



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2023년11월03일  
(11) 등록번호 10-2597640  
(24) 등록일자 2023년10월30일

(51) 국제특허분류(Int. Cl.)  
G06F 3/06 (2006.01) G06F 16/00 (2019.01)  
(52) CPC특허분류  
G06F 3/0641 (2013.01)  
G06F 16/1748 (2019.01)  
(21) 출원번호 10-2016-0149705  
(22) 출원일자 2016년11월10일  
심사청구일자 2021년11월09일  
(65) 공개번호 10-2017-0112952  
(43) 공개일자 2017년10월12일  
(30) 우선권주장  
62/314,918 2016년03월29일 미국(US)  
15/161,136 2016년05월20일 미국(US)  
(56) 선행기술조사문헌  
KR1020150035876 A  
KR1020170131274 A

(73) 특허권자  
삼성전자주식회사  
경기도 수원시 영통구 삼성로 129 (매탄동)  
(72) 발명자  
살라, 프레데릭  
미국 캘리포니아주 90046 로스앤젤레스 커슨 애비뉴 1300 노스 아파트 4  
후, 차오홍  
미국 캘리포니아주 95148 산호세 체민 데 리비에르 3461  
(74) 대리인  
특허법인 고려  
(뒷면에 계속)

전체 청구항 수 : 총 20 항

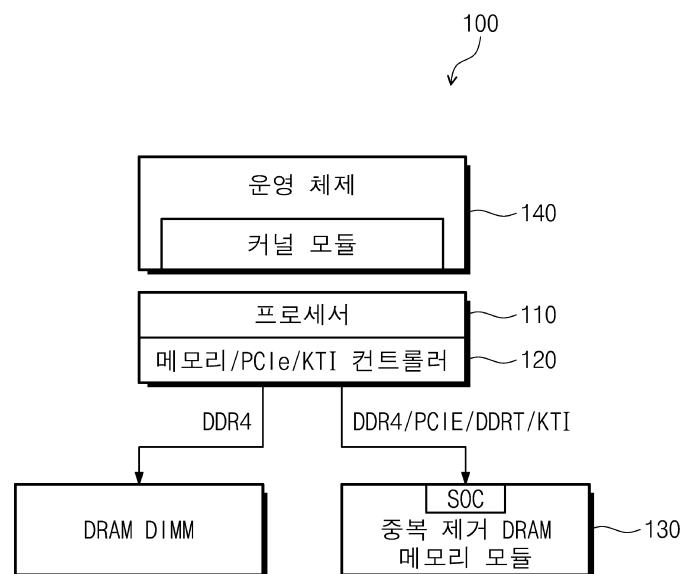
심사관 : 신현상

(54) 발명의 명칭 중복제거 어플리케이션을 즉시 처리하는 효율적인 메모리를 위한 최적화된 합스카치 복수의 해시 테이블들

(57) 요약

메모리 중복제거 방법은, 복수의 해시 테이블들 각각은 해시 함수에 대응하고, 복수의 해시 테이블들 각각은 물리적인 해시 버킷들로 구성되고, 각 물리적인 버킷은 웨이들로 구성되고 데이터를 저장하는, 복수의 해시 테이블들을 식별하는 단계, 복수의 가상 버킷(Virtual bucket)들 각각은 상기 물리적인 해시 버킷들 중 일부로 구성되  
(뒷면에 계속)

대표도 - 도1



고 상기 복수의 가상 버킷들 중 다른 가상 버킷과 적어도 하나의 상기 물리적인 해시 버킷들을 공유하는, 복수의 가상 버킷들을 식별하는 단계, 상기 가상 버킷들 중 해당 가상 버킷에 할당되어 저장된 데이터를 갖는 상기 복수의 물리적인 버킷들 각각을 식별하는 단계, 해시 값(Hash value)을 생성하기 위해 해시 함수들 중 해당 해시 함수에 따라 데이터 라인(Data line)을 해싱(Hashing)하는 단계, 해당 해시 테이블의 상기 가상 버킷들 중 해당 가상 버킷이 상기 해시 값에 따라 데이터 블록을 위한 사용 가능한 공간을 갖는지를 결정하는 단계, 상기 가상 버킷들 중 해당 가상 버킷이 상기 데이터 블록을 위한 공간을 가질 때까지 상기 가상 버킷들 중 해당 가상 버킷이 사용 가능한 공간을 갖지 않은 때 상기 가상 버킷들 중 해당 가상 버킷으로부터 상기 가상 버킷들 중 인접한 가상 버킷으로 순차적으로 데이터를 움직이는 단계, 및 상기 해당 가상 버킷에 상기 데이터 블록을 저장하는 단계를 포함할 수 있다.

(52) CPC특허분류

*G06F 16/2255* (2019.01)

*G06F 3/0625* (2013.01)

*G06F 3/0649* (2013.01)

*G06F 3/0656* (2013.01)

*G06F 3/0671* (2013.01)

(72) 발명자

**정, 홍중**

미국 캘리포니아주 94087 서니배일 파킹톤 애비뉴  
1295

**창, 무-티엔**

미국 캘리포니아주 95051 산타클라라 토리노 플레  
이스 비아 2920

## 명세서

### 청구범위

#### 청구항 1

복수의 해시 테이블(Hash table)들 각각은 해시 함수에 대응하고, 상기 복수의 해시 테이블들 각각은 물리적인 해시 버킷(Hash bucket)들로 구성되고, 각 물리적인 버킷은 웨이들(Ways)로 구성되고 데이터를 저장하는, 상기 복수의 해시 테이블들을 식별하는 단계;

복수의 가상 버킷(Virtual bucket)들 각각은 상기 물리적인 해시 버킷들 중 일부로 구성되고 상기 복수의 가상 버킷들 중 다른 가상 버킷과 적어도 하나의 상기 물리적인 해시 버킷들을 공유하는, 상기 복수의 가상 버킷들을 식별하는 단계;

상기 가상 버킷들 중 대응하는 하나의 가상 버킷에 할당되어 저장된 데이터를 갖는 상기 복수의 물리적인 버킷들 각각을 식별하는 단계;

해시 값(Hash value)을 생성하기 위해 해시 함수들 중 해당 해시 함수에 따라 데이터 라인(Data line)을 해싱(Hashing)하는 단계;

해당 해시 테이블의 상기 가상 버킷들 중 해당 가상 버킷이 상기 해시 값에 따라 데이터 블록을 위한 사용 가능한 공간을 갖는지 여부를 결정하는 단계;

상기 가상 버킷들 중 상기 해당 가상 버킷이 상기 데이터 블록을 위한 공간을 가질 때까지 상기 가상 버킷들 중 상기 해당 가상 버킷이 사용 가능한 공간을 갖지 않은 때 상기 가상 버킷들 중 상기 해당 가상 버킷으로부터 상기 가상 버킷들 중 인접한 가상 버킷으로 순차적으로 데이터를 움직이는 단계; 및

상기 가상 버킷들 중 상기 해당 가상 버킷에 상기 데이터 블록을 저장하는 단계를 포함하는 메모리의 중복제거 동작 방법.

#### 청구항 2

제 1 항에 있어서,

움직여진 데이터 블록에 해당하는 하나 이상의 검색 어드레스들(Lookup addresses)을 변경하기 위해 어드레스 검색 테이블 메모리를 업데이트하는 단계를 더 포함하는 메모리의 중복제거 동작 방법.

#### 청구항 3

제 1 항에 있어서,

상기 복수의 해시 테이블들 각각은 참조 카운트 라인(Reference count line), 서명 라인(Signature line), 그리고 홉워드 라인(Hopword line)을 더 포함하는 메모리의 중복제거 동작 방법.

#### 청구항 4

제 3 항에 있어서,

상기 가상 버킷에 대응하는 데이터를 포함하는 상기 물리적인 해시 버킷들을 나타내기 위한 홉워드 벡터(Hopword vector)를 생성하는 단계를 더 포함하는 메모리의 중복제거 동작 방법.

#### 청구항 5

제 4 항에 있어서,

상기 홉워드 벡터를 생성하는 단계는:

상기 복수의 가상 버킷들 각각을 위해, 상기 복수의 가상 버킷들 중 상대적인 가상 버킷의 상기 복수의 물리적인 해시 버킷들 각각이 상기 복수의 가상 버킷들 중 상기 상대적인 가상 버킷과 연동되는 데이터 블록을 포함하는지를 나타내기 위해 이진 표시자(Binary indicator)를 사용하는 단계를 포함하는 메모리의 중복제거 동작 방법

법.

**청구항 6**

제 3 항에 있어서,

상기 복수의 가상 버킷들 중 어떤 가상 버킷에 대응하는 데이터를 포함하는 상기 복수의 물리적인 해시 버킷들 중 어떤 물리적인 해시 버킷을 나타내기 위해 물리적인 해시 버킷들마다  $\log_2(H)$  비트들로 구성되는 합워드 값을 생성하는 단계를 더 포함하는 메모리의 중복제거 동작 방법.

**청구항 7**

제 6 항에 있어서,

상기 합워드 값을 생성하는 단계는 상기 복수의 물리적인 해시 버킷들 그리고 가상 버킷들의 쌍들과 연관되어 표현되는 위치에 데이터를 포함하는 상기 복수의 물리적인 해시 버킷들을 각각을 위한 유사 주소들로 구성되는 2차원 어레이를 생성하는 단계를 포함하는 메모리의 중복제거 동작 방법.

**청구항 8**

제 1 항에 있어서,

상기 복수의 해시 테이블은 휘발성 메모리에 저장되는 메모리의 중복제거 동작 방법.

**청구항 9**

제 8 항에 있어서,

상기 휘발성 메모리는 DRAM(Dynamic random access memory)으로 구성되는 메모리의 중복제거 동작 방법.

**청구항 10**

제 1 항에 있어서,

상기 해시 테이블을  $\log_2(H)$  비트들의 가상 버킷 이용 값 필드로 구성되는 물리적인 라인 ID(PLID)로 색인(Indexing)하는 단계를 더 포함하되, 상기 값은 상기 복수의 가상 버킷들 중 해당 가상 버킷의 데이터 블록들의 숫자와 동일한 메모리의 중복제거 동작 방법.

**청구항 11**

제 10 항에 있어서,

상기 복수의 가상 버킷들 중 상기 해당 가상 버킷에 항목을 쓰는 경우, 상기 가상 버킷 이용 값 필드를 1 증가시키는 단계를 더 포함하는 메모리의 중복제거 동작 방법.

**청구항 12**

제 1 항에 있어서,

상기 해당 해시 테이블이 가득 찬 경우, 버퍼 메모리에 상기 데이터 블록을 저장하는 단계를 더 포함하는 메모리의 중복제거 동작 방법.

**청구항 13**

중복제거 메모리를 위해 메모리에서 데이터 블록들의 중복을 감소시키는 중복제거 DRAM(Dynamic random access memory) 메모리 모듈에 있어서,

중복제거된 데이터 블록들을 저장하기 위한 해시 테이블 메모리;

상기 중복제거된 데이터 블록들에 대응하는 어드레스들을 저장하기 위한 어드레스 검색 테이블 메모리; 및

상기 중복제거 DRAM 메모리 모듈이 상기 해시 테이블 메모리로부터 상기 데이터 블록들을 검색하고 상기 상기 데이터 블록들을 전송하도록 하는 읽기 요청들을 수신하고, 그리고 상기 중복제거 DRAM 메모리 모듈이 상기 해시 테이블 메모리의 상기 데이터 블록들을 저장하도록 하는 쓰기 요청들을 수신하는 프로세서를 포함하는 중복

제거 DRAM 메모리 모듈.

**청구항 14**

제 13 항에 있어서,

상기 해시 테이블 메모리는, 상기 해시 테이블 메모리에 저장되는 해시 테이블들의 3차원 어레이로 구성되고, 상기 해시 테이블들 각각은 물리적인 해시 버킷들로 구성되고, 각 물리적인 해시 버킷은 웨이들(Ways)로 구성되고 상기 중복제거된 데이터 블록들을 저장하는 중복제거 DRAM 메모리 모듈.

**청구항 15**

제 14 항에 있어서,

상기 해시 테이블들 각각은 복수의 가상 버킷들을 더 포함하되, 각 가상 버킷들은 두 개 이상의 상기 물리적인 해시 버킷들로 구성되는 중복제거 DRAM 메모리 모듈.

**청구항 16**

제 15 항에 있어서,

상기 해시 테이블들 중 해당 해시 테이블 내에서 상기 복수의 가상 버킷들 중 인접한 가상 버킷들 사이에서 상기 중복제거된 데이터 블록들을 움직이는 중복제거 DRAM 메모리 모듈.

**청구항 17**

제 13 항에 있어서,

외부에서 제공되는 명령들 없이 데이터 중복제거를 수행하는 중복제거 DRAM 메모리 모듈.

**청구항 18**

제 13 항에 있어서,

상기 해시 테이블 메모리가 가득 찬 경우, 데이터를 저장하기 위한 버퍼 메모리를 더 포함하는 중복제거 DRAM 메모리 모듈.

**청구항 19**

메모리에서 데이터 블록들의 중복을 감소시키기 위한 중복제거 DRAM(Dynamic random access memory) 메모리 모듈에 있어서,

복수의 해시 테이블들 각각은 물리적인 해시 버킷들로 구성되고, 각 물리적인 해시 버킷은 웨이들(Ways)로 구성되고 상기 데이터 블록들을 저장하고, 상기 중복제거 DRAM 메모리 모듈에 저장되는 해시 테이블들의 3차원 어레이;

프로세서; 및

메모리를 포함하되, 상기 메모리는 명령들을 저장하고, 상기 명령들이 상기 프로세서에 의해 실행될 때, 상기 메모리는 상기 중복제거 DRAM 메모리 모듈이 상기 복수의 해시 테이블들 중 해시 테이블의 인접 가상 버킷들 사이에서 사전에 저장된 중복제거된 데이터 블록을 움직이도록 야기하고, 상기 가상 버킷들 각각은 두 개 이상의 물리적인 해시 버킷들로 구성되는 중복제거 DRAM 메모리 모듈.

**청구항 20**

제 19 항에 있어서,

상기 메모리는 명령들을 더 저장하고, 상기 명령들이 상기 프로세서에 의해 실행될 때, 상기 메모리는 중복제거 DRAM 메모리 모듈이 상기 사전에 저장된 중복제거된 데이터 블록이 움직여진 것으로부터 가상 버킷들 중 하나의 가상 버킷에 인커밍 데이터를 저장하도록 야기하는 중복제거 DRAM 메모리 모듈.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 합스카치 복수의 해시 테이블들에 관한 것으로, 좀 더 자세하게는 중복제거 어플리케이션을 즉시 처리하는 효율적인 메모리를 위한 최적화된 합스카치 복수의 해시 테이블들에 관한 것이다.

**배경 기술**

[0002] 데이터 중복제거는 메모리 장치의 용량 비용을 줄이기 위해, 메모리 장치에서 불필요한 데이터를 줄이는 것을 의미한다. 데이터 중복제거에서, 데이터 대상/항목(예를 들면, 데이터 파일)은 하나 또는 그 이상의 데이터 라인들/덩어리들/블락들로 나뉜다. 동일한 데이터로 구성된 복수의 데이터 블락들을 하나의 저장된 데이터 블락으로 연관시킴으로써, 데이터 블락들의 중복 복사본들은 컴퓨터 메모리에 의해 감소되거나 제거될 수 있고, 그렇게 함으로써 메모리 장치의 데이터 중복 블락들의 전체 양은 감소된다. 데이터의 불필요한 복사본들의 감소는 읽기 속도 및 메모리 대역폭을 증가시킬 수 있고, 잠재적으로 전력 절감을 야기할 수 있다.

[0003] 따라서, 만약 중복된 데이터 복사본들이 하나의 데이터 복사본으로 감소될 수 있으면, 물리적인 자원의 양은 동일하게 사용하면서 메모리 장치의 전체 사용 가능한 용량은 증가된다. 그 결과로 메모리 장치의 경제적 사용은 데이터 다시 쓰기(Rewrite) 횟수의 감소를 가능하게 하고, 메모리에 이미 저장된 중복된 데이터 블락들에 대한 쓰기 요청은 버려질 수 있으므로, 데이터 중복제거가 적용된 메모리 장치의 수명은 효과적으로 쓰기 내구성을 증가시킴으로써 오랫동안 늘어날 수 있다.

[0004] 일반적인 데이터 중복제거 방법론은 CPU 중심의 접근으로 중복제거 엔진이 CPU 또는 메모리 컨트롤러(Memory Controller; MC)에 집적되는 메모리 내 중복제거(in-memory deduplication) 기술을 사용할 수 있다. 이러한 방법론은 중복을 CPU 프로세서가 인식할 수 있도록 그리고 메모리 컨트롤러의 제어에 따라 중복제거 메모리 동작들(예를 들면, 콘텐츠 검색(Content lookups), 참조 카운트 업데이트(Reference Count Updates), 등)의 제공을 시도하도록 메모리 컨트롤러와 동작하는 중복제거 캐시(Deduplicated Cache; DDC)를 구현한다. 중복제거 방법론은 또한, 중요한 경로로부터 변환 패치(Fetch)를 제거하여 데이터 읽기를 증가시키기 위해 변환 라인들을 캐싱(Caching)하는 캐시이고, 색인 버퍼(Lookaside Buffer)와 유사할 수 있는, 직접 변환 버퍼(Direct Translation Buffer; DTB)를 구현할 수 있다.

[0005] 중복제거는 보통 하드 드라이브를 위해 사용된다. 그러나, DRAM(Dynamic Random Access Memory)과 같은 휘발성 메모리의 영역에서 미세한 중복제거를 제공하는데 관심이 있다.

[0006] 이러한 배경 기술에서 전술한 정보는 단지 발명의 이해를 돕기 위한 것이고, 그러므로 종래 기술을 구성하지 않는 정보를 포함할 수 있다.

**발명의 내용**

**해결하려는 과제**

[0007] 본 발명은 상술한 기술적 과제를 해결하기 위한 것으로, 본 발명은 중복제거 어플리케이션을 즉시 처리하는 효율적인 메모리를 위한 최적화된 합스카치 복수의 해시 테이블들을 제공할 수 있다.

**과제의 해결 수단**

[0008] 본 발명의 실시 예들의 측면들은 DRAM(Dynamic random access memory) 시스템에서 메모리 중복제거에 관한 것이다

[0009] 본 발명의 실시 예에 따라 메모리 중복제거 방법이 제공되고, 상기 방법은 복수의 해시 테이블들 각각은 해시 함수에 대응하고, 복수의 해시 테이블들 각각은 물리적인 해시 버킷들로 구성되고, 각 물리적인 버킷은 웨이들로 구성되고 데이터를 저장하는, 복수의 해시 테이블들을 식별하는 단계, 복수의 가상 버킷들 각각은 상기 물리적인 해시 버킷들 중 일부로 구성되고 상기 복수의 가상 버킷들 중 다른 가상 버킷과 적어도 하나의 상기 물리적인 해시 버킷들을 공유하는, 복수의 가상 버킷들을 식별하는 단계, 상기 가상 버킷들 중 해당 가상 버킷에 할당되어 저장된 데이터를 갖는 상기 복수의 물리적인 버킷들 각각을 식별하는 단계, 해시 값을 생성하기 위해 해시 함수들 중 해당 해시 함수에 따라 데이터 라인을 해싱하는 단계, 해당 해시 테이블의 상기 가상 버킷들 중 해당 가상 버킷이 상기 해시 값에 따라 데이터 블락을 위한 사용 가능한 공간을 갖는지를 결정하는 단계, 상기 가상 버킷들 중 해당 가상 버킷이 상기 데이터 블락을 위한 공간을 가질 때까지 상기 가상 버킷들 중 해당 가상

버킷이 사용 가능한 공간을 갖지 않은 때 상기 가상 버킷들 중 해당 가상 버킷으로부터 상기 가상 버킷들 중 인접한 가상 버킷으로 순차적으로 데이터를 움직이는 단계, 및 상기 가상 버킷들 중 해당 가상 버킷에 상기 데이터 블록을 저장하는 단계를 포함할 수 있다.

- [0010] 방법은 상기 움직여진 데이터 블록에 해당하는 하나 이상의 검색 어드레스들(Lookup addresses)을 변경하기 위해 어드레스 검색 테이블 메모리를 업데이트하는 단계를 더 포함할 수 있다.
- [0011] 상기 복수의 해시 테이블들 각각은 참조 카운트 라인, 서명 라인, 그리고 합워드 라인을 더 포함할 수 있다.
- [0012] 방법은 상기 가상 버킷에 대응하는 데이터를 포함하는 상기 물리적인 해시 버킷들을 나타내기 위한 합워드 벡터를 생성하는 단계를 더 포함할 수 있다.
- [0013] 상기 합워드 벡터를 생성하는 단계는, 상기 복수의 가상 버킷들 각각을 위해, 상기 복수의 가상 버킷들 중 상대적인 가상 버킷의 상기 복수의 물리적인 해시 버킷들 각각이 상기 복수의 가상 버킷들 중 상기 상대적인 가상 버킷과 연동되는 데이터 블록을 포함하는지를 나타내기 위해 이진 표시자를 사용하는 단계를 포함할 수 있다.
- [0014] 방법은 상기 복수의 가상 버킷들 중 어떤 가상 버킷에 대응하는 데이터를 포함하는 상기 복수의 물리적인 해시 버킷들 중 어떤 물리적인 해시 버킷을 나타내기 위해 물리적인 해시 버킷들마다  $\log_2(H)$  비트들로 구성되는 합워드 값을 생성하는 단계를 더 포함할 수 있다.
- [0015] 상기 합워드 값을 생성하는 단계는 상기 복수의 물리적인 해시 버킷들 그리고 가상 버킷들의 쌍들과 연관되어 표현되는 위치에 데이터를 포함하는 상기 복수의 물리적인 해시 버킷들을 각각을 위한 유사 주소들로 구성되는 2차원 어레이를 생성하는 단계를 포함할 수 있다.
- [0016] 상기 복수의 해시 테이블은 휘발성 메모리에 저장될 수 있다.
- [0017] 상기 휘발성 메모리는 DRAM(Dynamic random access memory)를 포함할 수 있다.
- [0018] 방법은 상기 해시 테이블을  $\log_2(H)$  비트들의 가상 버킷 이용 값 필드로 구성되는 물리적인 라인 ID(PLID)로 색인하는 단계를 더 포함할 수 있고, 상기 값은 상기 복수의 가상 버킷들 중 해당 가상 버킷의 데이터 블록들의 숫자와 동일할 수 있다.
- [0019] 방법은 상기 복수의 가상 버킷들 중 상기 해당 가상 버킷에 항목을 쓰는 경우, 상기 가상 버킷 이용 값 필드를 1 증가시키는 단계를 더 포함할 수 있다.
- [0020] 방법은 상기 해당 해시 테이블이 가득 찬 경우, 버퍼 메모리에 상기 데이터 블록을 저장하는 단계를 더 포함할 수 있다.
- [0021] 본 발명의 실시 예에 따라 중복제거 메모리를 위해 메모리에서 데이터 블록들의 중복을 감소시키는 중복제거 DRAM 메모리 모듈이 제공되고, 중복제거 DRAM 메모리 모듈은 중복제거된 데이터 블록들을 저장하기 위한 해시 테이블 메모리, 상기 중복제거된 데이터 블록들과 대응하는 어드레스들을 저장하기 위한 어드레스 검색 테이블 메모리, 및 상기 중복제거 DRAM 메모리 모듈이 상기 해시 테이블 메모리로부터 데이터 블록들을 검색하도록 하는 읽기 요청들을 수신하고, 그리고 상기 중복제거 DRAM 메모리 모듈이 상기 해시 테이블 메모리의 상기 데이터 블록들을 저장하도록 하는 쓰기 요청들을 수신하는 프로세서를 포함할 수 있다.
- [0022] 상기 해시 테이블 메모리는, 상기 해시 테이블 메모리에 저장되는 해시 테이블들의 3차원 어레이로 구성될 수 있고, 상기 해시 테이블들 각각은 물리적인 해시 버킷들로 구성되고, 각 물리적인 해시 버킷은 웨이들로 구성되고 상기 중복제거된 데이터 블록들을 저장할 수 있다.
- [0023] 상기 해시 테이블들 각각은 복수의 가상 버킷들을 더 포함하되, 각 가상 버킷들은 두 개 이상의 상기 물리적인 해시 버킷들로 구성될 수 있다.
- [0024] 중복제거 DRAM 메모리 모듈은 상기 해시 테이블들 중 해당 해시 테이블 내에서 상기 복수의 가상 버킷들 중 인접한 가상 버킷들 사이에서 상기 중복제거된 데이터 블록들을 움직일 수 있다.
- [0025] 중복제거 DRAM 메모리 모듈은 외부에서 제공되는 명령들 없이 데이터 중복제거를 수행할 수 있다.
- [0026] 중복제거 DRAM 메모리 모듈은 상기 해시 테이블 메모리가 가득 찬 경우, 데이터를 저장하기 위한 버퍼 메모리를 더 포함할 수 있다.
- [0027] 본 발명의 실시 예에 따라 중복제거 메모리를 위해 메모리에서 데이터 블록들의 중복을 감소시키기 위한 중복제

거 DRAM 메모리 모듈이 제공되고, 중복제거 DRAM 메모리 모듈은, 복수의 해시 테이블들 각각은 물리적인 해시 버킷들로 구성되고, 각 물리적인 해시 버킷은 웨이들로 구성되고 상기 데이터 블록들을 저장하고, 상기 중복제거 DRAM 메모리 모듈에 저장되는 해시 테이블들의 3차원 어레이, 프로세서, 및 메모리를 포함하되, 상기 메모리는 명령들을 저장하고, 상기 명령들이 상기 프로세서에 의해 실행될 때, 상기 메모리는 상기 중복제거 DRAM 메모리 모듈이 상기 복수의 해시 테이블들 중 해시 테이블의 인접 가상 버킷들 사이에서 사전에 저장된 중복제거된 데이터 블록을 움직이도록 야기하고, 상기 가상 버킷들 각각은 두 개 이상의 물리적인 해시 버킷들로 구성될 수 있다.

[0028] 상기 메모리는 명령들을 더 저장하고, 상기 명령들이 상기 프로세서에 의해 실행될 때, 상기 메모리는 중복제거 DRAM 메모리 모듈이 상기 사전에 저장된 중복제거된 데이터 블록이 움직여진 것으로부터 가상 버킷들 중 하나의 가상 버킷에 인커밍 데이터를 저장하도록 야기할 수 있다.

**발명의 효과**

[0029] 본 발명의 실시 예에 따른 중복제거 DRAM 메모리 모듈에 의하면, 메모리 접근은 감소될 수 있고, DRAM 시스템의 수명은 연장될 수 있다.

**도면의 간단한 설명**

[0030] 본 발명의 이들 및 다른 측면들은 명세서, 청구 범위, 그리고 첨부된 도면을 참조하여 이해될 수 있다.

도 1은 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처의 블록도이다.

도 2는 도 1의 실시 예의 중복제거 메모리 모듈 내 메모리 형태들의 블록도이다.

도 3은 도 2의 실시 예의 해시 테이블 메모리의 해시 테이블의 블록도이다.

도 4는 본 발명의 실시 예에 따른 복수의 해시 테이블 어레이의 블록도이다.

도 5a, 도 5b, 및 도 5c는 본 발명의 실시 예들에 따라 가상 버킷들과 특정 물리적인 버킷들을 연관시키기 위한 합워드들(Hopwords)을 생성하기 위한 2차원의 어레이들을 묘사한다.

도 6은 본 발명의 실시 예에 따른 해시 테이블 메모리의 데이터 블록들의 어드레싱을 위한 물리적인 라인 ID(PLID)의 블록도이다.

도 7은 본 발명의 실시 예에 따른, 합스카치 방법을 사용하는 메모리 모듈의 복수의 해시 테이블 어레이로 데이터를 쓰는 과정을 보여주는 순서도이다.

도 8은 본 발명의 실시 예에 따른, 메모리 모듈의 복수의 해시 테이블 어레이로부터 데이터를 읽는 과정을 보여주는 순서도이다.

**발명을 실시하기 위한 구체적인 내용**

[0031] 본 발명의 특징, 그리고 그것을 달성하는 방법은 실시 예의 상세한 설명 및 첨부된 도면들을 참조하면 명확해질 것이다. 이하, 예시적인 실시 예들은 유사한 참조 번호들은 유사한 구성 요소들을 지칭하는 첨부 도면을 참조하여 상세히 설명될 것이다. 그러나 본 발명은 여러 가지 다양한 형태로 구현될 수 있고, 본 명세서에서 단지 예시된 실시 예에 한정되는 것은 아니다. 오히려, 이러한 실시 예들은 이 개시가 철저하고 완전하게 되기 위한 예로서 제공되며, 당업자에게 본 발명의 특징 및 기능을 완전하게 전달할 것이다. 따라서, 본 발명의 기술 분야에서 통상의 지식을 가진 자가 본 발명의 특징 및 기능을 완벽하게 이해하기 위해 필요하지 않은 프로세스, 요소들, 및 기술들은 설명되지 않을 수 있다. 특별히 언급하지 않는 한, 유사한 참조 번호들은 첨부된 도면들 및 쓰여진 설명에서 유사한 구성요소들을 나타내고, 따라서 그것에 대한 설명은 반복되지 않을 것이다. 도면에서, 구성 요소, 층 및 영역들의 상대적인 크기는 명확성을 위해 과장될 수 있다.

[0032] 비록 여기에서 제 1, 제 2, 제3 등의 용어들은 다양한 요소들, 성분들, 영역들, 층들 그리고/또는 섹션들을 설명하기 위해 사용되지만, 이러한 요소들, 성분들, 영역들, 층들 그리고/또는 섹션들은 이러한 용어들로 인해 제한되지 않는 것으로 이해될 것이다. 이러한 용어들은 다른 요소, 성분, 영역, 층, 또는 섹션으로부터 하나의 요소, 구성, 영역, 층 또는 섹션을 구별하기 위해 사용된다. 따라서, 후술하는 제 1 구성 요소, 성분, 영역, 층, 또는 섹션은 본 발명의 사상 및 범위를 벗어나지 않고, 제 2 구성 요소, 성분, 영역, 층, 또는 섹션을 지칭할 수 있다.



- [0033] 하나의 요소 또는 도면에서 도시된 다른 구성 요소(들) 또는 특징(들)과의 특징적인 관계를 설명하기 위한 설명을 용이하게 하기 위해 “아래의”, “아래”, “낮은”, “특정 부분 아래”, “위에”, “상부”와 같은 공간적이고 상대적인 용어들이 여기에서 사용될 수 있다. 공간적이고 상대적인 용어들은 도면에서 묘사된 방향에 대해 사용 또는 동작에서 장치의 다른 방향들을 포함하도록 의도된 것이 이해될 것이다. 예를 들면, 만약 도면의 장치가 뒤집어지면, 다른 구성 요소들 또는 특징들의 “아래” 또는 “아래의” 또는 “특정 부분 아래”로 설명된 구성요소들은 다른 구성 요소들 또는 특징들의 “위로” 맞춰지게 된다. 따라서, “아래의” 또는 “특정 부분 아래”의 예시적인 용어들은 위 또는 아래 방향 모두를 포함할 수 있다. 장치는 다르게 맞춰질 수 있으며 (예를 들면, 90도 도는 다른 방향으로 회전됨) 그리고 공간적으로 상대적인 기술어들은 그에 따라 해석되어야 한다.
- [0034] 요소, 층, 영역, 또는 성분이 다른 요소, 층, 영역 또는 성분 “에”, “에 결합된”, “에 연결된” 것으로 언급되는 때, 그것은 다른 요소, 층, 영역, 또는 성분 “에 직접적으로”, “에 직접적으로 결합된”, “에 직접적으로 연결된” 일 수 있거나, 하나 또는 그 이상의 사이의 요소들, 층들, 영역들, 또는 성분들이 존재할 수 있다. 또한, 요소 또는 층이 두 개의 요소들 또는 층들 사이로 언급되는 때, 그것은 단지 요소 또는 층이 두 요소들 또는 층들 사이에 있을 수 있거나, 또는 하나 또는 그 이상의 사이의 요소들 또는 층들이 또한 존재할 수 있다.
- [0035] 다음의 예들에서, x축, y축, 그리고 z축은 직각 좌표 시스템의 3가지 축들로 한정되지 않고, 넓은 의미로 해석될 수 있다. 예를 들어, x축, y축, 그리고 z축은 서로 직교할 수 있고, 또는 서로 직교하지 않은 다른 방향들을 나타낼 수 있다.
- [0036] 본 명세서에서 사용된 용어들은 단지 특정한 실시 예들을 설명하기 위한 것이고, 본 발명을 제한하려는 것으로 의도되지 않았다. 본 명세서에서 사용된 바와 같이, 문맥상 명백하게 다르게 뜻하지 않는 한, 단수 형태 “하나”는 복수의 형태도 포함하는 것으로 의도된다. “구성되는”, “구성되고 있는”, “포함하는”, 그리고 “포함하고 있는” 용어들이 본 명세서에서 사용될 때, 이러한 용어들은 정해진 특징들, 정수들, 단계들, 동작들, 요소들, 그리고/또는 성분들이 존재를 명시하나, 하나 또는 그 이상의 다른 특징들, 정수들, 단계들, 동작들, 요소들, 성분들, 그리고/또는 그것들의 그룹들의 추가 또는 존재를 불가능하게 하지 않는다. 본 명세서에서 사용된 바와 같이, “그리고/또는” 용어는 하나 또는 그 이상의 열거된 항목들과 연관된 임의의 그리고 모든 조합들 포함한다. “적어도 하나”와 같은 표현들은 요소들 전체 리스트를 수정하고 그리고 리스트의 개별 요소들을 수정하지 않는다.
- [0037] 본 명세서에서 사용된 바와 같이, “대체로”, “약” 용어 그리고 이와 유사한 용어들은 근사치의 용어들로서 사용되고, 정도의 용어들로서 사용되지 않고. 본 발명의 당업자에 의해 식별되는 측정된 또는 계산된 값들의 고유한 변동을 고려하기 위한 것이다. 또한, 본 발명의 실시 예들을 기술할 때 “할 수 있다”의 사용은 “본 발명의 하나 이상의 실시 예들”을 의미한다. 본 명세서에서 사용된 바와 같이, “사용”, “사용되는”, 그리고 “사용된” 용어들은 “이용”, “이용되는”, 그리고 “이용된” 용어들의 동의어로 각각 간주 될 수 있다. 또한, “예시” 용어는 예 또는 그림을 의미한다.
- [0038] 특정 실시 예는 상이하게 구현 될 수 있는 경우, 특정 프로세스 순서는 설명된 순서와 다르게 수행 될 수 있다. 예를 들면, 설명된 연속적인 두 개의 프로세서들은 동시에 대체적으로 수행되거나 설명된 순서와 반대 순서로 수행될 수도 있다.
- [0039] 본 명세서에서 기술된 본 발명의 실시 예에 따른 전자 또는 전기 장치들 그리고/또는 다른 임의의 관련된 장치들 또는 요소들은 임의의 적합한 하드웨어, 펌웨어(예를 들면, Application Specific Integrated Circuit; ASIC), 소프트웨어, 또는 소프트웨어, 펌웨어, 그리고 하드웨어의 조합을 이용하여 구현될 수 있다. 예를 들어, 이러한 장치들의 다양한 요소들은 하나의 집적 회로(Integrated Circuit; IC) 칩 또는 분리된 IC 칩들로 형성될 수 있다. 또한, 이러한 장치들의 다양한 요소들은 유연한 인쇄 회로 필름(Flexible Printed Circuit Film), TCP(Tape Carrier Package), 인쇄 회로 기판(Printed Circuit Board; PCB) 위에 구현되거나 하나의 기판 위에서 형성될 수 있다. 또한, 이러한 장치들의 다양한 요소들은 컴퓨터 프로그램 명령들을 실행하고 본 명세서에서 설명된 다양한 기능들을 수행하기 위한 다른 시스템 요소들과 상호 작용하는 하나 이상의 컴퓨팅 장치들에서 또는 하나 이상의 프로세서들에서 수행되는 프로세스 또는 스레드(Thread)일 수 있다.
- [0040] 컴퓨터 프로그램 명령들은 예를 들면 RAM(Random Access Memory)과 같은 표준 메모리 장치를 이용하는 컴퓨팅 장치에서 구현되는 메모리 내에 저장된다. 컴퓨터 프로그램 명령들은 또한 예를 들면 CD-ROM, 플래시 드라이브(Flash Drive), 또는 그와 같은 다른 일시적이지만 읽기 가능한 미디어(Non-transitory Computer

Readable Media)에 저장될 수도 있다. 또한, 본 발명의 당업자는 본 발명의 예시적인 실시 예들의 사상 및 범위를 벗어나지 않고 다양한 컴퓨팅 장치들의 기능은 단일 컴퓨팅 장치에 통합되거나 집적되고, 특정 컴퓨팅 장치의 기능이 하나 또는 그 이상의 다른 컴퓨팅 장치들에 분산될 수 있음을 인식해야 한다.

[0041] 다르게 정의되지 않는 한, 본 명세서에서 사용된 모든 용어들(기술적 그리고 과학적 용어들을 포함하는)은 본 발명이 속한 기술 분야에서 당업자에 의해 일반적으로 이해되는 동일한 의미를 갖는다. 일반적으로 사용되는 사전에 정의된 이러한 용어들은 본 명세서 그리고/또는 관련 기술의 문맥에서 그들의 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의되지 않는 한 이상화되거나 지나치게 형식적인 감각으로 해석되지 않아야 한다.

[0042] 도 1은 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처의 블록도이다.

[0043] 도 1을 참조하면, 컴퓨터 메모리로서 기능하기 위해, 중복제거 메모리는 원본 데이터의 콘텐츠 그리고 중복제거된 고유의 메모리 블록들의 세트(Set) 사이의 관계를 기록하기 위한 “변환(Translation)”으로 알려진 기능을 수행하고, 기록된 관계는 압축된 형태로 기록된다. 예를 들면, 원본 데이터의 어드레스들은 검색 테이블(Lookup Table)에 저장될 수 있다.

[0044] 일반적으로, CPU의 프로세서(Processor, 110)는 물리적인 메모리(예를 들면, 중복제거 DRAM 메모리 모듈(130))로 직접적인 접근이 어렵고, 상술한 접근은 메모리 라인들(Memory Lines)의 어레이(Array)로 메모리 컨트롤러(120)에 의해 대신 관리된다. CPU 중심의 중복제거 시스템은 데이터가 메모리 시스템에 도달하기 전에 CPU 내부의 캐시 데이터를 구한다.

[0045] 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처(Deduplication DRAM System Architecture, 100)는 종래의 CPU 중심의 중복제거가 아닌 메모리 중심의 중복제거를 사용하고, 이는 중복제거 DRAM 메모리 모듈(130)은 프로세서(110)로부터의 명령들 없이도 메모리 중복제거를 수행하는 것을 의미한다. 중복제거 DRAM 시스템 아키텍처(100)는 또한 메모리의 용량 이득을 증가시키고, 그렇게 함으로써 고용량 메모리 솔루션을 제공하기 위해, 중복제거 DRAM 메모리 모듈(130)에 저장된 설정 가능한 중복제거 알고리즘을 사용한다. 즉, CPU 중심의 중복제거와 달리, 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처(100)는 RAM 모듈(예를 들면, 중복제거 DRAM 메모리 모듈(130)) 내 포함된 모든 중복제거 지능(Intelligence)을 갖는다. 따라서, 중복제거는 CPU 모듈(140)이 모르는 사이에 중복제거 DRAM 메모리 모듈(130) 내에서 수행되는 것이 가능하고, 그렇게 함으로써 중복제거 DRAM 메모리 모듈(130)의 용량이 증가되게 한다. 즉, 중복제거는 미세 혹은 정교(fine grain)하며, 그리고 휘발성 메모리(예를 들면, 중복제거 DRAM 메모리 모듈(130)) 내에서 동작하기 때문에, 본 발명의 실시 예에 따른 모든 중복제거 정보는 중복제거 DRAM 메모리 모듈(130) 자체 내에서 발생하고, 반면에 CPU 내 커널 모듈(Kernel Module, 140)은 중복제거 DRAM 메모리 모듈(130) 내에서 수행되는 중복제거 동작들의 세부 사항을 알지 못할 수 있다.

[0046] 본 발명의 실시 예는 중복제거 DRAM 메모리 모듈(130)로서 DRAM을 사용하여 설명하였으나, 다른 종류의 메모리가 본 발명의 다른 실시 예들에 사용될 수 있음이 이해되어야 한다. 또한, 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처(100)는 다양한 종류의 메모리와 인터페이싱(Interfacing)을 지원하는 것이 가능하다. 즉, 본 발명의 실시 예에 따른 중복제거 DRAM 메모리 모듈(130)은 메모리 컨트롤러(120)를 통해 다양한 다른 종류의 메모리 인터페이스들과 연관될 수 있다(예를 들면, DDR4(Double Data Rate Fourth-Generation Synchronous Dynamic Random-Access Memory), 컴퓨터와 하나 이상의 주변 장치들을 연결하기 위한 직렬 확장 버스 표준인 PCIe(Peripheral Component Interconnect Express), DDR-T, 그리고 KTI). 따라서, 중복제거 DRAM 시스템 아키텍처(100)에 중복제거 DRAM 메모리 모듈(130)을 집적하기 위해 다른 아키텍처들이 사용될 수 있다는 것을 주목해야 한다.

[0047] 또한, 본 발명의 실시 예를 구현하기 위해, 기존 DRAM 메모리 모듈에 약간의 변화(예를 들면, 드라이버 업그레이드(Driver Upgrade))가 있을 수 있지만, 소프트웨어 구현은 운영 체제(Operating System; OS)/CPU 모듈(140) 또는 프로세서(110)의 물리적인 변화 없이 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처(100)를 사용하게 할 수 있다.

[0048] 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처(100)는 중복제거, 콘텐츠 주소부여(Content Addressability), 보안(Security), 메모리 내 처리(Processor-in-memory; PIM), 연관된 어드레스가 그것에 의하여 DRAM 내 데이터 비트가 열 어드레스 및 행 어드레스 등의 교차점에 위치되는 셀에 저장되는 행 어드레스인 것을 DRAM에 알려주기 위해 DRAM에 전송된 신호인 RAS(Row Address Strobe)와 같은 DRAM의 지능적인 프로토콜

을 위해 중복제거 DRAM 메모리 모듈(130)에 SoC(System on Chip)를 구현할 수 있다.

- [0049] 중복제거 DRAM 시스템 아키텍처(100)는 또한 프로세서(110)가 메모리 컨트롤러(120)와 관련된 가상 밀도 관리(Virtual Density Management), 스마트 데이터 배치(Smart Data Placement), 그리고 DRAM 지능적인 APIs(Application Programming Interfaces), 등을 가능하게 하는 것을 야기하는 스마트 시스템 소프트웨어(Smart System Software)를 가질 수 있다.
- [0050] 중복제거 DRAM 메모리 모듈(130)은 다양한 폼 팩터들(form factors, 예를 들면, DIMM(Dual In-line Memory Module), 2.5In, FHHL(Full Height Half Length), HHHL(Half Height Half Length), FHFL(Full Height Full Length), 등)의 고용량 DRAM 메모리 모듈과 같은 3DS DRAM 구성 요소들을 더 가질 수 있다.
- [0051] 따라서, 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처(100)를 사용하는 메모리 중심의 중복제거 시스템을 제공하는 것에 의해, 중복제거 쓰기 프로세스(Deduplicate Write Process)는 메모리 인터페이스에서 직접적으로 수행될 수 있고, 그렇게 함으로써 중복제거 DRAM 메모리 모듈(130)의 용량이 증가된다.
- [0052] 도 2는 도 1의 실시 예의 중복제거 DRAM 메모리 모듈 내 메모리 형태들의 블록도이고, 도 3은 도 2의 실시 예의 해시 테이블 메모리의 해시 테이블의 블록도이다.
- [0053] 도 2를 참조하면, 본 발명의 실시 예에 따른 중복제거 DRAM 메모리 모듈(130)은 중복제거 알고리즘 아키텍처를 가질 수 있고, 여기서 중복제거 메모리 DRAM 모듈(130) 내부의 메모리 영역은 3개의 다른 영역들로 분류된다. 3개의 다른 영역들은 중복제거된 데이터 블록들(Blocks of data)이 저장되는 위치를 나타내기 위한 어드레스 검색 테이블 메모리(Address Lookup Table Memory; 이하 ALUTM, 210), 중복제거된 데이터 블록들을 저장하기 위한 해시 테이블 메모리(Hash Table Memory, 220), 그리고 해시 테이블 메모리의 해시 테이블의 해시 웨이들(Hash ways)이 가득 찬 때 데이터를 저장하기 위한 초과/버퍼 메모리(Overflow/Buffer Memory, 230)를 포함한다.
- [0054] 데이터 블록이 중복제거 DRAM 메모리 모듈(130)로 입력될 때, 중복제거 알고리즘은 데이터 블록이 새로운 것인지, 이전에 저장되지 않은 ALUTM(210) 내 임의의 대응하는 어드레스가 없는 데이터 블록인지를 결정하기 위해 동작할 수 있다. 이러한 동작을 수행하기 위해, 중복제거 알고리즘은 ALUTM(210)에 접근할 것이다. 동일한 데이터 블록들은 단지 하나의 항목으로 저장되는 것을 보장하기 위해, ALUTM(210) 내 포인터(예를 들면, 아래의 도 5에서 설명되는 물리적인 라인 ID(Physical Line ID; PLID))는 해시 테이블 메모리(220)에 동일한 데이터 블록 저장 위치를 나타낸다. 즉, ALUTM(210)은 해시 테이블 내에서 검색 어드레스 맵핑 포인터(Lookup Address Mapping Pointer, 예를 들면 PLID)와 연관되는 위치들(예를 들면, 어드레스들)을 위한 저장 장치이다. 따라서, 데이터 블록이 해시 테이블 메모리(220)에 이전에 저장되었다면, ALUTM(210) 내 포인터는 동일한 데이터 블록이 저장된 해시 테이블 메모리(220)의 어드레스를 가리키는 것이 가능하고, 그렇게 함으로써 데이터 블록의 중복 카피를 저장할 필요를 제거하고, 중복제거 DRAM 메모리 모듈(130)의 용량은 증가한다.
- [0055] 도 3을 참조하면, 메모리 중복제거는 높은 수준의 중복제거, 그리고 그에 대응하는, 중복제거 DRAM 메모리 모듈(130)의 큰 메모리 용량을 보장하기 위해, 상대적으로 효율적이지만 간단한 다중 방식 해시 테이블/해시 어레이(Multiple-way Hash Table/Hash Array, 380)를 사용할 수 있다. 본 발명의 실시 예에 따른 중복제거 DRAM 메모리 모듈(130)의 해시 테이블 메모리(220)는 하나 이상의 해시 테이블들(380)이 있고, 데이터 블록이 유일한지 여부를 결정하는데 있어서 그것들의 유용성을 위해 사용된다. 해시 테이블(380)은 해시 버킷들(Hash Buckets, 행들, 310) 그리고 해시 웨이들(Hash Ways, 열들, 320)로 구성되는 2차원 어레이로 생각될 수 있다. 즉, 본 발명의 실시 예에 따른 해시 테이블(380)은 m개의 해시 버킷들(310)을 행으로 포함하고, 각 해시 버킷(310)은 해시 버킷(310)의 용량을 나타내는 데이터 라인들/슬롯들/항목들/해시 웨이들(320)의 n개 열들을 포함한다(m 및 n은 정수).
- [0056] 해시 테이블 메모리(220)의 해시 웨이들(320)에 데이터 블록들이 저장되고, ALUTM(210)의 어드레스 포인터들은 특정 데이터 블록과 연관된 특정 해시 버킷(310) 그리고 특정 해시 웨이(320)를 나타내는 값들을 저장할 수 있다. 따라서, 어드레스(예를 들면, 64-비트 어드레스)는 ALUTM(210)로 색인될 수 있고, 그것으로부터, 어드레스에 대응하는 데이터 블록을 저장하는 해시 테이블(380)의 해시 버킷(310)의 연관된 해시 웨이(320)가 결정될 수 있다.
- [0057] 따라서, 쓰기 프로세스(예를 들면, 64-바이트(64-byte)의 데이터 쓰기) 동안, 쓰기 요청(즉, 하나 이상의 데이터 블록들로 구성되는 인커밍(Incoming) 데이터를 기록하기 위한 요청)을 수신한 후, 해시 값이 해당 해시 버킷(310) 및 해시 웨이(320)가 결정될 수 있기 위해 해시 함수/해시 알고리즘(즉, 인커밍 데이터가 “해시됨”)을

이용하여 인커밍 데이터에 대해 계산된다. 따라서, 해시 값은 데이터 블록이 어디에 배치되는지를 나타내거나, 또는, 데이터 블록(예를 들면, 64 바이트의 데이터 블록)이 중복인 경우, 해시 값은 데이터 블록이 해시 테이블 메모리(220)에 이미 저장된 위치를 나타낸다. 메모리에 데이터 콘텐츠가 추가됨에 따라, m 해시 버킷(310) 중 일부는 먼저 포화 상태에 도달할 수 있다. 따라서, 중복제거 DRAM 메모리 모듈(130)은 해시 테이블 메모리(220)로 들어갈 수 없는 데이터 블록들을 저장하기 위한 버퍼 메모리(230)를 사용하는 초과 제공(Overflow Provision)을 포함한다. 그 후, 원본 검색 어드레스(Original Lookup Address)는 검색될 수 있고, ALUTM(210)은 인커밍 데이터의 해싱(Hashing)으로부터 계산된 검색 어드레스에 따라 업데이트될 수 있다.

[0058] 시도된 쓰기 프로세스 동안, 해시 웨이들(320) 모두가 가득 찬 것으로 판단될 때, 버퍼 메모리(230)가 사용될 수 있다. 즉, 해시 테이블(380)이 가득 차면, 데이터는 버퍼 메모리(230)의 중복제거 되지 않은 초과 영역(non-deduplicated overflow region)에 배치될 수 있고, 그렇게 함으로써 중복제거 수준이 감소된다. 따라서, 버퍼 메모리(230)는 근본적으로 예약된, 표준의, 간단한 초과 메모리 영역이고, 가상 밀도 과잉-제공 관리 초과(virtual density over-provision management overflow)를 구현하기 위한 SOC 메모리 버퍼/캐시로 제공된다. 데이터가 버퍼 메모리(230)에 일단 배치되면, 더 이상 해시되지 않고, 그리고 더 이상 중복제거될 수 없다.

[0059] 컴퓨터 어플리케이션이 여러 번에 걸쳐 메모리에 동일한 시퀀스의 값을 저장하도록 시도하면, ALUTM(210)에 저장된 변환 어레이의 다중 항목들은 해시 테이블 메모리(220)에 저장된 데이터 블록의 동일한 어드레스를 참조하고, 여기서, ALUTM(210)의 항목들은 원본의 고유한 데이터 블록들보다 작고, 그렇게 함으로써 효율적인 압축이 달성될 수 있다.

[0060] m개의 해시 버킷들(310) 각각은 해시 버킷(310)의 해당 해시 웨이(320)를 나타내기 위한 고유한 식별자를 포함하는 참조/빈도 카운트 라인(340), 그리고 서명 라인(Signature Line, 330)을 더 포함할 수 있다. 각 해시 버킷(310)을 위한, 해당 서명 라인(330)은 비어있는 라인(Free Line)을 나타내기 위한 0, 또는 콘텐츠 검색 최적화를 위한 0이 아닌 이차적인 해시 값 중 어느 하나를 포함한다. 따라서, 콘텐츠 검색을 위해, 일반적으로 서명 라인의 0 항목에 기초하여 비어 있는 라인이 할당되도록 요청하는 서명 일치가 없거나, 후속의 데이터 라인 읽기 및 콘텐츠 비교가 중복의 존재를 확인하게끔 하는 하나의 서명 일치 존재한다. m개의 해시 버킷들(310) 각각은 아래의 도 5a, 도 5b, 및 도 5c에서 설명되는 합워드 라인(Hopword Line)을 더 포함할 수 있다.

[0061] 물리적인 라인 ID(PLID, 350)는 데이터를 해시 테이블(380)로 색인 하기 위해 사용될 수 있다. PLID(350)는 ALUTM(210), 해시 테이블 메모리(220), 또는 버퍼 메모리(230) 중 어느 하나로 구별되는 메모리 라인들을 식별하기 위해 사용될 수 있다. 각 메모리 라인은 해시 테이블(380)에 고유의 콘텐츠를 저장하기 위한 데이터 라인, 또는 각각의 PLID들(350)을 저장하고 프로세서 버스 어드레스(Processor Bus Address)로부터 해시 테이블(380)의 중복제거된 데이터 블록으로의 맵핑을 제공하기 위한 변환 라인(Translation Line) 중 어느 하나에 관련될 수 있다. 즉, 버스 어드레스는 변환 라인을 식별하고, 차례로 특정 데이터 라인을 지정하는 관련 있는 PLID(350)가 포함된 변환 라인 내 항목을 더 식별한다. 따라서, PLID(350)는 초과 태그(Overflow Tag)를 포함하도록 구현될 수 있고, 특정 해당 해시 테이블(380), 해당 해시 버킷 비트들, 그리고 PLID(350)에 대응하는 데이터 블록의 위치를 나타내는 해당 웨이 비트들(Way Bits)을 나타내기 위한 데이터를 포함할 수 있다.

[0062] 각 해시 버킷(310)에는, 해시 버킷들(310)로 데이터를 색인 하기 위해 사용되는  $\log_2(m)$ -비트 해시를 생성하는 알고리즘인 해시 함수/해시 알고리즘 “h(x)”와 연관된 것이 있다(예를 들면, 해시 테이블(380)이 8개 물리적인 해시 버킷들(310)을 갖고 있다면, 해시 테이블(380)의 해시 함수는 3-비트 해시를 생성할 것이다.). 즉, 해시 함수 h(x)는 상대적으로 큰 입력 데이터 양(예를 들면, 메모리에 저장되는 입력 데이터 파일)을 해시 함수 h(x)로 입력하는 것을 가능하게 하고, 출력 데이터(예를 들면, 해시 값)의 상당히 다른 작은 양이 해시 테이블(380)에 저장되도록 하기 위해 해시 함수 h(x)에 의해 생성되고 출력된다. 따라서, 서로 다른 데이터 세트들은 가끔 같은 해시 값으로 해시될 수 있으므로, 해시 함수 h(x)는 압축을 가능하게 한다.

[0063] 중복제거된 메모리로의 쓰기에 있어서, 데이터 파일에 대응하는 쓰기 요청을 수신한 후, 중복제거된 메모리는 첫 번째로 동일/중복 데이터 블록이 이미 해시 테이블(380)에 저장되어 있는지 여부를 결정하기 위해 중복 검색을 수행한다. 그러면 중복제거된 메모리는 ALUTM(210) 그리고 해시 테이블 메모리(220)의 항목들을 업데이트한다. 예를 들면, 참조/빈도 카운트 라인(340)은 해시 테이블 메모리(220) 내 원본 검색 어드레스(즉, 1씩 감소되는)의 빈도 카운트를 업데이트하여 업데이트될 수 있고, 참조 카운트가 0에 도달하면, 해당 데이터 블록은 삭제된다. 뿐만 아니라, 새로운 PLID(350)가 ALUTM(210)에서 생성된다.

[0064] 콘텐츠 검색으로 지칭될 수 있는 중복 검색 동안, 중복제거 DRAM 메모리 모듈(130)은 쓰여지기 위해 의도되는 데이터 파일 또는 그 일부의 기존의 인스턴스들(Pre-existing instances)을 찾는다. 해시 테이블 메모리(220)에

저장된 데이터의 기존의 인스턴트가 있는 경우, 중복 검색은 해당 데이터 라인을 가리키는 PLID(350)를 돌려준다. 데이터의 기존의 인스턴스를 찾지 못하면, 해당 데이터 블록을 위한 새로운 데이터 라인이 해시 테이블(380)의 공간을 할당하고, 그곳에 콘텐츠를 쓰고, 그리고 새로운 PLID(350)를 돌려주는 것에 의해 생성된다. 콘텐츠는 버스 어드레스에 의해 결정된 오프셋에서 ALUTM(210)에 PLID(350)을 저장하는 것에 의해 기록될 수 있다.

[0065] 해시 테이블(380)로 데이터 라인 “C” 를 삽입하기 위해, C의 해당 해시 함수 “h(C)” 는 수학적 연산으로 계산될 수 있다. 데이터 라인 C를 위해 해시 함수가 계산되면, 해시 테이블 T(h(C))의 행은 데이터 라인 C의 삽입을 허용하기 위해 사용 가능한 충분한 공간이 있는지를 알기 위한(또는 해시 테이블(380)에 중복되는 데이터 라인 C가 이미 있는지를 알기 위한) 콘텐츠 검색 동작에 의해 체크될 수 있다.

[0066] 전술한 바와 같이, 해시 테이블(380)의 각 해시 버킷(310)은 추가적으로 서명 라인(330) 그리고 참조/빈도 카운트 라인(340)을 더 포함하고, 서명 라인(330)의 서명들(332) 그리고 참조/빈도 카운트 라인(340)의 참조 카운트들(342)이 여러 수량들을 각 해시 버킷(310)에 채우기에 충분하도록 작게 설계될 수 있다는 사실로 인해 서명 라인(330) 그리고 참조/빈도 카운트 라인(340) 각각은 단지 하나의 해시 웨이(320)를 점유할 수 있다. 즉, 해시 테이블(380)에서, 해시 테이블(380)의 하나의 전체 열은 각각 해시 버킷들(310)에 속하는 서명 라인들(330)에 할당될 수 있고, 하나의 전체 열은 각각 해시 버킷들(310)에 속하는 참조/빈도 카운트 라인들(340)에 할당될 수 있다.

[0067] 데이터 라인 “C” 와 같은 실제 데이터 블록들이 해시 테이블(380)에 더해지면서, 해시 테이블(380)에는 ALUTM(210)에 저장된 해당 PLID(350)를 각 개별적인 중복제거된 데이터 라인의 해시 테이블(380) 내 어드레스에 매칭함으로써 나중에 접근될 수 있는 데이터가 채워지기 시작한다. 해시 테이블(380) 내 어드레스는 데이터가 위치하는 특정 해시 버킷(310) 그리고 특정 해시 웨이(320)를 식별하는 것(예를 들면, 해시 테이블(380)의 행 및 열을 식별하는 것)에 의해 식별될 수 있다. 따라서, 해시 테이블(380)에 저장되는 각 데이터 블록을 위해, 데이터 블록의 위치를 가리키는 ALUTM(210)에 저장되는 해당 PLID들(350)에 의해 식별되는 하나 이상의 해당 어드레스들이 있다. 해시 테이블(380)이 데이터로 가득 채워지면, 새로이 도입되는 데이터는 중복제거 되지 않은 초과 영역/버퍼 메모리(230)에 배치되고, 그렇게 함으로써 중복제거 수준이 감소된다.

[0068] 중복제거된 메모리로부터의 읽기에서, 중복제거된 메모리는 해시 테이블 메모리(220)로부터의 데이터 라인 또는 버퍼 메모리(230)로부터의 초과 라인 중 어느 하나의 복사본을 돌려준다. 예를 들면, 저장된 데이터가 읽혀지는 때, 읽기 요청을 수신한 후, 해시 테이블(380)의 해당 어드레스들이 ALUTM(210)에 저장된 PLID들(350)을 이용하여 검색된다. 그러면, 각 어드레스에 대응하는 블록들이 검색되고 재조립된다.

[0069] 도 4는 본 발명의 실시 예에 따른 복수의 해시 테이블 어레이의 블록도이다.

[0070] 도 4를 참조하면, 본 발명의 실시 예에 따른 중복제거 DRAM 시스템 아키텍처는 복수의 해시 테이블들(Multiple Hash Tables; MHT, 480)로 구성되는 해시 테이블 어레이(400)를 사용하고, 각각의 복수의 해시 테이블들(480)은 m개의 해시 버킷들(410)을 포함하고, 각 해시 버킷(410)은 n개의 해시 웨이들(420)을 포함한다. 비록, 본 발명의 실시 예는 해시 테이블들(480) 그리고 해시 버킷들(410)을 그것들의 크기가 일정한 것으로 설명하지만(예를 들면, m 그리고 n의 정수로서 설명된), 다른 실시 예들에 있어서, 동일한 복수의 해시 테이블 어레이 내에서 다른 해시 테이블들은 다른 수의 해시 버킷들을 가질 수 있고, 그리고 유사하게, 복수의 해시 테이블 어레이 내에서 다른 해시 버킷들, 또는 심지어 동일한 해시 테이블 내에서, 다른 수의 해시 웨이들을 가질 수 있다. 뿐만 아니라, 복수의 해시 테이블들(480)이 집합적으로 이용되어도, 어느 측면에서는, 서로 다른 해시 테이블들(480)은 서로 독립적이다(예를 들면, 서로 다른 해시 테이블들(480)은 각각의 서로 다른 해시 함수들을 가질 수 있거나, 공통된 해시 함수를 가질 수 있다.).

[0071] 해시 테이블의 어레이(400)가 k개의 병렬 해시 테이블들( $T_1, T_2, \dots, T_k$ , k는 정수)을 포함하고, 각 해시 테이블(480)은 분리되고, 독립적인 해시 함수들( $h_1(x), h_2(x), \dots, h_k(x)$ )을 각각 사용하는 경우, 각각의 해시 테이블들( $T_1, T_2, \dots, T_k$ )은 m개의 해시 버킷들(410)을 포함하기 때문에, 해시 함수들( $h_1(x), h_2(x), \dots, h_k(x)$ )은 여전히  $\log_2(m)$ -비트의 해시들을 생성하고, 그리고 각 물리적인 버킷(410)은 n개의 해시 웨이들(420)을 포함하기 때문에, 3차원(3D)의 해시 테이블 어레이(예를 들면, 복수의 해시 테이블들의 어레이)의 용량은  $m \times n \times k$  다.

[0072] 각 해시 테이블(480)은 어떻게 데이터가 색인되는지 결정하는 하나의 해시 함수와 부합할 수 있다. 쓰여지기 위한 인커밍 데이터를 해싱하는 것에 의해, 결과 계산(예를 들면, 검색 어드레스 및 키를 포함하는 해시 값)은 키 그리고 값이 비교될 수 있고, 그리고 값이 일치하는 경우, 해당 해시 버킷(410)의 참조/빈도 카운트 라인(340)

은 증가되고, 그렇게 함으로써 ALUTM(210)의 추가적인 PLID(350)는 특정 라인을 가리키는 것을 나타낸다.

- [0073] 종래 해시 테이블들과 달리, 본 발명의 실시 예에 따른 복수의 해시 테이블들(480)은 복수의 가상 해시 버킷들/가상 버킷들(Virtual Bucket; VB, 460)을 포함하고, 가상 버킷들(460)은 복수의 물리적인 해시 버킷들/물리적인 버킷들(410)로 구성된다. 이하에서 “물리적인 버킷”은 전술한 해시 버킷들(310)을 나타낼 것이고, 가상 버킷들(460)로부터 전술한 해시 버킷들(310)을 구별하기 위해 사용될 것이다.
- [0074] 각 가상 버킷(460)은 해당 해시 테이블(480)의 m개 물리적인 버킷들(410)의 H를 포함할 수 있고, H는 m 보다 작은 정수이다. 그러나, 동일한 해시 테이블(480)에서 가상 버킷들(460) 중 다른 가상 버킷들은 하나 이상의 물리적인 버킷들(410)을 공유할 수 있음을 주목해야 한다. 후술하는 바와 같이, 본 발명의 실시 예에 따른 가상 버킷들(460)을 사용하는 것에 의해, 제 4 차원이 3차원의 복수의 해시 테이블 어레이에 더해진다. 따라서, 데이터를 배치하고 정렬하는데 있어서 큰 유연성이 제공될 수 있고, 그렇게 함으로써 중복제거 DRAM 시스템 아키텍처의 효율 및 압축 비율이 높아진다.
- [0075] 본 발명의 실시 예는 또 다른 수준의 데이터 배치 유연성을 높이기 위해 가상 버킷들(460)을 사용하는데, 다른 가상 버킷들(460)에 의해 공유되는 다른 물리적인 버킷들(410)을 자유롭게 하기 위해, 해시 테이블들(480) 중 어느 하나에 저장된 데이터 블록은 해당 가상 버킷(460) 내에서, 또는 다른 물리적인 버킷(410)으로 움직이게 될 수 있다. 해시 테이블(480) 내에서 공간을 자유롭게 함으로써, 중복제거는 쓸모없거나/중복된 데이터를 제거함으로써 달성될 수 있다. 즉, 본 발명의 실시 예에 따른 가상 버킷들(460)의 사용에 의해, 제한된 해당 위치로의 해시 함수를 이용하여 데이터 라인을 해싱함으로 인하여 야기되는 엄격한 제한은 없고, 그리고 데이터는 “가까운 위치”의 해시 버킷(410)에 배치되는 것이 가능하고, 이는 초기 의도된(그러나 점유된) 물리적인 해시 버킷(410)을 포함하는 동일한 가상 버킷(460) 내에 존재하는 물리적인 버킷(410)을 나타낸다.
- [0076] 하나의 예에서, 컨텐츠(예를 들면, 데이터 라인 C)는 k개의 해시 테이블들( $T_1(h_1(C))$ ,  $T_2(h_2(C))$ , ...,  $T_k(h_k(C))$ ) 중 어느 하나의 물리적인 버킷들(410) 중 어느 하나로 배치될 수 있다. 만약 데이터 라인 C가  $T_1(h_1(C))$ 에 배치되고자 한다면, 데이터 라인 C가  $T_1(h_1(C))$ 로 표시되는 물리적인 버킷(410)으로 배치되기 위한 요청을 하는 대신에, 본 발명의 실시 예는 하나의 물리적인 버킷(410)보다 더 크고, 그리고  $T_1(h_1(C))$ 으로 표시되는 물리적인 버킷(410) 뿐만 아니라 H의 총 물리적인 버킷들(460)을 포함하는 가상 버킷(460)을 허용한다. 즉, 가상 버킷(460)은  $T_1(h_1(C))$ ,  $T_1(h_1(C)+1)$ ,  $T_1(h_1(C)+2)$ , ...,  $T_1(h_1(C)+H-1)$ 을 포함하는 해시 테이블(480) 내에서 정렬된 근접적이거나 인접한 H 물리적인 버킷들(410) 총합을 포함한다.
- [0077] 따라서, 가상 버킷들(460)은 데이터 블록들이 해시 테이블(480) 내에서 움직이게 되거나 미래 쓰기 동작들을 위해 공간을 자유롭게 하는 것을 가능하게 한다. 이전에 해시 테이블(480) (해시 테이블(480)의 물리적인 버킷들(410)을 포함하는 가상 버킷들(460) 내)로 들어간 데이터 블록들의 이동을 가능하게 하는 본 발명의 실시 예의 동작은 합스카치(Hopscotch)로 나타낼 수 있다. 메모리 중복제거를 위한 복수의 해시 테이블들(480)을 사용하는 합스카치 동작은 후술하는 바와 같이 개선될 수 있다.
- [0078] 첫 번째로, 중복제거 DRAM 모듈(130)은 해시 테이블(480)의 해시 함수의 결과로서 해시 테이블(480)로 데이터 라인 C를 삽입하려는 시도를 할 수 있다. 그러나, 종종 다른 데이터 라인이 동일한 해시 함수의 결과로 이전에 해시 테이블(480)에 들어갔을 수 있다. 즉, 다른 데이터 라인들은, 그것이 다름에도 불구하고, 해시 함수의 결과로서 해시 테이블(480)내 동일한 위치에 보내질 수 있다. 데이터 라인 C가 어디에 삽입되는지 결정하기 위해, 동작은 우선  $T(h(C))$ 로서 표현되는 물리적인 버킷(410)에서(또는 후속의) 첫 번째로 사용 가능한 물리적인 버킷(410)을 찾을 수 있다.
- [0079] 따라서, 데이터 라인 C를 어디에 쓸지 결정하는데 있어서,  $T(h(C))$ 로서 표현되는 초기에 의도된 물리적인 버킷(410)은 점유될 수 있으므로, 첫 번째로 사용 가능한 물리적인 버킷(410, 즉, 데이터 라인이 삽입될 수 있는 첫 번째 빈 공간)은  $T(h(C)+f)$ 로서 표현될 수 있고, 여기서 f는 0 이상이다.  $T(h(C))$ 로서 표현되는 물리적인 버킷(410)이 해당 가상 버킷(460)의 H 물리적인 버킷들(410)의 첫 번째 물리적인 버킷(410)이라 가정하면, 만약 f가 H보다 작다면(즉, 만약 동일 가상 버킷(460) 내 점유되지 않은 물리적인 버킷(410)이 존재하면), 데이터 라인 C는 해당 가상 버킷(460)으로 배치될 수 있다. 유사하게,  $T(h(C))$ 로서 표현되는 물리적인 버킷(410)이 해당 가상 버킷(460)의 두 번째 물리적인 버킷이면, f가 H-1보다 작으면, 데이터 라인 C는 해당 가상 버킷(460)으로 배치될 수 있다.
- [0080] 그러나, 해당 가상 버킷(460)의 첫 번째 물리적인 버킷(410)이 의도된 물리적인 버킷(410)이라 가정하고, 만약 f가 H보다 크거나 같으면(즉, 가상 버킷(460)의 물리적인 버킷(410)으로 데이터 라인 C가 들어갈 수 없는 경

우), 비록 데이터 라인 C가 그것의 가상 버킷(460)과 들어맞지 않을 지라도, 동작은 다음과 같은 방식으로 가상 버킷(460) 내 빈 공간을 만들기 위한 시도를 할 수 있다. 예를 들면, 본 발명의 실시 예에 따른 중복제거 DRAM 메모리 모듈(130)은  $T(h(C)+f-1)$ 로 표현되는 물리적인 버킷(410)이 그 안에 포함된 데이터를 갖는지 여부를 결정할 때까지  $T(h(C)+f-H)$ 로 표현되는 물리적인 버킷(410)에서 시작해서 물리적인 버킷들(410)을 찾을 수 있고, 그 다음에  $T(h(C)+f-H+1)$ 로 표현되는 물리적인 버킷(410), 그리고 기타 등등을 찾을 수 있다(예를 들면, 가상 버킷(460)의 처음부터 끝까지 스캔 할 수 있다.). 그 다음에, 중복제거 DRAM 메모리 모듈(130)은  $T(h(C)+f-H)$ 로부터  $T(h(C)+f-1)$ 까지의 물리적인 버킷들(410) 내에 들어있는 임의의 데이터 대상이 빈 공간  $T(h(C)+f)$ 으로 배치될 수 있는지 여부를 결정할 수 있다. 즉, 중복제거 DRAM 메모리 모듈(130)은  $T(h(C)+f-H)$ 로부터  $T(h(C)+f-1)$ 까지의 물리적인 버킷들 중 어떤 것이 물리적인 버킷  $T(h(C)+f)$ 을 갖는 공통의 가상 버킷(460) 내에 있는지 여부를 결정할 수 있고, 그렇게 함으로써 그 안에 들어있는 데이터가 움직일 수 있도록 가능케 한다. 그 다음에, 중복제거 DRAM 메모리 모듈(130)은 발견된 가장 초기의 그러한 데이터 대상을 빈 공간 내에 배치할 수 있고, 그렇게 함으로써  $T(h(C)+e$ ,  $e$ 는  $f$ 보다 작은 정수)로 표현되는 물리적인 버킷들(410) 내의 새로운 빈 공간을 만들 수 있다. 이러한 과정은  $e$ 가  $H$ 보다 작아질 때까지 반복될 수 있고(예를 들면, 데이터는 연속적인 방식(Cascading Fashion)으로 해시 테이블 내에서 움직이게 될 수 있다.), 그렇게 함으로써 해당 가상 버킷(460) 내에서 데이터 라인 C의 배치를 하기 위해 필요한 공간을 확보할 수 있다.

[0081] 예를 들면, 도 5b를 참조하면, 본 발명의 실시 예에 있어서, 물리적인 버킷 PB2를 의도된 물리적인 버킷(410)으로 할당할 수 있다. 의도된 물리적인 버킷 PB2는 가상 버킷 VB1과 연관되어 점유되었으므로, 가상 버킷 VB2는 처음부터 끝까지 스캔될 수 있다(예를 들면, 물리적인 버킷 PB2부터 물리적인 버킷 PB5까지). 물리적인 버킷들 PB3, PB4, 그리고 PB5 또한 점유되었으므로, 첫 번째로 사용 가능한 물리적인 버킷(410)은 물리적인 버킷 PB6이다((즉,  $f$ 가 4와 동일하고, 그리고  $f$ 가 그러므로  $H$ 와 같거나  $H$ 보다 크고, 그리고 첫 번째로 사용 가능한 물리적인 버킷(410)은 해당 가상 버킷 VB2에 있지 않다.). 따라서, 물리적인 버킷 PB5 내 데이터는 물리적인 버킷 PB6으로 움직이게 될 수 있고, 그렇게 함으로써 가상 버킷 VB2 내 공간을 확보하고, 데이터 라인 C는 해당 가상 버킷 VB2 내 (물리적인 버킷 PB5 내) 배치될 수 있다. 그러나, 의도된 물리적인 버킷이 PB1 이었다면(즉, 해당 가상 버킷(460)은 VB1), 처리 과정은 물리적인 버킷 PB4 내의 데이터가 가상 버킷 VB1으로부터 인접 가상 버킷 VB2로 즉, 물리적인 버킷 PB5의 새로이 확보된 공간으로 움직이게 될 수 있게 반복될 수 있다. 그 후에, 데이터 라인 C는 의도된 물리적인 버킷 PB1에 해당하는 가상 버킷 VB1의 물리적인 버킷 PB4에 쓰여질 수 있다.

[0082] 따라서, 다른 가상 버킷들(460)의 중복으로 여겨질 수 있는 다른 가상 버킷들(460)에 의한 특정 물리적 버킷들(410)의 공통 소유로 인하여, 데이터는 하나의 가상 버킷(460)에서 다른 가상 버킷(460)으로 움직이게 될 수 있고, 그렇게 함으로써 초기 해시 버킷(410)을 위한 공간을 창출할 수 있다.

[0083] 다른 실시 예에 있어서, 쓰기 과정 동안, 데이터 블록을 해시 테이블들의 어레이(400)에 쓰는 요청을 수신한 후, 중복제거 DRAM 메모리 모듈(130)은 기존의 항목이 이미 해시 테이블들(480) 중 하나에 있는지를 체크하기 위해, 데이터에 대해 의미가 있는 각 해시 테이블의 전체 가상 버킷(460)을 검색할 수 있다. 만약 첫 번째로 의도된 해시 테이블(480)이 가득 찬 경우, 그리고 데이터 블록이 첫 번째로 의도된 해시 테이블(480)에서 찾을 수 없는 경우(즉, 각 물리적인 버킷(410)의 각 해시 웨이(420)가 다른 데이터 블록에 의해 점유된 경우), 중복제거 DRAM 메모리 모듈(130)은 해시 테이블 어레이(400)의 다른 해시 테이블(480)로 데이터를 입력하는 것을 추구할 수 있다. 그러나, 복수의 해시 테이블 어레이(400)의 해시 테이블들(480) 전부가 가득 찬 경우, 데이터 블록은 버퍼 메모리(230)로 넘칠 수 있다(Spill over). 이러한 실시 예에 있어서, 해시 테이블 어레이(400) 내에서 데이터 이동은 중복제거 DRAM 메모리 모듈(130)에 의해 허용되지 않을 수 있다.

[0084] 도 5a, 도 5b, 및 도 5c는 본 발명의 실시 예들에 따라 가상 버킷들과 특정 물리적인 버킷들을 연관시키기 위한 합워드들(Hopwords)을 생성하기 위한 2차원의 어레이들을 묘사한다.

[0085] 도 5a, 도 5b, 및 도 5c를 참조하면, 본 발명의 실시 예에 따라, 다양한 가상 버킷들(460)은 합워드 값(591) 또는 합워드 벡터(592) 중 어느 하나를 이용하는 것 그리고, 데이터 움직임을 효과적으로 쫓기 위한 가상 버킷 이용 값을 이용하는 것에 의해 그들의 해당 물리적인 버킷들(410)과 연관될 수 있다. 각 점유된 물리적인 버킷(410)은 단지 하나의 가상 버킷(460)에 해당할 수 있으므로, 합워드 값(591) 또는 합워드 벡터(492)는 어떤 가상 버킷(460)이 각 점유된 물리적 버킷(410)에 해당하는지를 쫓기 위해 사용될 수 있다.

[0086] 본 발명의 예에서 4개의 가상 버킷들 VB0, VB1, VB2, 및 VB3 각각은 물리적인 버킷들 PB0, PB1, PB2, PB3, PB4, PB5, 및 PB6의 그룹으로부터 4개의 인접한 물리적인 버킷들이 서로 다른 세트를 갖는다(즉,  $H$ 는 4).

[0087] 예를 들면, 도 5a 및 도 5b를 참조하면, 합워드 벡터(592)는 물리적인 버킷 위치들 그리고 가상 버킷 위치들(예

를 들면, 유사 주소들(quasi-addresses))로 구성되는 2차원 어레이를 창출하는 것, 그리고 해당 물리적인 버킷들(410)에 해당하는 임의의 열에는 단지 하나의 1이 있음을 나타내는 각 가상 버킷(460)을 위한 데이터를 포함하는 각 물리적인 버킷(410)에 1(예를 들면, 2진 표시자)을 배치하는 것에 의해 결정될 수 있다. 따라서, 합워드 벡터(592)는 각 가상 버킷(460)을 위한 물리적인 버킷 사용을 쫓기 위해 사용될 수 있는 1들 또는 0들의 어레이를 포함할 수 있다. 본 발명의 예에 있어서, 물리적인 버킷들 PB0, PB1, 그리고 PB3는 첫 번째 가상 버킷 VB0를 위해 점유되고, 물리적인 버킷들 PB2 그리고 PB4는 두 번째 가상 버킷 VB1을 위해 점유되고, 단지 물리적인 버킷 PB5는 세 번째 가상 버킷 VB2를 위해 점유되고, 그리고 네 번째 가상 버킷 VB3는 점유되지 않는다.

[0088] 유사하게, 도 5c를 참조하면, 합워드 값(591)은 어느 가상 버킷들(460)이 점유되는 물리적인 버킷들에 해당되는지를 아는 것에 의해 점유되는 물리적인 버킷들(410)에 기초하여 창출될 수 있다. 합워드 값(591)은  $\log_2(H)$  길이의 비트들이 될 수 있다(H는 가상 버킷(460) 마다 물리적인 해시 버킷들(410)의 개수)

[0089] 합워드 벡터(592) 또는 합워드 값(591)의 정보는 각 해시 버킷(410)의 합워드 라인에 저장될 수 있고, 물리적인 버킷들(410) 그리고 가상 버킷들(460) 사이의 관계는 메모리에 색인될 수 있다.

[0090] 도 6은 본 발명의 실시 예에 따른 해시 테이블 메모리의 데이터 블록들의 어드레싱을 위한 물리적인 라인 ID(PLID)의 블록도이다.

[0091] 도 6을 참조하면, 본 발명의 실시 예에 따라, 수정된 PLID(650)가 제공된다. 본 발명의 실시 예에 의한 PLID(650)는 주소, 오프셋들, 테이블의 색인, 해시, 그리고 슬롯/웨이(Way), 그리고 가상 버킷들(460) 사이의 움직이는 항목들을 쫓기 위한 특정 가상 버킷(460)과 쌍을 이루는 키(651) 각각을 가리키는 복수의 비트들을 포함할 수 있다. 따라서, 만약 키(651)가 특정 가상 버킷(460)과 일치하면, 그 특정 가상 버킷(460)은 그곳에 쓰여진 데이터 항목을 포함할 수 있다.

[0092] 그러나, 다른 실시 예에 있어서, PLID(650)는 키(651)를  $\log_2(H)$  비트들 로 구성되는 가상 버킷 이용 값 필드(652, 예를 들면, 가상 버킷 색인)로 대체한다 (예를 들면, 16개 물리적인 버킷들의 높이를 갖는 가상 버킷은 PLID(650)의 4 비트 가상 버킷 이용 값 필드에 대응된다.). 가상 버킷 이용 값 필드(652)는 각 점유된 물리적인 버킷(410)에 어느 가상 버킷(460)이 대응하는지를 가리킨다. 따라서, 데이터 항목을 가상 버킷(460)에 쓰는 때, 가상 버킷(460)에 이미 존재하는 항목들의 수는 계산될 수 있고, 그리고 가상 버킷에 이미 존재하는 아이템들의 수에서 1을 더한 것과 동일한 P값은 가상 버킷 이용 값(652)으로 쓰여질 수 있다. PLID(650)의 가상 버킷 이용 값(652)을 이용하는 것에 의해, PLID(650)의 스토리지(Storage) 과부하는 감소될 수 있다.

[0093] 도 7은 본 발명의 실시 예에 따른, 합스카치 방법을 사용하는 메모리 모듈의 복수의 해시 테이블 어레이로 데이터를 쓰는 과정을 보여주는 순서도이다.

[0094] 도 7을 참조하면, S701 동작에서, 복수의 해시 테이블들은 식별될 수 있고, 해시 테이블들 각각은 해시 함수에 대응되고, 그리고 각각은 물리적인 해시 버킷들을 포함하고, 각 물리적인 해시 버킷은 해시 웨이들을 포함하고 데이터를 저장하도록 구성된다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 k개의 해시 테이블들(480)을 식별할 수 있고, 각각은 해시 함수  $h(x)$ 에 대응되고, 각각은 m개의 물리적인 해시 버킷들(410)을 포함하고, 각 물리적인 해시 버킷들은 n개의 해시 웨이들(420)을 포함한다.).

[0095] S702 단계에서, 복수의 가상 버킷들이 식별될 수 있고, 가상 버킷들 각각은 물리적인 해시 버킷들의 일부를 포함하고, 그리고 각각은 다른 가상 버킷과 적어도 하나의 물리적인 해시 버킷을 공유한다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 복수의 가상 버킷들(460)을 식별할 수 있고, 가상 버킷들(460) 각각은 m개의 물리적인 해시 버킷들(410)의 H를 포함하고, 그리고 도 4에 도시된 바와 같이, 각 가상 버킷(460)은 다른 가상 버킷(460)과 적어도 하나의 물리적인 해시 버킷들(410)을 공유한다.). S702a 단계에서, 복수의 가상 버킷들은  $\log_2(h)$  비트들의 가상 버킷 이용 값 필드, 그리고 가상 버킷들 중 대응하는 하나의 가상 버킷에서의 데이터 블록들의 수와 동일한 값을 포함하는 물리적인 라인 ID(PLID)로 해시 테이블을 색인하는 것, 그리고 가상 버킷들 중 대응하는 하나의 가상 버킷에 항목이 쓰여지는 때 가상 버킷 이용 값 필드를 1 증가시키는 것에 의해 식별될 수 있다(예를 들면, 도 6에 도시된 바와 같이, 가상 버킷들(460)은 가상 버킷 이용 값 필드(652), 그리고 가상 버킷들(460) 중 대응하는 하나의 가상 버킷에서의 데이터 블록들의 수와 동일한 값을 포함하는 물리적인 라인 ID(PLID, 650)로 해시 테이블(480)을 색인하는 것에 의해 식별될 수 있고, 가상 버킷 이용 값 필드(652)는 가상 버킷들(460) 중 대응하는 하나의 가상 버킷에 항목 또는 데이터 블록이 쓰여질 때 1이 증가될 수 있다.).

[0096] S703 단계에서, 물리적인 해시 버킷들에 저장된 데이터를 갖는 물리적인 해시 버킷들 각각은 가상 버킷들 중 대응하는 하나의 가상 버킷에 할당되는 것으로 식별될 수 있다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 도



55b 및 도 5c에서 도시된 바와 같이 PB0, PB1, PB2, PB3, PB4, 그리고, PB5에 저장된 데이터를 갖는 물리적인 해시 버킷들(410)을 가상 버킷들(460, VB0, VB1, 그리고 VB2) 중 대응하는 하나의 가상 버킷에 할당하여 식별할 수 있다.). S703a 단계에서, 데이터를 포함하는 물리적인 해시 버킷들 중 어떤 것이 가상 버킷들 중 어떤 것에 대응되는지를 나타내기 위한 합워드 벡터 또는 합워드 값을 생성하는 것에 의해 식별될 수 있다(예를 들면, 도 5b 및 도 5c에서 도시된 바와 같이, 중복제거 DRAM 메모리 모듈(130)은 데이터를 포함하는 물리적인 해시 버킷들(410) 중 어떤 것이 가상 버킷들(460) 중 어떤 것에 대응되는지를 나타내기 위한 합워드 벡터(592) 또는 합워드 값(591)을 생성할 수 있다).

[0097] S704 단계에서, 데이터 라인은 해시 값을 생성하기 위해 해시 함수들 중 대응하는 대응하는 하나의 해시 함수에 따라 해시될 수 있다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 메모리 컨트롤러(120)로부터 데이터 라인 C에 해당하는 쓰기 요청을 수신하고, 그리고 해시 값을 생성하기 위해 해시 함수들  $h(x)$  중 대응하는 대응하는 하나의 해시 함수에 따라 인커밍(incoming) 데이터를 해시할 수 있다.).

[0098] S705 단계에서, 해당 해시 테이블의 가상 버킷들 중 대응하는 하나의 가상 버킷이 해시 값에 따른 데이터 블록을 위한 사용 가능한 공간을 갖는지 여부가 결정될 수 있다(예를 들면, 도 5b 및 도 5c에서 도시된 바와 같이, 중복제거 DRAM 메모리 모듈(130)은 가상 버킷(460, VB3)이 데이터 블록을 위한 물리적인 버킷(PB6)에 공간을 갖는지 결정할 수 있다.).

[0099] S706 단계에서, 데이터는 가상 버킷들 중 대응하는 하나의 가상 버킷이 데이터 블록을 위한 공간을 가질 때까지 가상 버킷들 중 대응하는 하나의 가상 버킷이 사용 가능한 공간을 갖지 않는 경우 가상 버킷들 중 대응하는 하나의 가상 버킷으로부터 가상 버킷들 중 인접한 하나의 가상 버킷으로 순차적으로 움직이게 될 수 있다(예를 들면, 도 5b 및 도 5c에서 도시된 바와 같이, 중복제거 DRAM 메모리 모듈(130)은 가상 버킷(VB2)이 데이터 블록을 위한 공간을 가질 때까지 가상 버킷(VB2)이 임의의 다른 사용 가능한 물리적인 버킷들을 갖지 않는 경우 가상 버킷(VB2)의 물리적인 버킷(PB5)으로부터 가상 버킷(VB3)으로 데이터를 순차적으로 움직일 수 있고, 상술한 과정은 만약 가상 버킷(VB1)이 가상 버킷들(460) 중 대응하는 하나의 가상 버킷이면, 가상 버킷(VB1)의 물리적인 버킷(PB4)으로부터 가상 버킷(VB2)의 물리적인 버킷(PB5)으로 데이터를 움직이기 위해 반복될 수 있다.). S706a 단계에서, 어드레스 검색 테이블 메모리는 이동된 데이터 블록에 대응되는 하나 또는 그 이상의 검색 어드레스들로 변경하기 위해 업데이트될 수 있다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 해시 테이블 메모리(220)의 이동된 데이터 블록의 새로운 어드레스가 검색될 수 있도록 이동된 데이터 블록에 대응되는 하나 또는 그 이상의 어드레스 포인터들을 변경하기 위해 ALUTM(210)을 업데이트할 수 있다.).

[0100] S707 단계에서, 데이터 블록은 가상 버킷들 중 대응하는 하나의 가상 버킷에 저장될 수 있다(예를 들면, 도 5b 및 도 5c에 도시된 바와 같이, 중복제거 DRAM 메모리 모듈(130)은 만약 가상 버킷(VB1)이 의도된 가상 버킷(460)이면 데이터 블록을 가상 버킷(VB1)의 물리적인 버킷(PB4)에 저장할 수 있다. ). 만약, 가상 버킷(VB1)을 포함하는 해시 테이블(480)이 가득 찬 것으로 결정되면, 데이터 블록은 버퍼 메모리(230)에 저장될 수 있다.

[0101] 도 8은 본 발명의 실시 예에 따른, 메모리 모듈의 복수의 해시 테이블 어레이로부터 데이터를 읽는 과정을 보여주는 순서도이다.

[0102] S801 단계에서, 해시 테이블 어레이에 저장된 복수의 데이터 블록들에 대응하는 읽기 요청이 수신될 수 있다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 메모리 컨트롤러(120)로부터 데이터 라인 C로 구성되는 복수의 데이터 블록들에 해당하는 읽기 요청을 수신할 수 있고, 복수의 데이터 블록들은 해시 테이블 메모리(220)의 해시 테이블 어레이(400)에 저장되어 있다.).

[0103] S802 단계에서, 복수의 데이터 블록들에 대응하는 포인터들 중 해당하는 포인터들은 ALUTM(210)으로부터 검색될 수 있다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 ALUTM(210)으로부터 데이터 라인 C로 구성되는 복수의 데이터 블록들에 대응하는 어드레스 포인터들을 검색할 수 있다.).

[0104] S803 단계에서, 포인터들 중 해당하는 포인터들에 기초하는 복수의 데이터 블록들은 해시 테이블 메모리에서 접근될 수 있다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 해시 테이블 메모리(220)의 해시 테이블 어레이(400) 내의 검색된 어드레스 포인터들에 대응하는 상이한 어드레스들로부터 데이터 블록들을 접근하고 검색할 수 있다.).

[0105] S804 단계에서, 복수의 데이터 블록들은 재조립된 데이터를 생성하기 위해 재조립될 수 있다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 수신된 읽기 요청에 대응하는 데이터 라인 C와 동등할 수 있는 재조립된 데이터를 생성하기 위해 해시 테이블 메모리(220)로부터 검색된 데이터 블록들을 재조립할 수 있다.).

[0106] S805 단계에서, 제조된 데이터는 메모리 모듈로부터 메모리 컨트롤러로 전송될 수 있다(예를 들면, 중복제거 DRAM 메모리 모듈(130)은 데이터 라인 C를 메모리 컨트롤러(120)로 전송할 수 있다.).

[0107] 전술한대로, 데이터 중복제거는 본 발명의 실시 예에 따른 중복제거 DRAM 메모리 모듈을 사용하여 수행될 수 있다. 따라서, 메모리 접근은 감소될 수 있고, DRAM 시스템의 수명은 연장될 수 있다.

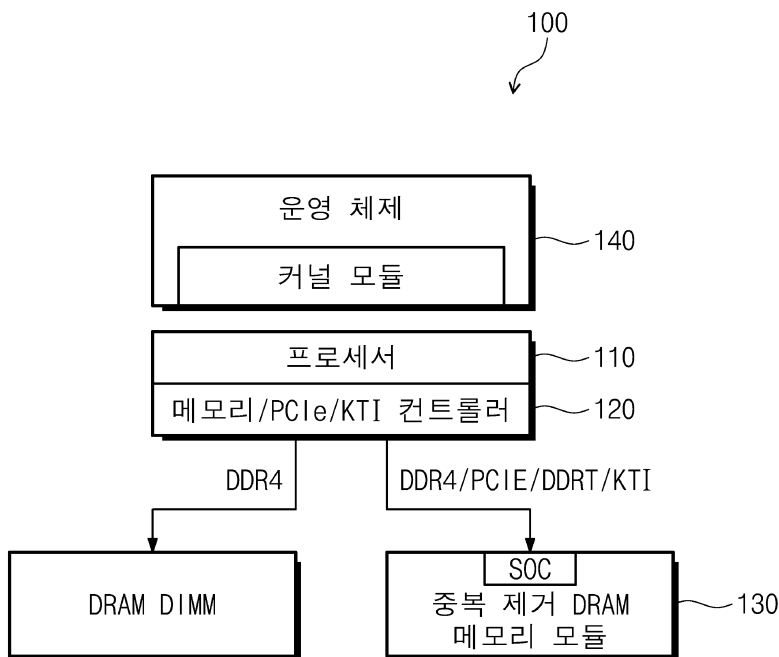
[0108] 전술한 내용은 예시적인 실시 예를 예시하며, 그것에 의해 한정되지 않는다. 비록 몇몇의 예시적인 실시 예들이 설명되었지만, 당업자는 예시적인 실시 예들에서 신규한 교시 및 예시적인 실시 예들의 이점에서 벗어나지 않고 다양한 변형이 가능한 것은 쉽게 이해할 것이다. 따라서, 그러한 모든 변형은 청구 범위에 기재된 예시적인 실시 예의 범위 내에 포함되는 것으로 의도된다. 청구 범위에서, 기능식 절들은 인용된 기능을 수행하는 구조들 그리고 구조적 균등물 뿐만 아니라 균등한 구조물들도 포함하는 것으로 의도된다. 그러므로, 전술한 내용은 개시된 특정한 실시 예들에 제한되지 않고, 그리고 개시된 예시적인 실시 예들의 변형들, 뿐만 아니라 다른 예시적인 실시 예들은 첨부된 청구항의 범위 내에 포함되는 것으로 의도되는 것으로 이해되어야 한다. 본 발명은 청구항에 포함되는 청구항의 균등물 내의, 이하의 청구 범위에 의해 정의된다.

**부호의 설명**

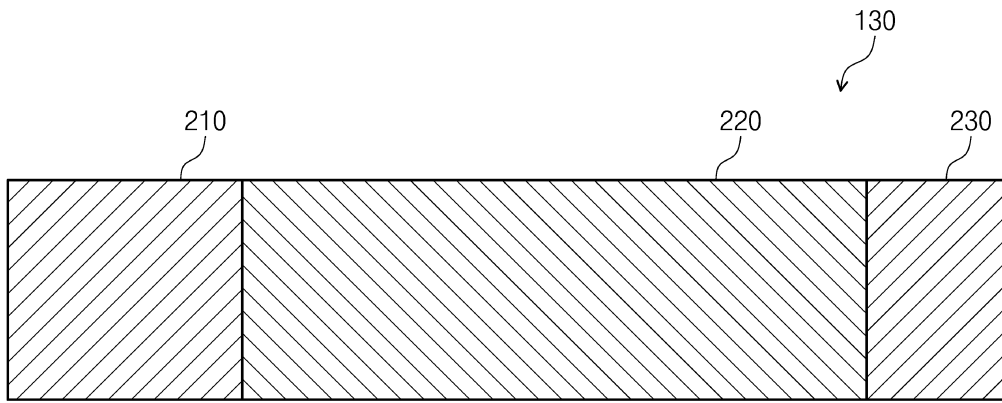
- [0109] 100: 중복제거 DRAM 시스템 아키텍처
- 110: 프로세서
- 120: 메모리/PCIe/KIT 컨트롤러
- 130: 중복제거 DRAM 메모리 모듈
- 140: 운영 체제
- 210: 어드레스 검색 테이블 메모리
- 220: 해시 테이블 메모리
- 230: 버퍼 메모리

**도면**

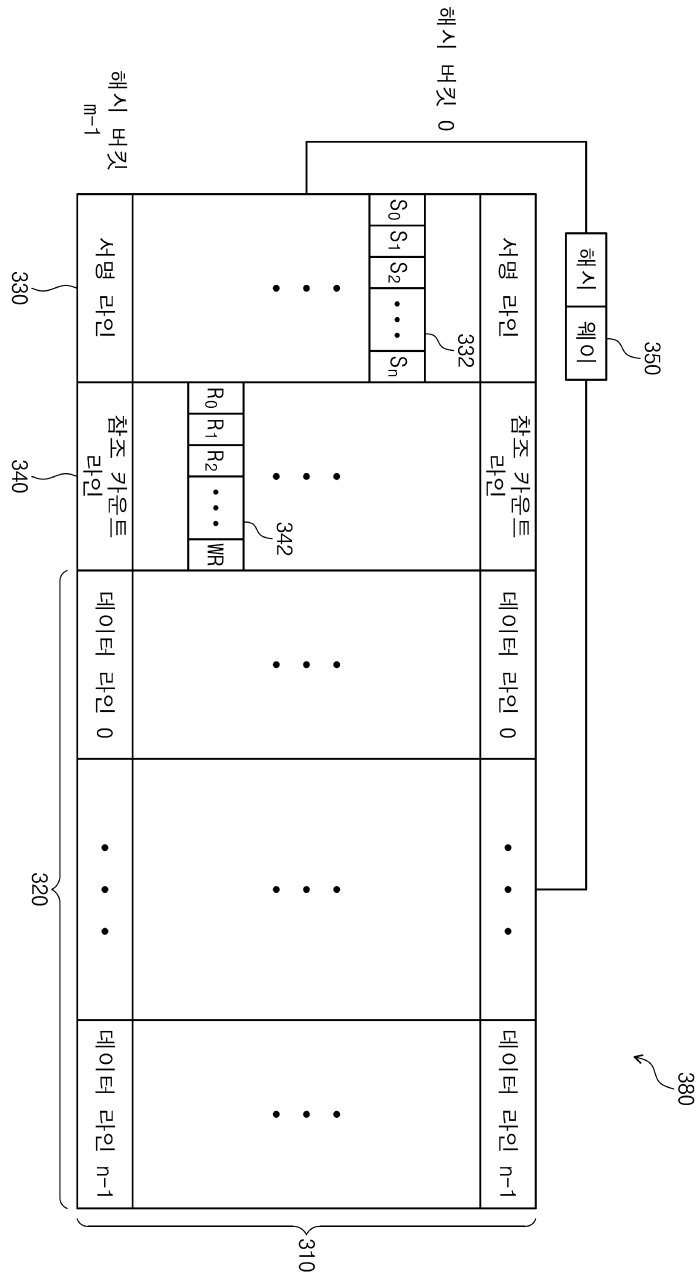
**도면1**



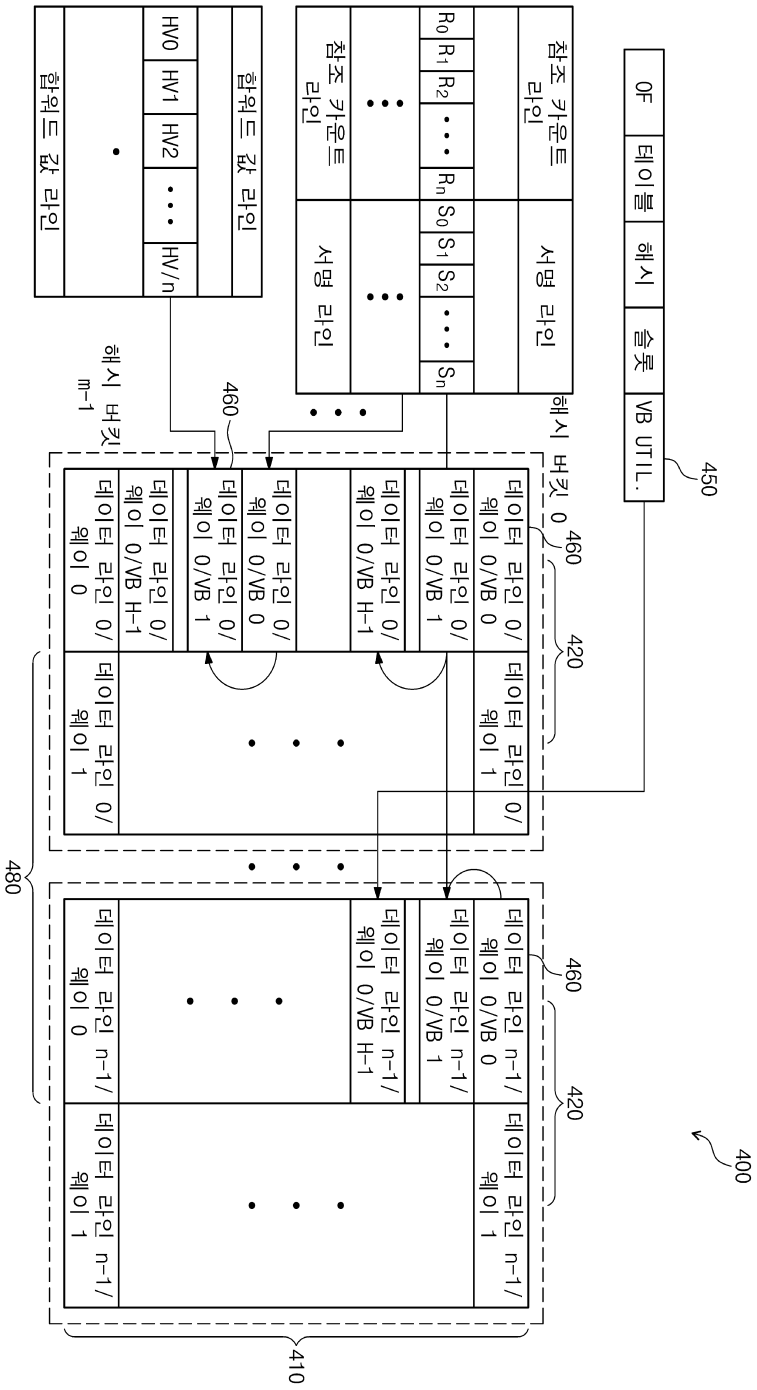
도면2



도면3



도면4



도면5a

	PB0	PB1	PB2	PB3	PB4	PB5	PB6
VB0	X0	X1	X2	X3			
VB1		X0	X1	X2	X3		
VB2			X0	X1	X2	X3	
VB3				X0	X1	X2	X3

410

592

도면5b

	PB0	PB1	PB2	PB3	PB4	PB5	PB6
VB0	1	1	0	1			
VB1		0	1	0	1		
VB2			0	0	0	1	
VB3				0	0	0	0

410

592

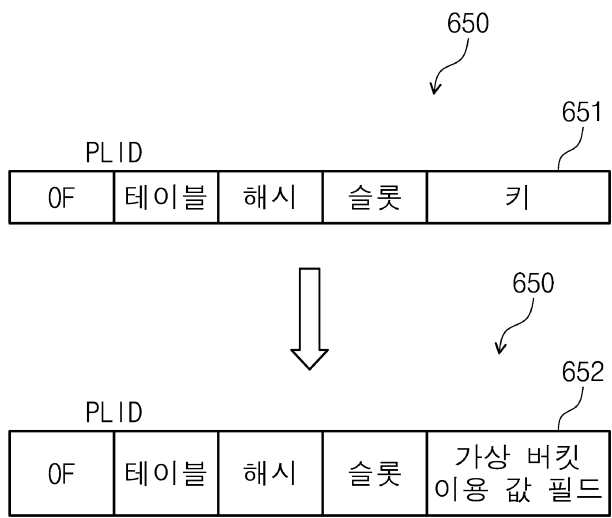
도면5c

	PB0	PB1	PB2	PB3	PB4	PB5	PB6
VB0	Y0	Y1		Y3			
VB1			Y2		Y4		
VB2						Y5	
VB3							

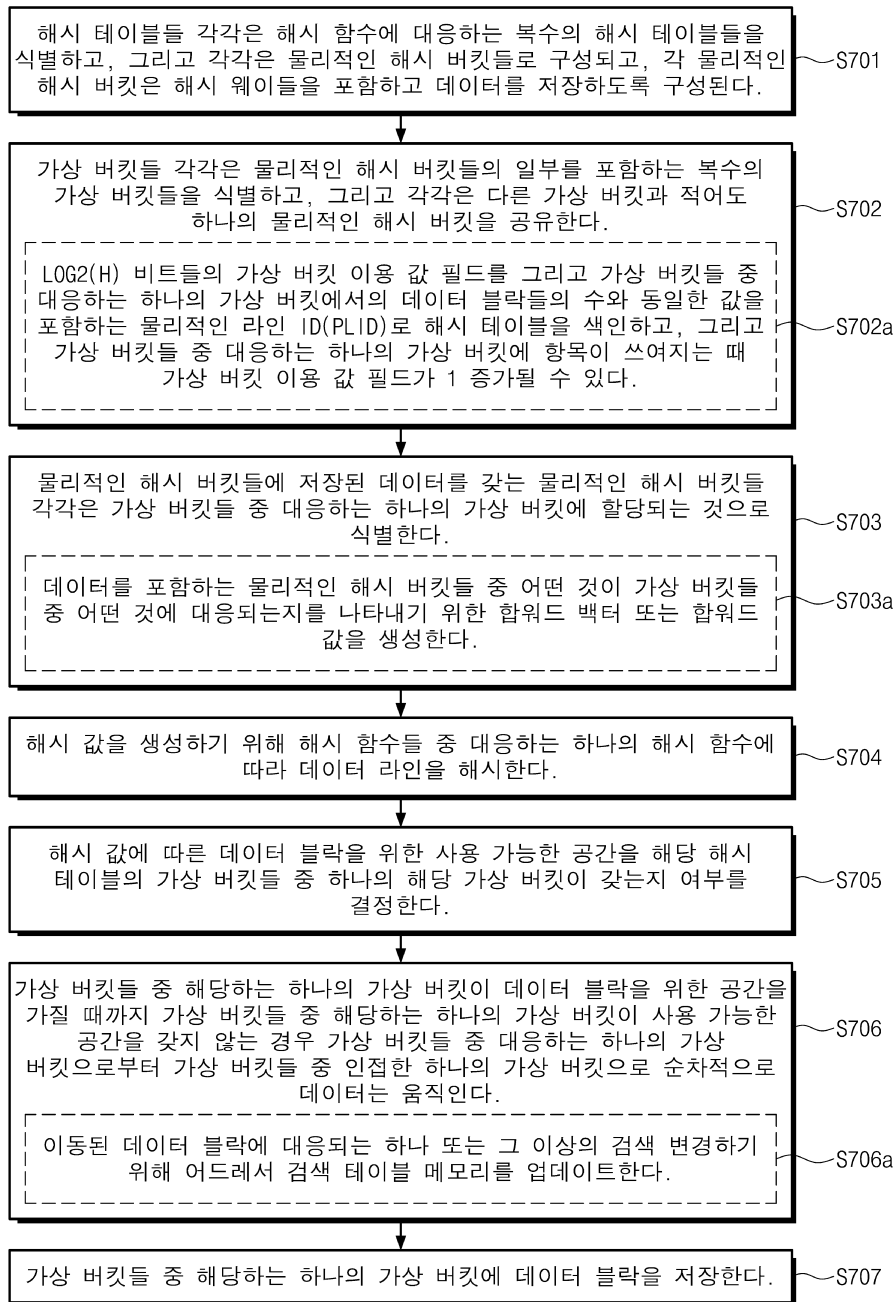
410

592

도면6



도면7



도면8

