Circuit Level Alternating-Direction-Implicit Approach to Transient Analysis of Power Distribution Networks

Weikun Guo and Sheldon X.-D. Tan

Department of Electrical Engineering
University of California, Riverside, CA 92521, USA
stan@ee.ucr.edu

Abstract— *This paper presents an efficient method for transient analysis of power distribution networks. We employ an iterative approach – Alternating-Direction-Implicit (ADI) method– to solving the circuit matrices of power distribution networks. In contrast to the existing finite-difference based ADI approach to power/ground grid analysis [7], where transmission-line was used to model the mesh-structured P/G grids (TLM-ADI), we apply ADI scheme to directly solve P/G circuits consisting of lumped RLC components discretized at each time step. The new algorithm does not have the stability issue associated with finite-difference based methods like TLM-ADI method and is able to handle mesh-like structured P/G grids which can be found in many ASIC applications and is more versatile than the TLM-ADI approach. New algorithm also preserves the linear time complexity. Experimental results on a number of large P/G grid circuits and comparison with Spice3f4 are presented.*

I. Introduction

Reliable on-chip power delivery has been considered one of grant challenges in the nano-regime and Gigahertz VLIS chip designs [8]. Signal integrity in the power/ground (P/G) bus is emerging as a limiting factor as P/ G grids experience the largest current flows in a chip and are more susceptible to current-induced reliability and functional failures. The quality and reliability issues of P/G grids come from severals reasons such as excessive IR drops, Ldi/dt noise, electromigration and resonance. Efficient transient analysis of P/G grids is required to precisely capture the dynamic voltage fluctuations on the P/G wires to guide the designs of reliable and working P/G networks. Since traditional SPICE-like circuit simulators do not scale well to cope with P/G grids with millions of extracted RLCK elements, new efficient transient simulation techniques are required to design robust P/G grids for Gigahertz SoC systems.

Several simulation techniques have been proposed [2, 5, 3, 4, 6, 9, 11, 14, 15] in the past. Basically there are two strategies to alleviate the circuit-size problems of P/G grid simulations. One way is to reduce the sizes of the circuits at the cost of some accuracy losses. Multigrid method [9, 15], hierarchical method [14], hierarchical model order reduction [4] are examples of such a strategy as mesh-structured circuits can be reduced by coarse grids in the multigrid method, by subcircuit reduction in the hierarchical method and by model order reduction technique. Another strategy exploits the fact that the P/G grids are linear circuits with RLCK elements. The resulting circuit matrices are symmetric positive definite at each discretized time step. As a result, iterative method is shown to be more efficient to solve such systems. Along this direction are preconditioned conjugate gradient (PCG) [3], and TLM-ADI methods [7]. TLM-ADI algorithm was shown to scale very well to attack large P/G grid circuits as it possesses linear time and space complexities. The algorithm starts with Maxwell equations and uses one of the finite-difference time-domain (FDTM) discretization technique, transmission-line modeling, to analyze the P/G grids. But, transmission line modeling requires that P/G grids have regular mesh-structured topologies as shown in Fig 2. The actual P/G grids, however, may be significantly different from such regular mesh structures. For instance, there may exist a number of RC/RLC elements for each segment in P/G grids to better model the distributed RC wires and some horizontal or vertical P/G segments may be removed to allivate signal net congestions.

In this paper, we propose a new approach to the transient analysis of large P/G grids. Our approach is based on circuit-level Alternating-Direction-Implicit iterative method. Instead of integrating the ADI method with finite-difference scheme in time-domain, we apply ADI method directly on linear circuits consisting of lumped RLC elements that are discretized at each time step. Since P/G grids at each time-discretized step become resistor-current only networks, their circuit matrices under nodal formulation are symmetric positive definite. Such *circuit-level* ADI scheme can be used to efficiently solve the P/G circuits with guaranteed convergence. Our approach is also flexible enough to analyze P/G grids with mesh-like structures, which makes our approach very amenable to analysis of P/G grids in VLSI designs ranging from middle performance ASICs to high-performance micro processors.

This paper is organized as follows. Section II reviews general ADI algorithm. Section III presents the new circuit-level ADI algorithm. Experimental results and comparison with Spice3f4 are described in Section IV. Section V concludes the paper.

II. Alternating Direction Implicit Method

The Alternating Direction Implicit method was introduced in the mid-50s by Peaceman and Rachford [10] for solving equations arising from finite difference discretization of Partial Differential Equations (PDF).

From iterative method's perspective, ADI method can be regarded as special relaxation method, where a large system is decomposed (relaxed) into a number of smaller systems such that each of them can be solved efficiently and the solution of the whole system is obtained from the solutions of the sub-systems in an iterative way.

In general, for linear system $Ax = b$, system matrix A can generally be split as $A = M - N$. The corresponding iteration equation become

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b, \tag{1}$$

For the iteration (1) to be practical, it must be easy to solve the linear system $Mx = y$. The Eq.(1) can be viewed as an iteration on a *preconditioned* system [13]

$$M^{-1}Ax = M^{-1}b, \tag{2}$$

where M is the preconditioner. For ADI algorithm, the system matrix A is decomposed into

$$A = H + V \tag{3}$$

where H and V are the elements (or discretized differential operations) in matrix A and are related to horizontal and vertical directions respectively. As a result, we end up with two sub-iterations for every ADI iteration. The Peaceman-Rachford ADI algorithm is as follows:

Peaceman-Rachford ADI()
1. **for** k = 0,1, ..., until convergence
2. Solve $(H + \rho I)x_{k+\frac{1}{2}} = (\rho I - V)u_k + b$
3. Solve $(V + \rho I)x_{k+1} = (\rho I - H)u_{k+\frac{1}{2}} + b$

Fig. 1: An Peaceman-Rachford ADI iteration.

Here ρ is a positive acceleration parameter. It can be seen that the preconditioner M and matrix N in Eq. (1) for an ADI iteration are

$$\begin{aligned} M &= \frac{1}{2\rho}(H + \rho I)(V + \rho I) \\ N &= \frac{1}{2\rho}(H - \rho I)(V - \rho I) \end{aligned}$$

It is shown that if the H and V are symmetric positive definite and $\rho > 0$, ADI algorithm always converges [13].

III. Circuit-Level ADI Algorithm for Transient Analysis of P/G Grids

In this section, we detail our circuit-level ADI algorithm. First we look at the P/G grid circuit shown in Fig. 2. To simplify our discussion, we do not include inductors in our P/G grids. But we can easily add inductors into our models.

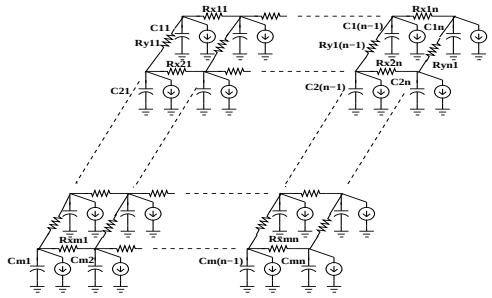

Fig. 2: A regular mesh-structured P/G grid

The behavior of such a P/G grid system can be expressed by modified nodal analysis (MNA) as following ordinary differential equation set

$$Gx + C\dot{x} = b(t) \tag{4}$$

where x is a vector of node voltages, and inductor and source currents; G is the admittance matrix; C includes the capacitance and inductance terms, and $b(t)$ consists of independent timing varying current sources and voltage sources.

By using Trapezoidal integration formula, Eq.(4) can be written as

$$(G/2 + C/h)x(t+h) = b(t+h) + (C/h - G/2)x(t) \tag{5}$$

where h is the time step. The equation can be further written as $A'x(t+h) = b'$ where $A' = (G/2 + C/h)$ and $b' = b(t+h) + (C/h - G/2)x(t)$. As a result, we need to solve a resistive-only network as matrix A' is not a function of time t. More specifically, at every time step, we replace every capacitor and inductor with a constant current source and a resistor in parallel (by using Norton's companion form).

Once we have a network with only resistors and constant current sources, we can decompose A' into two matrices $A' = H + V$, where H consists of stamps of all the horizontal resistive branches and V consists of stamps of the vertical resistive branches. For the grounded resistive branches, we can assign them either to H or V or evenly distribute them between H and V. After this we solve the following two subsystems sequentially

$$\begin{aligned} Hx(t + h/2) &= b(t + h/2) - Vx(t) & (6) \\ Vx(t+h) &= b(t+h) - Hx(t + h/2) & (7) \end{aligned}$$

Note that we can also add an acceleration parameter ρ as in Peaceman-Rachford ADI algorithm. From circuit-theory perspective, such a decomposition of A' into H and V matrices is equivalent to partitioning mesh structured circuit A' into a number of ladder structured subcircuits as shown in Fig. 3 where all the subcircuits represent the H matrix. Note that each vertical floating resistive branch is replaced by two current sources attached to its two terminals respectively in those subcircuits as shown in Fig. 4. The current sources actually are coming from $-Vx(t)$ terms in (6). So in Fig. 4, voltage V_2 and V_1 are

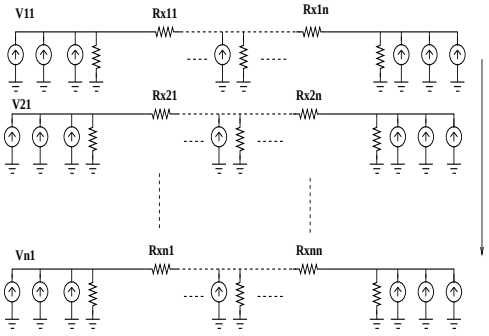

Fig. 3: The decomposed ladder subcircuits represented by matrix H

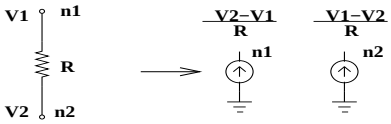

Fig. 4: The equivalent circuit for a vertical resistive branch in H

the voltages at time t and they become known constants at time $t+h$.

For a circuit with a ladder structure shown in Fig. 3, it can be solved linearly as its circuit matrix is tridiagonal. ADI algorithm first solves all the ladder subcircuits in y direction (the vertical direction) sequentially from top to bottom, which corresponds to Eq.(6), and then it solves all the ladder subcircuits in x direction (the horizontal direction) sequentially from left to right, which corresponds to Eq.(7). At each direction, once an unknown variable becomes known (solved) at present iteration, it will be used immediately in the solving of unsolved neighbor subcircuits.

Since each ladder subcircuit can be solved linearly, and we need to solve $2n$ subcircuits in both x and y direction (suppose we have a $n \times n$ mesh-structured circuit, the time complexity of one ADI iteration is $O(N)$, where $N(= n \times n)$ is the number of nodes in the circuit. If no strict convergence is required and a limited number of ADI iterations are applied at each time step at cost of some accuracy loss, ADI algorithm is of linear time complexity.

Our method can also be extended to analyze P/G grids where P/G segment between two horizontal wires or vertical wires many consists of a number of RC/RLC ladder chain circuits as shown in Fig. 5 For such a circuit, we

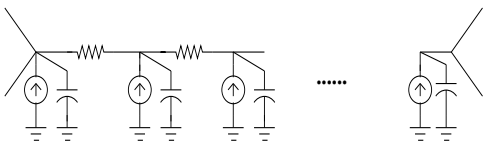

Fig. 5: A RC chain circuit in a P/G segment

can first reduce those chain circuit into a equivelent circuit consists of only the terminal nodes in an error free manner. Then the normal ADI partition of the reduced circuit matrix is carried out. Skipping the intermidiate nodes in the chain circuit reflects the fact that for ladder circuits, once we know the two terminal nodes which connect the ladder circuit to the rest of the circuit, all the intermediate nodes can be computed by the superposition principle.

IV. Experimental Results

The proposed simulation algorithm is implemented in C and $C++$. All the experimental results are obtained on a *Linux* workstation with dual 1GHz P-III processors and 1GMB memory. All the P/G circuits with time varying current sources are created based on real P/G grid circuits from industry. The *SuperLU* [12] package is used to solve each subcircuit with minimum degree pivoting.

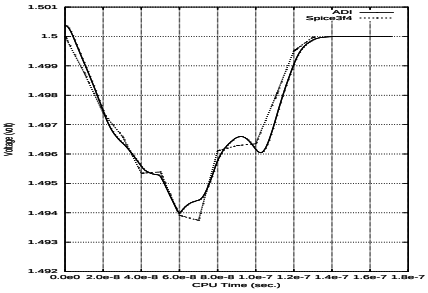

Fig. 6: Transient response of the new ADI algorithm compared with Spice3f4

First, we compare the waveforms of our algorithm with that of Spice3f4 in terms of accuracy. Fig. 6 shows the transient response of the new ADI method compared with that of Spice3f4 on a P/G grid circuit. The time interval is $[0, 170ns]$ with $1ns$ as time step for both new ADI algorithm and Spice3f4. The acceleration parameter ρ is set to 100. Our experience shows that if ρ is a small number (close to 1), the iteration may not be able to converge due to some numerical problems. While very large ρ seems slowing down the convergence rate. $\rho = 100$ gives a good trade off between speed and convergence. Also in our experiments, we only perform one ADI iteration for all the circuits. We notice that for some large circuits, one ADI iteration may leads to error larges than 10%. But the overall waveforms are still close the real ones.

We then test our program on a number of regular mesh-structured P/G networks with complexities ranging from 100 nodes to 1 million nodes. For each circuit, 100 steps are computed and the CPU run time versus number of nodes are shown in Fig. 7 for the new algorithm

It can be seen that the CPU time of ADI algorithm grow linearly with the number of nodes in the P/G grid circuits. Fig 7 compares the CPU time of ADI with Spice3f4. Both ADI algorithm and Spice3f4 compute 100 steps for each circuit. It can be seen that ADI algorithm is about two orders of magnitude faster than Spice3f4 for large P/G grids.

Last, we apply ADI algorithm to analyze mesh-like structured P/G grids. Table I report the circuit statistics and experimental results. The column *# rows* and *# cols* gives the number horizontal rows and vertical columns in

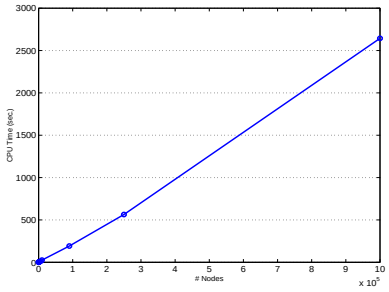

Fig. 7: CPU Time versus number of nodes

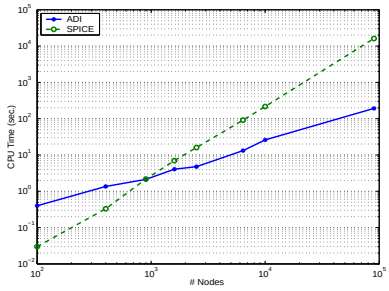

Fig. 8: Comparison with Spice3f4 in CPU time

each P/G grid circuit. *#elems* is the number of RC components in each P/G segment and *# node* is total number of nodes in the circuit.

TABLE I
STATISTICS ON THE ORIGINAL AND REDUCED NETWORKS

Ckts	# rows	# cols	# elems	#nodes	CPU (sec.)
a20_5	20	20	5	3440	1.74
a30_5	30	30	5	7800	4.42
a60_10	60	60	10	67320	56.55
a90_15	90	90	15	2.32×10^5	193.7
a120_20	120	120	20	5.57×10^5	456.63

Table I shows that the new ADI algorithm is very efficient to analyze mesh-like structured P/G grid circuits. The CPU time grows also linearly with the sizes of the P/G grid circuits.

V. CONCLUSION

In this paper, we have proposed a circuit-level alternating-direction implicit (ADI) iterative approach to analyzing large P/G grid circuits consisting of lumped RLC components. Our approach works at circuit-level and does not have the stability issue associated the existing transmission line modeling based ADI method. The new algorithm is guaranteed to converge as the discretized RLC network matrices are symmetric positive definite. The new algorithm is also able to handle mesh-like structured P/G grids, which can be found in many ASIC applications. The CPU time of new algorithm scales linearly with circuit sizes. Experimental results demonstrated that the new algorithm is about two orders of magnitude faster than Spice3f4 at small accuracy loss.

Acknowlagemenet:
This work is partially funded by the University of California Acedemic Senate Research Fund.

REFERENCES

[1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Reading, Massachusetts:Addison-Wesley Publishing Company, 1990.

[2] 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.

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

[4] 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.

[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] International Technology Roadmap for Semiconductors(ITRS), 2001 Edition, *http://public.itrs.net.*

[9] S. R. Nassif and J. N. Kozhaya, "Fast power grid simulation", in *Proc. ACM/IEEE Design Automation Conf.*, pp. 151-161, June, 2000.

[10] D. Peaceman and H. Rachford, "The numerical solution of elliptic and parabolic differential equations" *Journal of SIAM*, vol. 3, pp. 28-41, 1955.

[11] 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.

[12] http://www.nersc.gov/ xiaoye/SuperLU/

[13] Y. Saad, *Iterative methods for sparse linear systems*, Boston: International Thomson Publishing, 1996.

[14] M. Zhao, R. V. Panda, S. S. Sapatnekar and D. Blaauw, "Hierarchical analysis of power distribution networks", *IEEE Trans. Computer-Aided Design*, vol. 21, no. 2, pp. 159–168, Apr. 1990.

[15] Z. Zhu, B. Yao and C.-K. Cheng,"Power network analysis using an adaptive algebraic multigrid approach," *Proc. IEEE/ACM International Conf. on Computer-Aided Design*, pp. 105-108, 2003.