

Early-stage Power Grid Analysis for Uncertain Working Modes

Haifeng Qian
Department of ECE
University of Minnesota
Minneapolis, MN 55414
qianhf@ece.umn.edu

Sani R. Nassif
IBM Austin Research Labs
11400 Burnet Rd. MS/9460
Austin, TX 78758
nassif@us.ibm.com

Sachin S. Sapatnekar
Department of ECE
University of Minnesota
Minneapolis, MN 55414
sachin@ece.umn.edu

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 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 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.

Categories and Subject Descriptors

B.7.2 [Integrated Circuit]: Design Aids; B.8.2 [Performance and Reliability]: Performance Analysis and Design Aids

General Terms

Algorithms, Performance, Design, Reliability

Keywords

Power grid, Supply network, Random walk, Early estimation

1. INTRODUCTION

Power grid noise has become an increasing 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 [5][6][7]. All of these 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, [4] 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 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. 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. The number of working modes for the circuit may be

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD'04, April 18–21, 2004, Phoenix, Arizona, USA.
Copyright 2004 ACM 1-58113-817-2/04/0002 ...\$5.00.

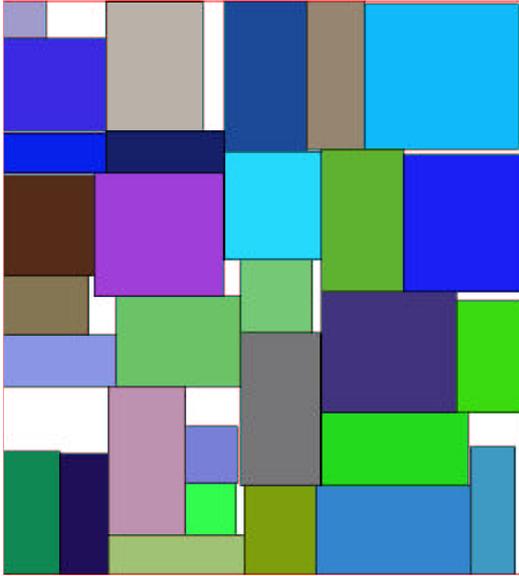


Figure 1: GSRC floorplan n30a.

very large (potentially exponential in the number of blocks), and it is often not possible to enumerate all such modes. Figure 1 illustrates a GSRC floorplan [2], a circuit consisting of 30 functional modules. In different working modes, some of the blocks are active and consuming current, while others are standing by.

One way to deal with this uncertainty is to perform a worst-case analysis assuming every block is on. This is clearly too pessimistic and produces false alarms, since such a working mode may never occur. Instead, we use some constraints to limit the analysis to working modes that are more likely to occur, and find the worst one 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} .
- 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 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 [1]. 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.

2. 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 [3]:

$$GX = \mathbf{I} \quad (1)$$

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:

$$\mathbf{I} = A\mathbf{J}\mathbf{I}_b \quad (2)$$

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, and J is a k -by- k matrix.

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. For example, if the non-zero entries in a column of A is $\{0.1, 0.7, 0.2\}$, the current drawn by this block is distributed among three nodes in power grid, each with 10%, 70% and 20% of the total current respectively. In reality, the block size is much larger than three: 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. 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 J would be an identity matrix. 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, J is a diagonal matrix with $j_{r,r} = 1$ if a block is on and $j_{r,r} = 0$ otherwise. Different J matrices represent different working modes of the circuit, and hence model the source of uncertainty.

Our approach is superficially similar to [4] 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, [4] uses a U matrix to model current constraints provided by the designer, and formulates a continuous linear programming problem. On the other hand, in our approach, we account for the origin of uncertainty and use $j_{r,r}$ as the 0-1 integer variables to be optimized.

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

$$\mathbf{X} = G^{-1}A\mathbf{J}\mathbf{I}_b \quad (3)$$

Our objective is to find the matrix J 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 i th entry in equation (3) can be written as

$$x_i = \sum_{r=1}^k c_r j_{r,r} b_r \quad (4)$$

where k is the number of blocks; $j_{r,r}$ is the r th diagonal entry of matrix J , with value 1 when the r th block is on, 0 when it is off; b_r is the r th entry of vector \mathbf{I}_b , i.e., the total current of the r th block; c_r 's are constant coefficients from equation (3).

The power-limit constraint mentioned in Section 1 becomes:

$$\sum_{i=1}^n (\mathbf{J}\mathbf{I}_b)_i \leq \frac{P_{max}}{V_{DD}} \quad (5)$$

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 only one out of l blocks is active can be written under this notation as

$$j_{r_1, r_1} + j_{r_2, r_2} + \dots + j_{r_l, r_l} \leq 1 \quad (6)$$

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

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

$$\begin{aligned} \max x_i &= \sum_{r=1}^k c_r j_{r,r} b_r \quad (7) \\ \text{subject to } \sum_{i=1}^n (\mathbf{J}\mathbf{I}_b)_i &\leq \frac{P_{max}}{V_{DD}} \\ j_{r_1, r_1} + j_{r_2, r_2} + \dots + j_{r_l, r_l} &\leq 1 \end{aligned}$$

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 is 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 J matrix 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 J matrix 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 J matrix.

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, the average drop at the third metal layer is 8.6mV. 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 J 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 J matrix.

3. HEURISTIC SOLUTION

The proposed algorithm is built on the power grid analyzer of [6], which is a statistical algorithm with complexity linear in circuit size, and it has the feature of localizing computation. This makes it especially useful when only part of the grid is to be analyzed.

The algorithm in [6] constructs a random walk "game" to model a linear equation set such as equation (1). Given a finite undirected connected graph representing a street map, 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. 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:

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

It is proven that $f(x)$ is equal to the voltage at the corresponding node x in the power grid. Therefore a node voltage can be estimated by performing a certain number of walks and computing the average money left in those experiments [6].

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

$$V_{estimate} = \frac{\sum_{motel\ i} n_i m_i}{M} \quad (9)$$

where M is the number of random walks, m_i is the price of

motel i , n_i is the number of days that the walker has stayed in motel i . The price of a motel is a function of the current load at the corresponding node in the power grid, as shown below:

$$m_i = \frac{I_i}{\sum_{j=1}^{degree(i)} g_j} \quad (10)$$

where I_i is the current load at node i , $degree(i)$ is the number of resistors connected to node i , and g_j 's are the conductances of these resistors. Therefore, we can rewrite equation (9) as a linear function of current loads:

$$V_{estimate} = \frac{\sum_{node\ i} \alpha_i I_i}{M} \quad (11)$$

where $\alpha_i = \frac{n_i}{\sum_{j=1}^{degree(i)} g_j}$

Then we substitute equation (2) into equation (11), and equation (11) becomes a linear combination of block currents:

$$V_{estimate} = \frac{\sum_{r=1}^k \beta_r j_{r,r} b_r}{M} \quad (12)$$

where $\beta_r = \sum_{node\ i} \alpha_i a_{i,r}$

Here, $j_{r,r}$ and b_r are as defined in equation (4), $a_{i,r}$ is the (i, r) entry of matrix A . Equation (12) is an approximation to (4).

The flow of proposed heuristic algorithm is as follows:

1. Run p random walks from node x , 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 (9) to (11) (12), obtain coefficients $\{\beta_1, \beta_2, \dots, \beta_k\}$.
3. Sort $\{\beta_1, \beta_2, \dots, \beta_k\}$. 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 that does not violate constraint (5) or (6), until no more blocks can be added to the active block list.

Note that the first three steps 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 J matrix is a near-worst-case J matrix, in terms of the voltage drop at node x . When entries in vector \mathbf{I}_b have different values, this problem is similar in flavor to the bin-packing problem, and the proposed heuristic does not guarantee optimality. However, since entries in vector \mathbf{I}_b 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 J matrix 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 J matrix 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 J matrices and voltage-drop values obtained, we pick the largest voltage-drop and its associated J matrix.
- If the objective is to find the worst-case J matrix 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 J matrix for average voltage drop.

In any case, a stopping criterion is required for step 3 of the proposed heuristic. In our implementation, $p = 10$, 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.

4. RESULTS

We use an industrial power grid and GSRC floorplans [2] 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 [2] 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 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 1. In order to study the performance of the proposed heuristic method in finding J matrix without interference of error from any other estimation step, we substitute the produced J 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 1: 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 | 48.33 | 234.4 | 234.4 | 250.8 |
| n30a | 30 | 49.00 | 274.2 | 274.3 | 304.3 |
| n50a | 50 | 49.53 | 190.9 | 191.0 | 247.4 |
| n100a | 100 | 51.42 | 223.3 | 223.5 | 236.7 |
| n200a | 200 | 56.66 | 238.2 | 240.4 | 266.0 |
| n300 | 300 | 66.38 | 282.1 | 283.4 | 303.4 |

Table 2: Numbers of node-voltage violations reported by the proposed heuristic and the pessimistic analysis.

| Floorplan | n10a | n30a | n50a | n100a | n200a | n300 |
|-------------|------|------|------|-------|-------|------|
| Heuristic | 113 | 120 | 44 | 69 | 91 | 91 |
| Pessimistic | 181 | 163 | 72 | 95 | 124 | 137 |

Table 1. 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 J 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.

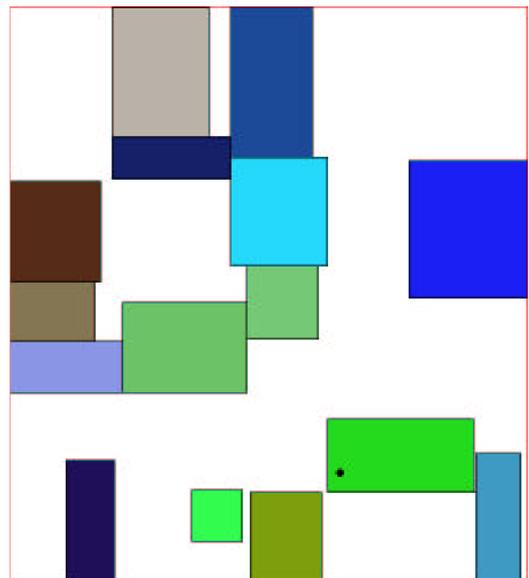
In Table 1, there are noticeable differences between constrained analysis and pessimistic analysis. This difference depends on the details of the most power-intensive region of the chip. For floorplan n50a, the largest voltage-drop node happens to be close to a corner of its block, and these two neighboring 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 2 shows the working mode corresponding to the J matrix found by the proposed heuristic for floorplan n30a.

Table 2 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 3 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 1 and Table 3, results from the proposed heuristic correlate well with those from ILP solver.

5. CONCLUSION

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.

**Figure 2: Near-worst working mode of GSRC floorplan n30a found by the proposed heuristic. The black dot marks location of the largest voltage drop.**

6. REFERENCES

- [1] J. P. Eckhardt and K. A. Jenkins. PLL phase error and power supply noise. *IEEE 7th Topical Meeting on Electrical Performance of Electronic Packaging*, pages 73–76, 1998.
- [2] GSRC Floorplan Benchmarks, available at www.cse.ucsc.edu/research/surf/GSRC/progress.html
- [3] C. Ho, A. E. Ruehli, and P. Brennan. The modified nodal approach to network analysis. *IEEE Transactions on Circuits and Systems*, 22(6):504–509, 1975.
- [4] D. Kouroussis and F. N. Najm. A static pattern-independent technique for power grid voltage integrity verification. *Proceedings of the ACM/IEEE Design Automation Conference*, pages 99–104, 2003.
- [5] J. Kozhaya, S. R. Nassif, and F. N. Najm. A multigrid-like technique for power grid analysis. *IEEE Transactions on Computer-Aided Design*, 21(10):1148–1160, 2002.

Table 3: 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 | 40.29 | 7.47 | 7.57 | 14.49 |
| n30a | 30 | 51.80 | 7.72 | 7.73 | 13.59 |
| n50a | 50 | 57.82 | 7.06 | 7.66 | 12.76 |
| n100a | 100 | 51.91 | 7.38 | 7.80 | 12.86 |
| n200a | 200 | 69.37 | 7.22 | 7.85 | 12.88 |
| n300 | 300 | 69.48 | 7.65 | 7.86 | 12.96 |

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

[7] 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*, 21(2):159–168, 2002.