A Methodology for the Simultaneous Design of Supply and Signal Networks

Haihua Su, *Member, IEEE*, Jiang Hu, *Member, IEEE*, Sachin S. Sapatnekar, *Fellow, IEEE*, and Sani R. Nassif, *Senior Member, IEEE*

***Abstract*—We present an early-stage global wire-design methodology that simultaneously considers the performance needs for both signal lines and power grids under congestion considerations. An iterative procedure is employed in which the global routing is performed according to a congestion map that includes the resource utilization of the power grid, followed by a step in which the power grid is adjusted to relax the congestion in crowded regions. This adjustment is in the form of wire removal in noncritical regions, followed by a wire-sizing step that overcomes the voltage noise after wire removal and a wire-width resizing that meets the maximum current-density constraint. Experimental results show that the overall routability can be significantly improved while the power-grid noise is maintained within both the voltage-drop and current-density constraints.**

***Index Terms*—Circuit optimization, design automation, power distribution, routing, sensitivity.**

I. Introduction

THE ROLE of interconnect has become increasingly critical in nanometer design and the need to meet stringent performance constraints has resulted in strong contention for scant routing resources. A major consumer of these resources is the power-distribution network, which must be designed to ensure reliable $V_{\rm dd}$ and ground levels and, therefore, requires the use of dense grids. On the other hand, global wires also compete for the same routing resources, as they often require shortest-path routes to meet their own performance requirements. Traditionally, these two have been designed independently, with the routing needs for a regular power grid being determined first, after which the remaining resources are calculated to provide routing resource budgets for the signal nets.

As the number and criticality of global signal wires becomes more dominant, such a methodology becomes unsustainable as the initial budgets may often be entirely unreasonable, so that *ad hoc* changes at the end are inadequate to meet the performance and/or routing-completion targets. Moreover, the use of a completely regular supply grid cannot be defended in the face of large variations in the voltage drops on a regular grid at different points of the chip. Therefore, in nanometer design, there is a strong need for a unified approach to the design of signal wires and power grids, with an integrated approach to routing resource management.

The considerations involved in the design of the power grid and the signal wires are quite different. Since the former is intended to provide a reliable $V_{\rm dd}$ or ground voltage throughout the circuit, a typical grid contains a dense and regular distributed mesh-like structure that is designed to meet constraints on parameters such as the *IR* drop noise. Since this entails the use of numerous wide wires, the routing resources used by the power grid are considerable. On the other hand, signal nets are typically tree structures that are designed to connect a source to a number of sinks under constraints related to the timing criticality of the net. The complexity of routing signal nets is exacerbated by their sheer number, so that the routes for any individual net cannot be determined without considering the contributions of the others to the congestion. Most often, global signal wires seek out shortest-path connections, although detours are often essential to navigate around regions of high traffic.

While it is convenient to build a regular power grid with a constant pitch (defined as the distance between adjacent wires in the grid), some degrees of freedom exist and it is desirable that they be exploited. For instance, the density of the grid in different parts of the chip need not necessarily be the same, since the use of a denser grid near hot spots with strong current sinking requirements would help in providing reliable voltages, while a sparser grid is adequate in less critical regions. This may further be combined with considerations for routing signal nets, so that in regions where the demand for routing resources from signal nets is high, a sparser power grid may be used as long as the performance constraints on the supply and ground lines can be met; likewise, signal nets are well advised to avoid the hot spots of the chip if possible, since these may need a locally dense power grid.

Strictly speaking, the results of global routing have an effect on the demands on the power grid, since the locations of buffers are determined by the routes that are chosen by signal wires. Since the number of buffers can be extraordinarily large in nanometer technologies, buffers can collectively be nontrivial contributors to power-grid current. However, these effects can be controlled at the methodology level. For example, using the model of [1], where buffers are predistributed and sprinkled into various functional blocks, we may estimate the current drawn by the buffers and include it in the current drawn by the functional block. As long as the routing is buffer-aware and deliberately seeks out regions with high buffer availability, the use

Manuscript received March 29, 2002; revised October 31, 2003. This work was supported in part by the National Science Foundation under Award CCR-0098117 and by the Semiconductor Research Corporation under Contract 99-TJ-714. This paper was recommended by Associate Editor T. Yoshimura.

H. Su and S. R. Nassif are with the IBM Corporation, Austin, TX 78758 USA (e-mail: haihua@us.ibm.com; nassif@us.ibm.com).

J. Hu is with the Electrical Engineering Department, Texas A&M University, College Station, TX 77843 USA (e-mail: jianghu@ee.tamu.edu).

S. S. Sapatnekar is with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: sachin@ece.umn.edu).

Digital Object Identifier 10.1109/TCAD.2004.837728

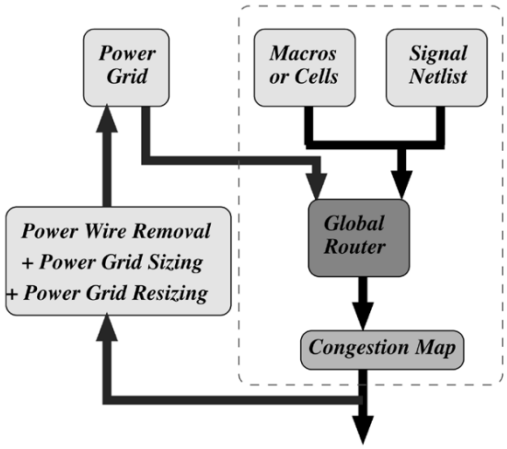


Fig. 1. Congestion-driven power-grid design and global routing.

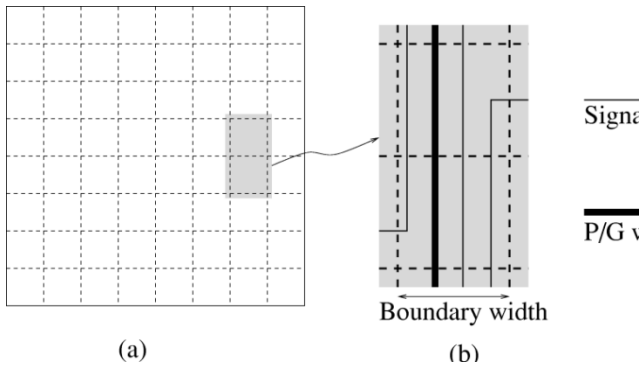


Fig. 2. Wire-congestion estimation based on tessellation on a chip.

of signal routing-independent current waveforms for the functional blocks can be justified in generating an initial power-grid design. Later in the design process, wire widening and decap assignment may be used to overcome second-order effects.

The idea of managing wire congestion in signal routing has long been a significant objective in global routing. Various congestion-driven techniques include sequential routing (e.g., [6]), rip-up-and-reroute (e.g., [24]), and multicommodity flow-based methods (e.g., [20]) have been proposed. Most of these techniques aim at solving the problem solely at the routing stage, assuming that the total routing resources are fixed. Recent publications [1], [11] have presented techniques for simultaneous global routing and resource allocation under performance constraints.

Power-grid optimization techniques have also been studied in [21], [23], and [26]. All of these aim at minimizing total power-wire area subject to the voltage drop and/or electromigration constraints, formulating the problem as a nonlinear program. Mitsuhashi and Kuh presented a mesh-based power/ground network topology optimization method for cell-based very large scale integrated circuits (VLSIs) in [15]. While [23] solves the problem by relaxing it to a sequence of linear programs, both [21] and [26] have employed gradient-based nonlinear-optimization techniques. The formulation in [21] handles the transient voltage-drop noise under worst-case switching events, while [26] considers the worst-case voltage drop and current density. The work in [13] and [14] develops techniques for shield insertion to control inductive effects, including methods that minimize both signal and power routing.

To the best of our knowledge, no published work performs a concurrent optimization of the power grid along with signal wires under routing-congestion constraints, and this is the subject of the work presented in this paper. Our proposed congestion-driven flow is illustrated in Fig. 1. The dashed rectangle corresponds to a more conventional global routing flow, where the routing budgets for power-grid lines are used to allocate routing resources for the signal lines, and these budgets are frozen throughout the design. In essence, our approach presents a new flow for early stage global wire planning that adds a feedback loop that permits the readjustment of the signal routing budgets by altering the power grid appropriately.

Our procedure begins by constructing the initial Steiner trees for the global signal wires without considering the congestion contributions of the power grid. The initial power grid is provided as an input to the algorithm (and may, perhaps, be a regular grid or a manually designed grid) and is assumed to be dense enough to sustain the switching activities of the functional blocks in the chip. Our proposed iterative scheme alternately: 1) reroutes the global signal wires and 2) adjusts the supply network to decongest regions of high contention where the voltage drop constraints are easily satisfied. The adjustments made to the power grid consist of wire removal from the grid in congested regions, sizing of the power grid to compensate for this removal, and wire-width resizing to meet the maximum current-density limit. The complexity of power-wire sizing cannot allow the adjustment procedure to be applied to the full on-chip power grid. Instead, a coarsened grid is usually used to approximate the performance of the full grid in early-stage global wire planning. Our approach incorporates a tight coupling between power-grid adjustments and the routing of signal wires to exploit the altered congestions that result from these adjustments, and aims to solve problems with severe congestion constraints, where conventional techniques are inadequate.

The remainder of the paper is organized as follows. Section II discusses the power-grid-aware congestion-estimation method and the global routing procedure. Next, an efficient power-grid noise-analysis technique is briefly described in Section III, followed by details of the heuristic power-wire removal and sizing scheme in Section IV. The overall flow of the algorithm is summarized in Section V, and experimental results on several benchmark circuits are shown in Section VI. Finally, Section VII concludes the paper.

II. Power Grid-Aware Signal Routing

As in global routing, we tessellate the entire chip into an array of grid cells, as shown in Fig. 2(a), and use the wiring information across the boundaries between neighboring grid cells to estimate the signal wire-congestion distributions. We denote the width, in micrometers, of a boundary b between two neighboring grid cells as $W(b)$. This width represents the limited resources that must be shared on each layer by the supply lines and the signal lines that traverse the boundary, as shown in Fig. 2(b).

In other words, the number of wires crossing b is inherently limited by the width $W(b)$, and $W(b)$ may partly or wholly be occupied by the crossing wires.

We represent the total width occupied by power-grid wires on boundary b as $P(b)$. If a power-grid wire p_i has a track width of $w(p_i)$, which includes its wire width and the required spacing from an adjacent wire, and there are a set of such wires, $p_1, p_2, \ldots, p_m$, that cross b, then $P(b) = \sum_{i=1}^{m} w(p_i)$ represents the space that is unavailable for signal wires to cross the boundary. Therefore, we subtract this quantity from the boundary width to obtain the space available for signal wires as $W(b) - P(b)$. Typically, a uniform track width $\bar{w}$ is applied to all the signal wires in the congestion estimation or global routing stage. Hence, signal wire congestion is often expressed in terms of the number of wiring tracks and the number of tracks available for signal wires is $T(b) = \lfloor (W(b) - P(b))/(\bar{w}) \rfloor$. If there are $S(b)$ signal wires that cross a boundary b, then the overflow on b is $\max(0, S(b) - T(b))$. All tile boundaries with positive overflow values form a congestion map for a chip. The wire density at b is represented as $S(b)/T(b)$, measuring the congestion of b. A common objective for global routing is to ensure that there is no boundary with the wire density greater than one, i.e., $S(b) \leq T(b)$.

In this work, the topology selection for the power grid is tightly coupled with the requirements of global routing for signal wires, and it is important to estimate the congestion contribution of the signal wires by implementing a fast global routing procedure. The technique used for this purpose is described in the remainder of this section, but it may be noted that this procedure may be replaced by any other computationally efficient method that shares a similar goal.

At the beginning of the algorithm, we perform a coarse global routing of all signal nets to obtain an estimate of the distribution of signal wire congestion at various boundaries, and feed this information to power/ground optimizer. The global routing technique used here is similar to [1], where a Steiner tree is initially constructed for each signal net using the AHHK algorithm [2] without considering congestion. The AHHK algorithm itself uses a two-step procedure for each net: it first constructs a minimum cost spanning tree for the net; this is then converted to a Steiner tree via a greedy overlap removal process. After the initial Steiner trees have been constructed, an iterative rip-up-and-reroute procedure is applied to further reduce the wire congestion. The outer loop, consisting of signal routing followed by power-grid optimization, is repeated iteratively until the constraints are satisfied or no further improvement is possible. In each iteration, the global routing solution from the previous iteration is used as a starting point for the rip-up-and-reroute procedure, with updated congestion values being used to direct the routing.

The rip-up-and-reroute procedure processes each net sequentially using the fixed net-ordering heuristic proposed in [16], continuing until the maximum wire density is no greater than one, or after two complete iterations, since the congestion reduction after the second iteration is limited. Each net undergoing rip-up-and-reroute is entirely deleted and then rerouted using an algorithm similar to the min–max tree algorithm in [6].

A min–max tree is a Steiner tree such that its maximum edge cost among all the edges is minimized. This tree is built on a tile graph that is the dual of the tessellation, where vertices correspond to tile rectangles and edges are used to connect the vertices corresponding to adjacent tiles in the graph. The weight on an edge is based on the wire density, so that for an edge that crosses over a boundary(edge) b, this is defined as

$$\text{weight}(b) = \begin{cases} \dfrac{S(b)+1}{T(b)-S(b)}, & S(b) < T(b) \\ L\dfrac{S(b)}{T(b)}, & \text{otherwise} \end{cases} \quad (1)$$

where L is a large number. Therefore, if the capacity is not exceeded, the weight is the number of wires crossing b, divided by the number of available tracks that remain. This is found to be particularly effective since it increases the penalty of using the boundary as the resource usage approaches the full boundary capacity, and beyond that, presents successively larger penalties for capacity violations.

III. Power-Supply Noise Analysis

As is usually done in power-grid analysis work [4], [5], we use the following linear circuit model to analyze the voltage-drop noise of the power-distribution network.

- The power grid is modeled as a resistive mesh.
- The cells/blocks are modeled as time-varying current sources connected between the power and ground plane.
- Decoupling capacitors are modeled as single lumped capacitors connected between power and ground.
- The top-level metal is connected to a package modeled as an inductance connected to an ideal constant voltage source.

This model leads to a large-scale linear circuit for power-grid analysis, which is efficiently analyzed using a technique outlined in Appendix A. The wire width optimization procedure for the supply network, described in Section B, also requires the computation of gradients, and this is performed using the simulation framework. Specifically, the transient adjoint sensitivity-analysis technique to calculate the sensitivity of the noise metric with respect to every tuning parameter, which, in our case, is the width of every power wire.

The noise and sensitivity-analysis techniques used here are substantially similar to our work on decoupling capacitor (decap) placement [22] and have, therefore, been described only very briefly here. The only difference is that the tuning parameters in decap placement are the decap values, while in this work, the tuning parameters correspond to the width of each power-grid wire. Consequently, current waveforms instead of voltage waveforms must be calculated for adjoint sensitivity computations in this work.

IV. Power-Grid Design Scheme

Starting with a dense grid that is guaranteed to meet the constraints on the supply network, our scheme iteratively sparsifies the grid to ease the wire congestion. In each iteration of the loop in Fig. 1, the power grid is adjusted using a three-step technique.

1) In the first step, a wire-removal heuristic is used to make the grid less dense in some regions, with due consideration paid to both the congestion information and the power-grid noise.
2) In the next step, a voltage drop-based wire-sizing step that adjusts the sizes of the remaining wires in the grid to compensate for the loss of wires.
3) This is followed by an additional wire-width widening step to meet the maximum current-density limit constraint.

Therefore, our procedure ensures satisfactory performance of the power grid, while utilizing just enough routing resources, so that the resources available to signal routing are maximized. The idea of altering the topology to meet current-density constraints has been studied in [15], but our work has several enhancements over this method. First, our method is explicitly congestion-driven and incorporates both signal and supply-net constraints in a concurrent optimization. Unlike their method, the wire-sizing approach presented in this section does not change the topology of the network, but that is performed in the wire-removal procedure. Second, our method analyzes transient noise while [15] operates under a dc-circuit analysis. Lastly, we differ in that we use a simplified solution that separates the electromigration solution from the nonlinear programming formulation, and solve it heuristically through a wire-width widening that is applied as a postprocessing step, since the transient problem is harder than the dc problem.

A. Power-Grid Wire-Removal Heuristic

As stated in Section II, the congestion is measured in proportion to the overflow value on each tile edge. All power wires that lie in congested regions are potential candidates for removal, except those that lie within hot spots where the voltage drop is significant. The rationale behind this is that whenever a power wire is removed, the performance of the overall grid is compromised, and this is all the more noticeable if this wire lies within a hot spot. Therefore, we define a power wire as *critical* if the worst-case voltage drop on it is beyond some specified threshold $\mathrm{Vdrop}_{\mathrm{th}}$. Critical wires are not candidates for removal even if they lie in congested regions.

In addition, we use the sensitivity of the total voltage noise integral with respect to each wire width as the measure of *criticality* of each noncritical power wire. The sensitivity, at the same time, is used to determine the gradients in wire sizing discussed in Section IV-B. Given several candidate power wires across one tile boundary with its overflow value larger than $\mathrm{OV}_{\mathrm{th}}$, the criticality of each wire determines an optimal order of removal for these wires.

The power-wire-removal heuristic proceeds as follows. First, the tile boundaries are sorted in decreasing order of their overflow values. Next, a transient adjoint sensitivity analysis is performed and all the wires are sorted according to the sensitivity values. This sorting arranges all noncritical power wires according to their importance to the overall performance of the power grid. Next, noncritical power wires are removed in the order of the sorted tile boundaries, provided that the overflow value at the boundary is larger than $\mathrm{OV}_{\mathrm{th}}$. Several practical considerations must be taken into account. First, the power-wire-removal process should be accompanied by dynamically updating the overflow value on every tile edge, so that subsequent wire removal is based on the updated congestion information. Second, a reasonable number of power wires must be removed in each iteration, and this depends on the values of $\mathrm{OV}_{\mathrm{th}}$ and $\mathrm{Vdrop}_{\mathrm{th}}$. Since the choice of these numbers is necessarily empirical and cannot be entirely relied on, we assert an upper bound for the number of power wires to be removed in any iteration to conserve the amount of computation required in the wire sizing step discussed in Section B. In our experiments, this upper bound is chosen to be 4%–6% of the total number of power wires. If a set of noncritical power wires are across one tile boundary, remove them in their increasing order of the sensitivity-based importance metric defined above. Through this rule, noncritical power wires with lower importance are removed first.

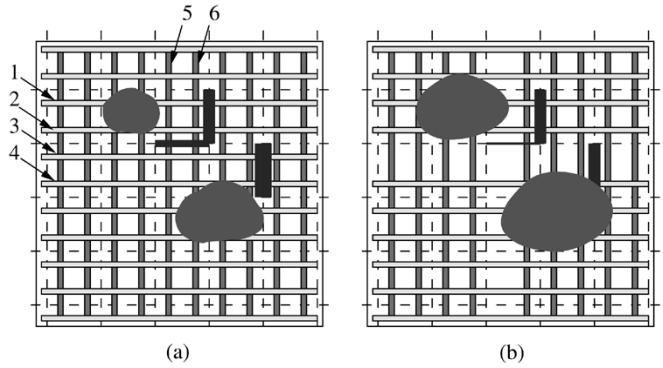


Fig. 3. Heuristic power-wire removal.

Fig. 3 illustrates a simple example of a power grid before and after a wire-removal step. The three dark lines denote congested tile boundaries whose overflow values exceed $\mathrm{OV}_{\mathrm{th}}$. The overflow values on the three boundaries are proportional to the line widths shown in the figure. The two shaded areas on the power grid represent regions where the worst-case voltage drop is larger than $\mathrm{Vdrop}_{\mathrm{th}}$. There are a total of six wires, marked 1–6, that cross the congested boundaries. Since four of them (wires 1, 2, 4, and 6) are critical wires, they are not candidates for removal. The boundaries are processed in the order of their congestion, so that first wire 3, and then wire 5, is removed from the grid. If there had been multiple candidates at any point, the one with the lowest importance metric would have been removed first. After wire removal, the area of the shaded region is increased due to the increased voltage drop, and the overflow values on every tile edge traversed by wires 3 and 5 are accordingly updated, and this translates into dark edges of reduced thickness in Fig. 3(b).

B. Voltage-Constrained Power-Grid Sizing

1) Problem Formulation: The second step in the adjustment of the power grid is related to sizing the wires in the grid to compensate for the increased voltage drop after the wire-removal step. The problem is directly formulated as a nonlinearly constrained nonlinear programming problem as follows:

$$\begin{aligned} \text{minimize} \quad & \mathrm{Area}(w_j) = \sum_{j=1}^{N_{\mathrm{wire}}} l_j \times w_j \\ \text{subject to} \quad & w_{\min} \le w_j \le w_{\max}, \quad j = 1 \ldots N_{\mathrm{wire}} \end{aligned}$$

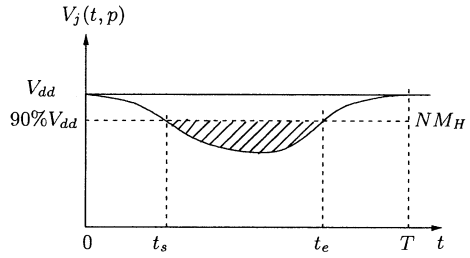


Fig. 4. Illustration of the voltage drop at a given node in the $V_{\rm dd}$ power grid. The area of the shaded region corresponds to the integral z at that node.

and

$$Z(w_j) < \epsilon \tag{2}$$

where ϵ is a very small number, and $N_{\rm wire}$ is the total number of wires in the grid.

The objective function that minimizes the total power-wire area is consistent with the goal of congestion reduction. The first constraint restricts every power-wire width to lie within a realistic range that is technology dependent. The second constraint requires the definition of the parameter Z, which is an effective metric for noise measurement that was proposed in [7]. This metric finds the integral of the noise violation and is zero if all constraints are satisfied. It punishes larger violations more severely than minor violations, and since it incorporates both the magnitude and time axes together, it is observed to be more practical than one that considers only the worst-case noise violation.

In the context of power-grid analysis, the noise at a node can be efficiently measured using the integral of the voltage drop below a user specified noise ceiling as

$$\begin{aligned} z_j(p) &= \int_0^T \max\{NM_H - v_j(t,p), 0\}\, dt \\ &= \int_{t_s}^{t_e} \{NM_H - v_j(t,p)\}\, dt. \end{aligned} \tag{3}$$

In (3), p represents the tunable circuit parameters which, in our case, are the widths of the power-grid wires. This idea is pictorially illustrated in Fig. 4, which shows the voltage waveform of one node on the $V_{\rm dd}$ grid. The noise metric for the entire circuit is defined as the weighted sum of all of the individual node metrics

$$Z = \sum_{j=1}^{K} a_j z_j(p), \qquad a_j = \frac{z_j}{\sum_{j=1}^{K} z_j} \tag{4}$$

where K is the number of nodes, and the weight a_j magnifies node j with larger voltage drop noise.

The nonlinear constraint function Z can be obtained by transient analysis of the power-grid circuit, and its sensitivity with respect to all the variables w_j can be calculated using the adjoint method discussed in Appendix B.

2) Optimization Scheme: We use a standard sequential quadratic programming (SQP) solver [3], [10] to solve the optimization problem. This solver requires users to provide subroutines to evaluate the objective and constraint functions and their derivatives with respect to each decision variable. The evaluations that are required for the SQP solver are illustrated in Fig. 5.

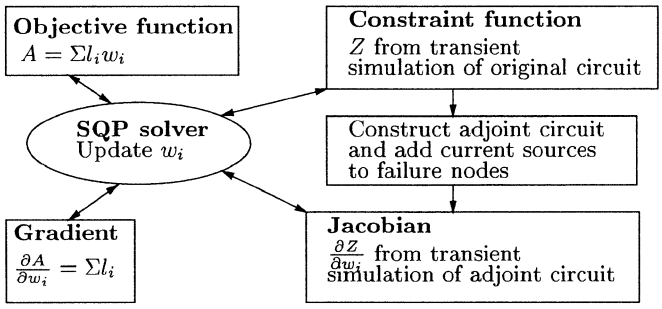


Fig. 5. Power-grid sizing procedure.

Practically, it is observed that the SQP solver converges slowly near the optimum. However, an approximate convergence is sufficiently good for our purposes, so that we require Z to be near-zero rather than exactly zero by choosing a sufficiently small ϵ.

C. Current-Density-Based Power-Grid Resizing

Electromigration is a concern related to the reliability of on-chip power grid since it directly impacts the lifetime of a chip. Empirically, the mean time to failure, a measure of the chip lifetime, has been found to vary exponentially with the average current density over time. For the purposes of our implementation, a simple current-density model, which ignores the skin effect, is used (this may be substituted by more sophisticated models) and the currents are assumed to flow through the entire cross section of each wire. The maximum current density of the entire grid, $I_{d_{\max}}$, is the peak current density among all the wire segments in this grid. A hard maximum current-density limit, $J_{\max}$, is asserted for the power grids in all metal layers. In order to guarantee that all the wires satisfy the current-density limit, we resize all the wires using

$$w_{\rm new} = \begin{cases} w_{\rm old} \times \dfrac{I_{d_{\max}}}{J_{\max}}, & I_{d_{\max}} > J_{\max} \\ I_{d_{\max}} \le J_{\max} & w_{\rm old} \end{cases}. \tag{5}$$

This is a heuristic way to deal with current density violation, working under the assumption that the optimal solution that meets the integral voltage drop constraint provides a good starting point for optimizing the current density, which we found to be a valid assumption in practice. Moreover, it was found that since the current-density violations are typically local in nature, only a small number of wires must be resized in this step, and the overall wire area and, hence, the voltage drop and current distribution, are not impacted significantly. We iteratively analyze the power grid again to update the current density until no violation can be found. Our experiments have shown that a few iterations are sufficient to remove all the current-density violations.

V. Overall Flow of the Algorithm

Having described all of the individual pieces of the algorithm, we now outline the overall flow of the procedure. The starting point is a given tessellation of a chip and a given initial power-grid construction, and the sequence of steps can be summarized as follows.

1) An initial power-grid-aware global routing step is carried out to route the signal nets. Each net is first routed without considering congestion, followed by an iterative rip-up-and-reroute, described in Section II, using the updated congestion information.
2) Based on the routes used by the power grid and the signal routes, a congestion map is generated and the overflow on each tile boundary is calculated. All tile boundaries whose overflow value exceeds the threshold $\mathrm{OV_{th}}$ are identified and are sorted in decreasing order of this value.
3) A transient simulation of the power grid is performed to identify critical power wires on which the worst-case voltage drop is above the threshold, $\mathrm{Vdrop_{th}}$.
4) Sort the noncritical power wires in increasing order of the criticality.
5) Based on the congestion map generated in Step 2, noncritical power wires are removed according to the heuristic described in Section IV-A.
6) To compensate for the removal of these wires, the remaining power-grid wires are sized using the nonlinear optimization procedure described in Section IV-B and resized heuristically using the method described in Section IV-C.
7) The congestion maps are updated, and the global routing is updated by performing rip-up-and-reroute based on the new congestions. At this point, the iterative loop is invoked so that the updated power grid and congestion map are fed back to Step 3. The stopping criterion for the iteration is that the maximum overflow should be no greater than 1, or that no further improvement is possible. The latter is easily identified if it is detected that the changes in the congestion map after rip-up-and-reroute are insignificant, and that no further deletion of the power wires is possible.

The initial routing takes $O(MN^3)$ time, where N is the number of pins for each net and M is the total number of nets. The rip-up-and-reroute has a complexity of $E \log\log E$, where E in our case is the total number of tile edges. The complexity of power-wire removal is $O(K \log K + P \log P + KPE)$, where K is the total number of tile edges with negative overflow values, P is total number of power wires and E is total number of tile edges. The first term stands for the complexity of sorting these tile edges, the second term is the worst-case complexity of sorting the criticality of noncritical power wires, and the third term comes from the power-wire removal and dynamic tile edge overflow update process. The power-grid analysis and sensitivity analysis has the complexity of one LU decomposition and two forward/backward substitutions. In practice, the cost of these computations is just over $O(n)$ for a sparse positive definite matrix, where n is the total number of nodes and inductance branches. For a nonlinear optimization problem with w decision variables, advanced implementations of the SQP solver have $O(w^2)$ cost. The complexity of current density-based power-grid resizing is proportional to the complexity of the transient simulation, which is $O(n)$. Therefore, the worst-case complexity for our wire sizing procedure ends up to be $O(I(n + w^2) + I_c(n))$, where I is the total number of nonlinear optimization iterations and I_c is the total number of iterations for the current density-based resizing. The efficiency of the gradient-based SQP solver relies largely on the initial solution and its distance away from an optimum. Practically, it is seen that the optimal solution can be reached in a limited number of iterations. Similarly, the current density violations can be removed within a very limited number of iterations.

TABLE I
TEST-CIRCUIT PARAMETERS

Circuit	B	N	Tile size
apte	9	77	31x36
ami33	33	112	30x28
ami49	49	368	33x34
playout	62	1294	30x26
ac3	27	200	29x28
hc7	77	430	34x39
a9c3	147	1148	33x31

TABLE II
COMPARISON OF THE CONGESTION IMPROVEMENT AFTER GLOBAL ROUTING USING THE CONVENTIONAL APPROACH AND OUR NEW METHOD

Circuit	Method	D_{max}	Overflow
apte	Traditional	2.50	28
	PSiCo	1.67	3
ac3	Traditional	2.20	251
	PSiCo	1.00	0
ami33	Traditional	3.33	109
	PSiCo	1.00	0
ami49	Traditional	1.88	195
	PSiCo	1.00	0
playout	Traditional	1.90	154
	PSiCo	1.00	0
hc7	Traditional	1.50	111
	PSiCo	1.00	0
a9c3	Traditional	1.43	198
	PSiCo	1.00	0

Several assumptions have been made in our flow, and we discuss their implications below.

- Our method always starts with a conservative power grid that meets the power budget. However, this may introduce a significant amount of routing congestion, and this is alleviated by our technique. The problem of jointly meeting the power and signal net constraints does not always have a feasible solution, and the optimal solution returned by the SQP solver can be considered as a commentary on the quality of the original design. When the solver ends up with no solution, or when it returns a solution that satisfies (or nearly satisfies) the voltage drop constraints but utilizes much larger power-grid resources than the original one, this indicates a poor design. In such a case, the original grid or certain modules of the chip should be redesigned to be able to meet all the requirements. Similarly, if the solver always returns a solution that is close to the original grid before the wire removal, routing congestion can barely be further improved by changing the power grid. In such cases, we will have to either try to remove a small segment of an entire power grid that crosses the most congested region and rerun the sizing optimization, or redesign the original power grid using a different pitch and/or width and restart our flow.
- Our power-grid removal and sizing operates on the entire length of each wire across the chip. It can easily be extended to remove the portion of a grid that occupies the most congested region. In this case, one long grid could be split into several shorter grids during the wire-removal

TABLE III
OPTIMAL POWER GRID RESULTS

Circuit	Before Optimization				After Optimizatioin				
	Area	Nodes	W	Drop	Area	Nodes	W	Drop	Z (V × ns)
apte	16.8%	3411	133	0.178V	14.0%	3318	121	0.165V	0.00
ac3	20.9%	10984	169	0.177V	14.5%	8736	133	0.171V	0.21e-3
ami33	16.1%	11267	171	0.178V	10.9%	8859	133	0.167V	0.00
ami49	17.4%	17360	179	0.179V	13.2%	15398	152	0.173V	1.49e-3
playout	19.3%	22414	233	0.177V	14.7%	16627	195	0.172V	0.62e-3
hc7	18.5%	27288	237	0.176V	12.5%	22160	208	0.177V	9.76e-3
a9c3	19.3%	31193	251	0.178V	13.7%	28168	216	0.175V	1.49e-2

process. Furthermore, wire sizing can potentially be extended to work on wire segments between adjacent nodes. For the purposes of this paper, we do not address these issues, particularly since they may cause the size of the problem to increase greatly and may cause other problems with the methodology.

- Our method is directed toward early design planning, and does not attempt to solve the problem on a full on-chip power grid later in the design. Such a problem is very different, since the power grid can have up to tens of millions of nodes, where it becomes infeasible to perform the nonlinear optimization which requires extensive transient and sensitivity analyses of the power-grid circuits. However, for such large-scale problems, our method can be extended by coarsening and widening the original power grid to some manageable size, and then fed into our flow to find the optimal grid solution. The fine grid can then be reconstructed according to the total optimal grid area and local density. This borrows the multigrid idea presented in [12], [17], and [25].
- Our congestion-driven engine will automatically guide global routing into less congested regions of a chip. Buffers are not considered at this early wire planning stange. This is typically the practice in industrial flows. If buffer effects are to be considered, generally speaking, making the wiring distribution more evenly will help in distributing the buffers. Moreover, buffer currents are significantly smaller than block or macro currents. Therefore, wire sizing can typically compensate the current redistribution due to the buffer change. Alternatively, our method can easily be adapted by considering the buffer cost during routing, and by adding the precharacterized buffer currents to the grid during the wire-removal and wire-resizing phases.
- Power wires are often used to shield signal wires to reduce the crosstalk noise. For crosstalk-sensitive nets, a wire-width inflation ratio can be applied to increase the cost of these nets, to account for the cost of the shields that can be inserted later on at the end of our flow. These additional shields will not impact the wiring congestion because their costs are already counted during routing. On the other hand, the addition of extra wires can only improve the overall power-grid performance.

VI. EXPERIMENTAL RESULTS

The power-grid analyzer and the global router have both been implemented in C++. The power-grid removal and sizing scheme, and the overall congestion-driven power-grid design and global routing flow has been written using Tcl. The wire-size optimization is performed using an off-the-shelf SQP solver [10]. All experiments are performed on an Intel Xeon 2.4-GHz server with 512-MB memory running Redhat Linux 9.0. The entire procedure is encapsulated in a flow called power-signal codesign (PSiCo).

TABLE IV
CPU TIME STATISTICS AND THE NUMBER OF ITERATIONS FOR THE BENCHMARKS

	apte	ac3	ami33	ami49	playout	hc7	a9c3
CPU (min)	5.7	14.2	19.36	21.84	44.32	69.48	59.58
# Iter.	5	5	6	5	4	4	4

We have tested our flow on seven benchmark circuits obtained from the authors of [1]. All designs correspond to a 0.18-μm technology and a supply voltage of 1.8 V. The time-varying current sources modeling the behavior of each functional block was not originally available in these benchmark circuits. These waveforms are constructed by modifying current waveforms from several industrial circuits by adjusting their magnitude according to the area of each block. However, this is not a critical assumption since our method is applicable to the exact waveforms where available.

Initially, six layers of regularly distributed power grids with fixed wire widths are generated for all of these examples, with each layer containing only horizontal or vertical wires. Since a majority of the global routes are typically seen on the third and fourth metal layers (M3 and M4), we perform the global routing and power-grid sizing on these two layers, assuming that the other layers are processed separately. The wire widths on these layers are assumed to be constrained within the range of 0.8 and 4 μm in our experiments. The initial power grid is constructed with a constant pitch in M3 and M4, such that, under the given set of time-varying current sources that represent each block, the worst-case voltage drop on the entire power grid is within a threshold, which in our experiments, is chosen to be 0.18 V. Table I lists the characteristics of each circuit, in terms of the total number of blocks B and nets N and the tile sizes.

The performance of PSiCo can be described in terms of two components: the global routing solution, and the performance of the final power grid, and these results are shown in Tables II and III, respectively. The wire congestion results of PSiCo are compared in Table II against the results of the traditional rip-up-and-reroute method, which corresponds to the result at the end of Step 1 in Section V. For each circuit, the maximum density $D_{\max}$, calculated as the maximum ratio of the utilization to the capacity across any tile boundary. Also shown is the overflow, i.e., the total amount by which the tile boundary capacities are

TABLE V
PERCENTAGE CHANGE IN THE AREA IN EACH STAGE OF EVERY ITERATION

		apte	ac3	ami33	ami49	playout	hc7	a9c3
1	Removal	-6.88%	-4.83%	-6.69%	-4.46%	-4.44%	-4.42%	-4.51%
	Sizing	-1.39%	-2.70%	-2.19%	-13.8%	-4.10%	-4.62%	-5.06%
	Resizing	+0.10%	+0.24%	+0.09%	+0.07%	+0.83%	+0.06%	+0.06%
2	Removal	-2.31%	-4.32%	-6.92%	-3.86%	-4.79%	-4.38%	-4.64%
	Sizing	-1.42%	-2.79%	-2.28%	-0.27%	-3.56%	-4.57%	-5.31%
	Resizing	+0.02%	+0.16%	+0.05%	+0.20%	+0.25%	+0.27%	+0.16%
3	Removal	-0.00%	-4.47%	-6.47%	-4.14%	-1.79%	-3.64%	-4.47%
	Sizing	-1.75%	-2.93%	-2.33%	-0.08%	-3.42%	-17.4%	-4.76%
	Resizing	+0.02%	+0.06%	+0.06%	+0.08%	+0.06%	+1.50%	+0.33%
4	Removal	-0.00%	-4.68%	-2.51%	-1.44%	-1.95%	-0.31%	-1.31%
	Sizing	-2.03%	-2.87%	-2.22%	+0.75%	-3.61%	-0.00%	-4.61%
	Resizing	+0.02%	+0.19%	+0.06%	+0.05%	+0.06%	+0.25%	+0.36%
5	Removal	-0.00%	-4.86%	-0.00%	-0.00%			
	Sizing	-2.31%	-2.60%	-2.74%	+0.48%	(converged)	(converged)	(converged)
	Resizing	+0.03%	+0.39%	+0.07%	+0.05%			
6	Removal			-1.19%				
	Sizing	(converged)	(converged)	-2.93%	(converged)			
	Resizing			+0.07%				

violated, summed over all boundaries. It can be seen that for all the cases PSiCo gives much better congestion results than the conventional method.

Performance metrics for the initial power grid and the power grid obtained by PSiCo are listed in Table III, in specific, the power-wire area as a percentage of the area available on the two layers, the total number of nodes, the number of wires (W) in the power grid and the worst-case voltage drop. Theses numbers after optimization are also shown in the table and may be compared with the corresponding numbers before optimization. It can be seen that an optimal grid that takes less percentage of the total chip area can still provide the sufficient amount of resources needed for signal routing.

In each of the test cases, all grids whose worst-case voltage drop exceeds 0.14 V (stricter than 10% $V_{\rm dd}$ because layers M3 and M4 instead of M1 and M2 are considered) are marked as critical wires not to be removed. The voltage drop constraint (noise margin) for noise computation is chosen to be 0.17 V, which is slightly stricter than 10% $V_{\rm dd}$. To aid the rate of convergence, we heuristically terminate the SQP solver when we detect that the worst-case voltage drop in the circuit is below 0.18 V, and it can be seen that the worst-case voltage drop always satisfies this requirement. We choose 5.8 $\times 10^5$ A/cm^2 [19] as the maximum current-density limit in our experiment. The current-density-based wire-width widening slightly increases the total wire area and therefore the worst-case voltage drop becomes better than 10% $V_{\rm dd}$ for all the testcases. The optimized noise integral Z for each circuit can be seen to be very small, and is nonzero due to the fact that voltage drops between 0.17 and 0.18 V are tagged as "violations" by the SQP solver.

Table IV lists the total number of iterations and the CPU time, in minutes, required for each circuit. It can be seen that the number of iterations is typically small, and that the CPU times for these circuits are very reasonable.

Lastly, Table V shows the percentage change in the area of power wires, in each iteration, at the three stages of our power-grid design scheme: congestion-driven wire removal ("Removal" in the table), voltage constrained wire sizing ("Sizing" in the table) and current density-based power-grid resizing ("Resizing" in the table). A positive sign indicates an area increase, while a negative sign indicates an area reduction.

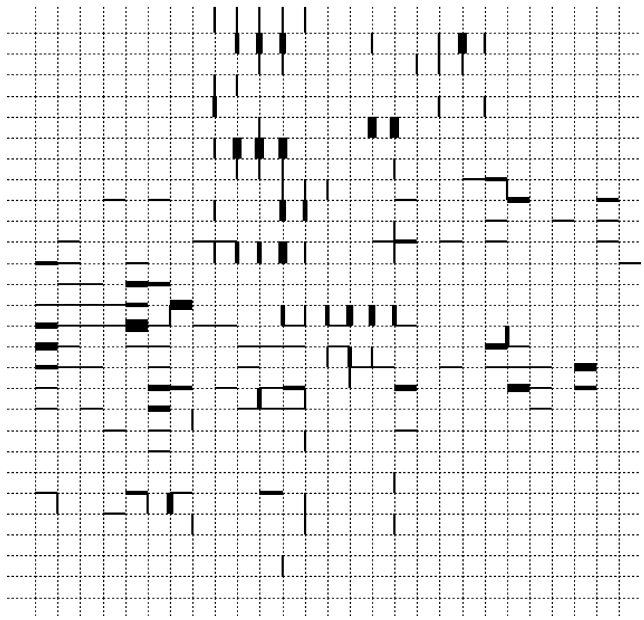


Fig. 6. Congestion map after the initial routing.

It is obvious that in the first stage of wire removal, this is negative, while in the third stage of current density-based resizing, it is positive, and this is seen in our results. In most of the cases, in the second stage where the SQP solver is applied, the total wire area is reduced, although in iteration 4 and 5 of "ami49," the SQP solver has to increase the total wire area to meet the voltage-drop constraint.

Fig. 6 shows the congestion map (29×28) of circuit ac3 after the initial routing. The overflow value on each tile boundary corresponds to the width of the dark line, so that congested regions are clearly identified. Through a transient simulation of the initial power-grid circuit, the worst-case voltage drop for nodes on layers M3 and M4 are analyzed and shown as a contour plot in Fig. 7. In Fig. 7, the small ovals represent GND c4 locations. Only the ground plane is shown because our experiments show voltage drops in the ground plane are dominant for this circuit. Hot spots are plotted as darkest regions in the figure. The chip cell placement and the final optimal power grid are shown in Fig. 8. By examining Figs. 6 and 7 and the locations and widths

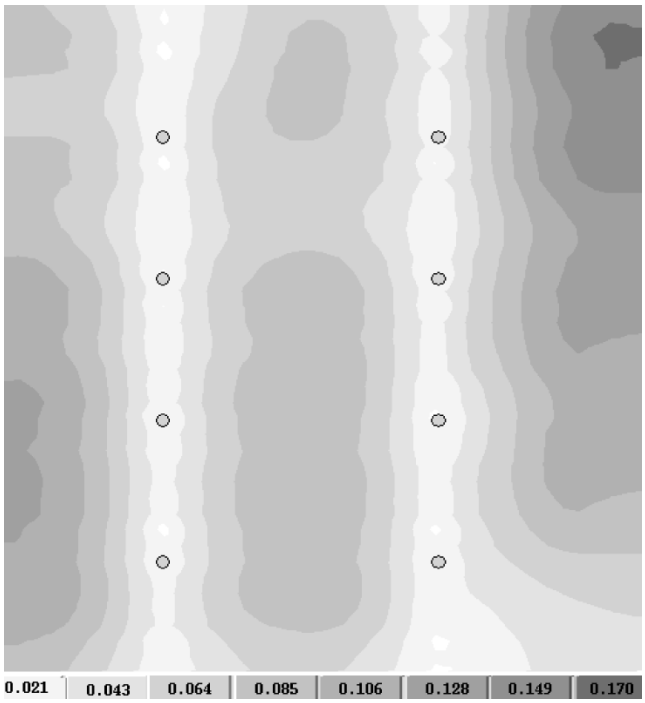


Fig. 7. Initial voltage-drop contour on the ground plane.

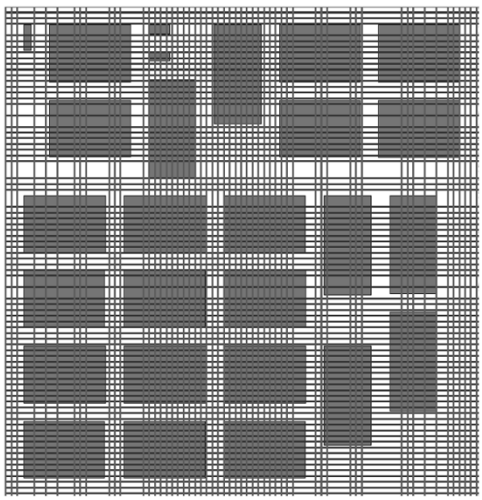


Fig. 8. Optimal M3 and M4 power grid of ac3.

of power wires in Fig. 8, it can be seen that power wires away from hot spots and across congested tile boundaries have been removed and wider and/or denser power wires lie around hot spots. However, the widths of the wires cannot be seen easily in Fig. 8, due to the limited resolution. Fig. 9 shows the ground plane worst-case voltage drop contour plot for layers M3 and M4 of the final optimal power grid.

VII. Conclusion

We have proposed a new design flow for the codesign of signal routes and power grids. The technique is guided by congestion maps, and proceeds by removing noncritical power wires greedily in congested areas, and rerouting the signal wires according to the updated congestions. The effects of removing power wires are compensated for by a gradient-based

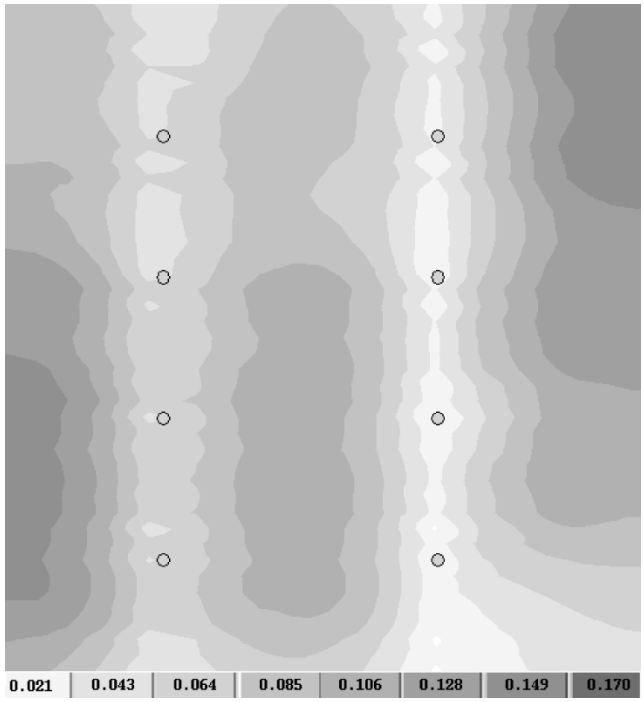


Fig. 9. Voltage-drop contour on the ground plane of the optimal power grid.

wire sizing for voltage noise and a heuristic wire widening for the current-density limit. Experimental results for several benchmark circuits are presented in this paper. Future work includes incorporating the multigrid idea into our flow and constructing a reliable back-mapping scheme according to the optimal power-grid area and density at the coarse level.

Appendix A
Power-Grid Analysis

The behavior of the linear circuit discussed in Section III is described using the modified nodal analysis [18] as follows:

$$Gx(t) + C\dot{x}(t) = u(t) \tag{6}$$

where x is a vector of node voltages and source and inductor currents, G is the conductance matrix, C includes both the decoupling capacitance and package inductance terms, and $u(t)$ includes the loads and voltage sources.

By applying the Backward Euler integration formula [18] to (6), we have

$$(G + C/h)x(t+h) = u(t+h) + x(t)C/h \tag{7}$$

where h is the time step for the transient analysis. If h is kept constant, only a single initial factorization of the matrix $G+C/h$ is required (as is done in [17] and [27]), leading to an efficient algorithm for transient analysis, where each time step requires only a forward/backward solution step. After the transient analysis of the circuit, the voltage waveform at every node and the current waveform across every circuit branch are known.

Appendix B
Sensitivity Computation

Adjoint sensitivity analysis is a standard technique for circuit optimization where the sensitivity of one performance function

with respect to many parameter values is required [8], [9], [18]. We use a method developed in [22] to find the sensitivity of the scalar function Z described by (4) with respect to every tuning parameter in the circuit. The only minor difference is that in [22], the tuning parameter is the set of decap widths, while in this paper, it corresponds to the width of each wire in the power grid.

The sensitivity of noise Z, with respect to all of the resistors in the circuit, can be computed from the following convolution [8], [9]:

$$\frac{\partial Z}{\partial R} = \int_0^T \varphi_R(T-t) i_R(t)\, dt \tag{8}$$

where $\varphi_R(\tau)$ is the current waveform flowing through the resistor R in the adjoint circuit.

The memory storage and speed problem for the waveform convolution is solved by applying a piecewise linear technique proposed in [22]. The complexity of the convolution calculation over $[0, T]$ is $O(N+M)$, where N and M are the number of linear segments on the original and adjoint current waveforms.

The sensitivities to the *width* of each wire width can then be calculated using the chain rule

$$\frac{\partial Z}{\partial w} = \sum_{i=1}^{P} \frac{\partial Z}{\partial R_i} \times \frac{\partial R_i}{\partial w} \tag{9}$$

where P is the total number of segments in one grid wire with the same width w. Since the resistance $R_i = \rho(l_i)/(wh)$, where l_i the length of the wire segment, w is the width of the wire, h and ρ are the thickness and resistivity of the metal layer that the power wire lies in, it is easily verified that (9) becomes

$$\frac{\partial Z}{\partial w} = -\sum_{i=1}^{P} \frac{\partial Z}{\partial R_i} \times \frac{\rho l_i}{hw^2}. \tag{10}$$

REFERENCES

[1] C. J. Alpert, J. Hu, S. S. Sapatnekar, and P. G. Villarrubia, "A practical methodology for early buffer and wire resource allocation," in *Proc. Design Automation Conference*, Las Vegas, NV, June 2001, pp. 189–194.

[2] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, and D. Karger, "Prim-dijkstra tradeoffs for improved performance-driven routing tree design," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 890–896, July 1995.

[3] P. T. Boggs and J. W. Tolle, *Sequential Quadratic Programming*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[4] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," in *Proc. Design Automation Conf.*, Anaheim, CA, June 1997, pp. 638–643.

[5] H. H. Chen and J. S. Neely, "Interconnect and circuit modeling techniques for full-chip power supply noise analysis," *IEEE Trans. Comp., Packag., Manufact. Technol. B*, vol. 21, pp. 209–215, Aug. 1998.

[6] C. Chiang and M. Sarrafzadeh, "Global routing based on steiner min–max tree," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 1318–1325, Dec. 1990.

[7] A. R. Conn, R. A. Haring, and C. Visweswariah, "Noise considerations in circuit optimization," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 1998, pp. 220–227.

[8] S. W. Director and R. A. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Trans. Circuit Theory*, vol. 16, pp. 318–323, Aug. 1969.

[9] P. Feldmann, T. V. Nguyen, S. W. Director, and R. A. Rohrer, "Sensitivity computation in piecewise approximation circuit simulation," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 171–183, Feb. 1991.

[10] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for SOL/QPSOL: A Fortran Package for Quadratic Programming*. Stanford, CA: Dept. Oper. Res., Stanford Univ., July 1983.

[11] J. Hu and S. S. Sapatnekar, "A timing-constrained algorithm for simultaneous routing of multiple nets," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, 2000, pp. 99–103.

[12] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "Fast power grid simulation," in *Proc. Design Automation Conf.*, Los Angeles, CA, June 2000, pp. 156–161.

[13] J. D. Z. Ma and L. He, "Simultaneous signal and power routing under Keff model," in *Proc. Int. Workshop Syst.-Level Interconnect Prediction*, Sonoma, CA, 2001, pp. 175–182.

[14] ——, "Toward global routing with *RLC* crosstalk constraints," in *Proc. Design Automation Conf.*, New Orleans, LA, 2002, pp. 669–672.

[15] T. Mitsuhashi and E. S. Kuh, "Power and ground network topology optimization for cell-based VLSIs," in *Proc. Design Automation Conf.*, Anaheim, CA, June 1992, pp. 524–529.

[16] R. Nair, "A simple yet effective technique for global wiring," *IEEE Trans. Computer-Aided Design*, vol. 6, pp. 165–172, Mar. 1987.

[17] S. R. Nassif and J. N. Kozhaya, "Fast power grid simulation," in *Proc. Design Automation Conf.*, Los Angeles, CA, June 2000, pp. 156–161.

[18] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, *Electronic and System Simulation Methods*. New York: McGraw-Hill, 1995.

[19] The International Technology Roadmap for Semiconductors (1999). [Online]. Available: http://public.itrs.net/Files/1999_SIA_Roadmap

[20] E. Shragowitz and S. Keel, "A global router based on a multicommodity flow model," *Intergration: VLSI J.*, vol. 5, no. 1, pp. 3–16, 1987.

[21] H. Su, K. H. Gala, and S. S. Sapatnekar, "Fast analysis and optimization of power/ground networks," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 2000, pp. 477–480.

[22] H. Su, S. S. Sapatnekar, and S. R. Nassif, "An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts," in *Proc. Int. Symp. Phys. Design*, San Diego, CA, 2002, pp. 68–73.

[23] X. Tan, C. J. R. Shi, D. Lungeanu, J. Lee, and L. Yuan, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," in *Proc. Design Automation Conf.*, New Orleans, LA, June 1999, pp. 156–161.

[24] B. S. Ting and B. N. Tien, "Routing and techniques for gate array," *IEEE Trans. Computer-Aided Design*, vol. 2, pp. 301–312, Oct. 1983.

[25] K. Wang and M. Marek-Sadowska, "On-chip power supply network optimization using multigrid-based technique," in *Proc. Design Automation Conf.*, Los Angeles, CA, June 2003, pp. 113–118.

[26] X. Wu, X. Hong, Y. Cai, C. K. Cheng, J. Gu, and W. Dai, "Area minimization of power distribution network using efficient nonlinear programming techniques," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, 2001, pp. 153–157.

[27] M. Zhao, R. V. Panda, S. S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw, "Hierarchical analysis of power distribution networks," in *Proc. Design Automation Conf.*, Los Angeles, CA, June 2000, pp. 481–486.

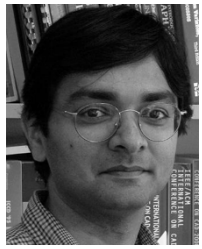
Haihua Su (S'00–M'03) received the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, in 2002.

After graduation, she joined the IBM Austin Research Laboratory, Austin, TX, as a Postdoctoral Fellow. She is currently with the IBM EDA, Austin, TX. Her research interests are in power grid and global connect analysis and optimization, global and macro level noise analysis, process variation, and leakage analysis.

Jiang Hu (S'98–M'01) received the B.S. degree in optical engineering from Zhejiang University, Hangzhou, China, in 1990, the M.S. degree in physics from the University of Minnesota, Duluth, in 1997, and the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, in 2001.

He was with IBM Microelectronics Division in 2001 and 2002. He is currently an Assistant Professor in the Department of Electrical Engineering, Texas A&M University, College Station. His current research interest is on VLSI design automation, especially on interconnect optimization, clock network synthesis, design for manufacturability, and uncertainty tolerant design.

Prof. Hu has served on the Technical Program Committee for the International Conference on Computer Design, the International Symposium on Physical Design, and the International Symposium on Circuits and Systems. He received a Best Paper Award at the Design Automation Conference in 2001.



Sachin Sapatnekar (S'86–M'88–SM'99–F'03) received the B.Tech. degree from the Indian Institute of Technology, Bombay, in 1987, the M.S. degree from Syracuse University, Syracuse, NY, in 1989, and the Ph.D. degree from the University of Illinois, Urbana-Champaign, in 1992.

From 1992 to 1997, he was an assistant professor in the Department of Electrical and Computer Engineering at Iowa State University, Ames. He is currently a Professor in the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis.

Dr. Sapatnekar received the National Science Foundation Career Award and best paper awards at the ACM/IEEE Design Automation Conference in 1997, 2001, and 2003, and at the IEEE International Conference on Computer Design in 1998. He has been an Associate Editor for the IEEE TRANSACTIONS ON VLSI SYSTEMS, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II. He has served on the Technical Program Committee for various conferences, and as Technical Program and General Chair for the Tau workshop and the International Symposium on Physical Design.

Sani R. Nassif (S'80–M'86–SM'02) received the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, 1986.

He worked for ten years at Bell Laboratories on various aspects of design and technology coupling including device modeling, parameter extraction, worst case analysis, design optimization, and circuit simulation. He joined the IBM Austin Research Laboratory, Austin, TX, in 1996, where he is presently managing the Tools and Technology Department, which focuses on design/technology coupling, timing simulation and analysis, testing, and low power design.