IMPLEMENTED TECHNIQUES FOR HANDLING SPIKES IN AN ASSIGNABLE DELAY SIMULATOR

S. A. Szygenda,[1] A. Lekkos,[2]

and J. Fike[3]

[1]The University of Texas

[2]Bell Telephone Labs

[3]Southern Methodist University

INTRODUCTION

The needs for Digital Logic Simulation to analyze design timing problems, such as spike and hazards, has been shown previously (1). In this paper, methods for detecting timing problems are investigated and analyzed. A technique is developed to perform spike and hazard analysis in a less pessimistic approach than has been used previously. It was implemented as an extension of the TEGAS2 simulation system (2) and is capable of associating different turn-on and turn-off propagation delays with an element.

In a formal sense, a hazard can be defined as the situation in which two or more inputs to a particular gate arrive at slightly different times, causing an anomaly in the output of the gate. A spike is the result of an attempt to change the output of a gate at a rate faster than its turn-on or turn-off propagation delay time. Turn-on delay is the time region associated with the signal going from 0 to 1, where turn-off delay is the period associated with an output transition from 1 to 0.

As larger and more complex digital circuits are designed, it is becoming increasingly necessary to analyze timing problems with reasonable computer time usage. A technique to perform such an analysis requiring economical quantities of computer time was investigated, designed and implemented as an extension of the TEGAS2 simulation system.

MODES AND TIME-FLOW OF THE TEGAS2 SYSTEM

The TEGAS2 system is a multi-mode simulator. It allows the simulation of a fault-free circuit (good machine) at various levels of detail: logic verification (Mode 1), three-value simulation (Mode 2) and design verification (Mode 3). TEGAS2 also allows the simulation of traditional stuck-at faults (stuck-at-one, stuck-at-zero) as well as other types of faults such as, complex, intermittent, and multiple faults. It is a table-driven simulator which makes extensive use of list-processing and dynamic storage allocation techniques. Gate elements can be assigned up to six different propagation delay values: these correspond to nominal, minimum and maximum delays for signal turn-on and turn-off. The significant event or selective trace signal propagation technique is used; when an element's value does not change, signal propagation does not occur. Hence an element is not re-evaluated if all input signals are the same as they were then the element was last evaluated.

The system employs a combination of next-event scheduling with a fixed time increment technique. When an element is re-evaluated due to changes on its input(s), activity is taking place in the simulation process. Assuming that changes on input(s) took place at time i and the element propagation delay is n, then at time i+n the element will (possibly) be changing to a new value. If the time of the change is within the fixed time increment range, the fixed time increment technique is used and the element is assigned the value at time i+n.

However, if time i+n is outside the fixed time range, then the next event technique is employed and the element is inserted in the next-event queue.

A detailed description of the system and its techniques is given in (2) and (3).

TECHNIQUES FOR DETECTING SPIKES AND HAZARDS IN NOMINAL DELAY SIMULATION

At simulation initialization time, a portion of the memory array is allocated to store the current values (CV) and the possible unknown state of the elements being simulated. Since this space is allocated dynamically, the function CV is used to extract either the true or indeterminate logical value of the element. Every time an element is evaluated and scheduled, either in the fixed time increment or next event chains, the real time the element will be settling to the proposed value is stored in the unused 18-left-most-bits of "array" CV. If for element i, the 18 left-most bits (CVTIME) of CV(i) are not zero, then element i is in transition between states. If (CVTIME) is zero, the element has settled to a defined discrete value.

Suppose an element is activated due to some signal propagation. The element is evaluated, and its corresponding entry in CV is tested. Assume entry CVTIME is not zero. Hence, the element is in a transition state. Since the CVTIME entry points

either to the fixed time increment or next-event chain, the scheduled node is located within one of those chains. The proposed final value and the new value are tested; if they are the same due to some hazard, element propagation is forced. Since these values are the same, there is no spike situation and the newest value is not scheduled. However, if the new and proposed values differ, there may be a spike. For Mode 1 spike, the selective trace option should be tested. In case the option was not requested, then the element being scheduled may not be in a transition state, since the signal value is propagated even if no changes occurred.

In order to make this point clear, consider an element having a current value of zero, which is scheduled to go to zero. If, due to a hazard, it is evaluated again and ready to be scheduled to go to one, this would appear to be a spike. Since the current value of the element is zero, and it is scheduled to "change" to zero, there is actually no transition in the signal, so the latest hazard does not cause any anomaly (spike) in the output of the element. However, if the current value of the element is not the same as the scheduled value, then the element is in a transition state and a spike is detected. If the selective trace is on, then the spike is found earlier, since the element value is not propagated unless it is different from its current value.

In mode 2 simulation, the value of the scheduled output is given the unknown status (X) and the proposed value is scheduled.

Finally, if the CVTIME entry of element i is zero, then no activity is present for the element. If the proposed value differs from the current value of the element or if selective trace is off, then the proposed new value for the element must be scheduled.

SPIKE AND HAZARD ANALYSIS FOR AMBIGUITY REGION SIMULATION

As defined previously, simulation mode 3 of the TEGAS2 system provides a means of simulating the ambiguity region of each signal. This region is specified by the minimum and maximum delays of the gate. For example, if a gate has a minimum delay of 2 units and a maximum delay of 7 units, its ambiguity region is said to be 5 units. If an input signal causes a change in the output of this gate, then the output will begin to change no earlier than 2 units after the input can begin to change, and will complete its change no later than 7 units after the latest time that the input can complete its change. The term "ambiguity region" is somewhat misleading, since unless other inputs to the gate are also changing, or unless some inputs exhibit a potential error, the only ambiguity associated with the output is that the exact time at which it begins to change, or completes its change, is not known. What is known, however, is the direction of change. If the initial value of the output is 0, and the final value is 1, then during the ambiguity region the signal values may be represented by "U", while in the opposite case, the signal may be represented by "D".

For example consider a two-input AND gate which is driven by a signal which goes from one to zero and back to one, remaining at zero for a time period shorter than the minimum delay of the gate. This causes scheduling of two successive changes in the output or transition (NV) portion of F's signal, followed by two successive changes in the settling (CV) portion. The result (unfiltered output for gate F) is that the gate output signal takes on a sequence of values which is impossible in the actual circuit.

In order to eliminate the occurrence of these and other impossible signal changes, a "filter" has been inserted in the signal update loop of the SIMM3 routine. The filter routine checks the updated signal value (from the time queue) against the previous values (from the CV, CV2 and CV3 arrays). Impossible signal changes are set to "PE"; normal changes are allowed to occur. These rules are given by the following matrix.

Previous Value \ Updated Value	0	1	U	D	E
0	0	1*	U	(E)	E
1	0*	1	(E)	D	E
U	(E)	1	U	(E)	E
D	0	(E)	(E)	D	E
E	0	1	U	D	E

(E) PE bits inserted "Filter"

* "Instantaneous" changes from 0 to 1 or from 1 to 0 are undesirable, but allowed.

The problem of adding differing rise and fall times in mode 3 is somewhat more complicated than for modes 1 and 2, due to the ambiguity region just described. Accordingly, a set of rules was developed to handle these conditions.

CONCLUSION

This paper has attempted to describe spike and hazard problems that are encountered in time delay logic simulation. The problems are subtle and need be thoroughly understood before one can devise solutions. Efficient solutions are even more elusive than the problems. The techniques described in this paper have been implemented and appear to offer efficient solutions to the problems.

REFERENCES

1. Szygenda, S.A., Rouse D., and Thompson, E.W. (1970b) "A Model and Implementation of a Universal Time Delay Simulator for Large Digital Nets", AFIPS Proc. of the May SJCC, pp. 207-216.

2. S. A. Szygenda, "TEGAS2--Anatomy of a General Purpose Test Generation and Simulation System for Digital Logic", Proc. of 9th Annual Design Automation Workshop, June 1972.

3. S.A. Szygenda, C.W. Hemming and J.M. Hemphill, "Time Flow Mechanisms for Use in Digital Logic Simulation", Proc. of the 5th Annual Conference on Applications of Simulation, December, 1971.