



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0095496  
(43) 공개일자 2020년08월10일

- (51) 국제특허분류(Int. Cl.)  
G06F 12/0817 (2016.01) G06F 12/0804 (2016.01)  
G06F 12/0831 (2016.01) G06F 12/0837 (2016.01)  
G06F 12/0891 (2016.01) G06F 12/126 (2016.01)  
G06F 12/14 (2006.01) G06F 9/30 (2018.01)  
G06F 9/38 (2006.01) G06F 9/46 (2006.01)  
G06F 9/52 (2018.01)
- (52) CPC특허분류  
G06F 12/0817 (2013.01)  
G06F 12/0804 (2013.01)
- (21) 출원번호 10-2020-7018112
- (22) 출원일자(국제) 2018년11월28일  
심사청구일자 없음
- (85) 번역문제출일자 2020년06월23일
- (86) 국제출원번호 PCT/GB2018/053442
- (87) 국제공개번호 WO 2019/110961  
국제공개일자 2019년06월13일
- (30) 우선권주장  
15/831,609 2017년12월05일 미국(US)

- (71) 출원인  
에이알엠 리미티드  
영국 캠브리지 씨비1 9엔제이 체리헌톤 풀번로드 110
- (72) 발명자  
로즈 앤드류 크리스토퍼  
영국 캠브리지 캠브리지셔 씨비 1 9엔제이, 체리헌톤 풀번로드 110, 에이알엠 리미티드  
그리셴드와이트 리차드 로이  
영국 캠브리지 캠브리지셔 씨비1 9엔제이 체리헌톤 풀번로드 110 에이알엠 리미티드  
사이드 알리 가산  
미국 텍사스 78735 오스틴 수트 100 빌딩 1 5707 사우스웨스트 파크웨이 엔시노 트레이스 에이알엠 인코포레이티드
- (74) 대리인  
이화익

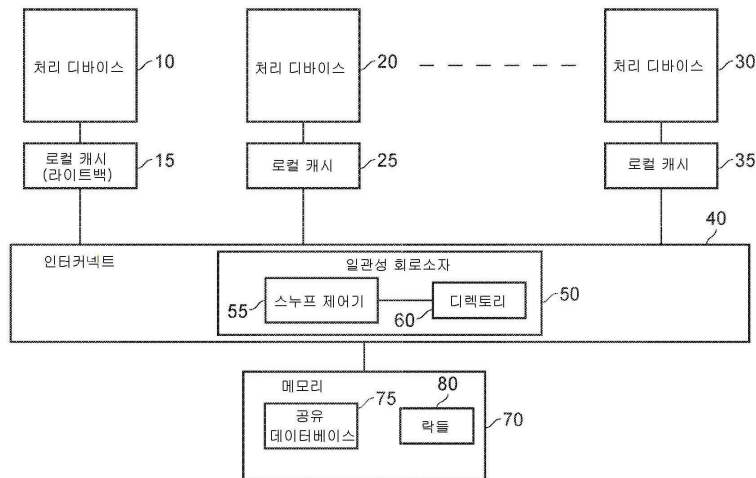
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 기록 연산의 처리 장치 및 방법

(57) 요약

기록 연산들을 처리하는 장치 및 방법을 제공한다. 이 장치는, 제1 메모리 영역내의 적어도 하나의 메모리 장소에 기억하기 위한 기록 데이터를 출력하는 하나 이상의 기록 연산들을 수행하도록, 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 적어도 하나의 명령과, 소유권 (뒷면에 계속)

대표도



권한의 확립 후 실행된 적어도 하나의 기록 명령을 포함하는, 명령들의 시퀀스를 실행하는 제1 처리 디바이스를 갖는다. 제1 처리 디바이스와 관련된 라이트백 캐시는, 하나 이상의 기록 연산들 동안에 출력된 기록 데이터를 기억하는데 사용된다. 일관성 회로소자는, 라이트백 캐시에 결합되고, 적어도 하나의 추가의 처리 디바이스와 관련된 적어도 하나의 추가의 캐시에 결합된다. 제1 처리 디바이스는, 기동 이벤트에 응답하여, 기록 데이터를 라이트백 캐시로부터 메모리에 기록시키도록 클린 연산을 개시한다. 또한, 일관성 회로소자는, 클린 연산에 응답하여, 상기 적어도 하나의 추가의 캐시와 상호작용하여 기록 데이터를 상기 적어도 하나의 추가의 처리 디바이스에 보이게 하기 위해서 하드웨어 프로토콜을 구현한다. 이것은, 특정한 시스템에서 캐시 일관성을 구현하기 위한 매우 효율적이고 비용 효율이 높은 메카니즘을 제공할 수 있다.

(52) CPC특허분류

*G06F 12/0831* (2013.01)

*G06F 12/0837* (2013.01)

*G06F 12/0891* (2013.01)

*G06F 12/126* (2013.01)

*G06F 9/30087* (2013.01)

*G06F 9/3857* (2013.01)

*G06F 9/3863* (2013.01)

*G06F 9/467* (2013.01)

*G06F 9/526* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

제1 메모리 영역내의 적어도 하나의 메모리 장소에 기억하기 위한 기록 데이터를 출력하는 하나 이상의 기록 연산들을 수행하도록, 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 적어도 하나의 명령과, 소유권 권한의 확립 후 실행된 적어도 하나의 기록 명령을 포함하는, 명령들의 시퀀스를 실행하는 제1 처리 디바이스;

제1 처리 디바이스와 관련되고 하나 이상의 기록 연산들 동안에 출력된 기록 데이터를 기억하도록 배치된 라이트백 캐시; 및

라이트백 캐시에 결합되고, 적어도 하나의 추가의 처리 디바이스와 관련된 적어도 하나의 추가의 캐시에 결합된, 일관성 회로소자를 구비하는 장치로서,

제1 처리 디바이스는, 기동 이벤트에 응답하여, 기록 데이터를 라이트백 캐시로부터 메모리에 기록시키도록 클린 연산을 개시하고;

일관성 회로소자는, 클린 연산에 응답하여, 상기 적어도 하나의 추가의 캐시와 상호작용하여 기록 데이터를 상기 적어도 하나의 추가의 처리 디바이스에 보이게 하기 위해서 하드웨어 프로토콜을 구현하는, 장치.

#### 청구항 2

제 1 항에 있어서,

장치는, 제1 처리 디바이스에 의해 기동된 소프트웨어 프로토콜과 일관성 회로소자에 의해 구현된 하드웨어 프로토콜의 조합에 의해 제1 메모리 영역에 관해 캐시 일관성 프로토콜을 구현하는, 장치.

#### 청구항 3

제 1 항 또는 제 2 항에 있어서,

일관성 회로소자는, 클린 연산에 응답하여, 적어도 하나의 추가의 캐시에 의해, 기록 데이터의 적어도 하나의 메모리 장소와 관련된 콘텐츠를 갖는 임의의 캐시 엔트리를 무효화시키도록 배치되는, 장치.

#### 청구항 4

선행하는 청구항 중 어느 한 항에 있어서,

상기 적어도 하나의 추가의 캐시는 복수의 캐시들을 포함하고;

일관성 회로소자는, 상기 복수의 캐시들에서의 캐시마다, 해당 캐시가 캐시하는 데이터를 갖는 메모리 장소들의 표시를 제공하는 레코드를 유지하고;

일관성 회로소자는, 기록 데이터를 적어도 하나의 추가의 처리 디바이스에 보이게 하기 위해서 상기 복수의 캐시들 중에서 어느 캐시들이 상호작용할지를 결정할 때 상기 레코드를 참조하도록 배치되는, 장치.

#### 청구항 5

제 4 항에 있어서,

일관성 회로소자는 스누프 회로소자인, 장치.

#### 청구항 6

선행하는 청구항 중 어느 한 항에 있어서,

다수의 메모리 영역들에 대한 락 표시를 유지하는 락 스토리지를 더 포함하고;

소프트웨어 프로토콜은, 제1 처리 디바이스에 의해 활성화될 때, 제1 메모리 영역에 대한 락 표시가 클리어인지를 판정하도록 배치되고, 상기 락 표시가 클리어인 경우에는, 또한 소프트웨어 프로토콜은, 제1 메모리 영역에 대해 락 표시를 설정하고, 제1 처리 디바이스가 제1 메모리 영역에 대한 소유권 권한을 갖는 것을 제1 처리 디바이스에게 확인하도록 배치되는, 장치.

#### 청구항 7

선행하는 청구항 중 어느 한 항에 있어서,

적어도 하나의 추가의 처리 디바이스에 기록 데이터를 보이게 하였다는 일관성 회로소자로부터의 통지시에, 제1 처리 디바이스는 소유권 권한을 해제하도록 배치되는, 장치.

#### 청구항 8

선행하는 청구항 중 어느 한 항에 있어서,

기동 이벤트는,

클린 연산을 개시하는 상기 시퀀스에서의 클리닝 명령의 제1 처리 디바이스에 의한 실행; 및

기록 데이터를 갖는 라이트백 캐시의 엔트리에 관해 기동된 퇴거 이벤트 중,

하나를 포함하는, 장치.

#### 청구항 9

제 8 항에 있어서,

클리닝 명령은, 클리닝될 데이터를 갖는 라이트백 캐시의 특별한 엔트리를 특정하는, 장치.

#### 청구항 10

제 8 항에 있어서,

클리닝 명령은 라이트백 캐시의 다중 엔트리들의 클리닝을 기동하고, 상기 장치는 다중 엔트리들이 특정되는 것을 가능하게 하는 정보를 기억하는 기억유닛을 더 구비하는, 장치.

#### 청구항 11

제 10 항에 있어서,

기억유닛은 제1 처리 디바이스에 의해 수행된 기록 연산들의 표시들을 기억하도록 배치되고;

클리닝 명령은 명령들의 시퀀스에서 종료 명령이고, 제1 처리 디바이스는:

기억유닛에 표시가 기억되는 각 기록 연산의 대상인 기록 데이터를, 라이트백 캐시로부터 메모리에 클린되게 하

고;

기억유닛에 기억된 기록 연산들의 표시들을 클리어함으로써,  
종료 명령의 실행에 응답하는, 장치.

### 청구항 12

제 11 항에 있어서,

메모리는 지속성의 지점을 형성하는 불휘발성 메모리이고, 종료 명령의 실행에 의해 기록 데이터는 지속성의 지점에 전파되는, 장치.

### 청구항 13

제 11 항 또는 제 12 항에 있어서,

제1 처리 디바이스는, 제1 처리 디바이스에 의해 수행된 기록 연산들의 표시들을, 명령들의 시퀀스에서 시작 명령 후에 기억유닛에 기억시키도록 한층 더 배치되는, 장치.

### 청구항 14

제 13 항에 있어서,

제1 처리 디바이스는 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한이 확립되어 있을 때 제1 메모리 영역에 대한 기록 연산만을 수행하도록 배치되고, 제1 메모리 영역내의 메모리 장소를 지정하는 진행중 기록 명령 전에 시작 명령이 있고 그 진행중 기록명령 후에 종료 명령이 있는, 장치.

### 청구항 15

제 2 항에 종속하는 경우 선행하는 청구항 중 어느 한 항에 있어서,

적어도 하나의 추가의 메모리 영역에 대해서, 캐시 일관성 프로토콜은 일관성 회로소자에 의해서만 구현되고, 제1 처리 디바이스는 적어도 하나의 추가의 캐시에 관해 일관성 회로소자에서 수행한 단계들을 거쳐 적어도 하나의 추가의 메모리 영역에의 배타적 기록 액세스를 얻도록 배치되는, 장치.

### 청구항 16

제1 처리 디바이스, 제1 처리 디바이스와 관련된 라이트백 캐시, 및 라이트백 캐시에 결합되고, 적어도 하나의 추가의 처리 디바이스와 관련된 적어도 하나의 추가의 캐시에 결합된 일관성 회로소자를, 갖는 장치에서 기록 연산들을 처리하는 방법으로서,

제1 처리 디바이스상에서, 제1 메모리 영역내의 적어도 하나의 메모리 장소에 기억하기 위한 기록 데이터를 출력하는 하나 이상의 기록 연산들을 수행하도록, 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 적어도 하나의 명령과, 소유권 권한의 확립 후 실행된 적어도 하나의 기록 명령을, 실행하는 단계;

하나 이상의 기록 연산들 동안에 출력된 기록 데이터를 라이트백 캐시내에 기억하는 단계;

기동 이벤트에 응답하여, 제1 처리 디바이스가, 기록 데이터를 라이트백 캐시로부터 메모리에 기록시키도록 클린 연산을 개시하게 하는 단계; 및

클린 연산에 응답하여, 일관성 회로소자가, 상기 적어도 하나의 추가의 캐시와 상호작용하여 기록 데이터를 상기 적어도 하나의 추가의 처리 디바이스에 보이게 하기 위해서 하드웨어 프로토콜을 구현하게 하는 단계를 포함

하는, 방법.

**청구항 17**

제1 메모리 영역내의 적어도 하나의 메모리 장소에 기억하기 위한 기록 데이터를 출력하는 하나 이상의 기록 연산들을 수행하도록, 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 적어도 하나의 명령과, 소유권 권한의 확립 후 실행된 적어도 하나의 기록 명령을 포함하는, 명령들의 시퀀스를 실행하는 제1 처리수단;

제1 처리수단과 관련되고 하나 이상의 기록 연산들 동안에 출력된 기록 데이터를 기억하는 라이트백 캐시 수단; 및

라이트백 캐시 수단에 결합하고, 적어도 하나의 추가의 처리수단과 관련된 적어도 하나의 추가의 캐시 수단에 결합하는, 일관성 수단을 구비하는 장치로서,

제1 처리수단은, 기동 이벤트에 응답하여, 기록 데이터를 라이트백 캐시 수단으로부터 메모리에 기록시키도록 클린 연산을 개시하고;

일관성 수단은, 클린 연산에 응답하여, 상기 적어도 하나의 추가의 캐시 수단과 상호작용하여 기록 데이터를 상기 적어도 하나의 추가의 처리수단에 보이게 하기 위해서 하드웨어 프로토콜을 구현하는, 장치.

**발명의 설명**

**기술 분야**

[0001] 본 기술은, 기록 연산들을 처리하는 장치 및 방법에 관한 것이다.

**배경 기술**

[0002] 데이터 처리 시스템은, 데이터에 관해 데이터 처리 연산들을 수행할 수 있는 다중 처리 디바이스들을 구비하는 경우도 있을 것이고, 그 각종 처리 디바이스들은 메모리에서의 공유 데이터에 액세스하도록 배치될 수도 있다. 종종, 상기 처리 디바이스들은, 그 처리 디바이스들에 의해 조작된 데이터가, 매번 메모리로부터 폐지되는데 요구된 데이터일 가능성이 있는 것보다 해당 데이터에 보다 빨리 액세스하게 캐시될 수도 있는 하나 이상의 레벨의 로컬 캐시를 구비할 것이다.

[0003] 그렇지만, 로컬 캐시들의 존재는, 가장 최신의 버전이 또 하나의 처리 디바이스의 로컬 캐시에서 캐시될 때 하나의 처리 디바이스가 유효기간이 지난 데이터에 액세스할 가능성이 있는 일관성 문제를 일으킬 수 있다.

[0004] 이러한 상황들을 위해 하드웨어 캐시 일관성 메커니즘을 개발하고 있지만, 그 메커니즘은, 하드웨어 면적 및/또는 실행시간의 점에서 구현하는데 상대적으로 값비쌀 수 있다.

**발명의 내용**

[0005] 제1 구성 예에서 제공하는 장치는, 제1 메모리 영역내의 적어도 하나의 메모리 장소에 기억하기 위한 기록 데이터를 출력하는 하나 이상의 기록 연산들을 수행하도록, 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 적어도 하나의 명령과, 소유권 권한의 확립 후 실행된 적어도 하나의 기록 명령을 포함하는, 명령들의 시퀀스를 실행하는 제1 처리 디바이스; 제1 처리 디바이스와 관련되고 하나 이상의 기록 연산들 동안에 출력된 기록 데이터를 기억하도록 배치된 라이트백(writeback) 캐시; 및 라이트백 캐시에 결합되고, 적어도 하나의 추가의 처리 디바이스와 관련된 적어도 하나의 추가의 캐시에 결합된, 일관성 회로소자를 구비하고; 제1 처리 디바이스는, 기동 이벤트에 응답하여, 기록 데이터를 라이트백 캐시로부터 메모리에 기록시키도록 클린(clean) 연산을 개시하고; 일관성 회로소자는, 클린 연산에 응답하여, 상기 적어도 하나의 추가의 캐시와 상호작용하여 기록 데이터를 상기 적어도 하나의 추가의 처리 디바이스에 보이게 하기 위해서 하드웨어 프로토콜을 구현한다.

[0006] 추가의 구성 예에서 제공한, 제1 처리 디바이스, 제1 처리 디바이스와 관련된 라이트백 캐시, 및 라이트백 캐시에 결합되고, 적어도 하나의 추가의 처리 디바이스와 관련된 적어도 하나의 추가의 캐시에 결합된 일관성 회로소자를, 갖는 장치에서 기록 연산들을 처리하는 방법은, 제1 처리 디바이스상에서, 제1 메모리 영역내의 적어도

하나의 메모리 장소에 기억하기 위한 기록 데이터를 출력하는 하나 이상의 기록 연산들을 수행하도록, 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 적어도 하나의 명령과, 소유권 권한의 확립 후 실행된 적어도 하나의 기록 명령을, 실행하는 단계; 하나 이상의 기록 연산들 동안에 출력된 기록 데이터를 라이트백 캐시내에 기억하는 단계; 기동 이벤트에 응답하여, 제1 처리 디바이스가, 기록 데이터를 라이트백 캐시로부터 메모리에 기록시키도록 클린 연산을 개시하게 하는 단계; 및 클린 연산에 응답하여, 일관성 회로소자가, 상기 적어도 하나의 추가의 캐시와 상호작용하여 기록 데이터를 상기 적어도 하나의 추가의 처리 디바이스에 보이게 하기 위해서 하드웨어 프로토콜을 구현하게 하는 단계를 포함한다.

[0007]

또 추가의 구성 예에서 제공한 장치는, 제1 메모리 영역내의 적어도 하나의 메모리 장소에 기억하기 위한 기록 데이터를 출력하는 하나 이상의 기록 연산들을 수행하도록, 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 적어도 하나의 명령과, 소유권 권한의 확립 후 실행된 적어도 하나의 기록 명령을 포함하는, 명령들의 시퀀스를 실행하는 제1 처리수단; 제1 처리수단과 관련되고 하나 이상의 기록 연산들 동안에 출력된 기록 데이터를 기억하는 라이트백 캐시 수단; 및 라이트백 캐시 수단에 결합하고, 적어도 하나의 추가의 처리수단과 관련된 적어도 하나의 추가의 캐시 수단에 결합하는, 일관성 수단을 구비하고; 제1 처리수단은, 기동 이벤트에 응답하여, 기록 데이터를 라이트백 캐시 수단으로부터 메모리에 기록시키도록 클린 연산을 개시하고; 일관성 수단은, 클린 연산에 응답하여, 상기 적어도 하나의 추가의 캐시 수단과 상호작용하여 기록 데이터를 상기 적어도 하나의 추가의 처리수단에 보이게 하기 위해서 하드웨어 프로토콜을 구현한다.

### 도면의 간단한 설명

[0008]

본 기술은, 아래의 첨부도면에 도시된 것처럼 그 예들을 참조하여, 예시로만 한층 더 설명하겠다:

도 1은 여기서 설명된 기술들을 구현할 수 있는 시스템의 일례의 블록도이고;

도 2는 도 1의 일관성 회로소자에 의해 유지될 수도 있는 디렉토리의 일례를 도시한 것이고;

도 3은 도 1의 시스템의 메모리내에 설치되어도 되는 락(lock) 스토리지의 일례를 도시한 것이고;

도 4a 및 4b는 하나 이상의 기록 연산들을 수행할 때 처리 디바이스내에서 수행되어도 되는 처리의 흐름도를 도시한 것이고;

도 5는 하나 이상의 기록 연산들을 수행한 처리 디바이스에 의해 개시된 클린 연산에 응답하여, 하나의 구성 예에서 일관성 회로소자에 의해 수행된 단계들을 도시하는 흐름도이고;

도 6은 도 1의 시스템에서 이용한 일관성 메카니즘이 기록 연산들을 수행할 메모리 영역에 어떻게 의존되게 할 수 있는지를 도시하는 흐름도이고;

도 7a 내지 7d는 구성 예에서 처리 디바이스들 중 하나에서 실행할 수도 있는 코드의 시퀀스들의 예를 도시한 것이고;

도 8은 클린될 필요가 있는 기록 연산들을 추적하기 위해, 일 구성 예에 있어서의 하나 이상의 처리 디바이스들에 설치될 수도 있는 FIFO(선입선출) 스토리지를 도시한 것이고;

도 9는 일 구성 예에서의 라이트백 캐시를 개략적으로 도시한 것이고;

도 10은 데이터 처리 디바이스가 일례에서 실행할 수도 있는 명령들의 시퀀스의 예를 도시한 것이다.

### 발명을 실시하기 위한 구체적인 내용

[0009]

자신의 로컬 캐시 구조를 갖는 다중 처리 디바이스들을 갖는 시스템에 있어서, 캐시 일관성 프로토콜은, 일반적으로, 각 처리 디바이스가 그 처리 디바이스들 중에서 공유되는 데이터의 동일한 뷰(view)를 갖도록 보장하기 위해서 구현된다. 이러한 캐시 일관성 프로토콜은, 예를 들면, 2개의 상이한 처리 디바이스에 대해서 ("레이싱(racing) 기록" 조건이라고도 할 수도 있는) 동시에 동일 데이터에 관해 기록 연산을 수행시키는 경우 일어날 수 있는 충돌 방식으로 메모리 장소와 관련된 데이터를 갱신할 수 없도록 보장할 필요가 있다. 따라서, 기록 연산이 특별한 메모리 어드레스에서의 데이터에 관해 요구될 때, 공지된 하드웨어 캐시 일관성 기법에 따라, 그 기록 연산을 수행하고 싶은 처리 디바이스는, 일관성 회로소자에 통지하여, 그 일관성 회로소자는, 그 밖의 처리 디바이스들 중 어느 하나가 해당 메모리 어드레스에 관해 충돌 기록을 수행할 위험없이, 기록 연산을 수행할 수 있도록 보장하기 위해서, 그 밖의 처리 디바이스들에서 보유할 수도 있는 임의의 캐시된 복사본들을 유념하

는 임의의 필요한 단계들을 수행한다. 또한, 일관성 회로소자는, 임의의 처리 디바이스가 이어서 메모리 어드레스에 액세스를 요구할 때마다, 가장 최신의 버전의 데이터를 취득하도록 보장할 수 있다.

[0010] 그렇지만, 다중 처리 디바이스들 중에서 데이터를 공유하는 일부 시스템에서, 발명자들은, 하나의 처리 디바이스만이 임의의 시점에서 특별한 메모리 장소에 기록할 자격을 갖는 것을 효과적으로 보장할 수 있는 소프트웨어 기법들도 가동할 준비가 되어 있을 수도 있다는 것을 인식하였다. 단지 예로, 데이터를 갖는 대규모 데이터베이스가 각종 처리 디바이스들 중에서 공유되는 랙 스케일(rack-scale) 컴퓨팅 시스템들에 있어서, 소프트웨어 기술들은, 임의의 특별한 시점에서 임의의 특별한 메모리 장소에 처리 디바이스가 기록할 수 있도록 제어하는데 사용되어도 된다. 따라서, 발명자들은, 이러한 시스템들에서, 캐시 일관성 메카니즘과 통상 관련된 기능성의 일부가 더 이상 일관성 회로소자에 의해 행해질 필요가 없을 것이라는 것을 인식하였다.

[0011] 특히, 발명자들은, 하드웨어 일관성 메카니즘에서 행한 기능성이, 처리 디바이스의 데이터 기록 권한을 확립하는 단계들과, 갱신된 기록 데이터를 그 밖의 처리 디바이스들에 보이게 하도록 기록 연산을 행한 후 취해진 추가의 단계들을 필수적으로 포함하였다는 것을 관찰하였다. 처리 디바이스가 임의의 특별한 메모리 장소에 기록하도록 효율적으로 제어하는 소프트웨어 메카니즘을 이미 제공한 시스템들에 있어서, 발명자들은, 캐시 일관성 프로토콜을 완전히 하드웨어로 구현되는 것을 요구하지 않고 캐시 일관성이 효과적으로 유지될 수 있는 것을 인식하였다. 특히, 데이터 기록 권한을 확립하는데 요구된 단계들은 소프트웨어로 행해질 수 있고, 이때의 하드웨어 메카니즘은 갱신된 기록 데이터를 그 밖의 처리 디바이스들에 보이게 하는 것을 확실하게 하는데 사용된다. 이에 따라, 시스템내에서 캐시 일관성을 지원하는 것과 관련된 (하드웨어 면적 및/또는 실행시간의 면에서) 비용을 상당히 감소시킬 수 있다.

[0012] 일 구성 예에서 제공 가능한 장치는, 제1 메모리 영역내의 적어도 하나의 메모리 장소에 기억하기 위한 기록 데이터를 출력하는 하나 이상의 기록 연산들을 수행하도록, 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 적어도 하나의 명령과, 소유권 권한의 확립 후 실행된 적어도 하나의 기록 명령을 포함하는, 명령들의 시퀀스를 실행하도록 배치된 제1 처리 디바이스를 구비한다. 제1 처리 디바이스는, 제1 처리 디바이스와 관련되고 하나 이상의 기록 연산들 동안에 출력된 기록 데이터를 기억하는 라이트백 캐시를 갖는다. 또한, 라이트백 캐시에 결합되고, 적어도 하나의 추가의 처리 디바이스와 관련된 적어도 하나의 추가의 캐시에 결합된, 일관성 회로소자를 제공한다.

[0013] 소프트웨어 프로토콜의 사용에 의해 보장할 수 있는 것은, 제1 처리 디바이스에 제1 메모리 영역에 관한 소유권 권한이 주어질 때, 그 밖의 처리 디바이스들이 해당 제1 메모리 영역에의 기록 연산들의 충돌을 행할 수 없다는 것이다. 그렇지만, 제1 처리 디바이스에서 생성한 기록 데이터를 그 밖의 처리 디바이스들에 보이게 하는 것을 보장할 필요도 있다.

[0014] 상술한 예들에 따라, 이것은, 제1 처리 디바이스가 기동 이벤트에 응답하여, 기록 데이터를 라이트백 캐시로부터 메모리에 기록시키도록 클린 연산을 개시함으로써 이루어진다. 여기서, 기록 데이터의 항목의 라이트백 캐시로부터 메모리에의 "클리닝"은, 라이트백 캐시에 기억된 해당 데이터 항목(즉, 여기에서 언급한 기록 데이터)의 수정된 복사본을 사용하여 메모리에 기억된 원래의 데이터 항목을 갱신한다는 것을 의미하는 것이라고 이해되어야 한다. 클리닝 연산이 행해지는 것에 더하여, 상기 일관성 회로소자는, 클린 연산에 응답하여, 기록 데이터를 적어도 하나의 추가의 처리 디바이스에 보이게 하는 하드웨어 프로토콜을 구현하기 위해서, 적어도 하나의 추가의 캐시와 상호작용하도록 배치되어 있다.

[0015] 이러한 해결방법에 따라, 처리 디바이스들상에서 실행되는 명령들을 기록하는 소프트웨어 프로그래머는, 소유권 권한을 확립하는데, 특히 제1 메모리 영역에 관해 기록 연산들을 행할 때 관련된 처리 디바이스가 먼저 소프트웨어 프로토콜을 사용하여 소유권 권한을 확립하려고 시도하도록 각 종 처리 디바이스들에서 실행한 명령들의 시퀀스내에 적절한 명령들을 포함하는데 요구된 소프트웨어 프로토콜을 알 필요가 있다. 이것이 의미하는 것은, 하드웨어 일관성 회로소자가 레이스 기록의 문제를 고려할 필요가 없다는 것인데, 그 이유는 소프트웨어가 이러한 문제를 피하는 역할을 하기 때문이다. 그렇지만, 그 후, 하드웨어 프로토콜은, 일관성 회로소자에 의해, 기록 데이터를 적어도 하나의 추가의 처리 디바이스에 보이게 하는 문제를 처리하도록 구현될 수 있다. 일관성 회로소자가 클린 연산의 통지에 응답하여 직접 요구된 단계들을 착수할 수 있으므로, 처리 디바이스들상에서 실행하는 소프트웨어는, 기록 연산들을 행하기 전에 그 밖의 처리 디바이스들이 보유한 데이터의 복사본들을 클린 및/또는 무효화하는 임의의 단계들을 취할 필요가 없거나, 그 데이터가 기록된 후 그 밖의 처리 디바이스들상에서 복사본들을 무효화하는 단계들을 취할 필요가 없다.

[0016] 그 결과, 채택된 해결방법은, 캐시 일관성 프로토콜이 소프트웨어 방식으로 구현될 수 있었던 상황보다 소프트

웨어 시점에서 훨씬 덜 부담된다. 또한, 완전한 캐시 일관성 프로토콜이 하드웨어로 구현되는 시스템보다 (하드웨어 면적 및/또는 실행시간의 면에서) 구현하는데 상당히 덜 비용이 드는데, 그 이유는, 요구된 하드웨어 프로토콜이 기록 데이터를 그 밖의 처리 디바이스들에 보이게 하는 문제와 단지 관련되면 되기 때문이고, 이때의 기록할 권한의 확립은 그 소프트웨어에 위임된다. 일관성 회로소자에 의해 구현된 하드웨어 프로토콜을 소프트웨어 프로그래머가 전체적으로 인식할 수 없게 될 수 있는데, 그 이유는 해당 기능성을 구현하기 위해서 그 처리 디바이스들상에서 실행된 코드에 명령들을 추가할 필요가 없기 때문이고, 대신에, 일관성 회로소자는 클린 연산에 의해 기동된 하드웨어로 요구된 기능성을 전체적으로 구현한다.

[0017] 따라서, 상술한 기술에 따라, 상기 장치는, 제1 처리 디바이스에 의해 기동된 소프트웨어 프로토콜과 일관성 회로소자에 의해 구현된 하드웨어 프로토콜의 조합에 의해 제1 메모리 영역에 관해 캐시 일관성 프로토콜을 구현하도록 배치될 수 있다는 것을 알 것이다. 이것은, 특히 하나의 처리 디바이스만이 임의의 특별한 메모리 장소에 한 번에 기록할 수 있는 적절한 소프트웨어 메카니즘들을 통해 이미 보장하는 시스템들에서, 캐시 일관성을 구현하는 매우 효율적인 메카니즘을 제공할 수 있다.

[0018] 일관성 회로소자가 클린 연산에 응답하는 방식은, 구현에 따라 달라질 수 있다. 일례에서, 일관성 회로소자는, 클린 연산에 응답하여, 적어도 하나의 추가의 캐시에 의해, 기록 데이터의 적어도 하나의 메모리 장소와 관련된 콘텐츠를 갖는 임의의 캐시 엔트리를 무효화시키도록 배치된다. 단순히 그 캐시 엔트리들을 무효화시키면 좋은데, 그 이유는, 이 시점에서, 당해 메모리 장소(들)에 대한 가장 최신의 데이터가 하나 이상의 기록 연산들의 수행 동안에 제1 처리 디바이스에 의해 생성되어 있는 것이라는 것이 알려져 있기 때문이다. 또한, 클린 연산은, 해당 최신 버전의 데이터가 메모리에 출력되게 함에 따라, 적어도 하나의 추가의 처리 디바이스들 중 어느 하나가 적어도 하나의 메모리 장소용 데이터를 요구할 때의 적절한 때에, 그 장소의 로컬 캐시에 미스(miss)가 있고, 정확한 데이터가 메모리로부터 검색될 것이다.

[0019] 상술한 방식으로 엔트리들을 무효화하는 것이 가장 단순한 해결방법을 취하는 것이지만, 그 대신에, 원하는 경우, 기록 데이터의 적어도 하나의 메모리 장소와 관련된 콘텐츠를 갖는 캐시 엔트리는, 그 엔트리를 무효화하는 대신에, 제1 처리 디바이스에 의해 생성된 것처럼 최신의 기록 데이터를 기억하도록 갱신될 수 있었다. 이것은, 해당 데이터가 미래의 어느 때에 요구되면, 이후에 메모리로부터 데이터가 재요구되어야 하는 것을 피할 것이다.

[0020] 일관성 회로소자는, 다양한 방식으로 연산하도록 배치될 수 있다. 예컨대, 일관성 회로소자는, 무효화 또는 갱신 요구를 추가의 처리 디바이스들과 관련된 모든 캐시들에 방송함으로써, 그 캐시들이 기록 연산들에 의해 영향을 받은 데이터의 복사본을 캐시중일 수도 있는지 아닌지를 단지 상기 클린 연산에 응답하도록 배치되어도 된다. 그렇지만, 다른 실시예에서, 일관성 회로소자는, 보다 많은 표적 연산(targeted operation)을 수행하도록 배치될 수 있다. 특히, 일례에서, 적어도 하나의 추가의 캐시는 복수의 캐시들을 포함하고, 일관성 회로소자는, 상기 복수의 캐시들에서의 캐시마다, 해당 캐시가 캐시하는 데이터를 갖는 메모리 장소들의 표시를 제공하는 레코드를 유지한다. 그 후, 일관성 회로소자는, 기록 데이터를 적어도 하나의 추가의 처리 디바이스에 보이게 하기 위해서 상기 복수의 캐시들 중에서 어느 캐시들이 상호작용할지를 결정할 때 상기 레코드를 참조하도록 배치된다.

[0021] 일관성 회로소자에 의해 레코드가 유지되는 이러한 일례에서, 일관성 회로소자는, 스누프 회로소자의 형태를 취할 수도 있다. 스누프 회로소자는, 메모리 어드레스들이 상기 시스템에서의 각종 캐시들에 의해 캐시되는 표시를 유지하고, 캐시 일관성 프로토콜을 구현하기 위해서 해당 정보를 사용하여 스누프 요구들을 특별한 캐시들에 대해 표적으로 삼을 수 있는, 공지된 형태의 캐시 일관성 회로소자다. 그렇지만, 상기 설명된 기술에 따라, 적어도 제1 메모리 영역에 대해서, 스누프 회로소자는 완전 캐시 일관성 프로토콜을 구현하는 것이 요구되지 않는데, 그 이유는, 데이터를 기록하는 소유권을 확립하는 측면이 대신에 소프트웨어 프로토콜에 위임되기 때문이다.

[0022] 소프트웨어 프로토콜을 상기 장치내에서 구현할 수 있는 방식이 많이 있다. 일례에서, 상기 장치는 다수의 메모리 영역들에 대한 락 표시를 유지하는 락 스토리지를 더 포함하고, 소프트웨어 프로토콜은 제1 처리 디바이스에 의해 활성화될 때, 제1 메모리 영역에 대한 락 표시가 클리어인지를 판정하도록 배치된다. 상기 락 표시가 클리어인 경우에, 그 후, 소프트웨어 프로토콜은, 락 표시가 제1 메모리 영역에 대해 설정되게 하고, 제1 처리 디바이스가 제1 메모리 영역에 대한 소유권 권한을 갖는 것을 제1 처리 디바이스에게 확인한다.

[0023] 이에 따라, 메모리 영역마다의 락 표시는, 관련된 메모리 영역이 갱신되고 있는 중일 때 표시하도록 설정될 수 있는 별도의 변수로서 보유된 표시의 형태로서 보여질 수 있다.

- [0024] 락 스토리지에 의해 추적된 각 메모리 영역은, 구현에 따라, 개개의 메모리 어드레스나, 일련의 메모리 어드레스들을 포함할 수도 있다. 각 메모리 장소는 메모리 어드레스에 의해 밝혀지므로, 락 스토리지에 의해 추적된 각 메모리 영역내에 하나 이상의 메모리 장소들이 있을 수도 있다.
- [0025] 일 구성 예에서, 적어도 하나의 추가의 처리 디바이스에 기록 데이터를 보이게 하였다든 일관성 회로소자로부터의 통지시에, 제1 처리 디바이스는 소유권 권한을 해제하도록 배치된다. 따라서, 제1 처리 디바이스는, 해당 제1 처리 디바이스에서 발생한 기록 데이터가 그 밖의 처리 디바이스들 중 어느 하나에 보이게 되도록 보증될 수 있을 때까지, 소유권 권한이 승인되어 있으면 그 소유권 권한을 유지할 것이다. 일부의 경우에, 이것은, 기록 데이터가 상기 시스템내의 일관성의 지점(예컨대, 그 데이터가 메모리에 기록되기 전에 일시적으로 기억되는 버퍼 스토리지)에 도달되어 있으므로 그 밖의 처리 디바이스들 중 어느 하나가 상기 영향을 받은 메모리 장소에서 상기 데이터를 요구할 때 그 밖의 처리 디바이스들이 해당 최신의 기록 데이터가 메인 메모리에 아직 기록되어 있지 않은 경우 그 최신의 기록 데이터를 상기 일관성의 지점으로부터 얻을 것을 보증될 수 있으면, 기록 데이터가 실제로 메인 메모리에 기록되어 있다는 것을 의미하지 않는다.
- [0026] 제1 처리 디바이스에 의해 클린 연산을 개시하는데 사용된 기동 이벤트는 다양한 형태를 취할 수 있다. 예컨대, 기동 이벤트는, 명령들의 시퀀스에서 클리닝 명령의 제1 처리 디바이스에 의한 실행, 해당 클리닝 명령이 클린 연산을 수행되게 하는 실행으로 인해 일어날 수도 있다. 또 하나의 예로서, 기동 이벤트는, 기록 데이터를 갖는 라이트백 캐시의 엔트리에 관해 퇴거 이벤트가 기동되면 생길 수도 있다. 이것은, 예컨대, 라이트백 캐시내의 용량 문제와, 특히 새로운 데이터가 상기 캐시에 할당되게 하는 공간을 만드는데 선택될 필요가 있는 빅티(victim) 엔트리로 인해, 일어날 수 있다. 예를 들면, 이러한 퇴거 이벤트가 기록 연산들 중 하나에 의해 기록되어 있는 엔트리를 빅티 엔트리로서 표적을 삼은 후, 해당 엔트리를 능동적으로 클린하는 클리닝 명령을 제1 처리 디바이스가 실행할지도 모르는 것이 가능하다.
- [0027] 클리닝 명령은 다양한 형태를 취할 수 있다. 예컨대, 단순한 경우에 있어서, 각 기록 명령과 관련된 클리닝 명령이 있을 수도 있어, 요구된 메모리 장소에 기록 데이터가 기록되면, 이하의 클린 명령은 해당 메모리 장소를 특정한다.
- [0028] 그렇지만, 다른 예에서, 단일의 클리닝 명령은 라이트백 캐시의 다중 엔트리들의 클리닝을 기동하도록 배치되어도 되고, 이러한 경우에 상기 장치는 그 다중 엔트리들이 특정되는 것을 가능하게 하는 정보를 기억하는 기억유닛을 더 구비하여도 된다. 예컨대, 공동 소유된 동시계류중인 미국특허출원 번호 15/501,278에 기재된 기술들은, 클리닝 명령이 거기에 정의된 종료 명령의 형태를 취할 수 있는 경우, 기록 연산들 후의 라이트백 캐시내의 다수의 특정된 엔트리들을 메인 메모리에 대해 클린시키도록, 활용될 수 있다. 상기 동시계류중인 건에서 설명된 예들에 있어서, 상기 메모리는 상기 시스템내의 지속성의 지점을 형성하는 불휘발성 메모리를 형성하고, 종료 명령의 실행에 의해 상기 특정된 엔트리들로부터의 기록 데이터는 지속성의 지점에 전과된다. 그렇지만, 본 기술에 있어서, 상기 종료 명령은, 메모리의 형태에 상관없이 사용될 수 있고, 상술한 메카니즘들과 조합될 때, 기록 연산들의 시퀀스에서 생성한 기록 데이터와 관련지어 캐시 일관성을 유지하는 매우 효율적인 기술을 제공할 수 있고, 이때의 소프트웨어 프로토콜은 그 밖의 처리 디바이스가 메모리 영역내에서 기록 연산들의 충돌을 수행할 수 없도록 보장하고, 그 후, 하드웨어 일관성 회로소자는 클린 연산에 응답하여 적어도 하나의 추가의 처리 디바이스에 기록 데이터 전부를 보이게 한다.
- [0029] 일 구성 예에서, 상기 기억유닛은 제1 처리 디바이스에 의해 수행된 기록 연산들의 표시들을 기억하도록 배치되어도 되고, 클리닝 명령은 명령들의 시퀀스에서 종료 명령이어도 되며, 제1 처리 디바이스는, 상기 기억유닛에 표시가 기억되는 각 기록 연산의 대상인 기록 데이터를, 라이트백 캐시로부터 메모리에 클린되게 하고; 또 상기 기억유닛에 기억된 기록 연산들의 표시들을 클리어함으로써, 종료 명령의 실행에 응답하여도 된다.
- [0030] 상술한 종료 명령 기술을 사용함으로써, 데이터 처리장치의 프로그래머가 (기록 명령들의 실행에 의해 개시된) 기록 연산들의 특별한 오더링이 메모리에 대해 일어나도록 보장하고 싶은 상황들에 있어서, 이것은, 프로그래머에 의해 상기 장치가 실시해야 하는 데이터 처리 연산들을 정의하는데 주어진 명령들의 시퀀스에 종료 명령을 부가하는 것에 의해 행해질 수 있다.
- [0031] 일부 구성 예에서, 제1 처리 디바이스는, 제1 처리 디바이스에 의해 수행된 기록 연산들의 표시들을, 명령들의 시퀀스에서 시작 명령 후에 기억유닛에 기억시키도록 한층 더 배치된다. 따라서, 추가의 명령은, 처리 디바이스가 응답하는 명령 세트에 부가될 수도 있는 것에 의해, 시스템 프로그래머가 (시작 명령 후이고 종료 명령으로 종료된) 기록 연산들의 "기록 세트"를 캡슐화할 수 있다. 그 후, 라이트백 캐시의 연산의 면에서와, (스케줄링 목적상, 특별한 명령들이 실행되는 오더(order)를 가변할 수도 있는) 제1 처리 디바이스의 연산에 의해서가 아

닌 면에서, 프로그래머가 원하는 (그 밖의 기록 연산들에 대하여 이 기록 세트의) 기록 오더가 메인 메모리의 콘텐츠에서 준수되도록 보장될 수 있다.

[0032] 일 구성 예에서, 제1 처리 디바이스는 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한이 확립되어 있을 때 제1 메모리 영역에 대한 기록 연산만을 수행하도록 배치되어도 되고, 제1 메모리 영역내의 메모리 장소를 지정하는 진행중 기록명령 전에 시작 명령이 있고 그 진행중 기록명령 후에 종료 명령이 있다.

[0033] 상술한 기술들이 메모리내의 메모리 영역들 전부에 관해 채택될 수 있지만, 그 밖의 예들에서, 상술한 기술들은 특별한 메모리 영역들에 제한될 수도 있고, 그 밖의 메모리 영역들에 대해서는, 풀(full) 하드웨어 캐시 일관성 기법이 원하는 경우 구현될 수도 있다. 따라서, 이러한 일 구성에서, 적어도 하나의 추가의 메모리 영역에 대해서, 캐시 일관성 프로토콜은 일관성 회로소자에 의해서만 구현되고, 제1 처리 디바이스는 적어도 하나의 추가의 캐시에 관해 일관성 회로소자에서 수행한 단계들을 거쳐 적어도 하나의 추가의 메모리 영역에의 배타적 기록 액세스를 얻도록 배치된다. 따라서, 적어도 하나의 추가의 메모리 영역에 대해서, 소프트웨어 프로토콜은 사용되지 않고, 그 대신에 상기 시스템내의 각종의 그 밖의 캐시들에 대해 일관성 회로소자에 의해 직접 취해진 단계들은 기록 연산들을 수행하기 전에 제1 처리 디바이스가 해당 메모리 영역에의 배타적 기록 액세스를 얻도록 보장하는데 사용된다.

[0034] 이제, 특별한 예들에 대해 도면들을 참조하여 설명하겠다.

[0035] 도 1은 상술한 기술들이 구현되어도 되는 시스템의 블록도이다. 다중 처리 디바이스들(10, 20, 30)은, 메모리(70)에 액세스하도록 배치된다. 각 처리 디바이스는, 처리 디바이스에 의해 처리되는 데이터의 로컬 복사본들을 일시적으로 내부에 기억하는 하나 이상의 레벨의 로컬 캐시(15, 25, 35)가 구비될 수도 있다. 각 데이터 값은 상기 메모리(70)내의 메모리 장소를 특정하는 관련 메모리 어드레스를 갖고, 표준 관행에 따라 데이터 값의 복사본은 전형적으로 해당 메모리 어드레스의 표시와 함께 캐시내에 기억된다. 로컬 캐시에서의 데이터가 갱신되면, 예컨대, 해당 데이터에 관해 상기 처리 디바이스에 의해 하나 이상의 기록 연산들이 행해진 결과, 그 갱신된 데이터는 메모리(70)에 역으로 기록될 수 있다. 상기 캐시들은 다양한 방식으로 배치될 수 있다. 예컨대, 캐시가 동시 기록 캐시로서 배치되면, 캐시 라인에 기억된 데이터에 대한 어떠한 갱신도, 갱신을 지연없이 메모리에 복제시키도록 인터커넥트(40)를 거쳐 메모리(70)에 전파된다. 그렇지만, 적어도 하나의 레벨의 로컬 캐시가 라이트백 캐시로서 배치되는 것이 일반적이고, 이때 그 캐시에 보유된 갱신된 버전의 데이터는 메모리에 즉시 전파되지 않는다. 대신에, 더티 플래그는, 메모리보다 더 최신인 데이터를 갖는 캐시 라인들과 관련될 수 있어, 그 캐시 라인 콘텐츠가 이후 상기 캐시로부터 퇴거될 때, 메모리에 대해 필요한 갱신을 할 수 있다. 여기서 설명된 기술들은, 라이트백 캐시로서 배치되는 적어도 하나의 레벨의 로컬 캐시를 갖는 임의의 처리 디바이스에 관련지어 적용되어도 된다. 도 1에서, 상기 캐시(15)는 라이트백 캐시로서 특정되지만, 일 구성 예에서, 로컬 캐시들(15, 25, 35)의 각각은 라이트백 캐시들이어도 된다.

[0036] 개개의 처리 디바이스들이 그들 각각의 로컬 캐시들(15, 25, 35)에서의 데이터의 복사본들을 국소적으로 캐시할 능력으로 인해, 기록 연산들의 충돌이 동일한 메모리 장소에 관해 상이한 처리 디바이스들내에서 발생하지 않도록 보장하고, 각 처리 디바이스가 공유 데이터의 동일한 뷰를 갖도록 보장하기 위해서, 캐시 일관성 프로토콜을 구현하는 것이 필요하다. 특히, 메모리에 보유된 데이터의 일부가 특별한 처리 디바이스에 따라 다를 수도 있지만, 종종, 그 처리 디바이스들이 메모리에 보유된 데이터의 일부를 공유할 경우이다. 일례로서, 상기 메모리는 공유 데이터베이스(75)를 포함하여도 되고, 각 처리 디바이스는 처리 태스크들이 공유 데이터베이스(75)에 보유된 정보와 관련지어 할당되어도 된다. 공유 데이터베이스에서의 특별한 데이터에 대해 다중 처리 디바이스들에 의해 충돌 갱신들이 행해지지 않도록, 또한 각 처리 디바이스에는 공유 데이터베이스(75)에 보유된 데이터의 일관된 뷰가 제공되도록, 보장하는 것이 중요하다.

[0037] 일 구성 예에서, 일관성 회로소자(50)는, 캐시 일관성 프로토콜의 적어도 일부를 구현하기 위한 인터커넥트(40)내에 구비될 수 있다. 캐시 일관성 프로토콜을 구현할 때, 착수될 필요가 있는 태스크가 2개인 것이 필수적이다. 특히, 하나의 처리 디바이스가 특별한 메모리 장소에 관해 기록 연산을 수행하고 싶을 때, 기록 연산을 하고 싶은 처리 디바이스가 해당 메모리 장소에 관해 해당 기록 연산을 수행하는 권한을 확립하고 있도록 보장하는 것이 필요하다. 이를 사용하여, 동일한 메모리 장소에 관해 발생하는 기록 충돌 가능성을 피할 수 있다. 게다가, 특별한 메모리 장소에 대한 데이터가 상기와 같은 기록 연산을 거쳐 갱신되어 있으면, 임의의 그 밖의 처리 디바이스가 해당 메모리 장소에 대한 액세스를 요구할 때마다, 갱신된 데이터를 참조하도록 보장하는 것이 중요하다. 따라서, 캐시 일관성 프로토콜의 일부로서, 그 밖의 처리 디바이스들 전부에게 임의의 기록 연산의 결과를 보이도록 보장하는 것이 필요하다.

- [0038] 상기 요구된 기능성을 고려하는 캐시 일관성 프로토콜을 완전히 하드웨어로 구현하기 위해 상기 인터커넥트내의 일관성 회로소자를 배치하는 것이 가능하지만, 여기서 기재된 기술들에 따라, 적어도 일부의 메모리 영역들에 대해서, 캐시 일관성 프로토콜의 구현은, 소프트웨어와 하드웨어와의 사이에 분할된다. 특히, 일관성 회로소자(50)는, 기록 연산의 결과로서 하나의 처리 디바이스에서 발생한 갱신된 데이터를 그 밖의 처리 디바이스들에 보이도록 보장하는데 사용될 수 있다. 그렇지만, 데이터를 메모리 장소에 기록하는 권한의 취득은 소프트웨어 프로토콜에 위임되어, 일관성 회로소자(50)는 해당 측면을 고려할 필요가 없다. 보다 상세히 후술하듯이, 소프트웨어 프로토콜은, 분할된 소프트웨어/하드웨어 메카니즘에 의해 제어하는 캐시 일관성을 갖는 메모리 영역내의 메모리 어드레스에 임의의 처리 디바이스가 기록 연산을 수행하기 전에, 해당 처리 디바이스가 관련된 메모리 영역에 관해 소유권 권한을 얻도록 보장하기 위해서, 메모리(70)에 유지된 락(lock)들(80)을 사용한 락 기반 메카니즘을 사용할 수 있다. 이러한 해결방법에 의해, 소프트웨어는 상기와 같은 메모리 영역에 관해 생기는 레이스 조건들을 피하는 역할을 맡고 있고, 그 후, 일관성 회로소자는 임의의 갱신된 기록 데이터를 그 밖의 처리 디바이스들에 보이게 하기 위해서 하드웨어 프로토콜을 구현한다.
- [0039] 여기서 설명된 기술에 따라, 하나의 기록 연산, 또는 일부의 경우에 일련의 기록 연산들이 지정된 메모리 영역의 하나 이상의 메모리 장소들내의 갱신된 기록 데이터를 생성하기 위해서 수행되어 있으면, 클린 연산은, 로컬 라이트백 캐시(15)내의 해당 갱신된 기록 데이터를 메모리(70)내의 적절한 메모리 장소들에 전파시키도록 이동된다. 일례에서, 메모리는, 불휘발성 메모리이므로, 상기 시스템내에서의 지속성의 지점을 나타낸다. 특히, 그 데이터가 메모리(70)에 커밋되어 있으면, 해당 데이터는 시스템으로부터 전원을 제거하는 경우에도 유지될 것이다. 따라서, 클린 연산은, 데이터를 상기 시스템내의 지속성의 지점에 전파시킬 수 있다. 그렇지만, 또 하나의 예에서, 상기 메모리는 불휘발성 메모리일 필요는 없다.
- [0040] 상술한 기술들에 따라, 일관성 회로소자는, 그 후, 클린 연산의 개시에 응답하여, 기록 데이터를 그 밖의 처리 디바이스들에 보이게 하기 위해서 다수의 일관성 액션들을 취하는 회로소자다.
- [0041] 예컨대, 처리 디바이스(10)가, 해당 처리 디바이스가 데이터를 제1 메모리 영역에 기록하기 위한 소유권 권한을 확립하기 위해 소프트웨어 프로토콜을 활성화하는 명령을 실행하고 나서, 제1 메모리 영역내의 적어도 하나의 메모리 장소에 대해 하나 이상의 기록 연산들을 수행하여서, 갱신된 기록 데이터를 로컬 라이트백 캐시(15)내에 기억시키는 상황을, 고려한다. 그 후, 처리 디바이스(10)는, 영향을 받은 기록 데이터 전부를 적절한 메모리 장소(들)에 기억하기 위한 메모리(70)에 전파시키기 위해서, 클린연산을 개시시키는 클린 명령을 실행하여도 된다. 그 후, 일관성 회로소자는, 그 밖의 처리 디바이스들(20, 30)이 해당 데이터에 액세스를 요구하는 경우 상기 갱신된 데이터를 참조하도록 보장하는 그 처리 디바이스들의 로컬 캐시들(25, 35)과 관련지어 단계들을 취함으로써 해당 클린 연산에 응답할 수 있다. 일 시나리오에 있어서, 일관성 회로소자는, 영향을 받은 메모리 장소(들)을 지정하는 각 로컬 캐시(25, 35)에 단지 무효화 요구들을 전파하도록 배치되어도 된다. 그 후, 각 로컬 캐시는, 각기 지정된 메모리 장소에 관해 검색을 수행하여 해당 메모리 장소에 대한 상기 캐시에 엔트리가 존재하는지를 판정한다. 존재할 경우, 그 후, 현재의 콘텐츠는 로컬 캐시내에서 무효화될 것이다. 이것은, 이후에 처리 디바이스가 해당 메모리 어드레스에서의 데이터에 대해 요구를 하면, 로컬 캐시에서 미스를 검출하여, 그 후 그 데이터가 메모리로부터 얻게 된 결과, 정확한 데이터가 처리 디바이스(20, 30)에 의해 수신되도록 보장할 것이다.
- [0042] 간단한 구성에 있어서 일관성 회로소자가 무효화 요구들을 그 밖의 로컬 캐시들 전부에 발송할 수도 있지만, 다른 실시예에서, 일관성 회로소자는, 스누프 제어기(55)와 관련 디렉토리(60)를 포함하는, 스누프 회로소자의 형태를 취할 수도 있다. 이 디렉토리는, 각 로컬 캐시(15, 25, 35)에 의해 데이터가 캐시되는 메모리 장소들을 표시하도록, 상기 스누프 제어기에 의해 유지된다. 이에 따라, 클린 연산에 의해 영향을 받은 메모리 장소나 메모리 장소들에 관하여, 스누프 제어기(55)는, 각종 로컬 캐시들 중 어느 로컬 캐시가 상기 영향을 받은 데이터의 캐시된 로컬 복사본을 가질지도 모르는 것을 특정하기 위해 상기 디렉토리내에서 검색을 수행할 수 있으므로, 단지 무효화 요구를 캐시마다 발송하는 대신에 관련 캐시들에 대한 무효화 요구들을 표적으로 삼을 수 있다.
- [0043] 이러한 해결방법에 의해 알 수 있는 것은, 일관성 회로소자가, 상기 갱신된 기록 데이터가 메모리에 기록되는 지점에서, 처리 디바이스들 전부가 해당 데이터의 일관된 뷰를 갖는 것이 보장될 수 있도록 보장하기 위해 클린 연산에 응답할 수 있다는 것이다.
- [0044] 클린 연산이 수행되기 전의 기간에, 또한 처리 디바이스(10)가 하나 또는 메모리 장소들에 대한 갱신된 기록 데이터를 발생하기 위해 하나 이상의 기록 명령들을 실행중인 동안에, 소프트웨어 프로토콜은 그 밖의 처리 디바이스들에 의해 그 하나 이상의 메모리 장소들에의 액세스를 규제할 수 있다. 일부의 경우에, 의도된 상기 데이

터의 소프트웨어 사용은 소프트웨어 프로토콜에 의해 또 하나의 처리 디바이스가 데이터의 오래된 값을 메모리로부터 판독할 수 있기 위한 것이고, 그 밖의 경우에는, 상기와 같은 판독연산이 그 때에 금지될 수도 있다.

[0045] 도 2는 도 1의 디렉토리(60)의 일 구성 예를 개략적으로 도시하는 도면이다. 본 예에서, 상기 디렉토리는 다수의 엔트리들(100, 105)을 갖고, 각 엔트리는, 하나의 메모리 어드레스나, 일련의 메모리 어드레스들을 특정하고 나서, 해당 메모리 어드레스 또는 일련의 메모리 어드레스들내의 데이터의 복사본을 캐시들이 가질 수도 있는 것을 특정한다. 이것은, 스누프 제어기에 의해, 앞서 언급된 클린 연산에 응답하여 로컬 캐시들에 대해 무효화 연산이 실시될 필요가 있는지를 판정하는데 사용될 수 있다. 도 2의 예에서, 체크 표시는 관련된 캐시가 캐시된 복사본을 보유할 수도 있는 것을 나타내는데 사용되고, X표는 관련된 캐시가 복사본을 보유하지 않는 것을 나타내는데 사용된다. 그 체크 표시와 X표가 임의의 적절한 값으로 인코딩될 수 있고, 예컨대, 논리 1값이 체크 표시를 나타내고 논리 0값이 X표를 나타낼 수도 있다는 것을 알 것이다.

[0046] 도 2는 단지 디렉토리(60)의 일 구성 예를 도시하는 것일 뿐이고, 임의의 그 밖의 적절한 구성은 수신된 클린 연산에 응답하여 캐시들에 대해 무효화 연산들이 실시될 필요가 있는 것에 관해 상기 스누프 제어기에 정보를 제공하는데 사용될 수 있었다는 것을 알 것이다.

[0047] 도 3은 사용될 수 있는 락 스토리지(80)의 구성 예를 도시한 것이다. 락 스토리지는, 각각 특별한 메모리 영역에 관계하는 복수의 엔트리들(110, 115, 120)을 가질 수 있다. 락 표시는, 메모리 영역마다 해당 영역내의 적어도 하나의 메모리 어드레스가 그의 데이터 값을 갱신시키는 중인지를 특정하는데 유지될 수 있다. 락 표시가 세트일 경우, 이것은 상기 영역내의 적어도 하나의 메모리 어드레스에 대한 데이터가 갱신중인 것을 나타내는 반면에, 락 표시가 클리어일 경우, 이것은 상기 영역내의 어드레스들 중 아무것도 갱신되는 중인 데이터를 포함하지 않는 것을 나타낸다.

[0048] 처리 디바이스가 락 스토리지에 의해 통제된 영역들 중 하나의 영역내의 메모리 어드레스에 관해 기록 연산을 수행하고 싶을 때, 해당 처리 디바이스에 의해 상기 관련된 락 표시를 설정하려고 시도하기 위해서 명령을 실행할 것이다. 그 관련된 락 표시가 현재 클리어일 경우, 그것은 설정되고, 확인응답은 처리 디바이스에 회신될 것이고, 이것은, 처리 디바이스가 지금 그 관련된 메모리 영역에 대한 소유권 권한을 확립하고 있음에 따라, 해당 메모리 영역에 관해 기록 연산을 자유롭게 수행하는 것을, 실제로 상기 처리 디바이스에 대해 특정한다. 그렇지만, 락 표시가 이미 세트일 경우, 처리 디바이스는, 락을 설정하려고 시도하고 있는 중인 명령 이후에 계속되지 않게 되고, 특히, 해당 스테이지에서 관련 영역내의 메모리 어드레스에 기록하려고 시도하는 임의의 기록 명령들 실행하면 안될 것이다. 일례에서, 관련 영역에 대한 락 표시가 클리어 상태로 복귀되는 시간, 그 시점에서, 락 표시가 다시 설정될 수 있을 때까지 실행은 일시 중지될 것이고, 처리 디바이스가 메모리 영역에 관해 지금 소유권 권한을 갖는 것을 특정하기 위해 상기 처리 디바이스에 발행된 확인응답 그 이후, 기록 명령(들)이 실행될 수 있다. 이러한 해결방법에 의해, 소프트웨어 프로토콜은, 임의의 특별한 시점에서, 단지 하나의 처리 디바이스만이 이러한 각 영역내의 메모리 장소에 관해 기록 연산들을 수행할 권한이 주어지도록 보장하기 위해서, 복수의 메모리 영역들에 대한 소유권 권한을 유지하는데 사용될 수 있다. 이것은 일관성 회로소자(50)의 요구된 연산들을 단순화하는데, 그 이유는, 특별한 처리 디바이스에 의해 배타적 소유권을 보장하도록 임의의 단계들을 취할 필요가 없고, 대신에 단지 이후의 클린 연산들에 응답하면 되어 처리 디바이스들이 데이터가 갱신되어 있는 후 그 데이터의 일관된 뷰를 갖도록 보장하기 때문이다.

[0049] 도 4a 및 4b는, 처리 디바이스가, 조합된 소프트웨어/하드웨어 메카니즘을 사용하여 캐시 일관성을 구현하는 특별한 메모리 영역내의 데이터를 갱신하고 싶을 때, 일 구성 예에서 해당 처리 디바이스내에서 수행된 단계들을 도시하는 흐름도를 제공한다.

[0050] 단계 200에서, 갯(get) 락 명령은, 원하는 메모리 영역에 관해 처리 회로소자에 의해 실행된다. 이것은 그 메모리내의 락들(80)이 액세스되게 하고, 특히 관련 영역에 대한 락 표시 상태가 평가되게 한다. 락 표시가 이미 세트일 경우, 상기 처리는, 단계 205에서, 락 표시가 현재 클리어라고 판정되는 시간, 그 시점에서 락 표시는 상기 처리 디바이스에 의해 갯 락 명령을 실행함으로써 락이 얻어진 것을 나타내도록 설정될 수 있을 때까지 일시 중지된다. 상기 락이 얻어지면, 이것은, 사실상, 요구하는 처리 디바이스가, 단계 210에 의해 나타낸 것처럼, 원하는 메모리 영역내의 장소들에 데이터를 기록하기 위한 소유권 권한을 확립하였다는 것을 의미한다. 이 스테이지에서, 소프트웨어 프로토콜은, 그 밖의 처리 디바이스들이 당해 메모리 영역에 기록 액세스하는 것을 방지하는데, 그 이유는 그들의 그 밖의 처리 디바이스들 중 어느 하나가 갯 락 명령을 실행하기 위한 것이면, 락 표시가 이미 세트이라고 판정됨에 따라서, 그들이 해당 메모리 영역에 대한 락을 얻을 수 없을 것이기 때문이다. 따라서, 단계 200, 205, 210은, 데이터를 기록하는 권한을 확립하는 소프트웨어 프로토콜을 구현하는 것으로서

보여질 수 있다.

- [0051] 단계 210 후에, 단계 215에서, 처리 디바이스는, 락 걸린(locked) 메모리 영역내의 메모리 장소들에 대한 기록 데이터를 발생하기 위해서, 하나 이상의 기록 연산들을 행할 수 있고, 이때의 발생된 기록 데이터는 처리 디바이스의 로컬 라이트백 캐시에 기억된다. 이 처리는, 클린 연산에 대한 기동이 단계 220에서 검출될 때까지 계속될 수 있다.
- [0052] 일 구성 예에서, 클린 연산에 대한 기동이 다양한 이유로 일어날 수 있다. 예를 들면, 단순한 경우에 있어서, 별도의 클린 명령은 각 기록 명령과 관련되어도 되어, 수행된 처리는 처리 디바이스가 그 락을 얻고 나서, 단일의 기록 연산을 원하는 메모리 장소에 대해 수행하기 위한 것이고, 그 후 클린 명령이 해당 기록 데이터에 관해 클린 연산을 기동하도록 실행되기 위한 것이다. 그렇지만, 또 하나의 구성 예에서는, 락 걸린 메모리 영역내의 메모리 어드레스들에 관해 일련의 기록 연산들을 함께 분류하는 것이 가능할 수도 있고, 이때의 단일의 클린 연산은 그 기록 연산들이 수행된 후 기동된다. 예를 들면, 이러한 기능성을 달성하기 위한 하나의 적절한 메카니즘은, 공통으로 소유한 동시계류중인 미국특허출원 번호 15/501,278에 기재된 메카니즘이고, 이로써 그 메카니즘의 전체 내용은, 참조로 포함되어 있다. 특히, 거기에 정의된 종료 명령은 클린 명령으로서 사용되어 클린 연산을 기동할 수 있고, 원하는 경우, 관련된 시작 명령은, 종료 명령에 접할 때 클린될 필요가 있는 기록 데이터를 갖는 기록 명령들의 시퀀스의 시작을 확인하는데 사용될 수 있다. 별도의 스토리지 구조는, 그 종료 명령에 응답하여 클린될 필요가 있는 메모리 장소들을 추적하게 유지될 수 있다.
- [0053] 단계 220에서 생길 수 있는 기동의 또 추가의 예로서, 이것은, 단계 215에서 수행된 기록 연산에 의해 생성된 갱신된 기록 데이터를 갖는 캐시 엔트리를 표적으로 삼는 퇴거 이벤트이어도 된다. 퇴거 이벤트는 예를 들면, 로컬 캐시내의 용량 문제들로 인해 생길 수도 있고, 여기서 빅타임 엔트리는 새로운 데이터가 상기 캐시에 할당될 공간을 만드는데 선택될 필요가 있다. 그 빅타임 엔트리가 단계 215에서 생성된 기록 데이터를 갖는 엔트리일 때, 이것은 단계 220에서 클린 연산을 기동할 수 있다. 일련의 기록 연산들이 시작 명령 및 종료 명령과 관련되는 앞선 구성 예를 이용하면, 상기 영??을 받은 캐시 엔트리들 중 하나에 관해 퇴거는, 클린연산을 적어도 해당 엔트리에 관해 기동시키는데 사용될 수 있거나, 원하는 경우, 퇴거 기동이 발생할 때 상기 시작 명령과 종료 명령의 사이에서 기록 데이터가 생성되어 있는 엔트리들 전부를 클리닝시킬 수 있다.
- [0054] 기동 이벤트가 단계 220에서 일어날 때, 처리는 단계 225로 진행되어, 메모리 콘텐츠를 갱신시켜 기록 연산(들)에 의해 생성된 상기 갱신된 데이터를 반영하도록, 락 걸린 메모리 영역에 관해 생성되어 있는 기록 데이터에 대해 클린 연산이 행해진다.
- [0055] 앞서 설명된 것처럼, 일관성 회로소자(50)는, 상기와 같은 클린 연산의 존재에 응답하여, 메모리에 기록중인 기록 데이터를 처리 디바이스들 전체에 보이게 되도록 보장하기 위해서 그 밖의 처리 디바이스들의 로컬 캐시들에서의 임의의 관련 엔트리들에 관해 무효화 연산들을 수행할 것이다. 단계 230에서, 요구된 메모리 영역에 대한 락을 얻은 처리 디바이스는, 일관성 회로소자로부터 기록 데이터가 그 밖의 처리 디바이스들에게 보인다는 확인을 기다린다. 다중 캐시 엔트리들이 클린 처리동안에 클린되어도 되는 상황들에 있어서, 단계 230에서는, 관련 기록 데이터 전부가 그 밖의 처리 디바이스들에게 보이고 있다는, 즉 요구된 무효화 연산들 전부가 수행되어 있다는 확인을 기다릴 것이다.
- [0056] 이러한 확인을 단계 230에서 수신하면, 클린 연산은 완료되고, 단계 235에서는 수행될 기록 연산들 또는 클린 연산이 더 있는지를 판정한다. 이것은, 예컨대, 락을 해제시키는 실행을 갖는 명령에 앞서 명령 시퀀스에서의 하나 이상의 추가의 기록 명령들이나 클린 명령들이 있는 경우라고 판정될 수 있다. 수행될 적어도 하나 더 기록 연산이나 클린 연산이 있을 경우, 처리는 단계 220으로 진행되어 또 하나의 클린 연산에 대한 기동이 있는지를 판정하고, 기동이 없을 경우 처리는 단계 215로 복귀되어 하나 이상의 추가의 기록 연산들을 수행한다.
- [0057] 단계 235에서 더 이상 수행될 기록 연산 또는 클린 연산이 없는지를 판정하면, 그 후, 처리 회로소자는, 단계 240에서 메모리 영역에 대한 락을 해제하는 해제(release) 락 명령을 실행하도록 배치되어도 된다. 이것은, 해당 영역에 대한 락 표시를 클리어 상태로 리셋하기 위해서 락 스토리지(80)에서의 적절한 엔트리에의 액세스를 수행함으로써 이루어진다.
- [0058] 도 5는 일례에서 일관성 회로소자(50)의 연산을 도시하는 흐름도다. 단계 300에서, 일관성 회로소자(50)는, 도 4a 및 4b를 참조하여 앞서 설명된 단계들을 수행중인 처리 디바이스로부터의 클린 연산의 통지를 기다린다. 클린 연산의 통지시, 일관성 회로소자에 의해, 메모리 콘텐츠를 갱신시키도록 단계 305에서 관련된 기록 데이터는 메모리(70)에 전파된다. 병렬로, 처리가 단계 310으로 진행되어, 만약에 있다면, 그 밖의 캐시들이 클린 연산과

관련된 메모리 어드레스에 대한 데이터를 기억할 수도 있는지를 특정하기 위해서 상기 디렉토리(60)를 참조한다. 그 후, 스누프 제어기(55)는, 단계 315에서 디렉토리 정보로 나타낸 각 캐시에 무효화 요구를 보내도록 배치된다. 무효화 요구는, 당해 메모리 어드레스를 특정하고, 해당 무효화 요구를 수신하는 로컬 캐시가 검색연산을 수행하게 하여 실제로 해당 메모리 어드레스에서의 데이터의 복사본을 캐시하는 엔트리가 있는지를 확인하여, 있을 경우 해당 엔트리는 무효화될 것이다. 이것은, 관련된 처리 디바이스가 해당 메모리 어드레스에서의 데이터를 요구하면 더 이상 로컬 캐시내의 히트를 수신하지 않고, 대신에 정확한 데이터가 메모리(70)로부터 검색되도록, 보장할 것이다.

- [0059] 단계 315에서의 상술한 예에서는 무효화 요구들이 스누프 제어기(55)에 의해 관련 로컬 캐시들에 보내지지만, 다른 해결방법에서는, 갱신 요구들은, 정확히 요구된 데이터를 반영하기 위해 로컬 캐시 콘텐츠를 갱신시키도록, 새로운 기록 데이터와 함께 보내질 수 있었다.
- [0060] 단계 320에서, 스누프 제어기(55)는, 요구된 무효화 전부가 발생하였다는 확인을 기다린다. 해당 확인이 수신되었다면, 단계 325에서, 스누프 제어기는, 클린 연산을 개시한 처리 디바이스에, 그 데이터가 지금 그 밖의 처리 디바이스들에 보인다는 것을 특정하도록 통지할 수 있다. 단계 310에서 상기 디렉토리가 그 밖의 캐시들 중 아무도 상기 요구된 메모리 어드레스에 대한 데이터를 기억하지 않을 수도 있는 것을 나타내면, 처리는 단계 310으로부터 단계 325로 직접 진행될 수 있다는 것을 주목해야 한다.
- [0061] 클린 연산이 다중 메모리 어드레스들에 관해 적용되는 경우들, 이를테면 이것은 앞서 언급된 시작 명령과 종료 명령을 사용할 때의 경우에, 무효화 요구들은 상기 갱신된 캐시 라인 콘텐츠의 각각에 대한 상기 요구된 캐시들에 보내질 수 있고, 단계 320에서 스누프 제어기는, 단계 325로 진행되기 전에 상기 갱신된 데이터 값들 전부에 관해 상기 요구된 무효화 전부가 발생하였다는 확인을 기다린다.
- [0062] 단계 310 내지 325는, 전체적으로 일관성 회로소자(50)에 의해 개시되고, 소프트웨어 프로그래머에게 어떠한 가시성도 요구하지 않고, 특히 상기 처리 디바이스들 중 어느 하나에 의해 실행될 어떠한 명시적 명령들도 요구하지 않는다는 것을 알 것이다. 따라서, 단계 310 내지 325는 상기 처리 디바이스 전부에 기록 데이터를 보이게 하도록 하드웨어 프로토콜을 구현한다는 것을 알 수 있다.
- [0063] 단계 305에서 출력된 데이터를 실제로 메모리에 기록하는 것이 많은 클록 사이클이 걸릴 수도 있지만, 기록 데이터는, 전형적으로 상기 캐시로부터 인터커넥트(40)에 전파된 직후 일관성의 지점(예컨대, 메모리에 대한 인터페이스와 관련된 하나 이상의 기록버퍼들)에 도달할 것이다. 도 5에서는, 일관성의 지점에, 적어도 단계 310에서의 체크가 수행되었을 때까지 도달된다고 가정하고, 이에 따라 단계 325에 도달될 때까지, 기록 데이터는 아직 메인 메모리에 역으로 기록되지 않은 경우에도, 상기 일관성의 지점으로부터 이용 가능할 것이라고 이해된다.
- [0064] 상기 요구된 캐시 일관성 프로토콜을 구현하기 위해 소프트웨어 프로토콜과 하드웨어 프로토콜의 조합을 이용하는 상술한 메카니즘이 처리 디바이스간에 공유될 수도 있는 상기 메모리(70)의 메모리 영역 전부에 관해 구현될 수 있지만, 다른 예에서 이러한 해결방법은 일부의 영역들에 사용되어도 되지만, 풀 하드웨어 캐시 일관성은 원하는 경우 기타의 영역들에 대해 구현되어도 된다. 이것은, 도 6의 흐름도로 개략적으로 나타내어진다. 특히, 단계 350에서는, 당해 영역이 풀 하드웨어 일관성을 요구하는지를 판정한다. 요구할 경우, 처리는 단계 355로 진행되어 일관성 회로소자를 사용하여 캐시 일관성 프로토콜을 하드웨어로 완전히 구현한다. 특히, 처리 디바이스들 전부에 갱신된 데이터가 보이도록 보장하기 위해서, 클린 연산에 응답하여 상술한 단계들을 구현할 뿐만 아니라, 임의의 처리 디바이스가 기록 연산을 수행하기 전에, 기록 연산을 수행하고 싶은 처리 디바이스가 해당 처리 디바이스에 대해 배타적인 것으로서 표시되는 캐시된 복사본을 갖도록 보장하기 위해 로컬 캐시들에 관해 일관성 액션들을 수행하기도 하는, 기존의 스누프 메카니즘들을 사용할 수 있다. 따라서, 이러한 해결방법은, 데이터를 기록하는 권한의 확립이 일관성 회로소자(50)에서 수행한 하드웨어 프로토콜내에 내장되는 것을 가능하게 한다. 공지된 하드웨어 캐시 일관성 기법은, 단계 355에서 사용될 수 있다.
- [0065] 그렇지만, 상기 영역이 풀 하드웨어 일관성을 요구하지 않으면, 처리는 단계 360으로 진행되어 그 영역에 기록할 권한은, 메모리(70)에서 락들(80)을 참조하여 앞서 설명된 소프트웨어 프로토콜 기술을 사용하여 관리된다. 그 후, 단계 365에서, 갱신된 기록 데이터가 처리 디바이스 전부에 보이게 하는데 요구된 단계들은, 앞서 설명된 하드웨어 프로토콜을 사용하여 관리되고, 일관성 회로소자(50)에 의해 구현된다.
- [0066] 도 7a 내지 7d는, 소프트웨어 프로토콜과 하드웨어 프로토콜을 조합으로 사용하여 캐시 일관성을 구현하는 상황들에 있어서 로컬 라이트백 캐시를 갖는 처리 디바이스에 의해 실행되어도 되는 명령 시퀀스들의 예를 도시한

것이다. 도 7a에 도시된 것처럼, 단일의 기록 명령과 관련 클린 명령은, 상기 갯 락 명령과 상기 해제 락 명령의 사이에서 실행되어도 된다. 다른 구성에서는, 도 7b에 도시된 것처럼, 앞서 언급된 시작 명령과 종료 명령을 사용하여 기록 명령들의 그룹을 특정할 수 있고, 이때의 종료 명령은 상기 일련의 기록 명령들에 의해 생성된 기록 데이터 전부에 관해 클린 연산을 개시하는데 사용된다. 도 7c 및 7d는 도 7a에 도시된 구성의 변형 예로, 특히 다중 기록 명령들 및 관련 클린 명령이 갯 락 명령과 해제 락 명령의 사이에서 실행되는, 추가의 다른 구성들을 도시한 것이다. 도 7d로부터, 특별한 기록 명령과 관련된 클린 명령이 반드시 상기 시퀀스에서의 기록 명령에 바로 후속할 필요는 없다는 것을 주목해야 한다.

[0067] 앞서 언급된 것처럼, 앞서 언급된 시작 명령과 종료 명령이 사용되는 구성 예에 있어서, 기억유닛은, 시작 명령과 종료 명령의 사이에서 처리 디바이스에 의해 수행된 기록 연산들의 표시들을 유지하도록, 이러한 시작 및 종료 명령을 포함하는 명령들의 시퀀스를 실행하는 상기 처리 디바이스와 관련지어 제공되어도 된다. 도 8은 FIFO 기억 회로소자(450)의 형태를 취하는 상기와 같은 기억유닛의 일 구성 예를 도시한 것이다.

[0068] 시작 명령과 종료 명령 사이에 끼어 있는 기록 명령에 접할 때, 처리 디바이스는, FIFO 제어(448)와, 이 기록 명령에 응답하여 행해지는 기록 연산의 표시가 FIFO 스토리지(446)의 엔트리에 추가되어야 하는 상기 관련된 라이트백 캐시에 대해, 가리킨다. 도 8은 FIFO 스토리지(446)의 4개의 엔트리가 현재, 기록 연산 A-D의 표시로 채워지는 구성 예를 도시한 것이다. 본 도면에 나타난 것처럼, 이 표시들은, 기록 연산들의 타겟 메모리 어드레스들(또는, 실제로, 라이트백 캐시에서의 대응한 엔트리를 특정하기에 충분한 일부 메모리 어드레스들만)을 포함할 수 있거나, 그 대신에, 대응한 엔트리가 라이트백 캐시에서 발견될 경우를 도시하는 캐시 블록 지시자들이어도 된다. FIFO 제어유닛(448)은, 요구된 대로, 엔트리들의 부가 및 엔트리들의 삭제(개별적으로 또는 FIFO의 전체 콘텐츠를)를 관리하는 FIFO 스토리지(446)의 전반적인 제어를 유지한다. FIFO 기억 회로소자(450)가, 라이트백 캐시에서의 특별한 데이터 항목이 FIFO 스토리지(446)에서의 대응한 엔트리를 갖는지에 관해 라이트백 캐시로부터의 질의에 응답하도록 요구되는 경우, FIFO 기억 회로소자(450)는 이 기능을 제공하도록 콘텐츠 어드레스 가능 방식으로 구성될 수 있다. 이러한 개별적인 질의가 수신될 수도 있을 때의 일례는, 라이트백 캐시로부터의 데이터 항목의 퇴거상에 있고, 이에 응답하여, FIFO 제어(448)는 FIFO 스토리지(446)에서의 대응한 엔트리를 제거시키도록 구성될 수 있다. 또는, 라이트백 캐시로부터의 콘텐츠 질의는, 캐시 일관성 메카니즘의 참여의 일부로서 상기 라이트백 캐시에 의해 수신된 스누프 질의에 응답하여 수신되어도 된다, 하지만 도 9를 참조하여 후술하듯이, 라이트백 캐시 자체는, FIFO 스토리지(446)에 이 콘텐츠 어드레스 가능 메모리 구성이 제공될 필요성을 피하므로 상기 처리 디바이스에 상대적으로 작은 부가로서 FIFO 스토리지(446)의 제공을 용이하게 하기 위해서, 대응한 엔트리가 현재 FIFO 스토리지(446)에 보유되는지 여부의 표시를 기억하도록 구성되어도 된다. 끝으로, 도 8에 도시된 것처럼, FIFO 제어(448)는, 한층 더, FIFO 스토리지(446)가 완전히 채워지는(이 때문에 추가의 엔트리들이 만들어질 수 없을)지를 감시하도록 구성된다. FIFO 스토리지(446)가 완전히 채워지는 경우에, FIFO 제어(448)는, 이 사실의 신호를 라이트백 캐시에 보내고 나서, 라이트백 캐시에서의 대응한 엔트리들을 메모리(70)에 재빠르게 역으로 기록시키고, 그 FIFO 스토리지(446)의 콘텐츠는 클리어된다.

[0069] 도 9는, 이 라이트백 캐시가 제어기(460)와 다수의 캐시 라인들(462)을 갖는 것을 도시하는, 라이트백 캐시(432)를 개략적으로 도시한 것이다. 캐시 제어기(460)는, 상기 관련된 처리 디바이스로부터 수신된 데이터 항목들과 제어신호들을 감시하고, 이때의 제어신호들은 처리 디바이스의 파이프라인과 FIFO 제어(448)와의 양쪽으로부터 수신된다. 캐시 라인들(462)은 처리 디바이스의 데이터 처리 연산에 있어서 처리 디바이스에서 사용한 데이터 항목들을 기억하도록 구성되고, 각 캐시 라인은 데이터 항목 자신들과 관련 상태 마커들(통상의 방식으로 더티/유효 등)의 양쪽을 기억한다. 그렇지만, 추가로, 여기서는 각 캐시 라인의 하나의 비트를, 클린 온 스누프(clean-on-snoop) 마커로서 취급될 수 있는 전용 마커(464)로서 정의하고, 캐시 라인들(462) 중 하나에서 히트하는 또 하나의 캐시/CPU로부터 스누프가 수신되면, 마커(464)의 존재는 해당 캐시 라인이 메모리에 대해 클린 되게 한다는 것을 주목한다. 캐시 라인의 퇴거는 단지 일례일 뿐이고, 따라서 상기 마커(464)는 클린 온 퇴거(clean-on-evict) 마커로서 보여질 수 있다. 도시된 구성에 있어서, 캐시 제어기(460)는 상기 시스템에서 그 밖의 캐시들로부터 수신된 스누프들을 감시하도록 구성되고, 상기 마커(464)가 세트인 캐시(432)의 캐시 라인(462)에 기억된 데이터 항목에 관한 또 하나의 캐시로부터의 스누프가 수신될 때, 캐시 제어기(460)는 즉시 해당 캐시 라인을 퇴거시키도록 구성된다. 또한, 이 이벤트는, FIFO 스토리지(446)에서의 임의의 대응한 콘텐츠가 클리어될 수 있도록 FIFO 제어(448)에 신호로 보내진다.

[0070] 도 10은 일례에서 처리 디바이스에 의해 실행되어도 되는 명령들의 시퀀스의 예를 도시한 것이다. 이 경우에는, 상기 시작 명령과 종료 명령을 사용하여, 상기 시스템내의 지속성의 지점을 형성하는 불휘발성 메모리에 대해 수행되는 일련의 기록들을 특정한다고 가정한다. 특히, 그 불휘발성 메모리는 불휘발성 바이트 어드레스 가능

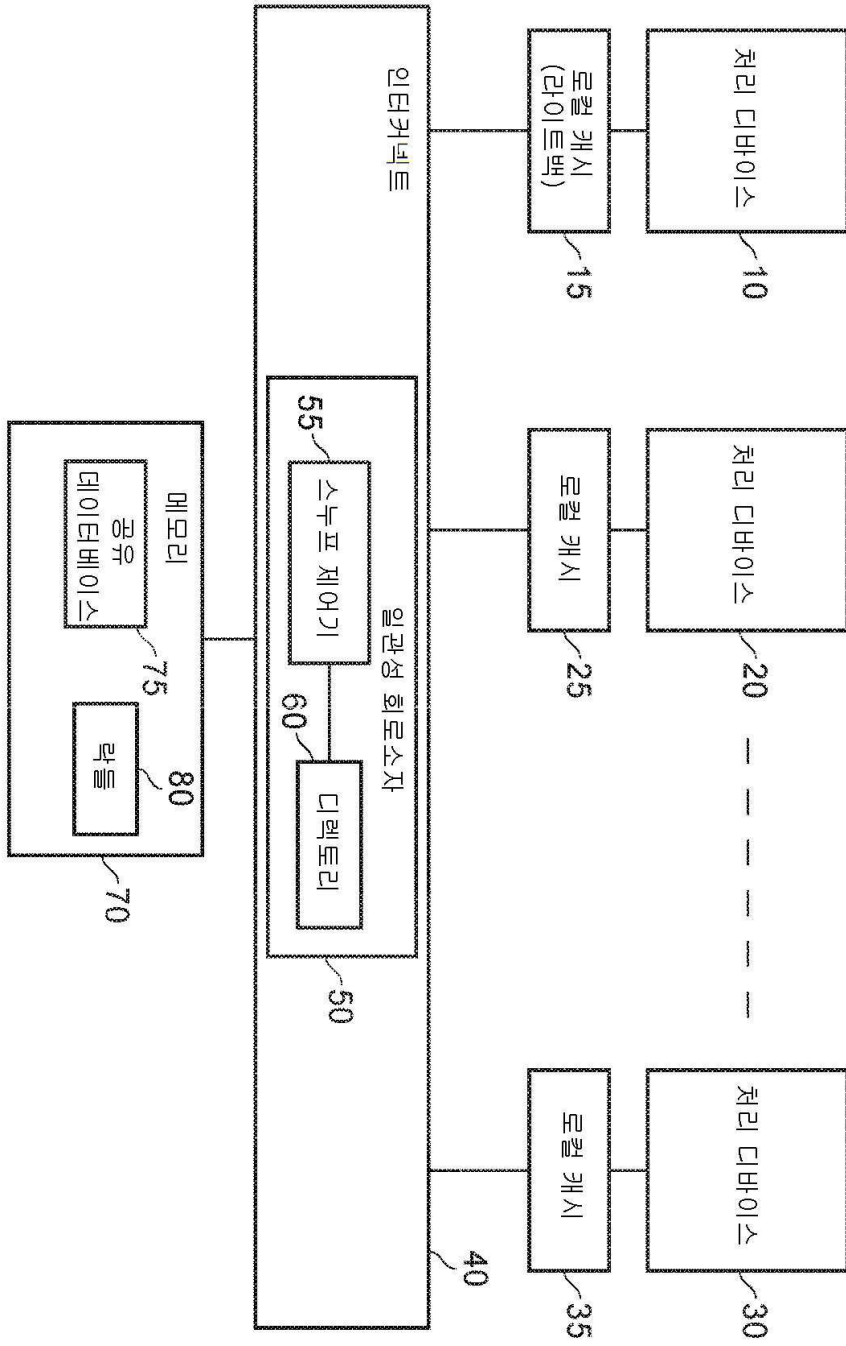
(NVBA) 메모리이어도 된다. (A라고 붙여진) 상기 시퀀스의 제1 부분은, 불휘발성 바이트 어드레스 가능 메모리에 기억되지 않은 데이터 항목들에 액세스하는 관독 명령, 기록 명령 및 데이터 처리 명령에 관계하는 다수의 명령들을 갖는다. 그 후, 상기 관독 명령, 데이터 처리 명령 및 기록 명령이 불휘발성 바이트 어드레스 가능 메모리에 기억되는 데이터 항목들에 (적어도 부분적으로) 관계하는 제1 영역(기록 세트)의 시작을 정의하는 nvba\_begin 명령(앞서 언급된 시작 명령의 일례)이 뒤이어 있다. 이 제1 영역(기록 세트)(B라고 붙여진)은 nvba\_end 명령(앞서 언급된 종료 명령의 일례)으로 종료된다. 그 후, 불휘발성 바이트 어드레스 가능 메모리에 기억된 데이터 항목들에 관계하지 않는 제2 세트의 명령들(A)이 뒤이어 있다. 끝으로, 불휘발성 기록 어드레스 가능 메모리에 기억된 데이터 항목들에 (적어도 부분적으로) 관계하는 관독 명령, 데이터 처리 명령 및 기록 명령(B)의 제2 영역(기록 세트)는, 제2 nvba\_begin 명령과 제2 nvba\_end 명령에 의해 정의된다. 상기 nvba\_begin 명령과 nvba\_end 명령의 제공에 의해 프로그래머는 기록 세트들을 정의할 수 있고, (서로에 대해) 본 기술들이 보존할 수 있는 오더링은, 액세스된 데이터 항목들이 불휘발성 메모리에 기억될 때에도, 라이트백 캐시에 의해 캐시되는 액세스를 보존할 수 있다.

[0071] 여기서 설명된 기술들에 따라, 상기 시작 및 종료 명령에 의해 특정된 각 블록 전에 갯 락 명령이 있고, 각 블록 후에 해제 락 명령이 있어서, 앞서 언급된 소프트웨어 프로토콜 메카니즘은, 기록 연산들 중 어느 하나를 수행하기 전에 기록할 권한을 처리 디바이스가 얻도록 보장하는데 사용될 수 있다. 그 결과, 그 후 일관성 회로소자에 의해 증분된 하드웨어 프로토콜은, 단지, 종료 명령에 의해 기동된 클린 연산이 발생하면 각종 그 밖의 처리 디바이스들에 해당 기록 데이터를 보이게 하도록, 보장하면 된다.

[0072] 본 출원에서, "...하도록 구성된"의 단어는, 장치의 요소가 상기 명시된 연산을 실시할 수 있는 구성을 갖는다는 것을 의미하는데 사용된다. 본 문맥에서, "구성"은, 하드웨어 또는 소프트웨어의 상호연결의 배치 또는 방식을 의미한다. 예를 들면, 상기 장치는 상기 명시된 연산을 제공하는 전용 하드웨어를 가져도 되거나, 프로세서 또는 다른 처리 디바이스는 그 기능을 행하도록 프로그래밍되어도 된다. "하도록 구성된"은, 상기 장치 요소가 임의의 방식으로 상기 명시된 연산을 제공하기 위해서 변경될 필요가 있다는 것을 암시하지는 않는다.

[0073] 여기서는 본 발명의 실시예들을 첨부도면을 참조하여 상세히 설명하였지만, 본 발명은 구체적인 실시예들에 한정되지 않고, 첨부된 청구항에서 기재된 것과 같은 본 발명의 범위 및 사상으로부터 벗어나지 않고 당업자가 변경, 부가 및 수정을 여러 가지로 실시할 수 있다는 것을 알 수 있다. 예를 들면, 본 발명의 범위로부터 벗어나지 않고 종속항의 특징들과 독립항의 특징들을 여러 가지로 조합할 수 있다.

도면  
도면1



도면2

일관성 회로소자에 의해 유지된 디렉토리

메모리 어드레스	캐시 N-1	캐시	
		1	0
100 어드레스 X	✓	---	x ✓
105 어드레스 Y	x		x ✓

60

도면3

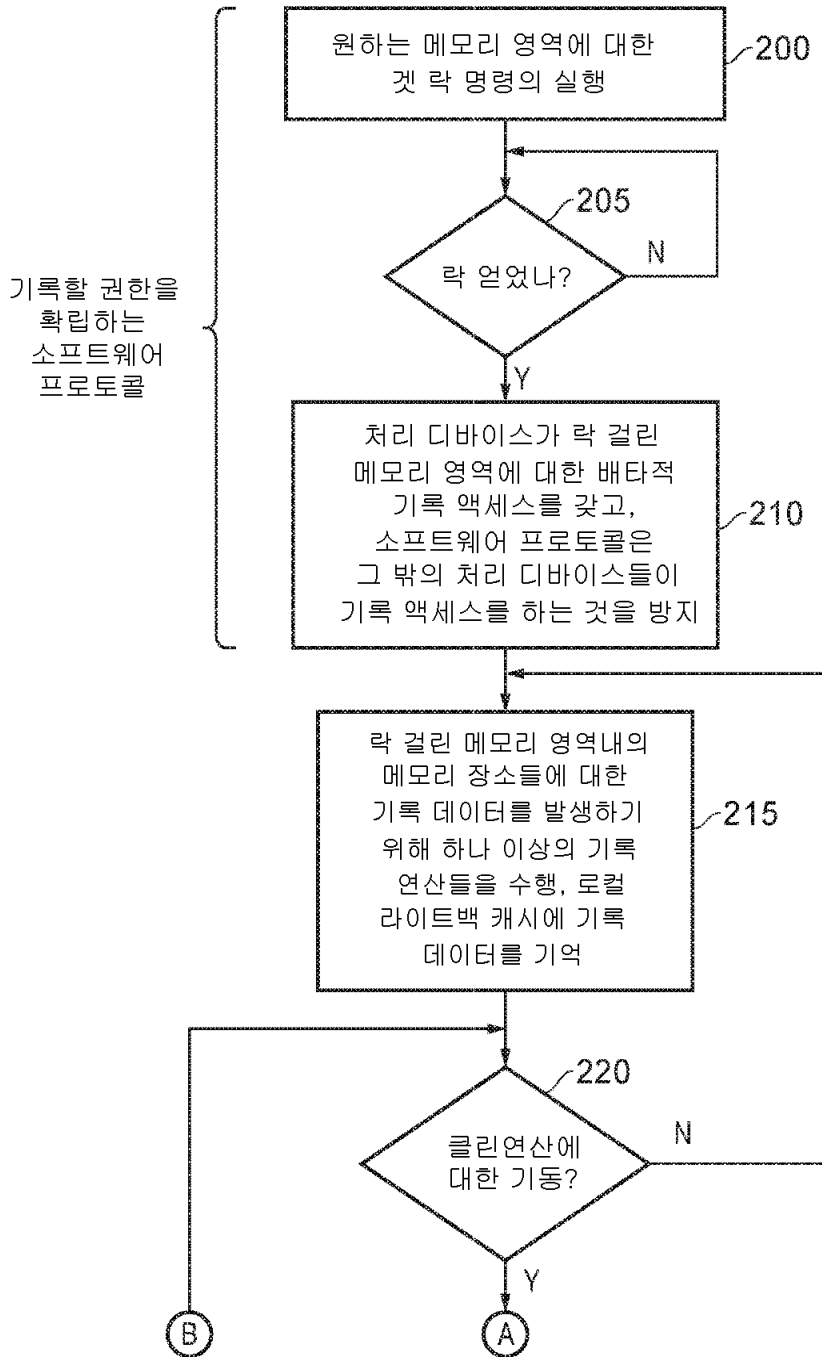
락 스토리지

메모리 영역	락 표시
110 영역 0	세트
115 영역 1	클리어
120 영역 2	클리어
⋮	⋮

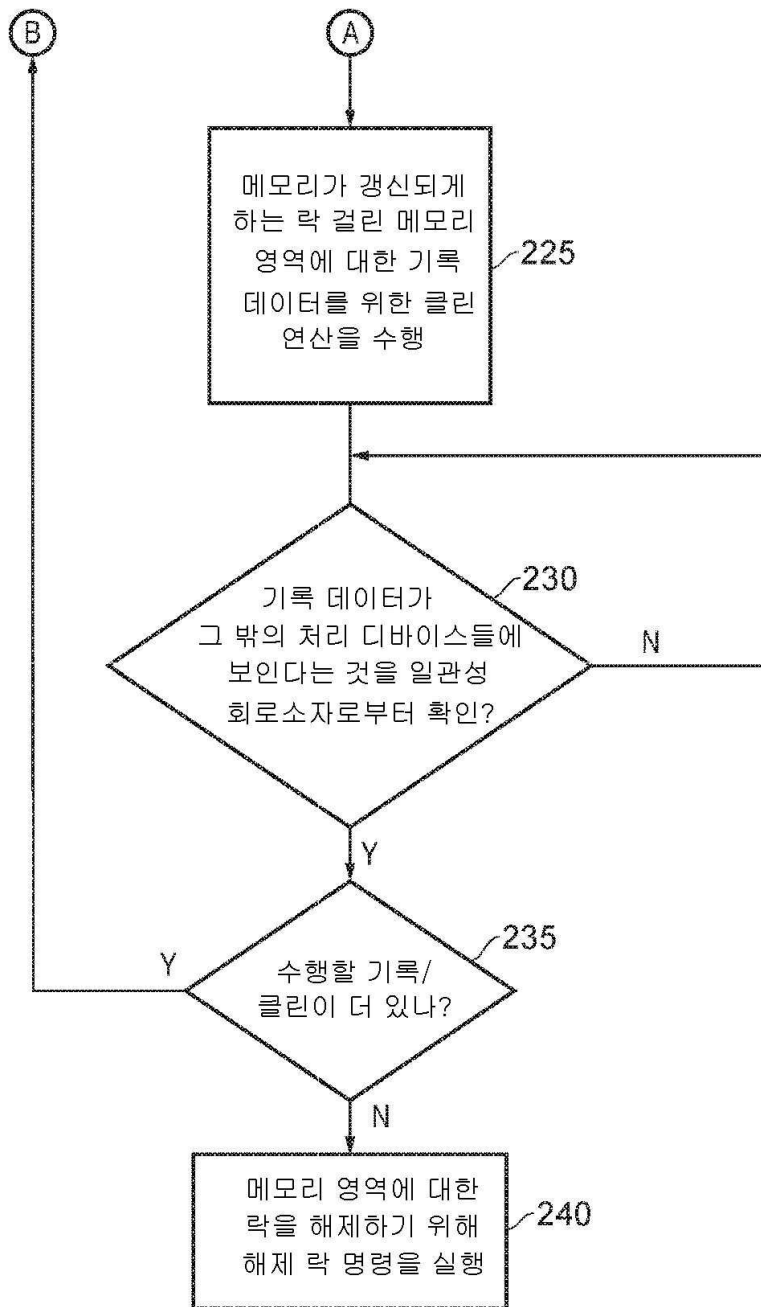
80

도면4a

처리 디바이스내

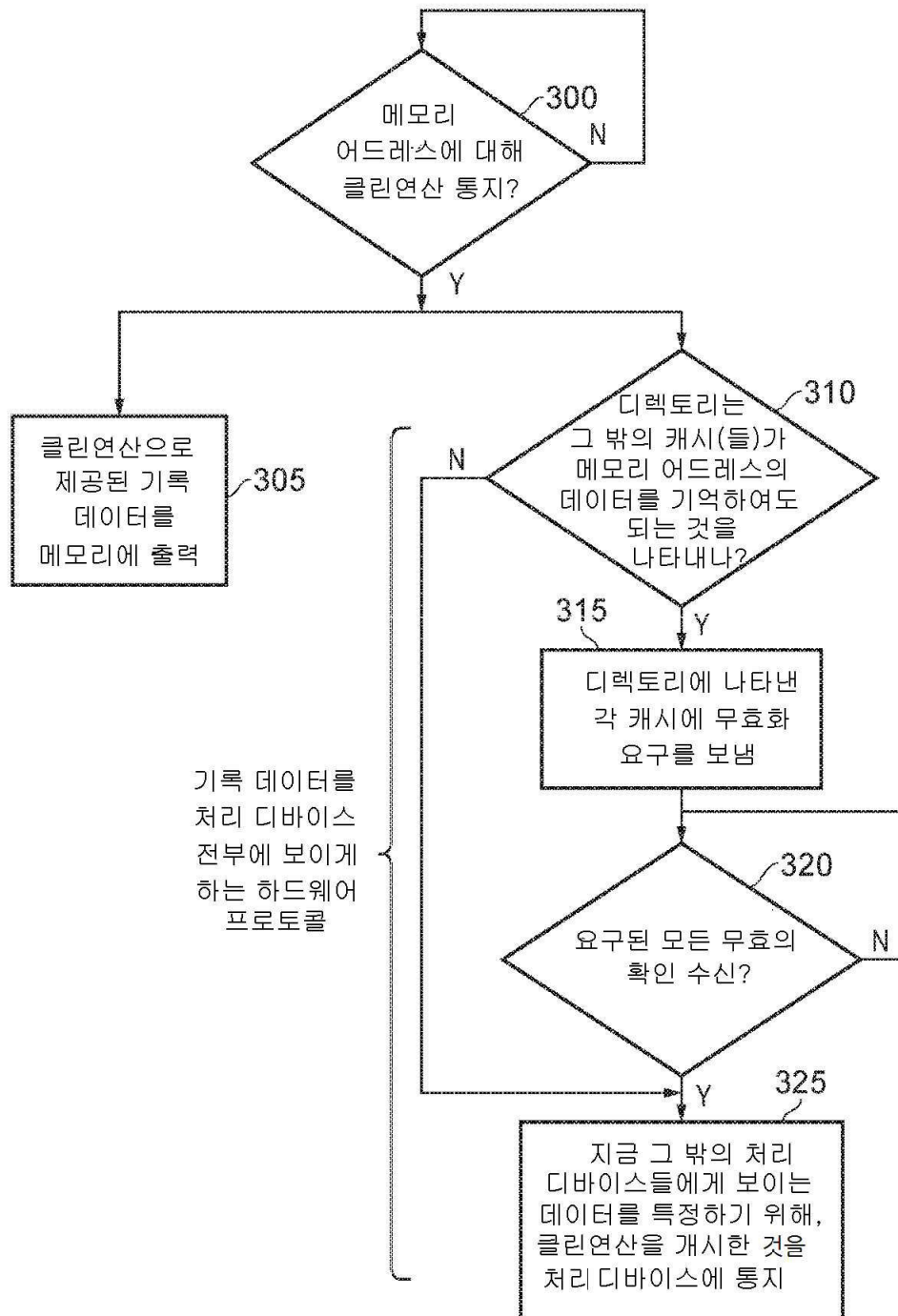


도면4b



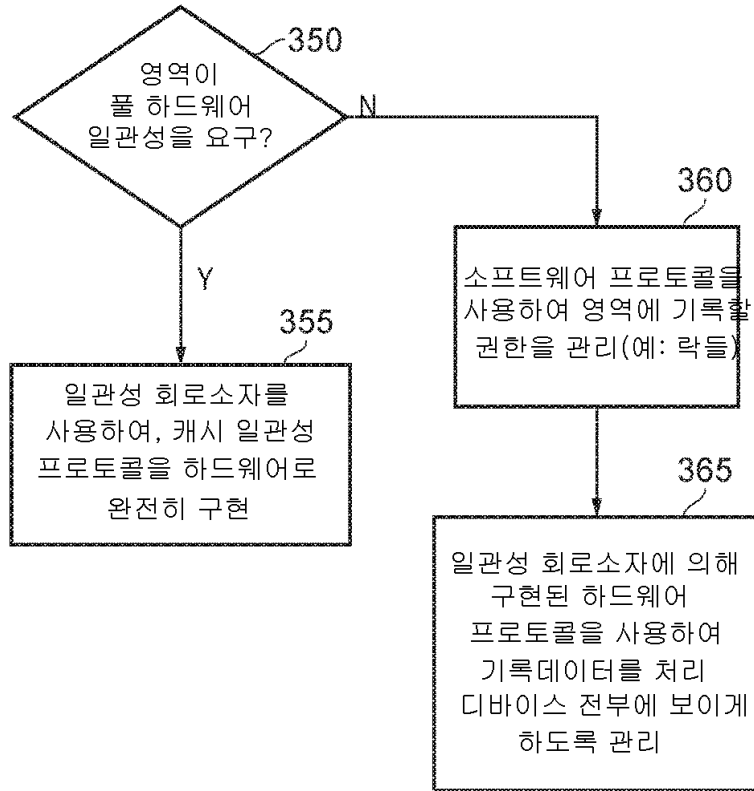
도면5

일관성 회로소자의 연산



도면6

메모리 영역 의존 일관성 메카니즘



도면7

GET LOCK  
WRITE  
CLEAN  
RELEASE LOCK

(a)

GET LOCK  
START  
WRITE  
WRITE  
.  
.  
END  
RELEASE LOCK

(b)

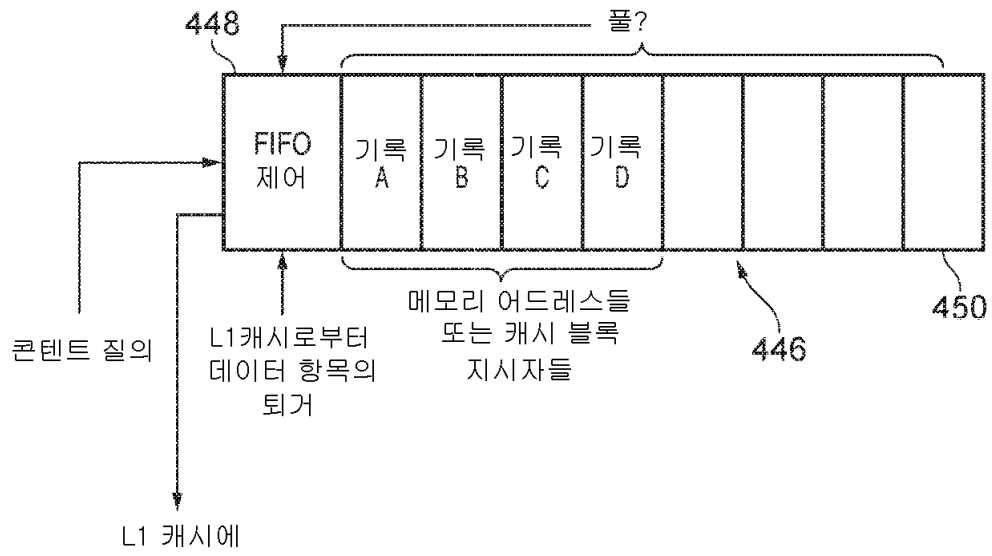
GET LOCK  
WRITE  
CLEAN  
WRITE  
CLEAN  
.  
.  
RELEASE LOCK

(c)

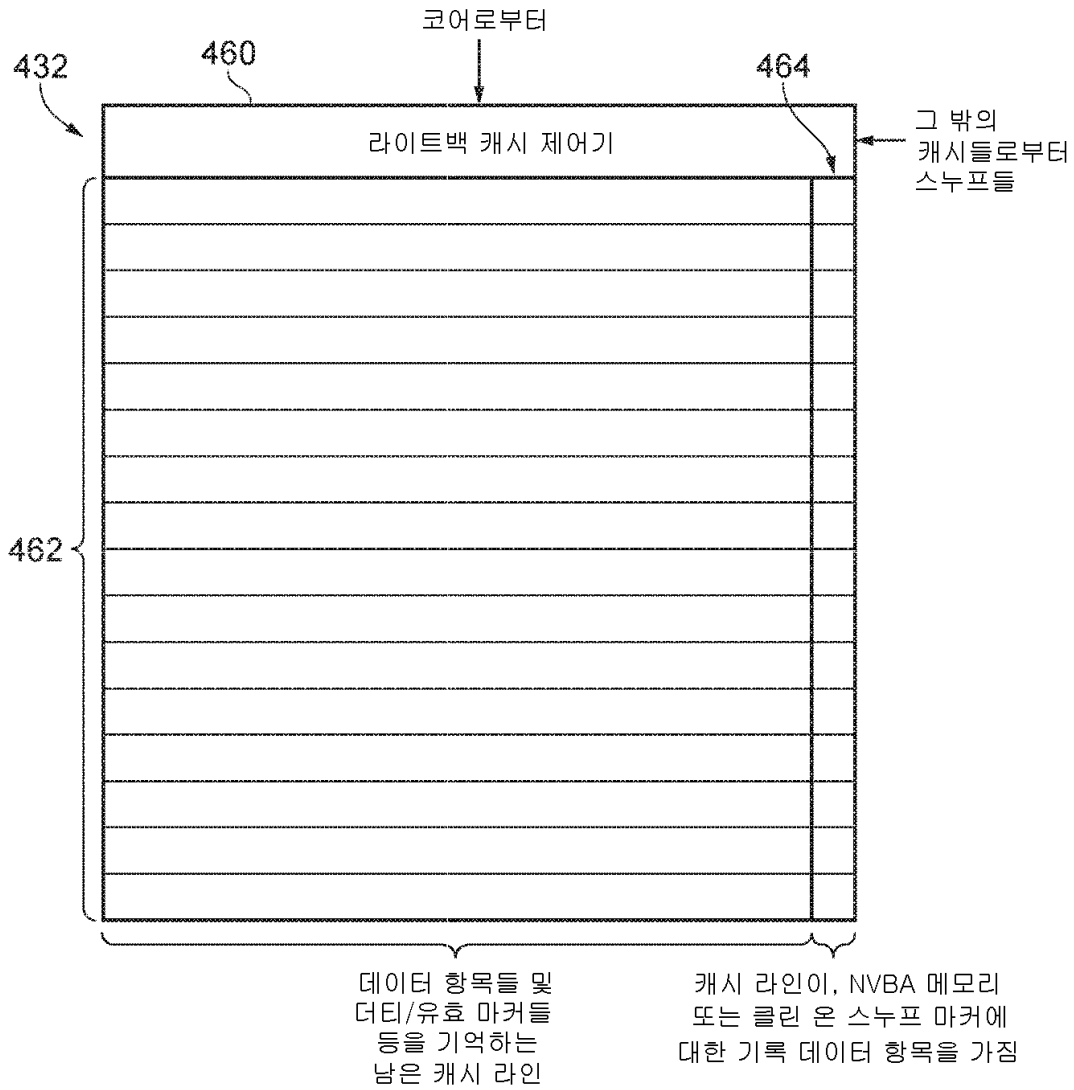
GET LOCK  
WRITE  
WRITE  
.  
.  
CLEAN  
CLEAN  
.  
.  
RELEASE LOCK

(d)

도면8



도면9



도면10

명령들의 시퀀스 예:

