



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2017년02월27일

(11) 등록번호 10-1710438

(24) 등록일자 2017년02월21일

(51) 국제특허분류(Int. Cl.)

G06F 12/08 (2016.01) G06F 9/345 (2006.01)

G06F 9/38 (2006.01)

(52) CPC특허분류

G06F 12/0864 (2013.01)

G06F 9/3455 (2013.01)

(21) 출원번호 10-2015-7017911

(22) 출원일자(국제) 2014년01월10일

심사청구일자 2016년10월28일

(85) 번역문제출일자 2015년07월03일

(65) 공개번호 10-2015-0106881

(43) 공개일자 2015년09월22일

(86) 국제출원번호 PCT/US2014/011051

(87) 국제공개번호 WO 2014/113288

국제공개일자 2014년07월24일

(30) 우선권주장

13/741,917 2013년01월15일 미국(US)

(56) 선행기술조사문현

US20020133672 A1

US20040030838 A1

US20060294510 A1

(73) 특허권자

웰컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

사쓰원, 피터 지.

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

벤쿠마한티, 수레쉬 케이.

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

코드레스쿠, 루시안

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(74) 대리인

특허법인 남앤드남

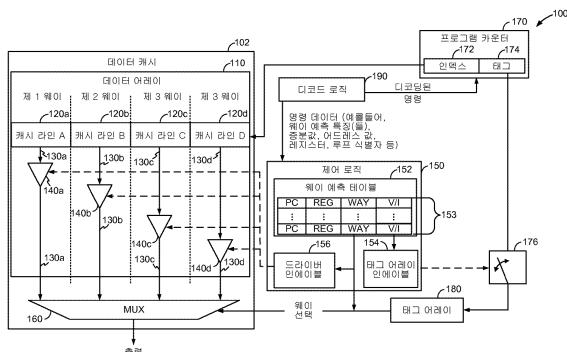
전체 청구항 수 : 총 29 항

심사관 : 임정목

(54) 발명의 명칭 데이터 캐시 웨이 예측

**(57) 요 약**

특정 실시예에서, 방법은 명령의 하나 이상의 웨이(way) 예측 특성들을 식별하는 단계를 포함한다. 방법은 또한 하나 이상의 웨이 예측 특성들의 식별에 기초하여, 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위하여 테이블을 선택적으로 판독하는 단계를 포함한다. 방법은 명령에 기초한 데이터 캐시의 다음 액세스가 웨이에 액세스할지에 대한 예측을 수행하는 단계를 더 포함한다.

**대 표 도**

(52) CPC특허분류

*G06F 9/3832* (2013.01)

*G06F 2212/6082* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

방법으로서,

상기 명령이 웨이(way) 예측 특성을 가지면, 상기 명령의 제 1 실행 동안:

상기 명령과 연관되고 그리고 상기 제 1 실행 동안 액세스되는 데이터 캐시의 하나 또는 그 초파의 캐시 라인들과 연관된 웨이를 식별하는 테이블의 엔트리(entry)를 식별하기 위하여 상기 테이블의 제 1 판독을 수행하는 단계;

상기 데이터 캐시의 제 2 액세스가 상기 웨이에 액세스할지 여부에 대한 예측을 수행하기 위해 산술 연산을 수행하는 단계 – 상기 제 2 액세스는 상기 명령의 제 2 실행에 대응하고, 상기 제 2 실행은 상기 제 1 실행에 후속하여 수행됨 –; 및

상기 예측에 응답하여, 상기 엔트리를 무효화하거나 또는 삭제할지 여부를 결정하는 단계; 및

상기 명령이 상기 웨이 예측 특성을 갖지 않으면, 상기 제 1 판독을 수행하지 않고 상기 명령을 실행하는 단계를 포함하는, 방법.

#### 청구항 2

제 1 항에 있어서,

상기 웨이 예측 특성은 상기 명령의 어드레싱 모드, 상기 명령의 명령 타입, 상기 명령이 루프에 포함되는지 여부에 대한 표시, 또는 이들의 임의의 조합을 포함하는, 방법.

#### 청구항 3

제 2 항에 있어서,

상기 명령의 어드레싱 모드가 자동 증분 어드레싱 모드인지 또는 베이스 플러스 오프셋 어드레싱 모드(base plus offset addressing mode)인지 결정하는 단계를 더 포함하는, 방법.

#### 청구항 4

제 3 항에 있어서,

상기 명령의 어드레싱 모드가 상기 자동 증분 어드레싱 모드를 포함한다는 결정에 응답하여, 상기 데이터 캐시의 특정 웨이를 식별하도록 상기 엔트리의 예측된 웨이 필드를 세팅하는 단계를 더 포함하고, 상기 예측된 웨이 필드는 상기 엔트리의 생성시 세팅되는, 방법.

#### 청구항 5

제 2 항에 있어서,

상기 명령의 명령 타입이 로드 타입(load type)인지 또는 저장 타입(store type)인지 결정하는 단계를 더 포함하는, 방법.

#### 청구항 6

제 2 항에 있어서,

상기 테이블은 상기 명령이 특정 루프에 포함된다는 표시에 응답하여 선택적으로 판독되는, 방법.

#### 청구항 7

제 1 항에 있어서,

특정 명령의 어드레싱 모드가 베이스 플러스 오프셋 어드레싱 모드를 포함한다는 결정에 응답하여 상기 데이터 캐시의 특정 웨이를 식별하도록 상기 테이블의 특정 엔트리의 예측된 웨이 필드를 세팅하는 단계를 더 포함하고, 상기 특정 엔트리는 상기 특정 명령의 특정 실행과 관련하여 생성되고, 그리고 상기 예측된 웨이 필드는 상기 특정 명령의 후속 실행에 기초하여 세팅되는, 방법.

### 청구항 8

제 1 항에 있어서,

상기 테이블이 상기 엔트리를 포함하는지 여부를 결정하는 단계;

웨이 예측을 제공하기 위하여 상기 엔트리가 유효한지 여부를 결정하는 단계; 및

상기 엔트리가 유효 예측 웨이를 표시한다는 결정에 응답하여, 상기 엔트리로부터 상기 예측된 웨이를 리트리브(retrieve)하고 그리고 상기 데이터 캐시의 상기 예측된 웨이를 선택적으로 구동하는 단계를 더 포함하는, 방법.

### 청구항 9

제 1 항에 있어서,

레지스터 위치의 데이터를 수정한 특정 명령을 식별하는 단계;

상기 테이블의 특정 엔트리가 상기 레지스터 위치에 대응하는 레지스터 식별자를 포함하는지 여부를 결정하는 단계;

상기 특정 엔트리가 상기 특정 명령에 대응하는지 여부를 결정하는 단계; 및

상기 특정 엔트리가 상기 특정 명령에 대응하지 않을 때 상기 특정 엔트리를 제거하거나 또는 무효화하는 단계를 더 포함하는, 방법.

### 청구항 10

프로세서로서,

명령을 디코딩하도록 구성된 디코드 로직; 및

상기 디코드 로직에 커플링된 제어 로직을 포함하고,

상기 제어 로직은,

명령이 웨이 예측 특성을 가지면, 상기 명령의 제 1 실행 동안:

상기 명령과 연관되고 그리고 상기 제 1 실행 동안 액세스되는 데이터 캐시의 하나 또는 그 초과의 캐시 라인들과 연관된 웨이를 식별하는 테이블의 엔트리를 식별하기 위하여 상기 테이블의 제 1 판독을 수행하고;

상기 데이터 캐시의 제 2 액세스가 상기 웨이에 액세스할지 여부에 대한 예측을 수행하기 위해 산술 연산을 수행하고 – 상기 제 2 액세스는 상기 명령의 제 2 실행에 대응하고, 상기 제 2 실행은 상기 제 1 실행에 후속하여 수행됨 –; 그리고

상기 예측에 응답하여, 상기 엔트리를 무효화하거나 또는 삭제할지 여부를 결정하고; 그리고

상기 명령이 상기 웨이 예측 특성을 갖지 않으면, 상기 제 1 판독을 수행하지 않고 상기 명령을 실행하도록

구성되는, 프로세서.

### 청구항 11

제 10 항에 있어서,

복수의 라인 드라이버들 – 상기 복수의 라인 드라이버들 중 적어도 하나의 라인 드라이버는 상기 예측에 기초하여 상기 제 2 액세스 동안 선택적으로 인에이블되거나 또는 디스에이블됨 –;

상기 명령에 기초하여 태그 룩업(tag lookup) 동작을 수행하도록 구성된 태그 어레이;

상기 복수의 라인 드라이버들에 응답하고 그리고 상기 태그 어레이 또는 상기 제어 로직으로부터 수신되는 웨이 선택 신호에 응답하는 멀티플렉서; 및

상기 태그 룩업 동작을 선택적으로 인에이블하거나 또는 디스에이블하도록 구성된 스위치를 더 포함하는, 프로세서.

### 청구항 12

제 11 항에 있어서,

상기 명령은 충분값 및 어드레스와 연관되고, 그리고

상기 예측을 수행하기 위한 상기 산술 연산의 수행은,

상기 어드레스에 상기 충분값을 가산함으로써 상기 명령의 충분된 어드레스를 결정하는 것; 및

상기 충분된 어드레스가 상기 어드레스와 동일한, 상기 데이터 캐시의 캐시 라인에 로케이팅되는지 여부를 결정하는 것을 포함하는, 프로세서.

### 청구항 13

제 12 항에 있어서,

상기 제어 로직은,

상기 충분된 어드레스가 상기 동일한 캐시 라인에 로케이팅된다는 결정에 기초하여 상기 제 2 액세스 동안 상기 태그 룩업 동작을 선택적으로 디스에이블하고; 그리고

상기 태그 룩업 동작이 디스에이블될 때 상기 멀티플렉서에 웨이 선택 신호를 제공하도록 추가로 구성되는, 프로세서.

### 청구항 14

제 12 항에 있어서,

상기 제어 로직은, 상기 충분된 어드레스가 상기 어드레스와 연관된 캐시 라인과 상이한 캐시 라인과 연관된다는 결정에 기초하여 상기 테이블의 상기 엔트리가 무효하다는 것을 표시하거나 또는 상기 엔트리를 제거하도록 추가로 구성되는, 프로세서.

### 청구항 15

장치로서,

명령을 디코딩하기 위한 수단; 및

테이블을 선택적으로 판독하기 위한 수단을 포함하고,

상기 선택적으로 판독하기 위한 수단은 상기 디코딩하기 위한 수단에 커플링되고, 그리고 상기 선택적으로 판독하기 위한 수단은,

명령이 웨이 예측 특성을 가지면, 상기 명령의 제 1 실행 동안:

상기 명령과 연관되고 그리고 상기 제 1 실행 동안 액세스되는 데이터 캐시의 하나 또는 그 초과의 캐시 라인들과 연관된 웨이를 식별하는 상기 테이블의 엔트리를 식별하기 위하여 상기 테이블의 제 1 판독을 수행하고;

상기 데이터 캐시의 제 2 액세스가 상기 웨이에 액세스할지 여부에 대한 예측을 수행하기 위해 산술 연산을 수행하고 – 상기 제 2 액세스는 상기 명령의 제 2 실행에 대응하고, 상기 제 2 실행은 상기 제 1 실행에 후속하여 수행됨 –; 그리고

상기 예측에 응답하여, 상기 엔트리를 무효화하거나 또는 삭제할지 여부를 결정하고; 그리고

상기 명령이 상기 웨이 예측 특성을 갖지 않으면, 상기 제 1 판독을 수행하지 않고 상기 명령을 실행하

도록

구성되는, 장치.

### 청구항 16

제 15 항에 있어서,

상기 웨이에 기초하여 데이터 캐시 라인을 선택적으로 구동하기 위한 수단을 더 포함하는, 장치.

### 청구항 17

제 15 항에 있어서,

상기 테이블의 특정 엔트리는 하나 또는 그 초과의 비트들의 값에 기초하여 예측된 웨이를 표시하고, 그리고 상기 하나 또는 그 초과의 비트들은 복수의 드라이버들의 각각의 드라이버를 선택적으로 인에이블하거나 또는 디스에이블하기 위하여 상기 복수의 드라이버들에 대해 마스크(mask)로서 적용되는, 장치.

### 청구항 18

명령들을 포함하는 비-일시적 컴퓨터 관독가능 저장 매체로서,

상기 명령들은, 프로세서에 의해 실행될 때, 상기 프로세서로 하여금,

명령이 웨이 예측 특성을 가지면, 상기 명령의 제 1 실행 동안:

상기 명령과 연관되고 그리고 상기 제 1 실행 동안 액세스되는 데이터 캐시의 하나 또는 그 초과의 캐시 라인들과 연관된 웨이를 식별하는 테이블의 엔트리를 식별하기 위하여 상기 테이블의 제 1 관독을 수행하게 하고;

상기 데이터 캐시의 제 2 액세스가 상기 웨이에 액세스할지 여부에 대한 예측을 수행하기 위해 산술 연산을 수행하게 하고 – 상기 제 2 액세스는 상기 명령의 제 2 실행에 대응하고, 상기 제 2 실행은 상기 제 1 실행에 후속하여 수행됨 –; 그리고

상기 예측에 응답하여, 상기 엔트리를 무효화하거나 또는 삭제할지 여부를 결정하게 하고; 그리고

상기 명령이 상기 웨이 예측 특성을 갖지 않으면, 상기 제 1 관독을 수행하지 않고 상기 명령을 실행하게 하는, 비-일시적 컴퓨터 관독가능 저장 매체.

### 청구항 19

제 18 항에 있어서,

상기 예측을 수행하기 위한 상기 산술 연산의 수행은,

상기 명령과 연관된 증분값을 식별하는 것;

증분된 어드레스 값을 결정하기 위하여 상기 명령과 연관된 어드레스 값에 상기 증분값을 가산하는 것; 및

상기 증분된 어드레스 값이 상기 웨이와 연관되는지 여부를 결정하는 것을 포함하는, 비-일시적 컴퓨터 관독가능 저장 매체.

### 청구항 20

제 18 항에 있어서,

상기 프로세서에 의해 실행될 때, 상기 프로세서로 하여금, 상기 테이블에서 상기 엔트리를 패풀레이트(populate)하게 하는 명령들을 더 포함하고, 상기 테이블의 각각의 엔트리는 프로그램 카운터 식별자, 레지스터 식별자, 및 예측된 웨이 식별자를 포함하는, 비-일시적 컴퓨터 관독가능 저장 매체.

### 청구항 21

제 18 항에 있어서,

상기 엔트리는 상기 명령이 하나 또는 그 초과의 명령들의 루프에 포함된다는 결정이 수행된 이후에 생성되는, 비-일시적 컴퓨터 판독가능 저장 매체.

### 청구항 22

제 18 항에 있어서,

상기 프로세서에 의해 실행될 때, 상기 프로세서로 하여금,

복수의 레지스터 위치들을 모니터링하게 하고;

상기 복수의 레지스터 위치들의 특정 레지스터 위치의 데이터가 수정되는지 여부를 결정하게 하고; 그리고 특정 엔트리가 상기 특정 레지스터 위치에 대응하는 레지스터 식별자를 포함하는지 여부를 결정하기 위하여 상기 테이블을 판독하게 하는 명령들을 더 포함하는, 비-일시적 컴퓨터 판독가능 저장 매체.

### 청구항 23

방법으로서,

명령의 제 1 실행 동안 증분값을 식별하는 단계;

상기 명령에 기초하여 상기 제 1 실행 동안 액세스되는 데이터 캐시의 웨이를 식별하는 단계;

제 1 증분된 어드레스 값을 결정하기 위하여 상기 명령과 연관된 어드레스 값에 상기 증분값을 가산하는 단계;

상기 제 1 증분된 어드레스 값이 상기 데이터 캐시의 웨이에 로케이팅되는지 여부를 결정하는 단계; 및

상기 제 1 증분된 어드레스가 상기 데이터 캐시의 웨이에 로케이팅된다는 결정에 응답하여 그리고 웨이 예측 특성을 갖는 명령에 대하여 컨디셔닝되는 테이블에서 상기 명령에 대응하는 엔트리를 파풀레이트하는 단계를 포함하는, 방법.

### 청구항 24

제 23 항에 있어서,

상기 테이블의 각각의 엔트리는 프로그램 카운터 식별자, 레지스터 식별자, 예측된 웨이 식별자, 유효성 비트 (validity bit), 또는 이들의 조합을 포함하는, 방법.

### 청구항 25

제 23 항에 있어서,

상기 명령이 자동 증분 명령 또는 베이스 플러스 오프셋 명령을 포함하는지 결정하는 단계를 더 포함하는, 방법.

### 청구항 26

제 23 항에 있어서,

상기 엔트리는 상기 액세스되는 데이터 캐시의 웨이를 식별하고, 상기 엔트리는 상기 명령의 상기 제 1 실행과 관련하여 파풀레이트되고, 그리고

상기 방법은,

특정 명령이 특정 웨이 예측 특성을 갖는 명령이면, 상기 명령의 제 2 실행 동안, 상기 테이블로부터 상기 엔트리의 제 1 판독을 수행하고, 제 2 증분된 어드레스 값을 계산하고, 상기 엔트리의 웨이 필드로부터 상기 웨이를 리트리브하고, 그리고 리브리브된 웨이에 대응하는 상기 데이터 캐시의 드라이버를 선택적으로 인에이블하는 단계; 및

상기 특정 명령이 상기 특정 웨이 예측 특성을 갖지 않으면, 상기 제 1 판독을 수행하지 않고 상기 특정 명령을 실행하는 단계를 더 포함하는, 방법.

### 청구항 27

제 23 항에 있어서,

상기 명령의 후속 실행의 증분된 어드레스 값이 상기 웨이와 상이한 웨이에 로케이팅된다는 결정에 응답하여 상기 테이블을 업데이트하는 단계를 더 포함하고, 상기 명령의 상기 후속 실행은 상기 제 1 실행 이후에 이루어지는, 방법.

### 청구항 28

제 23 항에 있어서,

상기 증분된 어드레스가 상기 어드레스와 연관된 캐시 라인과 상이한 캐시 라인과 연관된다는 결정에 기초하여 상기 테이블의 상기 엔트리가 무효하다는 것을 표시하거나 또는 상기 엔트리를 제거하는 단계를 더 포함하는, 방법.

### 청구항 29

제 23 항에 있어서,

레지스터 위치의 데이터를 수정한 특정 명령을 식별하는 단계;

특정 엔트리가 상기 레지스터 위치에 대응하는 레지스터 식별자를 포함하는지 여부를 결정하기 위하여 상기 테이블을 판독하는 단계;

상기 특정 엔트리가 상기 특정 명령에 대응하는지 여부를 결정하는 단계; 및

상기 특정 엔트리가 상기 특정 명령에 대응하지 않을 때 상기 특정 엔트리를 제거하거나 또는 무효화하는 단계를 더 포함하는, 방법.

### 청구항 30

삭제

## 발명의 설명

### 기술 분야

[0001] 본 출원은 "DATA CACHE WAY PREDICTION"라는 명칭으로 2013년 1월 15일에 출원된 미국 정식 특허 출원 번호 제13/741,917호의 우선권을 주장하며, 이 출원의 내용들은 그 전체가 인용에 의해 통합된다.

[0002] 본 개시내용은 일반적으로 데이터 캐시 메모리 시스템에 관한 것이다.

### 배경 기술

[0003] 기술의 진보는 더 소형화되고 더 강력한 컴퓨팅 디바이스들을 만들었다. 예를 들어, 작고 경량이어서 사용자들이 용이하게 휴대할 수 있는, 휴대용 무선 전화들, 개인휴대단말(PDA)들, 및 페이징 디바이스들과 같은 무선 컴퓨팅 디바이스들을 포함하는 다양한 휴대용 개인 컴퓨팅 디바이스들이 현재 존재하고 있다. 특히, 휴대용 무선 전화들, 예를 들어 셀룰라 전화들 및 인터넷 프로토콜(IP) 전화들은 무선 네트워크들을 통해 음성 및 데이터 패킷들을 통신할 수 있다. 게다가, 이러한 많은 무선 전화들은 그에 통합되는 다른 타입들의 디바이스들을 포함한다. 예를 들어, 무선 전화들은 또한 디지털 스틸 카메라, 디지털 비디오 카메라, 디지털 레코더, 및 오디오 파일 플레이어를 포함할 수 있다. 또한, 이러한 무선 전화들은 인터넷에 액세스하기 위하여 사용될 수 있는 웹 브라우저 애플리케이션과 같은 소프트웨어 애플리케이션들을 포함하는 실행가능 명령들을 프로세싱 할 수 있는 프로세서를 포함한다. 따라서, 이를 무선 전화들은 상당한 컴퓨팅 능력을 포함할 수 있다.

[0004] 프로세서의 데이터 캐시에 액세스하는 것은 상당한 량의 전력을 소비한다. 데이터 캐시는 통상적으로 복수의 캐시 라인들(예를 들어, 저장 위치들)을 각각 포함하는 다수의 세트들을 가진 데이터 어레이를 포함한다. 데이터 캐시는 또한 통상적으로 데이터 캐시의 적어도 하나의 캐시 라인(예를 들어, 캐시 블록)에 대응하는 드라이버를 각각 포함하는 다수의 웨이(way)들을 포함한다. 데이터 캐시에 저장된 데이터에 액세스하기 위한 명령에 응답하여, 드라이버들 모두는 데이터 어레이의 특정 세트의 웨이들을 (복수의 데이터 라인들을 통해) 멀티플

렉서로 구동(drive)시키기 위하여 인에이블(예를들어, 활성화)된다.

[0005] 드라이버들 모두가 인에이블되는 것과 병렬로(예를들어, 인에이블되는 것과 동시에), 데이터 어레이 내의 특정 캐시 라인을 식별하기 위하여 태그 루업 동작이 수행된다. 태그 루업 동작의 결과에 기초하여, (단일 캐시 라인에 대응하는) 단일 드라이버를 통해 제공된 데이터는 출력으로서 선택된다. 세트에 대한 모든 웨이들을 구동시키고 태그 루업 동작을 수행하면, 명령에 기초하여 단지 단일 캐시 라인으로부터의 데이터가 출력된다는 것을 고려할 때, 비효율적으로 전력이 소비되고 전력이 만들어진다.

[0006] 프로세서의 명령 캐시에 액세스하는 것과 관련하여 유사한 전력 소비 문제가 존재한다. 명령 캐시에 대한 액세스들은 자주 예측가능하며, 명령들의 예측가능 시퀀스들을 활용하는 예측 방법들은 구동될 명령 캐시의 특정 웨이를 식별하기 위하여 사용될 수 있다. 그러나, 데이터 캐시에 액세스하는 것은 명령 캐시에 액세스하는 것보다 더 복잡하고 덜 예측가능하다. 따라서, 명령 캐시 액세스들을 위하여 사용되는 예측 기술들은 데이터 캐시 액세스들을 예측하기 위해서는 적용가능하지 않을 수 있다. 부가적으로, 만일 예측 기술이 데이터 캐시에 적용되었다면, 성능 패널티(penalty) (예를들어, 프로세싱의 지연) 및 에너지 패널티가 액세스될 웨이의 각각의 예측실패(misprediction)(예를들어, 부정확한 예측을 만듬)로부터 발생할 것이다.

### 발명의 내용

[0007] 프로세서의 데이터 캐시에 대한 웨이 예측 기술은 명령에 대하여 구동될 데이터 캐시의 웨이(예를들어, 하나 이상의 캐시 라인들과 연관된 웨이)를 추적하고(예를들어, 모니터링하고) 예측하기 위하여 예측 테이블(예를들어, 웨이 예측 테이블)을 활용한다. 특정 실시예에서, 예측된 웨이(예를들어, 명령의 이전 실행으로서 구동되는 동일한 웨이)는 명령의 이전 실행에 기초한다. 프로세서에 의해 실행되는 각각의 명령에 대하여, 제어로직은 예측 테이블을 패풀레이트(populate)하고 유지하며 그리고/또는 활용하여 예측된 웨이를 식별하기 위하여 각각의 명령의 실행을 모니터링하고 추적할 수 있다. 예를들어, 데이터 캐시의 제어 로직은 특정 명령을 표시하는 (예를들어, 식별하는) 프로그램 카운터(PC) 식별자, 특정 명령에 대하여 액세스되는 웨이, 및 특정 명령에 의해 수정되는 레지스터 파일의 베이스 레지스터 위치에 기초하여 예측 테이블을 사용하여 하나 이상의 명령들의 실행을 추적할 수 있다.

[0008] 하나 이상의 웨이 예측 특성들(예를들어, 명령의 어드레싱 모드, 명령의 명령 타입, 루프에 명령이 포함된다는 표시 등)을 가진 명령이 실행될 때, 제어 로직은 예측된 웨이가 식별될 수 있는지의 여부를 결정하기 위하여 예측 테이블을 판독할 수 있다. 예를들어, 웨이 예측 특성은 명령이 예측가능한 다음 어드레스(예를들어, 명령의 다음 실행에 기초하여 리트리브(retrieve)되는 유효 어드레스가 (예를들어, 동일한 웨이를 통해) 동일한 캐시 라인으로부터 이용가능할 것이라는 것을 표시하는 예측가능한 액세스 패턴)를 가질 수 있다는 것을 표시하는 명령의 특성(예를들어, 모드, 명령 타입, 루프 내의 위치 등) 또는 컴포넌트(예를들어, op-code, 오퍼랜드, 비트값 등)일 수 있다. 제어 로직은 명령에 대응하는 예측 테이블에 엔트리가 존재하는지의 여부를 결정할 수 있다. 특정 실시예에서, 하나 이상의 웨이 예측 특성들은 자동-증분 어드레싱 모드(auto-increment addressing mode) 또는 베이스 플러스 오프셋 어드레싱 모드(base plus offset addressing mode)와 같은 모드(예를들어, 어드레싱 모드)를 포함할 수 있다. 예측된 웨이는 루프의 이전 반복 동안 명령의 이전 실행과 같은, 명령의 이전 실행 동안 사전에 액세스된 웨이일 수 있다.

[0009] 예측 테이블이 명령에 대한 예측된 웨이를 표시할 때, 제어 로직은 예측된 웨이에 대응하는 드라이버를 선택적으로 인에이블(예를들어, 턴온)할 수 있으며, 예측된 웨이와 다른 웨이들에 대응하는 하나 이상의 다른 드라이버들을 선택적으로 디스에이블(예를들어, 턴오프)할 수 있다. 제어 로직은 또한 예측 테이블이 명령에 대한 예측된 웨이를 표시할 때 태그 어레이의 태그 루업 동작을 (예를들어, 스위치를 사용하여) 선택적으로 디스에이블할 수 있다. 하나 이상의 드라이버들을 선택적으로 디스에이블하고 태그 루업 동작을 선택적으로 디스에이블함으로써, 프로세서에 의해 전력 절약들이 실현된다.

[0010] 특정 실시예에서, 방법은 명령의 하나 이상의 웨이 예측 특성들을 식별하는 단계를 포함한다. 방법은 또한 하나 이상의 웨이 예측 특성들의 식별에 기초하여, 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위하여 테이블을 선택적으로 판독하는 단계를 포함한다. 방법은 명령에 기초한 데이터 캐시의 다음 액세스가 웨이에 액세스할지에 대한 예측을 수행하는 단계를 더 포함한다.

[0011] 다른 특정 실시예에서, 프로세서는 명령의 하나 이상의 웨이 예측 특성들을 식별하도록 구성된 디코드로직을 포함한다. 프로세서는 또한 디코드 로직에 커플링된 제어 로직을 포함한다. 제어 로직은 하나 이상의 웨이 예측 특성들에 기초하여, 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위

하여 테이블을 선택적으로 판독하도록 구성된다. 제어 로직은 명령에 기초한 데이터 캐시의 다음 액세스가 웨이에 액세스할지에 대한 예측을 수행하도록 추가로 구성된다.

[0012] 추가 특정 실시예에서, 장치는 명령의 하나 이상의 웨이 예측 특성들을 식별하기 위한 수단을 포함한다. 장치는 또한 하나 이상의 웨이 예측 특성들의 식별에 기초하여, 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위하여 테이블을 선택적으로 판독하기 위한 수단을 포함한다. 장치는 명령에 기초한 데이터 캐시의 다음 액세스가 웨이에 액세스할지에 대한 예측을 수행하기 위한 수단을 추가로 포함한다.

[0013] 또 다른 특정 실시예에서, 비-일시적 컴퓨터 판독가능 매체는 명령들을 포함하며, 명령들은, 프로세서에 의해 실행될 때, 프로세서로 하여금 명령의 하나 이상의 웨이 예측 특성들을 식별하도록 한다. 비-일시적 컴퓨터 판독가능 매체는 프로세서로 하여금 하나 이상의 웨이 예측 특성들의 식별에 기초하여, 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위하여 테이블을 선택적으로 판독하도록 하는 명령들을 더 포함한다. 비-일시적 컴퓨터 판독가능 매체는 프로세서로 하여금 명령에 기초한 데이터 캐시의 다음 액세스가 웨이에 액세스할지에 대한 예측을 수행하도록 하는 명령들을 더 포함한다.

[0014] 또 다른 특정 실시예에서, 방법은 명령의 제 1 실행 동안 충분값을 식별하는 단계 및 명령에 기초하여 제 1 실행동안 액세스되는 데이터 캐시의 웨이를 식별하는 단계를 포함한다. 방법은 제 1 충분된 어드레스 값을 결정하기 위하여 명령과 연관된 어드레스 값에 충분값을 가산하는 단계를 더 포함한다. 방법은 또한 제 1 충분된 어드레스 값이 데이터 캐시의 웨이에 로케이팅되는지의 여부를 결정하는 단계를 포함한다. 방법은 제 1 충분된 어드레스가 데이터 캐시의 웨이에 로케이팅된다는 결정에 응답하여, 테이블에서 명령에 대응하는 엔트리를 파풀레이트하는 단계를 더 포함한다.

[0015] 개시된 실시예들에 의해 제공된 하나의 특정 장점은 (예를들어, 명령 타입, 명령 어드레싱 모드, 루프에 있는 명령의 식별 또는 이들의 조합에 기초하여) 하나 이상의 명령들에 대한 예측 테이블을 유지하는 웨이 예측 기술이다. 웨이 예측 테이블은 웨이 예측에 기초하여 하나 이상의 드라이버들을 선택적으로 인에이블 및/또는 디스에이블하도록 활용될 수 있다. 하나 이상의 드라이버들을 선택적으로 인에이블 및/또는 디스에이블함으로써, 데이터 캐시의 데이터 액세스 동안 전력 절약이 실현될 수 있다. 부가적으로, 웨이 예측 테이블의 각각의 엔트리와 연관된 레지스터 위치를 모니터링하고, 추적하며 그리고 저장함으로써, 엔트리에 대응하는 명령과 다른 명령이 레지스터 위치의 데이터(예를들어, 콘텐츠)를 수정할 때 발생할 잠재적인 예측실패들이 방지될 수 있다. 엔트리(예를들어, 웨이 예측)가 유효한 것으로 확인된 이후에 태그 루프 동작을 선택적으로 디스에이블 함으로써 추가 전력 이득들이 실현될 수 있다.

[0016] 본 개시내용의 다른 양상들, 장점들 및 특징들은 이하의 단락들, 즉 도면의 간단한 설명, 상세한 설명 및 청구범위를 포함하는 출원을 검토한 이후에 명백하게 될 것이다.

### 도면의 간단한 설명

[0017] 도 1은 데이터 캐시에 대해 웨이 예측을 활용하는 프로세서 시스템의 엘리먼트들의 제 1 예시적인 실시 예의 다이어그램이다.

[0018] 도 2는 데이터 캐시에 대해 웨이 예측을 수행하기 위한 방법의 제 1 예시적인 실시예의 흐름도이다.

[0019] 도 3은 데이터 캐시에 대해 웨이 예측을 수행하기 위한 방법의 제 2 예시적인 실시예의 흐름도이다.

[0020] 도 4는 웨이 예측들을 위하여 사용된 데이터 캐시의 데이터 어레이 및 루프에서 명령들을 포함하는 프로그램 코드의 예시적인 실시예의 블록도이다.

[0021] 도 5는 데이터 캐시에 대해 웨이 예측을 수행하기 위한 방법의 제 3 예시적인 실시예의 흐름도이다.

[0022] 도 6은 웨이 예측을 수행하기 위한 데이터 캐시 및 로직을 포함하는 무선 통신 디바이스의 특정 실시예의 블록도이다.

### 발명을 실시하기 위한 구체적인 내용

[0023] 도 1은, 웨이 예측 테이블(152)을 활용하는 프로세서 시스템(100)의 엘리먼트들의 제 1 특정 실시예를 예시한다. 프로세서 시스템(100)은 데이터 캐시(102), 제어 로직(150), 프로그램 카운터(170), 태그 어레이(180), 및 디코드 로직(190)을 포함한다. 데이터 캐시(102)는 복수의 캐시 라인들(120a-d)을 포함하는 데이터 어레이(110)를 포함한다. 특정 실시예에서, 데이터 캐시(102)는 세트-연관(set-associative) 데이터 캐시를 포

함한다.

[0019] [0024] 프로세서 시스템(100)은 프로그램에 포함된 명령들(예를들어, 일련의 명령)을 실행(예를들어, 프로세싱)하도록 구성된다. 프로그램은 루프 또는 다수의 루프들(여기서, 일련의 명령들이 한번 이상 실행됨)을 포함할 수 있다. 프로그램은, 명령이 예측가능한 다음 어드레스(예를들어, 실행될 다음 명령에 대한 유효 어드레스가 (예를들어, 동일 웨이를 통해) 동일 캐시 라인으로부터 이용가능할 것이라는 것을 표시하는 예측가능한 액세스 패턴)를 가질 수 있다는 표시인 하나 이상의 웨이 예측 특성들(예를들어, 명령의 어드레싱 모드, 명령의 명령 탑입, 명령이 루프에 포함된다는 표시 등)을 갖는 명령들과 같은 하나 이상의 명령들을 포함할 수 있다. 예를들어, 명령의 어드레싱 모드는 자동 증분 어드레싱 모드 및/또는 베이스 플러스 오프셋 어드레싱 모드를 포함할 수 있으며, 이는 데이터 캐시(예를들어, 데이터 캐시(102))의 데이터 어레이의 캐시 라인의 동작을 야기한다. 자동 증분 어드레싱 모드를 사용하는 명령(예를들어, 자동 증분 명령)은 레지스터 파일의 레지스터 위치(예를들어, 베이스 레지스터)(도시안됨)을 식별할 수 있고, 증분량(예를들어, 정수값, 예컨대 1 또는 2) 만큼 레지스터 위치에 저장되는 콘텐츠들(예를들어, 어드레스 데이터)을 수정(예를들어, 증분)할 수 있다. 베이스 플러스 오프셋 어드레싱 모드를 사용하는 명령(예를들어, 베이스 플러스 오프셋 명령)은 명령의 각각의 실행 동안 레지스터 위치(예를들어, 베이스 레지스터 위치)에 액세스할 수 있고, 명령의 각각의 연속 실행과 함께 베이스 레지스터 위치의 데이터에 오프셋을 가산할 수 있다.

[0020] [0025] 자동 증분 어드레싱 모드 및/또는 베이스 플러스 오프셋 어드레싱 모드를 사용하는 명령들이 루프의 일부로서 실행(예를들어, 몇 번 실행)될 때, 명령들은 각각, 명령의 다음 실행에 기초하여 리트리브되는 유효 어드레스가 데이터 어레이(110)의 동일 캐시 라인(120a-d)(예를들어, 동일 웨이)으로부터 이용가능할 것이라는 것을 표시하는 예측가능한 액세스 패턴을 포함할 수 있다. 따라서, 명령들의 실행 동안(예를들어, 루프의 한번 이상의 반복들 동안), 자동 증분 어드레싱 모드 또는 베이스 플러스 오프셋 어드레싱 모드를 사용하는 명령에 대해 액세스되는 데이터 캐시(102)의 특정 웨이가 식별될 수 있다. 자동 증분 어드레싱 모드 또는 베이스 플러스 오프셋 어드레싱 모드를 사용하는 명령이 동일 레지스터상에서 동작하기 때문에, 사후-증분된(post-incremented) 또는 오프셋 어드레스가 명령의 이전 실행과 동일한, 데이터 캐시(102)의 캐시 라인(예를들어, 동일한 웨이)에 액세스할 수 있다는 것을 결정(예를들어, 확인)하는 것이 가능할 수 있다. 따라서, 프로세서 시스템(100)은 하나 이상의 명령들에 대한 웨이 액세스들을 예측하기 위해, 아래에 설명되는 바와 같이, 예측 테이블(152)을 생성하고, 유지하고, 사용할 수 있다.

[0021] [0026] 데이터 캐시(102)는 데이터 어레이(110) 및 멀티플렉서(160)를 포함할 수 있다. 데이터 캐시(102)는 최근에 또는 빈번하게 사용된 데이터를 (캐시 라인에) 저장하도록 구성될 수 있다. 데이터 캐시(102)에 저장된 데이터는 메인 메모리(도시안됨)와 같은 다른 위치로부터 액세스되는 데이터보다 더 빠르게 액세스될 수 있다. 특정 실시예에서, 데이터 캐시(102)는 4-웨이 세트-연관 캐시와 같은 세트-연관 캐시이다. 부가적으로 또는 대안적으로, 데이터 캐시(102)는 제어 로직(150), 프로그램 카운터(170), 태그 어레이(180), 디코드 로직(190), 또는 이들의 조합을 포함할 수 있다.

[0022] [0027] 데이터 어레이(110)는 (프로세서 시스템(100)에 의해 실행되는) 명령의 실행 동안 액세스될 수 있다. 명령은 프로그램(예를들어, 일련의 명령들)에 포함될 수 있고, 프로그램의 루프(예를들어, 소프트웨어 루프)에 포함될 수도 있고 또는 포함되지 않을 수도 있다. 데이터 어레이(110)는 복수의 세트들(예를들어, 로우(row)들)을 포함하며, 이 로우들 각각은 도 1에 도시된 것처럼 제 1 웨이, 제 2 웨이, 제 3 웨이 및 제 4 웨이와 같은 복수의 웨이들(예를들어, 컬럼(column)들)을 포함한다. 웨이들 각각은 데이터 캐시(102)의 컬럼내의 다수의 캐시 라인들과 연관될 수 있고 데이터 캐시(102)의 각각의 세트의 대응하는 캐시 라인(120a-d)(예를들어, 단일 캐시 라인)과 연관될 수 있다. 복수의 웨이들은 프로그램의 실행동안 액세스될 수 있다. 복수의 웨이들의 각각의 웨이는 데이터 어레이(110)의 컬럼내의 다수의 캐시 라인들(예를들어, 저장 위치들)에 대응하는 데이터 라인(130a-d) 및 드라이버(140a-d)(예를들어, 라인 드라이버)를 포함할 수 있다. 예를들어, 제 1 웨이는 캐시 라인 A(120a)와 연관될 수 있으며 제 1 드라이버(140a) 및 제 1 데이터 라인(130a)을 포함하고, 제 2 웨이는 캐시 라인 B(120b)와 연관될 수 있으며 제 2 드라이버(140b) 및 제 2 데이터 라인(130b)을 포함하고, 제 3 웨이는 캐시 라인 C(120c)와 연관될 수 있으며 제 3 드라이버(140c) 및 제 3 데이터 라인(130c)을 포함하고, 제 4 웨이는 캐시 라인 D(120d)와 연관될 수 있으며 제 4 드라이버(140d) 및 제 4 데이터 라인(130d)을 포함한다.

[0023] [0028] 각각의 드라이버(140a-d)는 대응하는 캐시 라인(120a-d)(예를들어, 대응하는 캐시 블록)에 저장된 데이터가 대응하는 데이터 라인(130a-d)을 통해 데이터 어레이(110)로부터 판독(예를들어, 구동)되고 멀티플렉서(160)에 제공되도록 할 수 있다. 캐시 라인들(120a-d)의 특정 캐시 라인에 저장된 콘텐츠는 다수 바이트들(예

를들어, 32 바이트들 또는 64 바이트들)을 포함할 수 있다. 특정 실시예에서, 특정 캐시 라인은 순차적으로 어드레싱되는 메모리 위치들의 블록에 대응할 수 있다. 예를들어, 특정 캐시 라인은 8개의 순차적으로 어드레싱되는 메모리 위치들(예를들어, 8개의 4-바이트 세그먼트들)의 블록에 대응할 수 있다.

[0024]

[0029] 디코드 로직(190)은 프로세서 시스템(100)에 의해 실행될 하나 이상의 명령들(예를들어, 일련의 명령)을 수신할 수 있다. 디코드 로직(190)은 하나 이상의 명령들 중 특정 명령을 디코딩하고 그리고 (인덱스 부분(172), 태그 부분(174), 또는 이들의 조합을 포함하는) 디코딩된 명령을 프로그램 카운터(170)에 제공하도록 구성된 디코더를 포함할 수 있다. 디코드 로직(190)은 또한, 예컨대 데이터를 전송하거나 또는 하나 이상의 제어 레지스터들을 수정함으로써, 특정 명령과 연관된 명령 데이터를 제어 로직(150)에 제공하도록 구성될 수 있다. 예를들어, 명령 데이터는 디코딩된 명령(예를들어, 인덱스 부분(172) 및/또는 태그 부분(174)), 하나 이상의 웨이 예측 특성들, 특정 명령의 명령 타입(예를들어, 로드 타입, 저장 타입 등), 특정 명령의 모드(예를들어, 어드레싱 모드), 특정 명령과 연관된 레지스터 파일(도시안됨)의 하나 이상의 레지스터 위치들, 특정 명령과 연관된 증분 값(및/또는 오프셋 값), 특정 명령과 연관된 어드레스 값, 특정 명령이 루프(예를들어, 소프트웨어 루프)를 개시하는지, 루프를 종료하는지, 또는 루프에 포함되었는지 여부, 또는 이들의 조합을 포함할 수 있다. 하나 이상의 웨이 예측 특성들은, 특정 명령이 예측가능한 다음 어드레스(예를들어, 실행될 다음 명령에 대한 유효 어드레스가(예를들어, 동일 웨이를 통해) 동일 캐시 라인으로부터 이용가능할 것이라는 것을 표시하는 예측가능한 어드레스 패턴)를 가짐을 표시할 수 있다. 예를들어, 하나 이상의 웨이 예측 특성들은 특정 명령의 특성(예를들어, 어드레싱 모드, 명령 타입, 루프내의 포지션 등), 특정 명령의 컴포넌트(예를들어, op-code, 오퍼랜드, 비트 값, 증분 값, 레지스터 값 등), 또는 이들의 조합을 포함할 수 있다. 특정 명령의 어드레싱 모드는 자동 증분 어드레싱 모드 또는 베이스 플러스 오프셋 어드레싱 모드를 포함할 수 있다. 특정 명령의 명령 타입은 로드 타입 또는 저장 타입을 포함할 수 있다.

[0025]

[0030] 프로그램 카운터(170)는 디코드 로직(190)으로부터 수신되는 디코딩된 명령에 기초하여 실행될 명령을 식별할 수 있다. 프로그램 카운터(170)는 명령의 실행 동안 데이터 캐시(102)에 액세스하는데 사용될 수 있는 태그 부분(174) 및 인덱스 부분(172)(예를들어, 세트 인덱스 부분)을 포함할 수 있다. 명령이 실행될 때마다, 프로그램 카운터(170)는 실행될 다음 명령을 식별하도록 조절(예를들어, 증분)될 수 있다.

[0026]

[0031] 제어 로직(150)은, 웨이 예측 테이블(152), 태그 어레이 인에이블(154) 및 드라이버 인에이블(156)을 포함할 수 있다. 제어 로직(150)은, 아래에 추가로 설명되는 것처럼, 디코드 로직(190)으로부터 명령 데이터를 수신하고 명령 데이터의 적어도 일부에 기초하여 웨이 예측 테이블(152)에 액세스하도록 구성될 수 있다. 예를들어, 제어 로직(150)은 디코드 로직(190)으로부터 수신된 하나 이상의 웨이 예측 특성들에 기초하여 웨이 예측 테이블(152)에 선택적으로 액세스할 수 있다.

[0027]

[0032] 웨이 예측 테이블(152)은, 각각이 하나 이상의 필드들을 포함하는 하나 이상의 엔트리들(153)을 포함할 수 있다. 각각의 엔트리(153)는 상이한 명령에 대응할 수 있으며 프로그램 카운터(PC) 필드, 예측된 웨이(WAY) 필드, 레지스터 위치 식별자(REG) 필드, 유효/무효 필드(V/I), 또는 이들의 조합을 포함할 수 있다. 특정 엔트리에 대해, PC 필드는 프로세서 시스템(100)에 의해 실행되는 대응하는 명령을 식별할 수 있다. WAY 필드(예를들어, 예측된 웨이 필드)는 대응하는 명령이 실행되었던 마지막 시간에 미리 액세스된 (데이터 어레이(110)의) 웨이(예를들어, 액세스된 "마지막 웨이")를 식별하는 값(예를들어, 웨이 필드 식별자)을 포함할 수 있다. REG 필드는 대응하는 명령이 실행되었던 마지막 시간에 수정된 레지스터 파일의 레지스터 위치(도시안됨)를 식별할 수 있다. 예를들어, 레지스터 위치는 사후-증분 동작의 일부로서 명령의 실행에 기초하여 수정된 명령의 베이스 레지스터 위치일 수 있다. V/I 필드는 WAY 필드의 값이 유효인지 또는 무효인지의 여부를 식별할 수 있다. 예를들어, V/I 필드는 WAY 필드의 값이 예측된 웨이로서 이용될 수 있는지의 여부를 표시할 수 있다. 대안적으로 그리고/또는 부가적으로, V/I 필드는 엔트리가 유효인지 또는 무효인지의 여부를 표시할 수 있다. 웨이 예측 테이블(152)은 프로세서 시스템(100)의 프로세서 코어에 유지(예를들어, 저장)될 수 있고 그리고/또는 데이터 캐시(102)의 프리페치 테이블(prefetch table)에 포함되거나 또는 이와 연관될 수 있다. 특정 실시예에서, 웨이 예측 테이블(152)의 각각의 엔트리는 프로그램 카운터 식별자(예를들어, PC 필드), 특정 레지스터 위치 식별자(예를들어, REG 필드), 및 특정 예측된 웨이 식별자(예를들어, WAY 필드)를 포함한다.

[0028]

[0033] 제어 로직(150)은 디코드 로직(190)에 의해 제공되는 명령 데이터(예를들어, 실행될 명령에 대응하는 명령 데이터)에 액세스하도록 구성될 수 있다. 하나 이상의 웨이 예측 특성들과 같은 명령 데이터의 적어도 일부에 기초하여, 제어 로직(150)은 웨이 예측 테이블(152)이 명령에 대응하는 엔트리를 포함하는지의 여부를 결정할 수 있다. 예를들어, 명령의 명령 타입이 로드 타입 또는 저장 타입이라는 표시를 수신하는 것에 응답하여, 제어 로직(150)은 웨이 예측 테이블(152)을 선택적으로 판독할 수 있다. 특정 실시예에서, 명령 타입이 로드

타입 또는 저장 타입이 아니라면, 제어 로직(150)은 웨이 예측 테이블을 판독하지 않는다. 제어 로직(150)은, 웨이 예측 테이블(152)의 PC 필드들에 기초하여, 웨이 예측 테이블(152)이 명령에 대응하는 엔트리(153)를 포함하는지의 여부를 결정할 수 있다. 또 다른 특정 실시예에서, 명령의 어드레싱 모드가 자동 충분 어드레싱 모드 또는 베이스 플러스 오프셋 어드레싱 모드라는 표시를 수신하는 것에 응답하여, 제어 로직(150)은 웨이 예측 테이블(152)을 선택적으로 판독한다. 또 다른 특정 실시예에서, 제어 로직(150)은 명령이 루프에 포함된다는 표시를 수신하는 것에 응답하여, 웨이 예측 테이블(152)을 선택적으로 판독한다.

[0029]

[0034] 웨이 예측 테이블(152)이 명령에 대응하는 엔트리(153)를 포함하지 않는다는 결정에 기초하고 그리고 웨이 예측이 유용한(예를들어, 자동 충분 어드레스 모드를 가짐) 하나 이상의 웨이 예측 특성들과 명령이 연관된다는 결정에 기초하여, 제어 로직(150)은 웨이 예측 테이블(152)의 명령과 연관된 새로운 엔트리(153)를 생성(예를들어, 파플레이트)할 수 있다. 제어 로직(150)은 명령에 포함되는(예를들어, 그 명령에 의해 식별되는) 레지스터 위치 및 그 명령에 기초하여 액세스되는 데이터 어레이(110)의 웨이를 식별할 수 있다. 제어 로직(150)은 식별된 레지스터 위치 및 식별된 웨이에 기초하여 새로운 엔트리(153)의 REG 필드 및 WAY 필드를 각각 파풀레이트 할 수 있다. 따라서, 명령이 실행될 다음 시간에(예를들어, 루프의 다음 반복 동안에), 제어 로직(150)은 WAY 필드에 기초하여 명령의 이전 실행 동안에 액세스된 웨이를 식별할 수 있다. 특히, 엔트리가 생성될 때, WAY 필드의 값은 엔트리가 생성되게 했던 명령의 실행에 기초하여 액세스된 웨이를 표시하도록 세팅될 수 있다. REG 필드는 본원에서 추가로 설명되는 바와 같이 웨이 예측 테이블(152)을 유지하기 위해 제어 로직(150)에 의해서 사용될 수 있다.

[0030]

[0035] 제어 로직(150)은 명령의 후속(예를들어, 다음) 실행이 엔트리가 생성되게 했던 명령의 실행과 동일한 웨이에 액세스할지 여부를 또한 예측할 수 있다. 예를들어, 도 1과 관련하여 더 상세히 설명된 바와 같이, 제어 로직(150)은, 명령의 다음 실행이 예를들어 자동-충분 어드레싱 모드 명령 또는 베이스 플러스 오프셋 어드레싱 모드 명령에 따라 그 실행과 동일한 캐시 라인에 그리고 이에 따라 동일한 웨이에 액세스할지 여부를 예측(예를들어, 확인)하기 위한 산술 연산을 수행할 수 있다. 충분된 어드레스가 명령의 실행 동안에 액세스된 동일한 캐시 라인에 있지 않을 것이라는 결정(예를들어, 예측)이 수행될 때, 제어 로직(150)은 WAY 필드의 값이 명령의 후속 실행 동안에 사용될 예측된 웨이를 표시하기 위해 필요하지 않을 수 있다는 것을 표시하기 위하여 새로운 엔트리의 V/I 필드(예를들어, 유효성 비트)를 무효한 것으로 세팅할 수 있다. 충분된 어드레스가 명령의 실행 동안에 액세스된 동일한 캐시 라인에 있을 것이라는 결정(예를들어, 예측)이 수행될 때(예를들어, 캐시 라인 A(120a)에 액세스한 명령의 실행 및 충분된 어드레스와 연관된 값이 캐시 라인 A(120a)에 로케이팅됨), 제어 로직(150)은 WAY 필드의 값이 명령의 후속 실행 동안에 사용될 예측된 웨이를 표시함을 표시하기 위해서 새로운 엔트리의 V/I 필드(예를들어, 유효성 비트)를 유효한 것으로 세팅할 수 있다.

[0031]

[0036] 제어 로직(150)은 실행될 명령에 대한 웨이를 예측하기 위해 웨이 예측 테이블(152)을 사용할 수 있다. 제어 로직(150)은 각각의 엔트리(153)의 PC 필드에 기초하여 명령에 대응하는 웨이 예측 테이블(152)의 엔트리(153)를 식별하기 위해 웨이 예측 테이블(152)을 선택적으로 판독할 수 있다. 제어 로직(150)이 대응하는 엔트리(153)를 식별할 때 그리고 엔트리(153)가 유효한 것으로서 표시될 때, 제어 로직(150)은 엔트리(153)에 대한 WAY 필드의 값을 드라이버 인에이블(156)에게 제공함으로써(또는 이용가능하게 함으로써) 그 WAY 필드의 값을 웨이 예측으로서 사용할 수 있고, V/I 필드의 값을 태그 어레이 인에이블(154)에게 제공할 수 있다(또는 이용가능하게 할 수 있다).

[0032]

[0037] 드라이버 인에이블(156)은 웨이 예측 테이블(152)에서 식별되는 예측된 웨이에 기초하여 드라이버들(140a-d) 중 하나 이상을 선택적으로 활성화(예를들어, 턴온) 또는 비활성화(예를들어, 턴오프)하도록 구성될 수 있다. 특정 실시예에서는, 드라이버 인에이블(156)에 제공되는 WAY 필드의 값이 널 값(예를들어, 제로 값)일 때, 드라이버 인에이블(156)은 드라이버들(140a-d) 모두를 인에이블한다. 다른 특정 실시예에서, 드라이버 인에이블(156)은, 식별된(예를들어, 대응하는) 엔트리(153)의 V/I 필드의 값이 그 엔트리(153)의 WAY 필드의 값이 웨이 예측으로서 사용될 수 있음을 표시할 때, 그 엔트리(153)로부터의 예측된 웨이를 사용할 수 있다. 부가적으로, 드라이버 인에이블(156)은 WAY 필드에 의해 표시되는 예측된 웨이가 아닌 하나 이상의 웨이들의 적어도 하나의 드라이버(140a-d)를 선택적으로 디스에이블할 수 있다. 특정 실시예에서, WAY 필드는 예측된 웨이를 표시하는 하나 이상의 비트들(예를들어, 비트마스크)을 포함할 수 있고, 드라이버 인에이블(156)은 복수의 드라이버들(140a-d)의 각각의 드라이버를 선택적으로 인에이블 또는 디스에이블하기 위해 그 복수의 드라이버들(140a-d)에 비트마스크를 적용할 수 있다.

[0033]

[0038] 태그 어레이 인에이블(154)은 명령에 기초하여 선택될 웨이(예를들어, 캐시 라인(120a-d))를 식별하기 위해 태그 어레이(180)에서의 태그 루프 동작을 스위치(176)(또는 다른 메커니즘)를 통해 선택적으로 활성화(예

를들어, 인에이블) 또는 비활성화(예를들어, 디스에이블)하도록 구성될 수 있다. WAY 필드의 값이 웨이 예측으로서 사용될 수 있음을 V/I 필드의 값이 표시한다고 태그 인에이블 부분(154)이 결정할 때, 태그 인에이블 부분(154)이 스위치(176)의 동작을 통해 태그 룩업 동작을 선택적으로 디스에이블할 수 있다. WAY 필드의 값이 웨이 예측으로서 사용될 수 없음을 V/I 필드의 값이 표시할 때, 태그 어레이 인에이블(154)은 드라이버들(140a-d)이 인에이블되는 것과 병렬로 (예를들어, 동시에) 태그 룩업 동작이 수행되도록 스위치(176)를 선택적으로 인에이블할 수 있다.

[0034] 드라이버 인에이블(156)에 제공되는 WAY 필드의 값과 태그 어레이 인에이블(154)에 제공되는 V/I 필드의 값의 조합들이 드라이버 인에이블(156) 및/또는 태그 어레이 인에이블(154)의 동작을 관리하기 위하여 제공될 수 있다. 예를들어, WAY 필드의 값은 드라이버 인에이블(156)로 하여금 그 WAY 필드의 값에 상관없이 모든 드라이버들(140a-d)을 활성화(예를들어, 턴온)하도록 하는 널 값(예를들어, 제로 값)일 수 있고, 이 경우에는 태그 어레이 인에이블(154)이 스위치(176)를 선택적으로 인에이블하거나 선택적으로 디스에이블할 수 있다. 다른 예로서, WAY 필드의 값이 웨이 예측으로서 필요하지 않을 수 있음을 V/I 필드의 값이 표시할 때, 드라이버 인에이블(156)은 모든 드라이버들(140a-d)을 턴온 시킬 수 있고, 태그 어레이 인에이블(154)은 스위치(176)를 선택적으로 인에이블할 수 있다. 추가 예로서, WAY 필드의 값이 웨이 예측으로서 필요(예를들어, 사용)할 수 있음을 V/I 필드의 값이 표시할 때, 드라이버 인에이블(156)이 복수의 드라이버들(140a-d) 중 단일 라인 드라이버를 활성화(예를들어, 턴온)할 수 있고, 태그 어레이 인에이블(154)이 스위치(176)를 선택적으로 인에이블하거나 선택적으로 디스에이블할 수 있다. 대안적으로 또는 부가적으로, 태그 어레이 인에이블(154)은 레지스터 파일이 추적(예를들어, 모니터링)되고 있는지 여부, WAY 필드의 값 및 V/I 필드의 값을 제공하는 엔트리에 대응하는 명령이 (예를들어, 제어 로직(150)에 수신되는 명령 데이터에 기초하여) 루프에 포함되는 것으로서 식별되는지 여부, 또는 이들의 조합에 기초해서 스위치(176)를 선택적으로 인에이블 또는 디스에이블할 수 있다.

[0040] 제어 로직(150) 또는 그 제어 로직(150)에 커플링되는 다른 로직은, 임의의 명령이 웨이 예측 테이블(152)에서 식별되는 레지스터 위치에 있는 데이터를 수정(예를들어, 변경)하는지 여부를 추적(예를들어, 모니터링)하는 추적 로직을 포함할 수 있다. 추적 로직은 수정된 레지스터 위치의 값을 식별하고, 그 레지스터 위치의 식별을 제어 로직(150)에 제공할 수 있다. 제어 로직(150)은, 특정 엔트리(153)가 레지스터 위치에 대응하는 값을 갖는 REG 필드를 포함하는지 여부를 결정하기 위해서 웨이 예측 테이블(152)을 판독할 수 있다. 특정 엔트리(153)가 그러한 REG 필드를 포함한다는 결정에 기초하여, 제어 로직(150)은, 특정 엔트리(153)의 PC 필드가 레지스터 위치를 수정한 특정 명령에 대응하는지 여부를 결정할 수 있으며, 그 특정 엔트리(153)가 그 특정 명령에 대응하지 않을 때, 제어 로직(150)은, 특정 엔트리의 WAY 필드의 값이 웨이 예측으로서 필요(예를들어, 사용)하지 않을 수 있음을 표시하도록 그 특정 엔트리(153)의 V/I 필드의 값을 (예를들어, 무효로) 세팅할 수 있거나 또는 그 특정 엔트리(153)를 제거(예를들어, 삭제)할 수 있다.

[0041] 프로세서 시스템(100)의 동작 동안에, 디코드 로직(190) 및/또는 제어 로직(150)은 웨이 예측 테이블(152)이 실행될 명령에 대응하는 엔트리를 포함하는지 여부를 결정할 수 있다. 웨이 예측 테이블(152)이 엔트리를 포함하지 않을 때, 제어 로직(150)은 웨이 예측 테이블(152)에서 새로운 엔트리를 생성할 수 있다. 웨이 예측 테이블(152)이 엔트리를 포함할 때, 제어 로직(150)은 엔트리의 하나 이상의 필드들의 하나 이상의 값을 식별할 수 있다. 엔트리가 유효하지 않음(예를들어, 엔트리가 웨이 예측을 위해 사용될 수 없음)을 하나 이상의 필드들이 표시할 때, 제어 로직(150)은 데이터 어레이(110)의 복수의 드라이버들(140a-d) 모두를 인에이블할 수 있고, 태그 어레이(180)의 출력에 기초하여 멀티플렉서(160)에 웨이 선택 신호가 제공되게 할 수 있다. 엔트리가 유효함을 하나 이상의 필드들이 표시할 때, 제어 로직(150)은 복수의 드라이버들(140a-d) 중 하나 이상을 선택적으로 인에이블 및/또는 디스에이블하기 위해서 그리고 멀티플렉서(160)에 의한 선택을 제어하기 위해서 엔트리의 WAY 필드의 값을 사용할 수 있다. 제어 로직(150)은, 충분된 어드레스가 WAY 필드에 의해 표시되는 웨이와는 상이한 웨이에 대응하는 캐시 라인에 액세스할 것이라는 예측에 기초하여 또는 웨이 예측 테이블(152)에서 식별되는 레지스터 위치에 대한 수정을 식별하는 것에 기초하여 웨이 예측 테이블(152)의 하나 이상의 엔트리들을 업데이트할 수 있다. 프로세서 시스템(100)의 동작의 예가 도 4를 참조하여 아래에 설명된다.

[0042] 프로세서 시스템(100)에 의해 실행되는 명령들에 대한 웨이 예측 테이블(152)을 유지함으로써, 데이터 캐시(102)의 데이터 어레이(110)의 하나 이상의 드라이버들(140a-d)은 웨이 예측에 기초하여 선택적으로 디스에이블될 수 있고, 전력 이득이 데이터 캐시(102)의 데이터 액세스 동안 실현될 수 있다. 부가적으로, 각각의 엔트리(153)와 연관된 레지스터 위치(예를들어, REG 필드)를 추적하고 저장함으로써, 제어 로직(150)은, 엔트리에 대응하는 명령과 다른 명령이 웨이 예측 테이블(152)의 임의의 엔트리의 REG 필드에 의해 식별되는 특정 레지스

터 위치에 있는 데이터를 수정할 때, 잠재적인 예측실패들을 방지할 수 있다. 추가적인 전력 이득들이 태그 톡업 동작을 선택적으로 디스에이블시킴으로써 실현될 수 있다.

[0038] [0043] 도 2를 참조하면, 데이터 캐시와 연관된 웨이 예측을 수행하기 위한 방법(200)의 제 1 예시적인 실시예의 흐름도가 예시된다. 예를들어, 데이터 캐시는 도 1의 데이터 캐시(102)를 포함할 수 있다. 특정 실시예에서, 방법(200)은 도 1의 제어 로직(150)에 의해 수행될 수 있다.

[0039] [0044] 명령의 제 1 실행동안 중분 값이 식별된다(202). 중분 값은 자동 중분 어드레싱 모드를 사용하는 명령과 연관될 수 있다. 중분 값은 도 1의 디코드 로직(190)과 같은 디코드 로직에 의해 결정(예를들어, 식별)될 수 있다. 중분 값은 디코드 로직으로부터 도 1의 제어 로직(150)과 같은 제어 로직에 제공되는 명령 데이터에 포함될 수 있다. 제어 로직은 명령 데이터를 수신하고 중분 값을 식별할 수 있다. 제어 로직은 또한, 명령이 예측가능 액세스 패턴을 가질 수 있음을 표시하는 하나 이상의 웨이 예측 특성들과 그 명령이 연관되는지 여부를 결정할 수 있다. 특정 실시예에서, 제어 로직은, 명령이 하나 이상의 웨이 예측 특성들과 연관된다는 결정을 수행한 이후에 그 명령의 중분 값을 식별한다.

[0040] [0045] 명령의 제 1 실행동안 그 명령에 기초하여 액세스되는 데이터 캐시의 웨이가 식별된다(204). 예를들어, 데이터 캐시는 도 1의 데이터 캐시(102)일 수 있다. 제어 로직은 명령의 제 1 실행동안 액세스되는 웨이를 식별할 수 있다.

[0041] [0046] 중분된 어드레스 값을 결정하기 위해서 명령과 연관된 어드레스 값에 중분 값이 가산된다(206). 제어 로직은 중분된 어드레스를 결정하기 위해서 명령과 연관된 어드레스 값에 중분 값을 가산할 수 있다. 특정 실시예에서, 어드레스 값은 명령에 의해 식별되는 레지스터 위치에 저장된 어드레스 값일 수 있다. 레지스터 위치는 디코드 로직에 의해 제공되는 명령 데이터에 기초하여 제어 로직에 의해 식별될 수 있다.

[0042] [0047] 중분된 어드레스 값이 데이터 캐시의 웨이에 로케이팅되는지 여부가 결정된다(208). 제어 로직은 명령의 후속(예를들어, 다음) 실행이 명령의 실행(즉, 제 1 실행)과 동일한 웨이에 액세스할 것으로 예측되는지 여부를 결정할 수 있다.

[0043] [0048] 명령에 대응하는 엔트리는 테이블에서 파풀레이트된다(210). 엔트리는 중분된 어드레스가 데이터 캐시의 웨이에 로케이팅됨을 결정하는 것에 응답하여 테이블에서 파풀레이트될 수 있다. 제어 로직은 도 1의 웨이 예측 테이블(152)과 같은 웨이 예측 테이블에서 명령에 대응하는 엔트리를 파풀레이트할 수 있다(예를들어, 생성할 수 있다). 특정 실시예에서, 테이블에서 엔트리의 생성(예를들어, 파풀레이션)은 엔트리가 생성(예를들어, 파풀레이트)되기 전에 만족되는, 명령과 연관된 하나 이상의 조건들(예를들어, 자동 중분 어드레싱 모드, 명령의 타입, 루프에 있는 명령)에 대하여 컨디셔닝된다. 제어 로직은 엔트리가 명령(예를들어, PC 필드 값), 명령의 제 1 실행동안 액세스되는 데이터 캐시의 웨이(예를들어, WAY 필드 값), 레지스터 위치(예를들어, REG 필드 값), 명령의 후속(예를들어, 다음) 실행이 동일한 캐시 라인(예를들어, V/I 필드 값)에 액세스하는 것으로 예측되는지의 여부, 또는 이들의 조합을 식별하도록 엔트리의 하나 이상의 필드들을 파풀레이트할 수 있다.

[0044] [0049] 웨이 예측 테이블에서 명령에 대한 엔트리를 생성(예를들어, 파풀레이트)함으로써, 명령에 기초하여 액세스된 웨이는 기록 및 추적될 수 있다. 기록된 웨이는 데이터 캐시의 데이터 액세스 동안 전력 이득을 실현하도록 데이터 캐시의 하나 이상의 드라이버들을 선택적으로 인에이블 및/또는 디스에이블하기 위해 (예를들어, 드라이버들 모두가 턴온되는 것은 아님) 명령의 하나 이상의 후속 실행들 동안 제어 로직에 의해 경로 예측으로서 사용될 수 있다.

[0045] [0050] 도 3을 참조하면, 데이터 캐시와 연관된 웨이 예측을 수행하기 위한 방법(300)의 제 2 예시적인 실시예의 흐름도가 예시된다. 예를들어, 데이터 캐시는 도 1의 데이터 캐시(102)를 포함할 수 있다. 특정 실시예에서, 방법(300)은 도 1의 제어 로직(150)에 의해 수행될 수 있다.

[0046] [0051] 명령의 어드레싱 모드가 식별된다(302). 특정 실시예에서, 명령의 어드레싱 모드는 자동 중분 어드레싱 모드로서 식별된다. 자동 중분 어드레싱 모드를 가진 명령은 명령과 연관된 어드레스를 저장하는 레지스터 위치 및 중분값을 식별할 수 있다. 어드레싱 모드는 디코드 로직(190)과 같은 디코드 로직으로부터 수신된 명령 데이터에 기초하여, 도 1의 제어 로직(150)과 같은 제어 로직에 의해 결정될 수 있다(예를들어, 식별될 수 있다). 부가적으로, 명령과 연관된 타입(예를들어, 명령 타입)이 결정될 수 있다. 예를들어, 타입은 로드 타입 또는 저장 타입인 것으로 결정될 수 있다. 타입은 제어 로직에 의해 결정될 수 있다(예를들어, 식별될 수 있다)

[0047]

[0052] 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위하여 명령의 식별에 기초하여 테이블이 판독된다(304). 제어 로직은 테이블이 데이터 캐시의 웨이를 식별하는, 명령과 연관된 엔트리를 포함하는지의 여부를 결정할 수 있다. 예를들어, 제어 로직은 테이블에 액세스하고, 명령에 기초하여 테이블로부터 엔트리를 판독한다. 테이블은 도 1의 웨이 예측 테이블(152)을 포함할 수 있다. 테이블이 엔트리를 포함한다는 결정이 수행될 때, 제어 로직은 엔트리와 연관된 V/I 필드에 포함된 유효성(validity) 비트의 값에 기초하여 엔트리가 유효한 것으로 식별되는지 또는 무효한 것으로 식별되는지의 여부를 결정할 수 있다. V/I 필드는 엔트리에 포함될 수 있거나 또는 테이블과 완전히 다른 레지스터 위치 또는 버퍼에 저장될 수 있다. 유효성 비트의 값은 엔트리 또는 엔트리의 일부분(예를들어, 적어도 하나의 필드)이 명령에 대한 웨이 예측을 제공하는데 유효한지의 여부를 결정하기 위하여 제어 로직을 인에이블할 수 있다. 엔트리에 기초하여, 제어 로직은 데이터 캐시의 하나 이상의 드라이버들을 선택적으로 인에이블 또는 디스에이블하도록 (예를들어, 드라이버들 모두가 턴온되는 것이 아님) 엔트리에 포함된 경로 예측을 사용할 수 있다. 데이터 캐시의 하나 이상의 드라이버들을 선택적으로 디스에이블함으로써, 데이터 캐시의 데이터 액세스 동안 전력 이득이 실현된다.

[0048]

[0053] 명령에 기초한 데이터 캐시의 다음 액세스가 동일한 웨이에 액세스할 것인지에 대한 예측이 수행된다(306). 예를들어, 제어 로직은 명령에 기초한 데이터 캐시의 다음 액세스가 엔트리에 의해 식별되는 웨이에 액세스할 것인지의 여부를 예측할 수 있다. 제어 로직은 충분된 어드레스를 결정하기 위하여 명령과 연관된 레지스터 위치의 어드레스(예를들어, 레지스터 위치에 저장된 어드레스)에 명령과 연관된 충분값을 가산하고 충분된 어드레스가 어드레스와 동일한 데이터 어레이의 캐시 라인에 로케이팅되는지의 여부를 결정함으로써 예측을 수행할 수 있다.

[0049]

[0054] 데이터 캐시의 다음 액세스가 웨이에 액세스하지 않을 것이라고 예측이 수행될 때, 테이블의 엔트리가 무효화되거나 또는 삭제(예를들어, 제거)되는 308로 프로세싱이 진행된다. 예를들어, 제어 로직은 엔트리를 제거할 수 있거나 또는 어드레스(예를들어, 충분된 어드레스를 생성하기 위하여 충분되는 어드레스)를 포함하는 캐시 라인과 상이한 데이터 캐시의 캐시 라인에 충분된 어드레스가 존재할 것이라는 결정에 기초하여 테이블의 엔트리가 무효인 것을 표시할 수 있다. 대안적으로, 데이터 캐시의 다음 액세스가 웨이에 액세스할 것이라는 예측이 수행될 때, 테이블의 엔트리가 유지되는 310으로 프로세싱이 진행한다. 예를들어, 제어 로직은 충분된 어드레스가 어드레스와 동일한 데이터 어레이의 캐시 라인에 있다는 결정(예를들어, 예측)에 기초하여 웨이 예측을 제공하기 위하여 테이블의 엔트리를 유효로 유지할 수 있다. 제어 로직 또는 제어 로직과 다른 로직은 레지스터 위치를 포함하는 레지스터 파일을 모니터링 (예를들어 추적)할 수 있다. 레지스터 파일을 모니터링 (예를들어, 추적)하는 결과로서, 제어 로직은 레지스터 위치의 콘텐츠들이 변경되는 것에 응답하여 엔트리를 무효화하거나 또는 삭제할 수 있다. 레지스터 위치의 콘텐츠들은 다른 명령에 의해 변경될 수 있다.

[0050]

[0055] 테이블에 액세스함으로써, 명령에 기초하여 액세스되는 데이터 캐시의 이전 웨이는 명령의 실행을 위한 웨이 예측으로서 사용될 수 있다. 부가적으로, 제어 로직은 명령의 후속(예를들어, 다음) 실행이 명령의 실행과 동일한 캐시 라인에 액세스할 것인지의 여부를 결정(예를들어, 예측)할 수 있다. 후속 실행이 동일한 캐시 라인에 액세스할 것인지의 결정에 기초하여, 제어 로직은 엔트리를 제거하고, 엔트리의 하나 이상의 필드들을 업데이트하며 그리고/또는 엔트리의 하나 이상의 필드들을 유지할 수 있다. 테이블의 엔트리들을 업데이트 및 유지함으로써, 하나 이상의 웨이 예측들을 수행하고 예측실패들을 방지하기 위하여 테이블이 사용될 수 있고 필요할 수 있다.

[0051]

[0056] 도 4를 참조하면, 데이터 캐시의 로우(400)의 특정한 예시적인 실시예가 도시된다. 예를들어, 데이터 캐시는 도 1의 데이터 캐시(102)를 포함할 수 있다. 로우(400)는 제 1 캐시 라인 A(402), 제 2 캐시 라인 B(404), 제 3 캐시 라인 C(406) 및 제 4 캐시 라인 D(408)를 포함할 수 있으며, 이를 각각의 라인은 캐시 라인 경계들(410-414)에 의해 분리된다. 예를들어, 4개의 캐시 라인들(402-408)은 도 1의 캐시 라인들(120a-d)에 대응할 수 있다. 비록 4개의 대표적인 캐시 라인들(A-D)이 도시될지라도, 로우(400)가 4개를 초과하는 캐시 라인들 또는 4개 미만의 캐시 라인들을 포함할 수 있다는 것이 이해되어야 한다. 캐시 라인들(402-408) 각각은 복수의 세그먼트들을 포함할 수 있다. 예를들어, 제 1 캐시 라인 A(402)는 제 1 세그먼트(402a), 제 2 세그먼트(402b), 제 3 세그먼트(402c) 및 제 4 세그먼트(402d)를 포함한다. 특정 실시예에서, 각각의 캐시 라인(402-408)은 동일한 개수의 세그먼트들을 포함한다. 캐시 라인들(402-408) 각각은 대응 웨이와 연관될 수 있다.

[0052]

[0057] 로우(400)의 동작 및 사용을 예시하기 위하여, 대표적인 프로그램 루프(예를들어, 루프 코드(430))를 포함하는 대표적인 컴퓨터 명령들의 예시적인 실시예가 도 4에 도시된다. 명령들은 루프 상부 식별자(440)에서 시작하는 루프 코드(430)를 포함한다. 루프는 3개의 명령들(442, 444 및 446)을 포함한다. 루프는 종료 루프 지정자(448)에서 종료한다. 단순화된 예를 제공하기 위하여 프로그램 루프의 모든 양상들이 도시된 것은 아니

다. 예를들어, 루프 종료 조건 및 루프 반복들의 수는 간략화를 위하여 생략되었다.

[0053] 제 1 명령(442)은 자동 증분 어드레싱 모드(예를들어, 사후-증분 로드)를 포함하는 예시적인 로드 타입 명령이다. 특정 실시예에서, 제 1 명령(442)은 메모리 어드레스를 저장하는 레지스터 위치 R9에 액세스하고 레지스터 위치 R9의 메모리 어드레스를 사용하여, 데이터 캐시로부터의 메모리 어드레스에 대응하는 콘텐츠들(예를들어, 데이터)을 레지스트 위치 R1으로 로드하는 메모리 기록 명령이다. 레지스터 파일(도시안됨)은 복수의 레지스터 위치들을 포함할 수 있다. 레지스터 위치 R9에 의해 식별된 콘텐츠들이 레지스터 위치 R1에 로드된 이후에, 레지스터 위치 R9의 메모리 어드레스의 값은 2만큼 자동 증분된다(예를들어, 2의 사후증분). 따라서, 제 1 명령(442)은 2의 증분값을 가지고 베이스 레지스터 R9상에서 동작하는 것으로 간주될 수 있다. 레지스터 위치 R9의 콘텐츠들을 레지스터 위치 R1에 로드하기 위하여, 데이터 캐시의 특정 캐시 라인이 액세스될 수 있다. 특정 캐시 라인은 데이터 어레이의 특정 드라이버 및 특정 웨이와 연관된다.

[0054] 제 2 명령(444)은 대표적인 산술 명령이다. 특정 실시예에서, 제 2 명령(444)은 (대응하는 제 1 콘텐츠들(예를들어, 데이터)을 가진) 제 1 메모리 어드레스를 저장하는 레지스터 위치 R4 및 (대응하는 제 2 콘텐츠들을 가진) 제 2 메모리 어드레스를 저장하는 레지스터 위치 R5를 식별하는 추가 명령이다. 레지스터 위치 R4의 제 1 메모리 어드레스에 대응하는 제 1 콘텐츠들 및 레지스터 위치 R5의 제 2 메모리 어드레스에 대응하는 제 2 콘텐츠들은 함께 가산될 수 있으며, 이 합은 제 2 명령(444)에 기초하여 레지스터 위치 R9에 저장되는 제 3 메모리 어드레스에 대응하는 콘텐츠들(예를들어, 데이터)로서 저장될 수 있다.

[0055] 제 3 명령(446)은 자동 증분 어드레싱 모드를 포함하는 다른 로드 타입 명령을 포함할 수 있다. 예를들어, 제 3 명령(446)의 실행은 메모리 어드레스를 저장하는 레지스터 위치 R10(예를들어, 제 3 명령(446)의 베이스 레지스터)에 액세스하고 레지스터 위치 R10의 메모리 어드레스를 사용하여, 데이터 캐시로부터의 메모리 어드레스에 대응하는 콘텐츠들(데이터)을 레지스터 위치 R2에 로드할 수 있다. 콘텐츠들이 레지스터 위치 R2에 로드된 이후에, 레지스터 위치 R10의 메모리 어드레스의 값은 1만큼 증분될 수 있다(예를들어, 1의 사후-증분).

[0056] 루프 코드(430)를 실행하는 것은 한번 이상(예를들어, 한번 이상의 반복들로) 명령들(442, 444 및 446)을 실행하는 것을 포함한다. 루프 코드(430)의 제 1 반복 동안, 제 1 명령(442)이 실행된다. 제 1 명령(442)이 자동 증분 어드레싱 모드와 같은 하나 이상의 웨이 예측 특성들을 포함하기 때문에, 제어 로직(도시안됨)은 웨이 예측 테이블에서 제 1 명령(442)에 대응하는 엔트리를 생성할 수 있다. 예를들어, 도 1의 제어 로직(150) 또는 예시되지 않은 다른 제어 로직은 웨이 예측 테이블(152)에서 엔트리를 생성할 수 있다. 특정 실시예에서, 제어 로직은 웨이 예측이 제 1 명령(442)에 대응하는 엔트리를 포함하지 않는다는 결정 이후에 웨이 예측 테이블에서 엔트리를 생성할 수 있다.

[0057] 제 1 명령(442)이 자동 증분 어드레싱 모드를 포함하기 때문에, 제어 로직은 2의 증분값을 식별할 수 있다. 2의 증분값이 캐시 라인들(402-408)의 크기(예를들어, 4의 크기) 미만이기 때문에, 제어 로직은 액세스되는 다음 웨이가 동일한 캐시 라인에 대응할 것이라는 것(예를들어 동일한 캐시 라인내에 있을 것이라는 것)을 예측할 수 있다. 동일한 웨이가 다음 반복(예를들어, 제 1 명령(442)의 다음 실행) 동안 액세스될 것이라는 예측에 기초하여, 웨이는 제 1 명령(442)의 다음 실행 동안 웨이 예측으로서 식별될 수 있다.

[0058] 예시하자면, 루프 코드(430)의 제 1 반복 동안, 레지스터 위치 R9의 콘텐츠들은 제 3 캐시 라인 C(406)의 제 1(순차적인) 세그먼트(406a)를 가르킬 수 있다(450). 예를들어, 제 3 캐시 라인 C(406)는 4개의 세그먼트들, 예를들어 제 1 세그먼트(406a), 제 2 세그먼트(406b), 제 3 세그먼트(406c) 및 제 4 세그먼트(406d)를 포함할 수 있다. 따라서, 레지스터 위치 R9의 콘텐츠들을 2의 증분값 만큼 증분시키는 것은 레지스터 위치 R9의 콘텐츠들이 제 3 캐시 라인 C(406)의 제 3(순차적인) 세그먼트(406c)를 가르키도록 할 것이다(452). 따라서, 제 1 명령(442)에 대한 웨이 예측은 루프 코드(430)의 제 1 반복 동안 제 1 명령(442)의 실행 동안 사용되는(제 3 캐시 라인 C(406)에 대응하는) 웨이를 식별할 것이다. 따라서, 도 1과 관련하여 설명된 바와같이, 제 1 명령(442)에 대한 새로운 엔트리는 제어 로직(150)에 의해 웨이 예측 테이블(152)에 가산된다. 예를들어, 특정 엔트리는 하기의 필드들, 즉 PC = 0x10148(예를들어, 제 1 명령(442)에 대응함); WAY = 3; REG = R9; 및 V/I = 유효(예를들어, "1"의 데이터 값)중 하나 이상을 포함하도록 생성될 수 있다.

[0059] 제어 로직은 명령의 후속 실행이 동일한 캐시 라인에 액세스할 것이라는 결정에 기초하여 엔트리의 V/I 필드의 유효성 비트를(예를들어, 유효로 표시하도록) 세팅할 수 있다. 다른 실시예에서, 새로운 엔트리의 WAY 필드는 단지, 후속 실행이 동일한 캐시 라인에 액세스할 것이라는 결정(예를들어, 예측)이 수행될 때 액세스되는 웨이를 식별하기 위하여 패플레이트될 수 있다. 후속 실행이 동일한 캐시 라인에 액세스하지 않을 것이라는

예측이 수행될 때, 제어 로직은 새로운 엔트리의 WAY 필드를 널(null) 값(예를들어, 제로 값)으로 세팅할 수 있다. 다른 특정 실시예에서, 엔트리는 단지, 후속 실행이 동일한 캐시 라인에 액세스할 것이라는 예측이 수행될 때 웨이 예측 테이블에서 생성된다.

[0060] 명령에 기초하여 새로운 엔트리를 생성하는 것은 식별되는 하나 이상의 추가 웨이 예측 특성들(예를들어, 명령 데이터에 기초하여 도 1의 제어 로직(150)에 의해 수행되는 하나 이상의 추가 결정들)에 대하여 (예를들어 이러한 추가 웨이 예측 특성에 기초하여) 추가로 컨디셔닝될 수 있다. 예를들어, 새로운 엔트리는 단지, 자동 증분 어드레싱 모드 및/또는 베이스 플러스 오프셋 어드레싱 모드와 명령이 연관될 때 생성(예를들어, 패플레이트)될 수 있다. 다른 예로서, 새로운 엔트리는 단지, 명령이 루프에 포함될 때 생성(예를들어, 패플레이트)될 수 있다. 특히, 새로운 엔트리는 명령이 루프의 명령의 제 1 인스턴스일 때 생성될 수 있다. 특정 실시예에서는 명령이 루프에 포함되지 않을 때 엔트리가 생성되지 않는다.

[0061] 명령이 베이스 플러스 오프셋 어드레싱 모드를 사용할 때, 제어 로직(150)은 명령의 제 1 실행에 기초하여 명령의 후속 실행이 동일한 캐시 라인에 액세스할지에 관한 예측을 수행하도록 동작가능하지 않을 수 있다. 예를들어, 자동 증분 어드레싱 모드와 상이하게, 베이스 플러스 오프셋 어드레싱 모드를 실행하는 것은 명령의 각각의 연속 실행(예를들어, 루프의 각각의 연속 반복) 동안 미리 결정된 값(예를들어, 상수) 만큼 어드레스 위치를 충분시키지 않을 수 있다. 명령의 적어도 2개의 실행들은 베이스 플러스 오프셋 어드레싱 모드를 사용하여 명령의 스트라이드(stride)(예를들어, 오프셋)를 결정하는데 필요할 수 있다. 따라서, 명령이 베이스 플러스 오프셋 어드레싱 모드를 사용할 때, 새로운 엔트리는 명령의 제 1 실행 동안 생성될 수 있으나, 새로운 엔트리의 V/I 필드의 값은 명령의 제 2 실행(예를들어, 초기 실행 후 다음 실행) 때까지 WAY 필드의 값이 예측된 웨이로서 사용될 수 있다는 것을 표시하도록 세팅되지 않을 수 있다. 대안 실시예에서, 엔트리는 명령이 베이스 플러스 오프셋 어드레싱 모드를 사용할 때 명령의 제 1 실행에 기초하여 웨이 예측 테이블(152)에서 생성되지 않을 수 있다. 오히려, 명령의 제 1 실행에 기초하여, PC 필드와 연관된 값 및 WAY 필드와 연관된 값은 명령과 연관된 잠재적인 새로운 엔트리에 대해 식별될 수 있으며, PC 필드와 연관된 값 및 WAY 필드와 연관된 값은 웨이 예측 테이블(152)과 상이한 위치(및/또는 구조)에서 유지될 수 있다. 예를들어, 위치는 제어 로직(150)과 같은 제어 로직과 연관된 버퍼 또는 레지스터를 포함할 수 있다. 베이스 플러스 오프셋 어드레싱 모드를 사용하는 명령과 연관된 엔트리는 제 1 실행에 후속하는 명령의 실행에 기초하여 생성될 수 있다.

[0062] 특정 실시예에서, 제어 로직(150) 또는 다른 제어 로직은 웨이 예측 테이블(152)과 같은 웨이 예측 테이블내에서 레지스터로서 식별되는 (예를들어 REG 필드에서 식별되는) 레지스터 파일의 각각의 레지스터 위치(도시안됨)를 추적하는 추적 로직을 포함할 수 있다. 특정 실시예에서, 제어 로직(150) 또는 다른 제어 로직은 단지 유효 엔트리의 대응 REG 필드에서 식별되는 레지스터 파일의 레지스터 위치들을 추적한다. 예를들어, 레지스터 위치 R9가 제 1 명령(442)에 의해 사용되었고 레지스터 위치 R9가 웨이 예측 테이블(152)에 가산된 엔트리와 연관되었기 때문에, 추적 로직은 레지스터 위치 R9의 값을 수정하는 임의의 명령들을 모니터링할 것이다. 특정한 예시적인 예에서, 제 2 명령(444)은 레지스터 위치 R9의 값을 변경(예를들어, 수정)한다. 따라서, 추적 로직은 제 2 명령(444)과 같은 하나 이상의 명령들을 모니터링할 수 있으며, 레지스터 위치 R9의 값이 제 2 명령(444)에 의해 변경되었다는 것을 검출하는 것에 응답하여 제 1 명령(442)에 대응하는 웨이 예측 테이블(152)에서 엔트리를 무효화하거나(예를들어, V/I 필드를 무효로 세팅하거나) 또는 삭제(예를들어, 제거)할 수 있다. 따라서, 제 1 명령(442)의 후속 실행(예를들어, 다음 실행)시에, 웨이 예측 테이블(152)은 웨이 예측이 수행될 수 있게 하는 제 1 명령(442)과 연관된 유효 엔트리(또는 임의의 엔트리)를 포함하지 않을 것이다.

[0063] 루프 코드(430)로 진행할 때, 제 3 명령(446)은 자동 증분 어드레싱 모드를 포함할 수 있다. 제어 로직은 제 3 명령(446)의 레지스터 위치 R10(예를들어, 베이스 레지스터) 및 1의 증분값을 식별할 수 있다. 루프 코드(430)의 제 1 반복 동안, 레지스터 위치 R10의 콘텐츠들은 제 1 캐시 라인 A(402)의 제 1 (순차적인) 세그먼트(402a)를 가르킬 수 있다(420). 1의 증분값이 제 1 캐시 라인 A(402)의 크기(예를들어, 제 1 캐시 라인 A(402)는 4의 크기를 포함함) 미만이기 때문에, 제어 로직은 다음 웨이가 제 2(순차적인) 세그먼트(402b)에서 동일한 캐시 라인(402)내에 있을 것이라는 것을 예측할 수 있다(422). 동일한 웨이가 액세스될 것이라는 예측에 기초하여, 웨이는 제 3 명령(446)의 다음 실행 동안 웨이 예측으로서 식별될 수 있다. 따라서, 도 1과 관련하여 설명된 바와같이, 제 3 명령(446)에 대한 새로운 엔트리는 제어 로직(150)에 의해 웨이 예측 테이블(152)에 가산된다. 루프의 제 1 반복은 종료 루프 지정자(448)에서 종료한다.

[0064] 루프 코드(430)의 제 2 반복 동안, 제 1 명령(442)은 제 2 시간에 실행될 수 있다. 제어 로직(150)은 제 1 명령(442)에 대응하는 유효 엔트리에 대하여 웨이 예측 테이블(152)을 탐색할 수 있다. 엔트리가 루프 코드(430)의 제 1 반복 동안 웨이 예측 테이블(152)내에서 생성되어 저장되었기 때문에, 웨이 예측 테이블(152)은

제 1 명령(442)에 대응하는 프로그램 카운터 값과 연관된 PC 필드 값을 포함하는 엔트리를 가진다. 그러나, 추적 로직이 엔트리를 무효화하였기 때문에, 제 1 명령(442)에 기초한, 웨이 예측 테이블(152)의 룩업의 결과(예를들어 웨이 예측 테이블(152)의 엔트리를 판독하는 것)는 무효 엔트리의 표시일 것이다. 무효 엔트리는 제어 로직(150)이 웨이 예측 테이블(152)의 엔트리에 의해 표시된 WAY 필드의 값(예를들어, 웨이 예측)에만 의존하지 않을 수 있다. 따라서, 제어 로직(150)은 레지스터 위치 R9에 저장된 메모리 어드레스에 기초하여 태그 어레이(180)의 탐색(예를들어, 태그 룩업 동작)을 선택적으로 활성화(예를들어, 인에이블)할 것이다.

[0065] [0070] 특정 실시예에서, 제어 로직은 태그 룩업 동작을 동시에 (예를들어, 병렬로) 인에이블하면서 웨이 예측 테이블(152)의 엔트리에 표시된 WAY 필드의 값을 사용한다. 태그 룩업 동작을 동시에 인에이블하는 것은 WAY 필드의 값에 기초하여 예측실패(예를들어, 부정확한 웨이를 예측하는 것)가 발생하는지의 여부를 결정하기 위하여 제어 로직을 인에이블한다. 다른 실시예에서, 제어 로직은 모든 웨이들의 드라이버들을 동시에 (예를들어, 병렬로) 인에이블하여, 무효 엔트리에 대한 WAY 필드의 값에 의존하는 결과로서 예측실패로부터의 성능 패널티가 발생하지 않도록 태그 룩업 동작을 인에이블한다.

[0066] [0071] 루프 코드(430)를 통해 실행하는 것을 계속하면, 제 2 명령(444)에 대응하는 추가 동작이 다시 실행되고, 이후 프로세싱은 제 3 명령(446)을 실행하는 것으로 진행한다. 제 3 명령(446)이 자동 충분 어드레싱 모드를 포함하기 때문에, 제어 로직(150)은 웨이 예측 테이블(152)에 액세스(예를들어, 웨이 예측 테이블(152)을 판독)하며, (레지스터 위치 R0에 대응하는) 제 3 명령(446)과 연관된 이전에 저장된 엔트리를 식별한다. 이러한 경우에, 제 3 명령(446)과 연관된 엔트리는 유효하며, 제어 로직(150)은 웨이 선택 신호를 생성하고 (다른 웨이들 중 임의의 웨이를 활성화하지 않고) 선택된 (예를들어, 예측된) 웨이를 활성화하기 위하여 드라이버 인에이블(156)로부터 신호를 생성할 수 있다. 이러한 방식으로, 제 3 명령(446)의 제 2 실행을 수반하는 루프 코드(430)의 제 2 반복은 루프 코드(430)의 제 3 명령(446)의 제 1 실행 동안 사용되는 (이전에 액세스된 제 1 캐시 라인 A(402)에 대응하는) 이전에 저장된 웨이를 유리하게 선택한다. 따라서, 이전에 저장된 웨이는 웨이 예측으로서 사용될 수 있으며, 제어 로직(150)은 웨이 예측에 기초하여, 도 1의 복수의 드라이버들(140a-d)과 같은 복수의 드라이버들 중 단일 드라이버를 인에이블 (예를들어, 선택적으로 인에이블)할 수 있다. (예를들어, 모든 복수의 드라이버들(140a-d) 미만인) 단일 드라이버를 선택적으로 인에이블함으로써, 전력 이득은 도 1의 데이터 캐시(102)와 같은 데이터 캐시의 데이터 액세스 동안 실현될 수 있다.

[0067] [0072] 루프 코드(430)의 제 2 반복 동안, 레지스터 위치 R10의 콘텐츠들은 제 1 캐시 라인 A(402)의 제 2(순차적인) 세그먼트(402b)를 가르킬 수 있다(422). 제어 로직은, 레지스터 위치 R10의 콘텐츠들이, 1의 충분값 만큼 충분될 때, 루프 코드(430)의 제 3 반복동안 제 1 캐시 라인 A(402 및 424)의 제 3 (순차적인) 세그먼트(402c)를 가르키는 레지스터 위치 R10의 콘텐츠들이 될 것이라는 것을 계산함으로써 제 3 명령(446)과 연관된 다음 웨이가 동일한 캐시 라인(402)내에 있을 것이라는 것을 예측할 수 있다. 루프 코드(430)의 제 3 반복 동안 제 3 명령(446)의 예측된 웨이가 제 1 캐시 라인 A(402)내에 유지되기 때문에, 제 3 명령(446)과 연관된 엔트리의 WAY 필드의 값은 동일하게 유지될 수 있으며 (예를들어, 제 1 캐시 라인 A(402)에 대응하는 웨이), 제 3 명령(446)과 연관된 엔트리는 유효하게 유지된다. 종료 루프 지정자(448)의 프로세싱(예를들어, 실행)은 루프 코드(430)의 제 2 반복을 종료한다.

[0068] [0073] 루프 코드(430)는 앞서 설명된 추가 반복들을 통해 계속해서 프로세싱될 수 있다. 예를들어, 루프 코드(430)는 제 3 반복 및 제 4 반복을 거칠 수 있다. 루프 코드(430)의 제 3 반복에서 제 3 명령(446)의 실행 동안, 레지스터 위치 R10는 제 1 캐시 라인 A(402)의 제 3 (순차적인) 세그먼트(402c)를 가르킬 수 있다(424). 루프 코드(430)의 제 4 반복에서 제 3 명령(446)의 실행 동안, 레지스터 위치 R10는 제 1 캐시 라인 A(402)의 제 4 (순차적인) 세그먼트(402d)를 가르킬 수 있다(426). 루프 코드(430)의 제 4 반복에서 제 3 명령(446)의 실행 동안, 제어 로직은, 레지스터 위치 R10의 콘텐츠들이, 1의 충분값 만큼 충분될 때, 제 2 캐시 라인 B(404)를 가르키는 레지스터 위치 R10의 콘텐츠들이 될 것이라는 것을 계산함으로써, (예를들어, 루프 코드(430)의 제 5 반복동안) 제 3 명령(446)과 연관된 다음 웨이가 동일한 캐시 라인 A(402)내에 있지 않을 것이라는 것(예를들어, 경계(410)를 넘을 것이라는 것)을 예측할 수 있다(428). (제 3 명령(446)의 다음 실행과 연관된) 제 3 명령(446)의 예측된 어드레스가 제 1 캐시 라인 A(402) 밖에 있기 때문에, 제어 로직은 엔트리의 웨이 예측을 무효화하거나 또는 웨이 예측 테이블로부터 제 3 명령(446)과 연관된 엔트리를 삭제할 수 있다. 제어 로직이 엔트리를 무효화할 때, 엔트리는 루프 코드(430)의 제 5 반복 동안 새로운 (유효) 웨이 예측으로 업데이트될 수 있다. 대안적으로, 제어 로직이 엔트리를 삭제할 때, 새로운 엔트리는 제 5 반복 동안 생성될 수 있다.

[0069] [0074] 웨이 예측 테이블에서 하나 이상의 명령들에 대한 엔트리들을 생성(예를들어, 파풀레이트)하여 유지함으로써, 프로세서 시스템은 데이터 캐시에 대하여 웨이 예측들(예를들어, 웨이 예측 기술)을 구현(예를들어,

수행)하도록 인에이블될 수 있다. 데이터 캐시에 대하여 웨이 예측을 수행하는 것은 프로세서 시스템이 데이터 캐시의 특정 데이터 액세스를 동안 전력 이득을 실현하도록 한다. 예를들어, 웨이 예측 기술은 하나 이상의 명령들이 루프의 부분으로서 실행되는 (예를들어, 여러번 실행되는) 예측가능한 액세스 패턴을 가질 때 활용될 수 있다. 이러한 명령들은 자동 충분 어드레싱 모드 또는 베이스 플러스 오프셋 어드레싱 모드를 사용하는 명령들을 포함할 수 있다.

[0070] [0075] 도 5를 참조하면, 데이터 캐시와 연관된 웨이 예측을 수행하기 위한 방법(500)의 제 3 예시적인 실시예의 흐름도가 예시된다. 예를들어, 데이터 캐시는 도 1의 데이터 캐시(102)를 포함할 수 있다. 특정 실시예에서, 방법(500)은 도 1의 제어 로직(150)에 의해 수행될 수 있다.

[0071] [0076] 명령의 하나 이상의 웨이 예측 특성들이 식별된다(502). 하나 이상의 웨이 예측 특성들은 명령의 어드레싱 모드, 명령의 명령 타입, 명령이 루프에 포함되는지에 대한 표시, 또는 이들의 조합을 포함할 수 있다. 예를들어, 하나 이상의 웨이 예측 특성들은 도 1의 제어 로직(150)과 같은 제어 로직에 의해 또는 도 1의 디코드 로직(190)과 같은 디코드 로직에 의해 식별될 수 있다. 특정 실시예에서, 명령의 어드레싱 모드가 자동 충분 어드레싱 모드 또는 베이스 플러스 오프셋 어드레싱 모드인지에 대한 결정이 수행된다. 다른 특정 실시예에서, 명령의 명령 타입이 로드 타입인 또는 저장 타입인지에 대한 결정이 수행된다. 또 다른 특정 실시예에서, 명령이 하나 이상의 명령들의 루프에 포함되는지에 대한 결정이 수행된다. 디코드 로직은 명령 타입의 표시, 어드레싱 모드 또는 명령이 루프에 포함되는지의 여부를 제어 로직에 제공할 수 있다.

[0072] [0077] 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위하여 하나 이상의 웨이 예측 특성들의 식별에 기초하여 테이블이 선택적으로 판독된다(504). 제어 로직은 테이블이 명령에 대응하는 엔트리를 포함하는지의 여부를 결정하기 위하여 테이블을 판독할 수 있다. 예를들어, 도 1의 제어 로직(150)은 웨이 예측 테이블(152)을 선택적으로 판독할 수 있다. 테이블의 대응하는 엔트리는 엔트리에 포함된 하나 이상의 비트들의 값(예를들어, 엔트리의 WAY 필드의 값)에 기초하여 웨이(예를들어, 예측된 웨이)를 표시할 수 있다. 하나 이상의 비트들은 복수의 드라이버들의 각각의 드라이버를 선택적으로 인에이블 또는 디스에이블하기 위하여 복수의 드라이버들에 대해 마스크(mask)로서 적용될 수 있다. 제어 로직은 또한 엔트리가 유효한지의 여부를 결정할 수 있다. 특정 실시예에서, 예측된 웨이는 명령의 이전 실행에 기초하여 이전에 액세스된 웨이와 동일한 웨이이다. 예를들어, 제어 로직은 테이블로부터, 예측된 웨이를 식별 및 리트리브할 수 있으며, 엔트리가 유효할 때 하나 이상의 드라이버들을 선택적으로 인에이블 및/또는 디스에이블할 수 있다. 드라이버들(140a-d)과 같은 하나 이상의 드라이버들은 도 1의 데이터 캐시(102)와 같은 데이터 캐시에 포함될 수 있다.

[0073] [0078] 명령에 기초한 데이터 캐시의 다음 액세스가 웨이를 액세스할 것인지에 대한 예측이 수행된다(506). 예를들어, 제어 로직은 명령의 다음 실행이 실행과 동일한 캐시 라인 및 이에 따라 동일한 웨이에 액세스할 것인지의 여부를 예측(예를들어, 확인)하기 위한 산술 연산을 수행할 수 있다. 충분된 어드레스가 명령의 실행 동안 액세스되는 동일한 캐시 라인에 있지 않을 것이라는 결정(예를들어, 예측)이 수행될 때, 엔트리의 V/I 필드(예를들어 유효성 비트)는 WAY 필드의 값이 무효라는 것을 표시하도록 세팅될 수 있으며, 명령의 후속 실행 동안 사용될 예측된 웨이를 표시하는데 필요치 않을 수 있다. 충분된 어드레스가 동일한 캐시 라인에 있을 것이라는 결정이 수행될 때, 엔트리의 V/I 필드는 WAY 필드의 값이 유효하다는 것을 표시하도록 세팅될 수 있으며 이를 표시하는데 필요할 수 있다.

[0074] [0079] 테이블에 액세스함으로써, 명령에 기초하여 액세스된 데이터 캐시의 이전 웨이는 명령의 실행 동안 웨이 예측으로서 사용될 수 있다. 이전에 저장된 웨이는 웨이 예측으로 사용될 수 있으며, 하나 이상의 드라이버들은 웨이 예측에 기초하여 선택적으로 디스에이블(예를들어, 턴오프)될 수 있다. 하나 이상의 드라이버들을 선택적으로 디스에이블함으로써, 드라이버들 모두가 활성화(예를들어, 턴온)되는 것이 아니며, 전력 이득이 데이터 캐시의 데이터 액세스 동안 실현될 수 있다.

[0075] [0080] 도 2의 방법(200), 도 3의 방법(300), 도 5의 방법(500), 또는 이들의 조합은 필드 프로그램 가능 게이트 어레이(FPGA) 디바이스, 주문형 집적회로(ASIC), 프로세싱 유닛, 예를들어 중앙 프로세싱 유닛(CPU), 디지털 신호 프로세서(DSP), 제어기, 다른 하드웨어 디바이스, 펌웨어 디바이스 또는 이들의 임의의 조합에 의해 구현되거나 또는 그렇지 않은 경우에 수행될 수 있다. 예로서, 도 2의 방법(200), 도 3의 방법(300), 도 5의 방법(500) 또는 이들의 임의의 조합 중 임의의 것의 적어도 일부는 도 6과 관련하여 설명된 바와같이 메모리(632)에 저장된 명령들을 실행하는 프로세서(610)에 의해 구현될 수 있다.

[0076] [0081] 도 6은 멀티-비트 웨이 예측 마스크를 활용하는 캐시 메모리 시스템을 포함하는 디바이스(600)(예를들어, 통신 디바이스)의 특정 실시예의 블록도이다. 디바이스(600)는 무선 전자 디바이스일

수 있으며, 메모리(632)에 커플링된, 디지털 신호 프로세서(DSP)와 같은 프로세서(610)를 포함할 수 있다.

[0077] [0082] 프로세서(610)는 메모리(632)에 저장된 소프트웨어(660)(예를들어, 하나 이상의 명령들의 프로그램)을 실행하도록 구성될 수 있다. 프로세서(610)는 데이터 캐시(680) 및 제어 로직(686)을 포함할 수 있다. 예를들어, 데이터 캐시(680)는 도 1의 데이터 캐시(102)를 포함하거나 이에 대응할 수 있으며, 제어 로직(686)은 도 1의 제어 로직(150)을 포함하거나 또는 이에 대응할 수 있다. 데이터 캐시(680)는 데이터 어레이(682) 및 태그 어레이(684)를 포함할 수 있다. 데이터 어레이(682) 및 태그 어레이(684)는 각각 도 1의 데이터 어레이(110) 및 태그 어레이(180)에 대응할 수 있다. 데이터 어레이(682)는 도 1의 라인 드라이버들(140a-d)과 같은 복수의 라인 드라이버들을 포함할 수 있다. 제어 로직(686)은 웨이 예측 테이블(688)을 포함할 수 있다. 웨이 예측 테이블(688)은 도 1의 웨이 예측 테이블(152)을 포함할 수 있거나 또는 이에 대응할 수 있다. 예시적인 예에서, 프로세서(610)는 도 1의 프로세서 시스템(100) 또는 이의 컴포넌트들을 포함하거나 또는 이들에 대응할 수 있으며, 도 1-5의 실시예들 중 임의의 실시예 또는 이들의 임의의 조합에 따라 동작한다.

[0078] [0083] 특정 실시예에서, 프로세서(610)는 프로세서(610)와 같은 컴퓨터로 하여금 도 2의 방법(200), 도 3의 방법(300), 도 5의 방법(500) 또는 이들의 임의의 조합의 임의의 것의 적어도 일부분을 수행하도록 실행가능한, 메모리(632)와 같은 비-일시적 컴퓨터-판독가능 매체에 저장된 컴퓨터 실행가능 명령들(660)을 실행하도록 구성될 수 있다. 예를들어, 컴퓨터 실행가능 명령들(660)은 프로세서(610)로 하여금 명령의 하나 이상의 웨이 예측 특성들을 식별하도록 실행가능할 수 있다. 컴퓨터 실행가능 명령들(660)은 프로세서(610)로 하여금, 하나 이상의 웨이 예측 특성들의 식별에 기초하여, 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위하여 테이블을 선택적으로 판독하도록 추가로 실행가능하다. 컴퓨터 실행가능 명령들(660)은 프로세서(610)로 하여금, 명령에 기초한 데이터 캐시의 다음 액세스가 웨이에 액세스할지에 대한 예측을 수행하도록 추가로 실행가능하다.

[0079] [0084] 카메라 인터페이스(668)는 프로세서(610)에 커플링되며, 또한 비디오 카메라(670)와 같은 카메라에 커플링된다. 디스플레이 제어기(626)는 프로세서(610) 및 디스플레이 디바이스(628)에 커플링된다. 코더/디코더(CODEC)(634)는 또한 프로세서(610)에 커플링될 수 있다. 스피커(636) 및 마이크로폰(638)은 CODEC(634)에 커플링될 수 있다. 무선 인터페이스(640)는 안테나(642) 및 무선 인터페이스(640)를 통해 수신된 무선 데이터가 프로세서(610)에 제공될 수 있도록 프로세서(610) 및 안테나(642)에 커플링될 수 있다. 특정 실시예에서, 프로세서(610), 디스플레이 제어기(626), 메모리(632), CODEC(634), 무선 인터페이스(640) 및 카메라 인터페이스(668)는 시스템-인-패키지 또는 시스템-온-칩 디바이스(622)에 포함된다. 특정 실시예에서, 입력 디바이스(630) 및 전원(644)은 시스템-온-칩 디바이스(622)에 커플링된다. 더욱이, 특정 실시예에서, 도 6에 예시된 바와같이, 디스플레이 디바이스(628), 입력 디바이스(630), 스피커(636), 마이크로폰(638), 무선 안테나(642), 비디오 카메라(670), 및 전원(644)은 시스템-온-칩-디바이스(622) 외부에 있다. 그러나, 디스플레이 디바이스(628), 입력 디바이스(630), 스피커(636), 마이크로폰(638), 무선 안테나(642), 비디오 카메라(670), 및 전원(644)은 각각 인터페이스 또는 제어기와 같은, 시스템-온-칩 디바이스(622)의 컴포넌트에 커플링될 수 있다.

[0080] [0085] 설명된 실시예들 중 하나 이상의 실시예와 함께, 명령의 하나 이상의 웨이 예측 특성들을 식별하기 위한 수단을 포함하는 장치가 개시된다. 식별하기 위한 수단은 도 1의 제어 로직(150), 디코드 로직(190), 도 6의 프로세서(610), 제어 로직(686), 하나 이상의 웨이 예측 특성들을 식별하도록 구성된 하나 이상의 다른 디바이스들 또는 회로들, 또는 이들의 임의의 조합을 포함할 수 있다.

[0081] [0086] 장치는 또한 하나 이상의 웨이 예측 특성들의 식별에 기초하여, 데이터 캐시의 웨이를 식별하는, 명령과 연관된 테이블의 엔트리를 식별하기 위하여 테이블을 선택적으로 판독하기 위한 수단을 포함할 수 있다. 테이블을 선택적으로 판독하기 위한 수단은 도 1의 제어 로직(150), 도 6의 제어 로직(686), 테이블을 선택적으로 판독하도록 구성된 하나 이상의 다른 디바이스들 또는 회로들 또는 이들의 임의의 조합을 포함할 수 있다.

[0082] [0087] 장치는 또한 명령에 기초한 데이터 캐시의 다음 액세스가 웨이에 액세스할지에 대한 예측을 수행하기 위한 수단을 포함할 수 있다. 예측을 수행하기 위한 수단은 도 1의 제어 로직(150), 도 6의 제어 로직(686), 예측을 수행하도록 구성된 하나 이상의 다른 디바이스들 또는 회로들 또는 이들의 임의의 조합을 포함할 수 있다.

[0083] [0088] 장치는 또한 레지스터 식별자를 포함하고 예측가능한 다음 어드레스를 가진 명령을 디코딩하기 위한 수단을 포함할 수 있다. 디코딩하기 위한 수단은 도 1의 디코드 로직(190), 도 6의 프로세서(610), 실행될 명령을 디코딩하도록 구성된 하나 이상의 다른 디바이스들 또는 회로들 또는 이들의 임의의 조합을 포함할 수 있다.

[0084]

[0089] 장치는 또한 웨이에 기초하여 데이터 캐시 라인을 선택적으로 구동하기 위한 수단을 포함한다. 데이터 캐시 라인을 선택적으로 구동하기 위한 수단은 도 1의 라인 드라이버들(140a-c), 도 6의 데이터 어레이(682), 데이터 캐시 라인을 선택적으로 구동하도록 구성된 하나 이상의 다른 디바이스들 또는 회로들, 또는 이들의 임의의 조합을 포함할 수 있다.

[0085]

[0090] 개시된 실시예들 중 하나 이상은 디바이스(600)와 같은 시스템 또는 장치에서 구현될 수 있으며, 이 시스템 또는 장치는 모바일 폰, 셀룰러 폰, 위성 폰, 컴퓨터, 셋톱 박스, 엔터테인먼트 유닛, 내비게이션 디바이스, 통신 디바이스, 개인 휴대 단말(PDA), 고정 위치 데이터 유닛, 모바일 위치 데이터 유닛, 태블릿, 휴대용 컴퓨터, 데스크탑 컴퓨터, 모니터, 컴퓨터 모니터, 텔레비전, 튜너, 라디오, 위성 라디오, 뮤직 플레이어, 디지털 뮤직 플레이어, 휴대용 뮤직 플레이어, 비디오 플레이어, 디지털 비디오 플레이어, 디지털 비디오 디스크(DVD) 플레이어, 휴대용 디지털 비디오 플레이어 또는 이들의 조합을 포함할 수 있다. 또 다른 예시적인 비제한적인 예로서, 시스템 또는 장치는 원격 유닛들, 예를 들어 모바일 폰들, 핸드-헬드 개인 통신 시스템(PCS) 유닛들, 휴대용 데이터 유닛들, 예를 들어 개인 휴대 단말들, 글로벌 포지셔닝 시스템(GPS) 인에이블 디바이스들, 내비게이션 디바이스들, 고정 위치 데이터 유닛들, 예를 들어 미터 판독 장비, 또는 데이터 또는 컴퓨터 명령들을 저장 또는 리트리브하는 임의의 다른 디바이스 또는 이들의 임의의 조합을 포함할 수 있다.

[0086]

[0091] 도 1-6 중 하나 이상이 개시내용의 교시들에 따른 시스템들, 장치들 및/또는 방법들을 예시할 수 있을지도, 개시내용은 이들 예시된 시스템들, 장치들 및/또는 방법들로 제한되지 않는다. 개시내용의 실시예들은 프로세서 및 메모리를 포함하는 집적회로를 포함하는 임의의 디바이스에서 적절하게 사용될 수 있다.

[0087]

[0092] 당업자는 본원에 개시된 실시예들과 관련하여 설명되는 다양한 예시적인 논리 블록들, 구성들, 모듈들, 회로들, 및 알고리즘 단계들이 전자 하드웨어, 프로세서에 의해 실행되는 컴퓨터 소프트웨어, 또는 이들의 조합으로서 구현될 수 있음을 추가로 인식할 것이다. 다양한 예시적인 컴포넌트들, 블록들, 구성들, 모듈들, 회로들, 및 단계들이 일반적으로 이들의 기능적 관점에서 전술되었다. 이러한 기능이 하드웨어로 구현되는지, 또는 프로세서 실행가능 명령들로 구현되는지는 특정 애플리케이션 및 전체 시스템에 대해 부과된 설계 제한들에 의존한다. 당업자들은 설명된 기능을 각각의 특정 애플리케이션에 대해 다양한 방식들로 구현할 수 있지만, 이러한 구현 결정들이 본 개시내용의 범위로부터 벗어나는 것으로 해석되어서는 안 된다.

[0088]

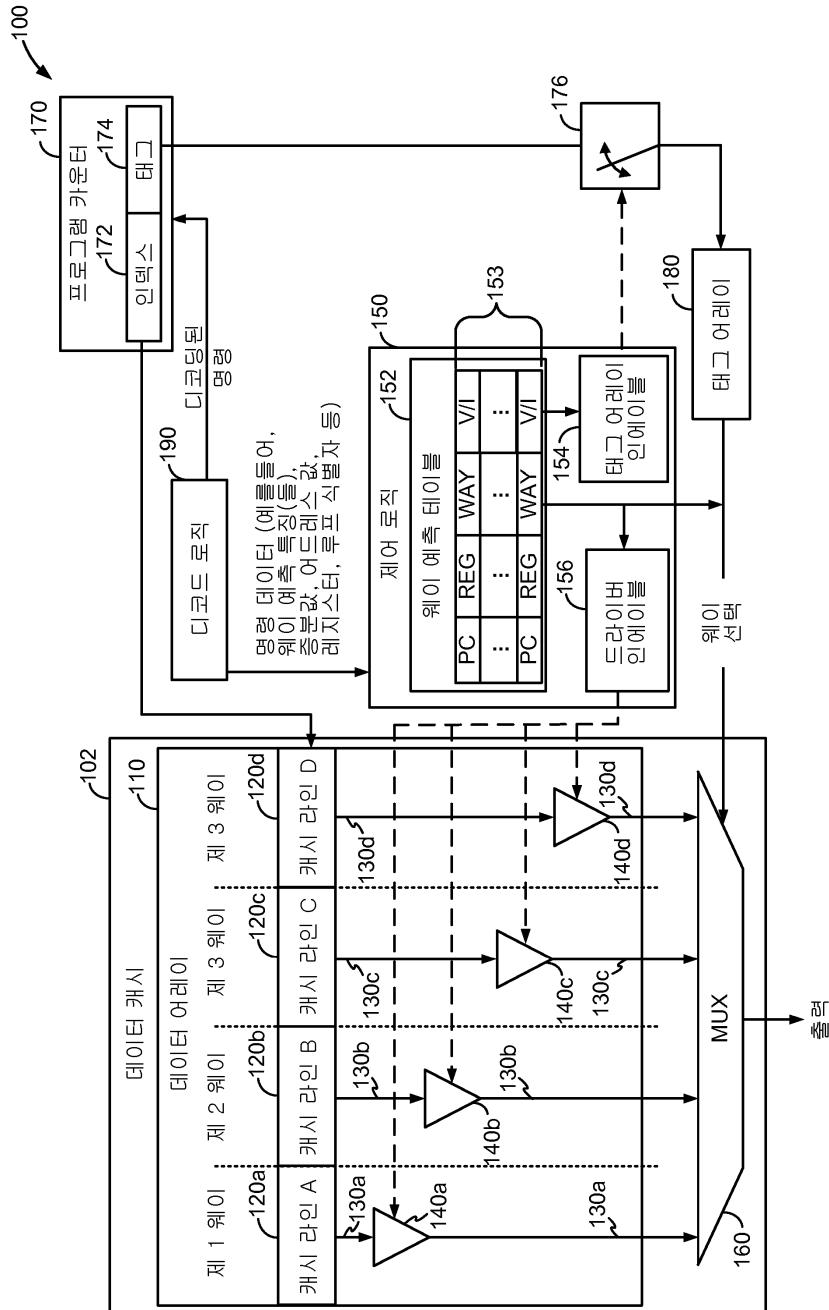
[0093] 본원에 개시된 실시예와 관련하여 설명된 알고리즘 또는 방법의 단계들은 직접적으로 하드웨어로, 프로세서에 의해 실행되는 소프트웨어 모듈로 또는 이들의 임의의 조합으로 구현될 수 있다. 소프트웨어 모듈은 랜덤 액세스 메모리(RAM), 플래쉬 메모리, 판독-전용 메모리(ROM), 프로그램가능 판독-전용 메모리(PROM), 소거가능 프로그램가능 판독-전용 메모리(EPROM), 전기적 소거가능 프로그램가능 판독-전용 메모리(EEPROM), 레지스터들, 하드 디스크, 착탈식 디스크, 컴팩트 디스크 판독-전용 메모리(CD-ROM) 또는 업계에 공지된 임의의 다른 형태의 비-일시적 저장 매체에 상주할 수 있다. 예시적인 저장 매체는, 프로세서가 저장 매체로부터 정보를 판독하고 저장 매체에 정보를 기록할 수 있도록 프로세서에 커플링된다. 대안적으로, 저장 매체는 프로세서에 통합될 수 있다. 프로세서 및 저장 매체는 주문형 집적회로(ASIC)에 상주할 수 있다. ASIC는 컴퓨팅 디바이스 또는 사용자 단말에 상주할 수 있다. 대안적으로, 프로세서 및 저장 매체는 컴퓨팅 디바이스 또는 사용자 단말에서 개별 컴포넌트들로서 존재할 수 있다.

[0089]

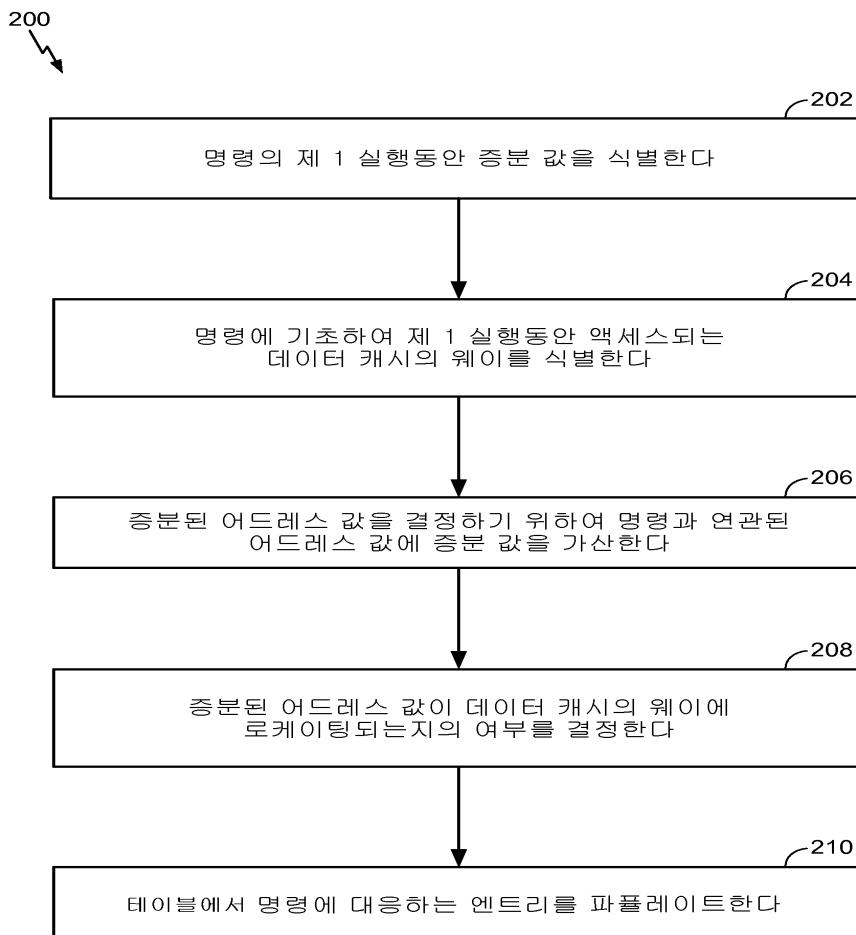
[0094] 개시된 실시예들의 이전 설명은 당업자가 개시된 실시예들을 이용하거나 또는 실시할 수 있도록 제공된다. 이들 실시예들에 대한 다양한 수정들은 당업자들에게 쉽게 명백할 것이며, 본원에서 정의된 원리들은 본 개시내용의 범위로부터 벗어남이 없이 다른 실시예들에 적용될 수 있다. 따라서, 본 개시내용은 본원에서 제시된 실시예들에 제한되는 것으로 의도되는 것이 아니라, 하기의 청구범위에 의해 정의된 원리들 및 신규한 특징들에 가능한 부합하는 가장 넓은 범위에 따른다.

## 도면

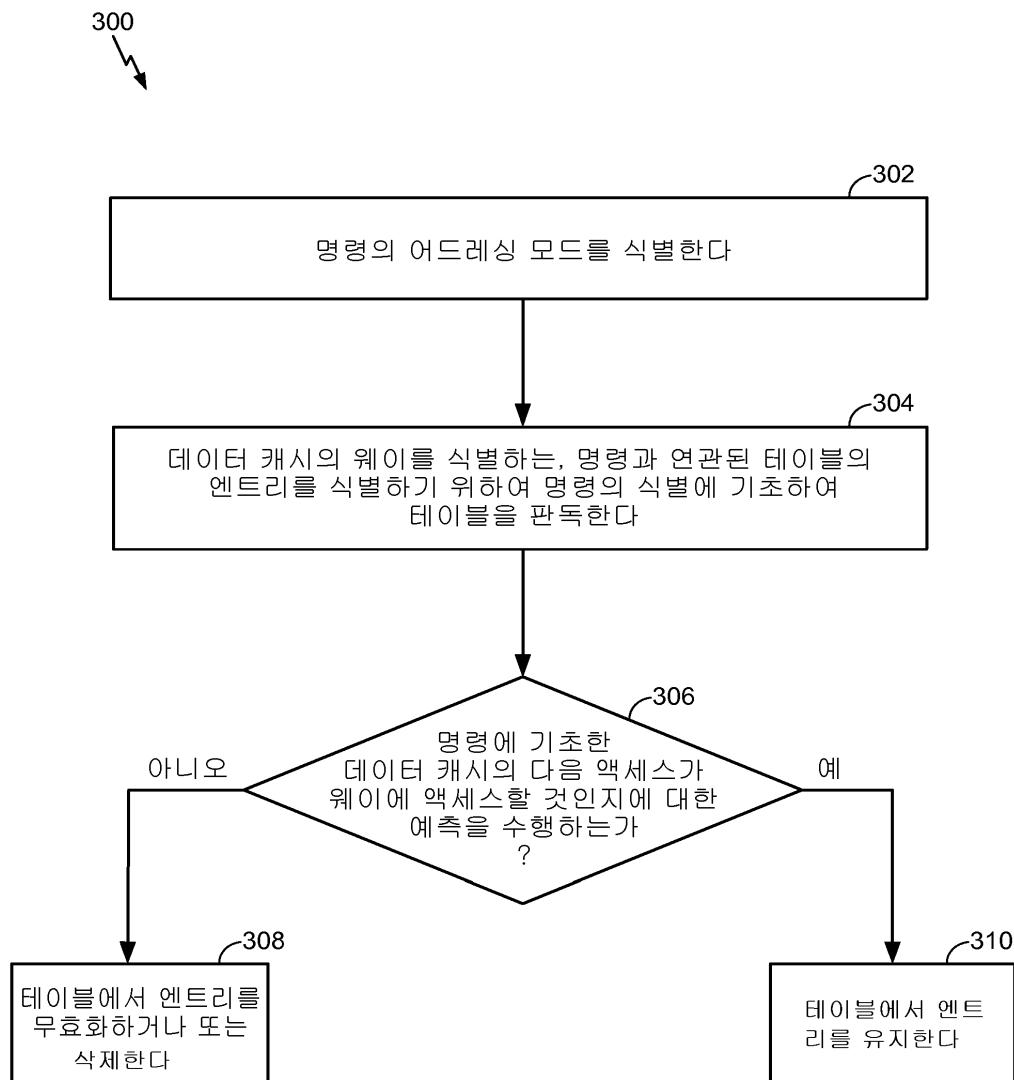
## 도면1



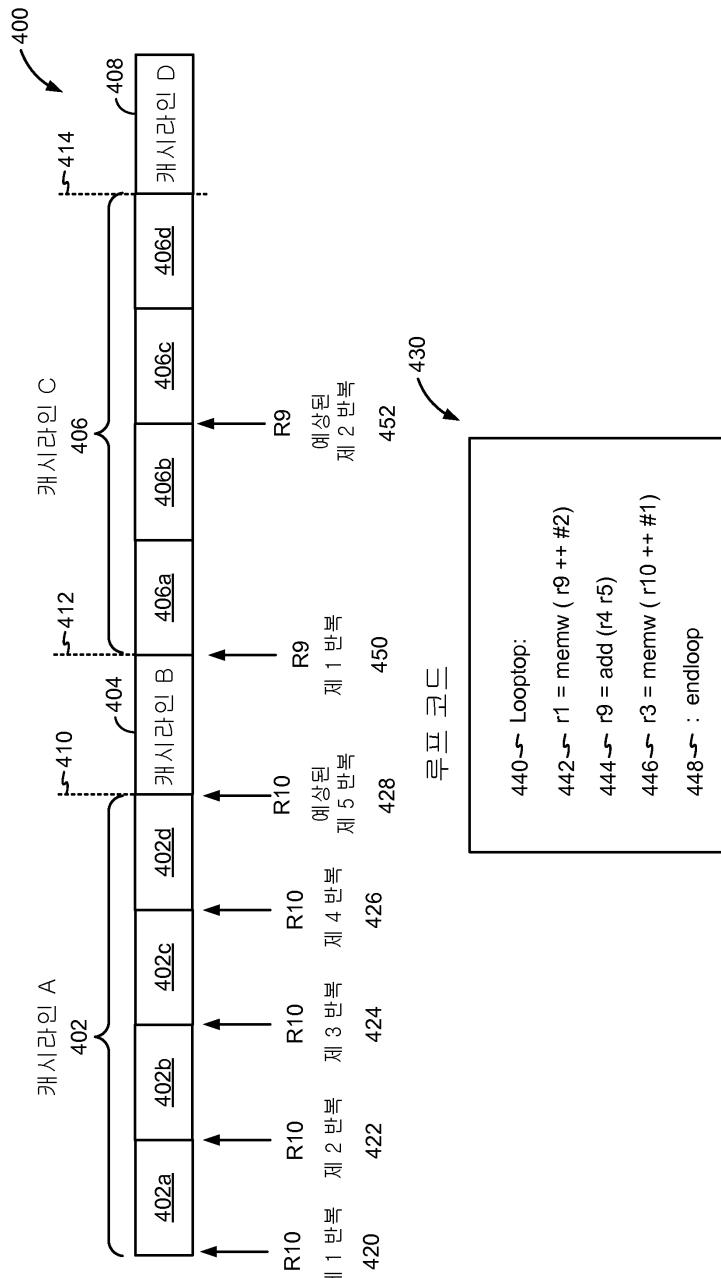
## 도면2



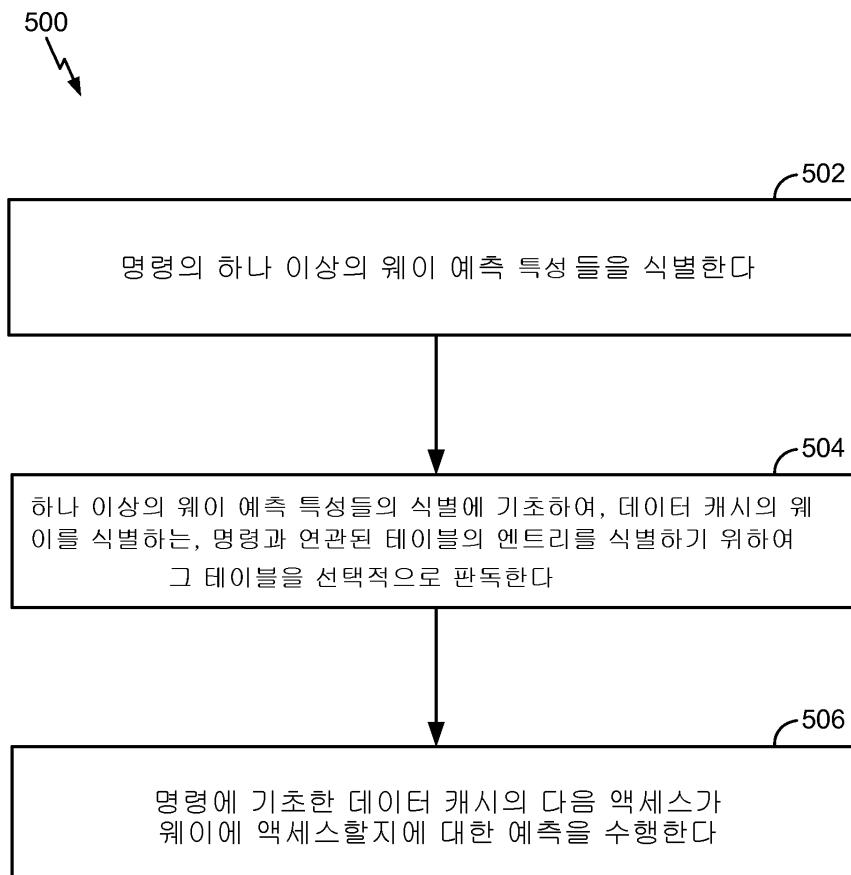
## 도면3



## 도면4



## 도면5



도면6

