



등록특허 10-2401592



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년05월24일
(11) 등록번호 10-2401592
(24) 등록일자 2022년05월19일

- (51) 국제특허분류(Int. Cl.)
G06F 12/0893 (2016.01) *G06F 12/0866*
(2016.01)
- (52) CPC특허분류
G06F 12/0893 (2013.01)
G06F 12/0866 (2013.01)
- (21) 출원번호 10-2018-0116617
- (22) 출원일자 2018년09월28일
심사청구일자 2021년09월28일
- (65) 공개번호 10-2019-0088864
- (43) 공개일자 2019년07월29일
- (30) 우선권주장
62/619,723 2018년01월19일 미국(US)
15/934,940 2018년03월23일 미국(US)
- (56) 선행기술조사문현
US08732403 A

- (73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
- (72) 발명자
창, 무 티엔
미국 캘리포니아주 95051 산타클라라 비아 토리노
플레이스 2920
- 창, 앤드류
미국 캘리포니아주 94022 로스알토스 오렌지 에비
뉴 679
(뒷면에 계속)
- (74) 대리인
특허법인 고려

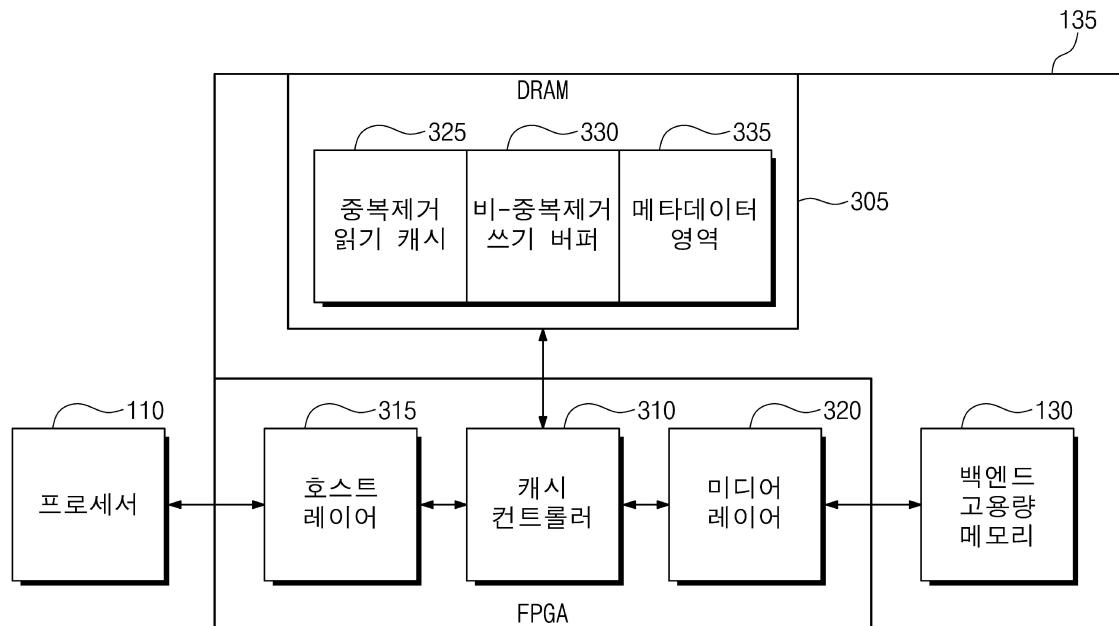
전체 청구항 수 : 총 20 항

심사관 : 안지현

(54) 발명의 명칭 중복제거 디램 캐시

(57) 요 약

본 발명의 하나의 실시 예에 따른 중복제거 캐시는 중복제거 읽기 캐시 및 비-중복제거 쓰기 버퍼를 포함하는 캐시 메모리, 중복제거 읽기 캐시를 이용하여 데이터 읽기 및 쓰기를 관리하고, 중복제거 읽기 캐시에 대한 쓰기 요청이 성공되었는지 여부를 가리키는 쓰기 상태 신호를 전송하도록 작동하는 중복제거 엔진 및 캐시 컨트롤러를
(뒷면에 계속)

대 표 도

포함하고, 캐시 컨트롤러는 요청에 포함된 어드레스가 중복제거 읽기 캐시에서 발견될 수 있는지 여부를 확인하는 캐시 히트/미스 확인 로직, 캐시 히트/미스 확인 로직이 어드레스가 중복제거 읽기 캐시에서 발견될 수 있는 것으로 가리키는 경우 캐시 메모리의 제1 데이터를 접근하는 히트 블록, 캐시 히트/미스 확인 로직이 어드레스가 중복제거 읽기 캐시에서 발견될 수 없는 것으로 가리키는 경우 백엔드 고용량 메모리의 제2 데이터를 접근하는 미스 블록 및 중복제거 읽기 캐시의 제1 데이터에 대한 접근에 대한 정보를 저장하는 히스토리 스토리지를 포함한다.

(72) 발명자

지양, 동옌

미국 캘리포니아주 95138 산호세 그라비나 코트
833

정, 홍종

미국 캘리포니아주 95032 로스 가토스 칼튼 애비뉴
120 6호

명세서

청구범위

청구항 1

중복제거(dedupable) 읽기 캐시(cache) 및 비-중복제거(non-dedupable) 쓰기 버퍼(buffer)를 포함하는 캐시 메모리;

상기 중복제거 읽기 캐시를 이용하여 데이터 읽기 및 쓰기를 관리하고, 상기 중복제거 읽기 캐시에 대한 쓰기 요청이 성공되었는지 여부를 가리키는 쓰기 상태 신호를 전송하도록 작동하는 중복제거 엔진; 및

캐시 컨트롤러를 포함하고,

상기 캐시 컨트롤러는,

요청에 포함된 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 있는지 여부를 확인하는 캐시 히트(hit)/미스(miss) 확인 로직;

상기 캐시 히트/미스 확인 로직이 상기 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 있는 것으로 가리키는 경우 상기 캐시 메모리의 제1 데이터를 접근하는 히트 블록;

상기 캐시 히트/미스 확인 로직이 상기 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 없는 것으로 가리키는 경우 백엔드(backend) 고용량 메모리의 제2 데이터를 접근하는 미스 블록; 및

상기 중복제거 읽기 캐시의 상기 제1 데이터에 대한 접근에 대한 정보를 저장하는 히스토리(history) 스토리지를 포함하는 중복제거 캐시.

청구항 2

제 1 항에 있어서,

상기 중복제거 읽기 캐시는 상기 중복제거 읽기 캐시의 물리적 용량이 지원할 수 있는 것보다 더 많은 캐시 라인들을 저장하는 것을 시뮬레이션하는 중복제거 캐시.

청구항 3

제 1 항에 있어서,

상기 캐시 메모리는 상기 중복제거 읽기 캐시의 캐시 라인들에 대한 정보를 저장하는 메타데이터 영역을 더 포함하는 중복제거 캐시.

청구항 4

제 1 항에 있어서,

상기 캐시 컨트롤러는 데이터를 쓰기 위한 제1 쓰기 요청을 프로세서로부터 수신하고, 상기 비-중복제거 쓰기 버퍼에 상기 데이터를 저장하도록 작동하고,

상기 중복제거 읽기 캐시의 캐시 라인이 상기 제1 쓰기 요청에 의해 수정되는 경우 상기 히트 블록은 상기 캐시 라인을 무효화시키도록 상기 중복제거 엔진으로 제2 쓰기 요청을 전송하도록 작동하는 중복제거 캐시.

청구항 5

제 4 항에 있어서,

상기 제2 쓰기 요청은 상기 중복제거 읽기 캐시의 상기 캐시 라인으로 0 값들을 쓰는 요청을 포함하는 중복제거 캐시.

청구항 6

제 4 항에 있어서,

상기 캐시 컨트롤러는 상기 캐시 메모리의 메타데이터 영역에 상기 캐시 라인이 유효하지 않은 것으로 표시하도록 더 작동하는 중복제거 캐시.

청구항 7

제 4 항에 있어서,

상기 히트 블록은 상기 제2 쓰기 요청에 응답하는 상기 중복제거 엔진으로부터 응답(acknowledgement) 신호를 수신하지 않고, 상기 중복제거 읽기 캐시로부터 제2 캐시 라인을 제거하고, 상기 중복제거 읽기 캐시로부터 상기 제2 캐시 라인을 제거한 후 상기 중복제거 엔진으로 상기 제2 쓰기 요청을 다시 전송하도록 더 작동하는 중복제거 캐시.

청구항 8

제 1 항에 있어서,

상기 캐시 컨트롤러는 데이터를 읽기 위한 읽기 요청을 프로세서로부터 수신하고 상기 프로세서로 상기 데이터를 전송하도록 작동하고,

상기 미스 블록은 상기 백엔드 고용량 메모리로부터 상기 데이터를 검색하고 상기 중복제거 읽기 캐시로 상기 데이터를 쓰기 위해 상기 중복제거 엔진으로 쓰기 요청을 전송하도록 작동하는 중복제거 캐시.

청구항 9

제 8 항에 있어서,

상기 미스 블록은 상기 쓰기 요청에 응답하는 상기 중복제거 엔진으로부터 확인 응답 신호를 수신하지 않고, 상기 중복제거 읽기 캐시로부터 캐시 라인을 제거하고, 상기 중복제거 읽기 캐시로부터 상기 캐시 라인을 제거한 후 상기 중복제거 엔진으로 상기 쓰기 요청을 다시 전송하도록 더 작동하는 중복제거 캐시.

청구항 10

중복제거 캐시의 쓰기 요청 처리 방법에 있어서,

데이터를 쓰기 위한 쓰기 요청을 수신하는 단계;

상기 데이터가 중복제거(dedupable) 읽기 캐시의 캐시 라인에 있는 것으로 판별하는 단계;

상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 단계; 및

상기 데이터를 비-중복제거(non-dedupable) 쓰기 버퍼에 저장하는 단계를 포함하고,

상기 중복제거 읽기 캐시는 캐시 메모리의 제1 영역이고, 상기 캐시 메모리는 제2 영역으로서 상기 비-중복제거 쓰기 버퍼를 포함하는 방법.

청구항 11

제 10 항에 있어서,

상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 단계는:

상기 캐시 메모리의 메타데이터 영역에 상기 캐시 라인이 유효하지 않은 것으로 표시하는 단계; 및

중복제거 엔진을 통해 상기 캐시 라인으로 0 값들을 쓰는 단계를 포함하는 방법.

청구항 12

제 11 항에 있어서,

상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 단계는 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답(acknowledgement)을 수신하는 단계를 더 포함하는 방법.

청구항 13

제 11 항에 있어서,

상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 단계는:

상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하지 않는 단계;

상기 중복제거 읽기 캐시로부터 제거할 제2 캐시 라인을 선택하는 단계;

상기 제2 캐시 라인을 무효화시키는 단계; 및

상기 중복제거 엔진을 통해 상기 제2 캐시 라인으로 0 값들을 쓰는 단계를 더 포함하는 방법.

청구항 14

제 10 항에 있어서,

상기 데이터를 비-중복제거 쓰기 버퍼에 저장하는 단계는 상기 데이터가 상기 중복제거 읽기 캐시의 상기 캐시 라인에 있는지 여부와 상관 없이 상기 데이터를 상기 비-중복제거 쓰기 버퍼에 저장하는 단계를 포함하는 방법.

청구항 15

중복제거 캐시의 읽기 요청 처리 방법에 있어서,

데이터를 읽기 위한 읽기 요청을 수신하는 단계;

상기 데이터가 중복제거(dedupable) 읽기 캐시의 복수의 캐시 라인들에 있지 않은 것으로 판별하는 단계;

백엔드(backend) 고용량 메모리로부터 상기 데이터를 읽는 단계;

상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 단계;

상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 단계; 및

상기 읽기 요청에 응답하여 상기 데이터를 전송하는 단계를 포함하고,

상기 중복제거 읽기 캐시는 캐시 메모리의 제1 영역이고, 상기 캐시 메모리는 제2 영역으로서 비-중복제거(non-dedupable) 쓰기 버퍼를 포함하는 방법.

청구항 16

제 15 항에 있어서,

상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 단계는 상기 중복제거 읽기 캐시에서 현재 데이터를 저장하고 있지 않은 상기 제1 캐시 라인을 선택하는 단계를 포함하는 방법.

청구항 17

제 15 항에 있어서,

상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 단계는:

상기 캐시 메모리의 메타데이터 영역으로부터의 중복제거 읽기 캐시 메타데이터 및 히스토리 데이터에 응답하여 상기 중복제거 읽기 캐시에서 제거할 현재 유효한 데이터를 저장하고 있는 상기 제1 캐시 라인을 선택하는 단계; 및

상기 제1 캐시 라인을 무효화시키는 단계를 포함하는 방법.

청구항 18

제 17 항에 있어서,

제1 캐시 라인을 무효화시키는 단계는:

상기 캐시 메모리의 상기 메타데이터 영역에 상기 제1 캐시 라인을 유효하지 않은 것으로 표시하는 단계; 및
상기 중복제거 엔진을 통해 상기 제1 캐시 라인으로 0 값들을 쓰는 단계를 포함하는 방법.

청구항 19

제 15 항에 있어서,

상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 단계는 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답(acknowledgement)을 수신하는 단계를 포함하는 방법.

청구항 20

제 15 항에 있어서,

상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 단계는:

상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하지 않는 단계;

상기 캐시 메모리의 메타데이터 영역으로부터의 중복제거 읽기 캐시 메타데이터 및 히스토리 데이터에 응답하여 상기 중복제거 읽기 캐시로부터 제거할 현재 유효한 데이터를 저장하고 있는 제2 캐시 라인을 선택하는 단계;

상기 제2 캐시 라인을 무효화시키는 단계; 및

상기 데이터를 상기 선택된 제2 캐시 라인으로 쓰는 것을 시도하기 위해 상기 중복제거 엔진으로 상기 데이터를 제공하는 단계를 포함하는 방법.

발명의 설명

기술 분야

[0001]

본 발명은 데이터 캐싱(caching)에 관한 것으로, 좀 더 상세하게는 캐시에서 중복제거 가능한 메모리에 관한 것이다.

배경 기술

[0002]

중복제거 가능한(dedupable) 메모리는 시스템에서 "이용 가능한" 메모리를 증가시키는 방식을 제공한다. 중복 데이터를 감지하고 오직 하나의 데이터 복사본을 저장함으로써, 중복 데이터 복사본을 저장하는데 이용될 수 있는 스토리지는 다른 데이터를 저장하는데 이용될 수 있다. 특히, 동일한 데이터 값에 대한 접근을 요청하는 다양한 애플리케이션들은, 비록 요청들이 다른 주소들을 이용하더라도, 모두 동일한 물리적 주소로 연결될 수 있다. 두 개, 세 개, 또는 더 많은 논리적 주소들은 동일한 물리적 주소에 매핑될 수 있기 때문에, 추가적인 데이터 복사본에 필요할 수 있는 스토리지는 다른 데이터를 저장하는데 이용될 수 있어 메모리가 물리적으로 저장 가능한 것보다 더 많은 전체 데이터를 저장하는 것처럼 보이게 할 수 있다.

[0003]

그러나 중복제거 가능한 메모리도 자체적인 문제들을 야기한다. 어느 주소가 동일한 데이터를 가리키는지 관리하는 것은 어렵고, 읽기 및 쓰기와 같은 데이터 접근들은 결국 다른 형태들의 스토리지에 비해 느릴 수 있다. 상대적으로 접근하는 것이 느린 데이터를 이용하는 경우(하드 디스크 드라이브와 같은) 반드시 큰 어려움이 있는 것은 아니지만, 더 빠른 스토리지 장치들에 적용되는 경우 이러한 지연은 상당한 제약일 수 있다. 스토리지 장치가 일반적으로 더 빠르게 동작할수록, 데이터 중복의 전체적인 영향은 더 커질 수 있다. 예를 들어, 동적 웨이퍼 액세스 메모리(DRAM; Dynamic Random Access Memory) 접근들은 솔리드 스테이트 디스크(SSD; Solid State Disk)를 또는 다른 형태의 플래시 메모리보다 데이터 중복에 연관된 지연들에 대해 훨씬 더 민감할 수 있고, 이는 하드 디스크 드라이브들보다 지연에 더 민감하다.

[0004]

캐시에서 중복제거 가능한 메모리를 이용하는 방식은 백엔드(backend) 메모리의 데이터에 대한 접근 속도를 높이기 위해 필요할 수 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 상술된 기술적 과제를 해결하기 위한 것으로써, 중복제거 가능한 메모리를 이용하는 캐시를 제공할 수 있다.

과제의 해결 수단

[0006] 본 발명의 하나의 실시 예에 따른 중복제거 캐시는 중복제거(dedupable) 읽기 캐시(cache) 및 비-중복제거(non-dedupable) 쓰기 버퍼(buffer)를 포함하는 캐시 메모리, 상기 중복제거 읽기 캐시를 이용하여 데이터 읽기 및 쓰기를 관리하고, 상기 중복제거 읽기 캐시에 대한 쓰기 요청이 성공되었는지 여부를 가리키는 쓰기 상태 신호를 전송하도록 작동하는 중복제거 엔진 및 캐시 컨트롤러를 포함하고, 상기 캐시 컨트롤러는 요청에 포함된 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 있는지 여부를 확인하는 캐시 히트(hit)/미스(miss) 확인 로직, 상기 캐시 히트/미스 확인 로직이 상기 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 있는 것으로 가리키는 경우 상기 캐시 메모리의 제1 데이터를 접근하는 히트 블록, 상기 캐시 히트/미스 확인 로직이 상기 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 없는 것으로 가리키는 경우 백엔드(backend) 고용량 메모리의 제2 데이터를 접근하는 미스 블록 및 상기 중복제거 읽기 캐시의 상기 제1 데이터에 대한 접근에 대한 정보를 저장하는 히스토리(history) 스토리지를 포함한다.

[0007] 본 발명의 하나의 실시 예에 따른 방법은 데이터를 쓰기 위한 쓰기 요청을 수신하는 단계, 상기 데이터가 중복제거 읽기 캐시의 캐시 라인에 있는 것으로 판별하는 단계, 상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 단계 및 상기 데이터를 비-중복제거 쓰기 버퍼에 저장하는 단계를 포함하고, 상기 중복제거 읽기 캐시는 캐시 메모리의 제1 영역이고, 상기 캐시 메모리는 제2 영역으로서 상기 비-중복제거 쓰기 버퍼를 포함한다.

[0008] 본 발명의 하나의 실시 예에 따른 방법은 데이터를 읽기 위한 읽기 요청을 수신하는 단계, 상기 데이터가 중복제거 읽기 캐시의 복수의 캐시 라인들에 있지 않은 것으로 판별하는 단계, 백엔드 고용량 메모리로부터 상기 데이터를 읽는 단계, 상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 단계, 상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 단계 및 상기 읽기 요청에 응답하여 상기 데이터를 전송하는 단계를 포함하고, 상기 중복제거 읽기 캐시는 캐시 메모리의 제1 영역이고, 상기 캐시 메모리는 제2 영역으로서 비-중복제거 쓰기 버퍼를 포함한다.

발명의 효과

[0009] 본 발명에 따른 캐시는 더 많은 데이터를 동일한 물리적 양의 메모리에 저장할 수 있다.

[0010] 또한, 본 발명에 따른 캐시는 중복제거 가능한 메모리에 대한 쓰기 고유의 지연을 방지할 수 있다.

도면의 간단한 설명

[0011] 도 1은 본 발명의 하나의 실시 예에 따른 중복제거 캐시를 사용하는 머신을 보여준다.

도 2는 도 1의 머신의 추가적인 세부 구성 요소들을 보여준다.

도 3은 도 1의 중복제거 캐시의 예시적인 레이아웃을 보여준다.

도 4는 중복제거 엔진이 포함된 도 1의 중복제거 캐시를 보여준다.

도 5는 도 3 및 도 4의 캐시 컨트롤러의 세부 구성 요소들을 보여준다.

도 6a 및 도 6b는 본 발명의 하나의 실시 예에 따른 도 1의 중복제거 캐시가 쓰기 요청을 처리하기 위한 예시적인 절차의 순서도를 보여준다.

도 7은 본 발명의 하나의 실시 예에 따른 도 1의 중복제거 캐시가 도 3의 중복제거 읽기 캐시에서의 캐시 라인을 무효화하기 위한 예시적인 절차의 순서도를 보여준다.

도 8a 내지 도 8c는 본 발명의 하나의 실시 예에 따른 도 1의 중복제거 캐시가 읽기 요청을 처리하기 위한 예시적인 절차의 순서도를 보여준다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 발명의 실시 예들에 대하여 참조가 상세하게 기재될 것이고, 첨부된 도면들에서 참조의 예시들이 설명된다.

후술되는 상세한 설명에서, 발명의 완벽한 이해를 가능하게 하기 위해 다양한 특정 세부 사항들이 기재된다. 그러나, 본 기술 분야에서의 통상의 기술자가 이러한 특정 세부 사항들 없이 발명을 실시할 수 있음을 자명할 것이다. 다른 경우들에서, 잘 알려진 방법들, 절차들, 구성 요소들, 회로들, 그리고 네트워크들은 실시 예들을 불필요하게 모호하도록 하지 않기 위해 상세하게 설명되지 않는다.

[0013] 제1, 제2 등의 용어들은 본 명세서에서 다양한 요소들을 설명하기 위해 사용 되더라도, 이러한 요소들은 이러한 용어들에 의해 제한되지 않는 것으로 이해될 것이다. 이러한 용어들은 오직 하나의 요소를 다른 요소와 구별하기 위해 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서, 제1 모듈은 제2 모듈로 칭해질 수 있고, 제2 모듈은 제1 모듈로 칭해질 수 있다.

[0014] 본 명세서에서 발명의 설명에서 사용되는 용어는 오직 특정 실시 예들을 설명하기 위한 목적으로 본 발명을 제한하려는 의도는 아니다. 발명의 설명과 첨부된 청구항들에서 사용된 바와 같이, 문맥이 명백하게 다른 것을 가리키지 않으면, 단수 형태의 용어는 복수 형태를 포함하도록 의도된다. 본 명세서에서 "그리고/또는" 용어는 열거된 항목들과 관련된 하나 이상의 가능한 조합들을 모두 지칭하는 것으로 이해될 것이다. "포함하다" 그리고/또는 "포함하는" 용어는 본 명세서에서 사용되는 경우, 기술된 특징, 정수들, 단계들, 동작들, 요소들, 그리고/또는 구성 요소들의 존재를 명시하는 것으로 이해될 것이다. 그러나 하나 이상의 다른 특징들, 정수들, 단계들, 동작들, 요소들, 구성 요소들, 그리고/또는 그것의 그룹들의 추가 또는 존재를 못하게 하는 것은 아니다. 도면들의 구성 요소들 및 특징들은 반드시 축적에 맞춰져 있지 않다.

[0015] 중복제거된 DRAM은 주어진 DRAM 유닛 내에서 향상된 논리적 용량을 제공하지만, 일반 DRAM 동작들보다 일반적으로 느린다. 이러한 느린 성능은 중복제거된 DRAM을 일반 DRAM 캐시로서 사용하지 못하게 할 수 있다.

[0016] 이러한 우려들을 해결하기 위해, 중복제거된 DRAM 캐시의 메모리는 중복제거 DRAM 읽기 캐시 및 중복제거되지 않은 쓰기 버퍼로 나눠질 수 있다. 중복제거 DRAM에 대한 쓰기는 느리기 때문에, 기존 DRAM은 쓰기 버퍼(WB)로 이용될 수 있다. 읽기 요청은 중복제거된 읽기 캐시(RC)로부터 (대부분) 제공될 수 있지만, 쓰기는 보통의 DRAM 쓰기 버퍼(WB)에 의해 제공될 수 있다.

[0017] 예를 들어, 물리적 DRAM이 8기가바이트(GB)이면, 물리적 DRAM은 4 GB 물리적 중복제거 DRAM RC(총 8GB 가상 용량을 제공하는) 및 4 GB 물리적 DRAM WB로 분할될 수 있다. 중복제거 엔진은 오직 중복제거 DRAM RC에 의해 이용되는 특정 4 GB 범위에 대하여 중복제거 매커니즘(mechanism)을 동작시킬 수 있다.

[0018] 그러나 중복제거 DRAM이 읽기 캐시로서 이용될 수 있더라도, 쓰기는 중복제거 DRAM에 여전히 영향을 줄 수 있다. 예를 들어, 호스트(host)로부터의 64 바이트(B) 업데이트는 중복제거 DRAM RC에 저장된 데이터에 영향을 줄 수 있거나(쓰기 히트(hit)), 데이터가 현재 중복제거 DRAM RC에 없는 경우(읽기 미스(miss)), 백엔드 고 용량 메모리로부터의 2킬로바이트(KB)가 필요할 수 있다. 중복제거 DRAM RC로 쓰는 상황을 해결하기 위해, 새로운 WR_Status 신호가 이용될 수 있다: 쓰기가 성공하면 중복제거 엔진은 ACK를 반환하고, 그렇지 않으면 중복제거 엔진은 NAK를 반환한다. 캐시 컨트롤러가 NAK를 수신하면, 캐시 컨트롤러는 캐시 채우기를 취소할 것이다(즉, 2 KB 채우기는 캐시되지 않음).

[0019] 따라서, 중복제거 DRAM RC 쓰기 히트 처리를 위해, 중복제거 DRAM 캐시는 캐시 라인(cache line)이 유효하지 않도록 RC 메타데이터(metadata)를 업데이트할 수 있고, 64 B의 0들을(캐시 라인을 유효하지 않도록 하는)을 중복제거 엔진으로 기입할 수 있다(이것은 중복제거 DRAM RC를 가비지 컬렉션(garbage collection)하는 것과 연관될 수 있음). 이러한 처리가 완료된 후, 중복제거 DRAM 캐시는 중복제거 DRAM RC 쓰기 히트가 없었던 것처럼 진행할 수 있다(즉, 중복제거 DRAM 캐시는 중복제거 DRAM RC 쓰기 미스가 있었던 것처럼 진행할 수 있음).

[0020] 중복제거 DRAM RC 채우기를 처리하기 위해, 중복제거 DRAM 캐시는 중복제거 DRAM RC로부터 제거할 캐시 라인을 선택하기 위해 히스토리 및 RC 메타데이터를 읽을 수 있다. 선택된 캐시 라인이 유효한 데이터를 포함하면, 중복제거 DRAM 캐시는 캐시 라인을 유효하지 않은 것으로 표시하기 위해 RC 메타데이터를 업데이트하고 캐시 라인이 유효하지 않도록 2KB의 0들을 중복제거 엔진으로 기입할 수 있다. 그리고(선택된 캐시 라인이 유효하지 않으면), 중복제거 DRAM 캐시는 중복제거 엔진으로 새로운 캐시 라인 데이터를 기입할 수 있고, 중복제거 엔진은 WR_Status 신호를 반환한다. 중복제거 엔진이 ACK를 반환하면, 중복제거 DRAM 캐시는 캐시 라인을 유효한 것으로 표시하기 위해 RC 메타데이터를 업데이트할 수 있다. 그렇지 않으면, 중복제거 엔진이 NAK를 반환하면, 중복제거 DRAM 캐시는 중복제거 DRAM RC로 데이터를 기입하기 위한 요청을 드롭(drop)시킬 수 있다.

[0021] 위의 본 발명은 중복제거 메모리를 사용하여 잠재적으로 낮은 쓰기 성능의 문제를 해결하기 위해 효과적이다. 그러나, 중복제거 가능한 메모리의 사용은 다른 문제를 야기한다: 그것은 쓰기 동작이 보장되지 않는다는 것이다

다. 예를 들어, 중복제거 DRAM RC 채우기 동작이 수행되지 않지만, 중복제거 DRAM RC의 모든 캐시 라인은 유효한 데이터를 저장하는 것으로 가정한다. 무효화를 위해 선택된 캐시 라인이 중복제거된 데이터를 가리키면, 즉, 캐시 라인에 의해 가리켜진 데이터가 다른 캐시 라인들에 의해 가리켜지면, 캐시 라인을 무효화하는 것이 임의의 물리적 메모리를 확보하지 않을 것이다. 또는, 데이터가 다른 캐시 라인에 의해 사용된다면, 중복제거 RC 쓰기 히트 시, 중복제거 엔진이 캐시 라인을 0 들로 덮어쓰기하지 못할 수 있다. 이러한 쓰기 완료 보장 부족의 문제에 대한 두 개의 해결책이 있다: 쓰기가 보장되지 않음을 수용하거나, 쓰기 동작 완료를 보장하도록 쓰기 동작을 다시 시도하는 것이다.

[0022] 쓰기 동작을 다시 시도하는 것을 달성하기 위해, 쓰기 재시도가 수행될 수 있다. 쓰기가 실패하면, 즉, 중복제거 엔진이 시도된 쓰기로부터 NAK 신호를 반환하면, 다른 캐시 라인은 무효화를 위해 선택될 수 있고(중복제거 DRAM 캐시의 구현에 따라, 라이트백(writeback)이 포함될 수 있음) 쓰기가 다시 시도된다.

[0023] 다른 캐시 라인을 무효화하는 것은 중복제거 엔진이 중복제거 DRAM RC로 쓰기 동작을 완료할 수 있도록 야기할 수 있으므로, 쓰기 재시도가 성공할 수 있다. 그러나 쓰기 재시도의 부분으로서 무효화를 위해 선택된 캐시 라인 또한 중복제거 대상인 데이터를 가리키면, 쓰기 재시도가 다시 실패할 것이다. 그러므로, 캐시 라인을 무효화하고 쓰기를 재시도하는 프로세스는 원하는 횟수만큼 반복적으로 수행될 수 있다. 일반적으로, 재시도 프로세스에 대한 종료 조건들은 성공적인 쓰기(중복제거 엔진으로부터의 ACK 신호에 의해 가리켜지는 바와 같이), 한 세트에서 모든 캐시 라인이 제외됨, 또는 미리 결정된 수의 쓰기 재시도이다. 모든 캐시 라인이 제외되거나 미리 결정된 수의 쓰기 재시도 중 하나가 발생되면, 중복제거 DRAM 캐시는 원하는 쓰기 동작을 성공적으로 완료하지 않고 프로세서로 결과를 반환하는 것으로 기본 설정(default)될 수 있다.

[0024] 도 1은 본 발명의 하나의 실시 예에 따른 중복제거 캐시를 사용하는 머신(machine)을 보여준다. 도 1에 있어서, 머신(105)이 도시된다. 머신(105)은 프로세서(110)를 포함할 수 있다. 프로세서(110)는 임의의 다양한 프로세서일 수 있다. 예를 들어, 프로세서(110)는 Intel Xeon, Celeron, Itanium, 또는 Atom 프로세서, AMD Opteron 프로세서, ARM 프로세서 등일 수 있다. 머신(105)에 있어서, 도 1이 하나의 프로세서(110)를 보여주지만, 머신(105)은 임의의 수의 프로세서들을 포함할 수 있고, 각각의 프로세서는 단일 코어 또는 멀티 코어 프로세서들일 수 있고, 임의의 원하는 조합으로 혼합될 수 있다. 프로세서(110)는 장치 드라이버(driver)(115)를 실행할 수 있고, 장치 드라이버(115)는 스토리지 장치(120)에 대한 접근을 지원할 수 있다: 다른 장치 드라이버들은 머신(105)의 다른 구성 요소들에 대한 접근을 지원할 수 있다.

[0025] 머신(105)은 메모리 컨트롤러(125)를 포함할 수 있고, 메모리 컨트롤러(125)는 메인 메모리(130)에 대한 접근을 관리하기 위해 사용될 수 있다. 메모리(130)는 플래시 메모리, DRAM(Dynamic Random Access Memory), SRAM(Static Random Access Memory), PRAM(Persistent Random Access Memory), FRAM(Ferroelectric Random Access Memory), MRAM(Magnetoresistive Random Access Memory), 또는 RNRAM(Non-Volatile Random Access Memory) 등과 같이 임의의 다양한 메모리일 수 있다. 메모리(130)는 다른 메모리 타입들의 임의의 원하는 조합일 수 있다. 머신(105)은 아래에 설명된 바와 같이 중복제거 캐시(135)를 더 포함할 수 있다.

[0026] 도 1이 서버(서버는 독립형(standalone) 또는 랙(rack) 서버일 수 있음)로서 머신(105)을 도시하지만, 본 발명의 실시 예들은 제한 없이 임의의 원하는 타입의 머신(105)을 포함할 수 있다. 예를 들어, 머신(105)은 데스크탑 또는 랩탑 컴퓨터 또는 본 발명의 실시 예들로부터 혜택을 받을 수 있는 임의의 다른 장치로 대체될 수 있다. 머신(105)은 특화된 휴대 컴퓨팅 장치들, 태블릿 컴퓨터들, 스마트폰들 및 다른 컴퓨팅 장치들을 포함할 수 있다.

[0027] 도 2는 도 1의 머신(105)의 추가적인 세부 사항들을 보여준다. 도 2에 있어서, 일반적으로, 머신(105)은 하나 이상의 프로세서(110)를 포함하고, 프로세서(110)는 메모리 컨트롤러(125) 및 클럭(205)을 포함할 수 있고, 클럭(205)은 머신(105)의 구성 요소들의 동작을 조정하는데 이용될 수 있다. 프로세서(110)는 메모리(130)에 연결될 수 있고, 메모리(130)는, 예로서, RAM(random access memory), ROM(read-only memory) 또는 다른 상태 보존 미디어를 포함할 수 있다. 프로세서(110)는 스토리지 장치(120) 및 네트워크 커넥터(connector)(210)에 연결될 수 있고, 네트워크 커넥터(210)는, 예를 들어, 이더넷(Ethernet) 커넥터 또는 무선 커넥터일 수 있다. 프로세서(110)는 버스(215)에 연결될 수 있고, 버스(215)에는, 다른 구성 요소들 중에서, 유저 인터페이스(220) 및 입출력(Input/Output) 엔진(225)을 이용하여 관리될 수 있는 입출력 포트가 부착될 수 있다.

[0028] 도 3은 도 1의 중복제거 캐시(135)의 예시적인 레이아웃을 보여준다. 도 3에 있어서, 중복제거 캐시(135)는 네 개의 일반적인 구성 요소들로 분할될 수 있다: DRAM(305), 캐시 컨트롤러(310), 호스트 레이어(layer)(315), 미디어 레이어(320), DRAM(305)은 중복제거 캐시(135)에 대해 실제 메모리로서 작동하고, 세 개의 영역들로 분할

될 수 있다: 중복제거 읽기 캐시(325), 비-중복제거 쓰기 버퍼(330), 메타데이터 영역(335). 중복제거 읽기 캐시(325)는 프로세서(110)에 의해 중복제거 캐시(135)로부터 읽혀질 수 있는 데이터를 저장하기 위한 중복제거 메모리로서 사용될 수 있다. 비-중복제거 쓰기 버퍼(330)는 프로세서(110)로부터 쓰여지는 데이터를 저장하기 위한 기준의(즉, 비-중복제거) 메모리로서 사용될 수 있다. 메타데이터 영역(335)은 중복제거 읽기 캐시(325) 및 비-중복제거 쓰기 버퍼(330)의 캐시 라인들(cache lines)에 대한 정보를 저장할 수 있다: 예를 들어, 메타데이터 영역(335)은 어떠한 캐시 라인들이 유효한지 또는 유효하지 않은지, 및 어떠한 캐시 라인들이 백엔드 고용량 메모리(130)로 쓰기 위해 버퍼링(buffering)되는지를 저장할 수 있다. 도 3에는 중복제거 엔진이 도시되지 않았으나, 중복제거 엔진은 중복제거 읽기 캐시(325)로 실제 데이터 쓰기를 관리한다: 중복제거 엔진은 도 4를 참조하여 아래에서 설명된다. 중복제거 읽기 캐시(325), 비-중복제거 쓰기 버퍼(330) 및 메타데이터 영역(335)은 DRAM(305)의 임의의 원하는 비율을 포함할 수 있다. 예를 들어, DRAM(305)이 총 8 기가바이트(GB)를 포함하면, 중복제거 읽기 캐시(325) 및 비-중복제거 쓰기 버퍼(330)는 각각 대략 스토리지의 4 GB를 포함할 수 있다(메타데이터 영역(335)은 상대적으로 DRAM(305)의 적은 양을 요구함). 예상되는 중복제거 비율을 고려하면, 중복제거 읽기 캐시(325)는 물리적 용량보다 더 큰 용량을 시뮬레이션할 수 있다. 예를 들어, 중복제거 읽기 캐시(325)가 4 GB의 물리적 메모리를 포함하고 2.0의 타겟 중복제거 비율을 가지면, 중복제거 읽기 캐시(325)는 8 GB의 가상 메모리를 시뮬레이션할 수 있다. 중복제거 캐시(135)는 중복제거 읽기 캐시(325)의 가상 용량만큼 캐시 라인들을 지원하고, 이는 중복제거 읽기 캐시(325)의 물리적 용량이 지원할 수 있는 캐시 라인들의 수를 초과할 수 있다.

[0029] 도 3이 중복제거 읽기 캐시(325) 및 비-중복제거 쓰기 버퍼(330)를 모두 포함하는 DRAM(305)을 도시하고 중복제거 읽기 캐시(325) 및 비-중복제거 쓰기 버퍼(330)가 다른 기능들을 제공하지만, 일부 기능이 중복될 수 있다. 예를 들어, 본 발명의 몇몇 실시 예들에 있어서, 비-중복제거 쓰기 버퍼(330)는 프로세서(110)로부터 쓰여진 데이터를 저장할 수 있지만(그리고 결국에는 백엔드 고용량 메모리(130)로 쓰여질 수 있음), 비-중복제거 쓰기 버퍼(330)는 "읽기 캐시"로서 동작할 수 있고(비록 중복제거 대상이 아닐지라도), 비-중복제거 쓰기 버퍼(330)로부터 데이터가 읽혀질 수 있다. 따라서, 비-중복제거 쓰기 버퍼(330)는 백엔드 고용량 메모리(130)로 쓰여지는 데이터를 위한 "일시적인(transient)" 스토리지 이상으로 간주될 수 있다. 이러한 본 발명의 실시 예들에 있어서, 특정 캐시 라인에 대하여 중복제거 읽기 캐시(325)를 확인하는 것에 대한 어떠한 언급은 비-중복제거 쓰기 버퍼(330)로부터의 캐시 라인에 대한 확인 및 중복제거 읽기 캐시(325) 또는 비-중복제거 쓰기 버퍼(330)로부터 적절하게 데이터를 읽는 것으로 이해될 수 있고, 비-중복제거 쓰기 버퍼(330)가 가득 차거나 비-중복제거 쓰기 버퍼(330)의 캐시 라인이 새로운 캐시 라인을 위한 공간을 만들기 위해 무효화 되는 경우에만 데이터는 백엔드 고용량 메모리(130)로 쓰여질 수 있다. 이러한 본 발명의 실시 예들에 있어서, 데이터가 중복제거 읽기 캐시(325) 또는 비-중복제거 쓰기 버퍼(330)에 저장되지 않은 경우에만 백엔드 고용량 메모리(130)는 요청된 데이터에 대하여 접근될 수 있다.

[0030] 캐시 컨트롤러(310)는 중복제거 캐시(135) 뒤에서 "두뇌"로서 역할을 할 수 있고, 캐시 컨트롤러(310)는 어떤 데이터가 쓰여지거나 읽혀지는지를 관리하고, 무효화 또는 쓰기에 대한 캐시 라인을 선택하는 등의 역할을 할 수 있다. 캐시 컨트롤러(310)는 프로세서(110)와 상호작용(즉, 프로세서(110)로부터 요청들/데이터를 수신하고 프로세서(110)로 데이터를 전송)하기 위해 호스트 레이어(315)를 이용할 수 있고, 메모리(130)(도 3의 "백엔드 고용량 메모리"로 표시됨)와 상호작용하기 위해 미디어 레이어(320)를 이용할 수 있다. 메모리(130)는 중복제거 캐시(135)를 포함하는 모듈의 일부분이거나, 중복제거 캐시(135)로부터 분리될 수 있다(그리고 통신 경로를 통해 통신됨).

[0031] 도 3에 도시된 바와 같이, 캐시 컨트롤러(310), 호스트 레이어(315) 및 미디어 레이어(320)는 FPGA(field programmable gate array)를 이용하여 구현될 수 있다. 그러나 본 발명의 실시 예들은 다른 하드웨어 디자인(PROM(Programmable Read Only Memory) 또는 EPROM(Erasable PROM) 또는 EEPROM(Electrically Erasable PROM)과 같은 변형 메모리, 또는 특별히 설계된 회로) 및 소프트웨어 디자인을 포함하는 임의의 원하는 구현을 지원할 수 있다. 또한, 도 3이 단일 FPGA를 이용하여 구현된 캐시 컨트롤러(310), 호스트 레이어(315) 및 미디어 레이어(320)를 도시하지만, 본 발명의 실시 예들은 네트워크 상호 연결과 같은 통신 경로를 사용하여 연결된 여러 분리된 구성 요소들을 이용하는 구현을 지원할 수 있다.

[0032] 도 4는 중복제거 엔진을 가지는 도 1의 중복제거 캐시(135)를 보여준다. 도 4에 있어서, DRAM(305), 캐시 컨트롤러(310), 호스트 레이어(315) 및 미디어 레이어(320)는 도 3에 도시된 구성 요소들과 동일할 수 있고, 메모리 컨트롤러(125)는 도 1의 메모리 컨트롤러(125)와 동일할 수 있다. 그러나 도 4에는 그밖에, 중복제거 엔진(405) 및 네트워크 인터커넥트(interconnect)(410)가 도시된다. 네트워크 인터커넥트(410)는 DRAM(305), 캐시 컨트롤러(310), 호스트 레이어(315) 및 미디어 레이어(320)와 같은 다른 구성 요소들 사이에 통신 경로를 제공하는 역할을 한다.

러(310) 및 중복제거 엔진(405) 사이에 통신을 제공할 수 있다. 중복제거 엔진(405)은 중복제거된 메모리(도 3의 중복제거 읽기 캐시(325)와 같은)를 포함하는 DRAM(305)의 부분들로부터의 데이터 읽기 및 DRAM(305)의 부분들에 데이터 쓰기를 관리할 수 있다(중복제거된 메모리를 포함하지 않는 DRAM(305)의 부분들은, 중복제거 엔진(405)을 이용하지 않고, 직접 접근될 수 있음).

[0033] 상술한 바와 같이, 중복제거 엔진(405)은 중복제거된 메모리를 관리한다. 이를 위해, 중복제거 엔진(405)은 중복제거 엔진(405)이 중복제거를 수행하도록 지시하는 신호들(도 4의 "Dedup"로 표시된 신호) 및 접근을 위한 어드레스와 사용을 위한 데이터를 명시하는 읽기/쓰기 명령들(도 4의 "RD/WR"로 표시된 신호)을 지원할 수 있다. 그러나, 중복제거 엔진(405)은 또한 쓰기 상태 신호(도 4의 "WR_Status"로 표시된 신호)를 제공할 수 있다. 쓰기 상태 신호는 특정 쓰기 동작이 성공인지 또는 실패인지 여부를 나타내는데 이용될 수 있다. 쓰기 상태 신호의 이용은 도 5를 참조하여 아래에서 더 설명된다.

[0034] 도 5는 도 3 및 도 4의 캐시 컨트롤러(310)의 세부 사항을 보여준다. 도 5에 있어서, 캐시 컨트롤러(310)는 위험 관리자(505), 캐시 히트/미스 확인(510), 히트 블록(515), 미스 블록(520) 및 히스토리 스토리지(525)를 포함할 수 있다. 도 1의 중복제거 캐시(135)에 의해 도 1의 프로세서(110)로부터의 데이터 요청을 수신하는 경우, 위험 관리자(505)는 데이터 종속성이 정확하게 처리되도록 하기 위해 읽기-후-쓰기 및 쓰기-후-읽기와 같은 커맨드들의 다양한 순서를 추적할 수 있다. 예를 들어, 캐시 컨트롤러(310)가 도 3의 비-중복제거 쓰기 베퍼(330)에 저장된 데이터를 쓰는 요청을 수신하고, 도 1의 백엔드 고용량 메모리(130)로 쓰여지기 전에 그 데이터를 읽는 요청을 수신하는 경우, 위험 관리자(505)는 도 3의 비-중복제거 쓰기 베퍼(330)로부터의 데이터에 접근하는 읽기 요청을 전달할 수 있다. 캐시 히트/미스 확인(510)은 접근되는 특정 어드레스가 도 1의 중복제거 캐시(135)에서 찾아질 수 있는지 여부를 판별할 수 있다. 이러한 판별은 도 3의 메타데이터 영역(335)에 접근함으로써 수행될 수 있으므로, 캐시 히트/미스 확인(510)은 도 3의 메타데이터 영역(335)에 접근하기 위해 읽기/쓰기 신호에 따라 읽기 요청을 전송할 수 있다.

[0035] 캐시 히트/미스 확인(510)이 특정 요청이 이미 도 1의 중복제거 캐시(135)에 있는 어드레스에 접근하는지 여부를 판별하면, 제어는, 적절하게, 히트 블록(515) 또는 미스 블록(520)으로 전달될 수 있다. 히트 블록(515)은 도 1의 중복제거 캐시(135)에 저장된 현재 캐시 라인에 접근하기 위해 이용될 수 있다. 미스 블록(520)은 도 3의 백엔드 고용량 메모리(130)로부터의 데이터에 접근하기 위해 이용될 수 있다. 히트 블록(515) 및 미스 블록(520)은 도 3의 DRAM(305)에 있는 데이터에 접근하기 위한 읽기/쓰기 신호에 따라 읽기 및/또는 쓰기 요청을 전송할 수 있다. 또한, 미스 블록(520)은 도 4의 중복제거 엔진(405)으로부터 쓰기 상태 신호를 수신할 수 있다. 마지막으로, 히스토리 스토리지(525)는 도 1의 중복제거 캐시(135)에 있는 캐시 라인의 접근 히스토리에 대한 정보를 판별하는데 이용될 수 있다(예를 들어, 적절한 경우 캐시 라인을 제외시키는 것으로 선택하기 위해). 또한, 도 5에 도시된 네트워크 인터커넥트들과 같이, 다양한 네트워크 인터커넥트들은 캐시 컨트롤러(310)의 다양한 요소들 사이에 통신을 지원할 수 있다.

[0036] 도 3의 중복제거 캐시(135) 및 캐시 컨트롤러(310)의 하드웨어가 설명되었고, 도 3의 중복제거 캐시(135)의 동작이 설명될 수 있다. 도 1의 중복제거 캐시(135)가 도 1의 프로세서(110)로부터 요청을 수신하는 경우, 두 가지 타입의 요청들 및 두 가지 가능한 캐시 결과들이 있다: 요청은 읽기 요청 또는 쓰기 요청일 수 있고, 문제의 데이터는 캐시 히트 또는 캐시 미스를 야기할 수 있다. 따라서 총 4 개의 발생 가능한 경우들이 있으며, 각각의 경우가 아래에서 설명된다.

읽기 요청, 캐시 히트

[0038] 도 1의 프로세서(110)가 읽기 요청을 발행하고, 데이터가 현재 도 3의 중복제거 읽기 캐시(325)에 저장되면, 도 5의 캐시 히트/미스 확인(510)은 도 3의 메타데이터 영역(335)을 읽은 후에 이 경우를 판별할 것이다. 문제의 데이터가 현재 도 3의 중복제거 읽기 캐시(325)에 있으므로, 제어는 도 5의 히트 블록(515)으로 전달할 수 있고, 히트 블록(515)은, 데이터가 도 3의 중복제거 읽기 캐시(325)로부터 읽혀지는 것을 요청하는 읽기 요청을 도 4의 중복제거 엔진(405)으로 전송할 수 있다. 도 4의 중복제거 엔진(405)은 데이터를 도 5의 히트 블록(515)으로 반환할 수 있고, 히트 블록(515)은 호스트 레이어(315)를 통해 도 1의 프로세서(110)로 데이터를 반환할 수 있다.

읽기 요청, 캐시 미스

[0040] 도 1의 프로세서(110)가 읽기 요청을 발행하고, 데이터가 현재 도 3의 중복제거 읽기 캐시(325)에 저장되어 있지 않으면, 도 5의 캐시 히트/미스 확인(510)은 도 3의 메타데이터 영역(335)을 읽은 후에 이 경우를 판별할 것

이다. 문제의 데이터가 현재 도 3의 중복제거 읽기 캐시(325)에 없으므로, 제어는 미스 블록(520)으로 전달할 수 있고, 미스 블록(520)은 도 3의 미디어 레이어(320)를 통해 도 3의 백엔드 고용량 메모리(130)로부터의 데이터를 요청할 수 있다.

[0041] 데이터가 읽혀지면, 도 5의 미스 블록(520)은 도 3의 중복제거 읽기 캐시(325)에 현재 유효한 데이터를 저장하고 있지 않은 캐시 라인이 존재하는지 여부를 판별할 수 있다. 도 3의 중복제거 읽기 캐시(325)에 현재 유효한 데이터를 저장하고 있지 않은 이용 가능한 캐시 라인이 존재하면, 도 5의 미스 블록(520)은 데이터를 저장하기 위해 그런 캐시 라인을 선택할 수 있다. 그렇지 않으면, 도 5의 미스 블록(520)은 도 3의 중복제거 읽기 캐시(325)로부터 제거하기 위한 유효한 데이터를 포함하는 캐시 라인을 선택할 수 있다. 이 선택 프로세스는 도 5의 히스토리 스토리지(525)로부터의 데이터를 이용할 수 있고, 제거하기 위한 캐시 라인을 선택하는데 임의의 원하는 알고리즘을 사용할 수 있다: LRU(Least Recently Used) 또는 LFU(Least Frequently Used)는 제거하기 위한 캐시 라인을 선택하는데 사용될 수 있는 잘 알려진 알고리즘들의 예시들이다.

[0042] 도 3의 중복제거 읽기 캐시(325)로부터 캐시 라인을 제거하기 위해, 도 5의 미스 블록(520)은 캐시 라인이 유효하지 않은 것으로 표시하는 것을 도 3의 메타데이터 영역(335)으로 쓰기 위해 도 3의 DRAM(305)으로 쓰기 동작을 전송할 수 있다. 미스 블록(520)은 도 3의 중복제거 읽기 캐시(325)에 실제 데이터를 겹쳐쓰기 위해 도 4의 중복제거 엔진(405)으로 쓰기 동작을 전송할 수 있다: 예를 들어, 캐시 라인을 채우기 위해 충분히 0 값들을 쓸 수 있다.

[0043] 캐시 라인이 제거되고 데이터가 무효화되면, 캐시 라인이 처음부터 이용 가능했던 것처럼, 캐시 라인은 자유롭게 데이터를 수신할 수 있다. 따라서, 도 5의 미스 블록(520)은 데이터(도 1의 백엔드 고용량 메모리(130)로부터 먼저 읽혀진)를 도 3의 중복제거 읽기 캐시(325)로 쓰기 위해 도 4의 중복제거 엔진(405)으로 쓰기 동작을 전송할 수 있다. 도 3의 캐시 컨트롤러(310)는 도 3의 호스트 레이어(315)를 통해 도 1의 프로세서(110)로 데이터를 반환할 수 있다.

쓰기 요청, 캐시 히트

[0045] 도 1의 프로세서(110)가 쓰기 요청을 발행하고, 데이터가 현재 도 3의 중복제거 읽기 캐시(325)에 저장되어 있으면, 도 5의 캐시 히트/미스 확인(510)은 도 3의 메타데이터 영역(335)을 읽은 후에 이 경우를 판별할 것이다. 문제의 데이터가 현재 도 3의 중복제거 읽기 캐시(325)에 있으므로, 제어는 히트 블록(515)으로 전달할 것이다.

[0046] 도 3의 중복제거 읽기 캐시(325)는 현재 교체될 데이터를 저장하므로, 도 3의 중복제거 읽기 캐시(325)의 캐시 라인의 데이터도 처리되어야 한다. 새로운 데이터가 도 3의 중복제거 읽기 캐시(325)의 캐시 라인으로 쓰여질 수 있지만, 중복제거된 메모리로 데이터를 쓰는 것은 상대적으로 느린 동작이다(그리고 도 1의 백엔드 고용량 메모리(130)에 저장된 데이터는 어느 경우에서도 업데이트될 필요가 있다). 따라서, 도 3의 중복제거 읽기 캐시(325)로 데이터를 쓰는 대신에, 도 3의 중복제거 읽기 캐시(325)의 캐시 라인은 무효화될 수 있다. 이 방식에서, 데이터가 프로세서(110)에 의해 나중에 읽히면, 새로운 값들은 도 3의 비-중복제거 쓰기 버퍼(330) (여전히 그곳에 존재하면) 또는 도 1의 백엔드 고용량 메모리(130)(위의 "읽기 요청, 캐시 미스"로 표시된 경우에서 설명된 바와 같이)로부터 검색될 수 있다.

[0047] 도 3의 중복제거 읽기 캐시(325)로부터 캐시 라인을 무효화하기 위해, 도 5의 히트 블록(515)은 캐시 라인을 유효하지 않은 것으로 표시하는 것을 도 3의 메타데이터 영역(335)으로 쓰기 위해 도 3의 DRAM(305)으로 쓰기 동작을 전송할 수 있다. 히트 블록(515)은 도 3의 중복제거 읽기 캐시(325)에 실제 데이터를 겹쳐쓰기 위해 도 4의 중복제거 엔진(405)으로 쓰기 동작을 전송할 수 있다: 예를 들어, 캐시 라인을 채우기 위해 충분한 0 값들을 쓸 수 있다.

[0048] 데이터가 도 3의 중복제거 읽기 캐시(325)로부터 무효화되면, 도 5의 히트 블록(515)은 도 3의 비-중복제거 쓰기 버퍼(330)로 데이터를 쓸 수 있다. 도 3의 비-중복제거 쓰기 버퍼(330)가 중복제거된 메모리를 사용하지 않으므로, 도 3의 비-중복제거 쓰기 버퍼(330)로 데이터를 쓰는 것은 도 3의 중복제거 읽기 캐시(325)로 데이터를 쓰는 것보다 빠르다. 이후, 예를 들어, 도 3의 비-중복제거 쓰기 버퍼(330)가 모두 채워지면, 적절한 시간에, 데이터는 도 3의 미디어 레이어(320)를 통해 도 3의 비-중복제거 쓰기 버퍼(330)로부터 도 3의 백엔드 고용량 메모리(130)로 플러시(flush)될 수 있다.

쓰기 요청, 캐시 미스

[0050] 도 1의 프로세서(110)가 쓰기 요청을 발행하고, 데이터가 현재 도 3의 중복제거 읽기 캐시(325)에 저장되어 있지 않으면, 도 5의 캐시 히트/미스 확인(510)은 도 3의 메타데이터 영역(335)을 읽은 후 이 경우를 판별할 것이

다. 문제의 데이터가 도 3의 중복제거 읽기 캐시(325)에 있지 않으므로, 제어는 미스 블록(520)으로 전달할 수 있다.

[0051] 도 5의 미스 블록(520)은 도 3의 비-중복제거 쓰기 버퍼(330)로 데이터를 쓸 수 있다. 이후, 예를 들어, 도 3의 비-중복제거 쓰기 버퍼(330)가 모두 채워지면, 적절한 시간에서, 데이터는 도 3의 미디어 레이어(320)를 통해 도 3의 비-중복제거 쓰기 버퍼(330)로부터 도 3의 백엔드 고용량 메모리(130)로 플러시될 수 있다.

[0052] 위의 설명은 전체 성능을 향상시키기 위해 캐시에서 어떻게 중복제거 메모리가 이용될 수 있는지를 보여준다. 그러나 중복제거 메모리의 사용은 특히 읽기 요청, 캐시 미스 및 쓰기 요청, 캐시 히트의 경우들을 야기시킬 수 있는 다른 문제를 발생시킬 수 있다. 이러한 두 경우들에서, 데이터는 도 3의 중복제거 읽기 캐시(325)로 쓰여질 수 있다. 문제는 도 3의 중복제거 읽기 캐시(325)가 물리적 용량보다 더 큰 용량을 시뮬레이션하는 사실로부터 생긴다. 캐시 라인이 이용 가능할 지라도, 도 3의 중복제거 읽기 캐시(325)의 물리적 메모리가 모두 채워질 수 있고, 이러한 경우에 데이터는 성공적으로 도 3의 중복제거 읽기 캐시(325)에 쓰여지지 않을 수 있다. 또한, 도 3의 중복제거 읽기 캐시(325)의 캐시 라인을 무효화하는 것은 임의의 물리적 메모리를 확보하지 않을 수 있다. 예를 들어, 무효화를 위해 선택된 캐시 라인이 다른 캐시 라인에 의해 참조되는 데이터를 참조하면, 선택된 캐시 라인을 무효화하는 것은 실제로 데이터를 도 3의 중복제거 읽기 캐시(325)의 물리적 메모리로부터 자유롭게 하지 않는다. 이러한 문제는 쓰기 보장의 부족으로 설명될 수 있다.

[0053] 한가지 해결책은 도 3의 중복제거 읽기 캐시(325)로 쓰기가 실패하는 것을 간단하게 수용하는 것이다. 이러한 해결책은, 적어도 읽기 요청, 캐시 미스의 경우에, 합리적이다: 최악의 경우에, 도 1의 백엔드 고용량 메모리(130)로부터의 접근을 요구하면서(적어도 데이터가 도 3의 중복제거 읽기 캐시(325)에 실제로 캐시될 수 있을 때까지), 적어도 데이터가 도 3의 중복제거 읽기 캐시(325)에 실제로 캐시될 수 있을 때까지 데이터는 실제로 캐시되지 않는다. 그리고 쓰기 요청, 캐시 히트의 경우에서도, 동일한 데이터를 접근하는 모든 캐시 라인들을 무효화하는 방법(도 3의 중복제거 읽기 캐시(325)로부터 부실 데이터(stale data)에 접근하는 캐시 라인들에 대한 접근을 막기 위해)이 있으면 해결책이 수용될 수 있다. 그러나 다른 해결책들이 존재한다.

[0054] 다른 해결책은 데이터가 성공적으로 도 3의 중복제거 읽기 캐시(325)에 쓰여질 수 있을 때까지(또는 재시도 임계 횟수에 도달할 때까지), 여러 캐시 라인들을 무효화하는 것이다. 도 3의 중복제거 읽기 캐시(325)로 데이터를 쓰는 것을 시도한 후, 도 4의 중복제거 엔진(405)이 쓰기 상태 신호에서 NAK를 반환하면, 도 3의 캐시 컨트롤러(310)(도 5의 히트 블록(515) 또는 도 5의 미스 블록(520)을 통해)는 무효화를 위해 도 3의 중복제거 읽기 캐시(325)로부터 캐시 라인을 선택할 수 있다. 선택된 캐시 라인이 무효화된 후, 도 3의 캐시 컨트롤러(310)(다시, 도 5의 히트 블록(515) 또는 도 5의 미스 블록(520)을 통해)는 도 3의 중복제거 읽기 캐시(325)로 원래의 쓰기를 재시도할 수 있다. 이러한 프로세스는, 도 3의 중복제거 읽기 캐시(325)로 쓰기가 성공하거나(쓰기 상태 신호에서 ACK에 의해 가리켜지는 바와 같이) 재시도의 임계 횟수가 발생될 때까지 필요한 만큼 반복될 수 있다.

[0055] 도 6a 및 도 6b는 본 발명의 하나의 실시 예에 따른 도 1의 중복제거 캐시(135)가 쓰기 요청을 처리하는 예시적인 절차의 순서도를 보여준다. 도 6a에 있어서, 605 블록에서, 도 3의 캐시 컨트롤러(310)는 도 1의 프로세서(110)로부터 쓰기 요청을 수신할 수 있다. 610 블록에서, 도 5의 캐시 히트/미스 확인(510)은 도 3의 중복제거 읽기 캐시(325)가 겹쳐쓰기되는 데이터를 포함하는 캐시 라인을 포함하는지 여부를 판별할 수 있다. 그렇지 않으면, 615 블록에서 도 5의 미스 블록(520)은 도 3의 비-중복제거 쓰기 버퍼(330)에 쓰기 요청으로부터의 데이터를 저장할 수 있다. 620 블록에서, 도 3의 캐시 컨트롤러(310)는 도 3의 비-중복제거 쓰기 버퍼(330)로부터 도 1의 백엔드 고용량 메모리(130)로 데이터를 플러시할 수 있다. 그리고 625 블록에서, 도 3의 캐시 컨트롤러(310)는 도 3의 비-중복제거 쓰기 버퍼(330)로부터 데이터를 삭제할 수 있다.

[0056] 반면에, 610 블록에서, 도 3의 중복제거 읽기 캐시(325)가 쓰기 요청에 의해 겹쳐쓰기된 데이터를 포함하는 캐시 라인을 포함하면, 630 블록에서(도 6b) 도 5의 히트 블록(515)은 캐시 라인을 무효화할 수 있다(도 3의 메타데이터 영역(335)에 캐시 라인을 유효하지 않은 것으로 표시하고 도 4의 중복제거 엔진(405)을 통해 캐시 라인으로 0 값들을 기입함으로써). 635 블록에서, 도 5의 히트 블록(515)은 도 4의 중복제거 엔진(405)이 ACK 또는 NAK 신호로 응답하는지 여부를 판별할 수 있다. 도 4의 중복제거 엔진(405)이 ACK 신호를 반환하면, 프로세싱은 쓰기 요청을 완료하기 위해 도 6a의 615 블록으로 계속될 수 있다. 재시도의 최대 횟수에 도달하였으면, 640 블록에서, 도 5의 히트 블록(515)은 재시도의 최대 횟수에 도달하였는지 여부를 판별할 수 있다. 재시도의 최대 횟수에 도달하지 않았으면, 645 블록에서, 도 5의 히트 블록(515)은 무효화를 위해 다른 캐시 라인을 선택할 수 있고, 프로세싱은 새롭게 선택된 캐시 라인을 무효화하기 위해 630 블록으로 돌아갈 수 있다. 그렇지 않으면, 650 블록에서, 히트 블록(515)(그리고 도 3의 캐시 컨트롤러(310))는 도 3의 중복제거 읽기 캐시(325)로부터의

데이터를 삭제하는 문제가 있었다고 보고할 수 있고, 그 후 프로세싱이 종료될 수 있다.

[0057] 도 7은 본 발명의 하나의 실시 예에 따른 도 1의 중복제거 캐시(135)가 도 3의 중복제거 읽기 캐시(325)의 캐시 라인을 무효화하기 위한 예시적인 절차의 순서도를 보여준다. 도 7에 있어서, 705 블록에서, 도 3의 캐시 컨트롤러(310)는 도 3의 메타데이터 영역(335)에서 캐시 라인을 유효하지 않은 것으로 표시할 수 있다(적절한 값을 도 3의 메타데이터 영역(335)으로 기입함으로써). 710 블록에서, 도 3의 캐시 컨트롤러(310)는, 부실 데이터가 DRAM 공간을 차지하지 않을 수 있도록, 문제의 캐시 라인의 데이터로 0 값을 쓰기 위해 도 4의 중복제거 엔진(405)으로 쓰기 요청을 전송할 수 있다.

[0058] 도 8a 내지 도 8c는 본 발명의 하나의 실시 예에 따른 도 1의 중복제거 캐시(135)가 읽기 요청을 처리하는 예시적인 절차의 순서도를 보여준다. 도 8a에 있어서, 805 블록에서, 도 3의 캐시 컨트롤러(310)는 도 1의 백엔드 고용량 메모리(130)(또는 도 3의 DRAM(305)에 데이터가 저장된다면, 도 3의 DRAM(305))로부터 데이터를 읽기 위한 읽기 요청을 수신할 수 있다. 810 블록에서, 도 5의 캐시 히트/미스 확인(510)은 요청된 데이터가 도 3의 중복제거 읽기 캐시(325)에서 발견되는지 여부를 확인할 수 있다. 요청된 데이터가 도 3의 중복제거 읽기 캐시(325)의 캐시 라인에서 발견된다면, 815 블록에서, 도 5의 히트 블록(515)은 요청된 데이터를 도 3의 중복제거 읽기 캐시(325)의 캐시 라인으로부터 읽기 위해 도 4의 중복제거 엔진(405)으로 읽기 요청을 전송할 수 있고, 820 블록에서, 도 3의 캐시 컨트롤러(310)는 요청된 데이터를 도 1의 프로세서(110)로 다시 전송할 수 있고, 이후, 프로세싱이 종료된다.

[0059] 요청된 데이터가 도 3의 중복제거 읽기 캐시(325) 또는 비-중복제거 쓰기 버퍼(330)의 캐시 라인에서 발견되지 않으면, 825 블록에서, 도 5의 미스 블록(520)은 도 1의 백엔드 고용량 메모리(130)로부터 데이터를 읽을 수 있다. 830 블록에서, 도 5의 미스 블록(520)은 데이터를 저장하기 위해 도 3의 중복제거 읽기 캐시(325)로부터 캐시 라인을 선택할 수 있다. 선택된 캐시 라인은 현재 유효한 데이터를 저장하고 있지 않은 캐시 라인일 수 있거나, 유효한 데이터를 저장한 캐시 라인(따라서 먼저 무효화가 필요함)일 수 있다. 835 블록에서(도 8b), 도 5의 미스 블록(520)은 선택된 캐시 라인이 유효한 데이터를 포함하는지 포함하지 않는지 여부를 판별한다. 선택된 캐시 라인이 유효한 데이터를 포함하면, 840 블록에서, 도 5의 미스 블록(520)은, 위의 도 7에서 서술된 예시적인 절차를 이용하여, 캐시 라인을 무효화할 수 있다. 캐시 라인이 유효한 데이터를 포함하지 않으면, 845 블록에서, 도 5의 미스 블록(520)은 도 3의 중복제거 읽기 캐시(325)의 선택된 캐시 라인으로 요청된 데이터를 쓰기 위해 도 4의 중복제거 엔진(405)으로 쓰기 요청을 전송할 수 있다. 즉, 845 블록에서, 도 5의 미스 블록(520)은 도 3의 중복제거 읽기 캐시(325)의 선택된 캐시 라인으로 요청된 데이터를 쓰는 것을 시도하기 위해 도 4의 중복제거 엔진(405)으로 요청된 데이터를 제공할 수 있다.

[0060] 850 블록에서, 도 5의 미스 블록(520)은 도 4의 중복제거 엔진(405)이 쓰기 요청에 응답하여 ACK 또는 NAK 중 어느 것을 반환하는지 여부를 판별할 수 있다. 도 3의 중복제거 읽기 캐시(325)가 자유로운 캐시 라인을 가지더라도 최대 물리적 용량에 도달했을 수 있으므로, 도 4의 중복제거 엔진(405)은 835 블록에서 선택된 캐시 라인이 유효한 데이터를 포함하는지 여부와 상관 없이 NAK를 반환할 수 있다. 도 4의 중복제거 엔진(405)이 ACK 신호를 반환하면, 쓰기는 성공했다: 855 블록에서, 도 5의 미스 블록(520)은 도 3의 메타데이터 영역(335)의 캐시 라인을 유효한 데이터를 포함하는 것으로 표시할 수 있고, 프로세싱은 도 1의 프로세서(110)로 요청된 데이터를 되돌려 보내기 위해 도 8a의 820 블록으로 계속될 수 있다.

[0061] 반면에, 도 4의 중복제거 엔진(405)이 NAK 신호를 반환하면, 도 4의 중복제거 엔진(405)은 도 3의 중복제거 읽기 캐시(325)의 캐시 라인으로 데이터를 쓸 수 없었다. 이러한 경우에 있어서, 860 블록에서(도 8c), 도 5의 미스 블록(520)은 재시도의 최대 횟수가 도달되었는지 여부를 판별할 수 있다. 재시도의 최대 횟수가 아직 도달되지 않았으면, 865 블록에서, 도 5의 미스 블록(520)은 무효화를 위한 다른 캐시 라인을 선택할 수 있고, 프로세싱은 새롭게 선택된 캐시 라인을 무효화하기 위해 840 블록으로 돌아갈 수 있다. 그렇지 않으면, 870 블록에서, 미스 블록(520)(그리고 도 3의 캐시 컨트롤러(310))는 도 3의 중복제거 읽기 캐시(325)로 데이터를 쓰기에 문제가 생겼다고 보고할 수 있고, 이후 프로세싱이 종료될 수 있다.

[0062] 도 3을 참조하여 위에서 설명된 바와 같이, 본 발명의 몇몇 실시 예들에 있어서, 도 3의 비-중복제거 쓰기 버퍼(330)는, 도 3의 중복제거 읽기 캐시(325)와 함께 기존의 캐시로서 동작할 수 있다. 이러한 본 발명의 실시 예들에 있어서, 요청된 캐시 라인을 위해 도 3의 중복제거 읽기 캐시(325)를 확인하거나 접근하는 것은 유사하게 도 3의 비-중복제거 쓰기 버퍼(330)를 확인하거나 접근하는 것을 포함하는 것으로 이해될 수 있다. 예를 들어, 도 8a에 있어서, 810 블록은 데이터를 포함하는 캐시 라인을 위해 도 3의 중복제거 읽기 캐시(325) 및 도 3의 비-중복제거 쓰기 버퍼(330) 모두를 확인하기 위해 수정될 수 있고, 815 블록은, 실제로 데이터가 어디에서 발

견되는지에 따라, 도 3의 중복제거 읽기 캐시(325) 또는 도 3의 비-중복제거 쓰기 버퍼(330)의 캐시 라인으로부터 데이터를 읽는 것으로 수정될 수 있다. 유사하게, 도 8c의 870 블록에 있어서, 데이터가 성공적으로 도 3의 중복제거 읽기 캐시(325)에 쓰여지지 않는다면, 데이터는 대신에 도 3의 비-중복제거 쓰기 버퍼(330)로 쓰여질 수 있다. 또는, 예시 순서도는 도 3의 중복제거 읽기 캐시(325)의 캐시 라인으로 먼저 데이터를 쓰는 것을 시도하기 보다, 도 3의 중복제거 읽기 캐시(325) 또는 도 3의 비-중복제거 쓰기 버퍼(330)로부터의 데이터가 쓰여질 캐시 라인을 선택하는 것으로 수정될 수 있다.

[0063] 도 6a 내지 도 8c에 있어서, 본 발명의 몇몇 실시 예들이 도시된다. 그러나 기술분야에서 숙련된 사람은, 블록들의 순서를 바꾸거나, 블록들을 생략하거나 도면에서 도시되지 않은 관계들을 포함함으로써, 본 발명의 다른 실시 예들이 가능하다는 것을 인정할 것이다. 명시적으로 설명되거나 설명되지 않았더라도, 순서도의 이러한 모든 변형은 본 발명의 실시 예들로 고려될 수 있다.

[0064] 본 발명의 실시 예들은 선행기술보다 몇 가지 기술적 이점을 제공한다. 먼저, 도 1의 중복제거 캐시(135)의 중복제거 메모리의 사용은, 더 많은(고유한) 데이터가 동일한 물리적 양의 메모리에 저장되도록 하면서(또는, 동일한 양의 데이터를 저장하기 위해 더 적은 메모리 공간을 사용하면서), 여러 복사본의 동일한 데이터가 도 3의 캐시 DRAM(305)에 저장되는 것을 막는다. 예를 들어, 도 3의 중복제거 읽기 캐시(325)가 2의 예상된 중복제거 비율을 가지는 4 GB의 메모리를 포함하면, 도 3의 중복제거 읽기 캐시(325)는 이론적으로 8 GB의 비-중복제거 캐시 메모리에 저장될 수 있는 만큼 고유한 데이터를 저장할 수 있다. 두 번째로, 도 3의 비-중복제거 쓰기 버퍼(330)를 사용함으로써, 애플리케이션들이 데이터를 쓰고 있는 상황들에 대하여(단순히 데이터를 읽는 상황보다) 중복제거 메모리에 대한 쓰기 고유의 지연이 방지된다. 세 번째로, 보장되지 않은 도 3의 중복제거 읽기 캐시(325)로의 쓰기에 대한 염려는 도 3의 중복제거 읽기 캐시(325)의 캐시 라인을 무효화시킨 후 쓰기 재시도를 지원함으로써 감소된다(도 3의 중복제거 읽기 캐시(325)로의 쓰기가 몇몇 캐시 라인 무효화 이후에도 성공하지 않을 가능성이 여전히 존재하지만, 이러한 발생 가능성은 매우 낮다).

[0065] 아래의 내용은 본 발명의 특정 실시 예들이 구현될 수 있는 적합한 장치 또는 장치들의 간략하고 일반적인 설명을 제공하기 위한 것이다. 장치 또는 장치들은 다른 장치들로부터 수신된 지시들, 가상 현실(VR; virtual reality) 환경과의 상호 작용, 생체 피드백, 또는 다른 입력 신호에 의한 것뿐만 아니라 키보드들, 마우스 등과 같은 종래의 입력 장치들로부터의 입력에 의해, 적어도 어느 정도는 제어될 수 있다. 본 명세서에서 사용된 바와 같이, 용어 "장치"는 단일 장치, 가상 머신, 또는 장치들과 통신으로 연결된 시스템, 가상 머신들, 또는 함께 동작하는 장치들을 포괄적으로 포함하는 것을 의미한다. 예시적인 장치들은 예를 들어, 자동차들, 기차들, 택시들 등과 같은 사적 또는 공적 운송 수단과 같은 운송 장치들뿐만 아니라, 개인용 컴퓨터들, 워크스테이션들, 서버들, 휴대용 컴퓨터들, 휴대용 장치들, 전화기들, 테블릿들 등과 같은 컴퓨팅 장치들을 포함한다.

[0066] 장치 또는 장치들은 프로그램 가능하거나 프로그램 가능하지 않은 로직 장치들 또는 로직 어레이들, 응용 주문형 집적 회로들(ASICs; Application Specific Integrated Circuits), 임베디드 컴퓨터들, 스마트 카드들 등과 같은 임베디드 컨트롤러들을 포함할 수 있다. 장치 또는 장치들은 네트워크 인터페이스, 모뎀, 또는 다른 통신 연결을 통해 하나 이상의 원격 장치들에 하나 이상의 연결을 활용할 수 있다. 장치들은 인트라넷, 인터넷, 근거리 네트워크들, 광역 네트워크들 등과 같은 물리적 그리고/또는 논리적 네트워크를 이용하여 상호 연결될 수 있다. 통상의 기술자는 네트워크 통신이 라디오 주파수(RF), 위성, 마이크로웨이브(microwave), IEEE(Institute of Electrical and Electronics Engineers) 802.11, 블루투스(bluetooth), 광, 적외선, 케이블, 레이저 등을 포함하는 다양한 유선 그리고/또는 무선 근거리 또는 원거리 운반체들 및 프로토콜(protocol)들을 활용할 수 있다는 것을 이해할 것이다.

[0067] 본 발명의 실시 예들은 기능들, 절차들, 데이터 구조들, 애플리케이션 프로그램들 등을 포함하는 관련 데이터와 같이 또는 관련 데이터를 참조함으로써 설명될 수 있다. 관련 데이터는 장치에 의해 접근된 경우 장치가 작업들을 수행하거나 추상적 데이터 유형들 또는 하위 레벨 하드웨어 컨텍스트들(contexts)을 정의하게 한다. 관련 데이터는, 예를 들어, RAM, ROM 등과 같은 휘발성 그리고/또는 불휘발성 메모리, 또는 하드 드라이브들, 플로피 디스크들, 광학 스토리지, 테이프들, 스토리지, 메모리 스틱들, 디지털 비디오 디스크들, 생물학적 스토리지 등을 포함하는 다른 스토리지 장치들 및 관련된 스토리지 미디어에 저장될 수 있다. 관련 데이터는 패킷들(packets), 직렬 데이터, 병렬 데이터, 전파된 신호들 등의 형태로 물리적 그리고/또는 논리적 네트워크를 포함하는 전송 환경들을 통해 전달될 수 있고, 압축되거나 암호화된 형태로 이용될 수 있다. 관련 데이터는 분산 환경에서 이용될 수 있고, 장치 접근을 위해 로컬(local)로 그리고/또는 원격으로 저장될 수 있다.

- [0068] 본 발명의 실시 예들은 하나 이상의 프로세서들에 의해 실행될 수 있는 명령어들을 포함하는 유형의 비일시적인 장치 관독 가능한 기록 매체를 포함할 수 있다. 명령어들은 본 명세서에서 설명된 대로 본 발명의 요소들을 수행하는 명령들을 포함한다.
- [0069] 상술된 다양한 방법의 동작들은, 다양한 하드웨어 및/또는 소프트웨어 구성(들), 회로들, 및/또는 모듈(들)과 같이, 동작들을 수행할 수 있는 적절한 수단에 의해 수행될 수 있다. 소프트웨어는 논리적 기능들을 수행하기 위한 실행 명령어들의 순서 목록을 포함할 수 있고, 단일 또는 멀티 코어 프로세서 또는 프로세서를 포함하는 시스템과 같이, 명령어 실행 시스템, 장치, 또는 디바이스에 의해 또는 연결되어 사용되기 위한 임의의 "프로세서 관독 가능 매체"에 포함될 수 있다.
- [0070] 본 명세서에서 개시된 실시 예들에 연관되어 설명된 블록들, 방법의 단계들 또는 알고리즘 및 기능들은 하드웨어에서, 프로세서에 의해 실행되는 소프트웨어 모듈에서, 또는 그 둘의 조합에서 직접 구현될 수 있다. 소프트웨어에서 구현되면, 기능들은 컴퓨터 관독 가능한 기록 매체에 하나 이상의 명령들 또는 코드로서 저장되거나 전송될 수 있다. 소프트웨어 모듈은 RAM(Random Access Memory), 플래시 메모리, ROM(Read Only Memory), EPROM(Electrically Programmable ROM), EEPROM(Electrically Erasable Programmable ROM), 레지스터들, 하드디스크, 제거 가능한 디스크, CD ROM 또는 알려진 임의의 다른 형태의 스토리지 매체에 있을 수 있다.
- [0071] 설명된 실시 예들을 참조하여 본 발명의 원리들을 설명한 바와 같이, 설명된 실시 예들은 이러한 원리들로부터 벗어나지 않고 배치 및 세부 사항에 있어서 변형될 수 있고, 임의의 방식으로 결합될 수 있다. 그리고, 상술한 내용은 특정 실시 예들에 초점이 맞춰졌을 지라도, 다른 구성들이 고려될 수 있다. 특히, "본 발명의 하나의 실시 예에 따라"와 같은 표현들 또는 본 명세서에서 사용된 이와 같은 표현들이라도, 이러한 문구들은 일반적으로 가능한 실시 예를 참조하기 위한 의미이고, 본 발명을 특정 실시 예의 구성으로 한정하려는 것은 아니다. 본 명세서에서 사용된 바와 같이, 이러한 용어들은 다른 실시 예들로 결합할 수 있는 동일한 또는 다른 실시 예들을 참조할 수 있다.
- [0072] 앞에서 설명한 실시 예들은 본 발명을 제한하는 것으로 해석되지 않는다. 비록 몇몇 실시 예들이 설명되었지만, 통상의 기술자들은 본 발명의 새로운 개시를 및 이점들로부터 벗어나지 않는 실시 예들에 대해 다양한 변형을 쉽게 할 수 있을 것이다. 이에 따라, 이러한 변형들은 청구항들에서 정의된 바와 같이 본 발명의 권리 범위 내에 포함된 것으로 의도된다.
- [0073] 본 발명의 실시 예들은 제한 없이, 다음과 같은 서술들로 확장될 수 있다.
- [0074] 서술 1. 본 발명의 하나의 실시 예에 다른 중복제거 캐시는:
- [0075] 중복제거 읽기 캐시 및 비-중복제거 쓰기 버퍼를 포함하는 캐시 메모리;
- [0076] 상기 중복제거 읽기 캐시를 이용하여 데이터 읽기 및 쓰기를 관리하고, 상기 중복제거 읽기 캐시에 대한 쓰기 요청이 성공되었는지 여부를 가리키는 쓰기 상태 신호를 전송하도록 작동하는 중복제거 엔진; 및
- [0077] 캐시 컨트롤러를 포함하고;
- [0078] 상기 캐시 컨트롤러는,
- [0079] 요청에 포함된 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 있는지 여부를 확인하는 캐시 히트/미스 확인 로직;
- [0080] 상기 캐시 히트/미스 확인 로직이 상기 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 있는 것으로 가리키는 경우 상기 캐시 메모리의 제1 데이터를 접근하는 히트 블록;
- [0081] 상기 캐시 히트/미스 확인 로직이 상기 어드레스가 상기 중복제거 읽기 캐시에서 발견될 수 없는 것으로 가리키는 경우 백엔드 고용량 메모리의 제2 데이터를 접근하는 미스 블록; 및
- [0082] 상기 중복제거 읽기 캐시의 상기 제1 데이터에 대한 접근에 대한 정보를 저장하는 히스토리 스토리지를 포함한다.
- [0083] 서술 2. 서술 1에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 캐시 컨트롤러는 상기 중복제거 읽기 캐시 및 상기 비-중복제거 쓰기 버퍼에서 데이터 종속성을 관리하는 위험 관리자를 더 포함한다.
- [0084] 서술 3. 서술 1에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐

시는 상기 중복제거 읽기 캐시에 물리적으로 적합한 것보다 더 많은 캐시 라인들을 저장하는 것을 시뮬레이션한다.

[0085] 서술 4. 서술 1에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 캐시 메모리는 상기 중복제거 읽기 캐시의 캐시 라인들에 대한 정보를 저장하는 메타데이터 영역을 더 포함한다.

[0086] 서술 5. 서술 1에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서:

[0087] 상기 캐시 컨트롤러는 데이터를 쓰기 위한 제1 쓰기 요청을 프로세서로부터 수신하고, 상기 비-중복제거 쓰기 버퍼에 상기 데이터를 저장하도록 작동하고;

[0088] 상기 중복제거 읽기 캐시의 캐시 라인이 상기 제1 쓰기 요청에 의해 수정되는 경우 상기 히트 블록은 상기 캐시 라인을 무효화시키도록 상기 중복제거 엔진으로 제2 쓰기 요청을 전송하도록 작동한다.

[0089] 서술 6. 서술 5에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 제2 쓰기 요청은 상기 중복제거 읽기 캐시의 상기 캐시 라인으로 0 값들을 쓰는 요청을 포함한다.

[0090] 서술 7. 서술 5에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 캐시 컨트롤러는 상기 캐시 메모리의 메타데이터 영역에 상기 캐시 라인이 유효하지 않은 것으로 표시하도록 더 작동한다.

[0091] 서술 8. 서술 5에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 히트 블록은 상기 제2 쓰기 요청에 응답하는 상기 중복제거 엔진으로부터 응답 신호를 수신하지 않고, 상기 중복제거 읽기 캐시로부터 제2 캐시 라인을 제거하고, 상기 중복제거 읽기 캐시로부터 상기 제2 캐시 라인을 제거한 후 상기 중복제거 엔진으로 상기 제2 쓰기 요청을 다시 전송하도록 더 작동한다.

[0092] 서술 9. 서술 1에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서:

[0093] 상기 캐시 컨트롤러는 데이터를 읽기 위한 읽기 요청을 프로세서로부터 수신하고 상기 프로세서로 상기 데이터를 전송하도록 작동하고,

[0094] 상기 미스 블록은 상기 백엔드 고용량 메모리로부터 상기 데이터를 검색하고 상기 중복제거 읽기 캐시로 상기 데이터를 쓰기 위해 상기 중복제거 엔진으로 쓰기 요청을 전송하도록 작동한다.

[0095] 서술 10. 서술 9에 따른 중복제거 캐시를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 미스 블록은 상기 쓰기 요청에 응답하는 상기 중복제거 엔진으로부터 확인 응답 신호를 수신하지 않고, 상기 중복제거 읽기 캐시로부터 캐시 라인을 제거하고, 상기 중복제거 읽기 캐시로부터 상기 캐시 라인을 제거한 후 상기 중복제거 엔진으로 상기 쓰기 요청을 다시 전송하도록 더 작동한다.

[0096] 서술 11. 본 발명의 하나의 실시 예에 따른 방법은:

[0097] 데이터를 쓰기 위한 쓰기 요청을 수신하는 단계;

[0098] 상기 데이터가 중복제거 읽기 캐시의 캐시 라인에 있는 것으로 판별하는 단계;

[0099] 상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 단계; 및

[0100] 상기 데이터를 비-중복제거 쓰기 버퍼에 저장하는 단계를 포함하고,

[0101] 상기 중복제거 읽기 캐시는 캐시 메모리의 제1 영역이고, 상기 캐시 메모리는 제2 영역으로서 상기 비-중복제거 쓰기 버퍼를 포함한다.

[0102] 서술 12. 서술 11에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 방법은:

[0103] 상기 비-중복제거 쓰기 버퍼로부터 백엔드 고용량 메모리로 상기 데이터를 풀려시하는 단계; 및

[0104] 상기 비-중복제거 쓰기 버퍼로부터 상기 데이터를 삭제하는 단계를 더 포함한다.

[0105] 서술 13. 서술 11에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 단계는:

[0106] 상기 캐시 메모리의 메타데이터 영역에 상기 캐시 라인이 유효하지 않은 것으로 표시하는 단계; 및

[0107] 중복제거 엔진을 통해 상기 캐시 라인으로 0 값들을 쓰는 단계를 포함한다.

[0108] 서술 14. 서술 13에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 상

기 캐시 라인을 무효화시키는 단계는 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하는 단계를 더 포함한다.

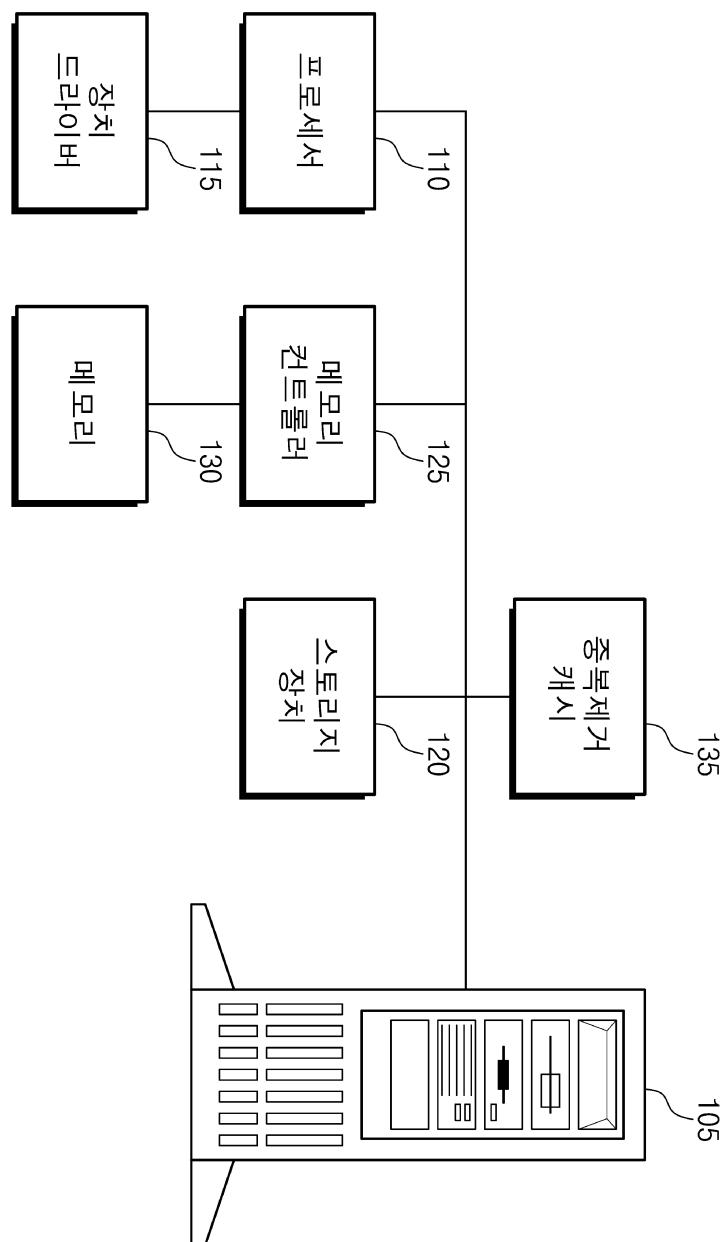
- [0109] 서술 15. 서술 13에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 단계는:
- [0110] 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하지 않는 단계;
- [0111] 상기 중복제거 읽기 캐시로부터 제거할 제2 캐시 라인을 선택하는 단계;
- [0112] 상기 제2 캐시 라인을 무효화시키는 단계; 및
- [0113] 상기 중복제거 엔진을 통해 상기 제2 캐시 라인으로 0 값들을 쓰는 단계를 더 포함한다.
- [0114] 서술 16. 서술 11에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 데이터를 비-중복제거 쓰기 버퍼에 저장하는 단계는 상기 데이터가 상기 중복제거 읽기 캐시의 상기 캐시 라인에 있는지 여부와 상관 없이 상기 데이터를 상기 비-중복제거 쓰기 버퍼에 저장하는 단계를 포함한다.
- [0115] 서술 17. 본 발명의 하나의 실시 예에 따른 방법은:
- [0116] 데이터를 읽기 위한 읽기 요청을 수신하는 단계;
- [0117] 상기 데이터가 중복제거 읽기 캐시의 복수의 캐시 라인들에 있지 않은 것으로 판별하는 단계;
- [0118] 백엔드 고용량 메모리로부터 상기 데이터를 읽는 단계;
- [0119] 상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 단계;
- [0120] 상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 단계; 및
- [0121] 상기 읽기 요청에 응답하여 상기 데이터를 전송하는 단계를 포함하고,
- [0122] 상기 중복제거 읽기 캐시는 캐시 메모리의 제1 영역이고, 상기 캐시 메모리는 제2 영역으로서 비-중복제거 쓰기 버퍼를 포함한다.
- [0123] 서술 18. 서술 17에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 단계는 상기 중복제거 읽기 캐시에서 현재 데이터를 저장하고 있지 않은 상기 제1 캐시 라인을 선택하는 단계를 포함한다.
- [0124] 서술 19. 서술 17에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 단계는:
- [0125] 상기 캐시 메모리의 메타데이터 영역으로부터의 중복제거 읽기 캐시 메타데이터 및 히스토리 데이터에 응답하여 상기 중복제거 읽기 캐시에서 제거할 현재 유효한 데이터를 저장하고 있는 상기 제1 캐시 라인을 선택하는 단계; 및
- [0126] 상기 제1 캐시 라인을 무효화시키는 단계를 포함한다.
- [0127] 서술 20. 서술 19에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 제1 캐시 라인을 무효화시키는 단계는:
- [0128] 상기 캐시 메모리의 상기 메타데이터 영역에 상기 제1 캐시 라인을 유효하지 않은 것으로 표시하는 단계; 및
- [0129] 상기 중복제거 엔진을 통해 상기 제1 캐시 라인으로 0 값들을 쓰는 단계를 포함한다.
- [0130] 서술 21. 서술 17에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 단계는 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하는 단계를 포함한다.
- [0131] 서술 22. 서술 17에 따른 방법을 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 단계는:
- [0132] 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하지 않는 단계;

- [0133] 상기 캐시 메모리의 메타데이터 영역으로부터의 중복제거 읽기 캐시 메타데이터 및 히스토리 데이터에 응답하여 상기 중복제거 읽기 캐시로부터 제거할 현재 유효한 데이터를 저장하고 있는 제2 캐시 라인을 선택하는 단계;
- [0134] 상기 제2 캐시 라인을 무효화시키는 단계; 및
- [0135] 상기 데이터를 상기 선택된 제2 캐시 라인으로 쓰는 것을 시도하기 위해 상기 중복제거 엔진으로 상기 데이터를 제공하는 단계를 포함한다.
- [0136] 서술 23. 본 발명의 하나의 실시 예에 따른 비-일시적 저장 매체는 저장된 명령어들을 포함하고, 명령어들이 머신에 의해 실행되는 경우, 명령어들은:
- 데이터를 쓰기 위한 쓰기 요청을 수신하는 것;
- [0138] 상기 데이터가 중복제거 읽기 캐시의 캐시 라인에 있는 것으로 판별하는 것;
- [0139] 상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 것; 및
- [0140] 상기 데이터를 비-중복제거 쓰기 버퍼에 저장하는 것을 야기하고,
- [0141] 상기 중복제거 읽기 캐시는 캐시 메모리의 제1 영역이고, 상기 캐시 메모리는 제2 영역으로서 상기 비-중복제거 쓰기 버퍼를 포함한다.
- [0142] 서술 24. 서술 23에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 비-일시적 저장 매체는 저장된 더 많은 명령어들을 포함하고, 명령어들이 상기 머신에 의해 실행되는 경우, 명령어들은:
- 상기 비-중복제거 쓰기 버퍼로부터 백엔드 고용량 메모리로 상기 데이터를 플러시하는 것; 및
- [0144] 상기 비-중복제거 쓰기 버퍼로부터 상기 데이터를 삭제하는 것을 야기한다.
- [0145] 서술 25. 서술 23에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 것은:
- 상기 캐시 메모리의 메타데이터 영역에 상기 캐시 라인이 유효하지 않은 것으로서 표시하는 것; 및
- [0147] 중복제거 엔진을 통해 상기 캐시 라인으로 0 값들을 쓰는 것을 포함한다.
- [0148] 서술 25에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 것은 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하는 것을 더 포함한다.
- [0149] 서술 27. 서술 25에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 상기 캐시 라인을 무효화시키는 것은:
- 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하지 않는 것;
- [0151] 상기 중복제거 읽기 캐시로부터 제거할 제2 캐시 라인을 선택하는 것;
- [0152] 상기 제2 캐시 라인을 무효화시키는 것; 및
- [0153] 상기 중복제거 엔진을 통해 상기 제2 캐시 라인으로 0 값들을 쓰는 것을 더 포함한다.
- [0154] 서술 28. 서술 23에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 데이터를 비-중복제거 쓰기 버퍼에 저장하는 것은 상기 데이터가 상기 중복제거 읽기 캐시의 상기 캐시 라인에 있는지 여부와 상관 없이 상기 데이터를 상기 비-중복제거 쓰기 버퍼에 저장하는 것을 포함한다.
- [0155] 서술 29. 본 발명의 하나의 실시 예에 따른 비-일시적 저장 매체는 저장된 명령어들을 포함하고, 명령어들이 머신에 의해 실행되는 경우, 명령어들은:
- 데이터를 읽기 위한 읽기 요청을 수신하는 것;
- [0157] 상기 데이터가 중복제거 읽기 캐시의 복수의 캐시 라인들에 있지 않은 것으로 판별하는 것;
- [0158] 백엔드 고용량 메모리로부터 상기 데이터를 읽는 것;
- [0159] 상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 것;

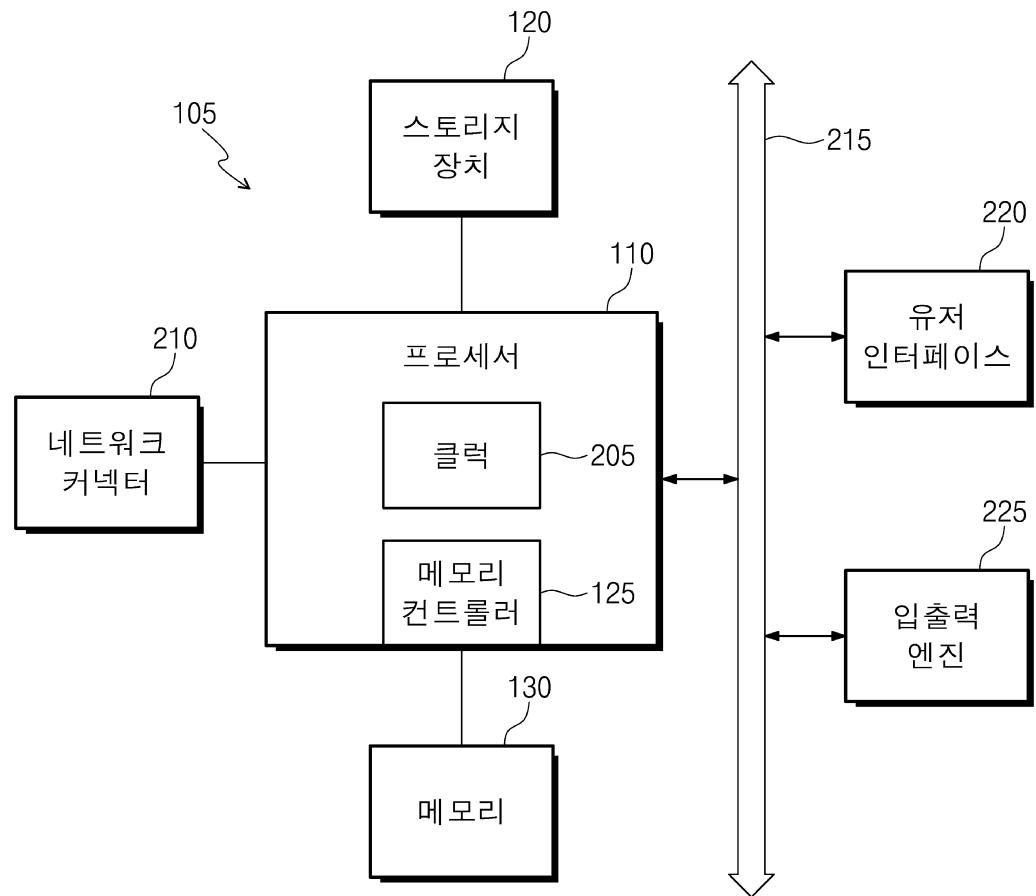
- [0160] 상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 것; 및
- [0161] 상기 읽기 요청에 응답하여 상기 데이터를 전송하는 것을 야기하고,
- [0162] 상기 중복제거 읽기 캐시는 캐시 메모리의 제1 영역이고, 상기 캐시 메모리는 제2 영역으로서 비-중복제거 쓰기 베퍼를 포함한다.
- [0163] 서술 30. 서술 29에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 것은 상기 중복제거 읽기 캐시에서 현재 데이터를 저장하고 있지 않은 상기 제1 캐시 라인을 선택하는 것을 포함한다.
- [0164] 서술 31. 서술 29에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 중복제거 읽기 캐시의 제1 캐시 라인을 선택하는 것은:
- [0165] 상기 캐시 메모리의 메타데이터 영역으로부터의 중복제거 읽기 캐시 메타데이터 및 히스토리 데이터에 응답하여 상기 중복제거 읽기 캐시에서 제거할 현재 유효한 데이터를 저장하고 있는 상기 제1 캐시 라인을 선택하는 것; 및
- [0166] 상기 제1 캐시 라인을 무효화시키는 것을 포함한다.
- [0167] 서술 32. 서술 31에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 제1 캐시 라인을 무효화시키는 것은:
- [0168] 상기 캐시 메모리의 상기 메타데이터 영역에 상기 제1 캐시 라인을 유효하지 않은 것으로 표시하는 것; 및
- [0169] 상기 중복제거 엔진을 통해 상기 제1 캐시 라인으로 0 값들을 쓰는 것을 포함한다.
- [0170] 서술 33. 서술 29에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 것은 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 응답을 수신하는 것을 포함한다.
- [0171] 서술 34. 서술 29에 따른 비-일시적 저장 매체를 포함하는 본 발명의 하나의 실시 예에 있어서, 상기 데이터를 상기 제1 캐시 라인으로 쓰는 것을 시도하기 위해 중복제거 엔진으로 상기 데이터를 제공하는 것은:
- [0172] 상기 중복제거 엔진으로부터 쓰기 상태 신호에 대한 확인 응답을 수신하지 않는 것;
- [0173] 상기 캐시 메모리의 메타데이터 영역으로부터의 중복제거 읽기 캐시 메타데이터 및 히스토리 데이터에 응답하여 상기 중복제거 읽기 캐시로부터 제거할 현재 유효한 데이터를 저장하고 있는 제2 캐시 라인을 선택하는 것;
- [0174] 상기 제2 캐시 라인을 무효화시키는 것; 및
- [0175] 상기 데이터를 상기 선택된 제2 캐시 라인으로 쓰는 것을 시도하기 위해 상기 중복제거 엔진으로 상기 데이터를 제공하는 것을 포함한다.
- [0176] 따라서, 본 명세서에서 설명된 실시 예들에 대한 다양한 치환의 관점에서, 상세한 설명 및 수반되는 자료는 단지 설명만을 위한 것이고, 본 발명의 권리 범위를 제한하지 않아야 한다. 따라서, 본 발명에서 청구되는 것은 아래의 청구항들의 범위 및 동등한 범위 내에 있을 수 있는 모든 변형을 포함한다.

도면

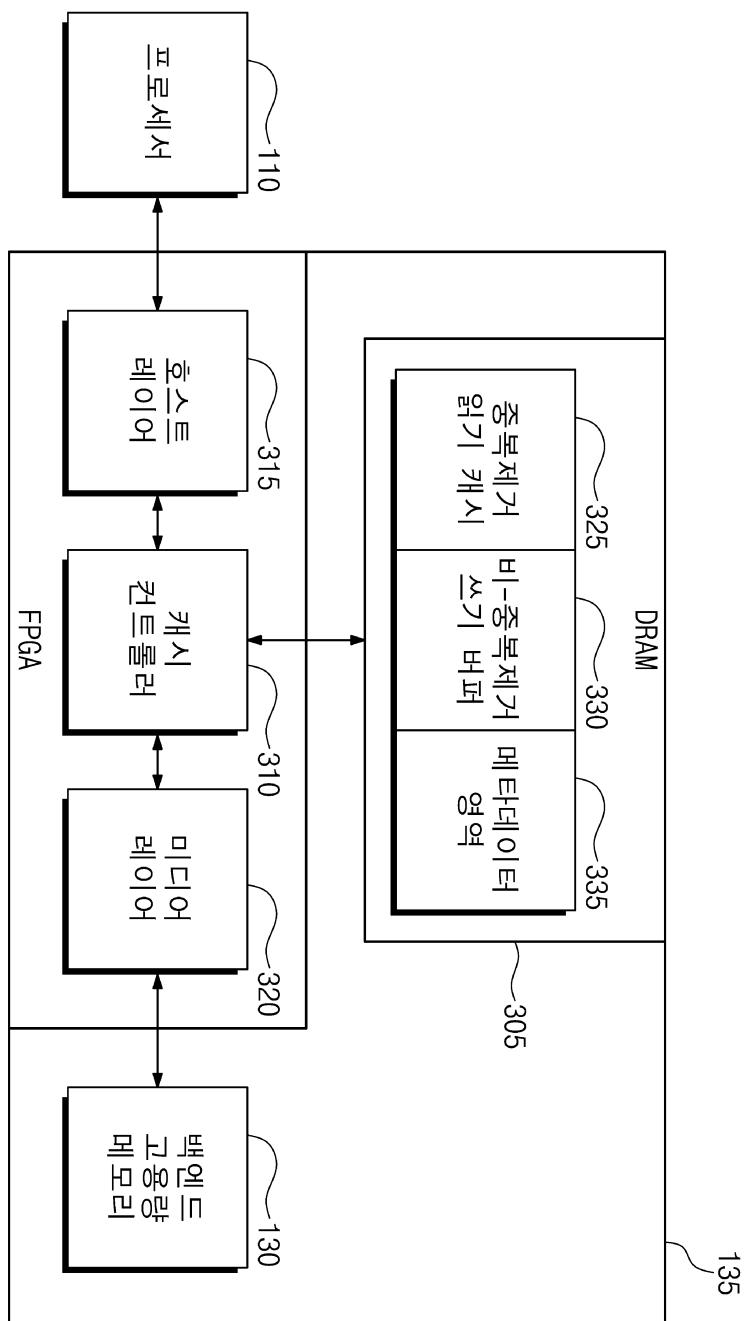
도면1



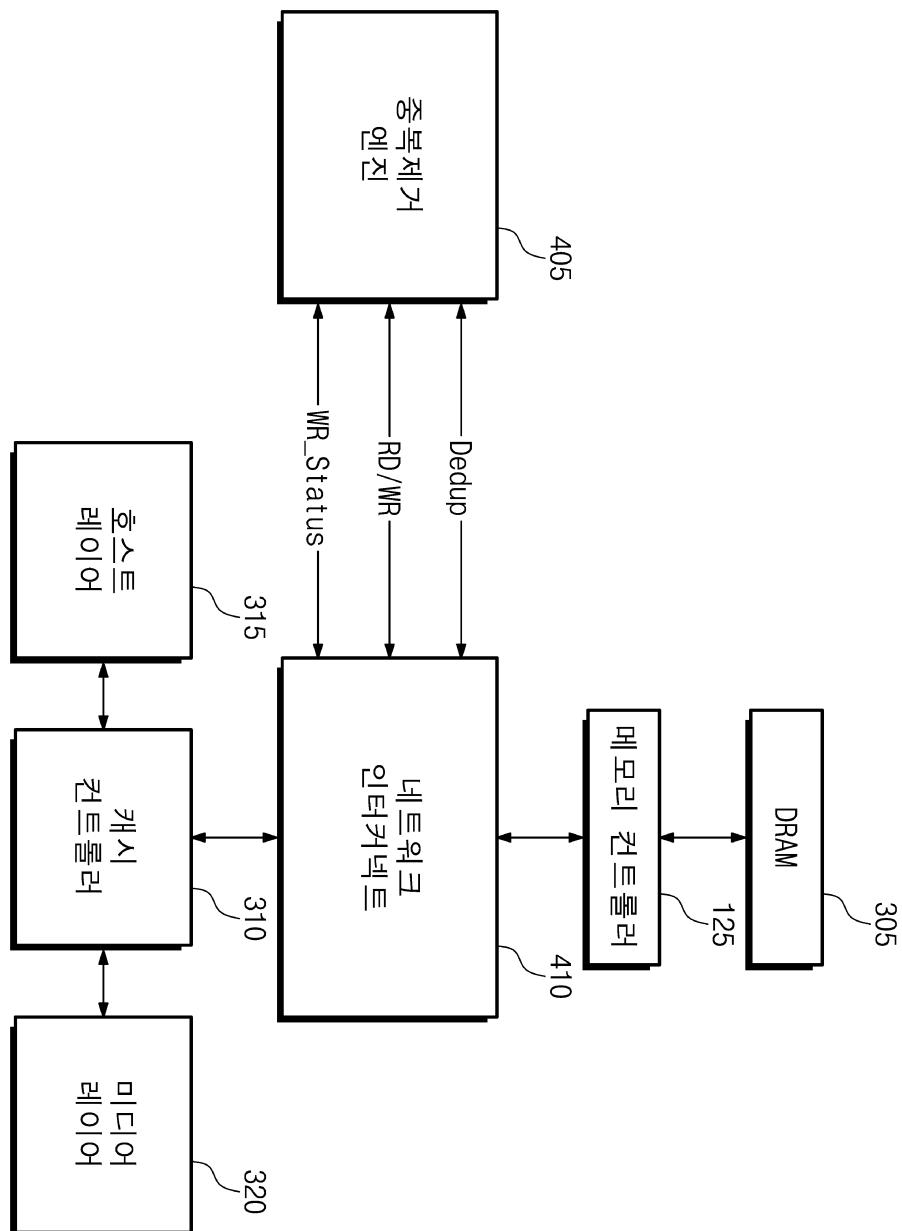
도면2



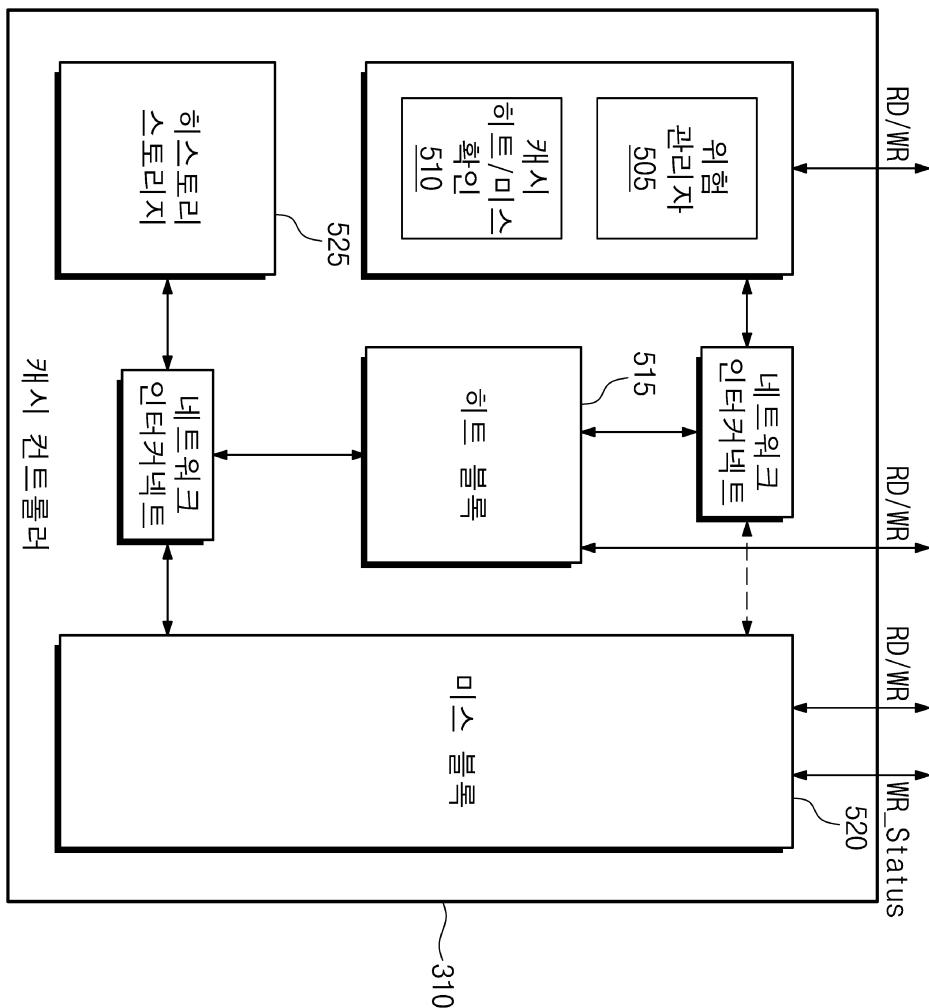
도면3



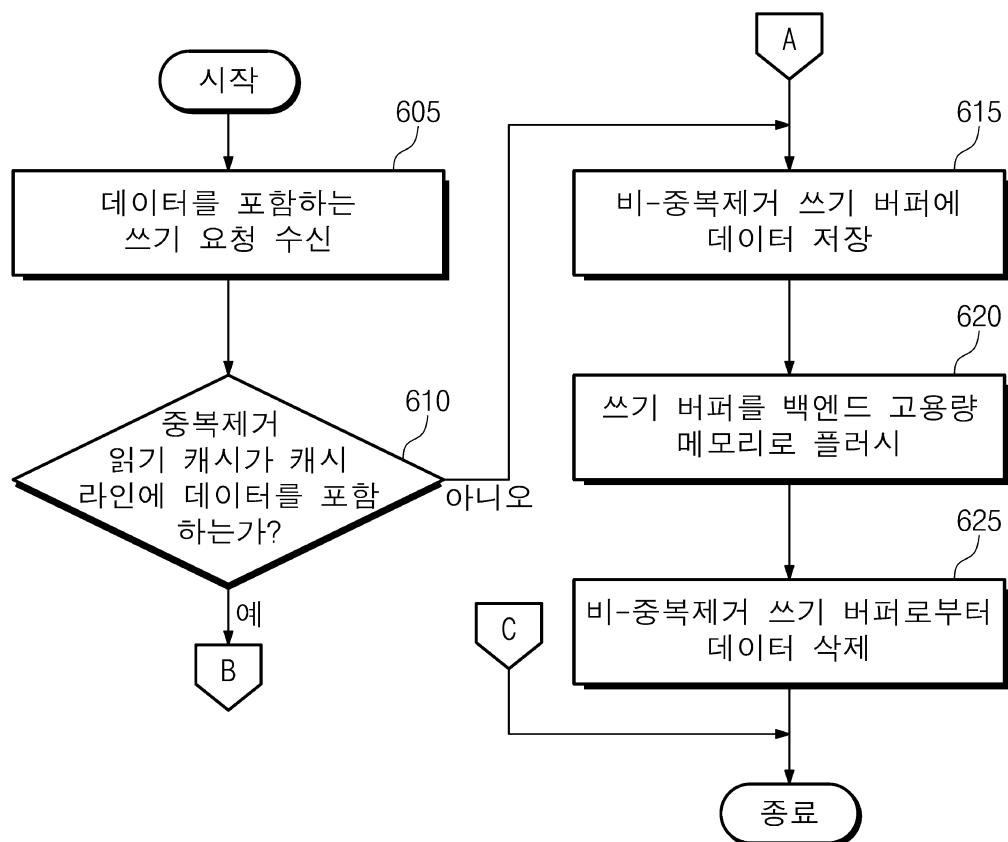
도면4



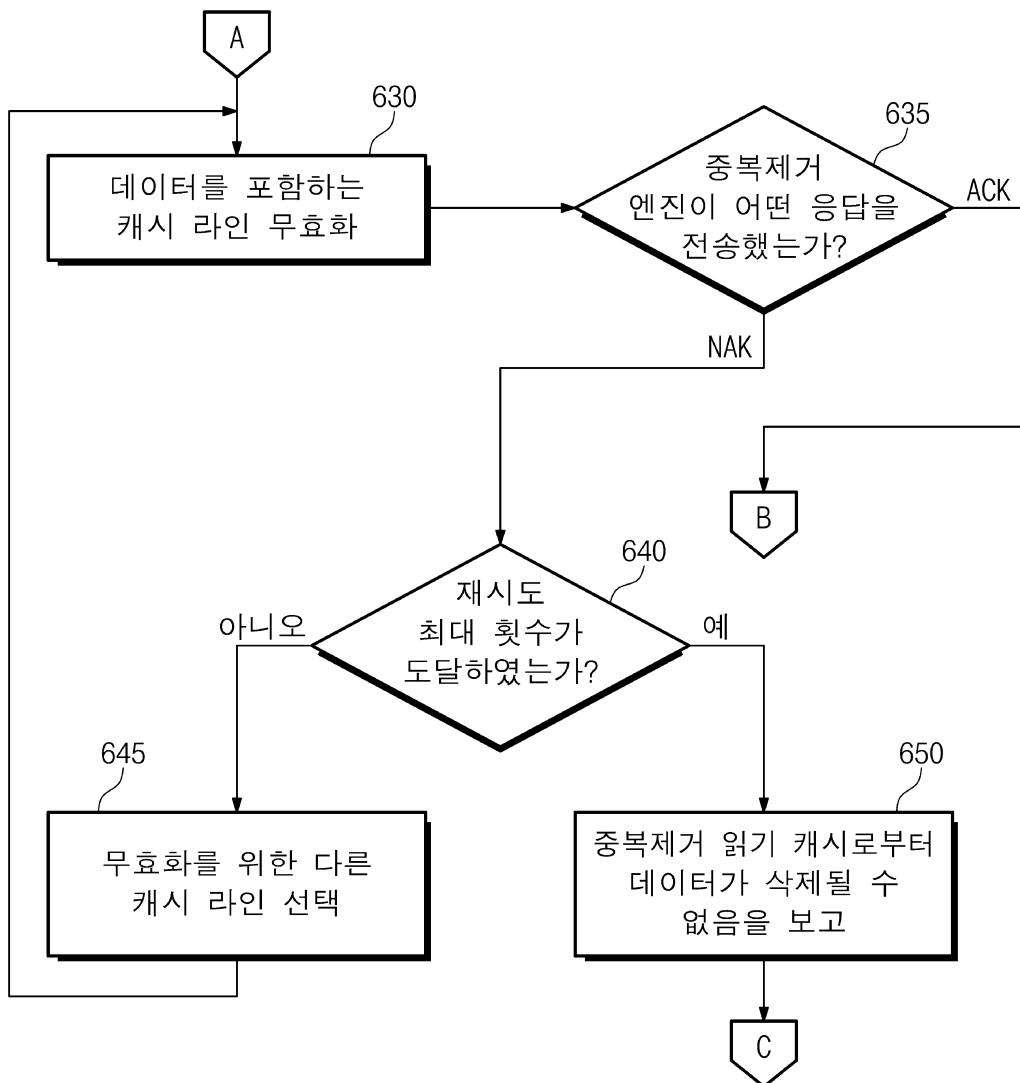
도면5



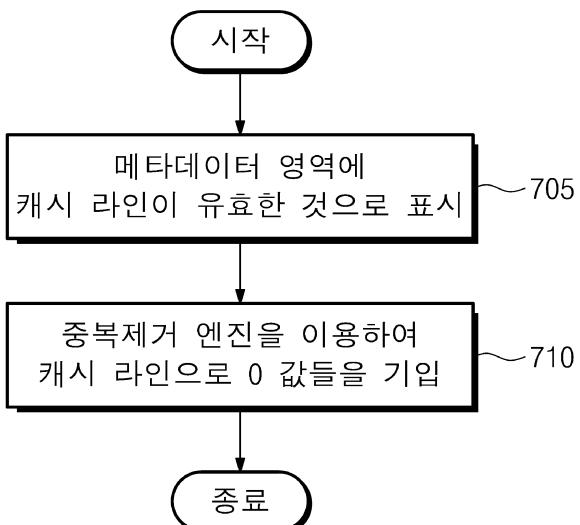
도면6a



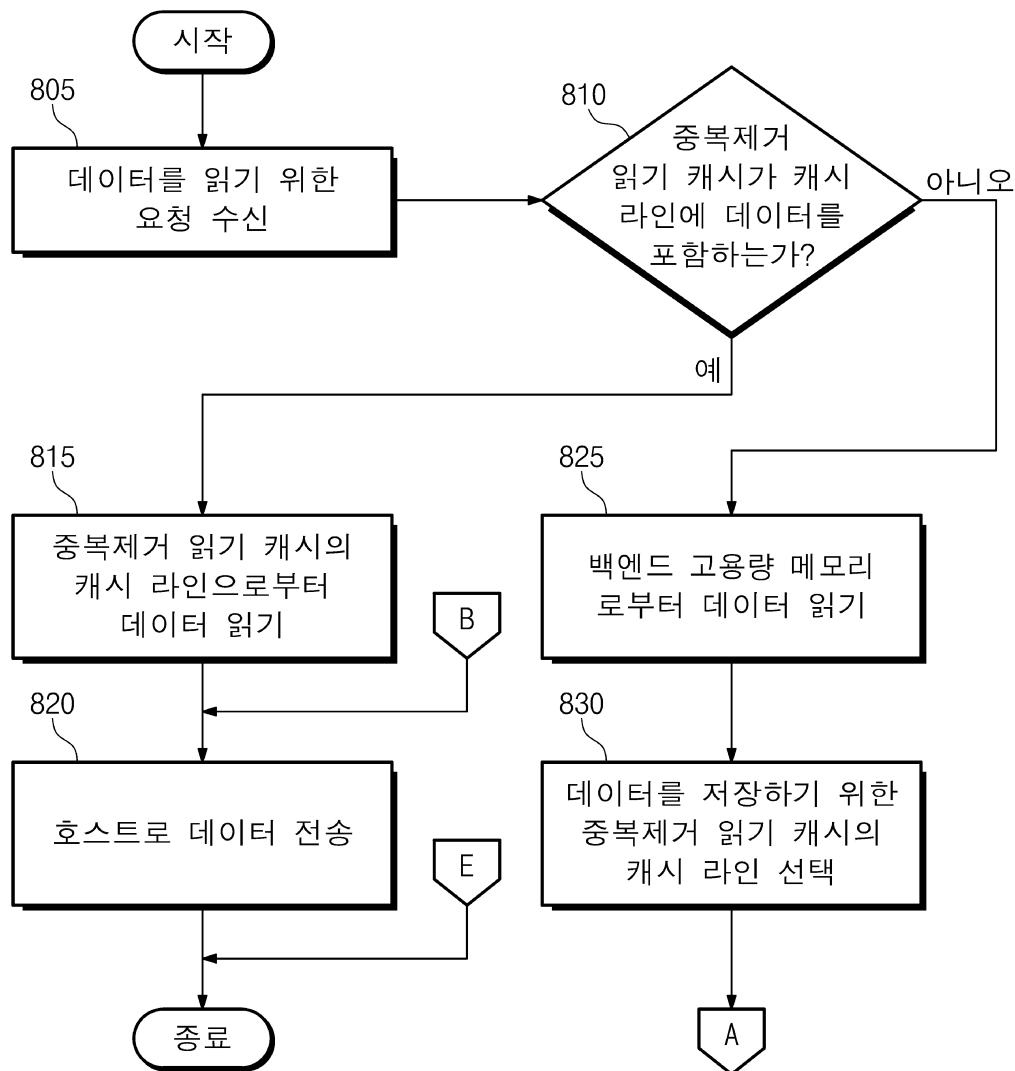
도면6b



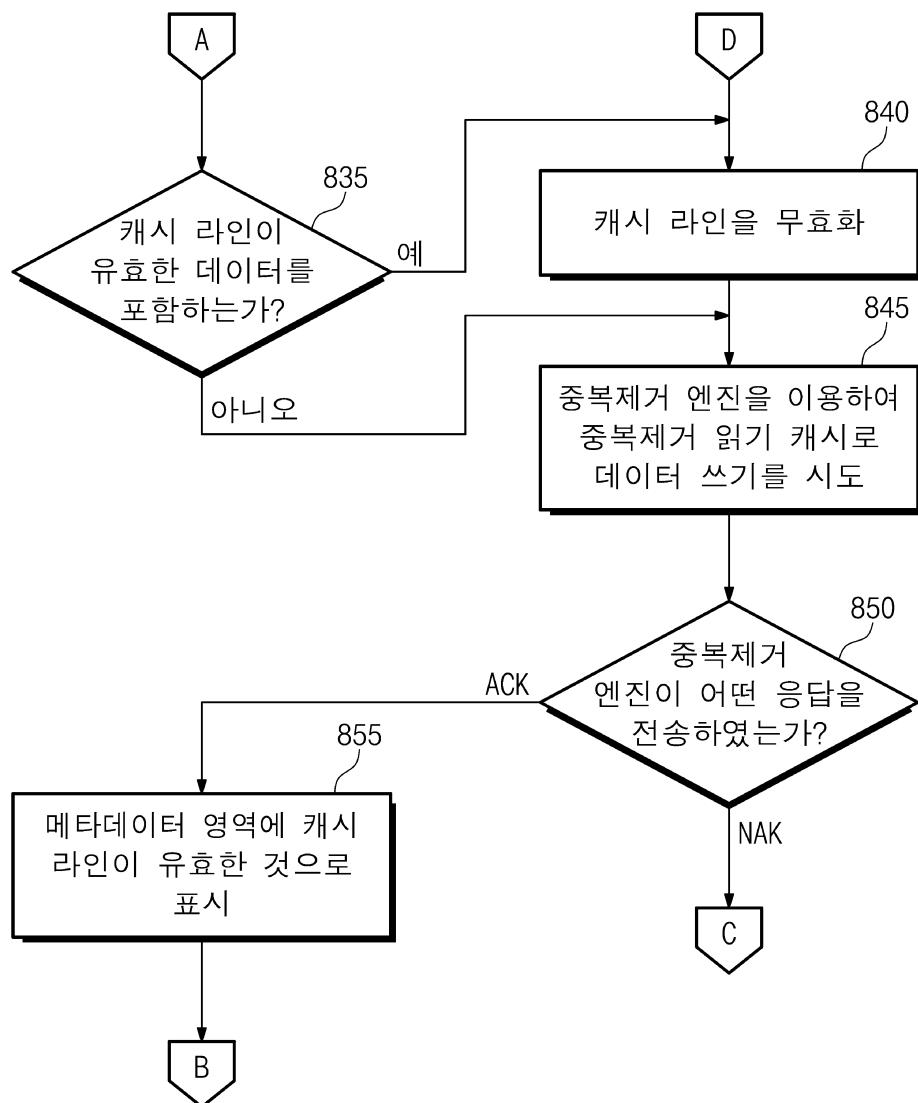
도면7



도면8a



도면8b



도면8c

