



등록특허 10-2433782



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년08월17일
(11) 등록번호 10-2433782
(24) 등록일자 2022년08월12일

- (51) 국제특허분류(Int. Cl.)
G06F 11/10 (2006.01) *G06F 12/0875* (2016.01)
G06F 9/30 (2018.01)
- (52) CPC특허분류
G06F 11/1048 (2013.01)
G06F 11/1032 (2013.01)
- (21) 출원번호 10-2018-7037143
- (22) 출원일자(국제) 2017년06월02일
심사청구일자 2020년05월19일
- (85) 번역문제출일자 2018년12월20일
- (65) 공개번호 10-2019-0021247
- (43) 공개일자 2019년03월05일
- (86) 국제출원번호 PCT/US2017/035713
- (87) 국제공개번호 WO 2017/222784
국제공개일자 2017년12월28일

(30) 우선권주장
15/192,981 2016년06월24일 미국(US)

(56) 선행기술조사문현

US07302619 B1

(뒷면에 계속)

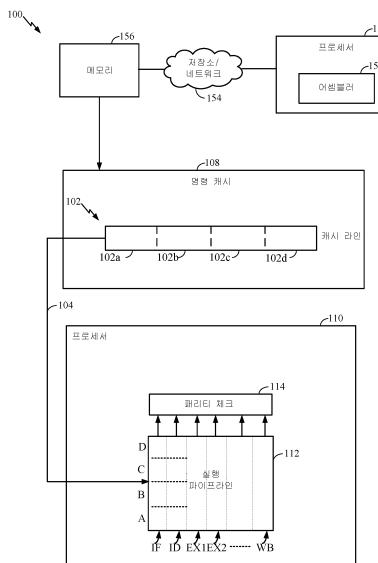
전체 청구항 수 : 총 13 항

심사관 : 김계준

(54) 발명의 명칭 명령 패킷들을 위한 패리티

(57) 요 약

명령들을 예리 체크하는 시스템들 및 방법은, 하나 이상의 명령들을 갖는 명령 패킷을 생성하고; 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하는지를 결정하고; 그리고 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하지 않는 경우, 미리 지정된 패리티와 매칭하도록 명령 패킷의 패리티를 변경하기 위해 명령 패킷의 비트를 사용하기 위한 어셈블러를 포함한다. 미리 지정된 패리티를 갖는 명령 패킷이 메모리에 저장되며, 최종적으로는 실행을 위해 프로세서에 의해 리트리브될 수 있다. 메모리로부터 리트리브된 명령 패킷에 예리가 존재하는 경우, 명령 패킷의 패리티를 미리 지정된 패리티와 비교하는 것에 기반하여 예리가 검출된다.

대 표 도 - 도1

(52) CPC특허분류

G06F 12/0875 (2013.01)

G06F 9/30145 (2013.01)

G06F 2212/452 (2013.01)

(56) 선행기술조사문헌

JP2009238168 A*

US20110307961 A1*

US20130036294 A1

An Architectural framework for support heterogeneous instructions set architectures, Silberman et al. IEEE, 1993.6.

US04005405 A*

*는 심사관에 의하여 인용된 문헌

명세서

청구범위

청구항 1

명령들을 에러 체크(error check)하는 방법으로서,

하나 이상의 명령들을 갖는 명령 패킷을 생성하는 단계;

상기 명령 패킷의 패리티(parity)가 미리 지정된 패리티와 매칭하는지를 결정하는 단계;

상기 명령 패킷의 패리티가 상기 미리 지정된 패리티와 매칭하지 않는 경우, 상기 미리 지정된 패리티와 매칭하도록 상기 명령 패킷의 패리티를 변경하기 위해 상기 명령 패킷의 비트를 사용하는 단계; 및

상기 미리 지정된 패리티를 갖는 상기 명령 패킷을 메모리에 저장하는 단계를 포함하고,

상기 명령 패킷은 VLIW(very long instruction word) 패킷이고, 그리고 상기 방법은, 상기 VLIW 패킷이 최대 가능한 개수 미만의 명령들을 포함한다는 것을 결정하고, 상기 VLIW 패킷에 NOP(no-operation) 필드를 삽입하고, 그리고 상기 명령 패킷의 패리티를 변경하기 위해 상기 명령 패킷의 비트로서 상기 NOP 필드의 비트를 사용하는 단계를 더 포함하는, 명령들을 에러 체크하는 방법.

청구항 2

제1항에 있어서,

상기 명령 패킷의 패리티를 상기 미리 지정된 패리티와 비교하는 것에 기반하여, 상기 메모리로부터 리트리브(retrieve)되는 상기 명령 패킷에 에러가 존재하는지를 결정하는 단계를 더 포함하는, 명령들을 에러 체크하는 방법.

청구항 3

제2항에 있어서,

상기 명령 패킷에 에러가 존재한다고 결정하고 그리고 예외(exception)를 생성하는 단계를 더 포함하는, 명령들을 에러 체크하는 방법.

청구항 4

제2항에 있어서,

상기 명령 패킷에 에러가 존재한다고 결정하고 그리고 디버거(debugger)에서의 분석을 위해 상기 에러를 기록하는 단계를 더 포함하는, 명령들을 에러 체크하는 방법.

청구항 5

제2항에 있어서,

프로세서에서의 상기 명령 패킷의 페치(fetch), 디코드, 하나 이상의 실행 스테이지들, 또는 라이트 백(write back)을 포함하는 하나 이상의 파이프라인(pipeline) 스테이지들 동안 상기 명령 패킷에 에러가 존재한다고 결정하는 단계를 포함하는, 명령들을 에러 체크하는 방법.

청구항 6

삭제

청구항 7

삭제

청구항 8

제1항에 있어서,

상기 비트는 상기 VLIW 패킷의 필드에 속하고, 상기 필드는 상기 VLIW 패킷의 명령들에 관련된 정보를 전달하는데 사용되는, 명령들을 에러 체크하는 방법.

청구항 9

제1항에 있어서,

상기 비트는 상기 명령 패킷의 미사용(unused) 비트인, 명령들을 에러 체크하는 방법.

청구항 10

제1항에 있어서,

상기 미리 지정된 패리티는 홀수 패리티 또는 짝수 패리티 중 하나인, 명령들을 에러 체크하는 방법.

청구항 11

장치로서,

프로세서 — 상기 프로세서는,

하나 이상의 명령들을 갖는 명령 패킷을 생성하고;

상기 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하는지를 결정하고; 그리고

상기 명령 패킷의 패리티가 상기 미리 지정된 패리티와 매칭하지 않는 경우, 상기 미리 지정된 패리티와 매칭하도록 상기 명령 패킷의 패리티를 변경하기 위해 상기 명령 패킷의 비트를 사용하도록

구성됨 —; 및

상기 미리 지정된 패리티를 갖는 상기 명령 패킷을 저장하도록 구성되는 메모리를 포함하고,

상기 명령 패킷은 VLIW(very long instruction word) 패킷이고, 그리고 상기 프로세서는, 상기 VLIW 패킷이 최대 가능한 개수 미만의 명령들을 포함한다는 것을 결정하고, 상기 VLIW 패킷에 NOP(no-operation) 필드를 삽입하고, 그리고 상기 명령 패킷의 패리티를 변경하기 위해 상기 명령 패킷의 비트로서 상기 NOP 필드의 비트를 사용하도록 추가로 구성되는, 장치.

청구항 12

제11항에 있어서,

상기 프로세서의 패리티 체크 블록은, 상기 명령 패킷의 패리티를 상기 미리 지정된 패리티와 비교하는 것에 기반하여, 상기 메모리로부터 리트리브되는 상기 명령 패킷에 에러가 존재하는지를 결정하도록 구성되는, 장치.

청구항 13

제12항에 있어서,

상기 프로세서는, 에러가 검출되는 경우 예외를 생성하도록 추가로 구성되는, 장치.

청구항 14

장치로서,

제1항 내지 제5항 및 제8항 내지 제10항 중 어느 한 항에 따른 방법을 수행하기 위한 수단을 포함하는, 장치.

청구항 15

코드를 포함하는 비-일시적인 컴퓨터-판독가능 저장 매체로서,

상기 코드는, 프로세서에 실행되는 경우, 상기 프로세서로 하여금 제1항 내지 제5항 및 제8항 내지 제10항 중

어느 한 항에 따른 방법을 수행하게 하는, 비-일시적인 컴퓨터-판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 개시된 양상들은 프로세싱 시스템의 명령 프로세싱에 관한 것이다. 더 구체적으로, 예시적인 양상들은, 명령 패킷들에서 에러들을 검출하기 위한 패리티(parity) 비트들의 사용에 관한 것이다.

배경 기술

[0002] 프로세서의 종래의 아키텍쳐에서, 프로세서는, 프로세싱 사이클에서, 명령 캐시 또는 메인 메모리와 같은 저장 위치로부터 하나 이상의 명령들을 수신할 수 있다. 각각의 프로세싱 사이클에서, 프로세서는, 하나 이상의 명령들을 하나 이상의 실행 유닛들에서 병렬로 실행할 수 있다. 하나 이상의 명령들 중 임의의 명령에서, 명령의 저장, 전달, 실행 등과 같은 명령의 라이프사이클(lifecycle)에 수반되는 스테이지들 또는 컴포넌트들 중 임의의 스테이지 또는 컴포넌트에서 비트 플립(flip)들 또는 에러들이 발생할 수 있다.

[0003] 예컨대, 캐시를 또는 메인 메모리와 같은 저장 위치들은, 에러들이 발생하기 쉬울 수 있는 SRAM(static random access memory), DDR(double data rate) 메모리 등과 같은 기술을 사용하여 구현될 수 있다. 비트 플립들은 또한, 저장 위치들로부터 프로세서의 실행 유닛들로 명령을 전달하는 데 수반되는 다양한 네트워크 버스들 및 와이어들에 도입될 수 있다. 또한, 실행 유닛들은, 명령을 연산하는 동안 로직 에러들에 또한 기억할 수 있다.

[0004] 명령의 라이프사이클의 각각의 스테이지에서 에러 체크(check)들을 부가하는 것은 비용이 많이 들 수 있고 일부 경우들에서는 비실용적일 수 있다. 예컨대, 명령 캐시의 각각의 캐시 라인 또는 메인 메모리/DDR의 각각의 저장 위치에 패리티 정보를 부가하는 것은 비용이 많이 들 수 있고, 저장 위치들의 에러들이 검출될 수 있는(그리고 일부 경우들에서는, 정정될 수 있는) 경우라 하더라도, 명령의 프로세싱의 다운스트림(downstream)에 비트 플립들이 도입될 수 있다. 또한, 명령을 전달하는 버스들 각각에 또는 명령을 연산하는 각각의 실행 유닛에 에러 체크 메커니즘들을 부가하는 것은 불가능할 수 있다. 따라서, 에러의 출처를 정확히 찾아내는 것은 어려울 수 있으며, 에러들이 체크되지 않은 채 남아 있는 경우, 그 에러들은 다양한 예외(exception)들 또는 시스템 장애들을 초래할 수 있다.

[0005] 따라서, 명령의 라이프사이클의 임의의 스테이지 및 임의의 컴포넌트 저장 위치들에서 또는 명령에 대한 저장 위치들 간의 운반에서 도입되는 에러들에 대한 엔드-투-엔드(end-to-end) 솔루션을 제공할 수 있는 효과적인 에러 체크 메커니즘들에 대한 대응하는 필요성이 존재한다.

발명의 내용

[0006] 본 발명의 예시적인 양상들은, 명령들을 에러 체크하기 위한 시스템들 및 방법들에 관련된다. 어셈블러(assembler)는, 하나 이상의 명령들을 갖는 명령 패킷을 생성하고; 명령 패킷의 패리티가 미리 지정된 (predesignated) 패리티와 매칭하는지를 결정하고; 그리고 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하지 않는 경우, 미리 지정된 패리티와 매칭하도록 명령 패킷의 패리티를 변경하기 위해 명령 패킷의 비트를 사용한다. 미리 지정된 패리티를 갖는 명령 패킷이 메모리에 저장되며, 최종적으로는 실행을 위해 프로세서에 의해 리트리브(retrieve)될 수 있다. 메모리로부터 리트리브된 명령 패킷에 에러가 존재하는 경우, 명령 패킷의 패리티를 미리 지정된 패리티와 비교하는 것에 기반하여 에러가 검출된다. 이러한 방식으로, 명령 패킷에 대한 엔드-투-엔드 에러 체크 능력이 제공된다.

[0007] 따라서, 예시적인 양상은 명령들을 에러 체크하는 방법에 관련되며, 그 방법은: 하나 이상의 명령들을 갖는 명령 패킷을 생성하는 단계; 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하는지를 결정하는 단계; 및 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하지 않는 경우, 미리 지정된 패리티와 매칭하도록 명령 패킷의 패리티를 변경하기 위해 명령 패킷의 비트를 사용하는 단계를 포함한다. 미리 지정된 패리티를 갖는 명령 패킷이 메모리에 저장되며, 메모리로부터 리트리브된 명령 패킷에 에러가 존재하는 경우, 명령 패킷의 패리티를 미리 지정된 패리티와 비교하는 것에 기반하여 에러가 결정된다.

[0008] 다른 예시적인 양상은 프로세서를 포함하는 장치에 관련되며, 그 프로세서는, 하나 이상의 명령들을 갖는 명령 패킷을 생성하고; 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하는지를 결정하고; 그리고 명령 패

킷의 패리티가 미리 지정된 패리티와 매칭하지 않는 경우, 미리 지정된 패리티와 매칭하도록 명령 패킷의 패리티를 변경하기 위해 명령 패킷의 비트를 사용하도록 구성된다. 메모리는 미리 지정된 패리티를 갖는 명령 패킷을 저장하도록 구성되며, 메모리로부터 리트리브된 명령 패킷에 에러가 존재하는 경우, 명령 패킷의 패리티를 프로세서의 패리티 체크 블록의 미리 지정된 패리티와 비교하는 것에 기반하여 에러가 결정된다.

[0009] 또 다른 예시적인 양상은 장치에 관련되며, 그 장치는: 하나 이상의 명령들을 갖는 명령 패킷을 생성하기 위한 수단; 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하는지를 결정하기 위한 수단; 및 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하지 않는 경우, 미리 지정된 패리티와 매칭하도록 명령 패킷의 패리티를 변경하기 위해 명령 패킷의 비트를 사용하기 위한 수단을 포함한다. 장치는, 미리 지정된 패리티를 갖는 명령 패킷을 저장하기 위한 수단, 및 메모리로부터 리트리브된 명령 패킷에 에러가 존재하는 경우, 미리 지정된 패리티와 명령 패킷의 패리티의 비교에 기반하여 에러를 결정하기 위한 수단을 더 포함한다.

[0010] 또 다른 예시적인 양상은 코드를 포함하는 비-일시적인 컴퓨터 관독가능 저장 매체에 관련되며, 코드는, 프로세서에 의해 실행되는 경우, 프로세서로 하여금 명령들을 에러 체크하는 방법을 수행하게 하고, 비-일시적인 컴퓨터 관독가능 저장 매체는, 하나 이상의 명령들을 갖는 명령 패킷을 생성하기 위한 코드; 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하는지를 결정하기 위한 코드; 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하지 않는 경우, 미리 지정된 패리티와 매칭하도록 명령 패킷의 패리티를 변경하기 위해 명령 패킷의 비트를 사용하기 위한 코드; 및 미리 지정된 패리티를 갖는 명령 패킷을 메모리에 저장하기 위한 코드를 포함한다. 비-일시적인 컴퓨터 관독가능 저장 매체는, 명령 패킷의 패리티를 미리 지정된 패리티와 비교하는 것에 기반하여, 메모리로부터 리트리브된 명령 패킷에 에러가 존재하는지를 결정하기 위한 코드를 더 포함한다.

도면의 간단한 설명

[0011] 첨부된 도면들은 본 발명의 양상들의 설명을 돋도록 제시되며, 양상들의 제한이 아니라 단지 양상들의 예시를 위해 제공된다.

[0012] 도 1은 본 개시내용의 일 양상에 따른 프로세싱 시스템을 예시한다.

[0013] 도 2는 본 개시내용의 양상들에 따른 예시적인 명령 패킷들을 예시한다.

[0014] 도 3은 본 개시내용의 일 양상에 따른, 명령들을 에러 체크하기 위한 방법을 예시한다.

[0015] 도 4는, 본 개시내용의 양상이 유리하게 이용될 수 있는 예시적인 컴퓨팅 디바이스를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0016] 본 발명의 양상들은, 본 발명의 특정 양상들에 관한 다음의 설명 및 관련 도면들에서 개시된다. 본 발명의 범위를 벗어나지 않으면서 대안적인 양상들이 고안될 수 있다. 부가적으로, 본 발명의 잘 알려진 엘리먼트들은, 본 발명의 관련 세부사항들을 불명료하게 하지 않기 위해, 상세하게 설명되지 않거나 생략될 것이다.

[0017] "예시적인"이라는 단어는, "예, 예증 또는 예시로서 기능하는" 것을 의미하도록 본원에서 사용된다. 본원에서 "예시적인" 것으로서 설명되는 임의의 양상은 반드시 다른 양상들에 비해 바람직하거나 유리한 것으로 해석될 필요는 없다. 유사하게, "본 발명의 양상들"이라는 용어는, 본 발명의 양상들 전부가 논의된 특성, 이점, 또는 동작 모드를 포함하는 것을 요구하지는 않는다.

[0018] 본원에서 사용된 용어는 특정 양상들을 설명하려는 목적만을 위한 것이며, 본 발명의 양상들을 제한하는 것으로 의도되지 않는다. 본원에 사용된 바와 같이, 맵락이 명확하게 달리 표시하지 않으면, 단수 형태들은 복수 형태들을 또한 포함하도록 의도된다. 추가로, "포함하다(comprise)", "포함하는(comprising)", "포함하다(include)" 및/또는 "포함하는(including)"이라는 용어들은, 본원에 사용될 때, 서술된 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 배제하지는 않는다는 것이 이해될 것이다.

[0019] 추가로, 많은 양상들은, 예컨대 컴퓨팅 디바이스의 엘리먼트들에 의해 수행될 동작들의 시퀀스들의 관점들에서 설명된다. 본원에 설명된 다양한 동작들은, 특정 회로들(예컨대, ASIC(application specific integrated circuit)들)에 의해, 하나 이상의 프로세서들에 의해 실행되는 프로그램 명령들에 의해, 또는 이 둘 모두의 결합에 의해 수행될 수 있다는 것이 인지될 것이다. 부가적으로, 본원에 설명된 동작들의 이러한 시퀀스는, 실행 시에, 연관된 프로세서로 하여금 본원에 설명된 기능성을 수행하게 할 컴퓨터 명령들의 대응하는 세

트가 저장된 임의의 형태의 컴퓨터 판독가능 저장 매체 내에서 완전히 구현되는 것으로 고려될 수 있다. 따라서, 본 발명의 다양한 양상들은 다수의 상이한 형태들로 구현될 수 있으며, 이들 전부는 청구된 요지의 범위 내에 있는 것으로 고려된다. 부가하여, 본원에 설명된 양상을 각각에 대해, 임의의 그러한 양상들의 대응하는 형태는, 예컨대, 설명된 동작을 수행 "하도록 구성된 로직"으로서 본원에서 설명될 수 있다.

[0016] [0020] 본 개시내용의 예시적인 양상들에서, 명령들에 대한 여러 체크들을 위한 메커니즘들은 명령들 그 자체에 내장될 수 있다(여기서, 종래의 명령들은, 명령들에 내장되거나 인코딩된 패리티 비트들과 같은 여러 체크 능력들을 갖지 않는다는 것이 인지됨). 예컨대, 하나 이상의 명령들의 그룹 또는 블록을 포함하는 명령 패킷을 고려하면, 명령 패킷에 몇몇 미사용(unused) 비트들이 존재할 가능성이 높다. 미사용 비트들은, 아키텍처적으로(architecturally) 예비된 비트들, 명령들의 블록에 의해 완전히 사용되지 않는 캐시 라인 또는 명령 워드의 부분들에서의 미사용 비트들, 하드웨어/실행 유닛들에 의해 무시될 수 있는 비트 패턴들 등일 수 있다. 명령 패킷의 미사용 비트들 중 하나 이상은, 예시적인 양상들에서, 여러 검출의 목적들을 위해 사용될 수 있다. 예컨대, 미사용 비트는(예컨대, 어셈블러에 의해) 패리티 비트로서 용도변경(repurpose)될 수 있는데, 즉, 미리 지정된 패리티(예컨대, 홀수 또는 짝수)와 매칭하게 명령 패킷의 패리티를 구성하도록 변경될 수 있다. 이어서, 명령 패킷은 메모리(예컨대, 메인 메모리, 이를테면 DDR, 플래시 메모리, RAM(random access memory) 등)에 저장될 수 있고, 후속하여, 예컨대, 명령 캐시에 공급될 수 있으며, 그 명령 캐시로부터, 프로세서에 의한 실행을 위해 명령 패킷이 리트리브될 수 있다. 위의 프로세스들의 임의의 포인트에서 명령 패킷의 패리티가(예컨대, 하드웨어에 의해 검출된 바와 같이) 미리 지정된 패리티와 매칭하지 않으면, 명령 패킷이 에러를 갖는다고 결정될 수 있다.

[0017] [0021] 위의 방식으로 미사용 비트로부터 생성되는 패리티에 기반하여, 명령 패킷에서 에러가 검출되면(즉, 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하지 않는다면), 다양한 액션 코스들(courses of action)이 가능하다. 일부 경우들에서, 명령 패킷은 에러를 갖는 것으로 마킹될 수 있고 그리고 향후의 분석 또는 복원을 위해 기록될 수 있다.

[0018] [0022] 일부 경우들에서, 명령 패킷이 미사용 비트들을 갖지 않는 경우라 하더라도, 예시적인 양상들에 따라 명령 패킷에 패리티 비트를 통합시키는 것이 여전히 가능할 수 있다. 예컨대, 명령 패킷은 당업계에 알려져 있는 바와 같은 VLIW(Very Long Instruction Word)의 형태로 명령들을 포함할 수 있으며, 여기서, 2개 이상의 명령들이 VLIW 패킷으로서 함께 제공된다. VLIW 패킷의 2개 이상의 명령들은 병렬로 실행되도록 설계된다. VLIW 패킷에 하나 이상의 미사용 비트들이 존재한다면, 하나 이상의 미사용 비트들 중의 미사용 비트는, VLIW 패킷의 패리티를 미리 지정된 또는 예상되는 패리티(예컨대, 홀수 또는 짝수)로 변경하는 데 사용될 수 있다. 그러나, 어떠한 미사용 비트들도 존재하지 않지만 단일 VLIW 패킷으로 그룹화되는 것이 가능한 최대 개수 미만의 명령들이 존재한다면, NOP(no-operation) 필드(즉, 실행되지 않는 더미(dummy) 명령)가 VLIW 패킷에 부가될 수 있고, NOP 필드의 비트가 패리티 비트로서 사용될 수 있다. 예컨대, VLIW 패킷의 끝을 지정할 수 있는 비트 패턴들과 같은 다른 아키텍처적으로 예비된 필드들이 또한, 미사용 비트 필드들이 용이하게 이용가능하지 않은 경우들에서, 패리티 비트를 생성하는 데 사용될 수 있다.

[0019] [0023] 이제, 위의 그리고 다른 예시적인 양상들이 도면들을 참조하여 아래에서 논의될 것이다. 일부 예시적인 양상들을 설명하기 위해 VLIW 패킷의 예가 사용될 것이지만, 본 개시내용의 양상들은, 하나 이상의 명령들을 포함하는 명령 패킷에 대한 임의의 다른 명령 포맷으로 쉽게 확장될 수 있다는 것이 이해될 것이다.

[0020] [0024] 이제 도 1을 참조하면, 예시적인 프로세싱 시스템(100)의 개략적인 표현이 예시된다. 도시된 바와 같이, 프로세싱 시스템(100)은, 본 개시내용의 양상들에 따라 명령 패킷들의 패리티를 생성하는 데 사용되는 어셈블러(152)를 포함할 수 있는 프로세서(150)를 포함할 수 있다. 어셈블러(152)는 소프트웨어로 구현될 수 있고, 프로세서(110)와 같은 프로세서에 의해 실행될 명령들을 어셈블링(assemble)하는 데 사용될 수 있다. 프로세서(150) 및 프로세서(110)가 별개로 도시되어 있지만, 이는 단지 예시를 위해서이며, 다양한 구현들에서, 프로세서들(150 및 110)은, 어셈블러를 실행할 뿐만 아니라 어셈블러에 의해 생성되는 명령들의 다운스트림 실행을 위한 실행 유닛들을 포함하는 동일한 프로세서 또는 프로세싱 유닛일 수 있다.

[0021] [0025] 도시된 바와 같이, 어셈블러(152)에 의해 생성되는 명령들 및/또는 명령 패킷들은, 중간 네트워크들/저장 매체(일반적으로, 참조 번호 154로 식별됨)를 통과하여, 메인 메모리(예컨대, DDR), 플래시 메모리, RAM 등 일 수 있는 메모리(156)에 저장될 수 있다. 메모리(156)로부터, 명령들은, (예컨대, 당업계에 알려져 있는 바와 같은, 다양한 요구-기반 페치(demand-based fetch)들, 프리페치(prefetch)들, 또는 이들의 조합들에 기반하여) 명령 캐시(108)에 제공될 수 있다.

- [0022] [0026] 프로세서(110)는 명령 캐시(108)에 커플링되는 것으로 도시된다. 프로세서(110)는, 명령 캐시(108)로부터 명령들을 수신하고, 예컨대, 실행 파이프라인(pipeline)(112)을 사용하여 명령들을 실행하도록 구성될 수 있다. 당업계에 알려져 있는 바와 같은, 데이터 캐시, L2(level 2) 캐시들 등과 같은 하나 이상의 다른 캐시들이 또한 프로세싱 시스템(100)에 존재할 수 있지만, 이들은 또한 단순화를 위해 예시되어 있지 않다.
- [0023] [0027] 프로세서(110)에서, 실행 파이프라인(112)이 도시되며, 이는, 명령 페치(IF; instruction fetch) 스테이지, 명령 디코드(ID; instruction decode) 스테이지, 하나 이상의 실행(EX1, EX2 등) 스테이지들, 및 라이트 백(WB; write back) 스테이지로서 대표적으로 예시되는 하나 이상의 파이프라인 방식(pipelined) 스테이지들을 포함할 수 있다. 당업자들은, 당업계에 알려져 있는 바와 같이, 실행 파이프라인(112)에 대한 다수의 수정들 및 부가들을 인지할 것이다.
- [0024] [0028] 일 양상에서, 명령 캐시(108)는 몇몇 캐시 라인들을 포함하며, 몇몇 캐시 라인들 중 캐시 라인(102)이 대표적으로 도시되어 있다. 캐시 라인(102)은, 하나 이상의 명령들, 또는 하나 이상의 명령들에 의해 형성되는 명령 패킷을 포함할 수 있다. 하나 초과의 명령들 또는 명령 패킷들 또는 이들의 조합들이 단일 캐시 라인에 포함될 수 있고; 더욱이, 명령 또는 명령 패킷이 하나 초과의 캐시 라인들에 걸쳐 있을 수 있다는 것이 이해될 것이다. 그러나, 개시된 양상들은 하나 이상의 캐시 라인들의 명령들 및/또는 명령 패킷들의 다양한 가능한 어레인지먼트들에 적용가능하다는 것을 유념하여, 본 개시내용의 단순화를 위해 하나 이상의 캐시 라인들의 명령들 및/또는 명령 패킷들의 다양한 어레인지먼트(arrangement)들이 총망라적으로 설명되어 있진 않다. 도시된 바와 같이, 캐시 라인(102)은, VLIW 명령들일 수 있는 한 세트의 명령들(102a-d)을 포함한다. 명령들(102a-d)은 레인(lane)들(A-D)을 각각 점유할 수 있으며, 여기서, 실행 파이프라인(112)은, 한 번에 하나의 명령을 실행하거나, VLIW 명령들의 경우에는, 2개 이상의 레인들(A-D)에서 2개 이상의 명령들이 동시에 또는 병렬로 동작하도록 구성될 수 있다. 명령들(102a-d)은, 도 1에서 버스(104)로서 총괄적으로 표현되는 다양한 상호연결부들 및 버스들을 통해 실행 파이프라인(112)에 제공될 수 있다.
- [0025] [0029] 인식될 바와 같이, 비트 에러들 또는 비트 플립들은 명령들(102a-d) 중 임의의 명령에서 발생할 수 있으며, 여기서, 이러한 에러들은, 명령들의 저장 위치(예컨대, 명령 캐시(108))에서; (예컨대, 버스(104)를 통한) 명령들의 전달 시에; 또는 실행 파이프라인(112)에서의 명령들의 실행 동안 발생할 수 있다. 명령들(102a-d)을 포함하는 VLIW 패킷의 집합적 패리티는 미리 지정된 패리티(예컨대, 홀수 또는 짝수 패리티)와 매칭할 것으로 예상된다. 예시적인 양상에서, 명령들(102a-d)을 포함하는 VLIW 패킷의 집합적 패리티가 미리 지정된 패리티와 본래 매칭하지 않았다면, VLIW 패킷의 패리티는, 미사용 비트들을 사용하여, 미리 지정된 패리티(예컨대, 홀수 또는 짝수 패리티)와 매칭하도록 (예컨대, 어셈블러(152) 또는 다른 소프트웨어에 의해) 변경되었을 수 있다. 이러한 방식으로 미리 지정된 패리티와 매칭하도록 VLIW 패킷의 패리티를 변경하는 데 사용될 수 있는 그러한 미사용 비트들의 다양한 예들이 도 2를 참조하여 설명될 것이다.
- [0026] [0030] 도 1에서, VLIW 패킷의 패리티가 미리 지정된 패리티에서 벗어나는지를 체크하기 위해, 패리티 체크(114)로서 지정된 블록이 제공될 수 있다. 도시된 바와 같이, VLIW 패킷의 명령들(102a-d)은, 다양한 구현들을 위해, 실행 유닛의 스테이지들 중 임의의 스테이지로부터 패리티 체크(114)로 공급될 수 있다. VLIW 패킷의 패리티가 미리 지정된 패리티에서 벗어나면, 패리티 체크(114)는 에러를 트리거링(trigger)할 수 있다. 당업자에 의해 이해될 수 있는 바와 같이, 패리티에 대해 단일 비트를 사용할 때, 단일 비트 에러가 검출될 수 있다. 따라서, 패리티 체크(114)는, 예컨대, 실행 유닛(112)으로부터 공급되는 명령들의 패리티에 기반하여 단일 비트 에러를 검출하기 위한 로직을 포함할 수 있다. 명령들(102a-d)과 같은 명령들의 라이프사이클의 임의의 포인트에서, 존재한다면, 패리티 및 에러들을 체크하기 위해, 유사한 패리티 체크 로직이 프로세싱 시스템(100)의 임의의 다른 컴포넌트 또는 스테이지에 부가될 수 있다. 본원에서 상세히 논의되진 않지만, 단일 비트 에러들을 에러 체크하는 위의 양상들을, 더 복잡한 에러 검출, 및 일부 경우들에서는, 하나 초과의 비트에서 에러들을 검출하고 일부 경우들에서는 또한 정정할 수 있는 에러 정정(예컨대, 에러 정정 코드(ECC; error correction code)를 구현함으로써 이루어짐)으로 확장시키는 것이 가능하다. 그러한 복잡한 에러 검출 및 정정 기법들은, 예컨대, 다수의 미사용 비트들이 명령 패킷 내에서 발견되고 ECC 비트들을 생성하기 위해 어셈블러(152)에 의해 사용되는 경우들에 적용가능할 수 있다.
- [0027] [0031] 이제 도 2를 참조하면, 참조 번호 200으로 지정된 명령 패킷이 캐시 라인(102)의 명령들(102a-d)을 포함하는 것으로 도시되어 있다. 명령 패킷(200)은, 일부 예들에서, 위에 논의된 바와 같은 VLIW 패킷일 수 있다. 도 2는, 명령 패킷(200)의 비트들(예컨대, 패리티 비트들로서 용도변경된 미사용 비트들 또는 예비된 비트들)이 패리티에 사용될 수 있는 예들을 도시한다. 명령 패킷(200)에 대해, 어셈블러(152)는, 명령 패킷(200)의 패리티가 이미 미리 지정된 패리티(예컨대, 짝수 또는 홀수)인지를 결정하고, 그렇다면, 예컨대, 패리티에

대한 추가적인 수정들 없이 명령 패킷(200)을 메모리(156)에 저장할 수 있다. 명령(200)의 패리티가 미리 지정된 패리티가 아니라면, 어셈블러(152)는, 미리 지정된 패리티와 매칭하도록 명령 패킷(200)의 패리티를 구성하기 위해 명령 패킷(200)의 비트(예컨대, 패리티를 지정하기 위해 용도변경된 미사용 비트 또는 예비된 필드)를 패리티 비트로서 사용하고, 예상되는 패리티를 갖는 명령 패킷(200)을 메모리(156)에 저장할 수 있다. 명령 패킷(200)이 (예컨대, 패리티 체크(114)에서) 에러 정정 다운스트림에 대해 테스트되는 임의의 포인트에서, 명령(200)의 패리티가 미리 지정된 패리티와 매칭하지 않으면, 명령 패킷(200)은 에러를 갖는 것으로 결정된다.

[0028] [0032] 이러한 방식으로 에러가 결정되면, 향후의 분석을 위해 에러가 로깅(log)될 수 있거나 예외가 생성될 수 있다. 그러한 에러 결정에 후속하는 정정 액션들은 본 개시내용에서 상세히 설명되지 않으며, 임의의 적절한 액션이 뒤따를 수 있다. 패리티에 사용되는 명령 패킷(200)의 미사용 비트들은 (그들이 미사용되므로) 어떤 식으로든 무시될 수 있고, 그러므로, 명령 패킷(200)의 패리티를 구성하기 위한 이러한 미사용 비트들에 대한 임의의 수정은 명령 패킷(200)의 명령들(102a-d)의 정상 동작/실행에 영향을 미치지 않을 것임이 인식될 것이다.

[0029] [0033] 패리티에 대사용될 수 있는 명령 패킷(200)의 비트들의 몇몇 예들이 이제 논의될 것이다. 일 예에서, 명령들 중 하나, 예컨대, 명령(102a)이 하나 이상의 미사용 비트들을 가질 수 있다. 예컨대, 명령(102a)은, 명령(102a)에 이용가능한 명령 공간(예컨대, 16 또는 32 비트)에서 이용가능한 비트들 전부를 요구하지는 않는 인코딩을 가질 수 있다. 비트(202)는 명령(102a)에서 그러한 미사용 비트일 수 있다. 어셈블러(152)는, 명령 패킷(200)의 패리티가 미리 지정된 패리티와 아직 일치하지 않는 경우, 명령 패킷(200)의 패리티를 미리 지정된 패리티와 매칭하게 만들기 위해 미사용 비트를 수정할 수 있다.

[0030] [0034] 다른 예에서, 어셈블러(152)는, 패리티 비트로서 사용하기 위해, 예비된 필드를 이용하는 것이 가능할 수 있다. 이와 관련하여, 명령들의 미사용 인코딩들이 패리티 비트로서 사용하기 위해 용도변경될 수 있다. 예컨대, 명령의 3개의 플레이버(flavor)들(예컨대, 로직 우측 시프트(right), 산술 우측 시프트, 및 좌측 시프트와 같은 시프트 명령의 3개의 플레이버들)이 존재한다면, 이러한 3개의 플레이버들은 3개의 상이한 인코딩들(예컨대, 시프트 명령들에 대한 연산 코드에 있는 2-비트 필드의 "00", "01", 및 "10")에 기반하여 구별될 수 있지만, 이러한 인코딩들 중 하나만이 2-비트 필드의 좌측 또는 최상위 비트 포지션에서 "1"을 가지며, 이는, 최상위 비트 포지션이 "1"일 때, 2-비트 필드의 우측 또는 최하위 비트 포지션이 패리티에 사용하기 위해 용도변경될 수 있다는 것을 의미한다.

[0031] [0035] 다른 예에서, 어셈블러는, 명령 패킷을 형성하도록 선택된 명령들 중 어떠한 명령도 패리티에 사용될 수 있는 미사용 비트들을 갖고 있지 않을 수 있다는 것을 인식할 수 있다. 그러한 경우에, 어셈블러는, 명령들 중 하나로서 NOP(no-operation)를 도입할 수 있다. NOP는, 실행되지 않는 더미 명령일 수 있다. 도시된 바와 같이, 일 예에서, 명령(102c)은 NOP(204)로서 생성될 수 있으며, 여기서, 명령(102c)의 모든 비트들이 본질적으로 미사용된다. 따라서, 이러한 경우에, 명령(102c)의 비트들 중 임의의 비트가 패리티에 사용될 수 있다.

[0032] [0036] 일부 예들에서, VLIW 패킷의 명령들의, 예컨대, 명령들의 사이즈, 명령들의 개수, 명령 경계(boundary)를 등을 지정하기 위한 VLIW 패킷 포맷들에 특정한 인코딩들이 패리티 비트를 생성하기 위해 어셈블러(152)에 의해 사용될 수 있다. 도 2에서 VLIW 패킷(200a)으로 지정된 명령 패킷(200)의 표현을 고려하면, 명령(102a)에 대한 필드에는, 후속하는 명령들(예컨대, 명령들(102b-d)에 대한 필드들에 있는 명령들 1-3)에 관련된 정보를 포함하는 프리픽스(prefix)가 있을 수 있다. 프리픽스는, VLIW 패킷(200a)에서 후속하는 명령들의 개수, 명령들 1-3의 대응하는 명령 길이들 등을 전달하기 위한 정보를 포함할 수 있지만, 이러한 정보를 전달하기 위해 명령(102a)의 이용가능한 비트들 전부를 요구하지는 않을 수 있다. 따라서, 어셈블러(152)는, VLIW 패킷(200a)의 패리티 지정을 위한 프리픽스의 하나 이상의 비트들을 사용하는 것이 가능할 수 있다.

[0033] [0037] 명령 패킷(200)을 VLIW 패킷(200b)으로 표현하는 것에서, 상이한 스타일의 인코딩이 도시되며, 여기서, VLIW 패킷(200b)이 계속 이어지는지 여부(즉, VLIW 패킷(200b)의 명령들이 대응하는 명령에 후속할 것인지 여부)(필드들(204a-d)에 대해 "K"), 또는 VLIW 패킷(200b)이 중단되는지 여부(필드(204d)에 대해 "S")를 표시하기 위해 대응하는 명령들(102a-d)의 필드들(204a-d)이 사용된다. 그러나 위의 예에서, 이러한 필드들(204a-d) 각각에 대해 2 비트들이 이용가능하다면, 2개의 가능성들(예컨대, "K" 또는 "S")을 구별하는 데 있어 하나의 비트가 충분할 수 있으며, 필드들(204a-d) 중 하나 이상의 필드의 나머지 하나의 비트는 VLIW 패킷(200b)의 패리티 지정을 위해 어셈블러(152)에 의해 사용될 수 있다.

[0034] [0038] 따라서, 어셈블러(152)는, 미사용 비트로부터 패리티 비트를 생성하거나 명령 패킷(200)의 비트를 용도변경하여 명령 패킷(200)의 패리티를 지정하기 위해, 위의 옵션들 또는 이들의 조합들 또는 변형들 중 임의의

하나 이상을 사용할 수 있다는 것을 알 수 있다. 따라서, 양상들은, 본원에 개시된 프로세스들, 기능들, 및/또는 알고리즘들을 수행하기 위한 다양한 방법들을 포함한다는 것이 인식될 것이다. 예컨대, 도 3에 예시된 바와 같이, 양상은, 명령들을 여러 체크하는 방법(300)(예컨대, 프로세서(110)에서의 명령들의 실행을 위해 어셈블러(152)에 의해 수행됨)을 포함할 수 있다.

[0035] [0039] 도시된 바와 같이, 블록(302)은, 하나 이상의 명령들을 갖는 명령 패킷을 생성하는 것을 포함할 수 있다(예컨대, 어셈블러(152)에서 명령들(102a-d)을 포함하는 명령 패킷(200)을 생성하며, 여기서, 어셈블러(152)는, 제한 없이, 동일한 프로세서(110)에 의해 또는 상이한 프로세서(150)에 의해 실행될 수 있고, 명령 패킷(200)은 VLIW 패킷일 수 있음).

[0036] [0040] 방법(300)은, 블록(304)에서, 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하는지를 결정하는 것을 포함할 수 있다(예컨대, 어셈블러(152)는, 명령 패킷(200)의 패리티가 미리 지정된 짝수 또는 홀수 패리티와 매칭하는지를 결정할 수 있음).

[0037] [0041] 블록(304)에서, 명령 패킷의 패리티가 미리 지정된 패리티와 매칭하지 않는다고 결정되면, 블록(306)에서, 미리 지정된 패리티와 매칭하도록 명령 패킷의 패리티를 변경하기 위해 명령 패킷의 비트를 사용한다. 예컨대, 어셈블러(152)는, 미리 지정된 패리티와 매칭하도록 명령 패킷의 패리티를 변경하기 위해 명령 패킷(200)의 비트(예컨대, 미사용 비트, NOP 필드의 비트, 예비된 비트 등)를 패리티 비트로서 사용할 수 있다.

[0038] [0042] 블록(308)에서, (블록(306)의 단계를 수행한 이후, 또는 명령 패킷의 패리티가 미리 지정된 패리티와 이미 매칭한다고 결정된 경우에는 블록(306)을 스kip하고 블록(304)으로부터 블록(308)에 도달함으로써) 미리 지정된 패리티와 매칭하는 패리티를 갖는 명령 패킷이, 예컨대, 메모리(156)에 저장된다.

[0039] [0043] 블록(308)으로부터, 방법(300)은, 예컨대, 명령 패킷(200)을 명령 캐시(108)에 저장하고, 그 명령 캐시(108)로부터, 예컨대, 프로세서(110)의 실행 파이프라인(112)에서 명령 패킷(200)이 페칭(fetch)되고, 디코딩되고, 그리고 실행되는 후속 스테이지들을 수반할 수 있다. 이러한 스테이지들 중 임의의 스테이지에서, 명령 패킷(200)의 패리티가 (예컨대, 프로세서(110)의 실행 파이프라인(112)의 IF, ID, EX1, EX2, WB 등의 스테이지들에 대한 패리티 체크(114)에 의해) 체크될 수 있다. 패리티가 미리 지정된 패리티와 매칭하지 않으면, 향후의 분석을 위해 로깅될 수 있는 에러가 결정되거나, 예외가 생성될 수 있다. 에러가 결정된 경우, (본 개시내용의 범위를 벗어나는) 임의의 적절한 정정 액션들이 또한 구현될 수 있다.

[0040] [0044] 이제 도 4를 참조하면, 예시적인 양상들에 따라 구성되는 컴퓨팅 디바이스의 블록도가 도시되며, 일반적으로 400으로 지정된다. 컴퓨팅 디바이스(400)는 도 1의 프로세서(110)를 포함한다. 예시적인 양상들에서, 프로세서(110)는 어셈블러(이를테면, 어셈블러(152))를 실행할 수 있고, 도 3의 방법(300)을 수행하도록 구성될 수 있다. 도 1을 참조하여 논의된 바와 같이, 프로세서(110)는 메모리(156)에 통신가능하게 커플링될 수 있으며, 메모리(156)와 프로세서(110) 사이에 명령 캐시(108)가 대표적으로 도시된다. 실행 파이프라인(112) 및 패리티 체크(114)가 또한 프로세서(110)에 도시되어 있지만, 도 1에 도시된 실행 파이프라인(112)의 세부사항들은 명확화를 위해 해당 도면에서 생략되어 있다. 명령 패킷(200)은 명령 캐시에 도시되며, 캐시 라인(102)에 속할 수 있다. 다양한 다른 세부사항들, 이를테면, 프로세서(110)와 메모리(156) 사이에 존재할 수 있는 하나 이상의 캐시들 및 다른 상호연결부들이 또한 단순화를 위해 도 4에 예시되어 있지 않다.

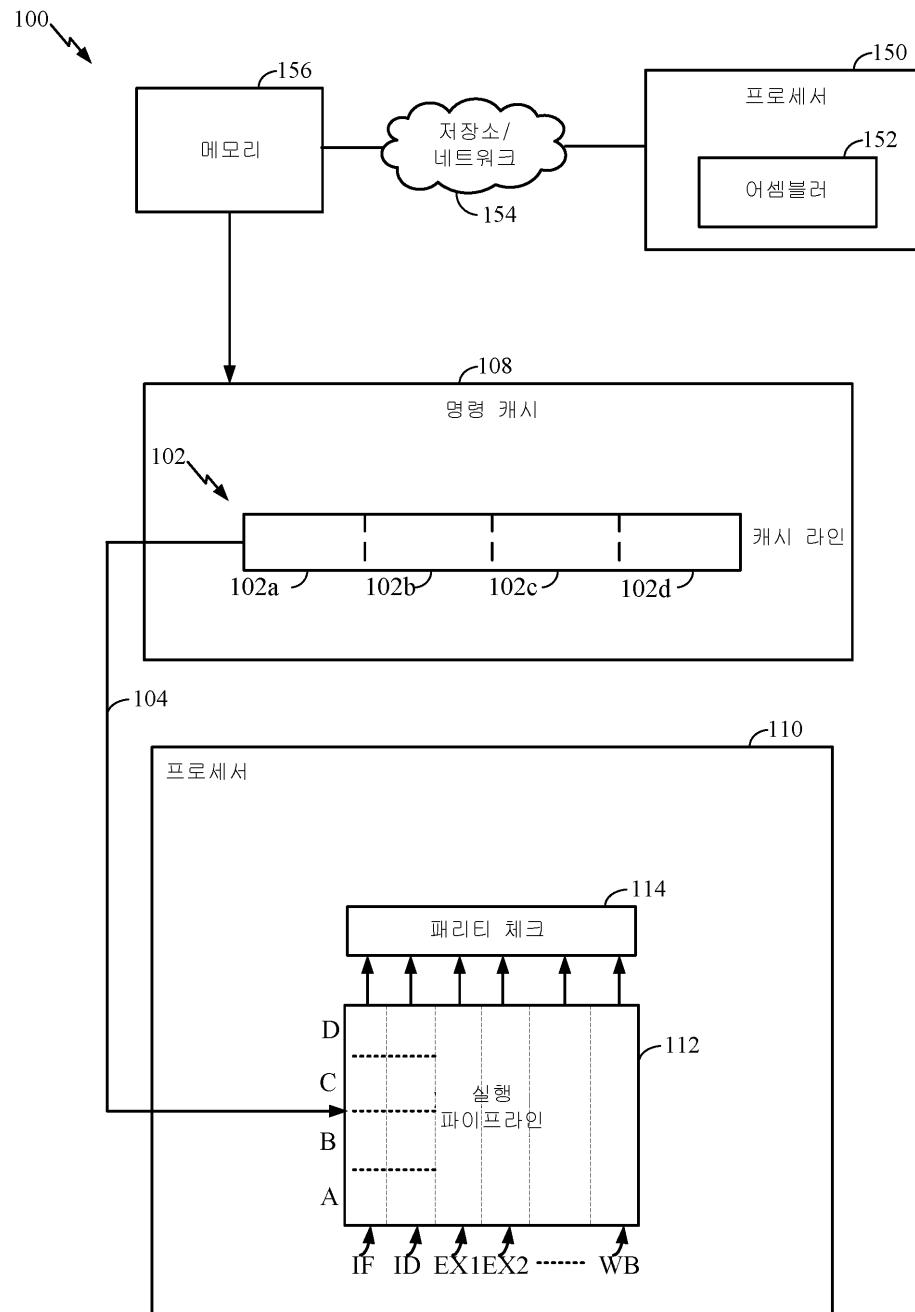
[0041] [0045] 일부 양상들에서, 도 4의 컴퓨팅 디바이스(400)는, 모바일 디바이스 또는 사용자 단말과 같은 무선 통신 디바이스로서 구성될 수 있다. 따라서, 일부 양상들에서, 도 4는, 파선들로 도시된 몇몇 선택적인 블록들을 포함할 수 있다. 예컨대, 컴퓨팅 디바이스(400)는 선택적으로, 디스플레이(428) 및 디스플레이 제어기(426) – 디스플레이 제어기(426)는 프로세서(110) 및 디스플레이(428)에 커플링됨 –; 프로세서(110)에 커플링되는 코더/디코더(CODEC; coder/decoder)(434)(예컨대, 오디오 및/또는 보이스 CODEC); CODEC(434)에 커플링되는 스피커(436) 및 마이크로폰(438); 및 무선 안테나(442) 및 프로세서(110)에 커플링되는 무선 제어기(440)(모뎀을 포함할 수 있음)를 포함할 수 있다.

[0042] [0046] 이러한 선택적인 블록들 중 하나 이상의 블록들이 존재하는 특정 양상에서, 프로세서(110), 디스플레이 제어기(426), 메모리(432), CODEC(434), 및 무선 제어기(440)는 시스템-인-패키지(system-in-package) 또는 시스템-온-칩 디바이스(422)에 포함될 수 있다. 입력 디바이스(430), 전력 공급부(444), 디스플레이(428), 스피커(436), 마이크로폰(438), 무선 안테나(442), 및 전력 공급부(444)는 시스템-온-칩 디바이스(422) 외부에 있을 수 있고 그리고 시스템-온-칩 디바이스(422)의 컴포넌트, 이를테면, 인터페이스 또는 제어기에 커플링될 수 있다.

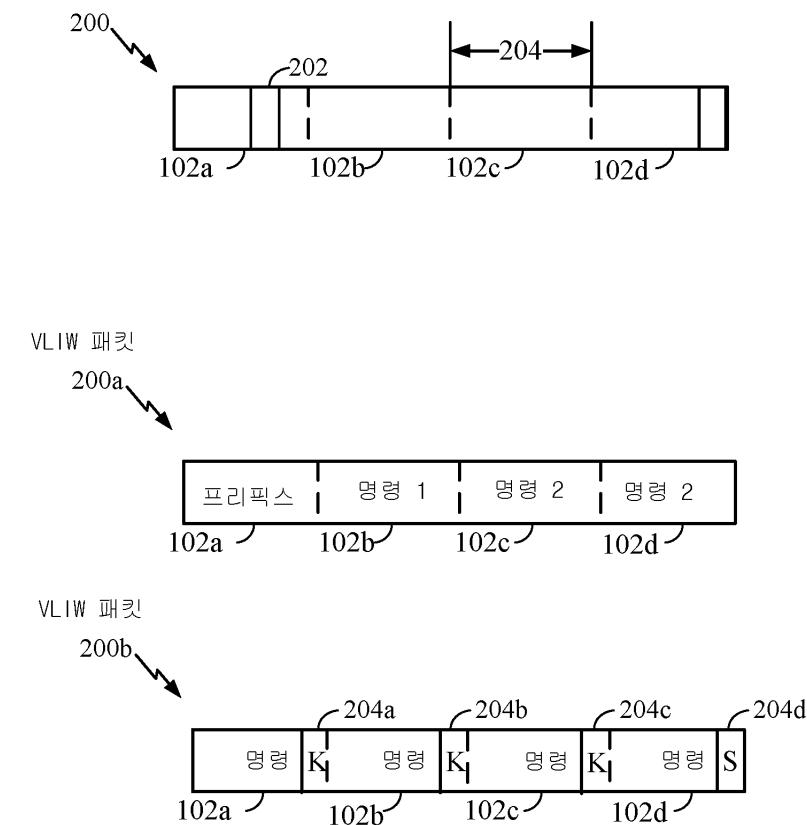
- [0043] [0047] 도 4가 컴퓨팅 디바이스(위에 언급된 바와 같이, 일부 양상들에서, 무선 통신들에 사용될 수 있음)를 도시하지만, 프로세서(110) 및 메모리(432)는 또한, 셋톱 박스, 뮤직 플레이어, 비디오 플레이어, 엔터테인먼트 유닛, 내비게이션 디바이스, 개인 휴대 정보 단말(PDA), 고정 위치 데이터 유닛, 컴퓨터, 랩톱, 태블릿, 통신 디바이스, 모바일 폰, 서버, 또는 다른 유사한 디바이스들에 통합될 수 있다는 것이 유의되어야 한다.
- [0044] [0048] 정보 및 신호들이 다양한 상이한 기술들 및 기법들 중 임의의 기술 및 기법을 사용하여 표현될 수 있음을 당업자들은 인식할 것이다. 예컨대, 위의 설명 전반에 걸쳐 참조될 수 있는 데이터, 명령들, 커맨드들, 정보, 신호들, 비트들, 심볼들, 및 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 자기 입자들, 광학 필드들 또는 광학 입자들, 또는 이들의 임의의 조합에 의해 표현될 수 있다.
- [0045] [0049] 추가로, 본원에 개시된 양상들과 관련하여 설명된 다양한 예시적인 로직 블록들, 모듈들, 회로들, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 이 둘의 조합들로서 구현될 수 있음을 당업자들은 인식할 것이다. 하드웨어와 소프트웨어의 이러한 상호교환가능성을 명확히 예시하기 위해, 다양한 예시적인 컴퓨트론트들, 블록들, 모듈들, 회로들, 및 단계들은 그들의 기능성의 관점에서 일반적으로 위에 설명되었다. 이러한 기능성이 하드웨어로서 구현되는지 또는 소프트웨어로서 구현되는지는 특정 애플리케이션 및 전체 시스템에 부과된 설계 제한들에 의존한다. 당업자들은 설명된 기능성을 각각의 특정 애플리케이션에 대해 다양한 방식들로 구현할 수 있지만, 이러한 구현 결정들이 본 발명의 범위를 벗어나게 하는 것으로 해석되어서는 안 된다.
- [0046] [0050] 본원에 개시된 양상들과 관련하여 설명되는 방법들, 시퀀스들, 및/또는 알고리즘들은 직접적으로 하드웨어로, 프로세서에 의해 실행되는 소프트웨어 모듈로, 또는 이 둘의 조합으로 구현될 수 있다. 소프트웨어 모듈은, RAM 메모리, 플래시 메모리, ROM 메모리, EEPROM 메모리, 레지스터들, 하드 디스크, 착탈식 디스크, CD-ROM, 또는 당업계에 알려져 있는 임의의 다른 형태의 저장 매체에 상주할 수 있다. 예시적인 저작 매체는, 프로세서가 저장 매체로부터 정보를 판독하고 저장 매체에 정보를 기입할 수 있도록 프로세서에 커플링된다. 대안으로, 저장 매체는 프로세서에 통합될 수 있다.
- [0047] [0051] 따라서, 본 발명의 양상은, 패리티 비트들을 사용하여 명령 패킷들에서 에러를 검출하기 위한 방법을 구현하는 컴퓨터 판독가능 매체를 포함할 수 있다. 따라서, 본 발명은 예시된 예들로 제한되지 않으며, 본원에 설명된 기능성을 수행하기 위한 임의의 수단은 본 발명의 양상들에 포함된다.
- [0048] [0052] 전술한 개시내용은 본 발명의 예시적인 양상들을 도시하지만, 첨부된 청구항들에 의해 정의되는 바와 같은 본 발명의 범위를 벗어나지 않으면서 다양한 변화들 및 수정들이 본원에서 이루어질 수 있음이 유의되어야 한다. 본원에 설명된 본 발명의 양상들에 따른 방법 청구항들의 기능들, 단계들 및/또는 동작들은 임의의 특정 순서로 수행될 필요는 없다. 또한, 본 발명의 엘리먼트들이 단수로 설명되거나 청구될 수 있지만, 단수에 대한 제한이 명시적으로 언급되지 않으면 복수가 고려된다.

도면

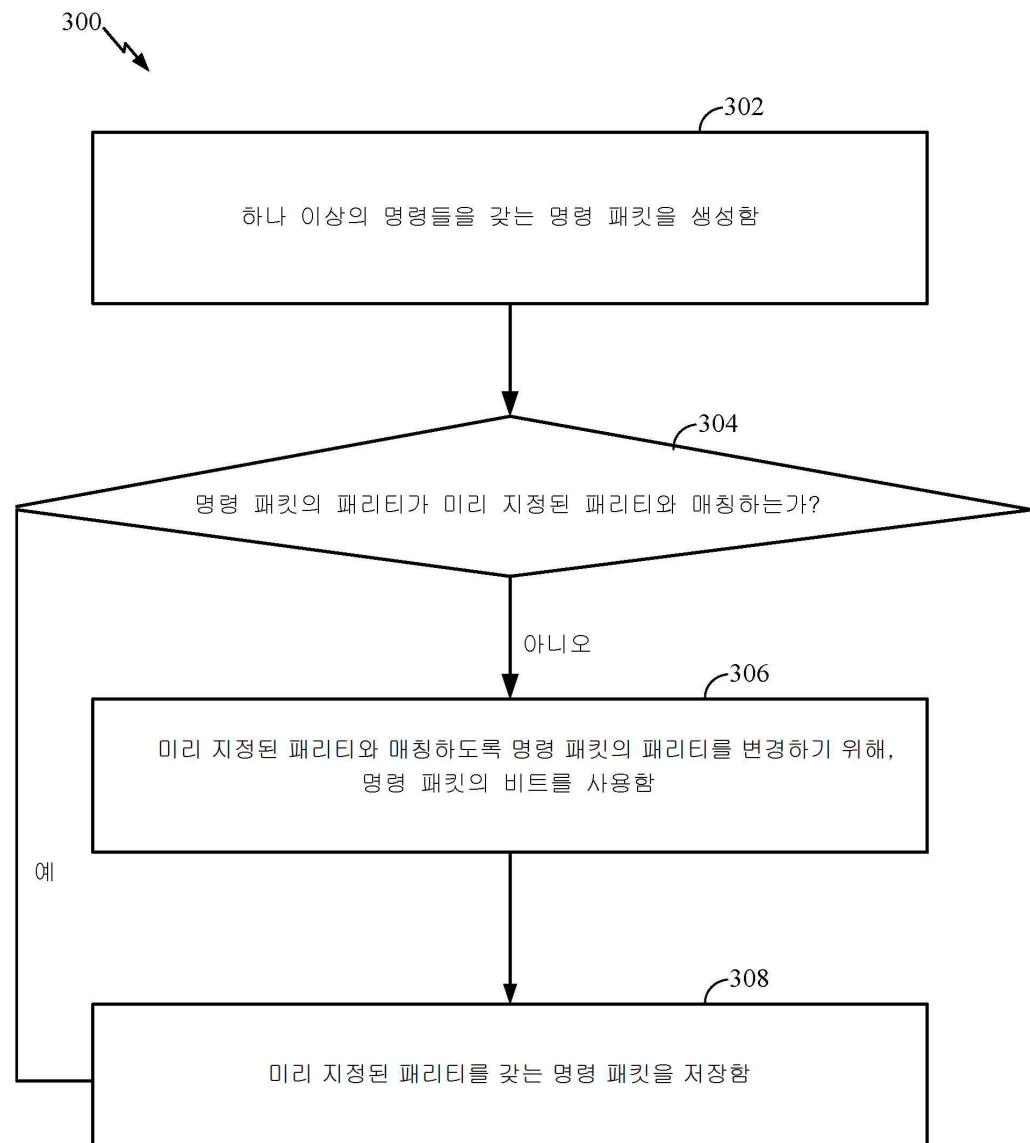
도면1



도면2



도면3



도면4

