System-Level Mitigation of WID Leakage Power Variability Using Body-Bias Islands *

Siddharth Garg
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
sgarg1@andrew.cmu.edu

Diana Marculescu
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
dianam@ece.cmu.edu

ABSTRACT

Adaptive Body Biasing (ABB) is a popularly used technique to mitigate the increasing impact of manufacturing process variations on leakage power dissipation. The efficacy of the ABB technique can be improved by partitioning a design into a number of "body-bias islands," each with its individual body-bias voltage. In this paper, we propose a system-level leakage variability mitigation framework to partition a multiprocessor system into body-bias islands at the processing element (PE) granularity at design time, and to optimally assign body-bias voltages to each island post-fabrication. As opposed to prior gate- and circuit-level partitioning techniques that constrain the global clock frequency of the system, we allow each island to run at a different speed and constrain only the relevant system performance metrics - in our case the execution deadlines. Experimental results show the efficacy of the proposed framework in reducing the mean and standard deviation of leakage power dissipation compared to a baseline system without ABB. At the same time, the proposed techniques provide significant runtime improvements over a previously proposed Monte-Carlo based technique while providing similar reductions in leakage power dissipation.

Categories and Subject Descriptors: B.7.0 [**HARDWARE**]: Integrated Circuits - *General*

General Terms: Design, Algorithms

1. INTRODUCTION

Variations in integrated circuit (IC) manufacturing process parameters such as gate length and channel doping concentration lead to variations in the power dissipation and performance characteristics of the fabricated die. In particular, due to the exponential dependency of leakage power on transistor threshold voltage, small variations in the process parameters that affect the threshold voltage can lead to significant variations in the leakage power dissipation of a transistor [11]. Furthermore, as process technology continues to scale, the contribution of leakage power to overall power dissipation is steadily increasing. As a result, design methodologies that can reduce leakage power variability with minimal impact on system performance are imperative.

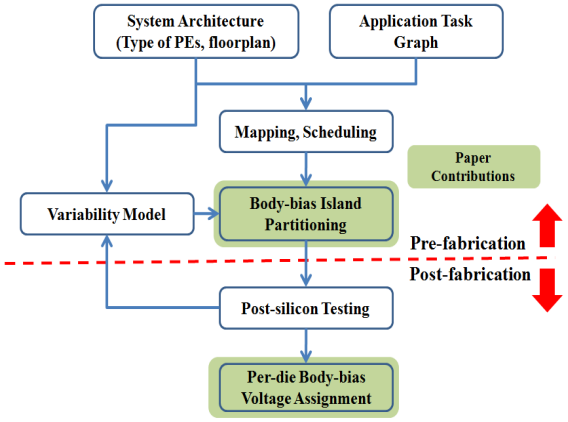

Figure 1: High-level overview of the proposed framework. The shaded boxes represent the novel contributions of our work.

Adaptive Body Biasing (ABB) is an effective *post-silicon* technique to minimize the process-driven variability in leakage power dissipation [14]. Specifically, a negative body-to-source voltage applied to an nMOS device, also called Reverse Body Biasing (RBB), reduces its leakage power, while applying a positive body-to-source voltage, or Forward Body Biasing (FBB), increases its leakage power. Ideally, each device on a chip would have its own individual, independently tunable body bias voltage to optimally combat WID process variations. Unfortunately, the area overhead and manufacturing cost of such a system would be prohibitive - in practice, therefore, ABB is implemented using either two global body-bias voltages (one for all nMOS devices and another for all pMOS devices) or by partitioning the die into a number of body-bias islands, with independently tunable body bias values for each island [14]. The second alternative provides the additional flexibility to address within-die (WID) sources of variability, without incurring the significant area overhead of individual body-bias voltages for each transistor. Once the system has been partitioned into body-bias islands at *design time*, the body bias voltages for each island can be tuned separately for each chip post-silicon based on the specific impact that process variability has on that particular chip. In this paper, we propose a framework for variability-aware, system-level partitioning of Multiple Processor Systems-on-Chip (MPSoCs) into body-bias islands at the processing element (PE) granularity. Each island receives an independent body-bias voltage that can be tuned post-silicon, and also sets its clock frequency based on the body-bias voltage it receives (RBB reduces clock frequency, while FBB increases clock frequency). Furthermore, each

* This research was funded in part by NSF Award CCR0702451 and SRC/GRC contracts 2005-HJ-1314 and 2008-HJ-1800.

CODES+ISSS'08, October 19–24, 2008, Atlanta, Georgia, USA.

die is constrained to meet the desired performance specifications - therefore, if some islands on a particular die are slowed down because they received a reverse body-bias, other islands must be sped-up (and forward body-biased) to compensate for the performance loss. The framework consists of two sub-problems for which we propose efficient and scalable algorithms: (1) deciding the optimal partitioning of the system into body-bias islands at design time, and (2) determining the optimal body-bias values for each island for every manufactured die based on post-silicon leakage power measurements. Figure 1 shows a high-level overview of the proposed work, highlighting our specific contributions.

As opposed to prior work on gate-level body-bias island partitioning, our focus on system-level partitioning is motivated by the need to address the impact of process variations as early in the design cycle as possible. Furthermore, instead of meeting a specified global clock frequency constraint, we can allow different parts of the design to run at different speeds as long as the performance targets are met. Finally, as intellectual property (IP) based designs become more prevalent, it may not be possible to partition an IP block into separate body-bias islands at the gate-level, thereby making a PE level granularity of partitioning inevitable.

2. RELATED WORK

There is a significant body of research over the last few years that addresses the analysis and mitigation of leakage power variability for digital ICs. In [11], the authors analyze the impact of various sources of manufacturing process variation on sub-threshold leakage power dissipation. [7] discusses design time techniques that can be used at the circuit, micro-architecture and system level to minimize leakage power dissipation in scaled technologies. Finally, [14] demonstrates the effectiveness of ABB in reducing the impact of both D2D and WID process variations on leakage power.

More recently, in [8], the authors propose a gate-level partitioning strategy to form body-bias clusters at design time. The primary goal is to reduce the variability in leakage power dissipation using ABB, with a constraint on the clock speed of the design. As opposed to [8], we allow different PEs in the system to operate at different clock speeds, based on the body-bias voltage of the island they lie in, but constrain the execution deadlines of the task graph running on the multiprocessor system. In [6], the authors also consider the case of systems partitioned into a number of body-bias clusters; however, the focus of this work is on the allocation of body-bias voltage assignments for each die *after* they have been manufactured. It does not address the *design time* partitioning of the system into body-bias islands. Furthermore, like [8], [6] concentrates on fully synchronous designs with constraints on the global clock frequency of the system, while we impose constraints only on system-level performance metrics.

There is comparatively little work that addresses leakage variability at the higher levels of design abstraction. In [12], the authors explore the use of dynamic fine-grained body-biasing at the module level for a superscalar out-of-order processor. In [15], the authors solve the post-silicon ABB aware module selection problem in high-level synthesis. Both these works assume a fine granularity of body-bias island partitioning in which each module lies in a separate body-bias island, while this paper concentrates on finding the optimal coarse grained partitioning.

3. PRELIMINARIES AND ASSUMPTIONS

We begin by discussing the assumptions and notation used in the rest of the paper. We begin with the system architecture and performance characterization, followed by the process variability model used and finally the impact of body biasing on the leakage power and the cycle time of the PEs.

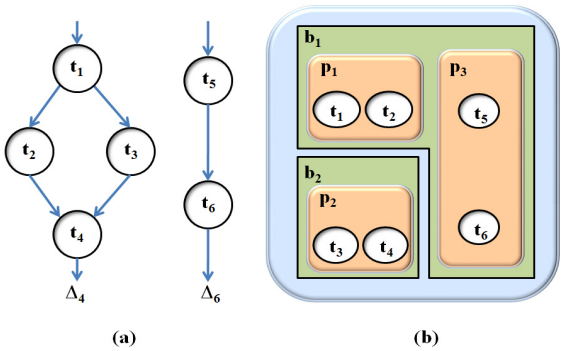

Figure 2: (a) An example task graph T with six tasks. (b) A multiprocessor system P with three processors.

3.1 Application Modeling

As mentioned before, we consider multiprocessor systems comprising of synchronous computational components, either hard IPs or embedded processors (referred to as *processing elements* or PEs), running an application specific task graph. The task graph is modeled as a directed acyclic graph (DAG), $G = (T, E)$, where a vertex $t \in T = \{t_1, t_2, \ldots, t_N\}$ represents a computation task and an edge $e \in E \subseteq T \times T$ represents control or data dependencies between tasks. The PEs in the system are represented by a set $P = \{p_1, p_2, \ldots, p_M\}$. We assume that the tasks have already been mapped on the processors in the system, i.e., there exists a known mapping function $M_1 : T \rightarrow P$ that maps each task in T to a unique processor in P. Finally, we assume that tasks mapped to the same processor are assigned a static priority based schedule and include any structural dependencies that arise from the schedule as edges in the set E. Figure 2 shows an example of a task graph with six tasks mapped to a multiprocessor system with three PEs. Following the terminology in [4], a task $t_i \in T$ is called a *terminal node* if it has no successors (tasks t_4 and t_6 in Figure 2) and an *input node* if it has no predecessors (tasks t_1 and t_5 in Figure 2).

To measure the performance of a task graph T mapped on the set of PEs P, we denote the execution time of task t_i ($1 \leq i \leq N$) on its PE, $M_1(p_i)$, by λ_i. The *completion time* of task t_i is denoted by Λ_i and can be computed as $\Lambda_i = \lambda_i$ for every *input node* in T and as $\Lambda_i = \lambda_i + max_{j:(t_j,t_i)\in E}(\Lambda_j)$ for all other nodes. Every terminal node t_i in the task-graph has an associated deadline constraint Δ_i, and the performance constraints are met only if $\Lambda_i \leq \Delta_i$ for every terminal node. For notational simplicity, in the rest of this paper, we will assume that every *terminal node* has the same deadline constraint Δ. However, we note that our framework does not need this assumption to be satisfied, and the results are demonstrated for a general case in which each terminal node has a different deadline constraint. The deadline constraints described above can be translated to *path based* constraints by defining a set of all paths, $\Phi = \{\phi_1, \phi_2, \ldots, \phi_S\}$, in the task graph. Each ϕ_i ($1 \leq i \leq S$) is a subset of tasks in T that form a path from an input node in T to a terminal node. Given these paths, we can express the deadline constraints as:

$$\sum_{i:t_i \in \phi_j} \lambda_i \leq \Delta, \forall j : \phi_j \in \Phi \quad (1)$$

3.2 Hardware Architecture

We assume that the system architecture consists of M heterogeneous PEs given by the set $P = \{p_1, p_2, \ldots, p_M\}$. Furthermore, each PE lies in one of K body-bias islands represented by set $B = \{b_1, b_2, \ldots, b_K\}$, where $K \leq M$. For example, in Figure 2, the three PE system is partitioned into two body-bias islands ($K = 2$), b_1 and b_2. The mapping of the PEs to body-bias islands

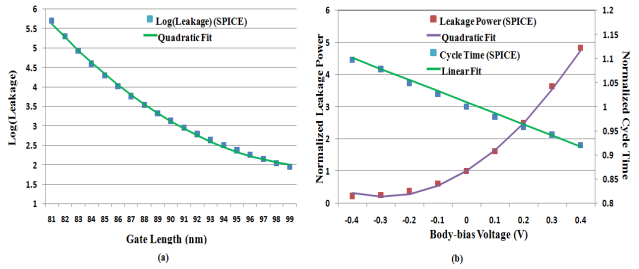

Figure 3: Impact of (a) gate length on the leakage power dissipation, and (b) body-bias voltage on cycle time and leakage power dissipation for $90nm$ technology.

is defined by a mapping function, $M_2 : P \rightarrow B$, that maps each PE in the design to one of the K body-bias islands at design time. This mapping function is *not* known in advance, and computing the best M_2 is one of the primary goals of this work.

The body-bias voltage and frequency of each island can be tuned post-fabrication on a per-die basis to minimize the leakage power dissipation of each die. As discussed in [14], it is not feasible to allow for a continuous range of body-bias voltage and frequency values. Instead, as in [14], we assume that the body-bias voltages and clock frequencies are restricted to a discrete set of uniformly spaced values, controlled by a digital input. Finally, since each island on a given die can potentially run at a different speed after body-biasing, we assume that inter-island communication occurs via point-to-point asynchronous FIFOs [3], while intra-island communication links can be fully-synchronous.

3.3 Variability Modeling

To model the impact of variability on the system, we define the *random vector* $L = \{L_1, L_2, \ldots, L_M\}$, where L_i is a *random variable* that represents the *intrinsic* leakage power dissipation (i.e., the leakage power dissipation in the absence of any body-biasing) of PE p_i. We compute the distribution of random vector L using Monte Carlo sampling, although any of the previously proposed analytical frameworks can as easily be used. We model both D2D and WID variations in transistor gate length and assume that the WID gate length variations are spatially correlated [5] with the correlation coefficient decaying exponentially with distance, as described in [16]. We note that while other sources of process variations can be modeled as well, gate length variations have been shown to be the dominant source of leakage variability [11]. Figure 3(a) shows the impact of gate length variation on leakage power dissipation for an inverter in a $90nm$ technology.

Now, we define the *random vector* $V = \{V_1, V_2, \ldots, V_M\}$ to represent the body-bias voltage that each PE in the design receives post-silicon. Even though it may be counter-intuitive to think of body-bias voltages as random variables, the "randomness" arises from the fact that each fabricated die receives a different body-bias voltage configuration based on its leakage power values. In fact, as we will show, computing the distribution of the random vector V is an important step in our body-bias partitioning framework.

3.4 Impact of Body-Biasing

Body-biasing works by modulating the threshold voltage of the transistors to which it is connected. Reverse body biasing (RBB) increases the threshold voltage and thereby decreases the leakage current of a device, but at the same time increases its delay by reducing the drive strength of the device. The opposite is true for forward body biasing (FBB). In the proposed framework, all gates within a PE receive the same body-bias voltage; therefore, a RBB leads to a decrease in leakage power and an increase in the cycle time (or decrease in clock frequency) for that PE. In [8], the authors show that the impact of body-biasing on delay and leakage power can be modeled as linear and quadratic functions of the body-bias voltages. Let $\tilde{L_i}$ be the leakage power dissipation random variable for PE p_i *after* body-biasing, we can write:

$$\tilde{L_i} = L_i(1 + \alpha V_i + \beta V_i^2), \forall i : p_i \in P \quad (2)$$

Furthermore, a change in the frequency of PE p_i due to body-biasing will also *proportionally* affect the execution latency of all the tasks mapped to it. If we denote by $\tilde{\lambda_i}$ the execution latency of task t_i *after* body-biasing, we can write:

$$\tilde{\lambda_i} = \lambda_i(1 - \gamma V_{M_1(i)}), \forall i : t_i \in T \quad (3)$$

Figure 3(b) shows the dependence of the cycle time and leakage power dissipation of a simulated five stage ring-oscillator in a $90nm$ technology on the body-bias voltage. As can be seen, the linear and quadratic fits are extremely accurate and are used to compute the coefficients α, β and γ in Equation 2 and Equation 3.

4. SYSTEM PARTITIONING INTO BODY-BIAS ISLANDS

Having described the assumptions and mathematical notation, we now move towards solving the body-bias partitioning problem, i.e., finding the best mapping, $M_2 : P \rightarrow B$, from PEs to body-bias islands. Recall that we have M PEs and K body-bias islands, which yields a design space of K^M possible partitioning options. Exhaustively searching the design space is clearly infeasible, even for small problem sizes. Instead, we solve this problem in two steps: we first assume that the multiple processor system is partitioned at the finest possible granularity, i.e., each PE lies in a separate body-bias island. Under this assumption, we then find the optimal body-bias distribution for each PE using a novel convex optimization based approach. Once we know the optimal distributions of body-bias voltages for each PE, we would like to assign the PEs that have similar body-bias voltage distributions (i.e., are more likely to be set at similar body-bias voltages) to the same body-bias island. This is done by defining an appropriate distance measure over the body-bias voltage distributions and projecting the resulting pair-wise distances matrix in a two-dimensional space. The final step involves clustering the points in the two-dimensional space using a single run of the *k-means* clustering algorithm. This yields the final design time partitioning of PEs into body-bias islands.

We note that a similar idea of computing body-bias distributions at the finest granularity has been used in [8], although their focus was on gate-level partitioning to meet global clock frequency constraints. Moreover, unlike the work in [8] that needs multiple runs of an optimization loop to compute the body-bias distributions, our proposed convex optimization formulation is run only once, yielding a *significant* improvement in run-time. Furthermore, the distance measure that we define over the computed body-bias voltage distributions is tuned to account for the significant degree of heterogeneity in the size, leakage power dissipation, and performance of the PEs that multiprocessor systems exhibit. Ignoring this heterogeneity can yield sub-optimal results.

4.1 Body-bias Voltage Distributions

The first step of the algorithm is to compute the body-bias voltage distributions V_i for each PE $p_i \in P$, assuming that each PE lies in a separate body-bias island. We first write the total leakage power dissipation random variable after the impact of body-biasing, L_{tot}, as:

$$L_{tot} = \sum_{i=1}^{M} \tilde{L_i} = \sum_{i=1}^{M} L_i(1 + \alpha V_i + \beta V_i^2) \quad (4)$$

We would now like to minimize the *expectation* of the total leakage power random variable. This gives the following objective function:

$$\min_{V} E(\sum_{i=1}^{M} L_i(1 + \alpha V_i + \beta V_i^2)) \quad (5)$$

Note that the feasible space of the minimization is over all possible distributions of the random vector V. We now add the performance constraints that ensure that the *deadline*, Δ, at each terminal node in the task graph T must be met. To do so, we can rewrite Equation 1 as:

$$\sum_{i:t_i \in \phi_j} \lambda_i(1 - \gamma V_{M_1(i)}) \leq_S \Delta, \forall j : \phi_j \in \Phi \quad (6)$$

It is important to note that the L.H.S. of Equation 6 is now a random variable that represents the latency of each path in the task graph under the impact of body-biasing. As a result, we have replaced the simple inequality in Equation 1 with a *stochastic inequality* constraint represented by $\leq_S$. The stochastic inequality constraint simply means that the random variable of path delay for each path must *always* be less than the deadline constraint, Δ; i.e., the constraint must be met with probability 1. This problem, as defined by the objective function in Equation 5 and the constraints represented by Equation 6, does *not* admit an efficient solution [2]. To make the problem tractable, we use an idea from the adjustable robust optimization approach presented in [1, 9] - we assume that the unknown variable, i.e., the random vector V, is an affine function of the known leakage power random vector L. In other words, we write each V_i as:

$$V_i = \sum_{j=1}^{M} s_{ij} f(L_j), \forall i : 1 \leq i \leq M \quad (7)$$

where $s_{ij} \in \mathbb{R}$ $(1 \leq i, j \leq m)$ and $f : \mathbb{R} \rightarrow \mathbb{R}$ can be any arbitrary function. In matrix form, Equation 7 can be written simply as $V = Sf(L)$, where $S \in \mathbb{R}^{M \times M}$. The advantage of this assumption is that it reduces the search space from all possible distributions of the random vector V to a search for the best matrix S, which makes the problem more tractable. Rewriting the objective function and constraints in terms of S, we get:

$$\min_{S} \alpha \sum_{i=1}^{M} \sum_{j=1}^{M} s_{ij} E(L_i f(L_j)) + \beta \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{k=1}^{M} s_{ij} s_{ik} E(L_i f(L_j) f(L_k)) \quad (8)$$

While we omit the detailed proof due to space constraints, we note that the objective function in Equation 8 is a convex function of the elements of S.

The constraints from Equation 6 can also be expressed in terms of S as follows:

$$\sum_{i:t_i \in \phi_j} \sum_{k=1}^{M} \lambda_i(1 - \gamma s_{M_1(i)k} f(L_k)) \leq_S \Delta, \forall j : \phi_j \in \Phi \quad (9)$$

To convert the *stochastic* inequality in Equation 9 to a regular inequality, we relax the constraint that delay of each path should *always* be less than the deadline to a constraint that ensures that the deadlines are met *with high probability*. If δ_i represents the path delay random variable for path ϕ_i $(i : \phi_i \in \Phi)$, we can rewrite Equation 9 as:

$$E(\delta_i) + \eta \sqrt{Var(\delta_i)} \leq \Delta, \forall i : \phi_i \in \Phi \quad (10)$$

where Var represents the variance of a random variable. The coefficient η determines the probability with which the path delay random variable is required to meet the deadline constraint; in particular, this probability is an increasing function of the coefficient, η. In our experiments, we find that setting $\eta = 3$ provides uniformly good results. Note that this is a *design time approximation* only; after fabrication, we ensure that *every* fabricated die meets the deadline constraints. While we omit details of the proof of convexity of the constraints described by Equation 10 due to space limitations, we note that the random variable δ_i $(\forall i : \phi_i \in \Phi)$ is a linear *weighted* sum of set of random variables, where the weights are real valued and unknown; i.e., the elements of the S matrix. Equation 10 can therefore be converted to an equivalent convex *second order cone* (SOC) constraint [2], one for each path in the task graph. We now obtain a convex program in standard form, with the minimization of a convex objective function (Equation 8), over a convex feasible region (Equation 10). This can be solved in polynomial time by any standard convex optimization toolbox (we use the publicly available *cvx* tool from Stanford University).

Note that the formulation described above is convex for any choice of the function f. In practice, we found that using $f(L_i) = \frac{1}{\sqrt{L_i}}$ provides the most accurate results across all the benchmarks we tested. This is, in fact, not surprising because: (1) the body-bias voltage is a decreasing function of leakage power dissipation, since PEs with high leakage tend to be set to a low body-bias voltage, and (2) the leakage power has a quadratic dependence on the body-bias voltage. Also, we note that the algorithm requires us to compute a number of expectation terms over functions of the leakage power dissipation random variables, for example $E(L_i f(L_j))(1 \leq i, j \leq n)$. Since we have characterized the distribution of L in Section 3.3, the expectation terms can be computed using either fast numerical integration algorithms or Monte Carlo estimation. We use the latter because of its ease of implementation. Finally, even though we said that the body-bias voltages are restricted to lie within a discrete set of values for practical reasons, it is clear that we have not enforced that constraint in our body-bias voltage prediction formulation. This is done on purpose - imposing that restriction would make the problem NP-complete. Furthermore, the purpose of this step is only to determine at *design time* which PEs are more likely to receive similar body-bias values, (not to actually assign these voltages to the fabricated die), thereby making our assumption reasonable.

4.2 Body-bias Voltage Clustering

In the previous Section we outlined a convex optimization approach to determine the body-bias voltage distribution of each PE, assuming that it lies in a separate body-bias island. We would now like to group PEs that are assigned similar body-bias voltage distributions together. This allows a coarse granularity partitioning of body-bias islands to mimic, as closely as possible, the finest granularity body-bias island partitioning in which each PE can independently tune its body-bias voltage. We begin by computing a "distance" between each pair of PEs in the design that represents their affinity to lie in the same body-bias island.

While computing the distance between two body-bias voltage distributions, it is insufficient to simply compare their statistics, such as mean and variance, of the distributions. Consider a scenario in which the body-bias voltage distributions of two PEs have the *same* mean and variance but are negatively correlated. Let us also assume that the mean of the distributions is zero. In this case, every time one of the PEs is tuned to a positive body-bias voltage (FBB), the other PE will receive an RBB and vice-versa. As a result, these PEs may not be good candidates for allocation to the same island, though a distance computed only using their mean

and variance terms may indicate otherwise. A second concern that is particularly relevant in the case of heterogeneous multiprocessor systems is that the PEs that contribute more to the overall leakage power dissipation should be less likely to be allocated to islands where their body-bias distributions are changed significantly from the ideal distributions with the finest granularity of body-bias partitioning. To account for both these factors, we define a distance measure d_{ij} between PE p_i and p_j as:

$$d_{ij} = E(L_i + L_j)E((V_i - V_j)^2), \forall 1 \leq i, j \leq M \quad (11)$$

where the distance between the voltage distributions of two PEs is quantified by the $E((V_i - V_j)^2)$ term, while the "inertia" of PEs that contribute significantly to the total leakage power is quantified by the $E(L_i + L_j)$ term.

The final step of the body-bias partitioning framework is to project the computed pairwise distances into a low dimensional space. As an analogy, it is useful to think of a situation in which we have the pairwise distance between a set of "cities" (PEs) and we want to reconstruct the locations of these cites on a 2-dimensional map. These reconstructed locations can then be used to cluster the "cities" into the desired number of groups. This problem falls under a broad family of techniques known as *multidimensional scaling* - we use the classical multidimensional scaling algorithm (also called Torgerson Scaling) presented in [13], to obtain, for each PE $p_i \in P$, a tuple (x_i, y_i) that represents its location in a 2D Euclidean space. We now use a standard *k-means* clustering algorithm on the set of M points in a 2D space to obtain the mapping $M_2 : P \rightarrow B$. This completes our description of the variability-aware design time partitioning of PEs into body-bias islands. In the next section, we discuss post-fabrication tuning of the body-bias voltages for each island for every fabricated die.

5. POST-FABRICATION TUNING

Once the PEs are clustered into K body-bias islands at design time and the design is manufactured, the next problem is to optimally select the body-bias voltage assignments for each manufactured die using the measured leakage power dissipation values for that die. Consider a single manufactured die with leakage power dissipation vector $l = \{l_1, l_2, \ldots l_M\}$, where l_i is the leakage power dissipation of PE p_i ($\forall i : p_i \in P$). Note that unlike the leakage power random vector L that was used in the design time partitioning, l is a set of *fixed numbers* for a given die, and is an instance of the L random vector. Let $v^* = \{v_1^*, v_2^*, \ldots, v_M^*\}$ be the optimum body-bias voltages for each PE on the die. Since our objective is to minimize the leakage power dissipation each die, we can write v^* as:

$$\min_{v} \sum_{i=1}^{M} l_i(1 + \alpha v_i + \beta v_i^2) \quad (12)$$

Next, we must constrain PEs that lie in the same body-bias island to receive the same voltage, based on the mapping from PEs to body-bias islands M_2. This gives us the following constraint:

$$v_i = v_j, \forall (i, j) : M_2(p_i) = M_2(p_j) \quad (13)$$

The performance constraints can be written as:

$$\sum_{i:t_i \in \phi_j} \sum_{k=1}^{M} \lambda_i(1 - \gamma v_{M_1(k)}) \leq \Delta, \forall j : \phi_j \in \Phi \quad (14)$$

Finally, as discussed in Section 3.2, the body-bias voltage values are, in practice, restricted to a discrete set of *uniformly spaced* values. Let this discrete set of voltage values be $vb = \{vb_1, vb_2, \ldots, vb_F\}$, where $vb_1 \leq vb_2 \ldots \leq vb_F$. This gives us a final set of constraints:

$$v_i \in \{vb_1, vb_2, \ldots, vb_F\}, \forall i : 1 \leq i \leq M \quad (15)$$

Unfortunately, restricting the body-bias voltage to a discrete set yields a computationally expensive combinatorial optimization problem. On the other hand, since the optimization algorithm needs to be run repeatedly for each manufactured die, its run-time needs to be kept as low as possible. With this in mind, we relax the constraints in Equation 15 to the following constraint:

$$vb_1 \leq v_i \leq vb_F, \forall i : 1 \leq i \leq M \quad (16)$$

to yield a quadratic programming problem that can be efficiently solved. Let the solution of the *relaxed* optimization problem, consisting of the objective function in Equation 12, and the constraints specified in Equation 13, Equation 14 and Equation 16, be represented by v'. To recover v^* from the relaxed solution v', we use a simple greedy algorithm that begins by setting each body-bias voltage from the relaxed solution, v', to the next highest voltage in the discrete set vb. Then, in each subsequent iteration, we reduce the body-bias voltage of the island that provides the maximum leakage power savings from this reduction, while still satisfying the performance constraints. The iteration stops when none of the islands can have their body-bias voltages reduced any further without violating the performance constraints.

6. EXPERIMENTAL RESULTS

For our experiments, we use the two largest applications from the E3S benchmark suite [4] - the *telecom* and *auto-industry* applications, consisting of 30 and 24 tasks respectively. The applications are mapped to heterogeneous multiprocessor platforms consisting of embedded processors drawn from the E3S embedded processor database. Specifically, the *telecom* application is mapped and scheduled to two separate multiprocessor systems consisting of 24 (*telecom-24*) and 28 processors (*telecom-28*), while the smaller *auto-industry* application is mapped and scheduled to two systems with 16 (*auto-16*) and 20 processors (*auto-20*) respectively. This gives a set of four benchmarks to test our proposed framework on. So as to model realistic designs, for each benchmark, we set the deadlines at the terminal nodes such that all the constraints are tight, i.e., *the terminal nodes do not have any additional slack*. Finally, the multiprocessor systems are floorplanned using an in-house, simulated annealing based slicing-tree floorplanning algorithm with an area minimization objective.

The process variation model and the impact of body-biasing on leakage and delay are characterized for a BPTM $90nm$ technology library, as described in Section 3.3 and Section 3.4. The total 3σ value of gate length variation is assumed to be 15% of the nominal gate length, and the spatial correlation model is configured such that correlations become negligible at half the die length [5].

As a baseline to compare the proposed design time body-bias island partitioning algorithm with, we use a Monte-Carlo based approach, similar to the one presented in [8] to compute the distribution of the body-bias voltage vector V. For fairness, we use the same clustering algorithm for the Monte-Carlo approach that was used for the proposed approach.

After design-time partitioning of the system using the proposed and Monte-Carlo approaches, which we will henceforth refer to as ***Proposed*** and ***MC*** respectively, we generate 5,000 post-fabrication die instances of the partitioned designs, and for each die, we assign optimal body-bias voltages for each island using the technique described in Section 5. The post-silicon voltage assignments are restricted to lie in a discrete set of 32 values, corresponding to a 5-bit digital control for the on-chip body-bias voltage generators [14]. Finally, to highlight the advantages of variability-aware partitioning proposed in this work, we also compare our results against a variability-unaware approach, similar to the one presented in [10], that considers only the nominal leakage power dissipation values

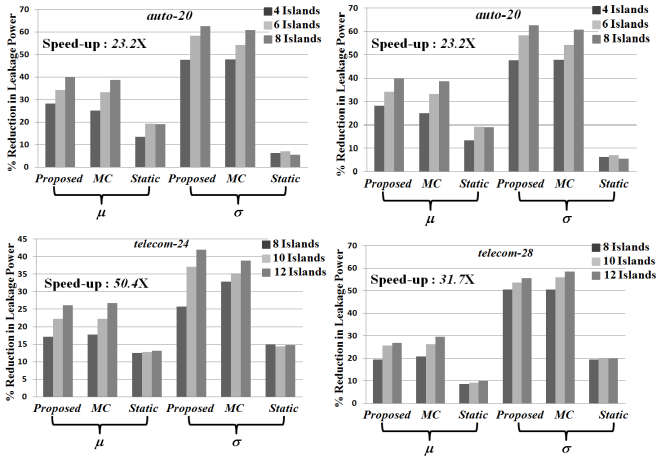

Figure 4: % reduction in the mean (μ) and standard deviation (σ) of leakage power dissipation as a function of the number of body-bias islands in the system for the *Proposed*, *MC* and *Static* approaches.

of each PE to create body-bias islands and statically assign them to their best body-bias voltages (referred to as ***Static***). Figure 4 shows the % reduction in the mean, μ, and standard deviation, σ, of the total leakage power dissipation compared to the original design for all four benchmarks and as a function of the number of body-bias islands into which the design is partitioned. Additionally, each graph in Figure 4 is annotated with the speed-up in run time of the partitioning algorithm obtained by using the *Proposed* approach compared to the *MC* approach for the given benchmark. From the results we can make the following observations:

- Both the *Proposed* and *MC* approaches lead to **significant reduction in the mean (average 28.25% and 28.24% reduction across all benchmarks) and standard deviation (average 48.27% and 48.1% reduction across all benchmarks) of the leakage power dissipation** as compared to the original design. On the other hand, the *Static* approach can only provide a 19.6% average reduction in the mean and only 14% reduction in the standard deviation of the leakage power dissipation.

- **The *Proposed* approach provides an average $38\times$ speed-up in design-time partitioning as compared to the *MC* approach**, without any loss in performance. This speed-up in run-time can be extremely crucial if optimal design-time partitioning into body-bias islands is included as an inner optimization loop within a larger variability aware framework that includes mapping, scheduling and floorplanning. We demarcate exploring this possibility as future work.

- Increasing the number of body-bias partitions helps the variability aware approaches more than the *Static* approach. **In fact, for the *Static* approach, increasing the number of body-bias islands provides no additional reduction in the standard deviation of the leakage power dissipation for three of the four benchmarks we studied**.

The results presented in this paper can be used by system-level designers to explore variability-aware decisions that provide a tradeoff between increased implementation complexity and reduced leakage variability - for example, the choice of adaptive versus static body-bias assignment, the number of body-bias islands and the number of body-bias voltage levels for each island.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an efficient and scalable framework for leakage variability-aware, system-level partitioning of a design into body-bias islands and post-silicon ABB voltage assignment for each island. Our results indicate the computational efficiency of our approach compared to a previously proposed Monte-Carlo partitioning approach (average $38\times$ speed-up), with similar leakage power savings. Our results indicate that up to 40% **reduction in the mean** and up to 62% **reduction in the standard deviation** of the leakage power can be obtained using the proposed framework, compared to the original design. As future work, we plan to integrate our design-time partitioning approach into a comprehensive variability-aware system-level mapping and floorplanning framework.

8. REFERENCES

[1] A. Ben-Tal et al. Adjustable robust solutions of uncertain linear programs. In *Math. Program.*, pages 351–376, 2004.

[2] S. Boyd and L. Vandenberghe. Convex Optimization. *Cambridge University Press*, 2004.

[3] T. Chelcea and S. Nowick. Robust interfaces for mixed-timing systems with application to latency-insensitive protocols. In *Proceedings of DAC*, 2001.

[4] R. Dick. E3s: The embedded system synthesis benchmarks suite.

[5] P. Friedberg et al. Modeling within-die spatial correlation effects for process-design co-optimization. In *Proceedings of ISQED*, 2005.

[6] J. Gregg and T. Chen. Optimization of individual well adaptive body biasing using a multiple objective evolutionary algorithm. In *Proceedings of the 6th ISQED*, 2005.

[7] J. Kao et al. Subthreshold leakage modeling and reduction techniques. In *Proceedings of ICCAD*, 2002.

[8] S. Kulkarni et al. A statistical framework for post-silicon tuning through body bias clustering. In *Proceedings of ICCAD*, 2006.

[9] M. Mani et al. Joint design-time and post-silicon minimization of parametric yield loss using adjustable robust optimization. In *Proceedings of ICCAD*, 2006.

[10] U. Ogras et al. Voltage-frequency islands partitioning for gals based networks-on-chip. In *Proceedings of DAC*, 2007.

[11] R. Rao et al. Statistical estimation of leakage current considering inter-and intra-die process variation. In *Proceedings of ISLPED*, 2003.

[12] R. Teodorescu et al. Mitigating parameter variation with dynamic fine-grain body biasing. In *Proceedings of MICRO*, 2007.

[13] W. S. Torgerson. Multidimensional scaling: theory and method. In *Psychometrika*, 1952.

[14] J. Tschanz et al. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. In *IEEE Journal of Solid-State Circuits*, 37(11), 2002.

[15] F. Wang et al. Variability-driven module selection with joint design time optimization and post-silicon tuning. In *Proceedings of ASP-DAC*, 2008.

[16] J. Xiong et al. Robust extraction of spatial correlation. *Proceedings of ISPD*, 2006.