Parallel Time-Domain Finite-Element Simulator of Linear Speedup and Electromagnetic Accuracy for the Simulation of Die–Package Interaction

Duo Chen, Dan Jiao, *Senior Member, IEEE*, and Cheng-Kok Koh, *Senior Member, IEEE*

***Abstract*— In this paper, we develop an almost embarrassingly parallel solution to an electromagnetic solver of linear complexity for overcoming the grand challenge of performing electromagnetically accurate co-simulation of die–package interaction. In this solution, through suitable basis functions and linear algebraic techniques, we directly and rigorously decompose the system matrix in a 3-D space to multiple matrices of 1-D sizes with negligible computational overhead. Each 1-D matrix is made tridiagonal and, hence, can be solved readily in linear complexity. We then achieve an almost embarrassingly parallel implementation of the fast electromagnetic solver with a low communication-to-computation ratio. Numerical experiments on a large-scale combined die–package system, involving more than 3.5 billion unknowns, have demonstrated superior performance of the proposed parallel transient simulator for simulating large-scale integrated circuits and package problems. In addition, the proposed solver is applicable to any arbitrarily shaped multilayer structure embedded in inhomogeneous materials.**

***Index Terms*— Electromagnetic analysis, finite-element methods, multicore, on-chip, package, parallel computing, transient simulation.**

I. Introduction

THE scaling of supply voltages and the increased level of integration have made the analysis and design of microelectronic systems increasingly challenging. The impact of noise due to signal switching, die–package interaction, power management techniques, substrate coupling, etc. can be seen at all levels of a power delivery network, from chip to package to mother board to the voltage regulator module. There is, therefore, a critical need for the co-simulation of the die and package to control the global electrical interaction and optimize performance as an integrated system. Prevailing circuit-based signal integrity paradigms are reaching their limits of predictive accuracy when applied to high-frequency settings. To sustain the scaling and integration of digital, analog, mixed-signal, and radio frequency (RF) circuitry, an electromagnetic solution is indispensable.

Manuscript received August 4, 2010; revised December 23, 2010; accepted January 14, 2011. Date of publication March 17, 2011; date of current version June 2, 2011. This work was supported in part by a grant from the 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. Recommended for publication by Associate Editor J. Tan 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: djiao@purdue.edu; chen256@purdue.edu; chengkok@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.2011.2111418

The co-simulation of the full chip and complete package results in numerical problems of ultra large scale, requiring billions and billions of parameters to describe them accurately. None of the existing computational electromagnetic methods has the capacity to simulate such an ultra large-scale system in fast CPU run time. Recent research on electromagnetic simulators of reduced computational complexity has brought the die–package co-simulation closer to reality. As many-core computing has become a new form of equivalent scaling to facilitate the continuation of Moore's Law, the co-simulation of full chip and complete package with uncompromised accuracy can be brought to reality in faster CPU run time 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 [1]. Linear (or optimal) speedup can be achieved only if the computation is embarrassingly parallel.

Typically, it is easier to obtain superior speedup when parallel computing is applied to a high-complexity algorithm. Indeed, it is more difficult to achieve speedup over a good (optimized) sequential algorithm. For example, it is easy to parallelize a naïve dense matrix-vector multiplication of $O(N^2)$ complexity, it is difficult, however, to parallelize a fast dense matrix-vector multiplication that has $O(N)$ complexity, with N being the matrix size. However, for high-complexity algorithms, even if one is able to achieve linear speedup, the resultant CPU time may still be too long to be tolerated in realistic integrated circuit (IC) and package design.

In recent years, efforts have been made in parallelizing fast computational electromagnetic methods in order to tackle large-scale engineering problems [2]–[9]. Both integral-equation-based and partial-differential-equation-based methods have been parallelized. Significant speedups have been achieved compared to their sequential counterparts. However, none of the existing fast electromagnetic solvers involving a linear matrix solution has achieved linear speedup.

The main contribution of this paper is the development of an almost embarrassingly parallel solution to a computational electromagnetic method of linear complexity for large-scale co-simulation of die–package interaction. The sequential algorithm is based on the orthogonal finite-element reduction-recovery (OrFE-RR) method developed in [10]. The OrFE-RR method is developed in the framework of a time-domain finite-element method [11]–[15]. In the OrFE-RR method, a linear system of equations, $\mathbf{A}x = b$, is *rigorously*

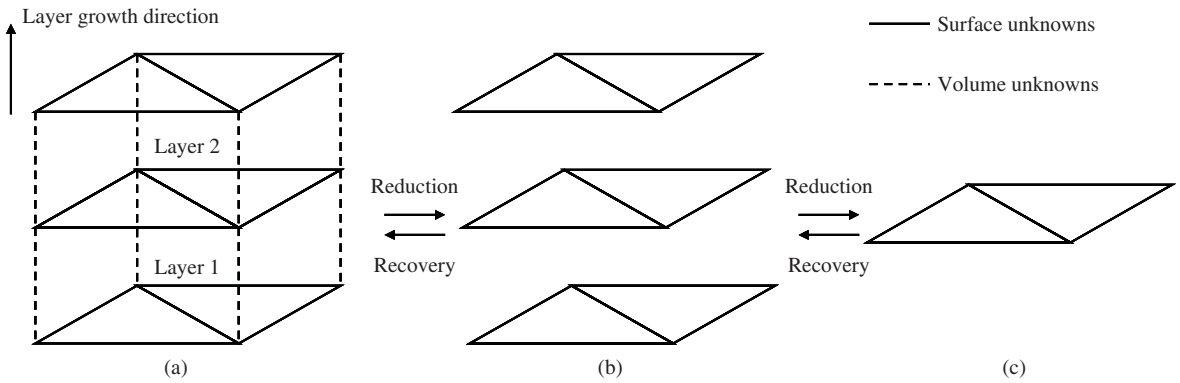

Fig. 1. Illustration of the reduction-recovery method in [10]. (a) 3-D layered system. (b) 2-D layered system. (c) Single-layer system.

reduced to an orders of magnitude smaller system, $\mathbf{A}_{\mathrm{r}} x_{\mathrm{r}} = b_{\mathrm{r}}$, with x_{r} being a subset of x.The reduction from $\mathbf{A}$ to $\mathbf{A}_{\mathrm{r}}$ is achieved with negligible computational cost by the development of a set of orthogonal prism vector basis functions. The reduced system, $\mathbf{A}_{\mathrm{r}} x_{\mathrm{r}} = b_{\mathrm{r}}$, is diagonal and, hence, can be solved in optimal complexity, i.e., linear complexity. The x_{r} solved from the reduced system is the same as that solved from the original system, i.e., $\mathbf{A}x = b$. Additionally, the rest of the unknown x can be recovered from x_{r} in linear complexity. The method entails no theoretical approximation. It applies to any arbitrarily shaped multilayer structure involving inhomogeneous materials. This method has been successfully applied to solve large-scale IC and package problems. For example, it can solve a sparse matrix involving more than 330 million unknowns associated with a large-scale die–package system in less than 200 s on a single 2.66-GHz Intel Xeon 5300 processor.

The recursive nature of the OrFE-RR algorithm renders its parallelization difficult. In this paper, we show how to utilize the orthogonal vector basis functions developed in the OrFE-RR method to achieve a natural decomposition of the system matrix into 1-D matrices, with each 1-D matrix being tridiagonal. Not only can we solve the large-scale IC and package problems in linear complexity, but we can also parallelize the OrFE-RR solution to achieve linear speedup. With both linear complexity and linear speedup achieved, we pave the way for the co-simulation of a full chip combined with a complete package. In addition to the co-simulation of die–package interaction, the proposed solver is applicable to the co-simulation of digital, analog, mixed-signal, and RF circuitry integrated on a single chip.

II. Electromagnetics-Based Analysis of IC and Package Problems

Consider electromagnetics-based analysis of a 3-D IC and package problem. The electric field $\mathbf{E}$ inside an IC and package problem satisfies the second-order vector wave equation

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

subject to certain boundary conditions. In (1), μ_r is relative permeability, μ_0 is free-space permeability, ε is the permittivity, and σ is the conductivity.

Among all the existing time-domain methods for solving (1), time-domain finite-element methods [11] have shown a great capability in handling both irregular geometries and arbitrary inhomogeneity. A time-domain finite-element solution of (1) results in the following system of ordinary differential equations:

$$\mathbf{T}\frac{d^2u}{dt^2} + (\mathbf{R} + \mathbf{V})\frac{du}{dt} + \mathbf{S}u + f = 0 \tag{2}$$

in which $\mathbf{T}$, $\mathbf{R}$, $\mathbf{V}$, $\mathbf{S}$ are square matrices, u is the unknown field vector, and f is the excitation vector. The elements of the matrices $\mathbf{T}$, $\mathbf{R}$ and $\mathbf{S}$ are given by

$$\begin{aligned} \mathbf{T}_{ij} &= \mu_0 \varepsilon < \mathbf{N}_i, \mathbf{N}_j >_V \\ \mathbf{R}_{ij} &= \mu_0 \sigma < \mathbf{N}_i, \mathbf{N}_j >_V \\ \mathbf{S}_{ij} &= \mu_r^{-1} < \nabla \times \mathbf{N}_i, \ \nabla \times \mathbf{N}_j >_V \end{aligned} \tag{3}$$

in which $\mathbf{N}_i$ and $\mathbf{N}_j$ are the vector basis functions used to expand unknown field **E,** and $< ., . >_V$ denotes volume integration. Matrix $\mathbf{V}$ is related to the absorbing boundary condition. By adopting a central difference scheme to approximate the first- and second-order time derivatives in (2), we obtain

$$\tilde{\mathbf{T}} u^{n+1} = (2\mathbf{T} - \Delta t^2 \mathbf{S}) u^n + [0.5 \Delta t (\mathbf{R} + \mathbf{V}) - \mathbf{T}] u^{n-1} - \Delta t^2 f^n \tag{4}$$

where

$$\tilde{\mathbf{T}} = \mathbf{T} + 0.5 \Delta t (\mathbf{R} + \mathbf{V}) \tag{5}$$

and Δt represents the time step. Obviously, u^{n+1} [the field value at the $(n+1)$th time step] can be solved in a time-marching fashion from the solution of u at previous time steps.

Matrix $\tilde{\mathbf{T}}$ can be very large in an electromagnetics-based simulation of realistic integrated circuits and packages. In addition, it is a general sparse matrix, the solution of which has been regarded as not embarrassingly parallelizable. In the following section, we will show a method that can make the solution of $\tilde{\mathbf{T}}$ embarrassingly parallelizable, thus permitting an electromagnetically accurate simulation of very large scale IC and package problems.

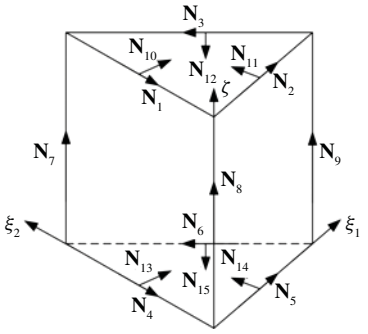

Fig. 2. Illustration of the orthogonal prism vector basis functions.

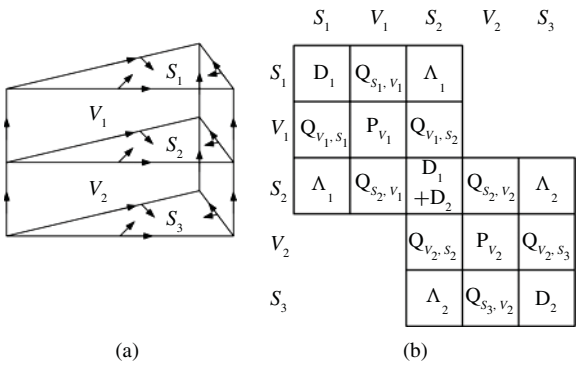

Fig. 3. (a) Orthogonal prism-element-based discretization and unknown ordering. (b) 3-D layered system matrix $\tilde{\mathbf{T}}$.

III. Modified Orthogonal Finite-Element Reduction-Recovery Method That is Embarrassingly Parallelizable

In the OrFE-RR method [10], a 3-D layered system matrix shown in Fig. 1(a) is reduced to a 2-D layered one, as shown in Fig. 1(b), without any computational cost. The 2-D layered system matrix is then reduced to a single-layered one, as shown in Fig. 1(c), with negligible computational overhead. This is achieved by developing a set of orthogonal prism vector basis functions. The reduced single-layer system is diagonal and, hence, can be solved readily. Once the unknowns on a single surface are solved, the unknowns on other surfaces, and the unknowns in the volume, are recovered in linear complexity. The accuracy and high capacity of this method have been demonstrated by both numerical and experimental results.

The recursive nature of the OrFE-RR algorithm renders its parallelization difficult. In this paper, we show how to utilize the orthogonal vector basis functions developed in the OrFE-RR method to achieve a natural decomposition of the system matrix into 1-D matrices, with each 1-D matrix being tridiagonal. Hence, not only can we solve $\tilde{\mathbf{T}}$ in (5) in linear complexity, but we can also parallelize its solution to achieve linear speedup.

A. Field Expansion by Orthogonal Prism Vector Bases

The orthogonal prism vector bases developed in [10] are used to expand the unknown $\mathbf{E}$ when solving (1). In each prism element, there are 15 vector bases from $\mathbf{N}_1$ to $\mathbf{N}_{15}$, as shown in Fig. 2. The degrees of freedom associated with bases $\mathbf{N}_i$ $(i = 1, 2, 3, \ldots, 6)$ are assigned on the upper and lower planes. Each of these bases is purely tangential to edge i at the midpoint of edge i. The degrees of freedom associated with bases $\mathbf{N}_i$ $(i = 7, 8, 9)$ are assigned along the vertical edges of a prism element. They also guarantee the tangential field continuity across the element interface. As shown in [10], basis functions $\mathbf{N}_i$ $(i = 1, 2, 3, \ldots, 6)$ are orthogonal to each other, $\mathbf{N}_i$ $(i = 10, 11, 12, \ldots, 15)$ are orthogonal to each other, and these two sets are also mutually orthogonal to each other. Additionally, because the bases on the lower and upper planes are perpendicular to the vector bases along the vertical edges, basis functions $\mathbf{N}_i$ $(i = 1, 2, 3, \ldots, 6)$ and $\mathbf{N}_i$ $(i = 10, 11, 12, \ldots, 15)$ are orthogonal to basis functions $\mathbf{N}_i$ $(i = 7, 8, 9)$.

B. Computation-Free Decomposition of the System Matrix into a Surface-Unknown-Based System and a Volume-Unknown-Based One

We discretize the computational domain into layers of prism elements, as shown in Fig. 3(a). Conductors are also discretized in order to capture the internal fields accurately. In each element, the electric field is expanded into orthogonal vector basis functions, as described in the previous section. The unknowns are ordered layer by layer. In each layer, the unknowns are divided into surface and volume unknowns. As shown in Fig. 3(a), the unknowns associated with the horizontal edges are surface unknowns, and the unknowns associated with the vertical edges are volume unknowns. The unknowns are then ordered from S_1 to V_1 to S_2 to V_2 and continued, resulting in a 3-D layered system matrix shown in Fig. 3(b). Although only two elements are shown in Fig. 3(a) for the convenience of illustration, the matrix structure resulting from a many-element discretization remains the same as that shown in Fig. 3(b).

Since the surface vector bases $\mathbf{N}_i$ $(i = 1$–$6, 10$–$15)$ are perpendicular to volume vector bases $\mathbf{N}_i$ $(i = 7$–$9)$ as seen from Fig. 2, and $\tilde{\mathbf{T}}$ in (5) solely comprises matrices formed by the inner product of $\mathbf{N}_i$ and $\mathbf{N}_j$, all the $\mathbf{Q}$ matrices in Fig. 3(b) vanish. As a result, the surface-unknown-based system is completely decoupled from the volume-unknown-based without any computation cost as follows:

$$\begin{aligned} \mathbf{P}_S x_S &= b_S \\ \mathbf{P}_{Vl} x_{Vl} &= b_{Vl} \quad l = 1, 2, \ldots, L \end{aligned} \tag{6}$$

in which $\mathbf{P}_S$ is the surface-unknown-based system, $\mathbf{P}_{Vl}$ is the volume-unknown-based system in layer l, and x_S and x_V represent the surface unknowns and volume unknowns, respectively. Note that (6) does not suggest that there is no coupling between volume unknowns and surface unknowns, they couple to each other through the right-hand side b at each time step, as can be seen from (4). The right-hand sides b_S and b_{Vl} in (6) denote the right-hand side of (4) corresponding to the surface unknowns, and volume unknowns respectively.

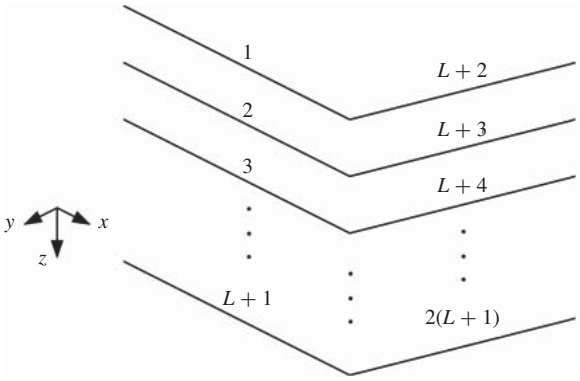

Fig. 4. Illustration of the new surface unknown ordering scheme (each solid straight line segment represents one edge basis).

C. Computation-Free Decomposition of the Surface-Unknown-Based System into 1-D matrices, with Each 1-D Matrix Tridiagonal

In $\mathbf{P}_s$, i.e., the surface-unknown-based part of Fig. 3(b), all $\mathbf{D}_l$ and Λ_l $(l = 1, 2, \ldots, L)$ matrices are diagonal due to the orthogonality of the vector basis functions. This matrix, however, is not amenable for parallelization. Next, we will show how we can permute $\mathbf{P}_s$ to make it block diagonal and, hence, embarrassingly parallelizable.

Assuming that the discretization results in N_s unknowns on a single surface, and L layers, and hence $L + 1$ surfaces. In the original OrFE-RR algorithm, we order N_s unknowns on the first surface, then N_s unknowns on the second surface, then N_s unknowns on the third surface, and continue to the N_s unknowns on the $(L + 1)$th surface. The resultant system is a block tridiagonal matrix with each block being diagonal. To facilitate parallelization, we change the ordering scheme as follows. Instead of ordering the N_s unknowns on a single surface layer by layer, we start from just *one* of the N_s unknowns on the first surface, we then order the counterpart of this single unknown on the second surface, then on the third surface, and continue to the last surface. We then go back to the first surface to order another surface unknown across all layers, we continue until all the N_s unknowns are traversed. Fig. 4 gives an illustration of the new ordering scheme.

As a result, we obtain a block diagonal matrix of N_s blocks, as shown in (7)

$$\mathbf{P}_S = \begin{bmatrix} \tilde{\mathbf{T}}_1 & & & \\ & \tilde{\mathbf{T}}_2 & & \\ & & \cdots & \\ & & & \tilde{\mathbf{T}}_{N_S} \end{bmatrix}. \tag{7}$$

More importantly, each diagonal block $\tilde{\mathbf{T}}_i$ $(i = 1, 2, \ldots, N_s)$ is a tridiagonal matrix of size $L + 1$

$$\tilde{\mathbf{T}}_i = \begin{bmatrix} \mathbf{D}_{1,i} & \Lambda_{1,i} & & \\ \Lambda_{1,i} & \mathbf{D}_{1,i} + \mathbf{D}_{2,i} & \Lambda_{2,i} & \\ & \cdots & \cdots & \Lambda_{L,i} \\ & & \Lambda_{L,i} & \mathbf{D}_{L+1,i} \end{bmatrix} \tag{8}$$

where $\mathbf{D}_{j,i}$ and $\Lambda_{j,i}$ denote the ith entries in matrices $\mathbf{D}_j$ and Λ_j, respectively. Essentially, the matrix $\mathbf{P}_S$ in (6) is decomposed into small matrices $\tilde{\mathbf{T}}_i$ of 1-D sizes. It is clear that such decomposition is computation-free. In addition, each $\tilde{\mathbf{T}}_i$ is a tridiagonal matrix. Its inverse can be stored and obtained efficiently in linear complexity [16]. The decoupled N_s subsystems can be readily distributed to a many-core/node platform to solve them in an embarrassingly parallel fashion.

D. Computation-Free Decomposition of the Volume-Unknown-Based System into 1-D matrices, with Each 1-D Matrix Tridiagonal

To facilitate parallelization, we split the volume-unknown-based system $\mathbf{P}_{Vl}$ as follows:

$$\mathbf{P}_{Vl} = \mathbf{P} - \mathbf{L} - \mathbf{U} \tag{9}$$

with $\mathbf{P}$ being the block diagonal matrix, $-\mathbf{L}$ the strict lower triangular part and $-\mathbf{U}$ the strict upper triangular part. Each diagonal block of $\mathbf{P}$ can be shown to be tridiagonal [10], and $\mathbf{L}$ and $\mathbf{U}$ are sparse matrices. The matrix $\mathbf{P}$ serves as an effective preconditioner of $\mathbf{P}_{Vl}$. It is shown by numerical experiments that, with $\mathbf{P}$, the iterative process of solving the second equation in (6) converges in a few iterations for realistic on-chip and package examples. By using $\mathbf{P}$ as the preconditioner, the second equation in (6) can be solved iteratively as

$$\mathbf{P}x^{(k+1)} = (\mathbf{L} + \mathbf{U})x^{(k)} + b_{Vl}, \quad k \geq 0 \tag{10}$$

with k being the iteration number. When (10) converges, $x^{(k+1)} = x^{(k)}$, and hence $(\mathbf{P}-\mathbf{L}-\mathbf{U})x^{(k+1)} = b_{Vl}$. From (9), it can be seen that $x^{(k+1)}$ is the solution of the volume equation in (6). It is proven in [10] that, because of the property of $\mathbf{P}_{Vl}$, the iteration in (10) converges for any right-hand side b_{Vl} and any initial vector $x^{(0)}$.

Since $\mathbf{P}_{Vl}$ is a block tridiagonal matrix, (10) can be simplified to

$$\mathbf{P}_{ii}x_i^{(k+1)} = \mathbf{L}_{i,i-1}x_{i-1}^{(k)} + \mathbf{U}_{i,i+1}x_{i+1}^{(k)} + d_i, \ 1 \leq i \leq N_x \tag{11}$$

in which i denotes the line index, as shown in [10 Fig. 6], and $\mathbf{P}_{ii}$ denotes the ith diagonal block of $\mathbf{P}$, this is formed by volume unknowns on line i. Since $\mathbf{P}_{ii}$ is a tridiagonal matrix, $\mathbf{P}_{ii}^{-1}b$ can again be performed in linear complexity. As a result, the decoupled $\mathbf{P}_{ii}$ can also be readily distributed to a many-core/node platform to solve them in an almost embarrassingly parallel fashion.

IV. Parallelization

In this section, we show how to parallelize the proposed fast electromagnetic solver, made embarrassingly parallelizable with a low communication-to-computation ratio.

A. Partition

Fig. 5 shows a 3-D layered system, which can be a combined chip–package system, assuming that z is the layer growth direction. We partition the system along x into n subdomains. The interface between two adjacent subdomains is parallel to the yz plane. Here, we assume that there are n lines (running along the y-direction), each of which has n volume unknowns. A uniform partitioning will therefore assign (n/P) lines to each node/core. There are, therefore,

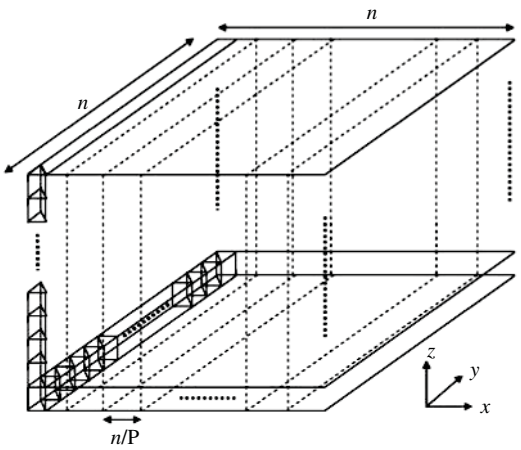

Fig. 5. Illustration of the partition scheme.

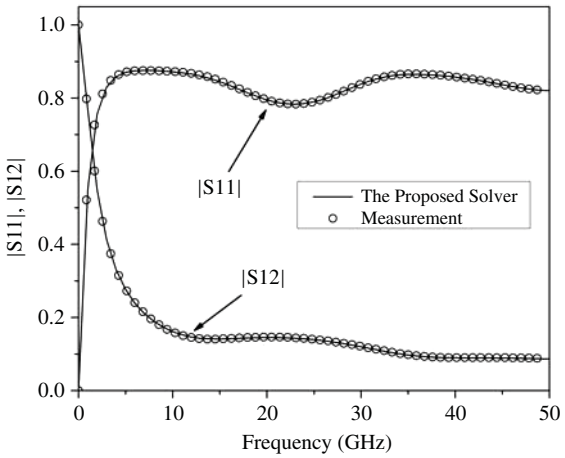

Fig. 6. Validation of the proposed parallel solver.

n^2/P volume unknowns to be computed in each node/core. As we observed from (16), only the two lines on the two vertical boundaries of a partition require volume unknowns that have been operated on by the two neighboring nodes/cores in the previous time step and in the previous iteration. The data transferred between two neighboring nodes/cores are therefore in the order of $O(n)$, resulting in a per-node/core communication-to-computation ratio of P/n, as the computation has a linear complexity.

B. Obtaining Right-Hand Side

Since the matrix structure is the same in each sudomain, we can perform the simulation in the same manner for all subdomains concurrently. The first task is to update the right-hand-side vector. We need to synchronize the field values on the interface between two adjacent subdomains after finishing the matrix–vector multiplications shown in the right-hand side of (4). Since the number of interface unknowns is very small compared to the unknowns in the whole system, the time spent on the communication between two adjacent nodes/cores is small. Further, if using a distributed system, the latency of the synchronization process can be hidden very well by computation. After obtaining the right-hand-side vector, we can simultaneously solve surface unknowns and volume unknowns in each subdomain, which is detailed in next two subsections.

C. Solving for Surface Unknowns

For the surface-unknown system within each sudomain, because the basis functions associated with surface edges are orthogonal, there exists no crosstalk between two surface edges in two different subdomains; hence the matrix within each subdomain is completely decoupled. Therefore, with the interface unknowns synchronized on the boundary between two adjacent subdomains, the process of solving surface unknowns is embarrassingly parallel across all the nodes/cores. The surface-unknown-based matrix handled by each node/core is a block diagonal one with each block tridiagonal, as shown in (7) and (8), and can be solved readily in linear complexity by each node/core.

D. Solving for Volume Unknowns

For the volume-unknown system, however, the parallelization procedure is more involved. Since the volume-unknown system is solved iteratively as shown in (11), based on the partition scheme shown in Fig. 5, we suffer from the crosstalk between two adjacent lines within a volume-unknown system. This is because we have to communicate the boundary data at each iteration step between two adjacent subdomains. Though the amount of data to be transferred at each iteration step is very small, we have to transfer the data for several times when solving for volume unknowns. The number of data transfer depends on the number of iterations. Since there are thousands of volume unknowns in a realistic IC and package structure, the total number of communications can be quite large. In addition, the computational complexity of solving the volume-unknown system is linear. As a result, the time spent on solving the volume unknowns is small in a sequential case. The total communication time of solving volume unknowns, therefore, cannot be neglected by relying on the partition scheme shown in Fig. 5 only. It was shown by our numerical experiments, in fact, that the volume-unknown communication time dominated the solution time. We therefore augmented the partition scheme as follows by developing an alternative approach to solve volume unknowns.

In orthogonal basis functions developed for triangular prism elements, there are 12 surface unknowns and 3 volume unknowns in each element. This indicates that the number of surface unknowns is greater than that of the volume unknowns. A typical ratio of the number of volume unknowns to the number of surface unknowns is 1:10 or even less. Aggregating the separated pieces of a volume-unknown system in one layer to an entire one on a distinct node/core, therefore, becomes feasible in this case. In fact, the communication time of the aggregation is negligible since the volume unknowns account for only a small portion of the entire system, the data aggregation only needs to be conducted twice, i.e., before and after solving the volume-unknown system. The amount of data to be transferred on each computer in this process can be calculated by LN_V/p, where N_V is the number of volume unknowns in one layer of the entire system, L is the

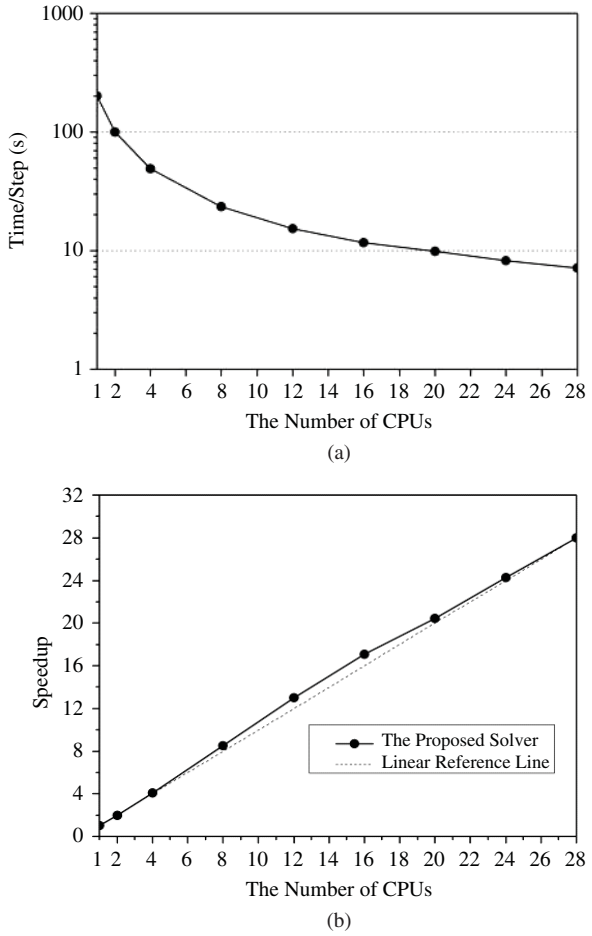

Fig. 7. (a) CPU time per step versus the number of CPUs. (b) Speedup versus the number of CPUs.

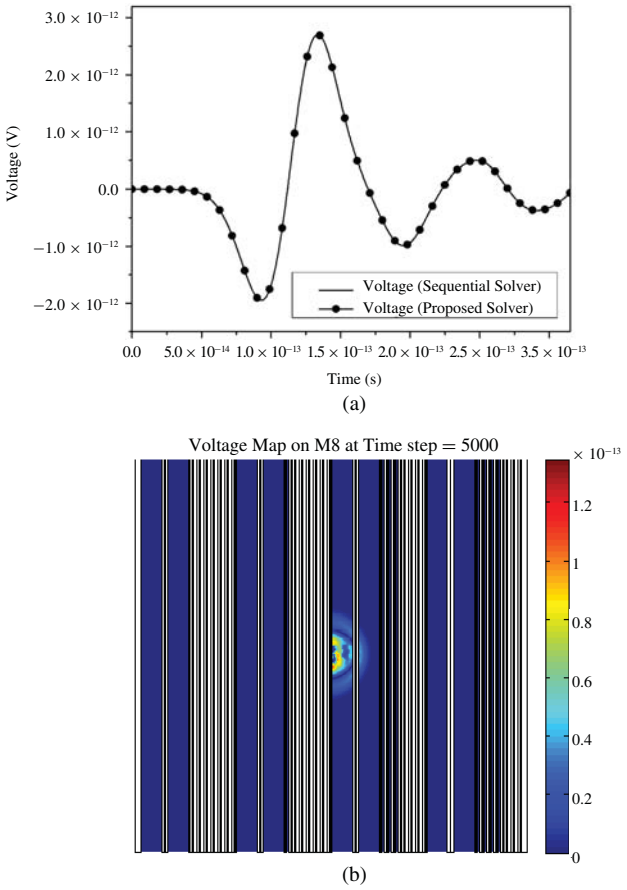

Fig. 8. Simulation of a 400×400 μm^2 combined die-package system from M2 to M8 to full package stack. (a) Waveforms of the output voltage. (b) Voltage map of the combined die-package system on M8 at a time instant.

number of layers, and P is the number of nodes/cores. We can see that, when increasing the number of nodes/cores, the amount of transferred data on each computer is decreased. Also, the amount of data to be transferred is trivial compared to the amount of data in the entire system, which is dominated by the surface unknowns. The communication time therefore is very small. In fact, since the surface-unknown system and the volume-unknown system are fully decoupled, they can be handled concurrently. We can use the computation in solving the surface-unknown system to overlap the communication of this process. Hence, the latency, which is small, can be very well hidden. The communication can be then neglected. It is worth mentioning that the separated volume-unknown system matrix parts are aggregated before the time marching, which only needs to be done once.

E. Overall Procedure

We update the right-hand-side vector within each subdomain and communicate the unknowns residing on the interface between two subdomains. The embarrassingly parallel matrix solution has two components: 1) solve the surface-unknown system in each subdomain assigned to each node/core, and 2) solve the volume-unknown system assigned to each node/core. There is no need to communicate data during the iteration. Before solving the volume-unknown system, we aggregate the volume unknowns in one layer and combine them in one distinct node/core. The latency can be hidden by the computation of solving surface unknowns. After solving the volume-unknown system, we redistribute the volume unknowns back to the subdomains assigned to each node/core. The communication time is small compared to the computation time. Steps 1) and 2) can be done concurrently.

V. Numerical Results

We implemented the proposed parallel transient simulator using message passing interface to fully utilize distributed machines in a cluster and multicores available in each machine. Each machine has the following hardware configuration.

CPU	Memory	Cache size
Intel(R) Xeon(R) CPU 5300 @ 2.66 GHz	32 GB	6144 KB

We first validated the proposed parallel electromagnetics-based transient simulator on a test-chip interconnect structure. The structure was fabricated using silicon processing technology on a test chip [17], [18]. The structure was of 300 μm width, and involved a 10-μm-wide strip in M2 layer, one ground plane in M1 layer, and one ground plane in M3 layer. The distance of this strip to the M2 returns at the left- and

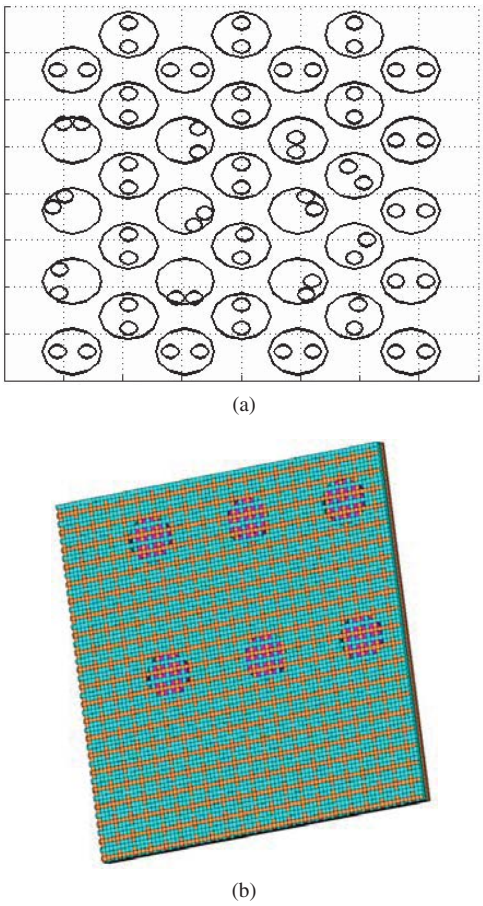

Fig. 9. Illustration of the die-package power delivery example. (a) Top view from the package side (big circles are holes; small circles are sampled via contacts). (b) Bottom view from the on-chip side.

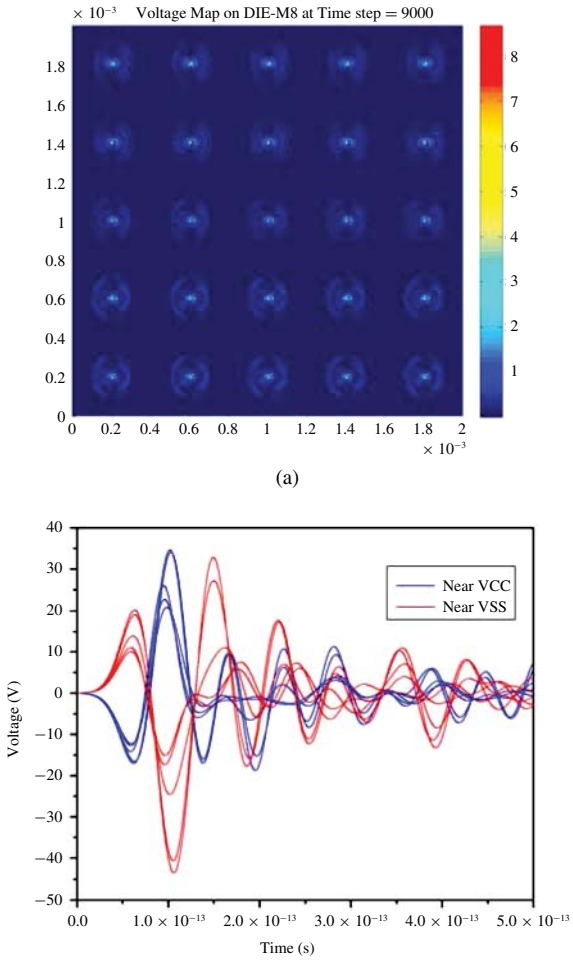

Fig. 10. Simulation of a 2000×2000 μm^2 combined die-package system from M4 to M8 to full package stack. (a) Voltage map on M8 at a time instant. (b) Voltage waveforms sampled at power and ground rails (VCC represents power, and VSS represents ground).

right-hand sides was 50 μm, the strip was 2000 μm long. The structure was discretized into 4 812 943 unknowns, and the time step was chosen as 1.2×10^{-16} s. The CPU cost at each time step on 8 computer nodes was 0.99 s. The *S*-parameters simulated by the proposed parallel algorithm were compared with measured data in Fig. 6. An excellent agreement can be observed.

Next, we simulated a large-scale combined die–package structure obtained from a real product, which was provided by Intel Corporation. The structure involved a complete on-die power grid from M2 to M8 and a complete 17-layer package for power delivery. The chip size was 400×400 μm^2. Due to the high integration density of the on-chip interconnects, the discretization of the structure resulted in 333 182 390 unknowns. The number of layers used in discretization was 1247 and the number of volume unknowns in each layer was 27 370. The total number of volume unknowns accounted for 10.2% of the total number of unknowns. We tested the performance of the proposed parallel transient simulator using this example. The CPU cost at each time step is plotted with respect to the number of CPUs in Fig. 7(a) with the speedup shown in Fig. 7(b), clearly, a linear speedup is observed. In the intermediate range between 8 and 20 CPUs, we even obtain a superlinear speedup. This can be interpreted as follows: since the entire system is large involving over 330 million unknowns, the cache missing issue becomes pronounced in the sequential implementation. By dividing the large system into small pieces, however, the cache missing issue can be alleviated. In addition, from Fig. 7(a), we observe that the sequential solver is already very fast. It solved the 330-million-unknown system in less than 200 s. By developing an embarrassingly parallel solution to the fast sequential solver, we are able to bring the co-simulation of a complete chip and full package with uncompromised accuracy even closer to reality. In Fig. 8(a), we plot the time-domain voltage simulated by the proposed parallel solver, which is shown to agree very well the sequential simulation. In Fig. 8(b), we plot the voltage map of the combined die-package system on M8 at a time instant.

The last example is an ultralarge-scale combined die–package structure that involves a chip area of 2000×2000 μm^2, which is illustrated in Fig. 9. The on-chip part contains a full power grid from M4 to M8. The package part has a full package stack of 17 layers. Again, the structure is from a real product provided by Intel Corporation. The on-chip structure involves a massive number of power and

ground rails, together with ∼1.1774 million vias. The package structure consists of power and ground strips, power and ground planes, holes, and vias that traverse the 17 package layers. The discretization of the entire structure results in 3 542 604 266 unknowns, which is over 3.5 billion. The number of unknowns on each surface is 789 388, the number of volume unknowns in each layer is 90 509. There are 25 current sources uniformly distributed across the chip area. The time step used was 0.5×10^{-16} s. In total, 10 001 steps were simulated. We used 12 nodes with eight cores on each node to simulate this example. The CPU cost at each time step, i.e., one matrix solve, was only 48.12 s. The performance of the proposed parallel solver could be even better if more caches are available on each computer core. In Fig. 10(a), we plot the voltage map of the M8 layer sampled at the 5000th time step. In Fig. 10(b), we show the voltage waveforms sampled in the combined die–package system.

VI. Conclusion

A parallel transient simulator, having electromagnetic accuracy, linear complexity, and linear speedup, was developed for large-scale IC and package simulation such as the co-simulation of die–package interaction that involves billions of unknowns. Numerical experiments have demonstrated its superior performance.

Acknowledgment

The authors would like to thank S. Chakravarty and W. Shi at Intel Corporation, Santa Clara, CA, for providing valuable inputs and suggestions to this paper.

References

[1] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proc. AFIPS Spring Joint Comput. Conf.*, vol. 30. 1967, pp. 483–485.

[2] S. Velamparambil and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2719–2727, Aug. 2005.

[3] O. Ergul and L. Gurel, "Efficient parallelization of the multilevel fast multipole algorithm for the solution of large-scale scattering problems," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2335–2345, Aug. 2008.

[4] G. Rodriguez, Y. Miyazaki, and N. Goto, "Matrix-based FDTD parallel algorithm for big areas and its applications to high-speed wireless communications," *IEEE Trans. Antennas Propag.*, vol. 54, no. 3, pp. 785–796, Mar. 2006.

[5] 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.

[6] J. Fostier and F. Olyslager, "An asynchronous parallel MLFMA for scattering at multiple dielectric objects," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2346–2355, Aug. 2008.

[7] Y.-J. Li and J.-M. Jin, "The FETI-DPEM method for the parallel solution of 3-D EM problems," in *Proc. IEEE Int. Symp. Antennas Propag.*, San Diego, CA, Jul. 2008, pp. 1–4.

[8] A. Ray, G. Kondayya, and S. V. G. Menon, "Developing a finite difference time domain parallel code for nuclear electromagnetic field simulation," *IEEE Trans. Antennas Propag.*, vol. 54, no. 4, pp. 1192–1199, Apr. 2006.

[9] W. Yu, M. R. Hashemi, R. Mittra, D. N. de Araujo, M. Cases, N. Pham, E. Matoglu, P. Patel, and B. Herrman, "Massively parallel conformal FDTD on a BlueGene supercomputer," *IEEE Trans. Adv. Packag.*, vol. 30, no. 2, pp. 335–341, May 2007.

[10] 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. Aid. Des. Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1138–1149, Aug. 2009.

[11] 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.

[12] R. Wang and J. M. Jin, "A symmetric electromagnetic-circuit simulator based on the extended time-domain finite element method," *IEEE Trans. Microw. Theory Technol.*, vol. 56, no. 12, pp. 2875–2884, Dec. 2008.

[13] Z. Lou and J.-M. Jin, "A novel dual-field time-domain finite-element domain-decomposition method for computational electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 54, no. 6, pp. 1850–1862, Jun. 2006.

[14] H. Gan and D. Jiao, "Hierarchical finite-element reduction-recovery method for large-scale transient analysis of high-speed integrated circuits," *IEEE Trans. Adv. Packag.*, vol. 33, no. 1, pp. 276–284, Feb. 2010.

[15] H. Gan and D. Jiao, "A time-domain layered finite element reduction recovery (LAFE-RR) method for high-frequency VLSI design," *IEEE Trans. Antennas Propag.*, vol. 55, no. 12, pp. 3620–3629, Dec. 2007.

[16] 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.

[17] D. Jiao, J. Zhu, and S. Chakrarvarty, "A fast frequency-domain eigenvalue-based approach to full-wave modeling of large-scale 3-D on-chip interconnect structures," *IEEE Trans. Adv. Packag.*, vol. 31, no. 4, pp. 890–899, Nov. 2008.

[18] 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.

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 at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

He is currently working as a Research Assistant in the On-Chip Electromagnetics Research Group, Purdue University. His current research interests include electromagnetics-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 worked at the 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 now an Associate Professor with tenure. She has authored two book chapters and over 130 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics, high-frequency digital, analog, mixed-signal, and radio frequency (RF) integrated circuit design and analysis, high-performance very large scale integration 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, and millimeter wave circuits, wireless communication, and bio-electromagnetics.

Dr. Jiao received the Ruth and Joel Spira Outstanding Teaching Award in 2010, the National Science Foundation CAREER Award in 2008, and the Jack and Cathie Kozik Faculty Startup Award in 2006, which recognizes an outstanding new faculty member in Purdue Electrical and Computer Engineering. She also received an ONR Award through the Young Investigator Program in 2006. 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 the 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 broad SPICE 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 10th recipient) for the timely and accurate 2- and 3-D 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 crosstalk. She was the winner of the Raj Mittra Outstanding Research Award given her by the University of Illinois at Urbana-Champaign in 2000. She has served as a reviewer for many IEEE journals and conferences.

Cheng-Kok Koh (S'92–M'98–SM'06) received the B.S. (with first class honors) and M.S. degrees in computer science from the National University of Singapore, Singapore, in 1992 and 1996, respectively, and the Ph.D. degree in computer science from the University of California, Los Angeles, in 1998.

He is currently a Professor of electrical and computer engineering at Purdue University, West Lafayette, IN. His current research interests include physical design of very large scale integrated circuits and modeling and analysis of large-scale systems.

Dr. Koh received the Lim Soo Peng Book Prize for Best Computer Science Student from the National University of Singapore in 1990, and the Tan Kah Kee Foundation Postgraduate Scholarship in 1993 and 1994. He received the GTE Fellowship and the Chorafas Foundation Prize from the University of California, Los Angeles, in 1995 and 1996, respectively. He was a recipient of the Association for Computing Machinery (ACM) Special Interest Group on Design Automation (SIGDA) Meritorious Service Award in 1998, the Chicago Alumni Award from Purdue University in 1999, the National Science Foundation CAREER Award in 2000, the ACM/SIGDA Distinguished Service Award in 2002, and the Semiconductor Research Corporation Inventor Recognition Award in 2005.