[29] L. Kleinrock, "Queing Systems, Vol. 1 and 2", John Wiley and Sons, Inc., 1975.

[30] M. Shwartz, "Computer-Communication Network Design and Analysis," Prentice Hall, Inc., 1977.

[31] G. Longo, ed., "Multi-User Communication Systems (CISM Courses Lectures Series), Springer-Verlag, 1981.

[32] W. Chou, ed., "Computer Communications, Vol. 1--Principles," Prentice Hall Inc., 1983.

[33] S. C. A. Thomopoulos, "A Unifying State Space Approach For Modeling, Estimation and Control of Random Access Communication Networks With and Without Collision-Resolution Algorithms," IEEE 1983 International Symposium on Information Theory, St. Jovite, Quebec, Canada, September 26-30, 1983. Also submitted for publication to the IEEE Transactions on Information Theory.

Proceed. Int. AMSE Conf. "Modelling & Simulation", Minneapolis, August 13-17, 1984, Vol. 2, p. 99-117

DIGITAL INTEGRATED CIRCUIT TRANSIENT ANALYSIS PROGRAM

Bogdan M. WILAMOWSKI, Douglas J. HAMILTON and Zbigniew J. STASZAK
University of Arizona, Tucson, AZ, USA

ABSTRACT

A program for transient analysis of very large digital integrated circuits, based on a CHArge COnservation principle, (CHACO), is described. A developed relatively simple explicit algorithm uses the unbalance of the conduction currents at each node to compute the charge stored on capacitances connected to each node.

The program is oriented on digital MOS circuits and is suitable only for transient analysis. A variety of circuits ranging from simple inverters to ring oscillators, transmission gates and flip-flops have been analyzed. For small-complexity bipolar circuits, the CPU time is similar to those needed for the SPICE2 circuit simulation program. However, for MOS circuits even with the increased complexity, the computing time is significantly shorter (15-100 times for a medium size MOS circuit, depending on accuracy of solution).

1. INTRODUCTION

To improve the computing speed of CAD tools, various approaches are used, such as sparse matrix techniques, implicit integration methods, a sparse tableau analysis method, a modified nodal analysis method, circuit decomposition or a modular approach e.g. [1],[2],[3]. Also a waveform relaxation method [4], which takes advantage of signal latency requires a very large memory and, in the case of circuits with many feedback loops, its efficiency is rather poor. A method to compress storage data by one or two orders of magnitude has recently been published [5].

In this paper, we describe a simple explicit method in which the computing time for MOS medium-size circuits is up to 15-100 times shorter than that required by the SPICE2 program [6], [7], the higher number being for the default values of optional parameters of SPICE2, which were used to obtain the required accuracy. SPICE2 is used here to check the accuracy of our solutions, and as a reference only for speed comparison.

2. PRINCIPLE OF THE ALGORITHM

At any node in the circuit consisting of nonlinear resistors, sources and capacitances, displacement current must flow if the algebraic sum of conduction currents flowing into a node is not zero. These displacement currents result from the charging of capacitances connected to the node. In general, a set of nonlinear differential equations must be solved:

$$
\begin{aligned}
&I_{1j}(V) + C_{1j}(v_{1j})\, dv_{1j}/dt = 0\\
&\cdots\cdots\cdots\cdots\cdots\\
&I_{ij}(V) + C_{ij}(v_{ij})\, dv_{ij}/dt = 0\\
&\cdots\cdots\cdots\cdots\cdots\\
&I_{Nj}(V) + C_{Nj}(v_{Nj})\, dv_{Nj}/dt = 0
\end{aligned}
\qquad (1)
$$

where: $I_{ij}(V)$ are the conduction currents flowing into node i, $C_{ij}(v_{ij})$ are grounded (i=j), and coupled ($i \neq j$) capacitances (both $I_{ij}(v)$ and $C_{ij}(v_{ij})$ may be nonlinear functions of voltage), v_{ij} are voltages between nodes j and i, and N is the total number of nodes.

If it is assumed that all voltages are known, the values of all conduction currents can be calculated. The net conduction current I_i flowing into a node for a time interval Δt produces a charge increment $\Delta Q_i = I_i \Delta t$ which is distributed among the capacitances connected to the node. In an integrated circuit the capacitances are nonlinear functions of node voltages of conduction currents; thus, if the node voltages are known, the capacitances can be calculated. Then, a network of capacitors of known capacitance can be analyzed with the previously calculated value of ΔQ_i at each node; from this the corresponding Δv_i at each node can be determined. For the next time interval, the node voltages are incremented by their appropriate Δv_i. New I_i, ΔQ_i and Δv_i are then computed.

The algorithm is thus implemented as follows:

1. If initial node voltages are not known, assume plausible values (perhaps 0) and compute the net conduction current at each node:

$$I_i = \sum_{j=1}^{K} I_{ij}(V) \qquad (2)$$

where the summation indicates that there are K conduction currents flowing into node i. Note that these may be nonlinear functions of various node voltages. In the static case, the I_i are, of course, all zero.

2. Calculate for each node the charge increment accumulated during Δt:

$$\Delta Q_i = \int_{t}^{t+\Delta t} I_i(t)\, dt \qquad (3)$$

Since the functions $I_i(t)$ are generally unknown, and only discrete values for the previous time steps were calculated, the shape of this function can be predicted using zero, first or second order interpolations methods.

With the zero order interpolation:

$$\Delta Q_i = \Delta t * I_i(t) \qquad (4)$$

with the first order interpolation:

$$\Delta Q_i = \Delta t * [1.5\, I_i(t) - 0.5\, I_i(t-\Delta t)] \qquad (5)$$

with the second order interpolation:

$$\Delta Q_i = \Delta t * [1.75\, I_i(t) - I_i(t-\Delta t) + 0.25\, I_i(t-2\Delta t)] \qquad (6)$$

In the last case, values from two previous time periods must be stored in memory. For the zero-order case, very short time steps must be used to achieve the required accuracy. As a compromise, the first-order case was used in examples to be given.

3. Using an appropriate interpolation method, predict the values of all nonlinear capacitances for the time $t+\Delta t$. For example, with the first order interpolation:

$$C(t+\Delta t) = 2 * C(t) - C(t-\Delta t) \qquad (7)$$

4. Compute Δv_i at each node. Since ΔQ_i are known, as are the capacitance values of all capacitors, the following set of equations representing the capacitor network must be solved:

$$
\begin{aligned}
&C_1 v_1 + \sum_{j=1}^{N} C_{1j}(v_j - v_1) = Q_1\\
&\cdots\cdots\cdots\cdots\cdots\\
&C_i v_j + \sum_{j=1}^{N} C_{ij}(v_j - v_i) = Q_i\\
&\cdots\cdots\cdots\cdots\cdots\\
&C_N v_N + \sum_{j=1}^{N} C_{Nj}(v_j - v_N) = Q_N
\end{aligned}
\qquad (8)
$$

where: C_i, C_{ij} are interpolated values of grounded and coupled capacitances, and Δv_i, Δv_j are increments of node voltages caused by the charge increment ΔQ_i.

Note that if there are N nodes Eq. (8) represents a system of N linear equations. Solution of these equations is discussed in the next section.

5. Compute the new values of node voltages.

$$v_i(t+\Delta t) = v_i(t) + \Delta v_i \tag{9}$$

In the implementation of the algorithm, the time period of interest, T, is divided into subintervals called external time steps. Data points are to be calculated at each external time step. The external time steps are divided into intervals called internal time step. An interactive procedure using the internal time steps is carried out to obtain convergence for each external time step.

3. SOLUTION OF THE CAPACITIVE NETWORK

As was discussed in step 4 of the algorithm, to compute v_i a network of capacitors must be analyzed. Such a network is characterized by a matrix equation in the general form:

$$[C_{ij}]\,[v_i] = [\Delta Q_i] \tag{10}$$

Since the capacitance values for a given time interval are held constant, Eq. (10) is linear. Moreover, since the capacitive network is passive, the main diagonal of the C_{ij} matrix is always dominant. Therefore, a simple Gauss-Seidel iterative procedure can be applied, and convergence is guaranteed.

In many cases, the grounded capacitors are dominant, and convergence is very rapid. In examples that have been analyzed with the Gauss-Seidel procedure, convergence was typically obtained in 10 to 20 iterations when the grounded capacitances were dominant. However, convergence is slow in cases where large capacitances between nodes occur. As an example, consider the circuit of Fig. 1. In this case, the capacitances between nodes are up to 10000 times greater than the grounded capacitances, and convergence was not obtained even after 100 iterations; this is illustrated in Fig. 2.

A method producing a more rapid convergence assumes an exponential relationship for the node voltage increments:

$$\Delta v_i(k+h) = \Delta v_i(k)*[1 - \exp(-h^{\alpha}{}_i)] \tag{11}$$

where: $\Delta v_i(k)$ is the voltage increment at node i for the k-th iteration, $\Delta v_i(k+h)$ is the voltage increment at node i for the k-th iteration, α_i is the "attenuation constant" defined by:

$$\alpha_i = 1/\ln[v_i(k-1)/v_i(k)] \tag{12}$$

Therefore, after a few steps of using the standard Gauss-Seidel iterative procedure, new predicted values of node voltages can be computed such as the k+100-th step. Fig. 3 shows that, even for networks with large capacitances between nodes (Fig. 1), rapid convergence is obtained.

4. SIMULATION OF BIPOLAR INTEGRATED CIRCUITS

To test the algorithm, a cascade of 4 bipolar inverter stages shown in Fig. 4 was analyzed. Such a saturating type of circuit has very nonlinear capacitances. The equivalent circuit for the bipolar transistors is shown in Fig. 5; functional dependences of the nonlinearities are given in Table 1.

Figure 6 shows the computed waveforms for the node voltages for a time increment of 1 nsec and total time period of 100 nsec. Note that there are no differences between those figures. All computations were performed on a VAX-11/780 computer; for our algorithm the CPU time was 9.10 seconds while with SPICE2 it was 23.66 seconds. The times are not very different due to the fact that the circuit contains large highly nonlinear capacitances which are not grounded and also static characteristics of bipolar transistors are very nonlinear in nature. Thus, the explicit algorithm does not have significant advantage over the implicit method used in SPICE2. In order to secure convergence for this particular circuit, the internal time step was 12 times smaller than the external time step of 1ns. In other words, the maximum possible internal time increment in computing the algorithm was 0.08 nsec. For larger internal time steps, convergence was not obtained.

5. SIMULATION OF MOS INTEGRATED CIRCUITS

The method described herein is more suitable for simulation of MOS circuits than bipolar circuits because the nonlinearities in the former are less severe. A cascade of 7 CMOS inverters shown in Fig. 8 was analyzed. The device

equivalent circuit is shown in Fig. 9, and the functional dependences of the parameters are given in Table 2. This MOS transistor model includes channel length modulation, body effect due to substrate biasing, and subthreshold conduction as well as normal and inverted operation. The model does not, however, include nonlinearities of the gate capacitances.

The computed node voltages of the circuit of Fig. 6 are shown in Fig. 10; results obtained with SPICE2 are shown in Fig. 11. No significant differences are observed. For our method the CPU time was 1.99 seconds, while for SPICE2 it was 190.72 seconds for the default values of optional parameters [7]. Since the MOS transistor parameters are inherently not as strongly nonlinear as those of bipolar transistor, an internal time step equal to the external time step (2ns) was used.

Comparison with the SPICE2 program was also performed for a cascade of 4 CMOS inverters and a cascade of 9 CMOS inverters. The CPU times for various circuits and computing options are summarized in Table 3.

Thus, our method is attractive for large circuits. To illustrate this property, a relatively large circuit shown in Fig. 12 was analyzed; the results are shown in Fig. 13. The CPU time for this circuit with 97 inverters (194 transistors) was only 23.04 seconds. It can, thus, be predicted that a circuit of 2000 MOS transistor circuit will require about 230 seconds of CPU time. Circuits with transmission gates, with flip-flops and ring oscillators, were also analyzed. Generally, CPU time increased proportionally with the number of circuit elements and number of nodes. Since static analysis does not precede transient analysis, no problems with convergence occur. Our program will always give a solution if a small enough internal time step is chosen.

6. CONCLUSION

The method we have described for the transient analysis of nonlinear networks of resistors, capacitors and sources requires a CPU time proportional to the circuit complexity in contrast to classical matrix methods CPU where time is proportional to N^2 or more sophisticated approaches which show dependence on N^m where m is in the range of 1.2 to 1.5.

While the Gauss-Seidel iteration procedure used here has the advantage of simplicity, it is not very suitable for circuits which have large inter-node capacitances. However, it should be noted that, for such cases, a significant reduction of computing time is possible.

Further reduction of CPU can be expected if means are used to take advantage of signal latency (temporary sparsity). The algorithm is so structured that it is relatively simple to omit computation at inactive nodes when this is warranted.

REFERENCES

[1] A. E. Ruehli, G. S. Ditlow, "Circuit analysis, logic simulation, and design verification for VLSI", Proc. IEEE, vol. 71, pp. 34-48, January 1983.

[2] G. D. Hachtel, R. K. Brayton, and F. Gustavson, "The sparse tableau approach to network analysis and design", IEEE Trans. Circuit Theory, vol. CT-18, pp. 101-113, January 1971.

[3] C. Ho, A. E. Ruehli, P. A. Brennan, "The modified nodal approach to network analysis", IEEE Trans. Circuits and Systems, vol. CAS-22, pp. 504-509, June 1975.

[4] E. Lelarasmee, A. E. Ruehli, A. L. Sangiovanni-Vincentelli, "The waveform relaxation method for time-domain analysis of large-scale integrated circuits", IEEE Trans. CAD Integrated Circuit and Systems, vol. CAD-1, pp. 131-145, July 1982.

[5] O. A. Palusinski, M. W. Guarini, "Improving the efficiency of relaxation based circuit simulated via functional linearization and polynomian representation of solutions", IEEE Conf. on IC CAD, Santa Clara, CA, September 12-15, 1983.

[6] L. W. Nagel, "SPICE2: computer program to simulate semiconductor circuits", University of California, Berkeley, ERL Memo, ERL-M520, May 1975.

[7] A. Vladimirescu, K. Zhang, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, "SPICE Version 2G User's Guide", University of California, Berkeley, Tech. Memo, August 10, 1981.

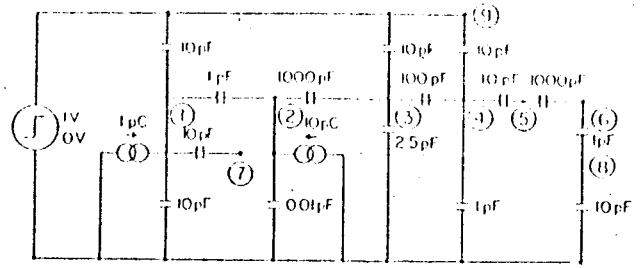

Fig. 1. Example of capactive circuit activated by voltage source applied to node 9 and by injected charges into nodes 1 and 2.

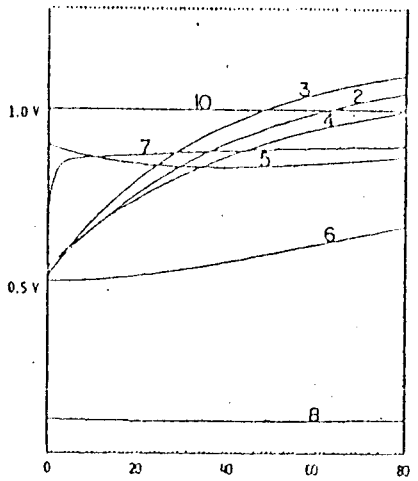

Fig. 2. Node voltage increments v_i for the circuit shown in Fig. 1 as a function of number of iterations of a Gauss-Seidel algorithm.

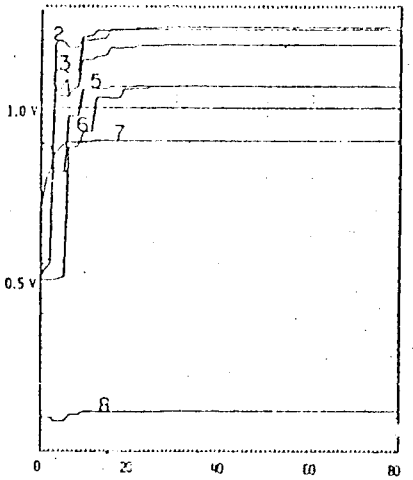

Fig. 3. Node voltage increments v_i for the circuit shown in Fig. 1 as a function of number of iterations of a modified Gauss-Seidel algorithm.

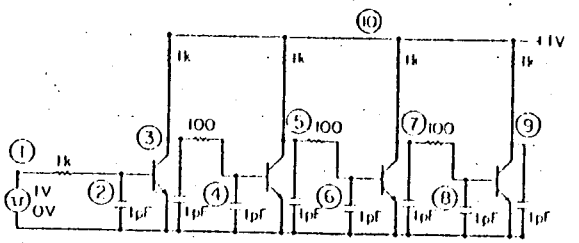

Fig. 4 Bipolar circuit analyzed with CHACO and SPICE2 programs.

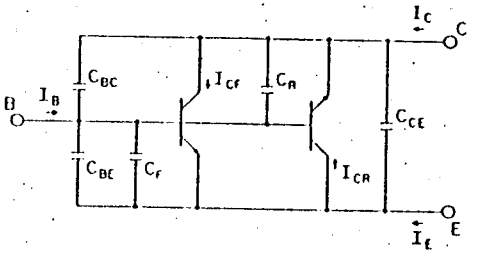

Fig. 5 · Equivalent circuit for the bipolar NPN transistor mode used in CHACO.

$$I_{CF} = I_{SF} * [\exp(V_{BE}/V_T) - 1] * (1 + V_{CE}/V_F)$$

$$I_{CR} = I_{SR} * [\exp(V_{BC}/V_T) - 1] * (1 + V_{EC}/V_R)$$

$$I_B = I_{CF}/\beta_F + I_{CR}/\beta_R$$

$$I_C = I_{CF} - I_{CR} * (1 + 1/\beta_R)$$

$$I_E = I_{CR} - I_{CF} * (1 + 1/\beta_F)$$

$$C_{BE} = C_{BEO} * (V_{EB} + V_B)^{-0.5}$$

$$C_{BC} = C_{BCO} * (V_{CB} + V_B)^{-0.5}$$

$$C_F = I_{CF} * \tau_F/V_T = I_{SR} * \tau_F/V_T * \exp(V_{BE}/V_T)$$

$$C_R = I_{CR} * \tau_R/V_T = I_{SR} * \tau_R/V_T * \exp(V_{BC}/V_T)$$

Table 1. Functional dependence of NPN bipolar transistor model.

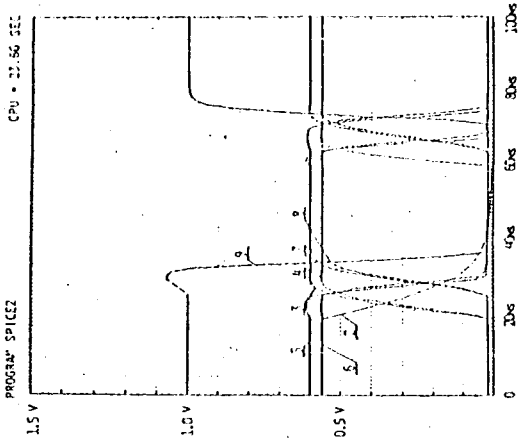

Fig. 7. Computed waveforms of the node voltages for the circuit of Fig. 4 obtained with program SPICE2.

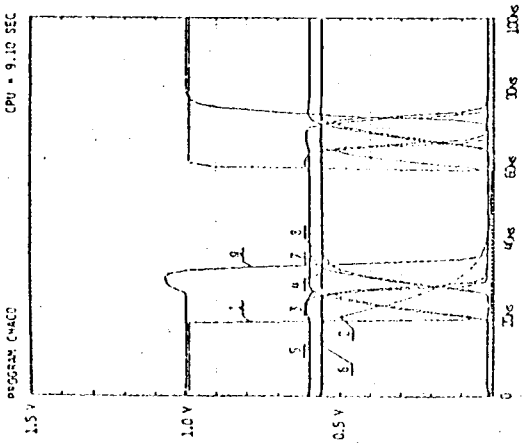

Fig. 6. Computed waveforms of the node voltages for the circuit of Fig. 4 obtained with program CMOS.

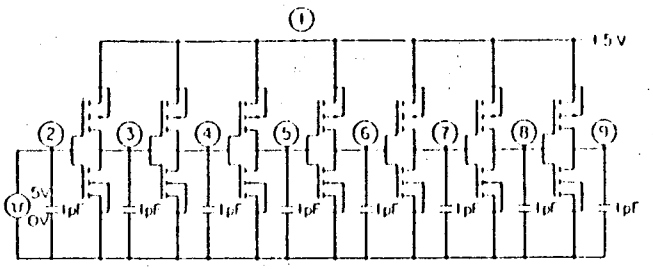

Fig. 8 Cascade of 7 CMOS inverters analyzed with CMOS and SPICE2 Programs.

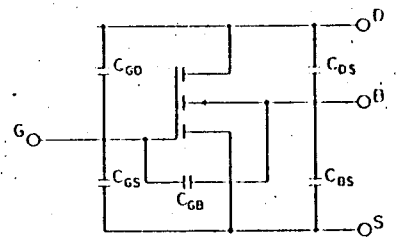

Fig. 9. Equivalent circuit for the N-channel MOS transistor mode used in CMOS.

V_1 = absolute value of V_{DS}, V_2 = smaller value of V_{SB} and V_{DB},

V_3 = greater value of V_{GS} and V_{GD},

$V_4 = V_3 - V_{Th}$, $V_5 = \Delta V * \exp(V_4/\Delta V - 1)$. $V_4 = V_5$ if $V_4 < \Delta V$,

V_6 = smaller value of V_4 and V_1,

$V_{Th} = V_{Th0} + \gamma * [(\phi - V_2)^{0.5} - \phi^{0.5}]$,

$I_D = \beta*(V_4 - 0.5*V_6) * V_6*(1 + \chi V_1)$. $I_D = -I_D$ if $V_{DS} < 0$

$C_{BS} = C_{BS0} * (V_{BS} + \phi)^{0.5}$, $C_{DS} = C_{DS0} * (V_{BD} + \phi)^{0.5}$

Table 2. Functional dependence of N - channel MOS transistor parameters.

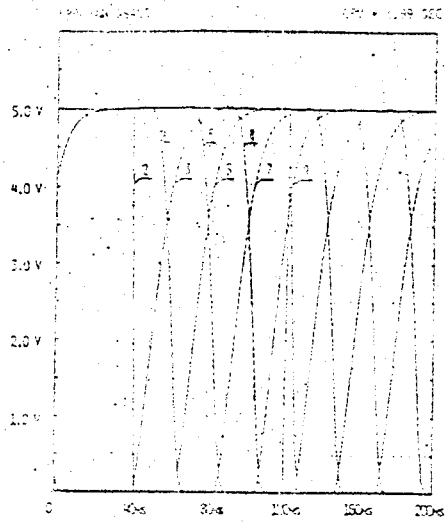

Fig. 10. Computed waveforms of the node voltages for the circuit of Fig. 8 obtained with program CINOO.

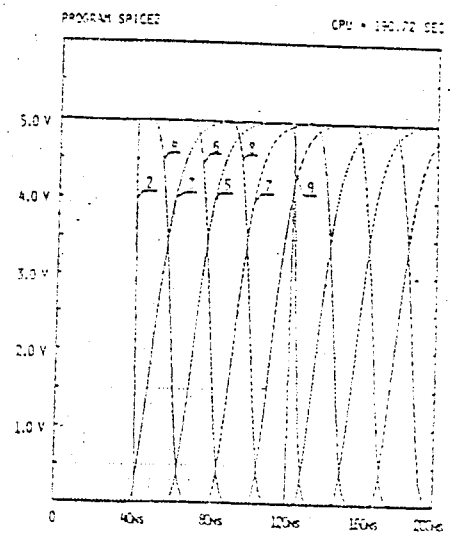

Fig. 11. Computed waveforms of the node voltages for the circuit of Fig. 8 obtained with program SPICE2.

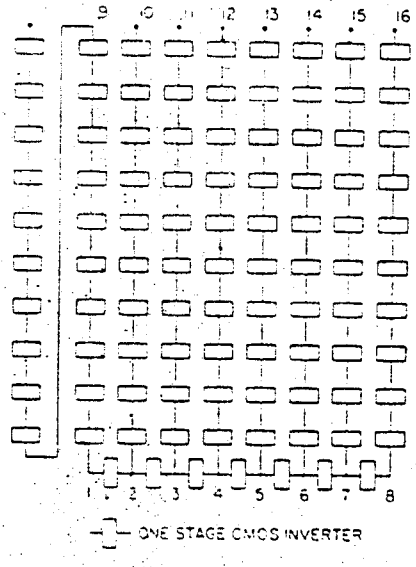

Fig. 12. Block diagram of the circuit containing 97 CMOS inverters for analysis with CINOO program.

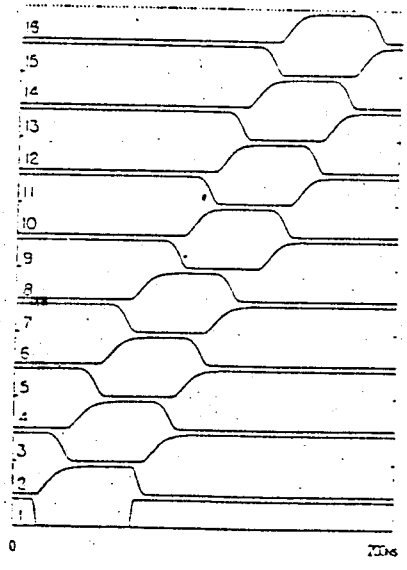

Fig. 13. Computed waveforms of some of node voltages for the circuit of Fig. 12.

Table 3.

CMOS INVERTERS	CHAOS	SPICE2			
		For Default Values of Optional Parameters		For Given Values of Optional Parameters*	
		IDS MODEL Level 2	Level 1	IDS MODEL Level 2	Level 1
4 stages 8 transistors 6 nodes	1.26 sec	75.78	27.90	20.14	16.25
stages transistors nodes	1.99 sec	190.72	54.99	31.04	25.20
9 stages 18 transistors 11 nodes	2.34 sec	264.99	74.29	37.65	30.39
42 stages 84 transistors 27 nodes	23.04 sec				

* RELTOL = 0.1, ABSTOL = 10nA, VNTOL = 10mA, CHGTOL = 10^{-17}, ITL2 = 20, ITL3 = 7

Proceed. Int. AMSE Conf. "Modelling & Simulation", Minneapolis, August 13-17, 1984, Vol. 2, p. 1-12

A METHODOLOGICAL APPROACH TO THE OPTIMIZATION OF VOLTAGE QUALITY IN DISTRIBUTION NETWORKS (*)

G. CARPINELLI - R. MONGELLUZZO - A. TESTA (**)

Abstract. The problem of the optimal joined choice of the transformer ratios in a radial distribution network is considered. The optimal condition is assumed to be the one minimizing a function of the voltage deviation costs. Resolution approaches are proposed taking into account the discreteness of problem variables. The situations that can occour both in public and industrial distribution networks are considered.

List of principal symbols:

N	: number of the load buses;
M	: number of branches (including transformers);
m	: number of the secondary substation transformers;
u	: number of the taps in the HV/MV substation transformer;
v	: number of the taps in the MV/LV secondary substation transformers;
D	: costs caused by slow voltage deviations;
α	: cost coefficient;
$D(c_i)$	: diagonal matrix whose elements along the diagonal are the components of the column vector [c];
[R], [X]	: column vectors of the branch resistances and reactances, respectively;
[A*]	: network topological matrix;
[A]	: reduced topological matrix;
[A]'	: transpose of the matrix [A];
[P], [Q]	: column vectors of the load active and reactive powers, respectively;
[P*], [Q*]	: column vectors of the branch active and reactive powers, respectively;
[ΔU]	: column vector of the branch voltage drops;
$[\Delta V_e]$	: column vector of the total voltage drops between the supplying node and the load nodes;
a_0	: HV/MV transformer ratio;
a_i	: ratio of the i-th MV/LV transformer;
a_0^*	: HV/MV transformer ratio value which minimizes the costs;

(*) Work supported by M.P.I. - Ministero della Pubblica Istruzione, Italy

(**) Guido Carpinelli, Energy Electrical Department, University of Naples, Italy; Raffaele Mongelluzzo and Alfredo Testa, Electrical Department, University of Calabria, Italy.