Variation-Aware Leakage Power Model Extraction for System-Level Hierarchical Power Analysis

Yang Xu*, Bing Li†, Ralph Hasholzner*, Bernhard Rohfleisch*, Christian Haubelt‡, Jürgen Teich‡
*Intel Mobile Communications, Munich, Germany
†Technische Universitaet Muenchen, Munich, Germany
‡University of Erlangen-Nuremberg, Erlangen, Germany
Email: {yang.a.xu, ralph.hasholzner, bernhard.rohfleisch}@intel.com*
b.li@tum.de† {haubelt, teich}@informatik.uni-erlangen.de‡

***Abstract*—System-level power analysis is commonly used in modern SoC design processes to evaluate power consumption at early design phases. With the increasing variations in manufacturing, the statistical characteristics of parameters are also incorporated in the state-of-the-art methods. However, the spatial correlation between modules still remains as a challenge for system-level statistical power analysis where power models generated from individual modules are used for analysis efficiency or IP protection. In this paper, we propose a novel method to extract variation-aware and correlation-inclusive leakage power models for fast and accurate system-level analysis. For each individual module we generate a power model with different correlation information specified by the module vendor or customer. The local random variables in the power models are replaced by the corresponding ones at system level to reconstruct the correlation between modules so that the accuracy of system-level analysis is guaranteed. Experimental results show that our method are very accurate while being 1000X faster than Monte Carlo simulation and 70X−100X faster than the flattened full chip statistical leakage analysis.**

I. Introduction

In modern system-on-chip (SoC) design methodologies, power consumption, together with other design constraints, e.g., performance and die size, is usually defined at very early design phases. Respecting these design constraints, the design space is explored to choose appropriate system architectures. Making correct design decisions at such early design phases is very important to avoid significant modification efforts and cost in later phases. Therefore, early and accurate system power consumption analyses are required to guarantee that all power consumption constraints are met. Additionally, owing to the increasing complexity of modern SoCs, the design space becomes very huge. Thus, fast power analysis methods are mandatory to permit efficient design space exploration.

To achieve high efficiency during design space exploration, system-level hierarchical power analysis methods are applied. For example, spread-sheet [1] and power-state based methods [2] [3] are often used to explore the power design space at early design phases. In these methods, the power consumption of each component is modeled by parameterized equations or annotated power values, which rely on constant process parameters or nominal/worst-case power values. As long as the process variations are small, such methods can provide useful power estimation at the early design phases. However, with the technology scaling down to nanometer regime, process variations become significant and non-negligible, thus making conventional power modeling methods inaccurate and face new challenges: Firstly, the power models with constant process parameters become inaccurate due to increasing process variations. Secondly, variations also cause huge deviations from nominal power dissipation values for the same hardware block, e.g., up to a 20 times variation in chip leakage power has been found in a 180 nm technology processor [4]. Consequently, accurate power evaluation results cannot be produced by simply annotating power states with nominal values. Finally, worst case analysis makes the power evaluation too pessimistic, which results in overengineering and high design cost.

Statistical power analysis methods have been introduced to handle the impact of process variations on power consumption. Because leakage power is highly sensitive to process variations while dynamic power is relatively immune to them, most of these methods focus on statistical leakage analysis (SLA) [5] [6] [7]. SLA has originally been proposed to analyze full chip leakage power considering process variations. Therefore, it cannot be directly integrated into a hierarchical power analysis environment. Although SLA has been applied to individual hardware components to extract statistical leakage power information for rough system-level power analysis [8], the correlations between modules cannot be easily incorporated at system level by using SLA due to lack of hierarchical analysis method. As we will show in this paper, neglecting these correlations during system-level power analysis may cause significant inaccuracy. Additionally, in modern SoC design processes, because of IP protection reasons, netlists of IP modules, which are mandatory inputs for SLA, are not always provided by the IP vendors, thus restricting the applicability of SLA. Furthermore, SLA usually deals with gate-level calculations, which may lead to runtime problem when applied to complex SoCs. All these facts make system-level hierarchical power analysis even more challenging.

The main contribution of this paper is a novel method

978-3-9810801-8-6/DATE12/ © 2012 EDAA

to extract variation-aware and correlation-inclusive leakage power models for system-level hierarchical power analysis. Within this method, SLA is firstly used to generate statistical leakage power models for system modules. Thereafter, a hierarchical statistical leakage analysis (HSLA) algorithm is introduced to replace the local independent random variables (R.V.s) in the generated statistical models with a new set of independent R.V.s at top level so that the correlations between modules can be taken into account during system-level power analysis. With this method on the one hand, IP vendors can provide hierarchical statistical power models to customers without disclosing netlists. Hence, intellectual property is protected. On the other hand, the customer can use these power models and the proposed HSLA method to perform a fast and accurate system-level power evaluation. Experimental results show that this method can produce very accurate leakage power distribution curves compared with Monte Carlo simulation. Furthermore, it is 1000X faster than Monte Carlo simulation and 70X-100X faster than full chip SLA. To the best of our knowledge, this paper is the first that deals with hierarchical statistical leakage power model extraction and analysis.

The rest of this paper is organized as follows: In Section II, related work will be reviewed. In Section III, we will describe an existing SLA method, which will be used as the basic engine for statistical leakage power analysis. The hierarchical statistical leakage model extraction and analysis will be detailed in Section IV. Thereafter, we will present experimental results in Section V and conclude the paper in Section VI.

II. Related Work

In this section, we will review related work in statistical leakage analysis. Many previous methods have been proposed to analyze leakage power under process variations. In [9] [10], the mean and variance of full chip subthreshold leakage currents are evaluated by analytic methods. Only global variations are considered in these two methods. In [11], both global and local variations are considered and the probability density function (PDF) of a full chip subthreshold leakage is derived. The effect of spatial correlations in local variations is first considered in [5] by partitioning a chip into grids and assuming perfect correlations among the devices in the same grid, but with a high computation complexity, $O(n^2)$, where n is the number of gates in the chip. To reduce this complexity, a spectral stochastic method is proposed in [7] where principle component analysis (PCA) is used to reduce the number of variables. In [6], PCA is applied to transform the spatially correlated R.V.s into linear form so that the computation of mean and variance of a lognormal random variable can be computed in $O(N_p)$ where N_p is the number of principle components. Recently, a linear algorithm for full chip leakage power analysis considering weak spatial correlation is proposed in [12]. The above methods can be used to analyze leakage power of full chips and individual modules statistically. But none of them considers the challenge in hierarchical statistical leakage analysis, i.e., how to incorporate the correlations between modules when they are instantiated at system level. In this paper, we will propose a method to solve this problem.

III. Statistical Leakage Power Modeling and Analysis

In this section, we will describe an existing SLA algorithm used in this paper. By using this algorithm, statistical leakage power, which contains local spatial correlation information for a single module, can be calculated. In this paper, we use the terms leakage power and leakage current interchangeably, since the difference between them is just a multiplicative factor V_{dd}.

Similar to [5] [6], leakage power is modeled in lognormal distribution form

$$I_l = e^{f(p)} \tag{1}$$

where $f(p)$ is a function of process parameters. Each process parameter is expressed as

$$p = p_0 + p_g + p_l \tag{2}$$

where p_0 is the nominal value of the parameter; p_g stands for the global variation shared by all gates while the local variation is modeled by p_l that is specific for each gate and is correlated to each other. It is assumed that p_l only models spatial correlation and does not contain any effects from global variation. For simplicity, two parameters are considered in this work, namely the transistor gate length (L) and the gate oxide thickness (T_{ox}). Nevertheless the proposed method can be easily extended to take other process parameters into account. As in [5] [7], L is taken as a spatially correlated parameter and T_{ox} as spatially uncorrelated. In addition, it is also assumed that correlation exists only among the same type of parameters and there is no correlation between different parameters, e.g., between L and T_{ox}. To model the local spatial correlation, the die area of each module is partitioned into $nrow \times ncol = n$ grids. All gates within the same grid are assumed to have the same local variation, represented by a random variable p_{l_i}, $i \in \{1, 2, ..n\}$. All n p_{l_i} form a vector $\mathbf{P}_l$ whose correlation matrix $\mathbf{C}$ can be obtained by calculating correlations between p_{l_i} and p_{l_j}, $j \in \{1, 2, ..n\}$ using an empirical formulation. In this paper we use the following exponential model [13]

$$\gamma(r) = e^{-r^2/\eta^2} \tag{3}$$

where r is the distance between the centers of two grids and η is the correlation length, though the proposed method is not bound to any specific correlation matrix.

A method similar to the one proposed in [6] is used to calculate the sum of leakage power. To simplify the mean and variance calculations, the local correlated random variable vector $\mathbf{P}_l$=$[p_{l_1}, p_{l_2}, ..., p_{l_n}]^T$ is decomposed by applying PCA [14]

$$\mathbf{P}_l = \mathbf{A}\mathbf{x} \tag{4}$$

where the transformation matrix $\mathbf{A}$, constructed by eigen-vectors of $\mathbf{C}$, is orthogonal. $\mathbf{x}$=$[x_1, x_2, ..., x_n]^T$ is a set of independent Gaussian R.V.s with mean being 0.

When represented in the form of (4), each random variable p_{l_i} can be expressed as a linear combination of $x_1, x_2, ..., x_n$ with the coefficients coming from the corresponding row of $\mathbf{A}$. Combining with (1) and the fact that $f(p)$ can be approximated by a first-order Taylor expansion at the nominal values of process parameters [5], the leakage power can be expressed as

$$I_l = e^{b_0 + b_{g_L} L_g + b_{g_{tox}} T_{oxg} + \sum_{i=1}^{n} b_i x_i} \quad (5)$$

where b_0 is the nominal value of $f(p)$. L_g and T_{oxg} are normalized global variations of L and T_{ox}, which are shared by all gates in hierarchical leakage analysis. $\sum_{i=1}^{n} b_i x_i$ is the local variation of L in linear form where x_i are independent variables in (4). Since T_{ox} is assumed to be spatially uncorrelated, there is no local variation portion for T_{ox}. b_{g_L}, $b_{g_{tox}}$ and b_i are all coefficients with fixed values. The total leakage power can be calculated recursively and expressed in the same form as (5) [6]. In each recursive step, $I_l^a + I_l^b$ is approximated by

$$\begin{aligned} I_l^c = & \; e^{c_0 + c_{g_L} L_g + c_{g_{tox}} T_{oxg} + \sum_{i=1}^{n} c_i x_i} \\ = & \; e^{c_0 + \sum_{i=1}^{n+2} c_i x_i} \end{aligned} \quad (6)$$

and c_0, c_i can be calculated by

$$\begin{aligned} c_0 = & \; \frac{1}{2} \log \left((E(I_l^a) + E(I_l^b))^4 \right) \\ & -\frac{1}{2} \log \left((E(I_l^a) + E(I_l^b))^2 + Var(I_l^a) + Var(I_l^b) \right. \\ & \left. + 2Cov(I_l^a I_l^b) \right) \end{aligned} \quad (7)$$

$$c_i = \log \left(\frac{E(I_l^a e^{x_i}) + E(I_l^b e^{x_i})}{(E(I_l^a) + E(I_l^b)) E(e^{x_i})} \right) \quad (8)$$

where $E(*)$, $Var(*)$, and $Cov(*)$ stand for mean, variance and covariance operations, respectively.

IV. Hierarchical Statistical Leakage Power Models

Hierarchical analysis methods are commonly applied during SoC design process to overcome increasing design complexities. SLA, which is usually used to analyze leakage power for full chips or individual modules, cannot guarantee the accuracy during hierarchical analysis due to the lack of information to incorporate correlations between modules at higher levels. In this section, we will first introduce two simple and so less accurate hierarchical statistical leakage power models that can be used in hierarchical design analysis. After that, a method that can extract very accurate hierarchical statistical leakage power models will be detailed.

A. Simple Hierarchical Leakage Model Extraction

The direct outputs of SLA are mean and variance of leakage power for each module, which are calculated by the following formulas:

$$E(I_l) = e^{\mu + \frac{1}{2}\sigma^2} \quad (9)$$

$$Var(I_l) = e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2} \quad (10)$$

where μ and σ^2 are mean and variance of the exponent of I_l in (5), respectively. The simplest hierarchical leakage power model can be extracted by simply ignoring all correlations and providing $E(I_l)$ and $Var(I_l)$ as model content for module j, i.e.,

$$LPM_{NoCorr.}^j = \{E(I_l^j),\, Var(I_l^j)\} \quad (11)$$

The leakage power of the entire system is calculated by summing up the leakage power of all modules

$$\mu_{total} = \sum_{j=1}^{m} E(I_l^j),\; \sigma_{total}^2 = \sum_{j=1}^{m} Var(I_l^j) \quad (12)$$

where m is the number of modules in the system. This method assumes no correlation between modules.

When the modules are instantiated at system level, leakage power from different modules are correlated with each other due to system-level spatial correlations as well as global correlations. Ignoring all correlations can generate very large inaccuracy. Therefore, we propose a simple yet efficient method that only considers global correlations to improve the accuracy. In order to take global correlations into account at system level, coefficients b_0, b_{g_L} and $b_{g_{tox}}$ in (5) are provided as model content, i.e.,

$$LPM_{GlobalOnly}^j = \{b_0^j,\, b_{g_L}^j,\, b_{g_{tox}}^j\} \quad (13)$$

so that leakage power of each module can be expressed in the form of (14).

$$I_l = e^{b_0 + b_{g_L} L_g + b_{g_{tox}} T_{oxg}} \quad (14)$$

Then the recursive calculation algorithm introduced in Section III can be applied again to calculate total leakage power.

B. Accurate Hierarchical Leakage Model Extraction

SLA can be easily extended to incorporate global correlations at system-level design as what we have shown above, because the R.V.s L_g and T_{oxg} in (5) that represent global variations are shared by all the gates in the system. It is, however, very hard for the SLA to incorporate spatial correlation at system level, since the R.V.s x_i in (5) that model the spatial correlation are generated from the correlation matrix $\mathbf{C}$ in Section III, which is different from module to module. Therefore, no spatial correlation can be established between modules by simple variable sharing.

The similar problem also exists in system-level statistical timing analysis. To solve this, a characterization method for hierarchical statistical timing model is first introduced in [15]. In this method, spatially correlated R.V.s in timing models are transformed into linear form by applying PCA. Then, when these modules are instantiated at system level, these variables are mapped back to their original variables first and later replaced by a new set of independent R.V.s. Thus, the correlation between modules can be established by sharing the same set of R.V.s. Nevertheless, this method has its own limitation: If the transformation matrix is not selected carefully, the reverse transformation may not be possible at

all. This limitation is resolved in [16] by selecting a complete eigenvector matrix as the transformation matrix.

In this paper, we will use a similar approach to solve the HSLA problem. This approach replaces the independent R.V.s in the leakage power models with a new set of independent R.V.s from system level so that the correlations between modules can be established by sharing the same set of variables.

After each module is analyzed by the method in Section III, statistical leakage power in the form of (5) can be generated. Usually, this step is done by IP vendors. However, if netlists are available, it can also be done by the customer himself. Besides b_0, b_{g_L}, $b_{g_{tox}}$ and $b_i x_i$, the grid size gs used in partitioning the module die area, module length ml and module width mw as well as the matrix **A** in (4) are provided to customers, too.

$$LPM_H^j = \{b_0^j,\, b_{g_L}^j,\, b_{g_{tox}}^j,\, b_i^j x_i^j,\, gs^j,\, ml^j,\, mw^j,\, \mathbf{A}^j\} \quad (15)$$

After the hierarchical leakage power model LPM_H^j of each IP_j is obtained, the system leakage power can be analyzed by applying HSLA. During HSLA, the customers first instantiate the obtained IP modules at system level and partition the area covered by the IP modules in the same way they are partitioned during SLA, i.e., using the provided grid size gs^j to partition the corresponding IP area. Then, the same grid size is used to partition the remaining system die area. Fig. 1, adapted from [16], shows an example of system die partitioning where two IP modules X and Y are instantiated. Thereafter, each grid is assigned a random variable $p_{l_i}^{sys}$ to represent the local variation at system level no matter if it is a normal grid or an irregular grid, e.g., the shadowed grid in Fig. 1 as an irregular grid. Since the size of such an irregular grid is smaller than a normal grid, it will not lose any modeling accuracy. If there are totally m grids after system die partitioning, the random variable vector $\mathbf{P}_l^{sys} = [p_{l_1}^{sys}, p_{l_2}^{sys}, ..., p_{l_m}^{sys}]^T$ has an $m \times m$ correlation matrix $\mathbf{C}^{sys}$. Similar to (4), it can be decomposed as

$$\mathbf{P}_l^{sys} = \mathbf{B}\mathbf{x}^{sys} \approx \mathbf{B}^k \mathbf{x}^{sys,k} \quad (16)$$

where **B** is formed by eigenvectors of $\mathbf{C}^{sys}$. $\mathbf{x}^{sys}$=$[x_1^{sys}, x_2^{sys}, \cdots, x_m^{sys}]$ are independent variables at system level with zero mean. Their standard deviations are formed by square root of eigenvalues of $\mathbf{C}^{sys}$ corresponding to eigenvectors in **B**. If there are eigenvalues that are very small compared with other larger eigenvalues, the corresponding variables in $\mathbf{x}^{sys}$ contribute relatively less than other variables. Therefore, these less contributing variables can be discarded to reduce the number of independent variables so that the run time can be improved. Assume $\mathbf{x}^{sys}$ is truncated to $\mathbf{x}^{sys,k}$ with k variables, $k < m$ and $\mathbf{B}^k$ is a truncated matrix of **B**. After decomposition, each random variable $p_{l_i}^{sys}$ can be expressed as a linear combination of $x_1^{sys}, x_2^{sys}, \cdots, x_k^{sys}$:

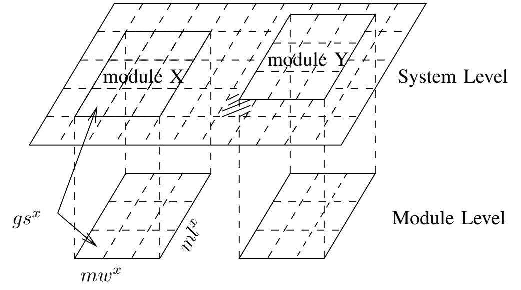

Fig. 1. System die partitioning

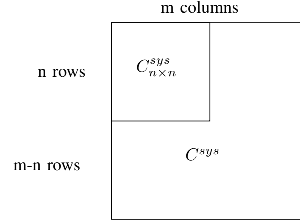

Fig. 2. Submatrix $\mathbf{C}_{n\times n}^{sys}$

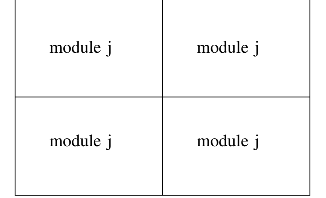

Fig. 3. Experimental circuit system

$$\begin{aligned} p_{l_1}^{sys} &= \beta_{11}x_1^{sys} + \beta_{12}x_2^{sys} + ... + \beta_{1k}x_k^{sys} \\ p_{l_2}^{sys} &= \beta_{21}x_1^{sys} + \beta_{22}x_2^{sys} + ... + \beta_{2k}x_k^{sys} \\ &\vdots \\ p_{l_m}^{sys} &= \beta_{m1}x_1^{sys} + \beta_{m2}x_2^{sys} + ... + \beta_{mk}x_k^{sys} \end{aligned} \quad (17)$$

where the coefficients $\beta_{i1}, \beta_{i2}, \cdots, \beta_{ik}$ $i \in \{1, 2, ..m\}$ come from the corresponding row i of $\mathbf{B}^k$.

In the following, we will take module Y as an example to illustrate the variable replacement. Since at system level, module Y is partitioned with the same grid size as the one used during SLA at module level, the area covered by module Y is still partitioned into n grids. Assuming the n R.V.s associated with the grids covered by module Y at system level, denoted as $\mathbf{P}_{l,n}^{sys}$, corresponding to the first n R.V.s in $\mathbf{P}_l^{sys}$, the correlation between $\mathbf{P}_{l,n}^{sys}$ can be represented by the $n \times n$ sub-matrix $\mathbf{C}_{n\times n}^{sys}$ at the upper-left corner of $\mathbf{C}^{sys}$ as shown in Fig. 2. Since the correlation matrix is determined by the distance between grids and module Y is partitioned in the same way it has been partitioned during SLA, this $\mathbf{C}_{n\times n}^{sys}$ sub-matrix is the same as the **C** of $\mathbf{P}_l$ during SLA in Section III, i.e., **C**=$\mathbf{C}_{n\times n}^{sys}$. Because both $\mathbf{P}_{l,n}^{sys}$ and $\mathbf{P}_l$ are standard Gaussian random variable vectors and they have the same correlation matrix, $\mathbf{P}_{l,n}^{sys}$ and $\mathbf{P}_l$ are equivalent.

$$\mathbf{P}_l = \mathbf{P}_{l,n}^{sys} \quad (18)$$

As $\mathbf{P}_{l,n}^{sys}$ correspond to the first n R.V.s in $\mathbf{P}_l^{sys}$, it can be written in PCA form as

$$\mathbf{P}_{l,n}^{sys} = \mathbf{B}_n^k \mathbf{x}^{sys,k} \quad (19)$$

where $\mathbf{B}_n^k$, a $n \times k$ matrix, is constructed by the first n rows of $\mathbf{B}^k$ in (16). Comparing (19) and (4), we find that in (19), $\mathbf{P}_{l,n}^{sys}$ is decomposed into a combination of k independent

R.V.s while in (4) $\mathbf{P}_l$ is decomposed into a combination of n independent R.V.s. These k R.V.s carry the information of correlations from other modules at system level. Therefore, to incorporate the correlations at system level, the old random variable vector $\mathbf{x}$ is replaced with a new set of independent R.V.s from system level. From (18) (19) and (4), the replacement is performed as

$$\mathbf{x} = \mathbf{A}^T \mathbf{P}_l = \mathbf{A}^T \mathbf{B}_n^k \mathbf{x}^{sys,k} \tag{20}$$

where $\mathbf{A}^T = \mathbf{A}^{-1}$ since $\mathbf{A}$ is orthogonal. By applying (20) to each SLA generated leakage power model in the form of (5), the leakage power model of each module becomes

$$I_l = e^{b_0 + b_{g_L} L_g + b_{g_{tox}} T_{oxg} + \sum_{i=1}^{k} d_i x_i^{sys}} \tag{21}$$

where x_i^{sys} are new independent R.V.s at system level and d_i are their corresponding coefficients, which are calculated by:

$$\mathbf{d} = \mathbf{b}\mathbf{A}^T \mathbf{B}_n^k \tag{22}$$

where $\mathbf{d}=[d_1, d_2, \cdots, d_k]$, $\mathbf{b}=[b_1, b_2, \cdots, b_n]$. Consequently, the correlation between modules is modeled by sharing the new set of independent R.V.s $\mathbf{x}^{sys,k}$. When all leakage power models are transformed into the form of (21), the leakage power of the entire system can be calculated by the recursive method introduced in Section III.

TABLE I lists the complete procedure of accurate HSLA using random variable replacement at system level.

TABLE I
PROCEDURE OF ACCURATE HSLA

	Steps	Executor
1.	analyze IP modules with SLA and generate local leakage power models.	IP vendor or customer
2.	instantiate IP modules at system level and partition system die with grids	customer
3.	decompose system-level correlated process parameters using PCA.	customer
4.	replace local independent variables for each module using (20).	customer
5.	system-level leakage power analysis.	customer

C. Computational Complexity

To analyze the computational complexity, typically the pre-characterization cost of step 1 and IP instantiation cost of step 2 in TABLE I are not taken into account. The cost of PCA in step 3 is $O(\tau N_g^3)$ where τ is the number of spatially correlated process parameters and N_g is the number of grids that the system die has been partitioned into. The cost of step 4 equals the cost of a matrix multiplication (22). Since $\mathbf{b}\mathbf{A}^T$ can be provided by the IP vendor or pre-calculated, the cost approximates to $O(\tau n_g k N_m)$ where k is the number of column of $\mathbf{B}_n^k$ and n_g is the number of grids a module is partitioned into and N_m is the number of modules in the system. Finally, the cost of step 5 is $O(N_m N_g)$. As the system-level process parameter decomposition with PCA is a common operation for system-level statistical analyses, e.g, hierarchical statistical timing analysis [16], it is possible to share its result and there is no need to calculate it for every analysis. Therefore, the overall complexity is $O(\tau n_g k N_m)$, which means our method scales linearly with the number of spatial correlated parameters.

V. Experimental Results

In this section, the results of applying our proposed HSLA method to the ISCAS89 benchmarks [17] are shown. The proposed method has been implemented in C++ and all the experiments were executed on a Linux machine with 2 GB memory and a 3.0 GHz CPU.

All benchmark circuits were synthesized with a 45nm library. The 3σ values of parameter variations for L and T_{ox} were set to 12% of the nominal parameter values, like in [7]. The proportions of inter-die variations and intra-die variations were set to 30% and 70%, respectively.

To test the proposed HSLA method, a series of experimental hierarchical circuit systems were built by placing four identical modules close to each other as shown in Fig. 3. In order to verify the accuracy of the proposed accurate model, we compared its results with the ones carried out by running Monte Carlo simulation with 10000 iterations. The input of Monte Carlo simulation is a set of flattened netlists of the experimental systems. To show the effectiveness of the HSLA method, we also compared the results with those produced by the two simple methods introduced in Section IV that ignore all correlations and consider global correlation only.

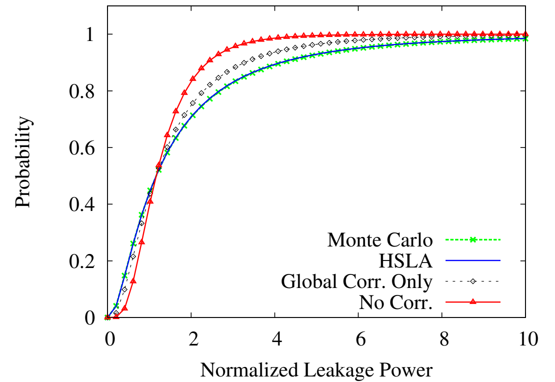

Fig. 4. CDF of leakage power of s641 based experimental system

Fig. 4 shows the normalized leakage power distribution (CDF) of an experimental system with four s641 modules. For comparison, the curves generated by the two simple methods are also illustrated in the same figures, showing that the curve generated by our HSLA method tracks the Monte Carlo curve very well and the ones generated from the simple methods deviate significantly from the Monte Carlo curve. When these methods are used to predict parametric yield for leakage, the simple methods trend to predict over-optimistic yields, e.g., at normalized leakage 3, both simple methods produce a much higher yield than Monte Carlo simulation. Therefore, we can conclude that the correlation from local variations and global variations has a remarkable effect on the system leakage

TABLE II
COMPARISON OF EXPERIMENTAL RESULTS

Circuits	No Corr.		Global Corr. Only		HSLA				Monte Carlo	SLA	
	μ err. %	σ err. %	μ err. %	σ err. %	μ err. %	σ err. %	Exe. T (s)	Exe. T0 (s)	Exe. T (s)	Exe. T (s)	Exe. T0 (s)
s298	-0.003	-43.11	-0.133	-20.15	0.002	-1.42	$< 1\mu s$	$< 1\mu s$	0.78	$< 1\mu s$	$< 1\mu s$
s420	-0.006	-43.58	-0.128	-18.78	-0.006	-0.58	$< 1\mu s$	$< 1\mu s$	1.17	$< 1\mu s$	$< 1\mu s$
s526	-0.003	-43.78	-0.118	-18.22	-0.003	-0.41	$< 1\mu s$	$< 1\mu s$	1.29	$< 1\mu s$	$< 1\mu s$
s641	-0.006	-43.35	-0.130	-18.33	-0.006	0.10	$< 1\mu s$	$< 1\mu s$	1.38	$< 1\mu s$	$< 1\mu s$
s713	-0.005	-43.30	-0.129	-18.21	-0.005	0.21	$< 1\mu s$	$< 1\mu s$	1.43	$< 1\mu s$	$< 1\mu s$
s820	-0.001	-41.24	-0.102	-13.30	0	1.29	$< 1\mu s$	$< 1\mu s$	2.12	0.01	0.01
s953	-0.003	-40.62	-0.105	-12.36	0.003	0.03	$< 1\mu s$	$< 1\mu s$	4.01	0.02	0.02
s1196	-0.013	-43.11	-0.095	-12.08	-0.013	-1.05	$< 1\mu s$	$< 1\mu s$	4.47	0.04	0.04
s1238	-0.016	-41.46	-0.096	-9.24	-0.016	0.60	$< 1\mu s$	$< 1\mu s$	5.56	0.04	0.04
s1423	-0.021	-41.53	-0.110	-9.82	-0.021	0.46	$< 1\mu s$	$< 1\mu s$	6.30	0.05	0.05
s5378	-0.032	-43.05	-0.092	-5.76	-0.032	0.13	0.01	$< 1\mu s$	27,37	0.23	0.22
s9234	-0.018	-45.10	-0.057	-3.72	-0.018	0.25	0.01	$< 1\mu s$	61.01	0.51	0.50
s13207	-0.007	-45.95	-0.037	-2.28	-0.007	0.69	0.05	0.01	143.80	1.25	1.21
s15850	-0.009	-46.41	-0.035	-2.03	-0.009	0.53	0.06	0.01	177.13	1.56	1.51
s38584	-0.013	-48.76	-0.022	-0.72	-0.013	0.11	1.35	0.17	1524.58	14.76	13.58
average	0.01	-43.59	-0.092	-10.99	-0.010	0.53	$average = \sum \lvert err._i \rvert / 15$				

power and ignoring the correlation effects during system-level leakage analysis can produce unacceptable inaccuracy.

The results of other experimental systems are shown in TABLE II, from which we can see that mean values are close for all three methods; ignoring all correlations causes the largest standard deviation error, around 43%; only considering global correlation can also generate large error, e.g., up to 20% compared with Monte Carlo simulation; in contrast, our proposed method with both correlations considered can produce very accurate results, i.e., with average error of $\sigma < 1\%$.

To evaluate the efficiency of the proposed method, we also compared its execution time with the Monte Carlo simulation as well as the full chip SLA described in Section III whose input is a set of flattened netlists of the experimental systems. In TABLE II, the execution time is shown as Exe. T and Exe. T0, representing execution time with and without system-level PCA calculation because the PCA cost can be eliminated by sharing PCA results with other statistical analysis. For most benchmarks, the execution time of our method is less than 1 μs. From the comparison, we can conclude that our method is at least 1000X faster than Monte Carlo simulation and when compared with full chip SLA it also has at least 10X-20X speedup, which makes it a suitable method for system-level hierarchical power analysis. It also shows that this speedup can be further enhanced to at least 70X-100X by utilizing shared system-level PCA results.

VI. CONCLUSIONS

In this paper, we proposed a new method to extract variation-aware and correlation-inclusive leakage power models for fast and accurate system power analysis. This method utilizes hierarchical leakage power analysis where random variables in SLA generated leakage power models are replaced by a new set of independent random variables so that correlations between modules at system level can be incorporated. With this method, the system leakage power analysis can produce very accurate results compared with Monte Carlo simulation. Furthermore, it is faster than Monte Carlo simulation by three orders of magnitude. When compared with full chip SLA, our method still has a 70X-100X speedup advantage.

ACKNOWLEDGEMENT

This work was supported in part by the Project PowerEval (funded by Bayerisches Wirtschafsministerium, support code IUK314/001).

REFERENCES

[1] D. Lidsky and J. M. Rabaey, "Early Power Exploration - A World Wide Web Application," in *DAC*, 1996, pp. 27–32.

[2] L. Benini, R. Hodgson, and P. Siegel, "System-Level Power Estimation and Optimization," in *ISLPED*, 1998, pp. 173–178.

[3] R. A. Bergamaschi and Y. W. Jiang, "State-Based Power Analysis for Systems-on-Chip," in *DAC*, 2003, pp. 638–641.

[4] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," *IEEE Micro*, vol. 25, pp. 10–16, November 2005.

[5] H. Chang and S. S. Sapatnekar, "Full-Chip Analysis of Leakage Power Under Process Variations, Including Spatial Correlations," in *DAC*, 2005, pp. 523–528.

[6] 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," in *DAC*, 2005, pp. 535–540.

[7] R. Shen, N. Mi, S. X.-D. Tan, Y. Cai, and X. Hong, "Statistical Modeling and Analysis of Chip-Level Leakage Power by Spectral Stochastic Method," in *ASP-DAC*, 2009, pp. 161–166.

[8] S. Chandra, K. Lahiri, A. Raghunathan, and S. Dey, "Considering Process Variations During System-Level Power Analysis," in *ISLPED*, 2006, pp. 342–345.

[9] 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," in *ISLPED*, 2002, pp. 19–23.

[10] A. Srivastava, R. Bai, D. Blaauw, and D. Sylvester, "Modeling and Analysis of Leakage Power Considering Within-Die Process Variations," in *ISLPED*, 2002, pp. 64–67.

[11] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical Estimation of Leakage Current Considering Inter- and Intra-Die Process Variation," in *ISLPED*, 2003, pp. 84–89.

[12] R. Shen, S. X.-D. Tan, and J. Xiong, "A Linear Algorithm for Full-Chip Statistical Leakage Power Analysis Considering Weak Spatial Correlation," in *DAC*, 2010, pp. 481–486.

[13] J. Xiong, V. Zolotov, and L. He, "Robust Extraction of Spatial Correlation," in *ISPD*, 2006, pp. 2–9.

[14] I. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[15] A. Goel, S. Vrudhula, F. Taraporevala, and P. Ghanta, "A Methodology for Characterization of Large Macro Cells and IP Blocks Considering Process Variations," in *ISQED*, 2008, pp. 200–206.

[16] B. Li, N. Chen, M. Schmidt, W. Schneider, and U. Schlichtmann, "On Hierarchical Statistical Static Timing Analysis," in *DATE*, 2009, pp. 1320–1325.

[17] F. Brglez, D. Bryan, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," in *ISCAS*, 1989, pp. 1929–1934.