ACCELERATING RELAXATION ALGORITHMS FOR CIRCUIT SIMULATION USING WAVEFORM NEWTON, ITERATIVE STEP SIZE REFINEMENT, AND PARALLEL TECHNIQUES

Jacob White, Resve Saleh, A. Sangiovanni-Vincentelli, A. R. Newton

Dept. of Electrical Engineering and Computer Sciences

University of California, Berkeley

Abstract

Two techniques are considered for accelerating relaxation-based circuit simulators. First, a new algorithm that attempts to combine the advantages of the Waveform Relaxation algorithm and Iterated Timing Analysis relaxation algorithm for moderately coupled multirate systems is presented. The algorithm is based on the extension to function spaces of the relaxation-Newton methods popular for solving algebraic systems. This Waveform Relaxation-Newton(WRN) technique is combined with an iterative stepsize refinement scheme which improves the accuracy of the numerical integration as the relaxation iterations approach convergence. As a second approach to accelerating relaxation algorithms, two techniques for parallelizing the classical WR algorithm are described, one based on mixed Gauss-Seidel/Gauss-Jacobi techniques, and the other on pipelining the computation. Finally, a technique for parallelizing WRN is described.

1. INTRODUCTION

The implicit multistep integration algorithms used in general purpose circuit simulation programs, like SPICE2[1], have proved to be reliable, but are computationally expensive when applied to large systems. This is because each step of the numerical integration requires the implicit solution of a large nonlinear algebraic system. If the circuit simulation program is intended for the simulation of mostly MOS digital circuits, then it is possible to exploit the properties of these types of circuits to improve its efficiency. In particular, that MOS digital circuits can be broken into loosely or uni-directionally coupled subsystems can be exploited by iterative decomposition algorithms, and that the different nodes in an MOS digital circuits change at very different rates can be exploited by multirate integration algorithms.

A variety of algorithms have been applied to the simulation of MOS digital circuits that attempt to exploit its loosely coupled and multirate nature[2,3,4]. We will focus on three relaxation-based methods. First is the Iterated Timing Analysis (ITA) algorithm used in the SPLICE1 and SPLICE2[5] programs, the second is the Waveform Relaxation(WR) algorithm[6] used in the RELAX2[7] program, and the third is a new algorithm based on combining some of the advantages of of ITA and WR. The paper is organized as follows. Section 2 contains a brief description of the first two relaxation algorithms along with a short discussion of their relative merits. In Section 3 we present a Waveform Relaxation-Newton (WRN)[8,9,10] algorithm that uses an iterative refinement strategy for selecting the numerical integration timesteps, and present experimental results that demonstrate its relative advantages compared to ITA and WR. In Section 4 three parallel algorithms are presented, two based on the WR algorithm, for which experimental results will be presented, and one based on the WRN algorithm.

2. A BRIEF INTRODUCTION TO RELAXATION TECHNIQUES

Given mild assumptions, the differential equations that describe MOS digital circuits have the following general form[7],

$$\dot{q}(v(t)) = f(v(t), u(t)) \qquad x(0) = x_0 \tag{1}$$

where $v(t)$ is usually the vector of node voltages, $u(t)$ is a set of inputs, $f(v(t), u(t))$, is usually the vector of the sums of currents incident at each node and $q(v(t))$ is the vector of node charges.

If the Backward-Euler implicit integration method is applied to solving Eqn. (1), the following algebraic equation results

$$F(v(t+h)) = \tag{2}$$

$$q(v(t+h)) - q(v(t)) - h\, f(v(t+h), u(t+h)) = 0.$$

In standard circuit simulation programs like SPICE2, the algebraic system represented by $F(v(t+h))$, in Eqn. 2, is solved for $v(t+h)$ using a modified Newton-Raphson algorithm, each step of which requires the evaluation of part of F and part of its Jacobian matrix, followed by a matrix solution.

If Eqn. 2 is solved using a nonlinear relaxation-Newton algorithm[12], as in the ITA algorithm, each element of the $v(t+h)$ vector, $v_i(t+h)$, is updated using only the F_i equation, and the i^{th} diagonal term of the Jacobian matrix of F.

Another approach to performing relaxation decomposition is the Waveform Relaxation Algorithm (WR), in which the decomposition is performed before discretizing with a numerical integration method, i.e. the relaxation algorithm is applied directly to the differential equation system. Each element of the vector of waveforms $v(t)$, $v_i(t)$, is updated by using a numerical integration algorithm to solve only the i^{th} differential equation.

As mentioned in the introduction, one of the properties of MOS digital circuits that relaxation methods are intended to exploit is that the different state variables of the system change at very different rates. By solving the differential equations in a decomposed fashion, the WR algorithm *intrinsically* allows different timesteps to be used for each differential equation in the system, so that each can use the largest timestep that accurately reflects the behavior of its associated state variable.

The ITA algorithm as implemented in the SPLICE1.6 and SPLICE2 programs also attempts to exploit this multirate property, by using an event-driven

selective trace algorithm[5]. Only the node voltages which are changing are updated at each time point. The remaining node voltages are updated using their respective values at the previous timepoint. In this way, the selective trace ITA algorithm takes advantage of a system for which most of the variables remain at an equilibrium state but does *not* take full advantage of a system for which the state variables have different rates of motion, but are not at equilibrium.

Both the ITA and WR algorithms attempt to exploit the loose or unidirectional coupling of MOS digital circuits, in that the relaxation converges rapidly when applied to loosely coupled systems. There are several types of circuits that can be divided into subcircuits of reasonable size only if moderate coupling between subcircuits is allowed. If a relaxation algorithm is used for such a circuit, many iterations will be required to achieve convergence. If such a moderately coupled system does not exhibit multirate behavior, then the ITA algorithm will be more efficient, because in ITA the computational cost of performing an iteration is lower than in WR. For ITA, only one Newton iteration is performed with each relaxation iteration. For the WR algorithm, each iteration involves solving nonlinear differential equations. These must be solved using an implicit integration method, each timestep of which will require that an algebraic system be solved by a Newton method, and to insure stability, the Newton iteration must be carried not just one step, but to convergence.

3. WAVEFORM RELAXATION-NEWTON AND ITERATIVE STEP REFINEMENT

It is possible to reduce the cost of each of the WR iterations by following the same strategy used in the algebraic relaxation-Newton algorithms. Instead of solving the nonlinear differential equations exactly with each relaxation iteration, they can be solved approximately, by performing one step of a function space or Waveform Newton method[13]. In presenting the precise algorithm for this Waveform relaxation-Newton algorithm (WRN) the following notation will be used.

$$v^{k,i}(t) = (v_1^k(t), \ldots, v_{i-1}^k(t), v_i^{k-1}(t), \ldots, v_n^{k-1}(t))^T. \quad (3)$$

Using this notation, the WRN algorithm for solving systems of the form of Eqn. 1 is given below.

Algorithm 1 (Gauss-Seidel WRN Algorithm)

$k \leftarrow 0$;
guess waveform $v^0(t)$; $t \in [0,T]$ such that $v^0(0) = v_0$
repeat {
  $k \leftarrow k+1$
  for all (i **in** N) {
    solve

$$\frac{d}{dt}[\, q_i(v^{k,i}(t)) + \frac{dq}{dv}(v^{k,i}(t))(v_i^k(t) - v_i^{k-1}(t)) \,] -$$

$$f_i(v^{k,i}(t)) + \frac{\partial f_i}{\partial v_i}(v^{k,i}(t))(v_i^k - v_i^{k-1}) = 0$$

    for ($v_i^k(t)$; $t \in [0,T]$),
      with the initial condition $v_i^k(0) = v_{i_0}$.
  }
}
until ($\max_{1 \le i \le n} \max_{t \in [0,T]} |v_i^k(t) - v_i^{k-1}| \le \epsilon$)

The amount of computation performed in the early iterations of the WRN can be reduced by using coarse numerical integration timesteps to solve the differential relaxation equations initially, and then refining the timesteps as the iterations progress. Specifically, the first relaxation iteration is computed with a numerical integration timestep equal to a user-supplied plotting increment. For subsequent relaxation iterations, the integration timesteps are chosen to be the same as the those in the previous iteration unless the *a posteriori* local truncation error estimates for the timestep from the previous iteration is too large. In that case, half the previous iteration timestep is used. This strategy has the advantage that, in general, timesteps will be placed more efficiently to control truncation error than if the standard predicted truncation error criteria is used. This is because the timestep selection is based on more accurate *a posteriori* error estimates available from previous relaxation iterations.

In Table 1 below, a comparison between the three methods is made for several different industrial circuits. The first example is a three inverter ring oscillator circuit. A substantial amount of floating capacitance makes the three inverters moderately coupled, but because the three nodes are oscillating at the same frequency, the circuit is not multirate. As can be seen in the table, the ITA algorithm is more efficient than WR. The second and third examples, a critical path from a microprocessor and the logic for a successive approximation register, are moderately coupled, and exhibit substantial multirate behavior. As expected, for these two circuits WR is more efficient than ITA. Because WRN allows full multirate, and has a low cost per iteration, the WRN algorithm is more efficient than either WR or ITA in all cases.

Table 1 - SPLICE2 vs RELAX2.3 vs WRN

Circuit	Mosfets	Nodes	SPLICE2	RELAX2.3	WRN
Ring Osc.	7	3	9.8	27	7.2
uP Control	116	66	194	160	137
SAR Rom	344	151	1025	1010	618

*On Vax11/785 running Unix

4. PARALLEL WAVEFORM RELAXATION ALGORITHMS

Relaxation techniques are particularly promising for parallel processing because the computational method naturally decomposes the problem. Recent results from a parallel implementation of the ITA algorithm indicates that significant parallelism is exploitable when relaxation methods are used[11].

An obvious way to parallelize the WR algorithm is to apply the Gauss-Jacobi version of WR. In this algorithm, the relaxation makes use of the waveforms computed at the previous iteration for all the subcircuits. All the subcircuits can then be analyzed independently by different processors.

One of the difficulties in applying the Gauss-Jacobi WR algorithm is that MOS digital circuits are highly directional, and, if the relaxation computation does not follow the signal flow, many iterations will be required to achieve convergence. Since large digital circuits are not one long chain but tend to be quite "wide", i.e., the outputs of gates fan out to more than one gate, it is possible to order the computation so that subcircuits in parallel "chains" can be computed in parallel, but the directionality of the circuit can still be followed by the relaxation computation. This will not allow for as much parallelism as the Gauss-Jacobi scheme, but can preserve some of the efficiency of the Gauss-Seidel scheme.

It is also possible to parallelize the WR algorithm while still preserving a strict ordering of the computation of the subcircuit waveforms by pipelining the waveform computation. In this approach, one processor would start

computing the transient response for a subcircuit. Once a first timepoint is generated, a second processor could begin computing the first timepoint for the second subcircuit, while the first processor computes the second timepoint for the first subcircuit. On the next step a third processor could be introduced, to compute the first timepoint for the third subcircuit, and so on.

The two algorithms were implemented on a 9 processor configuration of the Sequent Balance 8000 computer (larger configurations are available) a single bus shared-memory computer. The Balance system has a mutual exclusion primitive (or lock) that can be used to insure that only a single processor has access to protected sections of data. Both algorithms were implemented by taking advantage of the shared-memory architecture. A created list of subcircuits was placed in shared memory to allow any processor to compute the result for any subcircuit. A global queue was used and the locking mechanism applied to insure that no two processors could be working on the same subcircuit at the same time.

The results from several experiments for the two algorithms are given in Tables 2 and 3. As the results from the Eprom and microprocessor control circuit indicate, the timepoint pipelining algorithm makes much more efficient use of the available processors. In fact, as Table 3 shows, the timepoint pipelining algorithm running on the Balance 8000 runs substantially faster than the serial WR algorithm running on a Vax/780.

Table 2 - Gauss-Seidel/Gauss-Jacobi WR on several # of Processors.

Circuit	FET's	1	3	6	9
uP Control	66	595	338	270	259
Eprom	348	512	317	286	266

Table 3 - Timepoint Pipelining WR Algorithm on several # of Processors.

Circuit	FET's	1	3	6	9	VAX/780
uP Control	116	704	247	159	149	240
Eprom	348	745	265	185	182	212
Cmos Ram	428	3379	1217	642	496	960

The WRN algorithm combined with the iterative refinement stepsize strategy is more efficient than the WR algorithm for more tightly coupled systems, and it also has certain features that can be exploited on a parallel processor. To demonstrate this, consider solving the WRN iteration equation for v_i^k using the backward-Euler numerical integration algorithm. If τ_j is the time of the j^{th} timestep, the equation for $v_i^k(\tau_j)$ is

$$\left[\frac{\partial q_i}{\partial v_i}(v^{k,i}(\tau_j)) - h\frac{\partial f_i}{\partial v_i}(v^{k,i}(\tau_j)) \right](v_i^{k+1}(t+h) - v_i^k(\tau_j)) \qquad (4)$$

$$= hf(v^{k,i}(\tau_j)) - q_i(v^{k,i}(t+h))$$

$$+ q_i(v^{k,i}(\tau_{j-1})) + \frac{\partial q_i}{\partial v_i}(v^{k,i}(\tau_{j-1}))(v_i^{k+1}(\tau_{j-1}) - v_i^{k+1}(\tau_{j-1}))$$

where $v^{k,i}(t)$ is as defined in Eqn. 3. The step refinement strategy implies that before beginning the numerical computation of the v_i^k waveform, the number of timesteps that will be used, m, is known, as is each of the time discretization points, τ_j, $1 \leq j \leq m$. Also, $v^{k,i}(t)$ is known before beginning the computation of v_i^k. As examining Eqn. 4 reveals, the information available makes it possible to compute most of what is required to produce the m timepoint values, $v_i^k(\tau_j)$, $1 \leq j \leq m$, in parallel.

5. CONCLUSIONS

A new approach to relaxation algorithms based on waveform relaxation-Newton and an iterative step refinement strategy has been described, and its performance compared to existing relaxation algorithms. In addition, some of the issues connected with parallelizing WR have been described, and experimental results presented. Expansion of the scope of the relaxation techniques continues to include bipolar circuits and more advanced mosfet models. In addition, more advanced multirate timestep techiques are being investigated for the ITA algorithm and relaxation acceleration techniques are being added. Finally, additional parallel techniques are being explored to exploit different types of multiprocessors architectures.

ACKNOWLEDGEMENTS

The authors wish to thank Ken Kundert, Peter Moore, Don Webber, Guy Marong, for their contributions to the RELAX2 program, and the other members of the CAD Research group at Berkeley for their support. In addition, we would like to thank and F. Odeh at IBM's Yorktown Research group for many valuable discussions. The original version of Splice2 was written by Jim Kleckner. We would also like to thank Shiva Multisystems for the use of their equipment and facilities, and Sequent Computers for the use of their multiprocessor system. This research has been sponsored by DARPA under contract NESC-N39, NSERC of Canada, and grants from IBM, Philips, Hewlett-Packard, Texas Instruments and MICRO.

REFERENCES

[1] L.W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," Electronics Research Laboratory *Rep. No. ERL-M520, University of California, Berkeley, May 1975.*

[2] B.R. Chawla, H.K. Gummel, and P. Kozak, "MOTIS - an MOS timing simulator," *IEEE Trans. Circuits and Systems, Vol. 22, pp. 901-909, 1975*

[3] C. F. Chen and P. Subramaniam, "The Second Generation MOTIS Timing Simulator-- An Efficient and Accurate Approach for General MOS Circuits" *Proc. 1984 Int. Symp. on Circ. and Syst.*, Montreal, Canada, May 1984.

[4] K. Sakallah and S.W. Director, "An activity-directed circuit simulation algorithm," *Proc. IEEE Int. Conf. on Circ. and Computers*, October 1980, pp.1032-1035

[5] R. A. Saleh, J. E. Kleckner and A. R. Newton, "Iterated Timing Analysis and SPLICE1", *ICCAD'83 Digest*, Santa Clara, CA., 1983.

[6] E. Lelarasmee, A. E. Ruehli, A. L. Sangiovanni-Vincentelli, "The waveform relaxation method for time domain analysis of large scale integrated circuits," *IEEE Trans. on CAD of IC and Syst.*, Vol. 1, n. 3, pp.131-145, July 1982.

[7] J. White and A. Sangiovanni-Vincentelli, "Relax2.1 - A Waveform Relaxation Based Circuit Simulation Program" *Proc. 1984 Int. Custom Integrated Circuits Conference* Rochester, New York, June 1984.

[8] E. Lelarasmee, Private Communication, October, 1981

[9] M. Guarini and O. A. Palusinski, "Integration of Partitioned Dynamical Systems using Waveform Relaxation and Modified Functional Linearization," *1983 Summer Computer Simulation Proceedings*, Vancouver, Canada, July 1983.

[10] W.M.G. van Bokhoven, "An Activity Controlled Modified Waveform Relaxation Method" *1983 Conf. Proc. IEEE ISCAS*, Newport Beach, CA, May 1983.

[11] J. T. Deutsch "Algorithms and Architecture for Multiprocessor-Based Circuit Simulation", Ph.D. dissertation, U. C. Berkeley, Electronics Research Laboratory, 1985.

[12] J. M. Ortega and W.C Rheinbolt, *Iterative Solution of Nonlinear Equations in Several Variables* Academic Press, 1970.

[13] J. White, F. Odeh, A. Sangiovanni-Vincentelli, A. Ruehli, "Waveform Relaxation - Theory and Practice" *Trans. on Computer Simulation*, To appear.