

Simultaneous Power Fluctuation and Average Power Minimization during Nano-CMOS Behavioral Synthesis

Saraju P. Mohanty
Computer Science and Engineering
University of North Texas
P. O. Box 311366, Denton, TX 76203.
Email-ID: smohanty@cse.unt.edu

Elias Kougianos
Electrical Engineering Technology
University of North Texas
P. O. Box 310679, Denton, TX 76203.
Email-ID: eliask@unt.edu

Abstract

We present minimization methodologies and an algorithm for simultaneous scheduling, binding, and allocation for the reduction of total power and power fluctuation during behavioral synthesis. We consider resources of dual gate oxide thicknesses, dual threshold voltage, and dual power supply. Statistical variations in these parameters are explicitly taken into account by using Monte Carlo simulations to characterize a datapath component library which is then used during behavioral synthesis. The formulated multi-objective cost function is optimized for various resource and time constraints. We present results on several standard benchmarks where we observed significant reduction in total power (as high as 75% without time penalty) and elimination of total power fluctuation (as high as 76% without time penalty). To the best of the authors' knowledge, this is the first-ever behavioral synthesis work addressing fluctuation in total power consumption accounting for gate and subthreshold leakage and dynamic power.

1 Introduction and Contributions

The market demand for portability, performance and higher integration density of digital devices has made the scaling of CMOS devices inevitable. The major sources of power dissipation in a nanoscale CMOS circuit can be summarized as follows [1, 2, 9, 14]:

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

The power consumption as well as the fluctuation in power consumption of a circuit affect its operational attributes. Increase in power consumption is detrimental to battery life and affects the reliability of the device [4]. Fluctuations in power consumption decrease battery life due to

reduced efficiency in electrochemical conversion [6]. Power fluctuation also leads to larger power supply noise, due to self inductance and can also introduce significant noise in signal lines due to mutual inductance and capacitance (cross-talk.) High current peaks in short time spans can cause high heat dissipation in a localized area of the die which may lead to failures.

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 in design and synthesis frameworks. Designing for the worst case scenario may cause severe compromises on the performance of the device. Hence we consider a process variation aware average power and fluctuation minimization technique.

The principal power (dynamic current I_{dyn}) and leakage components (gate leakage I_{gate} , and subthreshold leakage I_{sub}) predominantly depend on the gate oxide thickness (T_{ox}), threshold voltage (V_{th}) and supply voltage (V_{DD}). Any methodology for power and power fluctuation reduction must focus on the variation of these parameters. Our methodology consistently does so and incorporates the variation in the model for average power and power fluctuation. Our reduction methodology also considers several possible design corners in resource and time constraints while optimizing a multicost objective function. We provide statistically characterized gate and functional unit models which were simulated at transistor level for obtaining the mean and standard deviation (S. D.) of gate leakage, subthreshold leakage, dynamic current and propagation delay with simultaneous variation of all process and design parameters.

The *contributions of this paper* can be summarized as: (a) introduction of an exhaustive statistical process variation aware datapath component library, (b) introduction of a

process variation aware power and fluctuation minimization method, (c) creation of a unique power and fluctuation minimization model for easy integration into behavioral synthesis tasks, and (d) exploration of all design corners of a dual- T_{ox} , dual- V_{th} and dual- V_{DD} technology through Simulated Annealing based optimization algorithm for reduction of all major components of active (I_{dyn}) and leakage (I_{gate} and I_{sub}) currents in power dissipation.

2 Related Research

A number of high-level synthesis works have appeared in the literature addressing average *dynamic power* or energy reduction during datapath synthesis. There are few research works minimizing peak power at behavioral level. In [8] the peak power reduction is achieved through simultaneous assignment and scheduling. In [15] ILP based scheduling and force directed scheduling are used to minimize peak power. In [13] data monitor operations are used for simultaneous reduction of peak power and peak power differential. In [11] a heuristic based scheme is proposed that minimizes peak power, peak power differential and average power. In [12] an ILP based method has been proposed for power fluctuation minimization, however dynamic power consumption has only been considered. The above works only account for dynamic consumption and do not consider both gate and subthreshold leakage and they also do not take process variation into consideration.

Additionally, *leakage power* reduction during behavioral synthesis is gaining attention. The authors in [7] proposed a dual- V_{th} technique for subthreshold leakage analysis and reduction during behavioral synthesis. The use of a multi-threshold CMOS approach for reduction of subthreshold leakage is presented in [5]. In [10] the authors have presented analytical models and a scheduling algorithm for reduction of gate leakage (direct tunneling current.)

At present, low power high-level synthesis works mostly address average dynamic power reduction only, while some of them address subthreshold leakage only, and a few address gate leakage only. In this paper we present a framework for integrated process variation aware power and fluctuation minimization with due consideration to gate as well as subthreshold leakage along with dynamic current.

3 Statistical Power Modeling

In this section we present our methodology as a current based power and fluctuation minimization model. 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 parameters to follow normal distributions.

The problem of minimization of power and fluctuation is decomposed into two minimization problems each having its own cost function: (i) minimization of total power including leakage and dynamic power, and (ii) minimization of total fluctuation in power consumption of datapath.

The minimization of total power (P) including gate leakage, subthreshold leakage, and dynamic power of the datapath circuit has a cost function defined by:

$$\chi_P^{DFG} = \alpha I_{gate}^{DFG}(\mu, \sigma) + \beta I_{sub}^{DFG}(\mu, \sigma) + \gamma I_{dyn}^{DFG}(\mu, \sigma) \quad (2)$$

where μ, σ are the mean and standard deviation of each current distribution and α, β and γ are weight and normalization factors used to tune the objective function to facilitate gate leakage, subthreshold leakage, or dynamic power minimization. The cost function for minimization of fluctuation depends on the total cycle-to-cycle power which can also be presented in terms of currents. The total current in a cycle c is given by:

$$I_{total}^c = I_{gate}^c(\mu, \sigma) + I_{sub}^c(\mu, \sigma) + I_{dyn}^c(\mu, \sigma). \quad (3)$$

This relation can be used to define the total cycle-to-cycle power fluctuation minimization (F) cost function for the overall datapath circuit corresponding the DFG as follows:

$$\chi_F^{DFG} = \sum_{c=1}^{N_{cc}-1} |I_{total}^c(\mu, \sigma) - I_{total}^{c+1}(\mu, \sigma)|, \quad (4)$$

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 (dI/dt) , where the change in time is continuous. However, during behavioral synthesis process the concept of time is manifested in the form of clock cycles and hence (dI/dt) can be estimated as cycle-to-cycle difference as presented above.

The overall minimization problem reduces to the minimization of the two cost functions defined above. The overall multicost objective function can then be expressed as:

$$\downarrow \chi_{(P \cup F)}^{Datapath} = \downarrow (\chi_P^{DFG}) + \downarrow (\chi_F^{DFG}). \quad (5)$$

This expression serves as the cost function for the synthesis framework which needs to be minimized for different constraints such as resource and time constraints using a simulated annealing approach, as discussed in Section 4.

The mean and standard deviation of the average power in clock cycle c are given by:

$$\begin{aligned} \mu_c &= \frac{1}{N_{FU}} \sum_{i=1}^{N_{FU}} \mu_{FU_i}, \\ \sigma_c &= \sqrt{\frac{1}{N_{FU}} \sum_{i=1}^{N_{FU}} \sigma_{FU_i}^2}. \end{aligned} \quad (6)$$

It is assumed that N_{FU} functional units are active during cycle c and FU_i is i -th instance of a functional unit, which may be an adder, subtractor, etc. made of transistors of specific T_{ox} with specific V_{th} and operated at V_{DD} , each having specific probability density functions.

The total power consumption 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:

$$\begin{aligned}\mu_{Power} &= \frac{1}{N_{cc}} \sum_{c=1}^{N_{cc}} \mu_c, \\ \sigma_{Power} &= \sqrt{\frac{1}{N_{cc}} \sum_{c=1}^{N_{cc}} \sigma_c^2},\end{aligned}\quad (7)$$

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

The delay in the datapath circuit is represented in terms of a mean (μ_{pd}) and S. D. (σ_{pd}) of the delay for the critical path given by:

$$\begin{aligned}\mu_{Delay} &= \frac{1}{N_{CP}} \sum_{i=1}^{N_{CP}} \mu_{pd_{FU_i}}, \\ \sigma_{Delay} &= \sqrt{\frac{1}{N_{CP}} \sum_{i=1}^{N_{CP}} \sigma_{pd_{FU_i}}^2}.\end{aligned}\quad (8)$$

N_{CP} is the number of functional units in the critical path of the DFG. N_{CP} is determined from the intermediate (partially) scheduled, bound, and allocated DFG.

4 Our Proposed Optimization Approach

In this section we present an algorithm that performs simultaneous scheduling, binding, and allocation during behavioral synthesis flow shown in Fig. 1. The simulated annealing based algorithm performs the minimization of the multi-objective cost function presented in Eqn. 5 under resource and time constraints. The entire synthesis framework is divided into four main engines as follows: characterization engine, process variation engine, input generation engine, and datapath and control generation engine.

The “characterization engine” is a process variation aware statistical model library generator. It takes a set of statistical inputs of design parameters (mean, μ and standard deviation, σ) and generates a set of statistical outputs (μ and σ) in terms of current and delay. This 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 statistical distribution of these inputs along with the statistical distribution of “ L ” is supplied to the engine. It then processes the input cube and generates a corresponding output cube consisting 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 “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. The power and fluctuation minimized datapath and control generated are represented through an RTL description which is processed by an “output generation engine” (not shown in Fig. 1.)

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. 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 and are made of the same type of transistors as their associated functional units. Voltage level converters are used when a low-voltage functional unit is driving a high-voltage functional unit.

We present a simulated annealing based algorithm in Fig. 2 that minimizes the current-fluctuation cost function in Eqn. 5. 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 and minimum fluctuation in 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 (nominal) resources to all the operations. 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.

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

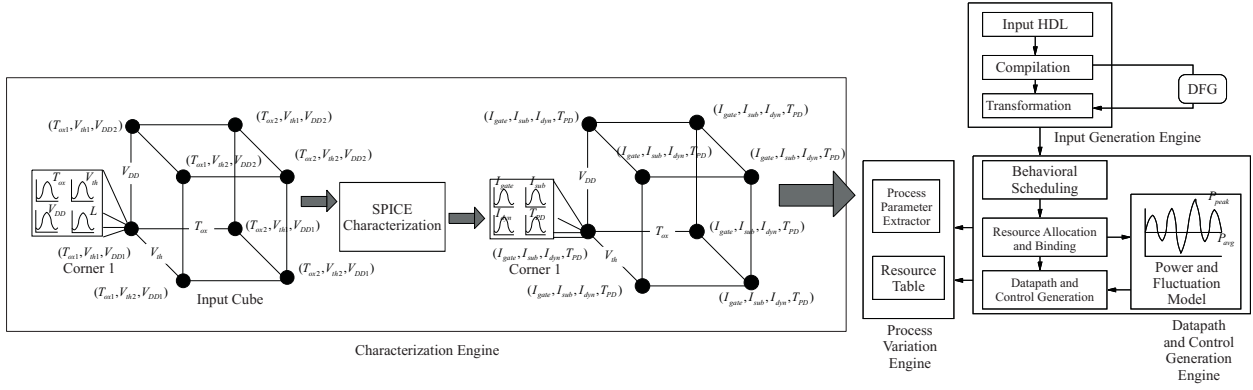


Figure 1. The proposed optimization approach during behavioral synthesis flow including statistical process variation and power and power fluctuation models. Corner 1, consisting of baseline parameters, is explicitly shown.

```

SA_Scheduler_Binder (UDFG, Constraints, Cells)
(01) Perform ASAP and ALAP scheduling.
(02) While there exists a schedule with available resources.
(03)    $i$  = Number of iterations.
(04)   Perform resource constrained ASAP.
(05)   Perform resource constrained ALAP.
(06)   Initial Solution  $\leftarrow$  ASAP Schedule.
(07)    $S \leftarrow$  Allocate_Bind().
(08)   Initial Cost  $\leftarrow$  Power_Cost( $S$ ).
(09)   While ( $i > 0$ )
(10)     Generate random transition from  $S$  to  $S^*$  with
           other assignments while satisfying constraints.
(11)      $\Delta$ -cost  $\leftarrow$  Cost( $S$ ) - Cost( $S^*$ )
(12)     if(  $\Delta$ -cost  $> 0$  ) then  $S \leftarrow S^*$ .
(13)      $i \leftarrow i - 1$ .
(14)   end While.
(15)   Decrement available resources.
(16) end While.
(17) return  $S$ .

```

Figure 2. Simulated Annealing based algorithm for minimizing the cost function

is generated. If this solution has lower cost than the current solution, the neighborhood solution is made the current solution. In generating a neighborhood solution we randomly select a node and check if a better resource (a resource with less power and that can contribute to fluctuation minimization) can be assigned in all possible clock cycles and that it satisfies a time constraint. We have not presented the pseudocode of the algorithm that generates the neighborhood solution due to lack of space. The *algorithm prioritizes* the design corners based on the total power and delay. It ensures that all non-critical path resources are assigned less leaky resources. After several trials we found that 50 iterations provide a good trade-off between algorithm performance and cost function reduction. The cost function used in the algorithm is presented in Fig. 3. PF_c is the power fluctuation for clock cycle c , θ and δ are the weights

```

Power_Cost (S, Cell Library)
(01)  $I_{Total_c} = Current(FU_i(V_{DD}, V_{th}, T_{ox}))$ 
(02)  $PF_c = |I_{Total_c} - I_{Total_{c-1}}|$ 
(03)  $I_{Total} = \sum_{c=1}^{N_{cc}} I_{Total_c}$ 
(04)  $PF_{Total} = \sum_{c=1}^{N_{cc}} PF_c$ 
(05) Cost_PDF =  $\theta * I_{Total} + \delta * PF_{Total}$ 
(06) Cost =  $a * \mu(Cost\_PDF) + b * \sigma(Cost\_PDF)$ 
(07) return Cost.

```

Figure 3. Algorithm for Cost Function Calculation

used to normalize the power fluctuation and total current and all the summations are 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 and S. D. with weights a and b , respectively. Depending on whether the objective of the optimization is performance or yield enhancement, a or b , respectively are assigned higher weights.

5 Process Variation Aware Component Library

Initially 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) [3]. The BSIM4 deck generated through BPTM represent a hypothetical 45nm CMOS process. The nominal values for design corner (1) are: $T_{ox} = 1.4$ nm, $V_{th} = 0.22$ V for NMOS, $V_{th} = -0.22$ 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 leakage, dynamic

and subthreshold current and delay probability density distributions (outputs.) In our current experiments we have not taken the statistical variation of L into account. This is because the variation of several parameters would make the optimization problem very complex. This does not affect the framework proposed in this paper and its inclusion is straightforward. It was observed that with normally distributed input parameters, the distribution of the output currents was lognormal while that of the propagation delay was normal.

A library of datapath components was developed using universal NAND logic. At the architectural level we followed a state independent approach by using the state average 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, 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 gate. We can thus 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:

$$\begin{aligned} \mu_{FU} &= N \mu_{NAND}, \\ \sigma_{FU} &= \sqrt{N} \sigma_{NAND}, \end{aligned} \quad (9)$$

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.

From these equations the mean and the variance of I_{gate} , I_{sub} , I_{dyn} , and delay T_{PD} for each of the functional units was calculated. The use of universal NAND gates simplifies the construction of the cell 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 central limit theorem, a realistic assumption for real-life designs. Characterization data for a sample design corner is shown in Fig. 4.

6 Experimental Results

We applied our power and fluctuation optimization technique to several standard high-level synthesis benchmarks

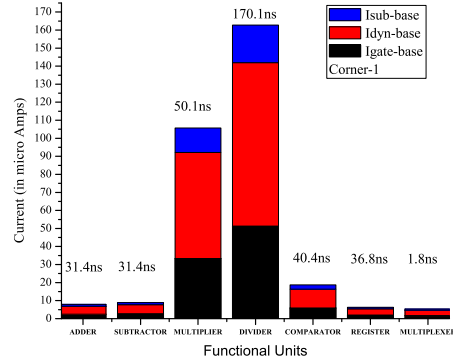


Figure 4. Values of I_{gate} , I_{sub} , I_{dyn} , and T_{PD} (stated at the top of the bar chart), for design corner (1) for various functional units in the following order: adder, subtractor, multiplier, divider, comparator, register and multiplexer.

presented in [11], with the inclusion of the simulated annealing algorithm implemented in C. We consider design corner 1 as the baseline. 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 ranging from 1.0 to 1.4. For each resource constraint these time constraints are applied and exhaustive experiments are performed. Typical simulation time for a benchmark circuit was in the range of 20 to 30 mins. The percentage reduction is calculated as $\Delta = (\text{Baseline} - \text{Final}) / (\text{Baseline}) * 100\%$. Similarly, percentage reductions for I_{gate} , I_{dyn} , I_{sub} , and total power fluctuation are calculated. We estimate the delay of the circuit as the sum of delays of the nodes in the longest path of the DFG.

The results are shown for selected benchmarks in Fig. 5. We have presented results for $\alpha = 1, \beta = 1, \gamma = 1$, as the multicost objective function's weighting factors. It is straightforward to investigate the results for variation of one or two parameters at a time by choosing the values of weighting factors appropriately. For example, $\alpha = 1, \beta = 0, \gamma = 0$ would optimize gate leakage, and $\alpha = 0, \beta = 1, \gamma = 0$ would optimize subthreshold leakage, and so on.

To the best of our knowledge, we did not find behavioral (high-level, or architectural) synthesis research having the same scope as the work presented in this paper i.e. accounting for gate leakage, subthreshold leakage, and dynamic power dissipation simultaneously. Hence, a fair comparison of the presented results is not possible. However, individual results in gate leakage, subthreshold leakage, dynamic power, and power fluctuation are comparable and considerably better than the related works discussed in Section 2.

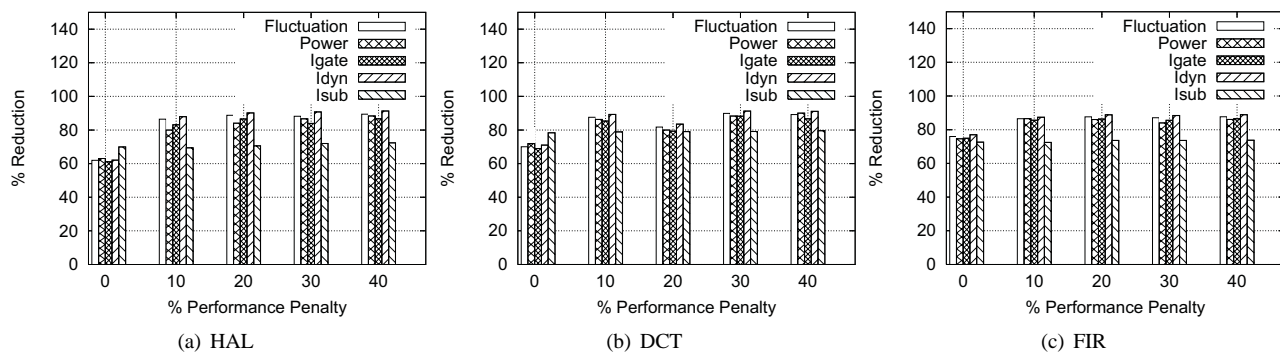


Figure 5. Experimental results showing the percentage reduction in various metrics for selected standard benchmarks and different percentage penalty.

7 Conclusion

In this work a novel process variation aware power and fluctuation minimization methodology was presented. An exhaustive model library was created considering the individual and combined variations of T_{ox} , V_{th} and V_{DD} using transistor level Monte Carlo simulations. The resulting characterization consisted of mean and S. D. of I_{gate} , I_{dyn} , I_{sub} and delay of various functional units. This model library was then used in the process variation aware power and fluctuation minimization model. Significant reduction in average power and fluctuation can be achieved using the proposed methodology.

References

- [1] 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*, 8(3):235–251, June 2000.
- [2] 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*, 48(8):1800–1810, August 2001.
- [3] 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*, pages 201–204, 2000.
- [4] A. Chandrakasan, S. Sheng, and R. W. Brodersen. Low-Power CMOS Digital Design. *IEEE Journal of Solid-State Circuits*, 27:473–483, Apr 1992.
- [5] C. Gopalakrishnan and S. Katkooi. KNAPBIND: An Area-Efficient Binding Algorithm for Low-Leakage Datapaths. In *Proceedings of 21st International Conference on Computer Design*, pages 430–435, 2003.
- [6] S. Gupta and F. N. Najm. Energy and Peak-Current Per-cycle Estimation at RTL. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(4):525–537, Aug 2003.
- [7] K. S. Khouri and N. K. Jha. Leakage power analysis and reduction during behavioral synthesis. In *Proceedings of International Conference on Computer Design*, pages 561–564, 2000.
- [8] R. S. Martin and J. P. Knight. Optimizing Power in ASIC Behavioral Synthesis. *IEEE Design and Test of Computers*, 13(2):58–70, Summer 1996.
- [9] S. P. Mohanty and E. Kougiianos. Modeling and Reduction of Gate Leakage during Behavioral Synthesis of NanoCMOS Circuits. In *Proceedings of the 19th IEEE International Conference on VLSI Design*, pages 83–88, 2006.
- [10] 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)*, pages 249–256, 2005.
- [11] S. P. Mohanty and N. Ranganathan. A Framework for Energy and Transient Power Reduction during Behavioral Synthesis. *IEEE Transactions on VLSI Systems*, 12(6):562–572, June 2004.
- [12] S. P. Mohanty, N. Ranganathan, and S. K. Chappidi. Power Fluctuation Minimization During Behavioral Synthesis using ILP-Based Datapath Scheduling. In *Proceedings of the IEEE International Conference on Computer Design*, pages 441–443, 2003.
- [13] V. Raghunathan, S. Ravi, A. Raghunathan, and G. Lakshminarayana. Transient Power Management Through High Level Synthesis. In *Proc. Int. Conf. Computer-Aided Design*, pages 545–552, 2001.
- [14] K. Roy, S. Mukhopadhyay, and H. M. Meimand. Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits. *Proceedings of the IEEE*, 91(2):305–327, February 2003.
- [15] W. T. Shiue. High-Level Synthesis for Peak Power Minimization using ILP. In *Proc. IEEE Int. Conf. Application Specific Systems, Architectures Processors*, pages 103–112, 2000.