PAPER *Special Issue on Special Section on "VLSI Design and CAD Algorithm"*

Partial Random Walks For Transient Analysis of Large Power Distribution Networks*

Weikun GUO[†a)], **Sheldon X.-D. TAN**[†b)], **Zuying LUO**[††c)], ***and*** **Xianglong HANG**[††d)],

SUMMARY This paper proposes a new simulation algorithm for analyzing large power distribution networks, modeled as linear RLC circuits, based on a novel partial random walk concept. The random walk simulation method has been shown to be an efficient way to solve for voltages of small number of nodes in a large power distribution network [2], but the algorithm becomes expensive to solve for voltages of nodes that are more than a few with high accuracy. In this paper, we combine direct methods like LU factorization with the random walk concept to solve power distribution networks when voltage waveforms from a large number of nodes are required. We extend the random walk algorithm to deal with general RLC networks and show that Norton companion models for capacitors and self-inductors are more amenable for transient analysis by using random walks than Thevenin companion models. We also show that by nodal analysis (NA) formulation for all the voltage sources, LU-based direct simulations of subcircuits can be speeded up. Experimental results demonstrate that the resulting algorithm, called partial random walk (PRW), has significant advantages over the existing random walk method especially when the VDD/GND nodes are sparse and accuracy requirement is high.

key words: *Random Walks, On-Chip Power Distribution, Simulation, Power/Ground Networks*

1. Introduction

Signal integrity on the on-chip power distribution networks has become a limiting factor for designing high performance VLSI systems in today's deep submicron technology. The challenges for designing and verifying a reliable on-chip power deliver network lie in the increasing sizes of the network circuits that typically contain millions RLC components. Conventional circuit analyzers (such as SPICE) cannot meet such demanding simulation tasks and efficient simulations are highly required to reduce the increasing design productivity gap in deep submicron design regime.

Different methods have been proposed in the past to address this problem [3–10]. The existing approaches include frequency domain analysis method [3], the hierarchical and macromodeling method [6, 10], preconditioned iterative method [5, 11], the multi-grid method [7] and equivalent circuit modeling method [8]. Recently Qian *et al* proposed a new statistical method to solve the power/ground (P/G) networks [2]. The new algorithm exploits the fact that there exist some center-bumped VDD/GND pads evenly distributed in some mesh-structured P/G grid networks (due to advanced flip-chip packaging technologies such as IBM C4 package technique). A random walk process is applied to solve the node voltages in a statistical way [12]. The advantage of this method is that it can efficiently solve for a small number of node voltages in a large P/G network without solving the whole network. But the typical verification of a P/G circuit requires the analysis of the whole network or at least a portion of the network, not just voltages of a few nodes. This is special true for general transient analysis. The random walk method, however, is not very efficient for such tasks, as every node has to be simulated individually. Lowering accuracy may speedup the random walking process, but it may not be accepted for some applications requiring knowledge of accurate IR drops or voltage fluctuations. Also the proposed random walk algorithm can only work for RC networks using Thevenin companion models for grounded capacitors [2]. Recently the same authors extended the pure random walks to solve the circuit in a hierarchical way [13]. But all the nodes are still solved by random walks and the circuit can be solved is still limited to RC networks.

In this paper, we propose a new and general random walk algorithm that combines direct methods like LU factorization with the random walk concept. The idea is to partition a large P/G network into a number of smaller subcircuits and to solve each or a specific subcircuit in two steps. First we use the random walk process to solve for the boundary nodes of each subcircuit. Second we apply the LU method to solve for the rest of nodes inside the subcircuit. Such a process can be processed subcircuit by subcircuit or a hierarchical way if the solution of the whole network is required. We extend the random walk algorithm to deal with RLC networks and show that Norton companion models for capacitor and self-inductors are more general

Manuscript received June 1, 2004.

[†]Department of Electrical Engineering, University of California, Riverside, CA 92521, USA

[††]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

a) E-mail: wguo@ee.ucr.edu

b) E-mail: stan@ee.ucr.edu

c) E-mail: luozy@mail.tsinghua.edu.cn

d) E-mail: hxl-dcs@mail.tsinghua.edu.cn

*Some preliminary results of this paper will be presented at IEEE International Symposium on Circuits and Systems (ISCAS), Vancouver, Canada, May, 2004. [1]. This work is partially sponsored by Cadence Design Systems Inc. San Jose, CA

than Thevenin companion models for transient analysis by random walks. We also show by using nodal analysis formulation, LU-based simulation for solving subcircuits can be speeded up. Our experimental results show that the partial random walk method can be one order of magnitude faster than the pure random walk method specially when VDD/GND nodes are sparse and accuracy requirements are high.

This paper is organized as follows. In Section 2, we briefly review the random walk algorithm. In Section 3, we illustrate the new partial random walk concept. The experimental results are shown in Section 4. Finally we conclude the paper in Section 5.

2. Review of Random Walk Analysis Algorithm

The random-walk based approach to solving linear network exploits the fact that a node voltage is equivalent to the probability of a statistical winning process via random walks, which starts from the node and ends up with a home (a node with known voltage) [12]. Such a statistical winning process has the property that a probability for a node to reach a home is a linear function of the probability of its neighbor nodes. As a result, the winning process can be mapped to formulas that are similar to the Kirchoff's current and voltage laws in a statistical way especially when the circuits are formulated in nodal analysis (NA) form where each node voltage can be explicitly represented by its neighbor node voltages [2, 12].

The random walk process will become reasonably fast and accurate if the unknown node voltages can be sufficiently determined by visiting nearby voltage-known nodes (like VDD/ GND nodes) that are not far away. As a result, the algorithm is suitable for P/G networks that have many VDD/GND nodes or pads (voltage-known nodes) evenly distributed inside a chip. Otherwise, it will take a significant long walk before the walking process can stop at a voltage-known node. Also the random walk method can easily make the tradeoff between accuracy and runtime. But our experimental results show that the runtime are sensitive to the accuracy requirement especially when VDD nodes are sparse and voltage fluctuations are large. Therefore, pure random walk process is inefficient and unreliable for solving a large number of nodes with high accuracy, which is critical for the detailed signal integrity verification of large P/G networks, which was also observed in the recent paper by the same authors [13].

Also for the transient analysis, Thevenin companion models, which consist of an equivalent resistor in series with a voltage source, are used for grounded capacitors in [2, 13]. Thevenin models, however, can't be applied to the floating capacitors as we can't walk through a voltage source, this is especially true for inductors as they are floating elements in typical RLC P/G grids and interconnect circuits considering parasitic inductive effects. We will show that Norton companion models will be more amenable and general for transient analysis by random walks than the Thevenin models. More detail discussion on this issue will be presented in subsection 3.2.

3. Partial Random Walk Algorithm

In this section, we detail our partial random walk based simulation algorithm for solving large RLC power distribution linear networks.

3.1 Basic Idea

The basic idea of the partial random walk concept is to allow the random walk process to solve boundary part of a subcircuit and then solve the rest of the node voltages inside the subcircuit by traditional direct methods like the LU factorization. The new algorithm combines the efficiency of both the random walk method and the LU factorization method to speed up the whole simulation process.

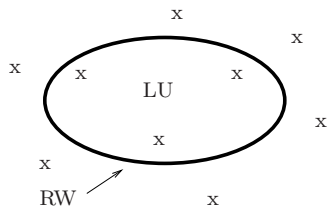

Fig. 1 The basic idea of partial random walks.

Specifically, we first partition a large P/G network into a number of small subcircuits such that LU method can solve them sufficiently fast. Then we apply the following two steps to solve each of the subcircuit: First we apply the random walk process to solve for all the boundary nodes of each subcircuit. Once the voltages of boundary nodes are known, we solve for voltages of the remaining nodes via LU method. This basic concept is illustrated in Fig. 1 where x represent the nodes with known voltages and the solid ellipse lines represent the boundary nodes. Since the number of boundary nodes (which grows linearly with the size of a subcircuit) is typically smaller than the number of nodes inside each subcircuit (which grows quadratically with the size of a mesh-structured subcircuit), the runtime cost of solving those boundary nodes by random walk processes will be reduced significantly while the rest of nodes can be solved by the LU method more efficiently.

We notice that how partitioning is performed or

subcircuit is defined depends on the specific verification application requirements. In our work, we employ two ways to do the partitioning if all the node voltages are to be computed. If the geometrical or layout information is known for each node in the circuit, the partitioning can be done based on the geometrical information of subcircuits as shown in Fig. 2. In this way, each subcircuit will have the similar size and the numbers of boundary nodes are minimized. The second approach is based on clustering and user-specified subcircuit size when no layout information is known. A connectivity-based clustering is performed to construct each subcircuit one by one until all the nodes are visited. In both methods, we want to make sure that the size of every subcircuit will be bounded and at the same time, the number of boundary nodes is minimized.

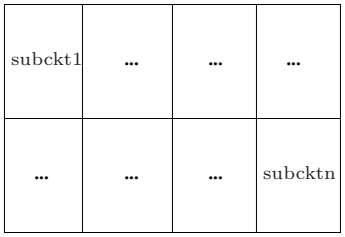

Fig. 2 Geometrically based partitioning.

3.2 Random Walks for RLC Networks

In [2], P/G networks modeled as RC circuits are analyzed. For transient analysis, Thevenin companion models with backward Euler integration are used for grounded capacitors as shown in Fig. 3. When the random walk process walks through the resistor $R_{eq}(t)$, it meets a home (home is referred to as a node with known voltage like voltage sources). But for a floating capacitor and a floating inductor, we can't directly use the Thevenin companion model as we can't walk through a voltage source directly in the original random walk process [2] as a voltage source is treated as a home.

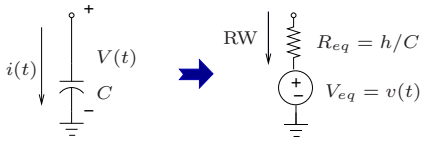

Fig. 3 A grounded capacitor and its Thevenin companion model in random walks.

There are two solutions to this problem. One is that we use *conditional money* concept (money here refers to the reward one obtains when one visits a node in the random walk process [2]) for the Thevenin companion model. Specifically, we do not collect the money at node x unless the branch we select to walk through is R_{eq} and the voltage source branch (we walk through the two branches in one step). The money is the equivalent voltage in the Thevenin companion models of capacitors or inductors. We call this approach as *conditional money random walk* for Thevenin companion models as the money is collected only when we walk through the voltage source branch. But such method requires special treatment for the equivalent circuits during the random walk process.

The second solution to this problem is more elegant. Instead of using Thevenin companion models, we use Norton companion models for both capacitors and inductors. In this way, normal random walk can be applied directly without any adjustment. Specifically, using the backward Euler integration and assuming the time step is h, a floating capacitor and its Norton companion model are shown Fig. 4 in a RC network. We then have the following equation to compute the probabilities of walking through each neighbor branch i and the money we need to collect at node x.

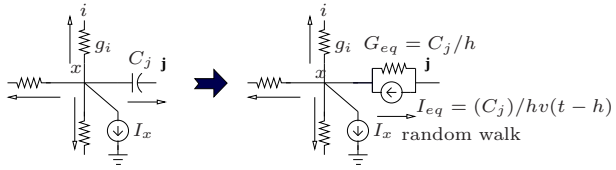

Fig. 4 A floating capacitor and its Norton companion model in random walks.

$$\sum g_i(v_x(t) - v_i(t)) + \frac{C_j}{h}(v_x(t) - v_j(t)) = -I_x + \frac{C_j}{h}(v_x(t-h) - v_j(t-h)). \quad (1)$$

Then we have,

$$v_x(t) = \sum \frac{g_i}{\sum g_i + \frac{C_j}{h}} v_i(t) + \frac{\frac{C_j}{h}}{\sum g_i + \frac{C_j}{h}} v_j(t) - \frac{I_x}{\sum g_i + \frac{C_j}{h}} + \frac{\frac{C_j}{h}(v_x(t-h) - v_j(t-h))}{\sum g_i + \frac{C_j}{h}} \quad (2)$$

The coefficient for each v_i or v_j is the probability that we will walk through branch i or j from node x. The last two items in the right-hand side of Eq.(2) are the money we collect at node x.

Note that we do not need to walk through an equivalent current source branch. The contribution of the equivalent current source is in the form of the money, which is $\frac{\frac{C_j}{h}(v_x(t-h) - v_j(t-h))}{\sum g_i + \frac{C_j}{h}}$ in the right-hand side of Eq.(2)). If we treat the $\frac{\frac{C_j}{h}}{\sum g_i + \frac{C_j}{h}}$ as the probability for $v_x(t-h) - v_j(t-h)$, then $v_x(t-h) - v_j(t-h)$ is exactly the equivalent voltage in the Thevenin companion model of the capacitor using backward Euler

integration in Fig. 3. So the Norton companion model is equivalent to the conditional money random walk for Thevenin companion model for capacitors.

For a floating inductor, we can also apply the conditional money random walk based on Thevenin companion models or the general random walks based on Norton companion model. Specifically, using the backward Euler integration, a floating inductor and its Norton companion model are shown Fig. 5 in a RL network. We then have the following equation to compute the probabilities of walking through each neighbor branch i and the money we need to collect at node x.

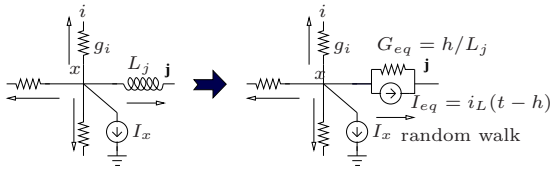

Fig. 5 A floating inductor and its Norton companion model in random walks.

$$\sum g_i(v_x(t)-v_i(t))+\frac{h}{L_j}(v_x(t)-v_j(t)) = -I_x-i_L(t-h). \quad (3)$$

Then we have,

$$v_x(t) = \sum \frac{g_i}{\sum g_i + \frac{h}{L_j}} v_i(t) + \frac{\frac{h}{L_j}}{\sum g_i + \frac{C_j}{h}} v_j(t) - \frac{I_x}{\sum g_i + \frac{h}{L_j}} - \frac{i_L(t-h)}{\sum g_i + \frac{h}{L_j}}. \quad (4)$$

Still the coefficient for each v_i or v_j is the probability that we will walk through branch i or j from node x. The last two items in the right-hand side of Eq.(4) are the money we collect at node x. The money $\frac{i_L(t-h)}{\sum g_i+\frac{h}{L_j}}$ comes from the equivalent current source. Similar to the capacitors, if we rewrite it as $\frac{\frac{h}{L_j} i_L(t-h)}{(\sum g_i+\frac{h}{L_j})\frac{h}{L_j}}$ and treat $\frac{\frac{h}{L_j}}{(\sum g_i+\frac{h}{L_j})}$ as the probability for $\frac{i_L(t-h)}{\frac{h}{L_j}}$, then $\frac{i_L(t-h)}{\frac{h}{L_j}}$ is exactly the equivalent voltage source in the Thevenin companion model of an inductor using backward Euler integration [14]. So the Norton companion model is equivalent to the conditional money random walk for Thevenin companion model for inductors.

Hence, it can be seen that Norton companion models are more general than the Thevenin companion models for the random walk process as both floating capacitors and inductors can be directly handled without changing the random walk process.

3.3 Nodal Analysis Formulation For Subcircuit Simulation

Once the boundary node voltages are known, the LU method is applied to solve the node voltage inside the subcircuit. But instead of stamping the voltage sources into the circuit matrix, we use equivalent current sources to avoid computing the current variables for each voltage sources in MNA formulation. This is equivalent to the nodal analysis (NA) formulation of the subcircuit. Experimental results show that NA formulation can speedup the LU simulation of the subcircuit.

Fig. 6 shows how a voltage source V_b can be converted to a current source. The equivalent current sources are also used for VDD/GND nodes inside the subcircuits to reduce the MNA matrix size and avoid computing the current variables for each voltage sources.

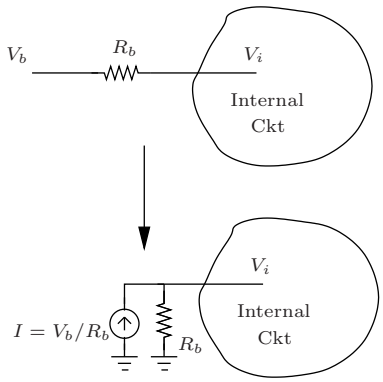

Fig. 6 Equivalent current sources for voltage sources of the boundary nodes and internal VDD/GND nodes.

From Fig. 6, we can see that if the number of VDD nodes is small compared to the total number of nodes, the CPU time improvement will not be significant. But NA formulation is still important or even necessary due to following reasons:

First, for the subcircuits, NA formulation can allow faster linear solving method like preconditioned conjugate gradient method [5] to solve the subcircuit as the circuit matrix under NA is symmetric positive definite.

Second, it was shown recently in [11] that MNA formulation for voltage sources for general RLC networks may lead to a larger condition number of the resulting circuit matrix even at very small time step. The reason is that the numerical values of the stamps from VDD nodes are constants in the circuit matrix and they do not change with different time step h values as L and C elements do (the equivalent conductor is h/L for inductor L and C/h for capacitor C using backward Euler integration). Large condition number may lead to more iterations for iterative approaches due to tighter error criteria and larger errors for direct methods like LU decomposition.

3.4 CPU Time Complexity Analysis

The CPU time for partial random walks consists of

two parts: (1) the random walk time for all the boundary nodes and (2) the time for solving all subcircuits.

If we assume that each subcircuit has the same size and the linear circuit has homogeneous structure throughout the whole circuit, the partial random walks will take linear time to solve a circuit. To see this, we know that the first part of the CPU time from random walks is linear as each random walk for a node will take approximately the same time on average. For the second part of the CPU time, since each subcircuit has the same size (and similar structure), solving each subcircuit by the LU method will take the same time. The CPU time is a linear function of the number of subcircuits, which in turn is linearly determined by the size of the whole circuit.

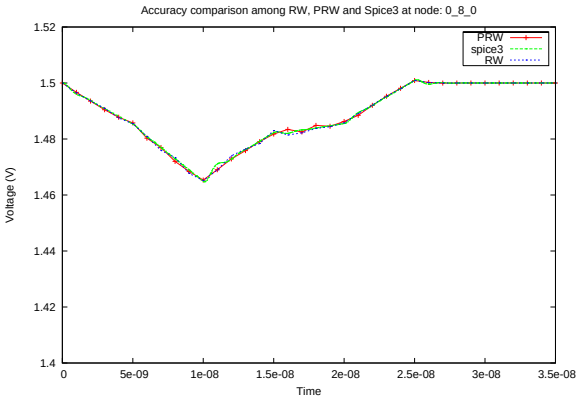

Fig. 8 Simulation waveforms by RW and PRW and Spice3f4 for a boundary node

4. Experimental Results

The proposed algorithm has been implemented in C++. We use SuperLU [15] to solve the linear equations from a subcircuit. All the experimental results are obtained on a Linux workstation with dual 1.6GHz AMD Athlon MP 2000+ CPUs and 2GB memory.

First, we compare the waveforms from partial random walk *PRW* and random walk algorithms *RW* with that of Spice3f4 on a RLC circuit with a 3721 nodes. The waveforms for an internal nodes and the waveforms for a boundary node are shown in Fig. 7 and Fig. 8. For both RW and PRW, we set delta to 0.01v, which is the absolute error margin for accuracy constraints. For instance, delta = 0.01v means that the error between the estimated one and newly measured one is less than $+/-0.01$ volt with very high probability (99%) [2]. The percentage of VDD nodes is also set to 20% (meaning that 20% nodes are VDD nodes). It can be seen that the waveforms from both PRW and RW are very accurate compared with Spice3f4.

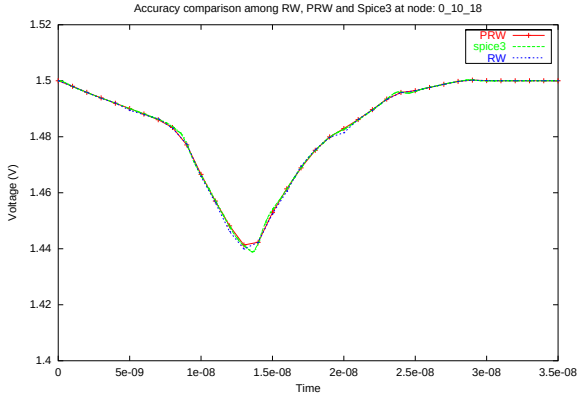

Fig. 7 Simulation waveforms by RW and PRW and Spice3f4 for an internal node

Second, we apply the proposed algorithm to solve a RLC P/G grid circuits with 63001 nodes and various ratios of VDD nodes. All the tap currents are modeled as PWL current sources attached to most of non-VDD nodes in the circuits. The performance of the proposed algorithm is also evaluated under different accuracy requirements. These performance results are compared with those of the pure random walk algorithm and Spice3f4.

Table 1 summarizes the CPU times for solving the whole circuit (all the nodes) for the three algorithms under different VDD percentages and accuracies in terms of deltas. For partial random algorithm, the circuit is partitioned into 25 subcircuits based on the geometrical information of the nodes and each subcircuit has about 2500 nodes. Each subcircuit is then solved sequentially by the partial random walk method.

From Table 1, we can see that PRW algorithm is faster than other two methods for all the cases. The speedups over RW become significant (about 10x) for high accuracy requirements (delta = 0.001v). But if the accuracy requirements are low as the case for delta = 0.1, both RW and PRW are much faster than Spice3f4.

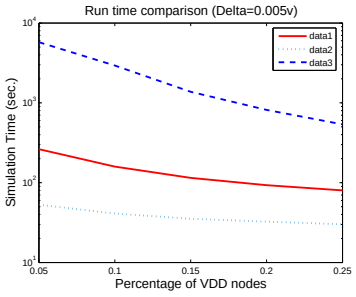

Fig. 9 The CPU time comparison of PRW, RW and Spice3f4 on simulation of the RLC circuit with delta = 0.005v.

Also as the percentage of VDD nodes goes up, the CPU times of both RW and PRW go down. This is expected as random walks have a better chance

Table 1 Runtime comparison of PRW, RW and Spice3f4 for a whole circuit simulation (sec.)

Precision		0.001v		0.005v		0.01v		0.05v		0.1v	
VDD	Spice3f4	PRW	RW	PRW	RW	PRW	RW	PRW	RW	PRW	RW
1%	9432	748	6122	60	310	41	157	37	127	37	127
5%	5751	576	5082	53	263	38	133	36	114	35	114
10%	2993	299	2587	41	159	35	106	34	102	34	102
15%	1367	182	1446	35	115	32	92	32	92	32	93
20%	815	113	872	32	93	31	84	31	84	31	84
25%	538	82	562	30	80	29	76	29	76	29	76

to hit VDD nodes (find home nodes) than before, which speeds up the walking process. This trend is clearly shown in Fig. 9, which gives the CPU times of the PRW, the RW methods and Spice3f4 under various VDD/GND node percentages in term of the total number of nodes under constant accuracy requirement (delta =0.005v).

Table 2 gives the CPU times of the three algorithms on RLC circuits with different sizes. *Speedup(X)* means the speedup of PRW over method X. The VDD percentages and deltas are also given in the table. The speedup of PRW over RW is about 6 on average. But if high accuracy is required, we expect the speedup will goes up as shown in Table 1.

Fig. 10 shows the CPU times versus circuit sizes for the three methods. We note that both RW and PRW are linear in terms of circuit sizes. This is expected as both RW and PRW are linear algorithms as analyzed in Subsection 3.4. Fig. 11 shows how the CPU times of the

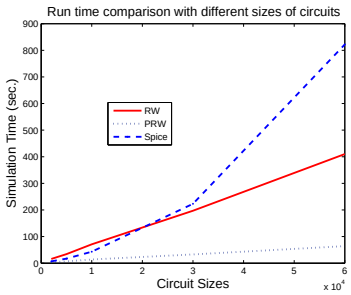

Fig. 10 The CPU time comparison with different circuit sizes.

PRW and RW methods change with the accuracy requirements for 1% and 25% VDD node percentage configurations. The accuracy requirement seemingly does not have a big impact on the CPU times for both RW and PRW until the accuracy requirement becomes high (delte $< 0.01v$).

But if we increase the voltage fluctuation magnitudes (by increasing the current source values by about 5x), the CPU time of both RW and PRW will become more sensitive to the accuracy requirements as shown in Fig. 12. The reason is that with larger voltage fluctuations, the random walk process will take longer (more walks) to converge as the voltage difference between each time step increases, which in turn will increase the

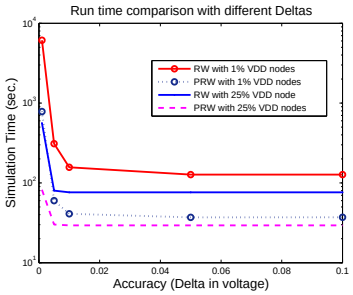

Fig. 11 The CPU times of PRW and RW run on one circuit with different VDD nodes.

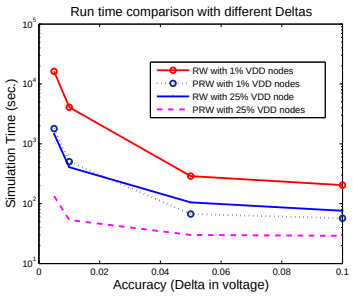

Fig. 12 Runtime comparison of PRW and RW run on one circuit with different VDD nodes with increased voltage fluctuations.

variations of the estimated voltages in each time step. As a result, a larger number of random walks is needed to meet the required accuracy as accuracy requirement is expressed in terms of absolute voltage values. So both RW and PRW are sensitive to the magnitudes of dynamic voltage fluctuations on the linear RLC networks. But the partial random walk algorithm is less susceptible to the magnitudes of voltage waveforms as only parts of nodes are solved by random walks.

Fig. 13 shows the impacts of the equivalent modeling (using NA formulation) on the CPU time for the whole circuit simulation with accuracy = 0.005 for different VDD node percentages. As VDD node percentage increases, more voltage sources become current sources, more CPU time can be saved by avoiding solving for currents of the voltage source. This becomes more evident when more VDD nodes are present. Notice that since circuit matrices based on the NA formulation are symmetric positive definite, Cholesky decomposition or iterative approaches can be used to further

Table 2 The CPU time of PRW, RW and Spice3f4 for different circuits (sec.)

Circuit size	Spice3f4	RW	PRW	Speedup(Spice3f4)	Speedup(RW)
2000	7	16	3	2.3	5.3
5000	17	34	7	2.4	4.9
10000	43	71	14	3.1	5.1
30000	223	197	33	6.8	6.0
60000	822	410	64	12.8	6.4

*delta=0.01, VDD nodes=20%, time step=1ns, time range:0-30ns

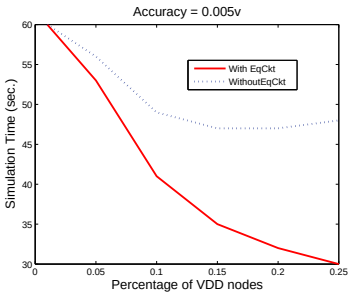

Fig. 13 Comparison with and without equivalent current source modeling.

speedup the direct simulation [16].

5. Conclusions

In this paper, we have proposed a new circuit simulation algorithm, which combines the recent proposed statistical based random walk concept with the LU factorization method to take advantage of both methods to speedup the simulation of large on-chip power distribution networks modeled as RLC circuits. We have extended the random walk method to deal with general RLC networks by using more general Norton companion models for numerical time-domain integration. We also show that nodal formulation of the subcircuit can speed up the simulation of subcircuits by the LU method. Our experimental results show the partial random walk algorithm can be significantly faster than the pure random walk algorithm when VDD/GND nodes are sparse and accuracy requirements are high. Also the CPU times of both methods are sensitive to the magnitude of voltage waveforms. But the partial random walk method is less susceptible to the voltage waveforms thus is more robust than the pure random walk method.

Acknowledgement

The authors would like to thank Haifeng Qian from the University of Minnesota for the discussion of the random walk algorithm.

References

[1] W. Guo, S.X.D. Tan, Z. Luo, and X. Hong, "Partial random walk for large linear network analysis," Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), pp.V173–176, 2004.

[2] H.F. Qian, S.R. Nassif, and S.S. Sapatnekar, "Random walks in a supply network," Proc. Design Automation Conf. (DAC), pp.93–98, 2003.

[3] G. Bai, S. Bobba, and I.N. Hajj, "Simulation and optimization of the power distribution network in VLSI circuits," Proc. Int. Conf. on Computer Aided Design (ICCAD), pp.481–486, 2000.

[4] H.H. Chen and D.D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," Proc. Design Automation Conf. (DAC), pp.638–643, 1997.

[5] T. Chen and C.C. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative method," Proc. Design Automation Conf. (DAC), pp.559–562, 2001.

[6] Y. Cao, Y. Lee, T. Chen, and C.C. Chen, "HiPRIME: hierarchical and passivity reserved interconnect macromodeling engine for RLKC power delivery," Proc. Design Automation Conf. (DAC), pp.379–384, 2002.

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

[8] Z. Pan, Y. Cai, S.X.D. Tan, and X.H. Z. Luo, "Transient analysis of on-chip power distribution networks using equivalent circuit modeling," Proc. Int. Symposium. on Quality Electronic Design (ISQED), 2004. to appear.

[9] H. Su, K.H. Gala, and S. Sapatnekar, "Fast analysis and optimization of power/ground networks," Proc. Int. Conf. on Computer Aided Design (ICCAD), pp.447–480, Nov. 2000.

[10] M. Zhao, R.V. Panda, S.S. Sapatnekar, and D. Blaauw, "Hierarchical analysis of power distribution networks," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol.21, no.2, pp.159–168, Feb. 2002.

[11] Z. Li and C.J.R. Shi, "A coupled iterative method for efficient time-domain simulation of nonlinear circuits with power/ground networks," Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), pp.V165–168, 2004.

[12] P.G. Doyle and J.L. Snell, "Random Walks and Electric Networks." arXiv:math.PR/0001057.

[13] H.F. Qian and S.S. Sapatnekar, "Hierarchical random-walk algorithms for power grid analysis," Proc. Asia South Pacific Design Automation Conf. (ASPDAC), 2004.

[14] J. Vlach and K. Singhal, Computer Methods for Circuit Analysis and Design, Van Nostrand Reinhold, New York, NY, 1995.

[15] "http://crd.lbl.gov/ xiaoye/superlu/."

[16] G.H. Golub and C.F.V. Loan, Matrix Computations, 3 ed., The Johns Hopkins University Press, Baltimore, MD, 1989.