



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0002581
(43) 공개일자 2020년01월08일

(51) 국제특허분류(Int. Cl.)
G06F 3/06 (2006.01) G11C 13/00 (2006.01)
(52) CPC특허분류
G06F 3/0611 (2013.01)
G06F 3/0644 (2013.01)
(21) 출원번호 10-2019-0001915(분할)
(22) 출원일자 2019년01월07일
심사청구일자 없음
(62) 원출원 특허 10-2018-0075930
원출원일자 2018년06월29일
심사청구일자 2018년07월06일

(71) 출원인
주식회사 멤레이
경기도 성남시 분당구 운중로 128, 303호(운중동, 마크시티그린)
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
정명수
인천광역시 연수구 송도과학로 85, 진리관C 304호 (송도동, 연세대학교)
박규영
인천광역시 연수구 송도과학로 85, 진리관C 304호 (송도동, 연세대학교)
(74) 대리인
유미특허법인

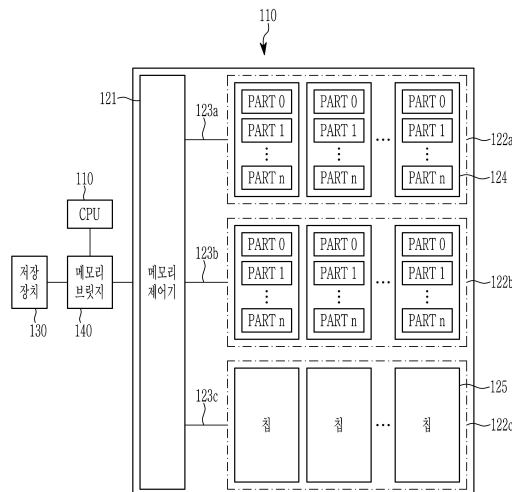
전체 청구항 수 : 총 22 항

(54) 발명의 명칭 메모리 제어 장치 및 이를 포함하는 메모리 시스템

(57) 요약

메모리 서브시스템과 메모리 제어기를 포함하는 메모리 시스템이 제공된다. 메모리 서브시스템은 상변화 메모리로 구현되는 복수의 제1 메모리 모듈과 상변화 메모리보다 쓰기 속도가 빠른 메모리로 구현되는 제2 메모리 모듈을 포함한다. 메모리 제어기는, 원 데이터가 나누어진 복수의 서브데이터로부터 논블록킹 부호를 생성하고, 논블록킹 부호를 상기 제2 메모리 모듈에 쓰고, 복수의 서브데이터를 각각 복수의 제1 메모리 모듈에 쓰며, 읽기 요청 시에 소정 조건 하에서 복수의 제1 메모리 모듈 중 일부 제1 메모리 모듈로부터 읽은 복수의 서브데이터 중 일부 서브데이터와 제2 메모리 모듈로부터 읽은 논블록킹 부호로부터 원 데이터를 재구성한다.

대표도 - 도1



(52) CPC특허분류

G06F 3/0647 (2013.01)

G06F 3/0658 (2013.01)

G11C 13/0021 (2013.01)

명세서

청구범위

청구항 1

제1 유형의 메모리로 구현되는 복수의 제1 메모리 모듈과 상기 제1 유형의 메모리보다 쓰기 속도가 빠른 제2 유형의 메모리로 구현되는 제2 메모리 모듈을 포함하는 메모리 서브시스템, 그리고

상기 복수의 제1 메모리 모듈과 상기 제2 메모리 모듈에 복수의 채널을 통해 연결되는 메모리 제어기를 포함하며,

상기 메모리 제어기는, 원 데이터가 나누어진 복수의 서브데이터로부터 논블록킹 부호를 생성하고, 상기 논블록킹 부호를 상기 제2 메모리 모듈에 쓰고, 상기 복수의 서브데이터를 각각 상기 복수의 제1 메모리 모듈에 쓰며, 읽기 요청 시에 소정 조건 하에서 상기 복수의 제1 메모리 모듈 중 일부 제1 메모리 모듈로부터 상기 서브 데이터를 읽지 않고 나머지 일부 제1 메모리 모듈로부터 읽은 상기 복수의 서브데이터 중 일부 서브데이터와 상기 제2 메모리 모듈로부터 읽은 상기 논블록킹 부호로부터 상기 원 데이터를 재구성하는

메모리 시스템.

청구항 2

제1항에서,

상기 메모리 제어기는, 중앙 처리 유닛(central processing unit, CPU)으로부터의 쓰기 요청에 따라 상기 쓰기 요청에 해당하는 데이터를 저장하는 캐시를 포함하는 메모리 시스템.

청구항 3

제2항에서,

상기 원 데이터가 상기 캐시에 저장되어 있는 경우, 상기 메모리 제어기는, 상기 캐시에 저장된 상기 원 데이터를 상기 메모리 서브시스템으로 마이그레이션할 때, 상기 논블록킹 부호를 상기 제2 메모리 모듈에 쓰고, 상기 복수의 서브데이터를 각각 상기 복수의 제1 메모리 모듈에 쓰는 메모리 시스템.

청구항 4

제3항에서,

상기 복수의 서브데이터는 제1 서브데이터와 제2 서브데이터를 포함하며,

상기 복수의 서브데이터를 각각 상기 복수의 제1 메모리 모듈에 쓸 때, 상기 메모리 제어기는 상기 제1 서브데이터를 상기 복수의 제1 메모리 모듈 중 대응하는 제1 메모리 모듈에 쓰고, 상기 제1 서브데이터의 쓰기 완료된 후에 상기 제2 서브데이터를 상기 복수의 제1 메모리 모듈 중 대응하는 제1 메모리 모듈에 쓰는

메모리 시스템.

청구항 5

제4항에서,

상기 복수의 서브데이터가 상기 복수의 제1 메모리 모듈로 쓰는 것인 완료된 후, 상기 메모리 제어기는 상기 원 데이터를 상기 캐시에서 제거하는 메모리 시스템.

청구항 6

제4항에서,

상기 제1 서브데이터를 상기 대응하는 제1 메모리 모듈에 쓸 때, 상기 메모리 제어기는 상기 논블록킹 부호를 상기 제2 메모리 모듈에 쓰는 메모리 시스템.

청구항 7

제4항에서,

상기 원 데이터를 상기 메모리 서브시스템으로 마이그레이션하는 동안 상기 원 데이터에 대해서 캐시 히트가 발생하는 경우, 상기 메모리 제어기는 상기 원 데이터의 마이그레이션을 중지하는 메모리 시스템.

청구항 8

제1항에서,

각 제1 메모리 모듈의 메모리 셀 어레이는 복수의 파티션으로 분할되어 있으며,

상기 소정 조건은 상기 복수의 서브데이터 중 다른 일부 서브데이터가 저장된 파티션에서 쓰기 동작이 진행 중인 조건을 포함하는

메모리 시스템.

청구항 9

제8항에서,

상기 메모리 제어기는, 중앙 처리 유닛(central processing unit, CPU)으로부터의 쓰기 요청에 따라 상기 쓰기 요청에 해당하는 데이터를 저장하는 캐시를 포함하며,

상기 소정 조건은 상기 원 데이터의 상기 읽기 요청에 대해서 캐시 미스가 발생하는 조건을 더 포함하는 메모리 시스템.

청구항 10

제2항에서,

상기 캐시는 비휘발성 메모리로 구현되는 메모리 시스템.

청구항 11

제1항에서,

상기 메모리 시스템은 컴퓨팅 디바이스의 CPU에 의해 사용되는 메인 메모리인, 메모리 시스템.

청구항 12

제1항에서,

상기 제1 유형의 메모리는 상변화 메모리인 메모리 시스템.

청구항 13

제1 유형의 메모리로 구현되는 제1 메모리 모듈 및 제2 메모리 모듈, 그리고 상기 제1 유형의 메모리보다 쓰기 속도가 빠른 제2 유형의 메모리로 구현되는 제3 메모리 모듈을 포함하는 메모리 서브시스템에 복수의 채널을 통해 연결되는 메모리 제어 장치로서,

중앙 처리 유닛(central processing unit, CPU)로부터의 쓰기 요청에 따라 상기 쓰기 요청에 해당하는 데이터를 저장하는 캐시, 그리고

상기 캐시에 저장된 원 데이터를 제1 서브데이터와 제2 서브데이터를 포함하는 복수의 서브데이터로 나누고, 상기 복수의 서브데이터로부터 논블록킹 부호를 생성하며, 마이그레이션 시에 상기 제1 서브데이터를 상기 제1 메모리 모듈에 쓰고, 상기 제2 서브데이터를 상기 제2 메모리 모듈에 쓰고, 상기 논블록킹 부호를 상기 제3 메모리 모듈에 쓰고, 상기 CPU로부터의 읽기 요청 시에 소정 조건 하에서 상기 제2 메모리 모듈로부터 상기 제2 서브데이터를 읽지 않고 상기 제1 메모리 모듈로부터 읽은 상기 제1 서브데이터와 상기 제3 메모리 모듈로부터 읽은 상기 논블록킹 부호로부터 상기 원 데이터를 재구성하는 메모리 제어기

를 포함하는 메모리 제어 장치.

청구항 14

제13항에서,

상기 메모리 제어기는 상기 제1 및 제2 서브데이터 중 어느 하나의 서브데이터를 상기 제1 및 제2 메모리 모듈 중 대응하는 메모리 모듈에 쓰고, 상기 어느 하나의 서브데이터의 쓰기 완료된 후에 상기 제1 및 제2 서브데이터 중 다른 하나의 서브데이터를 상기 제1 및 제2 메모리 모듈 중 대응하는 메모리 모듈에 쓰는

메모리 제어 장치.

청구항 15

제14항에서,

상기 제1 및 제2 서브데이터가 상기 제1 및 제2 메모리 모듈로 쓰는 것인 완료된 후, 상기 메모리 제어기는 상기 원 데이터를 상기 캐시에서 제거하는 메모리 제어 장치.

청구항 16

제14항에서,

상기 어느 하나의 서브데이터를 상기 대응하는 메모리 모듈에 쓸 때, 상기 메모리 제어기는 상기 논블록킹 부호를 상기 제3 메모리 모듈에 쓰는 메모리 제어 장치.

청구항 17

제14항에서,

상기 마이그레이션 동안 상기 원 데이터에 대해서 캐시 히트가 발생하는 경우, 상기 메모리 제어기는 상기 원 데이터의 마이그레이션을 중지하는 메모리 제어 장치.

청구항 18

제13항에서,

상기 제2 메모리 모듈의 메모리 셀 어레이는 복수의 파티션으로 분할되어 있으며,

상기 소정 조건은 상기 제2 서브데이터가 저장된 파티션에서 쓰기 동작이 진행 중인 조건을 포함하는

메모리 제어 장치.

청구항 19

제18항에서,

상기 소정 조건은 상기 원 데이터의 상기 읽기 요청에 대해서 캐시 미스가 발생하는 조건을 더 포함하는 메모리 제어 장치.

청구항 20

제13항에서,

상기 캐시는 비휘발성 메모리로 구현되는 메모리 제어 장치.

청구항 21

제13항에서,

상기 제1 유형의 메모리는 상변화 메모리인 메모리 제어 장치.

청구항 22

제13항 내지 제20항 중 어느 한 항에 기재된 상기 메모리 제어 장치, 그리고

상기 메모리 서브시스템

을 포함하는 메모리 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 메모리 제어 장치 및 이를 포함하는 메모리 시스템에 관한 것이다.

배경 기술

[0002] 반도체 메모리의 고성능화 및 저전력화 추세에 맞추어 차세대 반도체 메모리가 개발되고 있다. 이러한 차세대 반도체 메모리 중에서 상변화 물질(phase-change material)을 이용하는 상변화 메모리(phase-change memory, PCM), 특히 상변화 랜덤 액세스 메모리(phase-change random access memory, PRAM)가 있다. 상변화 메모리는 결정질 상태(crystalline state)와 비정질 상태(amorphous state)를 전환하는 상변화 물질을 사용하고, 결정질 상태와 비정질 상태의 저항을 차이에 기초하여서 데이터를 저장한다.

[0003] 이러한 상변화 메모리에서 현재 상태의 상변화 물질을 통해 데이터를 읽는 경우보다 상변화 물질의 상태를 변경할 필요가 있는 데이터 쓰기에 시간이 많이 걸린다는 문제점이 있다.

발명의 내용

해결하려는 과제

[0004] 본 발명이 이루고자 하는 과제는 응답 속도를 증가시킬 수 있는 상변화 메모리를 사용하는 메모리 시스템 및 메모리 제어 장치를 제공하는 것이다.

과제의 해결 수단

[0005] 본 발명의 한 실시예에 따르면, 상변화 메모리로 구현되는 복수의 제1 메모리 모듈과 상변화 메모리보다 쓰기 속도가 빠른 메모리로 구현되는 제2 메모리 모듈을 포함하는 메모리 서브시스템, 그리고 상기 복수의 제1 메모리 모듈과 상기 제2 메모리 모듈에 복수의 채널을 통해 연결되는 메모리 제어기를 포함하는 메모리 시스템이 제공된다. 상기 메모리 제어기는, 원 데이터가 나누어진 복수의 서브데이터로부터 논블록킹 부호를 생성하고, 상기 논블록킹 부호를 상기 제2 메모리 모듈에 쓰고, 상기 복수의 서브데이터를 각각 상기 복수의 제1 메모리 모듈에 쓰며, 읽기 요청 시에 소정 조건 하에서 상기 복수의 제1 메모리 모듈 중 일부 제1 메모리 모듈로부터 읽은 상기 복수의 서브데이터 중 일부 서브데이터와 상기 제2 메모리 모듈로부터 읽은 상기 논블록킹 부호로부터 상기 원 데이터를 재구성한다.

[0006] 상기 메모리 제어기는, 중앙 처리 유닛(central processing unit, CPU)으로부터의 쓰기 요청에 따라 상기 쓰기 요청에 해당하는 데이터를 저장하는 캐시를 포함할 수 있다.

[0007] 상기 원 데이터가 상기 캐시에 저장되어 있는 경우, 상기 메모리 제어기는, 상기 캐시에 저장된 상기 원 데이터를 상기 메모리 서브시스템으로 마이그레이션할 때, 상기 논블록킹 부호를 상기 제2 메모리 모듈에 쓰고, 상기 복수의 서브데이터를 각각 상기 복수의 제1 메모리 모듈에 쓸 수 있다.

[0008] 상기 복수의 서브데이터는 제1 서브데이터와 제2 서브데이터를 포함할 수 있다. 이때, 상기 복수의 서브데이터를 각각 상기 복수의 제1 메모리 모듈에 쓸 때, 상기 메모리 제어기는 상기 제1 서브데이터를 상기 복수의 제1 메모리 모듈 중 대응하는 제1 메모리 모듈에 쓰고, 상기 제1 서브데이터의 쓰기 완료된 후에 상기 제2 서브데이터를 상기 복수의 제1 메모리 모듈 중 대응하는 제1 메모리 모듈에 쓸 수 있다.

[0009] 상기 복수의 서브데이터가 상기 복수의 제1 메모리 모듈로 쓰는 것인 완료된 후, 상기 메모리 제어기는 상기 원 데이터를 상기 캐시에서 제거할 수 있다.

[0010] 상기 제1 서브데이터를 상기 대응하는 제1 메모리 모듈에 쓸 때, 상기 메모리 제어기는 상기 논블록킹 부호를 상기 제2 메모리 모듈에 쓸 수 있다.

[0011] 상기 원 데이터를 상기 메모리 서브시스템으로 마이그레이션하는 동안 상기 원 데이터에 대해서 캐시 히트가 발생하는 경우, 상기 메모리 제어기는 상기 원 데이터의 마이그레이션을 중지할 수 있다.

[0012] 각 제1 메모리 모듈의 메모리 셀 어레이는 복수의 파티션으로 분할되어 있을 수 있다. 이때, 상기 소정 조건은

상기 복수의 서브데이터 중 다른 일부 서브데이터가 저장된 파티션에서 쓰기 동작이 진행 중인 조건을 포함할 수 있다.

[0013] 상기 소정 조건은 상기 원 데이터의 상기 읽기 요청에 대해서 캐시 미스가 발생하는 조건을 더 포함할 수 있다.

[0014] 상기 캐시는 비휘발성 메모리로 구현될 수 있다.

[0015] 상기 메모리 시스템은 컴퓨팅 디바이스의 CPU에 의해 사용되는 메인 메모리일 수 있다.

[0016] 본 발명의 다른 실시예에 따르면, 상변화 메모리로 구현되는 제1 메모리 모듈 및 제2 메모리 모듈, 그리고 상변화 메모리보다 쓰기 속도가 빠른 메모리로 구현되는 제3 메모리 모듈을 포함하는 메모리 서브시스템에 복수의 채널을 통해 연결되는 메모리 제어 장치가 제공된다. 상기 메모리 제어 장치는 캐시 및 메모리 제어기를 포함한다. 상기 캐시는 CPU로부터의 쓰기 요청에 따라 상기 쓰기 요청에 해당하는 데이터를 저장한다. 상기 메모리 제어기는, 상기 캐시에 저장된 원 데이터를 제1 서브데이터와 제2 서브데이터를 포함하는 복수의 서브데이터로 나누고, 상기 복수의 서브데이터로부터 논블록킹 부호를 생성하며, 마이그레이션 시에 상기 제1 서브데이터를 상기 제1 메모리 모듈에 쓰고, 상기 제2 서브데이터를 상기 제2 메모리 모듈에 쓰고, 상기 논블록킹 부호를 상기 제3 메모리 모듈에 쓰고, 상기 CPU로부터의 읽기 요청 시에 소정 조건 하에서 상기 제2 메모리 모듈로부터 상기 제2 서브데이터를 읽지 않고 상기 제1 메모리 모듈로부터 읽은 상기 제1 서브데이터와 상기 제3 메모리 모듈로부터 읽은 상기 논블록킹 부호로부터 상기 원 데이터를 재구성한다.

[0017] 상기 메모리 제어기는 상기 제1 및 제2 서브데이터 중 어느 하나의 서브데이터를 상기 제1 및 제2 메모리 모듈 중 대응하는 메모리 모듈에 쓰고, 상기 어느 하나의 서브데이터의 쓰기 완료된 후에 상기 제1 및 제2 서브데이터 중 다른 하나의 서브데이터를 상기 제1 및 제2 메모리 모듈 중 대응하는 메모리 모듈에 쓸 수 있다.

[0018] 상기 제1 및 제2 서브데이터가 상기 제1 및 제2 메모리 모듈로 쓰는 것인 완료된 후, 상기 메모리 제어기는 상기 원 데이터를 상기 캐시에서 제거할 수 있다.

[0019] 상기 어느 하나의 서브데이터를 상기 대응하는 메모리 모듈에 쓸 때, 상기 메모리 제어기는 상기 논블록킹 부호를 상기 제3 메모리 모듈에 쓸 수 있다.

[0020] 상기 마이그레이션 동안 상기 원 데이터에 대해서 캐시 히트가 발생하는 경우, 상기 메모리 제어기는 상기 원 데이터의 마이그레이션을 중지할 수 있다.

[0021] 상기 제2 메모리 모듈의 메모리 셀 어레이는 복수의 파티션으로 분할되어 있을 수 있다. 이때, 상기 소정 조건은 상기 제2 서브데이터가 저장된 파티션에서 쓰기 동작이 진행 중인 조건을 포함할 수 있다.

[0022] 상기 소정 조건은 상기 원 데이터의 상기 읽기 요청에 대해서 캐시 미스가 발생하는 조건을 더 포함할 수 있다.

[0023] 상기 캐시는 비휘발성 메모리로 구현될 수 있다.

[0024] 본 발명의 또 다른 실시예에 따르면, 상기 메모리 제어 장치 및 상기 메모리 서브시스템을 포함하는 메모리 시스템이 제공된다.

발명의 효과

[0025] 본 발명의 한 실시예에 따르면, 상변화 메모리에서의 긴 쓰기 지연을 감출 수 있으며, 또한 읽기 요청의 차단 없이 읽기 요청의 데이터가 제공될 수 있다. 다른 실시예에 따르면, 로그 데이터를 저장하는 로깅 메커니즘 없이 메모리 연속성을 제공할 수 있다.

도면의 간단한 설명

[0026] 도 1은 본 발명의 한 실시예에 따른 컴퓨팅 디바이스의 개략적인 블록도이다.

도 2는 PRAM에서 하나의 메모리 셀을 개략적으로 나타내는 도면이다.

도 3은 도 2에 도시한 메모리 셀에 인가되는 전류를 나타내는 도면이다.

도 4는 도 2에 도시한 메모리 셀에 도 3에 도시한 전류가 인가될 때의 온도 변화를 나타내는 도면이다.

도 5는 본 발명의 한 실시예에 따른 PRAM 기반 가속기에서 PRAM 모듈의 개략적인 블록도이다.

도 6은 본 발명의 한 실시예에 따른 PRAM에서 메모리 셀 어레이의 파티션 방법의 한 예를 설명하는 도면이다.

도 7은 본 발명의 한 실시예에 따른 PRAM에서 파티션의 한 예를 설명하는 도면이다.

도 8은 본 발명의 한 실시예에 따른 메모리 시스템의 개략적인 블록도이다.

도 9는 본 발명의 다른 실시예에 따른 메모리 시스템의 개략적인 블록도이다.

도 10은 본 발명의 한 실시예에 따른 메모리 제어기에서 캐시 제어 모듈의 동작을 나타내는 흐름도이다.

도 11은 본 발명의 한 실시예에 따른 메모리 제어기에서 마이그레이션 모듈의 동작을 나타내는 흐름도이다.

도 12는 메모리 시스템에서의 로깅 메커니즘을 설명하는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0027] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.
- [0028] 도 1은 본 발명의 한 실시예에 따른 컴퓨팅 디바이스의 개략적인 블록도이다. 도 1은 가능한 컴퓨팅 디바이스의 한 예이며, 본 발명의 실시예에 따른 컴퓨팅 디바이스는 다른 다양한 구조로 구현될 수 있다.
- [0029] 도 1을 참고하면, 본 발명의 한 실시예에 따른 컴퓨팅 디바이스(100)는 중앙 처리 유닛(central processing unit, CPU)(110) 및 메모리(120)를 포함한다.
- [0030] 메모리(120)는 CPU(100)에 의해 액세스되어 사용된다. 어떤 실시예에서, 메모리(120)는 컴퓨팅 디바이스의 메인 메모리일 수 있다. 이 경우, 메모리(120)는 바이트 단위로 어드레스가 가능한(byte-addressable) 메모리일 수 있다.
- [0031] 어떤 실시예에서, 컴퓨팅 디바이스(100)는 별도의 저장 장치(130)를 더 포함할 수 있다.
- [0032] 어떤 실시예에서, 컴퓨팅 디바이스(100)는 메모리(120)와 저장 장치(130)를 CPU(110)와 연결하기 위해서 메모리 브릿지(140)를 더 포함할 수 있다. 메모리 브릿지(140)는 예를 들면 노스브릿지(northbridge) 또는 메모리 컨트롤러 허브(memory controller hub, MCH)일 수 있다.
- [0033] 다시 도 1을 참고하면, 메모리(120)로 사용되는 메모리 시스템(120)은 메모리 제어기(121) 및 메모리 서브시스템(122)을 포함하며, 메모리 서브시스템(122)은 메모리 제어기(121)에 복수의 채널(123a, 123b, 123c)을 통해 연결되어 있다. 메모리 제어기(121)는 CPU(110)로부터 요청, 예를 들면 입출력(input/output, I/O) 요청을 수신하고, 수신한 I/O 요청에 기초하여 메모리 서브시스템(122)에 액세스한다.
- [0034] 메모리 서브시스템(122)은 복수의 상변화 메모리(phase-change memory, PCM) 모듈(122a, 122b) 및 논블록킹(non-blocking) 메모리 모듈(122c)을 포함한다. 아래에서는 상변화 랜덤 액세스 메모리(phase-change random access memory, PRAM)를 PCM의 한 예로 설명한다. 또한 논블록킹 메모리 모듈(122c)로는 PRAM보다 쓰기 속도가 빠른 메모리가 사용되며, 아래에서는 논블록킹 메모리 모듈(122c)로 동적 랜덤 액세스 메모리(dynamic random access memory, DRAM) 모듈이 사용되는 것으로 설명한다.
- [0035] PRAM 모듈(122a, 122b)은 데이터를 저장하는 공간으로 제공되며, 각 PRAM 모듈(122a, 122b)은 대응하는 채널(123a, 123b), 즉 PRAM 채널(123a, 123b)을 통해 메모리 제어기(121)에 연결되어 있다. DRAM 모듈(122c)은 논블록킹 부호(non-blocking code)를 저장하는 공간으로 제공되며, DRAM 모듈(122c)도 대응하는 채널(123c), 즉 DRAM 채널(123c)을 통해 메모리 제어기(121)에 연결되어 있다. 논블록킹 부호는 원 데이터에서 일부 데이터가 빠지더라도 나머지 데이터와 함께 원 데이터를 재구성하는데 사용될 수 있는 부호이다.
- [0036] 메모리 제어기(121)는 I/O 요청에 따라 메모리 서브시스템(122)에 쓸 쓰기 데이터를 복수의 서브데이터(sub-data)로 나누고, 복수의 서브데이터에 기초해서 논블록킹 부호를 생성한다. 메모리 제어기(121)는 복수의 서브데이터를 복수의 PRAM 채널(123a, 123b)을 통해 복수의 PRAM 모듈(122a, 122b)로 쓰고, 논블록킹 부호를 DRAM 채널(123c)을 통해 DRAM 모듈(122c)로 쓴다. 또한 쓰기 데이터가 나누어져 기록 복수의 PRAM 모듈(122a, 122b) 중 일부 PRAM 모듈에서 읽기 동작이 불가능한 경우, 메모리 제어기(110)는 복수의 PRAM 모듈(122a, 122b) 중 읽기 동작이 가능한 일부 PRAM 모듈로부터 서브데이터를 읽고, DRAM 모듈(122c)로부터 논블록킹 부호를 읽은 후,

읽은 서브데이터와 논블록킹 부호로부터 원래의 쓰기 데이터를 재구성할 수 있다.

- [0037] 도 1에 도시한 것처럼, 두 개의 PRAM 모듈(122a, 122b)이 사용되는 경우, 메모리 제어기(121)는 쓰기 데이터를 두 개의 서브데이터로 나눌 수 있다. 예를 들면, 쓰기 데이터가 64 바이트 데이터인 경우, 메모리 제어기(121)는 쓰기 데이터를 두 개의 32 바이트 서브데이터로 나누어서 두 개의 PRAM 모듈(122a, 122b)에 각각 쓸 수 있다. 메모리 제어기(121)는 두 개의 서브데이터를 부호화하여 논블록킹 부호를 생성해서 DRAM 모듈(122c)에 쓸 수 있다. 또한 메모리 제어기(121)는 두 개의 PRAM 모듈(122a, 122b) 중 하나의 PRAM 모듈로부터 서브데이터를 읽고 DRAM 모듈(122c)에 논블록킹 부호를 읽어서, 읽은 서브데이터와 논블록킹 부호를 합쳐서 원래의 쓰기 데이터를 재구성할 수 있다.
- [0038] 도 1에는 두 개의 PRAM 모듈(122a, 122b)과 이에 대응하는 두 개의 PRAM 채널(123a, 123b)이 사용되는 예를 도시하였지만, PRAM 모듈의 개수는 두 개에 한정되지 않는다. 또한 위에서는 원 데이터를 두 개의 서브데이터로 나누는 예를 도 1을 참고로 하여 설명하였지만, 원 데이터가 나누어지는 서브데이터의 개수는 두 개에 한정되지 않고, 원 데이터는 두 개 이상의 서브데이터를 나누어질 수 있다.
- [0039] 예를 들면, 원 데이터는 네 개의 서브데이터(데이터 A, 데이터 B, 데이터 C, 데이터 D)로 나누어질 수 있다. 이 경우, 메모리 서브시스템(122)은 적어도 네 개의 PRAM 모듈(PRAM 모듈 A, PRAM 모듈 B, PRAM 모듈 C, PRAM 모듈 D)를 포함할 수 있으며, 데이터 A, 데이터 B, 데이터 C 및 데이터 D는 각각 PRAM 모듈 A, PRAM 모듈 B, PRAM 모듈 C 및 PRAM 모듈 D에 저장될 수 있다.
- [0040] 한 예에서, 메모리 제어기(121)는 데이터 A, 데이터 B, 데이터 C 및 데이터 D를 부호화하여서 하나의 논블록킹 부호를 생성하여 DRAM 모듈에 저장할 수 있다. 이 경우, 메모리 제어기(121)는 하나의 PRAM 모듈(예를 들면, PRAM 모듈 A)에서 데이터 A를 읽지 못하더라도, 나머지 세 개의 PRAM 모듈(예를 들면, PRAM 모듈 B, PRAM 모듈 C, PRAM 모듈 D)에서 읽은 세 개의 데이터(데이터 B, 데이터 C, 데이터 D)와 DRAM 모듈에서 읽은 논블록킹 부호로부터 원 데이터를 재구성할 수 있다.
- [0041] 다른 예에서, 메모리 제어기(121)는 데이터 A와 데이터 B를 결합한 데이터와 데이터 C와 데이터 D를 결합한 데이터를 부호화하여서 하나의 논블록킹 부호를 생성하여 DRAM 모듈에 저장할 수 있다. 이 경우, 메모리 제어기(121)는 두 PRAM 모듈(예를 들면, PRAM 모듈 A, PRAM 모듈 B)에서 데이터 A 및 데이터 B를 읽지 못하더라도, 나머지 두 PRAM 모듈(예를 들면, PRAM 모듈 C, PRAM 모듈 D)에서 읽은 두 데이터(데이터 C, 데이터 D)와 DRAM 모듈에서 읽은 논블록킹 부호로부터 원 데이터를 재구성할 수 있다.
- [0042] 또 다른 예에서, 메모리 제어기(121)는 데이터 A와 데이터 B를 부호화하여서 하나의 논블록킹 부호를 생성하여 DRAM 모듈에 저장하고, 데이터 C와 데이터 D를 부호화하여서 다른 하나의 논블록킹 부호를 생성하여 DRAM 모듈에 저장할 수 있다. 이 경우, 두 논블록킹 부호는 서로 다른 DRAM 모듈에 저장될 수 있다. 이 경우, 메모리 제어기(121)는 두 PRAM 모듈(예를 들면, PRAM 모듈 A, PRAM 모듈 C)에서 데이터 A 및 데이터 C를 읽지 못하더라도, 나머지 두 PRAM 모듈(예를 들면, PRAM 모듈 B, PRAM 모듈 D)에서 읽은 두 데이터(데이터 B, 데이터 D)와 DRAM 모듈에서 읽은 두 논블록킹 부호로부터 원 데이터를 재구성할 수 있다.
- [0043] 다시 도 1을 참고하면, 하나의 채널에 연결된 PRAM 모듈(122a 또는 122b)은 복수의 PRAM 칩(또는 PRAM 뱅크)(124)을 포함한다. 어떤 실시예에서, 각 PRAM 칩(또는 PRAM 뱅크)는 복수의 파티션(PART0-PARTn)으로 분할될 수 있다. 하나의 채널에 연결된 PRAM 모듈(122a 또는 122b)은 하나의 랭크(rank)로 이루어질 수 있으며, 또는 복수의 랭크로 확장될 수도 있다. 어떤 실시예에서, 하나의 채널에 연결된 복수의 랭크가 서로 독립적으로 동작할 수 있다면, 각 랭크가 독립적인 PRAM 모듈로 동작할 수 있다. 이 경우, 복수의 PRAM 모듈(즉, 복수의 랭크)이 하나의 채널을 공유할 수 있다. 하나의 채널에 연결된 DRAM 모듈(122c)은 복수의 DRAM 칩(또는 DRAM 뱅크)을 포함한다. 하나의 채널에 연결된 DRAM 모듈(122c)은 하나의 랭크(rank)로 이루어질 수 있으며, 또는 복수의 랭크로 확장될 수도 있다.
- [0044] 다음, 본 발명의 한 실시예에 따른 메모리 시스템(100)에 포함되는 PRAM(120)의 한 예에 대해서 설명한다.
- [0045] 도 2는 PRAM에서 하나의 메모리 셀을 개략적으로 나타내는 도면이며, 도 3은 도 2에 도시한 메모리 셀에 인가되는 전류를 나타내는 도면이고, 도 4는 도 2에 도시한 메모리 셀에 도 3에 도시한 전류가 인가될 때의 온도 변화를 나타내는 도면이다.
- [0046] 도 2에 도시한 메모리 셀은 하나의 예이며, 본 발명의 실시예에 따른 PRAM의 메모리 셀은 다양한 형태로 구현될 수 있다.

- [0047] 도 2를 참고하면, PRAM의 메모리 셀(200)은 상변화 소자(210)와 스위칭 소자(220)를 포함한다. 스위칭 소자(220)는 모스 트랜지스터, 다이오드 등의 다양한 소자로 구현될 수 있다. 상변화 소자(210)는 상변화막(211), 상변화막 위에 형성되어 있는 상부 전극(212) 및 상변화막(211) 아래에 형성되어 있는 하부 전극(213)을 포함한다. 예를 들면, 상변화막(210)은 상변화 물질로 게르마늄(germanium, Ge), 안티몬(antimony, Sb) 및 텔루르(tellurium, Te)의 혼합물("GST 물질"이라 불리기도 함)을 포함할 수 있다.
- [0048] 상변화 물질은 비교적 높은 저항율(resistivity)을 가지는 비정질 상태와 비교적 낮은 저항율을 가지는 결정질 상태를 전환할 수 있다. 이때, 상변화 물질의 상태는 가열되는 온도와 가열 시간에 의해 결정될 수 있다.
- [0049] 다시 도 2를 참고하면, 메모리 셀(200)에 전류가 인가되면, 인가된 전류는 하부 전극(213)을 통해 흐르게 된다. 짧은 시간 동안 전류가 메모리 셀(200)에 인가되면, 인가된 전류는 하부 전극(213)의 인접한 막을 가열한다. 이때, 가열 프로파일의 차이에 의해서 상변화막(211)의 일부(도 2에서 해칭된 부분)가 결정질 상태로 되거나 비정질 상태로 된다. 결정질 상태는 "셋(set) 상태"로, 비정질 상태로 "리셋(reset) 상태"로 불린다.
- [0050] 도 3 및 도 4를 참고하면, 높은 전류의 리셋 펄스(RESET)가 짧은 시간(t_{RST}) 동안 메모리 셀(200)에 인가될 때, 상변화막(211)은 리셋 상태로 된다. 리셋 펄스(RESET)의 인가에 따라 상변화막(211)의 상변화 물질이 가열되어 온도(T_r)가 녹는점 이상으로 되면, 상변화 물질이 녹은 후 식으면서 비정질 상태로 변화한다. 리셋 펄스(RESET)보다 낮은 전류를 가지는 셋 펄스(SET)가 리셋 펄스(RESET)보다 긴 시간(t_{SET}) 동안 상변화막(211)에 인가될 때, 상변화막(211)은 셋 상태로 된다. 셋 전류(SET)의 인가에 따라, 상변화 물질이 가열되어서 온도(T_s) 녹는점보다 낮은 결정화 온도에 도달하면 결정질 상태로 변화한다. 셋 펄스(SET)보다 낮은 전류가 인가되거나 짧은 시간 동안 전류가 인가되는 경우에는 이러한 리셋 상태 및 셋 상태가 유지되므로, 메모리 셀(200)에 데이터를 기입할 수 있다.
- [0051] 이때, 리셋 상태와 셋 상태는 각각 "1"과 "0"의 데이터로 설정될 수 있으며, 이는 메모리 셀(200)의 상변화 소자(210)의 저항율을 측정함으로써 감지될 수 있다. 이와는 달리, 리셋 상태와 셋 상태를 각각 "0"과 "1"의 데이터로 설정될 수도 있다.
- [0052] 따라서 메모리 셀(200)에 읽기 펄스(READ)를 인가함으로써 메모리 셀(200)에 저장된 데이터를 읽을 수 있다. 읽기 펄스(READ)는 낮은 전류로 짧은 시간(t_{READ}) 동안 인가되어 메모리 셀(200)의 상태를 변경하지 않을 수 있다. 읽기 펄스(READ)는 전류 크기가 셋 펄스(SET)보다 낮을 수 있고, 인가되는 시간(t_{READ})이 리셋 펄스(RESET)의 인가 시간(t_{RST})보다 짧을 수 있다. 메모리 셀(200)의 상변화 소자(210)의 저항율이 상태에 따라 다르므로, 상변화 소자(210)에 흐르는 전류의 크기 또는 상변화 소자(210)에서의 전압 강하의 크기로 메모리 셀(200)의 상태, 즉 메모리 셀(200)에 저장된 데이터를 읽을 수 있다.
- [0053] 한 실시예에서, 읽기 펄스(READ)를 인가하였을 때, 메모리 셀(200)에 걸리는 전압의 크기의 차이로 메모리 셀(200)의 상태를 읽을 수 있다. 이 경우, 리셋 상태에서 메모리 셀(200)의 상변화 소자(210)는 큰 저항을 가지므로, 상변화 소자(210)에서 감지된 전압인 큰 경우를 리셋 상태로, 상변화 소자(210)에서 감지된 전압이 작은 경우를 셋 상태로 결정할 수 있다. 다른 실시예에서, 메모리 셀(200)에 전압을 인가하였을 때, 출력되는 전류의 차이로 메모리 셀(200)의 상태를 읽을 수 있다. 이 경우, 상변화 소자(210)에서 감지된 전류가 작은 경우를 리셋 상태로, 상변화 소자(210)에서 감지된 전류가 큰 경우를 셋 상태로 결정할 수 있다.
- [0054] PRAM에서는 데이터 쓰기는 일련의 리셋 과정과 셋 과정을 통해서 수행되므로, 인가 시간이 긴 리셋 펄스로 인해 쓰기 동작이 읽기 동작보다 느리다.
- [0055] 도 5는 본 발명의 한 실시예에 따른 메모리 시스템에서 PRAM의 개략적인 블록도이다. 도 5에 도시한 PRAM은 하나의 PRAM 칩 또는 하나의 PRAM 뱅크일 수 있다.
- [0056] 도 5를 참고하면, PRAM(500)은 메모리 셀 어레이(510), 행 어드레스 버퍼(520), 행 데이터 버퍼(530), 행 디코더(540), 센스 앰프(sense amplifier)(550) 및 기입 드라이버(560)를 포함한다.
- [0057] 메모리 셀 어레이(510)는 대략 행 방향으로 뻗어 있는 복수의 워드 라인(도시하지 않음), 대략 열 방향으로 뻗어 있는 복수의 비트 라인(도시하지 않음), 그리고 이에 연결되어 있으며 대략 행렬의 형태로 배열된 복수의 메모리 셀(도시하지 않음)을 포함한다. 메모리 셀은 예를 들면 도 2를 참고로 하여 설명한 메모리 셀(200)일 수 있다.
- [0058] 행 어드레스 버퍼(520)와 행 데이터 버퍼(530)는 행 버퍼를 형성한다. 각 행 버퍼는 행 어드레스 버퍼(520)와 행 데이터 버퍼(530)에 의해 논리적으로 짝이 지워져 있으며, 버퍼 어드레스에 의해 선택된다.

- [0059] 행 어드레스 버퍼(520)는 메모리 컨트롤러(도시하지 않음)로부터 전달되는 명령 데이터와 어드레스(특히, 행 어드레스)를 저장한다. 행 데이터 버퍼(530)는 메모리 셀 어레이(510)에서 전달된 데이터를 저장한다.
- [0060] 어떤 실시예에서, PRAM(500)은 복수의 행 버퍼(520, 530)를 사용하기 위해서 비휘발성 메모리 인터페이스를 채용할 수 있다. 한 실시예에서, 비휘발성 메모리 인터페이스는 DDR(double data rate) 인터페이스, 예를 들면 LPDDR2(low-power double data rate 2)-NVM(non-volatile memory) 인터페이스일 수 있다. 행 어드레스 버퍼(520)는 비휘발성 메모리 인터페이스를 통해 행 어드레스 및 뱅크 어드레스를 수신하고, 행 데이터 버퍼(530)는 비휘발성 메모리 인터페이스를 통해 데이터를 출력한다.
- [0061] 행 디코더(540)는 행 어드레스를 복호하여서 메모리 셀 어레이(510)의 복수의 행에서 타깃 행을 선택한다. 즉, 행 디코더(540)는 메모리 셀 어레이(510)의 복수의 워드 라인 중 데이터를 읽거나 데이터를 쓸 워드 라인을 선택한다.
- [0062] 어떤 실시예에서, 메모리 컨트롤러에서 전달되는 행 어드레스는 상위 어드레스(upper address)와 하위 어드레스(lower address)를 포함할 수 있다. 이 경우, 상위 어드레스는 행 어드레스 버퍼(520)로 전달되고, 하위 어드레스는 행 디코더(540)로 직접 전달될 수 있다. 이때, 행 디코더(540)는 행 어드레스 버퍼(520)에 저장된 상위 어드레스와 직접 전달된 하위 어드레스를 결합하여서 타깃 행을 선택할 수 있다.
- [0063] 센스 앰프(550)는 메모리 셀 어레이(510)에 저장된 데이터를 읽는 동작을 수행한다. 센스 앰프(550)는 행 디코더(540)가 선택한 워드 라인에 연결된 복수의 메모리 셀로부터 복수의 비트 라인을 통해서 데이터를 읽을 수 있다. 기입 드라이버(560)는 입력되는 데이터를 메모리 셀 어레이(510)에 쓴다. 기입 드라이버(560)는 행 디코더(540)가 선택한 워드 라인에 연결된 복수의 메모리 셀에 복수의 비트 라인을 통해서 데이터를 쓸 수 있다.
- [0064] 어떤 실시예에서, PRAM(500)에서는 쓰기 동작이 읽기 동작보다 느리므로, 이를 해결하기 위해서 PRAM(500)은 입력되는 데이터를 버퍼에 먼저 저장한 후에, 이를 메모리 셀 어레이(510)에 쓸 수 있다. 이를 위해, PRAM(500)은 메모리 매핑된(memory-mapped) 레지스터로서 오버레이 윈도우(570, 580)를 포함할 수 있다. 오버레이 윈도우는 오버레이 윈도우 레지스터(570)와 프로그램 버퍼(580)를 포함할 수 있다. 한 실시예에서, 쓰기 데이터에 대한 정보(예를 들면, 첫 번째 데이터 어드레스와 기입될 바이트 수)가 레지스터(570)에 써진 후에, 쓰기 데이터가 프로그램 버퍼(580)에 저장된다. 다음 오버레이 윈도우 레지스터(570)에 소정의 값이 써지면, 프로그램 버퍼(580)에 저장된 데이터가 메모리 셀 어레이(510)에 기입된다. 이 경우, 메모리 컨트롤러는 오버레이 윈도우 레지스터(570)를 폴링함으로써 메모리 셀 어레이(510)로의 쓰기 동작이 완료되었는지를 확인할 수 있다.
- [0065] 도 6은 본 발명의 한 실시예에 따른 PRAM에서 메모리 셀 어레이의 파티션 방법의 한 예를 설명하는 도면이고, 도 7은 본 발명의 한 실시예에 따른 PRAM에서 파티션의 한 예를 설명하는 도면이다.
- [0066] 도 6을 참고하면, 어떤 실시예에서, 하나의 메모리 셀 어레이(510), 예를 들면 PRAM 뱅크는 복수의 파티션(PART0-PART15)으로 분할될 수 있다. 도 6에서는 메모리 셀 어레이(510)가 16개의 파티션(PART0-PART15)으로 분할되는 예를 도시하였다. 이때, 복수의 행 버퍼(520, 530)는 복수의 파티션(PART0-PART15)에 연결될 수 있다. 예를 들면, 각 파티션은 64 비트 병렬 입출력 처리를 할 수 있다.
- [0067] 어떤 실시예에서, 복수의 파티션(PART0-PART15)은 센스 앰프(도 5의 550) 및 행 디코더(도 5의 540)와 같은 읽기 회로를 공유할 수 있다.
- [0068] 도 7을 참고하면, 어떤 실시예에서, 각 파티션은 복수의 서브어레이(sub-array)를 포함할 수 있다. 서브어레이는 저항성 타일(resistive tile)이라 불릴 수 있다. 도 7에서는 하나의 파티션이 64개의 타일(Tile0-Tile63)을 포함하는 것으로 도시하였다.
- [0069] 각 타일은 복수의 비트 라인(예를 들면 2048 비트 라인)과 복수의 워드 라인(예를 들면 4096 워드 라인)에 연결되는 복수의 메모리 셀, 즉 PRAM 코어를 포함한다. 설명의 편의상, 도 7에서는 각 타일에 포함되어 있는 복수의 메모리 셀 중에서 하나의 메모리 셀과 이에 연결되어 있는 하나의 비트 라인(BL) 및 워드 라인(WL)만을 도시하였고, 메모리 셀을 형성하는 상변화 소자와 스위칭 소자를 각각 저항과 다이오드로 도시하였다.
- [0070] 각 타일에는 로컬 열 디코더(local column decoder)(앞으로 "LYDEC"이라 함)(710)가 연결되어 있을 수 있다. LYDEC(710)은 대응하는 타일의 복수의 비트 라인(BL)에 연결되어 있다. 또한 파티션에는 복수의 타일에 각각 대응하는 복수의 글로벌 비트 라인(GBL)이 형성될 수 있다. 각 글로벌 비트 라인(GBL)은 대응하는 타일의 복수의 비트 라인(BL)에 연결되어 있으며, 글로벌 열 디코더(global column decoder)(앞으로 "GYDEC"이라 함)에 연결되어 있을 수 있다. 어떤 실시예에서, LYDEC(710)은 GYDEC과 함께 대응하는 파티션의 대응하는 타일에서 비트

라인(BL)을 선택하는데 사용될 수 있다. 센스 앰프(도 5의 550)가 선택된 비트 라인(BL)을 통해 데이터를 읽거나, 기입 드라이버(도 5의 570)가 선택된 비트 라인(BL)을 통해 데이터를 쓸 수 있다.

[0071] 병렬화를 최대화하기 위해서, 각 타일에는 서브 워드라인 드라이버(sub word line driver)(앞으로 "SWD"라 함)(720)가 연결되어 있을 수 있다. 파티션에 글로벌 워드 라인(GWL)이 형성되고, 글로벌 워드 라인(GWL)은 메인 워드 라인 드라이버(main word line driver)(앞으로 "MWD"라 함)(730)에 연결되어 있을 수 있다. 이 경우 파티션에 형성되는 복수의 워드 라인(WL)이 글로벌 워드 라인(GWL)에 연결되어 있을 수 있다. 파티션 내의 모든 SWD는 MWD(730)에 연결되어 있다. 어떤 실시예에서, SWD(720)는 MWD(730)과 함께 대응하는 타일에서 워드 라인(WL)을 구동하는데 사용될 수 있다. 이때, 구동되는 워드 라인(WL)은 행 디코더(도 5의 540)에 의해 선택될 수 있다.

[0072] 도 6 및 도 7에 도시한 파티션 구조를 사용하는 경우, 파티션당 복수의 입출력(input/output, I/O) 동작(예를 들면 도 7의 예에서는 64개의 I/O 동작)을 동시에 수행할 수 있다. 또한 도 6 및 도 7에 도시한 것처럼 각 파티션에서 로컬 디코더와 워드 라인 드라이버가 사용되므로, 메모리 시스템은 서로 다른 파티션에 병렬로 접근할 수 있다. 그러나 서로 다른 파티션은 수신한 I/O 요청의 유형이 서로 다른 경우에만 동시에 I/O 서비스를 지원할 수 있다.

[0073] 예를 들면, 도 8에 도시한 것처럼, PRAM의 파티션 0에서 쓰기 동작이 진행되는 동안, 파티션 0과 다른 파티션, 예를 들면 파티션 1에서 읽기 서비스를 제공할 수 있다. 이와 같이 서로 다른 파티션에서 읽기와 쓰기를 병렬로 제공할 수 있으므로, 느린 쓰기 동작 진행되는 동안 다른 읽기 동작이 진행될 수 있다. 따라서 쓰기 동작이 완료되기를 기다릴 필요 없이, 쓰기 지연 시간 동안 많은 읽기 요청을 병렬로 처리할 수 있다.

[0074] 그러나 PRAM의 파티션 0에서 쓰기 동작이 진행되는 동안, 파티션 0에서 읽기 서비스를 제공할 수는 없다. 그러므로 파티션 0에서 쓰기 동작이 완료된 후에, 파티션 0에서 읽기 동작이 진행될 수 있다. 이 경우, 읽기 요청의 응답 시간(즉, 읽기 동작이 완료되는데 걸리는 시간)은 쓰기 지연 시간과 읽기 지연 시간의 합에 해당할 수 있다. 이와 같이, 읽기 요청의 응답 시간이 오래 걸리는 경우를 방지하기 위한 실시예에 대해서 설명한다.

[0075] 다음 본 발명의 한 실시예에 따른 메모리 시스템에 대해서 도 8을 참고로 하여 설명한다.

[0076] 도 8은 본 발명의 한 실시예에 따른 메모리 시스템의 개략적인 블록도이다.

[0077] 도 8을 참고하면, 본 발명의 한 실시예에 따른 메모리 시스템은 메모리 제어기(800)와 메모리 서브시스템을 포함한다.

[0078] 도 1을 참고로 하여 설명한 것처럼, 메모리 제어기(800)는 PRAM 모듈(122a, 122b)과 DRAM 모듈(122c)를 포함하는 메모리 서브시스템에 복수의 채널(123a, 123b, 123c)을 통해 연결되어 있으며, 캐시(810), 캐시 제어 모듈(820) 및 마이그레이션(migration) 모듈(830)을 포함한다.

[0079] 메모리 서브시스템(122)은 PRAM 채널(123a, 123b)을 통해 각각 메모리 제어기(800)에 연결되는 PRAM 모듈(122a, 122b)과 DRAM 채널(123c)을 통해 메모리 제어기(800)에 연결되는 DRAM 모듈(122c)를 포함한다. 아래에서는 설명의 편의상 두 개의 PRAM 채널(123a, 123b), 즉 PRAM 모듈(122a, 122b)이 사용되는 예를 설명하지만, 본 발명은 이에 한정되지 않는다.

[0080] 캐시(810)는 CPU로부터 쓰기 요청을 수신하는 경우에 CPU에서 제공되는 원 데이터를 저장하며, 또한 캐시 정보를 저장하고 있다. 어떤 실시예에서, 캐시(810)는 원 데이터를 복수의 서브데이터(DATA_A, DATA_B)로 나누어서 저장할 수 있다. 캐시(810)는 PRAM보다 쓰기 속도가 빠른 메모리, 예를 들면 DRAM 또는 자기저항 랜덤 액세스 메모리(magnetoresistive random access memory, MRAM)를 사용한다. 이에 따라 메모리 시스템에서의 PRAM에 의한 쓰기 지연을 CPU로부터 감출 수 있다.

[0081] 마이그레이션 모듈(830)은 캐시(810)에 저장된 서브데이터(DATA_A, DATA_B)로부터 논블록킹 부호를 생성한다. 마이그레이션 모듈(830)은 서브데이터(DATA_A, DATA_B)를 각각 대응하는 PRAM 채널(123a, 123b)을 통해 PRAM 모듈(122a, 122b)에 쓰고, 논블록킹 부호를 DRAM 채널(123c)을 통해 DRAM 모듈(122c)에 써서, 캐시(810)의 데이터를 메모리 서브시스템(123)으로 마이그레이션한다. 어떤 실시예에서, 마이그레이션은 메모리 제어기(800)의 휴지(idle) 시간 동안 진행될 수 있다.

[0082] 한 실시예에서, 논블록킹 부호를 생성하는 부호화로 배타적 논리합(exclusive OR, XOR) 연산을 사용할 수 있다. 이 경우, 64 바이트의 원 데이터가 두 개의 32 바이트의 서브데이터(DATA_A, DATA_B)로 나누어지는 경우, 32 바이트의 서브데이터(DATA_A, DATA_B)를 비트단위(bitwise) XOR 연산하여서 32 바이트의 논블록킹 부호가 생성될

수 있다. 예를 들면, "001011100110"의 원 데이터가 "001011"의 DATA_A와 "100110"의 DATA_B로 나누어지는 경우, 논블록킹 부호는 "101101"로 생성될 수 있다. 이때, DATA_A와 논블록킹 부호의 XOR 연산을 통해 "100110"의 DATA_B를 복구하거나, DATA_B와 논블록킹 부호의 XOR 연산을 통해 "001011"의 DATA_A를 복구할 수 있다.

[0083] 다른 실시예에서, XOR 외에 논블록킹 부호를 생성할 수 있는 다른 부호화가 사용될 수 있다. 이러한 부호화를 사용하는 경우, 논블록킹 부호의 크기를 XOR 연산에 비해서 줄일 수 있다.

[0084] 한 예로, 오류 정정 부호화(error correction coding, ECC)가 논블록킹 부호를 생성하는 부호화로 사용될 수 있다. 예를 들면, 저밀도 패리티 검사(low density parity check, LDPC) 부호화를 통해서 논블록킹 부호로 패리티 부호를 생성할 수 있다.

[0085] 다른 예로, 소거 부호화(eraser coding)가 논블록킹 부호를 생성하는 부호화로 사용될 수 있다. 소거 부호화에서는, 원 데이터를 n개의 데이터 청크(chunk)로 나누고, 소거 부호화 알고리즘을 사용해서 n개의 데이터 청크로부터 k개의 부호 청크를 생성할 수 있다. 사용 불가능한 데이터 청크의 개수만큼 부호 청크를 사용해서, n개의 청크를 통해 원 데이터를 복구할 수 있다.

[0086] 또 다른 실시예에서, 전형적인 PRAM 모듈 및 그 메모리 제어기에서 오류 정정을 위해 제공되는 부호화가 사용될 수 있다.

[0087] 메모리 제어기(800)가 CPU로부터 읽기 요청을 수신하는 경우, 캐시 제어 모듈(820)은 룩업 모듈(810)을 조회하여서 캐시 히트인지 캐시 미스인지를 판단한다. 읽기 요청이 히트이면 캐시 제어 모듈(820)은 캐시(810)로부터 데이터를 읽는다. 읽기 요청이 미스인 경우, 메모리 제어기(800)는 메모리 서브시스템(820)으로부터 데이터를 읽는다. 이때, 두 PRAM 모듈(122a, 122b) 중 어느 하나의 PRAM 모듈(122a, 122b)로부터 서브데이터를 읽을 수 없는 경우가 발생할 수 있다. 예를 들면, PRAM 모듈(122b)에서 서브데이터(122b)가 저장된 파티션과 동일한 파티션에서 쓰기 동작이 진행 중이면(즉, 파티션 충돌이 발생하면), PRAM 모듈(122b)로부터 서브데이터(122b)를 읽을 수 없다. 이 경우, 메모리 제어기(800)는 PRAM 모듈(122a)로부터 서브데이터(DATA_A)를 읽고, DRAM 모듈(122c)로부터 논블록킹 부호를 읽는다. 그리고 메모리 제어기(800)의 캐시 제어 모듈(820)은 논블록킹 부호와 서브데이터(DATA_A)로부터 원 데이터를 재구성한다. 예를 들면, 논블록킹 부호 생성을 위한 부호화로 XOR 연산이 사용된 경우, 캐시 제어 모듈(820)은 서브데이터(DATA_A)와 논블록킹 부호를 XOR 연산하여서 원 데이터를 재구성할 수 있다.

[0088] 이와 같이, 본 발명의 한 실시예에 따르면, 캐시 미스로 인해 메모리 서브시스템으로부터 데이터를 읽는 경우에, 파티션 충돌과 같은 이유로 PRAM 모듈에서 데이터를 바로 읽을 수 없더라도, 쓰기 동작이 완료될 때까지 기다릴 필요 없이 논블록킹 부호를 이용해서 읽기 데이터를 제공할 수 있다.

[0089] 도 9는 본 발명의 다른 실시예에 따른 메모리 시스템의 개략적인 블록도이다.

[0090] 도 9를 참고하면, 메모리 제어기(900)는 룩업 모듈(910), 캐시 제어 모듈(820) 및 마이그레이션 모듈(830)을 포함하며, 룩업 모듈(910)은 캐시(911) 및 마이그레이션 큐(912)를 포함한다.

[0091] 캐시(911)는 원 데이터(911a) 외에 룩업 테이블(911b)을 저장하고 있을 수 있다. 어떤 실시예에서, 룩업 테이블(911b)은 캐시 인덱스에 대응하는 복수의 엔트리를 포함할 수 있으며, 각 엔트리는 캐시(911)의 저장 공간에서의 캐시 엔트리(예를 들면, 캐시 라인)를 지시할 수 있다. CPU로부터의 수신되는 I/O 요청은 태그와 캐시 인덱스로 분리될 수 있다. 태그와 캐시 인덱스는 캐시(911)의 크기에 의해 결정될 수 있다. 예를 들면, 512K개의 캐시 인덱스를 사용하는 경우, I/O 요청의 어드레스(예를 들면, 32 비트)는 옅셋(예를 들면, 5비트)을 제외하고, 19 비트의 캐시 인덱스와 3 비트의 태그로 분리될 수 있다. 이 경우, 룩업 테이블(911b)은 19 비트의 캐시 인덱스, 즉 512K개의 캐시 인덱스에 대응할 수 있도록 512K개의 엔트리를 가질 수 있다. 한 실시예에서, 각 엔트리는 태그 어레이, 카운터 및 유효 비트를 포함할 수 있다. 태그 어레이는 I/O 요청의 어드레스의 태그를 저장하고, 유효 비트는 대응하는 캐시 엔트리에 데이터가 존재하는지를 지시할 수 있다. 또한 카운터는 대응하는 캐시 엔트리의 액세스에 따라 갱신될 수 있다.

[0092] 마이그레이션 큐(912)는 마이그레이션이 필요한 캐시 엔트리의 정보를 저장한다. 마이그레이션 큐(912)는 PRAM보다 쓰기 속도가 빠른 메모리, 예를 들면 DRAM 또는 MRAM으로 구현될 수 있다.

[0093] 캐시 제어 모듈(820)은 I/O 요청의 캐시 히트/미스 여부를 판단하기 위해서 룩업 모듈(910)로 조회(lookup) 명령을 전달할 수 있다. I/O 요청의 어드레스와 룩업 모듈(910)의 룩업 테이블(911b)에 기초해서 I/O 요청의 캐시 히트 또는 캐시 미스가 판단될 수 있다. 캐시 미스의 경우, 캐시 제어 모듈(820)은 빈 캐시 엔트리가 존재하는

지를 판단하기 위해서 룩업 모듈(910)로 찾기(find) 명령을 전달할 수 있다. 룩업 모듈(910)의 룩업 테이블(911b)에 기초해서 빈 캐시 엔트리가 존재하는지 판단될 수 있다. 캐시 제어 모듈(820)은 룩업 테이블(911b)의 갱신을 위해 갱신 명령을 룩업 모듈(910)로 전달할 수 있다. 이에 따라, 캐시 히트의 경우 캐시 히트의 캐시 엔트리에 대응하는 룩업 테이블의 카운터가 갱신될 수 있으며, 캐시 미스의 경우 룩업 테이블에 새로운 엔트리가 추가되고 카운터가 갱신될 수 있다.

[0094] 마이그레이션 모듈(830)은 마이그레이션을 시작하기 위해서 마이그레이션 큐(912)에서 헤드(head)가 포인트하는 정보(즉, 헤드 정보)를 읽기 위해서 룩업 모듈(910)로 읽기 명령을 전달할 수 있다. 그러면 룩업 모듈(910)은 마이그레이션 큐(912)에서 헤드 정보를 읽고, 헤드 정보에 해당하는 캐시 엔트리에 저장된 데이터의 마이그레이션이 시작될 수 있다. 마이그레이션 모듈(830)은 마이그레이션이 완료된 데이터를 캐시 엔트리에서 제거하기 위해서 제거 명령을 룩업 모듈(910)로 전달할 수 있다. 그러면 룩업 모듈(910)은 마이그레이션 큐(912)에서 헤드에 해당하는 큐 엔트리를 제거하고 다음 큐 엔트리로 헤드를 이동하고, 해당 캐시 엔트리에서 데이터를 제거할 수 있다. 룩업 모듈(910)은 새로운 엔트리가 추가되는 경우(즉, 캐시 엔트리에 새로운 데이터가 저장되는 경우), 삽입(insert) 명령을 통해 마이그레이션 큐(912)에 새로 입력된 캐시 엔트리의 정보를 추가할 수 있다.

[0095] 메모리 제어기(900)는 PRAM 모듈(122a, 122b) 및 DRAM 모듈(122c)를 각각 제어하기 위한 메모리 제어 유닛(MCU)(951, 952, 953)을 더 포함할 수 있다. 또한 메모리 제어기(900)는 캐시 제어 모듈(820)로부터의 읽기/쓰기에 따른 메모리 요청 또는 마이그레이션 제어 모듈(830)로부터의 마이그레이션에 따른 메모리 요청을 대응하는 메모리 제어 유닛(MCU)(951, 952, 953)으로 전달하기 위한 다중화기(940)를 더 포함할 수 있다.

[0096] 도 10은 본 발명의 한 실시예에 따른 메모리 제어기에서 캐시 제어 모듈의 동작을 나타내는 흐름도이다.

[0097] 도 10을 참고하면, 캐시 제어 모듈(도 8 또는 도 9의 820)은 캐시(도 8의 810 또는 도 9의 911)의 캐시 정보를 참조해서 CPU로부터의 요청이 캐시 히트인지 캐시 미스인지를 판단한다(S1010). 캐시 히트인 경우(S1010: 예), 캐시 제어 모듈(820)은 캐시를 통해 요청에 해당하는 서비스를 제공한다(S1020, S1030, S1035). 즉, 요청이 읽기 요청이면(S1020: 예), 캐시 제어 모듈(820)은 캐시(810 또는 911)로부터 읽기 요청에 대응하는 데이터를 읽는다(S1030). 요청이 쓰기 요청이면(S1020: 아니요), 캐시 제어 모듈(820)은 캐시(810 또는 911)에 쓰기 요청에 대응하는 데이터를 쓴다(S1035). 어떤 실시예에서, 캐시 제어 모듈(820)은 캐시 히트에 따라 캐시 정보를 갱신할 수 있다(S1050).

[0098] 캐시 미스인 경우(S1010: 아니요), 요청이 쓰기 요청이면(S1040: 예), 캐시 제어 모듈(820)은 캐시 정보를 참조해서 캐시(810 또는 911)에 빈 엔트리(즉, 캐시 라인)가 있는지 판단한다(S1045). 캐시(810 또는 911)에 빈 엔트리가 있으면(S1045: 예), 캐시 제어 모듈(820)은 캐시(즉, 캐시의 빈 엔트리)(810 또는 911)에 쓰기 요청에 해당하는 데이터를 쓴다(S1050). 어떤 실시예에서, 캐시 제어 모듈(820)은 데이터가 쓰여진 엔트리에 대한 정보를 캐시 정보에 추가하여서 캐시 정보를 갱신할 수 있다(S1050). 캐시(810 또는 911)에 빈 엔트리가 없으면, 캐시(810 또는 911)에서 메모리 서브시스템(도 8의 122)의 PRAM 모듈로 데이터가 써지는 중(즉, 마이그레이션되는 중)이므로, 캐시 제어 모듈(820)은 PRAM 모듈로의 쓰기가 완료될 때까지 대기한다(S1055: 아니요). PRAM 모듈로의 쓰기가 완료되어 캐시(810 또는 911)에 빈 엔트리가 생긴 후에(S1055: 예), 캐시 제어 모듈(820)은 캐시(810 또는 911)의 빈 엔트리에 쓰기 요청에 해당하는 데이터를 쓴다(S1050).

[0099] 캐시 미스인 경우, 요청이 읽기 요청이면(S1040: 아니요), 캐시 제어 모듈(820)은 현재 메모리 서브시스템(122)이 PRAM 모듈(도 8 또는 도 9의 122a, 122b)로부터 원 데이터에 해당하는 모든 서브데이터를 읽을 수 있는 조건을 만족하는지를 판단한다(S1060). 어떤 실시예에서, PRAM 모듈(122a, 122b) 모두에 쓰기가 진행 중이지 않으면, 메모리 서브시스템(122)이 PRAM 모듈(122a, 122b)로부터 원 데이터에 해당하는 모든 서브데이터를 읽을 수 있다. 또는 PRAM 모듈(122a, 122b) 중 어느 하나에 쓰기가 진행 중이더라도, 쓰기 진행 중인 파티션과 서브데이터를 읽을 파티션이 충돌하지 않는다면, 메모리 서브시스템(122)이 PRAM 채널(도 8 또는 도 9의 123a, 123b)를 통해 PRAM 모듈(122a, 122b)로부터 원 데이터에 해당하는 모든 서브데이터를 읽을 수 있다.

[0100] 메모리 서브시스템(122)이 PRAM 모듈(122a, 122b)로부터 원 데이터에 해당하는 모든 서브데이터를 읽을 수 있으면(S1060: 예), 캐시 제어 모듈(820)은 PRAM 모듈(122a, 122b)로부터 서브데이터를 읽어서 CPU로 전달한다(S1070). 메모리 서브시스템(122)이 일부 PRAM 모듈(예를 들면, 122b)로부터 서브데이터를 읽을 수 없으면(S1060: 아니요), 캐시 제어 모듈(820)은 데이터 읽기가 가능한 PRAM 모듈(122a)로부터 서브데이터를 읽고, DRAM 채널(도 8 또는 도 9의 123c)을 통해 DRAM 모듈(도 8 또는 도 9의 122c)로부터 논블록킹 부호를 읽는다(S1075). 캐시 제어 모듈(820)은 서브데이터와 논블록킹 부호로부터 원 데이터를 재구성하여 CPU로 전달한다(S1075).

- [0101] 도 11은 본 발명의 한 실시예에 따른 메모리 제어기에서 마이그레이션 모듈의 동작을 나타내는 흐름도이다.
- [0102] 도 11을 참고하면, 마이그레이션 모듈(도 8 또는 도 9의 830)은 캐시(810 또는 911)에 마이그레이션을 시작할 데이터가 저장되어 있는지를 판단한다(S1110). 어떤 실시예에서, 마이그레이션 모듈(830)은 마이그레이션 큐(도 9의 912)가 비어 있는지를 확인할 수 있다(S1110). 마이그레이션 큐(912)가 비어 있으면, 즉 캐시(810 또는 911)에 마이그레이션할 데이터가 없으면(S1110: 예), 마이그레이션 모듈(830)은 마이그레이션 큐(912)에 새로운 캐시 엔트리의 정보가 추가될 때까지(즉, 캐시(911)에 새로운 데이터가 저장될 때까지) 대기한다.
- [0103] 마이그레이션 큐(912)가 비어 있지 않으면, 즉 캐시(810 또는 911)에 마이그레이션할 데이터가 있으면(S1110: 아니요), 마이그레이션 모듈(830)은 캐시(810 또는 911)로부터 마이그레이션할 데이터를 읽고, 원 데이터가 나누어진 복수의 서브데이터로부터 논블록킹 부호를 생성한다(S1120). 어떤 실시예에서, 마이그레이션 모듈(830)은 마이그레이션 큐(912)에서 헤드 정보에 대응하는 캐시 엔트리에 저장된 데이터를 읽을 수 있다(S1120). 마이그레이션 모듈(830)은 서브데이터를 대응하는 PRAM 채널을 통해 대응하는 PRAM 모듈에 쓰고, 논블록킹 부호를 DRAM 채널(도 8 또는 도 9의 123c)을 통해 DRAM 모듈(도 8 또는 도 9의 122c)에 쓴다(S1130). 어떤 실시예에서, 마이그레이션 모듈(830)은 복수의 서브데이터 중 일부 서브데이터만 대응하는 PRAM 채널을 통해 대응하는 PRAM 모듈에 쓰고, 논블록킹 부호를 DRAM 채널을 통해 DRAM 모듈에 쓸 수 있다(S1130). 한 실시예에서, 원 데이터가 두 개의 서브데이터로 나누어지는 경우, 마이그레이션 모듈(830)은 하나의 서브데이터를 PRAM 채널(도 8 또는 도 9의 123a)을 통해 대응하는 PRAM 모듈(도 8 또는 도 9의 122a)에 쓸 수 있다(S1130). 모든 서브데이터를 모든 PRAM 채널을 통해 PRAM 모듈로 쓰는 경우, 읽기 요청이 발생했을 때 읽기 요청과 파티션 충돌이 발생하는 경우 읽기 요청이 차단될 수 있기 때문에, 일부 서브데이터에 대해서만 쓰기가 진행될 수 있다. 그러면 읽기 요청에 대해서 PRAM 모듈(122a)에서 파티션 충돌이 발생하더라도, 쓰기 진행 중이지 않은 다른 PRAM 모듈(도 8 또는 도 9의 122b)에서 읽은 데이터와 논블록킹 부호로 원 데이터를 재구성할 수 있다.
- [0104] 어떤 실시예에서, 쓰기 동작이 수행된 PRAM 모듈(122a)에서 쓰기가 완료되면(S1140: 예), 마이그레이션 모듈(830)은 나머지 일부 서브데이터를 대응하는 PRAM 채널을 통해 대응하는 PRAM 모듈에 쓸 수 있다(S1160). 한 실시예에서, 원 데이터가 두 개의 서브데이터로 나누어지는 경우, 마이그레이션 모듈(830)은 나머지 서브데이터를 PRAM 채널(도 8의 123b)을 통해 대응하는 PRAM 모듈(122b)에 쓸 수 있다(S1160).
- [0105] 어떤 실시예에서, 나머지 일부 서브데이터를 대응하는 PRAM 모듈에 쓰기 전에(S1160), 마이그레이션을 진행하고 있는 데이터에 대해서 캐시 히트가 발생하면(S1150: 예), 마이그레이션 모듈(830)은 나머지 일부 서브데이터에 대해서 쓰기를 진행하지 않고 마이그레이션을 종료할 수 있다. 캐시(911)의 데이터를 PRAM 모듈로 마이그레이션하는 중에, 해당 데이터에 대해서 캐시 히트가 발생하면, 요청한 데이터를 우선적으로 처리할 필요가 있기 때문에 마이그레이션을 종료할 수 있다. 캐시 히트가 발생하지 않으면(S1150: 아니요), 마이그레이션 모듈(830)은 나머지 일부 서브데이터를 대응하는 PRAM 채널을 통해 대응하는 PRAM 모듈에 쓸 수 있다(S1160).
- [0106] 나머지 일부 서브데이터의 쓰기가 완료하면(S1170: 예), 마이그레이션 모듈(830)은 캐시(911)에서 해당하는 캐시 엔트리의 데이터를 삭제한다(S1180). 어떤 실시예에서, 마이그레이션 모듈(830)은 마이그레이션 큐(912)에서 큐 엔트리를 삭제할 수 있다(S1180).
- [0107] 이와 같이, 본 발명의 한 실시예에 따르면, 쓰기 지연이 긴 PRAM 모듈 대신에 캐시에서 먼저 쓰기가 진행되므로, PRAM 모듈의 긴 쓰기 지연을 감출 수 있다. 또한 파티션 충돌이 있는 쓰기와 읽기가 동시에 진행될 수 있으므로, 읽기 요청의 차단 없이 읽기 요청의 데이터가 제공될 수 있다. 또한 메모리 제어기의 휴지 시간 동안 캐시의 데이터를 PRAM으로 마이그레이션해서 캐시를 비울 수 있으므로, 메모리 시스템의 전체적인 응답 속도를 높일 수 있다. 이에 따라, 본 발명의 한 실시예에 따른 메모리 시스템을 CPU의 메인 메모리로 사용할 수 있으며, 또한 DRAM과 달리 비휘발성 메모리 시스템을 제공할 수 있다.
- [0108] 일반적으로, 메모리 연속성(persistence)을 제공하기 위해서, 컴퓨팅 디바이스의 전원 장애 등으로 셧다운될 때 데이터가 유지되어야 할 뿐만 아니라, 컴퓨팅 디바이스가 다시 부팅되었을 때 정지되었던 위치에서 다시 동작이 수행될 수 있어야 한다. 이를 위해, 컴퓨팅 디바이스의 재부팅 시에 로그 영역(log area)을 먼저 탐색해서 종료되었던 시점으로 복구하고 컴퓨팅 디바이스를 구동하는 로깅 메커니즘(logging mechanism)이 사용될 수 있다.
- [0109] 도 12에 도시한 것처럼, 로깅 메커니즘을 위해 PRAM 모듈(1200)에는 데이터 저장 영역(1210) 외에 로그 영역(1220)이 존재한다. 메모리 제어기는 CPU로부터 로그 데이터(log data)를 위한 캐시 라인 플러시 명령(ciflush)을 수신하는 경우, 캐시의 데이터를 로그 영역에 저장한다. 다음 메모리 제어기는 데이터를 위한 캐시 라인 플러시 명령(ciflush)과 펜스 명령(sfense)을 수신하는 경우, 캐시의 데이터를 PRAM 모듈의 데이터 저장 영역에

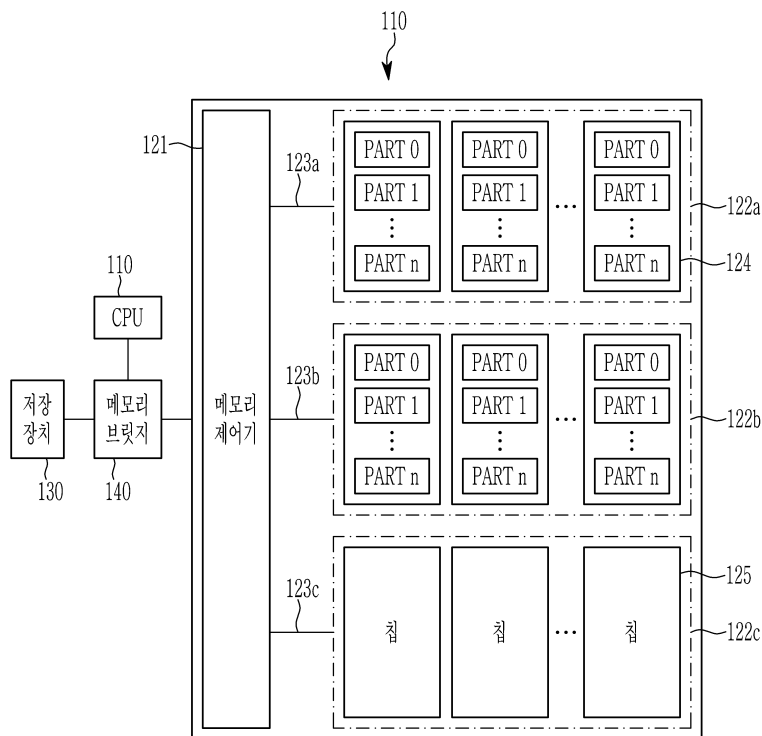
저장한다. 따라서 컴퓨팅 디바이스가 셧다운된 후에 재부팅되는 경우, 로그 영역의 로그 데이터가 정상적이라면 로그 데이터로 데이터를 복구할 수 있다. 로그 영역의 로그 데이터가 비정상적이라면 명령이 실행되지 않은 것으로 판단하고 무시할 수 있다. 이 경우, 캐시의 데이터를 PRAM의 데이터 저장 영역에 저장하기 전에 로그 데이터를 로그 영역(1220)에 중복해서 저장하므로, 데이터의 중복 문제가 발생하고, 명령 실행에서 속도 저하가 발생할 수 있다.

[0110] 그러나 본 발명의 한 실시예에 따르면, 마이그레이션을 통해 캐시(도 8의 810 또는 도 9의 911)에 저장된 데이터가 지속적으로 비휘발성 메모리인 PRAM 모듈에 기록될 수 있다. 이 경우, 캐시(810 또는 911)로 DRAM과 같은 휘발성 메모리가 사용하는 경우, 전원 장애가 발생하더라도 백업용 배터리를 통해 캐시의 데이터가 사라지기 전에 PRAM 모듈로 데이터를 기록할 수 있으므로, 로깅 메커니즘을 사용하지 않더라도 전원 장애로 인한 메모리 연속성을 제공할 수 있다. 어떤 실시예에서, 캐시(810 또는 911)가 비휘발성 메모리, 예를 들면 MRAM으로 구현될 수 있다. 이 경우, 전원 장애 등으로 컴퓨팅 디바이스가 셧다운되더라도 비휘발성인 캐시(810, 911)에 데이터가 보존되어 있으므로, 로깅 메커니즘 없이 메모리 연속성을 제공할 수 있으며, 또한 백업용 배터리를 사용하지 않을 수도 있다.

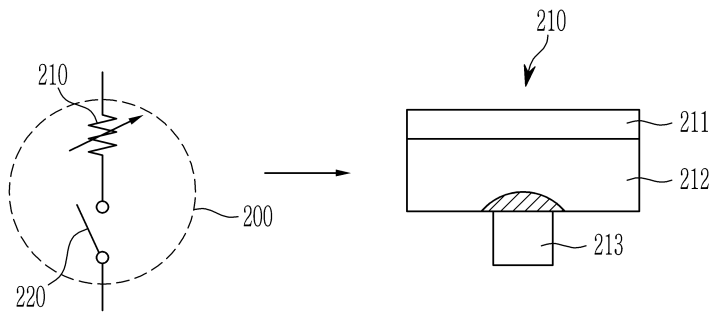
[0111] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

도면

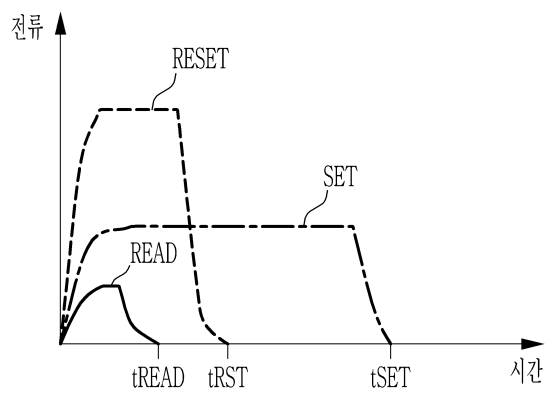
도면1



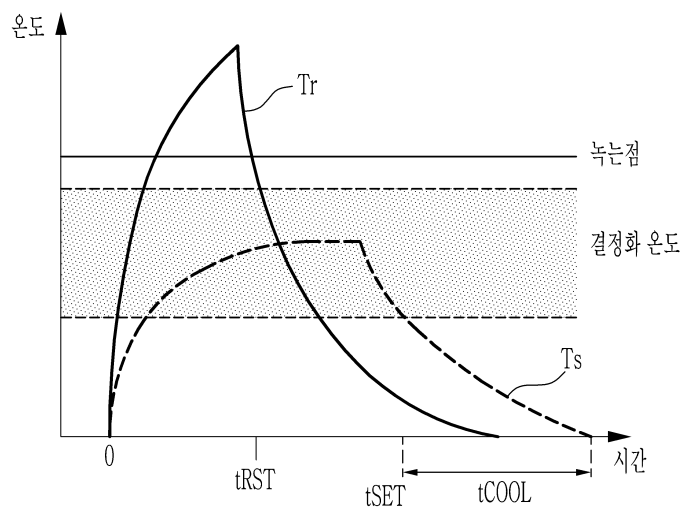
도면2



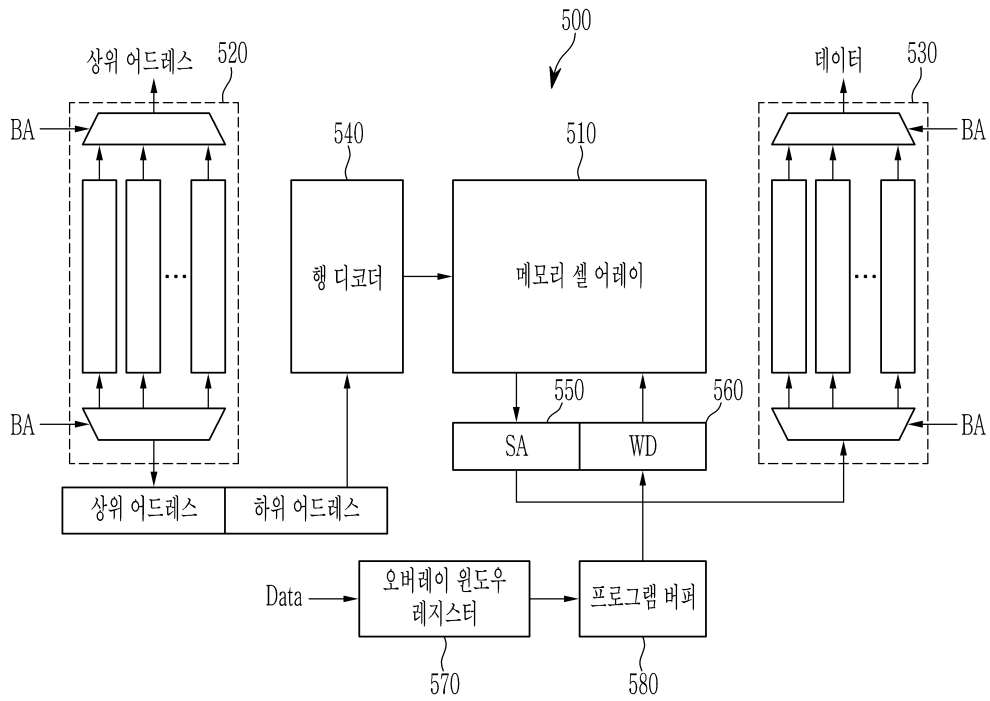
도면3



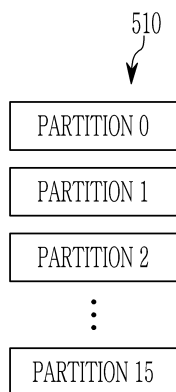
도면4



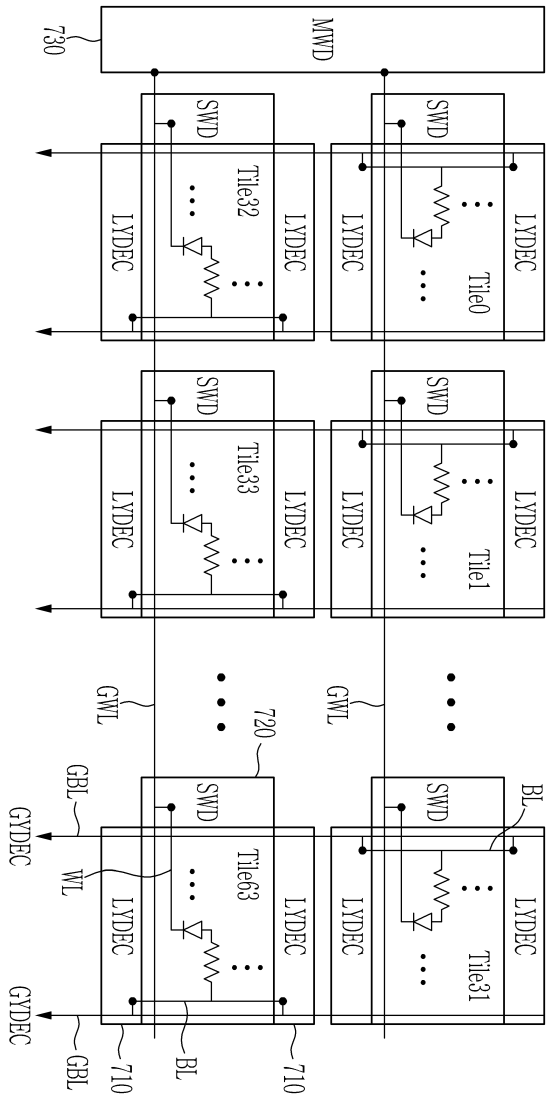
도면5



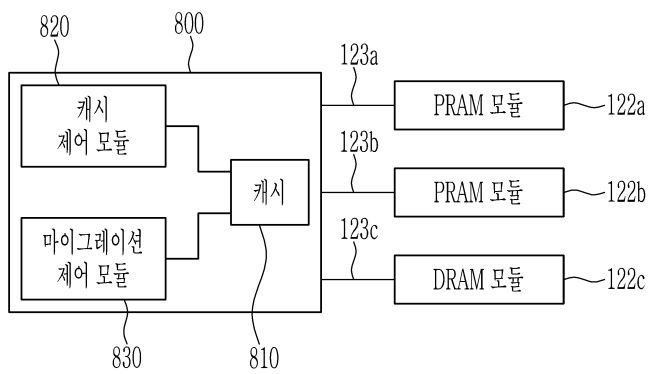
도면6



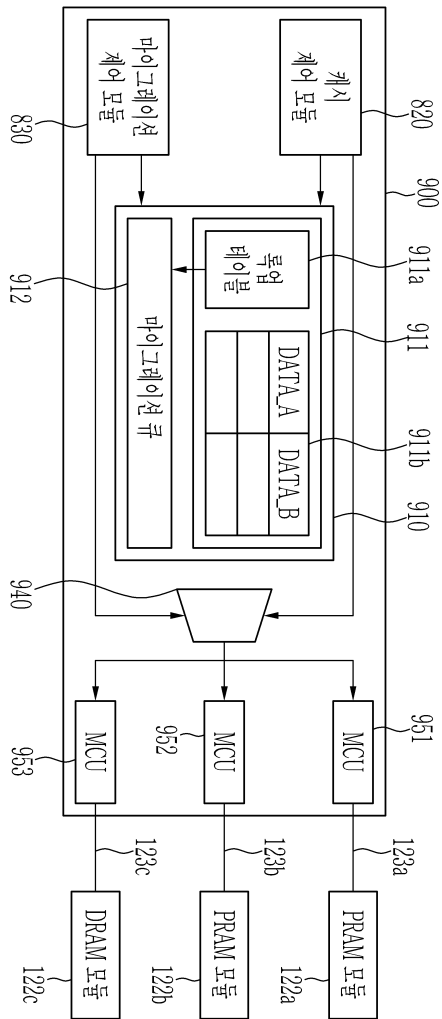
도면7



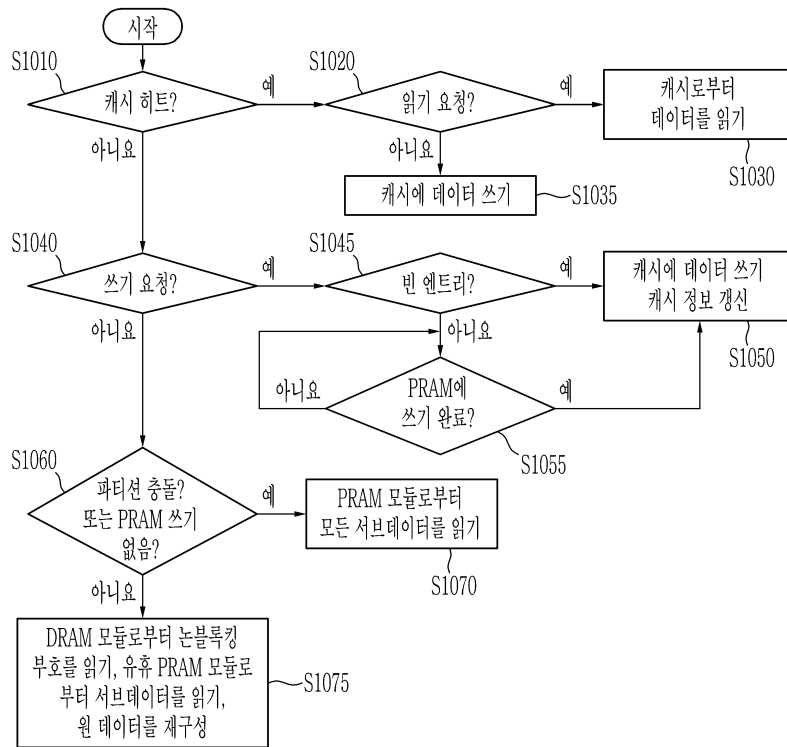
도면8



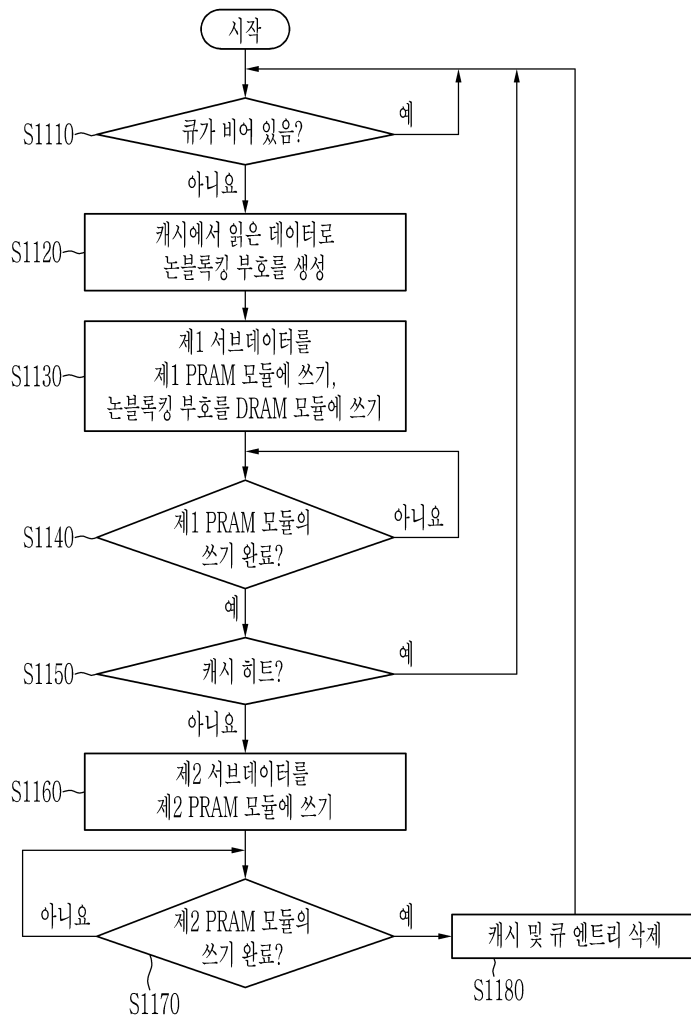
도면9



도면10



도면11



도면12

