The Parallel Simulation Model for Thin Film Deposition Using the DSMC Method

Anan Banharnsakun[1,*], Rardchawadee Silapunt[2], Tiranee Achalakul[1]

[1]*Department of Computer Engineering*

[2]*Department of Electronic and Telecommunication Engineering*
King Mongkut's University of Technology Thonburi,
Bangkok 10140, Thailand

[*]*Corresponding author: smilecolon@yahoo.com*

Abstract

Nowadays, the thin film deposition process is essential in the semiconductor fabrication process and used extensively in many applications. Progress in each of these areas depends upon the ability to selectively and controllably deposit thin films. This paper proposes a design framework and implementation of the simulation of thin film deposition processes. This presented work focuses on the deposition processes in two-dimensional geometries and the simulation is considered in the context of neutral flow Direct Simulation Monte Carlo (DSMC) simulations for semiconductor fabrication. Parallel computing is employed to improve the performance optimization of the system. This paper describes the simulation method and algorithms that are used to implement the simulation on a parallel system and presents the results of thin film deposition simulation.

KEY WORDS: Thin film deposition, parallel simulation model, Direct Simulation Monte Carlo (DSMC)

Introduction

The thin film technology is currently used extensively in many applications including microelectronics, optics, magnetic, corrosion resistant coatings, and micro-mechanics. The technology also plays an important role in semiconductor fabrication industries. The goal of the deposition process is to develop an ability to selectively and controllably deposit thin films on silicon wafers. This paper presents the design and the development of a parallel computational tool for deposition simulation. The tool allows the scientists to optimize the deposition process as well as to predict the optimal sticking coefficients that are used to control the uniformity and surface growth, for arbitrary input conditions. In order to study the process, several inputs such as, the type of gas, reactor size, and substrate size, can be parameterized.

Our work focuses on the deposition processes in two-dimensional geometries. The simulation tool is developed based on the neutral flow Direct Simulation Monte Carlo (DSMC). As sited in previous literatures, DSMC has a high computational cost when applies to a large substrate size. We thus, employ the parallel and distributed programming technique in order to keep the computational cost within an acceptable limit. The tool is designed to be executed in a grid environment where a high computational power is readily available. Our parallel implementation utilizes the message passing technique as a communication paradigm. The problem is partitioned using the domain decomposition method. The simulation domain is divided into several cell-grids and each grid is assigned to a separate processor. Particles transport and collision are computed independently in each processor. The inter-processor communications only occur when particles move out of the cell-grid bound.

When the steady state is reached, we collect the output data from the simulation, i.e., density, velocity and temperature of gas molecules. These values are then used to calculate the sticking coefficient based on the Langmuir model. The model describes ideal chemical mechanical absorption and is used to study the surface growth. Our simulation produces the result in the previously published theoretical ranges. Finally, the parallel performance and scalability are observed. From our

experiment, the average processing time decreased as more computing nodes are added to the computation. We can thus conclude that our simulation tool is fast, efficient, and accurate.

Related Research

The DSMC method has been first developed by Bird (1976). In recent years, there have been several review articles about the DSMC method. In this section, we discuss the related works in particle simulation based on DSMC and the parallel computing techniques that were previously used to optimize the performance. The research by Robinson and Harvey (1995) focused on the development of an effective parallel implementation of the DSMC method. The designed algorithm emphasized rarefied flows problem solving and addressed the effectiveness of the domain decomposition technique for DSMC method. The results concluded that the DSMC is perfectly suited to the data domain decomposition because the algorithm allowed each sub-domain to operate independently from each other. To enhance the DSMC method for use with a very large system, Dietrich and Boyd (1996) proposed the system called, ‘‘MONACO’’. MONACO targeted to build a high performance tool using object-oriented design. DSMC was modified and the implementation was rather complicated. DDAC software that was used to track molecules proposed by LeBeau (1999) provided an efficient implementation for the unstructured triangular grid and thus supported the irregular surface geometry. Although flexible, the software has high computational cost. In our work, we aim to provide a simple and fast, DSMC-based simulation tool that can be adapted to a series of simple applications. The experimental domain was thus assumed to be a uniform rectangular grid, which was often the case for many molecular dynamics simulation. After extensive reviews of the previous works, we have decided to adapt the concept presented by Robinson and Harvey (1995) for the thin film deposition process. The physical domain decomposition technique was utilized. Molecules tracking of rarefied flows were modeled. Our goal is to provide a parallel simulation tool that can run on an inexpensive cluster of PCs. The details of our parallel simulation algorithm and the experimental results are presented in the following sections.

Materials and methods

In this section, we describe a design framework and a sequential implementation of the thin film deposition algorithm based on the DSMC method (Bird, 1976; Bird, 1994). The proposed algorithm aims at simulating particles transport and collision as well as calculating the sticking coefficient from the simulation data based on the Langmuir model (Kolasinski, 2002; Masel, 1996; McCash, 2001). In the last, we present the parallel computing approach that is used to improve the performance optimization of the system.

Design framework

Our basic design framework of the thin film deposition system involves the particles simulation and the calculation of the sticking coefficient. The particle simulation was built based on the DSMC method. DSMC decoupled the continuous process of particles movement into two consecutive phases: transport and collision at each time step Δt. The Langmuir model was then applied to find the sticking coefficient based on the output obtained from DSMC. The design framework is divided into 8 major phases: system initial phase, particles injection phase, particles movement phase, particles sorting and indexing phase, particles collision phase, macroscopic sampling phase, average output data phase and sticking coefficient calculation phase.

System Initial Phase

The simulation domain model that illustrated in Figure 1 is mapped from physical domain space and divided into several cells which cell size should be smaller than the mean free path (λ) and is given by;

$$\lambda = \frac{1}{\sqrt{2}\pi\sigma^2 n} \qquad [1]$$

where n = the number density of any gas.

σ= collision cross section.

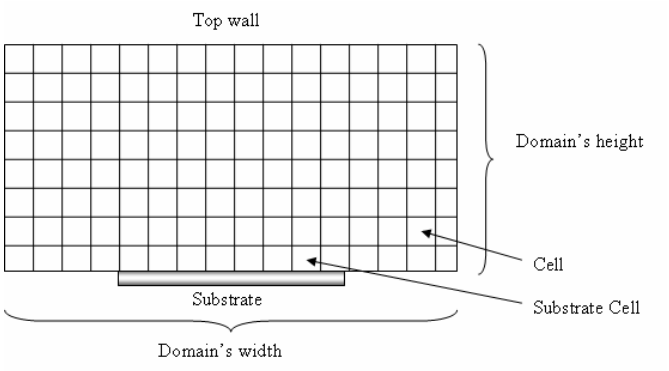

Figure 1: The simulation domain model in 2 dimensional.

The cells that are adjacent to the substrate are called a "substrate cell". The simulation input consists of a domain simulation's width and height, a substrate size, a number of particles released at the top wall, a number of time steps, a sampling time step, characteristics of simulated gas, and flow state conditions. These inputs can be parameterized in a simulation configuration file.

Particles injection phase

The particles are injected at the top wall (inflow boundary). An initial velocity of each particle is assigned following the Maxwell-Boltzmann distribution (Garcia, 2000) that can be written as

$$p(c) = 4\pi\left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} c^2 e^{\frac{-mc^2}{2kT}} \qquad [2]$$

where

k = Boltzmann constant (1.3806 x 10^{-23} J/K).

T = gas temperature (K).

m = molecular mass (kg)

Particles movement phase

In this phase, particles movement may or may not collide with a wall. A particle that flows downward without interactions with any boundary, is simulated using the following equation;

$$R_j = R_i + c_{rms}\,\Delta t \qquad [3]$$

where

R_i = an initial coordinate of a particle before moving.

R_j = a new coordinate of a particle after moving.

c_{rms} = a root mean square particle speed.

Δt = a time step.

Figure 2 shows how to simulate a particle in a case where boundary interaction is present. Each boundary (top, left, and right wall) is called, a Specular surface. When a particle strikes a Specular

surface, the time interval from an initial location, R_i, to the point of impact, R_w, can be determined by tracing a straight-line trajectory. The time interval can be calculated as shown in equation 4.

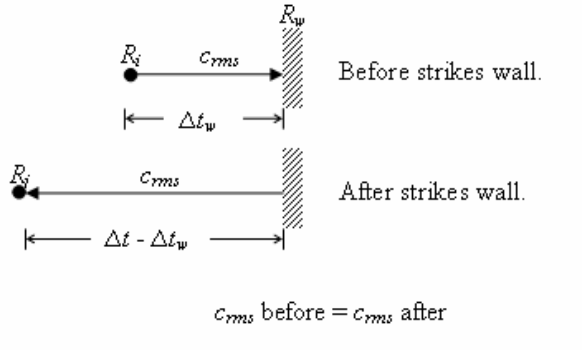

Figure 2: Particle interactions with specular boundary

$$\Delta t_w = (R_w - R_i)/c_{rms} \quad [4]$$

After a particle strikes a surface, it bounces off in a reversed direction with the same velocity it travels before the collision. The bouncing time interval is Δt - Δtw. The new position of a particle can be calculated using equation 5.

$$R_j = R_w + c_{rms}(\Delta t - \Delta t_w) \quad [5]$$

where

R_w = a point of impact.
Δt_w = the time of flight from the particle's initial position to the point of impact.

Particles that hit the outflow surface (bottom of a reactor) are simply removed from the simulation. These particles are re-initiated with a new coordinate at the top wall with a newly defined velocity following equation 2. Also, a cell grid block that each particle falls within is identified throughout the simulation using particles coordinates (x, y).

Particles sorting and indexing phase

This phase is the phase that prepares the sorted lists that used to select random particles from cell for calculates particle collision in next phase. This phase contains 3 steps to process the sorting and indexing.

1. Count the number of particles in each cell.
2. Build the indexed list by calculating a cumulative sum of the number of particles in each cell.
3. Build cross-reference list.

Particles collision phase

In this phase, we use the particle sorted and indexed array that obtain from previous phase to evaluate particle collision. Statistical techniques are used to determine the correct number of collisions in time proportional to the number of particles in the cell. The acceptance-rejection scheme (Garcia, 2000) is used to select collision pairs.

Macroscopic sampling phase

When the sampling time is reached, each substrate cell is evaluated. First, we count the number of particles in each cell and only the substrate cells that contain at least 20 particles are sampled. Macroscopic conditions are then collected from each substrate cells. The collected values include density, velocity in x- and y-coordinate, and root mean square speed. These values will be used to calculate the average data in the next phase.

Results averaging phase

After all sampled macroscopic data from each sampling period are collected, average values are calculated for each cell. The average values include density, velocity, temperature, pressure, and incident molecular flux.

Sticking coefficient calculation phase

Finally, in the last phase, we use the outputs from the particle simulation to calculate the coverage and the average of sticking coefficient under various pressures, densities and substrate temperatures based on Langmuir model.

The Parallel System

The particles simulation process is highly compute-intensive and requires a large memory usage. In order to keep the execution cost within an acceptable limit, an efficient parallel algorithm was designed for DSMC. In the algorithm, tasks are divided to a set of sub-tasks and each sub-task is worked on concurrently by different processors. The Manager-Worker model is employed in our parallel implementation. The model consists of one manager and multiple workers. The manager is responsible for initialization, load balancing, decomposing the data distributing the sub-tasks to workers, collecting and displaying the simulation results. The workers are actual computing nodes where the sub-tasks are being worked on. Once finished, the partial results will be returned to the manager. Inter-processor communications occur only between the manager and the workers based on the message-passing interface (MPI). The parallel DSMC algorithm model can be shown in Figure 3.

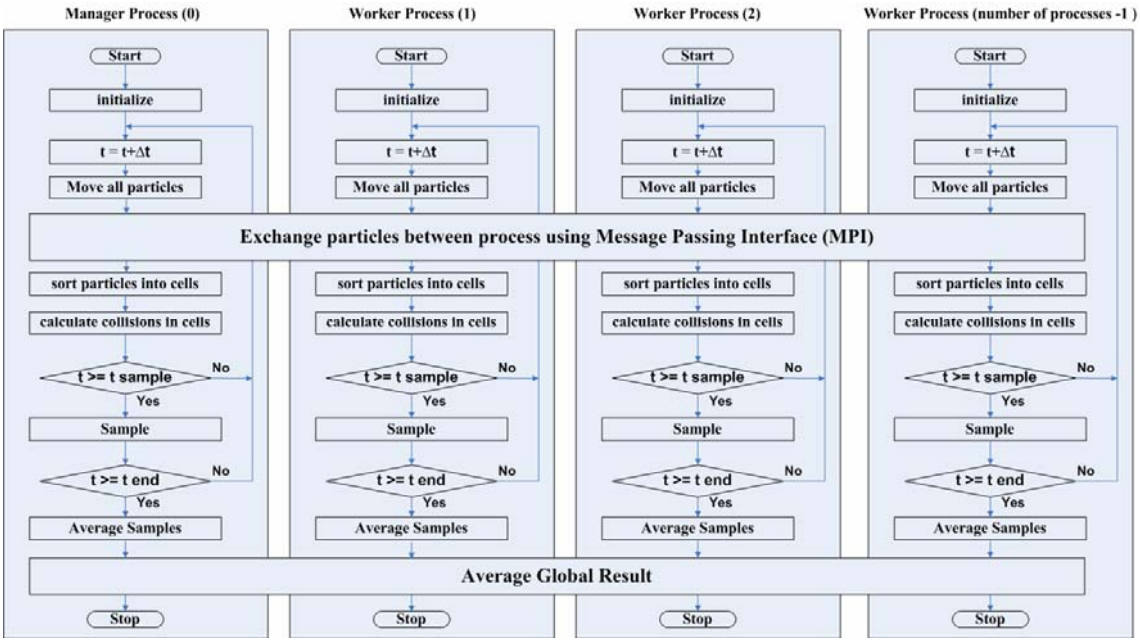

Figure 3: The Parallel DSMC algorithm flowchart

Results

There are several conditions that can effect the surface reaction directly, where the most common ones are density, pressure, and substrate temperature. We have experimented with these conditions and collected data on the accuracy of similarity measurement. Figures 4 and 5 represent our experimental data and results with different conditions. The parallel scalability results are also included in Figure 6.

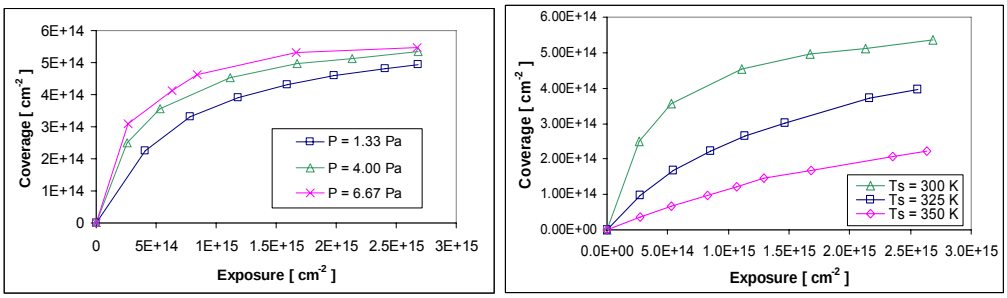

Figure 4: Coverage and exposure under various pressure conditions (left)
Coverage and exposure in various substrate temperatures (right)

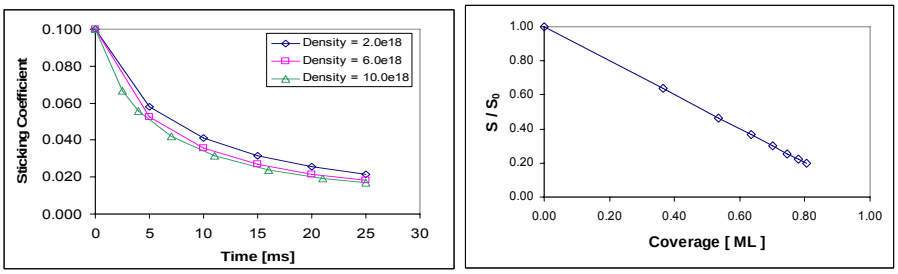

Figure 5: The change of sticking coefficient with CF_2 density (left)
The change in sticking coefficient with coverage (right)

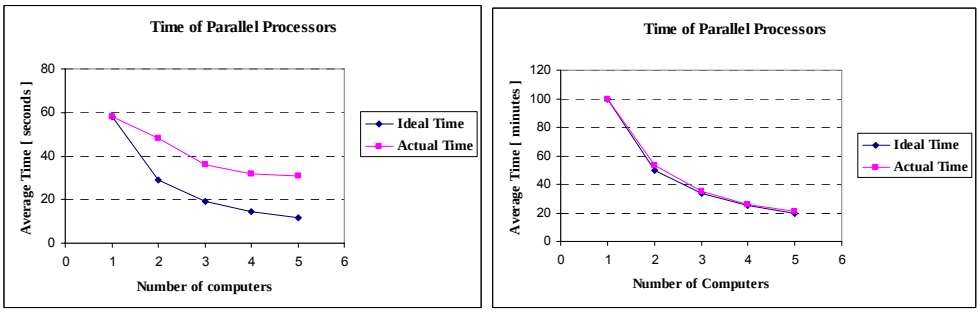

Figure 6: Scalability plot (amount of particles = 10^4 particles) (left)
Scalability plot (amount of particles = 10^6 particles) (right)

In our experiments we tested the system under several initial conditions: the growth of coverage with exposure in various pressure conditions, the change in sticking coefficient with coverage, the effect of substrate temperature on the coverage, and the sticking coefficient in various CF_2 densities. The system was able to accurately produce results which were consistent with the work presented in previous literature in all conditions.

Moreover, in order to keep the execution cost within an acceptable limit, we employed the parallel computing concept. An efficient parallel algorithm was designed for DSMC to run on a cluster of PCs and the implementation was done based on the message-passing interface (MPI). The static load balancing was also applied to eliminate the possibility of system imbalance. We analyzed the parallel scalability and discovered that, as more computers were added to the system, the average processing time decreased almost proportionally. We can conclude that with a large problem size, our parallel algorithm scale almost linearly.

Conclusions

This paper described a design framework and a parallel implementation of our thin film deposition system. The system involves the particles simulation and the calculation of the sticking

coefficient. The particles simulation was built based on the DSMC method, where transport and collision of particles at each time step (Δt) were modeled. The Langmuir model was then applied to find the sticking coefficient in order to monitor the film growth on the substrate's surface. In this work, we developed a parallel simulation tool that could be used to aid engineers in the optimization of the deposition process and the prediction of the surface growth for arbitrary input conditions. With these initial promising results, we believe that some further research will provide engineers with an efficient simulation tool for semiconductor fabrication industries in Thailand.

References

Bird, G.A. 1976. Molecular Gas Dynamics, Clarendon Press, Oxford, 238 pp.

Bird, G.A. 1994. Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Oxford: Clarendon Press, 458 pp.

Dietrich, S., and Boyd, I. 1996. Scalar and parallel optimized implementation of the direct simulation Monte Carlo method. Journal of Computational Physics 126: 328–342.

Garcia, A.L. 2000. Numerical Methods for Physics**,** Prentice Hall, New Jersey, 423 pp.

Kolasinski, K.W. 2002. Surface Science: Foundations of Catalysis and Nanoscience**,** John Wiley, Chichester, 305 pp.

LeBeau, G.J. 1999. A parallel implementation of the direct simulation Monte Carlo method. Computer Methods in applied Mechanics and Engineering 174: 319–337.

Masel, R.I. 1996. Principles of adsorption and reaction on solid surfaces, John Wiley, New York, 804 pp.

McCash, E.M. 2001. Surface chemistry, Oxford University Press, New York, 177 pp.

Robinson, C.D., and Harvey J.K. 1995. The development of an efficient direct simulation Monte Carlo computation scheme for gas flows in a parallel environment. pp. 33-42. Proc. 4th Inter. Parallel Computing Workshop.