



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2014-0077167  
(43) 공개일자 2014년06월23일

(51) 국제특허분류(Int. Cl.)  
*G06F 12/02* (2006.01) *G06F 13/16* (2006.01)  
(21) 출원번호 10-2014-7009069  
(22) 출원일자(국제) 2011년10월07일  
심사청구일자 없음  
(85) 번역문제출일자 2014년04월04일  
(86) 국제출원번호 PCT/US2011/055226  
(87) 국제공개번호 WO 2013/052061  
국제공개일자 2013년04월11일

(71) 출원인  
휴렛-팩커드 디벨롭먼트 컴퍼니, 엘.피.  
미국 텍사스주 77070 휴스턴 콤팩트 센터 드라이브  
웨스트 11445  
(72) 발명자  
고스틴 개리  
미국 텍사스주 75083 어빙 스위트스 400/475 웨스  
트 존 카펜터 프리웨이 1925  
워너 크래이그  
미국 텍사스주 75063 어빙 스위트스 400/475 웨스  
트 존 카펜터 프리웨이 1925  
벽하우스 존 더블유  
미국 콜로라도주 80528-9544 포트 콜린스 이스트  
하모니 로드 3404  
(74) 대리인  
제일특허법인

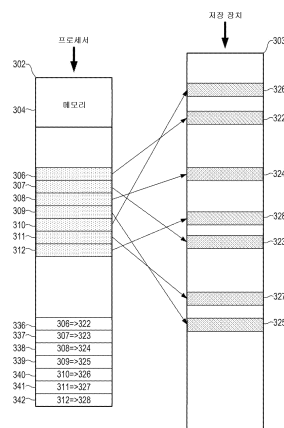
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 **매핑 영구형 저장장치**

**(57) 요약**

저장장치에 액세스하기 위한 컴퓨터 장치 및 관련 방법이 제공된다. 하나의 양태에서, 제어기는 저장장치의 데이터 블록의 어드레스 범위를 복수의 프로세서 중 적어도 하나의 프로세서의 액세스 가능한 메모리 어드레스 범위로 매핑한다. 또 다른 양태에서, 제어기는 복수의 프로세서에 의해 복수의 메모리 내에 캐시된 데이터 블록의 사본이 일관성이 있다는 것을 보장한다.

**대표도** - 도3



## 특허청구의 범위

### 청구항 1

제어기를 포함하는 컴퓨터 장치로서,

상기 제어기는

영구형 저장 매체 내의 데이터 블록의 어드레스 범위를 복수의 프로세서 중 적어도 하나의 프로세서의 액세스 가능한 메모리 어드레스 범위로 매핑하고,

상기 복수의 프로세서에 의해 복수의 메모리 내에 캐시된 데이터 블록의 요청된 부분의 사본이 일관성 있게 유지되는 것을 보장하는

컴퓨터 장치.

### 청구항 2

제 1 항에 있어서,

상기 적어도 하나의 프로세서는

애퍼처(aperture)를 선택하고 - 상기 애퍼처는 상기 적어도 하나의 프로세서의 상기 액세스 가능한 메모리 어드레스 범위 내에 존재하는 사전결정된 어드레스 범위이고, 상기 애퍼처의 상기 사전결정된 어드레스 범위는 상기 데이터 블록의 상기 어드레스 범위에 연관됨 -,

상기 데이터 블록의 요청된 부분에 액세스하도록 상기 애퍼처의 상기 사전결정된 어드레스 범위 내의 하나의 어드레스를 참조하는

컴퓨터 장치.

### 청구항 3

제 2 항에 있어서,

상기 제어기는 디스크립터를 저장하는 제어기이고, 상기 디스크립터는 상기 애퍼처와 상기 데이터 블록 사이의 연관을 포함하는

컴퓨터 장치.

### 청구항 4

제 3 항에 있어서,

상기 디스크립터는 상기 애퍼처에 연관되는

컴퓨터 장치.

### 청구항 5

제 4 항에 있어서,

상기 디스크립터는 상기 데이터 블록으로의 인덱스 또는 포인터와, 상기 애퍼처를 통해 액세스될 때 상기 데이터 블록에 대한 액세스 허용을 표시하는 적어도 하나의 플래그를 더 포함하는

컴퓨터 장치.

#### 청구항 6

제 4 항에 있어서,

상기 제어기는 하나의 프로세서에 의해 캐시된 디스크립터 및 상기 애퍼처의 사본이 다른 프로세서에 의해 캐시된 디스크립터 및 상기 애퍼처의 다른 사본과 일관성이 있음을 보장하는

컴퓨터 장치.

#### 청구항 7

제 6 항에 있어서,

상기 제어기는

이전에 요청된 데이터 블록에 대한 프로세서에 의한 수정이 상기 프로세서가 상기 디스크립터에 액세스하기 전에 저장되는 것을 보장하고,

이전에 요청된 데이터 블록에 대한 프로세서에 의한 수정이 상기 프로세서가 상기 데이터 블록에 액세스하기 전에 저장되는 것을 보장하는

컴퓨터 장치.

#### 청구항 8

제 1 항에 있어서,

상기 영구형 저장 매체는 멤리스터(memristor) 디바이스인

컴퓨터 장치.

#### 청구항 9

제어기를 포함하는 컴퓨터 장치로서,

상기 제어기는

복수의 프로세서 중 적어도 하나의 프로세서의 액세스 가능한 메모리 어드레스 범위 내의 사전결정된 어드레스 범위인 애퍼처를 멤리스터 저장 디바이스 내의 데이터 블록에 연관시키고 - 상기 애퍼처의 상기 사전결정된 어드레스 범위는 상기 데이터 블록의 어드레스 범위에 대응함 -,

상기 데이터 블록의 요청된 부분을 상기 애퍼처를 통해 상기 적어도 하나의 프로세서로 전달하고,

상기 복수의 프로세서에 의해 복수의 메모리 내에 캐시된 상기 데이터 블록의 요청된 부분의 사본이 일관성있게 유지되는 것을 보장하는

컴퓨터 장치.

#### 청구항 10

제 9 항에 있어서,

상기 적어도 하나의 프로세서는

상기 애퍼처를 선택하고,

상기 데이터 블록의 상기 요청된 부분에 액세스하도록 상기 애퍼처의 상기 사전결정된 어드레스 범위 내의 하나의 어드레스를 참조하는 컴퓨터 장치.

#### 청구항 11

제 9 항에 있어서,

상기 제어기는 상기 애퍼처와 상기 데이터 블록 사이의 연관을 포함하는 디스크립터를 저장하는 컴퓨터 장치.

#### 청구항 12

제 11 항에 있어서,

상기 디스크립터는 상기 애퍼처에 연관되는 컴퓨터 장치.

#### 청구항 13

제 12 항에 있어서,

상기 디스크립터는 상기 데이터 블록과 연관되는 인덱스 또는 포인터와, 상기 애퍼처를 통해 액세스될 때 상기 데이터 블록에 대한 액세스 허용을 표시하는 적어도 하나의 플래그를 더 포함하는 컴퓨터 장치.

#### 청구항 14

제 13 항에 있어서,

상기 제어기는 하나의 프로세서에 의해 캐시된 디스크립터 및 상기 애퍼처의 사본이 다른 프로세서에 의해 캐시된 디스크립터 및 상기 애퍼처의 다른 사본과 일관성이 있음을 보장하는 컴퓨터 장치.

#### 청구항 15

제 13 항에 있어서,

상기 제어기는

이전에 요청된 데이터 블록에 대한 프로세서에 의한 수정이 상기 프로세서가 상기 디스크립터에 액세스하기 전에 저장되는 것을 보장하고,

이전에 요청된 데이터 블록에 대한 프로세서에 의한 수정이 상기 프로세서가 상기 데이터 블록에 액세스하기 전에 저장되는 것을 보장하는

컴퓨터 장치.

**명세서**

**기술 분야**

[0001] 클라우드 컴퓨팅 및 휴대용 디바이스의 출현은 데이터에 대한 보다 빠르고 보다 신뢰할 만한 액세스에 대한 수요를 증가시켜왔다. 데이터베이스 성능은 프로세서에 의한 저장장치에 대한 평균 액세스 시간에 크게 의존한다. 지금까지의 시스템은 프로세서와 저장장치 사이의 데이터 전달의 성능을 향상시키기 위해 다양한 기법을 이용한다.

**도면의 간단한 설명**

- [0002] 도 1은 본 출원의 양태에 따른 컴퓨터 장치를 예시한다.
- 도 2는 본 출원의 양태에 따른 제어기의 세부 예시도이다.
- 도 3은 프로세서 및 저장장치의 예시적인 어드레스 공간이다.
- 도 4는 본 출원의 양태에 따른 흐름도를 예시한다.
- 도 5는 본 출원의 양태에 따라 저장되고 있는 데이터의 실무적인 예이다.
- 도 6은 본 출원의 양태에 따라 프로세서로 전달되고 있는 데이터의 실무적인 예이다.

**발명을 실시하기 위한 구체적인 내용**

[0003] RAM(random access memory)으로부터의 데이터 검색은 일반적으로 저장장치로부터의 검색보다 빠르다. 따라서, 프로세서와 저장장치 사이의 지연(latency)을 최소화 하기 위한 과거의 많은 시도는 많은 RAM 캐싱(caching) 전략을 포함한다. 그러나, 저장장치 내에서 이용 가능한 어드레스 범위는 대체로 프로세서의 메모리 어드레스 공간보다 넓다. 따라서, 프로세서가 메모리 캐시 내의 저장장치의 필요한 부분에 액세스하는 것을 보장하기 위해 캐시된 데이터는 저장장치로부터 그리고 저장장치로 자주 전달되어야 한다. RAM과 저장장치 사이의 지속적인 교환은 결코 바람직하지 않고 성능에 부정적인 영향을 미친다.

[0004] 본 출원의 양태는 프로세서를 직접 이용하여 저장장치에 액세스하는 컴퓨터 장치 및 방법을 제공한다. 하나의 양태에서, 영구형 저장 매체 내의 데이터 블록의 어드레스 범위는 복수의 프로세서 중 적어도 하나의 프로세서의 액세스 가능한 메모리 어드레스 범위로 매핑될 수 있다. 또 다른 양태에서, 복수의 메모리 내에 캐시된 데이터 블록의 요청된 부분들의 사본들의 일관성(consistency)이 유지된다.

[0005] 본 출원의 양태, 특징 및 장점은 예 및 첨부 도면에 대한 다음의 설명에 관련하여 검토될 때 이해될 것이다. 본 출원은 다음의 설명으로 제한되지 않으며, 오히려 본 출원의 범위는 첨부된 특허청구범위 및 균등물에 의해 정의된다.

[0006] 도 1은 본 출원의 양태에 따라 다양한 컴포넌트를 묘사하는 예시적인 컴퓨터 장치(100)의 개략적인 도면을 제공한다. 컴퓨터 장치(100)는 컴퓨터와 연결하는데 통상 사용되는 모든 컴포넌트를 포함할 수 있다. 예를 들어, 컴퓨터 장치(100)는 키보드, 마우스, 및/또는 펜-입력, 조이스틱, 버튼, 터치 스크린 등과 같은 다양한 다른 유형의 입력 디바이스는 물론 예를 들어 CRT, LCD, 플라즈마 스크린 모니터, TV, 프로젝터 등을 포함할 수 있는 디스플레이를 가질 수 있다. 컴퓨터 장치(100)는 프로세서(110 및 122), 메모리(112), 및 저장장치(114)도 또한 포함할 수 있다. 메모리(112)는 프로세서(110 및 122)에 의해 검색되고 실행될 수 있는 명령어(115)를 저장할 수 있다. 하나의 예에서, 메모리(112)는 DRAM(dynamic random access memory) 디바이스와 같은 RAM 디바이스일 수 있다. 또 다른 예에서, 메모리(112)는 DIMM(dual in-line memory module)으로서 구성된 복수의 메모리 세그먼트로 분할될 수 있다.

[0007] 프로세서(110 및 122)는 인텔<sup>®</sup> 코퍼레이션으로부터의 프로세서와 같은 많은 주지의 프로세서일 수 있다. 대안으로, 프로세서는 ASIC(application specific integrated circuit)과 같이 동작을 실행하기 위한 전용 제어기일 수 있다. 프로세서(110 및 122)는 국부 캐시(111 및 123)를 각각 포함할 수 있다. 각각의 국부 캐시는 SRAM(static random access memory) 디바이스로부터 단일 레벨 캐시 메모리로서 또는 캐시 계층(cache hierarchy)을 갖는 멀티레벨 캐시 메모리로서 형성될 수 있다. 프로세서(110 및 122)는 키보드, 마우스, 또는 모니터와 같은 기본적인 입력/출력 디바이스에 접속하는 입력/출력(I/O) 인터페이스(113)에도 또한 통신 가능하게 결합될 수 있다. 오직 두 개의 프로세서 및 한 개의 메모리가 도 1에 도시되어 있지만, 컴퓨터 장치(100)가

추가 프로세서 및 메모리를 가질 수 있으며 도 1이 단지 예시적이라는 것이 이해된다.

[0008] 명령어(115)는 프로세서에 의해 직접적으로{예를 들어, 기계어(machine code)} 또는 간접적으로(예를 들어, 스크립트(script)) 실행되는 임의의 명령어 세트를 포함할 수 있다. 그와 관련하여, 용어 "명령어(instruction)", "단계(step)", 및 "프로그램(program)"은 본 명세서에서 호환 가능하게 사용될 수 있다. 명령어는 예를 들어 객체 코드(object code) 내에 또는 소스 코드(source code)의 모듈 내에 임의의 컴퓨터 언어 또는 포맷으로 저장될 수 있다. 명령어(115)는 저장장치(114)에 데이터를 쓰고 저장장치(114)로부터 데이터를 읽도록 프로세서(110 및 122)를 구성할 수 있다.

[0009] 저장장치(114)는 정보가 프로세서(110 및 122)에 의해 검색되고, 조작되고, 저장되게 허용하는 저장 매체일 수 있다. 저장장치(114)는 1971년 레온 추아(Leon Chua)에 의해 상정된 맴리스터(memristor) 기술을 이용할 수 있다. 맴리스터는 메모리 저항기(memory resistor)의 약어이다. 맴리스터는 저항 상태가 이전의 상태 및 현재의 전기 바이어싱 조건(electrical biasing condition)에 의존하는 두 개의 터미널 패시브 디바이스(terminal passive device)이다. 간단한 구조 및 전기적 수동성(electrical passivity)을 고려할 때, 맴리스터는 저장 매체로서 사용하도록 조정될 수 있다. 그러나, 저장장치(114)는 PCM(phase change memory) 또는 STT-RAM(spin-torque transfer RAM)과 같이 다른 유형의 비휘발성 저장 디바이스일 수 있다. 또 다른 예에서, 저장장치는 플래시 메모리 또는 배터리 지원형 DRAM(battery backed DRAM)일 수 있다. 저장장치(114)도 또한 DIMM으로서 구성된 복수의 세그먼트로 분할될 수 있다. 또 다른 예에서, 저장장치(114)는 프로세서(110 및 122)로부터 물리적으로 떨어져 있지만 여전히 프로세서(110)에 의해 액세스 가능한 위치에 존재할 수 있다. 또 다른 예에서, 데이터는 복수의 저장 디바이스 전체에 걸쳐 분산될 수 있다.

[0010] 도 1은 저장장치(114)와 프로세서(110 및 122) 사이의 통신을 다루는 모든 유형의 칩셋 또는 제어 논리(control logic)일 수 있는 제어기(116)도 또한 도시한다. 제어기(116)는 PLD(programmable logic device), PLA(programmable logic array), ASIC 등일 수 있다. 제어기(116)와 프로세서(110 및 122)를 통신 가능하게 결합하기 위해 제 1 버스(118)가 이용될 수 있다. 제 1 버스(118)는 인텔<sup>®</sup> 퀵패스 인터컨넥트(QuickPath Interconnect)와 같은 고속/고주파수 시리얼 링크(high speed/frequency serial link)일 수 있다. 또 다른 예에서, 제 1 버스(118)는 2008년 8월 18일 배포된 하이퍼트랜스포트 사양(HyperTransport Specification), 수정(Revision) 3.1(하이퍼트랜스포트 테크놀로지 컨소시엄(HyperTransport Technology Consortium)) 및/또는 다른 수정을 준수할 수 있다. 제 2 버스(120)는 제 1 버스(118)와 동일한 또는 상이한 기술일 수 있다. 도 1이 프로세서(110), 프로세서(122), 및 저장장치(114)에 직접 결합된 제어기(116)를 도시하지만, 그들 사이에 다른 컴포넌트 또는 디바이스가 배열되어 구성될 수 있다는 것이 이해되어야 한다. 그와 관련하여, 프로세서(110), 프로세서(122), 메모리(112) 및 I/O 인터페이스(113) 사이에 다른 컴포넌트 또는 디바이스가 배열되어 구성될 수 있다는 것이 이해되어야 한다. 비록 컴퓨터 장치(100)의 모든 컴포넌트가 동일한 블록 내에 존재하는 것으로 도 1에 기능적으로 예시되지만, 컴포넌트가 동일한 물리적하우징 내에 저장될 수 있거나 저장되지 않을 수 있다는 것이 이해될 것이다.

[0011] 컴퓨터 장치(100)는 IEEE(Institute of Electrical and Electronic Engineers), 표준 802.3에 의해 망라된 표준을 구현하는 이더넷(Ethernet) 인터페이스와 같은 다른 컴퓨터 또는 디바이스와 통신하기에 적절한 회로(도시되지 않음)를 포함할 수 있다. 또 다른 예에서, 컴퓨터 장치(100)는 IEEE 802.11 표준에 따른 Wi-Fi(wireless fidelity) 인터페이스를 이용할 수 있다. 블루투스(Bluetooth) 또는 토큰 링(token ring)과 같은 다른 표준 또는 프로토콜이 이용될 수 있다는 것이 이해된다.

[0012] 도 2는 본 개시사항의 양태에 따른 제어기(116)의 상세도를 도시한다. 제어기(116)의 하나의 가능한 기능은 블록 201에 인용된 바와 같이 영구형 저장 매체 내의 데이터 블록의 하나의 어드레스를 복수의 프로세서 중 적어도 하나의 프로세서의 액세스 가능한 메모리 어드레스 범위로 매핑하는 것일 수 있다. 본 명세서에서 애퍼처(aperture)는 프로세서의 액세스 가능한 메모리 어드레스 범위 내의 사전결정된 어드레스 범위로서 정의된다. 디스크립터 표(descriptor table, 212)는 복수의 디스크립터(descriptor)를 저장할 수 있으며, 각각의 디스크립터는 애퍼처와 저장장치의 블록 사이의 연관을 포함할 수 있다. 디스크립터 내에서 애퍼처와 저장장치의 블록 사이의 연관이 변경될 수 있지만, 하나의 디스크립터는 오직 하나의 애퍼처에 연관될 수 있다. 애퍼처의 저장장치의 블록과의 연관은 디스크립터 내에 표시된 저장장치의 블록을 수정함으로써 변경될 수 있다. 그런 다음, 저장장치의 블록은 애퍼처의 어드레스 범위를 통해 프로세서 또는 I/O 디바이스에 의해 직접 액세스될 수 있다. 각각의 디스크립터는 애퍼처에 연관되는 저장장치의 블록의 포인터(pointer) 또는 인덱스(index), 및 애퍼처에 대응하는 저장장치의 블록에 대해 허용된 액세스 유형(예를 들어, 판독 액세스, 기록 액세스 등)을 표시하는 적

어도 하나의 플래그(flag)도 또한 포함할 수 있다. 도 2가 제어기(116) 내에 배열된 디스크립터 표(212)를 도시하지만, 디스크립터 표(212)가 제어기(116) 이외의 다른 위치에서 휘발성 또는 비휘발성 메모리 내에 저장될 수 있다는 것이 이해될 것이다. 디스크립터 표(212)는 하드웨어, 소프트웨어, 또는 하드웨어와 소프트웨어의 조합의 형태로 구현될 수 있다.

[0013] 제어기(116)의 또 다른 가능한 기능은 블록 202에 인용된 바와 같이 복수의 프로세서에 의해 복수의 메모리 내에 캐시된 데이터 블록의 요청된 부분들의 사본들이 일관성이 있다는 것(즉, 캐시 일관성)을 보장하는 것일 수 있다. 메모리는 캐시 메모리 또는 다른 국부 메모리일 수 있다. 제어기(116)는 예를 들어 프로세서와 저장장치 사이의 캐시 일관성을 관리하기 위한 하드웨어적 상태 머신(hardwired state machine)일 수 있는 캐시 일관성 엔진(117)을 포함할 수 있다. 캐시 일관성 엔진(117)은 하나의 프로세서에 의한 데이터의 갱신이 다른 프로세서에 의해 캐시된 동일한 데이터의 사본에 적용되는 것을 보장할 수 있다. 또 다른 예에서, 캐시 일관성 엔진(117)은 저장장치(114) 내의 데이터가 프로세서에 의해 캐시된 데이터와 일관성이 있다는 것을 보장할 수 있다. 디스크립터 표(212)와 마찬가지로, 캐시 일관성 엔진(117)도 또한 하드웨어, 소프트웨어, 또는 하드웨어와 소프트웨어의 조합의 형태로 구현될 수 있다.

[0014] 도 3에서, 어드레스 공간(302)은 프로세서의 예시적인 메모리 어드레스 공간을 표현하고, 어드레스 공간(303)은 저장장치의 예시적인 어드레스 공간을 표현한다. 프로세서에 의해 액세스 가능한 메모리 어드레스 범위는 일반적으로 어드레스 버스의 폭과 레지스터(register)에 의해 제한된다. 저장장치의 어드레스 범위는 대체로 프로세서에 의해 액세스 가능한 범위보다 넓다. 예를 들어, 저장장치(114)가 맴리스터 저장 디바이스이면, 저장장치(114)의 어드레스 범위는 대략  $2^{64}$  바이트일 수 있는 반면 프로세서의 어드레스 범위는 대략  $2^{46}$  바이트일 수 있다. 또한, 프로세서의 어드레스 공간의 일부는 다른 목적을 위해 유보될 수 있다. 도 3의 예에서, 어드레스 범위(304)는 메모리(112) 전용인 어드레스 공간 영역이다.

[0015] 하나의 예에서, 애플처의 크기는 컴퓨터 장치(100)의 운영 시스템에 의해 결정된 저장장치의 블록 크기와 일치할 수 있다. 또 다른 예에서, 명령어(115)는 애플처의 크기를 결정할 수 있다. 애플처는 저장장치(즉, 저장장치의 블록) 내의 동일한 크기의 어드레스 범위에 대응할 수 있다. 예를 들어, 도 3의 애플처(306 내지 312)는 데이터 블록(322 내지 328)에 각각 대응할 수 있다. 데이터 블록(322 내지 328)은 어드레스 공간(303)을 통해 산재되는 것으로 도시된다. 각각의 디스크립터(336 내지 342)는 애플처와 저장장치의 블록 사이의 연관성을 표현한다. 프로세서(110)가 예를 들어 저장장치의 블록(322)으로부터 데이터의 일부분을 요청하면, 도 3에 도시된 바와 같이, 프로세서(110)는 애플처(306)를 저장장치의 블록(322)과 연관시키는 디스크립터(336)를 사용할 수 있다. 애플처(306)가 프로세서(110)의 어드레스 공간 내에서 어드레스 범위를 가지므로, 프로세서는 애플처(306)의 어드레스를 참조함으로써 저장장치의 블록(322)의 일부분에 액세스할 수 있다. 따라서, 비록 프로세서의 어드레스가 프로세서의 어드레스 공간 외부에 존재할지라도, 프로세서(110)는 여전히 저장장치의 블록(322)에 액세스할 수 있다.

[0016] 시스템 및 방법의 하나의 실시예가 도 4 내지 도 6에 도시된다. 특히, 도 4는 데이터를 저장장치로부터 복수의 프로세서 중 하나의 프로세서로 직접 전달하기 위한 프로세스(400)의 흐름도를 예시한다. 용이하게 설명하기 위해, 다음의 실무적인 예는 제어기(116)와 프로세서(110) 사이의 통신에 집중한다. 도 5 및 도 6은 저장장치 데이터 전달의 양태를 예시한다. 도 5 및 도 6에 도시된 작용은 도 4의 흐름도를 참조하여 후술될 것이다.

[0017] 도 4의 블록 401에서, 프로세서는 애플처를 선택한다. 애플처를 저장장치의 블록과 연관시키기 전에, 블록 402에 도시된 바와 같이, 프로세서(110)는 애플처에 연관된 디스크립터가 캐시되었는지를 판단할 수 있다. 디스크립터가 캐시되지 않으면, 블록 404에 도시된 바와 같이, 캐시 오류가 발생하고, 프로세서(110)는 디스크립터를 요청할 수 있다. 디스크립터를 제공하기 전에, 블록 408에서, 제어기(116)는 애플처에 현재 연관된 데이터 블록의 이전에 요청된 부분이 국부 캐시(111)와 같은 메모리 내에 여전히 캐시되어 있는지를 판단할 수 있다. 데이터 블록의 이전에 요청된 부분이 메모리 내에 캐시되어 있으면, 블록 410에 도시된 바와 같이, 제어기(116)는 임의의 수정된 데이터를 저장장치(114) 내의 대응하는 데이터 블록으로 전달하면서 데이터 블록 부분을 메모리로부터 제거할 수 있다. 이는 프로세서가 디스크립터에 액세스하기 전에 데이터 블록에 대한 모든 수정이 저장장치(114) 내에 저장되어 기억되었다는 것을 보장할 수 있다. 도 5 내의 예를 참조하면, 프로세서(110)의 국부 캐시(111)는 애플처(306)를 갖는 것으로 도시되어 있으며, 애플처(306)는 저장장치(114)로부터의 블록(322)의 이전에 검색된 사본을 포함할 수 있다. 도 5의 시간  $t_1$ 에서, 프로세서(110)는 캐시 일관성 엔진(117)으로부터 디스크립터를 요청할 수 있다. 디스크립터를 프로세서(110)로 전송하기 전에, 시간  $t_2$ 에서, 제어기(116)의 캐시 일관성 엔진(117)은 애플처(306)의 콘텐츠를 국부 캐시(111)로부터 제거할 수 있다. 시간  $t_3$ 에서, 캐시 일

관성 엔진(117)은 애플처(306)의 콘텐츠를 저장장치(114) 내의 대응하는 어드레스 내에 저장할 수 있다. 도 5의 예에서, 디스크립터 표(212)의 디스크립터(336)는 애플처(306)가 저장장치 내의 어드레스(322)에 연관되는 것을 표시한다.

[0018] 도 4의 블록 412에서, 제어기(116)는 디스크립터를 프로세서(110)로 전송할 수 있다. 도 5에 도시된 바와 같이, 시간 t4에서, 프로세서(110)는 디스크립터를 수신할 수 있고, 프로세서가 디스크립터를 보정하게 허용하면서 디스크립터를 위치(502)에서 캐시(111) 내에 저장할 수 있다. 도 4를 다시 참조하면, 블록 406에 도시된 바와 같이, 프로세서는 대응하는 애플처를 상이한 저장장치의 블록과 연관시키도록 디스크립터를 보정할 수 있다. 이제, 도 6을 참조하면, 캐시 위치(502)에서 디스크립터는 애플처(306)에 연관된 데이터 블록을 322로부터 333으로 변화시키도록 변화될 수 있다. 도 4의 블록 407에서, 프로세서는 이제 새로 선택된 저장장치의 블록인 데이터 블록(333)으로 매핑되는 애플처의 일부분에 액세스할 수 있다. 블록 414에서, 프로세서(110)는 새로 매핑된 저장장치의 블록으로부터의 데이터의 요청된 부분이 선택된 애플처를 통해 캐시되었는지 판단할 수 있다. 데이터가 캐시되지 않았으면, 블록 417에 도시된 바와 같이, 캐시 오류가 발생하고, 프로세서는 애플처를 통해 저장장치로부터 데이터 블록의 일부분을 요청할 수 있다. 데이터를 보내기 전에, 블록 416에 도시된 바와 같이, 제어기(116)는 디스크립터가 캐시되었는지 판단할 수 있다. 디스크립터가 캐시되었으면, 블록 418에 도시된 바와 같이, 제어기(116)는 디스크립터를 제거할 수 있다. 블록 419에 도시된 바와 같이, 제어기는 보정된 디스크립터를 디스크립터 표 내에 저장할 수 있으며, 보정된 디스크립터는 애플처 매핑을 변화시킨다. 이는 프로세서가 데이터 블록에 액세스하기 전에 이전에 요청된 디스크립터에 대한 수정이 저장되는 것을 보장할 수 있다. 도 6의 예는 디스크립터 표(212)의 디스크립터(336) 내에 저장되어 있는 애플처(306)와 저장장치의 블록(333) 사이의 연관을 예시한다. 도 6의 시간 t1에서, 프로세서(110)는 애플처(306)를 통해 새로운 데이터 블록의 일부분을 요청할 수 있다. 캐시 일관성 엔진(117)은 시간 t2에서 디스크립터를 검색할 수 있고, 시간 t3에서 디스크립터를 디스크립터 표(112) 내에 저장할 수 있다. 시간 t4에서, 요청된 데이터 블록 부분은 애플처(306)를 통해 블록(333)으로부터 프로세서(110)로 전달될 수 있다. 도 4를 다시 참조하면, 도 4의 블록 420에서, 제어기(116)는 이제 데이터를 애플처(306)를 통해 저장장치로부터 프로세서로 전송할 수 있다. 블록 422에서, 프로세서는 데이터를 수신할 수 있고, 블록 424에서, 프로세서는 데이터에 액세스할 수 있다.

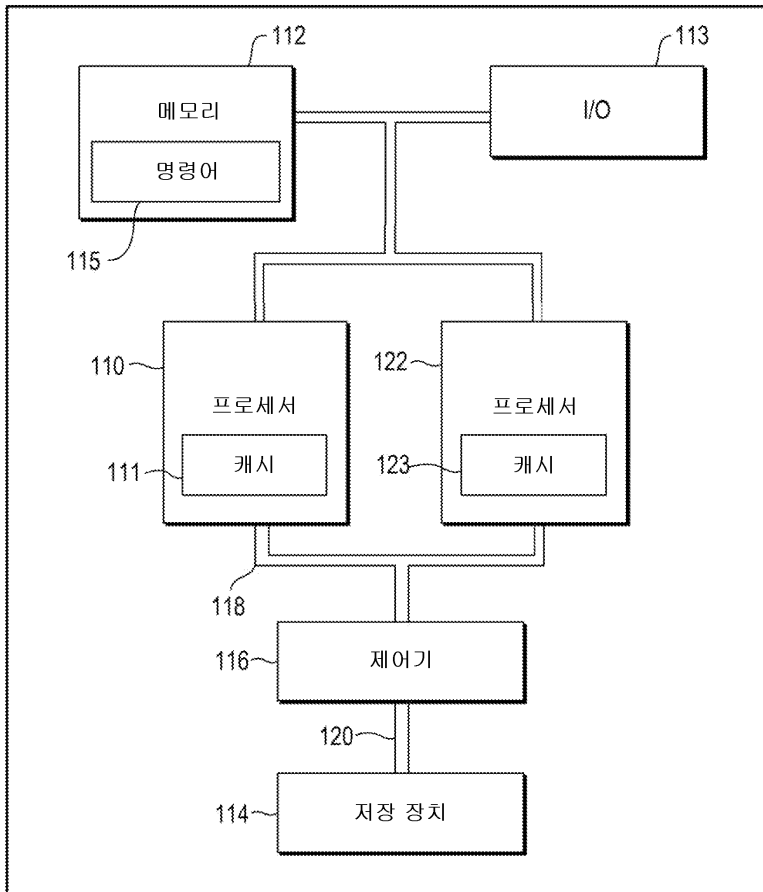
[0019] 유리하게는, 전송된 장치 및 방법은 프로세서의 어드레스 범위보다 넓은 어드레스 범위를 갖는 저장 디바이스로부터의 데이터에 프로세서가 직접 액세스하게 허용한다. 이와 관련하여, 프로그램이 데이터베이스에 액세스하는 속도가 향상된다. 결국, 최종 사용자는 보다 적은 데이터 검색 지연을 경험한다.

[0020] 본 명세서에서 개시가 특정 예에 관해 설명되었지만, 이들 예가 본 개시의 원리를 단지 예시한다는 점이 이해되어야 한다. 그러므로, 예가 다양하게 수정될 수 있는 것은 물론 첨부된 특허청구범위에 의해 정의된 바와 같이 개시된 예의 사상 및 범위를 벗어나지 않으면서 다른 배열이 고안될 수 있다는 점이 이해되어야 한다. 또한, 첨부 도면에는 특정 프로세스가 특정 순서로 도시되지만, 이러한 프로세스는 이러한 순서가 본 명세서에서 명시적으로 제시되지 않는 한 특정 순서로 제한되지 않는다. 오히려, 다양한 단계가 상이한 순서로 또는 동시에 이루어질 수 있고, 단계가 생략되거나 추가될 수 있다.

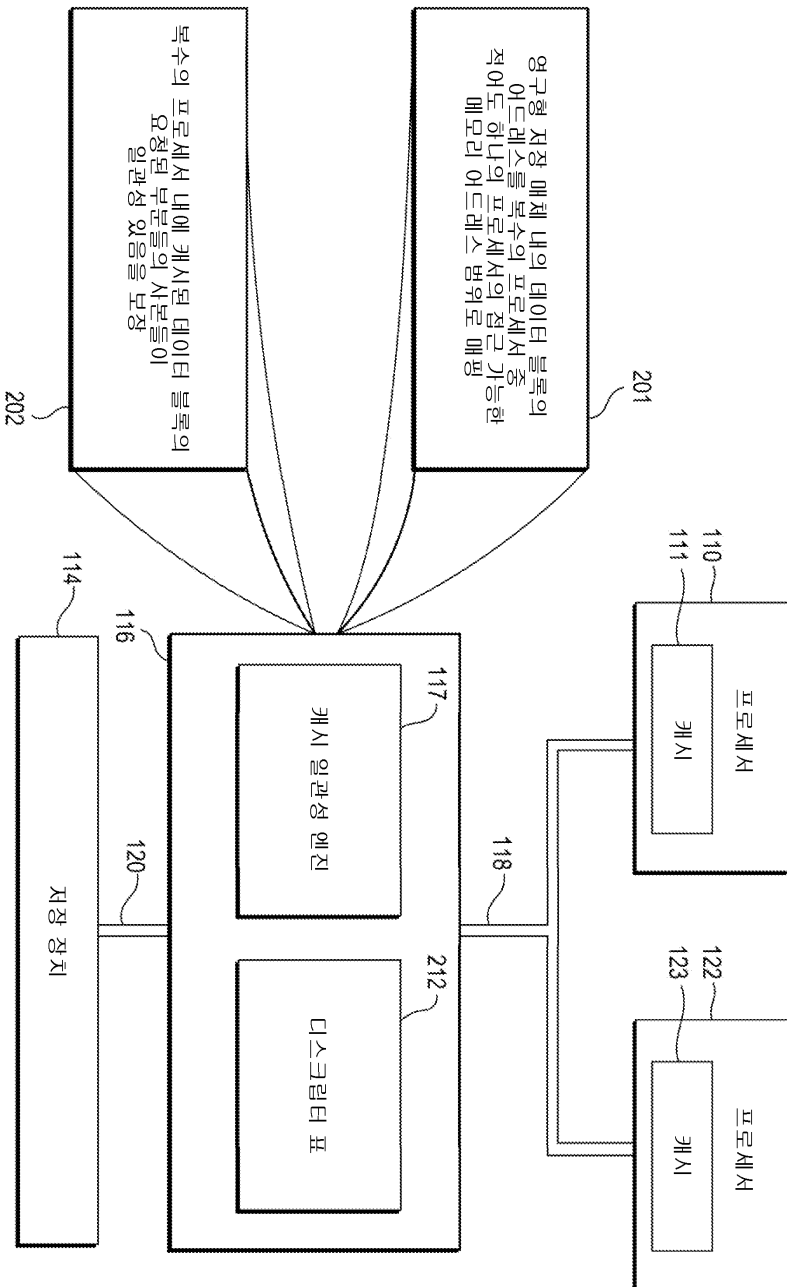
도면

도면1

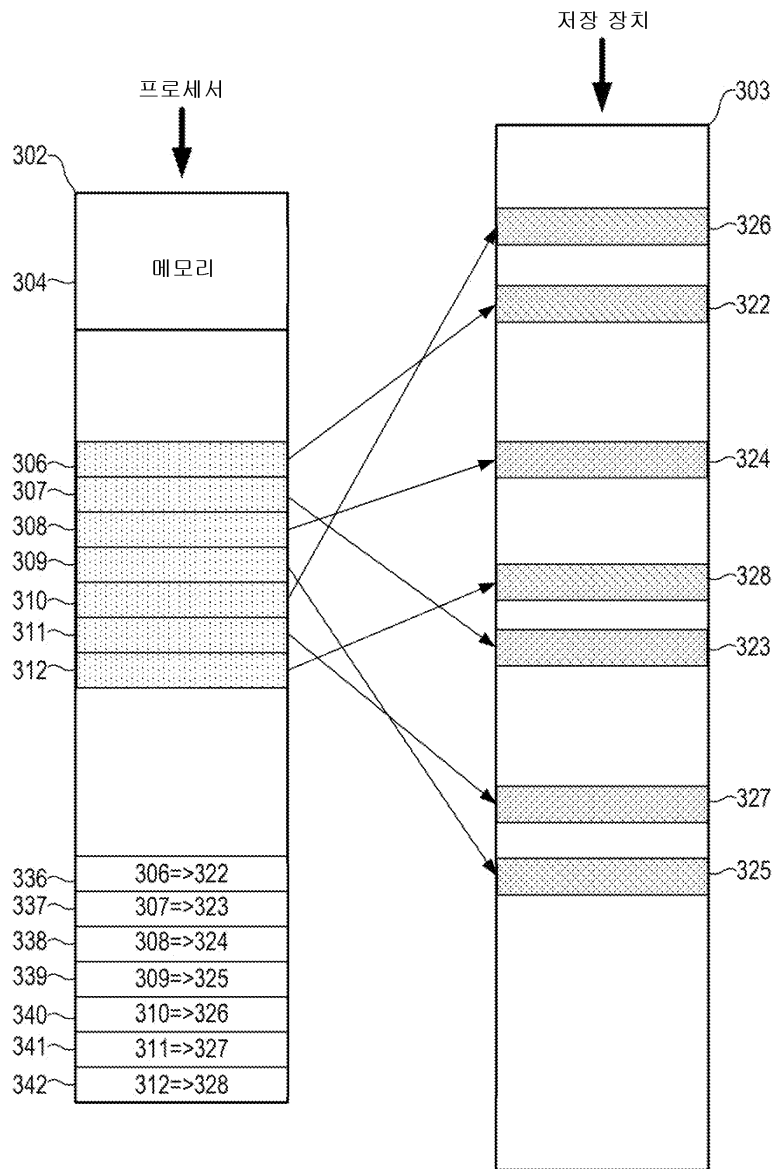
100



도면2

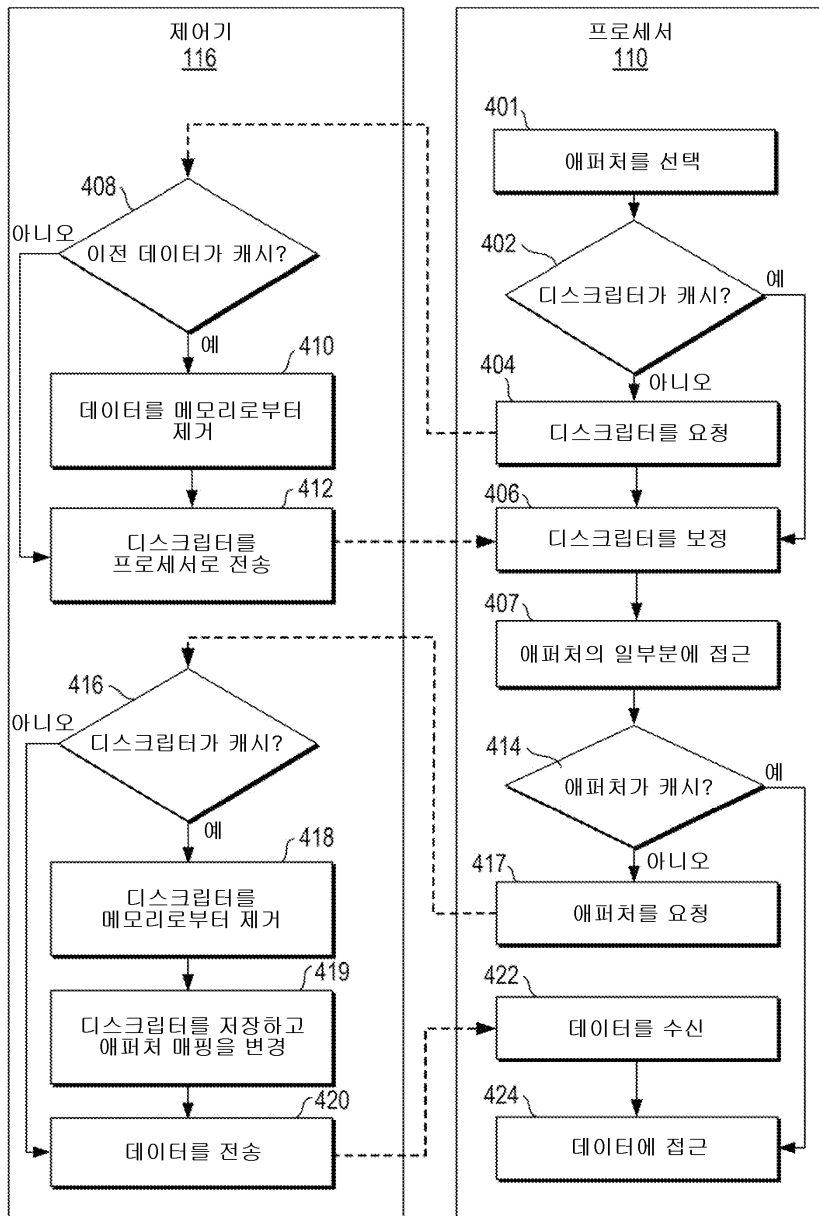


도면3

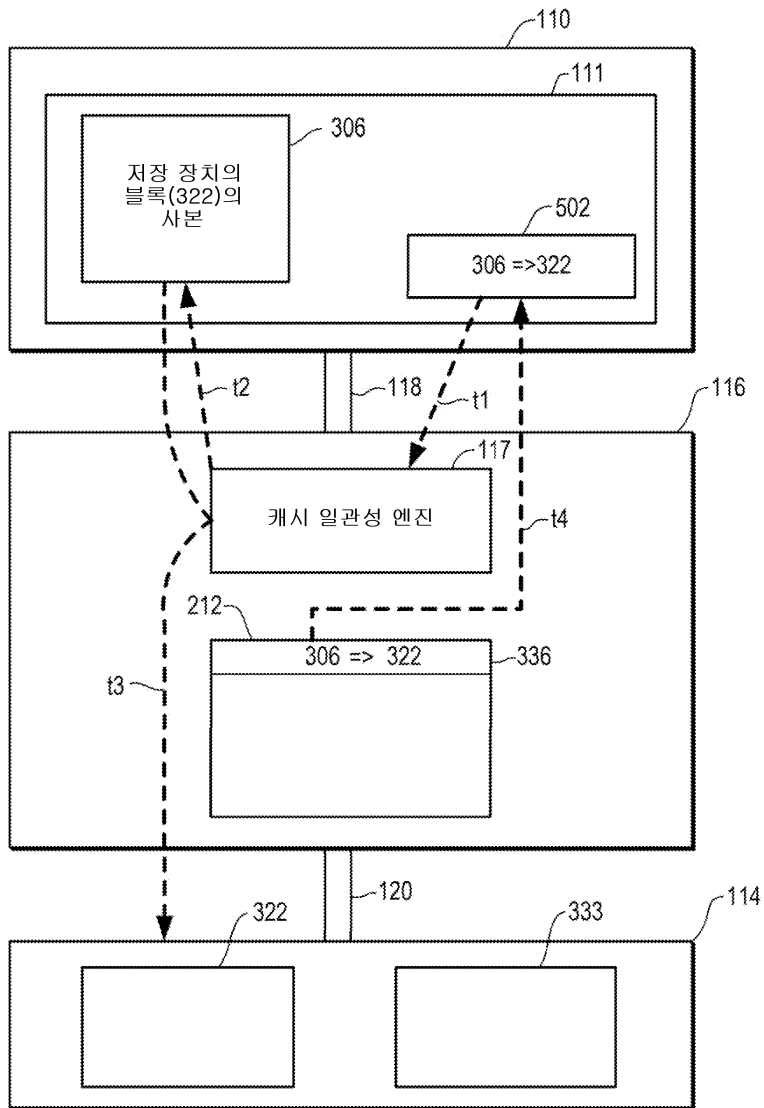


도면4

400



도면5



도면6

