(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2025년05월08일
(11) 등록번호 10-2805147
(24) 등록일자 2025년05월02일

(51) 국제특허분류(Int. Cl.)
G06F 12/0802 (2016.01) *G06F 12/02* (2018.01)
G06F 3/06 (2006.01)
(52) CPC특허분류
G06F 12/0802 (2013.01)
G06F 12/0246 (2013.01)
(21) 출원번호 10-2017-0021794
(22) 출원일자 2017년02월17일
심사청구일자 2021년12월20일
(65) 공개번호 10-2017-0098187
(43) 공개일자 2017년08월29일
(30) 우선권주장
15/048,080 2016년02월19일 미국(US)
(56) 선행기술조사문헌
US20140344503 A1*
(뒷면에 계속)
전체 청구항 수 : 총 18 항

(73) 특허권자
시게이트 테크놀로지 엘엘씨
미국 캘리포니아 94538 프레몬트 카토 로드 47488
(72) 발명자
시미오네스쿠, 호리아 크리스티안
미국 94404 캘리포니아 포스터 시티 피자로 레인 956
순다라라만, 바라크리쉬난
미국 78759 텍사스 오스틴 파이어오크 드라이브 7442
(뒷면에 계속)
(74) 대리인
특허법인(유)남아이피그룹, 특허법인 남앤남

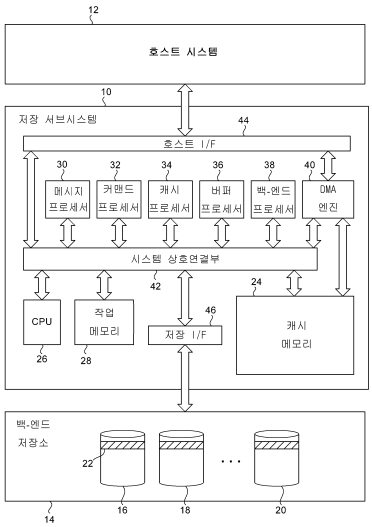
심사관 : 김결

(54) 발명의 명칭 **저장 서브시스템을 위한 연관적 및 원자적 라이트-백 캐싱 시스템 및 방법**

(57) 요 약

호스트로부터의 캐싱가능한 기입 요청에 대한 응답으로, 프리 리스트로부터 물리적 캐시 위치들이 할당되고, 대응하는 로직 어드레스들에 대해 임의의 판독 요청들이 계류중인지 여부에 상관없이 데이터 블록들이 그 캐시 위치들에 기입된다. 데이터가 기입된 이후, 그리고 또한, 임의의 판독 요청들이 대응하는 로직 어드레스들에 대하여 계류중인지 여부에 상관없이, 캐시 위치들을 로직 어드레스들과 연관시키기 위해 메타데이터가 업데이트된다. 유효 데이터를 갖는 각각의 캐시 위치들에 대하여 계류중인 데이터 액세스 요청들의 카운트가 유지되며, 캐시 위치는, 캐시 위치에 대하여 어떠한 데이터 액세스 요청들도 계류중이지 않다고 카운트가 표시하는 경우에만 프리 리스트로 리턴된다.

대 표 도 - 도1



(52) CPC특허분류
G06F 3/064 (2013.01)
(72) 발명자
네마와르카르, 샤산크
미국 78750 텍사스 오스틴 트리 벤드 드라이브 9702
킹, 래리 스테판
미국 78729 텍사스 오스틴 오스본 드라이브 8103
이쉬, 마크
미국 94552 캘리포니아 카스트로 밸리 그린리지 코트 18516
오라크, 샤이렌드라
미국 78750 텍사스 오스틴 애쉬톤 리지 9409

(56) 선행기술조사문헌
US6845426 B2
JP2013097416 A
KR1020130064518 A*
KR1020070048797 A*
US20040230737 A1*
US20020131310 A1*
*는 심사관에 의하여 인용된 문헌

명 세 서

청구범위

청구항 1

데이터 저장 서브시스템에서의 캐싱(caching)을 위한 방법으로서,

하나 또는 그 초과의 로직 어드레스들 및 상기 하나 또는 그 초과의 로직 어드레스들에 대응하게 기입될 하나 또는 그 초과의 데이터 블록들을 표시하는 기입 요청을 수신하는 단계;

상기 기입 요청에 대한 응답으로, 프리 리스트(free list)로부터의, 캐시 메모리 내의 하나 또는 그 초과의 물리적 위치들을 할당하는 단계;

상기 하나 또는 그 초과의 데이터 블록들을 상기 하나 또는 그 초과의 물리적 위치들에 저장하는 단계;

로직 어드레스에 대한 응답으로 해시 테이블(hash table) 슬롯을 결정하는 단계;

상기 해시 테이블 슬롯 내의 복수의 엔트리(entry)들 중 임의의 엔트리가 상기 로직 어드레스를 식별하는지 여부를 결정하는 단계;

상기 하나 또는 그 초과의 물리적 위치들을 식별하는 식별 정보를 하나 또는 그 초과의 데이터 구조들에 저장하는 단계;

상기 하나 또는 그 초과의 데이터 구조들에 대한 포인터(pointer)를 포함하도록 해시 테이블의 엔트리를 업데이트 하는 단계;

유효(valid) 데이터를 갖는 상기 캐시 메모리 내의 각각의 물리적 위치에 대하여 계류중인, 판독 요청들을 포함하는 데이터 액세스 요청들의 카운트를 유지하는 단계; 및

물리적 위치에 대하여 어떠한 데이터 액세스 요청들도 계류중이지 않다고 상기 카운트가 표시하는 경우, 상기 물리적 위치를 상기 프리 리스트로 리턴하는 단계를 포함하는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 2

제 1 항에 있어서,

상기 하나 또는 그 초과의 물리적 위치들을 할당하는 단계는, 상기 캐시 메모리에서의 상기 하나 또는 그 초과의 물리적 위치들의 순서에 상관없이, 상기 프리 리스트로부터 상기 하나 또는 그 초과의 물리적 위치들을 선택하는 단계를 포함하는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 3

제 1 항에 있어서,

상기 하나 또는 그 초과의 물리적 위치들을 할당하는 단계는, 상기 기입 요청과 연관된 상기 하나 또는 그 초과의 로직 어드레스들에 상관없이, 상기 프리 리스트로부터 상기 하나 또는 그 초과의 물리적 위치들을 선택하는 단계를 포함하는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 4

삭제

청구항 5

제 1 항에 있어서,

상기 하나 또는 그 초과의 물리적 위치들을 할당하는 단계는, 상기 하나 또는 그 초과의 물리적 위치들을 식별하는 정보를 포함하는 분산-수집 리스트(SGL; scatter-gather list)를 생성하는 단계를 포함하고,

상기 하나 또는 그 초과의 데이터 블록들을 상기 하나 또는 그 초과의 물리적 위치들에 저장하는 단계는,

상기 SGL을 직접 메모리 액세스(DMA; direct memory access) 엔진에 제공하는 단계; 및

상기 SGL에 대한 응답으로, 상기 DMA 엔진이 상기 하나 또는 그 초과의 데이터 블록들을 호스트 인터페이스로부터 상기 캐시 메모리로 전달하는 단계

를 포함하는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 6

제 1 항에 있어서,

상기 하나 또는 그 초과의 데이터 구조들은 링크된(linked) 리스트를 정의하는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 7

삭제

청구항 8

제 6 항에 있어서,

상기 방법은, 상기 해시 테이블 슬롯 내의 엔트리들 중 어떠한 엔트리도 상기 로직 어드레스를 식별하지 않는다고 결정하는 것에 대한 응답으로, 새로운 데이터 구조를 상기 링크된 리스트에 부가하는 단계를 더 포함하는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 9

제 8 항에 있어서,

상기 링크된 리스트 내의 상기 새로운 데이터 구조를 식별하는 정보를 최소 최근 사용 더티 리스트(least recently used dirty list)에 부가하는 단계를 더 포함하는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 10

제 1 항에 있어서,

상기 하나 또는 그 초과의 데이터 구조들 각각은 복수의 서브-구조들을 포함하며,

각각의 서브-구조는, 상기 캐시 메모리 내의 상기 물리적 위치들 중 하나를 식별하는 상기 식별 정보를 저장하도록 구성되고, 각각의 서브-구조는 추가로, 상기 식별 정보에 의해 식별되는 물리적 위치에 대하여 계류중인 데이터 액세스 요청들의 카운트를 저장하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 11

제 10 항에 있어서,

상기 각각의 서브-구조는 추가로, 상기 식별 정보에 의해 식별되는 물리적 위치가 더티 데이터를 포함하는지 여부를 표시하는 더티 표시자를 저장하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 방법.

청구항 12

데이터 저장 서브시스템에서의 캐싱을 위한 시스템으로서,

캐시 메모리; 및

프로세싱 시스템을 포함하며,

상기 프로세싱 시스템은,

하나 또는 그 초과의 로직 어드레스들 및 상기 하나 또는 그 초과의 로직 어드레스들에 대응하게 기입될 하나 또는 그 초과의 데이터 블록들을 표시하는 기입 요청을 수신하고;

상기 기입 요청에 대한 응답으로, 프리 리스트로부터의, 캐시 메모리 내의 하나 또는 그 초과의 물리적 위치들을 할당하고;

상기 하나 또는 그 초과의 데이터 블록들을 상기 하나 또는 그 초과의 물리적 위치들에 저장하고;

로직 어드레스에 대한 응답으로 해시 테이블 슬롯을 결정하고;

상기 해시 테이블 슬롯 내의 복수의 엔트리들 중 임의의 엔트리가 상기 로직 어드레스를 식별하는지 여부를 결정하고;

상기 하나 또는 그 초과의 물리적 위치들을 식별하는 식별 정보를 하나 또는 그 초과의 데이터 구조들에 저장하고;

상기 하나 또는 그 초과의 데이터 구조들에 대한 포인터를 포함하도록 해시 테이블의 엔트리를 업데이트하고;

유효 데이터를 갖는 상기 캐시 메모리 내의 각각의 물리적 위치에 대하여 계류중인, 판독 요청들을 포함하는 데이터 액세스 요청들의 카운트를 유지하고; 그리고

물리적 위치에 대하여 어떠한 데이터 액세스 요청들도 계류중이지 않다고 상기 카운트가 표시하는 경우, 상기 물리적 위치를 상기 프리 리스트로 리턴

하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

청구항 13

제 12 항에 있어서,

상기 프로세싱 시스템은, 상기 캐시 메모리에서의 상기 하나 또는 그 초과의 물리적 위치들의 순서에 상관없이, 상기 프리 리스트로부터 상기 하나 또는 그 초과의 물리적 위치들을 선택하도록 구성됨으로써 상기 하나 또는 그 초과의 물리적 위치들을 할당하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

청구항 14

제 12 항에 있어서,

상기 프로세싱 시스템은, 상기 기입 요청과 연관된 상기 로직 어드레스들에 상관없이, 상기 프리 리스트로부터 상기 하나 또는 그 초과의 물리적 위치들을 선택하도록 구성됨으로써, 상기 하나 또는 그 초과의 물리적 위치들을 할당하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

청구항 15

삭제

청구항 16

제 12 항에 있어서,

상기 프로세싱 시스템은, 상기 하나 또는 그 초과의 물리적 위치들을 식별하는 정보를 포함하는 분산-수집 리스트(SGL)를 생성하도록 구성됨으로써 상기 하나 또는 그 초과의 물리적 위치들을 할당하도록 구성되고,

상기 프로세싱 시스템은,

상기 SGL을 직접 메모리 액세스(DMA) 엔진에 제공하고,

상기 SGL에 대한 응답으로, 상기 DMA 엔진에 의해 상기 하나 또는 그 초과의 데이터 블록들을 호스트 인터페이스로부터 상기 캐시 메모리로 전달

하도록 구성됨으로써 상기 하나 또는 그 초과의 데이터 블록들을 상기 하나 또는 그 초과의 물리적 위치들에 저장하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

청구항 17

제 12 항에 있어서,

상기 하나 또는 그 초과의 데이터 구조들은 링크된 리스트를 정의하는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

청구항 18

삭제

청구항 19

제 17 항에 있어서,

상기 프로세싱 시스템은, 상기 해시 테이블 슬롯 내의 엔트리들 중 어떠한 엔트리도 상기 로직 어드레스를 식별하지 않는다고 결정하는 것에 대한 응답으로, 새로운 데이터 구조를 상기 링크된 리스트에 부가하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

청구항 20

제 19 항에 있어서,

상기 프로세싱 시스템은 추가로, 상기 링크된 리스트 내의 상기 새로운 데이터 구조를 식별하는 정보를 최소 최근 사용 더티 리스트에 부가하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

청구항 21

제 12 항에 있어서,

상기 하나 또는 그 초과의 데이터 구조들 각각은 복수의 서브-구조들을 포함하며,

각각의 서브-구조는, 상기 캐시 메모리의 상기 물리적 위치들 중 하나를 식별하는 상기 식별 정보를 저장하도록 구성되고, 각각의 서브-구조는 추가로, 상기 식별 정보에 의해 식별되는 물리적 위치에 대하여 계류중인 데이터 액세스 요청들의 카운트를 저장하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

청구항 22

제 21 항에 있어서,

상기 각각의 서브-구조는 추가로, 상기 식별 정보에 의해 식별되는 물리적 위치가 더티 데이터를 포함하는지 여부를 표시하는 더티 표시자를 저장하도록 구성되는, 데이터 저장 서브시스템에서의 캐싱을 위한 시스템.

발명의 설명

기 술 분 야

[0001] [0001] 본 발명은 일반적으로 데이터 저장 시스템들에 관한 것으로, 더 구체적으로는, 캐시(cache) 메모리를 갖는 데이터 저장 시스템들에 관한 것이다.

배 경 기 술

[0002] [0002] 캐시 메모리를 갖는 데이터 저장 서브시스템은, 호스트 컴퓨터 시스템과 백-엔드(back-end) 데이터 저장소, 이를테면 디스크 드라이브 어레이들 또는 비-휘발성(예컨대, 플래시) 메모리 어레이들 간에 빈번하게 액세스되는 데이터에 대한 버퍼로서 기능할 수 있다. 캐시 메모리를 갖는 데이터 저장 서브시스템은, 예를 들어, 호스트와 백-엔드 데이터 저장소 간의 낮은 데이터 액세스 레이턴시(latency)를 촉진시키는 가속기 카드(accelerator card)의 형태로 있을 수 있다. 캐싱(caching) 시스템은, 캐시 메모리에 저장된 데이터가 액세스되는 빈도를 결정할 수 있고, 오직 가장 빈번하게 액세스되는 데이터만을 캐시 메모리 내에 유지하는 한편, 가장 덜 빈번하게 액세스되는 데이터를 퇴거(evict)시킨다.

[0003] [0003] 캐싱 시스템에서, 데이터를 기입 또는 판독하기 위한 호스트로부터의 요청을 수신하는 것과 기입 또는 판독 동작을 완료하는 것 간의 시간의 양인 레이턴시를 최소화하는 것이 바람직하다. 레이턴시에 영향을 미치는 속성은, 호스트에 의해 식별된 로직 어드레스들을 캐시 메모리 내의 물리적 저장 위치들로 캐싱 시스템이 변환하는데 요구되는 검색(look-up) 시간, 및 그에 후속하여, 캐싱 시스템이 그 저장 위치들로 또는 그 저장 위치

들로부터 데이터를 전달하는데 요구되는 시간이다.

[0004] [0004] 일반적인 타입의 캐싱은 "라이트 백(write back)"으로 알려져 있는데, 이는, 기입 요청에서 호스트로부터 수신된 데이터가 캐시 메모리에 먼저 저장되어 호스트가 판독하는데 이용가능하게 한 다음에, 나중에, 레이턴시에 대해 덜 영향을 미치는 카피 동작을 조건들이 가능하게 할 때 백-엔드 저장소에 카피된다. 캐시 메모리에 저장되었지만 아직 백-엔드 저장소에 카피되지 않은 데이터는 일반적으로 "더티(dirty)" 데이터로 지칭된다. 라이트-백 캐싱의 단점은, 더티 데이터가 전력 중단들과 같은 시스템 장애들로 인한 손실에 취약할 수 있다는 것이다. 유사하게, 캐시 메모리에 데이터를 저장하는 프로세스를 중단시키는 그러한 장애들로부터 데이터 손실이 초래될 수 있으며, 이는 바람직하지 않게, 제 1 기입 요청과 연관된 데이터의 일부 및 후속 기입 요청과 연관된 데이터의 일부를 포함하는 저장 위치들의 시퀀스(sequence)를 초래할 수 있다.

[0005] [0005] 캐싱 시스템은, 직접-맵핑형(direct-mapped), 완전 연관적(fully associative), 또는 이러한 타입들의 혼합일 수 있다. 직접-맵핑형 캐싱 시스템에서, 데이터 블록의 로직 어드레스들은 캐시 메모리 내의 오직 하나의 어드레스 또는 물리적 위치에 맵핑되고, 이러한 어드레스 또는 물리적 위치에 데이터 블록이 저장될 수 있다. 예를 들어, 물리적 어드레스는 모듈식(modular) 산술: 캐시 어드레스 = 로직 어드레스 MOD(캐시 메모리 내의 어드레스들의 수)에 의해 컴퓨팅될 수 있다. 대조적으로, 완전 연관적 캐싱 시스템에서, 캐시 블록은 캐시 메모리 내의 임의의 물리적 메모리 위치에 저장될 수 있다. 완전 연관적 캐싱 시스템은 일반적으로, 메모리 사용 효율성 및 히트 레이트(hit rate)에 이롭다. 그러나, 완전 연관적 캐싱 시스템의 단점은, 식별된 로직 어드레스들을 캐시 메모리 내의 물리적 저장 위치들로 변환하기 위한 검색 프로세스가 느릴 수 있고 그에 따라 레이턴시에 영향을 미칠 수 있다는 것이다.

발명의 내용

[0006] [0006] 본 발명의 실시예들은 캐싱 시스템 및 동작 방법에 관한 것이다. 일 예시적인 실시예에서, 캐싱 시스템은, 캐시 메모리, 및 방법을 달성하도록 구성되거나 또는 프로그래밍되는 프로세싱 시스템을 포함한다.

[0007] [0007] 일 예시적인 실시예에서, 방법은, 하나 또는 그 초과의 로직 어드레스들 및 하나 또는 그 초과의 로직 어드레스들에 대응하게 기입될 하나 또는 그 초과의 데이터 블록들을 표시하는 기입 요청을 수신하는 단계; 기입 요청에 대한 응답으로, 프리 리스트(free list)로부터의 캐시 메모리 내의 하나 또는 그 초과의 물리적 위치들을 할당하는 단계; 하나 또는 그 초과의 로직 어드레스들에 대해 임의의 판독 요청들이 계류중(pending)인지 여부에 상관없이 하나 또는 그 초과의 데이터 블록들을 하나 또는 그 초과의 물리적 위치들에 저장하는 단계; 하나 또는 그 초과의 데이터 블록들이 하나 또는 그 초과의 물리적 위치들에 저장된 이후, 그리고 하나 또는 그 초과의 로직 어드레스들에 대하여 임의의 판독 요청들이 계류중인지 여부에 상관없이, 하나 또는 그 초과의 물리적 위치들을 하나 또는 그 초과의 로직 어드레스들과 연관시키기 위해 메타데이터(metadata)를 업데이트하는 단계; 유효(valid) 데이터를 갖는 캐시 메모리 내의 각각의 물리적 위치에 대하여 계류중인, 판독 요청들을 포함하는 데이터 액세스 요청들의 사용 카운트를 유지하는 단계; 및 물리적 위치에 대하여 어떠한 데이터 액세스 요청들도 계류중이지 않다고 사용 카운트가 표시하는 경우, 물리적 위치를 프리 리스트로 리턴하는 단계를 포함한다.

도면의 간단한 설명

[0008] [0008] 도 1은 본 발명의 예시적인 실시예에 따른 캐싱 시스템을 예시하는 블록도이다.

[0009] [0009] 도 2는 예시적인 실시예에 따른, 캐시 메모리의 일반화된 맵을 예시하는 개념도이다.

[0010] [0010] 도 3은 예시적인 실시예에 따른, 데이터 구조들 간의 예시적인 관계들을 예시하는 데이터 관계도이다.

[0011] [0011] 도 4는 예시적인 실시예에 따른, 예시적인 캐시 세그먼트 링크된 리스트(cache segment linked list) 및 연관된 예시적인 버퍼 블록들을 예시하는 다른 데이터 관계도이다.

[0012] [0012] 도 5는 예시적인 실시예에 따른 예시적인 분산-수집(scatter-gather) 리스트들을 예시한다.

[0013] [0013] 도 6a는 예시적인 실시예에 따른 기입 동작을 예시하는 흐름도이다.

[0014] [0014] 도 6b는, 도 6a의 흐름도에 연속되는 것이다.

[0015] [0015] 도 7a는 예시적인 실시예에 따른 판독 동작을 예시하는 흐름도이다.

[0016] 도 7b는, 도 7a의 흐름도에 연속되는 것이다.

[0017] 도 8은 예시적인 실시예에 따른, 시스템의 동작 방법을 예시하는 흐름도이다.

[0018] 도 9는 예시적인 실시예에 따른 플러시(flush) 동작을 예시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0009] [0019] 도 1에 예시된 바와 같이, 본 발명의 예시적인 또는 예증적인 실시예에서, 저장 서브시스템(10)은 호스트 시스템(12)과 백-엔드 저장소(14) 간에 캐싱을 제공한다. 호스트 시스템(12)은, 임의의 타입의 종래의 컴퓨터 시스템 또는 컴퓨터 시스템들의 네트워크일 수 있다. 백-엔드 저장소(14)는 임의의 타입의 종래의 데이터 대용량 저장 시스템일 수 있다. 예를 들어, 백-엔드 저장소(14)는, 다수의 물리적 데이터 저장 디바이스들(16, 18, 20 등)의 어레이를 포함할 수 있다. 그러한 물리적 데이터 저장 디바이스들(16, 18, 20 등)의 예들은, 디스크 드라이브들, 플래시 메모리 모듈들, 및 다른 타입들의 비-휘발성 데이터 저장 디바이스들을 포함한다.

[0010] [0020] 본원에 설명된 예시적인 실시예에서, 백-엔드 저장소(14) 내의 다수의 물리적 데이터 저장 디바이스들(16, 18, 20 등)의 어레이는, "RAID" 또는 "독립적인(또는 저렴한) 디스크들의 리던던트 어레이(redundant array of independent (or inexpensive) disks)" 산하에서 일반적으로 언급되는 원리들 중 하나 또는 그 초과를 따른다. 예를 들어, 스트라이핑(striping)으로 알려져 있는 일반적인 RAID 원리에 따라, 백-엔드 저장소(14)는 스트라이프들(22)의 단위들로 데이터를 저장할 수 있다. 물리적 데이터 저장 디바이스들(16, 18, 20 등) 각각은, 각각의 스트라이프(22)의 일 부분을 저장한다. 백-엔드 저장소(14)는 임의의 수의 물리적 저장 디바이스들(16, 18, 20 등)을 포함할 수 있다. (도 1에서의 생략 부호("...")는, 어레이에 포함되어 있지만 명확화의 목적들을 위해 개별적으로 도시되지 않은 부가적인 물리적 데이터 저장 디바이스들을 표시한다.) 스트라이핑과 같은 RAID 원리들은 당업자들에 의해 잘 이해되므로, 그러한 원리들은 본원에서 더 상세하게 설명되지 않는다. 또한, 디스크들과 같은 다수의 물리적 데이터 저장 디바이스들(16, 18, 20 등)을 포함한다는 것 또는 RAID 원리들을 따른다는 것 등과 같은 백-엔드 저장소(14)의 전술한 설명은, 단지 본 발명이 연관될 수 있는 예시적인 데이터 저장 환경 또는 컨텍스트(context)를 설명하는 목적만을 위해 제공되며, 제한하는 것으로 의도되지 않는다. 예를 들어, 본원에서 사용되는 바와 같은 용어 "로직 어드레스"는, 임의의 특정한 물리적 데이터 저장 환경, 구성, 방식 등으로 본 발명을 제한하는 것으로 해석되어서는 안된다.

[0011] [0021] 예시적인 실시예에서, 저장 서브시스템(10)은 캐시 메모리(24)를 포함한다. 캐시 메모리(24)는, 예를 들어, DDR-DRAM(double data rate dynamic random access memory)과 같은 임의의 타입을 가질 수 있다. 저장 서브시스템(10)은 또한, 중앙 프로세싱 유닛(CPU)(26) 및 작업(working) 메모리(28)를 포함한다. 작업 메모리(28)는, 예를 들어, 정적 RAM과 같은 임의의 타입을 가질 수 있다. CPU(26)는 일반화된 프로세싱 태스크들을 수행하는 반면, 저장 서브시스템(10)은, 다음의 특수화된 프로세싱 엘리먼트들, 즉, 메시지 프로세서(30), 커맨드 프로세서(32), 캐시 프로세서(34), 버퍼 프로세서(36), 백-엔드 프로세서(38), 및 직접 메모리 액세스(DMA; direct memory access) 엔진(40)을 더 포함한다. 예시적인 실시예에서, 저장 서브시스템(10)이 이들 특수화된 프로세싱 엘리먼트들을 포함하지만, 다른 실시예들은, 그러한 다른 실시예들에서 본원에 설명된 프로세싱 동작들 중 일부 또는 전부를 수행할 수 있는 더 적은 또는 더 많은 프로세싱 엘리먼트들을 포함할 수 있다. 저장 서브시스템(10)은 또한, 시스템 또는 매트릭스 버스들과 같은 시스템 상호연결부(42)를 포함하며, 이를 통해, 위에서 언급된 프로세싱 엘리먼트들이 서로 통신한다. 위에서 언급된 엘리먼트들 간에 다른 통신 또는 신호 경로들이 또한 포함될 수 있다. 호스트 인터페이스(44)(이를 통해, 저장 서브시스템(10)이 호스트 시스템(12)과 통신함) 및 저장 인터페이스(46)(이를 통해, 저장 서브시스템(10)이 백-엔드 저장소(14)와 통신함)가 또한 포함될 수 있다. 호스트 인터페이스(44)는, 예를 들어, PCIe(Peripheral Component Interconnect Express)와 같은 통신 버스 표준을 따를 수 있고, 연관된 PCIe 제어기를 포함할 수 있다. 메모리 인터페이스들 및 연관된 메모리 제어기들과 같은 다른 인터페이스들이 또한 포함될 수 있지만, 명확화의 목적들을 위해 도시되지 않는다. 도시되진 않지만, 저장 서브시스템(10)은, 호스트 시스템(12)의 마더보드 또는 백플레인(backplane)에 연결되는 가속기 카드의 일 부분을 정의할 수 있다. 위에서 언급된 프로세싱 엘리먼트들 중 일부 또는 전부는, 필드-프로그래밍가능 게이트 어레이(FPGA; ield-programmable gate array), 주문형 집적 회로(ASIC; application-specific integrated circuit), 또는 다른 디바이스와 같은 집적 회로 디바이스(도시되지 않음)에 포함될 수 있다.

[0012] [0022] 도 2에 예시된 바와 같이, 캐시 메모리(24)는, 아래에 설명되는 몇몇 타입들의 캐싱-관련 정보를 저장하는데 사용될 수 있다. 그러나, 다른 실시예들에서, 상이한 타입들의 데이터가 상이한 메모리들에 저장되거나

또는 임의의 다른 방식으로 구성될 수 있다. 예시적인 실시예에서, 캐시 메모리(24)는, 데이터 영역(48), 캐시 세그먼트(CS) 영역(50), 해시 테이블(hash table) 영역(52), 분산-수집 리스트(SGL; scatter-gather list) 영역(54), 및 기타(miscellaneous) 영역(56)을 포함한다. 당업자에 의해 인식되는 바와 같이, 도 2에서, 캐시 메모리(24)의 이들 영역들의 공간적 어레인지먼트(arrangement)들은, 명확화의 목적들을 위해 개념적인 방식으로 도시되고, 정보는 임의의 방식으로 캐시 메모리(24) 또는 다른 메모리에 맵핑되거나 또는 달리 배열될 수 있다.

[0013] [0023] 캐싱된 데이터는 버퍼 블록들로 지칭되는 단위들로 데이터 영역(48)에 저장된다. 단위는, 예를 들어, 4 KB(kilobytes)와 같은 데이터의 양을 정의한다. 용어 "블록"은 데이터가 연속적(contiguous)임을 의미한다. 위에서 언급된 스트라이프(22)가 예컨대 64 KB로 이루어지는 예시적인 실시예에서, 그에 따라, 각각의 스트라이프(22)는 16개의 버퍼 블록들에 대응한다. 아래에서 더 상세히 설명되는 바와 같이, 임의의 버퍼 블록은, 버퍼 블록들의 임의의 순서화에 상관없이 그리고 버퍼 블록들의 물리적 어드레스와 로직 어드레스 간의 임의의 관계에 상관없이 데이터 영역(48) 내의 임의의 이용가능한 물리적 위치(즉, 어드레스)에 캐싱 또는 저장될 수 있다. 결과적으로, 스트라이프(22)에 대응하는 버퍼 블록들은 반드시 서로 연속적으로 저장될 필요는 없다. 이러한 특성은 연관성으로 지칭된다.

[0014] [0024] 예시의 목적들을 위해, 몇몇 예시적인 버퍼 블록들(58, 60, 62, 64, 66, 68, 70, 72 등)이 데이터 영역(48) 내의 다양한 물리적 위치들에 저장된 것으로 도시된다. (데이터 영역(48) 내의 생략 부호들은, 명확화의 목적들을 위해 도시되지 않은, 부가적인 물리적 위치들에 있는 부가적인 버퍼 블록들을 표시한다.) 데이터 영역(48)의 저장 용량은 실질적으로 백-엔드 저장소(14)의 저장 용량보다 작을 수 있다. 예를 들어, 백-엔드 저장소(14)의 저장 용량은 대략 테라바이트 단위일 수 있는 반면, 데이터 영역(48)의 저장 용량은 대략 기가바이트 또는 메가바이트 단위일 수 있다. 프로세싱을 용이하게 하기 위해, 데이터 영역(48) 내의 물리적 위치는, 물리적 메모리 어드레스로부터의 오프셋(offset) 또는 인덱스(index)로서 기능하는 버퍼 블록 식별자(BBID; buffer block identifier)에 의해 식별될 수 있다. 예에 관하여 아래에 설명되는 바와 같이, 예시적인 버퍼 블록들(68, 66, 70, 62, 및 64)은 파선 화살표들에 의해 표시되는 방식으로 순서화되고, 예시적인 버퍼 블록(68)은 시퀀스의 처음이고, 예시적인 버퍼 블록(64)은 시퀀스의 마지막이다(부가적인 버퍼 블록들이 생략 부호에 의해 표시되지만, 명확화의 목적들을 위해 도시되진 않음). 함께 그룹화되어 있는 것이 아니라 데이터 영역(48) 전체에 걸쳐 분산된 것으로 도 2에 도시되는 그러한 예시적인 버퍼 블록들(58, 60, 62, 64, 66, 68, 70, 및 72)은 연관성을 예시하기 위한 것으로 의도된다.

[0015] [0025] 예시의 목적들을 위해, 몇몇 예시적인 캐시 세그먼트들(74, 76, 78 등)이 캐시 세그먼트 영역(50) 내의 물리적 위치들에 저장된 것으로 도시된다. 부가적인 캐시 세그먼트들이 생략 부호에 의해 표시되지만 명확화의 목적들을 위해 도시되진 않는다. 아래에 더 상세히 설명되는 바와 같이, 캐시 세그먼트는, 캐싱된 버퍼 블록들을 설명하는 메타데이터를 포함하는 데이터 구조이다.

[0016] [0026] 해시 테이블(80)이 캐시 세그먼트(102)에 관련되는 방식이 도 3에 예시된다. 해시 테이블(80)은 해시 테이블 영역(52)(도 2)에 저장될 수 있다. 호스트 시스템(12)에 의해 개시되는 기입 동작 또는 판독 동작(이들은 각각, 기입 입력/출력(I/O) 동작 및 판독 I/O 동작으로 또한 지칭될 수 있음)의 대상(subject)인 데이터의 로직 어드레스에 해시 함수(84)가 적용된다. 기입 또는 판독 동작은 일반적으로, 저장 서브시스템(10)이 기입 요청 또는 판독 요청, 기입 또는 판독될 데이터의 양의 표시, 및 데이터의 하나 또는 그 초과의 로직 어드레스들을 호스트(12)로부터 수신하는 것에 대한 응답으로 개시된다. 당업자에 의해 잘 이해되는 바와 같이, 용어 "로직 어드레스"는, 저장 서브시스템(10) 및 백-엔드 저장소(14)가 데이터에 대해 동작하는 물리적 어드레스 공간들과 대조적으로, 호스트 시스템(12)이 데이터에 대해 동작하는 어드레스 공간을 지칭한다.

[0017] [0027] 해시 테이블(80)은 다수(n)의 슬롯들을 포함하며, 그 슬롯들의 제 1 예시적인 슬롯(82), 제 2 예시적인 슬롯(84) 등 내지 다른 예시적인 슬롯(86) 그리고 마지막 또는 "n번째" 예시적인 슬롯(88)이 도시되며, 부가적인 슬롯들이 생략 부호들에 의해 표시되지만 명확화의 목적들을 위해 도시되진 않는다. 해시 테이블(80)은 임의의 수의 슬롯들을 가질 수 있지만, 그 수는 일반적으로 호스트 어드레스 공간에서의 로직 어드레스들의 수보다 실질적으로 적다. 해시 함수(84)의 예는 Slot = (LBA)MOD(n)이고, 여기서, "Slot"은 해시 테이블(80) 내의 슬롯에 대한 인덱스를 표현하고, "LBA"는 로직 어드레스를 표현하며, MOD 또는 모듈로(modulo)는 모듈식 산술 함수이다. 테이블을 인덱싱하기 위한 해시 함수의 사용은 본 기술분야에서 잘 이해되어 있으므로, 추가적인 세부사항들은 본원에서 설명되지 않는다.

[0018] [0028] 각각의 슬롯은 다수의 엔트리(entry)들(90)을 갖는다. 예를 들어, 해시 테이블(80)의 각각의 슬롯은 4개의 엔트리들(90)을 가질 수 있다. 해시 테이블 슬롯 당 단일 엔트리가 아니라 해시 테이블 슬롯 당 다수의

(즉, 2개 또는 그 초과) 엔트리들(90)을 이용하는 것은, 해시 테이블 어드레스 상충(conflict)들이 본 기술분야에서 일반적으로 지칭되는 바와 같은 "충돌(collision)들"을 최소화하는 것을 도울 수 있다. 아래에 설명되는 바와 같이, (미스(miss)의 경우에서) 슬롯 내의 임의의 비어있는(empty) 엔트리(90)가 기입 요청을 수행하는데 사용될 수 있다. 슬롯의 엔트리들 전부가 점유되어 있는 예시에서는, 그 후, 부가적인 엔트리들(92, 94 등)이 링크된 리스트의 형태로 부가될 수 있다.

[0019] [0029] 각각의 엔트리(90)는 로직 어드레스 필드(96), 캐시 세그먼트 식별자(CSID; cache segment identifier) 필드(98), 및 유효 엔트리 필드 또는 비트(V)(100)를 포함한다. 기입 및 판독 동작들의 예들에 관하여 아래에 설명되는 바와 같이, 캐시 세그먼트 식별자 필드(98)는, 캐시 세그먼트 영역(50)(도 2)에 저장된 캐시 세그먼트를 식별 또는 인덱싱하는 캐시 세그먼트 식별자(예컨대, 포인터(pointer))를 저장하도록 구성된다.

[0020] [0030] 캐시 세그먼트 식별자에 의해 식별된 각각의 캐시 세그먼트는, 도 3에 도시된 예시적인 캐시 세그먼트(102)의 구조를 가질 수 있다. 각각의 그러한 캐시 세그먼트는 다수의 캐시 세그먼트 리스트 엘리먼트들을 포함하며, 그 엘리먼트들의 제 1 예시적인 캐시 세그먼트 리스트 엘리먼트(104), 제 2 예시적인 캐시 세그먼트 리스트 엘리먼트(106) 등 내지 마지막 예시적인 캐시 세그먼트 리스트 엘리먼트(108)가 도시되며, 부가적인 캐시 세그먼트 리스트 엘리먼트들이 생략 부호들에 의해 표시되지만 명확화의 목적들을 위해 개별적으로 도시되진 않는다. 위에서 언급된 스트라이프(22)가 16개의 버퍼 블록들에 대응하는 예시적인 실시예에서, 각각의 캐시 세그먼트는, 각각의 캐시 세그먼트 리스트 엘리먼트가 그 버퍼 블록들 중 하나에 대응할 수 있도록, 16개의 캐시 세그먼트 리스트 엘리먼트들을 대응하게 가질 수 있다. 그럼에도 불구하고, 다른 실시예들에서, 각각의 캐시 세그먼트는 임의의 수의 캐시 세그먼트 리스트 엘리먼트들을 가질 수 있다.

[0021] [0031] 각각의 캐시 세그먼트 리스트 엘리먼트는, 다음의 플래그(flag) 필드들, 즉, 버퍼 블록 식별자(BBID) 필드(110); 유효 버퍼 블록 필드 또는 비트(V)(112); 더티 버퍼 블록 필드 또는 비트(D)(114); 플러시 버퍼 블록 필드 또는 비트(F)(116); 및 사용 카운트(CNT) 필드(118)를 포함한다. 이들 플래그 필드들에 저장된 플래그들이 사용되는 방식이 기입 및 판독 동작들에 관하여 아래에 논의되지만, 다음이 유의될 수 있다. 캐시 세그먼트 리스트 엘리먼트의 유효 (버퍼 블록) 비트(112)는, 그 캐시 세그먼트 리스트 엘리먼트의 버퍼 블록 식별자 필드(110)에 의해 식별되는 버퍼 블록이 유효인지 여부를 표시한다. 당업자에 의해 이해되는 바와 같이, 용어 "유효"는 일반적으로, 캐싱의 맥락에서, 데이터가 기입된 캐시 메모리 내의 위치들을 나타내기 위해 사용된다. 캐시 세그먼트 리스트 엘리먼트의 더티 (버퍼 블록) 비트(114)는, 그 캐시 세그먼트 리스트 엘리먼트의 버퍼 블록 식별자 필드(110)에 의해 식별되는 버퍼 블록이 더티인지 여부를 표시한다. 당업자에 의해 이해되는 바와 같이, 용어 "더티"는 일반적으로, 캐싱의 맥락에서, 아직 백-엔드 저장소(14)에 카피되지 않은 캐싱된 데이터를 지칭하기 위해 사용된다. 캐시 세그먼트 리스트 엘리먼트의 플러시 (버퍼 블록) 비트(116)는, 그 캐시 세그먼트 리스트 엘리먼트의 버퍼 블록 식별자 필드(110)에 의해 식별되는 버퍼 블록이 백-엔드 저장소(14)로 퇴거 또는 "플러싱(flush)"되는 프로세스에 있는지 여부를 표시한다. 캐시 세그먼트 리스트 엘리먼트의 사용 카운트 필드(118)는, 그 캐시 세그먼트 리스트 엘리먼트의 버퍼 블록 식별자 필드(110)에 의해 식별되는 버퍼 블록에 대하여 계류중인 데이터 액세스 요청들(판독 요청들 및 플러시 동작들을 포함함)의 수를 표시한다. 따라서, 캐시 세그먼트의 이들 필드들은, 그 캐시 세그먼트의 버퍼 블록 식별자 필드(110)에 의해 식별되는 버퍼 블록들의 양상들을 설명하는 메타데이터로서 기능한다.

[0022] [0032] 각각의 캐시 세그먼트는 또한, 이전 캐시 세그먼트 식별자 필드(120) 및 다음 캐시 세그먼트 식별자 필드(122)를 포함한다. 도 4에 예시된 바와 같이, 이들 필드들은 다수의 캐시 세그먼트들을 이중으로 링크된 리스트로 링크시키는데 사용될 수 있다. 예를 들어, 다른 예시적인 캐시 세그먼트(124)가 예시적인 캐시 세그먼트(102)에 링크될 수 있고, 또 다른 예시적인 캐시 세그먼트(126)가 예시적인 캐시 세그먼트(124)에 링크될 수 있는 그러한 식이다. 캐시 세그먼트들의 그러한 이중으로 링크된 리스트들의 다른 예들(도시되지 않음)은 임의의 수의 캐시 세그먼트들을 포함할 수 있다. 그러한 링크된 리스트의 각각의 캐시 세그먼트는, 데이터 영역(48)(도 1)에 저장된 하나 또는 그 초과의 버퍼 블록들을 (자신의 버퍼 블록 식별자 필드들(110)에 의해) 식별하거나 또는 그와 연관된다. 예를 들어, 도 4를 참조하고 그리고 도 2를 다시 부가적으로 참조하면, 예시적인 캐시 세그먼트(102)는 예시적인 버퍼 블록들(68, 66, 및 70)뿐만 아니라 다른 예시적인 버퍼 블록(128)까지의 부가적인 버퍼 블록들(개별적으로 도시되진 않지만 생략 부호에 의해 표현됨)을 식별하거나 그와 연관될 수 있고; 예시적인 캐시 세그먼트(124)는 또 다른 예시적인 버퍼 블록(130)뿐만 아니라 버퍼 블록(62)까지의 부가적인 버퍼 블록들(개별적으로 도시되진 않지만 생략 부호에 의해 표현됨)을 식별하거나 또는 그와 연관될 수 있으며; 예시적인 캐시 세그먼트(126)는 버퍼 블록(64)을 식별하거나 또는 그와 연관될 수 있다.

[0023] [0033] 위에서 언급된 스트라이프(22)가 16개의 버퍼 블록들에 대응하는 예시적인 실시예에 따르면, 캐시 세그먼트(102)의 16개의 캐시 세그먼트 리스트 엘리먼트들(도시되지 않음)은 16개의 예시적인 버퍼 블록들(68, 66, 70 등 내지 128)에 대응하고; 캐시 세그먼트(124)의 16개의 캐시 세그먼트 리스트 엘리먼트들(도시되지 않음)은 16개의 예시적인 버퍼 블록들(130 등 내지 62)에 대응함을 유의한다. 도 2 및 도 4에 도시된 예에서, 링크된 리스트의 마지막 캐시 세그먼트(126)는 오직 하나의 예시적인 버퍼 블록(64)에만 대응하는데, 이는, 이러한 예에서, 수반되는 버퍼 블록들의 총 수가 16의 배수가 아니기 때문임을 유의한다. 이러한 예에서, 캐시 세그먼트(126)의 하나의 캐시 세그먼트 리스트 엘리먼트는 유효 버퍼 블록 식별자를 포함할 것인 반면, 나머지 15개는 유효 버퍼 블록 식별자들을 포함하지 않을 것이다(그리고 그에 따라, 그들의 유효 비트들은 "0"일 것이거나 또는 디어써팅(de-assert)될 것임). 아래에 설명되는 바와 같이, 기입 요청은, 기입할 임의의 수의 버퍼 블록들을 표시할 수 있다. 캐시 세그먼트 내의 캐시 세그먼트 리스트 엘리먼트들의 수(예컨대, 16)보다 적은 버퍼 블록들을 표시하는 기입 요청은 완료를 위해 오직 하나의 캐시 세그먼트만 요구할 수 있는 반면, 그보다 많은 수의 버퍼 블록들을 표시하는 기입 동작은 (위에서 설명된 바와 같이 함께 링크된) 다수의 캐시 세그먼트들을 요구할 수 있다.

[0024] [0034] 버퍼 블록들이 저장된 데이터 영역(48) 내의 물리적 위치들을 식별하는 정보를 통신하기 위해, 데이터 구조들인 분산-수집 리스트(SGL)들이 이용될 수 있다. 임의의 수의 SGL들이 함께 링크될 수 있다. 예를 들어, 도 5에 예시된 바와 같이, 제 1 SGL(132)은 예시적인 버퍼 블록들(68, 66, 및 70)의 물리적 위치들을 식별하는 정보를 포함하는 분산-수집 엔트리(SGE; scatter-gather entry)들을 가질 수 있는 한편, 제 2 SGL(134)은 예시적인 버퍼 블록들(62 및 64)의 물리적 위치들을 식별하는 정보를 포함하는 분산-수집 엔트리(SGE)들을 가질 수 있다. 제 1 SGL(132)은 제 2 SGL(134)을 가리키는 링크(SGL ID)를 포함한다. 적어도 제 1 SGL(132)은 또한, 버퍼 블록들과 연관된 캐시 세그먼트의 캐시 세그먼트 식별자(CSID)를 포함할 수 있다.

[0025] [0035] 도 6a-6b의 흐름도에 의해 예시되는 바와 같이, 기입 요청은 다음과 같이 프로세싱될 수 있다. 블록(136)에 의해 표시되는 바와 같이, 저장 서브시스템(10)은 호스트 시스템(12)으로부터 기입 요청 통지를 수신한다. 기입 요청은, 하나 또는 그 초과의 로직 어드레스들 및 하나 또는 그 초과의 로직 어드레스들에 대응하게 기입될 하나 또는 그 초과의 데이터 블록들을 표시한다. 데이터는, 예컨대 4 KB와 같은, 버퍼 블록들과 동일한 사이즈의 블록들로 호스트(12)로부터 수신된다. 기입 요청은 임의의 수의 기입될 데이터 블록들을 표시할 수 있다. 도 1을 부가적으로 참조하면, 메시지 프로세서(30)가 기입 요청을 수신하여, 커맨드 프로세서(32)가 기입 동작을 개시하는 것을 허용하는 정보를 커맨드 프로세서(32)에 제공할 수 있다.

[0026] [0036] 블록(138)(도 6a)에 의해 표시되는 바와 같이, 버퍼 프로세서(36)(도 1)는, 프리 리스트(도시되지 않음)로부터의 데이터 영역(48) 내의 하나 또는 그 초과의 물리적 위치들을 할당함으로써 기입 동작의 개시에 응답한다. 프리 리스트는, 어떠한 유효 버퍼 블록들도 저장되지 않아서 그에 따라 버퍼 블록 저장에 대해 이용가능하거나 또는 "프리"인, 데이터 영역(48) 내의 물리적 위치들을 표시한다. 물리적 위치들은, 버퍼 블록들의 임의의 순서에 상관없이 선택되거나 또는 할당된다. 예를 들어, 도 2에서, 예시적인 버퍼 블록들(62, 64, 66, 68, 및 70)이 저장된 물리적 위치들은, 그 버퍼 블록들이 그 물리적 위치들에 저장되는 것을 초래한 기입 동작이 수행되었을 시에 그 물리적 위치들 각각이 프리 리스트에 리스팅되어 있었다는 점을 제외하면 서로 어떠한 관련도 갖지 않는다는 것을 유의한다. 예시적인 버퍼 블록들(62, 64, 66, 68, 및 70)의 로직 어드레스들은 그들 간의 파선 화살표들에 의해 표시된 순서를 따를 수 있지만, 그 순서는, 그 버퍼 블록들이 데이터 영역(48)에 저장되는 물리적 위치들과 어떠한 관련도 갖지 않는다.

[0027] [0037] 블록(140)에 의해 표시되는 바와 같이, 할당된 물리적 위치들을 식별하는 정보를 포함하는 하나 또는 그 초과의 SGL들(도시되지 않음)이 생성될 수 있다. SGL들은 DMA 엔진(40)(도 1)에 통신된다. 블록(142)에 의해 추가로 표시되는 바와 같이, DMA 엔진(40)은, 호스트(12)로부터의 기입 요청에 의해 식별된 데이터 블록들을 데이터 영역(48) 내의 할당된 물리적 위치들로 전달하기 위해 SGL 정보를 사용한다. 캐시 메모리(24)의 데이터 영역(48)에 저장된 데이터 블록들은 이제 위에 설명된 바와 같이 버퍼 블록들을 정의한다.

[0028] [0038] 아래에 설명되는 바와 같이, 기입 요청에 대한 응답으로 호스트(12)로부터 데이터 영역(48)으로 데이터를 전달하는 것에 후속하여, 캐시 프로세서(34)(도 1)는 기입 동작을 완료하기 위해 메타데이터를 업데이트한다. 기입 요청에 대한 응답으로의 기입 동작의 완료는, 데이터의 전달이 판독 동작에 의해 중단될 수 없다는 의미에서 원자적(atomic)임이 유의되어야 한다. 다른 방식으로 말하자면, 버퍼 블록들이 데이터 영역(48) 내의 물리적 위치들에 저장된 이후, 물리적 위치들을 로직 어드레스들과 연관시키기 위해, 임의의 판독 요청들이 그 로직 어드레스들에 대하여 계류중인지 여부에 상관없이 메타데이터가 업데이트된다. 트랜잭션(transaction) 프로세싱의 어휘(lexicon)에서, 캐시 프로세서(34)는 메타데이터를 업데이트함으로써 기입 트랜

잭션을 "커밋(commit)"하며, 그에 의해 원자성이 보장된다. 다음의 동작들은 그러한 메타데이터를 업데이트하는 것에 관한 것이다.

[0029] [0039] 블록(144)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는, 기입 요청에서 식별되는 하나 또는 그 초과의 로직 어드레스들을 위에서 설명된 해시 테이블(80)(도 3)에서 검색한다. 위에 설명된 바와 같이, 해시 함수(84)는 로직 어드레스를 해시 테이블(80) 내의 슬롯에 대한 인덱스로 리졸빙(resolve)한다. 블록(146)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는 로직 어드레스를 그 슬롯 내의 엔트리들 각각과 비교하여, 그 로직 어드레스에 데이터가 이미 캐싱되었다는 것을 표시하는 매치(match) 또는 "히트"가 존재하는지를 결정한다. 4개의 엔트리들 중 어떠한 것도 매치가 아니면, 캐시 프로세서(34)는, 위에 설명된 링크된 리스트를 따라 부가적인 엔트리들로 갈 수 있다. 어떠한 엔트리들도 로직 어드레스에 매칭하지 않으면, 해시 테이블 검색의 결과는 "미스"이다.

[0030] [0040] 해시 테이블 검색의 결과가 미스라고 캐시 프로세서(34)가 결정하면(블록 146), 블록(148)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는 새로운 캐시 세그먼트를 할당한다. 새로운 캐시 세그먼트는 위에 설명된 바와 같은 CSID에 의해 식별된다. 캐시 프로세서(34)가 CSID를 슬롯 엔트리들 중 이용가능한 하나에 저장하거나, 또는 슬롯 자체의 엔트리들 전부(예컨대, 4개)가 점유된 경우, 슬롯의 "링크" 엔트리의 CSID는, 다음 해시 링크를 새롭게 부가된 CSID로 셋팅하도록 업데이트된다. 새로운 CSID의 이전 해시 링크는, 해시 테이블의 참조된 슬롯의 "링크" 엔트리의 CSID로 셋팅된다. 그 후, 블록(150)에 의해 표시되는 바와 같이, 새롭게 할당된 캐시 세그먼트 내의 각각의 캐시 세그먼트 리스트 엘리먼트에 대해, 캐시 프로세서(34)는 SGL로부터 버퍼 블록 식별자 필드들(110)(도 3)로 버퍼 블록 식별자들을 카피하고, 그들의 유효 비트들 및 더티 비트들을 셋팅할 뿐만 아니라, 플러시 비트 및 사용 카운트를 클리어(clear)한다. 또한, 블록(152)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는, 새롭게 할당된 캐시 세그먼트의 캐시 세그먼트 식별자를 최소 최근 사용(LRU; least-recently used) 더티 링크된 리스트(도시되지 않음)의 테일(tail)에 부가한다. LRU 더티 링크된 리스트는, 아래에 설명되는 바와 같이, 캐싱된 데이터를 백-엔드 저장소(14)로 플러싱하는데 사용될 수 있다. 블록들(146, 148, 150, 및 152)에 의해 표시되는 동작들은 기입 요청에서의 데이터의 하나의 스트라이프(22)에 관한 것이다. 블록(154)에 의해 표시되는 바와 같이, 기입 요청이 하나 초과의 스트라이프(22)에 걸쳐 있으면, 이들 동작들은, 해시 테이블 검색의 결과가 미스인 각각의 스트라이프(22)에 대해 반복된다.

[0031] [0041] 해시 테이블 검색의 결과가 히트라고 캐시 프로세서(34)가 결정하면(블록 146), 블록(156)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는 슬롯 엔트리에 의해 식별되는 캐시 세그먼트를 판독한다. 그 후, 블록(158)에 의해 표시되는 바와 같이, "비어있는", 즉, 자신의 버퍼 블록 식별자 필드(110)에 유효 버퍼 블록 식별자를 아직 포함하고 있지 않은 각각의 캐시 세그먼트 리스트 엘리먼트에 대해, 캐시 프로세서(34)는, SGL로부터 그 버퍼 블록 식별자 필드(110)로 버퍼 블록 식별자를 카피한다. 그 캐시 세그먼트 리스트 엘리먼트에서, 캐시 프로세서(34)는 또한 다음의 플래그들(도 3)을 셋팅하는데, 유효 비트(112)는 "1"("참(true)")의 값으로 셋팅되고; 더티 비트(114)는 "1"("참")의 값으로 셋팅되며; 사용 카운트(118)는 0의 값으로 셋팅된다. 비어있지 않은, 즉, 유효 버퍼 블록 식별자를 포함하는 임의의 캐시 세그먼트 리스트 엘리먼트에 대해, 프로세싱은 블록(162)(도 6b)에서 계속된다.

[0032] [0042] 블록들(162 및 164)에 의해 표시되는 바와 같이, 캐시 세그먼트 리스트 엘리먼트의 플러시 비트(116)가 "0"("거짓(false)")이 아니거나 또는 캐시 세그먼트 리스트 엘리먼트의 사용 카운트(118)가 0 이외의 값을 포함한다고 캐시 프로세서(34)가 결정하면, 블록(166)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는, 플래그들(즉, 유효 비트(112), 더티 비트(114), 플러시 비트(116), 및 사용 카운트 필드(118)의 값들)을, 예를 들어, (오리지널(original) 리스트 엘리먼트 BBID 값에 의해 인덱싱되는 바와 같은) 캐시 메모리(24)(도 1)의 기타 영역(56)에 카피 또는 저장한다. 일단 플래그들이 저장되면, 캐시 프로세서(34)는, 블록(168)에 의해 표시되는 바와 같이, 캐시 세그먼트 리스트 엘리먼트의 버퍼 블록 식별자 필드(110)를 SGL로부터 획득된 버퍼 블록 식별자로 오버라이팅(overwrite)할 수 있다. 플래그들은 새롭게 기입된 버퍼 블록에 대한 것과 동일한 스테이지로, 즉, 플러시 = 0, 더티 = 1, 유효 = 1, 사용 카운트 = 0으로 셋팅된다.

[0033] [0043] 그러나, 캐시 세그먼트 리스트 엘리먼트의 플러시 비트(116)가 "0"("거짓")이고 캐시 세그먼트 리스트 엘리먼트의 사용 카운트(118)가 0의 값을 포함한다고 캐시 프로세서(34)가 결정하면(블록들 162 및 164), 블록(170)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는, 그 캐시 세그먼트 리스트 엘리먼트의 버퍼 블록 식별자 필드(110)에 있는 버퍼 블록 식별자를 할당해제(de-allocate)한다. 즉, 버퍼 블록 식별자는 위에서 언급된 프리 리스트로 리턴된다. 그 후, 캐시 프로세서(34)는, 블록(168)에 의해 표시되는 바와 같이, 그 캐시 세그먼

트의 버퍼 블록 식별자 필드들(110)을 SGL로부터 획득된 버퍼 블록 식별자들로 오버라이팅한다.

[0034] [0044] 또한, 해시 테이블 히트에 대한 응답으로, 캐시 프로세서(34)는, 블록(172)에 표시되는 바와 같이, LRU 더티 링크된 리스트를 업데이트한다. 더 구체적으로는, LRU 더티 링크된 리스트의 일 위치에 캐시 세그먼트 식별자가 이미 존재한다고 결정되면, 캐시 세그먼트 식별자가 그 위치로부터 제거되고, 그 캐시 세그먼트 식별자에 대한 새로운 위치가 LRU 더티 링크된 리스트의 테일에 부가(즉, 링크)된다. 따라서, 가장 최근에 기입된 캐시 세그먼트 식별자들이 LRU 더티 링크된 리스트의 테일로 이동된다. 아래에 설명되는 바와 같이, 이러한 방식으로 LRU 더티 링크된 리스트를 유지하는 것은, 덜 최근에 기입된(즉, 가장 오래된) 데이터를 백-엔드 저장소(14)로 퇴거시키거나 또는 플러싱하는 것을 가능하게 한다. 프로세스는 그 후, 블록(158)에 관하여 위에 설명된 바와 같이 계속된다.

[0035] [0045] 기입 요청이 하나 초과의 스트라이프(22)에 걸쳐 있으면, 위에서 설명된 동작들은, 블록(154)에 의해 표시되는 바와 같이, 해시 테이블 검색의 결과가 히트인 각각의 스트라이프(22)에 대해 반복된다. 기입 동작의 모든 스트라이프들(22)이 위에서 설명된 방식으로 프로세싱된 경우, 블록(174)에 의해 표시되는 바와 같이, 기입 동작이 완료되었음이 호스트(12)에 통지된다.

[0036] [0046] 도 7의 흐름도에 의해 예시되는 바와 같이, 판독 요청은 다음과 같이 프로세싱될 수 있다. 블록(180)에 의해 표시되는 바와 같이, 저장 서브시스템(10)은 호스트 시스템(12)으로부터 판독 요청을 수신한다. 판독 요청은, 데이터가 판독될 하나 또는 그 초과의 로직 어드레스들을 표시한다. 판독 요청은, 판독될 데이터의 양을 표시할 수 있다. 호스트 시스템(12)은, 예컨대 4 KB와 같은, 버퍼 블록들과 동일한 사이즈의 데이터의 블록들을 판독한다. 도 1을 부가적으로 참조하면, 메시지 프로세서(30)가 판독 요청을 수신하여, 커맨드 프로세서(32)가 판독 동작을 개시하는 것을 허용하는 정보를 커맨드 프로세서(32)에 제공할 수 있다.

[0037] [0047] 블록(182)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는, 해시 테이블(80)(도 3)에서 검색을 수행함으로써 판독 동작의 개시에 응답한다. 즉, 기입 동작들에 관하여 위에 설명된 것과 동일한 방식으로, 해시 함수(84)가 각각의 로직 어드레스에 적용되어 해시 테이블(80)로의 하나 또는 그 초과의 인덱스들을 생성한다. 각각의 인덱스는 판독될 하나의 슬롯을 표시한다. 해시 테이블(80)로부터 판독할 슬롯들의 수는 판독 요청에 의해 표시되는 데이터의 양에 의존한다. 위에 설명된 바와 같이, 각각의 슬롯은, 예를 들어, 64 KB에 대응할 수 있는 하나의 스트라이프(22)를 표현한다. 판독 요청은 다수의 스트라이프들(22)에 걸쳐 있을 수 있다. 표시된 슬롯들 각각이 판독되어 결과가 히트인지 또는 미스인지 여부가 결정된다. 더 구체적으로는, 판독 동작에 대한 3개의 가능한 결과들이 존재하는데, 요청된 데이터 전부가 데이터 영역(48)에 존재할 수 있거나(이러한 결과는 전체 히트(full hit)로 지칭됨); 요청된 데이터의 일부가 데이터 영역(48)에 존재할 수 있거나(이러한 결과는 부분 히트로 지칭됨); 또는 요청된 데이터 중 어떠한 것도 데이터 영역(48)에 존재하지 않을 수 있다(이러한 결과는 미스 또는 전체 미스로 지칭됨).

[0038] [0048] 블록(184)에 의해 표시되는 바와 같이, 해시 테이블 검색의 결과가 히트(이는, 전체 히트 또는 부분 히트일 수 있음)라고 캐시 프로세서(34)가 결정하면, 캐시 프로세서(34)는, 블록(186)에 의해 표시되는 바와 같이, 히트를 초래한 엔트리에 의해 표시되는 캐시 세그먼트들을 판독한다. 도 3에 관하여 위에 설명된 바와 같이, 그 캐시 세그먼트의 각각의 캐시 세그먼트 리스트 엘리먼트는, 그 캐시 세그먼트 리스트 엘리먼트의 버퍼 블록 식별자 필드(110)에서 식별되는 버퍼 블록이 유효인지 여부를 표시하는 유효 (버퍼 블록) 비트(112)를 갖는다. 판독 요청에서 표시된 로직 어드레스의 최하위 비트(least-significant bit)들은, 요청된 버퍼 블록들의 버퍼 블록 식별자들을 포함하는 캐시 세그먼트 리스트 엘리먼트들의 처음(beginning)을 식별하는데 사용될 수 있다. 그 후, 블록(188)에 의해 표시되는 바와 같이, 버퍼 블록이 식별된 캐시 세그먼트 리스트 엘리먼트의 유효 비트(112) 및 더티 비트(114) 각각에 의해 유효 및 더티인 것으로 결정되는 요청된 버퍼 블록들 각각에 대해, 캐시 프로세서(34)는, 그 캐시 세그먼트 리스트 엘리먼트의 사용 카운트 필드(118)의 값 또는 카운트를 증분(increment)한다. 블록(190)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는 그 후, 데이터 영역(48)으로부터 데이터를 판독하는데 사용될 하나 또는 그 초과의 SGL들(도시되지 않음)을 생성 및 파퓰레이팅(populate)한다. 캐시 프로세서(34)는 그 후, 블록(192)에 의해 표시되는 바와 같이, 판독 요청이 다른 스트라이프(22)에 걸쳐 있는지 여부를 결정한다. 다른 스트라이프(22)가 판독되어야 하면, 프로세싱은 블록(184)으로 리턴한다. 해시 테이블 검색의 결과가 미스라고 결정되면(블록 184), 캐시 프로세서(34)는, 블록(194)에 의해 표시되는 바와 같이 미스 카운트를 증분하고, 프로세싱은 블록(192)에 관하여 위에 설명된 바와 같이 계속된다.

[0039] [0049] 판독 요청과 연관된 스트라이프들 전부가 판독된 이후, 프로세싱은 블록(196)(도 7b)에서 계속된다. 캐시 프로세서(34)는, 결과가 전체 미스인지(블록 196)(즉, 스트라이프들 중 어떠한 것도 히트들이 아닌지) 또

는 전체 히트인지(블록 197)(즉 스트라이프들 전부가 히트들인지) 여부를 결정하기 위해, 위에서 언급된 미스 카운트를 사용할 수 있다. 결과가 전체 미스라고 캐시 프로세서(34)가 결정하면(블록 196), 블록(198)에 의해 표시되는 바와 같이, 백-엔드 프로세서(38)(도 1)가 요청된 데이터를 백-엔드 저장소(14)(도 1)로부터 판독한다. 백-엔드 저장소(14)로부터 호스트(12)로 데이터를 전달하는데 DMA 엔진(40)이 참여할 수 있다. 결과가 전체 히트라고 캐시 프로세서(34)가 결정하면(블록 197), 블록(199)에 의해 표시되는 바와 같이, 캐시 프로세서(34)는 요청된 데이터 전부를 데이터 영역(48)(도 2)으로부터 판독한다. 결과가 부분 히트(즉, 전체 미스(블록 196)도 아니고 전체 히트(블록 197)도 아님)라고 캐시 프로세서(34)가 결정하면, 캐시 프로세서(34)는, 블록(200)에 의해 표시되는 바와 같이, 요청된 데이터 중 데이터 영역(48)에서 발견될 수 없는 그 버퍼 블록들을 백-엔드 저장소(14)로부터 판독하고, 블록(201)에 의해 표시되는 바와 같이, 요청된 데이터 중 데이터 영역(48)에서 발견될 수 있는 그 버퍼 블록들을 데이터 영역(48)으로부터 판독한다. 데이터 전달이 완료된 경우, 블록(202)에 의해 표시되는 바와 같이, 판독 동작이 완료되었음이 호스트(12)에 통지된다.

[0040] [0050] 위에 언급된 바와 같이, SGL은, 데이터 영역(48)으로부터의 데이터의 전달을 용이하게 하기 위해 사용될 수 있다. 전체 히트가 아닌 부분 히트의 경우에서, 캐시 프로세서(34)는, 더티 및 더티가 아닌 버퍼 블록들을 식별하는 정보를 사용하여, 버퍼 블록들 중 어느 버퍼 블록들을 데이터 영역(48)으로부터 판독할지 그리고 어느 버퍼 블록들을 데이터 영역(48)에서 "건너 뛰고(skip over)" 그 대신 백-엔드 저장소(14)로부터 판독할지를 표시하는 정보를 SGL에 포함시킨다. 유효이고 더티인 버퍼 블록들은 데이터 영역(48)으로부터 판독되어야 하지만, 유효이고 더티가 아닌 버퍼 블록들은 백-엔드 저장소(14)로부터 판독될 수 있다. 캐시 관리자(34)는 SGL(또는 함께 링크된 다수의 SGL들)을 다음 중 어느 하나에, 즉, 부분 히트의 경우에서는 백-엔드 프로세서(38)에 전송하거나 또는 전체 히트의 경우에서는 DMA 엔진(40)에 전송한다. 블록(200)에 의해 표시되는 바와 같이, 요청된 버퍼 블록들 그 후, 데이터 영역(48), 백-엔드 저장소(14), 또는 데이터 영역(48) 및 백-엔드 저장소(14) 둘 모두의 결합으로부터 판독된다.

[0041] [0051] 도 8에 예시된 바와 같이, 캐싱 방법은, 하나 또는 그 초과의 로직 어드레스들 및 하나 또는 그 초과의 로직 어드레스들에 대응하게 기입될 하나 또는 그 초과의 데이터 블록들을 표시하는 기입 요청을 수신하는 단계(블록 204); 기입 요청에 대한 응답으로, 프리 리스트로부터의 캐시 메모리 내의 하나 또는 그 초과의 물리적 위치들을 할당하는 단계(블록 206); 하나 또는 그 초과의 로직 어드레스들에 대해 임의의 판독 요청들이 계류중인지 여부에 상관없이 하나 또는 그 초과의 데이터 블록들을 하나 또는 그 초과의 물리적 위치들에 저장하는 단계(블록 208); 하나 또는 그 초과의 데이터 블록들이 하나 또는 그 초과의 물리적 위치들에 저장된 이후, 그리고 하나 또는 그 초과의 로직 어드레스들에 대하여 임의의 판독 요청들이 계류중인지 여부에 상관없이, 하나 또는 그 초과의 물리적 위치들을 하나 또는 그 초과의 로직 어드레스들과 연관시키기 위해 메타데이터를 업데이트하는 단계(블록 210); 유효 데이터를 갖는 캐시 메모리 내의 각각의 물리적 위치에 대하여 계류중인 데이터 액세스 요청들 또는 "사용들"의 카운트를 유지하는 단계(블록 212); 및 물리적 위치에 대하여 어떠한 데이터 액세스 요청들도 계류중이지 않다고 사용 카운트가 표시하는 경우, 물리적 위치를 프리 리스트로 리턴하는 단계(블록 214)를 포함할 수 있다. 물리적 위치가 프리 리스트로 리턴되는 경우(블록 214), 대응하는 캐시 세그먼트 리스트 엘리먼트의 유효 비트가 디어써팅됨이 유의될 수 있다.

[0042] [0052] 데이터 영역(48)으로부터 백-엔드 저장소(14)로 데이터를 전달하는 것(이는 일반적으로, 캐시 메모리로부터 데이터를 퇴거시키는 것 또는 데이터를 플러싱하는 것으로 본 기술분야에서 지칭됨)은 상세히 설명되지 않는데, 이는, 데이터를 퇴거시키거나 또는 플러싱하는 것이 당업자에 의해 이해되는 종래의 방식으로 수행될 수 있기 때문이다. 간략하게, 도 9의 흐름도를 참조하면, 블록(216)에 의해 표시되는 바와 같이, 프리 리스트 내의 버퍼 블록들의 수가 임계치 미만으로 감소되었다고 결정된 경우, 버퍼 블록들은 데이터 영역(48)으로부터 백-엔드 저장소(14)로 플러싱될 수 있다. 블록(218)에 의해 표시되는 바와 같이, 위에 설명된 LRU 더티 링크된 리스트가 플러싱을 용이하게 할 수 있는데, 이는, 최소 최근 기입(least recently written) 버퍼 블록들을 식별하는 캐시 세그먼트가 LRU 더티 링크된 리스트의 헤드(head)에서 유지되기 때문이다. 그러한 최소 최근 기입 버퍼 블록들은 백-엔드 저장소(14)로 플러싱될 수 있고, LRU 더티 링크된 리스트가 업데이트되어 대응하는 캐시 세그먼트가 제거된다. 그러한 플러시 동작은 위에 설명된 기입 및 판독 동작들과 독립적으로, 즉, 백그라운드(background)에서 수행될 수 있음을 유의한다. 부가하여, 명확성의 목적들을 위해 도 9에 도시되진 않았지만, 하나 또는 그 초과의 버퍼 블록들이 플러싱된 이후, 사용 카운트는 도 8의 블록(212)에 의해 표시되는 바와 같이 감분(decrement)된다. 도 8의 블록(214)과 유사하게, 감분된 사용카운트가 0이면(이는 (BBID에 의해 식별되는) 물리적 위치에 대하여 어떠한 판독 요청들 또는 플러시 동작들(집합적으로 데이터 액세스 요청들로 본원에서 지칭됨)도 계류중이지 않다는 것을 표시함), 물리적 위치가 프리 리스트로 리턴되고, 대응하는 캐시 세그먼

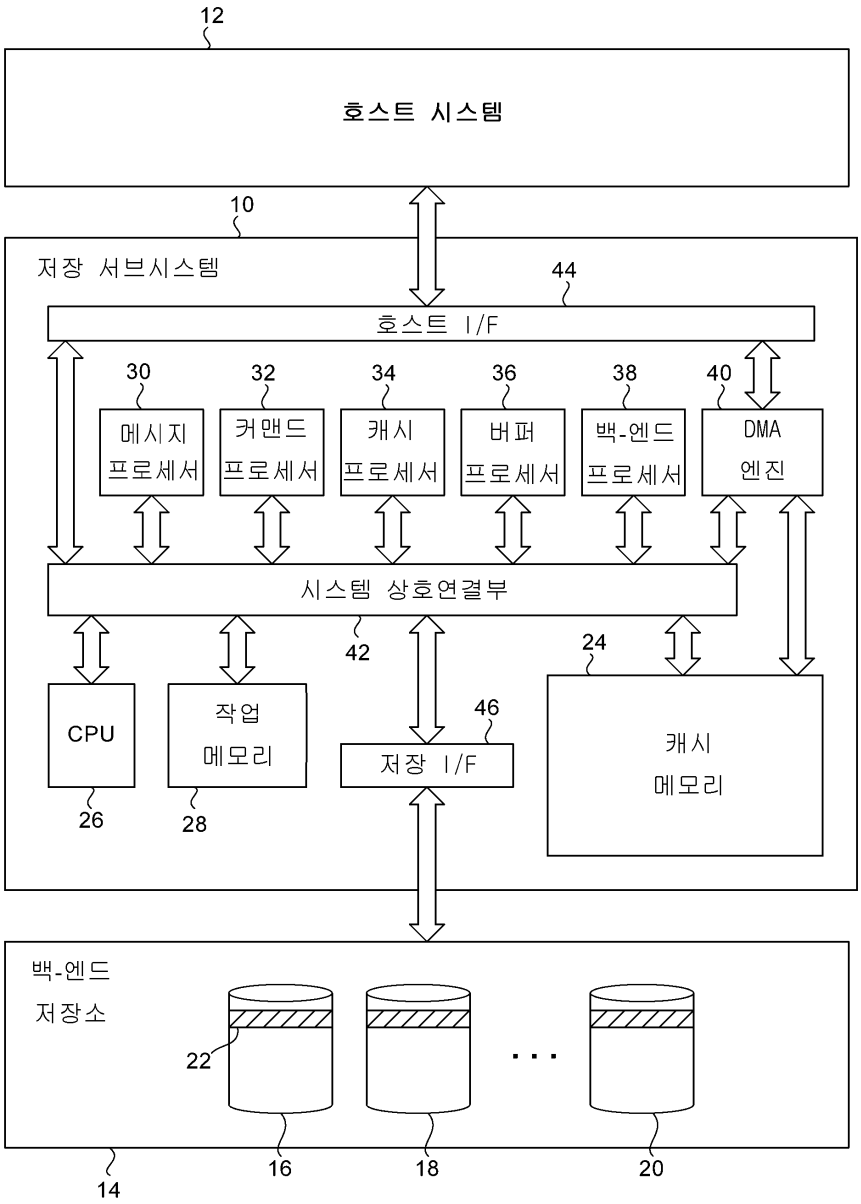
트 리스트 엘리먼트의 유효 비트가 디어써팅된다.

[0043] [0053] 도 6a-6b 및 도 7a-7b, 도 8, 및 도 9의 흐름도들은 단지 설명된 방법의 기초가 되는 로직의 예시 또는 예증적인 것으로만 의도됨이 이해되어야 한다. 당업자는, 다양한 실시예들에서, 설명된 방법들을 달성하기 위한 다양한 방식들 중 임의의 방식으로 저장 서브시스템이 구성될 수 있음을 인식할 수 있다. 몇몇 실시예들에서, 도 6a-6b, 도 7a-7b, 도 8, 및 도 9에 관하여 위에 설명된 단계들 또는 동작들은 임의의 적절한 순서 또는 시퀀스로 발생할 수 있으며, 서로 병렬로 또는 비동기적으로 발생하는 것을 포함한다. 몇몇 실시예들에서, 위에 설명된 단계들 또는 동작들을 다른 것들과 결합될 수 있거나 또는 생략될 수 있다. 명확화의 목적들을 위해, 도 1에서의 개별적인 프로세싱 엘리먼트들의 형태로 그리고 도 6a-6b, 도 7a-7b, 도 8, 및 도 9에서 개별적인 흐름도들의 형태로 도시되지만, 기본 로직은 하드웨어 또는 소프트웨어의 임의의 결합 간에 임의의 적절한 방식으로 모듈화되거나 또는 분산될 수 있다.

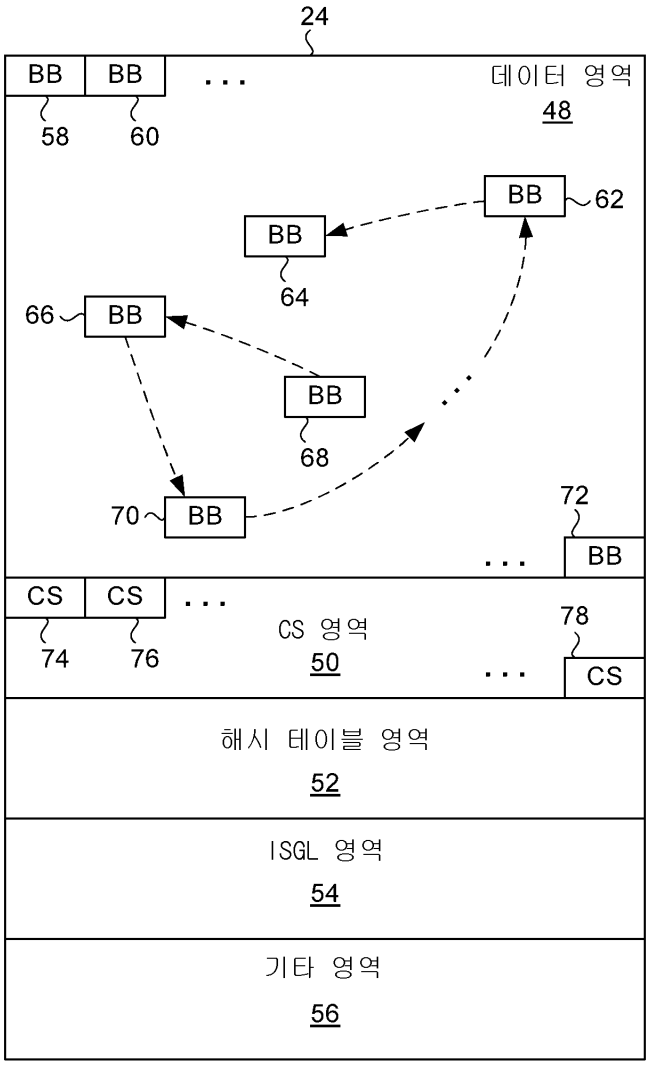
[0044] [0054] 본 발명은, 본 발명의 원리들 및 개념들을 나타내려는 목적을 위해 하나 또는 그 초과의 예시적인 실시예들을 참조하여 설명되었음이 유의되어야 한다. 본 발명은 이들 실시예들로 제한되지 않는다. 당업자들에 의해 이해될 바와 같이, 본원에서 제공되는 설명의 관점에서, 본원에서 설명된 실시예들에 대해 많은 변형들이 이루어질 수 있고, 모든 그러한 변형들이 본 발명의 범위 내에 있다.

도면

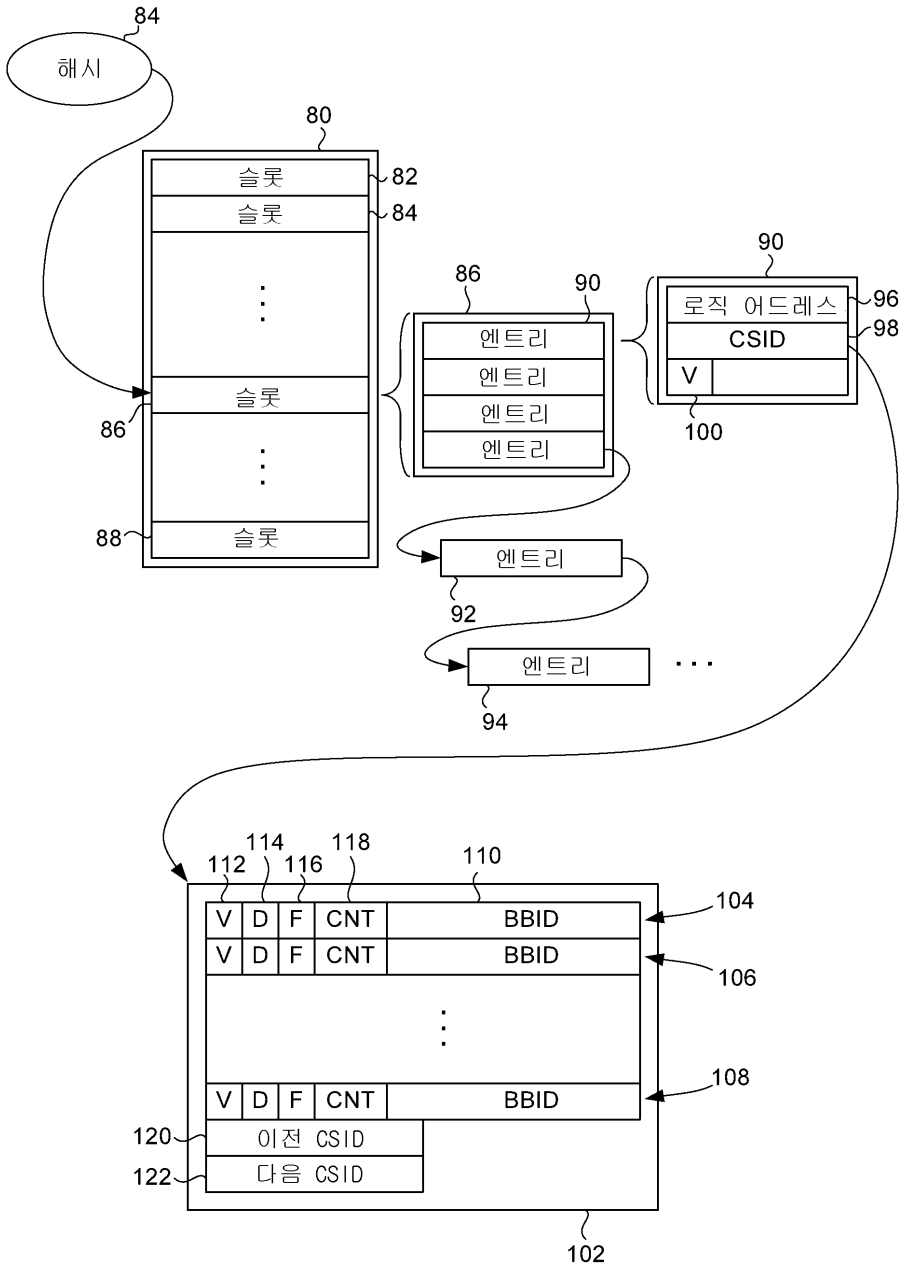
도면1


12
호스트 시스템
10
저장 서브시스템
44
호스트 I/F
30
메시지 프로세서
32
커맨드 프로세서
34
캐시 프로세서
36
버퍼 프로세서
38
백-엔드 프로세서
40
DMA 엔진
시스템 상호연결부
42
24
CPU
26
작업 메모리
28
46
저장 I/F
캐시 메모리
백-엔드 저장소
22
16
18
20
14

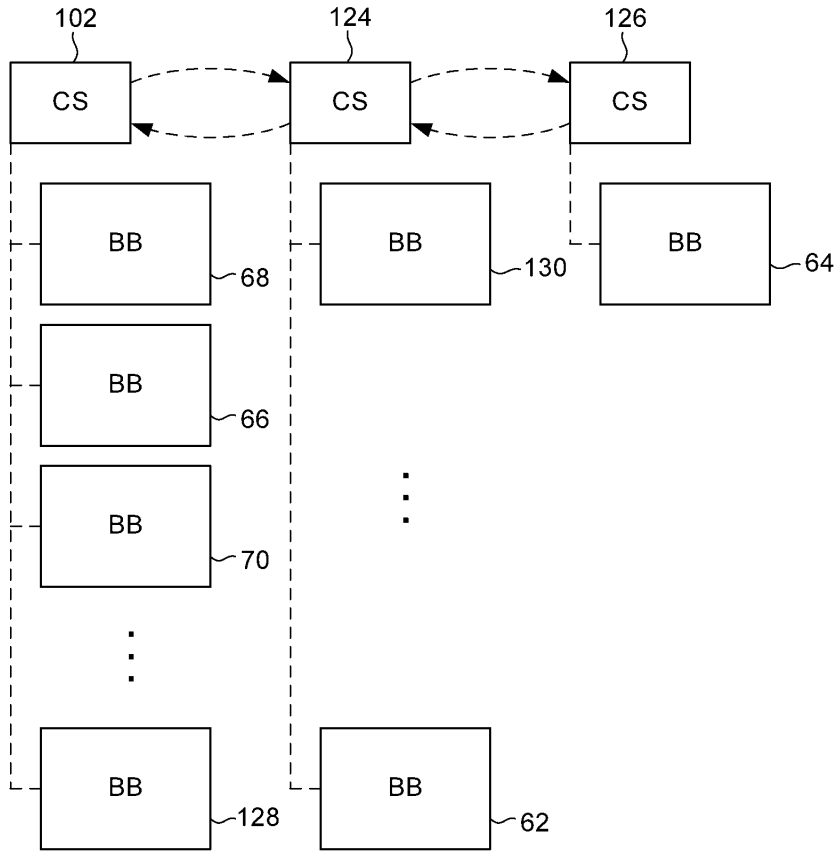
도면2


24
BB
BB
58
60
데이터 영역
48
BB
62
BB
64
66
BB
BB
68
70
BB
72
BB
CS
CS
74
76
CS 영역
50
78
CS
해시 테이블 영역
52
ISGL 영역
54
기타 영역
56

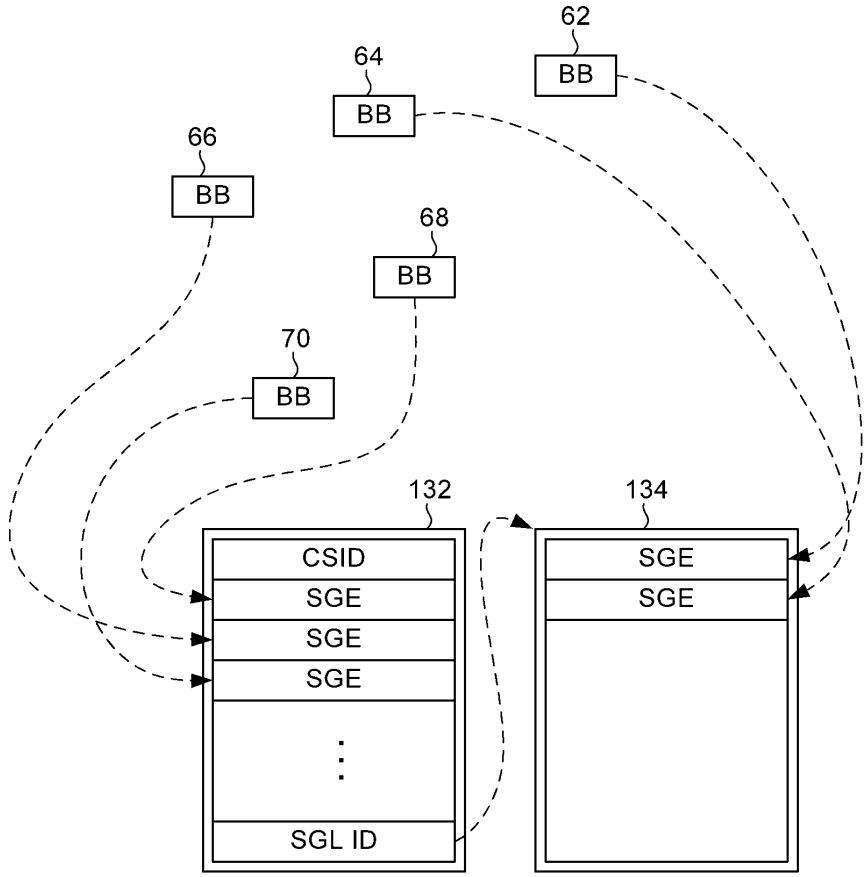
도면3


84
해시
80
슬롯
82
슬롯
84
86
슬롯
88
슬롯
86
90
엔트리
엔트리
엔트리
엔트리
90
로직 어드레스
96
CSID
98
V
100
엔트리
92
엔트리
94
114
112
116
118
110
V D F CNT BBID
104
V D F CNT BBID
106
108
V D F CNT BBID
120
이전 CSID
122
다음 CSID
102

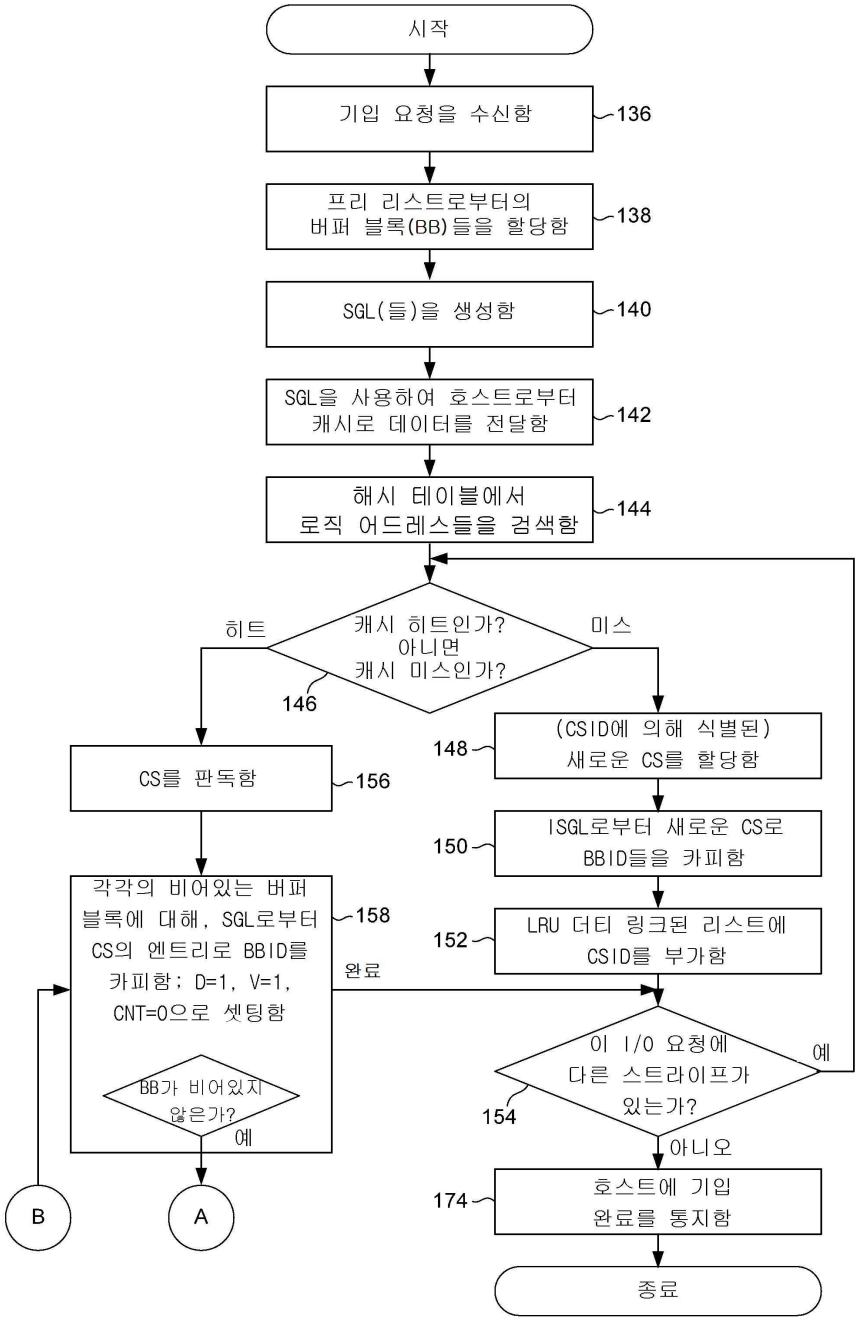
도면4



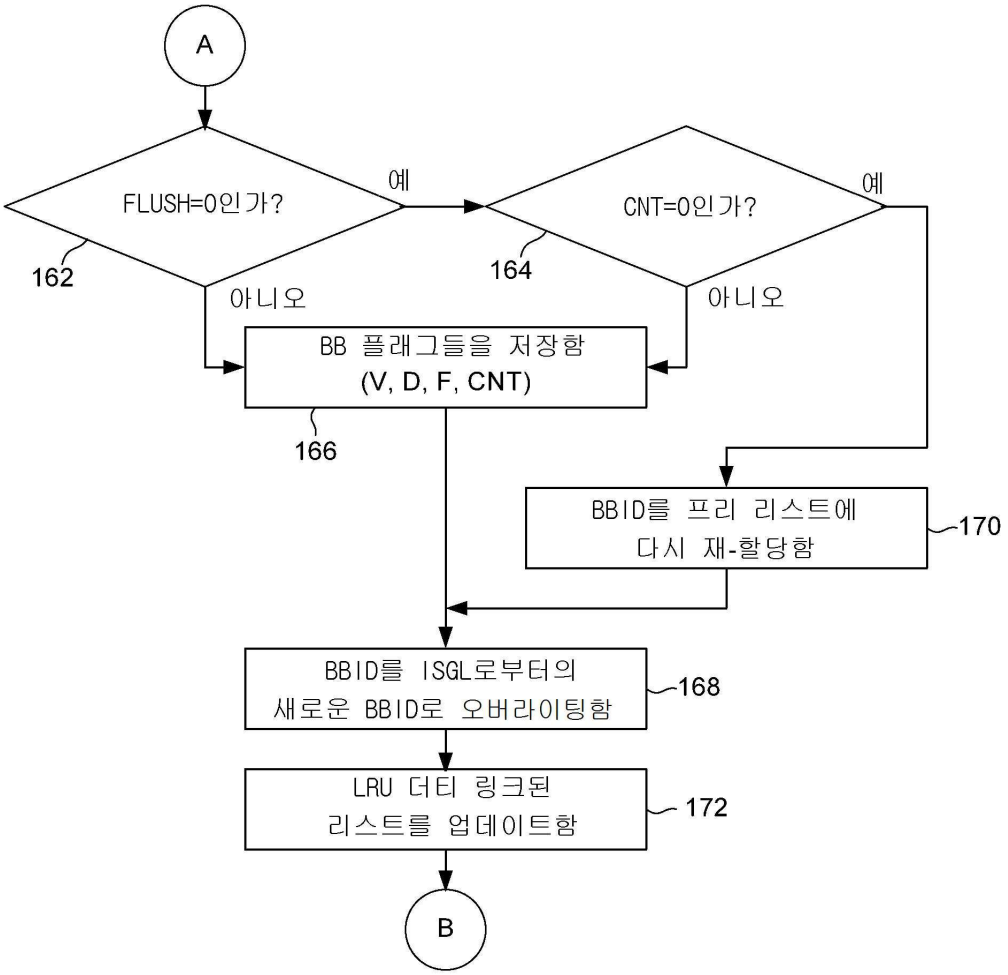
도면5


62
BB
64
BB
66
BB
68
BB
70
BB
132
134
CSID
SGE
SGE
SGE
SGL ID
SGE
SGE

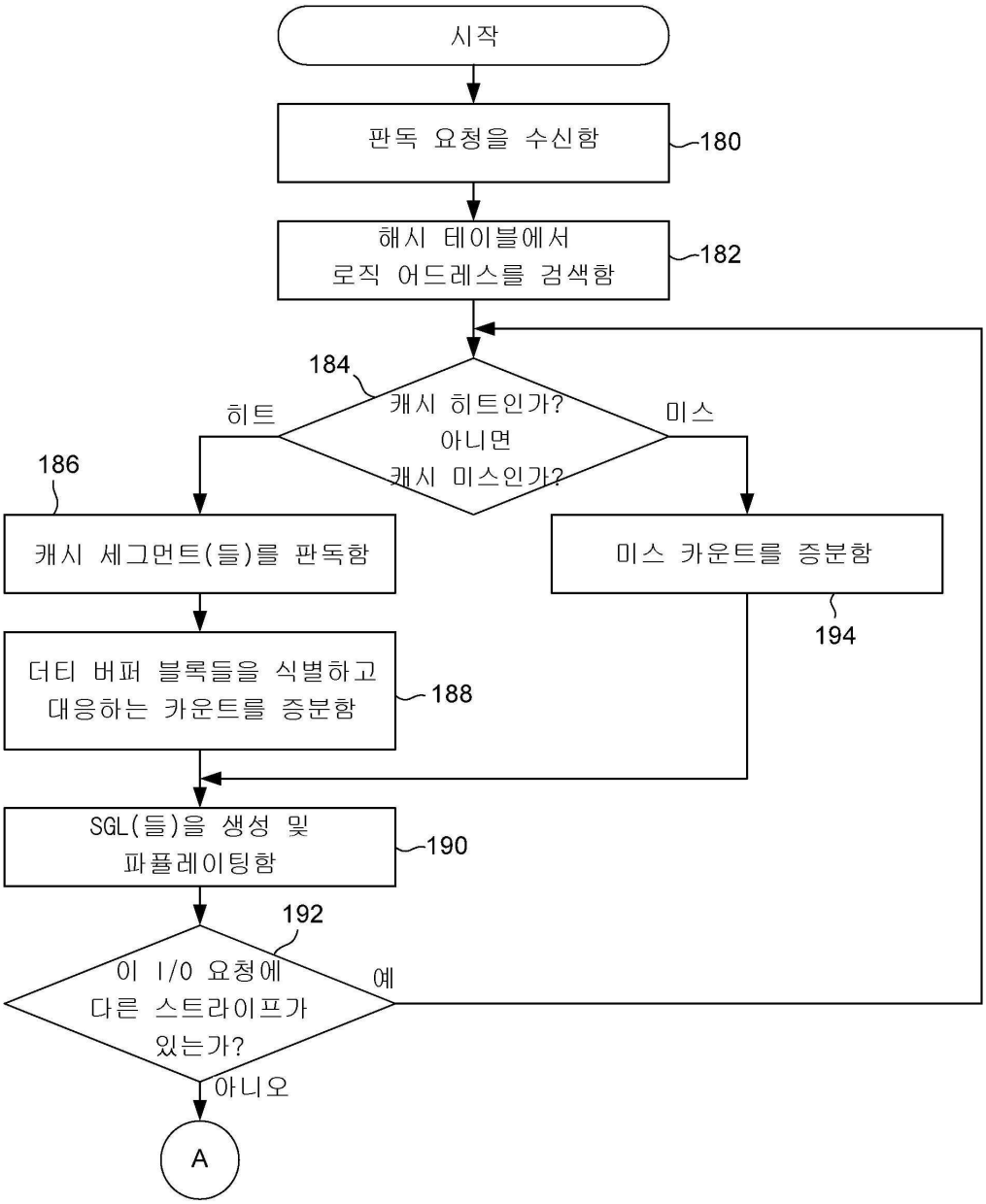
도면6a


시작
기입 요청을 수신함
136
프리 리스트로부터의
버퍼 블록(BB)들을 할당함
138
SGL(들)을 생성함
140
SGL을 사용하여 호스트로부터
캐시로 데이터를 전달함
142
해시 테이블에서
로직 어드레스들을 검색함
144
캐시 히트인가?
아니면
캐시 미스인가?
146
히트
미스
CS를 판독함
156
(CSID에 의해 식별된)
새로운 CS를 할당함
148
ISGL로부터 새로운 CS로
BBID들을 카피함
150
LRU 더티 링크된 리스트에
CSID를 부가함
152
각각의 비어있는 버퍼
블록에 대해, SGL로부터
CS의 엔트리로 BBID를
카피함; D=1, V=1,
CNT=0으로 셋팅함
158
BB가 비어있지
않은가?
예
완료
이 I/O 요청에
다른 스트라이프가
있는가?
154
예
아니오
호스트에 기입
완료를 통지함
174
종료
B
A

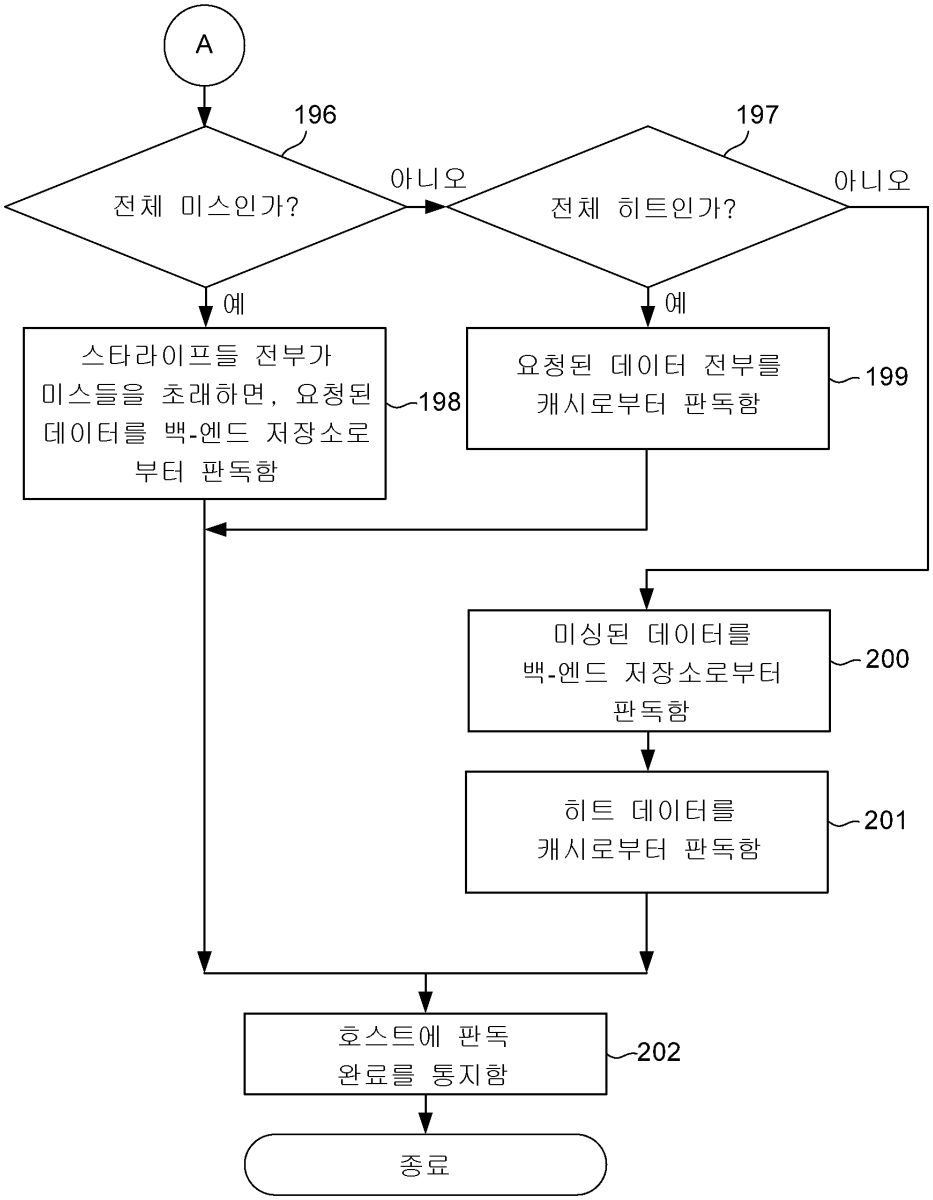
도면6b


A
FLUSH=0인가?
예
162
아니오
CNT=0인가?
예
164
아니오
BB 플래그들을 저장함
(V, D, F, CNT)
166
BBID를 프리 리스트에
다시 재-할당함
170
BBID를 ISGL로부터의
새로운 BBID로 오버라이팅함
168
LRU 더티 링크된
리스트를 업데이트함
172
B

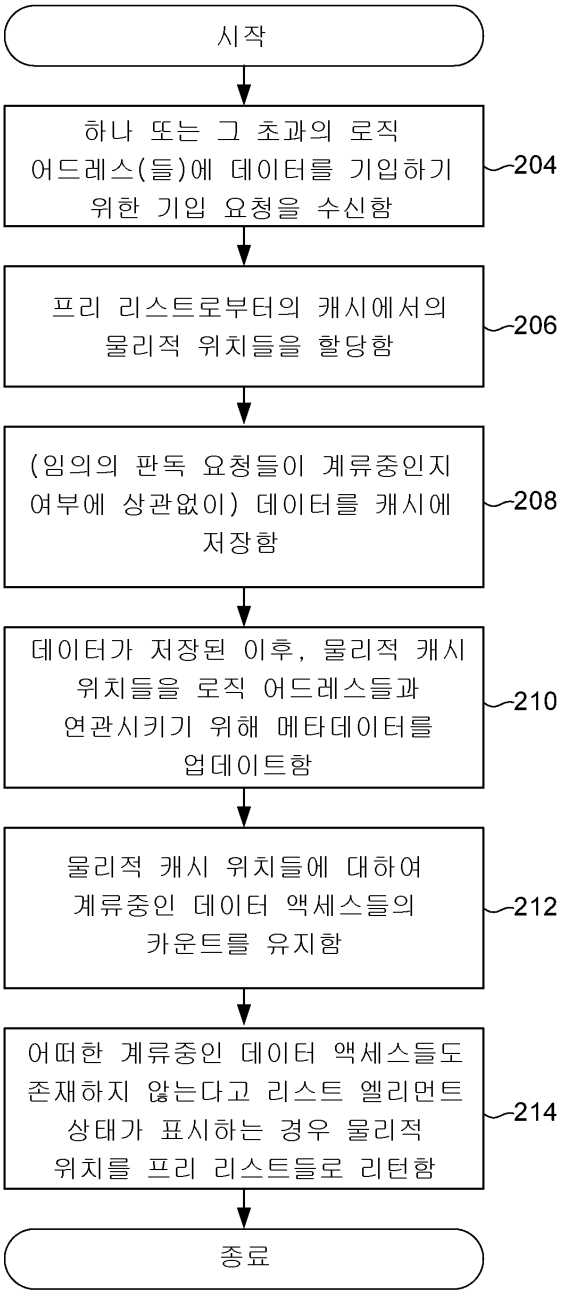
도면7a


시작
판독 요청을 수신함
180
해시 테이블에서
로직 어드레스를 검색함
182
184
캐시 히트인가?
아니면
캐시 미스인가?
히트
미스
186
캐시 세그먼트(들)를 판독함
미스 카운트를 증분함
194
더티 버퍼 블록들을 식별하고
대응하는 카운트를 증분함
188
SGL(들)을 생성 및
파퓰레이팅함
190
192
이 I/O 요청에
다른 스트라이프가
있는가?
예
아니오
A

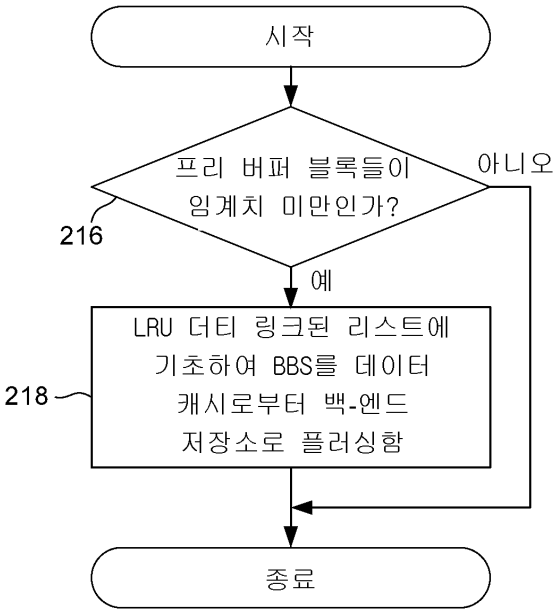
도면7b


A
196
전체 미스인가?
아니오
197
전체 히트인가?
아니오
예
예
스타라이프들 전부가 미스들을 초래하면, 요청된 데이터를 백-엔드 저장소로부터 판독함
198
요청된 데이터 전부를 캐시로부터 판독함
199
미싱된 데이터를 백-엔드 저장소로부터 판독함
200
히트 데이터를 캐시로부터 판독함
201
호스트에 판독 완료를 통지함
202
종료

도면8


시작
하나 또는 그 초과의 로직 어드레스(들)에 데이터를 기입하기 위한 기입 요청을 수신함
204
프리 리스트로부터의 캐시에서의 물리적 위치들을 할당함
206
(임의의 판독 요청들이 계류중인지 여부에 상관없이) 데이터를 캐시에 저장함
208
데이터가 저장된 이후, 물리적 캐시 위치들을 로직 어드레스들과 연관시키기 위해 메타데이터를 업데이트함
210
물리적 캐시 위치들에 대하여 계류중인 데이터 액세스들의 카운트를 유지함
212
어떠한 계류중인 데이터 액세스들도 존재하지 않는다고 리스트 엘리먼트 상태가 표시하는 경우 물리적 위치를 프리 리스트들로 리턴함
214
종료

도면9


시작
프리 버퍼 블록들이
임계치 미만인가?
216
아니오
예
LRU 더티 링크된 리스트에
기초하여 BBS를 데이터
캐시로부터 백-엔드
저장소로 플러싱함
218
종료