ARTICLE IN PRESS

INTEGRATION, the VLSI journal ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Statistical timing and power analysis of VLSI considering non-linear dependence

Lerong Cheng [a], Wenyao Xu [b,*], Fengbo Ren [a], Fang Gong [a], Puneet Gupta [a], Lei He [a]

[a] Electrical Engineering Department, University of California, Los Angeles, CA 90095, United States
[b] Computer Science and Engineering, State University of New York at Buffalo, NY 14260, United States

ARTICLE INFO

Article history:
Received 15 January 2012
Received in revised form
29 December 2013
Accepted 29 December 2013

Keywords:
Statistical modeling
VLSI
Yield analysis

ABSTRACT

Majority of practical multivariate statistical analysis and optimizations model interdependence among random variables in terms of the linear correlation. Though linear correlation is simple to use and evaluate, in several cases non-linear dependence between random variables may be too strong to ignore. In this paper, we propose polynomial correlation coefficients as simple measure of multi-variable non-linear dependence and show that the need for modeling non-linear dependence strongly depends on the end function that is to be evaluated from the random variables. Then, we calculate the errors in estimation resulting from assuming independence of components generated by linear de-correlation techniques, such as PCA and ICA. The experimental results show that the error predicted by our method is within 1% error compared to the real simulation of statistical timing and leakage analysis. In order to deal with non-linear dependence, we further develop a target-function-driven component analysis algorithm (FCA) to minimize the error caused by ignoring high order dependence. We apply *FCA* to statistical leakage power analysis and SRAM cell noise margin variation analysis. Experimental results show that the proposed FCA method is more accurate compared to the traditional PCA or ICA.

Published by Elsevier B.V.

1. Introduction

With the CMOS technology scaling down to the nanometer regime, process as well as operating variations have become a major limiting factor for integrated circuit design. These variations introduce significant uncertainty for both circuit performance and leakage power. Statistical analysis and optimization, therefore, has generated lot of interest in the VLSI design community.

Existing work has studied statistical analysis and optimization for timing [1–5], power [6–9], and spatial correction extraction [10]. Most of these papers assume independence between random variables when performing statistical analysis. In order to obtain independence, most existing works use linear transformations, such as principal component analysis (PCA) or independent component analysis (ICA), to de-correlate the data. However, when there is non-linear dependence between the random variables under consideration, both PCA and ICA cannot completely remove the dependence between random variables. PCA can only remove linear correlation[1] between random variables but cannot remove the high order dependence. On the other hand, ICA tries to minimize the mutual information between the random variables.[2] However being a linear operation, ICA often cannot completely remove the dependence between random variables.

In practice, the dependence between different variation sources is rarely linear (e.g., V_{th} is exponentially related to L_{eff}). Therefore, ignoring such non-linear dependencies can cause significant error in analyses. There are some existing techniques for handling arbitrary dependence, such as Copula [11] and total correlation [12]. However, the complexity of using Copula is exponentially related to the number of random variables. Mutual information [12] and total correlation [12] measure the dependence between random variables, however, it is not easy to apply them in the statistical analysis. Moreover, there is little work in *removing* dependence using such measures as is readily done using PCA for linear correlation.

There exists some nonlinear algorithms to decompose nonlinear dependent variation sources to independent components, such as nonlinear PCA [13] (or Kernel PCA) and nonlinear ICA [14]. Applying such algorithms may completely (or almost completely) remove

* Corresponding author.
E-mail address: wenyaoxu@buffalo.edu (W. Xu).

[1] Two random variables X_1 and X_2 are uncorrelated if and only if $E[X_1 \cdot X_2] = E[X_1] \cdot E[X_2]$. The linear correlation measures how likely one random variable may increase when the other one increases.

[2] The mutual information between two random variables X_1 and X_2, $I(X_1, X_2)$, are defined as $I(X_1, X_2) = \iint_{-\infty}^{\infty} f_{12}(x_1, x_2) \cdot \log\ (f_{12}(x_1, x_2)/f_1(x_1) \cdot f_2(x_2))\ dx_1\ dx_2$, where $f_1(x_1)$ and $f_2(x_2)$ are the marginal probability density function (PDF) of X_1 and X_2, respectively and $f_{12}(x_1, x_2)$ are the joint PDF of X_1 and X_2. $I(X_1, X_2)$ measures the dependence between X_1 and X_2, $I(X_1, X_2) = 0$ if and only if X_1 and X_2 are independent.

http://dx.doi.org/10.1016/j.vlsi.2013.12.004
0167-9260 Published by Elsevier B.V.

dependence between variation sources and results in independent components. However, such algorithms either express the variation sources as a very complicate function of independent components or do not give close form expressions to express variation source using independent components. Hence, such nonlinear transformations are not easy to use in statistical analysis and optimization.

Compared to the previous work [15], we analyzed the impact of nonlinear dependence on statistical analysis and evaluated the performance of the algorithm with more experiments in this paper. In sum, key contributions of this work are as follows:

- We propose *polynomial correlation coefficients* as a simple measure of non-linear dependence among random variables.
- We show that the importance of modeling non-linear dependence strongly depends on what is to be done with the random variables, i.e., the end function of random variables that is to be estimated.
- We develop closed form expressions to calculate error in the estimation of arbitrary moments (e.g., mean, variance, skewness) of the to-be estimated function as a result of assuming true independence of components generated by PCA or ICA techniques.
- We develop a target function driven component analysis algorithm (we refer to as *FCA*) which minimizes the error caused by ignoring non-linear dependence without increasing the computational complexity of statistical analysis.

The methods developed in this paper can be used to check whether linear de-correlation techniques like PCA will suffice for particular analysis problem. To the best of our knowledge, this is the first work to propose a systematic method to evaluate the need for complex non-linear dependence modeling for statistical analysis in VLSI design or otherwise. We apply our error estimation formula to the typical examples from computer aided VLSI design: statistical timing and leakage analysis. Experimental result shows that our estimation is within 1% error of simulation. Further we give two example applications of *FCA* algorithm: statistical leakage analysis and SRAM cell noise margin variation analysis. The experimental results show that the FCA is more accurate than regular PCA or ICA.

The rest of the paper is organized as follows: Section 3 theoretically calculates the impact of high order correlation, Section 4 applies the formula to statistical timing and leakage analysis and presents some experimental results, finally Section 5 presents the target function driven ICA algorithm to minimize the error caused by ignoring non-linear dependence and Section 6 concludes this paper.

2. Motivation and preliminaries

In this section, we show the limitations of using PCA and ICA to obtain independent random variables and propose the *polynomial correlation* measure.

PCA can only remove linear correlation between random variables but cannot remove the high order dependence. Independent random variables must be uncorrelated, but uncorrelated random variables are not necessarily independent. If we assume that the uncorrelated random variables are independent (as is done by most VLSI statistical analysis techniques), errors in the statistical calculations can be significantly large. Consider the following simple example. Let S_1 and S_2 be two independent random variables with standard normal distributions. Let $X_1 = S_1 + S_2, X_2 = S_1 S_2$. It is easy to find that X_1 and X_2 are uncorrelated, but certainly not independent. Let $f(X_1, X_2) = X_1^2 + X_1 X_2 + X_1^2 X_2^2 + X_2^2$. We can see that in order to compute the mean of $f(\cdot)$, not only the linear correlation but also the 4th order joint moments between X_1 and X_2 should be considered. Theoretically, the mean of f should be $E[f] = 9$. However, if we ignore the dependence between X_1 and X_2 and assume that they are independent, then we would calculate the mean of f as $E[f] = 5$. From the above example, we can see that ignoring high order dependence can cause large error even when computing the mean. Moreover, notice that in the above example, we know that variation source is a function of independent random variables S_1 and S_2 (i.e., we know the mixing function). However, in many real applications [16–19], this assumption does not hold true, which makes the problem of higher order dependence difficult to handle. ICA tries to minimize the mutual information between the random variables.

When $I(X_1, X_2)$ exists, X_1 and X_2 are independent if and only if $I(X_1, X_2) = 0$. Since it is still a linear operation, it cannot completely remove the dependence between random variables. Let us observe another simple example: let S_1 and S_2 be two independent random variables with standard normal distribution and $X_1 = S_1 + S_2$, $X_2 = S_1 S_2$. Then there will be no linear operations to decompose X_1 and X_2 to independent random variables.

3. Analysis of impact of nonlinear dependence

As discussed above, commonly used PCA and ICA techniques cannot provide fully independent random variable decomposition. In this section, we are going to study the impact of non-linear dependence on statistical analysis. We define the *ij*th order polynomial correlation coefficient between two random variables X_1 and X_2 as

$$\rho_{ij} = \frac{E[X_1^i X_2^j] - E[X_1^i]E[X_2^j]}{\sqrt{E[(X_1^i - E[X_1^i])^2] \cdot E[(X_2^j - E[X_2^j])^2]}}. \quad (1)$$

ρ_{ij}'s provide us with simple and good measures to estimate the impact of nonlinear dependence. Note that $-1 \le \rho_{ij} \le 1$ and that ρ_{11} is simply the linear correlation coefficient. In rest of this paper, we assume that the ρ_{ij}'s are known. In practice, ρ_{ij} can be computed from the samples of variation sources.

With the above definition, we will show how to evaluate the impact of non-linear dependence on statistical analysis. Let us consider the two random variable case first. Let f be a polynomial function (or Taylor expansion of an arbitrary function) of two random variables $\boldsymbol{X} = (X_1, X_2)^T$:

$$f(X) = \sum_{ij} a_{ij} X_1^i X_2^j. \quad (2)$$

Then

$$E[f(\boldsymbol{X})] = \sum_{ij} a_{ij} m_{ij}, \quad (3)$$

where $m_{ij} = E[X_1^i \cdot X_2^j]$ is the *ij*th joint moment of X_1 and X_2. If we ignore m_{ij}, then the error of mean estimation will be $a_{i,j}(m_{i,j} - m_{i,0} m_{0,j})$. That is, the importance of the *ij*th joint moment depends on the coefficient of the *ij*th joint moment in the Taylor expansion, $a_{i,j}$ and $m_{i,j} - m_{i,0} m_{0,j}$. We define

$$Q_{ij} = a_{i,j} \cdot \sqrt{m_{2i,0} \cdot m_{0,2j}}. \quad (4)$$

Then the mean can be expressed as

$$E[f(X_1, X_2)] = \sum_{ij} \rho_{i,j} \cdot Q_{i,j}, \quad (5)$$

where $\rho_{i,j}$ is the *ij*th order polynomial correlation coefficient between X_1 and X_2 as defined in (1). From the above equation, we find that the importance of the *ij*th order dependence depends on $Q_{i,j}$. The above equations illustrate the two random variable case.

In practice, principal component analysis (PCA) or independent component analysis (ICA) is used to obtain principal components or independent components, respectively. Assume that

$$P=(P_1,P_2)^T=W\cdot X \tag{6}$$

are the principal components (or independent components) obtained from PCA [20], where W is the transform matrix. Then the function f can be written as the function of P_1 and P_2:

$$f(X)=f(W^{-1}\cdot P)=\sum_{ij}c_{ij}P_1^iP_2^j. \tag{7}$$

Because P is a linear combination of X, it is easy to obtain the coefficients c_{ij}, from a_{ij} and the transform matrix W.

In practice, when high order dependence exists, P_1 and P_2 are not completely independent. In this section, we try to estimate the error caused by ignoring the high order dependence. We focus on mean, variance, and skewness calculation.

We express mean of f as

$$\begin{aligned}E[f(X)]&=f(W^{-1}\cdot P)=\sum_{ij}\rho_{p,i,j}\cdot T^{\mu}_{i,j}\\&=\sum_{ij}c_{ij}P_1^iP_2^j\\&=\sum_{ij}\rho_{p,i,j}\cdot T^{\mu}_{i,j},\\T^{\mu}_{i,j}&=c_{i,j}\cdot\sqrt{m^P_{2i,0}\cdot m^P_{0,2j}},\end{aligned} \tag{8}$$

where m^P_{ij} is the ijth joint moment of P_1 and P_2, and ρ^P_{ij} is the ijth order correlation coefficient between P_1 and P_2. Since P is a linear combination of X, it is easy to obtain joint moments m^P_{ij} and correlation coefficients ρ^P_{ij} can be easily calculated from the moments of X_i's m_{ij} and the transform matrix W. If we assume that these components are independent, i.e., we assume all the ρ^P_{ij} to be zero, then total error in mean estimation is

$$\Delta_\mu=\sum_{i\ge1,j\ge1}\rho^P_{ij}\cdot T^{\mu}_{i,j}. \tag{9}$$

Similar to the estimation of the error in mean, we may estimate the error in variance calculation. We first estimate the error of second order raw moment of $f(\cdot)$. $f^2(\cdot)$ can be expressed as a polynomial function of P_i's as

$$f^2(P_1,P_2)=\sum_{ij}d_{ij}P_1^iP_2^j, \tag{10}$$

where the coefficients d_{ij} can be calculated from c_{ij}'s. Then we may estimate the error of the second order raw moment of $f(\cdot)$

$$\Delta_2=E[f^2]-E[f'^2]=\sum_{i\ge1,j\ge1}\rho^P_{ij}\cdot T^{\sigma}_{i,j}, \tag{11}$$

$$T^{\sigma}_{i,j}=d_{i,j}\cdot\sqrt{m_{2i,0}\cdot m_{0,2j}}, \tag{12}$$

where f' is the function ignoring the dependence. Then the error of variance calculation if high order dependence is ignored is

$$\begin{aligned}\Delta_{\sigma^2}&=\Delta_2-2\mu'\Delta_\mu-\Delta_\mu^2\\&=\sigma_f^2-\sigma_{f'}^2\\&=E[f^2]-(E[f])^2-E[f'^2]+(E[f'])^2\\&=\Delta_2-2\mu'\Delta_\mu-\Delta_\mu^2,\end{aligned} \tag{13}$$

where μ' is the mean calculated by ignoring the high order dependence and Δ_μ is the error of mean calculation which is calculated in (9). In practice Δ_μ is much smaller compared to μ', therefore, we have

$$\Delta_{\sigma^2}\approx\Delta_2-2\mu'\Delta_\mu. \tag{14}$$

With the error of variance, we may also calculate the error of standard deviation:

$$\Delta_\sigma=\sqrt{\sigma'^2+\Delta_{\sigma^2}}-\sigma'\approx\frac{\Delta_{\sigma^2}}{2\sigma'}. \tag{15}$$

Besides mean and variance, skewness is also an important characteristic of statistical distributions. In order to estimate the error of skewness calculation, we first estimate the error of the third order raw moment Δ_3 in a similar way to

$$\Delta_3=E[f^3]-E[f'^3]=\sum_{i\ge1,j\ge1}\rho^P_{ij}\cdot T^{\gamma}_{ij}, \tag{16}$$

$$T^{\gamma}_{ij}=u_{ij}\cdot\sqrt{m_{2i,0}\cdot m_{0,2j}}, \tag{17}$$

where the coefficients u_{ij} can be calculated from c_{ij}. Then the error of skewness can be calculated as

$$\Delta_\gamma=\frac{E[f'^3]+\Delta_3}{(\sigma'+\Delta_\sigma)^3}-\frac{E[f'^3]}{\sigma'^3}\approx\frac{\Delta_3}{\sigma'^3}. \tag{18}$$

4. Case study of statistical leakage and timing analysis

Statistical analysis is widely used in integrated circuit design. In the section, we apply our error estimation techniques on the statistical timing and leakage power analysis.

4.1. Statistical leakage analysis

4.1.1. Single cell leakage

Generally, the leakage variation of a single cell is expressed as an exponential function of variation sources [21,8,7]

$$P_{leak}=P_0\cdot e^{c_{11}X_1+c_{12}X_1^2+c_{21}X_2+c_{22}X_2^2}, \tag{19}$$

where X_1 and X_2 are the variation sources, P_0 is the nominal leakage value, c_{ij}'s are the sensitivity coefficients for variation sources X_1 and X_2, respectively. Performing Nth order Taylor expansion to the above equation, we have

$$\begin{aligned}P_{leak}&=P_0\sum_{i,j=0}^{\infty}a_{i,j}X_1^iX_2^j\\&\approx P_0\sum_{i,j=0}^{N}a_{i,j}X_1^iX_2^j.\end{aligned} \tag{20}$$

Now we have the to-be estimated function in a polynomial form of variation sources. Then we may apply the method in Section 3 to estimate the error of mean, variance, and skewness when ignoring the high order dependence.

4.1.2. Full chip leakage

Full chip leakage power is calculated as

$$P_{chipleak}=\sum_{r\in C}P^r_{leak}\approx\sum_{i,j=0}^{N}q_{i,j}X_1^iX_2^j, \tag{21}$$

$$q_{i,j}=\sum_{r\in C}a^r_{i,j}, \tag{22}$$

where C is the set of all circuit elements in the chip and $a^r_{i,j}$ is the ijth order coefficient for the rth circuit element. From the above equation, we can see that the full chip leakage can be expressed as the Taylor expansion of the variation sources. Therefore, we may estimate the error of mean, variance, and skewness calculation as mentioned previously.

4.2. Statistical timing analysis

Next, we calculate the error in statistical timing analysis.

Please cite this article as: L. Cheng, et al., Statistical timing and power analysis of VLSI considering non-linear dependence, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2013.12.004

4.2.1. Gate delay

The delay of a single gate is usually expressed as a quadratic function of variation sources [4,22–28]

$$D = a_{11}X_1^2 + a_{22}X_2^2 + 2a_{12}X_1X_2 + b_1X_1 + b_2X_2 + d_0. \tag{23}$$

$A = (a_{ij})$ is the matrix of the second-order sensitivity coefficients of delay with respect to the variation sources, $B = (b_i)$ is the vector of the linear delay sensitivity coefficients, and d_0 is the nominal delay. We can apply the method in Section 3 to estimate the error of mean, variance, and skewness variation.

From the above equation, we see that the mean of delay variation is affected by the linear correlation between X_i's and does not depend on the high order joint moments. However, the delay variance and skewness are affected by high order covariances. This is because D is a quadratic function of X_i's, then the variance is a 4th order polynomial and the skewness is a 6th order polynomial of X_i's.

4.2.2. Full chip statistical static timing analysis (SSTA)

Due to many works on SSTA making a generic analysis of errors is difficult. For block based SSTA, there are two major operations, MAX and ADD. The ADD operation is straightforward because we may obtain the coefficients of the sum $D_s = D_1 + D_2$ by adding up the coefficients of D_1 and D_2.

For the MAX operation, the problem is more involved because there is no closed-form expression for the max of two random variables. There are several algorithms to approximate the MAX operation. The error of the MAX operation depends on which algorithm we use. As an example, we consider a commonly used algorithm namely, moment matching [23,27,28]. In the moment matching technique, to compute the max of two delay value $D_m = \max(D_1, D_2)$ with the second order canonical form similar to (23)

$$D_m = a_{m11}X_1^2 + a_{m22}X_2^2 + 2a_{m12}X_1X_2 + b_{m1}X_1 + b_{m2}X_2 + d_{m0}. \tag{24}$$

The joint moments between variation sources and the max, $E[X_i \max(D_1, D_2)]$, are first computed and MAX is expressed in the second order canonical form

$$\begin{aligned}
E[D_m] &= a_{m22}m_{20} + 2a_{m12}m_{11} + a_{m11}m_{02} + b_{m1}m_{10} + b_{m2}m_{01} + d_{m0},\\
E[X_1 \cdot D_m] &= a_{m22}m_{30} + 2a_{m12}m_{21} + a_{m11}m_{12} + b_{m1}m_{20}\\
&\quad + b_{m2}m_{11} + d_{m0}m_{10},\\
E[X_2 \cdot D_m] &= a_{m22}m_{21} + 2a_{m12}m_{12} + a_{m11}m_{03} + b_{m1}m_{11}\\
&\quad + b_{m2}m_{02} + d_{m0}m_{01},\\
E[X_1^2 \cdot D_m] &= a_{m22}m_{40} + 2a_{m12}m_{31} + a_{m11}m_{22} + b_{m1}m_{30}\\
&\quad + b_{m2}m_{21} + d_{m0}m_{20},\\
E[X_2^2 \cdot D_m] &= a_{m22}m_{22} + 2a_{m12}m_{13} + a_{m11}m_{04} + b_{m1}m_{12}\\
&\quad + b_{m2}m_{03} + d_{m0}m_{02},\\
E[X_1X_2 \cdot D_m] &= a_{m22}m_{31} + 2a_{m12}m_{22} + a_{m11}m_{13} + b_{m1}m_{21}\\
&\quad + b_{m2}m_{12} + d_{m0}m_{11}.
\end{aligned} \tag{25}$$

These equations are solved to obtain the coefficients a_{mij} and b_{mij} for D_m. Notice that the above equations contain the high order joint moments between X_1 and X_2, if the high order dependence is ignored, there will be error in the coefficients a_{mij} and b_{mij}. In order to estimate the error, we may use the correct dependence to compute the correct a_{mij} and b_{mij} and then compare to those calculated by ignoring the high order dependence.

4.3. Experiments

In this section, we show experimental results on some small benchmark circuits to validate our estimation techniques.

4.3.1. Dependent variation sources generation

In our experiment, we assume two variation sources: effective channel length L_{eff} and threshold voltage V_{th}. Since these two variation sources are dependent, to generate the dependent variation sample, we assume that the variation of gate length L_{gate} and dopant density N_{bulk} are independent.[3] We first generate samples of L_{gate} and N_{bulk} then we use ITRS 2005 *MASTAR4* (*M*odel for *A*ssessment of CMO*S* *T*echnologies *A*nd *R*oadmaps) tool [29–31] to obtain dependent samples of L_{eff} and V_{th} from the samples of L_{gate} and N_{bulk}. By applying PCA (or ICA) to the samples of L_{eff} and V_{th}, we obtain the marginal distribution for each principal component (or independent component).

In the experiment, we use the samples of L_{eff} and V_{th} with the exact dependence to perform SPICE Monte-Carlo simulation to calculate the exact distribution of leakage power (or delay), which is the golden result for comparison. We also assume each principal component (or independent component) from PCA (or ICA) to be independent. Then we calculate the leakage power (or delay) under such assumption and compare the result to that of the Golden case.

4.3.2. Experimental results

In our experiments, for L_{gate} and N_{bulk}, we assume a Gaussian distribution with 3σ of 5% of the nominal value. We use 10,000 Monte-Carlo simulations to calculate the golden case leakage power. For SPICE Monte-Carlo simulation, we assume BPTM 45 nm technology. Moreover, in our experiment, we only consider inter-die variation.

Table 1 illustrates the mean, standard deviation, and skewness of different cell delays. In the table, we compare the result of Monte-Carlo (MC) simulation, the result after fitting (after fitting), and result after applying PCA (PCA). Then we calculate the error caused by curve fitting (fitting error), the error when ignoring the nonlinear dependence (PCA error), and the error predicted by our algorithm above (predicted error). In the table, we also compare the result for two different delay (leakage) models, linear model (Lin) and quadratic delay (leakage) model (Quad). For the linear leakage model, we just fit the leakage power as the exponential of the linear function of variation sources, that is, not the square term in the power in (19). For the linear delay model, we just fit the gate delay as a linear function of variation sources, that is, no second order terms in (23).

From the table, we see that, as expected, the linear delay model leads to larger fitting error but almost does not depend on high order correlation. However, the quadratic delay model has smaller fitting error, but there is error (about 5%) of standard deviation if we ignore the non-linear correlation. Moreover, we see that error predicted by our algorithm (predicted error) is very close to the experimental result (PCA error). Table 2 illustrates the mean, standard deviation, and skewness of different cell leakage power. From the table, we can find a similar trend as delay except that in both linear and quadratic delay models, ignoring high order dependence may cause error in both mean and standard deviation.

We also show some full chip delay and leakage analysis for few ISCAS85 benchmarks in Tables 3 and 4. In the tables, we compare the result of Monte-Carlo (MC) simulation, the result of SSTA (SSTA) or statistical leakage analysis (stat leak), and delay after applying PCA (PCA). Then we calculate the error of SSTA (SSTA error) or statistical leakage analysis (stat leak error), the error when ignoring the nonlinear dependence (PCA error), and the

[3] Notice that in practice, L_{gate} and N_{bulk} cannot be easily measured in silicon. The only parameters we can measure is L_{eff} and V_{th}. That is, we can only extract the dependence between L_{eff} and V_{th} from the measured samples without knowing the exact variation of L_{gate} and N_{bulk}.

Table 1
Cell delay.

Gate	Fitting type	MC			After fitting		
		μ	3σ	γ	μ	3σ	γ
Inv	Lin	5.12	1.12	0.12	5.10	1.03	0.10
	Quad	5.12	1.12	0.12	5.14	1.14	0.12
Nand	Lin	9.29	1.95	0.14	9.20	1.84	0.11
	Quad	9.29	1.95	0.14	9.33	1.98	0.15
Nor	Lin	12.32	2.78	0.14	12.12	2.52	0.12
	Quad	12.32	2.78	0.14	12.38	2.85	0.14

Gate	Fitting type	PCA			Fitting error		
		μ	3σ	γ	μ	3σ	γ
Inv	Lin	5.09	1.02	0.10	−0.03	−0.09	−0.02
	Quad	5.15	1.02	0.11	0.02	0.02	0
Nand	Lin	9.18	1.82	0.11	−0.09	−0.11	−0.03
	Quad	9.36	1.89	0.13	0.04	0.03	0.01
Nor	Lin	12.11	2.51	0.12	−0.20	−0.22	−0.02
	Quad	12.41	2.69	0.13	0.06	0.07	0

Gate	Fitting type	PCA error			Predicted error		
		μ	3σ	γ	μ	3σ	γ
Inv	Lin	−0.01	−0.01	0	0	0	0
	Quad	0.01	−0.12	−0.01	0	−0.10	−0.01
Nand	Lin	−0.02	−0.02	0	0	0	0
	Quad	0.03	−0.09	−0.02	0	−0.11	−0.02
Nor	Lin	−0.01	−0.01	0	0	0	0
	Quad	0.03	−0.16	−0.01	0	−0.14	−0.01

Note:delay value is in ps.

Table 2
Cell leakage.

Gate	Fitting type	MC			After fitting		
		μ	3σ	γ	μ	3σ	γ
Inv	Lin	7.12	2.55	0.35	6.44	2.13	0.31
	Quad	7.12	2.55	0.35	7.15	2.61	0.36
Nand	Lin	11.28	3.84	0.34	10.18	3.19	0.31
	Quad	11.28	3.84	0.34	11.69	4.09	0.35
Nor	Lin	17.18	4.82	0.30	15.91	4.08	0.27
	Quad	17.18	4.82	0.30	17.72	5.13	0.32

Gate	Fitting type	PCA			Fitting error		
		μ	3σ	γ	μ	3σ	γ
Inv	Lin	6.32	2.02	0.27	−0.68	−0.42	−0.04
	Quad	7.05	2.49	0.31	0.03	0.06	0.01
Nand	Lin	9.97	3.01	0.27	−1.10	−0.65	−0.03
	Quad	11.34	3.91	0.31	0.41	0.25	0.01
Nor	Lin	15.69	3.94	0.24	−1.27	−0.74	−0.03
	Quad	17.42	4.96	0.28	0.54	0.31	0.02

Gate	Fitting type	PCA error			Predicted error		
		μ	3σ	γ	μ	3σ	γ
Inv	Lin	−0.08	−0.11	−0.04	−0.11	−0.13	−0.03
	Quad	−0.10	−0.12	−0.05	−0.12	−0.14	−0.04
Nand	Lin	−0.21	−0.18	−0.04	−0.19	−0.16	−0.03
	Quad	−0.35	−0.18	−0.04	−0.30	−0.17	−0.04
Nor	Lin	−0.22	−0.14	−0.03	−0.19	−0.16	−0.03
	Quad	−0.30	−0.17	−0.04	−0.31	−0.19	−0.05

Note:leakage value is in nW.

Table 3
Chip delay.

Bench mark	SSTA type	MC			SSTA		
		μ	3σ	γ	μ	3σ	γ
C17	Lin	42.2	14.7	0.10	41.3	13.9	0.09
	Quad	42.2	14.7	0.10	42.7	15.2	0.12
C499	Lin	320.2	105.5	0.14	318.2	102.9	0.13
0	0	−0.01					
	Quad	320.2	105.5	0.14	321.1	106.2	0.15
C880	Lin	674.4	221.3	0.12	671.2	215.2	0.11
	Quad	674.4	221.3	0.12	679.5	227.2	0.14
C3540	Lin	1241.2	413.2	0.13	1237.1	410.1	0.12
	Quad	1241.2	413.2	0.13	1249.7	420.2	0.15
C7522	Lin	1919.3	635.4	0.13	1911.3	631.2	0.11
	Quad	1919.3	635.4	0.12	1931.2	638.9	0.13
Bench mark	SSTA type	PCA			SSTA error		
		μ	3σ	γ	μ	3σ	γ
C17	Lin	41.2	13.8	0.09	−0.9	−0.8	−0.01
	Quad	42.0	15.0	0.11	0.5	0.5	0.02
C499	Lin	317.9	102.6	0.12	−2.0	−2.6	−0.01
	Quad	319.2	104.9	0.13	0.9	0.7	0.01
C880	Lin	671.5	214.8	0.11	−3.2	−6.1	−0.01
	Quad	676.7	225.3	0.13	5.1	5.9	0.2
C3540	Lin	1236.5	409.7	0.11	−4.1	−3.1	−0.01
	Quad	1247.1	418.2	0.13	8.5	7.0	0.2
C7522	Lin	1907.1	631.9	0.12	−8.0	−4.2	−0.02
	Quad	1929.4	639.5	0.13	11.1	3.5	0.01
Bench mark	SSTA type	PCA error			Predicted error		
		μ	3σ	γ	μ	3σ	γ
C17	Lin	−0.1	−0.1	0	0	0	−0.01
	Quad	−0.7	−0.2	−0.01	−0.6	−0.2	−0.01
C499	Lin	−0.3	−0.3	−0.01	0	0	−0.01
	Quad	−1.9	−1.3	−0.02	−2.1	−1.1	−0.02
C880	Lin	0.3	−0.4	0	0	−0	−0.0
	Quad	−2.8	−1.9	−0.01	−2.9	−1.8	−0.01
C3540	Lin	−0.6	−0.4	−0.01	−0.5	−0.5	−0.01
	Quad	−2.6	−2.0	−0.02	−2.7	−1.9	−0.01
C7522	Lin	−3.2	0.7	0.01	−2.9	0.6	0.01
	Quad	−1.8	0.6	0.00	−1.7	0.5	0.01

Note:delay value is in ps.

error predicted by our algorithm (predicted error). Notice that SSTA error and the statistical leakage analysis error are caused by both curve fitting and analysis algorithm. Similar to the single gate case, we see that error predicted by our algorithm (predicted error) is very accurate compared to the experimental result (PCA error). From the tables, we see that the error caused by non-linear dependence is not significant in the ISCAS85 circuit bench.[4]

5. Target function driven component analysis

In the previous section, we introduced the method to estimate the error caused by ignoring non-linear dependence and showed that it depends on the target function being estimated. As discussed in Section 1, linear operations cannot completely remove the dependence between variation sources. However, due to simplicity of application, linear operation is preferred. Therefore, in this section, we try to find an optimum linear transform to minimize the error of ignoring the non-linear dependence. The proposed algorithm, function driven component analysis (*FCA*), decomposes dependent variation sources into components so as to minimize error in the estimation of certain statistical measures of the target function.

In the rest of this section, we first present our algorithm and then apply it to statistical leakage analysis and SRAM cell noise margin variation analysis. Note that the method can also be applied to the variation analysis of emerging memory technologies, such as STT-RAM.

5.1. FCA algorithm

Let $f(X)$ be a polynomial function (or Taylor expansion of an arbitrary function) of an n-dimensional random vector $X=(X_1, X_2, \ldots X_n)^T$. The objective of the *FCA* is to find an $n\times n$ transfer matrix W and independent components $P=(P_1, P_2, \ldots, P_n)=W\cdot X$

[4] Especially for statistical timing analysis in this experiment, such error is less than 2%.

Table 4
Chip leakage. Note: leakage value is in μW for C17, and in mW for others.

Gate	Fitting type	MC			Stat leak		
		μ	3σ	γ	μ	3σ	γ
C17	Lin	430.2	120.3	0.28	415.3	113.3	0.25
	Quad	430.2	120.3	0.28	437.2	126.3	0.32
C499	Lin	9.14	3.21	0.32	8.54	2.92	0.29
	Quad	9.14	3.21	0.32	9.45	3.58	0.35
C880	Lin	22.3	7.58	0.34	20.2	6.95	0.29
	Quad	22.3	7.58	0.34	23.8	7.92	0.39
C3540	Lin	89.2	29.23	0.41	84.5	27.78	0.39
	Quad	89.2	29.23	0.41	92.5	30.29	0.43
C7522	Lin	162.2	56.21	0.38	155.3	55.51	0.35
	Quad	162.2	56.21	0.38	170.1	57.11	0.40
Gate	Fitting type	PCA			Stat Leak error		
		μ	3σ	γ	μ	3σ	γ
C17	Lin	413.2	109.2	0.23	−14.9	−7.0	−0.03
	Quad	431.5	122.2	0.30	7.0	6.0	0.02
C499	Lin	8.32	2.79	0.27	−0.60	−0.29	−0.03
	Quad	9.21	3.40	0.33	0.41	0.37	0.03
C880	Lin	19.1	6.63	0.26	−2.1	−0.63	−0.05
	Quad	23.3	7.76	0.37	1.5	0.34	0.05
C3540	Lin	82.1	27.15	0.37	−4.7	−1.45	−0.02
	Quad	90.7	30.01	0.42	3.3	1.04	0.02
C7522	Lin	152.1	55.13	0.31	−6.9	−0.80	−0.03
	Quad	168.2	56.72	0.39	7.9	0.90	0.02
Gate	Fitting type	PCA error			Predicted error		
		μ	3σ	γ	μ	3σ	γ
C17	Lin	−2.1	−4.1	−0.02	−2.3	−3.9	−0.02
	Quad	−5.7	−4.1	−0.02	−5.3	−3.8	−0.03
C499	Lin	−0.22	−0.13	−0.02	−0.19	−0.12	−0.02
	Quad	−0.24	−0.18	−0.02	−0.28	−0.17	−0.02
C880	Lin	−1.1	−0.32	−0.03	−0.9	−0.36	−0.02
	Quad	−0.5	−0.16	−0.02	−0.4	−0.18	−0.02
C3540	Lin	−2.4	−0.53	−0.02	−2.2	−0.59	−0.02
	Quad	−1.8	−0.28	−0.01	−1.5	−0.27	−0.01
C7522	Lin	−3.2	−0.38	−0.04	−2.9	−0.36	−0.03
	Quad	−1.9	−0.39	−0.01	−1.7	−0.37	−0.01

Note:leakage value is in μW for C17, and in mW for others.

to minimize the error of $f(WP)$ when assuming that all P_i's are independent. In statistical analysis, the error of $f(WP)$ is usually measured by mean, variance, and skewness. In this work, we consider the first-order analysis by matching the mean of $f(X)$. Those are

$$W = \underset{\Delta_\mu = 0}{\arg} \min \Delta, \tag{26}$$

$$\Delta = \Delta_\sigma + \varepsilon \Delta_\gamma, \tag{27}$$

$$\Delta_\mu = \mu_f - \mu_{f'}, \tag{28}$$

$$\Delta_\sigma = \sigma_f - \sigma_{f'}, \tag{29}$$

$$\Delta_\gamma = \gamma_f - \gamma_{f'}, \tag{30}$$

where μ_f, σ_f, and γ_f are the mean, standard deviation, and skewness of $f(X)$, respectively; $\mu_{f'}$, $\sigma_{f'}$, and $\gamma_{f'}$ are the mean, standard deviation, and skewness of $f(WP)$ when assuming that all P_i's are independent; ε is the weight factor for the skewness error. Since $f(X)$ is a polynomial function of X, similar to (9), (15), and (18), μ_f, σ_f, and γ_f can be expressed as a function of joint moments of X_i's, which are known; and $\mu_{f'}$, $\sigma_{f'}$, and $\gamma_{f'}$ can be expressed as a function of joint moments of P_i's. Considering $P=WX$, the joint moments of P_i's can be expressed as functions of W and the joint moments of X_i's. Hence, the error Δ can be expressed as a function of W and joint moments of X_i's. Therefore (26) becomes a non-linear programming problem. We use a non-linear programming solver to obtain the transfer matrix W.

A more general objective is[5]

$$W = \arg\min(\Delta_\mu + \varepsilon_1 \Delta_\sigma + \varepsilon_2 \Delta_\gamma).$$

Unlike the regular PCA or ICA, our *FCA* algorithm presented above tries to minimize the error for a target function f. That is, for different target function f, we may have different transfer matrix

[5] This is especially useful in cases where $\mu = 0$ has no solution.

W. In *FCA*, we need to obtain an $n \times n$ transfer matrix W, that is, we need to solve a n^2 variable non-linear programming problem. However, for any statistical analysis, *FCA* needs to be run only once. Moreover, *FCA* still uses linear operation to decompose the variation sources. Therefore, applying *FCA* does not increase the computational complexity of the statistical analysis compared to regular PCA or ICA.

In order to validate our algorithm, let us first take a look at the simple example we introduced in Section 1: let S_1 and S_2 be two independent random variables with standard normal distributions and $X_1 = S_1 + S_2$, $X_2 = S_1 S_2$. Estimate the mean of $f(X_1, X_2) = X_1^2 + X_1 X_2 + X_1^2 X_2^2 + X_2^2$. As discussed in Section 1, the correct value is $E[f(\cdot)] = 9$. If we apply PCA, because X_1 and X_2 are uncorrelated, they are just principal components. If we assume that principal components X_1 and X_2 are independent, we have $E[f(\cdot)] = 5$. If we apply fast kernel ICA [32] to obtain independent components, we will have $E[f(\cdot)] = 5.78$. If we use *FCA*, we have $E[f(\cdot)] = 8.54$. That is, FCA works better than PCA and ICA.

5.2. Experimental results

In this section, we show some examples to validate the *FCA* algorithm. As discussed in Section 4.3, non-linear dependence does not have significant impact on statistical timing analysis. In this section we show three examples of *FCA* in VLSI design: statistical leakage analysis, differential Opamp amplitude, and SRAM noise margin variation analysis.

5.2.1. Statistical leakage analysis

We first discuss statistical leakage analysis. Similar to Section 4.3, we assume two variation sources, effective L_{eff} and V_{th} and we only consider inter-die variation for the variation sources. We generate dependent variation samples of L_{eff} and V_{th} in the same way as Section 4.3.1. With the dependent samples, we use *FCA* (PCA or ICA) to decompose the variation sources and obtain the marginal distribution of each component. Then we generate sample of each component according to its marginal distribution. Assuming that the components are independent, we generate the samples of L_{eff} and V_{th}. Finally, we use these samples to run SPICE Monte-Carlo simulation to obtain leakage power. We use BPTM 45 nm technology in the experiment and assume supply voltage to be 1.0 V. For L_{gate} and N_{bulk}, we assume that they follow Gaussian distribution and the 3-sigma value is 5% of the nominal value.

In order to validate the accuracy of *FCA* we define three comparison cases: (1) samples generated from Mastar4 with the exact dependence, which is the golden case for comparison, (2) samples generated from PCA, and (3) samples generated from fast kernel ICA [32].

Table 5 illustrates the mean, standard deviation, skewness, 90%, 95%, and 99% percentile point of leakage of different logic cells. From the table, we see that the value obtained from *FCA* is closer to the exact value than PCA and ICA.[6] Table 6 illustrates the leakage comparison for full chip leakage power analysis. For full chip leakage power analysis, *FCA* may give out different decomposition matrices for different cells. In this experiment, we apply the decomposition matrix obtained from the inverter for all logic cells in the chip. From the table, we see that even *FCA* works well in full chip leakage analysis.

Table 7 illustrates the exact error and the estimated error (using the method in Section 4) of mean, standard deviation, and skewness for logic cells. From the table, we can find that the estimated error is close to the exact error and that *FCA* has a lower error than PCA or ICA.

[6] The run time for PCA and ICA is less than 0.1 s, and the run time for *FCA* is 0.4 s. However, because *FCA* needs to be run only once in the statistical analysis, such run time overhead is a non-issue.

Table 5
Logic cell leakage power comparison.

Gate	μ	3σ	γ	90%	95%	99%
INV						
Exact	7.15	2.61	0.36	10.12	10.75	11.37
PCA	7.05	2.49	0.31	9.85	10.34	11.01
ICA	7.07	2.47	0.32	9.91	10.41	11.09
FCA	7.10	2.56	0.34	10.02	10.59	11.21
NAND						
Exact	11.69	4.09	0.35	15.95	16.82	17.76
PCA	11.34	3.91	0.31	15.51	16.41	17.41
ICA	11.32	3.93	0.34	15.60	16.45	17.50
FCA	11.68	4.05	0.35	15.83	16.68	17.62
NOR						
Exact	17.72	5.13	0.32	23.89	24.95	26.11
PCA	17.42	4.96	0.28	23.44	24.61	25.94
ICA	17.51	5.01	0.30	23.51	24.64	25.99
FCA	17.71	5.09	0.31	23.74	24.85	26.01

Note:leakage value is in nW.

Table 6
Chip leakage power comparison.

Gate	μ	3σ	γ	90%	95%	99%
C17						
Exact	437.2	126.3	0.32	612.4	675.3	721.5
PCA	431.5	122.2	0.30	592.8	653.5	701.8
ICA	432.8	123.5	0.30	599.2	654.4	704.6
FCA	437.3	124.9	0.31	604.2	664.3	711.2
C499						
Exact	9.45	3.58	0.35	10.31	11.15	12.01
PCA	9.21	3.40	0.33	10.02	10.98	11.78
ICA	9.25	3.45	0.34	10.12	11.01	11.81
FCA	9.44	3.51	0.35	10.16	11.06	11.91
C880						
Exact	23.8	7.92	0.39	26.11	29.15	31.31
PCA	23.3	7.76	0.37	25.56	28.41	29.10
ICA	23.2	7.72	0.38	25.32	28.35	29.01
FCA	23.9	7.81	0.38	25.74	28.67	29.65
C3540						
Exact	92.5	30.29	0.43	107.2	119.5	131.2
PCA	90.7	30.01	0.42	103.1	114.3	124.6
ICA	90.9	30.12	0.42	104.1	116.3	125.6
FCA	92.1	30.13	0.42	105.6	116.9	128.4
C7522						
Exact	170.1	57.11	0.40	193.3	218.6	231.2
PCA	168.2	56.72	0.39	184.6	209.3	221.5
ICA	167.9	56.81	0.39	185.2	211.3	223.3
FCA	169.7	56.86	0.40	187.1	213.5	225.9

Note:leakage value is in.μW for C17, and in mW for others.

5.2.2. Differential amplifier analysis

The second application example for *FCA* is the simple one stage differential operation amplifier amplitude. We use the same device setting as the statistical leakage analysis in Section 5.2.1. The only difference is that in this experiment, we consider the mismatch of L_{eff} and V_{th} of the two input transistors. We assume that the 3-sigma of both mismatch variation is 5% of the nominal value.

Table 8 illustrates the mean, standard deviation, skewness, 90%, 95%, and 99% percentile point of amplitude of the Opamp. From the table, we see that the value obtained from *FCA* is closer to the exact

Please cite this article as: L. Cheng, et al., Statistical timing and power analysis of VLSI considering non-linear dependence, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2013.12.004

Table 7
Estimated error for the logic cell leakage power.

Gate	Exact error			Est error		
	μ	3σ	γ	μ	3σ	γ
INV						
PCA	−0.10	−0.12	−0.05	−0.11	−0.14	−0.04
ICA	−0.08	−0.14	−0.04	−0.10	−0.16	−0.04
FCA	−0.05	−0.05	−0.02	0	−0.14	−0.03
NAND						
PCA	−0.35	−0.18	−0.04	−0.39	−0.17	−0.03
ICA	−0.37	−0.16	−0.01	−0.34	−0.13	−0.02
FCA	−0.01	−0.04	−0.00	0	−0.06	−0.01
NOR						
PCA	−0.30	−0.17	−0.04	−0.28	−0.19	−0.04
ICA	−0.21	−0.12	−0.02	−0.24	−0.14	−0.03
FCA	−0.01	−0.04	−0.01	0	−0.05	−0.02

Note:leakage value is in nW.

Table 8
Opamp amplitude comparison.

Name	μ	3σ	γ	90%	95%	99%
Exact	742.4	65.0	0.96	682.2	646.3	712.8
PCA	749.7	61.4	0.84	687.4	648.6	615.3
ICA	748.0	62.1	0.85	686.2	648.7	615.2
FCA	743.7	62.6	0.88	685.9	647.7	615.0

Table 9
Estimated error for the Opamp amplitude.

Name	Exact error			Est error		
	μ	3σ	γ	μ	3σ	γ
PCA	7.3	−3.4	−0.12	6.9	−3.3	−0.10
ICA	5.6	−2.9	−0.11	5.2	−3.1	−0.09
FCA	1.3	−2.4	−0.08	0	−2.1	−0.05

value than PCA and ICA, which is similar to the leakage power variation analysis. In Table 9, we compare the exact error and the error estimated by the method in Section 3. We also observe that the estimated error is very close to the exact value as expected.

5.2.3. SRAM noise margin variation analysis

The third application example for *FCA* is the 6 T-SRAM cell noise margin (SNM). We use similar setting to the statistical leakage analysis in Section 5.2.1. In order to highlight the flexibility of *FCA*, in this experiment, we consider only within-die variation. That is, each transistor has its own variation. In practice, SNM is mainly affected by within die variation of the 4 transistors which make two inverters, and inter-die variation and variation of the two pass-transistor has little impact on SNM. Therefore, in our experiment, we only consider within-die variation for those 4 transistors. In this case, because we consider 4 transistors in an SRAM cell, there are 8 variation sources in an SRAM (L_{eff} and V_{th} for all 4 transistors). Notice that PCA and ICA provide the same transfer matrix for L_{eff} and V_{th} for all transistors, however because *FCA* tries to handle 8 variation sources together, it may provide different transfer matrices for different transistors.

Table 10 illustrates the mean, standard deviation, skewness, 90%, 95%, and 99% percentile point of noise margin of an SRAM. From the table, we find that the value obtained from *FCA* is closer to the exact value than PCA and ICA. Table 11 compares the exact

Table 10
SNM comparison.

Name	Nominal value			0.1678		
	μ (mV)	σ (mV)	γ	90% (mV)	95% (mV)	99% (mV)
Exact	148	29.4	0.091	105	84.4	65.5
PCA	141	30.7	0.077	107	89.2	69.0
ICA	140	30.3	0.079	109	88.4	68.1
FCA	147	28.9	0.094	106	85.8	67.2

Table 11
Estimated error for the SNM variation analysis.

Name	Exact error			Est error		
	μ	σ	γ	μ	σ	γ
PCA	−4.7	1.3	−0.014	−4.9	1.2	−0.015
ICA	−5.0	0.9	−0.012	−4.8	0.9	−0.014
FCA	−1.4	−0.5	0.003	0.0	−0.4	0.005

Note:the error of μ and σ is in mV.

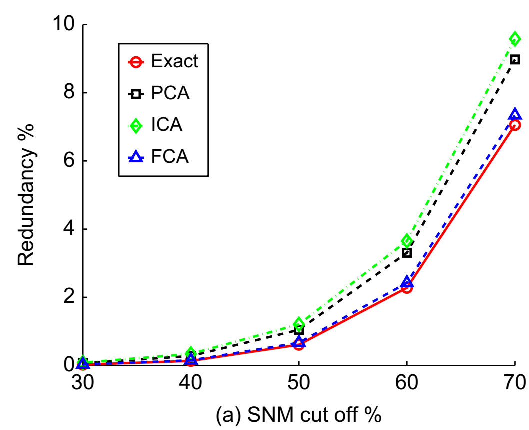


Fig. 1. Redundancy for Non-ECC scheme to achieve 99% yield rate.

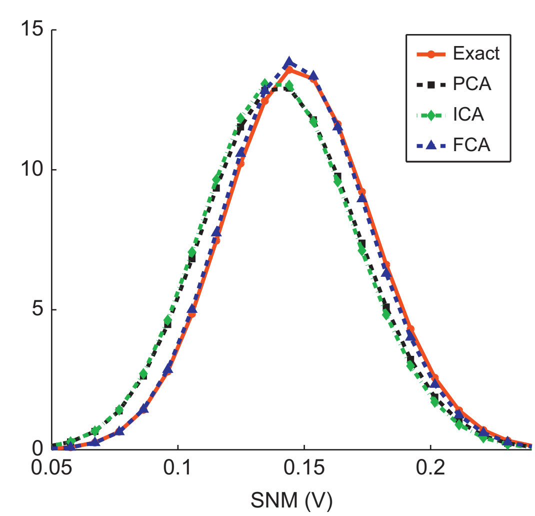


Fig. 2. SNM PDF comparison.

error and the error estimated by the method in Section 3. We also find that the estimated error is very close to the exact value.

With noise margin variation analysis, we may further estimate a number of redundant SRAM cells needed to ensure error correct SRAM array. We assume that the variation of all SRAM cells in the

Table 12
SRAM cell noise margin comparison assuming L_{gate} and N_{bulk} to be with skew-normal distribution.

Name	μ (mV)	σ (mV)	γ	90% (mV)	95% (mV)	99% (mV)
Exact	158	32.8	0.345	112	94.2	74.8
PCA	161	31.3	0.297	120	99.4	80.2
ICA	160	31.8	0.307	115	97.3	76.3
FCA	159	33.1	0.335	114	96.6	75.5

Table 13
Estimated error for the SNM variation analysis assuming L_{gate} and N_{bulk} to be with skew-normal distribution.

Name	Exact error			Est error		
	μ	σ	γ	μ	σ	γ
PCA	3.4	−1.5	−0.048	3.9	−1.1	−0.031
ICA	2.1	−1.0	−0.038	1.4	−0.7	−0.027
FCA	1.2	0.3	−0.010	0.0	0.2	−0.019

Note:the error of μ and σ is in mV.

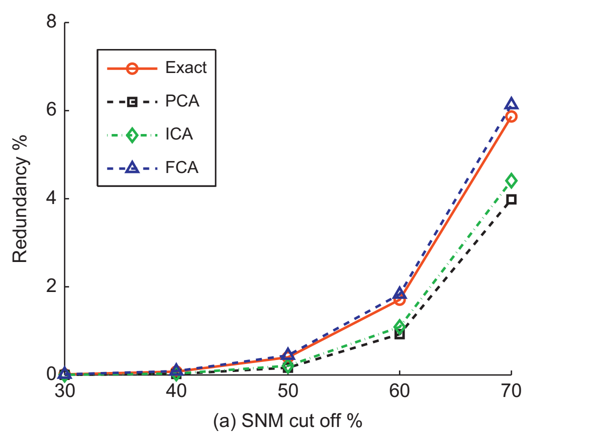


Fig. 3. Redundancy for Non-ECC scheme to achieve 99% yield rate assuming L_{gate} and N_{bulk} to be with skew-normal distribution.

array are independent and an SRAM cell is faulty when the noise margin is less than a cut off value. For non-ECC architecture, for simplicity, we calculate the number of redundant SRAM cells needed to achieve a certain percent yield. For the ECC scheme, the number of redundant SRAM cells depends on the coding. For simplicity, we estimate the Shannon Channel limit [33], which is the lower bound of the redundancy required to achieve no error coding.

Fig. 1 illustrates the percentage SRAM redundancy under different cut off SNM values. In the figure, the x-axis is the cut off SNM value, which is calculated as a certain percentage of the nominal value (0.152 V). The y-axis is the percentage redundancy. For the non-ECC scheme, we assume that the redundancy is to achieve 99% yield rate.[7] Fig. 2 compares the PDFs predicted by ICA, PCA, and *FCA* to the exact PDF. From the figures, we see that *FCA* predicts the redundancy more accurately than PCA or ICA.

We also ran experiments for different variation settings. In stead of assuming L_{gate} and N_{bulk} to be Gaussian, we assume that they follow skew-normal distribution with $\alpha = 10$ [34]. Table 12 illustrates the mean, standard deviation, skewness, 90%, 95%, and 99% percentile point of noise margin of an SRAM under such setting. Table 13 illustrates the estimated error for PCA, ICA, and *FCA* when assuming that all variation sources follow skew-normal distribution. Fig. 3 illustrates redundancy and Fig. 4 compares the PDFs. From the table and figure, we find that *FCA* works better than PCA and ICA under different variation settings.

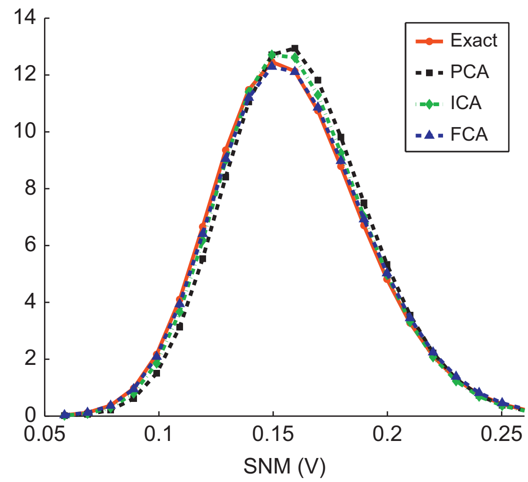


Fig. 4. SNM PDF comparison assuming L_{gate} and N_{bulk} to be with skew-normal distribution.

6. Conclusion

In this paper, we have proposed the first method to estimate the error of statistical analysis when ignoring the non-linear dependence using *polynomial correlation coefficients*. Such a method can be used to evaluate the accuracy of the linear de-correlation techniques like PCA for a particular analysis problem. As examples, we apply our technique to statistical timing and power analysis. Experimental result shows that the error predicted by our method is within 1% compared to the real simulation. We have further proposed a novel target function driven component analysis (*FCA*) algorithm to minimize the error caused by ignoring high order dependence. We apply such a technique to two applications of statistical analysis, statistical leakage power analysis and SRAM cell noise margin variation analysis. Experimental results show that the proposed FCA method is more accurate compared to the traditional PCA or ICA. In the future work, we will evaluate our work with larger-scale industrial circuits. Also, sparse approximation based parameter estimation methods [35–37] will be considered to reduce the need of measurements in the statistical model.

[7] This is just a simple estimation. In practice, because redundancy is needed for each row and column of SRAM array, the actual redundancy may be much higher.

References

[1] J.-J. Liou, A. Krstic, L.-C. Wang, K.-T. Cheng, False-path-aware statistical timing analysis and efficient path selection for delay testing and timing validation, in: Design Automation Conference, Proceedings, June 2002, pp. 566–569.
[2] J. Jess, K. Kalafala, S. Naidu, R. Otten, C. Visweswariah, Statistical timing for parametric yield prediction of digital integrated circuits, in: DAC 03, June 2003.
[3] M. Orshansky, A. Bandyopadhyay, Fast statistical timing analysis handling arbitrary delay correlations, in: DAC 04, June 2004.
[4] S. Bhardwaj, P. Ghanta, S. Vrudhula, A framework for statistical timing analysis using non-linear delay and slew models, in: ICCAD, November 2006.
[5] W. Xu, J. Wang, Y. Hu, J.-Y. Lee, F. Gong, L. He, M. Sarrafzadeh, In-place FPGA retiming for mitigation of variational single-event transient faults, IEEE Trans. Circuits Syst. Part I (TCAS-I) 58 (June) (2011) 1372–1381.
[6] L. Cheng, Y. Lin, L. He, Tracebased framework for concurrent development of process and FPGA architecture considering process variation and

Please cite this article as: L. Cheng, et al., Statistical timing and power analysis of VLSI considering non-linear dependence, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2013.12.004

reliability, in: Proceedings of the International Symposium on FPGAs, February 2008.

[7] C.H. Tan, J. Allen, Minimization of power in VLSI circuits using transistor sizing, input ordering, and statistical power estimation, in: Proceedings of the International Workshop on Low Power Design, April 1994, pp. 75–79.

[8] H.-Y. Wong, L. Cheng, Y. Lin, L. He, FPGA device and architecture evaluation considering process variations, in: ICCAD, November 2005.

[9] L. Cheng, W. Xu, F. Gong, Y. Lin, H.-Y. Wong, L. He, Statistical timing and power optimization of architecture and device for FPGAs, ACM Trans. Reconfigurable Technol. Syst. (TRTS) 5 (June) (2012).

[10] J. Xiong, V. Zolotov, L. He, Robust extraction of spatial correlation, in: Proceedings of the International Symposium on Physical Design, April 2006.

[11] R.B. Nelsen, An Introduction to Copulas, Springer, New York City, USA, 2007.

[12] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley and Sons, Hoboken, New Jersey, USA, 2006.

[13] Mark A. Kramer, nonlinear principal component analysis using autoassociative neural networks, AIChE J. 37 (2) (1991) 233–243.

[14] Christian Jutten, Juha Karhunen, Advances in blind source separation (BSS) and independent component analysis (ICA) for nonlinear mixtures, Int. J. Neural Sys. 14 (5) (2004) 267.

[15] L. Cheng, P. Gupta, L. He, Accounting for non-linear dependence using function driven component analysis, in: The 2009 Asia and South Pacific Design Automation Conference, 2009.

[16] L. Cheng, J. Xiong, L. He, Non-linear statistical static timing analysis for non-Gaussian variation sources, in: DAC, June 2007.

[17] Fengbo Ren, Henry Park, Chih-Kong Ken Yang, Dejan Markovic, Reference Calibration of Body-Voltage Sensing Circuit for High-Speed STT-RAMs, IEEE Transactions on Circuits And Systems—I: Regular Papers (TCAS-I) 60 (11) (2013) 2932–2939.

[18] L. Cheng, F. Gong, W. Xu, J. Xiong, L. He, M. Sarrafzadeh, Fourier series approximation for max operation in non-Gaussian and quadratic statistical static timing analysis, IEEE Trans. Very Large Scale Integr. Syst. (TVLSI) 20 (August) (2012) 1383–1391.

[19] F. Ren, D. Markovic, True energy-performance analysis of the MTJ-based logic-in-memory architecture (1-bit full adder), IEEE Trans. Electron Devices 57 (May) (2010) 1023–1028.

[20] K. Esbensen, P. Geladi, Principle component analysis, in: Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, August 1987.

[21] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, V. De, Parameter variations and impacts on circuits and microarchitecture, in: DAC 03, June 2003.

[22] L. Zhang, W. Chen, Y. Hu, J.A. Gubner, C.C.-P. Chen, Correlation-preserved non-Gaussian statistical timing analysis with quadratic timing model, in: DAC, June 2005, pp. 83–88.

[23] Y. Zhan, A.J. Strojwas, X. Li, L.T. Pileggi, Correlation-aware statistical timing analysis with non-Gaussian delay distribution, in: DAC, Anaheim, CA, June 2005, pp. 77–82.

[24] V. Khandelwal, A. Srivastava, A general framework for accurate statistical timing analysis considering correlations, in: DAC, June 2005, pp. 89–94.

[25] J. Singh, S. Sapatnekar, Statistical timing analysis with correlated non-Gaussian parameters using independent component analysis, in: ACM/IEEE International Workshop on Timing Issues, February 2006, pp. 143–148.

[26] H. Chang, V. Zolotov, C. Visweswariah, S. Narayan, Parameterized block-based statistical timing analysis with non-Gaussian and nonlinear parameters, in: DAC, Anaheim, CA, June 2005, pp. 71–76.

[27] Y. Zhan, A.J. Strojwas, D. Newmark, M. Sharma, Generic statistical timing analysis with non-Gaussian process parameters, in: Austin Conference on Integrated Systems and Circuits, May 2006.

[28] L. Cheng, J. Xiong, L. He, Non-Gaussian statistical timing analysis using second-order polynomial fitting, in: Proceedings of the Asia South Pacific Design Automation Conference, January 2008.

[29] International Technology Roadmap for Semiconductor, 2005, ⟨http://public.itrs.net/⟩.

[30] International Technology Roadmap for Semiconductor, A User's Guide to MASTAR4, 2005, http://www.itrs.net/models.html.

[31] T. Skotnicki, Heading for decananometer CMOS—is navigation among icebergs still a viable strategy?, in: Solid-State Device Research Conference, September 2000, pp. 19–33.

[32] H. Shen, S. Jegelka, A. Gretton, Fast kernel ICA using an approximate Newton method, in: 11th International Conference on Artificial Intelligence and Statistic, 2007.

[33] R.E. Blahut, Algebraic Codes for Data Transmission, Cambridge University Press, Cambridge, UK, 2002.

[34] A. Azzalini, The skew-normal distribution and related multivariate families, Scand. J. Stat. 32 (June) (2005) 159–188.

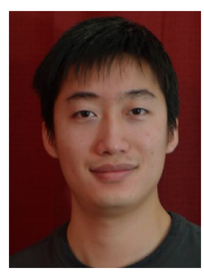
[35] X. Li, Finding deterministic solution from underdetermined equation: large-scale performance modeling by least angle regression, in: IEEE/ACM Design Automation Conference (DAC), June 2009, pp. 364–369.

[36] W.X.F. Ren, D. Markovic, Scalable and parameterized VLSI architecture for efficient sparse approximation in FPGAs and SOCs, IET Electron. Lett. 49 (November) (2013) 1440–1441.

[37] W. Xu, C. Gu, C. Li, M. Sarrafzadeh, Robust doppler radar demodulation via compressed sensing, IET Electron. Lett. 48 (October) (2012) 1428–1430.

Lerong Cheng received the B.S. degree in electronics and communications engineering from Zhongshan University, Guangzhou, China, in2001, the M.S. degree in electrical and computer engineering from Portland State University, Portland, OR, in 2003, and the Ph.D degree in electrical engineering from the University of California at Los Angeles, Los Angeles, in 2009. He is currently a Computer-Aided Design (CAD) Engineer with SanDisk Corporation, San Jose, CA. His current research interests include CAD of very large-scale integrated circuits and systems, programmable fabrics, low-power and high-performance designs, and statistical timing analysis.

Wenyao Xu received the Ph.D. degree from the Electrical Engineering Department, University of California, Los Angeles, CA, USA, in 2013. Currently, he is an Assistant Professor with the Computer Science and Engineering Department, University at Buffalo, the State University of New York (SUNY), New York, NY, USA. His current research interests include embedded system design, computational modeling, algorithm design, human computer interaction, integrated circuit design technologies, and their applications in medical and health applications. He received the Best Medical & Performance Application Paper Award of the IEEE Conference on Implantable and Wearable Body Sensor Networks in 2013 and the Best Demonstration Award of ACM Wireless Health Conference in 2011.



Fengbo Ren received the B.Eng. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2008, and the M.S. degree from University of California, Los Angeles, CA, USA, in 2010, where he is currently a Ph.D. candidate, specializing in circuit and embedded systems. In 2006, he studied as exchange student with the Department of Electronic & Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong. During the fall of 2009, he worked as an Engineer Intern with the Digital ASIC Group, Qualcomm Inc., San Diego, CA, USA, where he contributed to the low power design flow of a smart phone SOC. During the summer of 2012, he worked as a Ph.D. Intern with the Data Center Group, Cisco Systems Inc., San Jose, CA, USA, where he was involved in the FPGA emulation of a data center switch ASIC. His current research interests include circuit design and design optimization for STT-RAM and efficient DSP architectures for the sparse signal processing in compressivesensing applications.

Fang Gong received his B.S. degree from the Computer Science Department at the Beijing University of Aeronautics and Astronauticsin 2005. Also, he graduated from Computer Science Department at the Tsinghua University with M.S. degree in 2008. After that, he continues to work toward his Ph.D. degree in the Electrical Engineering Department at the University of California,Los Angeles. His research interests mainly focus on stochastic modeling and simulation for CAD. He also works on healthcare embedded system, parallel and distributed computing.

Puneet Gupta received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Delhi, New Delhi, India, in 2000, and the Ph.D. degree from the University of California at San Diego (UCSD), San Diego, in 2007. He co-founded Blaze DFM, Inc., Sunnyvale, CA (acquired by Tela, Inc.) in 2004 and was its Product Architect until 2007. He is currently a Faculty Member with the Department of Electrical Engineering, UCLA. He has authored over 80 papers, ten U.S. patents, and a book chapter. His current research interests include building high-value bridges across application–architecture–implementation–fabrication interfaces for loweredcost and power, increased yield, and improved predictability of integrated circuits and systems. Dr. Gupta was

Please cite this article as: L. Cheng, et al., Statistical timing and power analysis of VLSI considering non-linear dependence, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2013.12.004

a recipient of the NSF CAREER Award, the ACM/SIGDA Outstanding New Faculty Award, the SRC Inventor Recognition Award, the European Design Automation Association Outstanding Dissertation Award, and the IBM Ph.D. Fellowship. He has given tutorial talks at DAC, ICCAD, the International VLSI Design Conference, and the SPIE Advanced Lithography Symposium. He was the Technical Program Committee Member of DAC, ICCAD, ASPDAC, ISQED, ICCD, SLIP, and VLSI Design. He wasthe Program Chair of the IEEE DFM&Y Workshop in 2009 and 2010.

Lei He is a professor at electrical engineering department, University of California, Los Angeles (UCLA) and was a faculty memberat the University of Wisconsin, Madison between 1999 and 2002. He also held visiting or consulting positions with Cadence, Empyrean Soft, Hewlett-Package, Intel, and Synopsys, and was technical advisory board member for Apache Design Solutions and Rio Design Automation. Dr. He obtained Ph.D. degree in computer sciencefrom UCLA in 1999. His research interests include modeling and simulation, VLSI circuits and systems, and cyber physical systems. He has published on ebook and over 200 technical papers with 12 best paper nominations mainly from Design Automation Conference and International Conference on Computer-Aided Design and five best papers or best contribution awards including the ACM Transactions on Electronic System Design Automation 2010 Best Paper Award.