Variability-Aware Architecture Level Optimization Techniques for Robust Nanoscale Chip Design

Saraju P. Mohanty[a,c,3,*], Mahadevan Gomathisankaran[b,c,4], Elias Kougianos[a,d,5]

[a]*NanoSystem Design Laboratory (NSDL), Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA.*
[b]*Trusted Secure Systems Laboratory (TSSL), Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA*
[c]*Department of Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA.*
[d]*Department of Engineering Technology, University of North Texas, Denton, TX 76207, USA.*

Abstract

The design space for nanoscale CMOS circuits is vast, with multiple dimensions corresponding to process variability, leakage, power, thermal, reliability, security, and yield considerations. These design issues in the form of either objectives or constraints can be handled at various levels of digital design abstraction, such as architectural, logic and transistor. At the architectural level (a.k.a. Register-Transfer Level, RTL), there is a balanced degree of freedom for fast design exploration by exploring various values of design parameters. Correct design decisions at an early phase of the design cycle ensure that design errors are not propagated to lower levels of circuit abstraction, where it is costly to correct them. Moreover, design optimization at higher levels of abstraction provides a convenient way to deal with design complexity, facilitates design verification, and increases design reuse through intellectual property (IP) cores.

To achieve power-performance trade-offs, different architectural-level techniques have been proposed in the existing literature. This paper will briefly discuss selected RTL techniques which account for process variation. These existing approaches handle the optimization of different power components independently but do not effectively account for the inherent variation of process and design parameters. Thus, in this paper, a novel process variation aware statistical RTL optimization approach is presented. Assuming dual values of T_{ox}, V_{th}, and V_{DD}, gate-oxide leakage, subthreshold leakage, dynamic power, and performance are estimated for architectural units. Statistical variations in the parameters (T_{gate}, V_{th}, V_{DD}, and L_{eff}), are explicitly taken into account by using Monte Carlo simulations while characterizing the architectural units. The proportion of values of gate-oxide and subthreshold leakage and dynamic power in the total power consumption of these units is then analyzed. This analysis in essence gives a relative and integrated perspective of various power-performance tradeoffs against the baseline case, thus serving as a guideline to help designers make appropriate decisions. Experiments on several benchmarks show a significant reduction in gate-oxide and subthreshold leakage, dynamic, and total power.

Keywords: Nanoscale CMOS (Nano-CMOS); Process Variation; Architectural Level; Register-Transfer Level (RTL); Leakage Dissipation; Power Optimization

1. Introduction

Consumer electronic systems such as mobile smart phones, media players, high-definition television (HDTV), health monitoring devices, and various sensors have a profound impact on society. The Integrated Circuit (IC) is the main workhorse in consumer electronics [1]. Efficient design of ICs is one key driving factor for their omnipresence, from kitchens to spacecrafts. To meet the growth of ICs, the industry has resorted to aggressive device scaling [2]. It is estimated that one billion transistors per person were manufactured in 2010! Scaling has essentially provided the following advantages: (1) Reduced the cost of computing as a larger number of transistors are being packed in the same area. (2) Reduced the per transistor manufacturing cost. (3) Reduced power dissipation per transistor as smaller transistors need lower operating voltages. However, the overall power dissipation is still a big issue along with the leakage mechanisms. (4) Initiated the multicore era for high performance

*Corresponding author
Email addresses: saraju.mohanty@unt.edu (Saraju P. Mohanty), mgomathi@unt.edu (Mahadevan Gomathisankaran), elias.kougianos@unt.edu (Elias Kougianos)

[1] http://nsdl.cse.unt.edu
[2] http://tssl.cse.unt.edu/
[3] http://www.cse.unt.edu/~smohanty
[4] http://www.cse.unt.edu/~mgomathi
[5] http://etec.unt.edu/public/eliask/

computing even in mobile platforms as very large numbers of transistors can be packed in the same area. The challenges in nanoscale chip design include the following: variability, leakage, power, thermals, reliability, and yield [3, 4, 5]. Process variations from different sources have a profound effect on power, leakage and delay. The effect of process variations in delay will translate to uncertainty in clock width in multicycle or pipelined datapaths. This paper focuses on the prominent challenges, at the architectural level, for variability-tolerant power (leakage) optimal nanoelectronic chip design.

Figure 1: Proliferation of smart mobile devices is the key driving factor for low-power chip design.

At present, an increasing number of small and mobile electronic devices (Fig. 1) are being designed, thus heavily relying on battery power for portability. Power dissipation is an important design constraint in high-performance processor systems, system-on-chip (SoC) designs, as well as application specific integrated circuits. To meet the increasing demand of low-power chips with high performance and higher integration density and functionality of digital devices, VLSI design engineers are resorting to relentless scaling in process and design parameters of CMOS transistors. However, this scaling has resulted in a number of new concerns, including a new dimension of leakage current distribution and process variation [6]. The trends of these components are presented in Fig. 2 [7, 2]. Gate-oxide leakage which is dominant in sub-90nm traditional CMOS, is negligible in the case of high-κ based transistors.

The prominent current (power dissipation or leakage dissipation) components primarily depend on gate oxide thickness (T_{gate}), threshold voltage (V_{th}), supply voltage (V_{DD}), and effective device length (L_{eff}). Hence, any methodology for power reduction must focus on the variation of these process and design parameters. This focus has been the motivating factor to consider process variation during architectural-level optimization and to facilitate fast and correct design space exploration right at the early stages of the design cycle, targeted for design for manufacturing (DFM). This will ensure that wrong design decisions are not propagated to the lower levels of circuit abstraction, which may be costly to correct at that stage because of increasing complexity [6]. The novel methodology discussed in detail in this paper consistently does so and incorporates directly the variation in the model for various current components.

To achieve power-performance trade-offs, different solutions have been proposed in the architectural-level (or RTL) synthesis and optimization literature and include different technology dependent and technology independent methods. The technology dependent approaches include scaling of various process and design parameters such as T_{gate}, κ, L_{eff}, V_{th}, and V_{DD} through the use of technologies such as dual-V_{DD} and dual-V_{th}. These approaches handle the optimization of various components independently and do not address the variation of various process and design parameters in the nano-CMOS regime. The research proposed in this paper is further motivated by these important facts of nanoscale technology based architectural-level design exploration.

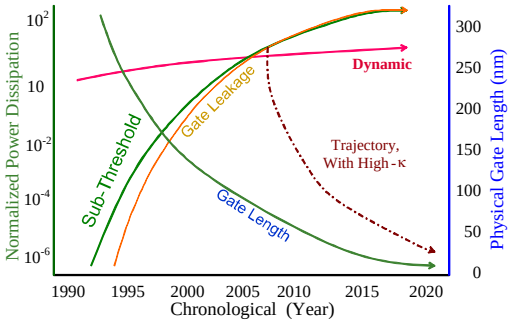


Figure 2: Prediction of trends of major sources of power dissipation in nanoscale chips.

The rest of the paper is organized as follows. The novel contributions of this paper are outlined in Section 2. The specific research issues and challenges arising in the nano-CMOS regime are presented in Section 3. Section 4 summarizes relevant prior related research in architectural-level or RTL optimization. The proposed architectural-level solution and optimization problem formulation is presented in Section 5. A new framework for statistical architectural-level optimization needed to handle nano-CMOS regime circuits is presented in Section 6. A few selected prior research papers dealing with process-variations at the architectural-level are briefly discussed in Section 7. The paper concludes in Section 8 with a summary and suggestions for future research.

2. Contributions of this Paper

In view of the optimization problem of nanoscale circuits at the architectural level (or register-transfer level), while simultaneously accounting for process variations, the *following research questions arise*:

- How do the architectural-level design phases (e.g. scheduling, binding) affect power, leakage, area, and yield in the presence of variations?
- Given architectural constraints, how to judiciously consider the design corners to obtain a power, leakage, and performance optimal circuit for given circuit constraints (from the specifications)?
- If T_{gate}, L_{eff}, V_{th}, V_{DD}, etc. are scaled simultaneously, will a power and performance optimal circuit that has minimal gate leakage, minimal subthreshold leakage, and minimal dynamic power consumption be obtained?

- How are the design characteristics affected by process variation?

Thus, there is a need for research to solve one or more of these questions for fast and accurate architectural-level design exploration of nanoscale chips.

The *contributions of this paper*, all relating to fast and effective architectural-level synthesis of nano-CMOS circuits, are in multiple forms, as summarized below:

- The paper presents detailed insight on the origin and types of process variations in nanoscale CMOS circuits.
- The paper briefly discusses selected architectural-level optimization research works handling power, leakage, or timing in a statistical manner to improve yield.
- A statistical behavioral synthesis framework is presented that can perform different forms of power and leakage optimization accounting for process variation while performing various behavioral synthesis tasks.
- As a specific detailed example, a novel methodology is proposed to characterize nano-CMOS architectural components for gate-oxide leakage, subthreshold leakage, dynamic power, and delay while simultaneously accounting for process variation. The gate and functional units are simulated for obtaining probability distributions (characterized by a mean μ and standard deviation σ) of different current components and delay as well as their correlations. Statistical variations in the parameters (T_{gate}, V_{th}, L_{eff}, and V_{DD}) are explicitly taken into account by using Monte Carlo simulations while characterizing the architectural units.
- The paper provides a comparative and integrated perspective of various power-performance tradeoffs weighed against a nominal case, thus serving as a guideline to help designers make effective decisions. The interdependency of T_{gate}, V_{th}, and V_{DD} scaling on various power (current) components is analyzed with and without process variation. Seven different cases are analyzed for various forms of power dissipation: (1) only T_{gate} scaling, (2) only V_{th} scaling, (3) only V_{DD} scaling, (4) simultaneous T_{gate} and V_{th} scaling, (5) simultaneous T_{gate} and V_{DD} scaling, (6) simultaneous V_{th} and V_{DD} scaling, and (7) simultaneous T_{gate}, V_{th} and V_{DD} scaling.
- A resource-time constrained algorithm is proposed for simultaneous scheduling, binding, and allocation for the reduction of total power accounting for gate-oxide leakage, subthreshold leakage, and dynamic power during process variation aware behavioral synthesis; it judiciously uses dual-T_{gate}, dual-V_{th}, and dual-V_{DD} technologies. The optimization algorithm handles a characterized library in the form of probability density functions (PDFs) and statistical correlations, unlike traditional algorithms which handle nominal data only.

3. Nanoscale Issues during Architecture Optimization

3.1. The Multidimensional Issues of Architectural-Level Design

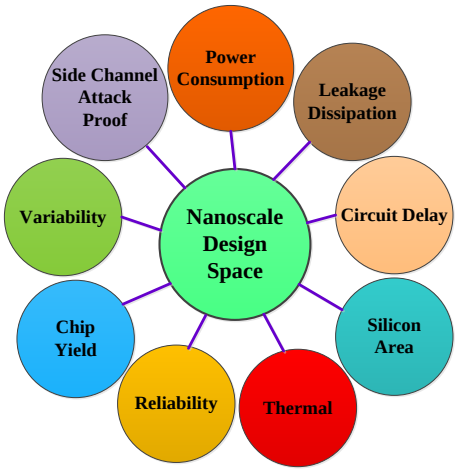


Figure 3: Design Space of Nanoscale Chips.

The designers of chips which are targeted to be manufactured using nanoscale technology face a very complex multidimensional design space as presented in Fig. 3 [6, 3].

Variability: Due to the use of nanoscale manufacturing processes, there is a big discrepancy between the design-phase characteristics and post-manufacturing-phase characteristics of a circuit. This is due to the complex lithography processes used in manufacturing which are close to the limits of the wavelength of light. The variability in process and design parameters is increasing with technology scaling. It affects design decisions, yield, and circuit performance and impact both non-recurring (as better skilled design engineers are needed) and recurring costs. This issue will be much more severe with the projected sub-22 nm technology in the next 5 years.

Leakage: Leakage in nanoscale transistors is increasing with technology progress. Quantum-mechanical tunneling which has been increasing from sub-90 nm, is projected to be quite severe in the future. The form of leakage which is present in both active and passive modes of circuits has been of concern. The leakage current affects the average power dissipation as well as the peak power dissipation of a circuit. The passive leakage is most significant for applications where the system goes to standby mode very often, e.g. smart phones and tablet-PCs.

Power Dissipation: Due to large numbers of transistors in a chip, the overall power dissipation of a chip is increasing. This consumption is further aggravated due to the use of multiple cores. Power dissipation affects energy consumption, cooling costs, packaging costs, and battery life.

Thermal or Temperature: The maximum temperature that can be reached by a chip during its operation is increasing due to increased power and leakage dissipation. The two types of thermal effects are the on-chip temperature and operating temperature. Thermal effects have negative impact on circuit performance, reliability, and cooling costs. Increase in temperature has several consequences including the following: (1) Increase in leakage of transistors. (2) Decrease in active current or driving capability of transistors. (3) Slowing down of the circuits. (4) Increase in the possibility of thermal runaway. (5) Decrease in the reliability.

Reliability: Circuit reliability is decreasing due to compound effects from process variations, power consumption, leakage dissipation, and thermals. It is estimated that every 10° Celsius increase in the temperature reduces the mean time to failure (MTF) of a chip by half. Designers need to consider reliability as an important factor.

Yield: Yield of the fabricated chips using nanoscale processes is decreasing due to increased process variations. The yield is not only affected by the defects introduced in the die but also due to the fact that chips do not meet the specifications due to process variation uncertainties.

Side Channel Attack Proof: This is in particular important for chip designs handling sensitive user data [8]. For example, encryption and watermarking chips handling passwords or costly multimedia data [9, 10, 11]. Side channel attacks gain information from the chip through timing information, power consumption, and electromagnetic leaks [12]. These attacks take advantage of the switching activities that take place during the execution of the static CMOS chip in which different capacitances are switched; this can be determined through power and timing analysis from which the watermarking or encryption keys can be determined, thus breaking the security. This is a very important challenge for present day design engineers as the chips in mobile devices process very sensitive information such as biometric data and financial information.

3.2. The Issue of Nanoscale Process Variation

This is one of the most important issue that has been driving VLSI research during the last decade. Process variation affects device parameters, chip performance, and the chip yield which in turn affects both the non-recurrent and recurrent costs of the chip. The non-recurrent cost increases as the designers need to have better skills to account for process variation by following new paradigms of design for manufacturability (DFM). The recurrent cost is affected as more iterations may be needed during manufacturing to produce a sufficient number of chips to meet the market demand.

Origin and Source: To closely understand the origin of process variations in nanoscale circuits, knowledge of manufacturing is essential. A broad view of the lithographic process is presented in Fig. 4 [13, 14]. To facilitate fabrication of circuits using nano-CMOS technology, more and more sophisticated lithographic, chemical, and mechanical processing steps are adopted. Different phases of manufacturing involve various processes such as the following: (1) Ion implantation (2) Chemical mechanical polishing (CMP) (3) Chemical vapor deposition (CVD) (4) Sub-wavelength lithography (5) Lens aberration (6) Materials flow (7) Gas flow (8) Thermal processes (9) Spin processes (10) Microscopic processes (11) Photo processes. It is anticipated that a subset or a superset of these processes will be used in manufacturing nanoscale chips. These processes introduce different types of variation which needs to be appropriately modeled and accounted for during the design phase.

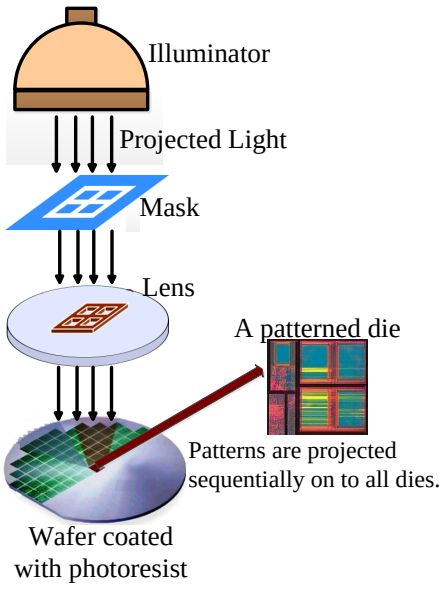


Figure 4: Broad perspective of the lithographic process. The patterns are projected sequentially onto all dies on the wafer, printing the desired circuit on the dies, one after another.

Impact on Device Parameters: Inherent uncertainty in the processes causes variations in process and design parameters [5, 15] such as the following: (1) supply voltage (V_{DD}), (2) NMOS threshold voltage (V_{thN}), (3) PMOS threshold voltage (V_{thN}), (4) NMOS gate dielectric thickness (T_{gateN}), (5) PMOS gate dielectric thickness (T_{gateP}), (6) NMOS channel length (L_{effN}), (7) PMOS channel length (L_{effP}), (8) NMOS channel width (W_{effN}), (9) PMOS channel width (W_{effP}), (10) NMOS gate doping concentration (N_{gateN}), (11) PMOS gate doping concentration (N_{gateP}), (12) NMOS channel doping concentration (N_{chN}), (13) PMOS channel doping concentration (N_{chP}), (14) NMOS source/ drain doping concentration (N_{sdN}), (15) PMOS source/ drain doping concentration (N_{sdP}), (16) metal wire thickness, and (17) via resistance. Process variations have profound effects on electrical parameters and overall performance of a chip and are manifested in the variation in power and delay and other attributes of the chip. They affect design margins and yield and may lead to loss of income in the ever-reducing time-to-market cycle.

Types of Process Variations: In general, process variations can be classified into various different types as depicted in Fig. 5. Parametric variations are a combination of wafer, reticle, and local variations [5, 16]. Wafer variations are combinations of global, linear, and radial variations. Reticle variations are caused by photo processes and local variations are caused by random microscopic processes. Global variations are originating in the fab (fabrication plant, from plant to plant), lot (set of wafers processed in bulk, from one lot to another), and wafer processes (from one wafer to another in a lot).

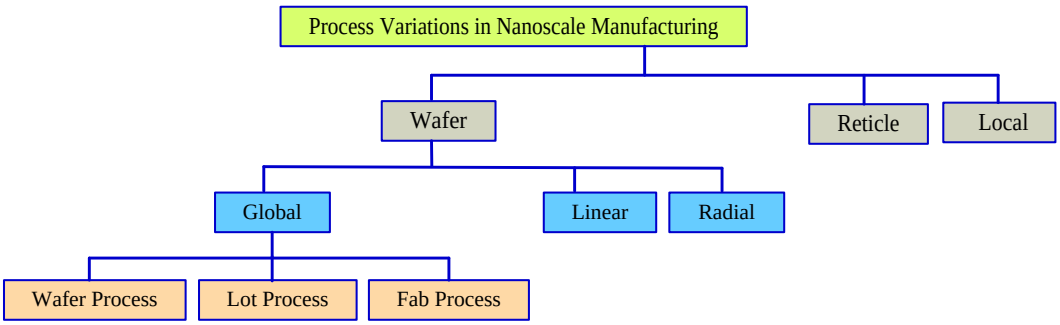


Figure 5: Types of process variations in nanoscale technology.

Design Phase Incorporation of Process Variation: The classical assumption that all transistors are alike is not valid for nanoscale chips. No two transistors in a chip and different chips of the same design are alike. It is absolutely critical that these variations are accounted for in any design decisions to make the circuits robust and to improve yield targeted for design for manufacturing (DFM). Otherwise, underestimation or overestimation of chip characteristics can occur which will lead to design errors and loss of yield. For the purpose of incorporating process variation during the design phase, they are modeled as presented in Fig. 6. These process variations are categorized as inter-die or intra-die and can be either systematic or random. They can also be global, local, and spatial or temporal [5]. Designers accommodate these process variations during the design phase as a DFM paradigm my emulating them, typically using Gaussian random distributions and occasionally non-Gaussian random distributions [17].

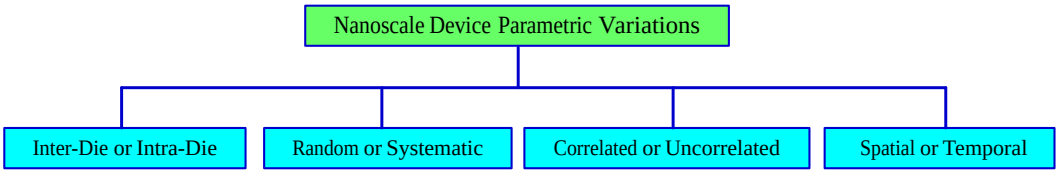


Figure 6: Classification of different types of process variation in nanoscale technology for modeling during the design phase.

3.3. The Issue of Power and Leakage Dissipation

In short-channel nano-CMOS transistors, several short channel effects (SCEs) become significant, such as drain induced barrier lowering (DIBL), large V_{th} roll-off, diminishing on-to-off current ratio, and band-to-band tunneling (BTBT). As a result, a drastic change has occurred in the leakage components of the device, both in the inactive and active modes of operation. The leakage current in short channel nanometer transistors has diverse forms, such as reverse-biased diode leakage, subthreshold leakage, SiO_2 tunneling current (leading to gate-oxide leakage), hot carrier gate current, gate-induced drain leakage (GIDL), and channel punch through current [18, 21, 20]. Each component has several forms and origins; they flow between different terminals and in different operating conditions of a transistor as shown in Fig. 7. While reverse-biased diode leakage and SiO_2 tunnel currents flow during both active and off states, the other currents flow during the off state only. In summary, the principal power components are due to gate-oxide leakage current (I_{gate}), subthreshold leakage (I_{sub}) current, and capacitive switching (dynamic) current (I_{dyn}) [18, 21, 22, 23]:

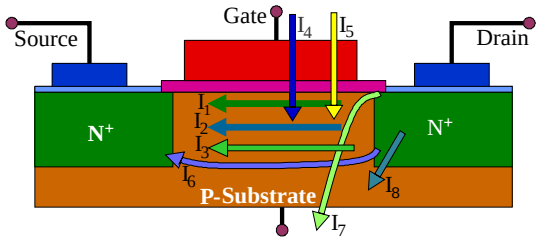


Figure 7: Various current flow paths in a nano-CMOS transistor during different states of its operation [18, 19, 20]: I_1 – drain to source active current (ON state), I_2 – drain to source short-circuit current (ON state), I_3 – subthreshold leakage (OFF state), I_4 – gate leakage (both ON and OFF states), I_5 – gate current due to hot carrier injection (both ON and OFF states), I_6 – channel punch through current (OFF state), I_7 – gate induced drain leakage (OFF state), I_8 – reverse-bias PN junction leakage (both ON and OFF states).

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{subthreshold}} + P_{\text{gate-leakage}}. \quad (1)$$

In the case high-κ/metal-gate nano-CMOS, the power dissipation has the following prominent components [3]:

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{subthreshold}} + P_{\text{GIDL}}. \tag{2}$$

Different currents or power are modeled differently. The dynamic power consumption is modeled as follows [24]:

$$P_{\text{dynamic}} = \alpha C_L V_{DD}^2 f, \tag{3}$$

where α is the activity factor, C_L is the total capacitive load, V_{DD} is the supply voltage, and f is the clock frequency. The subthreshold leakage current through a device is modeled as follows [24]:

$$I_{\text{subthreshold}} = \mu_0 \left(\frac{\epsilon_{ox}}{T_{ox}}\right)\left(\frac{W}{L}\right) v_{therm}^2 e^{1.8} \exp\left(\frac{V_{gs}-V_{th}}{\tau v_{therm}}\right)\left(1-\exp\left(\frac{-V_{ds}}{v_{therm}}\right)\right), \tag{4}$$

where V_{th} is the threshold voltage, τ is the subthreshold swing factor, V_{gs} is the gate-to-source voltage, V_{ds} is the drain-to-source voltage, v_{therm} is the thermal voltage, μ_0 is the zero-bias mobility, ϵ_{ox} is the oxide dielectric constant or relative permittivity, T_{ox} is the oxide thickness, W device width, and L device length. The gate-oxide leakage current which is due to direct tunneling for ultra-think oxide is modeled as follows [24]:

$$I_{\text{gate-leakage}} = \xi W L \left(\frac{V_{ox}}{T_{ox}}\right)^2 \exp\left(\frac{-\eta\left(1-\left(1-\frac{V_{ox}}{\phi_{ox}}\right)^{\frac{3}{2}}\right)}{\left(\frac{V_{ox}}{T_{ox}}\right)}\right), \tag{5}$$

where W is the width of the transistor, L is the channel length, V_{ox} is the potential drop across the thin oxide, T_{ox} is the oxide thickness, ϕ_{ox} is the barrier height for the tunneling particle (hole or electron), and ξ and η are physical parameters. These parameters are described as follows: $\xi = q^3/\left(16\pi^2\hbar\phi_{ox}\right)$ and $\eta = \left(4\sqrt{2m_{eff}}\phi_{ox}{}^{1.5}\right)/\left(3\hbar q\right)$; q is electronic charge, $\hbar$ is Plancks constant, and m_{eff} is the effective mass of the tunneling particle.

4. Overview of Prior Research in Architectural-Level Optimization

The current literature is rich in techniques for power optimization. These techniques are proposed for various levels of circuit abstraction, starting from system-level to silicon. As the level of abstraction goes lower, the complexity of the circuit increases, and the degrees of freedom, and thus power reduction opportunities, decrease. Hence, behavioral level (also known as high-level or algorithmic level) is an attractive level that provides a balanced degree of freedom for design space exploration, which is the focus of this paper. Several techniques, such as architecture-driven voltage scaling, operation reduction and substitution, pre-computation, and clock-gating have been proposed [25, 26, 27]. In addition, technology dependent techniques, such as dual-T_{gate}, dual-V_{th}, and dual-V_{DD} have been proposed for power optimization. An overview of the available techniques is presented in Fig. 8.

4.1. Traditional architectural-level Optimization

Shutdown or Island Technique: In [28] Dal and Mansouri discuss power islands which are motivated by shutdown or power gating techniques. In the power island technique, the circuit is partitioned into islands. Each island is a cluster of logic whose power can be controlled independently and hence can be completely powered down when idle thus eliminating the spurious switching activity and leakage in a great portion of the chip. In [29], Helms *et al.* further combine adaptive body biasing (ABB) with V_{DD}-islands for aggressive power reduction and process, voltage, and temperature (PVT) reduction.

Dual-T_{gate} or Dual-κ Based: In [30], Mukherjee *et al.* propose a gate oxide leakage minimization approach using dual-T_{gate} and dual-κ. Sultania *et al.* in [31] describe an algorithm developed to optimize the total leakage power by assigning dual T_{gate} values to transistors in a given circuit. In [32], Lee *et al.* describe a method developed for analyzing gate oxide leakage current in logic gates and suggest the use of pin reordering to reduce gate leakage. In [33], Sirisantana and Roy use multiple channel lengths and multiple gate oxide thicknesses for reduction of leakage.

Dual-V_{th} Based: Multiple threshold CMOS have been used by Pant *et al.*, [34] as well as Rao *et al.* [35] for subthreshold current reduction. Khouri and Jha [36] propose a dual-V_{th} technique for subthreshold leakage analysis and reduction during behavioral synthesis, targeting the least used modules as the candidates for leakage optimization. Gopalakrishnan and Katkoori in [37] also use the multiple threshold CMOS approach for reduction of subthreshold current during behavioral synthesis and propose binding algorithms for power, delay, and area trade-offs. In [38], Tang *et al.* propose algorithms for subthreshold leakage reduction using dual-V_{th} libraries during behavioral synthesis. In [39], Liu *et al.* applied probabilistic analysis to V_{th} variation. In [40], a dual V_{th} and dual T_{ox} technique is applied to SRAMs in order to reduce leakage. In [41], a dual V_{th}

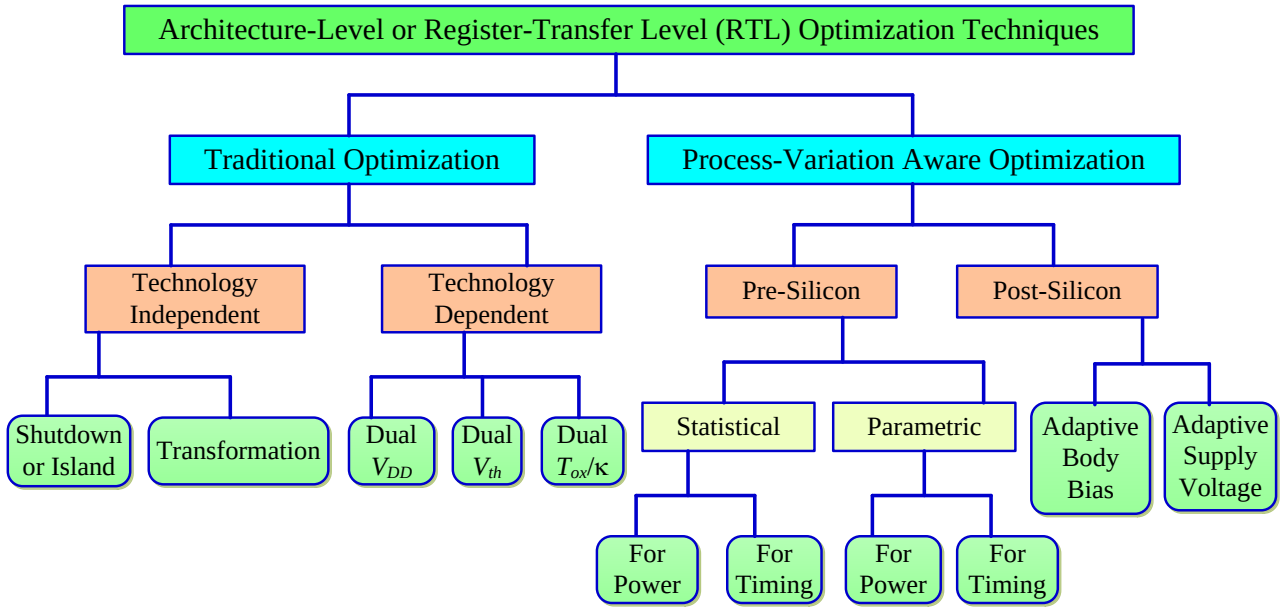


Figure 8: Overview of Architecture Level Optimization Techniques.

FPGA architecture is proposed in which the logic elements are used for dual V_{th} assignment. In [42], Wei *et al.* have tried to reduce leakage power by using high V_{th} transistors in non-critical paths and low V_{th} transistors in the critical paths.

Dual-V_{DD} Based: The prior research using this technique is quite mature and several approaches have been proposed in the literature over the last several years [43, 44, 45]. A certain type of circuitry called voltage-level converter is used for this purpose, but in turn it is an overhead for this kind of technology. The transistors on critical paths are operated at a higher supply voltage (V_{DDh}), whereas transistors on the non-critical paths are operated at a lower supply voltage (V_{DDl}) [46, 47].

4.2. Process-Variation Aware architectural-level Optimization

In digital chip design, process variations have been accounted in various levels, such as system-level, architectural-level, logic-level, transistor-level, and layout-level. In the existing architectural-level design and synthesis literature, pre-silicon or post-silicon techniques are proposed for maximizing the yield in the presence of variability [3, 48, 49]. Pre-silicon techniques are statistical or parametric based optimization approaches during the design phase, that use statistical (parametric) power, leakage, and timing analysis for design space exploration and maximize the parametric yield. A variety of approaches for scheduling, resource sharing, and module selection techniques have been proposed in the existing literature. Post-silicon techniques are approaches such as adaptive body biasing and adaptive supply voltage which are used to tune the fabricated chips such that the circuit yield can be optimized. The techniques are proposed for statistical power as well as timing optimizations. A detailed discussion of these related prior research is systematically presented in Section 7.

5. Architectural-Level Solutions

5.1. The Key Concept

The key concept of architectural-level variability aware optimization is presented in Fig. 9 [3, 6]. The major challenges arising in the nano-CMOS variation scenario are the correct understanding of the process variations and their modeling. Without proper models of variations, designers will include a substantial design margin or risk yield loss when they use traditional computer-aided design (CAD) or electronic design automation (EDA) tools that do not account for such variations. The magnitude of each leakage component of the device is mostly dependent on the device geometry, doping profiles and temperature. At nanometer dimensions, variations of these factors become comparatively more prominent. This prominence leads to the need to account for process variation during characterization and modeling and to integrate process variation in design and optimization frameworks. Moreover, designing for the worst-case scenario may cause severe compromises on the performance of the device.

It may be a noted that the process variation aware optimal RTL or architecture-level description of the chip needs to further go through optimizations are lower levels of design abstraction as shown in Fig. 9. At the logic and circuit level different optimizations including process variation aware optimizations can be performed. At the different levels of abstraction the

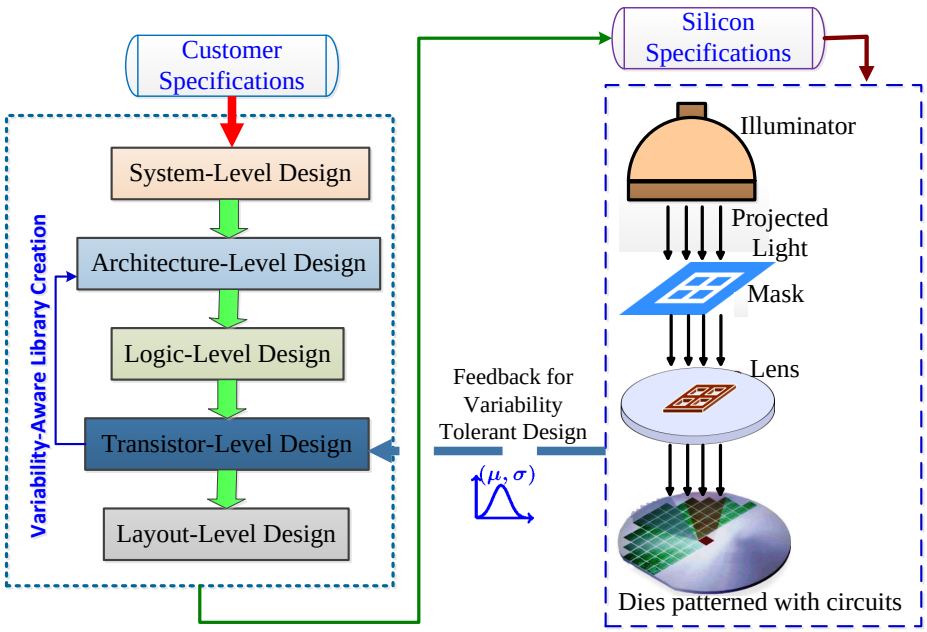


Figure 9: The key idea is to incorporate the process variation effects during architectural-level design flow.

granularity, tuning parameters, complexity, and degree of freedom changes. However, RTL is the focus of this paper as it is essential to make correct design decisions at this level before sending the design to lower levels where correcting a design is costly [6].

In the context of architectural-level optimization, *chip yield is defined as the probability of a design meeting power and performance constraints* [50, 51]. The power yield of a chip is defined as follows:

$$\text{Yield}_P = \text{Probability}\left(\text{Power} \leq \text{Power}_{Constraint}\right). \tag{6}$$

During the architectural-level design exploration $\text{Power}_{Constraint}$, which is the constraint on the power dissipation, can be calculated as a multiple of the average or peak power dissipation. The timing yield or delay of the chip is defined as follows:

$$\text{Yield}_D = \text{Probability}\left(\text{Delay} \leq \text{Delay}_{Constraint}\right). \tag{7}$$

During architectural-level design exploration $\text{Delay}_{Constraint}$, which is the constraint on the chip delay or timing, is calculated as a multiple of the critical path delay of the chip.

The power and delay yields can be targeted either separately or as simultaneous objectives. This possibility has led researchers to consider process-variation-aware power or delay minimization techniques. There are different approaches proposed in the current architectural-level or register-transfer level (RTL) research to target these yields. In the *statistical approach*, the statistical estimation of the various power-performance components, considering statistically varying process and design parameters, changes. Hence, statistical variation in each of these parameters translates to variation in each of the leakage components, thereby causing significant variations in nominal values. Feedback from manufacturing process recipes is accounted at either the transistor level (less accurate) or the physical level (more accurate). The other alternative is the *parametric-based approach* for architectural-level optimization. In this approach the feedback from the process recipes is taken into account at the architectural level and on-the-fly during optimization.

5.2. Formulation as RTL Statistical or Parametric Optimization Problem

At the architectural-level design exploration, let us assume that the datapath is specified as a sequencing data flow graph (DFG) [6]. In the DFG, denoted as $G(V, E)$, V represents the set of vertices and E represents the set of edges. Each vertex of the DFG represents an operation and each edge represents dependency. Let V be the set of all vertices and V_{CP} be the set of vertices in the critical path from the source of the DFG to the output or sink node. For simplicity, it is assumed that the DFG has a single source node and a single sink node as is the case of sequencing data flow graphs, which are directed acyclic graphs [6]. This assumption is sufficient for RTL design exploration; however, the logic level description may use other directed graphs with multiple input/output nodes [30]. The sequencing DFG is used for the following two approaches to account for process variation. The statistical approach is more widely used, but it is complex in terms of calculations. The parametric approach is fast and the process variation aware values are calculated on the fly.

The process variation aware power optimization problem during architectural-level design is stated as follows:
Given an unscheduled data flow graph (UDFG) $G(V, E)$, it is required to find the scheduled data flow graph (SDFG) with appropriate resource binding such that the total power (current) dissipation of the associated circuit is minimized while accounting for process variation and such that resource constraints (representative of silicon cost) and latency constraints (representative of circuit performance or delay) are satisfied.

5.2.1. Statistical Approach

The concept of tradeoff for optimization using the statistical approach is presented in Fig. 10 for propagation delay versus timing yield tradeoffs [52]. The propagation delay for a specific timing yield is calculated by finding the area under the probability distribution function (PDF) curve. For a 100% timing yield, the delay of the datapath component corresponds to point E. However, for a 90% timing yield (10% yield is sacrificed) the propagation delay corresponds to point B. Similarly, point C represents propagation delay for 94% timing yield and point D represents the propagation delay value corresponding to 97% of timing yield. In a similar fashion power yield tradeoff can be performed.

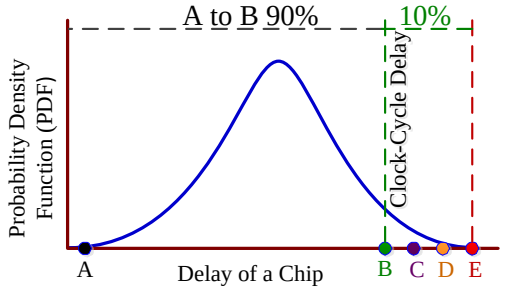


Figure 10: The concept of tradeoff in the statistical approach: the delay values of datapath components as a function of timing yield. A similar analogy can be used for power yield.

Architectural-level optimization can be formally stated as follows for statistical-based optimization. The silicon cost (resource-constrained) and performance (latency-constrained) driven power (current) minimization problem can thus be formulated as follows:

$$\text{Minimize}: \quad \hat{P}_{total}^{DFG}\left(\mu_P^{DFG}, \sigma_P^{DFG}\right), \tag{8}$$

such that the following resource and latency constraints, respectively, are satisfied:

$$\text{Allocated } (FU_{k,i}) \leq \text{Available } (FU_{k,i}) \mid \forall \text{ clock cycle } c, \tag{9}$$

$$\hat{D}_{CP}^{DFG}\left(\mu_D^{DFG}, \sigma_D^{DFG}\right) \leq \hat{D}_C\left(\mu_D^C, \sigma_D^C\right). \tag{10}$$

$\hat{P}_{total}^{DFG}\left(\mu_P^{DFG}, \sigma_P^{DFG}\right)$ in Eq. (8) represents the PDF of the total power dissipation due to the DFG, which can be presented as an equally weighted sum of the PDFs of all current components (dynamic $\hat{P}_{dyn}$, subthreshold $\hat{P}_{sub}$, and gate-oxide $\hat{P}_{gate}$) described as follows:

$$\hat{P}_{total}^{DFG} = \text{Stat-Sum}\left(\hat{P}_{dyn}^{DFG}(\mu, \sigma), \hat{P}_{sub}^{DFG}(\mu, \sigma), \hat{P}_{gate}^{DFG}(\mu, \sigma)\right), \tag{11}$$

where μ, σ are the mean and standard deviation of each of the current distributions and are different for different current components. It may be noted that dynamic, subthreshold, and gate-leakage dissipation may not happen in the same state of a transistor, as dynamic power is due to transitions, subthreshold leakage happens in the OFF state, and gate leakage takes place during both ON and OFF state. However, when calculated over a number of cycles and their average, the above approach is a simpler way for architectural-level design exploration. The resource constraints in Eq. (9) ensure that the total allocation of type k resource (functional units) of technology (or design corner) i is less than or equal to the total number of corresponding resources available for every control step (or clock cycle) c of the DFG. The type k refers to adder, subtractor, multiplier, etc.; i technology (or design corner) refers to the resource made of transistors of design corner i corresponding to specific values of parameters T_{ox}, V_{th}, and V_{DD}. The time constraint in Eq. (10) ensures that the PDF $\left(\hat{D}_{CP}^{DFG}\left(\mu_D^{DFG}, \sigma_D^{DFG}\right)\right)$ of the critical path (CP) delay is within the specified limit dictated by the PDF $\left(\hat{D}_C\left(\mu_D^C, \sigma_D^C\right)\right)$ of the delay constraint. For calculation of a single-value delay for use in optimization, the $(\mu + 3\sigma)$ worst case value is used. It may be noted that $(\mu \pm 3\sigma)$ captures process variations over 99.7% of the variation space and is a typical industrial metric to maximize the yield. Another option is to use $(\mu \pm 2\sigma)$ in which case the process variations are captured in 95% of the variation space, thus potentially reducing the yield by 5% or more.

The following solution is proposed to accurately account for process-variations from the lower level of design abstraction to higher levels of abstraction to enable fast design space exploration. By applying a hierarchical approach, the variations would be faithfully propagated from the lower level to the higher levels of abstraction. Then, we would express the power and performance attributes of these functional units as PDFs instead of single valued functions of parameters. Finally, statistical optimization approaches are presented in which mean (μ), standard deviations (σ) and correlations of PDFs would be considered while performing various tasks of behavioral synthesis such as scheduling or binding, and allocation.

The challenges posed in such a statistical based approach are as follows: (1) How to model variations at the lower level of circuit abstraction? (2) How to estimate power and performance at the higher levels of abstraction while accounting for

these variations? (3) How to optimize power and performance at the higher level of design abstraction while accounting for the variations to enable design space exploration? To solve the optimization problem presented in Eq. (8), (9), and (10) in the framework of behavioral synthesis, we present a simulated annealing algorithm. Even though algorithms for optimization are plentiful in the mathematics literature, we chose to follow the simulated annealing approach because our number of parameters is reasonably large and because faster convergence can be provided by this kind of algorithm [53, 54, 55] for rapid design space exploration.

5.2.2. Parametric Approach

In the previous section power, leakage, and delay models were developed by abstracting the variations of physical parameters. These models, which are expressed as PDFs are then used for power-driven optimization during architectural-level or RTL design. The optimization in this form will be quite sophisticated and involved. As an alternative, physical-aware or parametric-based optimizations are proposed, in which architectural-level analytical models are developed as functions of parameters [56, 3]. Then, process variation aware optimization will be performed on the fly. This approach is a paradigm shift from existing approaches and is anticipated to result in faster convergence for optimal solutions and design space exploration. Another advantage would be to accommodate any type of variation during the optimization process, Gaussian/non-Gaussian.

Architectural-level optimization can be formally stated as follows for parametric-based optimization. The silicon-cost constrained and time-constrained power minimization problem is formally stated as follows:

$$\text{Minimize}: \quad P_{total}^{DFG}\,(\text{Parameters: } \kappa, T_{gate}, V_{th}, V_{DD}, L_{eff}, W)\,, \tag{12}$$

such that the following resource and latency constraints, respectively, are satisfied:

$$\text{Allocated } (FU_{k,i}) \leq \text{Available } (FU_{k,i}) \mid \forall \text{ clock cycle } c, \tag{13}$$

$$D_{CP}^{DFG}\,(\text{Parameters: } \kappa, T_{gate}, V_{th}, V_{DD}, L_{eff}, W) \leq D_C. \tag{14}$$

The key idea of parametric-based solutions of the optimization problem is to describe the optimization objectives as functions of design variables. Then pick a value from the functions using a random value of input design parameters. The random value of the design parameters is selected using probability distributions dictated by process variations.

The challenges to solve the above optimization problem during architectural-level design are tremendous: (1) How to identify appropriate device parameters for different technologies? (2) How to identify appropriate device parameters affecting a specific target cost? (3) How to obtain architectural-level analytical models for the costs as functions of individual parameters, as well as functions of multiple parameters? (4) How to propagate the device-level information to architectural level in the above? (5) How to quantify switching activities for dynamic power and state dependent leakage for different technologies or for emerging technologies? (6) How to identify the PDFs for different parameters for different technologies? (7) How to account for compound and correlated effects of variations of different parameters? For example, variations on V_{th} and T_{gate} can affect subthreshold leakage and delay. (8) How can different optimization algorithms be formed to solve the above formulation while performing synthesis tasks, like scheduling, binding, allocation, etc. (9) How would optimization algorithms accommodate PDFs of parameters including T_{gate}, L_{eff}, V_{th}, and V_{DD}. (10) How to fix the clock cycle width for single cycle, multicycle, pipelined datapath in the above scenario?

6. A Proposed Novel Statistical Approach for Architecture Optimization

6.1. The Proposed RTL Optimization Flow

We present the overall framework for statistical low-power architectural-level optimization in Fig. 11. The optimization framework assumes a behavioral hardware description language (HDL) as input and generates a statistical power and delay optimal RTL description accounting for process variations. The generated RTL will go through logic and physical synthesis before being realized in silicon, a process which is beyond the scope of this paper. As shown in the figure, the entire behavioral synthesis framework is divided into several modules or engines as follows: input generation engine, datapath and control generation engine, characterization engine, process variation engine, power-delay estimation engine, and output generation engine.

Input generation engine: The input generation engine accepts input HDL descriptions, transforms and compiles them, and generates a sequencing data flow graph (DFG) for use by the proposed algorithm. Each vertex of the DFG represents an operation, and each edge represents a dependency. In the current paper sequencing DFG data structure has been used; however other data structures like control data flow graph (CDFG) is possible. The sequencing DFG does not support hierarchical entities, and conditional statements are handled with comparison operations. Each vertex has attributes that specify the operation type. At this step, technology-independent optimizations can be performed.

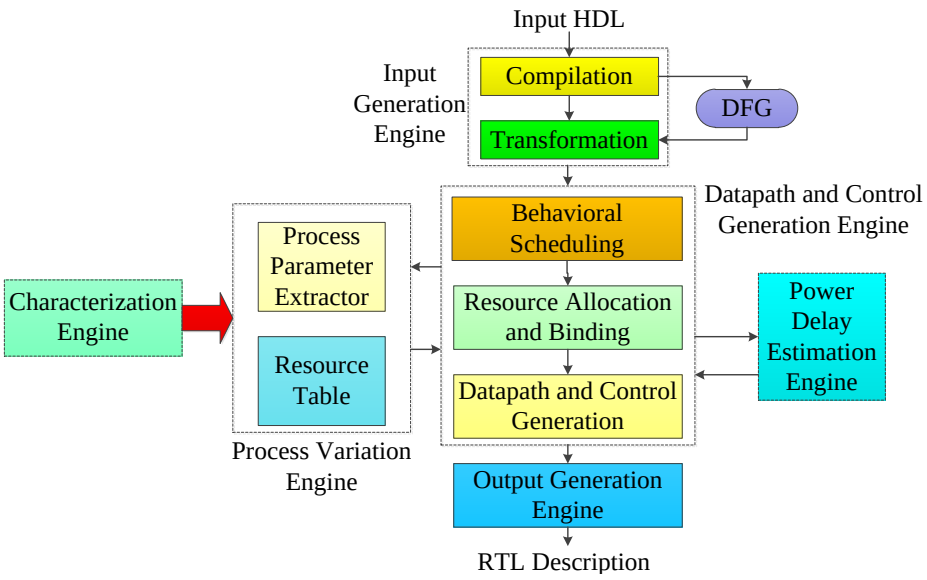


Figure 11: The proposed statistical behavioral synthesis framework for process variation aware design space exploration.

Datapath and control generation engine: The datapath and control generation engine is the principal unit of the process-variation-aware power and delay optimization framework. It carries out behavioral scheduling, resource allocation and binding and generates datapath and control following the power minimization model embedded in the engine. The model provides feedback to the modules carrying out scheduling, binding and allocation at each step of datapath and control generation so as to minimize the power. The process variation engine provides this engine with the statistical model library comprising of various resources with dual T_{gate}, V_{th}, and V_{DD}.

Characterization engine: The characterization engine forms a vital part of the process-variation-aware behavioral synthesis framework. It builds the datapath and component model library and provides statistical models of the functional units used to synthesize the datapath to the process variation engine. Here, the characterization engine is a process-variation-aware statistical model library generator. It takes a multiple set of statistical inputs and generates a set of statistical outputs in terms of current and delay. This engine can also be tuned to generate characterized data for other components. Since the subsequent current (power) model considers process variation in terms of dual-oxide thickness, dual-supply voltage, and dual-threshold voltage, the engine is supplied with a statistical distribution of the three parameters. The characterization engine considers the combination of the dual values of the three input parameters (T_{gate}, V_{th}, and V_{DD}) as eight corners of a design cube. The engine then processes the input cube and generates a corresponding output cube. The output consists of eight sets of current (P_{dyn}, P_{sub}, and P_{gate}) and propagation delay (D_{prop}) probability density functions, each set corresponding to a particular design corner of the input cube.

Process variation engine: The process variation engine consists of a process parameter extractor designed to supply the environment with the statistical data for the requested variable parameter. It also consists of a resource table populated by the characterization engine. This engine works in close coordination with the datapath and control generation engine for process-variation-aware power optimization during behavioral synthesis flow.

Power, leakage, and delay estimation engine: The power (current)-delay estimation engine calculates the PDFs of different current components and delay. It works in coordination with the characterization engine, the datapath control generation engine, etc. and estimates the PDFs for a DFG.

Output generation engine: The power performance optimized datapath and control generated are represented through an RTL description, which is processed by an output generation engine. This RTL is used to carry out the next phase of circuit synthesis, the logic synthesis.

6.2. Process-Variation-Aware Characterization and Study of Parameter Scaling

In this section, we present a hierarchical methodology to characterize architectural level units for gate-oxide leakage, subthreshold leakage, and dynamic power, as well as their propagation delays, as shown in Fig. 12. The dynamic power is calculated assuming a typical load C_{LOAD} as the capacitance of 10 inverters. For a 2-input NAND, the subthreshold leakage is $I_{sub_{NAND}} = \sum_{MOS_{OFF_i}} I_{sub_i}$. In the case of gate-leakage, the path of the current is from gate-to-source in the case of PMOS, whereas in the NMOS it is from gate to both drain and source. Thus, in the worst-case, the total gate-oxide tunneling current for a NAND gate is the sum of six different components as shown in Fig. 12. If the four possible states (00, 01, 10, and 11)

have gate-oxide tunneling currents ($I_{ox00}, I_{ox01}, I_{ox10}, I_{ox11}$), respectively, and assuming that all four states are equiprobable the average gate-oxide tunneling current of a 2-input NAND gate is $I_{gate_{NAND}} = (I_{ox00} + I_{ox01} + I_{ox10} + I_{ox11})/4$. The gate-oxide tunneling current is obtained by evaluating diffusion, channel and body components of the PMOS and NMOS devices from the SPICE model and summing them as $\sum_{MOS_i}(|I_{gs_i} + I_{gd_i} + I_{gcs_i} + I_{gcd_i} + I_{gb_i}|)$. In summary, we account for the *gate-oxide tunneling current of both NMOS and PMOS devices for both their ON and OFF states*.

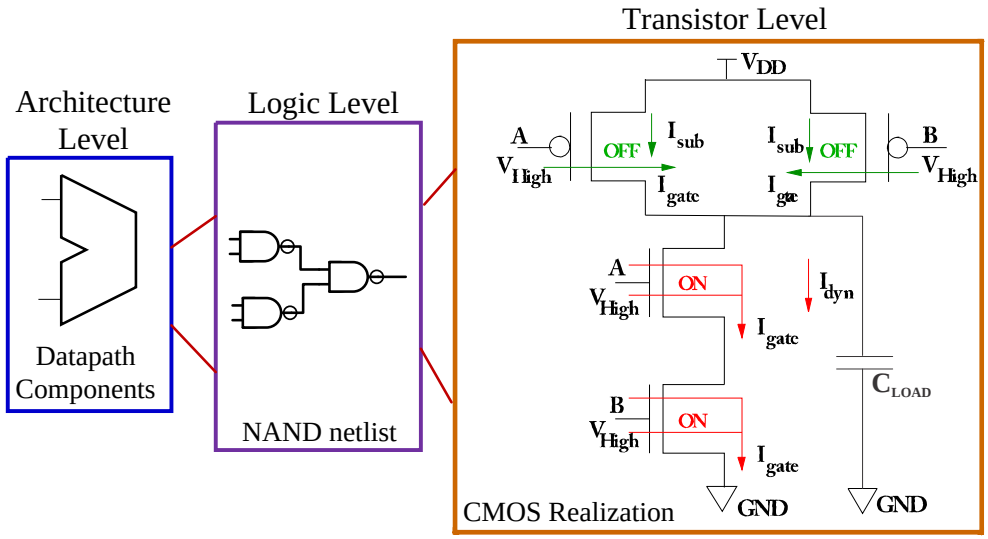


Figure 12: Three levels of abstraction in which datapath components are realized using 2-input NAND gates. The transistor level diagram shows worst-case tunneling current paths in the constituent NAND gates. The worst-case occurs when both inputs are high, i.e., $A = B = V_{High}$.

As an application of this approach, a 2-input NAND logic gate was designed, tested, and characterized at 45 nm nano-CMOS technology. We chose to use the Berkeley Predictive Technology Model (BPTM) because it is widely used [57]. The BSIM4 deck generated from the BPTM represents a hypothetical 45 nm CMOS process. The base (or nominal) values for design corner (1) are as follows: $T_{ox} = 1.4$ nm, $V_{th} = 0.2$ V for NMOS, $V_{th} = -0.2$ V for PMOS, $W : L = 4 : 1$ for NMOS, $W : L = 8 : 1$ for PMOS, and $V_{DD} = 0.7$ V. Via Monte Carlo simulations, we translated the process and design variations (inputs) into gate-oxide leakage, dynamic and subthreshold current, and delay probability density distributions (outputs). The input parameters T_{ox}, V_{th}, L_{eff}, and V_{DD} are assumed to be Gaussian with a standard deviation of 10% of the mean (demonstrated in Fig. 13). The distribution of each current, in addition to the mean and standard deviation, is also characterized by its correlation coefficient to the delay. This will ensure that when random assignments are made later by the optimization algorithm, the assigned gates maintain properly correlated properties. Variations are being modeled as Gaussian density functions, and the variability of different current (power) components is studied. This study will be useful for statistical process-aware current (power) optimization during RTL optimization.

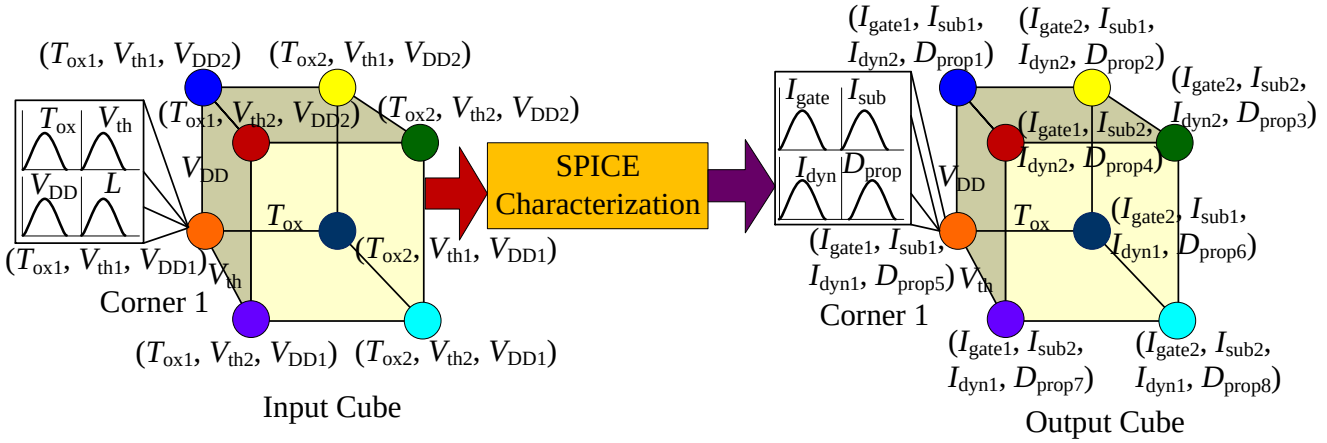


Figure 13: Monte Carlo simulation methodology to account for physical process variations and power supply variation in the power and performance of circuits. Corner 1 is assumed as the baseline corner.

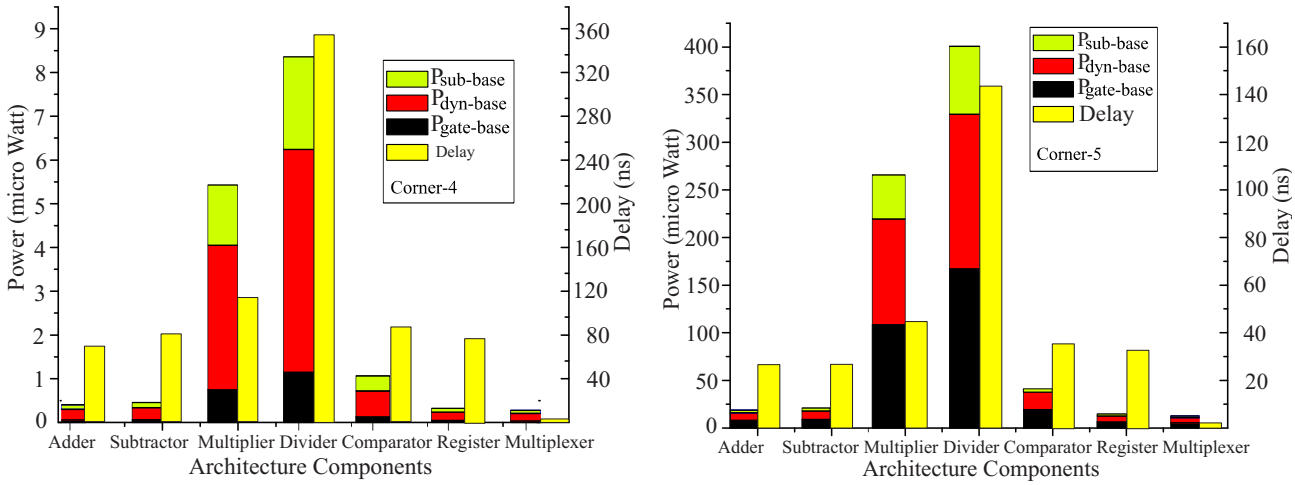
Although state-dependent data are obtained at the logic level, at the architectural level we followed a state independent approach. This was done by using the state averaged data derived from the characterized NAND gate. In order to account for the lognormal distribution of the currents at the gate level, we used the Central Limit Theorem (CLT). Since a typical functional unit is comprised of hundreds of NAND gates, according to the theorem, the gate-oxide leakage, dynamic and subthreshold leakage currents for the functional unit will be normally distributed even though the same currents are lognormally distributed for each individual logic gate. Based on the above discussion, we can model the currents and the delay for *functional units*, comprised by hundreds or thousands on NAND gates, by utilizing the characterized data for the 2-input NAND gate in two distinct approaches.

For simplicity in modeling, we can assume that the distributions for each gate are statistically independent of each other. The mean and variance of the currents can be derived as:

$$\mu_{FU} = \sum_{i=1}^{N} \mu_{\text{NAND}_i} \qquad \text{and} \qquad \sigma_{FU} = \sqrt{\sum_{i=1}^{N} \sigma^2_{\text{NAND}_i}}, \tag{15}$$

where there are N NAND gates in the implementation of the FU. The assumption of statistical independence for all gates in a given functional unit implies that there are no statistical correlations between adjacent gates due to spatial effects. From the above equations, the mean and the variance of I_{gate}, I_{sub}, and I_{dyn} for each of the functional units is calculated. The calculation of the mean and variance for the delay D_{prop} is also performed in a similar manner. However, for more accurate modeling, correlated distributions of the current and delay PDFs can be used to generate random assignments of NAND gates in a functional unit, and the unit can then be characterized via Monte-Carlo fast SPICE simulations. The cell (datapath component) library containing datapath components such as adder, subtractor, multiplier, divider, multiplexer, and register are constructed by using the universal NAND gate, because the NAND gate outperforms all other logic gates with respect to leakage and propagation delay for the same number of inputs [30, 36]. Other types of logic gates to build a datapath component library can be used with the above statistical expressions, provided that the number of individual logic gates in a functional unit is large enough to justify the use of the central limit theorem, a realistic assumption for real-life designs.

At the end of this procedure, a complete process and design-variation-aware cell library was obtained for use in the subsequent optimization procedure. Characterization nominal data for some sample design corners are shown in Fig. 14. It may be noted that the total current values are reduced and that the proportions of different components in the total current have changed. Only two corners are shown for brevity. In corner 5 versus corner 4, all parameters have been scaled, i.e., T_{gate} and V_{th} are increased while V_{DD} is decreased.



(a) Design corner 4: $T_{gate} = 1.7$ nm, $V_{th} = 0.3$ V, and $V_{DD} = 0.7$ V (b) Design corner 5: $T_{gate} = 1.4$ nm, $V_{th} = 0.2$ V, and $V_{DD} = 0.9$ V

Figure 14: Nominal results showing individual components of power consumption for different output corners (corner 5 and corner 4).

6.2.1. Analysis of effects of scaling on individual current components

Characterization data for some sample corners are shown in Fig. 14. We obtained a similar set of data for all eight corners but they are not shown here for brevity. For the analysis, we consider the eight corners of the cube as nominal corners. Corner 1 is considered the baseline corner, having values of V_{DD}, T_{gate}, and V_{th} as the standard values for a 45 nm nano-CMOS technology. The values in the other nominal corners vary with respect to this corner. These nominal corners are processed

through SPICE characterization, and the outputs obtained are the values of I_{dyn}, I_{sub}, I_{gate}, and D_{prop} respectively, which are treated as the eight corners of the output cube.

For the purposes of this analysis, we are considering the case of the highest power consuming datapath component, the divider. However, the trend is the same for other datapath components as well. In total, we had seven cases. At this point, we considered the nominal results without considering statistical distributions for simplicity. In the next section, we will consider the results with the statistical distributions. It should also be pointed out that in this discussion, we refer to "scaling" as the process of reduction of power. In that sense, scaling V_{DD} implies *decrease* in its value, but scaling T_{ox} and V_{th} implies an *increase* in their values.

Only T_{gate} scaling: This case may arise when T_{gate1} changes to T_{gate2}; e.g., (T_{gate1}, V_{th1}, V_{DD1}) versus (T_{gate2},V_{th1},V_{DD1}). In this case, we observe that all power components are reduced with an overall reduction of 89.4% achieved. As expected, the increase in oxide thickness results in a 62.3% delay penalty.

Only V_{th} scaling: Scaling V_{th} only [(T_{gate1},V_{th1},V_{DD1}) versus (T_{gate1},V_{th2},V_{DD1})]. In this case we observe that total power dissipation decreases by 48.1%, whereas the delay penalty is only 16.8%.

Only V_{DD} scaling: Scaling V_{DD} only [(T_{gate1},V_{th1},V_{DD1}) versus (T_{gate1},V_{th1},V_{DD2})]. In this case, we observe that total power dissipation decreases by 56.5% with a modest 20.2% delay penalty.

Simultaneous T_{gate} and V_{th} scaling: [(T_{gate1},V_{th1},V_{DD1}) versus (T_{gate2},V_{th2},V_{DD1})]. In this case, the combined effect of T_{gate} and V_{th} increases results in 93.4% reduction in power but a very significant 99.0% delay penalty. This is due to the inverse relation of the delay to *both* T_{gate} and V_{th} [58]:

$$D_{prop} \propto \left(\frac{1}{T_{gate}(V_{DD} - V_{th})^2} \right). \tag{16}$$

Simultaneous T_{gate} and V_{DD} scaling: [(T_{gate1},V_{th1},V_{DD1}) versus (T_{gate2},V_{th1},V_{DD2})]. As anticipated from Eq. (16), the delay penalty is again significant (102.5%) with similar reduction in power as in the previous case (92.3%).

Simultaneous V_{th} and V_{DD} scaling: [(T_{gate1},V_{th1},V_{DD1}) versus (T_{gate1},V_{th2},V_{DD2})]. In this case, we observe that since both V_{th} and V_{DD} have been scaled, by Eq. (16), we anticipate a more pronounced delay: 40.1%. The overall power reduction is not as large as when T_{ox} is scaled (due to the exponential dependence of gate leakage on T_{ox}): 70.0%.

Simultaneous T_{gate} and V_{th} and V_{DD} scaling: We note that the effect of scaling all parameters cannot be easily or accurately obtained from the responses (output results) of varying a single or a few parameter(s). [*(T_{gate1},V_{th1},V_{DD1})*] versus [*(T_{gate2},V_{th2},V_{DD2})*]: When all three parameters are scaled simultaneously, we obtain a power reduction of 94.2% and a worst-delay penalty of 150.8%. These performance results, indicated in Fig. 14, are not easily anticipated from simple analysis of the prior six cases (corners 2 through 7). This is difficult because of parameter interdependency and variation statistics, wherein comes the usefulness of our quick statistical library models and analysis methodology. This corresponds to the last column of Table 1. This case is represented in Fig. 14.

Table 1: Percentage (%) reduction in current dissipation and increase in propagation delay.

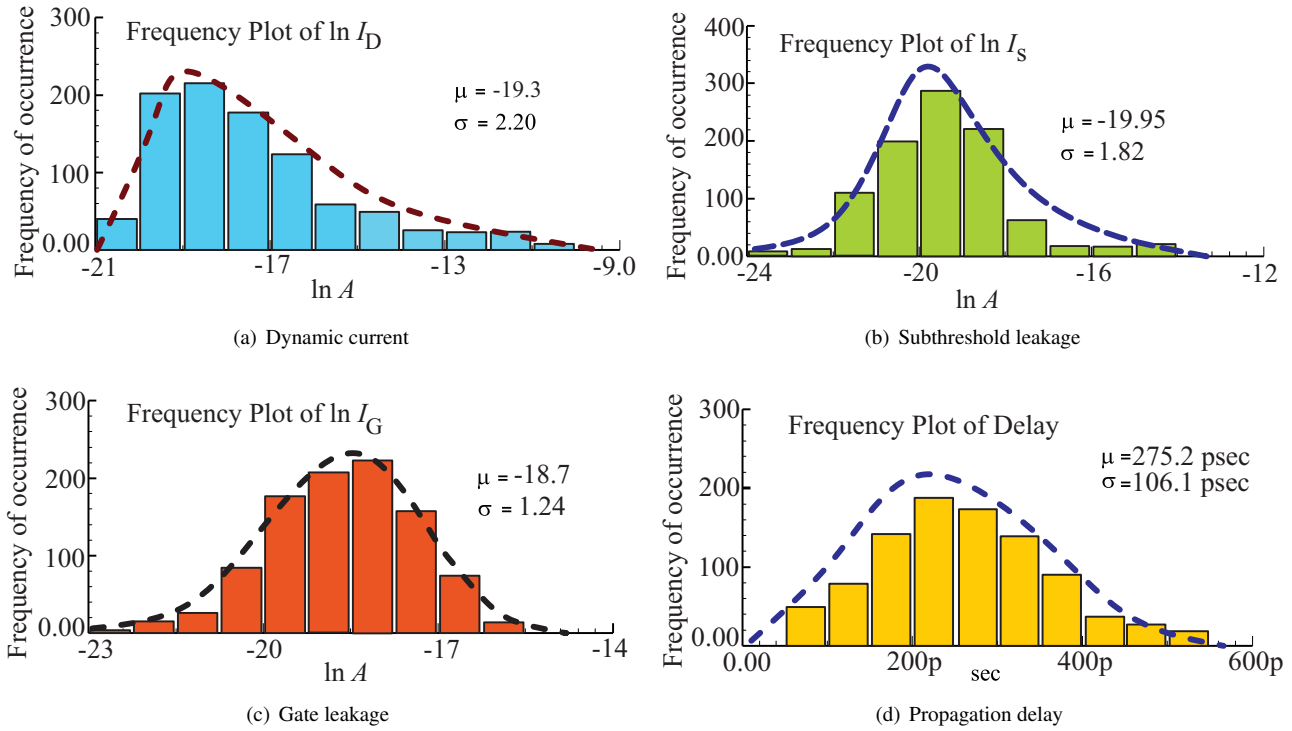
Current or Delay	Parameters Varied or Scaled with Respect to Base Line Corner						
	T_{gate} Case 1	V_{th} Case 2	V_{DD} Case 3	T_{gate} + V_{th} Case 4	T_{gate} + V_{DD} Case 5	V_{th} + V_{DD} Case 6	T_{gate} + V_{th} + V_{DD} Case 7
Gate-Oxide Leakage	94.1	15.4	70.1	96.2	97.8	71.6	98.9
Subthreshold Leakage	91.2	13.9	64.2	88.9	95.1	57.6	95.4
Dynamic Power	81.3	54.4	70.5	91.2	90.5	72.4	92.6
Total Current	89.4	48.1	56.5	93.4	92.3	70.0	94.2
Critical Path Delay	62.3	16.8	20.2	99.0	102.5	40.1	150.8

From the above discussions, it is evident that we cannot simply apply case 7 to obtain a globally power and performance optimal circuit. Hence, this discussion demonstrates the need for optimization algorithms for judicious choice of scaling and serves as a guiding factor for the optimization approach discussed in the next section.

6.2.2. Accounting for process variation

In this section, we describe the methodology by which statistical information on process and design parameter variability is translated into statistical information on power dissipation and delay (performance), as shown schematically in Fig. 13. It is observed that the variability of gate-oxide leakage, subthreshold leakage, and dynamic current is lognormal. This can be seen from the fact that the histograms for the *logarithms* of the currents are normal or (Gaussian). The variability of propagation delay is normal (or Gaussian).

The SPICE characterization engine considers the combination of the dual values of the three input parameters (T_{gate}, V_{th}, and V_{DD}) as eight corners of a design cube. In principle, this can be extended to any number of values of each input



(a) Dynamic current

(b) Subthreshold leakage

(c) Gate leakage

(d) Propagation delay

Figure 15: Effects of statistical process variation on dynamic, subthreshold-leakage and gate-oxide leakage current, and delay in a 2-input NAND gate.

parameters, which will accordingly increase the number of design corners. The engine then processes each corner of the input cube and generates a corresponding output cube. The output consists of eight sets of current (I_{gate}, I_{sub}, and I_{dyn}) and propagation delay (D_{prop}) PDFs, along with their correlations, each set corresponding to a particular design corner of the input cube. The provided statistical information for each input corner is used to generate $N = 1000$ Monte Carlo runs (per corner) which provides the statistical distributions of the output parameter.

It was observed that with normally distributed input parameters, the distribution of the output currents was lognormal (as expected from their exponential dependence on the inputs) whereas that of the propagation delay was Gaussian. Sample distributions of the logarithms of the currents (which are normally distributed) and the delay are shown in Fig. 15.

6.3. Stat-SAO: The Proposed Simulated Annealing Based Statistical Optimization Algorithm

In this section, we present an algorithm that performs simultaneous scheduling, binding, and allocation during the statistical RTL optimization flow presented in Section 6.1. The simulated annealing based algorithm performs the minimization of the cost function presented in Section 6.2 under resource and time constraints. We also present the methodology adopted in this paper for statistical modeling of power and delay. Initially, the current (power) and delay models are developed to capture the variability of current and delay for each cycle and then the overall circuit. We assume that the datapath is represented as a DFG derived from an HDL specification. In our analysis, we assume all statistical quantities to follow normal distributions. The delay of a control step is dependent on the delays of the functional unit, the multiplexer, and register. We assume that each node connected to the primary input is assigned two registers and one multiplexer, whereas the inner nodes of the DFG have one register and one multiplexer. The register and the multiplexer operate at the same supply voltage level (V_{DD}) as that of the functional unit they are associated with. Moreover, the register and the multiplexer are made of transistors of the same V_{th} and T_{ox} as that of the transistors of their associated functional units. Voltage level converters are used when a low-voltage functional unit is driving a high-voltage functional unit.

In this paper we propose a simulated annealing based algorithm because the number of parameters involved in optimization is large and a simulated annealing approach can facilitate faster convergence [53, 54, 55] compared to more sophisticated approaches like integer linear programming (ILP) [59]. ILP can provide a globally optimal solution, but its complexity can be substantial when so many parameters are handled together. Simulated annealing algorithms borrow ideas from Materials Science. Annealing is the process of heating and cooling a material slowly until it crystallizes. The atoms of this material have higher energies at very high temperatures. This capability gives the atoms a great deal of freedom in their ability to reconstruct themselves. As the temperature decreases, the energy of the atoms decreases. Analogous to the annealing process, the mobility of nodes in a DFG representing a data path circuit is dependent on the total available resources. Here, the nodes of a DFG are analogous to the atoms, and temperature is analogous to the total number of available resources. The mobility

of the nodes is dependent on the total number of available resources or functional units. We apply the annealing principle to our problem and explore the trade-offs between power and performance.

We present a simulated annealing based algorithm (Algorithm 1) that minimizes the cost function subjected to constraints. The inputs to the behavioral scheduler are an unscheduled DFG (UDFG), and the resource and/or time constraints. The resource constraints include the number of resources made of transistors of different design corners. The time constraints are expressed as the delay trade-off factor (DTF), which is a quantity that specifies the critical path delay of the target circuit with respect to the baseline's critical path delay. Given a time constraint, we need to determine an RTL implementation that has minimum total power consumption. The starting point of the algorithm is ASAP (as soon as possible) and ALAP (as late as possible) scheduling, which help in determining the mobility of vertices. The initial solution is the resource-constrained ASAP schedule with assignment of design corner 1 resources to all the operations (our base case, design corner 1, corresponds to the nominal V_{th}, T_{gate}, and V_{DD} values of the process.) This is done by the function Allocate_Bind. S represents a scheduled DFG with resource binding. The total current is determined as the weighted sum of currents of all the allocated resources; hence the minimum number of resources required for the schedule is determined and allocated. Once the execution of a clock cycle is finished, all the resources are assumed to be in the ready state before running the next clock cycle or control step.

Algorithm 1 *Stat-SAO*: The simulated annealing based optimization of statistical power and delay.

Available resource $\leftarrow$ Input resource constraint.
Get resource constrained ASAP schedule of the DFG. Get resource constrained ALAP schedule of the DFG.
Fix the number of control steps as the maximum of the ASAP and ALAP schedules.
Target Delay $\leftarrow DTF \times N_{cc} \times$ Delay of the slowest resource in the base line corner.
Determine the mobility graph from the resource constrained ASAP and ALAP schedules.
$\Theta \leftarrow$ Initial temperature.
while {There exists a schedule with available resources.} **do**
  $i \leftarrow$ Number of iterations.
  Initial Solution $\leftarrow$ Resource constrained ASAP schedule.
  $S \leftarrow$ Allocate_Bind(). Initial Cost $\leftarrow$ Cost(S), calculated using Algorithm 3.
  while $\{(i > 0)\}$ **do**
    Generate a mean value based corner prioritized transition from S to S^* with other assignments while satisfying constraints using Algorithm 2.
    Δ-cost $\leftarrow$ Cost(S^*) $-$ Cost(S), calculated using Algorithm 3.
    if $\left\{(\Delta\text{-cost} < 0) \text{ or } \left(\text{rand}(0,1) < e^{\left(-\frac{\Delta\text{-cost}}{\Theta}\right)}\right)\right\}$ **then**
      return $S \leftarrow S^*$.
    end if
    $i \leftarrow i - 1$.
  end while
  Decrement available resources. $\Theta \leftarrow$ Cooling Rate $\times \Theta$.
end while
Determine power, leakage, and delay of the circuit.
return S.

In the outer loop during each iteration, the number of resources is decreased; this decrease restricts the mobility of the nodes. The algorithm attempts to find a scheduled DFG with minimum cost for a given number of available resources. In the inner loop during each iteration, a neighborhood solution is generated. If this solution has lower cost than the current solution, the neighborhood solution is made the current solution. This way, the algorithm converges to a solution that has minimum cost. In generating a neighborhood solution we randomly select a node and check whether a better resource (a resource with less power) can be assigned in all possible clock cycles and ensure that it satisfies a time constraint. We have presented the pseudocode of the algorithm that generates the neighborhood solution in Algorithm 2 for mobile vertices that handle both multicycling and single-cycle datapaths. This *algorithm prioritizes* the design corners based on total current and delay. It ensures that all non-critical path resources are assigned less power-consuming resources.

The cost function of Algorithm 1 is calculated with the help of Algorithm 3. The total current and all the summations presented in the current cost functions are summations of PDFs. Therefore, the cost function itself is a PDF. We translate it into a single value by forming a weighted average of its mean (μ) and standard deviation (σ) with weights 1 and 3, respectively, to cover most of the sample points in a Gaussian distribution for maximizing the chip yield. The cost corresponding to the delay is calculated in a similar fashion as shown in the algorithm. The total cost associated with a scheduled DFG with specific resource allocation and binding is the product of the current cost and the delay cost. This product implicitly ensures that the

Algorithm 2 *Stat-SAO-Neighbor*: Generate a neighborhood solution for Stat-SAO.

Select a random vertex $v_i \in V$.
for all $\{$Possible cycles c in the mobility range.$\}$ **do**
  Choose a resource available as per the corner priority. Schedule v_i in step c.
  Adjust resource allocation Table accordingly. Total Delay $\leftarrow$ 0. /*Initialize Delay*/
  while $\{\forall\ v_i \in V$ execution of v_i is not done. $\}$ **do**
    for all $\{v_i \in V\}$ **do**
      if $\{$All predecessors of v_i finished execution, and v_i has not yet started execution, and required resource is available. $\}$ **then**
        start executing v_i
      end if
    end for
    for all $\{v_i \in V\}$ **do**
      if $\{v_i$ started execution and not yet finished$\}$ **then**
        var $\leftarrow v_i$, break; /*var–node executed*/
      end if
    end for
    Increment Total Delay by $(\mu + 3\sigma)$-delay of resource allocated in this iteration.
    if $\{$Total Delay $>$ Target Delay$\}$ **then**
      break; /*Time constraint failed*/
    end if
    for all $\{v_i \in V\}$ **do**
      if $\{v_i$ started execution and not yet finished$\}$ **then**
        Execute v_i for a period of delay of resource allocated in this iteration.
      else if $\{v_i$ finished execution$\}$ **then**
        mark v_i as completed. /*executed*/
      end if
    end for
  end while
end for

optimizer converges when the solution is optimal from a power and delay point of view. This way, the correlation of power and delay is accounted for (i.e., when power is high delay is low; and when power is low delay is high). .

The average power for the circuit is modeled considering a complete set of assignments and the computation of average total power (sum of average gate-oxide leakage, subthreshold leakage, and dynamic current at each stage in the datapath) in each cycle. It is assumed that N_{FU} functional units are active during cycle c and $FU_{k,i,v}$ is the v-th instance of a functional unit, which is of type k and made of technology corresponding to corner i. The $FU_{k,i}$ is a datapath component like adder, subtractor, etc. made of transistors of specific T_{ox} with specific V_{th} and operated at V_{DD} corresponding to corner i, each having specific PDFs. The variability in total power dissipation of the overall datapath circuit under optimization that is being specified by the DFG for this assignment is then given by summing over all cycles:

$$P_{total}^{DFG}\left(\mu_P^{DFG}, \sigma_P^{DFG}\right) = \text{Statistical-Summation}_{N_{cc}}\left(P_{total}^{c}\left(\mu_{total}^{c}, \sigma_{total}^{c}\right)\right). \tag{17}$$

The worst-case estimate for the total power value of the overall datapath is calculated as: $P_{total}^{DFG} = \mu_P^{DFG} + 3 \times \sigma_P^{DFG}$, where N_{cc} is the total number of cycles in the datapath. Implicit in the previous discussion is the assumption that each functional unit's delay and leakage are statistically independent from those of other units. It is possible to obtain correlations between functional units (to account, for example, for spatial correlations) but this only adds to the complexity of the characterization methodology, not the algorithm itself. For simplicity, and to make the exposition of the algorithms clearer, we do not include such correlations in this discussion.

The variability in the delay of the datapath circuit would be calculated in a similar fashion, but there is a distinct difference. The clock cycle width in the case of single-cycle datapath would be fixed as the worst-case delay (mean value) of any functional unit active in any control step. However, to quantify the variability in delay of the overall circuit, the mean and standard deviation similar to the current need to be considered. For clock cycle c the PDF representing the variability in delay can be represented as follows:

$$\hat{D}_{prop}^{c}\left(\mu_D^c, \sigma_D^c\right) = \text{Statistical-Maximum}\left(\hat{D}_{prop}^{FU_{k,i,v}}\left(\mu_D^{FU}, \sigma_D^{FU}\right)\right)_{\forall v}, \tag{18}$$

Algorithm 3 *Stat-SAO-Cost*: Cost calculation during Stat-SAO.

1: $P^c_{dyn}\left(\mu^c_{dyn}, \sigma^c_{dyn}\right) = \text{Statistical-Summation}_{\forall v \in c}\left(P^{FU_{k,i,v}}_{dyn}\right)$.
2: $P^c_{sub}\left(\mu^c_{sub}, \sigma^c_{sub}\right) = \text{Statistical-Summation}_{\forall v \in c}\left(P^{FU_{k,i,v}}_{sub}\right)$.
3: $P^c_{gate}\left(\mu^c_{gate}, \sigma^c_{gate}\right) = \text{Statistical-Summation}_{\forall v \in c}\left(P^{FU_{k,i,v}}_{gate}\right)$.
4: $P^c_{total}\left(\mu^c_{total}, \sigma^c_{total}\right) = \text{Statistical-Summation}\left(P^c_{dyn}\left(\mu^c_{dyn}, \sigma^c_{dyn}\right), P^c_{sub}\left(\mu^c_{sub}, \sigma^c_{sub}\right), P^c_{gate}\left(\mu^c_{gate}, \sigma^c_{gate}\right)\right)$.
5: $P^{DFG}_{total}\left(\mu^{DFG}_P, \sigma^{DFG}_P\right) = \text{Statistical-Summation}_{N_{cc}}\left(P^c_{total}\left(\mu^c_{total}, \sigma^c_{total}\right)\right)$.
6: $Cost^{DFG}_P = \mu^{DFG}_P + 3 \times \sigma^{DFG}_P$, the worst-case estimate for the total power value of the overall datapath.
7: $\hat{D}^c_{prop}\left(\mu^c_D, \sigma^c_D\right) = \text{Statistical-Maximum}\left(\hat{D}^{FU_{k,i,v}}_{prop}\left(\mu^{FU}_D, \sigma^{FU}_D\right)\right)_{\forall v}$, for all resources active in cycle c.
8: $D^c_{prop} = \mu^c_D + 3\sigma^c_D$, the worst-case estimate of the clock cycle delay.
9: $\text{Cost}^{DFG}_D = N_{cc} \times D^c_{prop}$, critical path delay for the DFG.
10: $\text{Cost} = Cost^{DFG}_P * Cost^{DFG}_D$, for overall DFG and corresponding datapth.
11: **return** Cost. // For the overall datapath for Stat-SAO algorithm iteration.

where it is assumed that N_{FU} functional units are active during cycle c and $FU_{k,i,v}$ is the v-th instance of a functional unit. The single value quantified as $D^c_{prop} = (\mu^c_D + 3\sigma^c_D)$ is a worst-case estimation of clock cycle time, and then the maximum is determined as a single-valued. The delay in the datapath circuit represents the delay for the critical path. This delay is given by the following expressions for N_{cc} number of cycles:

$$D^{DFG}_{CP} = N_{cc} \times D^c_{prop}. \tag{19}$$

The preceding models use the statistical process variation datapath component library containing base value, mean and standard deviation of currents and delay. Moreover, the above model is implemented in the optimization algorithm that performs simultaneous scheduling, binding, and allocation, which is based on the simulated annealing methodology described in this Section.

6.4. Experimental Results of Stat-SAO

In this section, we present the experimental results and our findings. The datapath component library characterization was performed using Cadence's analog design environment and analog circuit simulator and fast SPICE. The input vectors to the various functional units were generated using the autoregressive moving average model to capture correlations seen in actual circuits. On the other hand, the simulated annealing algorithm is implemented in C and integrated in our in-house behavioral synthesis tool [60]. The algorithms were exhaustively tested with several behavioral-level benchmark circuits for several constraints. However, we present the experimental results in this section for a selected set of circuits and resource-time constraints for brevity. The selected circuits are as follows [6, 60, 61]: autoregressive filter (ARF), band-pass filter (BPF), discrete cosine transformation (DCT) filter, elliptic wave filter (EWF), finite impulse response (FIR) filter, and MPEG motion vectors (MMV). The total number of nodes in the corresponding DFG ranges from 23 to 42.

For each benchmark circuit, we discuss results based on several sets of experiments. In the first set of experiments, we used a smaller number of low-cost resources and a higher number of high-cost resources. In the second set of experiments, we used a higher number of low-cost resources as compared to the first set of experiments. In the third set of experiments, we used a higher number of low-cost resources as compared to the second set of experiments. In the fourth set of experiments, we relaxed the resource constraints to study the time-constrained approach only. The time constraints are specified as a multiple of the critical path delay corresponding to this baseline case. We performed our experiments with different delay trade-off factors (time constraints) ranging from 1.0 to 1.4. For each resource constraint, these time constraints are applied and exhaustive experiments are performed. The resource constraints represent the functional units of different design corners available to the behavioral scheduling-binding algorithms. The sets of resource constraints were chosen so as to cover functional units consisting of different corners. They are representative of various forms of the corresponding RTL representation.

We consider design corner 1 (nominal T_{ox}, V_{th}, and V_{DD}) as the baseline. The percentage reduction is calculated as:

$$\Delta I = \left(\frac{P_{\text{Baseline}} - P_{\text{Optimal}}}{P_{\text{Baseline}}}\right) * 100\%. \tag{20}$$

This formula uses the $(\mu + 3\sigma)$-values of the various components of current as well as the total current for computation. The comparison is performed with the baseline design because the proposed methodology is a new methodology against the

standard practice, which is baseline design. This approach is followed by comparison with existing research works discussed in Section 4 such as [44, 45, 36, 37, 38]. The percentage reductions for P_{dyn}, I_{sub}, P_{gate}, and total power P_{total} are calculated for the overall datapath circuit. The experimental results consider the power and propagation delay of functional units and storage units present in the datapath circuit. After several trials, we found that 200 iterations provide a good trade-off between algorithm performance and cost function reduction. It was observed that typical simulation time for a benchmark circuit was in the range of 50 min. to 70 min., which proves that our algorithm converges to solutions in a very reasonable time. We used a dual-CPU Xeon quad-core 2.3 GHz based server with 24GB RAM. The experimental results are presented in Table 2.

Table 2: Experimental Results for Various RTL Benchmarks for Selected Constraints.

Benchmark Circuits	D_T	$P_{gate_{opt}}$ (μA)	ΔP_{gate} (%)	$P_{sub_{opt}}$ (μA)	ΔP_{sub} (%)	$P_{dyn_{opt}}$ (μA)	ΔP_{dyn} (%)	$P_{total_{opt}}$ (μA)	ΔP_{total} (%)
ARF	1.0	1607.6	308.9	65.3	857.1	308.9	81.5	4190.5	77.8
	1.1	1584.4	329.4	65.8	833.9	308.9	82.0	3629.7	80.8
	1.2	1334.2	362.2	71.2	657.8	360.4	85.8	3409.1	82.0
BPF	1.0	1217.3	290.1	66.7	650.7	290.1	82.2	432.4	70.9
	1.1	1195.4	310.7	67.3	606.8	290.1	83.4	377.9	74.5
	1.2	939.5	343.5	74.3	456.9	341.7	87.5	346.7	76.7
DCT	1.0	1617.9	308.9	61.1	994.0	308.9	76.1	543.2	67.9
	1.1	1530.5	308.9	63.2	956.6	308.9	77.0	439.0	74.0
	1.2	1384.9	341.7	66.7	944.1	308.9	77.3	399.5	76.3
EWF	1.0	1363.3	498.4	50.0	1036.2	498.4	62.0	299.6	73.0
	1.1	1205.2	531.2	55.8	1006.2	530.6	63.1	276.3	75.0
	1.2	891.6	584.5	67.3	725.3	582.2	73.4	260.1	76.5
FIR	1.0	810.3	282.4	67.6	465.2	303.0	81.4	313.8	69.1
	1.1	790.3	323.5	68.4	435.1	303.0	82.6	232.4	77.1
	1.2	772.8	344.1	69.2	172.6	354.5	93.1	217.5	78.6
MMV	1.0	852.8	163.8	52.1	559.1	163.8	68.6	216.1	70.1
	1.1	842.1	163.8	52.7	548.4	163.8	69.2	178.7	75.3
	1.2	813.6	184.4	54.3	535.9	196.0	69.9	151.2	79.1

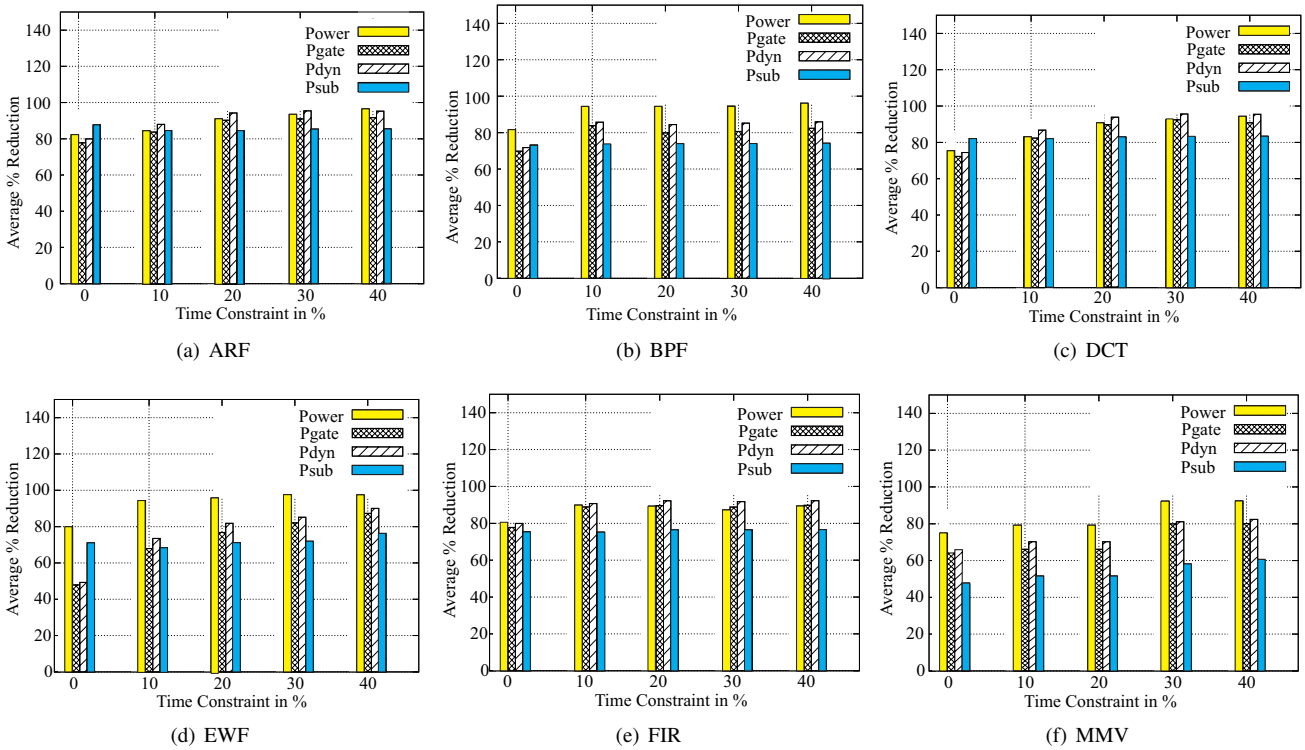
The experimental results are shown for selected RTL benchmarks in Fig. 16. For brevity, we present average percentage reduction data; the percentage reductions for each set were then averaged. The bar charts present graphical percentage data for four different time constraints with a DTF of 1.0 (or 0 %), 1.1 (or 10 %), 1.2 (or 20 %), 1.3 (or 30 %), and 1.4 (or 40 %), respectively. For each time constraint, the data is averaged for different resource constraints. We note that in all benchmarks, a $60-80\%$ reduction in total power can be achieved without any performance penalty. If a performance penalty is allowed in the algorithm, the total power reduction can be increased to 95% in some cases. Looking at the individual leakage components, we note the following:

- When a performance penalty is allowed in the form of a time constraint, in all cases the maximum reduction is achieved in dynamic power, followed by gate leakage with subthreshold leakage achieving the smallest reduction. This is consistent with the relative magnitude of the individual leakage components and is due to the fact that time constraints allow the use of low-leakage, low-performance functional units throughout the circuit, including the critical path.

- When a performance penalty is not allowed, there is no clear trend in the relative reduction of the individual components. In this case, the algorithm places high-leakage, high-performance functional units in the critical path. The overall reduction depends then on the relative number of critical vs. off-critical path components.

We also note that the ARF benchmark obtains the best results whereas the MMV benchmark shows the least improvement, assuming no delay penalty. This is consistent with the fact that the MMV is more complex (in terms of number of adders and multipliers) than the ARF and hence presents the algorithm with more optimization choices.

To the best of our knowledge, we did not find RTL research having the same scope as the one presented in this Section, i.e., accounting for dynamic power dissipation, subthreshold leakage, gate-oxide leakage together with process variation, and statistical optimization. In [50], power fluctuation is considered accounting for dynamic and leakage power dissipation. In [49], power and delay yield optimization is performed using multi-V_{th}/V_{dd}; however the current paper deals with all the forms of power including gate-oxide leakage and additionally uses T_{ox} as a tuning parameter. Hence, a fair comparison of the presented results of the current paper is not possible. The percentage reductions, the resulting penalty in delay, and the constraints need to be considered together to make appropriate judgment about the performance of the proposed research. In



(a) ARF (b) BPF (c) DCT

(d) EWF (e) FIR (f) MMV

Figure 16: Experimental results showing the percentage reduction in various leakage and dynamic power components for selected benchmarks. The X-axis represents various time constraints as percentage and the Y-axis represents percentage reduction. For each time constraint, the data is averaged for various resource constraints.

view of the low-power behavioral synthesis works, we provide a broader comparative perspective in Table 3. In this table, ΔP and ΔT_{pd} denote the percentage power reduction and percentage delay penalty, respectively, averaged over all constraints for a particular benchmark circuit. The data are provided wherever available. The work presented in [36] uses many different circuits than the rest of the works in Table 3; so we provide the overall average data. The work presented in [37] is area constrained so we did not get the delay penalty data. The results produced show that the dual-T_{ox} approach presented in this paper results in significant reduction in gate leakage with reasonable time penalty. This approach has outperformed the multi-V_{DD} approach for dynamic power reduction and multi-V_{th} for subthreshold leakage reduction at the same time with much lower delay penalty.

Table 3: A Broad Comparative Perspective with Existing Low-Power RTL Optimization Techniques.

Works →	Shiue [44]		Manzak [45]		Gopalakrishnan [37]		This Paper (Section 6)				
Power →	Dynamic Power				Subthreshold Leakage		Dynamic and Subthreshold/Gate-Leakage				
Method →	(Multi-V_{DD})				(Multi-V_{th})		(Judicious use of dual T_{ox}, V_{th}, or V_{DD})				
Benchmarks	ΔP	ΔT_{pd}	ΔP	ΔT_{pd}	ΔP	ΔT_{pd}	ΔP_{gate}	ΔP_{sub}	ΔP_{dyn}	ΔP_{total}	ΔT_{pd}
ARF	56.3	50.0	46.1	50.0	8.4	NA	85.9	82.9	87.5	86.1	20.0
DCT			34.1	50.0	NA	NA	83.1	80.0	87.1	87.1	20.0
EWF	56.3	50.0	35.7	50.0	19.7	NA	66.9	75.4	73.2	92.4	20.0
FIR			41.3	50.0	21.6	NA	84.6	75.1	87.1	85.0	20.0

7. Other Selected Nanoscale Process-Variation Aware Architectural-Level Techniques

7.1. Approaches for Variability Aware Power and Power-Fluctuation Minimization

One of the earliest approaches for variability aware power (including leakage) optimization at the architectural level is traced back to [50]. In this paper, resources of dual gate oxide thicknesses, dual threshold voltage, and dual power supply are considered. The statistical variations in these parameters are explicitly taken into account by using Monte Carlo simulations to characterize a datapath component library. The overall minimization problem reduces to the minimization of two cost

functions to facilitate minimization of total power consumption as well as total fluctuation in power consumption. The overall multicost objective function can then be expressed as:

$$\hat{\chi}_{PnF}^{DGF} = \text{Statistical-Difference}\left(\hat{\chi}_{P}^{DFG}, \hat{\chi}_{F}^{DFG}\right). \tag{21}$$

The minimization of total power (P) includes gate leakage, subthreshold leakage, and dynamic power of the datapath circuit. The total cycle-to-cycle power fluctuation minimization (F) cost function for the overall datapath circuit corresponding the DFG is calculated as follows:

$$\chi_{F}^{DFG} = \sum_{c=1}^{N_{cc}-1} |\text{Statistical-Difference}\left(P_{total}^{c}(\mu,\sigma), P_{total}^{c+1}(\mu,\sigma)\right)|, \tag{22}$$

where N_{cc} is the total number of cycles in the datapath. It may be noted that a more accurate estimate of fluctuation is a transient analysis in $\left(\frac{dI}{dt}\right)$, where the change in time is continuous. However, during the architectural-level design process the concept of time is manifested in the form of clock cycles and hence $\left(\frac{dI}{dt}\right)$ can be estimated as cycle-to-cycle difference as presented above for the architectural level. A simulated annealing based algorithm is proposed to minimize $\left(\hat{\chi}_{PnF}^{DGF}\right)$ during RTL tasks of simultaneous scheduling, allocation, and binding.

In [49], resource binding algorithm is proposed that uses multi-V_{th} and multi-V_{DD} library for power yield under a timing yield constraints. In essence, the process variation impact on multi-V_{th} and multi-V_{DD} RTL library based optimization is presented. The proposed methods could achieve power reduction as compared with traditional worst-case based deterministic approaches.However, gate leakage power and T_{ox} variations have not been accounted in the optimization.

7.2. Approaches for Variability Aware Timing Minimization

One of the earliest approaches for variability aware timing optimization at RTL is presented in [62]. The proposed approach relies on a simulated annealing based algorithm that guarantees timing yield under process variations. It iteratively invokes a statistical time analyzer while performing resource rebinding, operation shifting, and clock selection. For clock selection, average slack and event count product is calculated in which the average slack is based on the probability associated with a timing event.

Additional contributions to statistical timing optimization during architectural-level optimization are presented in [63, 64]. The key contribution is an algorithm for scheduling and resource binding tasks during RTL optimization with the objective of minimizing latency under yield constraint. Specifically, a branch-and-bound algorithm is proposed for scheduling and binding. The search in the algorithm is accelerated using a window-based search. In the algorithm, the window is defined as the maximum number of consecutive clock cycles satisfying resource constraints.

In [65] an algorithm is also proposed for variation-aware resource sharing and binding to improve performance yield. The performance yield of a circuit is defined as the probability that the target hardware meets the target performance constraints. The proposed algorithm has two forms. For performance optimization using the algorithm, the resource constraints is given as constraints and the performance yield is maximized. On the other hand, for area optimization using the algorithm, the performance yield requirement is given as constraint, and the area is minimized.

In [66], a variation-awarebinding/module selection algorithm is proposed. The algorithm relies on the lower-levels of the design hierarchy through a timing driven floor-planner guided by statistical static timing analysis. The algorithm incorporates spatial correlations of process variations in its optimization.

7.3. Post Silicon Techniques for Variability Tolerance

In [51], the authors claim that the chip yield can be increased by combining both the design time variation-aware optimization and post silicon tuning techniques, such as adaptive body biasing (ABB). The authors propose an algorithm that performs a module selection algorithm by combining the design-time optimization with ABB based postsilicon tuning to maximize design yield. The algorithm uses a two stage module selection approach. In the first stage, an iterative algorithm is used for power and timing variability aware module selection. The 2nd stage of the approach uses a sequential conic program to determine the optimal body bias for post-silicon tuning which influences design-time module selection.

In [67], an a run-time approach is proposed that detects critical path delay errors and recovers them at runtime. The proposed methodology handles common-case rather that worst-case critical path delays. In this approach, the classical design are modified to variable-tolerant design by introducing shadow logic to detect and recover from runtime errors. The designs results out of this approach have improved speed with minimal area penalty.

7.4. Integer Linear Programming (ILP) based Variability-Aware RTL Optimization

One of the earliest attempts for Integer Linear Programming (ILP) based process variation aware RTL optimization can be found in [68]. The authors present integer linear programming (ILP) formulations to enhance RTL timing variations by integrating overall timing yield constraints into scheduling and resource binding.

In [69], the authors present ILP formulations for the improvement of power yield. Specifically, a resource constraints ILP based algorithm is proposed to optimize leakage delay product (LDP). The optimization algorithm used dual-T_{ox} and dual-V_{th} technology for leakage optimization. A comparative perspective of these two technology for leakage optimization is presented.

In [70], the authors propose performance yield optimization for scheduling of RTL design process as an integer linear programming problem (ILP). The layout driven variation-aware binding algorithm ensures the yield enhancement. The ILP based algorithm maximizes the performance yield of the design for a given clock period and latency constraints.

7.5. A Taylor-Series Expansions Diagram (TED) Approach for Simultaneous Power and Delay Yield Optimization

In [52, 71], the authors have introduced a novel representation of DFG using a Taylor-Series Expansions Diagram (TED) to handle simultaneously power and delay yields at architectural-level. A TED, which was for the first time used for architectural-level optimization in this paper, is based on a non-binary decomposition principle that captures an entire class of structural solutions, rather than a single DFG. The TED is converted into a DFG by using decomposition which is optimized for a target design objective. After obtaining the DFG, statistical timing and power analysis follows to study the impact of process variation on propagation delay and leakage power. A variation-aware simultaneous scheduling and resource binding algorithm is presented to run on the DFG considering the performance constraints for leakage power yield enhancement.

8. Summary, Conclusions, and Future Research

8.1. Summary

The state-of-the art in architectural-level optimization of digital circuits addressing process variation is reasonably mature. Several statistical approaches have been presented to address yield for both power and timing. In particular, timing yield has received more attention. It is observed that, at present, low-power RTL optimization research mostly address dynamic power reduction only, while some works address subthreshold leakage only, and a few address gate-oxide leakage only. All of these forms of power reduction have been addressed individually but not simultaneously. Dual-V_{DD} methods only account for dynamic power consumption and do not consider gate-oxide leakage or subthreshold leakage. Dual-V_{th} methods only account for subthreshold leakage and do not consider gate-oxide leakage or dynamic power consumption. Dual-T_{gate} methods only account for gate-oxide leakage and do not consider dynamic power consumption or subthreshold leakage. Thus, independently they are inadequate to address the demand for power reduction in nano-CMOS circuits. If they are applied simultaneously without considering the interdependency of power and the parameters that affect it, they may not result in an optimal solution as becomes evident from the discussion in this paper. Thus, there is a need to develop optimization approaches that consider such interdependencies of parameters for scaling and judiciously use this scaling for global optimization. Moreover, the above discussed existing research works do not take process variation into consideration while doing power optimization which is very crucial for nano-CMOS circuits. Hence, a process-variation-aware statistical optimization approach addressing power yield is needed for nanoscale technologies which was discussed in this paper.

8.2. Conclusions

In this paper, sources and types of variability are discussed. An overview of several existing related research works was presented. As a demonstration of a concrete approach, a novel process-variation-aware statistical power characterization and optimization methodology was presented. An extensive functional unit model library was created by considering the individual and combined variations of T_{gate}, V_{th}, V_{DD} and L_{eff} via transistor-level Monte Carlo SPICE and fast SPICE simulations. The statistical variation of process and device parameters (assumed known) are thus transformed into a resulting characterization consisting of the mean and standard deviation of P_{dyn}, P_{sub}, and P_{gate} as well as propagation delay of the functional units and their correlations. The effect of scaling three parameters, T_{gate}, V_{th}, and V_{DD} on various power components was studied. It was observed that simultaneous, independent scaling of all three parameters may not result in the expected power-performance tradeoff, with the expectation based on the effect of individual parameter variations. Hence, power optimization techniques in circuit or process design, which resort to parameter selection/assignment techniques, need to do so judiciously. The proposed simulated annealing based algorithm that performs scheduling and binding is guided by these observations. Exhaustive experimentation with standard benchmark circuits proved that significant reduction in various components of power along with total power was achieved using the proposed methodology. Thus, the proposed algorithms, approach, and methodology can advance the state-of-the-art research in behavioral synthesis and can make them suitable to handle the challenges of complex nanoscale CMOS digital circuits.

8.3. Future Research Directions

Future research in architecture level optimization and design exploration can be in various fronts. It can be for different types of nanoelectronic technology as well as for different types of chips. We discuss selected options in this Section.

Existing architecture-level optimization methods are proposed for specific technology-based libraries and are not adaptable to technology change. They are primarily targeted towards CMOS and are not ready to meet the challenges of emerging technologies, such as double-gate FET (DGFET), Carbon Nano Tubes (CNT), Graphene nanoribbon FET (GNRFET) and Quantum Cellular Automata (QCA), which will eventually replace CMOS. Thus, the future phases of the current research will target double-gate FET (DGFET), and Graphene FET, which are being developed for low-leakage technology. RTL optimization considering hybrid nanoscale FET and carbon nanotube FET (CNTFET) interconnects needs further research.

The information which is processed by different chips and systems is often susceptible to copyright infringement and other security attacks. When watermarking and cryptography hardware is implemented using static nano-CMOS technology, it is susceptible to side channel attacks. These attacks gain information from the physical implementation of a cryptosystem rather than from theoretical weaknesses in the algorithms through timing information, power consumption, and electro- magnetic leaks. How RTL design exploration can be performed for encryption or watermarking chips with side-channel resilience as one design axis also needs research.

It is a fact that every current generation system, such as a mobile phone, is an analog/mixed-signal system-on-a-chip (AMS-SoC). However, existing tools do not handle their design exploration in a unified fashion (both analog and digital together). An important research direction of architecture-level design optimization is unified analog/mixed-signal (AMS) systems at the architecture-level for early design exploration.

Architectural level design exploration for micro/nano-electro-mechanical systems (MEMS/NEMS) needs research as they have significant applications in health monitoring and are very expensive to custom design. The first level of challenge is how to come up with a unified electrical and non-electrical component representation of the NEMS using a hardware description language (HDL).

Acknowledgements

S. P. Mohanty and E. Kougianos acknowledge NSF award DUE-0942629 for partial support for this paper. Some preliminary results of this research were presented in the conference paper [50].

References

[1] F. Tobajas, G. Callico, P. Perez, V. de Armas, R. Sarmiento, An Efficient Double-Filter Hardware Architecture for H.264/AVC Deblocking Filtering, IEEE Transactions on Consumer Electronics 54 (1) (2008) 131–139. doi:10.1109/TCE.2008.4470035.

[2] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, http://public.itrs.net.

[3] S. P. Mohanty, Unified Challenges in Nano-CMOS High-Level Synthesis, in: Proceedings of the International Conference on VLSI Design, 2009, pp. 531–531.

[4] T. Chantem, R. P. Dick, X. S. Hu, Temperature-Aware Scheduling and Assignment for Hard Real-Time Applications on MPSoCs, in: Proceedings of the Design, Automation and Test in Europe (DATE), 2008, pp. 288–293.

[5] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, N. J. Rohrer, High-Performance Cmos Variability In The 65-nm Regime And Beyond, IBM Journal of Research and Development 50 (4-5) (2006) 433–450.

[6] S. P. Mohanty, N. Ranganathan, E. Kougianos, P. Patra, Low-Power High-Level Synthesis for Nanoscale CMOS Circuits, Springer, 2008.

[7] J. G. Hansen, Design of CMOS Cell Libraries for Minimal Leakage Currents, Master's thesis, Dept. of Informatics and Mathematical Modelling, Computer Science and Engineering Technical University of Denmark (Fall, 2004).

[8] K. Tiri, P. Schaumont, I. Verbauwhede, Side-Channel Leakage Tolerant Architectures, in: Proceedings of the Third International Conference on Information Technology: New Generations (ITNG), 2006, p. 204209.

[9] S. P. Mohanty, A Secure Digital Camera Architecture for Integrated Real-time Digital Rights Management, Journal of Systems Architecture - Embedded Systems Design 55 (10-12) (2009) 468–480.

[10] S. P. Mohanty, N. Pati, E. Kougianos, A Watermarking Co-processor for New Generation Graphics Processing Units, in: Digest of Technical Papers International Conference on Consumer Electronics, IEEE, 2007, pp. 1–2.

[11] S. P. Mohanty, N. Ranganathan, K. Balakrishnan, Design of a Low Power Image Watermarking Encoder Using Dual Voltage and Frequency, in: Proceedings of the 18th International Conference on VLSI Design, 2005, pp. 153–158.

[12] J. Mathew, S. Banerjee, H. Rahaman, D. K. Pradhan, S. P. Mohanty, A. M. Jabir, On the Synthesis Of Attack Tolerant Cryptographic Hardware, in: Proceedings of the 18th IEEE/IFIP International Conference on Very Large Scale Integration of System-on-Chip (VLSI-SoC), 2010, pp. 286–291.

[13] V. H. Tuzel, A Level Set Method for an Inverse Problem Arising in Photolithography, Ph.D. thesis, The University Of Minnesota (July 2009).

[14] H. yu Chen, Y. wen Chang, Routing For Manufacturability And Reliability, IEEE Circuits and Systems Magazine 9 (3) (2009) 20–31. doi:10.1109/MCAS.2009.933855.

[15] P. G. Drennan, C. C. McAndrew, Understanding MOSFET mismatch for analog design, IEEE Journal of Solid-State Circuits 38 (3) (2003) 450–456.

[16] K. Singhal, Parametric Process Variations, personal Communication and Synopsys Booth at Design Automation Conference (DAC) (June 2007).

[17] L. Xie, A. Davoodi, Fast and Accurate Statistical Static Timing Analysis with Skewed Process Parameter Variation, in: Proceedings of the 9th International Symposium on Quality Electronic Design, 2008, pp. 712 –717. doi:10.1109/ISQED.2008.4479825.

[18] K. Roy, S. Mukhopadhyay, H. M. Meimand, Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits, Proceedings of the IEEE 91 (2) (2003) 305–327.

[19] S. P. Mohanty, E. Kougianos, Modeling and Reduction of Gate Leakage during Behavioural Synthesis of NanoCMOS Circuits, in: Proc. of International Conference on VLSI Design, 2006, pp. 83–88.

[20] J. A. Butts, G. S. Sohi, A Static Power Model for Architects, in: Proceedings of the 33rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-33), 2000, pp. 191–201.

[21] S. P. Mohanty, N. Ranganathan, Simultaneous Peak And Average Power Minimization During Datapath Scheduling, IEEE Transactions on Circuits and Systems I: Regular Papers 52 (6) (2005) 1157–1165.

[22] A. J. Bhavnagarwala, B. L. Austin, K. A. Bowman, J. D. Meindl, A Minimum Total Power Methodology for Projecting Limits of CMOS GSI, IEEE Transactions on VLSI Systems 8 (3) (2000) 235–251.

[23] K. A. Bowman, L. Wang, X. Tang, J. D. Meindl, A Circuit-Level Perspective of the Optimum Gate Oxide Thickness, IEEE Transactions on Electron Devices 48 (8) (2001) 1800–1810.

[24] S. P. Mohanty, E. Kougianos, O. Okobiah, Optimal Design of a Dual-oxide Nano-CMOS Universal Level Converter for Multi-V_{dd} SoCs, Analog Integrated Circuits and Signal Processing 72 (2012) 451–467. doi:10.1007/s10470-012-9887-7.
URL http://dx.doi.org/10.1007/s10470-012-9887-7

[25] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, R. W. Brodersen, Optimizing Power using Transformations, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 14 (1) (1995) 12–31.

[26] N. K. Jha, Low Power System Scheduling and Synthesis, in: Proceedings of the International Conference on Computer-Aided Design, 2001, pp. 259–263.

[27] S. P. Mohanty, N. Ranganathan, Energy Efficient Datapath Scheduling using Multiple Voltages and Dynamic Clocking, ACM Transactions on Design Automation of Electronic Systems (TODAES) 10 (2) (2005) 330–353.

[28] D. Dal, N. Mansouri, Power optimization with power islands synthesis, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 28 (7) (2009) 1025 –1037. doi:10.1109/TCAD.2009.2020717.

[29] D. Helms, O. Meyer, M. Hoyer, W. Nebel, Voltage- and abb-island optimization in high level synthesis, in: Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), 2007, pp. 153–158. doi:10.1145/1283780.1283814.

[30] V. Mukherjee, S. P. Mohanty, E. Kougianos, A Dual Dielectric Approach for Performance Aware Gate Tunneling Reduction in Combinational Circuits, in: Proceedings of the IEEE International Conference on Computer Design (ICCD), 2005, pp. 441–443.

[31] A. K. Sultania, D. Sylvester, S. S. Sapatnekar, Tradeoffs Between Gate Oxide Leakage and Delay for Dual T_{ox} Circuits, in: Proceedings of Design Automation Conference, 2004, pp. 761–766.

[32] D. Lee, D. Blaauw, D. Sylvester, Gate Oxide Leakage Current Analysis and Reduction for VLSI Circuits, IEEE Transactions on VLSI Systems 12 (2) (2004) 155–166.

[33] N. Sirisantana, K. Roy, Low-power Design using Multiple Channel Lengths and Oxide Thicknesses, IEEE Design and

Test of Computers 21 (1) (2004) 56–63.

[34] P. Pant, R. K. Roy, A. Chattejee, Dual-Threshold Voltage Assignment with Transistor Sizing for Low Power CMOS Circuits, IEEE Transactions on VLSI Systems 9 (2) (2001) 390–394.

[35] R. M. Rao, J. L. Burns, R. B. Brown, Circuit Techniques for Gate and Sub-Threshold Leakage Minimization in Future CMOS Technologies, in: Proceedings of the European Solid-State Circuits Conference, 2003, pp. 313–316.

[36] K. S. Khouri, N. K. Jha, Leakage power analysis and reduction during behavioral synthesis, IEEE Transactions on VLSI Systems 10 (6) (2002) 876–885.

[37] C. Gopalakrishnan, S. Katkoori, Knapbind: an area-efficient binding algorithm for low-leakage datapaths, in: Proceedings of 21st International Conference on Computer Design, 2003, pp. 430–435.

[38] X. Tang, H. Zhou, P. Banerjee, Leakage power optimization with dual-V_{th} library in high-level synthesis, in: Proceedings of the 42nd annual conference on Design automation, 2005, pp. 202–207.

[39] M. Liu, W. S. Wang, M. Orshansky, Leakage Power Reduction by Dual-Vth Designs Under Probabilistic Analysis of Vth Variation, in: Proceedings of International Symposium on Low Power Electronics and Design, 2004, pp. 2–7.

[40] B. Ameliford, F. Fallah, M. Pedram, Reducing the Sub-threshold and Gate-tunneling Leakage of SRAM Cells using Dual-Vt and Dual-Tox Assignment, in: Proceedings of Design Automation and Test in Europe, 2006, pp. 1–6.

[41] A. Kumar, M. Anis, Dual-Vt Design of FPGAs for Subthreshold Leakage Tolerance, in: Proceedings of International Symposium on Quality Electronic Design, 2006.

[42] L. W. et.al., Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications, IEEE Transactions on VLSI Systems 7 (1) (1999) 16–24.

[43] S. P. Mohanty, N. Ranganathan, V. Krishna, Datapath Scheduling Using Dynamic Frequency Clocking, in: Proceedings of the IEEE Computer Society Annual Symposium on VLSI, IEEE, 2002, pp. 58–63.

[44] W. T. Shiue, C. Chakrabarti, Low-Power Scheduling with Resources Operating at Multiple Voltages, IEEE Transactions on Circuits and Systems-II : Analog and Digital Signal Processing 47 (6) (2000) 536–543.

[45] A. Manzak, C. Chakrabarti, A Low Power Scheduling Scheme with Resources Operating at Multiple Voltages, IEEE Transactions on VLSI Systems 10 (1) (2002) 6–14.

[46] S. H. Kulkarni, D. Sylvester, High Performance level Conversion for Dual VDD Design, IEEE Transactions on VLSI Systems 12 (9) (2004) 926–936.

[47] C. Ping-Yuan, Y. Chien-Cheng, A Voltage Level Converter Circuit Design with Low Power Consumption, in: Proceedings of the 6th International Conference on ASIC, 2005, pp. 358–359.

[48] Y. Xie, Y. Chen, Statistical High-Level Synthesis under Process Variability, IEEE Design Test of Computers 26 (4) (2009) 78–87. doi:10.1109/MDT.2009.85.

[49] Y. Chen, Y. Wang, Y. Xie, A. Takach, Parametric Yield-Driven Resource Binding in High-Level Synthesis with Multi-V_{th}/V_{dd} Library and Device Sizing, Journal of Electrical and Computer Engineering 2012. doi:http://dx.doi.org/10.1155/2012/105250.

[50] S. P. Mohanty, E. Kougianos, Simultaneous Power Fluctuation and Average Power Minimization during Nano-CMOS Behavioral Synthesis, in: Proceedings of the 20th International Conference on VLSI Design, 2007, pp. 577–582.

[51] F. Wang, X. Wu, Y. Xie, Variability-driven module selection with joint design time optimization and post-silicon tuning, in: Proceedings of the Asia and South Pacific Design Automation Conference, 2008, pp. 2–9. doi:10.1109/ASPDAC.2008.4483963.

[52] S. Banerjee, J. Mathew, S. P. Mohanty, D. K. Pradhan, M. J. Ciesielski, A Variation-Aware Taylor Expansion Diagram-Based Approach for Nano-CMOS Register-Transfer Level Leakage Optimization, Journal of Low Power Electronics 7 (4) (2011) 471–481.

[53] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[54] V. Cerny, A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm, Journal of Optimization Theory and Applications 45 (1) (1985) 41–51.

[55] S. Devadas, A. R. Newton, Algorithms for hardware allocation in data path synthesis, IEEE Transactions on CAD of Integrated Circuits and Systems 8 (7) (1989) 768–781.

[56] S. P. Mohanty, R. Velagapudi, E. Kougianos, Physical-Aware Simulated Annealing Optimization of Gate Leakage In Nanoscale Datapath Circuits, in: Proceedings of the Conference on Design, Automation and Test in Europe, 2006, pp. 1191–1196.

[57] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, C. Hu, New Paradigm of Predictive MOSFET and Interconnect Modeling

for Early Circuit Design, in: Proceedings of the IEEE Custom Integrated Circuits Conference, 2000, pp. 201–204.

[58] R. J. Baker, H. W. Li, D. E. Boyce, CMOS: Circuit Design, layout, and Simulation, IEEE Press, 1998.

[59] S. P. Mohanty, N. Ranganathan, S. Chappidi, ILP Models For Energy And Transient Power Minimization During Behavioral Synthesis, in: Proceedings of the 17th International Conference on VLSI Design, IEEE, 2004, pp. 745–748.

[60] S. P. Mohanty, N. Ranganathan, A Framework for Energy and Transient Power Reduction during Behavioral Synthesis, IEEE Transactions on VLSI Systems 12 (6) (2004) 562–572.

[61] Express: High-Level Synthesis Bechmarks, http://express.ece.ucsb.edu/benchmark/.

[62] W. Hung, X. Wu, Y. Xie, Guaranteeing Performance Yield in High-Level Synthesis, in: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, 2006, pp. 303–309. doi:10.1109/ICCAD.2006.320050.

[63] J. Jung, T. Kim, Timing Variation-Aware High-Level Synthesis, in: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, 2007, pp. 424–428. doi:10.1109/ICCAD.2007.4397302.

[64] J. Jung, T. Kim, Scheduling and Resource Binding Algorithm Considering Timing Variation, Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 19 (2) (2011) 205–216. doi:10.1109/TVLSI.2009.2031676.

[65] F. Wang, Y. Xie, A. Takach, Variation-Aware Resource Sharing and Binding in Behavioral Synthesis, in: Proceedings of the Asia and South Pacific Design Automation Conference, 2009, pp. 79–84. doi:10.1109/ASPDAC.2009.4796445.

[66] G. Lucas, S. Cromar, D. Chen, Fastyield: Variation-aware, layout-driven simultaneous binding and module selection for performance yield optimization, in: Design Automation Conference, 2009. ASP-DAC 2009. Asia and South Pacific, 2009, pp. 61–66. doi:10.1109/ASPDAC.2009.4796442.

[67] A. Muttreja, S. Ravi, N. K. Jha, Variability-Tolerant Register-Transfer Level Synthesis, in: Proceedings of the International Conference on VLSI Design, 2008, pp. 621–628. doi:10.1109/VLSI.2008.114.

[68] Y. Chen, J. Ouyang, Y. Xie, ILP-Based Scheme For Timing Variation-Aware Scheduling and Resource Binding, in: Proceedings of the IEEE International SoC Conference, 2008, pp. 27–30. doi:10.1109/SOCC.2008.4641473.

[69] S. P. Mohanty, B. K. Panigrahi, ILP Based Leakage Optimization during nano-CMOS RTL Synthesis: A DOXCMOS Versus DTCMOS Perspective, in: Proceedings of the International Symposium on Biologically Inspired Computing And Applications, World Congress on Nature Biologically Inspired Computing, 2009, pp. 1367–1372. doi:10.1109/NABIC.2009.5393744.

[70] G. Lucas, D. Chen, Variation-Aware Layout-Driven Scheduling For Performance Yield Optimization, in: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2010, pp. 17–24. doi:10.1109/ICCAD.2010.5654344.

[71] S. Banerjee, J. Mathew, D. Pradhan, S. P. Mohanty, M. Ciesielski, Variation-Aware TED-Based Approach for Nano-CMOS RTL Leakage Optimization, in: Proceedings of the 24th International Conference on VLSI Design, 2011, pp. 304–309. doi:10.1109/VLSID.2011.40.

Saraju P. Mohanty is currently an Associate Professor at the Department of Computer Science and Engineering, University of North Texas, and the Director of the NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu). He obtained Ph.D. in Computer Science and Engineering from the University of South Florida in 2003, Masters degree in Systems Science and Automation from the Indian Institute of Science, Bangalore, India in 1999, and Bachelors degree (with Honors) in Electrical Engineering from Orissa University of Agriculture and Technology, Bhubaneswar, India in 1995. His research interest is in "Low-Power High-Performance Nanoelectronics". His funded by National Science Foundation and Semiconductor Research Corporation. He is an author of 150+ peer-reviewed journal and conference publications, and 2 books. The publications are well-received by the world-wide peers with a total of 1300+ citations resulting in an H-index of 20 and i10-index of 40 (from Google scholar). He has supervised 22 dissertations (Ph.D.) and theses (Masters) and the students are well-placed in industry and academia. He has received recognition as an inspirational faculty at UNT for the years 2008, 2009, 2011, and 2012. He serves on the organizing/program committee of several international conferences and editorial board of several international journals. He is a senior member of IEEE and ACM.

Mahadevan Gomathisankaran is an Assistant Professor in Computer Science and Engineering at the University of North Texas. He received his Ph.D. degree in Computer Engineering from Iowa State University. He is the recipient of IBM Ph.D. Fellowship award for the academic years 2004 and 2005. Mahadevan is interested in building secure computing systems. Towards that goal he has designed various cryptographic functions that achieve the required security with minimal circuit complexity, proposed new secure processor architecture that root the security in the hardware, and designed a testing framework that can test the security of the systems. He has published more than 20 articles in leading journals and conferences. He is an Associate Editor for the journal "Information Systems Security: A Global Perspective". He has served in technical program committees of several international conferences.

Elias Kougianos is currently an Associate Professor in the Department of Engineering Technology, at the University of North Texas (UNT), Denton, TX. He received a BSEE from the University of Patras, Greece in 1985 and an M.S. (EE) in 1987, an M.S. in Physics in 1988 and a Ph.D. in EE in 1997, all from Lousiana State University. From 1988 through 1997 he was with Texas Instruments, Inc., in Houston and Dallas, TX. Initially he concentrated on process integration of flash memories and later as a researcher in the areas of Technology CAD and VLSI CAD development. In 1997 he joined Avant! Corp. (now Synopsys) in Phoenix, AZ as a Senior Applications engineer and in 2001 he joined Cadence Design Systems, Inc., in Dallas, TX as a Senior Architect in Analog/Mixed-Signal Custom IC design. He has been at UNT since 2004. His research interests are in the area of Analog/Mixed-Signal/RF IC design and simulation and in the development of VLSI architectures for multimedia applications. He is author or co-author of over 70 peer-reviewed journal and conference publications. He is a senior member of IEEE.