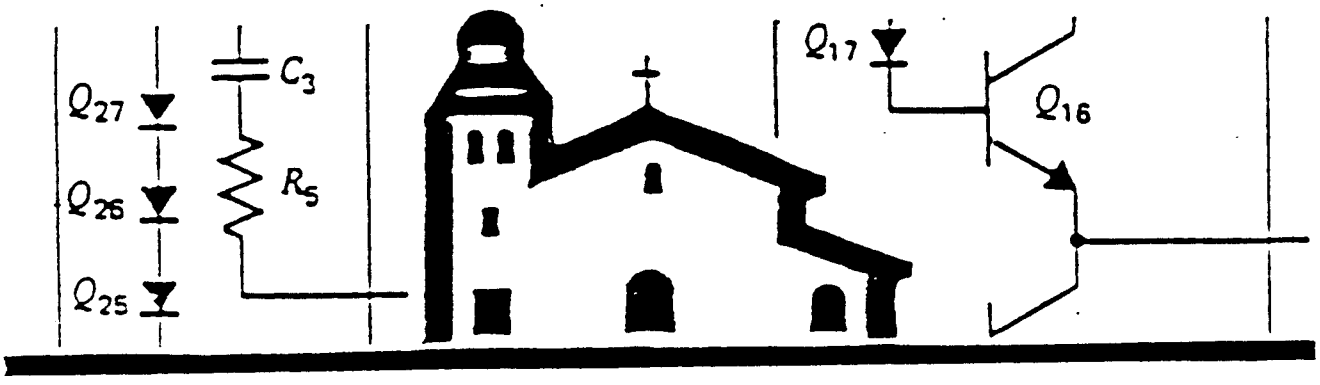
1986 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS

Le Baron Hotel
San Jose, California
May 5 – 7, 1986

86CH2255-8



CHARCO - IC TRANSIENT ANALYSIS PROGRAM

Bogdan M. Wilamowski, Zbigniew J. Staszak, Douglas J. Hamilton

Department of Electrical and Computer Engineering,
University of Arizona, Tucson, AZ 85721.

ABSTRACT

A program for transient analysis of large scale digital integrated circuits, based on a CHARge COntrol approach, is described. A relatively simple explicit algorithm has been developed. The proposed algorithm uses the unbalance of the conduction currents at each node to compute the charge stored on capacitances connected to that node. The program CHARCO was originally designed for digital MOS circuits and is suitable only for transient analysis. Circuits ranging from simple inverters to ring oscillators, transmission gates, and flip-flops have been analysed. For small-complexity bipolar circuits, the CPU time is similar to that needed for the SPICE2 circuit simulation program. However, for medium scale MOS circuits the computing time is 10 to 100 times shorter. The comparative advantage of the CHARCO program increases with circuit complexity, since the CPU time increases linearly with circuit size.

INTRODUCTION

Increased complexity of integrated circuits makes their analysis more difficult. To handle that, special purpose computer programs have to be utilized [1]. One of the problems arises from the fact that usually the CPU time increases much faster than the size of the analysed circuit.

In most of the circuit analysis programs transient circuit analysis is performed in such a way that for each time step multiple circuit linearization and solution of linear circuit equations is carried out leading to new values of voltages for the next linearization step. Waveform relaxation method developed at Berkeley [2] is a notable exception to that rule. The algorithm operates in the time domain using voltages as function of time rather than the instantaneous values. Though the method requires a large computer memory, it nevertheless allows for nearly 50-fold reduction in the CPU time in comparison with widely used circuit analysis program SPICE2 [3]. The disadvantage of this algorithm is that the CPU time is clearly dependent on circuit topology and its practical application is limited to digital circuits with low number of feedback loops.

In this paper, we describe a simple explicit method in which the computing time for MOS medium-size circuits is on average 10-100 times shorter than that required by the SPICE2 program, the higher number being for the default values of optional parameters of SPICE2, which were used to obtain the required accuracy. Explicit integration method has not been widely used because of convergence problems, requiring time steps smaller than the smallest time constants in the circuit [4][5]. It is true that this can cause problems for analog circuits. However, this method can be very attractive for digital IC simulation. Digital circuits are designed in such a way that all sub-circuits have time constants of the same order and they are usually constrained by technology limitations.

The program CHARCO has a similar input data structure to SPICE2. However, it is not identical since it is based on a different algorithm, is only for transient analysis and uses other control parameters.

PRINCIPLE OF THE ALGORITHM

It is assumed that the circuit to be analysed has the following properties:

- It contains capacitive elements, the values of which can be treated as constants in each time step. These values can be of course modified after each time step.

- It contains non-inertial elements with current(s) that may be non-linear functions of two or more nodal voltages.

CH2255-8/86/0000-0548 $1.00 © 1986 IEEE

In such a circuit, for the i-th node the algebraic sum of currents is zero and the following relation is fulfilled:

$$F_i(V_1, V_2, \ldots, V_N) + \sum_{j=1}^{N} C_{ij} \frac{dV_j}{dt} = 0 \quad (1)$$

The first component in expression (1) is the sum of conduction currents and the second is the sum of capacitive currents. The number of equations is equal to the number of independent nodes and the nodal voltages are the unknowns. The standard approach is to perform the linearization of such equations and then apply the Newton-Raphson and sparse matrix techniques.

The proposed algorithm is based on the assumption that for each time step the conduction currents are constant since the changes of voltage after each time step are so small that they do not affect in any significant way the flow of current in the circuit. In the simplest case the value of the conduction current can be taken from the preceding time step, or it can be the average value for this time step estimated with respect to changes in the preceding time steps. Even in the most precise case, the average value can be determined from one or more test time steps. It is worth noting that the values of conduction currents in (1) are given in explicit form, and the method for their computations depends only on the adopted description of element models.

Since the values of conduction currents are known and constant capacitances are assumed during a given time step, the problem simplifies to the solution of the set of linear equations. Each node is described by the expression:

$$\sum_{j=1}^{N} C_{ij} \frac{dV_j}{dt} = F_i \quad (2)$$

F_i corresponds to the values of the unbalanced conduction currents ΔI_i in the i-th node, and is computed explicitly. The resulting set of linear differential equations is numerically solved by introducing discrete time steps Δt. In the simplest case the forward Euler method can be used, and an algebraic set of linear equations is obtained:

$$[C_{NN}] [\Delta V] = [\Delta t * \Delta I] = [\Delta Q] \quad (3)$$

The above relation describes the capacitive network with forced nodal charges Q. The solution is the node voltage increments after each time step. Of course, one can adopt more precise methods of solution of the set of differential equations (2). A possible approach is to use a higher-order Runge-Kuta method or one of the implicit solution methods.

A physical interpretation of the algorithm is that it is based on controlled charge flow between capacitances of the circuit. At each time step, because of conduction currents, charge is distributed among circuit capacitances. The algorithm of the program is thus based on the Charge Control approach, hence the name CHARCO.

A passive capacitive network is described by a matrix with a dominant main diagonal. Therefore, calculations of charge distribution among capacitances can be performed by using simple iterative procedures (e.g. Gauss-Seidel). In the case of integrated circuits, the efficiency of such algorithms can be very good because grounded capacitances usually far outnumber coupling capacitances (except for saturated bipolar digital circuits), thus the dominance of the main diagonal is overwhelming and the convergence is fast. However, the convergence is slow for circuits with large coupling capacitances. For that case a method of rapid convergence was developed based on the assumption that the variation of node voltages with time can be approximated by an exponential function of time [5].

TEST RUNS

To test the algorithm and the program, circuits ranging from simple inverters to ring oscillators, transmission gates and flip-flops have been analysed. The equivalent circuits for the bipolar NPN and N-channel MOS transistors are presented in Figures 1 and 2, and the functional dependence of the respective transistor models are shown in Tables 1 and 2.

The CPU times for selected circuits and various computing options are summarized in Table 3. All computations were performed on a VAX-11/780 computer.

The method described is attractive for large circuits. Generally, the CPU time increased linearly with the number of circuit elements and number of nodes. Since static analysis does not precede transient analysis, no problem with convergence occurs in regenerative-type circuits. The program will always give a solution if a small enough internal time step is chosen.

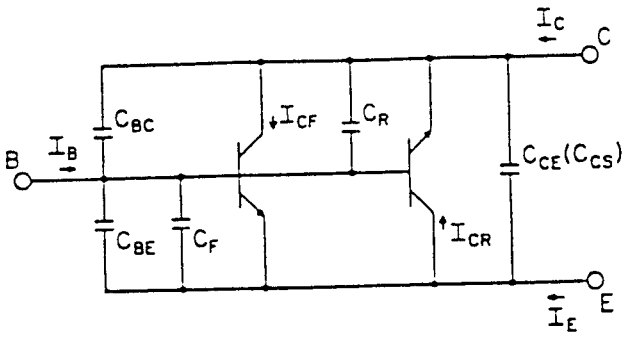


Fig.1. Equivalent circuit for the bipolar NPN transistor model.

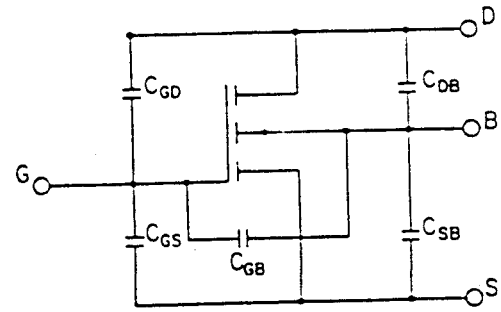


Fig.2. Equivalent circuit for the N-channel MOS transistor model.

TABLE 1

FUNCTIONAL DEPENDENCE OF THE NPN BIPOLAR TRANSISTOR MODEL

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

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

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

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

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

$$C_{BE} = C_{BEO} \cdot (V_{EB} + \phi)^{-0.5}$$

$$C_{BC} = C_{BCO} \cdot (V_{CB} + \phi)^{-0.5}$$

$$C_F = I_{CF} \cdot \tau_F / V_T$$

$$C_R = I_{CR} \cdot \tau_R / V_T$$

I_{SF}, I_{SR} - forward and reverse saturation currents,
V_T - electrothermal potential,
V_F, V_R - forward and reverse Early voltages,
β_F, β_R - forward and reverse current gains,
C_{BEO}, C_{BCO} - zero-bias depletion capacitances of B-E and B-C junctions,
ϕ - built-in potential,
τ_F, τ_R - storage times for B-E and B-C junctions (forward and reverse transit times).

TABLE 2

FUNCTIONAL DEPENDENCE OF THE N-CHANNEL MOS TRANSISTOR MODEL

$$V_1 = \text{abs}(V_{DS})$$

$$V_2 = V_{SB};\ \text{if } V_{DB} < V_{SB} \text{ then } V_2 = V_{DB}$$

$$V_3 = V_{GS};\ \text{if } V_{GD} > V_{GS} \text{ then } V_3 = V_{GD}$$

$$V_{TH} = V_{THO} + \gamma \cdot [(\phi - V_2)^{0.5} - \phi^{0.5}]$$

$$V_4 = V_3 - V_T$$

$$V_5 = \Delta V \cdot \exp[(V_4/\Delta V) - 1];$$

$$\text{if } V_4 < \Delta V \text{ then } V_5 = V_4,$$

$$\text{if } V_1 < V_4 \text{ then } V_5 = V_1$$

$$I_D = \beta \cdot (V_4 - 0.5 V_5) \cdot V_5 \cdot (1 + \lambda V_1);$$

$$\text{if } V_{DS} < 0 \text{ then } I_D = -I_D$$

$$C_{SB} = C_{SBO} \cdot (V_{BS} + \phi)^{0.5}$$

$$C_{DB} = C_{DBO} \cdot (V_{BD} + \phi)^{0.5}$$

V_{THO} - zero-bias threshold voltage,
γ - bulk threshold parameter,
ϕ - built-in potential,
ΔV - subthreshold conduction parameter,
β - device transconduction parameter,
λ - channel-length modulation parameter,
C_{SBO}, C_{DBO} - zero-bias S-B and D-B capacitances.

TABLE 3

RESULTS OF CPU TIMES FOR SELECTED CIRCUITS (IN SEC)

ANALYSED CIRCUIT	CHARCO	SPICE2G			
		For default values of optional parameters;		For given values of optional parameters;*	
4-stage bipolar inverter; 10 nodes, 4 transistors, 8 resistors	9.10	23.66			
CMOS inverters		MOS model			
		Level 2	Level 1	Level 2	Level 1
4 stages; 6 nodes, 8 transistors	1.26	75.78	27.90	20.14	16.25
7 stages; 9 nodes, 14 transistors	1.99	190.72	54.39	31.04	25.20
9 stages; 11 nodes, 18 transistors	2.34	264.99	74.29	37.65	30.39
97 stages; 99 nodes, 194 transistors	23.04				

RELTOL = 0.1, ABSTOL = 1 nA, VNTOL = 10 mA, CHGTOL = 1E-12, ITL2 = 20, ITL3 = 2 [4]

CONCLUSIONS

The results show, that the described algorithm is fast. This is specially evident for MOS circuits where the coupling capacitances between nodes are small and the characteristics of MOS transistors are less nonlinear than those of bipolar transistors, where large nonlinear collector-base capacitances and saturation may be encountered.

The method we have described for the transient analysis of nonlinear networks of resistors, capacitors and sources requires the CPU time proportional to the circuit complexity in contrast to other methods, where the CPU time rapidly increases with the increase of circuit nodes. The advantage of the proposed program is especially evident for large circuits. It also has small memory requirements because of a compressed form of data structure.

CPU time for large circuits can be further reduced by taking advantage of signal latency, that is, by identifying nodes at which voltages change negligibly during a time increment. In such cases, it is easy to temporarily omit some of the nodes during the analysis. Such an approach is rather difficult if standard matrix techniques are used.

One of the limitations of the algorithm is that the minimum time step has to be smaller than the smallest time constant. If one needs to perform the analysis for long time periods, than a simple approach is to artificially increase the smallest circuit time constants making the values of these time constants comparable to the time step. This guarantees the convergence of the numerical process, but does not significantly affect its accuracy.

REFERENCES

[1] A. E. Ruehli and G. S. Ditow, "Circuit analysis, logic simulation and design verification for VLSI," PROC. of the IEEE, Vol. 71, pp. 34-48: Jan. 1983.

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

[3] A. Vladimirescu, K. Zhang, A. R. Newton, D. O. Peterson, and A. Sangiovanni-Vincentelli," SPICE Version 2G User`s Manual, Univ. of California, Berkeley, Tech. Memo: Aug. 10, 1981.

[4] G. D. Hatchel and A. L. Sangiovanni-Vincentelli, "A survey of third generation simulation techniques," PROC. of the IEEE, Vol. 65, pp. 1264-1280: October 1981.

[5] R. A. Rohrer, H. Nosrati and K. W. Heizer, "Quasi-static control of explicit algorithms for transient analysis," IEEE Trans. CAD of ICAS, vol. CAD-3, pp. 226-234: July 1984.

[5] B. M. Wilamowski, D. J. Hamilton, and Z. J. Staszak, "Digital integrated circuit analysis program," PROC. INT. AMSE CONF. MODELING & SIMULATION," vol. 2, pp. 92-112, Minneapolis, MN, Aug. 13-17, 1984.