



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년03월24일

(11) 등록번호 10-1503865

(24) 등록일자 2015년03월12일

(51) 국제특허분류(Int. Cl.)

G06F 12/08 (2006.01) G06F 12/06 (2006.01)
G06F 9/38 (2006.01)

(21) 출원번호 10-2009-7026820

(22) 출원일자(국제) 2008년05월28일

심사청구일자 2013년05월21일

(85) 번역문제출일자 2009년12월22일

(65) 공개번호 10-2010-0022483

(43) 공개일자 2010년03월02일

(86) 국제출원번호 PCT/US2008/006746

(87) 국제공개번호 WO 2008/153799

국제공개일자 2008년12월18일

(30) 우선권주장

11/754,589 2007년05월29일 미국(US)

(56) 선행기술조사문헌

US05826052 A*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 94088-3453 서니베일 피.오.박스
3453 원 에이엠디 플레이스

(72) 발명자

라우터비호 케리

미국 캘리포니아 94022 로스 알토스 엘레나 로드
25712

홀로웨이 브루스 알.

미국 캘리포니아 95006 볼더 크릭 엘시 매 드라이
브 891

(뒷면에 계속)

(74) 대리인

박장원

전체 청구항 수 : 총 20 항

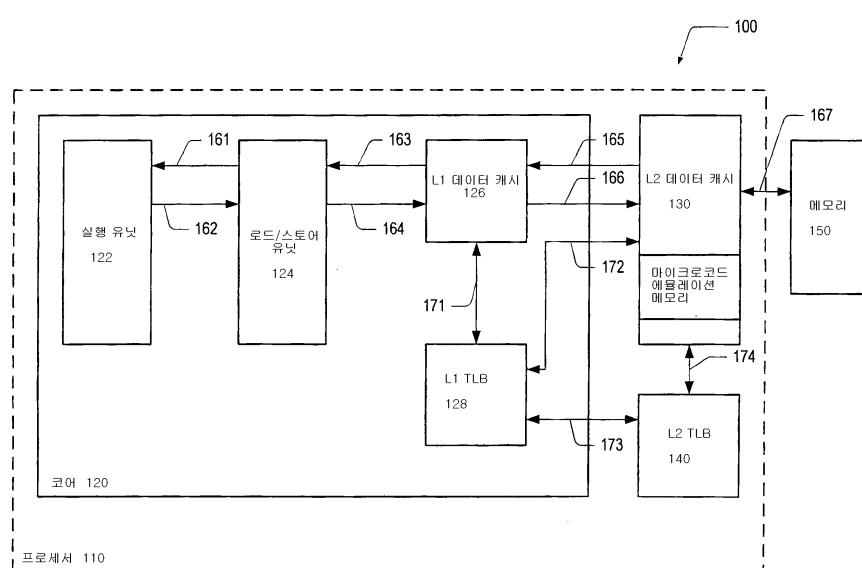
심사관 : 이명진

(54) 발명의 명칭 마이크로코드 애플레이션 메모리의 캐싱

(57) 요 약

프로세서는, 레벨-1 캐시와 상위-레벨 캐시를 포함하는 캐시 계층(cache hierarchy)을 포함한다. 프로세서는, 물리적 메모리 스페이스의 일부분을 상위-레벨 캐시의 일부분에 매핑시키며, 명령어들을 실행시키며, 상기 명령어들 중 적어도 일부는 마이크로코드를 포함하며, 마이크로코드가 상위-레벨 캐시의 상기 일부분에 액세스하는 것

(뒷면에 계속)

대 표 도 - 도1

을 허용하며; 그리고 마이크로코드를 포함하고 있지 않은 명령어들이 상위-레벨 캐시의 상기 일부분에 액세스하는 것을 금지한다. 물리적 메모리 스페이스의 제 1 부분은 마이크로코드에 의한 사용을 위해 영구 할당될 수 있다. 상기 프로세서는, 또한 상위-레벨 캐시의 제 1 부분의 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 상기 레벨-1 제 1 캐시의 제 1 부분으로 이동시키며, 마이크로코드가 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 허용하며, 그리고 마이크로코드를 포함하고 있지 않은 명령어들이 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 금지한다.

(72) 발명자

버틀러 마이클 제라드

미국 캘리포니아 95135 산호세 런닝 스프링 로드
6002

리에 선

미국 캘리포니아 95051 산타 클라라 그라나다 애버뉴 #203 3500

특허청구의 범위

청구항 1

프로세서로서,

레벨-1 제 1 캐시와 상위-레벨 캐시를 적어도 포함하는 캐시 계층(cache hierarchy)을 포함하여 이루어지며,

상기 프로세서는,

물리적 메모리 스페이스의 제 1 부분을 상기 상위-레벨 캐시의 제 1 부분에 매핑시키며;

명령어들을 실행시키며, 상기 명령어들 중 적어도 일부는 마이크로코드를 포함하며;

상기 명령어들 중 제 1 명령어를 디코딩하며;

상기 제 1 명령어에 대한 상기 디코딩에 응답하여 마이크로코드 액세스 신호가 검출된다면, 상기 제 1 명령어가 마이크로코드를 나타낸다라고 결정하고 그리고 상기 제 1 명령어가 상기 상위-레벨 캐시의 상기 제 1 부분에 액세스하는 것을 허용하고; 그리고

상기 제 1 명령어에 대한 상기 디코딩에 응답하여 마이크로코드 액세스 신호가 검출되지 않는다면, 상기 제 1 명령어가 마이크로코드를 나타내지 않는다라고 결정하고 그리고 상기 제 1 명령어가 상기 상위-레벨 캐시의 상기 제 1 부분에 액세스하는 것을 금지하도록 된 것을 특징으로 하는 프로세서.

청구항 2

제1항에 있어서,

상기 상위-레벨 캐시는 레벨-2 캐시인 것을 특징으로 하는 프로세서.

청구항 3

제1항에 있어서,

상기 물리적 메모리 스페이스의 제 1 부분은 마이크로코드에 의한 사용을 위해 영구 할당되는 것을 특징으로 하는 프로세서.

청구항 4

제1항에 있어서,

상기 프로세서는, 또한

상기 상위-레벨 캐시의 제 1 부분의 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 상기 레벨-1 제 1 캐시의 제 1 부분으로 이동시키며;

마이크로코드가 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 허용하며; 그리고

마이크로코드를 포함하고 있지 않는 명령어들이 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 금지하도록 된 것을 특징으로 하는 프로세서.

청구항 5

제1항에 있어서,

상기 프로세서는 또한,

상기 제 1 명령어의 하나 이상의 비트들이 기결정된 상태를 가짐을 검출함으로써 상기 디코딩 동안 상기 제 1 명령어가 마이크로코드를 나타낸다라고 결정하도록 된 것을 특징으로 하는 프로세서.

청구항 6

제5항에 있어서,

변환 색인 버퍼(translation lookaside buffer : TLB)를 더 포함하며,

마이크로코드를 포함하고 있지 않는 명령어들이 상기 물리적 메모리 스페이스의 제 1 부분에 액세스하는 것을 금지하는 것은,

상기 물리적 메모리 스페이스의 제 1 부분으로의 TLB 리필들(refills)을 허용하지 않는 것을 특징으로 하는 프로세서.

청구항 7

제1항에 있어서,

제 1 코어와 제 2 코어를 적어도 더 포함하며,

상기 제 2 코어는,

상기 물리적 메모리 스페이스의 제 2 부분을 상기 상위-레벨 캐시의 제 2 부분에 매핑하며;

명령어들을 실행시키며, 상기 명령어들 중 적어도 일부는 마이크로코드를 포함하며;

마이크로코드가 상기 상위-레벨 캐시의 제 2 부분에 액세스하는 것을 허용하며; 그리고

마이크로코드를 포함하고 있지 않는 명령어들이 상기 상위-레벨 캐시의 제 2 부분에 액세스하는 것을 금지하도록 된 것을 특징으로 하는 프로세서.

청구항 8

제7항에 있어서,

상기 제 1 코어는 상기 레벨-1 제 1 캐시를 포함하며 그리고 상기 제 2 코어는 레벨-1 제 2 캐시를 포함하며,

상기 제 1 코어는, 또한,

상기 상위-레벨 캐시의 제 1 부분에 매핑된 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 상기 레벨-1 제 1 캐시의 일부분으로 이동시키며;

마이크로코드가 상기 레벨-1 제 1 캐시의 상기 일부분에 액세스하는 것을 허용하며;

마이크로코드를 포함하고 있지 않은 명령어들이 상기 레벨-1 제 1 캐시의 상기 일부분에 액세스하는 것을 금지하도록 되어 있으며,

상기 제 2 코어는, 또한,

상기 상위-레벨 캐시의 제 2 부분에 매핑된 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 상기 레벨-1 제 2 캐시의 일부분으로 이동시키며;

마이크로코드가 상기 레벨-1 제 2 캐시의 상기 일부분에 액세스하는 것을 허용하며;

마이크로코드를 포함하고 있지 않은 명령어들이 상기 레벨-1 제 2 캐시의 상기 일부분에 액세스하는 것을 금지하도록 된 것을 특징으로 하는 프로세서.

청구항 9

물리적 메모리 스페이스의 제 1 부분을 프로세서의 캐시 계층의 상위-레벨 캐시의 제 1 부분에 매핑시키는 단계;

제 1 프로세서 코어에 의해 명령어들을 실행하는 단계, 상기 명령어들 중 적어도 일부는 마이크로코드를 포함하며;

상기 명령어들 중 제 1 명령어를 디코딩하는 단계;

상기 제 1 명령어에 대한 상기 디코딩에 응답하여 마이크로코드 액세스 신호가 검출된다면, 상기 제 1 명령어가 마이크로코드를 나타낸다라고 결정하고 그리고 상기 제 1 명령어가 상기 상위-레벨 캐시의 상기 제 1 부분에 액세스하는 것을 허용하는 단계; 그리고

상기 제 1 명령어에 대한 상기 디코딩에 응답하여 마이크로코드 액세스 신호가 검출되지 않는다면, 상기 제 1 명령어가 마이크로코드를 나타내지 않는다라고 결정하고 그리고 상기 제 1 명령어가 상기 상위-레벨 캐시의 상

기 제 1 부분에 액세스하는 것을 금지하는 단계

를 포함하는 것을 특징으로 하는 방법.

청구항 10

제9항에 있어서,

상기 상위-레벨 캐시는 레벨-2 캐시인 것을 특징으로 하는 방법.

청구항 11

제9항에 있어서,

상기 물리적 메모리 스페이스의 제 1 부분을 마이크로코드에 의한 사용을 위해 영구 할당하는 단계

를 더 포함하는 것을 특징으로 하는 방법.

청구항 12

제9항에 있어서,

상기 상위-레벨 캐시의 제 1 부분의 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 상기 프로세서의 레벨-1 제 1 캐시의 제 1 부분으로 이동시키는 단계;

마이크로코드가 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 허용하는 단계; 그리고

마이크로코드를 포함하고 있지 않는 명령어들이 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 금지하는 단계

를 더 포함하는 것을 특징으로 하는 방법.

청구항 13

제9항에 있어서,

상기 제 1 명령어의 하나 이상의 비트들이 기결정된 상태를 가짐을 검출함으로써 상기 디코딩 동안 상기 제 1 명령어가 마이크로코드를 나타낸다라고 결정하는 단계

를 더 포함하는 것을 특징으로 하는 방법.

청구항 14

제13항에 있어서,

상기 마이크로코드 액세스 신호가 어서트(assert) 되지 않는다면, 상기 물리적 메모리 스페이스의 제 1 부분에 명령어들이 액세스하는 것을 금지하는 단계는,

변환 색인 버퍼(TLB)로부터 상기 물리적 메모리 스페이스의 제 1 부분으로의 리필들(refills)을 허용하지 않는 것을 특징으로 하는 방법.

청구항 15

제9항에 있어서,

상기 물리적 메모리 스페이스의 제 2 부분을 상기 상위-레벨 캐시의 제 2 부분에 매핑하는 단계;

제 2 프로세서 코어에 의해 명령어들을 실행하는 단계, 상기 명령어들 중 적어도 일부는 마이크로코드를 포함하며;

상기 제 2 프로세서 코어에 의해, 마이크로코드가 상기 상위-레벨 캐시의 상기 제 2 부분에 액세스하는 것을 허용하는 단계; 그리고

상기 제 2 프로세서 코어에 의해, 마이크로코드를 포함하고 있지 않는 명령어들이 상기 상위-레벨 캐시의 상기 제 2 부분에 액세스하는 것을 금지하는 단계

를 더 포함하는 것을 특징으로 하는 방법.

청구항 16

제15항에 있어서,

상기 상위-레벨 캐시의 제 1 부분에 매핑된 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 상기 레벨-1 제 1 캐시의 일부분으로 이동시키는 단계;

상기 상위-레벨 캐시의 제 2 부분에 매핑된 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 레벨-1 제 2 캐시의 일부분으로 이동시키는 단계;

마이크로코드가 상기 레벨-1 제 1 캐시의 상기 일부분 및/또는 상기 레벨-1 제 2 캐시의 상기 일부분에 액세스하는 것을 허용하는 단계;

마이크로코드를 포함하고 있지 않은 명령어들이 상기 레벨-1 제 1 캐시의 상기 일부분에 액세스하는 것을 금지하는 단계; 그리고

마이크로코드를 포함하고 있지 않은 명령어들이 상기 레벨-1 제 2 캐시의 상기 일부분에 액세스하는 것을 금지하는 단계

를 더 포함하는 것을 특징으로 하는 방법.

청구항 17

컴퓨터 시스템으로서,

메모리와 하나 이상의 주변 디바이스들에 접속된 적어도 하나의 프로세서

를 포함하며,

상기 적어도 하나의 프로세서는 레벨-1 제 1 캐시와 상위-레벨 캐시를 적어도 포함하는 캐시 계층을 포함하며,

상기 적어도 하나의 프로세서는,

물리적 메모리 스페이스의 제 1 부분을 상기 상위-레벨 캐시의 제 1 부분에 매핑시키며;

명령어들을 실행시키며, 상기 명령어들 중 적어도 일부는 마이크로코드를 포함하며;

상기 명령어들 중 제 1 명령어를 디코딩하며;

상기 제 1 명령어에 대한 상기 디코딩에 응답하여 마이크로코드 액세스 신호가 검출된다면, 상기 제 1 명령어가 마이크로코드를 나타낸다라고 결정하고 그리고 상기 제 1 명령어가 상기 상위-레벨 캐시의 상기 제 1 부분에 액세스하는 것을 허용하고; 그리고

상기 제 1 명령어에 대한 상기 디코딩에 응답하여 마이크로코드 액세스 신호가 검출되지 않는다면, 상기 제 1 명령어가 마이크로코드를 나타내지 않는다라고 결정하고 그리고 상기 제 1 명령어가 상기 상위-레벨 캐시의 상기 제 1 부분에 액세스하는 것을 금지하도록 된 것을 특징으로 하는 컴퓨터 시스템.

청구항 18

제17항에 있어서,

상기 적어도 하나의 프로세서는, 또한,

상기 상위-레벨 캐시의 제 1 부분의 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 상기 레벨-1 제 1 캐시의 제 1 부분으로 이동시키며;

마이크로코드가 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 허용하며; 그리고

마이크로코드를 포함하고 있지 않는 명령어들이 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 금지하도록 된 것을 특징으로 하는 컴퓨터 시스템.

청구항 19

제17항에 있어서,

상기 적어도 하나의 프로세서는, 또한,

상기 제 1 명령어의 하나 이상의 비트들이 기결정된 상태를 가짐을 검출함으로써 상기 디코딩 동안 상기 제 1 명령어가 마이크로코드를 나타낸다라고 결정하도록 된 것을 특징으로 하는 컴퓨터 시스템.

청구항 20

제17항에 있어서,

상기 적어도 하나의 프로세서는 변환 색인 버퍼(TLB)를 더 포함하며,

마이크로코드를 포함하고 있지 않는 명령어들이 상기 물리적 메모리 스페이스의 제 1 부분에 액세스하는 것을 금지하기 위해, 상기 적어도 하나의 프로세서는 상기 물리적 메모리 스페이스의 제 1 부분으로의 TLB 리필들(refills)을 허용하지 않는 것을 특징으로 하는 컴퓨터 시스템.

명세서

기술 분야

[0001] 본 발명은 마이크로프로세서에 관한 것이며, 좀더 상세하게는 마이크로코드에 의한 복합 명령어(complex instruction)의 에뮬레이션에 관한 것이며, 더욱 상세하게는 이러한 에뮬레이션 동안에 이용되는 메모리의 캐싱(caching)에 관한 것이다.

배경 기술

[0002] 비록, 마이크로프로세서가 복합 명령어 집합 컴퓨터(complex instruction set computer : CISC) 아키텍처와 호환성을 유지하는 것이 바람직하지만, 다른 아키텍처들은 개선된 실행 속도 및 성능을 제공한다. 마이크로프로세서 설계자들은, CISC 명령어들을 에뮬레이팅함에 의한 고성능과 CISC 호환성 둘다를 획득하기 위해서 노력해 왔다. 예를 들어, 슈퍼 스칼라, 축소 명령어 집합 컴퓨터(reduced instruction set computer: RISC) 아키텍처는 CISC 명령어 에뮬레이션을 수행하는 마이크로코드를 포함할 수 있다. 에뮬레이션 프로세스 동안, 마이크로코드는 중간 값(intermediate value)을 세이빙(saving)하기 위해서 스크래치패드 메모리를 이용한다. 고성능을 유지하기 위해서는, 마이크로프로세서의 마이크로코드는 에뮬레이션 메모리에 가능한한 빠르게 액세스할 수 있는 것이 바람직하다.

[0003] 또한, 마이크로프로세서는, 계층적으로 정렬되어 있으며 그리고 복수개의 코어들 또는 실행 유닛(execution unit)들에 의해 공유되는 복수개의 메모리 캐시들을 포함하는 것이 통상적이다. 다양한 캐싱 아키텍처들이 이용되며 그리고 온-칩 캐시 및 오프-칩 캐시의 다양한 조합들을 포함한다. 캐시 또는 메모리로부터 데이터를 판독하는 메모리 동작들은, 본 명세서에서 좀더 간결하게 '로드(load)'라고 지칭될 수 있다. 캐시 또는 메모리에 데이터를 기입하는 메모리 동작들은, 본 명세서에서 좀더 간결하게 '스토어(store)'라고 지칭될 수 있다. 로드 또는 스토어는 특정 캐시 라인(또는 캐시 라인의 일부)을 타겟으로 할 수도 있으며 그리고 캐시 라인으로부터 로드되거나 또는 캐시 라인에 스토어될 데이터를 포함할 뿐만 아니라 타겟팅된 라인을 식별하는 어드레스를 포함할 수도 있다. 캐시 액세스가 메모리 액세스 보다 빠르기 때문에, 코어 또는 실행 유닛이 캐시에 액세스할 필요가 있을 때에 데이터가 캐시 내에 위치할 가능성(likelihood)을 증가시키기 위해서 다양한 캐싱 테크닉들이 이용되는바, 따라서 실행 속도를 향상시킬 수 있다. 결과적으로, 마이크로코드 에뮬레이션 메모리를 캐싱하는 것은, 시스템 메모리에 비하여 상대적으로 빠른 캐시 메모리의 액세스 시간의 성능 이점을 제공한다. 일반적으로 가장 짧은 액세스 시간들은, 통상적으로는 L1-캐시라 지칭되며 간략하게는 L1이라 지칭되는, 캐시 계층의 최하위 레벨에 관련된 것들이다. 그러므로, 마이크로코드 에뮬레이션 메모리를 L1에 캐시하는 것이 바람직하다. L1의 일부를 마이크로코드 에뮬레이션 메모리에 대해 영구 할당함으로써, 이러한 성능상의 이점이 종종 강화되어 왔다.

[0004] 물론, L1-캐시를 이용하는 성능상의 이점들은 다른 프로세스에게도 역시 이익이 될 것이다. 결과적으로, 임의의 프로세스에 대해서 L1-캐시 스페이스의 이용가능성을 증가시키기 위해서, L1-캐시를 가능한한 크게 만드는 것이 바람직하다. 하지만, L1의 사이즈를 증가시키는 것은, 마이크로프로세서의 비용 및 복잡도 역시 증가시킨다. 또한, 마이크로코드 에뮬레이션 메모리가 L1 내에 영구 할당된다면, 다른 프로세스들은 L1의 이 영역을 이용할 수 없다. 이러한 문제점들을 해결하기 위해서 필요한 것은, 마이크로코드 에뮬레이션 메모리를 캐싱하는 장점들을

유지하면서도, 모든 프로세스들에 대해서 소정 사이즈의 L1-캐시 내에서 스페이스의 이용가능성을 개선시키는 것이다.

발명의 상세한 설명

[0005] 프로세서, 컴퓨터 시스템, 및 방법들의 다양한 실시예들이 개시된다. 프로세서는 레벨-1 제 1 캐시와 상위-레벨 캐시를 적어도 포함하는 캐시 계층을 포함한다. 프로세서는 물리적 메모리 스페이스의 제 1 부분을 상위-레벨 캐시의 제 1 부분에 매핑시키며, 명령어들을 실행시키며, 상기 명령어들 중 적어도 일부는 마이크로코드를 포함하며, 마이크로코드가 상위-레벨 캐시의 상기 제 1 부분에 액세스하는 것을 허용하며; 그리고 마이크로코드를 포함하고 있지 않은 명령어들이 상위-레벨 캐시의 상기 제 1 부분에 액세스하는 것을 금지하도록 구성된다. 일 실시예에서, 상위-레벨 캐시는 레벨-2 캐시이다. 다른 실시예에서 물리적 메모리 스페이스의 제 1 부분은 마이크로코드에 의한 사용을 위해 영구 할당될 수 있다.

[0006] 본 발명의 실시예에서, 상기 프로세서는 상기 상위-레벨 캐시의 제 1 부분의 하나 이상의 캐시 라인들을 상기 상위-레벨 캐시로부터 상기 레벨-1 제 1 캐시의 제 1 부분으로 이동시키도록 구성된다. 또한, 상기 프로세서는 마이크로코드가 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 허용하며 그리고 마이크로코드를 포함하고 있지 않는 명령어들이 상기 레벨-1 제 1 캐시의 제 1 부분에 액세스하는 것을 금지하도록 구성된다.

[0007] 본 발명의 다른 실시예에서, 프로세서는 마이크로코드 액세스 신호를 검출하도록 구성된다. 또한, 프로세서는 상기 마이크로코드 액세스 신호가 어서트(assert) 되지 않는다면, 상기 물리적 메모리 스페이스의 제 1 부분에 명령어들이 액세스하는 것을 금지하며 그리고 상기 마이크로코드 액세스 신호가 어서트(assert) 된다면, 상기 물리적 메모리 스페이스의 제 1 부분에 명령어들이 액세스하는 것을 허용하도록 구성된다.

[0008] 본 발명의 다른 실시예에서, 상기 프로세서는 변환 색인 버퍼(translation lookaside buffer : TLB)를 더 포함하며, 마이크로코드를 포함하고 있지 않는 명령어들이 상기 물리적 메모리 스페이스의 제 1 부분에 액세스하는 것을 금지하는 것은, 상기 물리적 메모리 스페이스의 제 1 부분으로의 TLB 리필들(refills)을 허용하지 않도록 구성된다.

실시예

[0014] 비록, 본 발명은 많은 수정예들과 대안적인 형태들이 가능하지만, 특정한 실시예들이 도면에서 일례로서 도시되었으며 그리고 본 명세서에서 상세히 설명되었다. 하지만, 도면 및 발명의 상세한 설명은 개시된 특정한 형태에 본 발명을 한정하고자 의도된 것이 아니며, 오히려 그 반대로 본 발명은 첨부된 청구항들에 의해서 정의되는 본 발명의 사상 및 범위에 속하는 모든 수정예들, 등가물들 및 대안예들을 커버한다.

[0015] 도1은 컴퓨터 시스템(100)에 대한 일실시예의 일반화된 블록도이다. 예시된 실시예에서, 프로세서(110)는 메모리(150)에 접속되는 것으로 도시된다. 메모리(150)는 SDRAM, SRAM, ROM, DRAM 및/또는 다른 통상적인 메모리 디바이스들을 포함할 수 있다. 프로세서(110)는 코어(120), L2 데이터 캐시(130), L2 변환 색인 버퍼(L2 translation lookaside buffer : L2 TLB)(140)를 포함한다. 코어(120)는 실행 유닛(122), 로드/스토어 유닛(124), L1 데이터 캐시(126), L1 TLB(128)를 포함한다. L2 데이터 캐시(130)는 마이크로코드 애뮬레이션 메모리(135)를 포함한다. 대안적인 실시예에서, 프로세서(110)는 하나 이상의 코어를 포함할 수도 있으며, 각각의 코어는 레벨-1 데이터 캐시를 포함하고 그리고 각각의 코어는 하나의 레벨-2 데이터 캐시를 공유한다. 대안적인 실시예에서, L1 데이터 캐시(126)는 코어(120)로부터 분리될 수도 있다. 다른 대안적인 실시예에서는, 레벨-3 캐시와 같은 추가적인 캐시 레벨들이 컴퓨터 시스템(100)에 포함될 수도 있는바, 이들은 프로세서(110) 내에 포함될 수도 있거나 프로세서(110)로부터 분리될 수도 있다. 이러한 대안적인 실시예들 및 다른 대안적인 실시예들에서, 마이크로코드 애뮬레이션 메모리(135)는 레벨-1 보다 높은 임의의 캐시 레벨에 포함될 수도 있다. 또 다른 다양한 실시예들이 고려된다. 하지만, 쉽게 이해하기 위해서, 후술될 일례들에서는 마이크로코드 애뮬레이션 메모리(135)에 대해서 스페이스가 레벨-2 데이터 캐시 내에 영구 할당된다고 가정할 것이다.

[0016] 동작 동안, 실행 유닛(122)은 실행될 로드들의 데이터 부분(data portion of loads)을 로드/스토어 유닛(124)으로부터 링크(161)를 통해 수신할 수 있으며 그리고 스토어들의 데이터 부분을 링크(162)를 통해 로드/스토어 유닛(124)에게 전달할 수 있다. 로드/스토어 유닛(124)은 실행될 로드들의 데이터 부분을 L1 데이터 캐시(126)로부터 링크(163)를 통해 수신할 수 있으며 그리고 스토어들의 데이터 부분을 링크(164)를 통해 L1 데이터 캐시(126)에게 전달할 수 있다. L1 데이터 캐시(126)는 로드들의 데이터 부분을 L2 데이터 캐시(130)로부터 링크(165)를 통해 수신할 수 있으며 그리고 스토어들의 데이터 부분을 링크(166)를 통해 L2 데이터 캐시(130)에게

전달할 수 있다. L2 데이터 캐시(130)는 로드들의 데이터 부분을 메모리(150)로부터 링크(167)를 통해 수신할 수 있으며 그리고 스토어들의 데이터 부분을 링크(167)를 통해 메모리(150)에게 전달할 수 있다. L1 TLB(128)는, 링크(171)를 통해 L1 데이터 캐시(126)에 접속되고, 링크(172)를 통해 L2 데이터 캐시(130)에 접속되며, 링크(173)를 통해 L2 TLB(140)에 접속되는 것으로 도시된다. 또한, L2 TLB(140)는 링크(174)를 통해 L2 데이터 캐시(130)에 접속되는 것으로 도시된다.

[0017] L1 데이터 캐시(126), L1 TLB(128), L2 데이터 캐시(130), L2 TLB(140)는 통상적인 어드레스 변환 및 캐싱 기능을 수행할 수 있다. 예를 들어, L1 TLB(128)는 가상 어드레스들에서 물리적 어드레스들로의 매핑들을 캐시할 수 있다. 메모리 액세스 요청이 발생되는 때, 원하는 가상 어드레스에서 물리적 어드레스로의 매핑이 캐시되었는지를 알아보기 위해서 L1 TLB(128)가 체크될 수 있다. L1 TLB(128)에 캐시된 매핑들은, 원하는 캐시 라인이 L1 데이터 캐시(126) 내에 존재하는지를 판별하는데에 이용될 수 있다. 원하는 캐시 라인이 L1 데이터 캐시(126) 내에 존재하지 않는다면 즉, L1 캐시 미스(miss)가 있는 경우, 원하는 가상 어드레스에서 물리적 어드레스로의 매핑이 캐시되었는지를 알아보기 위해서 L2 TLB(140)가 체크될 수 있다. L2 TLB(140)에 캐시된 매핑들은, 원하는 캐시 라인이 L2 데이터 캐시(130) 내에 존재하는지를 판별하는데에 이용될 수 있다. L1 데이터 캐시(126)에서 캐시 미스가 발생하는 경우, 새로운 엔트리를 수용하기 위한 공간(room)을 만들어 주기 위해서, L1 데이터 캐시(126)에서 L2 데이터 캐시(130)로 캐시 라인이 퇴거(evict)될 수 있다. L1 TLB(128) 내의 대응 엔트리가 L2 TLB(140)로 옮겨질 수도 있다. L2 데이터 캐시(130)에서 새로운 엔트리를 수용하기 위한 공간(room)을 만들어 주기 위하여, L2 데이터 캐시(130)에서 메모리(150)로 캐시 라인이 퇴거(evict)될 필요가 있을 수도 있다. 원하는 캐시 라인과 L1 TLB(128)에 캐시된 결과에 대해서 새로운 어드레스 변환이 수행될 수도 있는 바, 이 프로세스는 TLB 리필(refill)로 지칭될 수도 있다. 마이크로코드 애플레이션 메모리(135)의 오염(corruption)을 설명하고 이를 예방하는 데이터 캐시들(126, 130)과 TLB들(128, 140)의 동작에 대한 좀더 상세한 내용은 하기에서 설명된다.

[0018] 도2는 프로세서(110)와 함께 이용될 수 있는 가상 메모리 및 캐시 아키텍처에 대한 일례를 도시한다. 도2에는 가상 메모리 어드레스 스페이스(210)가 도시되어 있는바, 그 일부는 물리적 메모리 어드레스 스페이스(220)에 매핑된다. 물리적 메모리 어드레스 스페이스(220)의 일부는 L2 캐시 스페이스(230)에 매핑되는 것으로 도시되며 또한, L2 캐시 스페이스(230)의 일부는 L1 캐시 스페이스(240)에 매핑된다. 프로세서(110) 상에서 실행되는 각각의 애플리케이션은 별도의 가상 메모리 어드레스 스페이스를 채용할 수도 있다. 도2에 도시된 가상 메모리 어드레스 스페이스(210)는 블록 211 - 215를 포함하는바, 이들 블록들은 물리적 메모리 어드레스 스페이스(220)에 매핑되는 가상 메모리의 부분들을 나타내며 그리고 주어진 시점에서 애플리케이션에 의해 액세스될 수 있다. 이와 유사하게, 물리적 메모리 어드레스 스페이스(220)는 블록 221 - 224를 포함하는바, 이들 블록들은 L2 캐시 스페이스(230)에 캐시되는 물리적 메모리의 부분들을 나타낸다. 이와 유사하게, L2 캐시 스페이스(230)는 블록 231 - 233를 포함하는바, 이들 블록들은 L1 캐시 스페이스(240)에 캐시되는 L2 캐시의 부분들을 나타낸다. 좀더 상세하게는, L2 캐시 스페이스(230)의 블록 231, 232, 233은 L1 캐시 스페이스(240)의 블록 242, 243, 241에 각각 매핑된다. 다양한 실시예에서, 전술한 각각의 블록은, 캐시 라인들의 세트 중 하나, 균일 사이즈의 블록들, 캐시 라인들(혹은 블록들)의 그룹, 가변 사이즈의 블록들을 나타낼 수 있다. 대안적인 실시예들에서, 가상 메모리 어드레스 스페이스(210), 물리적 메모리 어드레스 스페이스(220), L2 캐시 스페이스(230), L1 캐시 스페이스(240) 중 임의의 것은, 도2에 도시된 개수보다 더 많거나 혹은 더 적은 개수의 블록들을 포함할 수도 있다.

[0019] 일실시예에서, 물리적 메모리 스페이스(220) 내의 블록 221은 마이크로코드 애플레이션 메모리로서 예약될 수 있다. 또한, L2 캐시 스페이스(230) 내의 블록 231은 마이크로코드 애플레이션 메모리의 내용을 캐싱하기 위해서 영구 예약될 수 있다. 동작 동안, 마이크로프로세서(110)가 마이크로코드 애플레이션 메모리에 액세스하기를 원하는 경우, 블록 231은 레벨 1 캐시 내에 캐시될 수 있는바, 가령 도2에 도시된 바와 같이 블록 242에 캐시될 수 있다. 하지만, 블록 242는 블록 231과 달리, 마이크로코드 애플레이션 메모리의 사용을 위해서 영구 예약되지 않을 수도 있다. L1 캐시에 캐싱된 블록들은 프로그램 실행에 따라 때때로 변할 수도 있다. 따라서, 마이크로코드 애플레이션 메모리는 L1으로부터 L2로 퇴거될 수도 있는데, L2에서는 블록 231이 그 이용을 위해 예약된다. 본 발명의 일실시예에서, 마이크로코드 이외의 애플리케이션들 혹은 프로세스들이 마이크로코드 애플레이션 메모리에 액세스하는 것은, 물리적 메모리 스페이스의 블록 221에 관련된 L1 TLB 리필을 허용하지 않음으로써 금지될 수 있다.

[0020] 도3은 마이크로코드 애플레이션 메모리를 포함하는 메모리 계층에 액세스하기 위한 프로세스(300)에 대한 일례를 예시한 것이다. 메모리 액세스는, 실행 유닛에 의해 디코딩된 각 명령에 관련하여 마이크로코드 액세스 신호

(미도시)의 존재를 체크하는 것으로 시작될 수 있다(결정 블록 310). 예를 들어, 디코딩된 각 명령어의 하나의 비트가 마이크로코드 액세스 신호로서 이용될 수 있다. 대안적인 실시예에서, 마이크로코드 명령어들은 마이크로코드 액세스 신호로서 작용하는 특별한 옵코드(opcode)를 가질 수 있으며 그리고 이에 의해서 마이크로코드 명령어들은 마이크로코드로서 식별될 수 있다. 다양한 다른 마이크로코드 액세스 신호들 중 임의의 것이 실행 유닛으로부터 캐시 제어기로 전달되어, 명령어가 마이크로코드 명령어인지 아닌지를 나타낼 수 있다. 마이크로코드 액세스 신호가 검출된다면, 마이크로코드 에뮬레이션 메모리로의 액세스가 허용될 수 있으며(블록 320) 그리고 액세스가 완료된다.

[0021] 만일, 마이크로코드 액세스 신호가 검출되지 않는다면, 프로세스(300)는 다음과 같이 진행할 수 있다. 상기 액세스에 의해 타겟팅된 캐시 라인에 매칭되는 엔트리를 찾아내기 위해서 하나 이상의 TLB들이 검색된다(블록 330). 만일, 매칭되는 엔트리가 L1 TLB에서 검색된다면(결정 블록 340), 타겟팅된 상기 캐시 라인의 액세스될 수 있으며(블록 390) 그리고 액세스가 완료된다. 만일, 매칭되는 엔트리가 L1 TLB에서 검색되지 않았지만 L2 TLB에서 검색된다면(결정 블록 350), 타겟팅된 상기 캐시 라인은 L2 캐시로부터 L1 캐시로 옮겨질 수 있으며(블록 360), 타겟팅된 상기 캐시 라인의 액세스될 수 있으며(블록 390) 그리고 액세스가 완료된다. 만일, 매칭되는 엔트리가 L1 캐시 또는 L2 캐시 중 그 어느 곳에서도 발견되지 않는다면, 어드레스 변환이 수행될 수 있다(블록 370). 만일, 어드레스 변환의 결과가 마이크로코드 에뮬레이션 메모리 내에 위치한 타겟 어드레스를 생성한다면(결정 블록 380), 액세스가 금지될 수 있으며(블록 384) 액세스 시도가 종료된다. 만일, 어드레스 변환의 결과가 마이크로코드 에뮬레이션 메모리 내에 위치하지 않은 타겟 어드레스를 생성한다면(결정 블록 380), TLB 리필이 수행될 수 있으며(블록 382), 타겟팅된 상기 캐시 라인의 액세스될 수 있으며(블록 390) 그리고 액세스가 완료된다.

[0022] 도4는 레벨-1 캐시에서 마이크로코드 에뮬레이션 메모리를 액세스하기 위한 프로세스(400)의 일실시예를 예시한 도면이다. 마이크로코드 에뮬레이션 메모리를 타겟으로 한 액세스 요청은, 타겟팅된 캐시 라인이 L1 캐시 내에 캐싱되는지를 알아보기 위한 체크와 함께 시작된다(결정 블록 410). 만일 그렇다면, 타겟팅된 캐시 라인에 대한 액세스가 허용될 수 있으며(블록 420) 그리고 액세스가 완료된다. 만일, 타겟팅된 캐시 라인이 L1 캐시에 캐싱되지 않는다면, L2 캐시 내의 타겟팅된 캐시 라인의 예약 위치가 획득될 수 있다(블록 430). 이후, 타겟팅된 캐시 라인은 L2 캐시로부터 L1 캐시로 이동될 수 있다(블록 440). 일단 타겟 캐시 라인이 L1 캐시로 이동되면, 액세스가 허용될 수 있으며(블록 420) 그리고 액세스가 완료된다.

[0023] 이제 도5를 참조하면, 다양한 시스템 구성요소들에 접속된 L2 데이터 캐시(560) 및 마이크로코드 에뮬레이션 메모리(135)를 포함하는 컴퓨터 시스템(500)에 대한 일실시예가 도시되어 있다. 도시된 시스템에서 프로세서(510)는 주변회로(520) 및 메모리(530)에 접속된 것으로 도시된다. 주변회로(520)는, 통상적인 컴퓨터 시스템에서 찾을 수 있는, 네트워크 인터페이스, 타이밍 회로, 저장 매체, 입/출력 디바이스 등등과 같은 다양한 종류의 디바이스들을 포함한다. 메모리(530)는 SDRAM, SRAM, ROM, DRAM 및/또는 다른 통상적인 메모리 디바이스들을 포함할 수 있다. 프로세서(510)는 코어들(540A, 540B), 기입 합체 캐시(write coalescing cache)(550), 레벨-2 데이터 캐시(560), I/O 인터페이스(570)를 포함한다. I/O 인터페이스(570)는 각각의 코어(540)를 주변회로(520)에 접속시킬 수 있다. 참조 번호 뒤에 영문자가 부가된 구성요소들은, 본 명세서에서 참조 번호만으로 총체적으로 지칭될 수도 있다. 예를 들어, 코어들(540A, 540B)은 코어들(540)으로 지칭될 수도 있으며, 코어들(540) 중 특정되지 않은 하나의 코어 역시도 코어(540)라고 지칭될 수 있다.

[0024] 각각의 코어들(540)은 레벨-1 데이터 캐시(542), 스토어 로직 유닛(544), 로드/스토어 파이프라인(546)을 포함한다. 스토어 로직 유닛(544)(또는 '스토어 유닛'이라고 지칭되기도 함)은 로드/스토어 유닛의 일부를 나타낼 수도 있으며, 별도의 로직 유닛을 나타낼 수도 있으며 또는 이들의 조합을 나타낼 수도 있다. 스토어 로직(544)은 레벨-1 데이터 캐시(542)와 기입 합체 캐시(550) 둘다에 접속되므로, 코어(540)가 둘 중 임의의 캐시 레벨에 기입할 수 있다. 좀더 상세하게는, 스토어 로직(544)은, 스토어들(584)을 레벨-1 데이터 캐시(542)에게 전달할 수 있으며 그리고 스토어들(582)을 기입 합체 캐시(550)에 전달할 수 있다. 또한, 기입 합체 캐시(550)는 필(fill)(564)과 퇴거(evicts)(566)를 통해 레벨-2 데이터 캐시(560)에 접속될 수 있다. 기입 합체 캐시(550)는 스토어들(582)과 필(564)을 합체하여, 감소된 개수의 퇴거(566)를 생성할 수 있다. 또한, 레벨-2 데이터 캐시(560)는 각각의 레벨-1 데이터 캐시(542)에 접속될 수 있다. 좀더 상세하게는, 레벨-2 데이터 캐시(560)는 레벨-1 데이터 캐시(542)에게 필(562)을 전달할 수 있다. 또한, 레벨-2 데이터 캐시(560)는 메모리(530)에 양방향성으로(bi-directionally) 접속될 수 있다.

[0025] 동작 동안, 코어(540)는 명령어들의 스트림을 실행할 수 있는바, 상기 명령어들의 스트림은 디코딩되는 때에, L1 데이터 캐시(542)로부터 로드/스토어 파이프라인(546)으로의 로드(586)를 야기하며 및/또는 로드/스토어 파

이프라인(546)으로부터 스토어 로직(544)으로의 스토어(580)를 야기한다. 코어(540)에 의해 실행되는 명령어들은 마이크로코드의 실행을 포함할 수 있다. 마이크로코드의 실행이 마이크로코드 에뮬레이션 메모리(135) 내의 캐시 라인에 대한 액세스를 요구하는 경우, 타겟팅된 캐시 라인이 액세스될 수도 있으며, 그리고 필요하다면, 도4와 관련하여 전술한 프로세스를 이용하여 L2 데이터 캐시(560)로부터 L1 데이터 캐시(542)로 옮겨질 수도 있다. 일단, 타겟팅된 캐시 라인이 L1 데이터 캐시(542)로 옮겨지면, 로드(586) 및/또는 스토어(580, 584)를 통해 액세스될 수 있다.

[0026] 비록, 시스템(500)이 2개의 코어를 포함하는 것으로 도시되어 있지만, 대안적인 실시예에서는 3개 이상의 코어들이 포함될 수도 있으며 및/또는 각각의 코어는 실행 유닛들의 클러스터를 나타낼 수도 있다. 3개 이상의 코어들이 포함되는 다른 대안적인 실시예에서는 추가 레벨-2 캐시들이 또한 포함될 수도 있다. 더 나아가, 레벨-2 데이터 캐시(560)가 메모리(530)에 직접 접속되는 것으로 도시되어 있고 그리고 메모리(530)는 오프-프로세서 메모리(off-processor memory)인 것으로 도시되어 있지만, 프로세서(510)는 메모리 제어기 및/또는 온-프로세서 메모리(on-processor memory)를 포함할 수도 있다. 대안적으로는, 오프-프로세서 메모리 제어기는 레벨-2 데이터 캐시(560)를 메모리(530)에 접속시킬 수도 있다. 다양한 프로세서 코어들 및 메모리 구성들이 해당 기술분야의 당업자에게는 자명할 것이다.

[0027] 전술한 실시예들은 소프트웨어로 구성될 수도 있다는 점을 유의해야 한다. 이러한 실시예에서, 방법들 및/또는 매커니즘을 구현하는 프로그램 명령어는 컴퓨터 액세스가능 매체 상에 전달 및 저장될 수 있다. 프로그램 명령어를 저장하도록 구성된 다양한 유형의 매체가 이용가능하며 그리고 이는 하드 디스크, 플로피 디스크, CD-ROM, DVD, 플래시 메모리, 프로그램가능한 ROM(PROM), 임의접근메모리(RAM), 및 다른 다양한 유형의 휘발성 또는 비휘발성 저장소자를 포함한다. 컴퓨팅 디바이스에 의한 액세스를 위해 프로그램 명령어를 전달하도록 구성된 또 다른 형태의 매체는 전기 신호, 전자기 신호, 광 신호 또는 디지털 신호가 전달될 수 있는 네트워크, 무선, 및 위성 링크와 같은 지상(terrestrial) 및 비-지상(non-terrestrial) 통신 링크를 포함한다. 따라서, 다양한 실시예들은, 전술한 본 발명의 설명에 따라 구현되는 명령어들 및/또는 데이터를 컴퓨터 액세스가능 매체 상에서 수신, 송신 또는 저장하는 것을 또한 포함할 수 있다.

[0028] 비록, 전술한 실시예들은 상당히 자세한 정도까지 설명되었지만, 본 명세서에 개시된 바를 충분히 이해한다면, 다양한 변형예들 및 수정예들이 해당 기술분야의 당업자들에게 자명할 것이다. 다음의 청구항들은 이러한 모든 변형예들 및 수정예들을 포괄하는 것으로 해석되어야 한다.

산업상 이용 가능성

[0029] 일반적으로, 본 발명은 마이크로프로세서 분야에 적용가능하다.

도면의 간단한 설명

[0009] 도1은 컴퓨터 시스템의 일실시예에 대한 일반화된 블록도이다.

[0010] 도2는 가상 메모리와 캐시 아키텍처의 일실시예를 예시한 것이다.

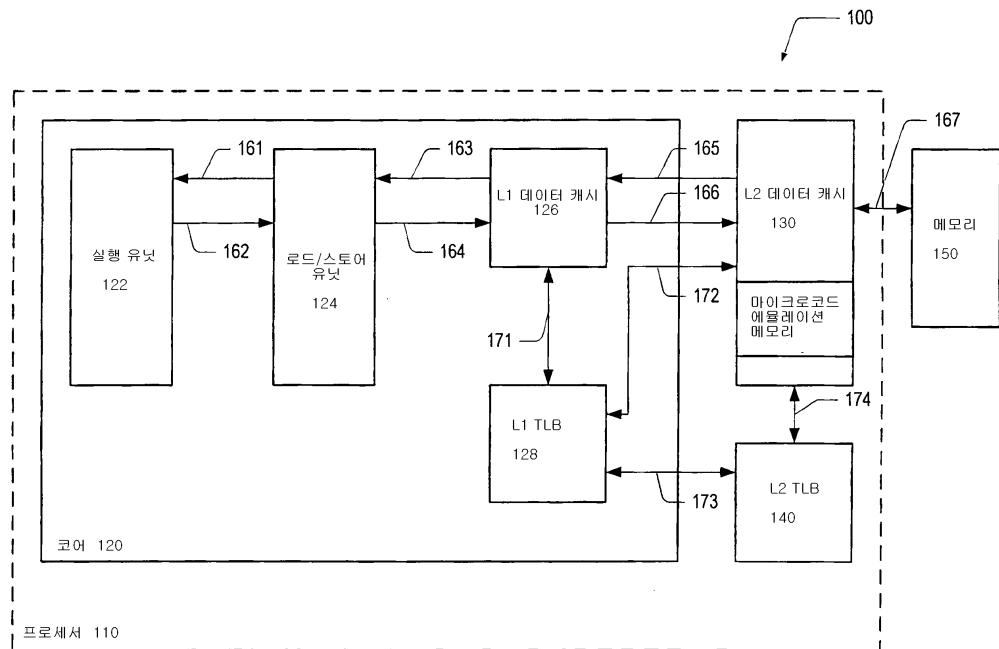
[0011] 도3은 마이크로코드 에뮬레이션 메모리를 포함하는 메모리 계층에 액세스하기 위한 프로세스의 일실시예를 예시한 것이다.

[0012] 도4는 레벨-1 캐시 내의 마이크로코드 에뮬레이션 메모리를 액세스하기 위한 프로세스의 일례를 예시한 것이다.

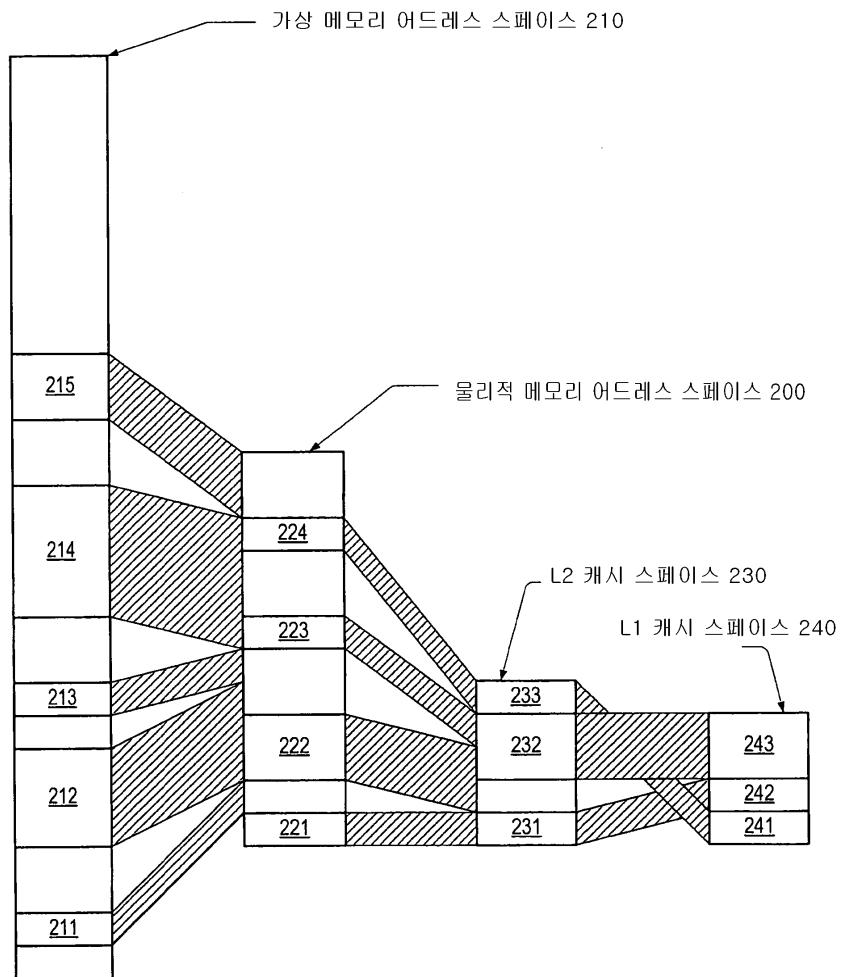
[0013] 도5는 다양한 시스템 구성요소에 결합된 L2 데이터 캐시 및 마이크로코드 에뮬레이션 메모리를 포함하는 컴퓨터 시스템의 일실시예에 대한 블록도이다.

도면

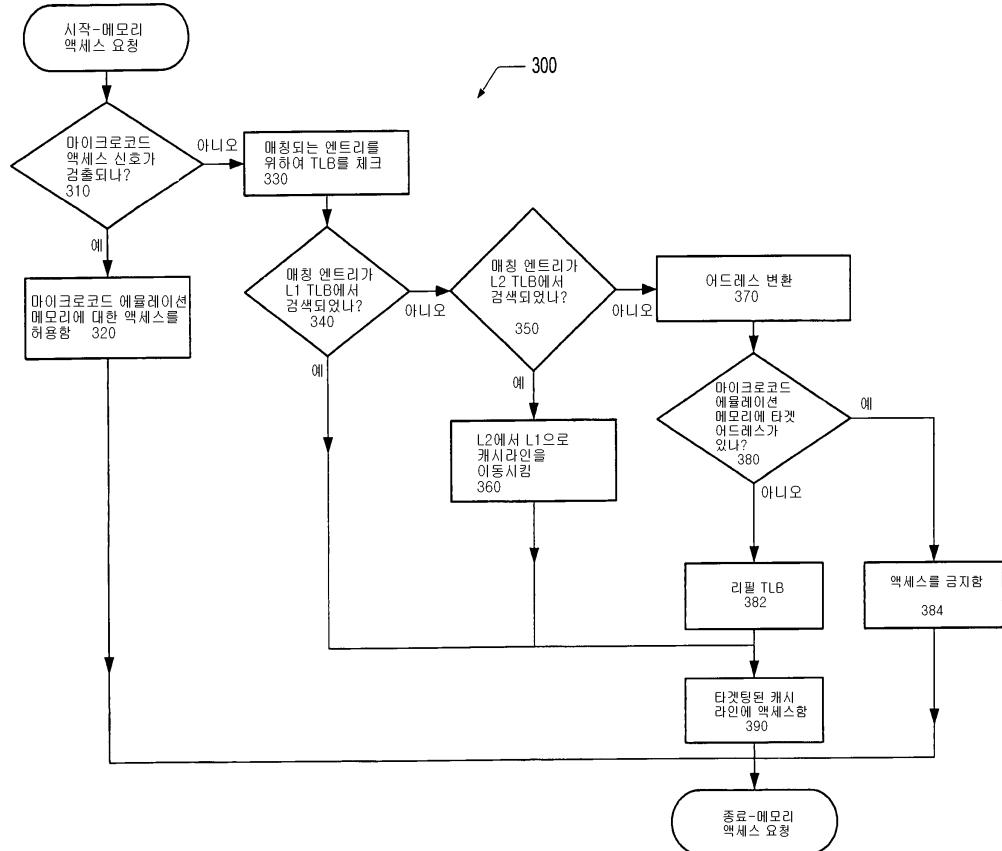
도면1



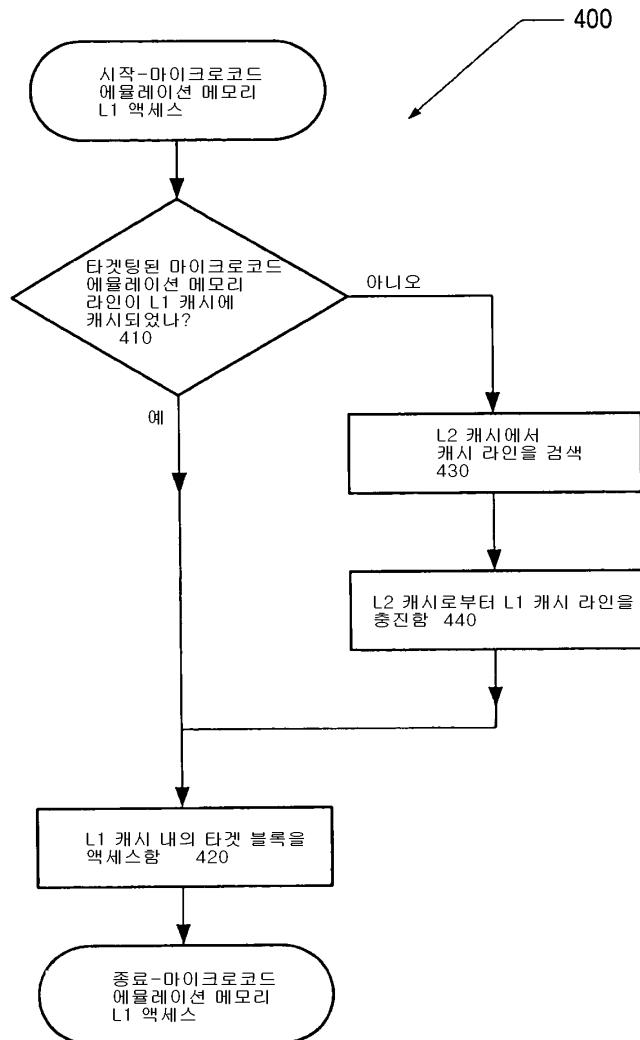
도면2



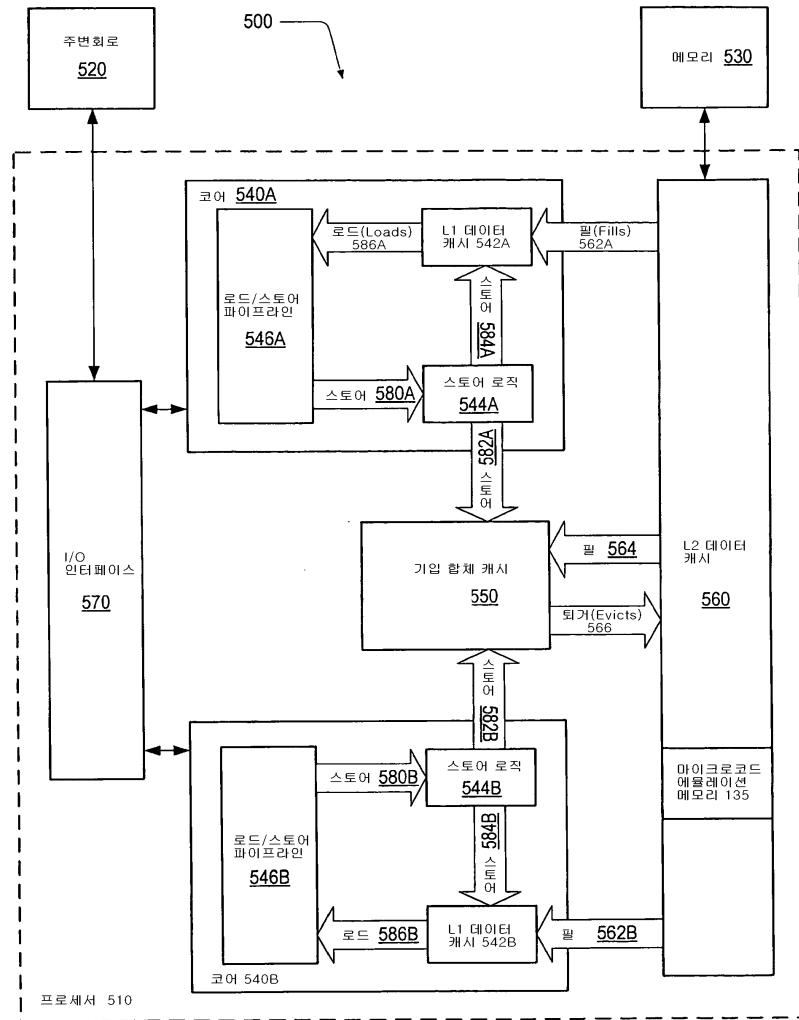
도면3



도면4



도면5



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 16

【변경전】

상기 상위-레벨 캐시로부터 상기 레벨-1

【변경후】

상기 상위-레벨 캐시로부터 레벨-1

【직권보정 2】

【보정항목】 청구범위

【보정세부항목】 청구항 12

【변경전】

상기 프로세서의 상기 레벨-1

【변경후】

상기 프로세서의 레벨-1