Large-Scale Full-Wave Simulation

Sharad Kapur
Integrand Software, Inc.
sharad@integrandsoftware.com

David E. Long
Integrand Software, Inc.
david@integrandsoftware.com

ABSTRACT

We describe a new extraction tool, EMX (Electro-Magnetic eXtractor), for the analysis of RF, analog and high-speed digital circuits. EMX is a fast full-wave field solver. It incorporates two new techniques which make it significantly faster and more memory-efficient than previous solvers. First, it takes advantage of layout regularity in typical designs. Second, EMX uses a new method for computing the vector-potential component in the mixed potential integral equation. These techniques give a speed-up of more than a factor of ten, together with a corresponding reduction in memory.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids

General Terms

Algorithms, design, verification

Keywords

Layout extraction, multipole, integral equations

1. INTRODUCTION

We believe that full-wave field solvers can be made practical for the extraction of the moderate-sized circuits and blocks typical of RF and analog design. If this can be done, the current hodgepodge of pattern matching, static approximations, and individual point tools for capacitance, inductance, substrate coupling, etc., can be eliminated for these applications. Towards this end, we introduce EMX, a new full-wave field solver targeted at high-speed IC design. It extracts interconnect and passive components, and includes all coupling effects and the interaction with the substrate. EMX is still a work-in-progress, but even in its present state, it is already ten to thirty times faster than current fast full-wave field solvers.

DAC 2004, June 7–11, 2004, San Diego, California, USA.

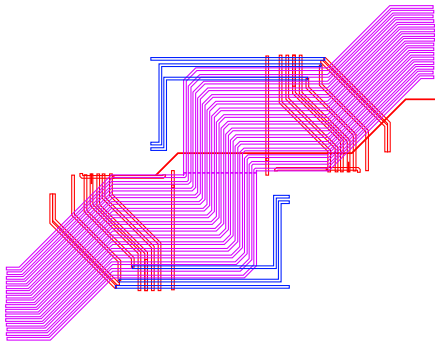

Figure 1: Regular routing on a 10 GHz chip

The main problem with a patchwork approach is in stitching the different effects together in a consistent way. Correctly accounting for coupling and frequency-dependent effects is difficult without a unified formulation that can handle all of the effects together.

The first generation of fast field solvers used dense matrix compression methods such as the Fast Multipole Method (FMM), the Precorrected-FFT (PFFT), and SVD compression [3, 4, 5, 6, 10]. All these methods decompose the matrix into an explicit direct part and a implicit far-field part. EMX uses the FMM, but also incorporates several new ideas. This paper describes two of them: a technique for exploiting structural regularity in IC layouts, and an efficient new method for computing the vector potential.

The first new idea in EMX is to exploit regularity. Typical IC layouts are very regular. For example, wires tend to be paths of constant width, the distance between adjacent routing is typically constant, and most routing is at 90 or 45 degree angles. Figure 1 shows typical routing on a 10 GHz chip that exhibits this sort of regularity.

The second key idea is that the vector potential interactions (the most significant cost in a full-wave solve) can be computed with about the same cost as a scalar interaction. The method depends on decomposing the currents into divergence-free and curl-free parts. The divergence-free parts give the dominant contributions to the vector potential, and these are captured exactly by the new method. The less-important contributions of the curl-free part are approximated, and we show that this approximation is accurate for IC problems.

The combination of these two ideas leads to significant time and memory savings. The largest example in section 5 is a quadrature VCO. EMX requires only about five minutes

to simulate this structure, and needs less than 100 MB of memory for the matrices.

2. FORMULATION

EMX uses a standard integral formulation for the problem. In the frequency domain, the stimulus electric field E_s is expressed in terms of ohmic losses, the vector potential A and the scalar potential ϕ:

$$E_s(r) = \frac{1}{\sigma} J(r) + j\omega A(r) + \nabla\phi(r). \quad (1)$$

The vector and scalar potentials are obtained by integrating over the conductors: $A(r) = \int G_A(r, r')J(r')\,dr'$, and $\phi(r) = \int G_\phi(r, r')\rho(r')dr'$. J is the current density, ρ is the charge density, G_A is the vector potential Green's function, and G_ϕ is the scalar potential Green's function.

For the numerical solution of the equations, the structure is discretized into triangular and rectangular elements, and a Galerkin scheme is applied. The individual basis functions are composed of linear Rao-Wilton-Glisson rooftop functions [7] defined on the shapes. To avoid ill-conditioning at low frequencies, we adopt a set of basis functions that decompose the current density into curl-free and divergence-free parts [9]. The composition of basis functions in terms of rooftops is expressed by an $r \times b$ sparse matrix V. Similarly, the divergence of each basis function is expressed by a $t \times b$ sparse matrix S. The matrix formulation of equation 1 is:

$$V^T \Omega V + j\omega V^T A V + S^T \Phi S = B.$$

To solve this system, EMX uses a Krylov-subspace iterative solver [1] combined with a kernel-independent FMM [3] for the application of the dense matrices A and Φ. In a full-wave solver, representing and applying the vector potential matrix A is the most significant cost.

3. REPRESENTATION OF A

There are up to three rooftop functions for each triangle, and up to four for each rectangle. Consequently, there are up to sixteen different interactions between a pair of shapes. This means that the FMM will require about an order of magnitude more direct interactions for representing A than for representing Φ.

We now describe a new method for compressing A based on the curl-free and divergence-free basis functions. A divergence-free function represents a current loop, as shown in figure 2. For each triangle in the loop, there are two rooftops: one representing current flow into the triangle, and one representing current flow out. The amount of current flowing into each triangle is equal to the amount flowing out, so there is no charge accumulation anywhere in the loop.

Consider the contribution $\int G_A(r', r)J(r)\,dr$ to the vector potential at r' due to the current flow $J(r)$ in the shaded triangle. $J(r)$ is the superposition of two rooftops. If ρ denotes the projection of r into the plane of the triangle, then $J(r) = w(\rho - \rho_1) - w(\rho - \rho_2)$, where w is some constant. But this is simply equal to $w(\rho_2 - \rho_1)$, i.e., *the current flow in a triangle due to a divergence-free basis function is a constant vector*. Hence the vector potential contribution is just $w(\rho_2 - \rho_1)\int G_A(r', r)\,dr$, and this only requires a simple scalar integral over the source triangle. Similarly, Galerkin testing of the vector potential using a divergence-free basis function requires only a scalar integral over each observation

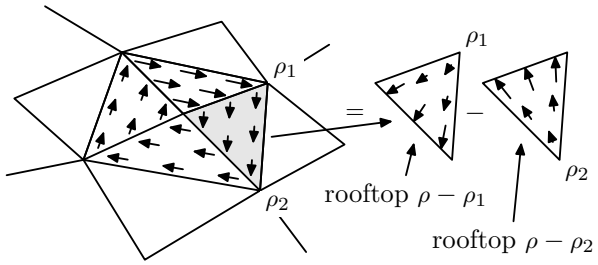

Figure 2: Divergence-free basis function

$w \leftarrow Vv$
for each shape i, initialize s_i to the vector $(0, 0, 0)$
for each rooftop i with coefficient w_i
    let j be the index of the shape for rooftop i
    let ρ_c be rooftop i evaluated at shape j's centroid
    $s_j \leftarrow s_j + w_i\rho_c$
define matrix A_s by $A_s(i, j) = \int_i \int_j G_A$, where the
    integrals are scalar integrals over shapes i and j
multiply componentwise: $u \leftarrow A_s s$
    That is, u is an array of vectors. The
    x-components of u are obtained by multiplying
    the x-components of s by A_s, etc.
for each rooftop i
    let j be the index of the shape for rooftop i
    let ρ_c be rooftop i evaluated at shape j's centroid
    $x_i \leftarrow \rho_c \cdot u_j$
return $V^T x$

Figure 3: Procedure for computing $V^T A V v$

triangle. These observations lead to the algorithm shown in figure 3 for computing $V^T A V$ times a vector.

This algorithm is exact for the part of $V^T A V$ that corresponds to divergence-free source functions interacting with divergence-free testing functions. For the other parts of $V^T A V$, it is only an approximation. The approximation is accurate because the current flow varies smoothly from shape to shape. Figure 4 shows such a comparison of $V^T A V v$ computed directly and using the method of figure 3. There are two examples: an inductor and a parallel-plate capacitor. The plot shows the relative error in $V^T A V v$, where v is the vector of basis function coefficients at solution. $V^T A V v$ for the inductor has minimal error well beyond the inductor's resonant frequency. The capacitor exhibits a consistent error of about 0.5%. Note however that the error in the vector potential for the capacitor is relatively insignificant, since the overall inductive effects there are parasitic and small.

4. EXPLOITING REGULARITY

Most layouts contain a large amount of regularity, such as wires of constant widths and at constant spacings. This regularity can be exploited if the mesh is made up of repeated instances of a few basic shapes. EMX uses a combination of heuristic methods to produce regular meshes.

Wire recognition is the most useful heuristic. Figure 5 shows the layout of an MCM balun with the shaded areas indicating recognized wires. Wires are meshed as indicated in figure 6. The elements near the edges have a width on the order of a skin depth. The lengths of most of the elements

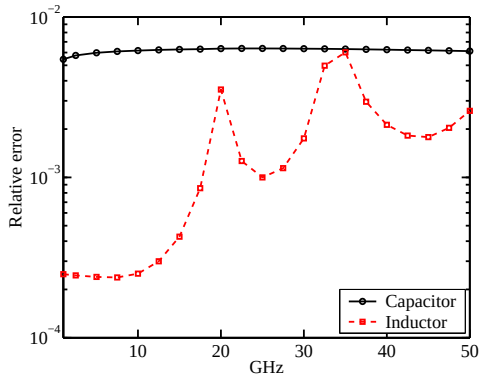

Figure 4: Relative error in computing V^TAVv

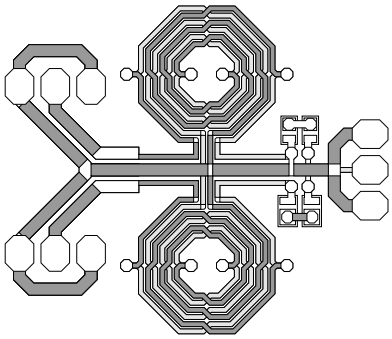

Figure 5: Recognized wires (shaded) in a balun

are chosen from among a few fixed sizes that are common to all the wires. In figure 6, the shaded shapes make up the regular part of the mesh. Shapes that have the same shading are isomorphic and represent repeated instances of a single canonical shape.

The meshing method for general regions is based on the medial axis [8]. Figure 7 shows the mesh of a general region, with isomorphic shapes shaded identically.

As figures 6 and 7 show, shapes are considered isomorphic even when they have been rotated and flipped in various ways. Individual shapes in the mesh are represented as structures containing a transform, an offset, and a reference to a canonical shape.

EMX takes advantage of mesh regularity to reduce both the time required to construct the dense matrix representation and the memory required to store it. The time is reduced by adding a cache to the integral computation routines. The cache typically reduces the time required for integrals by a factor of two to six.

Memory reduction comes from compressing the storage

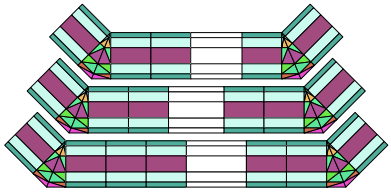

Figure 6: Wire mesh with isomorphic shapes shaded

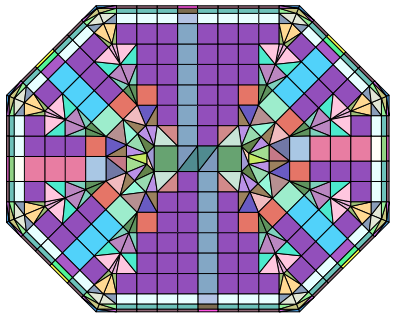

Figure 7: Medial axis mesh of a general region

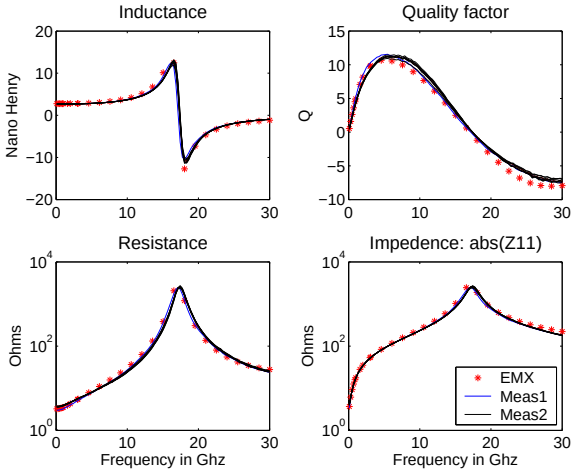

Figure 8: Inductor simulations and measurements

for the direct interactions. The numeric part of the direct interactions matrix is viewed as an array of numbers; many of the numbers are identical because of mesh regularity. Exploiting this duplication reduces memory by about a factor of three.

5. EXAMPLES

The following examples were used to demonstrate the performance of EMX: a 3 nH inductor from a foundry 0.25 μm inductor library (figure 8 shows simulations vs. measurements); a 3-pole filter composed of three inductors and seven capacitors and fabricated in an MCM technology; the routing in figure 1 from a 0.18 μm design; and a 5 GHz quadrature VCO [2] (figure 9) which was built in a 5-layer CMOS process. All examples were run on a 1.5 GHz Compaq Presario PC. Tolerances were set to give an accuracy of 1%. The discretization was set so as to achieve convergence in sensitive quantities like inductor Q.

Table 1 summarizes the results for all the examples. The table shows the number of ports in the example, the discretization level, the amount of memory required for the matrix representation, and the time to build the matrix representation and do the solves at a single frequency point. The memory and time statistics are given both with and without mesh regularization. Even relatively complicated layout has significant regularity, i.e., the ratio of the canonical shapes to the total number of shapes is small.

Example					No Regularization		With Mesh Regularization		
Name	Ports	Shapes	Basis Functions	Vector Elements	Matrix Memory	Time	Canonical Shapes	Matrix Memory	Time
3 nH inductor	2	3,400	7,700	11,600	15MB	25s	200	5MB	10s
routing	32	9,500	15,200	30,300	20MB	55s	210	7MB	35s
3-pole filter	2	14,700	23,900	46,500	49MB	90s	630	14MB	55s
VCO	6	57,800	94,500	188,700	306MB	560s	1,300	87MB	365s

Table 1: Summary of time and memory requirements

Simulator	Example	Shapes	Basis Functions	Time	Matrix Memory	Time IES^3/EMX	Memory IES^3/EMX
IES^3 [4]	CMOS Inductor (t)	11,500	16,000	500s	267MB	—	—
EMX	Square Inductor (r+t)	11,800	21,600	15s	6MB	33	44
EMX	Octagonal Inductor (t)	11,100	16,000	31s	11MB	16	24
EMX	Octagonal Inductor (r+t)	6,300	11,200	16s	8MB	31	33

Table 2: Comparison of EMX to IES^3

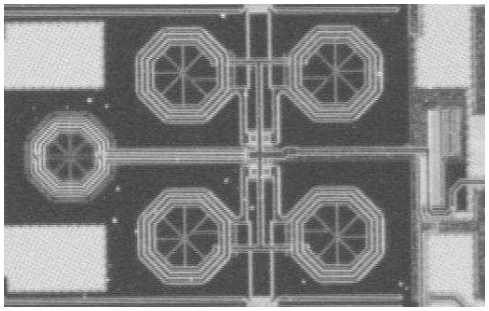

Figure 9: 5 GHz CMOS VCO

The fairest comparison we could make with other full-wave field solvers is to IES^3 [4]. IES^3 uses a similar formulation to EMX, but the mesh is based on Delaunay triangulation, and it uses collocation and SVD compression instead of a Galerkin method and the FMM. One of the examples given in that paper was a two-metal CMOS spiral inductor. We ran EMX on two similar inductors. Table 2 compares the simulators. The (t) and (r+t) in the second column indicate whether the mesh contains triangles, rectangles, or both.

6. CONCLUSION

Exploiting layout regularity and using a new representation of the vector potential interactions together give significant time and memory savings in a field solver. When coupled with fast frequency sweep methods, it should be possible to extract broadband models of complete RF blocks in minutes. EMX does not currently exploit hierarchy directly, but this is a natural extension of the ideas presented.

7. REFERENCES

[1] R. Barrett et al. *Templates for the Solution of Linear Systems*. SIAM, 1994.

[2] S. L. J. Gierkink, S. Levantino, R. C. Frye, C. Samori, and V. Boccuzzi. A low-phase-noise 5-GHz CMOS quadrature VCO using superharmonic coupling. *IEEE J. of Solid-State Circ.*, 38(7):1148–1154, July 2003.

[3] S. Kapur and D. E. Long. Large-scale capacitance calculation. In *Proc. 37th Design Automation Conf.*, pages 744–749, June 2000.

[4] S. Kapur, D. E. Long, and J. Zhao. Efficient full-wave simulation in layered, lossy media. In *Proc. Custom Integrated Circuits Conf.*, pages 211–214, May 1998.

[5] K. Nabors and J. K. White. FastCap: A multipole accelerated 3-D capacitance extraction program. *IEEE Trans. on CAD*, 10(11):1447–1459, Nov. 1991.

[6] J. R. Phillips and J. K. White. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Trans. on CAD*, 16(10):1059–1072, Oct. 1997.

[7] S. M. Rao, D. R. Wilton, and A. W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans. on Antennas and Propagation*, AP-30:409–418, May 1982.

[8] V. Srinivasan, L. R. Nackman, J.-M. Tang, and S. N. Meshkat. Automatic mesh generation using the symmetric axis transformation of polygonal domains. *Proc. IEEE*, 80(9):1485–1501, Sept. 1992.

[9] D. R. Wilton and A. W. Glisson. On improving the stability of the electric field integral equation at low frequency. In *Proc. IEEE Antennas and Propagation Soc. National Symp.*, pages 124–133, 1981.

[10] Z. Zhu, B. Song, and J. K. White. Algorithms in FastImp: A fast and wideband impedance extraction program for complicated 3-D geometries. In *Proc. 40th Design Automation Conf.*, pages 712–717, 2003.