A SIMULATION PROGRAM EMPHASIZED ON DC ANALYSIS OF VLSI CIRCUITS: SAMOC

Ying-Wei Jan and Janusz A. Starzyk

School of Electrical Engineering and Computer Science
Ohio University, Athens, OH 45701
ying-wei@bobcat.ent.ohiou.edu

ABSTRACT

This paper presents a simulation technique and a computer program for fast DC analysis of MOS transistor based VLSI circuits. Time domain analysis of a VLSI circuit is evaluated by a piecewise constant waveform approximation. This approximation is realized by repeatedly applying the developed DC analysis engine with initial conditions. All proposed methods such as device modeling, circuit partitioning and event-driven simulation were implemented and combined with such well known algorithms as modified nodal analysis (MNA) [1], Katzenelson algorithm [2] and Gaussian elimination in the form of a circuit simulation program: SAMOC. Time domain waveforms and benchmark circuit simulation results comparison of SPICE and SAMOC are presented.

I. INTRODUCTION

The amount of current, which passes through each device, and the magnitude of voltage at each interconnection node can characterize the behavior of an electronic circuit. Kirchhoff's laws (KCL, KVL) associated with the interconnection of electrical devices are applied to formulate the circuit equations. Electrical device models link the electronic design world with the mathematical representation. Development of these models must either follow established physical theories or be based on the experimental measurements of the electrical device. Circuit analysis is therefore achieved by formulating and solving the circuit equations.

Using digital computer programming techniques, the formulating and solving of circuit equations can be automated. Computer programs designed to analyze circuit behavior are called circuit simulators. With the help of circuit simulators, designers can predict circuit behavior, and gather statistical results to optimize their designs. Among circuit simulators, SPICE [3] and ASTAP [6], which employ precise device models and sophisticated numerical integration techniques, are used for accuracy oriented circuit simulation. However, for analyzing VLSI circuits, these accuracy oriented circuit simulators consume too much computation resource and became impractical.

For many circuit categories, precise behavior estimation of each device may not be necessary. Many simulators were designed to take advantage of a low cost of lower precision analysis and trade accuracy for analyzing speed. The adopted strategies are employing simplified device models, approximation based on linear techniques or trying to exploit the circuit latency. Among them, SPECS [4], employs piecewise constant device model, which represents a resistive device with segments of independent current source to reduce computational effort of the RC integration. In addition to special device models, SPECS also employs event-driven techniques to exploit the circuit latency.

Contemporary VLSI circuit systems are capable of executing fast and complex arithmetic operations, performing signal processing, and extracting information from databases. Most of the processed data are presented and stored in a form of DC voltages, charges or currents. For this reason, there is a necessity to develop a circuit simulator, which can directly determine the DC solution of a VLSI circuit according to the excitation and previous state of the analyzed circuit. In this kind of analysis, the output time domain waveform is presented using piecewise constant approximation as shown in Fig. 1.

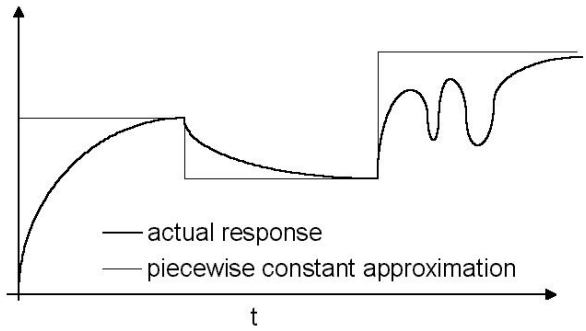

Fig.1 Piecewise constant waveform approximation

SAMOC (Switched-capacitor Analysis of MOs Circuit) is a circuit simulator developed for piecewise constant simulation of a VLSI circuit. SAMOC employs piecewise device modeling and contains a numerical analysis engine, circuit partitioning, and event driven mechanism. Section II describes the construction elements of SAMOC. A few simulation examples are shown in Section III as a functional verification of

SAMOC simulator. Section IV presents a benchmark circuit simulation result comparison between SPICE3f4 and SAMOC. Section V is the conclusion of this paper.

II. ORGANIZATION OF SAMOC PROGRAM

The circuit equation formulation algorithm employed by SAMOC is based on the modified nodal analysis (MNA). MNA is easy for computer-programming implementation, can handle controlled sources, open circuits and short circuits. Therefore, MNA is good for device modeling.

A. Piecewise Linear Device Model

There are two kinds of semiconductor devices supported in SAMOC: a MOS transistor and an ideal diode. Both are modeled by piecewise methods. The piecewise linear MOS transistor model is a resistive model. That is, the values of I_{ds} are functions of voltage V_{ds} and V_{gs} as shown in Fig. 2. There are 3 regions separated by values of V_{ds} and V_{gs} as shown in Fig. 3. In the cutoff region, a large resistor models the MOS transistor. In the linear region, a small resistor models the MOS transistor. In the saturated region, the MOS transistor is modeled by a large resistor and a voltage controlled current source (transconductance).

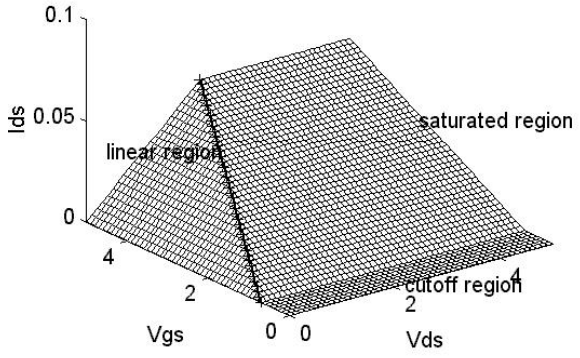

Fig. 2 Piecewise linear resistive MOS transistor model in SAMOC.

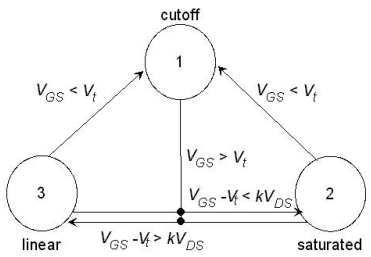

Fig. 3 Dynamic state transition diagram of a MOS transistor.

SAMOC also contains a piecewise linear model for circuits with ideal diodes. The diode model contains 3 working regions shown in Fig. 4 - independent voltage sources (regions 2 and 3), and open circuit (region 1). Fig. 5 illustrates the state transition diagram of the piecewise linear diode model. In the open circuit region (1), the transition of state depends on diode voltage. In the turned on region (2) and the breakdown region (3), the transition of state depends on diode current.

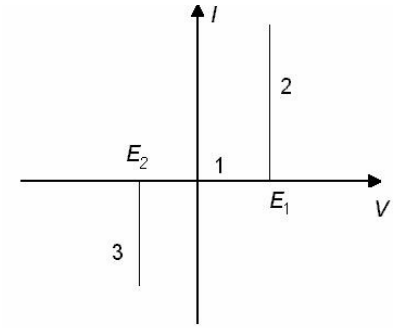

Fig. 4 Piecewise linear ideal diode model.

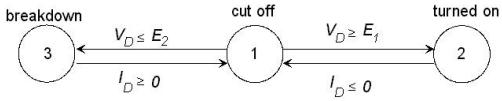

Fig. 5 Dynamic state transition diagram of an ideal diode.

By using piecewise linear device models within the employed MNA circuit formulation algorithm, piecewise linear equations will be created for the analyzed circuit. Katzenelson algorithm is used to determine the solution of the resulting piecewise linear system. Computational complexity of applying Katzenelson algorithm is proportional to the number of piecewise linear devices.

B. Circuit Partitioning

Circuit partitioning in SAMOC is used to separate a circuit into several blocks, and analyze each block independently from the rest of the analyzed circuit. Circuit simulation can be achieved by analyzing smaller blocks rather than the whole circuit. Since each block contains fewer devices, the Katzenelson algorithm converges with less iteration. In addition, the lower memory requirement and smaller computational effort for processing and storing circuit equations makes resource limited computer system capable of analyzing even big circuits.

Since SAMOC employs KCL and MNA, circuit partitioning can be realized by splitting nodes incident to voltage sources and ground. Fig. 6 illustrates an example of circuit partitioning by splitting nodes.

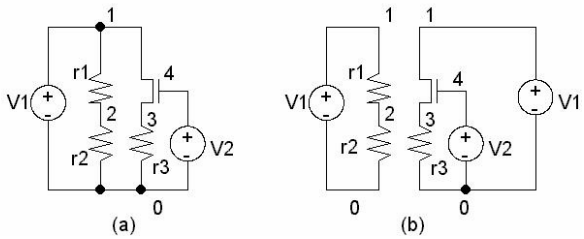

Fig.6 Circuit partitioning by splitting nodes.

In addition to splitting nodes, circuit partitioning can also be realized by splitting devices. In controlled sources (VCVS, VCCS, CCVS and CCCS), the controlling nodes can belong different circuit blocks than the controlled nodes. Fig. 7 illustrates 2 examples of circuit partitioning by splitting devices. The device shown in Fig. 7(a) is a voltage-controlled source (VCVS or VCCS). This device can be partitioned into 3 different blocks. Likewise, the device shown in Fig. 7(b) is a current-controlled source (CCVS or CCCS) which can be partitioned into 2 blocks. Same partitioning via device splitting scheme can be applied to MOS transistors and voltage controlled switches. A MOS transistor can be partitioned into 2 blocks, since there is no DC current flowing from gate to source or drain.

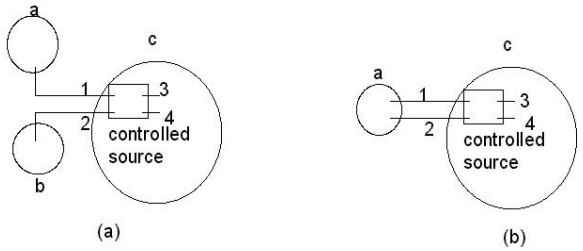

Fig. 7 Circuit partitioning by splitting devices.

C. Block-Signal Diagram and Topological Depth of a Circuit Block

Although each circuit block can be analyzed independently from other blocks, there is a posterior-prior relationship between the analyzed blocks. Circuit partitioning by splitting devices causes this relationship between blocks. For example, as shown in Fig. 7 (a), both blocks "a" and "b" should be analyzed before block "c". SAMOC employs a directed graph called **block-signal diagram** to represent and store the posterior-prior information between blocks.

Fig. 8 illustrates a block-signal diagram of an analyzed circuit. The block-signal diagram is constructed by 14 circuit blocks and 3 time varying devices. Blocks are represented by circles that are named by letters "A" to "N". The arrows represent both posterior-prior relationships and propagated signals between blocks. An easy approach to analyze the block-signal diagram is to assume all signals between blocks have the same delay time and assign a topological depth to each block. The smaller the topological depth, the earlier the time varying signals propagate to the block. That means the block should be analyzed prior to the blocks having larger topological depth. By applying this method, the time step should be small enough to capture variation of the signals in internal loops.

The rules of setting topological depths of circuit blocks are:

1. Find blocks without any prior block and set their topological depth to "0" (such as block "A" in Fig. 8). Blocks with topological depth "0" are considered to be static and only need to be analyzed once.
2. Set topological depth of blocks with time-varying excitations to "1". In Fig. 8, blocks "B", "C" and "D" have topological depth equal to "1".
3. Other blocks, which accept signals from blocks "B", "C" and "D" have their topological depths equal to the shortest distance from blocks "B", "C" and "D". In Fig. 8, blocks "E" to "L" have their topological depths assigned by this method.
4. The rest of circuit blocks, "M" and "N" in Fig. 8, have topological depths equal to "1". Although the time varying signals will never propagate to these blocks, they can possibly influence the simulated circuit. Their topological depths are set to 1 in order to make them influence other blocks.

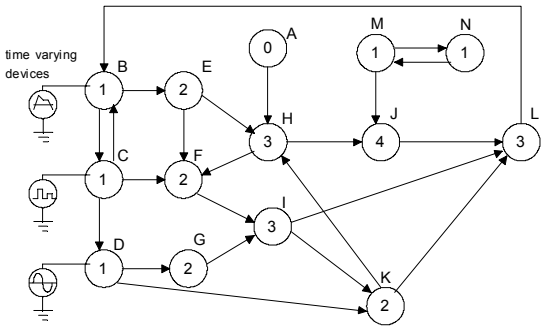

Fig. 8 A block-signal diagram contains 14 circuit blocks and 3 time varying devices.

After the topological depth of all blocks is set, a simulation queue can be created to manage the order of blocks for analysis. Fig. 9 illustrates the simulation queue created by the block-signal diagram shown in Fig. 8. After the blocks with the highest topological depth are analyzed, a solution of the simulated circuit is recorded.

D. Event-driven simulation

Event-driven simulation can exploit the circuit latency and save computational effort. Event-driven simulation techniques usually assume that a portion of the analyzed circuit remains static or only has an insignificant change

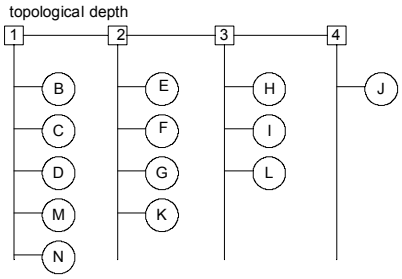

Fig. 9 A simulation queue.

during a certain period of time. Skipping the analysis of the static portion can therefore save a considerable amount of the computational effort. In a VLSI circuit simulation, event-driven simulation plays a very important role, because the output of interest is created in an instance and depends only on a small portion of the whole circuit.

SAMOC employs independent block analysis method, although there is a simulation queue to manage the block analysis order. In the simulation queue of SAMOC, each block switches from active to inactive (latent) after its analysis. Each analyzed block fires at all its posterior blocks, if the analyzed block's solution vector differs from the one before analysis. The fired inactive block may not necessary become activated, so that more latency of the analyzed circuit can be exploited. The fired-activated mechanism in SAMOC between blocks depends on the device(s), which cause(s) the posterior-prior relationship between blocks. If two blocks are linked by a controlled source (VCVS, VCCS, CCVS or CCCS), then the fired inactive block turns active unconditionally. If a voltage-controlled switch links two blocks, then the fired inactive block turns active only when switching status of the switch is altered.

If a MOS transistor links two blocks, then the firing prior block must contain the gate of the MOS transistor, and the fired posterior block must contain the source and drain of the MOS transistor. Terminal voltages V_{gs} and V_{ds} determine the MOS model in SAMOC. For the MOS transistor, which causes the posterior-prior relationship, its V_{gs} is changed and its V_{ds} remains the same. That is, $\Delta V_{gs} \neq 0V$ and $\Delta V_{ds} = 0V$. Fig. 10 illustrates all possible changes of ΔV_{gs}. The dots represent the previous value of V_{gs} and the arrows point to the new V_{gs}.

In cases 1,2,6 and 7 shown in Fig. 10, the changes of V_{gs} do not cause a change of the MOS's working region, so the fired block stays inactive. In cases 3 and 5, the ΔV_{gs} causes the MOS transistor to change its working region, so the fired block becomes activated. If the MOS transistor is in saturated region (case 4), then the fired block turns active unconditionally. In saturated region, a controlled source VCCS is used to describe the MOS behavior.

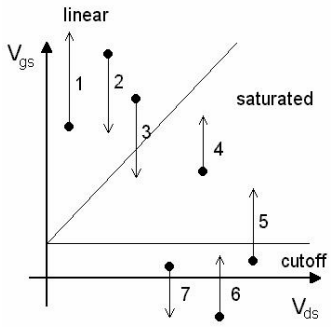

Fig. 10 An event caused by ΔV_{gs}.

During circuit initialization, all blocks are analyzed in order managed by the simulation queue to determine the working region of each piecewise linear device. All analyzed blocks turn inactive. During time domain analysis, the time varying excitations activate blocks with topological depth 1. The activated blocks are analyzed. SAMOC checks the new solution vectors. If the solution vectors changed, then their posterior blocks are fired at. After all blocks with topological depth 1 are processed, SAMOC begins to determine if the blocks with topological depth 2 shift from fired to active, and analyzes active blocks with topological depth 2. The identical procedure is performed at the next position of the simulation queue. Outputs of interest are recorded after one routine in the simulation queue is finished. SAMOC users determine the time step between two analysis instances. At each analysis instance, a solution vector is obtained. The output waveforms are constructed by linking the series of solution vectors.

III. FUNCTIONAL VERIFICATION OF SAMOC SIMULATION

SAMOC containing the algorithms presented in Section II is implemented in a form of C++ programming language. Examples of SAMOC simulation are presented in this section for functional verification of SAMOC simulation.

A. Waveform Comparison

To test and evaluate the piecewise resistive MOS model presented in Section II.A, an analog CMOS circuit was used to compare the SAMOC simulation with SPICE simulation. The circuit illustrated in Fig. 11 is a wide-sense transconductance amplifier. The circuit contains 9 MOS transistors. SAMOC partitioned it into 3 blocks, and the highest block topological depth is 2. Fig. 12 shows the input signal waveforms V+, V- and the output

V_o evaluated by PSPICE and SAMOC. The output waveform of SAMOC piecewise constant approximation is similar to PSPICE simulation results.

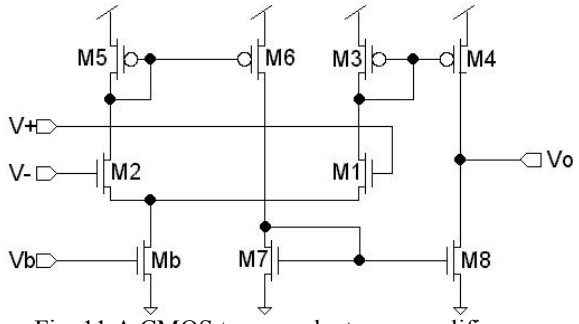

Fig. 11 A CMOS transconductance amplifier.

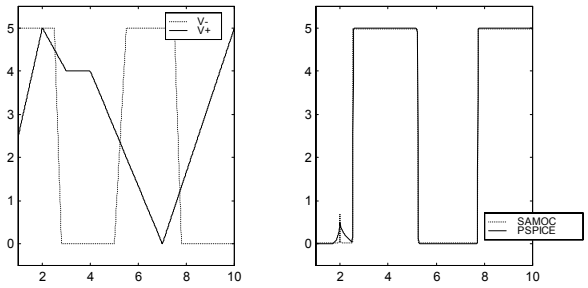

Fig. 12 The input and output signal waveforms.

B. Event-Driven Simulation Evaluation

In order to evaluate the efficiency improvement of the event-driven simulation, a 4 bit ripple counter was used to do the test. The counter shown in Fig. 13 is constructed by 4 NAND gate based master-slave JK flip-flops. The organiztion of the flip-flop is illustrated in Fig. 14. Each flip-flop contains 8 CMOS NAND gates and each NAND gate is constructed by 2 PMOS and 2 NMOS transistors. SAMOC supports hierachical **.subckt** netlist format useful for top-down design. The counter is simulated on the transistor level. The 4-bit counter contains 146 devices (2 voltage sources and 144 MOS transistors), 77 nodes and SAMOC partitioned it into 32 blocks. The sole time varying source is the clock signal. The highest block topological depth is 12.

Fig. 15 illustrates the SAMOC simulation results. There are 400 simulation instances in the presented results. The waveforms of generated signals (clock, A1, A2, A3 and A4) validate the SAMOC simulation. The sixth plot in Fig. 15 is the number of analyzed blocks as a function of the simulation instance. From this sixth plot, it is clear that during timing simulation, most blocks remain latent and the number of analyzed blocks increases when the output values of signals A1, A2, A3 or A4 change. Quantitatively, 961 blocks were triggered and analyzed during the 400 simulation instances, which means, on average, 2.4 of 32 blocks were analyzed at each simulation instance. The event-driven simulation mechanism saved about 92.49% of block analysis comparing to the simulation without event-driven approach.

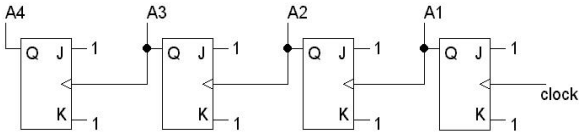

Fig. 13 A JK flip-flop based 4 bit ripple counter.

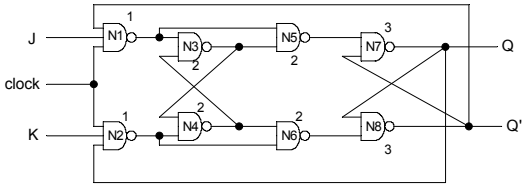

Fig. 14 A NAND gate based master-slave JK flip-flop.

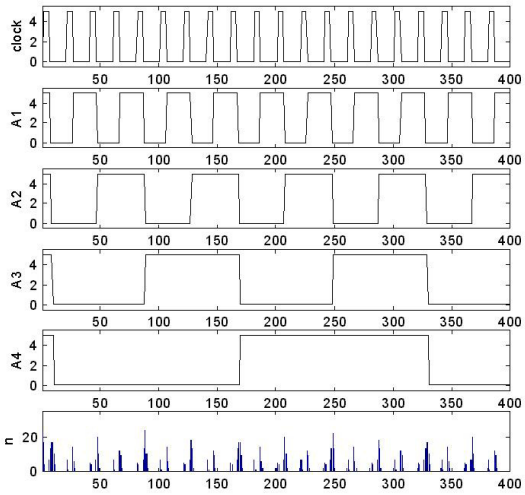

Fig. 15 The SAMOC simulation of the 4 bit ripple counter.

VI. BENCHMARK CIRCUIT SIMULATION

A number of benchmark circuits such as BJT and MOS circuits in SPICE netlist format can be downloaded from the web-site [7]. All reported in this section benchmark circuit simulations were performed in time domain and were completed on an IBM PC compatible computer with a Cyrix® 6x86MX PR200 CPU running at 166 MHz (66 x 2.5). The computer was equipped with 512 MB of EDO (extended data out) DRAM. Two operating systems, MS® Windows NT 4.0 with service pack 4 and Redhat® Linux 5.1 with kernel

2.0.34, were used. SPICE simulation was performed in Linux using Berkeley SPICE3f5. SAMOC simulation of the same set of benchmark circuits was performed in the same computer with the same hardware configuration as for the SPICE simulation while the operating system was switched from Linux to MS Windows NT.

The required simulation time is presented in Table 1. For some smaller circuits, SAMOC took more CPU time than SPICE. SAMOC requires more circuit initialization time by employing massive interconnected data structure, circuit partitioning and simulation queue maintenance. For larger circuits, SAMOC uses much less time than SPICE. Note that SPICE simulation of circuit Sqrt and Ram2k were not finished. Computer crashed because of not enough memory after the listed time.

circuit	MOSFETs	Capacitors	SPICE (sec.)	SAMOC (sec.)
ab_integ	31	22	1.38	3
Ab_opamp	31	24	1.69	10
Cram	60	42	15.36	9
Mux8	64	29	226.19	73
Toronto	58	33	4.72	5
Counter	220	0	28.43	6
B330	330	0	Failed	96
Voter25	74	0	4580	1
Sqrt	1118	1022	25314.3	1116
Ram2k	13880	156	23269.1	1869

Table 1. Required time of benchmark circuit simulation.

VII. CONCLUSION

A set of algorithms such as device modeling, circuit partitioning, event-driven simulation and piecewise constant approximation for fast time domain transistor-level simulation of a MOS circuit is presented in this paper and is implemented in the computer program called SAMOC for evaluation and research. The waveforms of SAMOC analysis of analog CMOS circuit were obtained and compared to SPICE simulation to evaluate the piecewise linear models. Event-driven simulation designed for piecewise device models and circuit partitioning were illustrated by analyzing a 4-bit ripple counter. In the presented simulation example, the computational efficiency improvement can be up to 92%. That is, on average, only 8% of blocks are required to be analyzed during each simulation instance.

Because of the simplified device model, circuit partitioning, event-driven simulation and piecewise constant waveform approximation, SAMOC uses less computer resource to analyze big circuits than traditional methods. A set of benchmark circuit simulations was performed in the same computer, and the comparison with Spice program was presented.

By using SAMOC, a circuit designer can determine functional design flaws in a short time before exhaustive precise simulation is performed, therefore, a reduction of the circuit design time can be achieved.

The most needed feature in SAMOC is delay estimation for intra- and inter-blocks. Delay estimation has been one of the most significant research topics of circuit simulation techniques. A delay estimation can be easily added into SAMOC without changing existing software architecture. Presented in Section II, the event queue setting is based on the topological depth that corresponds to setting based on a unit delay time. This setting can be replaced by a setting based on a real delay time needed for delay estimation. Hence, SAMOC is capable of handling the signal-racing problem. Furthermore, the block delay estimation can aid in developing adaptive simulation time step selection, which will improve both precision and efficiency of SAMOC simulation. Moreover, the separated calculation modules, such as independent block analysis and delay estimation, can benefit from the state-of-art multiprocessor computer systems.

REFERENCES

[1] C. W. Ho, A. E. Ruehli, and P. A, Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems,* vol. CAS-25, pp. 504-509, June 1975.

[2] J. Katzenelson, "An algorithm for solving nonlinear resistor networks," *The Bell System Technical Journal,* vol. 44 pp. 1605-1620, October 1965.

[3] L. W. Nagel, **SPICE2: A computer program to simulate semiconductor circuits**, Electron. Res. Lab., University of California, Berkeley, Rep., ERL-M520, May 1975.

[4] C. Visweswariah and R. A. Rohrer, "Piecewise approximation circuit simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. CAD-10(7), pp. 861-870, July 1991.

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

[6] W. T. Weeks *et al.,* "Algorithms for ASTAP -- A network analysis program," *IEEE Transactions on Circuit Theory,* vol. CT-20, pp. 628-634, November 1973.

[7] The URL site of the benchmark circuits is http://www.cbl.ncsu.edu/CBL_Docs/csim90.html