

Multi-pad Power/Ground Network Design for Uniform Distribution of Ground Bounce*

Jaewon Oh

Sun Microsystems, Inc.
901 San Antonio Road, MS: USUN03-202
Palo Alto, CA 94303-4900
(408) 774-8657

joh@eng.sun.com

Massoud Pedram

University of Southern California
Dept. of Electrical Engineering - Systems
Los Angeles, CA 90089-2562
(213) 740-4458

massoud@zugros.usc.edu

ABSTRACT

This paper presents a method for power and ground (p/g) network routing for high speed CMOS chips with multiple p/g pads. Our objective is not to reduce the total amount of the ground bounce, but to distribute it more evenly among the pads while the routing area is kept to a minimum. We first show that proper p/g terminal to pad assignment is necessary to reduce the maximum ground bounce and then present a heuristic for performing simultaneous assignment and p/g net routing. Experimental results demonstrate the effectiveness of our method.

1. INTRODUCTION

Ground bounce refers to inductively induced voltage fluctuation in the internal ground lines of a chip (V_{ss}). Similarly, “power bounce” and “signal bounce” are voltage fluctuations in the power (V_{dd}) and signal (e.g data bus) lines, respectively. Since their characteristics are similar, we will focus on the ground bounce.

The major component of inductance comes from the chip’s package pin and the lead wire that connects the pin to the pad on the die. In current technology, inductance on power/ground (p/g) nets can be ignored. However, this inductance may be comparable to the package inductance in the future as the minimum feature size decreases. For now, we assume that package inductance is dominant. We also ignore mutual inductance between the lead traces of the package.

P/g routing is done in two phases: (a) construction of the net topologies, and (b) determination of width of the wire segments. The topologies connect the p/g terminals of each cell to the p/g pads on the chip. Our work deals with the first task (see also [3], [4]). The second task (see also [5], [6]) determines the width of

the wire segments so that reliability constraints such as voltage drop and electromigration constraints are met.

The topologies can be general graph, multi-pad tree or single-pad tree as shown in Figure 1. In [1] and [2], it was shown that general graphs and multi-pad trees can always be reduced to single pad trees which have lower area while satisfying the same reliability constraints. Therefore we restrict our topologies to single-pad trees. The topology generators in [3] and [4] focused on planarity of the p/g net. However, the planarity can be achieved under special terminal and pad configurations (locations and number of pads and terminals). As our target is high performance chip design, we assume that there are enough metal layers and planarity is of no concern.

The remainder of this paper is organized as follows. In section 2, we give the background for ground bounce. We define the p/g network design problem and present a heuristic technique for solving it in sections 3 and 4. Sections 5 and 6 give our experimental results and concluding remarks.

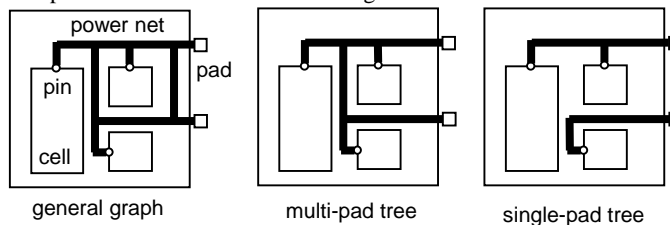


Figure 1: Various power/ground net topologies (taken from [2])

2. BACKGROUND

2.1 Delta-I noise

It is not uncommon to see ASIC chips that are designed correctly both in functionality and timing, but do not work when they are plugged into the sockets. It is likely that the designer did not fully appreciate the bounce noises. The bounce noise, also called simultaneous switching noise or delta-I noise, arises when multiple transistors switch simultaneously. The spontaneous current demand (di/dt) appears as a voltage drop ($L di/dt$) across the inductor which is present between the system board’s ground and the ground pad of the chip (similarly with power pads). Together with the capacitance in the current path, the voltage on the ground experiences LC oscillation. For on-chip drivers, the power pad simultaneously experiences the same noise with opposite polarity to that of the ground bounce noise. This is manifested in a variety of transient and permanent circuit malfunctions, including the appearance of undesirable glitches,

* This work was funded in part by DARPA under contract no. F33615-95-C1627 and SRC under contract no. 98-DJ-606.

the flipping of state bits in registers and memory, and additional signal delay due to reduced voltage across the power and ground. For more detailed explanation and analytical delta-I noise estimations, see [7], [8] and [9].

2.2 Distribution of delta-I noise

Our problem domain is p/g routing in sea-of-gates or full-custom designs. P/g routing refers to the process of building connections between the power/ground terminals of the cells and the pads. Pads are usually placed around the periphery of the chip die. Associated with the p/g terminals of every cell are the amount of maximum current changes (di/dt). We assume that these numbers are given. They can be obtained from simulations or analytical estimations. When two terminals are connected, we add up the di/dt of the two terminals. This is not always correct when the two terminals are correlated such that the maximums of di/dt do not occur at the same time. For simplicity, we assume that peak di/dt is simply additive.

The next step is to assign each terminal to a pad. Since we have chosen single pad tree topology, two terminals assigned to the same pad belong to the same tree. If the peak di/dt of each part is severely unbalanced, then some pads may experience too much noise while other pads experience little noise. For example in Figure 2, the partition in (b) is balanced while (a) is not. In (a), the pad A has di/dt almost twice that of pad B. In (b), both pads have nearly the same di/dt .

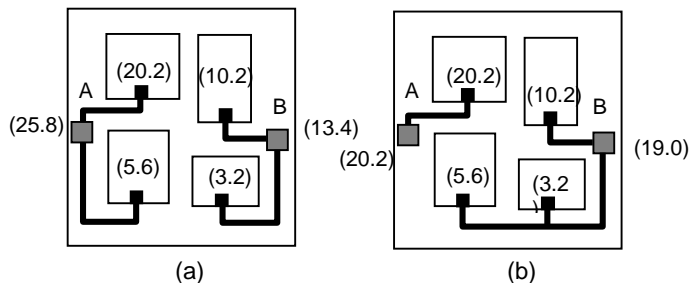


Figure 2: Two different routings for the same cell placement. Numbers inside the parentheses are the peak di/dt .

2.3 Limitation of Other Methods

On-chip decoupling capacitance (decap) helps reduce the ground bounce. The decap is placed across the V_{dd} and V_{ss} close to the cell as shown in Figure 3. The spontaneous current demand by the on-chip drivers is partially supplied by the decap rather than all the current being supplied from the outside of the chip.

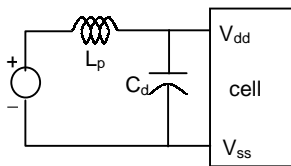


Figure 3: Decoupling capacitance C_d and pin inductance L_p

The decap is usually implemented using unused transistors in the vicinity of the cell. For large amount of di/dt , bigger capacitor should be used. Since the decap occupies silicon area, it may not always be possible to find such a big area near the cell. Besides, large decap may lead to unreliable operation of the chip as described next. Suppose a pad has exceptionally large amount of di/dt . Then in the routing tree that includes the pad, there will be a lot of on-chip decaps attached to tree to reduce the current

transient of the cells. As a result, the capacitance of the pad becomes very large. In Figure 3, the voltage source is as good as short for the AC current. Therefore the decap C_d and the inductance L_p form a LC tank. The tank has a resonance frequency of

$$\omega = \frac{1}{\sqrt{L_p C_d}}$$

If C_d is sufficiently large, ω can be close to the operating frequency of chip. In that case, the resonance occurs at the power lines over many cycles, i.e. a large voltage fluctuation builds up at the power lines and causes the circuit to fail. Therefore we cannot indefinitely increase the size of the decaps to reduce the ground bounce.

Wire sizing (increasing width) of the p/g net to meet the IR voltage drop and the electromigration constraints worsens the ground bounce problem. Since wire sizing reduces the resistance of the net, the amount of damping on the LC oscillation is reduced, therefore increasing the ground bounce. Hence wire sizing is not a solution to the ground bounce problem.

Another popular method for ground bounce reduction is to connect all the ground pads together to create a ring. This has the effect of sharing currents between pads. When a pad has excessive current flow, the current can be absorbed by the nearby pads. However, due to finite resistance of wires between pads, the excessively loaded pad still gets more ground bounce than its neighbors. Besides, if all the nearby pads also get excessive current, having connections among them is not very helpful.

3. Problem Definition

In the previous section, we have seen that uneven current distribution among pads cannot be fixed easily. So we need a method of balanced partitioning of pins based on their current requirements. At the same time, the partitioning should be done such that the routing area in each part is small. We define the following notations:

M: number of pads

Y: number of terminals

N: number of pins (= M + Y)

P = $\{p_1, p_2, \dots, p_N\}$: all the pins

P = $P_1 \cup P_2 \cup \dots \cup P_M$: disjoint partition of pins

where each P_i contains exactly one pad

We call this problem the “Minimum Cost Balanced Partitioning (MCBP)” problem.

Definition. Minimum Cost Balanced Partitioning (MCBP):

Given a number M of pads and a number Y of cell terminals with their geometrical locations and a set of current requirements (di/dt) for the terminals, partition the set of pins into M parts where each part has exactly one pad such that the total routing cost is minimized and di/dt of each part is no more than a bound B.

Since there is no net in the input of our problem (in fact, determining the nets is our objective), it does not fall on the category of traditional net-based partitioning problems. It is instead similar to the bin-packing problem. It can be shown that MCBP is NP-complete [12].

The parts should be formed such that they are balanced and the routing cost for each part is minimized. To achieve these two

objective simultaneously, we propose a “Min-Forest Heuristic” as follows.

4. Min-Forest Heuristic

In our Min-Forest heuristic, the routing area for each part is approximated as the cost (sum of the edge lengths) of Minimal Spanning Tree (MST). Spanning tree approximation of the routing area is certainly an overestimation of the real Steiner tree routing, but it can give at least a comparative measure of routing areas of different parts. Starting from the pads, we grow an MST for each pad using the well-known Prim’s algorithm [11]. At any moment during the growth of trees, there are exactly M MSTs (hence the name Min-Forest). When all the pins are included, these MSTs naturally constitute the partitioning solution. During the growth of trees, each MST competes to acquire a next available terminal that is not included in any MST. Some MSTs may compete to acquire the same terminal while others may compete to acquire different terminals. The winner of this competition is the one which makes the cost function smallest. These are the basics of our Min-Forest heuristic.

In the beginning, the parts P_1, P_2, \dots, P_M contain one pad each. Let T_1, T_2, \dots, T_M be the MSTs for each part. Also let $|T_i|$ be the cost of MST and $|P_i|$ be the size of the part P_i . The size of a part is the sum of di/dt of all the terminals in that part. To achieve both balanced partitioning and minimum routing cost, we propose the following cost function.

$$\frac{\sum_{i=1}^M |T_i|}{\sum_{i \neq j} |P_i| |P_j|} \quad (4.1)$$

The numerator favors low routing cost while the denominator favors balanced partitioning. This is similar to the cost function of ratio-cut partitioning problem [10] except that the numerator is the sum of the routing costs of each part rather than the cut size. Although this cost function helps obtain a balanced partitioning, it alone cannot return a balanced partitioning to the degree we want. So in our heuristic, we add a constraint whereby the growth of an MST stops if its size is more than $(1+\epsilon) Q/M$ where Q is the sum of all the di/dt on the chip and ϵ is a user defined parameter. For small ϵ , we get highly balanced partitioning. For large ϵ , the balance is simply determined by the cost function, and the degree of the balance may not be what we want.

In Prim’s algorithm, a single tree grows starting from the root node by continuously adding *light edges*. A light edge (u, v) is an edge whose edge length is minimum where u is a node already included in the tree and v is a node not included yet. Let v be light node of the tree. During the growth of the forest of MSTs, each MST has a light node that is not included in any MST. For each MST, we tentatively include its light node and evaluate the cost function of (4.1). Then we choose an MST which makes the cost function the lowest and let the MST acquire its light node. This procedure is iterated until all the nodes are included. Next is the summary of the Min-Forest heuristic.

Min-Forest(P : pins, ϵ : number)

Initialize each T_i to one pad tree;

Repeat

Select T_i and its light node u_i such that $|P_i \cup \{u_i\}| \leq (1+\epsilon)Q/M$ and the resulting cost function (4.1) is minimum;

$T_i \cup T_i + \{u_i\}$;

Until all the nodes are included;

5. Results

We implemented our algorithm in C++ on an Ultra Sparc 2. The benchmarks are generated randomly and tested. In the benchmarks, pads are located evenly around the periphery, and the locations of the terminals and their di/dt values are randomly and independently generated. For benchmark size 120 or lower, our program runs in sub-seconds. Below is the characteristics of our benchmarks.

Benchmark	# pins	# pads
B20	20	5
B30	30	4
B80	80	8
B120	120	10

Table 1: Benchmark set

5.1 Comparison with the conventional methods

We first consider a case where $|P_i| = \#$ of pins in P_i . Traditional p/g routing involves the partitioning of the entire chip area such that all the parts have approximately equal number of terminals and then assigns the closest pad from each part. Therefore, our Min-Forest heuristic with the above the size definition is a close approximation of the traditional method since each MST tries to acquire terminals in the vicinity of the pad where the MST started with. However, since the traditional method ignores the current, we can expect unbalanced di/dt distribution. Next we consider the case $|P_i| = \text{sum of } di/dt$. We tested our heuristic for both partition size definitions on B30 and B120 with $\epsilon = 0.2$.

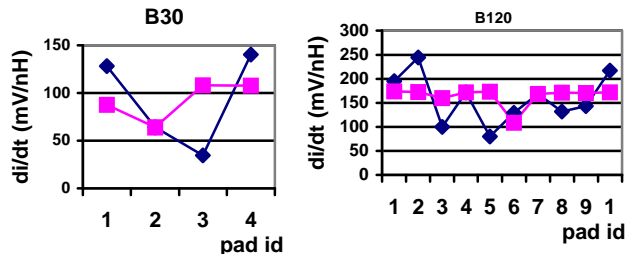


Figure 4: Current distribution comparison between the conventional method and our proposed method. Conventional method (diamond dots) vs. our method (square dots).

As can be seen in Figure 4, our method consistently returns more balanced current distributions compared with the conventional method. However due to enforcement of balanced current distribution in our method, the routing area is increased. This is shown in Figure 5.

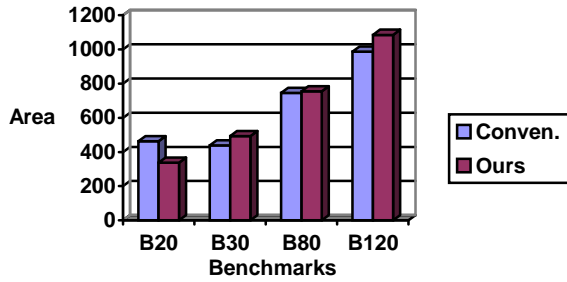


Figure 5: Comparison of the routing area between the conventional method and our proposed method.

5.2 Effects of ϵ

The effect of ϵ on partition balance is shown in Figure 6. We computed standard deviations of the di/dt for various values of ϵ . As can be seen from the figure, ϵ should be low to get balanced partitioning. However, low ϵ tends to result in large tree costs as can be seen in Figure 7.

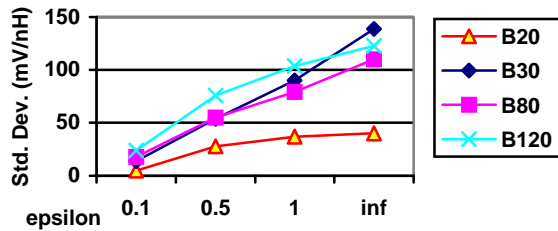


Figure 6: Standard deviations for different value of ϵ

6. CONCLUSION

We proposed a Min-Forest heuristic for the p/g net routing and partitioning for uniform distribution of ground bounce across the pads. We have shown that proper partitioning is essential in reducing the ground bounce as other methods have limitations in doing the job.

For a complete p/g routing, our algorithm can be separately applied on power and ground. If different p/g nets are used for on-chip and off-chip drivers, the algorithm can also be separately applied for these nets. Our algorithm can be used without modification when the pin inductances are different or when noise margins are different across the pins. In such cases, we can just enforce different partition sizes for each tree.

Our work is not a complete substitute for other ground bounce reduction techniques. Use of decaps or other techniques are still needed to reduce the total amount of ground bounce. However, if our work is used together with those techniques, it will give better distribution of noise and less chance of circuit failure.

Future work will include the extension of our method under the presence of activity correlation between cells (di/dt is not simply additive) and using other topologies such as multi-pad trees.

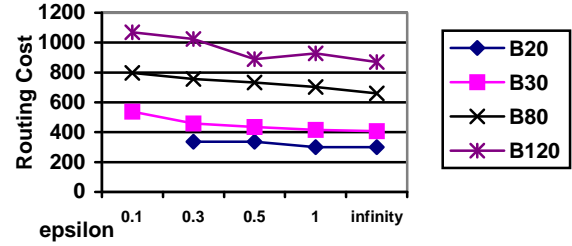


Figure 7: Routing cost vs. epsilon tradeoff. Missing data means no feasible solution at that epsilon

7. References:

- [1] K.-H. Erhard and F.M. Johannes, "Power/ground networks in VLSI: are general graphs better than trees?," *Integration, the VLSI journal*, vol. 14, pp. 91-109, 1992.
- [2] K.-H. Erhard, F.M. Johannes and R. Dachauer, "Topology Optimization Techniques for Power/Ground Networks in VLSI," *Proc. European Design Automation Conference*, pp. 362-367, 1992.
- [3] Zahir A. Syed and Abbas El Gamal, "Single Layer Routing of Power and Ground Networks in Integrated Circuits," *Journal of Digital Systems*, vol. 6, no. 1, 1982.
- [4] H. Cai, "Multi-pads, single layer power net routing in VLSI circuits," *Proc. ACM/IEEE Design Automation Conference*, pp. 183-188, 1988.
- [5] S. Chowdhury and Melvin A. Breuer, "Optimum Design of IC Power/Ground Nets Subject to Reliability Constraints," *IEEE Transactions on Computer-Aided Design*, vol. 7, no. 6, pp. 787-796, July, 1988.
- [6] Robi Dutta, Malgorzata Marek-Sadowska, "Automatic Sizing of Power/Ground Networks in VLSI," *Proc. IEEE/ACM Design Automation Conference*, pp. 783-786, 1989.
- [7] A. Vaidyanath, B. Thoroddsen, J.L. Prince, "Effect of CMOS Driver Loading Conditions on Simultaneous Switching Noise," *IEEE Trans. on Components, Packaging, and Manufacturing Technology - Part B*, vol. 17, no. 4, pp.480-485, Nov. 1994.
- [8] Y. Yang, J.R. Brews, "Design Trade-Offs for the Last Stage of an Unregulated, Long-Channel CMOS Off-chip Driver with Simultaneous Switching Noise and Switching Time Consideration," *IEEE Trans. on Components, Packaging and Manufacturing Technology - Part B*, vol. 19, no. 3, pp. 481- 486, Aug. 1996.
- [9] P. Larsson, C. Svensson, "Noise in Digital Dynamic CMOS Circuits," *IEEE J. of Solid-State Circuits*, vol. 12, no. 6, pp. 655-662, June 1994.
- [10] C. K. Cheng and Y.-C. A. Wei, "An Improved Two-way Partitioning Algorithm with Stable Performance," *IEEE Trans. on CAD*, vol. 10, no 12, pp. 1502-1511, Dec. 1991.
- [11] T. Cormen, C. Leiserson and R. Rivest, "Introduction to Algorithms," *The MIT Press*, pp. 337-343, 1990.
- [12] See our web site at <http://atrk.usc.edu/~joh>