
Volume 40, Issue 4
Completing Volume 40
July 2007
ISSN 0167-9260
ELSEVIER

INTEGRATION
the VLSI journal

SPECIAL ISSUE:
SYSTEM-LEVEL INTERCONNECT PREDICTION
GUEST EDITORS: IGOR L. MARKOV, LOUIS K. SCHEFFER and DIRK STROOBANDT

CONTENTS

Special issue on System-Level Interconnect Prediction
I.L. Markov, L.K. Scheffer and D. Stroobandt 381

Predicting reconfigurable interconnect performance in distributed shared-memory systems
W. Heirman, J. Dambre, I. Artundo, C. Debaes, H. Thienpont, D. Stroobandt and J. Van Campenhout 382

Waveform analysis and delay prediction for a CMOS gate driving *RLC* interconnect load
B.K. Kaushik, S. Sarkar and R.P. Agarwal 394

APWL-Y: An accurate and efficient wirelength estimation technique for hexagon/triangle placement
Y. Wei, S. Dong and X. Hong 406

A novel net-degree distribution model and its application to floorplanning benchmark generation
T. Wan and M. Chrzanowska-Jeske 420

Predictions of CMOS compatible on-chip optical interconnect
G. Chen, H. Chen, M. Haurylau, N.A. Nelson, D.H. Albonesi, P.M. Fauchet and E.G. Friedman 434

Regular papers

Substrate model extraction using finite differences and parallel multigrid
J.M.S. Silva and L.M. Silveira 447

[Continued on outside back cover

Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com
ScienceDirect
INTEGRATION, the VLSI journal 40 (2007) 516–524
INTEGRATION the VLSI journal
www.elsevier.com/locate/vlsi

Partitioning-based decoupling capacitor budgeting via sequence of linear programming ☆

Jeffrey Fan[a,*], Sheldon X.-D. Tan[a], Yici Cai[b], Xianlong Hong[b]

[a] Department of Electrical Engineering, University of California, Riverside, CA 92521, USA
[b] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Received 23 January 2006; received in revised form 16 October 2006; accepted 16 October 2006

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) as the optimization engine, and partitioning scheme for dealing with large-sized circuits. We show that by directly optimizing the decoupling capacitor (decap) areas as the objective function and using the time-domain adjoint method, SLP can deliver much better quality in terms of decap budget 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 larger 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.

Keywords: Power grid network; Decoupling capacitor (decap); Sequence of linear programming

1. 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 (for example, say 10%) of nominal values. Excessive voltage drops and ground bounces not only may 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 [1–8]. Besides early stage design techniques, such as 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. The modeling of power grid network is shown in Fig. 1. Conceptually thinking, 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 MOSFET transistors, and excessive on-chip decaps would not only consume on-chip area, but also cause more leakage power, lower yield, and lower resonant frequency [1]. Therefore, as long as power supply noises are constrained, decaps should be minimized. Note that the main purpose of adding decap is not to save the

☆ Some preliminary results of this paper were presented at Sixth International Conference on ASIC (ASICON'05), Beijing China [22] and International Symposium on Quality Electronic Design (ISQED'06), San Jose, CA [23]. This work was sponsored by NSF CAREER Award CCF-0448534, NSF REU CCF-0529754 and NSF OISE-0451688 and UC Micro #05-111 via Cadence Design Systems, Inc.
* Corresponding author. Tel.: +1 951 827 6161.
E-mail address: jfan@ee.ucr.edu (J. Fan).

0167-9260/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2006.10.002

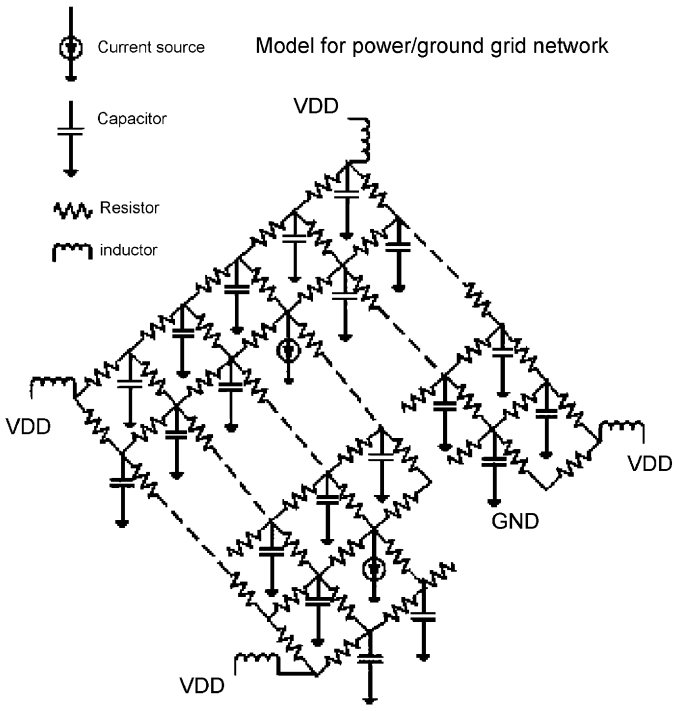

Fig. 1. The model of power grid network.

chip area. Instead, we try to reduce the voltage noises (or IR drops) on the power delivery networks. Also, decaps can help reducing the delay, due to the reduction of voltage drops. At the same time, we want to use as smaller decap as possible as decap will introduce more power consumption and occupy more chip resources (white space, WS). Actually, decaps may induce more leakage currents. Therefore, we should use them economically, which is the main goal of this paper.

The optimization of decap placement has been extensively studied in the past [2,4–9]. Some earlier works [2,9] 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 (QP) [4] or sequential quadratic programming (SQP) [6], conjugate gradient (CG) [5] or CG combined with binary search [7,8].

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 methods that perform transient simulations in internal loops will be prohibitive. To combat this, earlier works [6,10] proposed a method to reduce the P/G grid first and then apply standard optimization techniques. For larger circuits, some previous work [8] partitions the circuit into smaller sub-circuits before optimizing them individually.

In this paper, we propose an efficient decap allocation algorithm, which explicitly minimizes the decap areas 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. We only address the decap optimization for voltage drop here. However, the situation of ground bounces can be easily dealt within a similar fashion. To achieve higher efficiency with large circuits, a partition strategy similar to [8] 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 [8].

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 3 formulates decap budgeting into an SLP problem. Section 4 introduces the partition strategy and presents the flow of our partitioning-based SLP optimization. Experimental results are presented in Section 5, and Section 6 concludes the paper.

2. Problem formulation and review of previous methods

We assume that the circuit dynamics, modeled as piecewise linear (PWL) current sources, are given already. Those PWL current sources can be obtained by performing the logic or circuit simulations of the circuits (assuming ideal voltage supply networks). The timing issue in different blocks and storage elements will be considered during those logic/circuit simulations. If those simulations represent the typical circuit operations, the current sources may give good indications of the actual voltage drops over the time. The same assumption was also made in the existing decap allocation methods [2,4–9].

Existing on-chip decap budgeting algorithms basically fall into two categories [7]. One category [2,9,11,12] is to compute the current pattern around nodes where excessive IR drops occur, and then 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, it cannot allocate decaps in an optimal way.

The other category is based on sensitivity computation [4–8]. Those method compute the sensitivity of objective functions with respect to the variable decaps using SPICE-type transient simulation results. Fig. 2 illustrates the V_{dd} fluctuation at a node within one clock cycle. The *violation area* at node j is defined as

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

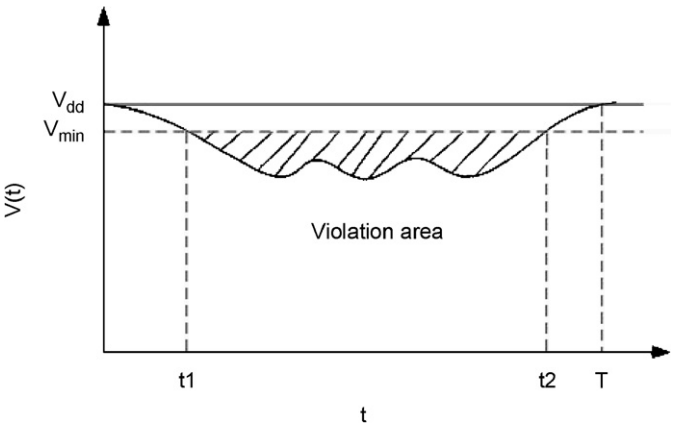

Fig. 2. Illustration of IR drop violation.

which equals the shaded area below the designer-defined V_{dd} threshold, V_{min}, in the figure. In this paper, a node is a physical node in a SPICE netlist where we can compute the nodal voltage.

The contribution of decap added at node i to removing violation at node j can be measured by *sensitivity*:

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

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 [13] can be readily used for sensitivity calculation, where two circuit simulations are involved, one for the original network and the other for the adjoint network. The sensitivity for capacitive elements can then be expressed as

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

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:

$$\min \sum_{j=1}^{m} g_j(c_1, \ldots, c_n). \tag{4}$$

Maximum decap constraint:

$$(c_i) \leqslant d_i, \quad d_i \geqslant 0. \tag{5}$$

Although there are other constraints depending upon the layout style, our formulation is general enough to be adapted to specific layout, as those constraints are typically expressed as linear constraints in terms of decap width or length (thus its values are also linear).

Recent work [10] improves the sensitivity-based decap allocation algorithm by using time-domain merged adjoint method for fast sensitivity calculation and explicitly considering the decap area optimization in the objective function. The merged adjoint method allows the sensitivity to be computed directly for the objective function instead of individual nodes. The CG optimization method was used to minimize the following objective function iteratively, which considers decap area explicitly:

$$\sum_{i \in \text{all nodes}} (\Delta c_i) + \alpha \sum_{j \in \text{all nodes}} g_j, \tag{6}$$

where the weighting factor α will keep changing in each CG iteration. Notice that in (6) we used Δc_i instead of c_i to denote added decaps at each iteration.

A drawback with CG method is the slow convergence of CG optimization. Theoretically, it takes n steps before CG converges, where n is the number of variables [14]. Moreover, α is estimated after each iteration by equating the two summation items in (6) [10]. The idea behind this is to let CG reduce violation area and decap area at the same time. However, such balance can be misleading for optimization—sometimes it tends to inflate unimportant part and renders optimization less efficient. For example, at later stages when there is very little violation left, a little bit more decap would suffice, but with a very large α, optimization is not very well oriented, because the direction of CG is decided by gradient of the objective function which is mainly determined by the value of α, instead of the sensitivity of violation with respect to decap itself.

More importantly, in practice, selection of α proves to be tricky and difficult. In order for CG to perform optimization efficiently, an objective function is supposed to be convex on its domain. If (6) is monotonously decreasing, line search for the minimum would end up with a full step and stop at the far end, adding decaps for all nodes with nonzero sensitivities and the maximum allowed decap for the node with the greatest sensitivity. This often leads to overestimation of the necessary decap. Even worse, when (6) is monotonously increasing, line search always ends at the starting point, and optimization is stuck at the current stage, since no decap would be added. The former corresponds to α being too large which favors violation too much, where the later corresponds to a too small α, favoring decap area too much.

Basically, all methods [4–8] follow the problem formulation scheme mentioned earlier. They only differ from one another in the optimization technique applied thereafter. The method in [4] applies the Lagrangian relaxation technique and feeds it into standard QP solver. Unfortunately, introducing Lagrangian variables increases the problem size and the complexity of QP is relatively high. Some [6] proposes to coarsen the circuit with the standard multigrid (SMG) scheme and then applies sequential quadratic programming (SQP) to optimize the reduced problem, and then 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 [15]. It also

has more limitations in the practical decap optimization scenarios [6].

Some previous work [5] develops a *merged adjoint method* from the adjoint method and bases its CG optimization on *merged sensitivity*. The idea is that, instead of computing s_{ij}, we compute the merged sensitivity

$$\sum_{j=1}^{m} s_{ij} = \frac{\partial \sum_{j=1}^{m} g_j(c_1, \ldots, c_n)}{\partial c_i} \tag{7}$$

by applying the superposition law for linear circuits in the adjoint method

$$\sum_{j=1}^{m} s_{ij} = \int_0^T (v'_{i,\mathrm{all}}(T-t)) \times \dot{v}_i(t)\,\mathrm{d}t \quad (i = 1, 2, \ldots, n). \tag{8}$$

One method [7] modifies the objective function described in [5] and combines the binary search with CG, thus it achieves significant speedup. The work in [8] further proposes a partitioning strategy before the optimization of each sub-circuit. However, using the merged sensitivity is a tradeoff for speed, as pointed out in [8], certain information is lost and optimization results are nonoptimal.

In this work, we apply the SLP method which directly minimizes decap budgets. Our approach is still sensitivity-based approach and require the SPICE-type simulations of original and adjoint networks in each iteration step. SLP was applied to sizing the P/G grid network before and was shown to be more efficient than CG methods [3]. To deal with large circuits, we follow the partition strategy in [8] to take advantage of the localized effects of adding decaps.

3. Decap problem formulated into SLP

In this section, we show how SLP can be applied to decap budgeting.

3.1. 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. The 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 CG optimization in terms of convergence rate [3].

3.2. Problem formulation

In contrast to the problem formulation described in Section 2, we choose decap areas 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 areas is equivalent to the minimization of total decap value.

Objective function:

$$\min \sum_{i \in \text{all nodes}} (c_i). \tag{9}$$

Violation elimination constraint:

$$\begin{aligned} g_j(c_1, \ldots, c_n) &= \int_0^T \max(V_{\min} - v_j(t), 0)\,\mathrm{d}t \\ &= \int_{t_{1(j)}}^{t_{2(j)}} (V_{\min} - v_j(t))\,\mathrm{d}t \\ &= 0. \end{aligned} \tag{10}$$

Maximum decap constraint:

$$(c_i) \leqslant d_i, \quad d_i \geqslant 0, \tag{11}$$

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

If the required minimum decap area is larger than the maximum decap allowed due to limited WSs, the optimization process will lead to no valid solution as we cannot remove all the violations in this case. As a result, designers have to alter the floorplan or the placement such that more WS will be allocated around the decap area.

In our problem, the only nonlinear portion is the constraint (10). 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. From (2), the first order approximation, g_j^*, can be expressed as

$$g_j^* = \sum_{i \in \text{all nodes}} s_{ij} \Delta c_i. \tag{12}$$

The violation area is eliminated when $g_j - g_j^* \leqslant 0$. As a result, to meet with the nonlinear violation constraint (10) at violation node j, we have the following linearized constraint for added decaps Δc_i:

$$g_j \leqslant g_j^* = \sum_{i \in \text{all nodes}} s_{ij} \Delta c_i. \tag{13}$$

Replacing constraint Eq. (10) with (13), we end up with a linear programming problem. Iteration continues until all violation is eliminated or no solution can be found.

3.3. 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. More 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

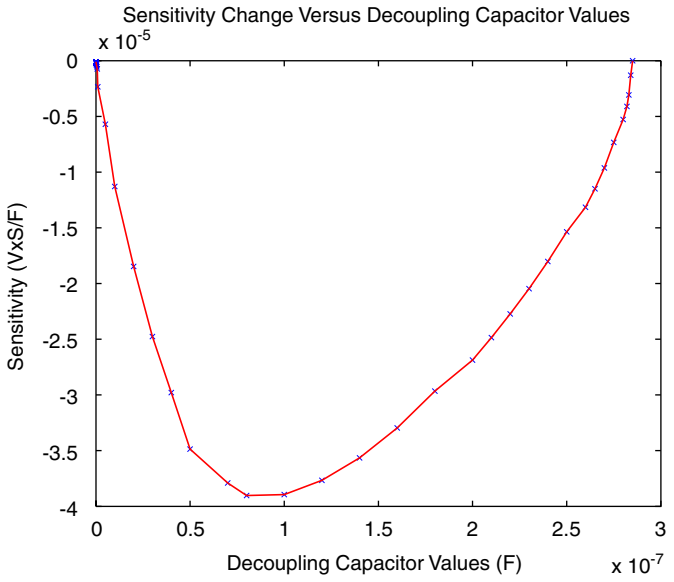

Fig. 3. Typical pattern of sensitivity change with added decap value.

a peak at some point and then goes down to zero when the violation goes away. This implies that we would overestimate decaps based on the smaller sensitivity calculated at beginning and closing stages.

Two problems arise. First, from (13), 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 (11). Second, we may overestimate decaps. This actually explains why existing sensitivity based methods like [4], 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 budget improvement. Usually a factor of 3–10 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

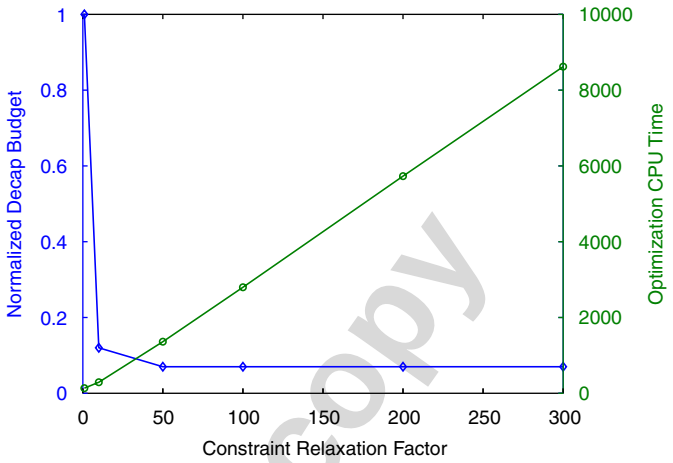

Fig. 4. Illustration of the constraint relaxation scheme.

some decaps to the original violation area before optimization.

Such an SLP algorithm explicitly tries to minimize the decap budget, in contrast to most previous methods which minimize the violation area. One algorithm [5] tries to minimize violation area and decaps at the same time, but balancing of the two is difficult and has potential flaws [7]. 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 [4,6] has a higher complexity and slower convergence rate than LP. CG [5] 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 [7,8] 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 5, our SLP method, when combined with the partition strategy introduced in the next section, usually yields a much smaller decap budget without much speed degradation.

4. Partitioning scheme for the localized decap allocation

4.1. 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 [15–18], 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 [19]. 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 [20], a graph-based multilevel minimum cut algorithm, with which a million-vertex graph can be processed in less than 1 min. 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 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. Thus, the scheme of partitioning 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 [8].

4.2. 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.

5. Experimental results

We implemented the proposed algorithm in C++. All experiments are carried out on a Linux PC with dual 3.0 GHz Xeon CPUs and 2 GB memory. We use SuperLU as the linear matrix solver [21] for all the test cases. 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 [8] on a number of P/G networks ranging from hundreds of nodes to half a million nodes. The LU decomposition (SuperLU package) is used as the matrix solver to solve the state equations for both algorithms in our experiment. Table 1 summarizes the comparison, where *CG* denotes the localized CG with the binary search method in [8], and *SLP* denotes the proposed localized SLP optimization method. The first four columns represent circuit names, the number of total nodes, violation nodes, and the number of partitions, respectively. The next four columns compare optimization time and decap budgets. 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.

The available WS in the circuit is the sum of WSs available in each node. Thus, the constraint of WS in terms of decap budget in each node is a parameter specified by users based on the given layout. We use the same parameter value for both CG and SLP methods for comparison. As it can be seen from Table 1, 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 [8], 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 manufacturing costs.

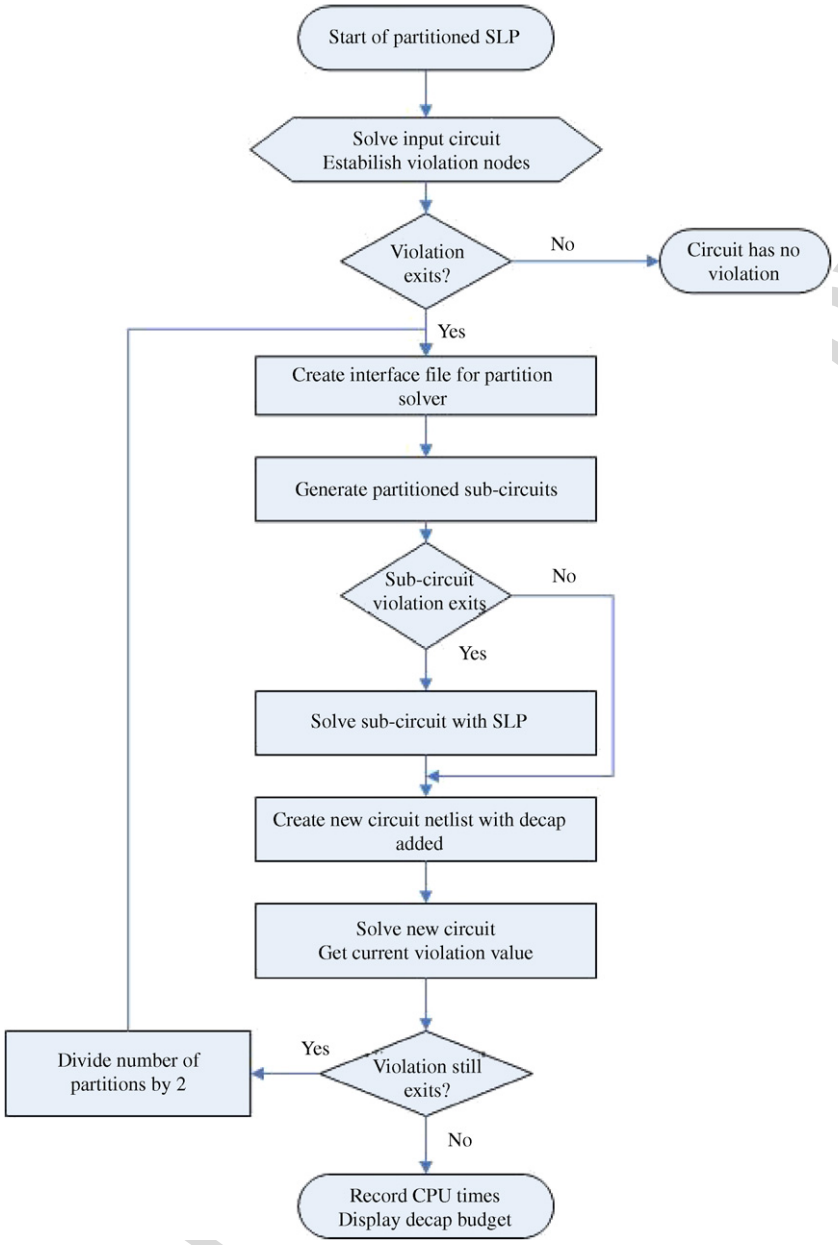

Fig. 5. The partitioning-based optimization flow.

Table 1
Comparison between existing partitioned decap optimization methods

Circuit name	Total nodes	Vio. nodes	No of part.	CG (F)	CG (s)	SLP (F)	SLP (s)	SLP/CG budget (%)
ckt1	185	4	2	3.78e − 7	12	1.76e − 7	17	0.46
ckt2	848	14	4	3.16e − 7	2	1.13e − 7	9	0.36
ckt3	6105	30	4	3.75e − 8	19	1.90e − 8	169	0.51
ckt4	8993	109	50	1.24e − 7	20	3.86e − 8	88	0.31
ckt5	29425	181	30	9.18e − 7	63	8.74e − 8	282	0.10
ckt6	89496	251	100	3.12e − 7	187	7.51e − 8	1078	0.24
ckt7	123280	301	100	1.63e − 7	263	4.21e − 8	1049	0.26
ckt8	536705	573	100	7.97e − 8	1226	2.22e − 8	3704	0.28

The reduction of decap area is mainly due to the explicit decap minimization formulated as the objective function in our problem, and the recovery of the quality loss introduced by merged adjoint methods [8,10]. 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 CG optimization. With partitioning, the speed is basically kept at the same level as [8].

As the partition number increases, optimization time decreases, but the decap budget does not change monotonously. The relationship of decap budgets and the partition number was discussed in [8]. For the SLP optimization, keeping sub-circuit size from becoming too large is important, otherwise the sensitivity calculation will be very slow. Usually several hundred nodes are appropriate.

The effect of constraint relaxation on optimization speed has been demonstrated in Fig. 4. For all test circuits, we applied adaptive constraint relaxation mentioned in Section 3.3. In reality, the constraint relaxation provides an ideal means to fine-tune the tradeoff between optimization speed and quality.

6. Conclusions

This paper has proposed a localized SLP optimization flow for on-chip decaps allocation [22,23]. In our problem formulation, decap budget is explicitly minimized as the objective function. In comparison to existing CG and other nonlinear programming methods, the SLP 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 showed 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. Moreover, one consideration is that the modeling of PWL current sources as the current sinks may be stochastic in nature. One possible direction of our future researches may include the statistical analysis of the variations in those current sources.

Acknowledgments

The authors would like to thank Zhenyi Qi, Hang Li, and I-Fan Liao for their contributions to this paper and the reviewers for their constructive suggestions to improve the presentation of this paper.

References

[1] S. Bobba, T. Thorp, K. Aingaran, D. Liu, IC power distribution challenges, in: Proceedings of International Conference on Computer Aided Design (ICCAD), 2001, pp. 643–650.
[2] S. Zhao, K. Roy, C.K., Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning, IEEE Trans. Computer-Aided Des. Integrated Circuits Syst. 21(1) (2002) 81–92.
[3] X.-D. Tan, C.-J. Shi, D. Lungeanu, J.-C. Lee, Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings, IEEE Trans. Computer-Aided Des. Integrated Circuits Syst. 22 (12) (2003) 1678–1684.
[4] H. Su, S.S. Sapatnekar, S.R. Nassif, Optimal decoupling capacitor sizing and placement for standard cell layout designs, IEEE Trans. Computer-Aided Des. Integrated Circuits Syst. 22 (4) (2003) 428–436.
[5] J. Fu, Z. Luo, X. Hong, Y. Cai, S.X.-D. Tan, Z. Pan, A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery, IEICE Trans. Fund. Electron. Commun. Comput. Sci. (IEICE) E87-A (12) (2004) 3273–3280.
[6] K. Wang, M. Marek-Sadowska, On-chip power supply network optimization using multigrid-based technique, IEEE Trans. Computer-Aided Des. Integrated Circuits Syst. 24 (3) (2005) 407–417.
[7] Z. Qi, H. Li, S.X.-D. Tan, L. Wu, Y. Cai, X. Hong, Fast decap allocation algorithm for robust on-chip power delivery, in: Proceedings of International Symposium on Quality Electronic Design (ISQED), 2005, pp. 542–547.
[8] H. Li, Z. Qi, S.X.-D. Tan, L. Wu, Y. Cai, X. Hong, Partitioning-based approach to fast on-chip decap budgeting and minimization, in: Proceedings of Design Automation Conference (DAC), 2005, pp. 170–175.
[9] H.H. Chen, D.D. Ling, Power supply noise analysis methodology for deep-submicron VLSI chip design, in: Proceedings of Design Automation Conference (DAC), 1997, pp. 638–643.
[10] J. Fu, Z. Luo, X. Hong, Y. Cai, S.X.-D. Tan, Z. Pan, A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery, in: Proceedings of Asia South Pacific Design Automation Conference (ASPDAC), 2004, pp. 505–510.
[11] L. Smith, Decoupling capacitor calculations for cmos circuits, in: Proceedings of IEEE Topical Meeting of Electrical Performance of Electronic Packaging, 1994, pp. 101–105.
[12] M. Pant, P. Pant, D. Wills, On-chip decoupling capacitor optimization using architectural level current signature prediction, in: Proceedings of IEEE Midwest Symposium Circuits and Systems, 2000, pp. 772–775.
[13] S.W. Director, R.A. Rohrer, Automated network design—the frequency-domain case, IEEE Trans. Circuit Theory 16 (3) (1969) 330–337.
[14] G.H. Golub, C.F.V. Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[15] H. Su, E. Acar, S.R. Nassif, Power grid reduction based on algebraic multigrid principles, in: Proceedings of Design Automation Conference (DAC), 2003, pp. 109–112.
[16] E. Chiprout, T. Nguyen, Power analysis of large interconnect grids with multiple sources using model reduction, in: Proceedings of European Conference on Circuit Theory and Design, 1999.
[17] S.R. Nassif, J.N. Kozhaya, Fast power grid simulation, in: Proceedings of Design Automation Conference (DAC), 2000, pp. 156–161.
[18] H.F. Qian, S.R. Nassif, S.S. Sapatnekar, Random walks in a supply network, in: Proceedings of Design Automation Conference (DAC), 2003, pp. 93–98.
[19] E. Chiprout, Fast flip-chip power grid analysis via locality and grid shells, in: Proceedings of International Conference on Computer Aided Design (ICCAD), 2004, pp. 485–488.
[20] G. Karypis, R. Aggarwal, V.K.S. Shekhar, Multilevel hypergraph partitioning: application in VLSI domain, IEEE Trans. Very Large Scale Integration (VLSI) Syst. 7 (1) (1989) 69–79.
[21] SuperLU, 2006. Superlu v3.0. ⟨http://crd.lbl.gov/xiaoye/SuperLU/⟩.
[22] J. Fan, I. Liao, S.X.-D. Tan, Y. Cai, X. Hong, Localized on-chip power delivery network optimization via sequence of linear programming, in: Proceedings of International Symposium on Quality Electronic Design (ISQED), 2006, pp. 272–277.
[23] Z. Qi, J. Fan, H. Li, S.X.-D. Tan, Y. Cai, X. Hong, On-chip decoupling capacitor budgeting by sequence of linear programming, in: IEEE International Conference on Application Specific Integrated Circuits (ASICON), 2005, pp. 70–73.

Jeffrey Fan received his Bachelor of Science degree in electronics engineering from National Chiao Tung University, Taiwan, ROC, and Master of Science degree in electrical engineering from State University of New York, Buffalo, in 1983 and 1987, respectively. He is currently a Ph.D. candidate in electrical engineering at the University of California, Riverside.

From 1988 to 2002, he held various senior technical positions at Western Digital, Emulex Corporation, Adaptec Inc., and Toshiba America. He served as Vice President of Vivavr Technology, Inc., and General Manager/Co-Founder of Musica Technologies, Inc. His research interests include very large scale integration simulation, modeling, and power grid optimization.

Sheldon X.-D. Tan received his Bachelor of Science and Master of Science degrees in electrical engineering from Fudan University, Shanghai, China in 1992 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Iowa, Iowa City, in 1999.

He is an Associate Professor in the Department of Electrical Engineering, University of California at Riverside. He was a faculty member in the Electrical Engineering Department of Fudan University from 1995 to 1996. He worked for Monterey Design Systems Inc. CA, from 1999 to 2001, and Altera Corporation CA, from 2001 to 2002. His research interests include several aspects of design automation for VLSI integrated circuits — modeling and simulation of analog/RF/mixed-signal VLSI circuits, high performance power and clock distribution network simulation and design, signal integrity, power modeling, thermal modeling, thermal optimization in VLSI physical and architecture levels and embedded system designs based on FPGA platforms.

Dr. Tan is the recipient of NSF CAREER Award in 2004. He also received the UC Regent's Faculty Fellowship in 2004. Dr. Tan received a Best Paper Award Nomination from 2005 IEEE/ACM Design Automation Conference, Best Paper Award from 1999 IEEE/ACM Design Automation Conference. He also co-authored the book ''Symbolic Analysis and Reduction of VLSI Circuits'' by Springer/Kluwer 2005. He is an associate editor for Journal of VLSI Design and served as a technical program committee member for ASPDAC, BMAS, ASPDAC, ISQED, and ICCAD.

Yici Cai received Bachelor of Science degree in Electronic Engineering from Tsinghua University in 1983, and Master of Science degree in Computer Science & Technology from Tsinghua University in 1986, She has been a professor in the Department of Computer Science & Technology, Tsinghua University, Beijing, China. Her research interests include signal integrity issues and DFM in VLSI physical design, power/ground distribution network design and optimization, high performance clock network design and optimization.

Xianlong Hong graduated in computational mathematics from Tsinghua University of Beijing in 1964. Since 1964, he joined Tsinghua University and now a professor in the department of computer science and technology. He was a visiting scholar in University of California, Berkeley, from 1991 to 1992 and in University of California, Los Angeles in 1995.

As a technical leader, he organized and managed to develop three generations of VLSI CAD system, the national projects in China from 1981 to 1991. Due to his contribution in developing VLSI CAD systems, he was awarded more then 15 Science and Technology Achievement Prizes from The Science and Technology Ministry and Education Ministry of China. His research interests are in physical design for VLSI circuits. He has authored and co-authored over 200 papers and 5 books around his area. He is an IEEE Fellow and served as an associate editor for IEEE Trans. on CAS and TPC co-chair of ASPDAC in 1999, 2004 and 2005.