Projection-Based Statistical Analysis of Full-Chip Leakage Power with Non-Log-Normal Distributions

Xin Li, Jiayong Le and Lawrence T. Pileggi
Department of ECE, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{xinli, jiayongl, pileggi}@ece.cmu.edu

ABSTRACT

In this paper we propose a novel projection-based algorithm to estimate the full-chip leakage power with consideration of both inter-die and intra-die process variations. Unlike many traditional approaches that rely on log-Normal approximations, the proposed algorithm applies a novel projection method to extract a low-rank *quadratic* model of the logarithm of the full-chip leakage current and, therefore, is not limited to log-Normal distributions. By exploring the underlying sparse structure of the problem, an efficient algorithm is developed to extract the non-log-Normal leakage distribution with *linear* computational complexity in circuit size. In addition, an *incremental* analysis algorithm is proposed to quickly update the leakage distribution after changes to a circuit are made. Our numerical examples in a commercial 90nm CMOS process demonstrate that the proposed algorithm provides 4x error reduction compared with the previously proposed log-Normal approximations, while achieving orders of magnitude more efficiency than a Monte Carlo analysis with 10^4 samples.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids – Verification

General Terms: Algorithms

Keywords: Statistical Analysis, Leakage Power

1. INTRODUCTION

As IC technologies move to nanoscale feature sizes, leakage power becomes increasingly large and significantly impacts the total chip power consumption. The predicted leakage power is expected to reach 50% of the total chip power within the next few technology generations [1]. Therefore, accurately modeling and analyzing leakage power has been identified as one of the top priorities for today's IC design problems.

The most important leakage components in nanoscale CMOS technologies include *sub-threshold leakage* and *gate tunneling leakage* [2]. The sub-threshold leakage models the weak inversion conduction when gate voltage is below the threshold voltage. At the same time, the reduction of gate oxide thickness facilitates tunneling of electrons through gate oxide, creating the gate leakage. Both of these leakage components are significant for sub-100nm technologies and must be considered for leakage analysis.

Unlike many other performances (e.g., delay), leakage power varies substantially due to process variations, which increases the difficulty of leakage estimation. As demonstrated in [3], leakage variations can reach 20x, while delays only vary about 30%. It has also been observed that leakage power is sensitive to both *inter-die* and *intra-die* variations. Intra-die variations model the individual, but spatially correlated, local variations within the same die. These intra-die variations must be modeled by many additional random variables, thereby significantly increasing the problem size of leakage analysis. For example, the total number of random variables can reach 10^3~10^6 to model the full-chip variations for a practical industry design.

Many works have been developed to capture the leakage variations [4]-[10]. Most of these approaches approximate the leakage variation as a log-Normal distribution. For that purpose, a first-order (i.e., linear) Taylor expansion is used to approximate the logarithm of the leakage current. Given the increasingly larger variations in nanoscale technologies, such a linear approximation can result in inaccurate results, especially because the leakage current has a strongly nonlinear dependency on process variations. As will be demonstrated by the numerical examples in Section 4, a 20% estimation error is observed by using the linear approximation for a commercial 90nm CMOS process.

To achieve higher accuracy, a quadratic approximation can be used, which, however, significantly increases the computational cost. For example, if the total number of random variables reaches 10^6, a quadratic approximation will result in a 10^6x10^6 quadratic coefficient matrix including 10^{12} coefficients!

The authors of [11] propose a projection-based approach (PROBE) to reduce the quadratic modeling cost. Instead of finding a full-rank quadratic model, PROBE attempts to find an optimal low-rank model by minimizing the approximation error. However, one major difference between leakage analysis and that addressed in [11] is the *problem size*. The PROBE algorithm is efficient for handling 10^1~10^2 random variables, while the full-chip leakage analysis involves 10^3~10^6 variables. The challenging problem here is how to make the quadratic modeling feasible for such a large problem size.

In this paper, we propose a novel projection-based algorithm to extract the optimal low-rank quadratic model for statistical leakage analysis. The proposed algorithm is facilitated by exploring the underlying *sparse* structure of the problem. Namely, the large number of intra-die variations only locally impact the leakage power in their neighborhood, as is demonstrated by many previous works, e.g., [10]. Considering this sparse property, we formulate the statistical leakage analysis problem into a special form that can be efficiently solved by the *Arnoldi algorithm* and the *orthogonal iteration* borrowed from matrix computations. As such, an accurate low-rank quadratic model can be extracted with *linear* computational complexity in circuit size.

Another important contribution of this paper is to propose a quadratic model compaction algorithm that converts a low-rank, high-dimensional quadratic leakage model to a full-rank, low-dimensional one without changing the probability distribution of the leakage. The probability density function (PDF) and the cumulative distribution function (CDF) of the low-dimensional

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

model is much easier to estimate using either Monte Carlo analysis or APEX [12].

The third contribution of this paper is to offer an incremental analysis capability to quickly update the leakage distribution after changes to a circuit are made. The proposed incremental analysis locally updates the leakage distribution and, therefore, achieves significant speedup over the full leakage analysis.

The remainder of this paper is organized as follows. In Section 2 we review the background materials and then propose our projection-based leakage analysis algorithm in Section 3. The efficacy of the proposed algorithm is demonstrated by numerical examples in Section 4. Finally, we conclude in Section 5.

2. BACKGROUND

2.1 Modeling Process Variations

Process variations are typically characterized into two broad categories: inter-die variations and intra-die variations. Inter-die variations represent the common/average variations across the die and can be modeled by using common random variables for all components in a chip. Intra-die variations represent the individual, but spatially correlated, local variations within the same die. A typical approach for modeling intra-die variations is to partition the entire die into a number of grids [10], as shown in Fig. 1. The intra-die variations in the same grid are fully correlated, while those in close (far-away) grids are strongly (weakly) correlated.

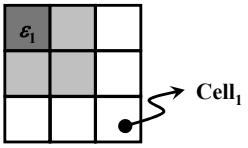

Fig. 1. Grid model for intra-die variations.

The process variations, both the inter-die and intra-die variations, are typically modeled as Normal distributions. Principal component analysis (PCA) [13] can be applied to decompose correlated Normal distributions into independent ones. After PCA, the process variations (e.g., ΔV_{TH}, ΔT_{OX} and ΔL) of each logic cell can be modeled as:

$$\Delta X_{Celli} = V_{Celli} E \,. \tag{1}$$

$\Delta X_{Celli} = [\Delta x_{Celli1}, \Delta x_{Celli2},...]^T$ denotes the parameter variations of the i-th logic cell. $E = [\varepsilon_1,\varepsilon_2,...,\varepsilon_N]^T$ stands for the random variables for modeling both inter-die and intra-die variations of the entire die. $\{\varepsilon_1,\varepsilon_2,...,\varepsilon_N\}$ are extracted by PCA. They are mutually independent and satisfy the standard Normal distribution (i.e., zero mean and unit standard deviation). N is the total number of these random variables, and it is typically large (e.g., 10^3~10^6) for practical industry designs. V_{Celli} captures the correlations among the random variables.

The size of V_{Celli} can be extremely large if there are a great number of random variables for modeling intra-die variations. However, ΔX_{Celli} only depends on the intra-die variations in its neighborhood; therefore, V_{Celli} is quite *sparse*. For example, referring to the grid in Fig. 1, ε_1 has little impact on Cell$_1$, since they are far away. In Section 3, we will show how this sparse property is utilized in our proposed leakage analysis algorithm to reduce the computational cost.

2.2 Statistical Leakage Analysis

Statistical leakage analysis typically starts from the leakage modeling for logic cells. Most previous works approximate the logarithm of the cell leakage current by a linear model:

$$\log\left(I_{Celli}\right) = B_{Celli}^T E + C_{Celli} \tag{2}$$

where I_{Celli} denotes the total leakage current (including both sub-threshold leakage and gate tunneling leakage) of the i-th cell, $B_{Celli} \in R^N$ and $C_{Celli} \in R$ are the linear model coefficients. Since the random variables $\{\varepsilon_1,\varepsilon_2,...,\varepsilon_N\}$ satisfy Normal distributions, $\log(I_{Celli})$ is the linear combination of multiple Normal distributions and, therefore, is also a Normal distribution [14]. It follows that I_{Celli} is a log-Normal distribution [14].

Given the leakage models of all individual cells, the full-chip leakage current is the sum of all cell leakage currents:

$$I_{Chip} = I_{Cell1} + I_{Cell2} + \cdots + I_{CellM} \tag{3}$$

where M is the total number of logic cells in a chip.

Eq. (3) implies that the full-chip leakage current is the sum of many log-Normal distributions. It can be approximated as a log-Normal distribution [10]. This is equivalent to approximating the logarithm of the full-chip leakage current by a linear model:

$$\log\left(I_{Chip}\right) = B_{Chip}^T E + C_{Chip} \tag{4}$$

where $B_{Chip} \in R^N$ and $C_{Chip} \in R$ are the model coefficients.

It is well-known that leakage current depends on input vector state. The cell and chip leakages in (2)-(4) can be the leakage currents for a fixed input state or the average leakage currents over all input states. In this paper, we will not distinguish these two cases.

The linear models in (2) and (4) are not sufficiently accurate for modeling the large-scale process variations that are expected for nanoscale technologies. This, in turn, suggests that applying quadratic models might be required to improve the accuracy:

$$\log\left(I_{Celli}\right) = E^T A_{Celli} E + B_{Celli}^T E + C_{Celli} \tag{5}$$

$$\log\left(I_{Chip}\right) = E^T A_{Chip} E + B_{Chip}^T E + C_{Chip} \tag{6}$$

where A_{Celli}, $A_{Chip} \in R^{N\times N}$, B_{Celli}, $B_{Chip} \in R^N$ and C_{Celli}, $C_{Chip} \in R$ are the model coefficients. In (5)-(6), the quadratic coefficient matrices A_{Celli} and A_{Chip} can be extremely large for capturing intra-die variations. This makes the quadratic modeling problem extremely challenging in practical applications.

2.3 Projection-based Modeling

The authors in [11] proposed a projection-based approach (PROBE) to reduce the quadratic modeling cost. The key difficulty of quadratic modeling is the need to compute all elements of the quadratic coefficient matrix, e.g., A_{Chip} in (6). This matrix is often rank-deficient in practical applications. Therefore, instead of finding the full-rank matrix A_{Chip}, PROBE approximates A_{Chip} by another low-rank matrix $\tilde{A}_{Chip}$ such that their difference $||A_{Chip}-\tilde{A}_{Chip}||_F$ is minimized. Here, $||\bullet||_F$ denotes the Frobenius norm, which is the square root of the sum of the squares of all matrix elements. The authors of [11] prove that the optimal rank-R approximation is:

$$\tilde{A}_{Chip} = \sum_{r=1}^{R} \lambda_{Chipr} P_{Chipr} P_{Chipr}^T \tag{7}$$

where $\lambda_{Chipr} \in R$ and $P_{Cellr} \in R^N$ are the r-th dominant eigenvalue and eigenvector of the matrix A_{Chip} respectively.

The PROBE algorithm proposed in [11] is efficient to handle 10^1~10^2 random variables, while the full-chip leakage analysis involves 10^3~10^6 variables. The challenging problem here is how to extract the dominant eigenvalues and eigenvectors for a 10^6x10^6 matrix.

3. PROJECTION-BASED ANALYSIS

We propose a projection-based analysis algorithm that is facilitated by exploring the underlying sparse structure of the leakage analysis problem. Specifically, we propose the following

methodology, which includes: 1) a two-step iterative algorithm for quadratic leakage modeling; 2) a quadratic model compaction algorithm for leakage distribution estimation; and 3) an incremental analysis algorithm for locally updating the leakage distribution.

3.1 Standard Cell Library Characterization

Our statistical leakage analysis starts from the standard cell library characterization where the objective is to approximate the leakage current of each logic cell by a regression model. Typically, there are only a few (e.g., 5~10) random variables for modeling the variations in one cell. Therefore, we can run SPICE simulations (or utilize measurement models if available) and apply PROBE [11] to fit the rank-K model for each cell:

$$\log(I_{Celli}) = \sum_{k=1}^{K} \lambda_{Cellik} \cdot \left(\tilde{P}_{Cellik}^T \cdot \Delta X_{Celli}\right)^2 + \tilde{B}_{Celli}^T \cdot \Delta X_{Celli} + C_{Celli} \quad (8)$$

where ΔX_{Celli} is defined in (1), and λ_{Cellik}, $\tilde{P}_{Cellik}$, $\tilde{B}_{Celli}$ and C_{Celli} are the model coefficients. Substituting (1) into (8) yields:

$$\log(I_{Celli}) = \sum_{k=1}^{K} \lambda_{Cellik} \cdot \left(P_{Cellik}^T E\right)^2 + B_{Celli}^T E + C_{Celli} \quad (9)$$

$$P_{Cellik} = V_{Celli}^T \tilde{P}_{Cellik} \qquad B_{Celli} = V_{Celli}^T \tilde{B}_{Celli} \quad (10)$$

where $P_{Cellik} \in R^N$, $B_{Celli} \in R^N$, and N is the total number of random variables for the entire die. The sizes of P_{Cellik} and B_{Celli} in (9) are much larger than the sizes of $\tilde{P}_{Cellik}$ and $\tilde{B}_{Celli}$ in (8). However, as discussed in Section 2.1, V_{Celli} is sparse. Therefore, both P_{Cellik} and B_{Celli} are *sparse* and contain many zeros.

For simplifying the notation, we define the following symbols to represent all cell leakage models in a matrix form:

$$\begin{aligned} \log(I_{Cell}) &= \left[\log(I_{Cell1}) \quad \log(I_{Cell2}) \quad \cdots \quad \log(I_{CellM})\right]^T \\ \Lambda_{Cellk} &= \left[\lambda_{Cell1k} \quad \lambda_{Cell2k} \quad \cdots \quad \lambda_{CellMk}\right]^T \\ P_{Cellk} &= \left[P_{Cell1k} \quad P_{Cell2k} \quad \cdots \quad P_{CellMk}\right] \\ B_{Cell} &= \left[B_{Cell1} \quad B_{Cell2} \quad \cdots \quad B_{CellM}\right] \\ C_{Cell} &= \left[C_{Cell1} \quad C_{Cell2} \quad \cdots \quad C_{CellM}\right]^T \end{aligned} \quad . \quad (11)$$

Comparing (11) with (9), it is easy to verify that:

$$\log(I_{Cell}) = \sum_{k=1}^{K} \Lambda_{Cellk} \otimes \left(P_{Cellk}^T E\right) \otimes \left(P_{Cellk}^T E\right) + B_{Cell}^T E + C_{Cell} \quad (12)$$

where $\otimes$ stands for the point-wise multiplication, i.e., $[a_1,a_2,...]^T \otimes [b_1,b_2,...]^T = [a_1b_1,a_2b_2,...]^T$.

3.2 Full-Chip Leakage Modeling

We next develop the algorithm to efficiently extract the low-rank quadratic model of the full-chip leakage current. As shown in (3), the full-chip leakage current is the sum of all cell leakage currents. Applying the log transform to both sides of (3) yields:

$$\log(I_{Chip}) = \log\left[e^{\log(I_{Cell1})} + e^{\log(I_{Cell2})} + \cdots + e^{\log(I_{CellM})}\right]. \quad (13)$$

Substituting (12) into (13) and applying a second order Taylor expansion, after some mathematical manipulations we obtain a quadratic model in the form of (6), where the model coefficients are given by:

$$C_{Chip} = \log\left(\frac{1}{\alpha}\right) \quad (14)$$

$$B_{Chip} = \alpha \cdot B_{Cell} \cdot \Phi \quad (15)$$

$$\begin{aligned} A_{Chip} = \alpha \cdot \sum_{k=1}^{K} P_{Cellk} \cdot \mathrm{diag}\left(\Phi \otimes \Lambda_{Cellk}\right) \cdot P_{Cellk}^T \\ + \frac{\alpha}{2} \cdot B_{Cell} \cdot \mathrm{diag}(\Phi) \cdot B_{Cell}^T - \frac{\alpha^2}{2} \cdot B_{Cell} \cdot \Phi\Phi^T \cdot B_{Cell}^T \end{aligned} \quad . \quad (16)$$

In (14)-(16), $\mathrm{diag}([a_1,a_2,...]^T)$ stands for the diagonal matrix with the elements $\{a_1,a_2,...\}$ and:

$$\alpha = \frac{1}{e^{C_{Cell1}} + e^{C_{Cell2}} + \cdots + e^{C_{CellM}}} \quad (17)$$

$$\Phi = \left[e^{C_{Cell1}} \quad e^{C_{Cell2}} \quad \cdots \quad e^{C_{CellM}}\right]^T. \quad (18)$$

The values of α and Φ in (17)-(18) can be computed with linear computational complexity. After α and Φ are known, the model coefficients C_{Chip} and B_{Chip} can be evaluated from (14)-(15). Because the matrix B_{Cell} in (15) is sparse, computing the matrix-vector product $B_{Cell}\Phi$ has linear computational complexity. Therefore, both C_{Chip} in (14) and B_{Chip} in (15) can be extracted with linear complexity.

The major difficulty, however, stems from the *non-sparse* quadratic coefficient matrix A_{Chip} in (16). This non-sparse feature can be understood from the last term at the right-hand side of (16). The vector Φ is dense and, therefore, $\Phi\Phi^T$ is a dense matrix. It follows that $B_{Cell}\Phi\Phi^T{B_{Cell}}^T$ is dense, although B_{Cell} is sparse. For this reason, it would be extremely expensive to explicitly construct the quadratic coefficient matrix A_{Chip} based on (16).

To overcome this problem, we propose a novel iterative algorithm that consists of two steps: Krylov subspace generation and orthogonal iteration. Instead of finding the full matrix A_{Chip}, the proposed algorithm attempts to find the optimal low-rank approximation of A_{Chip}.

A. Krylov Subspace Generation

As shown in (7), the optimal rank-R approximation of A_{Chip} is determined by the dominant eigenvalues $\{\lambda_{Chip1},\lambda_{Chip2},...,\lambda_{ChipR}\}$ and eigenvectors $\{P_{Chip1},P_{Chip2},...,P_{ChipR}\}$. The subspace generated by all linear combinations of these dominant eigenvectors is called the *dominant invariant subspace* [15] and is denoted as:

$$span\left\{P_{Chip1}, P_{Chip2}, \cdots, P_{ChipR}\right\}. \quad (19)$$

It is well-known that the dominant invariant subspace in (19) can be approximated by the following *Krylov subspace* [15]:

$$span\left\{Q_0, A_{Chip}Q_0, A_{Chip}^2Q_0, \cdots, A_{Chip}^{R-1}Q_0\right\} \quad (20)$$

where $Q_0 \in R^N$ is a non-zero vector that is not orthogonal to any dominant eigenvectors. We first develop the algorithm to extract the Krylov subspace which gives a good approximation of the dominant invariant subspace. The extracted Krylov subspace is then used as a starting point for the orthogonal iteration in Section 3.2.B such that the orthogonal iteration could converge to the dominant invariant subspace within a few iteration steps.

We adapt the Arnoldi algorithm from matrix computations [15] to generate the Krylov subspace. The Arnoldi algorithm has been applied to large-scale numerical problems and its numerical stability has been well-demonstrated for many applications, most notably, IC interconnect order reduction [16]. Fig. 2 summarizes a simplified implementation of the Arnoldi algorithm.

Step 3 in Fig. 2 is the key step of the Arnoldi algorithm. It computes the matrix-vector product $Q_r = A_{Chip}Q_{r\text{-}1}$. Since the matrix A_{Chip} is large and dense, Eq. (21) does *not* construct the matrix A_{Chip} explicitly. Instead, it computes $A_{Chip}Q_{r\text{-}1}$ *implicitly*, i.e., multiplying all terms in (16) by $Q_{r\text{-}1}$ separately and then adding them together. It is easy to verify that A_{Chip} in (16) is the sum of the products of many sparse or low-rank matrices.

Therefore, the implicit matrix-vector product in (21) can be computed with linear computational complexity. Taking the last term in (21) as an example, there are four steps to compute $B_{Cell}\Phi\Phi^T B_{Cell}{}^T Q_{r\text{-}1}$, including: 1) $S_1 = B_{Cell}{}^T Q_{r\text{-}1}$ (sparse matrix multiplied by a vector); 2) $S_2 = \Phi^T S_1$ (dot product of two vectors); 3) $S_3 = \Phi S_2$ (vector multiplied by a scalar); and 4) $S_4 = B_{Cell} S_3$ (sparse matrix multiplied by a vector). All these four steps have linear computational complexity.

1. Randomly select an initial vector $Q_0 \in R^N$.
2. $Q_1 = Q_0 / \|Q_0\|_F$.
   For $r = 2,3,...,R$

$$Q_r = \alpha \cdot \sum_{k=1}^{K} P_{Cellk} \cdot \mathrm{diag}\left(\Phi \otimes \Lambda_{Cellk}\right) \cdot P_{Cellk}^T \cdot Q_{r-1}$$

3. $$+\frac{\alpha}{2} \cdot B_{Cell} \cdot \mathrm{diag}\left(\Phi\right) \cdot B_{Cell}^T \cdot Q_{r-1} \quad . \qquad (21)$$

$$-\frac{\alpha^2}{2} \cdot B_{Cell} \cdot \Phi\Phi^T \cdot B_{Cell}^T \cdot Q_{r-1}$$

4. Orthogonalize Q_r to all Q_i ($i = 1,2,...,r\text{-}1$).
5. $Q_r = Q_r / \|Q_r\|_F$.
   End For
6. $$Q = \left[Q_R \quad \cdots \quad Q_2 \quad Q_1\right]. \qquad (22)$$

Fig. 2. Simplified Arnoldi algorithm.

B. Orthogonal Iteration

1. Start from the matrix $Q \in R^{N\times R}$ in (22).
2. $Q^{(1)} = Q$, where the superscript stands for the iteration index.
   For $i = 2,3,...$

$$Z^{(i)} = \alpha \cdot \sum_{k=1}^{K} P_{Cellk} \cdot \mathrm{diag}\left(\Phi \otimes \Lambda_{Cellk}\right) \cdot P_{Cellk}^T \cdot Q^{(i-1)}$$

3. $$+\frac{\alpha}{2} \cdot B_{Cell} \cdot \mathrm{diag}\left(\Phi\right) \cdot B_{Cell}^T \cdot Q^{(i-1)} \quad . \qquad (23)$$

$$-\frac{\alpha^2}{2} \cdot B_{Cell} \cdot \Phi\Phi^T \cdot B_{Cell}^T \cdot Q^{(i-1)}$$

4. $Q^{(i)}U^{(i)} = Z^{(i)}$ (QR factorization).
   End For
5. $$Q_{Chip} = Q^{(i)} \quad U_{Chip} = U^{(i)} \, . \qquad (24)$$

Fig. 3. Simplified orthogonal iteration algorithm.

The Krylov subspace computed from Fig. 2 is not exactly equal to the dominant invariant subspace. Starting from the matrix Q in (22), we further apply an orthogonal iteration [15] which exactly converges to the dominant invariant subspace. Theoretically, the orthogonal iteration can start from any matrix. However, since the Krylov subspace Q gives a good approximation of the dominant invariant subspace, using Q as the starting point helps the orthogonal iteration to reach convergence within a few iteration steps.

Fig. 3 shows a simplified implementation of the orthogonal iteration algorithm. In (23), $Q^{(i-1)} \in R^{N\times R}$ is a matrix containing only a few columns, because R is typically small (e.g., around 10) in most applications. Therefore, similar to (21), $Z^{(i)}$ in (23) can be computed with linear complexity. For the same reason, the QR factorization in Step 4 of Fig. 3 also has linear computational complexity, since $Z^{(i)} \in R^{N\times R}$ contains only a few columns.

The orthogonal iteration in Fig. 3 is provably convergent if the columns in the initial matrix Q are not orthogonal to the dominant invariance subspace [15]. After the orthogonal iteration converges, the optimal rank-R approximation of A_{Chip} is determined by Q_{Chip} and U_{Chip} in (24) [15]:

$$\tilde{A}_{Chip} = Q_{Chip} U_{Chip} Q_{Chip}^T \, . \qquad (25)$$

Combining (25) with (6) yields:

$$\log\left(I_{Chip}\right) = E^T \cdot \left(Q_{Chip} U_{Chip} Q_{Chip}^T\right) \cdot E + B_{Chip}^T E + C_{Chip} \qquad (26)$$

where C_{Chip} and B_{Chip} are given in (14)-(15).

The algorithms in Fig. 2 and Fig. 3 assume a given approximation rank R. In practice, the value of R can be iteratively determined based on the approximation error. For example, starting from a low-rank approximation, R should be iteratively increased if the modeling error remains large. In most cases, we find that selecting R in the range of 5~15 provides sufficient accuracy.

In summary, we developed a two-step iterative algorithm to extract the low-rank quadratic model of the full-chip leakage current. The proposed algorithm only involves simple vector operations and sparse matrix-vector multiplications; therefore, its computational complexity is *linear* in circuit size. In addition, it is not necessary to explicitly construct the matrix $\tilde{A}_{Chip}$ in (25). In the next subsection we will develop an algorithm that efficiently estimates the leakage current distribution.

3.3 Leakage Distribution Estimation

The quadratic function in (26) is N-dimensional, where N is typically large. It is not easy to estimate the leakage distribution directly from (26). Next we propose a quadratic model compaction algorithm that converts the high-dimensional model to a low-dimensional one, while keeping the leakage distribution unchanged.

1. Start from the quadratic model in (26).
2. $Q_{Comp}[U_{Comp}\ B_{Comp}] = [Q_{Chip}\ B_{Chip}]$ (QR factorization).
3. $$\Omega = Q_{Comp}^T E \, . \qquad (27)$$
4. $$\log\left(I_{Chip}\right) = \Omega^T \cdot \left(U_{Comp} U_{Chip} U_{Comp}^T\right) \cdot \Omega + B_{Comp}^T \Omega + C_{Chip} \, . \qquad (28)$$

Fig. 4. Quadratic model compaction algorithm.

Fig. 4 summarizes the proposed quadratic model compaction algorithm. The following two theorems prove that the quadratic models in (26) and (28) are equivalent and the random variables Ω defined in (27) are independent and satisfy the standard Normal distribution.

Theorem 1: The quadratic models in (26) and (28) are equivalent.
Proof: The QR factorization in Step 2 of Fig. 4 results in:

$$Q_{Chip} = Q_{Comp} U_{Comp} \quad B_{Chip} = Q_{Comp} B_{Comp} \, . \qquad (29)$$

Substituting (29) and (27) into (26) yields:

$$\begin{aligned} \log\left(I_{Chip}\right) &= E^T Q_{Comp} U_{Comp} U_{Chip} U_{Comp}^T Q_{Comp}^T E \\ &\quad + B_{Comp}^T Q_{Comp}^T E + C_{Chip} \\ &= \Omega^T \cdot \left(U_{Comp} U_{Chip} U_{Comp}^T\right) \cdot \Omega + B_{Comp}^T \Omega + C_{Chip} \end{aligned} \quad . \qquad (30)$$

Eq. (30) proves Theorem 1. ■

Theorem 2: Given a set of independent random variables E that satisfy the standard Normal distribution, the random variables Ω defined in (27) are independent and satisfy the standard Normal distribution.
Proof: Since the random variables Ω are the linear combinations of zero-mean Normal distributions, they are also zero-mean Normal distributions. The correlation matrix for Ω is given by:

$$\overline{\Omega \cdot \Omega^T} = \overline{Q_{Comp}^T E E^T Q_{Comp}} = Q_{Comp}^T \cdot \overline{EE^T} \cdot Q_{Comp} \qquad (31)$$

where $\bar{a}$ stands for the expected value of the random variable a. Remember that the random variables E are independent with the standard Normal distribution, i.e.:

$$\overline{EE^T} = I \tag{32}$$

where I is the identity matrix. In addition, the matrix Q_{Comp} is constructed from QR factorization and, therefore, is orthogonal:

$$Q_{Comp}^T Q_{Comp} = I\,. \tag{33}$$

Substituting (32)-(33) into (31) yields:

$$\overline{\Omega \cdot \Omega^T} = Q_{Comp}^T \cdot I \cdot Q_{Comp} = I\,. \tag{34}$$

Eq. (34) implies that the random variables Ω are uncorrelated. Uncorrelated Normal distributions are also independent [14]. ■

The quadratic function in (28) has a dimension of R+1 which is much smaller than N. Based on (28), the PDF/CDF of $\log(I_{Chip})$ can be extracted, for example, using either Monte Carlo analysis or APEX [12]. After that, the PDF/CDF of I_{Chip} can be easily computed by a simple nonlinear transform [14].

3.4 Incremental Leakage Analysis

Incremental leakage analysis facilitates a quick update on the leakage distribution after local changes to a circuit are made. For simplicity, we only detailedly discuss the case where one logic cell is changed. However, it should be noted that the proposed algorithm can be directly extended to handle the simultaneous change of multiple cells.

Assume that the i-th logic cell is changed (e.g., a low V_{TH} cell is replaced by a high V_{TH} cell to reduce leakage), resulting in:

$$I_{Chip}^{New} = I_{Chip}^{Old} - I_{Celli}^{Old} + I_{Celli}^{New} \tag{35}$$

where $I_{Chip}{}^{Old}$ ($I_{Chip}{}^{New}$) and $I_{Celli}{}^{Old}$ ($I_{Celli}{}^{New}$) respectively denote the leakage currents of the entire chip and the i-th cell before (after) the change.

Given the low-rank quadratic models of $\log(I_{Chip}{}^{Old})$, $\log(I_{Celli}{}^{Old})$ and $\log(I_{Celli}{}^{New})$, the objective of incremental leakage analysis is to quickly generate the low-rank model for $\log(I_{Chip}{}^{New})$. Compared with (3), Eq. (35) is much simpler and only contains a few terms. Therefore, updating the leakage distribution using (35) is much cheaper than the full leakage analysis from (3). The aforementioned iterative algorithm and compaction algorithm can be directly applied to (35). More details on the incremental analysis are not included in this paper due to the limited number of available pages.

4. NUMERICAL EXAMPLES

We demonstrate the efficacy of the proposed algorithm using ISCAS'85 benchmark circuits. All circuits are implemented in a commercial 90nm CMOS process. The numerical experiments in this section are performed on a SUN — 1GHz server.

4.1 Standard Cell Library Characterization

SPICE simulations were used to construct an approximate model of the logarithm of the cell leakage current as a function of the parameter variations: ΔV_{THP}, ΔV_{THN}, ΔT_{OX} , ΔW and ΔL. The distributions and correlations of these parameter variations are provided in the CMOS process design kit. Both linear and quadratic regression models are created for accuracy comparison. The low-rank quadratic models are extracted using the PROBE algorithm in [11].

Fig. 5 shows the approximation errors for three different logic cells. As the quadratic model is used, the approximation error is significantly reduced, e.g., dropping from 14% to 2% for NOR2.

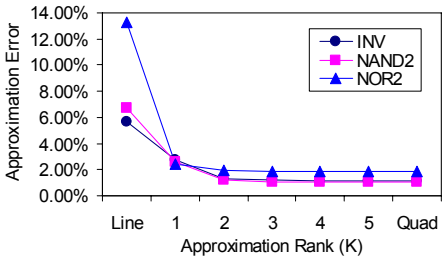

Fig. 5. Regression modeling error for cell leakage current.

In addition, a rank-2 quadratic model, instead of the full-rank model with rank 5, is sufficiently accurate in this example. It is also worth mentioning that similar error patterns are observed for other logic cells, although only three of them are shown in Fig. 5.

4.2 Statistical Leakage Analysis

Both inter-die and intra-die variations are considered for statistical leakage analysis. The grid model discussed in Section 2.1 (Fig. 1) is used to capture intra-die variations. We partition the circuit into extremely fine grids and one grid only contains one logic cell, thereby significantly increasing the problem size. Such a large problem size helps us to verify the efficiency and robustness of the proposed projection-based algorithm. Table 1 shows the number of random variables (i.e., the size of the vector E in (1)) for each benchmark circuit. Note that the problem size is greater than 17K for C7552.

Table 1. Number of random variables for ISCAS'85 circuits

Name	RV #	Name	RV #	Name	RV #
C17	35	C1355	2735	C5315	11540
C432	805	C1908	4405	C6288	12085
C499	1015	C2670	5970	C7552	17565
C880	1920	C3540	8350		

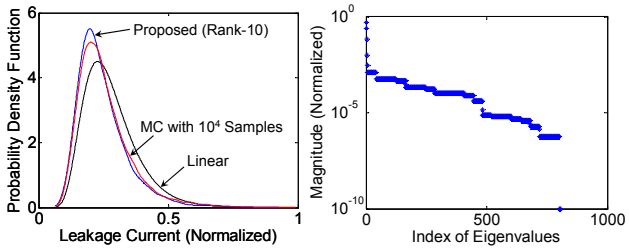

Fig. 6. Leakage analysis results for C432. (a) Probability density function. (b) Eigenvalue distribution.

A. Probability Density Function

Taking the circuit C432 as an example, Fig. 6(a) shows the leakage distributions extracted by three different approaches: the linear approximation, the rank-10 quadratic approximation and the Monte Carlo analysis with 10^4 samples. As shown in Fig. 6(a), the linear approximation yields large errors, especially at both tails of the PDF which are often the points of greatest concern.

B. Eigenvalue Distribution

For testing and comparison, we extract the full-rank quadratic leakage model for C432. Fig. 6(b) shows the magnitude of the eigenvalues of the quadratic coefficient matrix. Note that there are only a few dominant eigenvalues. Fig. 6(b) explains why the low-rank quadratic approximation is efficient in this example.

C. Accuracy and Speed

Table 2 and Table 3 compare the leakage analysis accuracy and cost for different modeling approaches. The worst-case leakage is measured at the 99% point on CDF. The error values in Table 2 are calculated against the Monte Carlo simulation with 10^4 samples. As shown in Table 2, the proposed low-rank quadratic approximation significantly reduces the maximal error from 21.01% to 5.37%. Using the full-rank quadratic approximation can modestly reduce the error further, however resulting in extremely expensive cost, as shown in Table 3. The proposed low-rank approximation achieves up to 10^5x speedup over the full-rank approximation and up to 10^3x speedup over the Monte Carlo analysis with 10^4 samples.

Table 2. Estimation error for worst-case leakage

Name	Linear	Proposed (Rank-10)	Full Quadratic
C17	12.74%	0.83%	0.26%
C432	14.93%	4.18%	3.17%
C499	20.64%	5.37%	4.29%
C880	15.23%	4.59%	3.57%
C1355	9.24%	1.33%	0.45%
C1908	8.91%	1.04%	—
C2670	10.85%	2.16%	—
C3540	11.44%	2.51%	—
C5315	11.85%	2.85%	—
C6288	21.01%	3.18%	—
C7552	11.13%	2.40%	—

Table 3. Computational cost for leakage analysis (Sec.)

Name	Proposed (Rank-10)	Full Quadratic	Monte Carlo (10^4 Samples)
C17	0.05	0.11	12.56
C432	0.07	1112.17	106.22
C499	0.08	3241.52	132.80
C880	0.11	17619.80	277.10
C1355	0.15	72026.90	413.49
C1908	0.22	—	756.65
C2670	0.33	—	1136.66
C3540	0.47	—	1890.86
C5315	0.67	—	2922.90
C6288	0.73	—	3128.96
C7552	1.14	—	6030.64

D. Incremental Analysis

For comparison, we change one gate in each benchmark circuit and apply the proposed incremental analysis algorithm to locally update the leakage value. Table 4 shows the computational cost of the incremental analysis. Compared with the second column in Table 3, the incremental analysis achieves up to 10x speedup. We expect that as the problem size increases further, the incremental analysis could achieve more speedup over the full leakage analysis.

Table 4. Incremental leakage analysis cost (Sec.)

Name	Time	Name	Time	Name	Time
C17	0.06	C1355	0.07	C5315	0.09
C432	0.06	C1908	0.08	C6288	0.09
C499	0.07	C2670	0.08	C7552	0.13
C880	0.07	C3540	0.09		

5. CONCLUSIONS

We have proposed a projection-based algorithm to capture the non-log-Normal leakage distributions that are expected in nanoscale technologies. The proposed algorithm has linear computational complexity in circuit size, which is facilitated by several novel algorithms, including: 1) a two-step iterative quadratic modeling algorithm; 2) a quadratic model compaction algorithm; and 3) an incremental analysis algorithm. Our numerical examples in a commercial 90nm CMOS process demonstrate that, compared with the popular log-Normal approximation, the proposed leakage analysis reduces the error from 21.01% to 5.37%, while achieving up to 10^3x speedup over the Monte Carlo analysis with 10^4 samples.

6. REFERENCES

[1] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2004.

[2] K. Roy, S. Mukhopadhyay and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. IEEE*, vol. 91, no. 2, pp. 305-327, Feb. 2003.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, Parameter variations and impact on circuits and microarchitecture," *IEEE DAC*, pp. 338-342, 2003.

[4] S. Narendra, V. De, S. Borkar, D. Antoniadis and A. Chandrakasan, "Full-chip sub-threshold leakage power prediction model for sub-0.18 μm CMOS," *IEEE ISLPED*, pp. 19-23, 2002.

[5] R. Rao, A. Srivastava, D. Blaauw and D. Sylvester, "Statistical estimation of leakage current considering inter- and intra-die process variation," *IEEE ISLPED*, pp. 84-89, 2003.

[6] R. Rao, A. Devgan, D. Blaauw and D. Sylvester, "Parametric yield estimation considering leakage variability," *IEEE DAC*, pp. 442-447, 2004.

[7] A. Srivastava, S. Shah, K. Agarwal, D. Sylvester, D. Blaauw and S. Director, "Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance," *IEEE DAC*, pp. 535-540, 2005.

[8] S. Mukhopadhyay, A. Raychowdhury and K. Roy, "Accurate estimation of total leakage current in scaled CMOS logic circuits based on compact current modeling," *IEEE DAC*, pp. 169-174, 2003.

[9] S. Mukhopadhyay, A. Raychowdhury and K. Roy, "Accurate estimation of total leakage in nanometer-scale bulk CMOS circuits based on device geometry and doping profile," *IEEE Trans. CAD*, vol. 24, no. 3, pp. 363-381, Mar. 2005.

[10] H. Chang, S. Sapatnekar, "Full-chip analysis of leakage power under process variations, including spatial correlations," *IEEE DAC*, pp. 523-528, 2005.

[11] X. Li, J. Le, L. Pileggi and A. Strojwas, "Projection-based performance modeling for inter/intra-die variations," *IEEE ICCAD*, pp. 721-727, 2005.

[12] X Li, J. Le, P. Gopalakrishnan and L. Pileggi, "Asymptotic probability extraction for non-Normal distributions of circuit performance," *IEEE ICCAD*, pp. 2-9, 2004.

[13] G. Seber, *Multivariate Observations*, John Wiley & Sons, 1984.

[14] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 2001.

[15] G. Golub and C. Loan, *Matrix Computations*, The Johns Hopkins Univ. Press, 1996.

[16] A. Odabasioglu, M. Celik and L. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm," IEEE Trans. CAD, vol. 17, no. 8, pp. 645-654, Aug. 1998.