Mathematically Assisted Adaptive Body Bias (ABB) for Temperature Compensation in Gigascale LSI Systems

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*— Process variations and temperature variations can cause both the frequency and the leakage of the chip to vary significantly from their expected values, thereby decreasing the yield. Adaptive Body Bias (ABB) can be used to pull back the chip to the nominal operational region. We propose the use of this technique to counter temperature variations along with process variations. We present a CAD perspective for achieving process and temperature compensation using bidirectional ABB. Mathematical models are used to determine the exact amount of body bias required to optimize the delay and leakage, and an algorithmic flow that can be adopted for gigascale LSI systems is provided.**

***Index Terms* : Delay, Leakage, Adaptive Body Bias (ABB), Process Variations, Temperature Variations, Nonlinear Programming Problem (NLPP), Enumeration**

I. INTRODUCTION

With technology scaling, the effects of process parameter variations and on-chip temperature variations have caused the delay and leakage of modern-day processors to vary significantly from their desired values. Some of the dies may satisfy the delay constraint but leak too much, while others may leak nominally but fail to meet the target frequency. Thus, a significant fraction of the total number of acceptable dies may fail to achieve the performance goals. This has led to the evolution of methodologies to perform post-silicon tuning for yield improvement. Adaptive Body Bias (ABB) provides a viable control technique that can counter the effects of on-chip variations.

Two of the significant contributors to on-chip variability arise from changes in process parameters and in the operating temperature. Process variations lead to fluctuations in parameters such as transistor channel lengths, oxide thicknesses, and dopant concentrations. These cause variations in the delay and leakage of the circuit, thereby affecting performance. On-chip temperature variations, on the other hand, change the mobilities of electrons and holes. An increase in the operating temperature causes the mobilities to decrease, thereby decreasing the I_{on} current, which, in turn, reduces 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 [1].

Adaptive Body Bias (ABB) is a dynamic technique that helps tighten the distribution of the *maximum operational frequency* and the *maximum leakage power* in the presence of WID variations, and thereby helps improve the yield significantly. It was first proposed by Wann *et al.* in [2] and was further explored by Kuroda [3] during the design of a DSP Processor. Bidirectional Adaptive Body Bias has been shown to reduce the impact of D2D and WID parameter variations on microprocessor frequency and leakage in [4], [5], [6] and [1]. Typically, devices that are slow but do not leak too much can be Forward Body Biased (FBB) to improve the speed, whereas devices are fast and leaky can be Reverse Body Biased (RBB) to meet the leakage budget. The work in [4], [7] performs process variation-based ABB, and divides the die into a set of WID variational regions. In each region, test structures, which are replicas of the critical path, are built. The delay and leakage of these test structures are measured, and 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 [4], shows that 100% of the dies can be salvaged, while 99% of them operate at the highest frequency bin.

Traditionally, ABB has been used only to compensate for process variations [4–6]. However, on-chip temperature changes can also significantly vary the delay and leakage of nanometer-scale devices, thereby necessitating the need to mitigate the effects of these thermal variations as well. Only a limited amount of work so far has addressed this problem, such as [8], which focuses purely on temperature effects. In this work, we apply a combination of temperature-based ABB (TABB) and process-based ABB (PABB) 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. We propose a general architecture and an implementation scheme to achieve this.

The contribution of this paper is to provide a strategy for determining the exact amount of bias required to achieve process and temperature compensation through a combination of simulation, probabilistic design and post-silicon tuning in order to maximize the yield subject to frequency and leakage constraints. This method is aptly termed PTABB (process and temperature-based adaptive body bias). The final set of PTABB voltages that can counter process and temperature variations at all operating conditions is thus a combination of PABB and TABB. We propose two methods to compute the TABB values, namely, an enumeration based method and a mathematical model based method. Enumeration based TABB involves simulating the circuit at discrete points in the solution space and finding the best solution. In contrast, mathematically assisted TABB assumes a continuous search space and provides an exact solution using a model that captures the effect of body bias on delay and leakage and a simple nonlinear programming problem (NLPP) formulation. PABB can be performed by building test structures with critical path replicas on each WID-variational region [4]. The exact amount of body bias to counter the effects of process variations at room temperature is determined by measuring the delay and leakage of the circuit, and choosing the optimal solution.

The concept of using mathematical models to formulate expressions for delay and leakage, and thereby to obtain exact solutions for the ABB voltages, is in itself a new and attractive approach. Compared to prior approaches that determine the exact body bias required during run-time by monitoring the delay and leakage (listed in [9]), our scheme uses a simple look-up table (similar in concept to that used in [8]), that stores these pre-computed values, and hence, only requires a temperature sensor to monitor the variations in on-chip temperature. This eliminates the need for circuits like leakage current monitor, substrate charge injector, self substrate bias, *etc.*, since the determination of the TABB voltages is carried out at the design stage. Further, the idea of *one-time compensation* for process variations and *run-time compensation* for temperature variations is effectively combined. The generation of these additional body bias voltages and their distribution on chip is not considered to be within the scope of our work. We present the algorithm, implementation and results of this novel scheme in the subsequent sections.

II. CENTRAL IDEA

In this section, we present an overall picture of the proposed implementation. The die is partitioned into several WID variational regions, and each of these regions is separately compensated. Our target technology in this work uses a triple well process although the idea can be generalized to any other process. The central body bias generator consists of a PVT invariant voltage reference source, and is capable of generating one pair of voltages applicable to the NWELL and PWELL of each WID variational region separately. (Alternatively, the body bias generators can be replicated in each WID zone to locally generate and distribute the required voltages, but a central PVT invariant voltage reference is still required). A temperature sensor is placed in each of these regions in order to detect on-chip temperature variations. A small ROM is fabricated in every region, whose look up table consists of a pair of voltages (v_{bn},v_{bp}) for each temperature being compensated.

The TABB values $(v_{bn}, v_{bp})_{TABB}$, that can compensate for temperature variations only, assuming ideal process conditions are determined during the design stage. Similar values $(v_{bn}, v_{bp})_{PABB}$, to compensate for process variations at room temperature, are calculated through post-silicon tuning. These values can be combined as shown in Fig. 1 to get one pair of bias values for every block (WID variational region) at each operating temperature for all the dies. The final bias pair, denoted by $(v_{bn}, v_{bp})_{PTABB}$, is programmed into the ROM.

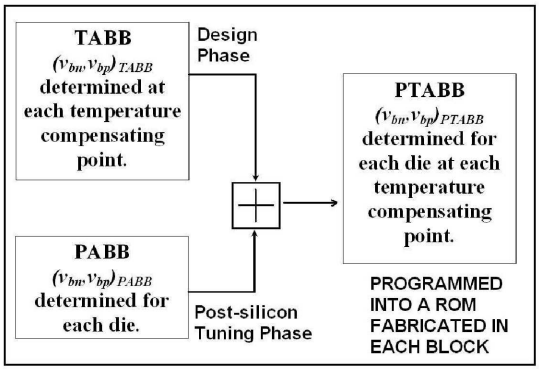

Fig. 1. **Generation of PTABB values for every block (WID variational region).**

When the circuit is in operation, the temperature sensor detects changes in the on-chip temperature. The corresponding values of v_{bn} and v_{bp} are read from the ROM and fed as inputs to the central bias generator. These voltages are generated by the central bias generator and distributed to NWELL and PWELL through the bias distribution network. The overall architecture is shown in Fig. 2.

III. PTABB ALGORITHM

In this section, we present the algorithm that determines the body bias required for process and temperature variation compensation. Since we assume the existence of a triple well process, the bodies of both NMOS and PMOS devices can be biased independently. However, the algorithm can be easily modified for a twin well process. We present SPICE-calibrated models that express the delay, and leakage in terms of the bias voltages and determine the optimal bias voltages based on operational constraints.

The effects of process and temperature variations on the delay of a combinational circuit can be represented as:

$$D = f(\mathbf{x}, T) \tag{1}$$

where D is the delay of the circuitry, $\mathbf{x}$ is a vector of process variables and T is the operating temperature of the chip. Let $\mathbf{x_0}$ and T_0 denote the values of the process and temperature variables

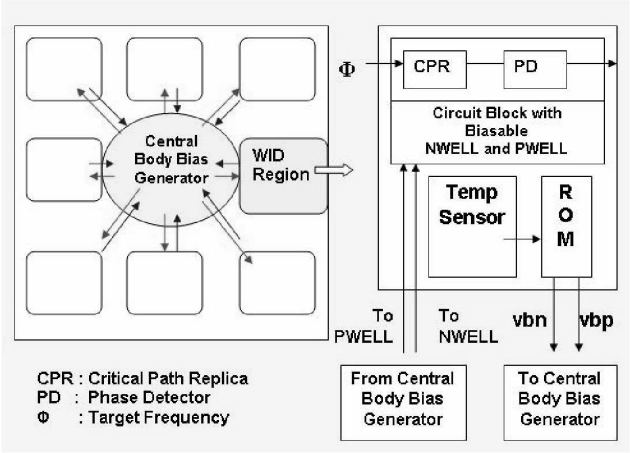

Fig. 2. **Block diagram for PTABB implementation.**

under ideal conditions where there is no variation. The increase in delay at any point $(\mathbf{x}, T)$ can be written as:

$$\Delta D = f(\mathbf{x}, T) - f(\mathbf{x_0}, T_0) \tag{2}$$

where $\mathbf{x}$ is the vector of process variables of a particular die, while T is the operating temperature of the die. If $\mathbf{x}$ and T are independent variables, the effect of simultaneously varying $\mathbf{x}$ and T, from $(\mathbf{x_0}, \mathbf{T_0})$ to $(\mathbf{x_1}, \mathbf{T_1})$ can be approximated as varying $\mathbf{x}$ and T individually from their original values and adding their effects, *i.e.*,

$$\Delta D \approx [f(\mathbf{x_0}, \mathbf{T_1}) - \mathbf{f}(\mathbf{x_0}, \mathbf{T_0})] + [\mathbf{f}(\mathbf{x_1}, \mathbf{T_0}) - \mathbf{f}(\mathbf{x_0}, \mathbf{T_0})] \tag{3}$$

where $f(\mathbf{x_0}, \mathbf{T_1})$ is the delay with temperature variations only, while $f(\mathbf{x_1}, \mathbf{T_0})$ is the delay considering the effect of process variations only.

The above assumption of independence is justified since process and temperature variations have different device level effects, and hence their impacts on the delay can be treated as independent of one another. Process variations affect parameters such as channel length, oxide thickness, and dopant concentration, thereby altering the delay, while temperature variations affect the mobilities of electrons and holes, which influences the on-current, and hence, the delay of the circuit. Further, the results shown in Table I indicate the validity of the assumption. The delay of a ring oscillator is measured through simulations performed using BPTM [10] 100nm model files at $T = 50^\circ C$ and $T = 75^\circ C$ at the two extreme process corners:

1) *Low-Vt* corner which is the case where process variations cause the threshold voltages of both NMOS and PMOS devices to decrease by 10%.
2) *High-Vt* corner which is the case where process variations cause an increase in both V_{tn0} and V_{tp0} by 10%.

The column labeled **Nom-Delay** in Table I indicates the delay at $T = 25^\circ C$ under ideal process conditions. The delay considering the effect of both process and temperature variations is shown in the column labeled **Delay**$_{PT}$ and the variation in delay calculated directly, using (2) is shown in column 7. Columns labeled **Delay**$_T$ and **Delay**$_P$ list the delay considering temperature variations and process variations respectively. The change in delay, expressed as a sum of the change in delays due to process and thermal effects using (3) is listed in column 8. It can be seen from the last column in the table that the difference in delay between the two measurements are negligible compared to the actual circuit delay values. Thus, we can indeed decompose the delay expressions into a temperature-dependent term and a process-dependent term. We use the above findings to perform temperature compensation and process compensation independently of each other.

TABLE I
DELAY DECOMPOSITION INTO PROCESS DEPENDENT AND TEMPERATURE DEPENDENT TERMS FOR A RING OSCILLATOR

Temp $^\circ C$	Process Corner	Nom-Delay $f(\mathbf{x_0}, \mathbf{T_0})$ (ps)	Delay$_{PT}$ $f(\mathbf{x_1}, \mathbf{T_1})$ (ps)	Delay$_P$ $f(\mathbf{x_1}, \mathbf{T_0})$ (ps)	Delay$_T$ $f(\mathbf{x_0}, \mathbf{T_1})$ (ps)	ΔD from Eqn (2) $f(\mathbf{x_1}, T_1) - f(\mathbf{x_0}, T_0)$ (ps)	ΔD from Eqn (3) $[f(\mathbf{x_0}, \mathbf{T_1}) - \mathbf{f}(\mathbf{x_0}, \mathbf{T_0})]+$ $[f(\mathbf{x_1}, \mathbf{T_0}) - \mathbf{f}(\mathbf{x_0}, \mathbf{T_0})]$(ps)	Difference (ps)
50	Low V_t	151.0	145.6	141.8	154.9	-5.4	-5.2	-0.2
50	High V_t	151.0	165.3	161.2	154.9	14.3	14.2	0.1
75	Low V_t	151.0	149.2	141.8	158.6	-1.8	-1.5	-0.3
75	High V_t	151.0	169.3	161.2	158.6	18.3	17.8	0.5

A. *Temperature Compensation*

Generally, the delay of a circuit exhibits negative temperature dependence, *i.e.* the delay increases with an increase in temperature due to a reduction in the mobility of electrons and holes. Hence, we need to forward body bias the devices to reduce the delay at higher operating temperatures, at the expense of leakage. However, at low-V_{dd} operations, the reduction in V_t has a higher impact than the reduction in mobility and an increase in temperature allows the circuits to operate at a higher speed. This effect, described as *positive temperature dependence*, can be used to achieve TABB as described in [11]. In such cases, the devices may be less forward biased (or relatively reverse body biased) at higher temperatures to achieve leakage savings. We hereby present two methodologies to determine the amount of FBB needed to meet the delay constraint, thereby minimizing leakage, for the general case of negative temperature dependence.

B. *Enumeration based TABB*

The task of ABB compensation is to determine the optimal value of the biases for the NWELL and PWELL, that brings the delay back to specifications, with a minimal leakage overhead. The basic idea of enumeration is to traverse through the entire search space and find this solution. 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. 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 optimal values is shown in Algorithm 1. We wish to operate the circuit at the highest possible frequency, and the target delay of the circuitry (D^*) is determined by a simulation at the nominal temperature. Since we have assumed negative temperature dependence, the delay of the circuit at a higher operating temperature is greater than D^*, hence requiring FBB. The circuit is simulated with the upper bound of the search space $(v_{bnmax}, v_{bpmax}$[1]$)$ to determine if maximum FBB can pull the circuit delay back to D^*. If the maximum applicable bias fails to meet the target delay, the operational frequency of the circuit block needs to be reduced. Otherwise, we set this as our initial solution and seek better solutions than (v_{bnmax}, v_{bpmax}) within the search space since (v_{bnmax}, v_{bpmax}) is overkill in terms of leakage. The circuit is simulated at each of the bias pair points and the solution that has the minimum leakage is chosen. If the final leakage of the block is still greater than the allocated budget, then the operational frequency is reduced, D^* thereby increased, and the process of enumeration is repeated.

Algorithm 1 Enumeration Algorithm for TABB

```
1: Simulate circuit with zero body bias at T = 25°C to obtain D*
2: Leakage budget for the circuit is denoted by L_max
3: for each temperature T being compensated do
4:    Measure best-case delay D(v_bnmax, v_bpmax)
5:    if D(v_bnmax, v_bpmax) ≥ D* then
6:       Reduce operational frequency
7:       {Maximum ABB cannot meet delay requirement; decrease target
         delay D*.}
8:    else
9:       Initial Solution = (v_bnmax, v_bpmax)
10:      L_min = ∞
11:   end if
12:   for (v_bn = v_bnmax to v_bnmin) do
13:      for (v_bp = v_bpmax to v_bpmin) do
14:         Apply (v_bn, v_bp) and simulate at temperature T
15:         if D(v_bn, v_bp) ≤ D* then
16:            {Likely solution since it meets delay.}
17:            if L(v_bn, v_bp) ≤ L_min then
18:               New solution = (v_bn, v_bp)
19:               L_min = L(v_bn, v_bp)
20:            end if
21:         else
22:            break inner for loop
23:            {Lower values of vbp do not meet delay.} [2]
24:         end if
25:         v_bp = v_bp − v_step
26:         {v_step is the minimum resolution of bias that can be applied.}
27:      end for
28:      if D(v_bn, v_bpmax) ≥ D* then
29:         break outer for loop
30:         {Lower values of vbn do not meet delay.} [2]
31:      end if
32:      v_bn = v_bn − v_step
33:   end for
34:   if L_min ≥ L_max then
35:      Reduce Operational Frequency and go to Line 5
36:   end if
37: end for
```

C. *Mathematically assisted TABB*

Enumeration over the entire two dimensional search space to determine the optimum bias ordered pair is a costly process for large circuits since it requires simulations at each bias value (worst case) and has a cost of $O(n^2)$, where n is the number of bias voltages available. Hence, we propose an efficient algorithm based on a simple nonlinear programming problem (NLPP) that requires the simulation of the circuit for delay and leakage at a few points only, to determine the exact body bias pair required. The crux of this method is as follows.

The delay and leakage of a circuitry can be altered by applying a bias voltage v_{bn} to the body of the NMOS transistors and $(V_{dd} - v_{bp})$ to that of the PMOS transistors. Since analytical expressions that can quantize the effect of body bias on delay and 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 both delay and leakage. Thus we have the expressions:

$$D(v_{bn}, v_{bp}) = D_0[\sum_{i=0}^{2}(\sum_{j=0}^{2} a_{ij} v_{bn}^j) v_{bp}^i] \quad (4)$$

$$L(v_{bn}, v_{bp}) = L_0[\sum_{i=0}^{2}(\sum_{j=0}^{2} b_{ij} v_{bn}^j) v_{bp}^i] \quad (5)$$

where D_0 and L_0 are the delay and leakage values at the given

[1]The actual voltage applied to the body of the PMOS transistors is $(V_{dd} - v_{bp})$. For simplicity, we refer to this as v_{bp}.

[2]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.

operating temperature under the condition where process variation effects are ignored. Since we have two variables v_{bn} and v_{bp}, it is desirable to model the effects of these individually, independently of each other and finally superpose their effects. In other words, we wish to re-write (4) as:

$$D(v_{bn}, v_{bp}) = D_0 f(v_{bn}) g(v_{bp}) \tag{6}$$

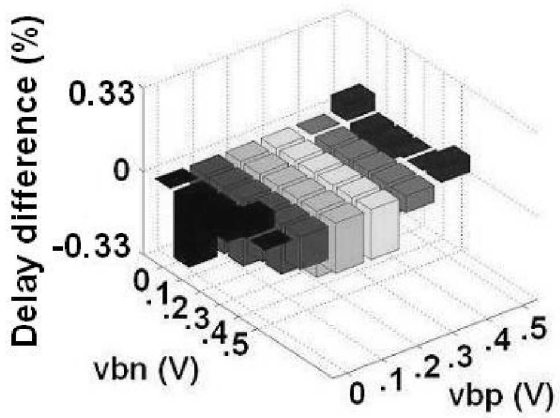

Fig. 3. **Accuracy of polynomial curve fits compared with HSPICE simulations for a ring oscillator at $T = 25°C$ and $V_{dd} = 1.0V$. Reported values are the difference in delays at each simulation point.**

We verified the possibility of this decomposition on the delay of a Ring Oscillator (RO) and the results are shown in Fig. 3. The reference delays of the RO following the application of body bias are measured through HSPICE simulations performed using BPTM 100nm model files. The delay due to varying v_{bn} only (measured at $v_{bp} = V_{dd}$) is approximated using a second order best fit curve as,

$$f(v_{bn}) = (1 + x_1 v_{bn} + x_2 v_{bn}^2) \tag{7}$$

Similarly, the delay due to varying v_{bp} only (measured at $v_{bn} = 0$) is approximated using the polynomial $g(v_{bp})$ as,

$$g(v_{bp}) = (1 + y_1 v_{bp} + y_2 v_{bp}^2) \tag{8}$$

The new delay of the ring oscillator at any point (v_{bn},v_{bp}) is calculated as a product of the polynomials $f(v_{bn})$ and $g(v_{bp})$. Finally, the difference between the reference values and the new delay values, calculated at each point, is shown in Fig. 3. It can be seen that this difference is negligible, thereby conforming the predicted trend. Hence (4) can be re-written as,

$$D(v_{bn}, v_{bp}) = D_0(1 + x_1 v_{bn} + x_2 v_{bn}^2)(1 + y_1 v_{bp} + y_2 v_{bp}^2) \tag{9}$$

However, leakage does not show a similar trend. The coefficients in (5) and (9) can be determined 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. The Nonlinear Programming Problem can be formulated as:

$$\text{minimize } L(v_{bn}, v_{bp}) = L_0[\sum_{i=0}^{2}(\sum_{j=0}^{2} b_{ij} v_{bn}^j) v_{bp}^i] \tag{10}$$

subject to

$$\begin{aligned} D(v_{bn}, v_{bp}) = D_0(1 + x_1 v_{bn} + x_2 v_{bn}^2)(1 + y_1 v_{bp} + y_2 v_{bp}^2) \le D^* \\ 0 \le v_{bn} \le v_{bn\,max} \\ 0 \le v_{bp} \le v_{bp\,max} \end{aligned} \tag{11}$$

Practically, if $D(v_{bn}, v_{bp})$ exceeds D^*, then the minimum leakage solution corresponds to the case where (11) is an equality. Therefore, the above problem can be solved by eliminating one of the variables v_{bn} or v_{bp} in (11) and finding the minimum value of L in (10) with respect to the other variable using the Newton-Raphson method to obtain the optimum solution, $(v_{bn}, v_{bp})_{TABB}$.

D. Process Compensation

In order to perform process compensation using ABB, a test structure consisting of the critical path replica is built in each of the WID variational regions. PABB is performed in [4] by applying an NMOS bias (v_{bn}) from an off-chip source and automatically adapting the PMOS bias to meet the target frequency. The process is repeated for all possible values of v_{bn} and the bias pair which results in lowest leakage is chosen as the final solution. This scheme requires a 5 bit counter and a DAC (Digital to Analog Converter) in the test structures, to automatically determine the PMOS bias for each NMOS bias applied.

This methodology can be simplified with the use of external voltage sources for biasing both the NWELL and PWELL and an NLPP formulation to determine the exact PABB values. The test structure now consists of a critical path replica and a phase detector only as shown in Fig. 2. The NLPP formulation outlined in the previous sub-section is employed to determine the exact PABB values. The coefficients in (5) and (9) are now determined by actual measurements on chip, instead of circuit simulations for the TABB case. Off-chip sources are used to bias the wells, and the delay and leakage values are measured at some points. The NLPP is formulated in an identical manner as that in (10) and (11), with D_0 and L_0 being the measured delay and leakage values of the *WID-variational region* at *nominal temperature*. The NLPP is solved to obtain the optimal bias values $(v_{bn}, v_{bp})_{PABB}$. The final value that can counter both process and temperature variations for each WID-variational region is calculated by summing the values obtained individually through PABB and TABB:

$$(v_{bn}, v_{bp})_{PTABB} = (v_{bn}, v_{bp})_{PABB} + (v_{bn}, v_{bp})_{TABB} \tag{12}$$

However if the final values are greater than $v_{bn\,max}$ or $v_{bp\,max}$, the solution must be legalized by considering the minimum of the sum of the leakage due to PABB and TABB. The NLPP must be re-formulated as:

$$\text{minimize } L(v_{bn}, v_{bp}) = L(v_{bn}, v_{bp})_{PABB} + L(v_{bn}, v_{bp})_{TABB} \tag{13}$$

subject to

$$\begin{aligned} D(v_{bn}, v_{bp})_{TABB} &\le D^* \\ D(v_{bn}, v_{bp})_{PABB} &\le D^* \\ v_{bn\,min} &\le v_{bnTABB} \le v_{bn\,max} \\ v_{bn\,min} &\le v_{bnPABB} \le v_{bn\,max} \\ v_{bn} &= v_{bnTABB} + v_{bnPABB} \\ v_{bn\,min} &\le v_{bn} \le v_{bn\,max} \\ v_{bp\,min} &\le v_{bpTABB} \le v_{bp\,max} \\ v_{bp\,min} &\le v_{bpPABB} \le v_{bp\,max} \\ v_{bp} &= v_{bpTABB} + v_{bpPABB} \\ v_{bp\,min} &\le v_{bp} \le v_{bp\,max} \end{aligned}$$

where $D(v_{bn}, v_{bp})_{TABB}$ and $L(v_{bn}, v_{bp})_{TABB}$ are the delay and leakage values from (5) and (9) considering temperature variations only while $D(v_{bn}, v_{bp})_{PABB}$ and $L(v_{bn}, v_{bp})_{PABB}$ are the delay and leakage values from (5) and (9) with process variations only. The limits $v_{bn\,min}$, $v_{bn\,max}$, $v_{bp\,min}$ and $v_{bp\,max}$ are determined by the process-technology used. The final values (v_{bn}, v_{bp}) are programmed into the ROM, as described in Fig. 1. When the circuit is in operation, these values are referenced from the ROM, based on the output of the temperature sensor and the corresponding bias values are applied to recover performance.

TABLE II
TABB COMPENSATION VALUES FOR ISCAS BENCHMARKS

	No Body Bias				Enumeration based TABB				Mathematically assisted TABB								Run-time ratio
									NLPP Solution				Solution after snapping[3]				
Bench mark	D^* (ns)	Temp (C)	Delay (ns)	Lkg (uW)	Delay (ns)	Lkg (uW)	v_{bn} (V)	v_{bp} (V)	Delay (ns)	Lkg (uW)	v_{bn} (V)	v_{bp} (V)	Delay (ns)	Lkg (uW)	v_{bn} (V)	v_{bp} (V)	
C17	0.067	50	0.070	0.067	0.067	0.167	0.1	0.3	0.067	0.159	0.11	0.27	0.067	0.167	0.1	0.3	1.50
C17	0.067	75	0.073	0.158	0.067	0.759	0.4	0.5	0.067	0.811	0.44	0.50	0.067	0.89	0.5	0.5	4.00
C432	0.902	50	0.941	4.78	0.897	11.5	0.2	0.2	0.902	8.58	0.13	0.13	0.907	9.53	0.1	0.2	0.51
C432	0.902	75	0.986	11.2	0.897	47.8	0.4	0.4	0.902	46.1	0.36	0.42	0.902	47.8	0.4	0.4	1.63
C880	0.763	50	0.801	2.90	0.757	8.09	0.2	0.3	0.763	6.83	0.16	0.24	0.757	8.09	0.2	0.3	0.52
C880	0.763	75	0.838	6.85	0.763	37.5	0.5	0.5	0.763	37.6	0.49	0.44	0.763	37.5	0.5	0.5	3.11
C1355	0.83	50	0.841	5.06	0.825	15.7	0.2	0.3	0.83	12.8	0.17	0.24	0.825	15.7	0.2	0.3	0.55
C1355	0.83	75	0.879	11.9	0.825	72.2	0.5	0.5	0.83	69.2	0.50	0.50	0.825	72.2	0.5	0.5	3.10
C3540	1.33	50	1.39	16.0	1.32	31.30	0.2	0.1	1.33	28.4	0.19	0.08	1.32	31.3	0.2	0.1	0.41
C3540	1.33	75	1.45	37.50	1.33	135	0.3	0.4	1.33	136	0.37	0.32	1.33	136	0.4	0.3	0.89
C5315	1.20	50	1.25	14.9	1.19	35.5	0.2	0.2	1.20	30.6	0.13	0.19	1.19	35.5	0.2	0.2	0.42
C5315	1.20	75	1.30	35.0	1.19	144	0.3	0.5	1.20	147	0.40	0.38	1.19	148	0.4	0.4	1.17
C6288	3.64	50	3.82	24.7	3.63	58.7	0.2	0.2	3.64	55.8	0.17	0.19	3.63	58.7	0.2	0.2	0.47
C6288	3.64	75	3.99	57.7	3.61	276	0.4	0.5	3.64	256	0.37	0.46	3.61	276	0.4	0.5	1.75

IV. RESULTS FOR ISCAS BENCHMARK CIRCUITS

In this section, we apply the above described design flow on 7 ISCAS combinational benchmark circuits and present the results obtained. A static timing analyzer (STA) is implemented to determine the delay and leakage of the benchmark circuits. The library consists of 26 gates (10 NOT gates, 5 NAND2 gates, 5 NOR2 gates, 3 NAND3 gates and 3 NOR3 gates) of different sizes, and has been characterized using HSPICE simulations performed using the BPTM 100nm technology [10] with $V_{dd} = 1.0V$. The benchmark circuits have been synthesized based on this library using SIS [12]. Since each ISCAS benchmark is rather small, we consider a test case where all of the benchmarks are placed in different regions of the same chip. Specifically, we assume that each of these benchmarks is in a different WID variational zone, and can be compensated independently of each other.

A. *Results of TABB*

To determine the optimal amount of TABB required, we assume that there are no process variations, and that the on-chip temperature varies from $25^\circ C$ to $75^\circ C$. We also choose $T = 50^\circ C$ and $T = 75^\circ C$ as the points at which we will determine the optimal bias required to maintain the delay. The results obtained through enumeration as well as mathematically assisted methods are explained below:

1) *Enumeration based TABB:* We assume that the devices can be body biased between the range of $[0V, 0.5V]$ with a step of $0.1V$. A step of $0.1V$ is chosen assuming that this is the lowest resolution of voltages that can be generated by the central body bias generator. Thus, 6 possible voltage levels exist for both v_{bn} and v_{bp}, leading to 36 candidate solutions. The benchmarks are simulated at these points, and the solution that satisfies the delay and has the minimum leakage is chosen as the final optimal solution, based on Algorithm 1. The results are tabulated in Table II.
2) *Mathematically assisted TABB:* At each of the temperature points, the delay of the benchmarks are measured at v_{bn} = $[0V,\ 0.1V,\ 0.2V,\ 0.3V,\ 0.4V,\ 0.5V]$ with $v_{bp} = V_{dd}$, and at v_{bp} = $[0V,\ 0.1V,\ 0.2V,\ 0.3V,\ 0.4V,\ 0.5V]$ with $v_{bn} = 0$. Leakage values are measured [4] at v_{bn}=$[0V,0.25V,0.5V]$ and v_{bp}=$[0V,0.25V,0.5V]$. Second order best fit expressions for delay and leakage are obtained as outlined in Section III-A. The NLPP is formulated, as explained in (10) and (11), and the solution obtained for different temperatures is tabulated in Table II. When the NLPP solutions are snapped to points in the discrete solution space, three options exist namely:
   a) Snap both v_{bn} and v_{bp} to the next higher voltage.
   b) Snap v_{bn} to the next higher voltage while v_{bp} to the nearest lower voltage.
   c) 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. The results after snapping are also shown in Table II.

It can be seen from the table that all benchmarks require FBB at higher operating temperatures to compensate for the increase in delay due to reduction in mobilities. Further, most of the NLPP solutions when snapped to the nearest discrete voltage levels give solutions which are identical to that obtained by enumeration. However, for C17 at $T = 75^\circ C$, mathematically assisted TABB returns a solution which is one grid higher for v_{bn} as compared with enumeration due to slight inaccuracies in the delay-leakage model. Due to the same reason, for C432 at $T = 50^\circ C$, mathematically assisted TABB returns a solution which is better than enumeration (but does not meet the delay requirement when back-annotated using STA). Similarly, solutions for C3540 and C5315 at $T = 75^\circ C$ are slightly inferior than the corresponding values obtained through enumeration. The final column in Table II compares the run-time for the two implementations measured on a Linux workstation with a 2.8GHz Pentium CPU. While it can be seen that mathematically assisted TABB is approximately two times faster than enumeration based TABB at $T = 50^\circ C$ (with the exception of the smallest benchmark C17), enumeration outperforms the former for most benchmarks at $T = 75^\circ C$. This is due to the fact that fewer bias pairs at $T = 75^\circ C$ satisfy the delay requirement, and hence the number of candidate solutions for enumerating is quite low. (At $T = 75^\circ C$, only (v_{bn}, v_{bp}) = (0.5,0.5) satisfies the delay requirement for C17,C880 and C1355, and hence enumeration is more than three times faster than mathematically assisted TABB.) However, at $T = 50^\circ C$, the search space for enumeration based TABB increases, and significant speed-up is obtained by the other method.

B. *Results of PABB*

While PABB is actually performed through post-silicon tuning, we perform the same using statistical simulations to get an overview of the nature of results obtainable by our method. The test structures to compensate for process variations in each WID variational zone are assumed to consist of a simple ring oscillator (RO) circuit. Simulations are performed on the ring oscillator using BPTM 100nm. model files [10]. The delay of the RO simulated at $V_{dd} = 1.0V$ and $T = 25^\circ C$ with nominal threshold voltage values ($V_{tn0} = 0.2607V$ and $V_{tp0} = -0.303V$) is $151ps$, while the leakage power

[3]The delay and leakage numbers reported are the STA values obtained after back-annotating the bias voltages.

[4]A minimum of 9 points is required for the leakage interpolation.

is $5.253nW$. We wish to maintain the delay of the RO at this value, denoted by D^*, despite process variations.

The effects of process variations on transistor threshold voltages are quantized using Gaussian distributions for V_{tn0} and V_{tp0}. For simplicity, it is assumed that the statistical distribution of transistor threshold voltages in each WID variational region is the same. This simplification helps us to perform Monte Carlo simulations with one set of Gaussian distribution parameters for transistor threshold voltages, and use the results over all benchmarks. In order to obtain an estimate of the yield without adaptive body bias, Monte Carlo simulations are performed on this ring oscillator with 50 runs at each temperature. If the delay of the RO does not meet the target value, it is assumed that the die fails to meet the delay requirement. The number of dies that satisfy the delay requirement at each temperature is shown in Fig. 4. It can be seen from the figure that only about 50% of the dies are acceptable at room temperature, and this number steadily decreases with increase in temperature. This is attributed to changes in threshold voltages caused by process variations, thereby necessitating compensation using PABB.

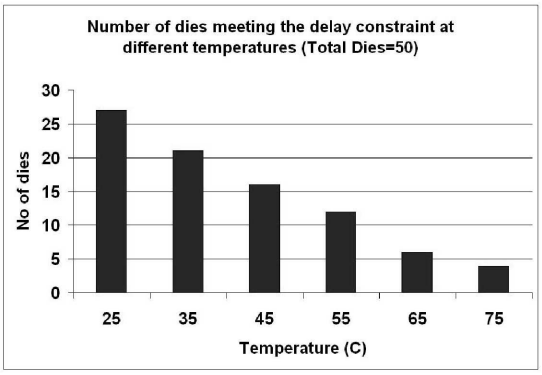

Fig. 4. **Yield at different temperatures for the die**

In order to determine the PABB voltages for each die, the delay and leakage distributions of the test-structure are characterized based on the method described in Section III D. The delay and leakage values with body biasing are measured through simulations, and second order polynomials, as indicated in (10) and (11) are obtained. The NLPP is formulated and solved for each die to determine its optimal bias. All 50 dies have been successfully biased. 42 dies require RBB for PMOS and FBB for NMOS while 6 dies require RBB for both NMOS and PMOS and the remaining 2 require FBB for both NMOS and PMOS. Most dies need FBB for PMOS to increase the speed and RBB for NMOS to minimize the leakage. This is consistent with the observation made by the authors in [1].

The PABB values can be combined with the TABB data obtained from the previous sub-section to determine the PTABB values required for each benchmark at each operating temperature, according to (12). The amount of dies which meet the delay requirement at $T = 50^\circ C$ and $T = 75^\circ C$ for the benchmark circuits and the nature of bias required is shown in Table III. Although 100% of the dies cannot be recovered at $T = 75^\circ C$, the yield can be significantly improved.

TABLE III
PTABB COMPENSATION FOR ISCAS BENCHMARKS

Bench mark	Temp (C)	Accepted Dies	P-FBB N-RBB	P-RBB N-FBB	P-FBB N-FBB
C17	50	50	27	0	23
C17	75	30	0	0	30
C432	50	50	35	0	15
C432	75	38	0	0	38
C880	50	50	24	0	26
C880	75	27	0	0	27
C1355	50	50	24	0	26
C1355	75	24	0	0	24
C3540	50	50	0	4	46
C3540	75	46	0	0	46
C5315	50	50	29	0	21
C5315	75	38	0	0	38
C6288	50	50	27	0	23
C6288	75	39	0	0	39

V. CONCLUSION

Temperature variations and process variations in nanometer-scale devices can cause the delay and leakage of dies to vary significantly. Bidirectional Adaptive Body Bias can be used to improve the yield of dies for reasonable ranges of operating temperatures. We propose an algorithm to compute the exact amount of body bias required to perform run-time compensation to counter thermal variations. We determine these bias values during the design stage using mathematical models and thereby eliminate the need for complex on-chip circuitry to monitor delay and leakage. We also present a unique methodology to decouple the effects of process and temperature variations. We use ABB to counter these individually, and finally combine the values effectively to achieve compensation under all operating conditions. The results indicate that ABB can be used as a successful means to combat both process and temperature variations and improve the performance of gigascale LSI systems.

REFERENCES

[1] 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". In *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 888–899, October 2003.

[2] 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 *International Symposium of VLSI Technology*, pages 159–163, June 1995.

[3] 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 *IEEE International Solid-State Circuits Conference*, pages 166–167, August 1996.

[4] J. W. Tschanz, J.T. Kao, S. G. Narendra, R. Nair, D.A. Antoniadis, A.P. 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*, 37(11):1396–1402, November 2002.

[5] 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*, 38(11):1838–1845, November 2003.

[6] 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*, 38(5):826–829, May 2003.

[7] 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*, 38(5):696–701, May 2003.

[8] 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 *European Solid State Circuits Conference*, pages 391–394, September 2004.

[9] T. Kuroda. "Low power CMOS digital design for multimedia processors". In *International Conference for VLSI and CAD*, pages 359–367, October 1999.

[10] Device Group at UC Berkeley. "Berkeley Predictive Technology Model", 2002. Available at `http://www-device.eecs.berkeley.edu/~ptm/`.

[11] H. Ananthan, C. H. Kim, and K. Roy. "Larger-than-Vdd forward body bias in sub-0.5V nanoscale CMOS" . In *International Symposium on Low Power Electronic Design*, pages 8–13, 2004.

[12] 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". Technical Report UCB/ERL M92/41, 1992. Available at `http://www-cad.eecs.berkeley.edu/research/sis`.