Symbolic Circuit-Noise Analysis and Modeling with Determinant Decision Diagrams *

Xiang-Dong Tan

Monterey Design Systems
Sunnyvale, CA 94089
e-mail: xtan@mondes.com

C.-J. Richard Shi

Department of Electrical Engineering
University of Washington, Seattle, WA 98195
e-mail: cjshi@ee.washington.edu

Abstract— In this paper, a new symbolic noise analysis and modeling technique is presented. The new method exploits the sharing of symbolic expressions in the noise models by using a recently introduced graph, called determinant decision diagrams (DDDs), for symbolic determinant representations. With efficient DDD-based graph manipulations, we are able to generate the exact noise models for analog blocks. Symbolic noise analysis and modeling on real analog circuit examples are presented and compared with SPICE noise simulation.

1. Introduction

Noise behavior is an important characteristic of analog circuits, as it usually determines the fundamental limit of the performance of analog circuits.

Numerical noise analysis for analog circuits in DC steady-state can be carried out efficiently by using the adjoint method [6]. But for system-level noise simulation, the adjoint method still can not offer adequate efficiency. Hierarchical noise analysis becomes an attractive alternative. In this method, noise models in terms of noise spectral densities at an output is first constructed for each circuit block and system-level noise is then analyzed by using the block-level noise models. Algorithms to generate approximated analog noise models have been reported in [1] by using numerical order-reduction techniques and in [2] by using symbolic approximation.

Simplified noise models, however, are only valid in a limited range of frequencies. In this paper we present a new approach to deriving the exact noise model in a closed rational form using a new symbolic analysis technique. The new method is based on determinant decision diagrams (DDDs) to represent the determinants of circuit matrices [7]. The new noise modeling technique consists of symbolic transfer function computation and the generation of noise spectral density functions by DDD-based graph manipulations. Such an application scenario is extremely amenable to symbolic analysis based on DDDs due to a huge amount of symbolic-term sharing among the transfer functions. Experimental results on practical analog circuits show that the computational cost for obtaining a number of transfer functions required by a noise model is almost equal to that of computing a single transfer function. By combining the exact noise models of analog blocks with DDD-based symbolic analysis, a significant speedup over SPICE-based noise simulation can be achieved.

The rest of the paper is organized as follows: Section 2 reviews device noise models and symbolic noise analysis. Section 3 presents our method for symbolic transfer function computation using determinant decision diagrams, DDD-based noise analysis and modeling techniques. Section 4 describes experimental results. Section 5 concludes the paper.

2. Noise Models and Symbolic Noise Analysis

1. Noise Models

Noise in integrated circuits is caused by some random physical phenomena which lead to small current and voltage fluctuations within circuit devices. Mathematically, noise is characterized in terms of mean and autocorrelation in the time domain, or the power spectral density in the frequency domain. The integration of noise spectral density over frequency gives the total noise power. The most important noise sources in integrated circuit devices are *thermal noise*, *shot noise* and *flicker noise* [4]

1. *Thermal* noise is also called white noise due to its independence of frequency. It is caused by the thermal motion of electrons. The spectral density function of thermal noise, $i_{th}^2/\Delta f$, can be given by

$$i_{th}^2/\Delta f = \frac{4KT}{R} \tag{1}$$

where K is Boltzman's constant ($1.38\times10^{-23}JK^{-}1$), T is the absolute temperature in Kelvin, and R is the resistance value.

*This work was sponsored by U.S. Defense Advanced Research Projects Agency (DARPA) under grant number F33615-96-1-5601 from the United States Air Force, Wright Laboratory, Manufacturing Technology Directorate, by Rockwell Semiconductor Systems, and by NSF/Industry Center for Design of Analog-Digital Integrated Circuits (CDADIC).

2. *Shot* noise is caused by the current through a PN junction consisting of discrete charge carriers randomly crossing a potential barrier. The noise spectral density of shot noise, $i_{sh}^2/\Delta f$, can be given by

$$i_{sh}^2/\Delta f = 2qI_d \tag{2}$$

where q is the electron charge and I_d the average junction current.

3. *Flicker* (or 1/f) noise can be found in all active devices. The origins of flicker noise are varied in different devices and are not well understood. The spectral density of flicker noise, $i_{fl}^2/\Delta f$, is given by

$$i_{fl}^2/\Delta f = K_{device}\frac{I^a}{f} \tag{3}$$

where K_{device} is a constant for a particular device and a is a constant in the range from 0.5 to 2.

In analog integrated circuits, resistors, bipolar junction transistors (BJTs) and MOSFET transistors are all noisy devices. Each of them may include several noise sources due to different physical phenomena. The widely used noise models for these devices are shown in Fig. 1 [4, 5]. Each noise source is defined as follows:

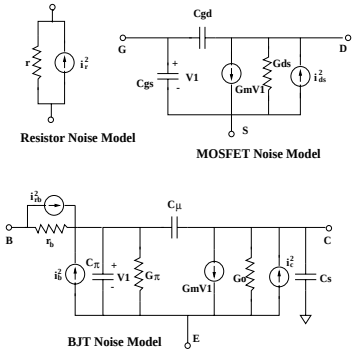

Fig. 1: Noise models for common IC devices.

$$i_r^2 = \frac{4KT}{r}\Delta f \tag{4}$$

$$i_{rb}^2 = \frac{4KT}{r_b}\Delta f \tag{5}$$

$$i_b^2 = 2qI_b\Delta f + K_{bjt}\frac{I^a}{f}\Delta f \tag{6}$$

$$i_c^2 = 2qI_d\Delta f \tag{7}$$

$$i_{ds}^2 = 4KT(2/3)g_m\Delta f + K_{mos}\frac{I^a}{f}\Delta f \tag{8}$$

2. Symbolic Noise Analysis

Since all the noise sources are uncorrelated and their contributions to an output can be calculated separately. Let $H_j(s)$ denote the transfer function from noise source i_j^2 to an output, then the noise voltage in the root mean square form (rms) at the output is given by

$$V_{out}(s) = \sqrt{\sum_{j=1}^{n} |H_j(s)|^2 i_j^2} \tag{9}$$

$V_{out}^2(s)/\Delta f$ is called output noise spectral density. With this, computing the noise spectral density function of an output essentially amounts to symbolic computation of a set of transfer functions with different inputs and the same output, and consequent squaring operations of each transfer function, and summation of the results. For practical circuits, however, one transfer function is already very lengthy, a number of symbolic functions along with the squaring operations will lead to a huge number of product terms in the resulting s-expanded form [2, 3]. So noise analysis in symbolic analyzer ISAAC [2] is limited to small analog circuits.

3. New Circuit Noise Modeling Method

We now introduce a new symbolic noise analysis and modeling method. It consists of two steps: (1) deriving the symbolic transfer function of each noise source, and (2) computing the noise spectral density function using s-expanded DDDs and algebraic operations.

1. Symbolic Transfer Function Computation

For a linear(ized), time-invariant analog circuit, we can use the modified nodal analysis formulation to describe its system of equations as follows:

$$\mathbf{Yx} = \mathbf{b} \tag{10}$$

where $\mathbf{x}$ is a vector of the node voltage variables and branch current variables, $\mathbf{Y}$ is the nodal admittance matrix and $\mathbf{b}$ represents the independent noise sources. We first present how a symbolic transfer function from a single noise source shown in Fig. 2 is derived by using DDDs. In

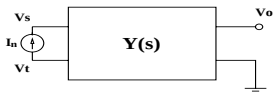

Fig. 2: A network with a single noise source.

this case, the excitation vector $\mathbf{b} = \{0, .., I_n, ..., -I_n, ..., 0\}$ has only two nonzero elements in row s and t. According to Cramer's rule, the transfer function from input current source I_n to the output V_o can be expressed as

$$H(s) = \frac{V_o}{I_n} = \frac{(-1)^{(s+o)}det(Y_{so}) - (-1)^{(t+o)}det(T_{to})}{det(Y)} \tag{11}$$

where Y_{so} is the matrix obtained by deleting row s and column o in Y. We note that all the transfer functions have the same denominator, $det(Y)$, but different numerators consisting of different first-order cofactors of $det(Y)$. So computing all the trans-resistance functions from various inputs to the an output is equal to representing $det(Y)$ and a number of its first-order cofactors. DDDs are extremely efficient to represent the expressions in a determinant and its cofactors due to the sharing of subsexpressions among them and the exploitation of such sharing.

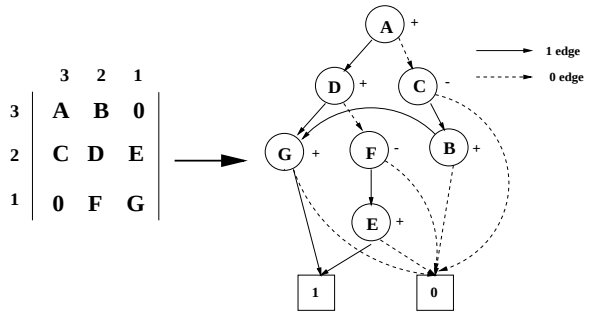

Fig. 3: A matrix determinant and its DDD.

A DDD example is shown in Fig. 3 with the corresponding determinant shown in the left-hand side. A DDD is a signed, rooted, directed, acyclic graph with two terminal vertices, namely the *0-terminal* vertex and the *1-terminal* vertex. Each non-terminal vertex is labeled by a symbol in the determinant denoted by a symbol a_i, and a sign denoted by $s(a_i)$. It *originates* two outgoing edges, called *1-edge* and *0-edge*. Each vertex a_i represents a symbolic expression $D(a_i)$ defined recursively as follows:

1. if a_i is the 1-terminal vertex, then $D(a_i) = 1$,
2. if a_i is the 0-terminal vertex, then $D(a_i) = 0$,
3. otherwise, $D(a_i) = a_i\ s(a_i)\ D_{a_i} + D_{\overline{a}_i}$.

where D_{a_i} and $D_{\overline{a}_i}$ represent, respectively, the vertices pointed by the 1-edge and 0-edge of a_i. A *1-path* in a DDD corresponds a product term in the original DDD, which is defined as a path from the root vertex (A in our example) to the 1-terminal including all symbolic symbols and signs of the vertices that originate all the 1-edges along the 1-path. In our example, there exist three 1-paths representing three product terms: ADG, $-AFE$ and $-CBG$. The root vertex represents the sum of these product terms.

It is shown in [7] that given a proper vertex ordering, the number of DDD vertices used to represent all the product terms in a symbolic determinant is order of magnitudes less than the number of product terms. This enables DDDs-based symbolic analysis to be able to handle circuits substantially larger than those can be handled by traditional symbolic methods.

2. Noise Analysis

Instead of presenting our method in a formal way, we illustrate this multi-function computation process by means of a simple RC filter shown in Fig. 4. The noise sources of noisy resistors in the circuit are also marked in the figure. The circuit matrix of the RC filter can be

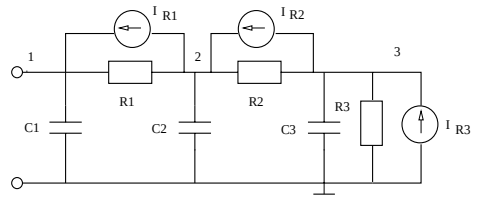

Fig. 4: An RC circuit example.

written as

$$\begin{bmatrix} \frac{1}{R_3} + sC_3 + \frac{1}{R_2} & -\frac{1}{R_2} & 0 \\ -\frac{1}{R_2} & \frac{1}{R_2} + sC_2 + \frac{1}{R_1} & -\frac{1}{R_1} \\ 0 & -\frac{1}{R_1} & \frac{1}{R_1} + sC_1 \end{bmatrix}$$

If we view each entry as one symbol, the resulting system determinant is shown in the left-hand side of Fig. 3. For each noise source, we have a transfer function:

$$H_{R_1}(s) = \frac{V_3}{I_{R_1}} = \frac{(-1)^{(1+3)} det(Y_{13}) - (-1)^{(2+3)} det(Y_{23})}{det(Y)}$$

$$H_{R_2}(s) = \frac{V_2}{I_{R_2}} = \frac{(-1)^{(2+3)} det(Y_{23}) - (-1)^{(3+3)} det(Y_{33})}{det(Y)}$$

$$H_{R_3}(s) = \frac{V_3}{I_{R_3}} = \frac{(-1)^{(3+3)} det(Y_{33})}{det(Y)}$$

Note that, in addition to the system determinant $det(Y)$, we also need three first order minors of $det(Y)$: $det(Y_{13}) = BE$, $det(Y_{23}) = BG$ and $det(Y_{33}) = DG - FE$. In fact, minor $det(Y_{33})$ and $det(Y_{23})$ already exist in $det(Y)$ as shown in Fig. 5. To represent $det(Y_{13})$, we need two extra DDD vertices. So we end up with a *total* of 9 vertices to represent all the transfer functions required for the output noise spectral density for this RC filter circuit.

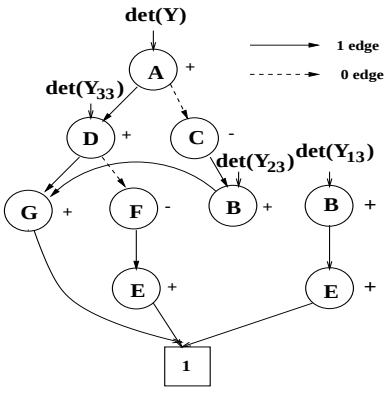

Fig. 5: DDD for representing noise transfer function.

3. Generation of Noise Spectral Density Functions

In hierarchical noise simulation, we first construct the noise model in terms of noise spectral density, $V_j^2(f)/\Delta f$, for each analog block j. Such noise models are function

in only frequency variable f so that it can be used as the lumped noise sources of the block in the higher level noise simulation.

To this end, we expand the transfer function into s-expended form where each coefficient of s^x is explicitly expressed by a *coefficient* DDD and the whole s-expanded polynomial is represented by an *s-expanded DDD* [8]. Consider the RC circuit example, we can rewrite the system equations of circuit in Fig. 4 into the following form:

$$\begin{bmatrix} a+bs & c & 0 \\ d & e+fs & g \\ 0 & h & i+js \end{bmatrix}$$

If we expand three product terms $(a+sb)(e+fs)(i+js)$, $(a+sb)(-h)(g)$ and $(-d)(c)(i+js)$ into the s-expended form, we have 12 product terms with different powers of s. We then group all the these product terms according to their powers of s, we then have the following four groups:

1. $aeis^0$,$-ahgs^0$, $-dcis^0$
2. $aejs^1$,$afis^1$,$beis^1$, $-bhgs^1$, $-dcjs^1$
3. $afjs^2$, $bejs^2$, $bfis^2$
4. $bfjs^3$

For each group we build a DDD representing all the product terms in the group, each DDD tree is the coefficient DDD and the whole DDD is the s-expanded DDD as shown in Fig. 6. It is shown that such an s-expanded

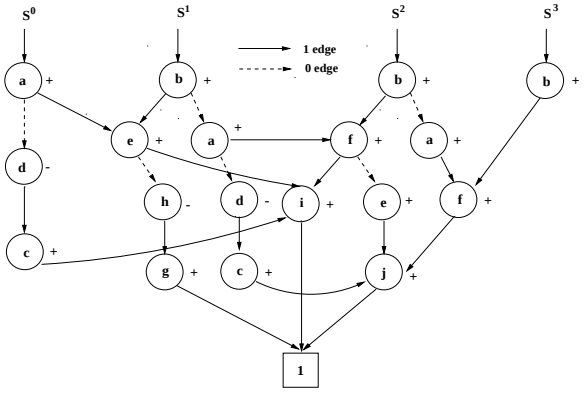

Fig. 6: Coefficient DDD for a matrix determinant.

DDD can be obtained in a very efficient way such that the time and space complexity are directly proportional to the size of original DDD and the highest power of s in the resulting s-expanded polynomials [8].

With s-expanded DDDs, we can easily obtain the numerical value of each coefficient DDD by efficient DDD evaluation operations [7]. The resulting transfer functions will take the following s-expanded rational form:

$$f(s) = \frac{\sum_{i=0}^{m} a_i s^i}{\sum_{j=0}^{n} b_j s^j} \tag{12}$$

where a_i and b_j are all numerical coefficients. From this form, we can easily perform the squaring operation to obtain $(f(s))^2$. Since all the device noise sources in terms of noise spectral density from circuit devices are either a constant or a linear function in frequency f, we can easily obtain the noise spectral density at the output by simple algebraic multiplications and additions as the denominators in all the transfer function are the same (determinant of the circuit matrix). Hence the noise spectral density function is still a rational function in f.

4. Experimental Results

The proposed algorithm has been implemented in a symbolic analyzer [7]. Experimental results from the state variable filter circuit shown in Fig. 7 are presented. The

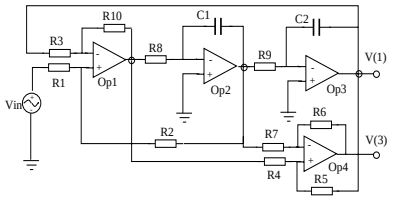

Fig. 7: State variable filter circuit.

filter circuit consists of 4 identical operational amplifiers (Opamps), which in turn are implemented by a CMOS cascode Opamp with 22 noisy MOSFETs as shown in Fig. 8. We first construct the noise model of the CMOS

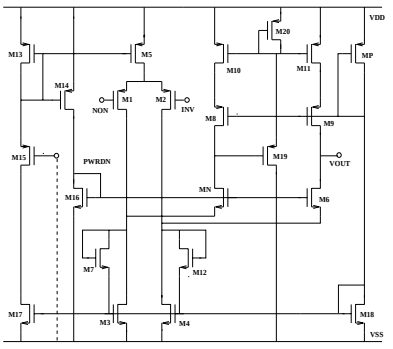

Fig. 8: CMOS cascode operational amplifier.

cascode Opamp. The noise of an Opamp circuit can be modeled by three independent noise sources as shown in the left-hand side of Fig. 9.

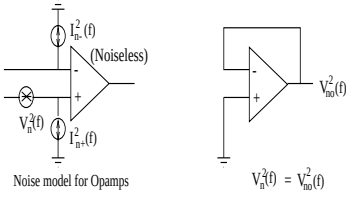

Fig. 9: Noise model for operational amplifier.

For an Opamp with MOSFET input stage, the two current, $I^2_{n-}(f), I^2_{n+}(f)$, noise sources can be ignored at low

frequencies. The voltage noise source, V_n^2, can be obtained by using the circuit configuration shown in the right-hand side of Fig. 9. All the current noise sources in circuit devices are taken from SPICE and fed into our program. The number of DDD vertices for representing the determinant of the circuit matrix of Cascode Opamp circuit is 5242, while the total number of DDD vertices used for all the transfer functions (about 44) is just 10636. This clear shows the power of DDD in exploiting the sharing among different transfer functions.

The computed exact voltage noise spectral density function for Cascode Opamp is shown as follows:

$$
\begin{array}{lll}
v^2(f)/\Delta f = & & \\
+1.49 \times 10^{-321} f^{27} & +1.72 \times 10^{-312} f^{26} & +2.66 \times 10^{-301} f^{25} \\
+3.11 \times 10^{-292} f^{24} & +5.83 \times 10^{-283} f^{23} & +1.10 \times 10^{-273} f^{22} \\
+6.17 \times 10^{-265} f^{21} & +1.61 \times 10^{-255} f^{20} & +3.66 \times 10^{-247} f^{19} \\
+1.26 \times 10^{-237} f^{18} & +1.12 \times 10^{-229} f^{17} & +5.40 \times 10^{-220} f^{16} \\
+1.45 \times 10^{-212} f^{15} & +1.13 \times 10^{-202} f^{14} & +7.39 \times 10^{-196} f^{13} \\
+9.87 \times 10^{-186} f^{12} & +8.22 \times 10^{-179} f^{11} & +3.57 \times 10^{-169} f^{10} \\
+1.73 \times 10^{-162} f^{9} & +3.81 \times 10^{-152} f^{8} & +8.48 \times 10^{-146} f^{7} \\
-8.68 \times 10^{-137} f^{6} & +2.27 \times 10^{-129} f^{5} & +1.47 \times 10^{-119} f^{4} \\
+5.17 \times 10^{-114} f^{3} & +2.41 \times 10^{-104} f^{2} & +2.61 \times 10^{-99} f^{1} \\
+4.52 \times 10^{-90} & & \\
\hline
f \cdot (5.73 \times 10^{-323} f^{28} & +1.38 \times 10^{-302} f^{26} & +3.85 \times 10^{-284} f^{24} \\
+3.78 \times 10^{-266} f^{22} & +1.42 \times 10^{-248} f^{20} & +2.52 \times 10^{-231} f^{18} \\
+4.32 \times 10^{-214} f^{16} & +4.91 \times 10^{-197} f^{14} & -1.90 \times 10^{-180} f^{12} \\
+2.43 \times 10^{-163} f^{10} & -1.43 \times 10^{-146} f^{8} & +5.90 \times 10^{-131} f^{6} \\
+7.18 \times 10^{-114} f^{4} & +2.13 \times 10^{-98} f^{2} & +1.21 \times 10^{-83}) \\
& & (13)
\end{array}
$$

We then perform noise analysis on the state-variable filter circuit, where each Opamp circuit is treated as a noiseless circuit with an input voltage noise source (derived above) cascaded at the positive terminal. Each transfer function from a noise source to the output in the filter circuit is computed by the DDD-based hierarchical decomposition method [9]. The resulting transfer functions are expanded into s-expanded forms. We then compute all the numerical coefficients of s^i in each transfer function. The resulting transfer functions are functions in only s or frequency f.

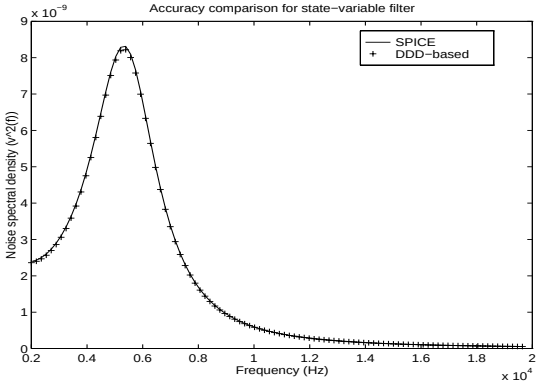

Fig. 10: Noise spectral densities by SPICE and DDD.

Figure 10 shows the noise spectral density of the state variable filter circuit computed by SPICE and by the proposed DDD-based method. It can be seen that the results are identical. We note that, the noise spectral density calculation from the resulting symbolic expression took 0.03 second, while SPICE noise analysis took 1.69 second. We used a Linux platform with an Intel Pentium-II CPU at 450 MHz clock rate, and 1000 frequency points were computed.

5. Conclusions

An efficient symbolic noise analysis and modeling technique has been proposed. It consists of symbolic computations of transfer functions and noise model generations by means of DDD-based graph manipulations. In contrast to other noise modeling techniques, the new method is capable of generating exact noise models of analog circuits efficiently. Experimental results have demonstrated the advantage of the proposed method over the simulation-based methods for the noise analysis of practical circuits.

References

[1] P. Feldmann and R. W. Freund, "Circuit noise evaluation by Pade approximation based model-reduction techniques," in *Proc. IEEE Int. Conf. Computer Aided Design (ICCAD)*, pp. 132-138, 1997.

[2] G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*, Kluwer Academic Publishers, 1991.

[3] G. Gielen, P. Wambacq and W. Sansen, "Symbolic analysis methods and applications for analog circuits: A tutorial overview," *Proc. IEEE*, vol. 82, no. 2, pp. 287–304, Feb. 1994.

[4] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, John Wiley & Sons, Inc., New York, 1984.

[5] L. W. Nagel, *SPICE2: A Computer Program to Simulate Semiconductor Circuits*, Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of California, Berkeley CA, May 1975.

[6] R. Rohrer, L. Nagel, R. Mayer and L. Weber, "Computationally efficient electronic-circuit noise calculations," *IEEE Journal of Solid-State Circuits*, vol. SC-6, no. 4, pp. 204–213, Aug. 1971.

[7] C.-J. Shi and X.-D. Tan, "Symbolic analysis of large analog circuits with determinant decision diagrams," in *Proc. IEEE Int. Conf. Computer Aided Design (ICCAD)*, pp. 366-373, 1997.

[8] C.-J. Shi and X.-D. Tan, "Efficient derivation of exact s-expended symbolic expressions for behavioral modeling of analog circuits," in *Proc. IEEE Custom Integrated Circuits Conference(CICC)*, pp. 463-466, 1998.

[9] X.-D. Tan and C.-J. Shi, "Hierarchical symbolic analysis of large analog circuits with determinant decision diagrams," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 318-321, June 1998.

[10] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, New York, 1994.