Efficient Simulation of Power/Ground Networks with Package and Vias *

Jin Shi[1], Yici Cai[1], Xianlong Hong[1], Shelton X-D Tan[2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R.China
{shi-j03}@mails.tsinghua.edu.cn {caiyc, hxl-dcs}@tsinghua.edu.cn
[2] Department of Electrical Engineering, University of California at Riverside, CA 92521, USA
{stan}@ee.ucr.edu

Abstract. As the number of metal layers and the frequency of VLSI continue to increase, the voltage droop on both the package and vias is becoming more pronounced. This paper analyzes the numerical problem encountered in simulation of power/ground networks together with C4 package and via model. A new preconditioned iterative method and associated acceleration technique are introduced to overcome shortages of previous methods. The new method not only is effective to speedup simulation without loosing any accuracy, but also extends the usage of preconditioned method to general circuit simulation using only a few additional memories due to its MNA formulation.

1. Introduction

As the operating frequency of modern CPU and ASIC chips is rising rapidly according to Moore's law, the voltage droop on power/ground (p/g) networks must be considered for performance estimations. In addition, recent research has shown that if the package models are ignored in the design of p/g networks, the operation of the chip may suffer from special current surge conditions [1]. In order to maintain a reliable voltage supply, sometimes more than 30% chip area is used for the p/g network. As a result, under present manufacture process, usually the p/g network is assigned to more than 4 metal layers. The excessive use of metal layers leads to larger number of vias, which is hundred times more than before. Also, in many cases provided by industrial, the resistance of the vias can be as hundred times larger than the sheet resistance at the same metal layer. Therefore, today's analysis tool should take via resistance into account. Although modeling and simulation of the p/g network are time consuming computational processes, several efficient techniques have been proposed to tackle the problem [2] [3] [4] [5]. Among these methods, the *preconditioned Krylov-subspace iteration* method is one of the most efficient and stable strategies and it is widely used in today's p/g network simulation. However, existing methods using preconditioner-based scheme requires that the circuit matrix is symmetric positive definite, which is satisfied by *Nodal Analysis* (NA) formulation of the circuit matrix. But NA formulation requires admittance forms for all the circuit elements and is not as general as *Modified Nodal Analysis* (MNA) form. If we use a

* This work is supported by National Natural Science Foundation of China (NSFC) 60476014 and National Hi-tech R&D Program of China 2005AA1Z1230

MNA formulation and include the package model and vias, we can find that serious numerical problems are created in simulation.

In our paper, we study these new numerical characters and make some improvements over the classical *Krylov-subspace iteration* based method. The new preconditioned method extends the application of *incomplete choleskey decomposition* from NA form to MNA form. It also speeds up the iteration process by exploring the new numerical characters. Experiment results show that the new method is very efficient and stable for performing demanding simulation of p/g networks in the presence of package and a large number of vias. This paper is organized like this: Section 2 introduces the new p/g network model with package and vias and then analyzes its new numerical characters of the resulting networks; Section 3 briefly reviews the previous NA based methods and their shortages on solving the new problem; Section 4 introduces our new preconditioned method which can be used in MNA form with more efficiency; Section 5 describes a speed up technique based on the new character of the matrix; Finally, Section 6 summarizes the whole paper.

2. New Model of P/G Network and Its Numerical Characters

2.1 Model of C4 Package

For the complexity of VLSI grows rapidly, general wire-bond based package such as CPGA, PPGA is becoming difficult to afford the large quantities of both I/O and power/ground pins due to large power dissipation, it will introduce obvious package resistance and inductance. On the other hand, flip-chip package is becoming more attractive because of its good electrical performance [6]. In our paper, we use a popular C4 package model provided by our industry partners to simulate the voltage droop at each die bump. Figure 1 shows a model for two layer metal trace C4 package. Here each bump on the die connects to a power/ground pin on package through two metal layers and one via. L_{top} , R_{top} , L_{bot} and R_{bot} represent the inductance and resistance of package metal trace while L_{via} and R_{via} represent the inductance and resistance of package via. There also exist package decoupling capacitance, so C_{pkg} , R_{cpkg} , L_{cpkg} are used to represent its parameters.

2.2 Model of Power Rails and Vias

There are many models to describe the power rail, i.e., R or RLC lumped models, transmission line models. Usually, the length of power rail between two adjacent vias is not long enough to cause transmission-line effect in the operating frequency. But this is not always true [7]. This implies that in typical p/g network, all the metal rails in the die are modeled as resistors. Capacitance and inductance are only considered when they affect the power supply significantly, e.g., the decoupling capacitor and the inductance and capacitance of the package. However, most previous modeling of p/g network neglect the vias between two metal layers, this is partially because there is not so many metal layers before, or at that time the via resistance can be neglected

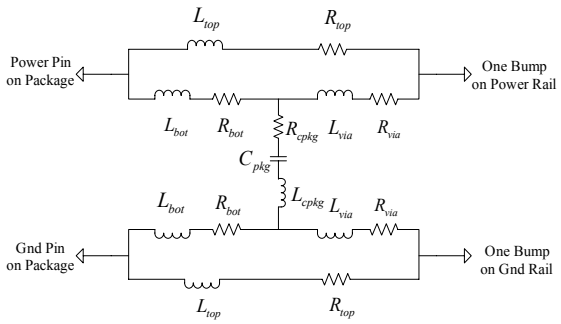

Fig. 1. Two metal layer C4 package model

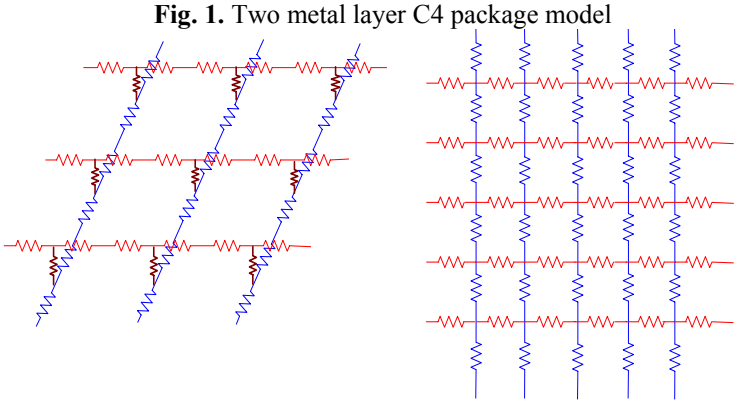

Fig. 2. P/G network models with and without vias

Table 1. Comparison of sheet resistance and via resistance

Metal Layer	Sheet Resistance m Ω	Metal Layer	Via Resistance m Ω
M7	22	M7-M6	300
M6	37	M6-M5	200
M5	55	M5-M4	800
M4	71	M4-M3	1250
M3	90	M3-M2	1300
M2	90		

when compared to sheet resistance. Table 1 gives typical sheet and via resistance according to an actual CMOS process. As we can see, via resistance exceeds the sheet resistance by more than 10x in layers close to M1. Other typical electrical parameters in different CMOS process can be found in [8]. Figure 2 illustrates our new model considering vias comparing with previous one. We can see in section 2.3 and section 5 that this different topology could introduce both good and bad characters in simulation. Also, this model is easy to extent to a RLC model.

2.3 Matrix Shape and New Numerical Characters

In this paper, we use MNA method to generate the circuit matrix. It is almost the same as the NA method except for dealing with independent voltage source and extra independent current variable. We'll introduce the matrix form in section 4. Here we

just focus on the topology, the stamp strategy and their effect on the matrix shape. According to MNA methods and trapezoidal discretization [9], any element in the p/g network would stamp a 'quadrangle fill' in the matrix, as shown in Figure 3. In our model, via is modeled as a resistor, so the connectivity of a node in the network is changed. Previously, each node is surrounded by four resistors in a mesh, two in the upper layer and two in the nether layer. But now, each node just connects to two resistors adjacent in the same layer as well as a via, as shown in Figure 2. Therefore, if we make the node sequence number increasing along with each rail, either horizontally or vertically in a certain layer, the matrix shape would be quite different as before which is shown in figure 4.

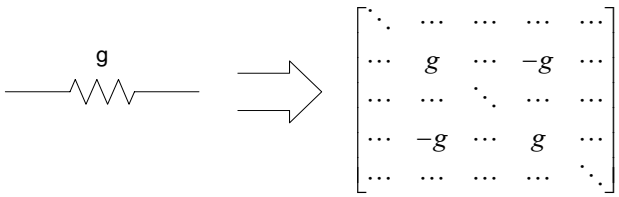

Fig. 3. 'Quadrilateral fill in' during matrix generation

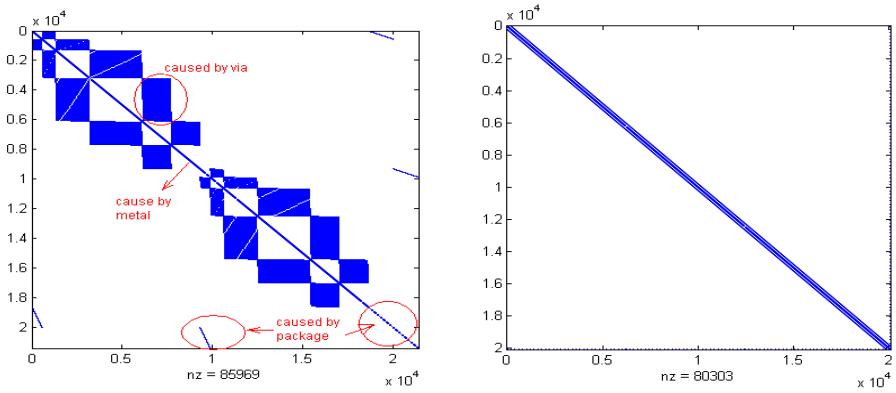

Fig. 4. Comparison between two typical simulation matrixes with and without vias and package

From Figure 4, first of all, we observe that the new matrix is wider in band obvious than previous one, which is usually a poor character in numerical simulation due to more fill-ins and slow convergence. Secondly, the distribution of the non-zero elements is quite different. In previous matrix, because of the mesh topology, non-zero elements caused by metal slices in power rails distribute around the main diagonal, not only stamp the main diagonal but also stamp other sub diagonals nearby. But now, the mesh topology is separated by vias, so the non-zero elements which represent metal slices only stamp in the main diagonal and two sub diagonals adjacent, while other non-zero elements caused by both package and vias are left outside the triple diagonals. This is an attractive numerical character which can be exploited (see Section 5). Thirdly, because both the resistance value of vias and package parameters after discreatization are magnitude away from the value of metal slices in power/ground rails, the distribution of large non-zero elements is extremely unbalanced. Figure 5 shows noticeable valleys near and far away from the main

diagonal, both of which can overwhelm the other elements in quantity. Also, the metal width usually decreases fast from the M7 to M2, which can cause the conductance varying dramatically in the diagonal of the matrix. This situation could become even worse as the number of layers increases. Figure 6 shows a typical numerical distribution in main diagonal. We can see that the maximum element is hundred times larger than the minimal one. All these unbalanced numerical distributions affect the eigenvalue of the matrix a lot according to *Gershgorin Circle Theorem*. Figure 7 compares the eigenvalue of two kinds of matrixes. It is obvious that the cluster effect becomes even worse and the max to min ratio of eigenvalue goes larger, both of which can directly lead to slow convergence rate of the iteration methods. Finally, because of the huge number of vias, the matrix is not as sparse as before.

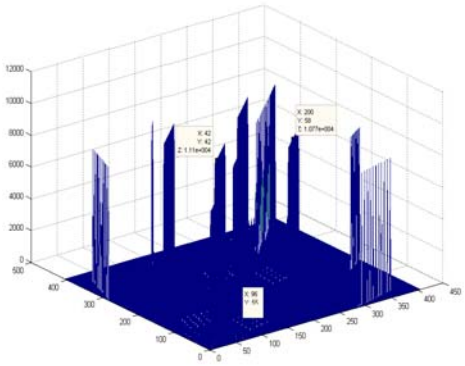

Fig. 5. Typical numerical distribution of new matrix

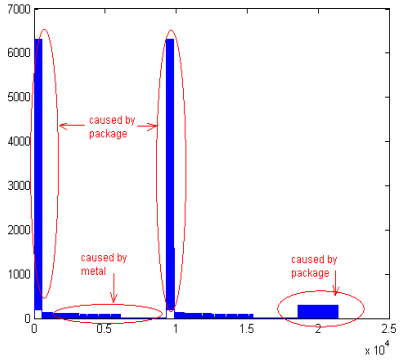

Fig. 6. Numerical distribution in main diagonal

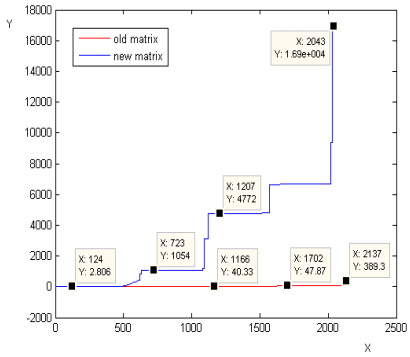

Fig. 7. Comparison of two kinds of matrix on eigenvalue

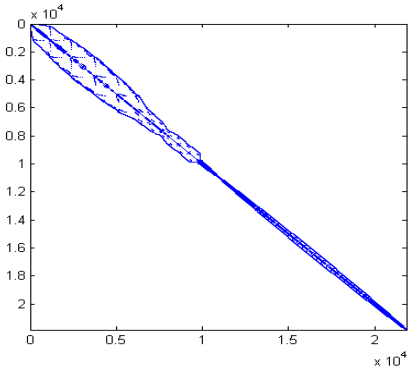

Fig. 8. Matrix after RCM ordering

3. Shortages of Previous Method

Many methods are constructed to perform efficient simulation of p/g network in the past. Among them the *Krylov-subspace iteration* method [2] is proved efficient, accurate and stable by many industrial cases, while the random walk method [4] is

criticized for its accuracy and the ADI method [3] is criticized for its stability. Without considering the package parameters and vias, we do not see such a stiff eigenvalue problem. So the system is relative easy to solve using iteration method. However, because method in [2] highly relies on its preconditioner, so changes in matrix shape and numerical characters affect its speed a lot. Usually, two kinds of preconditioners are widely used in this method, one is *incomplete LU decomposition* (ILU), and the other one is *incomplete choleskey decomposition* (ICD). Table 2 lists the iteration times of two kinds of matrixes. Here matrix I is a 12Kx12K matrix with package and vias while matrix II is a 12Kx12K matrix without package and vias. 'Zero fill in' *Preconditioned Conjugate Gradient* (PCG) method is used as the basic iteration method and both two preconditioner are tested. We can see that the inclusion of package and vias increases the iteration times a lot. Another shortage of previous iteration method is that CG method preconditioned by ILU and ICD can not be used to indefinite matrix directly, which limits it to circuits formulated in NA form [2]. This is a fatal weakness of it, which is widely criticized because general circuit simulation requires MNA formulation.

Table 2. Comparison of the iteration times

ILU	Iteration Times		ICD	Iteration Times	
residual	Matrix I	Matrix II	residual	Matrix I	Matrix II
1e-3	165	5	1e-3	166	5
1e-6	258	8	1e-6	258	8
1e-10	377	11	1e-10	375	11

4. New Preconditioned Method Fitting MNA

If the matrix is constructed in MNA form, its shape is given in (1). Here matrix G is a $(n+m)\times(n+m)$ matrix. A is a *sparse positive definite* (s.p.d) $n\times n$ M matrix while B is an $m\times n$ sparse matrix with rank equaling to m

$$G=\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \qquad G'=\begin{bmatrix} A & B^T \\ B & kI \end{bmatrix} \tag{1}$$

Here matrix B is caused by the independent voltage source (pads) and extra independent variable such as the current of inductor. Usually, a lot of bumps exist in C4 package, which can cause m easily to be more than one thousand according to our package model. Also, it causes m zero entries in the main diagnoal. So now matrix G is a indefinite matrix which has m negative eigenvalue. Although we can use another matrix G' to appoximate G and get preconditioner using ILU and ICD, the effectiveness of these preconditioner is degraded becuase m is realtive large. We begin our improvement with ICD method as it uses much less memory than ILU method while shares the same convergence speed.

Theorem 1: $L=\begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix}$ and $U=\begin{bmatrix} l_{11}^T & l_{21}^T \\ 0 & -l_{22}^T \end{bmatrix}$ are perfect preconditioners of G matrix, where l_{11} is the choleskey decomposition of matrix A, $l_{21}=B\cdot l_{11}^{-T}$, l_{22} is the

choleskey decomposition of matrix $l_{21} \cdot l_{21}^T$

Proof: Because l_{11} is the choleskey decomposition of matrix A, $l_{11} \cdot l_{11}^T = A$ holds. Also $l_{22} \cdot l_{22}^T = l_{21} \cdot l_{21}^T$ holds (here l_{21} is an mxn matrix while l_{22} is an mxm matrix). Choose two identity matrix I (nxn) and $I^{'}$ (mxm), considering $l_{21} \cdot l_{11}^T = B$, then we can get (2)-(3) below. Because L and U are triangular matrixes full in rank, so (2) can be rewritten as (3). This means that using L as left pre-multiplier and U as right pre-multiplier, the condition number of iteration matrix can be reduced to 1, so L and U are perfect preconditioners in theory. Also, the replacement of l_{11} and l_{11}^T by LU factors causes no affection to this theorem.

$$L\begin{bmatrix} I & 0 \\ 0 & -I^{'} \end{bmatrix} U = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & -I^{'} \end{bmatrix} \cdot \begin{bmatrix} l_{11}^T & l_{21}^T \\ 0 & -l_{22}^T \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & -l_{22} \end{bmatrix} \cdot \begin{bmatrix} l_{11}^T & l_{21}^T \\ 0 & l_{22}^T \end{bmatrix} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} = G \quad (2)$$

$$L^{-1} \cdot G \cdot U^{-1} = \begin{bmatrix} I & \\ & -I^{'} \end{bmatrix} \quad (3)$$

Theorem 2: Both the time and memory cost to generate L and U is low.

Proof: a) In practical, for A is an M matrix, the *incomplete choleskey decomposition* is guaranteed to exist [10]. So we can use it as approximate decomposition. Here both zero fill in strategy or threshold guided fill in strategy can be used to control the memory usage [10]. $l_{21} \cdot l_{21}^T$ is an s.p.d matrix further because of the connectivity of the pads and other independent variables, it is also a diagonal dominant matrix, so the incomplete choleskey decomposition exists too.

b) Usually, l_{11}^{-T} is not a sparse matrix, if we have to generate the full matrix, the memory usage is prohibited when n is very large. Luckily, we only need to compute $l_{21} = B \cdot l_{11}^{-T}$. Because the B matrix is very sparse (no more than two elements per row), and elements in l_{11}^{-T} can be known in advance without iteration as it is a triangular matrix, just little extra memory is needed when generating l_{21}.

c) Computation cost of l_{21} is low

$$p(i,i) = \frac{1}{l_{11}(i,i)} \qquad l_{11}^{-1}(i,j) = \begin{cases} i = j & p(i,i) \\ i < j & 0 \\ i > j & -p(i,i) \cdot \sum_{k=j}^{i-1} l_{11}(i,k) \cdot p(k,j) \end{cases} \quad (4)$$

$$l_{21}(i,j) = \sum_{b(i,k) \neq 0} b(i,k) \cdot l_{11}^{-T}(k,j) = \sum_{b(i,k) \neq 0} b(i,k) \cdot l_{11}^{-1}(j,k) \quad b(i,k) \in B \quad (5)$$

Theorem 3: *Symmetrical Lanczos* (symmlq) method [11] is a good method to solve our indefinite system using approximate preconditioner L and L^T

Proof: Because G is a symmetrical indefinite matrix, *Ritz Minimization principle* do not hold any more, which means the *cg* method can not be used to get the solution. Instead, *symmlq method* can solve the project system without minimize anything [11]. Further, *symmlq method* is more stable for matrixes that have extremely eigenvalue. However, *symmlq method* requires positive definite matrix as its preconditioner, direct use of matrix U as a preconditioner will cause problem, but if we use L^T, this problem is overcome. Although *Quasi-Minimal Residual method* (QMR) can take indefinite matrix as its preconditioner, the computation cost is much higher than *symmlq method* according to our experiments. Later, we can see that, by using a speed up technique, the approximate precondioner almost loses little in efficiency.

5. Balance Technique to Speed up

Usually, threshold guided fill in strategy is used to speed up the iteration. It is a trade off between speed and memory usage. The more fill in we allowed, the faster convergence speed it will reached. Although *Reverse Cuthill-McKee* (RCM) *ordering* [12] and *Minimum Degree Ordering* [13] techniques can be used to partially reduce the fill in and compress the matrix in the band form, they almost can not improve the convergence speed under the same threshold in our application as shown in Table 3. The original reason to use ordering is that under the same memory usage constraint, smaller threshold can be reached. However, in our new method below, we can see that this shortage is overcome. Figure 8 shows the shape of matrix in Figure 4 after RCM ordering, and Table 3 gives brief statistic information about the memory usage and the iteration times. The fill in ratio there is the memory usage of preconditioner divided by the half memory usage of the original matrix.

In section 2, we have analyzed the matrix numerical character and find that the poor eigenvalue is partially caused by the unbalanced distribution of large non-zero elements. Because the new topology has made the metal slice just to stamp in the triple diagonals only, we can utilized this character to reduce the condition number of this matrix which can lead to speed up directly. Our strategy is based on two steps. In the fist step, we reduce the value of non-zero elements outside the triple diagonals using balance transform B. In the second step, we raise the value of non-zero elements in the triple diagonals using balance transform A, as shown in Figure 9. If we design the transform carefully, we can guarantee that the original resistor is equivalent to the transformed two ports network using circuit theory. So no accuracy is sacrificed except introducing some virtual nodes. Because all non-zero elements in admittance matrix is caused by stamp of conductors (caused by metal, via, inductance and capacitance), an ideal situation is that if there is no memory usage constraint, the balance transform will cause the non-zero elements in the triple diagonals to become dramatically larger than that outside the triple diagonals. Therefore the admittance matrix can be treated as a triple diagonal matrix and become very easy to solve. However, this character does not exist in traditional mesh structure p/g network for elements do not just stamp in the triple diagonals of the matrix, and the balance transform even makes the condition number of the old matrix to become worse. In practical usage, sophisticated adaptive algorithm is generated to speed up the solving process considering both the package and via parameters and the memory usage.

Algorithm Name: adaptive balancer

Input: topology of the p/g network, electrical parameters, simulation time step, assistant memory N (T<N<kT). T is original node number, k is a constant.

Output: balanced matrix

1) compute typical conductance of metal slice and via g_m , g_v in all layers
2) discrete in time domain, get typical g_l , g_c
3) sort all typical conductance and get the maximal typical conductance $g_{\max}$
4) using balance transform B to reduce elements' value out of triple diagonals
5) for(i=1;i<=layer number;i++)

   {add p_i virtual nodes between two vias using balance transform A

   n_i is the number of actual nodes in layer i

$$p_i = \min\left\{\left\lceil \frac{N}{T}\cdot e^{-\frac{n_i}{T}} \right\rceil, \left\lceil \frac{g_{\max}}{g_{mi}} \right\rceil\right\} \qquad T = \sum_1^k n_i \}$$

6) output the matrix

In this algorithm, if the assistant memory N we allocated is not enough to perform an ideal balance, layers which have less metal slices will get the higher priority to be balanced. This algorithm is implemented using C++, and is tested on a Linux Workstation with 1 GHz CPU and 512M memory. Table 4 gives out the performance of this algorithm. In left part of the table we use MNA formulation and do iteration in symmlq form using our new preconditioner together with RCM ordering and balance techniques while in right part of the table ordinary pcg method in NA formulation and ICD precondition is used. From the table we can see that new algorithm is one magnitude faster than general method while the memory usage just rises slightly.

Also, the small average iteration times proved that our new precondioner could fit MNA formulation very well. Further, if the scale of power/ground network and the dense of the bumps increase, or RLC model is used, we can get more benefit by using this strategy as both of those models will cause worse condition number.

Table 3. Comparison between odering and no odering

No odering residual=1e-6		
threshold	Fill in ratio	Avg Iter times
1e-1	1.5	221
1e-2	2.79	93
1e-3	10.96	44
1e-4	48.41	17
Odering residual=1e-6		
threshold	Fill in ratio	Avg Iter times
1e-1	1.1	220
1e-2	2.32	95
1e-3	6.93	46
1e-4	18.55	17

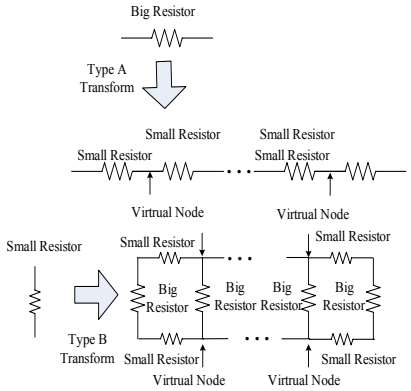

Fig. 9. Balance Transform A & B

6. Conclusion

In this paper, we have analyzed numerical characters of circuit matrices in the presence of package and massive vias in high-performance p/g networks. We then proposed a new preconditioned method, which can extent the preconditioned based iterative method to MNA form. Also by utilizing special characters, a novel balance technique was generated which can improve iteration convergence rate under a certain threshold without any accuracy loss. The experiment results show that this method is really efficient and stable while its memory usage is reasonable.

Table 4. Performance of the speed up technique

Node Num	Avg Iter times	CPU time (sec.)	Mem Usage (MB)	Speed Up Ratio	Node Num	Avg Iter times	CPU time (sec.)	Mem usage (MB)
5.5K	15	0.53	65	3.18	5.5K	208	1.69	59
23K	19	1.13	78	9.20	22K	293	10.4	70
130K	30	5.37	148	13.01	100K	456	70.2	136
450K	40	21.2	366	19.79	450K	710	419.6	349

Reference

1. Howard H. Chen and David D. Ling: Power Supply Noise Analysis Methodology for Deep-Submicron VLSI Chip Design, DAC97 Proceedings, 638-643
2. T.Chen and C.C.Chen: Efficient Large-Scale Power Grid Analysis Based on Preconditioned Krylov-Subspace Iterative Methods, DAC01 Proceedings, 559-562
3. Weikun Guo, Sheldon X. D. Tan: Circuit Level Alternationg-Direction-Implicit Approach to Transient Analysis of Power Distribution Networks, International Conference on ASIC Proceedings, 2003, Beijing, 246-249
4. Haifeng Qian, Sani R. Nassif, Sachin S. Sapatnekar: Random Walks in a Supply Network, DAC2003 Proceedings, 93-98
5. J.N.Kozhaya, Sani R.Nassif and Farid N.Najm: A Multigrid-Like Technique for Power Grid Analysis, IEEE Trans. Computer-Aided Design, vol.21, no.10, Oct. 2002, 1148-1160
6. Performance Characteristics of IC Packages, Intel Corp., 2000 http://www.intel.com/design/packtech/ch_04.pdf
7. Howard Johnson, Martin Graham and etc.: High-Speed Digital Design a Handbook of Black Magic, Pearson Education Press. 1997
8. Michael John, Sebastian Smith: Application-Specific Integrated Circuits, Pearson Education Press, 1997
9. L. W. Nagel: SPICE2: A Computer Program to Simulate Semiconductor Circuits, ERL Memo. No. UCB/ERL Vol M75/520 (1975)
10. Saad, Yousef, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, 1996.
11. Barrett, R., M. Berry, T. F. Chan, et al., Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, 1994
12. George, Alan and Joseph Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, 1981.
13. Amestoy, P. R., Davis, T. A., and Duff, I. S.: An approximate minimum degree ordering algorithm. SIAM J. Matrix Anal. Applic. 17, 4, 886,905.