Distributed Logic Simulation: Time-First Evaluation Vs Event-Driven Algorithms

S. Sundaram

Supercomputer Education and Research Center
Indian Institute of Science
Bangalore 560 012, India

L.M. Patnaik

Microprocessor Application Laboratory
Indian Institute of Science
Bangalore 560 012, India

Abstract

With the increasing complexity of VLSI circuits, simulation of digital circuits is becoming a more complex and time-consuming task. General purpose parallel processing machines are increasingly being used to speed up a variety of VLSI CAD applications. All previous works on mapping sequential logic simulation algorithms onto general purpose parallel machines were centered around using Event-Driven algorithm, and do not satisfactorily address the suitability of a particular sequential algorithm for parallel implementation. In this paper, we present analysis of two distributed simulation algorithms: the centralized-time Event-Driven algorithm and the Time-First evaluation algorithm, mapped onto a network of workstations. We present results over a wide range of ISCAS85 and ISCAS89 benchmark circuits, to show that the Time-First evaluation algorithm is likely to be a viable alternative to the event-driven algorithm in the domain of parallel logic simulation

1 Introduction

Logic simulation plays an important role as a verification tool in the process of VLSI design. Most widely used logic simulators employ the algorithm based on (1) Compiled code[3] (2) Event-driven[3] and (3) T-Algorithm[4]. Current trends in carrying out parallel logic simulation are concentrated towards mapping the simulation algorithm onto the processors of general purpose parallel machines. Most of the efforts concentrated towards mapping logic simulation algorithms on to general purpose parallel machines using event driven algorithm, and exploiting the concurrency inherent in the circuit being simulated(data parallelism) have come out with very little success[1, 7]. The speed up obtained on parallel/distributed simulation, depends on two major factors; the amount of simulation task concurrency possible, and the amount of messages that must be passed among processors. The time taken for simulation, the amount of communication overhead, and the load balancing are chosen as a measure to compare and address the suitability of sequential algorithms for parallel implementation. In this paper, we take a broader view, and compare the advantages and limitations of the Time-First evaluation and Event-Driven algorithms, in uniprocessor and multiprocessor environments.

The paper is organized as follows. The next section provides a detailed comparison of the sequential versions of the Event-Driven and Time-First evaluation algorithms, and quantifies the algorithms by simulating the ISCAS85 and ISCAS89 benchmark circuits on a uniprocessor. In section 3 we present a comparison of the amount of concurrency and the communication overhead of these two algorithms, on a parallel/distributed environment. The last section concludes the paper with a summary, that the Time-First evaluation algorithm performs better than the Event-Driven algorithm in the domain of parallel logic simulation.

2 Time - First Evaluation Vs Event-Driven Algorithms(Sequential Version)

The sequential Time-First evaluation algorithm (T-Algorithm) is compared with the conventional event-driven algorithm in the following subsections to bring out the advantages of using the T-Algorithm over the Event-Driven algorithm.

One of the basic advantages of the T-algorithm is that table lookup procedures are needed only once per each gate, independent of the number of events occurring at the input of a gate. Thus for longer the event sequence processed at any instant, smaller is the number of table lookup operations. Secondly, the operation of insertion of new events(scheduling) of the gate,

1063-9667/95 $04.00 © 1995 IEEE

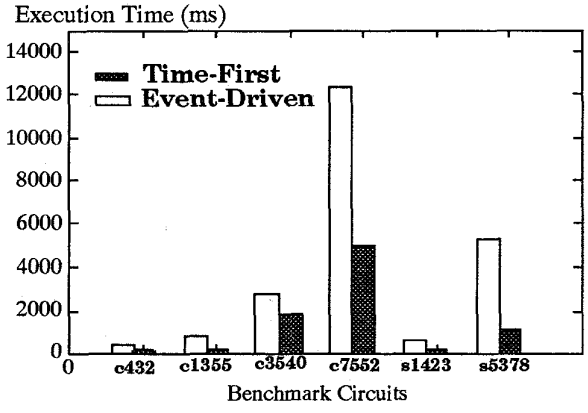

Figure 1: Comparison of Speeds of T-Algorithm and Event-Driven Algorithm

Table 1: Estimation of Memory Requirement

Circuit	*Number of Gates*	*Percentage Adjacency*
c432	160	58
c1355	546	77
c3540	1669	57
c7552	3512	47
s1423	657	64
s5378	2779	75

Table 2: Evaluation Time for Synchronous Sequential Circuits

Circuit	*Preprocessing Time(ms)*	*T-Algorithm Evaluation Time(ms)*	*Event-Driven Evaluation Time(ms)*
s1423	90	204	561
s5378	280	1148	5252

is a simple addition of an event at the end of a linked list. Since we use rise-fall delay model[3] for computation of the output of a gate, event cancelation operation amounts to only a deletion of an event from a linked list. Because of these reasons, the T-Algorithm runs about seven times faster than the event-driven algorithm[4] for combinational circuits. In our study, we have measured the speedup in simulation obtained on the ISCAS benchmark circuits, and have found that the T-algorithm runs 3-5 times faster compared to the event-driven algorithm, as shown in Figure 1.

We can minimize the memory requirement, using a heuristic method which controls the processing order of gates[4], and by releasing the memory storing the events which are not required for future evaluations of gates. The number of gates, whose outputs are connected to the next level gates, can be used as a measure to estimate the memory requirement. Higher the percentage of the gates, whose outputs are connected to the next level, smaller is the memory requirement. The tabulated values for the chosen ISCAS85 and ISCAS89 benchmark circuits are shown in Table 1.

If a simulated circuit includes short feedback loops, the short loops have to be replaced by functional blocks with memory. In the case of sequential circuits, as suggested in[2], a preprocessing step of zero delay event-driven simulation is required to calculate the future states of the circuit for the entire simulation period. Once the future states are known, the combinational portion of the circuit can be subjected to the T-Algorithm simulation for the entire simulation period. The results obtained by running the ISCAS89 benchmark circuits, are shown in Table 2. We can observe that the total simulation time in case of the T-algorithm adds up to less than that of the event-driven algorithm.

3 Time-First Evaluation Vs Event-Driven Algorithms (Distributed Version)

This section addresses the question, Time-First algorithm a more viable alternative than the Event-Driven algorithm in the domain of parallel logic simulation?. This is important to investigate since increased parallelism is the main motivation for using the Time-First evaluation algorithm.

3.1 Intrinsic Overheads

In case of the Event-Driven algorithm, there are two global synchronizations needed at each time instance, one after the wire update phase and the other one after the element evaluation phase. The Time-First evaluation algorithm requires only one synchronization point for each level of the circuit. The number of time instances in the Event-Driven algorithm is very large compared to the number of levels in a circuit in case of the T-Algorithm. Increase in the number of synchronization points, decreases the efficiency of the parallel algorithm.

In the parallel/distributed version of the Event-Driven algorithm, extra amount of element computation occurs in the wire update phase of the algorithm. The repeated presence of the same element at a time instance is the reason for the extra element computation. This happens when the logic value change occurs for different inputs of a gate. Though the element

Table 3: Concurrency of Tasks

Circuit	Event-Driven Algorithm		Time-First Evaluation Algorithm					
	Node Concurrency	Element Concurrency	Gates/Level			Cones/Level		
			Min.	Max.	Avg.	Min.	Max.	Avg.
c432	3	5	1	27	8	1	10	5
c1355	7	11	2	64	21	1	39	16
c3540	11	16	1	168	27	2	66	21
c7552	32	48	1	275	63	1	194	70
s1423	5	10	1	84	11	1	28	5
s5378	15	27	2	334	111	30	241	93

Table 4: Comparison of Communication Overheads

Circuit	Time-First Evaluation Algorithm		Event-Driven Algorithm
	Number of Gate Levels	Number of FFR Levels	Number of Times Communication Established
c432	17	13	175
c1355	24	16	321
c3540	60	27	1128
c7552	54	19	1279
s1423	59	38	368
s5378	25	9	549

is evaluated only once in the evaluation phase, the presence of the element in multiples(more than once) for computation in the wire update phase cannot be avoided. In the parallel Time-First evaluation algorithm, such unnecessary computations do not exist.

3.2 Concurrency of Tasks

The concurrency number is a measure of the available number of tasks which can be executed concurrently in a parallel machine. The intrinsic structure of the Event-Driven algorithm limits the extracted parallelism only to the events occurring at the same time instance. Therefore, from the profile of the number of events available at various time instances, we can calculate the average concurrency number for various benchmark circuits, for the parallel/distributed Event-Driven algorithm. In case of the Time-First evaluation algorithm, the number of gates or cones available at each level of the circuit is a measure of the concurrency number. Table 3 shows the average concurrency(assuming infinite number of processors) measured during the simulation of the benchmark circuits, in both the Event-Driven and Time-First evaluation algorithms.

3.3 Communication Overhead

In addition to the inherent synchronization overhead present in the algorithms, the number of times communication has to be established during the simulation, also contributes to the communication overhead. In case of parallel Event-Driven algorithm, communication is established at those time instances in the time queue, where at least a single element is available for evaluation. In the case of Time-First evaluation algorithm, the number of times communication is established is equal to the number of levels in the circuit. By comparing the number of levels in the Time-First evaluation algorithm and the number of instances in the Event-Driven algorithm, we can observe that the number of times communication is established in the Event-Driven algorithm is very large compared to that in the T-Algorithm. Therefore the communication overhead is usually high in the case of the Event-Driven algorithm. Table 4 shows the number

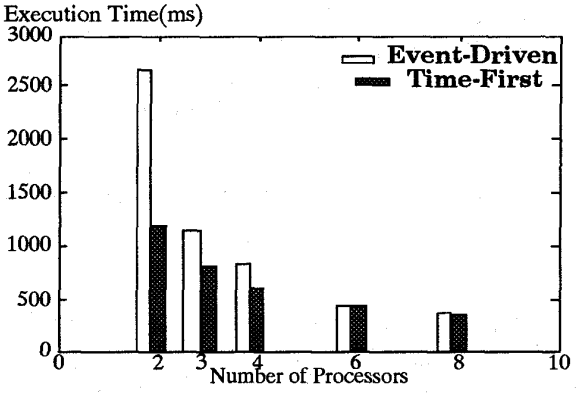

Figure 2: Evaluation Time of Event-Driven and T-Algorithm

of times communication has to be established during simulation in both the Event-Driven algorithm and the T-Algorithm.

Secondly, the type of data communicated also contributes to the communication overhead. In case of the Event-Driven algorithm, the repetition of similar data at various time instances increases the communication overhead. The inherent nature of the Time-First evaluation algorithm prevents the necessity of sending similar data to slave processors, thereby reducing the communication overhead. By using a good partitioning heuristic e.g. the FFR scheme described in[2], communication overhead can be reduced to a bare minimum.

4 Experimental Results

The performance figures obtained for the benchmark circuit c7552, using the Event-Driven algorithm and the Time-First evaluation algorithm are shown in Figures 2 and 3. From Figure 2, we can observe that the evaluation time on the Time-First evaluation algorithm is lesser in magnitude compared to the Event-Driven algorithm for the number of processors varying from 2 to 8. As the number of processors is increased, the difference in evaluation times of the al-

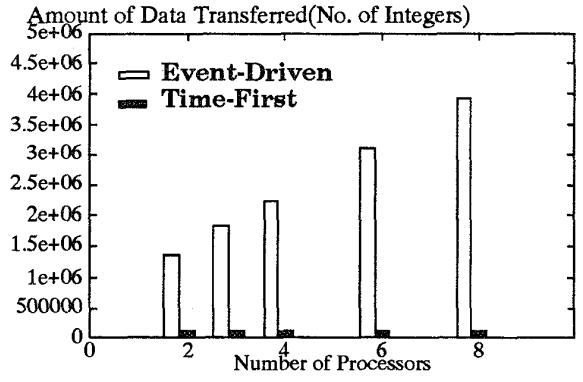

Figure 3: Communication Overhead of Event-Driven and T-Algorithms

Table 5: Evaluation Time for Sequential Circuits - Distributed Scheme

Circuit	Preprocessing Time(ms)	T-Algorithm Evaluation Time(ms)	Event-Driven Algorithm Evaluation Time(ms)
s1423	90	35	133
s5378	280	75	227

gorithms is narrowed down rapidly due to the decrease in the task granularity. The Figure 3 shows that the amount of data communicated for simulation in the Time-First algorithm is very less compared to that of the Event-Driven algorithm for the number of processors varying from 2 to 8. The results obtained by running the ISCAS89 benchmark circuits, using the distributed T-Algorithm and the Event-Driven algorithm, for an eight processor configuration are shown in Table 5. We can observe that the preprocessing time is comparatively high, and dominates the total simulation time in case of s5378 circuit. In case of the s1423 circuit the total simulation time using the T-Algorithm is less compared to the Event-Driven simulation time. We have to run for larger simulation time to predict the exact behavior of the T-Algorithm simulator to come up with a better scheme which reduces the total simulation time.

5 Conclusion

This paper presents a comparison of the performance of the Time-First evaluation algorithm and the Event-Driven algorithm, making a case that the Time-First evaluation algorithm is a viable alternative to the Event-Driven algorithm, for parallel/distributed logic simulation. The result is based on three major conclusions arrived at in this paper, (i) synchronization is only once at each level in the T-Algorithm, unlike the two global synchronizations at each time instance in the Event-Driven algorithm, (ii) the number of levels in the circuit is very less compared to the number of simulation time instances in the Event-Driven algorithm, and (iii) the concurrency number available for the Time-First evaluation algorithm is much more compared to the Event-Driven algorithm.

It is also argued that as circuits get larger and more complex in future, the number of gates available for evaluation will not show a major improvement in the Event-Driven algorithm, compared to the number of gates in a level in the Time-First evaluation algorithm, is bound to increase rapidly. Future work on simulation of sequential circuits, needs to look at running distributed zero delay Event-Driven simulation followed by the distributed Time-First evaluation algorithm. These reasons lead to the conclusion that the Time-First evaluation algorithm is likely to be a viable alternative to the Event-Driven algorithm in the domain of parallel logic simulation.

References

[1] L.P. Soule, *Parallel Logic Simulation: An Evaluation of Centralized Time and Distributed Time Algorithms*, Technical Report CSL-TR-92-527, Department of Electrical Engineering and Computer Science, Stanford University, California, June 1992.

[2] S. Sundaram, *Distributed Logic Simulation: Performance Studies of Event-Driven and Time-First Evaluation Algorithms*, Ph.D Thesis, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, March 1995.

[3] M.A. Breuer, and A.D. Friedman, *Diagonsis and Reliable Design of Digital Systems*, Computer Science Press, Rockville, 1976.

[4] N. Ishiura, H. Yasuura, and S. Yajima, *Time First Evaluation Algorithm for High-Speed Logic Simulation*, Proceedings of the International Conference on Computer Aided Design, ICCAD84, November 1984, pp. 197-199.

[5] F. Brglez, P. Pownall, and R. Hum, *Accelerated ATPG and Fault Grading via Testability Analysis*, Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS 1985, pp. 695-698.

[6] F. Brglez, D. Bryan, and K. Kozminski, *Combinational Profiles of Sequential Benchmark Circuits*, Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS 1989, May 1989.

[7] J.V. Briner, Jr.,*LDVSIM: A Mixed Level System Simulation*, Technical Report Number: TR 89-29, Duke University, Department of Computer Science, Durham, North Carolina, 1989.