

Cluster-based Parallel 3-D Monte Carlo Device Simulation

ASIM KEPKEP, UMBERTO RAVAIOLI* and BRIAN WINSTEAD

*Beckman Institute and Dept. of Electrical and Computer Engineering, University of Illinois
at Urbana-Champaign, Urbana, IL 61801, USA*

The recent improvements in the performance of commodity computer have created very favorable conditions for building high performance parallel machines from computer clusters. These are very attractive for 3-D device simulation, necessary to model properly carrier-carrier interaction and granular doping effects in deeply scaled silicon devices. We have developed a parallel 3-D Monte Carlo simulation environment customized for clusters using the Message Passing Library (MPI). The code has been tested on the supercluster of NCSA at the University of Illinois. We present here test results for an n-i-n diode structure, along with an analysis of performance for two different domain decomposition schemes.

Keywords: Monte Carlo; Parallel computation; Device simulation; Semiconductors

1. INTRODUCTION

In recent years there has been a continuous increase in the power of consumer computer products with a very large reduction in cost. This trend has eliminated most of the difference in performance between specially designed high performance processors and commodity microprocessors. Also, over the same period networking technology has matured, making a robust and high-performance networks available to end users without having to design special purpose hardware. As a result, clusters of computers using commodity hardware has become a cost-effective

way to build very high-performance parallel computers. It has been demonstrated that a computer built this way can reach performance levels of custom built supercomputers [1]. The other issue that helped promoting cluster-based supercomputers is the relatively recent standardization of communication libraries and their application programming interfaces with high level languages such as C/C++ and Fortran. Leading among these standards, MPI has become virtually available on all platforms, allowing one to port same code to other platforms with minimal changes [2]. The pressure to integrate more functionality on a single chip and the demand

* Address for correspondence: 3255 Beckman Institute, University of Illinois at Urbana-Champaign, 405 N. Mathews Avenue, Urbana, IL 61801, USA. Tel.: +1 217 244 5765, Fax: +1 217 244 4333, e-mail: ravaioli@uiuc.edu

for faster devices has caused device dimensions to shrink continuously. Therefore, device technology has progressed to the point where 3-D geometrical effects, discrete dopant ion distributions, and carrier-carrier interaction must be dealt with. Also, there is an interest in exploring 3-D geometries, where the functionality of the device can be enhanced by the availability of an additional degree of freedom [3, 4]. Although in many cases conventional simulation approaches (drift-diffusion or hydrodynamic method) are still useful, Monte Carlo transport simulation is becoming computationally more attractive, since the increase in dimensionality does not change the nature of the approach.

This paper describes the development of a Monte Carlo transport simulation environment for semiconductor devices, targeting implementation on cluster-based supercomputers. The code has been developed with a “message passing” model with relatively high communication costs.

2. DESIGN CONSIDERATIONS

Flexibility for future development has been the main consideration for the design of the code structure since a range of investigations, device structures, and physical effects need to be supported. As a consequence, any restrictive or *ad hoc* decisions may easily create difficulties in the applicability of the code to other situations. At the heart of the simulation environment is a complete 3-D Monte Carlo transport simulator code for silicon, which has evolved from a earlier 2-D full band Si-MOSFET simulator [5]. The 3-D code has been structured to handle rectangular brick-like domains. The inter-node boundaries are chosen to be planes, and while not sacrificing generality this choice helps reducing the computational overhead associated with detecting boundary crossing for the simulated particles. The transport simulator is essentially a meshless design, and it does not depend on a specific choice of force calculation scheme.

There are several opportunities for parallelism in the Monte Carlo scheme. Since the dynamics of particles in distant regions are relatively independent of each other, it is possible to decompose the geometry and assign parts of the device to individual process (or “nodes”). Each of these sub-domains can be treated independently from a computational point of view, with periodic recalculations of the forces, which provide the physical and the computational coupling between different regions. Another inter-domain interaction occurs when a particle crosses a boundary. As a result, one would like to find a partitioning of the device that minimizes inter-domain boundary crossings, to minimize not only computation but also the communication costs associated with node to node particle data transfers. A second source of parallelism is the possibility to spread the ensemble of the particles within a certain domain over a number of processors. This mode of parallelism is not being utilized in this particular implementations since one can split a single domain into multiple domains to spread the computational overhead. However, this source of parallelism can be advantageous, for instance in symmetric multi-processor systems (SMP), better suited for a shared memory programming paradigm, where parallelism across the particles can be exploited in a much more favorable manner than with a true message passing paradigm. As a result a mixed mode parallel code, with node internal parallelism across the ensemble utilizing shared memory, along with message passing among nodes might be an optimal solution. A third source of parallelism lies in the fact that electrons and hole can be treated independently during a time-step and there is a two-fold parallelism available if one uses a bipolar mode simulation.

The bandwidth and latency for the Myrinet network used in the cluster employed for this study are good enough to handle random message exchanges relatively easily. However, in order not to hinder the runtime performance on clusters without such a high performance network, keeping communication patterns simple and structured has

been an important design consideration. Therefore, the data structures containing the external device boundaries have been replicated on all nodes. While costing more memory, this allows one to completely evaluate the dynamics of a particle over a given time-step, locally on the node the particle was located at, in the beginning of the time-step. At the end of each time-step a set of messages are exchanged, passing messages containing the new state variables for particles crossing decomposition boundaries.

3. IMPLEMENTATION

The Monte Carlo simulator has been implemented using Message Passing Interface (MPI) as the underlying parallel communication library. MPI is an industry standard communications library for parallel programming. There are several implementations available, some of which are system specific since they are optimized to yield maximum performance on a given architecture. The advantage of using MPI is that while functionality of the

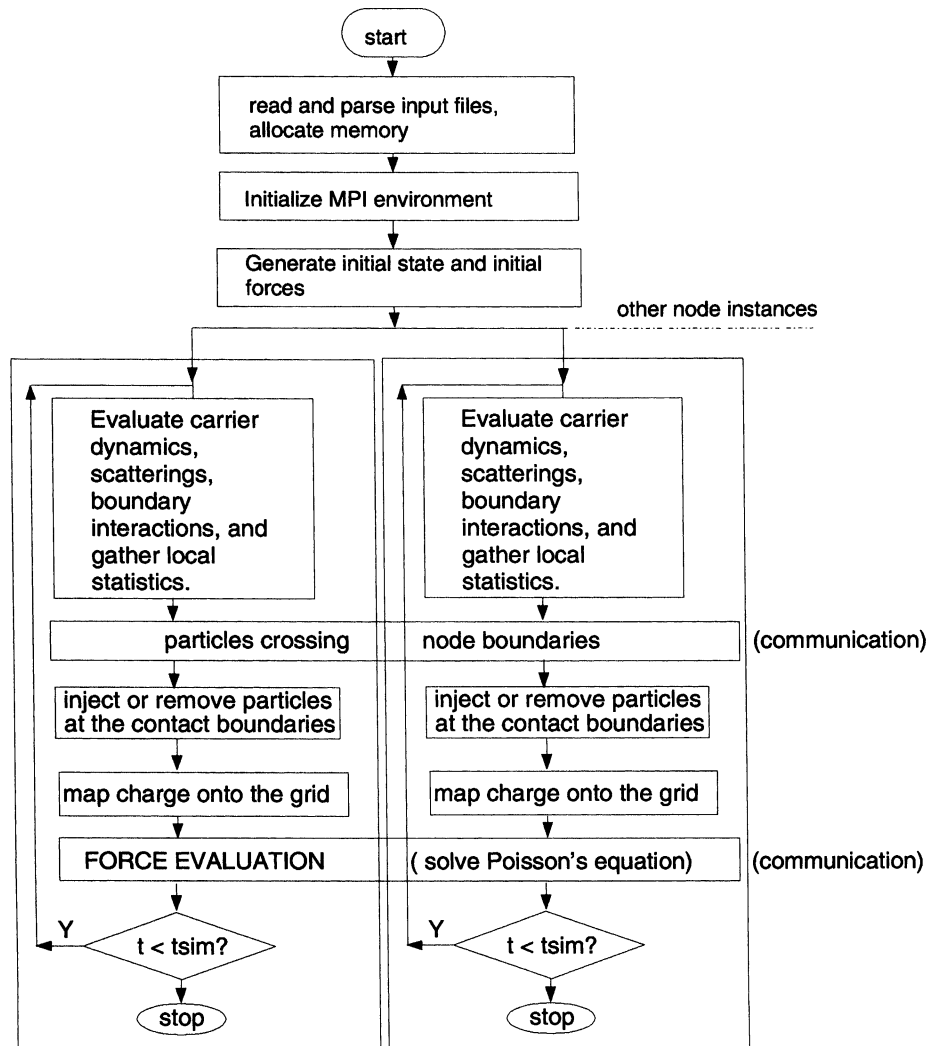


FIGURE 1 Flow chart for the parallel Monte Carlo simulator.

code is always retained, although the performance may not be optimal, when code is re-targeted for a new platform. Also, it is easier to deal with heterogeneous clusters, allowing computers with different architectures to be used, so that portions of the code where processor speed is indeed the bottleneck can be run on a higher performance computer allowing a better overall performance.

The schematic flow-chart of the program is shown in Figure 1. After initialization of the individual threads of execution, each branch reads in the input files, band structure and post scattering data tables followed by initialization of the MPI environment. Each task applies to a sub-domain with rectangular symmetry, defined by user. This is followed by state initialization for the particles, for instance a charge neutral equilibrium or the result of a previous simulation. In order to ensure independence of random number streams used by different tasks, the random number generator associated with each task is seeded with a seed offset from a global seed according to the task number. This saves some CPU time over a “leap-frog random generator” which would increase the CPU time spent in random number generator linearly with the number of threads. To check the randomness, tests with up to 64 random streams were generated, and the resulting streams were analyzed, showing sufficient independence.

After initial validation of the approach on a small cluster of DEC Alpha computers, the code has then been re-targeted for the NT supercluster at the National Center for Supercomputing Applications (NCSA). This cluster-based multi-computer is built from 64 Hewlett-Packard Kayak XU dual Pentium II processor computers. The high performance Myrinet network allows users to access one directional bandwidths of about 1.2 Gb/s with small message latency of about 20 μ s. A special implementation of MPI based on HPVM, specifically designed for Myrinet has been used on this platform. MPI-FM is a complete implementation of MPI 1.0 standard for development of FastMessages interface based on public domain MPICH code base.

The core node code is written in FORTRAN 77 with higher level layers written in ANSI C to enable dynamic memory management, as well as easy access to system management functions. This strategy allows one to achieve good performance on a variety of platforms, without resorting to intensive platform specific optimizations.

4. RESULTS AND CONCLUSIONS

An *n-i-n* structure has been simulated to test the 3-D Monte Carlo parallel implementation. The structure consists of a semiconductor slab, 0.5 μ m long with cross-section $0.2 \times 0.2 \mu$ m. The contacts located on both ends of the device have *n*-doping of 10^{18} cm^{-3} and are 0.1 μ m long. The mid-section of the device is undoped. The biased in this example is 0.25 V. Simulations have been conducted for two different decomposition schemes to see the impact of the time spent during message exchanges. In the first case, the device is decomposed into slabs, which are parallel to the longitudinal axis of the structure (Fig. 2). This configuration minimizes the message exchanges, since the particle velocity is favorably aligned. This

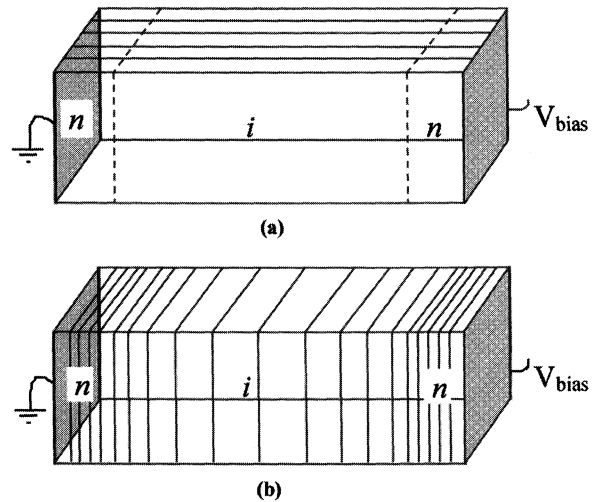


FIGURE 2 Simulated structure with: (a) sub-domains oriented along the longitudinal axis of the *n-i-n* structure and (b) with sub-domains oriented in the perpendicular direction.

also results in a well-balanced load among the sub-domains. In the second case, the device has been partitioned into sub domains with boundaries perpendicular to the longitudinal axis of the original structure. In order to keep load balance, an attempt is made to keep the number of particles in a given node approximately the same made. An ensemble of 40,000 particles was simulated for each case. In Figure 3, we show the speed-up obtained for the two test cases. The performance for the first partitioning scheme (small boundary crossing frequency) shows an almost linear behavior up to four processors and then it saturates. Of course, the saturation point will correspond to a much larger number of processors, as the problem size is scaled up to include many more particles and grid size. Note that here we are focusing on the parallelism of the transport procedure, and Poisson equation is solved in scalar form on a single processor. The related cost is only about 20% and it accounts for the slight deviation from the linear speed-up curve. In future large-scale simulations of complete MOSFET structures, a parallel version of the Poisson solver will be used, since the related computational cost will increase considerably with respect to the transport part of the code. This is a very important issue when charge-charge interaction and granular doping

effects are to be explored. The second decomposition scheme shows a marked degraded performance when more than two processors are used. This is just a consequence of the large number of boundary crossings along the conduction path, resulting in an enormous communication load. In this case, one should also expect a general improvement if the problem is scaled up, but, as the number of sub-domains increases with the number of processors, the performance should again start to degrade past a certain limit.

The partitioning scheme for the geometry of the structure is probably the most important factor in achieving good performance. Physical intuition and a good knowledge of the transport condition can help considerably in the design of a successful scheme. Our conclusion is that the domain decomposition should have the following properties: (i) volume to surface ratio of every sub-domain with sizable population should be maximized; (ii) the topology of the sub-domain should minimize boundary crossings in the direction of the current flow. For complex structures with asymmetries in the current flow, adaptive domain decomposition schemes would be necessary to achieve high performance, since the current flow patterns may not be easy to guess for the purpose of load balance.

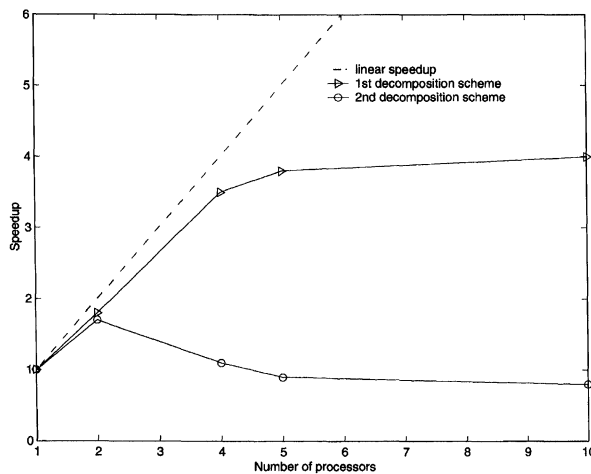


FIGURE 3 Speed-up as a function of number of nodes for the two partitioning schemes.

Acknowledgements

This work was partially supported by the Semiconductor Research Corporation contract 99-NJ-726, by the NSF Distributed Center for Advanced Electronics Simulation, grant ECS 98-02730 and by NCSA at the University of Illinois.

References

- [1] Chien, A. *et al.* (1999). "Design and Evaluation of an HPVM-Based Windows NT Supercomputer", *Int. J. High Performance Computing Applications*, 13(3), 201–219.
- [2] William Gropp, Ewing Lusk and Anthony Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface* (MIT Press, 1994).

- [3] Lorenzini, M. *et al.* (1999). Three dimensional Modeling of the Erasing Operation in Submicron Flash-EEPROM Memory Cell, *IEEE Trans. on Electron Dev.*, **46**(5), 975–983.
- [4] Tanaka, J. (1996). “Simulation of a High Performance MOSFET with a Quantum Wire Structure Incorporating a Periodically Bent Si-SiO₂ Interface”, *IEEE Trans. on Electron Dev.*, **43**(12), 2185–2198.
- [5] Duncan, A., Ravaoli, U. and Jakumeit, J. (1998). “Full-band Monte Carlo Investigation of hot carrier trends in the scaling of Metal-Oxide-Semiconductor Field-Effect Transistors”, *IEEE Trans. on Electron Dev.*, **45**(4), 867–876.

