



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2022년06월28일  
(11) 등록번호 10-2414157  
(24) 등록일자 2022년06월23일

- (51) 국제특허분류(Int. Cl.)  
G06F 12/0811 (2016.01) G06F 12/0846 (2016.01)  
G06F 12/0871 (2016.01)
- (52) CPC특허분류  
G06F 12/0811 (2013.01)  
G06F 12/0848 (2013.01)
- (21) 출원번호 10-2018-7017114
- (22) 출원일자(국제) 2016년09월20일  
심사청구일자 2021년09월17일
- (85) 번역문제출일자 2018년06월15일
- (65) 공개번호 10-2018-0085752
- (43) 공개일자 2018년07월27일
- (86) 국제출원번호 PCT/US2016/052722
- (87) 국제공개번호 WO 2017/105575  
국제공개일자 2017년06월22일
- (30) 우선권주장  
14/973,448 2015년12월17일 미국(US)
- (56) 선행기술조사문헌  
US20070143546 A1  
US20150205724 A1  
US20120144092 A1

- (73) 특허권자  
어드밴스드 마이크로 디바이시즈, 인코포레이티드  
미국 캘리포니아 95054 산타 클라라 어거스틴 드 라이브 2485
- (72) 발명자  
로호 가브리엘 에이치.  
미국 워싱턴 98007 벨뷰 엔이 12번 스트리트 15115
- (74) 대리인  
박장원

전체 청구항 수 : 총 19 항

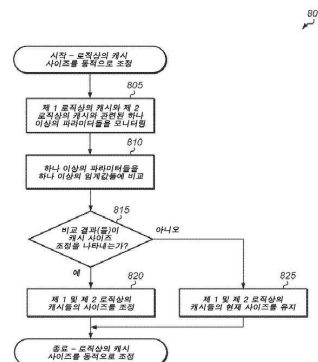
심사관 : 안지현

(54) 발명의 명칭 하이브리드 캐시

(57) 요약

하이브리드 캐시를 구현하기 위한 시스템들, 장치들, 및 방법들. 프로세서는 L2 캐시의 사이즈 및 L3 캐시의 사이즈를 프로세서가 동적으로 조정하는 것을 허용하는 하이브리드 L2/L3 캐시를 포함할 수 있다. 일부 실시예들에서, 프로세서는 멀티-코어 프로세서일 수 있고 코어들에 의한 사용을 위해 로직상의 L2 캐시 및 로직상의 L3 캐시로 파티션된 단일 캐시일 수 있다. 일 실시예에서, 프로세서는 로직상의 L2 및 L3 캐시들의 캐시 히트율을 추적할 수 있고 캐시 히트율에 기초하여 로직상의 L2 및 L3 캐시의 사이즈들을 조절한다. 다른 실시예에서, 프로세서는 어떤 애플리케이션이 프로세서에 의해 현재 실행되고 있는지에 기초하여 로직상의 L2 및 L3 캐시들의 사이즈들을 조절할 수 있다.

대표도 - 도8



(52) CPC특허분류  
*G06F 12/0871* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

프로세서에 있어서:

제 1 코어에 결합된 제 1 물리적 캐시;

제 2 코어에 결합된 제 2 물리적 캐시; 및

제어 유닛을 포함하고,

상기 제어 유닛은:

상기 제 1 물리적 캐시를 제 1 로직상의 캐시(logical cache) 및 제 2 로직상의 캐시의 제 1 부분으로 파티션(partition)하고,

상기 제 2 물리적 캐시를 제 3 로직상의 캐시 및 상기 제 2 로직상의 캐시의 제 2 부분으로 파티션하고 - 상기 제 1 로직상의 캐시 및 상기 제 3 로직상의 캐시는 캐시 계층(cache hierarchy)의 제 1 레벨을 나타내고, 상기 제 2 로직상의 캐시는 상기 캐시 계층의 제 2 레벨을 나타내며, 그리고 상기 제 1 로직상의 캐시 및 상기 제 2 로직상의 캐시 각각은 상위 레벨 메모리로부터 검색된 데이터를 저장하도록 구성됨 -,

상기 제 2 로직상의 캐시의 제 1 웨이(way)를 상기 제 1 로직상의 캐시에 재할당하고,

상기 제 2 로직상의 캐시의 제 1 웨이에 저장된 제 1 캐시 라인을 식별하고, 그리고

상기 제 1 웨이를 상기 제 2 로직상의 캐시로부터 상기 제 1 로직상의 캐시로 재할당한 것에 응답하여 상기 제 1 캐시 라인을 무효로하도록(invalidate) 구성되는,

프로세서.

#### 청구항 2

청구항 1에 있어서,

상기 제 1 로직상의 캐시는 상기 제 1 코어의 사적인(private) 캐시이고,

상기 제 3 로직상의 캐시는 상기 제 2 코어의 사적인 캐시이고, 그리고

상기 제 2 로직상의 캐시는 상기 제 1 코어 및 상기 제 2 코어 모두에 의해 공유되는,

프로세서.

#### 청구항 3

청구항 1에 있어서,

상기 제어 유닛은:

어드레스를 포함하는 요청을 수신하는 것에 응답하여, 상기 제 1 로직상의 캐시에서 룩업(lookup)을 수행하고;

상기 요청이 상기 제 1 로직상의 캐시에서 누락(miss)되었다고 결정한 것에 대한 응답으로:

상기 어드레스가 상기 제 2 로직상의 캐시의 제 1 부분을 선택한다고 결정하는 것에 응답하여 상기 제 2 로직상의 캐시의 제 1 부분에서 룩업을 수행하고, 그리고

상기 어드레스가 상기 제 2 로직상의 캐시의 제 2 부분을 선택한다고 결정하는 것에 응답하여 상기 제 2 로직상의 캐시의 제 2 부분에서 룩업을 수행하도록 더 구성됨,

프로세서.

**청구항 4**

청구항 1에 있어서,

상기 프로세서는, 상기 제 1 로직상의 캐시와 상기 제 3 로직상의 캐시 사이의 캐시 데이터의 캐시 히트율(cache hit rate) 및 마이그레이션 비율(migration rate) 중 하나 이상에 응답하여 상기 제 1 로직상의 캐시, 상기 제 2 로직상의 캐시 및 상기 제 3 로직상의 캐시 중 하나 이상의 사이즈를 변경하도록 더 구성된,

프로세서.

**청구항 5**

청구항 1에 있어서,

상기 프로세서는:

상기 제 2 로직상의 캐시의 원래의 위치를 타겟으로 하는 제 1 메모리 어드레스 상의 누락을 검출하는 것에 응답하여 제 1 요청을 메모리로 발송하고;

상기 제 1 요청을 메모리로 발송한 다음에 그리고 메모리로부터 제 1 필 데이터(fill data)를 수신하기 전에 상기 제 2 로직상의 캐시로부터 상기 제 1 로직상의 캐시로 상기 원래의 위치를 재할당하고, 상기 제 1 필 데이터는 상기 제 1 요청에 대응하고;

메모리로부터 상기 제 1 필 데이터를 수신하고;

상기 제 1 메모리 어드레스가 상기 제 2 로직상의 캐시의 원래의 위치와 상이한 상기 제 2 로직상의 캐시의 새로운 위치에 재매핑되었음을 결정하고; 그리고

상기 제 1 필 데이터를 상기 제 2 로직상의 캐시의 새로운 위치에 저장하도록 더 구성되는,

프로세서.

**청구항 6**

청구항 1에 있어서,

상기 프로세서는 제 1 애플리케이션 실행으로부터 제 2 애플리케이션 실행으로의 스위칭에 응답하여 상기 제 1 로직상의 캐시의 스토리지 용량을 축소시키고 상기 제 2 로직상의 캐시의 스토리지 용량을 증가시키도록 구성된,

프로세서.

**청구항 7**

하이브리드 캐시를 구현하기 위한 방법에 있어서,

프로세서의 제어 유닛에 의해, 제 1 코어에 결합된 제 1 물리적 캐시를 제 1 로직상의 캐시 및 제 2 로직상의 캐시의 제 1 부분으로 파티션하는 단계와;

상기 제어 유닛에 의해, 제 2 코어에 결합된 제 2 물리적 캐시를 제 3 로직상의 캐시 및 상기 제 2 로직상의 캐시의 제 2 부분으로 파티션하는 단계 - 상기 제 1 로직상의 캐시 및 상기 제 3 로직상의 캐시는 캐시 계층의 제 1 레벨을 나타내고, 상기 제 2 로직상의 캐시는 상기 캐시 계층의 제 2 레벨을 나타내며, 그리고 상기 제 1 로직상의 캐시 및 상기 제 2 로직상의 캐시 각각은 상위 레벨 메모리로부터 검색된 데이터를 저장하도록 구성됨 - 와;

상기 제어 유닛에 의해, 상기 제 2 로직상의 캐시의 제 1 웨이를 상기 제 1 로직상의 캐시에 재할당하는 단계와;

상기 제어 유닛에 의해, 상기 제 2 로직상의 캐시의 제 1 웨이에 저장된 제 1 캐시 라인을 식별하는 단계와; 그리고

상기 제 1 웨이를 상기 제 2 로직상의 캐시로부터 상기 제 1 로직상의 캐시로 재할당한 것에 응답하여 상기 제 1 캐시 라인을, 상기 제어 유닛에 의해, 무효로하는 단계를 포함하는,

하이브리드 캐시를 구현하기 위한 방법.

**청구항 8**

청구항 7에 있어서,

상기 제 1 로직상의 캐시는 상기 제 1 코어의 사적인 캐시이고,

상기 제 3 로직상의 캐시는 상기 제 2 코어의 사적인 캐시이고, 그리고

상기 제 2 로직상의 캐시는 상기 제 1 코어 및 상기 제 2 코어 모두에 의해 공유되는,

하이브리드 캐시를 구현하기 위한 방법.

**청구항 9**

청구항 7에 있어서,

어드레스를 포함하는 요청을 수신하는 것에 응답하여, 상기 제 1 로직상의 캐시에서 록업을, 상기 제어 유닛에 의해, 수행하는 단계와;

상기 요청이 상기 제 1 로직상의 캐시에서 누락되었다고 결정한 것에 대한 응답으로:

상기 어드레스가 상기 제 2 로직상의 캐시의 제 1 부분을 선택한다고 결정하는 것에 응답하여 상기 제 2 로직상의 캐시의 제 1 부분에서 록업을, 상기 제어 유닛에 의해, 수행하는 단계; 및

상기 어드레스가 상기 제 2 로직상의 캐시의 제 2 부분을 선택한다고 결정하는 것에 응답하여 상기 제 2 로직상의 캐시의 제 2 부분에서 록업을, 상기 제어 유닛에 의해, 수행하는 단계를 더 포함하는,

하이브리드 캐시를 구현하기 위한 방법.

**청구항 10**

청구항 7에 있어서,

상기 제 1 로직상의 캐시와 상기 제 3 로직상의 캐시 사이의 캐시 데이터의 캐시 히트율 및 마이그레이션 비율 중 하나 이상에 응답하여 상기 제 1 로직상의 캐시, 상기 제 2 로직상의 캐시 및 상기 제 3 로직상의 캐시 중 하나 이상의 사이즈를, 상기 프로세서에 의해, 변경하는 단계를 더 포함하는,

하이브리드 캐시를 구현하기 위한 방법.

**청구항 11**

청구항 7에 있어서,

상기 제 2 로직상의 캐시의 원래의 위치를 타겟으로 하는 제 1 메모리 어드레스 상의 누락을 검출하는 것에 응답하여 제 1 요청을, 상기 프로세서에 의해, 메모리로 발송하는 단계와;

상기 제 1 요청을 메모리로 발송한 다음에 그리고 메모리로부터 제 1 필 데이터를 수신하기 전에 상기 제 2 로직상의 캐시로부터 상기 제 1 로직상의 캐시로 상기 원래의 위치를, 상기 프로세서에 의해, 재할당하는 단계 - 상기 제 1 필 데이터는 상기 제 1 요청에 대응함 - 와;

상기 프로세서에 의해, 메모리로부터 상기 제 1 필 데이터를 수신하는 단계와;

상기 제 1 메모리 어드레스가 상기 제 2 로직상의 캐시의 원래의 위치와 상이한 상기 제 2 로직상의 캐시의 새로운 위치에 재매핑되었음을, 상기 프로세서에 의해, 결정하는 단계와; 그리고

상기 프로세서에 의해, 상기 제 1 필 데이터를 상기 제 2 로직상의 캐시의 새로운 위치에 저장하는 단계를 더 포함하는,

하이브리드 캐시를 구현하기 위한 방법.

**청구항 12**

청구항 7에 있어서,

제 1 애플리케이션 실행으로부터 제 2 애플리케이션 실행으로의 스위칭에 응답하여, 상기 프로세서에 의해, 상기 제 1 로직상의 캐시의 스토리지 용량을 축소시키고 상기 제 2 로직상의 캐시의 스토리지 용량을 증가시키는 단계를 더 포함하는,

하이브리드 캐시를 구현하기 위한 방법.

**청구항 13**

시스템에 있어서,

메모리; 및

프로세서를 포함하고,

상기 프로세서는:

제 1 코어에 결합된 제 1 물리적 캐시;

제 2 코어에 결합된 제 2 물리적 캐시; 및

제어 유닛을 포함하며,

상기 제어 유닛은:

상기 제 1 물리적 캐시를 제 1 로직상의 캐시 및 제 2 로직상의 캐시의 제 1 부분으로 파티션하고,

제 2 코어에 결합된 제 2 물리적 캐시를 제 3 로직상의 캐시 및 상기 제 2 로직상의 캐시의 제 2 부분으로 파티션하고 - 상기 제 1 로직상의 캐시 및 상기 제 3 로직상의 캐시는 캐시 계층의 제 1 레벨을 나타내고, 상기 제 2 로직상의 캐시는 상기 캐시 계층의 제 2 레벨을 나타내며, 그리고 상기 제 1 로직상의 캐시 및 상기 제 2 로직상의 캐시 각각은 상기 메모리로부터 검색된 데이터를 저장하도록 구성됨 -,

상기 제 2 로직상의 캐시의 제 1 웨이를 상기 제 1 로직상의 캐시에 재할당하고,

상기 제 2 로직상의 캐시의 제 1 웨이에 저장된 제 1 캐시 라인을 식별하고, 그리고

상기 제 1 웨이를 상기 제 2 로직상의 캐시로부터 상기 제 1 로직상의 캐시로 재할당한 것에 응답하여 상기 제 1 캐시 라인을 무효화하도록 구성되는,

시스템.

**청구항 14**

청구항 13에 있어서,

상기 제 1 로직상의 캐시는 상기 제 1 코어의 사적인 캐시이고,

상기 제 3 로직상의 캐시는 상기 제 2 코어의 사적인 캐시이고, 그리고

상기 제 2 로직상의 캐시는 상기 제 1 코어 및 상기 제 2 코어 모두에 의해 공유되는,

시스템.

**청구항 15**

청구항 13에 있어서,

상기 제어 유닛은:

어드레스를 포함하는 요청을 수신하는 것에 응답하여, 상기 제 1 로직상의 캐시에서 록업을 수행하고;

상기 요청이 상기 제 1 로직상의 캐시에서 누락되었다고 결정한 것에 대한 응답으로:

상기 어드레스가 상기 제 2 로직상의 캐시의 제 1 부분을 선택한다고 결정하는 것에 응답하여 상기 제 2 로직상의 캐시의 제 1 부분에서 록업을 수행하고, 그리고

상기 어드레스가 상기 제 2 로직상의 캐시의 제 2 부분을 선택한다고 결정하는 것에 응답하여 상기 제 2

로직상의 캐시의 제 2 부분에서 룩업을 수행하도록 더 구성된,  
시스템.

**청구항 16**

청구항 13에 있어서,  
상기 프로세서는, 상기 제 1 로직상의 캐시와 상기 제 3 로직상의 캐시 사이의 캐시 데이터의 캐시 히트율 및 마이그레이션 비율 중 하나 이상에 응답하여 상기 제 1 로직상의 캐시, 상기 제 2 로직상의 캐시 및 상기 제 3 로직상의 캐시 중 하나 이상의 사이즈를 변경하도록 더 구성된,  
시스템.

**청구항 17**

청구항 13에 있어서,  
상기 프로세서는:  
상기 제 2 로직상의 캐시의 원래의 위치를 타겟으로 하는 제 1 메모리 어드레스 상의 누락을 검출하는 것에 응답하여 제 1 요청을 메모리로 발송하고;  
상기 제 1 요청을 메모리로 발송한 다음에 그리고 메모리로부터 제 1 필 데이터를 수신하기 전에 상기 제 2 로직상의 캐시로부터 상기 제 1 로직상의 캐시로 상기 원래의 위치를 재할당하고, 상기 제 1 필 데이터는 상기 제 1 요청에 대응하고;  
메모리로부터 상기 제 1 필 데이터를 수신하고;  
상기 제 1 메모리 어드레스가 상기 제 2 로직상의 캐시의 원래의 위치와 상이한 상기 제 2 로직상의 캐시의 새로운 위치에 재매핑되었음을 결정하고; 그리고  
상기 제 1 필 데이터를 상기 제 2 로직상의 캐시의 새로운 위치에 저장하도록 더 구성되는,  
시스템.

**청구항 18**

청구항 13에 있어서,  
상기 프로세서는 제 1 애플리케이션 실행으로부터 제 2 애플리케이션 실행으로의 스위칭에 응답하여 상기 제 1 로직상의 캐시의 스토리지 용량을 축소시키고 상기 제 2 로직상의 캐시의 스토리지 용량을 증가시키도록 구성된,  
시스템.

**청구항 19**

청구항 3에 있어서,  
상기 어드레스는:  
상기 제 1 로직상의 캐시의 적어도 일부를 식별하는 하나 이상의 비트; 및  
상기 제 2 로직상의 캐시의 제 1 부분 및 상기 제 2 로직상의 캐시의 제 2 부분 중 하나를 식별하는 하나 이상의 비트를 포함하는  
프로세서.

**청구항 20**

삭제

**발명의 설명**

**기술 분야**

[0001] 본 출원에서 설명된 실시예들은 프로세서들에 관한 것으로 보다 상세하게는, 프로세서들에 캐시들을 이용하는 것에 관한 것이다.

**배경 기술**

[0002] 많은 최신 컴퓨팅 디바이스들 (예를 들어, 랩탑/데스크탑 컴퓨터들, 스마트 폰들, 셋-탑 박스들, 기기들, 등)은 하나 이상의 캐시들을 갖는 프로세싱 서브시스템들을 포함한다. 캐시들은 저장된 데이터에 대한 더 빠른 액세스를 가능하게 하기 위해서 컴퓨팅 디바이스내 메모리 계층의 상위 레벨들 (즉, 다른, 더 큰 캐시들 및/또는 메모리들)로부터 검색된 데이터를 저장하기 위해 사용될 수 있는 프로세싱 서브시스템내에 또는 그 근처에 일반적으로 더 적고, 빠른-액세스 메모리 회로들이다.

[0003] 일반적으로, 컴퓨터 시스템의 메인 메모리는 세분화(granularity)의 페이지 레벨에서 메모리 구조화를 가진다. 비록 임의의 다른 사이즈 페이지가 특정 구현예에 대하여 정의될 수 있지만 전형적으로, 페이지는 네 개의 킬로바이트 (KB) 페이지일 수 있다. 캐시 메모리 구조화(cache memory organization)는 일반적으로 세분화의 캐시 라인 레벨에 있다. 캐시 라인(cache line)은 연결 메모리 위치들에 저장된 바이트들의 세트이고, 이는 코히런스(coherency) 용도를 위한 유닛으로 취급된다. 본 출원에서 사용되는, 각각의 용어들 "캐시 라인(cache line)" 및 "캐시 블록(cache block)"은 상호변경 가능하다. 캐시 라인 내 바이트의 수는 디자인 선택에 따라 변화할 수 있고, 임의의 사이즈를 가질 수 있다.

[0004] 멀티-레벨 캐시 계층은 흔히 현대 프로세서들을 위해 채용된다. 예를 들어, 멀티-코어 프로세서에 대하여, 각각의 코어는 데이터 및/또는 지시들을 저장하기 위해 레벨-원 (L1) 캐시를 포함할 수 있다. 멀티-레벨 캐시 계층은 또한 전형적으로 코어들 가운데 공유되는 각 코어 마다 사적인(private) 레벨-투 (L2) 캐시 및 더 큰 레벨-쓰리 (L3) 캐시를 포함할 것이다. 프로세서는 프로세스하기 위한 상이한 데이터의 양 및 상이한 데이터 액세스 패턴들을 갖는 많은 상이한 유형들의 애플리케이션들을 실행시킬 수 있다. 일부 애플리케이션들은 큰 데이터의 양을 프로세스 할 수 있지만 다른 애플리케이션들은 데이터의 동일 부분들을 점진적으로 변경함으로써 작은 데이터의 양으로 작업할 수 있다. 그러나, 통상의 멀티-레벨 캐시 계층을 이용하는 것은 더 작은 최적의 효율에서 프로세서에 의해 실행되는 일부 애플리케이션들로 귀결될 수 있다.

**발명의 내용**

**해결하려는 과제**

**과제의 해결 수단**

[0005] 하이브리드 캐시를 구현하기 위한 시스템들, 장치들, 및 방법들이 고려된다.

[0006] 다양한 실시예들에서, 멀티-코어 프로세서는 멀티-레벨 캐시 계층을 포함할 수 있다. 상기 프로세서는 캐시 계층의 제 1 및 제 2 레벨을 각각의 상기 코어들에 의한 사용을 위해 논리적으로 파티션되는 단일 캐시로 함께 효율적으로 결합하는 캐시 구조화(cacheorganization)를 가질 수 있다. 다양한 실시예들에서, 상기 캐시는 레벨 투 (L2) 및 레벨 쓰리 (L3) 캐시들로 논리적으로 파티션되고, 각각의 코어는 소정의 로직상의 L2 및 L3 캐시에 할당된다. 다양한 실시예들에서, 상기 캐시 계층의 상기 제 1 및 제 2 레벨들의 사이즈들은 정적으로 설정 또는 동적으로 조절될 수 있다.

[0007] 예를 들어, 상기 로직상의 캐시들의 사이즈들은 생산할 때에 고정될 수 있다. 다른 실시예들에서, 로직상의 캐시들의 사이즈들은 현재 동작 상태들에 기초하여 동적으로 조절될 수 있다. 예를 들어, 프로세서는 더 작은 L2 캐시 희생으로 더 큰 L3 캐시로부터 이로울 수 있는 제 1 애플리케이션을 실행할 수 있다. 상기 제 1 애플리케이션을 실행할 때, 상기 프로세서는 상기 L2 캐시(들)의 사이즈를 줄일 수 있고 상기 L3 캐시의 사이즈를 증가시킬 수 있다. 이어서, 상기 프로세서는 상기 제 1 애플리케이션 실행으로부터 제 2 애플리케이션 실행으로 스위칭될 수 있고 상기 프로세서는 상기 제 2 애플리케이션은 더 작은 L3 캐시 희생으로 더 큰 L2 캐시로부터 이로울 것이라는 표시를 수신하거나 또는 결정할 수 있다. 따라서, 상기 프로세서는 상기 L3 캐시의 사이즈를 줄일 수 있고 상기 L2 캐시(들)의 사이즈(들)을 증가시킬 수 있다.

[0008] 추가 실시예들에서, 상기 프로세서는 상기 캐시 계층의 상이한 레벨들과 관련된 다양한 파라미터들을 모니터링할 수 있다. 이 실시예에서, 상기 프로세서는 상기 모니터링된 파라미터들에 기초하여 상기 캐시 계층의 두개 이상의 레벨들의 사이즈들을 동적으로 조절할 수 있다. 이들 파라미터들은 제 1 캐시의 캐시 히트율, 제 2 캐시의 캐시 히트율, 및 별개의 프로세서 코어들에 대응하는 상이한 제 1 캐시(들)간에 캐시 라인들의 마이그레이션 비율을 포함할 수 있다. 다양한 실시예들에서, 상기 프로세서는 하나 이상의 파라미터들을 하나 이상의 임계값들에 비교할 수 있다. 상기 비교(들)의 결과(들)에 기초하여, 상기 프로세서는 상기 제 1 캐시 (예를 들어, L2 캐시)의 사이즈를 줄일 수 있고 상기 제 2 캐시 (예를 들어, L3 캐시)의 사이즈를 증가시킬 수 있고, 상기 프로세서는 상기 제 1 캐시의 사이즈를 증가시킬 수 있고 상기 제 2 캐시의 사이즈를 줄일 수 있거나, 또는 상기 프로세서는 상기 제 1 및 제 2 캐시들의 현재 사이즈들을 유지할 수 있다.

[0009] 이들 및 다른 특징부들 및 장점들이 이하의 본 출원에 제공된 접근법들의 상세한 설명의 면에서 기술 분야에서 통상의 기술자들에 명확해질 것이다.

**도면의 간단한 설명**

[0010] 방법들 및 메커니즘들의 상기 및 추가 장점들이 이하의 설명과 함께 첨부한 도면들과 함께 이하의 설명을 참조하여 더 잘 이해될 것이다:

- 도 1은 컴퓨팅 시스템의 일 실시예의 블록 다이어그램이다.
- 도 2는 로직상의 캐시의 일 실시예의 블록 다이어그램이다.
- 도 3은 제어 유닛의 일 실시예의 블록 다이어그램이다.
- 도 4는 프로세서의 일 실시예의 블록 다이어그램이다.
- 도 5는 프로세서의 다른 실시예의 블록 다이어그램이다.
- 도 6은 세개의 상이한 시간 지점들에서 로직상의 캐시의 일 실시예의 블록 다이어그램이다.
- 도 7은 다수의 캐시 라인들의 태그 비트(tag bit)들을 보여주는 다이어그램들을 예시한다.
- 도 8은 캐시 사이즈들을 동적으로 조정하기 위한 방법의 일 실시예를 예시하는 일반화된 흐름도이다.
- 도 9는 캐시 사이즈들을 동적으로 조정하기 위한 방법의 다른 실시예를 예시하는 일반화된 흐름도이다.
- 도 10은 캐시 사이즈들을 조정할 때를 결정하기 위한 방법의 일 실시예를 예시하는 일반화된 흐름도이다.
- 도 11은 캐시 사이즈들을 조정할 때를 결정하기 위한 방법의 다른 실시예를 예시하는 일반화된 흐름도이다.
- 도 12은 캐시 계층에 마이그레이션 비율(migration rate)를 모니터링하기 위한 방법의 일 실시예를 예시하는 일반화된 흐름도이다.
- 도 13은 인접한 캐시 레벨들 사이에서 상이한 용량 할당(capacity allocation)을 활용하기 위한 방법의 일 실시예를 예시하는 일반화된 흐름도이다.
- 도 14는 필 데이터(fill data)를 포워딩(forward)하기 위한 방법의 일 실시예를 예시하는 일반화된 흐름도이다.
- 도 15는 로직상의 캐시의 부분을 재할당하기 위한 방법의 일 실시예를 예시하는 일반화된 흐름도이다.

**발명을 실시하기 위한 구체적인 내용**

[0011] 이하의 설명에서, 본 출원에 제시된 방법들 및 메커니즘들의 완벽한 이해를 제공하기 위한 많은 특정 세부사항들이 개시된다. 그러나, 관련 기술 분야의 통상의 기술자는 이들 특정 세부사항들 없이 다양한 실시예들이 실행될 수 있다는 것을 인식하여야 한다. 일부 경우들에서, 주지의 구조들, 컴포넌트들, 신호들, 컴퓨터 프로그램 지시들, 및 기술들이 본 출원에서 설명된 접근법들을 애매하게 하는 것을 피하기 위해 상세하게 도시되지 않는다. 예시의 명확성 및 단순화를 위하여, 도면들에 도시된 엘리먼트들은 반드시 축척에 맞도록 도시될 필요가 없다는 것이 인식될 것이다. 예를 들어, 엘리먼트들의 일부의 치수는 다른 엘리먼트들에 비하여 과장될 것이다.

[0012] 이제 도 1을 참조하여, 컴퓨팅 시스템(100)의 일 실시예를 예시하는 블록 다이어그램이 도시된다. 시스템 (100)은 프로세서 (105) 및 메모리 (130)를 포함할 수 있다. 시스템 (100)은 도면을 모호하게 하는 것을 피하기 위해 도 1에 도시되지 않은 다른 컴포넌트들 (예를 들어, 디스플레이, 입력/출력 (I/O) 인터페이스, 네트워크 인

터페이스)을 또한 포함한다. 실시예에 따라, 시스템 (100)은 모바일 디바이스 (예를 들어, 스마트폰, 태블릿), 컴퓨터, 텔레비전, 엔터테인먼트 디바이스, 서버, 등을 포함하는 컴퓨팅 시스템들의 임의의 다양한 유형들일 수 있다. 프로세서 (105)는 컴퓨터의 동작들을 수행하고 지시들을 실행하기 위한 프로세서의 임의의 유형 (예를 들어, 그래픽스 프로세싱 유닛 (GPU : graphics processing unit), 중앙 프로세싱 유닛 (CPU : central processing unit), 가속 프로세싱 유닛 (APU : accelerated processing unit))의 예시이다.

[0013] 일 실시예에서, 프로세서 (105)는 도 1에 도시된 바와 같이 네개의 코어들 (110A-D) 및 캐시들 (115A-D)을 갖는 멀티-코어 프로세서일 수 있다. 다른 실시예들에서, 프로세서 (105)은 다른 수의 코어들을 포함할 수 있다. 다양한 실시예들에서, 이하에서 더 상세하게 논의될 것 처럼, 각각의 캐시 (115A-D)는 로직상의 캐시(logical cache)들로 파티션(partition)되는 단일의 물리적 캐시일 수 있다. 각각의 코어 (110A-D)는 지시들 및/또는 데이터를 저장하기 위한 레벨-원 (L1) 캐시 (미도시)를 포함할 수 있다. 각각의 코어 (110A-D)는 대응하는 물리적 캐시 (115A-D)를 또한 포함할 수 있고 및/또는 대응하는 물리적 캐시에 결합될 수 있다. 캐시들 (115A-D)의 각각은 캐시된 데이터 및 지시들을 저장하기 위해 사용되는 메모리 회로들을 포함할 수 있다. 예를 들어, 캐시들 (115A-D)은 정적 랜덤 액세스 메모리 (SRAM :static random access memory), 내장 동적 랜덤 액세스 메모리 (eDRAM :embedded dynamic random access memory), DRAM, 더블 데이터 레이트 동기식 DRAM (DDRSDRAM :double data rate synchronous DRAM), 및/또는 다른 메모리의 유형들 회로들 중 하나 이상을 포함할 수 있다.

[0014] 일 실시예에서, 각각의 캐시 (115A-D)는 논리적으로 와 결합된 공유된 레벨-쓰리 (L3) 캐시의 구성 가능한 슬라이스(slice) (또는 부분)과 논리적으로 결합된 구성 가능한 레벨-투(two) (L2) 캐시를 포함할 수 있고, 프로세서 (105)는 이들 캐시 부분들의 사이즈들을 변경할 수 있다. 만약 프로세서 (105)가 소정의 양만큼 특정 캐시 (115A-D)의 L2 부분의 사이즈를 줄이면, 프로세서 (105)는 소정의 양 만큼 특정 캐시 (115A-D)의 L3 부분의 사이즈를 증가시킬 수 있다. 또는, 만약 프로세서 (105)가 소정의 양만큼 특정 캐시 (115A-D)의 L2 부분의 사이즈를 증가시키면, 프로세서는 소정의 양 만큼 특정 캐시 (115A-D)의 L3 부분의 사이즈를 축소시킬 수 있다. 다시 말해서, 각각의 캐시 (115A-D)는 L2 캐시와 L3 슬라이스 사이에서 분열되는 고정 스토리지 용량(fixed storage capacity)를 가진다. 따라서, 소정의 캐시 (115A-D)의 L2 캐시의 사이즈에 대한 임의의 변화는 소정의 캐시 (115A-D)의 L3 슬라이스의 사이즈에 대한 반대 변화를 일으킬 것이다.

[0015] L2 및 L3 캐시들을 동적으로 재사이즈(resize)하는 예가 도 1에 대하여 예시되고 설명되지만, 이것은 비 제한적인 예인 것을 의미하고, 프로세서는 캐시 계층의 임의의 두개 이상의 인접한 레벨들을 동적으로 재사이즈할 수 있다는 것이 이해되어야 한다. 유사하게, 본 개시 전체에서, L2 및 L3 캐시 부분들을 동적으로 재사이즈하는 예가 설명될 때, 이들 기술들은 캐시 계층의 다른 레벨들과 함께 이용될 수 있다는 것이 이해되어야 한다. 일부 실시예들에서, 캐시 계층의 세개 또는 그 이상 레벨들의 사이즈들이 본 출원에서 설명된 기술들을 이용하여 동적으로 조절될 수 있다.

[0016] 일 실시예에서, 프로세서 (105)는 각각의 캐시 (115A-D)의 L2 및 L3 부분들의 사이즈들을 동적으로 변화시킬 수 있다. 따라서, 이 실시예에서, L2 및 L3 캐시들의 스토리지 용량은 런 타임(run time)에서 변화될 수 있다. 일 실시예에서, L2 및 L3 캐시들의 구성은 소프트웨어 (예를 들어, 운영 체제(operating system), 하이퍼바이저, 애플리케이션들)에 의해 변형될 수 있다. 다른 실시예에서, 하드웨어 제어기 (예를 들어, 제어 유닛 (120))은 일부 목적들 (예를 들어, 성능 최대화)을 최적화하기 위해 L2 및 L3 할당들을 동적으로 재사이즈할 수 있고 시스템 성능 메트릭들 (예를 들어, L2 및 L3 캐시 히트율(hit rates))을 모니터링할 수 있다. 추가 실시예에서, L2 및 L3 캐시들의 구성은 소프트웨어 및 하드웨어의 조합에 의해 설정될 수 있다.

[0017] 다양한 실시예에서, 프로세서 (105)는 하나 이상의 상태들의 검출에 응답하여 각각의 캐시 (115A-D)의 L2 및 L3 부분들의 사이즈들을 동적으로 변화시킬 수 있다. 하나 이상의 상태(condition)들은 L2 및 L3 캐시들과 관련된 모니터링된 파라미터들 및/또는 프로세서 (105)에 의해 실행되는 애플리케이션들에 변화들에 의존될 수 있다. 프로세서 (105)는 L2 및 L3 캐시들의 히트율, 상이한 코어들 중에 상이한 L2 캐시들 간에 캐시 라인들의 마이그레이션 비율(migration rate), 및/또는 L2 및 L3 캐시들과 관련된 다른 파라미터들을 모니터링하도록 구성될 수 있다. 프로세서 (105)는 캐시들 (115A-D)의 L2 및 L3 부분들 간에 스토리지 용량을 동적으로 재할당할지 아닐지 여부를 결정하기 위해 이들 파라미터들을 하나 이상의 임계값들에 비교할 수 있다.

[0018] 다른 실시예에서, 프로세서 (105)는 실행되고 있는 애플리케이션들을 모니터링할 수 있고 실행되는 각각의 애플리케이션에 대하여 L2 및 L3 캐시 사이즈들의 최고의 할당을 결정할 수 있다. 일부 경우들에서, 프로세서 (105)는 캐시 사이즈들의 어떤 분배가 애플리케이션들을 보다 효율적으로 운용하는 것을 허용할 지를 결정하기 위해 애플리케이션들과 관련된 이력 행위 데이터(historical behavior data)를 유지할 수 있다. 다른 경우들에서,

각각의 애플리케이션은 애플리케이션에 대하여 선호되는 또는 최적의 캐시 사이즈들을 표시하는 미리 결정된 설정을 가질 수 있다. 예를 들어, 제 1 애플리케이션은 상대적으로 작은 L2 캐시들 및 상대적으로 큰 L3 캐시로부터 이룰 것이라는 것을 표시하는 설정을 제 1 애플리케이션은 포함할 수 있다. 따라서, 제 1 애플리케이션을 실행할 때, 프로세서 (105)는 각각의 코어의 L2 캐시의 사이즈를 줄일 수 있고 L3 슬라이스들의 사이즈를 증가시킬 수 있다. 그런 다음, 나중에, 제 2 애플리케이션은 상대적으로 큰 L2 캐시들 및 상대적으로 작은 L3 캐시로부터 이룰 것이라는 것을 표시하는 설정을 포함하는 제 2 애플리케이션을 프로세서는 실행시킬 수 있다. 따라서, 프로세서는 제 2 애플리케이션 실행에 응답하여 각각의 코어의 L2 캐시를 증가시킬 수 있고 그것들의 L3 슬라이스들의 사이즈를 줄일 수 있다.

[0019] 다른 실시예에서, 캐시들 (115A-D)의 L2 및 L3 캐시들의 사이즈들은 프로세서 (105)의 제작 또는 제조 프로세스의 일부로서 정적으로 조절될 수 있다. 이 실시예에서, L2 및 L3 캐시 구성은 제조 시간에 설정될 수 있다. 이것은 동일한 실리콘 디자인이 여러 가지 상이한 구성들로 배치되는 것을 허용한다. 이 실시예에서, 하나의 SoC(system on chip)가 다수의 SKU(stock keeping units)과 함께 이용될 수 있다. 예를 들어, 구성 퓨즈(configuration fuse)들이 캐시들 (115A-D)의 L2 및 L3 캐시들의 희망하는 구성을 설정하기 위해 사용될 수 있다. 대안적으로, 캐시들 (115A-D)의 L2 및 L3 캐시들의 사이즈들은 시스템 스타트업 동안에 캐시 사이즈들이 설정되도록 할 부트업(bootup) 동안에 BIOS가 관독하는 관독 전용 메모리 (ROM)의 구성 비트들에 저장될 수 있다.

[0020] 일 실시예에서, 제어 유닛 (120)은 각각의 캐시 (115A-D)가 가변-사이즈의 L2 캐시를 얼마나 많이 할당되는지 그리고 각각의 캐시 (115A-D)가 가변-사이즈의 L3 슬라이스를 위해 얼마나 많이 할당되는지를 결정하도록 구성될 수 있다. 비록 제어 유닛 (120) 및 메모리 제어기 (125)가 동일한 버스를 통하여 캐시들 (115A-D)에 연결되는 것으로 도시되지만, 이것은 단지 쉬운 예시를 위해 도시된 것이고, 제어 유닛 (120) 및 메모리 제어기 (125)는 캐시들 (115A-D)에 연결하기 위해 상이한 물리적 연결들을 활용할 수 있다는 것이 이해되어야 한다.

[0021] 메모리 (130)는 컴퓨팅 시스템 (100)의 "메인 메모리"를 형성하는 메모리 회로들을 포함할 수 있다. 메모리 (130)는 프로세서 (105)의 코어들 (110A-D)에 의한 사용을 위한 지시들 및 데이터를 저장하기 위해 사용될 수 있다. 메모리 (130)는 임의의 수, 용량, 및 유형의 메모리 디바이스들의 예시이다. 다양한 메모리의 유형들 (130)은 랜덤-액세스 메모리 (RAM : random-access memory), 정적 RAM (SRAM : static RAM), 동적 RAM (DRAM : dynamic RAM), 고속 CMOS, 고밀도 DRAM, eDRAM, 3D 적층된 메모리 (예를 들어, 적층된 DRAM), 인터포저기반의 통합 메모리, 멀티-칩 모듈들 (MCM : multi-chip modules), 마더보드 상에 오프-칩 DRAM, 비-휘발성 RAM (NVRAM : non-volatile RAM), 자기-광 스토리지 매체, 관독 전용 메모리 (ROM : read only memory), 동기식 DRAM (SDRAM : synchronous DRAM), 더블 데이터 레이트 SDRAM (DDRSDRAM : double data rate SDRAM), 상 변화 메모리, 스핀-전달 토크 자기 RAM, 멤리스터, 확대 데이터 출력 (EDO : extended data output) RAM, 램버스 (Rambus) RAM, 램버스DRAM, 소거형 프로그램 가능한 메모리 (EEPROM : erasable programmable memory), 고체-상태 메모리, 하드 디스크 드라이브, 광 스토리지 매체들, 등을 포함하는 시스템(100)에 이용될 수 있다.

[0022] 이제 도 2로 가서, 캐시(200)의 일 실시예의 블럭 다이어그램이 도시된다. 일 실시예에서, 각각의 캐시들 (115A-D) (도 1의)은 캐시 (200)의 컴포넌트들을 포함할 수 있다. 다른 실시예들에서, 다른 유형들 및/또는 구조들의 로직상의 캐시들이 이용될 수 있다. 도 2에 도시된 바와 같이, 캐시 (200)는 관련캐시 제어기 (215) 및 캐시 메모리 어레이 (210)를 갖는다. 일 실시예에서, 캐시 메모리 어레이 (210)는 세트 결합 캐시(set associative cache)로 구조화될 수 있다. 캐시 메모리 어레이 (210)는 복수의 'M' 세트들을 포함할 수 있고, 'M'은 실시예 마다 변화한다. 각각의 세트는 'N' 웨이들의 수를 포함할 수 있고, 'N' 또한 실시예마다 변화할 수 있고, 'M' 및 'N'은 양의 정수들이다.

[0023] 일 실시예에서, 각각의 세트의 'X' 웨이들의 수는 캐시 (200)의 L2 부분에 할당될 수 있고, 및 각각의 세트의 'N-X' 웨이들의 수는 캐시 (200)의 L3 부분에 할당될 수 있다. 따라서, 룩업(lookup)이 L2 부분을 수행하였을 때, 캐시 제어기 (215)는 목표가 된 어드레스에 대한 대응하는 세트의 'X' 웨이들의 수 내에서만 찾아볼 수 있다. 만약 룩업이 간과되면, L3 캐시의 'N-X' 방법들의 룩업이 수행될 수 있다. 소정의 코어에 대한 L3캐시의 룩업은 어드레스 인터리빙 기법(address interleaving scheme) 및 메모리 요청에 의해 목표가 된 어드레스에 의존하여 동일한 로직상의 캐시 또는 상이한 로직상의 캐시에 수행될 수 있다.

[0024] 캐시 제어기 (215)는 캐시 (200)내 동작들 제어하기 위해 다양한 기능들을 수행할 수 있다. 예를 들어, 캐시 제어기 (215)는 캐시 라인들을 저장, 무효로하고 및/또는 축출(evict)하고, 캐시 라인들에 대한 룩업들을 수행하고, 코히런스 동작들을 핸들링하고, 캐시 (200)로부터 캐시 라인들에 대한 요청들에 응답할 수 있다. 캐시 제어기 (215)는 캐시 (200)의 L2 부분과 L3 부분 사이에서 용량을 재지정하도록 또한 구성될 수 있다. 예를 들어,

일 실시예에서, 캐시 제어기 (215)는 캐시 (200)가 L2 및 L3 부분들로 어떻게 파티션되어야 하는지를 나타내는 업데이트들을 수신하기 위해 제어 유닛 (예를 들어, 도 1의 제어 유닛 (120))에 결합될 수 있다. 제어 유닛으로부터 업데이트 수신에 응답하여, 캐시 제어기 (215)는 캐시 (200)의 L2 및 L3 부분들의 사이즈들을 동적으로 재조정하도록 구성될 수 있다.

[0025] 이제 도 3을 참조하여, 제어 유닛 (300)의 일 실시예가 도시된다. 일 실시예에서, 도 1의 제어 유닛 (120)은 제어 유닛(300)의 컴포넌트들을 포함할 수 있다. 제어 유닛 (300)은 적어도 재매핑 유닛(remappingunit) (305) 및 테이블(들) (310)을 포함할 수 있다. 재매핑 유닛 (305)은 로직상의 캐시가 L2 및 L3 부분들 사이에서 재사이즈된 후에 로직상의 캐시 (예를 들어, 도 2의 캐시 (200))에 데이터를 재매핑하도록 구성될 수 있다. 웨이 또는 다른 부분이 로직상의 캐시의 L2 부분으로부터 L3 부분으로, 또는 로직상의 캐시의 L3 부분으로부터 L2 부분으로 재할당된 후에, 해당 웨이에 저장된 데이터는 올바른 위치에 매핑되지 않을 수 있다. 따라서, 재매핑 유닛 (305)은 L2 및 L3 구성이 바뀐 후에 데이터를 재매핑 및 마이그레이트(migrate)하도록 구성될 수 있다. 대안적으로, 데이터를 마이그레이트하는 대신에, 제어 유닛 (300)은 L2 및 L3 구성이 바뀐 후에 데이터를 무효로할 수 있다 .

[0026] 제어 유닛 (300)은 프로세서, 애플리케이션들, 및 캐시들과 관련된 다양한 파라미터들을 하도록 또한 구성될 수 있다. 일 실시예에서, 제어 유닛 (300)은 파라미터 값들을 하나 이상의 테이블(들) (310)을 저장할 수 있다. 테이블(들) (310)은 호스트 프로세서의 동작 상태들과 관련된 데이터를 저장하기 위해 이용될 수 있는 구조(들)의 임의의 수 및 유형을 예시한다. 예를 들어, 일 실시예에서, 제어 유닛 (300)은 호스트 프로세서에 의해 실행되는 복수의 애플리케이션들에 대한 테이블 (310A)에 선호되는 L2 및 L3 캐시 사이즈들을 저장할 수 있다. 대안적으로, 일부 실시예들에서, 테이블 (310A)를 이용하는 대신에, 제어 유닛 (300)은 실행되고 있는 현재 애플리케이션에 대한 소프트웨어로부터 설정을 검색하거나 또는 수신할 수 있다. 제어 유닛 (300)은 그런 다음 L2 및 L3 캐시들 사이에서 용량의 현재 할당을 조정하거나 또는 유지할지 여부를 결정하기 위해 설정을 활용할 수 있다.

[0027] 호스트 프로세서에 의해 실행되는 소정의 애플리케이션들에 대한 선호되는 L2 및 L3 캐시 사이즈들을 결정하기 위해서 테이블(310A)은 제어 유닛(300)에 의해 이용될 수 있는 테이블의 예이다. 임의 개수의 애플리케이션들에 대한 임의 개수의 설정들이 테이블 (310A)에 저장될 수 있다. 테이블 (310A)에 도시된 바와 같이, 제 1 애플리케이션은 각-코어 로직상의 캐시에 대하여 128KB의 선호되는 L2 캐시 사이즈 및 896KB의 선호되는 L3 슬라이스 사이즈를 가질 수 있고, 제 2 애플리케이션은 각-코어 로직상의 캐시에 대하여 384KB의 선호되는 L2 캐시 사이즈 및 640KB의 선호되는 L3 슬라이스 사이즈를 가질 수 있고, 제 3 애플리케이션은 각-코어 로직상의 캐시에 대하여 256KB의 선호되는 L2 캐시 사이즈 및 768KB의 선호되는 L3 슬라이스 사이즈, 등등을 가질 수 있다. 1MB가 L2 및 L3 부분들에 대한 총 각-코어 로직상의 캐시 사이즈이라고 이 논의의 목적을 위하여 가정될 수 있다. 각-코어 로직상의 캐시 사이즈에 대한 스토리지 용량의 양은 실시예에 따라 변할 수 있다.

[0028] 각각의 각-코어 로직상의 캐시에 대한 L2 및 L3 캐시 사이즈들을 조절할 때를 결정하기 위해 테이블(310B)은 제어 유닛(300)에 의해 이용될 수 있는 테이블의 다른 예이다. 테이블 (310B)은 하나 이상의 임계값들과 비교를 위해 이용될 수 있는 L2 및 L3 캐시들과 관련된 파라미터들의 현재 값들을 포함할 수 있다. 일 실시예에서, 소프트웨어 및/또는 다른 제어 로직은 테이블 (310B)에 저장된 임계값들을 프로그램하도록 구성될 수 있다. 다른 실시예에서, 테이블 (310B)에 저장된 임계값들은 미리 결정될 수 있다. 제어 유닛 (300)은 재사이즈후에 데이터를 재위치시키기 위해 또는 L2 및 L3 캐시 부분들을 동적으로 재사이즈하는 것을 돕기 위해 이용될 수 있는 다른 테이블들 (예를 들어, 재매핑 테이블)을 포함할 수 있다는 것에 유의하여야 한다.

[0029] 이제 도 4를 참조하여, 프로세서 (400)의 일 실시예가 도시된다. 프로세서 (400)는 코어들 (405A-D) 및 캐시들 (410A-D)을 포함할 수 있다. 다른 실시예들에서, 프로세서 (405)은 다른 수의 코어들 및 대응하는 캐시들을 포함할 수 있다. 각각의 캐시 (410A-D)는 L2 캐시 및 L3 슬라이스를 포함할 수 있고, L2 캐시 및 L3 슬라이스는 사이즈에서 조절가능하다. 각각 캐시 (410A-D)의 총 사이즈는 고정될 수 있어서, L2 부분의 사이즈에서 증가는 L3 부분의 사이즈에서 감소 만큼 오프셋될 것이다. 소정의 캐시 (410A-D)에 대한 L2 및 L3 부분들간의 사이즈 혼합은 조절될 수 있어서 프로세서 성능은 프로세서 (400)에 의해 실행되는 현재 동작 상태들 및/또는 현재 애플리케이션(들)에 대하여 최적화된다.

[0030] 도 4에 도시된 바와 같이 캐시들 (410A-D)의 자원들의 할당은 스페이스(space)가 캐시들 (410A-D)의 L2 부분과 L3 부분사이에 어떻게 할당될 수 있는 일 예를 예시한다. 도 4에 도시된 바와 같이, 캐시 (410A)의 두개의 방법들이 L2 캐시에 할당되고 캐시 (410A)의 네개의 방법들이 L3 캐시에 할당된다. 캐시 (410A)의 여섯개의 방법들이 있다고 이 논의의 목적을 위해 가정될 수 있고, L2 캐시에 할당된 캐시(410A)의 용량은 256 킬로바이트

(KB)이고 L3 캐시에 할당된 캐시(410A)의 용량은 512KB이라고 또한 가정될 수 있다. 캐시 (410A)에 도시된 L2 및 L3 부분들의 할당은 캐시들 (410B-D)에 대하여 또한 사용될 수 있다. 그러나, L2 및 L3 부분들에 대한 용량의 할당은 각각의 캐시들 (410A-D)에 같아야 할 필요는 없다는 것이 이해되어야 한다.

[0031] L3 캐시 (415)는 코어들 (405A-D) 간에 공유된 캐시일 수 있고, L3 캐시 (415)에 대하여 할당된 총 용량은 모든 네개의 L3 슬라이스들이 함께 더해졌을 때 2 메가바이트(MB)이다. 프로세서 (400)는 어드레스 인터리빙(address interleaving)을 활용할 수 있어서 소정의 코어 (405A-D)에 대한 요청은 소정의 코어 (405A-D)에 대응하는 캐시 (410A-D)와 상이한 캐시 (410A-D)의 L3 슬라이스에 매핑될 수 있다. 예를 들어, 코어 (405A)에 의해 생성된 메모리 요청들에 대한 어드레스들은 캐시 (410A), 캐시 (410B), 캐시 (410C), 및 캐시 (410D)의 L3부분들에 매핑될 수 있다. 다른 코어들이 유사한 어드레스 인터리빙 기법으로 모든 캐시들 (410A-D)에 또한 매핑될 수 있다.

[0032] 캐시들 (410A-D)이 L2 캐시에 할당된 웨이들의 제 1 부분 및 L3 캐시에 할당된 웨이들의 제 2 부분을 가질 수 있다는 것이 도 4에서 예시되었지만, 캐시들을 분할하기 위한 기법의 단지 일 예이다는 것이 이해되어야 한다. 다른 실시예들에서, 캐시들 (410A-D)은 L2 및 L3 부분들 사이에서 다른 치수 (예를 들어, 세트들, 뱅크(bank)들)의 부분들로 분할될 수 있다. 예를 들어, 다른 실시예에서, 각각의 캐시 (410A-D)에 대하여 제 1 수의 세트들이 L2 캐시에 할당될 수 있고 및 제 2 수의 세트들이 L3 캐시에 할당될 수 있다.

[0033] 이제 도 5로 가서, 프로세서(500)의 다른 실시예의 블록 다이어그램이 도시된다. 프로세서 (500)는 코어들 (505A-D) 및 캐시들 (510A-D)을 포함할 수 있다. 다른 실시예들에서, 프로세서 (505)은 다른 수의 코어들을 포함할 수 있다. 각각의 캐시 (510A-D)는 L2 부분 및 L3 부분을 포함할 수 있고, L2 및 L3 부분들은 사이즈에서 조절가능하다. 프로세서 (500)는 각각의 캐시 (510A-D)의 L2 및 L3 부분들을 동적으로 조절하도록 구성될 수 있다.

[0034] 캐시 구현예의 일 예가 도 5에 도시되고, 각각의 캐시 (510A-D)에 대하여 L2 캐시에 세계의 웨이들이 할당되고 L3 슬라이스에 세계의 웨이들이 할당된다. 일 실시예에서, 프로세서 (500)는 L2 캐시들 및 L3 캐시의 사이즈들이 조절된 후에 나중의 프로세서 (400)를 나타낼 수 있다. 이 실시예에서, L3 슬라이스들은 각각의 캐시 (510A-D)의 네개의 웨이들로부터 각각의 캐시 (510A-D)의 세계의 웨이들로 축소되었다. 프로세서 (500)는 하나 이상의 상태들을 검출한 것에 응답하여 L3 캐시 (515) 및 L2 캐시들의 사이즈를 조정할 수 있다. 캐시 사이즈들을 조정하기 위한 상태들의 예들이 도면들 8-13와 관련된 논의에서 이하에 보다 상세하게 설명될 것이다.

[0035] 도 5에 도시된 실시예에서, 각각의 웨이는 각각의 캐시 (510A-D)에 대하여 128KB의 데이터에 대응한다고 가정될 수 있다. 따라서, 캐시들 (510A-D)의 각각의 L2 캐시는 384KB의 용량을 포함하지만, 반면 캐시들 (510A-D)의 각각의 L3 슬라이스도 또한 384KB의 용량을 포함한다. 공유된 L3 캐시 (515)의 총 용량은 이 실시예에서 1.5MB이다. 다른 실시예들은 각 캐시 (510A-D) 마다 상이한 수의 웨이들 및/또는 각 웨이마다 다른 양의 용량을 가질 수 있다.

[0036] 일 실시예에서, 코어 (505A)는 그것의 L2 캐시에 캐시 라인 'X'의 카피(copy)를 캐시에 저장(cache) 할 수 있다. 코어 (505C)는 그것의 개별 L2 캐시에 캐시 라인 'X'의 카피를 또한 캐시에 저장 할 수 있다. 그러나, 코어 (505B)는 그것의 L2 캐시에 캐시 라인 'X'의 카피를 캐시에 저장하지 않는다고 이 논의의 목적을 위해 가정될 수 있다. 캐시 슬라이스 (510B)의 L2 슬라이스에서의 캐시의 누락(miss)시에, L3 어드레스 인터리빙이 캐시 라인 'X'를 코어 (505D)의 L3 슬라이스에 매핑한다고 가정하여 코어 (505B)는 요청을 코어 (505D)에 발송할 수 있다. 따라서, 룩업은 코어 (505D)의 L3 슬라이스에 캐시 라인 'X'를 찾기 위해 수행될 수 있고, 이는 그런 다음 코어 (505B)에 제공될 것이다.

[0037] 이제 도 6으로가서, 세계의 상이한 시간 지점들에서 캐시(600)의 일 실시예의 블록 다이어그램이 도시된다. 캐시 (600A)는 도 6의 최상부에 도시되고, 캐시 (600A)는 L2 캐시 (605A) 및 L3 슬라이스 (610A)를 포함한다. L3 슬라이스 (610A)는 다른 코어들에 대응하는 다른 캐시들 (미도시)의 L3 슬라이스들을 포함하는 더 큰 L3 캐시의 부분일 수 있다. 캐시 (600A)는 동작 상태들 변화에 응답하여 L2 캐시 (605A) 및 L3 슬라이스 (610A)의 사이즈들을 동적으로 조절하도록 구성될 수 있다. 캐시 (600A)에 도시된 바와 같이, L2 캐시 (605A)는 웨이들 0-2를 포함하고 L3 슬라이스 (610A)는 웨이들 3-7을 포함한다. 다른 실시예들에서, 캐시 (600A)은 다른 수의 웨이들을 포함할 수 있다. 캐시 (600A)는 또한 'N' 세트들을 포함할 수 있고, 여기서 'N'은 실시예마다 바뀔 수 있는 양의 정수이다. 캐시 라인 (625)은 캐시 (600A)의 세트 0의 웨이 3에 저장된 것으로 도시되고, 캐시 라인 (630)은 캐시 (600A)의 세트 1의 웨이 3에 저장된 것으로 도시되고, 캐시 라인 (635)은 캐시 (600A)의 세트 N의 웨이 3에 저장된 것으로 도시된다. 캐시 (600A)는 도면들 모호하게 하는 것을 피하기 위하여 미도시인 다른 캐시 라인들을 저장할 수 있다는 것에 유의한다.

- [0038] 캐시 (600A)의 용량은 하나 이상의 상태들을 감지 한 것, 소프트웨어로부터의 명령 및/또는 다른 이벤트(들) 수신에 응답하여 L2 및 L3 부분들 간에 동적으로 재할당된다는 것이 이 논의의 목적을 위하여 가정될 수 있다. 도 6의 중간에 캐시 (600B)의 예시는 L2 및 L3 부분들이 재사이즈 된 후 나중에 캐시 (600A)를 나타내도록 의도된다. 캐시 (600B)에 도시된 바와 같이, 웨이 3이 L3 슬라이스 (610B)로부터 L2 캐시 (605B)로 재할당되었다. 캐시 라인들 (625,630, 및 635)은 웨이3에 그것들의 동일한 물리적 위치에 잔존할 수 있지만, 그러나 그것들은 L3 슬라이스 (610B)라기 보다는 L2 캐시 (605B)에 이제 저장된다. 그러나, 캐시 라인들 (625,630, 및 635)은 룩업 (lookup)이 L2 캐시 (605B)에 수행될 때 캐시 제어기 (미도시)가 이들 캐시 라인들을 찾는 것을 방지하도록 L2 캐시 (605B)내 이들 위치들에 매핑하지 않을 수 있다.
- [0039] 따라서, L2 캐시 (605B) 및 L3 슬라이스 (610B)의 동적 재사이즈에 응답하여, 캐시 제어기는 캐시 라인들이 그것들의 어드레스들의 적절한 비트들에 기초하여 실제로 매핑될 위치들로 재사이즈함으로써 영향을 받을 캐시 라인들을 재매핑하고 마이그레이트하도록 구성될 수 있다. 도 6의 바닥에 캐시 (600C) ( L2 캐시 (605C) 및 L3 슬라이스 (610C)를 포함하는)의 예시는 캐시 라인들 (625,630, 및 635)이 L2 매핑에 대응하는 그것들의 어드레스 비트들에 기초하여 적절한 위치들로 재매핑된 후 나중에 캐시 (600B)를 나타내도록 의도된다. 캐시 라인 (625)은 L2 캐시 (605C)의 세트 1의 웨이 1로 재매핑되고 마이그레이트되었고, 캐시 라인 (630)은 L2 캐시 (605C)의 세트 N-1의 웨이 0로 재매핑되고 마이그레이트되었고, 캐시 라인 (635)은 L2 캐시 (605C)의 세트 0의 웨이 0로 재매핑되고 마이그레이트되었다고 이 논의의 목적을 위해 가정될 수 있다. 대안적으로, 캐시 라인들 (625,630, 및 635) 재매핑 및 마이그레이트보다는 오히려, 이들 캐시 라인들은 무효로 될(invalidated) 수 있다. 캐시 라인들이 나중에 다시 요청되면 그리고 요청된 때, 캐시 누락(cachemiss)일 것이고, 그런 다음 캐시 라인들은 어드레스 비트들 및 L2 매핑에 기초하여 적절한 위치들에 L2 캐시내로 삽입될 것이다.
- [0040] 이들 재-매핑들은 단지 재매핑 기법의 일 예를 예시하도록 의도되고, 재매핑 기법들은 영향을 미치는 캐시 라인들 및 캐시 매핑 기법들의 태그 비트들에 의존하여 바뀐다. 캐시 (600)의 구조화(organization)은 캐시의 하나의 가능한 구조화를 예시하도록 의도된다는 것이 이해되어야 한다. 다른 실시예들에서, 다른 수들의 세트들, 웨이들, 및/또는 다른 캐시 치수를 갖는 다른 구조들의 캐시들이 구현될 수 있다.
- [0041] 이제 도 7로 가서, 다수의 캐시 라인들의 어드레스 비트들을 예시하는 다이어그램들이 도시된다. 샘플 캐시 라인 어드레스 (700)이 도 7의 최상부에 도시되고, 샘플 캐시 라인 (700)은 실시예에 의존하여 임의 개수의 비트들을 포함한다. 어드레스 (700)의 비트들 (715)의 제 1 부분은 L3 캐시의 소정의 슬라이스 (예를 들어, 도 6의 L3 슬라이스 (610A))를 선택하기 위해 사용될 수 있고, 어드레스 (700)의 비트들 (720)의 제 2 부분은 소정 세트의 L2 캐시 (예를 들어, 도 6의 L2 캐시 (605A))를 선택하기 위해 사용될 수 있고, 어드레스 (700)의 비트들 (725)의 제 3 부분은 소정 세트의 L3 슬라이스 (예를 들어, L3 슬라이스 (610A))를 선택하기 위한 L3 태그 비트들로서 사용될 수 있다. 비록 L3 슬라이스 인덱스 비트들 (715)은 두개의 비트들을 포함하고 L2 세트 인덱스 비트들 (720) 및 L3 세트 인덱스 비트들 (725)은 세개의 비트들을 포함하는 것으로 도시되지만, 이것은 단지 일 실시예의 예시이고, 다른 실시예들은 다른 수들의 L3 슬라이스 인덱스 비트들 및 L2 및 L3 세트 인덱스 비트들을 가질 수 있다. 다양한 실시예들에서, L2 인덱스 비트들, L3 슬라이스 비트들, 및 L3 인덱스 비트들은 서로로부터 분리될 수 있다. 대안적으로, 다양한 실시예들에서 하나 이상의 비트들의 L2 인덱스, L3 슬라이스, 및 L3 인덱스가 중첩될 수 있다.
- [0042] 캐시 라인들 (625,630, 및 635)의 어드레스 비트들이 도 7의 하부에 도시된다. 캐시 라인들 (625,630, 및 635)의 L3 슬라이스 인덱스 비트들 '00'은 캐시 라인들 (625,630, 및 635)을, 개별적으로, 캐시 (600A)에 도시된 L3 슬라이스 (610A)로 매핑하기 위해 사용된다고 이 논의의 목적을 위해 가정될 수 있다. 캐시 라인들 (625,630, 및 635)의 L3 세트 인덱스 비트들은 캐시 라인들 (625,630, 및 635)을, 개별적으로, 캐시 (600A)에 도시된 L3 슬라이스 (610A)의 올바른 세트들로 매핑하기 위해 사용된다고 이 논의의 목적을 위해 또한 가정될 수 있다. 따라서, 캐시 라인 (625)의 어드레스의 L3 세트 인덱스 부분은 "000"이고, 이는 캐시 라인 (625)을 L3 슬라이스 (610A)의 세트 0에 매핑하고, 캐시 라인 (630)의 어드레스의 L3 세트 인덱스 부분은 "001"이고, 이는 캐시 라인 (630)을 L3 슬라이스 (610A)의 세트 1에 매핑하고, 캐시 라인 (635)의 어드레스의 L3 세트 인덱스 부분은 "111"이고, 이는 캐시 라인 (635)을 L3 슬라이스 (610A)의 세트 N-1에 매핑한다. 'N'은 이 특정 실시예에서 8과 같다고 가정될 수 있다. 그런다음 웨이 3이 L3 슬라이스 (610A)로부터 L2 캐시 (605B)로 재할당된 후에, 캐시 라인들 (625,630, 및 635)의 어드레스의 L2 세트 인덱스 부분들은 캐시 라인들 (625,630, 및 635)을, 개별적으로, 캐시 (600C)에 도시된 L2 캐시 (605C)의 올바른 세트들로 재매핑하고 마이그레이트하기 위해 사용될 수 있다. 따라서, 캐시 라인 (625)의 어드레스의 L2 세트 인덱스 부분은 "001"이고, 이는 캐시 라인 (625)을 L2 캐시 (605C)의 세트 1에 매핑하고, 캐시 라인 (630)의 어드레스의 L2 세트 인덱스 부분은 "111"이고, 이는 캐시 라인

(630)을 L2 캐시 (605C)의 세트 N-1에 매핑하고, 캐시 라인 (635)의 어드레스의 L2 세트 인덱스 부분은 "000"이고, 이는 캐시 라인 (635)을 L2 캐시 (605C)의 세트 0에 매핑한다.

- [0043] 도 7에 도시된 캐시 슬라이스 및 세트 매핑들을 결정하기 위해 사용되는 특정 어드레스 비트들의 예들은 단지 하나의 가능한 구현예라는 것이 이해되어야 한다. 다른 구현예들은 대응하는 캐시들의 L2 캐시들 및 L3 슬라이스들내 위치들로 캐시 라인들의 매핑을 결정하기 위해 다른 어드레스 비트들을 활용할 수 있다.
- [0044] 이제 도 8로 가서, 로직상의 캐시 사이즈들을 동적으로 조정하기 위한 방법(800)의 일 실시예가 도시된다. 논의 목적들을 위하여, 이 실시예에 단계들은 순차적인 순서로 도시된다. 이하에 설명되는 방법의 다양한 실시예들에서, 설명되는 하나 이상의 엘리먼트들 동시에, 도시된 것과 다른 상이한 순서로 수행될 수 있거나, 또는 완전히 생략될 수 있다는 것에 주목되어야 한다. 다른 추가 엘리먼트들이 원하는 때 또한 수행될 수 있다. 본 출원에서 설명된 임의의 다양한 시스템들 또는 프로세서들은 방법 (800)을 구현하도록 구성될 수 있다.
- [0045] 프로세서는 제 1 로직상의 캐시 및 제 2 로직상의 캐시와 관련된 하나 이상의 파라미터들을 모니터링할 수 있다 (블럭 (805)). 일 실시예에서, 단일 캐시는 제 1 로직상의 캐시 및 제 2 로직상의 캐시로 논리적으로 파티션될 수 있고, 제 1 로직상의 캐시는 L2 캐시를 나타내고 제 2 로직상의 캐시는 L3 캐시를 나타낸다. 다른 실시예들에서, 제 1 및 제 2 로직상의 캐시들은 프로세서의 캐시 계층의 다른 레벨들을 나타낼 수 있다. 하나 이상의 파라미터들은 제 1 로직상의 캐시 및/또는 제 2 로직상의 캐시의 캐시 히트율(cache hit rate), 제 1 로직상의 캐시 및/또는 제 2 로직상의 캐시의 캐시 누락율(cache miss rate), 상이한 제 1 로직상의 캐시들간에 캐시 라인들의 마이그레이션 비율(migration rate), 및/또는 다른 파라미터들을 포함할 수 있다.
- [0046] 다음, 프로세서는 하나 이상의 파라미터들을 하나 이상의 임계값들에 비교할 수 있다 (블럭 (810)). 예를 들어, 다양한 실시예들에서, 프로세서는 제 1 로직상의 캐시의 캐시 히트율을 제 1 임계값에 비교할 수 있고, 프로세서는 제 2 로직상의 캐시의 캐시 히트율을 제 2 임계값에 비교할 수 있고, 프로세서는 제 1 로직상의 캐시의 히트율과 제 2 로직상의 캐시의 히트율 사이의 차이값을 제 3 임계값에 비교할 수 있고, 프로세서는 마이그레이션 비율을 제 4 임계값에 비교할 수 있고, 및/또는 프로세서는 다른 비교들을 수행할 수 있다. 캐시 히트율 또는 캐시 라인 마이그레이션 비율이 산출되는 시간의 양은 실시예에 따라 변화할 수 있다. 일 실시예에서, 비교 결과들은 일부 실시예들에서 중요도의 순서에 따라 순위 매김(rank)될 수 있어서, 만약 가장 높은 순위에 매김된 비교 결과는 소정의 임계값이 초과된 것으로 귀결되고, 그런 다음 이것은 모든 다른 비교 결과들을 교체할 것이다. 다른 실시예에서, 비교 결과들은 동등하게 취급될 수 있어서 하나의 비교 결과는 다른 비교의 결과를 상쇄한다. 일반적으로, 실시예들은 하나 이상의 비교들을 개별적으로 또는 조합하여 (예를 들어, 제 1 및 제 2 비교 결과 양쪽이 개별 제 1 및 제 2 소정의 임계값들을 초과한 때) 고려되는 임의의 함수를 활용할 수 있다.
- [0047] 다음, 프로세서는 비교(들) 결과들이 제 1 및 제 2 로직상의 캐시들의 사이즈들이 조정되어야 하는지를 표시하는지를 결정할 수 있다(조건 블럭 (815)). 예를 들어, 비교 결과는 개선된 프로세서 성능을 달성하기 위해 제 1 로직상의 캐시의 사이즈가 축소되어야 하고 제 2 로직상의 캐시의 사이즈는 증가되어야 한다고 표시할 수 있다. 대안적으로, 비교 결과(들)은 개선된 프로세서 성능을 달성하기 위해 제 1 로직상의 캐시의 사이즈가 증가되어야 하고 제 2 로직상의 캐시의 사이즈는 축소되어야 한다고 표시할 수 있다.
- [0048] 만약 비교(들) 결과들이 제 1 및 제 2 로직상의 캐시들의 사이즈들이 조정되어야 한다고 표시하면(조건 블럭 (815), "예" 레그), 그러면 프로세서는 제 1 및 제 2 로직상의 캐시들의 사이즈들을 동적으로 조정할 수 있다 (블럭 (820)). 만약 비교(들) 결과들이 제 1 및 제 2 로직상의 캐시들의 사이즈들이 조정되어야 한다고 표시하지 않으면(조건 블럭 (815), "아니오" 레그), 그러면 프로세서는 제 1 및 제 2 로직상의 캐시들의 현재 사이즈들을 유지할 수 있다 (블럭 (825)). 블럭들 (820) 및 (825) 후에, 방법 (800)은 종료한다. 방법 (800)은 규칙적인 간격들상에서 수행될 수 있다는 것에 유의한다. 이하에 도면들 9-12는 특정 파라미터들이 모니터링되고 하나 이상의 임계값들에 비교되는 실시예들에 대한 추가 세부사항들을 제공한다.
- [0049] 이제 도 9로 가서, 로직상의 캐시 사이즈들을 동적으로 조정하기 위한 방법(900)의 다른 실시예가 도시된다. 논의 목적들을 위하여, 이 실시예에 단계들은 순차적인 순서로 도시된다. 이하에 설명되는 방법의 다양한 실시예들에서, 설명되는 하나 이상의 엘리먼트들 동시에, 도시된 것과 다른 상이한 순서로 수행될 수 있거나, 또는 완전히 생략될 수 있다는 것에 주목되어야 한다. 다른 추가 엘리먼트들이 원하는 때 또한 수행될 수 있다. 본 출원에서 설명된 임의의 다양한 시스템들 또는 프로세서들은 방법 (900)을 구현하도록 구성될 수 있다.
- [0050] 프로세서는 제 1 캐시의 캐시 히트율(cachehitrate)을 모니터링할 수 있다 (블럭 (905)). 대안적으로, 블럭 (905)에서, 프로세서는 제 1 캐시의 캐시 누락율을 모니터링할 수 있고, 이는 1 - 캐시 히트율과 같다. 일 실시

예에서, 프로세서는 멀티-코어 프로세서일 있고, 코어들의 수는 실시예 마다 변화할 수 있다. 다양한 실시예들에서, 단일 캐시는 제 1 (로직상의) 캐시 및 제 2 (로직상의) 캐시로 논리적으로 파티션된다. 일 실시예에서, 제 1 캐시는 L2 캐시일 수 있고, 제 2 캐시는 L3 캐시일 수 있다. 다음, 프로세서는 제 1 캐시의 히트율을 임계값에 비교할 수 있다 (블럭 (910)). 일 실시예에서, 임계값은 프로그램 가능할 수 있다.

[0051] 만약 제 1 캐시의 히트율이 임계값보다 더 크면 (조건 블럭 (915), "예" 레그), 그러면 프로세서는 제 1 캐시의 사이즈를 증가시킬 수 있고 제 2 캐시의 사이즈를 축소시킬 수 있다 (블럭 (920)). 만약 제 1 캐시의 히트율이 임계값보다 더 작으면 (조건 블럭 (915), "아니오" 레그), 그러면 프로세서는 제 1 캐시 및 제 2 캐시의 현재 사이즈를 유지할 수 있다 (블럭 (925)). 블럭들 (920) 및 (925) 후에, 방법 (900)은 종료한다. 일 실시예에서, 캐시 사이즈들이 제 1 캐시의 히트율에 기초하여 조절되어야 하는지를 결정하기 위해 방법 (900)은 규칙적인 간격들로 반복될 수 있다. 방법 (900)은 하나 이상의 다른 방법들 (예를 들어, 도면들 10-12의 방법들 (1000, 1100, 및 1200))과 동시에 프로세서에 의해 수행될 수 있다는 것에 유의한다. 일반적으로, 실시예들은 하나 이상의 비교들을 개별적으로 또는 조합하여 (예를 들어, 제 1 및 제 2 비교 결과 양쪽이 개별 제 1 및 제 2 소정의 임계값들을 초과한 때) 고려되는 임의의 함수를 활용할 수 있다.

[0052] 이제 도 10으로 가서, 캐시 사이즈들을 조정할 때를 결정하기 위한 방법(1000)의 일 실시예가 도시된다. 논의 목적들을 위하여, 이 실시예에 단계들은 순차적인 순서로 도시된다. 이하에 설명되는 방법의 다양한 실시예들에서, 설명되는 하나 이상의 엘리먼트들 동시에, 도시된 것과 다른 상이한 순서로 수행될 수 있거나, 또는 완전히 생략될 수 있다는 것에 주목되어야 한다. 다른 추가 엘리먼트들이 원하는 때 또한 수행될 수 있다. 본 출원에서 설명된 임의의 다양한 시스템들 또는 프로세서들은 방법 (1000)을 구현하도록 구성될 수 있다.

[0053] 프로세서는 제 1 캐시 및 제 2 캐시의 히트율을 모니터링할 수 있다 (블럭 (1005)). 제 1 (로직상의) 캐시가 단일 물리적 캐시내에서 제 2 (로직상의) 캐시와 결합된다고 이 논의의 목적을 위해 가정될 수 있다. 일 실시예에서, 제 1 캐시는 L2 캐시일 수 있고, 제 2 캐시는 L3 캐시일 수 있다. 다른 실시예들에서, 제 1 및 제 2 캐시들은 멀티-레벨 캐시 계층의 다른 캐시 레벨들일 수 있다. 다음, 프로세서는 제 1 캐시의 히트율과 제 2 캐시의 히트율 간의 차이값을 제 1 및 제 2 임계값들에 비교할 수 있다 (블럭 (1010)). 일 실시예에서, 제 1 및 제 2 임계값들은 프로그램 가능할 수 있다.

[0054] 만약 제 1 캐시의(FC) 히트율이 제 1 임계값만큼 제 2 캐시의 (SC) 히트율보다 더 크면 (조건 블럭 (1015), "예" 레그), 그러면 프로세서는 제 1 캐시의 사이즈를 증가시킬 수 있고 제 2 캐시의 사이즈를 축소시킬 수 있다 (블럭 (1020)). 만약 제 1 캐시의 히트율이 제 1 임계값만큼 제 2 캐시의 히트율보다 더 크지 않으면 (조건 블럭 (1015), "아니오" 레그), 그러면 프로세서는 제 2 캐시의 히트율이 제 2 임계값 만큼 제 1 캐시의 히트율보다 더 큰지를 결정할 수 있다(블럭 (1025)). 만약 제 2 캐시의 히트율이 제 2 임계값만큼 제 1 캐시의 히트율보다 더 크면 (조건 블럭 (1025), "예" 레그), 그러면 프로세서는 제 2캐시의 사이즈를 증가시킬 수 있고 제 1 캐시의 사이즈를 축소시킬 수 있다 (블럭 (1030)). 만약 제 2 캐시의 히트율이 제 2 임계값만큼 제 1 캐시의 히트율보다 더 크지 않으면 (조건 블럭 (1025), "아니오" 레그), 그러면 프로세서는 제 1 캐시 및 제 2 캐시의 현재 사이즈들을 유지할 수 있다(블럭 (1035)). 블럭들 (1020), (1030) 및 (1035) 후에, 방법 (1000)은 종료한다. 일 실시예에서, 캐시 사이즈들이 제 1 및 제 2 캐시들의 히트율간의 차이값에 기초하여 조절되어야 하는지를 결정하기 위해 방법 (1000)은 규칙적인 베이스들로 반복될 수 있다.

[0055] 이제 도 11로 가서, 캐시 사이즈들을 조정할 때를 결정하기 위한 방법(1100)의 다른 실시예가 도시된다. 논의 목적들을 위하여, 이 실시예에 단계들은 순차적인 순서로 도시된다. 이하에 설명되는 방법의 다양한 실시예들에서, 설명되는 하나 이상의 엘리먼트들 동시에, 도시된 것과 다른 상이한 순서로 수행될 수 있거나, 또는 완전히 생략될 수 있다는 것에 주목되어야 한다. 다른 추가 엘리먼트들이 원하는 때 또한 수행될 수 있다. 본 출원에서 설명된 임의의 다양한 시스템들 또는 프로세서들은 방법 (1100)을 구현하도록 구성될 수 있다.

[0056] 프로세서는 제 1 캐시 및 제 2 캐시의 히트율을 모니터링할 수 있다 (블럭 (1105)). 단일 물리적 캐시가 제 1 및 제 2 캐시들로 논리적으로 파티션된다고 이 논의의 목적을 위해 가정될 수 있다. 일 실시예에서, 제 1 캐시는 L2 캐시일 수 있고, 제 2 캐시는 L3 캐시일 수 있다. 다른 실시예들에서, 제 1 및 제 2 캐시들은 멀티-레벨 캐시 계층의 다른 캐시 레벨들일 수 있다. 다음, 프로세서는 제 1캐시의 히트율을 제 1 임계값에 비교할 수 있다 (블럭 (1110)).

[0057] 만약 제 1 캐시의 히트율이 제 1 임계값보다 더 작으면 (조건 블럭 (1115), "예" 레그), 그러면 프로세서는 제 2 캐시의 히트율을 제 2 임계값에 비교할 수 있다 (블럭 (1120)). 일 실시예에서, 제 1 및 제 2 임계값들은 프로그램 가능할 수 있다. 만약 제 1 캐시의 히트율이 제 1 임계값보다 더 크면 (조건 블럭 (1115), "아니오" 레

그), 그러면 프로세서는 제 1 캐시 및 제 2 캐시들의 현재 사이즈들을 유지할 수 있다 (블럭 (1125)). 일부 실시예들에서, 프로세서는 제 1 캐시의 히트율을 복수의 임계값들에 비교할 수 있고, 제 1 임계값은 상대적으로 낮은 임계값이다. 프로세서는 일부 시나리오들에서 제 1 캐시의 히트율을 상대적으로 높은 임계값에 비교할 수 있고, 프로세서는 상대적으로 높은 임계값을 초과하는 제 1 캐시의 히트율에 응답하여 제 1 캐시의 사이즈를 증가시킬 수 있고 제 2 캐시의 사이즈를 축소시킬 수 있다. 이런 실시예가 도 9의 방법 (900)과 관련하여 설명된다.

[0058] 만약 제 2 캐시의 히트율이 제 2 임계값보다 더 작으면 (조건 블럭 (1130), "예" 레그), 그러면 프로세서는 제 2 캐시의 사이즈를 증가시킬 수 있고 제 1 캐시의 사이즈를 축소시킬 수 있다 (블럭 (1135)). 만약 제 1 및 제 2 캐시의 히트율 둘 모두가 그것들의 대응하는 임계값들보다 작으면, 이것은 프로세서가 제 2 캐시에 피트(fit)들보다 더 많은 데이터를 액세스하고 있다는 것을 나타내고, 따라서 이 경우에, 프로세서는 제 2 캐시의 사이즈를 증가시키기 위해서 제 1 캐시의 사이즈를 줄임으로써 성능을 개선시킬 수 있다. 만약 제 2 캐시의 히트율이 제 2 임계값보다 더 크면 (조건 블럭 (1130), "아니오" 레그), 그러면 프로세서는 제 1 캐시 및 제 2 캐시들의 현재 사이즈들을 유지할 수 있다 (블럭 (1125)). 블럭들 (1125) 및 (1135) 후에, 방법 (1100)은 종료한다.

[0059] 이제 도 12로 가서, 캐시 계층에 마이그레이션 비율을 모니터링하기 위한 방법(1200)의 일 실시예가 도시된다. 논의 목적들을 위하여, 이 실시예에 단계들은 순차적인 순서로 도시된다. 이하에 설명되는 방법의 다양한 실시예들에서, 설명되는 하나 이상의 엘리먼트들 동시에, 도시된 것과 다른 상이한 순서로 수행될 수 있거나, 또는 완전히 생략될 수 있다는 것에 주목되어야 한다. 다른 추가 엘리먼트들이 원하는 때 또한 수행될 수 있다. 본 출원에서 설명된 임의의 다양한 시스템들 또는 프로세서들은 방법 (1200)을 구현하도록 구성될 수 있다.

[0060] 멀티-레벨 캐시 계층을 갖는 멀티-코어 프로세서는 멀티-레벨 캐시 계층의 소정 레벨의 캐시들간에 마이그레이션 비율을 모니터링할 수 있다(블럭 (1205)). 일 실시예에서, 소정 레벨은 캐시 계층의 L2 레벨일 수 있다. 다음, 프로세서는 마이그레이션 비율을 임계값에 비교할 수 있다 (블럭 (1210)). 일 실시예에서, 임계값은 프로그램 가능할 수 있다. 만약 마이그레이션 비율이 임계값 보다 더 크면 (조건 블럭 (1215), "예" 레그), 그러면 프로세서는 소정 레벨의 캐시들의 사이즈를 축소시킬 수 있고 캐시 계층의 다음 상위 레벨의 캐시(들)의 사이즈(들)를 증가시킬 수 있다 (블럭 (1220)). 일 실시예에서, 다음 상위 레벨은 캐시 계층의 L3 레벨일 수 있다. 만약 마이그레이션 비율이 임계값보다 더 작으면 (조건 블럭 (1215), "아니오" 레그), 그러면 프로세서는 소정 레벨 및 다음 상위 레벨의 캐시들의 현재 사이즈들을 유지할 수 있다 (블럭 (1225)). 블럭들 (1220) 및 (1225) 후에, 방법 (1200)은 종료한다.

[0061] 이제 도 13으로 가서, 인접한 캐시 레벨들간에 상이한 용량 할당(capacity allocation)을 이용하기 위한 방법 (1300)의 일 실시예가 도시된다. 논의 목적들을 위하여, 이 실시예에 단계들은 순차적인 순서로 도시된다. 이하에 설명되는 방법의 다양한 실시예들에서, 설명되는 하나 이상의 엘리먼트들 동시에, 도시된 것과 다른 상이한 순서로 수행될 수 있거나, 또는 완전히 생략될 수 있다는 것에 주목되어야 한다. 다른 추가 엘리먼트들이 원하는 때 또한 수행될 수 있다. 본 출원에서 설명된 임의의 다양한 시스템들 또는 프로세서들은 방법 (1300)을 구현하도록 구성될 수 있다.

[0062] 멀티-레벨 캐시 계층을 갖는 프로세서는 제 1 애플리케이션을 실행시킬 수 있다 (블럭 (1305)). 멀티-레벨 캐시 계층은 제 1 로직상의 캐시 (또는 제 1 캐시) 및 제 2 로직상의 캐시 (또는 제 2 캐시)의 부분을 포함하는 물리적 캐시를 포함할 수 있다. 일 실시예에서, 제 1 캐시는 L2 캐시일 수 있고, 제 2 캐시는 L3 캐시일 수 있다. 프로세서는 제 1 애플리케이션을 실행시키는 동안 물리적 캐시의 제 1 캐시 및 제 2 캐시의 제 1 용량 할당을 이용할 수 있다 (블럭 (1310)). 일 실시예에서, 제 1 용량 할당은 제 1 캐시에 할당된 제 1 용량 및 제 2 캐시에 할당된 제 2 용량을 포함할 수 있고, 제 1 용량 할당은 제 1 애플리케이션에 대하여 선호되는 할당일 수 있다. 제 1 용량 및 제 2 용량의 조합(또는 합)은 제 3 용량과 같을 수 있고, 제 3 용량은 소정의 물리적 캐시에 대하여 고정될 수 있다.

[0063] 다음, 프로세서는 제 2 애플리케이션 실행으로의 스위치를 검출할 수 있다 (블럭 (1315)). 제 2 애플리케이션 실행으로의 스위치 검출에 응답하여, 프로세서는 제 2 애플리케이션을 실행시키는 동안 물리적 캐시의 제 1 캐시 및 제 2 캐시간에 제 2 용량 할당을 이용할 수 있다 (블럭 (1320)). 제 2 용량 할당이 제 2 애플리케이션에 대하여 선호된다고 이 논의의 목적을 위해 가정될 수 있다. 제 2 용량 할당은 제 1 용량 할당과 상이하고, 제 2 용량 할당은 제 1 캐시에 할당된 제 4 용량 및 제 2 캐시에 할당된 제 5 용량을 포함한다고 또한 가정될 수 있다. 다시 말해서, 제 4 용량은 제 1 용량과 상이하고 제 5 용량은 제 2 용량과 상이하다. 추가적으로, 제 4 용량 및 제 5 용량의 조합은 제 3 용량과 같을 수 있다. 블럭들 (1320) 후에, 방법 (1300)은 종료한다. 일부 경우

들에서, 프로세서가 상이한 애플리케이션들 실행간에 스위칭할 때, 상이한 애플리케이션들은 제 1 및 제 2 캐시들에 대하여 동일한 선호되는 용량 할당을 가질 수 있다는 것에 유의한다. 따라서, 프로세서는 만약 상이한 애플리케이션들이 동일한 선호되는 용량 할당을 공유하면 상이한 애플리케이션들 간에 스위칭될 때 용량 할당을 바꾸지 않을 수 있다.

[0064] 이제 도 14로 가서, 필 데이터(filldata)를 포워딩 하기 위한 방법(1400)의 일 실시예가 도시된다. 논의 목적들을 위하여, 이 실시예에 단계들은 순차적인 순서로 도시된다. 이하에 설명되는 방법의 다양한 실시예들에서, 설명되는 하나 이상의 엘리먼트들 동시에, 도시된 것과 다른 상이한 순서로 수행될 수 있거나, 또는 완전히 생략될 수 있다는 것에 주목되어야 한다. 다른 추가 엘리먼트들이 원하는 때 또한 수행될 수 있다. 본 출원에서 설명된 임의의 다양한 시스템들 또는 프로세서들은 방법 (1400)을 구현하도록 구성될 수 있다.

[0065] 프로세서는 제 2 캐시의 제 1 부분으로부터 메모리로 제 1 요청을 발송할 수 있고, 제 1 요청은 제 1 메모리 어드레스를 타겟한다 (블럭 (1405)). 일 실시예에서, 프로세서는 제 2 캐시의 제 1 부분을 누락한 제 1 요청에 응답하여 메모리에 제 1 요청을 발송할 수 있다. 다양한 실시예들에서, 제 1 캐시는 로직상의 L2 캐시일 수 있고, 제 2 캐시는 로직상의 L3 캐시일 수 있다. 다른 실시예들에서, 제 1 및 제 2 캐시들은 멀티-레벨 캐시 계층의 다른 로직상의 레벨들일 수 있다.

[0066] 다음, 메모리로부터 제 1 요청에 대응하는 제 1 필 데이터를 수신하기 전에, 캐시 제어기는 제 1 요청을 메모리로 발송한 다음에 제 1 캐시로 제 2 캐시의 제 1 부분을 재할당할 수 있다 (블럭 (1410)). 그런다음, 캐시 제어기는 메모리로부터 제 1 필 데이터를 수신할 수 있다 (블럭 (1415)). 캐시 제어기는 제 1 메모리 어드레스가 제 2 캐시의 제 2 부분에 재매핑되었는지를 결정할 수 있다 (블럭 (1420)). 제 2 캐시의 제 1 부분의 제 1 캐시로의 재할당은 제 1 메모리 어드레스가 제 2 캐시의 제 2 부분에 재매핑되도록 한다고 이 논의의 목적을 위하여 가정될 수 있다. 제 1 메모리 어드레스가 제 2 캐시의 제 2 부분에 재매핑되었다고 결정된 것에 응답하여, 캐시 제어기는 제 1 필 데이터를 제 2 캐시의 제 2 부분에 포워딩할 수 있다 (블럭 (1425)). 다른 실시예에서, 캐시 제어기는 제 1 필 데이터를 제 2 캐시의 제 2 부분으로 포워딩하기 보다는 제 1 필 데이터를 드랍(drop)할 수 있다. 블럭들 (1425) 후에, 방법 (1400)은 종료한다.

[0067] 이제 도 15로 가서, 물리적 캐시의 일부를 재지정하기 위한 방법(1500)의 일 실시예가 도시된다. 논의 목적들을 위하여, 이 실시예에 단계들은 순차적인 순서로 도시된다. 이하에 설명되는 방법의 다양한 실시예들에서, 설명되는 하나 이상의 엘리먼트들 동시에, 도시된 것과 다른 상이한 순서로 수행될 수 있거나, 또는 완전히 생략될 수 있다는 것에 주목되어야 한다. 다른 추가 엘리먼트들이 원하는 때 또한 수행될 수 있다. 본 출원에서 설명된 임의의 다양한 시스템들 또는 프로세서들은 방법 (1500)을 구현하도록 구성될 수 있다.

[0068] 소정의 시간 지점에서, 프로세서는 물리적 캐시의 제 1 부분을 제 1 로직상의 캐시의 일부로서 이용할 수 있다 (블럭 (1505)). 일 실시예에서, 물리적 캐시는 제 1 로직상의 캐시 및 제 2 로직상의 캐시를 포함할 수 있고, 제 1 로직상의 캐시는 L2 캐시를 나타내고 제 2 로직상의 캐시는 L3 캐시(또는 L3캐시의 슬라이스)를 나타낸다. 실시예에 따라, 제 1 부분은 제 1 웨이(way), 제 1 세트(set), 제 1 뱅크(bank), 또는 물리적 캐시의 다른 부분일 수 있다. 하나 이상의 상태들의 감지 또는 이벤트에 응답하여, 프로세서는 물리적 캐시의 제 1 부분을 제 2 로직상의 캐시로 동적으로 재지정할 수 있다 (블럭 (1510)). 실시예에 따라, 이벤트는 제 1 애플리케이션으로부터 제 2 애플리케이션으로 스위칭될 수 있고, 캐시 히트율 검출이 임계값보다 크거나 또는 작고, 캐시 히트율 간에 차이 값 검출이 임계값을 초과하고, 및/또는 하나 이상의 다른 상태들일 수 있다.

[0069] 다음, 프로세서는 물리적 캐시의 제 1 부분을 제 2 로직상의 캐시의 일부로서 이용할 수 있다 (블럭 (1515)). 예를 들어, 프로세서는 블럭(1515)에서 물리적 캐시의 제 1 부분에 제 2 로직상의 캐시에 의해 수신된 캐시 라인을 저장할 수 있다. 블럭 (1515) 후에, 방법 (1500)은 종료할 수 있다. 프로세서는 프로세서의 동작의 장시간 기간동안 여러번 제 1 및 제 2 로직상의 캐시간에 용량을 동적으로 할당할 수 있고, 물리적 캐시의 제 1 부분은 여러번 제 1 및 제 2 로직상의 캐시 사이에서 왔다 갔다할 수 있다는 것에 유의한다.

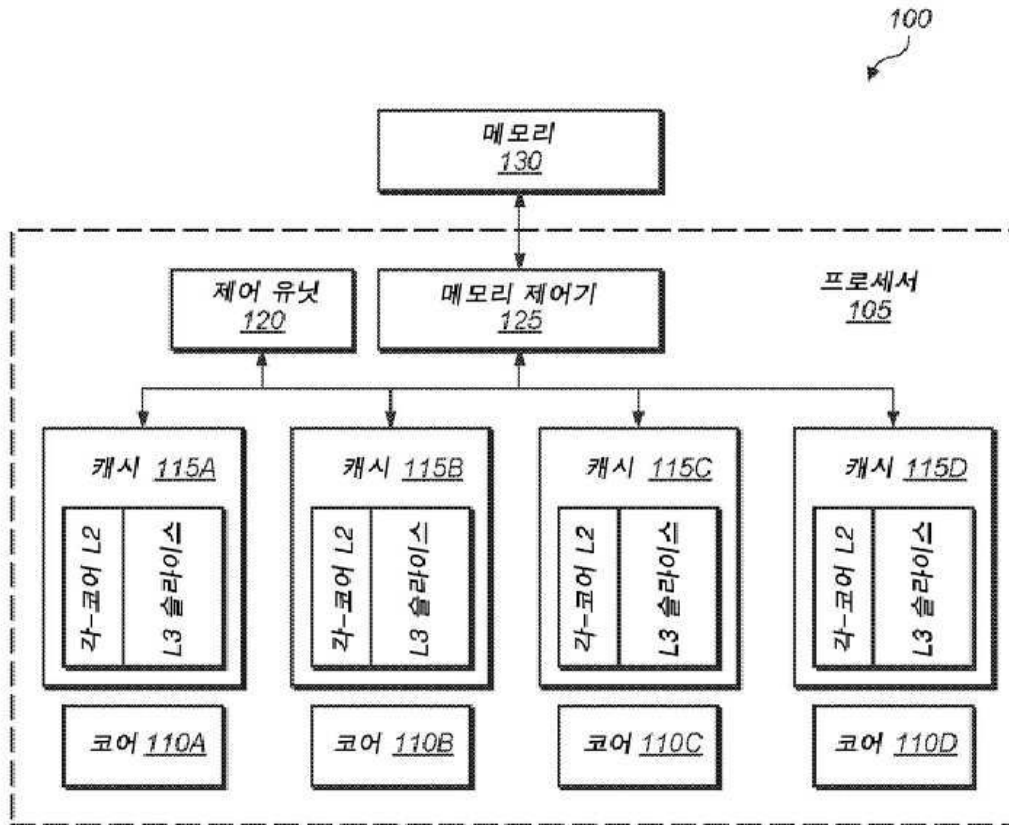
[0070] 다양한 실시예들에서, 소프트웨어 애플리케이션의 프로그램 지시들이 앞에서 설명된 방법들 및/또는 메커니즘들을 구현하기 위해 사용될 수 있다. 프로그램 지시들은 하이 레벨 프로그래밍 언어, 예컨대 C로 하드웨어의 행위를 설명할 수 있다. 대안적으로, 하드웨어 디자인 언어 (HDL)가 예컨대 Verilog가 사용될 수 있다. 프로그램 지시들은 비-일시적 컴퓨터 판독가능한 스토리지 매체상에 저장될 수 있다. 많은 유형들의 스토리지 매체가 이용 가능하다. 스토리지 매체는 프로그램 실행을 위해 데이터를 컴퓨팅 시스템에 첨부하고 프로그램 지시들을 제공하기 위해 사용동안에 컴퓨팅 시스템에 의해 액세스 가능할 수 있다. 컴퓨팅 시스템은 적어도 하나의 또는 그 이상 메모리들 및 프로그램 지시들을 실행하도록 구성된 하나 이상의 프로세서들을 포함할 수 있다.

[0071]

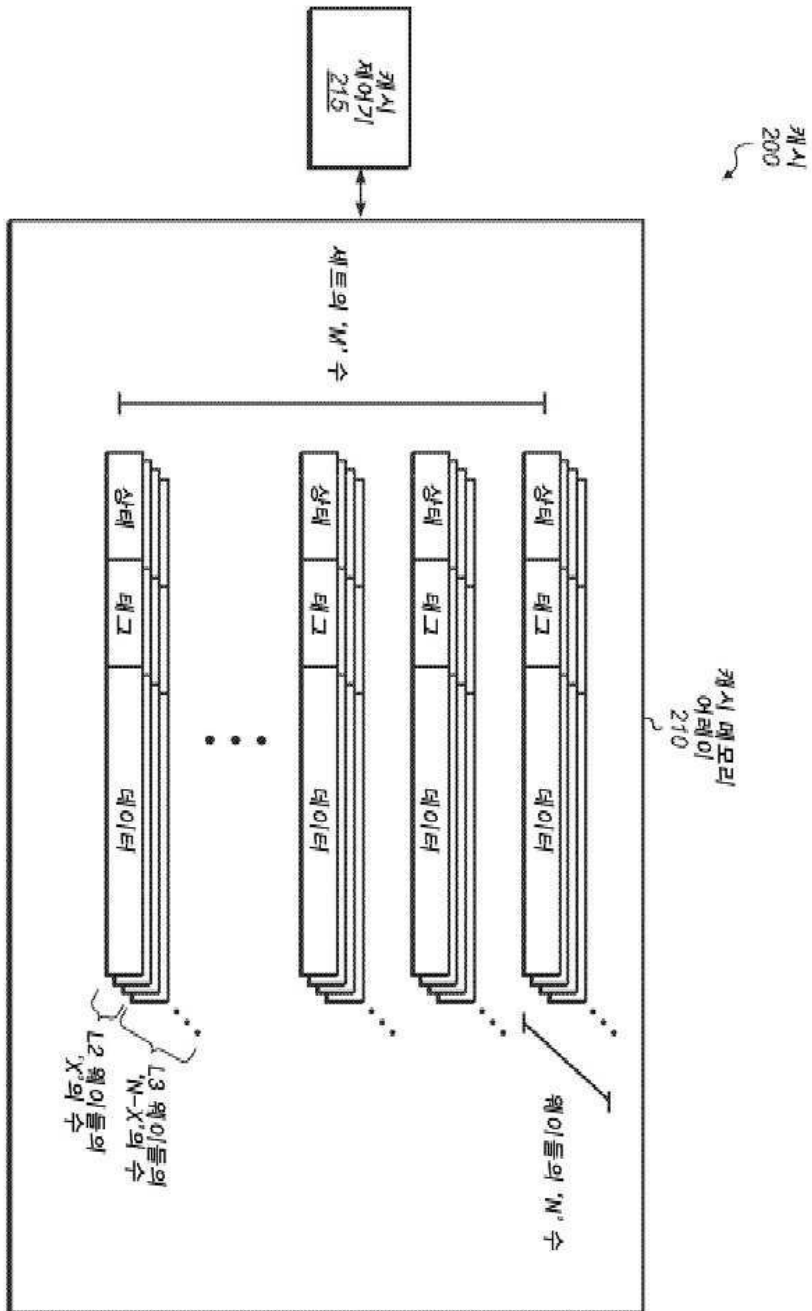
상기-설명된 실시예들은 단지 구현예들의 비 제한적인 예제인 것이 강조되어야 한다. 상기의 개시가 완전히 이해되면 많은 변형예들 및 수정예들은 당해 기술분야의 통상의 기술자들에 명확해질 것이다. 이하의 청구항들은 모든 이런 변형예들 및 수정예들을 아우르도록 해석되도록 의도된다.

도면

도면1



도면2



도면3

에플리케이션	선호 L2 사이즈	선호 L3 사이즈
첫째	128 KB	896 KB
두번째	384 KB	640 KB
세번째	256 KB	768 KB
...	...	...

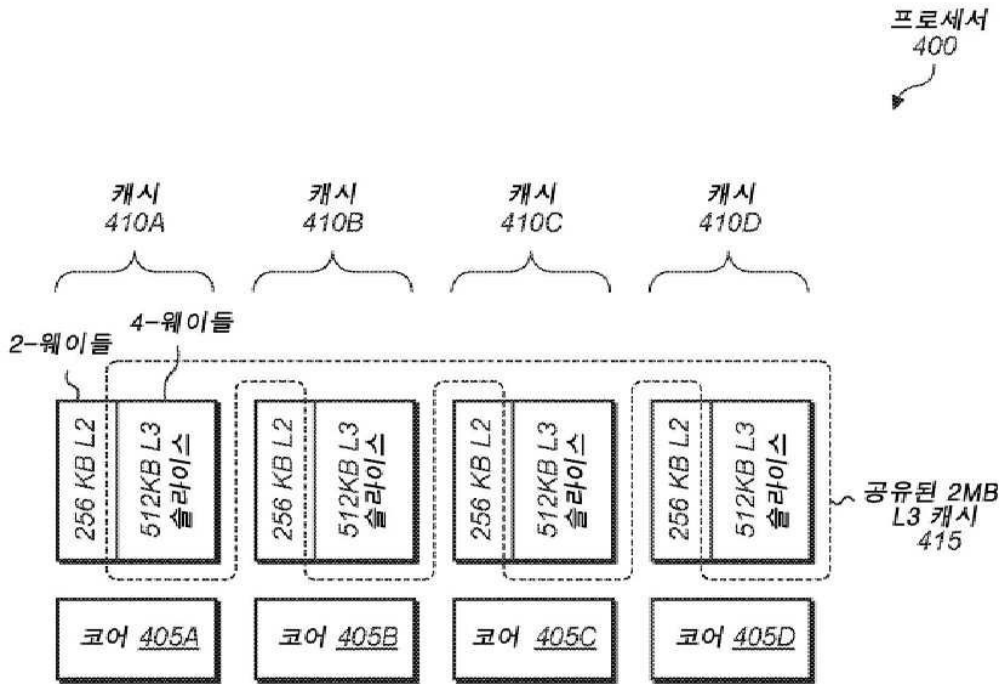
데이터  
310A

제어 유닛 300
제어 유닛 유닛 305
데이터(들) 310

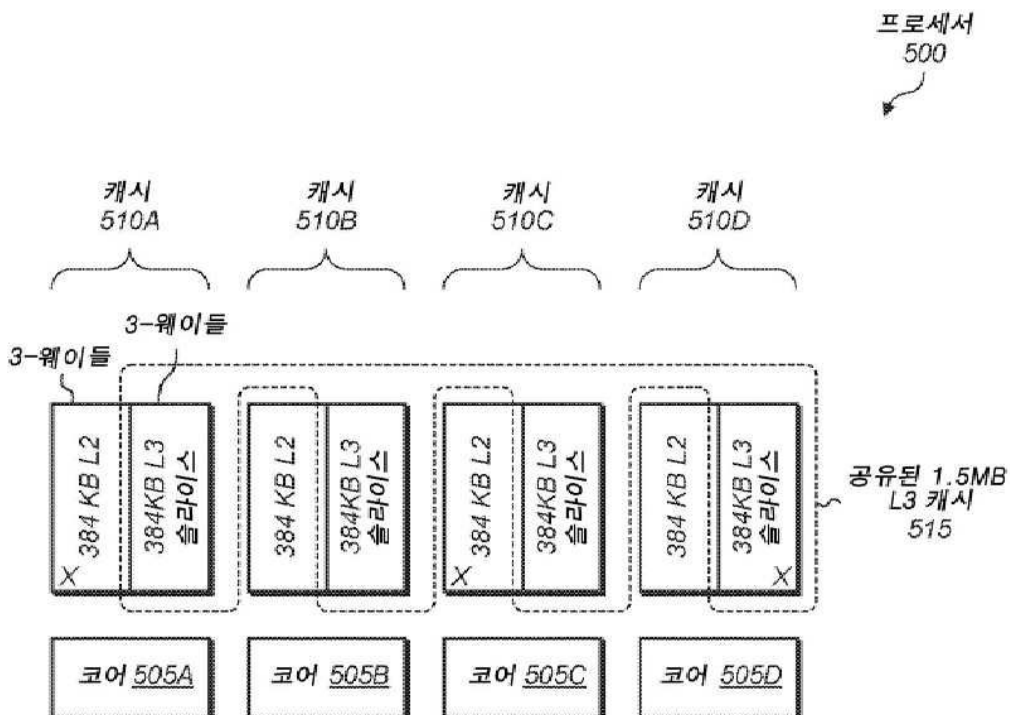
파라미터	현재 값	임계 값	임계값 초과?
제 1 캐시 히트율	36%	50%	아니오
제 2 캐시 히트율	62%	70%	아니오
제 1 캐시 히트율과 제 2 캐시 히트율간의 차이	26%	15%	예
마이크로세션 비율	18%	35%	아니오
...	...	...	...

데이터  
310B

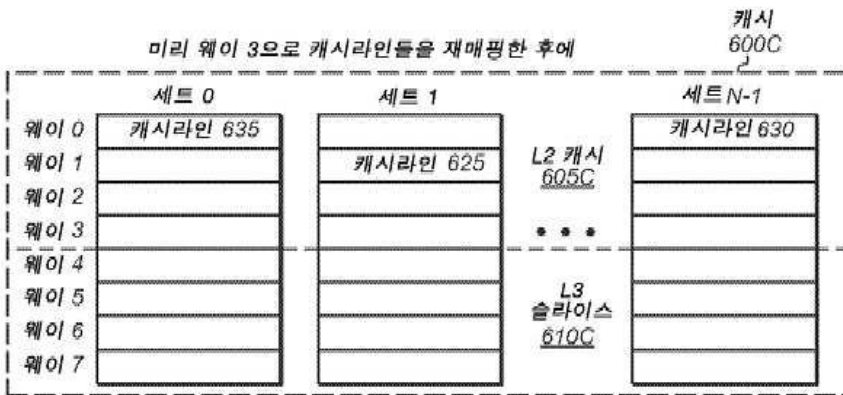
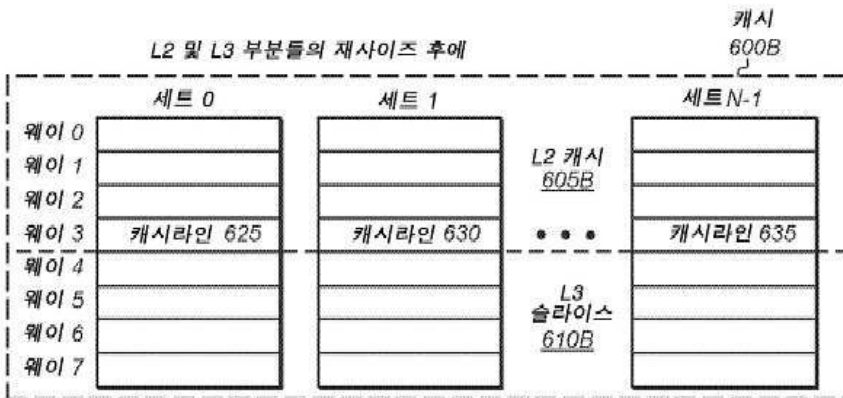
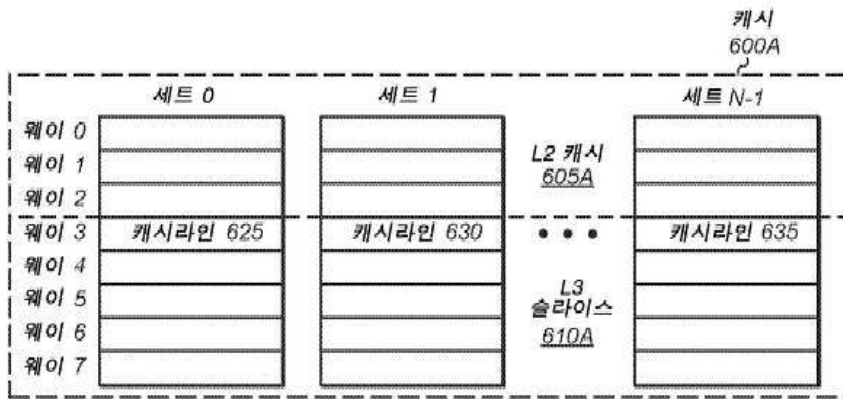
도면4



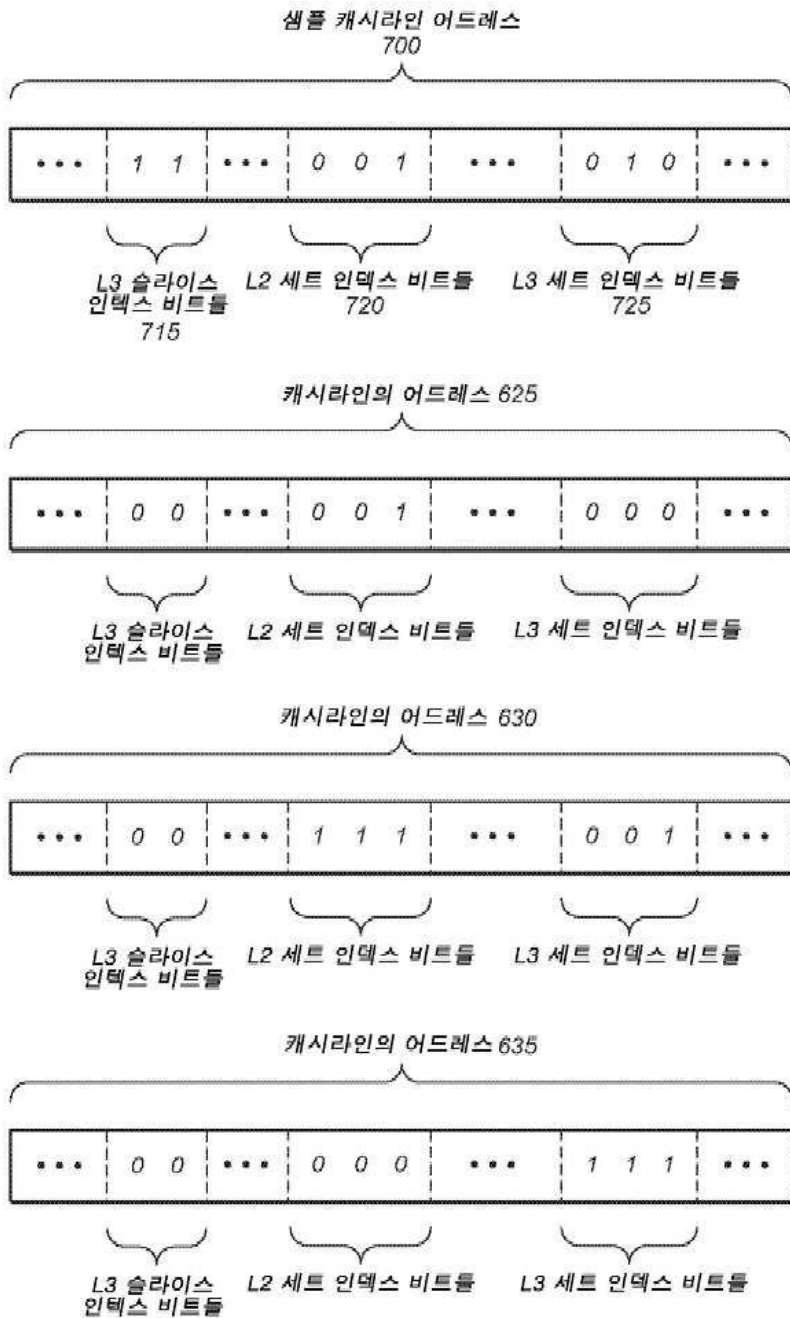
도면5



도면6

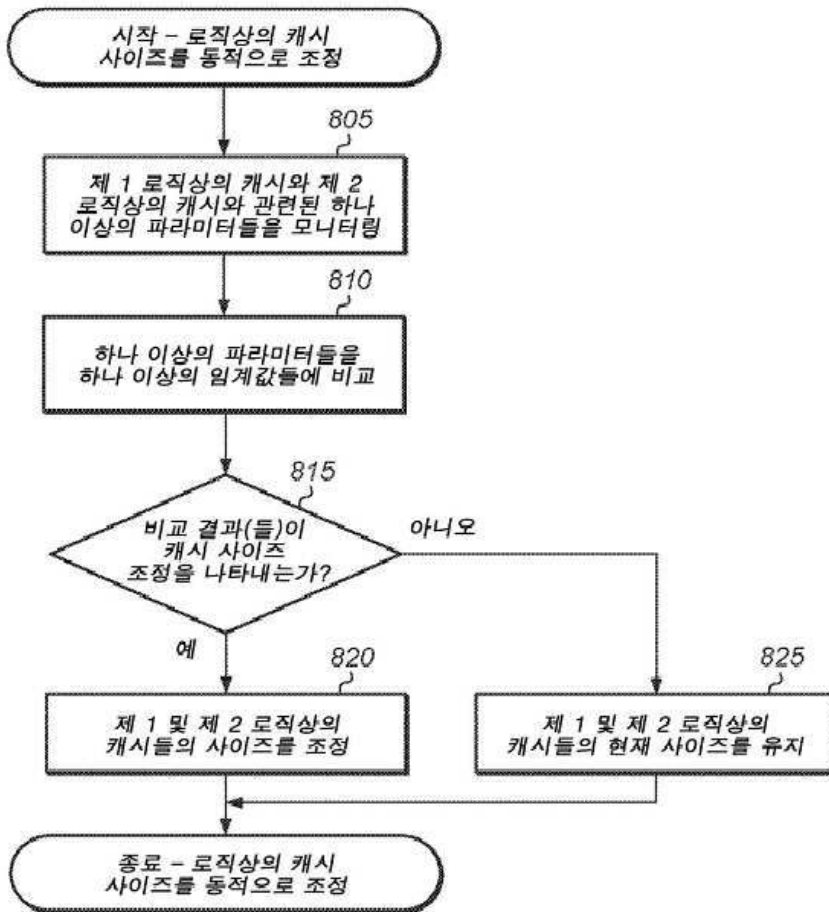


도면7



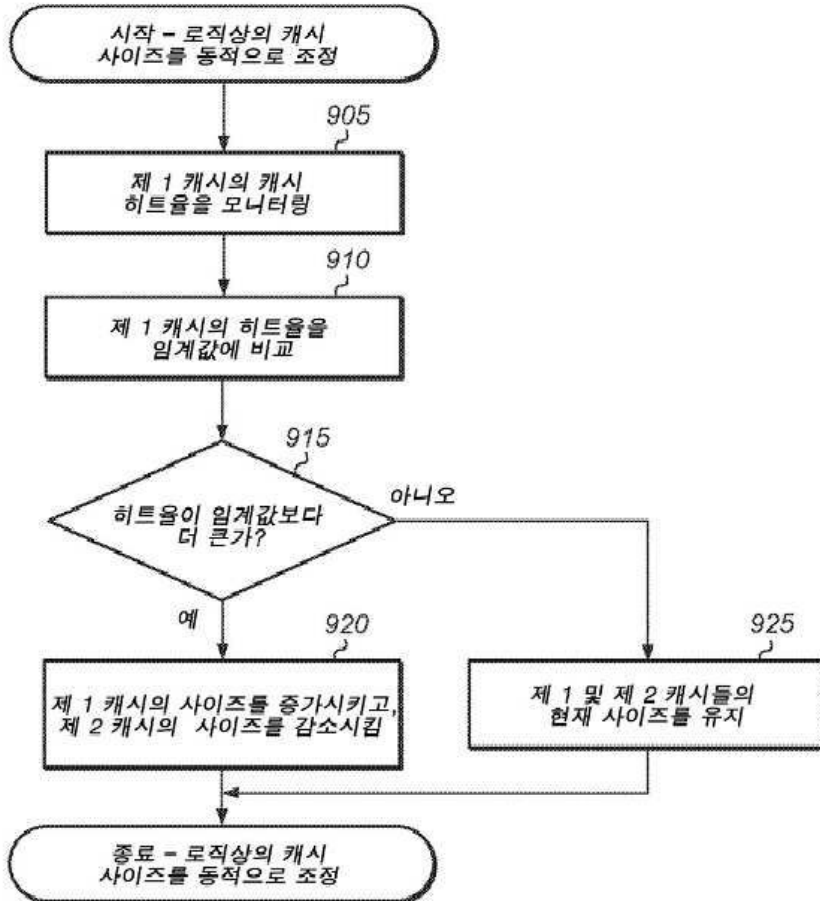
도면8

800



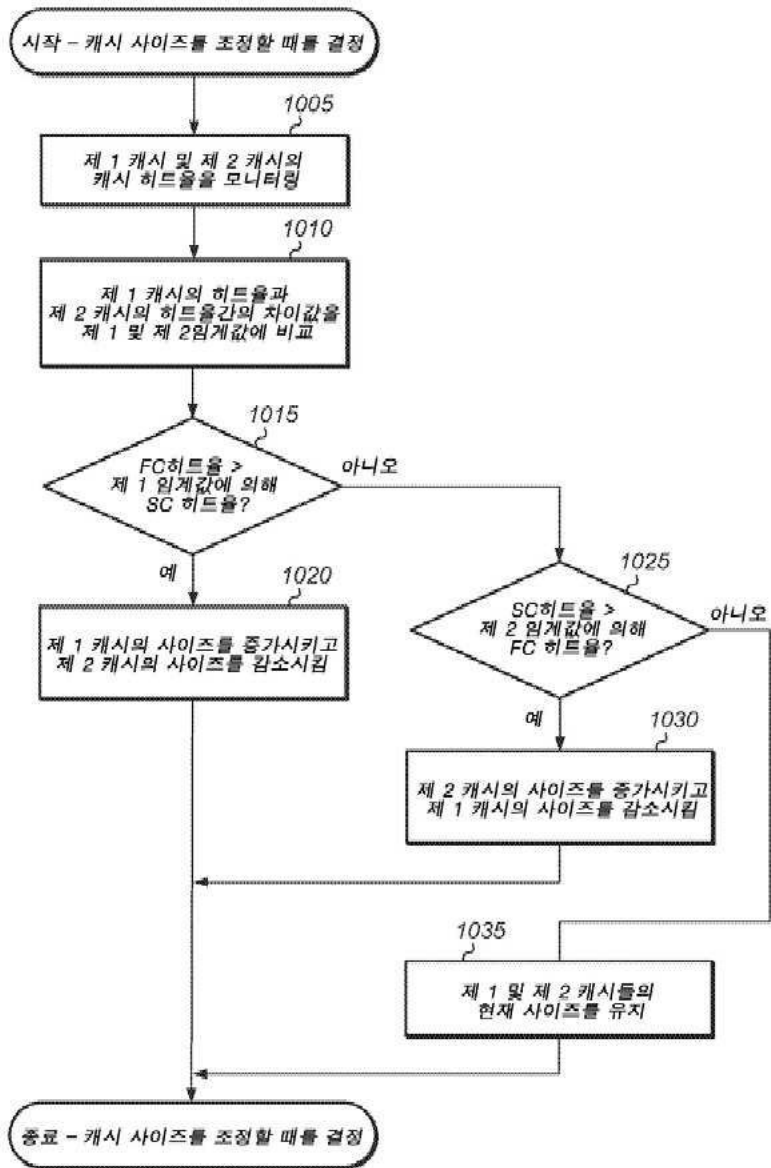
도면9

900



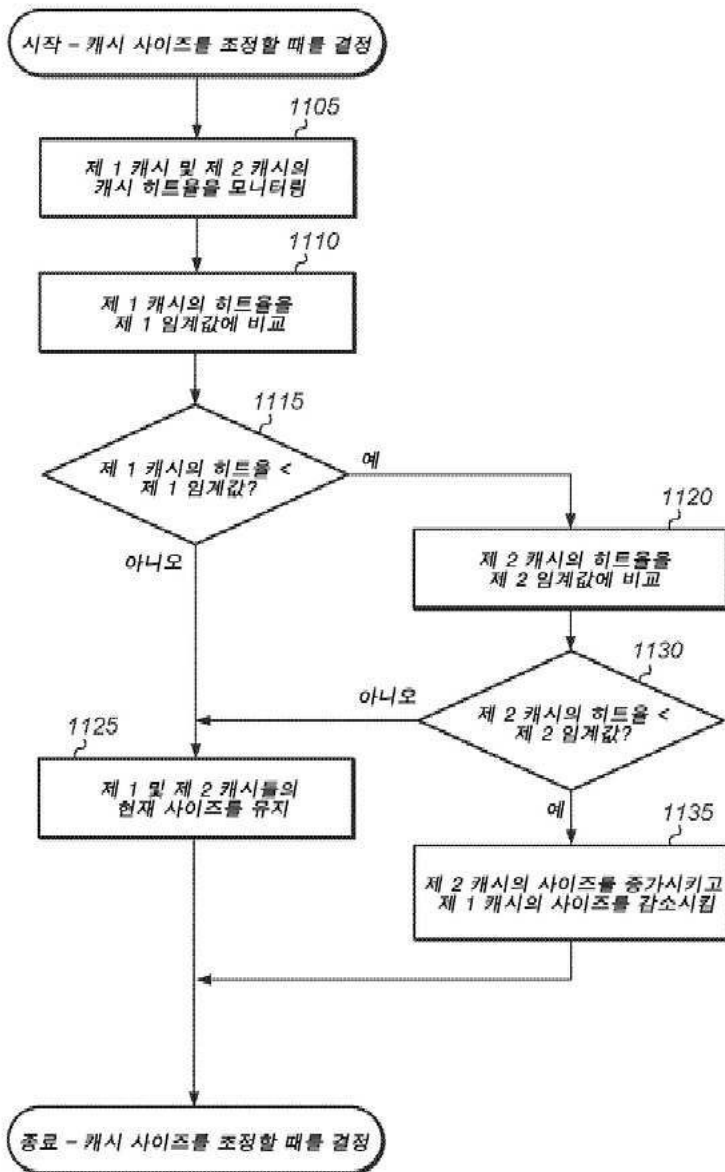
도면10

1000

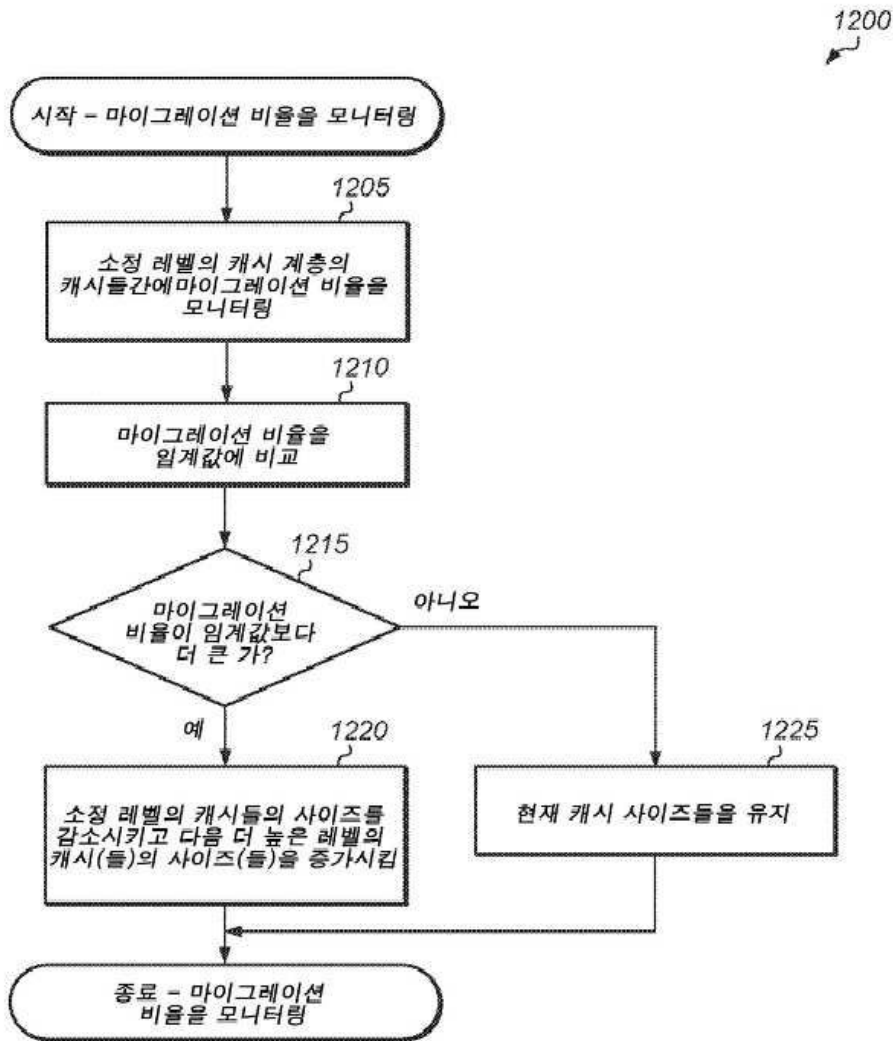


도면11

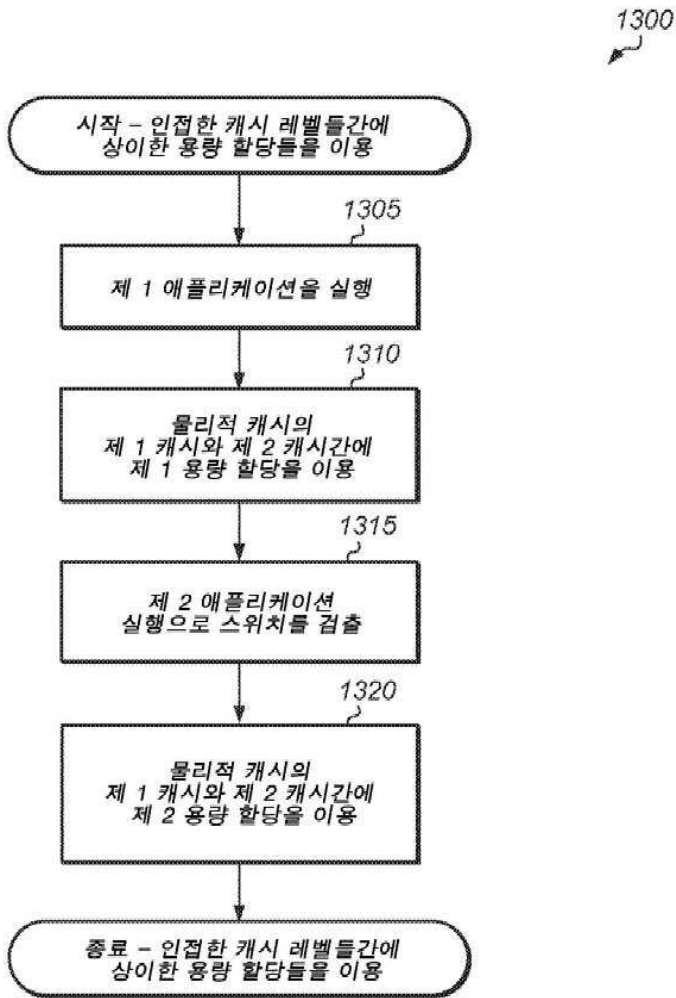
1100



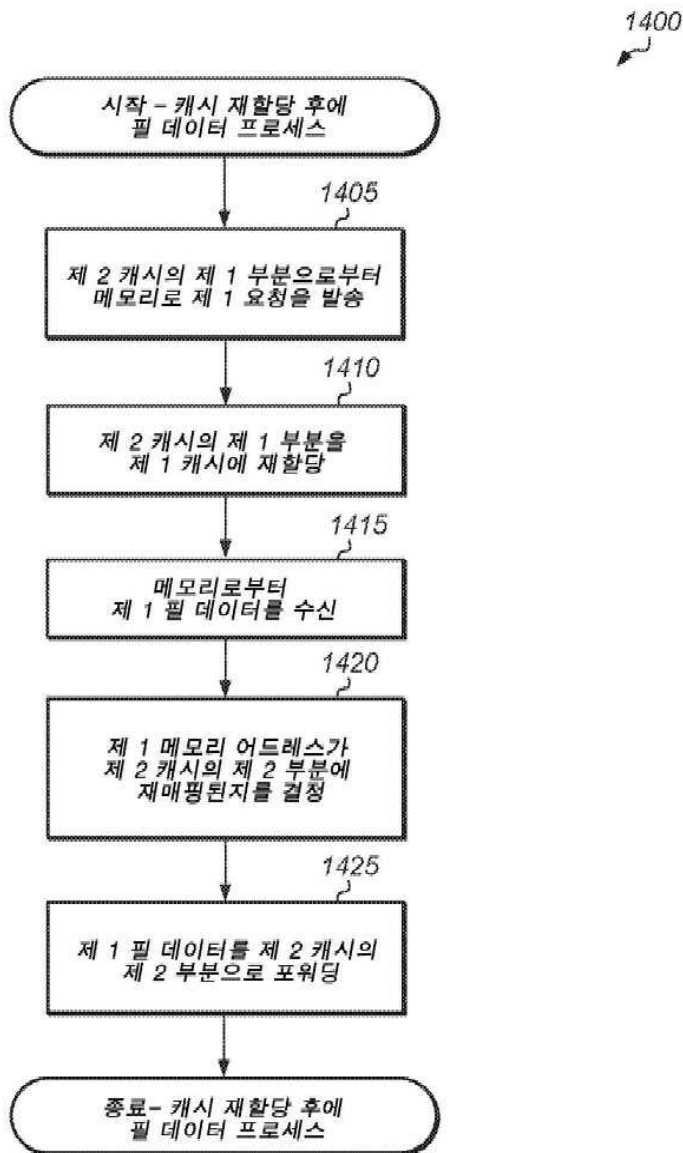
도면12



도면13



도면14



도면15

