Successive pad assignment algorithm to optimize number and location of power supply pad using incremental matrix inversion

Takashi Sato[1], Masanori Hashimoto[2], and Hidetoshi Onodera[1]

[1]Kyoto University, Kyoto, Japan, [2]Osaka University, Osaka, Japan

ABSTRACT

An efficient pad assignment algorithm to minimize voltage drop on a power distribution network is proposed. Combination of the successive pad assignment (SPA) and the incremental matrix inversion (IMI) provides an efficient assignment for both location and number of power supply pads. The SPA creates equivalent resistance matrix which preserves both pad candidates and power consumption points as external ports so that topological modification due to connection or disconnection between voltage sources and candidate pads are consistently represented. By reusing sub-matrix of equivalent matrix, the SPA greedily searches next pad location that minimizes the worst drop voltage. Each time the candidate pad is added, the IMI reduces computational complexity significantly. Experimental results show that the proposed procedures efficiently enumerate pad order in practical time.

I. INTRODUCTION

As predicted by Rents rule, number of I/O pins of a LSI package has been increasing rapidly as more number of devices is packed into a chip [1]. Typical pin count for recent LSI has reached several hundreds or over thousand and expected to increase even more. However, number of pins allotted to power supply and ground are usually limited. Recent technology advances have allowed us to include several millions of transistors on a chip, which increases power consumption. Therefore, inefficient assignment of the power supply pad may cause severe voltage drop. Number and location of supply pads have to be optimized. One of the difficulties for pad optimization is the size of power distribution network (PDN) to analyze. Even in the early planning stage, PDN becomes quite large. A methodology that is suitable for quick what-if analysis is required. Another difficulty is the topological modification of the PDN. Designers try to optimize pad number and location by connecting and disconnecting power supply to pads choosing among more than hundred pads. In order to explore vast combination of the pads, a simulation model must be reused to exploit analysis efficiency.

A lot of works on voltage drop analysis have been published since it has already become a real-life concern for industrial chip design [2, 3]. The use of various original and efficient numerical techniques is proposed to accelerate computational time [4, 5, 6]. These methods are suitably used once a PDN is fixed for analysis. However, when optimizing number and location of supply pads, circuit topology will be modified trial by trial. Therefore, full procedures such as setting up a new circuit matrix, symbolic factorization, numerical factorization, and forward-backward substitution, are required for every trial. Those processes become significant overhead when there are many pad combinations. Pad layout optimizations for signal integrity are discussed in [7] but their objectives do not include voltage drop minimization although they cover timing, simultaneous switching noise and crosstalk noise. Other works related to this subject are power network planning from reliability standpoint [8] and minimum number pad assignment with planar layout realization problem [9], but unfortunately there is no mentioning about voltage drop. Min-Forest heuristic is proposed in [10] to achieve uniform ground bounce on power/ground tree of predetermined number of pads. The authors in [11] propose to solve pad optimization problem utilizing linear programming framework with divide and conquer approach and candidate pruning. But the computational complexity required to solve a dense equivalent admittance matrix may still limit solvable problem size when the PDN cannot be adequately partitioned.

In this paper, we first build an equivalent resistance matrix to adopt for slight topology modificatin, then we utilize incremental matrix inverse approach to enhance computational complexity. The contributions of this paper are 1) an efficient voltage drop calculation applicable for set of circuits with different number or location of power supply pads, and 2) efficient calculation to greedily optimize pad number and location.

II. PROBLEM FORMULATION

In the following sections, static voltage drop is discussed — a PDN is modeled using resistors only. Followings are the assumptions for supply pad assignment problem.

- There is a set S_p of pads P_i to subset of which we assign ideal power supply represented by ideal voltage source. The size of S_p is n_p, i.e. $\{S_p \ni P_i | i = 1 \ldots n_p\}$.
- There exists a set S_q of current sink points Q_i which represents each circuit block's power consumption. The size of S_q is n_q, i.e. $\{S_q \ni Q_i | i = 1 \ldots n_q\}$. At all or part of the points in S_q, independent current source $j_i\,(i = 1 \ldots n_q)$ is connected to represent power consumption around Q_i.
- Voltage drop of a chip is observed at all points in S_q. Absolute supply voltage measured at Q_i is defined as V_{Q_i}. Drop voltage ΔV_{Q_i} equals $V_{dd} - V_{Q_i}$.

Figure 1 depicts a chip image including pads, circuit blocks, and current sink points. Inside circuit blocks are the current

0-7803-8736-8/05/$20.00 ©2005 IEEE.

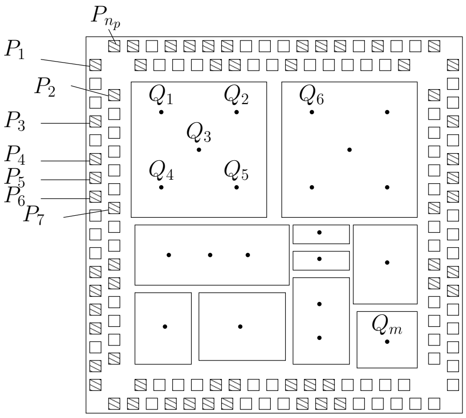


Fig. 1. Overview of candidate pads (P_i) and power consumption points (Q_i) distribution for a chip with peripheral pad. Q_i also serves as observation points for voltage drop.

sink points Q_i that represent power consumption and are used as drop voltage observation points. Larger block requires more number of representative points in general especially when power consumption in a block is distributed unevenly. PDN is modeled by resistors although they are not illustrated in Fig. 1 for simplicity. This example shows peripheral I/O pads but proposing algorithm is applicable for other types of I/O, such as area pad, without any modification. The algorithm can be also applied to pin assignment problem for a circuit block but we concentrate only on pad assignment problem in this paper.

Hatched pads P_i are candidates for making connections to external power supply through bonding wires or bumps. Remaining pads drawn without hatch are for other purposes, such as for signals. Larger number of pads can be included in a set S_p than the maximum number allowed for power supply since increasing combinations enlarges solution space and possibly yields better solution than starting from very limited choices. If the algorithm revealed that the number of power supply pads could be reduced, unused pads can be assigned to other signals. Or, the use of less expansive package, which usually have smaller outline and smaller number of pins, may become possible. Therefore, choosing a subset of S_p that gives minimum drop voltage at the observation points is our objective.

Based on the above presupposition, we define following pad assignment problems.

Problem 1 Given a PDN, power consumption distribution, and allowable number of power supply pads $n_a\,(n_a \leq n_p)$, determine the location of pads that maximize worst voltage drop V_{worst}, defined as $V_{worst} = \max_i(\Delta V_{Q_i})$.

Problem 2 Given a PDN, power consumption distribution, supply pad candidate set S_p, and voltage drop target V_t, determine the minimum number of pads and their locations so that $V_{worst} < V_t$ is satisfied.

III. Circuit model for the PDN

The largest difference between ordinary IR-drop calculation and pad assignment problem is whether circuit topology is finalized or not. In the pad assignment, all pads initially included in S_p may not be necessarily connected to the power supply — pads that are not connected to power supply must be treated as open ports. Figure 2 is an abstracted PDN model for pad assignment problem. On the right of PDN are the ports for current sink connection. In the figure, nodes in sets S_p and S_q are defined as external ports. Once voltage source connection to the pad are determined, voltage distribution of the circuit is calculated by solving linear equation constructed by using modified nodal approach:

$$\begin{pmatrix} Y & E \\ E^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{v_q} \\ \boldsymbol{i_p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{J_q} \\ \boldsymbol{v_p} \end{pmatrix} \tag{1}$$

for $\boldsymbol{v_q}$, or simpler nodal analysis formulation which replaces ideal supply voltage sources to Thevenin's equivalent current sources. Here, Y is a conductance matrix, E is an incident matrix for voltage sources, $\boldsymbol{v_q}$ is a nodal voltage vector for observation points, $\boldsymbol{v_p}$ is a pad voltage vector, $\boldsymbol{i_p}$ represents voltage source current vector, and $\boldsymbol{J_q}$ is current source vector. Eq. (1) is solved using direct methods, iterative methods [3], or various techniques such as [4, 5, 6].

Direct decomposition methods are efficient for repeatedly solving a set of IR-drop problems. As far as the circuit topology is fixed, recalculation of the same PDN with different current vector $\boldsymbol{J_q}$ is efficient since decomposed matrix, which consumes most of the calculation time to generate, can be reused. However, to solve for the problems defined in the previous section, Eq. (1) has to be repeatedly constructed and solved for slightly different connections of the supply sources. In this case, every time the circuit topology changes, the most time-consuming process of matrix decomposition is required for new circuits.

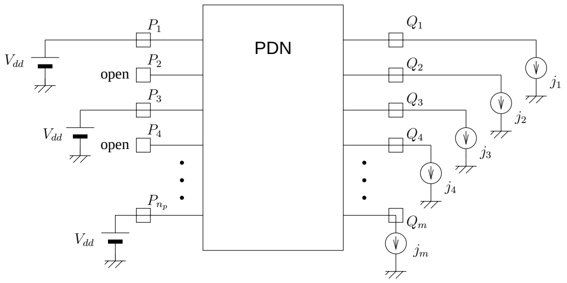


Fig. 2. Abstracted chip model for Fig. 1.

IV. Efficient IR drop calculation for exploring pad assignment combination

In this section, we propose the SPA algorithm that efficiently assigns pads to minimize voltage drop. Distinguishing characteristic of the SPA is to create an equivalent resistance matrix preserving possible all voltage supply pads and current source points as terminals, and solves it using the IMI.

A. Step 1: Determination of the first source point

To calculate observation node voltages in Fig. 2, at least one node voltage has to be defined. Otherwise, the conductance matrix in Eq. (1) becomes singular. Step 1 determines first supply point that will be used as the voltage reference in the following steps. We call this pad as the reference pad. If any supply pads are already fixed as supply pads, this step can be skipped. Since one of the fixed supply pads can be used as the reference pad.

Following is the procedure to determine reference pad.

1. Connect all pads P_i in set S_p to ground.
2. Connect all circuit currents j_i to sinks Q_i in set S_q.
3. Calculate currents at all pads $i_{P_1}, i_{P_2}, \ldots i_{P_{n_p}}$.
4. Find a pad P_k $(0 \leq k \leq n_p)$ on which the largest current flows $i_k = \max_\ell(i_{P_\ell})$ and use P_k as the reference P_{ref}.

From its derivation, P_{ref} has the lowest combined equivalent resistance for all current points Q_i. Thus the pad location for one pad case is obtained.

B. Step 2: Equivalent resistance matrix calculation

Once the P_{ref} is determined, then we derive equivalent resistance matrix R_{eq} through following procedures.

1. Connect reference pad P_{ref} to ground.
2. For each pad P_ℓ in set $S_r = S_q \cup S_p$ except for reference pad $P_\ell \neq P_{ref}$ $(\ell = 1 \ldots (n_p + n_q - 1))$, connect 1 ampere current source between P_ℓ and P_{ref}. Leave all other ports open including the ones in S_q.
3. Measure all port voltages as a column vector $\boldsymbol{v_{p_\ell}}$. Here, $\boldsymbol{v_{p_\ell}}$ is arranged as

$$\boldsymbol{v_{p_\ell}} = (V(Q_1), \ldots, V(Q_{n_q}), V(P_1), \ldots, V(P_{n_p}))^T. \tag{2}$$

Then set the voltage vector as ℓ-th column in R_{eq}.

$$R_{eq} = (\boldsymbol{v_{p_1}}, \ldots, \boldsymbol{v_{p_m}}, \boldsymbol{v_{p_{(n_q+1)}}} \cdots, \boldsymbol{v_{p_{(n_q+n_p-1)}}}) \tag{3}$$

R_{eq} can be efficiently obtained using direct methods. As mentioned earlier, once LU decomposition of the circuit matrix in Eq. (1) is found, it can be reused for different current connections on the right hand side. Calculation cost required for different current source connection is one forward substitution and one backward substitution.

C. Step 3: Calculation of voltage drop with different pad combination

Using R_{eq}, the voltage drop at the observation points are obtained as follows.

1. For equivalent resistance R_{eq},

$$\begin{pmatrix} \boldsymbol{v_q} \\ \boldsymbol{v_P} \end{pmatrix} = R_{eq} \begin{pmatrix} \boldsymbol{J_q} \\ \boldsymbol{i_P} \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{J_q} \\ \boldsymbol{i_P} \end{pmatrix}, \tag{4}$$

here R_{eq} is partitioned by number of observation points n_q, and number of candidate pads $n_p - 1$. $R_{21} = R_{12}^T$, $\boldsymbol{v_q}$ is observation point voltages and $\boldsymbol{J_q}$ are node voltage at S_q and current vector connected at S_q, respectively. $\boldsymbol{v_P}$ and $\boldsymbol{i_P}$ are voltage and current vectors for pads.

2. For pad k which is not connected to power supply, corresponding rows and columns in R_{eq} can be eliminated since current flowing into the pad is zero and its pad voltage is not interested. Let $\boldsymbol{v_{Ps}}, \boldsymbol{i_{Ps}}$ be voltage and current vectors for source connecting pads only.

$$\begin{pmatrix} \boldsymbol{v_q} \\ \boldsymbol{v_{Ps}} \end{pmatrix} = \begin{pmatrix} R_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{J_q} \\ \boldsymbol{i_{P_s}} \end{pmatrix} \tag{5}$$

H_{12} is column, H_{21} is row, and H_{22} is row and column eliminated sub matrix of size $(n_a - 1)$ from R_{12}, R_{21}, and R_{22}. n_a is a number of pads connecting to power supply as defined in problem 1.

3. Solving Eq. (5) for observation point voltages yield

$$\boldsymbol{v_q} = (R_{11} - H_{12}H_{22}^{-1}H_{21})\boldsymbol{J_q} + H_{12}H_{22}^{-1}\boldsymbol{v_{P_s}} \tag{6}$$

Since $\boldsymbol{v_{P_s}}$ is defined as relative supply voltage to the reference node, $\boldsymbol{v_{P_s}} = \boldsymbol{0}$ for the cases with single supply voltage. Then $\boldsymbol{v_q}$ becomes

$$\boldsymbol{v_q} = (R_{11} - H_{12}H_{22}^{-1}H_{21})\boldsymbol{J_q}. \tag{7}$$

The first term in Eq. (6) is the voltage drop due to the combination of chip power consumption and given PDN. The second term is voltage change due to the different supply voltages other than V_{dd}. $\boldsymbol{v_{P_s}}$ becomes non-zero only when the power supplies at the pads are non-ideal such as the case to include voltage drop in packages or printed circuit boards. Eq. (7) also consists of two terms. $R_{11}\boldsymbol{J_q}$ is the voltage drop when supply voltage is provided by reference node only. The second term $H_{12}H_{22}^{-1}H_{21}\boldsymbol{J_q}$ is voltage recovery by adding additional power supplies to other pads.

D. Successive pad assignment algorithm

Combining above procedures, we construct a pad assignment algorithm. The pad assignment is a combinatorial optimization problem in which combination explodes for large number of pads. Exhaustive search of all combination is prohibitive for this kind of problem. Therefore, we propose an algorithm which adopts greedy approach for assigning pad locations. A procedure *Successive_pad_assignment* (SPA) starts determining reference pad, then the equivalent resistance matrix is determined. After that, local optimal pad is determined one by one. The inner loop of i calculates local worst voltage drop for all possible pad assignment. A pad that yields smallest voltage drop will be selected then resistance matrix is reordered for the IMI.

E. Efficiency improvement through incremental matrix inverse

The SPA gives an approximation solution. It preserves previously selected pads, which prunes significant number of trials. However, as shown in Eq. (7), the SPA includes inverting a matrix $H_{22} \equiv A_{11,r}$ which is of size r to determine the r-th pad. Since H_{22} is generally a dense matrix from its construction, calculation complexity is in order of $\mathcal{O}(n_a^3)$ through direct methods. When n_p or n_a is much smaller than the original circuit nodes, which is usually the case, the use of equivalent resistance matrix is efficient. However, when the number of candidate pads is large, it again becomes a bottleneck. In the

Algorithm 1 Successive_pad_assignment()

```
determine_reference_pad
$R_{eq}$ =construct_equivalent_resistance_matrix
for $i = 1..n_p$ do
    $pad\_order(i) = i$;
end for
for $j = 1..n_p$ except for reference pad do
    for $i = j..n_p$ do
        $H_{12,(j)=}$ append column $i$ of $R_{eq}$ to $H_{12,(j-1)}$
        $H_{21,(j)=}$ append row $i$ of $R_{eq}$ to $H_{21,(j-1)}$
        $H^{-1}_{22,(j)}$ =incremental_ matrix_ inverse($H^{-1}_{22,(j-1)}$)
        $\boldsymbol{v_q} = (R_{11} - H_{12}H^{-1}_{22}H_{21})\ \boldsymbol{J_q}$
        store $(i_{min} = i, u, H)$ if $u = \max(\boldsymbol{v_q})$ is smallest
    end for
    $best\_index = i_{min}$
    store $(H^{-1}_{22,(j-1)}, H_{12,(j-1)}, H_{21,(j-1)})$
    $R_{eq}$ = swap_matrix($m + best\_index, m + j$)
    $pad\_order$=swap_vector($best\_index, j$)
    $best\_pad$=pad_order($best\_index$)
end for
```

SPA algorithm, as it preserves already selected pads, the matrix $H^{-1}_{22} \equiv A^{-1}_{11,(r-1)}$ of the size $(r-1)$ has been already calculated when trying to add the r-th pad. Here $A_{11,(r-1)}$ is a common sub-matrix of $A_{11,r}$. We utilize this matrix for an incremental inverse calculation of H_{22} to pursue further efficiency of the SPA algorithm.

Let A_{11} be a size r square matrix whose inverse is already calculated. A_{11} and its inverse are both symmetric by construction. We calculate a matrix B which is an inverse of A.

$$AB \equiv \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & a_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} E_{11} & 0 \\ 0 & 1 \end{pmatrix} \quad (8)$$

here, A_{12}, A_{21}, a_{22} are row and column vectors corresponding to the next supply pad candidate, respectively. A_{12} is a $(r-1) \times 1$ column vector, $A_{21} = A^T_{12}$, and E_{11} is size $(r-1)$ identity matrix. Solving each component in B being A^{-1}_{11} as known yields following set of equations.

$$\begin{aligned} B_{21} = B^T_{12} &= (C_{21}A_{12} - a_{22})^{-1}C_{21} && (9) \\ B_{11} &= A^{-1}_{11} - C^T_{21}B_{21} && (10) \\ b_{22} &= (1 - A_{21}B_{12})/a_{22}. && (11) \end{aligned}$$

Here, $C_{21} = A_{21}A^{-1}_{11}$. Computational complexities for above equations can be evaluated by number of multiplications required and is $\mathcal{O}(r^2)$ at the maximum which improves order of magnitude faster than conventional matrix inverse of $\mathcal{O}(r^3)$. Figure 3 shows CPU time comparison between conventional matrix inverse and incremental matrix inverse using Octave [12] on a 2.8 GHz Linux workstation. Single dense matrix inverse for more than 1500 port problem is less than 0.1 second, which is more than 100x faster than conventional method.

F. Discussions

Since construction of Eq. (7) is independent from $\boldsymbol{J_q}$, any current distribution can be applied to ports once the R_{eq} is constructed. This is a very good property. It enables what-if analysis on different current distributions because a chip usually has several scenarios or user modes. We understand that the SPA does not necessary give a very good solution but its efficiency enables analysis on multiple scenarios quickly. Choosing commonly assigned pads over different combinations of $\boldsymbol{J_q}$ should give more robust solution. There are also many occasions that a designer has solid power estimate for particular blocks while other blocks are not. Using proposed approach, the designer can play with power consumption of the blocks by changing $\boldsymbol{J_q}$ to see their impact on the voltage drop.

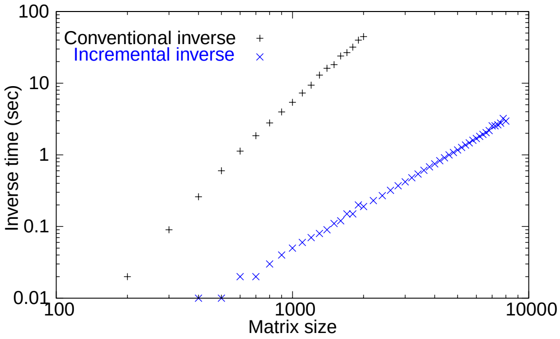


Fig. 3. CPU time required for R_{eq} inverse.

V. Experimental results

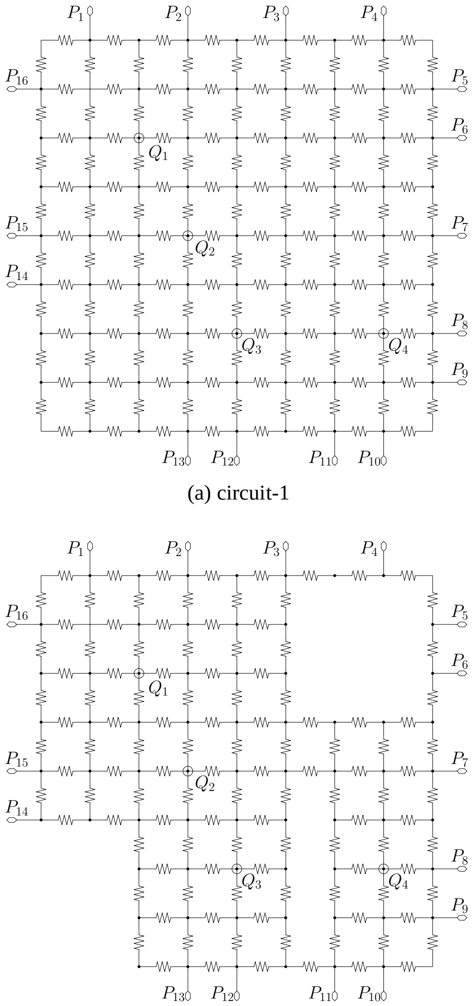
Figure 4 shows simple PDN models. There are 16 supply pad candidates $(P_1, \ldots, P_{16})$ and power supply voltages are 1.0 V for both circuits. Locations of the pads and current sink points are the same. The difference between two circuits is the PDN. Circuit-1 has uniform grid while circuit-2 contains obstacle blocks, such as analog circuits or power-gated circuits, to which the PDN in interest is prohibited to connect. Resistors in the figures are 100 – 200 mΩ. Representative currents at the observation points are: $(j_1, j_2, j_3, j_4) = (0.6, 0.5, 0.6, 0.3)$ A respectively.

TABLE I

Efficient and inefficient pad assignment. Numbers correspond to the pad numbers in Fig. 4.

Assign. order		1	2	3	4	5	6	7	8
	ckt	9	10	11	12	13	14	15	16
best	#1	12	2	8	15	6	11	16	14
		3	7	13	1	4	9	10	5
	#2	12	2	8	15	11	16	7	13
		3	14	1	10	9	4	5	6
worst	#1	5	4	6	3	2	1	16	7
		8	9	10	11	15	14	13	12
	#2	5	6	4	7	8	9	10	11
		13	12	3	14	15	16	1	2

Table I shows pad selection order calculated for circuit-1 and 2. For both circuits, the reference pad is determined using the procedure step-1, '*determine_initial_power_pad*'. Next, the second pad is assigned by calculating voltage drop $\boldsymbol{v_q}$ using all remaining pad candidates. The best choice as the second pad for the circuit-1 is P_2. P_2 is located on the opposite edge to the reference pad, which is quite reasonable. First four pads are



(a) circuit-1

(b) circuit-2

Fig. 4. Example circuits with different PDN.

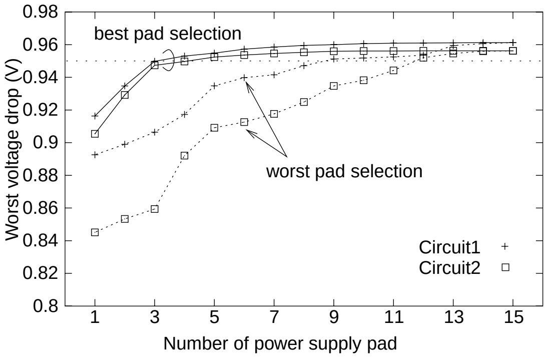


Fig. 5. The voltage drop for different pad combinations. Pin selections correspond to Table I. Poorly chosen pad set increases IR-drop significantly.

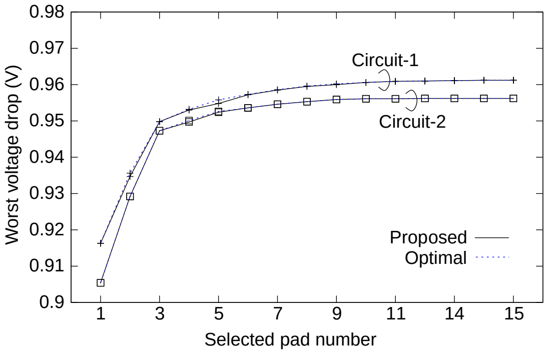


Fig. 6. Solution quality verification. The difference between pad assignment results using SPA and the optimal results by exaustive search is small.

selected from four different edges and located about the center of the edges since the resistor network structure used in circuit-1 is regular and power consumption spreads almost diagonally over the chip. For the circuit-2 with different resistor network but same current position, first 4 pads are the same. Figure 5 plots worst voltage drop as a function of number of selected pads. The sharp rise for the first 4 pads in best pad selection means that these are very critical for circuit-1. Table I also shows the example of ineffective pad selection order. Instead of picking up the best possible pad, we choose worst pad that least improved the worst voltage drop when adding a power supply pad. Compared with the best pad selection, inefficiency is obvious by first 4 selections to see that neighboring pads at the corner are selected in sequence. The worst voltage drop is accordingly large. The voltage drop differences between best and worst pad selection are 43 mV (110 %) and 88mV (200 %) for circuit-1 and -2, respectively with 3 pads. Here the error is defined as (voltage drop difference)/($V_{dd}-$voltage drop when all candidate pads are used).

If worst voltage drop target V_t were set to 50 mV, 5 pads are sufficient for the optimal pad selection for circuits-1 and -2. On the other hand, the poor pad selection requires 10 pads for circuit-1, and 12 pads for circuit-2. Required pad number differs much larger for larger circuits with more irregular shape and uneven current distribution. Combination of Tab. I and Fig. 5 also provides us with the critical pad for a particular power supply network and power consumption location. Pads P_{12}, P_2, P_8, and P_{15} are critically important pins. Selecting those pins compensates voltage drop significantly. We also find that assigning 5 power supply pads including critical pads is sufficient for these examples because the worst voltage drop is almost saturated beyond this point. Saturation of the voltage drop recovery curves in Fig. 5 suggests that there may exist many redundant power supply pad if they are assigned by human intuition only. We know that pad assignment is important and it has to be well considered when on-chip PDN is constructed.

We verified solution quality using assignment by exhaustive search (AES). Even for these small examples, the AES for all possible combination is too expansive since number of trials is sum of combination and it has an exponential dependence on n_p, which is $\sum_i^{n_p} \begin{pmatrix} n_p \\ i \end{pmatrix} = 2^{n_p}$. In circuit-1 and 2, total trials required for AES is 65,536 while SPA reduces it to 120. Figure 6 compares maximum voltage drop between the solutions obtained by the SPA and the AES. The largest difference is 1 mV and 0.4 mV respectively for circuit-1 and 2. This result confirms that the SPA calculates practically good solution in significantly reduced time.

For more practical examples, Tab. II compares computa-

TABLE II
COMPUTATIONAL TIME COMPARISON FOR VARIOUS CIRCUIT EXAMPLES.

circuit	#resistor	#node	# ports		SPA	SPA: CPU time (sec.)					random: CPU
			n_q	n_p	# trials	setup	step-1	step-2	step-3	total	total (sec.)
3	34080	22519	81	68	2278	3.1	1.4	12.8	3.0	20.3	306.0
4	170027	111767	81	68	2278	28.4	4.8	64.6	3.0	100.2	2257.6
5	417462	273896	81	68	2278	128.6	13.4	166.7	3.0	311.7	9656.0
6	12043	8340	200	100	4950	1.2	0.9	8.9	25.5	36.5	210.0
7	201000	121352	50	100	4950	39.2	6.4	77.8	6.0	129.4	4560.0
8	71097	43498	104	292	42486	9.8	2.4	71.0	360.5	443.7	3562.4
9	139288	91850	200	400	79800	28.7	4.4	215.6	2388.2	2636.9	13240.0

tional time of the SPA algorithm. Circuit 3 to 5 differs in modeling granularity. In circuit 3, PDNs in lower metal layers are reduced in order to make resistor network smaller. The number of observation points and candidate pads are equal for these three circuits. A column #trials indicates required number of simulations to find local optimum node required in the SPA. Setup is a time required to read PDN netlist and to formulate circuit matrices used as inputs in step-1 and step-2. Step 1 and 2 are DC analysis to determine reference pad and equivalent resistance. In these steps, we used general unsymmetrical version of sparse linear solver working on a 2.8 GHz processor [13, 14]. Even for a circuit with more than 400 K resistors, preparation for R_{eq} required only few minutes. Once R_{eq} is constructed, finding pad assignment order requires only 3 seconds and is independent of the original circuit size. Exploration of the pad assignment can be conducted very efficiently.

Circuits 6 through 9 are various examples with different observation points and pads. We see that number of observation points has weak impact on SPA calculation time but number of pads has strong effect. This comes from a fact that the SPA tries to find all pad order — number of trials to find a local optimum pad is $n_p(n_p-1)/2 \simeq \mathcal{O}(n_p^2)$. Being IMI as the innermost loop, overall calculation complexity becomes $\mathcal{O}(n_p^4)$. Although few hundred pads are sufficient for current pad assignment problem, further heuristics such as the divide and conquer approach described in [11] may be required to solve full pad ordering problem of the size $n_p > 500$. However, IMI can be combined with any of these heuristics.

The last column in Tab. II is estimated total calculation time for random pad order selection. Next pad is determined randomly from remaining candidates. Each time a new supply pad is added, the worst voltage drop is recalculated to see whether it satisfies target voltage. Simulation time required can be therefore estimated as $n_p \cdot (T_{setup} + T_{step1})$. SPA achieves up to 40x speed improvement in these examples and yield better solution for all cases.

VI. CONCLUSION

A pad assignment algorithm called SPA which minimizes voltage drop on a given power supply network is proposed. The proposed procedure enumerates both location and number of power supply pads. PDN reduction as equivalent resistance matrix which preserves both pads and on chip current sinks as ports efficiently calculates on-chip voltage drop and eliminates re-generation and decomposition of the circuit matrix. The incremental use of already obtained matrix inverse, SPA accelerates voltage drop calculation significantly. Experimental results showed significant speedup for exploring optimality of the assignment. Experiment also revealed that the inefficient pad selection increases number of power supply pads required.

REFERENCES

[1] SIA, *International Technology Roadmap for Semiconductors*, 2003 Edition, Assembly and Packaging; Interconnect.

[2] A. Dalal, L. Lev, and S. Mitra, "Design of an efficient power distribution network for the UltraSPARC-I^{TM}microprocessor," in *Proc. ICCD*, October 1995, pp. 118–123.

[3] A. Dharchoudhury, R. Panda, D. Blaauw, and R. Vaidyanathan, "Design and analysis of power distribution networks in PowerPC^{TM} microprocessors," in *Proc. DAC*, June 1998, pp. 738–743.

[4] J. Kozhaya, S. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," *IEEE Trans. on CAD*, vol. 21, no. 10, pp. 1148–1160, October 1998.

[5] Y.-M. Lee and C. C.-P. Chen, "The power grid transient simulation in linear time based on 3D alternating-direction-implicit method," in *Proc. DATE*, 2003, pp. 1020–1025.

[6] H. Qian, S. R. Nasif, and S. S. Sapatnekar, "Random walks in a supply network," in *Proc. DAC*, June 2003, pp. 93–98.

[7] M. Pedram, K. Chaudhary, and E. Kuh, "I/O pad assignment based on the circuit structure," in *Proc. ICCD*, October 1991, pp. 314–318.

[8] A. H. Ajami, , K. Banaerjee, A. Mehrotra, and M. Pefram, "Analysis of IR-drop scaling with implications for deep submicron p/g network designs," in *Proc. ISQED*, May 2003, pp. 35–40.

[9] M. Marek-Sadowska, "Pad assignment for power nets in VLSI circuits," *IEEE Trans. on CAD*, vol. 6, no. 4, pp. 550–560, July 1987.

[10] J. Oh and M. Pedram, "Multi-pad power/ground network design for uniform distribution of ground bounce," in *Proceedings Design Automation Conference*, June 1998, pp. 287–290.

[11] M. Zhao, Y. Fu, V. Zolotov, S. Sundareswaran, and P. Panda, "Optimal placement of power supply pads and pins," in *Proceedings Design Automation Conference*, June 2004, pp. 165–170.

[12] "GNU octave," http://www.octave.org/.

[13] O. Schenk, M. Hagemann, and S. Röllin, "Recent advances in sparse linear solver technology for semiconductor device simulation matrices," in *Proc. SISPAD*, September 2003, pp. 103–108.

[14] O. Schenk and K. Görtner, "Solving unsymmetric sparse systems of linear equations with PARDISO," *J. of Future Generation Computer Systems*, vol. 20(3), pp. 475–487, 2004.