Decoupling Capacitance Efficient Placement For Reducing Transient Power Supply Noise*

Xiaoyi Wang
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology
Tsinghua University
Beijing, China

Yici Cai
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology
TsingHua University
Beijing, China

Qiang Zhou
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology
Tsinghua University
Beijing, China

Sheldon X.-D. Tan
Department of Electrical Engineering
University of California, Riverside, CA, USA

Thom Eguia
Department of Electrical Engineering
University of California, Riverside, CA, USA

ABSTRACT

Decoupling capacitance (decap) is an efficient way to reduce transient noise in on-chip power supply networks. However, excessive decap may cause more leakage power, chip resource waste, and even lead to more design iterations. In this paper, we present a novel decap-efficient placement algorithm for transient power supply noise reduction. In contrast to traditional design flow, our approach considers decap impacts at the placement stage to seek the placement minimizing decap requirements while still satisfying the traditional placement objectives. In the new method, we first devise a fast procedure to assess the decap requirement for the force-based placement framework, in which the required decap is modeled as a density function over the chip. Then, we build a corresponding *supply and demand* system to adjust the placement in favor of minimizing decap. Finally, we develop a decap efficient placement algorithm with a new force induced by imbalance between power supply and power demands. Experimental results show that the new combined placement and decap optimization flow could reduce the minimum decap area by 35% with a wire length increase of only 0.5% at nearly the same computational cost, which is efficient for practical problems.

*This work is sponsored in part by NSFC grants under No. 60828008 and No. 60776026, in part by National Science and Technology Major Projects grant under No. 2008ZX01035-001-05 and in part by NSF grant under No. OISE-0929699.

ICCAD'09, November 2–5, 2009, San Jose, California, USA.

1. INTRODUCTION

As CMOS technology advances to the 45nm regime and below, the on-chip power supply integrity problem becomes a major design concern. With an increase in device density and a decrease in supply voltage, power/ground (P/G) networks are experiencing larger supply currents as well as smaller noise margins, making P/G network design more challenging.

Power supply degradation comes primarily from two types of voltage drops (noise). One is a static IR voltage drop owing to wire resistance in the power supply network. The other is a transient voltage drop caused by dynamical switching currents and the inductive effect of wires with large currents. Inserting decoupling capacitances is the most efficient way to reduce severe transient power noise. As more devices are packed into a chip, a large amount of decap is needed to maintain power delivery integrity. However, excessive decap could lead to other problems: (1) Decap is leaky, and the leakage of a decap cell will increase the power consumption substantially; (2) Decap could occupy the chip area, and thus decrease the yield of a chip; (3) Massive decap will compete with other components, such as clock buffers and routing. This resource competition may result in more design iterations. Therefore, it is important to minimize the decap as long as the dynamic supply noise could be restricted.

In order to avoid excessive decap insertion, many research works have been carried out to develop optimal decap insertion algorithms. In works [12] and [3], a mathematical programming based algorithm was proposed to minimize the decap insertion. However, these methods are not efficient enough for very large circuits. Therefore, many research efforts tried to improve the efficiency and quality of the decap optimization algorithms in the past. For instance, [8] proposed a partition-based decap minimization algorithm, which performs the decap optimization on a local region of a power grid. Meanwhile, [15] presented a macro-model based algorithm, which simplifies the decap problem by us-

ing a charge-based model, and then solving the resulting linear programming problem. The charge based method has been further improved by [13] where a better models for estimating the required charged was proposed. Furthermore, more fine-tuned algorithms [2], [4] were proposed to speedup the decap optimization. However, practical power grids are massive in size and require long analysis times, making optimization methods very time-consuming.

Moreover, these methods fail to consider the impact of physical layout to power supply networks. As we know, the underlying blocks or cells provide the current load to the power supply networks, with different layouts having unique impacts on power supply performance. Some research works have been proposed in this direction. For example, [14] considers the decap placement problem in the floorplan stage and minimizes the total area of the floorplan and decap. Meanwhile, [5] integrates the IR drops into the target function of placement to produce a placement with better static voltage drops. However, none of the existing methods consider the transient power supply noise and corresponding decap reduction at the placement stage. The reasons are that there are no obvious relationships between the required decap budget and placement results. The tradeoff between the required decap budget and other placement criteria, such as wire length, have not been thoroughly investigated. Further, the number of modules in placement is very large and the resulting power supply system becomes too complex to be analyzed efficiently as inner loops by traditional algorithms.

In this paper, we present a decap efficient placement algorithm to reduce the transient noise of power supply grids with minimal decap insertion. First, we transform the decap budgeting problem to a linear programming problem by the charge-based modeling approach [15]. Second, we model the decap requirement as an *electric density* function over the chip during placement. Essentially, the *electric density* represents the power supply demands. Regions with high electric densities require decap to enhance the power supply. Therefore, balancing the electric density will reduce the decap budget. Third, we establish a *supply and demand* system for the electric density and generate the move forces accordingly, which will spread modules in favor of reducing decap. Together with the other forces in the original force-directed placer (FDP) [11], the decap-induced forces will produce a placement with reduced decap budget as well as optimized wire length. The decap-efficient placement method explores the non-uniform power delivery capability of power grid networks to reduce the gaps between power supply and demands. Experimental results show that the new method can reduce the minimum decap area by 35% with a wire length increase of only 0.5% at nearly the same computational cost, which is efficient for practical problems.

2. PROBLEM FORMULATION

2.1 Placement Problem

The placement problem is formulated as an optimization problem with the quadratic wire length model :

$$\min_{(\boldsymbol{x},\boldsymbol{y})} \quad \Gamma = \Gamma_x + \Gamma_y \tag{1}$$

$$\text{subject to} \quad \Omega \leqslant \alpha \tag{2}$$

where Γ is the half perimeter wire length (HPWL), $(\boldsymbol{x}, \boldsymbol{y})$ is the position vector of modules, Γ_x and Γ_y are the HPWL in the x and y directions, respectively, Ω is the overlap ratio and α is the upper bound of the overlap ratio, usually at 10%–20%.

2.2 Decap Optimization Problem

The decap optimization problem involves the use minimum decap to avoid excessive voltage drop :

$$\min_{c_i} \quad \sum_i c_i \tag{3}$$

$$\text{subject to} \quad C \cdot \tfrac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} + G \cdot \boldsymbol{V} = \boldsymbol{J}$$

$$\boldsymbol{V} \geqslant V_{\text{thre}} \tag{4}$$

$$c_{int,i} \leqslant c_i \leqslant c_{max,i}$$

where c_i is the capacitance, which includes both intrinsic and decoupling capacitances where $c_{int,i}$ is the intrinsic capacitance at node i. $C = diag(c_1, \ldots, c_n)$, $\boldsymbol{V}$ is the transient voltage waveform vector, V_{thre} is the threshold voltage, $c_{max,i}$ is the maximum capacitance allowed at node i, G is the conductance matrix obtained by modified nodal analysis (MNA) and $\boldsymbol{J}$ is the current source vector. In our formulation, we ignore the inductances as they are still less important especially at the chip level [10].

2.3 Co-synthesis Problem

The decap efficient placement problem (a.k.a co-synthesis problem) seeks to find a placement in favor of reducing both wire length and decap amount:

$$\min_{(\boldsymbol{x},\boldsymbol{y})} \quad \Gamma + \lambda \cdot \textstyle\sum_i c_i \tag{5}$$

$$\text{subject to} \quad \Omega \leqslant \alpha$$

$$C \cdot \tfrac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} + G \cdot \boldsymbol{V} = \boldsymbol{J}$$

$$\boldsymbol{V} \geqslant V_{\text{thre}} \tag{6}$$

$$c_{int,i} \leqslant c_i \leqslant c_{\max,i}$$

where λ ($\lambda \geqslant 0$) is the weight parameter, which indicates how the decap amount should be traded-off with wire length.

2.4 Challenges of Co-synthesis Problem

The co-synthesis problem (5), however, is far more difficult than either the plain placement problem (1) or the decap optimization problem (3). The reason is that the decap variables c_i in the target function of the problem (5) are no longer independent variables but functions of module positions $(\boldsymbol{x}, \boldsymbol{y})$. Moreover, the decap optimization problem (3) is a sub-problem of the co-synthesis problem because the c_i in the co-synthesis problem (5) must also be the solution of the decap optimization problem (3) for the given placement. Otherwise the target function of problem (5) could be further decreased by solving the decap optimization problem (3). Since both the placement and decap optimization problems are well known difficult problems, we confront enormous challenges after putting them together as the co-synthesis problem (5):

1. As a sub-problem, decap optimization has to be solved in every iteration of the placement process, which is infeasible due to huge time consumption.

2. The decap amount in the target function of problem (5) is a strong non-linear function of module positions $(\boldsymbol{x}, \boldsymbol{y})$ which makes the co-synthesis problem very difficult (if not impossible) to solve.

3. The combined voltage drop constraints and overlap ratio constraints in (6) may make the placement process difficult to converge.

2.5 Proposed Decap-Efficient Placement and Decap Optimization Flow

In this paper, we instead seek to make the placement more decap aware and efficient before minimum decap insertions are performed. We show that in this way, decap can be substantially reduced with very small overhead compared to the existing placement and decap optimization flow. The basic idea behind the decap-efficient placement is that we can better match the power consumption from cells/modules with the power delivery capability of the power grid. Because this capability is not uniform over a chip (e.g. regions close to the power pads supply more power), we manage to place larger power consuming modules in regions with larger power delivery capability.

3. REVIEW OF THE ANALYTICAL PLACEMENT

There are several well-known placers proposed for the placement problem, such as [1, 6, 7, 11]. Among these placers, *Kraftwerk2* proposed in [11] is a force-directed quadratic placer using a novel quadratic wire-length model (bound2bound net model), which applies a Poisson's equation to remove the module overlap. The appeal of the Kraftwerk2 placer is that it uses analytical functions to model the wire-length and the overlap, which makes it easy to integrate the decap target function in the placement process. Therefore, we choose this algorithm as our baseline placement method.

Kraftwerk2 first models the net wire-length as quadratic functions using the bound2bound (B2B) net model as follows:

$$\begin{aligned} \Gamma_x &= \frac{1}{2}\boldsymbol{x}^T \cdot C_x \cdot \boldsymbol{x} + d_x^T \cdot \boldsymbol{x} + \text{const} \qquad (7) \\ \Gamma_y &= \frac{1}{2}\boldsymbol{y}^T \cdot C_y \cdot \boldsymbol{y} + d_y^T \cdot \boldsymbol{y} + \text{const} \end{aligned}$$

where the semi-positive definite symmetric matrices C_x, C_y represent the connectivity between movable modules and vectors d_x, d_y represent the connections between movable and fixed modules. (See [11] for more details about B2B net model).

Second, it introduces a *supply and demand* system to model the overlap to spread the modules over the chip. The *supply and demand* system consists of a supply density function $D_{\text{mod}}^{\text{sup}}(x,y)$ and a demand density function $D_{\text{mod}}^{\text{dem}}(x,y)$, where $D_{\text{mod}}^{\text{sup}}(x,y)$ represents the target module density distribution, i.e. how the modules should be distributed after the placement and $D_{\text{mod}}^{\text{dem}}(x,y)$ represents the distribution of modules in the current placement iteration. In order to make $D_{\text{mod}}^{\text{dem}}(x,y)$ approach to $D_{\text{mod}}^{\text{sup}}(x,y)$ as the placement progresses, it defines the potential function $\Phi(x,y)$ by Poisson's equation subject to Neumann's boundary condition:

$$\begin{aligned} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\Phi(x,y) &= D_{\text{mod}}^{\text{sup}}(x,y) - D_{\text{mod}}^{\text{dem}}(x,y) \qquad (8) \\ B.C. \qquad & \frac{\partial \Phi}{\partial n} = 0 \text{ (Neumann's B. C.)} \end{aligned}$$

where n is the unit norm vector to the boundary of placement region.

Then, the spread forces are generated as the gradient of the potential function $\Phi(x,y)$. For example, the spread force for module i at $(x_i^{'}, y_i^{'})$ in the x direction is defined as:

$$\begin{aligned} F_{x,i}^{\text{move}} &= w_i(x_i - \dot{x}_i) \\ &= w_i\left(\Delta x_i + \frac{\partial \Phi}{\partial x}|_{(x_i^{'}, y_i^{'})}\right) \end{aligned}$$

where w_i is the weight for module i, x_i is the position variable of module i, $x_i^{'}$ is the position of module i before the placement iteration, $\dot{x}_i = x_i^{'} - \frac{\partial \Phi}{\partial x}|_{(x_i^{'}, y_i^{'})}$ is the target position of module i under spread force and $\Delta x_i = x_i - x_i^{'}$ is the movement variable of module i.

If we denote the diagonal weight matrix $\dot{C}_x = \text{diag}(w_i)$, $\Delta\boldsymbol{x} = (\Delta x_1, \Delta x_2, \ldots, \Delta x_m)^T$ and the gradient of the x direction $\Phi_x = \frac{\partial \Phi}{\partial x}$, the move force vector is

$$F_x^{\text{move}} = \dot{C}_x(\Delta\boldsymbol{x} + \Phi_x)$$

Notice that the Poisson's equation (8) is subject to Neumann's boundary conditions. So this equation is solvable if and only if

$$\iint D_{\text{mod}}^{\text{dem}}(x,y)\,\text{dx}\,\text{dy} = \iint D_{\text{mod}}^{\text{sup}}(x,y)\,\text{dx}\,\text{dy}$$

i.e. the supply and demand system has to be balanced as mentioned in [11].

Besides the move forces F_x^{move}, there are two more forces in the force-directed placement. First is the net force, which is the derivative of the quadratic wire length function (7):

$$F_x^{\text{net}} = C_x \cdot \boldsymbol{x} + d_x$$

where $\boldsymbol{x} = [x_1, \cdots, x_i, \cdots]$ is the module position variable vector. Second is the hold force, which is supposed to hold the modules where they are if no move force is present :

$$F_x^{\text{hold}} = -(C_x \cdot \boldsymbol{x}^{'} + d_x)$$

where $\boldsymbol{x}^{'} = [x_1^{'}, \cdots, x_i^{'}, \cdots]$ is the current module position vector. The new placement is then determined to balance these three forces

$$\begin{aligned} & F_x^{\text{net}} + F_x^{\text{hold}} + F_x^{\text{move}} = 0 \\ \Longrightarrow \quad & (C_x + \dot{C}_x)\Delta\boldsymbol{x} = -\dot{C}_x\Phi_x \qquad (9) \end{aligned}$$

Forces in the y direction can be determined similarly, so we will describe the forces only in the x direction.

Algorithm 1 describes the Kraftwerk2 placement method. Quality control is generally done to scale the weight matrix $\dot{C}_x$ properly to guarantee convergence (overlap-free) and the quality (wire-length minimized) of placement. For more details, please refer to paper [11].

4. SUPPLY AND DEMAND SYSTEM FOR DECAP BUDGET

In order to move the placement in favor of reducing the decap budget, we have to generate additional forces corresponding to the decap requirements. We first need to estimate the decap requirement in every placement iteration. However, as we mentioned before, the decap budget subproblem (3) is too time consuming to solve in every iteration of placement. Therefore, we have to estimate the decap amount in a more efficient way.

Algorithm 1 Kraftwerk2 Placement Algorithm

Initial the placement
Initialize weight matrix $\dot{C}_x$
while $\Omega > \alpha$ **do**
  Create supply-and-demand $D^{\text{sup}}_{\text{mod}}(x,y)$,$D^{\text{dem}}_{\text{mod}}(x,y)$
  Calculate $\Phi(x,y)$ by (8)
  Create C_x,Φ_x
  Solve equation (9) w.r.t. $\Delta \boldsymbol{x}$
  Update module position $\boldsymbol{x}$ by $\Delta \boldsymbol{x}$
  Quality control for $\dot{C}_x$
end while
Legalize the placement

The basic idea of the decap budget evaluation is to 1) first transform the decap budget sub-problem (3) to a linear programming problem according to [15]; 2) then transform the linear programming decap problem into its dual problem; 3) use the equivalent charge to be supplied by the decap as the merit of the required decap amount, and model this merit as the *electric density* function over the chip; 4) finally, establish the *supply and demand* system for the electric density.

In the paper [15], a sequential linear programming (SLP) based algorithm is proposed for decap budgeting. The essential idea of this algorithm is to transform the voltage constraints $C \cdot \frac{d\boldsymbol{V}}{\text{dt}} + G \cdot \boldsymbol{V} = \boldsymbol{J}$ and $\boldsymbol{V} \geqslant V_{\text{thre}}$ in decap problem (4) into linear constraints by macro-modeling, integration and linear approximation as follows :

1. Suppose the $\boldsymbol{V}_2$ is the voltage vector of the sampling nodes of violation regions ,i.e. the regions where decaps are needed, and $[t_0, t_1]$ is the violation time window, i.e. the time period when the voltage drop is violated (see [15] for the detailed definition of violation regions and violation time window), then partition the equation $C \cdot \frac{d\boldsymbol{V}}{\text{dt}} + G \cdot \boldsymbol{V} = \boldsymbol{J}$ and do the integration as follows :

$$\begin{pmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{pmatrix} \cdot \begin{pmatrix} \int_{t_0}^{t_1} \boldsymbol{V}_1 \,\text{dt} \\ \int_{t_0}^{t_1} \boldsymbol{V}_2 \,\text{dt} \end{pmatrix} = \begin{pmatrix} \int_{t_0}^{t_1} \boldsymbol{J}_1 \,\text{dt} \\ \int_{t_0}^{t_1} \boldsymbol{J}_2 \,\text{dt} + \int_{t_0}^{t_1} \boldsymbol{I} \,\text{dt} \end{pmatrix}$$

   where the new item $\boldsymbol{I}$ is the current that is supposed to be drawn from decap.

2. Reformulate above equation using macro modeling (Schur complement) to obtain the charge transfer equation:

$$\boldsymbol{Q} \quad = \quad A \cdot \bar{\boldsymbol{W}} + \boldsymbol{B} \tag{10}$$

   where $\boldsymbol{Q} = \int_{t_0}^{t_1} \boldsymbol{I} \,\text{dt}$ is the charge supposed to be transferred from decap, $A = G_{22} - G_{12}^T G_{11}^{-1} G_{12}$ is the admittance matrix, $\bar{\boldsymbol{W}}$ is $\int_{t_0}^{t_1} \boldsymbol{V}_2 \,\text{dt}$ and

$$\boldsymbol{B} = G_{12}^T G_{11}^{-1} \int_{t_0}^{t_1} \boldsymbol{J}_1 \,\text{dt} - \int_{t_0}^{t_1} \boldsymbol{J}_2 \,\text{dt}$$

   is the equivalent charge drawn by current sources.

3. Transform constraints $\boldsymbol{V} \geqslant V_{\text{thre}}$ to linear constraints on voltage-time integral $\bar{\boldsymbol{W}}$. As shown in Figure 1, the voltage waveform should be above the V_{thre} after adding decap. Correspondingly, the integral of voltage waveform ($\bar{\boldsymbol{W}}$) should be no less than the shaded area. The linear approximation of the shaded area is used because this area remains unknown until the decap is added. Hence, the transient voltage constraint $\boldsymbol{V} \geqslant V_{\text{thre}}$ is transformed to:

$$\bar{\boldsymbol{W}} \quad \geqslant \tfrac{1}{2}(\boldsymbol{V}(t_0) + V_{\text{thre}})(t_1 - t_0)$$

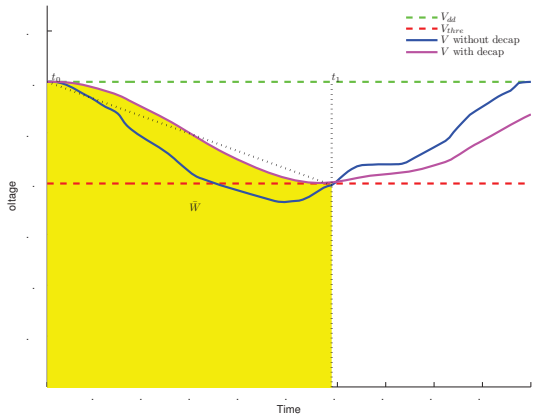

Figure 1: Voltage waveforms before & after adding decap

Since all the constraints in the decap problem (3) have been transformed to linear constraints, we end up with a linear programming problem :

$$\begin{aligned} \min \quad & \textstyle\sum_{i \in \boldsymbol{S}} c_i & (11) \\ \text{subject to} \quad & \boldsymbol{M}' \circ \boldsymbol{C} \geqslant A \cdot \bar{\boldsymbol{W}} + \boldsymbol{B} \\ & \bar{\boldsymbol{W}} \geqslant \boldsymbol{L} & (12) \\ & 0 \leqslant \boldsymbol{C} \leqslant C_{max} \end{aligned}$$

where $\boldsymbol{C} = (c_1, \ldots, c_m)^T$,$C_{max} = (c_{max,1}, \ldots, c_{max,m})^T$,

$$\boldsymbol{M}' = \begin{pmatrix} \boldsymbol{V}_1(t_0) - \boldsymbol{V}_1(t_1) \\ \boldsymbol{V}_2(t_0) - \boldsymbol{V}_2(t_1) \\ \vdots \\ \boldsymbol{V}_m(t_0) - \boldsymbol{V}_m(t_1) \end{pmatrix}, \boldsymbol{L} = \begin{pmatrix} (V_{\text{thre}} + \boldsymbol{V}_1(t_0)) \cdot T \\ (V_{\text{thre}} + \boldsymbol{V}_2(t_0)) \cdot T \\ \vdots \\ (V_{\text{thre}} + \boldsymbol{V}_m(t_0)) \cdot T \end{pmatrix} \tag{13}$$

where $\boldsymbol{S}$ is the set of sampling nodes, $m = |\boldsymbol{S}|$,$T = (t_1 - t_0)/2$ and the $\circ$ operation is the element-wise product.

Despite simplifying the decap problem by linearization, it is still unclear how the decap will be effected when the modules are moved on the chip. The reason is that it is the equivalent current (integration) $\boldsymbol{B}$ on which the module distribution has a direct impact. However, the equivalent current $\boldsymbol{B}$ appearing as the constraints in the problem (11) obscures the relationship between the decap amount and module positions. Therefore, we transform the linear programming problem (11) to its dual problem to make this relationship more explicit :

$$\begin{aligned} \max \quad & (\boldsymbol{L} - \boldsymbol{W})^T \cdot \bar{\boldsymbol{c}} - C_{max}^T \cdot \boldsymbol{q} & (14) \\ \text{subject to} \quad & \boldsymbol{M}' \circ (A^{-1} \cdot \bar{\boldsymbol{c}}) - \boldsymbol{q} \leqslant E \\ & A^{-1} \cdot \bar{\boldsymbol{c}} \geqslant 0 \\ & \bar{\boldsymbol{c}} \geqslant 0, \boldsymbol{q} \geqslant 0 \end{aligned}$$

where $\boldsymbol{W}$ is $-A^{-1} \cdot \boldsymbol{B}$, $\bar{\boldsymbol{c}}$ and $\boldsymbol{q}$ are the dual variables and $E = [1, \cdots, 1]^T$ is an all-one vector. Notice that $\boldsymbol{W} = \bar{\boldsymbol{W}}$ before decap allocation (i.e. $\boldsymbol{Q} = 0$). So $\boldsymbol{W}$ represents the same area as $\bar{\boldsymbol{W}}$ does in Figure 1.

As we know, the dual problem (14) has the same solution as the original decap problem (11), representing the

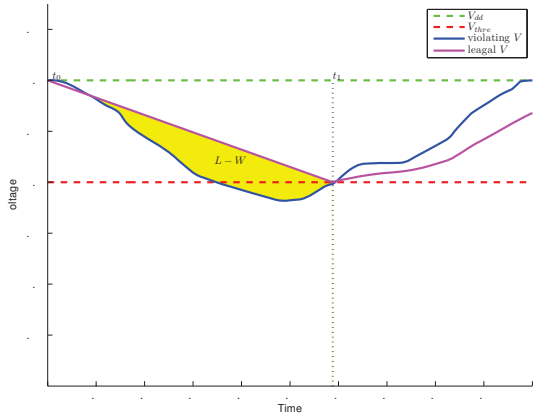

Figure 2: Discrepancy between the violating voltage waveform and legal voltage waveform

decap amount needed. Moreover, we can see that the decap amount is proportional to the 1-norm $\|\boldsymbol{L}-\boldsymbol{W}\|_1$. Actually, the vector $\boldsymbol{L}-\boldsymbol{W}$ represents the discrepancy between the violation voltage waveform and legal (expected) voltage waveform, as illustrated by the shaded area in Figure 2.

The $\boldsymbol{L}$ and $\boldsymbol{W}$ represent the *supply and demand* in the power distribution similarly to the module distribution: $\boldsymbol{W}$ is the voltage integral that the module demands and $\boldsymbol{L}$ is the voltage integral that the power supply network can supply. The discrepancy between the two will be remedied by added decap.

Intuitively, we do not want a significant imbalance between the demand and supply occurring locally in a chip such that large decaps have to be used to resolve the imbalance. One objective of placement is to reduce the imbalance by better matching the demand and supply, as the power supply capability is non-uniform over a chip. As a result, we want the placement to spread the imbalance more evenly throughout the entire chip so that the total decap requirement will be reduced.

To this end, we can interpolate the $\boldsymbol{W}$ and $\boldsymbol{L}$ on the chip area to obtain the *electric density* functions $W(x,y)$ and $L(x,y)$, respectively. Here we can find the similarity between the *electric density* functions and *module density* functions:

$$\begin{aligned} W(x,y) &\longleftrightarrow D^{\text{dem}}_{\text{mod}}(x,y) \\ L(x,y) &\longleftrightarrow D^{\text{sup}}_{\text{mod}}(x,y) \end{aligned}$$

Therefore, we could also establish the similar *supply and demand* system for power supply grids in the same manner as modeling the module distribution :

$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\Psi(x,y) = L(x,y)-W(x,y) \qquad (15)$$

$$B.C. \qquad \frac{\partial \Psi}{\partial n}=0 \ \text{(Neumann'sB. C.)}$$

The corresponding Poisson's equation (15) is similar to the module potential equation (8) except the supply function $L(x,y)$ cannot generally be balanced with demand function $W(x,y)$. This imbalance is intrinsic, which leads to the need to add decap for balancing. Nevertheless, we can still re-orthogonalize the supply and demand function to make the Poisson's equation (15) solvable. i.e. $\tilde{L}(x,y) = L(x,y) - \iint L\,\mathrm{dx}\,\mathrm{dy}$ and $\tilde{W}(x,y) = W(x,y) - \iint W\,\mathrm{dx}\,\mathrm{dy}$.

5. DECAP EFFICIENT PLACEMENT

5.1 Additional Move Forces for Reducing Decap

After the supply and demand system has been established for the decap budgeting, we can now generate the additional forces for the decap according to the potential function $\Psi(x,y)$ similar to the way the move forces F^{move} were generated.

For the module i at $(x_i^{'}, y_i^{'})$, the additional force in the x direction is defined as follows:

$$\begin{aligned} F^{\text{decap}}_{x,i} &= q_i(x_i-\hat{x}_i) \\ &= q_i\left(\Delta x_i + \frac{\partial \Psi}{\partial x}|_{(x_i',y_i')}\right) \end{aligned}$$

where $\hat{x}_i$ is the target point of module i according to decap potential $\Psi(x,y)$, while q_i is the weight of the module i. In order to make the decap decrease as quickly as possible, the module consuming larger current is expected to move faster. We choose the weight q_i according to the charge drawn by module i and then normalize it:

$$q_i = \frac{\int \boldsymbol{J}_i\,\mathrm{dt}}{\sum_k \int \boldsymbol{J}_k\,\mathrm{dt}}$$

In addition to the spread force $F^{\text{move}}_{x,i}$, we now have another move force $F^{\text{decap}}_{x,i}$. When combining the two move forces, we have to trade-off between them carefully to ensure:

1. The magnitude of $F^{\text{decap}}_{x,i}$ should be comparable with the magnitude of $F^{\text{move}}_{x,i}$ to make the modules move in favor of reducing decap effectively;
2. $F^{\text{decap}}_{x,i}$ is not disturbing the spread force $F^{\text{move}}_{x,i}$ too much so that the placement can still converge to be overlap-free.

This trade-off is achieved by scaling the gradient of potential function $\Psi(x,y)$ properly. First, the potential function $\Psi(x,y)$ is scaled up to the comparable magnitude as the $\Phi(x,y)$. As a consequence, the gradient of the potential is scaled up properly. Usually the potential $\Psi(x,y)$ will be scaled up by $\beta = \frac{2}{V_{\text{dd}}-V_{\text{thre}}}\cdot\lambda$, where λ is defined as a weight parameter in (5) and set to 1.0 in our algorithm.

Second, the gradient of $\Psi(x,y)$ is scaled down when it conflicts with the gradient of $\Phi(x,y)$. As shown in Figure 3, the target position according to the gradient $\nabla\Psi_i$ is $(\hat{x}_i,\hat{y}_i)$, which is contrary to the target position $(\dot{x}_i,\dot{y}_i)$ according to the gradient $\nabla\Phi_i$, where the gradient $\nabla\Psi_i$ is $\beta\cdot q_i\cdot(\frac{\partial\Psi}{\partial x},\frac{\partial\Psi}{\partial y})\Big|_{(x',y')}$ and gradient $\nabla\Phi_i$ is $w_i\cdot(\frac{\partial\Psi}{\partial x},\frac{\partial\Psi}{\partial y})\Big|_{(x',y')}$. Then we scale down the gradient to $\gamma_i\cdot\nabla\Psi_i$, as shown by the red arrow in Figure 3, to make sure the module does not go against the gradient $\nabla\Phi_i$, i.e. to make sure potential $\Phi(x,y)$ decreases. The scalar γ_i is calculated by

$$\gamma_i = \min\left(1,\ 0.9\times\frac{<\nabla\Phi_i\cdot\nabla\Phi_i>}{<\nabla\Psi_i\cdot\nabla\Phi_i>}\right) \qquad (16)$$

where operator $<\cdot>$ is the inner product.

After scaling the forces properly, we can obtain the new move force vector

$$F^{\text{decap}}_{x} = \dot{Q}_x\cdot(\Delta\boldsymbol{x}+\beta\cdot\Upsilon\cdot\Psi_x)$$

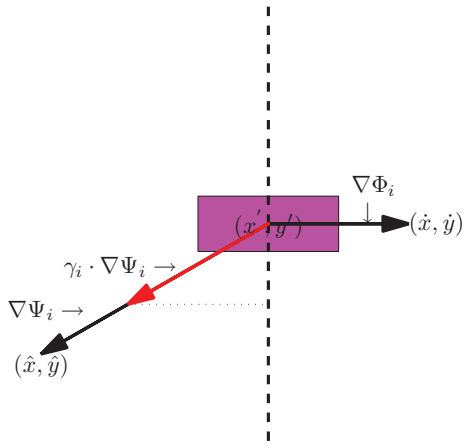

Figure 3: Scaling of the gradient of potential function.

where Ψ_x is the gradient vector in the x direction, $\dot{Q}_x = \text{diag}(\dot{q}_i)$ and $\Upsilon = \text{diag}(\gamma_i)$.

5.2 The Flow of Decap Efficient Placement Algorithm

Algorithm 2 Decap Aware Placement

```
Load the specification of power supply network
Initial the placement
Initialize weight matrices Ċx,Q̇x
Set β = 2/(Vdd − Vthre)
while Ω > α do
  Create supply-and-demand D^sup_mod(x, y),D^dem_mod(x, y)
  Calculate Φ(x, y) by solving (8)
  Calculate L and W by (13) and (14)
  Create demand-and-supply L(x, y),W(x, y) by interpolation
  Calculate Ψ(x, y) by solving (15)
  Create Cx,Φx,Ψx
  Calculate the scaling matrix Υ by (16)
  Solve equation (17) w.r.t. Δx
  Update module position x by Δx
  Quality control for both Ċx,Q̇x
end while
Legalize the placement
```

With the additional forces representing decap demands, we again need to balance all the forces in the placement

$$F_x^{\text{net}} + F_x^{\text{hold}} + (F_x^{\text{move}} + F_x^{\text{decap}}) = 0 \qquad (17)$$

$$\Longrightarrow \quad (C_x + \dot{C}_x + \dot{Q}_x)\Delta\boldsymbol{x} = -(\dot{C}_x\Phi_x + \beta\Upsilon\dot{Q}_x\Psi_x)$$

Plugging the equation (17) to the placement process, we present the decap efficient placement algorithm as shown in Algorithm 2.

6. EXPERIMENTAL RESULTS

Algorithm 2 has been implemented with MATLAB and C++, and is tested on a Linux workstation with 2.33G Hz Intel Xeon CPU and 8 GB memory. We also implemented Kraftwerk2 algorithm 1 with MATLAB and C++ for comparison. In both placement algorithms, we choose the upper bound of the overlap ratio to be $\alpha = 10\%$. The legalization of the placement in both algorithm 1 and algorithm 2 is carried out with Capo detail placer [1].

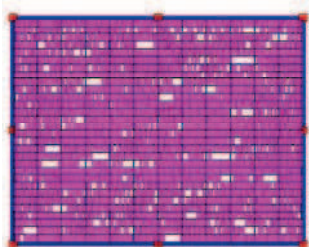

Figure 4: daiTest1 by algorithm 1

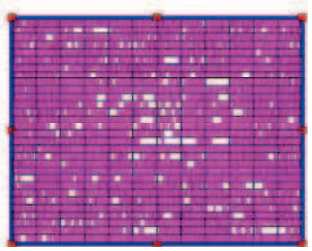

Figure 5: daiTest1 by algorithm 2

For the power supply, we use a mesh grid with a peripheral core ring and 8 pads uniformly allocated on the core ring. We specify $V_{dd} = 1.5V$ and $V_{\text{thre}} = 1.4V$. The power supply network is pre-allocated and the pitches and wire widths are adjusted properly before the placement, assuming all the current sources are uniformly distributed (because the current distribution remains unknown until the placement is given). Notice the pre-allocated power supply network is not necessarily a uniform grid and the more accurate non-uniform current source model obtained from statistical vectorless simulation is used in our co-synthesis algorithm 2. The *demand-supply* system (namely $\boldsymbol{L}$ and $\boldsymbol{W}$) is calculated by simulation of the power supply network, using the macro-modeling [16] and chain-reduction [9] techniques. Figure 4 and 5 show the placement results of daiTest1 test case obtained by algorithm 1 and 2 with the power supply network, respectively. After the global and detail placements, the decap budget problem (3) is then solved by the sensitivity based non-linear programming method [3], and the decap is placed to the empty space within each row of the placement, as in [3] and [12]. We choose the sensitivity based decap optimization algorithm because it produces minimal decap budget as reported.

We test our algorithm on the design benchmarks used in [3]. Table 1 shows the wire length and decap comparisons between the placement obtained by algorithm 1 and by algorithm 2. Column *#Node* is the number of modules in design. Column *#Iter* is the number of iterations in placement. Column *#Vio* is the number of violating nodes after placement. Column Δ *HPWL* and Δ *Decap* are the difference of the wire-length and decap amount obtained by our algorithm with respect to the original algorithm Kraftwerk. *P_Times(s)* is the CPU time of placement. *D_Times(s)* is the CPU time of decap optimization.

As we can see from Table 1, the placements obtained by our algorithm maintain wire length quality similar to that of the original placer, with an increase of only 0.48% on average. At the same time, on average of more than 35% decap has been reduced for the new placement. Also, we can see the iteration number of the placement has only increased slightly, which proves that our algorithm converges at a sufficient rate. The placement time of our algorithm is comparable to that of Kraftwerk due to the efficient decap assessment by macro-modeling.

7. CONCLUSION AND FUTURE WORK

		Kraftwerk2						P/G Placer							
Testcase	#Node	#Iter	HPWL(m)	P_time(s)	#Vio	Decap	D_Time(s)	#Iter	HPWL(m)	Δ HPWL	P_Time(s)	#Vio	Decap	Δ Decap	D_Time(s)
daiTest1	1309	36	0.0860957	2.48471	116	639.977 pf	79.906	22	0.0858811	-0.25 %	1.74753	69	358.287 pf	-44.02 %	48.1028
u_cnt1000	7492	21	1.12919	5.47127	287	2033.26 pf	361.618	22	1.12108	-0.72 %	9.43826	111	996.527 pf	-50.99 %	323.32
u05614	32112	38	3.23664	68.8028	5649	19108.8 pf	3244.18	44	3.34354	+3.30 %	112.355	3736	12486.9 pf	-34.65 %	2665.59
u08421	48168	36	4.9139	112.155	7488	29363.2 pf	3743.48	41	4.95232	+0.78 %	166.633	4409	18926.5 pf	-35.54 %	4086.49
u11228	64224	38	6.31085	195.307	9883	43819.2 pf	5774.39	45	6.36131	+0.80 %	315.778	5473	29574 pf	-32.51 %	7372.6
u14035	80280	42	8.55715	338.016	12695	49071 pf	8038.17	50	8.63332	+0.89 %	492.324	7029	36122.1 pf	-26.39 %	7250.13
u19649	112392	43	11.9325	552.366	16462	65788.6 pf	13428.6	52	11.9878	+0.46 %	749.973	11137	49478.2 pf	-24.79 %	15447.5
u28070	160560	42	16.9087	1079.2	19819	78125.3 pf	22163.2	48	16.8088	-0.59 %	1220.78	11283	62936.6 pf	-19.44 %	22856.5
u56140	321120	43	38.162	3723.38	19769	110051 pf	94077.6	42	38.0401	-0.32 %	3713.1	6193	49328.8 pf	-55.18 %	16422.5
Average										+0.48 %				-35.95 %	

Table 1: Comparison between the Kraftwerk and the decap aware placement algorithm

In this paper, we have proposed a new placement algorithm which considers the explicit reduction of decap under voltage drop noise constraints. By transforming decap budgeting to a linear programming problem, we developed a new decap-related merit that can be calculated fast enough to be integrated in the placement process. The new merit is modeled as an additional force in a force-directed placer to guide the module's move in favor of reducing decap budget. The decap-efficient placement method explores the non-uniform power delivery capability of power grid networks to reduce the gaps between power supply and demands. Experimental results show that the new method can reduce the minimum decap area by 35% with a wire length increase of only 0.5% at nearly the same computational cost, which is efficient for practical problems.

Placement with macro modules is more difficult to handle because 1) the cells inside the macro module have to be handled by macro-modeling first. 2) the legalization will move the macro modules obviously, which will destroy the power supply profile. Therefore, this work will be left for the future.

Also, our algorithm will be extended to consider the inductive noise on package, which is significant for dynamic power integrity.

8. REFERENCES

[1] J. A.Roy, S. N. Adya, D. A. Papa, and I. L. Markov. Min-cut floorplacement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(7):1313–1326, 2006.

[2] J. Fan, I.-F. Liao, X.-D. Sheldon, Y. Cai, and X. Hong. Localized on-chip power delivery network optimization via sequence of linear programming. In *Proceedings of the International Symposium on Quality Electronic Design*, pages 272–277, 2006.

[3] J. Fu, Z. Luo, X. Hong, Y. Cai, S. X.-D. Tan, and Z. Pan. A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery. In *Proceedings of the Asia South Pacific Design Automation Conference*, pages 505–510, jan 2004.

[4] A. B. Kahng, B. Liu, and S. X.-D. Tan. Efficient decoupling capacitor planning via convex programming methods. In *Proceedings of the International Symposium on Physical Design*, pages 102–107, 2006.

[5] A. B. Kahng, B. Liu, and Q. Wang. Supply voltage degradation aware analytical placement. In *Proceedings of the International Conference on Computer Design*, pages 437–443, 2005.

[6] A. B. Kahng and Q. Wang. A faster implementation of aplace. In *Proceedings of the International Symposium on Physical Design*, pages 218–220, 2006.

[7] A. Kennings and K. Vorwerk. Force-directed methods for generic placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(10):2076–2087, Oct 2006.

[8] H. Li, J. Fan, Z. Qi, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong. Partitioning-based approach to fast on-chip decoupling capacitor budgeting and minimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(11):2402–2412, Nov. 2006.

[9] Z. Pan, Y. Cai, S. X.-D. Tan, Z. Luo, and X. Hong. Transient analysis of on-chip power distribution networks using equivalent circuit modeling. In *Proceedings of the International Symposium on Quality Electronic Design*, pages 63–68, 2004.

[10] S. Pant and E. Chiprout. Power grid physics and implications for cad. In *Proceedings of the Design Automation Conference*, pages 199–204, 2006.

[11] P. Spindler, U. Schlichtmann, and F. M. Johannes. Kraftwerk2 – a fast force-directed quadratic placement approach using an accurate net model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(8), 2008.

[12] H. Su, S. S. Sapatnekar, and S. R. Nassif. An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts. In *Proceedings of the International Symposium on Physical Design*, pages 68–73, 2002.

[13] X. Wang, Y. Cai, , X. Hong, S. X.-D. Tan, and J. Relles. An efficient decoupling capacitance optimization using piecewise polynomial models. In *Proceedings of the Design, Automation and Test in Europe*, pages 1190–1195, March 2009.

[14] C. K. S. Zhao and K. Roy. Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21:81–92, Jan 2002.

[15] M. Zhao, R. Panda, S. Sundareswaran, S. Yan, and Y. Fu. A fast on-chip decoupling capacitance budgeting algorithm using macromodeling and linear programming. In *Proceedings of the Design Automation Conference*, pages 217–222, 2006.

[16] M. Zhao, R. V. Panda, S. S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw. Hierarchical analysis of power distribution networks. In *Proceedings of the Design Automation Conference*, pages 150–155, jun 05-09 2000.