

The results in Table I show that high fault coverage and differentiation coverage are achieved in all cases. By using the fault specific extracuster configuration algorithm, 100% fault coverage can be guaranteed at the cost of an increased number of configurations.

VIII. CONCLUSION

We have presented a hierarchical technique to define test configurations for the detection and diagnosis of interconnect faults in cluster-based FPGA architectures. We have used the concept of test transparency to define configurations which enable test access to the high-density logic cluster embedded within each FPGA tile. We have demonstrated that this technique can be used to successfully define a small set of test configurations which allow the detection and diagnosis of nearly all targeted interconnect faults.

REFERENCES

- [1] V. Lakamraju and R. Tessier, "Tolerating operational faults in cluster-based FPGAs," in *8th Int. ACM/SIGDA Symp. Field Programmable Gate Arrays*, Feb. 2000.
- [2] Altera Web Site [Online]. Available: <http://www.altera.com/>.
- [3] "Virtex data sheet," Xilinx Corp., 2000.
- [4] Atmel Corp., *Configurable Logic Design and Application Book*, 1997.
- [5] Altera Corp., "Altera apex data sheet," 2000.
- [6] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. New York: Kluwer, 1999.
- [7] M. Renovell, J. M. Portal, J. Figueras, and Y. Zorian, "Testing the interconnect of RAM-based FPGAs," *IEEE Design Test Comput.*, vol. 15, pp. 45–50, Jan.–Mar. 1998.
- [8] M. Renovell and Y. Zorian, "Different experiments in test generation for xilinx FPGAs," in *Proc. Int. Test Conf.*, 2000, pp. 854–862.
- [9] L. Zhao, D. M. H. Walker, and F. Lombardi, "Bridging fault detection in FPGA interconnects using i_{DDQ} ," in *Int. Symp. Field Programmable Gate Arrays*, Feb. 1998, pp. 95–104.
- [10] C. Stroud, S. Wijesuriya, C. Hamilton, and M. Abramovici, "Built-in self-test of FPGA interconnect," in *Int. Test Conf.*, Oct. 1998, pp. 404–441.
- [11] M. Renovell, J. M. Portal, J. Figueras, and Y. Zorian, "SRAM-based FPGAs: Testing the LUT/RAM modules," in *Proc. Int. Test Conf.*, Oct. 1998, pp. 1102–1111.
- [12] C. Metra, A. Pagano, and B. Ricco, "On-line testing of transient and crosstalk faults affecting interconnections of FPGA-implemented systems," in *Proc. Int. Test Conf.*, 2001, pp. 939–947.
- [13] G. Gibson, L. Gray, and C. Stroud, "Boundary scan access of built-in self-test for field programmable gate arrays," in *Proc. IEEE Int. ASIC*, Sept. 1997, pp. 57–61.
- [14] C. Stroud, E. Lee, and M. Abramovici, "BIST-based diagnostics of FPGA logic blocks," in *Proc. Int. Test Conf.*, Nov. 1997, pp. 539–547.
- [15] M. Abramovici, C. Stroud, C. Hamilton, S. Wijesuriya, and V. Verma, "Using roving STAR's for on-line testing and diagnosis of FPGA's in fault-tolerant applications," in *Proc. Int. Test Conf.*, Sept. 1999.
- [16] N. R. Shnidman, W. H. Mangione-Smith, and M. Potkonjak, "On-line fault detection for bus-based field programmable gate arrays," *IEEE Trans. VLSI Syst.*, vol. 6, pp. 656–666, Dec. 1998.
- [17] I. G. Harris, P. Menon, and R. Tessier, "BIST-based delay-path testing in FPGA architectures," in *Proc. Int. Test Conf.*, 2001, pp. 932–938.
- [18] J. G. Dastidar and N. A. Toubia, "Improving diagnostic resolution of delay faults in FPGA's by exploiting reconfigurability," in *IEEE Symp. Defect Fault Tolerance*, 2001, pp. 215–220.
- [19] A. Krasniewski, "Application-dependent testing of FPGA delay faults," in *Proc. EUROMICRO'99*, 1999, pp. 260–267.
- [20] —, "Enhancing detection of delay faults in FPGA-based circuits by transformations of LUT functions," in *Proc. IFAC Workshop Programmable Devices and Systems—PDS2000*, Feb. 2000, pp. 127–132.
- [21] I. G. Harris and R. Tessier, "Interconnect testing in cluster-based FPGA architectures," in *Proc. Design Automation Conf.*, June 2000.
- [22] —, "Diagnosis of interconnect faults in cluster-based FPGA architectures," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2000.
- [23] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field-Programmable Gate Arrays*. New York: Kluwer, 1992.

- [24] V. Betz and J. Rose, "Cluster-based logic blocks for FPGAs: Area-efficiency vs. input sharing and size," in *Proc. IEEE CICC*, 1997, pp. 551–554.
- [25] M. J. Y. Williams and J. B. Angel, "Enhancing testability of large-scale integrated circuits via test points and additional logic," *IEEE Trans. Comput.*, vol. C-22, no. 1, pp. 46–60, Jan. 1973.
- [26] M. Abramovici and P. R. Menon, "A practical approach to fault simulation and test generation for bridging faults," *IEEE Trans. Comput.*, vol. C-34, pp. 658–663, July 1985.

Power Grid Transient Simulation in Linear Time Based on Transmission-Line-Modeling Alternating-Direction-Implicit Method

Yu-Min Lee and Charlie Chung-Ping Chen

Abstract—The soaring clocking frequency and integration density demand robust and stable power delivery to support tens of millions of transistors switching. To ensure the design quality of power delivery, extensive transient power grid simulations need to be performed during the design process. However, the traditional circuit simulation engines are not scaled well for the complexity of power delivery. As a result, it often takes a long runtime and huge memory requirement to simulate a medium-sized power grid circuit. In this paper, the authors develop and present a new efficient transient simulation algorithm for power distribution. The proposed algorithm, transmission-line-modeling alternating-direction-implicit (TLM-ADI), first models the power delivery structure as transmission line mesh structure, then solves the transient modified nodal analysis matrices by the alternating-direction-implicit method. The proposed algorithm, with linear runtime and memory requirement, is also unconditionally stable which ensures that the time-step is not limited by any stability requirement. Extensive experimental results show that the proposed algorithm is not only orders of magnitude faster than SPICE but also extremely memory saving and accurate.

Index Terms—Alternating direction implicit, power grid, transient, transmission line modeling.

I. INTRODUCTION

The increase in the complexity of the very large scale integration (VLSI) chips and the decrease in the feature size of the chips demand larger grids for power distribution. This causes the designing and verifying of the power networks to become a challenging task. The inferiorly designed power distribution network can degrade the circuit performance, noise margin, and the reliability. Since the power grids are rapidly becoming a limiting factor in high-performance microprocessors, the ability to analyze power grids efficiently is a critical requirement to obtain a robust design [1]–[4].

Power is transferred through many complicated circuit structures. From the power supply through the PCB, packaging, I/O pins, C4-bump, and on-chip interconnect to the transistors, every portion of the circuit in the power delivery path plays a crucial role for the quality of power delivery and hence all of them need to be carefully modeled and designed. There are several sources that cause the degradation of the quality of power delivery systems such as IR drop, Ldi/dt drop,

Manuscript received January 24, 2001; revised November 19, 2001 and March 18, 2002. This work was supported by Avant! Corporation and Intel Corporation.

The authors are with the Department of Electrical and Computer Engineering, University of Wisconsin at Madison, Madison, WI 53706 USA.

Digital Object Identifier 10.1109/TCAD.2002.804082

and resonance issues. While IR drop can be simply verified by the dc analysis, the Ldi/dt drop issues need to be analyzed by transient simulation due to the differentiation nature of Ldi/dt drop.

To ensure the design quality of power delivery, extensive transient simulations are required during the design process. However, due to the complexity of on-chip power grids, it is computationally expensive to simulate all the transistors with power delivery structure. To effectively enhance the simulation speed, it has been proposed to decouple the power delivery structure simulation and transistors simulation [5]. That is, it first simulates transistors to get the drawn current waveforms and then performs linear simulation with those currents attached as independent current sources. In this way, the simulation can be effectively sped up since there are fewer elements in both circuits and the linear circuit can be simulated efficiently with only one LU decomposition. However, due to the large size and grid nature of the linear circuit, SPICE [6] does not perform well in this type of system and often takes days to complete the full simulation as well as needing many giga bytes of memory space. Hence, in order to facilitate the design of large scale power grids, it is crucial to develop an efficient transient simulation engine which is capable of performing the full-chip power grid analysis in a reasonable turnaround time.

Several techniques [7], [8] have been developed to speed up the analysis. Reference [7] presented the transmission matrix method to reduce the memory usage and CPU time for analysis. The method is based on a multi-input/multi-output 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. Recently, [8] proposed the preconditioned conjugate gradient (PCG) simulator to speed up the dc and transient simulation of the power delivery circuits. The PCG simulator is based on the preconditioned Krylov-subspace iterative method which has been shown to be significantly faster than traditional iterative methods without preconditioning. The PCG simulator is about 100 times faster than the SPICE and requires less than 70% memory space.

In this paper, we propose to use the transmission line modeling (TLM) [9] method to perform the time-domain simulation since TLM can capture both the IR and Ldi/dt drop. 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 [10]–[12], and performs electromagnetic analysis in the time domain instead of frequency domain. 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 the Yee cell structure to obtain electric and magnetic fields. Since voltage and current are the major focus for the VLSI power delivery analysis, the TLM method can be applied directly to perform power delivery transient simulation. The TLM method has been successively applied to analyze the LC networks by Gwarek [13]. Unfortunately, the size of time step is restricted by the minimum grid cell size (Courant stability condition as the standard FDTD method [12]). For example, for the VLSI technology with feature size as $0.1 \mu\text{m}$ and the dielectric permittivity as 4, the Courant limit is close to 0.47 fs. Thus it needs around 2.1×10^6 time steps to simulate a 1-ns period. Therefore, it is crucial to develop an unconditionally stable TLM method for efficient analysis.

To effectively reduce the stability limit requirement, several unconditional FDTD methods have been proposed and tested by [14]–[16] and showed good potential in the application of on-chip microwave analysis [14]. However, there is still no unconditionally stable TLM method in the literature to the authors' best knowledge. In this paper, we develop an unconditionally stable TLM algorithm, TLM-ADI,

which relaxes the time-step constraint enforced by the Courant stability limit for the traditional TLM method. This new time-stepping scheme is based on an innovative alternating-direction-implicit (ADI) method [17]. With this new method, the upper bound of the time step is only limited by the accuracy requirement rather than stability requirement. Thus, it greatly enhances the computational efficiency due to the reduction of number of time steps. Furthermore, the runtime and memory is linear with $O(N)$ (N , the total number of nodes) since at each time step it only solves around \sqrt{N} tridiagonal matrix equations with dimension $\sqrt{N} \times \sqrt{N}$. Extensive experimental results also show that our algorithm is not only orders of magnitude faster than SPICE but also extremely memory saving and accurate.

The rest of the paper is organized as follows. First, the review of the finite-difference method will be studied in Section II. Then, the numerical formulation of the proposed method will be derived, and its two main features, unconditional stability and linear run time, will be dressed in Section III. Finally, several numerical experiments, and the conclusion of this paper will be given in Sections IV and V.

II. POWER GRID MODELING AND SIMULATION WITH THE FINITE DIFFERENCE METHOD

The transmission line grids are used to model the power delivery structure as illustrated in Fig. 1. A serially connected resistor and inductor with a ground capacitor are used to represent each wire segment. The parameters r , l , and c are resistance, inductance, and capacitance per unit length, respectively. The Δx , and Δz are the internodal distance along the x and z direction, and $\Delta l = (\Delta x + \Delta z)/2$. Once the model has been set up, the system matrices are created by the transient nodal analysis. In each cell, by applying KCL at center node, and KVL around the loops of that node in both the x - y and y - z planes as shown in Fig. 2, the KCL and KVL equations for a node o_{ij} at position (x_i, z_j) can be written as (the independent current sources are ignored for simplicity)

$$\tilde{\mathbf{C}}_{ij} \frac{\partial}{\partial t} \mathbf{x}_{ij} = -\tilde{\mathbf{G}}_{ij} \mathbf{x}_{ij} \quad (1)$$

where \mathbf{x}_{ij} is the vector of nodal voltages of node o_{ij} and its four neighborhood points and branch currents through the four inductors. After assembling the KCL and KVL equations of all cells, the full system equations can be summarized as

$$\tilde{\mathbf{C}} \frac{\partial}{\partial t} \mathbf{x} + \tilde{\mathbf{G}} \mathbf{x} = \tilde{\mathbf{S}}_s(t) \quad (2)$$

where \mathbf{x} is the vector of nodal voltages and currents through the inductors and $\tilde{\mathbf{S}}_s(t)$ is the vector of voltage sources. The system equations are equivalent to the modified nodal analysis (MNA) equations.

Connection Between MNA and Transmission-Line Equation

Equation (2) can also be expressed in the transmission-line equation (TLE) formation and hence can be solved by related techniques such as TLM and FDTD methods [10].

For instance, after taking the limit of both sides of (1) with $\Delta x \rightarrow 0$, $\Delta z \rightarrow 0$, and $\Delta l \rightarrow 0$ (here, a uniform internodal distance is assumed, i.e., $\Delta x = \Delta z = \Delta l$), the general TLEs are

$$\frac{\partial v}{\partial t} = \frac{1}{2c} \left(-\frac{\partial i_x}{\partial x} - \frac{\partial i_z}{\partial z} \right) \quad (3)$$

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

$$\frac{\partial i_z}{\partial t} = \frac{1}{l} \left(-\frac{\partial v}{\partial z} - r i_z \right). \quad (5)$$

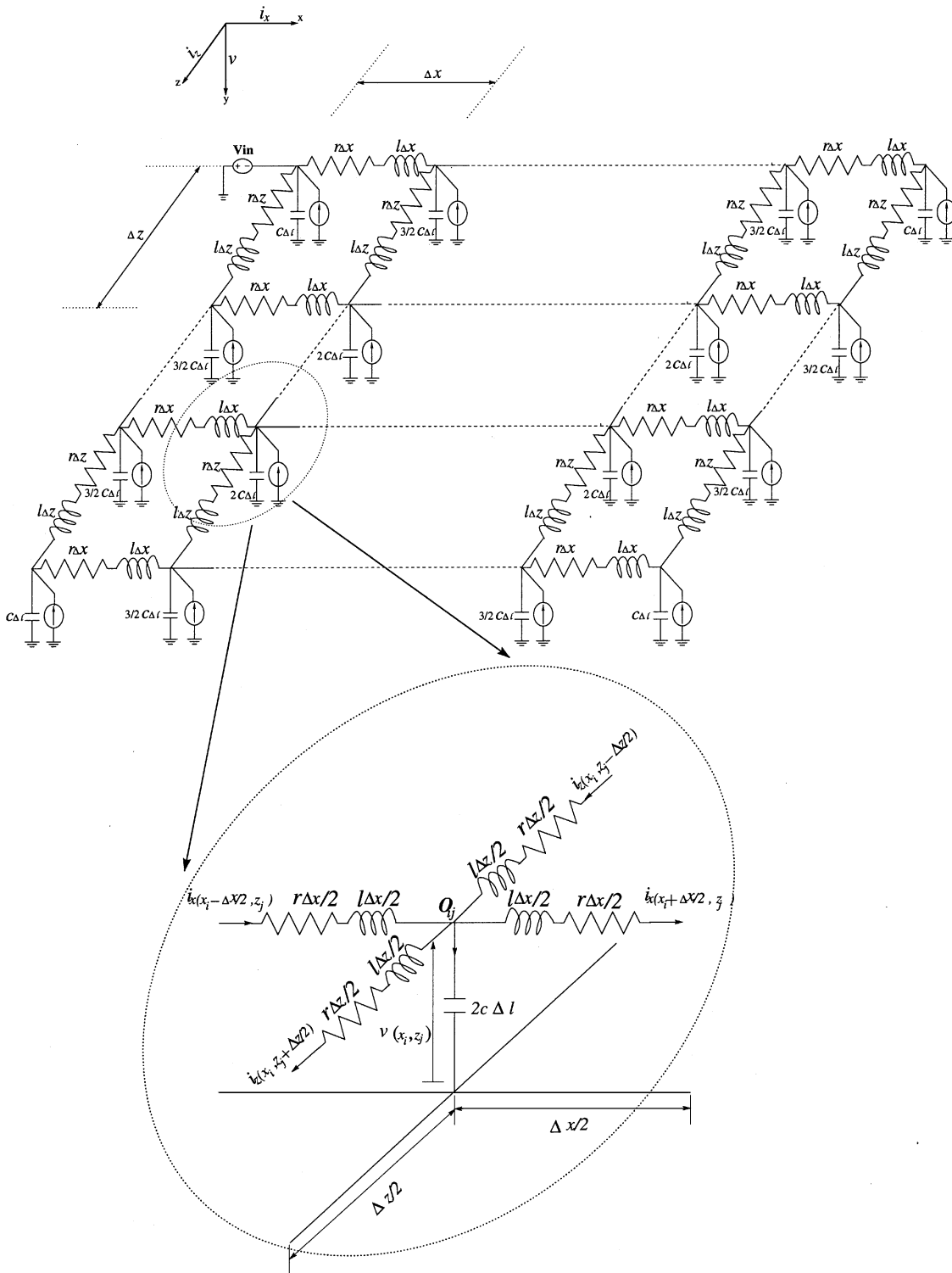


Fig. 1. The transmission line modeling of a power grid structure.

The basic concepts of finite difference schemes for solving the two-dimensional TLE are quite simple. First, the domain ($x-z-t$ planes) of the solution is subdivided by a net with a finite number of mesh points, as illustrated in Fig. 3. Each mesh point $(x_i, z_j, t_n) = (i\Delta x, j\Delta z, n\Delta t)$ is represented as " $|_{i,j}^n$." Then, the derivative at each mesh point is replaced by the finite difference. The finite difference can be done in many ways such as forward-difference, backward-difference, or centered-difference. For example, by using the centered-difference, the

$\partial v(x_i, z_j, t) / \partial t_{n+1/2}$ and $\partial i_x(x, z_j, t_{n+1/2}) / \partial x_i$ can be approximated as

$$\frac{\partial v(x_i, z_j, t)}{\partial t_{n+1/2}} \approx \frac{-v|_{i,j}^n + v|_{i,j}^{n+1}}{\Delta t} \tag{6}$$

$$\frac{\partial i_x(x, z_j, t_{n+1/2})}{\partial x_i} \approx \frac{-i_x|_{i-1/2,j}^{n+1/2} + i_x|_{i+1/2,j}^{n+1/2}}{\Delta x} \tag{7}$$

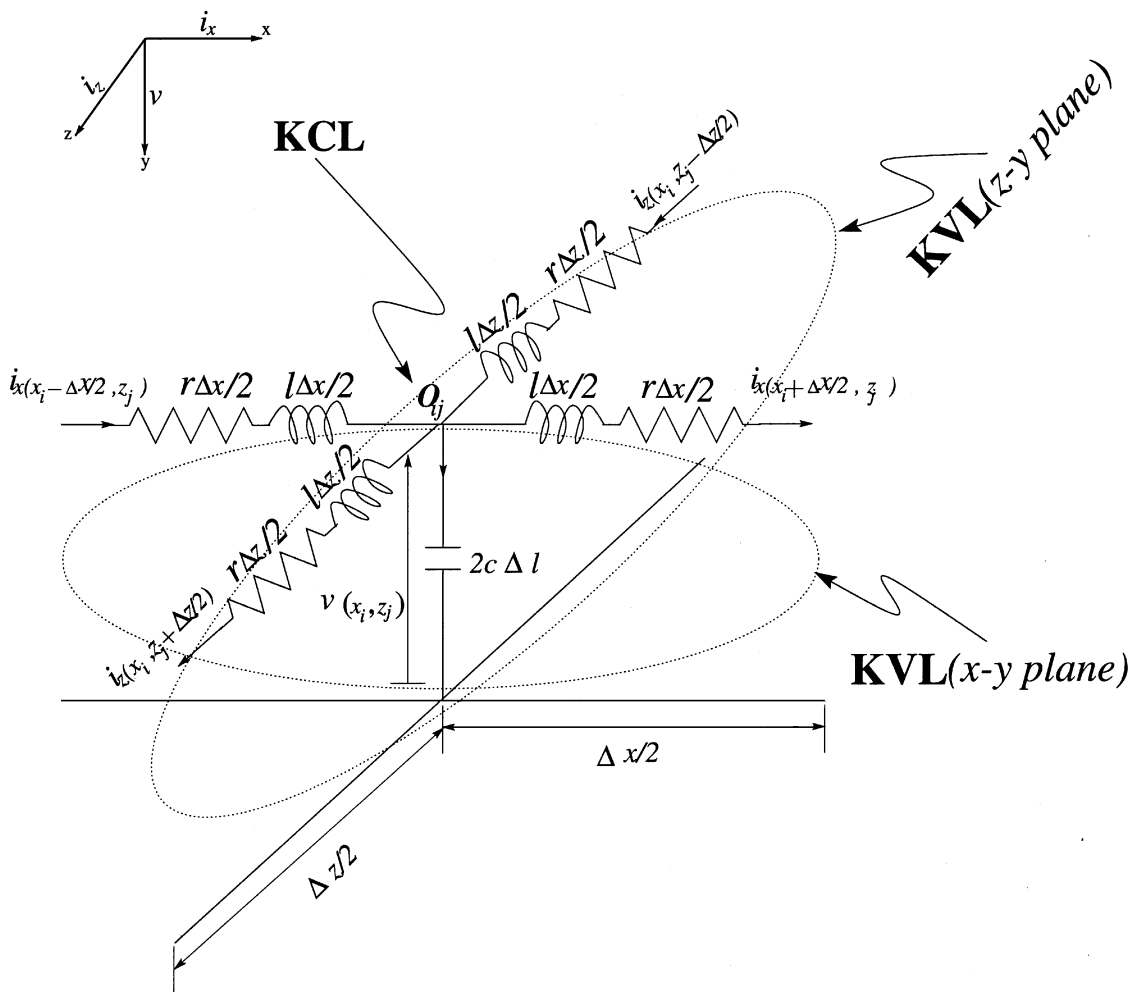


Fig. 2. KCL and KVL for a cell.

In addition, the $\partial i_x / \partial t$, $\partial i_z / \partial t$, $\partial i_z / \partial z$, $\partial v / \partial x$, and $\partial v / \partial z$ can be represented by the similar way.

Similarly, i_x and i_z can be approximated by the centered-time average as

$$i_x(x_{i+1/2}, z_j, t_n) \approx \frac{i_x|_{i+1/2, j}^{n-1/2} + i_x|_{i+1/2, j}^{n+1/2}}{2} \quad (8)$$

$$i_z(x_i, z_{j+1/2}, t_n) \approx \frac{i_z|_{i, j+1/2}^{n-1/2} + i_z|_{i, j+1/2}^{n+1/2}}{2} \quad (9)$$

Plugging the above approximated equations into (3)–(5) (for simplicity, we set $r = 0$ which is the LC circuit) we get the updating equations [13] as follows:

$$\begin{bmatrix} i_x|_{i+1/2, j}^{n+1/2} \\ i_z|_{i, j+1/2}^{n+1/2} \end{bmatrix} = \begin{bmatrix} i_x|_{i+1/2, j}^{n-1/2} \\ i_z|_{i, j+1/2}^{n-1/2} \end{bmatrix} \quad (10)$$

$$- \begin{bmatrix} -\frac{\Delta t}{l\Delta x} & \frac{\Delta t}{l\Delta x} & 0 \\ -\frac{\Delta t}{l\Delta z} & 0 & \frac{\Delta t}{l\Delta z} \end{bmatrix} \begin{bmatrix} v|_{i, j}^n \\ v|_{i+1, j}^n \\ v|_{i, j+1}^n \end{bmatrix} \quad (11)$$

$$v|_{i, j}^{n+1} = v|_{i, j}^n \quad (12)$$

$$- \begin{bmatrix} \frac{\Delta t}{2c\Delta x} & \frac{\Delta t}{2c\Delta z} \end{bmatrix} \begin{bmatrix} i_x|_{i+1/2, j}^{n+1/2} - i_x|_{i-1/2, j}^{n+1/2} \\ i_z|_{i, j+1/2}^{n+1/2} - i_z|_{i, j-1/2}^{n+1/2} \end{bmatrix} \quad (13)$$

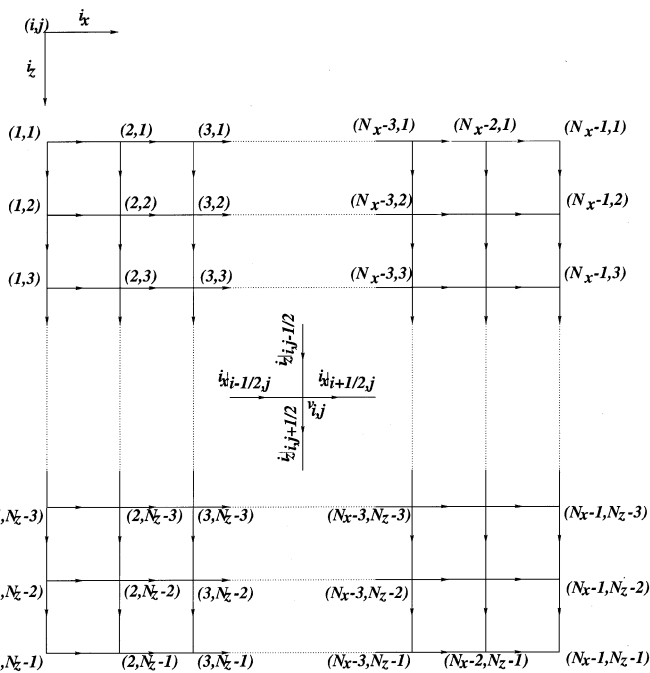


Fig. 3. Discretization of FDTD.

The above updating equations are a simple explicit finite-difference scheme. They can be easily done since only one unknown variable appears in each difference equation. It suffers on the Courant stability constraint [12] which is

$$\Delta t \leq \frac{1}{\frac{1}{\sqrt{lc}} \sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta z)^2}}}. \quad (14)$$

For example, for the VLSI technology with feature size as $0.1 \mu\text{m}$, and $1/\sqrt{lc}$ as a half of the light speed, the Courant limit is close to 0.47 fs. Thus it needs around 2.1×10^6 time steps to simulate a 1-ns period.

III. TRANSMISSION-LINE-MODELING ALTERNATING-DIRECTION-IMPLICIT METHOD

In this section, we first derive and present the TLM-ADI algorithm for the homogeneous case ($r, l, c, \Delta x$, and Δz are constants). Then, its stability is studied analytically. At the end, we generalize the TLM-ADI method to the inhomogeneous case ($r, l, c, \Delta x$, and Δz have different values at different segments) and show that the run time of the proposed algorithm is linear with $O(N)$ where N is the total number of nodes.

A. Homogeneous Case

The ADI method is a well-known method for solving the partial differential equations (PDE). The main feature of ADI is to sweep directions alternately. Here, we derive and present our unconditionally stable algorithm based on the ADI scheme. In contrast to the standard finite difference formulation with only one iteration to advance from the n th to $(n+1)$ th time step, the formulation requires one subiteration to advance from n th to $(n+1/2)$ th time step, and a second subiteration to advance from $(n+1/2)$ th to $(n+1)$ th time step. For example, considering the KCL equation (3), every term in the first subiteration is effectively discretized at $n+1/4$ as

$$\frac{\partial v}{\partial t} \Big|^{n+1/4} = \frac{1}{2c} \left(- \frac{\Delta i_x}{\Delta l} \Big|^n - \frac{\Delta i_z}{\Delta l} \Big|^{n+1/2} \right) \quad (15)$$

where $\Delta i_x = i_x(x + \Delta x/2, z) - i_x(x - \Delta x/2, z)$, and $\Delta i_z = i_z(x, z + \Delta z/2) - i_z(x, z - \Delta z/2)$. Note that the current terms are discretized at time steps n and $n+1/2$, giving an overall effect of $n+1/4$. Thus, the $\Delta i_x/\Delta l$ is evaluated explicitly from known data at time step n , while the $\Delta i_z/\Delta l$ is evaluated implicitly from as-yet known data at time step $n+1/2$. In the second subiteration, every term is effectively discretized at $n+3/4$ as

$$\frac{\partial v}{\partial t} \Big|^{n+3/4} = \frac{1}{2c} \left(- \frac{\Delta i_x}{\Delta l} \Big|^{n+1} - \frac{\Delta i_z}{\Delta l} \Big|^{n+1/2} \right). \quad (16)$$

Here, the current terms are discretized at time steps $n+1$ and $n+1/2$, giving an overall effect of $n+3/4$. Thus, the $\Delta i_z/\Delta l$ is evaluated explicitly from known data at time step $n+1/2$, while the $\Delta i_x/\Delta l$ is evaluated implicitly from as-yet known data at time step $n+1$.

This ADI scheme can also be applied to the KVL equations (4) and (5). The updating equations are listed below. Here, we have used the conventional semi-implicit formulation to evaluate the voltage and current terms at the appropriate time steps.

Subiteration 1: Advance i_x, i_z , and v from time step n to time step $n+1/2$

$$i_x \Big|_{i+(1/2),j}^{n+(1/2)} = \frac{4l-r\Delta t}{4l+r\Delta t} i_x \Big|_{i+(1/2),j}^n - \frac{2\Delta t}{\Delta x(4l+r\Delta t)} \cdot (v \Big|_{i+1,j}^n - v \Big|_{i,j}^n) \quad (17)$$

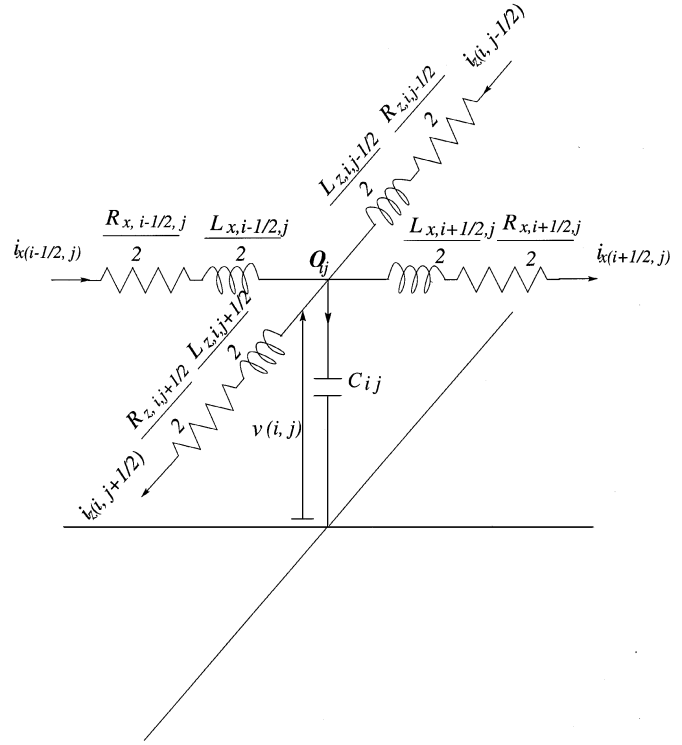


Fig. 4. Inhomogeneous case.

$$\begin{aligned} & [-\alpha_z \quad 1 + 2\alpha_z \quad -\alpha_z] \begin{bmatrix} v \Big|_{i,j-1}^{n+(1/2)} \\ v \Big|_{i,j}^{n+(1/2)} \\ v \Big|_{i,j+1}^{n+(1/2)} \end{bmatrix} \\ &= v \Big|_{i,j}^n - \frac{\Delta t}{4c\Delta l} [-1 \quad 1] \begin{bmatrix} i_x \Big|_{i-(1/2),j}^n \\ -i_x \Big|_{i+(1/2),j}^n \end{bmatrix} \\ &\quad - \frac{\Delta t(4l-r\Delta t)}{4c\Delta l(4l+r\Delta t)} [-1 \quad 1] \begin{bmatrix} i_z \Big|_{i,j-(1/2)}^n \\ -i_z \Big|_{i,j+(1/2)}^n \end{bmatrix} \end{aligned} \quad (18)$$

$$i_z \Big|_{i,j+(1/2)}^{n+(1/2)} = \frac{4l-r\Delta t}{4l+r\Delta t} i_z \Big|_{i,j+(1/2)}^n - \frac{2\Delta t}{\Delta z(4l+r\Delta t)} \cdot (v \Big|_{i,j+1}^{n+(1/2)} - v \Big|_{i,j}^{n+(1/2)}) \quad (19)$$

where

$$\alpha_z = \frac{(\Delta t)^2}{2c\Delta l\Delta z(4l+r\Delta t)}.$$

Equation (17) provides an explicit updating expression for i_x since it only depends on known values. Equation (18) provides an implicit updating expression for the voltage component v . It can be efficiently solved by LU decomposition in linear time since the matrix associated with this equation is tridiagonal. After solving each v , the i_z can be updated by plugging the values of v into (19).

Subiteration 2: Advance i_x, i_z , and v from time step $n+1/2$ to time step $n+1$

$$\begin{aligned} & i_z \Big|_{i,j+(1/2)}^{n+1} \\ &= \frac{4l-r\Delta t}{4l+r\Delta t} i_z \Big|_{i,j+(1/2)}^{n+(1/2)} - \frac{2\Delta t}{\Delta z(4l+r\Delta t)} \\ &\quad \cdot (v \Big|_{i,j+1}^{n+(1/2)} - v \Big|_{i,j}^{n+(1/2)}) \end{aligned} \quad (20)$$

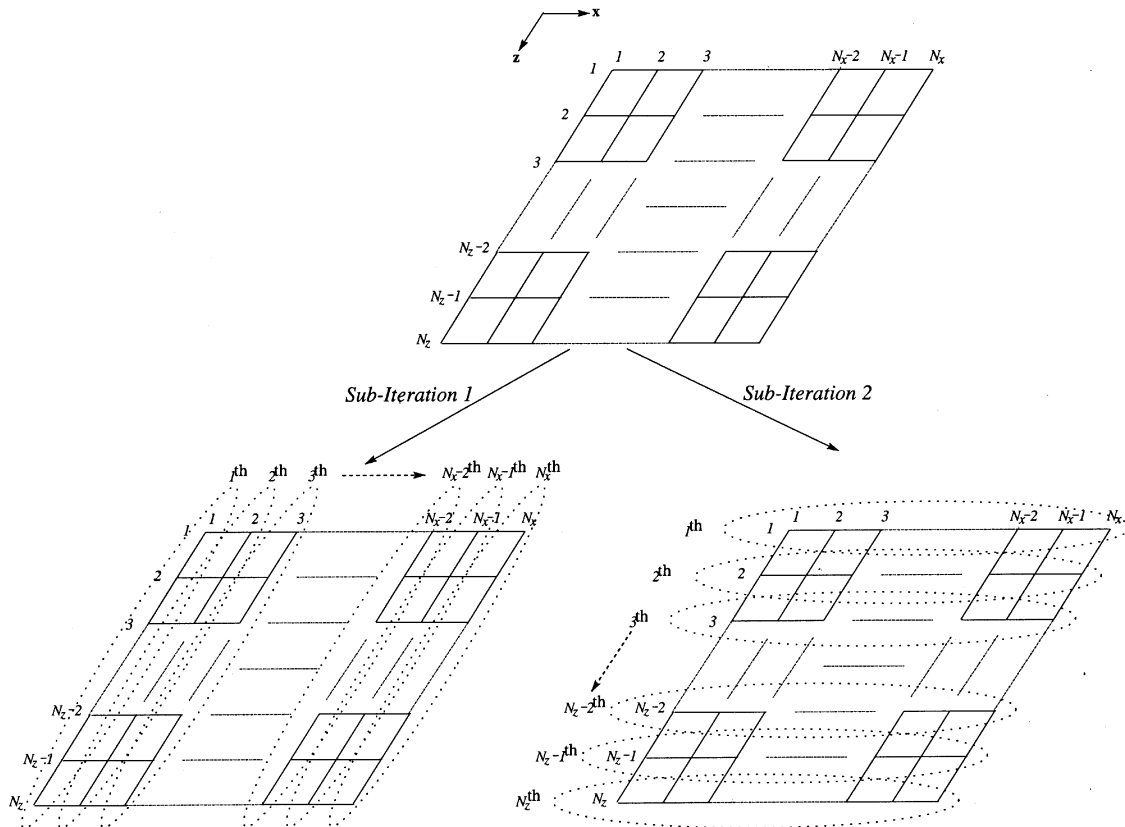


Fig. 5. Mesh points.

$$\begin{aligned}
 & \begin{bmatrix} -\alpha_x & 1 + 2\alpha_x & -\alpha_x \end{bmatrix} \begin{bmatrix} v|_{i-1,j}^{n+1} \\ v|_{i,j}^{n+1} \\ v|_{i+1,j}^{n+1} \end{bmatrix} \\
 &= v|_{i,j}^{n+(1/2)} - \frac{\Delta t(4l - r\Delta t)}{4c\Delta l(4l + r\Delta t)} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} i_x|_{i-(1/2),j}^{n+(1/2)} \\ -i_x|_{i+(1/2),j}^{n+(1/2)} \end{bmatrix} \\
 & \quad - \frac{\Delta t}{4c\Delta l} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} i_z|_{i,j-(1/2)}^{n+(1/2)} \\ -i_z|_{i,j+(1/2)}^{n+(1/2)} \end{bmatrix} \quad (21)
 \end{aligned}$$

$$\begin{aligned}
 & i_x|_{i+(1/2),j}^{n+1} \\
 &= \frac{4l - r\Delta t}{4l + r\Delta t} i_x|_{i+(1/2),j}^{n+(1/2)} - \frac{2\Delta t}{\Delta_x(4l + r\Delta t)} \\
 & \quad \cdot (v|_{i+1,j}^{n+1} - v|_{i,j}^{n+1}) \quad (22)
 \end{aligned}$$

where

$$\alpha_x = \frac{(\Delta t)^2}{2c\Delta l\Delta x(4l + r\Delta t)}.$$

Equation (20) provides an explicit updating expression for i_z because it only depends on known values. Equation (21) provides an implicit updating expression for the voltage component v . It can be efficiently solved by LU decomposition in linear time since the matrix associated with this equation is tridiagonal. Finally, the i_x can be updated by substituting the values of v into (22).

B. Stability Analysis

The general way of verifying the stability of a finite-difference algorithm is to put a sinusoidal traveling wave into the algorithm and make sure that the propagation gain of this traveling wave is no more than

TABLE I
TLM-ADI ALGORITHM

Input= $i_x|_i^n, i_z|_i^n, v|_i^n$ **Output=** $i_x|_i^{n+1}, i_z|_i^{n+1}, v|_i^{n+1}$

Begin

Sub-Iteration 1:

$$i_x|_{i+1/2,*}^{n+1/2} = \mathcal{D}_{xi} i_x|_{i+1/2,*}^n - \mathcal{D}_{vxi}(v|_{i+1,*}^n - v|_{i,*}^n) \quad \forall i = 1, \dots, N_x - 1$$

$$\Phi_i v|_{i,*}^{n+1/2} = v|_{i,*}^n - \mathcal{D}_{xvi}(i_x|_{i+1/2,*}^n - i_x|_{i-1/2,*}^n) - \mathcal{Z}_{zvi} i_z|_{i,*}^n \quad \forall i = 1, \dots, N_x$$

$$i_z|_{i,*}^{n+1/2} = \tilde{\mathcal{D}}_{zi} i_z|_{i,*}^n - \mathcal{Y}_{vzi} v|_{i,*}^{n+1/2} \quad \forall i = 1, \dots, N_x$$

Sub-Iteration 2:

$$i_z|_{*,j+1/2}^{n+1} = \mathcal{D}_{zj} i_z|_{*,j+1/2}^{n+1/2} - \mathcal{D}_{vzj}(v|_{*,j+1}^{n+1/2} - v|_{*,j}^{n+1/2}) \quad \forall j = 1, \dots, N_z - 1$$

$$\Psi_j v|_{*,j}^{n+1} = v|_{*,j}^{n+1/2} - \mathcal{D}_{zvj}(i_z|_{*,j+1/2}^{n+1/2} - i_z|_{*,j-1/2}^{n+1/2}) - \mathcal{Z}_{xvj} i_x|_{*,j}^{n+1/2} \quad \forall j = 1, \dots, N_z$$

$$i_x|_{*,j}^{n+1} = \tilde{\mathcal{D}}_{xj} i_x|_{*,j}^{n+1/2} - \mathcal{Y}_{vxj} v|_{*,j}^{n+1} \quad \forall j = 1, \dots, N_z$$

End

one for all frequencies. By applying the Von Neumann analysis [11] in the LC circuit ($r = 0$), we can analytically prove that the TLM-ADI method is unconditionally stable.

For each time step n , the instantaneous values of the i_x^n , i_z^n , and v_y^n in space across the grids are Fourier-transformed into the spatial spectral domain with respect to the x and z coordinates to provide a

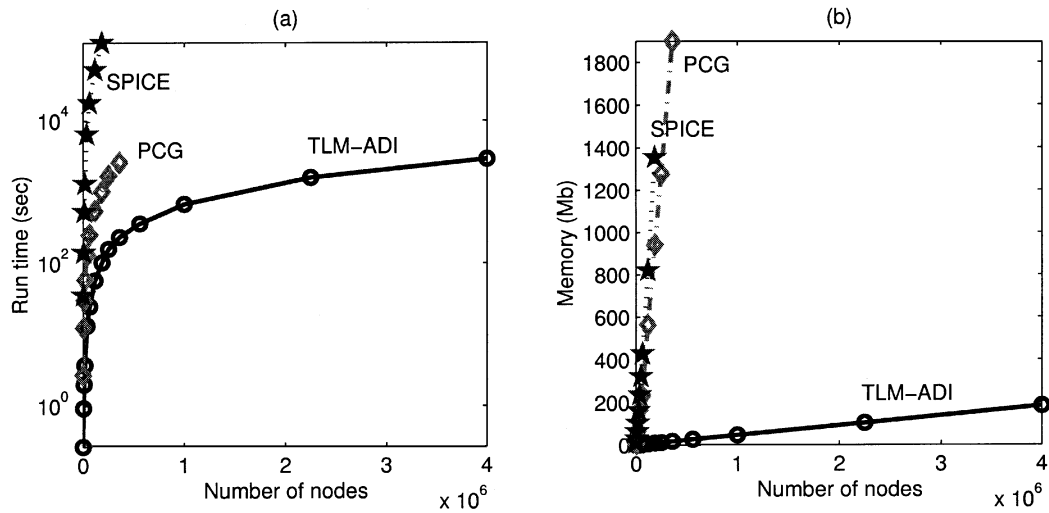


Fig. 6. Compare the (a) run time and (b) memory usages between TLM-ADI, PCG, and SPICE.

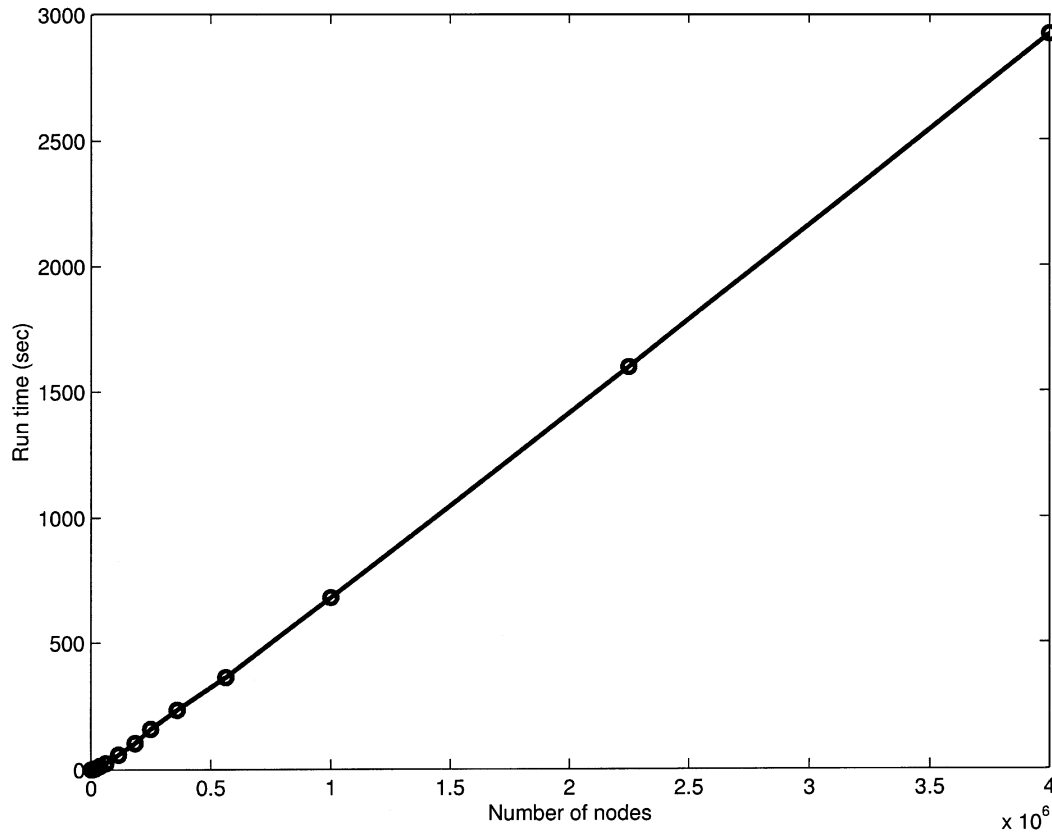


Fig. 7. Linear run time for TLM-ADI.

spectrum of spatial sinusoidal modes. By assuming the wavenumbers k_x and k_z along the x and z direction, respectively, these components can be represented as

$$i_x^n = I_{x0} e^{-j(k_x x + k_z z)} e^{jwn} \quad (23)$$

$$i_z^n = I_{z0} e^{-j(k_x x + k_z z)} e^{jwn} \quad (24)$$

$$v^n = V_0 e^{-j(k_x x + k_z z)} e^{jwn} \quad (25)$$

The composite vector in the spatial spectral domain at time-step n is denoted as

$$\mathbf{F}^n = \begin{bmatrix} i_x^n \\ i_z^n \\ v^n \end{bmatrix} \quad (26)$$

It can be shown that the Subiteration 1 [sees (17)–(19) with $r = 0$] can be written in the spatial spectral domain in matrix form as

$$\mathbf{F}^{n+1/2} = \mathbf{M}_1 \mathbf{F}^n \quad (27)$$

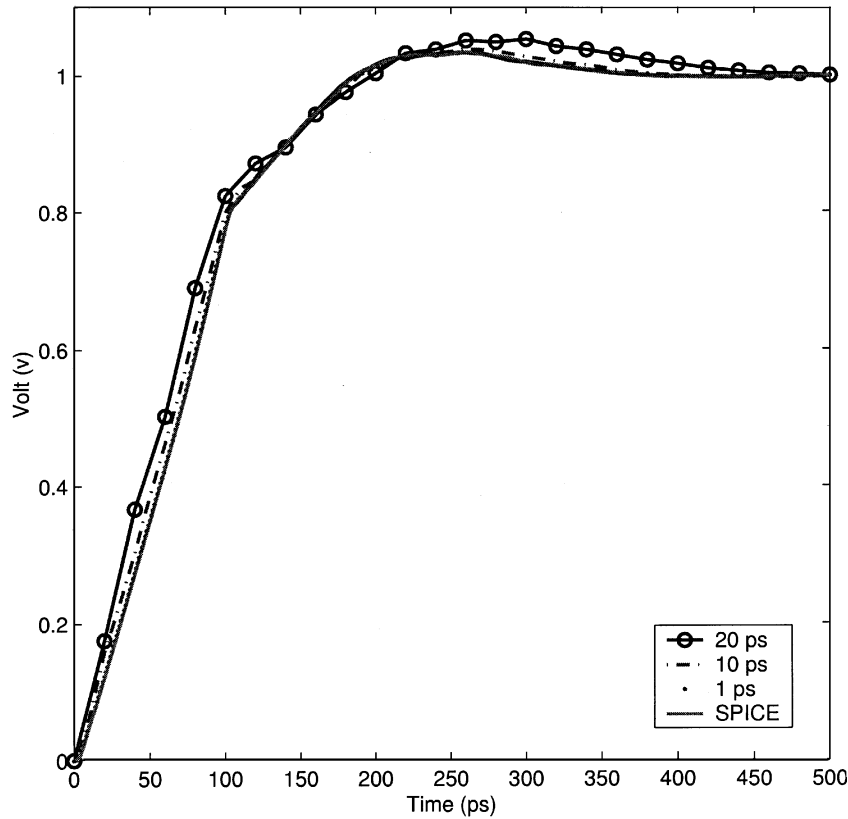


Fig. 8. DC transient response of TLM-ADI with different time steps.

where

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & jW_x \\ -\frac{\tilde{W}_x W_z}{Q_z} & \frac{1}{Q_z} & j\frac{W_z}{Q_z} \\ j\frac{\tilde{W}_x}{Q_z} & j\frac{\tilde{W}_z}{Q_z} & \frac{1}{Q_z} \end{bmatrix} \quad (28)$$

and

$$W_x = \frac{\Delta t}{l\Delta x} \sin \frac{k_x \Delta x}{2}; \quad \tilde{W}_x = \frac{\Delta t}{2c\Delta l} \sin \frac{k_x \Delta x}{2}$$

$$W_z = \frac{\Delta t}{l\Delta z} \sin \frac{k_z \Delta z}{2}; \quad \tilde{W}_z = \frac{\Delta t}{2c\Delta l} \sin \frac{k_z \Delta z}{2}$$

$$Q_x = 1 + W_x \tilde{W}_x; \quad Q_z = 1 + W_z \tilde{W}_z.$$

Similarly, it can be shown that Subiteration 2 [see (20)–(22) with $r = 0$] can be written in the spatial spectral domain in matrix form as

$$\mathbf{F}^{n+1} = \mathbf{M}_2 \mathbf{F}^{n+1/2} \quad (29)$$

where

$$\mathbf{M}_2 = \begin{bmatrix} \frac{1}{Q_x} & -\frac{W_x \tilde{W}_z}{Q_x} & j\frac{W_x}{Q_x} \\ 0 & 1 & jW_z \\ j\frac{\tilde{W}_x}{Q_x} & j\frac{\tilde{W}_z}{Q_x} & \frac{1}{Q_x} \end{bmatrix}. \quad (30)$$

Substituting (27) into (29), we get

$$\mathbf{F}^{n+1} = \mathbf{M}_2 \mathbf{M}_1 \mathbf{F}^n. \quad (31)$$

With the help of package MAPLE™, we can find the three eigenvalues of the composite matrix $\mathbf{M} = \mathbf{M}_2 \mathbf{M}_1$ as follows:

$$\lambda = 1, \quad \left(1 - W_x \tilde{W}_x - W_x \tilde{W}_x W_z \tilde{W}_z - W_z \tilde{W}_z \right. \\ \left. \pm 2 \sqrt{-W_x \tilde{W}_x - W_z \tilde{W}_z - W_x \tilde{W}_x W_z \tilde{W}_z} \right) \\ \left/ \left(1 + W_x \tilde{W}_x + W_z \tilde{W}_z + W_x \tilde{W}_x W_z \tilde{W}_z \right) \right.$$

By the detail analysis of the above eigenvalues, we are able to prove that the proposed algorithm is unconditionally stable in the following theorem.

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

Proof: The Von Neumann analysis [11] says that the iterative scheme is unconditionally stable if all the eigenvalues of the iterative matrix are less than or equal to one. Hence, we only need to prove all the eigenvalues are ones. The first eigenvalue is unity. For the rest two eigenvalues, we can easily see the arguments of the square roots in the numerators are all negative numbers. Hence, the square roots are imaginary numbers. By taking the magnitudes of the numerators, we find symbolically that they are exactly the same as denominators. So, the magnitudes of the eigenvalues are unity, regardless of the Δt . Therefore, we conclude that our TLM-ADI algorithm is unconditionally stable, and the Courant stability condition is removed.

C. Inhomogeneous Case and Linear Run Time

Generally, the parameters r , l , c , Δx , and Δz may have different values at different segments of the circuit. Hence, we extend the TLM-ADI method to handle more general situations, as illustrated in Fig. 4. The $R_{x,i\pm 1/2,j}$ and $L_{x,i\pm 1/2,j}$ are the equivalent resistances and inductances in x direction for a cell, $R_{z,i,j\pm 1/2}$ and $L_{z,i,j\pm 1/2}$

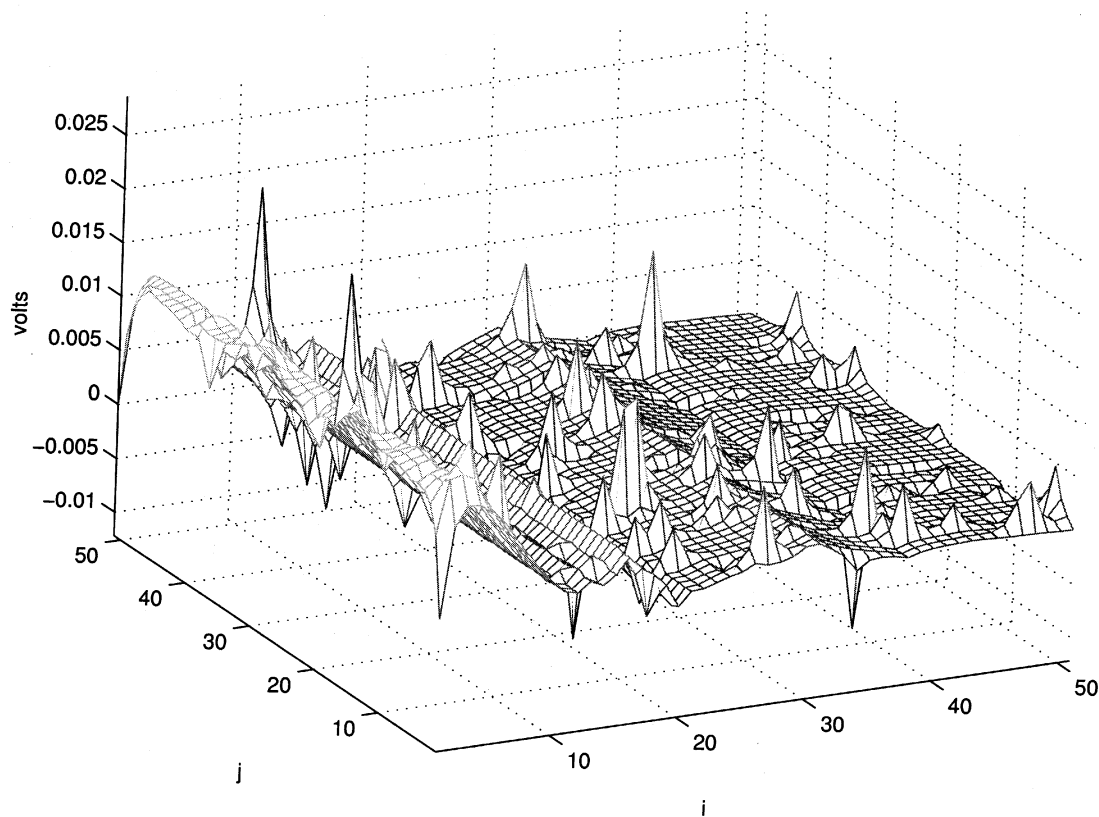


Fig. 9. A snapshot of the transient response with 1-ps time step.

are the equivalent resistances and inductances in z direction for a cell, and $C_{i,j}$ is the equivalent capacitance for a cell.

By applying the similar derivation of (17)–(22), we get the updating equations as follows.

For **Subiteration 1**, we divide the set of these N ($N = N_x \times N_z$) nodes by N_x subsets with each one containing N_z points at the z direction, as illustrated in Fig. 5

$$\mathbf{i}_x|_{i+(1/2),*}^{n+(1/2)} = \mathbf{D}_{xi}\mathbf{i}_x|_{i+(1/2),*}^n - \mathbf{D}_{vxi}(\mathbf{v}|_{i+1,*}^n - \mathbf{v}|_{i,*}^n) \quad \forall i = 1, \dots, N_x - 1 \quad (32)$$

$$\Phi_i \mathbf{v}|_{i,*}^{n+(1/2)} = \mathbf{v}|_{i,*}^n - \mathbf{D}_{xvi}(\mathbf{i}_x|_{i+(1/2),*}^n - \mathbf{i}_x|_{i-(1/2),*}^n) - \mathbf{Z}_{zvi}\mathbf{i}_z|_{i,*}^n \quad \forall i = 1, \dots, N_x \quad (33)$$

$$\mathbf{i}_z|_{i,*}^{n+(1/2)} = \tilde{\mathbf{D}}_{zi}\mathbf{i}_z|_{i,*}^n - \mathbf{Y}_{vzi}\mathbf{v}|_{i,*}^{n+(1/2)} \quad \forall i = 1, \dots, N_x \quad (34)$$

where \mathbf{D}_{xi} , \mathbf{D}_{vxi} and \mathbf{D}_{xvi} are $N_z \times N_z$ diagonal matrices, $\tilde{\mathbf{D}}_{zi}$ is a $(N_z - 1) \times (N_z - 1)$ diagonal matrix, \mathbf{Z}_{zvi} is a $N_z \times N_z$ lower 1 -banded matrix (only diagonal and the first lower subdiagonal terms are nonzeros), \mathbf{Y}_{vzi} is a $(N_z - 1) \times (N_z - 1)$ upper 1 -banded matrix (only diagonal and the first upper subdiagonal terms are nonzeros), Φ_i is a $N_z \times N_z$ tridiagonal matrix, $\mathbf{i}_x|_{i+(1/2),*}^n$ and $\mathbf{v}|_{i,*}^n$ are $N_z \times 1$ vectors ($n = 1/2, 1, 1 + 1/2, \dots$), and $\mathbf{i}_z|_{i,*}^n$ is a $(N_z - 1) \times 1$ vector ($n = 1/2, 1, 1 + 1/2, \dots$).

The computational load for $\mathbf{i}_x|_{i+(1/2),*}^{n+(1/2)}$ is $O(N_z)$ since it only depends on the known values. From (35), the coefficient matrix Φ_i asso-

ciated with updating $\mathbf{v}|_{i,*}^{n+(1/2)}$ is a tridiagonal matrix like (35). Therefore, the run time for updating $\mathbf{v}|_{i,*}^{n+(1/2)}$ is also $O(N_z)$

$$\Phi_i = \begin{bmatrix} \times & \times & 0 & \cdots & 0 \\ \times & \times & \times & \ddots & \vdots \\ 0 & \times & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \times & \times \\ 0 & \cdots & 0 & \times & \times \end{bmatrix} \quad (35)$$

The work load for $\mathbf{i}_z|_{i,*}^{n+(1/2)}$ is $O(N_z - 1)$ because it depends only on the known values after $\mathbf{v}|_{i,*}^{n+(1/2)}$ is updated. Therefore, the run time is $O(N_z)$ for each i .

Hence, the computational load for Subiteration 1 is $O(N)$.

For **Subiteration 2**, we divide the set of these N nodes by N_z subsets with each one containing N_x points at the x direction, as illustrated in Fig. 5

$$\mathbf{i}_z|_{*,j+(1/2)}^{n+1} = \mathcal{D}_{zj}\mathbf{i}_z|_{*,j+(1/2)}^{n+(1/2)} - \mathcal{D}_{zvz}(\mathbf{v}|_{*,j+1}^{n+(1/2)} - \mathbf{v}|_{*,j}^{n+(1/2)}) \quad \forall j = 1, \dots, N_z - 1 \quad (36)$$

$$\Psi_j \mathbf{v}|_{*,j}^{n+1} = \mathbf{v}|_{*,j}^{n+(1/2)} - \mathcal{D}_{zvj}(\mathbf{i}_z|_{*,j+(1/2)}^{n+(1/2)} - \mathbf{i}_z|_{*,j-(1/2)}^{n+(1/2)}) - \mathcal{Z}_{xvj}\mathbf{i}_x|_{*,j}^{n+(1/2)} \quad \forall j = 1, \dots, N_z \quad (37)$$

$$\mathbf{i}_x|_{*,j}^{n+1} = \tilde{\mathcal{D}}_{xj}\mathbf{i}_x|_{*,j}^{n+(1/2)} - \mathcal{Y}_{vxj}\mathbf{v}|_{*,j}^{n+1} \quad \forall j = 1, \dots, N_z \quad (38)$$

where \mathcal{D}_{zj} , \mathcal{D}_{zvz} , and \mathcal{D}_{zvj} are $N_x \times N_x$ diagonal matrices, $\tilde{\mathcal{D}}_{xj}$ is a $(N_x - 1) \times (N_x - 1)$ diagonal matrix, \mathcal{Z}_{xvj} is a $N_x \times N_x$ lower

l -banded matrix (only diagonal and the first lower subdiagonal terms are nonzeros), $\mathcal{Y}_{v_x j}$ is a $(N_x - 1) \times (N_x - 1)$ upper l -banded matrix (only diagonal and the first upper subdiagonal terms are nonzeros), Φ_i is a $N_x \times N_x$ tridiagonal matrix, $\mathbf{i}_x|_{*,j}^n$ and $\mathbf{v}|_{*,j}^n$ are $N_x \times 1$ column vectors ($n = 1/2, 1, 1 + 1/2, \dots$), and $\mathbf{i}_z|_{*,j+1/2}^n$ is a $(N_x - 1) \times 1$ vector ($n = 1/2, 1, 1 + 1/2, \dots$).

By the similar way, the computational load for Subiteration 2 is also $O(N)$.

Now, we can easily prove that the TLM-ADI algorithm has a linear run time at each time step.

Theorem 2: The run time of TLM-ADI algorithm is $O(N)$ at each time step, where N is the total number of nodes.

Proof: Two subiterations, 1 and 2, need to be performed for each time step. From the above discussion, we know their run times are both $O(N)$. Therefore, the total run time is $O(N)$.

At the end of this section we summarize our TLM-ADI algorithm in Table I.

IV. EXPERIMENTAL RESULTS

In this section, we present several numerical experiments. We implement the TLM-ADI method in C language, and perform on an Alpha workstation with Dual SLOTB 667 MHz Alpha 21 264 processors. For simplicity, the experiments are performed on the homogeneous case. We choose $r = 0.03 \Omega/\mu\text{m}$, $l = 1.26 \text{ pH}/\mu\text{m}$, $c = 0.024 \text{ fF}/\mu\text{m}$, and $\Delta x = \Delta z = 500 \mu\text{m}$ as the common parameters. Numerical results are carried out by using the TLM-ADI algorithm, the PCG algorithm [8], and the general circuit simulator SPICE. The software of PCG simulator which used the minimum degree recording is obtained from the authors of [8].

The run time and memory usages are shown in Fig. 6 with 1-ps time step and total 500 time steps. Fig. 6(a) and (b) show that the TLM-ADI method is not only about ten times faster than the PCG method, and over 1000 times faster than SPICE, but also extremely memory saving. In Figs. 6(b) and 7, we show that the memory requirement and run time for TLM-ADI are both linear with the total number of nodes. In Fig. 8, we examine the accuracy and unconditional stability of the TLM-ADI algorithm by simulating the dc transient response of a RLC circuit with 100 nodes, and 1-V dc voltage source excitation. The voltage waveform of TLM-ADI with 1-ps time step and the result from SPICE's at one node are overlapped as shown in Fig. 8. Fig. 8 also demonstrates the unconditional stability of TLM-ADI method. In this case, the Courant stability constraint is 1.9442 ps, while the time step of TLM-ADI is not limited by this stability constraint.

Finally, we use the TLM-ADI algorithm to simulate a 2601-node power grid in 1-ns clock period with time step as 1 ps. Fig. 9 shows the voltage distribution snapshot. Each node is attached a time-varying current source to model the drawn current.

V. CONCLUSION

An efficient TLM-ADI algorithm for transient power grid simulation is developed. Its unconditional stability and linear run time have been demonstrated. The numerical simulation also shows that the TLM-ADI algorithm not only speeds up orders of magnitude over the SPICE but also cuts down the memory requirement and the results agree very well with SPICE's.

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] 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.
- [6] L. W. Nagel, "SPICE2, A computer program to simulate semiconductor circuits," Univ. California-Berkeley, Tech. Rep. ERL-M520, May 1975.
- [7] 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.
- [8] 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.
- [9] C. Christopoulos, *The Transmission-Line Modeling Method TLM*: Oxford Univ. Press, 1995.
- [10] M. N. O. Sadiku, *Numerical Techniques in Electromagnetics*. New York: CRC Press, 1992.
- [11] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*: Wadsworth & Brooks/Cole, 1989.
- [12] A. Taflov and S. C. Hagness, *Computational Electrodynamics—The Finite-Difference Time-Domain Method*. New York: Artech, 2000.
- [13] W. K. Gwarek, "Analysis of arbitrarily shaped two-dimensional microwave circuits by finite-difference time-domain method," *IEEE Trans. Microwave Theory Techn.*, vol. 36, pp. 738–744, Apr. 1988.
- [14] C. P. Chen, N. Murugesan, T.-W. Lee, and S. C. Hagness, "FDTD-ADI: An unconditionally stable full wave Maxwell equation solver for VLSI modeling," in *Proc. ICCAD*, 2000, to be published.
- [15] T. Namiki, "A new FDTD algorithm based on alternating-direction implicit method," *IEEE Trans. Microwave Theory Techn.*, vol. 47, pp. 2003–2007, Oct. 1999.
- [16] F. Zheng, Z. Chen, and J. Zhang, "Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method," *IEEE Trans. Microwave Theory Techn.*, vol. 48, pp. 1500–1558, Sept. 2000.
- [17] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *J. Soc. Ind. Appl. Math.*, vol. 3, pp. 28–41, 1955.

Transition Time Modeling in Deep Submicron CMOS

P. Maurine, M. Rezzoug, N. Azemard, and D. Auvergne

Abstract—As generally recognized, the performance of a CMOS gate, such as propagation delay time or short circuit power dissipation, is strongly affected by the nonzero input signal transition time. This paper presents an analytical model of the transition time of CMOS structures. The authors first develop the model for inverters, considering Fast and Slow input signal conditions, over a large design range of input–output coupling capacitance and capacitive load. They then extend this model to more complex gates. The validity of the presented model is demonstrated through a comparison with HSPICE simulations on a 0.18 μm CMOS process.

Index Terms—Deep submicron, modeling, timing analysis.

Manuscript received February 19, 2001; revised July 11, 2001, October 30, 2001, and March 1, 2002. This paper was recommended by Editor-in-Chief K. Mayaram.

P. Maurine, N. Azemard, and D. Auvergne are with the Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, 34392 Montpellier, France.

M. Rezzoug is with Unilog, Grenoble, France.

Digital Object Identifier 10.1109/TCAD.2002.804088