B8.

AN EFFICIENT FEEDBACK PROCESSING METHOD FOR RELAXATION BASED FAST TIMING SIMULATION

Lei He, Kaihe Zhang, Pushan Tang

CAD Lab of E. E. Dept., Fudan University
Shanghai, China (200433)

Abstract An *event-driven self-adaptive window technique* for processing feedback problem in waveform relaxation based fast timing simulation is presented. A J-K Flip-Flop and several different kinds of counters are extensively tested to compare its performance with other methods. Also, its application in both switch level timing simulation and analog/digital mixed mode simulation is discussed.

Introduction

The aim of fast timing simulation is to generate time response of circuits with less detail and at higher speed than circuit simulation. This is achieved by partitioning the circuit into an interconnection of subcircuits and performing waveform calculation for each subcircuit in certain order. For MOS digital circuits, these subcircuits are often DC-Connected-Components (DCCs), and voltage-time equations which preserve the nonlinearities of transistors can be used [1]. In solving equations, waveform is descritized by preset voltage levels then time points when waveform reaches a discrete voltage level are computed. The entire analysis time is divided into time phases for each subcircuit. Waveform calculation is performed in each phase.

When feedback loops exist at the DCC level, the maximum set of all DCCs which have feedback loops between them is called as a Strongly-Connect-Component (SCC). A DCC which is not in any feedback loop is also viewed as a simple SCC. Then all SCC can be levelized to get an order for waveform calculation by one-way circuit technique in [2].

This scheme has been used in both switch level timing simulators [1,3] and analog/digital mixed mode timing simulators [4,5]. How to process feedback loops inside each SCC has very important influence on efficiency of this scheme, and this paper is focused on this problem. Section 2 reviews existing *dynamic window technique* [1] and *multi-level algorithm* [6]. Then, a new *event-driven self-adaptive window algorithm* is proposed in Section 3. Experimental results and performance comparison of different methods are given in Section 4, conclusions are drawn in Section 5.

* This project is supported by State Science & Technology Commission of China under Contract 85-703-01-06.

Previous Work

A *dynamic window technique* [1] is developed to process feedback loops inside SCC. A window is defined around each input waveform transition for a SCC, and several such windows may be merged into a single one if these windows overlap. All active DCCs in the SCC are analyzed during the first window, output waveforms are used to create new windows. DCC analysis is completed when all waveforms have converged within the first window. Then, the processing of the first window is finished and proceed to the second window, ..., . Each waveform transition is entirely in one of such windows, so none time phase will be split by window edges in waveform calculation.

The *dynamic window technique* is only efficient for small SCCs or non-hierarchical feedback loops, a *multi-level algorithm* is developed to enhance this technique when nested feedback loops exist in a SCC [4,6]. Two types of feedback loops are defined in this algorithm. If the number of DCCs in a feedback loop is less than a preset size, the feedback loop is defined as a strong feedback loop, otherwise, it is a weak feedback loop. Strong feedback loops may be nested inside a weak feedback. A weak feedback loop is processed by basic waveform relaxation with consideration of partial waveform convergence. During each iteration over the weak feedback loop, the strong feedbacks are processed using the *dynamic window technique* to obtain the latest version of its circuit waveform. Experimental results in [6] show that the *multi-level algorithm* is much more efficient than the *dynamic window technique* in a circuit containing two level feedback loops.

Event-driven Self-adaptive Window

There are several weaknesses in the *dynamic window technique* and the *multi-level algorithm*.

1) Partial waveform convergence [7] is not exploited in windows. Partial waveform convergence is referred to that input waveforms of a DCC do not differ significantly in the last two iterations during certain time interval. If it occurs, the analysis of the DCC can be bypassed in the time interval. Although partial waveform convergence is exploited when processing weak feedback loop in the *multi-level algorithm*, it is not exploited in the *dynamic window technique*.

2) Windows are easy to overlap resulting in increased window size. As the number of iterations is exponentially proportional to the size of a window [7], the larger the window size, the less efficient the *dynamic window technique*. This algorithm degradation may be happened in the following cases:

(i) when a SCC has complicated feedback loops or just "too many" DCCs, waveform transitions tend to overlap with each other;

(ii) in some circuits where the input transitions to a DCC always overlaps with the output transitions, then a window will constantly extend and eventually cover the entire analysis time. All benefits of window techniques will disappear.

In the *multi-level algorithm*, window extension is alleviated because the *dynamic window technique* is only used in processing strong feedback loops with only several DCCs. However, no windowing benefit will be obtained because the entire analysis time is used to process the weak feedback loop.

3) The ordering problem is always a difficult problem in the waveform relaxation based circuit simulation. A bad order causes non-necessary iterations so slow down the convergence of waveform. In the *multi-level algorithm*, iterations in strong feedback loops will be wasted if input waveforms to the strong feedback loops are not convergent in the weak feedback loop. To find *a priori* order is impossible, because the best order depends on both the circuit structure and dynamic waveform in circuits.

A new method called *event-driven self-adaptive window technique* is proposed in this paper. An event-queue is maintained for each SCC in order to find the best order for analysis. A waveform transition is defined as an event, all events are put into a queue in time-ascending order where time refers to the point when a transition begins. All events are processed successively. DCCs in the SCC driven by an event are analyzed, output waveforms which diverge from waveforms in the last iteration will create new events in the event-queue. Analysis of a SCC ends when there is none event in the queue or the time of the current event is later than the end point of the entire analysis time.

An event in processing provides all DCCs driven by the event with an initial window, which starts from the beginning time of the event and with a preset window size. The initial window will be independently adjusted for each DCC by the *self-adaptive window technique* given in Figure 1. First, window is adjusted to start from the time point when the DCC input waveforms of the last two iterations diverge. If the time interval between the new start time and the beginning time of the event is larger than the preset window size, analysis of the DCC is bypassed. Otherwise, analysis begins from the new start time and with the preset window size. In this method, partial waveform convergence is fully exploited.

Finally, the modified window is further adjusted by extending itself forward or backward, so that none time phase would be split by edges of the window and the accuracy of waveform calculation is guaranteed. Different from conventional event driven method, if a window for a DCC covers several events, all these events will be processed for this DCC.

Because windows in the *self-adaptive window technique* is dependent on each DCC, it is different from the dynamic window technique where the window is dependent on all DCCs in a SCC. So window size is no longer extended.

Examples

Both the two windowing techniques are implemented in a switch-level fast-timing simulator FTSIM [2]. A CMOS ring oscillator containing three inverters is first analyzed to see the performance of the *event-driven self-adaptive window technique*. Three schemes are used, and CPU time together with numbers of waveform calculation are listed in Table 1. In Table 1, BASIC stands for basic waveform relaxation without windowing technique, PWC stands for basic waveform relaxation considering partial waveform convergence for each DCC, and SW stands for the *event-driven self-adaptive window technique*. The SW method is 10 times faster than the BASIC method and twice faster than the PWC method. Actually, the PWC method can also be viewed as the *self-adaptive window technique* without event-driven method in this example. It costs twice calculation than the SW method just because a waveform is analyzed at least twice in waveform relaxation. Besides, it is observed that waveforms produced by the three schemes are indispensable.

Table 1 Analysis Time for Ring Oscillator

BASIC	PWC	SW
3.98s (1192)	0.76s (203)	0.62s (111)

Next, a J-K Flip-Flop is also analyzed by three schemes and testing results are listed in Table 2. The DW method in Table 2 stands for the *dynamic window technique* implemented by ourselves. The SW method is nearly 50 times faster than the DW method, which already degrades into the BASIC method in this circuit and is even slower than the BASIC method because of its implementation overhead. Waveforms produced by the three schemes are also indispensable. Waveform comparison between SPICE3c1 and FTSIM is shown in Figure 2.

Table 2 Analysis Time for J-K Flip-Flop

BASIC	DW	SW
52.37s (11008)	60.65s (10571)	1.26s (238)

All DCCs of a J-K Flip-Flop are in a SCC, whose feedback loops are shown in Figure 3. Several counters implemented by J-K Flip-Flops are finally analyzed. These circuits are: a 2-bit counter which is a J-K Flip-Flop containing 37 transistors; a synchronous 3-bit counter which contains 2 J-K Flip-Flops in a SCC; an asynchronous 4-bit counter which contains 2 J-K Flip-Flops in different SCCs; a synchronous 10-bit counter which contains 4 J-K Flip-Flops, one in a SCC, the other three in another SCC. Performance comparison between the DW method and the SW method is no longer necessary, because feedback loops in these circuits are much more complicated than those in a J-K Flip-Flop. Testing results are listed in Table 3, and waveforms of the 10-bit counter are shown in Figure 4.

Table 3 Analysis Time for Counters

	2 bit	3 bit	4 bit	10 bit
SW	2.93s (544)	5.71s (1076)	6.35s (801 +347)	16.68s (2467 +1411)
DW	121.31s (21142)	**	**	**
SPICE3	183.34s	382.04s	403.68s	**

Conclusions

An event-driven self-adaptive window technique is proposed in this paper. It has been implemented in a switch level fast timing simulator FTSIM. The new technique overcomes the main weaknesses of the dynamic window technique. Ring oscillator and several different counters implemented by J-K Flip-Flops are analyzed, more than 10 times speedup is observed in these circuits. The new technique has also been used in an analog/digital mixed-mode simulator [5], where analysis methods vary in different subcircuits. Besides, this method can be used in other waveform relaxation based timing simulation.

References

[1] V. Rao, D. Overhauser, I. Hajj and T. Trick, *Switch-Level Timing Simulation of MOS VLSI Circuits*, Boston: Kluwer Academic Publish, 1989.

[2] A.E. Ruehli, *Circuit Analysis, Simulation and Design*, New York: North-Holland Amsterdam, ch. 6, 1987.

[3] Lei He, Kaihe Zhang, Pushan Tang, "FTSIM -- A Switch-Level Fast-Timing simulator," submitted to *Acta Electronica Sinica.*

[4] D. Overhauser and I. Hajj, "IDSIM2: An Environment for Mixed-Mode Simulation," *Proc. 1990 CICC*, pp. 5.2.1 - pp. 5.2.4.

[5] Lei he, Song Chen, Kaihe Zhang and Pushan Tang, "Implement Analog/Digital Mixed Mode Simulation in SPICE," to be presented on the International Conference on CAD/Graphics, Beijing, China, Sept., 1993.

[6] D. Overhauser and I. Hajj, "Multi-Level Circuit Partitioning for Switch-Level Timing Simulation," *Proc. IEEE ISCAS-1988*, pp.1361-1364.

[7] A.R. Newton, A.L. Sangiovanni-Vincentelli, "Relaxation-Based Electrical Simulation," *IEEE Tran. on CAD*, vol. 3, No. 4, 1984.

```
PROCEDURE      SelfAdaptWindow
INPUT          DCC, InitialWindowStartTime,InitialWindowLength
OUTPUT         FinalWindowStartTime, FinalWindowEndTime
BEGIN
  InitialWindowEndTime = InitialWindowStartTime + InitialWindowLength
  IF      DCC has not been analyzed
  THEN    NewWindowStartTime = InitialWindowStartTime
  ELSE    NewWindowStartTime = the time point when the DCC input waveforms in the
          last two iterations diverge
  IF      NewWindowStartTime - InitialWindowStartTime > InitialWindowLength
  THEN{   FinalWindowStartTime = NewWindowStartTime
          FinalWindowEndTime = NewWindowStartTime
     }        /* the DCC is bypassed in this scheduling */
  ELSE{   NewWindowEndTime = NewWindowStartTime + InitialWindowLength
  IF      NewWindowStartTime is a time point which exactly separates two time phases
  THEN    FinalWindowStartTime = NewWindowStartTime
  ELSE    FinalWindowStartTime = the begin point of the time phase in which
          NewWindowStartTime is located
  IF      NewWindowEndTime is a time point which exactly separates two time phases
  THEN    FinalWindowEndTime = NewWindowEndTime
  ELSE    FinalWindowEndTime = the end point of the time phase in which
          NewWindowStartTime is located
     }
END
```

Figure 1 Self-Adaptive Window for Fast Timing Simulation

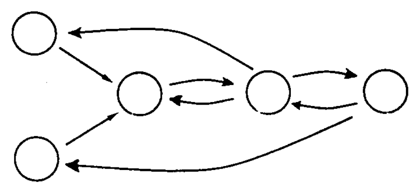

Figure 3 Feedback Loops in a J-K Flip-Flop

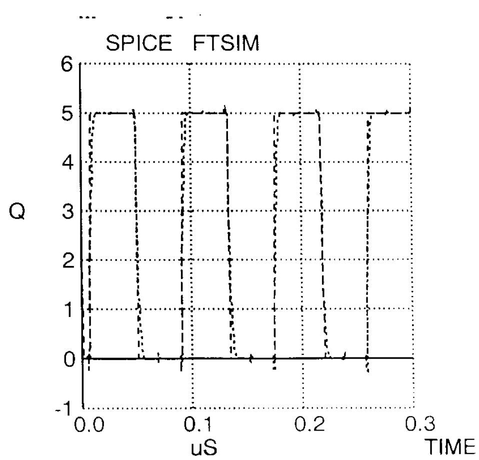

Figure 2 Waveform Comparison Between FTSIM & SPICE

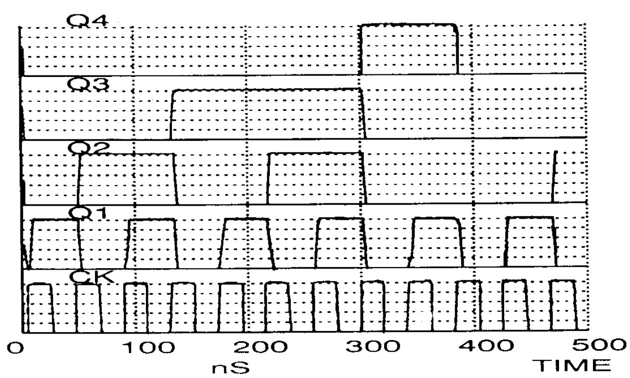

Figure 4 Waveforms in 10-bit Counter