Efficient Decoupling Capacitance Budgeting Considering Correlated Current and Leff Variation

ABSTRACT

This paper develops decoupling capacitance (decap) insertion to minimize the sum of time-domain power noise integral over all ports (in short, noise) subject to given decap area. We propose a stochastic current model efficiently yet accurately capturing temporal correlation between clock cycles, logic-induced correlation between ports, and current variation due to L_{eff} variation with spatial correlation. Then we develop an iterative alternative linear programming algorithm applicable to a variety of current models. Compared with the baseline model which assumes maximum currents at all ports, the model considering temporal correlation reduces noise by up to 80%, and the model considering both temporal and logic-induced correlations reduces noise by up to 94%. Compared with using deterministic L_{eff}, considering L_{eff} variation reduces the mean noise by up to 74% and 3σ noise by up to 92% when both applying the current model with temporal and logic-induced correlations. To the best of our knowledge, this is the in-depth study on power network design considering current correlations and L_{eff} variation.

1. INTRODUCTION

Reliable power grid design has been an active research topic. A variety of techniques such as P/G network sizing [1], topology optimization [2] and decoupling capacitance insertion and sizing, or *decap budgeting* have been studied. This paper focuses on the decap budgeting problem, which can be formulated as a constrained nonlinear optimization problem and solved by linear programming [3], quadratic programming [4] or conjugate gradient method [5, 6].

However, the aforementioned studies [3–6] on power network design have assumed a maximum current load at every port to guarantee the worst-case design scenario. Such a practice is too pessimistic as it ignores two important correlations. First, the current at a port exhibits *temporal correlation*, i.e., the current cannot attain maximum all the time, and depending on the functionality being performed, the current variation for certain period of clock cycles are correlated. Second, current loads at different ports are correlated due to the inherent logic dependency for a given design, hence exhibiting *logic-induced correlation*.

Moreover, [3–6] ignore that current loads are also affected by process variation, although [7] has considered process variation induced leakage variation for power grid analysis. While the leakage power is comparable to the dynamic power because not all components are active simultaneously in a large system-on-chip, we believe that the dynamic current is still dominant compared with the leakage current. As pointed out in [8], in 90nm regime the most significant variation source is the effective channel length (L_{eff}), and L_{eff} variation can be more than 30%. How to design a reliable P/G network in the presence of such variation is still an open problem in the literature.

p	total port number
L	max number of clock cycles for temporal correlation
$\hat{I}_k^i$	peak current at port k in clock cycle i
b_k^j	a vector of the current peaks at port k sampled every L clock cycles starting from cycle j.
$\mathcal{B}_k^j$	stochastic variable representing the set of b_k^j
$\tilde{b}_k^j$	a vector of several b_k^j with different L_{eff}
$\tilde{\mathcal{B}}_k^j$	stochastic variable representing the set of $\tilde{b}_k^j$
r_k	independent stochastic variables after ICA

Table 1: Notations for stochastic current model.

In this paper, we propose a novel stochastic current model for current loads on power network and take into account both temporal and logic-induced correlations, as well as the effects of systematic L_{eff} variation. We formulate a new decap budgeting problem with stochastic current model and propose an iterative alternative linear programming approach to solve it efficiently and effectively. Experiments using industrial designs show that under the same decap area and compared with the baseline model which assumes maximum currents at all ports the model considering temporal correlation reduces noise by up to 80%, and the model considering both temporal and logic-induced correlations reduces noise by up to 94%. Compared with using deterministic L_{eff}, considering L_{eff} variation reduces the mean noise by up to 74% and 3σ noise by up to 92% when both applying the current model with temporal and logic-induced correlations.

The remaining of the paper is organized as follows. We propose the stochastic current model in Section 2. We then formulate the decap budgeting problem with the stochastic current model in Section 3, and discuss the algorithms to solve this problem in Section 4. We present experiments in Section 5 and concludes in Section 6.

2. CURRENT MODELING

2.1 Background

Similar to the vectorless P/G analysis in [9], we partition a circuit into blocks such that different blocks are relatively independent. For each block, there are multiple ports connected to the power network, and each port is modeled as

a time-varying current load for power network. We apply extensive simulation to each block *independently* to get the current signatures. Because we ignore the interdependence between blocks, the obtained current signatures may lead to worst-case design for increased reliability.

We represent the current in one clock cycle as a triangular waveform with rising time, falling time, and peak value $\hat{I}$. The peak values vary in different clock cycles and over different ports. More importantly, there are strong correlation between different ports which we call *logic-induced correlation*. In addition, the currents of a port in different clock cycles are also correlated. For example, it might take a block several clock cycles to execute certain functions and the current profile inside those clock cycles are dependent to each other. For simplicity, we assume that for a given design, there is a maximum number of cycles L that determines *temporal correlation distance* so that currents that are less than L cycles apart are temporally correlated, otherwise they are temporally independent. For example, we can choose L as the largest number of clock cycles to finish one instruction. We call this type of correlation as *temporal correlation*.

In the following, we devise a stochastic model which can efficiently capture both the logic-induced correlation and temporal correlation. For simplicity of presentation, we summarize notations for the stochastic current model in Table 1.

2.2 Stochastic Model to Consider Current Interdependence

We record the peak currents at port k ($1 \le k \le p$, assuming total port number p) at different clock cycles, and put them into vectors, i.e.,

$$b_k^j = [\hat{I}_k^j, \hat{I}_k^{j+L}, \hat{I}_k^{j+2L}, \ldots], \quad 1 \le k \le p, 1 \le j \le L \qquad (1)$$

where $\hat{I}_k^j$ is the peak currents at port k in clock cycle j, and b_k^j is the set of peak currents sampled every L clock cycles starting from cycle j. For example, if the peak values in each clock cycle for port 1 are $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$, and for port 2 are $[0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08]$, and we choose $L = 2$, then

$$b_1^1 = [0.1, 0.3, 0.5, 0.7], \quad b_1^2 = [0.2, 0.4, 0.6, 0.8],$$
$$b_2^1 = [0.01, 0.03, 0.05, 0.07], \quad b_2^2 = [0.02, 0.04, 0.06, 0.08]. \quad (2)$$

We can model the peak current at each port as a stochastic process. Then the set of b_k^j is the samples for the stochastic variable $\mathcal{B}_k^j$ with its mean $\mu(\mathcal{B}_k^j)$ and standard deviation $\sigma(\mathcal{B}_k^j)$. From the way we construct b_k^j, it is clear that $b_k^{j_1}$ and $b_k^{j_2}$ exhibit temporal correlation, while $b_{k_1}^j$ and $b_{k_2}^j$ exhibit logic-induced correlation.

With those stochastic variables B_k^j's and their corresponding samples b_k^j's, we can compute the logic-induced correlation matrix $\rho(j; k_1, k_2)$ which describes the correlation between the peak currents at any two ports k_1 and k_2 in clock cycle j as

$$\rho(j; k_1, k_2) = \frac{cov(\mathcal{B}_{k_i}^j, \mathcal{B}_{k_2}^j)}{\sigma(\mathcal{B}_{k_1}^j)\sigma(\mathcal{B}_{k_2}^j)}, \quad (1 \le k_1, k_2 \le p), \qquad (3)$$

where $cov(\mathcal{B}_{k_1}^j, \mathcal{B}_{k_2}^j)$ are the covariance between $\mathcal{B}_{k_1}^j$ and $\mathcal{B}_{k_2}^j$, and $\sigma(\mathcal{B}_{k_1}^j)$ and $\sigma(\mathcal{B}_{k_2}^j)$ are their standard deviations, respectively. Similarly, the temporal correlation matrix $\rho(j_1, j_2; k)$ which describes the correlation between the peak currents between clock cycles j_1 and j_2 at a given port k can be computed as

$$\rho(j_1, j_2; k) = \frac{cov(\mathcal{B}_k^{j_1}, \mathcal{B}_k^{j_2})}{\sigma(\mathcal{B}_k^{j_1})\sigma(\mathcal{B}_k^{j_2})}, \quad (1 \le j_1, j_2 \le L). \qquad (4)$$

2.3 Extension to Leff Variation with Spatial Correlation

We use the intra-die variation model for L_{eff} based on [10], which showed that the variation are mainly spatially correlated but not random, i.e.,

$$L_{eff} = L_0 + L^{prox} + L^{spat} + \epsilon, \qquad (5)$$

where L_0 is the overall mean, L^{prox} is a discrete stochastic variable with a distribution determined by the frequency of each gate, L^{spat} corresponds to the spatial variation and ϵ is the local random variation. Because

$$\hat{I}_k^i \sim L_{eff}^{-0.5} t_{ox}^{-0.8} (V_{dd} - V_t), \qquad (6)$$

with L_{eff} variation, the samples $\tilde{b}_k^j$ for each stochastic variable $\tilde{B}_k^j$ becomes

$$\tilde{b}_k^j = b_k^j \left[\sqrt{\frac{\bar{L}_{eff,k}}{L_{eff,k}^1}}, \sqrt{\frac{\bar{L}_{eff,k}}{L_{eff,k}^2}}, \ldots \right], \qquad (7)$$

where $L_{eff,k}^i$ with different i are the samples of $L_{eff,k}$ for the macro corresponding to port k with the nominal value $\bar{L}_{eff,k}$, and b_k^j are the samples for $\mathcal{B}_k^j$ in the deterministic case as discussed in Section 2.2, In other words, if we have n samples for $L_{eff,k}$, then $\tilde{b}_k^j$ is a vector n times longer than b_k^j.

2.4 Correlation Removal via ICA

To efficiently handle the large size of samples $\tilde{b}_k^j$'s, we apply clustering to divide them into a small number of clusters by the K-means method [13]. We use the 2-norm to measure the distance between two samples. If two samples have small enough distance, they belong to one cluster. Therefore, the samples in each cluster tend to follow a similar distribution.

If all those variable $\tilde{B}_k^j$'s are Gaussian, then we can apply principal component analysis (PCA) to each cluster to remove the interdependence between the stochastic variables $\tilde{\mathcal{B}}_k^j$'s. However, this is not the case for our stochastic current model. Therefore, we use independent component analysis (ICA) that is applicable to non-Gaussian distribution. The input to ICA is the samples $\tilde{b}_k^j$ as well as their correlation matrices (3) and (4), and the output are a set of independent stochastic variables r_i and their corresponding coefficients $a_i(j,k)$ to reconstruct each $\tilde{\mathcal{B}}_k^j$, i.e.

$$\tilde{\mathcal{B}}_k^j = \sum_{i=1}^{q} a_i(j,k) r_i. \qquad (8)$$

The order q is determined for each design such that the relative error between the original currents and model predicted currents is less than 5%. The probability density functions (PDFs) for each r_i is also given in the output of ICA as a one-dimensional lookup table, based on which we can bound the range of r_i as

$$\underline{r_i} \le r_i \le \overline{r_i}, \qquad (9)$$

where $\underline{r_i}$ and $\overline{r_i}$ can be related to r_i's mean (μ) and variance (σ^2). For example, we can take $\underline{r_i}$ as $\mu - 4\sigma$, while $\overline{r_i} = \mu + 4\sigma$.

Therefore, assuming constant rising and falling times in describing the triangular current waveform, together with $a_i(j,k)$ which represents the i-th component of the peak current at port k in clock cycle j, we have all the necessary information to obtain the i-th time-varying current waveform component $u_i(t;j,k)$. If we denote T as the clock period, then $jT \le t \le (j+1)T$. Put those $u_i(t;j,k)$ at all ports in clock cycle j together as

$$u_i(t;j) = \begin{pmatrix} u_i(t;j,1) \\ u_i(t;j,2) \\ \vdots \\ u_i(t;j,p) \end{pmatrix}, \quad jT \le t \le (j+1)T, \qquad (10)$$

and then combine all the $u_i(t;j)$ in different clock cycles, we can get $u_i(t)$ with $0 \le t \le LT$. Finally, according to superposition theorem, we have

$$u(t) = \sum_{i=1}^{q} u_i(t) r_i, \quad 0 \le t \le LT. \qquad (11)$$

From now on, we will apply (11) to the optimization.

3. PROBLEM FORMULATION

3.1 Parameterized MNA for Decap Budgeting

The P/G network can be modeled as a linear RLC network with each segment and pad modeled as a lumped RLC element from extraction. The behavior of any linear RLC network with p ports of interests is fully described by its state representation following the modified nodal analysis (MNA)

$$Gx + C\frac{dx}{dt} = Bu, \qquad (12)$$

$$y = L_0^T x, \qquad (13)$$

where x is a vector of nodal voltages and inductor currents, u is a vector of current sources at all ports, G is the conductance matrix, C is a matrix that includes both inductance and capacitance elements, B and L_0 are port incident matrices, and y is the output voltages of interests at the p ports.

When a decap with size w_i is inserted into the power network at a given location, its impact can be considered by adjusting matrices G and C based on the location at the network and the size of the decap. Mathematically, it can be represented as

$$G = G_0 + \sum_{i=1}^{M} w_i \cdot G_{w,i}, \qquad (14)$$

$$C = C_0 + \sum_{i=1}^{M} w_i \cdot C_{w,i}, \qquad (15)$$

where G_0 and C_0 are the original matrices for the power network without decap, M is the total number of decaps, and $G_{w,i}$ and $C_{w,i}$ provide the stamping of a unit width decap at the i-th location. In reality, w_i has an upper bound, i.e.,

$$0 \le w_i \le \overline{w_i}. \qquad (16)$$

We call the MNA equation with G given by (14), C given by (15), and u given by (11) as *parameterized MNA formulation* for decap budgeting. One of the major advantages in using this parameterized MNA formulation is that it enables us to implicitly compute sensitivities efficiently and accurately, which will become clearer in the late part of this paper.

3.2 P/G Network Noise Modeling

Because of the duality between power and ground networks, in the following, we will focus our explanation on the power network design. But it is understood that the same formulation applies to the ground network design as well. Same as [4–6], we model the power network induced noise at a node as the integral of the voltage drop below a user specified noise ceiling $\overline{U}$ within one clock cycle T

$$z_i = \int_{\Omega_i} (\overline{U} - y_i(t))dt, \qquad (17)$$

where Ω_i is the time duration when voltage y_i drops below the noise ceiling $\overline{U}$, i.e.,

$$\Omega_i = \{t | y_i(t) \le \overline{U}\}. \qquad (18)$$

The figure of merit that measures the qualify of the whole power network design is defined as the sum of noise at all ports of interest, i.e.,

$$f = \sum_{i=1}^{p} \int_{\Omega_i} (\overline{U} - y_i(t))dt. \qquad (19)$$

We will call the noise measurement in (19) simply as noise in the rest of the paper.

3.3 Problem Formulation

The decap budgeting problem can be formulated as the following optimization problem:

FORMULATION 1. **Decap Budgeting**: *Given a power network modeled as a RLC network with specified power pads, time-varying current at different ports, and total available white space $\overline{W}$ for decoupling capacitance, the DecapOpt problem determines the places to insert decoupling capacitance and the sizes of each decoupling capacitance, such that the noise defined in (19) is minimized.*

The decap budgeting problem with stochastic current model can be mathematically represented as follows:

$$(\mathbf{P1}) \quad \min_{w_i} \quad \sup_{r_k} f = \sum_{i=1}^{p} \int_{\Omega_i} (\overline{U} - y_i(w_i, r_k; t))dt \qquad (20)$$

$$s.t. \quad 0 \le w_i \le \overline{w_i}, \quad 1 \le i \le M \qquad (21)$$

$$\sum_{i=1}^{M} w_i \le \overline{W} \qquad (22)$$

$$\underline{r_k} \le r_k \le \overline{r_k}, \quad 1 \le k \le q \qquad (23)$$

where voltage y_i is a function of w_i, r_k, and time t.

Problem (**P1**) is a constrained min-max optimization problem. The *sup* operation over all random variables r_k is to find the worst-case noise violation measures for a given power network design. This operation guarantees that all P/G network designs satisfy the given design constrains while considering both the temporal and logic-induced correlations among ports as well as systematic L_{eff} variation.

This is of particular use for ASIC-style designs, where the worst-case design performance has to be ensured for sign-off. The *min* operation over all decap sizes w_i is to find the optimal decap budgeting solution so that the worst-case noise violation is minimized.

4. ALGORITHMS

4.1 Iterative Alternative Programming

Because there exists no general technique to solve the constrained min-max problem (**P1**) optimally, we resort to an effective iterative optimization strategy, which we call *iterative alternative programming* (IAP). That is, instead of solving the min-max problem (**P1**) directly, we solve it by iteratively solving the following two sub-problems alternatively.

The first sub-problem assumes that all decaps' sizes w_i are known, hence the worst-case noise can be obtained by solving the following optimization problem

$$(\mathbf{P2}) \quad \sup_{r_k} \quad f = \sum_{i=1}^{p} \int_{\Omega_i} (\overline{U} - y_i(w_i, r_k; t))dt \quad (24)$$

$$s.t. \quad \underline{r_k} \leq r_k \leq \overline{r_k}, \quad 1 \leq k \leq q \quad (25)$$

The second sub-problem assumes that all random variables r_k have fixed values, hence the decap sizes to achieve the minimum noise can be obtained by solving the following optimization problem

$$(\mathbf{P3}) \quad \min_{w_i} \quad f = \sum_{i=1}^{p} \int_{\Omega_i} (\overline{U} - y_i(w_i, r_k; t))dt \quad (26)$$

$$s.t. \quad 0 \leq w_i \leq \overline{w_i}, \quad 1 \leq i \leq M \quad (27)$$

$$\sum_{i=1}^{M} w_i \leq \overline{W}. \quad (28)$$

Problem (**P3**) is exactly the deterministic version of the original problem formulation (**P1**).

The overall algorithm can be described in Algorithm 1, where *iter* is the current iteration number, *ITER* is the maximum iteration bound, and ϵ determines the stop criteria of the optimization procedure, i.e., it stops when the change of objective function $|\Delta f|$ is sufficiently small.

Algorithm 1 Iterative alternative programming.

```
INPUT: initial guess w_i, r_k;
OUTPUT: final solution w_i to problem (P1);
for iter = 0; |Δf| ≤ ε or iter ≤ ITER; iter++ do
  w_i = solve-P3(iter, w_i, r_k);
  r_k = solve-P2(iter, w_i, r_k);
  Compute objective function with new r_k and w_i;
end for
```

4.2 Efficient Sequential Programming

Both problems (**P2**) or (**P3**) are constrained nonlinear optimization problems, and there exits no general technique to solve them efficiently. Because the constraints in both problems are linear, if we can approximate the objective function f by a first-order linear function, the original problems would become linear programming (LP) problems[1]. Because efficient solvers exist for LP problems, we can solve the approximated problems more efficiently than solving the original problems directly. Therefore, we propose to solve the original (**P2**) or (**P3**) problem via sequential linear programming (sLP).

For now, let us assume that we know how to compute the first-order sensitivities of the objective function f with respect to changing variables, which will be discussed in Section 4.3. Therefore, we can easily obtain the linear approximation of the objective function. For example, for the objective function in problem (**P3**), the changing variables are all Δw_i. Therefore, we have the following linear approximation for the objective function

$$f \approx f_0 + \sum_{i=1}^{M} \frac{\partial f}{\partial w_i} \Delta w_i, \quad (29)$$

where f_0 is the current value of the objective function, and $\frac{\partial f}{\partial w_i}$ are the first-order sensitivities of f. Apparently, (29) is a linear function of Δw_i. By replacing (24) with (29), we obtain an approximated LP formulation for (**P3**).

A high-level description of the sequential programming algorithm to solve either problem (**P2**) or (**P3**) is shown in Algorithm 2, where *iter2* is the current iteration number, *ITER2* is the maximum iteration bound. The iterations stop when the change of objective function $|\Delta f|$ is smaller than ϵ_2, which is dynamically adjusted according to the iteration number *iter* in the outer-loop of Algorithm 1. We employ an exponential decreasing function to adjust ϵ_2 in this work. The idea is that when the out-loop iteration is small (or we are far from the optimal solution), we can have an early termination of the inner-loop optimization procedure as shown in Algorithm 2 early. But when the outer-loop iteration becomes large enough (or we are close to the optimal solution), we should spend more time in each inner-loop optimization to find a better global optimal solution. Parameter η is used to control the efforts that we should spend in the inner-loop's optimization.

Algorithm 2 Sequential linear programming for solving (**P2**) and (**P3**).

```
INPUT: iter, w_i, r_i;
OUTPUT: updated w_i for (P3) or r_i for (P2);
ε_2 = exp(-η·iter);
for iter2=0; |Δf| ≤ ε_2 or iter2 ≤ ITER2; iter2++ do
  Compute the first-order sensitivities of f;
  Formulate (P2) or (P3) as an LP problem;
  Call LP solver to solve the above problem;
  Compute objective function with new w_i or r_i;
end for
```

4.3 Sensitivity Computation

To solve (**P2**) and (**P3**) via sLP, we need to compute the sensitivity of the objective function f with respect to the design variables, i.e., either w_i or r_i. Because this computation is similar for both (**P2**) and (**P3**), we will focus our discussion on (**P3**) in the following.

The first-order sensitivities of the objective function f of problem (**P3**) are defined as

$$\frac{\partial f}{\partial w_i} = -\sum_{i=1}^{p} \int_{\Omega_i} \frac{\partial y_i}{\partial w_i} dt = -\sum_{i=1}^{p} \int_{\Omega_i} L_{0i}^T \frac{\partial x}{\partial w_i} dt, \quad (30)$$

[1]We can also extend our technique to approximate the objective function f by a second-order quadratic function, then the problem would become a quadratic programming (QP) problem.

For simplicity of presentation, we have loosely applied the derivative notation on a vector for component-wise derivative. To compute the sensitivity of f w.r.t. w_i, all we need to know is the sensitivity of the state vector x with respect to w_i. We use Taylor expansion to express x as follows

$$x = x_0 + \sum_{i=1}^{M} \alpha_i \Delta w_i + \ldots, \tag{31}$$

where α_i is the first-order sensitivity of x w.r.t. random variable w_i, i.e., we have

$$\frac{\partial x}{\partial w_i} = \alpha_i. \tag{32}$$

To compute these sensitivities, we recognize that x also satisfies the differential equation given by (12). By Laplace transformation and applying the *parameterized MNA formulation*, we re-write (12) as follows

$$(G + \sum_{i=1}^{M} \Delta w_i \cdot G_{w,i})x + s(C + \sum_{i=1}^{M} \Delta w_i \cdot C_{w,i})x = Bu. \tag{33}$$

By plugging (31) into (33), we obtain terms of Δw_i with different orders. By equating the zero-order terms of Δw_i from both left and right hand sides in (33), we obtain a set of equations as follows

$$(G + sC)x_0 = Bu. \tag{34}$$

By equating the first-order terms of Δw_i, we obtain sets of equations as follows for all $1 \leq i \leq M$

$$(G + sC)\alpha_i = -(G_{w,i} + sC_{w,i})x_0. \tag{35}$$

By applying the Backward Euler integration formula and assuming the time step as h, we can re-write (34) and (35) as follows

$$(G + \frac{C}{h})x_0(t+h) = Bu(t+h) + x_0(t)\frac{C}{h}, \tag{36}$$

$$(G + \frac{C}{h})\alpha_i(t+h) = -(G_{w,i} + \frac{C_{w,i}}{h})x_0(t+h) + \frac{x_0(t)C_{w,i} + \alpha_i(t)C}{h}. \tag{37}$$

Because all equations in (36) and (37) share the same left-hand side matrix, $(G + C/h)$, we only need to perform LU-factorization once, and then reuse the same factorization to solve for x_0 and α_i sequentially at each time step. This computation is efficient because it only involves some matrix-vector multiplications, and backward and forward substitutions.

In summary, we can compute the first-order sensitivities of the objective function f of problem (**P3**) by following the Algorithm 3. Note that once we know x_0 after solving (36), we can compute the voltage response at all ports of interests as $y = L_0^T x_0$. Hence the objective function f can be evaluated by following the definition in (19).

Algorithm 3 Sensitivity computation for (**P3**).

INPUT: w_i, r_k, h, T;
OUTPUT: f and α_i;
factorization: LU factorize $G + C/h$;
for $t = 0$; $t + h \leq T$; $t = t + h$ **do**
  Solve (36) for $x_0(t + h)$;
end for
for $t = 0$; $t + h \leq T$; $t = t + h$ **do**
  Solve (37) for $\alpha_i(t + h)$;
end for

5. EXPERIMENTAL RESULTS

In this section, we present experiments using four industrial P/G network designs. For each benchmark, we randomly selected 20% of total nodes as candidate nodes for decap insertion, i.e., $M = 20\%N$. We run experiments on a UNIX workstation with Pentium 2.66G CPU and 1G RAM. We partition the circuits according to the method in [9]. We use K-means clustering and the package FASTICA [11] to perform ICA. Finally, we use MOSEK as the linear programming solver [14] and random walk based simulator [12] with detailed input current waveform to obtain the noise reported in this section. This verifies not only our algorithm but also our stochastic current modeling.

5.1 Decap Budgeting without Leff variation

We compare three current models as shown in Table 1: maximum currents at all ports, stochastic model with logic-induced correlation only ($L = 1$), and stochastic model with both logic-induced and temporal correlation. For temporal correlation, we always use $L = 4$ since all circuits tested take at most four clock cycles to complete any one instruction. Table 1 reports the noise and runtime for the four benchmarks with different number of nodes. Compared with the baseline model with maximum currents at all ports [2], the model considering temporal correlation reduces noise by up to 80%; and the model considering both temporal and logic-induced correlations reduces noise by up to 94% (see bold in Table 2). This is because the first two models cannot model the currents effectively and lead to inserting unnecessarily large decaps in some regions. Thus, they result in more noise in the other regions since the total decap area is given. As for the runtime, model 2 needs about 1.5× more time than model 1, while model 3 needs about 2.3× more. The runtime overhead is the price we have to pay in order to achieve better designs.

In Fig. 1, we plot the time-domain responses at one randomly selected port for two optimization iterations by alternatively solving the problem (**P3**) and (**P2**). The benchmark has 1284 nodes. The initial waveform is denoted by "A0:initial". After performing decap sizing once by solving problem (**P3**) with a fixed choice of random variables r_k, we obtain the new waveform as denoted by "A1:(**P3**)". We then switch to solve problem (**P2**) by varying the values of those random variables r_k, but with fixed decap sizes w_i. We see that the waveform of the final worst-case voltage drop becomes worse compared to the deterministic solution; hence we obtain a new voltage drop waveform as denoted by "A2:(**P2**)". We then switch back to solve the decap sizing problem (**P3**) with fixed but newly updated choice of random variables r_k. At the end of this optimization, we arrive at a new voltage waveform as denoted by "A3:(**P3**)". Apparently, compared to "A1:(**P3**)", the new solution has smaller voltage drop. If we continue the same procedures by following the IAP algorithm given in Fig. 1, similar sequences of time domain voltage drop waveforms would repeat as we have shown in Fig. (1) until we converge to an optimal solution. Also, The voltage drop is reduced mostly

[2]We solve it by iteratively solving (**P3**) without altering to (**P2**).

Model 1	maximum currents at all ports
Model 2	stochastic model with logic-induced correlation
Model 3	Model 2 + temporal correlation

Node #	noise (V*s)			runtime (s)		
	model 1	model 2	model 3	model 1	model 2	model 3
1284	**6.33e-7**	**1.28e-7**	**4.10e-8**	**104.2**	**161.2**	**282.3**
10490	5.21e-5	1.09e-5	4.80e-6	973.2	1430	2199
42280	7.92e-4	5.38e-4	9.13e-5	2732	3823	5238
166380	1.34e-2	5.37e-3	2.28e-3	3625	5798	7821
avg	1	37.3%	11%	1	1.50X	2.26X

Table 2: Noise and runtime comparison between the three models.

Node #	V.R.	sLP			sLP+L_{eff}		
		μ (V*s)	3σ (V*s)	RT (s)	μ (V*s)	3σ (V*s)	RT (s)
1284	10%	9.28e-7	3.97e-7	184.2	6.14e-7	1.38e-7	332.8
	20%	9.43e-7	4.55e-7		6.38e-7	1.86e-7	(1.81X)
10490	10%	1.03e-4	4.79e-5	1121	7.22e-5	1.23e-5	3429
	20%	1.22e-4	6.38e-5		7.94e-5	2.06e-5	(3.06X)
42280	10%	2.29e-3	9.72e-4	2236	8.23e-4	1.01e-4	6924
	20%	2.43e-3	1.01e-3		8.28e-4	1.92e-4	(3.10X)
166380	**10%**	**2.06e-2**	**9.91e-3**	**3824**	**5.31e-3**	**8.32e-4**	**11224**
	20%	2.31e-2	1.03e-2		5.92e-3	9.33e-4	(2.93X)
avg	10%	1	1	1	49.5%	19.8%	2.73X
	20%	1	1		51.2%	24.7%	

Table 3: μ, 3σ and runtime (RT) comparison between sLP+L_{eff} and SLP. The variation amount (V.R.) represents the intra-die variation.

in the first optimization iteration denoted as "A1:(**P3**)". Afterward, the voltage drop reduction is relatively small. This observation is in agreement with the common knowledge about any sensitivity-based optimization techniques. In this particular example, we find that the first two iterations reduces the noise by 51.4%.

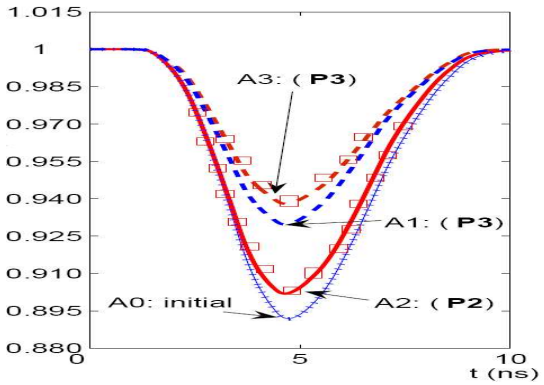

Figure 1: Time domain waveforms at one port after sLP for different iterations.

5.2 Leff Variation Aware Decap Budgeting

In the presence of variation, we want to minimize the worst-case noise for L_{eff} variation. We solve this via the proposed IAP technique in Algorithm 1. We denote our L_{eff} variation aware approach as sLP+L_{eff} and the counterpart as sLP. We compare the mean value μ and 3σ value of the noise distribution based on Monte Carlo simulation with 10,000 runs, and the results are reported in Table 3.

When the variation amount is 10% (20%), compared with using deterministic L_{eff}, considering L_{eff} variation reduces the mean noise by up to 74% and 3σ noise by up to 92% (see bold in Table 3), when both applying the current model with temporal and logic-induced correlations. Interestingly, we find that the relative improvement decreases as the process variation increases. This is expected, as usually the 3σ noise heavily depends on the variation. The larger the variation, the worse the 3σ value. As for the runtime between sLP and sLP+L_{eff}, sLP+L_{eff} needs about 2.7× more time than sLP on average.

6. CONCLUSIONS AND FUTURE WORK

This paper develops decoupling capacitance (decap) insertion to minimize time-domain power noise integral (in short, noise) subject to given decap area. We propose a stochastic current model efficiently yet accurately capturing temporal correlation between clock cycles, logic-induced correlation between ports, and impact of L_{eff} variation with spatial correlation. Then we develop an iterative alternative programming algorithm applicable to a variety of current models. Compared with the baseline method which assumes maximum currents at all ports, the method considering temporal correlation reduces noise by up to 80%, and the method considering both temporal and logic-induced correlations reduces noise by up to 94%. Compared with using deterministic L_{eff}, considering L_{eff} variation reduces the mean noise by up to 74% and 3σ noise by up to 92% when both applying the current model with temporal and logic-induced correlations. To the best of our knowledge, this is the in-depth study on power network design considering current correlations and L_{eff} variation.

It is clear that the proposed technique can be easily extended to consider other types of process variation (e.g., threshold variation) and design freedoms for designing reliable power supply networks. This will be our future work.

7. REFERENCES

[1] X. D. Tan and C. J. Shi, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," in *IEEE/ACM DAC*, pp. 78–83, 1999.

[2] K.-H. Erhard and et al, "Topology optimization techniques for power/ground networks in VLSI," in *DATE*, 1992.

[3] M. Zhao and et al, "A fast on-chip decoupling capacitance budgeting algorithm using macromodeling and linear programming," in *IEEE/ACM DAC*, 2006.

[4] H. Su and et al, "Optimal decoupling capacitor sizing and placement for standard-cell layout designs," *IEEE TCAD*, 2003.

[5] J. Fu and et al, "A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery," in *ASPDAC*, 2004.

[6] H. Li and et al, "Partitioning-based approach to fast on-chip decap budgeting and minimization," in *IEEE/ACM DAC*, 2005.

[7] I. A. Ferzli and F. N. Najm, "Statistical verification of power grids considering process-induced leakage current variations," in *IEEE/ACM ICCAD*, 2003.

[8] Y. Cao and L. T. Clark, "Mapping statisitical process variations toward circuit performance variability: An analytical modeling approach," in *IEEE/ACM DAC*, 2005.

[9] D. Kouroussis and et al, "Incremental partitioning-based vectorless power grid verification," in *IEEE/ACM ICCAD*, 2005.

[10] M. Orshansky and et al, "Impact of Spatial Intrachip Gate Length Variability on the Performance of High-speed Digital Circuits," *IEEE TCAD*, 2002.

[11] A. Hyvarinen and E. Oja, "A Fast Fixed-Point Algorithm for Independent Component Analysis," *Neural Computation*, 1997.

[12] H. Qian and et al, "Power Grid Analysis Using Random Walks," *IEEE TCAD*, 2005.

[13] A. Moore, "K-means and Hierarchical Clustering - Tutorial Slides,".

[14] http://www.mosek.com