A Fast Decoupling Capacitor Budgeting Algorithm for Robust On-Chip Power Delivery*

Jingjing Fu[1], Zuying Luo[1], Xianlong Hong[1], Yici Cai[1], Sheldon X.-D. Tan[2], Zhu Pan[1]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, P.R.China
[2]Department of Electrical Engineering, University of California at Riverside, Riverside CA, USA
Contact Author: Jingjing Fu, fujingjing00@mails.tsinghua.edu.cn

Abstract—In this paper, we present an efficient method to budget on-chip decoupling capacitors (decaps) to optimize power delivery networks in an area efficient way. Our algorithm is based on an efficient gradient-based non-linear programming method for searching the solution. Our contributions are an efficient gradient computation method (time-domain merged adjoint network) and a novel equivalent circuit modeling technique to speed up the optimization process. Experimental results demonstrate that the algorithm is capable of efficiently optimizing very large scale P/G networks.

I Introduction

Signal integrity in VLSI is emerging as a limiting factor in the nano-regime VLSI chip designs as technology scales. This is especially true on global networks like power/ground (P/G) networks where noise margins have been reduced greatly in VLSI designs due to decreasing supply voltages and the presence of excessive voltage drops coming from resistive and inductive effects. For dynamic voltage fluctuations on a P/G network, adding decap is regarded as the most efficient way to reduce such noises [7][8]. Therefore, some spare chip area will be reserved for decaps in the floorplanning and placement phases. In this paper, we focus standard-cell like ASIC layout structures as shown in Fig.1 where the white space are the spare space which can be used as for adding decaps for noise reduction of P/G networks as well as for adding buffers for optimizing timing in global signal networks and clock networks. As a result, given the same voltage fluctuation constraints, smaller decap area is more desirable as more spare space can be used for buffering in the later routing phases. The problem we try to solve in this paper is how to minimize the decap areas subject to the important P/G network integrity constraints, which include voltage fluctuation, electromigration and other design rules.

Due to the importance of reducing noise on the P/G networks, many significant research efforts have been carried out in the past to reduce noises while avoiding using excessive areas [1]-[8]. Methods in [2]-[6]size wire segments in P/G grids for optimizing area of P/G networks. Among them, conjugate gradient based method in [5] and equivalent circuit modeling method [6] can only be applied to resistor-only P/G networks. Methods proposed in [1][7][8] allocate and optimize decaps to reduce transient noises in P/G networks. Papers [1][7] focus on the full custom design styles and [8] targets the standard cell layout style where quadratic programming is used to allocate decaps without explicitly optimizing the area. Recently a multigrid-based decap allocation approach was proposed in [18]. But the method can only be applied to very regular mesh-structured P/G networks due to the geometric oriented grid complexity reduction process. Furthermore, the important capacitive and inductive effects on P/G wires were also ignored (which actually are difficult to be incorporated) in the method.

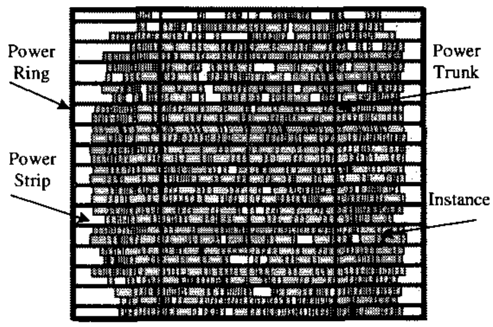

Fig. 1. A P/G network of standard cell based ASICs.

In this paper, we propose an efficient approach to reduce transient noises on P/G networks via decap allocation in an area efficient way based on standard-cell layout style. In our method, we model each segment of the power grid as lumped RLC element, which is in contrast to [8] and [18] where P/G wire segments are modeled as RC/R components. Parasitics from power pads and packages are also modeled as lumped RLC elements similar to[1]. The nonlinear devices or modules are modeled as time-varying current sources, which can be obtained by off-line logic simulation. We also consider both built-in on-chip decoupling capacitors (n-well capacitors and circuit capacitors) and add-on decoupling capacitors (thin-oxide capacitors) connected between power and ground. We haven't considered the mutual inductance in this work. And it will be the goal of our future work.

Different from [8], we explicitly consider area reduction as the primary objective as spare areas in a chip are premium resources and should be used more economically for different physical optimization processes. For instance, the left white space can be further allocated for buffering to improve timing in the later phases of physical design. The approach also considers electron-migration

*This work is part of the Project 60176016 Supported by National Science Foundation of China and is supported by the National Natural Science Foundation of China 60121120706, National Natural Science Foundation of USA CCR-0096383.

constraints that existing approach [8] failed to consider. The new approach consists of an efficient gradient-based non-linear programming technique [5] to minimize the decap area. As the most time-consuming step in gradient-based approach is the gradient computation via transient simulation of P/G networks, an equivalent circuit modeling scheme for RLC chain circuit is proposed to reduce the complexities of P/G networks for speeding up transient simulations. Experimental results show that our method uses smaller decap areas compared with a decap allocation scheme aiming at reducing voltage noise only and can optimize substantial large P/G networks than similar existing approaches.

This paper is organized as follows. Section 2 formulates the problem of the decap area minimization. The efficient non-linear programming technique and the equivalent circuit modeling method are described in Section 3. The time complexity of the algorithm is analyzed in section 4. Experimental results are presented in Section 5. Section 6 concludes the paper.

II Problem Formulation

For the similarity of power and ground networks, we will only describe the algorithm for power networks in this paper. Let $G = \{N,B,M\}$ be a power network with n nodes $N = \{1, ..., n\}$ and b RLC branches $B = \{1, ..., b\}$ and m nodes that are permitted to connect tunable decaps $M = \{1, ...,m\}$. Each branch (p, q) in B connects two nodes p and q with current flowing from p to q. For a node p, $v_p(t)$ is the node voltage.

A. Objective function:

The objective shown below is to minimize the decap area of a power network subject to design rules and power grid integrity related constraints:

$$\min A = \sum_{j\in M}\left(w_j \times H\right), \tag{1}$$

where H is the fixed height of standard cells and w_j is the width of the decap connected to node j.

B. The Constraints

1) The voltage drop constraints. To ensure the correct and reliable logic operation, the voltage drops from the power pads to the leaf nodes should be restricted. To efficiently measure the dynamic voltage drop, we follow the voltage drop noise metric definition in [8] and reformulate the voltage drop constraints in the following for power networks:

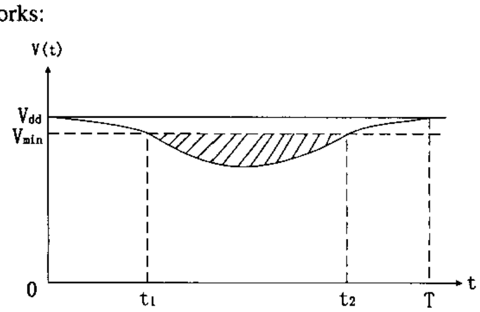

Fig. 2. Illustration of dynamic voltage drops.

$$s_i = \int_0^T \max(V_{\min} - v_i(t), 0)dt = \int_{t_1}^{t_2} (V_{\min} - v_i(t))dt \tag{2}$$

Where $[t_1,t_2]$ is the interval in which the constraint is violated as shown in Fig.2. There may exist several such intervals in one clock cycle.

2) The electro-migration constraints. Electro-migration effects on a power grid wire segment set an upper bound on the current density of the wire segment as the routing layer has fixed thickness. This constraint for branch (p,q) is expressed as $|i_{p,q}(t)| \leq w_{p,q} \times \sigma$, and can be re-written as $|v_p(t)-v_q(t)| \leq \rho l_{p,q}\sigma$, where σ is the maximal current density, ρ is the sheet resistance, and l_{pq} is the length of branch (p, q). Due to dynamic nature of $v_{p,q}(t)$, we reformulate the constraint as below:

$$t_{p,q} = \int_0^T \max\left[\left(\left|v_p(t) - v_q(t)\right| - \rho l_{p,q}\sigma\right), 0\right]dt = \int_{t_1}^{t_2} \left(\left|v_p(t) - v_q(t)\right| - \rho l_{p,q}\sigma\right)dt \tag{3}$$

where $[t_1,t_2]$ are the time interval in which the constraint is violated. Similarly, there may exist several such intervals in one clock cycle.

3) Decap Area Constraints. Because we essentially re-distribute the spare space around cells, the total width of all decaps in a row should be limited by the width of total spare area in a row:

$$dw_{ir} = \max\left(\sum_{jc\in ND(ir)} w_{ir,jc} - rw_{ir}, 0\right), ir \in NR, \tag{4}$$

where NR is the row set defined as $\{1,2,\ldots,N_{row}\}$ and N_{row} is the row number of the cells in the placement; $ND(ir)$ is the decap position set of ir^{th} row defined as $\{1,2,\ldots,nr_{ir}\}$ and nr_{ir} is the number of nodes where decaps can be attached in the ir^{th} row; $w_{ir,jc}$ is the width of the decap at row ir and position jc; and rw_{ir} is the width of total spare area in row ir.

4) Decap Maximum Width Constraints. The width of a decap in row ir should be bounded by the total width of spare area provided by the row.

$$ew_{ir,jc} = \max(w_{ir,jc} - rw_{ir}, 0), ir \in NR, jc \in ND(ir), \tag{5}$$

III Solution Based on Non-Linear Programming

The resulting problem is a nonlinear minimization problem as voltages and currents are nonlinear functions of decap geometries. As a result, we propose to use gradient-based conjugate gradient method to solve the optimization problem as gradients can be efficiently computed in time-domain.

At the beginning, all the decap widths are set to be zero. We then analyze the network to get the node voltage waveforms and the branch currents, and identify the constraint violations. In each optimization iteration, we use conjugate gradient method to update decap widths. This process stops when all the constraints are satisfied.

A. Formulation of Penalty Function

The first step is to transform the original constrained problem into a sequence of unconstrained problems. The transformation is accomplished by adding to the objective function a penalty term that gives a high cost for the constraint violations. We adopt a penalty function as below:

$$f = A + \alpha \cdot \left(\sum_{i \in N} s_i^2 + \sum_{(p,q) \in B} t_{pq}^2 + \sum_{ir \in NR} dw_{ir}^2 + \sum_{jc \in ND(ir)} ew_{ir,jc}^2 \right), \tag{6}$$

where A is the decap area defined by function (1), α is the penalty parameter, s_i is the voltage drop constraint violations defined by equation (2), $t_{p,q}$ is the current density constraint violations defined by equation (3), dw_{ir} and $ew_{ir,jc}$ is the decap width constraint violations defined by equation (4) and (5).

Let us set the penalty term p_t as

$$p_t = \alpha \cdot \left(\sum_{i \in N} s_i^2 + \sum_{(p,q) \in B} t_{pq}^2 + \sum_{ir \in NR} dw_{ir}^2 + \sum_{jc \in ND(ir)} ew_{ir,jc}^2 \right), \tag{7}$$

We then rewrite the penalty function as $f = A + p_t$. Meanwhile, we transform the original constrained problem to the problem of minimizing the penalty function (8),

$$\begin{aligned} &\min f = A + p_t \\ &= A + \alpha \cdot \left(\sum_{i \in N} s_i^2 + \sum_{(p,q) \in B} t_{pq}^2 + \sum_{ir \in NR} dw_{ir}^2 + \sum_{jc \in ND(ir)} ew_{ir,jc}^2 \right), \end{aligned} \tag{8}$$

B. Optimization Scheme

Given an initial penalty parameter α, we minimize the penalty function. Then we increase the value of the penalty parameter α for the next minimization iteration. The process continues until all the constraints are satisfied. The solution procedure can be described as below,

1. Set an initial value of penalty parameter α; initial decap width vector $W^{(0)}$; and error bound $\varepsilon_b > 0$.
2. Solve unconstrained minimization problem (8), obtain current width vector $W^{(k)}$.
3. If $p_t^{(k)} < \varepsilon_b$, then stop, else update penalty parameter α, set $k = k+1$ and turn to step 2.

In our approach, we use the *Flether-Reeves* (FR) conjugate gradient method to solve the unconstrained minimization problems.

C. Time-Domain Merged Adjoint Network Method

In order to efficiently compute the gradients used in conjugate gradient method, we directly compute the required gradient vector expressed in equation (9) instead of individual gradients of decaps. The resulting method is a time-domain merged adjoint network method. We notice that a similar optimization scheme was used in [5] in P/G network wire sizing based on DC analysis. The gradient of penalty function f with respect to the decap width vector W can be expressed as following:

$$\nabla f(W) = \left[\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}, \cdots, \frac{\partial f}{\partial w_j}, \cdots, \frac{\partial f}{\partial w_m} \right]^T, j \in M \tag{9}$$

With equations (1)(6), we can derive the following equation:

$$\begin{aligned} \frac{\partial f}{\partial w_j} &= \frac{\partial A}{\partial w_j} + \frac{\partial}{\partial w_j}\left[\alpha \cdot \left(\sum_{i \in N} s_i^2 + \sum_{(p,q) \in B} t_{pq}^2 + \sum_{ir \in NR} dw_{ir}^2 + \sum_{jc \in ND(ir)} ew_{ir,jc}^2 \right) \right] \\ &= H + 2\alpha \left(\sum_{i \in N} s_i \frac{\partial s_i}{\partial w_j} + \sum_{(p,q) \in B} t_{pq} \frac{\partial t_{pq}}{\partial w_j} + \sum_{ir \in NR} dw_{ir} \frac{\partial dw_{ir}}{\partial w_j} + \sum_{jc \in ND(ir)} ew_{ir,jc} \frac{\partial ew_{ir,jc}}{\partial w_j} \right) \end{aligned} \tag{10}$$

Assume that the decap c_j is located at rj^{th} row and pj^{th} position. Then, equation (10) can be transformed into

$$\begin{aligned} &\frac{\partial f}{\partial w_j} = H + 2\alpha \left(dw_{rj} + ew_{rj,pj} \right) + 2\alpha \left(\sum_{i \in N} s_i \frac{\partial s_i}{\partial w_j} + \sum_{(p,q) \in B} t_{pq} \frac{\partial t_{pq}}{\partial w_j} \right), \\ &rj \in NR, pj \in ND(rj) \end{aligned} \tag{11}$$

In the sequel, we focus on the derivative computations for the constraints s_i and t_{pq}. The relation between decap and its width is

$$c = \frac{\varepsilon_{ox}}{T_{ox}} \times w \times H$$

where ε_{ox} and T_{ox} are the permittivity and thickness of the gate oxide respectively. Thus, equation (11) can be transformed into

$$\begin{aligned} &\frac{\partial f}{\partial w_j} = H + 2\alpha \left(dw_{rj} + ew_{rj,pj} \right) + \frac{2\alpha \varepsilon_{ox} H}{T_{ox}} \left(\sum_{i \in N} s_i \frac{\partial s_i}{\partial c_j} + \sum_{(p,q) \in B} t_{pq} \frac{\partial t_{pq}}{\partial c_j} \right) \\ &rj \in NR, pj \in ND(rj) \end{aligned} \tag{12}$$

Assume decap c_j is connected to the nj^{th} node on the power network. According to the convolution method for sensitivity calculation presented in[17], we can obtain the following equations:

$$\frac{\partial s_i}{\partial c_j} = \int_0^T v'_{nj,B_i}(t) \times \dot{v}_{nj}(T-t)dt \tag{13}$$

$$\begin{aligned} \frac{\partial t_{pq}}{\partial c_j} &= \int_0^T v'_{nj,I_{pq}}(t) \times \dot{v}_{nj}(T-t)dt \\ &= \int_0^T \left[v'_{nj,B_p}(t) - v'_{nj,B_q}(t) \right] \times \dot{v}_{nj}(T-t)dt \end{aligned} \tag{14}$$

where $\dot{v}_{nj}(T-t) = \frac{dv_{nj}(T-t)}{dt}$ is the voltage differential with respect to time t at the nj^{th} node to which the decap c_j is connected in the original circuit, and $v'_{nj,B_i}(t)$, the voltage in the adjoint circuit, can be defined as following

$$\begin{aligned} &v'_{nj,B_i}(t) = Z(nj)v'_{B_i} = Z(nj)G^{-1}\left[B(i) + EC(i) + EL(i)\right] \\ &B(i) = [0,0,\cdots,0,-1,0,\cdots,0]^T, Z(nj) = [0,0,\cdots,0,1,0,\cdots,0] \end{aligned} \tag{15}$$

where G is the coefficient matrix of the nodal voltage equation $G*V=I$, $B(i)$ is a current vector in which all currents are zero except for the current injected into node i which is −1, and $Z(nj)$ is a position choice vector in which all items are zero except for position at the nj^{th} node, which is 1. $EC(i)$ and $EL(i)$ are equivalent current vectors which are functions of the states of capacitor and inductance.

Based on the fact that all adjoint networks have the same coefficient matrix as the original network, the merged adjoint network strategy is used for calculating all node sensitivities for required gradient components in equation (9).

Specifically, by using superposition, we can obtain the merged stimulus vector $I_{new}(t)$ due to constraint violations as:

$$I_{new}(t)=\sum_{i\in N}\left[s_i(t)B(i)\right]+\sum_{(p,q)\in B}\left\{t_{pq}(t)\times\left[B(p)-B(q)\right]\right\}, \quad (16)$$

With this merged stimulus, we can compute the responding voltage vector $X(t)$ as following:

$$X(t)=G^{-1}\left[I_{new}(t)+EC(t)+EL(t)\right]=G^{-1}I_{total}(t) \quad (17)$$

where $I_{total}(t)=I_{new}(t)+EC(t)+EL(t)$. As a result, we obtain node voltages of the merged adjoint network.

$$v'_{c_j}=Z(nj)X(t) \quad (18)$$

With equation (12) and (18), we can simply compute the gradient of penalty function for the descent direction as follows:

$$\begin{aligned}\frac{\partial f'}{\partial w_j}&=H+2\alpha\left(dw_{rj}+ew_{rj,pj}\right)+\frac{2\alpha\varepsilon_{ox}H}{T_{ox}}\int_0^T\left\{\left[Z(nj)\times X(t)\right]\times\dot{v}_{nj}(T-t)\right\}dt\\&=H+2\alpha\left(dw_{rj}+ew_{rj,pj}\right)+\frac{2\alpha\varepsilon_{ox}H}{T_{ox}}\int_0^T\left\{\left[Z(nj)\times G^{-1}\times I_{total}(t)\right]\times\dot{v}_{nj}(T-t)\right\}dt\end{aligned} \quad (19)$$

$rj\in NR, pj\in ND(rj), nj\in N$

In other words, we only need to simulate the power network twice. The first time period is from 0 to T with original time-varying current sources. The second time period is from T to 0 with Inew(t) as current sources. The voltage waveforms at node i got from the two simulations are convolved according to (19) to obtain the gradient with respect to wi. Notice that we need to cache all the voltage waveforms at the violated nodes.

D. Transient Simulation Based on Equivalent Circuit Modeling

Although many efficient simulation methods have been proposed in the past [1]-[3][8]-[16], the regular structures of RLC P/G networks on standard-cell layouts are not explicitly exploited. It was shown that there exist many regular structures in the P/G networks of standard-cell layouts as shown in Fig.3. For such a RLC chain circuit, we can reduce it into a simple resistor-only equivalent circuit and thus speed up the transient simulation of the P/G networks.

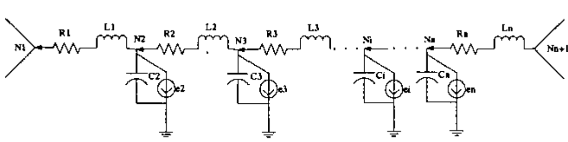

Fig. 3. A series RLC chain in a P/G network.

In our method, we combine equivalent circuit modeling with preconditioned conjugate gradient method (PCG) [5][9] to compute gradients. Specifically, at each time step, the numerical integration by using companion models in Norton's forms is performed and the original RLC circuit will become a resistor-only circuit. We then repeatedly apply the Y-Δ network transformation to reduce one node at a time until only the terminal nodes are left. After this, PCG algorithm solves the reduced network. Once the voltages at terminal nodes are solved, all the voltages at intermediate nodes of original circuits can be back solved using the superposition principle.

E. Decap Legalization

In typically standard cell designs, decaps are assigned as dummy decap cells. Since the solutions we obtain are continuous in terms of decap widths, we need to *legalize* the decap widths in the solutions when actually decaps are assigned. In our approach, we use some bigger decaps for absorbing smaller ones around them, and then round off the areas of the clustered decaps to form a number of dummy decaps.

IV. Analysis of Time Complexity

In this section, we briefly discuss the time complexity of the overall optimization algorithm. As mentioned above, in each area optimization iteration, we need to solve original network and adjoint network once, then get new searching direction and do line searching along this direction. The overall time complexity can be expressed as

$$T=\sum_{i=1}^{N_{iter}}\left(T_{so}(i)+T_{sa}(i)+\sum_{j=1}^{N_{line}(i)}T_{so}(i,j)\right) \quad (20)$$

where, N_{iter} is the total number of area optimization iterations; $N_{line}(i)$ is the number of line searching in iteration i; $T_{so}(i)$ and $T_{sa}(i)$ are the time to solve original network and adjoint network with PCG algorithm in iteration i; $T_{so}(i, j)$ is also the time to solve original network with PCG during line searching.

The components in (20) which consume most of the running time are the time for solving the network and its adjoint network. In our method, as we apply equivalent circuit modeling method to reduce P/G network complexities first, computation costs have been greatly reduced. Specifically, if there are total n nodes in the P/G network, n_{cross} is the number of nodes left after reduction and n_{mid} is the number of nodes suppressed. $n= n_{cross} + n_{mid}$. Typically n_{cross} is far less than n in the P/G networks of standard-cell layouts. As a result, the reduced P/G network is significantly smaller than the original network.

In Fig. 4, we define $\beta= n/n_{cross}$, which reflects the reduction ratio between the original P/G network and the reduced P/G network, and draw curve of CPU time v.s numbers of nodes in circuits. Four networks with different β are solved as examples. It is shown that the time complexity of P/G network simulation is reduced remarkably as β increases and the CPU time used by the new algorithm grows almost linearly when $\beta\geq 10$.

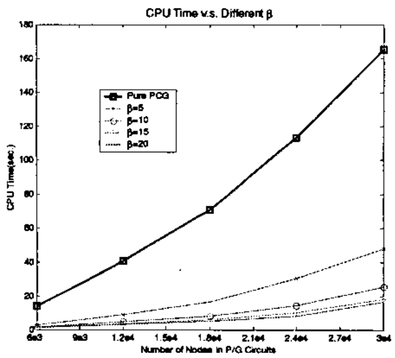

Fig. 4. Simulation time for different P/G structures.

V. Experimental Results

The proposed decap optimization algorithm has been implemented in *C* and *C++*. All the experimental results are obtained on *SUN UltraSparc* workstation V880 with 750MHz CPU and 2GB memory.

We tested our program with some real industry standard-cell circuits with pre-placement information in LEF/DEF format. Those circuits have complexities ranging from 744 nodes to 1.6 million nodes. Table I shows the parameters of each circuit. The violation nodes are nodes on the power grid that violate the voltage drop constraints or the electro-migration constraints.

Ideally, we would like to compare our work with some similar existing works such as [8]. But due to the proprietary nature of those algorithms and their test circuits, direct comparison is impossible. Instead, the proposed algorithm is compared with a simple sensitivity based decap allocation heuristic method that we developed, which is similar to sensitivity based method used in [2]. In the simple heuristic algorithm, we first calculate the sensitivity of all the decap candidate nodes and sort them in a descending order. Then we choose a number of the nodes with largest sensitivities to add a small number of decaps around them and also adjust the cell positions in row. If the violations still exist, the process is repeated until no violations are present. TABLE II summarizes the comparison results of two algorithms.

TABLE I. Parameters of Circuits

Name of circuits	Num of nodes	Num of violation nodes
U_cnt100	744	91
U_cnt500	3741	665
U05614	32112	3683

TABLE II. Optimization Efficiency Comparison

Name of circuits	Heuristic algorithm			Programming algorithm		
	Time (s)	Area of Decap (μm^2)	Num of decap	Time (s)	Area of Decap (μm^2)	Num of decap
U_cnt100	38.02	7951.429	128	109.62	5677.962	196
U_cnt500	223.59	40634.776	549	339.41	35007.882	797
U05614	954.09	245153.92	7736	1285.15	174591.12	9453

Table II clearly shows that area obtained by the algorithm proposed in this paper is significantly smaller than that obtained from the simple heuristic method although the new method takes a little bit more CPU time.

Compared with a similar standard-cell based decap allocation algorithm [8], our algorithm is able to handle much larger circuits. For instance, the largest circuit that we can optimize within 7.38 hours has 1,618,026 nodes, while the method in [8] can only size circuit with only 828 nodes. We test our algorithm with some large circuits and the results are listed in TABLE III. For all the test cases, the new algorithm successfully eliminates all the violation nodes.

TABLE III. Results of Large Circuits

Name of circuits	Num of nodes	Num of violation nodes	Running time(s)	Num of decaps	Area of decaps (mm^2)	Chip area (mm^2)
U19649	112392	10755	2455.03	10652	2.6465	9.4224
U56140	321120	11496	9216.20	14996	3.2290	36.7236
Nec_600k	1618026	61213	26579.31	5845	0.5223	183.3316

Fig.5 shows the distribution profile of violation nodes (nodes with violations) for the circuit c_cnt100 before optimization where violation nodes are marked by black color. Since the pads are located on the up-left and bottom-right corners of the chip, the violation nodes are mainly at the center of the circuit as expected.

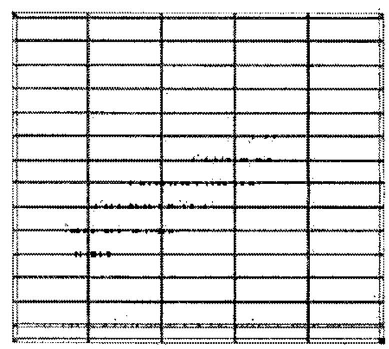

Fig. 5. Distribution of violation nodes.

Fig. 6 and Fig. 7 show the decap allocation results from the simple heuristic algorithm and our programming based algorithm for circuit c_cnt100 respectively. The dark (red) blocks are added decaps and the light (white) ones are spare space. We observe that our new algorithm not only saves more spare area but also spreads decaps closer to the violation nodes while the simple heuristic approach places decaps in more narrow areas which are far from some violation nodes. As a result, our algorithm can save more spare area for later buffering without forbidden placing zone.

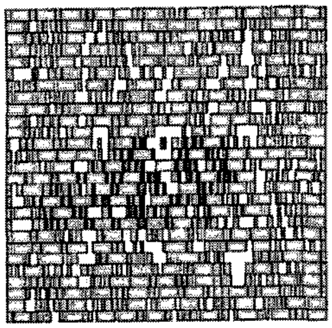

Fig. 6. Decap allocation result of the simple heuristic.

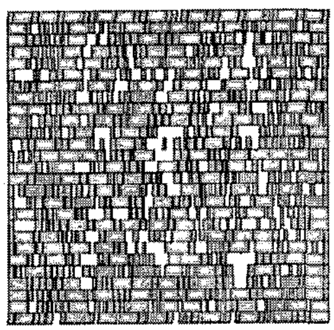

Fig. 7. Decap allocation result of the new algorithm.

VI. Conclusion

In this paper, we have proposed an efficient non-linear programming based decoupling capacitor budgeting algorithm. Our method utilizes conjugate gradient algorithm to search for the best solution. The proposed time-domain merged adjoint network method combined with a novel equivalent circuit modeling technique leads to very efficient computation of gradients, which is key to the overall efficiency of the optimization algorithm. Experimental results have demonstrated that the proposed algorithm is capable of optimizing P/G networks with million nodes in 7.38 hours. Since the new algorithm explicitly minimizes the area of decaps, it uses significantly smaller decap area compared with a simple heuristic method where the main objective is to just reduce P/G noise. As our new algorithm scales well for large P/G networks and results in less budgeted decap area, it can easily be incorporated into other physical design steps to achieve fast design closure.

References

[1] H. H. Chen, D. D. Ling. "Power supply noise analysis methodology for deep-submicron VLSI chip design." *Proc. 34th ACM/IEEE Design Automation Conf.*, pp. 638-643,1997.

[2] G. Bai, S. Bobba and I. N. Hajj, "Simulation and optimization of the power distribution network in VLSI circuits", *Proc. IEEE/ACM International Conf. on Computer-Aided Design.*, pp. 481–486, 2000.

[3] H.-H. Su, K. H. Gala, and S. S. Sapatnekar. "Fast analysis and optimization of power/ground networks." *Proc. IEEE/ACM International Conf. on Computer-Aided Design.*, pp. 477–482, 2000.

[4] X.-D. Tan and C.-J. Shi. "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programming." *Proc. 36th ACM/IEEE Design Automation Conf.*, pp. 78-83,1999.

[5] X.-H. Wu, X.-L. Hong, Y.-C. Cai, and et al. "Area minimization of power distribution network using efficient nonlinear programming techniques." *Proc. IEEE/ACM International Conf. on Computer-Aided Design.*, pp. 153–157, 2001.

[6] X.-D. Tan and C.-J. Shi. "Fast power-ground network optimization using equivalent circuit modeling," *Proc. 38th ACM/IEEE Design Automation Conf.*, pp. 550–554, 2001.

[7] S.-Y. Zhao, K.K. Roy, C.-K. Koh. "Decoupling capacitance allocation for power supply noise suppression." *Proc. IEEE/ACM International Symp. on Physical Design*, pp.66-71,2001.

[8] H.-H. Su, K.K. Roy, S. S. Sapatnekar, S. R. Nassif. "An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts. " *Proc. IEEE/ACM International Symp. on Physical Design*, pp.68-75,2002.

[9] T. Chen and C. C. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative method", *Proc. ACM/IEEE Design Automation Conf.*, pp. 559–562, June, 2001.

[10] A. Odabasioglu, M. Celik, and L. T. Pilleggi, "PRIMA: passive reduction-order interconnect macromodeling algorithm," *IEEE Trans. On Computer-Aided Design*, vol. 17, no. 8, pp.645-654, 1998.

[11] Y. Cao, Y. Lee, T. Chen and C. C. Chen, "HiPRIME: hierarchical and passivity reserved interconnect macromodeling engine for RLKC power delivery", *Proc. ACM/IEEE Design Automation Conf.*, pp. 379–384, June, 2002.

[12] D. Stark and M. Horowitz, "Techniques for calculating currents and voltages in VLSI power supply networks," *IEEE Trans. on Computer-Aided Design*, vol. 9, no. 2, pp. 126-132, February 1990.

[13] A. Dharchoudhury, R. Panda, D. Blaauw, R. Vaidyanathan, B. Tutuianu and D. Bearden, "Design and analysis of power distribution networks in power PC microprocessors", *Proc. ACM/IEEE Design Automation Conf.*, pp. 738–743, June, 1998.

[14] Y.-M. Lee, C.-P. Chen. "Power grid transient simulation in linear time based on transmission-line-modeling alternating-direction-implicit method." *Proc. IEEE/ACM International Conf. on Computer-Aided Design.*, pp. 75–80, 2001.

[15] J. N. Kozhaya, S. R. Nassif, and F.N. Najm. "A multigrid-like technique for power grid analysis. *IEEE Trans. Computer-Aided Design*, vol. 21, no.10, pp. 1148-1160, Oct. 2002.

[16] M. Zhao, R. V. Panda, S. S. Sapatnekar and D. Blaauw, "Hierarchical analysis of power distribution networks", *IEEE Trans. Computer-Aided Design*, vol. 9, no. 2, pp. 159–168, Apr. 1990.

[17] L.O. Chua and P.M. Lin, *Computer-Aided Analysis of Electronic Circuits*, Englewood Cliffs: Prentice-Hall, Inc. 1975.

[18] K. Wang and M. Marek-Sadowask, "On-chip power supply network optimization using multigrid-based technique", *Proc. ACM/IEEE Design Automation Conf.*, pp. 113–118, June, 2003.