Partial Random Walk For Large Linear Network Analysis

Weikun Guo, Sheldon X.-D. Tan, †Zuying Luo, †Xianlong Hong
Department of Electrical Engineering, University of California, Riverside, CA 92521
†Department of Computer Science and Technology, Tsinghua University, Beijing 100084,China

ABSTRACT

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 a small number of nodes in a lager power distribution network [6], but the algorithm becomes expensive to solve for nodes that are more than a few. In this paper, we combine direct methods like LU factorization with the random walk concept to solve power distribution networks when a significant number of node waveforms is required. We also apply an equivalent circuit modeling method to speed up the direct simulation of subcircuits. Experimental results show that the resulting algorithm, called partial random walk (PRW), has significant advantages over the pure random walk method especially when the VDD/GND nodes are sparse and accuracy requirement is high.

1. INTRODUCTION

Signal integrity on the on-chip power distribution networks has become a limiting factor for designing high performance VLSI systems in today 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[1][2][3][4][5][7][8]. These existing approaches include frequency domain analysis method[1], the hierarchical method [7], the multi-grid method [5]. Recently Qian *et al* proposes a new statistical method to solve the power/ground (P/G) networks. The new algorithm exploits the fact that there exist some VDD/GND nodes evenly distributed for some mesh-structured P/G grid networks (due to advanced packaging technologies like IBM C4 package). A random walk process is applied to solve the node voltages in a statistical way. 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 a few nodes. 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.

In this paper, we propose a new random walk algorithm that combines direct solution methods like LU factorization method 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 LU method to solve for the rest of nodes inside the subcircuit. Such process can be processed subcircuit by subcircuit or a hierarchical way if the solution of the whole network is required. Our experimental results show that the partial random walk can be one order of magnitude faster than the pure random walks 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 work concept. The experimental results are shown in Section 4. Finally we conclude the paper in Section 5.

2. REVIEW OF RANDOM WALK ALGORITHM

The random-walk based approach to solving linear network exploits the fact that Kirchoff's current and voltage laws can be mapped to a random walking process such that KVL and KCL equations are equivalent to statistic formulas describing the random walking process [6]. Such a statistical process will become reasonable fast and accurate if the unknown node voltages can be sufficiently determined by visiting nearly by 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 significant long walks before the walking process can stop at the 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 can be significantly slowed down even with a very small increase in the accuracy requirement. Therefore, pure random walk process is inefficient for solving a large number of nodes with high accuracy, which is critical for the detailed signal integrity verification of large P/G networks.

3. PARTIAL RANDOM WALK

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 nodes in the subcircuit by traditional direct methods like LU factorizations. The new algorithm combines the efficiency of both the random walk and LU factorization method to speed up the whole simulation process.

Specifically, we first partition a large P/G network into a number of small subcircuits such that LU method can solve them sufficiently fast. We notice that how a partitioning is performed or subcircuit is defined depends on the specific verification application. 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. 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 efficiently.

Notice that if a number of subcircuits are to be solved, the order to solve those subcircuits should be arranged such that boundary node voltages, which are computed by random walk processes, should be reused as much as possible.

3.2 Equivalent Circuits For Voltage Sources

To speedup the simulation of subcircuits by LU method, we need to reduce the matrix sizes of the subcircuits by transforming the known boundary voltages to equivalent current sources. Specifically, let's have system-equation set Av = b. Suppose that we partition the circuit into two circuits, one is small subcircuit *I* and another is rest of the circuit *R*. In between, there are some boundary nodes connecting the two circuits. As a result, the circuit equation set can be rewritten in the following form:

$$\begin{bmatrix} A^{II} & A^{IB} & 0 \\ A^{BI} & A^{BB} & A^{BR} \\ 0 & A^{RB} & A^{RR} \end{bmatrix} \begin{bmatrix} v^I \\ v^B \\ v^R \end{bmatrix} = \begin{bmatrix} B^I \\ B^B \\ B^R \end{bmatrix} \quad (1)$$

where A^{II} is the internal matrix associated with internal variable vector v^I, A^{IB} and A^{BI} are the connection matrices between internal nodes v^I and boundary nodes v^B. Suppose that we obtain the node voltages of boundary node v^B by random walk processes. Then we have the following equations according to (1).

$$A^{II} v^I + A^{IB} v^B = B^I \quad (2)$$

then

$$A^{II} v^I = B^I - A^{IB} v^B \quad (3)$$

As a result, we can solve Eq (3) to obtain all the internal nodes voltages. Notice that $A^{IB}v^B$ is the vector of independent current sources computed from the known voltages of the boundary node vector v^B, which can be computed easily as shown in Fig. 3. The equivalent current sources are also used for VDD/GND nodes inside the subcircuits to reduce the MNA matrix size. Experimental results show using the equivalent current sources can significantly reduce the simulation time of subcircuits by LU method.

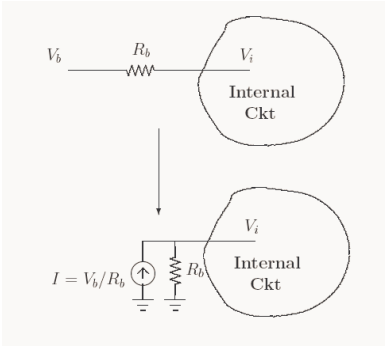

Figure 3. Equivalent current sources for voltage sources of the boundary nodes and internal VDD/GND nodes.

3.3 Extension to RLC network

In [6], P/G networks modeled as RC circuits are analyzed. In this subsection, we discuss how to extend this random walk to deal with RCL circuits. Here we consider self-inductor L.

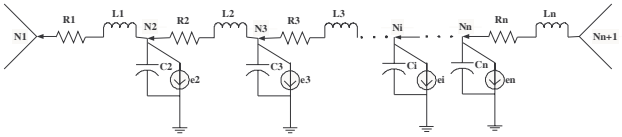

Figure 4. RLC Segments in P/G Circuits.

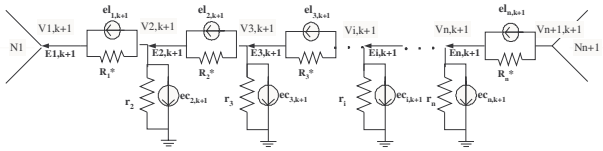

Figure 5. Simplified Discretized RLC Segments

For one RLC segment shown in Fig.4, if we apply trapezoidal integration with time step *h* and Norton equivalent circuit, we can transform the circuit in Fig.4 to the one in Fig.5. Specifically, during the transformation, a shunt capacitance branch *i* is transformed into an equivalent current and an equivalent resistor.

$$r_i = \frac{h}{2C_i}, \quad Ic_{i,k+1} = -\left(\frac{2C_i}{h} V_{i,k} + I_{i,k}\right), \quad (4)$$

where *k* is the time step index. Combine the current caused by devices and the equivalent current due to capacitor *i,* we obtain

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

Meanwhile, the floating R and L branches can be transformed into an equivalent current and an equivalent resistor connected in parallel.

$$R_i^* = 2L_i/h + R_i, \quad el_{i,k+1} = \left(\frac{h}{2L_i} V_{L,i,k} + I_{L,i,k}\right) \cdot \frac{2L_i/h}{2L_i/h + R_i}, \quad (6)$$

Let's define

$$g_{i-1,j,H} = \frac{1}{R^*_{i-1,j,H}}; \quad g_{i,j,H} = \frac{1}{R^*_{i,j,H}};$$

$$g_{i,j-1,V} = \frac{1}{R^*_{i,j-1,V}}; \quad g_{i,j,V} = \frac{1}{R^*_{i,j,V}};$$

$$g_{i,j} = g_{i-1,j,H} + g_{i,j,H} + g_{i,j-1,V} + g_{i,j,V} + \frac{1}{r_{i,j}},$$

where subscripts *H* and *V* mean horizontal and vertical direction. According to Fig.5, the KCL equation at node (*i,j*) can be formatted as the following

$$V_{i,j,k+1} = \frac{g_{i-1,j,H}}{g_{i,j}} V_{i-1,j,k+1} + \frac{g_{i,j,H}}{g_{i,j}} V_{i+1,j,k+1} + \frac{g_{i,j-1,V}}{g_{i,j}} V_{i,j-1,k+1} + \frac{g_{i,j,V}}{g_{i,j}} V_{i,j+1,k+1} \quad (7)$$

$$- \frac{1}{g_{i,j}} \left(Is_{i,j,k+1} - \left(\frac{2C_{i,j}}{h} V_{i,j,k} + I_{i,j,k}\right) + el_{i-1,j,H} - el_{i,j,H} - el_{i,j-1,V} + el_{i,j,V} \right)$$

Let's further define

$$I_{i,j,k+1} = \left(\frac{2C_{i,j}}{h} V_{i,j,k} + I_{i,j,k} \right) - el_{i-1,j,H} + el_{i,j,H} + el_{i,j-1,V} - el_{i,j,V}$$

Finally, we have

$$V_{i,j,k+1} = \frac{g_{i-1,j,H}}{g_{i,j}} V_{i-1,j,k+1} + \frac{g_{i,j,H}}{g_{i,j}} V_{i+1,j,k+1} + \frac{g_{i,j-1,V}}{g_{i,j}} V_{i,j-1,k+1} + \frac{g_{i,j,V}}{g_{i,j}} V_{i,j+1,k+1} + \frac{1}{g_{i,j}} I_{i,j,k+1} - \frac{1}{g_{i,j}} Is_{i,j,k+1} \quad (8)$$

Eq.(8) gives the probabilities of a random walking step from node (i,j) to its neighbor nodes (the coefficient of V_x is the probability to walking into each neighborhood node x) through resistor-inductance branches. Note that we will walk through the floating resistor and inductor, which are connected in series, in just one step.

4 EXPERIMENTAL RESULTS

The proposed algorithm has been implemented in C++. For the LU solver, we use SuperLU [9] to solve the linear equations from a subcircuit. The proposed algorithm is applied to analyze a RC P/G grid circuits with 63001 nodes and various ratios of VDD nodes. 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 SPICE.

Table 1 gives the CPU run times for the whole circuit, which is partitioned into 25 subcircuits, under different VDD percentages and accuracy in terms of delta, which is the absolute error margin for accuracy constraints. For instance, delta = 0.01v means that the error between the estimated one and real one is less than ± 0.01 volt with very high probability (99%)[6]. To obtain more accurate results, we also compare the two methods for simulating one particular subcircuit (50x50) of the circuit and the results are shown in Table 2. All the computations are carried out on a Linux workstation with dual 1.7GHz AMD processors and 2 GB memories.

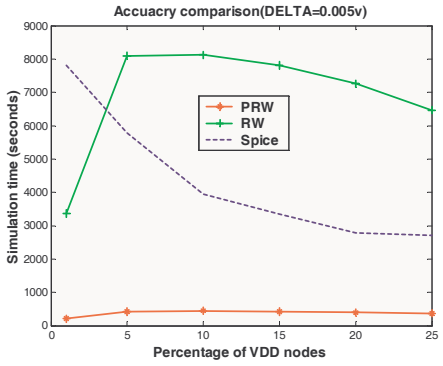

Figure 5. Runtime comparison of PRW, RW and SPICE on simulation of the whole circuit with delta = 0.005v

In each table, the first row shows the accuracy requirements in terms of different deltas. PRW and RW designate partial random walk and the pure random walk algorithms respectively. From Table 1 and Table 2, we can see that PRW algorithm is faster than other two methods for all the cases. The speedups over RW become significant (more than 10x) for high accuracy requirements (delta < 0.01v).

Figure 5 compares the runtimes of the PRW, the RW methods and SPICE under various VDD/GND node percentages in term of total number of nodes under constant accuracy requirement (delta =0.005v). For the random walk algorithm, theoretically if we decrease the number of VDD nodes, the random walk process will run slowly as it takes longer paths on average to find a home (node with known voltage). However, this is not always true, as the VDD nodes become very sparse, some node's voltages are small (we use 1.5volt as VDD value), as a result, the variations of the estimated voltages from true voltages become smaller, which in turn results in less number of random walks needed to meet the required accuracy as accuracy requirement is expressed in terms of absolute voltage values. So we may end up with small CPU runtimes as shown in Figure.5. But as the number of VDD node further increases, it eventually will help to save CPU time as shown in Figure. 5. For PRW method, since its random portion of run time is reduced, it becomes less sensitive to the changes of the VDD node percentages.

Figure 6 compares the runtime of the PRW and RW methods on simulating one particular subcircuit with 1% and 25% VDD node percentages respectively. The PRW method is faster than the RW method for all the cases. The speedup becomes more significant when the accuracy reaches 0.01 and beyond.

Figure 7 shows the impacts of the equivalent modeling on the CPU time for the whole circuit simulation with accuracy = 0.005 for different VDD node percentages. If no equivalent circuits are used, size of the circuit matrix will increase with the number of VDD nodes, as one VDD node introduces one extra node into the circuit matrix in MNA (modified nodal analysis) formulation, while the equivalent current of a voltage source reduces one node in circuit matrix, so there are two node difference in circuit matrix size for each VDD node added into a subcircuit. Since all the boundary nodes are voltage-known nodes, the MNA matrix size can be significantly reduced after we use equivalent modeling circuits for these boundary and VDD/GND nodes inside the subcircuits. As a result, the simulation time of LU method can be significantly reduced. On the contrary, as the matrix size increases, so does the CPU as shown in Figure 7 for simulation without using the equivalent circuits.

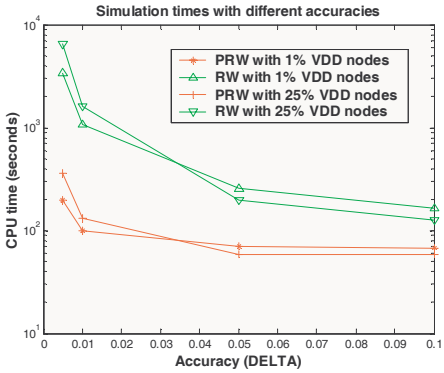

Figure 6. Runtime comparison of PRW and RW run on one subcircuit with different VDD nodes

5. CONCLUSION

In this paper, we have proposed a new circuit simulation algorithm, which combines the recent proposed statistical based random walk concept with LU factorization method to take advantage of both methods to speedup the simulation of large power distribution networks modeled RLC circuits. We also applied equivalent current circuits to speed up the simulation of subcircuits by LU method and extended the random talk method to deal RLC networks. Our experimental results show the significant speedup can be achieved over the pure random walk algorithm when VDD/GND nodes are spare and accuracy requirements are high.

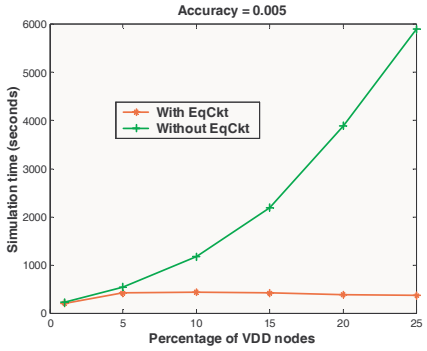

Figure 7. Comparison with and without equivalent current source modeling

Acknowledge

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

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] 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. 638–643, 1997.

[5] 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.

[6] H.-F. Qian, S.R. Nassif. S.S. Sapatnekar, "Random Walks in a Supply Network," *Proc. Design Automation Conference*, pp. 93-98, 2003.

[7] 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.

[8] 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.

[9] SuperLU, http://crd.lbl.gov/~xiaoye/SuperLU/

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

Precision		0.005v		0.01v		0.05v		0.1v	
Vdd	SPICE	PRW	RW	PRW	RW	PRW	RW	PRW	RW
1%	7790	197	3363	99	1078	70	254	68	163
5%	5767	409	8103	147	2700	68	255	66	157
10%	3928	433	8129	152	2255	64	265	63	151
15%	3321	410	7798	144	2086	63	252	61	142
20%	2774	382	7248	137	1775	60	204	58	135
25%	2693	359	6457	132	1623	59	196	58	125

Table 2. Runtime comparison of PRW and RW for one subcircuit simulation (sec.)

Precision	0.001v		0.005v		0.01v		0.05v		0.1v	
Vdd	PRW	RW	PRW	RW	PRW	RW	PRW	RW	PRW	RW
1%	112	3474	7	145	3.75	37.43	2.85	7.4	2.82	6.85
5%	378	7904	18	327	6.06	82	2.77	7.88	2.73	6.63
10%	481	8611	21	338	7.02	84	2.70	7.68	2.62	6.25
15%	451	8037	20	318	6.65	79	2.6	7.39	2.53	5.93
20%	503	7196	22	286	7.16	74	2.54	7.12	2.44	5.73
25%	459	6689	20	268	6.70	66	2.42	6.71	2.34	5.3