Early-stage Power Grid Analysis for Uncertain Working Modes

Haifeng Qian, Sani R. Nassif, *Senior Member, IEEE,* and Sachin S. Sapatnekar, *Fellow, IEEE*

***Abstract*— High performance integrated circuits are now reaching the 100-plus watt regime, and power delivery and power grid signal integrity have become critical. Analyzing the performance of the power delivery system requires knowledge of the current drawn by the functional blocks that comprise a typical hierarchical design. However, current designs are of such complexity that it is difficult for a designer to determine what a realistic worst-case switching pattern for the various blocks would be in order to maximize noise at a specific location. This paper uses information about the power dissipation of a chip to derive an upper bound on the worst-case voltage drop at an early stage of design. An exact integer linear programming (ILP) method is first developed, followed by an effective heuristic to speed up the exact method. A circuit of 43K nodes is analyzed within 70 seconds, and the worst-case scenarios found correlate well with the results from an ILP solver.**

***Index Terms*— Power grid, supply network, early-stage simulation, random walk.**

I. Introduction

INTEGRATED circuits are rapidly growing more and more power-intensive. For example, [13] reports an Itanium processor with a worst-case power dissipation of 130W, and a power dissipation of 110W for the average case; [7] reports an Alpha processor with an estimated power consumption of 100W. These increased power values imply rising average current carried by the power and ground grids. In combination with the reduction in VDD values, this implies that power grid noise is becoming a larger fraction of the supply voltage in successive technology generations. The net effect of this is a reduction in noise margins and an increase in the variability of gate delays. Some of the major causes for this increase can be attributed to increases in wire resistances and in the currents generated per unit area from one technology node to the next, which together cause IR drops on power grids to worsen. Since power grids play an important role in determining circuit performance, their accurate and efficient analysis is critical at all stages of the design cycle.

Several analyzers have been proposed to handle large circuit sizes efficiently [1][8][10][11][12][14][15]. Most of these techniques deal with the deterministic analysis of a power grid for a complete design; in other words, they assume that the current loads at bottom-layer nodes are given, and power grid analysis is performed subsequent to this. On the other hand, [6] proposes to perform analysis without deterministic current loads, and instead, uses current constraints to limit possible working modes, formulating a linear programming problem to find the worst voltage drops.

This paper is motivated by two issues that have not been adequately addressed by prior works:

- **To efficiently model *uncertain working modes*** Modern designs operate under a number of power modes, in each of which a different set of blocks may be on. This uncertainty can exert a large influence on power-grid performance. The current loads in our work are modeled not as constants, but as functions of the working mode of the chip, and we look at power grid analysis for these uncertain loads, to find the worst-case scenario associated with the largest voltage drop. To do so, we utilize information that was not used in deterministic analysis. For example, power budget is an increasingly useful piece of information that can be used during analysis when circuits operate under multiple power modes.
- **To perform *early stage analysis*** The most effective fixes to the power grid must be made *early* in the design cycle, when much of the details of the design are unknown. If one waits until later in the design flow, the number of available degrees of freedom for optimization reduces dramatically. This implies that it is important to analyze the grid early in the design process; however, the side-effect of this is that such analyses must operate under some uncertainty as to the exact loads.

This paper focuses on power grid design at early stages of design, under uncertain working modes. The information that is available at this stage, say, after floorplanning is that the chip is composed of a number of functional blocks whose positions are known. For example, an SRAM block on the chip can be modeled by one current source distributed over hundreds of power grids nodes. One may determine a reasonable estimate for current consumed by each block, and based on the position of a block, its proximity to VDD/GND pads is known. In different working modes, some of the blocks are active and consuming current, while others are standing by. The number of working modes for the circuit may be very large (potentially exponential in the number of blocks), and it is often not possible to enumerate all such modes.

One way to deal with this uncertainty is to perform a worst-case analysis assuming that every block is on. This is too pessimistic and produces false alarms, since such a working mode may never occur. Figure 1 shows a small illustrative

This work was supported in part by the SRC under contract 2003-TJ-1092, by the NSF under award CCR-0205227, and by an IBM Faculty Award.

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

S. R. Nassif is with the IBM Austin Research Laboratory, Austin, TX 78758 USA (e-mail: nassif@us.ibm.com).

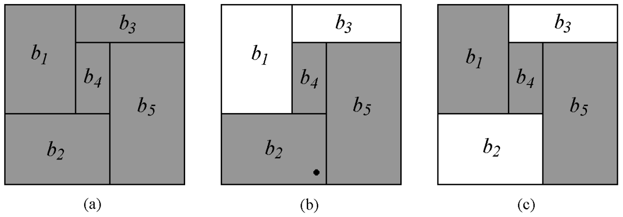
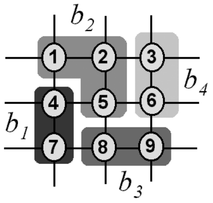


Fig. 1. A small example: (a) the floorplan with five blocks, (b) the working mode that causes the largest single-node voltage drop, with the location of the largest voltage drop marked with a black dot, and (c) the working mode that causes the largest average voltage drop.



$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \end{bmatrix} = \begin{bmatrix} & 0.3 & & \\ & 0.3 & & \\ & & & 0.2 \\ 0.5 & & & \\ & 0.4 & & \\ & & & 0.8 \\ 0.5 & & & \\ & & 0.3 & \\ & & 0.7 & \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} I_{b_1} \\ I_{b_2} \\ I_{b_3} \\ I_{b_4} \end{bmatrix}$$

Fig. 2. A small example of equation (2).

example to show the necessity of analyzing realistic working modes instead of making this pessimistic assumption. It is an artificial regular-structured power grid with 5884 nodes, four metal layers, and four VDD pads at the top metal layer. Each of the 2601 bottom-metal-layer nodes has a 2mA current load. The current loads are drawn by five functional blocks, as shown in Figure 1(a), and each current load is on when the corresponding block is active. If we assume all five blocks are on, the total power is 6.2W (the nominal VDD being 1.2V), the largest single-node voltage drop is 135mV, and the average voltage drop over all bottom-metal-layer nodes is 114mV. However, if we have the knowledge that the maximum power of the chip is 4W, then not all working modes can occur. By enumerating all possible working modes that obey the 4W constraint, Figure 1(b) is found to be the working mode that causes the largest single-node voltage drop, 99mV at the location indicated by the black dot, and Figure 1(c) is found to be the working mode that causes the largest average voltage drop, 73mV. Hence, for this small example, the pessimistic analysis overestimates the voltage drop by 36mV, and could produce false alarms.

In this paper, we use some constraints to limit the analysis to working modes that are more likely to occur, and find the worst among them. Examples of these constraints are:

- A power-limit constraint indicates that a design cannot consume more than a certain amount of power $P_{\max}$. This number can be provided by the power budget that is set as a constraint early in the design process.
- A synchronization constraint demands that two blocks always work together.
- An exclusivity constraint provides that only one of two RAM blocks may be accessed at a time, or that only one of three ALUs is active at a time, etc.

Under such constraints, the worst case working mode needs to be found, in terms of either the largest single-node voltage drop, or in terms of the largest average voltage drop. If the specified voltage drop design goal is violated, this must be fixed by assigning more routing resources to the power grid and/or moving certain blocks apart from each other. This type of early-stage optimization can substantially reduce the risk of later optimizations that may require expensive rip-up-and-reroutes.

Another example where the analysis of a power grid under uncertainty is also meaningful is the case when there is a critical noise-sensitive block in the design. For example, a phase-locked loop is sensitive to VDD noise, i.e., sensitive to the working mode of circuit units around it, and requires careful analysis [3]. In this case, the scenario that causes the largest voltage drop at these specific nodes must be found to guarantee correct analysis of the unit.

The problem discussed above is formulated in Section 2. Section 3 presents a heuristic solution. Simulation results are provided in Section 4, and Section 5 lists our concluding remarks.

II. Problem Formulation

Our approach will be based on DC analysis, as little is known about circuit waveform details at the early stage, and it is impractical to perform transient analysis. The DC analysis of a GND net is formulated as [5]:

$$G\mathbf{X} = \mathbf{I} \tag{1}$$

where G is the conductance matrix for the interconnected resistors, $\mathbf{X}$ is the vector of node voltages, and $\mathbf{I}$ is the vector of current loads. For a VDD net, the right-hand-side vector also contains perfect VDD sources, but if we look at the voltage drops, i.e., if we subtract every entry in $\mathbf{X}$ by VDD and reverse its sign, the formulation becomes the same as equation (1).

To investigate variations in load vector $\mathbf{I}$, we must account for the origin of current loads. In reality, vector $\mathbf{I}$ is composed of contributions from functional blocks, and can be formulated as:

$$\mathbf{I} = A \cdot \mathrm{diag}(\mathbf{w}) \cdot \mathbf{I_b} \tag{2}$$

where $\mathbf{I_b}$ is a k-dimensional vector of block currents, $\mathbf{I}$ is the n-dimensional vector of current loads, A is an n-by-k matrix, $\mathbf{w}$ is a k-dimensional vector with entries being 0 or 1, and $\mathrm{diag}(\mathbf{w})$ is a k-by-k diagonal matrix with diagonal entries equal to the entries in $\mathbf{w}$.

At an early stage of the design, only block-level estimates of the currents are available. Since these blocks are large and may cover many nodes of the power grid, typically $k \ll n$. The matrix A is an incidence matrix that describes the distribution of block currents, with each column corresponding to a block, such that the sum of all entries in a column is one. Figure 2 shows a small illustrative example with the number of blocks being $k = 4$ and the number of power grid nodes being $n = 9$. The first column of A indicates that, the current drawn by block b_1 is distributed among node 4 and node 7 in the power grid, each with 50% and 50% of the total block current I_{b_1} respectively; the second column indicates that block b_2 is supplied by nodes 1, 2 and 5 by 30%, 30% and 40% respectively, etc. In reality, the block size is much larger: for instance, an SRAM would be distributed over hundreds of sink nodes for the power grid. In the early design stage, matrix A can be constructed by assuming uniform distribution of block currents among nodes of each block, or, if we have more specific knowledge of the structure of a block, certain patterns can be assumed in the corresponding column of matrix A.

Each entry in $\mathbf{I}$ could consist of contributions from more than one block, because each bottom-layer node in the power grid typically provides power for multiple logic gates that could belong to different functional modules. Therefore, different columns of matrix A can overlap with each other. Also, leakage current contributions can be considered as a block that is always on and contributes to every entry in $\mathbf{I}$.

If all blocks were always on, then all entries in $\mathbf{w}$ are 1. However, this is typically not the case: for instance, if it is known that the maximum operating power for a circuit is $P_{\max}$, which is less than the sum of the power dissipated by all blocks, then clearly, we know that all blocks cannot be on simultaneously. Therefore, realistically, $\mathbf{w}$ is a switch vector with $w_r = 1$ if a block is on and $w_r = 0$ otherwise. Different $\mathbf{w}$ vectors represent different working modes of the circuit, and hence model the source of uncertainty. In Figure 2, $\mathbf{w} = [1, 1, 0, 1]^{\mathrm{T}}$, describing a working mode that blocks b_1, b_2 and b_4 are active, while block b_3 is off. For Figure 1(b), $\mathbf{w} = [0, 1, 0, 1, 1]^{\mathrm{T}}$; for Figure 1(c), $\mathbf{w} = [1, 0, 0, 1, 1]^{\mathrm{T}}$.

Our approach is superficially similar to [6] in terms of using upper bounds to constrain the maximum current drawn. However, unlike that approach, which solves the power grid late in the design process when much more information is known, we solve it under early uncertain conditions. Also, [6] uses a U matrix to model current constraints provided by the designer, and formulates a continuous linear programming problem, where the variables are n normalized node voltages. On the other hand, in our approach, we account for the origin of uncertainty and use the k variables w_r as the 0-1 integer variables to be optimized.

The solution to the system of equations (1) is therefore:

$$\mathbf{X} = G^{-1} \cdot A \cdot \mathrm{diag}(\mathbf{w}) \cdot \mathbf{I_b} \tag{3}$$

Our objective is to find the vector $\mathbf{w}$ that causes the largest value in solution vector $\mathbf{X}$, either for its maximum entry or for the average of its entries, under certain constraints. The ith entry in equation (3) can be written as

$$x_i = \sum_{r=1}^{k} c_r w_r I_{b_r} \tag{4}$$

where k is the number of blocks; w_r is the rth diagonal entry of vector $\mathbf{w}$, with value 1 when the rth block is on, 0 when it is off; I_{b_r} is the rth entry of vector $\mathbf{I_b}$, i.e., the total current of the rth block; c_r's are constant coefficients from equation (3).

The power-limit constraint mentioned in Section 1 becomes:

$$\mathbf{w}^{\mathrm{T}} \cdot \mathbf{I_b} \leq \frac{P_{\max}}{V_{DD}} \tag{5}$$

A synchronization constraint of multiple blocks being on and off together can be incorporated by considering these blocks as one single block, although they might be physically apart from each other. An exclusivity constraint that specifies that at most m out of l blocks is active can be written under this notation as

$$w_{r_1} + w_{r_2} + \cdots + w_{r_l} \leq m \tag{6}$$

where $r_1, r_2, \cdots, r_l$ are indexes of those blocks.

The estimation problem can now be set up as an integer linear programming (ILP) problem as follows:

$$\begin{aligned} \max x_i &= \sum_{r=1}^{k} c_r w_r I_{b_r} \\ \text{subject to } \mathbf{w}^{\mathrm{T}} \cdot \mathbf{I_b} &\leq \frac{P_{\max}}{V_{DD}} \\ w_{r_1} + w_{r_2} + \cdots + w_{r_l} &\leq m \end{aligned} \tag{7}$$

Note that synchronization constraints are implicitly included in assigning the blocks.

So far we have been dealing with the situation where each block has only two modes: it is either off or consuming a current amount given by the corresponding entry in $\mathbf{I_b}$. Now if we consider the case where each block has multiple working modes, when some blocks are consuming maximum currents, others may be also on, but in a low-consumption working mode. This can be modeled by multiple $\mathbf{I_b}$ vectors that represent possible patterns. By constructing and solving an ILP problem (7) for each $\mathbf{I_b}$ vector, we have a set of worst case x_i values, and the largest one among them is the real worst case for this node.

Conceptually, the worst case working mode of the entire circuit can be determined as follows. After constructing and solving the ILP formulation for every entry in vector $\mathbf{X}$, a worst-case $\mathbf{w}$ vector can be found for every node in the circuit, as well as its worst-case voltage drop. Then, if we pick the largest among these voltage-drop values, the corresponding $\mathbf{w}$ vector is the worst-case working mode for the whole circuit, in terms of the largest single-node voltage drop. If we are interested in the average voltage drop, we can use the sum of equation (4) from all nodes as the object function, and solve the resulting ILP problem for the worst case $\mathbf{w}$ vector.

In early power grid performance estimation, G is the global supply network, which corresponds to the top two or three metal layers. Simulating industrial circuits shows that major voltage drop occurs at the top few metal layers, and therefore the voltages at second or third layer nodes have fidelity on what will happen in a complete design. In fact, for our benchmark in its complete design, a worst case DC analysis shows that the average voltage drop at the bottom layer is

15mV, and the average drop at the third metal layer is 8.6mV: 57% of the overall voltage drop is in the top three metal layers. Although this assumption will control the dimension of G, it will still be very large.

The large size of G is one reason that affects the evaluation of equation (4) since the coefficients c_r require a knowledge of G^{-1}, and are expensive to compute. Secondly, when the number of blocks is large, the dimension of $\mathbf{w}$ is correspondingly large, which implies that the number of integer variables may be prohibitive for an ILP solver. For these reasons, it is impractical to construct equation (4) for every node and use a ILP solver to find the exact solution.

In the next section, we propose a heuristic method to find a near-worst $\mathbf{w}$ vector.

III. Heuristic Solution

As mentioned earlier, there are two issues that need to be resolved in order to find a fast solution:

- The cost function, equation (4), needs to be constructed without the knowledge of G^{-1}. In other words, we need to find x_i, the voltage drop at node i, as a linear function of block currents, without inverting matrix G. One existing power grid analyzer, [10], is capable of such computation, and is the basis of the proposed algorithm.
- The worst-case $\mathbf{w}$ vector needs to be found without using an ILP solver. A greedy heuristic is used in the proposed algorithm for this purpose.

A. Approximated Cost Function

The proposed algorithm is built on the power grid analyzer of [10], which is a statistical algorithm with a complexity that is linear in circuit size, and it has the feature of localizing computation, which makes it useful for constructing the cost function (4) that is concerned with the voltage drop at only one node. The algorithm in [10] constructs a random walk "game" to model a circuit described by a linear equation set such as equation (1). A finite undirected connected graph representing a street map is constructed with the same topology as the circuit. A walker starts from one of the nodes, and goes to one of the adjacent nodes every day with a certain probability. The walker pays an amount of money to a motel for lodging everyday, until he/she reaches one of the homes, which are a subset of the nodes that correspond to the voltage sources in the circuit. If the walker reaches home, his/her journey is complete and he/she will be rewarded a certain amount of money. The problem is to find the gain function:

$$f(y) = E[\text{total money earned } | \text{walk starts at node } y] \tag{8}$$

It is proven that $f(y)$ is equal to the voltage at the corresponding node y in the power grid, if the game is constructed as follows: the price of the motel at node i is

$$m_i = \frac{I_i}{\sum_{j=1}^{\text{degree}(i)} g_j} \tag{9}$$

where I_i is the current load at node i, $\text{degree}(i)$ is the number of resistors connected to node i, and g_j's are the conductances of these resistors; the probability of going from node i to its q^{th} neighbor is

$$p_{i,\text{neighbor}\, q} = \frac{g_q}{\sum_{j=1}^{\text{degree}(i)} g_j}\ , \quad q = 1, 2, \cdots, \text{degree}(i) \tag{10}$$

and the award for reaching home is

$$f(\text{home}) = \text{the corresponding voltage source value} \tag{11}$$

Under these settings, a node voltage can be estimated by performing a certain number of walks and computing the average money left in those experiments [10].

In the case of equation (1), the award for reaching home is zero according to equation (11), and the estimated $f(y)$ is essentially the average motel expenses in one walk. Thus, the estimated voltage can be written as

$$V_{\text{estimate}} = \frac{\sum_{\text{motel}\, i} n_i m_i}{M} \tag{12}$$

where M is the number of random walks, n_i is the number of days that the walker has stayed in motel i, i.e., the number of times that walks pass node i. Applying equation (9), we can rewrite equation (12) as a linear function of current loads:

$$V_{\text{estimate}} = \frac{\sum_{\text{node}\, i} \alpha_i I_i}{M} \qquad \text{where } \alpha_i = \frac{n_i}{\sum_{j=1}^{\text{degree}(i)} g_j} \tag{13}$$

Then we substitute equation (2) into equation (13), and equation (13) becomes a linear combination of block currents:

$$V_{\text{estimate}} = \frac{\sum_{r=1}^{k} \beta_r w_r I_{b_r}}{M} \qquad \text{where } \beta_r = \sum_{\text{node}\, i} \alpha_i a_{i,r} \tag{14}$$

Here, w_r and I_{b_r} are as defined in equation (4), $a_{i,r}$ is the (i, r) entry of matrix A. Equation (14) is an approximation to (4), the cost function of the ILP formulation.

Intuitively, the above computation enumerate current paths and compute the influence of the blocks on the voltage drop at node y. Not all current paths are considered, and those with larger influence on the voltage drop are more likely to be chosen. The result, equation (14), essentially describes the importance of each block.

B. The Proposed Heuristic

Now the goal of the ILP problem is to find the $\mathbf{w}$ vector that maximizes equation (14) under the constraints (5)(6). Note that the coefficients $\{\beta_1, \beta_2, \cdots, \beta_k\}$ are weights in equation (14), in other words, they determine the influence of each block on the voltage drop at node y. Because the power-limit constraint (5) restricts the sum of active block currents, activating blocks with large β's is likely to cause large voltage drop at node y. (Note that it is likely, not guaranteed, as will be discussed later.) However, the "likely" may not be true when competing blocks are involved in an exclusivity constraint (6): for example, if only one of the two blocks b_1 and b_2 can be on at a time, and if $\beta_1 > \beta_2$, but $\beta_1 I_{b_1} < \beta_2 I_{b_2}$, then the choice is complicated.

The flow of proposed heuristic algorithm is as follows:

1) Run p random walks from node y, where p is an integer parameter. Instead of calculating the walk results, we keep track of the motels [corresponding to power grid nodes] visited.
2) By the procedure from equation (12) to (13) (14), obtain coefficients $\{\beta_1, \beta_2, \cdots, \beta_k\}$.
3) Sort $\{\beta_1, \beta_2, \cdots, \beta_k\}$. Repeat the above process until this sorted sequence does not change any more, according to a stopping criterion described at the end of this section.
4) Greedily activate blocks one by one. Each time, activate the block with the largest β coefficient, which does not violate constraint (5) or (6), and which falls into one of the following three categories:
   - The block has no exclusivity constraint.
   - The block has exclusivity constraint(s), but activating it does not close any constraint, i.e., the constraint(s) allows at least one more later block.
   - The block has exclusivity constraint(s), and it has the largest βI_b product among all inactive blocks that are involved in the constraint(s).

   The greedy heuristic stops when no more blocks can be added to the active block list.

For the example in Figure 1, if the node marked with the black dot is chosen to be node y, and the above algorithm is performed, the final β ordering from step 3 is $\{\beta_2, \beta_5, \beta_4, \beta_1, \beta_3\}$. Because there is no exclusivity constraint (6) in this case, step 4 activates block b_2, then block b_5, then block b_4, and stops because no more block can be added without violating (5). Thus, the pattern in Figure 1(b) is found. (Because the black-dot location is not known beforehand, the algorithm has to be performed for every node in order to find the largest single-node voltage drop.)

Note that the first three steps of the algorithm are independent of the actual current loads of the blocks. In the case where multiple $\mathbf{I_b}$ vectors are considered, the algorithm only needs to go through the first three steps once, and simply repeats step 4 for each $\mathbf{I_b}$ vector. Thus the extra computational cost is low.

The generated $\mathbf{w}$ vector is a near-worst-case $\mathbf{w}$ vector, in terms of the voltage drop at node y. When entries in vector $\mathbf{I_b}$ have different values, this problem is similar in flavor to the NP-hard bin-packing problem [2], in the sense of finding a number of blocks with various current consumptions to fit into a fixed power budget, although there is a different optimization goal that is to maximize another linear function (14) of the chosen block currents. And the proposed heuristic does not guarantee optimality. However, since at the floorplanning stage, entries in vector $\mathbf{I_b}$, i.e., block currents, have similar order of magnitude, it is likely that the degree of suboptimality is minor.

The above process provides a heuristic that aims to find the worst-case $\mathbf{w}$ vector for a specific node in the power grid. This procedure can be adapted for several global objectives:

- If the objective is to find the worst-case $\mathbf{w}$ vector that causes the largest *maximum* voltage drop in the whole circuit over all working modes, we can apply the heuristic to every node, and then, among the $\mathbf{w}$ vectors and voltage-drop values obtained, we pick the largest voltage-drop and its associated $\mathbf{w}$ vector.
- If the objective is to find the worst-case $\mathbf{w}$ vector that causes the largest *average* voltage drop in the circuit over all working modes, we can modify step 1 of the heuristic to run a random walk from every node in the circuit, and the outcome would be the near-worst-case $\mathbf{w}$ vector for average voltage drop.

In any case, a stopping criterion is required for step 3 of the proposed heuristic. Our implementation empirically chooses $p = 10$, i.e., we check convergence after every 10 random walks, and look at a portion of the sorted β sequence, starting from the largest β's, such that the sum of the corresponding block currents is equal or less than $\frac{2P_{\max}}{V_{DD}}$. If this portion of the sorted sequence does not change after 10 walks, the algorithm claims convergence. When the number of blocks is large, it takes a long time to converge when there is no change in the sorted sequence. However, our primary interest is which blocks are active, and not the precise significance ranking of each block. Therefore, we loosen the stopping criterion to save unnecessary runtime, by defining a tolerance as $T = \lceil \frac{k}{20} \rceil$. If the position change of every block after 10 walks is less than T, the algorithm claims convergence. This stopping criterion is different from [10] in that the convergence of the estimated voltage drop value is not checked. The reason is that the goal of the heuristic is to quickly find the worst working mode for each node and report possible violations, while the accurate voltage drop values at critical nodes can later be obtained by any linear solver.

IV. Results

We use an industrial power grid, GSRC floorplans [4], and MCNC floorplans [9] to evaluate the proposed heuristic. The results are compared against exact solutions produced by an ILP solver, and results from a pessimistic analysis. All computations are carried out on a 2.8GHz P4-based Linux workstation.

Our power-grid benchmark is the top three layers of an industrial VDD net. It has 43,473 nodes, among which 19,395 third-layer nodes are to be analyzed. The total power is 26W if all circuit components are switching and consume maximum current, which includes 8W of leakage power that is assumed to be always on. The actual power limit is assumed to be 16W; in other words, since this includes 8W of leakage power, this implies that the active blocks cannot consume more than 8W switching current. The nominal VDD is 1.2V. Six GSRC floorplans [4] and five MCNC floorplans [9] are mapped onto this power grid, and third-layer current loads are grouped into blocks accordingly in each floorplan. Block boundaries are adjusted such that there are no white space with uncovered current loads. After we obtain one $\mathbf{I_b}$ vector for each floorplan, by multiplying random numbers between 0.5 and 1.5 to the entries of $\mathbf{I_b}$, we generate four extra $\mathbf{I_b}$ patterns for each of the six cases. Because we are unable to obtain functional description of floorplan blocks, we arbitrarily

TABLE I

Comparison of analysis methods for the largest single-point voltage-drop.

Floorplan	Number of blocks	Heuristic runtime(s)	Heuristic result(mV)	ILP exact result(mV)	Pessimistic result(mV)
n10a	10	48.14	234.4	234.4	250.8
n30a	30	48.79	274.2	274.3	304.3
n50a	50	49.44	190.8	191.0	247.4
n100a	100	51.05	223.4	223.5	236.7
n200a	200	55.87	238.2	240.4	266.0
n300	300	63.38	282.3	283.4	303.4
apte	9	48.18	193.2	193.3	214.5
xerox	10	48.13	225.1	225.1	243.1
hp	11	48.44	225.7	226.0	278.3
ami33	33	48.70	245.6	245.6	279.1
ami49	49	49.19	239.9	240.0	269.3

assign exclusivity constraints. The number of constraints for each floorplan is up to 21.

The comparison of largest single-point voltage-drop analysis using different strategies is shown in Table I, where the first six benchmarks are GSRC floorplans, and the rest are MCNC floorplans. In order to study the performance of the proposed heuristic method in finding $\mathbf{w}$ vector without interference of error from any other estimation step, we substitute the produced $\mathbf{w}$ into equation (3), use a direct linear solver to solve (3), and list the maximum entry of the solution vector in the fourth column of Table I. For this circuit size, it is already impractical to construct and evaluate the ILP equation (4) for every node. The ILP results listed in the fifth column are the exact answers by ILP analysis for the 50 highest-drop nodes found by the proposed heuristic. The last column is the result by solving equation (3) assuming $\mathrm{diag}(\mathbf{w})$ to be an identity matrix, i.e., assuming that all blocks are active. All three methodologies consider the five $\mathbf{I_b}$ patterns for each floorplan, and report the worst among the five results.

In Table I, there are noticeable differences between constrained analysis and pessimistic analysis. This difference depends on the details of the most power-intensive region of the chip. For floorplan n50a, the largest voltage-drop node happens to be close to a corner of its block, and these two neighboring blocks have an exclusivity constraint. Consequently, we see a 56mV overestimate by the pessimistic analysis. Although there may not be an exclusivity constraint in the power-intensive region of every chip design, the possibility of existence of such constraints makes the proposed heuristic superior to pessimistic analysis, thus avoiding false alarms, and consequently, ensuring that routing resources are not wasted. Figure 3 shows the working mode corresponding to the $\mathbf{w}$ vector found by the proposed heuristic for floorplan n30a.

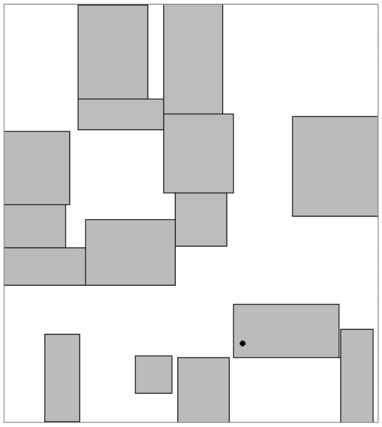


Fig. 3. Near-worst working mode of GSRC floorplan n30a found by the proposed heuristic. The black dot marks location of the largest voltage drop.

TABLE II

Numbers of node-voltage violations reported by the proposed heuristic and the pessimistic analysis.

Floorplan	n10a	n30a	n50a	n100a	n200a	n300
Heuristic	113	122	44	71	102	91
Pessimistic	181	163	72	95	124	137
Floorplan	apte	xerox	hp	ami33	ami49	
Heuristic	91	120	109	130	91	
Pessimistic	112	133	150	184	106	

Table II shows the comparison of number of node-voltage violations reported by different strategies, when the voltage-drop threshold is 80mV. In most cases, about one third of the violating nodes reported by pessimistic analysis are found legal by the proposed heuristic.

Table III shows the comparison of average voltage-drop analysis using different strategies. All results are for the average of 19,395 third-layer nodes. In this case, because only one ILP is required to be formulated and solved for each floorplan with each $\mathbf{I_b}$ vector, the fifth column is the exact solution.

In both Table I and Table III, results from the proposed heuristic correlate well with those from ILP solver. The difference between the two solutions is due to the fact that the proposed heuristic finds only a near-worst case, and does make mistakes on certain not-very-significant blocks. Consequently, the results are optimistic, compared with the ILP solution. One remedy is to use the power budget $P_{\max}(1+\delta)$ instead of $P_{\max}$, where δ is a small positive value, but this is not included in our implementation.

V. Conclusion and Extension

The problem of early-stage power grid analysis under the uncertainty of different working modes is investigated in this paper. A random-walk based heuristic algorithm is proposed to find the worst-case scenario. The method is tested on industrial circuits and is demonstrated to find the near-worst-case working mode with low runtime.

The proposed heuristic framework can be extended to perform electromigration (EM) check. To find the worst-case current density in a wire segment AB, we use the first two steps of the heuristic to find the value $(\beta_A - \beta_B)$ for each block; in step 3, we sort the positive ones of the $(\beta_A - \beta_B)$

TABLE III
COMPARISON OF ANALYSIS METHODS FOR THE LARGEST AVERAGE VOLTAGE-DROP.

Floorplan	Number of blocks	Heuristic runtime(s)	Heuristic result(mV)	ILP exact result(mV)	Pessimistic result(mV)
n10a	10	36.17	6.81	7.57	14.49
n30a	30	46.47	7.72	7.73	13.59
n50a	50	51.66	7.24	7.66	12.76
n100a	100	56.69	6.99	7.80	12.86
n200a	200	66.99	7.55	7.85	12.88
n300	300	67.03	7.47	7.86	12.96
apte	9	30.87	7.33	7.58	13.80
xerox	10	31.09	7.51	7.55	13.66
hp	11	36.12	7.43	7.60	14.73
ami33	33	36.21	7.34	7.69	13.71
ami49	49	46.41	7.43	7.68	12.99

values; then the outcome of step 4 would be the near-worse-case working mode that causes the largest current from node A to node B. Because EM check needs to be performed for both directions, the same procedure should be done for $(\beta_B - \beta_A)$ values as well.

REFERENCES

[1] T. Chen and C. C. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 559–562, 2001.

[2] E. G. Jr. Coffman, M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin packing: a survey," *Approximation Algorithms for NP-Hard Problems*, pp. 46–93. Boston, MA: PWS Publishing, 1997.

[3] J. P. Eckhardt and K. A. Jenkins, "PLL phase error and power supply noise," in *IEEE 7th Topical Meeting on Electrical Performance of Electronic Packaging*, pp. 73–76, 1998.

[4] GSRC Floorplan Benchmarks. [Online]. Available: http://www.cse.ucsc.edu/research/surf/GSRC/progress.html

[5] C. Ho, A. E. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 6, pp. 504–509, Jun. 1975.

[6] D. Kouroussis and F. N. Najm, "A static pattern-independent technique for power grid voltage integrity verification," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 99–104, 2003.

[7] J. A. Jr. Kowaleski, T. Truex, D. Dever, D. Ament, W. Anderson, L. Bair, S. Bakke, D. Bertucci, R. Castelino, D. Clay, J. Clouser, A. DiPace, V. Germini, R. Hokinson, C. Houghton, H. Kolk, B. Miller, G. Moyer, R. O. Mueller, N. O'Neill, D. A. Ramey, Y. Seok, J. Sun, G. Zelic, and V. Zlatkovic, "Implementation of an Alpha microprocessor in SOI," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 248–249, 2003.

[8] J. Kozhaya, S. R. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," *IEEE Transactions on Computer-Aided Design*, vol. 21, no. 10, pp. 1148–1160, Oct. 2002.

[9] MCNC Floorplan Benchmark Suite. [Online]. Available: http://www.cse.ucsc.edu/research/surf/GSRC/MCNC

[10] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Random walks in a supply network," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 93–98, 2003.

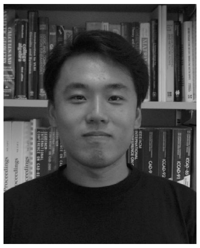
[11] H. Qian and S. S. Sapatnekar, "Hierarchical random-walk algorithms for power grid analysis," in *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 499–504, 2004.

[12] S. S. Sapatnekar and H. Su, "Analysis and optimization of power grids," *IEEE Design and Test of Computers*, vol. 20, no. 3, pp. 7–15, May/Jun. 2003.

[13] J. Stinson and S. Rusu, "A 1.5 GHz third generation Itanium processor," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 252–253, 2003.

[14] H. Su, K. H. Gala, and S. S. Sapatnekar, "Fast analysis and optimization of power/ground networks," in *ACM/IEEE International Conference on Computer-Aided Design Digest of Technical Papers*, pp. 477–480, 2000.

[15] M. Zhao, R. V. Panda, S. S. Sapatnekar, and D. Blaauw, "Hierarchical analysis of power distribution networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 2, pp. 159–168, Feb. 2002.



Haifeng Qian received the B.E. degree in microelectronics from Tsinghua University, Beijing, China, in 2000, and the M.S. degree in electrical engineering from the University of Texas at Dallas in 2002. Since August 2002, he has been a Ph.D. student in the University of Minnesota. He received a best paper award at DAC2003. In the summer of 2004, he worked at Synopsys, Sunnyvale, CA.

Sani R. Nassif received his Ph.D. from Carnegie-Mellon University in the eighties. 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 in 1996 where he is presently managing the tools and technology department, which is focused on design/technology coupling, timing simulation and analysis, testing, low power design and thermo-electric cooling.

Sachin S. Sapatnekar received the B.Tech. degree from the Indian Institute of Technology, Bombay in 1987, the M.S. degree from Syracuse University in 1989, and the Ph.D. degree from the University of Illinois at 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. He is currently the Robert and Marjorie Henle Professor in the Department of Electrical and Computer Engineering at the University of Minnesota.

He is an author of four books and a coeditor of one volume, and has published mostly in the areas of timing and layout. He has held positions on the editorial board of the IEEE Transactions on VLSI Systems, and the IEEE Transactions on Circuits and Systems II, the IEEE Transactions on CAD, and has been a Guest Editor for the latter. He has served on the Technical Program Committee for various conferences, and as Technical Program and General Chair for the Tau workshop and for the International Symposium on Physical Design. He has been a Distinguished Visitor for the IEEE Computer Society and a Distinguished Lecturer for the IEEE Circuits and Systems Society. He is a recipient of the NSF Career Award, three best paper awards at DAC and one at ICCD, and the SRC Technical Excellence award. He is a fellow of the IEEE.