KYBERNETIKA — VOLUME 26 (1990), NUMBER 3

SINGLE-SWEEP TIMING SIMULATION

PETR PAVLÍK, JIŘÍ VLACH

Timing simulators of digital integrated circuits provide an electrical simulation of time characteristics which can detect hazard states, spikes or race conditions in the logic design. The described *method* NMSS (*no matrix, single sweep*) can be used either as a simple tool for the personal computer, enabling the design of blocks of hundreds MOS transistors, or as a part of a mixed-mode simulator for large design systems.

The method uses a simplified *charge conservation model* of MOS transistors. The circuit matrices are not built, only vectors obtained by scanning the network description in a single sweep are stored. No algebraic equations are solved, the unknown values of node voltages are obtained by an explicit integration in the time domain, using a variable two-step predictor/corrector method. The method is powerful enough to handle transmission gates, floating capacitances, or oscillating circuits. However, it is inappropriate for bipolar transistors or stiff systems (e.g. memories).

The method was realized in the *timing simulator* WATTIME at the University of Waterloo, Canada. It combines the benefits of both electrical and logic approach: The *even-driven simulation* uses a *status of node*, similarly to logic simulators, but the designer can choose among analog, multilevel or binary output for each node. The simulation is about one and half order faster than SPICE.

An example of a ring oscillator is given which is a usual test for electrical simulators, showing the resulting waveforms both in the electrical and digital representation.

1. INTRODUCTION

In the analysis of digital circuits, there is a constant demand for speed improvement. The early timing simulators of the last decade such as MOTIS [1] sacrificed accuracy for speed. On the contrary, the classical simulators as SPICE [2] or ASTAP [3] have represented an about one and half order slower but reliable and accurate tool for the circuit analysis.

In this decade, the gap between classical and timing simulators was filled by relaxation-based simulators [4] which substitute certain part of the classical simulation by an iteration process. This can be done at three levels:

LEVEL 1: System of *differential equations* in the time domain (Waveform Relaxation, RELAX 2 [5])
LEVEL 2: System of *nonlinear equations* after discretization of differential equations of Level 1 (Iterated Timing Analysis, SPLICE 1.6 [6])
LEVEL 3: System of *linear equations* in the loop of the Newton method which solves nonlinear equations of Level 2.

At levels 2 and 3, integration and equation solving can be combined into a single block. These are so called Single Sweep (or single pass) methods, and their formulas can be derived as a limit case of levels 2 and 3 if only a single iteration is performed per time step. Such formulas lead to algorithms which are close to *timing simulators* and they often have this name.

Yet a more detailed classification can be made: By substituting certain part of a *classical* simulator by a single iteration, *implicit* single sweep methods are obtained (as the integration method in classical simulators is traditionally implicit). On the contrary, another class of *explicit* single sweep methods exists, and this one is explained in this paper. The explicit formulation also offers the means for no-matrix approach where the equations are formed by scanning device tables only. The combination of both principles, i.e. No Matrix and Single Sweep (NMSS) has become a theoretical background of the simulator WATTIME developed at the University of Waterloo, Canada [7].

2. EXPLICIT TIME DOMAIN INTEGRATION

In the time domain, the system equations of an MOS digital circuit are

$$\boldsymbol{Q}'(\boldsymbol{V}) + \boldsymbol{G}(\boldsymbol{V}) = \boldsymbol{J} \tag{2.1}$$

where $\boldsymbol{Q}'(\boldsymbol{V})$ denotes the time derivative of node charges (i.e. capacitive currents) and $\boldsymbol{G}(\boldsymbol{V})$ expresses the effect of nonlinearities connected to each node (i.e. resistive currents). The external current sources $\boldsymbol{J}$ are supposed to be zero (a voltage driven circuit).

Using a piecewise linear charge model, the term $\boldsymbol{Q}'$ can be linearized within certain region to obtain

$$\boldsymbol{C}\boldsymbol{V}' + \boldsymbol{G}(\boldsymbol{V}) = 0\,. \tag{2.2}$$

Assume that the resistive nonlinearities are described by equations $I_i = \boldsymbol{G}_i(\boldsymbol{V})$. Their values are updated in each new time instant t as they are functions of $\boldsymbol{V}(t)$. In the same manner, there must be made an update of the capacitance matrix $\boldsymbol{C}$ as it is also a function of $\boldsymbol{V}(t)$ (see Sect. 3).

The matrix $\boldsymbol{C}$ contains contributions from grounded capacitances, floating capacitances, and transcapacitances. As a necessary condition for further derivation it is assumed that each node has a nonzero capacitance (constant or dynamic) to ground at any time instant.

There are *two basic ways* how to integrate (2.2). The first way corresponding to classical simulators and their relaxation-based improvements applies an *implicit discretization formula*

$$V_n' = \frac{1}{h_n}(\alpha_0 V_n + \alpha_1 V_{n-1} + \ldots + \alpha_k V_{n-k}) \tag{2.3}$$

Substituting (2.3) into (2.2) a set of *nonlinear* equations is obtained

$$\boldsymbol{f}(\boldsymbol{V}_n) = \frac{\alpha_0}{h_n}\boldsymbol{C}\boldsymbol{V}_n + \boldsymbol{G}(\boldsymbol{V}_n) + \text{const.} = 0\,. \tag{2.4}$$

The implicit relaxation-based simulators either solve (2.4) as N decoupled one-dimensional nonlinear equations (Level 2) or as a whole system of N nonlinear equations (Level 3). The relaxation principle is then applied either as a sequence of one-dimensional Newton steps or the linear system of the multidimensional Newton step is decoupled and the variables $V_1, \ldots, V_N$ are relaxed.

When just one iteration per time step is performed (i.e. only a partial convergence is allowed) then the limit case for both levels is the *implicit single-sweep (timing) algorithm*

$$\boldsymbol{V}_n = \boldsymbol{V}_{n-1} - [\text{diag}\, \boldsymbol{J}_n(\boldsymbol{V}_{n-1})]^{-1} \boldsymbol{f}_n(\boldsymbol{V}_{n-1}) \tag{2.5}$$

where $\boldsymbol{J} = \partial \boldsymbol{f}/\partial \boldsymbol{V}$ is the Jacobian matrix of $\boldsymbol{f}$. Notice that only diagonal elements of $\boldsymbol{J}$ are necessary to be calculated. The equations (2.5) are decoupled, and they can be solved using either Gauss-Jacobi or Gauss-Seidel method. There exist also modifications which calculate the diagonal terms of $\boldsymbol{J}_n$ using a secant method.

The *second* basic approach which has been used in the simulator WATTIME is to integrate (2.2) *explicitly*. If the capacitance matrix $\boldsymbol{C}$ is diagonally dominant (the grounded capacitances at each node is the necessary but not sufficient condition) then it is possible to relax the diagonal elements C_{kk}. All off-diagonal terms are transferred on the right hand side, and the kth equation then becomes

$$C_{kk} V_k' = -G_k(V) - \sum_{\substack{j=1 \\ j \neq k}}^{N} C_{kj} V_j'\,. \tag{2.6}$$

As all external sources are only voltages, the current sources disappear ($\boldsymbol{J} = 0$).

Now instead of discretizing the equation (2.2) using (2.3) the equation (2.6) is *solved for the unknown* $\boldsymbol{V}'$:

$$V_k' = C_{kk}^{-1}\Big[-G_k(V) - \sum_{\substack{j=1 \\ j \neq k}}^{N} C_{kj} V_j'\Big] \tag{2.7}$$

Thus a set of differential equations

$$\boldsymbol{V}' = \boldsymbol{f}(\boldsymbol{V}, \boldsymbol{V}') \tag{2.8}$$

is to be integrated using an *explicit formula*

$$\boldsymbol{V}_n = \boldsymbol{V}_{n-1} + h_n(\beta_0 \boldsymbol{V}_n' + \beta_1 \boldsymbol{V}_{n-1}' + \ldots + \beta_k \boldsymbol{V}_{n-k}')\,. \tag{2.9}$$

Notice that *first* V' is found and just *next* an integration formula calculating V is applied. This suggests that the solution and the integration are divided into two steps.

Now the problem is how to estimate the value of V' on the right hand side of the equation (2.7). Explicit integration methods of the predictor/corrector type bring a useful tool to do this job: The *predictor* V_p is an approximation of V in a future time point. Therefore also *its derivative* represents our best known approximation of V'. The derivative is obtained easily as an analytical derivative of the predictor formula with respect to the variable step size. As the *corrector* does not require to know the future, the value V' obtained from the predictor step is used.

The common feature of all single sweep (SS) methods is to use only one iteration for the solution of circuit equations. Thus an explicit SS method uses a *single iteration* of (2.7) followed by a *single predictor/corrector* step. In case of good convergence, the corrector is skipped.

There is an important reason why only a single iteration is used: for the stability of explicit methods, it is important that a *short step* is used. The shorter step, the better is the estimate of V' for the substitution into (2.7). In the combination "equation solving + integration" the process appears like *shifting each iteration* of (2.7) a step h_n ahead instead of making several iterations at the end of a bigger step $H_n = \sum h_n$. Yet another but less significant advantage of the small step is the closer approximation of edges of the piecewise linear model of MOS transistor charge.

3. DEVICE MODELS IN THE SIMULATOR WATTIME

As the circuit model is the most time consuming part of the simulator, the *minimal number of model calls* is the goal (mostly one call per time step). There is also a reasonable compromise between the complexity of the model and the accuracy. WATTIME has two levels of MOS models:

1. *Constant threshold voltage and substrate potential. The conductance degradation* is *neglected.*

0. The same as Level 1 but also *Miller capacitances* C_{GD} and C_{GS} are neglected.

The model equations have been obtained as a time derivative of the model [8]. The less important terms were neglected. As Level 0 is a simplification of Level 1, only the latter will be described.

The condition of charge conservation is

$$Q_G + Q_S + Q_D + Q_B = 0\,. \tag{3.1}$$

The indices G, S, D and B correspond to gate, source, drain and substrate nodes. Assuming a constant substrate potential $V_B' = 0$ and making a time derivative,

$$Q_G' + Q_S' + Q_D' = 0 \tag{3.2}$$

is obtained. Denote

$$V_{\text{sat}} = V_G - V_S - V_T \tag{3.3}$$

where V_{sat} is the saturation voltage. According to its value, the model consists of three regions:

1. Off-region. $V_{\text{sat}} < 0$ and the channel D–S is closed. Then

$$I_{DS} = 0\,. \tag{3.4}$$

and as $V'_B = 0$, it follows that

$$Q'_G = C_{\text{ox}} V'_{GB} \doteq C_{\text{ox}} V'_G \tag{3.5}$$

where C_{ox} is the oxide capacitance. In this region, $Q'_D = Q'_S = 0$.

2. Saturation region. $0 \leqq V_{\text{sat}} \leqq V_{DS}$. The channel current I_{DS} is purely quadratic

$$I_{DS} = \frac{\beta}{2} V_{\text{sat}}^2 \tag{3.6}$$

while the charge changes as

$$Q'_G = \tfrac{2}{3} C_{\text{ox}} V'_{GS}\,. \tag{3.7}$$

Assuming that the channel charge is divided into 60% source charge and 40% drain charge and making a time derivative,

$$Q'_S = -\tfrac{2}{5} C_{\text{ox}} V'_{GS}\,, \quad Q'_D = -\tfrac{4}{15} C_{\text{ox}} V'_{GS} \tag{3.8}$$

is obtained. The corresponding transcapacitances are not symmetrical:

$$\begin{aligned} C_{GS} &= \tfrac{10}{15} C_{\text{ox}}\,, & C_{SG} &= \tfrac{6}{15} C_{\text{ox}}\,, \\ C_{GD} &= 0\,, & C_{DG} &= \tfrac{4}{15} C_{\text{ox}}\,, \\ C_{DS} &= -\tfrac{4}{15} C_{\text{ox}}\,, & C_{SD} &= 0\,. \end{aligned} \tag{3.9}$$

As the charge derivatives are sufficient for the simulation, the transcapacitances are used for illustration only, or for comparison with measured data [8].

3. Triode region. $V_{\text{sat}} > V_{DS}$. The channel current is

$$I_{DS} = \beta V_{DS}(V_{\text{sat}} - \tfrac{1}{2} V_{DS}) \tag{3.10}$$

while the gate charge becomes

$$Q'_G = \tfrac{1}{2} C_{\text{ox}}(V'_{GS} + V'_{GD})\,. \tag{3.11}$$

For 50/50 partition of the channel charge

$$Q'_S = -\tfrac{1}{2} C_{\text{ox}} V'_{GS}\,, \quad Q'_D = -\tfrac{1}{2} C_{\text{ox}} V'_{GD}\,, \tag{3.12}$$

just simple capacitances illustrate the model in this region: $C_{GD} = C_{GS} = \frac{1}{2} C_{\text{ox}}$.

4. NO-MATRIX APPROACH

For substitution of the MOS model into the kth circuit nonlinearity $G_k(V)$ and into the equation (2.7), the charge change contributions $\sum C_{kj}V_j'$ are necessary to be calculated. The minimal number of table scans is the matter of the optimal ordering of the substitution process. The cue is the form in which the contributions of each transistor to the right hand side of (2.7) are expressed.

Using the method NMSS, the channel currents I_{DS} are added to the right hand sides of Dth and Sth equations:

$$G_D(V) = \ldots + I_{DS}\,, \tag{4.1}$$
$$G_S(V) = \ldots - I_{DS}\,,$$

while the charge changes, as e.g. Q_G', correspond to the contributions to their right hand sides. For Q_G' it is the Gth equation:

$$\sum_{\substack{j=1 \\ j \neq G}}^{N} C_{Gj}V_j' = \ldots + Q_G'\,, \tag{4.2}$$

Notice that the indices G, D, S are *local* for each transistor. Thus two different transistors the gates of which are connected to different nodes have different values of G.

It is obvious that *no matrices are necessary to be stored*, and also no elements of the capacitance matrix, except of its diagonal. This is the *No Matrix method* which saves computer memory which is increasing only linearly with the number of MOS transistors. Also no multiple pointers are necessary (as it is common for sparse matrices).

Now while not using matrices, there still exist *two ways how to organize the sequence of computations*:

- Either for all nodes $k = 1, 2, \ldots, N$ the table of transistors is to be scanned if $G = k$, $D = k$, or $S = k$ holds true. This suggests the contribution to the kth node. A decision must follow in which region the transistor works, and after that the channel current I_{DS} and the charge changes Q_S' are calculated and added to the kth right hand side.
- Or all transistors are scanned in a *single sweep*, and partial sums of the Gth, Dth, and Sth right hand sides of (2.7) are updated by each transistor. Notice that *three* equations are updated at a time. This is the second principle of the method NMSS.

Both approaches have their advantages. For NMSS, the minimum of scans is the premium compared with better flexibility of the above described selective search. The quadratic terms for I_{DS} are calculated only once for each transistor, and they are immediately included into the equations for D and S.

There is a disadvantage of NMSS method (which can be overcome by parallel processing) that *only the Gauss-Jacobi* method can be applied, and not Gauss-

Seidel. The reason is *that all values of V' are known at the last instant*, that is after the addition of the contribution of the last transistor.

Let us summarize what is necessary to be stored for the substitution into the circuit equations and for the integration:

- Vector composed of C_{kk}, $k = 1, \ldots, n$.
- Vector of the right hand side of the equation (2.7). After addition of contributions of all transistors and after division by C_{kk}, this vector will be changed into the left hand side V'.
- Table of parameters of transistors containing G, D, S, β, V and static capacitances for all regions. This table remains constant during the whole simulation.
- Old values of V' from past time steps. They serve for integration formulas and for the estimation of V'_n.

5. THE INTEGRATION ALGORITHM

For the simulator WATTIME, the highest possible order of the integration method has been chosen which still does not involve any solution of equations for coefficients of variable-step formulas.

Such requirements are satisfied by a *variable two-step method* with a 2nd order predictor and a 3rd order corrector. There is another reason why this low order is sufficient: Further increase of order usually does not bring improvement neither in accuracy nor in convergence, and it may even cause stability problems.

Denote h_n as the length of the previous time step and λh of the next one. The variable order two-step method uses the values $V'(t + \lambda h)$, $V'(t)$ and $V'(t - h)$, denoted as V'_n, V'_{n-1}, and V'_{n-2}.

The second order variable-step Adams-Bashforth predictor formula is

$$V_n^{\mathrm{P}} = V_{n-1} + h_n\left[\left(1 + \frac{\lambda}{2}\right)V'_{n-1} - \frac{\lambda}{2}V'_{n-2}\right]. \tag{5.1}$$

while the third order variable-step Adams-Moulton corrector formula is

$$V_n = V_{n-1} + h_n\left[\left(\frac{1}{2} - \frac{\lambda}{6(1+\lambda)}\right)V'_n + \left(\frac{1}{2} - \frac{\lambda}{6}\right)V'_{n-1} + \frac{\lambda^2}{6(1+\lambda)}V'_{n-2}\right] \tag{5.2}$$

The integration is *easy to start*. For the first time step, the time point t_{n-2} is supposed to be in the far past. Therefore $h_n \to \infty$, and as $\lambda = h_{n+1}/h_n$, for the first step $\lambda = 0$ is set and the Adams-Bashforth formula degenerates into Euler forward

$$V_n = V_{n-1} + h_n V'_{n-1}, \tag{5.3}$$

and also the Adams-Moulton formula degenerates into the trapezoidal formula

$$V_n = V_{n-1} + h_n(\tfrac{1}{2}V'_n + \tfrac{1}{2}V'_{n-1}). \tag{5.4}$$

Thus for the first step the same formulas as for next steps can be used. The only difference is to set $\lambda = 0$ which excludes the terms with the unknown V'_{n-2}.

For the integration algorithm used in WATTIME, the corrector is expressed as a difference (*corrector* − *predictor*) which saves mathematical operations:

$$\Delta V_n = V_n - V_n^{\mathrm{P}} = h_n(\alpha V'_n + \beta V'_{n-1} + \gamma V'_{n-2}) \tag{5.5}$$

where $\alpha = (\frac{1}{2} - (\lambda/6(1+\lambda)))$, $\beta = -(\frac{1}{2} + (\lambda/3))$, $\gamma = \lambda\alpha$.

Notice that

$$\alpha + \beta + \gamma = 0 \tag{5.6}$$

which is a general property of the corrector-predictor difference. This can be proven when both predictor and corrector approximate a straight line where $V'_n = V'_{n-1} = V'_{n-2}$. As $\Delta V_n = 0$, and the derivative is not identically zero, the sum must be. As α, β, and γ are functions of λ only, the property will hold true in general case.

This property enables to formulate (5.5) as $\alpha(V'_n - V'_{n-2}) + \beta(V'_n - V'_{n-2})$ which saves one vector-by-scalar multiplication. The final formulas used in WATTIME first calculate the modified coefficients

$$p_1 = \frac{\lambda}{2} h_n, \quad p_2 = p_1 + h_n \tag{5.7}$$

which are substituted into the predictor formula

$$V_n^{\mathrm{P}} = V_{n-1} + p_2 V'_{n-1} - p_1 V'_{n-2}. \tag{5.8}$$

Next the coefficients of the corrector increment are calculated using the modified predictor coefficients

$$c_1 = \frac{p_1}{2} + \frac{p_2}{6}, \quad c_2 = \frac{c_1}{1+\lambda} \tag{5.9}$$

and they are substituted into the corrector-predictor increment formulas

$$V_n = V_n^{\mathrm{P}} + c_2(V'_n - V'_{n-2}) + c_1(V'_{n-2} - V'_{n-1}). \tag{5.10}$$

The stability of both predictor and corrector decreases with growing λ. This can be prevented either by truncation $\lambda \leqq 2$ or (and better) by control of the step size:

1. In each time step calculate the predictor value V_n^{P} according to the formulas (5.7) and (5.8).

2. Substitute V_n^{P} (and the guess of V'_n) into the circuit equations (2.7). Make a single iteration of (2.7) and calculate the new derivative V'_n.

3. If $V(t)$ is *almost linear*, set $V'_{n-1} = V'_n$, $V'_{n-2} = V'_{n-1}$. Prolong the step and go to 1.

If $V(t)$ is *very nonlinear*, contract the step and go to 1.

4. Calculate the corrector V_n using formulas (5.9) and (5.10).

5. If the corrector-predictor difference is too big, contract the step and go to 4.

6. Substitute V_n (and its last calculated derivative as a guess) into the circuit equations (2.7). Make a single iteration of (2.7) and calculate V'_n.

7. If $V(t)$ is *very nonlinear* (a rare case at this point), contract the step, and go to 4.

8. Set $V'_{n-1} = V_n$, $V'_{n-2} = V_{n-1}$, and go to 1.

It can be seen from this simplified algorithm that it is optimized in such a way that extra calls of the circuit model are allowed only as an emergency.

The prolongation factor λ is estimated from the *relative voltage tolerance* ε and the maximal voltage increment

$$\lambda = \frac{\varepsilon}{h_n \max\limits_i |V'_i(t)|}, \quad i = 1, \ldots, N. \tag{5.11}$$

During the simulation, the relative voltage tolerance ε changes dynamically within the limits

$$\varepsilon_{\min} \leqq \varepsilon \leqq \varepsilon_{\max}. \tag{5.12}$$

The value $\varepsilon_{\min}$ is given by the number of decimal places of the output voltage. The value $\varepsilon_{\max}$ is given by the maximal voltage swing $V_{\max} - V_{\min}$ and accuracy δ:

$$\varepsilon_{\max} = \delta(V_{\max} - V_{\min}). \tag{5.13}$$

For well designed MOS circuits, the accuracy $\delta = 5\%$ is in balance with the accuracy of models of MOS transistors. Even $\delta = 100\%$ has given stable results in all examples. If $\delta < 1\%$ is specified, the number of model evaluations starts to increase significantly. However, specifying a very low δ (say 0·01%) may help to distinguish the integration error from model inaccuracies in case of small test circuits.

The value of ε is controlled by the relative second difference μ

$$\mu = \max_i \frac{|V'_i(t + \lambda h) - V'_i(t)|}{\frac{1}{2}|V'_i(t + \lambda h) + V'_i(t)|}, \quad i = 1, \ldots, N, \tag{5.14}$$

According to μ, the linearity of $V(t)$ is estimated:

- almost linear $\mu \leqq 0{\cdot}25$
- acceptable $0{\cdot}25 \leqq \mu \leqq 1$
- very nonlinear $\mu > 1$

For average circuits, *in most time steps only the predictor is calculated.* The corrector is necessary in about 20% of steps, and about 10% of model calls are discarded due to step contraction.

6. EVENT-DRIVEN ANALOG SIMULATION

As the external signal sources are voltages, the system (2.2) consists of N equations for $M < N$ unknowns. The remaining $N - M$ voltages are known constants.

The usual elimination of constant variable V'_k would transfer it to the right hand

side, and the kth equation would not be built. However, the method NMSS does not make any difference between known and unknown variables. The first reason is that in ith equation *all variables* V_j', $j \neq i$ *are at the right hand side* (thus the system is built), and no transferring is necessary.

The second reason is that it is *expensive to discriminate between known and unknown voltages* when scanning the table of transistors. In WATTIME the contributions of transistors are added even to the equations for *known* voltages. These equations are only not used. Also during the simulation a known voltage may become unknown and vice versa.

WATTIME uses an *event driven algorithm* for marking known and unknown signals according to their *status*. The status may be *active external* (V known, $V' \neq 0$), *active internal* (V unknown), *latent external* (V known and constant),

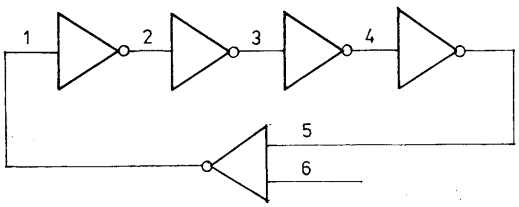

Fig. 1. Schematic diagram of a CMOS ring oscillator.

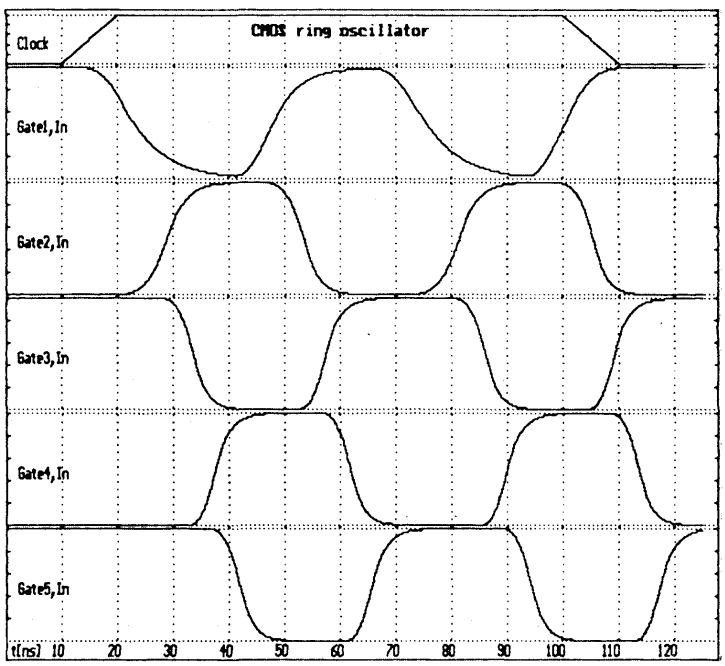

Fig. 2. Computer simulation of a CMOS ring oscillator.

latent internal (V does not change but it may) *and conditionally latent* (V does not change but there is a suspicion that it will).

After the change of any input signal the time of its expected end is put in stack and the node is marked as *active external*. Its fan-outs and their time derivatives are examined and their status changed. Thus the integration formulas work only with a linked list of equations belonging to nodes *active internal*.

WATTIME can mix analog and digital outputs according to the user's specifications from L–X–H through multilevel logic to analog signals. However, the internal calculation of all signals is analog.

7. EXAMPLE

A CMOS ring oscillator has been simulated by WATTIME (Turbo-Pascal 5.0, PC–XT 8 MHz, V20 processor, 20 seconds). The schematic diagram is in Fig. 1 and the hard copy of the computer screen is in Fig. 2. The circuit is essentially unstable while its simulation is still stable. Notice the wave spreading in the feed-back loop, each waveform being shifted by the gate delay. The comparison between the simulated and measured oscillation frequency of such circuit is a very efficient means for tuning the device models.

(Received August 1, 1989.)

REFERENCES

[1] B. R. Chawla, H. K. Gummel, and P. Kozak: MOTIS — An MOS timing simulator. IEEE Trans. Circuits and Systems *CAS-22* (1975), 901—909.

[2] A. Vladimirescu and Liu: The simulation of MOS integrated circuits using SPICE 2. Research Report UCB/ERL M 80/7, University of California Berkeley, 1980.

[3] W. T. Weeks et al.: Algorithms for ASTAP — A network analysis program. IEEE Trans. Circuit Theory *CT-20* (1973), 628—634.

[4] A. R. Newton and A. L. Sangiovanni-Vincentelli: Relaxation-based electrical simulation. IEEE Trans. Computer-Aided Design *CAD-3* (1984), 308—331.

[5] J. White and A. L. Sangiovanni-Vincentelli: RELAX 2: A new waveform relaxation approach to the analysis of LSI MOS circuit. Proc. IEEE Internat. Conf. Circuits Systems 1983, 756 to 759.

[6] J. Kleckner, R. Saleh and A. R. Newton: Electrical consistency is schematic simulation. Proc. IEEE Internat. Conf. Circuits Comput., October 1982, 30—34.

[7] P. Pavlík and J. Vlach: WATTIME — Waterloo timing simulator of MOS circuits. Proc. IEEE Internat. Conf. Circuits Systems 1986, 751—754.

[8] B. J. Sheu et al.: A compact IGFET charge model. IEEE Trans. Circuits and Systems *CAS-31* (1984), 745—748.

Ing. Petr Pavlík, CSc., Středisko výpočetní techniky ČSAV (General Computing Center — Czechoslovak Academy of Sciences), Pod vodárenskou věží 2, 182 07 Praha 8. Czechoslovakia.

Prof. Jiří Vlach, Department of Electrical Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1. Canada.