

Localized On-Chip Power Delivery Network Optimization via Sequence of Linear Programming*

Jeffrey Fan[†], I-Fan Liao[†], Sheldon X.-D. Tan[†], Yici Cai[§] and Xianlong Hong[§]

[†]Department of Electrical Engineering University of California, Riverside, CA 92521

[§]Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China

Abstract— In this paper, we propose an efficient algorithm to reduce the voltage noises for on-chip power/ground (P/G) networks of VLSI. The new method is based on the sequence of linear programming (SLP) method as the optimization engine and a localized scheme via partitioning for dealing with large circuits. We show that by directly optimizing the decap area as the objective function and using the time-domain adjoint method, SLP can deliver much better quality than existing methods based on the merged time-domain adjoint method. The partitioning strategy further improves the scalability of the proposed algorithm and makes it efficient for large circuits. The resulting algorithm is general enough for any P/G network. Experimental results demonstrate the advantage of the proposed method over existing state-of-the-art methods in terms of solution quality at a mild computation cost increase.

I. INTRODUCTION

In modern deep sub-micron and nanometer VLSI technology, signal integrity is among the most important concerns for circuit designers. With reduced noise margins and increased switching frequency, reliable on-chip power supply has become a critical factor for robust circuit performance. Power/ground (P/G) networks are devoted to supplying power to all on-chip modules. Extra design effort is often required to reduce voltage noises in P/G networks, so that the variation in power supply voltage and reference ground voltage is confined within a certain percentage (like 10%) of nominal values. Excessive voltage drops and ground bounces not only degrade noise margins and increase gate delays, but also lead to false logic switching and logic failure.

P/G network design and optimization has been studied extensively in previous works [18, 17, 1, 14, 19, 6, 12, 9]. Besides early stage design techniques like topology selection and wire-sizing, adding decoupling capacitance (decaps) has been accepted as an effective and standard approach to remove excessive instantaneous voltage variations induced by IR drops. As shown in Fig 1, decaps provide a reservoir of current that is instantly available for nearby switching components, thus removing spikes and glitches in the power rail. Intuitively, decaps have a strong local effect and should be placed around logic units that tend to draw large currents. Indeed, in some

early P/G designs decaps were added manually after the current pattern of digital modules was observed. However, On-chip decaps are typically manufactured using MOS transistors, and excessive on-chip decaps would not only consume on-chip area, but also cause more leakage power, low yield and lower resonant frequency [1]. Therefore, as long as power supply noises are constrained, decaps should be minimized, especially when the white space (WS) left after placement and routing is limited.

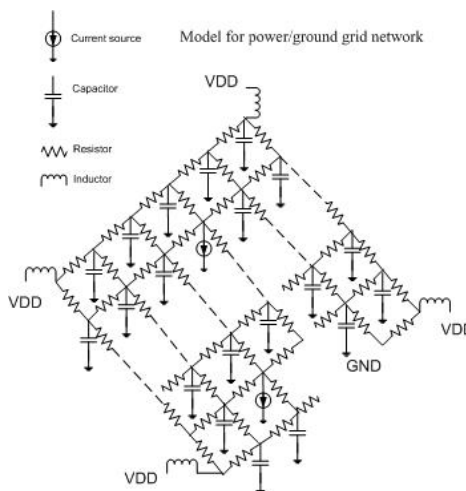


Fig. 1. The model of power grid network

Optimization of decap placement has been extensively studied in the past [14, 2, 17, 19, 6, 12, 9]. Some earlier works [2, 14] place decaps according to estimation of noises in the power supply caused by nearby digital modules, while more recent works treat it more mathematically as a nonlinear optimization problem and employ the adjoint method (or its variant) to compute sensitivity first, then adopting different optimization techniques like quadratic programming [17] or sequence of quadratic programming (SQP) [19], conjugate gradient (CG) [6] or CG combined with binary search [12, 9].

To compute sensitivity, transient simulations of the whole P/G network have to be carried out at every optimization step. Given the fact that the transient simulation of P/G networks with millions of nodes is already an extremely time-consuming task, the CPU time and memory cost of optimization meth-

*This work is funded by NSF CAREER Award CCF-0448534, NSF REU CCF-0529754 and NSF OISE-0451688.

ods that perform transient simulations in internal loops will be prohibitive. To combat this, [7, 19] proposed to reduce the P/G grid first and then apply standard optimization techniques. The work in [9] partitions the circuit before optimizing individual subcircuits.

In this paper, we propose an efficient localized decap allocation algorithm, which explicitly minimizes the decap area subject to voltage drops and other design rule constraints. We formulate the decap allocation problem as a linear programming problem and solve it by the sequence of linear programming (SLP) method. To achieve higher efficiency with large circuits, a partition strategy similar to [9] is employed to take advantage of the localized effect of decaps. The new algorithm is especially suitable for P/G grids with a few troubling spots. This is typically the case for a properly designed P/G grid, or when a priori decaps have already been added based on some simple estimation. Experimental results show that the new algorithm yields significantly less decap area than the recently proposed decap allocation algorithm with a mild computation cost increase [9].

The rest of this paper is organized as follows. The next section describes the decap optimization problem and briefly reviews existing sensitivity-based decap budgeting algorithms. Section III formulates decap budgeting into a sequence of linear programming problem. Section IV introduces the partition strategy and presents the flow of our partitioning based SLP optimization. Experimental results are presented in section V and the last section concludes the paper.

II. PROBLEM FORMULATION AND REVIEW OF PREVIOUS METHODS

Existing on-chip decap budgeting algorithms basically fall into two categories [12]. One [15, 11, 2, 14] is to compute the current pattern around nodes where excessive IR drops occur (Fig. 2) and estimate the amount of electric charge required for the current demand. The idea is straightforward but usually suffers from the difficulty of accurately estimating voltage drops and electric charge, therefore cannot allocate decaps in an optimal way.

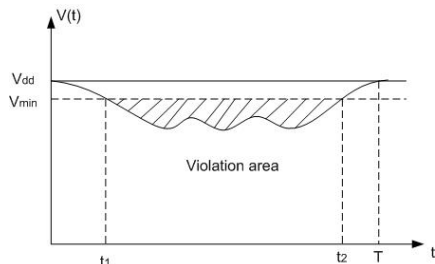


Fig. 2. Illustration of IR drop violation

Another category is based on sensitivity computation [17, 19, 6, 12, 9]. Fig 2 illustrates the VDD fluctuation at a node

within one clock cycle. The *violation area* at node j is defined as:

$$g_j(c_1, \dots, c_n) = \int_0^T \max(V_{min} - v_j(t), 0) dt \quad (1)$$

which equals the shaded area below the designer-defined VDD threshold V_{min} in the graph. The contribution of decap added at node i to removing violation at node j can be measured by *sensitivity*:

$$s_{ij} = \frac{\partial g_j(c_1, \dots, c_n)}{\partial c_i} \quad (2)$$

where c_i is the decap at node i , and n the number of nodes where decaps can be added. If all circuit nodes are available for adding decaps, n is simply the number of circuit nodes. The adjoint method [5] can be readily used for sensitivity calculation, where two circuit simulations, one for the original network and the other for the adjoint network are involved. The sensitivity for capacitive elements can then be expressed as:

$$s_{ij} = \int_0^T v'_{i,j}(T-t) \times \dot{v}_i(t) dt, \quad (i = 1, 2, \dots, n) \quad (3)$$

where $\dot{v}_i(t)$ is the derivative of the voltage waveform at node i in the original network, and $v'_{i,j}(T-t)$ is the waveform at node i in the adjoint network under unit step current excitation at violation node j . Once all s_{ij} are available, the decap optimization problem can be formulated as:

Objective function

$$\min \sum_{j=1}^m g_j(c_1, \dots, c_n) \quad (4)$$

Maximum decap constraint

$$(c_i) \leq d_i, \quad d_i \geq 0 \quad (5)$$

Basically all methods [17, 19, 12, 6, 9] followed the above problem formulation scheme, and they only differ from one another in the optimization technique applied thereafter. The method in [17] applies the Lagrangian relaxation technique and feeds it into a standard QP (Quadratic Programming) solver. Unfortunately, introducing Lagrangian variables increases the problem size and the complexity of QP is relatively high. Some [19] proposes to coarsen the circuit with the standard multigrid (SMG) scheme and then applies SQP (sequential programming) to optimize the reduced problem, and maps the obtained decaps back into the fine grid. SMG achieves impressive speedup for large circuits, but it is limited to regular mesh structure, thus cannot be applied to general industrial circuits [16]. It also has more limitations in the practical decap optimization scenarios [19].

Some previous work [6] develops *merged adjoint method* from adjoint method and bases its CG (conjugate gradient) optimization on *merged sensitivity*. The idea is that, instead of computing s_{ij} , we compute the merged sensitivity

$$\sum_{j=1}^m s_{ij} = \frac{\partial \sum_{j=1}^m g_j(c_1, \dots, c_n)}{\partial c_i} \quad (6)$$

applying the superposition law for linear circuits in the adjoint method

$$\sum_{j=1}^m s_{ij} = \int_0^T (v'_{i,all}(T-t)) \times \dot{v}_i(t) dt \quad (i = 1, 2, \dots, n) \quad (7)$$

One method [12] modifies the objective function described in [6] and combines the binary search with CG, thus it achieves significant speedup. The work in [9] further proposes a partitioning strategy before the optimization of each sub-circuit. However, using the merged sensitivity is a trade-off for speed, as pointed out in [9], certain information is lost and optimization results are non-optimal.

In this work, we apply the sequence of linear programming (SLP) method which directly minimizes decap budgets. SLP was applied to sizing the P/G grid network before and was shown to be more efficient than conjugate gradient methods [18]. To deal with large circuits, we follow the partition strategy in [9] to take advantage of the localized effects of adding decaps.

III. DECAP PROBLEM FORMULATED INTO SLP

In this section, we show how sequence of linear programming (SLP) can be applied to decap budgeting.

A. Sequence of Linear Programming

Sequence of linear programming(SLP) is to linearize the nonlinear part of a nonlinear optimization problem and solve the linearized problem with linear programming in an iterative way. SLP is similar to the *Newton-Raphson* method for solving nonlinear circuits, where the linearized circuit matrix (*Jacobian*) is solved iteratively. Thus SLP has a quadratic convergence rate and is usually better than conjugate gradient optimization in terms of convergence rate [18].

B. Problem Formulation

In contrast to the problem formulation described in Section II, we choose decap area as the sole objective to minimize and enforce violation elimination as a constraint. Since the area for a capacitor is linearly proportional to the capacitance value, minimization of total decap area is equivalent to minimization of total decap value.

Objective function

$$\min \sum_{i \in \text{allnodes}} (c_i) \quad (8)$$

Violation elimination constraint

$$\begin{aligned} g_j(c_1, \dots, c_m) &= \int_0^T \max(V_{min} - v_j(t), 0) dt \\ &= \int_{t_s^{(j)}}^{t_e^{(j)}} (V_{min} - v_j(t)) dt \\ &= 0 \end{aligned} \quad (9)$$

Maximum decap constraint

$$(c_i) \leq d_i, \quad d_i \geq 0 \quad (10)$$

where d_i is the maximum decap allowed at node i , a parameter decided by the available white space (WS) around node i and can be specified by designers. Notice that there may be other power integrity constraints, such as current density for electromigration. Extra constraints can be included, but we will ignore them here for simplicity.

In our problem, the only nonlinear portion is the constraint (9). Hence our first step is to linearize it with sensitivity s_{ij} defined in (2), which can be calculated with the time-domain adjoint method.

Specifically, given s_{ij} , to remove the IR drop violation at node j , we add decap at node i based on the first order approximation, and do the same for other candidate nodes with tunable decaps. As a result, to meet with the nonlinear violation constraint (9) at violation node j , we have the following linearized constraint for added decaps Δc_i :

$$g_j \leq \sum_{i \in \text{allnodes}} s_{ij} \Delta c_i \quad (11)$$

Replacing constraint Eq. (9) with (11), we end up with a linear programming problem. Iteration goes on until all violation is eliminated or no solution can be found.

C. Constraint Relaxation

The need for constraint relaxation is motivated by the observation that decap sensitivity is a function of decap values, and the relationship turns out to be nonlinear. Detailed study shows that the decap sensitivity (absolute value) of a node typically increases with added decap values at the same node as shown in Fig. 3. It reaches a peak at some point and then goes down to zero when the violation goes away. This implies that we would over-estimate decaps based on the smaller sensitivity calculated at beginning and closing stages.

Two problems arise. First, from (11), SLP may fail to find a feasible solution, as small sensitivities call for large decaps, given the same violation area. But allowable decap value is bounded by constraint (10). Second, we may overestimate decaps. This actually explains why existing sensitivity based methods like [17], which does not optimize decaps, tend to yield more than necessary decaps.

To resolve this issue, we intentionally *relax* the violation area constraint by artificially increasing all sensitivity values calculated at earlier stages by multiplying them, or equivalently, dividing violation, with a constant value. This relaxation will not change the relative magnitude among sensitivities, thus keeping optimization in the right direction.

The effect of constraint relaxation is illustrated in Fig. 4. As the constraint relaxation grows, the decap budget falls down and optimization time goes up. Constraint relaxation essentially refines optimization steps at the expense of slower convergence. However, after a certain point, increasing constraint factor would only slow down optimization at very little decap

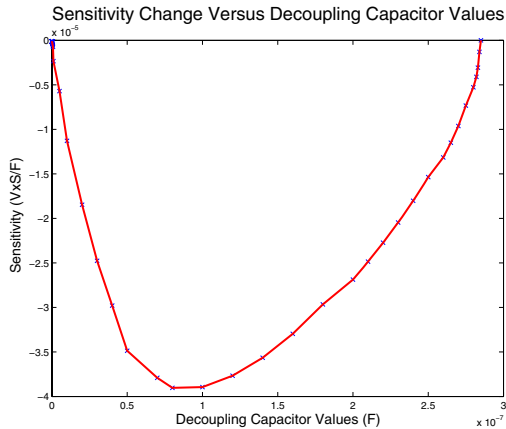


Fig. 3. Typical pattern of sensitivity change with added decap value

budget improvement. Usually a factor of 3 to 10 is used. A better strategy is to adjust this factor as optimization goes on. A fairly good heuristic strategy is to use a larger relaxation factor at the beginning and closing stage of optimization. This is motivated by the observation of sensitivity change in Fig. 3, since LP solver tends to overestimate decaps with smaller sensitivities at these two stages. Relaxation is not needed in the midway as it slows down the optimization without any benefit. Practically we can adjust the relaxation factor adaptively with the ratio of left violation area after adding some decaps to the original violation area before optimization.

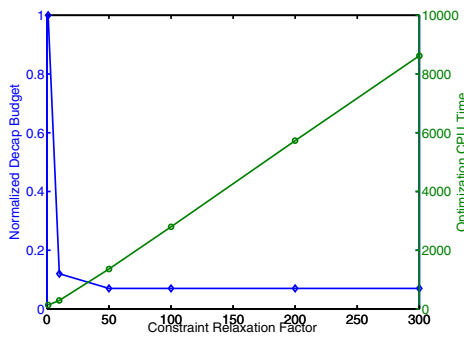


Fig. 4. Illustration of the constraint relaxation scheme

Such an SLP algorithm explicitly tries to minimize the decap budget, in contrast to most previous methods which minimize the violation area. One algorithm [6] tries to minimize violation area and decaps at the same time, but balancing of the two is difficult and has potential flaws [12]. In our algorithm, the use of the adjoint method, instead of the merged adjoint method, provides more flexibility for optimization and the results are expected to be closer to optimal. The only approximation here is the linearized sensitivity, but this problem is remedied somewhat with constraint relaxation and the iterative nature of SLP.

As for optimization efficiency, QP [17, 19] has a higher complexity and slower convergence rate than LP. CG [6] requires line search along each conjugate direction. Line search is extremely expensive since each objective evaluation equals one full circuit simulation. Also, the use of merged adjoint method provides great speedup but sacrifices optimization quality. The algorithm in [9, 12] is also based on the merged adjoint method, and further trades optimization quality for speeding up with a binary search. As it will be shown in section V, our SLP method, when combined with the partition strategy introduced in the next section, usually yields much smaller decap budget without much speed degradation.

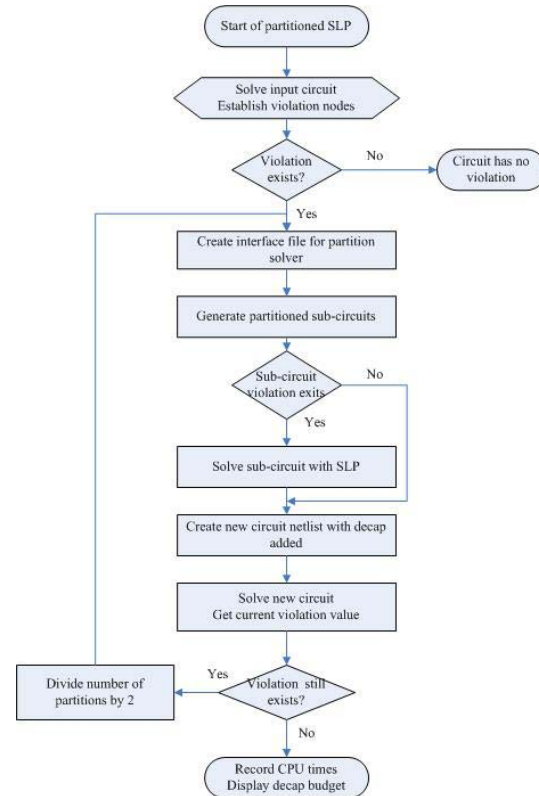


Fig. 5. The partitioning-based optimization flow

IV. PARTITIONING SCHEME FOR THE LOCALIZED DECAP ALLOCATION

A. Partition

All decap budgeting algorithms, especially sensitivity based ones described above, involve repeated transient analysis of the power grid. Since the adjoint method requires two circuit simulations to compute one sensitivity, computation time and memory storage can be prohibitive with a full-chip power grid model consisting of millions of nodes and devices. There exists some techniques for efficient analysis of huge P/G networks, like [4, 13, 10, 16], but they are either not accurate

enough, which is only efficient in DC analysis and has certain limitations, or consume great memory. So, for practical large industry P/G networks, the general divide-and-conquer strategy is desirable and has a great potential for parallel computation. In other words, the original circuit is partitioned into a number of sub-circuits, and each sub-circuit is optimized individually.

The partitioning strategy is especially suitable in the case of decap optimization, due to the localized effect of decaps for removing IR drops. In the newly emerged flip-chip packaging technology, this local effect becomes more prominent [3]. Violation nodes often cluster around several spots on the chip, and the number of them is usually only a small portion of the whole circuit. This is the typical situation, as decaps are usually added around power hungry models manually by designers. Optimization tools only fix some hot spots. If there are many violation nodes, the entire power network is probably not well designed in the first place, since relying on excessive decaps to remove IR drops for a badly designed P/G network would cause lots of side effects, such as excessive power consumption, unwanted resonance frequency and large leakage currents.

A fast partitioning strategy is preferred to deal with large circuits and we choose [8], a graph-based multilevel minimum cut algorithm, with which a million-vertex graph can be processed in one minute. This ensures that the partitioning step will scale to deal with very large circuits.

The boundary condition should always be taken care of, when partitioning is employed. To retain the communication among sub-circuits, the boundary node waveforms are converted to piece wise linear (PWL) voltage sources, thus the voltage waveforms remain the same, although each sub-circuit is simulated independently.

With this partitioning scheme, some partitions may not have any violation node, thus no optimization is needed and no decap will be added within these partitions. This makes sense since decaps should always be added near or around violation nodes. In many cases, this can be a great saving, since we often find that after partitioning, no simulation is needed for a great portion of the circuit. This further demonstrates the advantage of the divide and conquer strategy.

If not taken care of specially, violation nodes can appear on boundaries. Since their waveforms are treated as independent voltage sources, optimization is impossible. Therefore, violation nodes should be always be placed away from boundaries, which can be achieved by adjusting weighting factors of edges and vertices to influence the partitioning engine as done in [9].

B. Partitioning-based Decap Optimization Flow

The whole partitioning-based optimization flow is given in Fig 5. In each outer optimization iteration, simulation of the whole circuit is only performed twice: one at the beginning to record violation nodes and boundary nodes waveforms, and the second time at the end to verify that all violations are eliminated. In practice, most circuits are optimized within only one or two optimization cycles. Both the adjoint method based

sensitivity calculation and the SLP optimization are applied to each sub-circuit individually, but communication between partitions is well captured by boundary conditions. If parallel computing is employed, optimization of each sub-circuits can be carried out at the same time and efficiency can be further improved.

V. EXPERIMENTAL RESULTS

We implemented the proposed algorithm in C++. All experiments are carried out on a Linux PC with dual 3.0Ghz Xeon CPUs and 2GB memory. Test circuits are generated by the authors with realistic parameters for R, C and current sources based on industry designs. The off-chip inductive parasitic effects are also considered. Notice that our algorithm is general enough for any specific circuit or structure. The power noise tolerance is a user-specified parameter.

We compared our algorithm with the most recently published decap budgeting algorithm [9] on a number of P/G networks ranging from hundreds of nodes to half a million nodes. Table I summarizes the comparison, where *Localized CG* denotes the localized CG with the binary search method in [9], and *localized SLP* denotes the proposed localized SLP optimization method. The first four columns represent circuit names, the number of total nodes, violation nodes and partitions respectively. The next four columns compare optimization time and decap budgets. For a better comparison, decap budgets yielded by [9] are always normalized to 1. Parameters, including the voltage drop tolerance, and the maximum decap at each node, can be specified by users and are the same for both methods. For all these circuits, noise elimination is successfully achieved after optimization.

As can be seen from Table I, the localized SLP algorithm optimizes all circuits successfully. Although the speed is usually 3-5 times slower than the CG-binary search combined algorithm in [9], the reduction in decap budgets is impressive. For all circuits, our algorithm yields significantly smaller decap budgets, yet achieves the same violation elimination. For one of the test circuits, a reduction of 90% decap area is achieved. This means significant chip area saving, much less leakage power and cheaper manufacture costs.

The reduction of decap area is mainly due to the explicit decap minimization formulated as the objective in our problem, and the recovery of the quality loss introduced by merged adjoint methods [7, 9]. However, this comes at the expense of computation costs. To obtain sensitivity of violation to each of the decap candidate node, more simulations are needed, but this is compensated by the efficiency of linear programming over quadratic or conjugate gradient optimization. With partitioning, The speed is basically kept at the same level as [9].

As the partition number increases, optimization time decreases, but the decap budget does not change monotonously. Relationship of decap budgets and the partition number was discussed in [9]. For the SLP optimization, keeping sub-circuit size from being too large is important, otherwise the sensitivity calculation would be very slow. Usually several hundred nodes

TABLE I
COMPARISON BETWEEN THE EXISTING LOCALIZED DECAP OPTIMIZATION METHODS

Circuit	Total Nodes #	Violation Nodes #	Part. #	Localized CG		Localized SLP	
				Decap	Time(sec)	Decap	Time(sec)
ckt1	185	4	2	1	12	0.46	17
ckt2	848	14	4	1	2	0.36	9
ckt3	6015	30	4	1	19	0.51	169
ckt4	8,993	109	50	1	20	0.31	88
ckt5	29,425	181	30	1	63	0.10	282
ckt6	89,496	251	100	1	187	0.24	1078
ckt7	123,280	301	100	1	263	0.26	1049
ckt8	536,705	573	100	1	1226	0.28	3704

are appropriate.

The effect of constraint relaxation on optimization speed and has been demonstrated in Fig. 4. For all test circuits, we applied adaptive constraint relaxation mentioned in Sec. III C. In reality, constraint relaxation provides an ideal means to fine-tune the tradeoff between optimization speed and quality.

VI. CONCLUSIONS

This paper proposes a localized sequence of linear programming optimization flow for on-chip decoupling capacitors allocation. In our problem formulation, decap budget is explicitly minimized as the objective function. Compared to existing conjugate gradient and other nonlinear programming methods, the sequence of linear programming based method, with the time-domain adjoint method, demonstrates better solution quality. The partitioning strategy improves the scalability of the algorithm and makes it efficient for large circuits. The proposed algorithm is general enough for any P/G network. Experimental results show that for circuits without too many violation nodes, which is usually the case for well designed P/G grids, the proposed algorithm usually yields much smaller decap area at a mildly larger computation cost than the most recently published decap budgeting algorithms.

REFERENCES

- [1] 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.
- [2] 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.
- [3] E. Chiprout, "Fast flip-chip power grid analysis via locality and grid shells," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, Nov. 2004, pp. 485–488.
- [4] E. Chiprout and T. Nguyen, "Power analysis of large interconnect grids with multiple sources using model reduction," in *Proc. European Conference on Circuit Theory and Design*, Sept. 1999.
- [5] S. W. Director and R. A. Rohrer, "Automated network design – the frequency-domain case," *IEEE Trans. on Circuit Theory*, vol. 16, no. 3, pp. 330–337, Aug. 1969.

- [6] 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," vol. E87-A, no. 12, pp. 3273–3280, Dec. 2004.
- [7] —, "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.
- [8] G. Karypis, R. Aggarwal, and V. K. S. Shekhar, "Multilevel hypergraph partitioning: application in VLSI domain," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, pp. 69–79, March 1999.
- [9] H. Li, Z. Qi, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong, "Partitioning-based approach to fast on-chip decap budgeting and minimization," in *Proc. Design Automation Conf. (DAC)*, June 2005, pp. 170–175.
- [10] S. R. Nassif and J. N. Kozhaya, "Fast power grid simulation," in *Proc. Design Automation Conf. (DAC)*, 2000, pp. 156–161.
- [11] M. Pant, P. Pant, and D. Wills, "On-chip decoupling capacitor optimization using architectural level current signature prediction," in *Proc. IEEE Midwest Symp. Circuits and Systems*, 2000, pp. 772–775.
- [12] Z. Qi, H. Li, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong, "Fast decap allocation algorithm for robust on-chip power delivery," in *Proc. Int. Symposium on Quality Electronic Design (ISQED)*, 2005, pp. 542–547.
- [13] H. F. Qian, S. R. Nassif, and S. S. Sapatnekar, "Random walks in a supply network," in *Proc. Design Automation Conf. (DAC)*, 2003, pp. 93–98.
- [14] 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.
- [15] L. Smith, "Decoupling capacitor calculations for cmos circuits," in *Proc. IEEE Topical Meeting of Electrical Performance of Electronic Packaging*, 1994, pp. 101–105.
- [16] H. Su, E. Acar, and S. R. Nassif, "Power grid reduction based on algebraic multigrid principles," in *Proc. Design Automation Conf. (DAC)*, 2003, pp. 109–112.
- [17] 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.
- [18] X.-D. Tan, C.-J. Shi, D. Lungeanu, and J.-C. Lee, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 12, pp. 1678–1684, Dec. 2003.
- [19] K. Wang and M. Marek-Sadowska, "On-chip power supply network optimization using multigrid-based technique," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 407–417, Mar. 2005.