

A Recovery Algorithm of Linear Complexity in the Time-Domain Layered Finite Element Reduction Recovery (LAFE-RR) Method for Large-Scale Electromagnetic Analysis of High-Speed ICs

Houle Gan and Dan Jiao, *Senior Member, IEEE*

Abstract—Time-domain layered finite element reduction recovery (LAFE-RR) method was recently developed for large-scale electromagnetic analysis of high-speed integrated circuits (ICs). This method is capable of analytically and rigorously reducing the system matrix of a 3-D multilayer circuit to that of a single-layer one regardless of the original problem size. In addition, the reduced system matrix preserves the sparsity of the original system matrix. In this paper, an efficient algorithm is proposed to recover the volume unknowns in the time-domain LAFE-RR method. This algorithm constitutes a direct solution of the matrix formed by volume unknowns in each layer. This direct solution possesses a linear complexity in both central processing unit (CPU) time and memory consumption. The cost of matrix inversion is negligible. The cost of matrix solution scales linearly with the matrix size. Numerical and experimental results have demonstrated the accuracy and efficiency of the proposed algorithm.

Index Terms—Electromagnetic analysis, finite-element methods, time domain analysis, very large scale integrated circuits.

I. INTRODUCTION

CIRCUIT THEORY has guided very large scale integrated (VLSI) design and analysis for more than three decades. As on-chip circuits have scaled into the nanometer regime, full-wave electromagnetics-based analysis has increasingly become essential for three major reasons. 1) Reduced feature sizes: at the 45 nm processing technology node and beyond, it becomes necessary to print features that are several times less than the wavelength of light (193 nm) being used in optical lithography. In this regime, the wave nature of light is manifest. This induces extreme proximity effects, which need to be comprehended and compensated for by optical proximity correction (OPC). OPC determines the photomask patterns that enable drawn layout features to be faithfully and accurately reproduced by optical lithography onto the wafer. It has emerged as a major bottleneck in achieving efficient turnaround time for integrated circuit (IC) data preparation and high-yield manufacturing. The enabling technology of accurate model-guided OPC is computational electromagnetics. 2) Increased clock frequency: currently the clock frequency of microprocessors is in the giga-

hertz regime. Since it is necessary to analyze the chip response to harmonics five times the clock frequency, it is expected that interconnects would have to be analyzed with certain electromagnetic effects incorporated at high frequencies. The importance of electromagnetic (EM) analysis at tens of gigahertz has been quantitatively demonstrated, via simulation and real silicon measurements, in [22] and [6]. 3) Increased level of integration: this calls for increasing levels of the integration of RF, analog, and digital circuits on the same chip, which leads often to undesirable coupling and sometimes to system failure. For instance, switching currents induced by logic circuits cause ringing in the power-supply rails and in the output driver circuitry. This, in turn, couples through the common substrate to corrupt sensitive analog signals on the same chip. Prevailing circuit-based signal integrity paradigms are reaching their limits of predictive accuracy when applied to high-frequency mixed-signal settings. To sustain the scaling and integration of digital, analog, mixed-signal, and RF circuitry, full-wave electromagnetics-based analysis is indispensable.

In addition to the full-wave electromagnetics-based analysis, very-large-scale analysis is also essential for the design of high-speed VLSI circuits. Take a full-chip interconnect network for power delivery as an example; it involves more than seven metal layers, a large number of nonuniform dielectric stacks, $\sim 10\,000\ \mu\text{m} \times 10\,000\ \mu\text{m}$ chip area, and millions of vias and wires. In addition, active power managements result in large processor current variations and fast transient droops and noises in the global power supply network, which demands very large-scale electromagnetic analysis. Already, the impact of noise due to die-package interaction, substrate coupling, etc., can be seen at all levels of a power delivery network: from chip to package to motherboard to the voltage regulator module. The move towards integrating thousand cores on a single chip will exacerbate the problems even more. Therefore, it is important to develop large-scale electromagnetic modeling and simulation methods.

In recent years, researchers in both VLSI circuits and electromagnetic fields have initiated the development of innovative computational EM techniques for on-chip problems [1]–[16]. These techniques can be categorized into two classes: partial differential equation (PDE)-based solvers and integral equation (IE)-based ones. In the former, the finite difference time domain (FDTD) method is a representative one. Two-dimensional FDTD approaches have been developed for the full-wave modeling of on-chip transmission lines. For 3-D structures, an

Manuscript received August 14, 2007; revised November 5, 2007. First published March 31, 2008; last published August 6, 2008 (projected). This work was supported in part by the Office of Naval Research under Award N00014-06-1-0716 and in part by a grant from Intel Corporation. This work was recommended for publication by Associate Editor J. Tan upon evaluation of the reviewers comments.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: djiao@purdue.edu).
Digital Object Identifier 10.1109/TADVP.2008.920648

FDTD solver is much more computationally expensive. In the latter, the partial element equivalent circuit (PEEC) method is a representative one [1], [2], [8], [14]. PEEC was first utilized to solve quasistatic problems. It was then extended to full-wave analysis. Recently, surface-based PEEC methods have also been developed which reduce significantly the number of unknowns involved in a volume-integral-equation-based PEEC method [2], [8]. Another surface integral equation based formulation was developed in [3]. A fast precorrected FFT scheme was formulated to accelerate the iterative solution of the dense matrix equation. However, the multilayered dielectric has not yet been included. Integral equation based methods have also been developed in [7], [9], [10], [15], and [16] for the application to on-chip problems.

Ultra-large-scale IC design results in numerical problems of ultra large scale, requiring billions of parameters to describe them accurately. In general, to solve a matrix equation of N degrees-of-freedom, the optimal computational complexity one can hope for is linear complexity $O(N)$. For the VLSI circuit problem, however, even $O(N)$ is prohibitively high since N is too large. In [4]–[6], [12], [13], efforts have been made to reduce the problem size from $O(N)$ to $O(M)$ with M much less than N . In [12] and [13], a time-domain layered finite element reduction recovery (LAFE-RR) method was developed for high-frequency modeling and simulation of large-scale on-chip circuits. This method rigorously reduces the matrix of the original multilayer system to that of a single-layer one irrespective of the original problem size. More importantly, the matrix reduction is achieved analytically, and hence the CPU and memory overheads are minimal. In addition, the reduction preserves the sparsity of the original matrix. The method applies to any arbitrarily shaped multilayer structure.

The recovery of volume unknowns in the time-domain LAFE-RR method involves the solution of a matrix formed by volume unknowns in a single layer. Although this matrix is sparse, its computation can be expensive when the matrix size is large. An advanced sparse solver such as the multifrontal method [17], [18] requires much more than $O(M^2)$ operations and $O(M)$ storage to solve this matrix when M is large, with M being the matrix size. In this work, the complexity will be reduced to $O(M)$ in both memory and CPU time. This is the optimal complexity one can hope for to solve problems with M parameters.

The remainder of this paper is organized as follows. In Section II, the time-domain LAFE-RR method is outlined. In Section III, the proposed recovery algorithm of linear complexity is presented. In Section IV, numerical and experimental results are given to demonstrate the accuracy and efficiency of the proposed algorithm. Section V relates to our conclusions.

In the description that follows, the stack-growth direction is defined as x , the number of volume unknowns in a single layer is denoted by M , and the number of layers is denoted by L .

II. BRIEF REVIEW OF THE TIME-DOMAIN LAFE-RR METHOD

The electric field \mathbf{E} inside a 3-D integrated circuit satisfies the second-order vector wave equation

$$\nabla \times [\mu_r^{-1} \nabla \times \mathbf{E}(\mathbf{r}, t)] + \mu_0 \varepsilon \partial_t^2 \mathbf{E}(\mathbf{r}, t) + \mu_0 \sigma \partial_t \mathbf{E}(\mathbf{r}, t) = -\mu_0 \partial_t \mathbf{J}(\mathbf{r}, t) \quad \text{in } V \quad (1)$$

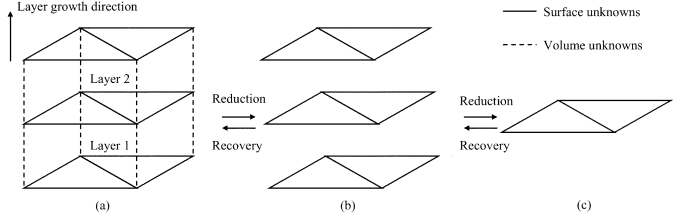


Fig. 1. Illustration of the LAFE-RR process. (a) Three-dimensional layered system, (b) 2-D layered system, and (c) single-layer system.

subject to certain boundary conditions. In (1), $\mu_r, \mu_0, \varepsilon, \sigma$ are relative permeability, free-space permeability, permittivity, and conductivity, respectively; \mathbf{J} is the current source; V is the computational domain that encloses the circuit. A time-domain finite-element solution of (1) and its boundary condition results in a system of ordinary differential equations [19]

$$\mathbf{T} \frac{d^2 u}{dt^2} + \mathbf{R} \frac{du}{dt} + \mathbf{S} u + w = j \quad (2)$$

in which \mathbf{T}, \mathbf{R} , and \mathbf{S} are square matrices, and u, w , and j are column vectors. Their elements are given by

$$\begin{aligned} \mathbf{T}_{ij} &= \mu_0 \varepsilon \langle \mathbf{N}_i, \mathbf{N}_j \rangle_V & \mathbf{R}_{ij} &= \mu_0 \sigma \langle \mathbf{N}_i, \mathbf{N}_j \rangle_V \\ \mathbf{S}_{ij} &= \mu_r^{-1} \langle \nabla \times \mathbf{N}_i, \nabla \times \mathbf{N}_j \rangle_V \\ w_i &= -\langle \mathbf{N}_i, P \rangle_S & j_i &= -\mu_0 \langle \mathbf{N}_i, \partial_t \mathbf{J} \rangle_V \end{aligned} \quad (3)$$

where \mathbf{N}_i are the vector bases used to expand the unknown electric field \mathbf{E} , P is an operator associated with the absorbing boundary condition, and $\langle \cdot, \cdot \rangle_V$ and $\langle \cdot, \cdot \rangle_S$ denote volume and surface integration, respectively.

When the problem size is large, it is difficult to solve matrix equation (2). The time-domain LAFE-RR method [12], [13] was developed to overcome this problem. In this method, the computational domain is discretized into prism elements as shown in Fig. 1(a). The prism elements extrude along the layer growth direction, which can be the x -direction, i.e., the stack-growth direction, or the y - and z -direction. The unknowns are ordered layer by layer. In each layer, the unknowns are divided into surface and volume ones. As shown in Fig. 1(a), the unknowns associated with the solid edges are surface unknowns; and the unknowns associated with the dashed edges are volume unknowns. The 3-D layered system matrix is then analytically reduced to a 2-D layered one, as shown in Fig. 1(b). The 2-D layered system matrix is then analytically reduced to a single-layer one, as shown in Fig. 1(c). The reduced single-layer matrix preserves the same sparse pattern as that of a single-layer matrix in the original system matrix, which enables an efficient computation. The method permits different layout structures in different layers. It is applicable to both triangular prism and brick element based discretization.

At each time step, once the surface unknowns in a single layer are known, the surface unknowns and the volume unknowns in other layers can be recovered. For example, the volume unknowns in layer l, x_{Vl} , are recovered by solving the following matrix equation:

$$\mathbf{P}_{Vlx_{Vl}} = b_{Vl} \quad l = 1, 2, \dots, L \quad (4)$$

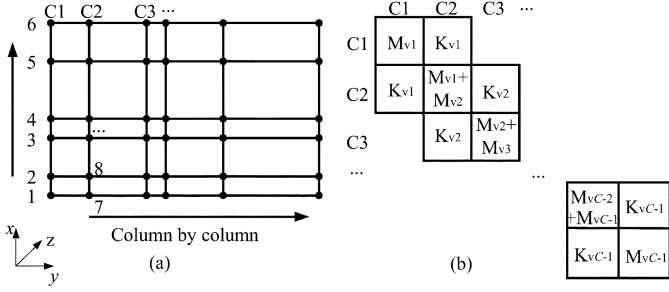


Fig. 2. Illustration of the mesh, ordering, and matrix pattern. (a) Mesh and ordering and (b) matrix pattern.

where b_{Vl} is the right-hand-side corresponding to the volume unknowns in layer l [12], which can be calculated from

$$b_{Vl} = (2\mathbf{T} - \Delta t^2 \mathbf{S} - \Delta t \mathbf{R}) u^n + [\Delta t \mathbf{R} - \mathbf{T}] u^{n-1} - \Delta t^2 w^n + j^n \quad (4')$$

in which superscript n denotes the n th time step, and Δt is the time step. Matrix \mathbf{P}_{Vl} is assembled from its elemental contribution as the following:

$$\mathbf{P}_{Vl,ij}^e = h_l (\mu_0 \varepsilon_l^e) \int_{\Omega^e} \xi_i \cdot \xi_j d\Omega \quad l = 1, 2, \dots, L \quad (5)$$

in which h_l is the thickness of layer l , ε_l^e is the permittivity in layer l and element e , ξ_i ($i = 1, 2, 3$) is the area coordinate (also known as the node basis function [20]), and Ω^e is the support of the triangular element e . \mathbf{P}_{Vl} is sparse. However, its computation can be very expensive when its dimension M is large. To solve this problem, an algorithm of $O(M)$ complexity is developed in this work.

III. RECOVERY ALGORITHM OF LINEAR COMPLEXITY

To recover the volume unknowns efficiently, \mathbf{P}_{Vl} is structured to be a block tridiagonal matrix through a proper ordering of unknowns. To make the matrix even better structured, the computational domain is discretized into brick elements. Since most of the on-chip structures are Manhattan-type structures, the geometry modeling capability would not be lost by using brick elements. Assuming volume unknowns are assigned along the z direction. Fig. 2(a) depicts an x - y cross-sectional view of these volume unknowns. Each volume unknown becomes a dot in this view.

The cross section is meshed into C columns along y and N_x rows along x . If the unknowns are ordered column by column from left to right, and in each column, the unknowns are ordered row by row from bottom to top. A matrix \mathbf{P}_{Vl} having a pattern shown in Fig. 2(b) will be generated. It is a block tridiagonal matrix. Furthermore, each diagonal and off-diagonal block is a tridiagonal matrix. Moreover, the matrices \mathbf{M}_{Vi} ($i = 1, 2, \dots, C-1$) are linearly proportional to each other, \mathbf{K}_{Vi} ($i = 1, 2, \dots, C-1$) are linearly proportional to each other, \mathbf{M}_{Vi} and \mathbf{K}_{Vi} are also linearly proportional to each other.

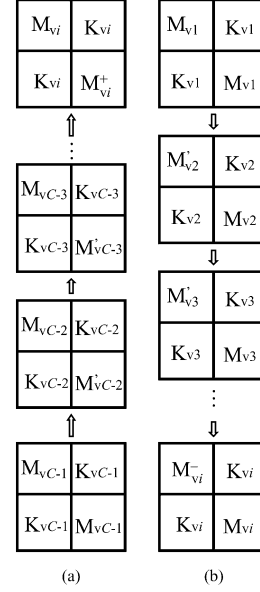


Fig. 3. (a) Bottom-up elimination procedure. (b) Top-down elimination procedure.

This is because \mathbf{M}_{Vi} and \mathbf{K}_{Vi} are assembled from their elemental contributions as

$$\mathbf{M}_{Vi}^e = \mu_0 \varepsilon_i^e \frac{l_x^e l_y^e l_z^e}{36} \begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}, \quad i = 1, 2, \dots, C-1,$$

$$\mathbf{K}_{Vi}^e = \mu_0 \varepsilon_i^e \frac{l_x^e l_y^e l_z^e}{36} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad i = 1, 2, \dots, C-1 \quad (6)$$

in which ε_i^e is the permittivity in segment i (the region formed between columns i and $i+1$) and element e , l_x^e , l_y^e , and l_z^e are the height, width, and thickness of the segment i and element e , respectively. Since x is the stack-growth direction, each segment shares the same permittivity configuration, l_x distribution, and l_z . As a result, \mathbf{M}_{Vi} and \mathbf{K}_{Vi} change with l_y only, and also in a linear fashion. Hence, if \mathbf{M}_{V1} is taken as the reference, \mathbf{M}_{Vi} and \mathbf{K}_{Vi} ($i = 1, 2, \dots, C-1$) can be obtained as follows:

$$\mathbf{M}_{Vi} = \frac{l_{yi}}{l_{y1}} \mathbf{M}_{V1} \quad (i = 2, 3, \dots, C-1)$$

$$\mathbf{K}_{Vi} = \frac{l_{yi}}{2l_{y1}} \mathbf{M}_{V1} \quad (i = 1, 2, \dots, C-1). \quad (7)$$

Next, we propose a scheme to analytically reduce \mathbf{P}_{Vl} that involves $C-1$ segments to a matrix that involves only a single segment. This scheme is analogous to that developed in [13], which reduces a 2-D layered system matrix to a single-layer one. Assuming the segment to be reduced to is i , a top-down-bottom-up procedure is performed to eliminate the other segments and project their contributions to segment i . For the bottom-up elimination, as shown in Fig. 3(a), the finite-element submatrix in the last segment, i.e., the $(C-1)$ th segment, is first formed. It is then eliminated with its contribution projected to the $(C-2)$ th segment. Next, the modified $(C-2)$ th segment is processed, and its contribution is projected to the $(C-3)$ th segment. This procedure is continued until the i th segment is reached. From bottom to top, Fig. 3 lists the submatrix in the $(C-1)$ th segment, the modified submatrix in

TABLE I
ACCURACY COMPARISON

Entries in x	x_7	x_{107}	x_{55300}	x_{543210}
Original scheme	2.110793	2.558760	1.462193	1.101502
Proposed algorithm	2.110793	2.558760	1.462193	1.101502

TABLE II
PERFORMANCE COMPARISON

	Original Scheme	This Algorithm
CPU time for the matrix factorization	90.236 s (Factorization of \mathbf{P}_V in (4))	<0.001 s (Inverse of \mathbf{M}_V in (14))
Memory cost for the matrix factorization	460 MB	1 KB
Matrix solution time	0.75 s	0.031 s

the $(C - 2)$ th segment, \dots , and the modified submatrix in the i th segment. Mathematically, it can be represented as

$$\begin{aligned} \mathbf{M}'_{V,C-2} &= \mathbf{M}_{V,C-2} + \mathbf{M}_{V,C-1} - \mathbf{K}_{V,C-1} \mathbf{M}'^{-1}_{V,C-1} \mathbf{K}_{V,C-1} \\ \mathbf{M}'_{V,C-3} &= \mathbf{M}_{V,C-3} + \mathbf{M}_{V,C-2} - \mathbf{K}_{V,C-2} \mathbf{M}'^{-1}_{V,C-2} \mathbf{K}_{V,C-2} \\ &\dots \\ \mathbf{M}'_{V,i} &= \mathbf{M}_{V,i} + \mathbf{M}_{V,i+1} - \mathbf{K}_{V,i+1} \mathbf{M}'^{-1}_{V,i+1} \mathbf{M}_{V,i+1}. \end{aligned} \quad (8)$$

The top-down elimination is depicted in Fig. 3(b). Mathematically, it can be written as

$$\begin{aligned} \mathbf{M}'_{V,2} &= \mathbf{M}_{V,1} + \mathbf{M}_{V,2} - \mathbf{K}_{V,1} \mathbf{M}'^{-1}_{V,1} \mathbf{K}_{V,1} \\ \mathbf{M}'_{V,3} &= \mathbf{M}_{V,2} + \mathbf{M}_{V,3} - \mathbf{K}_{V,2} \mathbf{M}'^{-1}_{V,2} \mathbf{K}_{V,2} \\ &\dots \\ \mathbf{M}'_{V,i} &= \mathbf{M}_{V,i-1} + \mathbf{M}_{V,i} - \mathbf{K}_{V,i-1} \mathbf{M}'^{-1}_{V,i-1} \mathbf{M}_{V,i-1}. \end{aligned} \quad (9)$$

As a result, matrix \mathbf{P}_{Vl} is reduced to a matrix of a single segment (segment i) as

$$\begin{pmatrix} \mathbf{M}'_{V,i} & \mathbf{K}_{V,i} \\ \mathbf{K}_{V,i} & \mathbf{M}'_{V,i} \end{pmatrix} \begin{pmatrix} x_{vi} \\ x_{v,i+1} \end{pmatrix} = \begin{pmatrix} b'_{vi} \\ b'_{v,i+1} \end{pmatrix} \quad (10)$$

in which x_{vi} are the volume unknowns residing on column i , and $x_{v,i+1}$ are the volume unknowns residing on column $i + 1$. In (10), $\mathbf{M}'_{V,i}$ carries all the contribution from i -segments (the segments left to segment i) to segment i , and matrix $\mathbf{M}'_{V,i}$ carries the contribution from $i+$ segments (the segments right to segment i) to segment i .

As shown in (7), all the $\mathbf{M}_{V,i}$ and $\mathbf{K}_{V,i}$ matrices are linearly proportional to each other. Therefore, (8) and (9) do not involve any computation. If $\mathbf{M}_{V,1}$ is taken as the reference, then $\mathbf{M}'_{V,i}$ and $\mathbf{M}'_{V,i}$ are simply $\mathbf{M}_{V,1}$ scaled by a certain coefficient made of l_{yi} ($i = 1, 2, \dots, C - 1$), which can be calculated by using (7)–(9).

The right-hand side of (10) is also obtained by a top-down-bottom-up procedure from the original right-hand side of (4).

For example, $b'_{V,i+1}$ can be obtained from a bottom-up recursive procedure as

$$\begin{aligned} b'_{V,C-1} &= b_{V,C-1} - \mathbf{K}_{V,C-1} \mathbf{M}'^{-1}_{V,C-1} b_{V,C} \\ b'_{V,C-2} &= b_{V,C-2} - \mathbf{K}_{V,C-2} \mathbf{M}'^{-1}_{V,C-2} b'_{V,C-1} \\ &\dots \\ b'_{V,i+1} &= b_{V,i+1} - \mathbf{K}_{V,i+1} \mathbf{M}'^{-1}_{V,i+1} b'_{V,i+2}. \end{aligned} \quad (11)$$

$b'_{V,i}$ can be obtained from a top-down recursive procedure as

$$\begin{aligned} b'_{V,2} &= b_{V,2} - \mathbf{K}_{V,1} \mathbf{M}'^{-1}_{V,1} b_{V,1} \\ b'_{V,3} &= b_{V,3} - \mathbf{K}_{V,2} \mathbf{M}'^{-1}_{V,2} b'_{V,2} \\ &\dots \\ b'_{V,i} &= b_{V,i} - \mathbf{K}_{V,i-1} \mathbf{M}'^{-1}_{V,i-1} b'_{V,i-1}. \end{aligned} \quad (12)$$

Equation (10) can be further analytically reduced to a matrix equation involving a single column as

$$\mathbf{M}''_{V,i} x_{V,i} = b''_{V,i} \quad (13)$$

where

$$\begin{aligned} \mathbf{M}''_{V,i} &= \mathbf{M}'_{V,i} - \mathbf{K}_{V,i} (\mathbf{M}'_{V,i})^{-1} \mathbf{K}_{V,i} \\ b''_{V,i} &= b'_{V,i} - \mathbf{K}_{V,i} \mathbf{M}'_{V,i} b'_{V,i+1}. \end{aligned}$$

There is no need to compute matrix inverse and matrix-matrix multiplication in (13) because $\mathbf{M}'_{V,i}$ and $\mathbf{K}_{V,i}$ are linearly proportional to each other. Matrix $\mathbf{M}'_{V,i}$ is hence again $\mathbf{M}_{V,1}$ scaled by a certain coefficient. Therefore, solving matrix (13) is equivalent to solving

$$\mathbf{M}_{V,1} x_{V,i} = f_{V,i} \quad (14)$$

in which $f_{V,i}$ is a corresponding scaling of $b''_{V,i}$.

$\mathbf{M}_{V,1}$ is a symmetric tridiagonal matrix. Its dimension is N_x . The inverse of a symmetric tridiagonal matrix belongs to the class of semiseparable matrices. There exist two sequences

$\{u_i\}, \{v_i\}, i = 1, 2, \dots, N_x$ [21] such that \mathbf{M}_{V1}^{-1} can be written as

$$\mathbf{M}_{V1}^{-1} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_{N_x} \\ u_1 v_2 & u_2 v_2 & \cdots & u_2 v_{N_x} \\ \vdots & \vdots & \ddots & \vdots \\ u_1 v_{N_x} & u_2 v_{N_x} & \cdots & u_{N_x} v_{N_x} \end{bmatrix}. \quad (15)$$

Denoting \mathbf{M}_{V1} as $\text{tri}(a_{1:N_x}, -b_{1:N_x-1})$, the sequences $\{u_i\}, \{v_i\}, i = 1, 2, \dots, N_x$ can be generated in $O(N_x)$ operations as follows:

$$\begin{aligned} d_{N_x} &= a_{N_x}, d_i = a_i - \frac{b_{i+1}^2}{d_{i+1}}, \quad i = N_x - 1, \dots, 1 \\ v_1 &= \frac{1}{d_1}, v_i = v_{i-1} \frac{b_i}{d_i}, \quad i = 2, \dots, N_x \\ \chi_1 &= a_1, \chi_i = a_i - \frac{b_i^2}{\chi_{i-1}}, \quad i = 2, \dots, N_x \\ u_{N_x} &= \frac{1}{\chi_{N_x} v_{N_x}}, u_{N_x-i} = \frac{b_{N_x-i+1}}{\chi_{N_x-i}} u_{N_x-i+1}, \\ & \quad i = 1, \dots, N_x - 1. \end{aligned} \quad (16)$$

Because the number of stacks is fixed at each processing technology node, N_x is a constant. Hence, the matrix inversion has a constant cost $O(N_x)$ in CPU time and memory (the storage of $\{u_i\}, \{v_i\}, i = 1, 2, \dots, N_x$). Furthermore, N_x is much smaller than M . For example, a typical N_x is 20. Therefore, the cost of matrix inversion is negligible.

The remaining task is to reduce the complexity of computing $\mathbf{M}_{V1}^{-1} f_{Vi}$. As shown in (15), the inverse of \mathbf{M}_{V1} is dense, which consists of N_x^2 parameters. Apparently, one needs $O(N_x^2)$ operations to obtain $\mathbf{M}_{V1}^{-1} f_{Vi}$. In fact, it can be obtained in linear complexity, $O(N_x)$. This is because of the compact representation of \mathbf{M}_{V1}^{-1} in $2N_x$ numbers containing in $\{u_i\}$ and $\{v_i\}$, which leads to an efficient computation of $\mathbf{M}_{V1}^{-1} f_{Vi}$ in linear complexity. Assuming f_{Vi} to be $\{f_1, f_2, \dots, f_{N_x}\}^T$, $\mathbf{M}_{V1}^{-1} f_{Vi}$ is evaluated as follows:

$$\begin{aligned} & u_1 \left(v_1 f_1 + \underbrace{v_2 f_2 + v_3 f_3 + v_4 f_4 + \cdots + v_{N_x} f_{N_x}} \right) \\ & \cdots \quad u_2 \left(v_2 f_2 + \underbrace{v_3 f_3 + v_4 f_4 + \cdots + v_{N_x} f_{N_x}} \right) \\ & \cdots \quad u_3 \left(v_3 f_3 + \underbrace{v_4 f_4 + \cdots + v_{N_x} f_{N_x}} \right) \\ & \cdots \quad \cdots \\ & \cdots \quad \cdots \quad \cdots \quad u_{N_x} v_{N_x} f_{N_x}. \end{aligned} \quad (17)$$

The underlying terms are those that do not need to be computed, as they can be reused from the previous step if one starts from the last row. As a result, for any row i ($i = 1, 2, \dots, N_x$), there is only one multiplication $v_i f_i$ that needs to be calculated together with one summation with the underlying term and one

multiplication with u_i . The same is true for the lower triangular part. Hence, the cost of $\mathbf{M}_{V1}^{-1} f_{Vi}$ scales as $O(N_x)$, with the constant multiplier in front of N_x equal to 7.

Once the unknowns on a single column are solved, the volume unknowns on other columns can be obtained recursively as follows:

$$\begin{aligned} x_{Vl} &= \mathbf{M}'_{Vl}^{-1} [b'_{Vl} - \mathbf{K}_{Vl} x_{V,l+1}], \quad l = i - 1, \dots, 2, 1 \\ x_{V,l+1} &= \mathbf{M}'_{Vl}^{-1} [b'_{V,l+1} - \mathbf{K}_{Vl} x_{Vl}], \quad l = i + 1, 2, \dots, C - 2 \\ x_{V,l+1} &= \mathbf{M}'_{Vl}^{-1} [b_{V,l+1} - \mathbf{K}_{Vl} x_{Vl}], \quad l = C - 1. \end{aligned} \quad (18)$$

Once again, there is no need to compute $\mathbf{M}'_{Vl}^{-1} \mathbf{K}_{Vl}$ as the two matrices are linearly proportional to each other. Since \mathbf{M}'_{Vl} is \mathbf{M}_{V1} scaled by a coefficient as can be seen from (7)–(9). The inverse calculated in (15) can be reused here. Applying the inverse of \mathbf{M}'_{Vl} to any vector can be performed in linear complexity as done in (17). Hence, the volume unknowns x_{Vl} ($l = 1, 2, \dots, C, l \neq i, l \neq i + 1$) in (18) are obtained in linear complexity. As a result, the cost of recovering all the volume unknowns, i.e., solving (4), scales with the number of unknowns linearly.

IV. NUMERICAL RESULTS AND EXPERIMENTAL VALIDATION

To demonstrate the accuracy of the proposed algorithm, a test-chip interconnect structure [22] was simulated. The structure was fabricated using conventional silicon processing technology. It involves three metal layers, and 13 inhomogeneous dielectric stacks. It is 2000 μm long. The structure was discretized into brick elements, resulting in 700 000 volume unknowns in a single layer. The solution of the volume unknowns recovered from the proposed algorithm was compared with those recovered from the original scheme [12], [13] that was based on a multifrontal method [17]. Table I lists the value of some entries arbitrarily selected from the unknown vector. Clearly, the accuracy of the proposed algorithm is validated.

In Table II, the performance of the proposed algorithm is compared against that of the original scheme. As shown in Table II, the CPU time and memory cost for the matrix factorization is negligible in the proposed algorithm. The matrix solution time, i.e., the CPU cost of obtaining M parameters, is 24 times faster.

In Fig. 4, the time domain waveforms of the test-chip interconnect are plotted. The current source is a derivative of a Gaussian pulse, as shown in Fig. 4(a). It is launched at the near end of the center wire of the interconnect structure. Fig. 4(b) and (c) plots the voltages sampled at the near- and far-end of this wire, respectively. An inductance effect can be clearly observed.

A Fourier transform is then performed, and the S -parameters are extracted from the time-domain waveforms. As can be seen from Fig. 5, the S -parameters simulated by the proposed algorithm are in excellent agreement with those generated by the original scheme. In addition, they match well with the measured data.

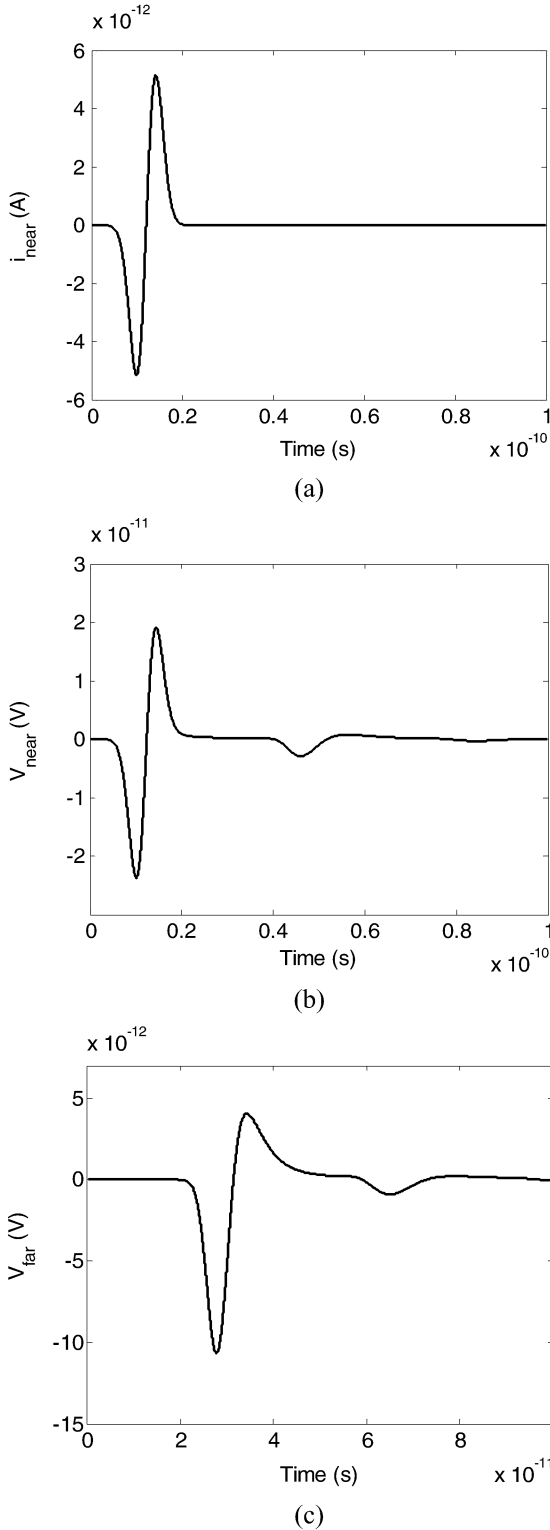


Fig. 4. Time-domain waveforms of a test-chip interconnect. (a) Current source launched at the near end. (b) Voltage sampled at the near end. (c) Voltage sampled at the far end.

V. CONCLUSION

In this paper, an efficient algorithm was developed to recover the volume unknowns in the time-domain LAPE-RR method. This algorithm constitutes a direct solution of the matrix formed by volume unknowns. It possesses the optimal computational complexity one can hope for to solve the volume-unknown-

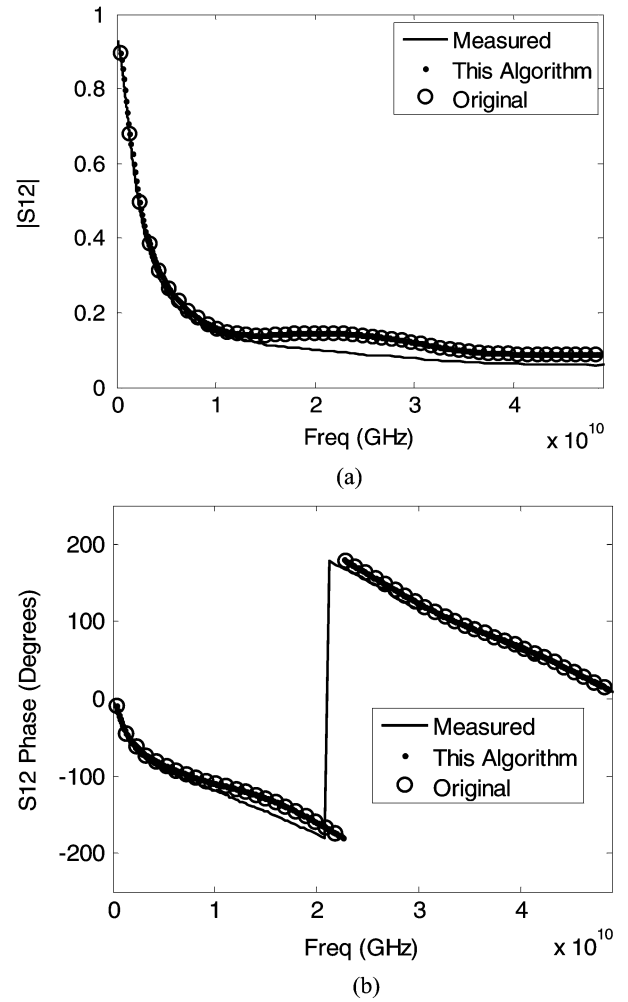


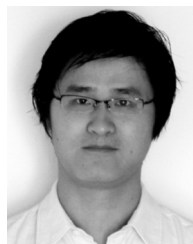
Fig. 5. S -parameters of a test-chip interconnect simulated by the proposed algorithm in comparison with the measured data and the S -parameters generated by the original scheme.

based matrix. The cost of matrix inversion is a constant: $O(N_x)$, which is negligible because N_x is a constant that is much smaller than M . The cost of matrix solution scales linearly with the number of unknowns in both CPU time and memory consumption. In addition, the constant multiplier in front of the number of unknowns is small (less than 10). Numerical and experimental results have demonstrated its accuracy and efficiency.

REFERENCES

- [1] P. J. Restle, A. E. Ruehli, S. G. Walker, and G. Papadopoulos, "Full-wave PEEC time-domain method for the modeling of on-chip interconnects," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 7, pp. 877–886, Jul. 2001.
- [2] A. Rong, A. C. Cangellaris, and L. Dong, "Comprehensive broadband electromagnetic modeling of on-chip interconnects with a surface discretization-based generalized PEEC model," in *Proc. IEEE 12th Topical Meeting Electr. Performance Electron. Packaging (EPEP)*, 2003, pp. 367–370.
- [3] Z. H. Zhu, B. Song, and J. K. White, "Algorithms in fastimp: A fast and wideband impedance extraction program for complicated 3-D geometries," *IEEE Trans. Comput.-Aided Design*, vol. 24, no. 7, pp. 981–998, Jul. 2005.
- [4] D. Jiao, S. Chakravarty, and C. Dai, "A layered finite element method for high-frequency modeling of large-scale three-dimensional on-chip interconnect structures," in *Proc. IEEE 15th Topical Meeting Electr. Performance Electron. Packaging (EPEP)*, 2006.

- [5] D. Jiao, S. Chakravarty, and C. Dai, "A layered finite-element method for electromagnetic analysis of large-scale high-frequency integrated circuits," *IEEE Trans. Antennas Propagat.*, vol. 55, no. 2, pp. 422–432, Feb. 2007.
- [6] D. Jiao, M. Mazumder, S. Chakravarty, C. Dai, M. Kobrinsky, M. Harmes, and S. List, "A novel technique for full-wave modeling of large-scale three-dimensional high-speed on/off-chip interconnect structures," in *Proc. Int. Conf. Simulation Semicond. Processes Devices (SISPAD)*, 2003, pp. 39–42.
- [7] S. Kapur and D. E. Long, "Large-scale full-wave simulation," in *Proc. IEEE Design Automation Conf.*, 2004.
- [8] D. Gope, A. E. Ruehli, C. Yang, and V. Jandhyala, "(S)PEEC: Time- and frequency-domain surface formulation for modeling conductors and dielectrics in combined circuit electromagnetic simulations," *IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 6, pp. 2453–2464, Jun. 2006.
- [9] Z. G. Qian, J. Xiong, L. Sun, I. T. Chiang, W. C. Chew, L. J. Jiang, and Y. H. Chu, "Crosstalk analysis by fast computational algorithms," in *Proc. IEEE 14th Topical Meeting Electr. Performance Electron. Packag.*, 2005, pp. 367–370.
- [10] F. Ling, V. I. Okhamtovski, W. Harris, S. McCracken, and A. Dengi, "Large-scale broad-band parasitic extraction for fast layout verification of 3-D RF and mixed-signal on-chip structures," *IEEE Trans. Microwave Theory Tech.*, vol. 53, no. 1, pp. 264–273, Jan. 2005.
- [11] Z. Y. Yuan, Z. F. Li, and M. L. Zou, "Computer-aided analysis of on-chip interconnects near semiconductor substrate for high-speed VLSI," *IEEE Trans. Comput.-Aided Design*, vol. 19, no. 9, pp. 990–998, Sep. 2000.
- [12] H. Gan and D. Jiao, "A time-domain layered finite element reduction recovery (LAFE-RR) method for high-frequency VLSI design," *IEEE Trans. Antennas Propagat.*, vol. 55, no. 12, pp. 3620–3629, Dec. 2007.
- [13] H. Gan and D. Jiao, "A fast and high-capacity electromagnetic solution for high-speed IC design," in *IEEE/ACM 2007 Int. Conf. Computer-Aided Design (ICCAD)*, pp. 1–6.
- [14] A. E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-22, no. 3, pp. 216–221, Mar. 1974.
- [15] A. E. Yilmaz, J. M. Jin, and E. Michielssen, "A parallel fft-accelerated transient field-circuit simulator," *IEEE Trans. Microwave Theory Tech.*, vol. 53, no. 9, pp. 2851–2865, Sep. 2005.
- [16] J. Yeo, S. Kwon, and R. Mittra, "Some novel techniques for fast simulation of mixed-signal IC and RF package design," in *Proc. IEEE Antennas Propag. Soc. Symp.*, Monterey, CA, 2004, vol. 3, pp. 3297–3300.
- [17] UMFPAK. [Online]. Available: <http://www.cise.ufl.edu/research/sparse/umfpak/>
- [18] I. S. Duff and J. K. Reid, "The multifrontal solution of indefinite sparse symmetric linear equations," *ACM Trans. Math. Softw.*, vol. 9, no. 3, pp. 302–325, 1983.
- [19] D. Jiao and J. M. Jin, "Finite element analysis in time domain," in *The Finite Element Method in Electromagnetics*. New York: Wiley, 2002, pp. 529–584.
- [20] J. M. Jin, *The Finite Element Method in Electromagnetics*, 1st ed. New York: Wiley, 1993.
- [21] G. Meurant, "A review on the inverse of symmetric tridiagonal and block tridiagonal matrices," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 3, pp. 707–728, Jul. 1992.
- [22] M. J. Kobrinsky, S. Chakravarty, D. Jiao, M. C. Harmes, S. List, and M. Mazumder, "Experimental validation of crosstalk simulations for on-chip interconnects using s-parameters," *IEEE Trans. Adv. Packag.*, vol. 28, no. 1, pp. 57–62, Feb. 2005.



Houle Gan received the B.S. and M.S. degrees in information science and electronic engineering from Zhejiang University, Hangzhou, China, in 2003 and 2006, respectively. He is currently pursuing the Ph.D. degree in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

In 2006, he was a System Engineer in Realsil Microelectronics, Inc., Suzhou, China. In September 2006, he joined On-Chip Electromagnetics Group, Purdue University, as a Research Assistant. His current research interest is computational electromagnetics for large-scale high-frequency integrated circuit design.



Dan Jiao (S'00–M'02–SM'06) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign in October 2001.

She was with the Technology CAD Division, Intel Corporation, until September 2005 as Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. In September 2005, she joined Purdue University, Lafayette, IN, as an Assistant Professor in the School of Electrical and Computer Engineering. She has authored two book chapters and over 70 papers in refereed journals and international conferences.

Her current research interests include high frequency digital, analogue, mixed-signal, and RF IC design and analysis, high-performance VLSI CAD, modeling of micro- and nano-scale circuits, computational electromagnetics, applied electromagnetics, fast and high-capacity numerical methods, fast time domain analysis, scattering and antenna analysis, RF, microwave, and millimeter wave circuits, wireless communication, and bio-electromagnetics.

Dr. Jiao received the NSF CAREER Award in 2008. In 2006, she received the Jack and Cathie Kozik Faculty Start-up Award, which recognizes an outstanding new faculty member in Purdue ECE. She also received an ONR award through Young Investigator Program. In 2004, she received the Best Paper Award from Intel's annual corporate-wide technology conference (Design and Test Technology Conference) for her work on generic broadband model of high-speed circuits. In 2003, she won the Intel Logic Technology Development (LTD) Divisional Achievement Award in recognition of her work on the industry-leading BroadSpice modeling/simulation capability for designing high-speed microprocessors, packages, and circuit boards. She was also awarded the Intel Technology CAD Divisional Achievement Award for the development of innovative full-wave solvers for high frequency IC design. In 2002, she was awarded by Intel Components Research the Intel Hero Award (Intel-wide she was the tenth recipient) for the timely and accurate two- and three-dimensional full-wave simulations. She also won the Intel LTD Team Quality Award for her outstanding contribution to the development of the measurement capability and simulation tools for high frequency on-chip cross-talk. She was the winner of the 2000 Raj Mittra Outstanding Research Award given by the University of Illinois at Urbana-Champaign. She has served as the reviewer for many IEEE journals and conferences.