Guest Editorial

AZZEDINE BOUKERCHE

Department of Computer Science, University of North Texas, Denton, TX 76203

INTRODUCTION

As the size of VLSI circuit designs becomes very large, involving thousands of gates, simulation of the design becomes a critical part of the validation and design analysis process prior to fabrication, since probing and repairing VLSI systems is impractical. Rather than dealing with voltages and currents at signal nodes, discrete logic states are defined, and simple operations are used to determine the new logic value at each node. This gives rise to the event driven approach for efficiently simulating VLSI systems.

Logic simulators are straightforward to implement if a centralized system is used. Unfortunately, logic simulation algorithms are well known for their computational requirements, and their execution time which can far exceed even the fastest available uniprocessor machines. It is therefore natural to attempt to use multiple processors to speed up the simulation process by exploiting the features of parallel architectures, and taking advantage of the concurrency inherent in VLSI systems. There are a number of ways that parallelism can be exploited to improve logic simulator performance [1]. Three primary mechanisms have been used for parallel circuit simulation: synchronous, conservative asynchronous, and optimistic asynchronous.

In the synchronous mechanisms (also referred to as global clock algorithms), all circuits are evaluated during every time step, whether or not their inputs have been changed. Although some performance benefits have been reported with these synchronous and loosely synchronous parallel simulations, other studies have shown that these techniques do not appear to scale [1, 3]. The difficulties in these schemes include determining when all logical gates (or the processors assigned to them) have completed a time step, and what the next time step should be. Furthermore, in spite of the equal number of logical gates in each partition, the processor will have unequal amount of activity. This is because the number of active elements (with events at their input) differs from partition to partition due to random distribution of activity [1].

In contrast, some analytical studies have shown that asynchronous parallel simulation techniques have significant potential to exploit the parallelism inherent in VLSI systems, and reduce the run-time of the simulation [1, 3]. This can have a profound economic impact speeding time to market while ensuring more thorough system design testing.

In the asynchronous approach, the VLSI system is viewed as a collection of subsystems that are simulated by a set of processes that communicate by sending/receiving timestamped events.

A subsystem may represent a collection of logical elements at varying level of abstraction; *e.g.*, transistors, NOR gates, Adders, flip-flops, *etc.*... State changes such as the change in output value of individual gate, are represented by event (messages) in the simulation. The traditional event-list and global clock in the sequential simulator do not exist in the parallel counterpart. Each process has its own input event queue, and its own local clock. Coordinating event evaluations and managing these queues are necessary to ensure correct simulation. A number of asynchronous parallel simulation strategies have been proposed. A good survey may be found in [2, 6]. The implementation of these strategies can be broadly categorized by their methods of dealing with the causality. Conservative synchronization techniques [2, 3] prevent their occurrence while optimistic methods [5] recover from their occurrence.

Conservative algorithms [2, 4, 6] are known to have low memory requirements. However avoiding or detecting and breaking deadlocks can reduce greatly the performance of these algorithms, given the large number of cycle in circuits. On the other hand, although optimistic algorithms are deadlock free, experience with Time Warp over the past several years has revealed two fundamental problems with time warp paradigm – that of memory management resulting from the necessity to save state and of instable behavior [1].

Nevertheless, asynchronous parallel simulation techniques are very attractive for logic simulations since they extract a great deal of parallelism, and VLSI systems exhibit a great deal of concurrency which is inherent in their normal operation.

Real progress has been made in the area of parallel logic simulation. The results obtained have shown the feasibility of applying parallel simulation (both conservative and optimistic) techniques to the problem of reducing the run-time of VLSI circuit simulation. A Number of factors have been shown to affect the performance of parallel logic simulation [1, 3]: Circuit characteristics that deter-

mine the inherent parallelism in the design; the timing granularity of the underlying logic simulator; the target architecture, the partitioning methods used to decompose the circuit into multiple sub-systems, and to balance the workload for each partition; and perhaps most importantly, the overhead of the time synchronization protocol used to coordinate the simulation across multiple processors.

With these difficulties in mind, and the ever increasing use of logic simulation and simulators for designing large and complex systems, VLSI systems have brought several challenges to the PDES¹ community. The challenges require not only extension and advances in current parallel logic simulation methodologies, but also the discovery of innovative approaches and techniques to deal with the rapidly expanding expectations of the VLSI system designers.

This special issue is intended to address these challenges and present the state of the art and the current advances in parallel logic simulation. The papers presented in this issue demonstrate clearly the effectiveness of parallel simulation techniques on the performance of logic simulation.

While the first two papers investigate the use of conservative synchronization strategies for logical circuits on shared-memory machines, the third and fourth papers investigate the performance of time warp. The last paper proposes a hybrid approach.

The first paper, “Conservative Circuit Simulation on Shared Memory Multiprocessors” by **Jorg Keler *et al.***, presents several new serialization free-parallel data structures which seem to have large impact on the performance of logic simulation. The efficiency of these data structures is based upon the use of parallel prefix operations. The authors consider first the PTHOR algorithm for simulation of logical circuits, which uses a conservative approach, then they show how attainable speedup can be increased by several changes in the data structures, including the memory management. The paper also examines

---

¹ Parallel Discrete Event Simulation.

another challenging problem – the partitioning problem of the simulated circuit among the parallel processors. Several dynamic partitioning algorithms are described and compared with each others.

The paper “Dynamic Cancellation: Selecting Time Warp Cancellation Strategies at Run-time” by **Philip Wilsey *et al.***, explores the use of optimistic parallel simulation for logic simulation. Previous studies have shown that the performance of time warp simulators can be affected by the cancellation strategy used to send anti-messages. The same studies have also indicated that neither the aggressive cancellation nor the lazy approach is superior to the other in all cases. Furthermore, no strategy exists to make *a priori* determination of the more favorable cancellation scheme. Philip Wilsey *et al.*, investigate the use of simulation time decision procedures to select cancellation strategies, *i.e.*, within a single parallel simulation both schemes may be employed by distinct logical processes (LP) and even across the simulation life time of an LP. Their results indicate that dynamic cancellation always outperforms the best static strategy.

As in many other parallel simulations, a good partitioning algorithm is the key to achieve good performance in parallel logic simulation, especially since the event granularity is relatively small compared to other types of simulations. The paper “Concurrency Preserving Partitioning (CPP) for Parallel Logic Simulation”, by **Hong, Kim and Jack Jean** presents an efficient, linear-time partitioning algorithm for parallel logic simulation, based on a linear ordering of vertices in a directed graph. The performance results of the algorithm presented in this paper show a good compromise with a high degree of concurrency, a balanced work-load and a resonable amount of interprocessor communication. The authors show that the optimistic parallel simulation algorithm with the CPP algorithm enables powerful parallel gate-level circuit simulators.

The paper, “Design and Implementation of Dynamic Load Balancing Algorithms for Roll-

back reduction in Optimistic PDES” by, **Sarkar Falguni and Sajal K. Das**, propose two algorithms for dynamic load balancing which reduce the number of rollbacks in an optimistic PDES systems. The first algorithm is based on load transfer mechanism between logical processes; while the second algorithm is based on the principal of evolutionary strategy, migrates logical processes between several pairs of physical processes. The performance results of the proposed schemes show a significant reduction of the number of rollbacks.

The paper, “Scalable Clustered Time Warp and Logic Simulation”, by **Herve Avril and Carl Tropper** introduces a new hybrid algorithm which is referred to as CTW. CTW has its roots in the problem of distributed logic simulation, and can be implemented on general purpose distributed memory architectures. It is a hybrid scheme in that it makes use of a Time Warp scheme between clusters of logical processes, and a sequential algorithm within each cluster. In their implementation, each cluster corresponds to a large number of gates in a circuit (such as functional logic elements). The proposed algorithm takes advantages of the fact that LPs within a cluster share the same address space, thereby making their synchronization and scheduling straightforward. Resting upon this platform, the authors propose a family of three checkpointing algorithms, each of which occupies a different point in the spectrum of possible trade-offs between memory usage and execution time. CTW seems to be a viable approach to logic simulation, and presents a promising method in the area of parallel logic simulation.

Many people have contributed to the creation of this special issue. Thanks are due to **Prof. G. Zobrist**, Editor-in-Chief, for his support, his encouragement and patience. Thanks are in order to **Prof. Mani K. Chandy** for his encouragement, and his motivation to have a special issue showing current advances in parallel simulation with real applications. All papers that have been submitted were reviewed by three and sometimes by four

reviewers. Two to three rounds of reviews have been done to ensure of the quality of the papers. Special thanks to all referees for their excellent contributions. Their hard work, their comments and suggestions really helped to improve the quality of the papers and this issue.

References

- Bailey, J. V., Briner, R. D. and Chamberlain (1994). "Parallel-Logic Simulation of VLSI Systems", *ACM Computing Surveys*, **26**(3), 255–295.
- Boukerche, A. and Tropper, C. (1995). "Parallel Simulation on the Hypercube Multiprocessor", *Distributed Computing*, Spring Verglas.
- Briner Jr. (1991). "Fast Parallel Simulation of Digital Systems", *Proc. of the 5th Workshop on parallel and Distributed Simulation*, pp. 71–77.
- Chandy, K. M. and Misra, J. (1979). "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs", *IEEE Trans. on Soft. Eng.*, **SE-5**, 440–452.
- Jefferson, D. R. (1985). "Virtual Time", *ACM Trans. Prog. Lang. Syst.*, **77**(3), 405–425.
- Misra, J. (1986). "Distributed Discrete-event Simulation", *ACM Computing Surveys*, **18**(1), 39–65.

Guest Editor's Biography

Azzedine Boukerche is an Assistant Professor of Computer Science at the University of North Texas. Prior to this, he was working as Senior Scientist at the Simulation Sciences Division,

Metron Corporation located in San Diego. He was employed as a Faculty Lecturer at the School of Computer Science (McGill Univ.) from 1993 to 1995. He also taught at Polytechnic of Montreal during the Academic year 1993–1994. He spent the 1991–1992 Academic year at the California Institute of Technology where he contributed to a project centered about the specification and verification of the software used to control interplanetary spacecraft operated by JPL/NASA Laboratory.

His current research interests include parallel simulation, wireless networks and mobile computing, distributed computing, distributed interactive simulation, and VLSI designs. Dr. Boukerche has published several research papers in these areas. He was the recipient of the best research paper award at the PADS'97. He was the Program Chair of the First and the Second International Workshop on Distributed Interactive Simulation and Real Time Applications (DIS–RT), and the General Chair of the principle symposium in Modeling Analysis, and simulation of Computer and Telecommunication Systems, MASCOTS, in 1998. He has been a member of the Program Committee of Parallel Simulation Conference (PADS) and MASCOTS. Dr. Boukerche is also a member of IEEE and ACM.