Decoupling Capacitor Planning With Analytical Delay Model on RLC Power Grid

Ye Tao and Sung Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology
yetao@ece.gatech.edu, limsk@ece.gatech.edu

***Abstract*— Decoupling capacitors (decaps) are typically used to reduce the noise in the power supply network. Because the delay of gates and interconnects is affected by the supply voltage level, decaps can be used to improve the circuit performance as well. In this paper, we present the analytical delay model under IR drop, Ldi/dt noise, and decaps to study how decaps affect both the gate and interconnect delay. Given a floorplanning solution, we study how to allocate the whitespace for decap insertion so that the delay is minimized under the given noise and area constraint. We employ the Sequential Linear Programming method to solve the non-linear whitespace allocation problem. Our experimental results show that intelligent decap allocating decap makes further delay reduction possible without adding any additional decap.**

I. Introduction

Signal integrity is a very important issue in VLSI technology. Simultaneous switching of digital circuit elements can cause considerable IR-drop and Ldi/dt noise in the power supply network. This power supply noise may cause logic faults. On-chip decoupling capacitors (decap) are widely used to mitigate the power-supply noise problem. By charging up during steady state conditions, decaps can be assumed as the role of the power supply and provide the current needed during the simultaneous switching of multiple functional blocks.

Non-ideal power supplies also affect circuit delay significantly. The IR drop and Ldi/dt noise along the power supply network decrease the Vdd level, which in turn slows down the gates connected to it. While buffers have been a popular method to reduce the circuit delay, decap is another useful method to improve circuit performance. If decaps are inserted carefully, both the power-supply noise and circuit delay problems can be mitigated simultaneously. Decaps are usually inserted in the whitespace of a layout. Since it is the floorplanning stage that determines the whitespace regions, decap planning is highly effective during the floorplanning stage for optimizating delay and noise. A designer can choose the floorplan with the best performance, because the noise/decap-aware delay varies in different floorplans. However, more optimization time will be paid for this selection. The results for our integrated floorplanning algorithm shows this trade-off.

The history of utilizing decaps for delay minimization is short. The work in [1] introduces a gate delay model with respect to decap. Delay in this model is a linear function of voltage drop, and the coefficients are determined by multiple regression analysis. The authors in [2] apply this model to decap planning problem to meet the delay target by Lagrange relaxation. Compared with [1] and [2], our work considers both gate and interconnect delay. In addition, our analytical model captures the highly non-linear relation between delay and decap. The work in [3] considers both power-supply noise and delay minimization in gate-level placement. Their delay model is related to the voltage change by a non-linear function.

Leakage is another significant issue in decap planning, because the leakage current flowing through decap raises the temperature of the chip. In a practical design, the leakage current is positive proportion to the area of decap, so the decap area should be restricted, although more decaps can obtain better performance on stabilizing the voltage level at each power grid. On the other side, if the voltage level is close to the ideal Vdd, decap insertion is not useful. Therefore, an efficient sequential linear programming algorithm is proposed to planning decap with the leakage and other constraints.

This material is based upon work supported by the National Science Foundation under CAREER Grant No. CCF-0546382.

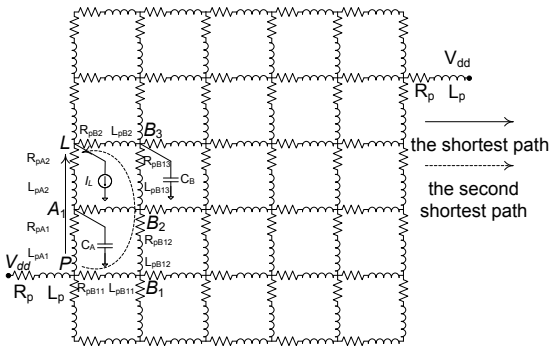

Fig. 1. Power-supply network.

Additonal work on decap-aware floorplanning for 2D circuits is presented by [4]. The objective is to minimize the floorplan area while suppressing the power supply noise below the specified limit. The authors of [5] presents a circuit model to allow non-adjacent decap access instead of only adjacent decap in [4]. Iteration of linear programming in decap planning is applied in [6], [7]. A time-domain method of power-grid analysis is proposed in [8]. Compared with these works, our work considers both noise and delay objectives. The contributions of this paper are as follows:

- We present an analytical resistance-inductance-capacitance (RLC) delay model that captures the non-linear relation between gate/interconnect delay and decap/IR-drop/Ldi/dt noise. Our model correctly captures the impact of decap planning on the noise and delay objectives, and can directly calculate the delay from the block at any place in the power grid. SPICE simulation results validate the usefulness of our formula.
- We present an effective floorplanning-level decap planning algorithm that utilizes the noise/decap-aware delay model for rigorous performance optimization under noise and area constraint. We employ a highly effective linear approximation to solve the original non-linear programming in a sequential fashion.

II. Decap-aware Delay and Noise Analysis

A. Power-supply network and RLC Circuit Model

The model of power-supply network used in this work is shown as in Figure 1. R_P and L_P are the package parasitics of the power pins, and others are on-chip components. The capacitors, such as C_A and C_B, are the combinations of the parasitic capacitors (not drawn in the power grid due to limited space) and the decap. We consider the nearest power-supply pin and both the shortest and second shortest

978-3-9810801-5-5/DATE09 © 2009 EDAA

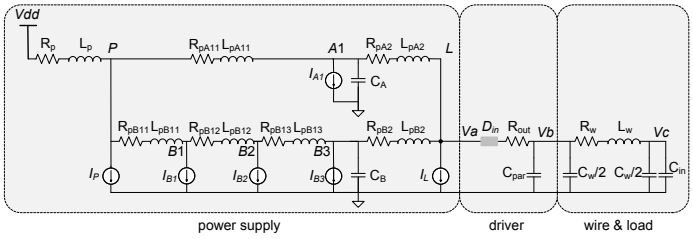

Fig. 2. Our physical and circuit model for noise/decap-aware delay analysis. The RLC circuit model consists of a non-ideal power supply network, decaps, a driver, a wire, and a load.

paths from that pin[1]. The SPICE simulation in [4] shows that the error caused by this assumption is less than 10%. More complex models without these two assumptions can be designed based on [9]. However, these complex models will be too time consuming for use in the floorplanning stage. Our model with these two limiting assumptions can also obtain good approximation as demonstrated by [4], so we apply both of them in the analysis.

Figure 2 shows the circuit model used for the noise/decap-aware delay analysis. The physical model consists of a non-ideal power supply (due to IR and Ldi/dt noise), decaps, a RC buffer model that is attached to the non-ideal power supply network, a wire, and a final capacitive load. This RLC-model has the following three major parts:

1) Power supply: the resistors, inductors, and decaps between the power pin and the load have the same notations as in Figure 1. I_P, I_{A1}, I_{B1}, I_{B2}, and I_{B3} are the current to the power-supply grids adjacent to P, $A1$, $B1$, $B2$, and $B3$ respectively (these loads are not shown in Figure 1 because of limited space). I_L is the current of the load without the output buffer. The non-ideal voltage level is denoted by V_a.
2) Driving buffer: we use a first-order circuit model to represent the driving buffer that is connected to the non-ideal power supply. The voltage level at the output of the buffer is denoted by V_b. D_{in} denotes the buffer delay, R_{out} denotes the output resistance, and C_{par} denotes the parasitic capacitance of the buffer.
3) Wire and load: we use the lumped π-model to represent the interconnect, where R_w, C_w, and L_w respectively denote the wire resistance, wire capacitance, and wire inductance. C_L denotes the loading capacitance, and V_c is the voltage level at the input of the load.

The delay analysis measures the delay from node V_a to V_c in Figure 2.

B. Delay Analysis

The voltage level at each node of the circuit in Figure 2 can be solved by Kirchhoff's current law (KCL) and Kirchhoff's voltage law (KVL). The circuit can be separated into five parts denoted by subscripts *g*, *c*, *l*, *v*, and *i*. They represent the branches containing resistors, capacitors, inductors, voltage sources, and current sources respectively. Because of space limited, we represent the circuit as the matrix form. $\mathbf{A}_g$ is incidence matrix of resistors, the same as for other parts. Therefore, KCL can be written as

$$\mathbf{A}_g\mathbf{I}_g + \mathbf{A}_c\mathbf{I}_c + \mathbf{A}_l\mathbf{I}_l + \mathbf{A}_v\mathbf{I}_v + \mathbf{A}_i\mathbf{I}_i = \mathbf{0}.$$

KVL can be written as

$$\mathbf{V}_g = \mathbf{A}_g^T\mathbf{V}_n, \mathbf{V}_c = \mathbf{A}_c^T\mathbf{V}_n, \mathbf{V}_l = \mathbf{A}_l^T\mathbf{V}_n,$$
$$\mathbf{V}_v = \mathbf{A}_v^T\mathbf{V}_n, \mathbf{V}_i = \mathbf{A}_i^T\mathbf{V}_n$$

[1]These two pathes for other scenarios of power-supply network can be generated similarly as in Figure 1, which is used for explaining easily.

where$\mathbf{V}_n$ is the vector of all node voltages. The constitutive relations of the resistance, capacitance, and inductance branches are $G\mathbf{V}_g = \mathbf{I}_g$, $sC\mathbf{V}_c = \mathbf{I}_c$, and $sL\mathbf{I}_L = \mathbf{V}_L$, where G, C, and L are the diagonal matrices whose elements are the values of conductance, capacitance, and inductance in the corresponding branch, respectively. The decaps and inductors along the power grids are already charged before this transient analysis. Therefore, an additional voltage source is connected to each decap in parallel, and an additional current source is connected to each inductor except L_W in series.

By applying the modified nodal analysis (MNA) method [10], the following matrix equation can be obtained:

$$(\mathbf{G} + s\mathbf{C})\mathbf{X} = \mathbf{b}$$

Where,

$$\mathbf{G} = \begin{bmatrix} \mathbf{A}_g G \mathbf{A}_g^T & \mathbf{A}_l & \mathbf{A}_v \\ \mathbf{A}_l^T & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_v^T & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{A}_c C \mathbf{A}_c^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -L & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$\mathbf{X} = \begin{bmatrix} \mathbf{V}_n & \mathbf{I}_l & \mathbf{I}_v \end{bmatrix}$, and $\mathbf{b} = \begin{bmatrix} -\mathbf{A}_i\mathbf{I}_i & 0 & \mathbf{V}_v \end{bmatrix}$. $\mathbf{X}$ is the vector of the transfer functions in s-domain. These transfer functions can be used to obtain the response in time domain. Therefore, the delay of V_c can be calculated from the moments of its transfer function. These moments are denoted as m_i $(i \geq 0)$, one element in

$$\begin{cases} \mathbf{m}_0 = \mathbf{G}^{-1}\mathbf{b} \\ \mathbf{m}_{i+1} = -\mathbf{G}^{-1}\mathbf{C}\mathbf{m}_i \quad (i \geq 0) \end{cases}$$

where $\mathbf{m}_i$ is the ith moment vector of the transfer functions of all nodes in the circuit, and $\mathbf{G}^{-1}$ is calculated by LU decomposition. Several formulas of delay calculation by moment match have been proposed in [7], [11], [10], and [12]. However, in the noise/decap-aware delay model, poles and zeros should both be considered, which is different with [12], because the decaps and inductors along the power grid have already been charged as part of the initial conditions. Therefore, a new set of formulas should be proposed. With consideration up to s^2, the truncated transfer function of V_c is

$$H(s) = \frac{b_0 + b_1 s + b_2 s^2}{1 + a_1 s + a_2 s^2}.$$

The moment expansion of $H(s)$ is

$$H(s) = b_0 + (b_1 - b_0 a_1)s + (b_2 - b_0 a_2 - b_1 a_1 + b_0 a_1^2)s^2 + \cdots.$$

Matching the moments of $H(s)$ to the moment of the transfer function of node c obtained before, we arrive at,

$$\begin{cases} b_0 = m_0 \\ b_1 = a_1 m_0 + m_1 \\ b_2 = a_2 m_0 + a_1 m_1 + m_2 \end{cases}, \quad \begin{cases} a_1 = \frac{m_2 m_3 - m_1 m_4}{m_1 m_3 - m_2^2} \\ a_2 = \frac{m_2 m_4 - m_3^2}{m_1 m_3 - m_2^2} \end{cases}.$$

Let $\Delta = \sqrt{|a_1^2 - 4a_2|}$ and $p_{1,2} = (a_1 \mp \Delta)/2a_2$. When $a_1^2 - 4a_2 > 0$, the poles of this system are $-p_1$ and $-p_2$. Therefore, the response of V_c is

$$v_c(t) = b_0 - \frac{b_0 - b_1 p_1 + b_1 p_1^2}{a_2 p_1 (p_2 - p_1)} e^{-p_1 t} - \frac{b_0 - b_1 p_2 + b_1 p_2^2}{a_2 p_2 (p_1 - p_2)} e^{-p_2 t}.$$

Because $p_1 < p_2$, the second term of response decreases much more rapidly than the first term. Therefore, only the dominant pole needs to be considered,

$$v_c(t) = b_0 - \frac{b_0 - b_1 p_1 + b_1 p_1^2}{a_2 p_1 (p_2 - p_1)} e^{-p_1 t}.$$

Assuming the signal rises from 0 to 50% of Vdd during the elapsed time, the solution to the following equation is the delay of the circuit under analysis:

$$b_0 - \frac{b_0 - b_1 p_1 + b_1 p_1^2}{a_2 p_1 (p_2 - p_1)} e^{-p_1 t} = \frac{1}{2}.$$

Solving for t, the analytical noise/decap-aware model is as follows:

$$D = D_{in} + \frac{1}{p_1} \ln \frac{b_0 - b_1 p_1 + b_1 p_1^2}{p_1 \Delta (b_0 - 0.5)}. \quad (1)$$

The total delay includes the buffer delay D_{in}, the value of which can be measured at different voltage levels and used during decap planning by table look-up.

Similarly, when $a_1^2 - 4a_2 < 0$, let $p = a_1/2a_2$ and $q = \Delta/2a_2$, the response is

$$v_c(t) = b_0 - e^{-pt} \left[M \sin(qt) + N \cos(qt) \right] / q$$

where

$$M = \frac{a_1 b_2 + 2a_2 b_0 - 2a_2 b_1}{2a_2^2}, N = \frac{a_2 b_0 - b_2}{a_2}.$$

Let $v_c(t) = 50\% Vdd$, this recursive equation can be solved by approximating based on the first moment, m_1. Therefore, the total delay is

$$D = D_{in} + \frac{(b_0 - 0.5) e^{-\frac{a_1 m_1}{2a_2}}}{\sqrt{M^2 + N^2}} - \frac{2a_2}{\Delta} \tan^{-1}(\frac{N}{M}). \quad (2)$$

When $a_1^2 - 4a_2 = 0$, the response is

$$v_c(t) = b_0 + Pe^{-pt} - Qte^{-pt}$$

where

$$P = \frac{b_1 - a_2 b_0}{a_2}, Q = \frac{2b_2 - a_1 b_1 + 2a_2 b_0}{a_1 a_2}.$$

Let $v_c(t) = 50\% Vdd$, the total delay is

$$D = D_{in} + \frac{P}{Q} - \frac{a_1}{2} LabmertW \left(\frac{1 - 2b_0}{a_1 Q} e^{\frac{2P}{a_1 Q}} \right). \quad (3)$$

$w = LabmertW(x)$ [13] is the solution of $w * exp(w) = x$, which can be solved by iteration. If the interconnection is a multi-terminal net, more nodes will be added into the circuit model. Our method can also be applied to calculate the moments at these additional nodes. To concentrate on the effect of the decaps, only two-terminal nets are used in this paper[2]. Section IV-A presents the SPICE simulation results of the delay model.

C. Noise Analysis

We use the method presented in [4] and [5] to calculate IR-drop and Ldi/dt noise from a given floorplan. A uniform RLC-mesh is used to model the P/G network. The edges of the mesh have resistive impedances and inductances. The mesh contains power supply nodes and power consuming nodes. The *dominant paths* of a block are the shortest path and the second shortest path from the nearest power supply to the block requiring most of the current. The voltage noise of a block, denoted V_{noise}^k, is simply the voltage change caused by the resistors and inductors along its dominant paths. We also consider the extra current provided by the decoupling capacitors added to some of the mesh nodes. Depending on the location of the blocks, some edges are included in more dominant paths than others. In this case, the voltage drop is larger on those "popular" edges and the blocks that use them.

In the worse case, a module would draw all of its switching current from its decap. Let $Q^{(k)} = \int_0^{t_s} I^{(k)}(t) \cdot dt$ denote the maximum charge drawn from the power supply by block $B^{(k)}$, where $I^{(k)}(t)$ is the current demand, and t_s is the switching period. The decap demand is calculated as follows:

$$\Theta^{(k)} = \max(1, \frac{V_{noise}^k}{V_{tol}}), \quad k = \{1, 2, \cdots, M\}, \quad (4)$$

$$C_k = \frac{(1 - 1/\Theta^{(k)}) Q^{(k)}}{V_{tol}}, \quad k = \{1, 2, \cdots, M\} \quad (5)$$

where V_{tol} denotes the voltage-drop constraint. This worst-case demand is used during our optimization as the demand.

[2]We decompose multi-pin nets into a set of source-sink two-pin nets, because our decap planing is in floorplan phase.

III. Noise and Delay-aware Decap Planning

A. Problem Formulation

The decaps are usually inserted in the whitespace of a layout. Since it is the floorplanning stage that determines the whitespace regions, decap planning is highly effective at the floorplanning stage. One way to plan the decaps is to allocate the whitespace after the floorplan is fixed (= post-floorplanning). This problem is stated as follows:

Noise/Delay-aware Decap Planning: Given a floorplanning solution, we seek an efficient way to allocate the whitespace in the floorplan for decap insertion. The objective is to minimize the delay induced by the combined effect of the IR drop, Ldi/dt noise, and decap. The constraints are the voltage drop, total current leakage, and the overall floorplan area.

We use the noise/decap-aware delay model shown in Equations (1-3) to measure point-to-point delay values in the floorplan. We note that the delay of a wire is a non-linear function of IR-drop, Ldi/dt noise, decap, and the wire length. We employ the sequential linear programming (SLP) method to solve this highly non-linear, post-floorplan whitespace allocation problem.

B. Non-linear Optimization Formulation

Considering the area, noise, and leakage constraints, our non-linear decap planning problem is formulated as follows:

NLP: Minimize

$$\sum_{i,j \in S_b} D_{ij}. \quad (6)$$

Subject to

$$\sum_{k \in S_w} x_{ik} \geq \frac{C_i}{C_{ox}}, \ i \in S_b \quad (7)$$

$$\sum_{i \in S_b} x_{ik} \leq A_k, \ k \in S_w \quad (8)$$

$$\sum_{i \in S_b} \sum_{j \in S_w} I_{ij} \leq I_{total} \quad (9)$$

$$x_{ik} = 0, \ ws_k \text{ is not adjacent to } blk_i \quad (10)$$

where S_b is the set of functional blocks, and S_w is the set of whitespace regions in the given floorplan. D_{ij} is the delay from block i to block j based on our noise/decap-aware delay model (= Equations (1-3)). x_{ik} is the portion of whitespace k adjacent to block i for decap insertion. C_{ox} is the capacitance per unit area. The area of whitespace j is A_j. The decap requirement of block i, denoted by C_i, is computed based on the voltage-drop constraint as shown in Equation (5).

Equation (6) is our objective function, where the total delay among all two-pin wires is minimized. Equation (7) is the decap demand (= voltage-drop noise) constraint, which states that the total amount of decap adjacent to a block should exceed its demand for decap. Our voltage-drop and decap demand analysis is presented in Section II-C. Equation (8) is the whitespace area constraint, which states that the total amount of whitespace used for decap allocation cannot exceed the amount available. Equation (9) limits the total current leakage of

decap, which is proportional to the total area of the decaps. Equation (10) states that only adjacent whitespace may be used to meet the decap demand of the blocks.

C. Sequential Linear Programming

We solve the non-linear optimization problem formulated in Section III-B using the sequential linear programming method, because the objective function of this problem is complicated, while the constraints are all linear. Our method consists of two steps, namely, initial planning, and iterative improvement. The goal of the initial planning step is to obtain a feasible solution for subsequent iterative improvement. During initial planning, our primary goal is to obtain a solution that satisfies the voltage-drop constraint, that is, a solution that meets the decap demand of the blocks. Note that the floorplan area will be expanded if the existing whitespace is not enough to meet the decap demand. In this case, we perform the voltage-drop analysis and decap demand computation again to reflect the changes in the floorplan.

We improve the delay of the initial solution during the iterative improvement step while suppressing noise. In this step, the floorplan area is not further expanded, but the decap allocation is re-adjusted to improve delay under the noise constraints. During the SLP-based improvement step, the non-linear delay objective is approximated linearly. In addition, a set of bounds are added to the solution space to ensure that the linear approximation is close enough to the original objective function. The SLP-based iterative improvement step terminates if the degree of updates in the solution is minimal.

1) Initial Planning Step: The initial planning step is formulated as the following LP:

Minimize

$$\sum_{i \in S_b} \left\lfloor \frac{C_i}{C_{ox}} - \sum_{k \in S_w} x_{ik} \right\rfloor \tag{11}$$

Subject to

$$\sum_{i \in S_b} x_{ik} \leq A_j, \ k \in S_w \tag{12}$$

$$x_{ik} = 0, \ ws_k \text{ is not adjacent to } blk_i \tag{13}$$

The objective is to minimize the difference between the decap demand and the decap adjacent to each block. When this difference becomes zero, the decap demand is fully met and the voltage-drop noise constraint is satisfied. We minimize this difference because the amount of existing whitespace in the given floorplan may not be enough. In this case, floorplan expansion is necessary to add more whitespace. Thus, we repeat this iterative LP minimization and floorplan expansion until our objective becomes zero, therefore, the decap demand is satisfied. [3] Constraints (12) and (13) in this LP are identical to Equations (8) and (9).

2) Iterative Improvement Step: Let C^n_{Ai} and C^n_{Bi} denote the amount of decap adjacent to block i along the shortest and the second shortest path respectively during the n-th iteration. The only variable is the decap in this step, because the floorplan is fixed. Therefore, the delay can be represented as $D^n_{ij} = f(C^n_{Ai}, C^n_{Bi})$. The SLP-based iterative improvement step proceeds as follows:

- step 0: set $n = 0$; Obtain C^n_{Ai} and $C^n_{Bi} (\forall i \in S_b)$ from the whitespace allocation results of the Initial Planning Step.
- step 1: $n = n + 1$;
- step 2: obtain $D^n_{ij} = f(C^n_{Ai}, C^n_{Bi}) (\forall i \in S_b)$, the linearized version of our delay objective function, using $f(C^{n-1}_{Ai}, C^{n-1}_{Bi}))$.
- step 3: solve LP^n, the LP of the n-th iteration.

[3]The total number of iterations depends on the noise constraint and the initial floorplan. Our experimental results show the amount of expansion needed for each circuit in Section IV-B.

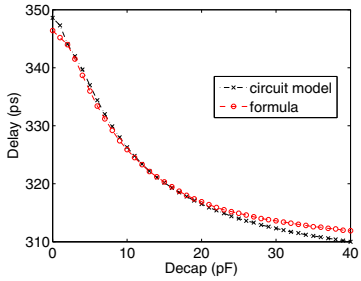

Fig. 3. Validation of our analytical noise/decap-aware delay model based on SPICE simulation.

- step 4: if $|D^n_{ij} - D^{n+1}_{ij}| < \epsilon \ (\forall i \in S_b)$, then terminate; otherwise, goto step 1.

Step 2 of the SLP requires linearization of the objective function used in the original NLP, that is, Equation (6). Therefore, $D^n_{ij} = f(C^n_{Ai}, C^n_{Bi})$, the non-linear function between the delay and decap, is approximated into a linear form within the following bounds:

$$C^n_{Ai} \in \left[C^{n-1}_{Ai} - \delta_{iA1}, C^{n-1}_{Ai} + \delta_{iA2}\right],$$
$$C^n_{Bi} \in \left[C^{n-1}_{Bi} - \delta_{iB1}, C^{n-1}_{Ai} + \delta_{iB2}\right]$$

where C^{n-1}_{Ai} and C^{n-1}_{Bi} denotes the assignment variables from the previous iteration. The following constraints are derived from the above bounds.

$$C_{ox} \sum_{k \in S_w} x^{n-1}_{iXk} - \delta_{iX1} \leq C_{ox} \sum_{k \in S_w} x^n_{iXk}, \ \forall i \in S_b, X = A \text{ or } B, \tag{14}$$

$$C_{ox} \sum_{k \in S_w} x^n_{iXk} \leq C_{ox} \sum_{k \in S_w} x^{n-1}_{iXk} + \delta_{iX2}, \ \forall i \in S_b, X = A \text{ or } B \tag{15}$$

where the additional notation X as a subscript of x means using only the decap along the path X.

The linear approximation of $f(C^n_{Ai}, C^n_{Bi})$ within the above bound is

$$g(C^n_{Ai}, C^n_{Bi}) = f(C^{n-1}_{Ai}, C^{n-1}_{Bi}) + a(C^n_{Ai} - C^{n-1}_{Ai}) + b(C^n_{Bi} - C^{n-1}_{Bi}) \tag{16}$$

where

$$a = \frac{f(C^{n-1}_{Ai} + \delta_{iA2}, C^{n-1}_{Bi}) - f(C^{n-1}_{Ai} - \delta_{iA1}, C^{n-1}_{Bi})}{\delta_{iA1} + \delta_{iA2}},$$
$$b = \frac{f(C^{n-1}_{Ai}, C^{n-1}_{Bi} + \delta_{iB2}) - f(C^{n-1}_{Ai}, C^{n-1}_{Bi} - \delta_{iB1})}{\delta_{iB1} + \delta_{iB2}}.$$

Our LP at the n-th iteration is formulated as follows:

Minimize

$$\sum_{i,j \in S_b} D^n_{ij}, \tag{17}$$

subject to (7), (8), (9), (10), (14), (15), (16)

$$D^n_{ij} = g(C^n_{Ai}, C^n_{Bi}).$$

IV. Experimental Results

We implemented our decap planning algorithm using C++/STL and ran it on a 1.2GHz Celeron-M processor with 1GB memory. The parameters used in this paper are based on $70nm$ technologies except for Table II, where we use $250nm$ for comparison with existing works. We report our results based on the MCNC and GSRC benchmark circuits. Power and ground pins are all assumed to be uniformly distributed on the boundary of the floorplan.

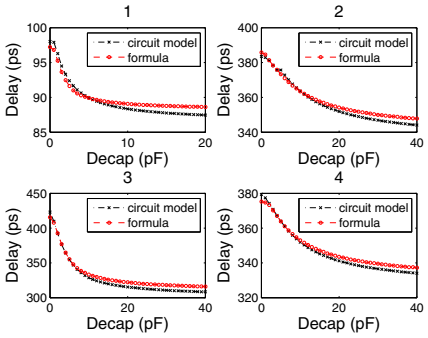

Fig. 4. Comparison of our model with SPICE simulation with other circuit parameters.

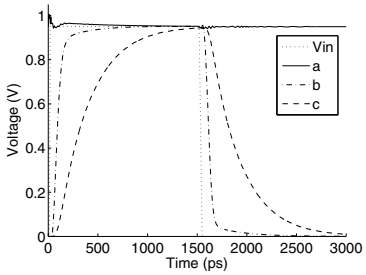

Fig. 5. SPICE simulation of the RLC network shown in Figure 2.

A. SPICE Simulation Results

Figure 3 shows the SPICE simulation of the RLC-model shown in Figure 2. We compare this to our analytical noise/decap-aware delay model, Equations (1-3). The length of the interconnect is $3mm$ so that we can analyze global connections common to block-level interconnects. We observe that our formulas closely match with the SPICE simulation. We also verify the non-linear relation between the delay and decap, where increasing decap size reduces the delay in a non-linear fashion.

Figure 4 shows simulation scenarios with different wire lengths, load sizes, power-supply networks, and distances from the block to the nearest decap. Table I shows the setups and maximum error of these experiments. We vary only one parameter for each experiment to observe the sensitivity of the model to each parameter.

Figure 3 and 4 show that our delay model is trustable in various scenarios on the floorplan. Figure 5 shows the SPICE-based voltage curves of various nodes for the RLC-circuit shown in Figure 2. V_{in} denotes the logic value applied to the input of the driver. $V(a)$ shows the non-ideal behavior of the power supply[4]. $V(b)$ corresponds to the output voltage of the driver, and $V(c)$ denotes the voltage at the input of the final load.

B. Decap Planning Results

We first compare our results to an existing work that is based on "effective decap distance" [5][5]. We use the $250nm$ technology parameters used in [5]. Table II shows the comparison between [5] and our algorithm in terms of floorplan area, decap cost, and runtime. Note that [5] does not report delay results since it only optimizes decap cost under a noise constraint. In addition, the current density values in both results are based on random assignment within the same range. Thus, a head-to-head comparison is not quite possible. However, these results show that our results are comparable to [5]. We use more up-to-date $70nm$ technology for all subsequent results.

[4]The curve of $V(a)$ is unsmooth when considering the effect of inductance.

[5]It is reported in [5] that the effective distance model outperforms the model in [4]. We obtained binary from the authors for comparison.

TABLE I
COMPARISON WITH DIFFERENT EXPERIMENTS

Figure	3	4-1	4-2	4-3	4-4
Wire length (mm)	3	1	3	3	3
Load size (times of minimum size)	25	25	100	25	25
Power path from decaps to the power pin (mm)	1 3	1 3	1 3	3 9	1 3
Power path from decaps to the block (mm)	1 1	1 1	1 1	1 1	3 3
Maximum error compared with SPICE simulation (%)	0.63	1.33	1.08	2.50	1.08

TABLE II
COMPARISON WITH EXISTING WORK [5]

	area	decap	runtime
apte			
noise-aware [5]	49662800	13.75	24
noise/delay-aware	50579600	12.56	19
xerox			
noise-aware [5]	21678300	5.20	29
noise/delay-aware	21142200	5.04	22
hp			
noise-aware [5]	9988280	1.76	34
noise/delay-aware	9857800	1.65	25
ami33			
noise-aware [5]	1237540	0.00	182
noise/delay-aware	1242130	0.17	188
ami49			
noise-aware [5]	40316000	10.83	448
noise/delay-aware	39261400	10.42	510

Table III shows a comparison among the following algorithms.

- Noise-C minimizes decap cost under a noise constraint (noise-C). Delay is ignored in this case. This is designed to reproduce results in [4] and [5] for a more up-to-date technology node. Therefore, only initial step is applied in this algorithm.
- Delay-O, noise-C minimizes delay (delay-O) under a noise constraint (noise-C). This algorithm corresponds to our non-linear programming formulation presented in Section III-C.
- "Integrated floorplan" integrates the delay-aware decap planning algorithm (= delay-O, noise-C) into the floorplanning process. Our Sequence-Pair-based [14] floorplanner performs decap planning at every move during the low temperature region of annealing, and choose the results with minimum delay comparing with others.

Note that the same value for the noise constraint is imposed on all of the above algorithms. Our two major observations are as follows:

- The "decap" and "% ave-dly reduce" columns show that the delay-oriented algorithm (= delay-O, noise-C) obtains better delay reduction at comparable decap cost compared to the pure noise-oriented algorithm (= noise-C). Therefore, further delay reduction is possible with a similar decap cost. Further investigation reveals that our delay-oriented algorithm allocats more decaps to the blocks with more interconnects incident to them. A similar trend can be observed from the "maximum delay" column (= maximum among all interconnect delay values).
- The area increase from floorplan expansion is reported under "% area expand" column. This area overhead, mainly used to satisfy the noise constraint, ranges from 0 to 12%. We note that the integrated floorplanning algorithm obtains smaller delay than other algorithms, because the delay of other floorplans may be smaller than the final floorplan after decap planning. Therefore, the selection of different floorplans can further reduce the delay in spite of the longer runtime. The total runtime is proportional to the number of iterations used in the SLP, and depends on the number of levels of temperature used for decap planning.

Lastly, Table IV shows the impact of the linearization bound on the delay function. Let the value of the decap in the pervious iteration is C, then the linearization bound is $[C/\rho, \rho C]$. We change ρ and

TABLE III
IMPACT OF DELAY OBJECTIVE AND NOISE CONSTRAINTS.

Circuit Algorithm	area	% area expand	decap	average delay	% ave-dly reduce	maximum delay	iteration	runtime
apte								
noise-C	54275600	11.87	3.09	1212	6.33	3962		17
delay-O, noise-C	54275600	11.87	2.98	1145	11.47	3821	6	19
integrated floorplan	57591300	10.37	3.38	1074	11.21	4015	32	37
xerox								
noise-C	21739700	5.82	1.23	419.4	3.29	1542		15
delay-O, noise-C	21739700	5.82	1.15	393.2	9.32	1479	7	23
integrated floorplan	22875200	8.56	1.34	369.5	9.97	1297	35	47
hp								
noise-C	9762910	0.58	0.42	715.7	2.07	2942		19
delay-O, noise-C	9762910	0.58	0.39	679.6	7.01	2808	7	28
integrated floorplan	9762910	0.58	0.39	679.6	7.01	2808	33	42
ami33								
noise-C	1278920	0.00	0.00	81.42	0.00	199.8		153
delay-O, noise-C	1278920	0.00	0.05	78.80	3.21	188.3	8	196
integrated floorplan	1302400	1.35	0.06	75.02	3.17	185.1	32	411
ami49								
noise-C	40678100	0.23	2.91	635.5	5.02	3212		280
delay-O, noise-C	40678100	0.23	2.86	569.4	10.87	2987	9	526
integrated floorplan	41398900	0.93	3.03	560.3	10.10	2850	30	1243
n50								
noise-C	223895	3.02	34.23	515.6	3.52	1976		393
delay-O, noise-C	223895	3.02	27.15	487.1	9.87	1723	6	425
integrated floorplan	226351	0.80	28.64	465.1	8.21	1756	32	685
n100								
noise-C	226295	3.21	65.90	489.1	4.19	1923		1856
delay-O, noise-C	226295	3.21	61.51	461.8	9.53	1708	8	1980
integrated floorplan	230851	4.51	66.70	431.8	9.02	1695	29	4665
n200								
noise-C	249987	0.97	86.67	491.2	4.57	2266		4896
delay-O, noise-C	249987	0.97	77.56	450.6	10.16	1915	9	9285
integrated floorplan	249987	0.97	77.56	450.6	10.16	1915	33	22497

TABLE IV
THE IMPACT OF ρ ON THE RESULTS OF N50

ρ	area	decap	average delay	% ave-dly reduce	max delay	itr	runtime
1.1	249987	29.33	485.2	10.23	1701	17	736
1.2	249987	27.15	487.1	9.87	1723	6	425
1.3	249987	30.67	492.2	8.93	1738	5	397
1.4	249987	34.82	498.6	7.75	1776	5	392

monitor the impact on delay. As expected, the smaller the bound is, the better the delay/decap quality is. However, this comes at the cost of longer runtime.

V. CONCLUSIONS

In this paper, we proposed a power-grid based RLC-model and its analytical formulas for gate and interconnect delay estimation. Our model considers the IR drop, the $L\mathrm{d}i/\mathrm{d}t$ noise, and the impact of decap. This model can calculate the delay of any two points in the floorplan, and the SPICE simulation shows it is accurate and practical. The objective of our decap planning formulation is to minimize the noise/decap-aware delay under noise, area, and leakage constraints. A sequential linear programming method is proposed to solve this non-linear optimization problem. Related experiments show that further delay reduction is possible without using more decaps.

REFERENCES

[1] S. Pant, D. Blaauw, V. Zolotov, and S. S. R. Panda, "Vectorless Analysis of Supply Noise Induced Delay Variation," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2003, pp. 184–191.

[2] S. Pant and D. Blaauw, "Timing-Aware Decoupling Capacitance Allocation in Power Distribution Networks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 527–541, 2007.

[3] C. Yeh and M. Sadowska, "Timing-Aware Power-Noise Reduction in Placement," in *Proc. Asia and South Pacific Design Automation Conf.*, 2007, pp. 757–762.

[4] S. Zhao, C. Koh, and K. Roy, "Decoupling Capacitance Allocation and Its Application to Power Supply Noise Aware Floorplanning," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 81–92, 2002.

[5] E. Wong, J. Minz, and S. K. Lim, "Decoupling Capacitor Planning and Sizing for Noise and Leakage Reduction," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2006, pp. 395–400.

[6] J. Fan, S. X.-D. Tan, Y. Cai, and X. Hong, "Partitioning-based Decoupling Capacitor Budgeting via Sequence of Linear Programming," in *Integration, the VLSI Journal*, 2007, pp. 516–524.

[7] 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 *Proc. ACM Design Automation Conf.*, 2006, pp. 217–222.

[8] 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 *Proc. Int. Symp. on Quality Electronic Design*, 2004, pp. 63–68.

[9] L. R. Zheng and H. Tenhunen, "Fast Modeling of Core Switching Noise on Distributed LRC Power Grid in ULSI Circuits," in *IEEE Trans. on Advanced Packaging*, 2001, pp. 245–254.

[10] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," in *IEEE Trans. on Circuits and Systems*, 1975, pp. 504–509.

[11] X. C. Li, J. F. Mao, and H. F. Huang, "Accurate analysis of interconnect trees with distributed RLC model and moment matching," in *IEEE Trans. on Microwave Theory and Techniques*, 2004, pp. 2199–2206.

[12] A. B. Kahng and S. Muddu, "An Analytical Delay Model for RLC Interconnects," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 1997, pp. 1507–1514.

[13] [Online]. Available: "http://mathworld.wolfram.com/LambertW-Function.html"

[14] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle packing based module placement," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1995, pp. 472–479.