



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2024년07월04일  
(11) 등록번호 10-2681251  
(24) 등록일자 2024년06월28일

- (51) 국제특허분류(Int. Cl.)  
G06F 9/30 (2018.01) G06F 9/4401 (2018.01)
  - (52) CPC특허분류  
G06F 9/30043 (2013.01)  
G06F 13/1689 (2013.01)
  - (21) 출원번호 10-2021-7022052
  - (22) 출원일자(국제) 2020년01월14일  
심사청구일자 2021년07월30일
  - (85) 번역문제출일자 2021년07월13일
  - (65) 공개번호 10-2021-0108973
  - (43) 공개일자 2021년09월03일
  - (86) 국제출원번호 PCT/EP2020/050757
  - (87) 국제공개번호 WO 2020/156797  
국제공개일자 2020년08월06일
  - (30) 우선권주장  
19154735.5 2019년01월31일  
유럽특허청(EPO)(EP)
  - (56) 선행기술조사문헌  
US06286095 B1\*  
US20090063898 A1\*  
US20110320764 A1\*  
US20150378737 A1\*
- \*는 심사관에 의하여 인용된 문헌

- (73) 특허권자  
인터내셔널 비즈니스 머신즈 코포레이션  
미국 10504 뉴욕주 아몬크 뉴오차드 로드
- (72) 발명자  
라이취, 크리스토프  
독일 보에블링겐 71032, 쉐나이처 스트리트 220,  
IBM 독일 리서치 앤드 디벨롭먼트 지엠비에이치  
크라에머, 마르코  
독일 보에블링겐 71032, 쉐나이처 스트리트 220,  
IBM 독일 리서치 앤드 디벨롭먼트 지엠비에이치  
(뒷면에 계속)
- (74) 대리인  
허정훈

전체 청구항 수 : 총 20 항

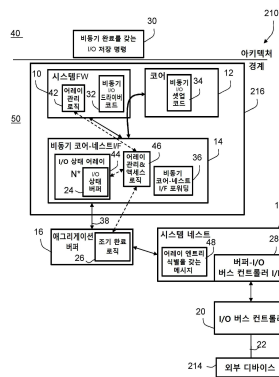
심사관 : 지정훈

(54) 발명의 명칭 입/출력 저장 명령의 처리

(57) 요약

입/출력 저장 명령(30)을 처리하기 위한 데이터 처리 시스템(210) 및 방법으로서, 입/출력 버스 컨트롤러(20)에 의해서 적어도 하나의 입/출력 버스(22)에 통신 가능하게 결합된 시스템 네스트(18)를 포함한다. 상기 데이터 처리 시스템(210)은, 코어(12), 시스템 펌웨어(10) 및 비동기 코어-네스트 인터페이스(14)를 포함하는 적어도 하나 (뒷면에 계속)

대표도 - 도1



의 데이터 처리 유닛(216)을 더 포함한다. 상기 데이터 처리 유닛(216)은 애그리게이션 버퍼(16)를 통해 상기 시스템 네스트(18)에 통신 가능하게 결합된다. 상기 시스템 네스트(18)는 상기 입/출력 버스(22)에 통신 가능하게 결합된 적어도 하나의 외부 디바이스(214)로부터 데이터를 비동기로 로드하고 및/또는 상기 적어도 하나의 외부 디바이스(214)에 데이터를 비동기로 저장하도록 구성된다. 상기 시스템 네스트(18)에서 상기 입/출력 저장 명령(30)의 실행이 완료되기 전에 상기 입/출력 저장 명령(30)을 완료하도록 상기 데이터 처리 유닛(216)이 구성된다. 상기 비동기 코어-네스트 인터페이스(14)는 다수의 입/출력 상태 버퍼들(24)을 갖는 입/출력 상태 어레이(44)를 포함한다.

(52) CPC특허분류

*G06F 9/4406* (2013.01)

*G06F 9/4411* (2013.01)

(72) 발명자

**레너트, 프랭크**

독일 보에블링겐 71032, 쉐나이처 스트리트 220,  
IBM 독일 리서치 앤드 디벨롭먼트 지엠비에이치

**클라인, 마티아스**

미국 뉴욕 12601, 포킵시, 사우스 로드 2455, 아이  
비엠 코포레이션

**브래드버리, 조나단**

미국 뉴욕 12601, 포킵시, 사우스 로드 2455, 아이  
비엠 코포레이션

**자코비, 크리스찬**

미국 뉴욕 12601, 포킵시, 사우스 로드 2455, 아이  
비엠 코포레이션

**드라이에버, 피터**

미국 뉴욕 12601, 포킵시, 사우스 로드 2455, 아이  
비엠 코포레이션

**벨마, 브렌던**

미국 뉴욕 12601, 포킵시, 사우스 로드 2455, 아이  
비엠 코포레이션

## 명세서

### 청구범위

#### 청구항 1

입/출력 저장 명령을 처리하기 위한 데이터 처리 시스템에서, 상기 데이터 처리 시스템은:

코어 및 시스템 펌웨어를 포함하는 데이터 처리 유닛을 포함하고, 상기 데이터 처리 유닛은 방법을 수행하도록 구성되며, 상기 방법은:

상기 입/출력 저장 명령을 사용하여 명시된 주소에 의해서 입/출력 함수(an input/output function)를 식별하는 단계(identifying) - 상기 입/출력 저장 명령은 주소를 통한 오프셋(an offset through an address), 전송될 적어도 하나의 데이터 및 전송될 데이터에 대한 포인터, 및 상기 데이터의 길이로 적어도 상기 입/출력 함수를 명시함 -;

상기 입/출력 함수에 대한 액세스가 주소 공간 상에서 그리고 게스트 인스턴스 레벨 상에서 허용되는 지를 확인하는 단계(verifying)

상기 데이터 처리 유닛과 다른 상기 데이터 처리 시스템의 선택된 컴포넌트에서 상기 입/출력 저장 명령의 실행이 완료되기 전에 상기 입/출력 저장 명령을 완료하는 단계(completing) - 상기 선택된 컴포넌트는 비동기로(asynchronously) 적어도 하나의 외부 디바이스로부터 데이터를 로드하고 적어도 하나의 외부 디바이스에 데이터를 저장함-;

실패한 비동기 실행의 데이터를 전송하는, 상기 데이터 처리 유닛에서 상기 입/출력 저장 명령의 비동기 실행 동안, 에러를 검출하는 것에 기초하여, 인터럽트를 통해 통지를 제공하는 단계(providing);

상기 입/출력 저장 명령의 완료 메시지(a completion message)를 수집하고(collect) 수신된 완료 메시지들에 기초하여 다수의 입/출력 상태 버퍼들 중 하나 또는 그 이상의 상태 버퍼들을 갱신하는 단계(updating); 그리고

상기 다수의 입/출력 상태 버퍼들 중 하나의 상태 버퍼가 미실행(pending) 입/출력 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 상기 입/출력 저장 명령의 실행을 지연하는 단계(delaying)를 포함하는

데이터 처리 시스템.

#### 청구항 2

제1항에 있어서, 상기 방법은 상기 다수의 입/출력 상태 버퍼들 중 하나의 상태 버퍼가 하나 또는 그 이상의 미실행(pending) 입/출력 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 비동기 입/출력 저장 명령 및 동기 입/출력 저장 명령 중 적어도 하나의 실행을, 상기 데이터 처리 유닛에 의해서, 지연하는 단계를 더 포함하는

데이터 처리 시스템.

#### 청구항 3

제1항에 있어서, 상기 선택된 컴포넌트는 애그리게이션 버퍼(an aggregation buffer)를 통해 상기 데이터 처리 유닛에 통신 가능하게 결합된 시스템 네스트(a system nest)인

데이터 처리 시스템.

#### 청구항 4

제3항에 있어서, 상기 데이터는, 상기 데이터의 길이가 정의된 크기를 초과하는 것에 기초하여, 상기 애그리게이션 버퍼에 다수의 데이터 패킷들로 조기 완료 메시지를 갖는 비동기 전송 메커니즘을 통해 상기 입/출력 저장 명령에 의해서 전송되는

데이터 처리 시스템.

**청구항 5**

제3항에 있어서, 상기 애그리게이션 버퍼는 상기 데이터 처리 유닛의 비동기 코어-네스트 인터페이스(an asynchronous core-nest interface)에 통신 가능하게 결합되고, 상기 비동기 코어-네스트 인터페이스는 상기 다수의 입/출력 상태 버퍼들을 포함하는 입/출력 상태 어레이를 포함하는

데이터 처리 시스템.

**청구항 6**

제1항에 있어서, 상기 방법은 메시지 상태들을 상기 선택된 콤포넌트로부터 수집하는 단계(collecting)를 더 포함하고, 상기 메시지 상태들은 상기 선택된 콤포넌트로부터 완료 상태들을 포함하는

데이터 처리 시스템.

**청구항 7**

제6항에 있어서, 상기 메시지 상태들은 입/출력 상태 버퍼 인덱스에 의해서 번호가 매겨지는(numbered)

데이터 처리 시스템.

**청구항 8**

제6항에 있어서, 상기 완료 상태는 입/출력 상태 버퍼 인덱스에 의해서 번호가 매겨지는(numbered)

데이터 처리 시스템.

**청구항 9**

제1항에 있어서, 상기 방법은 메시지 상태들을 상기 선택된 콤포넌트에 결합된 입/출력 버스 컨트롤러로부터 수집하는 단계(collecting)를 더 포함하고, 상기 메시지 상태들은 상기 선택된 콤포넌트로부터 완료 상태들을 포함하는

데이터 처리 시스템.

**청구항 10**

제1항에 있어서, 상기 데이터 처리 유닛은 비동기 코어-네스트 인터페이스(an asynchronous core-nest interface)를 포함하고, 상기 비동기 코어-네스트 인터페이스는 상기 선택된 콤포넌트로부터 리턴된 상태들을 수집하기 위해 상기 다수의 입/출력 상태 버퍼들을 포함하는 입/출력 상태 어레이를 포함하며, 상기 선택된 콤포넌트는 애그리게이션 버퍼(an aggregation buffer)를 통해 상기 데이터 처리 유닛에 통신 가능하게 결합된 시스템 네스트(a system nest)이고, 상기 애그리게이션 버퍼는 상기 비동기 코어-네스트 인터페이스(an asynchronous core-nest interface)에 통신 가능하게 더 결합되는

데이터 처리 시스템.

**청구항 11**

제10항에 있어서, 상기 시스템 펌웨어는 상기 입/출력 저장 명령을 처리하기 위한 비동기 입/출력 드라이버 코드와 입/출력 상태 버퍼들을 상기 입/출력 상태 어레이에 할당 및 할당해제 하고 그리고 새로운 저장 명령의 시작을 개시하는 어레이 관리 로직을 포함하는

데이터 처리 시스템.

**청구항 12**

제11항에 있어서, 상기 코어는 상기 비동기 입/출력 드라이버 코드의 상태 정보를 위한 메모리 요건들을 처리하기 위해 비동기 셋업 코드를 포함하는

데이터 처리 시스템.

**청구항 13**

제10항에 있어서, 상기 비동기 코어-네스트 인터페이스는 로컬 완료된 데이터(data with local completion)를 포워드 하기 위해 비동기 코어-네스트 인터페이스 포워딩 컴포넌트(an asynchronous core-nest interface forwarding component)를 포함하는

데이터 처리 시스템.

**청구항 14**

제10항에 있어서, 상기 애그리게이션 버퍼는 요청을 보낸 후(after sending a request) 프리 포 리유스 메시지(a free for reuse message)를 전달하기 위해 조기 완료 로직(an early completion logic)을 포함하는

데이터 처리 시스템.

**청구항 15**

적어도 하나의 처리 회로에 의해 판독 가능하고 입/출력 저장 명령을 처리하기 위한 방법을 수행하기 위한 명령들을 저장하는 컴퓨터 판독가능 스토리지 매체에 있어서, 상기 방법은

상기 입/출력 저장 명령을 사용하여 명시된 주소에 의해서 입/출력 함수(an input/output function)를 식별하는 단계(identifying) - 상기 입/출력 저장 명령은 주소를 통한 오프셋(an offset through an address), 전송될 적어도 하나의 데이터 및 전송될 데이터에 대한 포인터, 및 상기 데이터의 길이로 적어도 상기 입/출력 함수를 명시함 -;

상기 입/출력 함수에 대한 액세스가 주소 공간 상에서 그리고 게스트 인스턴스 레벨 상에서 허용되는 지를 확인하는 단계(verifying)

데이터 처리 시스템의 데이터 처리 유닛에 의해서, 상기 입/출력 저장 명령을 완료하는 단계(completing) - 상기 입/출력 저장 명령을 완료하는 단계는 상기 데이터 처리 유닛과 다른 데이터 처리 시스템의 선택된 컴포넌트에서 상기 입/출력 저장 명령의 실행이 완료되기 전에 완료하며, 상기 선택된 컴포넌트는 비동기(asynchronously) 적어도 하나의 외부 디바이스로부터 데이터를 로드하고 적어도 하나의 외부 디바이스에 데이터를 저장함-;

실패한 비동기 실행의 데이터를 전송하는, 상기 데이터 처리 유닛에서 상기 입/출력 저장 명령의 비동기 실행 동안, 에러를 검출하는 것에 기초하여, 인터럽트를 통해 통지를 제공하는 단계(providing);

상기 입/출력 저장 명령의 완료 메시지(a completion message)를 수집하고(collect) 수신된 완료 메시지들에 기초하여 다수의 입/출력 상태 버퍼들 중 하나 또는 그 이상의 상태 버퍼들을 갱신하는 단계(updating); 그리고

상기 다수의 입/출력 상태 버퍼들 중 하나의 상태 버퍼가 미실행(pending) 입/출력 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 상기 입/출력 저장 명령의 실행을 지연하는 단계(delaying)를 포함하는

컴퓨터 판독가능 스토리지 매체.

**청구항 16**

제15항에 있어서,

상기 방법은, 상기 다수의 입/출력 상태 버퍼들 중 하나의 상태 버퍼가 하나 또는 그 이상의 미실행(pending) 입/출력 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 비동기 입/출력 저장 명령 및 동기 입/출력 저장 명령 중 적어도 하나의 실행을, 상기 데이터 처리 유닛에 의해서, 지연하는 단계를 더 포함하는

컴퓨터 판독가능 스토리지 매체.

**청구항 17**

제15항에 있어서, 상기 데이터 처리 유닛은 비동기 코어-네스트 인터페이스(an asynchronous core-nest interface)를 포함하고, 상기 비동기 코어-네스트 인터페이스는 상기 선택된 컴포넌트로부터 리턴된 상태들을 수집하기 위해 상기 다수의 입/출력 상태 버퍼들을 포함하는 입/출력 상태 어레이를 포함하며, 상기 선택된 컴포넌트는 애그리게이션 버퍼(an aggregation buffer)를 통해 상기 데이터 처리 유닛에 통신 가능하게 결합된 시

시스템 네스트(a system nest)이고, 상기 애그리게이션 버퍼는 상기 비동기 코어-네스트 인터페이스(an asynchronous core-nest interface)에 통신 가능하게 더 결합되는

컴퓨터 판독가능 스토리지 매체.

**청구항 18**

입/출력 저장 명령을 처리하기 위한 컴퓨터 구현 방법에서, 상기 컴퓨터 구현 방법은:

상기 입/출력 저장 명령을 사용하여 명시된 주소에 의해서 입/출력 함수(an input/output function)를 식별하는 단계(identifying) - 상기 입/출력 저장 명령은 주소를 통한 오프셋(an offset through an address), 전송될 적어도 하나의 데이터 및 전송될 데이터에 대한 포인터, 및 상기 데이터의 길이로 적어도 상기 입/출력 함수를 명시함 -;

상기 입/출력 함수에 대한 액세스가 주소 공간 상에서 그리고 게스트 인스턴스 레벨 상에서 허용되는 지를 확인하는 단계(verifyng);

데이터 처리 시스템의 데이터 처리 유닛에 의해서, 상기 입/출력 저장 명령을 완료하는 단계(completeing) - 상기 입/출력 저장 명령을 완료하는 단계는 상기 데이터 처리 유닛과 다른 데이터 처리 시스템의 선택된 컴포넌트에서 상기 입/출력 저장 명령의 실행이 완료되기 전에 완료하며, 상기 선택된 컴포넌트는 비동기(asynchronously) 적어도 하나의 외부 디바이스로부터 데이터를 로드하고 적어도 하나의 외부 디바이스에 데이터를 저장함-

실패한 비동기 실행의 데이터를 전송하는, 상기 데이터 처리 유닛에서 상기 입/출력 저장 명령의 비동기 실행 동안, 에러를 검출하는 것에 기초하여, 인터럽트를 통해 통지를 제공하는 단계(providing);

상기 입/출력 저장 명령의 완료 메시지(a completion message)를 수집하고(collect) 수신된 완료 메시지들에 기초하여 다수의 입/출력 상태 버퍼들 중 하나 또는 그 이상의 상태 버퍼들을 갱신하는 단계(updating); 그리고

상기 다수의 입/출력 상태 버퍼들 중 하나의 상태 버퍼가 미실행(pending) 입/출력 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 상기 입/출력 저장 명령의 실행을 지연하는 단계(delaying)를 포함하는

컴퓨터 구현 방법.

**청구항 19**

제18항에 있어서, 상기 방법은, 상기 다수의 입/출력 상태 버퍼들 중 하나의 상태 버퍼가 하나 또는 그 이상의 미실행(pending) 입/출력 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 비동기 입/출력 저장 명령 및 동기 입/출력 저장 명령 중 적어도 하나의 실행을, 상기 데이터 처리 유닛에 의해서, 지연하는 단계를 더 포함하는

컴퓨터 구현 방법.

**청구항 20**

제18항에 있어서, 상기 데이터 처리 유닛은 비동기 코어-네스트 인터페이스(an asynchronous core-nest interface)를 포함하고, 상기 비동기 코어-네스트 인터페이스는 상기 선택된 컴포넌트로부터 리턴된 상태들을 수집하기 위해 상기 다수의 입/출력 상태 버퍼들을 포함하는 입/출력 상태 어레이를 포함하며, 상기 선택된 컴포넌트는 애그리게이션 버퍼(an aggregation buffer)를 통해 상기 데이터 처리 유닛에 통신 가능하게 결합된 시스템 네스트(a system nest)이고, 상기 애그리게이션 버퍼는 상기 비동기 코어-네스트 인터페이스(an asynchronous core-nest interface)에 통신 가능하게 더 결합되는

컴퓨터 구현 방법.

**청구항 21**

삭제

**청구항 22**

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

**발명의 설명**

**기술 분야**

[0001] [0001] 본 발명은 일반적으로 데이터 처리 시스템들에 관한 것으로서, 특히 다수의 외부 디바이스들에 대한 입/출력 저장 명령의 처리를 위한 방법, 컴퓨터 프로그램 제품 및 데이터 처리 시스템에 관한 것이다.

**배경 기술**

[0002] [0002] 컴퓨팅 환경은 다양한 유형의 어댑터들을 포함하는, 하나 또는 그 이상의 유형의 입/출력 디바이스들을 포함할 수 있다. 어댑터의 한 유형은 주변 콤포넨트 인터커넥트(PCI) 또는 주변 콤포넨트 인터커넥트 익스프레스(PCIe) 어댑터이다. 이 어댑터는 이 어댑터가 부착된 시스템과 이 어댑터 사이에 데이터를 통신하는데 사용되는 하나 또는 그 이상의 주소 공간들을 포함한다.

[0003] [0003] 일부 시스템들에서, 상기 어댑터와 결합된 중앙 처리 유닛(CPU)의 주소 공간의 일부는 상기 어댑터의 주소 공간 내의 데이터를 직접 조작하기 위해 스토리지를 액세스하는 CPU 명령들을 인에이블하는 상기 어댑터의 주소 공간으로 매핑된다(mapped).

[0004] [0004] PCI 또는 PCIe 어댑터들과 같은 어댑터들과의 통신은 어댑터들에게로 및 어댑터들로부터 데이터를 통신하기 위해 구체적으로 설계되고 통신을 위해 사용되는 제어 명령들에 의해서 용이하게 될 수 있다.

**발명의 내용**

**해결하려는 과제**

[0005] [0005] 종래 기술에서, 어댑터에 데이터를 저장하기 위한 저장 명령(a store instruction)은, 예를 들어, 실행을 위한 머신 명령을 획득하는 단계를 포함하며, 상기 머신 명령은 컴퓨터 아키텍처에 따른 컴퓨터 실행을 위해 정의되고, 상기 머신 명령은, 예를 들어, 어댑터로의 저장 명령(a store to adapter instruction)을 식별하는 오피코드 필드를 포함한다. 제1필드는 어댑터에 저장될 데이터를 포함하는 제1위치를 식별한다. 제2필드는 제2위치를 식별하는데, 상기 제2위치의 콘텐츠는, 상기 어댑터를 식별하는 함수 핸들(a function handle identifying the adapter), 데이터가 저장될 상기 어댑터 내의 주소 공간의 지정, 및 상기 주소 공간 내의 오프셋을 포함한다. 상기 머신 명령이 실행되면, 실행하는 단계는 상기 어댑터와 연관된 함수 테이블 엔트리(a function table entry)를 획득하기 위해 상기 함수 핸들을 사용하는 단계를 포함한다. 상기 어댑터의 데이터 주소는 상기 함수 테이블 엔트리 내의 정보 중 적어도 하나와 상기 오프셋을 사용하여 획득된다. 데이터는 상기 주소 공간의 지정에 의해서 식별되는 상기 주소 공간 내의 특정 위치의 제1위치로부터 저장되며, 상기 특정 위치는 상기 어댑터의 데이터 주소에 의해서 식별된다.

[0006] [0006] 대형 멀티-프로세서 시스템의 기존 특징은 타겟 존(a target zone) 내의 모든 프로세서들을 정지시키는(quiesce)능력이다. 정지 기능들(quiesce functions)은, 예를 들어, 시스템 업데이트들 또는 백업들을 수행하기 위한 하나의 프로세서 또는 프로세서들의 그룹의 상태를 일시적으로 중지 또는 변경하도록 작동한다. 일부의 예들에서, 정지 인터럽션(a quiesce interruption)은 시스템 자원들의 서브세트에 대해서만 적용 가능하다. 그러한 예들에서, 시스템은 다른 존들로 나누어질 수 있다. 하나의 존(타겟 존)에 적용 가능한 정지 연산에 대해, 상기 타겟 존 외부의 프로세서들은, 비록 새로운 변환들(translations)은 차단될 수 있을 지라도, 실행은 계속하도록 허용된다. 통상적으로, 적어도 하나의 시스템 컨트롤러 또는 다른 메커니즘이 시스템 내의 모든 물리적

프로세서들에 대해 정지를 방송하고, 정지 상태 정보를 수집하는 단계를 처리하며, 모든 프로세서들이 정지 요청을 시작하였거나, 또는 무시하고(필터링 하고) 있을 때, 요청 프로세서(a requesting processor)에 대해 표시한다.

[0007] 정지 컨트롤러(a quiesce controller)가 멀티-프로세서 시스템 내의 프로세서, 및 정지 요청을 수신하도록 구성된 정지 상태 머신(a quiesce state machine)에 통신 가능하게 결합될 수 있다. 상기 컴퓨터 시스템은 요청 프로세서로부터 상기 정지 컨트롤러에서 정지 요청을 수신하는 단계 -상기 요청 프로세서는 멀티-프로세서 시스템 내의 복수의 프로세서들 중 하나일 수 있음-, 및 상기 정지 요청이 상기 정지 상태 머신의 상태에 기초하여 수용되지 않음을 결정하는 단계를 포함하는 방법을 수행하도록 구성된다. 상기 방법은 또한, 상기 요청이 수용되지 않음에 기초하여, 상기 정지 요청이 거절되었다는 것을 표시하도록 구성된 거절 메시지를 생성하는 단계, 다른 정지 요청에 기초하는, 정지 커맨드(a quiesce command)가 상기 멀티-프로세서 시스템으로 방송(a quiesce command) 될 때까지 상기 거절 메시지를 홀드 하는 단계, 및 상기 정지 컨트롤러에 의해서 검출되는 상기 정지 커맨드의 방송에 기초하여 상기 요청 프로세서로 상기 거절 메시지를 보내는 단계를 포함한다.

**과제의 해결 수단**

[0008] 입/출력 저장 명령(an input/output store instruction)을 처리하기 위해 데이터 처리 시스템이 제안되고, 상기 데이터 처리 시스템은 입/출력 버스 컨트롤러에 의해서 적어도 하나의 입/출력 버스에 통신 가능하게 결합된 시스템 네스트(a system nest)를 포함한다. 상기 데이터 처리 시스템은 또한 코어(a core), 시스템 펌웨어(a system firmware) 및 비동기 코어-네스트 인터페이스(an asynchronous core-nest interface)를 포함하는 적어도 하나의 데이터 처리 유닛을 더 포함한다. 상기 데이터 처리 유닛은 애그리게이션 버퍼(an aggregation buffer)를 통해 상기 시스템 네스트에 통신 가능하게 결합된다. 상기 시스템 네스트는 상기 입/출력 버스에 통신 가능하게 결합된 적어도 하나의 외부 디바이스(at least one external device)로부터 데이터를 비동기로 로드하고(asynchronously load) 및/또는 상기 적어도 하나의 외부 디바이스(214)에 데이터를 비동기로 저장하도록(asynchronously store) 구성된다. 상기 비동기 코어-네스트 인터페이스는 다수의 입/출력 상태 버퍼들(multiple input/output status buffers)을 갖는 입/출력 상태 어레이(an input/output status array), 및 어레이 관리 및 액세스 로직(an array management and access logic)를 포함한다.

[0009] 상기 데이터 처리 시스템은 다음을 수행한다: (i) 주소를 통한 오프셋(an offset through an address), 전송될 데이터 및/또는 전송될 데이터에 대한 포인터, 및 상기 데이터의 길이로 적어도 하나의 입/출력 함수(an input/output function)를 명시하는 상기 입/출력 저장 명령을 상기 데이터 처리 시스템 상에서 실행되는(running) 운영 체제가 발행(issue); (ii) 상기 입/출력 저장 명령에 명시된 상기 주소에 의해서 상기 입/출력 함수를 식별하도록 상기 데이터 처리 유닛이 구성됨(configured); (iii) 상기 입/출력 함수에 대한 액세스가 주소 공간 상에서 그리고 게스트 인스턴스 레벨 상에서 허용되는 지를 확인하도록(verify) 상기 데이터 처리 유닛이 구성됨-상기 게스트는 상기 데이터 처리 시스템 상에서 실행됨-; (iv) 상기 시스템 네스트에서 상기 입/출력 저장 명령의 실행이 완료되기 전에 상기 입/출력 저장 명령을 완료하도록(complete) 상기 데이터 처리 유닛이 구성됨; (v) 상기 입/출력 저장 명령의 비동기 실행 동안, 상기 데이터 처리 유닛에 의해서 에러가 검출된다면, 상기 실패한 비동기 실행(the failed asynchronous execution)의 데이터를 전송하는, 인터럽트를 통해 상기 운영 체제에 통지하도록(notify) 상기 시스템 펌웨어가 구성됨; (vi) 상기 어레이 관리 및 액세스 로직이 상기 저장 명령의 완료 메시지(a completion message)를 수집하고( collect) 수신된 완료 메시지에 기초하여 상기 입/출력 상태 버퍼들을 갱신함(update); 및 (vii) 입/출력 상태 버퍼가 미실행 저장 명령들(pending store instructions)의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 상기 저장 명령의 실행을 상기 데이터 처리 유닛이 지연함(delay).

[0010] 따라서, 반복된 비동기 저장명령들의 명령 당 사이클들을 감소시키기 위해 다수의 미실행 비동기 저장 명령들(multiple outstanding asynchronous store instructions)이 동시에 허용될 수 있는 장점이 있다. 비동기 저장 명령들과 동기 로드/저장 명령들 사이에 순서(ordering)가 정의된다. 다수의 미실행 비동기 저장 명령들의 지원(supporting) 은 다수의 상태 메시지들의 북키퍼(book keeping) 및 상태 엔트리들과 응답들의 상관 관계(correlation)에 기초한다.

[0011] 본 발명의 제1 실시 예에 따른 데이터 처리 시스템은 입/출력 버스를 통해 상기 데이터 처리 시스템의 적어도 하나의 외부 디바이스로부터 로드하고 상기 적어도 하나의 외부 디바이스에 저장하는 명령들을 포함한다. 비동기 명령들은 데이터가 상기 외부 디바이스에 저장되기 전에 완료되고 한편 동기 명령들은 데이터가 상기 외부 디바이스에 저장되고 난 후에 완료된다. 여기서 기술하는 실시 예들에서는, PCI가 모든 다른 입/

출력 기술에 대해서 교환 가능하게 사용될 것이고, 본 발명의 실시 예를 PCI로 한정하지 않을 것이다.

- [0012] [0012] 본 발명의 실시 예들은 상기 아키텍처 경계 위에서 관측할 수 있는 엄격한 순서로(in a strictly ordered way) 입/출력 저장 명령 실행을 기술하지만 실제 실행은 데이터 처리 유닛(CPU)의 하드웨어 내의 순서를 벗어날 수 있다.
- [0013] [0013] 본 발명의 실시 예들에 따라, PCI 저장 명령이 PCIe 저장 효과(the PCIe store effect)의 비동기 실행 및 비동기 상태 처리로 실행될 수 있다. 비동기 안정적 실행(asynchronous reliable execution)은 본 발명의 데이터 처리 시스템의 마이크로아키텍처 내 안정적 포워딩 메커니즘들(reliable forwarding mechanisms)에 기초한다.
- [0014] [0014] 기존의 PCI 저장 및 저장 블록 명령(An existing PCI store and store block instruction)은 PCI 저장 데이터가 PCIe 인터페이스로 전달되고 완료가 처리 유닛으로 리턴되는 지점까지는 통상적으로 동기(synchronous)이다.
- [0015] [0015] PCI표준은, 비동기 센드-아웃(asynchronous send-out)으로 데이터를 집계하는 프로세서 내의 저장 큐(a store queue)를 통해 통상적으로 구현되는, PCI정보의 비동기 센드 커맨드(an asynchronous send command)만을 오직 요구한다.
- [0016] [0016] 바람직하게도, 본 발명의 실시 예들에 따라, 명령 당 사이클들에 관한 개선이 동기 PCI명령을 입/출력 저장 명령의 안정적 비동기 센드 프로세스(a reliable asynchronous send process)에 의해서 대체함으로써 달성될 수 있다.
- [0017] [0017] 전송될 데이터에 관해서 대체적으로 또는 추가적으로, 본 발명의 실시 예들에 따른 저장 명령은 또한, 데이터를 직접적으로 보유하는 대신에, 데이터를 메인 메모리로부터 폐지하기 위해 사용되는 메인 메모리에 대한 포인터를 명시할 수 있다.
- [0018] [0018] 게스트 인스턴스 레벨(Guest instance level)은 또한 단일 게스트 또는 호스트 상의 데이터 처리 시스템 상에 실행할 수 있음을 의미한다.
- [0019] [0019] 입/출력 함수 그 자체의 오프셋의 주소는 가상의, 물리적, 논리적 주소일 수 있다. 가상 및 논리적 주소들은 통상적으로 메모리 관리 유닛(MMU)를 통해서 물리적 주소로 변환되고, 물리적 주소는 어느 함수 및 오프셋을 의미하는지를 식별한다.
- [0020] [0020] 이러한 맥락에서 물리적 주소는 "게스트/운영 체제 내에서 액세스 가능한 주소 변환 계층에서 최하위 주소(lowest address)"를 의미한다.
- [0021] [0021] 바람직하게도, 상기 입/출력 상태 버퍼들은 상기 시스템 네스트 및/또는 상기 입/출력 버스 컨트롤러로부터 리턴된 상태들, 특히 상기 시스템 네스트로부터 완료 메시지를 수집할 수 있다. 이들 입/출력 상태 버퍼들은 상기 리턴된 상태들을 수집할 수 있는데, 이는 비동기 전송 프로세스를 지원하는 비동기 시스템 메시지 버퍼로서 동작하는 것이다. 바람직하게도, 상기 입/출력 상태 버퍼들은 신속한 응답을 위해 상기 비동기 코어-네스트 인터페이스 내 직접적으로 집적될 수 있다.
- [0022] [0022] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 데이터 처리 유닛은, 입/출력 상태 버퍼가 미실행 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 비동기 및/또는 동기 저장 명령들의 실행을 지연할 수 있다. 따라서, 다수의 입/출력 디바이스들에 대한 다수의 저장 명령들의 순서 있는 프로세스(an ordering process of a multiple of store instructions)는 효율적인 방식으로 처리될 수 있고, 이는 처리 시간을 절약하는 능력을 제공한다.
- [0023] [0023] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 입/출력 상태 버퍼들은 상기 시스템 네스트로부터 및/또는 상기 입/출력 버스 컨트롤러로부터 메시지 상태들을, 특히 상기 시스템 네스트로부터 완료 상태를 수집할 수 있다. 이 방식에 의해서 다른 저장 명령들의 완료 상태에 관한 정보가 질서 있고 효율적인 방식으로(in an ordered and efficient manner) 처리될 수 있다.
- [0024] [0024] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 메시지 상태들 및/또는 상기 완료 상태는 입/출력 상태 버퍼 인덱스에 의해서 번호가 매겨질(numbered) 수 있다. 상기 번호 매김은 다른 저장 명령들을 추가로 처리하기 위해 질서 있고 효율적인 방식으로 메시지들, 특히 완료 상태들의 처리 가능성을 인에이블한다.

- [0025] [0025] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 애그리게이션 버퍼는 비동기 버스를 통해 상기 비동기 코어-네스트 인터페이스에 통신 가능하게 결합될 수 있다. 따라서, 상기 외부 디바이스로 전송될 모든 데이터가 상기 애그리게이션 버퍼에 저장될 때까지 상기 애그리게이션 버퍼는 상기 비동기 코어-네스트 인터페이스에 의해서 직접적으로 보내진 데이터를 연속적으로(consecutively) 처리할 수 있다. 이 방식에 의해서 상기 비동기 코어-네스트 인터페이스로부터 데이터 전송을 위한 비동기 전송 메커니즘은 바람직하게 지원될 수 있다.
- [0026] [0026] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 만일 상기 소스 데이터의 길이가 8바이트를 초과한다면, 상기 데이터는 상기 애그리게이션 버퍼에 다수의 데이터 패킷들로 조기 완료 메시지를 갖는 비동기 전송 메커니즘을 통해 상기 입/출력 저장 명령에 의해서 전송될 수 있고, 그렇지 않다면, 상기 데이터는 하나의 데이터 패킷으로 전송될 수 있다. 상기 비동기 전송 메커니즘은 상기 센딩 디바이스가 이전의 상태(an earlier state)에서 리유스에 대해 프리(free for reuse)이기 때문에 바람직하다.
- [0027] [0027] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 시스템 펌웨어는 상기 입/출력 저장 명령을 처리하기 위한 비동기 입/출력 드라이버 코드를 포함할 수 있다. 따라서, 비동기 전송 메커니즘은 상기 데이터 처리 유닛으로부터 데이터를 상기 외부 디바이스로 전송하기 위해 사용될 수 있다.
- [0028] [0028] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 코어는 상기 비동기 입/출력 드라이버 코드의 상태 정보를 위한 메모리 요건들을 처리하기 위해 비동기 셋업 코드를 포함할 수 있다. 이 비동기 셋업 코드는 또한 상기 시스템 네스트 및 상기 입/출력 버스 컨트롤러에 대해 상기 애그리게이션 버퍼를 통한 비동기 전송 메커니즘을 용이하게 할 수 있다.
- [0029] [0029] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 비동기 코어-네스트 인터페이스는 로컬 완료된 데이터(the data with local completion)를 포워드 하기 위해 상기 비동기 코어-네스트 인터페이스 포워딩 컴포넌트(an asynchronous core-nest interface forwarding component)를 포함할 수 있다. 이 컴포넌트는 상기 비동기 코어-네스트 인터페이스 내의 하드웨어로 구현될 수 있다. 따라서, 상기 애그리게이션 버퍼로 데이터 패킷들의 데이터를 보내기 위한 바람직한 비동기 전송 모드가 지원될 수 있다.
- [0030] [0030] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 애그리게이션 버퍼는 요청을 보낸 후(after sending a request) 프리 포 리유스 메시지(a free for reuse message)를 전달하기 위해 조기 완료 로직(an early completion logic)을 포함할 수 있다. 이는 상기 시스템 네스트 및 상기 입/출력 버스 컨트롤러에 대한 상기 애그리게이션 버퍼를 통한 데이터의 전송 프로세스의 조기 계속(an early continuation)을 인에이블한다.
- [0031] [0031] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 상기 시스템 펌웨어는, 상기 입/출력 상태 버퍼들을 상기 입/출력 상태 어레이에 할당/할당해제 하고 및/또는 새로운 저장 명령의 시작을 개시하는, 어레이 관리 로직을 포함할 수 있다. 따라서, 아이들 상태 버퍼들이 추가의 저장 명령들에 기인될 수 있다(be attributed to). 저장 명령들의 질서 있는 처리가 효율적이고 시간절약 방식으로 처리될 수 있다.
- [0032] [0032] 본 발명의 데이터 처리 시스템의 바람직한 실시 예에 따라, 시스템 메시지는 -계층 물리 타겟 주소(a hierarchical physical target address), -SMT 스레드 소싱 또는 애그리게이트 버퍼 식별자(sourcing an SMT thread or an aggregate buffer identifier), -데이터의 길이, -입/출력 버스 주소, -입/출력 상태 버퍼 인덱스 중 하나를 포함할 수 있다. 따라서, 상기 데이터 처리 시스템을 통해 관련 정보의 바람직한 패싱(passing)이 보장될 수 있다.
- [0033] [0033] 또한, 데이터 처리 시스템의 적어도 하나의 외부 디바이스에 대해서 입/출력 저장 명령을 처리하기 위한 방법이 제안되고, 상기 데이터 처리 시스템은 입/출력 버스 컨트롤러에 의해서 적어도 하나의 입/출력 버스에 통신 가능하게 결합된 시스템 네스트를 포함한다. 상기 데이터 처리 시스템은 또한, 코어, 시스템 펌웨어 및 비동기 코어-네스트 인터페이스를 포함하는 적어도 하나의 데이터 처리 유닛을 포함한다. 상기 데이터 처리 유닛은 애그리게이션 버퍼를 통해 상기 시스템 네스트에 통신 가능하게 결합된다. 상기 외부 디바이스는 상기 입/출력 버스에 통신 가능하게 결합된다. 상기 비동기 코어-네스트 인터페이스는, 어레이 관리 및 액세스 로직뿐만 아니라, 다수의 입/출력 상태 버퍼들을 갖는 입/출력 상태 어레이를 포함한다.
- [0034] [0034] 상기 방법은: (i) 주소를 통한 오프셋, 전송될 데이터 및/또는 전송될 데이터에 대한 포인터, 및 상기 데이터의 길이로 적어도 하나의 입/출력 함수를 명시하는 상기 입/출력 저장 명령을 상기 데이터 처리 시스템 상에서 실행되는 운영 체제가 발행하는 단계; (ii) 상기 입/출력 저장 명령에 명시된 상기 주소에 의해서 상기

입/출력 함수를 식별하도록 상기 데이터 처리 유닛이 구성되는 단계; (iii) 상기 입/출력 함수에 대한 액세스가 주소 공간 상에서 그리고 게스트 인스턴스 레벨 상에서 허용되는 지를 확인하도록 상기 데이터 처리 유닛이 구성되는 단계-상기 게스트는 상기 데이터 처리 시스템 상에서 실행됨-; (iv) 상기 시스템 네스트에서 상기 입/출력 저장 명령의 실행이 완료되기 전에 상기 입/출력 저장 명령을 완료하도록 상기 데이터 처리 유닛이 구성되는 단계; (v) 상기 입/출력 저장 명령의 비동기 실행 동안, 상기 데이터 처리 유닛에 의해서 에러가 검출된다면, 상기 실패한 비동기 실행의 데이터를 전송하는, 인터럽트를 통해 상기 운영 체제에 통지하도록 상기 시스템 펌웨어가 구성되는 단계; (vi) 상기 어레이 관리 및 액세스 로직이 상기 저장 명령의 완료 메시지를 수집하고 수신된 완료 메시지에 기초하여 상기 입/출력 상태 버퍼들을 갱신하는 단계; 및 (vii) 입/출력 상태 버퍼가 미실행 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 상기 저장 명령의 실행을 상기 데이터 처리 유닛이 지연하는 단계를 포함한다.

- [0035] [0035] 따라서, 반복된 비동기 저장명령들의 명령 당 사이클들을 감소시키기 위해 다수의 미실행 비동기 저장 명령들(multiple outstanding asynchronous store instructions)이 동시에 허용될 수 있는 장점이 있다. 비동기 저장 명령들과 동기 로드/저장 명령들 사이에 순서(ordering)가 정의된다. 다수의 미실행 비동기 저장 명령들의 지원(supporting)은 다수의 상태 메시지들의 북키퍼링(book keeping) 및 상태 엔트리들과 응답들의 상관 관계(correlation)에 기초한다.
- [0036] [0036] 본 발명의 추가의 실시 예에 따른 방법은 입/출력 버스를 통해 상기 데이터 처리 시스템의 적어도 하나의 외부 디바이스로부터 로드하고 상기 적어도 하나의 외부 디바이스에 저장하는 명령들을 포함한다. 비동기 명령들은 데이터가 상기 외부 디바이스에 저장되기 전에 완료되고 한편 동기 명령들은 데이터가 상기 외부 디바이스에 저장되고 난 후에 완료된다. 여기서 기술하는 실시 예들 내에서는, PCI가 모든 다른 입/출력 기술에 대해서 교환 가능하게 사용될 것이고, 본 발명의 실시 예를 PCI로 한정하지 않을 것이다.
- [0037] [0037] 본 발명의 방법의 실시 예들은 상기 아키텍처 경계 위에서 관측할 수 있는 엄격한 순서로(in a strictly ordered way) 입/출력 저장 명령 실행을 기술하지만 실제 실행은 데이터 처리 유닛(CPU)의 하드웨어 내의 순서를 벗어날 수 있다.
- [0038] [0038] 본 발명의 방법의 실시 예들에 따라, PCI 저장 명령이 PCIe 저장 효과(the PCIe store effect)의 비동기 실행 및 비동기 상태 처리로 실행될 수 있다. 비동기 안정적 실행(asynchronous reliable execution)은 본 발명의 데이터 처리 시스템의 마이크로아키텍처 내 안정적 포워딩 메커니즘들(reliable forwarding mechanisms)에 기초한다.
- [0039] [0039] 기존의 PCI 저장 및 저장 블록 명령(An existing PCI store and store block instruction)은 PCI 저장 데이터가 PCIe 인터페이스로 전달되고 완료가 처리 유닛으로 리턴되는 지점까지는 통상적으로 동기(synchronous)이다.
- [0040] [0040] PCI표준은, 비동기 센드-아웃(asynchronous send-out)으로 데이터를 집계하는 프로세서 내의 저장 큐(a store queue)를 통해 통상적으로 구현되는, PCI정보의 비동기 센드 커맨드(an asynchronous send command)만을 오직 요구한다.
- [0041] [0041] 바람직하게도, 본 발명의 방법의 실시 예들에 따라, 명령 당 사이클들에 관한 개선이 동기 PCI명령을 입/출력 저장 명령의 안정적 비동기 센드 프로세스(a reliable asynchronous send process)에 의해서 대체함으로써 달성될 수 있다.
- [0042] [0042] 전송될 데이터에 관해서 대체적으로 또는 추가적으로, 본 발명의 실시 예들에 따른 저장 명령은 또한, 데이터를 직접적으로 보유하는 대신에, 데이터를 메인 메모리로부터 페치 하기 위해 사용되는 메인 메모리에 대한 포인터를 명시할 수 있다.
- [0043] [0043] 게스트 인스턴스 레벨(Guest instance level)은 또한 단일 게스트 또는 호스트 상기 데이터 처리 시스템 상에 실행할 수 있음을 의미한다.
- [0044] [0044] 입/출력 함수 그 자체의 오프셋의 주소는 가상의, 물리적, 논리적 주소일 수 있다. 가상 및 논리적 주소들은 통상적으로 메모리 관리 유닛(MMU)를 통해서 물리적 주소로 변환되고, 그 다음 물리적 주소는 어느 함수 및 오프셋을 의미하는지를 식별한다.
- [0045] [0045] 이러한 맥락에서 물리적 주소는 "게스트/운영 체제 내에서 액세스 가능한 주소 변환 계층에서 최하위 주소(lowest address)"를 의미한다.

- [0046] [0046] 본 발명의 방법의 바람직한 실시 예에 따라, 입/출력 상태 버퍼가, 미실행 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 비동기 및/또는 동기 저장 명령들의 실행을 상기 데이터 처리 유닛(216)이 지연할 수 있다. 따라서, 다수의 입/출력 디바이스들에 대한 다수의 저장 명령들의 순서 있는 프로세스(an ordering process of a multiple of store instructions)는 효율적인 방식으로 처리될 수 있고, 이는 처리 시간을 절약하는 능력을 제공한다.
- [0047] [0047] 본 발명의 방법의 바람직한 실시 예에 따라, 상기 입/출력 상태 버퍼들은 상기 시스템 네스트로부터 및/또는 상기 입/출력 버스 컨트롤러로부터 메시지 상태들을, 특히 상기 시스템 네스트로부터 완료 상태를 수집할 수 있고, 상기 메시지 상태들 및/또는 상기 완료 상태는 입/출력 상태 버퍼 인덱스에 의해서 번호가 매겨질 (numbered) 수 있다. 이 방식에 의해서 다른 저장 명령들의 완료 상태에 관한 정보가 질서 있고 효율적인 방식으로(in an ordered and efficient manner) 처리될 수 있다. 상기 번호 매김은 다른 저장 명령들을 추가로 처리하기 위해 질서 있고 효율적인 방식으로 메시지들, 특히 완료 상태들의 처리 가능성을 인에이블 한다.
- [0048] [0048] 본 발명의 방법의 바람직한 실시 예에 따라, 상기 시스템 펌웨어는, 상기 입/출력 상태 버퍼들을 상기 입/출력 상태 어레이에서 할당/할당해제 하고 및/또는 새로운 저장 명령의 시작을 개시하는, 어레이 관리 로직을 포함할 수 있다. 따라서, 아이들 상태 버퍼들이 추가의 저장 명령들에 기인될 수 있다(be attributed to). 저장 명령들의 질서 있는 처리가 효율적이고 시간절약 방식으로 처리될 수 있다.
- [0049] [0049] 본 발명의 방법의 바람직한 실시 예에 따라, 상기 방법은: (i) 상기 입/출력 저장 명령을 상기 운영 체제가 발행하는 단계; (ii) 프리 입/출력 상태 버퍼 인덱스를 상기 시스템 펌웨어가 할당하는 단계; 이용 가능한 프리 입/출력 상태 버퍼 인덱스가 없다면 프리 입/출력 상태 버퍼 인덱스를 대기하는 단계; (iii) 상기 저장 명령을 비동기 센드 엔진(asynchronous send engine)에 상기 시스템 펌웨어가 주입하는 단계; 다른 저장 명령에 의해서 차단되었다면 상기 저장 명령이 완료될 때까지 대기하는 단계; (iv) 데이터의 길이에 따라 달라지는 단계로서, 데이터의 길이가 8바이트를 초과한다면 저장 블록(store block)의 모든 데이터가 상기 애그리게이션 버퍼로 포워드 될 때까지 상기 애그리게이션 버퍼로 데이터 패킷을 보내도록 상기 시스템 펌웨어가 반복적으로 시스템 메시지를 발행하는 단계; 상기 시스템 메시지에 의해서 상기 데이터가 보내질 때까지는 상기 시스템 펌웨어가 대기하는 단계; 그 다음에 상기 애그리게이션 버퍼로 상기 데이터를 보내도록 시스템 메시지를 발행하는 단계; (v) 상기 애그리게이션 버퍼가 완료 메시지를 보내는 것을 대기하는 동안, 상기 데이터를 비동기로 단일 네스트 메시지로써 상기 입/출력 버스 컨트롤러로 포워드하도록 상기 시스템 펌웨어가 메시지를 상기 애그리게이션 버퍼로 발행하는 단계; (vi) 상기 애그리게이션 버퍼- 상기 애그리게이션 버퍼는 센드 연산 직후(right after send operation) 리유스에 대해 프리임-가, 상기 시스템 펌웨어로 신호하는(signaling back), 상기 네스트 메시지를 상기 시스템 네스트로 주입하는 단계; 그 다음 상기 애그리게이션 버퍼가 프리 포 리유스 메시지(a free for reuse message)를 보내는 단계; (vii) 상기 시스템 네스트가 상기 메시지를 상기 타겟 위치로 포워드하는 단계; (viii) 상기 입/출력 버스 컨트롤러가 상기 메시지를 수신하고 데이터 프레임 내의 데이터를 상기 입/출력 버스로 포워드 하는 단계; (ix) 상기 입/출력 버스 컨트롤러가 완료 메시지를 상기 시스템 네스트로 보내는 단계; (x) 상기 시스템 네스트가 상기 완료 메시지를 상기 원래 애그리게이션 버퍼로 포워드하는 단계; (xi) 상기 애그리게이션 버퍼가 완료 메시지를 상기 비동기 코어-네스트 인터페이스로 포워드하는 단계; (xii) 상기 비동기 코어-네스트 인터페이스가 상기 완료 상태를 상기 입/출력 상태 버퍼 인덱스를 위한 상기 입/출력 상태 버퍼에 저장하고 연산의 완료를 상기 시스템 펌웨어에 신호하는 단계; (xiii) 상기 시스템 펌웨어가 상기 입/출력 상태 버퍼 인덱스에 의해서 트래킹 하는 입/출력 상태 버퍼를 갱신하는 단계; (xiv) 에러가 있는 경우에 상기 시스템 펌웨어가 비동기로 결합들을 운영 체제로 신호하는 단계를 더 포함할 수 있다.
- [0050] [0050] 오직 단계(ii)만 데이터의 길이에 종속되고(dependent) 8바이트를 초과하는 데이터의 길이에 대해서 및 8바이트를 초과하지 않는 데이터의 길이에 대해서는 서로 다르다(different).
- [0051] [0051] 본 발명의 방법의 실시 예에 따라, 상기 데이터는, 저장 블록(store block)의 모든 데이터가 상기 애그리게이션 버퍼로 포워드 될 때까지, 상기 애그리게이션 버퍼로 슬라이스들로 전송되고, 상기 시스템 펌웨어는 상기 데이터가 상기 비동기 코어-네스트 인터페이스에 의해서 보내질 때까지 대기한다.
- [0052] [0052] 따라서, 만일 데이터가 8 바이트 보다 적다면 슬라이스들 내의 상기 애그리게이션 버퍼를 데이터 패킷들로 채우는 프로세스는 생략될 수 있고 상기 외부 디바이스에 대한 데이터의 전송 프로세스는 단일의 단계로 완료될 수 있다.
- [0053] [0053] 본 발명의 방법의 바람직한 실시 예에 따라, 만일 상기 데이터의 길이가 8바이트를 초과한다면, 상기 데이터는 상기 애그리게이션 버퍼에 다수의 데이터 패킷들로 조기 완료 메시지를 갖는 비동기 전송 메커니즘을 통

해 상기 입/출력 저장 명령에 의해서 전송될 수 있다. 상기 비동기 전송 메커니즘은 상기 센딩 디바이스(the sending device)가 이전의 상태(an earlier state)에서 리유스에 대해 프리(free for reuse)이기 때문에 바람직하다.

- [0054] [0054] 본 발명의 방법의 바람직한 실시 예에 따라, 상기 시스템 펌웨어는, 상기 입/출력 저장 명령을 처리하기 위한 비동기 입/출력 드라이버 코드를 포함할 수 있다. 따라서, 비동기 전송 메커니즘은 상기 데이터 처리 유닛으로부터 데이터를 상기 외부 디바이스로 전송하기 위해 사용될 수 있다.
- [0055] [0055] 본 발명의 방법의 바람직한 실시 예에 따라, 상기 코어는 상기 비동기 입/출력 드라이버 코드의 상태 정보를 위한 메모리 요건들을 처리하기 위해 비동기 셋업 코드를 포함할 수 있다. 이 비동기 셋업 코드는 또한 상기 시스템 네스트 및 상기 입/출력 버스 컨트롤러에 대해 상기 애그리게이션 버퍼를 통한 비동기 전송 메커니즘을 용이하게 할 수 있다.
- [0056] [0056] 바람직하게도, 상기 비동기 코어-네스트 인터페이스는 로컬 완료된 데이터(the data with local completion)를 포워드 하기 위해 상기 비동기 코어-네스트 인터페이스 포워딩 컴포넌트(an asynchronous core- nest interface forwarding component)를 포함할 수 있다. 따라서, 상기 애그리게이션 버퍼로 데이터 패킷들의 데이터를 보내기 위한 바람직한 비동기 전송 모드가 지원될 수 있다.
- [0057] [0057] 바람직하게도, 상기 애그리게이션 버퍼는 요청을 보낸 후(after sending a request) 프리 포 리유스 메시지(a free for reuse message)를 전달하기 위해 조기 완료 로직(an early completion logic)을 포함할 수 있다. 이는 상기 시스템 네스트 및 상기 입/출력 버스 컨트롤러에 대한 상기 애그리게이션 버퍼를 통한 데이터의 전송 프로세스의 조기 계속(an early continuation)을 인에이블 한다.
- [0058] [0058] 바람직하게도, 상기 입/출력 상태 버퍼들은 상기 시스템 네스트로부터 및/또는 상기 입/출력 버스 컨트롤러로부터 리턴된 상태들을, 특히 상기 시스템 네스트로부터 완료 메시지를 수집할 수 있다. 이들 입/출력 상태 버퍼들은 상기 비동기 전송 프로세스를 지원하는 비동기 시스템 메시지 버퍼로서 작용하는 리턴된 상태들을 수집할 수 있다.
- [0059] [0059] 본 발명의 방법의 바람직한 실시 예에 따라, 시스템 메시지는 -계층 물리 타겟 주소(a hierarchical physical target address), -SMT 스레드 소싱 또는 애그리게이트 버퍼 식별자(sourcing an SMT thread or an aggregate buffer identifier), -데이터의 길이, -입/출력 버스 주소, -입/출력 상태 버퍼 인덱스 중 하나를 포함할 수 있다. 따라서, 상기 데이터 처리 시스템을 통해 관련 정보의 바람직한 패싱(passing)이 보장될 수 있다.
- [0060] [0060] 또한, 데이터 처리 시스템의 적어도 하나의 외부 디바이스에 대해서 입/출력 저장 명령을 처리하기 위한 컴퓨터 프로그램 제품이 제안되고, 상기 데이터 처리 시스템은 입/출력 버스 컨트롤러에 의해서 적어도 하나의 입/출력 버스에 통신 가능하게 결합된 시스템 네스트를 포함한다. 상기 데이터 처리 시스템은 또한, 코어, 시스템 펌웨어 및 비동기 코어-네스트 인터페이스를 포함하는 적어도 하나의 데이터 처리 유닛을 포함한다. 상기 데이터 처리 유닛은 애그리게이션 버퍼를 통해 상기 시스템 네스트에 통신 가능하게 결합된다. 상기 외부 디바이스는 상기 입/출력 버스에 통신 가능하게 결합된다. 상기 비동기 코어-네스트 인터페이스는 다수의 입/출력 상태 버퍼들을 갖는 입/출력 상태 어레이, 그리고 어레이 관리 및 액세스 로직을 포함한다.
- [0061] [0061] 상기 컴퓨터 프로그램 제품은 프로그램 명령들이 구현된 컴퓨터 판독가능 스토리지 매체를 포함하며, 상기 프로그램 명령들은 컴퓨터 시스템이 방법을 수행하도록 상기 컴퓨터 시스템에 의해서 실행가능 하며, 상기 방법은: (i) 주소를 통한 오프셋, 전송될 데이터 및/또는 전송될 데이터에 대한 포인터, 및 상기 데이터의 길이로 적어도 하나의 입/출력 함수를 명시하는 상기 입/출력 저장 명령을 상기 데이터 처리 시스템 상에서 실행되는 운영 체제가 발행하는 단계; (ii) 상기 입/출력 저장 명령에 명시된 상기 주소에 의해서 상기 입/출력 함수를 식별하도록 상기 데이터 처리 유닛이 구성되는 단계; (iii) 상기 입/출력 함수에 대한 액세스가 주소 공간 상에서 그리고 게스트 인스턴스 레벨 상에서 허용되는 지를 확인하도록 상기 데이터 처리 유닛이 구성되는 단계 -상기 게스트는 상기 데이터 처리 시스템 상에서 실행됨-; (iv) 상기 시스템 네스트에서 상기 입/출력 저장 명령의 실행이 완료되기 전에 상기 입/출력 저장 명령을 완료하도록 상기 데이터 처리 유닛이 구성되는 단계; (v) 상기 입/출력 저장 명령의 비동기 실행 동안, 상기 데이터 처리 유닛에 의해서 예러가 검출된다면, 상기 실패한 비동기 실행의 데이터를 전송하는, 인터럽트를 통해 상기 운영 체제에 통지하도록 상기 시스템 펌웨어가 구성되는 단계; (vi) 상기 어레이 관리 및 액세스 로직이 상기 저장 명령의 완료 메시지를 수집하고 수신된 완료 메시지들에 기초하여 상기 입/출력 상태 버퍼들을 갱신하는 단계; 및 (vii) 입/출력 상태 버퍼가 미실행 저장 명령

들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 상기 저장 명령의 실행을 상기 데이터 처리 유닛이 지연하는 단계를 포함한다.

[0062] 또한, 전술 한 방법을 수행하기 위한 컴퓨터 판독가능 프로그램 명령들을 포함하는 데이터 처리 프로그램의 실행을 위한 데이터 처리 시스템이 제안된다.

[0063] 본 발명과 함께 전술 한 발명의 목적 및 장점들은 아래에 상세하게 기술한 실시 예 들로부터 가장 잘 이해될 수 있지만, 실시 예들로 한정되는 것은 아니다.

**도면의 간단한 설명**

[0064] 도 1은, 본 발명의 한 실시 예에 따라, 적어도 하나의 외부 디바이스에 대해서 입/출력 저장 명령을 처리하기 위한 데이터 처리 시스템의 블록도를 묘사한다.

[0065] 도 2는, 본 발명의 한 실시 예에 따라, 적어도 하나의 외부 디바이스에 대해서 입/출력 저장 명령을 처리하기 위한 방법의 메시지 시퀀스 차트를 묘사한다.

[0066] 도 3은, 본 발명의 한 실시 예에 따라, 적어도 하나의 외부 디바이스에 대해서 입/출력 저장 명령을 처리하기 위한 플로 차트의 제1부분을 묘사한다.

[0067] 도 4는, 본 발명의 한 실시 예에 따라, 적어도 하나의 외부 디바이스에 대해서 입/출력 저장 명령을 처리하기 위한 플로 차트의 제2부분을 묘사한다.

[0068] 도 5는, 본 발명에 따른 방법을 실행하기 위한 데이터 처리 시스템의 예시적 실시 예를 묘사한다.

**발명을 실시하기 위한 구체적인 내용**

[0065] 도면들에서, 동일 구성 요소들은 동일 참조번호들로 참조된다. 상기 도면들은 단지 개념적 표현들일 뿐이고, 본 발명의 구체적 파라미터들을 나타내고자 하는 것은 아니다. 더 나아가, 상기 도면들은 본 발명의 전형적인 실시 예만을 묘사하려고 했던 것이고, 그러므로 본 발명의 범위를 제한하려고 한 것은 아니다.

[0066] 여기서 기술하는 보여주기 실시 예들은 입/출력 저장 명령을 처리하기 위한 데이터 처리 시스템 및 방법이 제공되고, 상기 데이터 처리 시스템은 입/출력 버스 컨트롤러에 의해서 적어도 하나의 입/출력 버스에 통신 가능하게 결합된 시스템 네스트를 포함한다. 상기 데이터 처리 시스템은 또한, 코어, 시스템 펌웨어 및 비동기 코어-네스트 인터페이스를 포함하는 적어도 하나의 데이터 처리 유닛을 포함한다. 상기 데이터 처리 유닛은 애그리게이션 버퍼를 통해 상기 시스템 네스트에 통신 가능하게 결합된다. 상기 시스템 네스트는 상기 입/출력 버스에 통신 가능하게 결합된 적어도 하나의 외부 디바이스로부터 데이터를 비동기로 로드하고 및/또는 상기 적어도 하나의 외부 디바이스에 데이터를 비동기로 저장하도록 구성된다. 상기 비동기 코어-네스트 인터페이스는 다수의 입/출력 상태 버퍼들을 갖는 입/출력 상태 어레이, 그리고 어레이 관리 및 액세스 로직을 포함한다.

[0067] 상기 보여주기 실시 예들은 상기 방법을 위해 사용될 수 있고, 상기 방법은: (i) 주소를 통한 오프셋, 전송될 데이터 및/또는 전송될 데이터에 대한 포인터, 및 상기 데이터의 길이로 적어도 하나의 입/출력 함수를 명시하는 상기 입/출력 저장 명령을 상기 데이터 처리 시스템 상에서 실행되는 운영 체제가 발행하는 단계; (ii) 상기 입/출력 저장 명령에 명시된 상기 주소에 의해서 상기 입/출력 함수를 식별하도록 상기 데이터 처리 유닛이 구성되는 단계; (iii) 상기 입/출력 함수에 대한 액세스가 주소 공간 상에서 그리고 게스트 인스턴스 레벨 상에서 허용되는 지를 확인하도록 상기 데이터 처리 유닛이 구성되는 단계-상기 게스트는 상기 데이터 처리 시스템 상에서 실행됨-; (iv) 상기 시스템 네스트에서 상기 입/출력 저장 명령의 실행이 완료되기 전에 상기 입/출력 저장 명령을 완료하도록 상기 데이터 처리 유닛이 구성되는 단계; (v) 상기 입/출력 저장 명령의 비동기 실행 동안, 상기 데이터 처리 유닛에 의해서 예러가 검출된다면, 상기 실패한 비동기 실행의 데이터를 전송하는, 인터럽트를 통해 상기 운영 체제에 통지하도록 상기 시스템 펌웨어가 구성되는 단계; (vi) 상기 어레이 관리 및 액세스 로직이 상기 저장 명령의 완료 메시지를 수집하고 수신된 완료 메시지에 기초하여 상기 입/출력 상태 버퍼들을 갱신하는 단계; 및 (vii) 입/출력 상태 버퍼가 미실행 저장 명령들의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 상기 저장 명령의 실행을 상기 데이터 처리 유닛이 지연하는 단계를 포함한다.

[0068] 전송될 데이터에 관해서 대체적으로 또는 추가적으로, 본 발명의 일 실시 예들에 따른 저장 명령은 또한, 데이터를 직접적으로 보유하는 대신에, 데이터를 메인 메모리로부터 폐치하기 위해 사용되는 메인 메모리

에 대한 포인터를 명시할 수 있다.

- [0069] [0073] 게스트 인스턴스 레벨(Guest instance level)은 또한 단일 게스트 또는 호스트 상의 데이터 처리 시스템 상에 실행할 수 있음을 의미할 수 있다.
- [0070] [0074] 입/출력 함수 그 자체의 오프셋의 주소는 가상의, 물리적, 논리적 주소일 수 있다. 가상 및 논리적 주소들은 통상적으로 메모리 관리 유닛(MMU)를 통해서 물리적 주소로 변환되고, 물리적 주소는 어느 함수 및 오프셋을 의미하는지를 식별한다.
- [0071] [0075] 이러한 맥락에서 물리적 주소는 "게스트/운영 체제 내에서 액세스 가능한 주소 변환 계층에서 최하위 주소(lowest address)"를 의미한다.
- [0072] [0076] 도 1은, 본 발명의 한 실시 예에 따라, 적어도 하나의 외부 디바이스(214)에 대해서 입/출력 저장 명령(30)을 처리하기 위한 데이터 처리 시스템(210)의 블록도를 묘사한다. 데이터 처리 시스템(210)은 입/출력 버스 컨트롤러(20)에 의해서 적어도 하나의 입/출력 버스(22)에 통신 가능하게 결합된 시스템 네스트(18), 코어(12), 시스템 펌웨어(10) 및 비동기 코어-네스트 인터페이스(14)를 포함하는 데이터 처리 유닛(216)을 포함한다. 입/출력 버스 컨트롤러(20)는 또한 다수의 외부 디바이스들(214)에 다수의 입/출력 버스들(22)을 통해 결합될 수 있다.
- [0073] [0077] 데이터 처리 유닛(216)은 애그리게이션 버퍼(16)를 통해 시스템 네스트(18)에 통신 가능하게 결합된다. 시스템 네스트(18) 및 입/출력 버스 컨트롤러(20)의 일부로서 버퍼-입/출력 버스 컨트롤러 인터페이스(28)을 통해 입/출력 버스(22)에 통신 가능하게 결합된 외부 디바이스(214)로부터 데이터를 비동기로 로드하고 및/또는 외부 디바이스(214)에 데이터를 비동기로 저장하도록 시스템 네스트(18)는 구성된다.
- [0074] [0078] 애그리게이션 버퍼(16)은 비동기 코어-네스트 인터페이스(14)에 통신 가능하게 결합된다. 시스템 펌웨어(10)은 입/출력 저장 명령(30)을 처리하기 위한 비동기 입/출력 드라이버 코드(32)를 포함한다. 코어(12)는 비동기 입/출력 드라이버 코드(32)의 상태 정보를 위한 메모리 요건들을 처리하기 위해 비동기 셋업 코드(34)를 포함한다. 비동기 코어-네스트 인터페이스(14)는 로컬 완료된 데이터(the data with local completion)를 포워드 하기 위해 비동기 코어-네스트 인터페이스 포워딩 컴포넌트(36)를 포함한다. 애그리게이션 버퍼(16)는 요청을 보낸 후 프리 포 리유스 메시지(a free for reuse message)를 전달하기 위해 조기 완료 로직(26)을 포함한다. 애그리게이션 버퍼(16)는 비동기 버스(38)을 통해 비동기 코어-네스트 인터페이스(14)에 결합된다. 비동기 코어-네스트 인터페이스(14)는 다수의 입/출력 상태 버퍼들(24)을 갖는 입/출력 상태 어레이(44), 어레이 관리 및 액세스 로직(46)을 포함한다. 입/출력 상태 버퍼들(24)은 시스템 네스트(18)로부터 및/또는 입/출력 버스 컨트롤러(20)로부터 리턴된 상태들을, 특히 시스템 네스트(18)로부터 완료 메시지를 수집할 수 있다. 입/출력 상태 버퍼들(24)은 비동기 코어-네스트 인터페이스(14)에 직접적으로 집적된다. 어레이 엔트리의 식별을 갖는 메시지(48), 즉 입/출력 상태 버퍼들(24) 중 하나에 대한 완료 메시지는 시스템 네스트(18)에 의해서 수신될 수 있다.
- [0075] [0079] 본 발명의 방법의 일 실시 예에 따라, 데이터 처리 시스템(210) 상에서 실행되는 운영 체제는 주소를 통한 오프셋(an offset through an address), 전송될 데이터 및/또는 전송될 데이터에 대한 포인터, 및 데이터의 길이로 적어도 하나의 입/출력 함수를 명시하는 상기 입/출력 저장 명령(30)을 발행한다(issue). 데이터 처리 유닛(216)은 이에 의해서 입/출력 저장 명령(30)에 명시된 주소에 의해서 입/출력 함수를 식별하도록 이 구성된다. 데이터 처리 유닛(216)은 입/출력 함수에 대한 액세스가 주소 공간 상에서 그리고 게스트 인스턴스 레벨 상에서 허용되는지를 확인하도록 구성되며, 게스트는 데이터 처리 시스템(210) 상에서 실행된다. 데이터 처리 유닛(216)은 시스템 네스트(18)에서 입/출력 저장 명령(30)의 실행이 완료되기 전에 입/출력 저장 명령(30)을 완료하도록 구성된다. 시스템 펌웨어(10)는, 입/출력 저장 명령(30)의 비동기 실행 동안, 데이터 처리 유닛(216)에 의해서 에러가 검출된다면, 상기 실패한 비동기 실행의 데이터를 전송하는, 인터럽트를 통해 운영 체제에 통지하도록 구성된다.
- [0076] [0080] 어레이 관리 및 액세스 로직(46)은 저장 명령(30)의 완료 메시지를 수집하고 수신된 완료 메시지들에 기초하여 입/출력 상태 버퍼들(24)을 갱신한다. 데이터 처리 유닛(216)은 입/출력 상태 버퍼(24)가 미실행 저장 명령들(30)의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 저장 명령(30)의 실행을 지연한다(delay). 특히, 데이터 처리 유닛(216)은 입/출력 상태 버퍼(24)가 미실행 저장 명령들(30)의 완료에 관한 정보를 저장하기 위해 이용가능 하게 될 때까지, 비동기 및/또는 동기 저장 명령들(30)의 실행을 지연한다.
- [0077] [0081] 입/출력 상태 버퍼들(24)은 시스템 네스트(18)로부터 및/또는 입/출력 버스 컨트롤러(20)으로부터 메시

지 상태들을, 특히 시스템 네스트(18)로부터 완료 상태를 수집한다(collect). 메시지 상태들 및/또는 완료 상태는 바람직하게도 입/출력 상태 버퍼 인덱스에 의해서 번호가 매겨질(numbered) 수 있다.

- [0078] [0082] 시스템 펌웨어(10)은, 입/출력 상태 버퍼들(24)을 입/출력 상태 어레이(44)에서 할당/할당해제 하고 및/또는 새로운 저장 명령(30)의 시작을 개시하는, 어레이 관리 로직(42)를 포함한다.
- [0079] [0083] 입/출력 저장 명령(30)은 사용자 측(40)으로부터 시스템 하드웨어/펌웨어(50)를 분리하는 아키텍처 경계 너머 사용자 인터페이스(the user interface) 측의 데이터 처리 시스템 내 위치한다.
- [0080] [0084] 따라서, 만일 소스 데이터의 길이가 8바이트를 초과하면, 데이터는 입/출력 저장 명령(30)에 의해서 조기 완료 메시지를 갖는 비동기 전송 메커니즘(an asynchronous transmit mechanism with an early completion message) 을 통해 다수의 데이터 패킷들(in multiple data packets)로 애그리게이션 버퍼(16)에 전송되고, 그렇지 않으면 데이터는 하나의 데이터 패킷으로 전송된다.
- [0081] [0085] 본 발명의 데이터 처리 시스템의 일 실시 예에 다른 시스템 메시지는 계층 물리 타겟 주소(a hierarchical physical target address), SMT 스레드 소싱 또는 애그리게이트 버퍼 식별자(sourcing an SMT thread or an aggregate buffer identifier), 데이터의 길이, 입/출력 버스 주소, 또는 입/출력 상태 버퍼 인덱스 중 하나를 포함한다.
- [0082] [0086] 다수의 외부 디바이스들(214)에 대해 저장 명령(30)을 처리하기 위한 큐잉 및 순서 시맨틱스(the queueing and ordering semantics)는 바람직하게도 아래에서 기술하는 바에 따라 수행될 수 있다. 개별 SMT 스레드 대 입/출력 함수 관계(an individual SMT thread versus input/output function relation)를 위해, 모든 레거시 입/출력 로드/저장 명령은 프로세서 유닛(216)의 단일 스레드에 관해서 순서가 매겨질 수 있다(ordered). 신규의 입/출력 저장 명령들은 완전히 서로 맞서(against) 순서가 매겨져 있지 않다. 신규의 입/출력 저장 명령들은 레거시 입/출력 명령들에 맞서 순서가 매겨진다. 다른 입/출력 함수들을 위한 모든 입/출력 명령들은 서로에 맞서 순서가 매겨지지 않다.
- [0083] [0087] 도 2는, 본 발명의 한 실시 예에 따라, 하나의 외부 디바이스(214)에 대해서 입/출력 저장 명령(30)을 처리하기 위한 방법의 메시지 시퀀스 차트를 묘사한다.
- [0084] [0088] 도 2에서 도시된 바와 같이, 상기 방법은 운영 체제가 입/출력 저장 명령(30)을 발행하는 단계(issuing)로 시작한다. 단계 S101에서, 시스템 펌웨어(10)는 프리 입/출력 상태 버퍼 인덱스(a free input/output status buffer index)를 할당한다. 만일 이용 가능한 프리 입/출력 상태 버퍼 인덱스가 없다면, 시스템 펌웨어(10)는 대기한다. 단계 S103에서 시스템 펌웨어(10)는 상기 저장 명령이 비동기 센드 엔진(an asynchronous send engine)에 주입될(injected) 수 있는 지를 체크한다(check). 만일 이 것이 가능하다면, 프로세스는 계속 진행 된다. 만일 이 것이 불가능하다면, 상기 저장 명령은 지연되는데, 상기 지연을 일으키는 저장 명령들이 완료될 때까지 지연된다.
- [0085] [0089] 다음은, 단계들 S100 및 S104에 의해서 표시되었듯이, 시스템 펌웨어(10)는, 만일 데이터의 길이가 8바이트를 초과한다면, 애그리게이션 버퍼(16)로 데이터 패킷을 보내기 위해 시스템 메시지를 반복적으로 발행하고, 한편 데이터가 시스템 메시지에 의해 보내짐이 완료될 때 시스템 펌웨어(10)는 대기 중이 된다. 단계 S102 및 S106에서, 로컬 완료 메시지는 시스템 펌웨어(10)로 보내진다(sent back to).
- [0086] [0090] 그 다음, 단계 S108에서, 시스템 펌웨어(10)는 입/출력 버스 컨트롤러(20)에 대한 단일 네스트 메시지(single nest message)로서 데이터를 비동기로 포워드하기 위해 시스템 메시지를 애그리게이션 버퍼(16)로 발행하고, 한편 애그리게이션 버퍼(16)가 완료 메시지를 보내줄 것을 대기한다(waiting).
- [0087] [0091] 그 다음, 단계 S110에서, 애그리게이션 버퍼(16)는, 시스템 펌웨어(10)에 신호하는(signaling back to), 네스트 메시지를 시스템 네스트(18)에 주입하며, 애그리게이션 버퍼(16)는 상기 센드 연산 직후 리유스에 대해 프리(free for reuse right after the send operation)이다. 그 다음, 애그리게이션 버퍼(16)는 프리 포리유스 메시지(a free for reuse message)를 보낸다.
- [0088] [0092] 단계 S112에서, 시스템 네스트(18)는 상기 메시지를 타겟 위치로 포워드 하고, 뒤이어 단계 S114에서 입/출력 버스 컨트롤러(20)가 상기 메시지를 수신하여 입/출력 버스에 데이터 프레임으로(in a data frame) 데이터를 포워드하며, 그 뒤를 이어서 단계 S116에서 입/출력 버스 컨트롤러(20)는 완료 메시지를 시스템 네스트(18)로 보낸다.
- [0089] [0093] 그 다음 단계 S118에서, 시스템 네스트(18)는 원래(originating) 애그리게이션 버퍼(16)로 완료 메시지

를 포워드 하고, 뒤이어 단계 S120에서 애그리게이션 버퍼(16)는 완료 메시지를 비동기-코어-네스트 인터페이스(the asynchronous core-nest interface)(14)로 포워드 한다. 그 다음 단계 S122에서 비동기-코어-네스트 인터페이스(14)는 각각의 입/출력 상태 버퍼 인덱스에 대한 상태를 입/출력 버퍼(24)에 저장하고, 연산의 완료를 시스템 펌웨어(10)로 신호한다. 마지막으로, 단계 S123에서, 시스템 펌웨어(10)는 입/출력 상태 버퍼 인덱스에 의해서 트래킹하는 입/출력 버퍼(24)를 갱신한다. 입/출력 버퍼(24)는 이제 다시 프리(free)가 된다

- [0090] [0094] 데이터의 전송 동안 에러가 발생하는 경우에, 시스템 펌웨어(10)는 결함들(defects)을 운영 체제에 비동기로 신호한다.
- [0091] [0095] 전송될 데이터가 8바이트 보다 적은 경우에, 애그리게이션 버퍼(16)의 반복적인 채움은 스킵된다(skipped).
- [0092] [0096] 도 3은, 본 발명의 한 실시 예에 따라, 하나의 외부 디바이스(214)에 대해서 입/출력 저장 명령(30)을 처리하기 위한 플로 차트의 제1부분을 묘사하고, 도 4는 상기 플로 차트의 제2부분을 묘사한다.
- [0093] [0097] 데이터 처리 유닛의 시스템 펌웨어는 단계 S200에서 시작한다. 단계 202에서 시스템 펌웨어는 메시지를 통해서 레거시 입/출력 저장 블록 명령(a legacy input/output store block instruction)을 수신한다. 단계 S208에서, 비동기 응답을 갖는 혹은 갖지 않는 미실행 신규의 저장 명령(30)이 있는 지가 체크된다. 만일 있다면, 비동기-코어-네스트 인터페이스(14)가 더 이상 미실행 비동기 응답들을 갖지 않을 때까지 그 것은 대기 된다. 만일 없다면, 프로세스는 즉시 단계 S212 로 진행되어 비동기-코어-네스트 인터페이스(14)가 레거시 입/출력 저장 명령(30)을 메시지로서 보낸다. 그 다음 단계 S214에서 시스템은, 단계 S216에서 레거시 응답 메시지(a legacy response message)를 수신할 때까지, 대기 중이 된다. 그 다음 레거시 입/출력 저장 명령(30)은 단계 S218에서 메시지와 함께 종료되고 시스템 펌웨어(10)은 단계 S220에서 프로세스를 종료한다.
- [0094] [0098] 병렬로, 시스템 펌웨어(10)은 단계 S204에서 메시지에 의해서 입/출력 저장 명령(30)을 수신한다. 단계 S222에서, 프리 상태 슬롯이 존재하는지, 즉 입/출력 상태 버퍼(24)가 이용 가능한지가 체크된다. 만일 있다면, 그 슬롯은 단계 S224에서 사용됨(used)으로 마크되고, 저장 프로세스는, 도 4에서 묘사되는 플로 차트의 제2부분에서 묘사되는, 연결지점 A로 진행한다. 만일 없다면, 코어-네스트 인터페이스(14)가 단계 S226에서 이용 가능한 프리 슬롯을 가질 때까지 대기 된다.
- [0095] [0099] 비동기 실행 완료 메시지가 단계 S206에서 수신될 수 있고, 뒤이어 각각의 입/출력 상태 버퍼 인덱스로 단계 S228에서 트래킹하는 슬롯을 갱신할 수 있다. 그 다음 시스템 펌웨어(10)은 단계 S230에서 프로세스를 종료한다.
- [0096] [00100] 플로 차트의 제2 부분은 연결 지점 A로 시작하고, 도 4에서 묘사된다. 먼저, 단계 S304에서, 8바이트 이상이 전송될 것인지가 체크된다. 만일 그렇다면, 단계 S306에서 코어-네스트 인터페이스는 애그리게이션 버퍼를 16바이트까지의 메시지로 채운다. 로컬 완료 메시지가 단계 S310에서 보내지고, 단계 S304로 복귀할 때까지, 시스템 펌웨어는, 단계 S308에서, 대기 중이 된다. 만일 단계 S304의 체크에서 8바이트보다 적게 남아있다면, 플로는 단계 S312로 진행하고 여기서 코어-네스트 인터페이스는 비동기 입/출력 메시지를 보내고, 뒤이어 단계 S316에서의 버퍼 응답을 위해 단계 S314에서 대기 중이 된다. 그 다음 단계 S318에서, 저장 블록 종료 명령(a finish store block instruction)이 실행되고 플로는 단계 S320 에서 시스템 펌웨어의 종료로 종료된다.
- [0097] [00101] 단계 S328에서, 비동기 코어-네스트 인터페이스 로직은 아웃바운드 프로세스 루프를 시작하고, 뒤이어 단계 S322에서 애그리게이션 버퍼 완료 메시지를 수신하며, 단계 S324에서 데이터 메시지를 애그리게이션 버퍼로 포워드하고, 뒤이어 단계 S326에서 완료 메시지를 시스템 펌웨어로 다시 보낸다. 단계 S330에서 비동기 입/출력 센드 메시지가 수신되고 뒤이어 입/출력 센드 메시지가 애그리게이션 버퍼로 포워드 된다.
- [0098] [00102] 단계 S338에서, 애그리게이션 버퍼 로직은 아웃바운드 프로세스 루프를 시작하고, 뒤이어 단계 S334에서 데이터를 수신하며, 단계 S336에서 애그리게이션 버퍼에 데이터를 집합한다(agggregating). 애그리게이션 버퍼는 또한 단계 S340에서 입/출력 센드 메시지를 수신하고, 뒤이어 단계 S342에서 애그리게이션 버퍼로부터의 데이터를 입/출력 센드 메시지와 함께 포워드 한다. 그 다음, 단계 S344에서 애그리게이션 버퍼로부터의 응답 메시지가 코어-네스트 인터페이스를 통해 시스템 펌웨어로 보내진다.
- [0099] [00103] 이제 도 5를 참조하면, 데이터 처리 시스템(210)의 일 예의 개략도가 도시된다. 데이터 처리 시스템(210)은 적절한 데이터 처리 시스템의 오직 한 예이고, 여기서 기술되는 본 발명의 실시 예들의 사용 범위 또는 기능에 대하여 어떠한 제한을 제안하기 위해 의도된 것이 아니다. 이와 상관없이, 데이터 처리 시스템(210)은

지금까지 여기서 설명한 기능의 어떠한 것도 구현할 수 있고 및/또는 수행할 수 있다.

- [0100] [00104] 데이터 처리 시스템(210)에는, 다수의 다른 범용 또는 특수 목적 컴퓨팅 시스템 환경들 또는 구성들과 동작 가능한, 컴퓨터 시스템/서버(212)가 있다. 컴퓨터 시스템/서버(212)와 함께 사용하기에 적합 할 수 있는 잘 알려진 컴퓨팅 시스템들, 환경들 및/또는 구성들의 예들은 퍼스널 컴퓨터 시스템들, 서버 컴퓨터 시스템들, 썬(thin) 클라이언트들, 짝(thick) 클라이언트들, 핸드헬드 또는 랩탑 디바이스들, 멀티프로세서 시스템들, 마이크로프로세서-기반 시스템들, 셋톱 박스들, 프로그램 가능 가전 제품들, 네트워크 PC들, 미니 컴퓨터 시스템들, 메인 프레임 컴퓨터 시스템들 및, 상기 시스템들 또는 디바이스들 중 어느 하나를 포함하는 분산 형 클라우드 컴퓨팅 환경 등을 포함하지만 이에 한정되는 것은 아니다.
- [0101] [00105] 컴퓨터 시스템/서버(212)는, 컴퓨터 시스템에 의해 실행되는, 프로그램 모듈들과 같은, 컴퓨터 시스템-실행 가능 명령들의 일반적인 관점에서 설명 될 수 있다. 일반적으로, 프로그램 모듈들은 특정 작업들(particular tasks)을 수행하거나 특정 추상 데이터 유형들(particular abstract data types)을 구현하는 루틴들, 프로그램들, 객체들, 컴포넌트들, 로직들, 데이터 구조들 등을 포함 할 수 있다. 컴퓨터 시스템/서버(212)는 통신 네트워크를 통해 링크 된 원격 처리 디바이스들에 의해 작업들이 수행되는 분산형 클라우드 컴퓨팅 환경들을 포함하는 많은 컴퓨팅 환경에서 실시될 수 있다. 분산형 클라우드 컴퓨팅 환경에서, 프로그램 모듈들은 메모리 스토리지 디바이스들을 포함하는 로컬 및 원격 컴퓨터 시스템 스토리지 매체 모두에 위치 할 수 있다.
- [0102] [00106] 도 5에 도시 된 바와 같이, 데이터 처리 시스템(210)에서 컴퓨터 시스템/서버(212)는 범용 컴퓨팅 디바이스의 형태로 도시되어 있다. 컴퓨터 시스템/서버(212)의 컴포넌트들은 하나 또는 그 이상의 프로세서들 또는 프로세싱 유닛들(216), 시스템 메모리(228), 및 시스템 메모리(228)를 포함하는 다양한 시스템 컴포넌트들을 프로세서(216)에 연결시키는 버스(218)를 포함 할 수 있지만, 이에 한정되는 것은 아니다.
- [0103] [00107] 버스(218)는, 메모리 버스 또는 메모리 제어기, 주변장치 버스(a peripheral bus), 가속 그래픽 포트(an accelerated graphics port), 및 다양한 종류의 버스 아키텍처들 중 어느 하나를 사용하는 프로세서 또는 로컬 버스를 포함하는, 여러 유형의 버스 구조들 중 하나 또는 그 이상을 나타낸다. 예를 들어, 이러한 아키텍처들에는 ISA(Industry Standard Architecture) 버스, MCA (Micro Channel Architecture) 버스, EISA(Enhanced ISA) 버스, VESA(Video Electronics Standards Association) 로컬 버스, 및 주변 컴포넌트 인터커넥트(PCI)(Peripheral Component Interconnect) 버스가 포함되지만, 이에 한정되는 것은 아니다.
- [0104] [00108] 컴퓨터 시스템/서버(212)는 일반적으로 다양한 컴퓨터 시스템 판독 가능 매체를 포함한다. 이러한 매체는 컴퓨터 시스템/서버(212)에 의해 액세스 가능한 모든 이용 가능한 매체 일 수 있으며, 휘발성(volatile) 및 비 휘발성 매체, 착탈식(removable) 및 비착탈식 매체를 모두 포함한다
- [0105] [00109] 시스템 메모리(228)는 RAM(230) 및/또는 캐시 메모리(232)와 같은 휘발성 메모리 형태의 컴퓨터 시스템 판독 가능 매체를 포함 할 수 있다. 컴퓨터 시스템/서버(212)는 다른 착탈식/비착탈식, 휘발성/비휘발성 컴퓨터 시스템 저장 매체를 더 포함할 수 있다. 단지 예시로서, 스토리지 시스템(234)은 비착탈식, 비휘발성 자기 매체(도시되지 않았고 일반적으로 "하드 드라이브"로 지칭 됨)로부터의 판독 및 이들로의 기록을 위해 제공 될 수 있다. 도시되지 는 않았지만, 착탈식, 비휘발성 자기 디스크(예를 들어, "플로피 디스크")로부터의 판독 및 그 자기 디스크에의 기록을 위한 자기 디스크 드라이브, 및, CD-ROM, DVD-ROM 또는 다른 광 매체와 같은, 착탈식, 비휘발성 광 디스크로부터의 판독 또는 그 광 디스크에의 기록을 위한 광 디스크 드라이브가 제공 될 수 있다. 그러한 경우들에서, 각각은 하나 또는 그 이상의 데이터 매체 인터페이스들에 의해 버스(218)에 접속 될 수 있다. 후술하는 바와 같이, 메모리 (228)는 본 발명의 실시 예들의 기능들을 수행하도록 구성된 프로그램 모듈들의 세트 (예를 들어, 적어도 하나)를 갖는 적어도 하나의 프로그램 제품을 포함 할 수 있다.
- [0106] [00110] 프로그램 모듈들(242)의 세트 (적어도 하나)를 갖는 프로그램/유틸리티 (240)는 메모리(228)에 저장될 수 있으며, 이 들의 예에는 운영 체제뿐만 아니라, 하나 또는 그 이상의 애플리케이션 프로그램들, 다른 프로그램 모듈들, 및 프로그램 데이터가 포함되나, 이에 한정되지 는 않는다. 운영 체제, 하나 또는 그 이상의 애플리케이션 프로그램들, 다른 프로그램 모듈들, 및 프로그램 데이터 또는 이들의 조합의 각각은 네트워킹 환경의 구현을 포함 할 수 있다. 프로그램 모듈들(242)은 일반적으로 본 명세서에서 설명 된 바와 같은 본 발명의 실시 예들의 기능들 및/또는 방법들을 수행한다
- [0107] [00111] 컴퓨터 시스템/서버(212)는 또한, 키보드, 포인팅 디바이스, 디스플레이 (224) 등과 같은, 하나 또는 그 이상의 외부 디바이스들(214)과 통신 할 수 있다. 이들 외부 디바이스들(214)에는 사용자가 컴퓨터 시스템/

서버(212)와 상호 작용할 수 있게 하는 하나 또는 그 이상의 디바이스들; 및/또는 컴퓨터 시스템/서버(212)가 하나 또는 그 이상의 다른 컴퓨팅 디바이스와 통신 할 수 있게 하는 모든 디바이스들(예를 들어, 네트워크 카드, 모뎀 등)이 포함될 수 있다. 통신은 입력/출력(I/O) 인터페이스들(222)을 통해서 일어난다. 또한, 컴퓨터 시스템/서버(212)는 네트워크 어댑터(220)를 통해서 근거리 통신망(LAN), 일반 광역 통신망(WAN) 및/또는 공중 네트워크(예를 들어, 인터넷)과 같은 하나 또는 그 이상의 네트워크들과 통신 할 수 있다. 도시된 바와 같이, 네트워크 어댑터(220)는 버스(218)를 통해 컴퓨터 시스템/서버(212)의 다른 구성 요소들과 통신한다. 도시되지는 않았지만, 다른 하드웨어 및/또는 소프트웨어 컴포넌트들이 컴퓨터 시스템/서버(212)와 함께 사용될 수 있음을 이해하여야 한다. 이들의 예에는 마이크로 코드, 디바이스 드라이버들, 리턴던트 프로세싱 유닛들, 외장형 디스크 드라이브 어레이들, RAID 시스템들, 테이프 드라이브들 및 데이터 아카이브 스토리지 디바이스들이 포함되지만 이에 한정되지는 않는다.

[0108] [00112] 본 발명은 시스템, 방법, 및/또는 컴퓨터 프로그램 제품이 될 수 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독 가능 스토리지 매체를 포함할 수 있으며, 이 매체 상에 프로세서가 본 발명의 실시 예들을 수행하도록 하는 컴퓨터 판독 가능 프로그램 명령들을 갖는다.

[0109] [00113] 상기 컴퓨터 판독 가능 스토리지 매체는 명령 실행 디바이스에 의해 사용될 명령들을 유지 및 저장할 수 있는 유형의(tangible) 디바이스일 수 있다. 상기 컴퓨터 판독 가능 스토리지 매체는, 예를 들면, 전자 스토리지 디바이스, 자기 스토리지 디바이스, 광 스토리지 디바이스, 전자기 스토리지 디바이스, 반도체 스토리지 디바이스, 또는 전술한 것들의 모든 적절한 조합일 수 있으며, 그러나 이에 한정되지는 않는다. 컴퓨터 판독 가능 스토리지 매체의 더 구체적인 예들의 비포괄적인 목록에는 다음이 포함될 수 있다: 휴대용 컴퓨터 디스크, 하드 디스크, 랜덤 액세스 메모리(RAM), 판독-전용 메모리(ROM), 소거 및 프로그램가능 판독-전용 메모리(EPROM 또는 플래시 메모리), 정적 랜덤 액세스 메모리(SRAM), 휴대용 콤팩트 디스크 판독-전용 메모리(CD-ROM), 디지털 다용도 디스크(DVD), 메모리 스틱, 플로피 디스크, 천공-카드들 또는 명령들이 기록된 홈에 있는 용기된 구조들 같이 기계적으로 인코딩된 디바이스, 및 전술한 것들의 모든 적절한 조합. 본 명세서에서 사용될 때, 컴퓨터 판독 가능 스토리지 매체는 무선 전파들이나 다른 자유롭게 전파되는 전자기파들, 도파관이나 기타 전송 매체(예를 들어, 광섬유 케이블을 통해 전달되는 광 펄스들)를 통해 전파되는 전자기파들, 또는 선(wire)을 통해 전송되는 전기 신호들 같이 그 자체로 일시적인(transitory) 신호들로 해석되지는 않는다.

[0110] [00114] 본 명세서에 기술되는 컴퓨터 판독 가능 명령들은, 예를 들어, 인터넷, 근거리 통신망, 광역 통신망 및/또는 무선 네트워크 등의 통신망(네트워크)을 통해 컴퓨터 판독 가능 스토리지 매체로부터 각각 컴퓨팅/처리 디바이스들로 또는 외부 스토리지 디바이스로부터 외부 컴퓨터로 다운로드 될 수 있다. 상기 통신망은 구리 전송 케이블들, 광 전송 섬유들, 무선 전송, 라우터들, 방화벽들, 스위치들, 게이트웨이 컴퓨터들 및/또는 엣지 서버들을 포함할 수 있다. 각 컴퓨팅/처리 디바이스 내 네트워크 어댑터 카드 또는 네트워크 인터페이스는 상기 통신망으로부터 컴퓨터 판독 가능 프로그램 명령들을 수신하고 그 컴퓨터 판독 가능 프로그램 명령들을 각각의 컴퓨팅/처리 디바이스 내의 컴퓨터 판독 가능 스토리지 매체에 저장하기 위해 포워드 한다.

[0111] [00115] 본 발명의 연산들을 실행하기 위한 컴퓨터 판독 가능 프로그램 명령들은 Smalltalk, C++ 또는 그와 유사 언어 등의 객체 지향 프로그래밍 언어와 "C" 프로그래밍 언어 또는 그와 유사한 언어 등의 종래의 절차적 프로그래밍 언어들을 포함하여, 하나 또는 그 이상의 프로그래밍 언어들을 조합하여 작성된(written) 어셈블러 명령들, 명령-세트-아키텍처(ISA) 명령들, 머신 명령들, 머신 종속 명령들, 마이크로코드, 펌웨어 명령들, 상태-셋팅 데이터, 또는 소스 코드나 목적 코드일 수 있다. 상기 컴퓨터 판독 가능 프로그램 명령들은 전적으로 사용자의 컴퓨터상에서, 부분적으로 사용자의 컴퓨터상에서, 독립형(stand-alone) 소프트웨어 패키지로, 부분적으로 사용자의 컴퓨터상에서 그리고 부분적으로 원격 컴퓨터상에서 또는 전적으로 원격 컴퓨터나 서버상에서 실행될 수 있다. 위에서 마지막의 경우에, 원격 컴퓨터는 근거리 통신망(LAN) 또는 광역 통신망(WAN)을 포함한 모든 종류의 네트워크를 통해서 사용자의 컴퓨터에 접속될 수 있고, 또는 이 접속은 (예를 들어, 인터넷 서비스 제공자를 이용한 인터넷을 통해서) 외부 컴퓨터에 이루어질 수도 있다. 일부 실시 예들에서, 예를 들어 프로그램 가능 로직 회로, 필드-프로그램 가능 게이트 어레이들(FPGA), 또는 프로그램 가능 로직 어레이들(PLA)을 포함한 전자 회로는 본 발명의 실시 예들을 수행하기 위해 전자 회로를 맞춤화하도록 상기 컴퓨터 판독 가능 프로그램 명령들의 상태 정보를 활용하여 상기 컴퓨터 판독 가능 프로그램 명령들을 실행할 수 있다.

[0112] [00116] 본 명세서에서는 본 발명의 실시 예들에 따른 방법들, 장치들(시스템들), 및 컴퓨터 프로그램 제품들의 순서 예시도들 및/또는 블록도들을 참조하여 본 발명의 실시 예들이 기술된다. 순서 예시도들 및/또는 블록도들의 각 블록과 순서 예시도들 및/또는 블록도들 내 블록들의 조합들은 컴퓨터 판독 가능 프로그램 명령들에 의해

구현될 수 있다는 것을 이해할 수 있을 것이다.

[0113] [00117] 이들 컴퓨터 판독 가능 프로그램 명령들은 범용 컴퓨터, 특수목적용 컴퓨터, 또는 기타 프로그램가능 데이터 처리 장치의 프로세서에 제공되어 머신(machine)을 생성하고, 그렇게 하여 그 명령들이 상기 컴퓨터 또는 기타 프로그램가능 데이터 처리 장치의 프로세서를 통해서 실행되어, 상기 순서도 및/또는 블록도의 블록 또는 블록들에 명시된 기능들/동작들을 구현하기 위한 수단을 생성할 수 있다. 이들 컴퓨터 판독 가능 프로그램 명령들은 또한 컴퓨터 판독 가능 스토리지 매체에 저장될 수 있으며, 컴퓨터, 프로그램가능 데이터 처리 장치 및/또는 기타 디바이스들에 지시하여 명령들이 저장된 상기 컴퓨터 판독 가능 스토리지 매체가 상기 순서도 및/또는 블록도의 블록 또는 블록들에 명시된 기능/동작의 특징들을 구현하는 명령들을 포함하는 제조품(an article of manufacture)을 포함하도록 특정한 방식으로 기능하게 할 수 있다.

[0114] [00118] 상기 컴퓨터 판독 가능 프로그램 명령들은 또한 컴퓨터, 기타 프로그램가능 데이터 처리 장치, 또는 다른 디바이스에 로드 되어, 상기 컴퓨터, 기타 프로그램가능 장치 또는 다른 디바이스에서 일련의 동작 단계들이 수행되게 하여 컴퓨터 구현 프로세스를 생성하며, 그렇게 하여 상기 컴퓨터, 기타 프로그램가능 장치, 또는 다른 디바이스 상에서 실행되는 명령들이 순서도 및/또는 블록도의 블록 또는 블록들에 명시된 기능들/동작들을 구현할 수 있다.

[0115] [00119] 도면들 내 플로 차트 및 블록도들은 본 발명의 여러 실시 예들에 따른 시스템들, 방법들 및 컴퓨터 프로그램 제품들의 가능한 구현들의 아키텍처, 기능(functionality), 및 동작(operation)을 예시한다. 이와 관련하여, 상기 플로 차트 또는 블록도들 내 각 블록은 상기 명시된 논리적 기능(들)을 구현하기 위한 하나 또는 그 이상의 실행 가능한 명령들을 포함한 모듈, 세그먼트 또는 명령들의 일부분을 나타낼 수 있다. 일부 다른 구현들에서, 상기 블록에 언급되는 기능들은 도면들에 언급된 순서와 다르게 일어날 수도 있다. 예를 들면, 연속으로 도시된 두 개의 블록들은 실제로는 사실상 동시에 실행될 수도 있고, 또는 이 두 블록들은 때때로 관련된 기능에 따라서는 역순으로 실행될 수도 있다. 블록도들 및/또는 플로 차트 예시도의 각 블록, 및 블록도들 및/또는 플로 차트 예시도 내 블록들의 조합들은 특수목적용 하드웨어 및 컴퓨터 명령들의 명시된 기능들 또는 동작들, 또는 이들의 조합들을 수행하는 특수목적용 하드웨어-기반 시스템들에 의해 구현될 수 있다는 것에 또한 유의한다.

[0116] [00120] 본 발명의 다양한 실시 예들에 대한 설명은 예시의 목적으로 제공된 것이며, 빠짐없이 총 망라하거나 개시되는 실시 예들로 한정하려는 의도가 있는 것은 아니다. 기술된 실시 예들의 범위 및 정신을 벗어남이 없이 이 기술 분야에서 통상의 지식을 가진 자들에게 많은 수정들 및 변형들은 명백할 것이다. 본 명세서에 사용된 용어들은 상기 실시 예들의 원리, 실제 응용 또는 현장에서 발견된 기술들에 대한 기술적 개선을 가장 잘 설명하기 위해 그리고 본 명세서에 기술된 실시 예들을 이 기술 분야에서 통상의 지식을 가진 자들이 이해할 수 있도록 하기 위해 선택되었다.

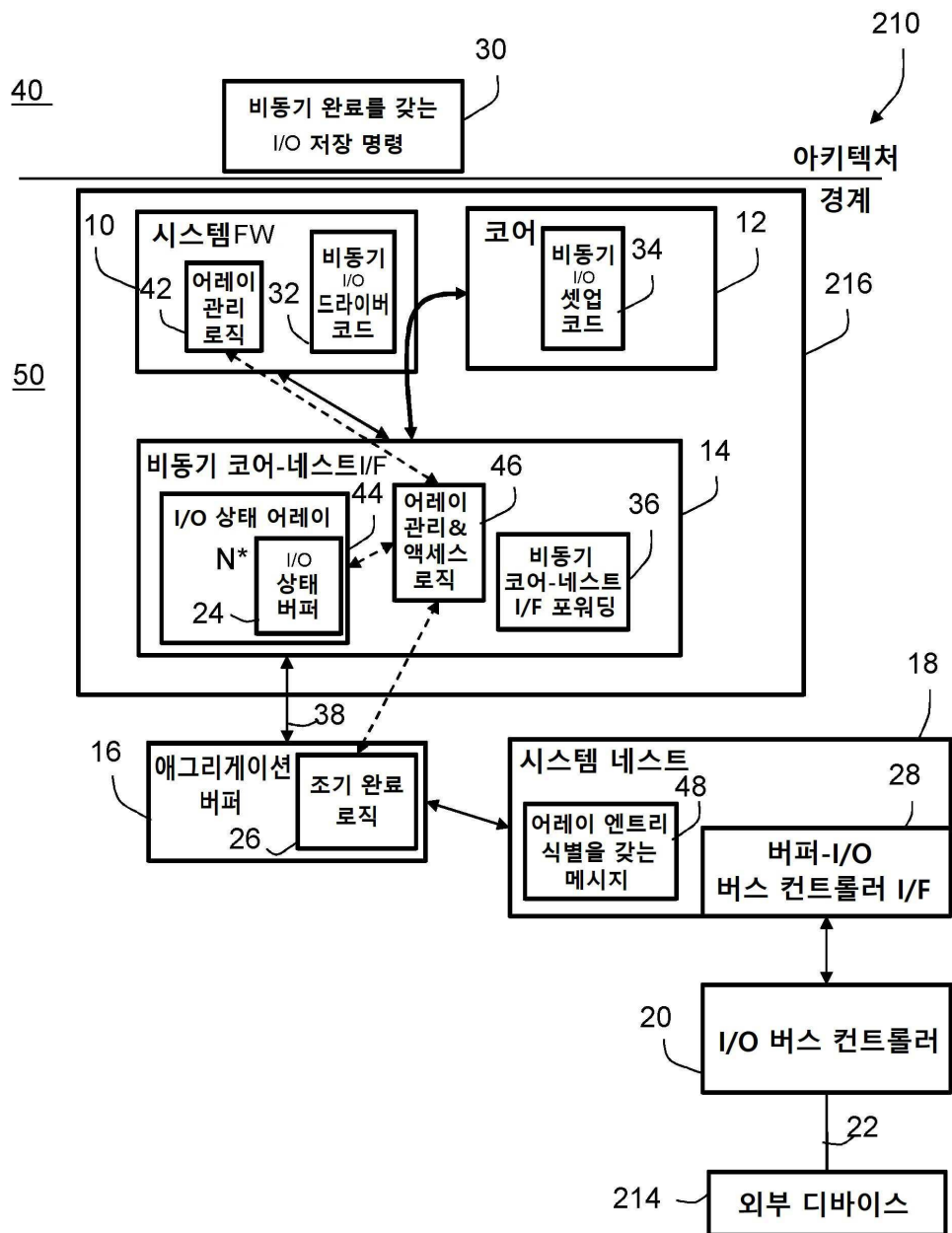
**부호의 설명**

- [0117] 10 시스템 펌웨어(system FW)
- 12 코어(core)
- 14 비동기 코어-네스트 인터페이스(async core-nest I/F)
- 16 애그리게이션 버퍼(aggregation buffer)
- 18 시스템 네스트(system nest)
- 20 입/출력 버스 컨트롤러(I/O bus controller)
- 22 입/출력 버스(I/O bus)
- 24 입/출력 상태 버퍼(I/O status buffer)
- 26 조기 완료 로직(early completion logic)
- 28 버퍼-입력출력 버스 컨트롤러 인터페이스(buffer-I/O bus controller I/F)
- 30 입/출력 저장 명령(I/O store instruction)
- 32 비동기 입력출력 드라이버 코드(async IO driver code)

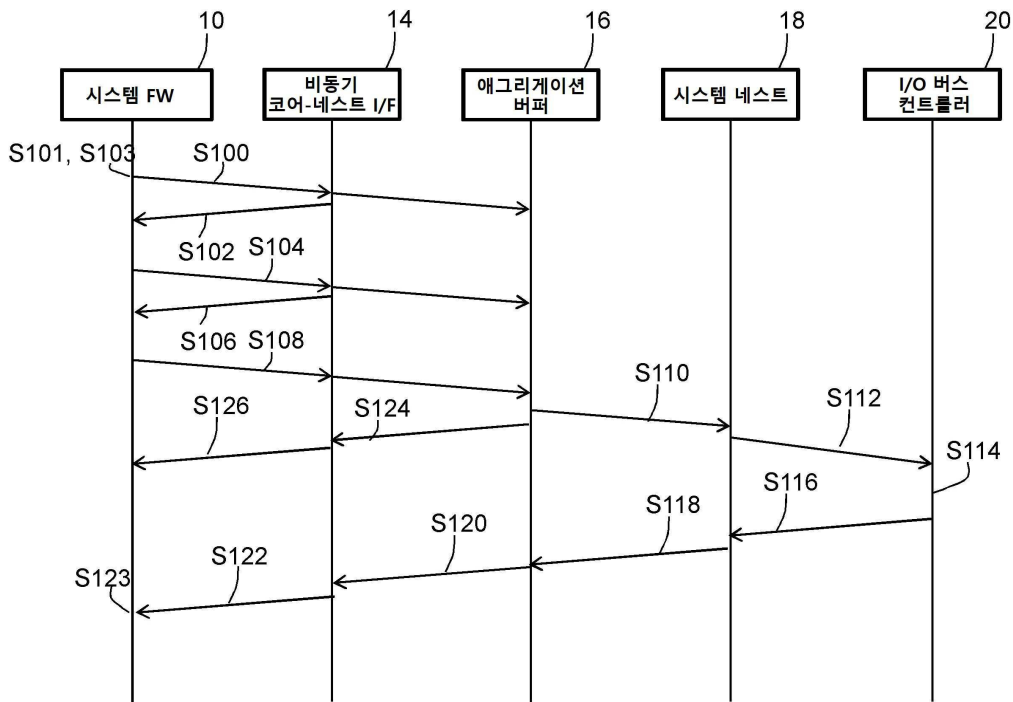
34	입/출력 셋업 코드(I/O setup code)
36	비동기 포워딩(async forwarding)
38	비동기 버스(async bus)
40	사용자 인터페이스(user I/F)
42	어레이 관리 로직(array management logic)
44	입/출력 상태 어레이(I/O status array)
46	어레이 관리 및 액세스 로직(array management & access logic)
48	어레이 엔트리 식별을 갖는 메시지(message with array entry identification)
50	시스템 하드웨어/펌웨어(system HW/FW)
210	데이터 처리 시스템(data processing system)
212	컴퓨터 시스템/서버(computer system/server)
214	외부 디바이스들(external devices)
216	CPU/데이터 처리 유닛(CPU / data processing unit)
218	입출력 버스(IO Bus)
220	네트워크 어댑터(network adapter)
222	입출력 인터페이스들(IO interfaces)
224	디스플레이(display)
228	메모리(memory)
230	RAM
232	캐시(cache)
234	스토리지 시스템(storage system)
240	프로그램/유틸리티(program/utility)
242	프로그램 모듈들(program modules)

도면

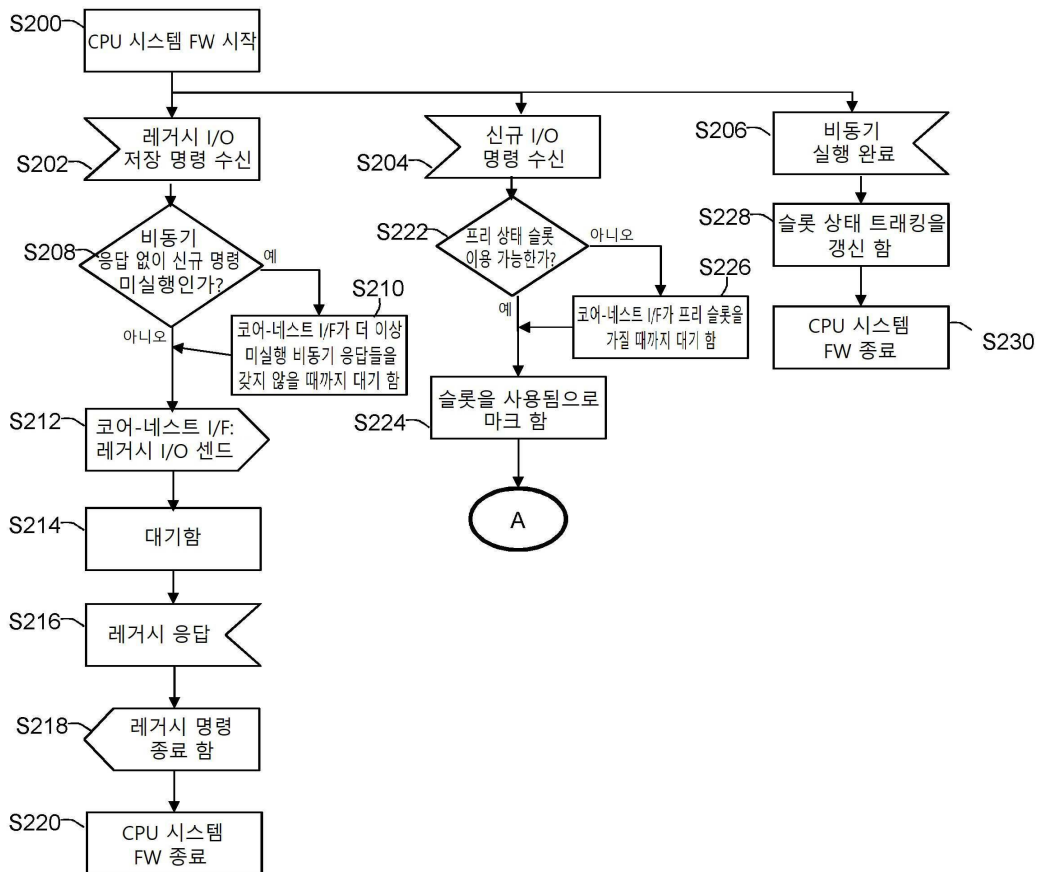
도면1



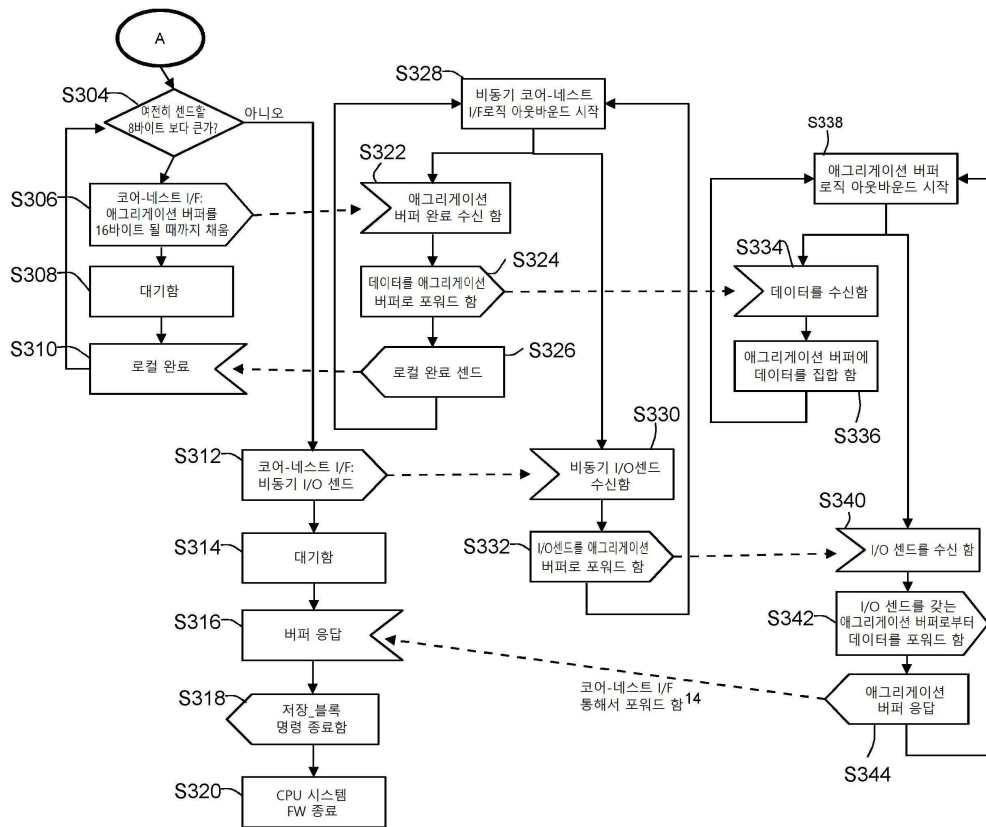
도면2



도면3



도면4



도면5

