Transient Sensitivity Computation for Transistor Level Analysis and Tuning

Tuyen V. Nguyen*, Peter O'Brien, and David Winston*****

*IBM Austin Research Laboratory, Austin, TX

** IBM EDA, Austin, TX ***IBM EDA, Hopewell Junction, NY

Abstract

This paper presents a general method for computing transient sensitivities using both the direct and adjoint methods in event driven controlled explicit simulation algorithms that employ piecewise linear device models. This transient sensitivity capability is intended to be used in a simulation environment for transistor level analysis and tuning. Results demonstrate the efficiency and accuracy of the proposed techniques. Examples are also presented to illustrate how the transient sensitivity capability is used in timing characterization and circuit tuning.

1.Introduction

Sensitivity information is extremely useful in assessing the performance and robustness of VLSI circuits, especially as design trends move continuously towards higher performance at lower costs and faster turn around times. These high performance designs require better exploitation of process technology capabilities and more dependence on transistor level analysis and verification. Information on how the design responds to changes in design variables and parasitics is particularly useful for verification and optimization. In fact, DC and frequency domain sensitivities have been widely used in the design of analog integrated circuits. For digital circuits, the information of interest such as timing, power, and noise requires transient analysis at the transistor level, especially as designs move into the deep submicron regime and beyond. Traditional gate level analysis using library based precharacterized delay models no longer provides adequate accuracy for high performance designs. Dynamic characterization of channel-connected components at the transistor level during static timing analysis using traditional circuit simulation is too computationally inefficient. For circuit tuning, this inefficiency problem becomes even more pronounced. Therefore, it is important to have an efficient circuit/timing simulation environment that can handle large and complex VLSI circuits. In addition, an efficient transient sensitivity computation will greatly enhance the core simulation engine for transistor level analysis and circuit tuning. In order to address these needs, this paper proposes a method for computing transient sensitivities in Adaptively Controlled Explicit Simulation (ACES)[1], an efficient event driven simulator that employs piecewise linear (PWL) device models.

In general, there are two well known methods for computing sensitivities: the direct[3] and the adjoint[2] methods. The advantages and disadvantages of these methods for sensitivity computation have been discussed in detail [3][4]. In general, the direct method is advantageous when the sensitivities of a large number of circuit responses with respect to a few circuit parameters are desired. On the other hand, the adjoint method is more advantageous when the sensitivities of a few performance functions with respect to a large number of circuit parameters are required. The latter is usually the case in practice. However, there are inherent limitations that make the computation of adjoint transient sensitivities expensive in traditional circuit simulation[3]. It is usually more convenient and possibly more efficient to incorporate the direct method for computing transient sensitivities in traditional or relaxation based circuit simulators[3][5]. As a result, the efficiency of the transient sensitivity computation is compromised in these circuit simulation environments. The key to resolve the above problems is to take advantage of the simplified device models and the event driven nature of the more efficient circuit/timing simulators[4][6][7].

The paper is organized as follows: an overview of the transient sensitivity computation using both the direct and adjoint methods in ACES is presented in Section 2. Section 3 presents some results to demonstrate the effectiveness of the sensitivity computation together with examples of how sensitivity information is used in timing characterization and circuit tuning. Section 4 concludes the paper.

2.Transient Sensitivity Computation in ACES

A brief review of ACES is given here to facilitate the discussion of the procedure to compute adjoint transient sensitivities in ACES. A detailed description of ACES can be found in [1]. ACES employs a controlled explicit numerical integration algorithm with PWL device models and circuit partitioning. The use of controlled explicit simulation allows ACES to compute directly the times to reach the breakpoints of the PWL models without the full matrix inversion required by implicit integration algorithms. Circuit partitioning is used to take advantage of circuit sparsity and latency. The combination of these ideas allows ACES simulation to be performed in an event driven manner.

2.1. Overview of Direct Sensitivity Computation in ACES

An overview of the procedure for computing transient sensitivities in ACES using the direct method is shown in Fig. 1. The original

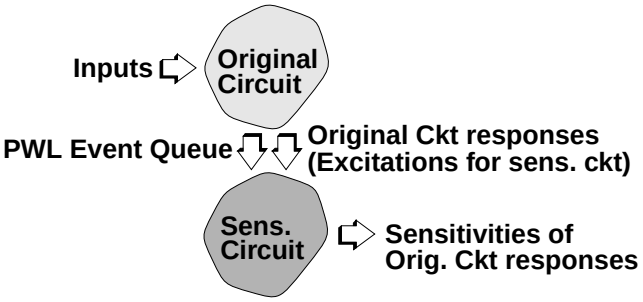

Fig. 1*: An overview of the direct method for computing transient sensitivities in ACES*

circuit is first simulated forward in time. The events due to the PWL elements are stored in an event queue during the original simulation. At each PWL event, the index of the parent partition and all the necessary information about the PWL element causing the event are stored. Then this PWL event queue and the appropriate responses of the original ACES simulation are used in the simulation of the associated sensitivity circuit. The sensitivity circuit has the same topology as the original circuit and its elements are derived by simply differentiating the BCRs of the elements of the original circuit with

0-7803-5832-X /99/$10.00

respect to the sensitivity parameter of interest. The circuit partitioning of the original circuit is also preserved for the sensitivity circuit. The PWL event queue captures the values of the time varying conductances of the resistive two terminal or multiterminal elements such as diodes, MOSFETs, or bipolar transistors in the sensitivity circuit. The excitations of the sensitivity circuit come from the selected responses, which are dependent upon the specified sensitivity parameter, of the original circuit. The simulation of the sensitivity circuit provides the sensitivities of all circuit responses with respect to the specified parameter. Note that the simulation of the sensitivity circuit can be performed concurrently with the simulation of the original circuit. However, for ease of implementation and for maintaining compatibility with the adjoint method, it is better to store the PWL event queue as well as necessary original circuit responses for the simulation of the sensitivity circuit after the simulation of the original simulation is completed.

2.2. Overview of Adjoint Sensitivity Computation in ACES

An overview of the procedure for computing transient sensitivities in ACES using the adjoint method is shown in Fig. 2. First, the original circuit is simulated forward in time. The events due to the PWL elements are stored in an event queue during the original simulation. At each PWL event, the index of the parent partition and all the necessary information about the PWL element causing the event are stored. Then the adjoint circuit is constructed from the original circuit and simulated backward in time. The inputs of the adjoint circuit are derived from the specified circuit performance function. These auxiliary excitations constitute the inputs for the adjoint simulation. The same partitions of the original circuit can be used for the adjoint simulation. During the adjoint simulation, the PWL event queue is traversed backward in time. At each event time, the information about the PWL element and its parent partition are retrieved for the analysis of the adjoint circuit at that particular time point. Finally, the results of the adjoint simulation are convolved with the results from the original simulation to provide the required sensitivities.

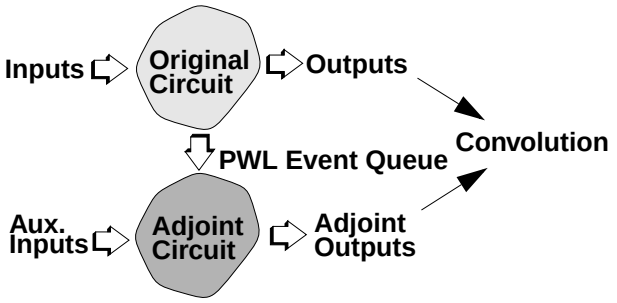

Fig. 2: *An overview of the adjoint method for computing transient sensitivities in ACES*

2.3. Piecewise Linear Circuit Elements in the Sensitivity and Adjoint circuits

In this subsection, we will derive the characteristic equations of the elements in the sensitivity and adjoint circuit corresponding to the PWL devices, which model all the resistive elements in ACES. The adjoint characteristic equations for other circuit elements can be found in [2][4]. Note again that the temporal variable is $\tau = t$ for the sensitivity circuit and $\tau = t_o + t_f - t$ for the adjoint circuit, In order to simplify the discussion, consider a two terminal device, the PWL characteristic of which is shown in Fig. 3. Formally, the i-v characteristic of a PWL two terminal resistive element can be written as

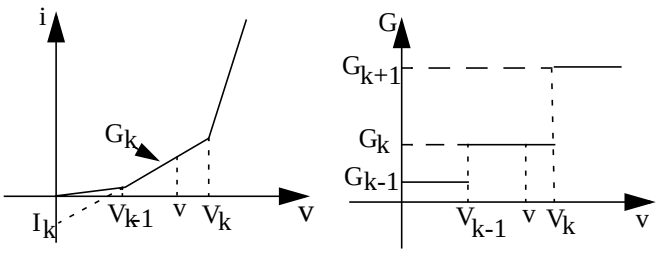

Fig. 3: *The i-v characteristic and the conductance of a two terminal PWL element*

$$i = \sum_{k=-\infty}^{\infty} (G_k v + I_k)[u(v - V_k) - u(v - V_{k+1})] \tag{1}$$

where V_k and V_{k+1} are the breakpoints in the PWL model, $(G_k v + I_k)$ is the equation representing the k^{th} linear segment of the model, and $u(v)$ is the unit step function. Note that the index is extended to infinity to simplify the notation. Changing the index in Eq.(1) yields

$$\begin{aligned} i &= \sum_{k=-\infty}^{\infty} (G_k v + I_k)u(v - V_k) - \sum_{k=-\infty}^{\infty} (G_{k-1} v + I_{k-1})u(v - V_k) \\ &= \sum_{k=-\infty}^{\infty} (\Delta G_k v + \Delta I_k)u(v - V_k) \end{aligned} \tag{2}$$

where $\Delta G_k = G_k - G_{k-1}$ and $\Delta I_k = I_k - I_{k-1}$. Hence, the conductance of the PWL model can be written as

$$\begin{aligned} G = \frac{\partial i}{\partial v} &= \sum_{k=-\infty}^{\infty} (\Delta G_k)u(v - V_k) + \sum_{k=-\infty}^{\infty} (\Delta G_k v + \Delta I_k)\delta(v - V_k) \\ &= \sum_{k=-\infty}^{\infty} (\Delta G_k)u(v - V_k) \end{aligned} \tag{3}$$

where $\delta(v)$ is the Dirac delta function and $(\Delta G_k v + \Delta I_k)\delta(v - V_k) = 0$ due to the continuity of the current at the breakpoints of the PWL model.

It can be shown [4] that for a general two terminal device, $i(t) = i(v(t))$, the corresponding element in the sensitivity and adjoint circuit can be described by

$$\hat{i}(\tau) = \left.\frac{\partial i}{\partial v}\right|_{\hat{v}(\tau)} \hat{v}(\tau) = G(\hat{v}(\tau))\hat{v}(\tau) \tag{4}$$

In ACES, it is more convenient and efficient to operate in the derivative space [1]. The corresponding element of the two terminal device described by Eq.(4) in the derivative space is given by

$$\frac{d\hat{i}}{d\tau} = G\frac{d\hat{v}}{d\tau} + \frac{\partial G}{\partial \tau}\hat{v} \tag{5}$$

Therefore, in the sensitivity/adjoint circuit, the impulse conductance of a PWL two terminal device in the derivative space can be written as

$$\frac{dG}{d\tau} = \frac{\partial G}{\partial \hat{v}}\frac{d\hat{v}}{d\tau} = \left\{ \sum_{k=-\infty}^{\infty} (\Delta G_k)\delta(\hat{v} - V_k) \right\}\dot{\hat{v}}(\tau)$$
$$= \left\{ \sum_{k=-\infty}^{\infty} (\Delta G_k)\delta[\dot{\hat{v}}(\tau)(\tau - \tau_k)] \right\}\dot{\hat{v}}(\tau) = \sum_{k=-\infty}^{\infty} (\Delta G_k)\delta(\tau - \tau_k) \quad (6)$$

where $\hat{v} - V_k = \dot{\hat{v}}(\tau)(\tau - \tau_k)$ with τ_k being the event time when the breakpoint V_k is reached during the simulation, and $\delta[\dot{\hat{v}}(\tau)(\tau - \tau_k)] = [\delta(\tau - \tau_k)]/\dot{\hat{v}}(\tau)$ by the scaling property of the impulse function. In other words, $(dG)/(d\tau)$ is a train of impulses, which is zero everywhere except at the event times τ_k 's corresponding to the element reaching one of its breakpoints. This derivation can be extended in a straightforward manner to multi-terminal PWL elements. For multi-terminal elements, the conductance matrix for the adjoint circuit is the transpose of the original conductance matrix.

3.Results

First a ring oscillator consisting of 5 NAND gates was selected to check the accuracy of the gradient computation. Repeated ACES simulations were performed and the delays of the ring oscillator are measured as the width of a chosen transistor was varied over a range of values. Then the sensitivities of the delays at a selected subset of these data points were obtained using ACES adjoint computation. The gradient lines using the computed sensitivities at these selected data points were generated and plotted against the delay data as a sanity check for the gradient computation. The results of the experiment are shown in Fig. 4. The delay data were obtained by varying a transistor

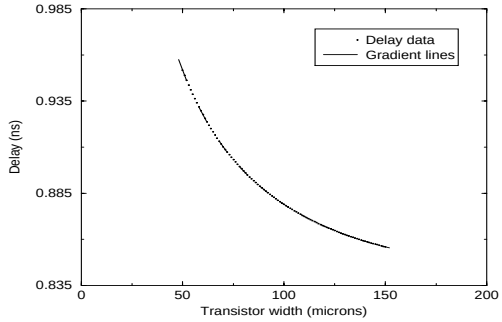

Fig. 4*: Plot of gradient lines obtained by ACES transient computation at various transistor widths versus the delay data over a range of transistor widths*

width from 50 microns to 150 microns in increments of 1 micron. The gradient lines were computed at an increments of 10 microns. These results demonstrate the excellent accuracy of the ACES gradient computation.

Next a number of realistic test circuits are used to assess the performance of the adjoint analysis in ACES. One adjoint analysis was performed for each circuit. Table 1 summarizes the run times of the ACES simulation and the associated adjoint simulation as well as direct simulation. Column 1 of the table lists the circuit names. Column 2 lists the number of MOSFETs in each circuit. Column 3 lists the run times for the original simulation. Column 4 lists the run times for the simulation of the adjoint circuit for one performance function with respect to one sensitivity parameter (one transistor width in this case). Column 5 lists the run times for the direct simulation in computing the sensitivity of one performance function with respect to one

Table 1: Simulation run times for the adjoint and direct methods in computing the sensitivity of one circuit function with respect to one sensitivity parameter

Circuit	No. of FETs	Orig. Sim.	Adjoint sim.	Direct sim.
XOR	20	0.29s	0.04s	0.06s
Error control	104	0.86s	0.07s	0.07s
Neural net	184	2.7s	0.23s	0.25s
Counter	220	2.91s	0.31s	0.32s
Controller	282	2.62s	0.24s	0.18s
Critical path	428	14.9s	0.68s	0.65s
Register	468	23.3s	3.18s	2.83s
Adder	866	20.5s	3.05s	3.82s
Zero leading counter	1513	30.4s	3.69s	4.39s
32bit error checking	3328	50.1s	11.8s	10.2s
ALU	44568	741.4s	63.7s	74.9s

width parameter. On the average, the cost of one adjoint or direct simulation varies between 10%-20% of the cost of the original simulation. In order to assess the cost of the adjoint methods as a function of the number of performance functions and the direct method as a function of the number of parameters, the counter circuit of 220 MOSFETs was used in computing the sensitivities of different number of functions with respect to a set of transistor widths. The results of this experiment are summarized in Fig. 5. For the direct method,

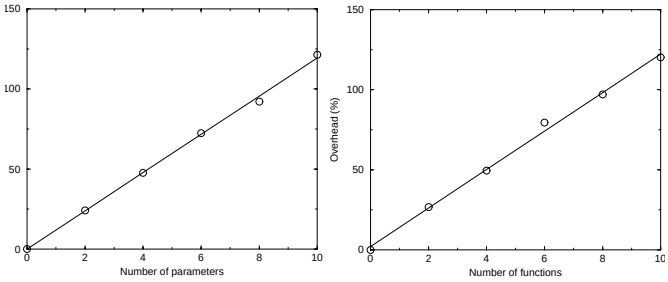

Fig. 5*: The overhead of the direct and adjoint methods over the original simulation as a function of the number of sensitivity parameters (for the direct method) and the number of the sensitivity performance functions (for the adjoint method)*

one performance function is chosen and the number of parameters is varied from 2 to 10 in increments of 2. For the adjoint method, one sensitivity parameter is chosen, and the number of sensitivity functions is varied from 2 to 10 in increments of 2. A linear least square fit is performed for each set of data points. As expected, the cost of the direct method varies linearly with the number of parameters and the cost of the adjoint method varies linearly with the number of sensitivity functions. In this particular example, the cost is about 12%

per parameter for the direct method and 12% per function for the adjoint method.

3.1. Timing abstraction for Transistor Level Timing Analysis

One of the applications where transient sensitivity computation can be useful is in timing abstraction where the net delay and output slew are represented as linear approximations as a function of load capacitance and input slew:

$$D = D_0 + (C - C_0)\frac{\partial D}{\partial C} + (S_{in} - S_{in0})\frac{\partial D}{\partial S_{in}}$$

$$S_{out} = S_{in0} + (C - C_0)\frac{\partial S_{out}}{\partial C} + (S_{in} - S_{in0})\frac{\partial S_{out}}{\partial S_{in}}$$

where D is the delay, C is the load capacitance, S_{in} is the input slew, and S_{out} is the output slew. D_0, C_0, S_{in0}, and S_{out0} have the same meanings except that these correspond to the nominal values for which the abstraction is performed. In cases where multiple inputs are required to generate an output transition, the above relations would sum the contributions of all the inputs. For complex logic circuits, this scheme can require a large number of sensitivities. The availability of the transient sensitivity computation described in the previous sections greatly improves the efficiency of the timing abstraction process, especially for transistor level timing analysis. In fact, ACES and the associated transient sensitivity computation has been used as the core simulation engine for a block based transistor level analyzer.

3.2. Transistor-level static circuit tuning

A useful application of transient circuit sensitivities is optimizing the performance of custom designs by sizing transistor widths. A well-known static transistor sizer is TILOS [8]. The basic idea of the TILOS optimization algorithm is, at each iteration, to identify and increase the width of the most sensitive (Δdelay/Δwidth) transistor on the most critical timing path. In TILOS, transistor networks are modeled by equivalent linear RC circuits, and a reduced-order Elmore [9] delay model is used to compute delays. Use of a simplified delay model provides some advantages to TILOS: the optimization problem becomes convex (there are no local optima), and the sensitivity of delay with respect to transistor widths is computed analytically. However, a significant disadvantage is the inaccuracy of the model.

We have implemented a static transistor sizer which is similar to TILOS, but which uses a more accurate delay model based on transient simulation with ACES. Convexity of the optimization problem is sacrificed in order to obtain more accurate delays, thus strict improvement at each iteration is not enforced. We have successfully optimized over a dozen circuits ranging in size from just 10 to over 4000 gates. Improvements in the critical path delay (at constant area) range from 2% to over 30% depending on the quality of the original design point, but at least some improvement was always obtained.

For example, an "area versus delay" trade-off curve generated for a 56-bit comparator containing 430 gates is shown in Fig. 6. The horizontal axis is critical path delay, and the vertical axis is total layout area. Our sizing starts at the lower-right end of the trade-off curve, and it progresses to the upper-left end of the curve by identifying and increasing the widths of the most sensitive devices located on critical timing paths. For this example, our sizing stops when the original area is recovered, and the critical path delay has been improved by 31%. Qualitatively, the importance of sensitivity calculation to the sizing algorithm may be observed by noticing how the slope of the trade-off curve starts out nearly horizontal (delay improvements relatively inexpensive in terms of area) and becomes more vertical (delay improvements relatively expensive in terms of area) as the sizing progresses.

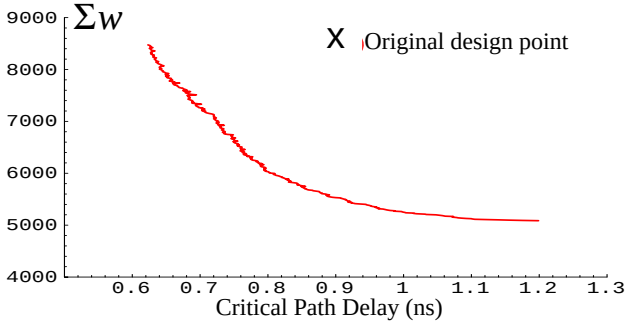

Fig. 6: *"Area versus delay" trade-off for a 56 bit comparator*

4.Conclusions

A general method for computing transient sensitivities using both the direct and adjoint methods in event driven simulation algorithms employing piecewise linear models such as ACES has been presented. Accuracy and efficiency of the algorithms has been demonstrated on various examples. The cost of the adjoint and direct simulation is about 10%-20% of the original nominal simulation. Applications in transistor level timing analysis and circuit tuning using transient sensitivity information have also been presented.

Acknowledgments

The authors would like to thank David LaPotin for his support and Anirudh Devgan for many useful discussions during the course of this work. We also would like to thank Alex Suess for his help with the implementation of the transistor level circuit tuner.

References

[1] A. Devgan and R. A. Rohrer, "Adaptively Controlled Explicit Simulation," *IEEE Trans. Computer-Aided Design*, Val. 13, pp.746-762, June 1994.

[2] S. W. Director and R. A. Rohrer, "The Generalized Adjoint Network and Network Sensitivities," *IEEE Tran. Circuit Theory*, vol.16, pp. 318-323, August 1969.

[3] D. A. Hocevar, P. Yang, T. N. Trick, and B. D. Epler, "Transient Sensitivity Computation for MOSFET Circuits," *IEEE Trans. Computer-Aided Design*, vol. 4, pp. 609-620, Oct. 1985.

[4] P. Feldmann, T. V. Nguyen, S. W. Director, and R. A. Rohrer, "Sensitivity Computation in Piecewise Approximate Circuit Simulation," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 171-183, Feb. 1991.

[5] C-J Chen and W-S Feng, "Relaxation-Based Transient Sensitivity Computation for MOSFET Circuits," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 173-185, Feb. 1995.

[6] T. V. Nguyen, P. Feldmann, S. W. Director, and R. A. Rohrer, "SPECS Simulation Validation with Efficient Transient Sensitivity Computation," *Proc. ICCAD*, Nov. 1989, pp. 252-255.

[7] T. V. Nguyen, A. Devgan, and O. J. Nastov, "Adjoint Transient Sensitivity Computation in Piecewise Linear Simulation," DAC 1998, pp.477-482

[8] J. P. Fishburn and A. E. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing," *IEEE International Conf. on Computer-Aided Design*, Nov. 1985, pp. 326-328.

[9] W. C. Elmore, "The Transient Analysis of Damped Linear Networks with Particular Regard to Wideband Amplifiers," *Jour. of Applied Physics*, Vol. 19, No. 1, pp. 55-63, 1948.