Fast Statistical Full-Chip Leakage Analysis for Nanometer VLSI Systems

RUIJING SHEN, SHELDON X.-D. TAN, and HAI WANG, University of California at Riverside
JINJUN XIONG, IBM Thomas J. Watson Research Center

In this article, we present a new full-chip statistical leakage estimation considering the spatial correlation condition (strong or weak). The new algorithm can deliver linear time, $O(N)$, time complexity, where N is the number of grids on chip. The proposed algorithm adopts a set of uncorrelated virtual variables over grid cells to represent the original physical random variables and the cell size is determined by the spatial correlation length. In this way, each physical variable is always represented by virtual variables locally. We prove the number of neighbor cells for each grid cell is not related to the condition of spatial correlation (from no correlation to 100% correlated), which leads to linear time complexity in terms of number of gates. We compute the gate leakage by the orthogonal polynomials-based collocation method. The total leakage of a whole chip can be computed by simply summing up the coefficients of corresponding orthogonal polynomials in each grid cell. Furthermore, we develop a look-up table to cache statistical information for each *type* of gate instead of calculating leakage for every single instance of gate on a chip. As a result, a new statistical leakage characterization in Standard Cell Library (SCL) is put forward. Furthermore, an incremental analysis algorithm is proposed to update the chip-level statistical leakage information efficiently after a few changes are made. The proposed method has no restrictions on static leakage models, or types of leakage distributions. The large circuit examples in 45nm CMOS process demonstrate the proposed algorithm is 1000X faster than a recently proposed grid-based method with similar accuracy and many orders of magnitude times speedup over the Monte Carlo method. Experimental results also show the incremental analysis provides about 10X further speedup. We expect the incremental analysis could achieve more speedup over the full leakage analysis for larger problem sizes.

Categories and Subject Descriptors: J.6 [**Computer-Aided Engineering**]: Computer-Aided Design

General Terms: Design, Algorithms

Additional Key Words and Phrases: Statistical analysis, spatial correlation, leakage power analysis

ACM Reference Format:
Shen, R., Tan, S. X.-D., Wang, H., and Xiong, J. 2012. Fast statistical full-chip leakage analysis for nanometer VLSI systems. ACM Trans. Des. Autom. Electron. Syst. 17, 4, Article 51 (October 2012), 19 pages.
DOI = 10.1145/2348839.2348855 http://doi.acm.org/10.1145/2348839.2348855

51

1. INTRODUCTION

Leakage power and its proportion in chip power dissipation have increased dramatically with technology scaling [ITRS 2008]. The dominant factors in leakage currents are subthreshold leakage currents I_{sub} and gate oxide leakage currents I_{gate}.

This work is funded by NSF grants CCF-1017090, OISE-1051797, CCF-1116882 and OISE-1130402.
Authors' addresses: R. Shen, S. X.-D. Tan (corresponding author), and H. Wang, Department of Electrical Engineering, University of California at Riverside, Riverside, CA 92521; email: stan@ee.ucr.edu; J. Xiong, IBM Thomas J. Watson Research Center, Yorktown, NY 10562.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permission may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2012 ACM 1084-4309/2012/10-ART51 $15.00
DOI 10.1145/2348839.2348855 http://doi.acm.org/10.1145/2348839.2348855

Table I. Useful Notations and Terminologies

Notation	Description
n/N	# of gates/grids on chip
m	# of types of gate in library
ρ	spatial related covariance
d	distance between two grids
η	correlation length
$I_{sub/gate}$	subthreshold/gate oxide leakage current
$T(j)$	Neighboring set of the jth grid
k	# of grids in neighboring set
$\vec{\xi}_{g,j}$	virtual random variable set located in $T(j)$
$\vec{\xi}_{loc}$	local virtual random variable set
H_i	the i-th Hermite polynomial
$I_{sub/gate,i,j}$	the coefficient for H_i in the j-th grid
C_S, C_G	look-up table for $I_{sub,i,j}$ and $I_{gate,i,j}$
G	gate mapping matrix

Subthreshold leakage currents rapidly increase for every technology generation (about 5× to 10× increase per generation [De and Borkar 1999]), and are highly sensitive to threshold voltage V_{th} variations owing to the exponential relationship between I_{sub} and V_{th}. On the other hand, as gate oxide thickness, T_{ox}, scales down, I_{gate} grows rapidly as I_{gate} has an exponential dependence on T_{ox}.

Both leakage currents are highly sensitive to process variations due to the exponential relation between the leakage current and variational parameters like effective channel lengths. As process-induced variability becomes more pronounced in the deep submicro regime [Nassif 2000], leakage variations become more significant, and traditional worst-case-based approaches will lead to extremely pessimistic and expensive overdesigned solutions. Statistical estimation and analysis of leakage powers considering process variability are critical in various chip design steps to improve design yield and robustness.

Recently, Bhardwaj et al. [2008] presented a unified approach for statistical timing and leakage current analysis using quadratic polynomials. However, this method only considers the long-channel effects and ignores the short-channel effects (ignoring channel length variables) for the gate leakage models. The coefficients of the orthogonal polynomials at gate level are computed directly by the interproduction via the efficient Smolyak quadrature method. The method also tries to reduce the number of variables via the moment matching method, which further speeds up the quadrature process at the cost of more added errors.

In this article, first, we present a new linear-time algorithm for statistical leakage analysis in the presence of any spatial correlation (from no spatial correlation to 100% correlated situation). The new algorithm exploits the following property: leakage current of a gate in the presence of spatial correlation is affected by process variations in the neighbor area. As a result, gate leakage current can be efficiently computed by considering the neighbor area in constant time. We adopt a newly proposed spatial correlation model where a new set of location-dependent uncorrelated virtual variables are defined over grid cells to represent original correlated random variables via fitting. To compute the statistical leakage current of a gate on the new set of variables, the orthogonal polynomials-based collocation method is applied and the variational gate leakages and total leakage currents are represented in an analytic form in terms of the random variables, which can give complete statistic information. The new method

considers both interdie and intradie variations and can work with any spatial correlations (strong or weak, as defined in Section 3.1). Unlike the existing approaches [Chang and Sapatnekar 2007; Heloue et al. 2007], the new method does not make any assumptions about the final distributions of total leakage currents for both gate and chip levels. In case of medium and strong correlations, the proposed method can also work in linear time by properly sizing the grid cells so both locality of correlation and accuracy are still preserved.

Furthermore, we bring forth a novel characterization of standard cell library for statistical leakage information and we have the following observations: (1) The set of neighbor cells is usually small ($\sim$ 10), and only considering the relative position, not the absolute position on chip. (2) As proved later, the number of neighbor cells involved in our model is not related to the strength (level) of spatial correlation. (3) The orthogonal polynomials-based stochastic collocation method is applied and the variational leakage of a gate is represented in an analytic form in terms of the virtual random variables, which can give complete distribution. (4) The gate-level leakage distribution is only related to the type of gates in a standard cell library. This statistical leakage characterization can be stored in a look-up table, which only needs to be built once for a standard cell library. And the full-chip leakage of any chip can be easily calculated by summing up certain items in the look-up table.

The main contributions of the this article are as follows.

(1) We applied the virtual-grid-based model for spatial correlation modeling in the statistical leakage analysis and making the resulting algorithm linear time for the first time for all the spatial correlation (weak or strong) cases.
(2) We proposed a new characterization in standard cell library for statistical leakage analysis. The corresponding algorithm can accelerate full-chip statistical analysis for all spatial correlation conditions (from weak to strong). To the best knowledge of the authors, the proposed approach is the first published algorithm which can guarantee $O(N)$ time complexity for all spatial correlation conditions.
(3) Furthermore, an incremental algorithm has been proposed. When a few local changes are made, only a small circuit (includes the changing gates) is involved in the updating process. Our numerical examples show the incremental analysis can achieve $10\times$ further speedup compared with the library-enabled full-chip analysis approach.

Experimental results on the PDWorkshop91 benchmarks on a 45nm technology show the proposed method using novel characterization in standard cell library is on average two orders of magnitude faster than the recently proposed method [Chang and Sapatnekar 2007] with similar accuracy. For weak correlation situation, more speedup can be observed.

The rest of this article is organized as follows: Section 2 reviews the existing works. Section 3 shows the process variational models and gate-level static leakage models used in this work. Section 4 presents the new spatial correlation models. Section 5 presents our chip-level analysis framework. Section 6 describes the new statistical characterization in standard cell library for leakage analysis, and how to speed up analysis in Section 5 based on it. Section 7 presents the experimental results and Section 8 concludes.

2. PRIOR WORKS

Many earlier works, such as Narendra et al. [2004], Mukhopadhyay and Roy [2003], Rao et al. [2004], and Wang et al. [2005], have been proposed for full-chip statistical leakage analysis considering the process variations. When the process-induced

variabilities are spatially correlated, the computational cost of the distribution of total leakage of the chip becomes quadratic – $O(n^2)$, where n is number of gates as the variance of a gate has to be computed with respect to all correlated gates. To mitigate this problem, several approaches have been proposed recently [Chang and Sapatnekar 2007; Heloue et al. 2007; Kim et al. 2010; Li and Shi 2006; Shen et al. 2009]. The work in Chang and Sapatnekar [2007] partitions the whole chip into many grid cells (the gate variables in a cell are identical) to reduce the number of variables at the loss of accuracy. This method works well when correlation is strong. The approach in Shen et al. [2009] also explores the strong spatial correlation by reducing the number of variables using Principal Component Analysis (PCA) and orthogonal polynomials to represent leakage currents at the gate and whole chip levels. The PCA-based approach in Li and Shi [2006] works well when random variables are uncorrelated. In Heloue et al. [2007], a linear time complexity method is proposed, but it assumes symmetric spatial correlation and considers only the channel length variations.

Based on Chang and Sapatnekar [2007], two different improvements have been done in Chang and Sapatnekar [2007] and Kim et al. [2010]. A hybrid version of Chang and Sapatnekar [2007] with PCA is provided in Chang and Sapatnekar [2007] for improved speed. On the other hand, Kim et al. [2010] proposed another approach that can achieve further speedup compared to Chang and Sapatnekar [2007] based on the approximation that the sum of two lognormal distributions is still a lognormal distribution. The assumption is one source of error in this case.

When the spatial correlation is weak, existing general grid-based approaches and PCA-based approach such as Shen et al. [2009], Chang and Sapatnekar [2007], and Kim et al. [2010] do not work well as the number of correlated variables cannot be reduced too much. Recently an efficient method was proposed [Ye and Yu 2009] to address this problem. The method is based on simplified gate leakage models and formulates the major computation tasks into matrix-vector multiplications via Taylor's expansion. It then applies fast numerical methods like the fast multipole method or the precorrected FFT method to compute the multiplication. However, this method assumes the gate-level leakage currents are purely lognormal, and the chip-level leakage is also approximated by lognormal distribution, which is not the case as we will show in the article. Also it can only give the mean and variances, not the complete distribution of the leakage powers.

3. VARITIONAL AND LEAKAGE MODELING

In this section, we present the spatial process variational models and gate leakage models used in this work.

3.1. Variational Models

For a gate/module in a chip, the variation of a process parameter X can be modeled as the following random variables

$$\Delta X = \Delta X_{inter} + \Delta X_{intra}, \quad (1)$$

where ΔX_{inter} and ΔX_{intra} are independent random variables which are the interdie and intradie components of X, respectively. ΔX_{inter} is the same for all gates in all grids. On the other hand, ΔX_{intra} is different for each gate, and may have spatial correlations. Since the spatial correlation is the focus of this work, ΔX is used to represent the intradie variation ΔX_{intra} for brevity's sake.

The spatial correlation coefficient between two grids, represent by ρ, can be modeled by an empirical formulation such as the exponential model [Xiong et al. 2007]. We have

$$\rho(d) = e^{-d^2/\eta^2} \in (0, 1), \quad (2)$$

where d is the distance between two grid centers and η is a constant, named the correlation length. Notice that (2) is based on the assumption of isotropy. As discussed in Xiong et al. [2007], the homogeneous isotropic random field is the simplest model for intradie variation with reasonable accuracy, where spatial correlation depends only on distance. And if nonisotropic variation is involved in other applications, the proposed technique may not work directly. But the isotropy assumption can be viewed at least as the first-order approximation to this correlation modeling problem.

From the mathematical theory, correlation coefficients ρ which are close to 1 indicate strong correlation, and ρ which are close to 0 indicate weak correlation. Therefore, the definition of strong and weak spatial correlation can be provided based on correlation length η as follows.

Definition 1. When $\eta \gg$ size of chip, which means $\rho(d) \sim 1$, we call the spatial correlation **strong**.

Definition 2. When $\eta \sim$ minimum distance between two gates, which means $\rho(d) \sim 0$, we call the spatial correlation **weak**.

More details about correlations can be found in Dowdy and Wearden [2004].

The spatial correlation can be captured by the spatial covariance matrix Ω_n, where n is the number of gates. The elements in Ω_n are modeled as (2), which are only related to d.

The transistor threshold voltage parameter, V_{th}, has the biggest impact on leakage current. V_{th} is observed to be the most sensitive to effective gate/channel length L and gate oxide thickness T_{ox}. ITRS [2008] indicates the variation of L is a primary factor for device parameter variation, and the number of dopants in channel results in an unacceptably large statistical variation of V_{th}.

Following the existing approaches, we also assume the process variations of L and T_{ox} follow multivariate normal distributions [Duvall 2000; Teodorescu et al. 2007], and both of them include interdie and intradie components as shown in (1). Since T_{ox} is in vertical layout feature dimension, and is mainly caused by chemical mechanical polishing processes, it only depends on local layout density and has no spatial correlation [Ouma et al. 2002]. Therefore, we focus on the spatial correlation of ΔL_{intra}. And the intradie variation in T_{ox} is modeled as uncorrelated random variables. Our approach can be easily extended to deal with multiple spatially correlated variables. Since the interdie part is much simpler, we only discuss the intradie component of variation in detail. In the following part ΔL is used to present the intradie component of L for short.

As mentioned in Section 2, the grid-based method and the PCA-based method have difficulty dealing with medium or weak spatial correlations. In this article, we try to address this outstanding issue and propose a new solution of full-chip statistical leakage analysis, which works for both strong or weak spatial correlations as shown in the following sections.

3.2. Static Leakage Modeling for Gates

Full-chip leakage current has two major components, subthreshold leakage current and gate leakage current. Here we describe the empirical models for both of them. The reason for doing this is to avoid full SPICE simulation of a gate to estimate a gate leakage currents for variational parameter settings.

The subthreshold leakage current, I_{sub}, is exponentially dependent on the threshold voltage, V_{th}. And V_{th} is observed to be most sensitive to gate oxide thickness T_{ox} and effective gate channel length L due to short-channel effects. When the change in L or T_{ox} is small, the precise relationship shows an exponential dependent effect on I_{sub},

with the effect of T_{ox} relatively weak. For the gate oxide leakage current, both channel length and oxide thickness have strong impacts on the leakage currents, which are exponential functions of the two variables.

In our work, we also follow the analytical expressions given in Chang and Sapatnekar [2007], which estimate the subthreshold leakage currents and the gate oxide leakage currents as

$$I_{sub} = e^{a_{1,s}+a_{2,s}L+a_{3,s}L^2+a_{4,s}T_{ox}^{-1}+a_{5,s}T_{ox}}, \quad (3)$$

$$I_{gate} = e^{a_{1,g}+a_{2,g}L+a_{3,g}L^2+a_{4,g}T_{ox}+a_{5,g}T_{ox}^2}, \quad (4)$$

where $a_{1,s}$ through $a_{5,s}$ and $a_{1,g}$ through $a_{5,g}$ are the fitting parameters of I_{sub} and I_{gate} for each unique input combination of a gate, respectively. Then we can use a look-up table to store the fitting parameters.

Since process variables L and T_{ox} follow Gaussian distribution, about 99.7% of values drawn from these distributions are within three standard deviations (known as the 3-sigma rule). Therefore, we choose 100 points for L and T_{ox} in their 3σ region linearly. After SPICE simulation at each point, the leakage currents are stored as the original curve. Then we can perform the curve fitting process.

4. VIRTUAL GRID-BASED SPATIAL CORRELATION MODEL

The virtual grid-based model is based on the observation that the leakage current of a gate in the presence of spatial correlation only correlates to its neighbor area. If we can introduce a set of uncorrelated variables to model the localized correlation, computing the leakage current of one gate can be done in a constant time by only considering its neighbor area. Hence total full-chip statistical leakage currents can then be computed by simply adding all the gate leakage currents together in terms of the virtual set of variables in linear time. Notice that the virtual random variables in different grids are always independent, which is different from a traditional grid-based model. This idea was proposed recently for fast statistical timing analysis [Chen et al. 2008] to address the computational efficient modeling for weak spatial correlation, which is similar to the PCA-based approach [Shen et al. 2009], but with different set of independent variables.

Specifically, the chip area is still divided into a set of grid cells. When the spatial correlation is weak enough to be ignored, the cell can become so small that one cell only contains one gate. Then we introduce a “virtual” random variable for each cell for one source of process variation.

These virtual random variables are *independent* and will be the basis for statistical leakage current calculation concerned with spatial correlation. Then we can express the original physical random variable of a gate in a grid cell as a linear combination of the virtual random variables of its own cell as well as its nearby neighbors. Since virtual random variable in each cell has specific location on-chip, such a location-dependent correlation model still retains the important spatial physical meaning (in contrast to PCA-based models). The grid partition can be made of any shape. We use hexagonal grid cells [Chen et al. 2008] in this article since they have minimum anisotropy for 2D space.

Here we define the distance between centers of two direct neighbor grid cells as the grid length d_c. Gates located in the same cell have strong correlation (larger than a given threshold value ρ_{high}) and are assumed to have the same parameter variations. And “spatial correlation distance” d_{max} is defined as the minimum distance beyond which the spatial correlation between any two cells is sufficiently small (or smaller than a given threshold value ρ_{low}) so we can ignore it.

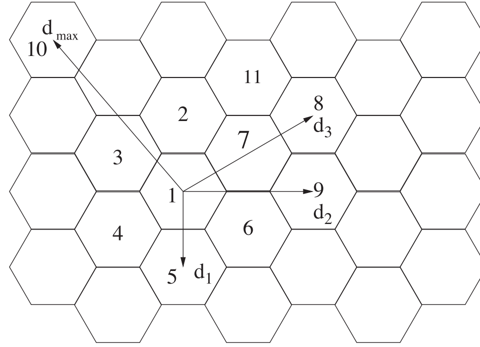

Fig. 1. Location-dependent modeling with the $T(i)$ of grid cell i defined as its seven neighbor cells.

In this model, the jth grid cell is associated with one virtual random variable $\xi_j \sim N(0,1)$, which is independent of all other virtual random variables. ΔL_j can then be expressed as its k closest neighbor cells. We introduce the concept of *correlation index neighbor* set $T(j)$ for cell j, and the corresponding variable vector, $\vec{\xi}_{g,j}$, is defined as

$$\vec{\xi}_{grid_j} = [\xi_q, q \in T(j)], \tag{5}$$

to model the spatial correlation of ΔL_j as

$$\Delta L_j = \sum\nolimits_{q \in T(j)} \alpha_q \cdot \xi_q. \tag{6}$$

For example, a hexagonal grid partition is used as shown in Figure 1, and if $T(i)$ for each cell is defined as its closest $k = 7$ neighbor cells, then ΔL located at cell (x_i, y_i) can be represented as a linear combination of seven virtual random variables located in its neighbor set. Take ΔL_1 in Figure 1 for instance, we have $\Delta L_1 = \alpha_1\xi_1 + \alpha_2\xi_2 + \ldots + \alpha_7\xi_7$.

This concept of virtual random variable helps to model the spatial correlation. Two cells close to each other will share more common spatial random variables, which means the correlation is strong. On the other hand, two cells physically far away from each other will share less or no common spatial random variables. In this way, the spatial correlation is modeled as a *homogeneous and isotropic random field* and the spatial correlation is only related to distance. That is to say, spatial correlation can be fully described by $\rho(d)$ in (2). d_{max} is the distance beyond which $\rho(d)$ becomes small enough to be approximated as zero.

Since $\rho(d)$ is only a function of distance, the number of unique distance values between two correlated grid cells equals the number of unique element values in Ω_N. From Figure 1, the spatial correlation distance equals to the distance between cell 1 and cell 10 which is $d_{max} = \sqrt{7}d_c$, and there are only three unique correlation distances d_1 to d_3. Correspondingly, there are only three unique elements in Ω_N, without including two special values: 0 for $d \geq d_{max}$ or 1 for distance within one cell.

Furthermore, the same correlation index can be used for all grid cells and the coefficient α_k should be the same for the same distance because of the homogeneousness and isotropy of spatial correlation. For the cell marked 1 in Figure 1, we only have two unique values among the seven coefficients, that is, we set $p_0 = \alpha_1$, $p_1 = \alpha_i, i = 2, 3, \ldots, 7$. In other words, we have

$$\Delta L_1 = p_0\xi_1 + p_1(\xi_2 + \ldots + \xi_7). \tag{7}$$

In this way, although there are seven random variables involved in the neighbor set, there are only two unknown coefficients left in the linear function in (7) due to the symmetry property of hexagonal partition.

According to (2), a nonlinear overdetermined system can be built to determine the two unique values of p_0, p_1 as follows.

$$\begin{aligned}\rho(0) &= E(\Delta L_1^2) = p_0^2 + 6p_1^2 \\ \rho(d_1) &= E(\Delta L_1, \Delta L_2) = 2p_0p_1 + 2p_1^2 \\ \rho(d_2) &= E(\Delta L_1, \Delta L_9) = 2p_1^2 \\ \rho(d_3) &= E(\Delta L_1, \Delta L_8) = p_1^2\end{aligned} \tag{8}$$

The system in (8) can be solved by formulating them as a nonlinear least square optimization problem. In the matrix form, we can rewrite (6) for a whole chip as

$$\Delta L = P_{N,N} \cdot \vec{\xi}, \tag{9}$$

where N is the number of grid cells, and $\vec{\xi} = [\xi_1, \xi_2, \ldots, \xi_N]$. According to (6), the correlation index set contains only k spatial random variables, which is a very small fraction of the total spatial random variables. As a result, $P_{N,N}$ is a sparse matrix. Every gate only is concerned with k virtual random variables, which has specific location information. Notice that the partition can be in any shape. For example, triangle or octagon partitions can be used for simple computation or higher accuracy. The only difference is the way to define grid size and the linear equations in (8) to determine coefficients of virtual random variables in (7).

Fundamentally, the PCA-based method performs a similar process and has a similar new transformation matrix between the original and new set of variables. We have

$$\Delta L = V_{n,n} \cdot \vec{\xi}, \tag{10}$$

where $V_{n,n}$ is the transformation matrix obtained from eigenvalue decomposition of the correlation matrix in PCA. The major difference is that $V_{n,n}$ is a dense matrix even though the original correlation matrix is sparse. This makes a huge difference especially when the spatial correlation is weak as eigendecomposition will take almost $O(n^3)$ to compute. The virtual independent spatial correlation model also works for medium and strong correlation cases, which will be shown in the next section.

5. LINEAR CHIP-LEVEL LEAKAGE POWER ANALYSIS METHOD

In this section, we will present the new full-chip statistical leakage analysis method. We first introduce the overall flow of the proposed method and highlight the major computing steps. The new algorithm flow is summarized in Figure 2.

The new algorithm consists of three major parts. The first part (Step 1 and 2) is precharacterization. Step 1 builds the analytic leakage expressions (3) and (4) for each type of gates, which only needs to be done once for a standard cell library. Step 2 deals with a small-sized nonlinear overdetermined system, which can be solved with any least square optimization algorithm. The second part (Step 3) generates a small set of independent virtual random variables and builds the analytic leakage current expressions and covariances for each gate on top of the new random variables. The final part (Step 4) computes the final full-chip leakage expressions by simple polynomial additions. From the final expressions, we can calculate important statistical information (like mean, variance, and even the whole distributions). In the following, we briefly explain some important steps.

Algorithm: NEW FULL-CHIP STATISTICAL LEAKAGE ANLAYSIS.

Input: standard cell lib, netlist, placement information of design, standard deviation of L and T_{ox}.
Output: analytic expression of the full-chip leakage currents in terms of Hermite polynomials.

(1) Generate a_1 through a_5 for I_{sub} and I_{gate} in (3) and (4) for each type of gates (Section 3.2).
(2) Solve (8) to determine coefficients in (7).
(3) Calculate the coefficients of Hermite polynomial of I_{sub} and I_{gate} for the leakage analytic expression for each gate.
(4) Calculate the analytic expression of the full-chip leakage current by simple polynomial additions and calculate mean value, standard deviation, PDF and CDF of the leakage current if required.

Fig. 2. The flow of proposed algorithm.

5.1. Computing Gate Leakage by the Orthogonal Polynomial Method

In the following, we briefly review the orthogonal polynomial-based modeling approaches. Note that for Gaussian and log-normal distributions, Hermite polynomial is the best choice as it leads to exponential convergence rate [Ghanem and Spanos 2003]. For non-Gaussian and nonlog-normal distributions, there are other orthogonal polynomials. The proposed method can be extended to other distributions with different orthogonal polynomials.

A random variable $y(\vec{\xi})$ with a limited variance can be approximated by Hermite polynomial expansion as [Ghanem and Spanos 2003]

$$y(\vec{\xi}) = \sum a_i H_i(\vec{\xi}), \tag{11}$$

where $\vec{\xi} = [\xi_1, \xi_2, ..., \xi_k]$ is a vector of independent identically distributed (i.i.d.) Gaussian random variables, $H_i(\vec{\xi})$ is Hermite polynomial, and a_i is the deterministic coefficient. a_i can be determined by

$$a_i = \frac{< y(\vec{\xi}), H_i(\vec{\xi}) >}{< H_i^2(\vec{\xi}) >} \approx \sum\nolimits_{l=0}^{S} y(\gamma_l) H_i(\gamma_l) w_l, \tag{12}$$

where γ_l and w_l are Gaussian Hermite quadrature abscissas (quadrature points) and weights, respectively. According to Hermite polynomials' characteristics, $< H_i^2(\vec{\xi}) >= 1$ in (12). The kernel of this technique is to calculate the coefficients of Hermite polynomials efficiently. For multiple random variables, which require multidimensional quadrature, Smolyak quadrature [Novak and Ritter 1999] is used as an efficient method to reduce the number of quadrature points. For z-th order Hermite polynomials and k-dimensional integration in (12), the size of Smolyak quadrature point set is $O(k^z/(z)!)$.

In our problem, $y(\vec{\xi})$ will be the leakage current for each gate, and eventually for the full chip. For the jth gate, from (6), ΔL_j only relates to k independent virtual random variables in $T(j)$. Since k is a small number, Step 3 in Figure 2 can be very efficient.

To compute the gate leakage current, we need to present both I_{sub} and I_{gate} of each gate in the second-order Hermite polynomials, respectively,

$$I_{sub}(\vec{\xi}_{grid_j}) = \sum\nolimits_{i=0}^{P} I_{sub,i,j} H_i(\vec{\xi}_{grid_j}), \tag{13}$$

$$I_{gate}(\vec{\xi}_{grid_j}) = \sum\nolimits_{i=0}^{P} I_{gate,i,j} H_i(\vec{\xi}_{grid_j}), \tag{14}$$

where $H_i(\vec{\xi}_{grid_j})$ are second-order Hermite polynomials defined as follows.

$$H_{0th} = 1,\ \ H_{1st} = \xi_i$$
$$H_{2nd,type1} = \xi_i^2 - 1,\ \ H_{2nd,type2} = \xi_i\xi_j,\quad (i \neq j) \tag{15}$$

And $I_{sub,i,j}$ and $I_{gate,i,j}$ are then computed by the numerical Smolyak quadrature method in (12).

Notice that the time complexity of computing leakage for a gate is $O(k^2)$. And the number of involved independent random variables k is very small compared to total number of gates. The analytic expression is also a function of those involved random variables.

5.2. Computation of Full-Chip Leakage Currents

After the leakage currents are calculated for each gate, we can proceed to compute the leakage current for the whole chip as follows.

$$I_{chip}(\vec{\xi}) = \sum\nolimits_{j=1}^{n}(I_{sub}(\vec{\xi}_{grid_j}) + I_{gate}(\vec{\xi}_{grid_j})) \tag{16}$$

The summation is done for each coefficient of Hermite polynomials. Then we obtain the analytic expression of the final leakage currents in terms of $\vec{\xi}$.

We can then obtain the mean value and variance of full-chip leakage current very easily as

$$\mu_{chip} = I_{chip,0th}, \tag{17}$$

$$\sigma_{chip}^2 = \sum I_{chip,1st}^2 + 2\sum I_{chip,2nd,type1}^2 + \sum I_{chip,2nd,type2}^2, \tag{18}$$

where $I_{chip,ith}$ is the leakage coefficient for ith Hermite polynomial of second order defined in (15). Since Hermite polynomials with orders higher than two have no contribution to mean value or standard deviation, second order is good enough for estimating μ_{chip} and σ_{chip} in (17) and (18).

5.3. Time Complexity Analysis

To analyze the time complexity, one typically does not count the precharacterization cost of Step 1 in Figure 2. Besides, Step 2 is solving a tiny size of least square problem (with the neighboring set used here, there are 4 equations in (8) and only two unknown variables p_0 and p_1 to be solved), which needs to be done only once for one chip. Therefore, the runtime in Step 2 is ignorable compared to the following steps. In Step 3, we need to compute the weights of the Level 2 k-dimensional Smolyak quadrature point set. For a quadratic model with $k+3$ variables, the number of Smolyak quadrature points is $S \sim O(k^2)$ based on the discussion in Section 5.1. So the time cost for generating the Smolyak quadrature points set is $O(k^2)$. In Step 4, we need to call (3) and (4) S times for each gate. In each call, we need to compute $k+3$ variables in the Hermite polynomials. The computational cost for the two steps is $(O(nk \times S))$, where n is the number of gates. After the leakage currents are computed for each gate, it takes $O(n(k+3))$ to compute the full-chip leakage current.

For the second-order Hermite polynomials, $S \propto k^2$, and the k is the number of grid cells in the correlated neighbor index set, which is a very small constant number (more detailed discuss of k will be presented in Section 6.5). As a result, the time complexity of our approach becomes linear – $O(n)$.

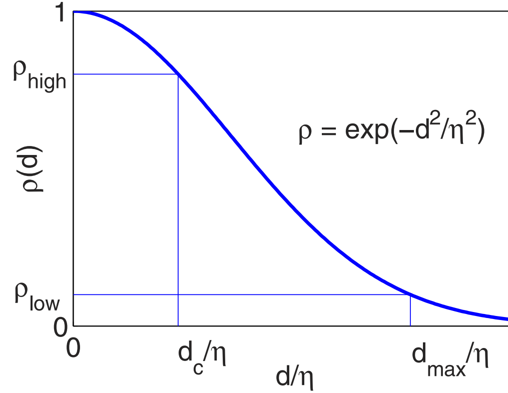

Fig. 3. Relation between $\rho(d)$ and d/η.

6. NEW STATISTICAL LEAKAGE CHARACTERIZATION

In this section, we will present why a new characterization modeling statistical leakage can be added to SCL, and how it can be applied in our new full-chip statistical leakage analysis method.

6.1. Acceleration by Look-Up Table Approach

The spatial correlation in (6) is related to the distance between two grid cells. As a result, neighbor set $T(i)$ represents the relative location, not the absolute location. In other words, a local neighbor set T and a local set of variables $\vec{\xi}_{loc} = [\xi_1, \ldots, \xi_k]$ can be shared by all the gates in all the cells.

The local neighbor set T and the coefficients in (6) are determined by d_{max}/d_c. From the specific spatial correlation model in (2), (as shown in Figure 3),

$$d_{max} = \eta\sqrt{-\ln(\rho_{low})},\;\; d_c = \eta\sqrt{-\ln(\rho_{high})}, \tag{19}$$

then the ratio of spatial correlation distance d_{max} over grid length d_c becomes

$$d_{max}/d_c = \sqrt{ln(\rho_{low})/ln(\rho_{high})}. \tag{20}$$

Once the threshold values ρ_{high} and ρ_{low} are set, d_{max}/d_c is not related to the correlation length η. This means we can determine the grid length once we know the spatial correlation distance for a specific correlation formula at cost of controlled errors (by ρ_{high} and ρ_{low}).

Furthermore, (20) shows the spatial correlation (strong or weak) has nothing to do with T and the virtual random variables used in our model. At the same time, the fitting parameters of static leakage in (3) and (4) are only related to the types of gates in a library. As a result, the coefficients of Hermite polynomials for the leakage of one gate are only functions of the type of the gate, ρ_{high} and ρ_{low}. Therefore, a simple look-up table can be used to store the coefficients of Hermite polynomials of each *type* of gates in the library. In other words, we do not need to compute the coefficients of Hermite polynomials for each gate, just look them up from the table instead. This makes a big difference, as the time complexity is reduced from $O(n)$ to $O(N)$, where n is the number of gates and N is the number of grid cells on chip.

For the look-up table, suppose Q is the number of Hermite polynomials involved and m is the number of gate types in the library, then it includes two matrices as follows.

$$C_S = \begin{pmatrix} I_{sub,1,1} & I_{sub,1,2} & \dots & I_{sub,1,m} \\ I_{sub,2,1} & I_{sub,2,2} & \dots & I_{sub,2,m} \\ \vdots & \vdots & \dots & \vdots \\ I_{sub,Q,1} & I_{sub,Q,w} & \dots & I_{sub,Q,m} \end{pmatrix} = \begin{pmatrix} \vec{I}_{sub,1} & \vec{I}_{sub,2} & \dots & \vec{I}_{sub,m} \end{pmatrix}$$

$$C_G = \begin{pmatrix} I_{gate,1,1} & I_{gate,1,2} & \dots & I_{gate,1,m} \\ I_{gate,2,1} & I_{gate,2,2} & \dots & I_{gate,2,m} \\ \vdots & \vdots & \dots & \vdots \\ I_{gate,Q,1} & I_{gate,Q,w} & \dots & I_{gate,Q,m} \end{pmatrix} = \begin{pmatrix} \vec{I}_{gate,1} & \vec{I}_{gate,2} & \dots & \vec{I}_{gate,m} \end{pmatrix} \quad (21)$$

Here $I_{sub,q,j}$ represents the coefficient of H_q for jth kind of gate in the library for sub-threshold leakage; and $I_{gate,q,j}$ represents the coefficient of H_q for jth kind of gate in the library for gate oxide leakage. C_S and C_G are $Q \times m$ matrices. To build up the look-up table, $\vec{I}_{sub,j}$ and $\vec{I}_{gate,j}$ are computed by the numerical Smolyak quadrature method in (12) for the jth kind of gate. Notice the table needs to be built only once and can be reused for different designs with different conditions of spatial correlations since the new algorithm is independent of spatial correlation length η or the circuit design information. In this way, the look-up table actually builds a new characterization in SCL, which presents the statistical leakage behavior of each standard cell.

6.2. Enhanced Algorithm

The enhanced new algorithm consists of two parts. The first part is precharacterization as shown in Figure 4. We build analytic leakage current expressions for each kind of gate on top of a small set of independent virtual random variables. For fixed values of ρ_{high}, ρ_{low}, and one library, a new characterization is added to the SCL by building a look-up table, which stores coefficients of Hermite polynomials of I_{sub} and I_{gate} for the leakage analytic expressions for each kind of gate. This process only needs to be done once for one LIBRARY, given ρ_{high} and ρ_{low}. Besides, it involves a small-size nonlinear overdetermined problem, which can be solved fast with any least square algorithm.

When we deal with full-chip statistical leakage analysis, the coefficients of local Hermite polynomials in the neighbor grid cell set for each cell can be simply calculated by the look-up table. After transferring the local coefficients to corresponding global positions, we can compute the final full-chip leakage expressions by simple polynomial additions. From the resulting expression, we can calculate other statistical information (like mean, variance, and even the whole distributions). The new algorithm flow is summarized in Figure 5. In the following, we briefly explain some important steps.

6.3. Computation of Full-Chip Leakage Currents

Here we define a gate mapping matrix as follows. We have

$$G_{N\times m} = \{g_{i,j}\}, \quad (22)$$

where $g_{i,j}$ represents the number of jth kind of gate in the library located in the ith grid cell. Then the coefficients of local Hermite polynomials in the neighbor set for all the cells on chip can be easily calculated by the look-up table as follows.

$$I_{sub,loc} = G \cdot C_S^T, \quad I_{gate,loc} = G \cdot C_G^T \quad (23)$$

Algorithm: CHARACTERIZATION OF STATISTICAL LEAKAGE INFORMATION IN SCL

Input: standard cell lib, ρ_{high}, ρ_{low}.
Output: look-up table for coefficients of Hermite polynomials of I_{sub} and I_{gate} for the leakage analytic expressions for each kind of gate.

(1) Generate fitting parameter matrices a_{sub} and a_{gate} of I_{sub} and I_{gate} in (3) and (4) for each type of gates (after SPICE simulation on each input pattern) (Section 3.2).
(2) Calculate d_{max}/d_c from ρ_{high} and ρ_{low} to determine the neighbor set. And then solve (8) to determine coefficients in (7).
(3) Generate Smolyak quadrature points set Θ_z^2 with corresponding weights.
(4) Calculate the coefficients of Hermite polynomials of I_{sub} and I_{gate} for the leakage analytic expressions for each kind of gate in library.

Fig. 4. The flow of statistical leakage characterization in standard cell library.

Algorithm: NEW FULL-CHIP STATISTICAL LEAKAGE ANALYSIS ALGORITHM.

Input: Look-up table for coefficients of Hermite polynomial of I_{sub} and I_{gate} for the leakage analytic expression for each kind of gate. netlist, placement information of design, standard deviation of L and T_{ox}.
Output: analytic expression of the full-chip leakage currents in terms of Hermite polynomials.

(1) For every grid cell on chip, calculate the coefficients of local Hermite polynomials in the neighbor cell set by the look-up table.
(2) Transfer the local coefficients to their corresponding global positions.
(3) Calculate the analytic expression of the full-chip leakage current by simple polynomial additions and calculate mean value, standard deviation, PDF and CDF of the leakage current if required.

Fig. 5. The flow of the proposed algorithm using statistical leakage characterization in standard cell library.

In order to get the full-chip leakage current, the local coefficients need to be transferred to their corresponding global positions

$$T(i) = (x_i, y_i) + T. \tag{24}$$

For the ith grid cell, the local set of random variables $\vec{\xi}_{loc}$ should be transferred to the corresponding positions in $T(i)$. Therefore, $I_{sub,loc}$ and $I_{gate,loc}$ can be transferred to the corresponding global coefficients based on the global virtual random variable set $\vec{\xi}$. For example, the coefficient of ξ_i in the ith cell is

$$I_{sub}(\xi_i) = \sum_{k, i \in T(k)} I_{sub,loc}(\xi_{T(k)-(x_k,y_k)}). \tag{25}$$

Next, we can proceed to compute the leakage current of the whole chip as follows.

$$I_{chip}(\vec{\xi}) = \sum I_{sub}(\vec{\xi}) + I_{gate}(\vec{\xi}) \tag{26}$$

The summation is done for each coefficient of global Hermite polynomials to obtain the analytic expression of the final leakage currents in terms of $\vec{\xi}$. We can then obtain the mean value, variance, PDF, and CDF of the leakage current very easily. For instance, the mean value and variance for the full-chip leakage current are

$$\mu_{chip} = I_{chip,0th}, \tag{27}$$

$$\sigma^2_{chip} = \sum I^2_{chip,1st} + 2\sum I^2_{chip,2nd,type1} + \sum I^2_{chip,2nd,type2}, \tag{28}$$

where $I_{chip,ith}$ is the leakage coefficient for ith Hermite polynomial of second order defined in (15).

6.4. Incremental Leakage Analysis

During the leakage-aware circuit optimizations, a few small changes might be made to the circuit. But we do not want to compute the whole chip leakage from scratch again. In this case, incremental analysis become necessary. In this section, we show how this can be done in our look-up-table-based framework.

First, we only consider the case where one gate is changed for brevity. After that, the general incremental approach is proposed to handle a number of gates.

Assume one gate located in the ith grid cell is changed (e.g., a jth type of gate is replaced by a $(j+1)$th type), resulting in

$$I^{new}_{chip} = I^{old}_{chip} - I^{old}_{grid-i} + I^{new}_{grid-i}, \tag{29}$$

where I^{new}_{chip} and I^{old}_{chip} denote the full-chip leakage currents after and before change, respectively; and I^{old}_{grid-i} and I^{new}_{grid-i} are the leakage currents in the ith grid cell before and after change, respectively.

As defined in (22), $g_{i,j}$ in the gate mapping matrix represents the number of jth kind of gate in the library located in the ith cell on a chip. Therefore, we can quickly generate the new gate mapping matrix G^{new} by updating only two elements in G^{old}

$$g^{new}_{i,j} = g^{old}_{i,j} - 1,$$

$$g^{new}_{i,j+1} = g^{old}_{i,j+1} + 1. \tag{30}$$

In the incremental analysis processes, we can consider the updating part as a small circuit, in which there is only one grid cell (the ith cell on chip) and only two types of gates in the library (the jth and the $(j+1)$th). Then the updating gate mapping matrix is

$$G^{update} = [-1 \quad 1], \tag{31}$$

and look-up tables in (21) using in the small circuit are only

$$C^{update}_S = [I_{sub,j}, I_{sub,j+1}]$$

$$C^{update}_G = [I_{gate,j}, I_{gate,j+1}], \tag{32}$$

where $I_{sub,j/(j+1)}$, $I_{gate,j/(j+1)}$ are the $j/(j+1)$th column in C_S and C_G, respectively.

In general, if a small part of the design is changed, which results in

$$I^{new}_{chip} = I^{old}_{chip} + I^{update}, \tag{33}$$

where

$$I^{update} = I^{new}_{changing-part} - I^{old}_{changing-part}. \tag{34}$$

As defined in (22), $g_{i,j}$ in gate mapping matrix represents the number of jth kind of gate in the library located in the ith cell on a chip. Therefore, we can quickly generate the new gate mapping matrix G^{new} by updating only the changing part in G^{old} as follows.

$$G^{new} = G^{old} + G^{update} \tag{35}$$

where

$$G^{update} = G^{new}_{changing-part} - G^{old}_{changing-part} \tag{36}$$

This is a sparse matrix since the changing part is much smaller compared to the size of the whole chip. Therefore, the updating process in (33) becomes

$$I^{new}_{chip} = I^{old}_{chip} + G^{update}C^{T}, \tag{37}$$

where C represent C_S for subthreshold leakage, and C_G for gate tunneling leakage. The sparsity of G^{update} makes the incremental analysis much less expansive than the full-blown chip leakage analysis.

6.5. Time Complexity Analysis

Considering statistical leakage analysis of a certain chip, for each grid cell, we need to do a weighted sum up of m kinds of gates in this cell for every coefficient in the neighbor set (size k). From (20), the number of grids in the neighbor set k is only dependent on ρ_{low} and ρ_{high}. If we choose ρ_{low} and ρ_{high} as 0.1 and 0.9, for example, k is 7. If ρ_{low} and ρ_{high} are 0.01 and 0.99, k is 19. Therefore, k is a pretty small number and is constant as long as ρ_{low} and ρ_{high} are defined by the user.

For a quadratic model with k variables, the number of coefficients is about $S \sim k^2$. So the time cost for this step is $O(k^2 \times m \times N)$, where N is the number of cells. For transferring the local coefficients to their global positions and summing them up, the time cost is $O(N)$. Next, it takes $O(N)$ to compute the full-chip leakage current. Since k and m are very small constant numbers, as a result, the time complexity of our approach becomes $O(N)$.

7. EXPERIMENTAL RESULTS

The proposed methods with and without using the look-up table have been implemented in Matlab 7.8.0. Since the leakage model for the method in Ye and Yu [2009] has to be purely log-normal (linear terms in exponent parts), we did not choose it for comparing purpose. All the experimental results are carried out in a Linux system with quad Intel Xeon CPUs with 2.99 Ghz and 16GB memory.

The methods for full-chip statistical leakage analysis were tested on circuits in the PDWorkshop91 benchmark set. The circuits were synthesized with Nangate Open Cell Library [NOCL 2012] and the placement is from MCNC [MCNC 2012], where we can get the distance information between two gates. And the distance of two grids is defined by the distance between the centers of two grids. The technology parameters come from the 45nm FreePDK Base Kit and PTM models [PTM 2012].

According to ITRS [2008], L and T_{ox} for high-performance logic in 45nm technology will be 18nm and 1.8nm, respectively. And the physical variation should be controlled within +/-12%. So the 3σ values of variations for L and T_{ox} were set to 12% of the nominal values, of which interdie variations constitute 20% and intradie variations, 80%. L is modeled as the sum of spatial correlated sources of variations, and T_{ox} is modeled as an independent source of variation. The same framework can be easily extended to include other parameters of variations. Both L and T_{ox} are modeled as Gaussian parameters. For the correlated L, the spatial correlation is modeled based

Table II. Summary of Test Cases Used in This Article

Circuit	*Gate #*	*Area/μm^2*	*Testcase*	$d_{max}/\mu m$	$d_c/\mu m$	*Grid #*
SC0	125	1459×1350	Case 1	2190	730	2×2
			Case 2	1095	365	4×4
SC1	1888	4892×4874	Case 3	1896	612	8×8
			Case 4	918	328	16×16
SC2	6417	10092×10466	Case 5	984	328	32×31
			Case 6	482	164	64×64
VLSI	2e6	SC2 × 256	Case 7	6301	2144	112×112

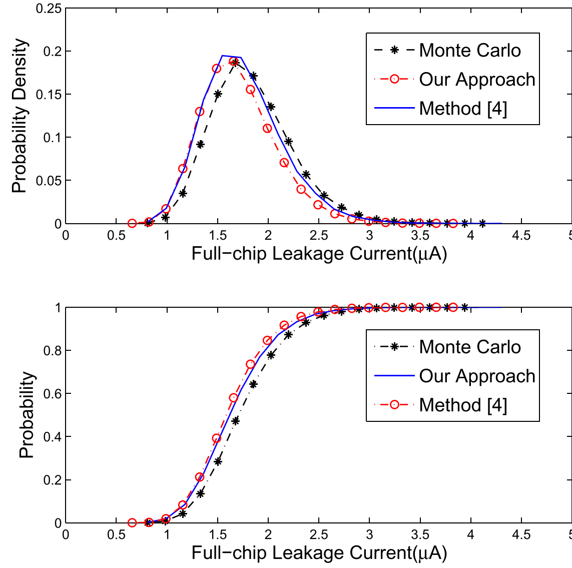

Fig. 6. Distribution of full-hip I_{leak} from MC, our approach and Hybrid grid method for Case 2.

on (2), and the partition adopts Figure 1. The test cases are given in Table II (all length units in μm), where test case "VLSI" is generated from duplicating SC2 as unit block to 16×16 array.

For comparison purposes, we performed Monte Carlo (MC) simulations with 50,000 runs using (3) and (4), the hybrid method in Chang and Sapatnekar [2007], which is called *Hybrid grid* method, and the proposed approaches on the benchmarks.

7.1. Full-Chip Leakage Distributions

Taking Case 2 for example, Figure 6 shows the full-chip leakage current distribution (PDF and CDF) of Case 2. Since the proposed methods with and without using the look-up table share the same accuracy, they are both represented by "Our approach" in Figure 6. Our method fits very well with the MC results, and has almost the same accuracy as Chang and Sapatnekar [2007].

7.2. Accuracy and CPU Time

The results of the comparison of mean value and standard deviations of full-chip leakage current are shown in Table III, where *New* is the proposed method. The average errors for mean and standard variance (σ) values of the new technique are 4.52% and

Table III. Accuracy Comparison of Different Methods Based on Monte Carlo

Test Case	*Grid #*	*Mean Value (μA)*			*Errors (%)*	
		MC	Hybrid Grid	New	Hybrid grid	New
Case1	2×2	3.311	3.105	3.169	-6.20	-4.28
Case2	4×4	3.310	3.105	3.169	-6.20	-4.28
Case3	8×8	30.04	28.88	30.46	-3.85	-1.38
Case4	16×16	30.04	28.88	30.46	-3.85	-1.38
Case5	32×32	191.6	179.0	182.7	-6.59	-4.65
Case6	64×64	191.6	179.0	182.7	-6.59	-4.65
Case7	112×112	–	–	2.6e4	–	–
Test Case	*Grid #*	*Standard Deviation (μA)*			*Errors (%)*	
		MC	Hybrid grid	New	Hybrid grid	New
Case1	2×2	0.904	0.837	0.861	-7.40	-4.69
Case2	4×4	0.594	0.547	0.548	-7.91	-7.74
Case3	8×8	5.713	5.494	5.417	-3.83	-5.18
Case4	16×16	5.307	5.400	5.067	1.75	-4.52
Case5	32×32	33.87	31.83	32.25	-6.02	-4.78
Case6	64×64	33.20	30.27	29.34	-8.83	-11.63
Case7	112×112	–	–	4.1e3	–	–

3.92%, respectively. For the Hybrid grid method Chang and Sapatnekar [2007], the average errors for mean value and σ are 4.12% and 3.83%, respectively. Table III shows these two algorithms have almost the same accuracy and our method can handle both strong and weak spatial correlations by adjusting grid size. For very large circuits such as Case7 Monte Carlo and the Hybrid grid method in Chang and Sapatnekar [2007] run out of memory but the proposed method still works.

Table IV compares the CPU times of MC, the Hybrid grid method in Chang and Sapatnekar [2007], the proposed method(*New*), and proposed method using statistical leakage characterization in SCL (shorted as LUT). This table shows the new method, *New*, is much faster than the Hybrid grid method in Chang and Sapatnekar [2007] and MC simulation. On average, the proposed algorithm has about 113X speedup over the Hybrid grid method Chang and Sapatnekar [2007] and many order of magnitudes over the MC method. And the speed of our approach is not affected by the total number of grid cells. If the spatial correlation is strong, which means d_{max} is large, d_c can be increased at the same time without loss of accuracy. So the number of neighbor grid cells in $T(i)$ will still be much smaller than the number of gates. The new method will be efficient and linear under both cases. Table IV also shows the proposed method can gain further speedup with the look-up table technique using statistical leakage characterization in SCL.

7.3. Incremental Analysis

For comparison purpose, one gate in each benchmark circuit is changed, and the proposed incremental algorithm is applied to update the leakage value locally. Table V shows the computational cost of the incremental analysis and the speedup over four different leakage analysis methods in Table IV. Compared with the look-up table approach (the fifth column in Table IV), the incremental analysis achieves $13 - 3.1e4$X speedup. As discussed in Section 6.4, the minicircuit for updating only contains a small constant number of terms. Therefore, when the problem size increases further, we

Table IV. CPU Time Comparison

Test Case	MC	Hybrid grid	New	LUT
Case1	83.14	2.96	0.10	0.0197
Case2	87.09	13.16	0.14	0.0236
Case3	828.42	26.24	0.86	0.0330
Case4	869.12	74.50	0.87	0.4559
Case5	7532.77	117.77	8.65	0.1522
Case6	7873.54	490.84	10.67	8.5462
Case7	–	–	2598	3.7313

Table V. Incremental Leakage Analysis Cost

Test	*Cost time(s)*	*Speedup over*			
Case	Incremental LUT	MC	Hybrid grid	New	LUT
Case1	3.78e-4	2.2e5	2.7e4	265	53
Case2	1.53e-4	5.7e5	8.1e4	915	157
Case3	0.0026	3.2e5	3.7e4	331	13
Case4	1.12e-4	7.8e6	6.7e5	7768	407
Case5	0.0095	7.9e5	1.1e5	911	16
Case6	2.77e-4	2.8e7	6.1e6	3.9e4	3.1e4

expect the incremental analysis could achieve more speedup over the full leakage analysis.

8. CONCLUSIONS

In this article, we have presented a new linear algorithm for full-chip statistical analysis of leakage currents in the presence of any condition of spatial correlation (strong or weak). The new algorithm adopts a set of uncorrelated virtual variables over grid cells to represent the original physical random variables with spatial correlation and the size of grid cell is determined by the correlation length. As a result, each physical variable is always represented by virtual variables in the local neighbor set. Furthermore, a look-up table is used to cache the statistical leakage information of each type of gate to avoid computing leakage for each gate instance, which builds a new characterization of the standard cell library for statistical leakage estimation. As a result, the full-chip leakage can be calculated with $O(N)$ time complexity, where N is the number of grid cells on chip. The new method maintains the linear complexity from strong to weak spatial correlation and has no limitation of leakage current model or variation model. This article also offers an incremental analysis capability to update the leakage distribution more efficiently when local changes to a circuit are made. Experimental results show the proposed method is about 1000X faster than the recently proposed method [Chang and Sapatnekar 2007] with similar accuracy and many orders of magnitude times over the Monte Carlo method on large circuit examples. The proposed incremental analysis can further achieve significant speedup over the full leakage analysis.

REFERENCES

Bhardwaj, S., Vrudhula, S., and Goel, A. 2008. A unified approach for full chip statistical timing and leakage analysis of nanoscale circuits considering intradie process variations. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst. 27,* 10, 1812–1825.

Chang, H. and Sapatnekar, S. S. 2007. Prediction of leakage power under process uncertainties. *ACM Trans. Des. Autom. Electron. Syst. 12,* 2, 1–27.

CHEN, R., ZHANG, L., ZOLOTOV, V., VISWESWARIAH, C., AND XIONG, J. 2008. Static timing: Back to our roots. In *Proceedings of the Asia South Pacific Design Automation Conference (ASPDAC)*. 310–315.

DE, V. AND BORKAR, S. 1999. Technology and design challenges for low power and high performance. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*. 163–168.

DOWDY, S. AND WEARDEN, S. 2004. *Statistics for Research*. Wiley.

DUVALL, S. G. 2000. Statistical circuit modeling and optimization. In *Proceedings of the International Workshop Statistical Metrology*. 56–63.

GHANEM, R. G. AND SPANOS, P. D. 2003. *Stochastic Finite Elements: A Spectral Approach*. Dover Publications.

HELOUE, K. R., AZIZI, N., AND NAJM, F. N. 2007. Modeling and estimation of full-chip leakage current considering within-die correlation. In *Proceedings IEEE/ACM Design Automation Conference (DAC)*. 93–98.

ITRS. 2008. International technology roadmap for semiconductors (ITRS) 2008 edition. `http://public.itrs.net`.

KIM, W., DO, K. T., AND KIM, Y. H. 2010. Statistical leakage estimation based on sequential addition of cell leakage currents. *IEEE Trans. Very Large Scale Integr. Syst. 18,* 4, 602–615.

LI, P. AND SHI, W. 2006. Model order reduction of linear networks with massive ports via frequency-dependent port packing. In *Proceedings of the Design Automation Conference (DAC)*. 267–272.

MCNC. 2012. MCNC benchmark circuit placements. `http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement/`.

MUKHOPADHYAY, S. AND ROY, K. 2003. Modeling and estimation of total leakage current in nano-scaled CMOS devices considering the effect of parameter variation. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*. 172–175.

NARENDRA, S., DE, V., BORKAR, S., ANTONIADIS, D. A., AND CHANDRAKASAN, A. P. 2004. Full-Chip subthreshold leakage power prediction and reduction techniques for sub-0.18-μm CMOS. *IEEE J. Solid-State Circ. 39,* 2.

NASSIF, S. 2000. Delay variability: Sources, impact and trends. In *Proceedings IEEE International Solid-State Circuits Conference*, 368–369.

NOCL. 2012. Nangate open cell library. `http://www.nangate.com/`.

NOVAK, E. AND RITTER, K. 1999. Simple cubature formulas with high polynomial exactness. *Construct. Approx. 15,* 4, 449–522.

OUMA, D. O., BONING, D. S., CHUNG, J. E., EASTER, W. G., SAXENA, V., MISRA, S., AND CREVASSE, A. 2002. Characterization and modeling of oxide chemical-mechanical polishing using planarization length and pattern density concepts. *IEEE Trans. Semicond. Manufact. 15,* 2, 232–244.

PTM. 2012. Predictive technology model. `http://www.eas.asu.edu/~ptm/`.

RAO, R., SRIVASTAVA, A., BLAAUW, D., AND SYLVESTER, D. 2004. Statistical analysis of subthreshold leakage current for VLSI circuits. *IEEE Trans. Very Large Scale Integr. Syst. 12,* 2, 131–139.

SHEN, R., MI, N., TAN, S. X.-D., CAI, Y., AND HONG, X. 2009. Statistical modeling and analysis of chip-level leakage power by spectral stochastic method. In *Proceedings of the Asia South Pacific Design Automation Conference (ASPDAC)*. 161–166.

TEODORESCU, R., GRESKAMP, B., NAKANO, J., SARANGI, S. R., TIWARI, A., AND TORRELLAS, J. 2007. A model of parameter variation and resulting timing errors for microarchitects. In *Workshop on Architectural Support for Gigascale Integration (ASGI)*.

WANG, J. M., SRINIVAS, B., MA, D., CHEN, C. C.-P., AND LI, J. 2005. System-level power and thermal modeling and analysis by orthogonal polynomial based response surface approach (OPRS). In *Proceedings of the International Conference on Computer Aided Design (ICCAD)*. 727–734.

XIONG, J., ZOLOTOV, V., AND HE, L. 2007. Robust extraction of spatial correlation. *IEEE Trans. Comp.-Aid. Des. Integr. Circ. Syst. 26,* 4.

YE, Z. AND YU, Z. 2009. An efficient algorithm for modeling spatially-correlated process variation in statistical full-chip leakage analysis. In *Proceedings of the International Conference on Computer Aided Design (ICCAD)*. 295–301.

Received September 2011; revised December 2011; accepted February 2012