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 optimizing 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.

Additional 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
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 network, 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 \(V_o\).

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 \(V_i\). \(D_{in}\) 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 \(\pi\)-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.

\( (G + sC)X = b \)

\[ G = \begin{bmatrix} A_2GA_T^T & A_1 & A_3 \\ A_T^T & 0 & 0 \\ A_T^T & 0 & 0 \end{bmatrix} \]

\[ C = \begin{bmatrix} A_2CA_T^T & 0 & 0 \\ 0 & -I & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

\[ X = [V_n \ I_1 \ I_2] \]

\[ b = [-A_1I_1 \ 0 \ V_o] \]

where \(V_n\) is the vector of all node voltages. The constitutive relations of the resistance, capacitance, and inductance branches are \(GV_g = I_p\), \(sC\) and \(sL\) 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:

\[ H(s) = \frac{b_0 + b_1s + b_3s^2}{1 + a_1s + a_2s^2}. \]

The moment expansion of \(H(s)\) is

\[ H(s) = b_0 + (b_1 - b_0a_1)s + (b_2 - b_0a_2 - b_1a_1 + b_0a_1^2)s^2 + \cdots. \]

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

\[ \begin{align*}
 b_0 &= m_0 \\
 b_1 &= a_1m_0 + m_1 \\
 b_2 &= a_2m_0 + a_1m_1 + m_2 \\
 a_1 &= \frac{m_2m_2 - m_1m_4}{m_1m_3 - m_2^2} \\
 a_2 &= \frac{m_2m_4 - m_3}{m_1m_3 - m_2^2}
\end{align*} \]

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

\[ v_c(t) = b_0 \left( -\frac{b_0 - b_1p_1 + b_1p_1^2}{a_2p_1(p_2 - p_1)}e^{-p_1t} - \frac{b_0 - b_1p_2 + b_1p_2^2}{a_2p_2(p_2 - p_1)}e^{-p_2t} \right). \]

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,

\[ v_c(t) = b_0 \left( -\frac{b_0 - b_1p_1 + b_1p_1^2}{a_2p_1(p_2 - p_1)}e^{-p_1t} \right). \]

\(^1\)These two paths for other scenarios of power-supply network can be generated similarly as in Figure 1, which is used for explaining easily.
Assuming the signal rises from 0 to 50% of \( V_{dd} \) during the elapsed time, the solution to the following equation is the delay of the circuit under analysis:

\[
b_0 = b_0 - b_1 p_1 + b_1 p_1^2 e^{-p_1 t} = \frac{1}{2} \]

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

\[
D = D_{in} + \frac{1}{p_1} \ln \left( b_0 - b_1 p_1 + b_1 p_1^2 \right) = \frac{1}{p_1} \ln \left( \frac{b_0 - b_1 p_1 + b_1 p_1^2}{p_1 (b_0 - 0.5)} \right). \tag{1}
\]

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

\[
v_c(t) = b_0 - e^{-p t} [M \sin(q t) + N \cos(q t)] / q
\]

where

\[
M = \frac{a_1 b_2 + 2a_2 b_0 - 2a_1 b_1}{2a_2}, \quad N = \frac{a_2 b_0 - b_2}{a_2}.
\]

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

\[
D = D_{in} + \frac{(b_0 - 0.5)e^{-\frac{a_1 t}{2a_2}} - 2a_2 \tan^{-1}(\frac{N}{M})}{\Delta} \tag{2}
\]

When \( a_1^2 - 4a_2 = 0 \), the response is

\[
v_c(t) = b_0 + P e^{-p t} - Q e^{-p t}
\]

where

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

Let \( v_c(t) = 50\% V_{dd} \), the total delay is

\[
D = D_{in} + P \frac{1 - \frac{2b_0}{a_1 Q} e^{-\frac{a_2 Q}{2a_2}}}{1 - \frac{2b_0}{a_1 Q} e^{-\frac{a_1 Q}{2a_2}}} \tag{3}
\]

\( w = LabmertW(x) \) [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 \( V_{n,noise} \), 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 \( Q^{(k)} = \int_0^{t_f} I^{(k)}(t) \cdot dt \) denote the maximum charge drawn from the power supply by block \( B^{(k)} \), where \( I^{(k)}(t) \) is the current demand, and \( t_f \) is the switching period. The decap demand is calculated as follows:

\[
\Theta^{(k)} = \max(1, \frac{V_{\text{noise}}^k}{V_{\text{tol}}}), \quad k = \{1, 2, \ldots, M\} \tag{4}
\]

\[
C_k = \frac{(1 - 1/\Theta^{(k)}) Q^{(k)}}{V_{\text{tol}}}, \quad k = \{1, 2, \ldots, M\} \tag{5}
\]

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

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/Decay-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-floorplanning 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:

\[
\text{NLP: Minimize } \sum_{i, j \in S_b} D_{ij}. \tag{6}
\]

Subject to

\[
\sum_{k \in S_w} x_{ik} \geq C_i / C_{\text{area}}, \quad i \in S_b \tag{7}
\]

\[
\sum_{i \in S_b} x_{ik} \leq A_k, \quad k \in S_w \tag{8}
\]

\[
\sum_{i \in S_b, j \in S_w} I_{ij} \leq I_{\text{total}} \tag{9}
\]

\[
x_{ik} = 0, \quad \omega_{sk} \text{ is not adjacent to } blk_i \tag{10}
\]

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. \( C_i / C_{\text{area}} \) 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. 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

$$\sum_{i \in S_b} \left[ \frac{C_i}{C_{ox}} - \sum_{k \in S_w} x_{ik} \right]$$

subject to

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

$$x_{ik} = 0, \ wk_k \text{ is not adjacent to } bl_k$$

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. Constraints (12) and (13) in this LP are identical to Equations (8) and (9).

2) Iterative Improvement Step: Let $C_{AI}^n$ and $C_{Bi}^n$ 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 $D_{ij}^n = f(C_{AI}^n, C_{Bi}^n)$. The SLP-based iterative improvement step proceeds as follows:

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

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.
**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 3nm 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. $V(b)$ 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]. 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.

![Fig. 4. Comparison of our model with SPICE simulation with other circuit parameters.](image)

![Fig. 5. SPICE simulation of the RLC network shown in Figure 2.](image)

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 allocates 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 $[C/p_ρC]$. We change $ρ$ and
TABLE III

<table>
<thead>
<tr>
<th>Circuit Algorithm</th>
<th>area</th>
<th>% area expand</th>
<th>decap</th>
<th>average delay</th>
<th>% ave-dly reduce</th>
<th>maximum delay</th>
<th>iteration</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>apl2</td>
<td>noise-C</td>
<td>54275600</td>
<td>11.87</td>
<td>3.09</td>
<td>1212</td>
<td>6.33</td>
<td>3962</td>
<td>17</td>
</tr>
<tr>
<td>delay-O, noise-C</td>
<td>54275600</td>
<td>11.87</td>
<td>2.98</td>
<td>1145</td>
<td>11.47</td>
<td>3821</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>integrated floorplan</td>
<td>57591300</td>
<td>10.37</td>
<td>3.38</td>
<td>1074</td>
<td>11.21</td>
<td>4015</td>
<td>32</td>
<td>37</td>
</tr>
<tr>
<td>xerox5</td>
<td>noise-C</td>
<td>21739700</td>
<td>5.82</td>
<td>1.23</td>
<td>419.4</td>
<td>3.29</td>
<td>1542</td>
<td>15</td>
</tr>
<tr>
<td>delay-O, noise-C</td>
<td>21739700</td>
<td>5.82</td>
<td>1.15</td>
<td>393.2</td>
<td>3.92</td>
<td>1479</td>
<td>7</td>
<td>23</td>
</tr>
<tr>
<td>integrated floorplan</td>
<td>22875200</td>
<td>8.56</td>
<td>1.34</td>
<td>369.5</td>
<td>9.97</td>
<td>1297</td>
<td>35</td>
<td>47</td>
</tr>
<tr>
<td>hp</td>
<td>noise-C</td>
<td>9762910</td>
<td>0.58</td>
<td>0.42</td>
<td>715.7</td>
<td>2.07</td>
<td>2942</td>
<td>19</td>
</tr>
<tr>
<td>delay-O, noise-C</td>
<td>9762910</td>
<td>0.58</td>
<td>0.39</td>
<td>679.6</td>
<td>7.01</td>
<td>2808</td>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>integrated floorplan</td>
<td>9762910</td>
<td>0.58</td>
<td>0.39</td>
<td>679.6</td>
<td>7.01</td>
<td>2808</td>
<td>17</td>
<td>32</td>
</tr>
<tr>
<td>ami53</td>
<td>noise-C</td>
<td>1278920</td>
<td>0.00</td>
<td>0.00</td>
<td>81.42</td>
<td>0.00</td>
<td>199.8</td>
<td>153</td>
</tr>
<tr>
<td>delay-O, noise-C</td>
<td>1278920</td>
<td>0.00</td>
<td>0.05</td>
<td>78.80</td>
<td>3.21</td>
<td>188.3</td>
<td>8</td>
<td>196</td>
</tr>
<tr>
<td>integrated floorplan</td>
<td>1302400</td>
<td>1.35</td>
<td>0.06</td>
<td>75.02</td>
<td>3.17</td>
<td>185.1</td>
<td>32</td>
<td>411</td>
</tr>
<tr>
<td>ami49</td>
<td>noise-C</td>
<td>40678100</td>
<td>0.23</td>
<td>2.91</td>
<td>635.5</td>
<td>5.02</td>
<td>3212</td>
<td>280</td>
</tr>
<tr>
<td>delay-O, noise-C</td>
<td>40678100</td>
<td>0.23</td>
<td>2.86</td>
<td>560.4</td>
<td>10.87</td>
<td>2987</td>
<td>9</td>
<td>526</td>
</tr>
<tr>
<td>integrated floorplan</td>
<td>41398900</td>
<td>0.93</td>
<td>3.03</td>
<td>560.3</td>
<td>10.10</td>
<td>2850</td>
<td>30</td>
<td>1243</td>
</tr>
<tr>
<td>n50</td>
<td>noise-C</td>
<td>223895</td>
<td>3.02</td>
<td>34.23</td>
<td>515.6</td>
<td>3.52</td>
<td>1976</td>
<td>393</td>
</tr>
<tr>
<td>delay-O, noise-C</td>
<td>223895</td>
<td>3.02</td>
<td>27.15</td>
<td>487.1</td>
<td>9.87</td>
<td>1723</td>
<td>6</td>
<td>425</td>
</tr>
<tr>
<td>integrated floorplan</td>
<td>226351</td>
<td>0.80</td>
<td>28.64</td>
<td>465.1</td>
<td>8.21</td>
<td>1756</td>
<td>32</td>
<td>685</td>
</tr>
<tr>
<td>n100</td>
<td>noise-C</td>
<td>226295</td>
<td>3.21</td>
<td>65.90</td>
<td>489.1</td>
<td>4.19</td>
<td>1923</td>
<td>1856</td>
</tr>
<tr>
<td>delay-O, noise-C</td>
<td>226295</td>
<td>3.21</td>
<td>61.51</td>
<td>461.8</td>
<td>9.53</td>
<td>1708</td>
<td>8</td>
<td>1980</td>
</tr>
<tr>
<td>integrated floorplan</td>
<td>230851</td>
<td>4.51</td>
<td>66.70</td>
<td>431.8</td>
<td>9.02</td>
<td>1695</td>
<td>29</td>
<td>4665</td>
</tr>
<tr>
<td>n30m</td>
<td>noise-C</td>
<td>249987</td>
<td>0.97</td>
<td>86.67</td>
<td>491.2</td>
<td>4.57</td>
<td>2266</td>
<td>4896</td>
</tr>
<tr>
<td>delay-O, noise-C</td>
<td>249987</td>
<td>0.97</td>
<td>77.56</td>
<td>450.6</td>
<td>10.16</td>
<td>1915</td>
<td>9</td>
<td>9285</td>
</tr>
<tr>
<td>integrated floorplan</td>
<td>249987</td>
<td>0.97</td>
<td>77.56</td>
<td>450.6</td>
<td>10.16</td>
<td>1915</td>
<td>33</td>
<td>22497</td>
</tr>
</tbody>
</table>

The impact of delay objective and noise constraints.

TABLE IV

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>area</th>
<th>decap</th>
<th>average delay</th>
<th>% ave-dly reduce</th>
<th>max delay</th>
<th>iter</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>249987</td>
<td>29.53</td>
<td>485.2</td>
<td>10.25</td>
<td>1701</td>
<td>17</td>
<td>736</td>
</tr>
<tr>
<td>1.2</td>
<td>249987</td>
<td>27.15</td>
<td>487.1</td>
<td>9.87</td>
<td>1723</td>
<td>6</td>
<td>425</td>
</tr>
<tr>
<td>1.3</td>
<td>249987</td>
<td>30.67</td>
<td>492.2</td>
<td>8.93</td>
<td>1758</td>
<td>5</td>
<td>397</td>
</tr>
<tr>
<td>1.4</td>
<td>249987</td>
<td>34.82</td>
<td>498.6</td>
<td>7.75</td>
<td>1776</td>
<td>5</td>
<td>392</td>
</tr>
</tbody>
</table>

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 $Ldi/dt$ 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 nonlinear optimization problem. Related experiments show that further delay reduction is possible without using more decaps.

REFERENCES