Transient Analysis of On-Chip Power Distribution Networks Using Equivalent Circuit Modeling[*]

Zhu Pan[1], Yici Cai[1], Sheldon X.-D. Tan[2], Zuying Luo[1], Xianlong Hong[1]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, P.R.China
[2]Department of Electrical Engineering, University of California at Riverside, Riverside CA, USA
stan@ee.ucr.edu

Abstract

This paper presents an efficient method to analyze power distribution networks in the time-domain. Instead of directly analyzing the integration approximated power/ground networks at each time step as previous methods did, the new method first builds the equivalent models for many series RLC-current chains based on their Norton's form companion models in the original networks, and then the Precondition Conjugate Gradient (PCG) based iterative method is used to solve the reduced networks. The solutions of the original networks then are back solved from that of the reduced networks. Our contribution is the introduction of an efficiency algorithm for reducing RLC power/ground network complexities by exploitation of the regularities in the power/ground networks. Experimental results show that the complexities of reduced networks are typically significantly smaller than that of the original circuits, which makes the new algorithm extremely fast. For instance, power/ground networks with more than one million branches can be solved in a few minutes on modern Sun workstations.

1. Introduction

Signal integrity in the power/ground (P/G) bus is emerging as a limiting factor in the nano-regime VLSI chip designs. As P/G grids experience the largest current flows in a chip, they are more susceptible to current-induced reliability and functional failures. Efficient transient analysis techniques of P/G grids are required to precisely capture the dynamic voltage fluctuations on the P/G wires for guiding the designs of a reliable and robust P/G networks. As millions of devices are integrated into a single chip, transient analysis of a power/ground network becomes a challenging task due to the increasing sizes of the networks. The traditional circuit simulators like SPICE are no longer able to meet the formidable tasks of analyzing P/G circuits with millions of extracted RLKC elements in a timely manner.

Due to the increasing importance of P/G integrity issues such as excessive IR drops, Ldi/dt noise, electro-migration and resonance issues, significant research efforts have been carried out to find efficient simulation approaches to P/G grid analysis [1-9] in the past. Among them, multi-grid method [8], hierarchical method [10], preconditioned conjugate gradient (PCG) [2], hierarchical model order reduction [3], and frequency domain analysis [1] are the latest methods proposed. Although significant improvements have been made to analyze large P/G grids by those proposed methods, the naturally regular physical structures of a P/G network is not fully explored to speedup transient analysis of networks.

Recently tree-mesh structured P/G circuits are analyzed in [10]. The tree portions of the P/G network are reduced via model order reduction method PRIMA[16] to obtain the equivalent circuits of RLC tree circuits. But some errors will be introduced. The methods in [4, 11] fully explore the naturally regular structures of a P/G network to reduce the P/G network complexities in an error-free fashion. But they can only be applied for DC analysis of resistor-only P/G networks.

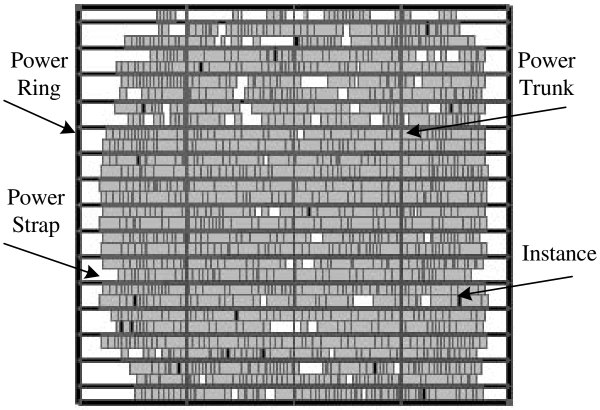

Fig.1. A P/G network of Standard Cell based ASIC layout

In this paper, we propose a new approach to the transient analysis of large P/G grids. Our new method is based on the observation that P/G grids, especially those used in many cell-based layout ASIC applications, consist of any series RLC chains [11-13] as shown in Fig. 2. We show that a compact equivalent model can be built for the

[*] This work is supported by the National Natural Science Foundation of China 60176016 and 90307017 and by the Senate Research funds of the University of California.

0-7695-2093-6/04 $20.00 © 2004 IEEE

IEEE COMPUTER SOCIETY

RLC chain circuit in each time step of approximate integration. With the equivalent chain models, the original networks can be reduced significantly and efficient transient analyses of vary large P/G networks become possible. As only resistors and constant current sources are present in the reduced P/G networks in each time step, the resulting circuit matrix of the network is symmetric positive definite (the voltage sources are also modeled by Norton equivalent circuits), preconditioned conjugate gradient based iterative method [14] can be used to solve the resulting circuit matrix very efficiently. Our contribution is the introduction of an efficiency algorithm for reducing RLC P/G network complexities in error-free manner by exploitation of the regularities in the power/ground networks. Experimental results show that at least two-order of magnitude speed up over SPICE is observed for large P/G networks.

This paper is organized as follows. Section II presents how equivalent circuit models are constructed for RLC chain circuits in P/G networks. Section III gives the general flow of the proposed algorithm. Experimental results are described in Section IV. Section V concludes the paper.

2. Construction of Equivalent Circuit Models

The P/G networks are typically mesh-structured in today's VLSI technology and consist of many cascaded RLC sections extracted from chip layouts as shown in Fig. 2 for a power network.

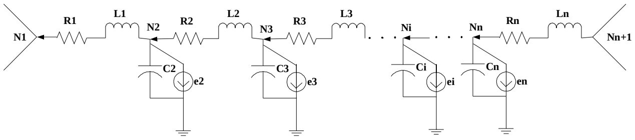

Fig. 2. A series RLC chain in a P/G network

Here R_i, C_i, L_i, denote, respectively, the resistance, capacitance, and inductance at P/G wire segment *i*. Capacitance C_i includes both the parasitic capacitances of the P/G wire *i* and the decoupling capacitances around. e_i is a time-varying current source that captures the dynamic current consumptions of circuit cells.N_1 and N_{N+1} are called *cross nodes* and rest of the nodes are called *intermediate nodes*. Our goal is to suppress all the intermediate nodes in this series RLC chain circuit and replace them by an electrically equivalent circuit that consists of only the cross nodes in each integration time step of transient analysis.

A. Integration Approximation In Norton's Form

Let *h* be the time step used at simulation step *k+1* as shown in Fig.3. For the capacitors and inductors, trapezoidal companion models of Norton's form are used. In this way, we will not introduce any extra nodes. Specifically, the current across a capacitor at *k+1* step is:

$$I_{c,k+1} = \frac{2C}{h} V_{c,k+1} - (\frac{2C}{h} V_{c,k} + I_{c,k}), \quad (1)$$

where $V_{c,k}, I_{c,k}, V_{c,k+1}, I_{c,k+1}$ denote the branch voltage and branch current of the capacitor on step k and step *k+1* respectively and *C* is the value of the capacitor.

Similarly the current through an inductor at step *k+1* is:

$$I_{L,k+1} = \frac{h}{2L} V_{L,k+1} + (\frac{h}{2L} V_{L,k} + I_{L,k}), \quad (2)$$

where $V_{L,k}, I_{L,k}, V_{L,k+1}, I_{L,k+1}$ denote the branch voltage and branch current of the inductor at step k and step *k+1* respectively and *L* is the value of the inductor.

Let's go back to the series RLC chain and mark each capacitor and inductor with the integration time step index at *k+1* step, the RLC chain between two cross nodes N_1 and N_{N+1} in Fig.3 become Fig.4. When all the capacitors and inductors are replaced by their companion models at time step *k+1*, $E_{i,k+1}$ denotes the current flowing into node *i* from node *i+1* on step *k+1*.

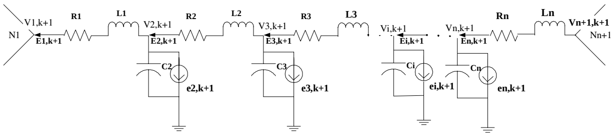

Fig. 3. The RLC chain circuit at time step k+1

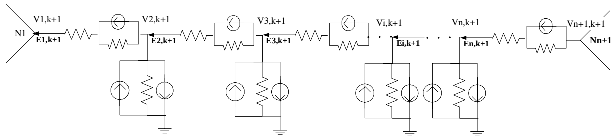

Fig. 4. The RLC chain circuit in terms of companion models at time step k + 1

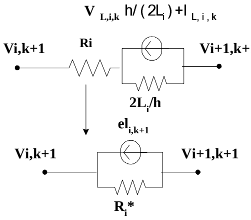

Fig. 5. Merge two resistors in a RLC section

Next, we merge two floating resistors for each RLC section as shown in Fig. 5 using Norton theory. Then we have:

$$R_i^* = 2L_i/h + R_i, \quad (3)$$

$$el_{i,k+1} = \left(\frac{h}{2L_i} V_{L,i,k} + I_{L,i,k}\right) \cdot \frac{2L_i/h}{2L_i/h + R_i}, \quad (4)$$

where $V_{L,i,k}$ and $I_{L,i,k}$ are the branch voltage and branch current of L_i at step k respectively. They can be represented with node voltages and branch currents.

$I_{L,i,k} = E_{i,k}$, $V_{L,i,k} = (V_{i+1,k} - V_{i,k}) - E_{i,k} R_i$.

We then combine two grounded current sources, one is coming from the independent current source and another is coming from capacitor's companion model, into one current source $ec_{i,k+1}$ for each RLC section:

$$ec_{i,k+1} = e_{i,k+1} - \left(\frac{2C_i}{h} V_{i,k} + I_{i,k} \right), \tag{5}$$

and rename the resistor from C$_i$'s companion model as $r_i = \frac{h}{2C_i}$. So the circuit in Fig. 4 can be simplified to the following circuit shown in Fig. 6, which is more ready for further reduction to be explained in the next subsection.

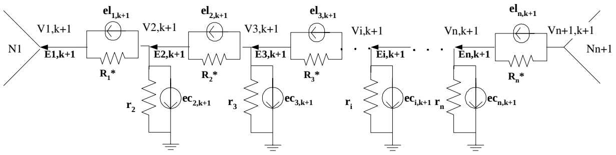

Fig.6. Simplified RLC chain circuit

B. Transformation From Y Model to π Model

In this subsection, we show how a RLC chain circuit shown in Fig.6 can be further reduced by an equivalent circuit consisting of only the cross nodes. This is done by repeatedly transforming a **Y** model circuit to a π model circuit as shown in Fig. 7:

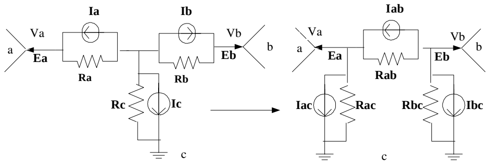

Fig.7. Y model circuit to π model circuit

In Fig. 7, E_a and E_b are the currents inflowing into node a and node b respectively, and the corresponding arrowheads show the current directions. To compute the equivalent currents and resistances shown in the right-hand side of Fig. 7, we first look at node a and we have:

$$V_b - V_a = R_a(E_a - I_a) + R_b(E_a + I_c + \tfrac{V_a + R_a(E_a - I_a)}{R_c} + I_b)$$
$$= \left(R_a + R_b + \tfrac{R_a R_b}{R_c}\right) E_a + R_b I_c + R_b I_b + \tfrac{R_b}{R_c} V_a - R_a I_a - \tfrac{R_a R_b}{R_c} I_a.$$

We define the following equivalent resistors:

$$R_{ab} = \frac{R_a R_b + R_a R_c + R_b R_c}{R_c}, \tag{6}$$

$$R_{ac} = \frac{R_a R_b + R_a R_c + R_b R_c}{R_b}, \tag{7}$$

$$R_{bc} = \frac{R_a R_b + R_a R_c + R_b R_c}{R_a}. \tag{8}$$

As a result, we have

$$E_a = \frac{V_b - V_a}{R_{ab}} - \frac{V_a}{R_{ac}} + \left(\frac{R_a I_a}{R_{ab}} - \frac{R_b I_b}{R_{ab}} \right) - \left(\frac{R_c I_c}{R_{ac}} - \frac{R_a I_a}{R_{ac}} \right)$$

Similar result can be obtained for node b:

$$E_b = \frac{V_a - V_b}{R_{ab}} - \frac{V_b}{R_{bc}} - \left(\frac{R_a I_a}{R_{ab}} - \frac{R_b I_b}{R_{ab}} \right) - \left(\frac{R_c I_c}{R_{bc}} - \frac{R_b I_b}{R_{bc}} \right)$$

We further define the following equivalent currents:

$$I_{ab} = \frac{R_a I_a}{R_{ab}} - \frac{R_b I_b}{R_{ab}}, \tag{9}$$

$$I_{ac} = \frac{R_c I_c}{R_{ac}} - \frac{R_a I_a}{R_{ac}}, \tag{10}$$

$$I_{bc} = \frac{R_c I_c}{R_{bc}} - \frac{R_b I_b}{R_{bc}}. \tag{11}$$

Finally we can represent E_a and E_b in terms of the those equivalent currents and resistances as follows:

$$E_a = \frac{V_b - V_a}{R_{ab}} - \frac{V_a}{R_{ac}} + I_{ab} - I_{ac}, \tag{12}$$

$$E_b = \frac{V_a - V_b}{R_{ab}} - \frac{V_b}{R_{bc}} - I_{ab} - I_{bc}. \tag{13}$$

Equations (12) and (13) essentially gives us the π model representation of the same circuit as shown in the right-hand side of Fig. 7.

C. Construction and Back Solving Algorithms for the Equivalent Circuits

By repeatedly applying the **Y** model to π model transformation starting from node N_1 until we reach node N_{n+1}, the whole chain circuit in Fig. 6 can be reduced into an equivalent circuit shown in Fig. 8:

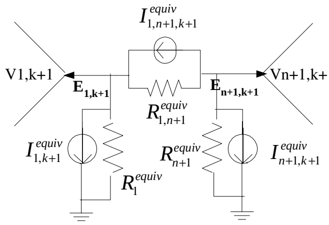

Fig. 8. The equivalent chain circuit model

The algorithm for computing R^{equiv}_1, R^{equiv}_{n+1}, $R^{equiv}_{1,n+1}$, $I^{equiv}_{1,k+1}$, $I^{equiv}_{n+1,k+1}$, and $I^{equiv}_{1,n+1,k+1}$ from the circuit in Fig. 6 is shown in Fig.9.

Once the reduced network has been solved, all the intermediate nodes of original circuits can be back solved using the superposition principle. As shown in Fig.6, for voltage $V_{i,k+1}$ at the node *i,* when the current flowing through the resistor R^*_i is obtained, the voltage $V_{i+1,k+1}$ at node *i+1* is equal to $V_{i,k+1}$ plus the voltage drop on R^*_i. The current flowing through the resistor R^*_{i+1} can be computed simply with KCL law. The detailed algorithm for commuting all the node voltages of intermediate nodes

$V_{j,k+1}$ ($j\in[2,N]$)is shown in Fig. 10.

3. OVERALL ALGORITHM

A. Algorithms Description

After the reduced network is obtained at each time step, preconditioned conjugate gradient (PCG) based iterative method [14] is used for the solutions as the resulting circuit matrix is symmetric positive definite and the circuit consists of only resistors and constant current sources. Assume initial node voltages are known and the number of RLC sections in a P/G network is *M*. The overall simulation algorithm is shown in Fig. 11:

Construct Equivalent Circuit Model
EQVCKT-CONSTRUCT()

```
{
    $R_1^{equiv} = \infty$; $I_{1,k+1}^{equiv} = 0$; $r_{n+1} = \infty$; $ec_{n+1,k+1} = 0$;
    $R_a = R_1^*$; $R_c = r_2$; $I_a = el_{1,k+1}$; $I_c = ec_{2,k+1}$;
    $j=2$;
    While j<=n do
    {
        $R_b = R_j^*$; $I_b = -el_{j,k+1}$;
        Compute $R_{ab}, R_{ac}, R_{bc}$ based on (6)-(8);
        Compute $I_{ab}, I_{ac}, I_{bc}$ based on (9)-(11);
        $j=j+1$;
        $R_a = R_{ab}$; $R_c = R_{bc} \| r_j$; $R_1^{equiv} = R_1^{equiv} \| R_{ac}$
        $I_a = I_{ab}$; $I_c = I_{bc} + ec_{j,k+1}$; $I_{1,k+1}^{equiv} = I_{1,k+1}^{equiv} + I_{ac}$;
    }
    $R_{n+1}^{equiv} = R_c$; $I_{n+1,k+1}^{equiv} = I_c$; $R_{1,n+1}^{equiv} = R_a$; $I_{1,n+1,k+1}^{equiv} = I_a$;
}
```

Fig.9. Algorithm for constructing equivalent circuit model

Back Solve for the Intermediate Node Voltages
BACK-SOLVER()

```
{
    $E_{1,k+1} = \frac{V_{n+1,k+1} - V_{1,k+1}}{R_{1,n+1}^{equiv}} + I_{1,n+1,k+1}^{equiv} - I_{1,k+1}^{equiv} - \frac{V_{1,k+1}}{R_1^{equiv}}$;
    for j=2 to n do
    {
        $V_{j,k+1} = V_{j-1,k+1} + R_{j-1}^*(E_{j-1,k+1} - el_{j-1,k+1})$;
        $E_{j,k+1} = E_{j-1,k+1} + \frac{V_{j,k+1}}{r_j} + ec_{j,k+1}$;
    }
}
```

Fig.10. Algorithm for back solving intermediate node voltages.

Overall Simulation Algorithm
PG-Solver()

```
{
    Initiate $V_i, R_i^*, r_i$ for $i\in[1,M]$;
    for k=1 to $n_{step}$ do
    {
        RENEW($ec_{i,k}, el_{i,k}$);
        For all RLC chains EQVCKT-CONSTRUCT();
        PCG- SOLVER () ;
        For all RLC chains BACK - SOLVER ();
    }
}
```

Fig.11. Overall simulation algorithm

n_{step} is the number of time steps used in the simulation. RENEW($ec_{i,k}$, $el_{i,k}$) is to compute the combined current sources from previous simulation results according to (4) and (5). EQVCKT-CONSTRUCT constructs the equivalent circuits for all the RLC chains as shown in Fig.9. PCG-SOLVER is the linear solver based on PCG algorithm for the solutions of the reduced network.. BACK-SOLVER back solves the node voltage of all intermediate nodes shown in Fig.10.

B. Time Complexity Analysis

Suppose a P/G network with *N* nodes, among them, N_{cross} is the number of cross nodes and N_{mid} is intermediate nodes and $N= N_{cross} + N_{mid}$. Typically N_{cross} is far less than *N* in standard cell based P/G networks of ASICs.

From previous subsection, we notice that EQVCKT-CONSTRUCT and BACK-SOLVER are of linear complexity. For the PCG-SOLVER, it was shown in [2,13] that it practically shows linear time or close to linear time complexity for many practical P/G networks due to sparsity of those circuits. As a result, given N_{cross} is far less than N_{mid} , the whole algorithm which consists of PCG-SOLVER(N_{cross}), EQVCKT-CONSTRUCT(*N*) and BACK-SOLVER(N_{mid}), is of linear or close to linear time complexity. This is illustrated in Fig. 12.

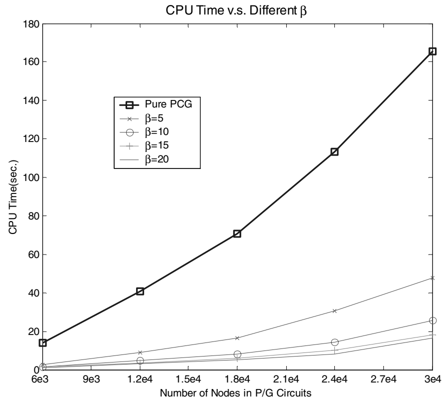

Fig.12. Time complexity comparison

In Fig. 12, we define β= N/ N_{cross}, which reflects the reduction ratio between original P/G grid and reduced P/G grid, and draw CPU time v.s number of circuit nodes curves for pure PCG and new method. Four networks with different β are solved for each total number of nodes. The CPU time for the Pure PCG method (no node reduction is performed) is the average time used for the four circuits as the CPU time of Pure PCG method does not change with β significantly. It is shown that the CPU time used by new algorithm grows almost linearly when $\beta \geq 10$. Also, as more reduction become possible with larger β , the increase rate of the CPU time with the number of nodes decreases.

4. Experimental Results

The proposed simulation algorithm is implemented in *C* and *C++*. All the experimental results are obtained on *SUN UltraSparc* workstation with 750MHz CPU and 1GB memory. The number of time steps, n_{step} is assigned 120 for a clock cycle according to the requirement of accuracy for both new algorithm and SPICE [8].

We tested our program on a number of P/G networks with complexities ranging from 2500 nodes to 1 million nodes. The statistics of those P/G circuits are shown in Table I.

First, we compare our method with that of SPICE in terms of accuracy. Both programs are used to solve a 50*50*5[1] P/G network. The results are shown in Fig.13. The two waveforms are essentially same. The maximum error is just 0.00289%.

Next, we compare our new simulation method with the SPICE and the Pure PCG algorithm. The effectiveness of equivalent circuit modeling is shown in Table I where the number of nodes after node reduction and the reduction ratio for each circuit are shown in column 3 and 4. It can be seen that complexities of the P/G network can be significantly reduced. On the other hand, we notice that the topologies of P/G grids do have impacts on the performance of the new algorithm.

Table I and Table II show that the pure PCG can't deal with much large P/G networks. This is because PCG will become memory expensive when large circuits are loaded. On the other hand, the new algorithm is much more memory efficient and powerful than the Pure PCG method.

From Table II, it is seen that the new algorithm is one order of magnitude faster than the pure preconditioned conjugate gradient method (PCG) and two orders of magnitude faster (50X-233X) than SPICE. The speedup of the new algorithm is comparable with the recently proposed P/G simulation algorithm in [3]. The capability of the new algorithm is also significantly improved compared with the pure PCG as shown in Table II. The maximum P/G networks can be solved by the PCG method is 800*800 with 640k nodes and it takes 13102.46 seconds to solve the circuits, while the new algorithm can easily solve 1000*1000 P/G network in 687.50 seconds. For the circuit 800*800, the PCG-SOLVER takes 181.73 seconds (which is the total time minus the times of EQVCKT-CONSTRUCT and BACK-SOLVER shown in the table), which implies that PCG-SOLVER only contributes 41.75% time cost of our algorithm. As SPICE bails out on very small circuits, we expect more Speedup will be seen on large circuits given the super-linear time complexity of SPICE.

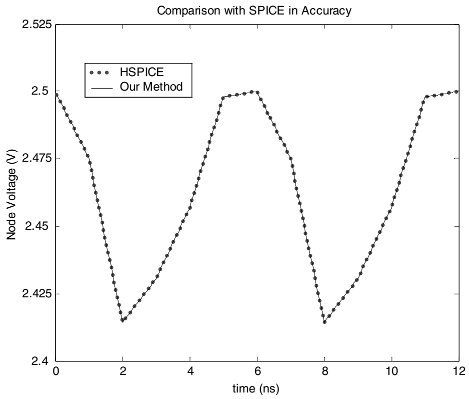

Fig.13. Accuracy comparison with SPICE

TableI Statistics on the original and reduced nets

P/G Grids	Node Seen By PCG		Reduction Ratio (X)
	Pure PCG	Our method	
50*50*10	2551	501	5
100*100*10	10101	1001	10
200*200*10	40201	2001	20
400*400*10	160401	4001	40
600*600*10	360601	6001	60
800*800*10	640801	8001	80
1000*1000*10	1001001	10001	100

We notice that is our new reduction method is a special form of Gaussian elimination process. But we exploit the regularities of P/G grids to follows the best order (such that no fill-ins are generated) for ladder circuits. While finding the best reduction order in Gaussian elimination process is a NP-hard problem in general [14].

5. Conclusion and Future Work

This paper proposes an efficient technique for analysis of RLC power distribution networks of deep sub-micron

[1] x*x*y p/g network has *x* strips, *y* trunks and each strap has *x+1* cell nodes.

TableII Comparison results on CPU times

P/G Grids	CPU time (sec.)					Speedup over	
	SPICE	Pure PCG	Our New Method			SPICE	Pure PCG
			EQVCKT-CONSTRUCT	BACK-SOLVER	Total time		
50*50*10	49.08	4.48	0.15	0.17	0.98	50.08	4.57
100*100*10	762.74	28.47	0.62	0.55	3.26	233.97	8.73
200*200*10	N/A	143.88	6.86	6.33	24.97	N/A	5.76
400*400*10	N/A	1391.55	34.87	30.40	115.82	N/A	12.04
600*600*10	N/A	3874.67	88.33	66.91	250.70	N/A	15.46
800*800*10	N/A	13102.46	135.68	117.85	435.26	N/A	30.10
1000*1000*10	N/A	N/A	211.13	190.37	687.50	N/A	N/A

VLSI systems in time-domain. At each time step, after integration approximation by Norton-form companion models, equivalent circuit models are built for many RLC chain circuits to reduce the P/G network complexities. Then precondition conjugate gradient (PCG) based iterative method is used to solve the reduced resistor-only networks. Experimental results demonstrate that the node reduction technique contributes at least one order of magnitude speedup over methods without node reduction and the resulting new algorithm is two order of magnitude faster than SPICE at almost no accuracy loss for many standard cell based P/G networks. The new algorithm takes 687.5 seconds to solve a P/G networks with more than 1 million nodes on a 750Mhz Sun workstation. Also the method can be easily extended to deal with tree-mesh structured P/G networks. This makes our algorithm a serious contender for attacking real industry P/G networks.

Reference

[1] G. Bai, S. Bobba and I. N. Hajj, "Simulation and optimization of the power distribution network in VLSI circuits", in *Proc. IEEE/ACM International Conf. on Computer-Aided Design*, pp. 481–486, 2000.

[2] T. Chen and C. C. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative method", *Proc. ACM/IEEE Design Automation Conf.*, pp. 559–562, June, 2001.

[3] Y. Cao, Y. Lee, T. Chen and C. C. Chen, "HiPRIME: hierarchical and passivity reserved interconnect macromodeling engine for RLKC power delivery", *Proc. ACM/IEEE Design Automation Conf.*, pp. 379–384, June, 2002.

[4] D. Stark and M. Horowitz, "Techniques for calculating currents and voltages in VLSI power supply networks," *IEEE Trans. on Computer-Aided Design*, vol. 9, no. 2, pp. 126-132, February 1990.

[5] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design", *Proc. 34th ACM/IEEE Design Automation Conf.*, pp. 683–643, 1997.

[6] A. Dharchoudhury, R. Panda, D. Blaauw, R. Vaidyanathan, B. Tutuianu and D. Bearden, "Design and analysis of power distribution networks in power PC microprocessors", *Proc. ACM/IEEE Design Automation Conf.*, pp. 738–743, June, 1998.

[7] Y.-M. Lee, C.-P. Chen, "Power grid transient simulation in linear time based on transmission-line-modeling alternating-direction-implicit method", *Proc. IEEE/ACM International Conf. on Computer-Aided Design.*, pp. 75–80, 2001.

[8] J.N. Kozhaya, S.R. Nassif, and F.N. Najm, *"A* multigrid-like technique for power grid analysis", *IEEE Trans. Computer-Aided Design*, vol. 21, no.10, pp. 1148-1160, Oct. 2002.

[9] M. Zhao, R. V. Panda, S. S. Sapatnekar and D. Blaauw, "Hierarchical analysis of power distribution networks", *IEEE Trans. Computer-Aided Design*, vol. 9, no. 2, pp. 159–168, Apr. 1990.

[10] H.-H. Su, K.H. Gala, and S.S. Sapatnekar, "Fast analysis and optimization of power/ground networks", *Proc. IEEE/ACM International Conf. on Computer-Aided Design.*, pp. 477–482, 2000.

[11] X.-D. Tan and C.-J. Shi, "Fast power-ground network optimization using equivalent circuit modeling," *Proc. 38th ACM/IEEE Design Automation Conf.*, pp. 550–554, 2001.

[12] C.-W. Yeh and Y.-S. Kang, "Cell-based layout techniques supporting gate-level voltage scaling for low power", *IEEE Trans. On Very Large Scale Integration Systems(VLSI)*, vol. 8, no.5 pp.629-633, October 2000.

[13] X.-H. Wu, X.-L. Hong, Y.-C. Cai, and et al., "Area minimization of power distribution network using efficient nonlinear programming techniques", *Proc. IEEE/ACM International Conf. on Computer-Aided Design*, pp. 153–157, 2001.

[14] G. Golub and C. Van Loan, *Matrix Computation*, 3rd Edition, Johns Hopkins University Press, 1996.

[15] J.A. Tierney, *Differential Equations,* Reading, Boston: Allyn and Bacon Incorporation, 1985.

[16] A. Odabasioglu, M. Celik, and L. T. Pilleggi, "PRIME: Passive reduction-order interconnect macromodeling algorithm," *IEEE Trans. On Computer-Aided Design*, vol. 17, no. 8, pp.645-654. 1998.