

# Transient Sensitivity Computation for Waveform Relaxation-Based Timing Simulation<sup>+</sup>

Chun-Jung Chen Jyuo-Min Shyu\* Wu-Shiung Feng

Dept. of Electrical Engineering  
National Taiwan University, Taipei, Taiwan, R.O.C  
\*Computer & Communication Research Laboratories  
Industrial Technology Research Institute, Hsinchu, Taiwan, R.O.C

## Abstract

Transient sensitivity computation is very important in many areas. 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, we compute the transient sensitivities for each "active" 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.

To further exploit the latency of the subcircuits (due to design parameters) during sensitivity computation, we also propose a scheme to schedule only the "active" subcircuits for WR sensitivity computation. This saves some CPU time.

The remaining of this paper describes the details of the proposed method and presents a number of examples to justify the algorithm. In section 2 we describe the method for computing the transient sensitivity for a subcircuit. Section 3 follows by computing the sensitivities for the entire circuit based on WR. Section 4 justifies the proposed method by comparing the computed results with those obtained by Direct method. Finally, we give a conclusion in section 5.

## 1. Introduction

Transient sensitivity computation is very important in applications such as circuit optimization [5, 6], 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 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 (WR)-based method to compute the transient sensitivities of a circuit. The method is similar to the WR algorithms [3] in computing timing waveforms. After partitioning the circuit into subcircuits, it first calculates the transient sensitivities of the subcircuits, then propagates each subcircuit's sensitivity to its succeeding subcircuits. The computation of *sensitivity waveforms* is iterated until all the sensitivity waveforms are converged. Experimental results on several test cases have shown that the computed results using the proposed method agree very well with those obtained by using Direct method.

<sup>+</sup>This work was supported by National Science Council (under contract NSC80-0404-E-002-35).

## 2. Sensitivity Computation for a Subcircuit

Let  $y = y(x, t)$  be the vector of the circuit variables of a 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:

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

consisting of column vectors  $S_{yx_j}$ ,  $1 \leq j \leq n_x$ , which are sensitivity vectors of  $y$  with respect to  $x_j$ 's. To simplify the derivation, we consider  $S_{yx_j}$  only, since other column vectors of  $S_{yx}$  can be calculated in the same way.

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

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

where  $'$  represents partial differentiation with respect to time. We differentiate (2) with respect to  $x_j$  to give

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

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

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

Note that in (4), if  $f$  were the system of differential equations for the entire circuit rather than a subcircuit, then (4) would be the sensitivity equation used in Direct methods.

To derive the sensitivity equation for a subcircuit, let  $w$  be the vector of the input variables of the subcircuit, and  $d$  be the vector of the design parameters of the entire 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 may or may not be in the subcircuit.) The system of differential equations for the subcircuit thus becomes:

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

Differentiating (5) with respect to  $d_j$ , we obtain the sensitivity equation for the subcircuit of interest:

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

where  $S_{yd_j} = \frac{\partial y}{\partial d_j}$ , and  $S_{wd_j} = \frac{\partial w}{\partial d_j}$ .

To solve (6), we apply the trapezoidal rule to approximate the derivative of  $m(x, t)$  with respect to  $t$  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,  $h$  is the step size, and  $O_m$  is the "old"  $m$ -related data at  $t_n$ . Equation (6) thus becomes:

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

or, equivalently,

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

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  $J_w = \left( \frac{\partial f}{\partial w} + \frac{\partial f}{\partial \dot{w}} \frac{2}{h} \right)$ , whose components are either known quantities in transient analysis or can be calculated easily.

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

### 3. Sensitivity Computation for the Entire Circuit

We use WR algorithm to compute the sensitivity of the entire circuit. The computation is performed after WR tran-

sient analysis, since we can then utilize the analysis results such as subcircuits, time steps, calculated transient response and differential equations data structures. Note that, while it is possible to combine the computation for transient response and sensitivity, a separate procedure for computing sensitivity allows reordering of the subcircuits, calculating sensitivity with respect to any design parameter interactively, as well as using different time steps from those selected during transient analysis. The following pseudo code shows the sensitivity WR algorithm:

```

PROCEDURE SensitivityWaveformRelaxation(j);
{ Sensitivity computation for  $d_j$  }
1: BEGIN
2:   Pick  $s_j$ , a subcircuit containing  $d_j$ ;
3:   Find a directed subgraph  $G_j$  which is
      composed of vertices and directed edges
      reached by Breadth-First Search (starting
      from  $s_j$  through the edges traversed
      by the search);
      { Vertex = subcircuit, Directed edge = "directed"
        connection between the subcircuits }
4:   k := 0;
5:   REPEAT
6:     k := k + 1; { Iteration for convergence }
7:     FOR(each subcircuit, in the topological order of  $G_j$ )
8:       DO BEGIN { The subcircuit loop }
9:         t := t_begin;
10:        WHILE(t <= t_end) DO
11:          BEGIN { Sensitivity loop }
12:            Update time_step, input variables  $w$  and
              circuit variables  $y$  from the stored
              timing waveforms;
13:            Assemble input sensitivities  $S_{wd_j}$ ,  $O_{Syd_j}$ 
              and  $O_{Swd_j}$ ;
14:            Compute  $\frac{\partial f}{\partial d_j}$ ,  $J_y$  and  $J_w$ ;
15:            Solve equation (9), and store  $S_{yd_j}$  into tables;
16:            t := t + time_step;
17:          END;
18:        END;
19:      UNTIL all sensitivity waveforms are converged;
20:   END

```

Note that, to exploit the latency of the subcircuits, we employ a scheme (Lines 2, 3, and 7) to schedule only the subcircuits "active" during sensitivity computation.

### 4. Experimental Results

We have implemented the proposed algorithms in our

timing simulator, MOSTIME [7], and run the program on SUN SPARC II workstation. To justify the method, we implemented the Direct method, and compared the sensitivities computed by each method for several test circuits. The first circuit is a gate chain, shown in Fig. 1(a). The connection of internal transistors for Nand and Not gates are also given (same for all examples). Fig. 1(b) shows the timing waveforms. Fig. 1(c) compares the sensitivities, which are both with respect to the width of m3 in X1. It is clear that very good agreement between the two results has been obtained. The next circuit is a clocked SR flip flop, shown in Fig. 2(a). Fig. 2(b) shows the timing waveforms, and Fig. 2(c) is the comparison results.

The third circuit is a full adder. Fig. 3(a) is the schematic of the adder. We use this circuit to demonstrate the reordering of subcircuits in sensitivity computation. The timing and sensitivity computations are shown in Fig. 3(b) and Fig. 3(c), respectively. We compared the CPU time for different design parameters in Fig. 3(d). We observed that the positions of the design parameters determine the order of the subcircuits for sensitivity computation, which in turn influences the CPU time used. The last circuit is a 8-bit adder, composed of 8 full adders stated in Fig. 3(a). Fig. 4(a) is its schematic and Fig 4(b) shows the sensitivity comparison on outputs V7 and V8.

Finally, we summarize the CPU time for the four circuits in Table 1. The CPU time used is for sensitivity computations for both WR-based and Direct methods. We note that more CPU time savings are obtained for larger circuits when WR-based method is used.

## 5. Conclusion

In this paper we proposed a new method to compute the transient sensitivities for WR-based timing simulation. After partitioning the circuit into subcircuits, we compute the sensitivity of each subcircuit and then "assemble" the entire sensitivities by WR method. Experimental results show that the proposed method provides sensitivity waveforms which are very close to those obtained by Direct method, in reasonably short time, and works well even though the circuit contains feedback loops.

## 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 Tran. on Computer-Aided Design*, vol. CAD-1, pp. 131-145, Aug. 1982.
- [4] T. V. Nguyen, P. Feldmann, S. W. Director and R. A. Rohrer, "SPECS simulation validation with efficient transient sensitivity computation," *ICCAD*, pp. 252-255, 1989.
- [5] W. T. Nye, D. Riley, 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] Jyuo-Min Shyu and A. Sangiovanni-Vincentelli, "ECSTASY: a new environment for IC design optimization," *ICCAD*, pp. 484-487, 1988.
- [7] Chun-Jung Chen, "MOSTIME-a MOS timing simulator," *Master Thesis*, Department of Electrical Engineering, National Taiwan University, 1989.

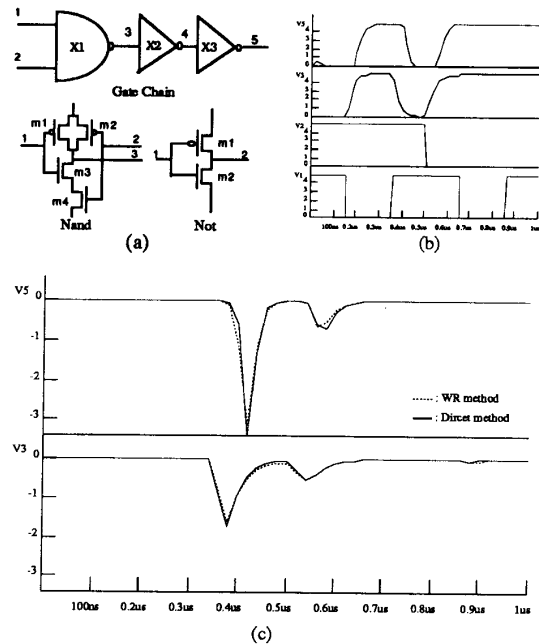


Fig. 1. (a) A gate chain and its internal connection. (b) Timing waveforms. (c) Sensitivity waveforms with respect to the width of m3 in X1.

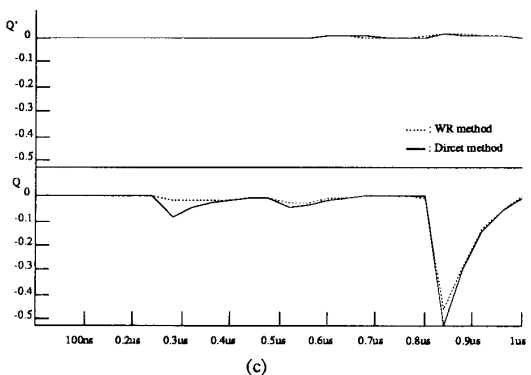
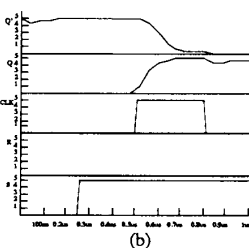
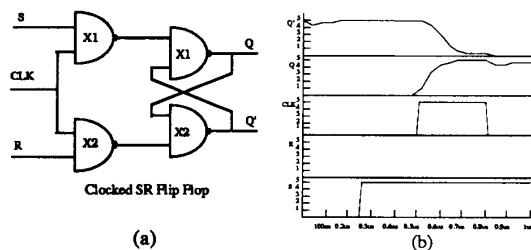
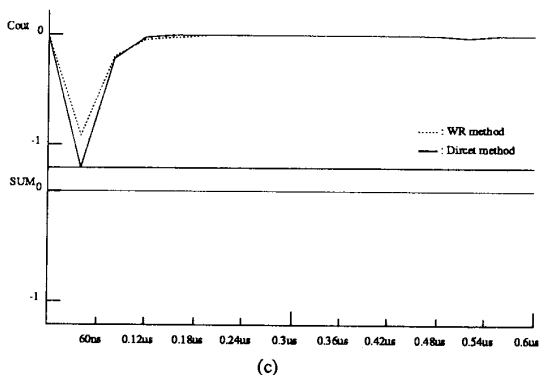
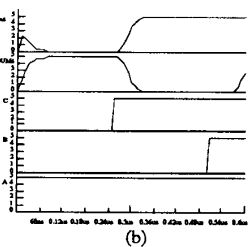
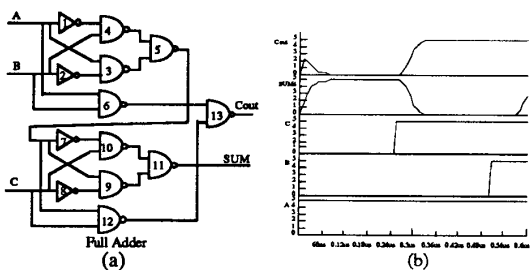


Fig. 2. (a) Clocked SR flip flop. (b) Timing waveforms. (c) Sensitivity waveforms with respect to the width of m2 in X1.



Design parameter	Subcircuit order for sensitivity computation	CPU time used	
		WR based method	Direct method
width of m1 in gate 1	1, 4, 5, 7, 9, 10, 11, 12, 13	.917	4.66
width of m2 in gate 6	6, 13	.233	4.66
width of m1 in gate 13	13	.15	4.66

CPU time is in SUN SPARC 2 seconds.

(d)

Fig. 3. (a) Full adder. (b) Timing waveforms. (c) Sensitivity waveforms with respect to the width of m2 in gate 6. (d) Sensitivity computations with respect to different design parameters.

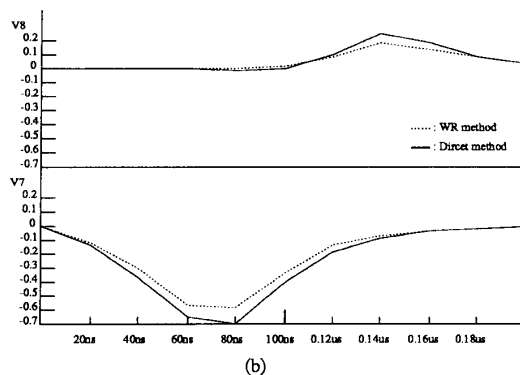
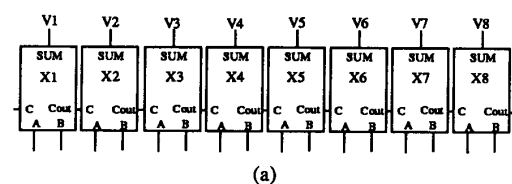


Fig. 4. (a) 8-bit adder. (b) Sensitivity waveforms with respect to the width of m2 in gate 4, shown in Fig. 3(a), of X7.

Table 1. Summary of experimental results.

Circuits	Number of transistors	Number of nodes	Number of iterations	CPU time used	
				WR based method	Direct method
Gate Chain	8	4	2	.45	.441
Clocked SR Flip Flop	16	8	2	.55	.65
Full Adder	44	22	2	.233	4.66
8-bit Adder	352	176	2	.833	4.67

CPU time is in SUN SPARC 2 seconds.