Efficient Techniques for Accurate Extraction and Modeling of Substrate Coupling in Mixed-Signal IC's

João Paulo Costa [*] Mike Chou [†] L. Miguel Silveira [*]

[*] INESC/Cadence European Laboratories
Dept. of Electrical and Comp. Engineering
Instituto Superior Técnico
Lisboa, 1000 Portugal

[†] Research Laboratory of Electronics
Dept. of Electrical Eng. and Comp. Science
Massachusetts Institute of Technology
Cambridge, MA, 02139, U.S.A.

Abstract

Accurate modeling of noise coupling effects due to crosstalk via the substrate is an increasingly important concern for the design and verification of mixed analog-digital systems. In this paper we present a technique to accelerate the model computation using BEM methods that can be used for accurate and efficient extraction of substrate coupling parameters in mixed-signal designs.

1 Introduction

The design of single chip mixed-signal systems is currently a very active research area, sparked by the ever continuous emphasis on compactness and cost reduction and the widespread growth and interest in wireless communications. A major challenge in the design of such chips is the need for accurate modeling of the parasitic noise coupling through the common chip substrate, between high-speed digital and high-precision analog components [1, 2, 3, 4]. Fast switching logic components inject current into the substrate causing voltage fluctuations which can affect the operation of sensitive analog circuitry through the body-effect.

It is generally accepted that for frequencies up to a few gigahertz, the substrate behaves resistively [3, 5]. Thus, in order to model coupling effects, it is sufficient to solve Laplace's equation inside the substrate with appropriate boundary and interface conditions. Many techniques have been used to solve this problem, including Finite Element (FEM) and Finite Difference (FD) methods [2, 6, 7, 8]. However, for complex layouts the number of unknowns resulting from the discretization of the three-dimensional substrate volume may become too large for practical purposes. Boundary-Element methods (BEM) are very appealing for the solution of this type of problems because the size of the matrix to be solved is dramatically reduced since only the relevant boundary features are discretized [9, 10, 11]. However they produce dense matrices which are expensive to store and factor directly. Iterative methods, such as Krylov subspace algorithms, combined with some sparsification technique to accelerate the computation of matrix-vector products can be very efficient for solving large BEM problem. Recently an eigendecomposition-based method was presented that eliminates the need for dense-matrix storage and can be used to accurately compute substrate models while taking all of the substrate edge effects into account [12]. Nevertheless its computation time for large, dense circuits is still high. In this paper we present an efficient technique for substrate coupling parameter extraction, based on a precorrected-DCT technique that extends the eigendecomposition-based technique and accelerates operator-vector application. The method provides a substantial speedup in terms of computation time and allows for efficient and accurate extraction of substrate models in large layouts.

In the next section we summarize the problem of modeling substrate coupling and review the functional eigendecomposition technique. Then we present the precorrected-DCT technique and show how it can be used to speed up the extraction process producing a model that can be used to perform coupled circuit-substrate simulation. Complexity comparisons between the proposed and currently available techniques is provided in Section 4. Computational results from applying the proposed technique to an actual layout are presented in Section 5. Finally conclusions are drawn in Section 6.

2 Background

2.1 Problem Formulation and Model computation

Assuming the electrostatic approximation, a commonly used model for the substrate is to consider it as a stratified medium composed of several homogeneous layers characterized by their conductivity, as shown in Figure 1 [3, 11] On the top of this stack of layers a number of ports or contacts, assumed planar, are defined, which correspond to the areas where the designed circuit interacts with the substrate. Figure 1 also shows schematically the resistive coupling model that we seek to extract. In the figure the substrate backplane is shown as electrically grounded. Floating substrates can also be handled with minimal modifications to the formulation.

Modeling the coupling through the substrate amounts to computing the relation between a set of injected currents and the resulting potential distribution. Thus, given a set of m contacts, we seek a model that relates the currents on those contacts, $\boldsymbol{I}_c$ to their voltage distribution $\boldsymbol{V}_c$. In practice, for reasons of accuracy it is necessary to discretize each of the contacts into a collection of n disjoint panels. Using a Galerkin scheme [13], the current density on each panel is assumed uniform and a set of equations relating the currents and potentials on all panels in the system is then formulated

$$\boldsymbol{\Phi}_p = \boldsymbol{Z}_p \boldsymbol{I}_p \tag{1}$$

where $\boldsymbol{I}_p, \boldsymbol{V}_p \in \mathbb{R}^n$ and $\boldsymbol{Z}_p \in \mathbb{R}^{n\times n}$, $\boldsymbol{I}_{p_i}$ denotes the total

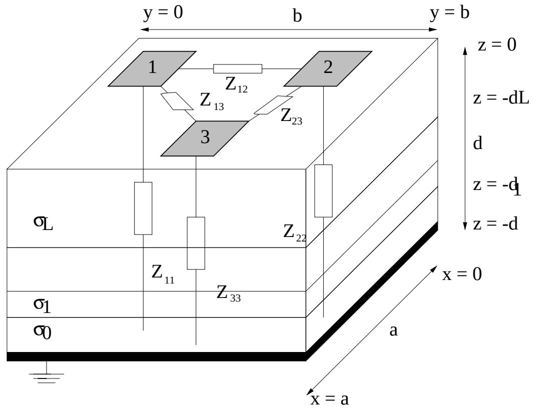

Figure 1: Cross-section of substrate showing a 3D model as an homogeneous multilayered system with contacts on the top surface.

current on panel i and $\boldsymbol{V}_{p_j}$ the average potential on panel j. Entry (i,j) in the impedance matrix $\boldsymbol{Z}_p$ is given by

$$\boldsymbol{Z}_{p_{ij}} = \frac{1}{a_i a_j}\int_{s_i}\int_{s_j} \boldsymbol{G}(x,y;x',y')\,da\,da' \qquad (2)$$

where a_i and a_j are respectively the areas of panel i and j and $\boldsymbol{G}(x,y;x',y')$ is the Green's function, which accounts for the problem's boundary and interface conditions, and relates the potential at some observation point (x,y) due to a unit current injected at some source point (x',y'). Clearly $\boldsymbol{Z}_p$ is dense since currents injected into any panel i produce a non-zero potential on every other panel j.

Extraction of the substrate coupling model is accomplished by appropriately setting the voltages at the contacts and solving (1) for the detailed current distribution. The current flowing into each contact is then obtained by summing the currents from all panels in the contact. The algorithm is similar to the standard capacitance extraction problem [14]: extraction of the full model requires m linear solves for a system with m contacts. The straightforward way to accomplish this task requires the computation of $\boldsymbol{Z}_p$ using (2). In [3] it was shown that the Green's function $\boldsymbol{G}(x,y;x',y')$ for the bounded substrate with grounded backplane can be written as a double infinite series of cosines in x and y. It was also shown that by truncating the series to M terms and rewriting the resulting equation, each entry $\boldsymbol{Z}_{p_{ij}}$ can be constructed from linear combinations of appropriate terms from a two-dimensional $M \times M$ array $\{F_{lm}\}$, which is computed once and for all with a Type-1 Discrete Cosine Transform (DCT) [15]. Given $\boldsymbol{Z}_p$, direct solution of (1) using Gaussian elimination is overwhelmingly expensive. Instead, an iterative algorithm such as the Generalized Minimum RESidual algorithm, GMRES [16]. However, each iteration of GMRES requires a matrix-vector product, $\boldsymbol{Z}_p\boldsymbol{I}_p^k$, at a cost of $\mathcal{O}(n^2)$ which, coupled with the $\mathcal{O}(n^2)$ memory required to store the dense $\boldsymbol{Z}_p$, limits the size of the problem to be analyzed to a few hundred panels.

2.2 Sparsification via Eigendecomposition

Recently, the application of well-known eigendecomposition techniques to this problem were proposed in order to speedup the matrix-vector product required at each iteration of GMRES [12]. In this method the computation of the matrix-vector product, which corresponds in essence to computing a set of average panel potentials given a substrate injected current distribution, is performed by means of an eigenfunction decomposition of the linear operator that relates injected currents to panel potentials (without loss of generality we will consider square layouts, $a = b$). In [12] (and previously in [3] albeit with a different objective) it was shown that the surface eigenfunctions of the linear integral operator $\mathcal{L}$ that relates injected currents to panel potentials are $\boldsymbol{\varphi}_{ij}(x,y) = \cos\left(i\pi x/a\right)\ \cos\left(j\pi y/a\right)$ (for a derivation of the eigenvalues, $\boldsymbol{\lambda}_{ij}$, see [3, 12]). In this technique, the global current density $\boldsymbol{q}(x,y)$ is first represented as

$$\boldsymbol{q}(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} q_{mn}\,\boldsymbol{\vartheta}\left(x - \frac{\beta(m)\,a}{M}, y - \frac{\beta(n)\,a}{M}\right) \quad (3)$$

where q_{mn} is the total current at panel (m,n), $\beta(r) = (r + 1/2)$ and $\boldsymbol{\vartheta}(x,y)$ is a square-bump function that serves as an averaging function, defined as

$$\boldsymbol{\vartheta}(x,y) = \begin{cases} \frac{M^2}{a^2} & \Leftarrow\ -\frac{a}{2M} \le x,y \le \frac{a}{2M} \\ 0 & \Leftarrow\ elsewhere. \end{cases}$$

Then, the eigenfunctions of the integral equation operator are used to expand $\boldsymbol{q}(x,y)$,

$$\boldsymbol{q}(x,y) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{ij}\ \boldsymbol{\varphi}_{ij}(x,y). \qquad (4)$$

If the coefficients a_{ij} can be quickly computed, then the potential can then be readily obtained as

$$\boldsymbol{\Phi}(x,y) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \boldsymbol{\lambda}_{ij}\ a_{ij}\ \boldsymbol{\varphi}_{ij}(x,y). \qquad (5)$$

Truncating (5) to a finite series with $M \times M$ coefficients, allows us to compute the average panel potential $\overline{\boldsymbol{\Phi}}_{pq}$ at position (p,q) as [12]

$$\overline{\boldsymbol{\Phi}}_{pq} = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} b_{ij}\cos\left(\frac{\beta(p)\,\pi\,i}{M}\right)\cos\left(\frac{\beta(q)\,\pi\,j}{M}\right) \quad (6)$$

It can be shown [12] that the $M \times M$ coefficients $\{a_{ij}\}$ can be computed via a forward Type-2 DCT, and then the $M \times M$ array $\{\overline{\boldsymbol{\Phi}}_{pq}\}$ via an inverse Type-2 DCT (both implemented efficiently with FFT's [15]). Thus, the $\boldsymbol{Z}_p\boldsymbol{I}_p^k$ product can be computed in $\mathcal{O}(2 \cdot M^2 \cdot \log_2 M)$ operations. If we assume that all panels are minimum sized cells on the $M \times M$ substrate grid, then if for instance only 10% of the cells are occupied by actual panels (*i.e.* $n = (0.1)M^2$), a realistic number, the cost of computing the dense $\boldsymbol{Z}_p\boldsymbol{I}_p^k$ product directly is $\mathcal{O}(0.01 \times M^4)$. At $M = 128$, eigendecomposition is already an order of magnitude faster than direct multiplication.

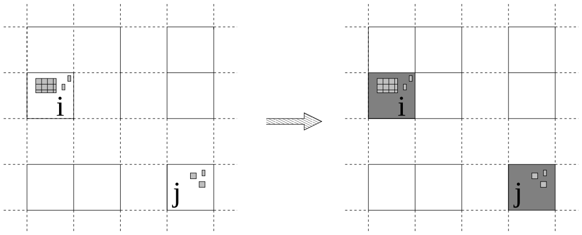

Figure 2: Representation of coarse grid discretization and current projection into cells.

3 Accelerating Potential Computation Via Precorrected-DCT

For large layouts, with many substrate contacts, even the eigendecomposition algorithm becomes overly expensive. Fortunately it is possible to significantly speedup the matrix-vector product using the precorrected-DCT (PcDCT) algorithm. The main idea behind the PcDCT algorithm is to realize that the effect of an injected current in a panel p_i on the potential of another far away panel p_j, can be considered the same for small variations in the distance between panels. Similar ideas have been used extensively in multipole-accelerated algorithms [17, 14, 11]. The approach developed here is reminiscent of other precorrected algorithms previously published [18] and relies on a simple coarser-grid projection method, which can be used in combination with the DCT to accelerate the computation of the matrix-vector product while taking into account all of the substrate boundary effects.

Consider a group of panels at some location (call it cell g_i) and another cell, g_j, far away from g_i. If a current is injected in some panel in g_i, this current will have a similar effect on every panel of g_j, as long as the distance between cells is *large*, where *large* depends of the substrate's profile. Similarly, if the same current is injected in another panel in g_i the effect on g_j will be the same. Thus, for the purposes of computing the effect on g_j's potential, it may suffice to distribute the current uniformly over the entire cell g_i (see Figure 2 for a representation). With the above considerations we can construct an algorithm to approximately compute a potential distribution due to an injected current distribution. This algorithm is composed of four main steps: a *projection step* to construct a coarse (cell-level) representation for the detailed panel current distribution, a *computation step* to obtain the coarse potential distribution resulting from the coarse current representation, an *interpolation step* to interpolate the detailed panel potentials from the coarse (cells) potential distribution, and a *correction step* to calculate the nearby panel interactions directly, and make the appropriate corrections. To this end, the substrate top surface is first divided into a coarse grid of cells (unrelated to the underlying fine panel discretization). The panel currents are then projected onto those cells, for instance such that the total sum of the panel currents in each cell is uniformly distributed over the cell. Given the panels' current vector $\boldsymbol{I_p}$, we can compute the cells' current vector $\boldsymbol{I_g}$ as

$$\boldsymbol{I_g} = \boldsymbol{W}\boldsymbol{I_p} \tag{7}$$

where s^2 is the number of grid cells, n is the number of panels and $\boldsymbol{W} \in \mathbb{R}^{s^2 \times n}$ is a simple incidence matrix indicating whether a panel belongs to a given cell or not. Once a vector of cell currents is obtained, the corresponding vector of cell potentials can be computed. For this operation, the eigendecomposition technique previously reviewed is used. This operation is represented as

$$\boldsymbol{\Phi}_g = \boldsymbol{\mathcal{L}}\, \boldsymbol{I}_g \tag{8}$$

where $\boldsymbol{\mathcal{L}}$ is again the linear operator that represents Poisson's equation, now applied to the coarse grid of cells, which makes the operation extremely efficient. Once the vector of cell potentials, $\boldsymbol{\Phi}_g$ is known, the corresponding panel potentials can be trivially obtained from

$$\tilde{\boldsymbol{\Phi}}_p = \boldsymbol{W^T}\boldsymbol{\Phi}_g. \tag{9}$$

$\tilde{\boldsymbol{\Phi}}_p$, given by (9), represents an approximation to the panel potentials computed under the assumption that all cells were distant enough from each other. Such assumption is clearly not true everywhere, since for instance cells are not far away from themselves nor from their nearest neighbor cells. Since the approximation is of poor accuracy for nearby panel interactions, it is necessary to compute the nearest-neighbor interactions directly for each panel, and to make appropriate corrections to $\tilde{\boldsymbol{\Phi}}_p$. This is done in a manner similar to the precorrected FFT scheme [18]. First, we remove from every cell potential the influence of the self and nearest neighbor cell's currents. The contribution of the current on a specific cell i to the average potential of another cell k, ψ_k^i can be written as $\psi_k^i = q_i\ h_k^i$, where q_i is the sum of all the injected currents in cell i and the h_k^i coefficients can be obtained from appropriate combinations of the result of a Type-1 DCT on the coefficients b_{ij} of (6). This DCT needs to be computed only once and discarded afterwards, thus its cost is negligible. Furthermore, we should emphasize that one needs to compute and store the h_k^i coefficients *only* for self and nearest-neighbor cell interactions. For every cell k, the total contribution to k's panel potentials from self and neighbor cells can then be written as

$$\boldsymbol{\Psi}_k = \sum_{i \in \mathcal{N}} h_k^i \boldsymbol{I}_{p_i} \tag{10}$$

where $\mathcal{N}$ represents the set of neighbor cells and $\boldsymbol{I}_{p_k}$ is the vector of panel currents in cell k

The second piece of the precorrection requires the computation of the exact near panel interactions, $\boldsymbol{V}_k$, for cell k. One way to perform this computation is to use the Green's function directly to determine the individual relations between panel currents and potentials for panels in neighbor cells and panels within the same cell. These relations can be represented as a set of small matrices $\boldsymbol{Z}_k^i$ which as a whole are simply a sparsified version of the large $\boldsymbol{Z}_p$ in (1). $\boldsymbol{V}_k$ can be written as

$$\boldsymbol{V}_k = \sum_{i \in \mathcal{N}} \boldsymbol{Z}_k^i \boldsymbol{I}_{p_i}. \tag{11}$$

Finally, the corrected potential of cell k's panels, $\mathbf{\Phi}_{p_k}$, is then given by

$$\mathbf{\Phi}_{p_k} = \tilde{\mathbf{\Phi}}_{p_k} - \mathbf{\Psi}_k + \boldsymbol{V}_k. \tag{12}$$

where the approximate self and nearest neighbor cell interactions are subtracted and the exact near panel interactions are added.

As described, the algorithm uses a simple coarse-grid projection scheme whereby the current for all panels in a cell is uniformly distributed over the cell. Theoretically, the choice of s, the cell discretization parameter, has negligible effect on accuracy. Higher accuracy can be obtained by using higher-order nearest neighbor cells in the exact computation or by refining the projection scheme, similarly to what is suggested in [19] in a different context.

4 Complexity Comparison

In this section we compare the efficiency of the proposed precorrected-DCT accelerated algorithm with the Green's function method (i.e. direct computation of $\boldsymbol{Z}_p$ and solution of (1) using GMRES) as well as the unaccelerated eigendecomposition method presented in [12]. In the following we will use m as the number of contacts, n as the number of panels, M as the number of cells per side in the eigendecomposition method and s as the number of PcDCT cells in each dimension.

For large circuits with several hundreds of contacts, the dominant cost of using the Green's function method is $\mathcal{O}(m\,K_G\,n^2)$ where K_G is the average number of GMRES iterations. Similarly, the cost of using the eigendecomposition method is $\mathcal{O}(4\,m\,K_E\,M^2\log_2 M)$ where K_E is the average number of GMRES iterations. Since n is typically on the order of M^2, the eigendecomposition method is clearly more efficient than the Green's function method. For the PcDCT method the cost is mainly dependent on the number of cells s^2. Assuming an uniform distribution of panels on the surface, the average number of panels per cell is n/s^2. The dominant cost factor in the computation of a GMRES iteration is $\mathcal{O}(4s^2\log s + 5n^2/s^2)$. The total cost is then $\mathcal{O}\left(mK_M(4s^2\log s + 5n^2/s^2)\right)$. For small s the second term is clearly dominant and the cost will decrease with increasing s. For increasing s the first term becomes dominant and the cost gradually becomes independent of n. For some intermediate values of s both terms are comparable and the total cost is seen to be much smaller than that of the eigendecomposition method (since $s \ll M$).

The storage requirements for all three methods are readily obtained. The Green's function method requires $\mathcal{O}(n^2)$ (mostly for storage of the dense $\boldsymbol{Z}_p$), while the eigendecomposition method requires $\mathcal{O}\left((u+1)M^2\right)$ with u being the the number of cosine modes per cell. Storage requirements for the PcDCT method is $\mathcal{O}(s^2 + 5n^2/s^2)$ where the first term corresponds to the DCT matrix, and the second term to the precorrection matrices. Dominance of each term as a function of s follows the behavior outlined for the computational cost and thus for appropriate values of s the PcDCT method is also more memory efficient.

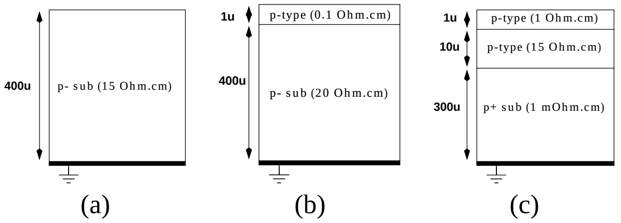

Figure 3: Example substrate profiles.

Method	**Green's**	**EigenD.**	**PcDCT**[1]	**PcDCT**[2]
Discretization	non-unif.	uniform	non-uniform	
# contacts	52	52	52	52
# panels	2647	17764	2647	2647
Avg # panels/contact	51	341	51	51
Size of DCT	512×512	256×256	32×32	64×64

Table 1: Relevant characteristics of the various solution methods when applied to the example problem.

It is easy to see that for layouts where the contacts are grouped into several separated clusters, an appropriate choice can be made such that the number of direct interactions is reduced and the method's performance will be superior. This coupled with the fact that for the precorrected-DCT method non-uniform panel discretizations are allowed, makes the method extremely efficient and allows for the extraction of larger, more complex circuits.

5 Computational Results

We present numerical experiments that show the efficiency of our extraction algorithm when compared to straightforward usage of the Green's function. For completeness we also include results for the basic (unaccelerated) eigendecomposition method, which allows us to account for the speedup provided by the proposed technique.

For an example problem with 52 contacts on a $256\mu m \times 256\mu m$ substrate [12], three different extractions were performed using different substrate profiles. The profiles used here were taken from [20] and have also been used elsewhere as benchmarks [11, 19]. The vertical substrate profiles are shown in Figure 3 and correspond respectively to technologies with single-layer, high resistivity and low resistivity substrates with grounded backplanes. An extraction was also performed using a *floating* single layer substrate. Table 1 shows the relevant characteristics of the example problem for the various solution methods. Note that accuracy constraints will limit the discretization used with each method. In these examples a non-uniform discretization was required for the Green's function method. Thus, a simple, yet efficient, non-uniform discretization algorithm was developed and it was also used with the precorrected-DCT algorithm. The standard eigendecomposition method, however, requires the use of a uniform discretization. The discretizations used were chosen such that similar accuracy was attained with all methods. The

Memory Usage	Green's	EigenD.	PcDCT^1	PcDCT^2
single layer	139MB	20.4MB	6.4MB	6.5MB
single + floating	139MB	20.7MB	6.3MB	6.4MB
high-resistivity	141.0MB	24.9MB	6.9MB	7.2MB
low-resistivity	138MB	22.5MB	6.4MB	6.6MB

Table 2: Memory requirements for complete extraction for the various methods.

CPU Time	Green's	EigenD.	PcDCT^1	PcDCT^2
single layer	25350	3018	243	294
single + floating	25944	3275	267	314
high-resistivity	123630	8406	687	830
low-resistivity	30966	4995	360	452

Table 3: CPU Time required for complete extraction for the various methods (seconds on a Sun Sparc Ultra-1).

eigendecomposition method has the highest accuracy because a minimal uniform discretization was used. For the PcDCT method the choice of s has, theoretically, a negligible effect on accuracy. The error in that method mainly depends on the *order* of the approximation scheme, as mentioned in Section 3.

Tables 2 and 3 detail respectively the memory and CPU time required by each method to extract the full substrate conductance matrix. The results in Table 2 show that the eigendecomposition method as well as the accelerated precorrected-DCT algorithm are considerably more efficient than the Green's function method in terms of memory requirements. The accelerated precorrected-DCT algorithm requires about 3 times less memory than its unaccelerated counterpart and almost 20 times less memory than the Green's function method. Table 3, on the other hand, shows that the precorrected-DCT method is extremely efficient in terms of computational cost showing speedups of over an order of magnitude over the unaccelerated version and as high as 180 over the Green's function method for the most CPU-intensive problem (on average two orders of magnitude speedup are obtained over the Green's function method).

6 Conclusions

In this paper we presented an efficient algorithm for extraction and modeling of substrate coupling effects in mixed-signal designs. The algorithm, based on a precorrected-DCT algorithm that can be used to accelerate operator application in BEM methods, was shown to be both computationally and memory efficient. Speedups of up to two orders for magnitude, together with significant memory savings were obtained on the examples presented, with controllable accuracy, thus enabling the analysis of large portions of mixed-signal integrated circuits.

Acknowledgments

This work was partially supported by the Defense Advanced Research Projects Agency, the National Science Foundation, the Portuguese JNICT programs PRAXIS XXI and FEDER under contracts 2/2.1/T.I.T/1661/95 and 2/2.1/T.I.T/1639/95 and grant BM-6853/95.

References

[1] David K. Su, Marc J. Loinaz, Shoichi Masui, and Bruce A. Wooley. Experimental results and modeling techniques for substrate noise in mixed-signal integrated circuits. *IEEE Journal of Solid-State Circuits*, 28(4):420–430, April 1993.

[2] T. A. Johnson, R.W. Knepper, V. Marcellu, and W. Wang. Chip substrate resistance modeling technique for integrated circuit design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, CAD-3(2):126–134, 1984.

[3] Ranjit Gharpurey. *Modeling and Analysis of Substrate Coupling in Integrated Circuits*. PhD thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA, June 1995.

[4] Bram Nauta and Gian Hoogzaad. How to deal with substrate noise in analog CMOS circuits. In *European Conference on Circuit Theory and Design*, pages Late 12:1–6, Budapest, Hungary, September 1997.

[5] Nishath Verghese. *Extraction and Simulation Techniques for Substrate-Coupled Noise in Mixed-Signal Integrated Circuits*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, August 1995.

[6] Sujoy Mitra, R. A. Rutenbar, L. R. Carley, and D. J. Allstot. A methodology for rapid estimation of substrate-coupled switching noise. In *IEEE 1995 Custom Integrated Circuits Conference*, pages 129–132, 1995.

[7] Balsha Stanisic, Nishath K. Verghese, Rob A. Rutenbar, L. Richard Carley, and David J. Allstot. Addressing substrate coupling in mixed-mode IC's: Simulation and power distribution systems. *IEEE Journal of Solid-State Circuits*, 29(3):226–237, March 1994.

[8] Ivan L. Wemple and Andrew T. Yang. Mixed-signal switching noise analysis using voronoi-tesselation substrate macromodels. In 32^{nd} *ACM/IEEE Design Automation Conference*, pages 439–444, San Francisco, CA, June 1995.

[9] T. Smedes, N. P. van der Meijs, and A. J. van Genderen. Extraction of circuit models for substrate cross-talk. In *International Conference on Computer Aided-Design*, San Jose, CA, November 1995.

[10] R. Gharpurey and R. G. Meyer. Modeling and analysis of substrate coupling in integrated circuits. In *IEEE 1995 Custom Integrated Circuits Conference*, pages 125–128, 1995.

[11] Nishath K. Verghese, David J. Allstot, and Mark A. Wolfe. Verification techniques for substrate coupling and their application to mixed-signal IC design. *IEEE Journal of Solid-State Circuits*, 31(3):354–365, March 1996.

[12] João P. Costa, Mike Chou, and L. Miguel Silveira. Efficient techniques for accurate modeling and simulation of substrate coupling in mixed-signal IC's. In *DATE'98 - Design, Automation and Test in Europe, Exhibition and Conference*, pages 892–898, Paris, France, February 1998.

[13] R. F. Harrington. *Field Computation by Moment Methods*. MacMillan, New York, 1968.

[14] K. Nabors and J. White. Fast capacitance extraction of general three-dimensional structures. *IEEE Transactions on Microwave Theory and Techniques*, June 1992.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipies in C*. Cambridge University Press, second edition, 1992.

[16] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, July 1986.

[17] V. Rohklin. Rapid solution of integral equation of classical potential theory. *J. Comput. Phys.*, 60:187–207, 1985.

[18] J. Phillips and J. White. A precorrected-FFT method for capacitance extraction of complicated 3-D structures. In *Proceedings of the Int. Conf. on Computer-Aided Design*, November 1994.

[19] Mike Chou and Jacob White. Multilevel integral equation methods for the extraction of substrate coupling parameters in mixed-signal ic's. In 35^{th} *ACM/IEEE Design Automation Conference*, pages 20–25, June 1998.

[20] Ranjit Gharpurey and Robert G. Meyer. Modeling and analysis of substrate coupling in integrated circuits. *IEEE Journal of Solid-State Circuits*, 31(3):344–353, March 1996.