

From Layout Directly to Simulation: A First-Principle-Guided Circuit Simulator of Linear Complexity and Its Efficient Parallelization

Qing He, Duo Chen, and Dan Jiao, *Senior Member, IEEE*

Abstract—In this paper, guided by electromagnetics-based first principles, the authors develop a transient simulator that allows for the simulation of an integrated circuit including both nonlinear devices and the layout of the linear network in linear complexity. The proposed circuit simulator rigorously captures the coupling between nonlinear circuits and the linear network. In addition, it bypasses the step of circuit extraction, producing a resistor–inductor–capacitor representation of the linear network without any numerical computation. Moreover, it permits an almost embarrassingly parallel implementation on a many-core computing platform, and hence enabling linear speedup. Application to die-package co-simulation as well as very large-scale on-chip circuits involving over 800 000 complementary metal-oxide semiconductor transistors and interconnects having hundreds of millions of unknowns has demonstrated the superior performance of the proposed first-principle-guided circuit simulator.

Index Terms—Circuit simulation, electromagnetic simulation, linear complexity, linear speedup, multi-core, nonlinear circuits, parallel computing, time-domain finite-element method.

I. INTRODUCTION

CIRCUIT simulation is an increasingly indispensable tool for the design of integrated circuits (ICs) and packages. The most prominent circuit simulation programs are simulation program with IC emphasis (SPICE) [1] and its derivatives. SPICE is highly capable of simulating active devices. In the early years, when interconnects and packages could be modeled simply as lumped elements, the linear network is a very small component of an IC. The exponentially increased complexity of ICs and packages, however, has made circuit simulation increasingly challenging.

The simulation of large-scale ICs and packages together with nonlinear transistors results in numerical problems of ultralarge scale, requiring billions of parameters to describe them accurately. In general, to solve a problem of N parameters, the optimal computational complexity one can hope

for is $O(N)$, i.e., linear complexity. Although there have been successes in speeding up the circuit simulation process [2]–[6], as yet, no linear complexity has been achieved. As many-core computing has become a new form of equivalent scaling to facilitate the continuation of Moore’s law, the simulation of very large-scale IC with uncompromised accuracy can be brought to reality in faster CPU runtime if we can exploit the parallelism provided by the many cores on a chip. In general, however, the speedup of an application running on a many-core computing platform over its sequential implementation is governed by Ahmdal’s law [7]. Linear (or optimal) speedup can be achieved only if the computation is embarrassingly parallel.

In addition to circuit-based co-simulation of the linear network and nonlinear circuits, the electromagnetics-based co-simulation has also been studied in the past [8]–[22]. Approaches to coupling both the first-order Maxwell’s equations and the second-order vector wave equations with the lumped circuit models have been developed. The field-circuit co-simulation algorithm has been explored in the framework of the finite-difference time-domain method [8]–[10], [21], [22], the time-domain finite-element method [11]–[18], and the time-domain integral equation method [19]–[20]. However, many of these algorithms were developed for the simulation of microwave and millimeter wave ICs. They often have been found not amenable to very large-scale integrated circuit design because of unique modeling challenges such as conductor loss, strong nonuniformity, large number of conductors, large aspect ratio, and large number of nonlinear devices [23]. In addition, the direct field-based representation of the linear network in these approaches may be too abstract to be put into practical use by circuit designers as they are more grounded in circuit theory. Moreover, existing field-based simulation approaches are not fast enough to meet real-time design needs. Neither linear complexity nor linear speedup has been achieved.

In this paper, going from layout directly to simulation, we develop a transient simulator that allows for the simulation of an IC of $O(N)$ size including both nonlinear devices and the layout of the linear network in $O(N)$ complexity, i.e., optimal complexity. In addition, it permits an almost embarrassingly parallel implementation on a many-core computing platform. Furthermore, the proposed circuit simulator possesses electromagnetic-physics-based accuracy, and hence can be employed to overcome the fundamen-

Manuscript received June 23, 2011; revised October 7, 2011; accepted November 17, 2011. Date of publication January 23, 2012; date of current version March 30, 2012. This work was supported in part by a grant from Intel Corporation, a grant from the Office of Naval Research under Award N00014-10-1-0482, and a grant from the National Science Foundation under Award 0747578 and Award 1065318. Recommended for publication by Associate Editor E.-P. Li upon evaluation of reviewers’ comments.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: qhe@purdue.edu; chen256@purdue.edu; djiao@purdue.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCPMT.2011.2179547

tal limits of circuit-principle-based analysis for high-speed and high-performance circuit design. Moreover, it captures the interaction between the nonlinear circuit and the linear network, with the nonlinear-linear coupling rigorously taken into consideration. In addition, it bypasses the step of linear-network extraction while retaining a resistor-inductor-capacitor (RLC) based perspective. By using the proposed first-principle-guided approach, an RLC-based representation of the linear network can be obtained without any numerical computation and without any approximation.

II. PROPOSED CO-SIMULATION OF NONLINEAR CIRCUITS AND LINEAR NETWORK GUIDED BY FIRST PRINCIPLES

Consider an integrated system consisting of a linear network and nonlinear circuits, as shown in Fig. 1. All the interconnects, packages, and passive components belong to the linear network. All the nonlinear devices are in the nonlinear block. In this section, we derive the system of equations that governs the co-simulation of the linear network and nonlinear circuits.

A. First-Principle-Based RLC Representation of the Linear Network

The physical phenomena in the linear network are governed by Maxwell's equations, which suggest

$$\begin{aligned} \nabla \times \left[\mu_r^{-1} \nabla \times \mathbf{E}(\mathbf{r}, t) \right] + \mu_0 \varepsilon \partial_t^2 \mathbf{E}(\mathbf{r}, t) \\ + \mu_0 \sigma \partial_t \mathbf{E}(\mathbf{r}, t) = -\mu_0 \partial_t \mathbf{J}(\mathbf{r}, t) \end{aligned} \quad (1)$$

where \mathbf{E} is electric field, μ_0 is free-space permeability, μ_r is relative permeability, ε is permittivity, σ is conductivity, \mathbf{J} is current density, and \mathbf{r} denotes a point in a 3-D space.

Equation (1) can be solved by either an integral equation or partial differential equation based approach. We employ a time-domain finite-element method to solve (1) and its boundary conditions [25]. Compared to other partial differential equation based approaches, a finite-element method permits an accurate modeling of arbitrary inhomogeneous materials as well as irregularly shaped geometries. Following the derivation given in [25], we obtain

$$\mathbf{T} \frac{d^2 u}{dt^2} + \mathbf{T}_\sigma \frac{du}{dt} + \mathbf{S} u = \frac{d\tilde{I}}{dt} \quad (2)$$

in which \mathbf{T} , \mathbf{T}_σ , and \mathbf{S} are square *sparse* matrices, u is the unknown field vector, and \tilde{I} is the vector of the currents injected into the linear system. The elements of the matrices \mathbf{T} , \mathbf{T}_σ , and \mathbf{S} are given by

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

where \mathbf{N}_i and \mathbf{N}_j are the vector basis functions used to expand unknown field \mathbf{E} , and $\langle \cdot, \cdot \rangle_V$ denotes a volume integration. Matrix \mathbf{T} is symmetric and positive definite, \mathbf{S} as well as \mathbf{T}_σ is symmetric semi-positive definite. Given the physical layout of an IC, sparse matrices \mathbf{T} , \mathbf{T}_σ , and \mathbf{S} can be readily obtained by assembling in $O(N)$ time, with N being the number of discretized edges in the computational domain. The constant

in front of N is very small, less than 40 in general, regardless of the size of N .

The elements of the current vector \tilde{I} are given by

$$\tilde{I}_i = -\mu_0 \langle \mathbf{N}_i, \mathbf{J} \rangle_V. \quad (4)$$

At the i th edge of a finite-element-based discretization, if a current source of magnitude I_i is attached, (4) can be evaluated as

$$\tilde{I}_i = -\mu_0 I_i l_i \quad (5)$$

where l_i is the length of the i th edge. From Fig. 1, it can be seen clearly that the current flowing into the linear system has two components. I_s which is a supply current, and I_{nl} which is injected from the nonlinear circuit. Hence, (5) can be written as

$$\tilde{I}_i = -\mu_0 (I_{s,i} + I_{nl,i}) l_i. \quad (6)$$

The voltage across the i th edge, V_i , can be evaluated from u after (2) is solved. For example, if the reference direction of the voltage is along the direction of the i th normalized vector basis, then

$$V_i = l_i u_i. \quad (7)$$

From (2), we immediately obtain an RLC-based representation of the linear network. To be specific, matrix \mathbf{T} corresponds to capacitance matrix \mathbf{C} , matrix \mathbf{T}_σ corresponds to the inverse of the resistance matrix \mathbf{R} , and matrix \mathbf{S} corresponds to the inverse of the inductance matrix \mathbf{L} . Their elements are readily known from (3), where the volume integration $\langle \cdot, \cdot \rangle_V$ is obtained analytically in the finite-element method without the need of any numerical computation. It is worth mentioning that the RLC model represented by (2) has a resolution as fine as the resolution we use to discretize the layout structure. To be specific, it is the RLC matrix characterizing the interaction among all the edges present in the discretized layout. In contrast, the RLC model obtained from conventional circuit extraction is for the interaction among selected terminals, the resolution of which is much coarser. In addition, notice that we do not have to perform matrix inverses to form \mathbf{R} and \mathbf{L} explicitly because we only need their expressions to perform matrix-vector multiplications when simulating (2). The extraction of the linear network is thus bypassed in the proposed circuit simulator. Furthermore, (2) will be co-simulated at each time step with the system of equations that governs the nonlinear devices. Thus, the time-dependent interaction between the linear network and nonlinear devices is rigorously accounted for. Moreover, obtained rigorously from electromagnetic-field-based first principles, (2) is guaranteed to be passive and stable. To explain, a finite-element-based discretization of Maxwell's equations yields a Hermitian-positive definite \mathbf{T} , a Hermitian-semi-positive definite \mathbf{T}_σ , and a Hermitian-semi-positive definite \mathbf{S} . As a result, the real part of the poles of (2) are always no greater than zero. Hence, the system is always stable. In addition, \mathbf{T}_σ 's being semi-positive definite guarantees the passivity of (2).

B. Modeling of Nonlinear Circuits

The nonlinear circuit shown in Fig. 1 can be modeled by

$$I_c = f(V_c, t) \quad (8)$$

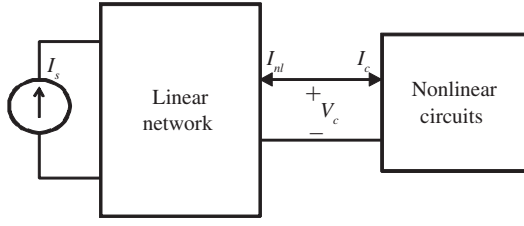


Fig. 1. Illustration of an IC system consisting of a linear network and nonlinear circuits.

where f is a nonlinear function, t is time, I_c is current, and V_c is voltage. If the nonlinear circuit is a network that consists of a number of nonlinear components, it can be analyzed by the modified nodal analysis [26]. Without loss of generality, the nonlinear network can be modeled by the following nonlinear system of equations:

$$\tilde{\mathbf{G}}(\cdot)x + \tilde{\mathbf{C}}(\cdot)\frac{dx}{dt} = b \quad (9)$$

where the unknown vector $x = [V_c, I_c, v_c, i_c]^T$, in which V_c , and I_c are, respectively, the voltage and current or vectors of voltage and current at the interface between the nonlinear circuits and the linear network as shown in Fig. 1, v_c is a vector of node voltages internal to the nonlinear circuit, and i_c is a vector of branch currents flowing through inductors and voltage sources, also internal to the nonlinear circuit. In a discretized physical layout, the interface between a nonlinear device and a linear network is composed of a group of edges, where the voltage drop across each edge makes V_c , which can be evaluated from field solution based on (7), and the current flowing along these edges makes I_c . In (9), $\tilde{\mathbf{G}}$ denotes a nonlinear mapping from x to b , and $\tilde{\mathbf{C}}$ denotes a nonlinear mapping from dx/dt to b . Both $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{C}}$ can be time dependent. The nonlinear model (8) is a special case of (9).

C. Combined System of Equations for Co-Simulation

From Fig. 1, it can be seen clearly that at the interface between the linear network and nonlinear circuits, the following system of equations satisfies

$$I_{nl} + I_c = 0. \quad (10)$$

Thus, to accurately obtain the transient response of an integrated nonlinear-linear system, we need to co-simulate (2), (9), and (10).

The combined system of (2), (9), and (10) can be written more compactly as the following:

$$\mathbf{T}\frac{d^2u}{dt^2} + \mathbf{T}_\sigma\frac{du}{dt} + \mathbf{S}u = \frac{d(\tilde{I}_s - \tilde{I}_c)}{dt}$$

$$\tilde{\mathbf{G}}(\cdot)x + \tilde{\mathbf{C}}(\cdot)\frac{dx}{dt} = b, \text{ where } x = [V_c, I_c, v_c, i_c]^T \quad (11)$$

where the entries of vectors \tilde{I}_s and \tilde{I}_c are

$$\tilde{I}_{s,i} = -\mu_0 I_{s,i} l_i \quad (12)$$

$$\tilde{I}_{c,i} = \mu_0 I_{c,i} l_i \quad (13)$$

as can be seen from (6).

It is worth mentioning that if the function f in (8) is linear and time independent, the co-simulation of (2), (9), and (10) is straightforward in the proposed circuit simulator. To explain, the functions f of a constant and linear resistor of resistance R , an inductor of inductance L , and a capacitor of capacitance C are

$$f_R = \frac{V_c}{R}, \quad f_L = \frac{1}{L} \int V_c dt, \quad \text{and} \quad f_C = C \frac{dV_c}{dt} \quad (14)$$

respectively. By substituting (10) and (14) into (6), and employing (7) and (8), it can be readily derived from (2) that if the lumped R , L , and C are attached to the i th edge in a finite-element-based discretization of the IC, they only contribute to the i th diagonal element of matrices \mathbf{T}_σ , \mathbf{S} , and \mathbf{T} , which amounts to adding $(\mu_0 l_i^2/R)$ to $\mathbf{T}_{\sigma,ii}$, $(\mu_0 l_i^2/L)$ to \mathbf{S}_{ii} , and $\mu_0 l_i^2 C$ to \mathbf{T}_{ii} , respectively. If the circuit connected to the linear network is nonlinear, the aforementioned approach becomes highly computationally expensive because the entire system matrix has to be factorized and solved at each time step, as the system becomes time dependent and nonlinear. In the following section, we propose an efficient algorithm to co-simulate the linear network and nonlinear circuits in linear complexity.

III. LINEAR-COMPLEXITY NONLINEAR-LINEAR CO-SIMULATION

A. Algorithm

Discretizing the first equation in (11) by a central difference scheme, we obtain

$$\mathbf{P}u^{n+1} = \left(2\mathbf{T} - \Delta t_e^2 \mathbf{S}\right) u^n + [0.5\Delta t_e \mathbf{T}_\sigma - \mathbf{T}] u^{n-1}$$

$$+ \Delta t_e^2 \left[\frac{d(\tilde{I}_s - \tilde{I}_c)}{dt} \right]^n \quad (15)$$

in which

$$\mathbf{P} = \mathbf{T} + 0.5\Delta t_e \mathbf{T}_\sigma \quad (16)$$

and Δt_e represents the time step used in the simulation of the linear network. The field value at the $(n+1)$ -th time step, u^{n+1} , can be solved in a time-marching fashion from the solution of u at previous two time steps.

The unknowns involved in (15) include unknowns in linear network and those attached to nonlinear circuits. The system of equations in (15) thus includes a subsystem of equations that is purely linear, and the other subsystem of equations that is nonlinear. The nonlinear equations correspond to the rows of (15) that have a nonzero \tilde{I}_c . For a system of equations like (15), if one eliminates the linear unknowns (via Gaussian elimination, for example) from (15) so that the resultant system is purely nonlinear and can be co-simulated with the nonlinear circuit equation in (11), the resultant system matrix is dense, which leads to a high computational cost because this dense matrix has to be solved at each Newton step. If one keeps all the linear and nonlinear unknowns in (15) and solves it as a whole, the Newton iteration will also involve the simulation of the linear system of equations. Neither approach could yield linear complexity in computation. Next, we will show how this challenge is overcome by the proposed method, for arbitrary

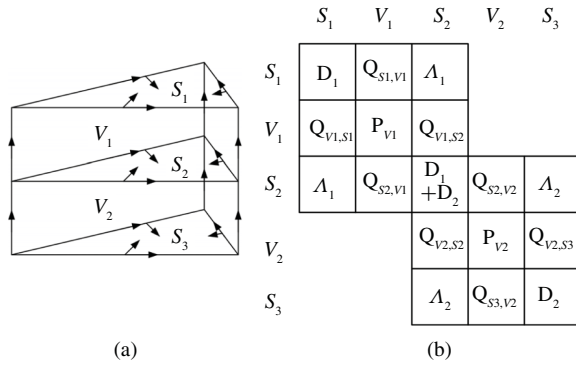


Fig. 2. (a) Orthogonal prism-element-based discretization and unknown ordering scheme. (b) 3-D layered system matrix \$\mathbf{P}\$ [27].

3-D layout structures connected to arbitrary nonlinear circuits in inhomogeneous materials.

We discretize the physical layout of an IC system into layers of triangular prism elements, as shown in Fig. 2(a). A triangular prism element is used because it is a natural choice for discretizing the geometry of ICs, which is straight in one direction and can be arbitrarily shaped in the other two directions. Even though one uses tetrahedral elements to discretize an IC, he would get layers of tetrahedral elements since the mesh has to be partitioned between layers. It is also worth mentioning that a 3-D structure that is arbitrarily shaped in all three directions can also be sliced into layers, and hence discretized into triangular prism elements. One just has to use a staircase approximation in geometrical modeling along the prism axis direction, the accuracy of which can be controlled by reducing the space step along that direction. In addition, the materials permitted by the proposed simulation algorithm can be arbitrarily inhomogeneous. It does not require materials to be layered. In other words, the material property can be different in each triangular prism element. Therefore, in each layer, the material can be inhomogeneous. The same is true across different layers.

We also discretize conductors in order to capture internal fields accurately. In each element, the electric field is expanded into orthogonal vector basis functions [27]. The unknowns are ordered layer by layer. In each layer, the unknowns are divided into surface and volume unknowns. As shown in Fig. 2(a), the unknowns perpendicular to the prism axis are called *surface unknowns*, and the unknowns along the prism axis are called *volume unknowns*. The unknowns are then ordered from \$S_1\$ to \$V_1\$ to \$S_2\$ to \$V_2\$ and continue, resulting in a 3-D layered system matrix shown in Fig. 2(b), which is \$\mathbf{P}\$ in (16).

Because the vector bases associated with surface unknowns are perpendicular to those associated with volume unknowns, as can be seen from Fig. 2(a), and \$\mathbf{P}\$ in (16) solely comprises matrices formed by the inner product of vector bases, as can be seen from (3), all the \$\mathbf{Q}\$ blocks in Fig. 2(b) vanish. As a result, in \$\mathbf{P}\$, which is illustrated in Fig. 2(b), the surface-unknown-based subsystem is completely decoupled from the volume-unknown-based subsystem without any computational

cost as the following:

$$\mathbf{P}_S u_S^{n+1} = b_S \quad (17)$$

$$\mathbf{P}_{Vl} u_{Vl}^{n+1} = b_{Vl} \quad l = 1, 2, \dots, L \quad (18)$$

in which \$u_S\$ denotes surface unknowns in the entire unknown set \$u\$, and \$u_{Vl}\$ denotes volume unknowns in layer \$l\$, \$\mathbf{P}_S\$ is the surface-unknown-based system matrix, \$\mathbf{P}_{Vl}\$ is the volume-unknown-based system matrix in layer \$l\$, \$L\$ is the total number of layers, and \$b_S\$, and \$b_{Vl}\$ are, respectively

$$b_S = \left\{ \left(2\mathbf{T} - \Delta t_e^2 \mathbf{S} \right) u^n + [0.5 \Delta t_e \mathbf{T}_\sigma - \mathbf{T}] u^{n-1} + \Delta t_e^2 \left[\frac{d(\tilde{I}_s - \tilde{I}_c)}{dt} \right]^n \right\}_s \quad (19)$$

and

$$b_{Vl} = \left\{ \left(2\mathbf{T} - \Delta t_e^2 \mathbf{S} \right) u^n + [0.5 \Delta t_e \mathbf{T}_\sigma - \mathbf{T}] u^{n-1} + \Delta t_e^2 \left[\frac{d(\tilde{I}_s - \tilde{I}_c)}{dt} \right]^n \right\}_{Vl} \quad (20)$$

where the subscript 's' and 'Vl', respectively, denote the rows of the right-hand side vector corresponding to the surface unknowns, and the rows corresponding to volume unknowns in layer \$l\$.

The \$\mathbf{P}_{Vl}(l = 1, 2, \dots, L)\$ is the diagonal block of \$\mathbf{P}\$ formed by volume unknowns in layer \$l\$. Examples of \$\mathbf{P}_{Vl}\$, \$\mathbf{P}_{V1}\$ and \$\mathbf{P}_{V2}\$, can be seen in Fig. 2(b). Because all the \$\mathbf{Q}\$ blocks are zero in Fig. 2(b), clearly, the \$\mathbf{P}_{Vl}\$ in one layer is fully decoupled from the \$\mathbf{P}_{Vl}\$ in another layer. As a result, the volume-unknown-based subsystem of (15) is naturally decomposed into subsystems in each layer, as shown by (18). The subsystem in each layer, \$\mathbf{P}_{Vl}\$, can be further decomposed into 1-D matrices, with each 1-D matrix being tridiagonal, as shown in [27].

From Fig. 2(b), the \$\mathbf{P}_S\$ in (17) can be written as

$$\mathbf{P}_S = \begin{bmatrix} \mathbf{D}_1 & \Lambda_1 & & & & \\ \Lambda_1 & \mathbf{D}_1 + \mathbf{D}_2 & \Lambda_2 & & & \\ & \Lambda_2 & \mathbf{D}_2 + \mathbf{D}_3 & & & \\ & & \dots & & & \\ & & & & \mathbf{D}_{L-1} + \mathbf{D}_L & \Lambda_L \\ & & & & \Lambda_L & \mathbf{D}_L \end{bmatrix} \quad (21)$$

which is a tridiagonal matrix since each \$\mathbf{D}_l\$ and \$\Lambda_l\$ (\$l = 1, 2, \dots, L\$) block is diagonal due to the orthogonality of the vector basis functions. The dimension of each \$\mathbf{D}_l\$ and \$\Lambda_l\$ block in (21) is \$N_s\$ by \$N_s\$, where \$N_s\$ is the number of unknowns on a single surface. The number of diagonal blocks in (21) is \$L + 1\$, where \$L\$ is the number of layers. To facilitate efficient co-simulation with nonlinear circuits, we permute unknowns in (21) to make it a block diagonal matrix as shown below

$$\mathbf{P}_S = \begin{bmatrix} \mathbf{T}_1 & & & \\ & \mathbf{T}_2 & & \\ & & \dots & \\ & & & \mathbf{T}_{N_s} \end{bmatrix} \quad (22)$$

in which each diagonal block \mathbf{T}_i ($i = 1, 2, \dots, N_s$) is a tridiagonal matrix of size $L + 1$

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{D}_{1,i} & \Lambda_{1,i} & & & \\ \Lambda_{1,i} & \mathbf{D}_{1,i} + \mathbf{D}_{2,i} & \Lambda_{2,i} & & \\ & & \cdots & \Lambda_{L,i} & \\ & & & \Lambda_{L,i} & \mathbf{D}_{L+1,i} \end{bmatrix} \quad (23)$$

where $\mathbf{D}_{l,i}$ and $\Lambda_{l,i}$, respectively, denote the i th entry in diagonal matrix \mathbf{D}_l , and Λ_l . The transformation from (21) to (22) can be understood as the following. In (21), we first order all the surface unknowns on surface 1 (S_1) as shown in Fig. 2(a), the number of which is N_s , we then order all the surface unknowns on surface 2 (S_2), and continue to the last surface, which is the bottom surface of the L th layer. In (22), the ordering scheme is different. We start from a *single* surface unknown on S_1 , we find its counterparts on S_2 , S_3 , etc., and order them one by one, resulting in a tridiagonal matrix \mathbf{T}_1 of size $L + 1$ as shown in (23); after that, we return to S_1 and order another surface unknown and its L counterparts on all the other surfaces, which yields \mathbf{T}_2 in (22); and we continue until the last surface unknown on S_1 and its counterparts on all the other surfaces are ordered.

With (22), we decompose the matrix \mathbf{P}_S in (21) into small tridiagonal matrices \mathbf{T}_i of 1-D size, which are fully decoupled. It is clear that such decomposition is computation free. With that, (17) is naturally decomposed to

$$\mathbf{T}_i u_{S,i}^{n+1} = b_{S,i}, \quad i = 1, 2, \dots, N_s \quad (24)$$

where $u_{S,i}$ and $b_{S,i}$ are, respectively, the i th subset of u_s and b_s in (17) corresponding to the \mathbf{T}_i block.

Without loss of generality, assume that the nonlinear circuits are attached to the layout of the linear network via surface unknowns. This is true in general because transistors switch at the bottom of a chip, which can be viewed as current sources aligned with the stack growth direction. The layer growth direction (prism axis) shown in Fig. 2(a) is generally chosen to be perpendicular to the dielectric stack growth direction so that the resultant cross section has a minimal size for computation efficiency. Thus, nonlinear devices are attached to surface unknowns. When attached to nonlinear circuits, (24) can be rewritten as the following.

For \mathbf{T}_i blocks that are attached to nonlinear circuits, (24) becomes

$$\mathbf{T}_i u_{S,i}^{n+1} = \tilde{b}_{S,i} - \Delta t_e^2 \left(\frac{d\tilde{I}_c}{dt} \right)^n. \quad (25)$$

For \mathbf{T}_i blocks that are not attached to nonlinear circuits, (24) becomes

$$\mathbf{T}_i u_{S,i}^{n+1} = \tilde{b}_{S,i} \quad (26)$$

where $\tilde{b}_{S,i}$ is the right-hand side of (19) without \tilde{I}_c . Because \tilde{I}_c is a nonlinear function of time, (25) is a nonlinear system of equations.

It is clear that, now, only (25) is nonlinear. All the other rows of equations in (15), which comprise (18) and (26), are linear. As (18) and (26) are fully decoupled from (25), they can be solved by a linear simulator without being affected by the nonlinear solution of (25). This is very different from

what happens in a conventional circuit simulator. Since the decomposition from a 3-D system to a 2-D system, i.e., (15) to (17)/(18), and the decomposition from a 2-D system to a 1-D system, i.e., (17) to (22), are not feasible in a conventional circuit simulator, the solution of nonlinear circuits significantly affects the efficiency of the linear simulation part in the simulation of a combined nonlinear-linear system.

Although (25) is already a small system made of fully decoupled \mathbf{T}_i blocks, each of which has a 1-D size, we can further improve its computational efficiency by separating the linear equations in (25) from its nonlinear ones and solving them separately, if not all the surface unknowns associated with \mathbf{T}_i are attached to the nonlinear devices. Take one \mathbf{T}_i block as an example, we divide unknowns in $u_{s,i}$ into two groups: one is completely inside the linear network and the other is attached to nonlinear circuits. The first group is denoted by $u_{s,ie}$ and the second is $u_{s,ic}$. We then cast (25) into the following form:

$$\begin{bmatrix} \mathbf{T}_{i,ee} & \mathbf{T}_{i,ec} \\ \mathbf{T}_{i,ec}^T & \mathbf{T}_{i,cc} \end{bmatrix} \begin{bmatrix} u_{s,ie}^{n+1} \\ u_{s,ic}^{n+1} \end{bmatrix} = \begin{bmatrix} \tilde{b}_{s,ie} \\ \tilde{b}_{s,ic} \end{bmatrix} + \begin{bmatrix} 0 \\ -\Delta t_e^2 \left(\frac{d\tilde{I}_c}{dt} \right)^n \end{bmatrix}. \quad (27)$$

By substituting the first equation into the second, (27) can be reduced to a system that only involves the nonlinear unknowns

$$\mathbf{T}'_{i,cc} u_{s,ic}^{n+1} + \Delta t_e^2 \left(\frac{d\tilde{I}_c}{dt} \right)^n = b_{i,c} \quad (28)$$

where

$$\mathbf{T}'_{i,cc} = \mathbf{T}_{i,cc} - \mathbf{T}_{i,ec}^T \mathbf{T}_{i,ee}^{-1} \mathbf{T}_{i,ec} \quad (29)$$

$$b_{i,c} = \tilde{b}_{s,ic} - \mathbf{T}_{i,ec}^T \mathbf{T}_{i,ee}^{-1} \tilde{b}_{s,ie}. \quad (30)$$

Here, since \mathbf{T}_i in (25) is a tridiagonal matrix as shown in (23), the Schur complement, $\mathbf{T}'_{i,cc}$, has a good property that it remains to be a tridiagonal matrix. To prove this, consider the mn -th element of $\mathbf{T}'_{i,cc}$, where $n > m + 1$, from (29), $\mathbf{T}'_{i,cc}^{(m,n)}$ can be evaluated as

$$\mathbf{T}'_{i,cc}^{(m,n)} = \mathbf{T}_{i,cc}^{(m,n)} - (\mathbf{T}_{i,ce})^{(m,n')} (\mathbf{T}_{i,ee}^{-1})^{(n',m')} (\mathbf{T}_{i,ec})^{(m',n)} \quad (31)$$

where $(\mathbf{T}_{i,ce})^{(m,n')}$ denotes the nonzero block in $\mathbf{T}_{i,ce}$, $(\mathbf{T}_{i,ec})^{(m',n)}$ denotes the nonzero block in $\mathbf{T}_{i,ec}$, n' denotes the set that contains all the column indexes of the nonzero elements in $\mathbf{T}_{i,ce}$ in the m th row, and m' contains all the row indexes of the nonzero elements in $\mathbf{T}_{i,ec}$ in the n th column, as can be seen from Fig. 3. In (31), we only need to consider $(\mathbf{T}_{i,ee}^{-1})^{(n',m')}$ block because other blocks in $\mathbf{T}_{i,ee}^{-1}$ do not participate in the computation due to zero columns in $\mathbf{T}_{i,ce}$ and zero rows in $\mathbf{T}_{i,ec}$. One should also realize that the $\mathbf{T}_{i,ee}^{-1}$ is a block diagonal matrix, with each block denoting the system matrix formed by the e -unknowns (in $u_{s,ie}$) sandwiched between two c -unknowns (in $u_{s,ic}$). To see this clearly, one can refer to Fig. 4, in which we plot all the edges that exist in a \mathbf{T}_i block, where nonlinear devices are attached to the red edges. In this figure, the c -unknowns correspond to the red edges, while the e -unknowns correspond to the black edges. Hence, the $\mathbf{T}_{i,ee}^{-1}$ is a block diagonal matrix, with each block denoting the system matrix formed by the black edges sandwiched between two red edges. Because \mathbf{T}_i in (25) is a tridiagonal matrix, and

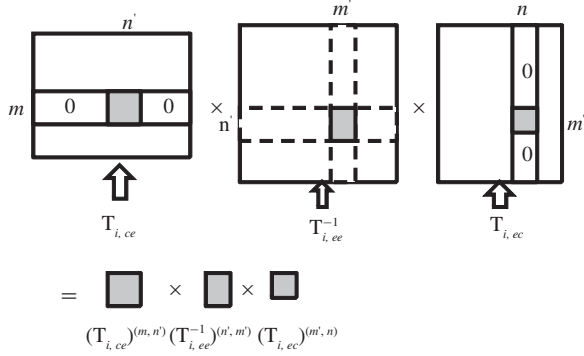


Fig. 3. Actual operation involved in $\mathbf{T}_{i,ec}^T \mathbf{T}_{i,ee}^{-1} \mathbf{T}_{i,ec}$.

$\mathbf{T}_{i,ee}$, $\mathbf{T}_{i,ec}$, $\mathbf{T}_{i,ce}$, and $\mathbf{T}_{i,cc}$ are its four sub-blocks as shown in (27), there are at most two nonzero elements in $(\mathbf{T}_{i,ce})^{(m,n')}$, which corresponds to the set $n' = \{g(m) - 1, g(m) + 1\}$, where $g(m)$ is the global index of m in (27). There are also at most two nonzero elements in $(\mathbf{T}_{i,ec})^{(m',n)}$, which corresponds to the set $m' = \{g(n) - 1, g(n) + 1\}$, where $g(n)$ is the global index of n in (27). If $n > m + 1$, then the two sets n' and m' do not belong to the same diagonal block in $\mathbf{T}_{i,ee}^{-1}$, which can be visualized from Fig. 4 also. As a result, the corresponding block $(\mathbf{T}_{i,ee}^{-1})^{(n',m')}$ is zero. Because $\mathbf{T}_{i,cc}^{(m,n)}$ is also zero when $n > m + 1$, from (31), we have $\mathbf{T}_{i,cc}^{(m,n)} = 0$ when $n > m + 1$. Since $\mathbf{T}_{i,cc}$ is also symmetric, we prove that $\mathbf{T}_{i,cc}$ is a tridiagonal matrix like \mathbf{T}_i .

From (25) to (28), the dimension of the nonlinear system is reduced from the dimension of \mathbf{T}_i to the actual number of nonlinear devices in one \mathbf{T}_i block. The 1-D (28) needs to be co-simulated with the nonlinear device model characterized by (9). Assembling (28) with (9) for each \mathbf{T}_i block in (22) that is attached to nonlinear circuits yields the following decoupled block tridiagonal matrix equations to be solved by the Newton–Raphson method

$$\begin{bmatrix} \mathbf{D}_{i,c,1} & \mathbf{OD}_{i,c,1} & & \\ \mathbf{OD}_{i,c,1} & \mathbf{D}_{i,c,1} & \mathbf{OD}_{i,c,2} & \\ & & \dots & \end{bmatrix} \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \dots \end{pmatrix} = \begin{pmatrix} \mathbf{F}(x_{i,1}) \\ \mathbf{G}(x_{i,2}) \\ \dots \end{pmatrix} \quad i = 1, 2, \dots \quad (32)$$

where i is the index of the \mathbf{T}_i block attached to nonlinear circuits, and x is the unknown set shown in (11), which includes the voltage and current at the interface between the nonlinear device and the layout corresponding to the \mathbf{T}_i block and the voltage and current internal to the nonlinear device. The number of diagonal blocks in (32) is the number of nonlinear devices attached to a single \mathbf{T}_i block, and the dimension of each diagonal block is the number of state variables in a single nonlinear device. The complexity of solving (32) at each Newton step is $O(p \times k^3)$ where p is the number of nonlinear devices attached to a single \mathbf{T}_i block and k is the number of state variables of each device. This leads to $O(k^2 M)$ complexity, where $M = p \times k$ is the total number of unknowns in the nonlinear system associated with a single \mathbf{T}_i block. Because k is a small constant that does not depend on M , we obtain a linear complexity. After all

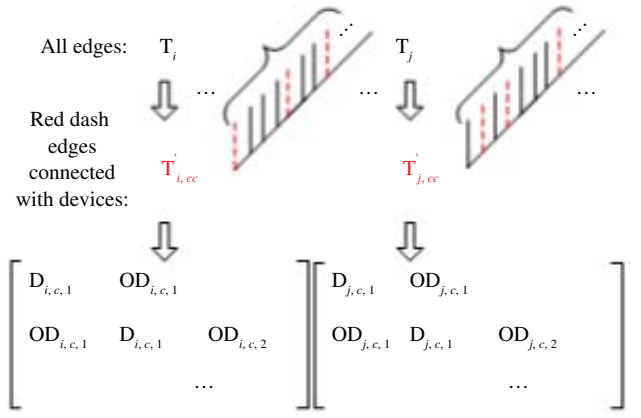


Fig. 4. Illustration of the nonlinear system. Enabled by the proposed circuit simulation algorithm, it only contains decoupled block tridiagonal matrices.

nonlinear unknowns are solved from (32), we substitute them into the first equation in (27) to solve linear unknowns in the \mathbf{T}_i block. This step, also, has a linear complexity because $\mathbf{T}_{i,ee}$ is tridiagonal. The above procedure is repeated for each \mathbf{T}_i block attached to nonlinear circuits in (22). Fig. 4 shows an example, in which two blocks, \mathbf{T}_i and \mathbf{T}_j , are attached to nonlinear devices at the red surface edges. The resultant block tridiagonal matrix (32) associated with \mathbf{T}_i is fully decoupled from that associated with \mathbf{T}_j because \mathbf{T}_i and \mathbf{T}_j blocks are fully decoupled in (22).

From the aforementioned procedure, it can be seen that in the proposed method, regardless of the number of nonlinear devices attached to the ICs, we only need to solve multiple fully decoupled block tridiagonal matrices (32), each of which is 1-D size. To be more specific, the number of blocks in (25), and hence the number of (32) we need to solve, is maximally N_s , for which all the \mathbf{T}_i blocks in (22) are attached to nonlinear devices. The maximum number of diagonal blocks in (32) is $L + 1$, for which each edge of the \mathbf{T}_i block is attached to a nonlinear device. The dimension of each diagonal block in (32) is the number of state variables in a single nonlinear device, k . The total cost of solving N_s fully decoupled matrices (32) is $O(N_s L)$, which is $O(N)$, and hence linear. This is true irrespective of the nonlinear circuit model characterized by (9).

B. Summary of the Overall Procedure

With the proposed method, the simulation of the combined nonlinear–linear system, (11), becomes the simulation of the following decoupled problems (33a)–(33d)

$$\mathbf{P}_{Vl} u_{Vl}^{n+1} = b_{Vl}, \quad l = 1, 2, \dots, L. \quad (33a)$$

For each \mathbf{T}_i block in (22) not attached to nonlinear circuits

$$\mathbf{T}_i u_{s,i}^{n+1} = \tilde{b}_{s,i}, \quad i = 1, 2, \dots \quad (33b)$$

For each \mathbf{T}_i block in (22) attached to nonlinear circuits, solve nonlinear equations

$$\begin{bmatrix} \mathbf{D}_{i,c,1} & \mathbf{OD}_{i,c,1} & & \\ \mathbf{OD}_{i,c,1} & \mathbf{D}_{i,c,1} & \mathbf{OD}_{i,c,2} & \\ & & \dots & \end{bmatrix} \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \dots \end{pmatrix} = \begin{pmatrix} \mathbf{F}(x_{i,1}) \\ \mathbf{G}(x_{i,2}) \\ \dots \end{pmatrix}, \quad i = 1, 2, \dots \quad (33c)$$

where $u_{s,i}^{n+1}$ in (27) is a subset of the above x_i vector.

Then

$$\mathbf{T}_{i,ee} u_{s,ie}^{n+1} = \tilde{b}_{s,ie} - \mathbf{T}_{i,ec} u_{s,ic}^{n+1}, \quad i = 1, 2, \dots \quad (33d)$$

where the solution of (33d) is obtained after $u_{s,ic}^{n+1}$ is solved from the nonlinear equation (33c).

The overall procedure of the proposed simulation is as follows.

Start the time-marching with two initial conditions u^n and u^{n-1} . To obtain u^{n+1} , which is composed of surface unknowns at the $(n+1)$ -th step, u_S^{n+1} , and the volume unknowns at the $(n+1)$ -th step, u_{VI}^{n+1} ($l = 1, 2, \dots, L$), do the following.

Step 1: Generating the right-hand side vectors used in (33). The $b_{VI}(l = 1, 2, \dots, L)$ is obtained based on (20) in which \tilde{I}_c is zero. The \tilde{b}_s is the right-hand side of (19) without \tilde{I}_c . The right-hand side of (33c) is obtained from (28) and the nonlinear system of (9) that describes the nonlinear circuit.

Step 2: Solve (33a) to obtain all the volume unknowns u_{VI}^{n+1} ($l = 1, 2, \dots, L$).

Step 3: Solve (33b) to obtain all the linear surface unknowns.

Step 4: All the nonlinear surface unknowns are associated with the \mathbf{T}_i blocks attached to nonlinear circuits in (22). For each \mathbf{T}_i block attached to nonlinear circuits, solve its corresponding nonlinear system (33c) by Newton's method to obtain the solution of nonlinear unknowns associated with the \mathbf{T}_i block, and after that, find the solution of linear unknowns associated with this block by using (33d).

Go back to Step 1 until the simulation of the required time window is finished.

The cost of (33a) is linear by using the volume unknown solver in the orthogonal finite-element reduction-recovery method [27]. The cost of (33b) is linear because each \mathbf{T}_i is a tridiagonal matrix [36]. The cost of (33c) is linear as analyzed in Section III-A. The cost of (33d) is again linear because of tridiagonal matrix $\mathbf{T}_{i,ee}$. As a result, the overall complexity of the proposed circuit simulator is linear.

It is worth mentioning that (33) is formulated based on the assumption that nonlinear circuits are attached to the layout of the linear network via surface unknowns. This assumption is true in general, as analyzed in Section III-A. In the rare circumstances in which one has to attach the nonlinear circuits through volume unknowns, the proposed simulation algorithm is equally applicable because the solution of (33a) has also been decomposed into the solution of multiple 1-D tridiagonal matrices as shown in [27]. The nonlinear circuit equation (9) again can be co-simulated with these fully decoupled 1-D tridiagonal matrices, yielding fully decoupled nonlinear block tridiagonal subsystems as shown in (33c).

IV. EFFICIENT PARALLELIZATION ON MANY-CORE PLATFORMS

As can be seen from (33), the proposed first-principle-guided linear-complexity circuit simulator permits an almost embarrassingly parallel implementation. The simulation of volume unknowns, shown by (33a), is fully decoupled from the simulation of surface unknowns. The simulation of linear surface unknowns, shown by (33b), is fully decoupled from

the simulation of nonlinear surface unknowns, shown by (33c). In (33b), each \mathbf{T}_i block can be simulated separately from the other \mathbf{T}_i blocks. Similarly, in (33c), each \mathbf{T}_i block attached to nonlinear circuits can be simulated separately from the other \mathbf{T}_i blocks attached to nonlinear circuits. These decoupled problems can be readily distributed to a many-core/node platform to solve them in an embarrassingly parallel fashion. In other words, at each time step, after the right-hand side is prepared based on the field solution obtained at previous two time steps in Step 1, Steps 2–4 described in Section III-B can be done concurrently.

As for the partition of the circuit, we implement it in the same way as reported in [28], where we developed a parallel simulator for simulating linear networks. Here, different from [28], we need to co-simulate nonlinear circuits and linear network. With the formulation shown in (33), we partition nonlinear circuits together with surface unknowns. The load across different nodes is balanced by the actual number of operations associated with the decoupled subsystems shown in (33) instead of based on geometry. The data communication involved in generating right-hand side vectors at each time instant is the same as that reported in [28].

V. STABILITY ANALYSIS

The time step required by the simulation of the linear network for maintaining stability can be different from that required by the Newton-method-based simulation of nonlinear circuits. We denote the former by Δt_e , the latter by Δt_c , and discuss their choices in the following.

For the linear network, a central-difference-based time-domain finite-element solution is guaranteed to be stable if the following condition is satisfied:

$$\Delta t_e \leq \frac{\sqrt{\lambda_{\max}}}{\sqrt{\rho(\mathbf{T}^{-1}\mathbf{S})}} \quad (34)$$

in which λ_{\max} is the value at which the roots of a characteristic equation start to have a magnitude greater than 1 [29], and $\rho(\cdot)$ denotes the spectral radius of matrix (\cdot) . An unconditionally stable time-domain finite-element solution [29], [30] permits the use of any large time step. One can make a choice of the time step solely from the perspective of accuracy. For the nonlinear circuit simulation, we utilize the SPICE-based criterion to choose the time step [31], [32]. A typical Δt_c for state-of-the-art nonlinear devices is from the order of 10^{-9} s to the order of 10^{-12} s, which is generally larger than that of the central-difference-based time-domain finite-element simulation of an on-chip linear network, but smaller than that permitted by an unconditionally stable time-domain finite-element method. One can employ two different time steps for nonlinear circuit simulation and linear network simulation. For the numerical examples simulated in this paper, the time step determined by a central-difference-based time-domain finite-element method was used for the co-simulation of linear networks and nonlinear circuits.

VI. SIMULATION RESULTS

We simulated a number of ICs and package problems to demonstrate the accuracy and performance of the proposed

TABLE I
PARAMETERS OF THE TRANSISTOR

PMOS	NMOS
$L_{\text{eff}} = 0.1 \text{ } (\mu\text{m})$	$L_{\text{eff}} = 0.095 \text{ } (\mu\text{m})$
$W = 1.185 \text{ } (\mu\text{m})$	$W = 0.145 \text{ } (\mu\text{m})$
$k' = 5.303 \times 10^{-4} \text{ } (\text{A}/\text{V}^2)$	$k' = 1.177 \times 10^{-4} \text{ } (\text{A}/\text{V}^2)$
$\lambda = 0.1 \text{ } (1/\text{V})$	$\lambda = 0.06 \text{ } (1/\text{V})$
$I_0 = 8 \times 10^{-15} \text{ } (\text{A})$	$I_0 = 2.3 \times 10^{-15} \text{ } (\text{A})$
$V_T = 0.0258 \text{ } (\text{V})$	$V_T = 0.0258 \text{ } (\text{V})$
$V_{TO} = -0.4 \text{ } (\text{V})$	$V_{TO} = 0.43 \text{ } (\text{V})$
$V_{B0} = 0.8 \text{ } (\text{V})$	$V_{B0} = 0.8 \text{ } (\text{V})$
$C_{j0} = 2 \times 10^{-3} \text{ } (\text{pF})$	$C_{j0} = 2 \times 10^{-3} \text{ } (\text{pF})$
$C_{jsw0} = 5.13 \times 10^{-4} \text{ } (\text{pF})$	$C_{jsw0} = 5.4 \times 10^{-4} \text{ } (\text{pF})$
$FC = 0.5$	$FC = 0.5$
$C_{GB0} = 200 \text{ } (\text{pF}/\text{m})$	$C_{GB0} = 200 \text{ pF}/\text{m}$
$C_{GS0} = 40 \text{ } (\text{pF}/\text{m})$	$C_{GS0} = 40 \text{ pF}/\text{m}$
$C_{GD0} = 40 \text{ } (\text{pF}/\text{m})$	$C_{GD0} = 40 \text{ pF}/\text{m}$
$R_E = R_D = 0$	$R_E = R_D = 0$

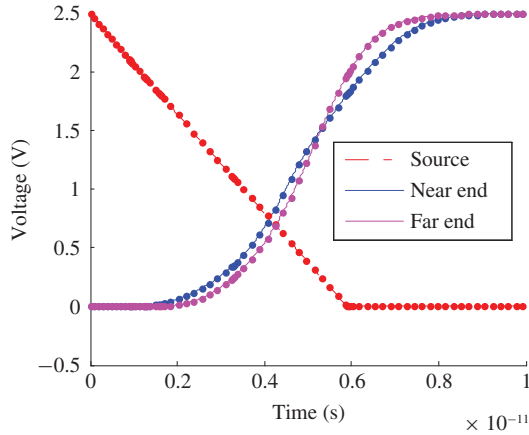


Fig. 7. Voltage simulated from a parallel-plate structure driven by an inverter and loaded by RC elements (solid lines are from the proposed method, stars are from SPICE).

we set up a lossless transmission line model to represent the structure. As can be seen clearly from Fig. 7, the proposed method for circuit simulation agrees very well with SPICE. During the time-marching process, the maximum number of Newton iterations was 7. The CPU time per step was 5×10^{-4} s with 1×10^5 total steps.

C. Lossy Parallel-Plate Structure Driven by a CMOS Inverter and Loaded by Lumped Elements

Next example was a lossy parallel-plate structure with the same dimension as simulated in the above. It was driven by the same inverter and loaded by the same resistors and capacitor. The thickness of the upper (lower) plates was $0.1 \text{ } \mu\text{m}$ and the metal conductivity was $5 \times 10^7 \text{ S}/\text{m}$. The dielectric between the two plates has a relative permittivity $\epsilon_r = 4$. In Fig. 8, we plot the voltages simulated from the proposed method at the near and far ends of the structure. Excellent agreement with

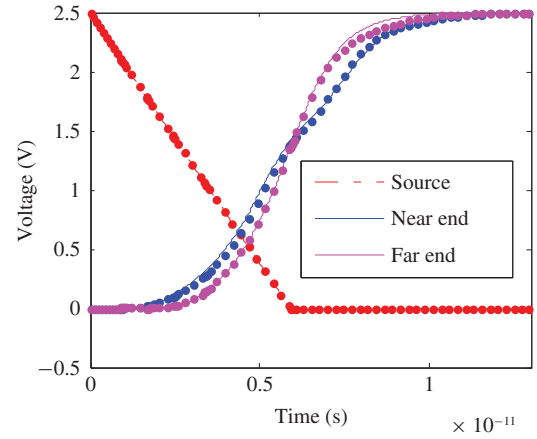


Fig. 8. Voltages simulated from a lossy parallel-plate interconnect driven by an inverter and loaded by RC elements (lines are results from the proposed method, stars are results from SPICE).

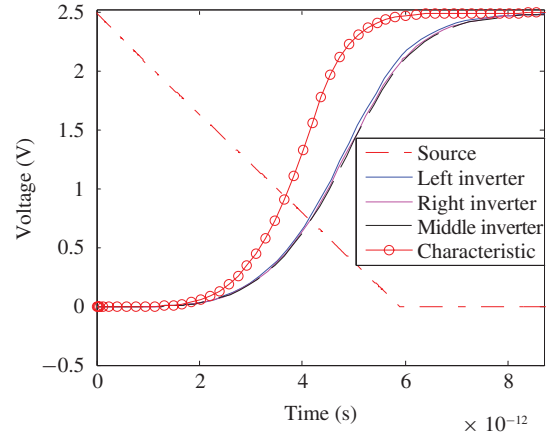


Fig. 9. Voltages simulated from a large-scale M4–M7 on-chip clock grid involving 58,800 inverters and an interconnect system of dimension 166,601,770.

SPICE can be observed. For comparison, we employed a lossy transmission line model to represent the structure for the use of SPICE simulation. In total, 1.3×10^5 time steps were simulated. Each step cost was 9×10^{-4} s. The maximum Newton iteration number used in the proposed simulation was 7.

D. Realistic M4–M7 Large-Scale On-Chip Clock-Grid Structure Occupying a Chip Area of $800 \times 900 \text{ } \mu\text{m}^2$

With the accuracy validated, next, we simulated a realistic large-scale on-chip clock-grid structure occupying a chip area of $800 \times 900 \text{ } \mu\text{m}^2$. The structure involved 420, 420, 210, and 210 interconnect wires in M4, M5, M6, and M7 layer, respectively. In between, there are a massive number of vias connecting orthogonal wires at different metal layers. Below M4 layer, there were 58,800 inverters. Among the 420 M4 wires, every other M4 wire was driven by inverters. Along the length of each M4 wire, 280 inverters were connected. The VDD node of each inverter (illustrated in the left part of Fig. 6) is attached to 2.5 V, the ground node is attached to 0 V. The input node is excited by a falling edge shown by the red dashed

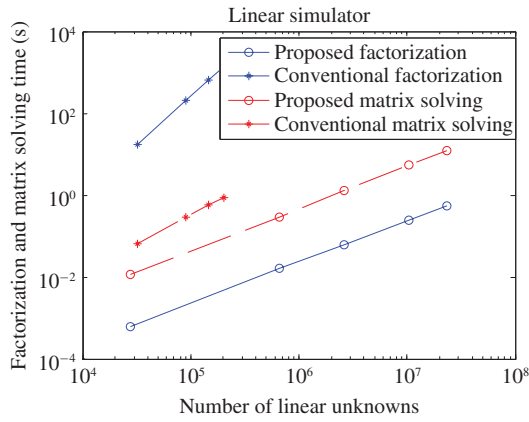


Fig. 10. CPU time of the proposed linear system solver versus number of unknowns in comparison with that cost by the conventional simulator.

line in Fig. 9. The output of the inverter is attached to the interconnect. Based on the algorithm proposed in Section III for nonlinear simulation, we formed 210 *decoupled* block tridiagonal systems, each of which has a form shown in (33c). Each block tridiagonal system had 280 diagonal blocks with the dimension of each block being 12. The discretization of the structure resulted in 166,601,770 linear unknowns. We used 24 cores for parallel simulation. Each core solved 2,450 nonlinear devices with a part of the linear structure. On average, only 7.19 s was used for one time step. The total number of time steps simulated was 1.7×10^5 . The width of the PMOS transistor used in this example was $0.85 \mu\text{m}$. In Fig. 9, we plot the output voltages of three inverters connected to three M4 wires in comparison with the characteristic output of the inverter. The three M4 wires are the leftmost, middle, and rightmost wire in M4, respectively. The maximum number of Newton iterations was 13 in this simulation.

E. Performance Test

We then tested the performance of the proposed first-principle-guided circuit simulator. First, we tested the sequential simulation. The sequential simulation includes the simulation of both nonlinear and linear system. To examine the performance of the linear system solver, we simulated the above clock-grid example occupying a chip area from 11.43×12.86 , 57.15×64.3 , 114.3×128.6 , 228.6×257.2 , to $342.9 \times 385.8 \mu\text{m}^2$, resulting in 27,742, 658,870, 2,618,230, 10,438,450, and 23,460,670 unknowns, respectively. To test the performance of the nonlinear solver, we used the linear structure having 23,460,670 linear unknowns with 226 interconnects in M4. The inverters were connected to M4 interconnects. Four cases were tested: 13,950 nonlinear devices driving 16 wires out of the 226 M4 interconnects; 67,950 nonlinear devices driving 76 wires; 135,450 nonlinear devices driving 151 wires; and 202,950 nonlinear devices driving all the 226 M4 wires.

In Fig. 10, we plot the matrix factorization time as well as matrix solution time versus the number of unknowns for the proposed linear system solver in comparison with the conventional linear simulator that employs the state-of-the-art

TABLE II
CPU TIME OF THE PROPOSED NONLINEAR SOLVER VERSUS THE NUMBER OF NONLINEAR DEVICES

Number of nonlinear devices	13, 950	67, 950	135, 450	202, 950
CPU time per step (s)	0.322	1.567	2.921	4.454

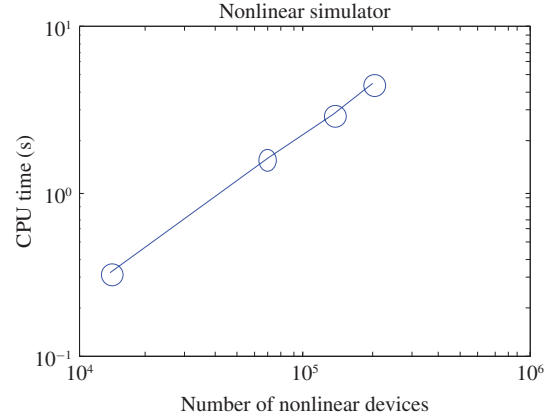


Fig. 11. Total CPU time of the proposed nonlinear system solver versus number of nonlinear devices.

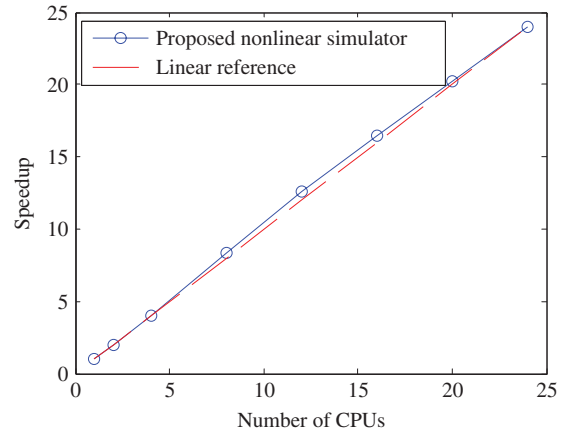


Fig. 12. Speedup of the parallelized circuit simulator.

sparse matrix solver such as a multi-frontal-based one [35]. The linear complexity and superior performance of the proposed linear system solver are clearly demonstrated. The proposed simulator costs less time in factorization than in matrix solving because in a sequential simulation, the surface-unknown-based system can be solved without factorization due to orthogonal vector bases. The CPU cost of the conventional linear simulator was not plotted across the entire range because the conventional simulator was not able to solve a larger number of unknowns due to large memory requirements. In Fig. 11, we plot the total CPU time of the proposed nonlinear simulator at each time step versus the number of nonlinear devices, from which the linear complexity of the proposed nonlinear simulator can also be clearly seen. Table II provided detail data of Fig. 11.

We then tested the performance of the proposed parallelization scheme. The large-scale clock-grid structure involving

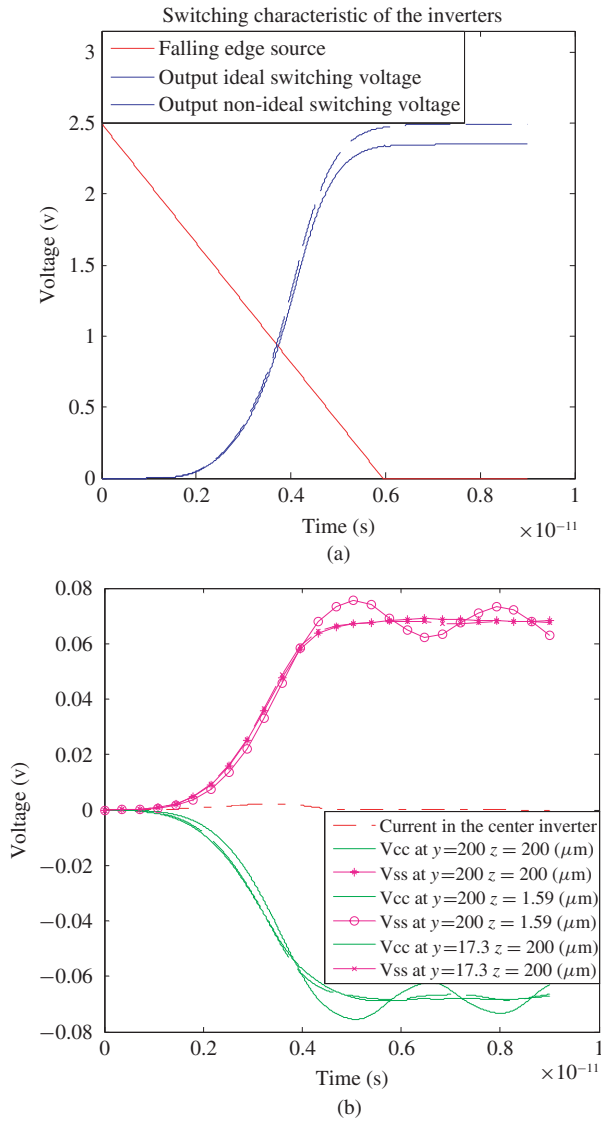


Fig. 13. Simulation of an M3–M8 on-chip power grid driven by 420 inverters involving over 162 million unknowns. (a) Input and output of a single inverter with an ideal power supply and a nonideal one. (b) Current drawn by one inverter and position-dependent transient VCC and VSS voltage droops observed in the power grid.

166, 601, 770 linear unknowns simulated for Fig. 9 was considered. The 420 M4 wires were all connected with nonlinear devices. In total, 816, 680 inverters were simulated. In Fig. 12, we plot the speedup versus the number of CPUs, from which a clear linear speedup is observed.

F. M3–M8 On-Die Power Grid of $400 \times 400 \mu\text{m}^2$ Chip Area

In addition to clock-grid analysis, the proposed simulator can also be used for many other analyses such as power-grid analysis and die-package co-simulation. We considered a $400 \times 400 \mu\text{m}^2$ on-die power grid at 90 nm technology node from M3 to M8, which was provided by Intel Corporation. There were 55 pairs of VCC (power rails) and VSS (ground rails) on M3, among which 21 pairs were attached to inverters. Each VCC and VSS pair was attached to 20 inverters uniformly distributed along the length of the power

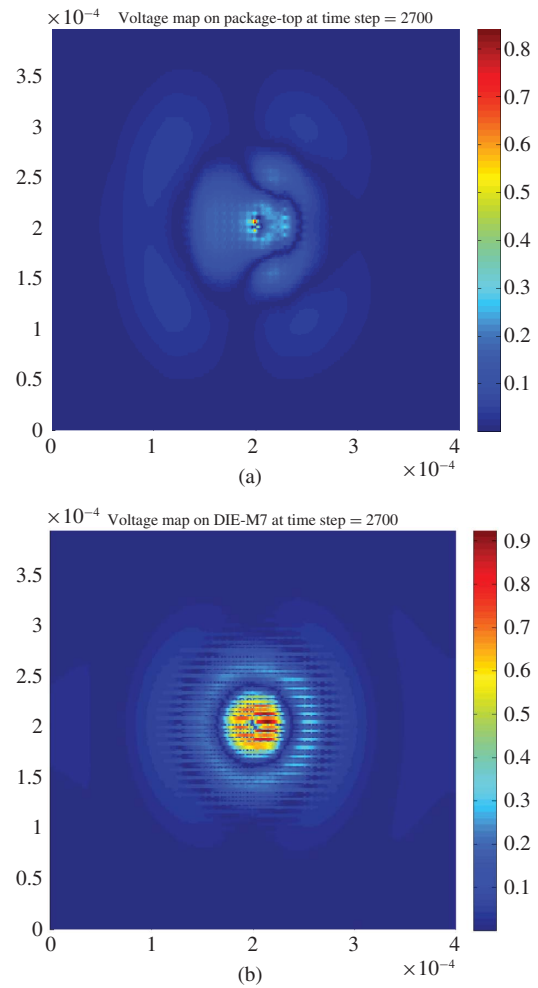


Fig. 14. Simulation of a combined die-package system from M2 to M8 to full 17 layers of package involving over 333 million unknowns. (a) Voltage map on the top layer of the package. (b) Voltage map on M7.

rail. In total, there were 420 inverters. In each inverter, the source node of the PMOS is attached to VCC, and the source node of the NMOS is attached to VSS. The DC bias is 2.5 V. A falling edge is applied to the input node, while the output node is left open. On the top, the M8 layer was connected to a package plane (treated as the potential reference) via four C4 VCC bumps and four C4 VSS bumps.

Due to the high integration density of the on-chip interconnects, the discretization resulted in 162, 316, 441 unknowns. The CPU time cost at each time step was less than 100 s on a single core. When running on multi-cores from 2 cores to 24 cores, the proposed circuit simulator also exhibits a clear linear speedup. In Fig. 13(a), we plot the input signal of an inverter located at the center of the chip and the output voltage of this inverter. The output voltage due to an ideal power supply and that resulting from the nonideal power supply provided by the power grid when all of the 420 inverters were switching are both shown. The effect of the nonideal power supply can be clearly seen. The voltage is calculated with respect to the package plane on the top. In Fig. 13(b), we plot the current in amperes drawn by the inverter located at the center of the

chip area. We also plot the voltage of the VCC and that of VSS sampled in M3 layer at $y = 200 \mu\text{m}$ and $z = 200 \mu\text{m}$ (center), $y = 200 \mu\text{m}$ and $z = 1.59 \mu\text{m}$, $y = 17.3 \mu\text{m}$ and $z = 200 \mu\text{m}$, where y the direction of M4 power rails, and z is the direction along M3 power rails. A dynamic voltage droop and the symmetry between VCC droop and VSS droop can be clearly observed.

G. Combined Die-Package With M2–M8 On-Die Structures and 17 Package Layers

The last example is a combined die-package structure that involved a complete on-die power grid from M2 to M8 and a complete 17-layer package for power delivery. The chip size was $400 \times 400 \mu\text{m}^2$. The discretization of the structure resulted in 333, 182, 390 unknowns. The total CPU time cost at each time step, i.e., one matrix solve, was less than 200 s on a single core. In Fig. 14, we plotted the voltage map at the top layer of the package and that on M7 sampled at one time instant.

VII. CONCLUSION

This paper presents a circuit simulator that has linear or optimal complexity. With an almost embarrassingly parallel implementation, the capacity of the simulator grows exponentially as the number of cores on a single processor chip grows exponentially. The proposed circuit simulator can be used to account for the full-chip (and beyond full-chip) interactions between nonlinear devices, substrate, on-die interconnects, and package. Because it rigorously captures electromagnetic physics, it can be used to guide the design of digital, analog, RF, and mixed-signal ICs from DC to very high frequencies. In addition to being used as a standalone circuit simulator, the proposed simulator can also directly interface with SPICE-like simulators in time domain to retain the strengths of device-centered simulators in simulating nonlinear devices.

ACKNOWLEDGMENT

The authors would like to thank S. Chakravarty and W. Shi at Intel Corporation, Santa Clara, CA, and C.-K. Koh at Purdue University, West Lafayette, IN, for providing valuable suggestions to this paper.

REFERENCES

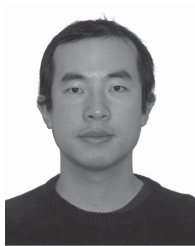
- [1] L. W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," UC Berkeley, Berkeley, CA, Tech. Rep. ERL-M520, May 1975.
- [2] T.-H. Chen, C. Luk, H. Kim, and C. C.-P. Chen, "INDUCTWISE: Inductance-wise interconnect simulator and extractor," in *Proc. Int. Conf. Comput. Aided Des.*, 2002, pp. 215–220.
- [3] J. Jain, C.-K. Koh, and V. Balakrishnan, "Fast simulation of VLSI interconnects," in *Proc. Int. Conf. Comput. Aided Des.*, San Jose, CA, Nov. 2004, pp. 93–98.
- [4] Z. Zhu, H. Peng, K. Rouz, M. Borah, C. Cheng, and E. Kuh, "Two-stage Newton-Raphson method for transistor level simulation," *IEEE Trans. Comput.-Aided Des.*, vol. 26, no. 5, pp. 881–895, May 2007.
- [5] H. Thornquist, E. Keiter, R. Hoekstra, D. Day, and E. Boman, "A parallel preconditioning strategy for efficient transistor-level circuit simulation," in *Proc. Int. Conf. Comput. Aided Des.*, 2009, pp. 410–417.
- [6] K. Sun, Q. Zhou, K. Mohanram, and D. Sorensen, "Parallel domain decomposition for simulation of large-scale power grids," in *Proc. Int. Conf. Comput. Aided Des.*, 2007, pp. 54–59.
- [7] G. Amdahl, "Validity of the single processor approach to achieving large-scale computing capabilities," in *Proc. AFIPS Conf.*, 1967, pp. 483–485.
- [8] M. Picket-May, A. Taflove, and J. Baron, "FDTD modeling of digital signal propagation in 3-D circuits with passive and active loads," *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 8, pp. 1514–1523, Aug. 1994.
- [9] P. Ciampolini, P. Mezzanotte, L. Roselli, and R. Sorrentino, "Accurate and efficient circuit simulation with lumped-element FDTD technique," *IEEE Trans. Microw. Theory Tech.*, vol. 44, no. 12, pp. 2207–2215, Dec. 1996.
- [10] C.-N. Kuo, B. Houshmand, and T. Itoh, "Full-wave analysis of packaged microwave circuits with active and nonlinear devices: An FDTD approach," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 5, pp. 819–826, May 1997.
- [11] M. Feliziani and F. Maradei, "Modeling of electromagnetic fields and electrical circuits with lumped and distributed elements by the WETD method," *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1666–1669, May 1999.
- [12] M. Feliziani and F. Maradei, "Circuit-oriented FEM: Solution of circuit field coupled problems by circuit equations," *IEEE Trans. Magn.*, vol. 38, no. 3, pp. 965–968, Mar. 2002.
- [13] K. Guilloard, M. F. Wong, V. F. Hanna, and J. Citerne, "A new global time-domain electromagnetic simulator of microwave circuits including lumped elements based on finite-element method," *IEEE Trans. Microw. Theory Tech.*, vol. 47, no. 10, pp. 2045–2048, Oct. 1999.
- [14] S.-H. Chang, R. Coccioli, Y. Qian, and T. Itoh, "A global finite-element time domain analysis of active nonlinear microwave," *IEEE Trans. Microw. Theory Tech.*, vol. 47, no. 12, pp. 2410–2416, Dec. 1999.
- [15] H. Tsai, Y. Wang, and T. Itoh, "An unconditionally stable extended (USE) finite-element time-domain solution of active nonlinear microwave circuits using perfectly matched layers," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 10, pp. 2226–2232, Oct. 2002.
- [16] R. Wang and J. M. Jin, "A symmetric electromagnetic-circuit simulator based on the extended time-domain finite element method," *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 12, pp. 2875–2884, Dec. 2008.
- [17] Q. He and D. Jiao, "Co-simulation of linear electromagnetic structures and non-linear devices in the time-domain finite-element reduction-recovery method," in *Proc. IEEE Int. Symp. Antennas Propag.*, Jun. 2009, pp. 1–4.
- [18] Q. He and D. Jiao, "Fast electromagnetics-based co-simulation of linear network and nonlinear circuits for the analysis of high-speed integrated circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 58, no. 12, pp. 3677–3687, Dec. 2010.
- [19] C. Yang and V. Jandhyala, "A time-domain surface integral technique for mixed electromagnetic and circuit simulation," *IEEE Trans. Adv. Packag.*, vol. 28, no. 4, pp. 745–753, Nov. 2005.
- [20] A. E. Yilmaz, J. M. Jin, and E. Michielssen, "A parallel FFT-accelerated transient field-circuit simulator," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 9, pp. 2851–2865, Sep. 2005.
- [21] S. M. S. Imtiaz and S. M. El-Ghazaly, "Physical simulation of complete millimeter-wave amplifiers using full-wave FDTD technique," in *Proc. IEEE MTT-S Int. Microw. Symp.*, Denver, CO, Jun. 1997, pp. 79–82.
- [22] P. Ciampolini, L. Roselli, G. Stopponi, and R. Sorrentino, "Global modeling strategies for the analysis of high-frequency integrated circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 47, no. 6, pp. 950–955, Jun. 1999.
- [23] D. Jiao, C. Dai, S.-W. Lee, T. R. Arabi, and G. Taylor, "Computational electromagnetics for high-frequency IC design," in *Proc. IEEE Int. Symp. Antennas Propag.*, Jun. 2004, pp. 3317–3320.
- [24] S. Grivet-Talocia, I. S. Stievano, and F. G. Canavero, "Hybridization of FDTD and device behavioral-modeling techniques [interconnected digital I/O ports]," *IEEE Trans. Electromagn. Compat.*, vol. 45, no. 1, pp. 31–42, Feb. 2003.
- [25] 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.
- [26] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, Jun. 1975.
- [27] D. Chen and D. Jiao, "Time-domain orthogonal finite-element reduction-recovery (OrFE-RR) method for electromagnetics-based analysis of large-scale integrated circuit and package problems," *IEEE Trans. Comput.-Aided Des.*, vol. 28, no. 8, pp. 1138–1149, Aug. 2009.
- [28] D. Chen, D. Jiao, and C.-K. Koh, "A parallel time-domain finite-element simulator of linear speedup and electromagnetic accuracy for the simulation of die-package interaction," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 1, no. 5, pp. 752–759, May 2011.

- [29] D. Jiao and J. M. Jin, "A general approach for the stability analysis of time-domain finite element method," *IEEE Trans. Antennas Propag.*, vol. 50, no. 11, pp. 1624–1632, Nov. 2002.
- [30] H. Gan and D. Jiao, "An unconditionally stable time-domain finite element method of significantly reduced computational complexity for large-scale simulation of IC and package problems," in *Proc. IEEE 18th Conf. Electr. Perform. Electron. Packag.*, Oct. 2009, pp. 1–4.
- [31] K. G. Nichols, T. J. Kazmierski, M. Zwolinski, and A. D. Brown, "Overview of SPICE-like circuit simulation algorithms," *IEE Proc.-Circuits Device Syst.*, vol. 141, no. 4, pp. 242–250, Aug. 1994.
- [32] K. S. Kundert, *The Designer's Guide to SPICE and Spectre*. Norwell, MA: Kluwer, 1995, ch. 4.
- [33] *The MOSIS Service* [Online]. Available: <http://www.mosis.com/Technical/Testdata/ibm-90-prm.html>
- [34] J. Rabaey, *Digital Integrated Circuit: A Design Perspective*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2003, ch. 5.
- [35] *UMFPACK5.0* [Online]. Available: <http://www.cise.ufl.edu/research/sparse/umfpack/>
- [36] 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.
- [37] D. A. White, "Orthogonal vector basis functions for time domain finite element solution of the vector wave equation," *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1458–1461, May 1999.



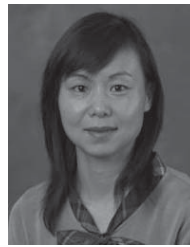
Qing He received the B.S. degree in electronic and information engineering from Zhejiang University, Hangzhou, China, in 2006, and the M.S. degree from the Graduate School of Chinese Academy of Sciences, Beijing, China, from 2006 to 2007. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering and the On-Chip Electromagnetics Group, Purdue University, West Lafayette, IN.

He was a Research Assistant with the Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing, from 2006 to 2007. His current research interests include computational electromagnetics, high-performance very large scale integration computer aided design, fast- and high-capacity numerical methods.



Duo Chen received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

He is a Research Assistant with the On-Chip Electromagnetics Research Group, West Lafayette. His current research interests include electromagnetic-based analysis of very large scale integration and package problems.



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

She was with Technology Computer Aided Design (CAD) Division, Intel Corporation, Santa Clara, CA, until September 2005, as a Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. In September 2005, she joined the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, as an Assistant Professor, where she is

currently an Associate Professor with tenure. She has authored two book chapters and over 140 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics, high frequency digital, analogue, and mixed signals, radio frequency-integrated circuit (RF IC) design and analysis, high-performance very large scale integration (VLSI) CAD, modeling of micro- and nano-scale circuits, applied electromagnetics, fast and high-capacity numerical methods, fast time-domain analysis, scattering and antenna analysis, RF, microwave, millimeter wave circuits, wireless communication, and bioelectromagnetics.

Dr. Jiao was among 100 engineers chosen for the National Academy of Engineering's U.S. Frontiers of Engineering Symposium in 2011. She received the Ruth and Joel Spira Outstanding Teaching Award in 2010, National Science Foundation CAREER Award in 2008, and the Jack and Cathie Kozik Faculty Startup Award, which recognizes an outstanding new Faculty Member in Purdue Electrical and Computer Engineering in 2006, an ONR Award through Young Investigator Program in 2006, and 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 2004. 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 in 2003. She was also awarded the Intel Technology CAD Divisional Achievement Award for the development of innovative full-wave solvers for high-frequency IC design and the Intel Hero Award (Intel-wide she was the tenth recipient) by Intel Components Research for the timely and accurate 2- and 3-D full-wave simulations in 2002. She 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 Raj Mittra Outstanding Research Award by the University of Illinois at Urbana-Champaign in 2000. She has served as a reviewer for many IEEE journals and conferences.