Transient Sensitivity Computation for Waveform Relaxation-Based Timing Simulation

Chun-Jung Chen, Jyuo-Min Shyu*, and Wu-Shiung Feng

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

*Computer & Communication Research Laboratories, Hsinchu, Taiwan, R.O.C.

Abstract

Transient sensitivity computation is important in performing circuit optimization. In this paper, we present a method to compute the transient sensitivities for Waveform-Relaxation (WR) based circuit simulation. By partitioning the circuit into subcircuits during WR simulation, we compute the transient sensitivities for each subcircuit and then propagate the calculated results to the succeeding subcircuits to iterate for the entire sensitivities. Experimental results are shown to demonstrate the effectiveness of our proposed method.

1. Introduction

Transient sensitivity computation is very important in applications such as circuit optimization[5] and simulation validation[4]. Direct methods[2,5] and adjoint methods[1] have been in wide use in circuit design community. These methods, while effective, are built in conventional circuit simulators such as SPICE, which are not efficient in dealing with large scale MOS digital circuits. Relaxation-based timing simulators have been proposed to speed up simulation time for large circuits. Unfortunately, no methods have been devised for computing transient sensitivities in relaxation-based simulators.

In this paper we propose a Waveform Relaxation-based method to compute the transient sensitivities of circuits, and this method is similar to those which use Waveform Relaxation to compute timing waveforms. The new method calculates the transient sensitivities of subcircuits separately, and then propagates each subcircuit's sensitivities to its succeeded subcircuits. The propagation of *sensitivity waveforms* is performed for all subcircuits and is iterated several times until all sensitivity waveforms have converged. We have succeeded in computing the sensitivities of MOS circuits by the new method, and the obtained sensitivity waveforms match those obtained by direct method well. Hence, we have successfully upgrade the sensitivity computation from *circuit-simulation* level into *timing-simulation* level.

The remaining of this paper describes the details of this new method and presents a number of examples to show the sensitivity computation in this method. In section 2 we describe the methods computing transient sensitivities in a divided subcircuit. In section 3 we discuss the assembling of entire sensitivities and provide an algorithm which based on Waveform Relaxation method. These methods had been implemented and joined with a Waveform Relaxation-based timing simulator, MOSTIME[6]. Section 4 demonstrates some examples run on it. To justify the proposed method, we compare these results with those obtained from direct sensitivity computation algorithms[5]. Finally, concluding remarks are given in section 5.

2. Sensitivity Computation for a Subcircuit

We will formulate the sensitivity equation, which deals a time-step and can be solved to acquire sensitivities of this time-step, for a subcircuit. In computing the sensitivities of a subcircuit, we use time-steps chosen in previous timing simulation, and utilize the sensitivity equation to calculate the subcircuit sensitivities step by step. The sensitivities of each time-step are then stored into memory tables to form the sensitivity waveforms. Once the sensitivity waveforms of whole-time interval have been generated, the sensitivity computation of this subcircuit stops. We will now derive sensitivity equation.

Let $y = y(x, t)$ be the vector of the circuit variables of the subcircuit. Where t is the time and x is the vector of design parameters of the subcircuit. Let y be a $1 \times n_y$ vector and x be $1 \times n_x$. To formulate the sensitivity equation for a subcircuit, we begin by computing S_{yx} , a $n_y \times n_x$ matrix

$$S_{yx} = \frac{\partial y(x, t)}{\partial x} \quad (1)$$

S_{yx} is composed of column vectors S_{yx_j} , $1 \leq j \leq n_x$, which are sensitivity vectors with respect to x_j , respectively. To simplify the derivation, we consider S_{yx_j} only. This is because that the other column vectors of S_{yx} can be calculated in the same way. Now we restate the direct method, which is the basis of our new method, for S_{yx_j} .

A subcircuit can be described as the following system of differential equations:

$$f(y(x, t), \dot{y}(x, t), x, t) = 0 \quad (2)$$

where a dot represents a partial differentiation with respect to time. Now we differentiate (2) with respect to x_j , which gives us

$$\frac{\partial f}{\partial y} \frac{\partial y}{\partial x_j} + \frac{\partial f}{\partial \dot{y}} \frac{\partial \dot{y}}{\partial x_j} + \frac{\partial f}{\partial x_j} = 0 \quad (3)$$

Then introducing $S_{yx_j} = \frac{\partial y}{\partial x_j}$ into (3), we get

$$\frac{\partial f}{\partial y} S_{yx_j} + \frac{\partial f}{\partial \dot{y}} \dot{S}_{yx_j} + \frac{\partial f}{\partial x_j} = 0 \quad (4)$$

If we redefine the f in (4) to be the system of differential equations on the whole circuit rather than a subcircuit, then (4) is the sensitivity equation used in direct method. In order to derive the sensitivity equation for a subcircuit, we have to consider other circuit variables. Let w be the vector for input variables of the subcircuit, and d be the vector for design parameters of whole circuit (Note that When we calculate the sensitivities of a subcircuit with respect to d_j , the j th element of d , the design parameter d_j may or may not be in this subcircuit). We restate the system of

subcircuit's differential equations as follows:

$$f(y(d, t), \dot{y}(d, t), w(d, t), \dot{w}(d, t), d, t) = 0 \quad (5)$$

Now, we differentiate (5) with respect to d_j to yield

$$\frac{\partial f}{\partial y} S_{y d_j} + \frac{\partial f}{\partial \dot{y}} \dot{S}_{y d_j} + \frac{\partial f}{\partial w} S_{w d_j} + \frac{\partial f}{\partial \dot{w}} \dot{S}_{w d_j} + \frac{\partial f}{\partial d_j} = 0 \quad (6)$$

where $S_{y d_j} = \frac{\partial y}{\partial d_j}$, and $S_{w d_j} = \frac{\partial w}{\partial d_j}$. Because d is now defined as a global vector, computations for any subcircuit can use (6), as long as the calculated sensitivities are with respect to d_j . Eq. (6) is the equation for subcircuit sensitivity computation in our new proposed method.

To solve (6), we apply the following trapezoidal(TR) rule to approximate $\dot{m}(x, t_{n+1})$, the derivative of $m(x, t)$ at time $t = t_{n+1}$:

$$\begin{aligned} \dot{m}(x, t_{n+1}) &= \frac{2m(x, t_{n+1})}{h} - \left(\frac{2m(x, t_n)}{h} + \dot{m}(x, t_n) \right) \\ &= \frac{2m(x, t_{n+1})}{h} - O_m \end{aligned} \quad (7)$$

where t_n is the current time, h the step size, and O_m is the "old" m -related data at t_n . Equation (6) thus becomes:

$$\begin{aligned} \left(\frac{\partial f}{\partial y} + \frac{\partial f}{\partial \dot{y}} \frac{2}{h} \right) S_{y d_j} &= - \left(\frac{\partial f}{\partial w} + \frac{\partial f}{\partial \dot{w}} \frac{2}{h} \right) S_{w d_j} + \frac{\partial f}{\partial y} O_{S y d_j} \\ &\quad + \frac{\partial f}{\partial \dot{w}} O_{S w d_j} - \frac{\partial f}{\partial d_j} \end{aligned} \quad (8)$$

or, equivalently,

$$J_y S_{y d_j} = -J_w S_{w d_j} + \frac{\partial f}{\partial \dot{y}} O_{S y d_j} + \frac{\partial f}{\partial \dot{w}} O_{S w d_j} - \frac{\partial f}{\partial d_j} \quad (9)$$

where $J_y = \left(\frac{\partial f}{\partial y} + \frac{\partial f}{\partial \dot{y}} \frac{2}{h} \right)$ is the Jacobian of f , and $J_w = \left(\frac{\partial f}{\partial w} + \frac{\partial f}{\partial \dot{w}} \frac{2}{h} \right)$.

Note that since the right-hand side of (9) are known at each time point, $S_{y d_j}$ can be calculated by solving a system of linear equations.

3. Sensitivity Computation for the Entire Circuit

The Waveform Relaxation(WR) algorithm has been approved to be an efficient and fast method for solving large scale MOS digital circuits, which are the major technology of VLSI circuits. This is partly because of the small feedback between adjacent MOS subcircuits, partly because of the numerical nature of MOS circuits and the remarkable property of WR method. The subcircuit computation method proposed in previous section calculates the transient sensitivities of a subcircuit only. We have to exploit some *propagating methods* to link and to interact these results with other subcircuits and eventually acquire converged values, and this is the same as timing simulation process. The timing propagating methods can achieve above functions in timing simulation, and we could use them in sensitivity simulation if they still work in sensitivity simulation. All of WR method, event-driven approach[10] and Waveform-Newton method[11] can yield converged waveforms among subcircuits in timing simulation. But could they still work in sensitivity simulation? In this paper, we use WR as the propagating method, and experimental results show that WR can work well.

To exploit WR method, we shall perform some specific steps. The major are partitioning of circuit into subcircuits, ordering of the subcircuits, scheduling of subcircuits analysis and waveform

processing. Before we discuss these steps for sensitivity computation, we have to clarify that the sensitivity WR, WR algorithm for sensitivity computation, can be combined with timing WR, WR algorithm for transient response computation. The sensitivity WR can also be performed individually, and it will forms a postprocess which trails the timing simulation. We consider the two different situation. If the two WR processes were combined together, we could save time for generating Jacobian matrices, L/U parts of linear system and values of circuit variables, the *computing information*. If they were separated, we could treat the computation of sensitivity more conveniently. In this situation, we could reorder the subcircuits, calculate sensitivity with respect to arbitrary design parameter at any time we need, use different time-steps to compute the sensitivities, and even use other propagating methods rather than WR method to assemble the entire sensitivity. In considering the experimental necessity, we compute the sensitivities in a post-process.

Now let's begin the investigating of WR method in sensitivity computation. The algorithm has been listed as following.

```

1: PROCEDURE SensitivityWaveformRelaxation;
2: BEGIN
3:   k := 0;
4:   REPEAT
5:     k := k + 1; (The convergence iteration.)
6:     FOR(subcircuit s, on the order got in ordering step) DO
7:       BEGIN (The subcircuit loop.)
8:         t := t begin;
9:         WHILE(t <= t end) DO
10:          BEGIN (The time loop.)
11:            update input variables w and circuit variables y
12:            that were obtained in timing simulation;
13:            recompute  $J_y$  and  $J_w$ ;
14:            FOR(all design parameter  $d_j$  in  $d$ ) DO
15:              BEGIN (For all design parameters interested.)
16:                solve equation (9), and then
17:                store  $S_{y d_j}$  into tables;
18:              END;
19:            t := t + time_step;
20:          END;
21:        UNTIL all sensitivity waveforms have converged;
22:      END

```

Typical WR method utilizes three programming loops to compute, and they are convergence-iterating, subcircuits and time loops. In our sensitivity case, there appears a new loop for all interested design parameters. We may regard the computation for a sensitivity waveform with respect to each design parameter as a specific WR process, but this is too expensive due to the overhead of recomputing the computing information in each process. In our algorithm, we put the design parameter loop, as listed from line 13 to 17, into the innermost loop, from line 9 to 19, of WR algorithm, from which we calculate sensitivity waveforms with respect to different designed parameters at the same time and then save the time for generating computing information.

4. Results

In this section we give some examples to demonstrate the transient sensitivities computed by our new method and compare these sensitivities to those computed by direct method. The first example is a gate-chain, which is composed of a NAND gate and two successive NOT gates, shown in Fig. 1(a). The connection of internal transistors of NAND and NOT gates are given in Fig. 1(b), which is valid in the remained examples. Fig. 1(c). represents the timing waveforms and sensitivity waveforms with respect to the width of $m1$ in $X1$. We compare these with those computed by direct method and show them in Fig. 1(d), and the good agreement between the two results is justified. This example illustrate the correctness of our new method in the one-way circuit, which certify that the sensitivity waveforms could be computed separately and then assembled correctly. Fig. 2(a). is a simple SR flip flop, a tightly coupled example. However, the sensitivity simulation converges, and it takes three iterations while the timing simulation takes five to converge. Fig. 2(b). exhibits sensitivity waveforms of each iteration and Fig. 2(c). compares them with those by direct method. It shows that WR method is quite suitable for assembling of entire sensitivities, since it takes less iterations to converge. We will confirm more resolutely on this after the third example, a ring-oscillator. In the ring-oscillator, sensitivity WR takes six iterations to converge while the timing WR takes eight. Fig. 3(a). is the ring-oscillator, Fig. 3(b). represents the sensitivity waveforms of each iterations and Fig 3(c). is the sensitivity comparison. Fig. 4. shows a bigger circuit, a full-adder. From Fig. 4(d), we see that the new proposed method still work well.

(a)

(b)

(c)

(d)

(a)

Fig. 1. (a) A gates chain. (b) The internal connection of gates. (c) Timing and sensitivity waveforms, which with respect to the width of $m1$ in $X1$ and their unit is $\nu/10^{-6}m$, which is the unit of sensitivities of remained ex-amples. (d) The comparison of sensitivity waveforms.

(b)

(b)

(c)

(c)

Fig. 2. (a) A simple SR flip flop and its timing waveforms. (b) The sensitivity waveforms, with respect to the width of m1 in X1, of each iteration. (c) The comparison of sensitivity waveforms.

Fig. 3. (a) A ring-oscillator. (b) The sensitivity waveforms, with respect to the width of m1 in X1, of each iteration. (c) The comparison of sensitivity waveforms.

(a)

(a)

Fig. 4. (a) The full adder. (b) The internal connection of XOR gate which is used in (a). (c) The timing waveforms. (d) Comparison of sensitivity waveforms, which with respect to the width of m3 in X1.

5. Conclusion and Future Works

In this paper we proposed a new method to compute transient sensitivity of circuits. We compute the sensitivity of each subcircuit and then assemble the entire sensitivities by WR method. Experimental results show that the proposed new method can work perfectly even though the circuit contains global feedback. Hence, we successfully upgrade the sensitivity computation from the circuit-simulation level into the timing-simulation level, and as a consequence, have it inherited benefits of timing simulation. Since timing simulation is flexible, we could add some time-saving techniques into the sensitivity WR, such as latency and partial waveform convergence checking. These are future goals of our research.

Reference

- [1] S. W. Director and R. A. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Trans. on Circuit Theory*, vol. CT-16, pp. 318-323, Aug. 1969.
- [2] D. A. Hocevar, P. Yang, T. N. Trick, and B. D. Epler, "Transient sensitivity computation for MOSFET circuits," *IEEE Trans. on Computer-Aided Design*, vol. CAD-4, pp. 609-620, Oct. 1985.
- [3] 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 Computer-Aided Design*, vol. CAD-1, pp. 131-145, Jul. 1982.
- [4] T. V. Nguyen, P. Feldmann, S. W. Director, R. A. Rohrer, "SPECS simulation validation with efficient transient sensitivity computation," *JCCAD*, pp. 252-255, 1989.
- [5] W. T. Nye, D. Riely, A. Sangiovanni-Vincentelli, and A. L. Tits, "DELIGHT.SPICE: an optimization-based system for the design of integrated circuits," *IEEE Trans. on Computer-Aided Design*, vol. CAD-7, pp. 501-519, Apr. 1988.
- [6] Chun-Jung Chen, "MOSTIME-a MOS timing simulator," *Master Thesis*, Department of Electrical Engineering, National Taiwan University, 1989.
- [7] Jyuo-Min Shyu, A. Sangiovanni-Vincentelli, J. P. Fishburn, A. E. Dunlop, "Optimization-based transistor sizing," *IEEE Journal of Solid-State Circuit*, vol. 23, pp. 400-409, 1988.
- [8] T. Shima, T. Sugawara, S. Moriyama, H. Yamada, "Table look-up MOSFET modeling system using a 2-D device simulator and monotonic piecewise cubic interpolation," *IEEE Trans. on Computer-Aided Design*, vol. CAD-2, Apr. 1983.
- [9] T. N. Trick, F. R. Colon, and S. P. Fan, "Computation of capacitor voltage and inductor current sensitivities with respect to initial conditions for the steady-state analysis of nonlinear periodic circuits," *IEEE Trans. Circuit Syst.*, vol. CAS-22, pp. 391-396, May 1975.
- [10] R. A. Saleh, and J. K. White, "Accelerating relaxation algorithms for circuit simulation using Waveform-Newton and step-size refinement," *IEEE Trans. on Computer-Aided Design*, vol. CAD-9, pp. 951-958, Sep. 1990.