

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.⁷
G06F 12/08

(11) 공개번호 특2000-0076502
(43) 공개일자 2000년 12월 26일

(21) 출원번호	10-2000-0002818
(22) 출원일자	2000년 01월 21일
(30) 우선권주장	9/240,348 1999년 01월 29일 미국(US)
(71) 출원인	인터내셔널 비지네스 머신즈 코포레이션 포만 제프리 엘 미국 10504 뉴욕주 아몬크
(72) 발명자	차니마크제이. 미국 10028 뉴욕주 뉴욕에이피티. 넘버 30 씨서드 애비뉴 1438 엠마필립지. 미국 06811 코넥티컷주 덴버리 팩스 덴로드 28 프레너 다니엘 에이. 미국 10520 뉴욕주 크로튼-온-허드슨 헤이스팅즈 에이브이. 28 푸작도마스알. 미국 06877 코넥티컷주 리지필드 스쿨 하우스 플레이스 9
(74) 대리인	주성민, 장수길

심사청구 : 있음

(54) 세트 예측을 사용하여 세트 연상 캐시에서 대기 시간을 감소시키기 위한 방법 및 장치

요약

세트 연상 메모리의 성능을 강화하고 전력 소모를 감소시키기 위해 세트 연상 메모리의 어느 세트를 액세스할 것인지에 관한 세트 예측 정보를 저장하고 사용하는 방법 및 장치가 개시된다. 세트 예측 정보는 세트 연상 명령 캐시 및 데이터 캐시에 액세스하는 대기 시간을 감소시키기 위해 브랜치 목표 버퍼, 명령 캐시, 피연산자 히스토리 표를 포함하는 다양한 위치에 저장된다.

대표도

도 2

색인어

메모리 시스템, 세트 연상 메모리, 명령 캐시, 데이터 캐시, 세트 선택 정보, 액세스 대기 시간.

명세서

도면의 간단한 설명

- 도 1은 종래 기술의 가상 태그 캐시의 블록도.
- 도 2는 물리적 태그 세트 연상 캐시에 대한 본 발명의 적용을 표시하는 블록도.
- 도 2A는 본 발명에 따른 방법의 적용을 도시하는 예시적인 흐름도.
- 도 3은 본 발명의 제2 실시예의 블록도.
- 도 4는 본 발명의 제3 실시예의 블록도.
- 도 5A, 5B, 5C는 본 발명의 세트 선택 정보에 대한 저장 장치로서 브랜치 히스토리 표, 명령 캐시, 피연산자 히스토리 표를 사용하는 것을 도시하는 도면.
- 도 6은 종래 기술의 피연산자 히스토리 표의 동작을 도시하는 블록도.
- 도 7은 중앙 처리 장치에서 본 발명의 적용을 도시하는 블록도.
- 도 8A 및 8B는 각각 종래 기술과 본 발명의 방법을 사용한 캐시 액세스를 도시하는 타이밍도.

<도면의 주요부분에 대한 부호의 설명>

10: 캐시 디렉토리

- 12: 캐시 어레이
- 14: 디렉토리 다중화기
- 16: 어레이 다중화기
- 18: 주소
- 26: 번역 룩 어사이드(TLB)
- 28: 세트 선택

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 컴퓨터 메모리 시스템에 관한 것으로, 구체적으로는 세트 연상 캐시(set-associative cache)에서 액세스 대기 시간(access latency)을 감소시키기 위한 방법 및 장치에 관한 것이다.

대개 캐시 메모리는 중앙 처리 장치(central processing unit: CPU)와 주메모리 사이에 위치한 작고 고속인 버퍼이다. 캐시는 현재 사용 중이라고 생각되는 주메모리의 내용을 임시적으로 유지하기 위해서 사용된다. 캐시의 내용을 교체하는 시간에 관한 판단은 일반적으로 가장 오래전 사용(least recently used: LRU) 알고리즘에 기초한다. LRU 알고리즘은 캐시 메모리 위치를 가장 최근에 사용된 주메모리의 내용으로 교체한다. 캐시 메모리의 정보는 주메모리의 정보보다 훨씬 짧은 시간에 액세스될 수 있다. 그러므로, CPU는 캐시에 페치되고/거나 저장될 명령 및/또는 피연산자를 대기하는데 시간이 덜 든다.

직접 대응 캐시는 주메모리의 어떤 특정 위치의 내용을 캐시의 특정 위치에 한정하여 저장한다. 반면, M방식 세트 연상 캐시는 각 주메모리 위치의 내용을 캐시의 M가지 위치 중 하나에 대응시킨다. 기본적으로 M방식 세트 연상 캐시는 M개의 동일한 직접 대응 캐시의 조합이다. 그러나, M방식 세트 연상 캐시로부터의 액세스 및 검색은 보다 복잡하다. M방식 세트 연상 캐시로의 매 메모리 액세스 때마다 M개의 동일한 직접 대응 캐시의 조합이 탐색되어야 하고 일치가 되면 적절한 데이터가 출력으로 선택되고 다중화되어야 한다. 미스가 발생하면, 어느 캐시 라인이 삭제되어 더 최근에 사용된 주메모리의 내용으로 재가입될지에 관해 M 개의 가능한 캐시 라인 중에서 선택되어야 한다.

도 1은 캐시 디렉토리(10), 캐시 어레이(12), 디렉토리 다중화기(14), 어레이 다중화기(16)를 포함하는 종래 기술의 가상 태그(virtually-tagged) 4방식 세트 연상(4-way set-associative) 캐시 메모리를 도시한다. 캐시 디렉토리(10)는 캐시 어레이(12)의 각 해당 위치에 대한 가상 주소를 포함한다. 캐시 어레이(12)는 캐시 디렉토리(10)의 해당 위치 또는 블록에 의해 지적된 주메모리 위치의 내용을 저장한다. 세트(set)는 캐시 어레이(12)의 열 및 캐시 디렉토리(10)의 해당 열로서 정의된다. 일치 등급(congruence class)은 캐시 어레이(12)의 행 및 캐시 디렉토리(10)의 해당 행으로서 정의된다. 블록 또는 위치는 특정 세트(열) 및 특정 일치 등급(행)의 교차로서 정의된다. 위치 또는 블록은 하나 이상 바이트의 데이터를 포함한다.

캐시 메모리에 제공된 주소(18)는 디렉토리 태그(20), 일치 등급(22), 블록 오프셋(24)을 포함한다. 디렉토리 태그(20)는 캐시 디렉토리(10)에서 원하는 세트(열)를 디렉토리 다중화기(14)를 통해 선택하기 위해 사용된다. 일치 등급 태그(22)는 캐시 디렉토리(10) 및 캐시 어레이(12) 모두의 원하는 일치 등급(행)을 선택하기 위해 사용된다. 블록 오프셋(24)은 원하는 블록 또는 위치 내에서 원하는 바이트를 선택하기 위해 사용된다. 디렉토리 다중화기(14)의 출력은 캐시 어레이(12)의 원하는 세트(열)를 어레이 다중화기(16)를 통해 선택하기 위해 사용된다.

결합적 캐시를 액세스할 때 대기 시간은 주소를 캐시 디렉토리(10)의 다중 세트에 걸쳐 저장된 태그와 비교해야 하기 때문에 직접 대응 캐시를 액세스할 때의 대기 시간보다 길다. 일치가 있으면, 일치하는 태그와 연관된 세트는 캐시 어레이(12)의 해당 세트로부터 출력을 선택하기 위해 사용된다. 캐시 어레이(12)의 출력은 최종적으로 레지스터 및 기능 유닛으로 라우트된다. 소위 "늦은 선택(late select) 문제"는 캐시 어레이(12)의 적절한 세트가 선택될 수 있기 전에 캐시 디렉토리(10)를 순람하기 위한 주소에 대한 필요 및 (물리적 태그 캐시가 사용되면) 잠재적으로 주소 번역에 대한 필요를 의미한다. 그러므로, 늦은 선택 문제는 세트 연상 캐시에서 대기 시간에 대해 악영향을 미친다.

그러므로, 세트 선택 정보가 캐시 디렉토리를 탐색하고 주소를 번역하기 전에 사용가능하다면 유리할 것이다.

캐시에 관한 더 상세한 사항은 본 명세서에 참조적으로 결합된 다음 참조 문헌에서 찾을 수 있다.

1. Emma 등에 허여된 미국특허제5,634,119호.
2. Chang, Sheldon S.L. Electrical and Computer Engineering III (1983).
3. Smith, Allan J. Cache Memories - ACM Computing Surveys vol. 14 (1982).
4. Cekleov M. 및 Dubois M. Virtual-address Caches - IEEE Micro (1997).

발명이 이루고자하는 기술적 과제

본 발명의 도시적인 실시예에 따르면, 세트 연상 캐시에서 액세스 대기 시간을 감소시키기 위한 방법에

있어서, 데이터는 최소한 한 선택 캐시를 통해 선택가능한 메모리의 위치로부터 독취되고, 상기 방법은 세트 선택 정보를 발생시키는 단계, 및 선택 캐시로부터의 메모리 선택 정보의 도착 이전에 세트 선택 정보가 메모리로부터의 데이터 검색에 대해 사용가능하게 되는 위치에 세트 선택 정보를 저장하는 단계를 포함하는 방법이 제공된다.

세트 연상 캐시에서 액세스 대기 시간을 감소시키기 위한 장치는 세트 선택 정보를 저장하기 위한 저장 수단, 주소를 수신하여 주소에 의해 결정된 M 세트의 데이터를 출력시키기 위한 M방식 세트 연상 캐시, 세트 선택 정보 및 세트 연상 주소 중의 하나를 다중화하기 위한 다중화기를 포함하되, 상기 세트 선택 정보는 데이터를 액세스하기 위한 세트 연상 주소 이전에 사용가능하다.

세트 연상 캐시의 전력 소모를 감소시키기 위한 장치는 세트 선택 정보를 저장하기 위한 세트 선택 저장 수단, 어레이 및 디렉토리-상기 디렉토리는 어레이의 주소의 세트 연상 태그 부분을 출력함을 포함한 M 방식 세트 연상 캐시, 상기 디렉토리로부터 주소의 태그 부분 및 M 세트의 데이터 중의 한 세트를 출력하기 위한 세트 선택 정보 중의 하나를 다중화하기 위한 다중화기를 포함한다.

또한, 본 발명에 따르면, 데이터 주소를 사용하는 세트 연상 메모리의 액세스 속도를 증가시키는 방법이 제공된다. 주소는 오프셋 부분, 일치 등급 인덱스, 태그 부분을 포함한다. 세트 연상 메모리는 어레이 및 디렉토리를 포함한다. 어레이는 데이터를 저장하고 복수의 어레이 일치 등급으로 분할된다. 어레이 일치 등급은 어레이 세트로 분할된다. 어레이 세트는 캐시 라인을 포함한다. 캐시 라인은 복수의 데이터를 포함한다. 디렉토리는 복수의 디렉토리 일치 등급으로 분할된다. 디렉토리 일치 등급은 각각이 디렉토리 엔트리를 포함하는 디렉토리 세트로 분할된다. 디렉토리 엔트리는 주소 태그 및 유효 비트, 패리티 등을 포함하는 다른 상대 정보를 포함한다. 디렉토리는 분할되어 주소 태그가 캐시 라인 중의 하나와 연관되도록 디렉토리 엔트리와 캐시 라인 사이에는 일대일 대응이 성립된다.

양호하게는, 단일 어레이 일치 등급의 세트-단일 어레이 일치 등급은 일치 등급 인덱스에 의해 규정됨-의 내용을 액세스하는 단계, 일치 등급 인덱스를 사용하여 단일 디렉토리 일치 등급-단일 디렉토리 일치 등급은 일치 등급 인덱스에 의해 규정됨-의 세트의 내용을 액세스하는 단계, 세트 선택 정보를 발생시키는 단계, 어레이 일치 등급의 세트를 선택하기 위해 세트 선택 정보를 사용하는 단계, 선택된 세트에서 캐시 라인으로부터 데이터를 출력하는 단계, 디렉토리 일치 등급의 선택된 세트의 주소의 태그에 태그 부분을 비교하는 단계, 선택된 일치 등급에서 주소 태그의 하나가 주소의 태그 부분과 동일하면 선택된 세트를 세트 선택 정보와 비교하는 단계, 선택된 일치 등급의 주소 태그중 어떤 것도 주소의 태그 부분과 동일하지 않으면 액세스가 성공적이지 아니었음과 캐시 라인으로부터 출력된 데이터가 비유효함을 표시하는 제1 제어 신호를 출력하는 단계, 선택된 세트가 세트 선택 정보와 동일하지 않으면 캐시 라인으로부터 데이터가 비유효함을 표시하는 제2 제어 신호를 출력하는 단계를 포함한다.

또한 본 발명에 따르면, 데이터 주소를 사용하여 세트 연상 메모리에서 액세스 시간을 감소시키는 장치가 제공된다. 주소는 오프셋 부분, 일치 등급 인덱스, 태그 부분을 포함한다. 세트 연상 메모리는 어레이 및 디렉토리를 포함하고, 어레이는 데이터를 포함한다. 어레이는 복수이 어레이 일치 등급으로 분할된다. 어레이 일치 등급은 어레이 세트로 분할되고 어레이 세트는 세트 연상 메모리의 세트 결합성을 결정한다. 어레이 세트는 캐시 라인을 포함하고, 캐시 라인은 복수의 데이터를 포함한다. 디렉토리는 복수의 디렉토리 일치 등급으로 분할된다. 디렉토리 일치 등급은 디렉토리 세트로 분할된다. 디렉토리 세트는 디렉토리 엔트리를 포함하고, 디렉토리 엔트리는 주소 태그를 포함한다. 디렉토리는 디렉토리 엔트리와 캐시 라인 사이에 디렉토리의 주소 태그가 캐시 라인의 최소한 하나와 연관되도록 일대일 대응이 성립하도록 분할된다. 세트 연상 메모리에서 액세스 시간을 감소시키기 위한 장치는 어레이 일치 등급-단일 어레이 일치 등급은 일치 등급 인덱스에 의해 규정된 일치 등급-의 세트의 내용을 액세스하기 위한 수단, 및 단일 디렉토리 일치 등급-단일 디렉토리 일치 등급은 일치 등급 인덱스에 의해 규정됨-의 세트의 내용을 액세스하기 위한 수단, 및 세트 선택 정보를 발생시키기 위한 수단, 및 세트 선택 정보를 사용하여 단일 어레이 일치 등급의 세트 중의 하나를 선택하는 수단 및, 선택된 세트에서 캐시 라인으로부터 데이터를 출력시키기 위한 수단 및, 주소의 태그 부분을 선택된 단일 디렉토리 일치 등급의 세트로부터 주소 태그와 비교하는 단계, 및 세트 선택 정보와 주소 태그를 포함하는 세트를 비교하는 수단, 및 데이터 액세스의 성공 및 캐시 라인으로부터의 데이터 출력의 유효성을 표시하는 제어 신호를 출력하는 수단을 포함한다.

발명의 구성 및 작용

세트 예측(set prediction)을 사용하여 세트 연상 캐시(set-associative cache) 액세스의 늦은 선택(late select) 단계를 대기할 필요가 제거된다. 본 발명은 명령 및 데이터 캐시의 성능을 개선하기 위해 세트 예측을 위한 세트 선택 정보를 사용하는 실시예를 개시한다.

도 2는 본 발명의 실시예의 블록도이다. 도시된 대로, 세트 선택(set selection)(28)은 물리적 주소화(physically-addressed) 세트 연상 캐시의 디렉토리 다중화기(directory MUX)(14) 및 어레이 다중화기(array MUX)(16)에 직접 인가된다. 도 1에 도시된 가상 주소화(virtually addressed) 캐시와 대조적으로 도 2에 도시된 물리적 주소화 캐시는 가상 주소(18)를 캐시 디렉토리(10)에서 사용되는 물리적 주소로 번역하기 위한 번역 룩 어사이드(translation look aside: TLB) 버퍼(26)를 포함한다. 세트 선택(28)을 어레이 다중화기(16)에 직접 제공하여, 디렉토리 다중화기(14)로부터 어레이 다중화기(16)로의 다중화기 선택 데이터의 사용가능 이전에 캐시 어레이부터 데이터가 검색될 수 있다.

본 발명의 도시적인 실시예에 따라, 세트 선택(28)에 대한 데이터는 브랜치 목표 버퍼(branch target buffer: BTB)(또한 브랜치 히스토리 표(branch history table: BHT)라 지칭됨), 또는 피연산자 히스토리 표(operand history table: OHT), 또는 명령 캐시를 포함하는 복수의 소스로부터 저장되거나 유도될 수 있다.

BTB는 브랜치 명령(branch instruction)에 대한 목표 주소를 예측하기 위해 사용되는 버퍼이다. BTB에

대한 이론은 대부분의 브랜치가 취해지거나 취해지지 않고 취해진다면 동일한 목표 주소를 가질 것이라는 것이다. 그러므로, 취해진 브랜치의 표는 표의 각 엔트리(entry)가 브랜치의 목표 주소가 뒤따르는 취해진 브랜치의 주소를 포함하도록 설정된다. 표의 엔트리는 브랜치가 만들어질 때 설정된다. 표가 다 채워지면, 새로운 엔트리를 부가는 대개 가장 오래전 사용(least recently used: LRU) 알고리즘을 통해 오래된 엔트리를 교체를 포함한다.

도 5A를 참조하면, 세트 선택 정보를 저장하기 위한 위치로서 BHT(30)를 사용하는 것을 도시한다. 명령 실행 도중에, 각 브랜치는 브랜치 주소를 사용하는 BHT(30)에서 참조되고, 찾아지면, 그 목표 주소는 페치되고(fetch) 실행 스트림에서 다음 명령이 된다. 브랜치 주소가 BHT에서 찾아지지 않으면, 취해지지 않은 것으로 간주된다. 브랜치 명령의 실행이 계속되는 동안 표는 따라서 갱신된다. 취해지지 않은 예측된 브랜치가 취해지지 않으면, 해당 BHT 엔트리는 삭제된다. 취해지지 않은 것을 예측된 브랜치가 취해지면, 새로운 엔트리가 설정된다. 예측된 타겟 주소가 틀린 것이면, 정정된 주소가 입력된다.

BHT(30)의 각 엔트리는 브랜치의 주소, 해당 브랜치 목표 주소, 세트 선택 정보를 유지한다. 각 브랜치는 프로그램 계수기로부터 유도된 명령 주소를 가지고 액세스된다. 세트 선택 정보는 어느 세트가 특정 BHT 엔트리에 의해 지적된 명령을 가장 최근에 포함했는지를 표시하고 그 BHT 엔트리에 해당하는 브랜치를 고려하여 명령 캐시로부터 페치될 적절한 세트를 예측하는데 사용된다. 페치될 브랜치 주소는 프로그램 계수기로부터 유도되고 BHT(30)의 브랜치 주소와 비교된다. 일치(match)가 찾아지면, 해당 브랜치 목표 주소 및 세트 선택 정보는 명령 캐시에 제공된다. 그리고, BHT에 저장된 세트 선택 정보로부터 예측된 세트를 사용하여, TLB 및 캐시 디렉토리의 출력이 적절한 세트를 선택하기 위해 사용될 때보다 빨리 적절한 명령이 명령 캐시로부터 페치될 수 있다.

BHT로부터 세트 선택을 구현하는 본 발명의 일 실시예가 도 2의 블록도 및 도 2A의 흐름도를 참조하여 설명될 것이다.

프로그램 계수기로부터 명령 페치 주소(대개 브랜치 명령을 포함하는 것으로 예측된 브랜치 명령 또는 명령 페치 그룹에 해당하는 프로그램 위치)를 브랜치 히스토리 표(BHT)로 페치한다. BHT는 명령 캐시에서 브랜치 목표의 예측 세트를 식별하는 브랜치 목표 주소 및 예측 세트 인덱스(predicted set index)를 제공한다(단계 200). 예측 세트 인덱스는 도 2에서 세트 선택(28) 역할을 한다.

주소(8)에서 명령 캐시로 브랜치 목표 주소를 제공한다. 목표 주소는 3 부분으로 분할된다. 부분 1은 디렉토리 태그(directory tag), 부분 2는 일치 등급 인덱스(congruence class index), 부분 3은 명령의 오프셋(offset)이다(단계 205).

병렬적으로, 일치 등급 인덱스를 캐시 디렉토리(10) 및 캐시 어레이(12)에 제공한다. 일치 등급에 대해 디렉토리 및 어레이 엔트리를 독취한다. 디렉토리 태그 부분을 디렉토리 다중화기(14)에 TLB를 통해 제공하고 세트 선택(28)을 디렉토리 다중화기(14) 및 어레이 다중화기(16)에 제공한다. 세트 선택은 (단계 3에서 독취한 일치 등급에 대한 어레이 엔트리의 세트로부터) 적절한 어레이 엔트리를 출력하기 위해 어레이 다중화기(16)를 선택한다(단계 210).

캐시 어레이(12)로부터 워드 출력을 식별하기 위해 오프셋을 사용하고 처리하기 위해 명령 큐(queue)(또는 버퍼)에 워드를 부가한다(단계 220).

디렉토리 다중화기(14)는 세트 선택(28)로부터의 디렉토리 태그가 캐시 디렉토리(10)로부터 단계 210에서 독취된 일치 등급에 대한 태그 엔트리 중의 하나와 일치하는 지를 식별한다(단계 225).

비일치(mismatch)(캐시 미스(cache miss))가 발생하면, 단계 220에서 명령 큐에 부가된 워드를 비유효화한다(단계 245). 예를 들면, 다음 수준 메모리 계층으로부터 명령/데이터 페치와 같은 종래의 방법으로 미스를 처리한다(단계 250). 명령 캐시 라인을 캐시에 놓는다(단계 255). BHT를 명령 캐시 라인의 새로운 세트로 갱신한다. 본 발명이 속하는 기술 분야의 숙련자에게 예측의 갱신 소스에 대한 기술은 공지되어 있다. 양호하게는 이 예에서 이러한 갱신을 수행하기 위해서, 예측을 위해 사용된 원래 BHT 엔트리의 결합을 기록한다(단계 260).

단계 225에서 세트 선택(28)으로부터의 디렉토리 태그가 TLB(26)에서 제공된 태그와 일치하면(캐시 히트(cache hit)), 예측이 맞는지를 판단해야 한다.

일치하는 디렉토리 태그의 세트에 의해 식별된 캐시 어레이(12)에서 명령의 결정 세트를 식별하는 결정 세트 인덱스(resolved set index)를 발생시킨다(단계 230). 병렬적으로, 결정 세트 인덱스를 어레이 다중화기(16) 및 세트 예측 결정 로직 회로(도시되지 않음)에 발송하고, (단계 210에서 독취된 일치 등급에 대한 어레이 엔트리의 세트로부터) 적절한 어레이 엔트리를 선택하기 위해서 어레이 다중화기(16)를 제어하기 위해 결정 세트 인덱스를 사용한다(단계 235).

결정 세트 인덱스가 세트 선택(28)로부터의 예측 세트 인덱스와 일치하는지를 예측 결정 로직 회로는 판단한다.

일치가 되면(예측이 맞으면), 결정 세트 인덱스에 해당하는 엔트리를 선택하여 어레이 다중화기(16)의 동작을 디스에이블시키고 명령 큐에 부가된 워드를 유효화한다(단계 240).

불일치가 되면(예측이 틀리면), 명령 큐에 부가된 워드를 비유효화시킨다(단계 270). 단계 240에서 판단된 캐시 어레이의 워드를 식별하기 위해서 오프셋을 사용하고 명령 큐에 워드를 부가시킨다(단계 275). 새로운 예측 세트가 결정 예측 세트와 일치함을 표시하기 위해 세트 선택 정보(28)의 엔트리를 갱신한다(단계 280).

본 발명의 상기 실시예에서 BHT가 BHT의 매 버퍼 또는 위치마다 부착된 세트 예측 필드를 포함한다는 것을 이 기술분야의 숙련가는 용이하게 알 수 있을 것이다. 브랜치 목표(주소)와 결합된 세트 선택 또는 세트 예측을 사용하여, 데이터는 번역 또는 낮은 선택 신호의 디렉토리 록업 발생(TLB(26), 캐시 디렉토

리(10), 디렉토리 다중화기(14)를 통해)을 대기하지 않고서 명령 캐시로부터 검색될 수 있다.

선택적으로, 부가적인 세트 선택 정보가 각 BHT 엔트리에 대해 할당될 수 있다. 부가적인 세트 선택 정보는 실패 또는 취해지지 않은 경로에 대한 적절한 세트를 규정할 수도 있다. 이러한 정보는 틀리게 예측된 브랜치로부터 복구할 때 또는 브랜치 명령이 캐시 블록의 최종 위치를 차지하고 실패 경로가 다른 캐시 블록에 저장될 때 유용하다. 목표 및 실패 경로가 다른 캐시 세트에 있음을 검출한 후, 두 세트는 동시에 명령 캐시로부터 폐치될 수 있다. 또한, 다중 브랜치가 단일 명령 캐시 라인 내에서 발생할 수 있으므로, BHT의 부가적인 세트 선택 정보는 각 브랜치에 대해 필드를 제공할 수도 있다.

대안적으로, 세트 선택 정보는 데이터 캐시의 성능을 개선하기 위해서 명령 캐시 내에 저장될 수 있다. 도 5B를 참조하면, 데이터 캐시 세트 선택 정보(100)는 명령에 의해 요구되는 피연산자의 예측 세트를 표시하는 명령 캐시 라인(120)에 첨부될 수 있다. 도 5B에 도시된 대로, D-캐시 세트 선택은 I-캐시 라인(120)의 해당되는 고정 부분(0 내지 8)과 연관되어 있다. 명령을 로드하거나 저장할 수 있는 명령은 로드되거나 저장될 피연산자를 포함하는 세트를 액세스하기 위해 데이터 캐시 세트 선택 정보를 사용한다. 데이터 캐시 세트 선택 정보는 피연산자 주소 발생 이전인 명령 캐시의 액세스 도중에 사용가능하므로, 데이터 캐시의 성능은 이 방법으로 상당히 개선될 수 있다.

도 6은 종래 기술의 피연산자 히스토리 표(OHT)의 동작을 도시한다. 프로그램 계수기(32)에 의해 폐치된 각 명령 주소는 OHT(36)의 명령 주소와 대응된다. 일치가 찾아지면, 해당 피연산자 주소는 폐치된다. 피연산자는 그리고나서 전디코드(pre-decode) 피연산자 버퍼의 명령과 대응된다. 명령이 디코디될 때, 이 명령과 연관된 피연산자는 사용가능하고, 즉시 실행될 수 있다.

본 발명의 또다른 도시적인 실시예에 따르면, 세트 선택 정보는 또한 데이터 캐시의 성능을 개선하기 위해서 OHT에 저장될 수 있다. OHT는 명령과 연관된 피연산자를 폐치하는데 걸리는 시간을 감소시킨다. OHT는 피연산자를 요구하는 다수의 명령이 명령을 실행할 때마다 동일한 피연산자 주소를 계산한다는 이론에 기초하고 있다. 예를 들면, 피연산자 주소는 기본 레지스터, 인덱스 레지스터, 명령에 의해 제공된 변위(displacement) 필드를 가산하여 계산될 수 있다. 각 엔트리가 명령 주소 및 그 명령에 대한 피연산자의 주소를 포함하는 표를 사용하여, 피연산자는 명령이 디코드되기 전에 폐치될 수 있다.

도 5C를 참조하면, 데이터 캐시 세트 선택 정보(140)는 OHT(36)의 각 엔트리에 첨부되고, 적절한 피연산자의 주소가 사용가능하며, 명령이 디코드되기 전에 피연산자는 데이터 캐시로부터 폐치될 수 있어서, 데이터 캐시의 액세스 대기 시간을 상당히 감소시킨다. 프로그램 계수기만이 OHT를 액세스하는데 요구되므로, 시스템이, 예를 들면, 피연산자에 대해 증가된 액세스 시간으로 인해 그렇게 요구한다면 피연산자는 명령을 폐치하기 전에도 폐치될 수 있다.

도 3은 종래의 캐시 디렉토리(10) 및 감소된 전력 소모에 대해 변형된 캐시 어레이(12)를 포함하는 본 발명의 실시예를 도시한다. 제1 실시예와 제2 실시예의 차이점 하나는 세트 선택 정보(28)가 캐시 어레이(12)의 액세스에 있어서 어느 세트가 출력될 지를 판단하는데 사용되기 보다는 캐시 어레이(12)를 액세스하기 전에 세트 선택 정보(28)가 사용되도록 어레이 다중화기(16)가 캐시 어레이(12)의 앞으로 이동되었다는 점이다. 캐시 어레이(12)는 계속 세트 연상이지만, 그러나, 단지 한 세트가 모든 특정 사이클에서 캐시 어레이로부터 검색된다. 그러므로, 액세스되는 세트의 수를 감소시켜 전력 소모도 감소된다. 세트 선택 정보(28)가 틀리다면, TLB(26)로부터의 낮은 선택 정보 및 캐시 디렉토리(10)는 맞는 어레이 정보를 선택하여 액세스하기 위해 사용될 수 있다.

도 4는 감소된 전력 소모를 위해 모두 변형된 캐시 어레이(12) 및 캐시 디렉토리(10)를 포함하는 본 발명의 실시예를 도시한다. 제2 및 제3 실시예 사이의 차이점 하나는 세트 선택 정보(28)가 캐시 디렉토리(10)의 액세스에 있어서 어느 세트를 출력할 지를 판단하는데 사용되기 보다는 캐시 디렉토리(10)를 액세스하기 전에 어느 세트의 태그를 액세스할지를 판단하는데 사용된다는 점이다. 세트 선택 정보(28)는 단지 한 세트만이 캐시 디렉토리(10) 및 캐시 어레이(12)로부터 출력되도록 캐시 어레이(12)의 세트는 물론 캐시 디렉토리(10)의 두 세트를 선택하는데 사용된다. 전력 소모는 캐시로의 액세스 도중에 액세스되어야 할 캐시 디렉토리(10) 및 캐시 어레이(12)의 세트 수를 감소시켜 감소된다. 세트 선택 정보(28)가 틀리다면, TLB(26) 및 캐시 디렉토리(10)로부터의 낮은 선택 정보는 맞는 어레이 정보를 선택하고 독취하기 위해 사용될 수 있다.

도 7은 CPU(40)에의 본 발명의 적용을 도시한다. 메모리 시스템(42)은 CPU(40)에 의해 연산될 명령 및 프로그램 데이터를 포함한다. 메모리 시스템(42)은 대개 주메모리(예를 들면, 동적 랜덤 액세스 메모리), 부저장장치(예를 들면, 자기/광학 디스크 드라이브, 전기적 삭제가능 프로그램가능 독취 전용 메모리)를 포함하는 계층적 구조를 구비한다.

메모리 시스템(42)은 자주 사용되는 명령 및 데이터를 각각 명령 캐시(44) 및 데이터 캐시(46)에 유지하기 위해서 명령 캐시(44) 및 데이터 캐시(46)에 인터페이스한다. 대개는 캐시에 저장된 명령 및 데이터는 캐시 라인 또는 블록으로 분할된다. 명령 폐치 유닛(48)은 명령 캐시(44)에 저장된 명령의 하나 이상의 블록을 명령 큐(50)에 폐치한다. 디코드 및 디스패치 유닛(52)은 명령 큐(50)로부터 명령을 독취하고 명령을 디코드하여 명령을 실행하기 위해 스케줄링한다. 스케줄링 기술의 상세한 설명은 본 명세서에 참조적으로 결합된 Hennessy 및 Patterson의 Computer Architecture: A Quantitative Approach, 2220-362쪽(1990)에서 찾을 수 있다. 조건적인 브랜치 명령은 브랜치 유닛(54)에서 실행되도록 스케줄링되고, 순차적 또는 비브랜치적 명령은 기능 유닛(56)에서 실행되도록 스케줄링된다.

로드 저장 유닛(58)은 데이터 캐시(56)로부터 기능 유닛(56)으로 필요한 만큼 데이터 블록을 폐치한다. 로드 저장 유닛(58)은 또한 데이터 캐시(56) 및 메모리 시스템(42)에 저장된 데이터 블록을 기능 유닛(56)의 명령 실행에 의해 발생한 데이터 값에 기초하여 갱신한다. CPU는 명령을 실행하기 위해 기능 유닛(56)에 피연산자를 제공하고 실행의 결과를 수신하는 레지스터 파일(60)을 포함한다. 완료 유닛(62)은 CPU(40)의 특정 구조에 의해 유효하다고 고려되는 차례로 명령의 실행을 완료한다. 기능 유닛(56)이 순서를 잘못하여 명령을 실행할 수 있기는 하지만, CPU(40)는 구조적 제약 조건에 따라 동일한

순서로 명령을 완료하도록 허용할 수도 허용하지 않을 수도 있다.

브랜치 유닛(54)은 조건적 브랜치의 결과를 계산하기 위해 사용되는 조건 레지스터 및 기능 유닛(56)에 의해 수행된 연산의 결과에 따라 조건 레지스터를 갱신하는 로직 회로를 포함한다. 메모리 시스템(42)으로부터 명령 큐(50)로 폐치될 명령의 블록을 식별하는 주소는 명령 폐치 레지스터(64)에 저장된다. 명령 폐치 레지스터(64)에 저장된 블록 주소는 브랜치 유닛(54)에 제공된다.

각 기능 유닛(56)은 디코드 및 디스패치 유닛(52)을 통해 발송된 명령 형태의 일부 또는 전부를 실행할 수 있다. 기능 유닛(56)은 입력 소스 피연산자를 레지스터 파일(60) 및/또는 데이터 캐시 유닛(46)으로부터 수신하거나 출력 결과를 기입할 수 있다. 대안예에서, 기능 유닛(56)은 비대칭적일 수 있고, 특수 유닛이 명령의 부분집합만을 실행할 수 있다. 이 경우, 레지스터 파일(60)은 대개 논리적으로 기능 유닛(56)의 기능에 해당하는 부분집합으로 분할된다. 레지스터 파일(60)의 논리적 분할은 구조적으로 가시적이거나 비가시적일 수 있음을 알자. 나아가, 레지스터 파일(60)은 논리적으로 분리되어 있으면서 물리적으로는 공통 회로를 공유할 수 있다. 또한, 디코드 디스패치 유닛(52)은 이 비대칭성을 알아야 하고 명령을 적절히 스케줄링해야 한다. 예를 들면, 기능 유닛(56)은 고정 소수점(fixed point) 연산을 수행하기 위한 하나 이상의 고정 소수점 유닛 및 부동 소수점(floating-point) 연산을 수행하기 위한 하나 이상의 부동 소수점 유닛 및 하나 이상의 소스 벡터에서 벡터 연산을 수행하기 위한 하나 이상의 벡터 유닛을 포함한다. 고정 소수점 유닛에 의해 수행된 고정 소수점 연산은, 예를 들면, 고정 소수점 가산, 고정 소수점 감산, 고정 소수점 적산, 고정 소수점 제산, 논리곱, 논리합, 배타적 논리합, 고정 소수점 비교, 고정 소수점 회전, 고정 소수점 쉬프트 연산을 포함할 수 있다. 부동 소수점 유닛에 의해 수행되는 부동 소수점 연산은, 예를 들면, 부동 소수점 가산, 부동 소수점 감산, 부동 소수점 적산, 부동 소수점 제산, 부동 소수점 비교 연산을 포함한다. 벡터 유닛에 의해 수행되는 벡터 연산은 특징적으로 고유의 고도한 병렬성을 가진다. 대개, 이러한 벡터 연산은 명령이 소스 벡터 내에서 다중 데이터 성분에서 동일한 연산을 수행하는 SIMD형 연산이다.

브랜치 유닛(54)은 또한 세트 선택 정보를 포함하는 브랜치 예측 장치를 포함한다. 브랜치 예측 장치는 명령의 블록이 조건적 브랜치 명령을 포함하지 않는지 또는 취해지지 않을 것으로 예측된 하나 이상의 조건적 브랜치 명령을 포함하는지를 판단하고, 브랜치 예측 장치는 다음 순차적 블록 주소(다음 순차적 블록 주소 발생기(68)에 의해 발생된)의 세트 및 블록 주소를 명령 폐치 레지스터(64)에 로드하도록 다중화기(66)를 제어한다. 이 경우, 다음 순차 블록은 메모리 시스템(42)로부터 명령 큐(50)로 폐치된다. 브랜치 예측 장치가 하나 이상의 명령이 취해질 것으로 예측된 브랜치 명령이라고 판단하면, 브랜치 예측 장치는 명령 폐치 레지스터(64)로 목표 주소의 세트 및 블록 주소를 로드하도록 다중화기(66)를 제어한다. 이 경우, 예측된 취해진 제1 브랜치 명령의 목표 명령을 포함하는 세트 및 블록은 메모리 시스템(42)으로부터 명령 큐(50)로 폐치된다. 이 처리는 명령 캐시의 액세스 대기 시간을 상당히 감소시키고 CPU(40)가 현재 실행하고 있는 블록의 앞으로 무한정 계속될 수 있다. 그러나, 명령 버퍼의 크기 및 일부 브랜치가 브랜치 예측 장치에 의해 틀리게 예측될 가능성의 증가하면 모든 더이상의 블록 전 폐치를 쓸모없게 하여 이 처리를 한정한다.

한 블록 내에서 취해질 것으로 예측된 하나 이상의 조건적 브랜치 명령이 있을 수 있음을 명백하게 알 수 있다. 이 장치가 예측된 취해진 제1 조건 브랜치 명령을 식별하기 위해서, 목표 명령의 바이트 주소가 명령 폐치 레지스터(64)에 저장되어 명령의 폐치 블록에서 메모리 시스템(42)에 의해 사용되지 않을 수 있다. 이 경우, 바이트 주소는 블록에서 목표 명령의 엔트리점을 표시하기 위해서 브랜치 예측 장치에 제공된다. 브랜치 예측 장치는 제공된 엔트리점 이후 순서대로 발생하는 취해질 것으로 예측된 제1 조건적 브랜치 명령을 식별하기 위해서 목표의 바이트 주소를 사용한다.

브랜치 예측 장치는 세트 선택 정보를 BTB, OHT, 명령 캐시 또는 디코드 히스토리 표에 저장된 데이터로부터 유도할 수 있다. 공지된 브랜치 예측 기술에 대한 설명은 다음과 참조 문헌에서 찾을 수 있다. 이 참조 문헌들은 본 명세서에 참조적으로 결합되어 있다.

1. Sussenguth에 허여된 미국특허제3,559,183호.
2. Losq 등에 허여된 미국특허제4,477,872호.
3. J. Smith의 A Study of Branch Prediction Strategies 24th Annual Symposium on Computer Architecture 135-148쪽 (1998년 5월).
4. Yeh 및 Patt의 A Comparison of Dynamic Branch Prediction that use Two Levels of Branch History Conference Proceedings Annual Symposium on Computer Architecture 257-266쪽 (1993).
5. P. Emma, J. Pomerene, T. Puzak, R. Rechtschaffen, F. Sparacio의 Operand History Table, IBM Technical Disclosure Bulletin 12-84, 3815-3816쪽.
6. Emma, J. Knight, J. Pomerene, T. Puzak, R. Rechtschaffen의 Hedge Fetch History Table, IBM Technical Disclosure Bulletin 02-89, 101-102쪽.
7. Simon Stely Jr 등에 허여된 미국특허제 5,235,697호.
8. L. Liu에 허여된 미국특허제 5,418,922호.
9. L. Liu의 Partial address directory for cache access, IEEE Transaction on Very Large Scale Integration (VLSI) Systems, 2(2), 1994년 6월, 226-240쪽.

(Liu는 SETLAT라는 고정 소수점을 사용하는데, 이는 연관된 캐시로부터 보다 빠르게 데이터를 검색하기 위해 로드 주소의 일부를 사용하여 인덱스된다. SETLAT이 번역을 필요로 하지 않고 종래의 디렉토리보다 협소하므로, 일치 등급으로부터 세트의 보다 신속한 선택을 가능하게 한다. Liu의 발명은 또한 가상 주소 번역을 캐시하는데 사용될 수 있고 싸이클 당 다중 캐시 액세스를 가능하게 한다.)

10. A. Agarwal 및 S. Pudar의 Column-Associative Caches: A Technique for Reducing the Miss Rate of Direct-Mapped Caches, The 20th Annual International Symposium on Computer Architecture (1993) 179-190쪽.

(Agarwal 및 Pudar는 직접 대응 캐시를 제2 해싱(second-hashing)을 사용하는 2가지 세트 연상 캐시와 같이 수행하도록 하는 방법을 개시한다. 이 기술로써, 세트의 가장 최근 사용된(MRU) 성분으로 보다 빠르게 액세스할 수 있다.)

11. B. Calder 및 d. Grunwald 및 J. Emer의 "Predictive sequential associative cache", IEEE High-Performance Computer Architecture Symposium Proceedings 1996, 244-253쪽.

(Agarwal 및 Pudar 위에 구현된 연구에서 Calder 등은 다른 해시 프로브(hash probe)를 가지고 여러번 직접 대응 캐시를 액세스하기 위한 적절한 캐시 해시 프로브를 선택하기 위해 예측 기술을 사용하는 방식을 개시한다.)

다시, 도 7을 참조하면, 처리기 유닛(40)은 처리기 유닛(40)에 의해 이러한 정보가 요청되기 전에 명령 및 데이터를 캐시 장치(44, 46)에 전페치(prefetch)하는 부가적인 하드웨어 장치를 포함할 수 있다. 예를 들면, 처리기 유닛(40)은 본 명세서에 참조적으로 결합된 주피(Jouppi)의 "Improving Direct-mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Pre-fetch buffers - Proc. of 17th annual Int'l Symposium on Computer Architecture", 1990년 5월 시애틀, IEEE Computer Society Press, 364-373쪽 및 파라찰라(Palacharla) 등의 "Evaluation Streaming Buffers as a Secondary Cache Replacement - Proc. of 21st annual Int'l Symposium on Computer Architecture", 1994년 4월 시카고, IEEE Computer Society Press, 24-33쪽에서 앞서 기술된 것과 같은 전페치 스트림 버퍼(prefetch stream buffer)를 포함할 수 있다.

또한, 전페치 명령(즉, 이러한 정보가 처리기 유닛(40)에 의해 요청되기 전에 명령 및 데이터를 캐시 메모리(44, 46)로 전페치하는 명령)은 메모리 시스템(42)에 저장된 명령의 시퀀스에 삽입될 수 있다. 본 명세서에 참조적으로 결합된 칼라한(Callahan) 등의 1991년 4월 "Software Prefetch Proceedings of the Fourth International Conference on Architecture Support For Programming Languages and Operating Systems"은 명령 세트에 전페치를 수행하는 새로운 명령을 추가하는 것을 개시하고 있다. IBM(상표명), Motorola(상표명), Apple(상표명)에 의해 함께 개발된 PowerPC(상표명)에서, 통상 터치 명령(touch-instruction)이라 지칭되는 Data-Cache-Block-Touch (dcbt) 명령은 메모리 블록을 캐시로 전페치하는데 사용된다. 또한, 본 발명의 양수인에 공동으로 양수되고 본 명세서에 참조적으로 결합된 P. Dubey에 허여된 미국특허 제5,774,685호(출원번호08/426,502)는 추측(speculative) 터치 명령을 설명한다.

이 전페치 명령은 명령 및 데이터(피연산자) 모두를 캐시 메모리(44, 46)로 전페치하는데 사용될 수 있다. 일반적으로, 이 전페치 명령은 데이터가 처리기 유닛(40)에 전달되지 않는 정만 제외하고는 로드 명령과 같이 동작한다. 구체적으로, 처리기 유닛(40)은 캐시 디렉토리를 페치 주소를 가지고 탐색하고 미스가 발생하면 메모리 시스템(42)에서 캐시 메모리(44, 46)로 정보를 전달하기 위한 장치를 포함한다. 이 전페치 명령은 컴파일러에 의해 정보가 필요시 캐시로 사용가능할 지를 보장하기 위해서 정보를 실제 사용하기 전에 명령 시퀀스에 삽입될 수 있다. 불행히도, 모든 경우에 캐시 미스를 방지하기 위해서 컴파일러가 전페치 명령을 삽입하는 것은 불가능할 것이다. 예를 들면, 전페치 주소는 데이터를 사용하는 명령이 실행될 때까지 알려지지 않을 수 있다. 피연산자를 간접적으로 참조하는 로드 명령을 고려하자. 즉, 레지스터는 메모리에 저장된 포인터를 로드한다. 이 경우, 전페치 주소는 데이터를 식별하는 포인터가 로드될 때까지 알려지지 않는다.

다른 경우에, 데이터를 실제 사용하는 것과 너무 근접하면 전페치 명령은 성능면에서 유리함이 거의 없을 것이다. 예를 들면, 전페치된 명령을 정보의 실제 사용보다 단지 한 명령 앞에 놓는다면 이는 전페치 명령을 전혀 삽입하지 않는 것에 비해 있다해도 아주 적은 성능 이점만을 보일 것이다. 그러므로, 어떤 경우에는 비유용한 전페치 명령을 무시하는 것이 유리할 수 있다. 선택적으로 전페치 명령을 실행하기 위한 장치의 예는 본 명세서에 참조적으로 결합된 T. Puzak에 의해 출원된 공동으로 양수된 미국특허출원 제08/599,833호에 설명되어 있다.

명령 및 데이터를 전페치하는 것 외에도, 처리기 유닛(40)은 전페치된 명령을 추측적으로 실행할 수 있다. 이러한 명령은 이러한 명령의 실행 결과가 추측이 틀린 경우에 유용하지 않을 것이라는 사실로 인해 "추측 명령(speculative instruction)"이라 지칭된다. 양호하게는 처리기 유닛(40)은 추측적으로 명령을 실행하는 경로를 식별하기 위해 브랜치 예측을 사용한다. 이 경우, 브랜치 유닛(54)은 예측의 유효성을 조건적 브랜치 명령의 결정에 따라 판별한다. 주어진 예측이 유효하다고 결정되면(추측이 유효하다고 판별되면), 유효한 예측에 따른 이 추측 명령의 실행 결과는 영구적인 것이 된다. 그러나, 주어진 예측이 비유효한 것으로 결정되면(즉, 추측이 비유효하다고 판별되면), 유효한 예측에 따른 이 추측 명령의 모든 실행 결과는 제거되고, 처리기 유닛(40)의 단계에 의한 이러한 추측 명령의 처리는 종료될 수 있다. 추측 명령은 양호하게는 명령 큐(50)에서 그렇게 표시된다.

또한, 처리기 유닛(40)의 여러 단계(브랜치 유닛(54), 디코드 및 디스패치 유닛(52), 로드/저장 유닛(58), 기능 유닛(56))는 양호하게는 주어진 단계에 의해 실행되고 있는 추측 명령을 식별하는 정보를 저장한다. 또한, 완료 유닛(62)은 양호하게는 이러한 추측 명령을 추측적인 것으로 실행 결과를 표시한다. 주어진 추측이 유효하다고 판단한 후에 브랜치 유닛(54)은 유효한 예측을 영구적인 것으로 의존하는 추측 명령의 실행 결과를 표시하여 이러한 명령(즉, 레지스터 파일(60) 또는 메모리 시스템(42)에 실행 결과를 기입)을 완료 유닛(62)이 완료하게 하기 위해서 완료 유닛(62)과 협동적으로 동작한다. 한편, 주어진 예측이 비유효하다고 판단한 후에, 브랜치 유닛(54)은 다음과 같이 양호하게 연산한다. 브랜치 유닛(54)은 비유효한 예측에 따라 명령 큐(50)에 남아있는 순리적인 명령을 제거하기 위해서 명령 페치 유닛(48)과 협동적으로 연산한다. 브랜치 유닛(54)은 비유효한 예측에 따른 순리적인 명령의 실행 결과를 제거하여 완료 유닛(62)이 이러한 명령을 완료하기 못하게 하기 위해 완료 유닛과 협동적으로

로 연산한다. 브랜치 유닛(54)은 비유효한 예측에 따른 추측 명령의 처리를 종료하기 위해서 처리기 유닛(40)의 여러 단계에서 협동적으로 연산한다.

세트 예측 정보를 사용하는 예가 도 8A 및 8B에 도시되어 있다. 도 8A 및 8B는 많이 파이프라인화된 캐시에 대한 타이밍도를 도시한다. 도 8A는 세트 예측이 없는 파이프라인화의 종래 방법을 도시하고 도 8B는 본 발명에 따른 세트 예측을 사용하는 방법을 도시한다. 도 8A에 도시된 종래 방법에서 캐시 액세스 및 어레이 독취는 영과 일 사이의 시간 주기 동안 발생됨을 볼 수 있다. 같은 시간 근처에 가상 주소 번역(V→R), 디렉토리 독취, 태그 점검이 데이터 A에 대해 수행된다. 늦은 선택(인덱스화되어 결정된 세트) 이후, 데이터 A는 시간 주기 2 동안 가용으로 된다. 도 8B에 도시된 대로 본 발명의 도시적인 방법을 사용하면 데이터 A는 늦은 선택이 주기 '2'에서 수행되기 전에 시간 주기 '1' 도중에 독취된 어레이 이후 가용이다. 데이터 'A'는 시간 주기 '2'에서 유효한 데이터로 확인된다.

세트 예측이 틀린 경우에는 캐시 액세스, 어레이 독취, 데이터 가용 단계는 제2 시간(시간 주기 '3' 동안)에 수행되어야 한다. 데이터 A는 시간 4까지는 가용으로 되지 않는다. 데이터 'B' 및 'C'는 유효 데이터일 수 있음을 알자. 이들은 재유효화된 데이터 'A' 이전에 가용으로 된다.

본 발명이 특정한 예로 기술되고 예시되었지만, 본 개시는 예시로서만 제시된 것이고, 본 기술의 숙련자는, 부분의 결합 및 배열에서의 많은 변화가 청구된 본 발명의 사상 및 범위로부터 벗어나지 않고 존재할 수 있다는 것을 알 수 있을 것이다.

발명의 효과

세트 예측을 사용하여 세트 연상 캐시 액세스의 늦은 선택 단계를 대기할 필요가 제거된다.

(57) 청구의 범위

청구항 1

세트 연상 캐시(set-associative cache)를 구비한 메모리에서 액세스 대기 시간을 감소시키기 위한 방법에 있어서,

최소한 하나의 선택 캐시(cache)를 통해 선택가능한 데이터 선택 정보(data select information)에 기초하여 상기 메모리의 위치로부터 데이터가 독취가능하며,

상기 방법은,

세트 선택 정보(set selection information)를 발생시키는 단계, 및

복수의 위치 중의 최소한 하나의 위치에 상기 세트 선택 정보를 저장하는 단계, 및

상기 최소한 하나의 선택 캐시를 통한 상기 데이터 선택 정보의 판단 및 도착 이전에 상기 메모리로부터 상기 데이터의 검색을 위해 상기 세트 선택 정보를 발송 및 사용하는 단계

를 포함하는 액세스 대기 시간 감소 방법.

청구항 2

제1항에 있어서, 상기 복수의 위치 중의 하나는 브랜치 목표 버퍼(branch target buffer)이고, 상기 액세스된 메모리는 명령 캐시(instruction cache)인 액세스 대기 시간 감소 방법.

청구항 3

제2항에 있어서, 상기 세트 선택 정보를 브랜치 목표 버퍼에 저장하는 단계는 상기 브랜치 목표 버퍼에 제2 세트 선택 정보를 저장하는 단계를 포함하되, 상기 제2 세트 선택 정보는 실패 경로(fall-through path)에 대한 세트를 예측하는 액세스 대기 시간 감소 방법.

청구항 4

제2항에 있어서, 상기 세트 선택 정보를 브랜치 목표 버퍼에 저장하는 단계는 상기 브랜치 목표 버퍼에 제2 세트 선택 정보를 저장하는 단계를 포함하되, 상기 제2 세트 선택 정보는 취해지지 않은 경로(not-taken path)에 대한 세트를 예측하는 액세스 대기 시간 감소 방법.

청구항 5

제1항에 있어서, 상기 복수의 위치 중의 하나는 명령 캐시에 있고, 상기 액세스된 메모리는 데이터 캐시인 액세스 대기 시간 감소 방법.

청구항 6

제1항에 있어서, 상기 복수의 위치 중의 하나는 피연산자 히스토리 표(operand history table)이고, 상기 액세스된 메모리 캐시는 데이터 캐시인 액세스 대기 시간 감소 방법.

청구항 7

제1항에 있어서,

상기 세트 선택 정보를 상기 데이터 선택 정보와 비교하는 단계, 및

상기 세트 선택 정보를 갱신하고 상기 비교로부터 불일치 판단시 독취된 비유효 데이터를 신호하는 단

계, 및

다음 수준 메모리 계층으로부터 명령 또는 데이터를 폐치하는 단계를 더 포함하는 액세스 대기 시간 감소 방법.

청구항 8

세트 연상 캐시에서 액세스 대기 시간을 감소시키기 위한 장치에 있어서,

- a) 세트 선택 정보를 저장하기 위한 저장 수단,
- b) 주소를 수신하고 상기 주소에 의해 판단된 M 세트의 데이터를 출력하기 위한 M방식(M-way) 세트 연상 캐시, 및
- c) 세트 선택 정보-상기 세트 선택 정보는 상기 데이터를 액세스하기 위한 상기 세트 연상 주소 이전에 사용가능하게 됨- 및 세트 연상 주소 중의 하나를 다중화하기 위한 다중화기를 포함하는 액세스 대기 시간 감소 장치.

청구항 9

제8항에 있어서, 상기 세트 선택 정보는 브랜치 목표 버퍼에 저장되거나 브랜치 목표 버퍼로부터 검색되고 상기 M방식 세트 연상 캐시는 명령 캐시인 액세스 대기 시간 감소 장치.

청구항 10

제8항에 있어서, 상기 세트 선택 정보는 명령 캐시에 저장되거나 명령 캐시로부터 검색되고 상기 M방식 세트 연상 캐시는 데이터 캐시인 액세스 대기 시간 감소 장치.

청구항 11

제8항에 있어서, 상기 세트 선택 정보는 피연산자 히스토리 표에 저장되거나 검색되고 상기 M방식 세트 연상 캐시는 데이터 캐시인 액세스 대기 시간 감소 장치.

청구항 12

세트 연상 캐시의 전력 소모를 감소시키기 위한 장치에 있어서,

- a) 세트 선택 정보를 저장하기 위한 세트 선택 저장 수단, 및
- b) 어레이(array) 및 디렉토리(directory)-상기 디렉토리는 주소의 세트 연상 태그 부분을 상기 어레이로 출력함-를 포함하는 M방식 세트 연상 캐시, 및
- c) 상기 디렉토리로부터의 주소의 상기 태그 부분과 상기 M 세트 데이터 중의 한 세트를 출력하기 위한 상기 세트 선택 정보 중의 하나를 다중화하기 위한 다중화기를 포함하는 전력 소모 감소 장치.

청구항 13

데이터 주소를 사용하는 세트 연상 메모리의 액세스 속도를 증가시키기 위한 방법에 있어서,

상기 각각의 주소는 오프셋(offset) 부분, 일치 등급 인덱스(congruence class index), 태그(tag) 부분을 포함하고, 상기 세트 연상 메모리는 어레이 및 디렉토리를 포함하고, 상기 어레이는 데이터를 저장하며, 상기 어레이는 복수의 어레이 일치 등급으로 분할되고, 상기 어레이 일치 등급은 어레이 세트로 분할되며, 상기 어레이 세트는 데이터를 구비하는 캐시 라인(cache line)을 포함하고, 상기 디렉토리는 복수의 디렉토리 일치 등급으로 분할되며, 상기 디렉토리 일치 등급은 디렉토리 세트로 분할되고, 상기 디렉토리 세트는 디렉토리 엔트리(entry)를 포함하며, 상기 디렉토리 엔트리는 주소 태그를 포함하며, 상기 디렉토리는 상기 디렉토리 엔트리와 상기 캐시 라인 사이에 상기 주소 태그가 상기 캐시 라인의 최소한 하나와 연관되도록 일대일 대응되도록 분할되며,

상기 방법은,

상기 일치 등급 인덱스를 사용하여 단일 어레이 일치 등급 - 상기 단일 어레이 일치 등급은 상기 일치 등급 인덱스에 의해 규정됨 - 의 세트의 내용을 액세스하는 단계,

상기 일치 등급 인덱스를 사용하여 단일 디렉토리 일치 등급-상기 단일 디렉토리 일치 등급은 상기 일치 등급 인덱스에 의해 규정됨-의 세트의 내용을 액세스하는 단계,

세트 선택 정보를 발생시키는 단계, 및

상기 디렉토리에 의해 규정된 상기 단일 일치 등급의 출력 이전에 상기 어레이 일치 등급의 상기 세트를 선택하기 위해서 상기 세트 선택 정보를 사용하는 단계, 및

상기 세트 선택 정보에 기초하여 상기 선택된 세트의 상기 캐시 라인으로부터 상기 데이터를 출력하는 단계

를 포함하는 액세스 속도 증가 방법.

청구항 14

제13항에 있어서,

상기 태그 부분과 상기 디렉토리 일치 등급의 상기 선택된 세트의 상기 주소 태그를 비교하는 단계, 및
상기 선택된 일치 등급의 상기 주소 태그 중의 하나가 상기 주소의 상기 태그 부분과 동일하면 상기 선택된 세트를 상기 세트 선택 정보와 비교하는 단계, 및

상기 선택된 일치 등급의 상기 주소 태그의 어느 것도 상기 주소의 상기 태그 부분과 일치하지 않으면
상기 액세스가 성공적이지 않음 및 상기 캐시 라인으로부터 출력된 상기 데이터가 비유효함을 표시하는
제1 제어 신호를 출력하는 단계, 및

상기 선택된 세트가 상기 세트 선택 정보와 동일하지 않으면 상기 캐시 라인으로부터의 상기 데이터가
비유효함을 표시하는 제2 제어 신호를 출력하는 단계

를 포함하는 액세스 속도 증가 방법.

청구항 15

제13항에 있어서, 상기 주소는 제1 분할부 및 제2 분할부를 포함하되, 상기 제1 분할부는 어레이 오프셋
부분, 어레이 일치 등급 인덱스, 어레이 태그 부분을 포함하고, 상기 제2 분할부는 디렉토리 부분, 디렉
토리 일치 등급 인덱스, 디렉토리 태그 부분을 포함하며, 상기 어레이 일치 등급 인덱스 및 상기 디렉토
리 일치 등급 인덱스는 다른 길이를 가지고, 상기 어레이 및 상기 디렉토리는 다른 세트 결합성을 포함
하는 액세스 속도 증가 방법.

청구항 16

제15항에 있어서, 출력될 상기 캐시 라인으로부터 데이터를 선택하기 위해서 상기 오프셋 부분을 사용하
는 단계를 더 포함하는 액세스 속도 증가 방법.

청구항 17

제13항에 있어서, 예측된 세트가 아닌 다른 세트가 상기 어레이 태그 부분과 일치하는 주소 태그를 가진
다면 상기 선택된 세트의 상기 캐시 라인으로부터 데이터를 출력하는 단계를 더 포함하는 액세스 속도
증가 방법.

청구항 18

제15항에 있어서, 예측된 세트가 아닌 다른 세트가 상기 디렉토리 태그 부분과 일치하는 상기 주소 태그
를 포함한다면 상기 선택된 세트의 상기 캐시 라인으로부터 데이터를 출력하는 단계를 더 포함하는 액세
스 속도 증가 방법.

청구항 19

제13항 또는 제15항에 있어서, 상기 세트 선택 정보는 브랜치 목표 버퍼에 의해 제공되는 액세스 속도
증가 방법..

청구항 20

제13항 또는 제15항에 있어서, 피연산자 액세스에 대한 상기 세트 선택 정보는 명령 액세스 이전에 제공
되는 액세스 속도 증가 방법.

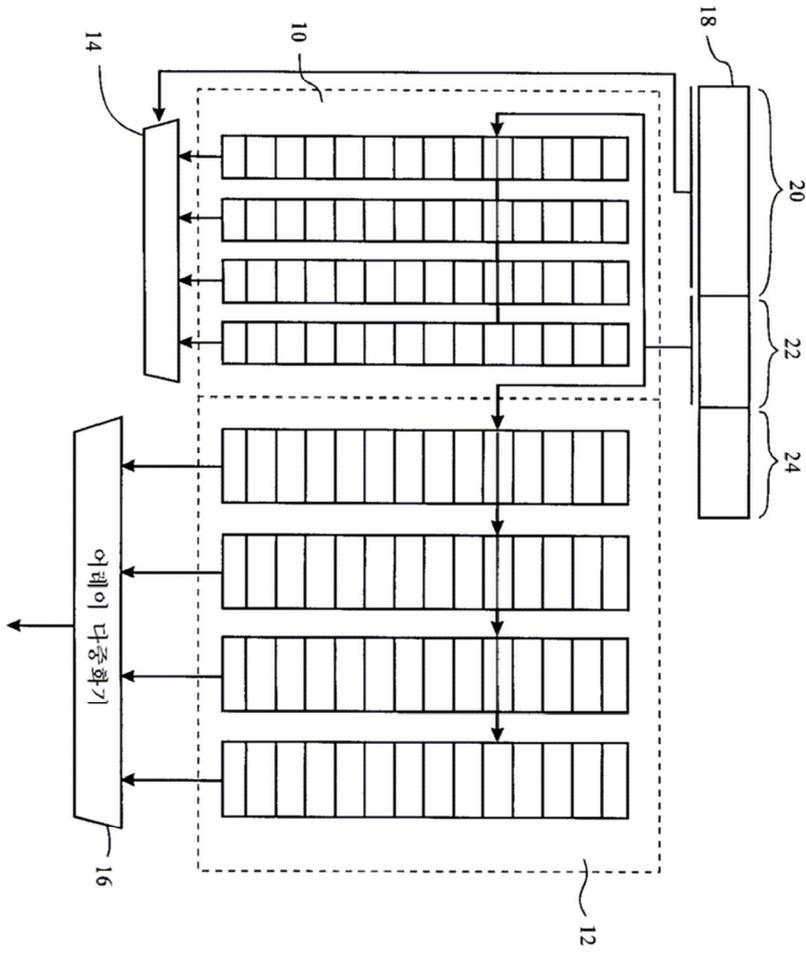
청구항 21

제13항 또는 제15항에 있어서, 피연산자 액세스에 대한 상기 세트 선택 정보는 피연산자 히스토리 표에
의해 제공되는 액세스 속도 증가 방법.

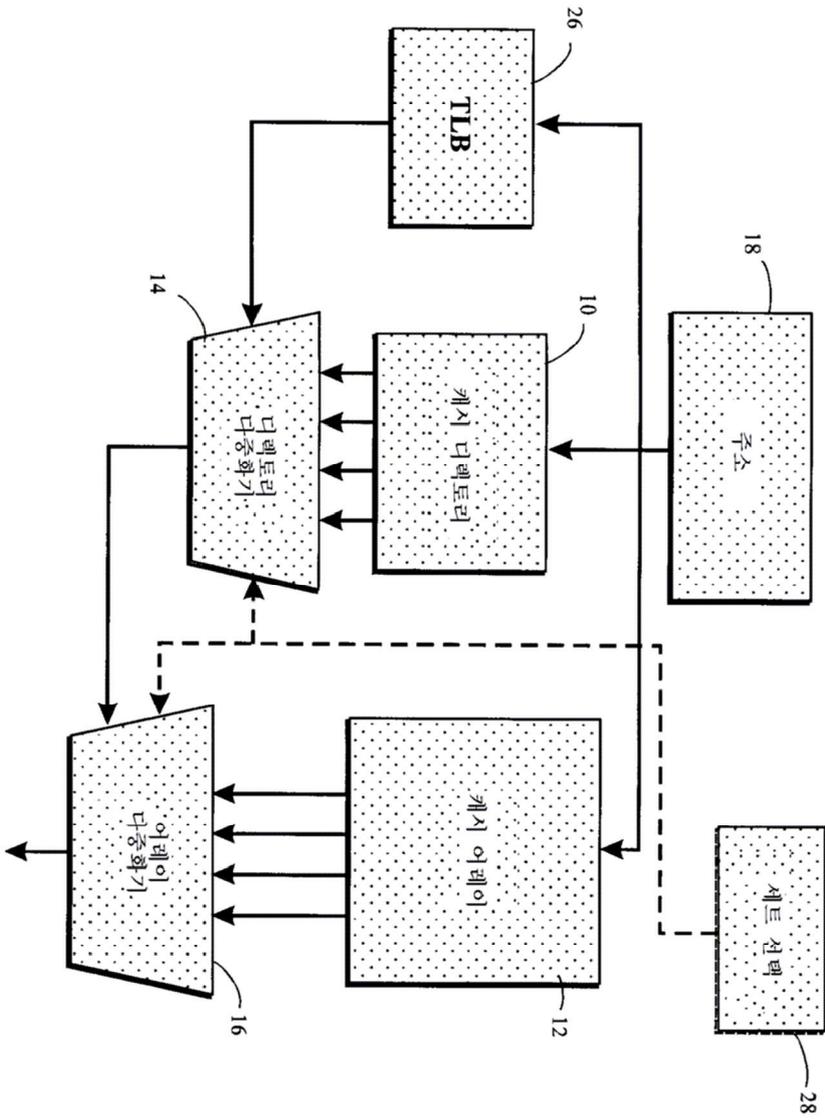
도면

도면1

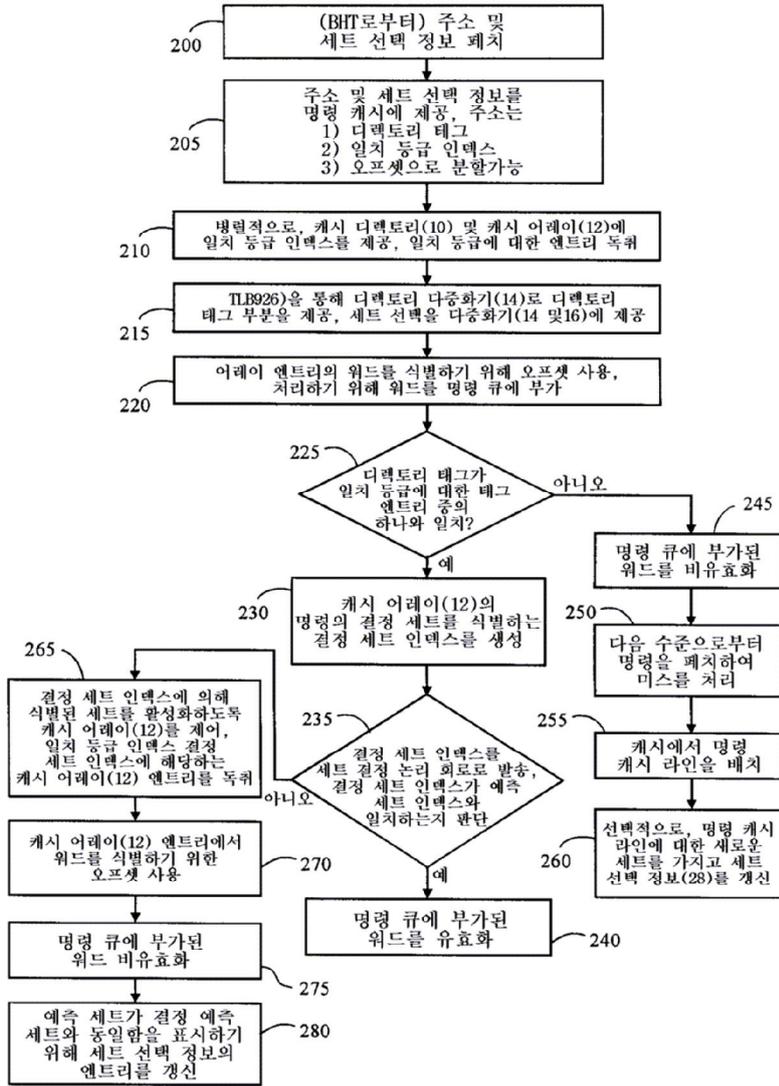
(종래 기술)



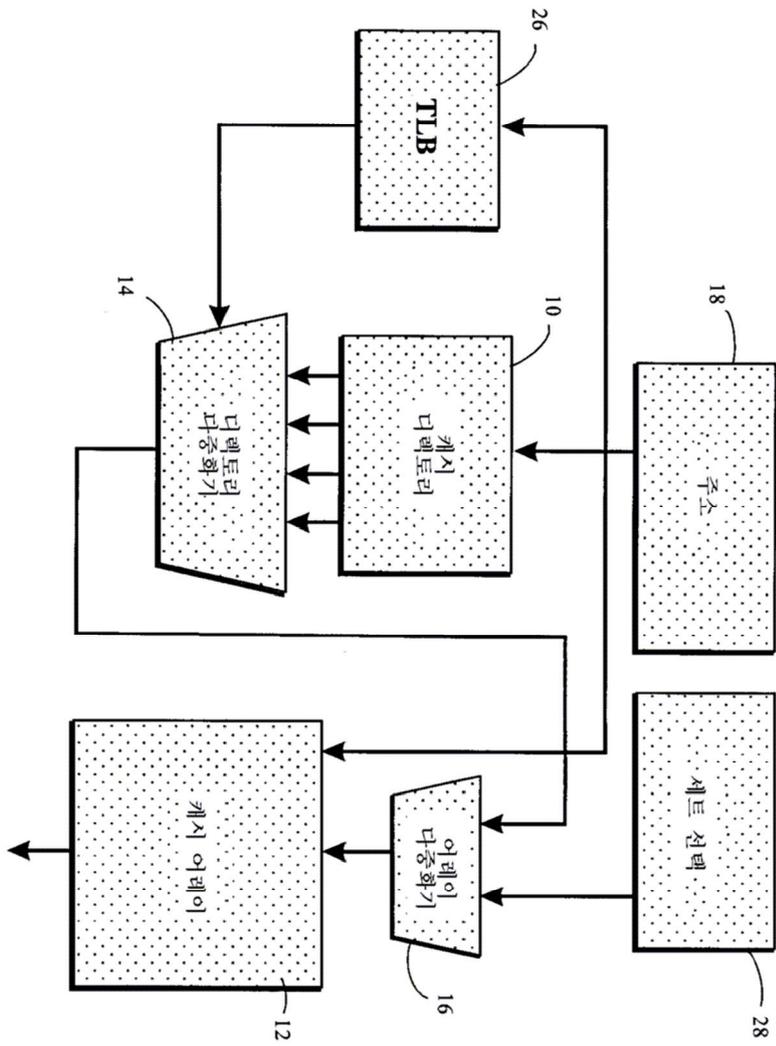
도면2



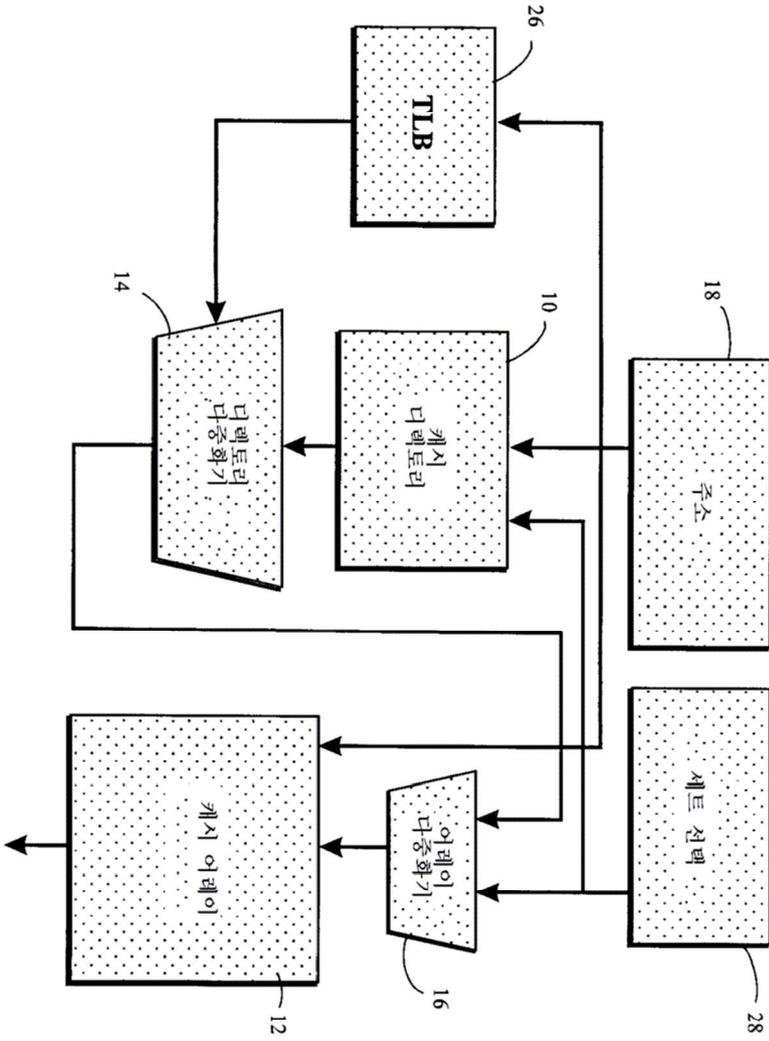
도면2a



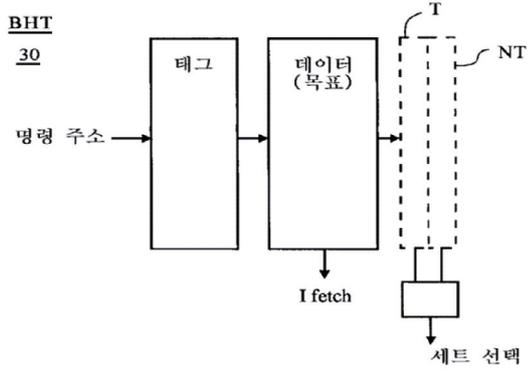
도면3



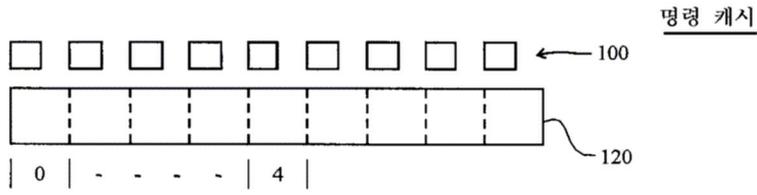
도면4



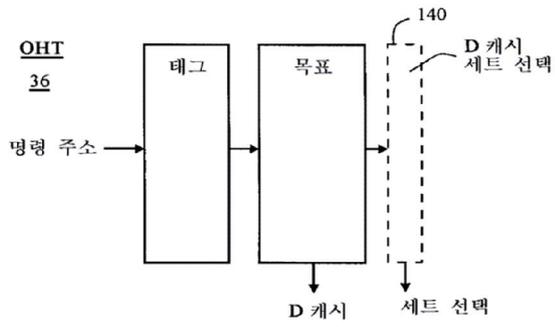
도면5a



도면5b



도면5c



도면6

(종래 기술)

