Fast Placement Optimization of Power Supply Pads *

Yu Zhong
Dept. of Electrical and Computer Engineering
Univ. of Illinois at Urbana-Champaign
Urbana, IL, 61801
e-mail: yuzhong@uiuc.edu

Martin D. F. Wong
Dept. of Electrical and Computer Engineering
Univ. of Illinois at Urbana-Champaign
Urbana, IL, 61801
e-mail: mdfwong@uiuc.edu

Abstract— Power grid networks in VLSI circuits are required to provide adequate input supply to ensure reliable performance. In this paper, we propose algorithms to find the placement of power pads that minimize not only the worst voltage drop but also the voltage deviation across the power grid. Our algorithm uses simulated annealing to minimize the total cost of voltage drops. The key enabler for efficient optimization is a fast localized node-based iterative method to compute the voltages after each movement of pads. Experimental results show that our algorithm demonstrates good runtime characteristics for power grids with large numbers of pad candidates in multi-million-size circuits. For a 16-million-node power grid with 646 thousand pad candidates, our algorithm took 72 minutes to improve the worst voltage drop from $0.398V$ to $0.196V$ and reduce the deviation of voltages on the power grid from $0.134V$ to $0.024V$.

I. Introduction

IC power distribution systems are designed to provide necessary voltages and currents to the transistors that perform the logic functions of a chip. The supply voltages are assumed to be constant across the chip to ensure reliable performance. The voltage values on the power grids fluctuate due to increased resistances of metal lines, high current levels, and package pin inductances. The resulting IR drop on the grid reduces noise margin and increases gate delay, which causes a serious performance impact [1], [2].

However, with the rapid increase in the complexity of very large scale integration (VLSI) circuits, the design and analysis of power grids have become a challenging task. When the network is very large, typically 1 million to 100 million nodes, even DC analysis becomes a critical issue of design due to limitations of the computational resources (e.g., runtime and memory usage). The design of power grid becomes even more difficult due to the bottleneck of simulation. Recently, [8] proposed a node-based iterative method for fast simulation of multi-million size power grids. To reduce the impact of IR drop on power grids, research has been directed in different ways. The simplest approach is to widen the lines that experience the largest voltage drops, since increasing the width decreases the resistance and hence the IR drop [6], [4], [5]. However, this may not always be possible due to the constraints in the routing area.

In this paper, we present a more aggressive solution to reduce the IR drop: we try to find an optimal placement of a set of power pads such that the IR drop is minimized on the power grid. The number of power pads required for a chip depends on various factors such as the size of the design, its power consumption, and the design of the power network [7]. Power pad placement is a difficult problem, because the number of candidate pad locations can be extremely large. In the case of a wire-bond package, the candidate locations are all possible pad locations on the peripheral power ring. For a high performance processor using a flip-chip package, a ball-grid array (called C4 bumps) forms the candidate set, where the power supply connections can be at various points within the chip. One of the most important advantages of C4 pads is the ability to place power and ground pads anywhere on the die to reduce the IR drop, as opposed to just the periphery.

The previous works related to power pad optimization include [3] and [7]. In [3], a heuristic for pad assignment and power routing is given, but only multi-tree topologies are considered. In [7], the authors proposed a mixed integer linear program (MILP) using macromodeling techniques to minimize the number of power pads. They selected a set of nodes as observation nodes to represent the worst voltage drop. However, the worst voltage locations can shift with change of pad locations, so the calculation of worst voltage drops is not accurate. Moreover, the number of power pad candidate locations is also extremely large as discussed above. MILP is very expensive for large number of integer variables, which is unfortunately equal to the number of pad candidate locations. Consequently the method in [7] is not efficient for large number of pad candidates.

We propose a method in this paper to find the optimal positions for a set of power pads, such that not only the worst IR drop is reduced, but also the IR drop over the whole power grid becomes more uniform. Given a fixed number of power pads and a set of pad candidate locations, we first use the method in [8] to compute the voltage values on the power grid for the initial placement of power pads. Then, we pick one power pad to move to another available candidate location and recompute the voltage values on the power grid. In this paper, we propose an efficient localized node-based iterative method to recompute the voltage values efficiently after each power pad movement. This computation is localized, and its time complexity is independent of the total power grid size. Based on this fast voltage recomputation methodology, we develop a simulated annealing algorithm to minimize the voltage drops on the power grid. The algorithm shows good scalability characteristics for large power grids with large numbers of pad candidate locations. Note that it is impractical and prohibitive to use the method of [8] directly to recompute the voltage values after each movement. It will take 26 minutes at each power pad movement for a 16-million-node problem, and the total runtime of simulated

*This work was partially supported by the National Science Foundation under grant CCR-0306244.

1-4244-0630-7/07/$20.00 ©2007 IEEE.

annealing will be 180 days.

The rest of the paper is organized as follows. In section 2, we first give a review of the improved node-based method in [8]. Then in section 3, we describe an efficient implementation of node-based iterative method to recompute the voltage values after each power pad movement. In section 4, we propose a simulated annealing algorithm to optimize the placement of power pads. Finally, we present our experimental results in section 5 to demonstrate the effectiveness of this algorithm.

II. Node-based Iterative Method

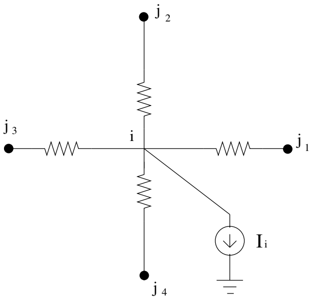

Fig. 1. A representative node in the power grid.

Due to the structure of power grids, iterative methods turn out to be good solution methodologies. In this section, we give a brief review of the efficient iterative algorithms proposed in [8]. The power grid model consists of wire resistances, Vdd pads, and current sources that represent the currents drawn by logic gates and functional blocks. If we apply Kirchoff's current law on a single node i in the power grid, as shown in Figure 1, we obtain voltage at node i as

$$V_i = \frac{\sum_{j \in N_i} g_{ij} V_j}{\sum_{j \in N_i} g_{ij}} - \frac{I_i}{\sum_{j \in N_i} g_{ij}} \quad (1)$$

Here I_i is the current drain at node i, N_i is the set of nodes adjacent to node i, and g_{ij} is the conductance between the two neighboring nodes i and j. The generic node-based method is defined as follows. Pick a node i in the power grid and update its voltage V_i according to Equation (1). Iteratively update the node voltages one node at a time until it converges to the exact solution.

The authors of [8] also present the improved node-based method, of which rate of convergence is an order of magnitude faster than the generic node-based method. The main iteration formula is

$$V_i^{(k+1)} = \omega \bar{V}_i^{(k+1)} + (1-\omega) V_i^{(k)} \quad (2)$$

where $\bar{V}_i$ denotes a generic node-based iteration as in Equation (1), and ω is the extrapolation factor. For a given initial placement of power pads, we use the improved node-based iterative method to compute the voltages in the power grid.

In Section 4, we will propose a simulated annealing based power pad placement algorithm. In this algorithm, whenever we change the location of a power pad, we need to recompute the voltages to find the effect of the move on the IR drops. If we use the method in [8] directly, it takes about 26 minutes to recompute the voltages after one move for a 16-million-node problem. If the total number of movements in simulated annealing is 10000 (which is less than what we have observed in our experiments), then the total runtime will be 180 days. In the next section, we propose a fast and efficient methodology to recompute voltage values after power pad movements.

III. Power Pad Location Update

After computing the initial voltages on the power grid, we start changing the locations of power pads to minimize the voltage drops. In this section, we focus on the basic operation of moving a power pad on a candidate grid structure, and the fast recomputation of voltage values. This operation will be the basis of the simulated annealing algorithm we propose in Section 4 to minimize the voltage drops.

A. Power Pad Candidate Locations

Based on the ball grid array structure (called C4 bumps), we model the potential power pad location candidates as a coarse grid, as shown in Figure 2. Note that our methods are not limited to a grid structure, but can be applied to arbitrary patterns of candidate locations such as peripheral power ring.

In Figure 2, the white dots represent the power pad candidate locations, which form an underlying coarse grid, and the black dots represent the current locations of the power pads. We can only move the power pads to the white dots, which are the power pad candidate locations. After a power pad is moved from its current location to another candidate location, the voltage values in the power grid need to be recomputed. In the following subsection, we describe how to recompute voltages in a fast and effective way.

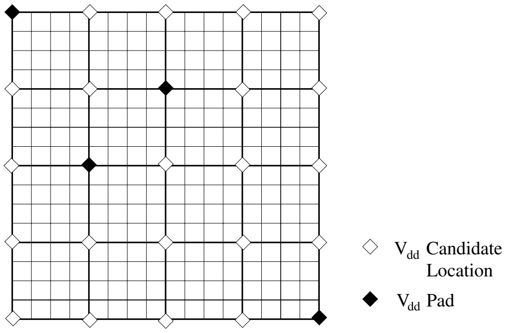

Fig. 2. An example of the power pad candidate locations.

B. Fast Voltage Update

Since the node-based iterative methods described in Section 2 are effective in voltage computations of power grids, we use these methods to update the voltage values after each power pad movement.

The movement of a power pad can be decomposed into two parts: deleting a Vdd pad from its old location at node x and adding it to a new location at node y. For simplicity, we first discuss how to compute the voltage change if we delete a Vdd pad from node x. The computation of adding a Vdd pad is similar.

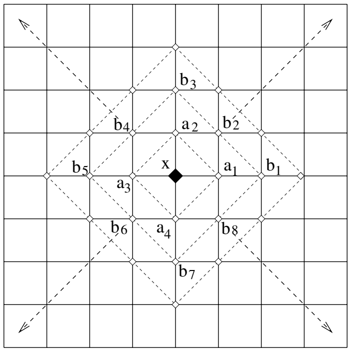

Fig. 3. An illustration of breadth first traversal for changing a Vdd pad at node x.

In node-based iterative method, the voltage at one node is determined by its neighbors as in Equation (2). If we delete a Vdd pad at node x, this action gives the immediate voltage change at node x, of which new voltage can be computed by Equation (1). This voltage change at node x will first influence its direct neighbors $a \in N_x$, where N_x is the set of nodes adjacent to node x. Then the voltage changes at nodes $a \in N_x$ will continuously influence the voltages on nodes $b \in N_a$, where N_a is the set of nodes adjacent to nodes a, and so on. In other words, the voltage change at source node x will propagate out like a wave on the power grid until it covers all the nodes on the grid. Then it is intuitive to recompute the new voltages of nodes in the order of this wave propagation. Here, we use breadth first traversal to visit the neighbors of node x as shown in Figure 3, and recompute their voltages. We use improved node-based method in Equation (2) to iteratively update those node voltages from the source node x in the order of wave propagation until it converges to the exact solution. Now, it is straightforward to extend this method to update the voltages when we move one Vdd pad from node x to node y. The only difference here is that there are two point sources x and y in the breadth first traversal described above.

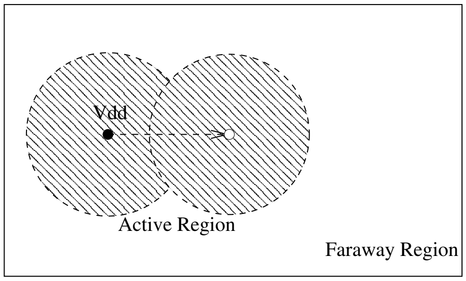

Fig. 4. Localized voltage computation after one power pad movement. The active region is defined to contain nodes that have voltage change more than error bound δ_{active} at current iteration.

The efficiency of updating voltages at each movement can be improved by taking advantage of the property of localization for node-based iterative method. Since the change at source node x or y will propagate out and vanish after some distance, the computation can be limited to a small region because of the inherent locality of the problem. In Figure 4, the shaded circles represent the active region after a power pad is moved from one point to another. Since the nodes that are far away from the old and new locations of this Vdd pad are unlikely to be influenced by this movement, we do not need to recompute their voltages. If the old and new locations of the Vdd pad are close to each other, the active region of deleting and adding a power pad may have some overlapping region, and the number of nodes which we need to recompute will become even smaller.

```
Voltage Recomputation after One Pad Movement
// $x, y$: The old and new locations of power pad
// $V^{(0)}$: The initial voltage values for all nodes
// $\epsilon \ll 1$: error bound used to control number of iterations
// $\delta_{active} \ll 1$: error bound used to determine active region
  $k = 0$ // iteration number
  repeat //begin iteration
    k = k+1
    $nodeQ$=empty first-in first-out queue
    $nodeQ$=$nodeQ \cup x \cup y$
    repeat // begin breadth first traversal
      $i$ = $nodeQ$.extractFirst()
      Compute $V_i^{(k)}$ using Equation (2)
      if $|V_i^{(k)} - V_i^{(k-1)}| > \delta_{active}$
        for each neighbor node $j \in N_i$ and $j \notin nodeQ$
          $nodeQ = nodeQ.pushback(j)$
    until $nodeQ$ is empty traversal
  until $max_i |V_i^{(k)} - V_i^{(k-1)}| < \epsilon$
```

Fig. 5. The algorithm of fast voltage computation after one power pad movement.

Note that we do not limit the size of the active region to a fixed number, since it may change under different conditions. Instead, we set up some error bound $\delta_{active} \ll 1$ to check if the wave propagation vanishes or not. In one iteration, if the voltage change at some node is smaller than this error bound, $\delta V < \delta_{active}$, we terminate this iteration and start a new iteration from the wave sources x and y again. The algorithm of fast update voltages for one pad movement is shown in Figure 5.

The runtime of each movement is independent of the size of the power grid, since the number of nodes to update is always limited to the active region. So, the method we propose to recompute voltages after one pad movement is efficient and scalable for large problems. This method is especially well-suited to be used within a simulated algorithm framework to optimize power pad placements, as will be discussed in the next section.

IV. Power Pad Placement Optimization

We propose a simulated annealing algorithm in this section to optimize the power pad placements with the objective of minimizing voltage drops. The temperature schedule is of the form $T_m = r \times T_{m-1}$, where m is the temperature index in simulated annealing. A typical value for r is 0.85. At each temperature, a number of power pad movements are attempted.

A. Cost Function

Our goal of design is not only minimizing the worst voltage drop in the power grid, but also reducing the standard deviation of voltage drops so that the voltage values on the whole power grid will become more uniform. The standard deviation of the voltages is defined as $\sigma = \sqrt{\sum_i (V_i - \mu)^2 / N}$, where V_i is the voltage at node i, N is the number of nodes, and $\mu = \sum_i V_i / N$ is the average voltage.

Then, the objective function is defined as

$$C(\mathbf{V}) = \alpha V_{worst}^2 + \beta \sum_i V_i^2 / N \qquad (3)$$

where V_i is the voltage at node i, N is the number of nodes on the power grid, and V_{worst} is the worst voltage drop. Here, α and β are constants to make a tradeoff between the worst voltage drop and the deviation in voltage drops. The advantage of choosing V_i^2 instead of V_i is that more effort will be given to reduce the larger voltage drops. Consequently, the voltage values on the whole power grid will become more uniform.

B. Power Pad Movement

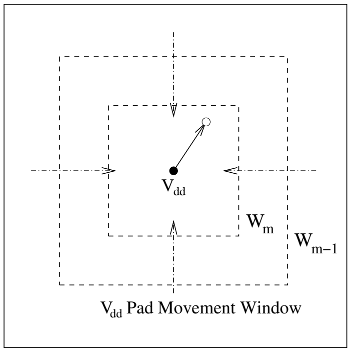

Fig. 6. While temperature drops, the window of moving a Vdd pad to another candidate location shrinks.

Initially, we place all power pads uniformly on the power grid. Then, we use the following scheme to move these power pads. We first pick a Vdd pad randomly, and then we move it to another empty power candidate location. If the change in the total cost function (denoted as ΔC) is less than zero, then we accept this move. Otherwise, we accept this move with probability equal to $e^{-\frac{\Delta C}{T_m}}$, where T_m is the current temperature.

Once we pick a random voltage pad as a move candidate, we randomly choose its new location within a window centered at its old location, as shown in Figure 6. As the temperature decreases, we shrink this window gradually using the formula $W_m = r \times W_{m-1}$, where W_m is the window size (as shown in Figure 6), and r is the temperature scaling factor. Eventually, at low temperatures, the Vdd pads are restricted to move only to their neighboring pad location candidates.

C. Fast Iterative Method

Recall that we use node-based iterative method to recompute the voltage values after each move, as in Section 3. In this method, we use the error bound ϵ to determine the convergence condition of voltage computations, as given in Figure 5. The convergence characteristics of a circuit with 250K nodes is demonstrated in Figure 7. In this figure, we can see the tradeoff between the number of iterations (proportional to runtime) and maximum error. Observe that reducing the runtime by 70% only incurs $3mV$ maximum error.

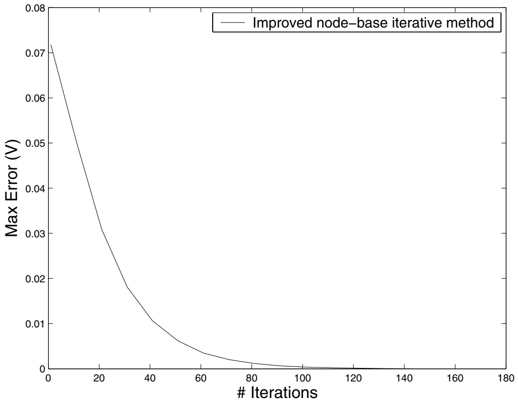

Fig. 7. The tradeoff between runtime and maximum error in the improved node-based method.

Using this tradeoff, we can improve the efficiency of the simulated annealing algorithm by introducing some tolerable error at high temperatures. When the temperature is high, estimating the change of cost in a fast way is more important than accurate voltage computations. However, as the temperature decreases, we improve the accuracy of the node-based iterative method gradually. Eventually, at the low temperatures, we make it converge to the exact solution. Specifically, we set $\epsilon = \epsilon_0$ in Figure 5 at initial temperature, which leads to fast voltage computations with small errors. As the temperature is decreased, we use the same scale factor r to update $\epsilon_m = r \times \epsilon_{m-1}$ so that the accuracy of voltage computation is increased with the drop of temperature.

V. Experimental Results

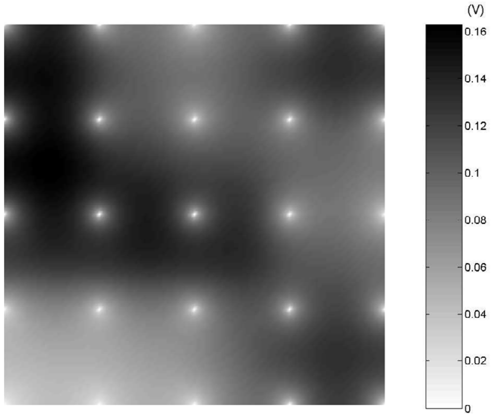

Fig. 8. In circuit P1, the IR drop on the power grid with initial uniform placement of power pads before optimization.

We have performed experiments to demonstrate the effectiveness of our power pad placement optimization algorithm. Our computations were carried out on a Linux PC with 2.8-GHz CPU and 4-GB memory. All the algorithms were implemented in C++.

For the purpose of illustrating the effect of our algorithm, we start with a relatively small-sized circuit with 10K nodes (denoted as P1 in Table 1). Figure 8 displays the IR drops in

TABLE I
RUNTIME AND QUALITY COMPARISON.

				Initial Design		After Pad Optimization		
Circuits	#nodes	#pads	#PCLs	MaxV(V)	σ(V)	MaxV(V)	σ(V)	time (m:s)
P1	10K	25	441	0.161	0.086	0.068	0.009	0:09
P2	251K	121	10201	0.191	0.098	0.097	0.013	1:13
P3	1M	121	40401	0.254	0.114	0.148	0.018	5:06
P4	4M	441	161604	0.234	0.110	0.121	0.016	21:37
P5	16M	441	646416	0.398	0.134	0.196	0.024	72:50

P1 for the initial placement of power pads before optimization, where power pads are uniformly distributed on the power grid. Here, we have applied the improved node based method [8], described in Section 2, to compute the IR drop of each node, which is the difference between the actual node voltage and the standard Vdd value. Before optimization, the worst IR drop in this circuit is 0.161 V.

Note that for C4 bumps, power pads can be placed at various points within the chip. In circuit P1, there is a grid of 21×21 candidate pad locations, which are uniformly distributed on the power grid. So, the objective of our power pad placement algorithm is to assign the 25 available power pads in P1 to those candidate pad locations so that the IR drops are minimized. Figure 9 illustrates the results of our algorithm. Observe that IR drops become much more uniform compared to the standard placement of Figure 8, and the maximum IR drop reduces to $0.068V$.

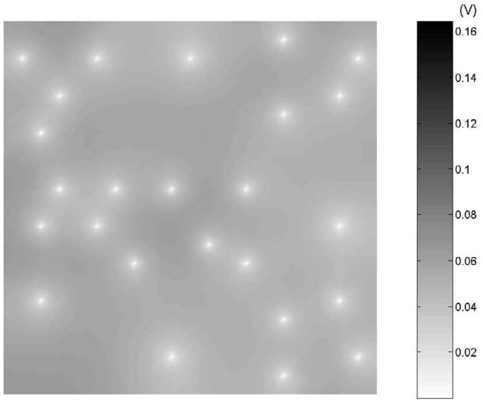

Fig. 9. In circuit P1, the IR drop on the power grid after optimization with 441 pad candidate locations.

To demonstrate the effectiveness of the proposed algorithm, we apply it on different circuits with number of nodes ranging from 10K to 16M as in Table I. The number of power pads and the number of pad location candidates (denoted as #PCLs in the table) are given for each circuit in the third and fourth columns of the table. A comparison of columns 5 and 7 shows the effectiveness of our methodology in terms of reducing the worst IR drops. Observe that the improvement in the worst IR drop is up to 58%. On the other hand, columns 6 and 8 compare the standard deviation of voltage drops. Observe that after the optimization, the standard deviation in voltage drops reduce by up to 89%. Note that the voltages reported in columns 7 and 8 are obtained from the simulation with optimized pad configuration. Column 9 shows the runtime of our power pad placement optimization algorithm, which is scalable even for circuits with multi-million nodes.

VI. CONCLUSION

In this paper, we studied the problem of power pad placement optimization for power grids. We developed an efficient localized node-based iterative method to compute the voltage changes after each movement of power pads. The complexity of updating the voltage values after each movement is independent of the total size of the power grid, which makes it scalable for large problems. Based on this method, we developed a simulated annealing based power pad placement algorithm to minimize the IR drops. Our experiments show that our algorithm not only significantly reduces the worst IR drops, but also makes the IR drops more uniform throughout the power grid. Furthermore, our algorithm is efficient, and demonstrates good runtime characteristics even for large number of power pad location candidates in multi-million-size circuits.

REFERENCES

[1] M. K. Gowan, L. L. Biro, and D. B. Jackson. Power considerations in the design of the alpha 21264 microprocessor. *35th DAC*, 1998.

[2] Y. M. Jiang and K. T. Cheng. Analysis of performance impact caused by power supply noise in deep submicron devices. *36th DAC*, 1998.

[3] J. Oh and M. Pedram. Multi-pad power/ground network design for uniform distribution of ground bounce. *Annual ACM IEEE DAC*, pages 157–162, 1998.

[4] S. X. Tan and C. R. Shi. Fast power/ground network optimization based on equivalent circuit modeling. *Proc. DAC*, 2001.

[5] T. Y. Wang and C. P. Chen. Power/ground mesh area optimization using multigrid-based technique. *DATE*, 2003.

[6] X. Wu. Area minimization of power distribution network using efficient nonlinear programming techniques. *ICCAD*, pages 153 – 157, 2001.

[7] M. Zhao, Y. Fu, V. Zolotov, S. Sundareswaran, and R. Panda. Optimal placement of power supply pads and pins. *Annual ACM IEEE DAC*, pages 165 – 170, 2004.

[8] Y. Zhong and M. D. F. Wong. Fast algorithms for IR drop analysis in large power grid. *ICCAD*, 2005.