Process Variation Aware Simultaneous Leakage and Dynamic Power Minimization during Nano-CMOS Behavioral Synthesis

Saraju P. Mohanty, Member of IEEE, Elias Kougianos, Senior Member of IEEE, and Priyadarsan Patra, Senior Member of IEEE

***Abstract*— In nano-CMOS technology, capacitive switching power, gate-oxide leakage, and subthreshold leakage are each significant and one can not be ignored with respect to the other. To achieve power-performance trade-offs different research works have been proposed in the high-level synthesis literature including scaling of various process and design parameters. These approaches handle the optimization of these different components independently and/or do not effectively account for the variation of different process and design parameters. Thus, they do not result in a power and performance optimal circuit that has minimal gate-oxide leakage, minimal subthreshold leakage, and minimal dynamic power consumption for a target performance or area. A resource-time constrained algorithm is proposed in this paper for simultaneous scheduling, binding, and allocation for reduction of the total power (both leakage and dynamic) while taking into account process variation during behavioral synthesis. Assuming dual values of T_{ox}, V_{th}, and V_{DD} for a particular K and L, the values for gate-oxide leakage, subthreshold leakage, dynamic power, and performance are estimated for architectural units such as adder, multiplexor, and multiplier, etc. Statistical variations in the parameters, each assumed to be Gaussian, 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 in essence gives a relative and integrated perspective of various power-performance tradeoffs against the nominal case, thus serving as a guideline to help designers to take appropriate decisions. Experiments on several standard benchmarks show a significant reduction in gate-oxide and subthreshold leakage, dynamic, and total power dissipation. To the best of the authors' knowledge, this is the first-ever behavioral synthesis work simultaneously addressing gate-oxide and subthreshold leakage and dynamic power together with process variation.**

S. P. Mohanty is with Computer Science and Engineering, University of North Texas, Denton, TX 76203, E-mail: smohanty@cs.unt.edu. Elias Kougianos is with Electrical Engineering Technology, University of North Texas, Denton, TX 76203, E-mail: eliask@unt.edu. Priyadarsan Patra is with Microprocessor Technology Labs, Intel Corporation, Portland, OR 97124, Email: priyadarsan.patra@intel.com.

This archival journal paper is based on our following shorter peer-reviewed conference versions:
S. P. Mohanty and E. Kougianos, "Simultaneous Power Fluctuation and Average Power Minimization during Nano-CMOS Behavioral Synthesis", in *Proceedings of the 20th IEEE International Conference on VLSI Design (VLSID)*, pp. 577-582, 2007.
S. P. Mohanty, E. Kougianos, D. Ghai, and P. Patra, "Interdependency Study of Process and Design Parameter Scaling for Power Optimization of Nano-CMOS Circuits under Process Variation", in *Proceedings of the 16th ACM/IEEE International Workshop on Logic and Synthesis (IWLS)*, pp. 207-213, 2007.

This work is partially supported by NSF award number 0702361.

I. Introduction and Motivation

Today an increasing number of electronic devices are being designed to be small and mobile thus heavily relying on battery power for portability. Power dissipation is an important design constraint in high performance processor systems, system on a chip (SoC) designs as well as application specific integrated circuits. To meet the increasing demand of low-power VLSI circuits with high performance and higher integration density and functionality of digital devices, VLSI design engineers are resorting to relentless scaling in process as well as design parameters of CMOS transistors. However, this has resulted in a number of new concerns which include a new reality of leakage current distribution. Both dynamic and static power are significant fractions of total power dissipation in a nanoscale CMOS circuit.

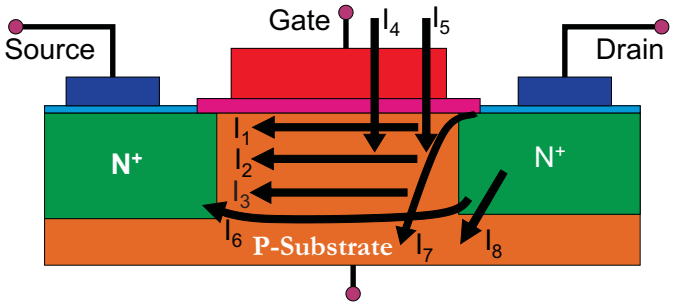


Fig. 1. Current flow paths in a nanoscale CMOS transistor during power dissipation in different states of its operation [1], [2], [3]: I_1 - drain to source active current (ON state), I_2 - drain to source short circuit current (ON state), I_3 - subthreshold leakage (OFF state), I_4 - gate leakage (both 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).

In short channel nano-CMOS transistors, several short channel effects (SCE) 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, there has been a drastic change in the leakage components of the device both in the inactive as well as active mode 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, and channel punch through current [1], [2], [3]. Each one of them has several forms and origins; they flow between different terminals and in different operating conditions of a transistor as shown in Fig. 1. While biased diode leakage and SiO_2 tunnel currents flow during both active and sleep mode of the circuit, the other currents flow during the sleep mode only.

The ITRS prediction of major sources of power dissipation, such as dynamic current, subthreshold leakage and gate leakage is presented in Fig. 2 [4], [5]. The major sources of power dissipation in a nanoscale CMOS circuit can be summarized as follows [2], [6], [7], [1]:

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

Thus, 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}).

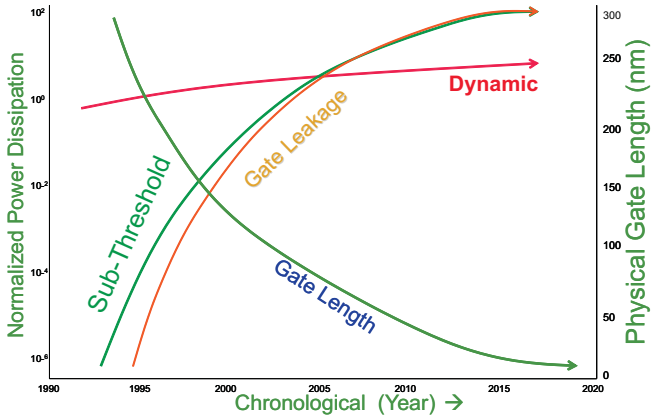


Fig. 2. Prediction of trends of major sources of power dissipation in nanoscale CMOS transistors and circuits [4], [5]. It is predicted that gate-oxide leakage and subthreshold leakage will dominate the total power dissipation of nano-CMOS circuits and hence need designers attention.

The prominent current/power components predominantly depend on the gate oxide thickness (T_{ox}), threshold voltage (V_{th}), supply voltage (V_{DD}), and device length ($L_(e)$). Hence, any methodology for power reduction must focus on the variation of these process and design parameters. This has been the motivating factor to consider process variation during high-level synthesis and facilitate fast and correct design space exploration right at the early stages of the design cycle. 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 due to increasing complexity. Our methodology consistently does so and incorporates the variation in the model for average power. Our reduction methodology also considers several possible design corners in a resource and time constrained approach by optimizing a multicost objective function. We provide statistically characterized gate and functional unit models which were simulated at transistor level for obtaining the mean (μ) and standard deviation (S.D. σ) of gate-oxide leakage, subthreshold leakage, capacitive-switching current and propagation delay with simultaneous variation of all process and design parameters. Although $L_(e)$ is an important parameter, we will not consider it in the present paper for brevity; however, we note that the presented methodology can be easily extended, at the expense of additional runtime, to include $L_(e)$ scaling and variation.

To achieve power-performance trade-offs different research works have been proposed in the high-level synthesis literature including scaling of various process and design parameters, such as T_{ox}, K, L, V_{th}, and V_{DD} through the use of technologies such as dual-V_{DD}, dual-V_{th}, etc. These research handle the optimization of various components independently and without addressing the variation of various process and design parameters in the nanoscale CMOS regime. In view of the optimization regime above, the following question arises: (i) If T_{ox}, L, V_{th}, and 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 ? (ii) How the results will be affected due to process variation? (iii) Given architectural constraints, how to judiciously scale such that a global power and performance optimal circuit is obtained? The research proposed in this paper is further motivated by these important facts. A resource-time constrained algorithm is proposed in this paper for simultaneous scheduling, binding, and allocation for the reduction of total (accounting leakage and dynamic) power and process variation during behavioral synthesis that judiciously uses dual-T_{ox}, dual-V_{th}, and dual-V_{DD} technology.

The rest of the paper is organized as follows. In Section II the novelty and contributions of this paper is outlined. Section III summarizes relevant prior research works that consider various forms of power optimization using dual-T_{ox}, dual-V_{th}, and dual-V_{DD} technology. In Section IV We briefly describe some specific research issues and challenges arising in the nanoscale CMOS regime. The optimization problem formulation is presented in Section V. Section VI propose a new framework for statistical behavioral synthesis needed to handle nano-CMOS regime circuits. In Section VII, a hierarchical methodology to characterize architectural level units for gate leakage, subthreshold leakage, and dynamic power, as well as their propagation delay is presented along with investigation of the effects of process variation on scaling. The proposed optimization approach during high-level synthesis is presented in Section VIII. Section IX discusses the experimental results. The paper concludes in Section X.

II. Novel Contributions of this Paper

The contributions of this paper, all relating to fast and effective high-level synthesis of nanoscale CMOS circuits, are in multiple forms as summarized below.

- A statistical high-level synthesis framework is presented that can perform different forms of power optimization while accounting for process variation.
- A novel methodology is proposed to characterize nano-CMOS based architectural components for gate-oxide leakage, subthreshold leakage, and dynamic power while simultaneously accounting for process variation. Statistical variations in the parameters are explicitly taken into account by using Monte Carlo simulations while characterizing the architectural units.

- We rovide a comparative and integrated perspective of various power-performance tradeoffs weighed against a nominal case, thus serving as a guideline to help designers take effective decisions. The interdependency of T_{ox}, V_{th}, and V_{DD} scaling on various power (current) components is analyzed with and without process variation. Seven different cases, such as (1) only T_{ox} scaling, (2) only V_{th} scaling, (3) only V_{DD} scaling, (4) simultaneous T_{ox} and V_{th} scaling, (5) simultaneous T_{ox} and V_{DD} scaling, (6) simultaneous V_{th} and V_{DD} scaling, and (7) simultaneous T_{ox} and V_{th} and V_{DD} scaling, are analyzed for various forms of power dissipation.
- An efficient 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, and process variation during behavioral synthesis.

III. Related Prior Research Works

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 opportunity, reduce. So high-level or behavioral level is an attractive level and provides balanced degree of freedom for design space exploration. Several techniques such as, architecture-driven voltage scaling, operation reduction and substitution, precomputation, and clock-gating have been proposed [8], [9], [10]. In addition, technology dependent techniques, such as dual-T_{ox}, dual-V_{th}, and dual-V_{DD} have been proposed that consider various form of scaling for power optimization. However, these scaling methods have been applied relatively independently. For example, researchers apply dual-V_{th} to reduce subthreshold leakage without studying its impact on gate leakage and dynamic power. On the other hand, the goal of this research is to study the interdependency of these scaling methods from the power (current) and performance (propagation delay) point of view.

A. Dual-T_{ox}/K Research

In [11] Mukherjee *et al.* have proposed a gate oxide leakage minimization approach using dual-T_{ox} and dual-K. Mohanty *et al.* in [12] have presented analytical models and a datapath scheduling algorithm for reduction of tunneling current. The heuristic assigns higher thickness resources to more leaky nodes (multipliers), but does not address the area overhead. In [13], Lee *et al.* developed a method for analyzing gate oxide leakage current in logic gates and suggested utilizing pin reordering to reduce gate leakage. Sultania *et al.* in [14], developed an algorithm to optimize the total leakage power by assigning dual T_{ox} values to transistors in a given circuit. In [15] Sirisantana and Roy use multiple channel lengths and multiple gate oxide thickness for reduction of leakage. In [16] Mukhopadhyay *et al.* have carried out extensive modeling and estimation of total leakage current of CMOS devices considering the effect of parameter variation.

B. Dual-V_{th} Research

Multiple threshold CMOS have been used by Pant *et al.*. [17] as well as Rao *et al.* [18] for subthreshold current reduction. Khouri and Jha [19] proposed 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 [20], [21] also use the multi threshold CMOS approach for reduction of subthreshold current during high-level synthesis and propose binding algorithms for power, delay, and area trade-off. They used a clique partitioning approach in [21] and a knapsack based binding algorithm in [20]. In [22], Liu *et al.* have applied probabilistic analysis to V_{th} variation. The analysis of dual V_{th} design methodology is done in the presence of large variations in threshold voltage. In [23], a dual V_{th} and dual T_{ox} technique is applied to SRAMs in order to reduce leakage. In [24], a dual V_{th} FPGA architecture has been proposed in which the logic elements are used for dual V_{th} assignment. In [25], Wei *et al.* have tried to reduce the leakage power by using high V_{th} transistors in the non-critical paths, and low V_{th} transistors in the critical paths.

C. Dual-V_{DD} Research

The research using this techniques is quite mature and several approaches have been proposed in the literature over the last several years [10], [26], [27], [28], [29]. A certain type of circuitry called voltage-level converters are used for this purpose, but in turn it is an overhead for this kind of technology. The transistors on critical paths are operated on a higher supply voltage (V_{DDH}), whereas transistors on the non-critical paths are operated on a lower supply voltage (V_{DDL}) [30] and [31].

D. Summary and Observation from Prior Research Works

In summary we observe that at present, low power high-level synthesis works mostly address dynamic power reduction only, while some of them 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. The dual-V_{DD} methods only account for dynamic power consumption and do not consider either gate-oxide leakage or subthreshold leakage. The dual-V_{th} methods only account for subthreshold leakage and do not consider either gate-oxide leakage or dynamic power consumption. The dual-T_{ox} methods only account for gate-oxide leakage and do not consider either 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 causing parameters, they may not results in optimal solution as will become evident from discussions in this paper. Thus, there is a need for development of optimization approaches that consider such interdependencies of parameters to be scaled and judiciously use scaling for global optimization. Moreover, the above discussed existing research works do not take process variation

into consideration which is very crucial for nano-CMOS circuits. Hence, process variation aware statistical optimization approach is needed for the new technologies.

IV. Process Variation Issues and Challenges and Our Solution for High-Level Synthesis

To facilitate fabrication of circuits using nanoscale CMOS technology more and more sophisticated lithographic, chemical, and mechanical processing steps are adopted. The uncertainty in the processes, such as ion implantation, chemical mechanical polishing (CMP), chemical vapor deposition (CVD), etc. involved in nano-CMOS fabrication has caused variations in process parameters [32], [33], [34] such as channel length, gate-oxide thickness, threshold voltage, metal wire thickness, via resistance, etc. These variations are categorized as inter-die variations and intra-die (systematic or random) and are treated as global, local, and spatial variations. The above process variations have profound effect on electrical parameter and performance variations in a VLSI circuit. The situation is further worsened due to variations of temperature, power supply voltage, wear-out, and use history. All these are manifested in the variation in power and delay, and other attributes of CMOS circuits. These variations can be either temporal or spatial in nature [32]. They ultimately affect design margins and yield and may lead to loss of money in the ever reducing time-to-market scenario.

The major challenge arising in the nanoscale CMOS variation scenario is the correct understanding of the process variations and their modeling. Without proper models of variations, designers will include substantial design margin or risk yield loss when they use traditional CAD tools not accounting 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 leads to the need of accounting for process variation during characterization and modeling and also integrating process variation in design and synthesis frameworks. Moreover, designing for the worst case scenario may cause severe compromises on the performance of the device. This has led us to consider a process variation aware power minimization technique. We perform statistical estimation of the various power-performance components considering statistically varying process and design parameter 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. The challenges posed in such a scenario are as follows: (i) How to model variations at the lower level of circuit abstraction? (ii) How to estimate power and performance at the higher levels of abstraction while accounting for variations? (iii) How to optimize power and performance at the higher level of design abstraction while accounting for the variations to enable design space exploration?

The following solution to account for the variations from the lower level of design abstraction accurately is proposed to enable fast design space exploration and optimization at higher levels of abstraction. Following a hierarchical approach we propose to pre-characterize datapath components (functional units) in which the variations would be faithfully propagated from the lower level to the higher levels of abstraction. Then express the power and performance attributes of these functional units as probability density functions instead of single valued functions of parameters. Finally, develop statistical optimization approaches in which mean (μ) and standard deviations (σ) of probability density functions would be considered while performing various tasks of high-level synthesis such as scheduling or binding, and allocation.

V. Optimization Problem Formulation and Our Proposed Solution for Behavioral Synthesis

Let us assume that the datapath is specified as a sequencing data flow graph (DFG) [35]. 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, we assume that the DFG has a single source node and a single sink node as is the case of sequencing data flow graph [35], which is a directed acyclic graph.

The problem can be then 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.

The above can be stated as an optimization problem. The cost (resource constrained) and performance (latency constrained) driven power (current) minimization problem can thus be formulated as follows:

$$\text{Minimize} \quad I_{total}^{DFG}\left(\mu_I^{DFG}, \sigma_I^{DFG}\right), \tag{2}$$

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

$$\text{Allocated } (FU_{k,i}) \leq \text{Available } (FU_{k,i}) \mid \forall \text{ clock cycle } c, \tag{3}$$

$$D_{CP}^{DFG}\left(\mu_D^{DFG}, \sigma_D^{DFG}\right) \leq D_C\left(\mu_D^C, \sigma_D^C\right). \tag{4}$$

$I_{total}^{DFG}\left(\mu_I^{DFG}, \sigma_I^{DFG}\right)$ in Eqn. 2 represents the probability density function (PDF) of the total current dissipation due to the DFG, which can be presented as an equally weighted sum of probability density functions (PDF) of all current (gate-oxide I_{gate}, subthreshold I_{sub}, and dynamic I_{dyn}) components as presented below:

$$I_{total}^{DFG} = I_{gate}^{DFG}(\mu, \sigma) + I_{sub}^{DFG}(\mu, \sigma) + I_{dyn}^{DFG}(\mu, \sigma), \tag{5}$$

where, μ, σ are the mean and standard deviation of each of the current distributions and are different for different components. The resource constraints in Eqn. (3) 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. and 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 Eqn. (4) ensures that the probability density function $\left(D_{CP}^{DFG}\left(\mu_D^{DFG}, \sigma_D^{DFG}\right)\right)$ of the critical path (CP) delay is within the specified limit dictated by the probability density function $\left(D_C\left(\mu_D^C, \sigma_D^C\right)\right)$ of the delay constraint (C).

To solve the optimization problem presented in Eqn. (2), (3), and (4) in the framework of high-level synthesis we present a Simulated Annealing based algorithm. Even though algorithms for optimization are plentiful in the mathematics literature we chose to follow the Simulated Annealing approach as our number of parameters is reasonably large and faster convergence can be provided by this kind of algorithms [36], [37].

VI. Statistical Low-Power Behavioral Synthesis for Fast and Accurate Design Space Exploration

We present the overall framework for statistical low-power behavioral synthesis in Fig. 3. The synthesis framework assumes behavioral VHDL as input and generates a statistical power and statistical delay optimal RTL description accounting for process variations. As can be seen from the figure, the entire high-level synthesis framework is divided into four main modules or engines as follows:

- Input generation engine
- Datapath and control generation engine
- Characterization engine
- Process variation engine
- Power-Delay estimation engine
- Output generation engine

A. Input generation engine

The "input generation engine" accepts the input HDL, does compilation and transformation, 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. The DFG does not support hierarchical entities and conditional statements are handled using comparison operations. Also, each vertex has attributes that specify the operation type. At this step technology independent optimizations can be performed.

B. Datapath and control generation engine

The datapath and control generation engine is the principal unit of the process variation aware power minimization synthesis 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_{ox}, V_{th} and V_{DD}.

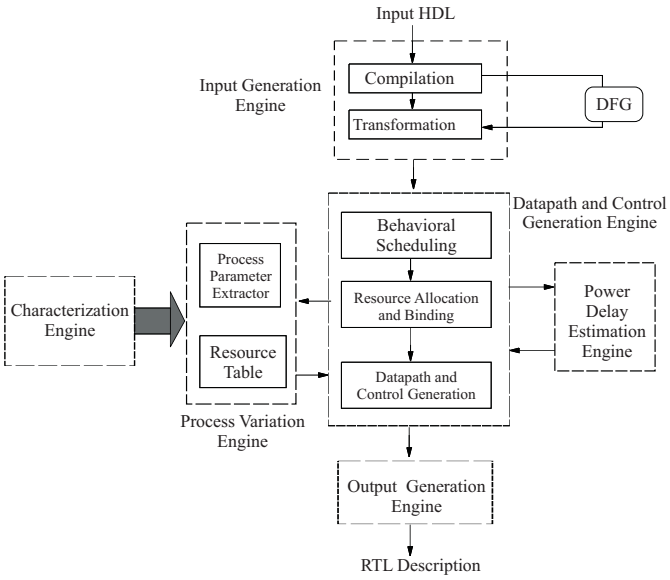


Fig. 3. The proposed statistical behavioral synthesis framework for process variation aware design space exploration.

C. Characterization engine

The "characterization engine" forms a vital part of the process variation aware 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 power and fluctuation 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_{ox}, 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 (I_{gate}, I_{sub} and I_{dyn}) and propagation delay (T_{PD}) probability density functions, each set corresponding to a particular design corner of the input cube.

D. Process variation engine

The "process variation engine" consists of a process parameter extractor which is 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. The datapath and control generation engine is the principal unit of the process variation aware power and fluctuation minimization synthesis flow.

E. Power-Delay estimation engine

The power(current)-delay estimation engines calculate the probability density functions of different current components

and delay. It works in co-ordination with the "characterization engine", the "datapath-control generation engine", etc. and estimates the probability density functions for a given data flow graph (DFG).

F. 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 logic synthesis.

VII. 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 delay as shown in Fig. 4. Initially a 2-input NAND gate was designed and tested for functional correctness using a nano-CMOS technology.

A 2-input NAND gate was designed and tested using Cadence tools for functional correctness at a $45nm$ effective channel length (L). We chose to use the Berkeley Predictive Technology Model (BPTM) as it is widely used [38]. The BSIM4 deck generated through BPTM represent a hypothetical $45nm$ CMOS process. The base (or nominal) values for design corner (1) is as follows: $T_{ox} = 1.4nm$, $V_{th} = 0.22V$ for NMOS, $V_{th} = -0.22V$ for PMOS, W:L = 4:1 for NMOS, W:L = 8:1 for PMOS, and $V_{DD} = 0.7V$.

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 process and design variations are assumed to be Gaussian in nature. This is demonstrated in Fig. 5. While state dependent data are obtained at the logic level and 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 leakage, dynamic and subthreshold currents for the total unit will be normally distributed even though the same currents are lognormally distributed for each individual logic gates.

Based on the above discussion, we can model the currents and the delay for the functional units by utilizing the characterized data for the 2-input NAND gate. The total current in the functional unit can be defined as the sum of currents in the individual NAND gates comprising the unit. Assuming that the distributions for each gate are statistically independent of each other, the mean and variance of the currents can be derived as:

$$\begin{aligned} \mu_{FU} &= N\ \mu_{NAND}, \\ \sigma_{FU} &= \sqrt{N}\ \sigma_{NAND}, \end{aligned} \tag{6}$$

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. The approach presented here can be modified to account for such cases, but for simplicity it is not included here.

From the above equations the mean and the variance of I_{gate}, I_{sub} and I_{dyn} for each of the functional units was calculated. The calculation of the mean and variance for the delay T_{PD} also was performed in a similar manner. The use of universal NAND gates simplifies the construction of the cell (datapath component) library containing functional units like, adder, subtractor, multiplier, etc. The use of other types of logic gates to build datapath component library can be done using the above statistical expressions provided 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 to be used in the subsequent optimization procedure. Characterization data for some sample design corners is shown in Fig. 6.

A. Accounting for Process Variation

In this section we describe the methodology via which the statistical information regarding process and design parameter variability is translated into statistical information regarding power dissipation and delay (performance), as shown schematically in Fig. 5.

The SPICE characterization engine considers the combination of the dual values of the three input parameters (T_{ox}, V_{th} and V_{DD}) as eight corners of a design cube. 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 (T_{PD}) probability density functions, 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) while 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. 7.

B. Analysis of Effects of Scaling on Individual Power Components

Characterization data for various corners are shown in Fig. 6. For the analysis, we consider the 8 corners of the cube as nominal corners. Corner 1 is considered as the baseline corner having values of V_{DD}, T_{ox}, and V_{th} as the standard values for a $45nm$ nano-CMOS technology. The values in the other nominal corners are varied with respect to this corner. Then these nominal corners are processed through spice characterization, and the outputs obtained are the values of I_{gate}, I_{sub}, I_{dyn}, and T_{pd} respectively, which are treated as the eight corners of the output cube.

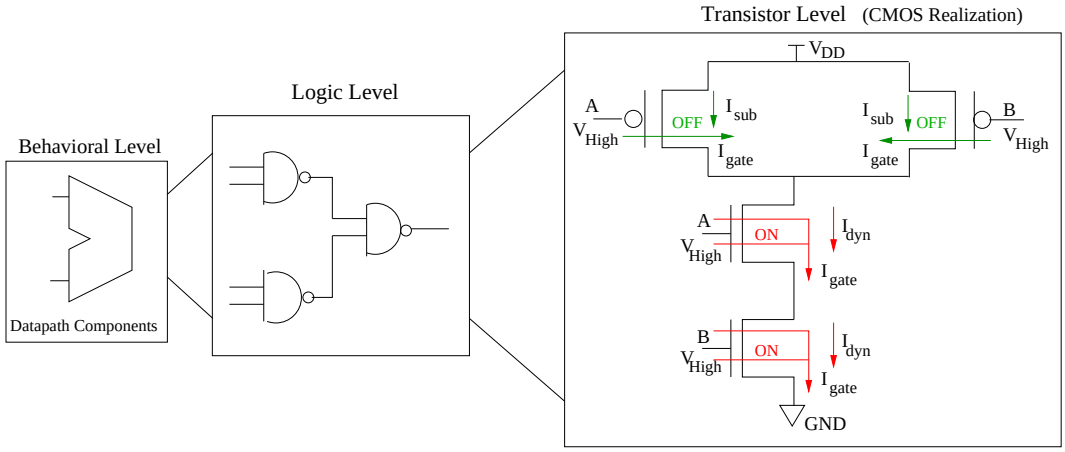


Fig. 4. Three levels of abstraction in which datapath components are realized using 2-input NAND gates. The transistor level diagram shows worst case tunneling current paths in the constituent NAND gates. The worst-case occurs when both inputs are high i.e. $A = B = V_{High}$. In this case, the path 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 6 different components as shown above. If the four possible states (00, 01, 10 and 11) have gate-oxide tunneling current ($I_{ox00}, I_{ox01}, I_{ox10}, I_{ox11}$), respectively, and assuming that all four states are equiprobable the average gate-oxide tunneling current of a 2-input NAND gate is $I_{gate_{NAND}} = \left(\frac{I_{ox00}+I_{ox01}+I_{ox10}+I_{ox11}}{4}\right)$. 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*. Similarly, for a 2-input NAND, subthreshold leakage is $I_{sub_{NAND}} = \sum_{MOS_{OFF_i}} I_{sub_i}$ and dynamic current is $I_{dyn_{NAND}} = \sum_{MOS_{ON_i}} I_{dyn_i}$.

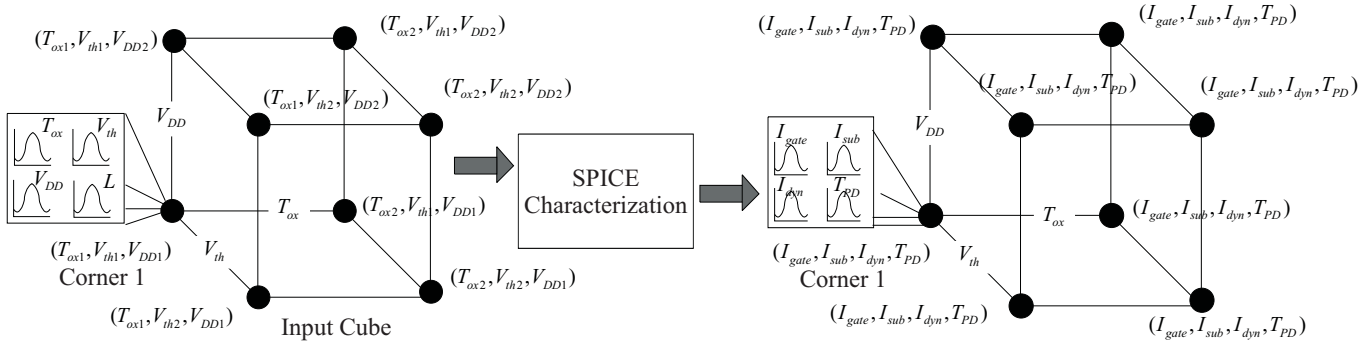
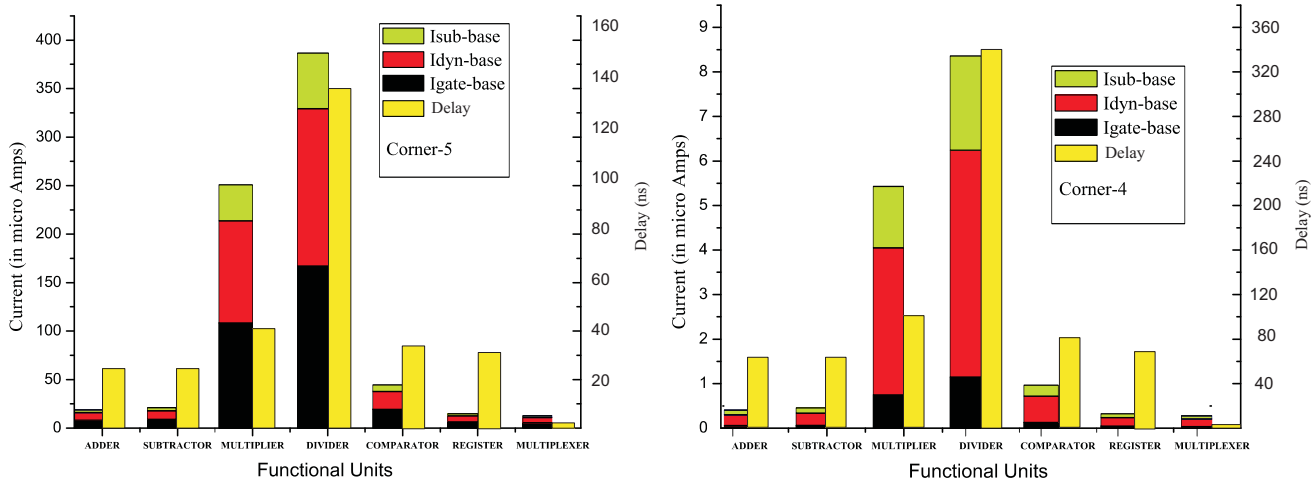
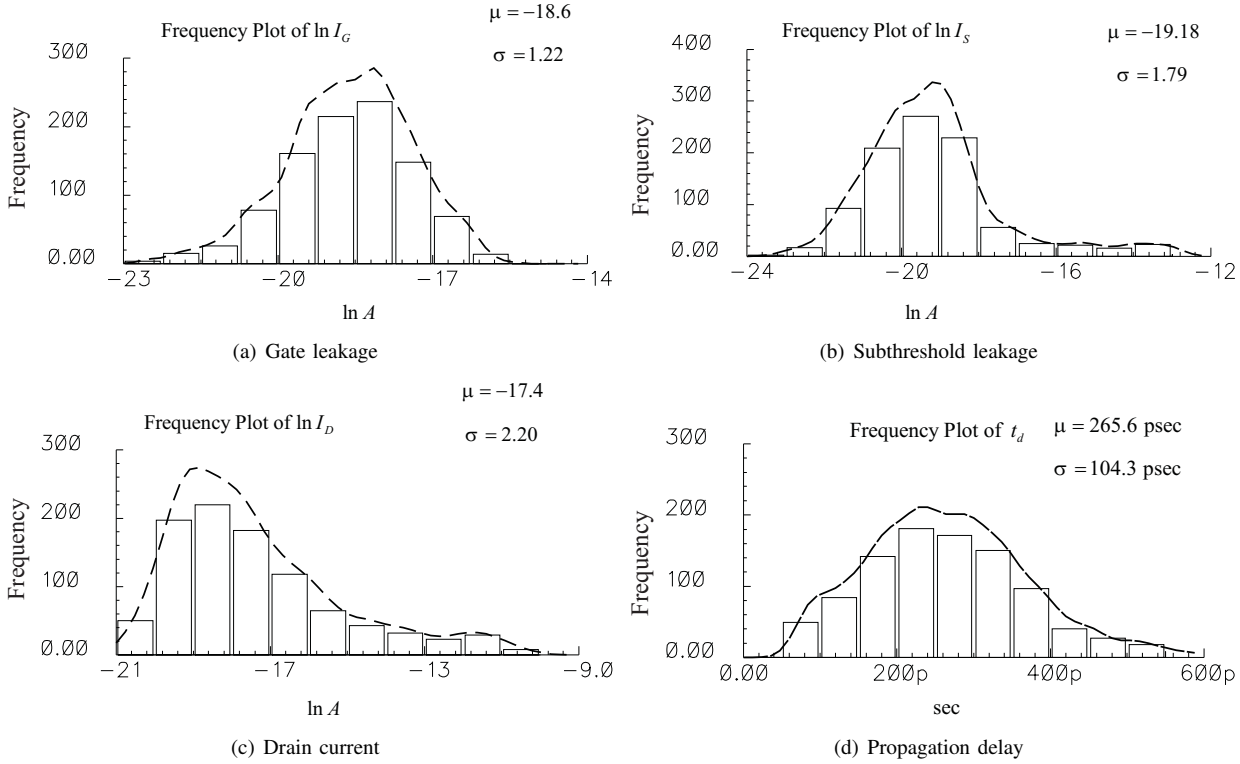


Fig. 5. Monte Carlo simulation methodology to account physical process variations and power supply variation in the power and performance of circuits. Variations are being modeled as Gaussian density functions and variability of different current (power) component is studied which will be useful for statistical process current (power) optimization during high-level synthesis.



(a) Design corner 5: $T_{ox} = 1.4nm$, $V_T = 0.22V$, and $V_{DD} = 0.9V$ (b) Design corner 4: $T_{ox} = 1.7nm$, $V_T = 0.25V$, and $V_{DD} = 0.7V$

Fig. 6. Nominal results showing individual components of power consumption for different output corners. It may be noted that the total current values are reduced and the proportions of different components in the total current have changed. Only two corners are shown for brevity. In corner 5 vs. corner 4, all parameters have been scaled i.e. T_{ox} and V_{th} are increased and V_{DD} is decreased.



(a) Gate leakage (b) Subthreshold leakage

(c) Drain current (d) Propagation delay

Fig. 7. Effects of statistical process variation on gate-oxide leakage, subthreshold leakage, dynamic current, and propagation delay in a 2-input NAND gate. It is observed that variability of gate-oxide leakage, subthreshold leakage, dynamic current is log-normal in nature and variability of propagation delay is normal in nature.

For the purposes of this analysis, we are considering the case of a divider only. But the trend is the same for other datapath components as well. In total we had 7 cases. These are nominal results without considering statistical distributions. In the next section we will consider the results with the 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.

1) Only T_{ox} scaling: This case may arise when T_{ox1} changes to T_{ox2}, e.g. (T_{ox1}, V_{th1}, V_{DD1}) versus (T_{ox2},V_{th1},V_{DD1})). In this case we observe that all power components are reduced with an overall reduction of 91.6% achieved. As expected, the increase in oxide thickness results in a 65.8% delay penalty.

2) Only V_{th} scaling: Scaling V_{th} only ((T_{ox1},V_{th1},V_{DD1}) versus (T_{ox1},V_{th2},V_{DD1})). In this case we observe that total power dissipation decreases by 46.1% while the delay penalty is only 17%.

3) Only V_{DD} scaling: Scaling V_{DD} only ((T_{ox1},V_{th1},V_{DD1}) versus (T_{ox1},V_{th1},V_{DD2})). In this case we observe that total power dissipation decreases by 57.8% with a modest 21.4% delay penalty.

4) Simultaneous T_{ox} and V_{th} Scaling: ((T_{ox1},V_{th1},V_{DD1}) versus (T_{ox2},V_{th2},V_{DD1})). In this case the combined effect of T_{ox} and V_{th} increase results in 94.8% reduction in power but a very significant 100% delay penalty. This is due to the inverse relation of the delay to *both* T_{ox} and V_{th} [39]:

$$t_d \sim \frac{1}{T_{ox}(V_{DD} - V_{th})^2}. \tag{7}$$

5) Simultaneous T_{ox} and V_{DD} Scaling: ((T_{ox1},V_{th1},V_{DD1}) versus (T_{ox2},V_{th1},V_{DD2})). As anticipated from Eqn. 7, the delay penalty is again significant (101.4%) with similar reduction in power as in the previous case (93.4%).

6) Simultaneous V_{th} and V_{DD} Scaling: ((T_{ox1},V_{th1},V_{DD1}) versus (T_{ox1},V_{th2},V_{DD2})). In this case we observe that since both V_{th} and V_{DD} have been scaled, by Eqn. 7 we anticipate a more pronounced delay: 42.1%. 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}): 71.1%.

7) Simultaneous T_{ox} and V_{th} and V_{DD} Scaling: We note that the effect of scaling all the parameters cannot be easily or accurately obtained from the responses (output results) of varying a single or a few parameter(s).

((T_{ox1},V_{th1},V_{DD1}) versus (T_{ox2},V_{th2},V_{DD2})): When all three parameters are scaled simultaneously, we obtain a power reduction of 97.8% and a worst delay penalty of 142.9% when averaged over all units. These performance results, indicated in Fig. 6, are not easily anticipated from simple analysis of the prior 6 cases (corners 2 thru 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 I. This case is represented in Fig. 6.

From the above discussions it is evident that we can not simply apply case 7 to obtain a globally power and perfor-

TABLE I

PERCENTAGE (%) REDUCTION IN CURRENT DISSIPATION AND INCREASE IN PROPAGATION DELAY

Current or Delay	**Parameters varied or scaled**						
	T_{ox} case-(1)	V_{th} case-(2)	V_{DD} case-(3)	T_{ox} + V_{th} case-(4)	T_{ox} + V_{DD} case-(5)	V_{th} + V_{DD} case-(6)	T_{ox} + V_{th} + V_{DD} case-(7)
Gate-oxide Leakage	96.2	11.3	69.2	97.7	98.8	72.7	99.3
Subthreshold Leakage	94.3	10.5	63.4	89.8	97.9	59.6	96.3
Dynamic	88.3	52.6	70.1	94.3	93.4	73.5	96.8
Total Current	91.6	46.1	57.8	94.8	96.4	71.1	97.8
Critical Path Delay	65.8	17.0	21.4	100.0	101.4	42.1	142.9

mance 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.

VIII. OUR PROPOSED OPTIMIZATION APPROACH

In this section we present an algorithm that performs simultaneous scheduling, binding, and allocation during the statistical behavioral synthesis flow presented in Section VI. The simulated annealing based algorithm performs the minimization of the cost function presented in Section V 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. As the power consumption can be represented in terms of the corresponding currents we present our methodology as a current based power. We assume that the datapath is represented as a Data Flow Graph (DFG) derived from a hardware description language (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 while 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 as the number of parameters involved in optimization is large and simulated annealing approach can facilitate faster convergence in reasonable time [36], [37] compared to more sophisticated approaches like integer linear programming (ILP). 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 crystalizes. The atoms of this material have higher energies at very high temperatures. This gives the atoms a great deal of freedom in their ability to reconstruct themselves. As the temperature decreases the energy of the atoms decrease. Analogous to the annealing process, the mobility of nodes in a DFG (data flow graph representing 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 data flow graph (UDFG), and the resource and/or time constraints that include a number of different types of resources from different design corners. 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_{ox}, 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, so 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 ready state before running the next clock cycle or control step.

In the outer loop during each iteration the number of resources is decreased, which restricts the mobility of the nodes. The algorithm attempts to find an RTL that has minimum leakage 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 function (minimum power fluctuation and total power). In generating a neighborhood solution we randomly select

Algorithm 1 Simulated Annealing based algorithm for minimizing the cost function

Perform ASAP and ALAP scheduling.
while {There exists a schedule with available resources} **do**
  $i =$ Number of iterations.
  Perform resource constrained ASAP.
  Perform resource constrained ALAP.
  Initial Solution $\leftarrow$ ASAP Schedule.
  $S \leftarrow$ Allocate_Bind().
  Initial Cost $\leftarrow$ Power_Cost(S).
  while $\{(i > 0)\}$ **do**
    Generate random transition from S to S^* with other assignments while satisfying constraints.
    Δ-cost $\leftarrow$ Cost(S) $-$ Cost(S^*)
    if $\{\Delta\text{-cost} > 0\}$ **then**
      return $S \leftarrow S^*$.
    end if
    $i \leftarrow i - 1$.
  end while
  Decrement available resources.
end while
Determine variability current and delay of the circuit.
return S.

a node and check if a better resource (a resource with less power) can be assigned in all possible clock cycles and 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 which handle both multicycling and single cycle datapath. The *algorithm prioritizes* the design corners based on the total current and delay. It ensures that all non-critical path resources are assigned less power consuming resources. After several trials we found that 50 iterations provide a good trade-off between the algorithm performance and the cost function reduction.

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 probability density functions (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 α and β, respectively. Depending on whether the objective of the optimization is performance or yield enhancement, α or β, respectively are assigned higher weights. 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 product of current-cost and delay-cost.

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 leakage, subthreshold leakage and dynamic current at each stage in the datapath) in each cycle. The mean and standard deviation of each component of

Algorithm 2 Algorithm to generate random transition for mobile vertices in DFG.

Select a random vertex $v_i \in V$.
for all {Possible cycles c in the mobility range} **do**
  if {If low-power resource is available for c} **then**
    Schedule v_i in step c
    Adjust resource allocation Table accordingly.
    TotalDelay=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=v_i, break; /*var–node executed*/
        end if
      end for
      Increment TotalDelay by delay of resource allocated in this iteration.
      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 if
end for

current in clock cycle c are given by:

$$\mu^c_{I_{component}} = \frac{1}{N_{FU}} \sum_{v=1}^{N_{FU}} \mu_I^{FU_{k,i,v}}, \tag{8}$$

$$\sigma^c_{I_{component}} = \sqrt{\frac{1}{N_{FU}} \sum_{v=1}^{N_{FU}} {\sigma_I^{FU_{k,i,v}}}^2}. \tag{9}$$

Here 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 $FU_{k,i}$ may be an 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 probability density functions. The mean and standard deviation of I^c_{total} is then calculated as follows:

$$\mu^c_{total} = \frac{1}{3}\left(\mu^c_{gate} + \mu^c_{sub} + \mu^c_{dyn}\right), \tag{10}$$

Algorithm 3 Algorithm for Cost Calculation.

1: $I^c_{gate}\left(\mu^c_{gate}, \sigma^c_{gate}\right) = \sum_{\forall v \in c} I^{FU_{k,i,v}}_{gate}$
2: $I^c_{sub}\left(\mu^c_{sub}, \sigma^c_{sub}\right) = \sum_{\forall v \in c} I^{FU_{k,i,v}}_{sub}$
3: $I^c_{dyn}\left(\mu^c_{dyn}, \sigma^c_{dyn}\right) = \sum_{\forall v \in c} I^{FU_{k,i,v}}_{dyn}$
4: $I^c_{total}\left(\mu^c_{total}, \sigma^c_{total}\right) = I^c_{gate}\left(\mu^c_{gate}, \sigma^c_{gate}\right) + I^c_{sub}\left(\mu^c_{sub}, \sigma^c_{sub}\right) + I^c_{dyn}\left(\mu^c_{dyn}, \sigma^c_{dyn}\right)$
5: $I^{DFG}_{total}\left(\mu^{DFG}_I, \sigma^{DFG}_I\right) = \sum_{c=1}^{N_{cc}} I^c_{total}\left(\mu^c_{total}, \sigma^c_{total}\right)$
6: $Cost_I = \alpha * \mu^{DFG}_I + \beta * \sigma^{DFG}_I$
7: Calculate $Cost^{FU}_D = \alpha * \mu^{FU}_D + \beta * \sigma^{FU}_D$, for each FU active in c.
8: **for all** {Control step c} **do**
9: **if** {Single Cycle Datapath } **then**
10: $Cost^c_D =$ maximum $\left(Cost^{FU}_D\right)$
11: **else**
12: $Cost^c_D =$ minimum $\left(Cost^{FU}_D\right)$
13: **end if**
14: **end for**
15: $Cost_D = \sum_{c=1}^{N_{cc}} Cost^c_D$.
16: Cost $= Cost_I * Cost_D$.
17: **return** Cost.

$$\sigma^c_{total} = \sqrt{\frac{1}{3}\left({\mu^c_{gate}}^2 + {\mu^c_{sub}}^2 + {\mu^c_{dyn}}^2\right)}. \quad (11)$$

The total current dissipation of the overall datapath circuit under synthesis that is being specified by the DFG for this assignment is then given by summing over all cycles:

$$\mu^{DFG}_I = \frac{1}{N_{cc}} \sum_{c=1}^{N_{cc}} \mu^c_{total}, \quad (12)$$

$$\sigma^{DFG}_I = \sqrt{\frac{1}{N_{cc}} \sum_{c=1}^{N_{cc}} {\sigma^c_{total}}^2}, \quad (13)$$

where N_{cc} is the total number of cycles in the datapath.

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. Similarly, the clock cycle width for the multicycle datapath is fixed as the fastest functional unit delay (considering mean value). However, to quantify the variability in delay of the overall circuit, the mean and standard deviation similar to the current needs to be considered. For clock cycle c the variability in delay can be quantified as follows:

$$\mu^c_D = \frac{1}{N_{FU}} \sum_{v=1}^{N_{FU}} \mu^{FU_{k,i,v}}_D, \quad (14)$$

$$\sigma^c_D = \sqrt{\frac{1}{N_{FU}} \sum_{v=1}^{N_{FU}} {\sigma^{FU_{k,i,v}}_D}^2}. \quad (15)$$

Here 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 delay in the datapath circuit represents the delay for the critical path. This is given by the following expressions for N_{cc} number of cycles:

$$\mu^{DFG}_D = \frac{1}{N_{cc}} \sum_{c=1}^{N_{cc}} \mu^c_D, \quad (16)$$

$$\sigma^{DFG}_D = \sqrt{\frac{1}{N_{cc}} \sum_{c=1}^{N_{cc}} {\sigma^c_D}^2}. \quad (17)$$

The above presented models use the statistical process variation datapath component library containing base value, mean and S. D. 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 paper.

IX. Experimental Results

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 Spectre circuit simulator. On the other hand, the simulated annealing algorithm is implemented in C and integrated in the behavioral synthesis tool borrowed from [40]. The algorithms were exhaustively tested with several behavioral level benchmark circuits for several constraints. We present the experimental results in this section for a selected set of benchmarks and constraints.

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 oxide thicknesses available to the behavioral scheduling-binding algorithms. The sets of resource constraints were chosen so as to cover functional units consisting of different oxide thickness. They are representatives of various forms of the corresponding RTL representation.

We applied our optimization technique to several standard high-level synthesis benchmark borrowed from [40]. We consider design corner 1 (nominal T_{ox}, V_{th} and V_{DD}) as the baseline. The percentage reduction is calculated as:

$$\Delta I = \left(\frac{I_{\text{Baseline}} - I_{\text{Final}}}{I_{\text{Baseline}}}\right) * 100\%. \quad (18)$$

This formula uses the mean (μ) of the various components of current as well as the total current for computation. The

percentage reductions for I_{gate}, I_{dyn}, I_{sub}, and total current I_{total} are calculated for the overall datapath circuit. The experimental results take into account the current and propagation delay of functional units and storage units present in the datapath circuit. For brevity we present average percentage reduction data; the percentage reductions for each set were then averaged. It is observed that typical simulation time for a benchmark circuit was in the range of 20 to 30 mins, which proves that our algorithm converges to solutions is a very reasonable time. We selected $\alpha = 1$ and $\beta = 1$ as the values of the cost function's weighting factors, while calculating the cost.

The experimental results are shown for selected benchmarks in Fig. 8. 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 IIR benchmark obtains the best results while the HAL benchmark shows the least improvement, assuming no delay penalty. This is consistent with the fact that the IIR is more complex (in terms of number of adders and multipliers) than the HAL.

To the best of our knowledge, we did not find behavioral (high-level, or architectural) synthesis research works having the same scope as the work presented in this paper i.e. accounting for gate-oxide leakage, subthreshold leakage, dynamic power dissipation together, process variation, and statistical optimization. Hence, a fair comparison of the presented results is not possible. However, individual results in gate-oxide leakage, subthreshold leakage, and dynamic power, are comparable and considerably better than the related prior research works. However, in view of the low power behavioral synthesis works, we provide a broader comparative perspective in Table II. 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 [41] uses a different set of benchmark circuits than the rest of the works in Table II, so we provide the overall average data. The work presented in [20] is area constrained so we did not get the delay penalty data. The results produced show that dual-T_{ox} approach presented in this paper results in significant reduction in gate leakage with reasonable time penalty. This has outperformed the multi-V_{dd} approach for dynamic power reduction and multi-V_{Th} for subthreshold leakage reduction.

X. Summary and Conclusions

In this work a novel process variation aware power characterization and optimization methodology was presented. The methodology is developed in the framework of behavioral synthesis for nano-CMOS circuits. An extensive functional unit model library was created by considering the individual and combined variations of T_{ox}, V_{th}, and V_{DD} via transistor level Monte Carlo 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 I_{gate}, I_{dyn}, and I_{sub} as well as propagation delay of the functional units. The effect of scaling of three parameters, T_{ox}, V_{th} ,and V_{DD} on various power (current) components was studied. It was observed that simultaneous, independent scaling of all three 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 current (power) along with total current can be achieved using the proposed methodology. Thus, the proposed algorithms, approach, and methodology can advance the state-of-the-art research in high-level synthesis and can make them suitable to handle the challenges of complex nanoscale CMOS digital circuits.

References

[1] K. Roy, S. Mukhopadhyay, and H. M. Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, February 2003.

[2] Saraju P. Mohanty and Elias Kougianos, "Modeling and Reduction of Gate Leakage during Behavioural Synthesis of NanoCMOS Circuits," in *Proceedings of International Conference on VLSI Design*, Jan 2006.

[3] J. A. Butts and 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.

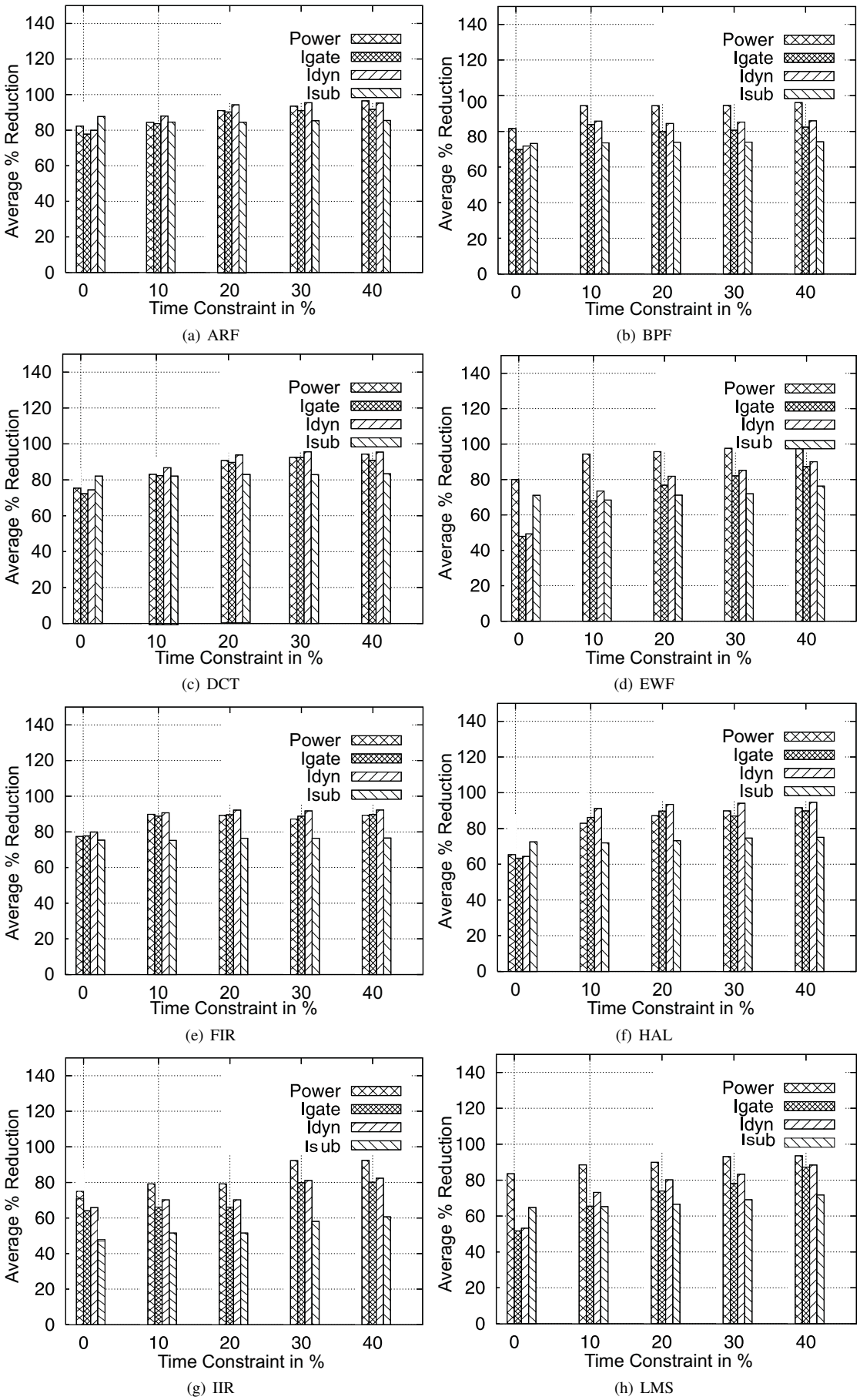
[4] J. G. Hansen, "Design of CMOS Cell Libraries for Minimal Leakage Currents," M.S. thesis, Dept. of Informatics and Mathematical Modelling, Computer Science and Engineering Technical University of Denmark, Fall, 2004.

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

[6] A. J. Bhavnagarwala, B. L. Austin, K. A. Bowman, and J. D. Meindl, "A Minimum Total Power Methodology for Projecting Limits of CMOS GSI," *IEEE Transactions on VLSI Systems*, vol. 8, no. 3, pp. 235–251, June 2000.

[7] K. A. Bowman, L. Wang, X. Tang, and J. D. Meindl, "A Circuit-Level Perspective of the Optimum Gate Oxide Thickness," *IEEE Transactions on Electron Devices*, vol. 48, no. 8, pp. 1800–1810, August 2001.

[8] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Brodersen, "Optimizing Power using Transformations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 1, pp. 12–31, Jan 1995.



(a) ARF (b) BPF

(c) DCT (d) EWF

(e) FIR (f) HAL

(g) IIR (h) LMS

Fig. 8. Experimental results showing the percentage reduction in various leakage and dynamic power for selected standard benchmarks.

TABLE II

A BROAD COMPARATIVE PERSPECTIVE WITH EXISTING LOW POWER HIGH-LEVEL SYNTHESIS TECHNIQUES

Works →	Shiue 2000		Manzak 2002		Khouri 2002		Gopalakrishnan 2003		This Work				
Power → Method →	Dynamic Power (Multi-V_{dd})				Subthreshold Leakage (Multi-V_{th})				Gate/Subthreshold Leakage and Dynamic (Judicious use of dual T_{ox}, V_{th}, or V_{dd})				
Benchmarks	ΔP	ΔT_{pd}	ΔP	ΔT_{pd}	ΔP	ΔT_{pd}	ΔP	ΔT_{pd}	ΔI_{gate}	ΔI_{sub}	ΔI_{dyn}	ΔI_{total}	ΔT_{pd}
ARF	56.3	50.0	46.1	50.0			8.4	–	86.8	83.8	88	87.6	20
DCT			34.1	50.0	58.0	–			84.2	81.0	88.0	88.0	20
EWF	56.3	50.0	35.7	50.0			19.7	–	68.9	76.5	75.4	93.6	20
FIR			41.3	50.0			21.6	–	85.6	76.0	88.0	86.0	20

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

[10] S. P. Mohanty and N. Ranganathan, "Energy Efficient Datapath Scheduling using Multiple Voltages and Dynamic Clocking," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 10, no. 2, pp. 330–353, April 2005.

[11] V. Mukherjee, S. P. Mohanty, and 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.

[12] S. P. Mohanty, V. Mukherjee, and R. Velagapudi, "Analytical Modeling and Reduction of Direct Tunneling Current during Behavioral Synthesis of Nanometer CMOS Circuits," in *Proceedings of the 14th ACM/IEEE International Workshop on Logic and Synthesis (IWLS)*, 2005, pp. 249–256.

[13] D. Lee, D. Blaauw, and D. Sylvester, "Gate Oxide Leakage Current Analysis and Reduction for VLSI Circuits," *IEEE Transactions on VLSI Systems*, vol. 12, no. 2, pp. 155–166, February 2004.

[14] A. K. Sultania, D. Sylvester, and 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.

[15] N. Sirisantana and K. Roy, "Low-power Design using Multiple Channel Lengths and Oxide Thicknesses," *IEEE Design and Test of Computers*, vol. 21, no. 1, pp. 56–63, Jan-Feb 2004.

[16] S. Mukhopadhay and K. Roy, "Modeling and estimation of total leakage current in nano-scaled CMOS devices considering the effect of parameter variation," in *Proceedings of the IEEE International Symposium on Low Power Design*, 2003, pp. 172–175.

[17] P. Pant, R. K. Roy, and A. Chattejee, "Dual-Threshold Voltage Assignment with Transistor Sizing for Low Power CMOS Circuits," *IEEE Transactions on VLSI Systems*, vol. 9, no. 2, pp. 390–394, April 2001.

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

[19] K. S. Khouri and N. K. Jha, "Leakage power analysis and reduction during behavioral synthesis," in *Proceedings of International Conference on Computer Design*, 2000, pp. 561–564.

[20] C. Gopalakrishnan and 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.

[21] C. Gopalakrishnan and S. Katkoori, "Resource allocation and binding approach for low leakage power," in *Proceedings of 16th International Conference on VLSI Design*, 2003, pp. 297–302.

[22] Michael Liu, Wei-Shen Wang, and Michael 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*, Aug 2004, pp. 2–7.

[23] Behnam Ameliford, Farzan Fallah, and Massoud 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*, March 2006, pp. 1–6.

[24] Akhilesh Kumar and Mohab Anis, "Dual-Vt Design of FPGAs for Subthreshold Leakage Tolerance," in *Proceedings of International Symposium on Quality Electronic Design*, March 2006.

[25] Liqiong Wei and et.al., "Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications," *IEEE Transactions on VLSI Systems*, vol. 7, no. 1, pp. 16–24, March 1999.

[26] R. S. Martin and J. P. Knight, "Using Spice and Behavioral Synthesis Tools to Optimize ASICs' Peak Power Consmpution," in *Proceedings of the 38th Midwest Symposium on Circuits and Systems*, 1996, pp. 1209–1212.

[27] M. Johnson and K. Roy, "Datapath Scheduling with Multiple Supply Voltages and Level Converters," *ACM Transactions on Design Automation of Electronic Systems*, vol. 2, no. 3, pp. 227–248, July 1997.

[28] W. T. Shiue and C. Chakrabarti, "Low-Power Scheduling with Resources Operating at Multiple Voltages," *IEEE Transactions on Circuits and Systems-II : Analog and Digital Signal Processing*, vol. 47, no. 6, pp. 536–543, June 2000.

[29] J. M. Chang and M. Pedram, "Energy Minimization using Multiple Supply Voltages," *IEEE Transactions on VLSI Systems*, vol. 5, no. 4, pp. 436–443, Dec 1997.

[30] S. H. Kulkarni and D. Sylvester, "High Performance level Conversion for Dual VDD Design," *IEEE Transactions on VLSI Systems*, vol. 12, no. 9, pp. 926–936, Sept 2004.

[31] Ching Ping-Yuan and Yu Chien-Cheng, "A Voltage Level Converter Circuit Design with Low Power Consumption," in *Proceedings of the 6th International Conference on ASIC*, Oct 2005, pp. 358–359.

[32] A. E. Gattiker W. Haensch B. L. Ji S. R. Nassif E. J. Nowak D. J. Pearson K. Bernstein, D. J. Frank and N. J. Rohrer, "High-performance CMOS variability in the 65-nm regime and beyond," *IBM Journal of Research and Development*, vol. 50, no. 4/5, pp. 433–449, July-September 2006.

[33] K. Singhal and V. Visvanathan, "Statistical device models from worst case files and electrical test data," *IEEE Transanction on Semiconductor Manufacturing*, vol. 12, no. 4, pp. 470–484, November 1999.

[34] P. G. Drennan and C. C. McAndrew, "Understanding MOSFET mismatch for analog design," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, pp. 450–456, March 2003.

[35] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, Inc., 1994.

[36] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[37] V. Cerny, "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[38] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and 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.

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

[40] S. P. Mohanty and N. Ranganathan, "A Framework for Energy and Transient Power Reduction during Behavioral Synthesis," *IEEE Transactions on VLSI Systems*, vol. 12, no. 6, pp. 562–572, June 2004.

[41] K. S. Khouri and N. K. Jha, "Leakage power analysis and reduction during behavioral synthesis," *IEEE Transactions on VLSI Systems*, vol. 10, no. 6, pp. 876–885, December 2002.