(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년12월27일
(11) 등록번호 10-2482316
(24) 등록일자 2022년12월23일

(51) 국제특허분류(Int. Cl.)
G06T 1/20 (2018.01) *G05B 19/418* (2006.01)
(52) CPC특허분류
G06T 1/20 (2013.01)
G05B 19/41865 (2013.01)
(21) 출원번호 10-2021-7028894(분할)
(22) 출원일자(국제) 2019년06월21일
심사청구일자 2022년02월09일
(85) 번역문제출일자 2021년09월08일
(65) 공개번호 10-2021-0112421
(43) 공개일자 2021년09월14일
(62) 원출원 특허 10-2021-7001978
원출원일자(국제) 2019년06월21일
심사청구일자 2021년01월20일
(86) 국제출원번호 PCT/US2019/038586
(87) 국제공개번호 WO 2019/246588
국제공개일자 2019년12월26일
(30) 우선권주장
16/015,302 2018년06월22일 미국(US)
(56) 선행기술조사문헌
KR1020080080954 A*
(뒷면에 계속)
전체 청구항 수 : 총 20 항

(73) 특허권자
어플라이드 머티어리얼스, 인코포레이티드
미국 95054 캘리포니아 산타 클라라 바우어스 애브뉴 3050
(72) 발명자
에마니, 샴 순더
미국 95014 캘리포니아 쿠퍼티노 콩그레스 플레이스 10084
(74) 대리인
특허법인 남앤남

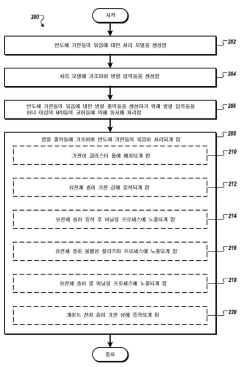
심사관 : 이후락

(54) 발명의 명칭 **기판 라우팅 및 스루풋 모델링을 위한 그래픽 처리 유닛의 사용**

(57) 요 약

통합 기판 처리 시스템에서 기판 처리 시퀀스를 스케줄링하기 위한 방법들, 시스템들 및 비-일시적 컴퓨터 판독 가능 매체가 본 명세서에 개시된다. 처리 디바이스는 반도체 기판들의 묶음에 대한 처리 모델을 생성한다. 처리 모델은 통합 기판 처리 시스템의 각각의 처리 챔버 내의 각각의 반도체 기판에 대해 대응하는 시작 시간을 정의한다. 하나 이상의 그래픽 처리 유닛(GPU)들은 처리 모델에 기초하여 병렬 입력들을 생성하고, 반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 병렬 입력들을 하나 이상의 GPU들의 복수의 코어들에 의해 동시에 처리한다. 병렬 입력들 각각은 대응하는 병렬 출력을 생성하기 위해 하나 이상의 GPU들의 복수의 코어들의 개별 코어 상에서 처리된다. 처리 디바이스는 병렬 출력들에 기초하여 통합 기판 처리 시스템에서 반도체 기판들의 묶음이 처리되게 한다.

대 표 도 - 도2



(52) CPC특허분류
G05B 19/41885 (2013.01)
G05B 2219/34417 (2013.01)
G05B 2219/35499 (2013.01)
G05B 2219/45031 (2013.01)
G06T 2200/28 (2013.01)

(56) 선행기술조사문헌
KR1020110036172 A*
KR1020070085719 A*
JP2017537312 A
JP2013225184 A
US20080216077 A1
US20070179652 A1
KR1020150100532 A
KR1020110124363 A
KR1020110110341 A
KR1020100135784 A
KR1020030007454 A
*는 심사관에 의하여 인용된 문헌

명 세 서

청구범위

청구항 1

방법으로서,

기판 처리 시스템에서의 기판 프로세스들과 연관된 복수의 시작 시간들을 포함하는 행렬을 수신하는 단계;

하나 이상의 GPU(graphics processing unit)들 중 제1 GPU에 의해 상기 행렬에 기초하여 복수의 행렬들을 생성하는 단계 - 상기 복수의 행렬들은 상기 행렬의 수퍼세트들임 -; 및

병렬 출력들을 생성하기 위해 상기 복수의 행렬들을 상기 하나 이상의 GPU들의 복수의 코어들에 의해 동시에 처리하는 단계 - 상기 복수의 행렬들의 각각의 행렬은 상기 복수의 코어들 중 개별 코어 상에서 처리되고, 스케줄은 상기 병렬 출력들에 기초하여 생성되고, 기판들은 상기 스케줄에 기초하여 상기 기판 처리 시스템에 의해 처리됨 -

를 포함하는,

방법.

청구항 2

제1 항에 있어서,

상기 스케줄에 기초하여 상기 기판 처리 시스템의 스루풋을 예측하는 단계를 더 포함하는,

방법.

청구항 3

제1 항에 있어서,

상기 스케줄에 기초하여 상기 기판 처리 시스템에서 상기 기판들이 처리되게 하는 단계를 더 포함하는,

방법.

청구항 4

제1 항에 있어서,

상기 기판 프로세스들과 연관된 복수의 시작 시간들은:

상기 기판 처리 시스템 내에서의 기판의 이송과 연관된 하나 이상의 시작 시간들을 포함하는,

방법.

청구항 5

제1 항에 있어서,

상기 기판 처리 시스템에서의 기판 프로세스들과 연관된 복수의 시작 시간들은:

상기 기판 처리 시스템의 하나 이상의 처리 챔버들에서의 기판을 처리하는 것과 연관된 하나 이상의 시작 시간들을 포함하는,

방법.

청구항 6

제1 항에 있어서,

상기 병렬 출력들을 생성하기 위해 상기 복수의 행렬들을 동시에 처리하는 단계는:

상기 복수의 행렬들 각각에 대해, 값들의 복수의 세트들 중 대응하는 세트의 값들을 생성하도록 상기 복수의 행렬들 각각을 해결하는 단계 - 상기 값들의 복수의 세트들의 각각의 세트는 상기 기판 프로세스들 중 하나 이상의 프로세스의 개개의 지속기간에 대응함 -; 및

상기 값들의 복수의 세트들 중, 상기 기판 프로세스들 중 하나 이상의 프로세스의 최소 지속기간에 대응하는 세트의 값들을 선택하는 단계 - 상기 병렬 출력들은 상기 세트의 값들에 대응함 -

를 포함하는,

방법.

청구항 7

제1 항에 있어서,

상기 기판 처리 시스템에서의 결함에 기초하여 업데이트된 행렬을 수신하는 단계;

상기 업데이트된 행렬에 기초하여 업데이트된 복수의 행렬들을 생성하는 단계; 및

업데이트된 병렬 출력들을 생성하기 위해 상기 업데이트된 복수의 행렬들을 상기 하나 이상의 GPU들에 의해 동시에 처리하는 단계 - 상기 기판 처리 시스템에서의 기판들을 처리하기 위한 업데이트된 스케줄은 상기 업데이트된 병렬 출력들에 기초하여 생성됨 -

를 더 포함하는,

방법.

청구항 8

제7 항에 있어서,

상기 스케줄에 기초하여 상기 기판 처리 시스템에서 상기 기판들을 라우팅하는 단계; 및

상기 업데이트된 스케줄에 기초하여 상기 기판 처리 시스템에서 하나 이상의 기판들을 다시 라우팅하는 단계

를 더 포함하는,

방법.

청구항 9

제1 항에 있어서,

상기 제1 GPU에 의해, 상기 복수의 행렬들의 제1 수량이 상기 제1 GPU의 제1 복수의 코어들의 제2 수량을 초과함을 결정하는 단계; 및

상기 복수의 행렬들의 제1 서브세트를 상기 제1 GPU에 할당하고 상기 복수의 행렬들의 제2 서브세트를 상기 하나 이상의 GPU들의 제2 GPU에 할당하는 단계를 더 포함하며,

상기 제1 GPU는 상기 제1 서브세트를 처리하고 상기 제2 GPU는 상기 제2 서브세트를 처리하는,

방법.

청구항 10

시스템으로서,

메모리; 및

상기 메모리에 결합된 그래픽 처리 유닛(GPU)을 포함하며, 상기 GPU는:

기판 처리 시스템에서의 기판 프로세스들과 연관된 복수의 시작 시간들을 포함하는 행렬을 수신하고;

상기 행렬에 기초하여 복수의 행렬들을 생성하고 - 상기 복수의 행렬들은 상기 행렬의 수퍼세트들임 -; 그리

고

병렬 출력들을 생성하기 위해 상기 복수의 행렬들을 상기 GPU의 복수의 코어들에 의해 동시에 처리되게 하는 - 상기 복수의 행렬들의 각각의 행렬은 상기 복수의 코어들 중 개별 코어 상에서 처리되고, 스케줄은 상기 병렬 출력들에 기초하여 생성되고, 기판들은 상기 스케줄에 기초하여 상기 기판 처리 시스템에 의해 처리됨 -,

시스템.

청구항 11

제10 항에 있어서,

상기 메모리 및 상기 GPU에 결합된 프로세서를 더 포함하고, 상기 프로세서는:

상기 스케줄에 기초하여 상기 기판 처리 시스템의 스루풋을 예측하는 것; 또는

상기 스케줄에 기초하여 상기 기판 처리 시스템에서 상기 기판들이 처리되게 하는 것

중 하나 이상을 하는,

시스템.

청구항 12

제10 항에 있어서,

상기 기판 프로세스들과 연관된 복수의 시작 시간들은:

상기 기판 처리 시스템 내에서 기판을 이송하는 것과 연관된 제1 시작 시간들; 또는

상기 기판 처리 시스템의 하나 이상의 처리 챔버들에서 기판을 처리하는 것과 연관된 제2 시작 시간들

중 하나 이상을 포함하는,

시스템.

청구항 13

제10 항에 있어서,

상기 병렬 출력들을 생성하기 위해 상기 복수의 행렬들을 동시에 처리하기 위해, 상기 GPU는:

상기 복수의 행렬들 각각에 대해, 값들의 복수의 세트들 중 대응하는 세트의 값들을 생성하도록 상기 복수의 행렬들 각각을 해결하고 - 상기 값들의 복수의 세트들의 각각의 세트는 상기 기판 프로세스들 중 하나 이상의 프로세스의 개개의 지속기간에 대응함 -; 그리고

상기 값들의 복수의 세트들 중, 상기 기판 프로세스들 중 하나 이상의 프로세스의 최소 지속기간에 대응하는 세트의 값들을 선택하는 - 상기 병렬 출력들은 상기 세트의 값들에 대응함 -,

시스템.

청구항 14

제10 항에 있어서,

상기 GPU는 추가로:

상기 기판 처리 시스템에서의 결함에 기초하여 업데이트된 행렬을 수신하고;

상기 업데이트된 행렬에 기초하여 업데이트된 복수의 행렬들을 생성하고; 그리고

업데이트된 병렬 출력들을 생성하기 위해 상기 업데이트된 복수의 행렬들을 동시에 처리하는 - 상기 기판 처리 시스템에서의 기판들을 처리하기 위한 업데이트된 스케줄은 상기 업데이트된 병렬 출력들에 기초하여 생성됨 -,

시스템.

청구항 15

제14 항에 있어서,

상기 메모리 및 상기 GPU에 결합된 프로세서를 더 포함하고, 상기 프로세서는:

상기 스케줄에 기초하여 상기 기판 처리 시스템에서 상기 기판들을 라우팅하고; 그리고

상기 업데이트된 스케줄에 기초하여 상기 기판 처리 시스템에서 하나 이상의 기판들을 다시 라우팅하는,

시스템.

청구항 16

명령들이 저장된 비-일시적 컴퓨터 판독 가능 매체로서,

상기 명령들은 그래픽 처리 유닛(GPU)에 의해 실행될 때 상기 GPU로 하여금,

기판 처리 시스템에서의 기판 프로세스들과 연관된 복수의 시작 시간들을 포함하는 행렬을 수신하는 동작;

상기 행렬에 기초하여 복수의 행렬들을 생성하는 동작 - 상기 복수의 행렬들은 상기 행렬의 수퍼세트들임 -; 및

병렬 출력들을 생성하기 위해 상기 복수의 행렬들을 상기 GPU의 복수의 코어들에 의해 동시에 처리하는 동작 - 상기 복수의 행렬들의 각각의 행렬은 상기 복수의 코어들 중 개별 코어 상에서 처리되고, 스케줄은 상기 병렬 출력들에 기초하여 생성되고, 기판들은 상기 스케줄에 기초하여 상기 기판 처리 시스템에 의해 처리됨 -

을 포함하는 동작들을 수행하게 하는,

비-일시적 컴퓨터 판독 가능 매체.

청구항 17

제16 항에 있어서,

상기 동작들은:

상기 스케줄에 기초하여 상기 기판 처리 시스템의 스루풋을 예측하는 동작; 또는

상기 스케줄에 기초하여 상기 기판 처리 시스템에서 상기 기판들이 처리되게 하는 동작

중 하나 이상을 더 포함하는,

비-일시적 컴퓨터 판독 가능 매체.

청구항 18

제16 항에 있어서,

상기 기판 프로세스들과 연관된 복수의 시작 시간들은:

상기 기판 처리 시스템 내에서 기판을 이송하는 것과 연관된 제1 시작 시간들; 또는

상기 기판 처리 시스템의 하나 이상의 처리 챔버들에서 기판을 처리하는 것과 연관된 제2 시작 시간들

중 하나 이상을 포함하는,

비-일시적 컴퓨터 판독 가능 매체.

청구항 19

제16 항에 있어서,

상기 병렬 출력들을 생성하기 위해 상기 복수의 행렬들을 동시에 처리하는 동작은:

상기 복수의 행렬들 각각에 대해, 값들의 복수의 세트들 중 대응하는 세트의 값들을 생성하도록 상기 복수의 행렬들 각각을 해결하는 동작 - 상기 값들의 복수의 세트들의 각각의 세트는 상기 기판 프로세스들 중 하나 이상

의 프로세스의 개개의 지속기간에 대응함 -; 및

상기 값들의 복수의 세트들 중, 상기 기판 프로세스들 중 하나 이상의 프로세스의 최소 지속기간에 대응하는 세트의 값들을 선택하는 동작 - 상기 병렬 출력들은 상기 세트의 값들에 대응함 -

을 포함하는,

비-일시적 컴퓨터 판독 가능 매체.

청구항 20

제16 항에 있어서,

상기 동작들은:

상기 기판 처리 시스템에서의 결함에 기초하여 업데이트된 행렬을 수신하는 동작;

상기 업데이트된 행렬에 기초하여 업데이트된 복수의 행렬들을 생성하는 동작; 및

업데이트된 병렬 출력들을 생성하기 위해 상기 업데이트된 복수의 행렬들을 동시에 처리하는 동작 - 상기 기판 처리 시스템에서의 기판들을 처리하기 위한 업데이트된 스케줄은 상기 업데이트된 병렬 출력들에 기초하여 생성됨 -

을 더 포함하는,

비-일시적 컴퓨터 판독 가능 매체.

발명의 설명

기 술 분 야

[0001] [001] 본 개시내용은 통합 처리 시스템에서의 기판들의 이송에 관한 것으로, 보다 구체적으로는 그래픽 처리 유닛(GPU: graphics processing unit)을 사용하여 통합 처리 시스템에서의 기판 라우팅 및 스루풋 모델링을 개선하는 것에 관한 것이다.

배 경 기 술

[0002] [002] 반도체 처리에서는, 다수의 처리 동작들을 갖는 특정 처리 레시피(recipe)들을 사용하여 반도체 기판들 상에 다층 피처(feature)들이 제작된다. 처리 환경(예컨대, 제어된 환경)에서 기판들을 제거하지 않고도 프로세스 시퀀스를 수행하도록 다수의 프로세스 챔버들을 통합하는 클러스터 툴이 일반적으로 반도체 기판들의 처리에 사용된다. 프로세스 시퀀스는 일반적으로 클러스터 툴 내의 하나 이상의 처리 챔버들에서 완료되는 디바이스 제작 동작들 또는 프로세스 레시피 동작들의 시퀀스로서 정의된다. 프로세스 시퀀스는 일반적으로 (예컨대, 전자 디바이스 제작을 위한) 다양한 기판 처리 동작들을 포함할 수 있다.

[0003] [003] 클러스터 툴들은 기판들을 서로 다른 포지션들로 이동시키고, 사용자 입력에 기초하여 기판들에 대해 프로세스들을 실행하는 것을 담당하는 시퀀서를 포함할 수 있다. 시퀀서는 더 큰 스루풋이 달성될 수 있도록 기판 이동들을 개선하도록 구성된다. 클러스터 툴에서 기판들을 이송하는 동안, 시퀀서는 또한 프로세스 엔지니어들 또는 사용자들에 의해 지정된 모든 제약들이 충족됨을 확인한다. 종래의 접근 방식들은 휴리스틱 방식인데, 즉 각각의 제품은, 토폴러지들을 다루는 커스텀 소프트웨어 코드 및 클러스터 툴이 자신을 찾을 수 있는 가장 일반적인 통계로 작성된다. 신제품들에 대해 이 코드를 작성하는 것은 시간 소모적이며 안정화하는 데에도 또한 시간이 오래 걸린다.

발명의 내용

[0004] [004] 다음은 본 개시내용의 일부 양상들의 기본적인 이해를 제공하기 위한 본 개시내용의 간단한 요약이다. 이 요약은 본 개시내용의 광범위한 개요는 아니다. 이는 본 개시내용의 핵심 또는 중요한 엘리먼트들을 식별하지도, 본 개시내용의 특정 구현들의 임의의 범위 또는 청구항들의 임의의 범위를 기술하지도 않는 것으로 의도된다. 그 유일한 목적은 본 개시내용의 일부 개념들을 뒤에 제시되는 보다 상세한 설명에 대한 서론으로서 간단한 형태로 제시하는 것이다.

[0005] [005] 본 개시내용의 한 양상에서, 방법은 반도체 기판들의 묶음(batch)에 대한 처리 모델을 생성하는 단계를

포함할 수 있다. 처리 모델은 통합 기판 처리 시스템의 각각의 처리 챔버 내의 각각의 반도체 기판에 대해 대응하는 시작 시간을 정의할 수 있다. 이 방법은 처리 모델에 기초하여 병렬 입력들을 생성하는 단계를 더 포함할 수 있다. 이 방법은 반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 병렬 입력들을 하나 이상의 GPU들의 복수의 코어들에 의해 동시에 처리하는 단계를 더 포함할 수 있다. 병렬 입력들 각각은 대응하는 병렬 출력을 생성하기 위해 하나 이상의 GPU들의 복수의 코어들의 개별 코어 상에서 처리된다. 이 방법은 병렬 출력들에 기초하여 통합 기판 처리 시스템에서 반도체 기판들의 묶음이 처리되게 하는 단계를 더 포함할 수 있다.

도면의 간단한 설명

[0006] [006] 본 개시내용은 첨부 도면들의 도(figure)들에서 제한으로서가 아니라 예로서 예시된다.

[007] 도 1은 특정 실시예들에 따른 컴퓨팅 환경을 예시한다.

[008] 도 2는 특정 실시예들에 따라, 병렬 출력들에 기초하여 반도체 기판들이 처리되게 하기 위한 방법의 흐름도이다.

[009] 도 3은 특정 실시예들에 따라, 타임테이블에 기초하여 반도체 기판들의 스루풋을 예측하기 위한 방법의 흐름도이다.

[0010] 도 4는 특정 실시예들에 따라, 하나 이상의 GPU들에 의해 병렬 입력들을 처리하기 위한 방법의 흐름도이다.

[0011] 도 5는 특정 실시예들에 따른 컴퓨팅 플랫폼을 예시한다.

발명을 실시하기 위한 구체적인 내용

[0007] [0012] 본 명세서에서는 (예컨대, 클러스터 툴 내의 기판들의) 기판 라우팅 및 스루풋 예측에 관한 기술들이 설명된다. 대개, 기판 라우팅을 위한 새로운 기판 처리 시퀀스들(예컨대, 각각의 기판 순서에 대한 상이한 기판 처리 시퀀스)이 수신된다. 기판 처리 시퀀스들은 업데이트될 수 있다. 기판 처리 시퀀스들은 고장들에 반응해야 한다(예컨대, 클러스터 툴의 일부에서의 고장에 대한 응답으로 기판을 다시 라우팅해야 한다). 처리 챔버들의 수가 증가함에 따라 기판 처리 시퀀스들의 복잡성이 증가할 수 있다. 기판 처리는 엄격한 타이밍 제어를 겪을 수 있으므로, 기판 처리 중에 발생할 수 있는 각각의 이벤트가 고려되어야 한다. 이벤트들은 로봇 스케줄들, 예방적 유지보수 작업들, 더미 웨이퍼 이동들, 세정, 복잡한 시퀀스들, 또는 동일한 기판에 의한 동일한 챔버로의 다수의 방문들 중 하나 이상을 포함할 수 있다. 기판 처리는 큐 시간(예컨대, 프로세스가 완료된 후 처리 챔버에서 기판들이 대기하는 시간의 양)에 대한 제한들을 가질 수 있다. 다음 처리 챔버가 이용 가능하지 않다면 또는 로봇이 다른 재료들을 이동시키고 있다면 큐 시간이 늘어날 수 있다.

[0008] [0013] 기판들의 기판 라우팅 및/또는 스루풋 예측을 위한 소프트웨어 코드가 작성될 수 있다. 기판 라우팅 및/또는 스루풋 예측을 위한 소프트웨어 코드를 작성하는 것은 시간이 오래 걸릴 수 있고, 안정화하는 데 오랜 시간이 걸릴 수 있으며, 발생할 수 있는 이벤트들 및 고장들을 고려하지 않을 수 있다(예컨대, 부정확할 수 있음).

[0009] [0014] 본 명세서에 개시되는 디바이스들, 시스템들 및 방법들은 기판 라우팅 및 스루풋 모델링을 위한 하나 이상의 GPU들의 사용을 제공한다. 기판 라우팅(예컨대, 복잡한 시퀀싱 요건들에 따른) 및 스루풋은 처리 모델을 사용하여 모델링될 수 있다. 처리 모델은 이벤트들, 고장들, 큐 시간 제한들 등 중 하나 이상을 고려할 수 있다. 처리 모델에 기초하여 병렬 입력들이 생성될 수 있으며, 하나 이상의 GPU들의 코어들이 병렬 입력들을 동시에 처리하여 병렬 출력들을 생성할 수 있다. 병렬 출력들은 처리 지속기간을 최소화하기 위한 값들(예컨대, 서로 다른 처리 챔버들에서의 서로 다른 프로세스들에 대한 시작 시간들 등)을 표시할 수 있다. 기판들이 처리될 수 있고, 병렬 출력들에 기초하여 기판 스루풋이 예측될 수 있다. 결함 검출에 대한 응답으로, 업데이트된 처리 모델이 결함을 기초로 생성될 수 있고, 업데이트된 병렬 입력들이 생성될 수 있으며, 업데이트된 병렬 입력들이 하나 이상의 GPU들의 코어들에 의해 처리되어, 기판들이 다시 라우팅되게 하기 위한 또는 스루풋을 다시 예측하기 위한 업데이트된 병렬 출력들을 생성할 수 있다.

[0010] [0015] 기판 라우팅 및 스루풋 모델링을 위해 하나 이상의 GPU들을 사용하는 것은 기술적 이점들을 제공한다. 기술적 이점들은, 처리 모델을 생성하고 처리 모델을 단시간 내에(예컨대, 실시간으로, 몇 초 내에 등) 해결함으로써 새로운 처리 시퀀스들(예컨대, 새로운 기판 순서, 처리 시퀀스에 대한 업데이트들 등)에 대한 솔루션들을 찾는 것을 포함한다. 기술적 이점들은 또한, 결함을 검출하고, 결함에 기초하여 처리 모델을 업데이트하고,

업데이트된 처리 모델을 단시간 내에 해결함으로써 고장들에 반응하는 것을 포함한다. 기술적 이점들은 또한, 기판 처리를 위한 타임테이블들을 생성하고, 고장들에 응답하여 타임테이블들을 업데이트하는 것을 포함하는데, 여기서 타임테이블들은 큐 시간 제한들을 충족한다.

[0011] [0016] 본 개시내용의 양상들은 에너지 소비, 대역폭, 처리 지속기간 등의 상당한 감소의 기술적 이점들을 야기한다. 본 개시내용의 양상들은 기판 처리 동안 대기 시간을 감소시키며, 이는 전체 에너지 소비를 감소시킨다. 처리 모델을 생성하고, 클라이언트 디바이스의 하나 이상의 GPU들을 사용하여 처리 모델을 해결하고, 해결된 처리 모델에 기초하여 타임테이블을 생성하고, 타임테이블을 송신하는 클라이언트 디바이스는 처리 모델, 처리 모델에 기초하는 병렬 입력들, 또는 해결된 처리 모델 중 하나 이상을 네트워크를 통해 다른 컴포넌트들에 송신하는 클라이언트 디바이스보다 더 적은 대역폭을 사용한다.

[0012] [0017] 도 1은 특정 실시예들에 따른 컴퓨팅 환경(150)을 예시한다. 컴퓨팅 환경(150)은 제어기(160) 및 클라이언트 디바이스(170)를 포함한다. 클라이언트 디바이스(170)는 하나 이상의 GPU들(180)을 포함할 수 있다. 각각의 GPU(180)는 병렬 입력들을 동시에 처리하여 병렬 출력들을 생성할 수 있는 다수의 코어들(예컨대, 수백 개의 코어들)을 포함할 수 있다. 제어기(160)는 네트워크(190)를 통해 클라이언트 디바이스(170)와 통신한다. 컴퓨팅 환경(150)은 클러스터 툴(100)을 포함할 수 있다. 클러스터 툴(100)은 기판 처리를 위해 사용될 수 있다. 본 명세서에서 설명되는 방법들은 프로세스 시퀀스를 수행하도록 구성된 다른 툴들에 사용될 수 있다. 일례로, 도 1의 클러스터 툴(100)은 캘리포니아 산타 클라라 소재의 Applied Materials, Inc.로부터 상업적으로 입수할 수 있는 Endura® 클러스터 툴일 수 있다.

[0013] [0018] 클라이언트 디바이스(170)는 서로 다른 스테이지들에서 기판이 방문할 가능한 처리 챔버들 및 각각의 처리 챔버 내에서 실행될 대응하는 프로세스를 설명하는 시퀀스 레시피를 (예컨대, 사용자 입력을 통해) 수신할 수 있다. 클라이언트 디바이스(170)는 처리 모델을 생성하고, 처리 모델에 기초하여 병렬 입력들을 생성하고, 하나 이상의 GPU들(180)의 코어들에 의해 병렬 입력들을 동시에 처리하여 병렬 출력들을 생성할 수 있다. 클라이언트 디바이스(170)는 병렬 출력들에 기초하여 클러스터 툴(100) 내에서 처리될 기판들의 묶음에 대한 프로세스 스케줄(예컨대, 기판들이 더 짧은 지속기간 내에 처리될 수 있도록 기판 이동들에 대한 스케줄)을 생성할 수 있다. 예를 들어, 클라이언트 디바이스(170)는 클러스터 툴(100)의 수학적 모델을 생성한 다음, 클러스터 툴(100) 내에서 기판들을 이송하는 개선된 방법에 대한 솔루션을 제공할 뿐만 아니라 클러스터 툴(100)의 정의된 제약들을 충족하도록 모델을 최적화한다.

[0014] [0019] 일부 실시예들에서, 클라이언트 디바이스는 (예컨대, 클라이언트 툴(100)에 결합되지 않고) 타임테이블에 기초하여 클러스터 툴(100)의 스루풋을 예측할 수 있다. 일부 실시예들에서, 클라이언트 디바이스(170)는 기판 처리 스케줄(예컨대, 타임테이블) 및 명령들을 제어기(160)로 전송할 수 있다. 제어기(160)는 기판 처리 스케줄 및 명령들에 기초하여 클러스터 툴(100)에 의해 반도체 기판들의 묶음이 처리되게 할 수 있다. 시퀀스 레시피에 대한 업데이트들, 클러스터 툴(100)의 고장들 등에 대한 응답으로, 클라이언트 디바이스(170)는 업데이트된 처리 스케줄 및 명령들을 생성할 수 있다.

[0015] [0020] 클러스터 툴(100)은 진공 기밀 처리 플랫폼(101) 및 팩토리 인터페이스(102)를 포함한다. 플랫폼(101)은 복수의 처리 챔버들(110, 108, 114, 112, 118, 116) 및 적어도 하나의 로드락(load-lock) 챔버(120)를 포함하며, 이러한 챔버들은 진공 기판 이송 챔버들(103, 104)에 결합된다. 팩토리 인터페이스(102)는 로드락 챔버(120)에 의해 이송 챔버(104)에 결합된다.

[0016] [0021] 일 실시예에서, 팩토리 인터페이스(102)는 적어도 하나의 도킹 스테이션, 적어도 하나의 기판 이송 로봇(138) 및 적어도 하나의 기판 정렬기(140)를 포함한다. 도킹 스테이션은 하나 이상의 FOUP(front opening unified pod)들(128)을 받아들이도록 구성된다. 도 1의 실시예에서는 2개의 FOUP(128A, 128B)들이 도시된다. 기판 이송 로봇(138)은 기판을 팩토리 인터페이스(102)로부터 로드락 챔버(120)로 이송하도록 구성된다.

[0017] [0022] 로드락 챔버(120)는 팩토리 인터페이스(102)에 결합된 제1 포트 및 제1 이송 챔버(104)에 결합된 제2 포트를 갖는다. 로드락 챔버(120)는 압력 제어 시스템에 결합되는데, 압력 제어 시스템은 이송 챔버(104)의 진공 환경과 팩토리 인터페이스(102)의 실질적으로 주위(예컨대, 대기) 환경 사이에서 기판을 통과시키는 것을 가능하게 하도록 필요에 따라 챔버(120)를 펌프 다운(pump down) 및 통기시킨다.

[0018] [0023] 제1 이송 챔버(104) 및 제2 이송 챔버(103) 내에는 각각 제1 로봇(107) 및 제2 로봇(105)이 배치된다. 이송 챔버(104)에는 2개의 기판 이송 플랫폼들(106A, 106B)이 배치되어 로봇들(105, 107) 간의 기판 이송을 가능하게 한다. 플랫폼들(106A, 106B)은 이송 챔버들(103, 104)에 개방될 수 있거나, 이송 챔버들(103, 104) 각

각에서 서로 다른 작동 압력들이 유지되게 하도록 이송 챔버들(103, 104)로부터 선택적으로 격리(즉, 밀봉)될 수 있다.

[0019] [0024] 제1 이송 챔버(104)에 배치된 로봇(107)은 로드락 챔버(120)와 처리 챔버들(116, 118)과 기판 이송 플랫폼들(106A, 106B) 간에 기판들을 이송할 수 있다. 제2 이송 챔버(103)에 배치된 로봇(105)은 기판 이송 플랫폼들(106A, 106B)과 처리 챔버들(112, 114, 110, 108) 간에 기판들을 이송할 수 있다.

[0020] [0025] 클라이언트 디바이스(170)는 기판들의 리스트, 리스트의 각각의 기판에 대한 대응하는 처리 시퀀스, 및 리스트 내의 각각의 기판에 대한 대응하는 시퀀스에서 각각의 프로세스에 대한 대응하는 처리 챔버(예컨대, 처리 챔버들(112, 114, 110, 108))에 기초하여 스케줄(예컨대, 타임테이블)을 생성할 수 있다.

[0021] [0026] 도 2 - 도 4는 특정 실시예들에 따라 하나 이상의 GPU들을 사용하여 (예컨대, 기판 라우팅 및/또는 스루풋 예측을 위해) 하나 이상의 GPU들에 의한 병렬 입력들을 처리하기 위한 방법들(200, 300, 400)의 흐름도들이다. 방법들(200, 300, 400)은 하드웨어(예컨대, 회로, 전용 로직, 프로그래밍 가능 로직, 마이크로코드 등), 소프트웨어(이를테면, 처리 디바이스, 하나 이상의 GPU들, 범용 컴퓨터 시스템 또는 전용 머신 상에서 실행되는 명령들), 펌웨어, 마이크로코드, 또는 이들의 조합을 포함할 수 있는 처리 로직에 의해 수행될 수 있다. 일 실시예에서, 방법들(200, 300, 400)은 부분적으로는 클라이언트 디바이스(170)에 의해 수행될 수 있다. 일부 실시예들에서, 비-일시적 저장 매체는, 클라이언트 디바이스(170)(예컨대, 클라이언트 디바이스(170)의 하나 이상의 GPU들 또는 처리 디바이스 중 적어도 하나)에 의해 실행될 때, 클라이언트 디바이스(170)(예컨대, 하나 이상의 GPU들 또는 처리 디바이스 중 적어도 하나)로 하여금 방법들(200, 300, 400)을 수행하게 하는 명령들을 저장한다.

[0022] [0027] 설명의 단순화를 위해, 방법들(200, 300, 400)은 일련의 동작들로서 도시되고 설명된다. 그러나 본 개시내용에 따른 동작들은 다양한 순서들로 그리고/또는 동시에 그리고 본 명세서에서 제시 및 설명되지 않는 다른 동작들과 함께 발생할 수 있다. 더욱이, 개시되는 청구 대상에 따라 방법들(200, 300, 400)을 구현하기 위해 예시된 모든 동작들이 수행되는 것은 아닐 수 있다. 추가로, 당해 기술분야에서 통상의 지식을 가진 자들은, 방법들(200, 300, 400)이 대안으로, 상태도 또는 이벤트들을 통해 일련의 상호 관련 상태들로서 표현될 수 있다고 이해 및 인식할 것이다.

[0023] [0028] 도 2는 특정 실시예들에 따라, 병렬 출력들에 기초하여 반도체 기판들이 처리되게 하기 위한 방법(200)의 흐름도이다.

[0024] [0029] 도 2를 참조하면, 블록(202)에서 클라이언트 디바이스(170)가 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 반도체 기판들의 묶음에 대한 처리 모델을 생성한다.

[0025] [0030] 처리 모델을 생성하기 위해, 클라이언트 디바이스(170)는 기판들의 묶음에서 각각의 기판에 대한 시퀀스를 정의할 수 있다. 일 실시예에서, 클라이언트 디바이스(170)는 사용자로부터 각각의 기판에 대한 시퀀스를 수신한다. 예를 들어, 사용자는 처리 시퀀스를: 착수, 증착, 어닐링, 에칭, 어닐링, 증착, 종료로서 정의할 수 있다. 수학적으로, 기판들의 첫 번째 묶음은 $\{W_i\}$로서 정의될 수 있으며, 여기서 i는 1 내지 n의 범위이다. 일부 실시예들에서, 각각의 기판(W_i)은 동일한 시퀀스의 동작들을 거칠 수 있다. 그 시퀀스의 동작들은 수학적으로 $\{s_i\}$로서 표현될 수 있으며, 여기서 i는 1 내지 n의 범위이다. 따라서 각각의 기판(W_i)은 클라이언트 디바이스(170)에 의해 정의된 시퀀스로 각각의 동작(s_i)을 거칠 수 있다.

[0026] [0031] 처리 모델을 생성하기 위해, 클라이언트 디바이스(170)는 처리 시퀀스의 각각의 동작에 대해 각각의 기판에 처리 챔버를 할당할 수 있다. 예를 들어, 도 1을 참조하면, 기판들의 묶음 내의 각각의 기판에 대한 프로세스 시퀀스를 가능하게 하도록 챔버들(108, 110, 112, 114, 116, 118)로부터 적합한 챔버들이 선택될 수 있다. 특정 예에서, 챔버들(116, 118)은 CVD(chemical vapor deposition) 챔버일 수 있고; 챔버들(108, 114)은 DPN(decoupled plasma nitridation) 챔버일 수 있으며; 챔버들(110, 112)은 RTP(rapid thermal process) 챔버일 수 있다. 하나 이상의 냉각 챔버들이 기판 이송 플랫폼(106A, 106B) 위에 포지셔닝될 수 있다. 이에 따라, 클러스터 툴(100)에서 배열을 결정할 때, 클라이언트 디바이스(170)는 처리 시퀀스의 각각의 프로세스 동작 및 동작들 간의 전환들에 대해 챔버들, 로드락들 및 로봇들을 할당할 수 있다.

[0027] [0032] 클라이언트 디바이스(170)는 처리 챔버들의 할당에 기초하여 처리 모델을 생성할 수 있다. 일반적으로, 시간(T_x)에 각각의 기판(W_x)이 시작한다. 각각의 시퀀스 동작(s_i)의 프로세스 지속기간은 D_s로서

정의되며, 여기서 s는 시퀀스의 동작 번호이다. 예를 들어, D_3은 시퀀스 동작(s_3)의 프로세스 시간이다. 일반적으로, 프로세스 챔버가 완료한 프로세스 이후에 프로세스 챔버에서 기판이 대기할 수 있다. 대기 시간은 Q_{xs}로서 정의되며, 여기서 x는 기판 번호이고 s는 시퀀스 동작 번호이다. 예를 들어, Q_{21}은 시퀀스 동작(s_1)에서의 기판(W_2)의 대기 시간으로서 해석된다. 앞선 정의들을 고려해 볼 때, $T_x + D_1 + Q_{11}$과 같은 시간에 기판(W_1)이 동작(s_1)을 시작한다. 일반화하면, 다음과 같은 시간에 기판(W_1)이 임의의 동작(s_i)을 시작할 것이다:

[0028]
$$T_x + \sum_{j=1}^{s-1} D_j + \sum_{j=1}^{s-1} Q_{1j}$$

[0029] [0033] 블록(202)은 클러스터 툴(100) 내의 각각의 처리 챔버에 대해, 클라이언트 디바이스(170)가 시퀀스 제약을 정의하는 것을 더 포함할 수 있다. 시퀀스 제약들은 기판들의 묶음에서 모든 기판들을 처리하는 데 걸리는 시간을 줄이거나 궁극적으로는 최소화하는 목표에 도움이 된다. 이것은 제어기(160)가 기판들을 가능한 한 빨리 클러스터 툴(100)로 보내고 클러스터 툴(100)로부터 기판들을 리트리브할 것임을 의미할 것이다. 그렇게 하기 위해, 클라이언트 디바이스(170)는 프로세스 모델을 생성하기 위해 선형 최적화의 원리를 활용한다.

[0030] [0034] 선형 최적화는 선형 관계들로 표현되는 요건들을 갖는 수학적 모델(예컨대, 행렬)에서 "최상"의 결과(예컨대, 가장 짧은 프로세스 시간)를 달성하는 방법이다. 수학적으로, 이는 다음과 같이 표현될 수 있다:

[0031] [0035] 다음을 최소화한다:

[0032]
$$\sum_{i=1}^{n} C_1 X_1$$

[0033] [0036] 다음을 조건으로 하며:

[0034]
$$A_{11}X_1 + A_{12}X_2 + A_{13}X_3 + \ldots \leq B_1$$

[0035]
$$A_{21}X_1 + A_{22}X_2 + A_{23}X_3 + \ldots \leq B_2$$

[0036]
$$\ldots$$

[0037]
$$A_{m1}X_1 + A_{m2}X_2 + A_{m3}X_3 + \ldots \leq B_m$$

[0038] [0037] 여기서 X_i는 변수들($\{A_{mn}\} \in \mathbb{Q}$, $\{B_i\} \in \mathbb{Q}$, $\{C_i\} \in \mathbb{Q}$)이다.

[0039] [0038] 이 원리를 위에 적용하면, 클라이언트 디바이스(170)는 다음을 최소화하며:

[0040]
$$\sum_{i=i}^{n} A_i T_i + \sum_{j=1}^{N} \sum_{k=1}^{m} B_j Q_{jk}$$

[0041] [0039] 여기서 A_i, B_j는 시작 시간 변수들(T_i) 및 대기 시간들(Q_{jk})에 각각 적용될 수 있는 가중치들이다. 예를 들어, 가중치들은 반도체 제조 프로세스의 추가 특징들에 관련될 수 있다. 일 실시예에서, 기판이 처리 챔버에서 처리가 완료된 후에 실행될 세정 프로세스에 대한 응답으로 가중치들이 조정될 수 있다. 다른 실시예에서, 가중치들은 클러스터 툴(100) 전반에 걸친 "더미" 기판 이동에 대한 응답으로 조정될 수 있다. 다른 실시예에서, 가중치들은 로봇이 단일 블레이드 로봇인지 아니면 이중 블레이드 로봇인지에 대한 응답으로 조정될 수 있다. 다른 실시예에서, 가중치들은 처리 챔버가 일괄 처리 챔버인 것에 대한 응답으로 조정될 수 있는데, 즉 처리 챔버는 2개 이상의 기판들을 동시에 처리할 수 있다. 또 다른 실시예에서, 가중치들은 기판이 특정 처리 챔버를 재방문할 것을 요구하는 기판 처리 시퀀스에 대한 응답으로 조정될 수 있다.

[0042] [0040] 일반적으로는, 이전 기판이 처리를 완료될 때까지, 주어진 기판이 주어진 처리 챔버에 들어갈 수 없을

때 제약들이 정의될 수 있다. 수학적으로, 시퀀스 동작(s_s)에서 동일한 처리 챔버를 사용하는 2개의 기판들(W_x, W_y)이 있다고 가정한다. W_y 전에 W_x가 챔버에 도착한다. 이에 따라, W_y에 대한 시작 시간은 W_x의 시작 시간 + 동작(s_s)의 지속기간 + 동작(s_s) 이후의 W_x 대기 시간을 초과한다. 시작 시간의 정의를 사용하면, 제약은 다음과 같이 표현될 수 있다:

[0043]
$$T_x + \sum_{j=1}^{s-1} D_j + \sum_{j=1}^{s-1} Q_{xj} + D_s + Q_{xs} \leq T_y + \sum_{j=1}^{s-1} D_j + \sum_{j=1}^{s-1} Q_{yj}$$

[0044] [0041] 제약은 기판 라우팅을 최적화하도록(예컨대, 기판 라우팅에 관련된 하나 이상의 문제들을 해결하도록) 해결될 수 있다. 예를 들어, 시퀀스 동작에 사용되는 각각의 처리 챔버 및 모든 각각의 연속 기판 쌍에 대해, 즉 각각의 (x, y)에 대해 $T_x + \sum_{j=1}^{s-1} D_j + \sum_{j=1}^{s-1} Q_{xj} + D_s + Q_{xs} \leq T_y + \sum_{j=1}^{s-1} D_j + \sum_{j=1}^{s-1} Q_{yj}$ 를 조건으로 $\sum_{i=1}^{n} T_i + \sum_{j=1}^{N} \sum_{k=1}^{m} Q_{jk}$ 가 최소화될 수 있으며, 여기서 W_x, W_y는 시퀀스 동작(s_s)에서 연속적으로 동일한 처리 챔버를 사용한다. 다른 예에서, 제약은 로봇의 움직임들을 최소화하도록 해결될 수 있다. 다른 예에서, 제약은 챔버의 유휴 시간을 최소화하도록 해결될 수 있다. 다른 예에서, 제약은 챔버 결함 및 시스템이 모든 기판들을 계속 처리할 수 없는 것에 대한 응답으로, FOUP들로 다시 보내질 수 있는 기판들의 최소 수를 결정하여, 기판들의 나머지는 생산을 중단하지 않고 처리될 수 있도록 해결될 수 있다. 일부 실시예들에서, 제약은 기판들의 스루풋을 예측하도록 해결될 수 있다.

[0045] [0042] 클라이언트 디바이스(170)는 동시에 모든 시퀀스 제약들에 기초하여 묶음 내의 모든 기판들에 대한 처리 모델을 생성할 수 있다. 일부 실시예들에서, 클라이언트 디바이스(170)는 각각의 기판에 동일한 처리 시퀀스가 할당되는 것에 대한 응답으로 즉시 처리 모델을 생성할 수 있다.

[0046] [0043] 블록(204)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 하나 이상의 GPU들(180) 중 제1 GPU(180A)에 의해) 처리 모델에 기초하여 병렬 입력들을 생성한다. 예를 들어, 처리 모델은 행렬일 수 있고, 병렬 입력들의 생성은 행렬에 기초하여 행렬들을 생성하는 것을 포함할 수 있다.

[0047] [0044] 행렬에서의 하나 이상의 값들은 변수에 대응할 수 있다. 예를 들어, 하나 이상의 값들은 기판들의 수의 변수에 대응할 수 있다. 일부 변수들은 대응하는 값들에 대한 요건들을 가질 수 있다. 예를 들어, 기판들의 수의 변수는 대응하는 값들이 정수여야 한다는 요건을 가질 수 있다. 클라이언트 디바이스(170)는 대응하는 변수의 요건들을 충족하지 않는 모든 값들에 대한 추가 행렬들을 생성할 수 있다. 예를 들어, 클라이언트 디바이스(170)는 행렬에서 정수여야 하는 변수(예컨대, 기판들의 수)에 대응하는, 정수가 아닌 값을 식별할 수 있다. 클라이언트 디바이스(170)는 정수가 아닌 값(예컨대, 3.5)을 정수가 아닌 값보다 더 큰 제1 정수(예컨대, 4)로 대체함으로써 복수의 행렬들 중 제1 행렬을 생성할 수 있다. 클라이언트 디바이스(170)는 정수가 아닌 값(예컨대, 3.5)을 정수가 아닌 값보다 더 작은 제2 정수(예컨대, 3)로 대체함으로써 복수의 행렬들 중 제2 행렬을 생성할 수 있다. 클라이언트 디바이스(170)는 클라이언트 디바이스(170)의 GPU(180)를 통해 병렬 입력들을 생성할 수 있다. 제1 행렬 및 제2 행렬은 행렬의 수퍼세트들일 수 있으며, 여기서 수퍼세트들은 정수가 아닌 값을 정수(예컨대, 제1 행렬에서는 4, 제2 행렬에서는 3)로 대체하는 추가 제약을 포함한다.

[0048] [0045] 블록(206)에서, 클라이언트 디바이스(170)는 반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 병렬 입력들을 하나 이상의 GPU들(180)의 코어들에 의해 동시에 처리한다. 병렬 입력들 각각(예컨대, 복수의 행렬들 중 대응하는 행렬)은 대응하는 병렬 출력을 생성하기 위해 하나 이상의 GPU들(180)의 복수의 코어들의 개별 코어 상에서 처리될 수 있다. 코어들 각각은 행렬들 각각에 대해, 반도체 기판들의 묶음의 처리를 위한 대응하는 세트의 값들을 생성하도록 대응하는 행렬을 해결할 수 있다. 클라이언트 디바이스(170)는 반도체 기판들의 묶음의 처리를 위한 최소 처리 지속기간에 대응하는 값들의 세트를 (예컨대, GPU(180)를 통해) 선택할 수 있다.

[0049] [0046] 일부 실시예들에서, 클라이언트 디바이스(170)는 병렬 출력들에 기초하여 (클러스터 툴(100)의 통합 기판 처리 시스템의) 스루풋을 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 예측할 수 있다. 스루풋을 예측하는(그리고 예측 스루풋이 디스플레이되게 하는) 것에 대한 응답으로, 클라이언트 디바이스(170)는 예측 스루풋에 기초하여 처리 모델에 대한 업데이트들을 (예컨대, 사용자 입력을 통해) 수신할 수 있고, 흐름은 블록(202)으로 돌아갈 수 있다. 처리 모델에 대한 추가 업데이트들을 수신하지 않는 것에 대한 응답으로, 흐름

은 블록(208)에서 계속될 수 있다.

[0050] [0047] 블록(208)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 병렬 출력들(예컨대, 최소 처리 지속기간에 대응하는 선택된 세트의 값들)에 기초하여 반도체 기판들이 처리되게 한다. 일부 실시예들에서, 클라이언트 디바이스(170)는 병렬 출력들에 기초하여 기판들에 대한 타임테이블을 생성한다(예컨대, 타임테이블은 각각의 기판의 시작 시간(T_x) 및 각각의 처리 챔버에서의 기판 처리 순서를 포함한다). 클라이언트 디바이스(170)는 선택적으로, 타임테이블을 제어기(160)에 송신할 수 있다. 클라이언트 디바이스(170)와 제어기(160)가 동일한 것인 그러한 실시예들에서, 클라이언트 디바이스(170)는 타임테이블을 송신할 필요가 없다.

[0051] [0048] 블록(208)에서, 클라이언트 디바이스(170)는 도 1의 클러스터 툴(100)과 같은 통합 클러스터 툴 내의 기판 상에서의 전기 층들의 증착을 위해 프로세스 시퀀스에 따라 반도체 기판들의 묶음이 처리되게 할 수 있다. 블록(208)은 블록들(210-220) 중 하나 이상을 포함할 수 있다. 블록(210)에서, 클라이언트 디바이스(170)는 기판이 클러스터 툴(100)에 배치되게 할 수 있다.

[0052] [0049] 블록(212)에서, 클라이언트 디바이스(170)는 유전체 층이 기판 상에 증착되게 할 수 있다. 유전체 층은 금속 산화물일 수 있고, ALD 프로세스, MOCVD 프로세스, 종래의 CVD 프로세스 또는 PVD 프로세스에 의해 증착될 수 있다.

[0053] [0050] 블록(214)에서, 클라이언트 디바이스는 (예컨대, 증착 프로세스 이후에) 기판이 증착 후 어닐링(PDA: post deposition anneal) 프로세스에 노출되게 할 수 있다. PDA 프로세스는 캘리포니아 산타 클라라 소재의 Applied Materials, Inc.로부터 상업적으로 입수할 수 있는 Radiance® RTP 챔버와 같은 급속 어닐링 챔버에서 수행될 수 있다.

[0054] [0051] 블록(216)에서, 클라이언트 디바이스(170)는 플라즈마 처리 층을 형성할 유전체 재료를 치밀화하기 위해 유전체 층이 불활성 플라즈마 프로세스에 노출되게 할 수 있다. 불활성 플라즈마 프로세스는 불활성 가스를 DPN(decoupled plasma nitridation) 챔버 내로 유동시킴으로써 수행되는 디커플드(decoupled) 불활성 가스 플라즈마 프로세스를 포함할 수 있다.

[0055] [0052] 블록(218)에서, 클라이언트 디바이스(170)는 기판 상에 배치된 플라즈마 처리 층이 열 어닐링 프로세스에 노출되게 할 수 있다.

[0056] [0053] 블록(220)에서, 클라이언트 디바이스(170)는 게이트 전극 층이 어닐링된 유전체 층 위에 증착되게 할 수 있다. 게이트 전극 층은 예를 들어, LPCVD 챔버를 사용하여 증착된 다결정질 Si, 비정질 Si, 또는 다른 적절한 재료일 수 있다.

[0057] [0054] 다시 도 1을 참조하면, 클러스터 툴(100)은 제어기(160)와 통신할 수 있다. 제어기(160)는 클러스터 툴(100) 내의 각각의 기판 처리 챔버(108, 110, 112, 114, 116, 118)의 프로세스 파라미터들의 제어를 돕는 제어기일 수 있다. 추가로, 제어기(160)는 클러스터 툴(100)에서 처리될 기판들의 시퀀싱 및 스케줄링을 도울 수 있다. 블록(208)에서, 클라이언트 디바이스(170)는 병렬 출력들에 기초하여 반도체 기판들이 처리되게 하도록 제어기(160)가 클러스터 툴(100)의 처리 파라미터들을 제어하게 할 수 있다.

[0058] [0055] 도 3은 특정 실시예들에 따라, 타임테이블에 기초하여 반도체 기판들의 스루풋을 예측하기 위한 방법(300)의 흐름도이다. 방법(300)은 반도체 기판들이 도 1의 클러스터 툴(100)에서 처리되게 할 수 있다. 다른 예들에서, 방법(300)은 반도체 기판들이 다른 클러스터 툴들 상에서 처리되게 할 수 있다. 일부 실시예들에서, 모든 기판들(W_i)이 동일한 시퀀스의 동작들을 거치는 것은 아니다.

[0059] [0056] 도 3을 참조하면, 블록(302)에서 클라이언트 디바이스(170)는 처리될 반도체 기판들의 리스트를 반도체 기판들의 묶음에 기초하여 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 생성한다. 예를 들어, 클러스터 툴(100)의 처리 챔버에 들어가도록 2개의 기판들(예컨대, W_x, W_y)이 선택될 수 있다.

[0060] [0057] 블록(304)에서, 클라이언트 디바이스(170)는 대응하는 처리 시퀀스를 반도체 기판들의 리스트에 대응하는 각각의 반도체 기판에 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 할당한다. 클러스터 툴(100)에 들어가도록 선택된 각각의 기판에 대한 시퀀스가 정의될 수 있다. 일 실시예에서, 클라이언트 디바이스(170)는 사용자로부터 각각의 기판에 대한 시퀀스를 수신한다. 예를 들어, 사용자는 처리 시퀀스를: 착수, 증착, 어닐링, 에칭, 어닐링, 증착, 종료로서 정의할 수 있다. 그 시퀀스의 동작들은 수학적으로 $\{s_i\}$로서 표

현될 수 있으며, 여기서 i는 1 내지 n의 범위이다. 따라서 $\{s_i\}$의 엘리먼트가 $\{s_j\}$의 엘리먼트와 동일하지 않게, W_x는 동작들의 세트 $\{s_i\}$를 포함하고, W_y는 동작들의 세트 $\{s_j\}$를 포함한다.

[0061] [0058] 블록(306)에서, 클라이언트 디바이스(170)는 반도체 기판들의 리스트 내의 각각의 반도체 기판에 대해, 대응하는 처리 챔버를 대응하는 처리 시퀀스의 각각의 프로세스에 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 할당한다. 예를 들어, 도 1을 참조하면, 앞서 블록(402)에서 정의된 프로세스 시퀀스를 가능하게 하도록 챔버들(108, 110, 112, 114, 116, 118)로부터 적합한 챔버들이 선택될 수 있다. 특정 예에서, 챔버들(116, 118)은 CVD(chemical vapor deposition) 챔버일 수 있고; 챔버들(108, 114)은 DPN(decoupled plasma nitridation) 챔버일 수 있으며; 챔버들(110, 112)은 RTP(rapid thermal process) 챔버일 수 있다. 하나 이상의 냉각 챔버들이 기판 이송 플랫폼(106A, 106B) 위에 포지셔닝될 수 있다. 이에 따라, W_x에 대해, 클라이언트 디바이스(170)는 세트 $\{s_i\}$ 내의 각각의 동작에 처리 챔버를 할당하고, W_y에 대해, 클라이언트 디바이스(170)는 세트 $\{s_j\}$ 내의 각각의 동작에 처리 챔버를 할당한다. 따라서 클러스터 툴(100)에서 배열을 결정할 때, 클라이언트 디바이스(170)는 처리 시퀀스의 각각의 프로세스 동작 및 W_x, W_y에 대한 동작들 간의 전환들에 대해 챔버들, 로드락들 및 로봇들을 할당할 수 있다.

[0062] [0059] 블록(308)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 반도체 기판들의 리스트(예컨대, 클러스터 툴(100)에 들어가도록 선택된 모든 기판들), 각각의 반도체 기판에 대한 대응하는 처리 시퀀스, 및 각각의 반도체 기판에 대한 각각의 프로세스에 대해 대응하는 처리 챔버(예컨대, 처리 챔버 할당)에 기초하여 처리 모델을 생성한다. 예를 들어, 클라이언트 디바이스(170)는 기판들(W_x, W_y)에 대한 처리 챔버 할당에 기초하여 모델을 생성한다. 일부 실시예들에서, 블록(308)은 클러스터 툴(100) 내의 각각의 처리 챔버에 대해, 클라이언트 디바이스(170)가 시퀀스 제약을 정의하는 것을 포함할 수 있다. 시퀀스 제약들은 기판들의 묶음에서 모든 기판들을 처리하는 데 걸리는 시간을 줄이거나 궁극적으로는 최소화하는 목표에 도움이 될 수 있다. 직관적으로, 이것은 제어기(160)가 기판들을 가능한 한 빨리 클러스터 툴(100)로 보내고 클러스터 툴(100)로부터 기판들을 리트리브할 것임을 의미할 것이다. 그렇게 하기 위해, 클라이언트 디바이스(170)는 프로세스 모델을 생성하기 위해 선형 최적화의 원리를 활용한다.

[0063] [0060] 예를 들어, 클라이언트 디바이스(170)는, 기판들(W_x, W_y)이 이들의 처리 시퀀스들 동안 이동할, 클러스터 툴(100) 내의 각각의 처리 챔버에 대한 시퀀스 제약을 생성할 수 있다. 클라이언트 디바이스(170)는 위에서 논의한 방법들에 따라 시퀀스 제약들을 생성할 수 있다.

[0064] [0061] 일부 실시예들에서, 기판들의 묶음 내의 각각의 기판에 대한 시퀀스는 동일하지 않을 수 있다. 이에 따라, 클라이언트 디바이스(170)는 2개의 기판들(즉, W_x, W_y)로 시작하여 묶음 내의 모든 기판들이 추가될 때까지 추가 기판(예컨대, W_z)을 추가함으로써 부분별 처리를 위한 타임테이블을 생성할 수 있다.

[0065] [0062] 블록(310)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 기판들의 묶음에 분석될 임의의 기판들이 남아 있는지 여부를 결정한다. 기판들의 묶음에 분석될 기판들이 남아 있다면, 흐름은 블록(312)으로 진행한다. 그러나 블록(310)에서 클라이언트 디바이스(170)가 기판들의 묶음에 남아 있는 기판들이 없다고 결정한다면, 흐름은 블록(314)으로 진행한다.

[0066] [0063] 블록(312)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 기판(예컨대, W_z)을 처리될 기판들의 리스트에 추가하는데, 즉, 클라이언트 디바이스(170)는 W_z를 처리될 기판들(W_x, W_y)에 추가한다. 그 다음, 방법(300)은 기판들(W_x, W_y, W_z)을 이용한 분석을 위해 블록(304)으로 되돌아간다.

[0067] [0064] 블록(314)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 GPU(180)에 의해) 처리 모델에 기초하여 병렬 입력들을 생성한다. 블록(314)은 도 2의 블록(204)과 유사할 수 있다.

[0068] [0065] 블록(316)에서, 클라이언트 디바이스(170)는 반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 병렬 입력들을 하나 이상의 GPU들(180)의 코어들에 의해 동시에 처리한다. 블록(316)은 도 2의 블록(206)과 유사할 수 있다.

[0069] [0066] 블록(318)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의

해) 블록(316)에서 생성된 병렬 출력들에 기초하여 기판들의 묶음에 대한 타임테이블(예컨대, 스케줄)을 생성한다. 예를 들어, 타임테이블은 각각의 기판의 시작 시간(T_x) 및 각각의 처리 챔버에서의 기판 처리 순서를 포함한다.

[0070] [0067] 일부 실시예들에서, 블록(320)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 타임테이블에 기초하여 스루풋을 예측한다(예컨대, 스루풋 모델링을 수행한다). 예를 들어, 클라이언트 디바이스(170)는 타임테이블에 기초하여 툴(예컨대, 도 1의 클러스터 툴(100))이 설정된 양의 시간 내에(예컨대, 1시간 내에) 처리할 수 있는 기판들의 수를 예측할 수 있다. 블록(320)에서, 클라이언트 디바이스(170)(예컨대, 그리고 GPU)는 클러스터 툴(예컨대, 클러스터 툴(100))에 연결될 수 있는 것이 아니라, 스루풋을 예측하기 위한 수학적 모델 솔버(solver)로서 사용될 수 있다. 일부 실시예들에서, 클라이언트 디바이스(170)는 예측 스루풋을 생성하고 예측 스루풋을 다른 디바이스에 송신한다. 일부 실시예들에서, 클라이언트 디바이스(170)는 예측 스루풋을 생성하고 예측 스루풋을 클라이언트 디바이스(170)의 GUI(graphical user interface)를 통해 사용자에게 디스플레이한다.

[0071] [0068] 블록(322)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 임의의 업데이트들이 있는지 여부를 결정한다. 업데이트들은 클라이언트 디바이스(170)의 GUI를 통한 사용자 입력을 통해 수신될 수 있다. 업데이트들은 처리 모델의 적어도 하나의 제약, 기판들의 리스트, 적어도 하나의 처리 시퀀스 또는 적어도 하나의 할당된 처리 챔버 중 하나 이상에 대한 수정들(또는 이들의 새로 수신된 것들)일 수 있다. 예를 들어, 예측 스루풋을 디스플레이 또는 송신하는(디스플레이되게 하는) 것에 대한 응답으로, 클라이언트 디바이스(170)는 (예컨대, 제약, 리스트, 처리 시퀀스, 처리 챔버, 처리 모델 등에 대한) 하나 이상의 업데이트들을 수신하여 예측 스루풋에 대한 업데이트들의 영향들(예컨대, 예측 스루풋을 변경하는 것)을 결정할 수 있다. 업데이트들이 있다면, 흐름은 블록(324)으로 진행한다. 그러나 블록(322)에서 클라이언트 디바이스(170)가 업데이트들이 없다고 결정한다면, 흐름은 종료된다.

[0072] [0069] 블록(324)에서, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) (업데이트들에 기초하여) 업데이트된 처리 모델을 생성하도록 처리 모델을 업데이트하고, 흐름은 블록(314)에서 계속된다. 원하는 예측 스루풋이 달성될 때까지 블록(314)에서부터 블록(324)까지의 흐름이 계속될 수 있다.

[0073] [0070] 스루풋 예측을 통해, 클라이언트 디바이스(170)는 초기 스테이지들에서 임의의 새로운 장비 아키텍처를 평가하여 스루풋이 어떻게 될 것인지를 결정할 수 있으며, 많은 대안들 중 최상의 대안이 투자 및 추가 개발을 위해 선택될 수 있다. 기존 툴들의 경우, 클라이언트 디바이스(170)는 스루풋에 대한 임의의 변경들의 영향들을 정량화하기 위해 스루풋을 예측하기 위한 모델링을 수행할 수 있다. 변경은 기판 처리 동작들에, 툴 토폴러지에, 또는 프로세스 제약들 중 임의의 프로세스 제약에 있을 수 있다. 클라이언트 디바이스(170)가 스루풋을 예측하는 것은 고객들에게 정확한 스루풋 추정치들을 제공할 수 있다. 클라이언트 디바이스(170)가 스루풋을 예측하는 것은, 결함 또는 예상치 못한 이벤트가 발생할 때 실제 툴이 어떻게 반응할지를 시뮬레이션하는 데 사용될 수 있다. 이러한 시뮬레이션들의 결과들은 몇 분 내에 이용 가능할 수 있으므로, 이는 테스트 및 개발에서 항목들을 절약한다.

[0074] [0071] 일부 실시예들에서, 스루풋을 예측하는(예컨대, 그리고 처리 모델을 업데이트하는) 것에 대한 응답으로, 클라이언트 디바이스는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 타임테이블에 기초하여 반도체 기판들의 묶음이 처리되게 한다. 예를 들어, 클라이언트 디바이스(170)는 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 타임테이블을 제어기(160)에 송신할 수 있고, 타임테이블에 기초하여 제어기(160)가 기판 처리를 시작하게(예컨대, 기판 처리를 시작하도록 클러스터 툴(100)을 제어하게) 할 수 있다. 클라이언트 디바이스(170)와 제어기가 동일한 것인 그러한 실시예들에서, 클라이언트 디바이스(170)는 타임테이블을 송신하지 않을 수 있다.

[0075] [0072] 도 4는 특정 실시예들에 따라, 하나 이상의 GPU들에 의해 병렬 입력들을 처리하기 위한 방법(400)의 흐름도이다.

[0076] [0073] 도 4를 참조하면, 블록(402)에서 클라이언트 디바이스(170)가 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 반도체 기판들의 묶음에 대한 처리 모델을 생성한다. 블록(402)은 도 2의 블록(202) 또는 도 3의 블록들(302-312) 중 하나 이상과 유사할 수 있다.

[0077] [0074] 블록(404)에서, 클라이언트 디바이스(170)는 제1 코어들을 포함하는 제1 GPU(180A)에 의해 처리 모델을

수신한다. 클라이언트 디바이스(170)는 GPU들(180)의 클러스터(예컨대, 2개 이상의 GPU들)를 포함할 수 있다. 일부 실시예들에서, 제1 GPU(180A)는 GPU들(180)의 클러스터의 마스터 GPU(예컨대, 마스터 노드)이다.

[0078] [0075] 블록(406)에서, 클라이언트 디바이스(170)는 제1 GPU(180A)에 의해 처리 모델에 기초하여 병렬 입력들을 생성한다. 블록(406)은 도 2의 블록(204)과 유사할 수 있다.

[0079] [0076] 블록(408)에서, 클라이언트 디바이스(170)는 제1 GPU(180A)에 의해, 병렬 입력들의 제1 수량이 제1 GPU(180A)의 제1 코어들의 제2 수량을 초과하는지 여부를 결정한다. 병렬 입력들의 제1 수량이 제1 코어들의 제2 수량을 초과하지 않는다면(예컨대, 500개의 병렬 입력들의 제1 수량이 786개의 제1 코어들의 제2 수량을 초과하지 않는다면), 흐름은 블록(410)으로 진행한다. 그러나 블록(408)에서, 클라이언트 디바이스(170)(예컨대, 제1 GPU(180A))가 병렬 입력들의 제1 수량이 제1 코어들의 제2 수량을 초과한다(예컨대, 1,000개의 병렬 입력들의 제1 수량이 786개의 제1 코어들의 제2 수량을 초과한다)고 결정한다면, 흐름은 블록(412)으로 진행한다.

[0080] [0077] 블록(410)에서, 클라이언트 디바이스(170)는 반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 병렬 입력들을 제1 GPU(180A)의 제1 코어들에 의해 동시에 처리한다. 블록(410)은 도 2의 블록(206)과 유사할 수 있다.

[0081] [0078] 블록(412)에서, 클라이언트 디바이스(170)는 제1 GPU(180A)에 의해, 병렬 입력들의 제1 서브세트를 제1 GPU(180A)의 제1 코어들에 그리고 병렬 입력들의 제2 서브세트를 제2 GPU(180B)의 제2 코어들에 할당한다. 일부 실시예들에서, 제1 GPU(180A)는 병렬 입력들의 실질적으로 절반을 제1 GPU(180A)의 제1 코어들에 그리고 병렬 입력들의 실질적으로 절반을 제2 GPU(180B)의 제2 코어들에 할당한다. 일부 실시예들에서, 제1 GPU(180A)는 병렬 입력들을 3개 이상의 GPU들(180)에 할당한다.

[0082] [0079] 블록(414)에서, 클라이언트 디바이스(170)는 반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 제1 GPU(180A)의 제1 코어들에 의해 제1 서브세트를 그리고 제2 GPU(180B)의 제2 코어들에 의해 제2 서브세트를 동시에 처리한다. 블록(414)은 도 2의 블록(206)과 유사할 수 있다.

[0083] [0080] 블록(416)에서, 클라이언트 디바이스(170)는 병렬 출력들에 기초하여 타임테이블을 생성한다. 블록(416)은 도 3의 블록(318)과 유사할 수 있다.

[0084] [0081] 블록(418)에서, 클라이언트 디바이스(170)는 선택적으로, 타임테이블에 기초하여 스루풋을 예측한다. 블록(418)은 도 3의 블록(320)과 유사할 수 있다. (예컨대, 처리 모델, 제약, 처리 시퀀스, 처리 챔버 등에 대한) 업데이트들의 수신에 대한 응답으로, 처리 모델이 업데이트될 수 있고 흐름은 블록(402)에서 계속될 수 있다.

[0085] [0082] 블록(420)에서, 클라이언트 디바이스(170)는 타임테이블에 기초하여 반도체 기판들의 묶음이 처리되게 한다. 블록(420)은 도 3의 블록(322) 또는 도 2의 블록(208)과 유사할 수 있다.

[0086] [0083] 블록(422)에서, 클라이언트 디바이스(170)는 (예컨대, 통합 기판 처리 시스템에서) 결함이 발생했는지 여부를 (예컨대, 클라이언트 디바이스(170)의 처리 디바이스에 의해) 결정한다. 클라이언트 디바이스(170)가 결함이 발생했다고 결정한다면, 흐름은 블록(402)에서 계속되며, 여기서는 클라이언트 디바이스가 결함에 기초하여, 업데이트된 처리 모델을 생성한다. 그러나 블록(422)에서, 클라이언트 디바이스(170)가 결함이 발생하지 않았다고 결정한다면(예컨대, 그리고 기판 처리가 종료되었다면), 방법(400)이 종료될 수 있다.

[0087] [0084] 도 5는 특정 실시예들에 따른 컴퓨팅 플랫폼(500)을 예시한다. 컴퓨팅 플랫폼(500)은 제어기(510)(예컨대, 제어기(160)) 및 클라이언트 디바이스(550)(예컨대, 클라이언트 디바이스(170))를 포함한다. 제어기(510)는 처리 디바이스(512), 메모리(514), 저장소(516) 및 네트워크 인터페이스(518)를 포함한다. 일부 실시예들에서, 제어기(510)는 이에 결합된 하나 이상의 입력/출력(I/O: input/output) 디바이스들(520)을 더 포함할 수 있다. 처리 디바이스(512)는 메모리(514)에 저장된 프로그램 코드(522)와 같은 프로그래밍 명령들을 리트리브하고 실행한다. 처리 디바이스(512)는 단일 처리 디바이스, 다수의 처리 디바이스들, 다수의 처리 코어들을 갖는 단일 처리 디바이스, 프로세서, CPU(central processing unit) 등을 나타내도록 것으로 포함된다.

[0088] [0085] 저장소(516)는 디스크 드라이브 저장소일 수 있다. 단일 유닛으로 도시되었지만, 저장소(516)는 고정 디스크 드라이브들, 착탈식 메모리 카드들, 광 저장소, NAS(network attached storage) 또는 SAN(storage-area-network)과 같은 고정 및/또는 착탈식 저장 디바이스들의 조합일 수 있다. 네트워크 인터페이스(518)는 제어기(510)가 네트워크(530)(예컨대, 네트워크(190))를 통해 예를 들어, 클라이언트 디바이스(550)와 같은 다른 컴퓨터들과 통신할 수 있게 하는 임의의 타입의 네트워크 통신들일 수 있다.

[0089] [0086] 클라이언트 디바이스(550)는 처리 디바이스(552), 메모리(554), 저장소(556) 및 네트워크 인터페이스(558)를 포함한다. 일부 실시예들에서, 클라이언트 디바이스(550)는 이에 결합된 하나 이상의 I/O 디바이스들(560)을 더 포함할 수 있다. 처리 디바이스(552)는 단일 처리 디바이스, 다수의 처리 디바이스들, 다수의 처리 코어들을 갖는 단일 처리 디바이스, 프로세서, CPU 등을 나타내도록 포함된다. 클라이언트 디바이스(550)는 하나 이상의 GPU들(580)(예컨대, GPU들(180))을 더 포함할 수 있다.

[0090] [0087] 처리 디바이스(552)는 처리 모델 생성기(562), 타임테이블 생성기(564) 및 예측 스루풋 생성기(565)를 포함할 수 있다. 처리 모델 생성기(562)는, 처리 시퀀스의 각각의 동작에 대해 각각의 기판에 처리 챔버를 할당하고, 후속하여 처리 챔버 할당에 기초하여 처리 모델(572)을 생성하도록 구성될 수 있다. 예를 들어, 처리 모델 생성기(562)는 도 2 - 도 4와 관련하여 위에서 논의한 하나 이상의 블록들의 프로세스들을 실행하도록 구성될 수 있다. 생성된 처리 모델은 저장소(556)에 저장될 수 있다. 예를 들어, 처리 모델(572)은 저장소(556)에 있을 수 있다. 타임테이블 생성기(564)는 병렬 출력(574)에 기초하여 처리 타임테이블을 생성하도록 구성된다. 예를 들어, 타임테이블 생성기(564)는 도 3의 블록(318) 또는 도 4의 블록(416)에 따라 위에서 논의한 프로세스들을 실행하도록 구성될 수 있다. 생성된 타임테이블들은 저장소(556)에 저장될 수 있다. 예를 들어, 타임테이블(576)은 저장소(556)에 있을 수 있다. 예측 스루풋 생성기(565)는 타임테이블에 기초하여 스루풋을 예측하도록 구성된다. 예를 들어, 예측 스루풋 생성기(565)는 도 3의 블록(320) 또는 도 4의 블록(418)에 따라 위에서 논의한 프로세스들을 실행하도록 구성될 수 있다.

[0091] [0088] 메모리(554)는 프로그램 코드(566)를 포함한다. 처리 디바이스(552) 또는 하나 이상의 GPU들(580) 중 하나 이상은 메모리(554)에 저장된 프로그램 코드(566)와 같은 프로그래밍 명령들을 리트리브하고 실행할 수 있다. 프로그램 코드(566)는 (예컨대, 처리 스케줄에 기초하여, 타임테이블에 기초하여, 병렬 출력들에 기초하는 식으로) 기판들의 묶음이 처리되게 하는 명령들을 실행하도록 구성될 수 있다. 예를 들어, 프로그램 코드(566)는 도 2 - 도 4와 관련하여 위에서 논의한 하나 이상의 블록들을 포함할 수 있다.

[0092] [0089] 하나 이상의 GPU들(580)은 코어들(586, 588)을 포함할 수 있다(예컨대, GPU(580A)는 코어들(586A-N)을 포함하고 GPU(580N)는 코어들(588A-N)을 포함한다). GPU들(580) 중 하나 이상은 병렬 입력 생성기(582), 병렬 출력 생성기(584) 또는 병렬 출력 선택기(592) 중 하나 이상을 포함할 수 있다. 병렬 출력 생성기(584)는 코어(586 또는 588)를 포함할 수 있다.

[0093] [0090] 일부 실시예들에서, GPU(580A)는 처리 모델(572)을 수신하고 병렬 출력(574)을 출력한다(예컨대, GPU(580A)는 병렬 입력 생성기(582A) 및 병렬 출력 선택기(592A)를 포함한다). 일부 실시예들에서, 하나 이상의 GPU들(580)은 병렬 입력들을 수신하고 병렬 출력들을 출력한다(예컨대, 처리 디바이스(552)는 병렬 입력 생성기(582A) 및 병렬 출력 선택기(592A)를 포함한다).

[0094] [0091] 일례로, 처리 모델 생성기(562)는 반도체 기판들의 묶음에 대한 처리 모델(572)을 생성할 수 있다(예컨대, 도 2의 블록(202), 도 3의 블록들(302-312), 도 4의 블록(402) 등). 병렬 입력 생성기(582A)는 처리 모델(572)을 수신할 수 있고 처리 모델(572)에 기초하여 병렬 입력들을 생성할 수 있다(예컨대, 도 2의 블록(204), 도 3의 블록(314), 도 4의 블록들(404-406) 등).

[0095] [0092] 병렬 입력 생성기(582A)가 병렬 입력들의 제1 수량이 GPU(580A)의 제1 코어들(586)의 제2 수량을 초과하지 않는다고 결정하는 것에 대한 응답으로, 병렬 입력 생성기(582A)는 병렬 출력 생성기(584A)에 병렬 입력들을 송신한다(예컨대, 병렬 입력 생성기(582A)는 병렬 입력들 각각을 병렬 출력 생성기(584A)의 개별 코어들(586)에 분배한다).

[0096] [0093] 병렬 입력 생성기(582A)가 병렬 입력들의 제1 수량이 GPU(580A)의 제1 코어들(586)의 제2 수량을 초과한다고 결정하는 것에 대한 응답으로, 병렬 입력 생성기(582A)는 2개 이상의 병렬 출력 생성기들(584)에 병렬 입력들을 송신한다(예컨대, 병렬 입력 생성기(582A)는 병렬 입력들을 2개 이상의 병렬 출력 생성기들(584)의 개별 코어들(586, 588)에 분배한다).

[0097] [0094] 병렬 입력 생성기(582A)가 병렬 입력들의 제1 수량이 클라이언트 디바이스(550)의 GPU(580A)의 전체 코어들의 제3 수량을 초과한다고 결정하는 것에 대한 응답으로, 병렬 입력 생성기(582A)는 동시에 처리될 제1 세트의 병렬 입력들을 GPU들의 코어들에 분배할 수 있다. 코어들 각각에 대해, 코어가 이용 가능하다면(예컨대, 대응하는 병렬 입력의 처리를 완료했으면), 병렬 입력 생성기(582A)는 이용 가능한 코어에 다른 병렬 입력을 분배할 수 있다. 병렬 입력 생성기(582A)는 모든 병렬 입력들이 처리될 때까지 병렬 입력들을 이용 가능한 코어들에 계속 분배할 수 있다.

[0098] [0095] 병렬 출력 선택기(592)(예컨대, GPU(580A)의 병렬 출력 선택기(592A))는 병렬 출력들을 비교하여 (예컨대, 모든 병렬 출력들과 비교하여) 최소 처리 지속기간을 제공하는 병렬 출력(574)을 선택할 수 있다. 일부 실시예들에서, 병렬 출력 선택기(592)는 GPU(580A)에 위치된다. 일부 실시예들에서, 병렬 출력 선택기(592)는 GPU들(580) 각각에 위치된다. 일부 실시예들에서, 병렬 출력 선택기(592)는 처리 디바이스(552)에 위치된다.

[0099] [0096] 타임테이블 생성기(564)는 (예컨대, 병렬 출력 선택기(592)에 의해 선택된) 병렬 출력(574)을 수신할 수 있고 병렬 출력(574)에 기초하여 타임테이블(576)을 생성할 수 있다. 네트워크 인터페이스(558)는 타임테이블(576)을 수신하고 타임테이블(576)을 네트워크(530)를 통해 제어기(510)의 네트워크 인터페이스(518)에 송신하여, 타임테이블(576)을 기초로 반도체 기판들의 묶음이 기판 처리 시스템에서 처리되게 할 수 있다.

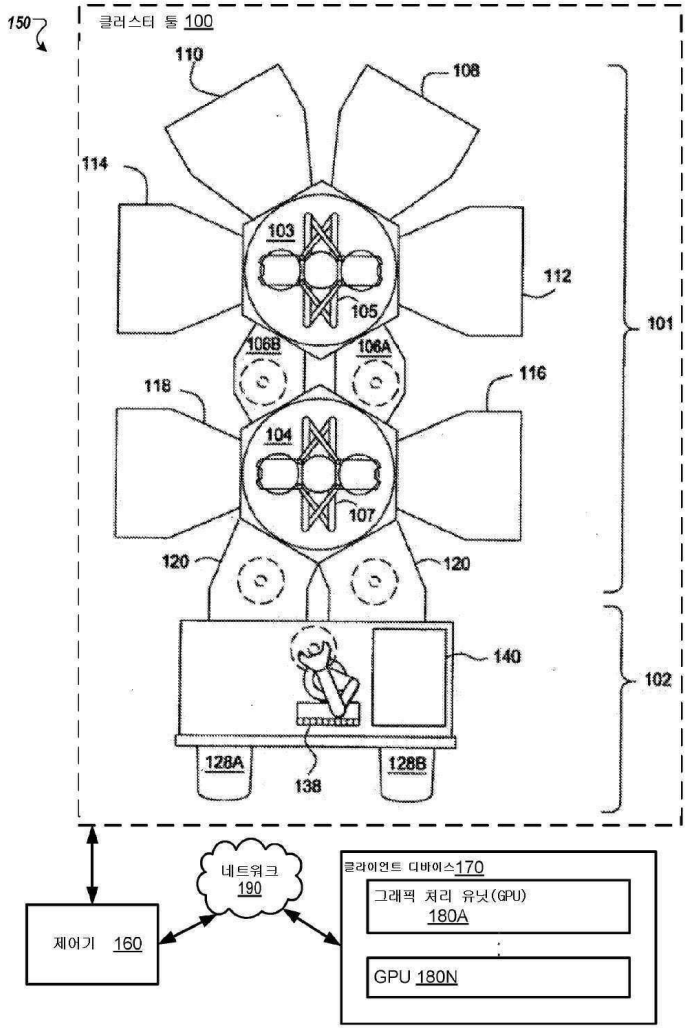
[0100] [0097] 전술한 내용은 본 명세서에서 설명되는 실시예들에 관한 것이지만, 실시예들의 기본 범위를 벗어나지 않으면서 다른 실시예들 및 추가 실시예들이 안출될 수 있다. 예를 들어, 본 개시내용의 양상들은 하드웨어 또는 소프트웨어로, 또는 하드웨어와 소프트웨어의 조합으로 구현될 수 있다. 본 명세서에서 설명되는 일 실시예는 컴퓨터 시스템과 함께 사용할 프로그램 제품으로서 구현될 수 있다. 프로그램 제품의 프로그램(들)은 (본 명세서에서 설명되는 방법들을 포함하는) 실시예들의 기능들을 정의하며, 다양한 컴퓨터 판독 가능 저장 매체들 상에 포함될 수 있다. 예시적인 컴퓨터 판독 가능 저장 매체들은: (ⅰ) 정보가 영구적으로 저장되는 기록이 불가능한 저장 매체(예를 들어, 컴퓨터 내의 읽기 전용 메모리 디바이스들, 이를테면 CD-ROM 드라이브에 의해 판독 가능한 CD-ROM 디스크들, 플래시 메모리, ROM 칩들 또는 임의의 타입의 고체 상태 비휘발성 반도체 메모리); 및 (ⅱ) 변경 가능한 정보가 저장되는 기록 가능한 저장 매체(예를 들어, 디스켓 드라이브 또는 하드 디스크 드라이브 내의 플로피 디스크들 또는 임의의 타입의 고체 상태 랜덤 액세스 반도체 메모리)를 포함하지만, 이에 제한되는 것은 아니다. 이러한 컴퓨터 판독 가능 저장 매체들은 개시된 실시예들의 기능들을 지시하는 컴퓨터 판독 가능 명령들을 보유할 때, 본 개시내용의 실시예들이다.

[0101] [0098] 선행하는 예들은 예시이고 제한이 아니라는 것이 당해 기술분야에서 통상의 지식을 가진 자들에게 인식될 것이다. 그에 대한 모든 치환들, 확장들, 등가물들 및 개선들은, 본 명세서를 읽고 도면들을 연구할 때 당해 기술분야에서 통상의 지식을 가진 자들에게 명백하고 본 개시내용의 진정한 사상 및 범위 내에 포함되는 것으로 의도된다. 따라서 다음의 첨부된 청구항들은 이러한 교시들의 진정한 사상 및 범위 내에 속하는 것으로서 이러한 모든 수정들, 치환들 및 등가물들을 포함하는 것으로 의도된다.

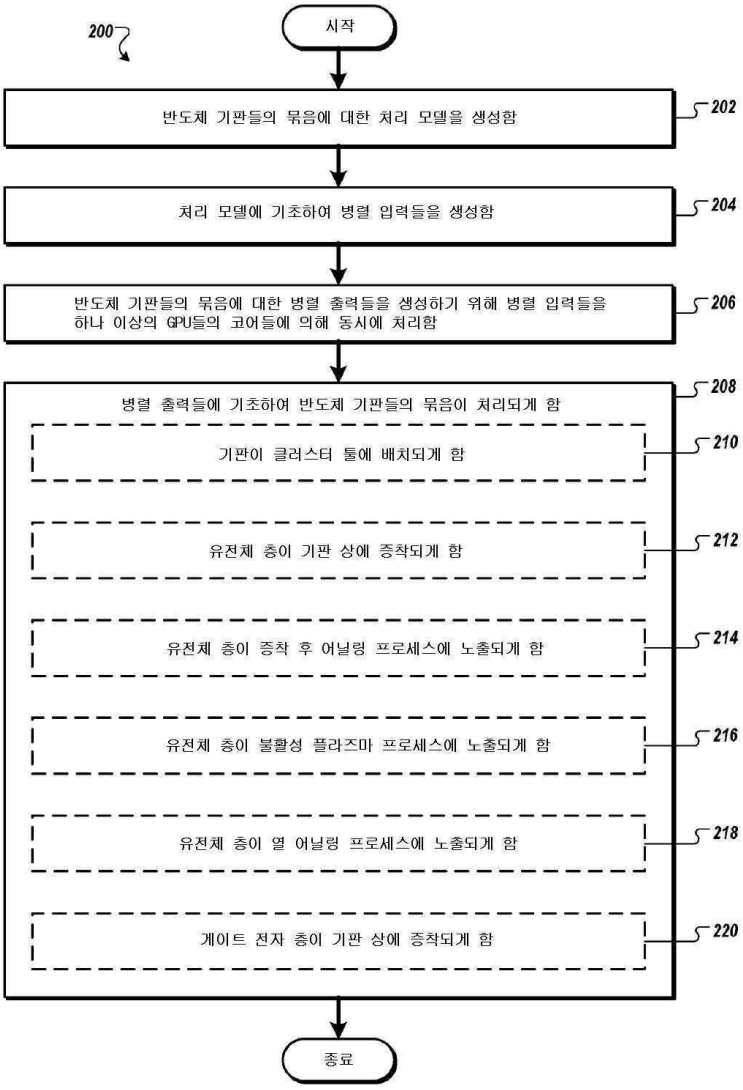
[0102] [0099] 전술한 내용은 본 개시내용의 실시예들에 관한 것이지만, 본 개시내용의 기본 범위를 벗어나지 않으면서 본 개시내용의 다른 실시예들 및 추가 실시예들이 안출될 수 있으며, 본 개시내용의 범위는 하기의 청구항들에 의해 결정된다.

도면

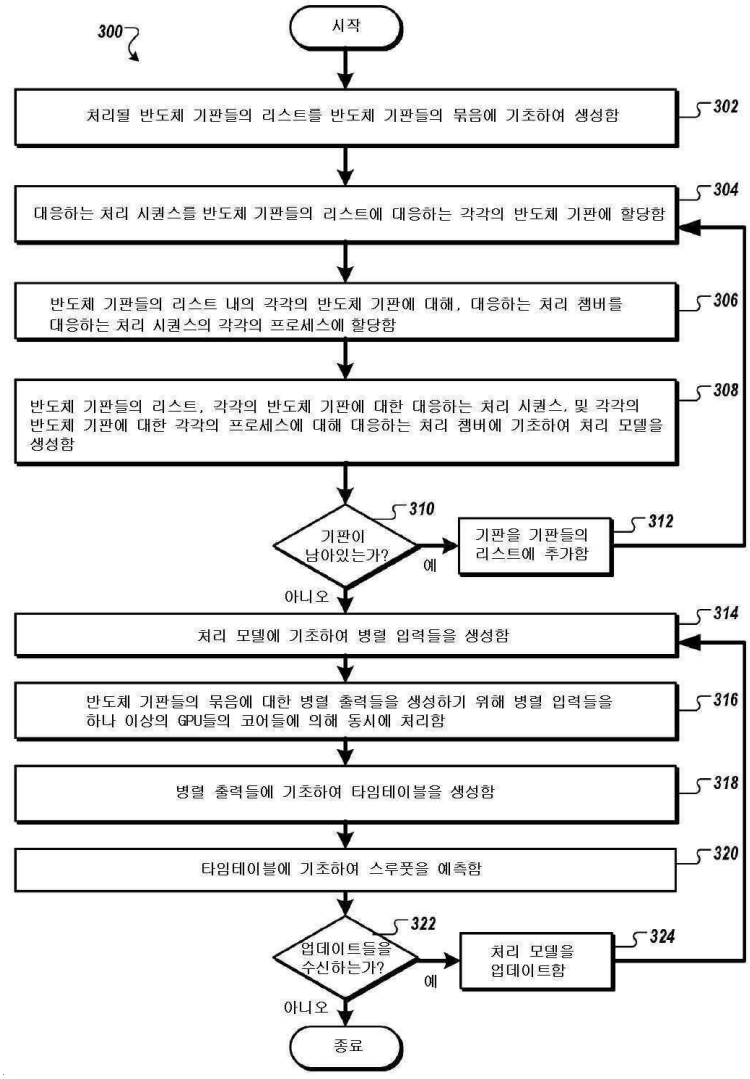
도면1


150
클러스터 툴 100
110
108
114
103
112
105
101
106B
106A
116
118
104
107
120
120
140
102
128A
138
128B
네트워크
190
제어기 160
클라이언트 디바이스170
그래픽 처리 유닛(GPU)
180A
GPU 180N

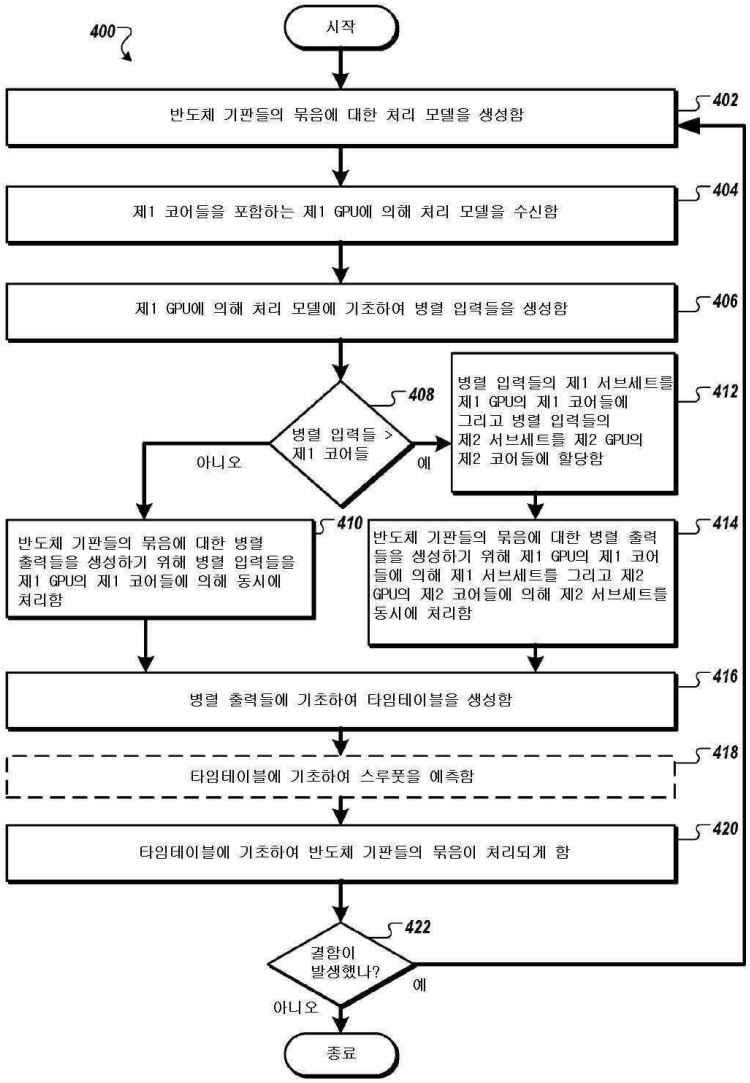
도면2


200
시작
반도체 기판들의 묶음에 대한 처리 모델을 생성함
202
처리 모델에 기초하여 병렬 입력들을 생성함
204
반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 병렬 입력들을 하나 이상의 GPU들의 코어들에 의해 동시에 처리함
206
병렬 출력들에 기초하여 반도체 기판들의 묶음이 처리되게 함
208
기판이 클러스터 툴에 배치되게 함
210
유전체 층이 기판 상에 증착되게 함
212
유전체 층이 증착 후 어닐링 프로세스에 노출되게 함
214
유전체 층이 불활성 플라즈마 프로세스에 노출되게 함
216
유전체 층이 열 어닐링 프로세스에 노출되게 함
218
게이트 전자 층이 기판 상에 증착되게 함
220
종료

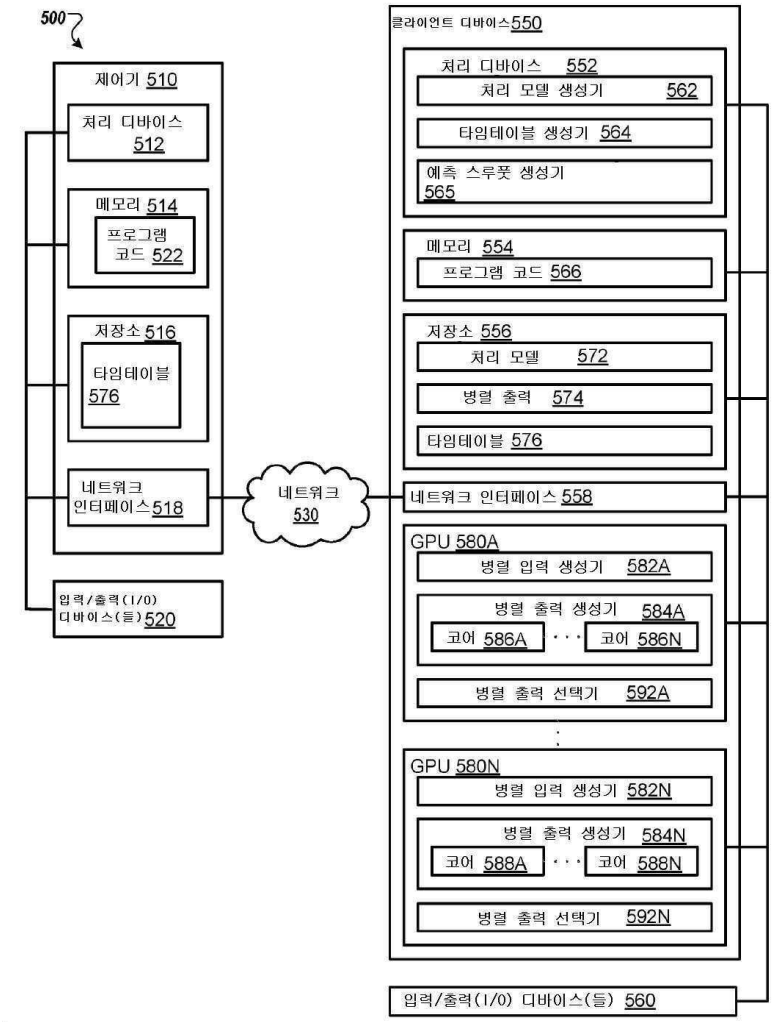
도면3


300
시작
처리될 반도체 기판들의 리스트를 반도체 기판들의 묶음에 기초하여 생성함
302
대응하는 처리 시퀀스를 반도체 기판들의 리스트에 대응하는 각각의 반도체 기판에 할당함
304
반도체 기판들의 리스트 내의 각각의 반도체 기판에 대해, 대응하는 처리 챔버를 대응하는 처리 시퀀스의 각각의 프로세스에 할당함
306
반도체 기판들의 리스트, 각각의 반도체 기판에 대한 대응하는 처리 시퀀스, 및 각각의 반도체 기판에 대한 각각의 프로세스에 대해 대응하는 처리 챔버에 기초하여 처리 모델을 생성함
308
기판이 남아있는가?
310
예
기판을 기판들의 리스트에 추가함
312
아니오
처리 모델에 기초하여 병렬 입력들을 생성함
314
반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 병렬 입력들을 하나 이상의 GPU들의 코어들에 의해 동시에 처리함
316
병렬 출력들에 기초하여 타임테이블을 생성함
318
타임테이블에 기초하여 스루풋을 예측함
320
업데이트들을 수신하는가?
322
예
처리 모델을 업데이트함
324
아니오
종료

도면4


400
시작
반도체 기판들의 묶음에 대한 처리 모델을 생성함
402
제1 코어들을 포함하는 제1 GPU에 의해 처리 모델을 수신함
404
제1 GPU에 의해 처리 모델에 기초하여 병렬 입력들을 생성함
406
병렬 입력들 > 제1 코어들
408
아니오
예
병렬 입력들의 제1 서브세트를 제1 GPU의 제1 코어들에 그리고 병렬 입력들의 제2 서브세트를 제2 GPU의 제2 코어들에 할당함
412
반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 병렬 입력들을 제1 GPU의 제1 코어들에 의해 동시에 처리함
410
반도체 기판들의 묶음에 대한 병렬 출력들을 생성하기 위해 제1 GPU의 제1 코어들에 의해 제1 서브세트를 그리고 제2 GPU의 제2 코어들에 의해 제2 서브세트를 동시에 처리함
414
병렬 출력들에 기초하여 타임테이블을 생성함
416
타임테이블에 기초하여 스루풋을 예측함
418
타임테이블에 기초하여 반도체 기판들의 묶음이 처리되게 함
420
결함이 발생했나?
422
예
아니오
종료

도면5


500
제어기 510
처리 디바이스 512
메모리 514
프로그램 코드 522
저장소 516
타임테이블 576
네트워크 인터페이스 518
입력/출력(I/O) 디바이스(들) 520
네트워크 530
클라이언트 디바이스 550
처리 디바이스 552
처리 모델 생성기 562
타임테이블 생성기 564
예측 스루풋 생성기 565
메모리 554
프로그램 코드 566
저장소 556
처리 모델 572
병렬 출력 574
타임테이블 576
네트워크 인터페이스 558
GPU 580A
병렬 입력 생성기 582A
병렬 출력 생성기 584A
코어 586A
코어 586N
병렬 출력 선택기 592A
GPU 580N
병렬 입력 생성기 582N
병렬 출력 생성기 584N
코어 588A
코어 588N
병렬 출력 선택기 592N
입력/출력(I/O) 디바이스(들) 560