論文99-36D-3-7

반도체 공정 시뮬레이션을 위한 초고속 병렬 연산 알고리즘

(Massive Parallel Processing Algorithm for Semiconductor Process Simulation)

李 濟 熙 *, 潘 用 瓚 *, 元 太 映 *

(Jaehee Lee, Yongchan Ban, and Taeyoung Won)

요 약

본 연구에서는 3차원 반도체 공정 시뮬레이션의 효율성과 성능을 향상시키기 위하여, 병렬 유한요소법 수치해석기에 사용이 적합한 디라우니 병렬 메쉬 생성기 및 표면 전진 메쉬 생성기를 개발하였고, 이를 위하여 개선된 성능을 보이는 수정된 하부구조법 병렬 유한요소법 수치해석기를 개발하였다. 또한, 행렬 계산 알고리즘의 병렬화를 확산 및 산화 시뮬레이터에 적용하여, 직렬 계산 시 3시간이 소요되는 확산 시뮬레이션과 비평탄 구조를 지니는 R-LOCOS 등의 연산을 8개의 프로세서를 병렬로 사용하여 15분만에 계산하였다. 과다한 계산 시간을 요하는 몬테카를로 수치해석 방법의 효율성을 높이고자, 병렬 연산 알고리즘을 몬테카를로 연산에 적용하였다. 또한, 스퍼터링 증착장치 시스템의 타겟 입자 분포 특성을 병렬 연산 몬테카를로 방식으로 계산하였다. 3000개의 이온을 주입하였을 경우 단일 프로세서에서 13,000초의 계산시간이 소요되었으나, 30개의 프로세서를 병렬로 사용하였을 때 520초의 시간을 소비하여, 25 이상의 스피드업 특성을 얻었다. 또한, 몬테카를로 계산의 최적화 연구를 통해서 3차원 스퍼터링 증착장치에서 연쇄 충돌 계산 수행시의 최적 이온의 개수는 30,000임을 확인하였다.

Astract

In this paper, a new parallel computation method, which fully utilize the parallel processors both in mesh generation and FEM calculation for 2D/3D process simulation, is presented. High performance parallel FEM and parallel linear algebra solving technique was showed that excessive computational requirement of memory size and CPU time for the three-dimensional simulation could be treated successively. Our parallelized numerical solver successfully interpreted the transient enhanced diffusion (TED) phenomena of dopant diffusion and irregular shape of R-LOCOS within 15 minutes. Monte Carlo technique requires excessive computational requirement of CPU time. Therefore high performance parallel solving technique were employed to our cascade sputter simulation. The simulation results of Our sputter simulator allowed the calculation time of 520 sec and speedup of 25 using 30 processors. We found the optimized number of ion injection of our MC sputter simulation is 30,000.

* 正會員, 仁荷大學校 電子電氣컴퓨터工學部

(School of Electrical and Computer Engineering, Inha University)

※ 본 연구는 1998년도 인하대학교 교내연구비의 지원으로 수행되었습니다.

接受日字:1998年9月24日, 수정완료일:1999年2月26日

Ⅰ. 서 론

반도체 VLSI 칩을 제조하는 경우에 에피 성장 및 산화 이온 주입, 확산, 증착, 식각 등의 반도체 단위 공정을 거치게 되는데, 실제의 생산라인에서 제조하기에 앞서서 소자의 제조 공정을 컴퓨터를 이용한 프로세스 시뮬레이션을 수행하면, 실제로 LSI를 시험 제작

하여 전기적 특성을 검증하는 시간을 단축하고 모의 실험할 수 있기 때문에 반도체 소자의 제조 비용을 감소하고 제조 기간을 단축하는 것이 가능하게 된다. 또한, 공정 시뮬레이터로는 반도체 소자 내부의 물리량을 계산하고 있기 때문에, 반도체 소자 내부에서 불순물의 거동을 해석하는 것도 가능하게 된다. 반도체 공정을 컴퓨터로 모의 실험한다는 것은 반도체 공정의 3차원적 형상을 표현하는 물리적 미분방정식을 적절한 방법으로 계산하여 그 결과를 구한다는 의미이다. 복잡한 구조의 비선형 미분 방정식을 정확하게 계산하기 위해서 유한 요소법과 같은 수치해석 기법을 이용한다. 수치해석 계산 시간은 생성된 노드 수에 대해서 지수적으로 비례하므로 노드 수를 적절히 제어해야 하고, 또한 노드의 위치에 따른 메쉬 형태에 의존하여 해의 정확도가 변화되므로 메쉬의 형태를 제어해야 한다.

차기 세대 0.13 ~ 0.07 ㎛ CMOS 공정의 복잡성을 감안할 때, 소요되는 노드의 개수와 CPU 부담은 기하급수적으로 증가할 것으로 예측된다. 실제적으로, 반도체 공정을 시뮬레이션 할 때에 있어서 전체 실행 CPU시간 가운데 90% 이상을 PDE의 Assemble 단계 및 LU Decomposition 단계에서 사용하게 된다. 따라서, Sparse Jacobian Matrix를 Assemble하고 계산하는 과정을 병렬로 처리하기 위한 노력을 시도하고 있다. 즉, 현재 주요 TCAD Vendor 가운데 하나인, 미국의 Silvaco사는 Parallel Matrix Assembly와 Parallel Sparse LU Decomposition을 단일 프로세서가 탑재된 Silicon Graphics Machine을 여러개 병렬로 연결하여 계산한 결과를 발표한 바 있으며, 미국 스텐포드 대학에서는 각 병렬 컴퓨터의 부하를 적절히 조절하여 병렬 분산 효율성을 높이는 연구를 수행하고 있다[1] [2].

따라서, 본 연구에서는 불순물(B, P, As 등의 도판트와 점결함)의 농도 분포를 계산할 수 있는 3차원 병렬 도판트 시뮬레이터와 몬테카를로 스퍼터 반도체 공정 시뮬레이터를 위한 초고속 병렬 분산 처리기를 개발하고 그 결과를 보고하고자 한다.

Ⅱ. 병렬 처리용 메쉬 생성기 및 FEM 병렬 연산기

지금까지 병렬 유한요소법 수치해석 알고리즘은 다양하게 개발되어 왔으나, 대부분의 병렬 수치해석 알고리즘은 반복법(iteration method)을 사용하여 병렬 알고리즘을 개발하였다. 대표적으로 많이 사용되는 영역 분할법(domain decomposition method)은 전체 영역을 세부 영역으로 분할한 후, 각 세부 영역을 병렬로 계산하고 세부 영역간의 경계조건을 반복법(iterative method)으로 구하는 방법이다. 따라서, 계산 결과가 수렴할 때까지 매 반복 계산에서 데이터를 주고 받아야 하며, 매 반복 계산마다 프로세스를 동기(synchronization)시켜야 한다[3]. 결과적으로, 데이터의 전송 량이 많기 때문에 타 프로세서에 대한 의존성이 커지고, 각 프로세서에서의 수렴 속도가 일정하지 않을 경우 각 프로세서의 부하 균형(workload balance)이 이루어지지 않아 병렬처리의 효율성이 떨어지는 단점을 가지고 있다.

이러한 단점을 보완하는 방법으로, 데이터 전송을 최소화하고 각 프로세서의 독립성을 보장하는 하부 구조법(substructure method)이 있다[4]. 하부 구조법은 할당된 영역에서 시스템 방정식을 만들어서 각 하부 영역의 경계 노드만의 관계식을 만들고, 이 값들을 이용하여 전체 시스템의 계산을 수행한다. 가장 큰 장점은 프로세서간의 데이터 전송이 최소화된다는 것과, 전체 행렬 계산에서 내부 노드는 제외하고 경계 노드만으로 첫 번째 계산을 수행하므로 행렬 크기가 크게 감소하여 계산시간의 이득을 가져올 수 있다는 것이다.

그러나, 각 하부 영역의 내부 노드와 경계 노드의 번호 정의가 어려워 비구조형 메쉬(unstructured mesh)나 복잡한 형태의 경계에서 하부구조법을 적용하는데 어려움이 있어서 지금까지는 구조형 메쉬(sturctured mesh)에만 적용되어 왔고, 메쉬 생성에서 유한요소법 계산에 이르기까지 전체를 완전히 병렬화한 연구가 없었다. 따라서, 본 연구에서는 비구조형 메쉬 생성에서 유한요소법 계산에 이르기까지 전체 계산을 병렬화하여 프로세서의 효율을 극대화 할 수 있는 수정된 하부 구조법을 개발하였다.

1. 영역분할

그림 1에 본 연구에서 수행한 병렬 계산 순서도와 4개의 프로세서를 예로 하여 영역 분할한 결과를 도시하였다. 영역 분할은 주어진 프로세서의 수에 따라 수행하므로 4개의 프로세서가 할당되었다고 가정하여 전체 영역을 4개로 나누었다. 검은 색을 칠한 노드는 프로세서 사이의 경계를 의미한다.

이제 4개의 영역으로 나뉜 것 중에서 PE0에 해당하는 영역만을 보면, 분할된 영역에서 다른 프로세서에 독립적으로 메쉬를 형성하게 되는데, 이를 위해서 본 연구에서는 보로노이 다이아그램을 이용한 디라우니 메쉬 생성과 표면 전진 알고리즘(advancing front algorithm)을 이용한 메쉬 생성 기술의 2가지 방법을 개발하였다.

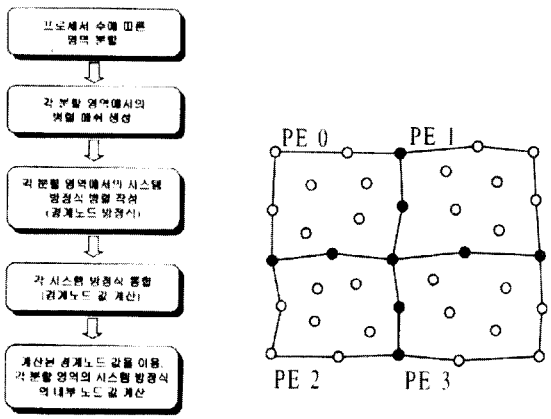


그림 1. 본 연구에서 수행한 병렬 계산 순서도와 4개의 프로세서를 예로 하여 영역 분할한 결과

Fig. 1. Schematic diagram of the flowchart for the parallel processing and of the exemplary structure.

2. 병렬 메쉬 생성

그림 2에 도시한 디라우니 메쉬 생성기는 먼저 난수 발생기를 이용하여 임의의 위치에 노드를 생성시킨 후 보로노이 다이아그램을 형성한 다음 디라우니 메쉬를 형성한다. 이와 같은 알고리즘은 일반적으로 많이 사용되는 방법이지만 경계영역을 정확하게 설정할 수 없다는 단점을 가진다. 따라서 디라우니 메쉬 생성기와 동시에 표면 전진 알고리즘을 이용 한 메쉬 생성기도 제작하였다.

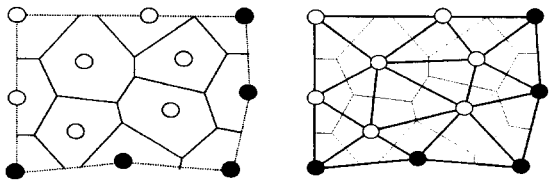


그림 2. 디라우니 메쉬 생성: (a) 보로노이 다이아그램, (b) 디라우니 삼각형 생성

Fig. 2. A Plot showing Delaunay mesh generation (a) Voronoi diagram, and (b) Delaunay triangulation.

그림 3에 표면 전진 알고리즘에 의한 메쉬 생성 기법을 도시하였다. 경계에 해당되는 메쉬는 메쉬가 없는 공간에 대하여 디라우니 조건을 만족하는 최적의 위치를 계산하고, 노드를 새로 생성한다. 새로 생성된 노드는 표면의 노드, 에지(edge)와 연결되어 새로운 메쉬를 형성한다. 이와 같은 방법을 계속 수행하여 결국 최종의 메쉬를 형성한다.

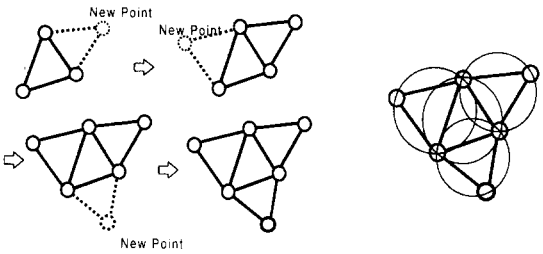


그림 3. 표면 전진 알고리즘을 이용한 메쉬 생성

Fig. 3. A plot showing a mesh generation with advancing front algorithm.

그림 4와 그림 5도에 상기 두가지 방법에 의한 메쉬 생성의 결과를 도시하였다. 그림 4는 2개의 하부 영역에서 보로노이 다이아그램 형성을 이용한 디라우니 메쉬 생성결과를 도시하였다. 경계에 해당되는 노드는 계산 초기에 먼저 정의되어, 각 프로세서가 데이터를 공유한다. 각 프로세서는 내부 노드만을 생성하고 노드 생성이 완료되면 보로노이 다이아그램을 계산한다.

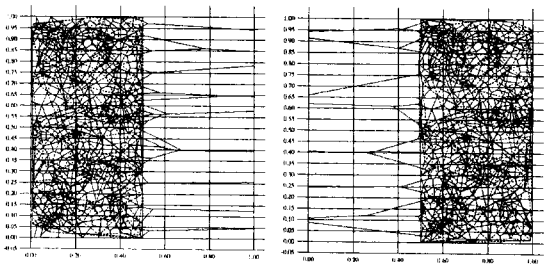


그림 4. 2개의 하부 영역에서 보로노이 다이아그램 형성과정을 통한 디라우니 메쉬 생성결과

Fig. 4. A Plot showing the Delaunay triangulation superimposed on the corresponding Voronoi diagram in two sub-region.

그림 5에는 표면 전진 알고리즘을 이용하여 생성한 메쉬의 예를 도시하였다. 유한요소법을 사용하여 계산을 수행할 때, 그다지 중요하지 않은 영역은 메쉬 간격을 넓게 하고 중요하게 계산될 영역은 메쉬 간격을

좁게 하면 계산 오차를 증가시키지 않으면서도 행렬 크기를 줄일 수 있어서 계산 효율을 높일 수 있다. 그림 5도의 왼쪽 아래 영역은 의도적으로 메쉬 크기를 줄인 결과를 보여준다.

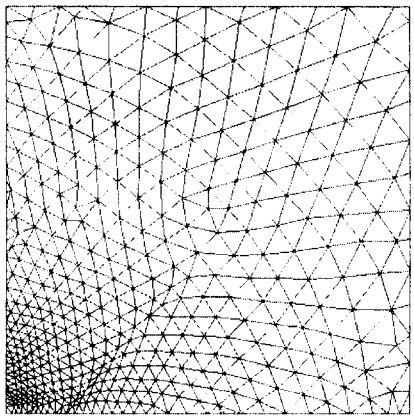


그림 5. 표면 전진 알고리즘을 이용한 메쉬 생성 결과
Fig. 5. A plot showing the Advancing front mesh generation.

3. 시스템 방정식 작성

각 영역의 메쉬를 형성한 후, 형성된 메쉬의 노드 순서를 적절히 조화시키면, 즉 내부 노드를 먼저 번호 매김하고, 그런 다음 경계노드를 번호 매김하면 유한요소법의 특성상 전체 행렬이 영역별로 형성 된다. 그림 6에는 처음에 가정한 네 영역중에서 PE0에 해당되는 영역만을 도시하였다. 그림 7의 오른쪽 강성 행렬(stiffness matrix)에서 4가지 영역으로 형성된 부 행렬(sub matrix) 중 첨자 ii는 내부 노드만에 의해 구성된 행렬이 되고, 첨자 bb는 외부노드 만의 값들로 형성된 부 행렬을 나타낸다. 대각의 ib와 bi는 내부 노드와 외부 노드가 상호 영향을 미치는 행렬로 구성되어 있다. 이렇게 만들어진 강성 행렬을 간단한 수학으로 다시 표현하면 외부 노드는 내부노드의 함수로 나타낼 수 있다. 따라서 A*와 b*는 내부 노드의 영향을 고려한 외부 노드만의 행렬식으로 다시 표현되고, 따라서, 그림 6의 마지막의 식과 같이 새로운 강성 행렬을 만들 수 있다.

4. 시스템 방정식 통합

앞서 구한 경계 노드만의 관계식을 이용하여 전체 영역을 다시 정의할 수 있다. 그림 7에 도시된 것과 같이 4개의 프로세서에서 만든 각각의 시스템 방정식을 다시 하나의 영역으로 조합한 시스템 방정식으로 나타내면, 처음에 각각의 프로세서에서 만들었던 강성 행렬 과 동일한 형태를 지니게 된다. 전체영역의 경계 노드는 경계조건에 의해서 주어지는 값이므로 Xb의 값은 알고있다. 따라서, Xi의 값은 다음과 같이 간단한 행렬 계산으로 구할 수 있다.

이와 같이 외부노드의 관계만으로 수정하면, 전체 N 개 노드 중 내부가 R개 경계가 N - R개의 노드를 가진다고 가정할 때, 행렬 계산에서는 크기의 제곱에 비례하여 계산시간이 증가하므로, R 만큼의 계산시간이 단축되는 것이 아니라 그 제곱만큼의 시간이 감소하는 효과를 얻을 수 있다.

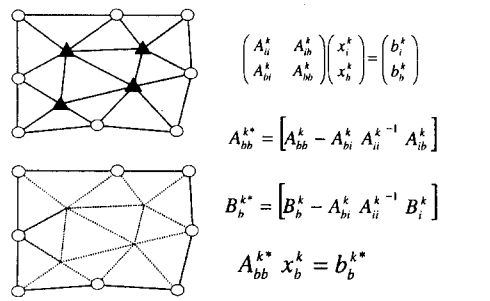


그림 6. 각 하부 영역의 시스템 행렬 생성
Fig. 6. A generation of system matrix for each sub-region.

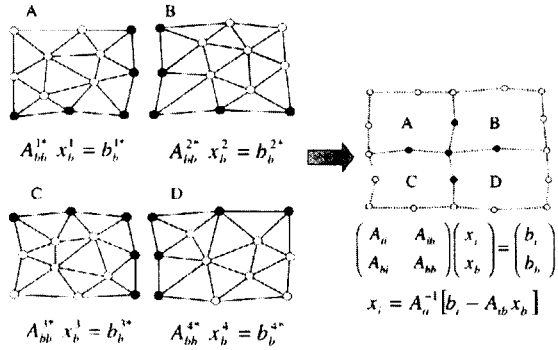
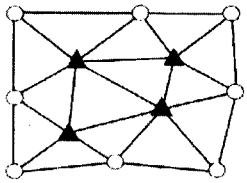


그림 7. 통합된 시스템 방정식 작성
Fig. 7. A schematic diagram showing integrated system equation.

5. 각 분할 영역의 내부 노드값 계산

이제 각 분할 영역의 경계 값은 앞서 언급한 방법을 이용하여 구한 후, 그림 8과 같이 각각의 하부 영역에서 독립적으로 내부 노드를 계산하게 된다. 앞서 전체 영역에서 계산한 방법과 마찬가지로 경계 영역의 값들을 알고 있으므로 단순한 행렬 계산만 가지고 내부 노드의 값을 구할 수 있다.

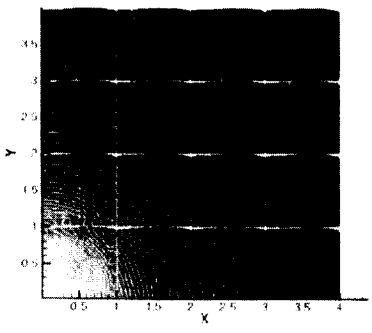


$$\begin{pmatrix} A_{ii}^k & A_{ib}^k \\ A_{bi}^k & A_{bb}^k \end{pmatrix} \begin{pmatrix} x_i^k \\ x_b^k \end{pmatrix} = \begin{pmatrix} b_i^k \\ b_b^k \end{pmatrix}$$

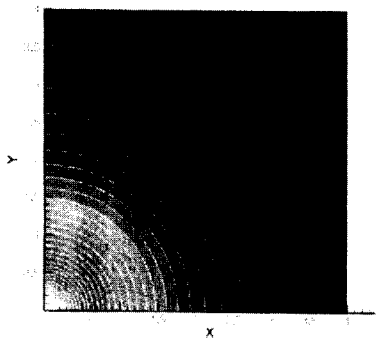
$$x_i^k = A_{ii}^{k-1} \left[b_i^k - A_{ib}^k \, x_b^k \right]$$

그림 8. 각 하부 영역의 내부 노드 값 계산
Fig. 8. A schematic diagram of internal node calculation for each sub-region.

본 연구에서 제안한 수정된 하부 구조법은 각 프로세서간의 데이터 전송이 단 일 회만 발생하므로 각 프로세서의 독립성을 크게 향상시킬 수 있고, 계산해야 하는 노드의 수를 크게 감소시킴으로써 초고속 병렬처리를 위한 효율성을 높일 수 있었다. 확산 미분 방정식을 수정된 하부구조법으로 계산한 결과가 그림 9에 도시되었다. x, y 좌표가 (0, 0)인 위치에 확산 원이 존재하고 주변으로 확산되는 과정을 16개의 CPU를 이용하여 계산한 결과이다. 병렬 연산은 CRAY T3E MPP 수퍼 컴퓨터를 사용하였고, MPI 병렬 인터페이스를 이용하여 병렬 프로세서 수를 제어하였다. 그림 9(a)는 디라우니 메쉬를 이용한 계산 결과이고 그림 9(b)는 표면 전진 메쉬를 이용한 계산 결과이다. 그림 9(a)의 계산 결과는 많은 오차를 가지고 있다. 이는 본 메쉬 생성 영역과 같이 사각형의 영역을 정의하는 경우 디라우니 메쉬 생성에서 생성 알고리즘 상 경계 영역을 정확히 정의하기가 어려울 뿐 아니라 수렴에 문제점을 가지고 있기 때문이다. 따라서, 그림 9(b)와 같이 표면 전진 알고리즘을 이용한 메쉬 생성 기법을 이용하여 영역을 정확히 정의하였다.



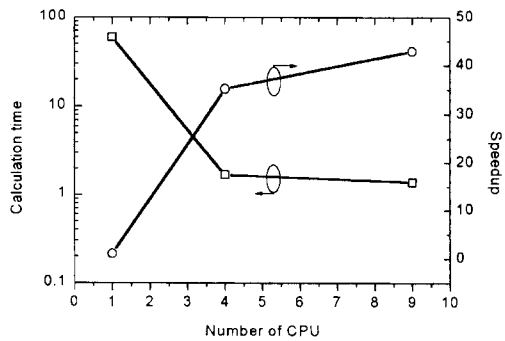
(a)



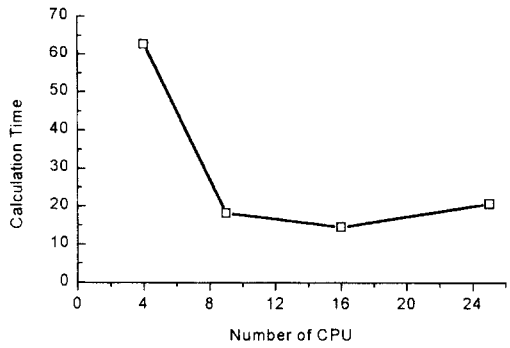
(b)

그림 9. 16개의 프로세서로 계산한 확산 결과: (a) 디라우니 메쉬 생성 결과 (b) 표면 전진 메쉬 생성 결과
Fig. 9. Simulation results of diffusion with 16 CPU by (a) Delaunay mesh generation method, and (b) advancing front method.

병렬 수행에 따른 계산 시간의 효율성을 측정하였다. 그림 10에 프로세서 수에 따른 계산 시간 및 스피드업 특성을 비교하였다. 단일 프로세서로 계산하는 것에 비해서 4개의 병렬 프로세서로 동일 계산을 수행하였을 때, 35의 스피드 업 특성을 얻었다. 이는 강성 행렬의 계산에 있어서 행렬의 크기가 감소함에 따라서 제곱에 비례하는 계산 시간의 감소 때문인 것으로 판단되며, 따라서, 병렬 연산을 통하여 효과적인 계산 시간의 절감을 가져올 수 있다는 결론을 내릴 수 있다. 그러나, 그림 10(b)에 도시된 바와 같이 계산량에 따라 적절한 병렬 프로세서 수를 결정하지 않으면 오히려 효율성을 떨어뜨리는 결과를 보인다. 이는, 계산 량에 비해서 데이터 전송 량이 상대적으로 많아지면 프로세서가 데이터를 전송하거나 받는 시간 동안 대기 상태로 있기 때문이다. 따라서, 계산량에 따라서 프로세서 수를 적절히 조절함으로써 최적의 계산효율을 얻을 수 있다는 결론을 얻을 수 있었다.



(a)



(b)

그림 10. 계산 프로세서 수에 따른 수행 시간 및 스피드 업 특성 비교

Fig. 10. The characteristics of calculation time and speedup as a function of number of CPU.

Ⅲ. 초고속 병렬 연산 도판트 시뮬레이터

병렬 컴퓨팅 환경 개발을 위해서 이미 작성된 시리얼 코드를 병렬화 하는 작업은 상당한 시간을 요하며, 거의 프로그램을 다시 개발하는 것과 같은 막대한 노력이 필요하다. 따라서, 기존의 코드를 병렬 코드로 변형시키는 노력 대신에 손쉽게 효율성을 높이고, 병렬 컴퓨팅의 장점을 살릴 수 있는 방법을 모색하였다[8].

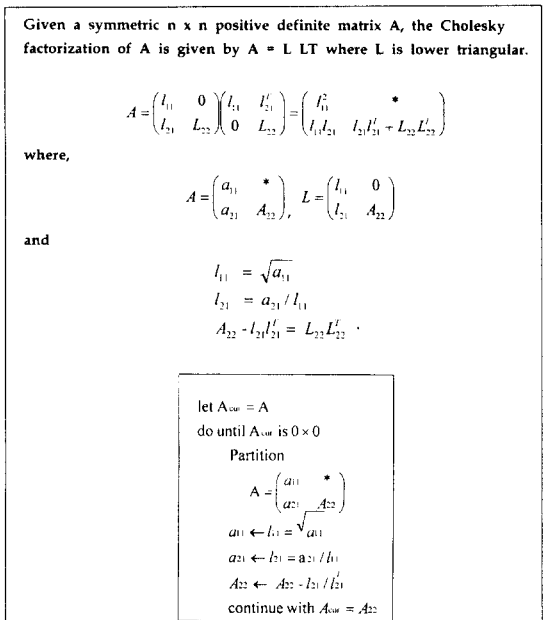


그림 11. 행렬 계산 알고리즘

Fig. 11. Algorithm for matrix calculation.

그 결과로, 앞장에서 제시한 방법과 같이 병렬 수치해석 프로그램을 새로이 제작함과 동시에, 병렬 행렬 계산기를 개발하였다. 도판트 시뮬레이터는 유한 요소법을 이용한 수치해석 알고리즘을 사용하므로 행렬 계산에 대부분의 시간을 소비한다. 따라서 행렬 계산 알고리즘을 병렬화 하면 비록 최적의 병렬 코드는 아니지만 효과적인 결과를 얻을 수 있었다. 그림 11에 행렬 계산을 위한 알고리즘을 도시하였다[5].

그림 12에 노드 수가 각각 1000, 3000, 5000일 때, 병렬 연산으로 수행한 행렬 계산 결과를 보여주고 있다. 노드 수가 1000개일 때, 풀어야 하는 행렬 크기는 1000 x 1000이 되고, 각 노드는 더블로 정의되어 있으므로 필요한 메모리 사이즈는 1 x 10^6이 된다. 따라서, 노드 수가 3000개를 초과했을 때, 메모리 문제 때문에 계산을 수행하지 못하였다.

노드가 1000개 일때, 즉 1000 x 1000행렬을 계산하는 데는 많은 시간이 소요되지 안았지만, 노드가 증가함에 따라서 계산시간은 크게 증가함을 볼 수 있다. 한 예로 8개의 프로세서인 경우 1000개의 노드를 계산하는데는 약 3.5초가 소요되었지만 3000개의 노드를 계산하는 데에는 40초, 5000개의 노드를 계산하는 데는 156초의 시간이 소요되었다. 본 계산에서는 시간 의존적 2차 미분 방정식을 계산하므로 이와 같은 행렬을 수십 차례 수행한다. 따라서 병렬 계산을 통해서 상당히 많은 시간을 효과적으로 감소시킬 수 있었다.

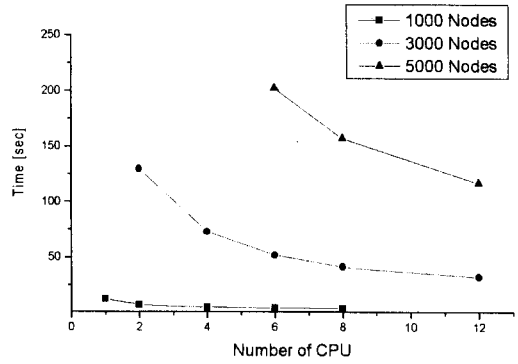


그림 12. 프로세서 변화에 따른 계산 수행 시간 특성

Fig. 12. The characteristics of calculation time as a function of number of CPU.

1. 과도 증속 확산(TED) 시뮬레이션

INPROS 확산 시뮬레이터는 반도체 소자의 공정 및 소자 특성을 분석하기 위하여 그림 13과 같은 과정으로 계산을 수행한다. 초기에 사용자의 입력 사항에 준하여 INPROS 메쉬 생성기로 3차원 메쉬를 형

성한 다음, 3D INPROS 확산 계산기에서 유한요소법(FEM) 수치해석 방식으로 확산 계산을 수행한다. 3차원 메쉬 생성기는 사면체를 기본형으로 사용하며, 구조형 메쉬 생성기 및 비구조형 메쉬 생성기를 가지고 있다. 또한 적응 메쉬 생성기능을 포함하고 있어, 불필요한 메쉬의 삭제나 정확도의 증가를 위해 노드 삽입 기능을 가지고 있다. 확산 모델은 고농도 확산 의존성 및 불순물의 전계 의존성, 점결함 의존성, 접합면에서의 분리 현상(segregation), 점결함의 벌크 재결합, 표면 재결합 등의 모델을 포함하고 있다. 3차원 계산은 막대한 계산 시간을 요함으로 병렬 FEM 계산기를 개발하였다. 유한요소법 수치해석기는 전체 강성 행렬을 푸는데 대부분의 계산 시간을 소비한다. 따라서, 병렬 행렬 계산기를 INPROS에 접목시켜 계산시간을 단축시켰다. 확산 공정 시뮬레이션 결과를 이용하여 문턱전압의 변화를 확인하기 위하여 2차원 소자 시뮬레이터인 PISCES-IIB에 INPROS 시뮬레이션 결과를 입력한다. 3차원 데이터를 2차원으로 변환하기 위하여 데이터 변환기를 거치게 된다. PISCES의 결과를 이용하여 문턱전압을 추출하고, RSCE 효과를 관측할 수 있도록 시스템 환경을 구축하였다.

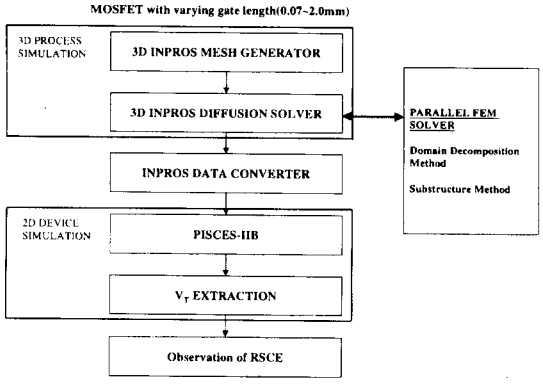


그림 13. 과도증속확산 시뮬레이션의 흐름도
Fig. 13. A schematic diagram of flowchart for transient enhanced diffusion.

그림 14는 실리콘 기판에 붕소를 1×10^{18} cm^{-3} 농도로 에피 성장시키고, 인을 1×10^{14} cm^{-2} 의 도우즈, 50 keV의 에너지로 이온주입한 후, 750 ℃에서 2시간 동안 열처리한 후의 불순물의 분포이다. 이온주입으로 이한 격자 구조의 손상이 없을 때는 750 ℃로 열처리하였을 경우, 붕소의 재확산이 거의 발생하지 않았으며, 점결함을 고려하지 않은 시뮬레이션 결과도 전혀 재확산이 생기지 않음을 확인하였다. 그러나, 붕소나 인이 이온주입되면, 점결함이 발생하고, 생성된 점결함은 그 확산 속도가 불순물에 비해 매우 빨라서 기판의 내부에서 균일하게 분포하게 된다. 그림 14의 결과와 같이 이온 주입 공정 후의 붕소의 불순물 분포는 그 형태가 초기 형태와 크게 차이가 남을 확인할 수 있었고, 이는 SIMS의 결과와 일치함을 확인 하였다[6]. 저온에서의 점결함에 의한 문턱전압의 변화를 확인하기 위하여 일반적인 CMOS 공정을 시뮬레이션 하였다. CRAY T3E MPP 병렬 컴퓨터를 이용하여 시뮬레이션하였으며, 직렬 계산을 하였을 경우 3시간이 소요되었고, 8개의 CPU로 병렬 컴퓨팅하였을 경우 15분이 소요되었다. 이 결과를 이용하여 INPROS의 데이터 변환기로 2차원 단면에 대한 데이터 포맷을 만들고, 2차원 PISCES 소자 시뮬레이터를 이용하여 소자 특성을 시뮬레이션하였다.

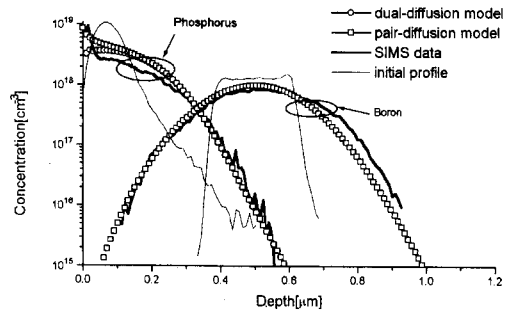


그림 14. 1×10^{14} cm^{-2}의 도즈와 50keV의 에너지로 인을 이온주입한 후, 750℃에서 2시간동안 열처리공정을 수행하였을 때의 붕소 확산 분포도
Fig. 14. A plot showing boron profiles after implanting phosphorus with a dose of 1×10^{14} cm^{-2} and an energy of 50 keV, followed by annealing at 750℃ for 2 hours.

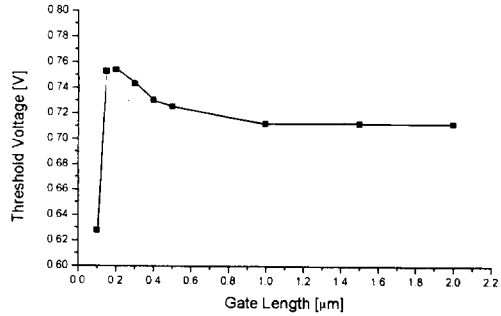


그림 15. 게이트 길이의 변화에 대한 NMOS 소자의 문턱전압 변화 특성
Fig. 15. Threshold voltage of NMOS devices as a function of gate length.

그림 15도에 계산된 문턱전압을 도시하였다. 채널 길이가 0.5㎛이하에서 채널 길이가 감소할수록 문턱전압이 계속 증가하여 RSCE 현상을 보임을 관측할 수 있었다.

2. R-LOCOS 산화 시뮬레이션

본 연구진이 개발한 산화 공정 시뮬레이터를 이용하여 미세 소자 제조에 있어서 소자 격리 구조로 적용되고 있는 R-LOCOS(Recessed Local Oxidation of Silicon)의 시뮬레이션을 수행하고자 식각된 실리콘 구조에 대한 3차원 산화 공정을 수행하였다.

R-LOCOS 성장을 위한 초기 구조로서, 질화막 마스크 두께 2,000Å, 패드 산화막 200Å의 구조에서 초기에 실리콘 기판을 2,000Å 식각하였다. 이러한 구조에 대하여 1,000℃에서 60분간 습식 산화 공정을 수행한 결과를 그림 16에 도시하였다.

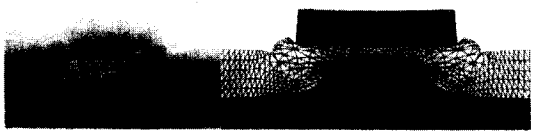


그림 16. R-LOCOS 시뮬레이션 결과와 실험 결과 비교
Fig. 16. A comparison of R-LOCOS between simulation and SEM.

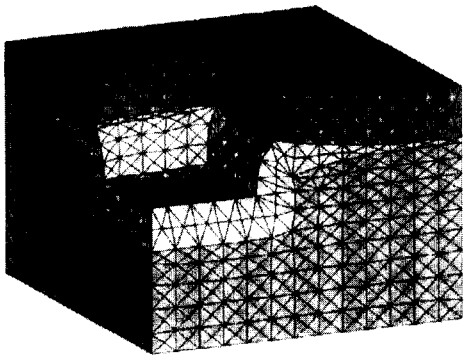


그림 17. Hole 구조 마스크에 대한 R-LOCOS 성장 시뮬레이션 결과
Fig. 17. A simulation results of R-LOCOS process for hole structure.

R-LOCOS 시뮬레이션 결과와 실험 결과를 비교한 결과, 시뮬레이션의 결과가 잘 일치함을 볼 수 있다[17]. 그림 17에 나타낸 것은 Hole 구조의 마스크에 대한 R-LOCOS 성장 시뮬레이션 결과이다. R-LOCOS 시뮬레이션 결과에서 대각선 방향 단면의 새부리가 더 짧음을 확인하였다. 이것은 산화제의 확산 분포를 통하여 설명할 수 있는데, 마스크의 코너 영역에서 산화제의 분포가 상대적으로 적기 때문이고, 코너 영역의 위쪽에서는 주변에 분포하는 tensile 스트레스가 사라지고, 아래쪽 영역에서는 강한 compressive 스트레스의 분포가 나타나기 때문이다.

Ⅳ. 초고속 병렬 연산 스퍼터 시뮬레이터

플라즈마 소스에서 입사한 이온과 충돌하여 방출되는 타겟 입자의 분포를 계산하기 위한 3차원 스퍼터 시뮬레이터는, 먼저 시뮬레이터 실행을 위한 입력 데이터를 지정한 다음, 입자 충돌간의 핵 에너지 손실 및 전자 에너지 손실을 계산하고, 충돌 산란각을 결정한다. 연쇄 충돌(collision cascade)을 위한 알고리즘 및 충돌 후 입자들의 자취를 결정하고 스퍼터 되어 표면을 떠나는 입자의 에너지, 방출각, 평균 스퍼터율을 결정하게 된다[9]. 그림 18의 순서도에서 몬테카를로 방법을 이용한 병렬 처리 알고리즘을 도시하였다. 초기에 사용자의 공정조건을 입력받는다. 그런다음, 병렬처리를 위해 사용하고자 하는 프로세서를 설정하고, 초기화 시킨다. 각각 연산이 할당된 프로세서는 다른 프로세서와는 독립적으로 계산을 수행한다. 지정된 개수 만큼의 연산을 끝내면 그 결과를 중앙 프로세서에 전송하고, 중앙 프로세서는 전송받은 데이터를 적절히 처리한 다음 후처리계로 데이터를 전송한다.

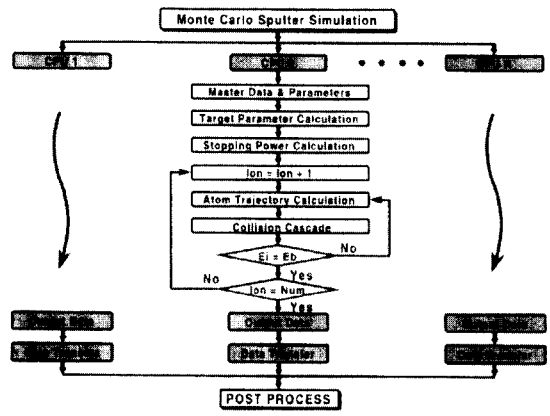


그림 18. 몬테카를로 기법을 이용한 병렬 처리 알고리즘 순서도
Fig. 18. A schematic diagram of flowchart for parallel Monte Carlo algorithm.

몬테카를로 방식은 통계적 방식의 수치해석 프로그

랩이므로 각 프로세서가 주어진 조건에 따라 독립적으로 통계적 계산을 수행하므로 프로세서 상호간의 데이터 전달 시간이 계산시간에 비해 매우 적으므로 매우 높은 효율을 얻을 수 있다. 프로세서 수에 대한 계산 수행시간 및 스피드업(speedup) 특성을 그림 19에 도시하였다. 본 결과는 입사이온의 개수가 3,000개일 때의 결과이다.

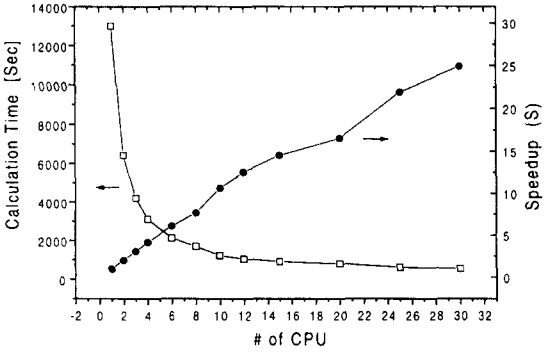


그림 19. 프로세서 수에 대한 계산시간 비교 및 스피드업(speedup) 특성

Fig. 19. The characteristics of calculation time and speedup as a function of number of CPU.

그림 20에 입사 이온에 대한 방출 입자의 형태를 도시하였다. 스퍼터율은 단위 입사 이온에 대한 타겟 방출 원자의 수로 정의된다. 타겟은 비정질이고 타겟 원자의 질량, 원자 번호, 원자 전위, 원자 밀도, 표면 구속 에너지(surface binding energy) 등과 같은 타겟에서의 여러 가지 정보는 시뮬레이터의 입력 함수로써 정의하였다. 마찬가지로, 타겟으로 입사하는 이온에 대한 정보도 자동으로 입력되도록 하였다. 스퍼터 타겟으로는 Cu, Al을 사용하였고, 입사하는 이온은 Ar을 사용하여 시뮬레이션 하였다.

입사이온의 개수를 증가시킬수록 에러가 감소하고, 보다 신뢰성있는 결과를 얻을 수 있었다. 그러나, 그에 따른 계산시간이 크게 증가하므로 결과의 신뢰성을 보장받는 한계 내에서 최적화 된 이온의 개수를 얻어내야 한다. 그림 21에 이온의 개수를 증가시켰을 때 감소되는 에러의 변화를 도시하였다. 그림에 도시된 바와 같이 10,000개의 이온에서 이온 수가 증가할수록 발생되는 에러의 수는 선형적으로 감소하다가 30,000개 이상의 이온에서는 에러의 변화가 거의 없어진다. 따라서, 30,000개 이상의 이온에서는 동일한 결과를 얻었고, 30,000개의 이온 주입이 계산의 효율을 최대한 높일 수 있는 최적치임을 확인하였다.

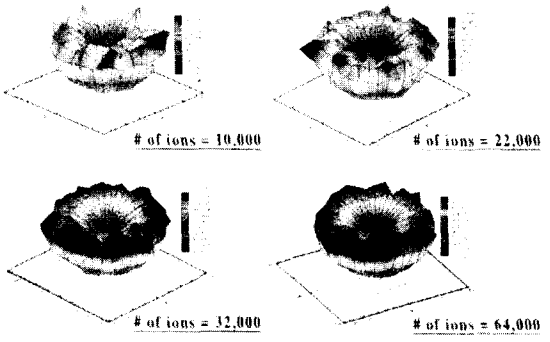


그림 20. 입사 이온 수의 변화에 대한 방출 입자의 형태 변화

Fig. 20. Plots showing the distribution dependency of ejected particles on number of ion.

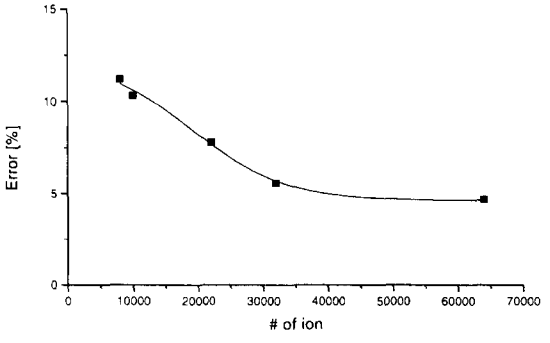
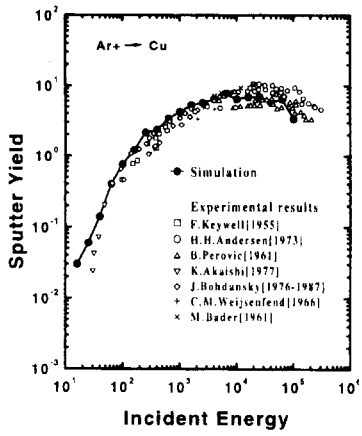


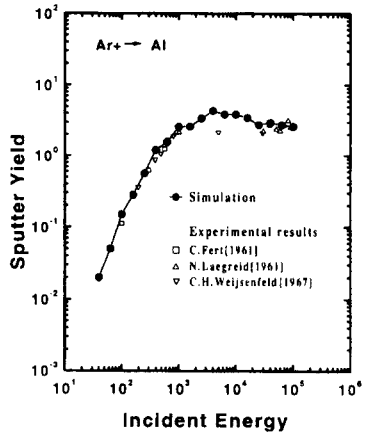
그림 21. 이온의 개수를 증가시켰을 때 감소되는 에러의 변화

Fig. 21. The error dependency on the number of ions.

그림 22는 수직으로 입사하는 Ar 이온의 에너지에 대한 타겟 원자의 스퍼터율을 나타낸다. 본 연구의 계산 결과에 나타난 바와 같이 입사 이온의 에너지가 커질수록 타겟에서 발산되는 입자의 분포도 많아진다. 이온의 에너지가 커짐에 따라 타겟으로 전달되는 에너지도 커지게 되고, 연쇄 충돌 효과가 많아짐으로 스퍼터율이 증가하게 된다. 그러나, 고에너지 영역(10KeV 이상)에서는 스퍼터율이 점차 감소하는 경향을 보였는데, 입사 에너지가 커지면 산란각에는 영향을 미치지 않는 전자 에너지 손실이 핵 에너지 손실에 비해서 더욱 커지므로 이온은 타겟으로 더욱 깊이 들어가고 타겟 원자의 재 산란은 타겟 내부에서 발생되므로 스퍼터율은 작게 된다. 본 시뮬레이터에서 수행한 Ar 입사 이온의 에너지에 대한 Cu, Al 타겟의 스퍼터율 분포는 실험 결과와 일치함을 확인하였다.



(a)



(b)

그림 22. 수직으로 입사하는 Ar 이온의 에너지 [eV] 에 대한 타겟 원자의 스퍼터율 [#/ion] 분포: (a) Cu 타겟, (b) Al 타겟에 Ar이 입사하는 경우 입사 에너지 변화에 따른 스퍼터율의 의존성을 보여주고 있다.

Fig. 22. Sputtering yield distributions versus the incident energy of Ar ion at normal incidence: It is shown that the sputter yield of (a) Cu target, and (b) Al target depend upon Ar ion energy.

V. 결 론

디라우니 병렬 메쉬 생성기 및 표면 전진 메쉬 생성기를 제작하여 병렬 유한요소법 수치해석기에 사용이 적합한 메쉬 생성기를 개발하고, 수정된 하부구조법 병렬 유한요소법 수치해석기를 개발하여, 반도체 공정 시뮬레이션의 효율성과 성능을 향상시키기고자 하였다. 개발된 병렬 메쉬 생성기 및 병렬 유한요소법 수치해석기는 기존의 병렬 컴퓨팅 알고리즘에 비해 병렬 프로세서의 독립성을 높이고 데이터 전송을 최소화하는 특징을 가진다. 또한, 기존의 직렬계산 코드를 병렬 연산하기 위하여, 유한요소법 계산에 있어서 가장 많은 계산 시간을 요하는 행렬 계산 루틴을 병렬화 하였다. 이를 확산 및 산화 시뮬레이터에 적용하였으며, 직렬 계산시 3시간이 소요되는 확산 시뮬레이션의 결과가 8개의 프로세서를 사용하여 15분만에 계산되었다. 과다한 계산 시간을 요하는 몬테카를로 수치해석 방법의 효율성을 높이고자 병렬 연산 알고리즘을 개발하였다. 스퍼터 시스템의 타겟 입자 분포 특성을 병렬 연산 몬테카를로 방식으로 계산하였다. 3000개의 이온을 주입하였을 경우 단일 프로세서에서 13,000초의 계산시간이 소비되었으며, 30개의 프로세서를 사용하였을 때 520초의 시간을 소비하여 25의 스피드업 특성을 얻었다. 또한, 몬테카를로 계산의 최적화 연구를 통해서 3차원 스퍼터 연쇄충돌 계산 수행시의 최적 이온의 개수는 30,000임을 확인하였다. 본 병렬 연산 연구를 통하여 과다한 메모리 및 계산 시간을 요하는 3차원 반도체 공정 시뮬레이션 문제를 해결할 수 있는 방법을 제시하였다.

참 고 문 헌

[1] R. Strasser and S. Selberherr, "Parallel and distributed TCAD simulations using dynamic load balancing," *Proc. of SISPAD 98*, pp. 89-92, 1998.

[2] A. Schwerin and A. Spitzer, "Industrial demands on process and device simulation," *Proc. of SISPAD 98*, pp. 348-355, 1998.

[3] Hideo Fukumori, Yoichi Kono, and Yoichi Muraoka, International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN 96), pp. 124-130, 1996.

[4] St. Doltsinis and S. Nolting, Computer Methods in Mechanics and Engineering, vol. 89, pp. 497-521, 1991.

[5] P. Alpatov, et al., Proc. of the SIAM Parallel Processing Conference, 1997

[6] J. Lee, S. Yoon, Y. Kim, T. Won, J. Kim, and D. Lee, Int. Conference on SISPAD 97, pp. 301-303, 1997.

[7] S. Yoon, J. Lee, T. Won, B. Cho, J. Kim,

and D. Lee, "3-D Adaptive Simulation of Thermal Oxidation Process," Proceedings of the 26th European Solid State Device Research Conference (ESSDERC'96), pp. 351-354, September 1996.

[8] Marc Snir, "MPI: The Complite Reference," The MIT Press, 1996.

[9] Y. Ban and T. Won, "3D Modeling of Sputter Process with Monte Carlo Method," Simulation of Semiconductor Processes and Devices 1998 (SISPAD 98), pp. 161-164, Belgium, September 2-4, 1998.

저 자 소 개

李 濟 熙(正會員) 第 35卷 D編 第 5號 參照

1968년 10월 5일생. 1992년 인하대학교 응용물리학과(공학사). 1994년 인하대학교 전자재료공학과(공학석사). 1994년~1999년 인하대학교 전자재료공학과 (공학박사). 주관심 분야는 반도체 소자 및 공정 시뮬레이션, 병렬 컴퓨팅 등

元 太 映(正會員) 第 35卷 D編 第 5號 參照

1959년 2월 21일생. 1981년 서울대학교 전자공학과(공학사). 1983년 한국과학기술원 전기 및 전자공학과(공학석사). 1989년 미국 University of Illinois at Urbana-Champaign 전자공학과(공학박사). 1989년 ~ 1990년 미국 State University of New York 조교수. 1990년 ~ 1991년 삼성전자(주) 수석 연구원. 1991년 ~ 현재 인하대학교 공과대학 전자전기컴퓨터공학부 부교수. 주관심 분야는 반도체 소자 및 공정 등임

潘 用 瓚(正會員) 第 35卷 D編 第 5號 參照

1974년 7월 7일생. 1997년 인하대학교 전자재료공학과(공학사), 1997년 ~ 1999년 인하대학교 전자재료공학과(공학석사), 1999년 ~ 현재 인하대학교 전자재료공학과 박사과정. 주관심 분야는 반도체 공정 물리, 시뮬레이션 등임