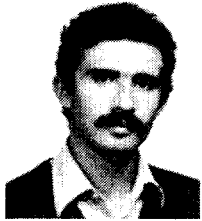


- [18] M. R. Garey and D. S. Johnson, *Computers and Intractability*. San Francisco: W. H. Freeman, 1978.
- [19] E. L. Lawler, "Optimal sequencing of a single machine subject to precedence constraints," *Manag. Sci.*, vol. 19, no. 5, pp. 544-546, Jan. 1973.

\*



Giovanni De Micheli (S'79) was born in Milano, Italy, in 1955. He received the Dr.Eng. degree (summa cum laude) in nuclear engineering from Politecnico di Milano, Italy, in 1979 and the M.S. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1980, where he is currently a doctoral candidate.

He spent the fall of 1981 as Consultant in Residence at Harris Semiconductor, Melbourne, FL. His research interests include several aspects of the computer-aided design of integrated circuits, with particular emphasis on simulation of large scale circuits and the automated synthesis of functional units for VLSI digital circuits.

Mr. De Micheli was granted the Fulbright Scholarship in 1980, the Rotary International Fellowship in 1981 and the IBM Fellowship for VLSI in 1982 and 1983. At The 1983 IEEE-ACM Design Automation Conference, he was given the Best Paper Award.



Alberto Sangiovanni-Vincentelli (M'74-SM'81-F'83) received the Dr.Eng. degree (summa cum laude) from the Politecnico di Milano, Italy, in 1971. From 1971 to 1977 he was with the Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano, Italy, where he held the position of Research Associate. Assistant and Associate Professor. In 1976 he joined the Department of Electrical Engineering and Computer Sciences, where he is presently Professor and Vice Chairman.

He is a consultant in the area of computer-aided design to several industries. His research interests are in various aspects of computer-aided design of integrated circuits, with particular emphasis on VLSI simulation and optimization.

Dr. Sangiovanni-Vincentelli was Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and is Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and a member of the Large-Scale Systems Committee of the IEEE Circuits and Systems Society. He was the Guest Editor of a special issue of the TRANSACTIONS ON CIRCUITS AND SYSTEMS ON CAD for VLSI. He has been elected Executive Vice-President of the IEEE Circuits and Systems Society in 1983.

In 1981 he received the Distinguished Teaching Award of the University of California. At the 1982 IEEE-ACM design Automation Conference he was given a Best Paper and a Best Presentation Award. In 1983 he received the Quillemin-cauer Award for the best paper published in the IEEE TRANSACTIONS ON CAS and CAD in 1981-1982.

Dr. Sangiovanni-Vincentelli is a member of ACM and Eta Kappa Nu.

## Symmetric Displacement Algorithms for the Timing Analysis of Large Scale Circuits

GIOVANNI DE MICHELI, STUDENT MEMBER, IEEE, ARTHUR RICHARD NEWTON, MEMBER, IEEE, AND ALBERTO SANGIOVANNI-VINCENTELLI, FELLOW, IEEE

**Abstract**—Symmetric displacement techniques for the timing analysis of VLSI circuits are introduced. Their numerical properties such as stability and accuracy are investigated on different classes of circuits.

### I. INTRODUCTION

WHILE CIRCUIT SIMULATORS (e.g., [1], [2]) can provide accurate time-domain current and voltage waveforms from a device level description of an integrated circuit, as the size of the circuit increases, the cost and memory requirements of such an analysis become prohibitive. For small circuits, the simulation time is generally dominated by the

time required to evaluate device model equations [3], but as circuit size increases, or if more efficient modeling techniques are used, an increasing fraction of time is spent solving the sparse-matrix circuit equations [3], [4].

Several methods that exploit regularity and inactivity of large circuits have been proposed to speed up the transient simulation. Among these methods, reviewed in [5], timing simulators occupy a relevant position.

Timing simulators decouple the circuit equations using nonlinear simultaneous displacement [6] or successive displacement [7], [8] relaxation methods. In MOTIS [7], the Backward Euler formula was used to discretize the time derivative operator, and a nonlinear Gauss-Jacobi-like relaxation technique [9] was adopted to decouple the node equations at the nonlinear equation level. The algorithms of the timing simulators MOTIS-C [7] and SPLICE1 [8] perfected this technique. In particular, SPLICE used a nonlinear "Gauss-Seidel-like" technique. For most circuits, this decoupling maintains a

Manuscript received September 24, 1982; revised March 29, 1983. This work was supported by The Rotary Foundation of Rotary International and by the Defense Advanced Research Projects Agency (DOD) under Contract N00039-K-0251.

The authors are with the Department of Electrical Engineering and Computer Sciences and the Electronics Research Laboratory, University of California, Berkeley, CA 94720.

linear relationship between the number of circuit elements and the simulation time required per time-point of the analysis. An added advantage of the decoupled analysis is that event-driven selective trace algorithms may be used easily, and independent control of the time-step within any sub-block of the circuit is also possible. These techniques can provide substantial savings in large digital circuits where often only a small fraction of the circuit nodes are actively changing state at any one time [10]. It is important to point out that none of these programs carried the iteration of the relaxation methods to convergence: only one sweep was taken. Because of this, the numerical properties such as stability of the integration formulas used to discretize the derivative operator no longer hold. These methods have indeed to be considered as *new integration methods*.

A major drawback with the use of timing analysis is that tightly coupled feedback loops, or bidirectional circuit elements, can cause severe inaccuracies during the analysis. In a "one-sweep" relaxation-based circuit simulator, these elements must be treated as special-case elements and special techniques must be used. One such element that has limited the application of relaxation-based analysis is the floating capacitor.

A *floating capacitor* is a capacitor whose nodes are connected neither to ground nor to a fixed voltage source. Floating capacitors are often important elements in the design and in the characterization of integrated circuits. As an example, a bootstrapped inverter is shown in Fig. 1(a). The value of the bootstrap capacitor  $C_b$  is generally large compared to the values of the associated parasitic grounded capacitors  $C_1$  and  $C_2$ . Fig. 1(b) shows a depletion-load NMOS inverter. The value of the intrinsic gate-drain feed-through capacitance  $C_{gd}$  is often small compared to other circuit parasitics at the gate and drain nodes. However, the effect of  $C_{gd}$  on circuit performance is significant due to the large voltage gain of the stage.

The MOTIS program [6] avoids the problem of analyzing floating capacitors by not allowing the user to include them in the circuit description. The effect of a floating capacitor is then approximated by altering the values of the grounded capacitors at appropriate nodes in the circuit. If the operation of a circuit depends on a floating capacitor, a functional macro-model may be used [11].

In the MOTIS-C program [7], isolated floating capacitors are processed by maintaining the node coupling across the floating branch and solving the resulting  $2 \times 2$  nodal circuit matrix at each time-point. This approach could be extended to deal with arbitrary connections of  $N$  floating capacitors, but this would require the solution of  $N + 1$  coupled equations at each time-point and, hence, reduce the advantages of the node decoupling approach.

In [12], an integration scheme is proposed that retains the nice properties of timing simulation algorithms and provides better accuracy. While this approach has proven effective for timing simulation on a range of MOS circuits, the numerical properties of this scheme can be demonstrated only on some special cases at this time.

Recently, Kahan has proposed a family of "one-sweep"

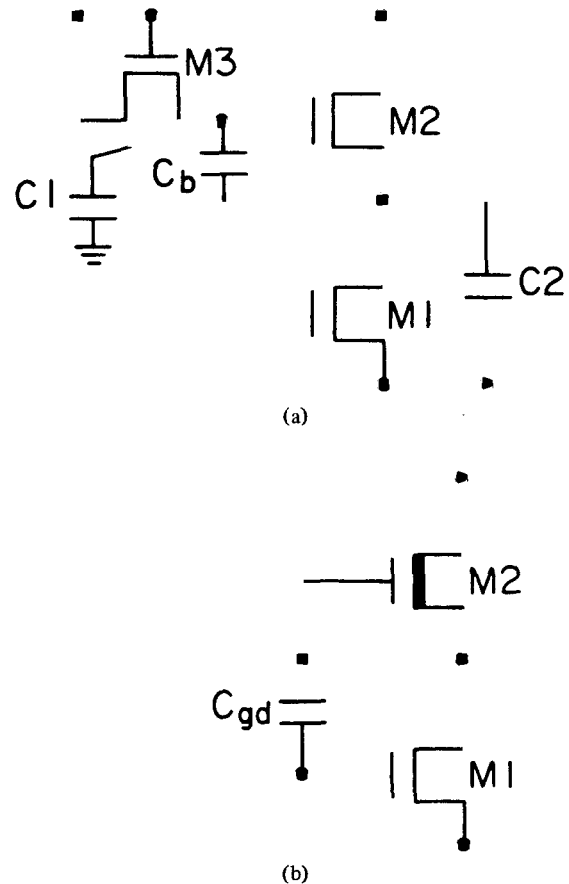


Fig. 1. (a) Enhancement-load nMOS bootstrapped inverter. (b) Depletion load nMOS inverter.

symmetric displacement methods for the integration of large systems of ordinary differential equations [13]. One of the methods in the family has been analyzed in [14], [15] for the timing simulation of large MOS digital circuits with no floating capacitors. This method is shown to have better numerical properties than any of the "one-sweep" displacement techniques used in timing simulation. However, the application of these techniques to the case when circuit elements include floating capacitors is an open problem.

In this paper, we propose a number of new techniques for the time-domain simulation of MOS circuits based on "one-sweep" symmetric displacement methods. Following standard numerical analysis procedures for the characterization of integration methods, we investigate rigorously their numerical properties by introducing test problems that are simple enough to be studied analytically and yet complex enough to provide insight on how they will behave in general. Then, we describe the implementation of these methods in timing simulators and give some experimental results that emphasize the better accuracy and stability of symmetric displacement techniques.

## II. SYMMETRIC DISPLACEMENT TECHNIQUES

The MOS circuits considered here are assumed to be lumped, time-invariant, nonlinear circuits where inductive effects can be considered negligible, and each node has a capacitance to ground. It is also assumed that the node equations of the circuit exist and can be written in the form

$$C(v(t))\dot{v}(t) = -f(v(t), u(t)), \quad 0 \leq t \leq T \quad (2.1)$$

$$v(0) = v^0$$

where  $v(t) \in \mathbb{R}^n$  is the vector of node voltages at time  $t$ ,  $\dot{v}(t) \in \mathbb{R}^n$  is the vector of time derivatives of  $v(t)$ ,  $u(t) \in \mathbb{R}^n$  is the vector of node input currents at time  $t$ ,  $C(\cdot) \triangleq \{c_{i,j}\}: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  represents the nodal capacitance matrix,  $f: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ , and

$$f(v, u(t)) = [f_1(v, u(t)), f_2(v, u(t)), \dots, f_n(v, u(t))]^T \quad (2.2)$$

where  $f_i(v, u(t))$  is the sum of the currents flowing out of the capacitors connected to node  $i$ . To solve (2.1), the time derivative operator is discretized to yield a nonlinear algebraic system of equations which must be solved at each time step. For the sake of simplicity, it is assumed that the Backward Euler integration formula is used

$$\dot{v}^{k+1} = \frac{v^{k+1} - v^k}{h} \quad (2.3)$$

where  $\dot{v}^{k+1}$  is the vector of time derivatives of  $v$  computed at  $t^{k+1}$ ,  $v^p$ ,  $p = k, k+1$  is the node voltage vector computed at time  $t^p$ , and  $h \triangleq t^{k+1} - t^k$  is the strictly positive stepsize.

Applying (2.3) to (2.1)

$$g(v^{k+1}, v^k, u(t^{k+1})) = \frac{C(v^{k+1})}{h} v^{k+1} - \frac{C(v^k)}{h} v^k + f(v^{k+1}, u(t^{k+1})) = 0. \quad (2.4)$$

Conventional circuit simulators [1], [2] solve (2.3) by using the Newton-Raphson method to yield

$$Y^{k+1,j} [v^{k+1,j+1} - v^{k+1,j}] = -g(v^{k+1,j}, v^k, u(t^{k+1})) \quad (2.5)$$

where  $v^{k+1,p}$ ,  $p = j, j+1$  is the node voltage vector  $v^{k+1}$  computed at the  $p$ th Newton-Raphson iteration, and  $Y^{k+1,j}$  is defined by

$$Y^{k+1,j} = \frac{\partial g}{\partial v^{k+1}}(v^{k+1,j}, v^k, u(t^{k+1})) = \frac{\partial f}{\partial v^{k+1}}(v^{k+1,j}, u(t^{k+1})) + \frac{C(v^{k+1,j})}{h} + \frac{1}{h} \frac{\partial C}{\partial v^{k+1}}(v^{k+1,j}) [v^{k+1,j} - v^k]. \quad (2.6)$$

In the linear case, (2.4) becomes

$$Y v^{k+1} = \frac{C}{h} v^k + u(t^{k+1}) \quad (2.7)$$

where  $Y \triangleq \tilde{G} + C/h$ ;  $\tilde{G}$ ,  $C \in \mathbb{R}^{n \times n}$  are the conductance and node capacitance matrices of the circuit.

Standard circuit simulators solve (2.5) or (2.7) using direct methods, such as sparse Gaussian Elimination, while timing simulators approximate the solution of (2.4) by taking only one step of a nonlinear displacement method such as Gauss-Jacobi [6] or Gauss-Seidel [7], [8].

The key idea of this paper is to use symmetric displacement techniques for equation solution in MOS electrical simulators. Three algorithms, which use symmetric displacement techniques at different levels of the solution process,

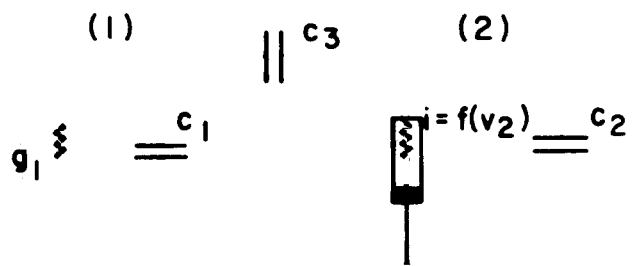


Fig. 2. Test circuit.

are presented. Algorithm 1 uses only one Newton-Raphson iteration to approximate the solution of (2.4) and only one iteration of the symmetric displacement Gauss-Seidel algorithm [9] to approximate the solution of (2.5). Algorithm 2 uses one iteration of nonlinear symmetric Gauss-Seidel algorithm to approximate the solution of (2.4). Algorithm 3 uses an intermediate time-step  $t^{k+1/2} = t^k + h/2$ . For the first half-step, one iteration of the forward nonlinear Gauss-Seidel algorithm is taken to compute  $v^{k+1/2}$ . The second half-step uses one iteration of the backward nonlinear Gauss-Seidel algorithm to compute  $v^{k+1}$  given  $v^{k+1/2}$ .

### 2.1. Algorithm 1

Only one Newton-Raphson step is taken here, and the initial value for  $v^{k+1}$  is  $v^k$ . Then (2.6) becomes

$$Y^{k+1,1} \triangleq Y^k = \frac{\partial f}{\partial v^{k+1}}(v^k, u(t^{k+1})) - \frac{C(v^k)}{h} \quad (2.8)$$

and (2.5) can be written

$$Y^k v^{k+1} = \frac{C(v^k)}{h} v^k + \frac{\partial f}{\partial v^{k+1}}(v^k, u(t^{k+1})) v^k - f(v^k, u(t^{k+1})) \quad (2.9)$$

$$= Y^k v^k - f(v^k, u(t^{k+1})) \triangleq b^k. \quad (2.10)$$

Now consider the splitting

$$Y^k = D^k - Y_l^k - Y_u^k \quad (2.11)$$

where  $D^k \in \mathbb{R}^{n \times n}$  is a diagonal matrix and  $Y_l^k (Y_u^k) \in \mathbb{R}^{n \times n}$  is a strictly lower (upper) triangular matrix. Note that  $D^k$  is nonsingular since it was assumed that each node of the circuit had a capacitance to ground.

The solution of (2.9) is now approximated by taking one symmetric Gauss-Seidel step such that  $v^{k+1}$  is computed by the following sequence.

Forward step:

$$[D^k - Y_l^k] v^* = b^k + Y_u^k v^k. \quad (2.12a)$$

Backward step:

$$[D^k - Y_u^k] v^{k+1} = b^k + Y_l^k v^* \quad (2.12b)$$

where  $v^*$  is an intermediate node voltage vector. Note that (2.10) can be solved easily since the coefficient matrices for (2.12a) and (2.12b) are lower and upper triangular, respectively.

*Example 2.1.* If Algorithm 1 is applied to the circuit shown in Fig. 2, from (2.8)

$$Y^k = \begin{bmatrix} g_1 + \frac{c_1 + c_3}{h} & -\frac{c_3}{h} \\ -\frac{c_3}{h} & g_2 + \frac{c_2 + c_3}{h} \end{bmatrix} \quad (2.13)$$

where

$$g_2^k = \frac{\partial f(v_2^k)}{\partial v_2} \quad (2.14)$$

$$C^k = C = \begin{bmatrix} c_1 + c_3 & -c_3 \\ -c_3 & c_2 + c_3 \end{bmatrix} \quad (2.15)$$

and

$$b^k = Y^k v^k - \begin{bmatrix} 0 \\ f(v_2^k) \end{bmatrix}. \quad (2.16)$$

The forward step is computed as follows:

$$\begin{bmatrix} g_1 + \frac{c_1 + c_3}{h} & 0 \\ -\frac{c_3}{h} & g_2 + \frac{c_2 + c_3}{h} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^* \\ = \begin{bmatrix} 0 & \frac{c_3}{h} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^k + \begin{bmatrix} g_1 + \frac{c_1 + c_3}{h} & -\frac{c_3}{h} \\ -\frac{c_3}{h} & g_2 + \frac{c_2 + c_3}{h} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^k \\ - \begin{bmatrix} 0 \\ f(v_2^k) \end{bmatrix} \quad (2.17a)$$

and the backward step computed as

$$\begin{bmatrix} g_1 + \frac{c_1 + c_3}{h} & -\frac{c_3}{h} \\ 0 & g_2 + \frac{c_2 + c_3}{h} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^{k+1} \\ = \begin{bmatrix} 0 & 0 \\ \frac{c_3}{h} & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^* + \begin{bmatrix} g_1 + \frac{c_1 + c_3}{h} & -\frac{c_3}{h} \\ -\frac{c_3}{h} & g_2 + \frac{c_2 + c_3}{h} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^k \\ - \begin{bmatrix} 0 \\ f(v_2^k) \end{bmatrix}. \quad (2.17b)$$

To study the numerical properties of Algorithm 1 it is useful to see how it specializes to linear circuits. If the circuit is linear,  $Y^k = Y$  as in (2.7). If  $Y$  is split into the components  $Y = D - Y_l - Y_u$  then

$$v^{k+1} = M_1(h) v^k - F_1(h) u(t^{k+1}) \quad (2.18)$$

where

$$M_1(h) = [D - Y_u]^{-1} \left[ \frac{C}{h} + Y_l (D - Y_l)^{-1} \left( \frac{C}{h} + Y_u \right) \right] \quad (2.19)$$

and

$$F_1(h) = [D - Y_u]^{-1} [I - (D - Y_l)^{-1}]. \quad (2.20)$$

$M_1(h)$  is referred to as the *companion matrix* of Algorithm 1.

*Remark 2.1:* Algorithm 1 requires the computation of  $Y^k$  at each time-step. This computation in turn requires the computation of the Jacobian of  $f(v, u(t))$ . ■

## 2.2. Algorithm 2

Algorithm 2 uses one iteration of a nonlinear symmetric displacement method to approximate the solution of (2.4). The Newton-Raphson method is used to approximate the solution of the  $n$  scalar nonlinear equations resulting from the use of the nonlinear displacement method. In timing simulators only one step of the Newton-Raphson iteration for each scalar equation in Algorithm 2.

Let

$$\tilde{v}^{l,i} \triangleq \begin{cases} [v_1^l, v_2^l, \dots, v_{i-1}^l, v_i^{l-1/2}, \dots, v_n^{l-1/2}]^T & \text{if } 2l \text{ is odd} \\ [v_1^{l-1/2}, v_2^{l-1/2}, \dots, v_i^{l-1/2}, v_{i+1}^l, \dots, v_n^l]^T & \text{if } 2l \text{ is even.} \end{cases} \quad (2.21)$$

Then the nonlinear symmetric Gauss-Seidel-Newton step is defined by the following.

Forward step:

$$v_i^{k+1/2} = v_i^k - \frac{g_i(\tilde{v}^{k+1/2,i}, v^k, u(t^{k+1}))}{\frac{\partial g_i}{\partial v_i}(\tilde{v}^{k+1/2,i}, v^k, u(t^{k+1}))}, \quad i = 1, 2, \dots, n. \quad (2.22a)$$

Backward step:

$$v_i^{k+1} = v_i^{k+1/2} - \frac{g_i(\tilde{v}^{k+1,i}, v^k, u(t^{k+1}))}{\frac{\partial g_i}{\partial v_i}(\tilde{v}^{k+1,i}, v^k, u(t^{k+1}))}, \quad i = n, n-1, \dots, 1 \quad (2.22b)$$

where the subscript  $i$  denotes the  $i$ th component of a vector.

*Example 2.2.* When Algorithm 2 is applied to the circuit shown in Fig. 2, (2.4) becomes

$$\begin{bmatrix} g_1 v_1^{k+1} \\ f(v_2^{k+1}) \end{bmatrix} + \frac{C}{h} v^{k+1} - \frac{C}{h} v^k = 0 \quad (2.23)$$

and component-wise

$$g_1 v_1^{k+1} + \frac{c_1 + c_3}{h} (v_1^{k+1} - v_1^k) - \frac{c_3}{h} (v_2^{k+1} - v_2^k) = 0 \quad (2.24a)$$

$$f(v_2^{k+1}) - \frac{c_3}{h} (v_1^{k+1} - v_1^k) + \frac{c_2 + c_3}{h} (v_2^{k+1} - v_2^k) = 0. \quad (2.24b)$$

Let

$$g_2^p = \frac{\partial f(v_2^p)}{\partial v_2}, \quad p = k, k+1/2.$$

The forward step is

$$v_1^{k+1/2} = v_1^k - \frac{g_1 v_1^k}{g_1 + \frac{c_1 + c_3}{h}} \quad (2.25a)$$

$$v_2^{k+1/2} = v_2^k - \frac{f(v_2^k) - \frac{c_3}{h}(v_1^{k+1/2} - v_1^k)}{g_2^k + \frac{c_2 + c_3}{h}} \quad (2.25b)$$

and the backward step is

$$v_2^{k+1} = v_2^{k+1/2} - \frac{f(v_2^{k+1/2}) - \frac{c_3}{h}(v_1^{k+1/2} - v_1^k) + \frac{c_2 + c_3}{h}(v_2^{k+1/2} - v_1^k)}{g_2^{k+1/2} + \frac{c_2 + c_3}{h}} \quad (2.25c)$$

$$v_1^{k+1} = v_1^k - \frac{g_1 v_1^{k+1/2} + \frac{c_1 + c_3}{h}(v_1^{k+1/2} - v_1^k)}{g_1 + \frac{c_1 + c_3}{h}} \quad (2.25d)$$

*Remark 2.2:* For linear circuits, Algorithm 2 yields the same equations as Algorithm 1. Hence, the companion matrix of Algorithm 2,  $M_2$ , is identical to  $M_1$ .

*Remark 2.3:* Algorithm 2 requires the computation of  $2n$  diagonal entries of the Jacobian of  $f(v, u(t))$  only. In general, it requires less computation than Algorithm 1.

### 2.3. Algorithm 3

Algorithm 3 uses an intermediate time-step  $t^{k+1/2} = t^k + h/2$ . For the first half-step,  $v^{k+1/2}$  is computed by one iteration of the nonlinear Gauss-Seidel-Newton algorithm. For the second half-step,  $v^{k+1}$  is computed by one iteration of the backward nonlinear Gauss-Seidel-Newton algorithm. The intermediate result,  $v^{k+1/2}$ , is discarded after  $v^{k+1}$  has been computed. With

$$\tilde{g}(v^m, v^l, u(t^m)) \triangleq \frac{2C(v^m)}{h} v^m - \frac{2C(v^m)}{h} v^l - f(v^m, u(t^m)). \quad (2.26)$$

Algorithm 3 may be written as

Forward step:

$$v_i^{k+1/2} = v_i^k - \frac{\tilde{g}_i(\tilde{v}^{k+1/2, i}, v^k, u(t^{k+1/2}))}{\frac{\partial \tilde{g}_i}{\partial v_i}(\tilde{v}^{k+1/2, i}, v^k, u(t^{k+1/2}))}, \quad i = 1, 2, \dots, n. \quad (2.27a)$$

Backward step:

$$v_i^{k+1} = v_i^{k+1/2} - \frac{\tilde{g}_i(\tilde{v}^{k+1, i}, v^{k+1/2}, u(t^{k+1}))}{\frac{\partial \tilde{g}_i}{\partial v_i}(\tilde{v}^{k+1, i}, v^{k+1/2}, u(t^{k+1}))}, \quad i = n, n-1, \dots, 1 \quad (2.27b)$$

where  $\tilde{v}^{l, i}$  is defined by (2.21).

*Example 2.3.* If Algorithm 3 is applied to the circuit shown in Fig. 2 then, from (2.23), (2.24a), and (2.24b)

the forward step is

$$v_1^{k+1/2} = v_1^k - \frac{g_1 v_1^k}{g_1 + 2 \frac{c_1 + c_3}{h}} \quad (2.28a)$$

$$v_2^{k+1/2} = v_2^k - \frac{f(v_2^k) - \frac{2c_3}{h}(v_1^{k+1/2} - v_1^k)}{g_2^k + 2 \frac{c_2 + c_3}{h}} \quad (2.28b)$$

and the backward step is

$$v_2^{k+1} = v_2^{k+1/2} - \frac{f(v_2^{k+1/2})}{g_2^k + 2 \frac{c_2 + c_3}{h}} \quad (2.28c)$$

$$v_1^{k+1} = v_1^{k+1/2} - \frac{g_1 v_1^{k+1/2} - \frac{2c_3}{h}(v_2^{k+1} - v_2^{k+1/2})}{g_1 + 2 \frac{c_1 + c_3}{h}} \quad (2.28d)$$

For linear circuits, Algorithm 3 is defined by

$$[D - Y_l] v^{k+1/2} = \left[ \frac{2C}{h} + Y_u \right] v^k + u(t^{k+1/2}) \quad (2.29a)$$

$$[D - Y_u] v^{k+1} = \left[ \frac{2C}{h} + Y_l \right] v^{k+1/2} + u(t^{k+1}). \quad (2.29b)$$

Hence

$$v^{k+1} = M_3(h) v^k + [D - Y_u]^{-1} [u(t^{k+1}) + [D - Y_l]^{-1} u(t^{k+1/2})] \quad (2.30)$$

where

$$M_3(h) \triangleq [D - Y_u]^{-1} \left[ \frac{2C}{h} + Y_l \right] [D - Y_l]^{-1} \left[ \frac{2C}{h} + Y_u \right] \quad (2.31)$$

is the companion matrix of Algorithm 3.

*Remark 2.3.* Algorithm 3 requires the computation of  $2n$  diagonal entries of the Jacobian of  $f(v, u(t))$ , as does Algorithm 2. It will be shown in Section III that Algorithm 3 has better stability and accuracy properties than Algorithms 1 and 2.

### III. NUMERICAL PROPERTIES OF THE ALGORITHMS

In many cases, the numerical properties of an integration method, such as stability and accuracy, can be studied using test problems [16], [17] which are simple enough to allow a theoretical analysis but still sufficiently general to provide insight into how the integration method will behave in more complex situations. For the analysis of the usual multistep methods, the test problem consists of a linear, time-invariant, zero-input, asymptotically-stable, differential equation.

Unfortunately, this simple test problem cannot be used to evaluate the displacement techniques described in Section II.

In fact, each variable of the system of differential equations is treated differently according to the order in which the equations are processed. Hence, a more complex test problem must be used. The test problem used to analyze the numerical properties of the algorithms presented in Section II are circuits which belong to the following class:

*Definition 3.1:* A circuit is said to be a test circuit of class  $\tau$  if:

- (i) it consists of positive, linear, time-invariant resistors and capacitors and linear, time-invariant voltage-controlled current sources;
- (ii) each node has a capacitor to ground;
- (iii) it is asymptotically stable.

For test circuits of class  $\tau$ , (2.1) may be written

$$\begin{aligned} C\dot{v}(t) &= -\tilde{G}v \\ v(0) &= v^0 \end{aligned} \quad (3.1)$$

where, by (i) and (ii) above,  $C$  is strictly diagonally dominant and  $\tilde{G}$  can be decomposed into  $G + G_m$ , where  $G$  is the nodal conductance matrix of the circuit with no controlled sources, and  $G_m$  is the nodal conductance matrix of the controlled sources alone. Note that  $G$  is diagonally dominant and symmetric. The state equation for a test circuit of class  $\tau$  is

$$\begin{aligned} \dot{v}(t) &= Av \\ v(0) &= v^0 \end{aligned} \quad (3.2)$$

where  $A \triangleq -C^{-1}(G + G_m)$  and, by (iii),  $\sigma(A)$ , the spectrum of  $A$ , is in  $\mathbb{C}_0^-$ , the open left half plane. When Algorithms 1, 2, and 3 are applied to a test circuit of class  $\tau$ , the sequence of the computed solutions is given by

$$v^{k+1} = M_i(h)v^k, \quad i = 1, 2, 3 \quad (3.3)$$

where  $M_i(h)$  is the companion matrix of the method defined in (2.19) and (2.31). Following [16] and [17], the numerical properties of the methods will be investigated using fixed step-size  $h$  yielding

$$v^{k+1} = [M_i(h)]^{k+1}v^0. \quad (3.4)$$

Unfortunately, we cannot prove the numerical properties of Algorithms 1, 2, and 3 on circuits of class  $\tau$ , and we specialize to subclasses  $\tau_j$ . In defining the subclasses  $\tau_j$  we try to identify the largest classes for which the numerical properties of interest help. Since subclasses  $\tau_j \subset \tau$  will be used to test the various methods, the definition of numerical properties are also relative to a class.

*Definition 3.2. (stability):* Algorithm  $i$ ,  $i = 1, 2, 3$  is stable for a test circuit of class  $\tau_j$ , if  $\exists \delta > 0$ ,  $\exists N > 0$  such that  $\forall v^0 \in \mathbb{R}^n$ ,  $\exists \bar{k} > 0$  such that

$$\|v^k\| < N \quad \forall k \geq \bar{k} \quad \forall h \in [0, \delta) \quad (3.5)$$

where  $\{v^k\}$  is the sequence generated by the algorithm applied to the test circuit of class  $\tau_j$ .

*Definition 3.3. ( $\tilde{A}$ -stability):* Algorithm  $i$ ,  $i = 1, 2, 3$  is  $\tilde{A}$ -stable for a test circuit of class  $\tau_j$  if  $\exists N > 0$  such that  $\forall v^0 \in \mathbb{R}^n$ ,  $\exists \bar{k}$  such that

$$\|v^k\| < N \quad \forall k \geq \bar{k} \quad \forall h \in [0, \infty) \quad (3.6)$$

where  $\{v^k\}$  is the sequence generated by the algorithm applied to the test circuit of class  $\tau$ .

*Remark 3.1:* The concept of  $\tilde{A}$ -stability [14], [15] is related to the concept of  $A$ -stability [17]. An algorithm which is  $A$ -stable for a large class  $\tau_j$  would be highly efficient since the stepsize is only limited by accuracy considerations, as in the case of implicit backward differentiation formulas [18]. ■

The definition of stability requires that the sequence of  $\{v^k\}$  be bounded for small values of the stepsize  $h$ ; the definition of  $\tilde{A}$ -stability required that the sequence of  $\{v^k\}$  be bounded for all positive values of  $h$ . The following proposition relates the boundedness of the sequence  $\{v^k\}$  with the spectrum of  $M(h)$ .

*Proposition 3.1:* The sequence of vectors  $\{v^k\}$  defined by (3.4) is bounded for a given value of the stepsize  $\bar{h}$  iff the spectrum of  $M(\bar{h})$  is contained in the unit ball  $\mathcal{B}(0, 1)$ , i.e.,  $\sigma(M(\bar{h})) \subseteq \mathcal{B}(0, 1)$  and no multiple zero of the minimal polynomial of  $M(h)$  has modulus equal to one [19].

The following proposition may be derived immediately from Proposition 3.1:

*Proposition 3.2:* An integration algorithm is stable ( $\tilde{A}$ -stable) iff  $\exists \delta > 0$  such that  $\forall h \in [0, \delta)(\forall h \in [0, \infty))$  the spectrum of  $M(h)$  is contained in the unit ball  $\mathcal{B}(0, 1)$  and no multiple zero of the minimal polynomial of  $M(h)$  has modulus equal to one.

We define now the largest class of circuits for which it is possible to prove the stability of Algorithms 1, 2, and 3.

*Definition 3.4:* A test circuit of class  $\tau$  is said to be of class  $\tau_1$  if all capacitors have one node connected to ground, i.e.,  $C \equiv C_d$  where  $C_d$  is a diagonal matrix.

*Theorem 3.1:* Algorithms 1 and 2 are stable for any test circuit of class  $\tau_1$ .

Note that class  $\tau_1$  is the subset of circuits  $\tau$  with no floating capacitors. The properties of "one-sweep" integration methods for such circuits are extensively described in [14] and [15].

We consider now Algorithm 3 and we prove that it is stable for a class of circuits containing floating capacitors. Let  $C$  be split as

$$C = C_d - C_l - C_u \quad (3.7)$$

where  $C_d$ ,  $C_l$ , and  $C_u$  are diagonal, lower triangular, and upper triangular, respectively. Let

$$H = -\frac{1}{2}[(C_d - C_u)^{-1} + (C_d - C_l)^{-1}]\tilde{G}. \quad (3.8)$$

*Definition 3.5:* A test circuit of class  $\tau$  is said to be of class  $\tau_2$  if  $\sigma(H) \subset \mathbb{C}_0^-$ ; i.e., matrix  $H$  has eigenvalues in the open left half-plane.

*Theorem 3.2:* Algorithm 3 is stable for any circuit of class  $\tau_2$ .

*Remark 3.2:* Since it is easy to see that a circuit of class  $\tau_1$  is also of class  $\tau_2$ , Algorithm 3 is stable for any test circuit of class  $\tau_1$ . Hence, it is "more" stable than Algorithm 1 and 2. ■

We define now the class of test circuits for which Algorithm 3 is  $\tilde{A}$ -stable.

*Definition 3.6:* Let

$$W \triangleq (C_d - C_l)\tilde{G} + \tilde{G}(C_d - C_u) \quad (3.9a)$$

$$V \triangleq (C_d - C_u)\tilde{G} + \tilde{G}(C_d - C_l). \quad (3.9b)$$

A test circuit is said to be of class  $\tau_3$  if:

- (i)  $\tilde{G} = G = G_d$ , where  $G_d$  is a diagonal matrix, i.e., the circuit resistive elements are only two terminal resistors with one terminal connected to ground;
- (ii) either  $W$  and  $V$  are positive definite or  $G$  is positive definite and  $W$  and  $V$  are positive semidefinite.

**Theorem 3.3:** Algorithm 3 is  $\tilde{A}$ -stable for any test circuit of class  $\tau_3$ .

We show now other interesting stability properties of Algorithm 3 on a larger class of circuits.

**Definition 3.7:** A test circuit of class  $\tau$  is said to be class  $\tau_4$  if:  $\tilde{G} = G = G_d$  is a diagonal positive definite matrix; i.e., the circuit resistive elements are only two terminal resistors with one terminal connected to ground and all nodes have one resistor connected to ground.

**Theorem 3.4:** For any test circuit in  $\tau_4$ ,  $\exists \bar{h}$ , such that

$$\|M_3(h)\|_2 < 1 \quad \forall h \in (\bar{h}, \infty) \quad (3.10)$$

and in particular

$$\sigma(M_3(h)) \subset \mathcal{B}(0, 1) \quad \forall h \in (\bar{h}, \infty). \quad (3.11)$$

**Remark 3.3:** This result shows that Algorithm 3 does not introduce instabilities for test circuits of class  $\tau_4$  if the step-size is chosen large enough. Note that in general integration algorithms are stable for values of the stepsize bounded from above. This result is peculiar of Algorithm 3. ■

Note that  $\tau_3 \subseteq \tau_4$ . We can prove that Algorithm 3 is  $\tilde{A}$ -stable for any circuit of class  $\tau_4$  if the circuit equations (2.1) are prescaled as follows:

$$\begin{aligned} \bar{C}\bar{v} &= -\bar{v} \\ \bar{v}(0) &= \bar{v}^0 \end{aligned} \quad (3.12)$$

where  $\bar{C} = G^{-1/2}CG^{-1/2}$ , and  $\bar{v} = G^{1/2}v$ . If Algorithm 3 is applied to (3.12) then the companion matrix becomes

$$\bar{M}_3(h) = (\bar{D} - \bar{Y}_u) \left[ \frac{2\bar{C}}{h} + \bar{Y}_l \right]^{-1} (\bar{D} - \bar{Y}_l) \left[ \frac{2\bar{C}}{h} + \bar{Y}_u \right]^{-1} \quad (3.13)$$

where  $\bar{Y} = 2\bar{C}/h + I$  and  $\bar{Y} = \bar{D} - \bar{Y}_l - \bar{Y}_u$ ;  $\bar{D}$  is diagonal,  $\bar{Y}_l$  and  $\bar{Y}_u$  are strictly lower and upper triangular, respectively.

**Theorem 3.5:** Algorithm 3 is  $\tilde{A}$ -stable for any test circuits of class  $\tau_4$ , provided that the circuit equations are prescaled as in (3.12).

**Remark 3.5:** For circuits in  $\tau_4$ , Algorithm 3 is  $\tilde{A}$ -stable no matter what the values of the floating capacitors in the circuit are. ■

We address now the accuracy of the symmetric integration algorithms.

**Definition 3.8:** Let  $v(t^k)$  be the exact value of the solution of a test circuit of class  $\tau$  at time  $t^k$ . Let  $v^k$  be the computed solution at time  $t^k$ , assuming  $v^{k-1} = v(t^{k-1})$ , i.e., no error has been made in computing the value of  $v$  at the previous time-point. Letting  $h \triangleq t^k - t^{k-1}$ , the local truncation error for a test circuit of class  $\tau$  is

$$\epsilon = \|v(t^k) - v^k\|. \quad (3.14)$$

If  $\epsilon = O(h^{r+1})$ ,  $r$  is said to be the order of the integration algorithm for the test circuit.

For Algorithms 1, 2, and 3 applied to a test circuit of class  $\tau$ , we have

$$M_i(h) = I + h \frac{\partial M_i(0)}{\partial h} + O(h^2); \quad i = 1, 2, 3 \quad (3.15)$$

and

$$v(t^k) = e^{hA} v(t^{k-1}) = [I + hA] v^{k-1} + O(h^2). \quad (3.16)$$

Therefore

$$\epsilon = \left\| [I + hA] v^{k-1} - I + h \frac{\partial M_i(0)}{\partial h} v^{k-1} \right\| + O(h^2) \quad (3.17)$$

$$\leq \left\| A - \frac{\partial M_i(0)}{\partial h} \right\| \|v^{k-1}\| |h| + O(h^2), \quad i = 1, 2, 3. \quad (3.18)$$

If the test circuit is of class  $\tau_1$ , then  $\partial M_i(0)/\partial h = A$  as proven in Theorems 3.1 and 3.2. Hence, Algorithms 1, 2, and 3 are first-order integration methods.

Unfortunately, for test circuits of class  $\tau$ , (3.18) is the closest estimate of the error, i.e., the integration algorithms are not even first-order methods. However, experimental results show that the algorithms behave well on a large set of MOS circuit examples.

When existing timing simulators, such as the timing simulation part of the SPLICE.1A program, are used to analyze MOS circuits, nonphysical oscillatory components may appear in the computed solutions, as shown in [11] and [21] if the step-size is not chosen carefully. These parasitic components are generated by the numerical approximation and, in particular, by the displacement method used.

We will now determine bounds on such oscillatory components for the algorithms introduced in Section II. We start by defining a new class of test circuits.

**Definition 3.9:** A circuit of class  $\tau$  is said to be of class  $\tau_5$  if:

- (i)  $C = C_d$ , where  $C_d$  is a diagonal matrix, i.e., every capacitors in the circuit is grounded;
- (ii) the eigenvalues of  $A = -C^{-1}\tilde{G}$  are real, i.e.,  $\sigma(A) \subset \mathbb{R}$ .

Test circuits in this class have no oscillatory components since the eigenvalues of  $A$  are real. Therefore, any oscillatory component in the computed solution is due to the numerical approximation.

**Proposition 3.3:** Oscillatory parasitic components are present in the computed solution if the spectrum of the companion matrix  $M(h)$  contains complex conjugate eigenvalues.

Since  $\tau_5 \subset \tau_1$  and Algorithms 1, 2, and 3 are first-order integration methods for circuits in  $\tau_1$

$$\max_i |\text{Im}(\xi_i)| = O(h^2); \quad \xi_i \in \sigma(M_i(h)), \quad i = 1, 2, 3. \quad (3.19)$$

This ensures that the oscillatory components of the computed solutions can be made negligible by choosing a sufficiently small step-size. Algorithm 3 has stronger properties. In fact, we can prove that it "behaves well" also for circuits containing floating capacitors.

**Theorem 3.6:** For any circuit belonging to subclass  $\tau_4$

$$\max_i |\text{Im} \xi_i| = O(h^2), \quad \xi_i \in \sigma(M_3(h)). \quad (3.20)$$

Note that  $C$  and  $G$  are positive definite and symmetric for circuits in  $\tau_4$ . Hence, the eigenvalues of  $A = -C^{-1}G$  are all real.

This result shows that numerical oscillatory components introduced by Algorithm 3 are bounded by  $O(h^2)$  also for circuits containing floating capacitors. Moreover, we can prove that Algorithm 3 does not introduce oscillatory components for all step sizes, for test circuits of subclass  $\tau_6$ .

*Definition 3.10:* A test circuit is said to be of class  $\tau_6$  if

- (i)  $[2C/h + Y_I]$  and  $(D - Y_I)^{-1}$  commute;
- (ii) the eigenvalues of  $A = -C^{-1}\tilde{G}$  are real, i.e.,  $\sigma(A) \subset \mathbb{R}$ .

*Theorem 3.7:* For any test circuit belonging to subclass  $\tau_6$

$$\sigma(M_3(h)) \subset \mathbb{R}, \quad \forall h \in [0, \infty). \quad (3.21)$$

#### IV. EXPERIMENTAL RESULTS

The accuracy and stability properties of the algorithms have been established for test circuits of restricted classes. Note that MOS circuits do not belong to any of the  $\tau_j$  classes introduced before. To verify the usefulness of these methods in the analysis of VLSI MOS circuits, it is important to test these properties on actual MOS circuits used in digital design. For this purpose, an experimental timing simulator has been developed to explore these different algorithms [21]. Since our aim is to test accuracy and stability, the simulator does not perform an event-driven analysis of the circuit. In the next section we show how event-driven symmetric algorithms can be implemented.

Different benchmark circuits, which are used as building blocks for VLSI systems, have been successfully simulated. The circuits have floating capacitors and each node has a capacitor to ground. Nonlinear capacitors from MOS transistor terminal nodes to ground are defined by the MOS internal model.

Though the numerical properties proved in Section III are valid for limited classes of circuits, the simulations show that the algorithms yield stable and accurate solutions for usual MOS circuits. The node voltage waveforms never show parasitic oscillatory components. Moreover, the voltage waveforms well agree with those computed with circuit simulator SPICE2.

The circuit in Fig. 3(a) is an enhancement-load NMOS bootstrapped inverter. A floating capacitor, bootstrap, allows the load transistor to be turned hard on when the input voltage is rising. This enhances the speed of the circuit. Fig. 3(b) shows the computed voltage waveform of the output node, in response to a delayed input voltage pulse.

The circuit shown in Fig. 4(a) is a MOS-TTL interface. It consists of three stages of depletion-load MOS inverters, followed by a push-pull enhancement-load MOS driver stage. The output capacitor simulates the driven load. Fig. 4(b) shows the computed voltage waveform of the output node, in response to a delayed input voltage pulse.

The circuit in Fig. 5(a) is a NMOS register. A pass transistor is floating between nodes 3 and 4. The initial condition is:  $V_1 = \text{HIGH}$ ,  $V_3 = \text{LOW}$ ,  $V_4 = \text{HIGH}$ ,  $V_5 = \text{LOW}$ . A delayed voltage pulse is applied to the gate of the pass transistor. At the end of the transient  $V_4 = \text{LOW}$  and  $V_5 = \text{HIGH}$ . Fig. 5(b) shows the voltage waveforms of nodes 4 and 5.

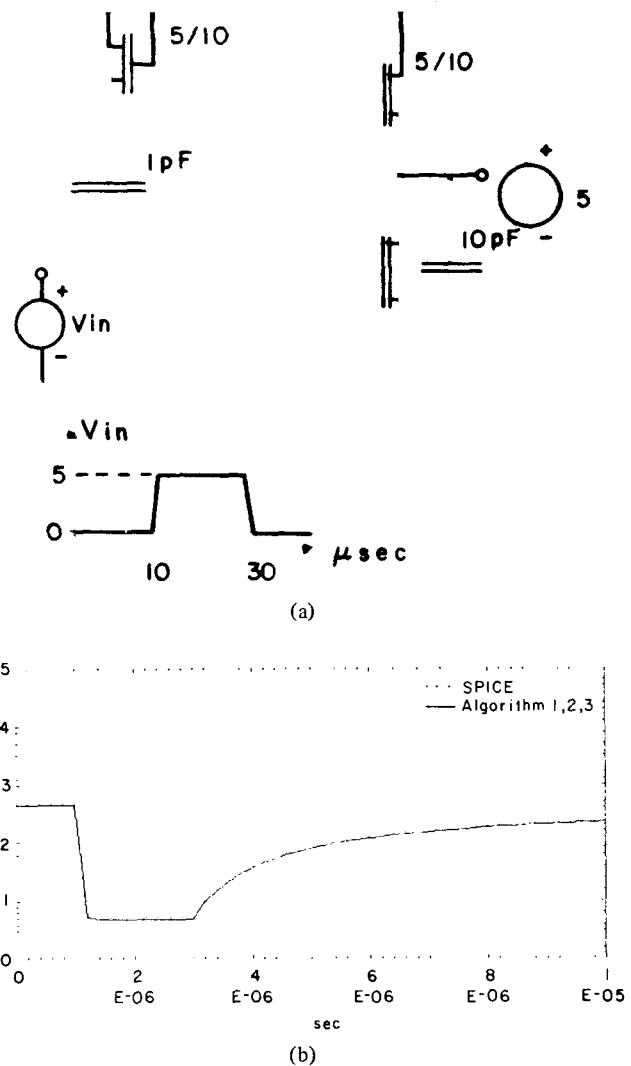


Fig. 3. (a) Enhancement-load NMOS bootstrapped inverter. (b) Enhancement-load NMOS bootstrapped inverter: Computed waveforms.

The circuit shown in Fig. 6(a) is a multiplexer realized with a tree of NMOS pass transistors. The initial condition is  $V_4 = \text{LOW}$ ,  $V_5 = \text{LOW}$ , and  $V_6 = \text{HIGH}$ . Node 1 is kept HIGH. A delayed voltage pulse is applied to the address lines  $A_0$ ,  $A_1$ , and  $A_2$ . At the end of the transient  $V_5 = \text{HIGH}$  and  $V_6 = \text{LOW}$ . Fig. 6(b) shows the voltage waveforms of nodes 5 and 6.

#### V. IMPLEMENTATION OF EVENT-DRIVEN SYMMETRIC ALGORITHMS

In this section we discuss the significance of the symmetric algorithms to the design of new timing simulators. An effective implementation of the symmetric integration algorithms in a timing simulator requires an ordering of the equations that allows one to exploit the advantages of the relaxation-based integration algorithms. A large part of the computing time saving achieved by timing simulators is due to *event-driven* analysis, which allows the simulation of only the active parts of a circuit at each time-point [22], [8]. The effect of event-driven selective trace analysis is to dynamically sort the circuit nodes for processing, according to the flow of the signal in the network. Selective trace algorithms have been successfully



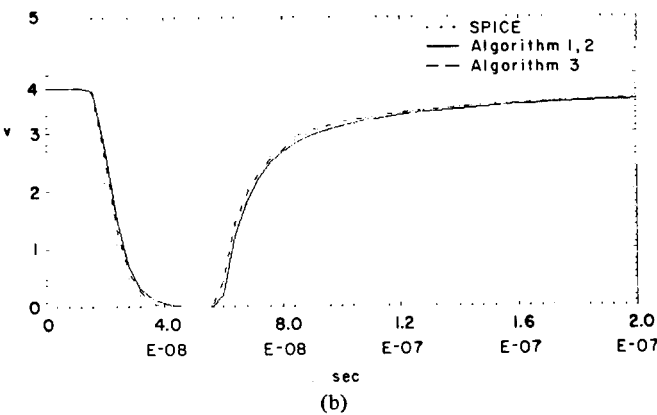
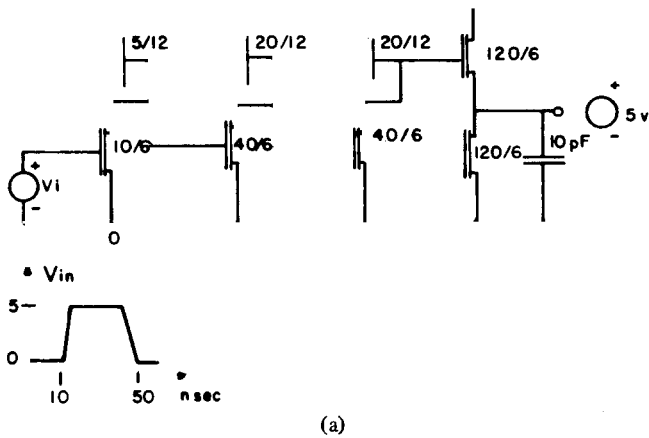


Fig. 4. (a) nMOS-TTL interface. (b) nMOS-TTL interface: Computed waveforms.

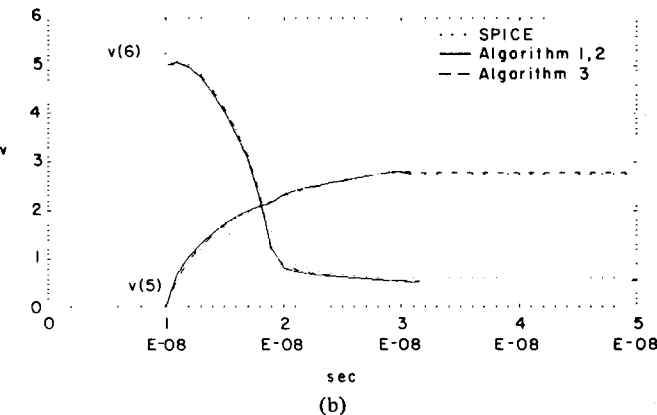
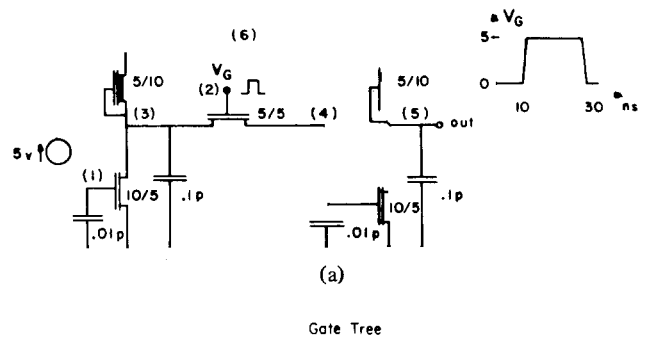


Fig. 5. (a) nMOS shift-register. (b) nMOS shift-register: Computed waveforms.

TABLE I  
NUMERICAL PROPERTIES OF THE SYMMETRIC DISPLACEMENT ALGORITHMS FOR DIFFERENT CLASSES OF CIRCUITS

	Stability	$\tilde{A}$ -stability	1 <sup>st</sup> order accuracy	Numerical oscillations bounded by $O(h^2)$	No numerical oscillations
Algorithm 1	$\tau_1$		$\tau_1$	$\tau_5$	
Algorithm 2	$\tau_1$		$\tau_1$	$\tau_5$	
Algorithm 3	$\tau_2$	$\tau_3$	$\tau_1$	$\tau_4 \cup \tau_5$	$\tau_6$

implemented in connection with the nonlinear Gauss-Seidel algorithm in the SPLICE1 program [8].

Symmetric displacement algorithms can be implemented in connection with event-driven analysis by coupling the selective trace algorithm with a special bilateral integration algorithm. In particular, subcircuits containing strongly bilateral elements can be solved using Algorithm 3, while unilateral elements can be solved by the usual Gauss-Seidel integration algorithm. The bilateral integration algorithm detects the subnetworks of bilateral elements and pushes the corresponding nongrounded nodes on a stack.

In the setup phase of the simulation every node  $i$  of the circuit is associated to a list  $L_i$  of the nodes connected to  $i$  by bilateral elements, and to a flag  $F_i$ , which is set to true if node  $i$  is connected to bilateral elements.

The algorithm is described in Pidgin C. Let  $i$  be the scheduled node at any time-point and  $i^*$  the node scheduled before  $i$ . Let  $h$  be the stepsize.

*Bilateral Integration Algorithm*

```

if ( $F_i$  is TRUE) {
  if ( $\bar{F}_{i^*}$  is TRUE) {
     $h = h/2$ 
    for (each node  $n$  in  $L_i$ ) {
      if ( $n$  is not in the stack) {
        schedule  $n$  immediately after  $i$ ;
        solve the forward half-step of Algorithm 3 (2.27a) for node  $i$ ;
        push  $i$  onto the stack;
      }
    }
  }
  else {
    while (stack not empty) {
      pop a node,  $m$ , from the stack;
      solve backward half-step of Algorithm 3 (2.27b) for node  $m$ ;
    }
    if ( $\bar{F}_{i^*}$  is TRUE)  $h = 2h$ ;
    solve nonlinear Gauss-Seidel step for node  $i$ ;
  }
}
    
```

VI. CONCLUSIONS

In this paper we addressed the problem of determining the numerical stability and accuracy of integration algorithms for the timing analysis of large scale circuits. In particular, we have proposed three symmetric displacement algorithms and

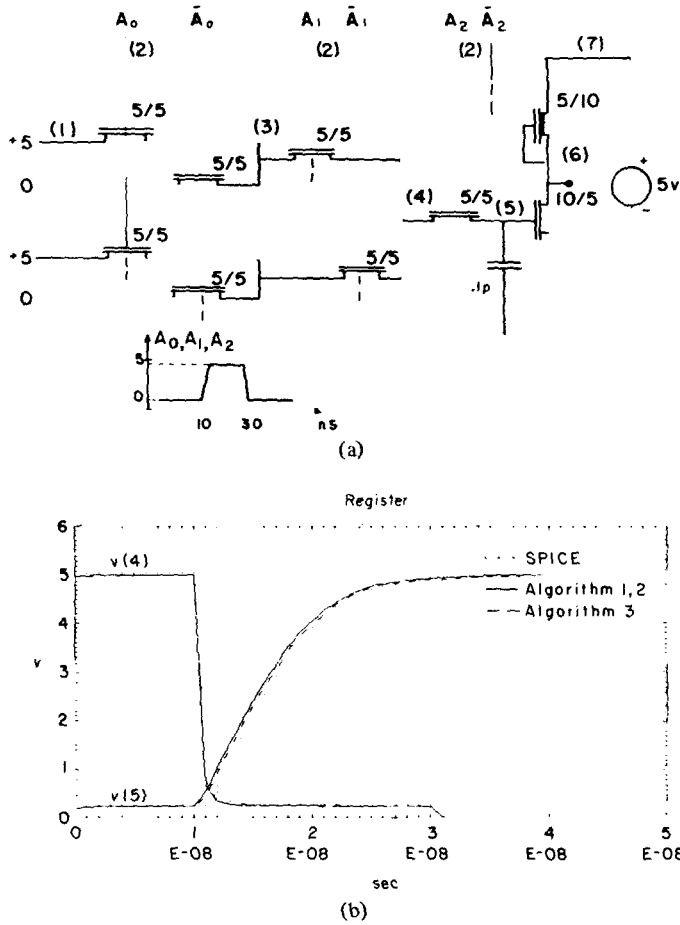


Fig. 6. (a) nMOS multiplexer. (b) nMOS multiplexer: Computed waveforms.

we have investigated the related numerical properties of each algorithm. Table I summarizes our results. Based on the results obtained in [14], [15] and the results presented in Sections III and IV, we conclude that symmetric displacement algorithms can be effectively used for the timing simulation of MOS circuits containing floating capacitors. Note that the numerical behavior of all these methods is intrinsically limited because only one step of displacement is computed at each time-point. Recent results [23]-[25] have shown that better numerical properties can be obtained by iterating the displacement step to convergence at the expense of an increased time and memory requirement.

## VII. ACKNOWLEDGMENTS

The authors wish to thank Dr. R. Brayton and Dr. E. Lelarasmee for many helpful discussions.

## APPENDIX

### Proof of Theorem 3.1:

Let  $M(h) = M_i(h)$ ,  $i = 1, 2$ . Let  $\tilde{G}$  be split according to

$$\tilde{G} = \tilde{G}_d - \tilde{G}_l - \tilde{G}_u \quad (\text{A1})$$

where  $\tilde{G}_d$ ,  $\tilde{G}_l$ , and  $\tilde{G}_u$  are diagonal, lower triangular, and upper triangular, respectively. From (2.19) and (A1),  $M(h)$  can be

written in the following form:

$$M(h) = [C + h(\tilde{G}_d - \tilde{G}_u)]^{-1} \cdot [C + h\tilde{G}_l[C + h(\tilde{G}_d - \tilde{G}_l)]^{-1}[C + h\tilde{G}_u]]. \quad (\text{A2})$$

For  $h = 0$ ,  $M(0) = I$ . Taking the derivative with respect to stepsize  $h$

$$\begin{aligned} \frac{dM(h)}{dh} = & -[C + h(\tilde{G}_d - \tilde{G}_u)]^{-1}(\tilde{G}_d - \tilde{G}_u)M(h) \\ & + [C + h(\tilde{G}_d - \tilde{G}_u)]^{-1} \\ & \cdot \{\tilde{G}_l[C + h(\tilde{G}_d - \tilde{G}_l)]^{-1}[C + h\tilde{G}_u] \\ & - h\tilde{G}_l[C + h(\tilde{G}_d - \tilde{G}_l)]^{-1}(\tilde{G}_d - \tilde{G}_l) \\ & \cdot [C + h(\tilde{G}_d - \tilde{G}_l)]^{-1}[C + h\tilde{G}_u] \\ & + h\tilde{G}_l[C + h(\tilde{G}_d - \tilde{G}_l)]^{-1}\tilde{G}_u\} \end{aligned} \quad (\text{A3})$$

$$\frac{dM(0)}{dh} = -C^{-1}(\tilde{G}_d - \tilde{G}_u) + C^{-1}\tilde{G}_l = -C^{-1}\tilde{G} = A \quad (\text{A4})$$

where  $dM(0)/dh$  is the derivative of  $M(h)$  evaluated at  $h = 0$ . Hence,  $M(h)$  can be expressed as a power series of  $h$

$$M(h) = I + hA + O(h^2). \quad (\text{A5})$$

By the spectral mapping theorem [19]

$$\begin{aligned} \sigma(M(h)) = \{ \xi_i | \xi_i = 1 + h\lambda_i + O(h^2); \\ \lambda_i \in \sigma(A); i = 1, 2, \dots, \sigma \}. \end{aligned} \quad (\text{A6})$$

From (A6)

$$|\xi_i| = |1 + h\lambda_i + O(h^2)|, \quad i = 1, 2, \dots, \sigma$$

and

$$|\xi_i|^2 = [1 + h \operatorname{Re}(\lambda_i)]^2 + [h \operatorname{Im}(\lambda_i)]^2 + O(h^2). \quad (\text{A7})$$

Since  $M(0) = I$ , its eigenvalues are all 1, and 1 is a simple zero of the minimal polynomial of the identity matrix. Therefore, from Proposition 3.2 it is sufficient to show that

$$\sigma(M(h)) \subset \mathcal{B}(0, 1), \quad \forall h \in (0, \delta) \quad (\text{A8})$$

or

$$|\xi_i|^2 < 1, \quad \forall h \in (0, \delta), \quad i = 1, 2, \dots, \sigma. \quad (\text{A9})$$

From (A7)

$$\begin{aligned} 2 \operatorname{Re}(\lambda_i) + h(\operatorname{Re}^2(\lambda_i) + \operatorname{Im}^2(\lambda_i)) + O(h) < 0, \\ i = 1, 2, \dots, \sigma \end{aligned} \quad (\text{A10})$$

$$2 \operatorname{Re}(\lambda_i) + O(h) < 0, \quad i = 1, 2, \dots, \sigma. \quad (\text{A11})$$

Since by assumption  $\operatorname{Re}(\lambda_i) < 0$ ,  $i = 1, 2, \dots, \sigma$ ,  $\exists \delta > 0$ , such that  $\forall h \in (0, \delta)$

$$\sigma(M(h)) \subset \mathcal{B}(0, 1) \quad (\text{A12})$$

and from Proposition 3.2 the Algorithms are stable. ■

### Proof of Theorem 3.2:

From (2.31), (3.7), and (A1) the companion matrix  $M_3(h)$  for a circuit of class  $\tau$  can be expressed as

$$M_3(h) = \begin{bmatrix} (C_d - C_u) + \frac{h}{2}(\tilde{G}_d - \tilde{G}_u) \\ (C_d - C_l) + \frac{h}{2}(\tilde{G}_d - \tilde{G}_l) \end{bmatrix}^{-1} \begin{bmatrix} (C_d - C_u) + \frac{h}{2}\tilde{G}_l \\ (C_d - C_l) + \frac{h}{2}\tilde{G}_u \end{bmatrix} \quad (A13)$$

and  $M_3(0) = I$ . Since  $M_3(0) = I$ , its eigenvalues are all 1, and 1 is a simple zero of the minimal polynomial of the identity matrix. Therefore, from Proposition 3.2 it is sufficient to show that

$$\sigma(M(h)) \subset \mathfrak{B}(0, 1), \quad \forall h \in (0, \delta).$$

Now, let

$$\begin{aligned} P &= C_d - C_u \\ Q &= \tilde{G}_d - \tilde{G}_u \\ R &= \tilde{G}_l \end{aligned}$$

and taking the derivative of  $M_3(h)$  with respect to the step-size  $h$

$$\begin{aligned} \frac{\partial M_3(h)}{\partial h} &= \frac{1}{2} \left\{ - \left[ P + \frac{h}{2} Q \right]^{-1} Q \left[ P + \frac{h}{2} Q \right]^{-1} \left[ P + \frac{h}{2} R \right] \right. \\ &\quad + \left. \left[ P + \frac{h}{2} Q \right]^{-1} R \right\} \left\{ \left[ P^T + \frac{h}{2} Q^T \right]^{-1} \left[ P^T + \frac{h}{2} R^T \right] \right\} \\ &\quad + \frac{1}{2} \left\{ \left[ P + \frac{h}{2} Q \right]^{-1} \left[ P + \frac{h}{2} R \right] \right\} \\ &\quad \cdot \left\{ - \left[ P^T + \frac{h}{2} Q^T \right]^{-1} Q^T \left[ P^T + \frac{h}{2} Q^T \right]^{-1} \right. \\ &\quad \cdot \left. \left[ P^T + \frac{h}{2} R^T \right] + \left[ P^T + \frac{h}{2} Q^T \right]^{-1} R^T \right\} \quad (A14) \end{aligned}$$

and

$$\frac{\partial M_3(0)}{\partial h} = \frac{1}{2} [-P^{-1}(Q - R)] + \frac{1}{2} [-P^T^{-1}(Q^T - R^T)] \quad (A15)$$

$$= -\frac{1}{2} [(C_d - C_u)^{-1} + (C_d - C_l)^{-1}](G + G_m) \quad (A16)$$

$$\triangleq H. \quad (A17)$$

Hence

$$M_3(h) = I + hH + O(h^2) \quad (A18)$$

and

$$\sigma(M_3(h)) = \{ \xi_i | \xi_i = 1 + h\lambda_i + O(h^2)\lambda_i \in \sigma(H) \}. \quad (A19)$$

$$\begin{aligned} &\left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2^2 \\ &= \left\| (C_d - C_u) \left( C_d + \frac{h}{2} \tilde{G} - C_u \right)^{-1} \right\|_2^2 \\ &= \max_{x \neq 0} \left\langle (C_d - C_u) \left[ C_d - C_u + \frac{h}{2} \tilde{G} \right]^{-1} x, (C_d - C_u) \left[ C_d - C_u + \frac{h}{2} \tilde{G} \right]^{-1} x \right\rangle \end{aligned} \quad (A29)$$

Hence, following an argument similar to the one used in Theorem 1, it is easy to show that  $\exists \delta$  s.t.

$$\sigma(M_3(h)) \subset \mathfrak{B}(0, 1) \quad \forall h \in (\sigma, \delta). \quad (A20)$$

This completes the proof.  $\blacksquare$

*Proof of Theorem 3.3:*

For any circuit of class  $\tau$ , we showed in Theorem 3.2 that  $M_3(0) = I$ . Therefore, its eigenvalues are all 1, and 1 is a simple zero of the minimal polynomial of the identity matrix. Therefore, it is sufficient to show that  $\|M_3(h)\|_2 < 1 \quad \forall h > 0$ , which implies that all the eigenvalues of the companion matrix lie inside the unit disk. To prove this we need the following lemmas:

*Lemma 3.1:* If

$$\left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2 < 1 \quad \forall h > 0 \quad (A21a)$$

$$\left\| \left[ \frac{2C}{h} + Y_u \right] (D - Y_l)^{-1} \right\|_2 < 1 \quad \forall h > 0. \quad (A21b)$$

Then

$$\|M_3(h)\|_2 < 1 \quad \forall h > 0. \quad (A22)$$

*Proof:*

$$\|M_3\|_2 = \|M_3^T\|_2 \quad (A23)$$

$$= \left\| \left\{ (D - Y_u)^{-1} \left[ \frac{2C}{h} + Y_l \right] (D - Y_l)^{-1} \left[ \frac{2C}{h} + Y_u \right] \right\}^T \right\|_2 \quad (A24)$$

$$= \left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \left[ \frac{2C}{h} + Y_u \right] (D - Y_l)^{-1} \right\|_2 \quad (A25)$$

$$\leq \left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2 \left\| \left[ \frac{2C}{h} + Y_u \right] (D - Y_l)^{-1} \right\|_2 \quad (A26)$$

$$< 1. \quad (A27)$$

*Lemma 3.2:* If  $G$  is a diagonal matrix and either  $W$  is positive definite or  $G$  is positive definite and  $W$  positive semidefinite, then

$$\left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2 < 1 \quad \forall h > 0. \quad (A28)$$

*Proof:*

$$\begin{aligned} &\left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2^2 \\ &= \left\| (C_d - C_u) \left( C_d + \frac{h}{2} \tilde{G} - C_u \right)^{-1} \right\|_2^2 \\ &= \max_{x \neq 0} \left\langle (C_d - C_u) \left[ C_d - C_u + \frac{h}{2} \tilde{G} \right]^{-1} x, (C_d - C_u) \left[ C_d - C_u + \frac{h}{2} \tilde{G} \right]^{-1} x \right\rangle \end{aligned} \quad (A29)$$

$$\langle x, x \rangle \quad (A30)$$

Let

$$y \triangleq \left[ C_d - C_u + \frac{h}{2} \tilde{G} \right]^{-1} x. \quad (\text{A31})$$

Then

$$\begin{aligned} & \left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2^2 \\ &= \max_{y \neq 0} \frac{\langle (C_d - C_u)y, (C_d - C_u)y \rangle}{\left\langle \left[ C_d - C_u + \frac{h}{2} \tilde{G} \right] y, \left[ C_d - C_u + \frac{h}{2} \tilde{G} \right] y \right\rangle} \end{aligned} \quad (\text{A32})$$

$$= \max_{y \neq 0} \frac{\langle y, (C_d - C_l)(C_d - C_u)y \rangle}{\langle y, (C_d - C_l)(C_d - C_u)y \rangle + \frac{h^2}{4} \langle y, \tilde{G}^2 y \rangle + \frac{h}{2} \{ \langle \tilde{G}y, (C_d - C_u)y \rangle + \langle (C_d - C_u)y, \tilde{G}y \rangle \}} \quad (\text{A33})$$

$$= \max_{y \neq 0} \frac{\langle y, (C_d - C_l)(C_d - C_u)y \rangle}{\langle y, (C_d - C_l)(C_d - C_u)y \rangle + \frac{h^2}{4} \langle y, \tilde{G}^2 y \rangle + \frac{h}{2} \langle y, Wy \rangle} \quad (\text{A34})$$

Since  $(C_d - C_l)(C_d - C_u) = (C_d - C_l)(C_d - C_l)^T$  is a positive definite matrix, then

$$\left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2 < 1 \quad \forall h > 0. \quad (\text{A35})$$

*Lemma 3.3:* If  $G$  is a diagonal matrix and either  $V$  is positive definite or  $G$  is positive definite and  $V$  positive semidefinite, then

$$\left\| \left[ \frac{2C}{h} + Y_u \right] (D - Y_l)^{-1} \right\|_2 < 1 \quad \forall h > 0. \quad (\text{A36})$$

*Proof:* The proof follows the same argument of Lemma 3.2. ■

*Proof of Theorem 3.3—Continuation:*

For any circuit in class  $\tau_3$ , from Lemmas 3.1, 3.2, and 3.3 follows that:

$$\|M_3(h)\|_2 < 1 \quad \forall h > 0 \quad (\text{A37})$$

and therefore Algorithm 3 is  $\tilde{A}$ -stable. ■

*Proof of Theorem 3.4:*

Let

$$a(y) = \langle y, (C_d - C_l)(C_d - C_u)y \rangle. \quad (\text{A38})$$

Then

$$a(y) > 0 \quad \forall y \neq 0 \quad (\text{A39})$$

and from Lemma 3.2

$$\begin{aligned} & \left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2^2 \\ &= \max_{y \neq 0} \frac{a(y)}{a(y) + \frac{h}{2} \left\{ \frac{h}{2} \langle y, G^2 y \rangle + \langle y, Wy \rangle \right\}} < 1 \end{aligned} \quad (\text{A40})$$

$$\forall h > \bar{h}_1 = -2 \frac{\langle y, Wy \rangle}{\langle y, G^2 y \rangle} \quad (\text{A41})$$

It can be shown similarly that

$$\left\| \left[ \frac{2C}{h} + Y_u \right] (D - Y_l)^{-1} \right\|_2^2 < 1 \quad \forall h > \bar{h}_2 = -2 \frac{\langle y, Vy \rangle}{\langle y, G^2 y \rangle} \quad (\text{A42})$$

and  $\forall h > \bar{h} = \max(h_1, h_2)$  both the following inequalities are satisfied:

$$\left\| \left[ \frac{2C}{h} + Y_l \right] (D - Y_u)^{-1} \right\|_2 < 1 \quad (\text{A43a})$$

$$\left\| \left[ \frac{2C}{h} + Y_u \right] (D - Y_l)^{-1} \right\|_2 < 1. \quad (\text{A43b})$$

Hence, it follows from Lemma 3.1 that

$$\|M_3(h)\|_2 < 1 \quad \forall h \in (\bar{h}, \infty) \quad (\text{A44})$$

and in particular

$$\sigma(M_3(h)) \subset \mathfrak{B}(0, 1) \quad \forall h \in (\bar{h}, \infty). \quad (\text{A45})$$

*Proof of Theorem 3.5:*

Let

$$\bar{C} = \bar{C}_d - \bar{C}_l - \bar{C}_u. \quad (\text{A46})$$

Then

$$W = V = 2\bar{C}_d - \bar{C}_u - \bar{C}_l. \quad (\text{A47})$$

Matrices  $W$  and  $V$  are symmetric and diagonally dominant. Hence, they are positive definite and by Lemma 3.1, 3.2, and 3.3 follows:

$$\|M_3(h)\|_2 < 1 \quad \forall h > 0 \quad (\text{A48})$$

and in particular

$$\sigma(M_3(h)) \subset \mathfrak{B}(0, 1) \quad \forall h \in (0, \infty). \quad (\text{A49})$$

Moreover,  $M_3(0) = I$ , its eigenvalues are all 1, and 1 is a simple zero of the minimal polynomial of the identity matrix. This completes the proof. ■

*Proof of Theorem 3.6:*

From (A18)

$$M_3(h) = I + hH + O(h^2) \quad (\text{A50})$$

where

$$H = [(C_d - C_u)^{-1} + (C_d - C_l)^{-1}]G. \quad (A51)$$

Matrix  $H$  is similar to

$$\tilde{H} = G^{1/2}M_3G^{-1/2} \quad (A52)$$

$$= G^{1/2}[(C_d - C_u)^{-1} + (C_d - C_l)^{-1}]G^{-1/2}. \quad (A53)$$

Matrix  $\tilde{H}$  is symmetric and therefore has real eigenvalues. By similarity  $H$  also has real eigenvalues. Hence

$$\sigma(M_3(h)) = \{\xi | \xi = 1 + h\lambda + O(h^2), \quad \lambda \in \sigma(H)\} \quad (A54)$$

and

$$\max_l |\operatorname{Im} \xi_l| = O(h^2). \quad (A55)$$

*Proof of Theorem 3.7:*

$$M_3(h) = (D - Y_u)^{-1} \left[ \frac{2C}{h} + Y_l \right] (D - Y_l)^{-1} \left[ \frac{2C}{h} + Y_u \right] \quad (A56)$$

$$= (D - Y_u)^{-1} (D - Y_l)^{-1} \left[ \frac{2C}{h} + Y_l \right] \left[ \frac{2C}{h} + Y_u \right] \quad (A57)$$

$$\cdot [(D - Y_u)^T (D - Y_u)]^{-1} \left[ \frac{2C}{h} + Y_l \right] \left[ \frac{2C}{h} + Y_l \right]^T. \quad (A58)$$

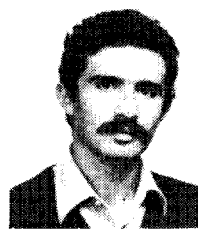
Hence, matrix  $M_3(h)$  is the product of two real symmetric matrices and matrix  $[(D - Y_u)^T (D - Y_u)]$  is positive definite. With an argument similar to the one used in Theorem 3.6 it is easy to show that  $M_3(h)$  has real eigenvalues for all values of the stepsize. ■

## REFERENCES

- [1] W. Nagel, "SPICE2, A computer program to simulate semiconductor circuits," University of California, Berkeley, Memo No. ERL-M520, May 1975.
- [2] "Advanced statistical analysis program (ASTAP)" Program reference manual, Pub. No. SH20-1118-0, IBM Corp. Data Proc. Div., White Plains, NY.
- [3] A. R. Newton and D. O. Pederson, "Analysis time, accuracy and memory requirement tradeoffs in SPICE2," in *Proc. Asilomar Conf.*, 1978.
- [4] A. Vladimirescu and D. O. Pederson, "Performance limitations of the CLASSIE circuit simulation program," in *Proc. IEEE ISCAS* (Rome, Italy), May 1982.
- [5] G. D. Hachtel and A. Sangiovanni Vincentelli, "A survey of third generation simulation techniques," in *Proc. IEEE*, vol. 65, Oct. 1981.
- [6] B. R. Chawla, H. K. Gummel, and P. Kozak, "MOTIS—An MOS timing simulator," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 901-909, Dec. 1975.
- [7] S. P. Fan, M. Y. Hsueh, A. R. Newton, and D. O. Pederson, "MOTIS-C: A new circuit simulator for MOS LSI circuits," in *Proc. IEEE ISCAS*, Apr. 1977.
- [8] A. R. Newton, "The simulation of large scale integrated circuits," Memo UCB/ERL M78/52, July 1978.
- [9] M. Ortega and W. C. Reinboldt, *Iterative Solutions of Nonlinear Equations in Several Variables*. New York: Academic, 1970.
- [10] A. R. Newton, "Timing, logic and mixed-mode simulation for large MOS integrated circuits," in NATO Advanced Study Institute on Computer-Aided Design for VLSI circuits," Sijhoff and Noordhoff, pp. 175-239, 1981.

- [11] M. Y. Hsueh, A. R. Newton, and D. O. Pederson, "New approaches to modeling and electrical simulation of LSI logic circuits," *J. d'Electronique*, pp. 403-413, 1977.
- [12] A. R. Newton, "The timing analysis of floating capacitors for timing simulations," in *Proc. 13th IEEE Asilomar Conf. Circuits Systems and Computers* (Asilomar, CA), Nov. 1979.
- [13] W. Kahan, private notes, 1975.
- [14] G. De Micheli and A. Sangiovanni-Vincentelli, "Numerical properties of algorithms for the timing analysis of MOS VLSI circuits," presented at ECCTD, The Hague, Holland, Aug. 1981.
- [15] G. De Micheli and A. L. Sangiovanni-Vincentelli, "The characterization of integration algorithms for the timing analysis of VLSI circuits," *Int. J. Circ. Theor. Appl.*, vol. 10, pp. 299-309, Oct. 1982.
- [16] C. W. Gear, *Numerical Initial Value Problems for Ordinary Differential Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
- [17] G. Dahlquist and A. Byorch, *Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
- [18] R. K. Brayton, F. G. Gustavson, and G. D. Hachtel, "A new efficient algorithm for solving differential algebraic systems using backward differentiation formulas," *Proc. IEEE*, vol. 60, pp. 98-108, June 1972.
- [19] L. Zadeh and C. A. Desoer, *Linear System Theory: The State Space Approach*. New York: McGraw Hill, 1963.
- [20] G. De Micheli, A. R. Newton, and A. L. Sangiovanni-Vincentelli, "New algorithms for timing analysis of large circuits," in *Proc. IEEE ISCAS* (Houston), Apr. 1980.
- [21] G. De Micheli, "New algorithm for timing analysis of MOS circuits," Master Report, University of California, Berkeley, 1980.
- [22] S. A. Szygenda and E. W. Thompson, "Modeling and digital simulation for design verification and diagnosis," *IEEE Trans. Comput.*, vol. C-25, pp. 1242-1253, Dec. 1976.
- [23] E. Lelarsmee, A. Ruheli, and A. L. Sangiovanni-Vincentelli, "The waveform relaxation method for the time-domain analysis of large scale integrated circuits," *IEEE Trans. CAD*, vol. CAD-1, no. 3, pp. 131-145, Aug. 1982.
- [24] E. Lelarsmee, "The waveform relaxation method for the time-domain analysis of large scale integrated circuits: theory and applications," Ph.D. dissertation, Univ. California, Berkeley, and Memo UCB/ERL M82/40.
- [25] J. E. Kleckner, R. A. Saleh, and A. R. Newton, "Electrical consistency in schematic simulation," in *Proc. Int. Circ. Conf.* (New York), pp. 30-33, Sept. 1982.

\*



Giovanni De Micheli (S'79) was born in Milano, Italy, in 1955. He received the Dr. Eng. degree (summa cum Laude) in nuclear engineering from the Politecnico di Milano, Italy, in 1979 and the M.S. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1980, where he is currently a doctoral candidate.

He spent the fall of 1981 as Consultant in Residence at Harris Semiconductor, Melbourne, Florida. His research interests include several

aspects of the computer aided design of integrated circuits, with particular emphasis on simulation of large scale circuits and the automated synthesis of functional units for VLSI digital circuits.

Mr. De Micheli was granted a Fulbright Scholarship in 1980, a Rotary International Fellowship in 1981, and a IBM Fellowship for VLSI in 1982 and 1983. At the 1983 IEEE-ACM Design Automation Conference he was given a Best Paper Award.

\*



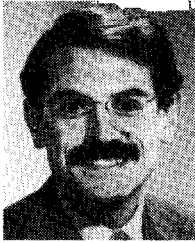
Arthur Richard Newton (S'73-M'78) was born in Melbourne, Australia, on July 1, 1951. He received the B.Eng. and M.Eng.Sci. degrees from the University of Melbourne, Melbourne, Australia, in 1973 and 1975, respectively, and the Ph.D. degree from the University of California, Berkeley, in 1978.

He is currently an Associate Professor in the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, and a Consultant to Silicon Systems Inc.,

Tektronix, and Intel for computer-aided design of integrated circuits. His research interests include all aspects of the computer-aided design of integrated circuits with emphasis on simulation, automated layout techniques, and design methods for VLSI integrated circuits.

Dr. Newton is a member of Sigma Xi.

\*



Alberto Sangiovanni-Vincentelli (M'74-SM'81-F'83) received the Dr. Eng. degree (summa cum laude) from the Politecnico di Milano, Italy, in 1971. From 1971 to 1977 he was with the Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano, Italy, where he held the position of Research Associate, Assistant, and Associate Professor. In 1976 he joined the Department of Electrical Engineering and Computer Sciences where he is presently Professor and Vice Chairman.

He is a consultant in the area of computer aided design to several industries. His research interests are in various aspects of computer aided design of integrated circuits, with particular emphasis on VLSI simulation and optimization.

Dr. Sangiovanni-Vincentelli was Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and is Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and a member of the Large-Scale Systems Committee of the IEEE Circuits and Systems Society. He was the Guest Editor of a special issue of the TRANSACTIONS ON CIRCUITS AND SYSTEMS ON CAD FOR VLSI. He has been elected Executive Vice-President of the IEEE Circuits and Systems Society in 1983.

In 1981 he received the Distinguished Teaching Award of the University of California. At the 1982 IEEE-ACM Design Automation Conference he was given a Best Paper and a Best Presentation Award. In 1983 he received the Guillemin-Cauer Award for the best paper published in the IEEE TRANSACTIONS ON CAS and CAD in 1981-1982.

Dr. Sangiovanni-Vincentelli is a member of ACM and Eta Kappa Nu.

## A Study of Variance Reduction Techniques for Estimating Circuit Yields

DALE E. HOCEVAR, MEMBER, IEEE, MICHAEL R. LIGHTNER, MEMBER, IEEE, AND  
TIMOTHY N. TRICK, FELLOW, IEEE

**Abstract**—The efficiency of several variance reduction techniques (in particular, importance sampling, stratified sampling, and control variates) are studied with respect to their application in estimating circuit yields. This study suggests that one essentially has to have a good approximation of the region of acceptability in order to achieve significant variance reduction. Further, all the methods considered are based, either explicitly or implicitly, on the use of a model. The control variate method appears to be more practical for implementation in a general purpose statistical circuit analysis program. Stratified sampling is the most simple to implement, but yields only very modest reductions in the variance of the yield estimator. Lastly, importance sampling is very useful when there are few parameters and the yield is very high or very low; however, a good practical technique for its implementation, in general, has not been found.

### I. INTRODUCTION

STATISTICAL CIRCUIT DESIGN is a broad topic, and over the years researchers have formulated and classified many topics within the general area. A main concern is the

Manuscript received July 12, 1982, revised January 26, 1983. This work was supported by the National Science Foundation under Grant ENG 78-17815.

D. E. Hocevar was with the Coordinated Science Laboratory, and Department of Electrical Engineering, University of Illinois, Urbana, IL. He is now employed by Arco Oil & Gas Co., Dallas, TX.

M. R. Lightner is with the Coordinated Science Laboratory, and Department of Electrical Engineering, University of Illinois, Urbana, currently on leave at the University of Colorado, Boulder, CO.

T. N. Trick is with the Coordinated Science Laboratory, and Department of Electrical Engineering, University of Illinois, Urbana, IL 61801.

manufacturing yield of a circuit. Intuitively, circuit yield is the percentage of acceptable circuits achieved in production. A circuit designer is usually interested in estimating the yield in advance for use as a circuit evaluation measure. This is the yield analysis problem. Secondly, the designer may want to improve this yield estimate via parametric design. This is the yield maximization problem or yield optimization problem. Yield analysis is the subject of this paper.

In order to estimate yield for analysis and design, one first defines a set of performance or response functions, along with constraints for those functions, for each circuit as

$$g_i(x) \leq B_i, \quad i = 1, \dots, m. \quad (1)$$

These functions are usually only known implicitly via simulations, and thus can be very costly to evaluate. Assuming that the components or circuit parameters can be statistically described by a probability density function (*pdf*),  $p(x)$ ,  $x \in R^n$ , one can express circuit yield as

$$Y = \int_{R_a} p(x) dx \quad (2)$$

where  $R_a$  is the region of acceptability; that is, the region in which the constraints (1) are satisfied.

Methods of statistical circuit design can be classified as either deterministic or statistical in nature. The deterministic methods use optimization theory and are definitely more mature than Monte Carlo techniques [1]-[18]. However, determinis-