



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년08월24일
(11) 등록번호 10-1059335
(24) 등록일자 2011년08월18일

(51) Int. Cl.
G06F 9/06 (2006.01) G06F 9/30 (2006.01)
G06F 9/32 (2006.01)
(21) 출원번호 10-2009-7008741
(22) 출원일자(국제출원일자) 2007년09월28일
심사청구일자 2009년04월28일
(85) 번역문제출일자 2009년04월28일
(65) 공개번호 10-2009-0061075
(43) 공개일자 2009년06월15일
(86) 국제출원번호 PCT/US2007/079864
(87) 국제공개번호 WO 2008/039975
국제공개일자 2008년04월03일
(30) 우선권주장
11/536,743 2006년09월29일 미국(US)
(56) 선행기술조사문헌
KR1020050113499 A*
US06157980 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
칼컴 인코포레이티드
미국 캘리포니아 샌디에고 모어하우스
드라이브5775 (우 92121-1714)
(72) 발명자
스미쓰, 로드니 웨인
미국 92121 캘리포니아 샌디에고 모어하우스 드라
이브 5775
스텝펠, 브라이언 마이클
미국 92121 캘리포니아 샌디에고 모어하우스 드라
이브 5775
(74) 대리인
남상선

전체 청구항 수 : 총 16 항

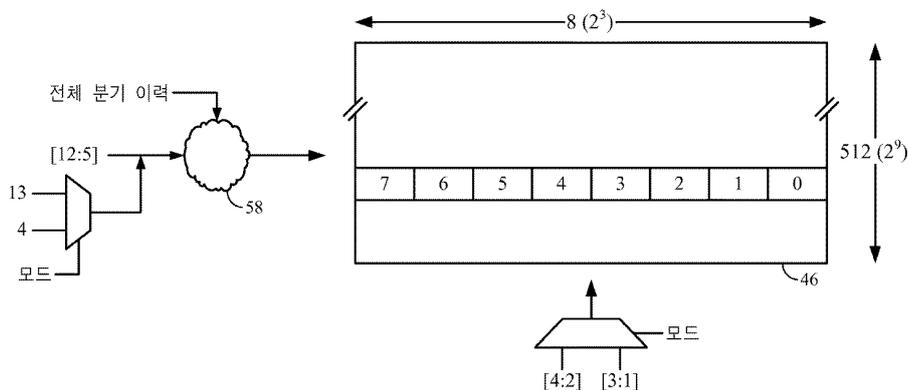
심사관 : 복진요

(54) 가변 길이 명령 세트 실행 모드들을 가지는 프로세서 내의 BHT의 효율적 사용방법

(57) 요약

적어도 제 1 최소 명령 길이를 가지는 제 1 명령 세트 실행 모드 및 더 작은 제 2 최소 명령 길이를 가지는 제 2 명령 세트 실행 모드에서 명령들을 실행하는 프로세서에서, 라인 및 카운터 인덱스 어드레스들은 분기 이력 테이블(BHT) 내의 모든 카운터를 액세스하도록 형성되며, 현재 명령 세트 실행 모드에 기초하여 멀티플렉싱된 인덱스 어드레스 비트들의 수를 감소시킨다. 일 실시예에서, BHT 라인 내의 카운터들은 상기 BHT의 절반이 하나의 명령 세트 실행 모드에서 각각의 액세스에 대해 파워 다운될 수 있는 방식으로 정렬되고 인덱싱된다.

대표도



특허청구의 범위

청구항 1

적어도 제 1 최소 명령 길이를 가지는 제 1 명령 세트 실행 모드 및 더 작은 제 2 최소 명령 길이를 가지는 제 2 명령 실행 모드에서 명령들을 실행하는 가변 길이 명령 세트 프로세서에서 분기 이력 테이블(branch history table ; BHT) 내의 라인들을 인덱싱하기 위한 어드레스를 형성하는 방법으로서, 각각의 명령 세트 실행 모드는 상이한 인덱스 어드레스를 가지며, 상기 방법은,

명령 어드레스의 공통 라인 인덱스 어드레스 비트들을 식별하는 단계 - 여기서, 상기 공통 라인 인덱스 어드레스 비트들은 상기 분기 이력 테이블(BHT)을 인덱싱함에 있어서 상기 제1 및 제2 명령 세트 실행 모드들 모두에 의해 사용되며, 상기 공통 라인 인덱스 어드레스 비트들은 상기 제1 및 제2 명령 세트 실행 모드들에 의해 사용되는 명령 어드레스의 상이한 위치들을 점유함 -;

명령 세트 실행 모드 표시자에 기초하여, 상기 제 1 명령 세트 실행 모드에 대해 독점적인(exclusive) 최상위 라인 인덱스 어드레스 비트들과 상기 제 2 명령 세트 실행 모드에 대해 독점적인 하나 이상의 최하위 라인 인덱스 어드레스 비트들 간에 멀티플렉싱하는 단계;

연결된 어드레스를 형성하기 위해서 상기 공통 어드레스 비트들과 상기 멀티플렉싱된 라인 어드레스 비트들을 연결(concatenate)시키는 단계; 및

상기 연결된 어드레스를 사용하여 상기 제1 및 제2 명령 세트 실행 모드들 모두에서 사용될 수 있는 카운터 값들을 리트리브(retrieve)하기 위해 상기 BHT 내의 라인들을 인덱싱하는 단계를 포함하는,

방법.

청구항 2

제1항에 있어서,

BHT 내의 라인들을 인덱싱하기 전에 전체(global) 분기 이력 값을 사용하여 상기 연결된 어드레스를 해싱처리(hashing)하는 단계를 추가로 포함하는,

방법.

청구항 3

제2항에 있어서,

상기 전체 분기 이력 값을 사용하여 상기 연결된 어드레스를 해싱처리(hashing)하는 단계는 상기 연결된 어드레스와 전체 분기 이력 값 간에 배타적-논리합(logical exclusive-or)을 수행하는 단계를 포함하는,

방법.

청구항 4

제1항에 있어서,

인덱싱된 라인에 대한 카운터 인덱스 어드레스는 상기 제 1 명령 세트 실행 모드의 카운터 인덱스 어드레스와 상기 제 2 명령 세트 실행 모드의 카운터 인덱스 어드레스 간의 멀티플렉싱에 의해 형성되는,

방법.

청구항 5

제1항에 있어서,

상기 제 1 최소 명령 길이는 32비트이고, 상기 제 2 최소 명령 길이는 16비트인,

방법.

청구항 6

제4항에 있어서,

상기 BHT는 8개 카운터 각각의 512 라인들을 포함하고,

상기 공통 라인 인덱스 어드레스 비트들은 비트 [12:5]를 포함하고;

상기 제 1 명령 세트 실행 모드에 독점적인 하나 이상의 최상위 라인 인덱스 어드레스 비트들은 비트 13을 포함하고; 그리고

상기 제 2 명령 세트 실행 모드에 독점적인 하나 이상의 최하위 라인 인덱스 어드레스 비트들은 비트 4를 포함하는,

방법.

청구항 7

제5항에 있어서,

인덱싱된 라인에 대한 카운터 인덱스 어드레스는 제 1 명령 세트 실행 모드에서의 어드레스 비트들 [4:2]과 제 2 명령 세트 실행 모드에서의 어드레스 비트들 [3:1] 간의 멀티플렉싱에 의해 형성되는,

방법.

청구항 8

적어도 제 1 최소 명령 길이를 가지는 제 1 명령 세트 실행 모드 및 더 작은 제 2 최소 명령 길이를 가지는 제 2 명령 세트 실행 모드에서 명령들을 실행하는 가변 길이 명령 세트 프로세서에서 분기 이력 테이블(BHT) 내의 카운터들에 액세스하기 위한 라인 및 카운터 인덱스 어드레스들을 형성하는 방법으로서,

상기 제 2 명령 세트 실행 모드의 라인 인덱스 어드레스를 사용하여 상기 BHT 내의 라인들을 인덱싱하는 단계;

명령 세트 실행 모드 표시자에 기초하여, 상기 제 1 명령 세트 실행 모드에 독점적인 하나 이상의 라인 인덱스 어드레스 비트들의 최상위 위치들 및 상기 제 2 명령 세트 실행 모드에 독점적인 하나 이상의 카운터 인덱스 어드레스 비트들의 최하위 위치들 간에 멀티플렉싱하는 단계; 및

멀티플렉싱된 비트들과 연결된, 상기 제 2 명령 세트 실행 모드의 최하위 카운터 인덱스 어드레스 비트들을 제외한 모두를 사용하여 각각의 BHT 라인 내의 카운터들을 인덱싱하는 단계를 포함하는,

방법.

청구항 9

삭제

청구항 10

삭제

청구항 11

제8항에 있어서,

상기 제 1 최소 명령 길이는 32비트이고 상기 제 2 최소 명령 길이는 16비트인,

방법.

청구항 12

제11항에 있어서,

상기 BHT는 8개 카운터들 각각의 512라인들을 포함하고,

상기 제 2 명령 세트 실행 모드의 라인 인덱스 어드레스 비트들은 비트 [12:4]를 포함하고,

상기 제 1 명령 세트 실행 모드에 독립적인 하나 이상의 최상위 라인 인덱스 어드레스 비트들은 비트 13을 포함하고 상기 제 2 명령 세트 실행 모드에 독립적인 하나 이상의 최하위 카운터 인덱스 어드레스 비트들은 비트 1을 포함하고, 그리고

상기 제 2 명령 세트 실행 모드의 상기 최하위 카운터 인덱스 어드레스 비트들을 제외한 모두는 비트들 [3:2]를 포함하는,

방법.

청구항 13

적어도 32 비트 명령 길이를 가지는 32 비트 명령 세트 실행 모드 및 16 비트 명령 길이를 가지는 16 비트 명령 실행 모드에서 명령들을 실행하는 가변 길이 명령 세트 프로세서에서 분기 이력 테이블(branch history table ; BHT) 내의 라인들을 인덱싱하기 위한 어드레스를 형성하는 방법으로서, 상기 32 비트 명령 세트 실행 모드 및 상기 제16 비트 명령 세트 실행 모드 각각은 상이한 어드레스 비트 위치들에서의 비트들을 사용하는 상이한 인덱스 어드레스를 사용하며, 상기 방법은,

명령 어드레스의 공통 어드레스 비트들을 식별하는 단계 - 여기서, 상기 공통 어드레스 비트들은 상기 분기 이력 테이블(BHT)을 인덱싱함에 있어서 상기 32 비트 및 16 비트 명령 세트 실행 모드들 모두에 의해 사용되며, 상기 공통 어드레스 비트들은 32 비트 명령 어드레스 및 16 비트 명령 어드레스의 비트들을 포함함 -;

멀티플렉싱된 어드레스 비트를 식별하기 위해서 명령 세트 실행 모드 표시자에 기초하여, 상기 32 비트 명령 어드레스의 비트 13 및 상기 16 비트 명령 어드레스의 비트 4 간에 멀티플렉싱하는 단계;

연결된 어드레스를 형성하기 위해서 상기 공통 어드레스 비트들과 상기 멀티플렉싱된 어드레스 비트들을 연결(concatenate)시키는 단계; 및

상기 연결된 어드레스를 사용하여 상기 BHT 내의 라인들을 인덱싱하는 단계를 포함하는,

방법.

청구항 14

제13항에 있어서,

BHT 내의 라인들을 인덱싱하기 전에 전체(global) 분기 이력 값을 사용하여 상기 연결된 어드레스를 해싱처리(hashing)하는 단계를 추가로 포함하는,

방법.

청구항 15

제14항에 있어서,

상기 전체 분기 이력 값을 사용하여 상기 연결된 어드레스를 해싱처리(hashing)하는 단계는 상기 연결된 어드레스와 전체 분기 이력 값 간에 배타적-논리합(logical exclusive-or)을 수행하는 단계를 포함하는,

방법.

청구항 16

제13항에 있어서,

인덱싱된 라인에 대한 카운터 인덱스 어드레스는 상기 32 비트 명령 세트 실행 모드의 카운터 인덱스 어드레스 및 상기 16 비트 명령 세트 실행 모드의 카운터 인덱스 어드레스 간의 멀티플렉싱에 의해 형성되는,

방법.

청구항 17

제16항에 있어서,

상기 BHT는 각각 8개의 카운터 각각의 512개의 라인들을 포함하는,

방법.

청구항 18

제13항에 있어서,

인텍싱된 라인에 대한 카운터 인텍스 어드레스는 상기 32 비트 명령 세트 실행 모드의 어드레스 비트들 [4:2] 및 상기 16 비트 명령 세트 실행 모드의 어드레스 비트 [3:1] 간의 멀티플렉싱에 의해 형성되는,

방법.

명세서

기술분야

[0001] 본 발명은 일반적으로는 프로세서들의 분야에 관한 것이고, 더 구체적으로는 가변 길이 명령 세트 실행 모드들을 가지는 프로세서 내의 분기 이력 테이블에 대한 효율적인 구성에 관한 것이다.

배경기술

[0002] 프로세서들에 대한 통상적인 명령 세트 아키텍처들은 균일한 명령 길이를 가진다. 즉, 상기 명령 세트 내의 모든 명령들은 동일한 개수의 비트들(예를 들어, 16 또는 32)를 포함한다. 가변 길이 명령 실행 모드들을 가지는 프로세서들 - 상기 프로세서는 상이한 비트 길이들을 가지는 명령들을 실행할 수 있음 - 이 당해 기술분야에 공지된다. 예를 들어, ARM 아키텍처의 최신 버전들은 32비트 명령 세트 실행 모드(ARM mode)로 실행되는 종래의 32비트 ARM 명령들 뿐만 아니라 16비트 명령 세트 실행 모드로 실행되는 16비트 명령들을 포함한다.

[0003] 가변 길이 명령들을 실행하는 프로세서들이 가지는 한 가지 문제점은 명령들이 균일한 메모리 영역들(boundary)에 있지 않다는 것이다. 따라서, 명령들(또는, 명령들과 연관된 종속적인 구성물들)을 통해 증분하거나 혹은 랜덤하게 어드레스지정(address)하는 동작들 또는 회로들은 균일 증분 혹은 어드레스 지정(addressing) 방식을 사용할 수 없다. 오히려, 이들은 현재 실행중인 명령의 길이들에 기초하여 상기 어드레스 지정 방식 즉, 현재 명령 세트 실행 모드를 변경시켜야 한다.

[0004] 대부분의 최신 프로세서들은 파이프라인 아키텍처를 사용하는데, 각각이 다수의 실행 단계들을 가지는 순차적 명령들이 실행시 오버랩된다. 최대 성능을 위해, 상기 명령들은 파이프라인을 통해 연속적으로 플로우(flow)해야 한다. 명령들이 상기 파이프라인 내에서 중지하게 하는 임의의 상황은 성능에 악영향을 미친다. 만약 명령들이 상기 파이프라인으로부터 제거되고(flush) 후속적으로 재-페치(re-fetch)되어야 하는 경우, 성능 및 전력 소모 모두에 악영향을 준다.

[0005] 실질적으로, 모든 실제 프로그램들이 조건부 분기 명령들을 포함하며, 이들의 실제 분기 동작들은 상기 명령이 파이프라인 내에서 심도있게 평가된 후에야 비로소 알려진다. 상기 분기 명령의 실제 평가를 대기함으로 인해 초래되는 중지를 회피하기 위해, 대부분의 최신 프로세서들은 분기 예측의 몇몇 형태를 사용하며, 이에 의해 조건부 분기 명령들의 분기 동작은 파이프라인에서 조기에 예측된다. 예측된 분기 평가에 기초하여, 상기 프로세서는 예측된 어드레스 - 분기 타겟 어드레스(상기 분기가 취해지는 것으로 예측되는 경우) 또는 분기 명령 이후 다음의 순차적 어드레스(예를 들어, 상기 분기가 취해지지 않는 것으로 예측되는 경우) - 로부터 명령들을 추론적으로 패치하고 실행한다. 실제 분기 동작이 결정될 때, 분기가 잘못 예측된 경우, 추론적으로 패치된 명령들이 상기 파이프라인으로부터 제거되어야 하며, 올바른 다음 어드레스로부터 새로운 명령들이 패치되어야 한다. 잘못된 분기 예측에 응답하여 명령들을 추론적으로 패치하는 것은 프로세서 성능 및 전력 소모에 악영향을 미친다. 결과적으로, 분기 예측의 정확성이 중요한 설계 목적이다.

[0006] 분기 예측의 몇몇 방법들은 예측되는 분기 명령 및/또는 동일한 코드 내의 다른 분기 명령들의 분기 평가 이력에 기초한다. 실제 코드의 확장적인 분석은 최근 과거 분기 평가 패턴들이 미래 분기 명령들의 평가의 좋은 표시자일 수 있음을 나타낸다.

[0007] 분기 예측의 한가지 공지된 형태는 최근 분기 평가들을 저장하기 위해 분기 이력 테이블(BHT)을 사용하는 것이다. 일 예로서, 상기 BHT는 다수의 포화 카운터들을 포함할 수 있으며, 이들의 MSB들은 두 가지 모드(bimodal)의 분기 예측자로서의 역할을 한다. 예를 들어, 각각의 카운터는, 각각에 가중치 예측 값이 할당된 다음과 같은 4개의 상태들 중 하나를 가정하는 2 비트 카운터를 포함할 수 있다.

- [0008] 11 - 강하게 예측되고 취해짐
- [0009] 10 - 약하게 예측되고 취해짐
- [0010] 01 - 약하게 예측되고 취해지지 않음
- [0011] 00 - 강하게 예측되고 취해지지 않음
- [0012] 카운터는 해당 분기 명령이 "취해지는" 것으로 평가될 때마다 증분되고, 상기 명령이 "취해지지 않는" 것으로 평가될 때마다 감소된다. 상기 카운터의 MSB는 두 가지 모드 분기 예측자이며, 이것은 예측의 강도 또는 가중치에 상관없이, 분기가 취해지거나 취해지지 않는 것을 예측할 것이다. 포화 카운터는, 한 방향으로의 단일 분기 평가가 나머지 다른 방향으로 포화된 카운터의 예측을 바꾸지 않을 것이기에, 빈번하지 않은 분기 평가의 예측 에러를 감소시킨다.
- [0013] "로컬(local)" BHT의 경우, 상기 분기 명령의 분기 평가가 예측되고 있는 각각의 분기 명령은 단일 BHT 카운터와 연관된다. 따라서, 상기 BHT는 분기 명령 어드레스 (BIA)의 일부분을 사용하여 인덱싱된다. 많은 최신 프로세서들은 단일 패치 동작에서, 블록들 또는 패치 그룹들로 다수의 명령들을 패치한다. 이 경우, 상기 블록 또는 패치 그룹과 연관된 어드레스는, 본 명세서에서 사용된 바와 같이, BIA로 간주된다. "글로벌(global)" BHT의 경우, 최근의 전체 분기 평가 이력은 상기 BHT 카운터들을 인덱싱하기에 앞서 상기 BIA와 연관되거나 (gselect) 또는 상기 BIA를 사용하여 해싱처리된다.
- [0014] 상이한 명령 길이들을 가지는 명령 세트는 BHT들의 설계를 복잡하게 한다. 특히, 상기 BHT는 이상적으로는, 각각의 카운터가 분기 명령과 연관되기 때문에, 각각의 다른 명령 세트 실행 모드에서 다르게 인덱싱되고, 상기 명령들은 상이한 명령 세트 실행 모드들에서 서로 다른 메모리 영역에 위치한다. 한 가지 공지된 솔루션 (solution)은 최대 명령 길이에 기초하여 상기 BHT 사이즈를 간단하게 하지만 최소 명령 길이에 기초하여 이를 어드레스지정(address)하는 것이다. 이러한 솔루션은 상기 테이블의 큰 부분들을 비워 두거나, 더 긴 분기 명령들과 연관된 중복 엔트리들로 채운다. 또다른 공지된 솔루션은 각각 다른 명령 세트 실행 모드에서 상기 명령 어드레스의 상이한 부분을 사용하여 효율적으로 BHT 인덱스 어드레스들을 멀티플렉싱하는 것이다. 상기 솔루션은 다수의 멀티플렉서들을 부가하며, 이는 실리콘 영역 및 전력 소모를 증가시킨다. 그러나 더 중요하게는 그것이 중요 경로에 지연을 부가함으로써 사이클 시간을 증가시키고 프로세서 성능에 악영향을 미치는 것이다.

발명의 상세한 설명

- [0015] 하나 이상의 실시예들에 따라, BHT 라인 및 카운터 인덱스 어드레스들은 BHT 내의 모든 카운터들에 액세스하고 현재 명령 세트 실행 모드에 기초하여 멀티플렉싱된 어드레스 비트수를 감소하도록 형성된다. 또한, 일 실시예에서, BHT 라인 내의 카운터들은 상기 BHT의 절반이 하나의 명령 세트 실행 모드에서 각각의 액세스에 대해 파워 다운 될 수 있는 방식으로 정렬된다.
- [0016] 일 실시예는 적어도 제 1 최소 명령 길이를 가지는 제 1 명령 세트 실행 모드 및 더 작은 제 2 최소 명령 길이를 가지는 제 2 명령 실행 모드에서 명령들을 실행하는 가변 길이 명령 세트 프로세서에서 분기 이력 테이블 (branch history table ; BHT) 내의 모든 라인을 인덱싱하기 위한 어드레스를 형성하는 방법으로서, 이에 의해 각각의 명령 세트 실행 모드는 상이한 네이티브 (native) 인덱스 어드레스를 가지는 방법에 관한 것이다. 상기 제 1 및 제 2 명령 세트 실행 모드들의 공통 네이티브 라인 인덱스 어드레스 비트들이 식별된다. 상기 제 1 명령 세트 실행 모드에 대해 독립적인 최상위 네이티브 라인 인덱스 어드레스 비트들과 상기 제 2 명령 세트 실행 모드에 대해 독립적인 최하위 네이티브 라인 인덱스 어드레스 비트들이 명령 세트 실행 모드 표시자에 기초하여 멀티플렉싱된다. 상기 공통 어드레스 비트들과 멀티플렉싱된 어드레스 비트들이 연결되고, BHT 내의 라인들은 연결된 어드레스를 사용하여 인덱싱된다.
- [0017] 또다른 실시예는 적어도 제 1 최소 명령 길이를 가지는 제 1 명령 세트 실행 모드 및 더 작은 제 2 최소 명령 길이를 가지는 제 2 명령 세트 실행 모드에서 명령들을 실행하는 가변 길이 명령 세트 프로세서에서 분기 이력 테이블(BHT) 내의 모든 카운터를 액세스하기 위한 라인 및 카운터 인덱스 어드레스들을 형성하는 방법에 관한 것이다. 상기 BHT 내의 라인들은 상기 제 2 명령 세트 실행 모드의 네이티브 라인 인덱스 어드레스를 사용하여 인덱싱된다. 상기 제 1 명령 세트 실행 모드에 독립적인 최상위 네이티브 라인 인덱스 어드레스 비트 및 상기 제 2 명령 세트 실행 모드에 독립적인 최하위 네이티브 카운터 인덱스 어드레스 비트는 명령 세트 실행 모드 표시자에 기초하여 멀티플렉싱된다. 각각의 BHT 내의 카운터들은, 멀티플렉싱된 비트들과 연결된, 제 2 명령 세트 실행 모드의 최하위 네이티브 카운터 인덱스 어드레스 비트들을 제외한 모두를 사용하여 인덱싱한다.

[0018] 또다른 실시예는 적어도 제 1 최소 명령 길이를 가지는 제 1 명령 세트 실행 모드 및 더 작은 제 2 최소 명령 길이를 가지는 제 2 명령 세트 실행 모드에서 명령들을 실행하는 가변 길이 명령 세트 프로세서에서 다수의 카운터들을 포함하는 분기 이력 테이블(BHT)을 동작시키는 방법에 관한 것이다. 전체(full) BHT는 상기 제 2 명령 세트 실행 모드에서 BHT 액세스들 동안 파워링된다. 상기 BHT의 오직 절반만이 상기 제 1 명령 세트 실행 모드에서 BHT 액세스들 동안 파워링된다.

[0019] 또다른 실시예는 프로세서에 관한 것이다. 상기 프로세서는 패치(fetch) 스테이지를 가지며, 적어도 제 1 최소 명령 길이를 가지는 제 1 명령 세트 실행 모드 및 더 작은 제 2 최소 명령 길이를 가지는 제 2 명령 세트 실행 모드에서 명령들을 실행하도록 동작하는 명령 실행 파이프라인을 포함한다. 상기 프로세서는 또한, 상기 제 1 명령 세트 실행 모드에서, 분기 이력 테이블(BHT)의 오직 절반만이 파워링되도록, 상기 패치 파이프라인 스테이지에서 명령 어드레스들에 의해 정렬되고 인덱싱된 다수의 카운터들을 포함하는 BHT를 포함한다.

실시예

[0024] 도 1은 프로세서(10)의 기능 블록도를 도시한다. 상기 프로세서(10)는 명령 유닛(12) 및 하나 이상의 실행 유닛들(14)을 포함한다. 상기 명령 유닛(12)은 명령 흐름의 중앙 집중 제어를 상기 실행 유닛(14)에 제공한다. 상기 명령 유닛(12)은, 명령측 변환 색인 버퍼(ITLB)(18)에 의해 관리되는 메모리 어드레스 변환 및 허가(permission)들을 사용하여, 명령 캐시(16)로부터 명령들을 패치한다.

[0025] 실행 유닛들(14)은 명령 유닛(12)에 의해 디스패치되는(dispatched) 명령들을 실행한다. 상기 실행 유닛들(14)은 범용 레지스터들(GPR)(20)을 판독 및 기록하고, 데이터 캐시(22)로부터 데이터에 액세스하며, 메모리 어드레스 변환 및 허가(permission)들은 주 변환 색인 버퍼(TLB)(24)에 의해 관리된다. 다양한 실시예들에서, 상기 ITLB(18)는 TLB(24)의 일부분의 카피를 포함할 수 있다. 대안적으로, 상기 ITLB(18) 및 TLB(24)는 일체화될 수 있다. 유사하게, 프로세서(10)의 다양한 실시예들에서, 상기 명령 캐시(16) 및 데이터 캐시(22)는 일체화되거나 통합될 수 있다. 명령 캐시(16) 및/또는 데이터 캐시(22) 내의 미스(miss)들은 제 2 레벨 또는 도 1에서 통합된 명령 및 데이터 캐시(26)로서 도시된 L2 캐시(26)로의 액세스를 야기하지만, 다른 실시예들은 개별적인 L2 캐시들을 포함할 수 있다. L2 캐시(26)에서의 미스들은 메모리 인터페이스(30)의 제어하에, 주(오프-칩) 메모리(28)로의 액세스를 야기한다.

[0026] 명령 유닛(12)은 프로세서(10) 파이프라인의 패치(32) 및 디코드(36) 스테이지들을 포함한다. 패치 스테이지(32)는 명령들을 검색하기 위해 명령 캐시(16) 액세스들을 수행하는데, 상기 액세스들은 L2 캐시(26) 액세스 및/또는 원하는 명령들이 명령 캐시(16) 또는 L2 캐시(26)에 각각 상주하지 않는 경우 메모리(28) 액세스를 포함할 수 있다. 상기 디코드 스테이지(36)는 검색된 명령들을 디코딩한다. 명령 유닛(12)은 추가적으로, 상기 디코드 스테이지(36)에 의해 디코딩된 명령들을 저장하기 위한 명령 큐(38), 및 적절한 실행 유닛들(14)로 큐잉된 명령들을 디스패치하기 위한 명령 디스패치 유닛(40)을 포함한다.

[0027] 분기 예측 유닛(BPU)(42)은 조건부 분기 명령들의 실행 동작을 예측한다. 패치 스테이지(32)에서 명령 어드레스들은 명령 캐시(16)로부터의 명령 패치들과 병렬로 분기 타겟 어드레스 캐시(BTAC) (44) 및 분기 이력 테이블(BHT)(46)을 액세스한다. BTAC(44)에서의 히트는 이전에 평가되어 취해진 분기 명령을 표시하고, 상기 BTAC(44)는 상기 분기 명령의 마지막 실행의 분기 타겟 어드레스(BTA)를 제공한다. 상기 BHT(46)는 분석된(resolved) 분기 명령들에 대응하는 분기 예측 기록들을 유지하며, 상기 기록들은 알려진 분기들이 이전에 평가되어 취해졌는지 취해지지 않았는지의 여부를 표시한다. 상기 BHT(46)는, 예를 들어, 분기 명령의 이전 평가들에 기초하여, 분기가 취해질 것인지 취해지지 않을 것인지에 대한 약 - 강(weak to strong) 예측들을 제공하는 포화 카운터들을 포함할 수 있다. 상기 BPU(42)는 분기 예측들을 추정하기 위해 BTAC(44)로부터의 히트/미스 정보 및 상기 BHT(46)로부터의 분기 이력 정보를 평가한다. 상기 BPU(42)는 이전 분기 예측들의 정확도를 트래킹하는 전체 분기 이력 레지스터, 테이블 또는 회로 등과 같은, 분기 예측 정확도를 증가시키기 위한 부가 회로(미도시)를 포함할 수 있다.

[0028] 도 2는 패치 스테이지(32) 및 명령 유닛(12)의 분기 예측 회로를 더 상세하게 도시한 기능 블록도이다. 도 2의 점선은, 반드시 직접 연결일 필요는 없는, 기능상의 액세스 관계들을 도시함을 주목하라. 패치 스테이지(32)는 다양한 소스들로부터 명령 어드레스들을 선택하는 캐시 액세스 스티어링 로직(48)을 포함한다. 사이클 당 하나의 명령 어드레스가 명령 패치 파이프라인에서 시작되며, 본 실시예에서 상기 파이프라인은 3개 스테이지들, 패치 1 스테이지(50), 패치 2 스테이지 (52), 및 패치 3 스테이지(54)를 포함한다.

[0029] 캐시 액세스 스티어링 로직(48)은 패치 파이프라인에 론칭(launch)하기 위해 다양한 소스들로부터 명령 어드레

스들을 선택한다. 하나의 명령 어드레스 소스는 (하나의 명령, 또는 다수의 명령들을 포함하는 블록 또는 패치 그룹과 연결될 수 있는) 다음 순차적 명령 어드레스를 생성하기 위해 패치 1 파이프라인 스테이지(50)의 출력에 대해 동작하는 증분기(56)의 출력이다. BPU(42)가 취해진 것으로 평가되는 것으로 상기 분기를 예측하는 경우, 또다른 명령 어드레스 소스는, 분기 명령의 이전 실행의 분기 타겟 어드레스를 제공하는 BTAC(44)이다. 다른 명령 어드레스 소스들은 예외 관리자(exception handlers), 인터럽트 벡터 어드레스들 등을 포함한다.

[0030] 패치 1 스테이지(50) 및 패치 2 스테이지(52)는 명령 캐시(16) 및 BTAC(44)로의 동시적이며 병렬적인 2-스테이지 액세스들을 수행하며 동시에 제 1 사이클 내에서 BHT(46)에 액세스한다. 특히, 패치 1 스테이지(50) 내의 명령 어드레스는, (명령 캐시(16) 내의 히트 또는 미스를 통해) 상기 어드레스와 연결된 명령들이 상기 명령 캐시(16)에 상주하는지의 여부; 알려진 상기 분기 명령이 (명령 캐시(16) 내의 히트 또는 미스를 통해) 상기 명령 어드레스와 연결되는지의 여부를 확인하고, (BHT(46) 내의 카운터 값을 통해) 분기 예측을 획득하기 위해 제 1 캐시 액세스 사이클 동안 명령 캐시(16), BTAC(44), 및 BHT(46)를 액세스한다. 후속하는 제 2 캐시 액세스 사이클에서, 명령 어드레스는 패치 2 스테이지(52)로 이동하고, 명령들은 상기 명령 캐시(16)로부터 사용 가능하고 그리고/또는 명령 어드레스가 개별 캐시(16,44)에 히트하는 경우, 분기 타겟 어드레스(BTA)는 BTAC (44)로부터 사용가능하다.

[0031] 만약 명령 어드레스가 명령 캐시(16)에서 미스된다면, L2 캐시 (26) 액세스를 시작하는 패치 3 스테이지(54)로 진행한다. 당업자는 패치 파이프라인이 예를 들어, 명령 캐시(16) 및 BTAC(44)의 액세스 타이밍에 따라, 도 2에 도시된 실시예보다 더 많거나 더 적은 레지스터 스테이지들을 포함함을 쉽게 인지할 것이다.

[0032] 여기에서 사용된 바와 같이, 용어 "기능 명령 어드레스"는 최하위 단에서 사용되지 않는 비트들을 무시하고, 명령들을 액세스하는데 실제로 사용되는 바이트 메모리 어드레스의 비트들을 지칭한다. 기능 명령 어드레스들은 상이한 명령 길이들에 대해 상이한 바이트 메모리 어드레스를 사용하여 형성된다. 예를 들어, 4096개 16비트 명령들은 바이트 메모리 어드레스 비트들[12:1]에 의해 어드레스 지정된다. 비트 0은 어떠한 개별 바이트도 어드레스 지정되지 않았으므로 16비트 명령들에 대한 기능 명령 어드레스의 일부분이 아니다. 유사하게, 4096개의 32 비트 명령들은 바이트 메모리 어드레스 비트 [13:2]에 의해 어드레스 지정된다. 상기 바이트 메모리 어드레스 비트들[1:0] 중 어느 것도, 바이트 또는 하프워드 둘 다 어드레스 지정되지 않았으므로, 32비트 명령들에 대한 기능 명령 어드레스의 일부분이 아니다.

[0033] BHT(46)의 일 실시예의 기능 블록도는 도 3에 도시된다. BHT(46)는, 라인당 8개 카운터들이 있는, 512개 라인들로서 정렬되는 4096개의 2비트 카운터들을 포함한다. 따라서, 라인 인덱스 어드레스는 9 비트를 포함하고, 카운터 인덱스 어드레스는 3비트를 포함한다. 그러나, 서로 다른 최소 길이 명령들을 실행하는 다른 명령 세트 실행 모드들에서, 라인 및 카운터 어드레스 비트들은, 기능 명령 어드레스들에서의 차이들에 따라 다를 것이다. 특히, 네이티브 라인 인덱스 어드레스는 32비트 명령 세트 실행 모드에서 명령 어드레스 비트 [13:5]를 포함하고, 16비트 명령 세트 실행 모드에서 비트 [12:4]를 포함한다. 유사하게, 상기 네이티브 카운터 인덱스 어드레스는 32비트 명령 세트 실행 모드에서 명령 어드레스 비트 [4:2]를 포함하고, 16비트 명령 세트 실행 모드에서 비트 [3:1]를 포함한다.

[0034] 여기에 사용된 바와 같이, 특정 최소 명령 길이를 가지는 주어진 명령 세트 실행 모드에 대한 용어 "네이티브" 라인 및 인덱스 어드레스들은 모든 명령들이 동일한 길이인 경우 BHT(46)를 액세스하기 위해 사용된다. 특히, 네이티브 카운터 인덱스 어드레스는 라인마다 모든 2^n 개의 카운터들을 고유하게 인덱싱하는데 필요한 최하위 n 개 기능 명령 어드레스 비트들을 포함한다. 네이티브 라인 인덱스 어드레스는 상기 BHT(46) 내의 모든 2^m 개의 라인들을 고유하게 인덱싱하는데 필요한 기능 명령 어드레스의 그다음 m 초과 최상위 비트를 포함한다.

[0035] 선택기로서 명령 세트 실행 모드를 사용하는, 네이티브 라인과 카운터 인덱스 어드레스들 간의 멀티플렉싱은, 수용불가능한 전력 소모 및 액세스 지연을 초래하는 모든 $(m+n)$ 비트들(도 3의 실시예에서 12비트)의 멀티플렉싱을 요구한다.

[0036] 도 3에 도시된 실시예에서, 라인 인덱스 어드레스는 32비트 및 16비트 명령 세트 실행 모드 둘 다에 대해 공통인 네이티브 라인 인덱스 어드레스 비트들 - 즉, 비트 [12:5] - 을 취하고 이들 공통 비트들을 연결시킴으로써 형성되고, 상기 비트들은 대응하는 명령 세트 실행 모드일 때 각각의 개별 네이티브 라인 인덱스 어드레스들에 대해 독점적이다. 즉, (32 비트 명령 세트 실행 모드의 네이티브 라인 인덱스 어드레스에 대해 독점적인) 비트 13 및 (16비트 명령 세트 실행 모드의 네이티브 라인 인덱스 어드레스에 대해 독점적인) 비트 4는, 9 비트 라인 인덱스 어드레스를 형성하기 위해, 현재 명령 세트 실행 모드에 기초하여 함께 멀티플렉싱되고, 상기 공통 비트

[12:5]에 연결된다.

- [0037] 도 3에 도시된 바와 같이, 이러한 9비트 라인 인덱스 어드레스는, BHT(46)의 라인들을 인덱싱하기 전에, 예를 들어, 해시 함수 또는 다른 로직(58)에 의해, 전체 분기 이력 정보와 선택적으로 조합될 수 있다. 이러한 실시예에서, 전체(full) 네이티브 카운터 인덱스 어드레스들은, 현재 명령 세트 실행 모드에 기초하여 멀티플렉싱된다. 특히, 32비트 명령 세트 실행 모드에서의 비트 [4:2] 및 16비트 명령 세트 실행 모드에서의 비트 [3:1]는, 각각의 라인 내에서 상기 카운터들을 인덱싱한다. 이는, 전체 네이티브 라인 및 카운터 인덱스 어드레스 비트들이 인덱싱되는 경우, 12에 대조되어, BHT(46)에 대한 전체 4개 멀티플렉서들에 대해, 상기 라인 인덱스 어드레스에서의 한 비트 및 카운터 인덱스 어드레스에서의 3비트의 멀티플렉싱을 요구한다.
- [0038] 도 4는 BHT(46)의 또다른 실시예를 도시하며, 상기 BHT(46)에서, 오직 단일 비트 멀티플렉서만이 32비트 및 16비트 명령 세트 실행 모드들 모두에서 BHT(46) 내의 모든 카운터를 완전히 사용하기 위해 요구된다. 상기 라인 인덱스 어드레스는 더 작은 최소 명령 길이 - 이 경우, 16비트 명령 세트 실행 모드 - 를 가지는 명령 세트 실행 모드에 대한 네이티브 라인 인덱스 어드레스, 즉, 비트 [12:4]이다. 여기서 다시, 상기 라인 인덱스 어드레스는 로직(58)에서 전체 분기 이력에 의해 수정될 수 있다. 상기 카운터 인덱스 어드레스는 32비트 및 16비트 명령 세트 실행 모드 모두에 대해 공통인 네이티브 카운터 인덱스 어드레스 비트 - 즉, 비트 [3:2] - 를 취하고 이들 공통 비트에 연결시킴으로써 형성되며, 상기 비트는 각각의 연결된 네이티브 라인/카운터 인덱스 어드레스에 대해 독점적이다. 즉, (32비트 명령 세트 실행 모드의 네이티브 라인 인덱스 어드레스에 대해 독점적인) 비트 13 및 (16비트 명령 세트 실행 모드의 네이티브 인덱스 어드레스에 대해 독점적인) 비트 1은, 3비트 카운터 인덱스 어드레스를 형성하기 위해, 현재 명령 세트 실행 모드에 기초하여 함께 멀티플렉싱되고 상기 공통 비트 [3:2]에 연결된다.
- [0039] 추가적으로, 도 4의 실시예에서, BHT(46)은 두 개의 절반들로 분할되며, 상기 절반들 각각은 개별적으로 파워링된다. 추가적으로, 카운터들은, 짝수 카운터 어드레스들(즉, 0으로 끝남)을 가지는 카운터들이 상기 라인의 절반에 있고, 홀수 카운터 인덱스 어드레스를 가지는 카운터들(즉, 1로 끝남)이 상기 라인의 나머지 절반에 있도록, 각각의 라인에 정렬된다. 32비트 명령 세트 실행 모드에서, 상기 카운터 인덱스 어드레스의 최하위 비트가 비트 13이므로, 상기 모드에 있을 경우, 상기 BHT(46)의 오직 절반만이 파워업될 필요가 있다 - 상기 BHT(46)의 활성인 절반은 비트 13에 의해 선택된다. 이는 상기 프로세서에서 32비트 명령 세트 실행 모드로 동작할 경우 상당한 전력 절감을 나타낸다.
- [0040] 도 2에 도시된 프로세서(10)의 패치 스테이지(32)의 실시예에서, 패치 n 스테이지 레지스터들(50, 52, 54) 각각은 4개의 하프워드(8바이트)를 유지한다. 따라서, 각각의 패치 n 스테이지 레지스터 (50, 52, 54)는 32비트 명령 세트 실행 모드에서 최대 2개 명령을, 그리고 16비트 명령 세트 실행 모드에서 최대 4개 명령들을 유지한다. 따라서, BHT(46)은 32비트 모드에서 최대 2개 카운터 값들, 그리고 16비트 모드에서 최대 4개의 카운터 값들을 검색할 수 있다. 모든 8개의 카운터 값들이 동일한 사이클 내에서 액세스되는 경우는 없다.
- [0041] 도 4의 BHT(46) 실시예에서, 상기 카운터들은 특정 BHT(46) 액세스에 대해 상호 독점적인 카운터 값들 - 즉, 둘 다 동시에 액세스될 수 없는 카운터들 - 이 직접 인접하도록 각각의 라인에 추가로 정렬된다. 이는, 도 4에서 출력 멀티플렉서들의 제 1 행에 의해 도시된 바와 같이, BHT(46)에서 멀티플렉싱되도록 하여 BHT(46) 출력들에서의 배선 혼잡을 감소시킨다. 카운터들의 정렬에 대한 이러한 제약은 카운터들을 병치시킴으로써 도 4의 실시예에서 달성되며, 상기 카운터들의 카운터 인덱스 어드레스들은 그들의 최상위 비트만 상이하다. 예를 들어, 오직 비트 3에서만 상이한 카운터 인덱스 어드레스들 0x1 (001) 및 0x5 (101)은 인접한다. 출력 멀티플렉서들의 제 1 레벨 각각에 대한 선택 입력은 연결된 명령 어드레스의 비트 3이다.
- [0042] 32비트 명령 세트 실행 모드에서, 최대 2개 카운터 값들은 단일 BHT(46) 액세스에서 검색될 수 있다. 도 4에 도시된 멀티플렉서들의 제 2 레벨은 BHT(46)의 홀수 측에 대한 BHT(46)의 짝수 측의 출력들을 스티어링한다. 여기서 또한, 이는 BHT(46) 출력에서의 배선 혼잡을 감소시키고 32비트 명령 실행 모드에서 카운터 값들의 검색을 간소화한다.
- [0043] 본 발명이 32비트 및 16비트 명령 세트 실행 모드들에 대해서 설명되었지만, 본 발명은 이러한 구현에 제한되지 않으며, 어떠한 가변 길이 명령 세트 프로세서에도 유리하게 응용될 수 있다. 추가적으로, 카운터 값들의 특정 배열 및 멀티플렉싱은 임의의 특정 구현에 대해 본 발명을 최적화하도록 당업자에 의해 변경될 수 있다. 일반적으로, 본 발명은 본 발명의 특정 특징들, 양상들 및 실시예들에 대해 여기에서 기술되었지만, 많은 변경들, 수정들 및 다른 실시예들이 본 발명의 넓은 범위 내에서 가능하고, 따라서, 모든 변경들, 수정들 및 실시예들이 본 발명의 범위 내에 있는 것으로 간주되어야 함이 명백하다. 따라서, 본 발명의 실시예들은 모든 양상들에서

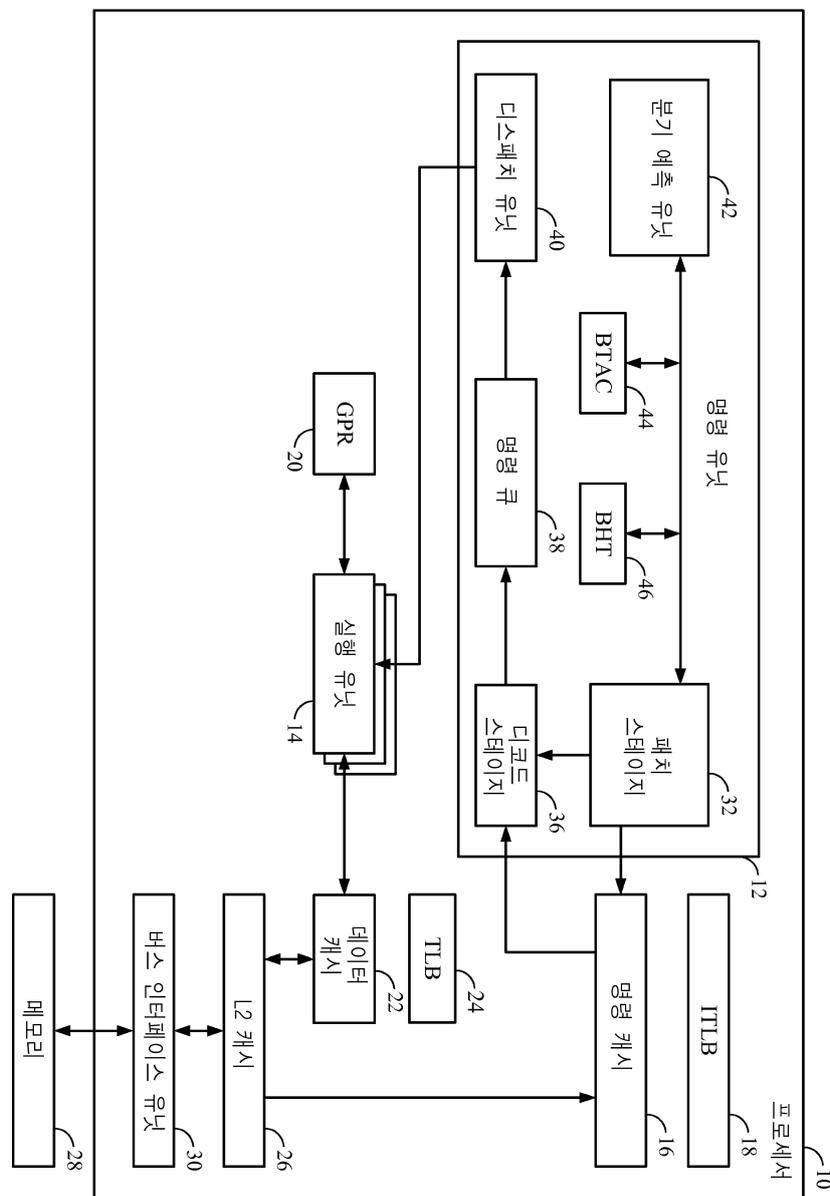
제한적이 아니라 예시적인 것으로서 해석되어야 하고, 첨부되는 청구항들의 의미 및 등가 범위내에서의 모든 변경들이 여기에 포함되는 것으로 의도된다.

도면의 간단한 설명

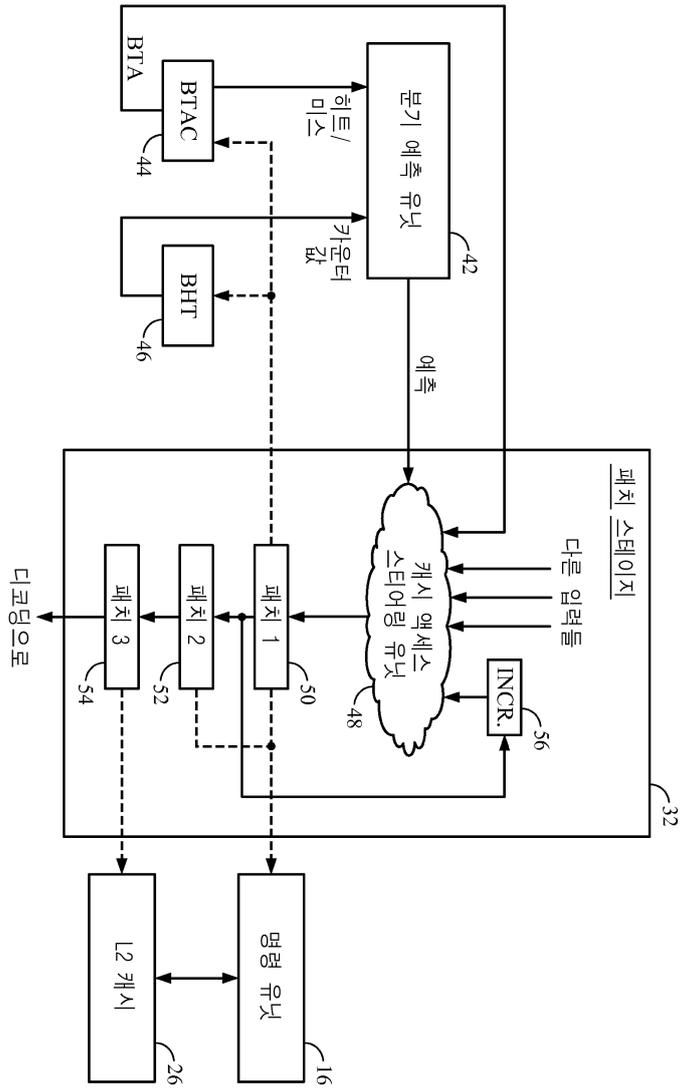
- [0020] 도 1은 프로세서의 기능 블록도이다.
- [0021] 도 2는 프로세서의 패치 스테이지에 대한 기능 블록도이다.
- [0022] 도 3은 일 실시예에 따른 BHT의 기능 블록도이다.
- [0023] 도 4는, BHT의 오직 절반만이 하나의 프로세서 명령 세트 실행 모드 동안 과워링되는, 다른 실시예에 따른 기능 블록도이다.

도면

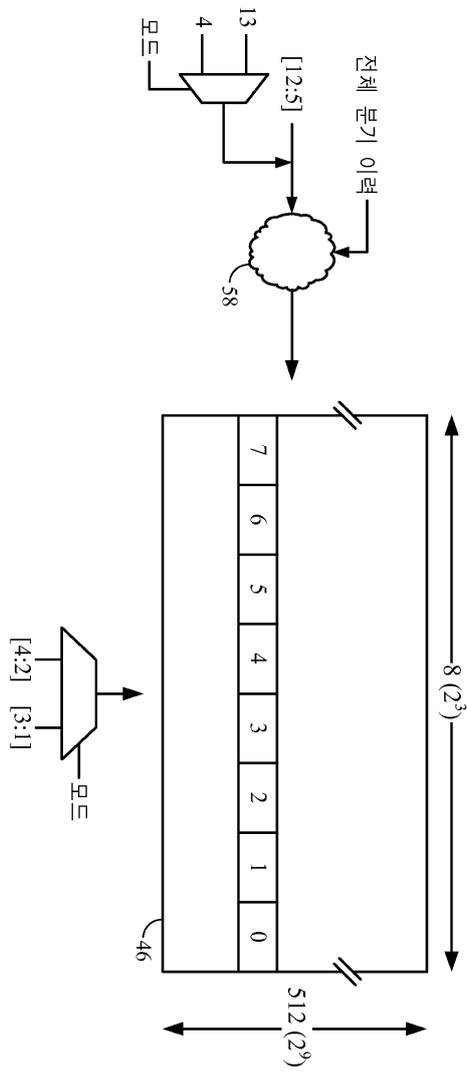
도면1



도면2



도면3



도면4

