

# 3D Accelerated Electromagnetic Integral Equation Solvers on Parallel Processors for Microelectronic Simulation

Chuanyi Yang, Swagato Chakraborty, Dipanjan Gope, and Vikram Jandhyala

Advanced Computational Engineering Lab, Department of Electrical Engineering, University of Washington Seattle WA 98195 [jandhyala@ee.washington.edu](mailto:jandhyala@ee.washington.edu) 206-543-2186

## ABSTRACT

Simulation of distributed effects in modern day microelectronic circuits is gaining in importance owing to rapid increase in packing density, clock speed and stringent accuracy criteria. Boundary element-based field solution is becoming one of the chosen trends in these simulation tasks owing to its efficiency. Simulation of complex microelectronic circuits can be further accelerated using compressed matrix representation and fast matrix vector products for an accelerated iterative solution. For solving even larger problems in acceptable time and limited local memory resources the fast solution algorithm can be carried out on distributed memory clusters as shown here. This paper deals with the description of the basic algorithms, load balancing techniques and other important aspects of distributed memory –accelerated field solver.

## 1. INTRODUCTION

In the past years VLSI design and analysis has been governed by Kirchoff's current and voltage laws that can be described as D.C. approximation of Maxwell's law. With the steady rise in clock-speed and also with the increased packaging density of modern integrated circuits, Maxwell's equation-based electromagnetic solvers are becoming extremely important. There are a wide class of electromagnetic simulation methodologies that have been developed for this purpose, as for example the Finite Difference Time Domain (FDTD) method, the Finite Element Method (FEM) and the Boundary Element Method (BEM). BEM is based on the integral form of Maxwell's equations and has become popular over the last few years owing to its requirement of less number of unknowns to characterize the same problem compared to volumetric PDE methods.

Integral equation based methods involve a dense system-matrix the solution of which presents a time and memory bottleneck in cases of large scale problems. There exist the genre of accelerated Krylov subspace based approaches to iteratively solve a compressed representation of the system matrix. These methods typically exploit the physical properties of electromagnetic interactions captured through the Green's function and are available in existing literature, and include the fast-multipole method (FMM) [1-2], adaptive integral method (AIM) [3], FFT based methods [4-5] and QR based algorithms [6-7] amongst others. These algorithms

reduce the time and memory requirements for storage and solution of the system matrix almost to linear complexity with respect to the number of unknowns. However, for real-life complex circuit layouts even the fast methods fall short of acceptable efficiency targets and ability to fit the problem in a single processor's memory. As a result circuit designers are often forced to partition the circuit and perform simulations on individual partitioned sub-circuits. While the partitioning strategy still produces acceptable results in many situations, this leads to neglecting coupling effects. In view of the extremely dense packing density of components in today's integrated circuit, the coupling effects can be significant in reality and ignoring them often leads to unacceptable discrepancies between the simulation results and true circuit behavior. On the other hand parallel implementations of fast algorithms provide the designers a way to include the coupling effects in full details, while performing the simulation within practical time and memory constraints. Parallel implementations of fast algorithms appear to be the most promising simulation strategy for handling practical circuit layouts. Moreover, the shift in the microprocessor road-map towards multiple core configurations implies the attractive possibility of parallel processing on user desktops.

In this paper algorithms and techniques related to a parallel implementation of a fast multilevel QR based solver [7] are presented. The pre-determination of ranks of far-field interactions and the unwrapped multilevel version of QR is exploited to enable load balancing and to reduce inter-processor communication requirements throughout the iterative process respectively. In contrast, earlier QR techniques based on adaptive binary tree configurations, as well as fast multipole techniques requiring explicit multilevel tree traversal are significantly more complex to parallelize in terms of load balancing and minimizing inter-node communication.

## 2. INTEGRAL EQUATION FORMULATIONS

### 2.1 Capacitance extraction

To extract the capacitance matrix, the geometry is discretized into triangular surface elements, and piecewise constant bases functions are defined on the triangular elements to model the charge densities at the conductor-dielectric and dielectric-dielectric interfaces. Subsequently electrostatic equations are

written out enforcing the impressed potential on the conductor surface and continuity of the normal component of the electric displacement vector across the dielectric interfaces. These equations in discretized form can be written out as

$$\begin{pmatrix} \bar{Z}_{CC} & \bar{Z}_{CD} \\ \bar{Z}_{DC} & \bar{Z}_{DD} \end{pmatrix} \begin{pmatrix} \sigma_c \\ \sigma_d \end{pmatrix} = \begin{pmatrix} \mathbf{V} \\ \mathbf{0} \end{pmatrix} \quad (2.1)$$

where  $\sigma_c$  is the unknown charge vector on the conductor-to-dielectric interface and  $\sigma_d$  is the charge charge vector on the dielectric-to-dielectric interface.  $\mathbf{V}$  is a set of potentials enforced on the conductor-to-dielectric interface. The capacitance matrix can be extracted from the solution as,

$$\begin{pmatrix} C_{11} & C_{12} & \dots & C_{1N} \\ C_{21} & C_{22} & \dots & C_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N1} & C_{N2} & \dots & C_{NN} \end{pmatrix} \begin{pmatrix} Q_{c1} \\ Q_{c2} \\ \vdots \\ Q_{cN} \end{pmatrix} = \begin{pmatrix} V_{c1} \\ V_{c2} \\ \vdots \\ V_{cN} \end{pmatrix} \quad (2.2)$$

where  $N$  is the total number of conductors,  $Q_{c1}, Q_{c2}, \dots, Q_{cN}$  are total charge on the conductors and  $V_{c1}, V_{c2}, \dots, V_{cN}$  are the set of potentials enforced on the conductors.

## 2.2 Inductance extraction

To compute the inductance and resistance matrix, ports are defined on the structure for excitation, and the frequency dependent complex multi-port impedance matrix is computed. Subsequently, the inductance matrix is computed from the imaginary part of the impedance at respective frequencies, and the resistance matrix is constructed from the real part. Conductors are modeled by the electric field integral equation, and the divergence-free electric current flowing on the surface of conductors is enforced by restricting them to flow in loops. Surface impedance approximation has been used to model resistive loss.

$$E^s(\mathbf{J}) = -j\omega \frac{\mu}{4\pi} \int_S \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|} \mathbf{J}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} ds' - Z_s \mathbf{J}(\mathbf{r}) \quad (2.3)$$

Surface of conductors are discretized into triangles and to describe the current flow, RWG basis function are defined over each triangle pair, which share a common edge. These basis functions are combined to produce loop basis functions. Upon application of the boundary condition on the conductor surface, a matrix equation representing the ‘‘magnetostatic’’ (with phase effects built into the full-wave Green’s function) interaction is set up as

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1M} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{M1} & \alpha_{M2} & \dots & \alpha_{MM} \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_M \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ V_{si} \\ 0 \end{pmatrix} \quad (2.4)$$

where  $M$  is the total number of RWG edges,  $i_1, i_2, \dots, i_M$  are the unknown RWG basis function coefficients and  $V_{si}$  is the delta gap voltage excitation located at certain port  $i$  of the

structure. Subsequently, the matrix equation is solved, and the multiport parameters are obtained from the solution.

$$\begin{pmatrix} Z_{11} & Z_{12} & \dots & Z_{1K} \\ Z_{21} & Z_{22} & \dots & Z_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{K1} & Z_{K2} & \dots & Z_{KK} \end{pmatrix} \begin{pmatrix} I_{p1} \\ I_{p2} \\ \vdots \\ I_{pK} \end{pmatrix} = \begin{pmatrix} V_{s1} \\ V_{s2} \\ \vdots \\ V_{sK} \end{pmatrix} \quad (2.5)$$

where  $K$  is the total number of ports,  $I_{p1}, I_{p2}, \dots, I_{pK}$  are the port currents and  $V_{si}$  is the delta gap voltage excitation located at certain port  $i$  of the structure.

## 3. PARALLEL MULTILEVEL LOW-RANK DECOMPOSITION ALGORITHM

The serial multilevel low-rank decomposition algorithm achieves efficient far-field sub-matrix compression through leveraging an oct-tree decomposition hierarchy and a merged interaction list. In this algorithm, the regions occupied by arbitrary shaped electromagnetic structures are hierarchically decomposed into small cubes and these cubes are represented by an oct-tree data structure. The near field interactions are represented by a neighbor list at the finest level and far-field interactions are represented by interaction lists at all levels. For illustration purposes, a two-dimensional case is shown in figure 1:

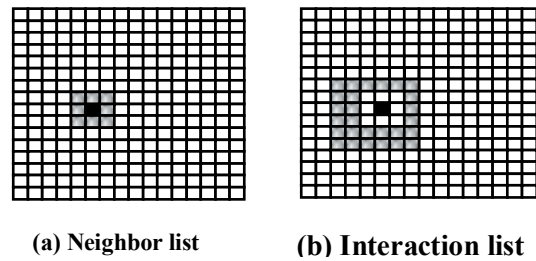


Figure 1: Neighbor list and Interaction list

### 3.1 Load balancing of near and far field interaction matrix construction and storage

Near field interactions are represented by neighbor lists at the finest level and they can be grouped into a linked list data structure as shown by figure 2:

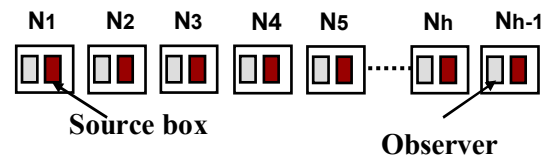


Figure 2. Link list of neighbors

Each node denotes a neighbor interaction list and contains information on source and observer regions. To minimize the inter-processor communication, each processor builds its own copy of data structure that has a relatively trivial demand on both time and memory. As long as the workload associated with each node can be predetermined, the overall work load can be evenly distributed to each processor resulting in a

scalable algorithm. By extracting the number of sources and number of observers at each node of the data structure, it is possible to find the exact load (modulo integer work at each processor) of near field MoM matrices set up for each processor as follows :

$$N_{work-load} = \sum_{i=1}^h \frac{m_i \times n_i}{np} \quad (3.1)$$

where  $h$  is the total number of nodes,  $m_i$  is the number of observer,  $n_i$  is the number of source and  $np$  is the number of processors.

MoM matrices representing near field interactions are constructed after the work load distribution as shown by figure 3:

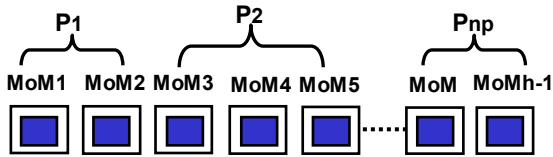


Figure 3. Near field MoM matrices

Similar to near field interactions, merged interaction lists [7] representing far field interaction across different levels can also be grouped into a link list as shown by figure 4:

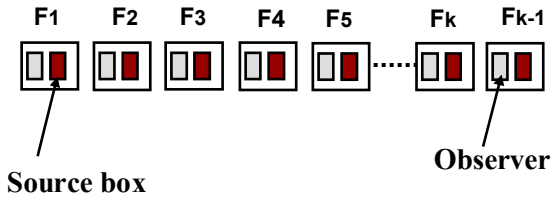


Figure 4. Link list of merged interaction list

Unlike the MoM matrices construction, it is not possible to predetermine the exact overall work load of far field interaction since the Q and R matrices have to be created before the exact rank of each node can be known. However, as discussed in [7], the oct-tree decomposition structure provides a predetermined rank-map for far field interaction sub-matrices across all the levels. A close estimate on each node's work load can be obtained by utilizing the predetermined rank-map. Work load of far field Q and R matrices set up for each processor is derived as following:

$$F_{work-load} = \frac{\sum_{i=1}^k (m_i + n_i) \times r_i}{np} \quad (3.2)$$

where  $k$  is the total number of nodes,  $m_i$  is the number of observer,  $n_i$  is the number of source,  $r_i$  is predetermined rank

of certain type of merged interaction list and  $np$  is the number of processors. As shown by figure 5, after work load distribution, each processor creates its own Q and R matrices.

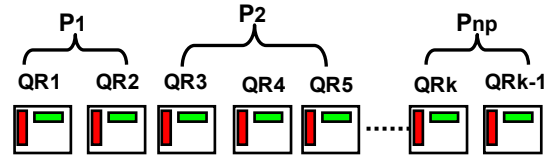


Figure 5. Far field QR matrices

### 3.2 Load balancing of matrix-vector multiplication

Given the fact that the work load of near and far field matrices set up on each processor is proportional to that of matrix-vector multiplication, load balancing scheme discussed above also applies to that of matrix-vector multiplication. As shown by figure 6, after near and far field matrices set up, each processor conducts its local matrix-vector multiplication, where  $V$  is the unknown vector.

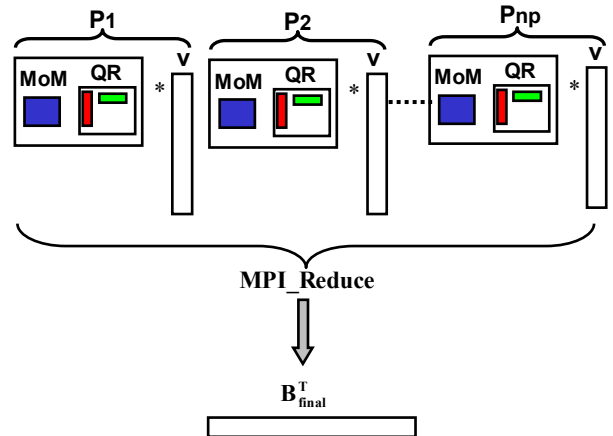


Figure 6. Matrix-vector multiplication

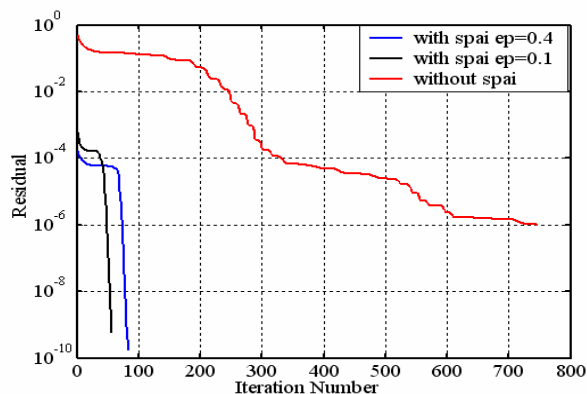
After matrix-vector multiplication, an MPI\_Reduce operation is carried out to obtain the global matrix-vector product  $B_{final}^T$ , that collects vector data from each processor.

### 3.3 Parallel sparse approximate inverse preconditioner

A scalable preconditioner based on the sparse approximate inverse algorithm (SPAI) is implemented and leads to significant speed-up on the iterative solve process. To construct the preconditioner, a sparse matrix  $Z$  is extracted from the near field interaction and used to generate the preconditioner  $M$  for fast convergence.

$$\bar{M}\bar{A}x = \bar{M}b$$

Figure 8 shows one example of the parallel SPAI preconditioner  $M$  drastically speeding up the iterative solving process.



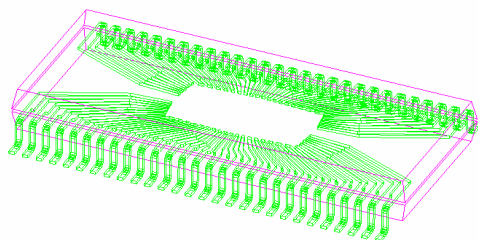
**Figure 7. SPAI preconditioner performance**

In figure 7,  $ep$  denotes a threshold to control the quality of the preconditioner which functions as an efficiency-memory tradeoff

## 4. RESULTS

### 4.1 Capacitance extraction and efficiency test

For the purpose of validation and efficiency testing, capacitance extraction of two test structures were carried out using the proposed method..



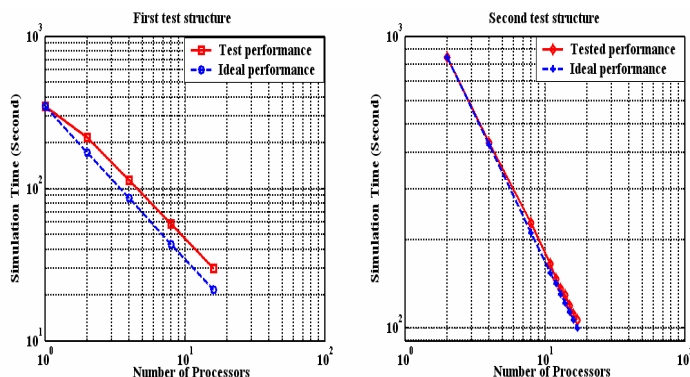
**Figure 8. Packaging structure**

The first test structure is a packaging structure that includes two dielectric layers and a 56-packaging-lead structure sandwiched in between the two dielectric layers. Both dielectric layers have relative permittivity of 11.9. The second test structure, as shown in Figure 9, is one with ten copies of the 56-packaging-lead structure contained in the first test structure. Please note that periodicity is not required or exploited in the proposed technique in any way.



**Figure 9. 560-packaging-lead structure**

To test the efficiency of the load balancing schemes applied in the parallel algorithm, capacitance extraction of the first test structure was run on 1 to 16 processors. The number of unknowns on the first test structure is 7,107. Capacitance extraction of the second test structure was run on 2 to 16 processors. The number of unknowns on the second test structure is 563,216. The test performance of the proposed method and assumed ideal performance were plotted in the same figure.



**Figure 10. Simulation time vs. processor number,**

As shown by figure 10, the proposed method has linear scalability in both test cases owing to the balanced workload distribution.

### 4.2 Spiral inductor Q-factor extraction

The inductance extraction functionality of the proposed method can be used to extract the Q-factor of spiral inductors mostly appeared in RF circuit design. As shown by figure 11, a 3D spiral inductor is connected to the ground by probe patches. Top and bottom ground are connected through shorting vias. There are total 4527 RWG basis functions defined over the triangle mesh. The simulation was run on 18 processors and simulation time was 42 seconds. The extracted Q-factor was plotted together with that from HFSS as shown by figure 12.

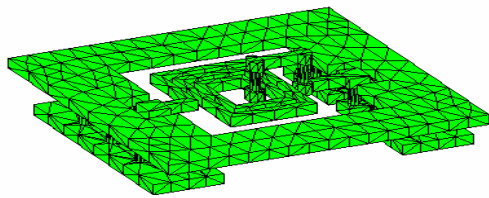


Figure 11. Spiral inductor

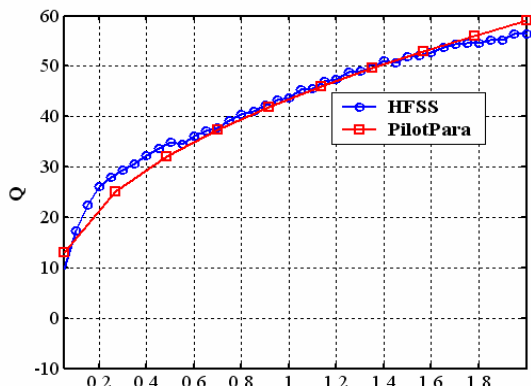


Figure 12. Correlation of HFSS result with PilotPara result

The next simulation case is to extract the Q-factor of the same spiral inductor as above, but which co-exists with another five in a co-planar structure as shown by figure 13. The number of unknowns in this simulation is 30216 and it took 367 seconds to run with 18 processors.

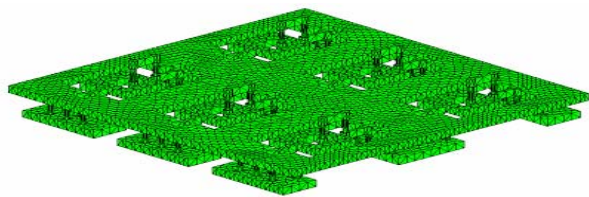


Figure 13. Co-planar spiral inductor

As shown by figure 14, the Q-factor of the co-planar case is lower than that of single inductor case due its mutual coupling including resistive and radiative losses to other three inductors.

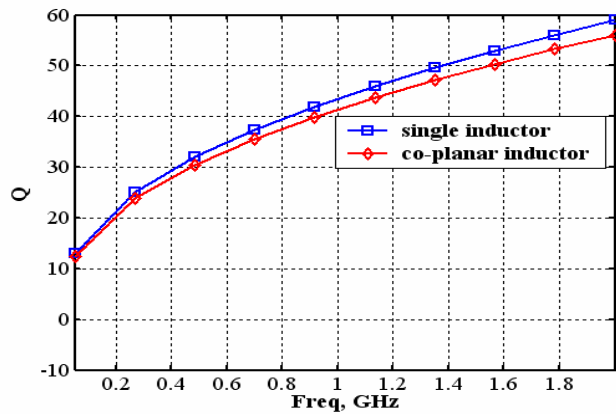


Figure 14. Single inductor Q-factor vs. co-planar inductor

## 5. CONCLUSIONS

This paper presents a parallel multilevel low-rank decomposition algorithm for parasitic extraction of electrical large structures. Linear scalability is achieved through load balancing schemes on near and far field matrices construction. It is expected that the proposed scheme will enable large-scale distributed simulation due to the fact that both fast multilevel simulation technology and parallel processing are simultaneously exploited.

## 6. REFERENCES

- [1] R. Coifman, V. Rokhlin and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription", *IEEE Trans. Antennas Propagat. Mag.*, vol. 35, pp. 7-12, June 1993.
- [2] L.J. Jiang and W.C. Chew, "Modified fast inhomogeneous plane wave algorithm from low frequency to microwave frequency", *IEEE Antennas and Propag. Soc. Int. Symp.*, vol. 2, pp. 22-27, June 2003.
- [3] E. Bleszynski, M. Bleszynski and T. Jaroszewicz, "AIM: Adaptive Integral Method for Solving large-scale electromagnetic scattering and radiation problems", *Radio Science*, vol. 31, pp. 1225-1251, Sept-Oct 1996.
- [4] N. Yuan, T.S. Yeo, X.C. Nie and L.W. Li, "A Fast Analysis of Scattering and Radiation of Large Microstrip Antenna Arrays", *IEEE Trans. on Antennas and Propagation*, vol. 51, pp. 2218-2226, Sep 2003.
- [5] Z. Zhu, B. Song and J. White, "Algorithms in FastImp: a fast and wide-band impedance extraction program for complicated 3-D geometries", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 981 - 998, July 2005
- [6] S.Kapur and D.Long, "IES<sup>3</sup>: a fast integral equation solver for efficient 3-dimensional extraction", *International conference on Computer Aided Design*, pp. 448-455, Nov. 1997.
- [7] D.Gope and V.Jandhyala, "Oct-Tree Based Multilevel Low-Rank Decomposition Algorithm for Rapid 3D Parasitic Extraction", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol-23, pp. 1575 - 1580, Nov. 2004.