Statistical Decoupling Capacitance Allocation By Efficient Numerical Quadrature Method

Thom Jefferson A. Eguia, Ning Mi, Sheldon X.-D. Tan
Department of Electrical Engineering
University of California, Riverside, CA 92521, USA
E-mail: stan@ee.ucr.edu

Abstract—In this paper, we propose a novel statistical decap allocation method to reduce the voltage drop noise in the presence of variational leakage current sources. The new method can derive the closed form of decoupling capacitance (decaps) in terms of variational parameters from the variational leakage currents. It treats the deterministic decap method as a black box and can work with any existing simulation-based methods. The new method employs the orthogonal polynomials to represent the variational gate leakages in a closed form first. Decap distributions are then computed by a fast multi-dimentional Gaussian quadrature method with sparse grid technique. Experimental results show that the proposed method can deliver order of magnitudes speedup over the Monte Carlo method with almost the same accurancy.

I. Introduction

Reliable on-chip power delivery is one of the major concerns for 90nm and below VLSI technology. This situation becomes worse as technology continues to scale to 45nm and below due to increasing process-induced variability [19]. The process induced variations manifest themselves at different levels (wafer level, die-level and within a die) and are caused by different sources (lithograph, materials, aging etc) [4, 15]. Some of the variations are systematic, like those caused by chemical mechanical polishing (CMP), while some are purely random like the doping density of impurities and edge roughness. As technology moves to 65nm and comes near to 45nm, variation will become more and more pronounced for both systemic and random components.

A major source of variation comes from leakage, which are exponential functions of the threshold voltages. They are typically expressed as random variables due to the variations of channel length and gate oxide thickness. As a result, the leakage currents are very sensitive to those parameter variations. It was shown in [12] that leakage variations for $90nm$ can be 20 *times*. Based on the ITRS 2006 [1], the leakage power accounts for more than 60% at $45nm$ technology. This has many consequences in chip design, especially in the design of power grid networks. In this paper, we model these leakage current sources as random variables of log-normal distribution due to threshold voltage variation, which are modeled as random variables of Gaussian (normal) distribution. [10, 14].

The effective way to reduce noise in the package is to add decoupling capacitors (decaps) into the power grid networks [6, 17]. For VLSI design, the inductive wire impedance can no longer be ignored at high frequencies, and the on-chip noise becomes more profound as inductance scales poorly with wire sizing. As a result, on-chip decaps are indispensable for robust on-chip power supply and many approaches have been proposed in the past [3,8,13,20,23]. On the other hand, on-chip decaps are usually manufactured as gate capacitance of transistors. As the supply voltage continues scaling, leakage currents due to reduced threshold voltage and dielectric leakage prevent excessive use of decaps [2]. Thus, the economic use of on-chip decaps is relatively important, especially in the presence of leakage current variations.

Some existing works were proposed for allocating decaps considering process variations [7,21]. In [7], the orthogonal polynomial base method is used for calculating the mean and variance of a node voltage drop, therefore obtaining 3σ range of the voltage drop. Based on the range, statistical decap budget is calculated using sensitivity-based methods. However, it is very time consuming to calculate 3σ range of voltage drop specially when more random variables are considered. Also, no closed form in terms of parameter variables is given for the final decap budget. In [21], the authors proposed to optimize decap budgeting by using a successive linear programming method after obtaining the current variational ranges modeling process and operational variations. The method may not be very scalable as programming-based solutions are very expensive for a larger number of variables.

In this paper, we propose a novel statistical decap optimization algorithm considering the effect of process-induced leakage current variations. The new method can provide a general framework to derive the decap budgets for each node in a closed form in terms of the variational parameters, such as the channel length, threshold voltage, etc. It can accommodate various spatial correlations via orthogonal decomposition. The new method employs orthogonal polynomials to represent the variational gate leakages in a closed form first. Then the decap distributions are computed by a fast multi-dimensional Gaussian quadrature method with sparse grid technique. In each sampling of the Gaussian quadrature, the sensitivity-based conjugate gradient method is used in this paper to compute the decap values. But it is understood that the new method can work with any existing decap optimization methods. The new

The work is funded in part by NSF CAREER Award No. CCF-0448534, in part by NSF grant under No. OISE-0623038 and in part by National Natural Science Foundation of China (NSFC) grant under No. 60828008.

method can reuse all existing deterministic decap allocation methods, and is very general and scalable. Experimental results show that the new algorithm is much faster than Monte Carlo based method with similar accuracy.

The rest of the paper is organized as follows: Section 2 presents the statistical decap allocation problem and models. Section 3 introduces our leakage-dominated variation modeling for computing power grid voltage variations. The orthogonal polynomial chaos based stochastic simulation methods are reviewed in section 4. Also, the variational leakage current model and traditional way to do statistical analysis are introduced in this section. In section 5, we present our new Gaussian Quadrature and Sparse grid based method for the variational decap budget problem subjecting to log-normal leakage current variations. Finally, we demonstrate experimental results in section 6 to validate our method, with conclusions and future works in section 7.

II. Statistical Decap Allocation Problem

We first present the deterministic decap optimization problem, followed by the new problem considering process variations.

In decap optimization, we first need to represent the violation area for a power grid, which is calculated as:

$$a_j = \int_0^T max(v_j(t) - V_{th}, 0)dt \tag{1}$$

Here, V_{th} represents the threshold voltage and $v_j(t)$ is the node voltage at time t. This represents the area above the threshold voltage as the ground bounces for ground networks as shown in fig. 1.

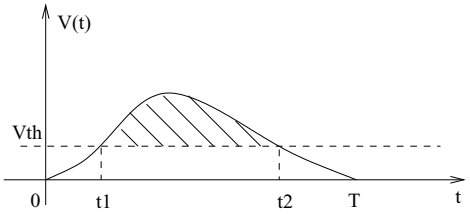

Fig. 1. Illustration of violation area

Then, an optimization method is used to minimize the violation area to get decap budget subject to constraint, which is the layout space around corresponding node, as:

$$\min \sum_{j=1}^{s} a_j(c_1, ..., c_n) \tag{2}$$

subject to constraints

$$c_i \leq d_i, di \geq 0 \tag{3}$$

s is the number of violation nodes and $c_1, ..., c_n$ are the capacitance to be added. n is the number of decap nodes and d_i is the maximum decap allowed at node i. Some optimization methods such as conjugate gradient (CG) [8, 23], which still remains as the most reliable and accurate method, can be used to solve the decap allocation problem.

In this paper, we assume that we have a number of independent (uncorrelated) transformed orthonormal Gaussian random variables $\xi_i(\theta)$, where $i = 1, ..., n$. They serve to model the device threshold voltage variations or channel length variations. The spatial correlation in the intradie variation can be processed by using Principal Component Analysis (PCA) to transform correlated Gaussian random variables to independent Gaussian random variables, which is suitable for either strongly correlated dependent or weakly correlated random variables. Also, Independent Component Analysis(ICA) can be used to transform correlated non-Gaussian random variables to uncorrelated Gaussian random variables.

Let Θ denote the process sampling space. Let $\theta \in \Theta$, $\xi_{id} : \theta \rightarrow R$ denote a normalized Gaussian variable, and $\xi(\theta) = [\xi_{1d}(\theta), ..., \xi_{rd}(\theta)]$ is a vector of r correlated Gaussian variables. After an orthogonal transformation operation, we obtain the independent random variable vectors $\xi = [\xi_1, ..., \xi_n]$. In general, $n \leq r$. Therefore, given the process variations, the Modified Nodal Analysis (MNA) of the power grid network becomes

$$Gv(t, \xi(\theta)) + C\frac{dv(t, \xi(\theta))}{dt} = I(t, \xi(\theta)) \tag{4}$$

Note that the input current vector, $I(t, \xi(\theta))$, consists of both deterministic and stochastic components.

The problem we need to solve is how to efficiently calculate the mean and variances of decap values (budgets) due to variational current sources. For given current inputs, decap values are computed to prevent the *IR* drops from exceeding the user-defined thresholds. With those statistical decap information, we may add the decaps in a consecutive manner, means plus three standard deviations for instance, to ensure that the *IR* drop constraints are satisfied with higher possibility.

III. Modeling of On-Chip Power Delivery Voltage Variations

In order to consider the statistical effect on decap budget estimation, the voltage variation induced by process-related current variation should first be dealt with. A typical way of modeling this phenomena is to introduce statistical current sources, which depend on circuit parameters such as channel length, width, and gate oxide thickness, etc.

Among all factors, the sub-threshold leakage current variation is most important since it has a rapid increasing rate (about 5X-10X increase per technology generation [5]). In addition, it is highly sensitive to threshold voltage V_{th} variations, due to the exponential relationship between sub-threshold current I_{off} and threshold voltage V_{th} as shown below [22],

$$I_{off} = I_{s0}e^{\frac{V_{gs}-V_{th}}{nV_T}}(1 - e^{\frac{-V_{ds}}{V_T}}) \tag{5}$$

where I_{s0} is a constant related to the device characteristics, V_T is the thermal voltage, and n is a constant.

In this paper, we only treat the current sources represented in the circuit as having leakage current stochastic perturbations. The leakage current is modeled as log-normal variation, due to the channel length variable, whose variation is modeled as Gaussian (normal) distribution. Other variation factors, such as wire length on MNA matrix will be considered in the future.

We first assume we have a number of (correlated or uncorrelated) ortho-normal random Gaussian variables $\xi(\theta), i = 1, ..., n$, which actually model the channel length and the device threshold voltage variations. Then, we consider leakage current variation and other circuit variation together in the power grid circuit.

We randomly generate $I(t, \xi(\theta))$, which is based on the log-normal distribution, solve (4) in time domain for each sampling and compute the mean and variances based on sufficient samplings. Obviously, MC will be computationally expensive. However, MC will give the most reliable results and is the most robust and flexible method.

IV. Statistical Stochastic Spectrum Analysis

In this section, we briefly review the orthogonal polynomial chaos (PC) based stochastic methods.

A. Review of Hermite polynomial chaos

In the following, a random variable $\xi(\theta)$ is expressed as a function of θ, which is the random event. Hermite PC utilizes a series of orthogonal polynomials (with respect to the Gaussian distribution) to facilitate stochastic analysis [24]. These polynomials are used as orthogonal basis to decompose a random process in a similar way that sine and cosine functions are used to decompose a periodic signal in a Fourier series expansion.

Given a random variable $v(t, \xi)$ with certain variance, where $\xi = [\xi_1, \xi_2, ...\xi_n]$ denotes a vector of ortho-normal Gaussian random variables with zero mean, the random variable can be approximated by a truncated Hermite PC expansion as follows: [9]

$$v(t, \xi) = \sum_{k=0}^{P} a_k H_k^n(\xi) \tag{6}$$

where n is the number of independent random variables, $H_k^n(\xi)$ is n-dimensional Hermite polynomials, and a_k are the deterministic coefficients. The number of terms P is given by

$$P = \sum_{k=0}^{p} \frac{(n-1+k)!}{k!(n-1)!} \tag{7}$$

where p is the order of the Hermite PC. If only one random variable is considered, the one-dimensional Hermite polynomials are expressed as follows:

$$H_0^1(\xi) = 1, H_1^1(\xi) = \xi, H_2^1(\xi) = \xi^2 - 1, H_3^1(\xi) = \xi^3 - 3\xi, ... \tag{8}$$

Hermite polynomials are orthogonal with respect to Gaussian weighted expectation (the superscript n is dropped for simple notation):

$$< H_i(\xi), H_j(\xi) > = < H_i^2(\xi) > \delta_{ij} \tag{9}$$

where δ_{ij} is the Kronecker delta and $< *, * >$ denotes an inner product defined as:

$$< f(\xi), g(\xi) > = \frac{1}{\sqrt{(2\pi)^n}} \int f(\xi) g(\xi) e^{-\frac{1}{2}\xi^T \xi} d\xi \tag{10}$$

Thus, the coefficient, a_k, is found by a projection operation onto the Hermite PC basis:

$$a_k(t) = \frac{< v(t, \xi), H_k(\xi) >}{< H_k^2(\xi) >}, \ \forall k \in \{0, ..., P\}. \tag{11}$$

B. Hermite PCs for log-normal leakage current variations

In this part, we will present the leakage current variation considering channel length or threshold voltage variation, which is expressed as a random variable of log-normal distribution.

Let $g(\xi)$ be the Gaussian random variable, denoting threshold voltage or device channel length. Let $l(\xi)$ be the random variable obtained by taking the exponential of $g(\xi)$

$$l(\xi) = e^{g(\xi)}, \ g(\xi) = ln(l(\xi)) \tag{12}$$

Obviously, for the MOS device leakage current equation (5), leakage current, $I_{off} = cI_l(V_{th}) = ce^{-V_{th}}$, where the leakage component $I_l(V_{th})$ is a log-normal random variable. Express the mean and the variance of $g(\xi)$ as μ_g and σ_g^2, then the mean and variance of $l(\xi)$ are

$$\mu_l = e^{(\mu_g + \frac{\sigma_g^2}{2})} \tag{13}$$

$$\sigma_l^2 = e^{(2\mu_g + \sigma_g^2)}[e^{\sigma_g^2} - 1] \tag{14}$$

respectively. For a general Gaussian variable $g(\mathbf{x})$, it can always be represented as

$$g(\mathbf{x}) = \sum_{i=0}^{n} \xi_i g_i \tag{15}$$

where ξ_i are orthonormal Gaussian variables. i.e. $< \xi_i, \xi_j >= \delta_{ij}$, $< \xi_i >= 0$ and $\xi_0 = 1$. Note that such form can always be obtained by using Karhunen-Loeve orthogonal expansion method [9].

V. New method for decap allocation problem: QuadDecap

The new method is based on the efficient multi-dimensional numerical Gaussian quadrature. We first review the numerical Gaussian quadrature method, followed by the improved Smolyak quadrature.

A. Gaussian quadrature technique

The Gaussian quadrature method is an efficient numerical method to compute the definite integral of a function [11]. We apply it to compute the coefficients $a_k(t)$

in (11). We review the method based on the Hermite polynomial below.

Our goal is to compute the integral equation $< x(\xi), H_j(\xi) >$ numerically. In this case, this problem boils down to the one-dimensional numerical quadrature problem based on Hermite polynomials [18]. Specifically, for Hermite polynomials, we have

$$< x(\xi), H_k(\xi) > = \frac{1}{\sqrt{(2\pi)}} \int x(\xi) H_k(\xi) e^{-\frac{1}{2}\xi^2} d\xi \quad (16)$$

$$\approx \sum_{i=0}^{P} x(\xi_i) H_i(\xi_i) w_i \quad (17)$$

Here, $\xi = \{\xi\}$, contains only one random variable. ξ_i and w_i are Gaussian Hermite quadrature abscissas (quadrature points) and weights.

The Quadrature rule states that if we select the roots of Pth Hermite Polynomial as the quadrature points, the quadrature is exact for all polynomials of degree $2P-1$ or less for (16). This is called (P-1)-level accuracy of the Gaussian-Hermite quadrature.

For multiple random variables, a multi-dimensional quadrature is required. The traditional way to compute a multi-dimensional quadrature is to use a direct tensor product based on one dimensional Gaussian Hermite quadrature abscissas and weights [16]. With this method, the number of quadrature points needed for n dimension (variables) and Pth level is about $(P+1)^n$, which is well known as the curse-of-dimensionality.

B. Sparse grid technique

Smolyak quadrature [16] is used as an efficient method to reduce the number of quadrature points (also called sparse grid quadrature). Let's define one-dimensional sparse grid quadrature point set $\Theta_1^P = \{\gamma_1, \gamma_2, ..., \gamma_P\}$, which uses $P+1$ points to achieve degree $2P+1$ of exactness. The level-P sparse grid for n-dimensional quadrature chooses points from the following set:

$$\Theta_n^P = \bigcup_{P+1 \le |\vec{i}| \le P+n} (\Theta_1^{i_1} \times ... \times \Theta_1^{i_n}) \quad (18)$$

where $|\vec{i}| = \sum_{j=1}^{n} i_j$. The corresponding weight is:

$$w_{j_{i_1}...j_{i_n}}^{i_1...i_n} = (-1)^{P+n-|\vec{i}|} \begin{pmatrix} n-1 \\ n+P-|\vec{i}| \end{pmatrix} \prod_m w_{j_{i_m}}^{i_m} \quad (19)$$

where $\begin{pmatrix} n-1 \\ n+P-|\vec{i}| \end{pmatrix}$ is a combination number and w is the weight for the corresponding quadrature points. It was shown that interpolation on a Smolyak grid ensures an error bound for the mean-square error [16]

$$|E_P| = O(N_P^r (log N_P)^{(r+1)(n-1)}),$$

where N_P is the number of quadrature points and k is the order of the maximum derivative that exist for the delay function. The number of quadrature points increases as $O(\frac{n^P}{(P)!})$.

It can be shown that a sparse grid of at least level P is required for an order P representation. The reason is that the approximation contains order P polynomials for both $x(\xi)$ and $H_j(\xi)$ for some j. So there exists $x(\xi)H_j(\xi)$ with order $2P$, which requires a sparse grid of at least level P with degree $2P+1$ of exactness.

Therefore, level 2 and level 1 sparse grids are required for quadratic and linear models, respectively. The number of quadrature points is about $2n$ and $2n^2$ for the linear and the quadratic model respectively. The time cost is about the same as the Taylor-conversion method, while keeping the accuracy of homogenous chaos expansion.

In addition to the sparse grid technique, we also employ several accelerating techniques summarized as follows:

- When n is too small, the number of quadrature points for sparse grid may be larger than that of direct tensor product of Gaussian quadrature. For example, if there are only 2 variables, the number is 5 and 14 for level 1 and 2 sparse grid, compared to 4 and 9 for direct tensor product. In this case, the sparse grid will not be used.
- The set of quadrature points (18) may contain the same points with different weights. For example, the level 2 sparse grid for 3 variables contain 4 instances of the point (0,0,0). Combining these points by summing the weights reduces the computation of $x(\vec{\gamma}_i)$ by 3 times.

C. New statistical decap optimization algorithm

After introducing all the important pieces of related works, we are now ready to present our new algorithm, called *QuadDecap*, for solving decap allocation considering leakage variation. Fig. 2 is the flowchart of the proposed algorithm.

Algorithm: QUADDECAP

Input: A power grid network with variational leakage currents $I(t)$

Output: the HPC coefficients of decap values of specific nodes, $d(\xi)$

1. Generate the one-dimensional quadrature point set of second order Θ_1^2 and weight set w.
2. Generate the n-dimensional Smolyak quadrature point sets of second order Θ_n^2 and corresponding weight set w_n.
3. For i = 1 to size(Θ_n^2)
4. Perform decap budget calculation (using CG method) based on variational leakage current $I(t, \gamma_i)$.
5. end
6. Compute the coefficients of Hermite polynomials for the decap values $d(\xi)$

Fig. 2. The proposed QuadDecap algorithm.

As mentioned in Section 3, the leakage variation follows lognormal distribution. The leakage current is written as:

$$I(t, \xi(\theta)) = I_d(t) + I_v(t, \xi(\theta)) \quad (20)$$

$I_d(t)$ represents the deterministic currents. $I_v(t,\xi(\theta)$ represents the variational part when considering threshold voltage variation or effective channel length variables, which is model as Gaussian(normal) distribution.

$$I_v(t,\xi(\theta)) \sim e^{g(t,\xi)} = e^{\sum_{j=0}^{n} g_j(t)\xi_j} \tag{21}$$

Using Gaussian Quadrature and Smolyak Quadrature schemes, we can obtain sampling points as in step. 1 and step. 2 as

$$\gamma_i = \{\gamma_1^{i_1}, \gamma_1^{i_2}, ..., \gamma_1^{i_n}], \gamma_i \in \Theta_n^2 \tag{22}$$

The decap budget in each sample $d\xi$ is calculated based on these sampling points, $\gamma_i \in \Theta_n^2$, using the conjugate gradient(CG) method [8, 23]. The results of $d\gamma_i$ can be used for calculating the Hermite Polynomial coeffients as:

$$t_k = \frac{< d(\xi), H_i(\xi) >}{< H_i^2(\xi) >} = \frac{\sum_{i=0}^{M} d(\gamma_i) H_k(\gamma_i) w_i}{< H_i^2(\xi) >} \tag{23}$$

Therefore, the decap values can be written in terms of Hermite Polynomial bases as

$$\begin{aligned} d(\xi) &= \sum_{k=0}^{N} t_k H_k(\xi) \\ &= t_0 + \sum_{k=1}^{n} t_k \xi_k + \sum_{k=1}^{2n} t_{n+k}(\xi_k^2 - 1) + \\ &\quad \sum_{k=1}^{n-1}\sum_{j=k}^{n} t_{2n+k+j}\xi_k\xi_j \end{aligned} \tag{24}$$

With the Hermite Polynomial form of variational decap budget, some statistical parameters, such as mean and variance, can be calculated as

$$\begin{aligned} E(v(t,\xi)) &= v_0(t) \\ Var(v(t,\xi)) &= \sum_{k=1}^{n} t_k var(\xi_k) + \sum_{k=1}^{2n} t_{n+k} var(\xi_k^2 - 1) + \\ &\quad \sum_{k=1}^{n-1}\sum_{j=k}^{n} t_{2n+k+j} var(\xi_k\xi_j) \\ &= \sum_{k=1}^{n} t_k^2 + 2\sum_{k=1}^{n} t_{n+k}^2 + \sum_{k=1}^{n-1}\sum_{j=k}^{n} t_{2n+k+j}^2 \end{aligned} \tag{25}$$

VI. **Experimental results**

This section summarizes the experimental results of the proposed QuadDecap method for a number of power grid networks. All of the proposed methods have been implemented in Matlab, Perl and C/C++. All the experimental results are carried out in a Linux system with dual Intel Xeon CPUs with 3.06Ghz and 1GB memory.

In order to determine the accuracy of QuadDecap, we compared the mean and variance of its decap budgets to those of the Monte Carlo method. Furthermore, we compared the distribution and cumulative distribution curves of the decap budgets on an arbitrary node within a specified circuit. Lastly, we compared the run times of both methods, and calculated the speed up of the QuadDecap method over the Monte Carlo method.

In the MC method, we first generate each sample netlist file by modifying the current source value in the original circuit netlist. Each individual current source in PWL format will be added by log-normal based leakage current variations. We assume that the DC value of leakage current is assumed to be 300 μA in this experiment. We adopt this value based on TSMC 180 nm technologies. The leakage currents are 530 μA and -268 μA for N-channel and P-channel respectively. With this DC value as mean, we generate log-normal leakage current distribution with a certain standard deviation. The final current value will be the addition of deterministic PWL value and the log-normal leakage current random value at the specific sample point. Then each file will be processed to obtain the decap budget at a specified node. In QuadDecap, the decap samples are generated using the same Monte Carlo method. However, the sampling points are generated from a sparse grid, which are used as the random variables to generate the log-normal sample value. The resulting decap sample values are then collected and placed in the QuadDecap method, in which the coefficients are calculated.

A large number of optimization runs will be performed in order to derive an accurate decap budget distribution spectrum for both QuadDecap and Monte Carlo methods. For the sake of practical run-time from MC approach, we select relatively small circuit sizes in our experiment. In addition, the optimization parameter for decap budget is manually set such that the total violation node number is within 10% of total nodes.

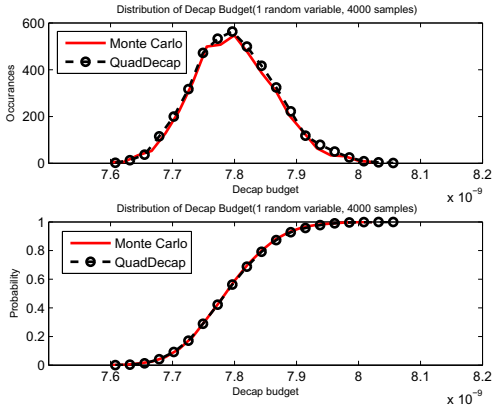

Fig. 3. Distribution curves of decap budget at node 2 of *gridrc_5* with one variable of variance 0.1

Fig. 3 gives the decap budget distribution and cumulative distribution curves of a specified node within the circuit *gridrc_5* of 225 nodes.

The QuadDecap and Monte Carlo results of 4000 samples are plotted. The dash-circle line represents the QuadDecap results, while the solid line represents the Monte Carlo results. The test was performed considering one random variable with a variance of 0.1. The graphs show a close match between the respective curves.

In the case of two random variables, we geometrically divide the circuit into two sections. Following the same procedure as in one variable MC simulation, we add stochastic input with variances of 0.1 and 0.2 into each respective section, and then calculate the decap budget for each individual circuit. We follow the same circuit division procedure when considering ten random variables, adding variances of 0.1 into each section. Similarly, the QuadDecap procedure considering multiple random variables follows the quad decap procedure considering that of one. The variance for each random variable in the two variable tests is 0.1 and 0.2, while the ten variable tests have variances of 0.1.

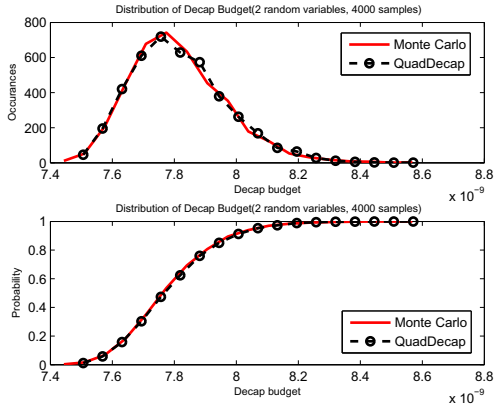

Fig. 4. Distribution curves of decap budget at node 2 of *gridrc_5* with two variables at variances of 0.1 and 0.2 respectively

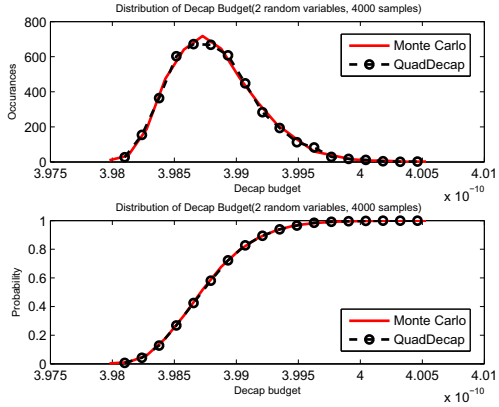

Fig. 5. Distribution curves of decap budget at node 2 of *gridrc_25* with two variables at variances of 0.1 and 0.2 respectively

Fig. 4 gives the decap budget distribution and cumulative distribution curves of a specified node within the circuit *gridrc_5* of 225 nodes. Similarly, Fig. 5 gives the decap budget distribution and cumulative distribution curves of a specified node within the circuit *gridrc_25* of 5625 nodes. The QuadDecap and Monte Carlo results of 4000 samples are plotted. The dash-circle line represents the QuadDecap results, while the solid line represents the Monte Carlo results. Both tests were performed considering two random variables with variances of 0.1 and 0.2 respectively. The graphs show a close match between the respective curves.

TABLE I

MEAN AND VARIANCE COMPARISON BETWEEN QUADDECAP AND THE MONTE CARLO METHOD, 1 VARIABLE WITH VARIANCE 0.1, 4000 SAMPLES

#node	μ_MC	μ_QD	μ_err(%)	σ_MC	σ_QD	σ_err(%)
225	7.80e-9	7.80e-9	4.40e-2	4.40e-5	4.40e-5	7.86e-2
3300	5.12e-11	5.12e-11	5.96e-3	2.01e-11	2.02e-11	0.14
5625	3.98e-10	3.98e-10	1.72e-3	2.35e-10	2.30e-10	2.09
14400	-	8.01e-13	-	-	1.41e-16	-

TABLE II

MEAN AND VARIANCE COMPARISON BETWEEN QUADDECAP AND THE MONTE CARLO METHOD, 2 VARIABLES WITH VARIANCE 0.1 AND 0.2, 4000 SAMPLES

#node	μ_MC	μ_QD	μ_err(%)	σ_MC	σ_QD	σ_err(%)
225	7.82e-9	7.82e-9	1.49e-3	2.40e-4	2.37e-4	1.10
3300	5.12e-11	5.12e-11	2.40e-3	1.04e-10	1.06e-10	1.37
5625	3.98e-10	3.98e-10	2.13e-5	1.23e-9	1.24e-9	1.17
14400	-	8.01e-13	-	-	7.36e-16	-

Table. I shows the mean and variance values of a specified node on different sized circuits. μ_MC and μ_QD represent the mean values for the Monte Carlo and quad decap respectively, while $\mu\_err(\%)$ represents the relative error between the two values in percentage. σ_MC and σ_QD represent the variance values for the Monte Carlo and quad decap respectively, while $\sigma\_err(\%)$ represents the relative error between the two values in precentage. The test was performed considering one random variable with a variance of 0.1, and was conducted for 4000 sample runs. Similarly, Table. II shows the mean and variance values for the same set of circuits considering two random variables with variances of 0.1 and 0.2 respectively. Lastly, Table. III shows the mean and variance values of two different sized circuits considering 10 random variables with variances of 0.1. All three tests show a small percentage error between the Monte Carlo and QuadDecap mean values, all of which are below one percent. Despite being larger, the variance error results are still relatively small at values no greater than four percent. These results exhibit the scalability of the QuadDecap method in terms of accuracy among different sets of random variables and variances.

Table. IV shows the speed up of QuadDecap over Monte Carlo for 4000 sample runs on different sized circuits with 2

TABLE III

MEAN AND VARIANCE COMPARISON BETWEEN QUADDECAP AND THE MONTE CARLO METHOD, 10 VARIABLES WITH VARIANCES OF 0.1, 4000 SAMPLES

#node	μ_MC	μ_QD	μ_err(%)	σ_MC	σ_QD	σ_err(%)
225	7.83e-9	7.84e-9	7.80e-2	5.17e-4	5.14e-4	0.71
3300	5.11e-11	5.11e-11	3.30e-3	2.41e-10	2.32e-10	3.52

TABLE IV

RUN TIME COMPARISON BETWEEN QUADDECAP AND THE MONTE CARLO METHOD, 2 VARIABLES WITH VARIANCE 0.1 AND 0.2, 4000 SAMPLES

Ckt	#node	MC(s)	QuadDecap(s)	Speedup
gridrc_5	225	2.52e4	109.05	231.40
gridrc_12	3300	9.14e4	392.14	232.65
gridrc_25	5625	1.30e5	624.74	207.29
gridrc_40	14400	-	7.79e3	-

TABLE V

RUN TIME COMPARISON BETWEEN QUADDECAP AND THE MONTE CARLO METHOD, 10 VARIABLES WITH VARIANCES OF 0.1, 4000 SAMPLES

Ckt	#node	MC(s)	QuadDecap(s)	Speedup
gridrc_5	225	2.90e4	788.62	36.82
gridrc_12	3300	9.50e4	4.36e3	21.77

random variables (variances are 0.1 and 0.2 repsectively). *MC* and *Quad Decap* represent the run times of Monte Carlo and QuadDecap methods respectively. Lastly, Table. V compares the run times considering 10 random variables with variances of 0.1. The total run time recorded for the QuadDecap method includes the generation of quadrature points, the Monte Carlo runs which generate the decap samples, as well as the calculation of the decap coefficients. The two variable test shows an approximate speed up of 200 times. Meanwhile, the ten variable tests show a speed up of approximately 20 times. However, it is observed that the speed up degrades, as the run time for decap sampling increases with the size of the circuit as well as with the number of variables. This is even more evident in the ten variable tests where more sample points are required. The speed up, however, should continually increase if the amount of sample runs for the multi-variable and large circuit tests also increase. This would show an increase in the run time of the Monte Carlo method, while the QuadDecap, whose number of decap samples remain the same, should maintain an approximately similar run time. Since it is impractical to run bigger ciruits at such large sample numbers, the amount of runs remained fixed at 4000.

VII. **Conclusions and Future Works**

In this paper, we have proposed a framework of statistical decap budget analysis. The approach is based on an orthogonal representation of both variational leakage currents and final decap budgets. We applied efficient numerical multi-dimensional Gaussian quadrature to compute the coefficients of the decap budgets in orthogonal polynomials. It is very general and can work with any existing simulation-based decap optimization method. The resulting method shows the same accuracy compared to the Monte Carlo simulations in terms of mean and 3δ decap budget range estimations, with run times that are orders of magnitude faster than MC.

REFERENCES

[1] "International technology roadmap for semiconductors(itrs) 2005, 2006 update," 2006, http://public.itrs.net.

[2] S. Bobba, T. Thorp, K. Aingaran, and D. Liu, "IC power distribution challenges," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2001, pp. 643–650.

[3] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," in *Proc. Design Automation Conf. (DAC)*, 1997, pp. 638–643.

[4] C. Chiang and J. Kawa, *Design for Manufacturability*. Springer, 2007.

[5] V. De and S. Borkar, "Technology and design challenges for low power and high performance," in *Proc. Int. Symp. on Low Power Electronics and Design(ISLPED)*, Aug. 1999, pp. 163–168.

[6] R. Downing, P. Gebler, , and G. Katopis, "Decoupling capacitor effects on switching noise," *IEEE Trans. on Components, Hybrids, and Manufacturing Technology*, vol. 16, no. 5, pp. 484–489, Aug. 1993.

[7] J. Fan, N. Mi, and S. X.-D. Tan, "Voltage drop reduction for on-chip power delivery considering leakage current variations," in *Proc. IEEE Int. Conf. on Computer Design (ICCD)*. Piscataway, NJ, USA: IEEE Press, 2007, pp. 78–83.

[8] J. Fu, Z. Luo, X. Hong, Y. Cai, S. X.-D. Tan, and Z. Pan, "A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, Jan. 2004, pp. 505–510.

[9] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*. Dover Publications, 2003.

[10] P. Ghanta, S. Vrudhula, R. Panda, and J. Wang, "Stochastic power grid analysis considering process variations," in *Proc. European Design and Test Conf. (DATE)*, vol. 2, 2005, pp. 964–969.

[11] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, 3rd ed. Cambridge University, 1996.

[12] T. Karnik, S. Borkar, and V. De, "Sub-90nm technologies-challenges and opportunities for CAD," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, San Jose, CA, Nov 2002, pp. 203–206.

[13] H. Li, Z. Qi, S. Tan, L. Wu, Y. Cai, and X. Hong;, "Partitioning-based approach to fast on-chip decap budgeting and minimization," *Proc. Design Automation Conf. (DAC)*, pp. 170 – 175, May 2005.

[14] N. Mi, J. Fan, and S. X.-D. Tan, "Analysis of power grid networks considering lognormal leakage current variations with spatial correlation," in *Proc. IEEE Int. Conf. on Computer Design (ICCD)*, 2006, pp. 56–62.

[15] S. Nassif, "Delay variability: sources, impact and trends," in *Proc. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, Feb 2000, pp. 368–369.

[16] E. Novak and K. Ritter, "Simple cubature formulas with high polynomial exactness," *Constructive Approximation*, vol. 15, no. 4, pp. 449–522, Dec 1999.

[17] C. Paul, "Effectiveness of multiple decoupling capacitors," *IEEE Trans. on Electromagnetic Compatibility*, vol. 34, no. 2, pp. 130–133, May 1992.

[18] W. H. Press, S. A. Teukolsky, W. T. Veterling, and B. P. Flannery, *Numerical Recipes in C: The art of Scientific Computing*. Cambridge University Press, 1992.

[19] R. Rutenbar, "Next-generation design and EDA challenges," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, January 2007, keynote speech.

[20] C. K. S. Zhao, K. Roy, "Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 1, pp. 81–92, Jan. 2002.

[21] Y. Shi, J. Xiong, C. Liu, and L. He, "Efficient decoupling capacitance budgeting considering operation and process variations," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1253–1263, Jul 2008.

[22] B. G. Streetman and S. Banerjee, *Solid-State Electronic Devices*. Prentice Hall, 2000, 5th ed.

[23] H. Su, S. S. Sapatnekar, and S. R. Nassif, "Optimal decoupling capacitor sizing and placement for standard cell layout designs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 4, pp. 428–436, April 2003.

[24] D. Xiu and G.Karniadakis, "Modeling uncertainty in flow simulations via generalized polynomial chaos," *J. of Computational Physics*, no. 187, pp. 137–167, 2003.