



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년08월30일
 (11) 등록번호 10-1893544
 (24) 등록일자 2018년08월24일

- | | |
|--|--|
| (51) 국제특허분류(Int. Cl.)
G06F 12/08 (2016.01) G06F 12/12 (2016.01)
G11C 7/10 (2015.01)
(21) 출원번호 10-2014-7017417
(22) 출원일자(국제) 2012년11월21일
심사청구일자 2017년11월21일
(85) 번역문제출일자 2014년06월24일
(65) 공개번호 10-2014-0098199
(43) 공개일자 2014년08월07일
(86) 국제출원번호 PCT/US2012/066217
(87) 국제공개번호 WO 2013/081932
국제공개일자 2013년06월06일
(30) 우선권주장
13/307,776 2011년11월30일 미국(US)
(56) 선행기술조사문헌
US20070204107 A1
US20100250856 A1 | (73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 95054 산타 클라라 어거스틴 드
라이브 2485
(72) 발명자
로 가브리엘 에이치.
미국 워싱턴 98007 벨뷰 노스이스트 12번 스트리트
15115
힐 마크 디.
미국 위스콘신 53726 매디슨 체임벌린 애버뉴
2124
(74) 대리인
박장원 |
|--|--|

전체 청구항 수 : 총 19 항

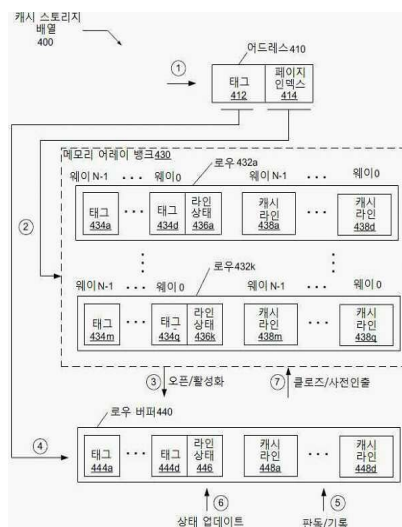
심사관 : 임정복

(54) 발명의 명칭 물리적 로우에 함께 저장된 태그 및 데이터를 구비한 DRAM 캐시

(57) 요약

컴퓨팅 시스템의 대형 로우-기반 메모리에서 효율적 캐시 데이터 액세스를 위한 시스템 및 방법이 개시된다. 컴퓨팅 시스템은 프로세싱 유닛과 집적 3차원(3D) 동적 랜덤 액세스 메모리(DRAM)를 포함한다. 프로세싱 유닛은 캐시로 3D DRAM을 이용할 수 있다. 3D DRAM의 메모리 어레이 블록 내 멀티플 로우 중 각각의 로우는 적어도 멀티플 캐시 태그와, 멀티플 캐시 태그에 의해 표시되는 멀티플 대응 캐시 라인을 저장한다. 프로세싱 유닛으로부터 메모리 요청을 수신함에 응답하여, 3D DRAM은 수신한 메모리 요청 내 캐시 태그에 의해 표시되는 주어진 캐시 라인 상에서 수신한 메모리 요청에 따라 메모리 액세스를 수행한다. 멀티플 DRAM 거래를 이용하기 보다는, 단일, 복합 DRAM 거래를 이용하여 대기시간 및 전력 소모를 감소시킬 수 있다.

대표도 - 도4



명세서

청구범위

청구항 1

방법으로서,

적어도 하나 이상의 캐시 태그와 상기 하나 이상의 캐시 태그에 의해 표시되는 하나 이상의 캐시 라인을 동적 랜덤 액세스 메모리(DRAM) 내 복수의 로우들 중 동일 로우 내에 저장하는 단계;

상기 DRAM에 연결된 프로세싱 유닛으로부터 메모리 요청을 수신하는 단계; 및

상기 메모리 요청을 수신함에 응답하여, 상기 수신된 메모리 요청 내의 제 1 캐시 태그에 의해 표시되는 주어진 캐시 라인에 대해서, 상기 수신된 메모리 요청에 따라 메모리 액세스를 수행하는 단계를 포함하며,

상기 메모리 액세스를 수행하는 단계는,

상기 복수의 로우들의 각각의 로우에 대한 단일 관독을 수행하는 단계, 상기 각각의 로우는 복수의 캐시 태그들과 복수의 캐시 라인들을 저장하며;

상기 제 1 캐시 태그를 상기 각각의 로우에 저장된 상기 복수의 캐시 태그들과 비교하기 전에, 상기 복수의 캐시 태그들 및 상기 복수의 캐시 라인들을 상기 각각의 로우로부터 로우 버퍼로 복사하는 단계; 및

상기 수신된 메모리 요청 내의 캐시 태그와 상기 각각의 로우의 상기 복수의 캐시 태그들을 비교하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 2

제1항에 있어서,

상기 비교하는 단계 이후에, 상기 수신된 메모리 요청 내의 상기 제 1 캐시 태그가 상기 각각의 로우에 저장된 하나 이상의 캐시 태그들 중 제 2 캐시 태그와 매칭하는지를 결정하는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 3

제2항에 있어서,

상기 각각의 로우 내의 상기 제 2 캐시 태그의 위치는 상기 각각의 로우 내의 상기 주어진 캐시 라인의 위치를 나타내는 것을 특징으로 하는 방법.

청구항 4

제3항에 있어서,

상기 DRAM 내의 제어 로직이, 상기 수신된 메모리 요청 내의 캐시 태그와 상기 각각의 로우의 상기 복수의 캐시 태그들을 비교하는 단계를 수행하는 것을 특징으로 하는 방법.

청구항 5

제1항에 있어서,

상기 메모리 액세스를 수행하는 단계는,

상기 각각의 로우에 대해 DRAM 오픈 스테이지(DRAM open stage)를 수행하는 동작;

상기 복수의 캐시 태그들 및 상기 복수의 캐시 라인들을 상기 로우 버퍼에 저장하는 동작; 및

상기 메모리 액세스에 의해 수정된 임의의 데이터 및 상기 메모리 액세스에 의해 수정되지 않은 데이터를 상기 각각의 로우에 다시 복사하는 동작

을 중단없는 동작들의 블록으로서 수행하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 6

제3항에 있어서,

상기 메모리 액세스를 수행하는 단계는, 상기 각각의 로우에 대한 단일 DRAM 클로즈 스테이지(single DRAM closing stage)를 포함하는 것을 특징으로 하는 방법.

청구항 7

제3항에 있어서,

상기 각각의 로우의 상기 복수의 캐시 태그들과 상기 복수의 캐시 라인들은 하나의 물리적 구조로 결합되는 것을 특징으로 하는 방법.

청구항 8

제6항에 있어서,

상기 프로세싱 유닛은 범용 마이크로프로세서, 그래픽 프로세싱 유닛(GPU), 가속 프로세싱 유닛(APU), 및 필드 프로그래머블 게이트 어레이(FPGA) 중 적어도 하나인 것을 특징으로 하는 방법.

청구항 9

동적 랜덤 액세스 메모리(DRAM)로서,

복수의 로우들, 각각의 로우는 적어도 (i) 복수의 캐시 태그들과, (ii) 상기 복수의 캐시 태그들에 의해 표시되는 복수의 캐시 라인들을 저장하도록 구성되고; 그리고

제어 회로를 포함하며,

상기 제어 회로는,

메모리 요청을 수신하고; 그리고

상기 메모리 요청을 수신함에 응답하여, 상기 수신된 메모리 요청 내의 제 1 캐시 태그에 의해 표시되는 주어진 캐시 라인에 대해서, 상기 수신된 메모리 요청에 따라 메모리 액세스를 수행하도록 구성되며,

상기 메모리 액세스를 수행하는 것은,

상기 복수의 로우들의 각각의 로우에 대한 단일 판독을 수행하는 것, 상기 각각의 로우는 복수의 캐시 태그들과 복수의 캐시 라인들을 저장하며;

상기 제 1 캐시 태그를 상기 각각의 로우에 저장된 상기 복수의 캐시 태그들과 비교하기 전에, 상기 복수의 캐시 태그들 및 상기 복수의 캐시 라인들을 상기 각각의 로우로부터 로우 버퍼로 복사하는 것; 및

상기 수신된 메모리 요청 내의 캐시 태그와 상기 각각의 로우의 상기 복수의 캐시 태그들을 비교하는 것을 포함하는 DRAM.

청구항 10

제9항에 있어서,

상기 제어 회로는 또한, 상기 수신된 메모리 요청 내의 상기 제 1 캐시 태그가 상기 각각의 로우에 저장된 하나 이상의 캐시 태그들 중 제 2 캐시 태그와 매칭하는지를 결정하도록 구성되는 것을 특징으로 하는 DRAM.

청구항 11

제10항에 있어서,

상기 복수의 로우들 각각은 상기 복수의 캐시 태그들에 대응하는 메타 데이터를 저장하도록 또한 구성되고,

상기 주어진 캐시 라인을 저장하는 각각의 로우에 대한 단일 판독으로 상기 메모리 액세스를 수행하는 것은, 상

기 메모리 액세스에 기초하여 상기 메타 데이터를 업데이트하는 것을 포함하는 DRAM.

청구항 12

제9항에 있어서,

상기 메모리 액세스를 수행하기 위하여, 상기 DRAM은,

상기 각각의 로우에 대해 DRAM 오픈 스테이지(DRAM open stage)를 수행하는 동작;

상기 복수의 캐시 태그들 및 상기 복수의 캐시 라인들을 상기 로우 버퍼에 저장하는 동작;

상기 메모리 액세스에 의해 수정된 임의의 데이터 및 상기 메모리 액세스에 의해 수정되지 않은 데이터를 상기 각각의 로우에 다시 복사하는 동작; 및

상기 각각의 로우에 대한 DRAM 클로즈 스테이지(DRAM closing stage)를 수행하는 동작

을 중단없는 동작들의 블록으로서 수행하도록 구성되는 것을 특징으로 하는 DRAM.

청구항 13

시스템 인 패키지(SIP)로서,

메모리 요청을 발생시키도록 구성된 제 1 칩 상의 프로세싱 유닛; 및

복수의 로우들을 포함하는 제 2 칩 상의 동적 랜덤 액세스 메모리(DRAM)를 포함하며,

상기 DRAM은 상기 제 1 칩에 연결되고, 상기 DRAM은,

적어도 복수의 캐시 태그들과 상기 복수의 캐시 태그들에 의해 표시되는 복수의 캐시 라인들을 상기 복수의 로우들 중 동일 로우 내에 저장하고;

상기 프로세싱 유닛으로부터 메모리 요청을 수신하고; 그리고

상기 수신된 메모리 요청 내의 제 1 캐시 태그에 의해 표시되는 주어진 캐시 라인에 대해서, 상기 수신된 메모리 요청에 따라 메모리 액세스를 수행하도록 구성되며,

상기 메모리 액세스를 수행하는 것은,

상기 복수의 로우들의 각각의 로우에 대한 단일 판독을 수행하는 것, 상기 각각의 로우는 복수의 캐시 태그들과 복수의 캐시 라인들을 저장하며;

상기 제 1 캐시 태그를 상기 각각의 로우에 저장된 상기 복수의 캐시 태그들과 비교하기 전에, 상기 복수의 캐시 태그들 및 상기 복수의 캐시 라인들을 상기 각각의 로우로부터 로우 버퍼로 복사하는 것을 포함하며,

상기 메모리 액세스를 수행하기 위해, 상기 DRAM은,

상기 각각의 로우에 대해 DRAM 오픈 스테이지(DRAM open stage)를 수행하는 동작;

상기 복수의 캐시 태그들 및 상기 복수의 캐시 라인들을 상기 로우 버퍼에 저장하는 동작;

상기 메모리 액세스에 의해 수정된 임의의 데이터 및 상기 메모리 액세스에 의해 수정되지 않은 데이터를 상기 각각의 로우에 다시 복사하는 동작; 및

상기 각각의 로우에 대한 DRAM 클로즈 스테이지(DRAM closing stage)를 수행하는 동작

을 중단없는 동작들의 블록으로서 수행하도록 구성되는 것을 특징으로 하는 시스템 인 패키지.

청구항 14

제13항에 있어서,

상기 DRAM은 또한, 상기 수신된 메모리 요청 내의 상기 제 1 캐시 태그가 상기 각각의 로우에 저장된 복수의 캐시 태그들 중 제 2 캐시 태그와 매칭하는지를 결정하도록 구성되는 것을 특징으로 하는 시스템 인 패키지.

청구항 15

제14항에 있어서,

상기 각각의 로우 내의 상기 제 2 캐시 태그의 위치는 상기 각각의 로우 내의 상기 주어진 캐시 라인의 위치를 나타내는 것을 특징으로 하는 시스템 인 패키지.

청구항 16

제13항에 있어서,

상기 DRAM은, 상기 수신된 메모리 요청 내의 캐시 태그와 상기 각각의 로우의 상기 복수의 캐시 태그들을 비교하도록 구성된 제어 로직을 포함하는 것을 특징으로 하는 시스템 인 패키지.

청구항 17

제16항에 있어서,

상기 주어진 캐시 라인을 저장하는 각각의 로우에 대한 단일 관독으로 상기 메모리 액세스를 수행하는 것은, 상기 메모리 액세스에 기초하여 메타 데이터를 업데이트하는 것을 포함하는 시스템 인 패키지.

청구항 18

제13항에 있어서,

상기 각각의 로우의 상기 복수의 캐시 태그들과 상기 복수의 캐시 라인들은 하나의 물리적 구조로 결합되는 것을 특징으로 하는 시스템 인 패키지.

청구항 19

제13항에 있어서,

상기 복수의 캐시 태그들 및 상기 복수의 캐시 라인들은 메모리 어레이의 주어진 बैं크에 저장되고, 그리고 상기 메모리 액세스가 시작되면, 상기 주어진 बैं크 내의 데이터에 대해 동작하기를 기다리고 있는 다른 메모리 요청들은 상기 메모리 액세스가 완료될 때까지 대기하도록 지시되는 것을 특징으로 하는 시스템 인 패키지.

청구항 20

삭제

발명의 설명

기술 분야

[0001] 본 발명은 컴퓨팅 시스템에 관한 것이고, 특히, 컴퓨팅 시스템의 대형 로우-기반 메모리에서 효율적인 캐시 데이터 액세스에 관한 것이다.

배경 기술

[0002] 반도체 제조 프로세스가 진보하고 온-다이 기하 치수가 감소함에 따라, 반도체 칩이 더 많은 기능 및 성능을 제공한다. 그러나, 설계 문제는 프로세싱 및 집적 회로 설계 분야의 현대 기술에서도 여전히 발생하고 있고, 이는 잠재적인 이점을 제한할 수 있다. 한가지 문제는 인터커넥트 지연이 2차원 평면 레이아웃 칩의 일련의 세대에서 단위 길이 당 계속 증가한다는 점이다. 또한, 개별 칩 간의 높은 전기 임피던스가 대기시간을 증가시킨다. 추가적으로, 칩을 벗어나 다른 다이에 이르는 신호는 이러한 긴 신호 루트에 대한 와류 커패시턴스 증가로 인해 이 신호들에 대한 전력 소모가 크게 증가할 수 있다(가령, 10배 내지 100배).

[0003] 다른 설계 문제는 다량의 데이터에 액세스하는 대부분의 소프트웨어 애플리케이션이, 연산 시간이 일반적으로 메모리 대역폭에 의해 결정된다는 점에서, 통상적으로 메모리에 구속된다는 것이다. 오프-칩 동적 랜덤 액세스 메모리(DRAM)에 대한 메모리 액세스 대기시간은 수백 내지 수천의 클럭 사이클일 수 있고, 프로세서 설계에서 코어의 숫자 증가는 메모리 대역폭 문제를 두드러지게 하였다. 최근에, 단일 회로에 수직 및 수평으로 모두 집적되는 능동 전자 부품들의 2개 이상의 층을 포함하는 3차원 집적 회로(3D IC)에서 진보가 이루어지고 있다. 시스템 인 패키지(SiP) 또는 칩 스택 멀티-칩 모듈(MCM)이라 알려진 3D 패키징은 개별 칩들을 단일 패키지 내에

적층시킴으로써 공간을 절약한다. 이러한 층 내의 구성요소들은 수직으로든 수평으로든 관계없이 온-칩 시그널링을 이용하여 통신한다. 이러한 시그널링은 알려진 2차원 평면 레이아웃 회로에 비해 인터커넥트 신호 지연이 감소한다.

[0004] 위 설명에서의 제조 경향은 마이크로프로세서 패키지 내에서 기가바이트 단위의 집적 메모리를 이끌 수 있다. 일부 경우에, 오프-칩 메모리에 액세스하기 전에 추가적인 온-칩 스토리지가 최종-레벨 캐시(LLC)와 같은 로우-기반 메모리로 사용될 수 있다. 추가 메모리에 의해 실현되는 실패율 감소는 프로세서와 오프-칩 메모리 사이의 대기시간 갭을 감추는 것을 돕는다. 그러나, 로우-기반 메모리에 대한 캐시 액세스 메커니즘은 이러한 추가 집적 메모리용으로 불충분할 수 있다. 멀티-기가바이트 캐시를 위한 수백 메가바이트와 같은 대형 태그 데이터 어레이는 마이크로프로세서 다이 상에 배치하기에 비싸고 비실용적일 수 있다.

[0005] 64-바이트 라인으로부터 4-킬로바이트(KB) 라인까지 성장하는 것과 같이, 추가적인 집적 메모리를 위한 데이터 캐시 라인의 크기 증가는, 집적 메모리 내 캐시 라인의 수와 대응하는 태그의 크기를 모두 감소시킨다. 그러나, 더터 비트 및 일관성 정보는 원래 캐시 라인 크기(64-바이트 라인)의 입도(granularity) 상에서 여전히 유지될 수 있다. 추가적으로, 몇바이트만을 표적으로 할 때에도 전체 4KB 라인에 액세스할 수 있기 때문에, 데이터 전송이 과도한 대역폭을 소모할 수 있다.

[0006] 집적 DRAM 내 추가 캐시의 데이터 및 태그를 저장 및 액세스하면서 DRAM 액세스 메커니즘을 이용하는 것은 상당한 전력을 소실시킨다. 추가적으로, 이러한 메커니즘은 특히, 고도로 연관된(연상) 온-패키지 캐시용으로, 많은 대역폭을 소모하고, 태그 및 데이터가 순차적 방식으로 판독됨에 따라 너무 많은 시간을 소비한다. 따라서, 온-패키지 DRAM은 많은 추가 데이터 스토리지를 제공하지만, 캐시 및 DRAM 액세스 메커니즘은 비효율적이다.

[0007] 위에 비추어볼 때, 컴퓨팅 시스템의 대형 로우-기반 메모리에서의 효율적 캐시 데이터 액세스를 위한 효율적 방법 및 시스템이 요망된다.

발명의 내용

해결하려는 과제

[0008] 컴퓨팅 시스템의 대형 로우-기반 메모리에서 효율적 캐시 데이터 액세스를 위한 시스템 및 방법이 고려된다.

과제의 해결 수단

[0009] 일 실시예에서, 컴퓨팅 시스템은 프로세싱 유닛과 집적 동적 랜덤 액세스 메모리(DRAM)를 포함한다. 프로세싱 유닛의 예는 범용 마이크로프로세서, 그래픽 프로세싱 유닛(GPU), 가속 프로세싱 유닛(APU), 등을 포함한다. 집적 DRAM은 3차원(3D) DRAM일 수 있고, 프로세싱 유닛과 함께 시스템-인-패키지(SiP)에 포함될 수 있다. 프로세싱 유닛은 캐시로 3D DRAM을 이용할 수 있다.

[0010] 다양한 실시예에서, 3D DRAM은 태그 어레이 및 데이터 어레이를 모두 저장할 수 있다. 3D DRAM의 메모리 어레이 뱅크 내 멀티플 로우 중 각각의 로우는 하나 이상의 캐시 태그와, 하나 이상의 캐시 태그에 의해 표시되는 하나 이상의 대응 캐시 라인을 저장할 수 있다. 프로세싱 유닛으로부터 메모리 요청을 수신함에 응답하여, 3D DRAM은 수신한 메모리 요청 내 캐시 태그에 의해 표시되는 주어진 캐시 라인 상에서 수신한 메모리 요청에 따라 메모리 액세스를 수행할 수 있다. 메모리 액세스 수행은 주어진 캐시 라인을 저장하는 멀티플 로우 중 각자의 로우의 단일 판독을 포함할 수 있다. 멀티플 DRAM 거래를 이용하기 보다는, 단일, 복합 DRAM 거래를 이용하여 대기시간 및 전력 소모를 감소시킬 수 있다.

[0011] 이러한 실시예 및 그의 다른 실시예가 다음의 설명 및 도면을 참조할 때 더욱 이해될 수 있을 것이다.

도면의 간단한 설명

- [0012] 도 1은 컴퓨팅 시스템의 일 실시예의 개략적 블록도다.
- 도 2는 시스템-인-패키지(SiP)의 실시예의 개략적 블록도다.
- 도 3은 컴퓨팅 시스템의 다른 실시예의 개략적 블록도다.
- 도 4는 캐시 스토리지 배열에 액세스하는데 사용되는 시퀀스 스텝의 일 실시예의 개략적 블록도다.
- 도 5는 효율적인 인-패키지 DRAM 액세스 방법의 일 실시예의 개략적인 순서도다.

발명이 다양한 변형 및 대안의 형태를 취할 수 있지만, 구체적인 실시예들이 예를 들어 도면으로 도시되고 여기서 세부적으로 설명된다. 그러나, 도면 및 도면에 대한 상세한 설명은 개시되는 특정 형태로 발명을 제한하고자 함이 아니며, 대신에, 발명은 첨부 청구범위에 의해 규정되는 바와 같은 본 발명의 사상 및 범위 내에 있는 모든 변형예, 등가물, 및 대안들을 커버하는 것이다.

발명을 실시하기 위한 구체적인 내용

- [0013] 다음의 설명에서, 본 발명의 완전한 이해를 돕기 위해 수많은 구체적 세부사항들이 제시된다. 그러나, 발명이 이러한 구체적 세부사항없이 실시될 수 있음을 당업자는 이해하여야 한다. 일부 예에서, 잘 알려진 회로, 구조, 및 기술은 본 발명의 본질을 흐리지 않기 위해 세부적으로 제시되지 않았다.
- [0014] 도 1을 참조하면, 컴퓨팅 시스템(100)의 일 실시예의 개략적 블록도가 도시된다. 도시되는 바와 같이, 마이크로프로세서(110)는 대응하는 하나 이상의 캐시 메모리 서브시스템(124a-124b)에 연결되는 하나 이상의 프로세서 코어(122a-122b)를 포함할 수 있다. 마이크로프로세서는 인터페이스 로직(140), 메모리 컨트롤러(130), 시스템 통신 로직(126), 및 공유 캐시 메모리 서브시스템(128)을 또한 포함할 수 있다. 일 실시예에서, 마이크로프로세서(110)의 도시되는 기능은 단일 집적 회로 상에 통합된다. 다른 실시예에서, 도시되는 기능은 컴퓨터 마더보드 상의 칩셋에 통합된다.
- [0015] 일 실시예에서, 마이크로프로세서(110)는 모바일 컴퓨터, 스마트폰, 또는 태블릿; 데스크탑; 서버; 또는 그외 다른 것 내의 독립형 시스템이다. 일 실시예에서, 시스템 통신 로직(116)은 시스템 버스다. 다른 실시예에서, 마이크로프로세서(110)는 디스크 메모리(162), DRAM(170), 주변 입/출력(I/O) 장치, 다른 프로세싱 유닛(가령, 다른 범용 마이크로프로세서)(이는 중앙 프로세싱 유닛(CPU), ; 그래픽 프로세싱 유닛(GPU), 가속 프로세싱 유닛(APU), 필드 프로그래머블 게이트 어레이(FPGA), 등이라 불릴 수도 있음)에 프로세서 코어(122a-122b)를 연결하기 위해 다양한 프로토콜 중 하나를 이용하는 로직(126)에 시스템 버스 컨트롤러를 통합한다. 이러한 실시예에서, 시스템 통신 로직(126)은 메모리 컨트롤러(130) 및 인터페이스 로직(140)의 기능을 대체 또는 통합할 수 있다.
- [0016] 컴퓨팅 시스템(100) 내 구성요소들의 추가적인 설명은 짧게 제공된다. 컴퓨팅 시스템(100)이 범용 마이크로프로세서(110)를 포함하는 것으로 일 실시예에서 도시되지만, 다른 용도로 사용되는 다른 실시예에서, 마이크로프로세서(110)는 다른 타입의 프로세싱 유닛으로 대체될 수 있다. 다른 타입의 프로세싱 유닛은 그래픽 프로세싱 유닛(GPU), 필드 프로그래머블 게이트 어레이(FPGA), 또는 가속 프로세싱 유닛(APU)을 포함할 수 있다. 일반적으로, APU는 추가적인 프로세싱 기능을 포함하는 칩이다. 추가적인 프로세싱 기능은 범용 CPU 외부에서의 하나 이상의 연산을 가속시키는데 사용될 수 있다. 일 실시예에서, APU는 GPU, FPGA, 또는 다른 프로세싱 유닛과 동일한 다이 상에 통합되는 범용 CPU를 포함할 수 있어서, 전력 소모를 감소시키면서 이러한 유닛들 간의 데이터 전송 속도를 개선시킨다. 다른 실시예에서, APU는 비디오 프로세싱 및 다른 애플리케이션-전용 가속기를 포함할 수 있다.
- [0017] 컴퓨팅 시스템(100)에 사용되는 주어진 타입의 프로세싱 유닛에 관계없이, 소프트웨어 애플리케이션이 점점 더 많은 데이터에 액세스함에 따라, 메모리 서브시스템이 더 과중하게 이용된다. 대기시간이 더 중대해진다. 인터커넥트 대기시간을 감소시키기 위해 더 많은 온-칩 메모리 스토리지가 사용될 수 있다. 예를 들어, 각각의 캐시 메모리 서브시스템(124a-124b)은 각각 프로세서 코어(122a-122b)에 대한 메모리 대기시간을 감소시킬 수 있다. 추가적으로, 마이크로프로세서(110)는 오프-칩 DRAM(170) 및/또는 오프-칩 디스크 메모리(162)에 액세스하기 전에 최종-레벨 캐시(LLC)로 공유 캐시 메모리 서브시스템(128)을 포함할 수 있다.
- [0018] 캐시 메모리 서브시스템(124a-124b, 128)에 의해 제공되는 추가 메모리에 의해 실현되는 실패율 감소는 프로세서 코어(122a-122b) 중 주어진 하나와 오프-칩 메모리 사이에서 대기시간을 감추는 것을 돕는다. 그러나, 각각의 캐시 메모리 서브시스템(124a-124b, 128)에 대해 이용할 공간이 제한되어 있다. 따라서, 각각의 캐시 메모리 서브시스템(124a-124b, 128)에 대해 각자 크기가 제한되고, 상당한 수의 액세스가 DRAM(170) 및/또는 디스크 메모리(162)와 같은 오프-칩 메모리로 여전히 전송된다.
- [0019] 메모리 대기시간 감소를 위한 시도를 계속하면서, 온-칩 메모리 스토리지를 더 제공하기 위해 다른 기술이 이용될 수 있다. 도 1에 도시되지 않지만, 일 실시예에서, 마이크로프로세서(110)는 3차원 집적 회로(3D IC)를 이용하여 최종-레벨 캐시(LLC)를 제공할 수 있다. 컴퓨팅 시스템에 대한 이러한 실시예의 추가적인 세부사항은 나중에 도 2 및 도 3의 설명에서 제공된다. 3D 집적 DRAM은 오프-칩 메모리 액세스를 감소시키기 위해 작은 대기시간의 인터커넥트와 상당히 많은 양의 추가 온-칩 메모리 스토리지를 제공할 수 있다. 마이크로프로세서(110) 내

의 인터페이스 로직은 3D 집적 DRAM에 대해 어드레스 변환을 제공할 수 있다. 3D 집적 DRAM 내의 제어 로직은 3D 집적 DRAM 내의 멀티플 메모리 어레이 뱅크 내 멀티플 로우 중 주어진 로우의 액세스를 위해 사용되는 전형적인 스텝을 변경시킬 수 있다. 3D IC를 이용하는 실시예를 더 설명하기 전에, 도 1에 도시되는 컴퓨팅 시스템 (100) 내 구성요소들의 추가적인 설명이 제공된다.

- [0020] 각각의 프로세서 코어(122a-122b)는 주어진 명령어 세트에 따라 명령어를 실행하기 위한 회로를 포함할 수 있다. 예를 들어, x86 명령어 세트 구조(ISA)가 선택될 수 있다. 대안으로서, Alpha, PowerPC 또는 다른 명령어 세트 구조가 선택될 수 있다. 일 실시예에서, 각각의 프로세서 코어(122a-122b)는 주어진 ISA의 명령어를 처리하는데 사용되는 슈퍼스칼라, 멀티-스레디드 마이크로아키텍처를 포함할 수 있다.
- [0021] 캐시 메모리 서브시스템(124a-124b, 128)은 데이터의 블록을 저장하도록 구성되는 고속 캐시 메모리를 포함할 수 있다. 여기서 사용되는 바와 같이, "블록"은 일관성 용도를 위한 유닛으로 취급되는, 인접 메모리 위치에 저장되는 한 세트의 바이트다. 여기서 사용되는 바와 같이, 각각의 용어 "캐시 블록", "블록", "캐시 라인", 및 "라인"은 상호호환가능하다. 일부 실시예에서, 블록은 캐시의 할당 및 할당해제의 유닛일 수도 있다. 블록 내 바이트의 수는 설계 선택에 따라 변경될 수 있고, 임의의 크기를 가질 수 있다. 추가적으로, 각각의 용어 "캐시 태그", "캐시 라인 태그", 및 "캐시 블록 태그"는 상호호환가능하다.
- [0022] 각각의 캐시 메모리 서브시스템(124a-124b)은 대응하는 캐시 컨트롤러에 연결되는 캐시 메모리 또는 캐시 어레이를 포함할 수 있다. 캐시 메모리 서브시스템(124a-124b, 128)은 캐시의 계층으로 구현될 수 있다. (계층 내에서) 프로세서 코어(122a-122b) 근처에 위치하는 캐시는, 요망될 경우, 프로세서 코어(122a-122b) 내로 통합될 수 있다. 이러한 레벨의 캐시는 멀티-레벨 계층의 레벨-1(L1)일 수 있다. 일 실시예에서, 캐시 메모리 서브시스템(124a-124b) 각각은 L2 캐시 구조를 나타내고, 공유 캐시 메모리 서브시스템(128)은 L3 캐시 구조를 나타낸다. 다른 실시예에서, 캐시 메모리 서브시스템(114) 각각은 L1 캐시 구조를 나타내고, 공유 캐시 서브시스템(118)은 L2 캐시 구조를 나타낸다. 다른 실시예도 가능하고 고려된다.
- [0023] 일반적으로, 프로세서 코어(122a-122b)는 데이터 및 명령어를 위해 캐시 메모리 서브시스템(124a-124b)에 각각 액세스한다. 여기서 사용되는 바와 같이, 캐시 메모리 서브시스템에 관한 "액세스"라는 용어는 대응하는 요청 어드레스의 요청받은 데이터가 캐시에 존재할 경우 캐시 히트를 나타낼 수 있는 판독 또는 기록 요청 작동을 수행하는 것을 의미한다. 대안으로서, 요청받은 데이터가 캐시에 존재하지 않을 경우 판독 또는 기록 작동이 캐시 미스로 나타날 수 있다.
- [0024] 요청받은 블록이 각각의 캐시 메모리 서브시스템(124a-124b)에 또는 공유 캐시 메모리 서브시스템(128)에서 발견되지 않는 경우와 같이, 캐시 미스가 발생할 경우, 판독 요청이 발생되어 메모리 컨트롤러(130)에 송신된다. 메모리 컨트롤러(130)는 요청받은 블록에 대응하는 어드레스를 변환할 수 있고, 판독 요청을 메모리 버스(150)를 통해 오프-칩 DRAM(170)에 송신할 수 있다. 오프-칩 DRAM(170)은 오프-칩 디스크 메모리(162)로부터 I/O 컨트롤러 및 버스(160)와 메모리 버스(150)를 통해 데이터로 충전될 수 있다.
- [0025] 요청받은 블록을 갖는 대응하는 캐시 필 라인은 원래의 판독 또는 기록 요청을 완성하기 위해 오프-칩 DRAM(170)으로부터 캐시 메모리 서브시스템(124a-124b) 중 대응하는 하나에 운반될 수 있다. 캐시 필 라인은 캐시의 하나 이상의 레벨에 배치될 수 있다. 추가적으로, 캐시 필 라인은 캐시 내 대응하는 세트 내에 배치될 수 있다. 대응하는 세트 내에 가용한 웨이(ways)가 없을 경우, 통상적으로 최소 최근 사용(LRU) 알고리즘은 세트 내 어떤 웨이가 데이터를 되겨시키고 캐시 필 라인 데이터에 의해 대체될 것인지를 결정한다. 통상적으로, 할당(allocation)은 캐시 계층의 하위 레벨로부터 인출되는 캐시 필 라인을 특정 캐시에 대한 캐시 미스에 후속하여 특정 캐시의 웨이 내로 저장함을 의미한다.
- [0026] 오프-칩 디스크 메모리(162)는 데이터의 비-휘발성, 랜덤 액세스 보조 스토리지를 제공할 수 있다. 일 실시예에서, 오프-칩 디스크 메모리(162)는 하나 이상의 하드 디스크 드라이브(HDD)를 포함할 수 있다. HDD는 통상적으로 하나 이상의 회전 디스크를 포함하고, 각각은 자기 매체로 코팅된다. 이 디스크들은 분당 수천 회전의 속도로 회전한다. 자기 액추에이터는 회전 디스크 위에 자기 판독/기록 장치를 배치하는 기능을 한다.
- [0027] 다른 실시예에서, 오프-칩 디스크 메모리(162)는 고상 디스크(SSD)를 이용한다. 고상 디스크는 고상 드라이브로도 불릴 수 있다. SSD는 HDD 인터페이스를 에뮬레이션할 수 있지만, SSD는 HDD에 발견되는 전기역학적 장치보다는 지속적 데이터를 저장하기 위해 고상 메모리를 이용한다. 예를 들어, SSD는 플래시 메모리의 뱅크들을 포함할 수 있다.
- [0028] 오프-칩 DRAM(170)은 집적 회로 내에 별도의 커패시터에 데이터의 각각의 비트를 저장하는 일 타입의 동적 랜덤

액세스 메모리일 수 있다. 커패시터는 충전 또는 방전될 수 있다. 이러한 두 상태를 이용하여 일 비트의 2개의 논리 값을 나타낼 수 있다. DRAM(170)은 비트 당 단일 트랜지스터 및 커패시터를 이용할 수 있다. 온-칩 동기식 RAM(SRAM)에 사용되는 6개의 트랜지스터에 비해, DRAM은 훨씬 높은 밀도에 도달할 수 있다. HDD 및 플래시 메모리와 달리, DRAM(170)은 비-휘발성 메모리가 아닌 휘발성 메모리일 수 있다. DRAM(170)은 전력을 제거하자마자 데이터를 잃을 수 있다.

[0029] 오프-칩 DRAM(170)은 멀티-채널 메모리 구조를 포함할 수 있다. 이러한 타입의 구조는 사이에 더 많은 통신 채널들을 더함으로써 메모리 컨트롤러(130)에 대한 데이터의 전송 속도를 증가시킬 수 있다. 멀티-채널 구조는 복수의 메모리 모듈 및 마더보드 및/또는 복수의 채널을 지지할 수 있는 카드를 이용한다.

[0030] 일 실시예에서, 각각의 메모리 모듈은 메모리 컨트롤러(130)에 대한 각각의 인터페이스에 대해 동일한 프로토콜을 가질 수 있다. 프로토콜의 일례는 더블 데이터 레이트(DDR) 타입의 프로토콜이다. 프로토콜은 클럭 사이클 당 데이터 전송의 수, 신호 전압 레벨, 신호 타이밍, 신호 및 클럭 위상 및 클럭 주파수와 같은 정보 전송에 사용되는 값을 결정할 수 있다. 프로토콜 예는 DDR2 SDRAM, DDR3 SDRAM, GDDR4(그래픽스 더블 데이터 레이트, 버전 4) SDRAM, 및 GDDR5(그래픽스 더블 데이터 레이트, 버전 5) SDRAM을 포함한다. 메모리 컨트롤러(130)는 메모리 채널에 인터페이스하기 위한, 그리고 대응 프로토콜을 따르기 위한, 제어 회로를 포함할 수 있다. 추가적으로, 메모리 컨트롤러(130)는 메모리 요청을 대기(queuing)시키기 위한 요청 큐를 포함할 수 있다.

[0031] 이제 도 2를 살펴보면, 시스템-인-패키지(SiP)(200, 240)의 실시예의 개략적 블록도가 도시된다. 3차원(3D) 패키징이 컴퓨팅 시스템 내에서 사용될 수 있다. 이러한 타입의 패키징은 시스템 인 패키지(SiP)로 불릴 수 있다. SiP는 하나 이상의 3차원 집적 회로(3D IC)를 포함한다. 3D IC는 단일 회로 내에 수직으로 및/또는 수평으로 집적되는 능동 전자 구성요소들의 2개 이상의 층을 포함한다. 일 실시예에서, 인터포저-기반 집적을 이용할 수 있고, 이 경우 3D IC가 프로세싱 유닛(220) 옆에 배치될 수 있다. 대안으로서, 3D IC가 다른 IC 바로 위에 적층될 수 있다.

[0032] 다이-적층 기술은 고-대역폭 및 저-대기시간 인터커넥트로 동일 패키지에 함께 실리콘(집적 칩)의 복수의 개별 조각들을 물리적으로 적층시키는 제조 프로세스다. 다이는 실리콘 인터포저 상에서 나란히 또는 서로의 바로 위에 수직으로, 적층될 수 있다. SiP용 일 구조는 프로세싱 유닛 옆에 및/또는 위에, 하나 이상의 DRAM 칩을 적층하는 것이다. DRAM 칩을 적층함으로써, 시스템에 대한 매우 큰 캐시가 프로세싱 유닛용으로 실현될 수 있다. 일 실시예에서, 이러한 대형 캐시는 수백 MB(또는 그 이상) 수준의 크기를 가질 수 있다.

[0033] 도시되는 바와 같이, 일 실시예에서, SiP(200)는 수평 저-대기시간 인터커넥트(210)를 통해 프로세싱 유닛(220)과 통신하는 하나 이상의 3차원(3D) DRAM(230, 232) 및 프로세싱 유닛(220)을 포함할 수 있다. 다시, 프로세싱 유닛(220)은 마이크로프로세서로도 불릴 수 있는 범용 CPU, 그래픽 프로세싱 유닛(GPU), 가속 프로세싱 유닛(APU), 필드 프로그래머블 게이트 어레이(FPGA), 또는 캐시와 같은 로우-기반 메모리를 이용하는 다른 데이터 프로세싱 장치일 수 있다.

[0034] 인-패키지 수평 저-대기시간 인터커넥트(210)는 SiP가 사용되지 않을 때 감소된 길이의 인터커넥트 신호 대 (versus) 긴 오프-칩 인터커넥트를 제공한다. 인-패키지 수평 저-대기시간 인터커넥트(210)는 프로세싱 유닛(220) 및 3D DRAM(230, 232)과 같은 칩이 회로 보드 상의 별도의 패키지 내에 장착되는 것처럼 특정 신호 및 프로토콜을 이용할 수 있다. SiP(200)는 패키지 외부 연결(214)에 도달하는 후면 비아 또는 관통-벌크 실리콘 비아(212)를 추가로 포함할 수 있다. 패키지 외부 연결(214)은 입/출력(I/O) 신호 및 전력 신호에 사용될 수 있다.

[0035] 다른 실시예에서, SiP(240)는 프로세싱 유닛(220) 바로 위에 적층되는 3D DRAM(250)을 포함한다. 도시되지 않지만, SiP(200) 및 SiP(240) 각각에 대하여, 복수의 칩 또는 소자 층들이 서로 위에 적층될 수 있고, 직접적인 수직 인터커넥트(216)가 이들 간을 뚫고 연결된다. 서로 다른 소자 층들 사이를 뚫을 수 있는 수직 인터커넥트(216)의 크기 및 밀도는 3D IC 제조에 사용되는 기반 기술에 기초하여 변한다.

[0036] 이제 도 3을 살펴보면, 3차원(3D) DRAM을 이용하는 컴퓨팅 시스템(300)의 일 실시예의 개략적인 블록도가 도시된다. 앞서 설명한 회로 및 로직은 동일하게 번호가 부여된다. 컴퓨팅 시스템(300)은 앞서 설명한 시스템 인 패키지(SiP)와 같은 3차원(3D) 패키징을 이용할 수 있다. 컴퓨팅 시스템(300)은 SiP(310)를 포함할 수 있다. 일 실시예에서, SiP(310)는 앞서 설명한 프로세싱 유닛(220)과, 저-대기시간 인터커넥트(340)을 통해 통신하는 3D DRAM(330)을 포함할 수 있다. 인-패키지 저-대기시간 인터커넥트(340)는 SiP가 사용되지 않을 때 긴 오프-칩 인터커넥트보다 짧은 길이로, 수평 및/또는 수직일 수 있다.

- [0037] SiP(310)의 일 실시예가 DRAM 메모리 기술을 이용하는 것으로 도시되지만, 하나 이상의 버퍼 또는 다른 대등한 구조를 포함하는 로우-기반 액세스 기법을 이용하는 다른 메모리 기술도 가능하고 고려된다. 다른 메모리 기술의 예는 상변화 메모리, 스핀-토크-전달 저항 메모리, 멤리스터, 등을 포함한다.
- [0038] 프로세싱 유닛(220)은 소프트웨어 애플리케이션에 의해 설명되는 알고리즘을 위한 연산 및 작업을 수행하기 위한 실행 엔진(322)을 포함할 수 있다. 일 실시예에서, 실행 엔진(322)은 범용 CPU의 하나 이상의 프로세서 코어를 포함할 수 있다. 다른 실시예에서, 실행 엔진(322)은 적어도 그래픽 렌더링 및 3D 그래픽 애플리케이션을 위한 데이터-중심 작업을 수행하기 위해 셰이더, 다른 그래픽 프로세싱 블록, 그리고 GPU의 단일-명령어-멀티플-데이터(SIMD) 파이프라인을 포함할 수 있다. 주어진 알고리즘에 따른 데이터 프로세싱을 위한 회로의 다른 예도 가능하고 고려된다.
- [0039] 프로세싱 유닛(220)은 메모리 컨트롤러(326), 메모리 버스(150), 및 I/O 컨트롤러 및 버스(160)를 통해 디스크 메모리(162)와 같은 오프-패키지 메모리와 통신할 수 있다. 도시되는 바와 같이, 일 실시예에서, 컴퓨팅 시스템(300)은 앞서 설명한 컴퓨팅 시스템(100)에 포함되는, 오프-패키지 DRAM(170)을 포함할 수 없다. 온-패키지 3D DRAM(330)은 매우 큰 온-패키지 메모리 스토리지를 제공할 수 있고, 이는 오프-패키지 메모리 액세스의 수를 감소시키고, 긴 메모리 대기시간을 감춘다. 다른 실시예에서, 컴퓨팅 시스템(300)은 오프-패키지 DRAM(170)을 여전히 포함할 수 있다.
- [0040] 프로세싱 유닛(220)은 I/O 장치 및 다른 프로세싱 유닛에 대한 인터페이스 로직을 포함할 수 있다. 인터페이스 로직은 설명의 편의성을 위해 도시되지 않는다. 프로세싱 유닛(220)은 3D DRAM(330)과 통신하는 인터페이스 로직(324)을 또한 포함할 수 있다. 이러한 통신에 사용되는 프로토콜, 어드레스 포맷, 및 인터페이스 신호는 오프-패키지 DRAM(170)에 사용되는 프로토콜, 어드레스 포맷, 및 인터페이스 신호와 유사할 수 있다. 그러나, 3D DRAM(330)이 최종-레벨 캐시(LLC)로 사용될 때, 이러한 통신에 조정이 이루어질 수 있다. 예를 들어, 프로세싱 유닛(220)으로부터 3D DRAM(330)으로 전송되는 메모리 요청은 메모리 어레이 뱅크(332a-332b) 중 하나 내의 각자의 로우를 식별하는, DRAM 어드레스에 부가된, 캐시 태그를 포함할 수 있다. 수신한 캐시 태그를 이용하여, 3D DRAM(330) 내의 식별된 주어진 로우에 저장되는 캐시 태그에 비교할 수 있다.
- [0041] 다른 DRAM 토폴로지와 마찬가지로, 3D DRAM(330)은 멀티플 메모리 어레이 뱅크(332a-332b)를 포함할 수 있다. 각각의 뱅크(332a-332b)는 로우 버퍼(334a-334b) 중 각자 하나씩을 포함할 수 있다. 각각의 로우 버퍼(334a-334b)는 메모리 어레이 뱅크(332a-332b) 내 멀티플 로우 중 액세스된 로우에 데이터를 저장할 수 있다. 액세스된 로우는 수신한 메모리 요청 내 DRAM 어드레스에 의해 식별될 수 있다. 제어 로직(336)은 로우 버퍼에 저장된 하나 이상의 캐시 태그와, 수신한 메모리 요청 내 캐시 태그 사이에서 태그 비교를 수행할 수 있다. 추가적으로, 제어 로직은 수신한 DRAM 어드레스 내 비트 필드보다 캐시 태그 비교 결과를 이용함으로써 로우 버퍼의 칼럼 액세스를 변경할 수 있다.
- [0042] 제어 로직(336)은 로우 버퍼 내에 저장되는 메타데이터를 업데이트할 수 있다. 메타데이터는 적어도 최소 최근 사용(LRU) 값, 더티 비트, 및 캐시 일관성 값을 포함할 수 있다. 제어 로직(336)은 식별된 로우 및 로우 버퍼의 액세스를 동기화하여, 멀티플 DRAM 거래를 단일, 복합 거래로 변화시킬 수 있다. 이러한 단일, 복합 거래는 3D DRAM(330) 내의 데이터 및 제어 라인의 활성화 및 프리차지를 수행하여, 한번에, 식별된 로우에 액세스하고, 한번에, 로우 버퍼에 저장된 수정 콘텐츠를 식별된 로우에 풋-백(put back)할 수 있다. 추가적인 설명이 이어진다.
- [0043] 이제 도 4를 참조하면, 캐시 스토리지 배열(400)에 액세스하는데 사용되는 시퀀스 스텝의 일 실시예를 설명하는 개략적 블록도가 도시된다. 데이터는 다양한 방식으로 3D DRAM 내에 저장될 수 있다. 예를 들어, 주어진 메모리 어레이 뱅크 내의 각각의 로우는 세트-연관 캐시 조직을 이용하여 데이터를 저장할 수 있다. 저장된 데이터는 캐시 라인으로 분할될 수 있다. 각각의 로우는 동일 로우 내의 캐시 라인에 대응하는 캐시 태그를 또한 저장할 수 있다. 더욱이, 각각의 로우는 대체 상태, 캐시 일관성 상태, 더티 비트, 등과 같이, 주어진 캐시 라인에 대응하는 메타데이터를 저장할 수 있다.
- [0044] 캐시 스토리지 배열(400)의 일 실시예가 DRAM 메모리 기술의 이용을 설명하고 있으나, 하나 이상의 버퍼 또는 다른 대등 구조를 포함한 로우-기반 액세스 기법을 이용하는 다른 메모리 기술도 가능하고 고려된다. 다른 메모리 기술의 예는 상변화 메모리, 스핀-토크-전달 저항 메모리, 멤리스터, 등을 포함한다.
- [0045] 3D DRAM 내 하나 이상의 메모리 어레이 뱅크 중 주어진 하나가 도 4에 도시된다. 메모리 어레이 뱅크(430)는 멀티플 로우(432a-432k)를 포함할 수 있다. 각각의 로우(432a-432k)는 N-웨이 세트-연관 캐시 조직을 위한 캐시

태그 및 캐시 라인을 저장할 수 있고, N은 1보다 큰 정수다. 8-웨이, 16-웨이, 32-웨이, 등과 같은 서로 다른 개수의 웨이가 선택될 수 있다. 예를 들어, 로우(432a)는 필드(434a-434d)에 캐시 태그를 저장한다. 로우(432a)는 필드(438a-438d)에 대응하는 캐시 라인의 데이터 부분을 저장할 수 있다. 따라서, 메모리 어레이 뱅크(430)는 캐시 조직의 태그 어레이 및 데이터 어레이를 단일 물리 구조로 조합한다. 로우(432a) 내 필드(434a, 434d)와 같이, 각 로우의 일부분을 이용하여, 필드(438a-438d)와 같이, 로우의 나머지에 저장되는 데이터 요소에 대응하는 태그 엔트리를 홀딩할 수 있다.

[0046] 메모리 어레이 뱅크(430)의 로우(432a-432k) 내 태그 스토리지는 3D DRAM의 총 크기의 고정된 분율일 수 있다. 일 실시예에서, 캐시 라인 또는 캐시 블록의 데이터 부분은 64바이트의 데이터를 저장한다. 다른 크기도 가능하고 고려된다. 그러나, 캐시 라인의 크기는 메모리 어레이 뱅크(430)에 사용되는 캐시 스토리지 배열(400)로 인해 변하지 않을 수 있다. 캐시 스토리지 배열(400) 및 시퀀스 스텝은 64-바이트 크기와 같이, 캐시 라인 크기를 동일하게 유지시키면서, 3D DRAM의 전체 크기를 갖는 태그 스토리지 스케일을 가능하게 할 수 있다. 어떤 온-칩 SRAM 스토리지도 프로세싱 유닛 내 태그 스토리지를 위해 이용될 수 없다.

[0047] 일 실시예에서, 메모리 어레이 뱅크(430)는 30-웨이 세트-연관 캐시 조직을 이용한다. 각각의 로우(432a-432k)는 32개의 웨이를 저장하기에 충분한 공간을 가질 수 있지만, 그 중 2개는 대응하는 캐시 태그 정보를 저장하는데 이용될 수 있다. 일례에서, 캐시 라인은 64바이트이고, 캐시 태그는 4바이트다. 본 예에서, 2개의 캐시 라인은 2라인 x 64바이트 = 128 바이트의 스토리지를 제공한다. 본 예에서, 태그 스토리지는 2개의 캐시 라인에 의해 통상적으로 점유되는 공간에 배치된다. 30개의 캐시 라인의 경우, 태그 스토리지는 30웨이 x 4바이트 = 120 바이트를 이용한다. 추가의 8바이트를 이용하여, 대체 결정, 사전인출 힌트, 프로파일링 데이터, 더티 비트, 캐시 일관성 상태, 등을 개선시키기 위한 추가 정보와 같이, 추가적인 메타데이터를 저장할 수 있다.

[0048] 로우(432a-432k)의 각각의 로우는 저장되는 캐시 라인에 대응하는 메타데이터를 저장할 수 있다. 예를 들어, 로우(432a)는 필드(436a)에 상태 메타데이터를 저장한다. 저장되는 메타데이터는 각각의 대응 캐시 라인을 위해 다음 중 하나 이상 - 유효 비트, 대응하는 캐시 블록을 소유하는 소스를 표시하는 캐시 블록 오퍼 인코딩, 캐시 컨트롤러에 의해 이용되는 캐시 대체 알고리즘과 관련하여 사용되는 최소 최근 사용(LRU) 퇴거 정보, 수정, 배타, 소유, 공유, 무효, 등과 같은 캐시 일관성 상태를 지칭하는 표시, 더티 비트, 사전인출 힌트 정보, 프로파일링 데이터, 등을 적어도 포함할 수 있다. 다른 타입의 상태 정보도 가능하고 고려된다.

[0049] 메모리 어레이 뱅크(430)의 로우에 저장된 정보로 계속하여, 로우(432k)는 로우(432a)에서와 유사한 포맷으로 정보를 저장할 수 있다. 예를 들어, 로우(432k)는 필드(434m-434q)에 태그를, 필드(436k)에 상태 메타데이터를, 그리고 필드(438m-438q)에 대응하는 캐시 라인의 데이터 부분을 저장할 수 있다. 로우(432a-432k)에 저장되는 정보가 특정 순서 및 인접 공간에서 도시되지만, 인접한지 여부에 관계없이 데이터 저장을 위한 다른 배치도 가능하고 고려된다. 특정 스토리지 배열이 설계 교환에 기초하여 선택될 수 있다.

[0050] 로우 버퍼(440)는 로우(432a-432k) 중 선택된 로우로부터 관독되는 데이터 및 메타데이터를 저장할 수 있다. 로우 버퍼(440)는 로우(432a-432k)와 유사한 스토리지 배열을 이용할 수 있다. 예를 들어, 로우 버퍼(440)는 필드(444a-444d) 내 N 웨이의 세트-연관 캐시 조직을 위한 캐시 태그 정보를 저장할 수 있다. 로우 버퍼(440)는 필드(448a-448d) 내 대응하는 캐시 라인의 데이터 부분을 저장할 수 있다. 로우 버퍼(440)는 필드(446) 내, 대응하는 상태 메타데이터(state metadata)와 같은, 상태 정보(status information)를 저장할 수 있다.

[0051] 일례에서, 캐시 라인은 64 바이트이고, 캐시 태그는 4바이트이며, 메모리 어레이 뱅크(430)는 30-웨이 세트-연관 캐시 조직을 이용한다. 각각의 로우(432a-432k)는 태그 스토리지를 위해 2 웨이, 그리고 데이터 스토리지를 위해 30 웨이를 이용할 수 있다. 2웨이는 메타데이터 스토리지를 또한 포함할 수 있다. 따라서, 각각의 로우(432a-432k)는 32 웨이 x 64 바이트/웨이 = 2,048 바이트 또는 2 키로-바이트(KB)의 스토리지를 포함한다. 마찬가지로, 로우 버퍼(440)는 2KB의 스토리지를 포함한다. 각각의 로우(432a-432k)는 정보 저장을 위해 DRAM 메모리 기술을 이용할 수 있다. 그러나, 로우 버퍼(440)는 레지스터, SRAM 셀, 또는 다른 저장 메커니즘을 이용할 수 있다. 따라서, 로우 버퍼(440)의 반복적 액세스는 로우(432a-432k) 중어떤 하나의 반복된 액세스보다 더 효율적일 수 있다.

[0052] 3D DRAM은 대응하는 프로세싱 유닛에 대한 초대형 인-패키지 메모리 스토리지를 저장할 수 있다. 이러한 대형 메모리 스토리지는 최종-레벨 캐시로 사용될 수 있다. 일 실시예에서, 캐시 라인 또는 캐시 블록의 데이터 부분은 64바이트의 데이터를 저장한다. 다른 크기도 가능하고 고려된다. 3D DRAM은 적어도 수백 메가바이트(MB) 또는 그 이상의 크기를 가질 수 있다. 메모리 어레이 뱅크(430)는 3D DRAM 내에 멀티플 뱅크 중 하나의 뱅크일 수 있다. 일례에서, 3D DRAM은 512MB의 크기를 갖고, 이는 8,388,608 개의 개별 64-바이트 캐시 라인을 저장할 수

있다. 대응하는 캐시 태그는 4 내지 8 바이트의 크기를 가질 수 있다. 4바이트 캐시 태그의 경우, 대응하는 태그 어레이는 $8,388,608 \times 4$ 바이트 = 32 MB의 크기를 갖는다. 이러한 대형 태그 어레이는 프로세싱 유닛 내 온-칩 SRAM에 쓸모있게 저장되지 않을 수 있다. 대신에, 이러한 대형 태그 어레이가, 도 4에 도시되는 바와 같이, 3D DRAM 내에 저장될 수 있다.

[0053] 3D DRAM에 저장되는 캐시 라인에 대응하는 태그, 상태 정보, 및 데이터에 액세스하기 위해 스텝(1-7)의 시퀀스가 도 4에 도시된다. 메모리 어레이 뱅크(430)가 동일 로우 내의 태그 어레이 및 데이터 어레이를 저장하는 캐시로 사용될 때, 로우(432a-432k) 중 주어진 로우에 대한 스텝(1-7)을 이용하는 시퀀스와는 다른 액세스 시퀀스는 큰 대기시간을 가질 수 있다. 예를 들어, DRAM 액세스는 통상적으로, 제 1 활성화 또는 오픈 스테이지, 전체 로우의 콘텐츠를 로우 버퍼 내로 복사하는 스테이지, 태그 판독 스테이지, 태그 비교 스테이지, 칼럼 액세스를 포함하는 데이터 판독 또는 기록 액세스 스테이지, 제 1 사전인출 또는 클로즈 스테이지, 제 2 활성화 또는 오픈 스테이지, 전체 로우의 콘텐츠를 다시 로우 버퍼 내로 복사하는 스테이지, 태그 판독 스테이지, 태그 비교 스테이지, 매칭 태그에 대응하는 상태 정보를 위한 업데이트 스테이지, 및 제 2 사전인출 또는 클로즈 스테이지를 포함한다.

[0054] 메모리 어레이 뱅크(430) 내의 액세스 스텝을 계속하면, 다른 로우에 저장되는 다른 데이터가 그 동안에 액세스될 경우 로우 버퍼의 각각의 액세스 이후에 하나 이상의 추가적인 사전인출 및 활성화 스테이지가 포함될 수 있다. 단일 캐시 액세스를 위해 복수의 DRAM 거래를 이용하는 대신에, 스텝(1-7)의 시퀀스를 이용하여 캐시 액세스를 단일 DRAM 거래로 변환할 수 있다. 활성화/오픈, 칼럼 액세스, 판독, 기록, 및 사전인출/클로즈와 같은, 서로 다른 DRAM 작동들 각각은 각자 서로 다른 대기시간을 갖는다. 활성화 및 사전인출 작동은 판독 및 기록 작동보다 훨씬 큰 대기시간을 가지는 경우가 잦다. 따라서, 단일, 복합 거래를 생성하는데 사용되는 스텝(1-7)의 시퀀스는 개별 활성화 및 사전인출 작동의 수를 감소시킬 수 있다.

[0055] 시퀀스(1) 중, 프로세싱 유닛으로부터의 메모리 요청이 3D DRAM에 의해 수신될 수 있다. 메모리 요청은 3D 집적 제조 프로세스를 통해 가용한 수평 또는 수직의 짧은 저-대기시간 인터커넥트 루트를 가로질러왔을 수 있다. 완전한 어드레스의 일부분이 어드레스(410)로 도시된다. 필드(412, 414)는 각각 캐시 태그 및 페이지 인덱스를 저장할 수 있다. 완전한 어드레스의 다른 부분은 채널 인덱스, 뱅크 인덱스, 서브 어레이 인덱스, 등 중 하나 이상을 포함하여, 3D DRAM 내의 메모리 어레이 뱅크(430)를 식별할 수 있다. 시퀀스(2) 중, 로우(432a-432k) 중 주어진 로우가 페이지 인덱스(414)에 의해 다른 로우로부터 선택될 수 있다.

[0056] 시퀀스(3) 중, 활성화 또는 오픈 스테이지가 실행되어 로우(432a-432k) 중 선택된 로우의 전체 콘텐츠를 로우 버퍼(440) 내로 복사한다. 3D DRAM(또는 일반적으로 DRAM) 내 로우(432a-432k) 중 선택된 로우를 판독하는 것은, "소멸" 작동이다. 선택된 로우의 전체 콘텐츠의 복제본이 로우 버퍼(440)에서 유지된다. 메모리 어레이 뱅크(430)의 로우로부터 로우 버퍼 내로 데이터를 판독하는 이러한 프로세스는 "활성화" 또는 "오픈"이라 불린다. 서로 다른 로우로부터의 데이터에 액세스하여야 할 때, 로우 버퍼(440)의 콘텐츠가 "사전인출"되거나 "클로즈"될 수 있고, 이는 로우 버퍼(440)에 저장된 데이터의 현재 값을 메모리 어레이 뱅크(430) 내로 배치한다.

[0057] 캐시 태그를 이용하여, 멀티플 캐시 라인 중 어느 캐시 라인이 선택된 로우 내에서 액세스되고 있는지를 결정할 수 있다. 예를 들어, 30-웨이 세트-연관 캐시 조직에서, 로우(432a)가 선택될 때, 필드(434a-434d)에 저장되는 캐시 태그 값을 이용하여, 필드(438a-438d)에 저장된 30 캐시 라인 중 어느 캐시 라인에 액세스되고 있는지를 결정할 수 있다. 어드레스(410) 내 필드(412)에 저장되는 캐시 태그는 로우 버퍼(440)에 저장되는 멀티플 캐시 라인 중 대응하는 캐시 라인의 위치를 파악하기 위해 비교 로직에 사용될 수 있다.

[0058] 시퀀스(4) 중, 필드(412)에 저장된 캐시 태그 값이 로우 버퍼(440) 내 필드(444a-444d) 중 하나에 저장된 캐시 태그 값과 일치하고 필드(446)에 저장된 대응하는 캐시 라인 상태가 대응 캐시 라인이 유효 캐시 라인임을 표시할 때, 캐시 라인 히트가 발생할 수 있다. 일 실시예에서, 이러한 태그 비교 작동은 단일 연속 버스트에서 필드(444a-444d)에 저장된 캐시 태그 각각을 판독할 수 있다. 일례에서, 캐시 태그는 앞서 예에서 설명한 바와 같이 2 캐시 라인과 동등한 크기의 공간에 저장된다. 이러한 2 캐시 라인 가치의 정보가 단일 작동에서 로우 버퍼(440)로부터 판독될 수 있다. 다음에, 비교 로직이 필드(444a-444d)로부터 판독된 캐시 태그 각각을 필드(412)에 저장된 캐시 태그 값에 비교할 수 있다.

[0059] 다른 실시예에서, 수신한 어드레스로부터 하나 이상의 비트가, 태그 비교 로직을 위해 필드(444a-444d)에 저장된 캐시 태그의 서브그룹을 선택할 수 있다. 태그 매치가 결정될 경우, 3D DRAM 액세스의 후속 단계가 진행될 수 있다. 매치가 발생한다고 결정되지 않을 경우, 필드(444a-444d)에 저장된 캐시 태그의 제 1 서브그룹과는 다른 서브그룹이 태그 비교 로직을 위해 선택될 수 있다. 이러한 메커니즘이 3개 이상의 서브그룹으로 일반화될

수 있다. 이러한 메커니즘은 태그 비교에 사용되는 판독 작동의 총 크기 및/또는 총 집합 버스트 길이 및/또는 버스 점유도를 감소시킬 수 있다.

- [0060] 시퀀스(5) 중, 로우 버퍼(440)에 저장되는 멀티플 캐시 라인 중 주어진 캐시 라인이 태그 비교 결과에 기초하여 선택된다. 이러한 칼럼 액세스는 필드(444a-444d)의 캐시 태그와, 필드(446)에 저장되는 캐시 라인 상태 정보와 같이, 수신한 어드레스에 저장되는 그리고 로우 버퍼(440)에 저장되는 정보에 기초한다. 선택된 주어진 캐시 라인은 수신한 메모리 요청에 기초하여 판독 또는 기록된다. 일 실시예에서, 오프셋 값은 수신한 어드레스에 저장될 수 있고, 액세스될 선택된 캐시 라인 내의 특정 바이트 또는 워드를 표시하는데 사용될 수 있다. 판독 또는 기록 작동은 로우 버퍼(440)에 저장된 콘텐츠에 바로 작용할 수 있다.
- [0061] 시퀀스(6) 중, 선택된 캐시 라인에 대응하는 필드(446)에 저장된 상태 정보가 업데이트된다. 예를 들어, 캐시 일관성 상태가 변경되었을 수 있고, LRU 대체 상태가 변경되었을 수 있으며, 캐시 블록 오너 식별자(ID)가 변경되었을 수 있고, 사전인출 힌트 정보 및 프로파일링 데이터가 변경되었을 수 있으며, 등등이다. 다시, 업데이트 작동은 로우 버퍼(440)에 저장된 콘텐츠에 바로 작용할 수 있다. 시퀀스(7) 중, 사전인출 또는 클로즈 스테이지가 실행되어, 로우 버퍼(440)의 전체 콘텐츠를 다시 로우(432a-432k) 중 선택된 로우 내로 복사한다. 적어도 캐시 라인들 중 주어진 캐시 라인과 상태 정보와 같이, 로우 버퍼(440) 내 콘텐츠 중 일부가 이제 수정된다.
- [0062] 도 3에 도시되는 제어 로직(336)과 같은 제어 로직은, 시퀀스(3-7)에서 수행되는 작동들이 중단없는 작동의 블록으로 발생함을 보장할 수 있다. 예를 들어, 시퀀스(3)에서 오픈 작동이 개시되면, 메모리 어레이 뱅크(430) 내 데이터에 작용하기를 기다리는 다른 메모리 요청이, 시퀀스(3-7)가 완성될 때까지 제어 로직에 의해 계속 대기하도록 지시받을 수 있다. 이러한 3D DRAM의 단일, 복합 액세스는 추가적인 고비용 활성화 및 사전인출 대기 시간을 피한다. 3D DRAM의 이러한 단일, 복합 액세스는, 단일 활성화 대기시간, 단일 사전인출 대기시간, 캐시 라인 상에서 수행되는 단일 판독/기록 작동, 상태 정보 업데이트를 위한 단일 기록 작동, 및 태그 판독 및 비교를 위한 대기시간에 대한 단일 메모리 요청에 대응하는 액세스의 대기시간을 유지시킬 수 있다.
- [0063] 이제 도 5를 살펴보면, 효율적인 인-패키지 DRAM 액세스를 위한 방법(500)의 일 실시예의 개략적인 순서도가 도시된다. 설명을 위해, 본 실시예 및 나중에 설명되는 방법의 후속 실시예의 단계들이 순차적으로 도시된다. 그러나 다른 실시예에서, 일부 단계는 도시된 것과는 다른 순서로 이루어질 수 있고, 일부 단계는 동시에 수행될 수 있으며, 일부 단계는 다른 단계와 조합될 수 있고, 일부 단계는 생략될 수 있다.
- [0064] 하나 이상의 컴퓨터 프로그램 또는 소프트웨어 애플리케이션이 블록(502)에서 실행된다. 프로세싱 유닛은 이러한 애플리케이션들을 실행할 수 있다. 프로세싱 유닛의 예는 앞서 제시된 바 있다. 프로세싱 유닛은 3D DRAM과 같은, 3D 집적 메모리를 또한 포함하는 시스템-인-패키지(SiP) 내에 놓일 수 있다. 프로세싱 유닛은 캐시로 3D DRAM을 이용할 수 있다.
- [0065] 블록(504)에서, 프로세싱 유닛은 프로세싱 유닛 내의 캐시 메모리 서브시스템 내의 주어진 메모리 요청 미스를 결정할 수 있다. 블록(506)에서, 프로세싱 유닛은 3D DRAM과 같이 인-패키지 집적 DRAM 캐시에 주어진 메모리 요청에 대응하는 어드레스를 전송할 수 있다. 어드레스는 온-칩 캐시에 액세스하기 위해 프로세싱 유닛 내에 사용되는 대응하는 캐시 어드레스로부터 변환되는 DRAM 어드레스에 추가하여 비-변환 캐시 태그를 포함할 수 있다. 블록(508)에서, 3D DRAM 내의 제어 로직은 3D DRAM 내 메모리 어레이 뱅크 내의 어드레스에 대응하는 주어진 로우를 식별할 수 있다.
- [0066] 블록(510)에서, 3D DRAM 내의 제어 로직은 주어진 로우를 활성화시키고 오픈시킬 수 있다. 블록(512)에서, 주어진 로우의 콘텐츠는 복사되어 로우 버퍼에 저장될 수 있다. 블록(514)에서, 로우 버퍼 내의 태그 정보가 어드레스의 태그 정보와 비교될 수 있다. 블록(506-512)에서 설명되는 단계들은 도 4와 관련하여 앞서 설명한 시퀀스(1-4)에 대응할 수 있다.
- [0067] 태그 비교가 태그 히트가 발생하지 않는다고 결정할 경우(조건 블록(516)), 블록(518)에서, 메모리 요청이 메인 메모리에 전송될 수 있다. 메인 메모리는 오프-칩 비-집적 DRAM 및/또는 오프-칩 디스크 메모리를 포함할 수 있다. 태그 비교가 태그 히트 발생을 결정할 경우(조건 블록(516)), 블록(520)에서, 로우 버퍼 내 대응하는 캐시 라인 상에서 판독 또는 기록 작동이 수행된다.
- [0068] 블록(522)에서, 캐시 라인을 위한 상태 정보를 저장하는 로우 버퍼 내 대응하는 메타데이터가 업데이트된다. 블록(524)에서, 로우 버퍼에 저장된 수정 및 비수정 정보 모두가 다시 주어진 로우에 복사된다. 사전인출 및 클로즈 스테이지가 수행되어 데이터를 이동시킨다.
- [0069] 상술한 실시예는 소프트웨어를 포함할 수 있다. 이러한 실시예에서, 방법 및/또는 메커니즘을 구현하는 컴퓨터

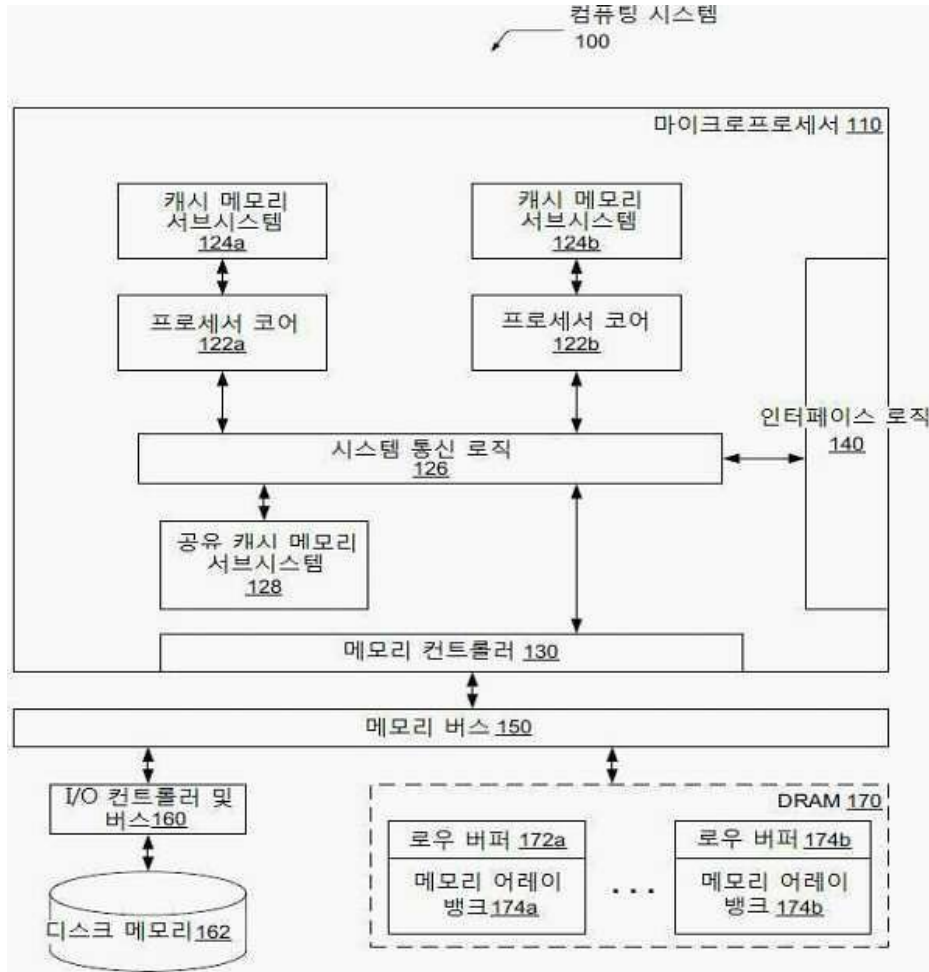
관독가능 매체 상에서 저장되거나 운반될 수 있다. 프로그램 명령어를 저장하도록 구성되는 수많은 타입의 매체가 가용하고, 하드 디스크, 플래시 디스크, CD-ROM, DVD, 플래시 메모리, 프로그래머블 ROM(PROM), 랜덤 액세스 메모리(RAM), 및 다양한 다른 형태의 휘발성 또는 비-휘발성 스토리지를 포함한다. 일반적으로 말해서, 컴퓨터 액세스가능 스토리지 매체는 컴퓨터에 명령어 및/또는 데이터를 제공하기 위해 사용 중 컴퓨터에 의해 액세스가능한 임의의 스토리지 매체를 포함할 수 있다. 예를 들어, 컴퓨터 액세스가능 스토리지 매체는 자기 또는 광학 매체와 같은 스토리지 매체, 예를 들어, 디스크(고정식 또는 제거가능형), 테이프, CD-ROM, 또는 DVD-ROM, CD-R, CD-RW, DVD-R, DVD-RW, 또는 블루-레이를 포함할 수 있다. 스토리지 매체는 RAM(가령, 동기식 동적 RAM(SDRAM), 더블 데이터 레이트(DDR, DDR2, DDR3, 등) SDRAM, 저-전력 DDR(LPDDR2, 등) SDRAM, 램버스 DRAM(RDRAM), 정적 RAM(SRAM), 등), ROM, 플래시 메모리, 비-휘발성 메모리(가령, 플래시 메모리), 범용 시리얼 버스(USB) 인터페이스, 등과 같은 주변 인터페이스를 통해 액세스가능)와 같은 휘발성 또는 비-휘발성 메모리 매체를 더 포함할 수 있다. 스토리지 매체는 네트워크 및/또는 무선 링크와 같은 통신 매체를 통해 액세스가능한 스토리지 매체와, 마이크로일렉트로메카니컬 시스템(MEMS)을 포함할 수 있다.

[0070] 추가적으로, 프로그램 명령어는 GDS II 스트림 포맷(GDSII)와 같은 데이터베이스 포맷 또는 VHDL, Verilog와 같은 디자인 언어(HDL), 또는 C와 같은 하이 레벨 프로그래밍 언어로 하드웨어 기능의 하이 레벨 레지스터-트랜스퍼 레벨(RTL) 디스크립션 또는 거동-레벨 디스크립션을 포함할 수 있다. 일부 경우에, 디스크립션(description)은 합성 라이브러리로부터 게이트들의 리스트를 포함하는 네트리스트를 제공하기 위해 디스크립션을 합성할 수 있는 합성 틀에 의해 관독될 수 있다. 네트리스트는, 시스템을 포함하는 하드웨어의 기능을 또한 나타내는, 한 세트의 게이트를 포함한다. 네트리스트는 그 후 마스크에 적용될 기하적 형상을 설명하는 데이터 세트를 생성하도록 배치 및 루팅될 수 있다. 마스크는 그 후 시스템에 대응하는 회로 또는 반도체 회로를 생성하도록 다양한 반도체 제조 스텝에 사용될 수 있다. 대안으로서, 컴퓨터 액세스가능 스토리지 매체 상의 명령어는 요망되는 바에 따라, (합성 라이브러리가 있는 또는 없는) 네트리스트, 또는 데이터 세트일 수 있다. 추가적으로, 명령어는 Cadence®, EVE®, 및 Mentor Graphics®와 같은 판매사로부터 하드웨어 기반 타입 에뮬레이터에 의해 에뮬레이션 용도로 이용될 수 있다.

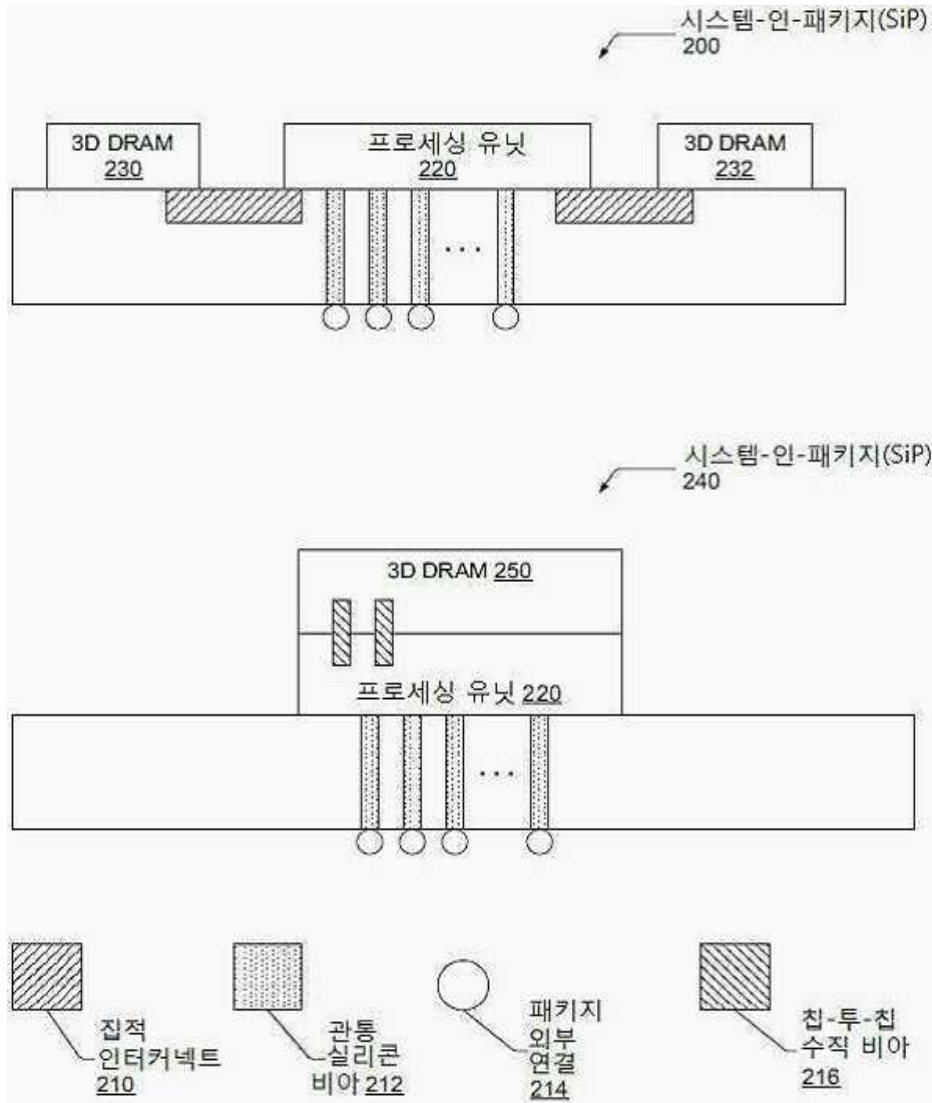
[0071] 위 실시예가 상당히 세부적으로 설명되었으나, 위 개시내용을 완전히 이해할 경우 수많은 변형예 및 수정예가 다 업자에게 명백해질 것이다. 다음의 청구범위는 이러한 모든 변형예 및 수정예를 포괄하는 것으로 해석되어야 한다.

도면

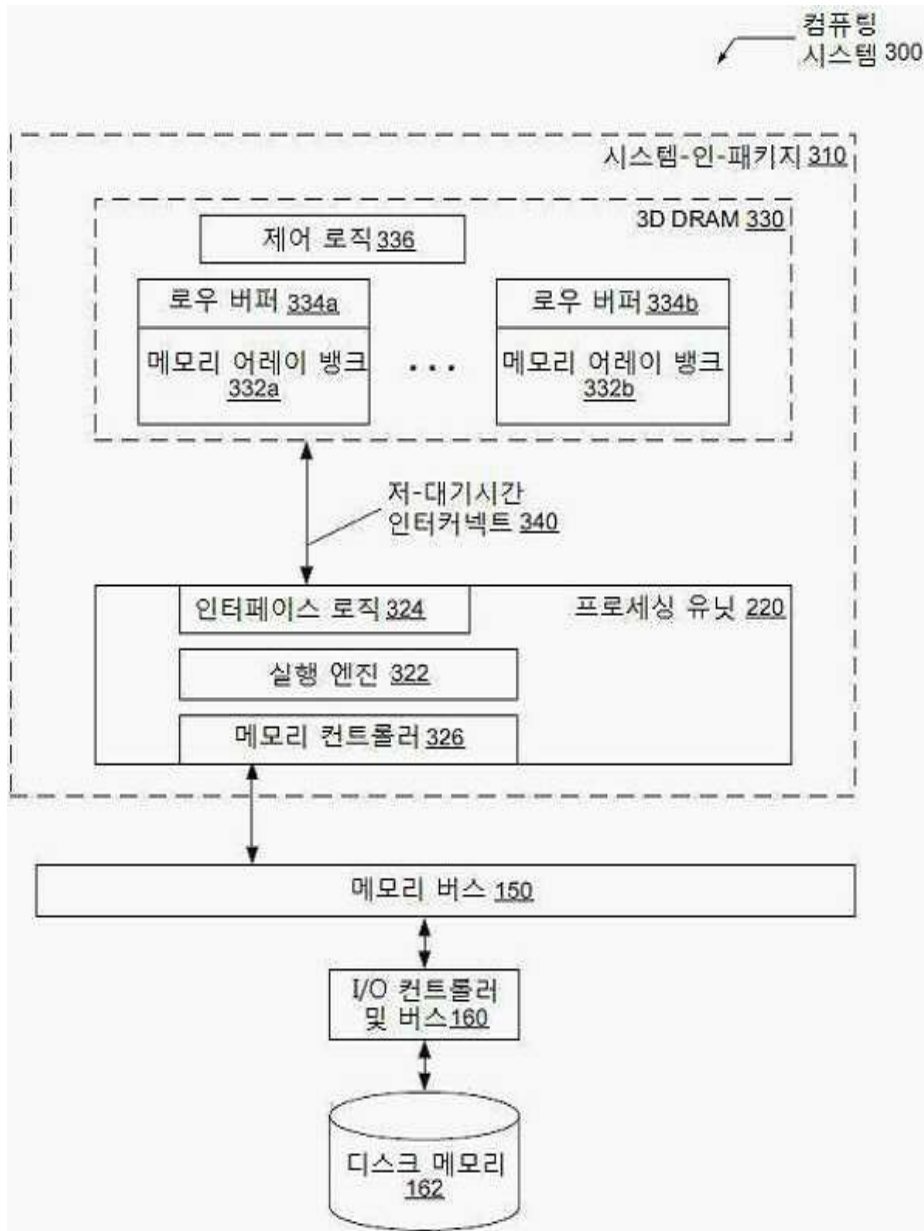
도면1



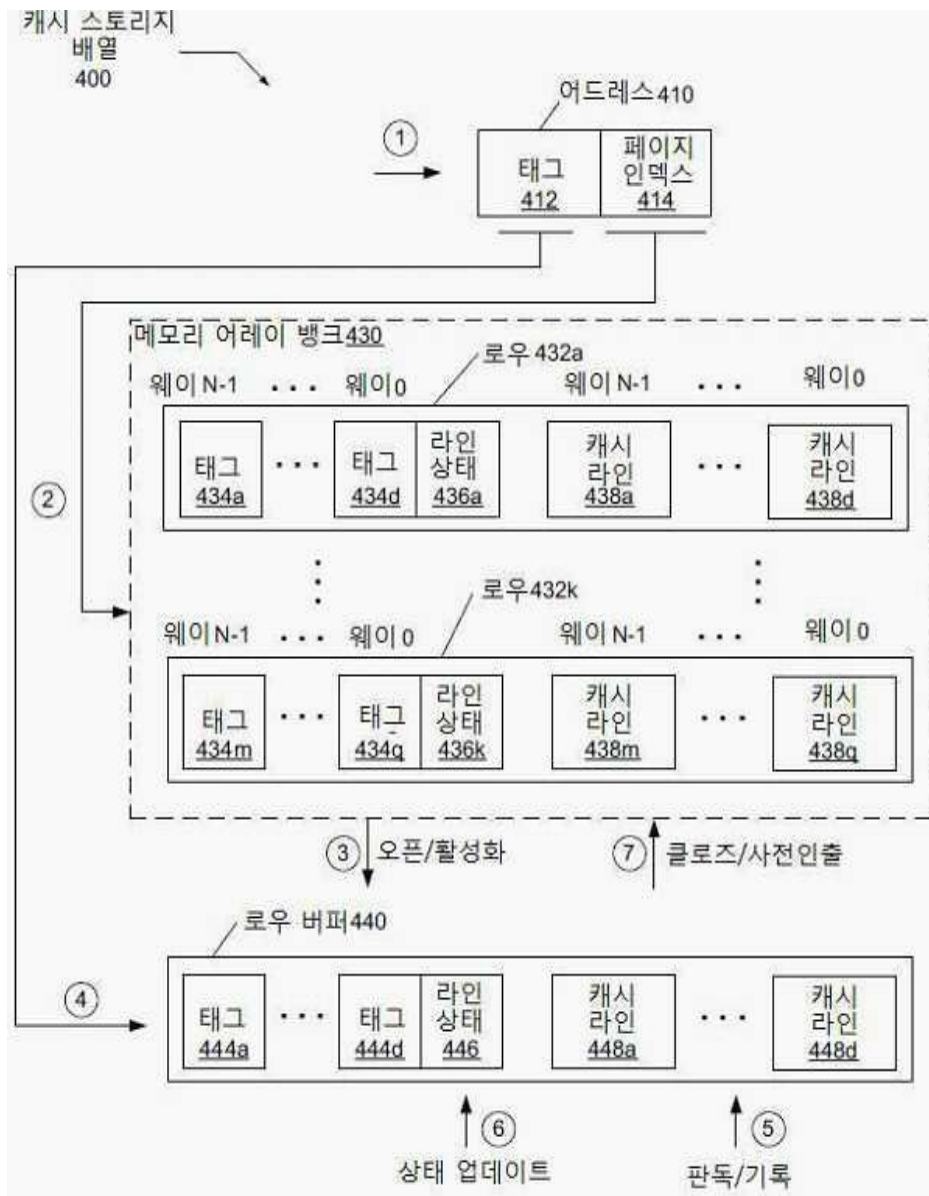
도면2



도면3



도면4



도면5

