

반도체 공정을 위한 OpenMP와 MPI 기반의 FDTD 시뮬레이션 연산 환경 구축

Implementation of FDTD simulation using OpenMP and MPI for semiconductor process

이 승 일, 이 철 훈
충남대학교

Lee seung-il, Lee cheol-hoon
Choongnam Univ.

요약

반도체 공정에서는 소자 내부의 물리량 계산을 통해 결점을 검출하는 시뮬레이션을 수행하게 된다. 이를 위해 3차원적 형상을 표현하여 물리적 미분 방정식을 계산하는 유한 차분 시간 영역 알고리즘(Finite-Difference Time-Domain, 이하 FDTD)과 같은 수치해석 기법이 사용된다. 반도체 테스트 범위의 확장으로 인해 시뮬레이션 사이즈 또한 커지고 있는 추세이다. 이에 따라 하나의 프로세서에서 수행할 수 없는 문제가 발생하기도 한다. 이를 해결하기 위해 본 논문에서는 openMP와 MPI를 이용한 하이브리드 컴퓨팅 기법을 바탕으로 다중 GPU 제어를 통해 시뮬레이션 환경을 구축하였으며 정상 동작함을 확인하였다.

I. 서론

최근 발달하는 컴퓨터 성능에 맞추어 어플리케이션에서 처리해야 하는 데이터의 연산량이 점차적으로 증가하는 추세이다. 이에 따른 요구에 맞추어 높은 성능을 가지는 단일 컴퓨터 한 대를 이용한 순차 프로그래밍에서 네트워크를 통해 연결되어있는 다수의 컴퓨터를 이용한 혹은 단일 컴퓨터 내의 다수의 코어를 이용하는 병렬, 분산 프로그래밍이 발전하고 있다. GPGPU를 이용한 성능개선의 대상 중 하나로, 공정 시뮬레이터로는 반도체 소자 내부의 물리량을 계산하고 있기 때문에 반도체 소자 내부의 불순물의 거동을 해석하는 것이 가능하다. 이러한 모의 실험은 3차원적 형상을 표현하는 물리적 미분방정식을 계산하여 그 결과를 구하게 되는데, 복잡한 구조의 비선형 미분 방정식을 정확하게 계산하기 위해 FDTD와 같은 수치해석 기법에 적용 가능하다. 반도체 집적도 향상으로 인하여 테스트를 위한 3차원 매트릭스의 사이즈가 증가하는 추세이고, 큰 사이즈의 FDTD를 수행하기 위하여 OpenMP 및 MPI와 이를 총괄하는 클러스터링 기법이 진행되어 왔다.

본 논문에서는 2장에서 관련 연구로서 FDTD, OpenMP, MPI에 대해 소개하고, 3장에서 OpenMP와 MPI를 이용한 FDTD 알고리즘 설명을, 4장에서는 테스트 결과를 보이며, 마지막으로 5장에서는 결론 및 향후 연구과제를 기술한다.

II. 관련연구

1. FDTD

1966년 Yee가 처음으로 제안한 FDTD는 맥스웰(Maxwell) 방정식을 시간영역 상에서 수치 해석적으로

풀어내는 방법이다. 알고리즘이 간단하며 전파해석과 관련된 다양한 문제에 적용할 수 있는 유연성 때문에 FDTD는 전자기파 산란 해석에 광범위하게 쓰인다. FDTD 방법은 수치해석으로써 임의의 구조를 갖는 물체의 마이크로파 해석에 유용하고 복잡한 구조를 가지는 광소자의 마이크로파 해석에 적합하다. 해석하고자 하는 대상에 FDTD를 적용하여 프로그래밍 하기 위해서는 먼저 구조적으로 격자화하고, 각 위치에서의 매질을 결정하고, 계산 시간 간격 등의 초기 조건을 결정하여야 한다. 그 후 각 위치에서의 자계를 계산하고 계산된 결과를 사용하여 전계를 계산한다. 다음으로는 경계조건을 적용하고 전원을 인가하는 과정을 일정 시간 동안 반복하여 수행한다.[1]

2. MPI

MPI는 집합 연산, 사용자 정의 데이터 타입과 토폴로지, 다양한 방식의 통신 등의 기능들과 이기종 병렬 아키텍처(Heterogeneous Parallel Architecture)에 대한 지원 등을 제공하며, 코드가 실행되는 동안 프로세스의 수를 변화시키는 동적 프로세스 관리, 원격 메모리 접근 병렬 I/O 등의 지원을 목표로 한다. MPI는 각 프로세스에 하나의 등급(RANK)과 통신 그룹을 부여하여 점대점(point-to-point) 통신을 가능하도록 한다.

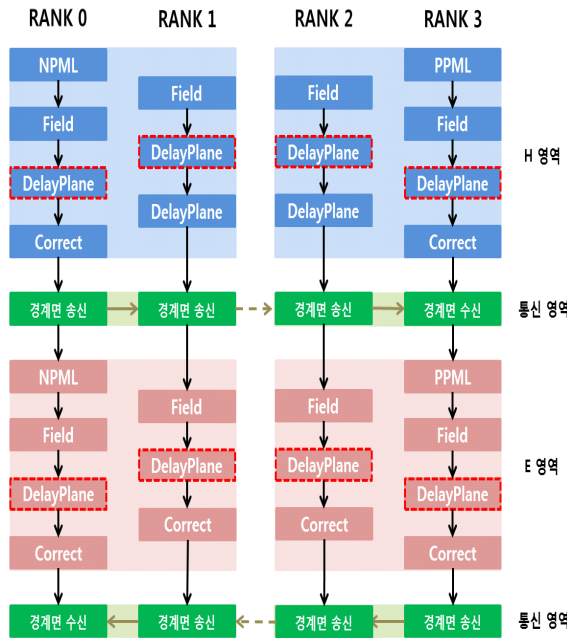
3. OpenMP

OpenMP는 공유 메모리 형태 멀티프로세서의 병렬 프로그래밍 모델을 위한 API(Application Programming Interface)로 SMP(Symmetric Multiprocessing)타입의 컴퓨터 아키텍처에서 C, C++, Fortran 언어를 지원하며, 어플리케이션이 이용할 수 있도록 컴파일로 디렉티브를 삽입함으로써 서버루틴 라이브러리를 덧붙여 병렬 처리가

가능하도록 한다. 이를 확장하여 컴퓨터 클러스터 상에서 구동할 수 있다.

III. 본론

본 논문은 OpenMP와 MPI를 동시에 사용하는 경우를 설명하기 위해 두 대의 GPU를 가지는 두 대의 컴퓨터를 사용하는 것으로 가정하여 설명한다.



▶▶ 그림 1. OpenMP와 MPI를 사용한 FDTD

그림 1. 은 OpenMP와 MPI를 동시에 사용하여 수행하는 FDTD의 알고리즘이다. 전계 영역과 자계 영역에 대한 결과값을 얻기 위해 PML, Field, Delay, Correct 연산을 시뮬레이션을 위한 사이클 수만큼 반복적으로 수행한다.

MPI를 사용하기 위해 사용되는 각각의 GPU에 RANK라는 식별자를 사용하여 구별하고 각 GPU는 RANK에 맞게 분할되어 할당받은 매트릭스의 시뮬레이션을 수행한다. 시뮬레이션 간 전계(H), 자계(E)영역 연산 후 각 스텝마다 GPU의 RANK에 맞추어 이전 RANK에 위치한 매트릭스 연산의 결과값을 받아들이고 자신의 매트릭스 연산에 대한 결과값을 다음 RANK에 전송하는 과정을 수행한다.

이러한 결과값 송·수신 과정에서 인접한 RANK가 동일한 컴퓨터 내에 존재할 경우 OpenMP를 이용하여 공유메모리를 통한 결과값 송·수신을 한다. 다른 컴퓨터에 존재할 경우 MPI를 이용하여 결과값을 송·수신 한다[2].

NPML의 경우, 매트릭스의 가장 아랫면을 담당하는 RANK의 GPU만 수행하며 PPML는 가장 윗면을 담당하는 RANK의 GPU가 수행한다. DelayPlane의 경우 해당 Z축을 포함하는 RANK에서만 수행하게 된다.[3]

IV. 실험 환경 및 결과

표 1. 테스트 종류

테스트 번호	테스트 방법
1	단일 GPU 사용
2	OpenMP 사용
3	MPI 사용
4	OpenMP + MPI 사용

표1은 OpenMP와 MPI를 혼합하여 구현한 알고리즘의 정확성을 파악하기 위해 구성한 테스트 종류이다.

표 2. 테스트 종류별 출력되는 Intensity 비교

테스트 사이클	1	2	3	4
1	3.0000	3.0000	3.0000	3.0000
2	2.9252	2.9252	2.9252	2.9252
3	2.4403	2.4403	2.4403	2.4403
4	1.7188	1.7188	1.7188	1.7188
5	1.0450	1.0450	1.0450	1.0450

표2는 각 테스트 수행 시 사이클마다 출력되는 Intensity값이며, 테스트의 방법에는 차이가 있지만 모두 동일한 결과를 출력함을 알 수 있다.

V. 결론

본 논문에서는 반도체 공정에서 결점 여부를 파악하는데 사용되는 FDTD 시뮬레이션을 다중의 GPU를 사용하여 매트릭스를 분할 후 각 GPU에 할당하여 결과값 송·수신을 가능하게 하였다. 또한 분할된 매트릭스를 담당하는 GPU의 위치(RANK)에 따른 FDTD 수행 방식을 고려해 한 대의 GPU를 사용하여 FDTD를 수행한 결과와 일치함을 확인 하였다. 동일한 컴퓨터에 위치한 GPU간에는 OpenMP를 그렇지 않은 경우에는 MPI를 이용하여 각 GPU 간 FDTD 시뮬레이션 수행에 필요한 결과값을 송·수신을 통하여 가능하게 하였다.

향후 연구과제로는 결과값 송·수신에 소요되는 시간을 단축하여야 할 것이다.

■ 참고 문헌 ■

- [1] David M. Sheen, Sami M. A., Mohamed D. A., 'Application of the three-dimensional Finite-Difference Time-Domain method to the analysis of planar microstrip circuits', IEEE Transaction on Microwave Theory and Techniques, vol. 38, pp 849-857, 1990.
- [2] 박종현, 이정수, 김준성 "클러스터 환경의 병렬처리를 통한 FDTD 알고리즘의 성능 향상 분석", 대한전자공학회 추계학술대회, 제32권, 제2호.
- [3] 정복재 "CUDA로 구현한 FDTD 알고리즘의 OpenMP 기술 적용 및 성능 측정", 한국컴퓨터정보학회, 제 21권 제 1호.
- [4] 데이비드 B. 커크, 대규모 병렬 프로세서 프로그래밍 CUDA를 이용한 실용적 접근, BJ퍼블릭, 2010.
- [5] David M. Sheen, Sami M. A., Mohamed D. A., 'Application of the three-dimensional Finite-Difference Time-Domain method to the analysis of planar microstrip circuits', IEEE Transaction on Microwave Theory and Techniques, vol. 38, pp 849-857, 1990.