Short Papers

The Power Grid Transient Simulation in Linear Time Based on 3-D Alternating-Direction-Implicit Method

Yu-Min Lee and Charlie Chung-Ping Chen

***Abstract*—The rising power consumption and clock frequency of very large scale integration technology demand robust and stable power delivery. Extensive transient simulations on large-scale power delivery structures are required to analyze power delivery fluctuation caused by dynamic IR drop and Ldi/dt drop as well as package and on-chip resonance. In this paper, we develop a novel and efficient transient simulation algorithm for the power distribution networks. Our algorithm, three-dimensional (3-D) transmission-line-modeling alternating-direction-implicit (TLM-ADI) method, first models the power delivery structure as 3-D transmission line shunt-node structure and transfer those equations to the telegraph equation. Finally, we solve it by the alternating direction implicit method. The 3-D TLM-ADI method, with *linear runtime and memory requirement*, is also *unconditionally stable*, which ensures that the time steps are not limited by any stability requirement. Extensive numerical simulation results show that the proposed algorithm is not only over 300 000 times faster than SPICE but also extremely memory saving and accurate.**

***Index Terms*—Alternating direction implicit, power/ground, simulation, transmission-line-modeling.**

I. Introduction

Due to the ever-increasing clock frequency and the aggressively shrinking feature sizes of the very large scale integration (VLSI) technology, robust power distribution network is crucial to ensure the quality of power delivery of VLSI chips. This makes the issues of the design and verification of the power grid analysis more important. The improper design of power grids can degrade the circuit performance and the reliability. To obtain a robust design, numerous researchers studied the impact and proposed solutions of this problem [1]–[5].

There are many sources of power fluctuation such as IR drop, Ldi/dt drop, and resonance issues. Although the IR drop can be simply examined by the dc analysis, the Ldi/dt drop issues need to be analyzed by the transient simulation due to the differentiation nature of Ldi/dt drop. Hence, extensive transient simulations are required during the design process to ensure the design quality of power delivery. Reference [6] decoupled the power delivery structure, and transistors simulation to enhance the simulation speed. However, owing to the tremendous amount of the power delivery elements, general purpose circuit simulators such as SPICE [7] require long runtime and memory consumption. Therefore, it is crucial to develop efficient power grid transient simulation engines.

Several techniques [8]–[10] have been developed to speed up the analysis. Reference [8] presented the transmission matrix method to reduce the memory usage and CPU time for analysis. The method is based on a multiinput, multioutput transfer function which enables the entire power distribution network to be computed as the product of several small individual sparse square matrices. The transmission matrix method is 7–13 times faster than SPICE and saves memory requirements. Reference [9] developed an efficient modified nodal analysis (MNA) solver to speed up the dc and transient simulation of the power delivery circuits. This MNA solver is based on the preconditioned Krylov subspace-iterative method, which has been shown to be significantly faster than traditional iterative methods without preconditioning. It is about 20–50 times faster than the SPICE, requires less than 70% memory space, and the solution is exactly the same as SPICE's.

Recently, *EE Times* reported one of the most promising methods, the TLM-ADI method, which was proposed by Lee and Chen [10]. They proposed to use the transmission line modeling (TLM) [11] method to perform the time-domain simulation. TLM is closely related to the finite-difference time-domain (FDTD) method, which is one of the most popular and powerful computational electromagnetic techniques in the microwave simulation field [12]–[14]. The TLM method differs from FDTD in the sense that it utilizes transmission line cells to model the structure and directly solves the voltage and current quantities while FDTD uses Yee cell structure to obtain electric and magnetic fields. Since voltages and currents are the major focus of the VLSI power delivery analysis, TLM method can be applied directly to perform power delivery transient simulation. The TLM method has been successfully applied to analyze the two-dimensional (2-D) LC networks by Gwarek [15]. Unfortunately, the time-step size is restricted by the minimum grid cell size (Courant stability condition as the standard FDTD method [14]).

Lee and Chen [10] tried to directly solve the KCL and KVL equations by utilizing the transmission line equations. Although their method is an unconditionally stable alternating-direction-implicit [16] (ADI) scheme for the 2-D power grid networks, it cannot be directly extended to the three-dimensional (3-D) power grids. In this paper, instead of solving the KCL and KVL equations, we first set up the transmission line equations of the 3-D power grid networks. Then, we transfer those equations to the telegraph equation, and develop an unconditionally stable ADI algorithm, which relaxes the time-step constraint. With this new method, the upper bound of the time step is only limited by the accuracy requirement rather than the stability requirement. Thus, it greatly lightens the computational load due to the reduction of number of time steps. Furthermore, the runtime and memory is linear with the number of total nodes N since the method only solves around $N^{2/3}$ *tridiagonal* matrix equations with dimension $N^{1/3} \times N^{1/3}$ at each time step. Extensive experimental results show that our algorithm is not only orders of magnitude faster than SPICE but also extremely memory saving and accurate.

The remainder of the paper is organized as follows. First, the review of the finite-difference algorithm, and the relation between the modified nodal analysis (MNA), transmission line equations (TLE), and the telegraph equation will be studied in Section II. Then, the derivation of the 3-D TLM-ADI algorithm with its two main features, unconditional stability and linear runtime, will be presented in Section III. Finally, the numerical experiments and conclusion of this paper will be given in Sections IV and V.

II. Power Grid Modeling and Simulation With the Finite Difference Method

The power distribution networks are modeled by a 3-D shunt-node structure of the transmission line grids as illustrated in Fig. 1. Since

Manuscript received June 19, 2002; revised November 4, 2002. This work was supported in part by the National Science Foundation under Grants CCR-0093309 and CCR-0204468, and by Intel Corporation. This paper was recommended by Associate Editor F. N. Najm.

Y.-M. Lee is with the Department of Communication Engineering, National Chiao Tung University, Hsin-chu 300, Taiwan (e-mail: yumin@cm.nctu.edu.tw).

C. C.-P. Chen is with the Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: cchen@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TCAD.2003.818373

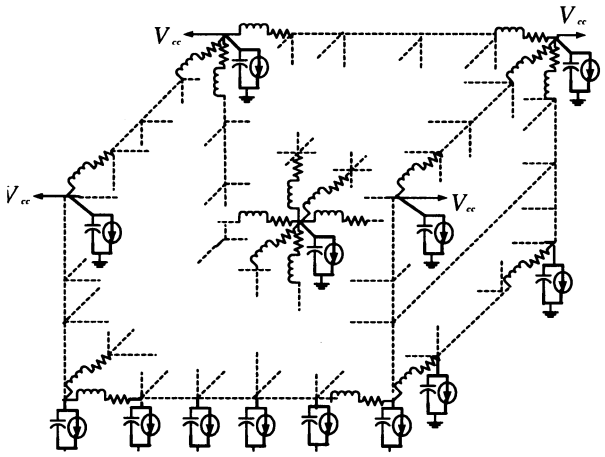

Fig. 1. Power grids modeling.

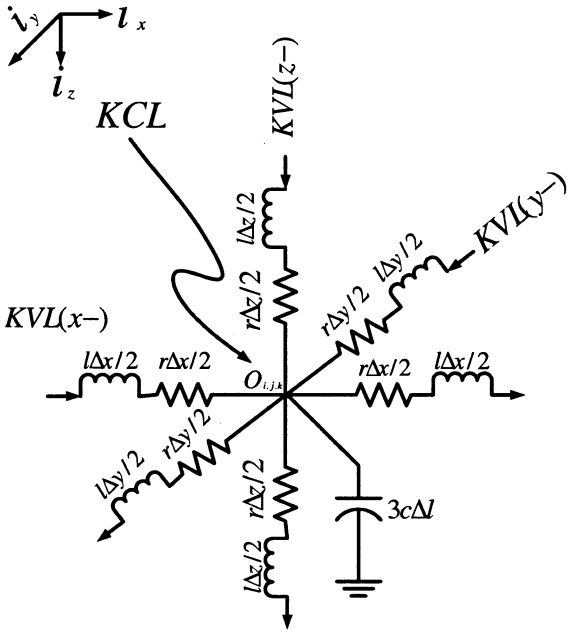

Fig. 2. KCL and KVL for a cell.

the structures of the ground and power networks are the same, Fig. 1 only shows the power delivery networks. This model results in identical formulations for both the analysis of the power and ground networks. For simplicity, in the remainder of this paper, the analysis of the power distribution network is assumed. For each cell as shown in Fig. 2, the wire segments are represented by resistors and inductors connected in series in x and y directions with a capacitor connected to the ground networks, and the vias are modeled as resistors and inductors connected in series in z direction. The parameters r, l, and c are resistance per unit length, inductance per unit length, and capacitance per unit length, respectively. Once the circuit model has been set up, the system matrices are created by using the transient nodal analysis. First, the KCL at the center node $O_{i,j,k}$, and the KVL along the x, y, and z directions of the center node are applied to each cell, as shown in Fig. 2. The KCL and KVL equations for a node $O_{i,j,k}$ at position (x_i, y_j, z_k) can be written as follows (the independent current sources have been ignored for simplicity):

$$\tilde{\mathbf{C}}_{ijk}\frac{\partial}{\partial t}\mathbf{x}_{ijk} = -\tilde{\mathbf{G}}_{ijk}\mathbf{x}_{ijk}. \tag{1}$$

Then, assembling the KVL and KCL equations for each cell, the full system equations can be represented as

$$\tilde{\mathbf{C}}\frac{\partial}{\partial t}\mathbf{x} + \tilde{\mathbf{G}}\mathbf{x} = 0 \tag{2}$$

where $\mathbf{x}$ is the vector of nodal voltages and branch currents. The above system equations are equivalent to the modified nodal analysis (MNA) equations.

1) Connection Between MNA and Transmission Line Equations: Multiplying both sides of (1) by the inverse of $\tilde{\mathbf{C}}_{ijk}$, and approaching $\Delta x, \Delta y, \Delta z$, and Δl to zeros with the uniform internodal distance assumption $(\Delta x = \Delta y = \Delta z = \Delta l)$, leads to the following:

$$\frac{\partial v}{\partial y}t = \frac{1}{3c}\left(-\frac{\partial i_x}{\partial x} - \frac{\partial i_y}{\partial y} - \frac{\partial i_z}{\partial z}\right) \tag{3}$$

$$\frac{\partial i_x}{\partial t} = \frac{1}{l}\left(-\frac{\partial v}{\partial x} - r i_x\right) \tag{4}$$

$$\frac{\partial i_y}{\partial t} = \frac{1}{l}\left(-\frac{\partial v}{\partial y} - r i_y\right) \tag{5}$$

$$\frac{\partial i_z}{\partial t} = \frac{1}{l}\left(-\frac{\partial v}{\partial z} - r i_z\right). \tag{6}$$

The above equations are the general transmission line equations which can be solved by the related techniques such as TLM and FDTD methods [12]. The procedures and concepts of the general finite-difference methods for solving the 3-D TLE are quite simple. First, the domain ($x-y-z-t$ planes) of the solution is discretized by a net with a finite number of mesh points $(x_i, y_j, z_k, t_n) = (i\triangle x, j\triangle y, k\triangle z, n\triangle t)$, which is denoted as $\cdot^n_{i,j,k}$. Then the derivatives at each mesh point are replaced by the finite difference. There are many ways to perform the finite difference, such as forward, backward, or central difference. For example, by using the central difference, the $\partial v(x_i, x_j, z_k, t)/\partial t_{n+1/2}$ and $\partial i_x(x, y_j, z_k, t_{n+1/2})/\partial x_i$ can be approximated as

$$\frac{\partial v(x_i, y_j, z_k, t)}{\partial t_{n+1/2}} \approx \frac{-v^n_{i,j,k} + v^{n+1}_{i,j,k}}{\triangle t} \tag{7}$$

$$\frac{\partial i_x(x, y_j, z_k, t_{n+1/2})}{\partial x_i} \approx \frac{-i^{n+1/2}_{x_{i-1/2,j,k}} + i^{n+1/2}_{x_{i+1/2,j,k}}}{\triangle x}. \tag{8}$$

The $\partial i_x/\partial t, \partial i_y/\partial t, \partial i_z/\partial t, \partial i_y/\partial y, \partial i_z/\partial z, \partial v/\partial x, \partial v/\partial y,$ and $\partial v/\partial z$ also can be approximated by the similar way. The branch current, i_x, can be approximated by the central-time-average

$$i_x(x_{i+1/2}, y_j, z_k, t_n) \approx \frac{i^{n-1/2}_{x_{i+1/2,j,k}} + i^{n+1/2}_{x_{i+1/2,j,k}}}{2}. \tag{9}$$

The i_y, and i_z can also be approximated in a similar way. Plugging the above approximations into (3)–(6), we can get a simple explicit finite-difference updating scheme, which is an extension of the 2-D circuit [15]. Each nodal voltage and branch current at each time step can be easily solved, since only one unknown variable appears in each difference equation. This scheme suffers on the Courant stability constraint, [12], [13] which is

$$\triangle t \le \frac{1}{\frac{1}{\sqrt{lc}}\sqrt{\frac{1}{(\triangle x)^2} + \frac{1}{(\triangle y)^2} + \frac{1}{(\triangle z)^2}}}. \tag{10}$$

As the feature size of VLSI technology is decreasing to 0.1 μm, and with $1/\sqrt{lc}$ being one-half of lightspeed, the Courant limit is close to 0.3838 fs. Thus, it needs around 2.57×10^6 time steps to simulate a 1-ns period.

2) Connection Between TLE and Telegraph Equation: In order to solve (3)–(6), we can first differentiate (3)–(6) with respect to t, x, y, and z, then combine the results with (3). This leads to a second order partial differential equation as follows:

$$3lc\frac{\partial^2 v}{\partial t^2} + 3rc\frac{\partial v}{\partial t} - \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) = 0. \tag{11}$$

After that both sides of (11) are divided by $3lc$ to give the telegraph equation

$$\frac{\partial^2 v}{\partial t^2} + a\frac{\partial v}{\partial t} - b\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) = 0 \tag{12}$$

where $a = r/l$, and $b = 1/3lc$.

Hence, we can solve the telegraph equation (12) instead of solving the transmission line (3)–(6). Extending the one-dimensional (1-D) simple implicit FDTD method [17] of the telegraph equation to the above 3-D telegraph equation, (12) becomes

$$\begin{aligned}&\frac{v_{i,j,k}^{n+1} - 2v_{i,j,k}^{n} + v_{i,j,k}^{n-1}}{(\triangle t)^2} + a\frac{v_{i,j,k}^{n+1} - v_{i,j,k}^{n-1}}{2\triangle t}\\ &\quad - b\left\{\frac{v_{i+1,j,k}^{n+1} - 2v_{i,j,k}^{n+1} + v_{i-1,j,k}^{n+1}}{(\triangle x)^2}\right.\\ &\quad + \frac{v_{i,j+1,k}^{n+1} - 2v_{i,j,k}^{n+1} + v_{i,j-1,k}^{n+1}}{(\triangle y)^2}\\ &\quad \left. + \frac{v_{i,j,k+1}^{n+1} - 2v_{i,j,k}^{n+1} + v_{i,j,k-1}^{n+1}}{(\triangle z)^2}\right\} = 0.\end{aligned} \tag{13}$$

Although this simple implicit scheme is unconditionally stable, we need to solve a heptadiagonal system of algebraic equations at each time step. Therefore, the computational time is extremely huge.

III. The 3-D TLM-ADI Method

In this section, we will derive the 3-D TLM-ADI scheme of the simple implicit FDTD method (13) by using an general ADI procedure [18]. After the derivation, the two main features of the 3-D TLM-ADI algorithm, unconditional stability and linear runtime, will be addressed. Finally, we will extend our proposed method to the power grids with nonuniform internodal distances.

The ADI method is a well-known method for solving the partial differential equation (PDE). The main feature of ADI is to sweep directions alternately. In contrast to the standard finite-difference formulation with only one iteration to advance from the nth to $(n+1)$th time step, the formulation of the ADI method requires multilevel intermediate steps to advance from the nth to $(n+1)$th time step.

Equation (13) can be rewritten as

$$\left(I + \sum_{m=1}^{3} A_m\right) v_{i,j,k}^{n+1} - 2c_0 v_{i,j,k}^{n} + c_1 v_{i,j,k}^{n-1} = 0 \tag{14}$$

where the operators of I, A_ms, and the constants of c_0, c_1 are defined as

$$Iv_{i,j,k}^{n} \triangleq v_{i,j,k}^{n} \tag{15}$$

$$A_1 v_{i,j,k}^{n} \triangleq -\rho_x\left(v_{i+1,j,k}^{n} - 2v_{i,j,k}^{n} + v_{i-1,j,k}^{n}\right) \tag{16}$$

$$A_2 v_{i,j,k}^{n} \triangleq -\rho_y\left(v_{i,j+1,k}^{n} - 2v_{i,j,k}^{n} + v_{i,j-1,k}^{n}\right) \tag{17}$$

$$A_3 v_{i,j,k}^{n} \triangleq -\rho_z\left(v_{i,j,k+1}^{n} - 2v_{i,j,k}^{n} + v_{i,j,k-1}^{n}\right) \tag{18}$$

$$c_0 \triangleq \frac{1}{(\triangle t)^2}\Big/\left(\frac{1}{(\triangle t)^2} + \frac{a}{2\triangle t}\right) \tag{19}$$

$$c_1 \triangleq \left(\frac{1}{(\triangle t)^2} - \frac{a}{2\triangle t}\right)\Big/\left(\frac{1}{(\triangle t)^2} + \frac{a}{2\triangle t}\right) \tag{20}$$

the constants of ρ_x, ρ_y, and ρ_z are

$$\rho_x = \frac{b}{(\triangle x)^2}\Big/\left(\frac{1}{(\triangle t)^2} + \frac{a}{2\triangle t}\right) \tag{21}$$

$$\rho_y = \frac{b}{(\triangle y)^2}\Big/\left(\frac{1}{(\triangle t)^2} + \frac{a}{2\triangle t}\right) \tag{22}$$

$$\rho_z = \frac{b}{(\triangle z)^2}\Big/\left(\frac{1}{(\triangle t)^2} + \frac{a}{2\triangle t}\right) \tag{23}$$

TABLE I
THE 3-D TLM-ADI ALGORITHM

The 3D TLM-ADI Algorithm

Input= $v_{i,j,k}^{n}$, $v_{i,j,k}^{n-1}$ $\forall i,j,k$ **Output=** $v_{i,j,k}^{n+1}$ $\forall i,j,k$

Begin

Sub-Iteration 1:

$$\begin{aligned}&-\rho_x v_{i+1,j,k}^{n+1(1)} + (1+2\rho_x)v_{i,j,k}^{n+1(1)} - \rho_x v_{i-1,j,k}^{n+1(1)}; \quad \forall i,j,k\\ &\quad = -(A_2 + A_3)v_{i,j,k}^{n+1(*)} + (2c_0 v_{i,j,k}^{n} - c_1 v_{i,j,k}^{n-1})\end{aligned}$$

Sub-Iteration 2:

$$\begin{aligned}&-\rho_y v_{i,j+1,k}^{n+1(2)} + (1+2\rho_y)v_{i,j,k}^{n+1(2)} - \rho_y v_{i,j-1,k}^{n+1(2)}\\ &\quad = v_{i,j,k}^{n+1(1)} + A_2 v_{i,j,k}^{n+1(*)}; \qquad \forall i,j,k\end{aligned}$$

Sub-Iteration 3:

$$\begin{aligned}&-\rho_z v_{i,j,k+1}^{n+1(3)} + (1+2\rho_z)v_{i,j,k}^{n+1(3)} - \rho_z v_{i,j,k-1}^{n+1(3)}\\ &\quad = v_{i,j,k}^{n+1(2)} + A_3 v_{i,j,k}^{n+1(*)}; \qquad \forall i,j,k\end{aligned}$$

End

and set

$$v_{i,j,k}^{n+1(*)} = 2v_{i,j,k}^{n} - v_{i,j,k}^{n-1} \tag{24}$$

which is a prediction of $v_{i,j,k}^{n+1}$ by the extrapolation method.

Then splitting (14) by using an ADI procedure as in [18], we get a set of recursion relations as follows:

$$\begin{aligned}(I + A_1)v_{i,j,k}^{n+1(1)} &= -(A_2 + A_3)v_{i,j,k}^{n+1(*)}\\ &\quad + \left(2c_0 v_{i,j,k}^{n} - c_1 v_{i,j,k}^{n-1}\right)\end{aligned} \tag{25}$$

$$(I + A_2)v_{i,j,k}^{n+1(2)} = v_{i,j,k}^{n+1(1)} + A_2 v_{i,j,k}^{n+1(*)} \tag{26}$$

$$(I + A_3)v_{i,j,k}^{n+1(3)} = v_{i,j,k}^{n+1(2)} + A_3 v_{i,j,k}^{n+1(*)} \tag{27}$$

where $v_{i,j,k}^{n+1(1)}$, $v_{i,j,k}^{n+1(2)}$ are the intermediate solutions and the desired solution is $v_{i,j,k}^{n+1} = v_{i,j,k}^{n+1(3)}$.

Finally, expanding A_1, A_2, and A_3 on the left side of (25)–(27), we get the 3-D TLM-ADI algorithm as in Table I.

A. Unconditional Stability

The general way to verify the stability of a finite-difference kind algorithm is to put a elemental solution into the algorithm and make sure that the amplitude of the propagation gain is no more than one. By applying the Von Neumann analysis [13], we can analytically prove that our 3-D TLM-ADI method is unconditionally stable. Consider the elemental solution of (12)

$$v_{i,j,k}^{n} = K^n e^{I(ik_x\triangle x + jk_y\triangle y + kk_z\triangle z)} \tag{28}$$

where k_x, k_y, and k_z are the wave numbers along the x, y, and z direction, respectively, and K is propagation gain. Putting this elemental solution into the 3-D TLM-ADI algorithm, and with some manipulations, we get

$$\begin{aligned}K^2 - \frac{2(R_xR_y + R_yR_z + R_zR_x + R_xR_yR_z + c_0)}{(1+R_x)(1+R_y)(1+R_z)}K\\ + \frac{R_xR_y + R_yR_z + R_zR_x + R_xR_yR_z + c_1}{(1+R_x)(1+R_y)(1+R_z)} = 0\end{aligned} \tag{29}$$

where

$$R_x = 4\rho_x \sin^2(k_x\triangle x/2) \tag{30}$$

$$R_y = 4\rho_y \sin^2(k_y\triangle y/2) \tag{31}$$

$$R_z = 4\rho_z \sin^2(k_z\triangle z/2). \tag{32}$$

The solutions of (29) are equal to

$$K = \frac{\Lambda + c_0 \pm \sqrt{\mathcal{D}}}{(1+R_x)(1+R_y)(1+R_z)} \tag{33}$$

where

$$\mathcal{D} = \sqrt{(\Lambda + c_0)^2 - (1+R_x)(1+R_y)(1+R_z)(\Lambda + c_1)}$$
$$\Lambda = R_x R_y + R_y R_z + R_z R_x + R_x R_y R_z.$$

By examining the amplitude of K, we are able to prove that the 3-D TLM-ADI algorithm is unconditionally stable in the following theorem.

Theorem 1: The 3-D TLM-ADI algorithm is unconditionally stable.

Proof: To prove that the 3-D TLM-ADI method is unconditionally stable, we need to show the amplitude of the gain factor K is less than or equal to one. Let us consider the following two cases.

- **Case 1:** $\mathcal{D} \geq 0$.

  From (19) and (30)–(32), we know that $\Lambda + c_0$ is greater or equal to zero. Hence

$$|K| \leq \frac{\Lambda + c_0 + \sqrt{\mathcal{D}}}{(1+R_x)(1+R_y)(1+R_z)} = \frac{\Lambda + c_0 + \sqrt{\mathcal{D}}}{\Lambda + c_0 + R_x + R_y + R_z + c_0 - c_1}.$$

  We only need to prove $\mathcal{D} \leq (R_x + R_y + R_z + c_0 - c_1)^2$, since $c_0 - c_1$ is also greater than zero

$$\mathcal{D} - (R_x + R_y + R_z + c_0 - c_1)^2 = -(R_x + R_y + R_z)(1+R_x)(1+R_y)(1+R_z) \leq 0.$$

  Therefore, $|K| \leq 1$.
- **Case 2:** $\mathcal{D} \leq 0$.

$$|K|^2 = \frac{\Lambda + c_1}{(1+R_x)(1+R_y)(1+R_z)} = \frac{\Lambda + \frac{\frac{1}{(\triangle t)^2} - \frac{a}{2\triangle t}}{\frac{1}{(\triangle t)^2} + \frac{a}{2\triangle t}}}{\Lambda + 1 + R_x + R_y + R_z} \leq 1.$$

Therefore, the 3-D TLM-ADI method is unconditionally stable from the above derivations. ♢

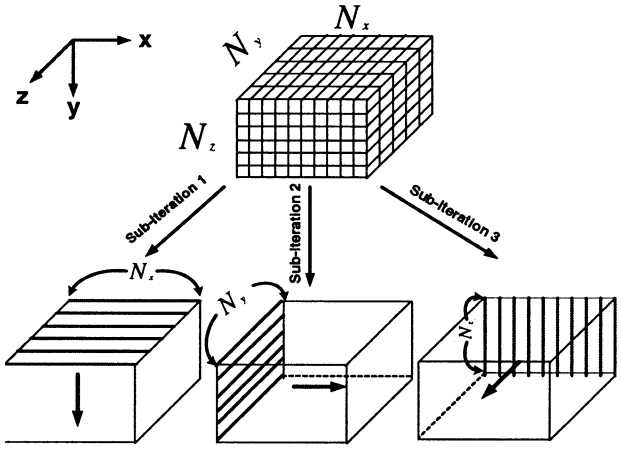

Fig. 3. Mesh points.

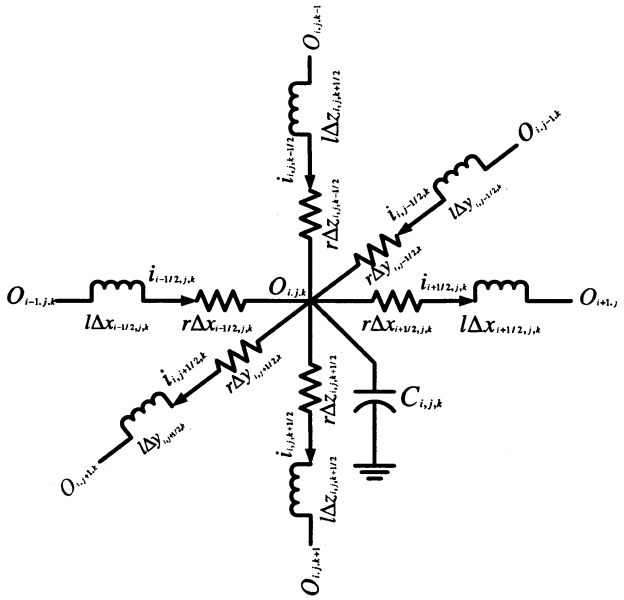

Fig. 4. Nonuniform internodal distance cell.

B. Linear Runtime

There are three subiterations need to be performed for each time step. By analyzing the runtime of each subiteration as shown in Table I, we are able to prove the computational load of the 3-D TLM-ADI algorithm is linear time at each time step in the following theorem.

Theorem 2: The runtime of 3-D TLM-ADI algorithm is $O(N)$ at each time step, where $N = N_x \times N_y \times N_z$ is the number of total nodes.

Proof: Let us consider Subiteration 1 in Table I. We can divide the set of these N nodes by $N_y \times N_z$ subsets with each one containing N_x points in the x direction, as illustrated in Fig. 3. Since only three unknown variables need to be solved in the updating equation with each (i,j,k), the coefficient matrix $\mathbf{\Phi}_{j,k}$ associated with updating $v'_{:,j,k}$ s is a tridiagonal matrix as (34) at each subset. Therefore, the runtime of updating $v'_{:,j,k}$s is linear with $O(N_x)$. There are $N_y \times N_z$ subsets in Subiteration 1. Hence, the computational load of the Subiteration 1 is $O(N_x \times N_y \times N_z)$ at each time step.

The runtime of Subiterations 2 and 3 is also $O(N)$ in a similar way. Hence, the total runtime of the 3-D TLM-ADI algorithm is $O(N)$ at each time step. ♢

$$\mathbf{\Phi}_{j,k} = \begin{bmatrix} \times & \times & 0 & \cdots & 0 \\ \times & \times & \times & \ddots & \vdots \\ 0 & \times & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \times & \times \\ 0 & \cdots & 0 & \times & \times \end{bmatrix}. \tag{34}$$

C. Nonuniform Grids

Generally, the internodal distance ($\Delta x, \Delta y$, and Δz) may be different for different cells in the power grids. Hence, we are going to extend the 3-D TLM-ADI method to handle this situation, as illustrated in Fig. 4. The parameters r and l are resistance per unit length, and inductance per unit length, respectively. $C_{i,j,k}$ is the equivalent capacitance, and $\triangle x_{i\pm 1/2,j,k}$, $\triangle y_{i,j\pm 1/2,k}$, and $\triangle z_{i,j,k\pm 1/2}$ are the internodal distances in the x, y, and z directions for a cell for which the center point is $O_{i,j,k}$, respectively.

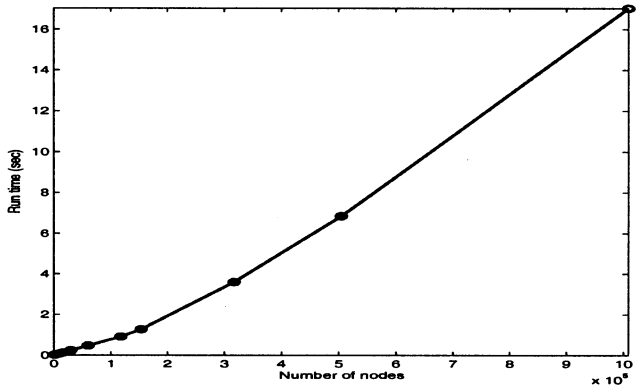

Fig. 6. Linear runtime for the 3-D TLM-ADI method.

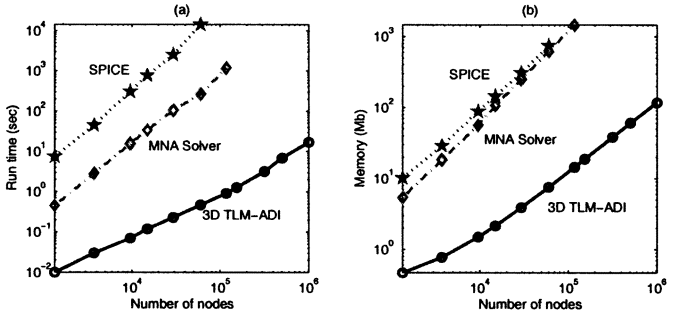

Fig. 5. Comparison of the (a) runtime and (b) memory usages between the 3-D TLM-ADI, MNA solver, and SPICE.

We first set up the KCL and KVL equations for this cell, and apply the similar derivation of (11)–(13). The 3-D simple implicit FDTD method of the telegraph equation becomes

$$\frac{v_{i,j,k}^{n+1}-2v_{i,j,k}^{n}+v_{i,j,k}^{n-1}}{(\Delta t)^2}+a\frac{v_{i,j,k}^{n+1}-v_{i,j,k}^{n-1}}{2\Delta t} - b_{i,j,k}\left\{\frac{v_{i-1,j,k}^{n+1}-v_{i,j,k}^{n+1}}{\Delta x_{i-1/2,j,k}}-\frac{v_{i,j,k}^{n+1}-v_{i+1,j,k}^{n+1}}{\Delta x_{i+1/2,j,k}} + \frac{v_{i,j-1,k}^{n+1}-v_{i,j,k}^{n+1}}{\Delta y_{i,j-1/2,k}}-\frac{v_{i,j,k}^{n+1}-v_{i,j+1,k}^{n+1}}{\Delta y_{i,j+1/2,k}} + \frac{v_{i,j,k-1}^{n+1}-v_{i,j,k}^{n+1}}{\Delta z_{i,j,k-1/2}}-\frac{v_{i,j,k}^{n+1}-v_{i,j,k+1}^{n+1}}{\Delta z_{i,j,k+1/2}}\right\}=0 \tag{35}$$

where $a = r/l$, and $b_{i,j,k} = 1/lC_{i,j,k}$.

After utilizing the same procedure as in Section III, the recursion relations of the 3-D TLM-ADI method for the nonuniform internodal distance case have the same form as (25)–(27) except for the definition of the operators A_ms (see Appendix A).

IV. Experimental Results

The 3-D TLM-ADI algorithm is implemented in C language and performed on a Pentium IV 1.2 GHZ machine. The values of r, l, and c are equal to 0.03 Ω/μm, 1.26 pH/μm, and 0.024 fF/μm, respectively. The length of each wire segment is between 15 and 100 μm, and the resistance of via is 3 Ω. Numerical results are also carried out by using the MNA solver developed by [9] and the general circuit simulator SPICE.

The comparison of runtime and memory usages is shown in Fig. 5 with ten time-step periods. The power grid model introduced in Section II is used to construct the test sets. The size of the test circuits starts from 1350 nodes ($15 \times 15 \times 6$) to 1 008 600 nodes ($410 \times 410 \times 6$).

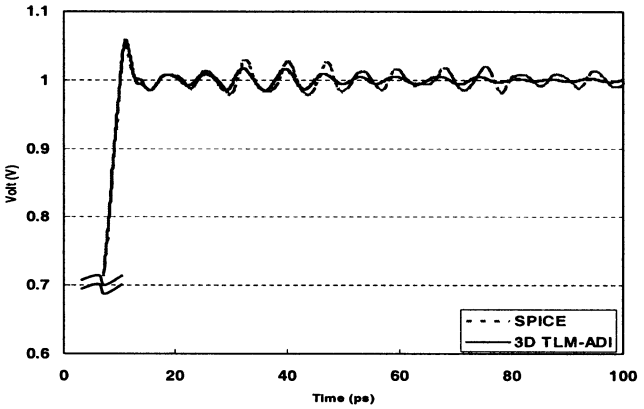

Fig. 7. DC transient response comparison between SPICE and the 3-D TLM-ADI method.

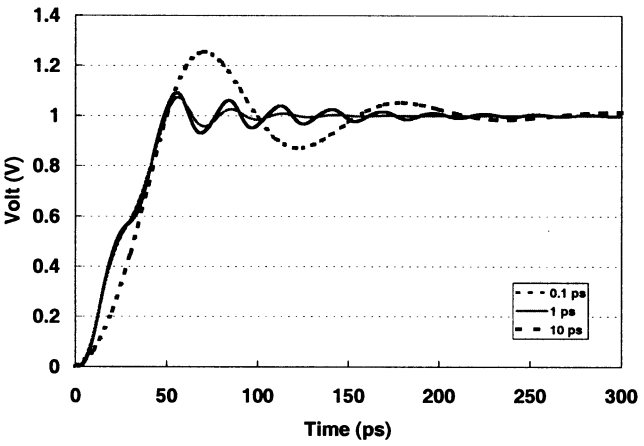

Fig. 8. DC transient response of the 3-D TLM-ADI method with different time steps.

Fig. 5(a) and (b) show that the 3-D TLM-ADI method is not only about 455 times faster than the MNA solver [9] and over 11 000 times faster than SPICE, even though the grid size is only around 30 000 nodes ($70 \times 70 \times 6$), but also extremely memory saving. The same tendency that the speedup increases with larger circuit size is also shown in Fig. 5(a). In Figs. 5(b) and 6, we demonstrate that the memory requirement and runtime for the 3-D TLM-ADI are both linear with the total number of nodes. To present the accuracy of the 3-D TLM-ADI algorithm, we simulate an *RLC* circuit with 900 ($15 \times 15 \times 4$) nodes and 0.1-ps time step. The Courant stability constraint is 0.317 49 ps in this case. Fig. 7 shows that the waveform of the 3-D TLM-ADI method agrees well with SPICE's at an arbitrary node in the power grids.

The unconditional stability of the 3-D TLM-ADI method is demonstrated in Fig. 8 with a 75-node *RLC* circuit. The Courant stability constraint is 1.5874 ps in this case. Fig. 8 shows that the time step of the 3-D TLM-ADI method is not limited by the above stability constraint but only limited by the accuracy requirement.

V. Conclusion

We have developed and implemented an efficient ADI algorithm for the transient power grids simulation, and proved its unconditional stability and linear runtime. The numerical experimental results also show that the 3-D TLM-ADI algorithm not only speeds up orders of magnitude over the SPICE but also reduces lots of the memory requirements, and the results agree well with SPICE's.

APPENDIX A

$$A_1 v^n_{i,j,k} \triangleq -\rho_{i,j,k}\frac{v^n_{i-1,j,k}-v^n_{i,j,k}}{\Delta x_{i-1/2,j,k}} - \frac{v^n_{i,j,k}-v^n_{i+1,j,k}}{\Delta x_{i+1/2,j,k}} \quad (36)$$

$$A_2 v^n_{i,j,k} \triangleq -\rho_{i,j,k}\frac{v^n_{i,j-1,k}-v^n_{i,j,k}}{\Delta y_{i,j-1/2,k}} - \frac{v^n_{i,j,k}-v^n_{i,j+1,k}}{\Delta y_{i,j+1/2,k}} \quad (37)$$

$$A_3 v^n_{i,j,k} \triangleq -\rho_{i,j,k}\frac{v^n_{i,j,k-1}-v^n_{i,j,k}}{\Delta z_{i,j,k-1/2}} - \frac{v^n_{i,j,k}-v^n_{i,j,k+1}}{\Delta z_{i,j,k+1/2}} \quad (38)$$

$$\rho_{i,j,k} = b_{i,j,k}\left/\left(\frac{1}{(\triangle t)^2}+\frac{a}{2\triangle t}\right)\right. . \quad (39)$$

ACKNOWLEDGMENT

The authors would like to thank T.-H. Chen for supplying the MNA solver code [9]. The authors would also like to thank the reviewers for their invaluable comments.

REFERENCES

[1] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," in *Proc. Design Automation Conf.*, 1997, pp. 638–643.
[2] M. K. Gowan, L. L. Biro, and D. B. Jackson, "Power considerations in the design of the alpha 21 264 microprocessor," in *Proc. Design Automation Conf.*, 1998, pp. 726–731.
[3] A. Dharchoudhury, R. Panda, D. Blaauw, and R. Vaidyanathan, "Design and analysis of power distribution networks in PowerPC microprocessors," in *Proc. Design Automation Conf.*, 1998, pp. 738–743.
[4] Y.-M. Jiang and K.-T. Cheng, "Analysis of performance impact caused by power supply noise in deep submicron devices," in *Proc. Design Automation Conf.*, 1999, pp. 760–765.
[5] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "Multigrid-like technique for power grid analysis," in *Proc. ICCAD*, 2001, pp. 480–487.
[6] M. Zhao, R. V. Panda, S. S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw, "Hierarchical analysis of power distribution networks," in *Proc. Design Automation Conf.*, 2000, pp. 150–155.
[7] L. W. Nagel, "SPICE2, A Computer Program to Simulate Semiconductor Circuits," Univ. California, Berkeley, CA, Tech. Rep. ERL-M520, 1975.
[8] J.-H. Kim, M. Swaminathan, and Y. Suh, "Modeling of power distribution networks for mixed signal applications," in *Proc. 2001 IEEE EMC Int. Symp.*, vol. 2, 2001, pp. 1117–1122.
[9] T.-H. Chen and C. P. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," in *Proc. Design Automation Conf.*, 2001, pp. 559–562.
[10] Y.-M. Lee and C. P. Chen, "Power grid transient simulation in linear time based on transmission-line-modeling alternating-direction-implicit method," in *Proc. ICCAD*, 2001, pp. 75–80.
[11] C. Christopoulos, *The Transmission-Line Modeling Method TLM*. London, U.K.: Oxford Univ. Press, 1995.
[12] M. N. O. Sadiku, *Numerical Techniques in Electromagentics*. Boca Raton, FL: CRC, 1992.
[13] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*. London, U.K.: Chapman & Hall, 1989.
[14] A. Taflove and S. C. Hagness, *Computational Electrodynamics—The Finite-Difference Time-Domain Method*. Norwood, MA: Artech House, 2000.
[15] W. K. Gwarek, "Analysis of arbitrarily shaped two-dimensional microwave circuits by finite-difference time-domain method," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 738–744, Apr. 1988.
[16] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *J. Soc. Ind. Applicat. Math.*, vol. 3, pp. 28–41, 1955.
[17] W. J. Minkowycz, E. M. Sparrow, G. E. Schneider, and R. H. Pletcher, *Handbook of Numerical Heat Transfer*. New York: Wiley, 1988.
[18] J. Douglas and J. E. Gunn, "A general formulation of alternating direction implicit methods, Part I: Parabolic and hyperbolic problems," *Numer. Math.*, vol. 6, pp. 428–453, 1964.

Hierarchical Whitespace Allocation in Top-Down Placement

Andrew E. Caldwell, *Member, IEEE*, Andrew B. Kahng, *Senior Member, IEEE*, and Igor L. Markov, *Member, IEEE*

***Abstract*—Increased transistor density in modern commercial ICs typically originates in new manufacturing and defect prevention technologies [15], [16]. Additionally, better utilization of such *low-level* transistor density may result from improved software that makes fewer assumptions about physical layout in order to reliably automate the design process. In particular, recent layouts tend to have large amounts of whitespace, which is not handled properly by older tools. We observe that a major computational difficulty arises in partitioning-driven top-down placement when regions of a chip lack whitespace. This tightens balance constraints for min-cut partitioning and hampers move-based local-search heuristics such as Fiduccia–Mattheyses. However, the local lack of whitespace is often caused by very unbalanced distribution of whitespace during previous partitioning, and this concern is emphasized in chips with large overall whitespace.**

This paper focuses on accurate computation of tolerances to ensure smooth operation of common move-based iterative partitioners, while avoiding cell overlaps. We propose a mathematical model of hierarchical whitespace allocation in placement, which results in a simple computation of partitioning tolerance purely from relative whitespace in the block and the number of rows in the block. Partitioning tolerance slowly increases as the placer descends to lower levels, and relative whitespace in all blocks is limited from below (unless partitioners return "illegal" solutions), thus preventing cell overlaps. This facilitates good use of whitespace when it is scarce and prevents very dense regions when large amounts of whitespace are available.

Our approach improves the use of the available whitespace during global placement, thus leading to smaller whitespace requirements. Existing techniques, particularly those based on simulated annealing [21], [10], can be applied after global placement to bias whitespace with respect to particular concerns, such as routing congestion, heat dissipation, crosstalk noise and DSM yield improvement.

***Index Terms*—Algorithms, design automation, integrated circuit layout.**

I. INTRODUCTION

The progression of Moore's law [18], [15] for commercial ICs has been so far maintained by steady increases in device densities as a result of innovations in manufacturing and defect prevention technologies [16]. At the same time, device density for a given process generation is also limited by the capabilities of EDA software and the assumptions made by software developers.

Historically, utilization rates increased (i.e., whitespace decreased) steadily with the introduction of three-layer, four-layer, and even five-layer metal technologies. In contrast with the preceding two-layer metal regime, three or more layers of metal brought the following changes: 1) the need for routing channels disappeared; 2) double-back (shared power and ground rail) standard-cell styles removed all

Manuscript received July 15, 2002; revised January 13, 2003.

A. E. Caldwell was with the University of California, Los Angeles, CA 90095 USA. He is now with Everychip Inc., Mountain View, CA 94041 USA (e-mail: andy@everychip.com).

A. B. Kahng was with the University of California, Los Angeles, CA 90095 USA. He is now with the Departments of Computer Science and Engineering and Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0114 USA (e-mail: abk@ucsd.edu).

I. L. Markov was with the University of California, Los Angeles, CA 90095 USA. He is now with the Department of Electrical Engineering and Computer Science, Univ. of Michigan, Ann Arbor, MI 48109-2122 USA (e-mail: imarkov@umich.edu).

Digital Object Identifier 10.1109/TCAD.2003.818375