of latency constraints, have been described in [4]–[8]. Such schemes have been used in low power and embedded systems, where leakage power minimization is the main objective. The second scheme, known as adaptive body bias (ABB), involves recovering dies impacted by process variations through post-silicon tuning. Adaptive body bias is a dynamic control technique, used to tighten the distribution of the maximum operational frequency and the maximum leakage power, in the presence of WID variations. It was first proposed by Wann *et. al.* in [9] and was further explored by Kuroda [10] during the design of a DSP processor. The main goal of this scheme is to ensure that maximum number of dies operate in the highest frequency bin, thereby increasing the yield of the fabrication process [11], [12]. The focus of our work is such high performance systems, whose frequency of operation is desired to be maintained at the highest value.

Bidirectional adaptive body bias has been shown to reduce the impact of D2D and WID parameter variations on microprocessor frequency and leakage in [2], [11]–[14]. Typically, devices that are slow but do not leak too much can be Forward Body Biased (FBB) to improve the speed, whereas devices that are fast and leaky can be Reverse Body Biased (RBB) to meet the leakage budget. The work in [11], [15] performs process variation-based ABB, and divides the die into a set

of WID-variational regions. In each region, test structures that are replicas of the critical path, are built. The delay and leakage values of these test structures are measured, and are used to determine the exact body bias values that are required to counter process variations at room temperature. The application of a WID-ABB technique for one-time compensation during the test phase, in [11], shows that 100% of the dies can be salvaged, while 99% of them operate at frequencies within the fastest bin.

Traditionally, ABB has been used only to compensate for process variations [11], [13], [14]. However, on-chip temperature changes can also significantly vary the delay and leakage of nanometer-scale devices, thereby necessitating the mitigation of the effects of these thermal variations as well. Only a limited amount of work so far has addressed this problem, such as [16], which focuses purely on temperature effects. In this work, we apply a combination of temperature-based ABB, and a process-based ABB to permit the circuit to recover from changes due to both temperature and process variations. In order to be able to adaptively body bias all of our dies at all operating temperatures, we utilize an efficient self-adjusting mechanism that can sense the operating temperature, and thereby dynamically regulate the voltages that must be applied to the body of the devices to meet the performance constraints.

There are two kinds of control systems to select the body bias voltages, namely a look-up table based system [8] and a critical path replica based system [3], [11]. A detailed explanation of these control systems is presented in the next section. Our work assumes a look-up table based control system, where the body bias voltages must be pre-computed, so that they can be written into such a look-up table, so as to be able to compensate for both one-time (process) variations as well as run-time (thermal) variations. In order to populate the look-up table, this control scheme involves applying different body bias voltages to the CUT (circuit under test), measuring the delay and the leakage, and thereby choosing the most optimal configuration that meets the requirements. Expectedly, if there is a fine-grained distribution of body bias voltages, such enumeration schemes lead to a large amount of time spent on a tester, and hence may not prove to be cost-effective.

Thus, the main purpose of our work is to be able to efficiently determine the exact amount of bias required to achieve process and temperature compensation, and populate the look-up table, such that the time spent on the tester is minimized. We propose two methods to compute the final body bias values, namely the PTABB (Process and Temperature Adaptive Body Bias) algorithm and the PABB-TABB (Process Adaptive Body Bias-Temperature Adaptive Body Bias) algorithm. Both these methods use mathematical models to express the delay and leakage as functions of the NMOS and the PMOS transistor body bias voltages. A two variable nonlinear programming problem (NLPP) is formulated and an optimizer is used to determine the configuration that meets the delay requirement, and thereby minimizes the overall leakage.

While the PTABB algorithm involves measuring the delay and leakage at sample points for each individual die or WID-variational region, at each compensating temperature, the PABB-TABB algorithm involves measurements only at the nominal operating temperature. The PABB-TABB algorithm splits the original problem into two sub-problems, namely compensating for process variations at nominal temperature (PABB), and compensating for thermal variations under ideal process conditions (TABB). The final set of bias voltages is simply a combination of the PABB and TABB voltages. Thus, this scheme minimizes the number of tester measurements, and eliminates the need to test at each operating temperature.

The body bias voltages obtained using these two methods are compared against the golden results, determined by enumerating over the entire search space. The enumeration algorithm is suitably designed keeping in view of the nature of the solution, to reduce the overall run-time by pruning unnecessary computations. The PTABB and PABB-TABB algorithms are applied to different ISCAS85 combinational benchmarks, at various temperature and process corners, for a 65nm as well as a 45nm technology. The results demonstrate the ability of the PTABB and the PABB-TABB algorithms to closely predict the body bias voltages. Accuracy and tester time trade-offs between the various approaches are also explored. An architectural implementation for this scheme is also elaborated.

The rest of the paper is organized as follows. Section II elaborates the necessity of a look-up table based control system, and outlines the problem statement of populating the look-up table with the least amount of tester cost. We also provide a generic implementation architecture for this scheme. Section III presents the enumeration algorithm, the PTABB algorithm as well as the PABB-TABB algorithm. Section IV presents the results for ISCAS85 benchmarks synthesized on 65nm and 45nm PTM technologies [17]. Inferences drawn from this work are presented in Section V.

II. Body Bias Control Systems

In this section, we provide an overview of the body bias control mechanism, and define the problem statement. Our circuit block in consideration is a high performance digital VLSI system, whose frequency of operation we wish to maintain at a constant value, under all operating conditions. Process parameter variations can alter the delay of the various gates in the circuit, and hence can affect the overall operational frequency of the system. Thus, we must compensate for process variations. Similarly, an increase in the on-chip temperature can cause a reduction in the mobility of the electrons and the holes, and an increase in the subthreshold current, on account of reduction in the threshold voltage, V_{th}. The delay of the circuit increases if the effect of mobility dominates and this phenomenon is known as negative temperature dependence. The opposite effect, known as positive temperature dependence [18], [19] is seen in low-voltage operations, especially in the sub-90nm technologies due to the reduction in V_{th} with increasing temperature, and a subsequent increase in subthreshold current, that drives the gates faster. However, an increase in the subthreshold current implies larger leakage. Our algorithm applies to both these cases, which require different kinds of threshold voltage compensation,

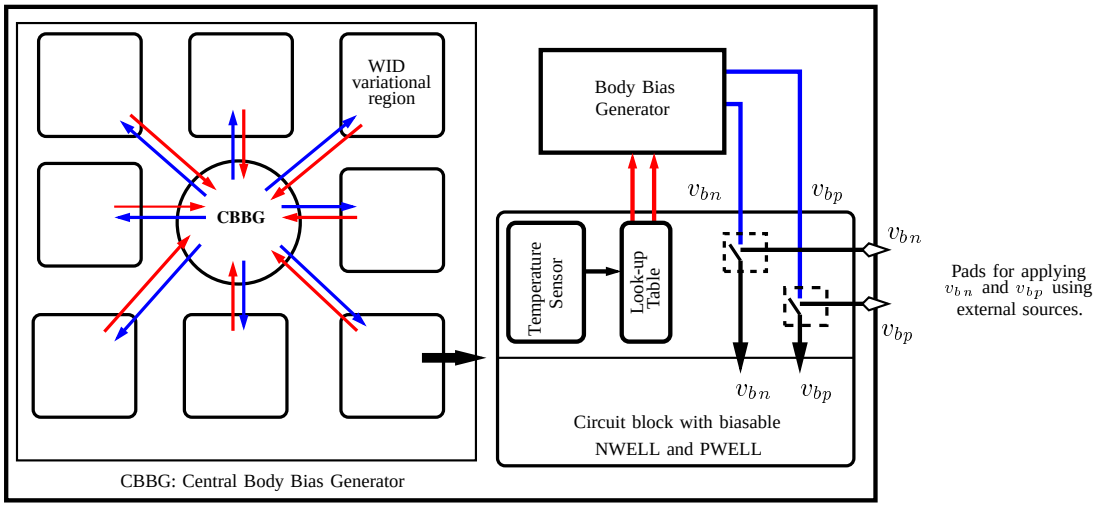

Fig. 1. A generic ABB implementation architecture showing the structure of the WID-variational regions

namely FBB to increase the speed of the circuit, or RBB to decrease the leakage current, respectively.

Thus, our key idea is to ensure that we counter the effect of process and temperature variations on the delay and the leakage of the circuit by body biasing our devices. Our experimental set-up assumes that the foundry is capable of supporting a triple well process, enabling us to bias both the N-well and the P-well, but the algorithm can be easily modified for any other process. Further, we assume that the target frequency of operation is determined by simulating the circuit at the nominal temperature (say, $T = 50°C$, for example), and ideal process conditions. The body bias pair, denoted by (v_{bn}, v_{bp}), when applied to the body of the NMOS and the PMOS[1] transistors, respectively, meets the delay requirement and minimizes the overall circuit leakage. The range of operating temperatures, and the extent of process variations, over which we are able to successfully bias the wells, each depends on the minimum and maximum limits imposed on the body bias voltages themselves, due to device physics restrictions. Additionally, the maximum amount of body bias is also constrained by the permissible leakage budget of the circuit block, since FBB reduces the delay at the expense of an increase in the leakage. The exact resolution of bias voltages is primarily determined by constraints on generating and routing these voltages to every biasable well in the circuit.

A. Overview of the Control Systems

As stated in the previous section, the control mechanism necessary to ensure that the requisite voltages are selected can either be built using a critical path replica based control system or a look-up table based control system. The hardware on-chip control set-up, as built in [3], [11], requires a test structure in the form of critical path replicas, which is expected to accurately reflect the behavior of the entire circuit, and the impact on delay and leakage due to on-chip variations. The control circuit consists of a delay monitor, phase comparator, decoder, digital-analog converter (DAC), and such other precision hardware to automatically select the bias pair, (v_{bn}, v_{bp}). Although such schemes are self-adapting, and require minimal post-silicon testing, a few sample critical path replicas might be unable to reflect the exact nature of process and thermal variations on the actual circuit, which consists of millions of paths. Experimental results in [11] indicate that a minimum of 14 critical path replicas per test-chip are required to accurately determine the die frequency of microprocessors, for a 130nm based process. The increased impact of process variations in sub-100nm technologies is likely to require a larger number of critical path replicas to be fabricated per test-chip to ensure a high level of confidence in the frequency measurements for a 65nm or a 45nm based design. This may lead to a substantial area overhead. Further, if the test circuits are large, they measure their own variations, which may not be the same as that of the actual circuit. Thus, the additional area overhead imposed by the number of critical path replicas and their inaccuracies, coupled with the need for PVT (process, voltage, and temperature) invariant hardware, call for better control mechanisms.

A viable alternative to the critical path replica based control system is the look-up table based control system. In this case, every block is equipped with a look-up table [3], [16] that can store the bias values (v_{bn}, v_{bp}). These are the precomputed optimal values that can compensate for thermal and process parametric variations. Each entry in the look-up table corresponds to a different temperature point. These entries are calibrated off-line through post-silicon measurements, with the aid of an efficient algorithm, i.e., using software. The look-up table is assumed to be built using a simple ROM like structure, and is populated during post-silicon testing. When the circuit is in operation, the entries in the look-up table are keyed based on the operating temperature, which is measured by a temperature sensor, as shown in [16]. The output of the table is fed to the body bias network to generate and route the appropriate voltages, thereby providing run-time compensation.

[1]The actual voltage applied to the body of the PMOS transistors is $(V_{dd} - v_{bp})$, where V_{dd} is the supply voltage.

The look-up table based control system eliminates the various issues associated with using critical path replicas as test structures, to capture the effect of process and thermal variations, on the entire chip. Since the body bias voltages are already precomputed, they may be immediately applied to the entire chip, to compensate for on-chip temperature variations, without affecting the run-time operation. An overall architectural implementation of this control scheme is explained in the next subsection.

Further, the effect of voltage variations, as well as aging, can be incorporated by adding appropriate sensors, and introducing an additional entry, i.e., supply voltage (V_{dd}), along with v_{bn} and v_{bp}, to the look-up table. The algorithms can be modified accordingly, to determine the optimal body bias and supply voltage configuration, to overcome the effects of process and thermal variations, and temporal degradation. A practical example of a system that uses the above scheme, and compensates for PVT variations, as well as aging, is seen in a 90nm-based design in [20].

B. Implementation

In this subsection, we provide a circuit implementation overview for the look-up table control scheme based body bias compensation network. Considering WID-variations, and assuming that both the N-well and the P-well can be body biased, we propose an implementation as shown in Fig. 1. The chip is partitioned into several WID-variational regions, each of which must be compensated independently. Our implementation assumes a central body bias network capable of generating the requisite voltage to each block. Alternatively, each block may have its own body bias generation and distribution network. Each WID-variational region is equipped with a temperature sensor that is capable of tracking variations in on-chip operating temperature. The temperature sensor references a ROM, that stores the (v_{bn}, v_{bp}) values for each compensating temperature, in the form of a look-up table. The output of the look-up table feeds the central (or local) body bias generator, and accordingly generates the required voltages. These voltages are then routed to the corresponding N and P wells. The NMOS and PMOS body bias voltages may be applied by external sources during testing. Once the final voltages are determined, and the look-up table has been populated, the switches can be closed and the requisite voltages required for compensation are supplied from the on-chip body bias generation network.

C. Problem Statement

While the look-up table based control circuit described above has minimal area overhead, the key to this approach lies in the efficiency of the software that generates the voltage values that must be written into this table. Unless this procedure is carefully devised, it could lead to a large amount of tester time, especially for a batch processing unit, such as manufacturing of microprocessors or ASICs, where the test time and time to market are extremely crucial. Thus, the crux of the problem lies in developing an efficient way of calculating the body bias voltages that can compensate for process and temperature variations, using minimal tester measurements. Our work tackles this problem, and we devise two different algorithms to determine the body bias voltages in order to populate the look-up table using minimum number of tester measurements. These algorithms are based on mathematical models for the delay and leakage of the circuit block, and are characterized based on minimal tester measurements. The performance of these algorithms is compared with a slower enumeration procedure that is always guaranteed to yield the optimal solution, if it exists.

III. Algorithms for PTABB

In this section, we explain the enumeration procedure, and the mathematically assisted ABB algorithms, namely the PTABB and the PABB-TABB algorithms for determining the body bias voltages, in order to populate the look-up table.

A. Enumeration

Algorithm 1 Enumeration (L_{max}, T_S, v_{step})

1: {L_{max} = Leakage budget for the circuit}
2: {T_S = Set of temperatures at which we are compensating for variations}
3: {There is one entry in the look-up table $\forall T \in T_S$}
4: Simulate the circuit with zero body bias at $T = T_0$ (nominal temperature), with ideal process parametric variables to obtain its delay D^*.
5: **for** each $T \in T_S$ **do**
6: {On-chip temperature of the CUT = T}
7: Apply (v_{bnmax}, v_{bpmax}) to the CUT.
8: Measure the best-case delay $D(v_{bnmax}, v_{bpmax})$
9: **if** $D(v_{bnmax}, v_{bpmax}) \geq D^*$ **then**
10: {Maximum FBB cannot meet delay; reduce the target frequency of operation.}
11: Choose target delay D^*, s.t. $D(v_{bnmax}, v_{bpmax}) < D^*$
12: **end if**
13: $L_{min} = \infty$
14: {v_{step} is the minimum resolution of bias that can be applied.}
15: **for** $v_{bn} = v_{bnmax}$: $-v_{step}$: v_{bnmin} **do**
16: **for** $v_{bp} = v_{bpmax}$: $-v_{step}$: v_{bpmin} **do**
17: Apply (v_{bn}, v_{bp}) to the CUT at temperature T.
18: Measure $D(v_{bn}, v_{bp})$ and $L(v_{bn}, v_{bp})$ on the tester.
19: **if** $D(v_{bn}, v_{bp}) \leq D^*$ **then**
20: {Feasible solution}
21: **if** $L(v_{bn}, v_{bp}) \leq L_{min}$ **then**
22: Solution = (v_{bn}, v_{bp})
23: $L_{min} = L(v_{bn}, v_{bp})$
24: **end if**
25: **else**
26: break
27: {Lower values of v_{bp} do not meet delay.}
28: **end if**
29: **end for**
30: **if** $D(v_{bn}, v_{bpmax}) \geq D^*$ **then**
31: break
32: {Lower values of v_{bn} do not meet delay.}
33: **end if**
34: **end for**
35: **if** $L_{min} \geq L_{max}$ **then**
36: {Leakage exceeds budget; must operate at a lower frequency.}
37: Increase target delay D^* iteratively.
38: Go to line 9.
39: **end if**
40: **end for**
41: Populate look-up table with (v_{bn}, v_{bp}) for each $T \in T_S$.

The task of enumeration is to traverse through the entire search space and find the optimal solution, i.e., the solution that meets the delay requirement, and thereby has minimal

leakage. However, since it is infeasible to find the delay and leakage over all possible values of v_{bn} and v_{bp}, we discretize the voltage levels and perform the enumeration over a limited set of values. Further, such a discretization is essential, since the body bias generation network is itself capable of generating only fixed number of voltage levels. The maximum amount of FBB that can be applied is restricted by the diode turn on voltage of the source-substrate junction and is process-dependent. The minimum resolution of voltage that can be applied is set by the designer and is constrained by the bias generation network.

A method for determining the values of the optimal bias pair points (v_{bn}, v_{bp}) is shown in Algorithm 1. We wish to operate the circuit at the highest possible frequency, and hence the desired delay D^* of the circuit under test (CUT), is pre-determined by a *simulation* at the nominal temperature, under ideal process conditions. The delay of the circuit under the influence of process and temperature variations is now measured on the tester, with the N-well and the P-well forward biased to the maximum extent, i.e., $v_{bn} = v_{bnmax}$, and $v_{bp} = v_{bpmax}$. This is the minimum delay of the circuit achievable using body bias. This step is performed to ensure that the delay of the circuit with maximum FBB is less than or equal to D^*.

If the maximum applicable bias fails to meet the target delay, i.e., if the effects of process and temperature variations on the delay are so drastic, that they cannot be negated by applying maximum FBB, the operational frequency of the circuit block must be reduced. Otherwise, we set this as our initial solution and seek solutions better than (v_{bnmax}, v_{bpmax}) within the search space, since (v_{bnmax}, v_{bpmax}) has a high leakage overhead. Each of the bias pair points is applied to the CUT, and the delay and leakage values are measured. Since the delay increases monotonically with decreasing body bias, if a bias pair (v_{bn1}, v_{bp1}) does not satisfy the delay requirement, all bias pairs with $(v_{bn} \leq v_{bn1})$ and $(v_{bp} \leq v_{bp1})$ fail to meet the delay requirement and hence can be directly eliminated. Thus, the search space can be effectively pruned during run-time. Eventually, the bias pair point that meets the delay requirement, and has the minimum leakage, is chosen as the optimal solution. If the leakage of the block exceeds the allocated leakage budget, then it implies that the amount of FBB required to meet the delay specifications causes the leakage to go beyond permissible limits, and the final solution is infeasible. Hence, we must decrease the target frequency such that lower amount of FBB can meet the delay, and thereby the leakage budget as well. The exact amount by which the target delay D^* must be increased depends on the topology of the circuit and may be determined iteratively, by enumerating and checking to see if the final solution meets the leakage budget or not.

It can be seen that if there are n different voltage levels for both v_{bn} and v_{bp}, the run-time is given by the time taken to iterate through the loops in lines 15 and 16, and is hence of the order $O(n^2)$. If there are k different temperature compensatory points, then the run time expressed in terms of the total number of tester measurements that must be performed per WID-variational block, is of the order $O(kn^2)$. The granularity of the body bias voltages, and the number of temperature compensatory points depend on the exact nature of the circuit, and the extent of variations that can be tolerated. Thus, while enumeration is guaranteed to yield the correct solution, the cost incurred in terms of the number of tester measurements required to populate the look-up table is extremely high, making it an expensive proposition if n and k are large. However, it must be noted that the run-time is actually dependent on the nature of the solution. If process variations have caused the devices to become slower, and if we are determining the bias values at some $T > T_0$ (assuming negative temperature dependence), then it is possible that the solution lies close to (v_{bnmax}, v_{bpmax}), and hence the procedure converges to the final solution in only a few iterations of the loops in lines 15 and 16.

B. Mathematically Assisted ABB Algorithms

While the enumeration algorithm is very accurate, a large number of delay and leakage measurements may be required before obtaining the final solution, and the cost incurred in testing may be extremely high. Hence, we seek algorithms which have a lower run-time as compared with the $O(n^2)$ enumeration procedure. In this subsection, we explore two such efficient algorithms that can reduce the run-time of the body-bias voltage selection process, without much loss in accuracy. Our algorithms are based on a simple nonlinear programming problem (NLPP) formulation that requires the tester measurements for delay and leakage at fewer sample points only (in comparison with the enumeration algorithm).

The mathematically assisted ABB algorithms are based on models for the delay and leakage of the circuit as a function of the body bias voltages, v_{bn} and v_{bp}. Since analytical expressions that can quantize the effect of body bias on the delay and the leakage at the circuit level do not exist, we use polynomial best fit curves to realize these models. Simulation results show that second order polynomials in both v_{bn} and v_{bp} provide a reasonably accurate model of the delay and the logarithm of the leakage. Thus, we have the expressions

$$D(v_{bn}, v_{bp}) = D_0 \sum_{i=0}^{2} \sum_{j=0}^{2} a_{ij} v_{bn}^j v_{bp}^i \quad (1)$$

$$L(v_{bn}, v_{bp}) = L_0 e^{\sum_{i=0}^{2} \sum_{j=0}^{2} b_{ij} v_{bn}^j v_{bp}^i} \quad (2)$$

where D_0 and L_0 are the delay and leakage values at the given operating temperature, and process conditions, without any body bias. Note that the coefficients in D and L can be obtained by simulating the circuit at well-spaced sample points. The desired accuracy for these curve-fitted expressions determines the number of points chosen to obtain the best-fit curve, although a minimum of nine points is required to uniquely determine the nine a_{ij} and the b_{ij} unknowns. These terms can be easily computed by using polynomial curve-fitting techniques.

In order to evaluate the accuracy of the model with respect to actual data, the delay and leakage values computed using the model in Equations (1–2) with nine sample points, are compared with the values from SPICE based simulations, over different v_{bn} and v_{bp} values. The results indicate that on

average, the delay and the leakage (logarithm of the leakage) computed using the model fall within 2-3% of the actual values obtained through simulations. Further, the models preserve the monotonicity of the delay and the leakage curves, with respect to increasing body bias values.

The NLPP can now be formulated as:

$$\text{minimize } L(v_{bn}, v_{bp}) = L_0 e^{\sum_{i=0}^{2}\sum_{j=0}^{2} b_{ij} v_{bn}^j v_{bp}^i} \quad (3)$$

subject to

$$\begin{aligned} D(v_{bn}, v_{bp}) = D_0 \sum_{i=0}^{2}\sum_{j=0}^{2} a_{ij} v_{bn}^j v_{bp}^i &\leq D^* \\ v_{bnmin} \leq v_{bn} &\leq v_{bnmax} \\ v_{bpmin} \leq v_{bp} &\leq v_{bpmax} \end{aligned} \quad (4)$$

where D^* is the desired delay constraint on the circuit under all operating conditions. The above problem can be easily solved using a standard nonlinear optimizer to obtain the final values of (v_{bn}, v_{bp}). We now present two different algorithms using the above framework to determine the body bias voltages for process and temperature compensation.

*1) **PTABB Algorithm**:* The PTABB (Process Temperature Adaptive Body Bias) algorithm solves the problem of optimal voltage selection by assuming a continuous search space in (v_{bn}, v_{bp}). However, since the final solution can take only a finite number of values (multiples of v_{step}), we propose a heuristic to discretize the results obtained. In the PTABB approach, the delay and the leakage values are measured at different well-spaced points along the (v_{bn}, v_{bp}) grid, and the coefficients in Equations (1) and (2) are computed. The NLPP is then solved and the final body bias pair is determined. The process is repeated for each compensating temperature. The procedure is described in Algorithm 2. The algorithm is similar to the enumeration procedure described in Algorithm 1, except that the doubly nested for-loops and subsequent computations in lines 15–34 of Algorithm 1 are replaced by simple measurements (lines 13–18 of Algorithm 2), followed by solving a two-variable NLPP to determine the optimal configuration. Note that the outermost for loop that runs for each $T \in T_S$ is exactly identical to that in Algorithm 1.

Unlike the enumeration procedure, the PTABB algorithm assumes a continuous search space. Hence the final solution must be snapped to the discrete grid space. Three options exist for snapping, namely:

1) Snap both v_{bn} and v_{bp} to the next higher voltage.
2) Snap v_{bn} to the next higher voltage while v_{bp} to the nearest lower voltage.
3) Snap v_{bp} to the next higher voltage while v_{bn} to the nearest lower voltage.

The delay and leakage of these three points are compared and the best solution is chosen. As seen from the results in the next section, the above heuristic gives accurate solutions.

It is clear from the algorithm that a minimum of nine tester measurements are required for characterizing the delay and the leakage models. In general, the number of tester measurements is equal to m^2, where m is the number of sample v_{bn}/v_{bp} values at which we are measuring the delay and leakage. Thus,

Algorithm 2 PTABB (L_{max}, T_S, v_{step})

1: $\{L_{max}$ = Leakage budget for the circuit$\}$
2: $\{T_S$ = Set of temperatures at which we are compensating for variations$\}$
3: {There is one entry in the look-up table $\forall T \in T_S$.}
4: Simulate the circuit with zero body bias at $T = T_0$ (nominal temperature), with ideal process parametric variables to obtain its delay D^*.
5: **for** each $T \in T_S$ **do**
6: {On-chip temperature of the CUT = T}
7: Apply (v_{bnmax}, v_{bpmax}) to the CUT
8: Measure the best-case delay $D(v_{bnmax}, v_{bpmax})$
9: **if** $D(v_{bnmax}, v_{bpmax}) \geq D^*$ **then**
10: {Maximum FBB cannot meet delay; reduce the target frequency of operation.}
11: Choose target delay D^*, s.t. $D(v_{bnmax}, v_{bpmax}) < D^*$
12: **end if**
13: **for** $v_{bn} = v_{bnmin} : \frac{(v_{bnmin}+v_{bnmax})}{2} : v_{bnmax}$ **do**
14: **for** $v_{bp} = v_{bpmin} : \frac{(v_{bpmin}+v_{bpmax})}{2} : v_{bpmax}$ **do**
15: Apply (v_{bn}, v_{bp}) to the CUT
16: Measure $D(v_{bn}, v_{bp})$ and $L(v_{bn}, v_{bp})$ on the tester.
17: **end for**
18: **end for**
19: Compute coefficients for delay and leakage in Equations (1) and (2).
20: Formulate NLPP and solve for $(v_{bn_{PT}}, v_{bp_{PT}})$.
21: Snap voltages $(v_{bn_{PT}}, v_{bp_{PT}})$ to discrete grid points (nearest v_{step} value) using heuristic.
22: {Final voltage pair denoted by (v_{bn}, v_{bp}).}
23: Compute $L_{min} = L(v_{bn}, v_{bp})$
24: **if** $L(v_{bn}, v_{bp}) \geq L_{max}$ **then**
25: {Leakage exceeds budget; must operate at a lower frequency.}
26: Increase target delay D^* iteratively.
27: Go to line 9.
28: **end if**
29: **end for**
30: Populate look-up table with (v_{bn}, v_{bp}) for each $T \in T_S$.

the run-time for the entire process is of the order $O(km^2)$, where k is the number of temperature points at which we are compensating for variations. Since m is generally less than n, the run time of the PTABB algorithm is better than that of the enumeration procedure[2]. However, unlike the enumeration procedure, which rapidly computes the final solution if it lies close to (v_{bnmax}, v_{bpmax}), the run-time of the PTABB algorithm is always fixed, since each circuit block requires the same number of tester measurements to characterize the delay and the leakage functions.

*2) **PABB-TABB Algorithm**:* Although the PTABB algorithm significantly improves the run-time, it requires a minimum of nine measurements at each compensating temperature. Besides, it may be time-consuming to test the CUT at each of the k different temperature values. Hence, in order to further reduce the time spent on the tester, we propose the PABB-TABB algorithm. The algorithm is based on the key observation that the effects of process and temperature variations on the circuit delay can be orthogonalized.

Decoupling Process and Temperature Variations: The delay of a gate can be expressed as the time taken to charge or discharge its capacitive load, and is given by:

$$D = \frac{C_L V_{dd}}{I_{avg}} \quad (5)$$

[2]If n is comparable with m, there may not be much savings obtainable with using PTABB algorithm. However, using a resolution of merely three or four different values for v_{bn}/v_{bp} in the body bias generation network is rather unlikely, and hence m can be assumed to be smaller than n.

where I_{avg} can be written as:

$$I_{avg} = \mu C_{ox} \frac{W}{L} (V_{gs} - V_{th})^{\alpha} \quad (6)$$

using the alpha-power law model. Note that μ is a function of temperature given by:

$$\mu = \frac{q}{kT} D \quad (7)$$

Further, V_{th} is given by:

$$\begin{aligned} V_{th} &= V_{th_0} + \gamma(\sqrt{|V_{SB} - 2\phi_F|} - \sqrt{|-2\phi_F|}) \\ \text{where } \phi_F &= \phi_T ln\left(\frac{N_A}{n_i}\right) \\ \text{and } \gamma &= \sqrt{\frac{2q\epsilon N_a}{C_{ox}}} \end{aligned} \quad (8)$$

The V_{th_0} term in the above equation is given by

$$\begin{aligned} V_{th_0} &= V_{fb} + \phi_S + \sqrt{\frac{4\epsilon q N_A \phi_S}{C_{ox}}} \\ \text{where } \phi_S &= \frac{kT}{q} ln\left(\frac{N_A}{n_i}\right) \end{aligned} \quad (9)$$

In Equations (5–9), the symbols have their usual meanings [21].

Note that while the operating temperature affects the mobility term in Equation (7), and the ϕ_S term in V_{th_0}, random fluctuations during deposition affect the dopant concentration N_A, and changes in device geometry due to proximity effects in photolithography [1] affect t_{ox}, W and L. Thus, it can be seen that process variations and thermal variations impact different parameters, and hence their effects are uncorrelated. In other words, if the process parameters are represented as a lumped vector $\mathbf{P}$, and temperature by T, then the delay of the circuit can be represented by the function $D(\mathbf{P}, T)$, where the elements of $\mathbf{P}$, and T are independent variables. Applying a Taylor series approximation about the point $(\mathbf{P_0}, T_0)$, which corresponds to the ideal process and nominal operating temperature case, we can write:

$$D(\mathbf{P}, T) \approx D(\mathbf{P_0}, T_0) + \bigtriangledown_{\mathbf{P}} D\Big|_{(\mathbf{P_0}, T_0)} \Delta\mathbf{P} + \frac{\partial D}{dT}\Big|_{(\mathbf{P_0}, T_0)} \Delta T \quad (10)$$

Further, assuming a locally linear approximation around the vicinity of $(\mathbf{P_0}, T_0)$, the delay at any other point $(\mathbf{P_1}, T_1)$ is given by:

$$\begin{aligned} D(\mathbf{P_1}, T_1) - D(\mathbf{P_0}, T_0) \approx\ & [D(\mathbf{P_1}, T_0) - D(\mathbf{P_0}, T_0)] + \\ & [D(\mathbf{P_0}, T_1) - D(\mathbf{P_0}, T_0)] \end{aligned} \quad (11)$$

The above equation can be re-stated as:

$$\Delta D(\mathbf{P}, T) \approx \Delta D(\mathbf{P})\big|_{T=T_0} + \Delta D(T)\big|_{\mathbf{P}=\mathbf{P_0}} \quad (12)$$

where $\Delta D(\mathbf{P}, T)$ is the increase in the delay around the nominal value $D(\mathbf{P_0}, T_0)$, $\Delta D(\mathbf{P})$ the increase in the delay due to process variations only, and $\Delta D(T)$ the increase in the delay due to thermal variations only. Thus, the change in the delay at any point can be expressed as the sum of the changes in delays due to process and temperature variations evaluated independently of each other. Note that the above approximation is valid since the range of delay values that can be compensated by ABB is not very large, and hence such an approximation does not lead to a significant loss of accuracy. We will support this by showing the results obtained through simulations on a ring oscillator.

Ring Oscillator Simulations: The validity of the above approximation is shown using Monte Carlo simulations performed on an 11 stage ring oscillator at various temperature and process corners, for a 65nm technology [17]. The results are shown in Fig. 2. The data is collected through a Monte Carlo simulation for 600 different simulation points that correspond to varying values of V_{thn} (threshold voltage of NMOS transistors), V_{thp} (threshold voltage of PMOS transistors), L_{eff} (effective length of the transistors), and T. All variables are assumed to be uniformly distributed with V_{thn} ranging from 0.415V to 0.431V (mean value μ = 0.423V), V_{thp} ranging from –0.373V to –0.357V (μ = –0.365V), L_{eff} ranging from 0.064um to 0.066um (μ = 0.065um), and T from 30 to 70°C (μ = 50°C). The percentage error in estimating the delay using Equation (12) is computed with respect to the actual delay values without this approximation, and the data is grouped into different percentage bins. The number of simulation points lying in each bin is plotted in the graph. As seen from the figure, the error in $\Delta D(\mathbf{P}, T)$ evaluated using the approximation in (12) as against the actual simulation results, (i.e., $\Delta D(\mathbf{P}, T) = D(\mathbf{P}, T) - D(\mathbf{P}_0, T_0)$) ranges between ± 1.36%, thus supporting the validity of the approximation in (12).

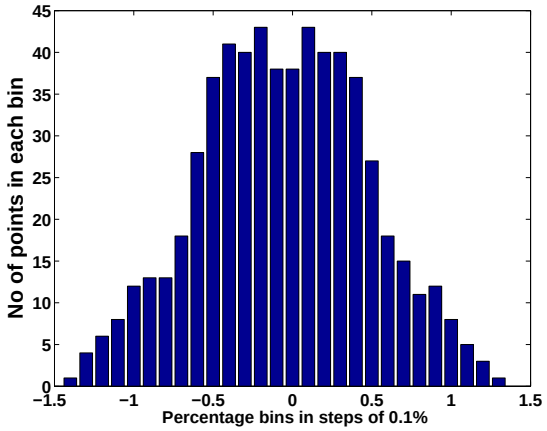

Fig. 2. Error in estimating the delay of the ring oscillator using Equation (12). The values in x-axis represent the percentage bins in steps of 0.1%: $[-1.4, -1.3)$....$[1.3, 1.4)$. The y-axis plots the number of simulation points lying in each bin.

PABB-TABB Computations: The decoupling of delay into process and temperature-dependent components enables us to consider the effect of process and temperature variations independently of each other, compensate for them separately, and finally merge the values. In other words, we can treat the given problem as two independent sub-problems:

- Compensation for process variations (PABB) at nominal operating temperature.
- Compensation for temperature variations (TABB) at ideal process conditions.

For a given WID-variational block, and for a certain temperature, each of these compensations provide one pair of body bias values that can be represented as (v_{bn_P}, v_{bp_P}) and (v_{bn_T}, v_{bp_T}), respectively. The final body bias voltages that can compensate for process as well as temperature variations can be computed using the following equation:

$$\begin{aligned} v_{bn_{PT}} &= v_{bn_P} + v_{bn_T} \\ v_{bp_{PT}} &= v_{bp_P} + v_{bp_T} \end{aligned} \quad (13)$$

In the above equation, we assume that the final body bias voltages after addition still satisfy the upper and lower bounds imposed by device physics restrictions. The proof of this equation is provided in the Appendix.

Summary of the Algorithm: Based on the above discussion, the PABB-TABB algorithm can be outlined as follows:

We split the original problem of finding the body bias pair at every compensating temperature point for each WID-variational region into two independent problems, namely temperature compensation at ideal process conditions (TABB) and process compensation at nominal operating temperature (PABB). Note that TABB involves deterministic simulations and can hence be performed at design time, prior to manufacturing. While the nonlinear programming approach as outlined in Section III-B.1 can also be applied to the TABB case, the body bias voltages (v_{bn_T}, v_{bp_T}) can simply be computed using the enumeration algorithm[3] as outlined in Section III-A, for better accuracy at the expense of larger simulation times.

For the TABB scheme, we perform one set of simulations at each compensating temperature, in order to characterize the delay and the leakage polynomials in Equations (1–2). Since, the polynomials consist of nine unknowns, nine simulations over different values of v_{bn} and v_{bp} are performed, and these polynomials are characterized. This step can be performed before fabrication, since it is performed on a "nominal design", i.e., assuming no process variations, to precompute the values of (v_{bn_T}, v_{bp_T}).

While the TABB scheme does not require any tester measurements, the PABB approach involves one set of tester measurements, i.e., a minimum of nine measurements, at the nominal temperature to characterize the delay and leakage functions in Equations (1) and (2), respectively. The voltages (v_{bn_P}, v_{bp_P}) can be computed by following the same method as outlined in Section III-B.1, with $T = T_0$. The final voltages for each temperature are computed by adding the TABB voltages with the PABB voltages. Note that this process of adding the individual voltages is physically valid only if the final voltages lie within the bounds imposed by device physics restrictions, (i.e., $v_{bnmin} \leq v_{bn} \leq v_{bnmax}$, and $v_{bpmin} \leq v_{bp} \leq v_{bpmax}$). Hence, if the addition causes the voltages to exceed the upper or the lower limits, a legalization procedure is necessary to ensure that the final voltages are valid. The legalization procedure formulates an NLPP with additional constraints, and forces the final voltages to lie within the limits.

The NLPP is formulated as follows:

$$\text{minimize } L(v_{bn_P}, v_{bp_P}) + L(v_{bn_T}, v_{bp_T}) \quad (14)$$

[3]Note that in this case, the measurements on the CUT at various points, as stated in the algorithm are replaced by deterministic circuit simulations.

Algorithm 3 PABB-TABB (L_{max}, T_S, v_{step})

1: $\{L_{max}$ = Leakage budget for the circuit$\}$
2: $\{T_S$ = Set of temperatures at which we are compensating for variations$\}$
3: Simulate circuit with zero body bias at $T = T_0$ (nominal temperature) and ideal process conditions to obtain its delay D^*.
4: At the nominal temperature T_0, measure the delay D, and leakage L of the CUT on the tester.
5: Apply maximum body bias to the CUT.
6: Measure the best-case delay $D(v_{bnmax}, v_{bpmax})$
7: **if** $D(v_{bnmax}, v_{bpmax}) \geq D^*$ **then**
8: {Maximum FBB cannot meet delay; reduce the target frequency of operation.}
9: Choose target delay D^*, s.t. $D(v_{bnmax}, v_{bpmax}) < D^*$.
10: **end if**
11: **for** $v_{bn} = v_{bnmin} : \frac{(v_{bnmin}+v_{bnmax})}{2} : v_{bnmax}$ **do**
12: **for** $v_{bp} = v_{bpmin} : \frac{(v_{bpmin}+v_{bpmax})}{2} : v_{bpmax}$ **do**
13: Apply (v_{bn}, v_{bp}) to the CUT at temperature T_0.
14: Measure $D(v_{bn}, v_{bp})$ and $L(v_{bn}, v_{bp})$ on the tester.
15: **end for**
16: **end for**
17: Compute coefficients in delay and leakage from Equations (1) and (2), respectively.
18: Formulate NLPP and solve for (v_{bn_P}, v_{bp_P}).
19: Compute $L_{min} = L(v_{bn_P}, v_{bp_P})$
20: **if** $L_{min} \geq L_{max}$ **then**
21: {Leakage exceeds budget; must operate at a lower frequency.}
22: Increase target delay D^* iteratively.
23: Go to line 7.
24: **end if**
25: **for** each $T \in T_S$ **do**
26: Pre-compute (v_{bn_T}, v_{bp_T})
27: $(v_{bn_{PT}}, v_{bp_{PT}}) = (v_{bn_P}, v_{bp_P}) + (v_{bn_T}, v_{bp_T})$
28: **if** $v_{bn_{PT}}$ or $v_{bp_{PT}}$ outside limits **then**
29: Legalize by solving for $(v_{bn_{PT}}, v_{bp_{PT}})$ using (14) and (15).
30: **end if**
31: Discretize by snapping to nearest v_{step} value (grid point).
32: {Final solution denoted by (v_{bn}, v_{bp}).}
33: **end for**
34: Populate look-up table with (v_{bn}, v_{bp}) for each $T \in T_S$.

subject to

$$\begin{aligned} D(v_{bn_T}, v_{bp_T}) &\leq D^* \\ D(v_{bn_P}, v_{bp_P}) &\leq D^* \\ v_{bnmin} &\leq v_{bn_T} \leq v_{bnmax} \\ v_{bnmin} &\leq v_{bn_P} \leq v_{bnmax} \\ v_{bn} &= v_{bn_T} + v_{bn_P} \\ v_{bnmin} &\leq v_{bn} \leq v_{bnmax} \\ v_{bpmin} &\leq v_{bp_T} \leq v_{bpmax} \\ v_{bpmin} &\leq v_{bp_P} \leq v_{bpmax} \\ v_{bp} &= v_{bp_T} + v_{bp_P} \\ v_{bpmin} &\leq v_{bp} \leq v_{bpmax} \end{aligned} \quad (15)$$

where $D(v_{bn_T}, v_{bp_T})$ and $L(v_{bn_T}, v_{bp_T})$ are the delay and leakage values from Equations (1) and (2) considering temperature variations only while $D(v_{bn_P}, v_{bp_P})$ and $L(v_{bn_P}, v_{bp_P})$ are the delay and leakage values from Equations (1) and (2) with process variations only. The limits v_{bnmin}, v_{bnmax}, v_{bpmin} and v_{bpmax} are determined by the process-technology used. The legalization procedure is a heuristic, and is mostly applied when compensating at high temperatures for the slow process corner, or at low temperatures for the fast process corner. The procedure is necessary because in most cases the optimal solution has RBB for NMOS in order to minimize

the leakage, and FBB for PMOS to restore the speed. Hence, for extreme process and temperature corners, the summing in Equation (13) may cause the voltages to exceed the limits. The complete algorithm is outlined in Algorithm 3. The final solution must still be discretized, and the same heuristic as that used for the PTABB case can be used here.

Time Complexity of PABB-TABB Algorithm: The key aspect of the PABB-TABB algorithm is that it requires only one set of tester measurements at the nominal temperature, since the temperature compensatory terms are pre-computed, during the design stage itself. Thus the run-time of the algorithm is $O(m^2)$, where m is the number of different v_{bn} (or v_{bp}) points at which we are measuring the delay and leakage, since we require only one set of measurements at the nominal temperature for process compensation. If we choose m as three, then the run-time is practically a constant. The results of the PABB-TABB algorithm, as explained in the next section show that the method is accurate in terms of determining the optimal body bias voltages, and thus provides a good run-time/accuracy trade-off.

A summary of the three algorithms described above is presented in Table I.

C. Temperature-Leakage Feedback in Circuits

Traditionally, the delay of a logic gate increases with temperature due to the reduction in the mobilities of the electrons and the holes. The leakage of the circuit also increases at higher temperatures due to the increase in the subthreshold conduction upon a decrease in the threshold voltage. Since the speed of the circuit decreases at higher temperatures, our control scheme requires the application of FBB to restore performance. This causes an increase in leakage, which can further increase the on-chip temperature, thereby leading to the possibility of a positive feedback loop culminating in thermal runaway.

However, it must be noted that a reasonably good nominal design will not be at the edge of the strong temperature-leakage feedback point, and certainly not close to thermal runaway due to process and temperature variations. Hence, the control scheme presented in the paper is justified for a high performance system. However, if the design is constrained by power, reverse body bias may be applied at higher temperatures to recover leakage, at the expense of a reduction in speed. Under such schemes, our algorithms determine the least amount of reverse body bias, sufficient to ensure that the leakage is within budget, thereby still maximizing performance.

Further, with technology scaling, and the rising impact of subthreshold conduction, the decrease in V_{th} with temperature may dominate the decrease in the mobility of devices, and therefore lead to a trend, where the circuits run at higher speeds, at increasing temperatures. This scenario is known as positive temperature dependence or inverted temperature dependence [22]–[24]. Under such circumstances, at higher temperatures, reverse body bias may be applied, without loss in performance, thereby ensuring that the leakage is within the budget. Similarly, at lower temperatures, forward body bias may be applied to speed up the circuits. At lower temperatures, since the nominal leakage is significantly lower than the budget, the overhead due to forward body bias still does not cause the leakage to exceed the budget. This control mechanism is particularly desirable, since it leads to a negative feedback at higher temperatures.

Thus, while the exact nature of entries in the look-up table depends on the temperature dependence of the circuit, the delay of the circuit at any given temperature is a monotonically decreasing function of v_{bn} and v_{bp} (within the limits of operation). Hence, the optimal body bias selection algorithms work independently of positive or negative temperature dependence of the circuit. For circuits that show negative temperature dependence, as we shall see from the results in the next section, the amount of body bias required to compensate for temperature variations increases with temperature.

IV. Simulation Results

In this section, we test the enumeration, PTABB, and PABB-TABB algorithms by performing a series of simulations to determine the optimal body bias voltages, which are written into the look-up table. Our experimental setup assumes that the test-chip consists of ten different ISCAS85 combinational benchmarks of various sizes. Further, the chip is partitioned such that each of these benchmarks is placed in a separate WID-variational region. Each of the ten WID-variational regions is equipped with a look-up table, and a temperature sensor, as shown in Fig. 1. Simulations are performed on these combinational benchmarks, synthesized using SiS [25], on PTM [17] 65nm and 45nm technologies. We have chosen T = 50°C as the nominal operating temperature, and the supply voltage V_{dd} as 1V, for both the technologies. A library consisting of 5 NOT gates, 5 NAND2 gates, 5 NOR2 gates, 3 NAND3 gates, and 3 NOR3 gates, of different sizes is considered for synthesis. We further assume that the range of body bias voltages that can be applied to the bodies of the NMOS and PMOS devices is -0.4V to 0.4V. In order to demonstrate the ability of the algorithms to compensate for temperature variations, the benchmarks are simulated at T = 35°C, T = 50°C, and T = 65°C. Similarly, the impact of process variations is simulated by altering the V_{th} of both the NMOS and the PMOS devices, and L_{eff} of all transistors, as shown in Table II. The effect of process variations is simulated by choosing the parameters for the "fast" and "slow" process corners as follows:

TABLE II
PROCESS CORNERS

	65nm Technology			45nm Technology		
	Nominal	Fast	Slow	Nominal	Fast	Slow
$V_{thn}(V)$	0.423	0.416	0.430	0.466	0.456	0.475
$V_{thp}(V)$	-0.365	-0.359	-0.371	-0.412	-0.403	-0.42
$L_{eff}(\mu$m)	0.065	0.064	0.066	0.045	0.044	0.046

1) ±1.5% variation in V_{thn} and V_{thp} over the nominal values for 65nm technology, and ±2% variation for 45nm technology.

TABLE I

SUMMARY OF THE ALGORITHMS (n = NUMBER OF BODY BIAS VOLTAGES, m = NUMBER OF v_{bn}/v_{bp} VALUES FOR INTERPOLATION, k = NUMBER OF TEMPERATURE COMPENSATORY POINTS)

	Enumeration	PTABB Algorithm	PABB-TABB Algorithm
Accuracy	Highest	Intermediate	Lowest
Search Space	Discrete	Continuous	Continuous
Run Time	Slowest	Intermediate	Fastest
Complexity (Number of tester measurements)	$O(kn^2)$	$O(km^2)$	$O(m^2)$
Tester Measurement Points	Each compensatory temperature	Each compensatory temperature	Room temperature

2) 1nm variation is L_{eff} for both 65nm and 45nm technologies.

Our goal is to determine the final body bias voltages using the algorithms described in the previous section. These voltages can then be written into the look-up table, which in our case consists of three rows and three columns as shown in Table III.

TABLE III

STRUCTURE OF OUR LOOK-UP TABLE

T	v_{bn}	v_{bp}
$35°C$	...	...
$50°C$	...	...
$65°C$	...	...

Based on the values in Table II, the performance spread for the benchmarks is computed. Simulations are performed at the following nine different operating points, represented as ordered pairs $(\mathbf{P}, T)$, where $\mathbf{P}$ represents the process corner, and T, the operating temperature in $°C$: (Nominal, 35), (Nominal, 50), (Nominal, 65), (Fast, 35), (Fast, 50), (Fast, 65), (Slow, 35), (Slow, 50), and (Slow, 65). The delay and leakage of the benchmarks at these points are computed for the NBB (no body bias) case. The delay of the circuits is minimum at (Fast, 35) while (Slow, 65) corresponds to the slowest case. The leakage of the benchmarks is lowest at (Slow, 35) and highest at (Fast, 65). The variation in delay and leakage is computed with T = 50° C, and nominal process corner as the mean value. The benchmarks show an average of ±13% variation in delay and 0.52X to 1.87X variation in leakage for 65nm technology, and ±12% variation in delay and 0.48X to 2.67X variation in leakage for 45nm technology. Note that the variations are expectedly larger for the 45nm technology, as compared with the 65nm technology. Such a widespread range of variations calls for post-silicon tuning through ABB.

As it will be seen from the results in Table V, ABB is capable of meeting the delay requirement for each of these cases. The optimal solutions for the extreme cases ((Slow, 65) and (Fast, 35)) both lie within the limits of permissible body bias voltages. This ensures that our region of operation is well defined, providing means for optimization, and thereby guarantees feasible solution at all simulation points. Thus, ABB can recover up to 13% variations in delay for 65nm technology, and up to 16% variations in delay for 45nm technology.

For each of the benchmark circuits, the optimal solution (v_{bn}, v_{bp}) that meets the delay requirement and minimizes the leakage at the given process and temperature corner, is first determined using the enumeration algorithm (Algorithm 1) from Section III-A, with v_{step} = 0.05V. This represents the globally optimal solution, which we call the "golden" solution. In order to determine the coefficients of delay and leakage in Equations (1) and (2) for the PTABB algorithm (Algorithm 2), the delay and leakage values are measured at nine different points, such that $v_{bn} = [-0.4, 0, 0.4]$ and $v_{bp} = [-0.4, 0, 0.4]$, respectively. The coefficients are determined by performing second degree polynomial interpolation. The NLPP is solved in Matlab [26], and the final values are snapped using the heuristic presented in Section III-B.1.

In order to determine the body bias voltages using the PABB-TABB algorithm, the process compensating values are first determined by using the NLPP formulation as outlined in Equation (3). The delay and leakage values are measured at nine well-spaced points as indicated above, at the nominal temperature, for the given process corner. The NLPP is solved to obtain (v_{bn_P}, v_{bp_P}). Similarly, the delay and leakage values are measured at the nominal process corner, at each temperature, and the NLPP is solved to determine the bias pair (v_{bn_T}, v_{bp_T}). The values are then added using Equation (13), and a legalization procedure (Algorithm 3) is called if either of the voltages is $>$ 0.4V or $<$ –0.4V. The bias values are then snapped using the heuristic in Section III-B.1.

Ten different benchmarks of varying sizes are thus simulated and the optimal body bias values are computed. The average v_{bn} and v_{bp} values returned by each of the algorithms is tabulated in Table IV for both 65nm and 45nm technologies. It can be seen that for most cases, the average values returned by these algorithms closely match the golden solutions returned by enumeration. Over the range of operating temperatures considered, the benchmark circuits show negative temperature dependence. Hence, with increasing temperature, the amount of body bias required to compensate for temperature variations, at a given process corner, increases with temperature, as can be seen from Table IV.

The complete set of results for the largest benchmark C6288 is shown in Table V for both 65nm and 45nm technologies. The data in the rows titled **Nominal** represents the delay and the leakage at the ideal temperature and process conditions, and is hence the same across all columns for a given technology. The entries in the rows titled **NBB** indicate the delay and the leakage at the given operating corner for the zero body bias case. Either the delay or the leakage is greater than its corresponding nominal value, implying that there is a need for compensation to ensure optimal performance. The rows titled **Enumeration** tabulate the delay, leakage, v_{bn} and v_{bp} returned by the enumeration algorithm. Each of these values represents

TABLE IV

AVERAGE v_{bn} AND v_{bp} VALUES (IN (V)) FOR ISCAS85 BENCHMARKS

		65nm technology								45nm technology							
		Nominal		Fast			Slow			Nominal		Fast			Slow		
Algorithm	T	35	65	35	50	65	35	50	65	35	65	35	50	65	35	50	65
Enumer-	v_{bn}	-0.28	0.02	-0.39	-0.28	-0.16	-0.20	-0.02	0.38	-0.31	0.06	-0.40	-0.39	-0.31	-0.18	0.10	0.39
ation	v_{bp}	-0.11	0.32	-0.24	-0.06	0.16	0.07	0.28	0.40	-0.07	0.28	-0.35	-0.05	0.20	0.10	0.25	0.37
PTABB	v_{bn}	-0.27	0.07	-0.40	-0.27	-0.13	-0.18	0.05	0.31	-0.30	0.09	-0.40	-0.39	-0.24	-0.23	0.07	0.37
	v_{bp}	-0.11	0.27	-0.23	-0.04	0.15	0.09	0.22	0.40	-0.08	0.29	-0.32	-0.04	0.14	0.15	0.26	0.40
PTABB	v_{bn}	-0.26	0.10	-0.40	-0.26	-0.10	-0.17	0.09	0.34	-0.28	0.11	-0.40	-0.40	-0.23	-0.22	0.09	0.39
Snapped	v_{bp}	-0.10	0.27	-0.21	-0.04	0.15	0.10	0.21	0.40	-0.08	0.30	-0.30	-0.02	0.14	0.14	0.26	0.40
PABB-TABB	v_{bn}	-0.38	-0.05	-0.40	-0.39	-0.21	-0.21	-0.07	0.23	-0.39	-0.03	-0.40	-0.40	-0.31	-0.23	-0.08	0.34
	v_{bp}	-0.03	0.36	-0.28	0.03	0.23	0.10	0.30	0.40	-0.01	0.40	-0.39	-0.04	0.26	0.18	0.38	0.40
PABB-TABB	v_{bn}	-0.37	-0.04	-0.38	-0.38	-0.20	-0.21	-0.07	0.25	-0.39	0.00	-0.40	-0.37	-0.31	-0.23	-0.07	0.36
Snapped	v_{bp}	-0.03	0.36	-0.27	0.04	0.25	0.12	0.33	0.40	0.01	0.39	-0.35	-0.04	0.28	0.20	0.39	0.40

TABLE V

SIMULATION RESULTS FOR C6288: T IN $(^\circ C)$, D IN (ps), L IN (μW), v_{bn} IN (V), v_{bp} IN (V)

		65nm technology								45nm technology							
		Nominal		Fast			Slow			Nominal		Fast			Slow		
	T	35	65	35	50	65	35	50	65	35	65	35	50	65	35	50	65
Nominal	D^*	4080	4080	4080	4080	4080	4080	4080	4080	4126	4126	4126	4126	4126	4126	4126	4126
	L^*	25.77	25.77	25.77	25.77	25.77	25.77	25.77	25.77	28.98	28.98	28.98	28.98	28.98	28.98	28.98	28.98
NBB	D	3822	4361	3957	3879	4133	4054	4333	4648	3842	4480	3544	3781	4041	4144	4484	4805
	L	20.18	32.42	30.40	38.62	48.25	13.35	16.99	21.41	22.96	35.98	50.72	62.88	76.68	13.93	17.67	22.10
Enumer	D	4070	4074	4063	4059	4067	4065	4079	4072	4094	4118	4116	4089	4098	4099	4110	4104
-ation	L	7.64	54.95	6.83	16.34	40.52	9.49	24.68	120.07	9.17	59.64	8.94	20.33	47.02	10.45	32.39	142.33
	v_{bn}	-0.30	-0.05	-0.40	-0.35	-0.15	-0.15	-0.05	0.35	-0.30	0.05	-0.40	-0.35	-0.30	-0.15	0.10	0.40
	v_{bp}	-0.10	0.35	-0.25	0.00	0.15	0.05	0.30	0.40	-0.05	0.30	-0.35	-0.05	0.20	0.10	0.25	0.40
PTABB	D	4080	4080	4080	4080	4080	4080	4080	4080	4126	4126	4126	4126	4126	4126	4126	4126
	L	7.87	62.63	7.01	16.96	39.95	8.61	26.88	121.81	8.77	76.54	9.04	19.45	52.65	10.20	35.30	140.06
	v_{bn}	-0.26	0.07	-0.40	-0.27	-0.14	-0.18	0.04	0.28	-0.27	0.12	-0.40	-0.39	-0.21	-0.21	0.10	0.39
	v_{bp}	-0.11	0.27	-0.22	-0.05	0.14	0.10	0.22	0.40	-0.09	0.29	-0.32	-0.03	0.13	0.15	0.25	0.40
PTABB	D	4051	4075	4049	4039	4089	4011	4074	4111	4098	4038	4078	4084	4198	4140	4050	4104
	L	8.34	62.09	7.47	16.71	40.87	10.55	28.25	88.60	9.21	78.79	9.45	21.01	46.60	9.53	37.79	142.33
Snapped	v_{bn}	-0.25	0.10	-0.40	-0.25	-0.10	-0.15	0.05	0.30	-0.25	0.15	-0.40	-0.40	-0.20	-0.20	0.15	0.40
	v_{bp}	-0.10	0.25	-0.20	-0.05	0.15	0.10	0.25	0.40	-0.10	0.30	-0.30	0.00	0.15	0.15	0.25	0.40
PABB	D	4080	4080	4080	4080	4080	4080	4080	4080	4126	4126	4126	4126	4126	4126	4126	4126
	L	7.55	60.49	5.91	15.87	45.64	9.81	28.41	102.70	8.77	64.14	8.30	18.22	56.60	12.59	37.59	127.56
-TABB	v_{bn}	-0.37	-0.05	-0.40	-0.39	-0.21	-0.20	-0.06	0.22	-0.39	0.01	-0.40	-0.40	-0.28	-0.17	-0.01	0.36
	v_{bp}	-0.03	0.35	-0.29	0.03	0.22	0.10	0.29	0.40	0.02	0.40	-0.38	-0.02	0.25	0.16	0.36	0.40
PABB	D	4125	4076	4063	4111	4027	4066	4079	4141	4085	4092	4116	4084	4086	4116	4205	4104
	L	8.02	60.45	6.83	18.28	47.30	9.76	24.69	76.76	10.51	67.29	8.94	21.01	56.05	10.62	29.95	142.33
-TABB	v_{bn}	-0.35	0.00	-0.40	-0.40	-0.20	-0.20	-0.05	0.25	-0.40	0.05	-0.40	-0.40	-0.25	-0.20	-0.05	0.40
Snapped	v_{bp}	-0.05	0.35	-0.25	0.05	0.25	0.10	0.30	0.40	0.05	0.35	-0.35	0.00	0.25	0.20	0.40	0.40

the "golden" solution, i.e., the body bias pair, when applied to the circuit, meets the target delay, with the lowest leakage value. The rows titled **PTABB** compute the solution using the PTABB algorithm, while the rows titled **PTABB Snapped** return the v_{bn} and v_{bp} values after the grid snapping heuristic. The solution is back-annotated to compute the corresponding delay and leakage, by performing SPICE simulations (using a timing-leakage analyzer). The rows titled **PABB-TABB** show the optimal solution obtained as a sum of the PABB and TABB bias values using (13). The values are snapped using the grid snapping heuristic, and the results are shown in the rows titled **PABB-TABB Snapped**. The delay and the leakage for this case is also computed using SPICE simulations, after back-annotating the solution obtained using the PABB-TABB algorithm.

Ideally, we would expect the v_{bn} and v_{bp} values for the PTABB and the PABB-TABB algorithms, after snapping, to match with the golden results obtained by enumeration. However, in some cases, the values do not match exactly, resulting in higher, or lower body biases, and thereby causing the delay or leakage to vary from the results obtained through enumeration, as can be seen from Table V. In a few cases, the leakage returned by the PTABB and PABB-TABB snapped algorithms is less than that obtained by the enumeration algorithm. However, the delay for such cases (after back-annotating in SPICE), is higher than the target delay D^*. These are attributable to errors in the interpolated delay and leakage values computed using the expressions in Equations (1) and (2). The error in the leakage values returned by each of the schemes as opposed to the leakage returned by enumeration is calculated, and the values are averaged for the ten benchmarks, over all process and temperature corners. While PTABB shows an average of 7% mismatch in leakage numbers for both 65nm and 45nm technology, PABB-TABB shows 12% mismatch for 65nm technology and 14% mismatch for 45nm technology. Nevertheless, in most cases, the values returned by PTABB and PABB-TABB algorithm are such that their delay and leakage values are only slightly higher or lower

than the globally optimal solution returned by the enumeration algorithm, and hence these solutions may be considered as locally optimal. As an example, if enumeration returns a value (v_{bn}, v_{bp}), then PTABB/PABB-TABB algorithms after snapping might return a value ($v_{bn} + 0.05V$, $v_{bp} - 0.05V$), whose delay and leakage values are almost identical with that of the enumeration solution.

In order to evaluate the accuracy of the two algorithms, the (v_{bn}, v_{bp}) values obtained using these algorithms after snapping are back-annotated to measure the delay and the leakage values of the respective benchmarks using our timing-leakage analyzer (built using a SPICE based library). The error in the delay values between this grid-snapped solution and the globally optimal solution computed using the enumeration algorithm is calculated for the benchmarks at all simulation points. The results are shown in Table VI for both 65nm and 45nm technologies. We have used the error in delays as a metric to determine the accuracy of the algorithms, since an inaccurate estimate of the body bias values reflects as an inaccurate measure of the delay of the circuit.

TABLE VI
ERROR IN DELAY VALUES RETURNED BY PTABB SNAPPING AND PABB-TABB SNAPPING ALGORITHMS

	No. of points in each bin			
	65nm		45nm	
% Error	PTABB	PABB-TABB	PTABB	PABB-TABB
>-2	0	0	0	0
[-2,-1.5)	1	6	5	2
[-1.5,-1)	2	4	12	6
[-1,-0.5)	8	7	13	10
[-0.5,0)	14	11	10	15
0	12	8	7	7
(0,0.5]	15	9	9	7
(0.5,1]	18	18	9	7
(1,1.5]	8	13	3	8
(1.5,2]	2	4	5	9
>2	0	0	7	9
Total	80	80	80	80

The results from Table VI show that the most of the solutions fall within 2% of the desired target delay D^*, thereby showing that the values computed by the two algorithms, when back-annotated, return "almost" optimal solutions. It can be observed that the results indicate a better match for 65nm technology as opposed to 45nm technology, since the impact of process variations increases with technology scaling. Further, the absolute error in the delay and leakage values computed through SPICE back-annotated simulations as against the enumeration results is calculated. The results indicate that the bias values through PTABB Snapping lead to an average 0.68% variation in delay and 6.51% variation in leakage for 65nm technology, and 0.92% and 8.89% variations in delay and leakage, respectively, for 45nm technology. Similarly, PABB-TABB Snapping leads to an average 0.89% error in delay values and 12.45% error in leakage values for 65nm technology, and 0.93% and 11.20% errors in delay and leakage values, respectively, for 45nm technology.

A comparison of the run-times for each of the algorithms, computed over all benchmarks, is provided in Table VII. The run-time is computed as the number of tester measurements required for the algorithm to obtain the optimal solution, for a given WID-variational region. While the worst case run-time of the enumeration scheme is of order $O(kn^2)$, the average run-time from our simulations, computed across all benchmarks, over the eight different process-temperature corners, is reported in the table. Each of these eight cases requires body-bias compensation of a different nature, and hence, the run-time of the enumeration algorithm varies in each case. The run-time for PABB-TABB Snapped includes the three measurements per compensatory temperature, required for the grid-snapping heuristic, and is hence given by $m^2 + 3k$. However, for the PTABB Snapped case, the delay-leakage model in Equations (1) and (2) can itself be used for computing the snapped values, and hence the run-time is simply km^2. Although the error in the delay and leakage values computed using the solutions returned by the PABB-TABB algorithm after snapping, is higher than that for the PTABB algorithm, its run-time is the smallest among the three methods, thereby providing a reasonable accuracy/run-time trade-off. Further, if the number of temperature points chosen to compensate for thermal variations is higher than three (in our case), and if the overhead in testing at each temperature is considered in the run-time analysis, the trade-off may be more economically viable.

TABLE VII
RUN TIME FOR ABB ALGORITHMS (m = 3, n = 17, k = 3)

Algorithm	65nm	45nm
Enumeration	453	468
PTABB Snapped	27	27
PABB-TABB Snapped	18	18

V. CONCLUSION

While the effects of process and temperature variations in the sub-90nm technologies continue to significantly thwart the yield of the fabrication process, post-silicon tuning methods have evolved to tighten the distribution of the delay and the leakage of these chips. Adaptive Body Bias (ABB) provides a viable tuning mechanism to ensure optimal performance or leakage savings as desired. While the implementation of the ABB control system can either be achieved using a critical path replica or with look-up table based methods, the look-up table method calls for optimization to reduce the amount of time spent on the tester. Two different algorithms, namely the PTABB algorithm and the PABB-TABB algorithm are proposed to provide reasonable accuracy/run-time trade offs as against a simple enumeration scheme to solve the problem of optimal body-bias voltage selection. The results, obtained through thorough simulations over a wide range of data demonstrate the ability of ABB to meet the performance constraints, and also show the accuracy of our schemes over 65nm and 45nm PTM technologies. Accuracy and tester time trade-offs for the algorithms developed by us are discussed, and an implementation overview is also provided.

REFERENCES

[1] D. Boning and S. Nassif, *Design of High Performance Microprocessor Circuits*, ch. Models of Process Variations in Device and Interconnect, pp. 98–115. IEEE Press, 2000.

[2] 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 Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, pp. 888–899, October 2003.

[3] M. Miyazaki, G. Ono, and T. Kawahara, "Optimum Threshold-Voltage Tuning for Low-Power High-Performance Microprocessor," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 17–20, May 2005.

[4] L. Yan, J. Luo, and N. K. Jha, "Joint Dynamic Voltage Scaling and Adaptive Body Biasing for Heterogeneous Distributed Real-Time Embedded Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 1030–1041, July 2005.

[5] L. Yan, J. Luo, and N. K. Jha, "Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Heterogeneous Distributed Real-Time Embedded Systems," in *Proceedings of the IEEE International Conference on Computer Aided Design*, pp. 30–37, November 2003.

[6] S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Lower Power Microprocessors under Dynamic Workloads," in *Proceedings of the IEEE International Conference on Computer Aided Design*, pp. 721–725, November 2002.

[7] S. Yajuan, W. Zuodong, and W. Shaojun, "Energy-aware Supply and Body Biasing Voltage Scheduling Algorithm," in *Proceedings of the International Conference on Solid State and Integrated Circuits Technology*, pp. 1956–1959, October 2004.

[8] A. Andrei, M. Schmitz, P. Eles, Z. Peng, and B. M. Al-Hashimi, "Overhead-Conscious Voltage Selection for Dynamic and Leakage Energy Reduction of Time-Constrained Systems," in *Proceedings of Design, Automation and Test in Europe*, pp. 518–523, February 2004.

[9] C. H. Wann, H. Chenming, K. Noda, D. Sinitsky, F. Assaderaghi, and J. Bokor, "Channel Doping Engineering of MOSFET with Adaptable Threshold Voltage using Body Effect for Low Voltage and Low Power Applications," in *Proceedings of the International Symposium of VLSI Technology*, pp. 159–163, June 1995.

[10] T. Kuroda, T. Fujita, S. Mita, T. Nagamatu, S. Yoshioka, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9 V 150 MHz 10 mW 2-D Discrete Cosine Transform Core Processor with Variable-Threshold-Voltage Scheme," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 166–167, August 1996.

[11] J. W. Tschanz, J. Kao, S. G. 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 Journal of Solid-State Circuits*, vol. 37, pp. 1396–1402, November 2002.

[12] J. Tschanz, S. Narendra, A. Keshavarazi, and V. De, "Adaptive Circuit Techniques to Minimize Variation Impacts on Microprocessor Performance and Power," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 9–12, May 2005.

[13] J. W. Tschanz, S. G. Narendra, Y. Ye, B. A. Bloechel, S. Borkar, and V. De, "Dynamic Sleep Transistor and Body Bias for Active Leakage Power Control of Microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 1838–1845, November 2003.

[14] J. W. Tschanz, S. G. 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 Journal of Solid-State Circuits*, vol. 38, pp. 26–829, May 2003.

[15] S. Narendra, A. Keshavarzi, B. A. Bloechel, S. Borkar, and V. De, "Forward Body Bias for Microprocessors in 130-nm Technology Generation and Beyond," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 696–701, May 2003.

[16] G. Ono, M. Miyazaki, H. Tanaka, N. Ohkubo, and T. Kawahara, "Temperature Referenced Supply Voltage and Forward-Body-Bias Control (TSFC) Architecture for Minimum Power Consumption," in *Proceedings of the European Solid State Circuits Conference*, pp. 391–394, September 2004.

[17] Device Group at Arizona State University, "Predictive Technology Model." Available at `http://www-eas.asu.edu/ptm`.

[18] H. Ananthan, C. H. Kim, and K. Roy, "Larger-than-Vdd Forward Body Bias in sub-0.5V Nanoscale CMOS," in *Proceedings of the IEEE International Symposium on Low Power Electronic Design*, pp. 8–13, October 2004.

[19] V. Gerousis, "Design and Modeling Challenges for 90nm and 50nm," in *Proceedings of the IEEE Custom Integrated Circuit Conference*, pp. 353–360, Septemer 2003.

[20] J. W. Tschanz, N. S. Kim, S. Dighe, J. Howard, G. Ruhl, S. Vangal, S. Narendra, Y. Hoskote, H. Wilson, C. Lam, M. Shuman, C. Tokunaga, D. Somasekhar, S. Tang, D. Flnan, T. Karnik, N. Borkar, N. Kurd, and V. De, "Adaptive Frequency and Biasing Techniques for Tolerance to Dynamic Temperature-Voltage Variations and Aging," in *Proceedings of the IEEE Internation Solid-State Circuits Conference*, pp. 292–294, February 2007.

[21] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*. New York, USA: Cambridge University Press, 1999.

[22] V. Gerousis, "Design and Modeling Challenges for 90nm and 50nm," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 353–360, 2003.

[23] A. Dasdan and I. Hom, "Handling Inverted Temperature Dependence in Static Ti ming Analysis," *ACM Transactions on Design and Automation of Electronic Systems*, vol. 11, pp. 306–324, April 2006.

[24] K. Kanda, K. Nose, H. Kawaguchi, and T. Sakurai, "Design Impact of Positive Temperature Dependence on D rain Current in sub-1V CMOS VLSIs," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 1559–1564, 1999.

[25] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis," Tech. Rep. UCB/ERL M92/41, University of California, Berkeley, 1992. Available at `http://www-cad.eecs.berkeley.edu/research/sis`.

[26] Matlab Reference Manual. Available at `http://www.mathworks.com/ access/helpdesk/help/techdoc/matlab.shtml`.

VI. APPENDIX

In this section, we provide a proof to Equation (13) that is used to compute the PTABB body bias voltages as a function of the individual PABB and TABB voltages.

Theorem 1 *The optimal body bias voltages for process and temperature compensation can be computed as the sum of the voltages obtained by compensating for process and temperature variations independently of each other, i.e.,*

$$\begin{aligned} v_{bn_{PT}} &= v_{bn_P} + v_{bn_T} \\ v_{bp_{PT}} &= v_{bp_P} + v_{bp_T} \end{aligned} \tag{16}$$

We first prove the above theorem by showing that the body bias that meets the delay for the PTABB case can be expressed as the sum of the body biases that meet the delays for the PABB and the TABB cases, respectively.

Proof: Neglecting the effect of second order terms in Equation (1), i.e., using a first order Taylor series approximation, we can re-write the expression as,

$$D(v_{bn}, v_{bp}) = D_0(1 + av_{bn})(1 + bv_{bp}) \tag{17}$$

For the PTABB case, we can write,

$$D^* = D_{PT}(1 + av_{bn_{PT}})(1 + bv_{bp_{PT}}) \tag{18}$$

where D_{PT} is the delay without any body bias, and D^* is the target delay (same as $D(\mathbf{P_0}, T_0)$), while $(v_{bn_{PT}}, v_{bp_{PT}})$ is the final solution. Similarly, the delays for the PABB and the TABB cases can be represented as:

$$\begin{aligned} D^* &= D(\mathbf{P_1}, T_0)(1 + av_{bn_P})(1 + bv_{bp_P}) \\ D^* &= D(\mathbf{P_0}, T_1)(1 + av_{bn_T})(1 + bv_{bp_T}) \end{aligned} \tag{19}$$

Note that simulation results have shown that the coefficients of delay for the PABB, TABB, and PTABB cases are almost similar and hence we use the same constants a and b. Re-arranging the terms in (18) and (19), we have

$$\begin{aligned} D(\mathbf{P_1}, T_1) - D(\mathbf{P_0}, T_0) &= D^* \left(\frac{1}{1 + av_{bn_{PT}}} \frac{1}{1 + bv_{bp_{PT}}} - 1 \right) \\ D(\mathbf{P_1}, T_0) - D(\mathbf{P_0}, T_0) &= D^* \left(\frac{1}{1 + av_{bn_P}} \frac{1}{1 + bv_{bp_P}} - 1 \right) \\ D(\mathbf{P_0}, T_1) - D(\mathbf{P_0}, T_0) &= D^* \left(\frac{1}{1 + av_{bn_T}} \frac{1}{1 + bv_{bp_T}} - 1 \right) \end{aligned}$$

Using binomial expansion for the fractional expressions, and neglecting higher order terms, the above equations can be simplified as:

$$\begin{aligned} D(\mathbf{P_1}, T_1) - D(\mathbf{P_0}, T_0) &= D^* \left(\left(1 - a v_{bn_{PT}}\right) \left(1 - b v_{bp_{PT}}\right) - 1 \right) \\ D(\mathbf{P_1}, T_0) - D(\mathbf{P_0}, T_0) &= D^* \left(\left(1 - a v_{bn_P}\right) \left(1 - b v_{bp_P}\right) - 1 \right) \\ D(\mathbf{P_0}, T_1) - D(\mathbf{P_0}, T_0) &= D^* \left(\left(1 - a v_{bn_T}\right) \left(1 - b v_{bp_T}\right) - 1 \right) \end{aligned}$$

Substituting the above terms in (12), we have

$$\begin{aligned} D^*(a v_{bn_{PT}} + b v_{bp_{PT}} - ab v_{bn_{PT}} v_{bp_{PT}}) &= \\ D^*(a v_{bn_T} + b v_{bp_T} - ab v_{bn_T} v_{bp_T}) &+ \\ D^*(a v_{bn_P} + b v_{bp_P} - ab v_{vbn_P} v_{bp_P}) & \end{aligned} \tag{20}$$

Neglecting the quadratic terms involving the product of v_{bn} and v_{bp}, since v_{bn} and v_{bp} are both $\ll 1$, we have:

$$\begin{aligned} a v_{bn_{PT}} + b v_{bp_{PT}} &\approx a(v_{bn_T} + v_{bn_P}) + \\ & \quad b(v_{bp_T} + v_{bp_P}) \end{aligned} \tag{21}$$

Hence Equation (16) is proved. ■

We now prove that the body bias voltage pair that minimizes the leakage of the circuit, under the delay constraint also satisfies the above equation. The proof is as follows:

Proof: As stated in the previous part of the proof, the delay of the circuit as a function of v_{bn} and v_{bp} can be written as:

$$D(v_{bn}, v_{bp}) = D_0(1 + a v_{bn})(1 + b v_{bp}) \tag{22}$$

Neglecting the second order effects of the quadratic term obtained by the product of v_{bn} and v_{bp}, we can write the above equation as:

$$D = D_0(1 + a v_{bn} + b v_{bp}) \tag{23}$$

Thus, we can express v_{bp} in terms of v_{bn} as:

$$v_{bp} = \frac{D - D_0(1 + a v_{bn})}{b D_0} \tag{24}$$

Similarly, neglecting the second order effects in Equation (2), i.e., using a Taylor series expansion, the leakage of the circuit can be written as:

$$\begin{aligned} L(v_{bn}, v_{bp}) &= \frac{1}{e} L_0 e^{(1 + c v_{bn})(1 + d v_{bp})} \\ lnL &= lnL_0 - 1 + (1 + c v_{bn})(1 + d v_{bp}) \end{aligned} \tag{25}$$

Expressing v_{bp} in terms of v_{bn} using (24), we have

$$lnL = lnL_0 - 1 + (1 + c v_{bn}) \left(1 + d \left(\frac{D - D_0(1 + a v_{bn}}{b D_0}\right)\right) \tag{26}$$

Since the final solution minimizes the leakage, we can solve for v_{bn} by differentiating the above equation with respect to v_{bn} and setting the RHS to zero. Thus, we have

$$\frac{d}{d v_{bn}}(1 + c v_{bn}) \left(1 + d \left(\frac{D - D_0(1 + a v_{bn}}{b D_0}\right)\right) = 0 \tag{27}$$

Simplifying, we get:

$$2acdD_0 v_{bn} = D_0(bc - ad) + cd(D - D_0) \tag{28}$$

Substituting $D = D(\mathbf{P_0}, T_0) = D^*$, $D_0 = D_{PT}$, and $v_{bn} = v_{bn_{PT}}$ in the above equation, we have:

$$2acdD_{PT} v_{bn_{PT}} = D_{PT}(bc - ad) - cd(\Delta D_{PT}) \tag{29}$$

Similarly, for the PABB and the TABB cases, we can write

$$\begin{aligned} 2acdD_P v_{bn_P} &= D_P(bc - ad) - cd(\Delta D_P) & (30) \\ 2acdD_T v_{bn_T} &= D_T(bc - ad) - cd(\Delta D_T) & (31) \end{aligned}$$

Adding the above two equations, we have

$$2acd(D_P v_{bn_P} + D_T v_{bn_T}) = (D_P + D_T)(bc - ad) - cd(\Delta D_P + \Delta D_T) \tag{32}$$

Subtracting (32) from (29), and using Equation (12), we have

$$2acd\left(D_{PT} v_{bn_{PT}} - D_P v_{bn_P} - D_T v_{bn_T}\right) = (D_{PT} - D_P - D_T)(bc - ad)$$

Since $D_{PT} = D^* + \Delta D_{PT}$, $D_P = D^* + \Delta D_P$, and $D_T = D^* + \Delta D_T$, using $\Delta D_{PT} = \Delta D_P + \Delta D_T$ from Equation (12), we can write

$$2acd\left(D_{PT} v_{bn_{PT}} - D_P v_{bn_P} - D_T v_{bn_T}\right) = -D^*(bc - ad)$$

Simulation results have shown that using $v_{bn_{PT}} = v_{bn_P} + v_{bn_T}$ in the LHS of the above equation closely matches the value of the RHS, for various process and temperature corners. Hence, we conclude that $v_{bn_{PT}}$ can be determined using Equation (13). Similarly, it can be shown that $v_{bp_{PT}} = v_{bp_P} + v_{bp_T}$. Thus, the optimal bias pair that meets the delay requirement and minimizes the circuit leakage can be computed using Equation (13). ■