Atomic Scale Simulator for Silicon Bulk Processing

Kevin R. Lind, Cray Research, Inc., Livermore, CA and *Tomas Diaz de la Rubia,* Lawrence Livermore National Laboratory, Livermore, CA

ABSTRACT: *We describe a project to enhance ion implantation technology by simulating the physical process on the T3D. This involves a suite of molecular dynamics programs plus a process simulator. In particular, we are developing a tight binding molecular dynamics (TBMD) code for the T3D as part of this package. We detail the development issues and plans for the TBMD code, and the impact on the target technology.*

Introduction

This paper reports progress on a project to enhance ion implantation technology by supercomputer simulation on the Cray T3D. This work is being done as part of the High Performance Parallel Processing Project - Industrial Computing Initiative (H4P-ICI) in association with the Cray Parallel Applications Technology Program (PATP). H4P-ICI is a CRADA package within DOE designed to transfer national laboratory computing technology to private industry with the goal of enhancing national competitiveness. This project is developing a suite of programs to advance ion implantation technology from an emerging to a production technology by process simulation, resulting in improvement in productivity and reliability.

What Is Ion Implantation Technology?

Ion implantation is a silicon doping technique for creating very finely structured devices [1]. Ions of the dopant substance are precisely implanted into the appropriate sites on the silicon substrate to produce sub-micron components, such as transistors. The ions produce defects within the silicon which can disrupt the ordered molecular structure of the silicon and enhance the diffusion of the dopant, thereby compromising the reliability and functionality of the device [2]. In short, it is necessary to determine the optimum implantation characteristics for the ions and the resulting damage and diffusion to determine the best ways to make these devices and the device densities and reliabilities which can be attained. This can best be done by computer simulation, which reduces costs by limiting the number of physical trials required for device development, and by allowing a large number of options to be explored, more than could be investigated by experiment alone.

The Simulation Package

The simulation spans a number of physical regimes, and so a number of molecular dynamics applications are required for a full simulation of the ion implantation process. Classical molecular dynamics techniques [3] (in which the ion and substrate are treated using classical interatomic potentials) can be applied to the initial implantation, since the energies and time scales are such that classical interactions dominate. After the initial implantation, mean energies and displacements require that quantum mechanical effects must be taken into account. Traditionally, this has been done using ab initio codes [4], in which all nuclei and electrons are treated in full quantum mechanical generality. This results in a very accurate simulation, but is extremely costly in terms of computer time. On the other hand, the classical codes are very efficient in terms of computer time, but fail to take quantum effects into account. Ideally, we need to combine the efficiency of the classical algorithm and the quantum mechanical considerations of the ab initio techniques in an algorithm which can operate in the intermediate regime, where motions are basically classical but forces have a significant quantum mechanical contribution.

This is the goal of the tight-binding molecular dynamics (TBMD) [5] algorithm, which is used in conjunction with the other two codes as part of this simulation package. The TBMD algorithm parameterizes the electron structure as a set of orbitals, and uses these orbitals to construct the attractive terms in the interaction Hamiltonians. These are combined with the repulsion terms to construct force terms which act on the atoms, which are updated in position as independent particles. While this is more computationally intensive than the classical calcu-

lation, it does take quantum effects into account to a good approximation. On the other hand, while TBMD is less accurate than an ab initio quantum mechanical calculation, it is far less expensive in terms of computational resources, allowing a large number of atoms to be simulated for a sufficiently long time scale.

The results of these simulations are used as parameters in the full process simulation, including the effects of defect kinetics and dopant diffusion, using a Monte Carlo code [6] to select a variety of conditions which may be expected during an actual processing run. An ensemble of these runs is used to generate statistics and characteristics of the simulated devices, from which can be determined optimal manufacturing methodologies and anticipated device characteristics and reliabilities.

In summary, the classical MD code is used to simulate the initial implantation of the ion and the initial energies and displacements in the resulting defects. The ab initio code is used to generate a parameterization of the atomic orbitals in the dopant and substrate atoms, which is input to the TBMD code. The TBMD code then updates the substrate defect structure and dopant migration to generate parameters to feed into the full device simulation routine, which uses a Monte Carlo code to select an ensemble of events that give a reasonable model of the entire process. The final result of the simulation is a characterization of the device itself which can be used to guide future simulations, with the goal of maximizing yield and reliability, while minimizing production cost. These results are compared to experiment to confirm the results of the simulation. In this way, the industrial process is refined and brought to production status at a fraction of the time and cost of traditional, non-computational development projects.

In this paper, the classical code will be discussed briefly, and the TBMD code will be discussed extensively. The ab initio code is under the purview of another CRADA within this initiative, and will not be discussed.

The Classical MD Code

The classical molecular dynamics code provides the source term for the diffusion. This code has proven effective at describing the point defect production mechanisms as a function of ion mass and energy [7], i.e., it provides an accurate model of the damage to the lattice structure of the substrate and the initial position of the implanted ion for various ions implanted at various velocities. It also provides valuable information on the stability of a single damage region under thermal annealing conditions, i.e., how quickly and widely the initial damage recovers at the temperatures required for the implantation procedure. The outputs from this stage are: 1) the displacements of lattice atoms produced by implanted boron and arsenic ions at energies relevant for .25 CMOS fabrication; 2) information on the validity of models for extended defect formation (long-range effects on substrate structure); and 3) critical parameters of the amorphization of silicon at high dose, i.e., the breakdown of the substrate when heavily bombarded by implantation ions. This code currently gives linear performance scaling of about 1.5 GFlops for 128 processors on the Cray T3D, and even in the nonlinear regime still provides almost 2 GFlops for 256 processors.

The TBMD Code

Purpose and Description

The TBMD code simulates the subsequent energetics and evolution of large defect and dopant clusters, as well as refining the modeling of the mass transport during high temperature annealing, i.e., looking at the response of the substrate and ion to the initial implantation as the defects spread over a wide region. The initial implementation of this algorithm used an O(N^3) solver for the eigenvalues and eigenvectors; an efficient O(N) approximate solver has also been implemented on both the T3D and the Power Challenge, but is currently limited in accuracy. The current O(N^3) code has confirmed existing TBMD calculations of silicon vacancy and interstitial energies, and is being used to carry out additional calculations and to develop new parameterizations for dopant diffusion in silicon. The O(N) code has proven useful in calculating the recrystallization of amorphous silicon at annealing temperatures. Development work on the O(N) algorithm continues as a high priority; the computing requirements of the O(N) and O(N^3) algorithms cross over at about 90 silicon atoms, so the O(N) algorithm promises greatly improved size and time simulation capabilities once the stability and accuracy of the solver has been improved.

Parallelization and Optimization

The parallelization and optimization of the TBMD O(N) code has provided some interesting challenges, which are the current emphasis in the Cray Research contribution to this project. The work sharing potential for this algorithm is very good; since this is a particle-based scheme, the particles can be divided up between all available processors and speed-up is essentially linear, allowing for longer runs with more processors. However, the current implementation solves data sharing by placing a copy of all data on all processors, which severely limits the size of the problem which can be run to what will fit in the memory of a single processor. This was done because of the intensive communications requirements in constructing and solving for the eigenvectors of the Hamiltonian. While distribution of the large data arrays is the first priority of future development, this must be done without seriously compromising performance; the code must run for both large numbers of atoms and for many timesteps.

The details of the data distribution are set by the communication requirements between particles in the O(N) code. The quantum mechanical information communicated is limited by a cutoff radius, so there is no need for every particle to have information on every single other particle in the simulation. This means that, if particles are assigned to processors based on location, communications can be minimized to some extent even if the data is fully distributed. Furthermore, since nearby particles are likely to need similar information, remote data on a given

particle can be used by multiple particles on the same processor, again reducing communication requirements. The implementation of this level of data parallelism has merely required some data distribution of the larger arrays (indexing data by processor and local index) and implementation of remote communication routines with some loop reordering. It appears useful to allow many of the smaller arrays to be copied rather than distributed, in order to minimize communication costs and programming difficulties.

The electrostatic terms have proven even easier to deal with; while they are global, they are also easily implemented using global sums and limited irregular data access of small arrays. Thus, the non-quantum terms merited little attention, since the volume of remote data is relatively small and can be easily replicated across all processors if desirable.

The real difficulties arise in the solution for the eigenvalues, for this requires the use of, not only neighboring particles, but the neighbors of the neighbors. Also, the loop ordering does not appear to be amenable to the sharing of remote data by multiple particles on-processor, so that multiple remote communication of identical data to the same processor is frequently required. To some extent, this can be overlapped with calculation, but is still anticipated to be the performance bottleneck for this program.

In the implementation itself, the location arrays are first broadcast one at a time to all other processors and used to construct lists of neighbors for all local processors. The broadcast is used to retain the ordering of the lists from the serial version, although it is not clear that this is truly necessary, or that later reordering would not be a more efficient way to deal with the ordering. Point-to-point communication would put less stress on the communication network than multiple broadcasts, and would also allow aggregate location information for each processor to be used to eliminate unnecessary communication stages. Thus, simple improvements should reduce the communication requirements in this stage to reasonable levels.

From the localization information, which is now held locally, the Hamiltonians can be constructed without additional cost for remote communication. The repulsion energy can be added to the quantum mechanical attraction terms using local data and copied arrays, retaining the local flavor of the calculation. Thus, the construction of the Hamiltonian has minimal communication requirements.

The eigenvalue solution is the next, and most computation and communication intensive, stage of the program; hence, most of the optimization effort will center here. The neighbor lists will be used to get data from the neighbors of neighbors, and so neighbors on other processors will have to provide lists of their neighbors, which will then be requested remotely by each particle. A great deal of duplicate remote communication takes place here, and it is not immediately clear how to solve this problem. Some form of data reuse will need to be worked out, which may require extensive reordering and even the introduction of large intermediate arrays. Even if the latter should become necessary, the data requirements would remain far less than those for the non-distributed data algorithm, and the reduction in communication costs is critical to meet the anticipated simulation needs for the project as a whole.

Once the eigenvector solutions are worked out, the remainder of the calculations involve the use of local data to construct forces, with some global sums of small amounts of data. The remote communication requirements in this final stage are minimal, and no optimization effort should prove necessary at that point.

Remote Communication Model

This code as a whole requires a moderate to large amount of remote communication, much of which is irregular, and so the choice of method is important. The use of the CRAFT model was eliminated by the power-of-two requirements (which would have wasted a great deal of memory for efficient communication organization), the lack of access to cache, and the difficulty in truly optimizing communication. PVM/MPI were eliminated because low latency was more important than portability, and also, the buffer space requirements would have made significant inroads into available memory. The shared memory (shmem) model provides the best combination of low latency, efficient use of memory, access to cache, and remote communication optimization, and is used despite the disadvantages of programming difficulty and non-portability.

Summary

This project will greatly enhance ion implantation technology by providing simulated device manufacturing information that will help improve device capability and reliability at reduced cost and time. In particular, TBMD is a critical component of the simulation package, providing accurate information on intermediate energy and time scale energetics and kinetics. It currently provides good work scaling but poor data scaling, enabling simulations to be run for many time steps, but only on a small number of particles. Current efforts are directed at distributing the data and optimizing communication so that large simulations can be run for many timesteps, which will allow full-scale simulations to be run with reasonable computing resources. This will complete the full capability of the entire simulation process, and will allow the device development team to take full advantage of computer simulation to advance ion implantation technology into a real production phase.

References

[1] *The National Technology Roadmap for Semiconductors*, Semiconductor Industry Association (SIA) (1994)

[2] Poate, J. S., *Ion Implantation and Beam Processing*, Orlando, Academic Press (1984)

[3] Allen, M. P. and Tildesley, D. J., *Computer Simulation of Liquids*, Oxford Scientific Publications (1986)

[4] Car, R. and Parinello, M., Phys. Rev. Lett. **55**, 2471 (1985)

[5] Slater, J. C. and Koster, G. F., Phys. Rev. **94**, 1498 (1954)

[6] Jaraiz, M., Gilmer, G. H., and Diaz de la Rubia, T., Phys. Rev. Lett. **68**, 409 (1996)

[7] Caturla, M.-J., Diaz de la Rubia, T., and Gilmer, G. H., Phys. Rev. B (submitted)