Power/Ground Network and Floorplan Cosynthesis for Fast Design Convergence

Chen-Wei Liu, *Student Member, IEEE*, and Yao-Wen Chang, *Member, IEEE*

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. Furthermore, 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 cosynthesize P/G network with floorplan. To make the cosynthesis 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 cosynthesis based on an efficient P/G network-analysis scheme and the B^*-tree floorplan representation. We integrate the cosynthesis 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.

Index Terms—Electromigration, floorplanning, IR drop, physical design, power/ground (P/G) analysis, power integrity, simulated annealing (SA).

I. 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. Furthermore, 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 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 [23].

Fig. 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

C.-W. Liu is with Synopsys Taiwan Ltd., Taipei 110, Taiwan, R.O.C.

Y.-W. Chang is with the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: ywchang@cc.ee.ntu.edu.tw).
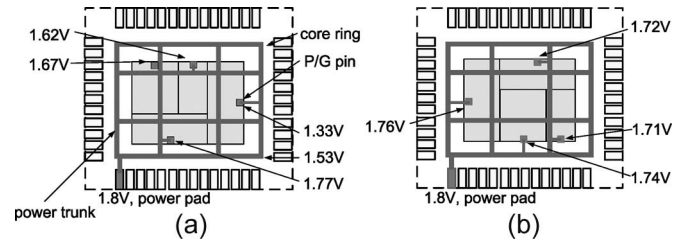
Fig. 1. (a) Instance of floorplan and its P/G network structure. The worst case voltage at the P/G pins is about 26% of the supply voltage. (b) Floorplan with smaller worst case voltage drops. The worst case voltage drop is about only 5%.

from a core ring into modules inside as a power trunk, an intersection of a vertical and a horizontal power lines as 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. Fig. 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. It was pointed out in [23] that 5% IR drop in supply voltage may slow down circuit performance by as much as 15% or more. Furthermore, it is typical to limit the voltage drop within 10% of the supply voltage to guarantee proper circuit operation [6]. 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.

A. 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 determination to plan the wiring topology of a P/G network [2], [15], [17]–[19], etc., and 2) P/G wire sizing to meet the current density and reliability constraints [4], [21].

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 postlayout stage when the entire chip design is completed and detailed layout and current information are known [see Fig. 2(a) for the traditional design flow]. It is, however, often very difficult and computationally expensive to fix the P/G network synthesis at the postlayout

Fig. 2. (a) Traditional design flow. (b) Design flow proposed in [22].

TABLE I
COMPARISON OF THIS PAPER AND THE DAC-2004 WORK

	Our work	DAC-2004 [25]
Design flow	P/G network co-synthesis with floorplan	P/G network analysis performed after floorplanning
P/G structure	Mesh structure	Ring structure
Network model	Resistor network	Resistor tree

based, different from the mesh structure that is generally used in modern P/G network design. 2) The P/G network is not cosynthesized with floorplan. Instead, if the postfloorplanning P/G network analysis fails the power-integrity constraints, a manual step of refloorplanning occurs. The semiautomatic process inevitably increases the design time and possibly increase time-consuming human trial-and-error cycles.

B. Our Contributions

In this paper, we present a method for floorplan and P/G network cosynthesis (i.e., optimize both the floorplan design and power integrity) 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 to speed up the cosynthesis and, then, integrate the cosynthesis step into a commercial design flow to develop an effective power-integrity (IR drop)-driven design flow. Experimental results based on real-world circuit designs 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 in [22], our method has the following advantages. 1) What we propose here is an automatic floorplan and P/G network cosynthesis method, which optimizes both floorplan design and power integrity, instead of simple P/G network analysis incorporated after floorplanning and a semi-automatic power-integrity-driven design flow, as that proposed in [22]. 2) In contrast to the simple resistor tree handled in [22], we work on the mesh-based P/G network structure, which is most popular in modern IC design.

See Table I for the comparison between our design flow and that proposed in the DAC-2004 work [22].

The remainder of this paper is organized as follows. Section II formulates the floorplan and P/G network-cosynthesis-design problem. Section III describes our design flow. Section IV presents our power network and floorplan cosynthesis algorithm. Section V describes the detailed implementation of the design flow. Section VI reports the experimental results, and finally, Section VII gives the concluding remarks and some future work.

stage. Therefore, researchers started to consider the P/G network analysis at an earlier design stage [6], [22], [23].

Dharchoudhury *et al.* proposed a design flow with different modes of power-grid analysis incorporated between stages of the design flow [6]. This paper shows that considering power-integrity analysis at an earlier stage can significantly improve design convergence. Yim *et al.* in [23] 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 [22] [see Fig. 2(b) for their design flow]. They developed a P/G network-analysis method based on a resistor tree model for power-integrity checking to be applied after the floorplanning stage. If the P/G network fails the check, a new floorplan is generated. The iteration continues until the P/G network passes the power-integrity checking. Their results show that the proposed flow leads to fewer iterations than the traditional flow for design convergence.

It is very reasonable that [6], [22], and [23] 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 postlayout 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.

To derive an efficient P/G network-analysis method, Wu and Chang in [22] enabled their design flow by working on a simple resistor-tree model to make their postfloorplanning P/G network analysis faster. However, their work has the following deficiencies/drawbacks: 1) Their P/G network structure is tree-

II. PROBLEM FORMULATION

The problem of floorplan and P/G network cosynthesis is formulated as follows: Given a floorplan of m modules, pre-placed 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 [22]: Let $G = \{N, B\}$ be a P/G network with n nodes $N = \{1, 2, \ldots, n\}$ and k branches $B = \{1, 2, \ldots, k\}$. Each branch b 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 b, respectively. Let r_{sq} be the sheet resistivity (unit Ω per square) and $V_i(I_i)$ be the voltage (current) at node i (branch b). Then, the resistance r_i of branch b is $r_i = (V_{i_1} - V_{i_2})/I_i = r_{\text{sq}}l_i/w_i$.

At the early stage of power analysis, we need a fast analysis for the P/G network. For this reason, a sophisticated model (for example, state-of-the-art P/G network simulation techniques [3], [16]) for the P/G network is often too time-consuming and, thus, infeasible for the cosynthesis. 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.

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

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

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

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.

2) **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

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

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

3) **The electromigration constraints**:

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

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

III. PROPOSED DESIGN FLOW

In this section, we describe our design flow, which is illustrated in Fig. 3. 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 cosynthesis 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 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. The implementation of our design flow will be discussed in more detail in Section V. It should be noted that we work on the uniform mesh structure for the P/G routing, and we only generate a conceptual P/G network at
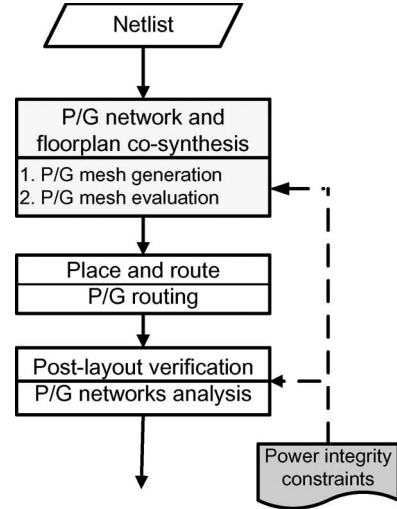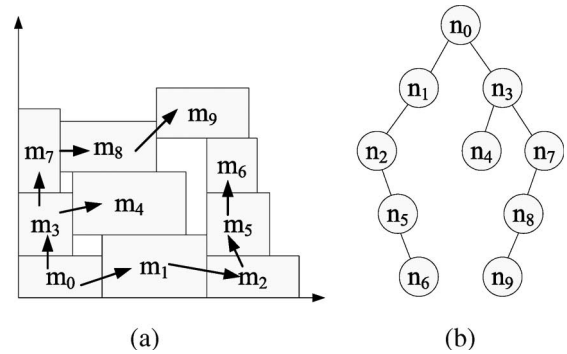

Fig. 3. Proposed design flow.

Fig. 4. (a) Admissible placement. (b) B^*-tree representing the placement.

the floorplanning stage for the P/G network analysis, while the actual P/G network is constructed during the placement and routing stage.

IV. FLOORPLAN AND P/G NETWORK COSYNTHESIS

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

Fig. 4(a) and (b) shows an admissible placement and its corresponding B^*-tree, respectively. 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 be easily determined by traversing the tree once [1], and we can apply a contour structure

[8] to compute the y-coordinates in amortized linear time. The computation for the coordinates of modules is also referred to as packing.

Our cosynthesis algorithm is based on SA. Each step of the SA algorithm perturbs the current solution by a random neighboring solution, chosen with a probability that depends on the difference between the values of a cost function and on a global parameter T (called the temperature); the temperature is gradually decreased during the annealing process. The dependence is that the current solution changes almost randomly when T is large, but increasingly "downhill" (greedy) as T approaches zero. The mechanism for "uphill" moves saves SA from being stuck at local minima, which is the major drawback of a greedy method.

The cost function of the traditional SA-based floorplanning is given by

$$\Psi = \alpha W + \beta A, \qquad 0 < \alpha; \quad \beta < 1; \quad \alpha + \beta = 1 \quad (2)$$

where W is the wirelength, A is the area, and α and β are the weighting parameters. The cost is evaluated after each solution perturbation.

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

$$\Psi = \alpha W + \beta A + \gamma \Phi + \omega \frac{A}{D_{\text{pitch}}^2},$$
$$0 < \alpha, \beta, \gamma, \omega < 1; \qquad \alpha + \beta + \gamma + \omega = 1 \quad (3)$$

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 the weighting parameters. The term A/D_{pitch}^2 represents the number of grids of the P/G mesh, which is used as the density cost of the P/G mesh. This term can balance the tradeoff between P/G network performance and routing resource of P/G networks.

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. Section IV-A describes how a P/G mesh is generated during the SA process. Section IV-B presents a technique for modeling macrocurrent sources. Section IV-C describes our P/G network-analysis method, and Section IV-C1 describes how to estimate the IR drop on each pin. Section IV-D presents a heuristic to tune the P/G network during SA. Section IV-E provides a technique for speeding up the SA process. Section IV-F summaries the cosynthesis algorithm.

A. 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
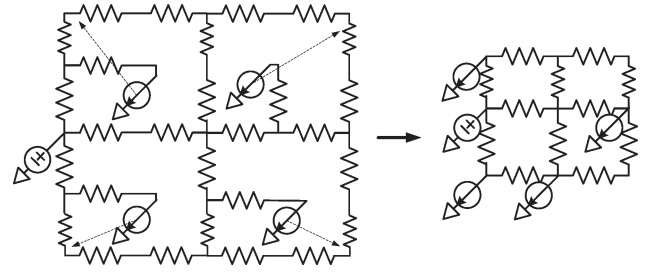

Fig. 5. Model-size reduction by connecting each current source to its nearest node.
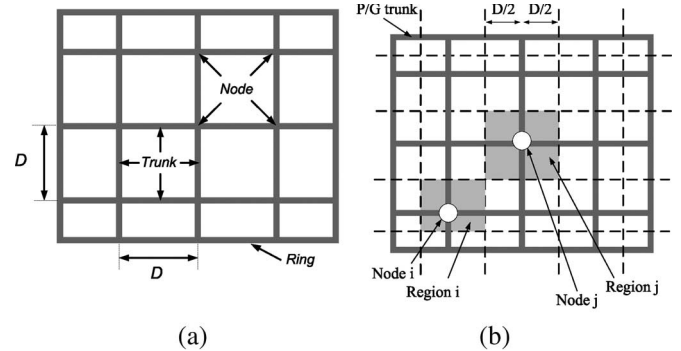

Fig. 6. (a) Uniform P/G mesh. (b) Floorplan with a P/G mesh divided into regions.

structure for the P/G network, since it is widely used in modern very large-scale integrated 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. Fig. 6(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 IV-F.

The complexity of the P/G 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 (see Fig. 5 for an illustration). For convenience, we divide the floorplan into n regions, where n is the number of the nodes. The divided floorplan is illustrated in Fig. 6(b). The borderline of the two regions is the centerline between the two nodes, such that the node is the nearest one for any point in the region.

B. Macrocurrent-Source Modeling

In [9], it is shown that the result of static P/G analysis can be an upper bound for that of the 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
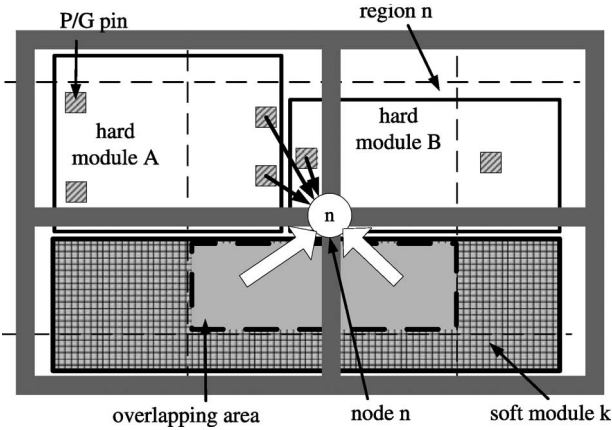
Fig. 7. Example of the P/G analysis. The dashed lines denote the boundaries of the regions and the gray area denotes the overlap of the soft module k and the region n. Each pin in the module A absorbs 0.3-A current and each pin in the module B absorbs 0.5-A current. The soft module k contains 30 standard cells of the same size. The largest current-consuming cell draws 30-mA current, the second one draws 29-mA current, and so on. Therefore, the smallest cell draws 1-mA current.

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 all possible combinations of standard cells in the soft module k, such that for each combination $S_n \in S(A_r, k)$, $\sum_{\forall i \in S_n} A_r(i) \le A_r$ ($A_r(i)$ 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 Fig. 7 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 one pin of the module B. Assume that the gray area is equal to the total area of ten cells. Thus, there are ten cells with
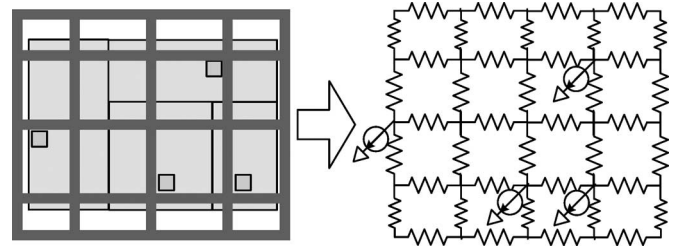

Fig. 8. Global power mesh and its equivalent circuit model.

from 30 to 21 mA current of the module k being attached to node n. Therefore, the current source attached to the node n consumes $0.3 \times 2 + 0.5 + (0.03 + 0.021) \times 10/2 = 1.355$ A 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.

C. 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 the floorplan and P/G network cosynthesis 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 IV-B) drawn by the modules for static P/G network analysis. As the P/G mesh example shown in Fig. 8, the chip is composed of four modules. The P/G wires are modeled as resistors. A P/G pin in a hard module is modeled as a current source.

The static analysis of a P/G network is formulated as follows [14]:

$$\mathbf{G}\mathbf{x} = \mathbf{i} \qquad (5)$$

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 symmetric sparse positive-definite matrix for a general resistor network.

We can solve (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 section. Thus, the number n is within a tractable range (see Fig. 5).

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

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

where r_{sqv} and r_{sqh} 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. Here, we assume the voltages on the nearby global trunks are the same. Because the IR drop of a global trunk is induced by currents from many pins, we can neglect the effect of a single pin.

For example, the left pin of the module B in Fig. 7 is estimated by the voltage of the node n, which is 1.78 V. The current consumption of the pin is 0.5 A, the horizontal sheet resistivity is 5 mΩ/sq, the vertical sheet resistivity is 4 mΩ/sq, the respective vertical and horizontal distances from the pin to the node n are 5 and 3 μm, and the width of a strap is 1 μm. The estimated voltage of the pin is $1.78 - 0.5 \times \max(0.005 \times (5/1), 0.004 \times (3/1)) = 1.77$ V. 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:

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

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 IV-A), 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 Fig. 7 as an example. The vertical and horizontal distances between the center of the gray area and the node n are 6 and 0 μm, respectively. The estimated voltage of the module k with respect to the node n is $1.78 - ((0.03 + 0.021) \times (10/2)) \times 0.004 \times (6/1) = 1.774$ V. 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 1.774 V. Now, we can verify the power-integrity constraints (recall Section II). 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 IV. The function Φ is given as follows:

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

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 ($\text{Vdd} - V_{\text{min}, p_i}$ 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 the 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.

D. P/G Network Cosynthesis Heuristic

For a constant P/G mesh pitch, we observed that the cosynthesis algorithm often fails to converge, because very few floorplans in the solution space satisfy all constraints with the pitch. Therefore, we shall develop a method to adjust the pitch for better design convergence. 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 wire length and area at lower temperatures, leading to high-quality floorplan solutions. Note that the 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 the desired floorplan solutions. We update the P/G mesh pitch D_{pitch} at each temperature by multiplying k_i, which is defined as follows:

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

where $\Phi_{\text{avg}, i}$ is the average of Φ at the temperature of the ith iteration during the SA process and $\hat{\Phi}$ is the expected average of Φ, which is a user-specified parameter. The floorplans generated at the same temperature form a solution subspace. Fig. 9 shows an example of pitch updating during SA process. If D_{pitch} becomes too small (the P/G mesh density becomes too large), the Φ_{avg} will become smaller than $\hat{\Phi}$, making k larger than one and pushing D_{pitch} back to a normal size. We can treat $\hat{\Phi}$ as a factor for controlling the P/G network density. A larger $\hat{\Phi}$ results in a sparser P/G network, while a smaller $\hat{\Phi}$ results in a denser one.

It is clear that a denser power mesh will incur routing congestion and decrease routability. Therefore, we add A/D_{pitch}^2 into the cost function to control the power mesh density. This mechanism can prevent the power mesh density from growing too large and, thus, increasing the routing congestion.

E. Feasible B^*-trees With Power Mesh Constraints

In this section, 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.
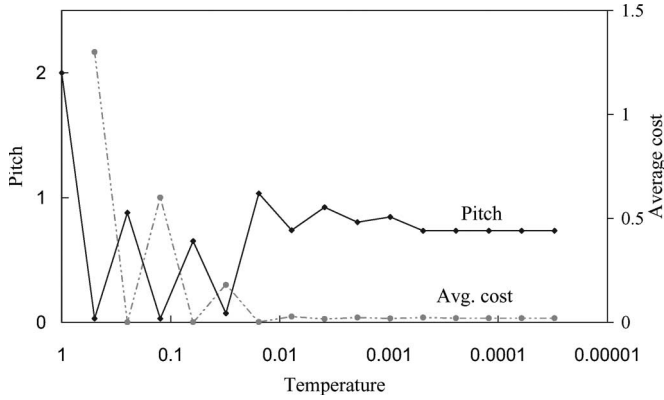
Fig. 9. Example of pitch updating during the SA process. Pitch and $\hat{\Phi}$ are initialized to 2 and 0.02, respectively.

Fig. 10. Boundary modules and their corresponding B^*-tree branches.

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, and power pads are inserted by shifting the preplaced pads. 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 P/G network cosynthesis: 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 $\Upsilon_F (\Upsilon_T)$ 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 Fig. 10. 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 Fig. 10, $m_0(n_0)$ 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
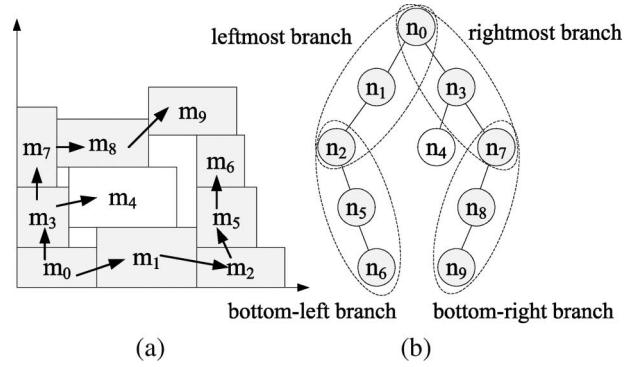
the same power group are placed in the order according to the boundary "ring."

To explore the B^*-tree nodes corresponding to the boundary-ring modules of a floorplan, we shall first identify the tree nodes associated with boundary modules. Let the leftmost branch (rightmost branch) of a B^*-tree denote the path formed by the root and its leftmost (rightmost) descendants. For example, nodes n_0, n_1, and n_2 (n_0, n_3, and n_7) in the B^*-tree of Fig. 10 form the leftmost (rightmost) branch. Let the bottom-left branch (bottom-right branch) of a B^*-tree denote the path formed by the end of the leftmost (rightmost) branch and its rightmost (leftmost) descendants. For example, nodes n_2, n_5, and n_6 (n_7, n_8, and n_9) in the B^*-tree of Fig. 10 form the bottom-left (-right) branch. Extending the findings in [12] by Lin *et al.*, we can identify the modules in the boundary ring based on the four feasibility conditions of B^*-trees for the boundary modules listed below.

Property 1—Boundary Properties [12]:
1) Bottom-boundary condition: The nodes corresponding to the bottom-boundary modules must be in the leftmost branch of a B^*-tree.
2) Left-boundary condition: The nodes corresponding to the left-boundary modules must be in the rightmost branch of a B^*-tree.
3) Right-boundary condition: For the right-boundary modules, their corresponding nodes are in the bottom-left branch of a B^*-tree with the left child for each node in the path being deleted.
4) Top-boundary condition: For the top-boundary modules, their corresponding nodes are in the bottom-right branch of a B^*-tree with the right child for each node in the path being deleted.

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 $\Upsilon_T = L_T \oplus R_T^r$: To consider the IR-drop optimization during floorplanning, as mentioned earlier, we prefer to place modules of a power-hungry group along the chip boundary where the power pads can be assigned. According to Theorem 1, we shall make the nodes
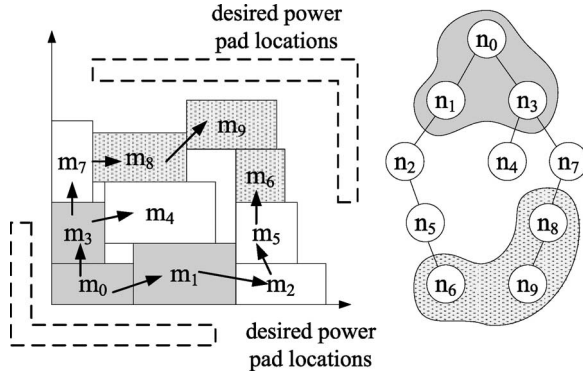
Fig. 11. Example of a power-feasible floorplan with two power groups: $\{m6, m8, m9\}$ and $\{m0, m1, m3\}$. The desired power-pad locations are encircled by the dashed lines.

corresponding to the modules of a power-hungry group in the boundary ring Υ_T. In other words, we prefer to make those nodes a sublist of the ring Υ_T during the perturbation in SA. As shown in the example of Fig. 11, 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 cosynthesis.

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 the following operations.

1) Op1: Rotate a module.
2) Op2: Swap two modules in the power-hungry groups or not in any power-hungry group.
3) 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. For Op3, we move a node n_i to another place. After Op3, if n_i is a member of the power group P and its new position is feasible, we delete n_i and randomly insert n_i to a position that satisfies the boundary property and the adjacent property of P; otherwise, no transformation is needed. Fig. 12 shows an example of the transformation for Op3.

F. Cosynthesis Algorithm

Fig. 13 summarizes our floorplaning algorithm. Given the inputs of the module information, initial P/G pitch D_{pitch}, and power-integrity constraints, we start with the SA process (see lines 2–24). At the beginning of the SA, we randomly explore the solution space to get an average cost to normalize each
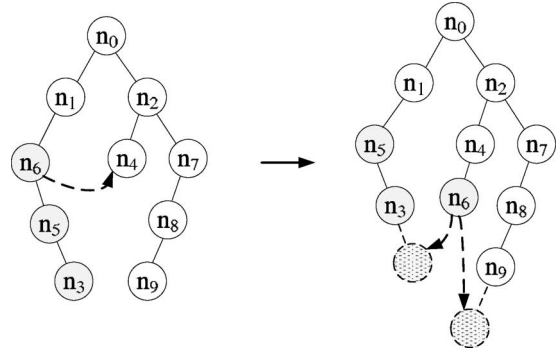

Fig. 12. Example of transformation for Op3. The power group in the figure consists of $n3$, $n5$, and $n6$. The left figure is the B^*-tree before Op3 (move n_6 to n_4's left child). The dotted nodes denote the potential positions for insertion.

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};

Fig. 13. P/G network and floorplan cosynthesis algorithm.

objective in the cost function (line 3). Then, we get an initial solution and an initial temperature (lines 4–6) and launch the SA 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 (5) using our linear solver and estimate the IR drop of each P/G pin by (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 (3) and check if the floorplan is accepted with the probability $e^{-\Delta\Psi/T}$ (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 $\Phi_{\text{avg},i}$ and
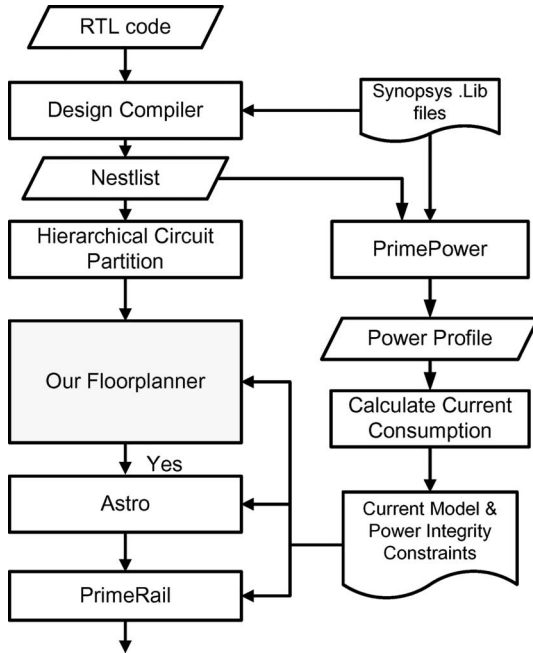
Fig. 14.   Our design flow is integrated into Synopsys's ASIC design flow.

k_i and, then, update the mesh pitch D_{pitch} by $k_i D_{\text{pitch}}$ to cosynthesize 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).

V. OVERALL DESIGN FLOW

In this section, we describe the overall P/G network and floorplan cosynthesis flow. The detailed design flow is depicted in Fig. 14. Given an RTL code, we use Synopsys's Design Compiler to generate a netlist using a standard-cell library and a memory generator. After the netlist is generated, we obtain the power profile using Synopsys's PrimePower. The current consumption of each cell or macro is determined by the peak current of all the simulation frames. By using the original hierarchy information, hard macros such as memory modules are taken as hard modules, and the remaining netlist is partitioned into soft modules. We assigned extra deadspace to each module to increase the routability. The macroinformation and the power profile is fed into the cosynthesis floorplanner to generate a feasible floorplan. According to the floorplan, we used the hard-region-group mode to place the standard cells in soft modules during the placement stage. After the placement stage is completed, we perform P/G network routing using Astro's preroute function. PrimeRail, a cell-based and transistor-level P/G simulation tool by Synopsys, is used to check the feasibility of the P/G networks. PrimeRail provides fast and reliable P/G network analysis after the P/G network is routed. Then, we perform clock synthesis and detailed routing to complete the design.

VI. 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 [20]. We developed the linear solver using the reformulated modified nodal analysis (MNA) [10] and the conjugate-gradient method with incomplete LU preconditioner [3].

We conducted three experiments based on the three sets of benchmarks. Two are real designs and one set of benchmarks are modified MCNC benchmark circuits. We did not compare with [22] 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 this paper is intended for the tree-based P/G network analysis).

In the first two experiments, we implemented two real designs—public OpenRISC1200 and picoJavaII available from [11] and from SUN.

A. OpenRISC1200—32-b RISC

For OpenRISC1200, we chose the 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 $V_{\text{min}} = 1.62$ V for the power and $V_{\text{max}} = 0.162$ V 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 (default placement without any additional option).

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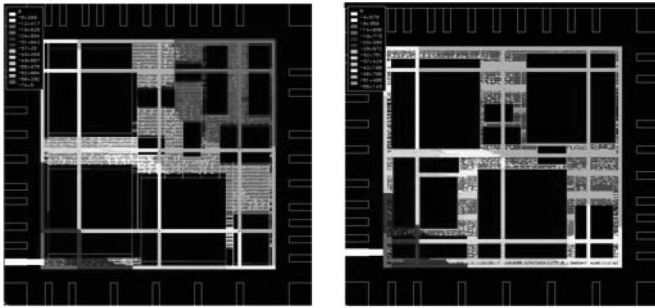
Methodology C)   Our proposed design flow.

In methodologies A) and B), we used Astro autofloorplan to replace our cosynthesis 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.

Table II list the comparisons. Table II 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 wire length and average delay, compared to the design methodology B). In particular, as shown

TABLE II
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

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%
Total CPU (sec)	505	346	135	2.61X
Floorplanning (sec)	132	85	42	2.02X
Placement (sec)	208	143	48	2.98X
PrimeRail (sec)	164	112	38	2.95X
Iterations	4	3	1	-

TABLE III
COMPARISONS OF THE FIRST-ITERATION RESULTS OF ASTRO AND OUR
DESIGN FLOW. SINCE 0.13-μm TECHNOLOGY IS MORE SENSITIVE
TO IR-DROP AND ELECTROMAGNETIC EFFECTS, THERE
ARE MANY VIOLATIONS IN THE ASTRO FLOW AND
ARE VERY DIFFICULT TO FIX MANUALLY

picoJavaII	Astro	Ours	comparison
Die Area (mm^2)	2.12	2.018	5.05%
Wirelength (μm)	2158205	2145652	0.59%
Avg. Delay (ns)	3.24	3.28	-1.22%
Max IR-drop(mv)	215.71	89.20	141.8%
Violations	439	0	439
Mesh dimension	24x24	12x9	5.33 x
CPU (sec)	4605	4653	-1.03%

(a)   (b)

Fig. 15.   (a) Voltage-drop map of methodology B). (b) Voltage-drop map of methodology C).

in Table II, our methodology required only one iteration to get the reported results while methodologies A) and B) needed several iterations. 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 everytime. 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 Fig. 15(a) and (b), respectively. As shown in the figures, there are significantly large red regions—denoting IR-drop violations—in Fig. 15(a) [methodology B)] while methodology C) solves those violations [see Fig. 15(b)]. Detailed routing was also performed after the resulting floorplan passed the AstroRail analysis to complete the whole design process.

B. picoJavaII—Java Chip

For picoJavaII, we chose the more advanced TSMC 0.13-μm process technology and the Artisan 0.13-μm cell library. The

supply voltage is 1.3 V, and the IR-drop limitation is 0.13 V in this process technology. Since, in this technology, the design is more sensitive to power-integrity constraints, we used a denser initial P/G grid. The pitch was initialized as 60 μm, and the wire width was given by 4 μm. We compared the first-iteration results of Astro and our design flow (see Table III). Within one iteration, our design flow significantly reduces the maximum IR drop, solves all the violations, and synthesizes a coarser P/G grid.

C. MCNC Benchmarks

The third 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 cosynthesis floorplanner with the power-feasibility consideration for solution-space reduction presented in Section IV-E; and 3) our cosynthesis floorplanner 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 IV. 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 3× 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 cosynthesis methodology is effective for power-integrity optimization for fast design convergence.

VII. CONCLUDING REMARKS

We have presented an effective floorplan and power-integrity cosynthesis flow for faster design convergence. Experimental

TABLE IV
COMPARISON OF THE ORIGINAL B^*-TREE FLOORPLANNER AND OUR COSYNTHESIS FLOORPLANNER WITH AND WITHOUT THE
POWER-FEASIBILITY CONSIDERATION FOR SOLUTION-SPACE REDUCTION, WHERE "WL" DENOTES THE WIRE LENGTH,
"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)
OURS WITHOUT SOLUTION SPACE REDUCTION

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

results have shown that our design methodology is more efficient on real designs compared with commercial design flow.

In this paper, we have focused our discussions on static P/G analysis. For future more advanced design, the dynamic behavior of the P/G network is an important issue. However, it is much more time-consuming to do dynamic analysis than static analysis. It is desirable to develop an efficient yet sufficiently accurate dynamic P/G analysis scheme to facilitate future P/G network and floorplan cosynthesis. Our algorithm can be extended to support dynamic P/G analysis for future more advanced designs. We can model the decaps as modules and cosynthesize them with floorplan to optimize the silicon area. Our model can be applied to the dynamic MNA for the dynamic P/G analysis, which is shown in the following equation:

$$Gx(t) + C\dot{x}(t) = i(t) \qquad (10)$$

where C is the capacitance matrix. Using the forward or backward Euler method, a preconditioned iterative linear solver can solve the equation efficiently.

Another research direction lies in the design of P/G network structures. In this paper, our P/G network structure is based on uniform meshes. Some advanced industrial designs are using hierarchical P/G networks, which use nonuniform P/G structures for different P/G subnetworks. To consider nonuniform P/G structures, we can precalculate possible P/G networks for modules and build a macrolookup table, as Singh and Sapatnekar did in [17]–[19]. Then, we can floorplan the corresponding modules and the P/G subnetworks at the same time. For each floorplan candidate, we assemble the global networks by P/G subnetworks and apply the macromodels directly from the lookup table to estimate the performance of the global network. This would also give a potential hierarchical cosynthesis approach.

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. ACM Des. Autom. Conf.*, 2000, pp. 458–463.

[2] H. Chen, C.-K. Cheng, A. B. Kahng, M. Mori, and Q. Wang, "Optimal planning for mesh-based power distribution," in *Proc. Asia and South Pac. Des. Autom. Conf.*, 2004, pp. 444–449.

[3] T.-H. Chen and C. C.-P. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," in *Proc. ACM Des. Autom. Conf.*, 2001, pp. 559–562.

[4] S. Chowdhury, "Optimum design of reliable IC power networks having general graph topologies," in *Proc. ACM Des. Autom. Conf.*, 1989, pp. 787–790.

[5] T. T. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: 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. ACM Des. Autom. Conf.*, 1998, pp. 738–743.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins Univ. 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. ACM Des. Autom. Conf.*, 1999, pp. 268–273.

[9] D. Kouroussis and F. N. Najm, "A static pattern-independent technique for power grid voltage integrity verification," in *Proc. ACM Des. Autom. Conf.*, 2003, pp. 99–104.

[10] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "Multigrid-like technique for power grid analysis," in *Proc. ICCAD*, 2001, pp. 480–487.

[11] *OpenRISC Project*. [Online]. Available: http://www.opencores.org/

[12] J.-M. Lin, H.-E. Yi, and Y.-W. Chang, "Module placement with boundary constraints using B*-trees," *Proc. Inst. Electr. Eng.—Circuits, Devices Syst.*, vol. 149, no. 4, pp. 251–256, Aug. 2002.

[13] S. Lin and N. Chang, "Challenges in power-ground integrity," in *Proc. ICCD*, 2001, pp. 651–654.

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

[15] T. Mitsuhashi and E. S. Kuh, "Power and ground network topology optimization for cell based VLSIs," in *Proc. ACM Des. Autom. Conf.*, 1992, pp. 524–529.

[16] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Power grid analysis using random walks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 8, pp. 1204–1224, Aug. 2005.

[17] J. Singh and S. S. Sapatnekar, "Topology optimization of structured power/ground networks," in *Proc. ACM Int. Symp. Phys. Des.*, 2004, pp. 116–123.

[18] ——, "Partition-based algorithm for power grid design using locality," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 4, pp. 664–677, Apr. 2005.

[19] ——, "Congestion-aware topology optimization of structured power/ ground networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 5, pp. 683–695, May 2005.

[20] *Source Code*. [Online]. Available: http://cc.ee.ntu.edu.tw/ywchang/ research.html

[21] K. Wang and M. Marek-Sadowska, "On-chip power supply network optimization using multigrid-based technique," in *Proc. ACM Des. Autom. Conf.*, 2003, pp. 113–118.

[22] S.-W. Wu and Y.-W. Chang, "Efficient power/ground network analysis for power integrity-driven design methodology," in *Proc. ACM Des. Autom. Conf.*, 2004, pp. 177–180.

[23] J.-S. Yim, S.-O. Bae, and C.-M. Kyung, "A floorplan-based planning methodology for power and clock distribution in ASICs," in *Proc. ACM Des. Autom. Conf.*, 1999, pp. 766–771.

Chen-Wei Liu (S'05) received the B.S. degree in electrical engineering and the M.S degree in electronics engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 2003 and 2005, respectively.

He is currently with Synopsys Taiwan Ltd., Taipei. His current research interests include computer-aided design and circuit simulation and analysis.

Yao-Wen Chang (S'94–M'96) received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1988, and the M.S. and Ph.D. degrees from the University of Texas at Austin in 1993 and 1996, respectively, all in computer science.

He is a Professor of the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University. He is currently also a Visiting Professor at Waseda University, Kitakyushu, Japan. He was with IBM T. J. Watson Research Center, Yorktown Heights, NY, in summer 1994. From 1996 to 2001, he was on the faculty of National Chiao Tung University, Hsinchu, Taiwan. His current research interests lie in VLSI physical design, design for manufacturing, and FPGA. He has been working closely with industry on projects in these areas.

Dr. Chang received an award at the 2006 ACM ISPD Placement Contest, Best Paper Award at ICCD-1995, and eight Best Paper Nominations from DAC-2007, ISPD-2007, DAC-2005, 2004 ACM TODAES, ASP-DAC-2003, ICCAD-2002, ICCD-2001, and DAC-2000. He has received many awards for research performance, such as the 2005 and 2006 First-Class Principal Investigator Awards and the 2004 Mr. Wu Ta You Memorial Award from the National Science Council of Taiwan, the 2004 MXIC Young Chair Professorship from the MXIC Corporation, and for excellent teaching from National Taiwan University and National Chiao Tung University. He is an Editor of the *Journal of Computer and Information Science*. He currently serves on the ACM/SIGDA Physical Design Technical Committee and the technical program committees of a few important conferences on VLSI design automation, including ASP-DAC (topic chair), DAC, DATE, FPT, GLSVLSI, ICCAD, ICCD, ISPD, SOCC, and VLSI-DAT. He is currently the Chair of the Design Automation and Test (DAT) Consortium of the Ministry of Education, Taiwan, a member of the Board of Governors of the Taiwan IC Design Society, and a member of the IEEE Circuits and Systems Society, ACM, and ACM/SIGDA.