



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0043631
(43) 공개일자 2021년04월21일

(51) 국제특허분류(Int. Cl.) G06F 9/38 (2006.01)	(71) 출원인 어드밴스드 마이크로 디바이시스, 인코포레이티드 미국 캘리포니아 95054 산타 클라라 어거스틴 드 라이브 2485
(52) CPC특허분류 G06F 9/3806 (2013.01) G06F 9/3814 (2013.01)	(72) 발명자 야라바르티 아디시아 미국 캘리포니아 95054 산타 클라라 어거스틴 드 라이브 2485
(21) 출원번호 10-2021-7007316	(74) 대리인 박장원
(22) 출원일자(국제) 2019년06월18일 심사청구일자 없음	
(85) 번역문제출일자 2021년03월10일	
(86) 국제출원번호 PCT/US2019/037804	
(87) 국제공개번호 WO 2020/055471 국제공개일자 2020년03월19일	
(30) 우선권주장 16/127,093 2018년09월10일 미국(US)	

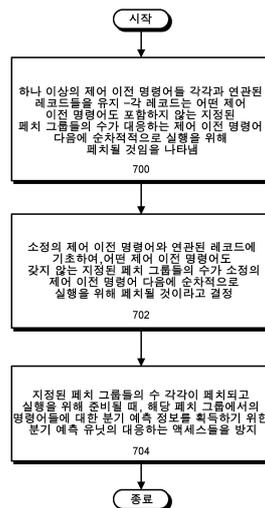
전체 청구항 수 : 총 22 항

(54) 발명의 명칭 **폐치 그룹의 순차에 대한 분기 예측 유닛에 대한 액세스 제어**

(57) 요약

프로그램 코드에서의 명령어들을 실행할 때 제어 이전 명령어들(CTI들, control transfer instructions)을 핸들링하는 전자 디바이스가 설명된다. 전자 디바이스는 분기 예측 기능 블록 및 순차적 폐치 로직 기능 블록을 포함하는 프로세서를 가진다. 순차적 폐치 로직 기능 블록은 CTI와 연관된 레코드에 기초하여, 사전에 CTI를 포함하지 않는 것으로 결정되었던 명령어들의 지정된 폐치 그룹들의 수가 CTI 다음에 순차적으로 실행을 위해 폐치될 것이라고 결정한다. 지정된 폐치 그룹들의 수 각각이 폐치되고 실행을 위해 준비될 때, 순차적 폐치 로직은 해당 폐치 그룹에서의 명령어들에 대한 분기 예측 정보를 획득하기 위한 상기 분기 예측 기능 블록의 대응하는 액세스들을 방지한다.

대표도 - 도7



(52) CPC특허분류
G06F 9/3842 (2013.01)

명세서

청구범위

청구항 1

프로그램 코드에서의 명령어들을 실행할 때 제어 이전 명령어들(CTI들, control transfer instructions)을 핸들링하는 전자 디바이스에 있어서,

프로세서로서:

분기 예측 기능 블록; 및

순차적 페치 로직 기능 블록;을 포함하는, 상기 프로세서를 포함하며,

상기 순차적 페치 로직 기능 블록은:

CTI와 연관된 레코드에 기초하여, 사전에 CTI들을 포함하지 않는 것으로 결정되었던 명령어들의 지정된 페치 그룹들의 수가 상기 CTI 다음에 순차적으로 실행을 위해 페치될 것이라고 결정하고;

상기 지정된 페치 그룹들의 수 각각이 페치되고 실행을 위해 준비될 때, 해당 페치 그룹에서의 명령어들에 대한 분기 예측 정보를 획득하기 위한 상기 분기 예측 기능 블록의 대응하는 액세스들을 방지하는, 전자 디바이스.

청구항 2

제1항에 있어서, 상기 프로세서는:

상기 분기 예측 기능 블록과 연관된 분기 타겟 버퍼(BTB, branch target buffer);를 더 포함하며,

상기 순차적 페치 로직 기능 블록은:

지정된 유형의 CTI를 검출 시, 후속 CTI가 페기(retire)되기 전에 순차적으로 페기되는 페치 그룹들의 수의 카운트를 유지하고;

상기 CTI와 연관된 상기 BTB에서의 엔트리에, 상기 페치 그룹들의 수의 상기 카운트를 나타내는 레코드를 저장하는, 전자 디바이스.

청구항 3

제2항에 있어서, 상기 지정된 유형의 CTI는 정적 타겟 명령어를 갖는 무조건 CTI들 또는 이행되지 않는(non-taken) 경로가 정적 타겟 명령어로 향하는 조건부 CTI들을 포함하는 것인, 전자 디바이스.

청구항 4

제2항에 있어서, 상기 순차적 페치 로직 기능 블록은:

상기 CTI와 연관된 상기 BTB에서의 상기 엔트리에, 상기 페치 그룹들의 수의 상기 카운트가 임계 값보다 클 때에만 상기 페치 그룹들의 수의 상기 카운트를 나타내는 상기 레코드를 저장하는, 전자 디바이스.

청구항 5

제1항에 있어서, 상기 프로세서는:

순차적 페치 테이블(SFT, sequential fetch table);을 더 포함하며,

상기 순차적 페치 로직 기능 블록은:

CTI 페기를 위해 타겟 명령어의 페기 시, 후속 CTI가 페기되기 전에 순차적으로 페기되는 페치 그룹들의 수의 카운트를 유지하고;

상기 CTI에 대한 상기 타겟 명령어와 연관된 상기 SFT에서의 엔트리에, 상기 페치 그룹들의 수의 상기 카운트를 나타내는 레코드를 저장하는, 전자 디바이스.

청구항 6

제5항에 있어서, 상기 순차적 페치 로직 기능 블록은:

상기 CTI에 대한 상기 타겟 명령어와 연관된 상기 SFT에서의 상기 엔트리에, 상기 페치 그룹들의 수의 상기 카운트가 임계 값보다 클 때에만 상기 페치 그룹들의 수의 상기 카운트를 나타내는 상기 레코드를 저장하는, 전자 디바이스.

청구항 7

제1항에 있어서, 상기 지정된 페치 그룹들의 수 각각이 페치되고 실행을 위해 준비될 때, 분기 예측 정보를 획득하기 위한 상기 분기 예측 기능 블록의 대응하는 액세스들을 방지할 때, 상기 순차적 페치 로직 기능 블록은:

상기 지정된 페치 그룹들의 수와 동일하게 카운터를 설정하도록;

각 페치 그룹이 페치되고 실행을 위해 준비되며 상기 분기 예측 기능 블록의 상기 대응하는 액세스가 방지될 때, 상기 카운터를 감소시키도록; 그리고

상기 카운터가 제로에 도달한 후, 하나 이상의 후속 페치 그룹이 페치되고 실행을 위해 준비됨에 따라, 상기 하나 이상의 후속 페치 그룹에서의 명령어들에 대한 분기 예측 정보를 획득하기 위해 상기 분기 예측 기능 블록의 대응하는 액세스들이 수행되도록 허용하도록 구성되는, 전자 디바이스.

청구항 8

제1항에 있어서, 상기 분기 예측 기능 블록은:

분기 타겟 버퍼(BTB); 및

분기 방향 예측기;를 포함하며,

상기 분기 예측 기능 블록의 상기 액세스들을 방지할 때, 상기 순차적 페치 로직 기능 블록은 적어도 상기 BTB 및 상기 분기 방향 예측기의 액세스들을 방지하는, 전자 디바이스.

청구항 9

제1항에 있어서, 상기 지정된 페치 그룹들의 수 각각이 페치되고 실행을 위해 준비되는 동안:

CTI들과 연관된 레코드들을 저장하는 하나 이상의 기능 블록의 액세스들을 방지하고;

상기 하나 이상의 기능 블록을 절전 모드에 두는, 전자 디바이스.

청구항 10

제1항에 있어서, 각 페치 그룹은 상기 프로세서에서의 실행을 위해 준비되기 위해 동일한 이전 동작에서 페치되는 미리 결정된 수의 명령어를 포함하는, 전자 디바이스.

청구항 11

제1항에 있어서, 상기 분기 예측 기능 블록의 대응하는 액세스들을 방지하는 것은 전력을 소비하는 적어도 하나의 동작을 방지함으로써, 전력을 절감하는 것을 포함하는, 전자 디바이스.

청구항 12

분기 예측 기능 블록 및 순차적 페치 로직 기능 블록을 포함하는 프로세서를 갖는 전자 디바이스에서 프로그램 코드에서의 명령어들을 실행할 때 제어 이전 명령어들(CTI)을 핸들링하기 위한 방법으로서,

상기 순차적 페치 로직 기능 블록에 의해, CTI와 연관된 레코드에 기초하여, 사전에 CTI들을 포함하지 않는 것으로 결정되었던 명령어들의 지정된 페치 그룹들의 수가 상기 CTI 다음에 순차적으로 실행을 위해 페치될 것이라고 결정하는 단계; 및

상기 지정된 페치 그룹들의 수 각각이 페치되고 실행을 위해 준비될 때, 상기 순차적 페치 로직 기능 블록에 의해, 해당 페치 그룹에서의 명령어들에 대한 분기 예측 정보를 획득하기 위한 상기 분기 예측 기능 블록의 대응

하는 액세스들을 방지하는 단계를 포함하는, 방법.

청구항 13

제12항에 있어서, 상기 프로세서는 상기 분기 예측 기능 블록과 연관된 분기 타겟 버퍼(BTB)를 더 포함하고, 상기 방법은:

지정된 유형의 CTI를 검출 시, 상기 프로세서에 의해, 후속 CTI가 폐기되기 전에 순차적으로 폐기되는 폐치 그룹들의 수의 카운트를 유지하는 단계; 및

상기 순차적 폐치 로직 기능 블록에 의해, 상기 CTI와 연관된 상기 BTB에서의 엔트리에, 상기 폐치 그룹들의 수의 상기 카운트를 나타내는 레코드를 저장하는 단계를 더 포함하는, 방법.

청구항 14

제13항에 있어서, 상기 지정된 유형의 CTI는 정적 타겟 명령어를 갖는 무조건 CTI들 또는 이행되지 않는(non-taken) 경로가 정적 타겟 명령어로 향하는 조건부 CTI들을 포함하는 것인, 방법.

청구항 15

제13항에 있어서,

상기 CTI와 연관된 상기 BTB에서의 상기 엔트리에, 상기 폐치 그룹들의 수의 상기 카운트가 임계 값보다 클 때에만 상기 폐치 그룹들의 수의 상기 카운트를 나타내는 상기 레코드를 저장하는 단계를 더 포함하는, 방법.

청구항 16

제12항에 있어서, 상기 프로세서는 순차적 폐치 테이블(SFT)을 더 포함하고, 상기 방법은:

CTI에 대한 타겟 명령어의 폐기 시, 상기 프로세서에 의해, 후속 CTI가 폐기되기 전에 순차적으로 폐기되는 폐치 그룹들의 수의 카운트를 유지하는 단계; 및

상기 순차적 폐치 로직 기능 블록에 의해, 상기 CTI에 대한 상기 타겟 명령어와 연관된 상기 SFT에서의 엔트리에, 상기 폐치 그룹들의 수의 상기 카운트를 나타내는 레코드를 저장하는 단계를 더 포함하는, 방법.

청구항 17

제16항에 있어서,

상기 CTI에 대한 상기 타겟 명령어와 연관된 상기 SFT에서의 상기 엔트리에, 상기 폐치 그룹들의 수의 상기 카운트가 임계 값보다 클 때에만 상기 폐치 그룹들의 수의 상기 카운트를 나타내는 상기 레코드를 저장하는 단계를 더 포함하는, 방법.

청구항 18

제12항에 있어서, 상기 지정된 폐치 그룹들의 수 각각이 폐치되고 실행을 위해 준비될 때, 분기 예측 정보를 획득하기 위한 상기 분기 예측 기능 블록의 대응하는 액세스들을 방지하는 단계는:

상기 순차적 폐치 로직 기능 블록에 의해, 상기 지정된 폐치 그룹들의 수와 동일하게 카운터를 설정하는 단계;

각 폐치 그룹이 폐치되고 실행을 위해 준비되며 상기 분기 예측 기능 블록의 상기 대응하는 액세스가 방지될 때, 상기 순차적 폐치 로직 기능 블록에 의해, 상기 카운터를 감소시키는 단계; 및

상기 카운터가 제로에 도달한 후, 하나 이상의 후속 폐치 그룹이 폐치되고 실행을 위해 준비됨에 따라, 상기 순차적 폐치 로직 기능 블록에 의해, 상기 하나 이상의 후속 폐치 그룹에서의 명령어들에 대한 분기 예측 정보를 획득하기 위해 상기 분기 예측 기능 블록의 대응하는 액세스들이 수행되도록 허용하는 단계를 포함하는, 방법.

청구항 19

제12항에 있어서, 상기 분기 예측 기능 블록은 분기 타겟 버퍼(BTB) 및 분기 방향 예측기를 포함하고, 상기 분기 예측 기능 블록의 상기 액세스들을 방지하는 단계는 적어도 상기 BTB 및 상기 분기 방향 예측기의 액세스들을 방지하는 단계를 포함하는, 방법.

청구항 20

제12항에 있어서, 각 페치 그룹은 상기 프로세서에서 실행을 위해 준비되기 위해 동일한 이전 동작에서 페치되는 미리 결정된 수의 명령어를 포함하는, 방법.

청구항 21

제12항에 있어서, 상기 지정된 페치 그룹들의 수 각각이 페치되고 실행을 위해 준비되는 동안:

상기 순차적 페치 로직 기능 블록에 의해, CTI들과 연관된 레코드들을 저장하는 하나 이상의 기능 블록의 액세스들을 방지하는 단계; 및

상기 순차적 페치 로직 기능 블록에 의해, 상기 하나 이상의 기능 블록을 절전 모드에 두는 단계를 더 포함하는, 방법.

청구항 22

제12항에 있어서, 상기 분기 예측 기능 블록의 대응하는 액세스들을 방지하는 단계는 전력을 소비하는 적어도 하나의 동작을 방지함으로써, 전력을 절감하는 단계를 포함하는, 방법.

발명의 설명

기술 분야

[0001] 정부 권리

[0002] 본 발명은 미국 에너지국(DOE, Department of Energy)이 수여한 로렌스 리버모어 국립 연구소(주계약 번호 DE-AC52-07NA27344, 부계약 번호 B620717)에 따른 패스포워드 프로젝트(PathForward Project)에 따라 정부의 지원을 받아 만들어졌다. 정부는 본 발명에 특정한 권리를 가진다.

배경 기술

[0003] 전자 디바이스들에서의 많은 프로세서들(예를 들어, 마이크로 프로세서들)은 프로그램 코드에서의 명령어들을 실행하는 효율성을 개선하기 위한 동작들을 수행하는 기능 블록들을 포함한다. 예를 들어, 일부 프로세서들은 명령어들의 하나 이상의 이전 실행 인스턴스의 레코드들에 기초하여 명령어 실행의 경로들 또는 흐름들(즉, 명령어들이 실행을 위해 페치될 메모리에서의 어드레스 시퀀스들)을 예측하는 데 사용되는 예측 기능 블록들을 포함한다. 일반적인 예측 기능 블록 중 하나는 점프 및 리턴과 같은 프로그램 코드에서의 제어 이전 명령어들(CTI들, control transfer instructions)의 레졸루션을 예측하는 분기 예측 기능 블록이다. 분기 예측 기능 블록들은 CTI들이 실행될 때 CTI들의 거동, 이를테면 CTI들의 "이행되는(taken)" 또는 "이행되지 않는(not taken)" 레졸루션들, 이행되는 CTI들에 대한 타겟 명령어들 등을 모니터링하고 기록한다. 사전에 레코딩된 CTI들의 거동은 프로그램 코드를 실행하는 동안 CTI들을 다시 접할 시(예를 들어, 페치된 명령어들에서 CTI들을 검출할 시 등), CTI들의 현재 실행들의 레졸루션들을 예측하는 데 사용된다. 예측된 레졸루션들에 기초하여, 프로세서는 CTI 자체가 준비되고 실행되는 동안 CTI 이후에 예측된 경로를 따라 실행할 명령어들을 추론에 의해 페치하고 준비한다. 이러한 프로세서들은, 진행하기 전에 CTI들의 레졸루션들을 결정하기를 대기하고 CTI들로부터의 경로들의 고정된 선택들을 추론에 의해 따르는 프로세서들과 대조적으로, CTI들이 실행될 때 이어지는 경로들일 가능성이 더 큰 CTI들로부터의 경로들을 추론에 의해 따라, 지연을 낮추고/거나 복구 동작들을 줄일 수 있다.

[0004] 일부 프로세서들에서, 임의의 CTI들에 대한 예측된 레졸루션들이 프로그램 코드 실행 경로들을 지시하기 위해 가능한 한 빨리 이용 가능하도록 하기 위해 실행할 페치된 명령어들을 준비하는 프로세스의 초기에 모든 명령어들에 대해 분기 예측 기능 블록들이 자동으로 액세스된다. 그러나, CTI 명령어들은 통상적으로 프로그램 코드의 작은 부분만 형성하기 때문에, 분기 예측 기능 블록들의 많은 액세스들은 CTI들이 아닌 명령어들에 대한 것이다 - 그리고 이에 따라 동적 에너지를 낭비한다. 분기 예측 기능 블록의 각 액세스에 소비되는 전력 등의 관점에서 관련 비용을 가진다는 점을 감안할 때, 분기 예측 기능 블록의 불필요한 액세스를 방지하는 것이 바람직하다.

도면의 간단한 설명

[0005] 도 1은 일부 실시 예들에 따른 전자 디바이스를 도시한 블록도를 나타낸다.

- 도 2는 일부 실시 예들에 따른 프로세서를 도시한 블록도를 나타낸다.
 - 도 3은 일부 실시 예들에 따른 분기 예측 유닛을 도시한 블록도를 나타낸다.
 - 도 4는 일부 실시 예들에 따른 분기 타겟 버퍼를 도시한 블록도를 나타낸다.
 - 도 5는 일부 실시 예들에 따른 순차적 페치 로직을 도시한 블록도를 나타낸다.
 - 도 6은 일부 실시 예들에 따른 순차적 페치 테이블을 도시한 블록도를 나타낸다.
 - 도 7은 일부 실시 예들에 따라 분기 예측 유닛의 액세스들을 방지하기 위해 제어 이전 명령어들과 연관된 레코드들을 사용하기 위한 프로세스를 도시하는 흐름도를 나타낸다.
 - 도 8은 일부 실시 예들에 따라 레코드들을 저장하기 위해 분기 타겟 버퍼가 사용될 때 제어 이전 명령어들과 연관된 레코드들을 유지하기 위해 수행되는 동작들을 도시하는 흐름도를 나타낸다.
 - 도 9는 일부 실시 예들에 따라 대응하는 제어 이전 명령어들 다음에 순차적으로 페치되는 제어 이전 명령어들을 포함하지 않는 페치 그룹들의 카운트로 분기 타겟 버퍼를 업데이트하기 위한 동작들의 타임 라인 도해를 나타낸다.
 - 도 10은 일부 실시 예들에 따라 레코드들을 저장하기 위해 순차적 페치 테이블이 사용될 때 제어 이전 명령어들과 연관된 레코드들을 유지하기 위해 수행되는 동작들을 도시하는 흐름도를 나타낸다.
 - 도 11은 일부 실시 예들에 따라 대응하는 CTI들 다음에 순차적으로 페치되는 CTI들을 포함하지 않는 페치 그룹들의 카운트로 순차적 페치 테이블을 업데이트하기 위한 동작들의 타임 라인 도해를 나타낸다.
 - 도 12는 일부 실시 예들에 따라 분기 예측 유닛의 액세스들을 방지하기 위해 지정된 수를 사용하여 CTI 다음에 순차적으로 페치될 CTI들을 포함하지 않는 지정된 페치 그룹들의 수를 결정하기 위한 프로세스를 도시하는 흐름도를 나타낸다.
- 도면들 및 설명 전체에 걸쳐, 동일한 참조 부호들은 동일한 도면 요소들을 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0006] 이하의 설명은 당업자가 설명된 실시 예들을 만들고 사용할 수 있게 하기 위해 제시되며, 특정 적용 예 및 그 요건의 상황에서 제공된다. 설명된 실시 예들에 대한 다양한 변형이 당업자들에게 용이하게 명백할 것이고, 여기에 정의된 일반적인 원리들은 다른 실시 예들 및 적용 예들에 적용될 수 있다. 따라서, 설명된 실시 예들은 도시된 실시 예들에 제한되지 않고, 본 출원에 개시된 원리들 및 특징들과 일치하는 가장 넓은 범위에 따른다.
- [0007] **용어**
- [0008] 다음의 설명에서, 실시 예들을 설명하기 위해 다양한 용어들이 사용된다. 다음은 이러한 용어들 중 두 용어에 대한 간략하고 일반적인 설명이다. 용어들은 명확성과 간결성을 위해 여기에서 언급되지 않은 상당한 추가 양태들을 가질 수 있고, 이에 따라 설명은 용어들을 제한하려는 의도가 아니다.
- [0009] **기능 블록:** 기능 블록은 집적 회로 요소, 이산 회로 요소 등과 같은 하나 이상의 상호 관련된 회로 요소의 그룹, 모음 및/또는 집합을 나타낸다. 회로 요소들은 회로 요소들이 적어도 하나의 속성을 공유한다는 점에서 "상호 관련"된다. 예를 들어, 상호 관련된 회로 요소들은 특정 집적 회로 칩 또는 이의 부분에 포함되거나, 특정 집적 회로 칩 또는 이의 부분 상에서 제조되거나, 또는 특정 집적 회로 칩 또는 이의 부분에 다른 방식으로 결합될 수 있고, 소정의 기능들(컴퓨터 또는 처리 기능들, 메모리 기능들 등)의 수행에 수반될 수 있으며, 공통 제어 요소에 의해 제어될 수 있는 등이다. 기능 블록은 단일 회로 요소(예를 들어, 단일 집적 회로 로직 게이트)에서 수백만 또는 수십억 개의 회로 요소들(예를 들어, 집적 회로 메모리)에 이르기까지 임의의 수의 회로 요소들을 포함할 수 있다.
- [0010] **제어 이전 명령어:** 제어 이전 명령어(CTI)는 실행될 때, 다른 순차적 명령어 실행 흐름에서 점프, 변위 또는 불연속성을 유발하거나 유발할 수 있는 프로그램 코드에서의 명령어이다. CTI들은 점프, 호출, 반환 등과 같은 "무조건" CTI들을 포함하며, 이것들은 명령어 실행이 첫 번째 메모리 어드레스들에서의 명령어들, CTI들로부터, 두 번째 메모리 어드레스들에서의 명령어들, 또는 "타겟 명령어들"로 자동으로 점프하게 한다. CTI들은 또한 조건부 점프 명령어들 등과 같은 "조건부" CTI들을 포함하며, 이것들은 초과, 동일, 비제로 등의 조건들을 포함하거나, 조건들과 연관되거나, 또는 조건들에 의존한다. 대응하는 조건이 충족될 때(예를 들어, 참, 거짓 등), 조

건부 CTI는 CTI로부터 두 번째 메모리 어드레스에서의 명령어로 명령어 실행의 점프를 야기하지만, 조건이 충족되지 않을 때, 명령어 실행은 CTI 다음에 순차적으로 계속된다. 예를 들어, 조건부 분기 명령어는 조건 확인 명령어 및 조건부 CTI(또는 단일 조합된 명령어)를 사용하여 구현될 수 있으며, 조건이 충족될 때, 분기는 "이행되고(taken)," 명령어 실행이 타겟 명령어로 점프하고, 조건이 충족되지 않을 때 분기 명령어는 "이행되지 않거나(not taken)" 또는 "실행되지 못하고(falling through)," 명령어 실행은 순차적으로 계속된다. CTI들은 타겟 명령어들의 어드레스들이 실행 시간에 동적으로 지정되는 "간접" 무조건 및 조건부 CTI들을 포함한다. 예를 들어, CTI의 타겟 명령어의 어드레스는 이전 명령어에 의해 프로세서 레지스터 또는 다른 위치에서 컴퓨팅되고 저장될 수 있으며, 그 다음 CTI를 실행 시 명령어 실행이 점프할 어드레스를 결정하는 데 사용될 수 있다.

[0011] **개요**

[0012] 설명되는 실시 예들은 전자 디바이스에서의 프로세서를 포함한다. 프로세서는 컴퓨터 동작들, 메모리 동작들, 다른 기능 블록들 및 디바이스들과의 통신 등을 수행하는 중앙 처리 유닛(CPU) 코어, 하나 이상의 캐시 메모리, 통신 인터페이스들 등과 같은 기능 블록들을 포함한다. 프로세서는 또한 가능한, 분기 예측 정보를 획득하기 위한 프로세서에서의 분기 예측 기능 블록의 액세스들을 방지하기 위한 동작들을 수행하는 순차적 페치 로직 기능 블록을 포함한다. 설명된 실시 예들에서, 지정된 크기(예를 들어, 32 바이트, 64 바이트 등)의 명령어들의 블록들인 페치 그룹들은 캐시 메모리 또는 메인 메모리로부터 그룹으로서 페치되고 프로세서에서의 명령어 실행 파이프라인에서 실행을 위해 준비가 된다. 소정의 페치 그룹이 페치되고 실행을 위해 준비될 때, 순차적 페치 로직은 소정의 페치 그룹과 연관된 레코드를 확인하여(이러한 레코드가 존재하는 경우) 사전에 제어 이전 명령어들("CTI들")을 포함하지 않는 것으로 결정되었던, 소정의 페치 그룹 다음에 순차적으로 페치될 페치 그룹들의 수를 결정한다. 예를 들어, 일부 실시 예들에서, 순차적 페치 로직은 페치 그룹들의 수를 결정하기 위해 소정의 페치 그룹에서 CTI와 연관된 레코드를 확인한다. 또 다른 예로서, 일부 실시 예들에서, 순차적 페치 로직은 페치 그룹들의 수를 결정하기 위해 소정의 페치 그룹에서 CTI에 대한 타겟 명령어와 연관된 레코드를 확인한다. 페치 그룹들의 수에 CTI들이 없기 때문에, 페치 그룹들은 순차적으로 페치될 것이다 - 그리고 분기 예측 정보가 필요하지 않을 것이다. 페치 그룹들의 수 각각이 순차적으로 페치되고 실행을 위해 준비됨에 따라, 순차적 페치 로직은 분기 예측 정보를 획득하기 위한 분기 예측 기능 블록의 액세스들을 방지한다. 예를 들어, 순차적 페치 로직은 분기 예측 정보를 획득하기 위한 분기 예측 기능 블록에서의 분기 타겟 버퍼(BTB), 분기 방향 예측기 등의 확인을 방지할 수 있다.

[0013] 일부 실시 예들에서, BTB는 각각의 CTI들 다음에 순차적으로 페치될 CTI들을 갖지 않는 페치 그룹들의 수를 나타내는 상술한 레코드들의 일부 또는 전부를 저장하기 위해 사용된다. 이러한 실시 예들에서, BTB는 CTI들에 대한 레졸루션 정보와 함께, CTI 다음에 순차적으로 페치될 CTI(있는 경우)가 없는 페치 그룹의 수에 대한 정보를 포함한다. 동작 동안, 소정의 페치 그룹에서 지정된 유형의 CTI를 접할(예를 들어, 지정된 페치 그룹 내에서, 지정된 유형의 CTI를 검출) 시, 순차적 페치 로직 기능 블록은 후속 CTI가 폐기되기(즉, 실행을 완료하고 프로세서의 아키텍처 상태로 커밋될 준비가 되기) 전에 소정의 페치 그룹 다음에 순차적으로 실행을 위해 페치되는. 페치 그룹들의 수의 카운트를 유지한다. 그 다음, 순차적 페치 로직은 CTI와 연관된 BTB에서의 엔트리에, 페치 그룹들의 수의 카운트를 저장한다. 그 다음, BTB의 엔트리에서의 카운트는 페치 그룹들의 수에 대한 분기 예측 정보를 획득하기 위한 분기 예측 기능 블록의 액세스들을 방지하기 위해 여기에서 설명된대로 사용된다.

[0014] 일부 실시 예들에서는, 순차적 페치 테이블(SFT, sequential fetch table)이 각각의 CTI들 다음에 순차적으로 페치될 CTI들을 갖지 않는 페치 그룹들의 수를 나타내는 상술한 레코드들의 일부 또는 전부를 저장하기 위해 사용된다. 이러한 실시 예들에서, SFT에서의 엔트리들은 CTI들에 대한 타겟 명령어들과 연관되어 있으므로, SFT에서의 각 레코드는 대응하는 CTI의 특정 경로/레졸루션에 따라 순차적으로 페치될 CTI를 갖지 않는 페치 그룹들의 수에 대한 표시를 포함한다. 동작 동안, 소정의 페치 그룹에서의 CTI에 대한 타겟 명령어(즉, CTI로부터 이행되는 경로 상에 있는지 이행되지 않는 경로 상에 있는지 여부에 관계 없이, CTI의 타겟이 되는 명령어)의 폐기를 접할 시, 순차적 페치 로직 기능 블록은 후속 CTI가 폐기되기 전 순차적으로 실행을 위해 페치되는 페치 그룹들의 수의 카운트를 유지한다. 그 다음, 순차적 페치 로직은 타겟 명령어와 연관된 SFT에서의 엔트리에, 페치 그룹들의 수의 카운트를 저장한다. 그 다음, SFT의 엔트리에서의 카운트는 페치 그룹들의 수에 대한 분기 예측 정보를 획득하기 위한 분기 예측 기능 블록의 액세스들을 방지하기 위해 여기에서 설명된대로 사용된다.

[0015] 일부 실시 예들에서, 분기 예측 정보를 획득하기 위한 분기 예측 기능 블록의 액세스들을 방지하기 위해 상술한 레코드들을 사용할 때, 순차적 페치 로직은 대응하는 레코드(BTB, SFT 또는 다른 곳으로부터)로부터 페치 그룹들의 수를 획득하고 페치 그룹들의 수와 동일하게 카운터를 설정한다. 각 후속 페치 그룹이 페치되고 실행을 위해 준비될 때, 순차적 페치 로직은 카운터를 감소하고 상술한 바와 같이 분기 예측 기능 블록에 대한 액세스

를 방지한다. 카운터가 제로에 도달한 후, 하나 이상의 후속 페치 그룹이 페치되고 실행을 위해 준비됨에 따라, 순차적 페치 로직은 하나 이상의 후속 페치 그룹의 명령어들에 대한 분기 예측 정보를 획득하기 위해 분기 예측 기능 블록의 대응하는 액세스들이 수행되도록 허용한다. 즉, 순차적 페치 로직은 카운터를 사용하여 지정된 페치 그룹들의 수에 대한 분기 예측 기능 블록의 액세스들을 방지한 다음, 후속 페치 그룹들에 대한 분기 예측 기능 블록의 액세스들을 수행하기 시작한다. 이러한 실시 예들 중 일부에서, 순차적 페치 로직은 또한 카운터가 제로보다 크면 레코드들 자체의 확인을 중지한다 - 그리고 레코드들이 저장되는 기능 블록들의 부분들(예를 들어, BTB, SFT 등)의 전원이 꺼지거나 절전 모드(예를 들어, 감소된 공급 전압 등이 공급됨)에 놓이거나 그렇지 않으면 전력 소비를 감소시키도록 설정될 수 있다(예를 들어, 클록 게이팅되는 등).

[0016] 페치 그룹들에 대한 브랜치 예측 정보를 획득하기 위한 브랜치 예측 기능 블록의 액세스들을 방지하기 위해 소정의 CTI 다음에 순차적으로 페치될 CTI들을 포함하지 않는 페치 그룹들의 수를 나타내는 레코드들을 사용함으로써, 설명된 실시 예들은 분기 예측 기능 블록의 불필요한 액세스들을 방지할 수 있다. 이는 프로세서, 그리고 보다 일반적으로 전자 디바이스에서 전력 소비를 감소시키는 데 도움이 될 수 있다. 전력 소비 감소는 전자 디바이스의 사용 비용을 낮춰, 배터리 전력을 보다 효율적으로 사용할 수 있게 하며, 이는 전자 디바이스에 대한 사용자 만족도를 높일 수 있다.

[0017] **전자 디바이스**

[0018] 도 1은 일부 실시 예들에 따른 전자 디바이스(100)를 도시한 블록도를 나타낸다. 도 1에서 볼 수 있는 바와 같이, 전자 디바이스(100)는 프로세서(102) 및 메모리(104)를 포함한다. 일반적으로, 프로세서(102) 및 메모리(104)는 하드웨어로, 즉 다양한 회로 요소들 및 디바이스들을 사용하여 구현된다. 예를 들어, 프로세서(102) 및 메모리(104)는 하나 이상의 개별 반도체 칩을 포함하여, 하나 이상의 반도체 칩 상에서 전체가 제조될 수 있고, 별개의 회로 요소들과 조합하여 반도체 칩들로 형성될 수 있으며, 별개의 회로 요소만으로 제조될 수 있는 등이다. 여기에 설명된 바와 같이, 프로세서(102) 및 메모리(104)는 특정 페치 그룹들에 대한 분기 예측 기능 블록의 액세스들을 방지하기 위한 동작들을 수행한다.

[0019] 프로세서(102)는 전자 디바이스(100)에서 컴퓨터 및 다른 동작들(예를 들어, 제어 동작들, 구성 동작들 등)을 수행하는 기능 블록이다. 예를 들어, 프로세서(102)는 하나 이상의 마이크로 프로세서, 중앙 처리 유닛(CPU) 코어, 및/또는 다른 처리 메커니즘일 수 있거나 이를 포함할 수 있다.

[0020] 메모리(104)는 전자 디바이스(100)를 위한 메모리(예를 들어, "메인" 메모리)의 동작들을 수행하는 전자 디바이스(100)에서의 기능 블록이다. 메모리(104)는 4 세대 더블 데이터 레이트 동기식 DRAM(DDR4)과 같은 휘발성 메모리 회로들 및/또는 전자 디바이스(100)에서의 기능 블록들에 의한 사용을 위해 데이터 및 명령어들을 저장하기 위한 다른 유형들의 메모리 회로들, 및 메모리 회로들에 저장된 데이터 및 명령어들의 액세스들을 핸들링하고 다른 제어 또는 구성 동작들을 수행하기 위한 제어 회로들을 포함한다.

[0021] 전자 디바이스(100)는 예시 목적상 단순화된다. 그러나, 일부 실시 예들에서, 전자 디바이스(100)는 추가적인 또는 상이한 기능 블록들, 서브 시스템들, 요소들, 및/또는 통신 경로들을 포함한다. 예를 들어, 전자 디바이스(100)는 디스플레이 서브 시스템들, 전력 서브 시스템들, 입력-출력(I/O) 서브 시스템들 등을 포함할 수 있다. 전자 디바이스(100)는 일반적으로 여기에서 설명된 동작들을 수행하기에 충분한 기능 블록들 등을 포함한다.

[0022] 전자 디바이스(100)는 컴퓨터 동작들을 수행하는 임의의 디바이스일 수 있거나, 이에 포함될 수 있다. 예를 들어, 전자 디바이스(100)는 데스크탑 컴퓨터, 랩탑 컴퓨터, 웨어러블 컴퓨팅 디바이스, 태블릿 컴퓨터, 가상 또는 증강 현실 장비, 스마트폰, 인공 지능(AI) 또는 기계 학습 디바이스, 서버, 네트워크 기기, 장난감, 시청각 장비, 가전 기기, 차량 등, 및/또는 이들의 조합들일 수 있거나 이에 포함될 수 있다.

[0023] **프로세서**

[0024] 상술한 바와 같이, 전자 디바이스(100)는 마이크로 프로세서, CPU 코어, 및/또는 다른 처리 메커니즘일 수 있는 프로세서(102)를 포함한다. 도 2는 일부 실시 예들에 따른 프로세서(102)를 도시한 블록도를 나타낸다. 특정 기능 블록들이 도 2에 도시되어 있지만, 일부 실시 예들에서, 상이한 배열들, 연결, 수단 및/또는 유형들의 기능 블록들이 프로세서(102)에 존재할 수 있다. 일반적으로, 프로세서(102)는 여기에 설명된 동작들을 수행하기에 충분한 기능 블록들을 포함한다.

[0025] 도 2에서 볼 수 있는 바와 같이, 프로세서(102)에서의 기능 블록들은 프론트 엔드 서브 시스템(200), 실행 서브 시스템(202), 또는 메모리 서브 시스템(204)의 일부인 것으로 고려될 수 있다. 프론트 엔드 서브 시스템(200)은 메모리 서브 시스템(204)에서의, 또는 이와 조합하여 캐시 메모리들 또는 메인 메모리로부터 명령어들을 획득하

기 위한 동작들을 수행하고 실행 서버 시스템(202)에서의 실행 유닛 기능 블록들에 디스패치하기 위한 명령들을 준비하는 기능 블록들을 포함한다.

[0026] 프론트 엔드 서버 시스템(200)은 폐치된 명령어들의 실행을 위한 준비 및 디코딩과 관련된 동작들을 수행하는 기능 블록인 명령어 디코드(206)를 포함한다. 명령어 디코드(206)는 L1 명령어 캐시(216), L2 캐시(218), L3 캐시(도시되지 않음), 또는 메인 메모리(도시되지 않음)로부터, N-바이트 폐치 그룹들에서의 명령어들(예를 들어, 32-바이트 폐치 그룹에서의 네 개의 명령어들)을 폐치하거나 다른 방법으로 수신한다. 그 다음, 명령어 디코드(206)는 가능한 병렬로, 폐치 그룹에서의 명령어들을 각각의 마이크로 동작들로 디코딩한다. 그 다음, 명령어 디코드(206)는 실행을 위해 실행 서버 시스템(202)에서의 적절한 실행 유닛으로 포워딩될 명령어 디스패치(208)로 마이크로-op들을 전송한다.

[0027] 프론트 엔드 서버 시스템(200)은 또한 다음 폐치 그룹이 폐치될, 프로그램 카운터, 또는 메모리에서의 어드레스를 결정하기 위한 동작들을 수행하는 기능 블록인 다음 PC(210)를 포함한다. 다음 PC(210)는 프로그램 카운터의 초기 또는 현재 값에 기초하여, 프로그램 카운터에 대한 다음에 순차 값을 컴퓨팅한다. 예를 들어, 32-바이트 폐치 그룹들이 주어지면, 다음 PC(210)는 다음 어드레스 = 현재 어드레스+32 바이트를 컴퓨팅할 수 있다. 이행되는 CTI들이 프로그램 흐름을 변경하지 않을 때, 프론트 엔드 서버 시스템(200)은 메모리에서의 대응하는 순차 어드레스들로부터 폐치 그룹들을 폐치하기 위해 다음 PC(210)에 의해 컴퓨팅되는 프로그램 카운터의 순차 값들을 사용한다.

[0028] 프론트 엔드 서버 시스템(200)은 폐치 그룹들에서 CTI들의 레졸루션들을 예측하고 프로그램 카운터 및 이에 따라 후속 폐치 그룹이 폐치되는 메모리에서의 어드레스를 변경하기 위한 동작들을 수행하는 기능 블록인 분기 예측 유닛(212)을 더 포함한다. 다시 말해, 분기 예측 유닛(212)은 CTI 거동의 하나 이상의 레코드를 사용하여, CTI들의 "이행되는(taken)" 또는 "이행되지 않는(not-taken)" 레졸루션을 예측하고 이행되는 CTI들에 대해 예측된 타겟 어드레스를 제공한다. CTI들이 분기 예측 유닛(212)에 의해 이행되는 것으로 예측될 때, 다음 PC(210)에 의해 제공되는 다음 또는 후속 프로그램 카운터는 분기 예측 유닛(212)에 의해 반환된 타겟 어드레스를 사용하여 대체될 수 있다.

[0029] 도 3은 일부 실시 예들에 따른 분기 예측 유닛(212)을 도시한 블록도를 나타낸다. 분기 예측 유닛(212)이 도 3에서 다양한 기능 블록들을 갖는 것으로 도시되어 있지만, 분기 예측 유닛(212)은 본 설명의 목적상 단순화되며; 일부 실시 예들에서는 상이한 기능 블록들이 분기 예측 유닛(212)에 존재한다. 예를 들어, 일부 실시 예들에서, 다중 레벨 분기 예측, 분기 패턴 예측기들, 다중 레벨 분기 타겟 버퍼들 및/또는 방향 예측기들, 및/또는 다른 분기 예측 메커니즘들 또는 기술들이 사용되고, 대응하는 기능 블록들이 분기 예측 유닛(212)에 포함된다. 일반적으로, 분기 예측 유닛(212)은 여기에서 설명된 동작들을 수행하기에 충분한 기능 블록들을 포함한다.

[0030] 도 3에서 볼 수 있는 바와 같이, 분기 예측 유닛(212)에서의 기능 블록들은 제어기(300), 방향 예측기(302), 및 분기 타겟 버퍼(BTB, branch target buffer)(304)를 포함한다. 제어기(300)는 방향 예측기(302) 및 분기 타겟 버퍼(304)의 업데이트 및 조회, 다른 기능 블록들과의 통신 등과 같은 분기 예측 유닛(212)의 동작들을 수행하기 위한 회로 요소들을 포함한다. 방향 예측기(302)는 각 엔트리가 CTI와 연관된 어드레스 및 CTI의 이행되거나 이행되지 않는 레졸루션에 대한 표시를 저장하기 위한 다수의 엔트리들을 가지는 룩업 테이블, 리스트 등과 같은 레코드를 포함한다. 예를 들어, 어드레스 A에서의 CTI에 대해, 방향 예측기(302)는 어드레스 A 또는 이에 기초한 값을 CTI의 이행되거나 이행되지 않는 레졸루션의 대응하는 예측과 연관시키는 엔트리(예를 들어, 포화 카운터 등)를 포함할 수 있다. 분기 타겟 버퍼(304)는 각 엔트리가 CTI와 연관된 어드레스 및 CTI에 대한 타겟 어드레스에 대한 표시를 저장하기 위한 다수의 엔트리들을 가지는 룩업 테이블, 리스트 등과 같은 레코드를 포함한다. 예를 들어, 어드레스 A에서의 CTI에 대해, 분기 타겟 버퍼(304)는 어드레스 A 또는 이에 기초한 값을 CTI의 타겟 명령어에 대한 대응하는 절대 또는 상대 어드레스와 연관시키는 엔트리를 포함할 수 있다. 명령어들을 실행하는 동안, 제어기(300)는 CTI 명령어들의 실제 결과들에 기초하여, 방향 예측기(302) 및/또는 분기 타겟 버퍼(304)에서의 대응하는 엔트리들을 저장 및/또는 업데이트할 수 있으며, 이에 의해 CTI 명령어 레졸루션들의 상술한 예측에 사용되는 값들을 저장할 수 있다.

[0031] 일부 실시 예들에서, 분기 타겟 버퍼(304)는 분기 타겟 정보에 더하여, 후속 CTI들이 폐기되기 전에 CTI들 다음에 순차적으로 실행을 위해 폐치되었던 폐치 그룹들의 수의 카운트들의 레코드들을 저장하기 위해 사용된다. 이러한 실시 예들에서, 각 엔트리는 이러한 카운트가 이용 가능하다면, 대응하는 CTI와 연관된 카운트, 예를 들어, 숫자, 문자열 등이 저장되는 위치를 포함한다. 예를 들어, 카운트는 엔트리들에 이러한 목적으로 리저브

된 8 비트에 저장될 수 있다. 도 4는 일부 실시 예들에 따른 분기 타겟 버퍼(304)를 도시한 블록도를 나타낸다. 분기 타겟 버퍼(304)가 특정 정보를 저장하는 것으로 도시되었지만, 분기 타겟 버퍼(304)는 본 설명의 목적상 단순화되며; 일부 실시 예들에서 정보의 상이한 배열들이 분기 타겟 버퍼(304)에서의 엔트리들에 저장된다. 일반적으로, 분기 타겟 버퍼(304)는 여기에서 설명된 동작들을 수행하기에 충분한 정보를 저장한다.

[0032] 도 4에서 볼 수 있는 바와 같이, 분기 타겟 버퍼(304)는 다수의 엔트리들(408)을 포함하며, 각 엔트리는 어드레스(ADDR)(400), 분기 타겟(402), 카운트(404), 및 메타 데이터(406)를 포함한다. 어드레스(400)는 엔트리가 정보를 유지하는 CTI의, CTI에 대한 또는 다른 방법으로 CTI와 연관된 어드레스 또는 이에 기초한 값을 저장하기 위해 사용된다. 분기 타겟(402)은 엔트리가 정보를 유지하는 - 그리고 이에 따라 CTI가 다시 실행될 때 타겟 명령어의 어드레스를 예측하는 데 사용될 수 있는 - CTI에 대한 타겟 명령어의 어드레스, 또는 이에 기초한 값을 저장하기 위해 사용된다. 카운트(404)는 후속 CTI가 폐기되기 전에 엔트리가 정보를 유지하고 있는 CTI 다음에 순차적으로 실행을 위해 폐치되었던 폐치 그룹들의 수의 카운트를 저장하기 위해 사용된다. 메타 데이터(406)는 유효한 비트들, 허가 비트들 등과 같은 엔트리, 카운트, 및/또는 CTI에 대한 또는 이와 연관된 정보를 저장하기 위해 사용된다.

[0033] 도 2로 돌아가면, 프론트 엔드 서브 시스템(200)은 가능한, 분기 예측 정보를 획득하기 위한 분기 예측 기능 블록의 액세스들을 방지하기 위한 동작들을 수행하는 기능 블록인 순차적 폐치 로직(214)을 더 포함한다. 순차적 폐치 로직(214)은 폐치 그룹들과 연관된 레코드들을 사용하여 CTI를 포함하지 않는다고 사전에 결정되었던 소정의 폐치 그룹(또는 이 안의 명령어) 다음에 순차적으로 폐치될 폐치 그룹들의 수를 결정한다. 폐치 그룹들의 수 각각이 후속하여 폐치됨에 따라, 순차적 폐치 로직(214)은 분기 예측 정보를 획득하기 위한 분기 예측 유닛(212)의 액세스들을 방지한다.

[0034] 도 5는 일부 실시 예들에 따른 순차적 폐치 로직(214)을 도시한 블록도를 나타낸다. 순차적 폐치 로직(214)이 도 5에서 다양한 기능 블록들을 갖는 것으로 도시되어 있지만, 순차적 폐치 로직(214)은 본 설명의 목적상 단순화되며; 일부 실시 예들에서는 상이한 기능 블록들이 분기 순차적 폐치 로직(214)에 존재한다. 예를 들어, 순차적 폐치 로직(214)이 순차적 폐치 테이블(502)을 포함하는 것으로 도시되었지만, 일부 실시 예들에서, 순차적 폐치 로직(214)은 순차적 폐치 테이블을 포함하지 않거나 사용하지 않는다. 대신, 순차적 폐치 로직(214)은 대응하는 동작들을 수행하기 위해 분기 타겟 버퍼(304)에서의 엔트리들에 저장된 카운트 정보를 사용한다. 일반적으로, 순차적 폐치 로직(214)은 여기에서 설명된 동작들을 수행하기에 충분한 기능 블록들을 포함한다.

[0035] 도 5에서 볼 수 있는 바와 같이, 순차적 폐치 로직(214)에서의 기능 블록들은 제어기(500) 및 순차적 폐치 테이블(502)을 포함한다. 제어기(500)는 순차적 폐치 테이블(502)(또는 분기 타겟 버퍼(304))의 업데이트 및 조회, 다른 기능 블록들과의 통신 등과 같은 순차적 폐치 로직(214)의 동작들을 수행하기 위한 회로 요소들을 포함한다. 순차적 폐치 테이블(502)은 각 엔트리가 CTI의 타겟 명령어와 연관된 어드레스 및 후속 CTI가 폐기되기 전에 순차적으로 실행을 위해 폐치되었던 폐치 그룹들의 수의 카운트의 레코드를 저장하기 위한 다수의 엔트리들을 가지는 록업 테이블, 리스트 등과 같은 레코드를 포함한다. 도 6은 일부 실시 예들에 따른 순차적 폐치 테이블(502)을 도시한 블록도를 나타낸다. 순차적 폐치 테이블(502)이 특정 정보를 저장하는 것으로 도시되었지만, 순차적 폐치 테이블(502)은 본 설명의 목적상 단순화되며; 일부 실시 예들에서 정보의 상이한 배열들이 순차적 폐치 테이블(502)에서의 엔트리들에 저장된다. 일반적으로, 순차적 폐치 테이블(502)은 여기에서 설명된 동작들을 수행하기에 충분한 정보를 저장한다.

[0036] 도 6에서 볼 수 있는 바와 같이, 순차적 폐치 테이블(502)은 다수의 엔트리들(606)을 포함하며, 각 엔트리는 CTI 타겟 어드레스(ADDR)(600), 카운트(602), 및 메타 데이터(604)를 포함한다. CTI 타겟 어드레스(600)는 엔트리가 정보를 유지하는 CTI에 대한 타겟 명령어의, 타겟 명령어에 대한 또는 다른 방법으로 타겟 명령어와 연관된 어드레스 또는 이에 기초한 값을 저장하기 위해 사용된다. 일반적으로, CTI의 "타겟" 명령어는 CTI가 실행될 때 프로그램 흐름이 점프하는 명령어이고, 이에 따라 메모리에서의 어드레스이다. 정적 타겟 점프 명령어와 같은 정적으로 정의된 타겟이 있는 무조건 CTI의 경우, 타겟 명령어가 하나만 있다 - 그리고 이에 따라 프로그램 흐름은 항상 CTI 명령어로부터 메모리에서의 동일한 어드레스로 점프한다. 그러나, 조건부 CTI의 경우, 적어도 두 개의 타겟 명령어들(프로그램 코드에서의 이행되는 경로 및 이행되지 않는 경로 상에 있는)이 있고, 임의의 수의 타겟 명령어가 있을 수 있다. 예를 들어, 타겟 명령어가 실행 시간에 지정되는 간접 CTI는 이행되지 않는 경로가 순차적이라도, 이행되는 경로 상에 임의의 수의 타겟 명령어를 가질 수 있다. 순차적 폐치 테이블(502)은 CTI들에 대한 타겟 명령어들과 연관된 레코드들을 저장하기 때문에, 소정의 CTI에 대한(그리고 이에 따라 소정의 CTI로부터의 경로) 각각의 가능한 타겟 명령어는 순차적 폐치 테이블(502)에서 연관된 별개의 엔트리를 가질 수 있다. 카운트(602)는 후속 CTI가 폐기되기 전에 대응하는 CTI 다음에 순차적으로 실행을 위해 폐치

되었던 페치 그룹들의 수의 카운트를 저장하기 위해 사용된다. 메타 데이터(604)는 유효한 비트들, 허가 비트들과 같은 엔트리 및/또는 카운트에 대한 또는 이와 연관된 정보를 저장하기 위해 사용된다.

- [0037] 페치 그룹들의 수들을 결정하기 위한 상술한 레코드들을 저장하기 위해 순차적 페치 테이블(502)이 사용되는 실시 예들에서, 프로그램 카운터, 즉 소정의 페치 그룹이 페치될 어드레스를 수신 시, 제어기(500)가 연관된 어드레스를 갖는 엔트리가 순차적 페치 테이블(502)에 존재하는지를 결정하기 위해 순차적 페치 테이블(502)에서 조회를 수행한다. 다시 말해, 조회는 소정의 페치 그룹에서의 어드레스 범위에서의 다수의 어드레스들 내에 존재하는 명령어, 즉 CTI의 타겟 명령어의 어드레스가 순차적 페치 테이블(502)에서 찾아질 것인지를 결정한다. 만약 그렇다면, 제어기(500)는 카운트(602)로부터 대응하는 카운트를 획득한 다음 분카운트를 페치 그룹들의 수로 사용하여 기 예측 정보를 획득하기 위한 분기 예측 유닛(212)의 액세스들을 방지한다. 그렇지 않으면, 순차적 페치 테이블(502)에서 일치하는 어드레스가 찾아지지 않을 때, 제어기(500)는 분기 예측 유닛(212)의 액세스들을 방지하지 않는다, 즉 분기 예측 정보의 획득이 정상적으로 진행되도록 허용한다.
- [0038] 페치 그룹에서의 CTI에 대한 분기 예측 정보에 대한 조회를 수행하는 부분으로서, 페치 그룹들의 수들을 결정하기 위한 상술한 레코드들을 저장하기 위해 분기 타겟 버퍼(304)가 사용되는 실시 예들에서, 분기 예측 유닛(212)은 분기 타겟 버퍼(304)로부터, 이러한 타겟 어드레스가 존재한다면, CTI에 대해 예측되는 타겟 명령어들의 어드레스를 획득한다. 분기 예측 유닛(212)은 또한 분기 타겟 버퍼(304)로부터, 카운트(404)를 획득하는데, 이는 순차적 페치 로직(214)에서 제어기(500)로 반환된다. 제어기(500)는 분기 예측 정보를 획득하기 위한 분기 예측 유닛(212)의 액세스들을 방지하기 위한 페치 그룹들의 수로서 카운트를 사용한다. 그렇지 않으면, 분기 타겟 버퍼(304)에서 일치하는 어드레스 및/또는 카운트가 찾아지지 않을 때, 제어기(500)는 분기 예측 유닛(212)의 액세스들을 방지하지 않는다, 즉 분기 예측 정보의 획득이 정상적으로 진행되도록 허용한다.
- [0039] 일부 실시 예들에서, 순차적 페치 테이블(502)은 단지 제한된 수의 엔트리들(예를 들어, 32개의 엔트리들, 64개의 엔트리들 등)만을 포함하고 이에 따라 프로세서(102)의 동작 동안 용량이 채워지게 될 수 있다. 순차적 페치 테이블(502)이 가득 찰 때, 새로운 정보가 순차적 페치 테이블(502)에 저장되기 위해서는 엔트리에서의 기존 정보를 덮어 써질 필요가 있을 것이다. 일부 실시 예들에서, 순차적 페치 테이블(502)에서의 엔트리들은 하나 이상의 대체 정책, 가이드 라인 등을 사용하여 제어기(500)에 의해 관리된다. 이러한 실시 예들에서, 덮어 써질 엔트리를 선택할 때, 엔트리는 대체 정책들, 가이드 라인들 등에 따라 선택된다. 예를 들어, 제어기(500)는 LRU(least-recently-used) 대체 정책을 사용하여 순차적 페치 테이블(502)에서의 엔트리들에서의 정보를 관리할 수 있다.
- [0040] 순차적 페치 로직(214)이 도 2에서 다른 기능 블록들과 별개인 단일 기능 블록인 것으로 도시되었지만, 일부 실시 예들에서, 순차적 페치 로직(214)의 일부 또는 전부는 도 2에 도시된 다른 기능 블록들에 포함될 수 있음을 유념한다. 이러한 실시 예들에서, 순차적 페치 로직(214)의 결과라고 간주되는 동작들이 다른 기능 블록들의 회로 요소들에 의해 수행될 수 있다. 일반적으로, 순차적 페치 로직(214)은 도 2에 도시된 프로세서(102)에서의 회로 요소들의 특정 위치들에 관한 제한 없이, 설명된 동작들을 수행하는 데 사용되는 다양한 회로 요소들을 포함한다.
- [0041] 도 2로 돌아가면, 실행 서브 시스템(202)은 정수 실행 유닛(222) 및 부동 소수점 실행 유닛(224)(총칭하여 "실행 유닛들")을 포함하며, 이것들은 각각 정수 및 부동 소수점 명령어들을 실행하기 위한 동작들을 수행하는 기능 블록들이다. 실행 유닛들은 하드웨어 재명명, 실행 스케줄러들, 산술 로직 유닛들(ALU), 부동 소수점 곱셈 및 덧셈 유닛들(부동 소수점 실행 유닛(224)), 레지스터 파일들 등과 같은 요소들을 포함한다.
- [0042] 실행 서브 시스템은 또한 실행된 명령어들의 결과들이 대응하는 명령어들이 실행을 완료한 후, 그러나 결과가 프로세서(102)의 아키텍처 상태로 커밋되기(예를 들어, 캐시 또는 메모리에 기록되고 다른 동작들에 사용하기 위해 이용 가능하게 만들어지기) 전에 유지되는 기능 블록인 폐기 큐(226)를 포함한다. 일부 실시 예들에서, 특정 명령어들은 프로그램 순서를 벗어나 실행될 수 있고, 폐기 큐(226)는 순서를 벗어난 명령어들의 결과들이 다른 순서를 벗어난 명령어들에 대해 적절하게 폐기되도록 하는 데 사용된다.
- [0043] 일부 실시 예들에서, 폐기 큐(226)는 후속 CTI가 폐기되기 전에 CTI 다음에 순차적으로 실행을 위해 폐치되는 페치 그룹들의 수들의 카운트들을 유지하기 위한 동작들 중 적어도 일부를 수행한다. 예를 들어, 일부 실시 예들에서, 프론트 엔드 서브 시스템(200)(예를 들어, 명령어 디코드(206))은 실행 서브 시스템(202)에서 실행될 CTI들과 함께, 명령어들이 CTI들이라는(또는 그렇지 않다는) 표시를 포함한다. 예를 들어, 프론트 엔드 서브 시스템(200)은 실행 서브 시스템(202)을 통해 명령어들을 수신하는 메타 데이터 비트들에 지정된 플래그 비트를 설정할 수 있다. CTI인 것으로 표시된 명령어를 접할 시, 폐기 큐(226)는 CTI인 것으로 표시된 후속 명령어가

폐기되기 전에 폐기되는 폐치 그룹들(또는 보다 일반적으로, 개별 명령어들)의 카운트를 유지하기 시작할 수 있다. 그 다음, 카운트는 여기에서 설명된 바와 같이 향후 사용을 위해 저장되기 위해 순차적 폐치 로직(214)으로 전달될 수 있다. 일부 실시 예들에서, 카운트들이 대응하는 임계를 초과하지 않는 한, 카운트들은 폐기 큐(226)에 의해 보고되지 않는다(또는 임의의 레코드들을 업데이트하는 데 사용되지 않는다).

[0044] 메모리 서브 시스템(204)은 데이터의 액세스들과 같은 동작들을 핸들링하기 위한 제어 회로들 뿐만 아니라, 명령어들 및/또는 데이터를 사용하는 기능 블록들 부근에 명령어들 및/또는 데이터의 제한된 수들의 카피들을 저장하기 위한 휘발성 메모리 회로들을 포함하는 기능 블록들인 계층 구조의 캐시들을 포함한다. 계층 구조는 두 개의 레벨들을 포함하는데, 레벨 1(L1) 명령어 캐시(216) 및 L1 데이터 캐시(220)는 제1 레벨 상에 있고 L2 캐시(218)는 제2 레벨 상에 있다. 메모리 서브 시스템(204)은 메모리(104)에 통신 가능하게 결합되고 외부 L3 캐시(도시되지 않음)에 결합될 수 있다. 메모리(104)는 명령어들 및/또는 데이터(예를 들어, 디스크 드라이브 또는 고체 상태 드라이브)(도시되지 않음)를 위한 장기 저장 장치로서 기능하는 비 휘발성 대용량 저장 디바이스에 결합될 수 있다.

[0045] **제어 이전 명령어들과 연관된 순차적 폐치 그룹들의 레코드들을 사용하여 분기 예측기에 액세스하는 것을 방지**

[0046] 설명된 실시 예들에서, 전자 디바이스에서의 프로세서(예를 들어, 프로세서(102))는 CTI들과 연관된 레코드들을 사용하여 CTI들 다음에 순차적으로 폐치될 CTI들을 포함하지 않는 폐치 그룹들의 수를 결정하고 폐치 그룹들의 수 동안, 대응하는 분기 예측 정보를 획득하기 위한 분기 예측 유닛의 액세스들을 방지한다. 도 7은 일부 실시 예들에 따라 분기 예측 유닛의 액세스들을 방지하기 위해 CTI들과 연관된 레코드들을 사용하기 위한 프로세스를 도시하는 흐름도를 나타낸다. 도 7에 도시된 동작들은 일부 실시 예에 의해 수행되는 동작들의 일반적인 예로서 제시됨을 유념한다. 다른 실시 예들에 의해 수행되는 동작들은 상이한 순서로 수행되는 상이한 동작들 및/또는 동작들을 포함한다. 도 7에서의 예에 대해, 프로세서(102)와 유사한 내부 배열을 갖는 전자 디바이스에서의 프로세서는 다양한 동작들을 수행하는 것으로 설명된다. 그러나, 일부 실시 예들에서, 상이한 내부 배열을 갖는 프로세서가 설명된 동작들을 수행한다.

[0047] 도 7에 도시된 동작들은 프로세서가 하나 이상의 CTI 각각과 연관된 레코드들을 유지할 때 시작되며, 각 레코드는 CTI를 포함하지 않는 지정된 수의 폐치 그룹이 대응하는 CTI 다음에 순차적으로 폐치될 것임을 나타낸다(단계 700). 이러한 동작 동안, 순차적 폐치 로직에서의 제어기(예를 들어, 순차적 폐치 로직(214)에서의 제어기(500))는 폐기 큐(예를 들어, 폐기 큐(226))로부터 CTI 또는 CTI에 대한 식별자의 표시 및 CTI 다음에 순차적으로 폐치될 폐치 그룹들의 수를 수신한다. 예를 들어, 폐치 그룹들이 네 개의 명령어들을 포함하는 실시 예를 가정하면, 폐기 큐가 CTI를 접한 다음, 다음 CTI가 폐기되기 전에 129개의 명령어들을 카운트하면, 폐기 큐는 CTI의 신원, 값 129 또는 또 다른 값, 이를테면 32(이는 폐치 그룹들의 수를 나타 내기 위해 가장 가까운 정수로 내림된 129/4이다)와 함께, 순차적 폐치 로직과 통신할 수 있다. (여기서 사용될 때, CTI를 "접하는 것"은 CTI와 연관된 프로세서 플래그에 기초하여 폐기 명령어들 중에서 CTI를 검출하는 것, CTI 및/또는 후속 명령어들 또는 결과 값들 등과 연관된 명령어 흐름에서 거동 또는 패턴을 검출하는 것 등을 포함한다.) 그 다음, 제어기는 지정된 폐치 그룹들의 수를 나타내기 위해 CTI와 연관된 레코드를 업데이트한다.

[0048] 도 8 및 도 10은 도 7의 단계 700에 대해 설명된 바와 같이 하나 이상의 CTI와 연관된 레코드들을 유지하기 위해 수행되는 동작들을 도시하는 흐름도들을 나타낸다. 도 8은 일부 실시 예들에 따라 레코드들을 저장하기 위해 분기 타겟 버퍼가 사용될 때 CTI들과 연관된 레코드들을 유지하기 위해 수행되는 동작들을 도시하는 흐름도를 나타낸다. 도 10은 일부 실시 예들에 따라 레코드들을 저장하기 위해 순차적 폐치 테이블이 사용될 때 CTI들과 연관된 레코드들을 유지하기 위해 수행되는 동작들을 도시하는 흐름도를 나타낸다. 도 8 및 도 10에 도시된 동작들은 일부 실시 예에 의해 수행되는 동작들의 일반적인 예로서 제시됨을 유념한다. 다른 실시 예들에 의해 수행되는 동작들은 상이한 동작들, 상이한 순서로 수행되는 동작들, 및/또는 상이한 기능적 블록들에 의해 수행되는 동작들을 포함한다.

[0049] 도 8에 도시된 동작들은 폐기 큐가 지정된 유형의 CTI의 폐기를 접할 시, 후속 CTI가 폐기되기 전에 순차적으로 폐기되는 폐치 그룹들의 수(또는 보다 일반적으로 폐기되는 개별 명령어들)의 카운트를 유지하기 시작할 때 시작된다(단계 800). 이러한 동작을 위해, 폐기 큐는 지정된 유형의 CTI 명령어의 폐기를 모니터링한 다음, 다음 CTI 명령어 전에 폐기되는 후속 명령어들의 카운트를 유지한다. 예를 들어, 폐기 큐는 다음 CTI가 폐기될 때까지 지정된 유형의 CTI 명령어 다음 폐기되는 각 명령어마다 카운터를 증분시킬 수 있다.

[0050] 상술한 바와 같이, 도 8에 도시된 실시 예에서 레코드들을 저장하기 위해서는 분기 타겟 버퍼가 사용된다. 일부 실시 예들에서, 분기 타겟 버퍼는 CTI(예를 들어, 메모리에서의 CTI 명령어의 어드레스) 또는 CTI가 포함된 폐

치 그룹에 대한 식별자에 의해 인덱싱된 분기 타겟 레코드들을 저장한다. 이러한 이유로, 이러한 실시 예들에서, 분기 타겟 버퍼에서의 엔트리에서의 카운트 정보는 CTI 또는 페치 그룹과만 연관될 것이고 - CTI(가능하게는 다수의)의 타겟 명령어(들)와는 연관되지 않을 것이다. 다시 말해, 상술한 데이터의 단편인 카운트 정보는 각각의 타겟 명령어를 통해 CTI로부터의 단일 결정 경로만을 나타낸다. 따라서 분기 타겟 버퍼에서의 이러한 레코드들은 정적 타겟 명령어 조건부 CTI들 및/또는 동적으로 지정된 타겟 명령어들을 가지는 CTI들의 가능한 모든 레졸루션들에 대한 순차적 페치 그룹들의 카운트를 결정하기 위해 안정적으로 사용될 수 없다. 이에 따라, 일부 실시 예들에서, "지정된" 유형의 CTI는 무조건 정적 타겟 명령어 CTI 또는 조건부 CTI로부터의 이행되지 않는 경로이다.

[0051] 그 다음, 페기 큐는 카운트를 순차 페치 로직으로 전달하며, 이는 CTI 또는 페치 그룹과 연관된 분기 타겟 버퍼에서의 엔트리에, 페치 그룹들의 수의 카운트를 나타내는 레코드를 저장한다(단계 802). 이러한 동작을 위해, 순차적 페치 로직은 이를테면 CTI의 어드레스 또는 이에 기초한 값을 결정하고 분기 타겟 버퍼에서의 대응하는 엔트리를 결정함으로써 레코드가 저장될 분기 타겟 버퍼에서의 엔트리를 결정한다. 다음으로, 순차적 페치 로직은 결정된 엔트리에, 페치 그룹들의 수를 나타내는 레코드를 저장한다. 예를 들어, 분기 타겟 버퍼에서의 엔트리가 8 비트 값에 대한 공간을 포함할 때, 순차적 페치 로직은 엔트리에 카운트를 나타내는 8 비트 값을 저장한다. 엔트리의 크기가 카운트를 저장하기에 불충분한 경우(예를 들어, 8-비트 카운트 값의 경우 카운트는 255를 초과한다), 엔트리에 디폴트 또는 제로 값이 저장될 수 있다 - 이에 의해 특정 CTI에 대해 분기 예측기의 액세스들이 수행될 것임을 나타낸다(방지되지 않음). 대안적으로, 최대 값이 엔트리에 저장될 수 있으며, 이에 의해 분기 예측기의 액세스들 중 적어도 일부의 방지를 가능하게 할 수 있다.

[0052] 도 9는 일부 실시 예들에 따라 대응하는 CTI들 다음에 순차적으로 폐기되는 CTI들을 포함하지 않는 페치 그룹들의 카운트로 분기 타겟 버퍼를 업데이트하기 위한 동작들의 타임 라인 도해를 나타낸다. 도 9에 도시된 바와 같이, 시간은 왼쪽에서 오른쪽으로 진행되고, 시간 동안 각각 프로그램 코드로부터의 별도의 명령어 집합들을 포함하는 페치 그룹들의 수(FG)이 폐기되고, 실행을 위해 준비(예를 들어, 디코딩, 디스패치 등)되고, 실행되며, 폐기된다. 각 페치 그룹은 다수의 개별 명령어들(예를 들어, 네 개, 여섯 개 등)을 포함하고 페치 그룹들에서의 명령어들 중 세 개의 CTI 명령어들이 있다.

[0053] 도 9에서, 페치 그룹(902)에서의 두 번째 명령어인 제1 CTI 명령어(CTI1)는 명령어 흐름이 첫 번째 어드레스에서의 페치 그룹(902)으로부터 두 번째 비 순차적 어드레스에서의 페치 그룹(904)으로 점프하게 하는 정적 무조건 분기 명령어이며, 이는 페치 그룹(902)에서의 두 번째 명령어로부터 페치 그룹(904)의 초기 명령어로의 화살표에 의해 도시된다. CTI1(지정된 CTI의 유형들 중 하나)의 폐기를 검출 시, 페기 큐는 후속 CTI - 이는 페치 그룹(910)에서의 세 번째 명령어인 정적 타겟 어드레스를 갖는 이행되지 않는 조건부 CTI2이다 - 가 폐기되기 전에 순차적으로 폐기되는(즉, 구성 명령어들이 폐기되는) 페치 그룹들의 수의 카운트를 유지한다. CTI2가 폐기되기 전에 세 개의 페치 그룹들(페치 그룹들(904-908))이 폐기되기 때문에, 페기 큐는 CTI1의 신원(예를 들어, 이의 어드레스, 프로세서 명령어 태그 또는 내부 식별자 등) 및 카운트 값 3 또는 또 다른 값, 이를테면 폐기되는 명령어들의 수 - 14를 순차적 페치 로직으로 전달한다. CTI1의 식별자 및 값을 수신 시, 순차적 페치 로직은 도 9에서의 분기 타겟 버퍼의 예에서 CTI1에 대한 엔트리에 도시된 바와 같이 분기 타겟 버퍼에서의 대응하는 엔트리를 업데이트한다. 도 9에서 분기 타겟 버퍼에 도시된 바와 같이 CTI1 및 CTI2는 CTI1 및 CTI2의 어드레스들 또는 대응하는 페치 그룹들과 연관된 어드레스들과 같이 CTI1 및 CTI2를 나타내거나 식별하는 다른 값들일 수 있음을 유념한다.

[0054] CTI1에 대한 카운트의 끝 역할을 한 제2 CTI 명령어(CTI2)는 또한 페기 큐가 제2/대응 카운트를 시작하게 한다. 상술한 바와 같이, CTI2는 이행되지 않으므로, 페치 그룹들은 다음 어드레스들로부터 순차적으로(다음 PC 기능 블록에 의해 생성되는 바와 같이) 폐기되고 페기 큐는 페치 그룹(916)에서의 후속 CTI(CTI3)(이는 프로그램 흐름을 페치 그룹(918)으로 향하게 한다)가 폐기될 까지 대응하는 카운트를 유지한다. CTI3가 폐기되기 전에 두 개의 전체 페치 그룹들(페치 그룹들(912-914))이 폐기되기 때문에, 페기 큐는 CTI2의 신원 및 카운트 값 2 또는 또 다른 값, 이를테면 폐기되는 명령어들의 수 - 10을 순차적 페치 로직으로 전달한다. CTI2의 식별자 및 값을 수신 시, 순차적 페치 로직은 도 9에서의 분기 타겟 버퍼의 예에서, CTI2로서 도시된 바와 같이, 분기 타겟 버퍼에서의 대응하는 엔트리를 업데이트한다. CTI2는 조건부 CTI의 이행되지 않는 경로 케이스이고 이에 따라 지정된 유형의 CTI들 중 하나라는 점을 유념한다 - 일부 실시 예들에서 이행되는 경로는 유사하게 트래킹되고 레코딩되지 않는다.

[0055] 도 10으로 진행하면, 도 10에 도시된 동작들은 페기 큐가 CTI에 대한 타겟 명령어의 폐기를 접할 시, 후속 CTI가 폐기되기 전에 순차적으로 폐기되는 페치 그룹들의 수(또는 보다 일반적으로 폐기되는 개별 명령어들의 카

운트를 유지하기 시작할 때 시작된다(단계 1000). 이러한 동작을 위해, 폐기 큐는 CTI 그리고 그 다음 CTI의 타겟 명령어(즉, CTI 다음에 순차적으로 다음 명령어)의 폐기를 모니터링하고 다음 CTI 명령어 전에 폐기되는 후속 명령어들의 카운트를 유지한다. 예를 들어, 폐기 큐는 다음 CTI가 폐기될 때까지 지정된 유형의 타겟 명령어 다음 폐기되는 각 명령어마다 카운터를 증분시킬 수 있다.

[0056] 상술한 바와 같이, 도 10에 도시된 실시 예에서 레코드들을 저장하기 위해서는 순차적 폐치 테이블이 사용된다. 일부 실시 예들에서, 순차적 폐치 테이블은 CTI들에 대한 타겟 명령어들과 연관된 레코드들을 저장한다. 다시 말해, 이러한 실시 예들에서, 순차적 폐치 테이블은 CTI에 대한 각각의 가능한 타겟 명령어와 연관된 레코드 - 그리고 이에 따라 CTI로부터 이행되는 각각의 경로를 포함할 수 있다. 이러한 이유로, 다양한 유형들의 CTI가 이러한 실시 예들에서 핸들링될 수 있다(즉, 순차적 폐치 카운트들이 레코딩되게 할 수 있다). 이는 카운트 정보가 분기 타겟 버퍼에 저장되는 실시 예들과 상이하다는 점을 상기한다 - 이는 CTI들 또는 폐치 그룹들과 연관된 카운트 정보를 저장하고 - 대응하는 타겟 명령어(들)은 저장하지 않는다 - 그리고 이에 따라 지정된 유형들의 CTI들만 핸들링할 수 있다.

[0057] 그 다음, 폐기 큐는 카운트를 순차적 폐치 로직으로 전달하며, 이는 타겟 명령어와 연관된 순차적 폐치 테이블에서의 엔트리에, 폐치 그룹들의 수의 카운트를 나타내는 레코드를 저장한다(단계 1002). 이러한 동작을 위해, 순차적 폐치 로직은 이를테면 타겟 명령어의 어드레스 또는 이에 기초한 값을 결정하고 순차적 폐치 테이블에서의 대응하는 엔트리를 결정함으로써 레코드가 저장될 순차적 폐치 테이블에서의 엔트리를 결정한다. 다음으로, 순차적 폐치 로직은 결정된 엔트리에, 폐치 그룹들의 수를 나타내는 레코드를 저장한다. 예를 들어, 순차적 폐치 테이블에서의 엔트리가 8 비트 값에 대한 공간을 포함할 때, 순차적 폐치 로직은 엔트리에 카운트를 나타내는 8 비트 값을 저장한다. 엔트리의 크기가 카운트를 저장하기에 불충분한 경우(예를 들어, 8-비트 카운트 값의 경우 카운트는 255를 초과한다), 엔트리에 디폴트 또는 제로 값이 저장될 수 있다 - 이에 의해 특정 타겟 명령어에 대해 분기 예측기의 액세스들이 수행될 것임을 나타낸다(방지될 수 없음). 대안적으로, 최대 값이 엔트리에 저장될 수 있으며, 이에 의해 분기 예측기의 액세스들 중 적어도 일부의 방지를 가능하게 할 수 있다.

[0058] 도 11은 일부 실시 예들에 따라 대응하는 CTI들 다음에 순차적으로 폐치되는 CTI들을 포함하지 않는 폐치 그룹들의 카운트로 순차적 폐치 테이블을 업데이트하기 위한 동작들의 타임 라인 도해를 나타낸다. 도 11에 도시된 바와 같이, 시간은 왼쪽에서 오른쪽으로 진행되고, 시간 동안 각각 프로그램 코드로부터의 별도의 명령어 집합들을 포함하는 폐치 그룹들의 수(FG)이 폐치되고, 실행을 위해 준비(예를 들어, 디코딩, 디스패치 등)되고, 실행되며, 폐기된다. 각 폐치 그룹은 다수의 개별 명령어들(예를 들어, 네 개, 여섯 개 등)을 포함하고 폐치 그룹들에서의 명령어들 중 세 개의 CTI 명령어들이 있다.

[0059] 도 11에서, 폐치 그룹(1102)에서의 두 번째 명령어인 제1 CTI 명령어(CTI1)는 명령어 흐름이 첫 번째 어드레스에서의 폐치 그룹(1102)으로부터 두 번째 비 순차적 어드레스에서의 폐치 그룹(1104)으로 점프하게 하는 정적 무조건 분기 명령어이며, 이는 폐치 그룹(1102)에서의 두 번째 명령어로부터 폐치 그룹(1104)의 초기 명령어로의 화살표에 의해 도시된다. ADDR1에서의 타겟 명령어의 폐기를 검출 시, 폐기 큐는 후속 CTI - 이는 폐치 그룹(1112)에서의 첫 번째 명령어인 정적 타겟 어드레스를 갖는 이행되지 않는 조건부 CTI2이다 - 가 폐기되기 전에 순차적으로 폐기되는(즉, 구성 명령어들이 폐기되는) 폐치 그룹들의 수의 카운트를 유지한다. CTI2가 폐기되기 전에 세 개의 전체 폐치 그룹들(폐치 그룹들(1104-1108))이 폐기되기 때문에, 폐기 큐는 ADDR1에서의 타겟 명령어의 신원(예를 들어, 이의 어드레스, 프로세서 명령어 태그 또는 내부 식별자 등) 및 카운트 값 3 또는 또 다른 값, 이를테면 폐기되는 명령어들의 수 - 13을 순차적 폐치 로직으로 전달한다. ADDR1에서의 타겟 명령어의 식별자 및 값을 수신 시, 순차적 폐치 로직은 도 11에서의 순차적 폐치 테이블의 예에서 ADDR1에 대한 엔트리에 도시된 바와 같이 순차적 폐치 테이블에서의 대응하는 엔트리를 업데이트한다. 도 11에서 순차적 폐치 테이블에 도시된 바와 같이 ADDR1 및 ADDR2는 어드레스들 ADDR1 및 ADDR2 또는 대응하는 타겟 명령어들을 나타내거나 식별하는 다른 값들일 수 있음을 유념한다.

[0060] ADDR2에서의 CTI2에 대한 타겟 명령어의 폐기는 폐기 큐가 제2/대응 카운트를 시작하게 한다. 상술한 바와 같이, CTI2는 이행되지 않으므로, 타겟 명령어는 순차적으로 다음 어드레스들인 ADDR2에 있다 - 그리고 폐치 그룹들은 ADDR2로부터 폐치된다. 폐기 큐는 폐치 그룹(1116)에서의 후속 CTI(CTI3)(이는 프로그램 흐름을 폐치 그룹(1118)으로 향하게 한다)가 폐기될 까지 대응하는 카운트를 유지한다. CTI3가 폐기되기 전에 두 개의 전체 폐치 그룹들(폐치 그룹들(1112-1114))이 폐기되기 때문에, 폐기 큐는 ADDR2에서의 타겟 명령어의 신원(예를 들어, 이의 어드레스, 프로세서 명령어 태그 또는 내부 식별자 등) 및 카운트 값 2 또는 또 다른 값, 이를테면 폐기되는 명령어들의 수 - 13을 순차적 폐치 로직으로 전달한다. ADDR2에서의 타겟 명령어의 식별자 및 값을 수신 시, 순차적 폐치 로직은 도 11에서의 순차적 폐치 테이블의 예에서 ADDR2에 대한 엔트리에 도시된 바와 같이 순차적

폐치 테이블에서의 대응하는 엔트리를 업데이트한다.

- [0061] 일부 실시 예들에서, 폐기 큐는 분기 타겟 버퍼 및/또는 순차적 폐치 로직에 카운트들을 보고하기 위해 적어도 하나의 임계를 채용한다. 이러한 실시 예들에서, 폐치 그룹들의 임계 개수 미만이 카운팅될 때, 폐기 큐는 CTI 및/또는 카운트를 순차적 폐치 로직에 보고하지 않는다. 이러한 방식으로, 폐기 큐는 엔트리들이 빠르게 덮어 쓰지게 함으로써 카운트가 저장되는 레코드(즉, 분기 타겟 버퍼 또는 순차적 폐치 테이블)를 스래싱(thrashing)하는 것을 방지할 수 있다. 일부 실시 예들에서, 임계SMS 동적이고 조건들이 지시하는 대로 실행 시간에 설정/재설정될 수 있다. 일부 실시 예들에서, 레코드(예를 들어, 분기 타겟 버퍼)에서의 엔트리들은 대응하는 CTI 또는 타겟 명령어와 후속 CTI 사이에서 임계 개수 미만의 폐치 그룹들이 발생할 때 각자의 카운트들이 무효인 것으로 마킹되거나 디폴트 값(예를 들어, 0)으로 설정되게 할 수 있다.
- [0062] 폐기 큐가 도 8 내지 도 11에서의 예들에 대해 후속 CTI가 폐기되기 전에 순차적으로 폐기되는 폐치 그룹들(또는 보다 일반적으로 폐기되는 개별 명령어들)의 수의 카운트를 유지하는 것으로 설명되었지만, 일부 실시 예들에서 상이한 기능 블록이 카운트를 유지한다. 일반적으로, 설명된 실시 예들에서, 개별 CTI 명령어들 및/또는 특정 유형들의 CTI 명령어들을 식별하고 식별된 CTI 명령어들 사이의 명령어들의 수를 카운팅할 수 있는 임의의 기능 블록이 카운트를 유지할 수 있다. 예를 들어, 일부 실시 예들에서, 디코드 유닛 및/또는 분기 예측 유닛이 카운트를 유지하기 위한 동작들의 일부 또는 전부를 수행하지만, 잘못 예측된 분기들에 대해 다양한 롤백 및 복구 동작들이 수행될 수 있다.
- [0063] 도 7로 돌아가면, 후속하여 프로그램 코드를 실행할 때, 순차적 폐치 로직은 소정의 CTI와 연관된 레코드에 기초하여, CTI를 갖지 않는 지정된 폐치 그룹들의 수가 CTI 다음에 순차적으로 실행을 위해 폐치될 것임을 결정한다(단계 702). 여기서 사용될 때, CTI와 "연관된" 레코드는 레코드들을 저장하기 위해 분기 타겟 버퍼가 사용되는 실시 예들의 경우 CTI 자체 또는 대응하는 폐치 그룹과 연관된 레코드일 수 있거나, 레코드들을 저장하기 위해 순차적 폐치 테이블이 사용되는 실시 예들의 경우 CTI의 타겟 명령어와 연관된 레코드일 수 있다. 일반적으로, CTI 또는 타겟 명령어 중 어느 하나를 접할 시(예를 들어, 어드레스 또는 명령어들과 연관된 다른 정보에 기초하여 폐치 그룹에서 CTI 또는 타겟 명령어를 검출 시), 순차적 폐치 로직은 적절한 레코드로부터 순차적으로 폐치될 CTI를 갖지 않는 폐치 그룹들의 수의 카운트를 획득한다. 상술한 바와 같이, 카운트는 특정 폐치 그룹들의 수를 나타내는 숫자, 문자열 또는 다른 값이 될 수 있다.
- [0064] 그 다음, 순차적 폐치 로직은 지정된 폐치 그룹들의 수 각각이 폐치되고 실행을 위해 준비될 때, 해당 폐치 그룹에서의 명령어들에 대한 분기 예측 정보를 획득하기 위한 분기 예측 유닛(예를 들어, 분기 예측 유닛(212))의 대응하는 액세스들을 방지한다(단계 704). 일반적으로, 이러한 동작 동안, 순차적 폐치 테이블은 브랜치 예측 유닛의 액세스들을 억제, 차단 또는 다른 방식으로 방지하며, 이에 의해 불필요하게 전력을 소비하는 것 등을 방지한다. 일부 실시 예들에서, 순차적 폐치 로직은 분기 예측 유닛의 액세스들을 방지하는 것 외에도, 또한 단계 704에서의 동작이 완료될 때까지(단계 702에 도시된 바와 같이) 카운트에 대한 확인을 수행하는 것을 방지한다 - 그리고 순차적 폐치 테이블과 같은 기능 블록들 또는 분기 예측 유닛의 일부 또는 전부를 저전력 모드에 둘 수 있다.
- [0065] 도 12는 일부 실시 예들에 따라 분기 예측 유닛의 액세스들을 방지하기 위해 지정된 수를 사용하여 CTI 다음에 순차적으로 폐치될 CTI들을 포함하지 않는 지정된 폐치 그룹들의 수를 결정하기 위한 프로세스를 도시하는 흐름도를 나타낸다. 도 12에 도시된 동작들은 일부 실시 예에 의해 수행되는 동작들의 일반적인 예로서 제시됨을 유념한다. 다른 실시 예들에 의해 수행되는 동작들은 상이한 동작들, 상이한 순서로 수행되는 동작들, 및/또는 상이한 기능적 블록들에 의해 수행되는 동작들을 포함한다. 도 12의 동작은 도 7의 단계들 702-704에 대한 일반적인 용어들로 설명되고 이에 따라 도 12는 단계들 702-704에 대해 보다 상세한 설명을 제공한다.
- [0066] 도 12에서의 동작들을 위해, CTI 명령어들(또는 보다 구체적으로는 이의 타겟 명령어)와 연관된 순차적 폐치 레코드들(즉, CTI들 이후 순차적으로 폐치될 CTI 명령어를 포함하지 않는 폐치 그룹들의 수를 나타내는 레코드들)이 도 7 및 도 10 내지 도 11에 대해 상술한 바와 같이 순차적 폐치 테이블에 저장된다고 가정한다. 그러나, 여기서의 다른 곳에서 설명된 바와 같이, 순차적 폐치 레코드들이 분기 타겟 버퍼에 저장되는 실시 예에 대해 유사한 동작들이 수행됨을 유념한다. 또한 프로그램 카운터(다음 PC 기능 블록에 의해 제공되는 바와 같은)는 단계 1200 전에, CTI에 대한 타겟 명령어를 포함하는 폐치 그룹이 폐치될 것임을 나타낸다고 가정한다. 또한 폐치 그룹들은 네 개의 명령어들을 포함한다고 가정한다. 그러나, 이러한 값들 및 조건들은 도 12에서의 예를 제시하기 위해 사용되는 것이고, 모든 실시 예들에서 동일하지는 않다.
- [0067] 도 12에서의 동작들은 순차적 폐치 로직(예를 들어, 순차적 폐치 로직(214))이 폐치 그룹에서의 명령어의 어드

레스 또는 식별자에 기초하여 순차적 페치 테이블로부터, 페치 그룹 다음에 순차적으로 페치될 지정된 페치 그룹들의 수를 획득할 때 시작된다(단계 1200). 이러한 동작을 위해, 순차적 페치 로직은 프로그램 카운터를 사용하여 순차적 페치 테이블에서의 페치 그룹에서의 네 개의 명령어들 중 어느 하나에 대해 일치 또는 "적중(hit)"이 있는지 여부를 확인한다. 예를 들어, 순차적 페치 로직은 순차적 페치 테이블에서의 엔트리들의 일부 또는 전부와 비교되는 프로그램 카운터(즉, 페치 어드레스)로부터 해시 값 또는 또 다른 인덱스 값을 컴퓨팅할 수 있다. 또 다른 예로서, 순차적 페치 로직은 프로그램 카운터에 기초하여 페치 그룹에서의 명령어들에 대한 모든 어드레스들을 컴퓨팅할 수 있고 각 컴퓨팅된 어드레스를 순차적 페치 테이블에서의 각 엔트리와 비교할 수 있다. 상술한 바와 같이, 페치 그룹에서의 명령어는 CTI에 대한 타겟 명령어이므로, 순차적 페치 테이블에서 일치가 찾아지고 적중이 발생한다. 이에 따라, 순차적 페치 로직은 순차적 페치 테이블에서의 일치하는 엔트리로부터 지정된 페치 그룹들의 수를 판독한다.

[0068] 그 다음, 순차적 페치 로직은 지정된 페치 그룹들의 수와 동일하게 카운터를 설정한다(단계 1202). 예를 들어, 순차적 페치 로직은 지정된 페치 그룹들의 수 또는 이의 표현을 전용 카운터 레지스터 또는 다른 메모리 위치에 저장할 수 있다.

[0069] 각 페치 그룹이 페치되고 실행을 위해 준비될 때, 순차적 페치 로직은 분기 예측 유닛의 대응하는 액세스를 방지한다(단계 1204). 예를 들어, 순차적 페치 로직은 하나 이상의 제어 신호를 어서트(assert)하여 분기 예측 유닛에서의 회로 요소들이 액세스 동작들을 수행하는 것을 방지할 수 있고/거나, 어드레스 또는 관련 값들이 분기 예측 유닛으로 전송되는 것을 방지할 수 있고/거나, 클록들을 중지할 수 있고/거나, 회로 요소들의 전원을 끌 수 있고/거나 분기 예측 유닛의 대응하는 액세스를 방지하기 위한 다른 동작들을 수행할 수 있다. 이러한 동작을 위해, 일부 실시 예들에서, 분기 방향 레졸루션, 분기 어드레스 획득 등과 같은 분기 예측 유닛의 둘 이상의 별도의 가능한 병렬 액세스들 각각은 모두 상술된 기술들의 일부 또는 전부를 사용하여 분기 예측 유닛에서의 각각의 기능 블록들이 관련 동작들을 수행하는 것을 방지함으로써 모두 방지된다.

[0070] 도 12에는 도시되지 않았지만, 일부 실시 예들에서, 카운터가 제로가 아닌 페치 그룹들에 대해서도 순차적 페치 테이블의 액세스들이 방지된다. 이는 페치 그룹들이 CTI들을 포함하지 않고 이에 따라 CTI들에 대한 타겟 명령어들을 포함하지 않는 것으로 알려져 있기 때문에, 이러한 조치가 불필요하게 된다. 일부 실시 예들에서, 순차적 페치 테이블은 카운터가 제로가 아닌 동안 절전 모드에 놓인다. 예를 들어, 순차적 페치 테이블은 클록들을 제어하는 것을 중지시키고(예를 들어, 클록 게이팅을 통해), 전력을 감소시키고, 인에이블 신호들을 디어써트시킬 수 있는 등이다.

[0071] 또한, 순차적 페치 로직은 각 페치 그룹이 페치되고 실행을 위해 준비될 때 카운터를 감소한다(단계 1206). 예를 들어, 순차적 페치 로직은 전용 카운터 레지스터 또는 다른 메모리 위치에서의 카운터의 값을 하나씩 감소시킬 수 있고, 카운터를 다음 낮은 값 또는 이의 표현 등으로 전환할 수 있는 등이다.

[0072] 카운터가 제로에 도달할 때까지(단계 1208), 순차적 페치 로직은 계속해서 분기 예측 유닛의 액세스들을 방지하고(단계 1204), 페치 그룹들이 페치되고 실행을 위해 준비될 때 카운터를 감소한다(단계 1206). 카운터가 제로에 도달한 후, 즉 지정된 페치 그룹들의 수 중 마지막 페치 그룹이 페치되고 실행을 위해 준비된 후, 하나 이상의 후속 페치 그룹이 페치되고 실행을 위해 준비될 때, 순차적 페치 로직은 분기 예측 정보를 획득하기 위한 분기 예측 유닛의 대응하는 액세스들을 허용한다(단계 1210). 다시 말해, 카운터가 제로와 동일할 때, 순차적 페치 로직은 분기 타겟 및 분기 방향 예측과 같은 일반적인 분기 예측 동작들이 수행되도록 허용한다. 이러한 방식으로, 순차적 페치 로직은 분기 예측 유닛(그리고 가능하게는 순차적 페치 테이블)의 불필요한 액세스들을 방지하기 위해 카운터가 제로가 아닐 때 분기 예측 액세스들(그리고 가능하게는 순차적 페치 테이블 액세스들)을 차단한다.

[0073] **다중 스레드 프로세서**

[0074] 일부 실시 예들에서, 전자 디바이스(100)에서의 프로세서(102)는 다중 스레드 프로세서이고 이에 따라 두 개 이상의 별도의 명령어 실행 스레드들을 지원한다. 일반적으로, 다중 스레드 프로세서는 각 별도의 스레드에 전용되는 기능 블록들 및/또는 하드웨어 구조들을 포함하지만, 스레드들 간에 공유되고/거나 하나보다 많은 스레드에 대해 각각의 동작들을 수행하는 기능 블록들 및/또는 하드웨어 구조들을 포함할 수도 있다. 예를 들어, 분기 예측 유닛 및 순차적 페치 유닛과 같은 기능 블록들은 모든 스레드들에 대해 각각 분기 예측 동작들 및 분기 예측 유닛 액세스 방지를 수행할 수 있다. 다른 예로서, 순차적 페치 테이블(순차적 페치 테이블을 사용하는 실시 예들에서)은 모든 스레드들(또는 다수의 스레드들의 일부 조합)에 사용되는 단일 순차적 페치 테이블로서 구현되거나 각 스레드가 대응하는 별도의 순차적 페치 테이블을 갖도록 스레드별로 구현될 수 있다. 이러한 실시 예

들에서, 각각의 순차적 페치 테이블에서의 레코드들은 관련 스프레드에 대해 유지되고 다른 스프레드들에 대한 순차적 페치 테이블들에 유지되는 레코드들과는 상이할 수 있다. 또 다른 예로서, 순차적 페치 로직은 스프레드별로 분기 예측 유닛의 액세스들을 방지할 수 있고, 이에 따라 여기서 설명된 바와 같이 대응하는 스프레드에 대해 분기 예측 유닛의 액세스들을 방지하기 위해 사용되는 각 스프레드에 대해 별도의 독립적인 카운터를 유지할 수 있다.

[0075] 상술한 바와 같이, 일부 실시 예들에서, CTI 다음 CTI를 포함하지 않는 순차적으로 페치된 페치 그룹들에 대한 분기 예측 유닛의 액세스들을 방지하는 동안, 순차적 페치 로직은 또한 순차적 페치 테이블의 액세스들을 방지하고 순차적 페치 테이블을 절전 모드에 둘 수 있다. 이러한 실시 예들에서, 다수의 스프레드들이 순차적 페치 테이블에 의존할 때(하나의 순차적 페치 테이블이 두 개 이상의 스프레드들에 대한 레코드들을 유지하는 데 사용될 때), 순차적 페치 테이블은 다른 스프레드들을 서비스하기 위해 전체 전력 모드/활성 상태로 유지될 수 있다(그리고 이에 따라 절전 모드로 전환되지 않을 것이다). 분기 예측 유닛에 대해서도 동일하게 적용되며; 단일 스프레드만이 분기 예측 유닛을 사용하고 있을 때, 분기 예측 유닛은 액세스들이 방지될 때 절전 모드에 놓일 수 있다. 그러나, 분기 예측 유닛이 둘 이상의 스프레드들에 의해 사용될 때, 분기 예측 유닛은 다른 스프레드들을 서비스하기 위해 최대 전력 모드/활성 상태로 남을 수 있다. 그러나, 여기에서 설명된 바와 같이 특정 스프레드들에 대해 특정 액세스들은 이루어지지 않는다.

[0076] 일부 실시 예들에서, 전자 디바이스(예를 들어, 전자 디바이스(100), 및/또는 이의 일부 부분)는 비일시적 컴퓨터 판독 가능 저장 매체 상에 저장된 코드 및/또는 데이터를 사용하여 여기에서 설명된 동작들 중 일부 또는 전부를 수행한다. 보다 구체적으로, 상기 전자는 컴퓨터 판독 가능한 저장 매체로부터 코드 및/또는 데이터를 판독하고 설명된 동작들을 수행할 때 코드를 실행하고/거나 데이터를 사용한다. 컴퓨터 판독 가능한 저장 매체는 전자 디바이스에 의한 사용을 위한 코드 및/또는 데이터를 저장하는 임의의 디바이스, 매체, 또는 이들의 조합일 수 있다. 예를 들어, 컴퓨터 판독 가능 저장 매체는 플래시 메모리, 랜덤 액세스 메모리(eDRAM, RAM, SRAM, DRAM, DDR, DDR2/DDR3/DDR4 SDRAM 등), 판독 전용 메모리(ROM) 및/또는 자기 또는 광학 저장 매체들(예를 들어, 디스크 드라이브들, 자기 테이프, CD, DVD들)을 포함하여, 휘발성 메모리 또는 비휘발성 메모리를 포함할 수 있지만, 이에 제한되지는 않는다.

[0077] 일부 실시 예들에서, 하나 이상의 하드웨어 모듈은 여기에서 설명된 동작들을 수행한다. 예를 들어, 하드웨어 모듈들은 하나 이상의 프로세서/코어/중앙 처리 장치들(CPU), 주문형 반도체(ASIC) 칩들, 필드 프로그래밍 가능 게이트 어레이들(FPGA), 컴퓨터 유닛들, 임베디드 프로세서들, 그래픽 프로세서들(GPU)/그래픽 코어들, 파이프라인들, 가속 처리 장치들(APU, Accelerated Processing Units) 및/또는 기타 프로그램 가능 로직 장치들을 포함할 수 있지만, 이에 제한되지는 않는다. 그러한 하드웨어 모듈들이 활성화될 때, 하드웨어 모듈들은 동작들의 일부 또는 전부를 수행한다. 일부 실시 예들에서, 하드웨어 모듈들은 동작들을 수행하기 위한 명령어들(프로그램 코드, 펌웨어 등)을 실행함으로써 구성되는 하나 이상의 범용 회로를 포함한다.

[0078] 일부 실시 예들에서, 여기에서 설명된 구조들 및 메커니즘들의 일부 또는 전부를 나타내는 데이터 구조(예를 들어, 프로세서(102), 메모리(104), 및/또는 이의 일부 부분)는 데이터베이스 또는 전자 디바이스에 의해 판독되고 구조들 및 메커니즘들을 포함하는 하드웨어를 제조하기 위해, 직접 또는 간접적으로 사용될 수 있는 다른 데이터 구조를 포함하는 비일시적 컴퓨터 판독 가능한 저장 매체 상에 저장된다. 예를 들어, 데이터 구조는 Verilog 또는 VHDL과 같은 상위 수준 설계 언어(HDL)의 하드웨어 기능에 대한 거동 수준 기술 또는 레지스터 전송 수준(RTL) 기술일 수 있다. 기술은 전술한 구조들 및 메커니즘들을 포함하는 하드웨어의 기능을 나타내는 합성 라이브러리로부터 게이트들/회로 요소들의 리스트를 포함하는 네트리스트를 생성하기 위해 설명을 합성할 수 있는 합성 툴에 의해 판독될 수 있다. 그 다음 네트리스트를 배치하고 라우팅하여 마스크들에 적용할 기하학적 모양들을 기술하는 데이터 세트를 생성할 수 있다. 그 다음 마스크들은 전술한 구조들 및 메커니즘들에 대응하는 반도체 회로 또는 회로들(예를 들어, 집적 회로들)을 생산하기 위해 다양한 반도체 제조 단계들에서 사용될 수 있다. 대안적으로, 컴퓨터 액세스 가능한 저장 매체 상의 데이터베이스는 네트리스트(합성 라이브러리가 있거나 없는) 또는 필요시, 데이터 세트, 또는 그래픽 데이터 시스템(GDS) II 데이터일 수 있다.

[0079] 본 설명에서, 변수들 또는 지정되지 않은 값들(즉, 값들의 특정 인스턴스들이 없는 값들에 대한 일반적인 설명들)은 N과 같은 문자들로 표현된다. 여기서 사용될 때, 본 설명에서의 상이한 위치들에서 동일한 문자들을 사용할 수 있음에도 불구하고, 각 경우에서 변수들 및 지정되지 않은 값들은 반드시 동일하지는 않다, 즉 일반적인 변수들 및 지정되지 않은 값들의 일부 또는 전부에 대해 상이한 가변량들 및 값들이 있을 수 있다. 다시 말해, 본 설명에서 변수들 및 지정되지 않은 값들을 나타내는 데 사용되는 N 및 임의의 다른 문자들은 반드시 서로 관

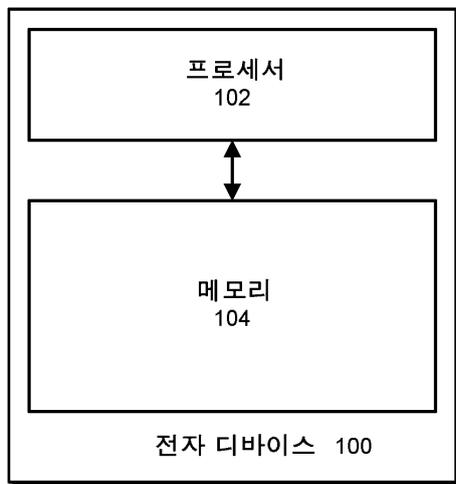
련이 있는 것은 아니다.

[0080] 표현 "등등" 또는 "등"은 여기서 사용될 때 및/또는 케이스, 즉 등이 연관되는 리스트의 요소들 중 "적어도 하나"의 등가물을 제시하는 것으로 의도된다. 예를 들어, "시스템이 첫 번째 동작, 두 번째 작업 등을 수행한다"라는 문장에서, 시스템은 첫 번째 동작, 두 번째 동작, 및 다른 동작들 중 적어도 하나를 수행한다. 또한, 등과 연관된 리스트의 요소들은 예들의 집합 중 예들일 뿐이다 - 그리고 예들 중 적어도 일부는 일부 실시 예들에서 나타나지 않을 수 있다.

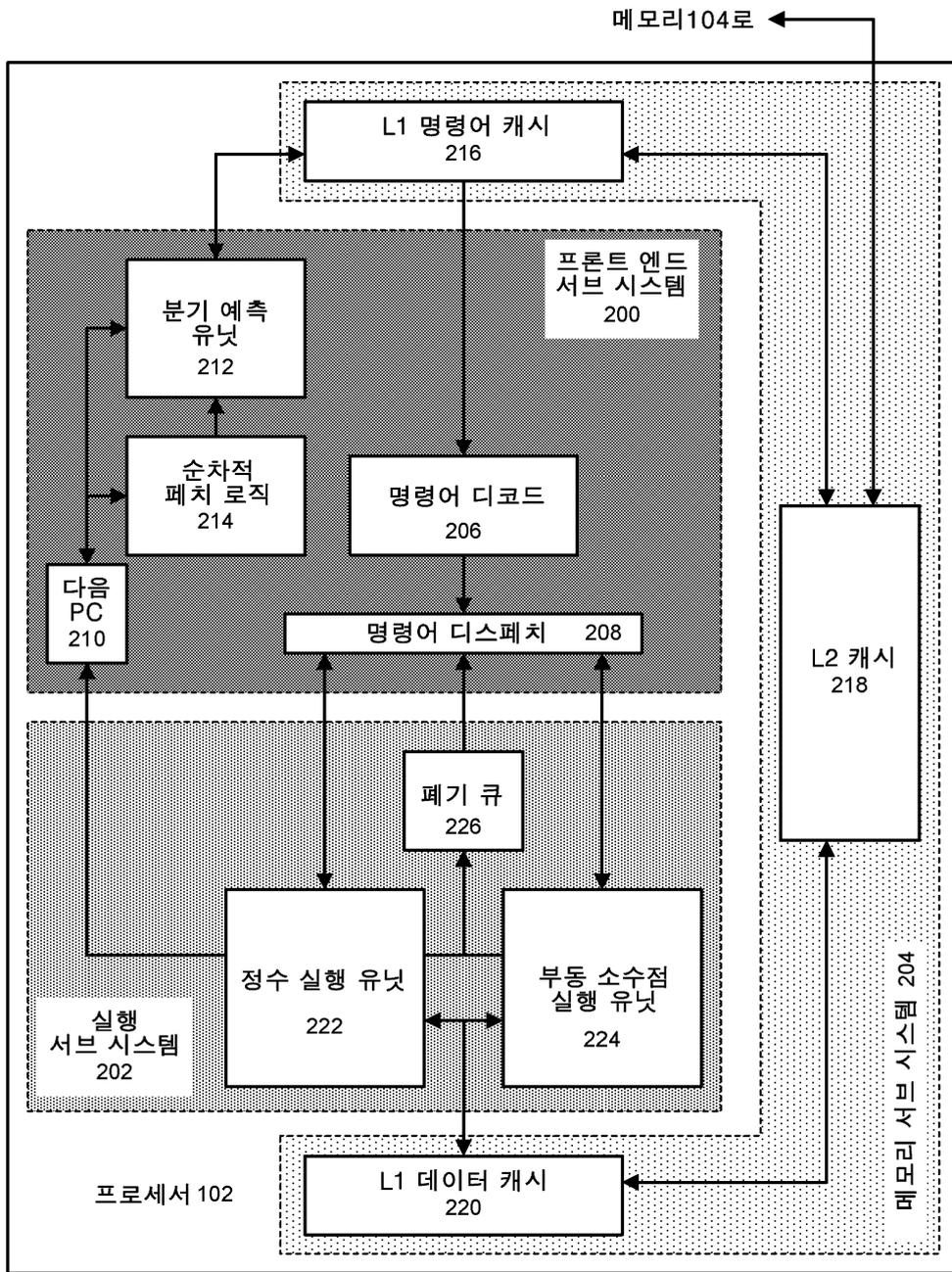
[0081] 실시 예들에 대한 앞에서의 설명들은 단지 예시 및 설명의 목적들로 제시되었다. 그것들은 철저하거나 본 실시 예들을 개시된 형태들로 제한하는 것으로 의도되지 않는다. 따라서, 많은 수정 및 변형이 당업자에게 명백할 것이다. 또한, 상기한 개시 내용은 실시 예들을 제한하려는 것이 아니다. 실시 예들의 범위는 첨부된 청구범위에 의해 한정된다.

도면

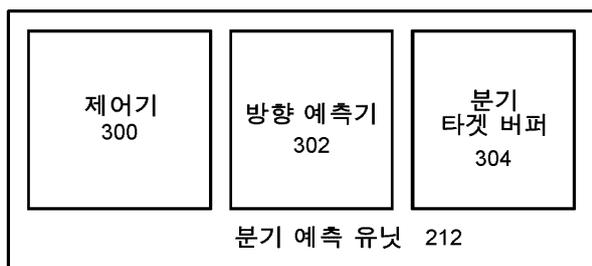
도면1



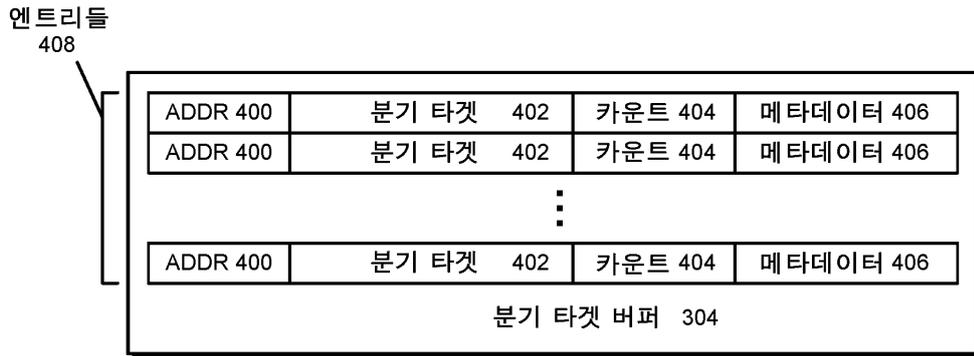
도면2



도면3



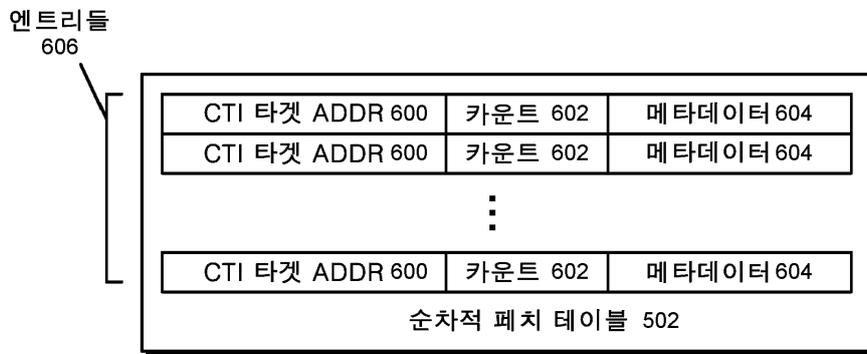
도면4



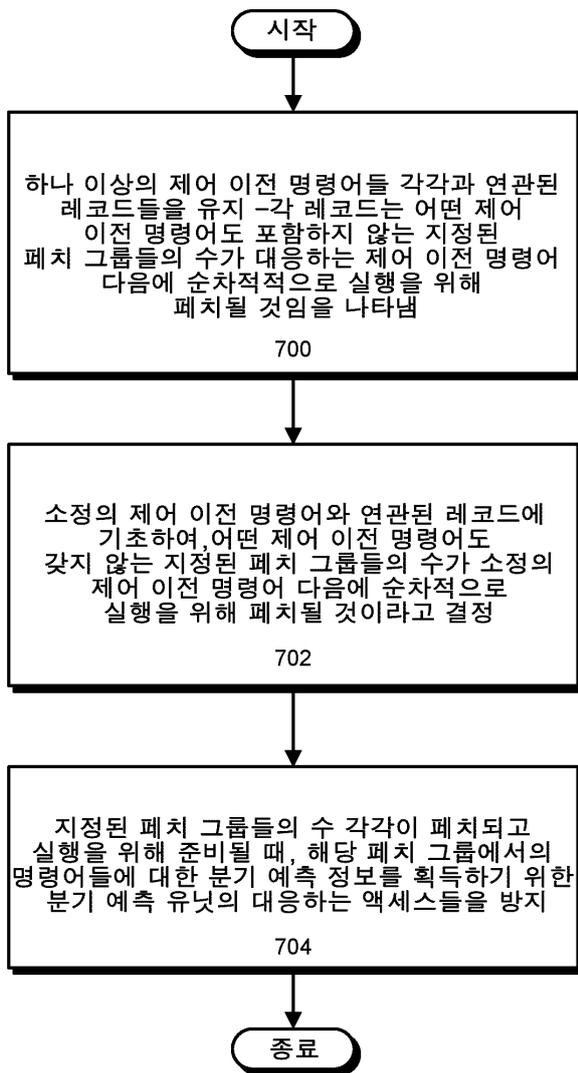
도면5



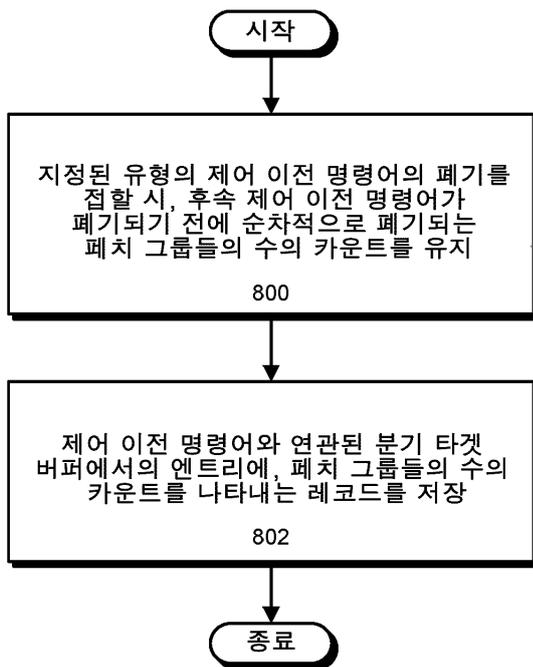
도면6



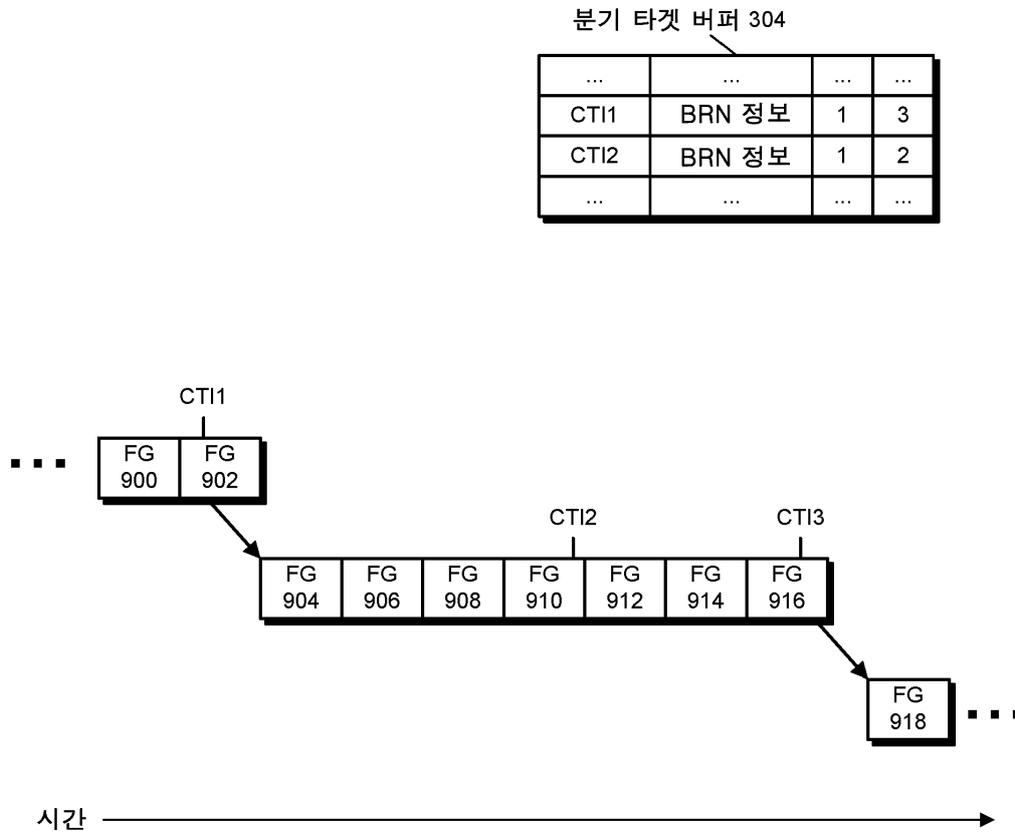
도면7



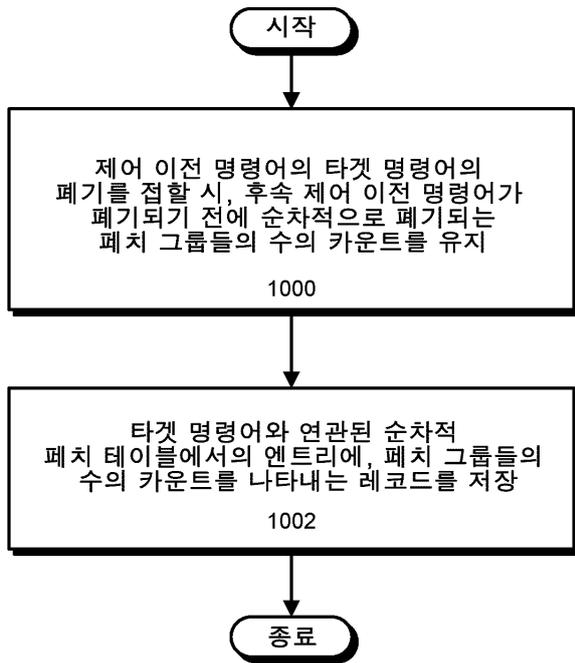
도면8



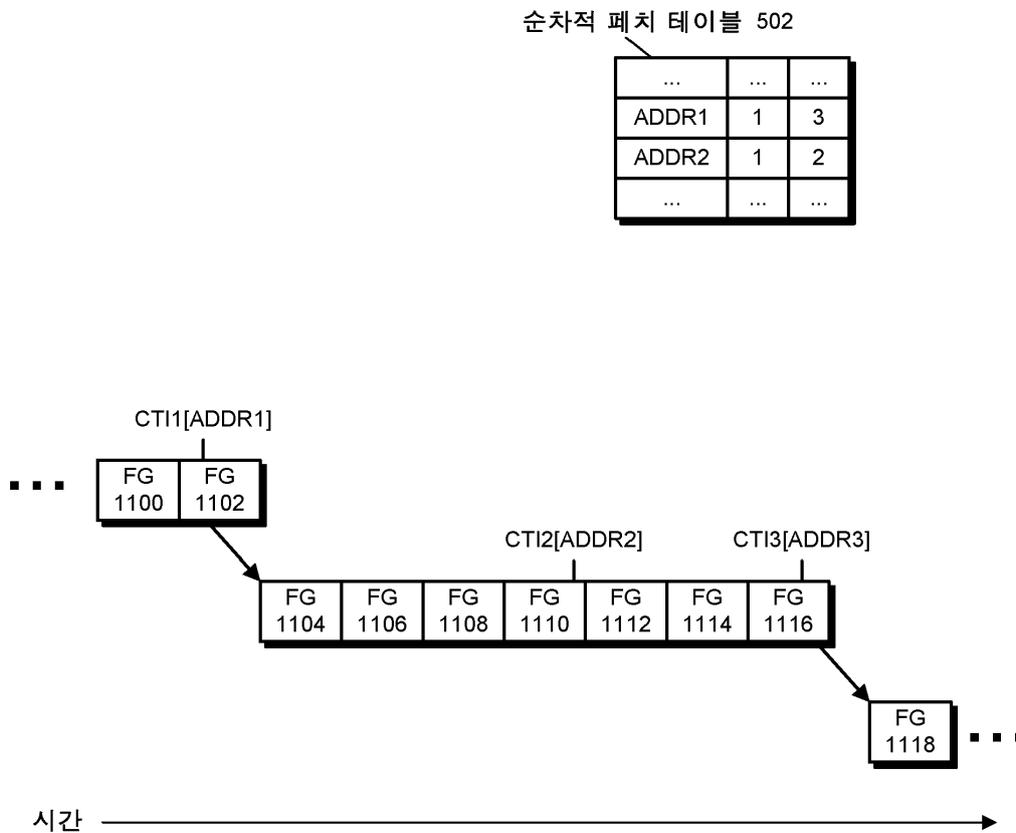
도면9



도면10



도면11



도면12

