



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0139610  
(43) 공개일자 2014년12월05일

(51) 국제특허분류(Int. Cl.)  
**G06F 12/08** (2006.01)  
(21) 출원번호 10-2014-7030486  
(22) 출원일자(국제) 2013년04월01일  
심사청구일자 **없음**  
(85) 번역문제출일자 2014년10월29일  
(86) 국제출원번호 PCT/US2013/034847  
(87) 국제공개번호 WO 2013/149254  
국제공개일자 2013년10월03일  
(30) 우선권주장  
13/435,539 2012년03월30일 미국(US)

(71) 출원인  
어드밴스드 마이크로 디바이스, 인코포레이티드  
미국 캘리포니아 94088-3453 서니베일 피.오.박스  
3453 원 에이엠디 플레이스  
(72) 발명자  
맨 스릴라타  
미국 오리건 97212 포트랜드 노스이스트 12번 에  
비뉴 2115  
벌철 윌리엄 엘.  
미국 텍사스 78751 어스틴 에비뉴 에프 3913  
(뒷면에 계속)  
(74) 대리인  
박장원

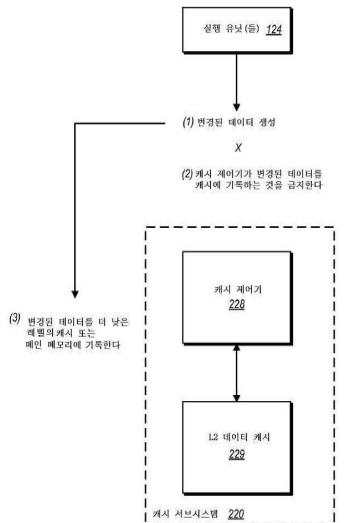
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 고속 캐시를 셋다운시키는 장치 및 방법

### (57) 요 약

고속 캐시를 셋다운시킬 수 있는 장치 및 방법이 개시된다. 일 실시예에서, 캐시 서브시스템은 캐시 메모리 및 이 캐시 메모리에 연결된 캐시 제어기를 포함한다. 캐시 제어기는, 캐시 서브시스템에 전력을 복구할 때, 변경된 데이터를 캐시 메모리에 배타적으로 기록하는 것을 금지하도록 구성된다.

**대 표 도** - 도5



(72) 발명자

고빈단 매드후 살바나 시비

미국 텍사스 78759 어스틴 아파트먼트 308 그레이  
트 헐스 트레일 10050

오코널 제임스 앤.

미국 텍사스 78717 어스틴 메디나 그린스 드라이브  
10520

셜트 미첼 제이.

미국 텍사스 78739 어스틴 미트라 드라이브 6517

## 특허청구의 범위

### 청구항 1

캐시 서브시스템으로서,

캐시 메모리에 연결하기 위한 캐시 제어기를 포함하되,

상기 캐시 제어기는, 상기 캐시 서브시스템에 전력을 복구한 것에 응답하여, 변경된 데이터를 상기 캐시 메모리에 배타적으로 기록하는 것을 금지하도록 구성된 것인 캐시 서브시스템.

### 청구항 2

제1항에 있어서, 상기 캐시 제어기는, 상기 변경된 데이터가 상기 캐시 메모리보다 더 낮은 메모리 계층에서 적어도 하나의 추가적인 위치에 기록된 경우에도, 전력을 복구한 것에 후속하여 변경된 데이터가 상기 캐시 메모리에 기록되도록 구성된 것인 캐시 서브시스템.

### 청구항 3

제2항에 있어서, 상기 캐시 제어기는, 상기 변경된 데이터가 더 낮은 레벨의 캐시에 기록된 경우에도, 전력을 복구한 것에 후속하여 변경된 데이터가 상기 캐시 메모리에 기록되도록 구성된 것인 캐시 서브시스템.

### 청구항 4

제2항에 있어서, 상기 캐시 제어기는, 상기 변경된 데이터가 메인 메모리에 기록된 경우에도, 전력을 복구한 것에 후속하여 변경된 데이터가 상기 캐시 메모리에 기록되도록 구성된 것인 캐시 서브시스템.

### 청구항 5

제1항에 있어서, 상기 캐시 제어기는, 변경된 데이터가 상기 캐시 메모리에 기록되는 것을 금지하도록 구성되고, 상기 캐시 메모리보다 더 낮은 메모리 계층에서 적어도 하나의 추가적인 위치에 변경된 데이터가 기록되도록 더 구성된 것인 캐시 서브시스템.

### 청구항 6

제5항에 있어서, 상기 캐시 제어기는 변경된 데이터가 상기 메모리 계층에서 더 낮은 레벨의 캐시에 기록되도록 구성된 것인 캐시 서브시스템.

### 청구항 7

제5항에 있어서, 상기 캐시 제어기는 변경된 데이터가 메인 메모리에 기록되도록 구성된 것인 캐시 서브시스템.

### 청구항 8

제1항에 있어서, 상기 캐시 제어기는 임계값에 도달될 때까지 변경된 데이터가 상기 캐시 메모리에 배타적으로 기록되는 것을 금지하도록 구성되고, 상기 캐시 제어기는 상기 임계값에 도달된 것에 후속하여 변경된 데이터가 상기 캐시 메모리에 배타적으로 기록되도록 더 구성된 것인 캐시 서브시스템.

### 청구항 9

제8항에 있어서, 상기 임계값은 이벤트의 개수인 것인 캐시 서브시스템.

### 청구항 10

제9항에 있어서, 상기 이벤트는 변경된 데이터를 메모리 계층에 있는 적어도 하나의 저장 유닛에 기록하는 인스턴스(instance)인 것인 캐시 서브시스템.

### 청구항 11

제8항에 있어서, 상기 임계값은 상기 캐시 서브시스템에 전력이 복구된 시간 기간인 것인 캐시 서브시스템.

### 청구항 12

방법으로서,

캐시 서브시스템에 전력을 복구하는 단계; 및

상기 캐시 서브시스템에 전력을 복구한 것에 응답하여 변경된 데이터가 상기 캐시 메모리에 배타적으로 기록되는 것을 금지하는 단계를 포함하는 방법.

### 청구항 13

제12항에 있어서, 상기 금지하는 단계는 캐시 제어기에 의해 수행되고, 상기 방법은,

임계값에 도달되기 전에 상기 변경된 데이터가 상기 캐시 메모리에 배타적으로 기록되는 것을 금지하는 단계를 상기 캐시 제어기에 의해 수행하는 단계; 및

상기 캐시 제어기가 상기 임계값에 도달된 것에 후속하여 변경된 데이터를 상기 캐시 메모리에 배타적으로 기록될 수 있게 하는 단계를 더 포함하는 방법.

### 청구항 14

제13항에 있어서, 상기 임계값은 이벤트의 미리 결정된 개수인 것인 방법.

### 청구항 15

제14항에 있어서, 상기 이벤트는 변경된 데이터를 메모리 계층에 있는 적어도 하나의 저장 유닛에 기록하는 인스턴스인 것인 방법.

### 청구항 16

제13항에 있어서, 상기 임계값은 상기 캐시 서브시스템에 전력이 복구된 시간 기간인 것인 방법.

### 청구항 17

제13항에 있어서, 상기 캐시 서브시스템에 전력을 복구한 것과 상기 임계값에 도달한 것 사이의 기간 동안 상기 캐시 메모리에 그리고 더 낮은 레벨의 캐시 메모리와 메인 메모리 중 적어도 하나에 변경된 데이터를 기록하는 단계를 더 포함하는 방법.

### 청구항 18

제13항에 있어서, 변경된 데이터가 상기 캐시 메모리에 기록되는 것을 금지하면서 상기 캐시 메모리보다 낮은 메모리 계층에 있는 적어도 하나의 추가적인 위치에 변경된 데이터를 기록하는 단계를 더 포함하는 방법.

### 청구항 19

제18항에 있어서, 상기 적어도 하나의 추가적인 위치는 더 낮은 레벨의 캐시 메모리에 있는 것인 방법.

### 청구항 20

제18항에 있어서, 상기 적어도 하나의 추가적인 위치는 메인 메모리에 있는 것인 방법.

### 청구항 21

제13항에 있어서, 상기 프로세서 코어가 상기 임계값에 도달하기 전에 아이들 상태가 된 것에 응답하여 상기 캐시 서브시스템을 포함하는 프로세서 코어로부터 전력을 제거하는 단계를 더 포함하는 방법.

### 청구항 22

시스템으로서,

적어도 하나의 프로세서 코어를 구비하는 프로세서를 포함하되, 상기 적어도 하나의 프로세서 코어는 캐시 서브

시스템을 포함하고, 상기 캐시 서브시스템은,

제1 캐시 메모리; 및

상기 제1 캐시 메모리에 연결된 캐시 제어기를 포함하며, 상기 제1 캐시 제어기는, 상기 제1 프로세서 코어에 전력을 복구할 때, 변경된 데이터를 상기 제1 캐시 메모리에 배타적으로 기록하는 것을 금지하도록 구성된 것인 시스템.

#### 청구항 23

제22항에 있어서, 상기 프로세서는 메모리 계층에서 상기 제1 캐시 메모리보다 더 낮은 제2 캐시 메모리를 더 포함하며, 상기 시스템은 상기 프로세서에 연결된 메인 메모리를 포함하고, 상기 메인 메모리는 상기 메모리 계층에서 상기 제2 캐시 메모리보다 더 낮은 것인 시스템.

#### 청구항 24

제23항에 있어서, 상기 캐시 제어기는, 변경된 데이터의 블록이 상기 제2 캐시 메모리와 상기 메인 메모리 중 적어도 하나에 기록된 경우에도, 변경된 데이터의 블록이 상기 제1 캐시 메모리에 기록될 수 있도록 구성된 것인 시스템.

#### 청구항 25

제23항에 있어서, 상기 적어도 하나의 프로세서 코어가 변경된 데이터의 블록을 생성한 것에 응답하여, 상기 캐시 제어기는 변경된 데이터의 블록이 상기 제1 캐시 메모리에 기록되는 것을 금지하도록 구성되고, 상기 프로세서 코어는 상기 변경된 데이터의 블록이 상기 제2 캐시 메모리와 상기 메인 메모리 중 적어도 하나에 기록되도록 구성된 것인 시스템.

#### 청구항 26

제22항에 있어서, 상기 제1 제어기는, 임계값에 도달된 경우, 변경된 데이터를 상기 제1 캐시 메모리에 배타적으로 기록하는 것을 금지하는 것을 중단시키도록 구성된 것인 시스템.

#### 청구항 27

제26항에 있어서, 전력 관리 유닛을 더 포함하되, 상기 전력 관리 유닛은 상기 임계값에 도달하기 전에 상기 적어도 하나의 프로세서 코어가 아이들 상태가 된 것으로 결정난 것에 응답하여 상기 적어도 하나의 프로세서 코어로부터 전력을 제거하도록 구성된 것인 시스템.

#### 청구항 28

컴퓨터 시스템에서 실행가능한 프로그램에 의해 동작되는 데이터 구조를 포함하는 비일시적인 컴퓨터 관독가능한 매체로서, 상기 프로그램은 상기 데이터 구조에 작용하여 상기 데이터 구조에 의해 설명된 회로를 포함하는 집적 회로를 제조하는 공정 부분을 수행하고, 상기 데이터 구조에 설명된 회로는,

캐시 메모리에 연결된 캐시 제어기를 포함하며, 상기 캐시 제어기는, 상기 캐시 서브시스템에 전력을 복구할 때, 변경된 데이터를 상기 캐시 메모리에 배타적으로 기록하는 것을 금지하도록 구성된 것인 컴퓨터 관독가능한 매체.

#### 청구항 29

제28항에 있어서, 상기 데이터 구조에 의해 설명된 상기 캐시 제어기는 임계값에 도달된 것에 응답하여 변경된 데이터를 상기 캐시 메모리에 배타적으로 기록하는 것을 금지하는 것을 중단시키도록 구성된 것인 컴퓨터 관독가능한 매체.

#### 청구항 30

제28항에 있어서, 상기 데이터 구조는 다음 데이터 유형 중 하나 이상을 포함하는 것인 컴퓨터 관독가능한 매체:

HDL(high-level design language) 데이터;

RTL(register transfer level) 데이터;

그래픽 데이터 시스템(Graphic Data System: GDS) II 데이터.

## 명세서

### 기술분야

[0001] 본 발명은 집적 회로에 관한 것으로, 보다 상세하게는, 프로세서에 있는 캐시 서브시스템에 관한 것이다.

### 배경기술

[0002] 집적 회로 기술이 발전함에 따라, 트랜지스터의 특징부 사이즈가 계속 감소하고 있다. 이에 의해 더 많은 회로들이 단일 집적 회로에 다이에 구현될 수 있게 되었다. 이는 집적 회로에 더 많은 기능을 구현할 수 있게 하였다. 다수의 코어를 구비하는 프로세서는 집적 회로에 구현될 수 있는 기능이 증가한 하나의 예이다.

[0003] 다수의 코어를 구비하는 프로세서의 동작 동안, 적어도 하나의 코어가 비활성(inactive)일 경우들이 있을 수 있다. 이러한 경우에, 비활성인 프로세서 코어는 전력 다운(power down)되어 전체 전력 소비를 감소시킬 수 있다. 아이들(idle) 상태의 프로세서 코어를 전력 다운시키는 것은 캐시를 포함하는, 내부에 구현된 각종 서브시스템을 전력 다운시키는 것을 포함할 수 있다. 일부 경우에, 캐시는 프로세서 코어가 전력 다운되는 것으로 결정된 시간에 변경된 데이터를 저장할 수 있다. 변경된 데이터가 프로세서 코어에 있는 캐시에 고유한 것이라면, 이 데이터는 더 낮은 레벨의 캐시에 (예를 들어, 레벨 1 또는 L1 캐시로부터 레벨 2 또는 L2 캐시로) 기록되거나 또는 메모리에 라이트백(write back)될 수 있다. 변경된 데이터가 더 낮은 레벨의 캐시에 기록되거나 또는 메모리에 라이트백된 후, 캐시는 프로세서 코어의 다른 부분이 전력 다운될 준비가 되면, 전력 다운될 준비가 될 수 있다.

### 발명의 내용

#### 과제의 해결 수단

[0004] 고속 캐시를 셧다운(shutdown)시킬 수 있는 장치 및 방법이 개시된다. 일 실시예에서, 캐시 서브시스템은 캐시 메모리 및 이 캐시 메모리에 연결된 캐시 제어기를 포함한다. 이 캐시 제어기는 캐시 서브시스템에 전력을 복구할 때, 변경된 데이터를 이 캐시 메모리에 배타적으로 기록하는 것을 금지하도록 구성된다.

[0005] 일 실시예에서, 방법은 캐시 메모리를 포함하는 캐시 서브시스템에 전력을 복구하는 단계를 포함한다. 본 방법은 변경된 데이터가 캐시 메모리에 배타적으로 기록되는 것을 금지하는 단계를 더 포함한다.

### 도면의 간단한 설명

[0006] 본 발명의 다른 측면은 아래에 간략히 설명된 첨부 도면을 참조하여 이하 상세한 설명을 판독하면 보다 명확해질 것이다.

도 1은 컴퓨터 시스템의 일 실시예의 블록도;

도 2는 다수의 코어 및 공유된 캐시를 구비하는 프로세서의 일 실시예의 블록도;

도 3은 캐시 서브시스템의 일 실시예의 블록도;

도 4는 전력을 복구할 때 임계값에 도달되기 전에 캐시로부터 변경된 데이터를 배제하는 캐시 서브시스템을 동작시키는 방법의 일 실시예의 흐름도;

도 5는 기록 바이패스 모드(write bypass mode)에서 캐시 서브시스템을 동작시키는 방법의 일 실시예의 흐름도;

도 6은 기록 바이패스 모드에서 동작을 예시하는 캐시 서브시스템의 일 실시예의 블록도;

도 7은 라이트-쓰루 모드(write-through mode)에서 동작을 예시하는 캐시 서브시스템을 동작시키는 방법의 일 실시예의 흐름도;

도 8은 라이트-쓰루 모드에서 동작을 예시하는 캐시 서브시스템의 일 실시예의 블록도;

도 9는 캐시 서브시스템의 일 실시예를 설명하는 데이터 구조를 포함하는 컴퓨터 판독가능한 매체의 일 실시예

를 도시하는 블록도.

본 발명은 각종 변형과 대안적인 형태로 제공될 수 있으나, 첨부 도면에는 예로서 특정 실시예만이 도시되고 본 명세서에 상세히 설명된다. 그러나, 첨부 도면 및 상세한 설명은 개시된 특정 형태로 본 발명을 제한하려고 의도된 것이 아니고, 오히려 본 발명은 첨부된 청구범위에서 한정된 본 발명의 사상과 범위 내에 있는 모든 변형, 균등률 및 대안을 포함하려고 의도된 것으로 이해된다.

### 발명을 실시하기 위한 구체적인 내용

[0007]

본 발명은 전력을 복구할 때 제한된 시간 동안 캐시 메모리가 변경된 데이터를 메모리 계층에서 다른 위치에 배타적으로 저장하는 것을 금지하는 방법 및 장치에 관한 것이다. 이 제한된 시간은 임계값에 의해 한정될 수 있다. 종래 기술의 캐시 서브시스템에서, 캐시를 전력 다운시켜 이를 슬립(sleep) 상태(예를 들어, 대응하는 프로세서 코어가 아이들 상태일 때)에 두는 것은 캐시 제어기가 변경된 데이터를 대응하는 캐시의 저장 위치에서 조사(examine)하는 것을 포함할 수 있다. 변경된 데이터가 하나 이상의 저장 위치에서 발견된 경우, 이 변경된 데이터는 메모리 계층에서 더 낮은 다른 캐시에 (예를 들어, L1 캐시로부터 L2 캐시로) 기록되거나 또는 메인 메모리에 기록될 수 있다. 이와 대조적으로, 본 발명의 캐시 서브시스템은 아직 임계값에 도달되지 않은 경우 변경된 데이터를 캐시 메모리에서 조사함이 없이 전력 다운될 수 있다. 캐시 메모리가, 임계값에 도달되기 전에 메모리 계층에서 다른 캐시에 및 메모리에 변경된 데이터를 배타적으로 저장하는 것이 금지되므로, 전력 다운하기 전에 캐시를 체크할 필요가 없다. 따라서, 이러한 캐시 서브시스템을 포함하는 프로세서 코어 또는 다른 기능 유닛은 변경된 데이터가 존재하는지 여부를 결정하는 것에 의해 초래되는 고유 지연 없이 이 기능 유닛이 아이들 상태일 때 전력 다운되어 전력을 절감할 수 있다. 일반적으로, 본 명세서에 설명된 바와 같이 프로세서 코어(또는 다른 기능 유닛)에 구현된 캐시 서브시스템은 슬립 상태로부터 빠져나와서 짧은 지속시간 동안 작업(task)을 수행하고 나서 변경된 데이터를 검색하고 이를 다시 메모리 또는 다른 캐시에 기록하는 것에 의해 초래되는 지연 없이 슬립 상태에 다시 신속히 놓일 수 있다.

[0008]

임계값은 여러 방식으로 구현될 수 있다. 일 실시예에서, 임계값은 캐시 서브시스템으로 전력이 복구된 시간으로부터 미리 결정된 시간 기간일 수 있다. 미리 결정된 시간 시간이 경과되기 전에, 캐시 제어기는 변경된 데이터를 대응하는 캐시에 배타적으로 기록하는 것을 금지할 수 있다. 캐시 서브시스템(및/또는 이 서브시스템이 구현된 유닛)이 미리 결정된 시간 기간이 경과되기 전에 아이들 상태가 되었다면, 이 서브시스템은 변경된 데이터를 캐시에서 검색하고 발견된 변경된 데이터를 다른 캐시 또는 메인 메모리에 기록함이 없이 다시 전력 다운될 수 있다. 캐시 서브시스템이 미리 결정된 시간 기간이 경과되기 전에 아이들 상태가 아니라면, 캐시 제어기는 변경된 데이터가 대응하는 캐시에 배타적으로 기록될 수 있게 한다.

[0009]

다른 실시예에서, 임계값은 특정 개수의 이벤트의 발생에 의해 한정될 수 있다. 이 이벤트는 캐시 퇴거(cache eviction), 변경된 데이터가 실행 유닛에 의해 생성된 인스턴스(instance), 캐시로 가거나 및/또는 캐시로부터 오는 트래픽의 양 등일 수 있다. 일반적으로, 이벤트는 캐시 서브시스템과 연관된 회로에서 발생하는 처리 활동(processing activity)의 레벨을 나타낼 수 있는 임의의 유형일 수 있다. 임계값이 이벤트-기반인 실시예에서, 임계값에 도달된 시간은 캐시 서브시스템이 통전되는 하나의 인스턴스로부터 변할 수 있다.

[0010]

캐시 서브시스템이 전력 온된 것과 임계값에 도달된 것 사이의 기간 동안 변경된 데이터를 처리하는 것은 여러 방식으로 달성될 수 있다. 일 실시예에서, 캐시 서브시스템은 라이트-쓰루 모드에서 동작할 수 있다. 라이트-쓰루 모드에서 동작할 때, 변경된 데이터는 캐시 및 메모리 계층에서 더 낮은 다른 저장 위치에 (예를 들어, 더 낮은 캐시 또는 메인 메모리에) 기록될 수 있다. 따라서, 변경된 데이터는 캐시에 더하여 메모리 계층에서 더 낮은 위치에 저장된다. 그리하여, 변경된 데이터는 메모리 계층에서 더 낮은 적어도 하나의 저장 위치에 이미 저장되어 있으므로, 이로부터 전력을 제거하기 전에 캐시로부터 변경된 데이터를 복사하고 이를 라이트백할 필요가 없다. 캐시 서브시스템은, 임계값에 도달될 때 또는 전력이 제거될 때 라이트-쓰루 모드에서의 동작을 중단(discontinue)시킬 수 있다. 라이트-쓰루 모드에서의 동작은 슬립(또는 다른 전력 부재) 상태로부터 캐시에 전력이 복구될 때 재개될 수 있다.

[0011]

다른 실시예에서, 캐시 서브시스템은 기록-바이패스 모드에서 동작할 수 있다. 기록 바이패스 모드에서 동작할 때, 캐시 제어기는 변경된 데이터가 캐시에 기록되는 것을 금지할 수 있다. 대신, 기록 바이패스 모드에서 동작하는 동안 생성된 변경된 데이터는 메모리 계층에서 적어도 하나의 더 낮은 레벨의 저장 위치에 기록된다. 예를 들어, L1 데이터 캐시를 위한 캐시 서브시스템이 기록-바이패스 모드에서 동작하고 있다면, 실행 유닛에 의해 생성된 변경된 데이터는 L2 캐시, L3 캐시 및/또는 메인 메모리에 기록될 수 있다. 캐시 서브시스템은 임계값에 도달한 것에 응답하여 또는 전력이 제거될 때 기록-바이패스 모드에서의 동작을 중단시킬 수 있다. 기

록-바이패스 모드에서의 동작이 재개하는 것은 전력이 캐시 서브시스템에 복구될 때 발생할 수 있다.

[0012] 변경된 데이터가 메모리 계층에서 동일한 레벨의 다른 캐시에 상이한 전력 도메인으로 저장될 수 있는 실시예도 가능하고 고려될 수 있는 것으로 이해된다.

[0013] 일부 실시예에서, 다수의 캐시 및 대응하는 서브시스템은 전술한 모드들 중 하나의 모드에서 동작될 수 있는 것으로 이해된다. 예를 들어, L1 캐시 및 L2 캐시를 구비하는 프로세서 코어에서, 대응하는 캐시 서브시스템은 라이트-쓰루 모드 또는 기록-바이패스 모드 중 하나의 모드에서 동작할 수 있다. 따라서, 2개의 상이한 캐시가 동일한 전력 분배 회로에 연결되면, 신속한 셋다운이 수행되는 이득이 여전히 획득될 수 있다.

[0014] 나아가, 다수의 레벨의 캐시 메모리가 전술한 모드에서 동작할 수 있는 실시예에서, 두 캐시 서브시스템은 동일한 모드에서 동작할 필요가 없다. 예를 들어, L1 캐시는 기록-바이패스 모드에서 동작할 수 있는 반면, L2 캐시는 라이트-쓰루 모드에서 동작할 수 있다.

[0015] 도 1은 컴퓨터 시스템(10)의 일 실시예의 블록도이다. 도시된 실시예에서, 컴퓨터 시스템(10)은 메모리(6)에 연결된 집적 회로(IC)(2)를 포함한다. 도시된 실시예에서, IC(2)는 이 실시예에서 프로세서 코어인 다수의 프로세서 코어(11)를 구비하는 시스템온칩(system on chip: SOC)이다. 각종 실시예에서, 프로세서 코어의 개수는 하나이거나, 또는 IC 다이에 구현될 수 있을 만큼 많이 있을 수 있다. 멀티 코어 실시예에서, 프로세서 코어(11)는 서로 동일하거나(즉, 대칭적인 멀티 코어이거나) 또는 하나 이상의 코어가 서로 상이할 수 있다(즉, 비대칭 멀티 코어일 수 있다). 프로세서 코어(11)는 하나 이상의 실행 유닛, 캐시 메모리, 스케줄러, 분기 예측(branch prediction) 회로 등을 각각 포함할 수 있다. 나아가, 각 프로세서 코어(11)는 컴퓨터 시스템(10)을 위한 메인 메모리로 기능할 수 있는 메모리(6)에 액세스 요청을 선언(assert)하도록 구성될 수 있다. 이러한 요청은 판독 요청 및/또는 기록 요청을 포함할 수 있고, 노쓰 브리지(north bridge)(12)에 의해 각 프로세서 코어(11)로부터 초기에 수신될 수 있다. 메모리(6)에 액세스 요청은 특정 명령의 실행에 응답하여 개시될 수 있고, 프리페치 동작(prefetch operation)에 응답하여 개시될 수도 있다.

[0016] I/O 인터페이스(13)는 도시된 실시예에서 노쓰 브리지(12)에 연결된다. I/O 인터페이스(13)는 컴퓨터 시스템(10)에서 사우쓰 브리지 디바이스(south bridge device)로 기능할 수 있다. 다수의 상이한 유형의 주변 버스들이 I/O 인터페이스(13)에 연결될 수 있다. 이 특정 예에서, 버스 유형은 주변 컴포넌트 상호 연결(peripheral component interconnect: PCI) 버스, PCI-확장된(PCI-X), PCIE(PCI 익스프레스) 버스, 기가비트 이더넷(gigabit Ethernet: GBE) 버스 및 USB(universal serial bus)를 포함한다. 그러나, 이들 버스 유형은 예시적인 것이고, 많은 다른 버스 유형이 I/O 인터페이스(13)에 연결될 수 있다. 각종 유형의 주변 디바이스(미도시)는 주변 버스 중 일부 또는 전부에 연결될 수 있다. 이러한 주변 디바이스는 키보드, 마우스, 프린터, 스캐너, 조이스틱 또는 다른 유형의 게임 제어기, 매체 레코딩 디바이스, 외부 저장 디바이스, 네트워크 인터페이스 카드 등을 포함한다(그러나 이들로 제한되지 않는다). 대응하는 주변 버스를 통해 I/O 유닛(13)에 연결될 수 있는 주변 디바이스의 적어도 일부는 직접 메모리 액세스(direct memory access: DMA)를 사용하여 메모리 액세스 요청을 선언할 수 있다. (판독 및 기록 요청을 포함할 수 있는) 이들 요청은 I/O 인터페이스(13)를 통해 노쓰 브리지(12)에 전달될 수 있다.

[0017] 도시된 실시예에서, IC(2)는 컴퓨터 시스템(10)의 디스플레이(3)에 연결된 그래픽 처리 유닛(14)을 포함한다. 디스플레이(3)는 평판 패널 LCD(liquid crystal display), 플라즈마 디스플레이, CRT(cathode ray tube) 또는 임의의 다른 적절한 디스플레이 유형일 수 있다. GPU(14)는 각종 비디오 처리 기능을 수행하고 처리된 정보를 시각적 정보로 출력하기 위해 디스플레이(3)에 제공할 수 있다.

[0018] 도시된 실시예에서 메모리 제어기(18)는 노쓰 브리지(12)에 통합될 수 있으나, 다른 실시예에서 노쓰 브리지(12)와는 별개일 수 있다. 메모리 제어기(18)는 노쓰 브리지(12)로부터 전달된 메모리 요청을 수신할 수 있다. (프리페치를 포함하는) 판독 요청에 응답하여 메모리(6)로부터 액세스된 데이터는 메모리 제어기(18)에 의해 노쓰 브리지(12)를 통해 요청 에이전트(requesting agent)에 전달될 수 있다. 기록 요청에 응답하여, 메모리 제어기(18)는 노쓰 브리지(12)를 통해 요청 에이전트로부터 기록될 요청 및 데이터를 수신할 수 있다. 다수의 메모리 액세스 요청이 주어진 시간에 펜딩(pending) 중인 경우, 메모리 제어기(18)는 이들 요청들 사이를 중재(arbitrate)할 수 있다.

[0019] 도시된 실시예에서 메모리(6)는 일 실시예에서 복수의 메모리 모듈로 구현될 수 있다. 각 메모리 모듈은 장착된 하나 이상의 메모리 디바이스(예를 들어, 메모리 칩)를 포함할 수 있다. 다른 실시예에서, 메모리(6)는 IC(2)가 장착될 수 있는 마더보드 또는 다른 캐리어 상에 장착된 하나 이상의 메모리 디바이스를 포함할 수 있

다. 더 다른 실시예에서, 메모리(6)의 적어도 일부는 IC(2) 자체의 다이 상에 구현될 수 있다. 전술한 각종 구현의 조합을 구비하는 실시예도 가능하고 고려될 수 있다. 메모리(6)는 동작 동안 IC(2)와 사용하기 위한 랜덤 액세스 메모리(random access memory: RAM)를 구현하는데 사용될 수 있다. 구현된 RAM은 정적 RAM(SRAM) 또는 동적 RAM(DRAM)일 수 있다. 메모리(6)를 구현하는데 사용될 수 있는 DRAM의 유형은 더블 데이터 레이트(double data rate: DDR) DRAM, DDR2 DRAM, DDR3 DRAM 등을 포함한다(그러나 이들로 제한되지 않는다).

[0020] 도 1에 명시적으로 도시되지는 않았으나, IC(2)는 프로세서 코어(11)의 외부에 하나 이상의 캐시 메모리를 더 포함할 수 있다. 후술되는 바와 같이, 각 프로세서 코어(11)는 L1 데이터 캐시 및 L1 명령 캐시를 포함할 수 있다. 일부 실시예에서, 각 프로세서 코어(11)는 대응하는 L2 캐시와 연관될 수 있다. 각 L2 캐시는 대응하는 프로세서 코어의 내부 또는 외부에 있을 수 있다. 프로세서 코어(11) 중에 공유된 L3 캐시는 일 실시예에서 IC(2)에 포함될 수 있다. 일반적으로, IC(2)의 각종 실시예는 캐시 메모리를 다수의 상이한 레벨로 구현할 수 있고, 여기서 캐시 메모리의 일부는 프로세서 코어들 사이에 공유되는 반면, 다른 캐시 메모리는 프로세서 코어(11)들 중 특정한 코어에 전용될 수 있다.

[0021] 도시된 실시예에서 노쓰 브리지(12)는 IC(2)의 각종 기능 유닛의 전력 소비를 모니터링하고 이를 제어하는데 사용될 수 있는 전력 관리 유닛(15)을 더 포함한다. 보다 구체적으로, 전력 관리 유닛(15)은 IC(2)의 다른 기능 유닛 각각의 활동 레벨을 모니터링할 수 있고, 주어진 기능 유닛이 아이들 상태(예를 들어, 특정 시간 기간 동안 활동이 없는 상태)인 것으로 결정된 경우 전력 관리 액션을 수행할 수 있다. 나아가, 전력 관리 유닛(15)은 아이들 기능 유닛이 활성화되어 작업을 수행할 필요가 있는 경우에 전력 관리 액션을 수행할 수 있다. 전력 관리 액션은 전력을 제거하는 것, 클록 신호를 게이팅하는 것, 전력을 복구하는 것, 클록 신호를 복구하는 것, 동작 전압을 감소시키거나 증가시키는 것, 및 클록 신호의 주파수를 감소시키거나 증가시키는 것을 포함할 수 있다. 일부 경우에, 전력 관리 유닛(15)은 프로세서 코어(11)들 중에 작업 부하를 재배정(re-allocate)하여 각 코어들이 열 설계 상의 전력 한계 내에 유지될 수 있도록 할 수 있다. 일반적으로, 전력 관리 유닛(15)은 IC(2)의 다른 기능 유닛으로의 전력을 제어하여 분배하는 것과 관련된 기능을 수행할 수 있다.

[0022] 도 2는 프로세서 코어(11)의 일 실시예의 블록도이다. 프로세서 코어(11)는 시스템 메모리(예를 들어, 도 1의 메모리(6))에 저장된 명령을 실행하도록 구성된다. 이를 명령 중 많은 명령은 메모리(6)에 저장된 데이터에 작용할 수 있다. 메모리(6)는 컴퓨터 시스템 전체에 걸쳐 물리적으로 분배될 수 있고 및/또는 하나 이상의 처리 노드(11)에 의해 액세스될 수 있다는 것이 주목된다.

[0023] 도시된 실시예에서, 프로세서 코어(11)는 L1 명령 캐시(106) 및 L1 데이터 캐시(128)를 포함할 수 있다. 프로세서 코어(11)는 아래에서 보다 상세히 설명된 명령 캐시(106)에 연결된 프리페치 유닛(108)을 포함할 수 있다. 디스패치 유닛(dispatch unit)(104)은 명령 캐시(106)로부터 명령을 수신하고 동작을 스케줄러(들)(118)에 디스패치하도록 구성될 수 있다. 하나 이상의 스케줄러(118)는 디스패치 유닛(104)으로부터 디스패치된 동작을 수신하고 동작을 하나 이상의 실행 유닛(들)(124)에 발행(issue)하도록 연결될 수 있다. 실행 유닛(들)(124)은 하나 이상의 정수(integer) 유닛, 하나 이상의 부동 소수점(floating point) 유닛을 포함할 수 있다. 적어도 하나의 부하-저장 유닛(126)은 도시된 실시예에서 실행 유닛(124) 중에 포함된다. 실행 유닛(들)(124)에 의해 생성된 결과는 하나 이상의 결과 버스(130)에 출력될 수 있다(명료함을 위해 본 명세서에서는 단일 결과 버스가 도시되어 있으나, 다수의 결과 버스도 가능하고 고려될 수 있다). 이 결과는 이후 발행된 명령에 오퍼랜드 값(operand value)으로 사용되거나 및/또는 레지스터 파일(116)에 저장될 수 있다. 퇴거 큐(retire queue)(102)는 스케줄러(들)(118) 및 디스패치 유닛(104)에 연결될 수 있다. 퇴거 큐(102)는 각 발행된 동작이 퇴거될 수 있을 때를 결정하도록 구성될 수 있다.

[0024] 일 실시예에서, 프로세서 코어(11)는 x86 아키텍처(인텔 아키텍처-32 또는 IA-32로도 알려진 것)에 호환가능하도록 설계될 수 있다. 다른 실시예에서, 프로세서 코어(11)는 64-비트 아키텍처에 호환가능할 수 있다. 다른 아키텍처에 호환가능한 프로세서 코어(11)의 실시예도 더 고려될 수 있다.

[0025] 프로세서 코어(11)는 많은 다른 컴포넌트를 더 포함할 수 있다는 것이 주목된다. 예를 들어, 프로세서 코어(11)는 명령 스레드(thread)를 실행할 때 분기를 예측하도록 구성된 분기 예측 유닛(미도시)을 포함할 수 있다. 일부 실시예(예를 들어, 독립 프로세서로 구현된 경우)에서, 프로세서 코어(11)는 메모리(6)에 대한 판독 및 기록을 제어하도록 구성된 메모리 제어기를 더 포함할 수 있다.

[0026] 명령 캐시(106)는 디스패치 유닛(104)에 의해 페치(fetch)하기 위한 명령을 저장할 수 있다. 명령 코드는 프리페치 유닛(108)을 통해 시스템 메모리(200)로부터 코드를 프리페치하는 것에 의해 저장하기 위해 명령 캐시(106)에 제공될 수 있다. 명령 캐시(106)는 각종 구성(예를 들어, 세트-결합(set-associative), 전체-결합

(fully-associative) 또는 직접-맵핑)으로 구현될 수 있다.

[0027] 프로세서 코어(11)는 L2 캐시(129)와 더 연관될 수 있다. 도시된 실시예에서, L2 캐시(129)는 프로세서 코어(11) 내부에 있고 이 프로세서 코어와 동일한 전력 도메인에 포함된다. L2 캐시(129)가 프로세서 코어(11)의 외부에 있고 이 프로세서 코어와 다른 전력 도메인에 포함된 실시예도 가능하고 고려될 수 있다. 명령 캐시(106)가 명령을 저장하는데 사용될 수 있고 데이터 캐시(128)는 데이터(예를 들어, 오퍼런드)를 저장하는데 사용될 수 있는 반면, L2 캐시(129)는 명령 및 데이터를 저장하는데 사용된 단일화된 캐시일 수 있다. 그러나, 별개의 L2 캐시들이 명령 및 데이터를 위해 구현된 실시예도 가능하고 고려될 수 있다.

[0028] 디스패치 유닛(104)은 실행 유닛(들)(124)에 의해 실행가능한 동작 및 오퍼런드 어드레스 정보, 중간 데이터 및 /또는 변위(displacement) 데이터를 출력할 수 있다. 일부 실시예에서, 디스패치 유닛(104)은 특정 명령을 실행 유닛(들)(124) 내에서 실행가능한 동작으로 디코딩하는 디코딩 회로(미도시)를 포함할 수 있다. 간단한 명령은 단일 동작에 대응할 수 있다. 일부 실시예에서, 보다 복잡한 명령은 다수의 동작에 대응할 수 있다. 레지스터(register)의 업데이트를 수반하는 동작을 디코딩할 때, 레지스터 파일(116)에서 레지스터 위치를 예비(reserved)하여 추측(speculative) 레지스터 상태를 저장할 수 있다(대안적인 실시예에서, 재정렬 버퍼를 사용하여 각 레지스터에 하나 이상의 추측 레지스터 상태를 저장할 수 있고, 레지스터 파일(116)은 각 레지스터에 수행된 레지스터 상태를 저장할 수 있다). 레지스터 맵(134)은 소스 및 목적지 오퍼런드의 논리적 레지스터 이름을 물리적 레지스터 번호로 변환(translate)하여 레지스터에 새이름(renaming)을 제공할 수 있다. 레지스터 맵(134)은 레지스터 파일(116)에서 현재 배정되어 있거나 배정되어 있지 않은 레지스터를 추적할 수 있다.

[0029] 도 2의 프로세서 코어(11)는 비정렬 실행(out of order execution)을 지원할 수 있다. 퇴거 큐(102)는 레지스터 판독 및 기록 동작을 위해 원래의 프로그램 시퀀스를 추적하고, 추측 명령 실행 및 잘못 예측된 분기의 복구(branch misprediction recovery)를 수행하고 정밀한 예외를 제공할 수 있다. 일부 실시예에서, 퇴거 큐(102)는 추측 레지스터 상태(예를 들어, 재정렬 버퍼와 유사함)에 데이터 값을 저장하는 것에 의해 레지스터에 새이름을 지원할 수 있다. 다른 실시예에서, 퇴거 큐(102)는 재정렬 버퍼와 유사한 기능을 하지만 데이터 값을 저장하지는 못할 수 있다. 동작이 퇴거될 때, 퇴거 큐(102)는 추측 레지스터 상태를 저장하고, 현재 자유로운 레지스터인지를 나타내는 신호를 레지스터 맵(134)에 제공하는 것이 더 이상 필요치 않는 레지스터를 레지스터 파일(116)에서 배정 해제할 수 있다. 이들 상태들을 생성한 동작이 겹중될 때까지 레지스터 파일(116)에 (또는 대안적인 실시예에서, 재정렬 버퍼에) 추측 레지스터 상태를 유지하는 것에 의해, 잘못 예측된 경로를 따라 추측으로-실행된 동작의 결과는 분기 예측이 부정확한 경우 레지스터 파일(116)에서 무효화될 수 있다.

[0030] 일 실시예에서, 레지스터 파일(116)의 주어진 레지스터는 실행된 명령의 데이터 결과를 저장하도록 구성될 수 있고, 실행된 명령에 의해 업데이트될 수 있는 하나 이상의 플래그 비트(flag bit)를 저장할 수 있다. 플래그 비트는 (예를 들어, 덧셈 또는 곱셈 동작을 한 결과 올림수(carry) 또는 오버플로우(overflow) 상황이 존재하는 것을 나타내는) 후속 명령을 실행할 때 중요할 수 있는 각종 유형의 정보를 전달할 수 있다. 구조적으로, 플래그를 저장하는 플래그 레지스터가 한정될 수 있다. 따라서, 주어진 레지스터에의 기록은 논리적 레지스터 및 플래그 레지스터를 업데이트시킬 수 있다. 모든 명령이 하나 이상의 플래그를 업데이트하는 것은 아니라는 것이 주목된다.

[0031] 레지스터 맵(134)은 물리적 레지스터를, 동작할 목적지 오퍼런드로 지정된 특정 논리적 레지스터(예를 들어, 구조화된/architected) 레지스터 또는 마이크로 구조로 지정된 레지스터)에 할당(assign)할 수 있다. 디스패치 유닛(104)은 레지스터 파일(116)이 주어진 동작에서 소스 오퍼런드로 지정된 논리적 레지스터에 할당된 이전에 배정된 물리적 레지스터를 구비하는 것으로 결정할 수 있다. 레지스터 맵(134)은 이 논리적 레지스터에 가장 최근에 할당된 물리적 레지스터에 태그를 제공할 수 있다. 이 태그는 레지스터 파일(116)에서 오퍼런드의 데이터 값에 액세스하거나 또는 결과 버스(130)에 결과를 전달하는 것을 통해 데이터 값을 수신하는데 사용될 수 있다. 오퍼런드가 메모리 위치에 대응하는 경우, 오퍼런드 값은 (레지스터 파일(116)에 결과를 전달하고 및/또는 저장하기 위해) 부하-저장 유닛(126)을 통해 결과 버스에 제공될 수 있다. 오퍼런드 데이터 값은 스케줄러(들)(118) 중 하나에 의해 동작이 발행될 때 실행 유닛(들)(124)에 제공될 수 있다. 대안적인 실시예에서, 오퍼런드 값은 (동작이 발행될 때 대응하는 실행 유닛(124)에 제공되는 대신에) 동작이 디스패치될 때 대응하는 스케줄러(118)에 제공될 수 있다는 것이 주목된다.

[0032] 본 명세서에 사용된 바와 같이, 스케줄러는 동작이 실행 준비가 되고 준비 동작을 하나 이상의 실행 유닛에 발행할 때를 검출하는 디바이스이다. 예를 들어, 예약 스테이션은 하나의 유형의 스케줄러일 수 있다. 실행 유닛마다 독립적인 예약 스테이션이 제공되거나, 또는 동작을 발행하는 중앙 예약 스테이션이 제공될 수 있다.

다른 실시예에서, 퇴거 때까지 동작을 유지하는 중앙 스케줄러가 사용될 수 있다. 각 스케줄러(118)는 실행 유닛(124)으로 발행을 대기하는 여러 펜딩 동작에 대한 동작 정보(예를 들어, 동작 및 오퍼런드 값, 오퍼런드 태그 및/또는 중간 데이터)를 보유할 수 있다. 일부 실시예에서, 각 스케줄러(118)는 오퍼런드 값을 저장하지 않을 수 있다. 대신, 각 스케줄러는 레지스터 파일(116)에서 이용가능한 발행된 동작과 결과를 모니터링하여 오퍼런드 값이 실행 유닛(들)(124)에 의해 (레지스터 파일(116) 또는 결과 버스(130)로부터) 관독되는데 이용가능할 때를 결정할 수 있다.

[0033] 프리페치 유닛(108)은 명령 캐시(106)에 저장하기 위해 메모리(6)로부터 명령 코드를 프리페치할 수 있다. 도시된 실시예에서, 프리페치 유닛(108)은 여러 특정 코드 프리페치 기술 및 알고리즘 중 2개 이상의 상이한 것을 사용할 수 있는 하이브리드 프리페치 유닛이다. 프리페치 유닛(108)에 의해 구현된 프리페치 알고리즘을 사용하여 레지스터 및/또는 캐시에 프리페치되고 로딩될 수 있는 데이터의 어드레스를 생성할 수 있다. 프리페치 유닛(108)은 생성된 어드레스 중에서 프리페치 동작의 주어진 경우를 수행하는데 사용되는 어드레스를 선택하도록 중재를 수행하도록 구성될 수 있다.

[0034] 전술한 바와 같이, 프로세서 코어(11)는 L1 데이터 및 명령 캐시를 포함하고, 적어도 하나의 L2 캐시와 연관된다. 일부 경우에, 별개의 L2 캐시는 데이터 및 명령을 위해 각각 제공될 수 있다. L1 데이터 및 명령 캐시는 메모리 계층의 일부일 수 있고, 이 계층에서 프로세서 코어(11)의 구조화된 레지스터 아래에 있을 수 있다. L2 캐시(들)는 메모리 계층에서 L1 데이터 및 명령 캐시 아래에 있을 수 있다. 명시적으로 도시되지는 않았으나, L3 캐시가 더 존재할 수 있고(및 다수의 프로세서 코어(11)들 중에서 공유될 수 있고), 여기서 L3 캐시는 메모리 계층에서 임의의 및 모든 L2 캐시 아래에 있을 수 있다. 메모리 계층에서 캐시 메모리의 각종 레벨 아래에 메인 메모리가 있을 수 있고, 디스크 저장매체(또는 플래시 저장매체)는 메인 메모리 아래에 있을 수 있다.

[0035] 도 3은 예시적인 캐시 서브시스템의 일 실시예를 도시하는 블록도이다. 이 특정 예에서, 캐시 서브시스템은 프로세서 코어의 L2 데이터 캐시에 관한 것이다. 그러나, 여기에 도시된 일반적인 배열은 대응하는 캐시에 변경된 데이터를 저장하는 캐시 서브시스템에도 적용될 수 있다.

[0036] 도시된 실시예에서, 캐시 서브시스템(220)은 L2 데이터 캐시(229) 및 캐시 제어기(228)를 포함한다. 21 데이터 캐시는 데이터(예를 들어, 오퍼런드)를 저장하는데 사용될 수 있고, 각종 구성(예를 들어, 세트-결합, 전체-결합 또는 직접-맵핑)으로 구현될 수 있는 캐시이다.

[0037] 캐시 제어부(228)는 관독 및 기록 동작을 위해 L2 데이터 캐시(229)에의 액세스를 제어하도록 구성된다. 도 3에 도시된 특정 구현에서, 캐시 제어기(228)는 L2 데이터 캐시(229)로부터 데이터를 관독하여 이 데이터를 실행 유닛(들)(124)에 (또는 특정 명령을 실행하기 위한 실행 유닛에 의해 액세스되는 레지스터에) 제공할 수 있다. 나아가, 캐시 제어기(228)는 내부에 저장된 데이터가 오래된 것일 때 또는 새로운 데이터를 추가하기 위해 제거될 때 캐시 라인의 퇴거를 수행할 수 있다. 캐시 제어기(228)는 다른 캐시 서브시스템과 (예를 들어, L1 캐시를 위한 캐시 제어기와) 통신할 수 있을 뿐만 아니라 메모리 제어기와 통신하며 데이터가 메모리 계층에서 더 낮은 레벨의 저장 위치에 기록될 수 있게 한다.

[0038] 도시된 실시예에서 캐시 제어 유닛(228)에 의해 제공된 다른 기능은 변경된 데이터가 L2 데이터 캐시(229)에 기록되고 이 캐시에 배타적으로 저장될 수 있을 때를 제어가능하다. 캐시 제어기(228)는 실행 유닛(들)(124)에 의해 실행된 명령으로부터 초래되는 데이터를 수신하고, 이 데이터를 L2 데이터 캐시(229)에 기록하는 것을 제어할 수 있다. 이 실시예에서, 캐시 제어기(228)는 캐시 서브시스템(220)에 전력을 복구할 때 특정 시간 기간 동안 변경된 데이터가 L2 데이터 캐시(229)에 배타적으로 기록되는 것을 금지할 수 있다. 즉, 특정 시간 기간 동안, 캐시 제어기(228)는 변경된 데이터가 메모리 계층에서 더 아래쪽 다른 위치에 기록되지 않는 한, 변경된 데이터가 L2 데이터 캐시(229)에 기록되는 것을 금지하거나, 또는 변경된 데이터가 L2 데이터 캐시(229)에 기록되는 것을 금지할 수 있다.

[0039] 캐시 제어기가 L2 데이터 캐시(229)에 변경된 데이터를 배타적으로 기록하고 저장하는 것을 금지하는 시간 기간은 임계값에 기초하여 결정될 수 있다. 이 임계값은 시간-기반 또는 이벤트-기반일 수 있다. 도시된 실시예에서, 캐시 제어기(228)는 미리 결정된 시간 임계값에 대해 캐시 서브시스템(220)에의 전력을 복구한 이후의 시간 기간을 추적하도록 구성된 타이머(232)를 포함한다. 도시된 실시예에서 캐시 제어기(228)는 특정 개수의 미리 한정된 이벤트의 발생(예를 들어, 변경된 데이터가 실행 유닛에 의해 생성된 인스턴스, 명령이 실행된 인스턴스 우, 메모리에 액세스된 인스턴스 등)을 카운트하고 이를 추적하도록 구성된 이벤트 카운터(234)를 더 포함한다. 카운트된 이벤트의 개수는 대응하는 임계값과 비교될 수 있다. 각종 실시예에서, 캐시 제어기(228)는 타이머(232) 또는 이벤트 카운터(234) 중 하나만을 포함할 수 있다는 것이 주목된다. 일반적으로, 임계값을 구현하는

임의의 적절한 메커니즘이 캐시 제어기(228)의 주어진 실시예에 포함될 수 있다.

[0040] 캐시 서브시스템(220)에 전력을 복구하는 것에 후속하여 임계값에 도달되거나 또는 이 임계값을 초과하는 경우, 캐시 제어기(228)는 L1 데이터 캐시가 메모리 계층에서 더 낮은 다른 위치에 변경된 데이터를 배타적으로 저장하는 것을 금지하는 것을 중단시킬 수 있다. 임계값에 도달한 것에 후속하여 실행 유닛(또는 다른 소스)에 의해 변경된 데이터가 발행되면 변경된 데이터는 되거 전에 추가적인 라이트백을 요구함이 없이 L2 데이터 캐시(229)에 기록될 수 있다.

[0041] 일부 경우에, 임계값은 캐시 서브시스템(220) 또는 대응하는 기능 유닛(예를 들어, 전술한 바와 같이 프로세서 코어(11)) 전에는 도달되지 않을 수 있다. 이러한 경우에, 캐시 서브시스템(220)(및 대응하는 기능 유닛)은 전력을 제거하는 것에 의해 슬립 상태에 놓일 수 있다. 이 경우에 임계값에 도달되지 않았으므로, L2 데이터 캐시(229)는 변경된 데이터를 저장하지 않게 된다. 따라서, 변경된 데이터가 L2 데이터 캐시(229)에 저장되어 있지 않으므로, 변경된 데이터를 캐시에서 검색하거나 또는 발견된 변경된 데이터를 메모리 계층에서 더 낮은 위치에 라이트백할 필요가 없다. 이에 의해 캐시를 전력 다운하는 결정이 이루어지면 슬립 상태에 들어가는데 드는 시간 기간을 상당히 감소시킬 수 있다. 그 결과 전력 소비가 감소될 수 있다. 나아가, 슬립 상태에 신속히 들어가고 빠져나가는 능력이 있어 이에 의해 캐시 서브시스템(및 대응하는 기능 유닛)은 단기 작업을 수행하도록 전력 투입되고 나서 슬립 상태로 다시 신속히 전력 다운될 수 있다.

[0042] 도 4는 전력 복구시에 및 임계값에 도달되기 전에 캐시로부터 변경된 데이터를 배제하는 캐시 서브시스템을 동작시키는 방법의 일 실시예의 흐름도를 도시한다. 본 명세서에 설명된 방법(400)의 실시예는 (예를 들어, 전술한 바와 같이) 프로세서 코어 또는 다른 유형의 처리 노드에서 구현된 캐시 서브시스템에 관한 것이다. 그러나, 유사한 방법이 다른 기능 유닛의 일부이거나 이와 별개로 구현되는지 여부에 상관없이 임의의 캐시 서브시스템에 적용될 수 있다.

[0043] 방법(400)은 캐시 서브시스템을 포함하는 처리 노드에 전력을 복구하는 것으로 시작한다(블록 405). 처리 노드에 전력을 복구할 때, 명령의 실행이 시작할 수 있다(블록 410). 명령을 실행하는 것은 실행 유닛 또는 다른 적절한 회로에 의해 수행될 수 있다. 일부 경우에, 명령을 실행하면 메모리로부터 캐시로 이전에 제공된 데이터를 변경시킬 수 있다. 그러나, 임계값에 도달하기 전의 시간 동안, 캐시 제어기는 캐시가 변경된 데이터를 메모리 계층에서 다른 저장 위치에 배타적으로 저장하는 것을 금지할 수 있다(블록 415). 일 실시예에서, 이것은 변경된 데이터가 캐시에 기록되는 것에 더하여 메모리 계층에서 더 낮은 적어도 하나의 다른 위치에 기록되게 하는 것에 의해 달성될 수 있다. 다른 실시예에서, 이것은 변경된 데이터를 캐시에 기록하는 것을 금지하고, 이 변경된 데이터를 메모리 계층에서 더 낮은 레벨의 저장 위치에 기록하는 것에 의해 달성될 수 있다. 캐시가 변경된 데이터를 메모리 계층에서 더 낮은 다른 레벨의 위치에 배타적으로 저장하는 것을 금지하는 것은, 임계값에 도달되지 않는 한, 계속될 수 있다.

[0044] 임계값에 도달되지 않았으나(블록 420, 아니오), 처리 노드가 아이들 상태가 아닌 경우(블록 425, 아니오), 처리는 계속될 수 있다(블록 425). 임계값에 도달되지 않고(블록 420, 아니오), 처리 노드가 아이들 상태인 경우(블록 425, 예), 처리 노드는 전력을 제거하는 것에 의해 슬립 모드에 놓일 수 있다(블록 430). 전력을 제거하기 전에 임계값에 도달되지 않았으므로, 내부에 배타적으로 저장된 변경된 데이터를 캐시에서 검색하거나 또는 이 변경된 데이터를 메모리에 또는 메모리 계층에서 더 낮은 레벨의 캐시에 라이트백할 필요가 없다. 따라서, 슬립 모드에 진입하는 것이, 변경된 데이터가 캐시 메모리에 배타적으로 저장되었다면 가능할 수 있었던 것보다 더 고속으로 달성될 수 있다.

[0045] 처리 노드가 아이들 상태가 되기 전에 임계값에 도달되면(블록 420, 예), 캐시 제어기는 변경된 데이터가 캐시 메모리에 배타적으로 저장될 수 있게 한다. 처리 노드가 아이들 상태가 아닌 경우(블록 425), 처리는 계속될 수 있고, 이 경우 캐시 제어기는 변경된 데이터가 캐시에 배타적으로 기록될 수 있게 한다. 임계값에 도달되면, 블록(420)은 처리 노드가 아이들 상태가 될 때까지 '예' 경로에 유지될 수 있다는 것이 주목된다. 처리 노드가 아이들 상태가 되면(블록 425, 예), 전력은 처리 노드로부터 제거되어 이를 슬립 상태에 놓을 수 있다. 그러나, 처리 노드가 아이들 상태가 되기 전에 임계값에 도달되었으므로, 캐시 메모리는 슬립 모드에 진입하기 전에 변경된 데이터를 검색할 수 있다. 캐시에서 발견된 변경된 데이터는 메모리에 또는 더 낮은 레벨의 캐시 메모리에 라이트백될 수 있다.

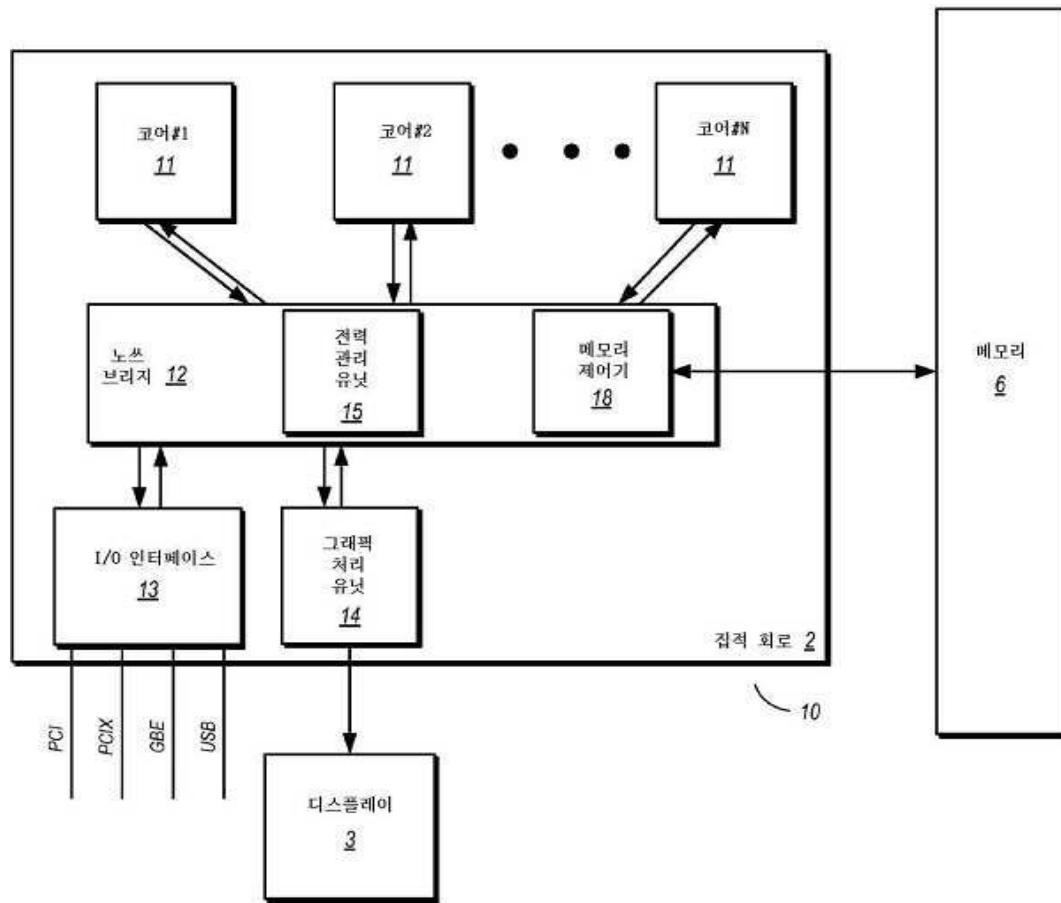
[0046] 도 5 및 도 6은 기록-바이패스 모드라고 지칭되는 모드에서 캐시 서브시스템의 동작을 도시한다. 동작은 도 3에 전술한 캐시 서브시스템(220)의 실시예에 대하여 설명되지만, 본 명세서에 설명된 방법은 캐시 서브시스템의 다른 실시예에서 수행될 수 있는 것으로 이해된다.

- [0047] 도 5에 도시된 바와 같이, 기록 바이패스 모드에서 동작할 때 캐시 제어기(228)는 변경된 데이터가 L1 데이터 캐시(228)에 기록되는 것을 금지할 수 있다. 변경된 데이터는 특정 명령을 실행하는 동안 실행 유닛(들)(124)에 의해 생성될 수 있다(1). 캐시 제어기(228)는 변경된 데이터가 L2 데이터 캐시(229)에 기록되는 것을 금지할 수 있다(2). 변경된 데이터는 더 낮은 레벨의 캐시 메모리 또는 메인 메모리 중 적어도 하나에 기록된다(3). 따라서, L2 데이터 캐시(229)는 기록 바이패스 모드에서 동작할 때 변경된 데이터를 수신하거나 저장하지 못한다.
- [0048] 도 6은 기록-바이패스 모드에서의 동작을 더 도시한다. 방법(500)은 캐시 서브시스템에 전력을 복구하는 (예를 들어, 슬립 상태를 빠져나가는) 것으로 시작한다(블록 505). 방법은 일부 경우에 변경된 데이터를 생성할 수 있는 명령의 실행을 더 포함한다(블록 510). 명령의 실행에 응답하여 변경된 데이터가 생성되면(블록 515, 예), 캐시 제어기는 변경된 데이터가 대응하는 캐시에 기록되는 것을 금지할 수 있고, 이를 더 낮은 레벨의 캐시 또는 메인 메모리에 기록되게 할 수 있다(블록 520). 명령이 변경된 데이터를 생성하지 않는 경우(블록 515, 아니오), 방법은 블록(525)으로 진행할 수 있다.
- [0049] 임계값에 도달되지 않았고(블록 525, 아니오), 캐시 서브시스템과 연관된 처리 노드가 아이들 상태에 있지 않는 경우(블록 530, 아니오), 방법은 블록(510)으로 리턴한다. 임계값에 도달되지 않았으나(블록 525, 아니오), 처리 노드가 아이들 상태가 되었다면(블록 530, 예), 캐시 서브시스템(및 대응하는 처리 노드)은 전력을 제거하는 것에 의해 슬립 상태에 놓일 수 있다(블록 535). 이 예에서는 임계값에 도달되지 않았으므로, 이 변경된 데이터를 캐시에 기록하는 것이 금지되었으므로 변경된 데이터를 캐시에서 검색할 필요가 없다.
- [0050] 임계값에 도달되면(블록 525, 예), 변경된 데이터를 캐시에 기록하는 것을 허용하면서 처리가 계속될 수 있다(블록 540). 변경된 데이터는 캐시에 배타적으로 기록되고 저장될 수 있다. 캐시는 변경된 데이터가 새로운 데이터를 위해 퇴거될 때까지 또는 캐시 서브시스템이 전력 다운될 때까지 변경된 데이터를 배타적으로 저장할 수 있다. 2개의 이벤트 중 어느 하나가 일어나면, 변경된 데이터는 더 낮은 레벨의 캐시에 또는 메인 메모리에 기록될 수 있다. 블록(545)에서, 처리 노드는 전력이 제거될 수 있는 아이들 상태까지 동작을 계속할 수 있다(블록 535).
- [0051] L2 캐시가 공유된 캐시(즉, 데이터 및 명령을 저장하는 캐시)인 실시예에서, 기록 바이패스 모드의 변동이 구현될 수 있다. 이러한 실시예에서, 임계값에 도달되기 전에, L2 캐시는 명령 캐시로서 배타적으로 동작될 수 있다. 그리하여, 임계값에 도달되지 않았다면, 데이터는 L2 캐시에 기록되지 않는다. 그리하여, 대응하는 캐시 서브시스템이 아이들 상태가 된 시간에 의해 임계값에 도달되지 않은 경우, 이 시스템은 데이터가 기록되지 않았으므로 변경된 데이터를 L2에서 검색할 필요 없이 슬립 상태에 놓일 수 있다. 한편, 캐시 서브시스템이 아이들 상태에 있기 전에 임계값에 도달되면, (변경된 및 변경되지 않은) 데이터를 L2 캐시에 기록하는 것이 이후 허용될 수 있다.
- [0052] 도 7 및 도 8은 라이트-쓰루 모드라고 지칭되는 모드에서 캐시 서브시스템의 동작을 도시한다. 동작은 도 3에 전술한 캐시 서브시스템(220)의 실시예에 대하여 설명되지만, 본 명세서에 설명된 방법은 캐시 서브시스템의 다른 실시예에서 수행될 수 있는 것으로 이해된다.
- [0053] 도 7에 도시된 바와 같이, 라이트-쓰루 모드에서 동작하는 동안 변경된 데이터를 L1 데이터 캐시에 기록하는 것은 변경된 데이터를 메모리 계층에서 더 아래쪽 저장 위치에 추가적으로 기록하는 것과 수반될 수 있다. 변경된 데이터는 특정 명령의 실행 동안 실행 유닛(들)(124)에 의해 생성될 수 있다(1). 캐시 제어기(228)는 변경된 데이터를 L2 데이터 캐시(229)에 기록하는 것에 의해 응답할 수 있다(2). 나아가, 변경된 데이터는 더 낮은 레벨의 캐시와 같은 메모리 계층에서 더 아래쪽 적어도 하나의 저장 위치에 또는 메인 메모리에 기록될 수 있다(3). 변경된 데이터가 더 낮은 레벨의 캐시에 기록된 경우에, 변경된 데이터는 적어도 2개의 상이한 위치에 저장되어서, L2 데이터 캐시(229)에 배타적이지 않다. 변경된 데이터가 메모리에 라이트백되면, 이 메모리는 L2 데이터 캐시(229)에서 대응하는 더티 비트(dirty bit)를 제거하여, 변경된 데이터 상태를 제거할 수 있다.
- [0054] 라이트-쓰루 모드에서의 동작은 도 8에 더 도시된다. 방법(700)은 캐시 서브시스템에 전력을 복구하는 (예를 들어, 슬립 상태를 빠져나가는) 것으로 시작한다(블록 705). 방법은 일부 경우에 변경된 데이터를 생성할 수 있는 명령의 실행을 더 포함한다(블록 710). 변경된 데이터가 명령의 실행에 응답하여 생성되면(블록 715, 예), 캐시 제어기는 변경된 데이터가 대응하는 캐시에 기록되는 것을 허용할 수 있고, 데이터가 더 낮은 레벨의 캐시 또는 메인 메모리에 기록될 수 있게 한다(블록 720). 명령이 변경된 데이터를 생성하지 않는 경우(블록 715, 아니오), 방법은 블록(725)으로 진행할 수 있다.

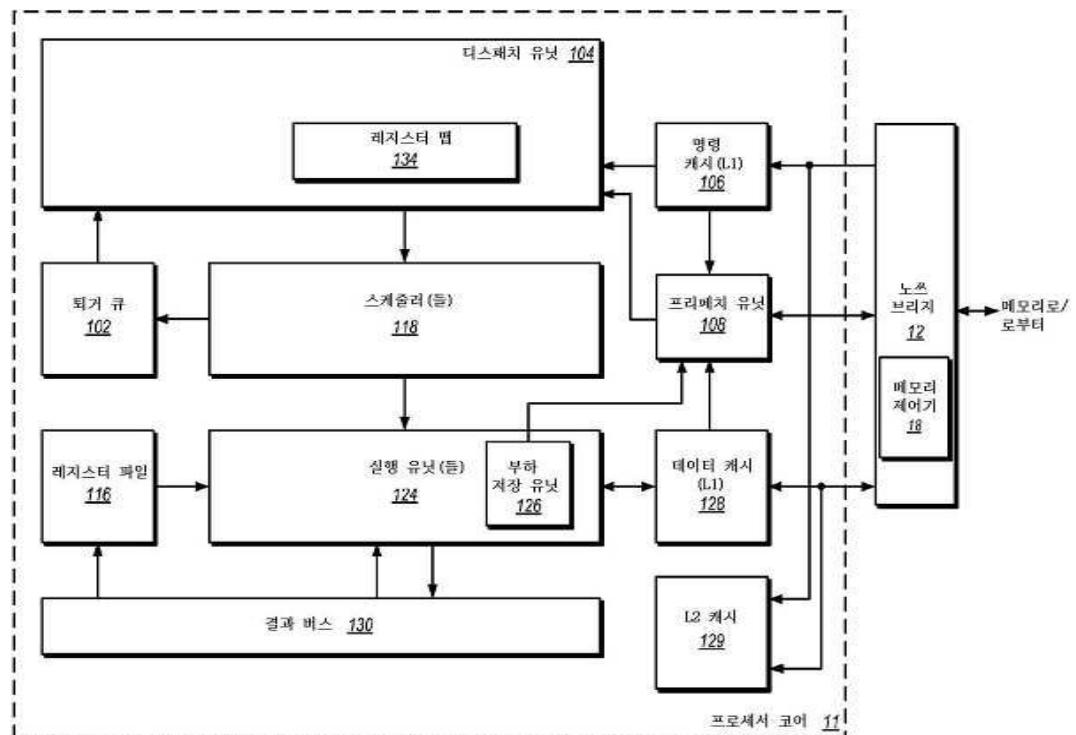
- [0055] 임계값에 도달되지 않았고(블록 725, 아니오), 캐시 서브시스템과 연관된 처리 노드가 아이들 상태가 아닌 경우(블록 730, 아니오), 방법은 블록(710)으로 리턴한다. 임계값에 도달되지 않았으나(블록 725, 아니오), 처리 노드가 아이들 상태가 된 경우(블록 730, 예), 캐시 서브시스템(및 대응하는 처리 노드)은 전력을 제거하는 것에 의해 슬립 상태에 놓일 수 있다(블록 735). 이 예에서는 임계값에 도달되지 않았으므로, 캐시에 기록된 변경된 데이터가 메모리 계층에서 더 아래쪽 적어도 하나의 저장 위치에 더 저장되기 때문에 변경된 데이터를 캐시에서 검색할 필요가 없다.
- [0056] 임계값에 도달되면(블록 725, 예), 변경된 데이터가 캐시에 기록되는 것을 허용하면서 처리가 계속될 수 있다(블록 740). 변경된 데이터는 캐시에 배타적으로 기록되고 저장될 수 있다. 캐시는 변경된 데이터가 새로운 데이터를 위해 퇴거될 때까지 또는 캐시 서브시스템이 전력 다운될 때까지 변경된 데이터를 배타적으로 저장할 수 있다. 2개의 이벤트 중 어느 하나가 일어나면, 변경된 데이터는 더 낮은 레벨의 캐시 또는 메인 메모리에 기록될 수 있다. 블록(745)에서, 처리 노드는 전력이 제거될 수 있는 아이들 상태까지 동작을 계속할 수 있다(블록 735).
- [0057] 다음으로 도 9를 참조하면, 도 9는 시스템(10)을 대표하는 데이터베이스(905)를 포함하는 컴퓨터 액세스가능한 저장 매체(900)의 블록도를 도시한다. 일반적으로 말하면, 컴퓨터 액세스가능한 저장 매체(900)는 사용 동안 컴퓨터에 의해 액세스되며 명령 및/또는 데이터를 컴퓨터에 제공하는 비일시적인 저장 매체를 포함할 수 있다. 예를 들어, 컴퓨터 액세스가능한 저장 매체(900)는 예를 들어, 자기 또는 광 매체, 예를 들어, 디스크(고정식 또는 이동식), 테이프, CD-ROM 또는 DVD-ROM, CD-R, CD-RW, DVD-R, DVD-RW 또는 블루 레이(Blu-Ray)와 같은 저장 매체를 포함할 수 있다. 저장 매체는 휴발성 또는 비휘발성 메모리 매체, 예를 들어, RAM(예를 들어, 동기 동적 RAM(SDRAM), 더블 데이터 레이트(DDR, DDR2, DDR3 등) SDRAM, 저전력 DDR(LPDDR2 등) SDRAM, 램버스 DRAM(RDRAM), 정적 RAM(SRAM) 등), ROM, 플래시 메모리, USB(Universal Serial Bus) 인터페이스와 같은 주변 인터페이스를 통해 액세스가능한 비휘발성 메모리(예를 들어 플래시 메모리) 등을 더 포함할 수 있다. 저장 매체는 MEMS(microelectromechanical system) 및 네트워크 및/또는 무선 링크와 같은 통신 매체를 통해 액세스가능한 저장 매체를 포함할 수 있다.
- [0058] 일반적으로, 시스템(10)을 대표하는 데이터(905) 및/또는 컴퓨터 액세스가능한 저장 매체(900)에 저장된 부분은 프로그램에 의해 판독되고 직접 또는 간접 사용되어, 시스템(10)을 포함하는 하드웨어를 제조할 수 있는 데이터베이스 또는 다른 데이터 구조일 수 있다. 예를 들어, 데이터베이스(905)는 베릴로그(Verilog) 또는 VHDL과 같은 하이 레벨 설계 언어(high level design language: HDL)에서 하드웨어 기능의 거동-레벨 설명(behavioral-level description) 또는 레지스터-전달 레벨(register-transfer level: RTL) 설명일 수 있다. 이 설명은 합성 도구에 의해 판독될 수 있고 이 합성 도구는 이 설명을 합성하여 합성 라이브러리로부터 게이트 리스트를 포함하는 네트리스트(netlist)를 생성할 수 있다. 이 네트리스트는 시스템(10)을 포함하는 하드웨어의 기능을 나타내는 게이트 세트를 포함한다. 네트리스트는 마스크에 적용될 기하학적 형상을 설명하는 데이터 세트를 생성하도록 배치되고 라우팅될 수 있다. 마스크를 각종 반도체 제조 단계에 사용하여 시스템(10)에 대응하는 반도체 회로 또는 회로들을 생성할 수 있다. 대안적으로, 컴퓨터 액세스가능한 저장 매체(900)의 데이터베이스(905)는 원하는 네트리스트(합성 라이브러리를 갖거나 없는) 또는 데이터 세트이거나 또는 그래픽 데이터 시스템(Graphic Data System: GDS) II 데이터일 수 있다.
- [0059] 컴퓨터 액세스가능한 저장 매체(900)는 시스템(10)의 표현을 저장하지만, 다른 실시예에서는 IC(2)를 포함하는 시스템(10)의 원하는 일부, 에이전트(예를 들어, 처리 코어(11), I/O 인터페이스(13), 노쓰 브리지(12), 캐시 서브시스템 등)의 임의의 세트 또는 에이전트의 일부의 표현을 저장할 수 있다.
- [0060] 본 발명은 특정 실시예를 참조하여 설명되었으나, 실시예는 예시적인 것일 뿐 본 발명의 범위는 이들 실시예로 제한되는 것은 아닌 것으로 이해된다. 설명된 실시예에 대한 어떠한 변동, 변형, 추가 및 개선도 가능하다. 이들 변동, 변형, 추가 및 개선은 이하 특허청구범위 내에 있는 본 발명의 범위 내에 있을 수 있다.

## 도면

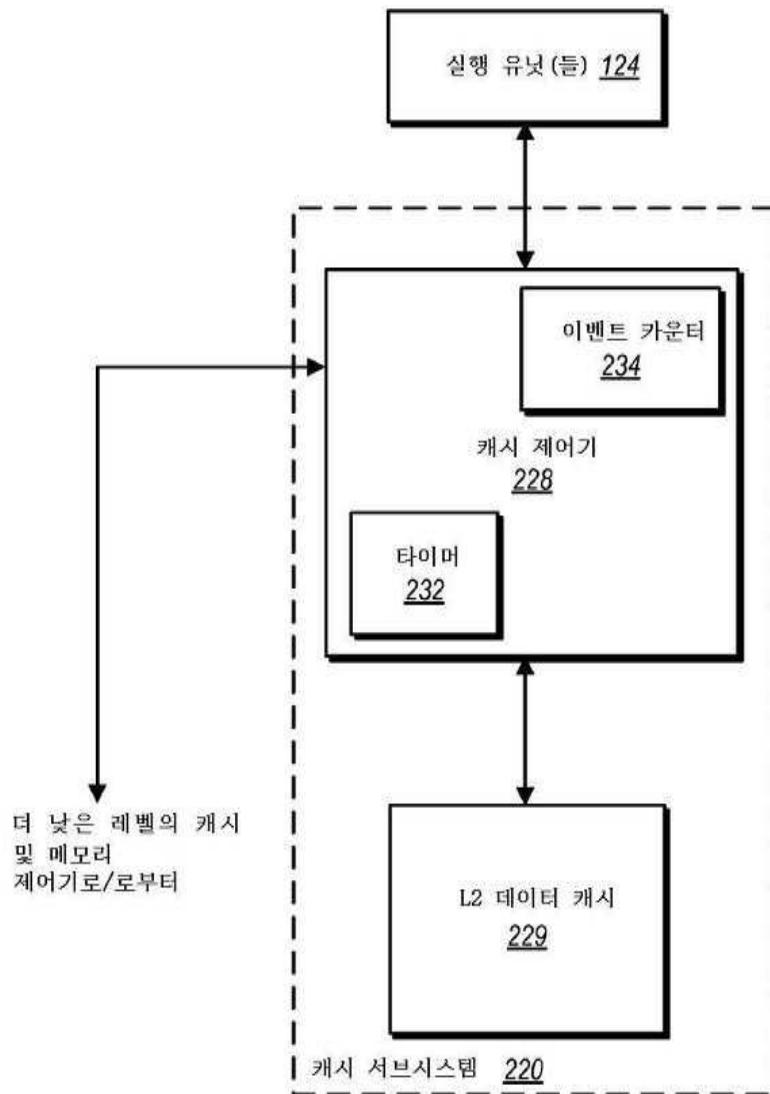
## 도면1



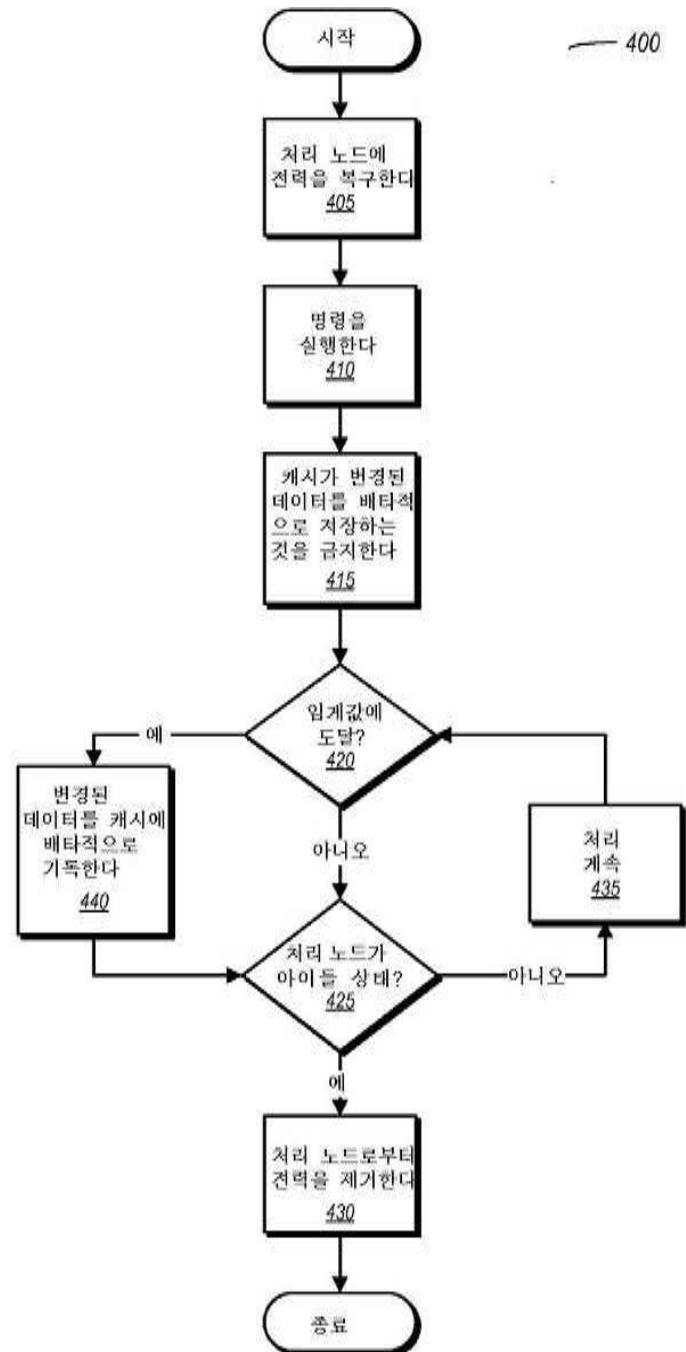
## 도면2



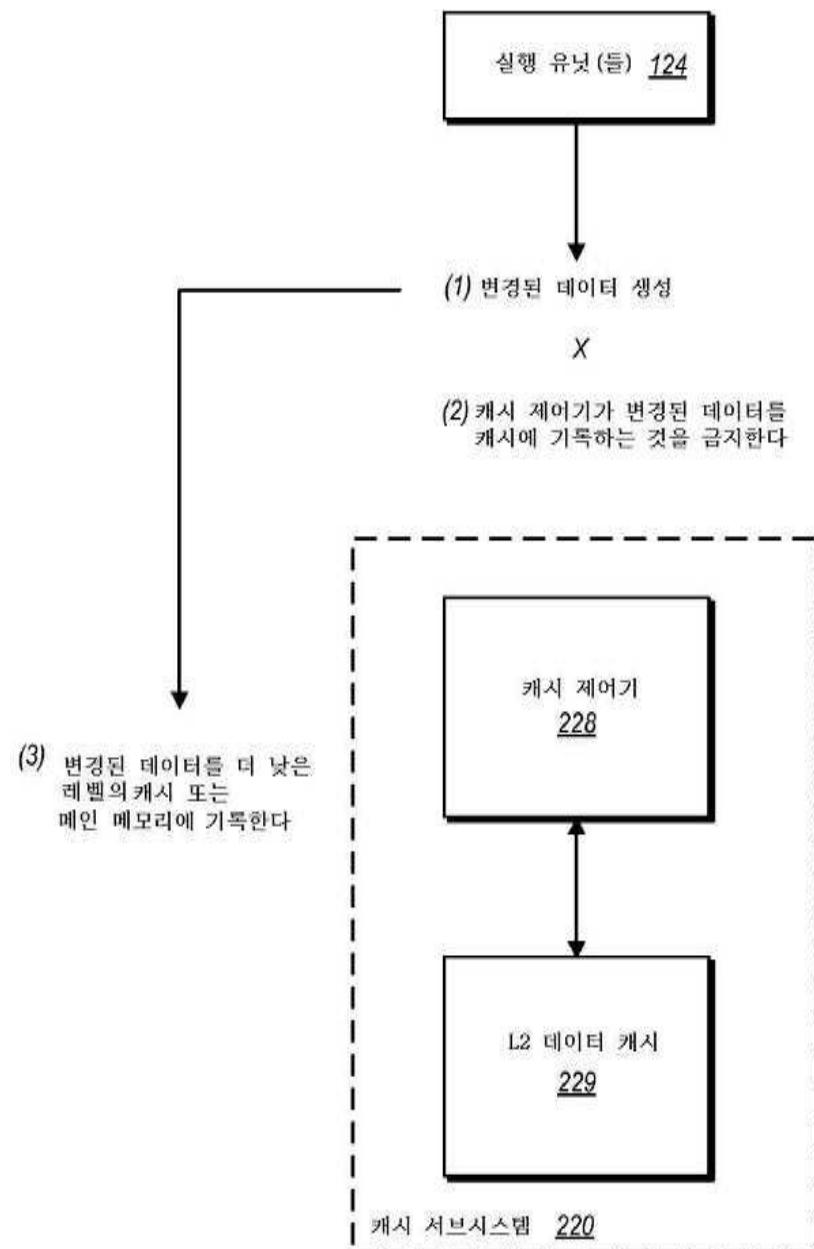
## 도면3



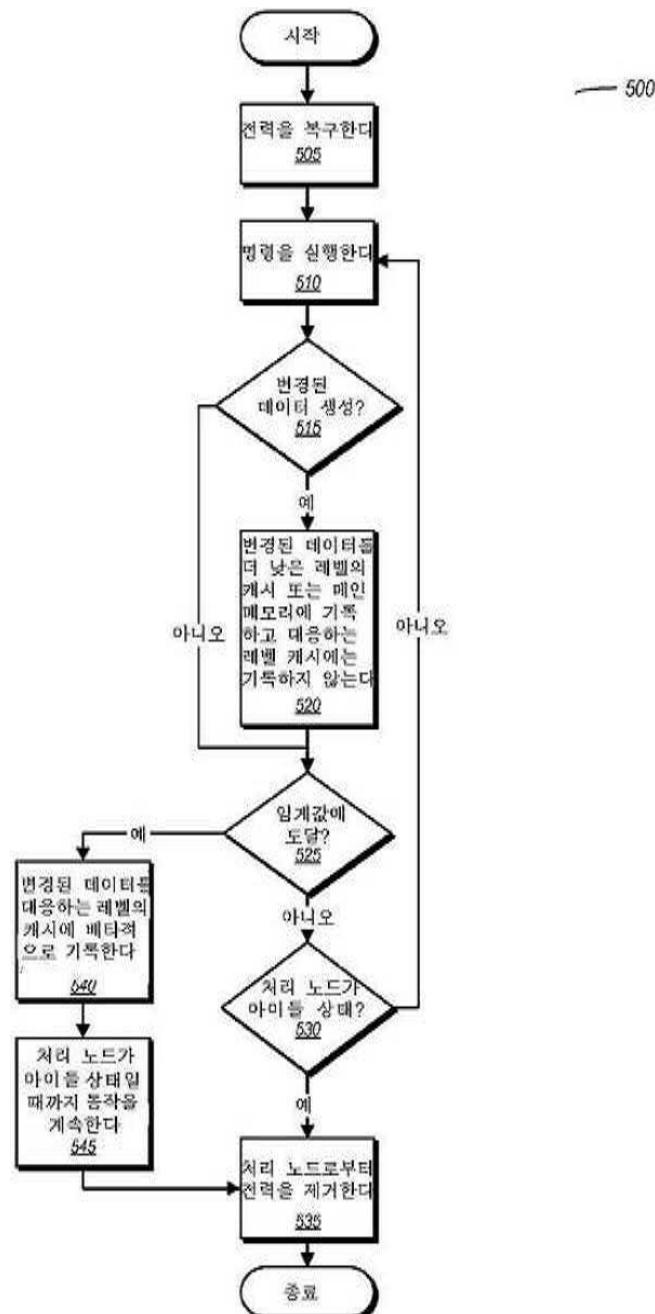
## 도면4



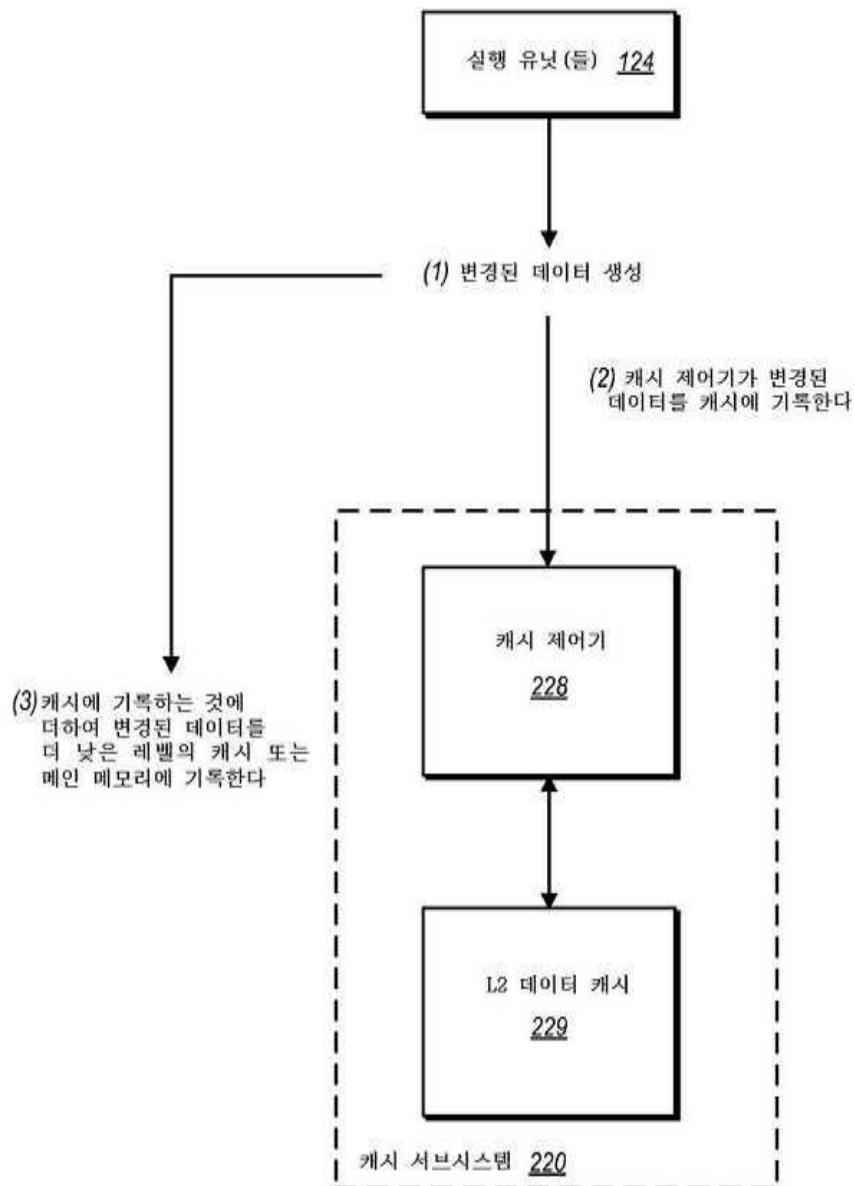
## 도면5



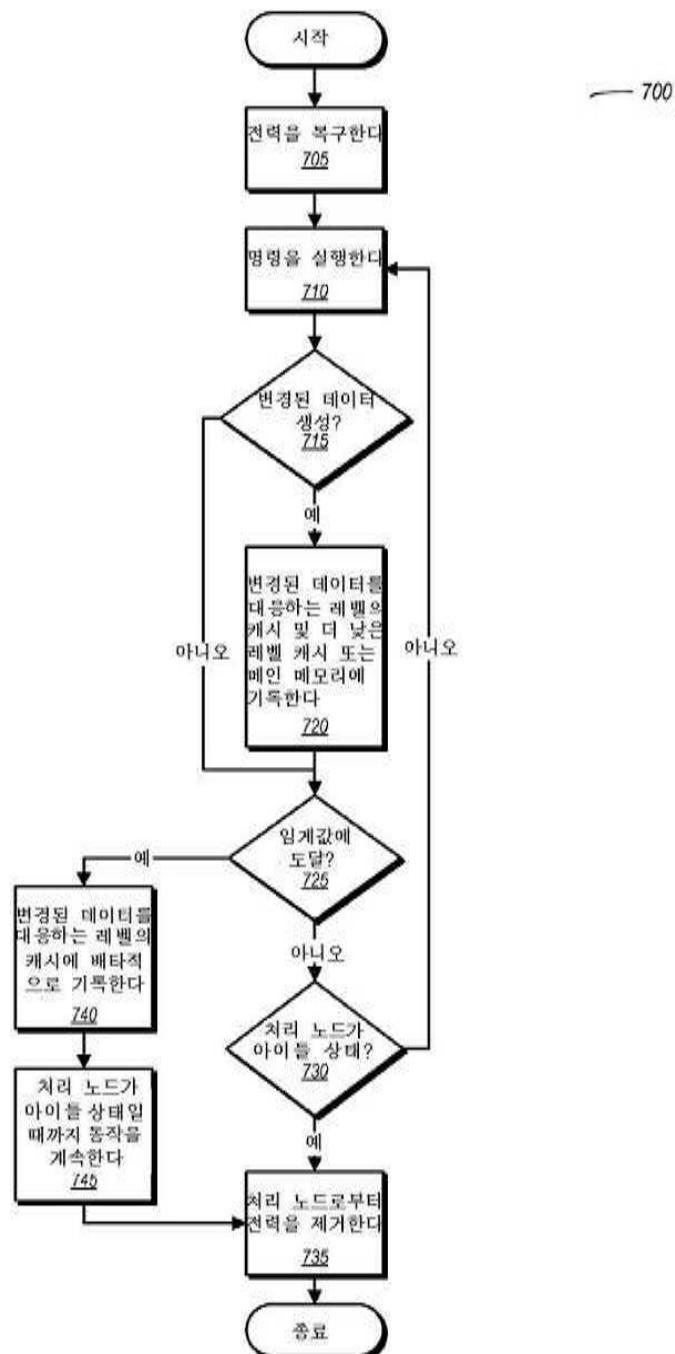
## 도면6



## 도면7



## 도면8



도면9

