Maximizing Yield in Near-Threshold Computing under the Presence of Process Variation

Nathaniel A. Conos, Saro Meguerdichian, Sheng Wei, and Miodrag Potkonjak
Computer Science Department
University of California, Los Angeles
{conos, saro, swei, miodrag}@cs.ucla.edu

***Abstract*—Near-Threshold Computing (NTC) shows potential to provide significant energy efficiency improvements as it alleviates the impact of leakage in modern deep sub-micron CMOS technology. As the gap between supply and threshold voltage shrink, however, the energy efficiency gains come at the cost of device performance variability. Thus, adopting near-threshold in modern CAD flows requires careful consideration when addressing commonly targeted objectives. We propose a process variation-aware near-threshold voltage (PV-N_{vt}) gate sizing framework for minimizing power subject to performance yield constraints. We evaluate our approach using an industrial-flow on a set of modern benchmarks. Our results show our method achieves significant improvement in leakage power, while meeting performance yield targets, over a state-of-the-art method that does not consider near-threshold computing.**

I. Introduction

Power and performance continue to be the top design metrics for optimization in modern and pending IC technologies. The near-threshold computing (NTC) paradigm has been shown to provide significant energy efficiency improvements of 10X by scaling supply voltage (V_{dd}) to comparable levels to the nominal threshold voltage (V_{th}). However, several challenges must be addressed when incorporating NTC in modern CAD flows, such as their significant performance cost and high susceptibility to performance variations due to the lessening gap between supply and threshold voltages [25]. As devices continue to scale into the deep sub-micron regime, these issues will be further compounded by process variation.

Process variation (PV) is an unavoidable side product of modern and pending silicon implementation technologies. As a ramification of PV, each transistor, gate, and wire on each integrated circuit that realizes a particular design has unique physical (e.g. channel length) and manifestational (e.g. power and delay) properties [1][2][3]. When considering such implications in variations for a near-threshold voltage (N_{vt}) design where the gap between V_{dd} is reduced, the effects on respective leakage and delay components are magnified.

PV may eliminate most of the potential gains from one technology generation or design optimization [2]. For example, due to the impact of PV, the classic design approaches that aim to optimize delay and power, such as [4], would not produce optimal solutions. Therefore, there is an increasingly pressing need for the design practice to switch from a fixed deterministic domain to a usually infinite probabilistic and statistical domain, in order to reflect the changes brought about by the existence of PV [5].

In this paper, we consider an IC design optimization problem of performing a process variation-aware near-threshold voltage (PV-N_{vt}) gate sizing method. Our goal is to optimize the yield of a given IC design (i.e., to maximize the total number of ICs that meet a certain set of power and delay specifications). We achieve this goal by gate sizing and selecting threshold voltage settings for the gates on the circuit in such a way that the maximum number of ICs can meet the power and delay requirements. In order to reflect the impact of PV, we use a scenario-based approach by creating a set of scenarios (IC samples) that are representative of the PV model.

First, we perform simultaneous delay and power optimization of a circuit to meet a target performance and power budget. Once a specified threshold is satisfied, the next step is to optimize the yield of the set of scenarios. For this step, we have developed an iterative heuristic algorithm that efficiently identifies the most problematic sections of the circuit (through our partitioning scheme) in terms of the objective function target (e.g., delay and leakage power). The critically determined sections are then made more resilient to PV in order to maximize yield. Each partition is optimized such that the observed maximally benefiting configuration (e.g., size and V_{th}) is selected, such that the overall yield is improved, while taking into account of the global circuit optimization search space. We validate our approach using statistical re-sampling techniques on various generated scenarios. The procedure is constructive in nature and can be repeated to improve the design at the expense of additional run-time.

II. Motivation

We begin by providing a small realistic example demonstrating the advantages and challenges encountered when considering an NTC-enabled design. Consider Figure 1d, which shows a small representative circuit composed of eight cells (six inverters, one 2-input nand gate, and one 4-input nand gate). Special considerations must be made when enabling NTC for optimizing the circuit under conflicting objectives, such as delay and power. Figure 1a and 1b presents delay vs. leakage power (log scale), when performing four separate of individual cell V_{th} modifications on N1 and INV5, independently. A plot of 1000 generated circuit instances is provided against each V_{th} circuit configuration with delay variations following a normal PV model.

As shown in Figure 1a, the performance (delay) variability is significant when all cells are at a NTC setting, 200 ps vs 50 ps for non-NTC. However, the gains in leakage power

978-1-4799-1170-7/13/$31.00

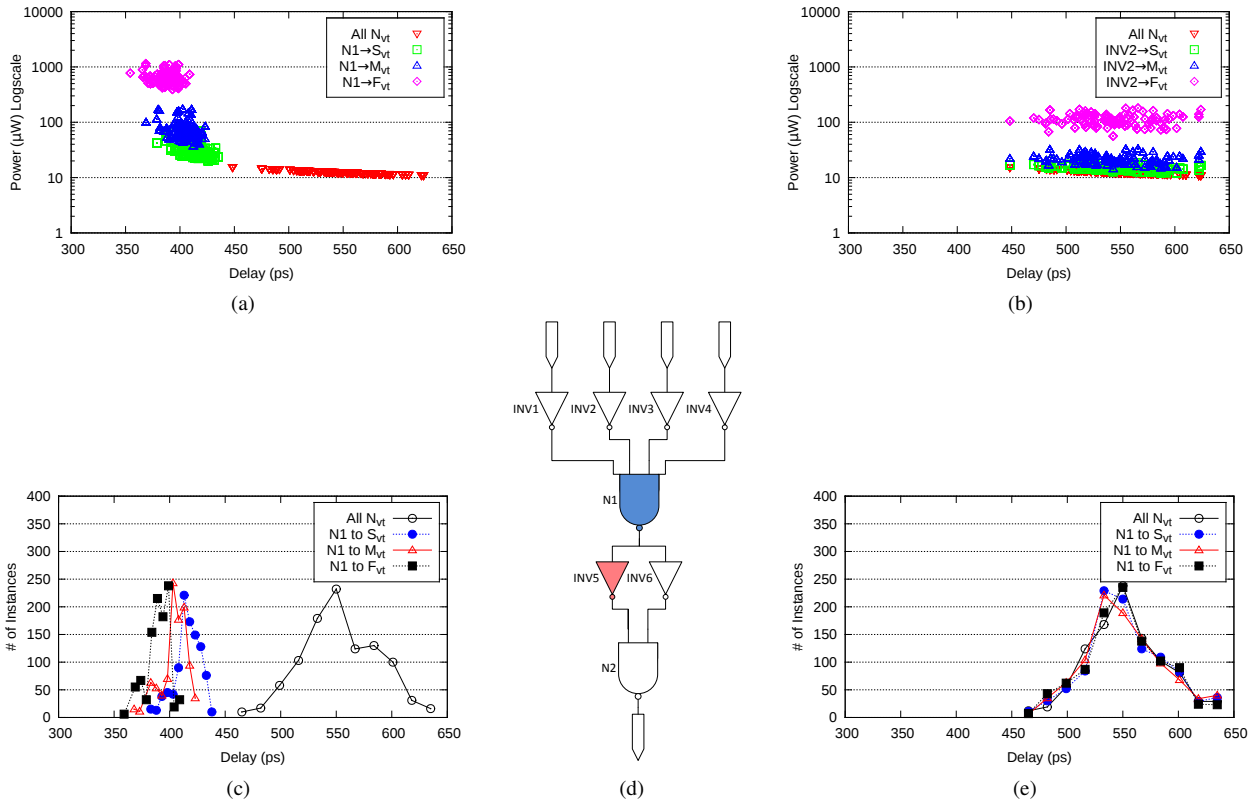


Fig. 1: Example circuit with inverters (INV1 to INV6) and nand gates (N1 and N2): (a, b) distribution of 1000 circuit instances and achieved delay when decreasing V_t starting from $N_{vt} \rightarrow \{S_{vt}, M_{vt}, F_{vt}\}$, selectively for gate N1 (a) and gate INV5 (b); (c, e) circuit instance distribution vs circuit delays when performing V_{th} adjustments on gates N1 and INV5 (d).

reductions are significant (exponential). Thus, selecting cells with lower V_{th} addresses performance variation in a design, at the cost of exponential leakage power overhead. Selecting the proper cell to be altered, however, requires identifying cells which maximally reduces variation. For instance, altering INV5 provides no additional benefit in addressing the performance variability, as paths that pass through INV6 are still affected by PV (Figure 1b). In contrast, altering N1 minimizes the delay variations from the four inputs (INV1 to INV4), and outputs to INV5 and INV6, simultaneously. Therefore, when attempting to maximize yield, it is key to identify cells that participate on many paths in order to suppress performance variation in a design, as shown in Figures 1c over 1e.

The significant variations that are more susceptible in an NTC-enabled design motivate the requirement of handling other uncertainty factors, such as dealing with spatial correlations. For example, due to an arbitrary model with complex correlations, identifying the cells that maximally suppress variation cannot be modeled through standard statistical approaches. In order to address the issue, a scenario-based approach is utilized to generate instances to optimize based on a given PV model.

III. Related Work

A. Process Variation

Worst case analysis (WCA) is widely used in industry to deal with the impact of PV [6]. WCA considers the worst-case parameter values due to PV, environmental, and aging effects [33][34]. WCA is used in both verification and design processes. In the verification process, WCA is used to verify the target IC against the specification in the worst case, which is helpful in reducing technical risk and improving system reliability. However, in the design process, WCA produces overly conservative designs that over-estimate the impact of PV. Such over-estimation may complicate the design process and, more importantly, result in unnecessary performance degradation.

In order to solve the WCA issues, researchers have been promoting statistical circuit modeling in IC design and analysis [5]. The goal of statistical modeling is to search for alternatives to WCA that provide more accurate representations of PV. As the demand on performance, power, and density continues to increase in modern IC design, statistical modeling plays an increasingly important role in achieving greater gains compared to worst-case design.

Recent work has focused on the manifestation properties of an IC under the impact of PV. Sarangi et al. propose a timing error model resulting from systematic and random PV effects [2]. In [7], an analysis of the leakage power distribution due to PV is given and used to predict the CDF/PDF of the total chip leakage. Alkabani et al. propose an approach for post-silicon leakage power reduction through input vector control (IVC) that takes into account PV [8]. Wei et al. have developed gate-level characterization techniques for quantifying PV effects in addressing hardware security [35]-[39].

B. Scenario-based Analysis

Scenario-based approaches [9][10][11] have been used to solve design optimization problems with uncertain constraints, referred to as chance-constrained problems. The main idea is to use a sampling of the constraints to approximate the infinite space of variable constraints caused by the variations. Calafiore et al. prove that the solution of the scenario problem maps approximately to the original problem with uncertain constraints; this work also provides an explicit bound on the number of samples that are needed to obtain a specified levels of robustness [9]. [10] obtains a convex approximation to the chance-constrained problem and extends it to an ambiguous set of constraints, where the random distribution of the variations belongs to a convex compact set instead of a fixed distribution.

C. Gate Sizing

Gate sizing has been a crucial task for accomplishing simultaneous optimization of delay, power, and area since the very early beginnings of CAD [12]. In the mid-80s, Fishburn and Dunlop proposed a provably optimal approach to transistor sizing [4]. It proposes an optimal gate sizing scheme to meet the delay constraint by using convex programming, but it does not consider the impact of process variation. More recently, there have been several efforts to optimize manifestational characteristics of ICs for a design of interest in the presence of PV using gate sizing. Zhu et al. proposes a gate sizing and clustering approach to optimize leakage energy adaptive body bias, which takes into consideration the process variations in different instances of ICs [13]. [14] discusses a gate-sizing algorithm to minimize the number of failing chips considering process variation. It compares the variation-aware design with the worst-case approaches and confirms the gain obtained from the former. [15] proposes a geometric programming-based heuristic approach to gate sizing. More recently, researchers from Intel have held yearly discrete cell sizing contests to expose the challenges encountered in an industrial flow [26]. However, only leakage power is accounted with no PV model assumed. Additional improvements when accurate operating conditions such as temperature, gate switching, and input vector state leakage computations are accounted for[31][32], however, PV is not considered in their models as well.

Other works use statistics-based approaches to capture the uncertainty stemming from PV [16][17][18]. The spatial correlations in L_{eff} inherent in the PV model, however, are too complex to be captured by simple statistical models. Our scenario-based approach is simple, generic, and flexible; it can be applied to any number of optimization tasks because it does not rely on assumptions about the uncertainty model. Instead, the uncertainty model is used to generate scenarios to be used as a training set for optimization, with the idea that if the training scenarios are representative, then the optimization will work well for any set of instances. Calafiore and Campi prove that for convex programs the scenario approach provides a solution that satisfies most constraints with high probability given enough samples, establishing a theoretical bound [9]. Therefore, the scenario approach has the same theoretical underpinnings as traditional convex optimization methods. However, our main contribution is to extend the scenario approach to essentially optimize known NP-hard problems (e.g. discrete gate sizing [28]) well, by combining it with generic optimization techniques (e.g. iterative improvement) and statistical analysis. Although a mathematical result can no longer be theoretically proven in this case, a statistical interval of confidence for various sample set sizes can be established.

IV. Power and Delay

We use the delay and power models proposed in [19] in our design process. These models connect gate-level delay and power properties with the physical level properties such as gate width (W), gate length (L), and threshold voltage (V_{th}). Equation (1) shows the gate-level delay model [19], with supply voltage V_{dd}, subthreshold slope n, mobility μ, oxide capacitance C_{ox}, gate width W, gate length L, thermal voltage $\phi_t = (kT/q)$, DIBL factor σ, threshold voltage V_{th}, and delay and model fitting parameters k_{tp} and k_{fit}. Load capacitance C_L is proportional to the sum of the interconnect and driving pin capacitance's of its fan-out gates. Thus, assuming the interconnect to be fixed, the gate propagation delay (t_p) is affected by the sizing of its immediate fan-out cell(s), requiring careful sizing options to be selected during the sizing procedure.

$$t_p = \frac{k_{tp} \cdot C_L \cdot V_{dd}}{2 \cdot n \cdot \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot (\frac{kT}{q})^2} \cdot \frac{k_{fit}}{(ln(e^{\frac{(1+\sigma)V_{dd}-V_{th}}{2 \cdot n \cdot (kT/q)}} + 1))^2} \quad (1)$$

Note that the parameters other than W, L, and V_{th} are transistor-level properties that can be derived using transistor-level simulation. Since they are not affected by PV and therefore do not impact the design process, we assume they are constant values in the model. Equation (2) describes the leakage power model that depends on the value of W/L.

$$P_{leakage} = 2 \cdot n \cdot \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot (\frac{kT}{q})^2 \cdot V_{dd} \cdot e^{\frac{\sigma \cdot V_{dd} - V_{th}}{n \cdot (kT/q)}} \quad (2)$$

The gate-level switching power model [19] is described by Equation (3), where α is the activity factor and f is the frequency.

$$P_{switching} = \alpha \cdot C_L \cdot V_{dd}^2 \cdot f \quad (3)$$

The delay and power models indicate the trade-off between delay and power in terms of gate sizing. For example, increasing W of a gate reduces its delay but increases its leakage and switching powers. Simultaneously, the decision increases the delays and switching powers of the *input* gates of the directly affected gate. It is our goal in this paper to determine an optimized set of gate widths and V_{th}'s to meet specified delay and power requirements.

A. Process Variation

Several models have been proposed to capture the impact of PV [20][21][22], which formulate ΔL as a random distribution or a combination of multiple distributions to reflect the spatial correlation on a chip as well as the inter-chip variations. We follow the quad-tree model proposed by Cline et al. [22], which considers the spatial correlations among gates. In the quad-tree model, the gate-level property (e.g. L) subject to PV is

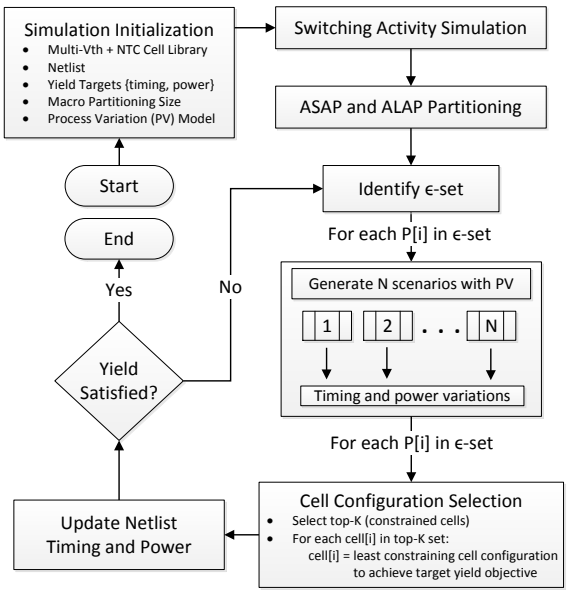


Fig. 2: Process variation-aware NTC-enabled Design Flow.

distributed into multiple levels, with a different number of grids allocated on each level. The grids on each level are assigned variation values that follow a normal distribution. We calculate the total value of the target gate-level property as the sum of the variations on each level of the grids to which the corresponding gate belongs.

The quad-tree model for L is shown in Eq. (4), where ΔL_{ij} is the quantitative variation of the i-th level and j-th grid to which the gate belongs; μ_i and σ_i are parameters of the normal distribution at level i.

$$\Delta L = \sum_i \Delta L_{ij}, \qquad where \quad \Delta L_{ij} \sim N(\mu_i, \sigma_i) \quad (4)$$

We adopt the Gaussian distribution proposed by Asenov et. al. [23] in regards to V_{th} variation, which is based on the simulation of random dopant distribution.

V. Technical Approach

Figure 2 highlights the several steps in our NTC-enabled PV-aware yield optimization framework. Each step is discussed in the next following subsections.

A. Cell Switching Activity

As the gap between V_{th} and V_{dd} is reduced in NTC (Eq. 2), switching power starts to become the dominant power (Eq. 3). Therefore, accurate knowledge in determining which cells have high switching activity (SA), is crucial for maximizing energy efficiency for a given design, especially for NTC-enabled designs. As a pre-processing step, we perform gate-level event simulation by applying a set of 100K randomly generated input vectors to the primary inputs of a given circuit and record the SA for each respective net (wire) of the design. Accurate input stimulus from actual applications may be used instead to improve accuracy. Only the SA of nets are recorded for computing switching activity since each gate in our studied benchmarks is driven by only one net. Therefore, the total switching power for a given design can be computed as the sum of all net switching power using Eq. 3, where the C_L = C_{net}+C_{pin}. Here, C_{net} is the interconnect and C_{pin} is the sum of driven outgoing pin capacitance's.

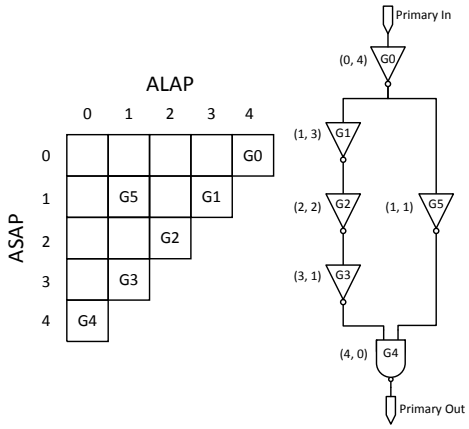


Fig. 3: *As soon as possible* (ASAP) and *as late as possible* (ALAP) gate logic mapping.

B. Logic-depth Indexing

We employ an *as-soon-as-possible* (ASAP) and *as-late-as-possible* (ALAP) cell indexing principle for partitioning a circuit into distinct groups. The ASAP index represents the maximum number of logic stages an input signal is required to propagate (primary input source) before it is considered valid at its output pin. The ASAP index for a particular cell is the max ASAP value among its inputs. The ALAP index represents the maximum number of stages the output signal of a cell must propagate before reaching the inputs of all its transitive primary outputs. As a result, cells in a given design can simply be indexed (grouped) by its respective ASAP and ALAP index (Figure 3).

The ASAP and ALAP index values can be used to provide useful structural knowledge of a given circuit, as well as properties useful for constructing circuit partitions. Figure 3 illustrates a simple circuit comprised of six gates. The right figure shows the two possible paths that the output signal from G0 is required to propagate (P1 = {G0, G1, G2, G3, G4} and P2 = {G0, G5, G4}) before in route to the primary output. The left figure shows a triangular-matrix with each box containing the cell(s) with matching the ASAP and ALAP indexes. The max depth is 5 (0 to 4) and can be represented by the P1 (or the diagonal-edge of triangular-matrix), which is more likely to form the critical path over P2 since the number of stages that a signal must propagate (5 vs 3) is larger. It is important to note, however, that although P2 is less critical in terms of depth than P1, the ASAP and ALAP indexing scheme cannot guarantee that P2 is less critical (e.g., delay) than P1. However, we foresee that under the most general case, the ASAP and ALAP indexing can provide key insight in identifying which cells are more likely to participate in the critical paths, thus, enabling efficient power and delay trade-offs to be performed during optimization.

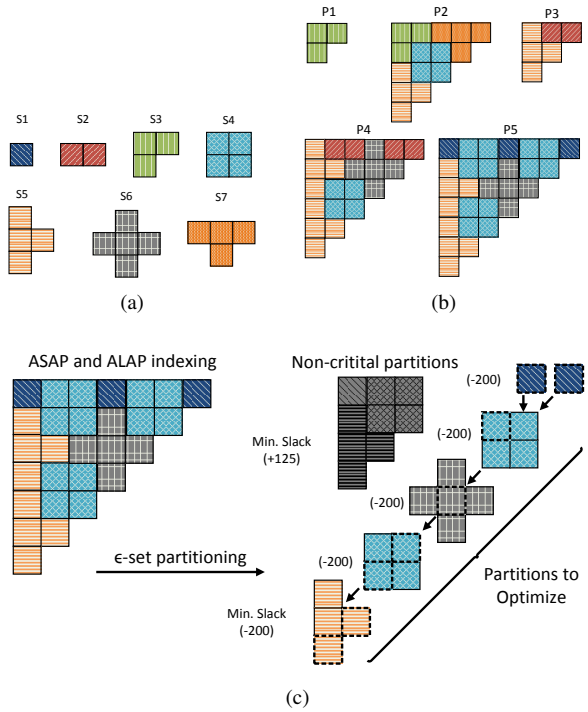


Fig. 4: ASAP and ALAP circuit partitioning : (a) 7 enabled micro-partitions; (b) 5 example macro-partitions using micro-partitions; c) ϵ-set of macro-partitions satisfying ϵ-delta slack.

C. Circuit Partitioning

A major challenge in circuit partitioning is identifying the metrics to group cells by as well as the group size. We define the following objectives in establishing our circuit partitioning scheme: 1) partitions are configured such that each independent optimization captures and maintains the global optimization picture accurately; 2) partition size is kept small (e.g., hundreds of gates) to improve simulation time, reducing the number of computations normally required when performing sensitivity analysis (e.g., power vs delay trade-off) across the entire circuit; and 3) multi-threaded support by optimizing partitions independently across a set of available threads. We group cells with similar delay affinities by using their respective ASAP and ALAP indexes as a partitioning mechanism. Our ASAP and ALAP grouping analysis showed that most circuits provide enough sparseness in group sizes, with most individual ASAP and ALAP groups ranging from less than 1% to 5% for a given circuit, translating to few tens to few thousands of cells.

We define seven micro-partitions in defining the building blocks for circuit partitions (Figure 4a), which can be used to form larger macro-partitions of larger sizes (Figure 4b). The partition shapes are chosen carefully such that they any circuit depth (ASAP, ALAP) can be constructed by the elementary micro-partitions. Each micro-or macro-partition can be independently treated as its own circuit. However, it is important to maintain coherency across its transitively affected input and output groups; this can be achieved by preserving the cell configurations of connecting inputs and outputs to the group. For example, the cell drivers of a partition can be treated as the primary input virtual drivers with fixed configurations, and the load capacitance driven by its primary outputs as virtual loads. Delay and power coherency is maintained by invoking a circuit update procedure after each global optimization phase. The update procedure updates affected partitions during the last iteration with the latest configurations of any fan-in/fan-out neighbors (e.g., input drivers and output loads).

ASAP and ALAP indexing also provide a useful property that can be utilized for constructing circuit partitions composed of only critical partitions. A dependency graph may be formed by connecting macro-partitions (from primary-in to primary out) via ASAP and ALAP indexing. For example, a given partition at ASAP stage i and ALAP stage j can only have input sources from ASAP and ALAP indexes who's index satisfies $\leq i-1$ and $\geq j+1$, respectively. For instance, referring to Figure 3, cell G3 cannot have a source from G5 since it violates this property. This property useful for constructing ϵ-sets, explained later, which are composed of macro-groups that are within an ϵ-delta slack (Figure 4c).

D. Cell Configuration Selection

We adopt a maximally constrained, minimally constraining heuristic for selecting cell configurations. The maximally constrained principle states that the configurations for the most difficult cells (or groups) are determined early in the optimization while there is still slack in the design with respect to the desired objectives (e.g., timing and power). The minimally constraining principle states that the configuration is chosen such that it minimally impacts the optimization search space (e.g., available future moves) [29].

1) Identifying Critical Partitions: We first compute the accurate delay, slack, and power of the entire circuit *without PV* and identify partitions that are within a predefined ϵ-delta of the critical slack. These cell ASAP and ALAP micro-partitions are used to form an ϵ-set of macro-partitions with paths leading from the primary inputs to primary outputs (Figure 4c). The next step is to identify partitions which are more likely to be impacted by process variation. This can be achieved by generating 100 circuit scenarios (instances) to quantify the impacted of PV in regards delay and power yields. In our experiments, we found that the number of iterations can be kept small if the number of cells per group is small with high-confidence. With this method, the variation of a macro-partition can be quantified by the following equations:

$$P^j_{var} = \sum_{i}^{|P^j|} Cell^i_{var} \tag{5}$$

$$Cell^i_{var} = \sum_{k}^{|F_{in}|} Std_i \cdot Slk_i \tag{6}$$

The delay variation of a partition is the sum of the partitions primary input/output delay cell variation ($Cell_{var}$) for each cell $i = 1$ to $|P^j|$, where $|P^j|$ represents the number of cells in partition j (Eq. 5). The delay variation of a cell is simply the sum of the products between each input pin $k = 1$ to $|F_{in}|$ and its corresponding slack Slk_i. Note that only computing the $Cell_{var}$ of a partitions primary inputs/outputs is necessary since they connect to paths to other fan-in/fan-out partitions, effectively preserving the variational impact on the global circuit delay and power.

In order to reduce the number of required computations, only groups included in ϵ-set are considered during an optimization phase, since the rest are considered non-critical in terms of achieving the desired yield objective (e.g., delay and/or power). The ϵ parameter may be adjusted to consider more groups, enabling more partitions to be optimized simultaneously towards a desired objective.

2) Quantifying Group and Cell Difficulty: Given ϵ partitions, the next step is to identify the top K-difficult cells within a partition. This is quantified by the formulation below:

$$Cell_{diff} = |F_{out}| \cdot |F_{in}| \cdot (-Slk) \quad (7)$$

$|F_{out}|$ and $|F_{in}|$ represents the fan-out and fan-in length of cell, and Slk the slack of a given cell and represents the minimum Slk value across a cell's input pins. We, therefore, identify the most difficult gates within a particular group as the top-K cells via max($Cell_{diff}^{i=1\ to\ |P_j|}$), where $|P_j|$ represents the number of cells in partition j. We use Eq. 7 to model an individual cell's difficulty, since the number of paths potentially affected by a given cell is attributed by the number of signals that must propagate through it, accounted by the number of fan-in and fan-out connections. Slk is used in order to identify cells within a group that are likely to participate in the critical paths.

Given the top-K most difficult cells within a partition, the last step is to determine the cell configuration or move (e.g., size, V_{th}) with an incremental/decremental offset of *one* that benefits the target objective. As an example, assuming under delay-constrained optimization, we can classify the result of a considering a valid move for each cell into three categories:

i) Power and delay reduction.
ii) Power reduction and constant delay.
iii) Power reduction and delay increase.

It is important to account only for valid moves; moves that lead to load or slew violations are disregarded. Valid moves are then assigned priorities in the precedence class order of i, ii, and iii. Moves that benefit both power and delay (class i) are always selected over moves belonging in classes ii and iii, and are compared against other moves within its own class as the product of power and delay savings. If no class i moves exists, then class ii moves are selected by the maximum total power improvement. If only class iii moves are found, the move that produced the maximum $\frac{benefit}{cost}$ is selected. The above objective concepts may be applied inversely when a power-constrained delay minimization objective is set.

To prevent the algorithm from being stuck in a local minima, only K-cells are configured during an optimization iteration. Based from our experiments, we set K=10% of the total group cell count, which can be adjusted to affect convergence rate. Once a cell configuration is chosen, it is locked and cannot be altered until the completion of an optimization iteration. An optimization iteration is complete once partitions belonging in the ϵ-set have been visited. Optimization iterations are repeated until a convergence criteria is satisfied (Figure 2).

VI. Simulation Setup

We evaluate our approach using industrial benchmarks included in the ISPD Design Contest 2012 suite [26]. Each design was optimized in accordance to industrial imposed constraints, satisfying as max load capacitance and input slew limits. Power and timing results were generated using an in-house timer implemented in C++, which we correlated in good spirit against Synopsis PrimeTime to be within 1e-3 error. We extend the original cell library to support near-threshold cells (N_{VT}), which were generated using analytical models from Markovic et al. [19]. A variation factor ($3\delta/\mu$) of 30% was used to model the variation of a standard inverter.

TABLE I: Target clock (delay) for each benchmark (col. 2); the achieved delays (1000 instances) when considering PV (col. 3); and adjusted target clock with PV for using [30] (col. 4).

Circuit	Target Clock (ps)	Avg. !NTC-PV	Adj. !NTC-PV
dma	1800	1878	1500
pci_bridge32	1400	1626	1200
vga_lcd	1400	1689	1250
des_perf	1600	1754	1450
b19	4300	4546	3800

We perform Monte Carlo simulations to obtain the best fitting parameters that minimized error against the standard cell library. We obtain fitting parameters independently for both delay and leakage with respect to each cell table entry: 1) delay type {rise, fall} per {delay, input transition}; and 2) capacitance and input {rise, fall} transition index. The fit was performed across the three V_{th} (S_{vt} = 0.33V, M_{vt} = 0.27V, and F_{vt} = 0.20V), and V_{dd} = 0.70V [26]. The final model resulted with error less than 5% per cell delay (e.g., rise/fall input transitions), and less than 1% in leakage. The fitting parameters were used to generate an NTC cell library configured with N_{vt} = 0.68V. Note that the corresponding cell library is no longer used when applying PV factors into the design since the affected factors we consider in our PV-model (Section IV-A). The obtained fitting parameters are applied directly during the delay and leakage table look-up. To validate our approach, we simulate each design against the look-up-table model and achieved power and timing errors within 8% to the reference industrial tool using the original and NTC-generated cell library for the circuits we consider here.

VII. Experimental Results

We compare our approach NTC against an Non-NTC (!NTC) multi-V_{th} gate-sizing method proposed by Li et al. [30]. Their approach achieved competitive results against solutions obtained from the ISPD 2012 design contest [26] and [27]. Due to a NTC cell library compatibility issues with their tool, we only compare their method using the original ISPD multi-V_{th} library (non-NTC library). Additionally, circuits *leon3mp* and *netcard* are also omitted due to tool issues.

Due to performance impact that is incurred when enabling NTC, for timing comparisons, we relax the target delay constraints (2X slower) used in the original ISPD design contest suite (Table I-col. 2). To ensure fair timing and power analysis when considering PV, we used the our in-house timer to report timing and power results under PV (see Section VI) for solutions obtained by both methods. Due to space constraints, we omit reporting detailed simulation run-times of our approach, but note that our approach achieved 1.2X to 3.5X run-time increase over the compared method in [30]. This is expected since our approach requires additional computational overhead for performing tasks described in Section V.

TABLE II: Average (avg.), max (+), min(-) delays when optimizing under NTC-enabled (NTC) non-NTC-enabled (!NTC) [30].

Circuit	NTC				!NTC			
	Avg.	(+)	(-)	Std.	Avg.	(+)	(-)	Std.
dma	1475	1791	1403	48.07	1728	1780	1701	12.6
pci_bridge	1358	1583	1307	31.0	1479	1497	1458	7.16
vga_lcd	1212	1399	1150	40.8	1377	1399	1353	11.7
des_perf	1449	1554	1406	23.5	1475	1504	1450	11.1
b19	4167	4273	4059	36.4	3902	3940	3865	14.7

To identify equivalent timing and power constraints for fair comparisons, we first generate solutions using the method reported in [30] to achieve reference clock targets shown in Table I. Next, the obtained solution is fed into our PV-aware timer to obtain the actual PV-enabled delay and power results. Timing violations are expected, as shown by non-PV-aware result, which are on average 8% (20% max) slower than the original intended target delay. Therefore, new target clocks are determined that achieves 100% yield (Table I-col. 4) under the non-NTC method. Experimentally, we found that the original target clocks were required to be scaled lower by 12.1% on average.

The non-NTC result for each circuit is considered as the base comparison assuming all circuits (1000 instances) that achieve 100% yield in timing and power. We use this as the target yield objective for our NTC approach, since setting all cells to non-NTC configurations would naturally minimize varation in a design. Therefore, the objective of our approach is to optimize each benchmark using our PV-aware NTC-enabled framework to attain similar yields to that of the non-NTC approach for the given reference clock targets.

A. NTC-enabled PV-aware Optimization

Table II presents the actual timing achieved by NTC and non-NTC. As shown, the achieved standard deviations by NTC were found to be up to 4.3X (3.2X) larger than the achieved standard deviation from the non-NTC delay results. Thus, in order to achieve 100% yield targets when enabling NTC-design, larger guard bands for timing and power should be enforced. To understand how delay is affected accross a large set of scenarios, Figure 5 presents the achieved max, mean, and min delays normalized to their respective reference clocks in Table III. Clearly, non-NTC results achieves significantly lower performance variation over NTC. However, it is important to note that the mean result for each benchmark is closer to the minimum result. Thus, this means that the majority of circuit instances lie closer to the minimum result.

Table III compares the total power (avg., max, min) for both methods. The results were acquired from 1000 generated circuit instances using our PV model. As shown, NTC achieve significant total power reduction of up to 4.42X (3.30X avg.) over the non-NTC method. The power improvements result from the savings in leakage power, achieving 37.4X max reduction (24.1X avg.) over non-NTC. To achieve this under NTC, more cells have to be up-sized to larger cell configurations in order to meet timing constraints, thus, increasing switching power up to 1.55X (1.24X avg.). However, the total power is still reduced using the NTC approach due to the dominant leakage power component. For example, under the non-NTC approach, leakage power made up 77% (avg.) of the total power budget vs. 12.1% (avg.) for the NTC.

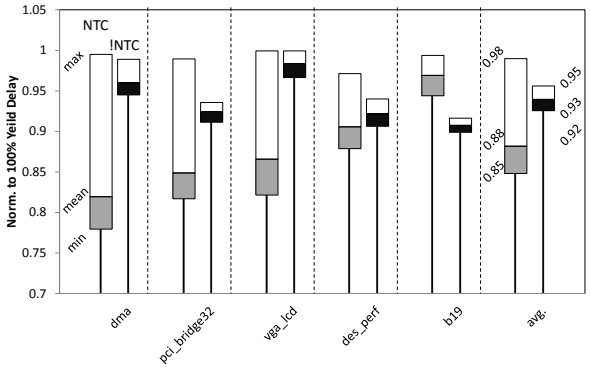
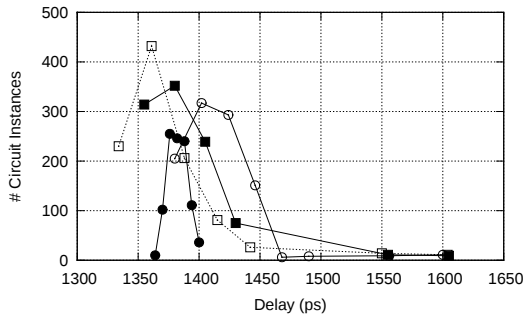
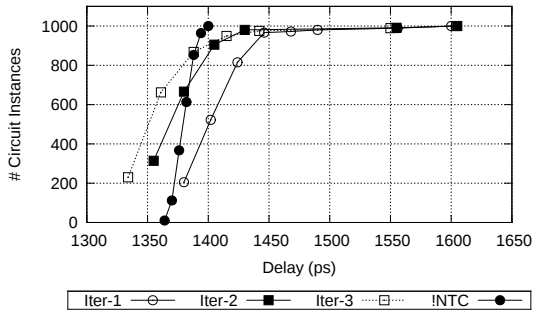


Fig. 5: Max (top-tier), mean (mid-tier), and min (bottom-tier) target delay ratios among 1000 generated instances with respect to NTC (left) and !NTC (right) [30].



(a)



(b)

Fig. 6: Timing yield optimization for circuit pci_bridge32: (a) frequency distribution graph compares 3 iterations using our NTC approach against a !NTC ; (b) cumulative dist. graph.

Figures 6a-6b indicate how delay target yields are achieved through successive iterations using our approach. Additional iterations show improvements in both the number of circuit instances that satisfy the target clock of 1400 ps (100% for !NTC). A 90% yield is achieved after performing 3 design iterations, achieving an overal 3.13X (avg.) total power reduction for all valid circuit instances.

VIII. Conclusion

We have presented a framework for maximizing performance and power yield constraints for near-threshold

TABLE III: Total power results when optimizing under NTC-enabled (NTC) and non-NTC (!NTC) [30]: Shown are the average (avg.), max (+), and min(-), results corresponding to total, leakage, and switching power values. The ratio $\frac{!NTC}{NTC}$ represents the total power reduction factor.

Circuit	Comb. Cells	Total Power (mW)						Ratio	Leakage Power (mW)							Switching Power (mW)					
		NTC			!NTC				NTC			!NTC				NTC			!NTC		
		avg.	(+)	(-)	avg.	(+)	(-)	avg.	avg.	(+)	(-)	avg.	(+)	(-)	avg.	(+)	(-)	avg.	(+)	(-)	avg.	(+)	(-)
dma	23109	44	46	43	138	141	135	**3.13**	8.2	9.5	7.3	115	118	110	35.8	36	35.7	23	23	25			
pci_bridge32	29844	36	38	35	113	115	110	**3.13**	5	6.7	4.1	87	88	84	31	31	30	26	27	26			
vga_lcd	147812	115	118	110	509	529	485	**4.42**	16.1	17.2	15.5	420	430	400	98	100	94	89	99	85			
des_perf	102427	175	200	164	533	554	526	**3.02**	10.2	12.6	8.7	381	402	330	164	187	155	152	152	196			
b19	212674	260	297	251	733	762	722	**2.81**	21.4	23.1	20.2	546	550	542	238	273	230	187	212	180			

computing-enabled designs, under the presence of process varation (PV). We apply an NTC-enabled paradigm using popular optimization techniques, such as gate-sizing and multi-threshold cell assignment, while strictly adhering to industrial imposed design constraints, such as load and slew limits. The focal point of our framework is a logic-depth circuit-level partitioning scheme for efficiently characterizing and identifiying critical circuit sections related to timing and power variations of a given circuit. We utilize a scenario-based approach by generating a large set of circuit instances, which is used to identify which cell configurations in an NTC search space maximally benefit solution towards the target yield objectives.

We compare our approach against a state-of-the-art non-NTC approach and show significant reductions in total power of up to 4.4X (3.3X avg) for the same timing and power yield constraints. Consequently, the savings in power results with signficant performance varations that should be addressed when optimizing for maximal yield. The standard deviation in performance for NTC-enabled designs are shown to be 4.3X max (3.2X avg.) than that of non-NTC approach. However, we show that on average, only a few circuits (less than 5%) make up instances that violate imposed timing yield constraints for NTC-enabled designs.

REFERENCES

[1] S. Borkar et al., "Parameter variations and impact on circuits and microarchitecture," *DAC*, 338-342, 2003.

[2] S. Sarangi et al., "VARIUS: a model of process variation and resulting timing errors for microarchitects," *IEEE T-SM*, 3-13, 2008.

[3] B. E. Stine et al., "Analysis and decomposition of spatial variation in integrated circuit processes and devices," *IEEE T-SM*, 24-41, 1997.

[4] J. Fishburn et al., "TILOS: a posynomial approach to transistor sizing," *ICCAD*, 326-328, 1985.

[5] S. Duvall, "Statistical circuit modeling and optimization," *IEEE International Workshop on Statistical Metrology*, 56-63, 2000.

[6] R. Kendall, "Worst case analysis methods for electronic circuits and systems to reduce technical risk and improve system reliability," White Paper, *Intuitive Research and Technology Corporation*, http://www.irtc-hq.com/WCA_white_paper_gen.pdf

[7] H. Chang et al., "Full-chip analysis of leakage power Under process variations, including spatial correlations," *DAC*, 523-528, 2005.

[8] Y. Alkabani et al., "Input vector control for post-silicon leakage current minimization in the presence of manufacturing variability," *DAC*, 606-609, 2008.

[9] G. Calafiore et al., "The scenario approach to robust control design," *IEEE T-AC*, 742-753, 2006.

[10] A. Nemirovski et al., "Convex approximations of change constrained programs," *SIAM*, 969-996, 2006.

[11] J. Luedtke et al., "A sample approximation approach for optimization with probabilistic constraints," *SIOPT*, 674-699, 2008.

[12] A. Ruehli et al., "Power and timing optimization of large digital systems," *ISCAS*, 402-405, 1976.

[13] C. Zhuo et al., "Variation-aware gate sizing and clustering for post-silicon optimized circuits," *ISLPED*, 105-110, 2008.

[14] A. Davoodi et al.,"Variability driven gate sizing for binning yield optimization," *VLSI*, 683-692, 2008.

[15] D. Patil et al., "A new method for design of robust digital circuits," *ISQED*, 676-681, 2005.

[16] M. R. Guthaus et al., "Gate sizing using incremental parametrized statistical timing analysis," *ICCAD*, 1026-1033, 2005.

[17] E. T. A. F. Jacobs et al., "Gate sizing using a statistical delay model," *DATE*, 283-290, 2002.

[18] J. Cong et al., "Robust gate sizing via mean excess delay minimization," *ISPD*, 10-14, 2008.

[19] D. Markovic et al., "Ultralow-power design in near-threshold region," *IEEE*, 237-252, 2010.

[20] B. Cheng et al., "Evaluation of statistical technology generation LSTP MOSFETs," *Solid-State Electronics*, 767-772, 2009.

[21] S. Roy et al., "Where do the dopants go?" *Science*, 388-390, 2005.

[22] B. Cline et al., "Analysis and modeling of CD variation for statistical static timing," *ICCAD*, 60-66, 2006.

[23] A. Asenov, "Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 um MOSFET's: a 3-D atomistic simulation study," *IEEE T-ED*, 2505-2513, 1998.

[24] T. Hastie et al., *The Elements of Statistical Learning.* Springer, 2001.

[25] R.G. Dreslinski et al., "Near-threshold computing: reclaiming Moore's Law through energy efficient integrated circuits," *IEEE*, 253-266, 2010.

[26] M. M. Ozdal et al., "The ISPD-2012 discrete cell sizing contest and benchmark Suite," *ISPD*, 161-164, 2012.

[27] J. Hu et al., "Sensitivity-guided metaheuristics for accurate discrete gate sizing," *ICCAD*, 233-239, 2012.

[28] W. N. Li, "Strongly NP-hard discrete gate-sizing problems," *ICCD*, 1045-1051, 1993.

[29] Z. Zhang et al., "Gradual relaxation techniques with applications to behavioral synthesis," *ICCAD*, 529-536, 2003.

[30] L. Li et al., "An efficient algorithm for library-based cell-type selection in high-performance low-power designs," *ICCAD*, 226-232, 2012.

[31] N. A. Conos et al., "Gate sizing in the presence of gate switching activity and input vector control," *VLSI-SOC*, 2013.

[32] N. A. Conos et al., "A temperature-aware synthesis technique for simultaneous delay and leakage optimization," *ICCD*, 2013.

[33] J. B. Wendt et al., "Improving energy efficiency in sensing subsystems via near-threshold computing and device aging," *IEEE Sensors*, 2013.

[34] S. Wei et al., "Low power FPGA design using post-silicon device aging," *FPGA*, 2013.

[35] S. Wei et al., "Gate-level characterization: foundations and hardware security applications," *DAC*, 222-227, 2010.

[36] S. Wei et al., "Integrated circuit digital rights management techniques using physical level characterization," *DRM*, 3-14, 2011.

[37] S. Wei et al., "Malicious circuitry detection using thermal conditioning," *TIFS*, 1136-1145, 2011.

[38] S. Wei et al.,"Scalable hardware trojan diagnosis," *VLSI*, 1049-1057, 2012.

[39] S. Wei et al., "Gate characterization using singular value decomposition: foundations and applications," *TIFS*, 765-773, 2012.