



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0130513  
(43) 공개일자 2015년11월23일

- (51) 국제특허분류(Int. Cl.)  
*G06F 9/30* (2006.01) *G06F 9/38* (2006.01)
- (52) CPC특허분류  
*G06F 9/30058* (2013.01)  
*G06F 9/30054* (2013.01)
- (21) 출원번호 10-2015-7029133
- (22) 출원일자(국제) 2014년03월14일  
심사청구일자 없음
- (85) 번역문제출일자 2015년10월13일
- (86) 국제출원번호 PCT/US2014/029778
- (87) 국제공개번호 WO 2014/145101  
국제공개일자 2014년09월18일
- (30) 우선권주장  
13/833,844 2013년03월15일 미국(US)
- (71) 출원인  
퀄컴 인코포레이티드  
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자  
스미스, 로드니 웨인  
미국 92121 캘리포니아주 샌 디에고 모어하우스 드라이브 5775  
(뒷면에 계속)
- (74) 대리인  
특허법인 남엔드남

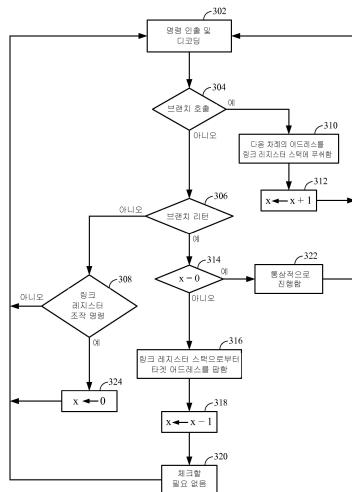
전체 청구항 수 : 총 19 항

(54) 발명의 명칭 프로세서에서 브랜치 리턴 명령들을 실행하는 속도를 개선시키기 위한 방법

**(57) 요 약**

링크 레지스터 스택을 활용함으로써, 프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 장치 및 방법이 개시된다. 프로세서는 브랜치 카운터를 포함하고, 이 브랜치 카운터는 0으로 초기화되고, 그리고 프로세서가 브랜치 호출 명령 이외의 링크 레지스터 조작 명령을 디코딩할 때마다, 0으로 셋팅된다. 브랜치 카운터는, 브랜치 호출 명령이 디코딩되고 어드레스가 링크 레지스터 스택에 푸쉬될 때마다 1씩 증가된다. 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 그리고 브랜치 카운터가 0이 아니라면, 디코딩된 브랜치 리턴 명령에 대한 타겟 어드레스가 링크 레지스터 스택으로부터 팝되고, 브랜치 카운터는 감소되며, 그리고 정확도를 위해 타겟 어드레스를 체크할 필요가 없다.

**대 표 도 - 도3**



(52) CPC특허분류  
*G06F 9/3806* (2013.01)

(72) 발명자

**맥클베인, 마이클 스코트**

미국 92121 캘리포니아주 샌 디에고 모어하우스 드  
라이브 5775

**스텝웰, 브라이언 마이클**

미국 92121 캘리포니아주 샌 디에고 모어하우스 드  
라이브 5775

**브라운, 멜린다 제이.**

미국 92121 캘리포니아주 샌 디에고 모어하우스 드  
라이브 5775

**스트리트, 다렌 유진**

미국 92121 캘리포니아주 샌 디에고 모어하우스 드  
라이브 5775

---

## 명세서

### 청구범위

#### 청구항 1

프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 방법으로서,

상기 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령들을 디코딩하는 것에 응답하여, 상태 머신을 초기 상태로 셋팅하는 단계; 및

브랜치 호출 명령 또는 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 상태 머신을, 상기 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령을 디코딩한 이후 상기 프로세서에 의해 디코딩된 브랜치 호출 명령들의 수를 표시하는 상태로 변경하는 단계 –상기 프로세서는 상기 디코딩된 브랜치 호출 명령들과 쌍을 이루는 브랜치 리턴 명령들을 디코딩하지 않았음–

를 포함하는,

프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 방법.

#### 청구항 2

제 1 항에 있어서,

상기 상태 머신의 상태가, 상기 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령을 디코딩한 이후 적어도 하나의 브랜치 호출 명령이 디코딩되었음을 표시한다면, 상기 프로세서가 브랜치 리턴 명령을 인출 및 디코딩하는 것에 응답하여, 링크 레지스터 스택으로부터 타겟 어드레스를 팝(pop)하고, 상기 타겟 어드레스를 다음 차례의 명령을 인출하기 위한 어드레스로서 사용하는 단계 –적어도 하나의 디코딩된 브랜치 호출 명령과 쌍을 이루는 적어도 하나의 브랜치 리턴 명령은 디코딩되지 않았음–; 및

정확도에 대해 상기 타겟 어드레스를 체크하는 것 없이, 상기 브랜치 리턴 명령의 실행을 완료하는 단계를 더 포함하는,

프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 방법.

#### 청구항 3

프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 방법으로서,

상기 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령들을 디코딩하는 것에 응답하여, 카운터를 초기 값으로 셋팅하는 단계;

상기 프로세서가 브랜치 호출 명령을 디코딩하는 것에 응답하여, 상기 카운터를 제1 상수만큼 증가시키는 단계; 및

상기 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 상기 카운터를 제2 상수만큼 증가시키는 단계를 포함하는,

프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 방법.

#### 청구항 4

제 3 항에 있어서,

상기 프로세서가 상기 카운터의 값이 상기 초기 값과 동일하지 않음을 결정하는 것에 응답하여, 그리고 상기 프로세서가 상기 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 링크 레지스터 스택으로부터 상기 브랜치 리턴 명령에 대한 타겟 어드레스를 팝하는 단계; 및

정확도에 대해 상기 타겟 어드레스를 체크하는 것 없이, 상기 브랜치 리턴 명령의 실행을 완료하는 단계

를 더 포함하는,

프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 방법.

### 청구항 5

제 4 항에 있어서,

상기 초기 값은 0이고,

상기 제1 상수는 1과 동일하고, 그리고

상기 제2 상수는 마이너스 1(negative one)과 동일한,

프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 방법.

### 청구항 6

제 3 항에 있어서,

상기 초기 값은 0이고,

상기 제1 상수는 1과 동일하고, 그리고

상기 제2 상수는 마이너스 1과 동일한,

프로세서에서 브랜치 호출 및 브랜치 리턴 명령들을 실행하기 위한 방법.

### 청구항 7

장치로서,

링크 레지스터;

브랜치 카운터; 및

프로세서

를 포함하고,

상기 프로세서는,

상기 링크 레지스터에 기록하는, 브랜치 호출 명령들 이외의 명령들을 디코딩하는 것에 응답하여, 상기 브랜치 카운터를 초기 값으로 셋팅하고;

브랜치 호출 명령을 디코딩하는 것에 응답하여, 상기 브랜치 카운터를 제1 상수만큼 증가시키고; 그리고

브랜치 리턴 명령을 디코딩하는 것에 응답하여, 상기 브랜치 카운터를 제2 상수만큼 증가시키는,

장치.

### 청구항 8

제 7 항에 있어서,

링크 레지스터 스택

을 더 포함하고,

상기 프로세서는,

상기 카운터의 값이 상기 초기 값과 동일하지 않음을 결정하는 것에 응답하여, 그리고 상기 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 상기 링크 레지스터 스택으로부터 상기 브랜치 리턴 명령에 대한 타겟 어드레스를 꺼내고; 그리고

정확도에 대해 상기 타겟 어드레스를 체크하는 것 없이, 상기 브랜치 리턴 명령의 실행을 완료하는,

장치.

### 청구항 9

제 8 항에 있어서,  
상기 초기 값은 0이고,  
상기 제1 상수는 1과 동일하고, 그리고  
상기 제2 상수는 마이너스 1과 동일한,  
장치.

### 청구항 10

제 7 항에 있어서,  
상기 초기 값은 0이고,  
상기 제1 상수는 1과 동일하고, 그리고  
상기 제2 상수는 마이너스 1과 동일한,  
장치.

### 청구항 11

제 7 항에 있어서,  
상기 장치는 셀룰러 전화 및 기지국으로 구성된 세트로부터 선택되는,  
장치.

### 청구항 12

프로세서로서,  
브랜치 카운터;  
링크 레지스터 스택;  
카운터를 셋팅하기 위한 수단 —상기 카운터를 셋팅하기 위한 수단은, 상기 프로세서가 브랜치 호출 명령들 이  
외의 링크 레지스터 조작 명령들을 디코딩하는 것에 응답하여, 상기 브랜치 카운터를 초기 값으로 셋팅함—; 및  
카운터를 증가시키기 위한 수단  
을 포함하고,  
상기 카운터를 증가시키기 위한 수단은, 상기 프로세서가 브랜치 호출 명령을 디코딩하는 것에 응답하여, 상기  
브랜치 카운터를 제1 상수만큼 증가시키고, 그리고 상기 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답  
하여, 상기 브랜치 카운터를 제2 상수만큼 증가시키는,  
프로세서.

### 청구항 13

제 12 항에 있어서,  
링크 레지스터 스택을 팝하기 위한 수단  
을 더 포함하고,  
상기 링크 레지스터 스택을 팝하기 위한 수단은, 상기 프로세서가 상기 브랜치 카운터의 값이 상기 초기 값과  
동일하지 않음을 결정하는 것에 응답하여, 그리고 상기 프로세서가 상기 브랜치 리턴 명령을 디코딩하는 것에  
응답하여, 상기 링크 레지스터 스택으로부터 상기 브랜치 리턴 명령에 대한 타겟 어드레스를 팝하고, 그리고  
상기 프로세서는, 정확도에 대해 상기 타겟 어드레스를 체크하는 것 없이, 상기 브랜치 리턴 명령의 실행을 완

료하는,

프로세서.

#### 청구항 14

제 13 항에 있어서,

상기 초기 값은 0이고,

상기 제1 상수는 1과 동일하고, 그리고

상기 제2 상수는 마이너스 1과 동일한,

프로세서.

#### 청구항 15

제 12 항에 있어서,

상기 초기 값은 0이고,

상기 제1 상수는 1과 동일하고, 그리고

상기 제2 상수는 마이너스 1과 동일한,

프로세서.

#### 청구항 16

비-일시적 컴퓨터-관독가능 매체로서,

프로세서에 의해 실행 가능한 명령들을 저장하고, 상기 프로세서는, 상기 명령들을 실행할 때 방법을 수행하며,

상기 방법은,

상기 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령들을 디코딩하는 것에 응답하여, 카운터를 초기 값으로 설정하는 단계;

상기 프로세서가 브랜치 호출 명령을 디코딩하는 것에 응답하여, 상기 카운터를 제1 상수만큼 증가시키는 단계; 및

상기 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 상기 카운터를 제2 상수만큼 증가시키는 단계를 포함하는,

비-일시적 컴퓨터-관독가능 매체.

#### 청구항 17

제 16 항에 있어서,

상기 방법은,

상기 프로세서가 상기 카운터의 값이 상기 초기 값과 동일하지 않음을 결정하는 것에 응답하여, 그리고 상기 프로세서가 상기 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 링크 레지스터 스택으로부터 상기 브랜치 리턴 명령에 대한 타겟 어드레스를 팝하는 단계; 및

정확도에 대해 상기 타겟 어드레스를 체크하는 것 없이, 상기 브랜치 리턴 명령의 실행을 완료하는 단계를 더 포함하는,

비-일시적 컴퓨터-관독가능 매체.

#### 청구항 18

제 17 항에 있어서,

상기 초기 값은 0이고,  
 상기 제1 상수는 1과 동일하고, 그리고  
 상기 제2 상수는 마이너스 1과 동일한,  
 비-일시적 컴퓨터-판독가능 매체.

### 청구항 19

제 16 항에 있어서,  
 상기 초기 값은 0이고,  
 상기 제1 상수는 1과 동일하고, 그리고  
 상기 제2 상수는 마이너스 1과 동일한,  
 비-일시적 컴퓨터-판독가능 매체.

## 발명의 설명

### 기술 분야

[0001] 본 발명의 실시예들은 마이크로프로세서들에 관한 것이다.

### 배경 기술

[0002] 브랜치 호출 명령(call branch instruction)은, 서브루틴(함수)을 구현하는 코드의 섹션으로 프로그램 제어가 분기하도록 허용한다. 예컨대, 브랜치 호출 명령이 실행될 때, 프로그램 카운터의 어드레스는 증가되고, 이후에 재호출되도록 링크 레지스터에 로딩되며, 그리고 호출된 서브루틴을 표현하는 코드의 섹션으로 프로그램 제어가 분기하도록, 호출된 서브루틴에 있는 코드의 첫 번째 라인의 어드레스가 프로그램 카운터에 로딩된다. 코드의 섹션의 끝에서의 브랜치 리턴 명령(return branch instruction)은 프로그램 제어가 링크 레지스터에 저장된 어드레스로 리턴하게 하며, 이로써 서브루틴을 호출했던 브랜치 호출 명령 이후의 프로그램 순서에서 다음 차례의 명령으로 리턴된다.

[0003] 네스트(nested) 또는 재귀(recursive) 서브루틴들(함수들) – 즉, 서브루틴이 다른 서브루틴(자신일 수 있음)을 호출함 – 을 구현하기 위해, 소프트웨어는 메모리에서 스택을 활용할 수 있다. 예컨대, 제1 서브루틴 호출로 인해 어드레스가 링크 레지스터에 로딩된 경우 그리고 제1 서브루틴을 표현하는 코드 내에서 제2 서브루틴이 호출되는 경우, 제1 서브루틴이 호출되었을 때 링크 레지스터에 로딩되었던 어드레스는 소프트웨어 컨벤션(software convention)에 의해 메모리에서 스택의 상단으로 푸쉬(push)되고, 제2 브랜치 명령 이후의 프로그램 순서에서 다음 차례의 명령을 표현하는 어드레스가 링크 레지스터에 로딩된다. 호출된 제2 서브루틴에 대한 브랜치 리턴 명령이 실행될 때, 링크 레지스터에 저장된 어드레스가 프로그램 카운터에 로딩되고, 이후, 제1 서브루틴이 완료될 때 제1 브랜치 명령의 (프로그램 순서에서) 바로 뒤의 명령으로 프로그램 제어가 리턴할 수 있도록, 스택의 상단에 있는 어드레스가 스택으로부터 팝(pop)되어 링크 레지스터에 로딩된다.

### 발명의 내용

[0004] 본 발명의 실시예들은, 링크 레지스터 스택을 활용하여, 브랜치 리턴 명령들을 실행하기 위한 프로세서 사이클들의 개수를 감소시키기 위한 시스템들 및 방법들에 관한 것이다.

[0005] 실시예에서, 방법은, 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령들을 디코딩하는 것에 응답하여, 상태 머신을 초기 상태로 셋팅하는 단계; 및 브랜치 호출 명령 또는 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 상태 머신을, 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령을 디코딩한 이후 프로세서에 의해 디코딩된 브랜치 호출 명령들의 수를 표시하는 상태로 변경하는 단계 – 프로세서는 대응하는 브랜치 리턴 명령들을 디코딩하지 않았음 – 를 포함한다. 방법은, 상태 머신의 상태가, 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령을 디코딩한 이후 적어도 하나의 브랜치 호출 명령이 디코딩되었음을 표시한다면, 프로세서가 브랜치 리턴 명령을 인출(fetch) 및 디코딩하는 것에 응답하여, 링크 레지스터

스택으로부터 타겟 어드레스를 팝하고, 타겟 어드레스를 다음 차례의 명령을 인출하기 위한 어드레스로서 사용하는 단계 – 대응하는 브랜치 리턴 명령은 디코딩되지 않았음–; 및 정확도에 대해 타겟 어드레스를 체크하는 것 없이, 브랜치 리턴 명령의 실행을 완료하는 단계를 더 포함한다.

[0006] 다른 실시예에서, 방법은, 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령들을 디코딩하는 것에 응답하여, 카운터를 초기 값으로 설정하는 단계; 프로세서가 브랜치 호출 명령을 디코딩하는 것에 응답하여, 카운터를 제1 상수만큼 증가시키는 단계; 및 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 카운터를 제2 상수만큼 증가시키는 단계를 포함한다. 방법은, 프로세서가 카운터의 값이 초기 값과 동일하지 않음을 결정하는 것에 응답하여, 그리고 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 링크 레지스터 스택으로부터 브랜치 리턴 명령에 대한 타겟 어드레스를 팝하는 단계; 및 정확도에 대해 타겟 어드레스를 체크하는 것 없이, 브랜치 리턴 명령의 실행을 완료하는 단계를 더 포함한다.

[0007] 다른 실시예에서, 장치는, 링크 레지스터; 브랜치 카운터; 및 프로세서를 포함하고, 이 프로세서는, 링크 레지스터에 기록하는, 브랜치 호출 명령들 이외의 명령들을 디코딩하는 것에 응답하여, 브랜치 카운터를 초기 값으로 설정하고; 브랜치 호출 명령을 디코딩하는 것에 응답하여, 브랜치 카운터를 제1 상수만큼 증가시키고; 그리고 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 브랜치 카운터를 제2 상수만큼 증가시킨다. 장치는, 링크 레지스터 스택을 더 포함하고, 프로세서는, 카운터의 값이 초기 값과 동일하지 않음을 결정하는 것에 응답하여, 그리고 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 링크 레지스터 스택으로부터 브랜치 리턴 명령에 대한 타겟 어드레스를 팝하고; 그리고 정확도에 대해 타겟 어드레스를 체크하는 것 없이, 실행을 완료하고 브랜치 리턴 명령을 회수(retire)한다.

[0008] 다른 실시예에서, 프로세서는, 브랜치 카운터; 링크 레지스터 스택; 카운터를 설정하기 위한 수단 – 카운터를 설정하기 위한 수단은, 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령들을 디코딩하는 것에 응답하여, 브랜치 카운터를 초기 값으로 설정함–; 및 카운터를 증가시키기 위한 수단을 포함하고, 카운터를 증가시키기 위한 수단은, 프로세서가 브랜치 호출 명령을 디코딩하는 것에 응답하여, 브랜치 카운터를 제1 상수만큼 증가시키고, 그리고 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 브랜치 카운터를 제2 상수만큼 증가시킨다. 프로세서는, 링크 레지스터 스택을 팝하기 위한 수단을 더 포함하고, 링크 레지스터 스택을 팝하기 위한 수단은, 프로세서가 브랜치 카운터의 값이 초기 값과 동일하지 않음을 결정하는 것에 응답하여, 그리고 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 링크 레지스터 스택으로부터 브랜치 리턴 명령에 대한 타겟 어드레스를 팝하고, 그리고 프로세서는, 정확도에 대해 타겟 어드레스를 체크하는 것 없이, 실행을 완료하고 브랜치 리턴 명령을 회수한다.

[0009] 다른 실시예에서, 스토리지 매체는, 프로세서에 의해 실행 가능한 명령들을 저장하고, 프로세서는, 명령들을 실행할 때 방법을 수행하며, 이 방법은, 프로세서가 브랜치 호출 명령들 이외의 링크 레지스터 조작 명령들을 디코딩하는 것에 응답하여, 카운터를 초기 값으로 설정하는 단계; 프로세서가 브랜치 호출 명령을 디코딩하는 것에 응답하여, 카운터를 제1 상수만큼 증가시키는 단계; 및 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 카운터를 제2 상수만큼 증가시키는 단계를 포함한다. 방법은, 프로세서가 카운터의 값이 초기 값과 동일하지 않음을 결정하는 것에 응답하여, 그리고 프로세서가 브랜치 리턴 명령을 디코딩하는 것에 응답하여, 링크 레지스터 스택으로부터 브랜치 리턴 명령에 대한 타겟 어드레스를 팝하는 단계; 및 정확도에 대해 타겟 어드레스를 체크하는 것 없이, 브랜치 리턴 명령의 실행을 완료하는 단계를 더 포함한다.

## 도면의 간단한 설명

[0010] 첨부된 도면들은, 본 발명의 실시예들의 설명을 돋기 위해 제시되고, 오로지 실시예들의 예시를 위해서 제공되며, 실시예들의 제한을 위해서는 제공되지 않는다.

[0011] 도 1은 실시예에 따른 마이크로프로세서를 추상화한다.

[0012] 도 2는 실시예에 따른 명령 흐름을 예시한다.

[0013] 도 3은 실시예에 따른 흐름도를 예시한다.

[0014] 도 4는 실시예를 포함하는 무선 통신 시스템이다.

## 발명을 실시하기 위한 구체적인 내용

[0015] 본 발명의 양상들은, 본 발명의 특정 실시예들에 관한 하기의 설명 및 관련 도면들에서 개시된다. 본

발명의 범위로부터 벗어남 없이, 대안적 실시예들이 창안될 수 있다. 부가하여, 본 발명의 잘 알려진 엘리먼트들은, 본 발명의 관련 세부사항들을 모호하게 하지 않기 위하여, 상세히 설명되지 않거나 또는 생략될 것이다.

[0012] 단어 "예시적"은 "예, 실례, 또는 예시로서의 역할을 하는"을 의미하도록 본원에서 사용된다. "예시적"인 것으로서 본원에서 설명되는 임의의 실시예가 반드시 다른 실시예들보다 바람직하거나 또는 유리한 것으로서 이해되어어서는 안된다. 마찬가지로, 용어 "본 발명의 실시예들"은, 본 발명의 실시예들 전부가 논의되는 특징, 장점 또는 동작 모드를 포함할 것을 요구하지 않는다.

[0013] 본원에 사용되는 용어는 단지 특정 실시예들을 설명하는 목적만을 위한 것이고, 본 발명의 실시예들의 제한인 것으로 의도되지 않는다. 본원에 사용된 바와 같이, 문맥이 그렇지 않다고 명확히 표시하지 않는 한, 단수 형태들은 복수 형태들을 또한 포함하도록 의도된다. 용어들 "포함한다", "포함하는", "구비한다" 및/또는 "구비하는"이 본원에서 사용될 때 진술되는 특징들, 정수들, 단계들, 동작들, 엘리먼트들, 및/또는 컴포넌트들의 존재를 특정하지만, 하나 또는 그 초과의 다른 특징들, 정수들, 단계들, 동작들, 엘리먼트들, 컴포넌트들, 및/또는 이들의 그룹들의 존재 또는 부가를 배제하지 않음이 추가로 이해될 것이다.

[0014] 추가로, 많은 실시예들이 예컨대 컴퓨팅 디바이스의 엘리먼트들에 의해 수행될 액션들의 시퀀스들 면에서 설명된다. 본원에 설명되는 다양한 액션들이 특정 회로들(예컨대, 애플리케이션 특정 집적 회로(ASIC)들)에 의해, 하나 또는 그 초과의 프로세서들에 의해 실행되고 있는 프로그램 명령들에 의해, 또는 둘 다의 결합에 의해 수행될 수 있음이 인식될 것이다. 부가하여, 본원에 설명되는 액션들의 이들 시퀀스는, 컴퓨터 명령들의 대응하는 세트를 그 내부에 저장한 임의의 형태의 비-일시적 컴퓨터-판독가능 스토리지 매체 내에서 완전히 구현되는 것으로 간주될 수 있고, 이 컴퓨터 명령들은, 실행 시, 연관된 프로세서로 하여금 본원에 설명되는 기능을 수행하게 할 것이다. 따라서, 본 발명의 다양한 양상들은 다수의 상이한 형태들로 구현될 수 있고, 이들 전부는 청구되는 발명의 요지의 범위 내에 있는 것으로 고려된다. 부가하여, 본원에 설명되는 실시예들 각각에 대해, 임의의 이러한 실시예들의 대응하는 형태는, 예컨대, 설명되는 액션을 수행"하도록 구성된 로직"으로서 본원에서 설명될 수 있다.

[0015] 실시예들은, 본원에서 링크 레지스터 스택으로 불리는 하드웨어 구조를 사용함으로써, 소프트웨어 스택의 사용을 개선시킬 것으로 예상된다. 적절한 상황들 하에서, 타겟 어드레스의 정확도를 검증할 필요 없이, 브랜치 리턴 명령에 대한 타겟 어드레스가 링크 레지스터 스택으로부터 파이프라인으로 포워딩될 수 있도록, 링크 레지스터 값들(어드레스들)을 저장하기 위해 링크 레지스터 스택이 사용된다.

[0020] 도 1은 실시예에 따른 마이크로프로세서(100)의 단순화된 추상이다. 파이프라인(102)이 데이터 캐시(104)로부터 데이터를 수신하고, 명령 캐시(106)로부터 명령들을 수신한다. 명령들을 디코딩 및 실행하기 위해, 제어기(108)가 파이프라인(102) 내의 다양한 스테이지들 및 실행 유닛들(예컨대, 인출 스테이지, 디코드 스테이지 등)을 제어한다. 실행된 명령들로부터의 결과들은, 그들이 완료될 때, 레지스터 파일(110), 데이터 캐시(104), 및 메모리(112)에 의해 표현된 다른 버퍼들 또는 메모리 유닛들에 저장될 수 있다. 메모리(112)는 메모리 계층을 표현할 수 있다. 배경기술에서 설명된 바와 같은 소프트웨어 스택은 추상 메모리 타입이고, 113으로 라벨링된다.

[0021] 레지스터 파일(110)은 복수의 레지스터들을 포함하는데, 도 1에는 세 개의 레지스터들: 프로그램 카운터 레지스터(114); 링크 레지스터(116); 및 소프트웨어 스택(113)의 상단에 대한 포인터를 저장하기 위한 스택 포인터 레지스터(118)가 예시된다.

[0022] 링크 레지스터 스택은, 도 1에서 120으로 라벨링된 하드웨어 메모리 구조로서 예시된다. 링크 레지스터 스택(120)은 마이크로프로세서(100)와 동일한 다이(칩) 상에 위치된 레지스터 파일일 수 있고, 레지스터 파일(110)과는 별개의 구조로서 도시된다. 실제, 링크 레지스터 스택(120)은 레지스터 파일(110)의 일부일 수 있지만, 링크 레지스터 스택(120)이 마이크로프로세서(100)의 아키텍처링된 레지스터들의 일부가 아닐 수 있음을 표시하기 위해, 링크 레지스터 스택(120)은 별개의 구조로서 도시된다. 링크 레지스터 스택(120)은 124로 라벨링된 브랜치 카운터를 포함한다. 브랜치 카운터(124)의 역할은 이후에 논의된다.

[0023] 실시예의 경우, 제어기(108)는, 파이프라인(102)이 브랜치 호출 명령을 디코딩할 때, (프로그램 순서에서) 다음 차례의 명령의 어드레스가 링크 레지스터 스택(120)에 푸쉬되도록 구성된다. (배경기술에서 논의된 바와 같이, 이 어드레스는 또한 링크 레지스터(116)에 값으로서 저장되는데, 링크 레지스터(116)에 저장된 이전 값은 소프트웨어 스택(113)에 푸쉬되고, 스택 포인터 레지스터(118)에 저장된 어드레스에 의해 포인팅된다.)

[0024] 서브루틴(함수)이 호출되는, 소프트웨어 프로그램의 각각의 브랜치 호출 명령에 대해, 서브루틴의 끝에

는 브랜치 리턴 명령이 존재한다. 브랜치 명령들의 한 쌍으로서 각각의 이러한 브랜치 호출 명령 및 브랜치 리턴 명령이 보일 수 있다. 즉, 브랜치 리턴 명령은 브랜치 호출 명령과 쌍을 이룬다. 이 점을 염두에 두고, 브랜치 카운터(124)는 디코딩된 브랜치 호출 명령들의 수를 추적하기 위한 구조로서 설명될 수 있는데, 파이프라인(102)은 디코딩된 브랜치 호출 명령들과 쌍을 이루는 브랜치 리턴 명령들을 디코딩하지 않았고, 파이프라인(102)은 링크 레지스터 조작 명령에 맞닥뜨리지 않았다. 링크 레지스터 조작 명령은 링크 레지스터(116)에 기록하는, 브랜치 호출 명령 이외의 임의의 명령을 지칭한다.

[0021] [0025] 브랜치 카운터(124)는 0으로 초기화되고, 파이프라인(102)이 링크 레지스터 조작 명령을 디코딩할 때마다, 0으로 설정된다. 브랜치 호출 명령이 디코딩되고 명령 어드레스가 링크 레지스터 스택(120)에 푸쉬될 때마다, 브랜치 카운터(124)의 값이 1만큼 증가되고; 그리고 브랜치 리턴 명령이 디코딩되고 명령 어드레스가 링크 레지스터 스택(120)으로부터 팝될 때마다, 브랜치 카운터(124)의 값이 1만큼 감소된다.

[0022] [0026] 제1 명령 어드레스가 링크 레지스터 스택(120)에 푸쉬된 이후 어떠한 링크 레지스터 조작 명령들도 존재하지 않는 한, 링크 레지스터 스택(120)에 저장된 값들(어드레스들)은 링크 레지스터(116)에 기록된 값들(어드레스들)의 정확한 히스토리를 표현한다. 따라서, 브랜치 카운터(124)의 값이 0을 초과하는 한, 링크 레지스터 스택(120)의 상태는 링크 레지스터(116)의 정확한 히스토리를 표현한다.

[0023] [0027] 실시예의 경우, 제어기(108)는, 파이프라인(102)이 브랜치 리턴 명령을 디코딩할 때, 그리고 브랜치 카운터(124)에 저장된 값이 0을 초과한다면, 링크 레지스터 스택(102)의 상단에 있는 어드레스가, 디코딩된 브랜치 리턴 명령에 대한 타겟 어드레스로서 파이프라인(102)에 포워딩되도록 구성된다. 그 이유는 브랜치 카운터(124)에 저장된 값이 0을 초과하기 때문에, 링크 레지스터 스택(102)으로부터 팝되는 어드레스는 정확한 타겟 어드레스이고, 그러므로 실행을 완료하고 회수하려는 디코딩된 브랜치 리턴 명령에 대해, 정확도를 위한 체크는 수행될 필요가 없다.

[0024] [0028] 그러나, 파이프라인(102)이 브랜치 리턴 명령을 디코딩할 때 브랜치 카운터(124)의 값이 0일 경우에는, 링크 레지스터 스택(120)의 상태가 링크 레지스터(116)의 정확한 히스토리를 표현한다는 보장이 없고, 이 경우, 제어기(108)는 파이프라인(102)을 통상적인 방식으로 관리하며, 이로써 배경기술에서 논의된 바와 같이, 예측되는 타겟 어드레스를 제공하기 위해 소프트웨어 스택(113)이 사용되고, 디코딩된 브랜치 리턴 명령이 실행을 완료하고 회수하도록 허용되기 이전에, 정확도에 대한 체크가 수행된다.

[0025] [0029] 브랜치 카운터(124)를 사용함으로써, 링크 레지스터(116)를 조작하는(링크 레지스터(116)에 기록하는), 끼어드는 링크 레지스터 조작 명령들이 존재하지 않는 한, 실시예들은, 호출된 서브루틴이 다른 서브루틴을 호출하는 네스트 서브루틴들의 경우에 타겟 어드레스들에 대해 링크 레지스터 스택(120)을 사용할 수 있다.

[0026] [0030] 이러한 실시예에 따른 프로세서는 프로세서 사이클들을 절약하는 것으로 예상되고, 이로써 전력과 시간이 절약된다.

[0027] [0031] 도 2는 코드의 세그먼트에서 명령 흐름을 고려함으로써 실시예를 예시한다. 문자들 A 내지 L은 코드 세그먼트의 명령들을 표현한다. 도 2의 실시예를 설명하는 목적들을 위해, 변수 x가 브랜치 카운터(124)의 값을 표기한다. 코드 세그먼트를 실행하기 이전에, 라벨 200에 의해 표시된 바와 같이, 브랜치 카운터(124)의 값은 0으로 설정된다.

[0028] [0032] 도 2에 예시된 코드 세그먼트의 첫 번째 명령은 명령 A, 즉 L1으로 라벨링된 명령으로의 브랜치 호출 명령이고, 이 L1으로 라벨링된 명령은 도 2의 명령 D이다. 브랜치 호출 명령 A를 디코딩 및 실행 시, 명령 B에 대한 어드레스가 링크 레지스터(116)에 기록되고, 또한 링크 레지스터 스택(120)에 푸쉬된다. 브랜치 카운터(124)가 1만큼 증가되고, 이는 도 2의 라벨 201에 의해 표시된다. 브랜치 카운터(124)의 값은 이제 1이다.

[0029] [0033] 프로그램 카운터 레지스터(114)는 이제 명령 D의 어드레스로 설정되고, 202로 라벨링된 화살표에 의해 표시된 바와 같이, 파이프라인(102)에서의 명령 흐름은 명령 A로부터 명령 D로 진행된다. 이후, 도 2의 204, 206, 및 208로 라벨링된 화살표들에 의해 표시된 바와 같이, 명령 흐름은 명령 D로부터 명령 E로, 명령 E로부터 명령 F로, 그리고 명령 F로부터 명령 G로 진행된다.

[0030] [0034] 명령 G는 L2로 라벨링된 명령으로의 브랜치 호출 명령이고, 이 L2로 라벨링된 명령은 도 2의 명령 J이다. 명령 J의 어드레스가 프로그램 카운터 레지스터(114)에 로딩되고, 210으로 라벨링된 화살표에 의해 표시된 바와 같이, 제어가 명령 J로 가게 된다. 명령 H에 대한 어드레스가 링크 레지스터 스택(120)에 푸쉬되고, 링크 레지스터(116)에 기록된다. 라벨 212에 의해 표시된 바와 같이, 브랜치 카운터(124)는 1만큼 증가된다.

브랜치 카운터(124)의 값은 이제 2와 동일하다.

[0031] [0035] 프로그램 카운터 레지스터(114)가 명령 J의 어드레스로 셋팅되었기 때문에, 214 및 216으로 라벨링된 화살표들에 의해 표시된 바와 같이, 파이프라인(102)에서의 명령 흐름은 명령 J로부터 명령 K로, 이후, 명령 K로부터 명령 L로 진행된다.

[0032] [0036] 명령 L은 브랜치 리턴 명령이고, 그리고 브랜치 카운터(124)의 값(도 2의 x)이 0을 초과하기 때문에, 링크 레지스터 스택(120)의 상태는 정확한 것으로 알려지며, 결과적으로, 링크 레지스터 스택(120)의 상단으로부터 팝되는 타겟 어드레스는 정확하다. 그러므로, 명령 L이 실행 및 완료될 때, 팝된 어드레스를 정확도에 대해 체크할 필요가 없다. 이러한 팝된 어드레스는 명령 H에 대한 어드레스이다. 라벨 218에 의해 표시된 바와 같이, 명령 L을 디코딩 및 실행 시, 브랜치 카운터(124)는 1만큼 감소된다. 브랜치 카운터(124)의 값은 이제 1이다.

[0033] [0037] 도 2에 예시된 예를 계속 하면, 명령 L에 대한 타겟이 명령 H의 어드레스이기 때문에 –여기서, 명령 H는 프로그램 순서에서 브랜치 호출 명령 G의 바로 뒤의 명령임–, 220으로 라벨링된 화살표에 의해 표시된 바와 같이, 프로그램 제어가 명령 H로 가게 된다.

[0034] [0038] 명령 H가 브랜치 리턴 명령으로서 디코딩되고, 그리고 브랜치 카운터(124)의 값이 0을 초과하기 때문에, 링크 레지스터 스택(120)의 상태가 정확함이 알려지고, 그러므로 링크 레지스터 스택(120)으로부터 팝된 타겟 어드레스는 정확한 타겟 어드레스이다. 이 타겟 어드레스는 명령 B에 대한 명령 어드레스이고, 명령 H가 실행 및 완료할 때, 이 타겟 어드레스는 정확도에 대해 체크될 필요가 없다. 명령 H의 실행 시, 라벨 222에 의해 표시된 바와 같이, 브랜치 카운터(124)는 1만큼 감소되고, 224로 라벨링된 화살표에 의해 표시된 바와 같이, 제어가 명령 B로 가게 된다. 브랜치 카운터(124)의 값은 이제 0이다.

[0035] [0039] 앞서 설명된 바와 같이, 제어기(108)는, 링크 레지스터 조작 명령, 즉 링크 레지스터(116)에 기록하는, 브랜치 호출 명령 이외의 명령을 디코딩 및 실행 시, 브랜치 카운터(124)의 값이 0으로 셋팅되도록 구성된다. 브랜치 리턴 명령이 디코딩되고 브랜치 카운터(124)의 값이 0일 때 실행을 시작하는 경우, 링크 레지스터 스택(120)의 상태가 정확하다는 보장이 없다. 그러므로, 예측되는 타겟 어드레스가 소프트웨어 스택(113)으로부터 팝되는 경우, 브랜치 리턴 명령이 실행을 완료하고 회수하기 이전에, 이 예측되는 타겟 어드레스는 정확도에 대해 체크되어야 한다.

[0036] [0040] 도 3은 실시예에 따른 흐름도를 예시한다. 302로 라벨링된 단계로부터 시작하여, 명령이 명령 캐시(106)로부터 인출되고 디코딩될 때, 도 3의 결정 흐름은 디코딩된 명령이 브랜치 호출 명령인지, 브랜치 리턴 명령인지, 또는 링크 레지스터 조작 명령인지에 따라 좌우된다. 일반성의 손실 없이, 이를 세 개의 명령들 중 어느 명령이 디코딩된 명령인지를 결정할 때의 순서가 도 3에서 라벨들 304, 306, 및 308로 표시된다. 그러나, 다른 실시예들에 대해, 이를 결정들이 도 3에 표시된 바와 같이 순서화되지 않을 수 있음이 이해되어야 한다.

[0037] [0041] 단계 304를 참조하면, 명령이 브랜치 호출 명령인지의 여부에 관해 결정이 이루어진다. 명령이 브랜치 호출 명령일 경우, 단계 310에서 표시된 바와 같이, (프로그램 순서에서) 다음 차례의 명령의 어드레스가 링크 레지스터 스택(120)에 푸쉬되고, 단계 312에 의해 표시된 바와 같이, 브랜치 카운터(124)의 값이 1만큼 증가된다( $x \leftarrow x + 1$ ). 도 3은 단계 312의 액션을 구현한 이후 단계 302로 리턴하는 흐름도를 도시한다. 단계 312로부터 단계 302로의 데이터 흐름이, 단계 312 이후 실제 실시예 파이프라인에서 취해지는 다음 차례의 액션이 반드시 인출 명령임을 암시하는 것이 아니라, 데이터 흐름은, 실시예에 의해 취해지는 액션들을 설명하기 위하여, 단지 편의를 위해 이와 같이 예시된다.

[0038] [0042] 단계 306을 참조하면, 명령이 브랜치 리턴 명령인지의 여부에 관해 결정이 이루어진다. 명령이 브랜치 리턴 명령이고, 브랜치 카운터(124)의 값이 0을 초과하는 경우(단계 314), 파이프라인(102)에 포워딩되고 프로그램 카운터 레지스터(114)에 기록되도록 타겟 어드레스가 링크 레지스터 스택(120)으로부터 팝되고(단계 316); 브랜치 카운터(124)는 1만큼 감소되며(단계 318); 명령이 실행 및 완료할 때, 정확도에 대해 타겟 어드레스를 체크할 필요가 없다(단계 320).

[0039] [0043] 그러나, 단계 314에서, 브랜치 카운터(124)의 값이 0임이 결정되는 경우, 마이크로프로세서(100)는 통상적인 방식으로 진행하고, 이로써 예측되는 타겟 어드레스가 사용되는 경우, 이 예측되는 타겟 어드레스는 정확도에 대해 체크되어야 한다(단계 322).

[0040] [0044] 도 3은 단계들 320 또는 322의 액션들을 구현한 이후 단계 302로 리턴하는 흐름도를 도시하지만, 앞서

논의된 바와 같이, 이러한 프로세스 흐름들은 단지 편의를 위해 도입된다.

[0041] [0045] 단계 308을 참조하면, 명령이 링크 레지스터(116)를 조작하는(예컨대, 링크 레지스터(116)에 기록하는) (브랜치 호출 명령 이외의) 임의의 명령인지의 여부에 관해 결정이 이루어진다. 명령이 링크 레지스터(116)를 조작하는 임의의 명령일 경우, 단계 324에서 표시된 바와 같이, 브랜치 카운터(124)의 값은 0으로 리셋된다( $x \leftarrow 0$ ). 그렇지 않다면, 제어가 다시 단계 302로 가게 된다. 다시, 단계들 308 및 324의 액션들을 구현한 이후 단계 302로 리턴하는 흐름도는 단지 편의를 위해 도입된다.

[0042] [0046] 우리는 브랜치 카운터(124)가 0을 지나서 감소될 수 없다는 컨벤션을 가정한다. 즉, 파이프라인(102)이 브랜치 리턴 명령을 디코딩하지만, 파이프라인(102)이 브랜치 호출 명령 이외의 링크 레지스터 조작 명령을 디코딩했었기 때문에 브랜치 카운터(124)가 이미 0으로 셋팅된 경우가 발생할 수 있다. 그 경우, 브랜치 카운터(124)는 0에서 머무른다.

[0043] [0047] 일반적으로, 상태 머신이 브랜치 카운터(124)의 액션들을 구현할 수 있음이 인식되어야 하는데, 상태는, 끼어드는 링크 레지스터 조작 명령을 디코딩한 이후 프로세서(100)에 의해 디코딩된 브랜치 호출 명령들의 수를 추적하고, 디코딩된 브랜치 호출 명령들과 쌍을 이루는 브랜치 리턴 명령들은 아직 디코딩되지 않았다.

[0044] [0048] 예컨대, 카운터가, 브랜치 호출 명령이 디코딩될 때마다 1만큼 증가되거나, 브랜치 리턴 명령이 디코딩될 때마다 1만큼 감소되거나, 또는 링크 레지스터 조작 명령이 디코딩될 때마다 0으로 리셋되는 것은 필요하지 않다. 특정 예로서, 카운터의 방향(direction)이 반전될 수 있는데, 카운터는 자신의 최대 값으로 초기화되고, 브랜치 호출 명령이 디코딩될 때마다 감소되고, 그리고 브랜치 리턴 명령이 디코딩될 때마다 증가된다. 따라서, 도 3에 의해 표현된 흐름도는 이 특정 예에 대해 조정될 수 있다.

[0045] [0049] 도 3에 대하여 설명된 액션들은 제어기(108)에서 실현되는 하드웨어로 구현될 수 있다. 몇몇 실시예들에 대해, 도 3에 예시된 액션들 중 일부 또는 전부가 제어기(108)에서 실행되는 명령들에 의해 실현될 수 있는데, 이러한 명령들은 펌웨어 또는 소프트웨어로서 보이고, 이러한 명령들은 메모리(112)에 저장될 수 있다.

[0046] [0050] 도 4는 실시예들이 애플리케이션을 발견할 수 있는 무선 통신 시스템을 예시한다. 도 4는 기지국들(404A, 404B, 및 404C)을 포함하는 무선 통신 네트워크(402)를 예시한다. 도 4는 406으로 라벨링된 통신 디바이스를 도시하고, 이 통신 디바이스는 모바일 셀룰러 통신 디바이스, 예컨대, 소위 스마트폰, 태블릿, 또는 셀룰러 전화 네트워크에 적절한 어떤 다른 종류의 통신 디바이스, 예컨대 컴퓨터일 수 있다. 통신 디바이스(406)는 모바일일 필요가 없다. 도 4의 특정 예에서, 통신 디바이스(406)는 기지국(404C)과 연관된 셀 내에 위치된다. 화살표들 408 및 410은 각각 업링크 채널 및 다운링크 채널을 그림으로 표현하며, 이 업링크 채널 및 다운링크 채널에 의해, 통신 디바이스(406)가 기지국(404C)과 통신한다.

[0047] [0051] 실시예들은, 예컨대, 통신 디바이스(406), 또는 기지국(404C), 또는 둘 다와 연관된 데이터 프로세싱 시스템들에서 사용될 수 있다. 도 4는, 본원에서 설명된 실시예들이 사용될 수 있는 많은 애플리케이션들 중에서 단 한 개의 애플리케이션을 예시한다.

[0048] [0052] 당업자들은 정보 및 신호들이 다양한 상이한 기술들 및 기법들 중 임의의 것을 사용하여 표현될 수 있음을 인식할 것이다. 예컨대, 위의 설명 전체에 걸쳐 참조될 수 있는 데이터, 명령들, 커맨드들, 정보, 신호들, 비트들, 심볼들, 및 칩들은 전압들, 전류들, 전자기파들, 자기 필드들 또는 입자들, 광학 필드들 또는 입자들, 또는 이들의 임의의 결합에 의해 표현될 수 있다.

[0049] [0053] 추가로, 당업자들은, 본원에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리적 블록들, 모듈들, 회로들, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 둘 다의 결합들로서 구현될 수 있음을 인식할 것이다. 하드웨어 및 소프트웨어의 이러한 상호 교환 가능성을 명확히 예시하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들, 및 단계들이 위에서 일반적으로 그들의 기능 면에서 설명되었다. 이러한 기능이 하드웨어로서 구현되는지 또는 소프트웨어로서 구현되는지는, 특정 애플리케이션, 및 전체 시스템에 부과되는 설계 제한들에 따라 좌우된다. 당업자들은 설명된 기능을 각각의 특정 애플리케이션에 대해 다양한 방식들로 구현할 수 있지만, 이러한 구현 결정들이 본 발명의 범위로부터 벗어남을 유발하는 것으로서 해석되어서는 안된다.

[0050] [0054] 본원에 개시된 실시예들과 관련하여 설명된 방법들, 시퀀스들 및/또는 알고리즘들은 직접적으로 하드웨어로, 프로세서에 의해 실행되는 소프트웨어 모듈로, 또는 둘의 결합으로 구현될 수 있다. 소프트웨어 모듈은 RAM 메모리, 플래시 메모리, ROM 메모리, EPROM 메모리, EEPROM 메모리, 레지스터들, 하드 디스크, 탈착 가능한 디스크, CD-ROM, 또는 기술분야에서 알려진 임의의 다른 형태의 스토리지 매체에 있을 수 있다. 도 1의 특정

예에서, 소프트웨어 모듈 또는 모듈들은 메모리(112)에 저장될 수 있다. 예시적 스토리지 매체가 프로세서에 커플링되고, 따라서 프로세서는 스토리지 매체로부터 정보를 관독할 수 있고 정보를 스토리지 매체에 기록할 수 있다. 대안적으로, 스토리지 매체는 프로세서에 통합될 수 있다.

[0051]

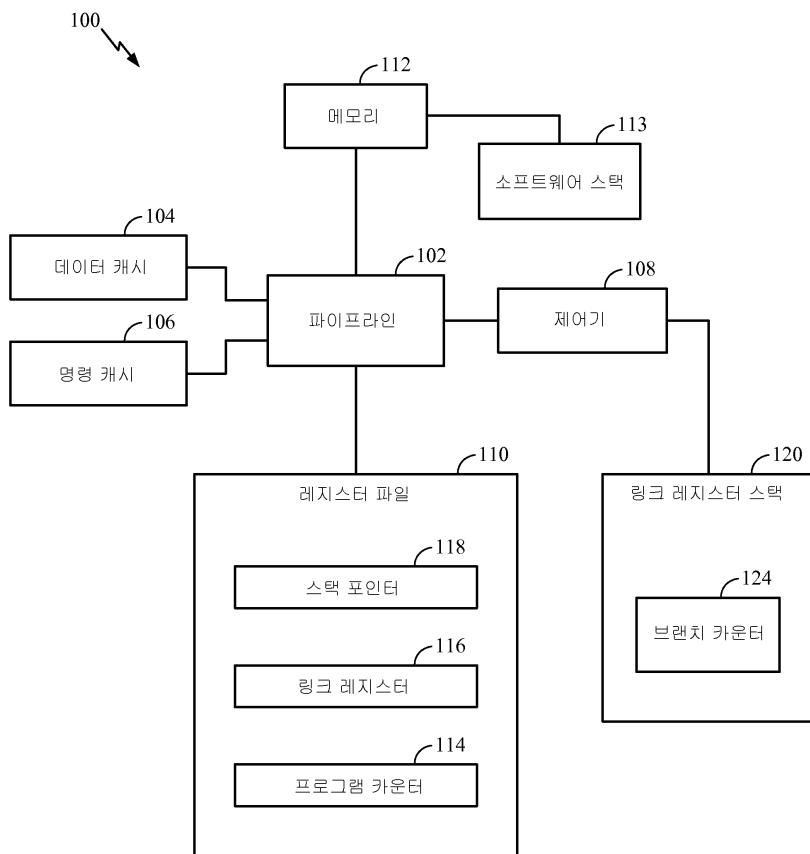
[0055] 따라서, 본 발명의 실시예는, 브랜치 리턴 명령들을 실행하는 속도를 개선시키기 위한 방법을 구현하는 비-일시적 컴퓨터-관독가능 매체를 포함할 수 있다. 따라서, 본 발명은 예시된 예들로 제한되지 않으며, 본원에 설명된 기능을 수행하기 위한 임의의 수단이 본 발명의 실시예들에 포함된다.

[0052]

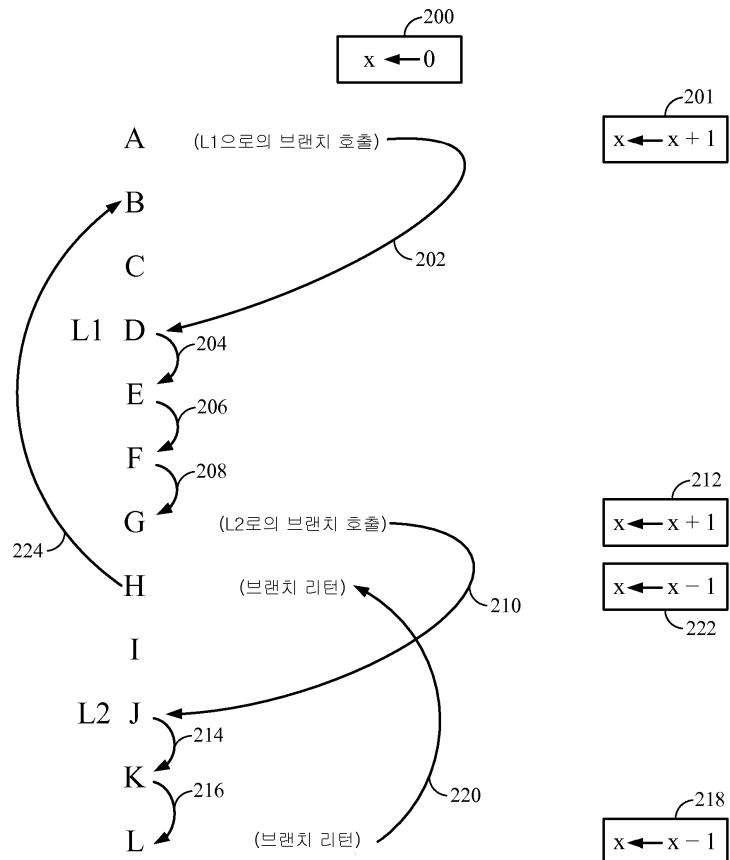
[0056] 앞선 개시물이 본 발명의 예시적 실시예들을 도시하지만, 첨부된 청구항들에 의해 정의된 바와 같은 본 발명의 범위로부터 벗어남 없이, 다양한 변경들 및 수정들이 본원에서 이루어질 수 있음이 주목되어야 한다. 본원에 설명된 본 발명의 실시예들에 따른 방법 청구항들의 기능들, 단계들 및/또는 액션들은 임의의 특정 순서로 수행될 필요가 없다. 또한, 본 발명의 엘리먼트들이 단수형으로 설명 또는 청구될 수 있지만, 단수형으로의 제한이 명시적으로 진술되지 않는 한, 복수형이 고려된다.

## 도면

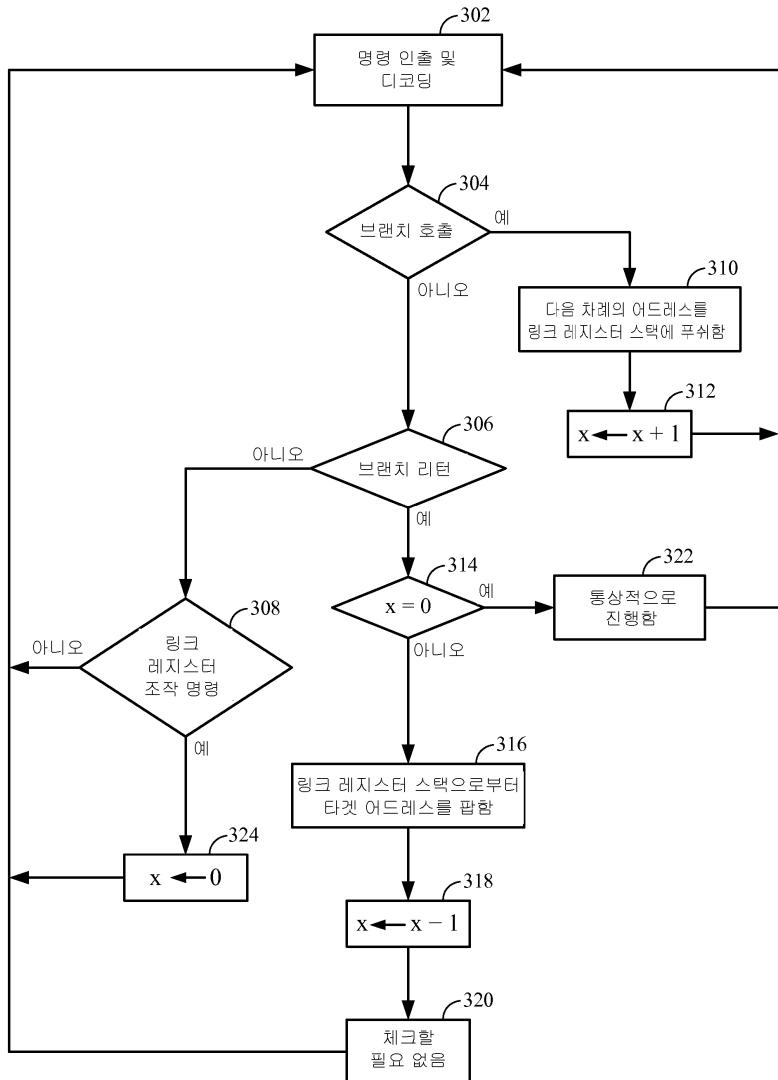
### 도면1



## 도면2



## 도면3



도면4

