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

A R T I C L E   I N F O

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

Keywords:
Statistical modeling
VLSI
Yield analysis

A B S T R A C T

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 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. However being a linear operation, ICA often cannot completely remove the dependence between random variables.

In practice, the dependence between different variable sources is rarely linear (e.g., \( V_{th} \) is exponentially related to \( I_{eq} \)). 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

---

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 \( I(X_1;X_2) = \int f_{X_1},X_2 f_{X_1} f_{X_2} \log \left( \frac{f_{X_1},X_2}{f_{X_1} f_{X_2}} \right) dx_1 dx_2 \), where \( f_{X_1}(x_1) \) and \( f_{X_2}(x_2) \) are the marginal probability density function (PDF) of \( X_1 \) and \( X_2 \) respectively, and \( f_{X_1},X_2(x_1,x_2) \) 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.
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 $f(X_1, X_2) = X_1^2 + X_2^2$. We can see that in order to compute the mean of $f(X)$, 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 $l(X_1, X_2)$ exists, $X_1$ and $X_2$ are independent if and only if $l(X_1, X_2) = 0$. 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

$$
\rho_{ij} = \frac{E[X_1^i X_2^j] - E[X_1] E[X_2]^j}{\sqrt{E[X_1^i - E[X_1]^i]^2 E[X_2^j - E[X_2]^j]^2}}
$$

$\rho_{ij}$’s provide us with simple and good measures to estimate the impact of nonlinear dependence. Note that $-1 \leq \rho_{ij} \leq 1$ and that $\rho_{ij}$ is simply the linear correlation coefficient. In rest of this paper, we assume that the $\rho_{ij}$’s are known. In practice, $\rho_{ij}$ 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 $X = (X_1, X_2)^T$:

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

Then

$$
E[f(X)] = \sum_{i,j} a_{ij} m_{ij},
$$

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 $a_{ij} (m_{ij} - m_{0i} m_{0j})$. That is, the importance of the $ij$th joint moment depends on the coefficient of the $ij$th joint moment in the Taylor expansion, $a_{ij}$ and $m_{ij} - m_{0i} m_{0j}$. We define

$$
Q_{ij} = a_{ij} \cdot \sqrt{m_{20,i} \cdot m_{02,j}}.
$$

Then the mean can be expressed as

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

where $Q_{ij}$ 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 $Q_{ij}$. 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
\[ P = (P_1, P_2)^T = W \cdot X \]  
(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 \( P_2 \):
\[ f(X) = f(W^{-1} \cdot P) = \sum_{j} c_j P_1^j P_2^j. \]  
(7)

Because \( P \) is a linear combination of \( X \), it is easy to obtain the coefficients \( c_j \) from \( a_j \) 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
\[ Ef(X) = f(W^{-1} \cdot P) = \sum_{j} \rho_{P,ij} \cdot T_{ij}^p \]
\[ = \sum_{j} c_j P_1^j P_2^j \]
\[ = \sum_{j} \rho_{P,ij} \cdot T_{ij}^w \]
\[ T_{ij}^p = c_{ij} \cdot \sqrt{m_{2i,0} \cdot m_{0,2j}}, \]  
(8)
where \( m_{ij} \) is the \( ij \)th joint moment of \( P_1 \) and \( P_2 \), and \( \rho_{P,ij} \) is the \( ij \)th order correlation coefficient between \( P_1 \) and \( P_2 \). Since \( P \) is a linear combination of \( X \), it is easy to obtain joint moments \( m_{ij} \) and correlation coefficients \( \rho_{P,ij} \) can be easily calculated from the moments of \( X \)'s \( m_{ij} \) and the transform matrix \( W \). If we assume that these components are independent, i.e., we assume all the \( \rho_{P,ij} \) to be zero, then total error in mean estimation is
\[ \Delta_{\mu} = \sum_{i \geq 1, j \geq 0} \rho_{P,ij} \cdot T_{ij}^p. \]  
(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 \( P_1 \)'s as
\[ f^2(P_1, P_2) = \sum_{j} d_{ij} P_1^j P_2^j, \]  
(10)
where the coefficients \( d_{ij} \) can be calculated from \( c_j \)'s. Then we may estimate the error of the second order raw moment of \( f(\cdot) \)
\[ \Delta_2 = Ef^2(\cdot) - Ef^2 = \sum_{i \geq 1, j \geq 1} \rho_{P,ij} \cdot T_{ij}^p, \]  
(11)
\[ T_{ij}^p = d_{ij} \cdot \sqrt{m_{2i,0} \cdot m_{0,2j}}, \]  
(12)
where \( f \) is the function ignoring the dependence. Then the error of variance calculation if high order dependence is ignored is
\[ \Delta_{\sigma^2} = \Delta_2 - 2\mu' \Delta_{\mu} - \Delta_{\mu}^2 \]
\[ = \sigma_f^2 - \sigma_f^{2} \]
\[ = Ef[f(\cdot)^2] - Ef[f(\cdot)]^2 - Ef[f(\cdot)^2] + Ef[f(\cdot)]^2 \]
\[ = \Delta_2 - 2\mu' \Delta_{\mu} - \Delta_{\mu}^2, \]  
(13)
where \( \mu' \) is the mean calculated by ignoring the high order dependence and \( \Delta_{\mu} \) is the error of mean calculation which is calculated in (9). In practice \( \Delta_{\mu} \) is much smaller compared to \( \mu' \), therefore, we have
\[ \Delta_{\mu} \approx \Delta_2 - 2\mu' \Delta_{\mu}. \]  
(14)

With the error of variance, we may also calculate the error of standard deviation:
\[ \Delta_{\sigma} = \sqrt{\sigma^2 + \Delta_{\sigma^2} - \sigma' \approx \Delta_{\sigma^2} \frac{\sigma}{2\sigma}}, \]  
(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 \( \Delta_3 \) in a similar way to
\[ \Delta_3 = Ef[f^3(\cdot)] - Ef[f^3] = \sum_{i \geq 1, j \geq 1} \rho_{P,ij} \cdot T_{ij}^p, \]  
(16)
\[ T_{ij}^p = u_{ij} \cdot \sqrt{m_{2i,0} \cdot m_{0,2j}}, \]  
(17)
where the coefficients \( u_{ij} \) can be calculated from \( c_{ij} \). Then the error of skewness can be calculated as
\[ \Delta_r = \frac{Ef[f^3(\cdot) + \Delta_{\mu}] - Ef[f^3] + \Delta_{\mu}^3}{(\sigma + \Delta_{\sigma})^3}. \]  
(18)

4. Case study of statistical leakage and timing 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]
\[ P_{\text{leak}} = P_0 \cdot \exp(X_1 + c_1 X_1^2 + c_2 X_2^2 + c_3 X_3^2), \]  
(19)
where \( X_1 \) and \( X_2 \) are the variation sources, \( P_0 \) is the nominal leakage value, \( c_i \)'s are the sensitivity coefficients for variation sources \( X_i \) and \( X_2 \), respectively. Performing \( N \)th order Taylor expansion to the above equation, we have
\[ P_{\text{leak}} = P_0 \sum_{i,j=0}^{N} a_{ij} X_1^i X_2^j \]
\[ \approx P_0 \sum_{i,j=0}^{N} a_{ij} X_1^i X_2^j, \]  
(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
\[ P_{\text{chip,leak}} = \sum_{r \in C} p_{r, \text{leak}} \approx \sum_{i,j=0}^{N} q_{ij} X_1^i X_2^j, \]  
(21)
\[ q_{ij} = \sum_{r \in C} a_{r,ij}, \]  
(22)
where \( C \) is the set of all circuit elements in the chip and \( a_{r,ij} \) is the \( ij \)th order coefficient for the \( r \)th 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.
4.2.1. Gate delay

The delay of a single gate is usually expressed as a quadratic function of variation sources [42,28–28]

\[ D = a_{11}X_1^2 + a_{22}X_2^2 + 2a_{12}X_1X_2 + b_1X_1 + b_2X_2 + d_0. \]  

(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 \( X_i \).

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 can apply the method in Section 3 to estimate the error of the MAX operation. The error of the MAX operation depends on which variables. There are several algorithms to approximate the MAX operation. The error of the MAX operation is affected by the linear correlation between \( X_i \)’s and \( Xi \)’s and result after applying PCA (PCA). Then we calculate the error of mean, variance, and skewness variation.

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 (Model for Assessment of CMOS Technologies And Roadmaps) 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 \( 3 \sigma \) 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.**

<table>
<thead>
<tr>
<th>Gate</th>
<th>Fitting type</th>
<th>MC</th>
<th>After fitting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\mu$</td>
<td>$3\sigma$</td>
</tr>
<tr>
<td>Inv</td>
<td>Lin</td>
<td>5.12</td>
<td>1.12</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>5.12</td>
<td>1.12</td>
</tr>
<tr>
<td>Nand</td>
<td>Lin</td>
<td>9.29</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>9.29</td>
<td>1.95</td>
</tr>
<tr>
<td>Nor</td>
<td>Lin</td>
<td>12.32</td>
<td>2.78</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>12.32</td>
<td>2.78</td>
</tr>
<tr>
<td></td>
<td>PCA</td>
<td>$\mu$</td>
<td>$3\sigma$</td>
</tr>
<tr>
<td>Inv</td>
<td>Lin</td>
<td>5.09</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>5.15</td>
<td>1.02</td>
</tr>
<tr>
<td>Nand</td>
<td>Lin</td>
<td>9.18</td>
<td>1.82</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>9.36</td>
<td>1.89</td>
</tr>
<tr>
<td>Nor</td>
<td>Lin</td>
<td>12.11</td>
<td>2.51</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>12.41</td>
<td>2.69</td>
</tr>
<tr>
<td></td>
<td>PCA error</td>
<td>$\mu$</td>
<td>$3\sigma$</td>
</tr>
<tr>
<td>Inv</td>
<td>Lin</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Nand</td>
<td>Lin</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.03</td>
<td>0.09</td>
</tr>
<tr>
<td>Nor</td>
<td>Lin</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.03</td>
<td>0.16</td>
</tr>
</tbody>
</table>

*Note:* delay value is in ps.

**Table 2**

**Cell leakage.**

<table>
<thead>
<tr>
<th>Gate</th>
<th>Fitting type</th>
<th>MC</th>
<th>After fitting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\mu$</td>
<td>$3\sigma$</td>
</tr>
<tr>
<td>Inv</td>
<td>Lin</td>
<td>7.12</td>
<td>2.55</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>7.12</td>
<td>2.55</td>
</tr>
<tr>
<td>Nand</td>
<td>Lin</td>
<td>11.28</td>
<td>3.84</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>11.28</td>
<td>3.84</td>
</tr>
<tr>
<td>Nor</td>
<td>Lin</td>
<td>17.18</td>
<td>4.82</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>17.18</td>
<td>4.82</td>
</tr>
<tr>
<td></td>
<td>PCA</td>
<td>$\mu$</td>
<td>$3\sigma$</td>
</tr>
<tr>
<td>Inv</td>
<td>Lin</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Nand</td>
<td>Lin</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.03</td>
<td>0.09</td>
</tr>
<tr>
<td>Nor</td>
<td>Lin</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.03</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>PCA error</td>
<td>$\mu$</td>
<td>$3\sigma$</td>
</tr>
<tr>
<td>Inv</td>
<td>Lin</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Nand</td>
<td>Lin</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.03</td>
<td>0.09</td>
</tr>
<tr>
<td>Nor</td>
<td>Lin</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>0.03</td>
<td>0.16</td>
</tr>
</tbody>
</table>

*Note:* leakage value is in nW.
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 \( P = (p_1, p_2, \ldots, p_n) = W \cdot X \)

---

Table 3

<table>
<thead>
<tr>
<th>Bench mark</th>
<th>SSTA type</th>
<th>MC</th>
<th>SSTA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>μ</td>
<td>3σ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>μ</td>
<td>3σ</td>
</tr>
<tr>
<td>C17 Lin</td>
<td>42.2</td>
<td>14.7</td>
<td>0.10</td>
</tr>
<tr>
<td>C17 Quad</td>
<td>42.2</td>
<td>14.7</td>
<td>0.10</td>
</tr>
<tr>
<td>C499 Lin</td>
<td>320.2</td>
<td>105.5</td>
<td>0.14</td>
</tr>
<tr>
<td>C499 Quad</td>
<td>320.2</td>
<td>105.5</td>
<td>0.14</td>
</tr>
<tr>
<td>C880 Lin</td>
<td>674.4</td>
<td>221.3</td>
<td>0.12</td>
</tr>
<tr>
<td>C880 Quad</td>
<td>674.4</td>
<td>221.3</td>
<td>0.12</td>
</tr>
<tr>
<td>C3540 Lin</td>
<td>1241.2</td>
<td>413.2</td>
<td>0.13</td>
</tr>
<tr>
<td>C3540 Quad</td>
<td>1241.2</td>
<td>413.2</td>
<td>0.13</td>
</tr>
<tr>
<td>C7522 Lin</td>
<td>1919.3</td>
<td>635.4</td>
<td>0.13</td>
</tr>
<tr>
<td>C7522 Quad</td>
<td>1919.3</td>
<td>635.4</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Note: delay value is in ps.

---

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

\[
W = \arg \min_{\Delta_{\mu} = 0} \Delta,
\]

\[
\Delta = \Delta_{\mu} + \epsilon \Delta_{\gamma},
\]

\[
\Delta_{\mu} = \mu_f - \mu_f',
\]

\[
\Delta_{\sigma} = \sigma_f - \sigma_f',
\]

\[
\Delta_{\gamma} = \gamma_f - \gamma_f',
\]

where \( \mu_f, \sigma_f, \) and \( \gamma_f \) 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; \( \epsilon \) is the weight factor for the skewness error. Since \( f(X) \) is a polynomial function of \( X \), similar to (9), (15), and (18), \( \mu_f, \sigma_f, \) and \( \gamma_f \) 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 joint moments of \( X_i \)'s. Hence, the error \( \Delta \) 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\footnote{This is especially useful in cases where \( \mu = 0 \) has no solution.}

\[
W = \arg \min (\Delta_{\mu} + \epsilon_1 \Delta_{\sigma} + \epsilon_2 \Delta_{\gamma}).
\]

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

<table>
<thead>
<tr>
<th>Gate</th>
<th>Fitting type</th>
<th>MC Stat leak error</th>
<th>Stat leak</th>
<th>Stat leak error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \mu )</td>
<td>( 3\sigma )</td>
<td>( \gamma )</td>
</tr>
<tr>
<td>C17</td>
<td>Lin</td>
<td>430.2</td>
<td>120.3</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>430.2</td>
<td>120.3</td>
<td>0.28</td>
</tr>
<tr>
<td>C499</td>
<td>Lin</td>
<td>9.14</td>
<td>3.21</td>
<td>0.32</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>9.14</td>
<td>3.21</td>
<td>0.32</td>
</tr>
<tr>
<td>C880</td>
<td>Lin</td>
<td>22.3</td>
<td>7.58</td>
<td>0.34</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>22.3</td>
<td>7.58</td>
<td>0.34</td>
</tr>
<tr>
<td>C3540</td>
<td>Lin</td>
<td>89.2</td>
<td>29.23</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>89.2</td>
<td>29.23</td>
<td>0.41</td>
</tr>
<tr>
<td>C7522</td>
<td>Lin</td>
<td>162.2</td>
<td>56.21</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>Quad</td>
<td>162.2</td>
<td>56.21</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Note: leakage value is in \( \mu W \) for C17, and in \( mW \) for others.
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 \( f(X_1, X_2) = X_1^2 + X_1X_2 + X_2^2 \). As discussed in Section 1, the correct value is \( E(f(c)) = 9 \). 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 \( E(f(c)) = 5 \). If we apply fast kernel ICA to obtain independent components, we will have \( E(f(c)) = 5.78 \). If we use FCA, we have \( E(f(c)) = 8.54 \). 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 \( L_{\text{eff}} \) and \( V_{\text{th}} \), and we only consider inter-die variation for the variation sources. We generate dependent variation samples of \( L_{\text{eff}} \) and \( V_{\text{th}} \) 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 \( L_{\text{eff}} \) and \( V_{\text{th}} \). 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 \( L_{\text{gate}} \) and \( N_{\text{bulk}} \), 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 Master4 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.[3] 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 power 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.

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 \( L_{\text{eff}} \) and \( V_{\text{th}} \) 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
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 (\(L_{\text{eff}}\) and \(V_{\text{th}}\) for all 4 transistors). Notice that PCA and ICA provide the same transfer matrix for \(L_{\text{eff}}\) and \(V_{\text{th}}\) for all transistors, however because FCA tries to handle 8 variation sources together, it may provide different transfer matrices for different transistors.

Table 10 compares 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 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 array. We assume that the variation of all SRAM cells in the

![Fig. 1. Redundancy for Non-ECC scheme to achieve 99% yield rate.](image1)

![Fig. 2. SNM PDF comparison.](image2)
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 percentile 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 percentile 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.

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.

References

[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
Lei He is a professor at electrical engineering department, University of California, Los Angeles (UCLA) and was a faculty member at the University of Wisconsin, Madison between 1999 and 2002. He also held visiting or consulting positions with Cadence, Empyrean Soft, Hewlett-Packard, 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 science from 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.