



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0132242
(43) 공개일자 2015년11월25일

- (51) 국제특허분류(Int. Cl.)
G06F 9/30 (2006.01) *G06F 9/38* (2006.01)
- (52) CPC특허분류
G06F 9/30098 (2013.01)
G06F 9/30189 (2013.01)
- (21) 출원번호 10-2015-7027770
- (22) 출원일자(국제) 2014년03월14일
심사청구일자 없음
- (85) 번역문제출일자 2015년10월06일
- (86) 국제출원번호 PCT/US2014/029876
- (87) 국제공개번호 WO 2014/145160
국제공개일자 2014년09월18일
- (30) 우선권주장
13/841,576 2013년03월15일 미국(US)
- (71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
스트리트, 다웬, 유진
미국 92121 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
스템펠, 브라이언, 마이클
(뒷면에 계속)
- (74) 대리인
특허법인 남엔드남

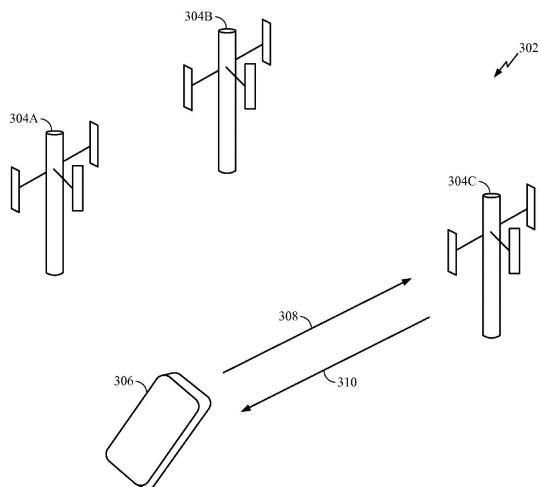
전체 청구항 수 : 총 35 항

(54) 발명의 명칭 컨텍스트-종속 명령들을 위한 성능의 최적화

(57) 요약

프로세서는 레지스터 필드의 현재 값의 컨텍스트 내에서 프로세싱된 명령들을 저장하기 위한 큐를 포함하고, 여기서 일부 실시예들에 대해, 명령은 프로세싱 시에 레지스터 필드에 종속하여 정의되지 않거나 정의된다. 기록 명령(레지스터 필드에 기록하는 명령)이 실행된 후에, 실행된 기록 명령에 종속하는 명령들을 포함하는 임의의 엔트리들에 대한 큐가 탐색된다. 각각의 그러한 엔트리는, 엔트리 내의 명령이 프로세싱된 시간에서의 레지스터 필드의 값을 저장한다. 그러한 엔트리가 큐에서 발견되고, 레지스터 필드의 그의 저장된 값이 기록 명령이 레지스터 필드에 기록한 값과 매칭하지 않는다면, 프로세서는 파이프라인을 플러싱하고, 명령을 정확하게 실행하기 위한 상태에서 재시작한다.

대 표 도



(52) CPC특허분류

G06F 9/3842 (2013.01)

G06F 9/3863 (2013.01)

(72) 발명자

스피어, 토마스, 필립

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

스미스, 로드니, 웨인

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

맥클레인, 마이클, 스코트

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

독서, 캐네스, 알렌

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

다이펜더퍼, 제임스, 노리스

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

명세서

청구범위

청구항 1

방법으로서,

레지스터 필드의 현재 값에 의해 결정된 컨텍스트 내에서 명령을 프로세싱하는 단계 – 상기 명령은 기록 명령에 종속하고, 상기 기록 명령은 구성 값을 상기 레지스터 필드에 기록하고, 상기 명령은 상기 기록 명령이 실행되기 전에 프로세싱됨 – ,

상기 명령을 큐(queue) 내의 엔트리에 저장하는 단계,

상기 레지스터 필드의 현재 값을 상기 엔트리에 저장하는 단계,

상기 기록 명령이 실행된 후에, 상기 엔트리에 대한 큐를 탐색하는 단계, 및

상기 엔트리에 저장된 현재 값과, 상기 기록 명령이 실행된 후의 구성 값을 비교하는 단계를 포함하는,

방법.

청구항 2

제 1 항에 있어서,

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 모든 동작중인(in-flight) 명령들의 파이프라인을 플러싱(flushing)하는 단계, 및 상기 기록 명령 후에 프로그램 순서에서 다음의 명령의 어드레스로 프로그램 카운터를 설정하는 단계를 더 포함하는,

방법.

청구항 3

제 2 항에 있어서,

상기 명령을 프로세싱하는 단계는,

상기 현재 값이 제 1 값이면, 정의된 바와 같이 상기 명령을 프로세싱하는 단계, 및

상기 현재 값이 상기 제 1 값이 아니면, 정의되지 않은 바와 같이 상기 명령을 프로세싱하는 단계를 더 포함하는,

방법.

청구항 4

제 1 항에 있어서,

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 상기 명령 및 프로그램 순서에서 상기 명령 다음의 모든 동작중인 명령들의 파이프라인을 플러싱하는 단계, 및 상기 명령의 어드레스로 프로그램 카운터를 설정하는 단계를 더 포함하는,

방법.

청구항 5

제 4 항에 있어서,

프로세서의 상태 정보를 상기 엔트리에 저장하는 단계 – 상기 상태 정보는 상기 명령이 프로세싱되는 시간에서의 상태 정보임 – , 및

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 상기 엔트리로부터 상기 상태 정보를 리트

리브(retrieve)하는 단계, 및 상기 프로세서를 상기 상태 정보에 의해 표시된 상태로 복구하는 단계를 더 포함하는,

방법.

청구항 6

제 1 항에 있어서,

상기 명령은 벡터 명령인,

방법.

청구항 7

제 1 항에 있어서,

상기 레지스터 필드는 하나 이상의 필드들을 포함하는,

방법.

청구항 8

제 1 항에 있어서,

상기 명령을 프로세싱하는 단계는,

상기 현재 값이 제 1 값이면, 정의된 바와 같이 상기 명령을 프로세싱하는 단계, 및

상기 현재 값이 상기 제 1 값이 아니면, 정의되지 않은 바와 같이 상기 명령을 프로세싱하는 단계를 더 포함하는,

방법.

청구항 9

제 1 항에 있어서,

상기 명령이 완료(commit)될 때, 상기 큐로부터 상기 엔트리를 제거하는 단계를 더 포함하는,

방법.

청구항 10

프로세서로서,

필드를 갖는 레지스터,

큐,

파이프라인,

제어기를 포함하고, 상기 제어기는,

상기 파이프라인이 기록 명령을 실행하기 전에, 상기 파이프라인으로 하여금 상기 레지스터 필드의 현재 값에 의해 결정된 컨택스트 내에서 명령을 프로세싱하게 하고 – 상기 명령은 상기 기록 명령에 종속하고, 상기 기록 명령은 구성 값을 상기 레지스터 필드에 기록함 – ,

상기 명령을 상기 큐 내의 엔트리에 저장하고,

상기 레지스터 필드의 현재 값을 상기 엔트리에 저장하고,

상기 파이프라인이 상기 기록 명령을 실행한 후에, 상기 엔트리에 대한 큐를 탐색하고, 그리고

상기 엔트리에 저장된 현재 값과, 상기 파이프라인이 상기 기록 명령을 실행한 후의 구성 값을 비교하도록 구성 되는,

프로세서.

청구항 11

제 10 항에 있어서,

상기 프로세서는 프로그램 카운터를 더 포함하고,

상기 제어기는 추가로, 상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면,

모든 동작중인 명령들의 파이프라인을 플러싱하고, 그리고

상기 기록 명령 후에 프로그램 순서에서 다음의 명령의 어드레스로 상기 프로그램 카운터를 설정하도록 구성되 는,

프로세서.

청구항 12

제 11 항에 있어서,

상기 제어기는 추가로 상기 파이프라인으로 하여금,

상기 현재 값이 제 1 값이면, 정의된 바와 같이 상기 명령을 실행하게 하고, 그리고

상기 현재 값이 상기 제 1 값이 아니면, 정의되지 않은 바와 같이 상기 명령을 실행하게 하도록 구성되는,

프로세서.

청구항 13

제 10 항에 있어서,

상기 프로세서는 프로그램 카운터를 더 포함하고,

상기 제어기는, 상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면,

상기 명령 및 프로그램 순서에서 상기 명령 다음의 모든 동작중인 명령들의 파이프라인을 플러싱하고, 그리고

상기 명령의 어드레스로 상기 프로그램 카운터를 설정하도록 구성되는,

프로세서.

청구항 14

제 13 항에 있어서,

상기 제어기는,

프로세서의 상태 정보를 상기 엔트리에 저장하고 — 상기 상태 정보는 상기 파이프라인이 명령을 프로세싱하는 시간에서의 상태 정보임 — ,

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 상기 엔트리로부터 상기 상태 정보를 리트 리브하고, 상기 프로세서를 상기 상태 정보에 의해 표시된 상태로 복구하도록 구성되는,

프로세서.

청구항 15

제 13 항에 있어서,

상기 제어기는 상기 파이프라인으로 하여금,

상기 현재 값이 제 1 값이면, 정의된 바와 같이 상기 명령을 실행하게 하고, 그리고

상기 현재 값이 상기 제 1 값이 아니면, 정의되지 않은 바와 같이 상기 명령을 실행하게 하도록 구성되는,

프로세서.

청구항 16

제 10 항에 있어서,

상기 제어기는 상기 파이프라인으로 하여금,

상기 현재 값이 제 1 값이면, 정의된 바와 같이 상기 명령을 실행하게 하고, 그리고

상기 현재 값이 상기 제 1 값이 아니면, 정의되지 않은 바와 같이 상기 명령을 실행하게 하도록 구성되는,

프로세서.

청구항 17

제 10 항에 있어서,

상기 제어기는, 상기 파이프라인이 상기 명령을 완료할 때, 상기 큐로부터 상기 엔트리를 제거하도록 구성되는,

프로세서.

청구항 18

제 10 항에 있어서,

상기 명령은 벡터 명령인,

프로세서.

청구항 19

제 10 항에 있어서,

상기 레지스터 필드는 하나 이상의 필드들을 포함하는,

프로세서.

청구항 20

장치로서,

레지스터 필드의 현재 값에 의해 결정된 컨텍스트 내에서 명령을 프로세싱하기 위한 수단 – 상기 명령은 기록 명령에 종속하고, 상기 기록 명령은 구성 값을 상기 레지스터 필드에 기록하고, 상기 명령은 상기 기록 명령이 실행되기 전에 프로세싱됨 – ,

상기 명령을 큐 내의 엔트리에 저장하기 위한 수단,

상기 레지스터 필드의 현재 값을 상기 엔트리에 저장하기 위한 수단,

상기 기록 명령이 실행된 후에, 상기 엔트리에 대한 큐를 탐색하기 위한 수단, 및

상기 엔트리에 저장된 현재 값과, 상기 기록 명령이 실행된 후의 구성 값을 비교하기 위한 수단을 포함하는, 장치.

청구항 21

제 20 항에 있어서,

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 모든 동작중인 명령들의 파이프라인을 플러싱하기 위한 수단, 및 상기 기록 명령 후에 프로그램 순서에서 다음의 명령의 어드레스로 프로그램 카운터를 설정하기 위한 수단을 더 포함하는,

장치.

청구항 22

제 20 항에 있어서,

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 상기 명령 및 프로그램 순서에서 상기 명령 다음의 모든 동작중인 명령들의 파이프라인을 플러싱하기 위한 수단, 및 상기 명령의 어드레스로 상기 프로그램 카운터를 설정하기 위한 수단을 더 포함하는,
장치.

청구항 23

제 22 항에 있어서,

프로세서의 상태 정보를 상기 엔트리에 저장하기 위한 수단 – 상기 상태 정보는 상기 명령이 프로세싱되는 시
간에서의 상태 정보임 – , 및

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 상기 엔트리로부터 상기 상태 정보를 리트
리브하기 위한 수단, 및 상기 프로세서를 상기 상태 정보에 의해 표시된 상태로 복구하기 위한 수단을 더 포함
하는,
장치.

청구항 24

제 20 항에 있어서,

상기 명령을 프로세싱하기 위한 수단은,

상기 현재 값이 제 1 값이면, 정의된 바와 같이 상기 명령을 프로세싱하기 위한 수단, 및

상기 현재 값이 상기 제 1 값이 아니면, 정의되지 않은 바와 같이 상기 명령을 프로세싱하기 위한 수단을 더 포함
하는,

장치.

청구항 25

제 20 항에 있어서,

상기 명령이 완료될 때, 상기 큐로부터 상기 엔트리를 제거하기 위한 수단을 더 포함하는,

장치.

청구항 26

제 20 항에 있어서,

상기 레지스터 필드는 하나 이상의 필드들을 포함하는,

장치.

청구항 27

제 20 항에 있어서,

상기 명령은 벡터 명령인,

장치.

청구항 28

명령어들이 저장된 비일시적인 컴퓨터-판독 가능 매체로서,

상기 명령어들은 제어기로 하여금 방법을 수행하게 하고, 상기 방법은,

레지스터 필드의 현재 값에 의해 결정된 컨텍스트 내에서 명령을 프로세싱하는 단계 — 상기 명령은 기록 명령에 종속하고, 상기 기록 명령은 구성 값을 상기 레지스터 필드에 기록하고, 상기 명령은 상기 기록 명령이 실행되기 전에 프로세싱됨 — ,

상기 명령을 큐 내의 엔트리에 저장하는 단계,

상기 레지스터 필드의 현재 값을 상기 엔트리에 저장하는 단계,

상기 기록 명령이 실행된 후에, 상기 엔트리에 대한 큐를 탐색하는 단계, 및

상기 엔트리에 저장된 현재 값과, 상기 기록 명령이 실행된 후의 구성 값을 비교하는 단계를 포함하는,

비일시적인 컴퓨터-판독 가능 매체.

청구항 29

제 28 항에 있어서,

상기 방법은,

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 모든 동작중인 명령들의 파이프라인을 플러싱하는 단계, 및 상기 기록 명령 후에 프로그램 순서에서 다음의 명령의 어드레스로 프로그램 카운터를 설정하는 단계를 더 포함하는,

비일시적인 컴퓨터-판독 가능 매체.

청구항 30

제 28 항에 있어서,

상기 방법은,

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 상기 명령 및 프로그램 순서에서 상기 명령 다음의 모든 동작중인 명령들의 파이프라인을 플러싱하는 단계, 및 상기 명령의 어드레스로 프로그램 카운터를 설정하는 단계를 더 포함하는,

비일시적인 컴퓨터-판독 가능 매체.

청구항 31

제 30 항에 있어서,

상기 방법은,

프로세서의 상태 정보를 상기 엔트리에 저장하는 단계 — 상기 상태 정보는 상기 명령이 프로세싱되는 시간에서의 상태 정보임 — , 및

상기 엔트리에 저장된 현재 값이 상기 구성 값과 매칭하지 않는다면, 상기 엔트리로부터 상기 상태 정보를 리트리브하는 단계, 및 상기 프로세서를 상기 상태 정보에 의해 표시된 상태로 복구하는 단계를 더 포함하는,

비일시적인 컴퓨터-판독 가능 매체.

청구항 32

제 28 항에 있어서,

상기 명령을 프로세싱하는 단계는,

상기 현재 값이 제 1 값이면, 정의된 바와 같이 상기 명령을 프로세싱하는 단계, 및

상기 현재 값이 상기 제 1 값이 아니면, 정의되지 않은 바와 같이 상기 명령을 프로세싱하는 단계를 더 포함하는,

비일시적인 컴퓨터-판독 가능 매체.

청구항 33

제 28 항에 있어서,
상기 방법은,
상기 명령이 완료될 때, 상기 큐로부터 상기 엔트리를 제거하는 단계를 더 포함하는,
비일시적인 컴퓨터-판독 가능 매체.

청구항 34

제 28 항에 있어서,
상기 레지스터 필드는 하나 이상의 필드들을 포함하는,
비일시적인 컴퓨터-판독 가능 매체.

청구항 35

제 28 항에 있어서,
상기 명령은 벡터 명령인,
비일시적인 컴퓨터-판독 가능 매체.

발명의 설명**기술 분야**

[0001] 본 발명은 마이크로프로세서들에 관한 것이며, 더 상세하게는, 작동이 컨텍스트-종속 명령들을 프로세싱 할 때, 마이크로프로세서들의 성능을 최적화하는 것에 관한 것이다.

배경 기술

[0002] 일부 현대 마이크로프로세서 아키텍처들에 대한 명령 세트들은, 작동이 컨텍스트-종속 명령들(예를 들면, 부동 소수점(floating point) 벡터 명령들)을 포함한다. 벡터 명령(또는 동작)은 결과를 제공하기 위해 하나 이상의 데이터 벡터들에 대해 동작하고, 여기서 결과는 또한 벡터일 수 있다. 벡터 명령은 또한 SIMD(Single Instruction Multiple Data) 동작이라 칭해질 수 있다.

[0003] 파일라이닝된 그리고 비순차적인(out-of-order) 실행 아키텍처들을 갖는 마이크로프로세서들이 고성능으로 컨텍스트-종속 명령들을 구현하는 것이 바람직하다.

발명의 내용

[0004] 본 발명의 실시예들은 벡터 부동 소수점 명령들에 대한 성능을 최적화하고, 더 일반적으로, 하나 이상의 레지스터 필드들로의 기록 명령에 종속하는 명령들의 성능을 최적화하기 위한 시스템들 및 방법들에 관한 것이며, 여기서 하나 이상의 레지스터 필드들은 명령에 대한 컨텍스트를 결정한다.

[0005] 실시예에서, 방법은 레지스터 필드의 현재 값에 의해 결정된 컨텍스트 내에서 명령을 프로세싱하는 단계 — 명령은 기록 명령에 종속하고, 기록 명령은 구성 값을 레지스터 필드에 기록하고, 명령은 기록 명령이 실행되기 전에 프로세싱됨 —, 명령을 큐(queue) 내의 엔트리에 저장하는 단계, 레지스터 필드의 현재 값을 엔트리에 저장하는 단계, 기록 명령이 실행된 후에, 엔트리에 대한 큐를 탐색하는 단계, 및 엔트리에 저장된 현재 값과, 기록 명령이 실행된 후의 구성 값을 비교하는 단계를 포함한다.

[0006] 다른 실시예에서, 프로세서는 필드를 갖는 레지스터, 큐, 파일라인, 및 제어기를 포함한다. 제어기는 파일라인으로 하여금 레지스터 필드의 현재 값을 통해 결정된 컨텍스트 내에서 명령을 프로세싱하게 하고 — 벡터 명령은 기록 명령에 종속하고, 기록 명령은 구성 값을 레지스터 필드에 기록함 —, 명령을 큐 내의 엔트리에 저장하고, 레지스터 필드의 현재 값을 엔트리에 저장하고, 파일라인이 기록 명령을 실행한 후에, 엔트리에 대한 큐를 탐색하고, 그리고 엔트리에 저장된 현재 값과, 파일라인이 기록 명령을 실행한 후의 구성 값을 비교하도록 구성된다.

[0007]

[0007] 다른 실시예에서, 장치는 레지스터 필드의 현재 값에 의해 결정된 컨텍스트 내에서 명령을 프로세싱하기 위한 수단 – 명령은 기록 명령에 종속하고, 기록 명령은 구성 값을 레지스터 필드에 기록하고, 명령은 기록 명령이 실행되기 전에 프로세싱됨 – , 명령을 큐 내의 엔트리에 저장하기 위한 수단, 레지스터 필드의 현재 값을 엔트리에 저장하기 위한 수단, 기록 명령이 실행된 후에, 엔트리에 대한 큐를 탐색하기 위한 수단, 및 엔트리에 저장된 현재 값과, 기록 명령이 실행된 후의 구성 값을 비교하기 위한 수단을 포함한다.

[0008]

[0008] 다른 실시예에서, 비일시적인 컴퓨터-판독 가능 매체는 명령어들을 저장하고, 명령어들은, 명령어들을 실행할 때, 제어기로 하여금 방법을 수행하게 하고, 상기 방법은, 레지스터 필드의 현재 값에 의해 결정된 컨텍스트 내에서 명령을 프로세싱하는 단계 – 명령은 기록 명령에 종속하고, 기록 명령은 구성 값을 레지스터 필드에 기록하고, 명령은 기록 명령이 실행되기 전에 프로세싱됨 – , 명령을 큐 내의 엔트리에 저장하는 단계, 레지스터 필드의 현재 값을 엔트리에 저장하는 단계, 기록 명령이 실행된 후에, 엔트리에 대한 큐를 탐색하는 단계, 및 엔트리에 저장된 현재 값과, 기록 명령이 실행된 후의 구성 값을 비교하는 단계를 포함한다.

도면의 간단한 설명

[0009]

[0009] 첨부한 도면들은, 본 발명의 실시예들의 설명을 보조하기 위해 제시되며, 실시예들의 제한이 아니라 오로지 실시예들의 예시를 위해서만 제공된다.

[0010] 도 1은 실시예에 따른 프로세서의 추상적 개념이다.

[0011] 도 2a는 실시예에 따른, 컨텍스트를 체크하고 보수하는 방법을 예시한다.

[0012] 도 2b는 실시예에 따른, 컨텍스트를 체크하고 보수하는 방법을 예시한다.

[0013] 도 3은 실시예들이 애플리케이션을 발견할 수 있는 통신 시스템을 예시한다.

발명을 실시하기 위한 구체적인 내용

[0010]

[0014] 본 발명의 양상들이, 본 발명의 특정 실시예들에 관련하여 이하의 설명 및 관련 도면들에 개시된다. 대안적인 실시예들이 본 발명의 범위로부터 벗어나지 않고 창안될 수 있다. 추가적으로, 본 발명의 잘-알려진 엘리먼트들은 본 발명의 관련 세부사항들을 모호하게 하지 않기 위해 상세하게 설명되지 않거나 또는 생략될 것이다.

[0011]

[0015] 단어 "예시적인"은 본원에서 "예, 예시, 또는 예증으로서 기능하는 것"을 의미하도록 사용된다. "예시적인"으로서 본원에 설명된 임의의 실시예는 반드시 다른 실시예들보다 바람직하거나 유리한 것으로서 해석되지는 않는다. 유사하게, 용어 "본 발명의 실시예들"은, 본 발명의 모든 실시예들이 논의된 특징, 이점 또는 동작의 모드를 포함하도록 요구하지 않는다.

[0012]

[0016] 본원에 이용된 용어는, 오직 특정한 실시예들을 설명하기 위한 목적으로, 본 발명의 실시예들을 제한하도록 의도되지는 않는다. 본원에 이용된 바와 같이, 문맥이 그렇지 않은 것으로 명확하게 지시하지 않는 한, 단수 형태들은 복수 형태들도 또한 포함하는 것으로 의도된다. 용어들 "포함하다(comprises)", "포함하는 (comprising)", "포함하다(includes)" 및/또는 "포함하는(including)"은, 본원에 이용되는 경우, 언급된 특징들, 정수들, 단계들, 동작들, 엘리먼트들, 및/또는 컴포넌트들의 존재를 특정하지만, 하나 이상의 다른 특징들, 정수들, 단계들, 동작들, 엘리먼트들, 컴포넌트들, 및/또는 이들의 그룹들의 존재 또는 부가를 배제하는 것은 아니라는 것을 더 이해할 것이다.

[0013]

[0017] 게다가, 수많은 실시예들이, 예를 들어, 컴퓨팅 디바이스의 엘리먼트들에 의해 수행될 동작들의 시퀀스들과 관련하여 설명된다. 특정 회로들(예컨대, 주문형 집적회로들(ASIC들)), 프로그램 명령들을 실행하는 하나 이상의 프로세서들, 또는 이들 둘의 조합이 본원에 설명된 다양한 동작들을 수행할 수 있다는 점이 인식될 것이다. 추가적으로, 본원에 설명된 동작들의 이러한 시퀀스는, 실행 시에, 관련 프로세서로 하여금 본원에 설명된 기능을 수행하게 하는 대응 세트의 컴퓨터 명령들이 저장된 임의의 형태의 컴퓨터 판독가능 저장 매체 내에서 전체적으로 구현되는 것으로 고려될 수 있다. 따라서, 본 발명의 다양한 양상들은 다수의 상이한 형태들로 구현될 수 있고, 이 형태들 모두는 청구된 청구대상의 범위 내에 있는 것으로 고려된다. 또한, 본원에 설명된 실시예들 각각에 대해, 임의의 이러한 실시예들의 대응하는 형태는, 예를 들어, 설명된 동작을 수행하도록 "구성된 로직"으로서 본원에 설명될 수 있다.

[0014]

[0018] 실시예를 설명하는 것을 용이하게 하기 위해, 도 1에 예시된 프로세서(100)의 추상적 개념에 대해 먼저

참조가 이루어지고, 여기서 간략함을 위해 프로세서에서 종래에 발견되는 모든 기능적 유닛들이 도시되지는 않는다. 실시예들이 애플리케이션들을 발견하는 프로세서(100)는 슈퍼스칼라, 비순차적인 아키텍처를 가질 수 있지만, 이것이 요구은 아니다. 그러나, 도 1에 예시된 프로세서(100)가 명령들을 추측으로(speculatively) 실행 또는 프로세싱하고, 추측 실행 또는 프로세싱이 부정확한 것으로 알려지면 적절히 복구하기 위한 능력을 갖는다고 가정된다. 예를 들면, 많은 프로세서들은 브랜치 명령의 타겟 어드레스를 예측함으로써 브랜치 명령을 추측으로 실행하고, 예측된 타겟 어드레스가 나중에 부정확한 것으로 알려지면 정확한 명령 흐름을 적절히 복구 및 실행하기 위한 능력을 갖는다.

[0015] [0019] 명령 프리페치 스테이지(104), 명령 디코드 스테이지(106), 레지스터 재명명(rename) 스테이지(108) 및 명령 큐 및 디스패치 스테이지(110)를 포함하는 파이프라인(102)이 도 1에 도시된다. 실제로, 예를 들면, 명령이 완료(commit)(은퇴)할 때, 결과들을 메모리 위치들 또는 아키텍처 레지스터들에 기록하기 위한 다양한 스테이지들과 같은 더 많은 스테이지들이 파이프라인에 존재할 수 있지만, 간략함으로 위해, 파이프라인(102)에서 단지 4 개의 스테이지들이 예시된다. 제어기(112)는 파이프라인(102)에서 동작들을 제어한다. (여기서, 용어들 완료 및 은퇴는 동의어로서 사용되고, 말하자면, 명령 완료는 또한 명령 은퇴를 의미하도록 취해질 수 있음.)

[0016] [0020] 메모리 계층의 부분으로서 고려될 수 있는 다양한 캐시들 및 오프-칩 메모리를 포함하는 메모리 저장소로부터 명령들 및 데이터가 리트리브(retrieve)되고, 이에 기록된다. (114)로 라벨링된 기능적 블록은 이러한 메모리 계층을 추상화하고, 논의의 편의를 위해, 하나 이상의 메모리 기능적 유닛들은 간단히 메모리 계층(114)으로 지칭될 수 있다. 메모리 계층(114)을 구성하는 메모리의 일부는 제어기(112) 또는 다른 기능적 유닛들에 통합될 수 있고, 비일시적인 컴퓨터-판독 가능 저장 매체들로 지칭될 수 있다. 일부 실시예들에서, 메모리 계층(114)은, 제어기(112) 또는 다른 기능적 유닛들이 본원에 설명된 다양한 동작들을 수행할 수 있는 저장된 명령들을 포함할 수 있다.

[0017] [0021] 도 1에 예시된 다른 기능적 유닛들은 물리적 레지스터 파일(116), 한 세트의 실행 유닛들(118) 및 재정렬 버퍼(120)이다. 재정렬 버퍼(120)는 비순차적인 실행을 용이하게 하고, 일부 실시예들에서, 순환 버퍼로서 구현될 수 있고, 여기서 명령들이 명령 큐 및 디스패치 스테이지(110)에 입력될 때, 명령들이 버퍼에 입력되고, 명령들이 완료(은퇴)할 때, 재정렬 버퍼(120)를 탈출한다. 명령들은, 명령들이 재정렬 버퍼(120)에 입력된 순서로 완료한다.

[0022] SIIQ(speculative instruction information queue)(122)로 라벨링된 다른 버퍼는, 추측으로 프로세싱된 동작중인(in-flight) 명령이 정확한 명령 흐름의 부분이 아닌 것으로 나중에 발견되고, 따라서 완료되지 않아야 하는 경우에, 프로세서(100)의 상태를 복원하는데 필요할 수 있는 다양한 레지스터 값들 및 아마도 상태 정보와 함께, 추측으로 예측된 컨텍스트 내에서 파이프라인(102)에서 조기에 프로세싱된 명령들을 저장하는데 사용된다. 이러한 버퍼는 추측 명령 정보 큐로 지칭되고, 나중에 더 상세히 논의된다.

[0023] 벡터 프로세싱은 음성 또는 이미지 프로세싱과 같은 많은 신호 프로세싱 애플리케이션들에서 중요한 능력이다. 벡터 프로세싱 명령은 때때로 SIMD(Single Instruction Multiple Data) 명령으로 지칭되고, 여기서 단일 명령은 다중 데이터에 대해 동작한다. 벡터는 연속적인 엘리먼트들 사이의 일정한 스트라이드(stride)를 갖는 다양한 메모리 위치들로 고려될 수 있다. 다른 관점에서, 벡터는 연속적인 엘리먼트들 사이의 일정한 스트라이드를 갖는 다양한 레지스터 위치들로 고려될 수 있다. 예를 들면, 물리적 레지스터들의 어레이 $\{R_i, i = 0, 1, 2, \dots, 15\}$ 에 대해, 벡터는 레지스터 위치들 $R\{(i), i = 0, 2, 4, \dots, 14\}$ 로 구성될 수 있고, 여기서 스트라이드는 2이다.

[0024] 로드-저장 아키텍처 프로세서에서, 벡터의 엘리먼트들은, 벡터 명령에 따라 동작되기 전에, 로드 명령으로 레지스터들로 로딩된다. 예를 들면, 벡터 합산 명령은 제 1 세트의 소스 레지스터들 내의 값들을 제 2 세트의 소스 레지스터들 내의 값들에 합산하고, 결과들을 한 세트의 목적지 레지스터들에 기록할 수 있다. 특정 예로서, 제 1 세트의 소스 레지스터들 $\{RX(i) | i = 0, 1, 2, 3\}$ 내의 값들은 제 2 세트의 소스 레지스터들 $\{RY(i) | i = 0, 1, 2, 3\}$ 내의 값들에 합산될 수 있고, 결과들은 한 세트의 목적지 레지스터들 $\{RZ(i) | i = 0, 1, 2, 3\}$ 에 기록되고, 여기서 각각의 $i = 0, 1, 2, 3$ 에 대해, 레지스터 $RZ(i)$ 의 값은 레지스터들 $RX(i)$ 및 $RY(i)$ 내의 값들의 합산이다. 이러한 특정 예에서, 벡터의 길이는 4이고, 스트라이드는 1이다.

[0025] 벡터 프로세싱에 사용된 소스 레지스터 및 목적지 레지스터 파일들은 기능적 유닛(116)에 의해 포함되는 물리적 레지스터들로서 또는 도 1에 명시적으로 도시되지 않은 다른 물리적 레지스터 파일로서 보여질 수 있다.

- [0022] [0026] 벡터 명령들에 의해 소비되는 벡터 길이, 스트라이드 및 다양한 다른 파라미터들은 하나 이상의 제어 레지스터들에 저장된다. 그러한 제어 레지스터의 예는 도 1에서 124로 라벨링되고, FPSCR(Floating Point Status and Control Register)(124)로 지칭된다. 파라미터들을 FPSCR(124)에 기록(복제)하기 위한 명령은, 필요한 파라미터들에 의존하는 벡터 명령들 전에(프로그램 순서로) 코드의 블록에 삽입된다. 이어서, FPSCR(124)로의 기록 명령에 대한 이러한 벡터 명령들의 종속성이 존재한다.
- [0023] [0027] 특히, 벡터 가능 프로세서에 대한 일부 명령 세트들은, 이러한 필드들 내의 값들에 기초하여 벡터 명령들의 작동을 구성하는 FPSCR(124) 내의 하나 이상의 필드들에 기록하는 기록 명령을 포함한다. 스트라이드 또는 길이 필드들은 FPSCR(124)에 포함될 수 있는 예시적인 필드들이지만, 다른 필드들이 또한 포함될 수 있다. 이들 하나 이상의 필드들은 도 1에서 125로 라벨링된다. 논의의 편의를 위해 그리고 일반화의 손실 없이, 우리는 필드로서 125로 지칭하지만, 이것은 1보다 더 많은 비트, 및 FPSCR(124)의 하나보다 더 많은 필드를 스파닝 할 수 있다.
- [0024] [0028] 예를 들면, 필드(125)에 기록된 비-제로 값은, 벡터 명령 실행이 정의되지 않은 명령인 컨텍스트를 정의 할 수 있어서, 벡터 명령 실행이 시도되는 경우에, 프로세서(100)가 인터럽트 핸들러(interrupt handler)를 실행하게 한다. 일부 실시예들에서, 필드(125)가 비-제로 값으로 설정되면, 벡터 명령들의 세트의 적절한 서브세트만이 정의되지 않게 된다. 즉, 일부 실시예들에서, FPSCR(124) 내의 필드(125)를 무시하는 일부 벡터 명령들이 존재할 수 있다.
- [0025] [0029] 논의의 편의를 위해, 기록 명령은, FPSCR(124) 내의 필드(125)에 기록하는 명령을 지칭할 것이고, 필드(125)는 구성 필드 또는 구성 값으로 지칭된다.
- [0026] [0030] 기록 명령 및 벡터 명령과 같은 종속 명령 둘 모두가 파이프라인(102) 내에 있을 때, 기록 명령이 구성 필드(125)에 기록할 수 있기 전에, 벡터 명령에 대한 피연산자들(operands)이 이용 가능한 것이 발생할 수 있다. 예를 들면, 기록 명령이 구성 필드(125)에 기록하는 것인 값은, 종속 벡터 명령이 실행될 준비가 될 때, 여전히 실행이 완료되지 않은 다른 명령의 결과일 수 있다. 종속성 위험을 회피하기 위해, 일부 종래 기술의 프로세서들은 컨텍스트 동기화를 구현할 수 있고, 이로써 기록 명령에(프로그램 순서에서) 후속하는 모든 벡터 명령들은, 기록 명령이 완료될 때까지 실행되지 않는다. 일부 다른 종래 기술의 프로세서들은, 벡터 명령을 디코딩할 때, (프로그램 순서에서) 모든 이전의 기록 명령들이 완료될 때까지 홀드(hold)를 도입할 수 있다. 그러나, 그러한 방법들은 성능을 저하시키고, 파이프라인 버블들을 발생시킬 수 있다.
- [0027] [0031] 성능을 개선하기 위해, 실시예들은 종속 명령들의 추측 프로세싱 및 실행을 수행한다. 이러한 실시예에서, 종속 명령들은 벡터 명령들이지만, 구성 가능한 특권(privilege) 또는 보안 종속성을 갖는 명령들 또는 특정 아키텍처 레지스터들을 액세스하는 명령들을 비롯하여(이에 제한되지 않음), 다른 타입들의 종속 명령들이 본 개시의 범위 내에 있다는 것을 당업자들은 이해할 것이다. 더 상세하게는, 실시예들은 FPSCR(124)의 현재 값들에 의해 결정된 컨텍스트 내에서 벡터 명령을 프로세싱하기 시작한다. 기록 명령이 아직 완료되지 않고 동작중이지만, (프로그램 순서에서) 나중의 종속 벡터 명령이 프로세싱을 시작하기 위해 이용 가능하면, 그 종속 벡터 명령은 그의 프로세싱 시간에서 FPSCR(124)의 현재 상태에 종속하여 프로세싱된다.
- [0028] [0032] 기록 명령에 종속한 벡터 명령이 프로세싱될 때, 벡터 명령의 프로세싱 시에 FPSCR(124)의 현재 컨텍스트(즉, 구성 필드(125)의 값)가 정확한 컨텍스트가 아닌 것이 발생할 수 있다. 예를 들면, 현재 컨텍스트는 벡터 명령이 정의된 바와 같이 프로세싱되지 않은 것일 수 있지만, 현재 컨텍스트가 정확한 경우에, 벡터 명령은 사실상 정의된 바와 같이 프로세싱되어야 한다. 대안적으로, 벡터 명령의 프로세싱 시에 FPSCR(124)의 현재 컨텍스트는 벡터 명령이 정의되는 것이 발생할 수 있고, 정확한 컨텍스트에서 프로세싱될 때, 벡터 명령은 아직 정의되지 않아야 한다.
- [0029] [0033] 따라서, 벡터 명령이 부정확한 컨텍스트(즉, 본원에서 파이프라인(102)이 벡터 명령을 프로세싱하기 시작할 때, 구성 필드(125)의 값을 나타내는 컨텍스트)에서 프로세싱된다고 결정되면, 실시예들은, 결국 벡터 명령이 정확한 컨텍스트에서 프로세싱 및 실행될 것이도록 복구 절차를 수행해야 한다.
- [0030] [0034] 복구 프로세스를 용이하게 하기 위해, 실시예들은 추측 명령 정보 큐(122)를 사용한다. 이러한 큐는 브랜치 명령 큐(BIQ)와 동일한 구조를 가질 수 있고, 일부 실시예들에서, 추측 명령 정보 큐(122)는 BIQ 자체일 수 있지만, 벡터 명령들이 또한 브랜치 명령들에 부가하여 큐잉된다. 일부 실시예들에서, 명령 Vopp(127)의 어드레스 및 원하는 페치 어드레스에서 실행을 재개하는데 요구되는 임의의 다른 비-컨텍스트-종속 정보를 비롯하

여(이에 제한되지 않음), 상태 정보는 또한 복구 프로세스 동안에 프로세서(100)를 정확한 상태에 놓는 것을 돋도록 포함될 수 있다. 실시예들이 BIQ로서 기능적 유닛(122)을 참조하여 설명될 수 있지만, 구조를 BIQ로 제한하지 않기 위해 상이한 텀(term)을 사용하는 것이 유용하고, 따라서 큐는 추측 명령 정보 큐로서 지칭된다.

[0031] [0035] 도 1에 예시된 예시적인 실시예에서, 재정렬 버퍼(120)에서 126으로 라벨링된 명령 VMSR은 값을 FPSCR(124) 내의 구성 필드(125)에 기록하기 위한 명령을 표기한다. (VMSR 용어는 ARM 명령 세트로부터 취해지고, 이것은 콘텐츠를 시스템 레지스터로부터 FPSCR로 전송하기 위한 명령을 나타낼 수 있다. 이러한 용어는 단지 편의상 차용되고, 실시예들은 많은 상이한 아키텍처들 및 명령 세트들에서 애플리케이션들을 발견할 수 있다.) 재정렬 버퍼(120) 내의 127로 라벨링된 후속 명령 Vopp는 명령 VMSR(126)에 종속하는 벡터 명령을 표기한다. 128로 라벨링된 화살표는 이러한 종속성을 나타낸다.

[0032] [0036] 파일라인(102)이 명령 Vopp(127)를 프로세싱하기 시작할 때, 이것은 추측 명령 정보 큐(122)에 배치된다. 130으로 라벨링된 화살표는 132로 라벨링된 엔트리로서 도시된 추측 명령 정보 큐(122)로의 배치를 표기한다. 엔트리(132)는 2 개의 부분들을 포함하고, 132a로 라벨링된 부분은 벡터 명령 및 컨텍스트 필드(133)를 포함하고, 132b로 라벨링된 부분은 상태 정보를 포함한다. 일부 실시예들은 상태 정보를 포함하지 않을 수 있다. 예를 들면, 엔트리(132)와 같은 엔트리에 상태 정보를 저장하는 일부 실시예들에서, 상태 정보는, 벡터 명령이 부정확한 컨텍스트에서 프로세싱되었다고 결정될 때 복구 프로세스 동안에 프로세서(100)를 정확한 상태로 복귀하는데 사용된다. 엔트리(132) 내의 컨텍스트 필드(133)는, 엔트리(132)에 대한 연관된 명령 Vopp(127)가 프로세싱된 컨텍스트를 나타낸다. 엔트리와 연관된 벡터 명령이 완료되거나 플러싱될 때, 그 엔트리는 추측 명령 정보 큐(122)로부터 제거된다.

[0033] [0037] 기록 명령("기록"은 FPSCR(124)로의 기록을 지칭함)이 동작중인 일부 시점에서, 임의의 종속 벡터 명령들이 부정확한 컨텍스트에서 프로세싱되었다는지에 대한 결정이 이루어진다. 예를 들면, 명령 VMSR(126)과 명령 Vopp(127) 사이에 어떠한 개재 기록 명령들도 존재하지 않는다면, 명령 Vopp(127)는 명령 VMSR(126)에 종속할 수 있고, 명령 Vopp(127)가 은퇴하기 전에, 명령 Vopp(127)이 정확한 컨텍스트에서 프로세싱되는 것을 보장하기 위해 주의해야 된다.

[0034] [0038] 일부 실시예들에서, 기록 명령이 완료될 때, 정확한 컨텍스트에 대한 결정이 이루어진다. 도 1에 예시된 예에 대해 계속하면, 명령 VMSR(126)이 기록 버퍼(120)의 상부(또는 헤드)에 있고, 완료할 준비될 때, 명령 VMSR(126)이 구성 필드(125)에 기록하는 값은 종속적인 동작중인 벡터 명령을 갖는 추측 명령 정보 큐(122) 내의 각각의 엔트리에 대한 컨텍스트 필드와 비교된다. 예를 들면, 컨텍스트 필드(133)의 값은, 엔트리(132) 내의 명령 Vopp(127)가 정확한 컨텍스트에서 프로세싱되었다는지를 결정하기 위해 명령 VMSR(126)로부터 생긴 구성 필드(125)에 대한 값과 비교된다. 추측 명령 정보 큐(122) 내의 벡터 명령은, 어떠한 개재 기록 명령들도 존재하지 않는다면, 명령 VMSR(126)에 종속한다.

[0035] [0039] 일부 실시예들에서, 명령 VMSR(126)이 완료할 준비가 되지만 추측 명령 정보 큐(122) 내의 종속 벡터 명령이 부정확한 컨텍스트에서 프로세싱되었다고 발견되는 경우에, 명령 VMSR(126)이 은퇴되고(이것은, 명령 VMSR(126) 앞의 모든 명령들이 또한 은퇴하였다는 것을 암시함), 모든 명령들이 파일라인 및 재정렬 버퍼(120)로부터 플러싱되고, 프로그램 카운터(134)가 명령 VMSR(126)에 대해 프로그램 순서에서 바로 후속하는 명령의 어드레스로로딩된다. 이러한 경우에, 상태 정보는 추측 명령 정보 큐(122)에 저장될 필요가 없는데, 왜냐하면 명령 VMSR(126)이 은퇴하고 프로그램 제어가 다음의 순차적인 명령으로 시작할 때, 아키텍처 레지스터들 모두가 정확한 상태에 있기 때문이다.

[0036] [0040] 일부 실시예들은, 부정확한 컨텍스트에서 프로세싱될 수 있는 종속 벡터 명령들에 대한 추측 명령 정보 큐(122)를 탐색하기 전에, 명령 VMSR(126)이 재정렬 버퍼(120)의 상부에 도달하기를 대기할 수 없다. 예를 들면, 일부 실시예들에서, 명령 VMSR(126)이 실행될 때(여기서, 명령 VMSR(126)의 실행은 자신이 FPSCR(124)에 기록한 값이 알려져 있다는 것을 의미함), 명령 VMSR(126)이 완료될 준비가 되지 않을지라도, 부정확하게 프로세싱된 종속 벡터 명령들에 대한 추측 명령 정보 큐(122)가 탐색된다.

[0037] [0041] 다른 실시예에서, 명령 Vopp(127)와 같은 SIIQ(122) 내의 명령이 은퇴할 준비가 될 때(예를 들면, 완료하거나 정의되지 않은 인터럽트를 취할 준비가 됨), 명령 Vopp(127)는 구성 필드(125) 내의 아키텍처 컨텍스트에 대해 컨텍스트 필드(133)를 비교한다. 컨텍스트 필드(133) 및 구성 필드(125)가 매칭하면, 명령 Vopp(127)는 종래의 방식으로 은퇴한다. 그러나, 컨텍스트 필드(133) 및 구성 필드(125)가 매칭하지 않는다면, 복구 절차가 수행된다. 예를 들면, 종속 명령 Vopp(127) 및 종속 명령 다음의 모든 명령들 둘 모두가 파일라인 및 재정렬 버퍼에서 플러싱되고, 프로그램 카운터(134)는 종속 명령 Vopp(127)를 폐칭하도록 설정되고, 상태 정보

(132b)는 정확한 상태를 복구하는데 사용된다.

[0038] 일부 실시예들에서, 명령 Vopp(127)가 설명된 체크를 수행하기 위해 은퇴할 준비가 될 때까지 대기할 필요가 없을 수 있다. 명령 Vopp(127)가 은퇴할 준비가 되기 전에, 구성 필드(125)에 대한 (명령 VMSR(126)과 같은 명령에 의해서든 또는 그렇지 않든지) 계류중인 어떠한 가능한 업데이트들도 존재하지 않는다면, 명령 Vopp(127)는 구성 필드(125)와 컨텍스트 필드(133)를 비교하고, 명령 Vopp(127)가 은퇴할 준비가 될 때까지 대기하지 않고서 필요한 경우 복구 절차를 수행할 수 있다.

[0039] 추측 명령 정보 큐(122)는 또한, 프로세서(100)가 부정확하게 실행된 브랜치 명령으로부터 복구할 수 있도록 브랜치 명령들 및 연관된 상태 정보를 저장하는데 사용될 수 있다. 이것은 도 1의 엔트리(136)를 포함하는 이유이다. (엔트리(136)는 상태 정보와 다른 정보를 포함할 수 있지만, 논의의 편의를 위해, 엔트리(136)가 상세히 설명되지 않는데, 왜냐하면 이것은 당분야에 잘 알려져 있기 때문이다.) 이러한 경우에, 추측 명령 정보 큐(122)는 이전에 논의된 바와 같이, BIQ의 기능을 포함한다. 그러나, 실시예들은 추측 명령 정보 큐(122)에 BIQ의 기능을 포함시키지 않을 수 있지만, 추측 명령 정보 큐(122)를 구현하는데 사용되는 물리적 버퍼와 분리된 BIQ에 대한 다른 물리적 버퍼를 사용할 수 있다.

[0040] 도 2a는 앞서 논의된 하나 이상의 실시예들에 의해 사용되는 방법을 예시한다. 상자(202)에 표시된 바와 같이, 컨텍스트 종속 명령들(가령, 벡터 명령)이 프로세싱되고, 추측 명령 정보 큐(122)에 배치된다. (프로그램 순서에서) 이전의 기록 명령이 실행되지 않지만, 컨텍스트 종속 명령이 프로세싱되고, 여기서 기록 명령은 값을 구성 필드(125)에 기록하는 명령이다. 명령은, 명령의 프로세싱 시에 구성 필드(125)의 값에 의해 결정된 컨텍스트 내에서 프로세싱된다.

[0041] 명령은 엔트리의 부분으로서 추측 명령 정보 큐(122)에 저장되고, 여기서 엔트리는 또한, 명령이 프로세싱된 때, 구성 필드(125)의 값과 동일한 값을 갖는 컨텍스트 필드를 포함한다.

[0042] 구성 필드(125)에 기록하는 기록 명령의 실행 후에(상자 204), 그 기록 명령에 종속하는 명령을 갖는 엔트리에 대한 추측 명령 정보 큐(122)가 탐색된다(상자 206). 이러한 탐색은, 기록 명령이 재정렬 버퍼(120) 앞에 있기 전에 수행될 수 있거나, 이것은, 기록 명령이 재정렬 버퍼(120) 앞에 도달하고 완료될 때 수행될 수 있다.

[0043] 어떠한 그러한 엔트리도 발견되지 않는다면, 도 2a의 결정(208) 및 타원형(210)에 표시된 바와 같이, 어떠한 특수 동작도 취해지지 않고, 파이프라인은 관습적으로 진행된다. 그러나, 기록 명령에 종속하는 명령을 포함하는 엔트리가 발견되면, 결정(211)에서, 명령이 정확한 컨텍스트에서 프로세싱되었는지를 결정하기 위해, 컨텍스트 필드(133)의 값은 기록 명령과 연관된 구성 필드 값과 비교된다. 명령이 정확한 컨텍스트에서 프로세싱되었다면, 프로세스 흐름은 어떠한 특수 동작도 취해지지 않는다는 것을 나타내기 위해 타원형(210)으로 복귀한다.

[0044] 그러나, 명령이 부정확한 컨텍스트에서 프로세싱되었다면, 상자(212)에 표시된 바와 같이, 부정확한 컨텍스트에서 명령을 프로세싱한 것으로부터 적절히 복구하기 위해 프로세서(100)에 대해 다양한 동작들이 취해진다. 상자(212)에서, 모든 동작중인 명령들이 파이프라인(102) 및 재정렬 버퍼(120)로부터 플러싱되고, 프로그램 카운터(134)는, (프로그램 순서에서) 기록 명령 바로 다음에 명령이 폐칭되도록 설정된다.

[0045] 도 2b는 앞서 논의된 실시예들 중 다른 실시예에 의해 사용되는 방법을 예시한다. 상자(202)에 표시된 바와 같이, 컨텍스트-종속 명령들(가령, 벡터 명령들)은 프로세싱되고, 추측 명령 정보 큐(122)에 배치된다. (프로그램 순서에서) 이전의 기록 명령이 실행되지 않았지만, 컨텍스트-종속 명령이 프로세싱되고, 여기서 기록 명령은 값을 구성 필드(125)에 기록하는 명령이다. 명령은, 명령이 프로세싱되는 시간에서 구성 필드(125)의 값에 의해 결정된 컨텍스트 내에서 프로세싱된다.

[0046] 명령은 엔트리의 부분으로서 추측 명령 정보 큐(122)에 저장되고, 여기서 엔트리는 또한, 명령이 프로세싱된 때 구성 필드(125)의 값과 동일한 값을 갖는 컨텍스트 필드를 포함한다.

[0047] 컨텍스트-종속 명령이 실행된 컨텍스트가 정확한지에 대한 체크는 2 개의 상이한 방식들로 수행될 수 있다. 하나의 실시예에서, 상자(224)에서, 컨텍스트-종속 명령이 은퇴할 준비가 될 때, 방법은 체크를 수행하기 위해 상자(211)로 계속된다. 다른 실시예에서, 상자(226)에서, 컨텍스트-종속 명령은, (명령 VMSR(126)과 같은 명령에 의해서든 또는 그렇지 않든지) 구성 값(125)에 대한 임의의 가능한 계류중인 업데이트들이 존재하는지를 결정하기 위해 체크한다. 계류중인 가능한 업데이트들이 존재하면, 방법은 대기하기 위해 상자(226)로 복귀한다. 어떠한 계류중인 가능한 업데이트들도 존재하지 않는다면, 방법은 체크를 수행하기 위해 상자(211)로 계

속된다.

[0048] 상자(211)에서, 컨텍스트 필드(133)의 값은, 명령이 정확한 컨텍스트에서 프로세싱되었는지를 결정하기 위해 기록 명령과 연관된 구성 필드 값과 비교된다. 명령이 정확한 컨텍스트에서 프로세싱되었다면, 프로세스 흐름은 어떠한 특수 동작도 취해지지 않는다는 것을 나타내기 위해 타원형(210)으로 복귀한다.

[0049] 그러나, 명령이 부정확한 컨텍스트에서 프로세싱되었다면, 상자(214)에 표시된 바와 같이, 부정확한 컨텍스트에서 명령을 프로세싱한 것으로부터 적절히 복구하기 위해 프로세서(100)에 대해 다양한 동작들이 취해진다. 상자(214)에서, 명령 및 (프로그램 순서에서) 명령 바로 다음의 모든 동작중인 명령들이 파이프라인(102) 및 재정렬 버퍼(120)로부터 플러싱되고, 프로그램 카운터(134)는, 명령이 폐칭될 다음 명령이 되도록 설정되고, 명령으로 엔트리에 저장된 상태 정보는 명령을 프로세싱하기 위해 프로세서(100)의 상태를 정확한 상태로 복구하는데 사용된다.

[0050] [0054] 일부 실시예들에서, 제어기(112)는 메모리 계층(114)과 같은 비일시적인 컴퓨터-판독 가능 매체에 저장된 펌웨어 또는 명령들을 실행함으로써 도 2a 또는 도 2b의 방법들을 수행할 수 있다.

[0051] [0055] 도 3은, 실시예들이 애플리케이션을 발견할 수 있는 무선 통신 시스템을 도시한다. 도 3은 기지국들(304A, 304B, 및 304C)을 포함하는 무선 통신 네트워크(302)를 도시한다. 도 3은, 소위 스마트 폰으로 지칭되는 모바일 셀룰러 통신 디바이스, 태블릿, 또는 컴퓨터 또는 컴퓨터 시스템과 같은 셀룰러 폰 네트워크에 적합한 일부 다른 종류의 통신 디바이스일 수 있는, 306으로 라벨링된 통신 디바이스를 도시한다. 통신 디바이스(306)는 모바일일 필요는 없다. 도 3의 특정 예에서, 통신 디바이스(306)는 기지국(304C)과 연관된 셀 내에 위치된다. 화살표들(308 및 310)은 업링크 채널 및 다운링크 채널 각각을 삽화식으로 표현하고, 통신 디바이스(306)는 업링크 채널 및 다운링크 채널에 의해 기지국(304C)과 통신한다.

[0052] [0056] 실시예들은, 예를 들어, 통신 디바이스(306)와, 또는 기지국(304C)과, 또는 둘 다와 연관된 데이터 프로세싱 시스템들에 이용될 수 있다. 도 3은, 본원에 설명된 실시예들이 채용될 수 있는 수많은 것들 중에서 오직 하나의 애플리케이션들만을 도시한다.

[0053] [0057] 실시예들은 벡터 명령들 및 FPSCR(124) 내의 하나 이상의 필드들에 대한 그들의 종속성의 문맥에서 설명되었다. 그러나, 실시예들은 벡터 명령들에 제한되지 않고, 그들의 컨텍스트에 대해 레지스터 내의 하나 이상의 필드들에 종속하는 다른 타입들의 명령들에 대한 애플리케이션을 발견할 수 있다. 예를 들면, 실시예들은, 자원의 생성과 그의 종속 소비 사이의 파이프라인에서 큰 스펜, 종속-생성 상태에 대한 비교적 드문 업데이트들, 및 종속-소비 명령들에 대한 고성능 요구들이 존재할 수 있는 프로세서 아키텍처들에서 애플리케이션을 발견할 수 있다. 이로써, 설명된 실시예들은 한 타입의 예측 방식으로서 보여질 수 있고, 여기서 종속 명령의 컨텍스트를 정의하는 상태는 상태가 종속 명령에 의해 소비되는 시간과 종속 명령이 완료된 시간 사이에서 변하지 않도록 예측된다. 이어서, 상태에 대한 변화들은 예측 실패를 발생시킨다. 이것은, 종속성이 조건부 또는 타겟 중 어느 하나를 수반하지 않고 비-브랜치 명령들을 수반한다는 점에서, 브랜치 예측 기술들과 구별되지만 유사한다.

[0054] [0058] 당업자들은, 정보 및 신호들이 다양한 상이한 기술들 및 기법들 중 임의의 기술 및 기법을 사용하여 표현될 수 있음을 인식할 것이다. 예를 들어, 상기 설명 전반에 걸쳐 언급될 수 있는 데이터, 명령들, 커맨드들, 정보, 신호들, 비트들, 심볼들, 또는 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 자기 입자들, 광학 필드들, 입자들 또는 이들의 임의의 결합에 의해 표현될 수 있다.

[0055] [0059] 추가적으로, 당업자들은, 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 회로들, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 이 둘의 결합들로서 구현될 수 있음을 인식할 것이다. 하드웨어와 소프트웨어의 이러한 상호교환가능성을 명확히 예시하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들, 및 단계들이 그들의 기능의 관점들에서 일반적으로 설명된다. 당업자가 인식할 바와 같이, 그러한 기능이 하드웨어로서 구현될지 또는 소프트웨어로서 구현될지는, 특정한 애플리케이션 및 전체 시스템에 부과된 설계 제약들에 의존한다. 당업자들은 설명된 기능을 각각의 특정한 애플리케이션에 대해 다양한 방식들로 구현할 수 있지만, 그러한 구현 결정들이 본 발명의 범위를 벗어나게 하는 것으로서 해석되지는 않아야 한다.

[0056] [0060] 본원에 개시된 실시예들과 관련하여 설명된 방법들, 시퀀스들 및/또는 알고리즘들은, 직접 하드웨어로 구현되거나, 프로세서에 의해 실행되는 소프트웨어 모듈로 구현되거나, 또는 이 둘의 조합으로 구현될 수 있다. 소프트웨어 모듈은 RAM 메모리, 플래시 메모리, ROM 메모리, EEPROM 메모리, EEPROM 메모리, 레지스터들, 하드

디스크, 탈착식 디스크, CD-ROM, 또는 당업계에 공지된 임의의 다른 형태의 저장 매체에 상주할 수 있다. 예시적인 저장 매체는, 프로세서가 저장 매체로부터 정보를 판독하고 저장 매체에 정보를 기록할 수 있도록, 프로세서에 커플링될 수 있다. 대안적으로, 저장 매체는 프로세서에 통합될 수 있다.

[0057]

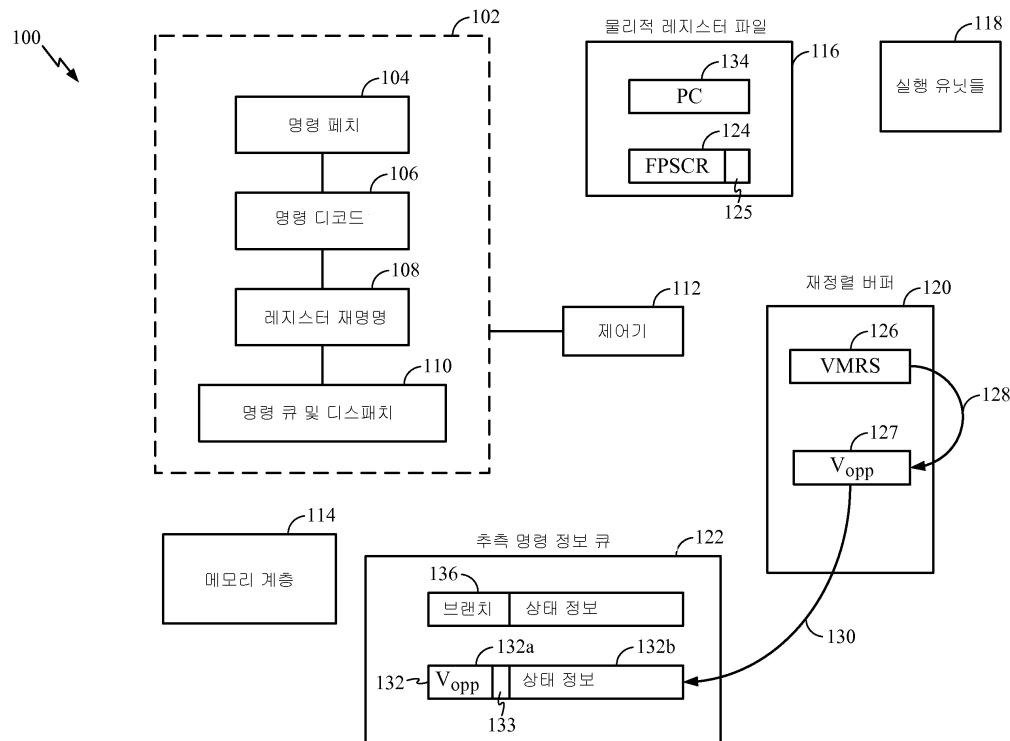
[0061] 따라서, 본 발명의 실시예는 앞서 설명된 방법을 구현하는 컴퓨터 판독 가능 매체들을 포함할 수 있다. 따라서, 본 발명은 예시된 실시예들로 제한되지 않고 본원에 설명된 기능을 수행하기 위한 임의의 수단이 본 발명의 실시예들에 포함된다.

[0058]

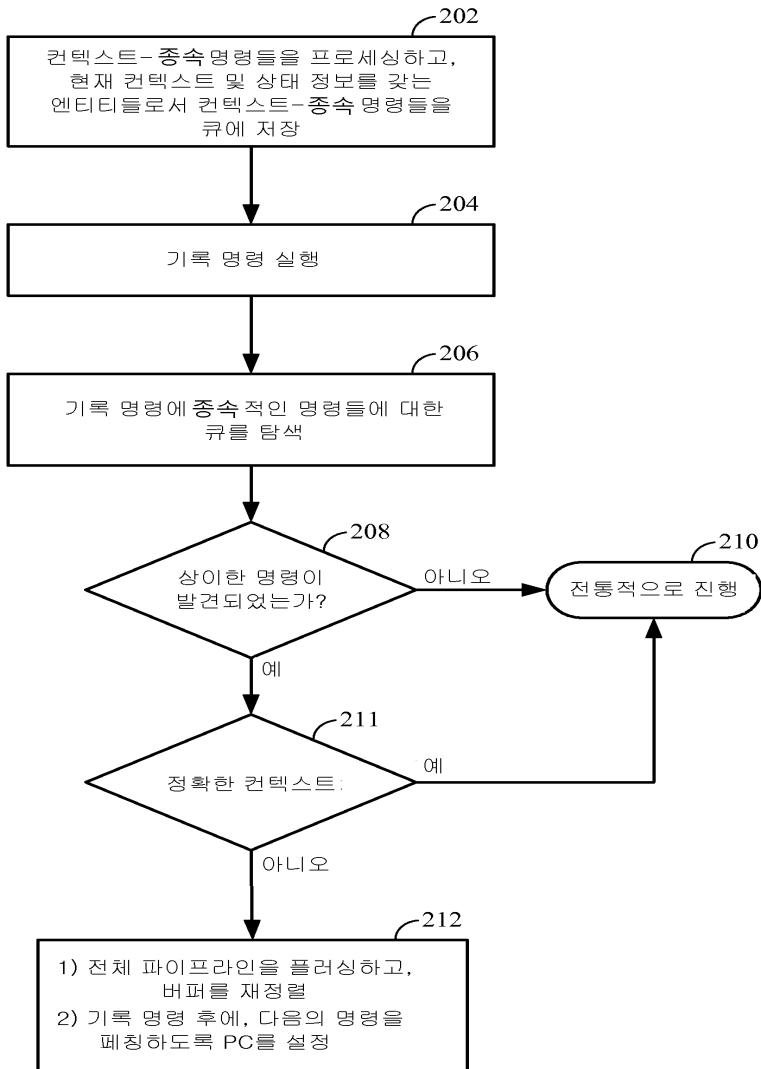
[0062] 전술한 본 발명이 본 발명의 예시적인 실시예들을 도시하지만, 첨부된 청구항들에 의해 정의된 바와 같은 본 발명의 범위를 벗어나지 않으면서 다양한 변화들 및 변경들이 행해질 수 있음을 주목해야 한다. 본 명세서에 설명된 본 발명의 실시예들에 따른 방법 청구항들의 기능들, 단계들 및/또는 동작들은 임의의 특정한 순서로 수행될 필요는 없다. 또한, 본 발명의 엘리먼트들이 단수로 설명 또는 청구될 수 있지만, 단수로의 제한이 명시적으로 언급되지 않으면, 복수가 고려된다.

도면

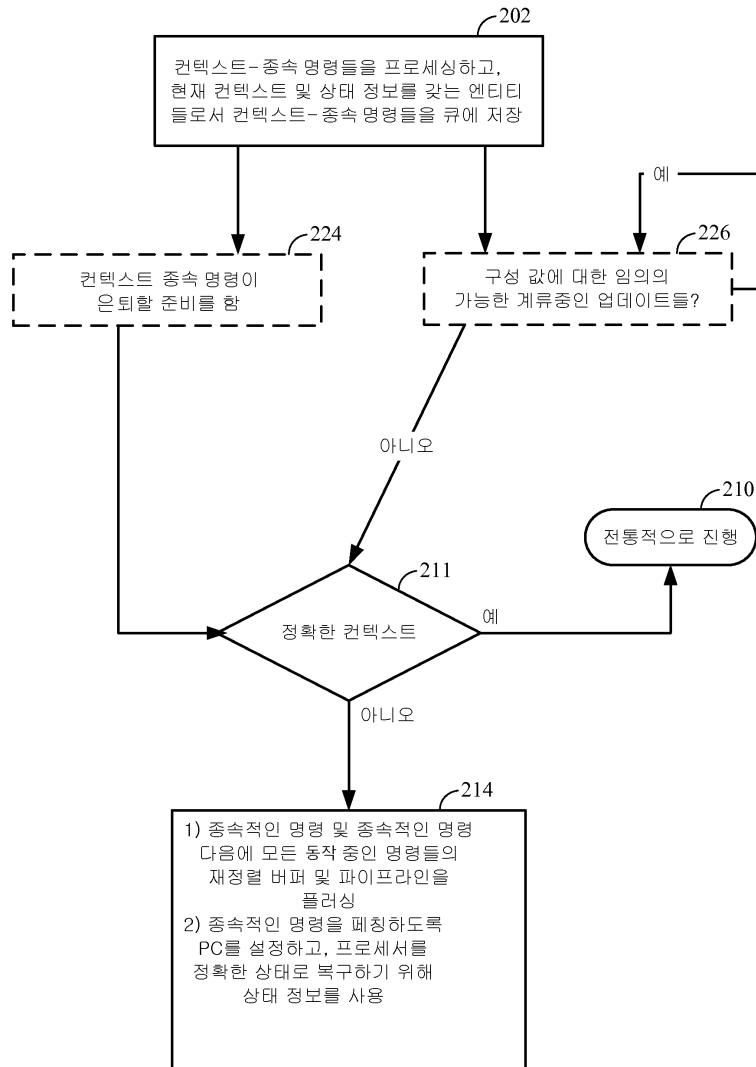
도면1



도면2a



도면2b



도면3

