Floorplan and Power/Ground Network Co-Synthesis for Fast Design Convergence *

Chen-Wei Liu
Synopsys Taiwan Limited
Room 3108, 31F, 333, Section 1, Keelung Road,
Taipei 110, Taiwan
dustin@synopsys.com

Yao-Wen Chang
Graduate Institute of Electronics Engineering
and Department of Electrical Engineering
National Taiwan University
Taipei 106, Taiwan
ywchang@cc.ee.ntu.edu.tw

Abstract

As technology advances, the metal width decreases while the global wire length increases. This trend makes the resistance of the power wire increase substantially. Further, the threshold voltage scales nonlinearly, raising the ratio of the threshold voltage to the supply voltage and making the voltage (IR) drop in the power/ground (P/G) network a serious problem in modern IC design. Traditional P/G network analysis methods are often very computationally expensive, and it is thus not feasible to co-synthesize P/G network with floorplan. To make the co-synthesis feasible, we need not only an efficient, effective, and flexible floorplanning algorithm, but also a very efficient, yet sufficiently accurate P/G network analysis method. In this paper, we present a method for floorplan and P/G network co-synthesis based on an efficient P/G network analysis scheme and the B-tree floorplan representation. We integrate the co-synthesis into a commercial design flow to develop an effective power integrity (IR-drop) driven design methodology. Experimental results based on a real-world circuit design and the MCNC benchmarks show that our design methodology successfully fixes the IR-drop errors earlier at the floorplanning stage and thus enables the single-pass design convergence.*

Categories and Subject Descriptors: B.7.2 [Integrated Circuits]: Design Aids

General Terms: Algorithms, Experimentation, Performance

Keywords: Floorplanning, Simulated Annealing, Power Integrity, IR Drop, Power/Ground Analysis

1. INTRODUCTION

As technology advances, the metal width decreases while the global wire length increases. This trend makes the resistance of the power wire increase substantially. Further, the threshold voltage scales nonlinearly, raising the ratio of the threshold voltage to the supply voltage and making the voltage (IR) drop in the P/G network a serious challenge in modern IC design [13]. Due to the IR-drop, supply voltage in logic may not be an ideal reference. This effect may weaken the driving capability of logic gates, reduce circuit performance, slow down slew rate (and thus increase power consumption), and lower noise margin [19].

Figure 1(a) shows a chip floorplan of four modules and the P/G network. As shown in the figure, we refer to a pad feeding supply voltage into the chip as a *power pad*, the power line enclosing the floorplan as a *core ring*, a power line branching from a core ring into modules inside as a *power trunk*, an intersection of a vertical and a horizontal power lines a *P/G node*, and a pin in a module that absorbs current (connects to a core ring or a power trunk) as an *P/G pin*. To ensure correct and reliable logic operation, we shall minimize the IR drops from the power pad to the P/G pins in a P/G network. Figure 1(a) shows an instance of voltage drop in the power supply line, in which the voltage drops by almost 26% at the rightmost P/G pin. As [19] pointed out that 5% IR drop in supply voltage may slow down circuit performance by as much as 15% or more. Therefore, IR drop is a first-order effect and can no longer be ignored during the design process, and it is desired to consider the P/G network synthesis during early physical design (e.g., floorplanning) for reliable circuit operation.

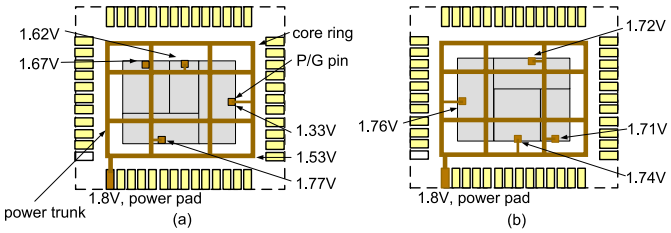


Figure 1: (a) An instance of floorplan and its P/G network structure. The worst-case voltage at the P/G pins is about 26% of the supply voltage. (b) A floorplan with smaller worst-case voltage drops. The worst-case voltage drop is about only 5%.

1.1 Previous Work

The problem of P/G network synthesis has been studied extensively in the literature. An important problem of P/G network synthesis is to use the minimum amount of wiring area for a P/G network under the power integrity constraints such as IR drops and electromigration. There are two major tasks for the synthesis: (1) P/G network topology determi-

*This work was partially supported by National Science Council of Taiwan under Grant No's. NSC 94-2215-E-002-030, NSC 94-2220-E-002-001, and NSC 94-2752-E-002-008-PAE.

ISPD'06, April 9–12, 2006, San Jose, California, USA.

nation to plan the wiring topology of a P/G network [2][15], etc. and (2) P/G wire sizing to meet the current density and reliability constraints [4] [17].

As the design complexity increases dramatically, it is necessary to handle the IR-drop problem earlier in the design cycle for better design convergence. Most existing commercial tools deal with the IR-drop problem at the post-layout stage when entire chip design is completed and detailed layout and current information are known. It is, however, often very difficult and computationally expensive to fix the P/G network synthesis at the post-layout stage. Therefore, researchers started to consider the P/G network analysis at an earlier design stage [6] [18] [19].

Dharchoudhury et al. proposed a design flow with different modes of power grid analysis incorporated between stages of the design flow [6]. The work shows that considering power integrity analysis at earlier stage can significantly improve design convergence. Yim, Bae, and Kyung in [19] presented an early floorplan-based P/G network planning methodology. Recently, Wu and Chang proposed a power integrity-driven design methodology of performing P/G network analysis after floorplanning [18].

It is very reasonable that [6], [18], and [19] can significantly improve design convergence. At the floorplanning stage, a prototype of the chip is determined in this stage, and the power consumption for each module and the positions for modules and P/G pins become available, making the P/G network analysis feasible at this stage. Furthermore, it is intrinsically more flexible to fix any power integrity problem at this stage than at the post-layout stage when most module positions and wiring are fixed. However, there is a significant difficulty in doing the early P/G network analysis: Traditional P/G network analysis methods are often very computationally expensive and are thus not feasible to be incorporated into the floorplanning design. To make the power integrity-driven design flow feasible, we need a very efficient, yet sufficiently accurate P/G network analysis method.

1.2 Our Contributions

In this paper, we present a method for floorplan and P/G network *co-synthesis* based on an efficient, yet sufficiently accurate P/G network analysis scheme for the mesh P/G structure and the efficient B*-tree floorplan representation [1]. We develop a P/G network aware method to reduce the floorplan solution space and thus speed up the co-synthesis, and then integrate the co-synthesis step into a commercial design flow to develop an effective power integrity (IR-drop) driven design flow. Experimental results based on a real-world circuit design and the MCNC benchmarks show that our design methodology successfully fixes the IR-drop errors earlier at the floorplanning stage and thus enables the single-pass design convergence. Different from the work [18], our method has the following advantages:

- What we propose here is an automatic floorplan and P/G network *co-synthesis* method, instead of simple P/G network analysis incorporated after floorplanning and a semi-automatic power integrity-driven design flow, as that proposed in [18].
- In contrast to the simple resistor tree handled in [18], we work on the mesh-based P/G network structure, which is the most popular structure in modern IC design.

The remainder of this paper is organized as follows. Section 2 formulates the floorplan and P/G network co-synthesis design problem. Section 3 describes our design flow. Section 4 presents our P/G network and floorplan co-synthesis algorithm. Section 5 reports the experimental results, and Section 6 concludes this paper.

2. PROBLEM FORMULATION

The problem of floorplan and P/G network co-synthesis is formulated as follows: Given a floorplan of m modules, the number of power pads for the whole chip and the power consumption for each module, the objective is to obtain a feasible floorplan and simultaneously generate a corresponding P/G network that satisfies the power constraints. Before presenting the power integrity constraints, we introduce the notations for describing a P/G network used in [18]: Let $G = \{N, B\}$ be a P/G network with n nodes $N = \{1, 2, \ldots, n\}$ and b branches $B = \{1, 2, \ldots, b\}$. Each branch i in B connects two nodes: i_1 and i_2 with current flowing from i_1 to i_2. Let l_i and w_i be the length and width of branch i, respectively. Let $r_\square$ be the sheet resistivity (unit Ω per square), and V_i (I_i) be the voltage (current) at node i. Then the resistance r_i of branch i is $r_i = (V_{i_1} - V_{i_2})/I_i = r_\square l_i/w_i$. At the early stage power analysis, we need a fast analysis for the P/G network. For this reason, a sophisticated model for the P/G network is often too time-consuming and thus infeasible for the co-synthesis. In this paper, we use the resistive model for P/G networks and the static current source model. We consider the power integrity constraints as follows:

- **The IR-drop constraints:**
  For every P/G pin i, its corresponding voltage V_i must satisfy the following constraints:

  $V_i \geq V_{min,k}$ for each power pin i of module k,
  $V_i \leq V_{max,k}$ for each ground pin i of module k,

  where $V_{min,k}(V_{max,k})$ is the minimum (maximum) voltage required at the injection point of a P/G network for module k.

- **The minimum width constraints:**
  The width of a P/G line must be greater than the minimum width allowed in the given technology. The constraint is given by

  $$w_i = \frac{r_\square l_i I_i}{V_{i_1} - V_{i_2}} \geq w_{i,min}, \quad (1)$$

  where $w_{i,min}$ is the given constraint.

- **The electromigration constraints:**

  $|V_{i_1} - V_{i_2}| \leq r_\square l_i \sigma$ (i.e., $I_i/w_i \leq \sigma$), for each $i \in B$

  where σ is a constant for a particular routing layer with a fixed thickness.

3. THE PROPOSED DESIGN FLOW

In this section, we describe our design flow, which is illustrated in Figure 2. The netlist is the circuit generated in high-level synthesis. We partition the circuit into hard modules (hard macros) and soft modules (groups of standard cells). The P/G network and floorplan co-synthesis generates a P/G network and a floorplan that satisfy all power integrity constraints.

With a feasible floorplan, we perform placement and routing which include detailed placement, P/G routing, clock

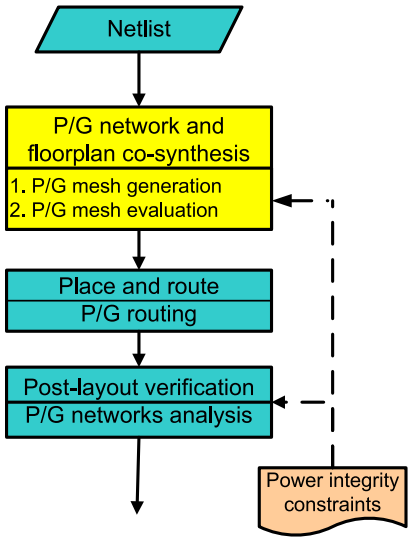


Figure 2: The proposed design flow.

tree synthesis, and detailed routing. Finally, the final P/G network is analyzed, and simulation is performed to check the correctness of the final design.

4. FLOORPLAN AND P/G NETWORK CO-SYNTHESIS

In this section, we present our floorplan and P/G network co-synthesis algorithm. Our floorplanning algorithm adopts the B*-tree floorplan representation [1] and uses simulated annealing (SA). We shall first review the B*-tree floorplan representation. Given an admissible placement [8], we can construct a unique B*-tree in linear time to model the placement. Figure 3(a) shows an admissible placement and its

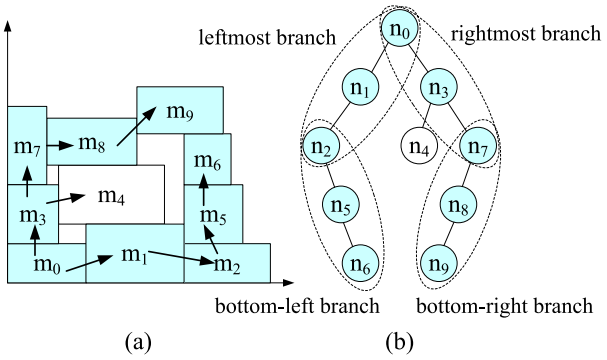


Figure 3: Boundary modules and their corresponding B*-tree branches.

corresponding B*-tree illustrated in Figure 3(b). A B*-tree is an ordered binary tree whose root corresponds to the module on the bottom-left corner. Similar to the depth-first search (DFS) procedure, we construct a B*-tree T for an admissible placement in a recursive fashion: Starting from the root, we first recursively construct the left subtree and then the right subtree. Let R_i denotes the set of modules located on the right-hand side and adjacent to m_i. The left child of the node n_i corresponds to the lowest module in R_i that is unvisited. The right child of n_i represents the lowest module located above and with its x-coordinate equal to that of m_i. Given a B*-tree, the x-coordinates of all modules can easily be determined by traversing the tree once [1], and we can apply a contour data structure [8] to compute the y-coordinates in amortized linear time. The computation for the coordinates of modules is also referred to as *packing*. The SA algorithm requires a cost function to guide the optimization. The cost function of the traditional SA-based floorplanning is given by

$$\Psi = \alpha W + \beta A, \quad 0 < \alpha, \beta < 1, \quad \alpha + \beta = 1, \tag{2}$$

where W is the wirelength, A is the area, and α and β are weighting parameters. To perform power integrity driven floorplanning, we add a penalty for violating the power integrity constraints and the P/G mesh density cost in the cost function. The cost function becomes

$$\begin{aligned} \Psi &= \alpha W + \beta A + \gamma \Phi + \omega \frac{A}{D_{pitch}^2}, \\ & 0 < \alpha, \beta, \gamma, \omega < 1, \quad \alpha + \beta + \gamma + \omega = 1, \end{aligned} \tag{3}$$

where Φ is the penalty function of power integrity violations and D_{pitch} is the pitch of the P/G mesh which will be discussed in later sections, and α, β, γ, and ω are weighting parameters. The term A/D_{pitch}^2 is the density cost of the P/G mesh which affects the routing resource. The cost function is calculated after packing a B*-tree to obtain a corresponding floorplan. To obtain the penalty function of power integrity violations, we first generate a P/G mesh for the floorplan and then evaluate the P/G mesh. In the following sections, we discuss the P/G mesh generation and the evaluation method.

4.1 P/G Mesh Generation

In order to evaluate the performance of the actual P/G network of a floorplan at the floorplanning stage, we generate a conceptual P/G network for the floorplan. We use the mesh structure for the P/G network, since it is widely used in modern VLSI chips to reduce the IR-drop effects. By specifying the pitch of the power lines, we can determine the dimension of the P/G mesh. A uniform mesh can then be generated easily by evenly distributing the power lines. Figure 4(a) shows a uniform mesh.

The pitch D_{pitch} of the P/G mesh is determined during the SA process and depends on the average value of the P/G network penalty function Φ. We will detail the determination of D_{pitch} in Section 4.6. The complexity of the P/G

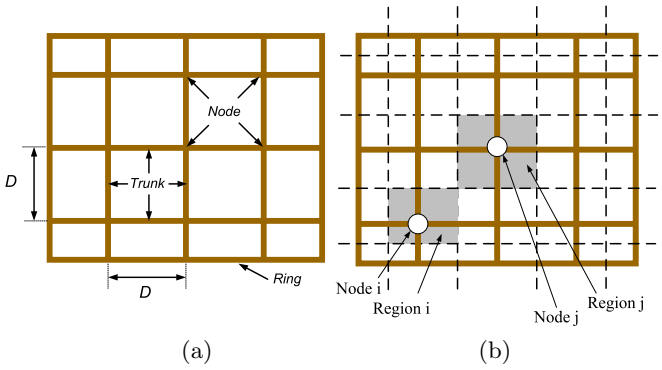


Figure 4: (a) A uniform P/G mesh. (b) A floorplan with a P/G mesh divided into regions.

mesh analysis mainly depends on the number of nodes of

the mesh. To reduce the complexity, we make a reasonable approximation by attaching all current sources to the intersection nodes of the vertical and horizontal power lines. That is, every P/G pin is connected to its nearest node with a power strap, and the length of the strap is the Manhattan distance between the P/G pin and the node. For convenience, we divide the floorplan into n regions, where n is the number of the nodes. The divided floorplan is illustrated in Figure 4(b). The border line of two regions is the center line between the two nodes such that the node is the nearest one for any point in the region.

4.2 Macro Current Source Modelling

In [9], it is shown that the result of static P/G analysis can be an upper bound for that of dynamic analysis by using the peak current. Therefore, we shall consider static analysis using constant current sources with the maximum current. Now we introduce how to estimate the maximum current consumption of hard modules and soft modules. For hard modules, we connect a P/G pin to the corresponding (center) node of the region where the pin is located, and the pin absorbs the estimated maximum current consumed by the pin, which is obtained by the pattern-based power simulation. At the floorplanning stage, we do not have the exact placement of the standard cells in the soft module. For soft modules, therefore, our current model is based on the worst-case scenario. We use the *maximum possible current function*, $I_{max}()$, to determine the current assigned to the nodes. The definition of $I_{max}(A_r, k)$, the maximum possible current in the specified region of the soft module k with size A_r, is as follows:

$$I_{max}(A_r, k) = \max_{S(A_r,k)} \left(\sum_{\forall i \in S_n} I_c(i) \right), \qquad (4)$$

where $S(A_r, k)$ is the set of sets of standard cells in the soft module k, such that for each set $S_n \in S(A_r, k)$, $\sum_{\forall i \in S_n} A_r(i) \leq A_r$ ($A_r(i)$ is the area of the standard cell i) and $I_c(i)$ is the maximum estimated current drawn by the cell i. The problem of solving $I_{max}()$ can be formulated as a 0-1 knapsack problem [5]: The area is the total weight that one can carry, the area of a cell is the weight of an item, and the current drawn by the cell is the value of the item. Our goal is to take as valuable a load as possible while the total weight of items does not exceed a given total weight constraint. Since the 0-1 knapsack problem is NP-complete [5], it is computationally expensive to solve the problem exactly. Therefore, we resort to an approximation by assuming that each standard cell can be divided freely. Then the maximum possible current can be approximated efficiently in linear time using the fractional knapsack algorithm [5]. As Figure 5 illustrates, for the soft module k overlapping with the region n, $I_{max}(A_{ov}(n, k), k)$ amount of current is assigned to the node n, where $A_{ov}(n, k)$ is the amount of the area k overlapping with n. Taking the node n as an example, its region (region n) contains two pins of the module A and three pins of the module B. Assume that the gray area is equal to the total area of 10 cells. Thus, there are 10 cells with from 30 mA to 21 mA current of the module k being attached to node n. Therefore, the current source attached to the node n consumes $0.3 \times 2 + 0.5 + (0.03 + 0.021) \times 10/2 = 1.355A$ current. Since the external voltage supply is typically connected to the ring, all voltage sources are assigned to the nodes on the ring. Then, the number of voltage supplies and the maximum current per supply node depend on the power budget of the design.

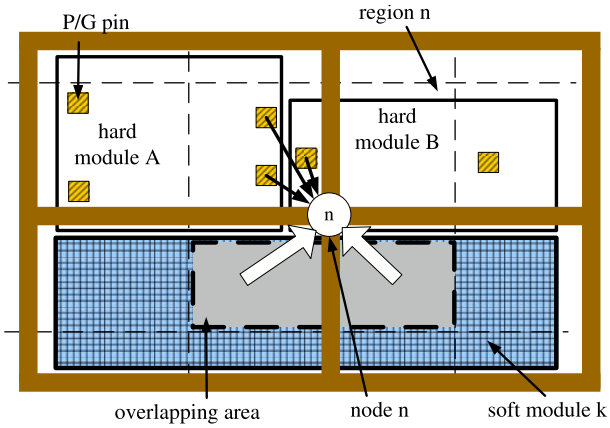


Figure 5: An example of the P/G analysis. The dashed lines denote the boundaries of the regions, and the gray area denotes the overlap of the soft module k and the region n. Each pin in the module A absorbs $0.3A$ current and each pin in the module B absorbs $0.5A$ current. The soft module k contains 30 standard cells of the same size. The largest current-consuming cell draws $30mA$ current, the second one draws $29mA$ current, and so on. Therefore the smallest cell draws $1mA$ current.

4.3 P/G Networks Analysis

After the P/G network is generated, we analyze the P/G mesh with the floorplan. Traditional analysis for a complete and accurate P/G network is very computationally expensive and unaffordable for integrating with floorplanning. Our objective for floorplan and P/G network co-synthesis is to derive an efficient scheme for the P/G network analysis based on the technology information available at the floorplanning stage. We apply the resistive P/G network model [14] and use the maximum current (see Section 4.2) drawn by the modules for static P/G network analysis. As the P/G mesh example shown in Figure 6, the chip is composed of four modules. The P/G wires are modelled as resistors. A P/G pin in a hard module is modelled as a current source. The static analysis of a P/G network is formulated

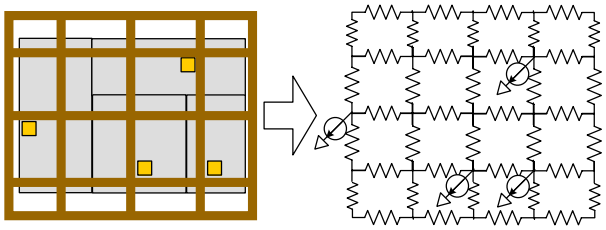


Figure 6: A global power mesh and its equivalent circuit model.

as follows [14]:

$$\mathbf{Gx} = \mathbf{i}, \qquad (5)$$

where $\mathbf{G}$ is the conductance matrix for the resistor, $\mathbf{x}$ is the vector of node voltages, and $\mathbf{i}$ is the vector of current loads. The dimensions of $\mathbf{i}$ and $\mathbf{x}$ are equal to the number of nodes in the P/G network, and $\mathbf{G}$ is a sparse positive definite matrix for a general resistor network.

We can solve Equation (5) efficiently by using an iterative method for the sparse matrix such as the preconditioned conjugated gradient method and/or other Krylov subspace

methods [7]. The time complexity of solving the equation is $O(n)$, where n is the number of the nodes in the mesh. As mentioned in the preceding section, we reduce the number of nodes by an approximation presented in the preceding subsection. Thus the number n is within a tractable range.

4.3.1 P/G Network Estimation

Once the voltage of each node is obtained, we can estimate the voltage at each P/G pin based on the voltage of the closest (connected) node and the distance of the P/G pin. For a hard module, the voltage of a P/G pin is estimated by the voltage of the closest node minus the largest possible voltage drop over the strap connecting the node and the pin. For a P/G pin j and its corresponding node i, the estimation is given by

$$V_j = V_i - I_j \max\left(r_{\square h}\frac{Dx_{ij}}{w_{hstrap}}, r_{\square v}\frac{Dy_{ij}}{w_{vstrap}}\right), \quad (6)$$

where $r_{\square v}$ and $r_{\square h}$ are the respective sheet resistivity of the vertical and horizontal metal layers, w_{hstrap} and w_{vstrap} are the widths of the respective vertical and horizontal straps, Dx_{ij} and Dy_{ij} are the respective vertical and horizontal distances between pin j and node i. For example, the left pin of the module B in Figure 5 is estimated by the voltage of the node n, which is 1.78 V. The current consumption of the pin is $0.5A$, the horizontal sheet resistivity is $5m\Omega/\square$, the vertical sheet resistivity is $4m\Omega/\square$, the respective vertical and horizontal distances from the pin to the node n are $5\mu m$ and $3\mu m$, and the width of a strap is $1\mu m$. The estimated voltage of the pin is $1.78 - 0.5 \times \max(0.005 \times \frac{5}{1}, 0.004 \times \frac{3}{1}) = 1.77V$. For a soft module, we use the distance between the center of the overlapping area and the node as the length of the strap. The voltage is estimated by the lowest supply voltage of the soft module k (a module may be attached to more than one node) as follows:

$$V_k = \min_{S_{ov}}\left(V_i - I_{k,i}\max\left(r_{\square h}\frac{Dx_{ik}}{w_{hstrap}}, r_{\square v}\frac{Dy_{ik}}{w_{vstrap}}\right)\right), \quad (7)$$

where S_{ov} is the set of nodes responsible for the soft module k, $I_{k,i}$ is the current supplied by node i (the estimated current in Section 4.1), and Dx_{ik} and Dy_{ik} are the respective horizontal and vertical distances between the node i and the center of the overlapped area. Again let us take the node n in Figure 5 as an example. The vertical and horizontal distances between the center of the gray area and the node n are $6\mu m$ and 0μ, respectively. The estimated voltage of the module k with respect to the node n is $1.78 - ((0.03 + 0.021) \times \frac{10}{2}) \times 0.004 \times \frac{6}{1} = 1.774V$. Assume that this is the lowest voltage among all the estimated voltages calculated from all regions overlapped with the module k. Thus, the estimated voltage of the module k is $1.774V$. Now we can verify the power integrity constraints (recall Section 2). The IR-drop constraints is verified by checking the IR drop of each P/G pin, and the electromigration constraints can be verified by checking the current flowing through every branch of the P/G mesh.

Now we can derive Φ, the penalty function of power integrity violations mentioned in Section 4. The function Φ is given as follows:

$$\Phi = \theta\frac{|B_{em}|}{|B|} + (1-\theta)\frac{\sum_{\forall pv_i \in P_v} v_{pv_i}}{\sum_{\forall p_i \in P} V_{lim,p_i}}, \quad 0 < \theta < 1, \quad (8)$$

where θ is a weighting parameter, B_{em} is the set of branches violating electromigration constraints, B is the total branches of the P/G mesh, v_{pv_i} is the amount of the violation at the pin pv_i, P is the set of all P/G pins, P_v is the set of violating P/G pins, and V_{lim,p_i} is the IR-drop constraint of the P/G pin p_i ($Vdd - V_{min,p_i}$ for a power pin and V_{max,p_i} for a ground pin). The first part of the right-hand side denotes the ratio of branches violating the electromigration constraints over total branches, and the second part denotes the ratio of the amount of IR-drop violation over the total amount of possible violations. The denominators are for the penalty normalization.

4.4 P/G Network Co-synthesis Heuristic

According to our experience, if the pitch is carefully chosen, the algorithm can find desired floorplans with very few constraint violations at high temperatures and continue to optimize wirelength and area at lower temperatures, leading to high-quality floorplan solutions. Note that IR drop and the current per branch decrease as the density of the mesh increases; therefore, we can reduce the P/G violation penalty Φ by increasing the density of the mesh. Since the density of a P/G mesh is proportional to A/D_{pitch}^2, we can control D_{pitch} instead of the density for convenience. By controlling D_{pitch} during the SA process, we can obtain desired floorplan solutions. We update the P/G mesh pitch D_{pitch} at each temperature by multiplying k_i, which is defined as follows:

$$k_i = \frac{\hat{\Phi}}{\Phi_{avg,i}}, \quad (9)$$

where $\Phi_{avg,i}$ is the average of Φ at the temperature of the ith iteration during the SA process, and $\hat{\Phi}$ is expected average of Φ, which a user-specified parameter. The floorplans generated at the same temperature form a solution sub-space. Specifying $\hat{\Phi}$, we can control the average Φ of the solution sub-space and statistically control the proportion of the feasible solutions in the solution sub-space.

4.5 Feasible B*-trees with Power Mesh Constraints

In this subsection, we study the properties of the B*-tree with the P/G network considerations and develop techniques to reduce the solution space to speed up the search for desired floorplans. Finding the best positions of modules to optimize the P/G mesh is a very complex problem. Our idea is motivated by the linear circuit theory: the IR drop of a P/G pin is proportional to the effective resistance between the P/G pin and the power pad. Therefore, the closer the P/G pin is placed to the power pad, the smaller IR drop we can get. Based on this fact, we can place the modules which consume larger current near the boundary of the floorplan, and then place power pads close to them. To implement this idea, we sort the modules by their power consumption and cluster the leading modules, which are called *power-hungry modules* to form groups. In our implementation, we choose 10% of total modules to be power-hungry modules. The size of a group depends on the total size of the member modules, which is a user specified parameter. Note that each group should contain at least one module. We refer to these groups as *power-hungry groups*. Each power-hungry group is assigned with a power pad and the number of the groups equals the number of available power pads. In order to reduce the IR drops of power-hungry groups, we prefer to place the modules in the power-hungry groups along the boundary of the floorplan And we will place each pad next to a power-hungry group.

We have two goals for the floorplan and power/ground network co-synthesis: (1) place power-hungry groups along

the chip boundary, and (2) maintain all the power-hungry modules in power-hungry groups, which can be accomplished by careful perturbations and will be discussed later. For the first goal, we should identify the boundary modules of the floorplan. Now we explore the feasibility conditions of the B*-tree to search for desired floorplan solutions. Let the *boundary ring* Υ_F (Υ_T) of the floorplan F (the B*-tree T) be the ordered list of the boundary modules in F (T) (say, in the counter-clockwise sequence starting from the module at the bottom-left corner). For example, $\Upsilon_F =< m_0, m_1, m_2, m_5, m_6, m_9, m_8, m_7, m_3 >$ ($\Upsilon_T =< n_0, n_1, n_2, n_5, n_6, n_9, n_8, n_7, n_3 >$) in the floorplan F (the B*-tree T) of Figure 3. Notice that by the name "ring", we can consider the succeeding element of the last element in the "list" to be the first element of the list. For the example of Figure 3, m_0 (n_0) is the succeeding element of m_3 (n_3). We shall make all modules of the power groups belong to the modules in the boundary ring such that the modules of the same power group are placed in the order according to the boundary "ring."

Extending the findings in [12] by Lin et al., we can identify the modules in the boundary ring based on the feasibility conditions of B*-trees for boundary modules. Let the root of the B*-tree T be r, the DFS order of the tree traversal on the leftmost and the bottom-left branches of T be L_T, and the DFS order of the tree traversal on the rightmost and the bottom-right branches of T be R_T. Let the reverse of a sequence L be L^r. Then, we have $\Upsilon_T = L_T \oplus R_T^r$. Here, "$\oplus$" denotes the concatenation operation of two lists.

THEOREM 1. *(Boundary Ring)* $\Upsilon_T = L_T \oplus R_T^r$.

According to Theorem 1, we shall make the nodes corresponding to the modules of a power-hungry group in the boundary ring Υ_T. In other words, we prefers to make those nodes a sublist of the ring Υ_T during the perturbation in simulated annealing. As shown in the example of Figure 7, the power group $\{m0, m1, m3\}$ ($\{m6, m8, m9\}$) is placed on the left and the bottom (the right and the top) boundaries close to the bottom-left (top-right) corner, and they are adjacent modules in the ring Υ_F. We say a floorplan to be *power-feasible* if the power-hungry modules in each power-hungry group are modules in the desired locations of the boundary ring. Therefore, it is desirable to keep a power-feasible floorplan during solution perturbation to achieve the second goal of the co-synthesis. While perturbing the tree, we should maintain the power-feasibility of the B*-tree. The operations to perturb a B*-tree [1] with the IR-drop consideration are listed as follows:

- Op1: Rotate a module.
- Op2: Swap two modules in the power-hungry groups or not in any power-hungry group.
- Op3: Move a module to another place that maintains power-feasibility.

Op1 only exchanges the width and height of a module without changing the B*-tree topology while Op2 and Op3 do. Therefore, in order to maintain the power-feasibility, we shall only swap two modules in power-hungry groups or not in any power-hungry group for Op2, and move a module to another place that maintains power-feasibility for Op3. Otherwise, we might need to transform the B*-tree to maintain the power-feasibility.

4.6 The Co-Synthesis Algorithm

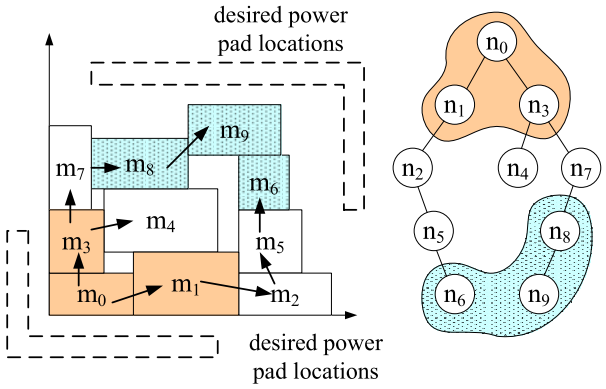


Figure 7: An example of a power-feasible floorplan with two power groups: $\{m6, m8, m9\}$ and $\{m0, m1, m3\}$. The desired power pad locations are encircled by the dashed lines.

Figure 8 summarizes our floorplaning algorithm. Given inputs of the module information, initial P/G pitch D_{pitch}, and power integrity constraints, we start with the simulated annealing process (see lines 2–24). At the beginning of simulated annealing, we randomly explore the solution space to get an average cost to normalize each objective in the cost function (line 3). Then we get an initial solution and an initial temperature (lines 4–6) and launch the simulated annealing process. At each temperature, we anneal for N times, where N is a number proportional to the number of modules (line 8). After each perturbation (line 9), we compute the coordinates of all modules and construct a P/G mesh (lines 10–11). Then we calculate the voltage of each node of the mesh by solving Equation (5) using our linear solver and estimate the IR drop of each P/G pin by Equations (6) and (7) (lines 12–13). Then we calculate the P/G mesh penalty function Φ and accumulate it for the average bookkeeping (line 14). Next we update the cost function by Equation (3) and check if the floorplan is accepted with the probability $e^{\frac{-\Delta\Psi}{T}}$ (lines 15–20). If the current floorplan S has a lower cost than the best floorplan S_{best} found so far, S is chosen as the best floorplan (line 20). Next, we calculate $\Phi_{avg,i}$ and k_i, and then update the mesh pitch D_{pitch} by $k_i D_{pitch}$ to co-synthesize the P/G mesh (lines 21–22). At the end of the SA loop, we decrease the temperature T by multiplying a constant r (line 23).

5. EXPERIMENTAL RESULTS

The proposed algorithm was implemented in the C++ language on a Sun Blade 2000 workstation with one 1 GHz CPU and 8 GB RAM. It was built on the public B*-tree distribution available at [16]. We developed the linear solver using the reformulated modified nodal analysis (MNA) [10] and the conjugate gradient (CG) method with incomplete LU (ILU) pre-conditioner [3].

We conducted two experiments based on two sets of benchmarks: One is a real design, and another set of benchmarks are modified MCNC benchmark circuits. We did not compare with [18] because the resistor *tree* model used in their analyzer incurred very large errors with the mesh structure P/G networks. Thus, it could not generate a feasible solution. (Note that the work is intended for the tree-based P/G network analysis.)

In the first experiment, we implemented a real design—the public *OpenRISC1200* available from [11]. We chose the

Algorithm: Power Integrity Aware Floorplanning
1 Read initial settings: module information, initial pitch D_{pitch}, power integrity constraints, and power consumption data;
2 **do**
3 Get an average cost to normalize the cost;
4 Get an initial power-feasible floorplan S;
5 $S_{best} \leftarrow S$;
6 Get a temperature $T > 0$;
7 Start with the simulated annealing process;
8 **for** 1 **to** N
9 Perturb the floorplan and maintain power feasibility;
10 Pack the floorplan;
11 Construct a P/G mesh;
12 Calculate the voltage of each node of the mesh;
13 Estimate the IR drop of each P/G pin;
14 Calculate and accumulate Φ;
15 Calculate Ψ;
16 $\triangle\Psi \leftarrow \Psi(S') - \Psi(S)$;
17 **if** $\triangle\Psi \leq 0$ **then** $S \leftarrow S'$;
18 **else if** $\triangle\Psi > 0$ **then**
19 $S \leftarrow S'$ with probability $e^{\frac{-\triangle\Psi}{T}}$;
20 **if** $\Psi(S) < \Psi(S_{best})$ **then** $S_{best} \leftarrow S$;
21 Calculate $\Phi_{avg,i}$ and k_i;
22 $D_{pitch} \leftarrow k_i D_{pitch}$;
23 $T \leftarrow rT$;
24 **while** not converged or not cooled down;
25 **return** S_{best};

Figure 8: The P/G network and floorplan co-synthesis algorithm.

UMC 0.18 μm process technology and the Artisan 0.18 μm cell library, and used Synopsys's Design Compiler and Artisan's Memory Generator to synthesize the netlist. For the UMC 0.18 μm technology, the maximum allowable IR drop is 10% of the supply voltage. We used the worst-case supply voltage, which is 1.62 V. Thus, the IR-drop constraints are $V_{min} = 1.62$ for the power and $V_{max} = 0.162$ for the ground. We used metal5 and metal6 for the P/G networks. The resistivity is 0.095 for metal5 and 0.055 for metal6. The width of the metal wire is 30 μm for the P/G networks and 0.24 μm for the straps. We used the conceptual P/G mesh as a guideline for the real P/G mesh to ensure that the resulting floorplan fit into the real P/G mesh. The initial vertical and horizontal power wire pitches are both 700 μm.

We compared the performance of the following three design methodologies:

Methodology A: The Synopsys design flow using Astro autofloorplan and Astro placer with the plain option.

Methodology B: The Synopsys design flow using Astro autofloorplan and Astro placer with the plain and IR-drop-driven placement option.

Methodology C: Our proposed design flow.

In methodologies A and B, we used Astro autofloorplan to replace our co-synthesis floorplanner. After placement, we routed the P/G networks and ran AstroRail to check the feasibility of the P/G networks. If there is any violation, the floorplanner will adjust the design until the P/G networks pass the check.

Tables 1 and 2 list the comparisons. Table 1 gives the comparisons of the resulting die areas, wirelength, average delays, utilization of the cell area (the total cell area divided by the die area), and the maximum IR drops. The maximum IR drop was reported by AstroRail. As shown in the table, our design methodology C can improve the die area by 15.9% and the maximum IR drop by 41.8% with comparable wirelength and average delay, compared to the design methodology B. In particular, as shown in Table 2, our methodology required only one iteration to get the reported results while Methodologies A and B needed several iterations. In Table 2, the CPU time is given by the summation of the runtimes of all design stages for all iterations. As mentioned earlier, we fixed the module overlapping problem of Astro autofloorplan by moving the hard modules manually because Astro autofloorplan generated a similar floorplan every time. Note that we did not count the time for manual adjusting for fair comparison.

It should also be noted that our floorplanner can obtain a much better die area than Astro autofloorplan because the Astro autofloorplan cannot legalize hard macros automatically. We had to remove the overlaps manually. Since most of the floorplans generated by Astro autofloorplan cannot fit into the outline, we need to enlarge the chip to accommodate the hard macros. In contrast, the B*-tree-based floorplanners does not have the legalization problem because it performs packing to pack modules one by one. The voltage-drop maps of Methodologies B and C are shown in Figures 9(a) and (b), respectively. As shown in the figures, there are significantly large *red* regions—denoting IR-drop violations—in Figure 9(a) (Methodology B) while Methodology C solves those violations (see Figure 9(b)). Detailed routing was also performed after the resulting floorplan passed the AstroRail analysis to complete the whole design process.

OpenRISC1200	A*	B*	C	C vs. B
Die Area (mm^2)	3.86	3.86	3.33	15.9%
Wirelength (μm)	1655463	1539125	1540172	-0.1%
Avg. Delay (ns)	8.62	8.54	8.55	-0.1%
Utilization (%)	62	62	72	13.9%
Max IR-drop(mv)	80.18	78.20	55.14	41.8%

Table 1: Comparisons of the results of the different methodologies. Note that A and B are not fully automatic because Astro autofloorplan cannot legalize the overlapping modules.

CPU Runtime	A	B	C	C vs. B
Total (sec)	505	346	135	2.56X
Floorplanning (sec)	132	85	42	2.02X
Placement (sec)	208	143	48	2.98X
AstroRail (sec)	165	118	45	2.62X
Iterations	4	3	1	-

Table 2: Comparison of the CPU runtime and design iteration of the different methodologies.

The second experiment was tested on five MCNC benchmark circuits implemented with the TSMC 0.25 μm technology. We used metal3 and metal4 for the P/G networks. The resistivity of the two metal layers is 0.075 Ω per square. The IR-drop constraints are $V_{min} = 2.25$ and $V_{max} = 0.25$, and the maximum allowable IR drop is 250 mV. We gave each circuit two power pads and randomly assigned the peak current on each P/G pin of the modules. The initial vertical and horizontal power wire pitches are both 600 μm. We compared three floorplanners: (1) the plain public B*-tree floorplanner, (2) our co-synthesis floorplanner with the power-feasibility consideration for solution space reduction presented in Section 4.5, and (3) our co-synthesis floor-

Circuit	Plain B*-tree floorplanner					Ours with solution space reduction					Ours without solution space reduction				
	WL (m)	A (mm^2)	M.I.D (mV)	# Vio.	T (s)	WL (m)	A (mm^2)	M.I.D (mV)	# Vio.	T (s)	WL (m)	A (mm^2)	M.I.D (mV)	# Vio.	T (s)
apte	435.5	48.21	290.0	6	1.1	452.1	48.8	244.2	0	43.2	440.4	49.8	243.5	0	165.2
xerox	387.6	20.42	1667.8	39	3.3	410.2	22.4	237.5	0	47.3	401.5	21.3	242.2	0	122.3
hp	155.5	9.56	1855.3	38	3.2	189.5	11.7	241.3	0	24.0	187.1	11.2	243.6	0	58.2
ami33	58.4	1.31	818.7	99	8.8	73.2	1.2	249.9	0	20.2	69.0	1.4	249.9	0	43.4
ami49	864.6	39.86	1867.4	195	42.2	779.9	44.2	249.6	0	450.0	832.8	39.8	249.9	0	1412.0
comp.	0.98	0.97	5.28	377	0.03	0.99	1.04	0.99	0	0.32	1	1	1	0	1

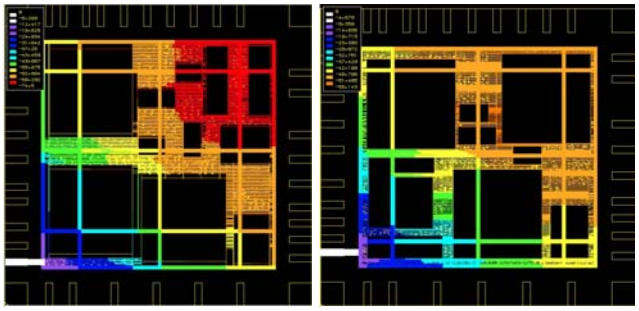
Table 3: Comparison of the original B*-tree floorplanner and our co-synthesis floorplanner with and without the power-feasibility consideration for solution space reduction.

planner without the power-feasibility consideration for solution space reduction. Both (2) and (3) considered power-integrity constraints while (1) did not.

The results are listed in Table 3, where "WL" denotes the wirelength, "A" stands for area, "M.I.D" gives the maximum IR drop, "#Vio." gives the number of IR-drop violations, and "T" gives the runtime. The values in the row "comparison" gives the normalized averages with respect to the results of the floorplanner (3). Note that it is reasonable that our floorplanner consumed much longer CPU time because our floorplanner performed also the P/G network analysis. As shown in the table, our floorplanners (2) and (3) can fix all the IR-drop violations and still keep reasonable wirelength and area, and the floorplanner with the power-feasibility considerations for solution space reduction can speed up the search by about 3X on average, revealing the effectiveness of the power-feasibility considerations for solution space reduction. The overall experimental results show that our floorplan and P/G network co-synthesis methodology is effective for power-integrity optimization for fast design convergence.

6. CONCLUSION

We have presented an effective floorplan and power integrity co-synthesis flow for faster design convergence. Experimental results have shown that our design methodology is very effective in power integrity aware design.



(a) (b)

Figure 9: (a) The voltage drop map of Methodology B. (b) The voltage drop map of Methodology C.

7. REFERENCES

[1] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, and S.-W. Wu. B*-trees: A new representation for non-slicing floorplans. In *Proc. of DAC*, pages 458–463, 2000.

[2] H. Chen, C.-K. Cheng, A. B. Kahng, M. Mori, and Q. Wang. Optimal planning for mesh-based power distribution. In *Proc. of ASP-DAC*, pages 444–449. IEEE Press, 2004.

[3] T.-H. Chen and C. C.-P. Chen. Efficient large-scale power grid analysis based on preconditioned krylov-subspace iterative methods. In *Proc. of DAC*, pages 559–562. ACM Press, 2001.

[4] S. Chowdhury. Optimum design of reliable ic power networks having general graph topologies. In *Proc. of DAC*, pages 787–790. ACM Press, 1989.

[5] T. T. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[6] A. Dharchoudhury, R. Panda, D. Blaauw, R. Vaidyanathan, B. Tutuianu, and D. Bearden. Design and analysis of power distribution networks in PowerPC microprocessors. In *Proc. of DAC*, pages 738–743. ACM Press, 1998.

[7] G. H. Golub and V. L. Charles F. *Matrix Computations*. Johns Hopkins University Press, 1996.

[8] P.-N. Guo, C.-K. Cheng, and T. Yoshimura. An O-tree representation of non-slicing floorplan and its applications. In *Proc. of DAC*, pages 268–273. ACM Press, 1999.

[9] D. Kouroussis and F. N. Najm. A static pattern-independent technique for power grid voltage integrity verification. In *Proc. of DAC*, pages 99–104. ACM Press, 2003.

[10] J. N. Kozhaya, S. R. Nassif, and F. N. Najm. Multigrid-like technique for power grid analysis. In *Proc. of ICCAD*, pages 480–487. IEEE Press, 2001.

[11] OpenRISC project, http://www.opencores.org/.

[12] J.-M. Lin, H.-E. Yi, and Y.-W. Chang, Module placement with boundary constraints using B*-trees, in IEE Proceedings–Circuits, Devices and Systems, Vol. 149, No. 4, pp. 251–256, August 2002. (EI/SCI)

[13] S. Lin and N. Chang. Challenges in power-ground integrity. In *Proc. of ICCD*, pages 651–654. IEEE Press, 2001.

[14] V. Litovski and M. Zwolinski. *VLSI Circuit Simulation and Optimization*. Chapman & Hall, 1997.

[15] J. Singh and S. S. Sapatnekar. Topology optimization of structured power/ground networks. In *Proc. of ISPD*, pages 116–123. ACM Press, 2004.

[16] Source code, http://cc.ee.ntu.edu.tw/∼ywchang/research.html.

[17] K. Wang and M. Marek-Sadowska. On-chip power supply network optimization using multigrid-based technique. In *Proc. of DAC*, pages 113–118. ACM Press, 2003.

[18] S.-W. Wu and Y.-W. Chang. Efficient power/ground network analysis for power integrity-driven design methodology. In *Proc. of DAC*, pages 177–180. ACM Press, 2004.

[19] J.-S. Yim, S.-O. Bae, and C.-M. Kyung. A floorplan-based planning methodology for power and clock distribution in ASICs. In *Proc. of DAC*, pages 766–771. ACM Press, 1999.