



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0136382  
(43) 공개일자 2017년12월11일

(51) 국제특허분류(Int. Cl.)  
G06F 3/06 (2006.01) G11C 7/10 (2015.01)  
(52) CPC특허분류  
G06F 3/0659 (2013.01)  
G06F 3/0656 (2013.01)  
(21) 출원번호 10-2016-0068400  
(22) 출원일자 2016년06월01일  
심사청구일자 없음

(71) 출원인  
주식회사 맵레이  
서울특별시 강남구 도산대로 221, 703호 (신사동, 동남빌딩)  
연세대학교 산학협력단  
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)  
(72) 발명자  
이재수  
인천시 연수구 송도과학로 85, 진리관C 304호 (송도동, 연세대학교)  
정명수  
인천시 연수구 송도과학로 85, 진리관C 304호 (송도동, 연세대학교)  
박규영  
인천시 연수구 송도과학로 85, 진리관C 304호 (송도동, 연세대학교)  
(74) 대리인  
유미특허법인

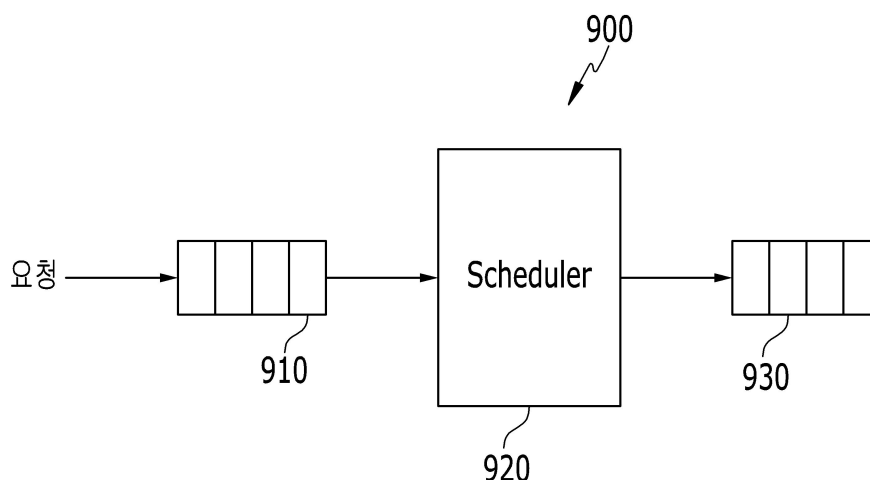
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 메모리 컨트롤러, 그리고 이를 포함하는 메모리 모듈 및 프로세서

### (57) 요약

상변화 메모리를 사용하며 메모리 셀 어레이가 복수의 파티션으로 구분되어 있는 메모리의 메모리 컨트롤러가 제공된다. 요청 큐에는 메모리로 데이터 쓰기를 요청하는 쓰기 요청과 메모리로부터 데이터 읽기를 요청하는 읽기 요청이 입력된다. 스케줄러는 복수의 파티션 중에서 제1 파티션에서 쓰기 동작이 진행되고 있는 제1 조건을 포함하는 충돌 확인 조건이 만족되는 경우, 제1 파티션에서의 쓰기 동작과 읽기 동작이 충돌하지 않는 제2 파티션에 해당하는 읽기 요청이 요청 큐에 존재하면, 제2 파티션에 해당하는 읽기 요청에 기초하여 제2 파티션을 위한 읽기 명령을 생성한다.

대표도 - 도9



(52) CPC특허분류

*G06F 3/0658* (2013.01)

*G11C 7/1057* (2013.01)

*G11C 7/1084* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

상변화 메모리를 사용하며 메모리 셀 어레이가 복수의 파티션으로 구분되어 있는 메모리의 메모리 컨트롤러로서,

상기 메모리로 데이터 쓰기를 요청하는 쓰기 요청과 상기 메모리로부터 데이터 읽기를 요청하는 읽기 요청이 입력되는 요청 큐, 그리고

상기 복수의 파티션 중에서 제1 파티션에서 쓰기 동작이 진행되고 있는 제1 조건을 포함하는 충돌 확인 조건이 만족되는 경우, 상기 제1 파티션에서의 쓰기 동작과 읽기 동작이 충돌하지 않는 제2 파티션에 해당하는 읽기 요청이 상기 요청 큐에 존재하면, 상기 제2 파티션에 해당하는 읽기 요청에 기초하여서 상기 제2 파티션을 위한 읽기 명령을 생성하는 스케줄러

를 포함하는 메모리 컨트롤러.

#### 청구항 2

제1항에서,

상기 읽기 명령의 생성에 사용되는 읽기 요청은 상기 제2 파티션에 해당하는 읽기 요청 중에서 가장 오래된 읽기 요청을 포함하는 메모리 컨트롤러.

#### 청구항 3

제1항에서,

상기 메모리는 상기 메모리 셀 어레이로부터 읽은 데이터가 저장되는 복수의 행 데이터 버퍼를 포함하며,

상기 복수의 행 데이터 버퍼 중에서 상기 제2 파티션에 해당하는 읽기 요청에 해당하는 데이터에 히트하는 행 데이터 버퍼가 존재하면, 상기 스케줄러는 상기 히트하는 행 데이터 버퍼를 선택하는

메모리 컨트롤러.

#### 청구항 4

제1항에서,

상기 메모리는 상기 메모리 셀 어레이로부터 읽은 데이터가 저장되는 복수의 행 데이터 버퍼를 포함하며,

상기 제2 파티션에 해당하는 읽기 요청에 해당하는 데이터에 상기 복수의 행 데이터 버퍼에 히트하지 않으면, 상기 스케줄러는 상기 복수의 행 데이터 버퍼 중에서 가장 오래된 데이터를 저장하고 있는 행 데이터 버퍼에 상기 제2 파티션에 해당하는 읽기 요청에 해당하는 데이터를 저장하는

메모리 컨트롤러.

#### 청구항 5

제1항에서,

상기 충돌 확인 조건은, 상기 제1 파티션에서 상기 쓰기 동작이 진행되고 있는 동안 읽기 동작을 위해 열린 워드 라인에 해당하는 읽기 요청이 상기 요청 큐에 존재하지 않는 제2 조건을 더 포함하는 메모리 컨트롤러.

#### 청구항 6

제5항에서,

상기 제1 조건을 만족하고 상기 제2 조건을 만족하지 않는 경우, 상기 스케줄러는 상기 열린 워드 라인에 해당

하는 읽기 요청에 기초하여서 상기 읽기 명령을 생성하는 메모리 컨트롤러.

#### 청구항 7

제1항에서,

상기 메모리에서 쓰기 동작이 진행되고 있지 않는 경우, 상기 요청 큐에서 가장 오래된 요청이 쓰기 요청이면, 상기 스케줄러는 가장 오래된 쓰기 요청에 기초하여 쓰기 명령을 생성하는 메모리 컨트롤러.

#### 청구항 8

제7항에서,

상기 스케줄러는 상기 가장 오래된 쓰기 요청과 상기 가장 오래된 쓰기 요청과 동시에 쓰기가 가능한 메모리 셀에 해당하는 쓰기 요청에 기초하여 상기 쓰기 명령을 생성하는 메모리 컨트롤러.

#### 청구항 9

제1항에서,

상기 메모리에서 쓰기 동작이 진행되고 있지 않는 경우, 상기 요청 큐에서 가장 오래된 요청이 읽기 요청이면, 상기 스케줄러는 읽기 명령을 생성하는 메모리 컨트롤러.

#### 청구항 10

제9항에서,

읽기 동작을 위해 열린 워드 라인에 해당하는 읽기 요청이 상기 요청 큐에 존재하는 경우, 상기 스케줄러는 상기 열린 워드 라인에 해당하는 읽기 요청에 기초하여서 상기 읽기 명령을 생성하는 메모리 컨트롤러.

#### 청구항 11

상변화 메모리를 사용하며 메모리 셀 어레이가 복수의 파티션으로 구분되어 있는 메모리의 메모리 컨트롤러로서,

상기 메모리로 데이터 쓰기를 요청하는 쓰기 요청과 상기 메모리로부터 데이터 읽기를 요청하는 읽기 요청이 입력되는 요청 큐, 그리고

상기 복수의 파티션 중에서 제1 파티션에서 쓰기 동작이 진행되고 있는 경우, 소정의 읽기 요청에 기초하여서 읽기 명령을 생성하는 스케줄러

를 포함하는 메모리 컨트롤러.

#### 청구항 12

제11항에서,

상기 소정의 읽기 요청은 상기 제1 파티션에서의 쓰기 동작과 읽기 동작이 충돌하지 않는 제2 파티션에 해당하는 읽기 요청을 포함하는 메모리 컨트롤러.

#### 청구항 13

제12항에서,

상기 읽기 명령의 생성에 사용되는 읽기 요청은 상기 제2 파티션에 해당하는 읽기 요청 중에서 가장 오래된 읽기 요청을 포함하는 메모리 컨트롤러.

#### 청구항 14

제11항에서,

상기 소정의 읽기 요청은 읽기 동작을 위해 열린 워드 라인에 해당하는 읽기 요청을 포함하는 메모리 컨트롤러.

#### 청구항 15

제11항에서,

상기 메모리에서 쓰기 동작이 진행되고 있지 않는 경우, 상기 요청 큐에서 가장 오래된 요청이 쓰기 요청이면, 상기 스케줄러는 가장 오래된 쓰기 요청과 상기 가장 오래된 쓰기 요청과 동시에 쓰기가 가능한 메모리 셀에 해당하는 쓰기 요청에 기초하여 상기 쓰기 명령을 생성하는 메모리 컨트롤러.

#### 청구항 16

제1항 내지 제15항 중 어느 한 항에 따른 메모리 컨트롤러, 그리고

상기 메모리

를 포함하는 메모리 모듈.

#### 청구항 17

제1항 내지 제15항 중 어느 한 항에 따른 메모리 컨트롤러를 포함하며, 상기 메모리와 시스템 버스를 통해 연결되는 프로세서.

### 발명의 설명

#### 기술 분야

[0001] 본 발명은 메모리 컨트롤러, 그리고 이를 포함하는 메모리 모듈 및 프로세서에 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 메모리의 셀 어레이는 복수의 파티션(partition)으로 구분되어서 사용될 수 있지만, 현재 메모리 셀 어레이에 데이터를 쓰거나 읽기 위한 스케줄링을 할 때 파티션이 고려되지 않고 있다.

### 발명의 내용

#### 해결하려는 과제

[0004] 본 발명이 이루고자 하는 과제는 파티션을 고려한 스케줄링을 할 수 있는 메모리 컨트롤러, 그리고 이를 포함하는 메모리 모듈 및 프로세서를 제공하는 것이다.

#### 과제의 해결 수단

[0005] 본 발명의 한 실시예에 따르면, 상변화 메모리를 사용하며 메모리 셀 어레이가 복수의 파티션으로 구분되어 있는 메모리의 메모리 컨트롤러가 제공된다. 상기 메모리 컨트롤러는 요청 큐와 스케줄러를 포함한다. 상기 요청 큐에는 상기 메모리로 데이터 쓰기를 요청하는 쓰기 요청과 상기 메모리로부터 데이터 읽기를 요청하는 읽기 요청이 입력된다. 상기 스케줄러는, 상기 복수의 파티션 중에서 제1 파티션에서 쓰기 동작이 진행되고 있는 제1 조건을 포함하는 충돌 확인 조건이 만족되는 경우, 상기 제1 파티션에서의 쓰기 동작과 읽기 동작이 충돌하지 않는 제2 파티션에 해당하는 읽기 요청이 상기 요청 큐에 존재하면, 상기 제2 파티션에 해당하는 읽기 요청에 기초하여서 상기 제2 파티션을 위한 읽기 명령을 생성한다.

[0006] 상기 읽기 명령의 생성에 사용되는 읽기 요청은 상기 제2 파티션에 해당하는 읽기 요청 중에서 가장 오래된 읽기 요청을 포함할 수 있다.

[0007] 상기 메모리는 상기 메모리 셀 어레이로부터 읽은 데이터가 저장되는 복수의 행 데이터 버퍼를 포함할 수 있다. 상기 복수의 행 데이터 버퍼 중에서 상기 제2 파티션에 해당하는 읽기 요청에 해당하는 데이터에 히트하는 행 데이터 버퍼가 존재하면, 상기 스케줄러는 상기 히트하는 행 데이터 버퍼를 선택할 수 있다.

- [0008] 상기 메모리는 상기 메모리 셀 어레이로부터 읽은 데이터가 저장되는 복수의 행 데이터 버퍼를 포함할 수 있다. 상기 제2 파티션에 해당하는 읽기 요청에 해당하는 데이터에 상기 복수의 행 데이터 버퍼에 히트하지 않으면, 상기 스케줄러는 상기 복수의 행 데이터 버퍼 중에서 가장 오래된 데이터를 저장하고 있는 행 데이터 버퍼에 상기 제2 파티션에 해당하는 읽기 요청에 해당하는 데이터를 저장할 수 있다.
- [0009] 상기 충돌 확인 조건은, 상기 제1 파티션에서 상기 쓰기 동작이 진행되고 있는 동안 읽기 동작을 위해 열린 워드 라인에 해당하는 읽기 요청이 상기 요청 큐에 존재하지 않는 제2 조건을 더 포함할 수 있다.
- [0010] 상기 제1 조건을 만족하고 상기 제2 조건을 만족하지 않는 경우, 상기 스케줄러는 상기 열린 워드 라인에 해당하는 읽기 요청에 기초하여서 상기 읽기 명령을 생성할 수 있다.
- [0011] 상기 메모리에서 쓰기 동작이 진행되고 있지 않는 경우, 상기 요청 큐에서 가장 오래된 요청이 쓰기 요청이면, 상기 스케줄러는 가장 오래된 쓰기 요청에 기초하여 쓰기 명령을 생성할 수 있다.
- [0012] 상기 스케줄러는 상기 가장 오래된 쓰기 요청과 상기 가장 오래된 쓰기 요청과 동시에 쓰기가 가능한 메모리 셀에 해당하는 쓰기 요청에 기초하여 상기 쓰기 명령을 생성할 수 있다.
- [0013] 상기 메모리에서 쓰기 동작이 진행되고 있지 않는 경우, 상기 요청 큐에서 가장 오래된 요청이 읽기 요청이면, 상기 스케줄러는 읽기 명령을 생성할 수 있다.
- [0014] 읽기 동작을 위해 열린 워드 라인에 해당하는 읽기 요청이 상기 요청 큐에 존재하는 경우, 상기 스케줄러는 상기 열린 워드 라인에 해당하는 읽기 요청에 기초하여서 상기 읽기 명령을 생성할 수 있다.
- [0015] 본 발명의 다른 실시예에 따르면, 상변화 메모리를 사용하며 메모리 셀 어레이가 복수의 파티션으로 구분되어 있는 메모리의 메모리 컨트롤러가 제공된다. 상기 메모리 컨트롤러는 요청 큐와 스케줄러를 포함한다. 상기 요청 큐에는 상기 메모리로 데이터 쓰기를 요청하는 쓰기 요청과 상기 메모리로부터 데이터 읽기를 요청하는 읽기 요청이 입력된다. 상기 스케줄러는, 상기 복수의 파티션 중에서 제1 파티션에서 쓰기 동작이 진행되고 있는 경우, 소정의 읽기 요청에 기초하여서 읽기 명령을 생성한다.
- [0016] 상기 소정의 읽기 요청은 상기 제1 파티션에서의 쓰기 동작과 읽기 동작이 충돌하지 않는 제2 파티션에 해당하는 읽기 요청을 포함할 수 있다.
- [0017] 상기 읽기 명령의 생성에 사용되는 읽기 요청은 상기 제2 파티션에 해당하는 읽기 요청 중에서 가장 오래된 읽기 요청을 포함할 수 있다.
- [0018] 상기 소정의 읽기 요청은 읽기 동작을 위해 열린 워드 라인에 해당하는 읽기 요청을 포함할 수 있다.
- [0019] 상기 메모리에서 쓰기 동작이 진행되고 있지 않는 경우, 상기 요청 큐에서 가장 오래된 요청이 쓰기 요청이면, 상기 스케줄러는 가장 오래된 쓰기 요청과 상기 가장 오래된 쓰기 요청과 동시에 쓰기가 가능한 메모리 셀에 해당하는 쓰기 요청에 기초하여 상기 쓰기 명령을 생성할 수 있다.
- [0020] 본 발명의 또 다른 실시예에 따르면, 위에서 설명한 메모리 컨트롤러와 메모리를 포함하는 메모리 모듈이 제공된다.
- [0021] 본 발명의 또 다른 실시예에 따르면, 위에서 설명한 메모리 컨트롤러를 포함하는 프로세서가 제공된다. 상기 프로세서는 상기 메모리와 시스템 버스를 통해 연결되어 있다.

### 발명의 효과

- [0022] 본 발명의 한 실시예에 따르면, 쓰기 동작이 진행되는 동안 쓰기 동작과 충돌 없이 다른 파티션에서 읽기 동작이 동시에 수행될 수 있다.

### 도면의 간단한 설명

- [0023] 도 1은 상변화 메모리에서 하나의 메모리 셀을 개략적으로 나타내는 도면이다.
- 도 2는 도 1에 도시한 메모리 셀에 인가되는 전류를 나타내는 도면이다.
- 도 3은 도 1에 도시한 메모리 셀에 도 2에 도시한 전류가 인가될 때의 온도 변화를 나타내는 도면이다.
- 도 4는 본 발명의 한 실시예에 따른 메모리의 개략적인 블록도이다.

도 5는 본 발명의 한 실시예에 따른 메모리에서의 파티션의 한 예를 나타내는 도면이다.

도 6은 본 발명의 한 실시예에 따른 메모리에서 오버레이 윈도우 레지스터의 한 예를 나타내는 도면이다.

도 7 및 도 8은 각각 본 발명의 한 실시예에 따른 메모리 컨트롤러를 개략적으로 나타내는 도면이다.

도 9는 본 발명의 한 실시예에 따른 메모리 컨트롤러의 개략적인 블록도이다.

도 10a 및 도 10b는 각각 본 발명의 한 실시예에 따른 메모리 컨트롤러에서의 요청 스케줄링 방법을 나타내는 흐름도이다.

도 11은 본 발명의 또 다른 실시예에 따른 메모리 컨트롤러의 개략적인 블록도이다.

도 12는 본 발명의 한 실시예에 따른 메모리 컨트롤러를 포함하는 메모리 모듈을 개략적으로 나타내는 도면이다.

도 13은 본 발명의 한 실시예에 따른 메모리 컨트롤러를 포함하는 프로세서를 개략적으로 나타내는 도면이다.

### 발명을 실시하기 위한 구체적인 내용

- [0024] 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.
- [0025] 본 발명의 실시예에서는 PRAM을 상변화 메모리의 한 예로 설명하지만, 본 발명의 실시예는 PRAM에 한정되지 않고 다양한 상변화 메모리에도 적용될 수 있다.
- [0026] 먼저 도 1, 도 2 및 도 3을 참고로 하여 PRAM에서의 데이터 읽기/쓰기에 대해서 설명한다.
- [0027] 도 1은 상변화 메모리에서 하나의 메모리 셀을 개략적으로 나타내는 도면이며, 도 2는 도 1에 도시한 메모리 셀에 인가되는 전류를 나타내는 도면이고, 도 3은 도 1에 도시한 메모리 셀에 도 2에 도시한 전류가 인가될 때의 온도 변화를 나타내는 도면이다.
- [0028] 도 1에 도시한 메모리 셀은 하나의 예이며, 본 발명의 실시예에 따른 상변화 소자의 메모리 셀은 다양한 형태로 구현될 수 있다.
- [0029] 도 1을 참고하면, PRAM의 메모리 셀(100)은 상변화 소자(110)와 스위칭 소자(120)를 포함한다. 스위칭 소자(120)는 MOS 트랜지스터, 다이오드 등의 다양한 소자로 구현될 수 있다. 상변화 소자(110)는 상변화막(111), 상변화막 위에 형성되어 있는 상부 전극(112) 및 상변화막(111) 아래에 형성되어 있는 하부 전극(113)을 포함한다. 예를 들면, 상변화막(110)은 상변화 물질로 게르마늄(germanium, Ge), 안티몬(antimony, Sb) 및 텔루르(tellurium, Te)의 혼합물("GST 물질"이라 불리기도 함)을 포함할 수 있다.
- [0030] 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.
- [0031] 다시 도 1을 참고하면, 메모리 셀(100)에 전류가 인가되면, 인가된 전류는 하부 전극(113)을 통해 흐르게 된다. 짧은 시간 동안 전류가 메모리 셀(100)에 인가되면, 인가된 전류는 하부 전극(113)의 인접한 막을 가열한다. 이때, 가열 프로파일의 차이에 의해서 상변화막(111)의 일부(도 1에서 해칭된 부분)가 결정질 상태로 되거나 비정질 상태로 된다. 결정질 상태는 "셋(set) 상태"로, 비정질 상태로 "리셋(reset) 상태"로 불린다.
- [0032] 도 2 및 도 3을 참고하면, 높은 전류의 리셋 전류(210)가 짧은 시간( $t_{RST}$ ) 동안 메모리 셀(100)에 인가될 때, 상변화막(111)은 리셋 상태로 된다. 리셋 전류(210)의 인가에 따라 상변화막(111)의 상변화 물질이 가열되어 온도(310)가 녹는점 이상으로 되면, 상변화 물질이 녹은 후 식으면서 비정질 상태로 변화한다. 리셋 전류(210)보다 낮은 셋 전류(220)가 리셋 전류(210)보다 긴 시간( $t_{SET}$ ) 동안 상변화막(111)에 인가될 때, 상변화막(111)은 셋 상태로 된다. 셋 전류(220)의 인가에 따라, 상변화 물질이 가열되어서 온도(320) 녹는점보다 낮은 결정화 온도에 도달하면 결정질 상태로 변화한다. 셋 전류(220)보다 낮은 전류가 인가되거나 짧은 시간 동안 전류가 인가되는 경우에는 이러한 리셋 상태 및 셋 상태가 유지되므로, 메모리 셀(100)에 데이터를 기입할 수 있다.
- [0033] 이때, 리셋 상태와 셋 상태는 각각 "1"과 "0"의 데이터로 설정될 수 있으며, 이는 메모리 셀(100)의 상변화 소



자(110)의 저항율을 측정함으로써 감지될 수 있다. 이와는 달리, 리셋 상태와 셋 상태를 각각 "0"과 "1"의 데이터로 설정될 수도 있다.

[0034] 따라서 메모리 셀(100)에 읽기 전류(230)를 인가함으로써 메모리 셀(100)에 저장된 데이터를 읽을 수 있다. 읽기 전류(230)는 낮은 크기로 짧은 시간( $t_{READ}$ ) 동안 인가되어 메모리 셀(100)의 상태를 변경하지 않을 수 있다. 읽기 전류(230)는 크기(IR)가 셋 전류(220)보다 낮을 수 있고, 인가되는 시간( $t_{READ}$ )이 리셋 전류(210)의 인가 시간( $t_{RST}$ )보다 짧을 수 있다. 메모리 셀(100)의 상변화 소자(110)의 저항율이 상태에 따라 다르므로, 상변화 소자(110)에 흐르는 전류의 크기 또는 상변화 소자(110)에서의 전압 강하의 크기로 메모리 셀(100)의 상태, 즉 메모리 셀(100)에 저장된 데이터를 읽을 수 있다.

[0035] 한 실시예에서, 읽기 전류(230)를 인가하였을 때, 메모리 셀(100)에 걸리는 전압의 크기의 차이로 메모리 셀(100)의 상태를 읽을 수 있다. 이 경우, 리셋 상태에서 메모리 셀(100)의 상변화 소자(110)는 큰 저항을 가지므로, 상변화 소자(110)에서 감지된 전압인 큰 경우를 리셋 상태로, 상변화 소자(110)에서 감지된 전압이 작은 경우를 셋 상태로 결정할 수 있다. 다른 실시예에서, 메모리 셀(100)에 전압을 인가하였을 때, 출력되는 전류의 차이로 메모리 셀(100)의 상태를 읽을 수 있다. 이 경우, 상변화 소자(110)에서 감지된 전류가 작은 경우를 리셋 상태로, 상변화 소자(110)에서 감지된 전류가 큰 경우를 셋 상태로 결정할 수 있다.

[0036] 일반적으로, 복수의 메모리 셀(100)이 대략 행렬 형태로 배열되어 메모리를 형성하고, 동일한 행에서 여러 개의 열에 형성된 메모리 셀(100)에 동시에 데이터를 쓰고 있다. 따라서 여러 개의 열에 형성된 메모리 셀(100)에 데이터를 쓰기 위해서는 리셋 상태로 변경할 메모리 셀(100)에 리셋 전류(210)를 공급한 후에 셋 상태로 변경할 메모리 셀(100)에 셋 전류(220)를 공급할 수 있다. 이 경우, 데이터를 쓰기 위한 기입 시간( $t_{PGM}$ )은 리셋 전류(210)의 인가 시간( $t_{RST}$ )과 셋 전류(220)의 인가 시간( $t_{SET}$ )의 합에 해당하는 시간( $t_{RST} + t_{SET}$ )이 된다. 이와는 달리, 리셋 전류(210)와 셋 전류(220)를 동시에 인가하는 경우, 리셋 전류(210)의 인가 시간( $t_{RST}$ )과 셋 전류(220)의 인가 시간( $t_{SET}$ ) 중 긴 시간에 해당하는 시간(예를 들면,  $t_{SET}$ )이 기입 시간( $t_{PGM}$ )으로 된다.

[0037] 또한 리셋 전류(210)나 셋 전류(220)를 메모리 셀(100)에 인가하기 위해서 드라이버는 먼저 리셋 전류(210)나 셋 전류(220)를 흘리기 위한 전압으로 승압하고 승압된 전하를 충전하고 있어야 하므로, 리셋 전류(210)나 셋 전류(220)를 메모리 셀(100)에 인가하기 전에 전하 펌핑(charge pumping)을 위한 시간( $t_{CHG}$ )이 요구된다.

[0038] 다시 도 3을 참고하면, 메모리 셀(100)의 상변화 물질이 가열된 후에 식을 때까지 쿨링 시간이 더 필요하다. 상변화 물질이 식기 전에 메모리 셀(100)에서 데이터를 읽는 경우, 데이터가 정상적으로 읽히지 않을 수 있다. 따라서 읽기 전까지 쿨링 시간( $t_{COOL}$ )이 추가로 요구될 수 있다.

[0039] 따라서 데이터의 쓰기를 완료하는데 걸리는 쓰기 지연 시간( $t_{WRT}$ )은 수학적 1처럼 주어질 수 있다. 이러한 쓰기 지연 시간( $t_{WRT}$ )은 메모리 셀(100)로 데이터의 기입을 시작한 후에 해당 메모리 셀(100)이 데이터의 읽고 쓰기가 가능한 상태로 되는데 걸리는 시간일 수 있다.

## 수학적 1

[0040] 
$$t_{WRT} = t_{PGM} + \max(t_{CHG}, t_{COOL})$$

[0041] 도 4는 본 발명의 한 실시예에 따른 메모리의 개략적인 블록도이며, 도 5는 본 발명의 한 실시예에 따른 메모리에서의 파티션의 한 예를 나타내는 도면이다. 도 4에 도시한 메모리는 하나의 메모리 칩 또는 하나의 메모리 뱅크일 수 있다.

[0042] 도 4를 참고하면, 메모리(400)는 메모리 셀 어레이(410), 명령 버퍼(421), 행 주소 버퍼(422), 행 디코더(430), 센스 앰프(sense amplifier)(440), 행 데이터 버퍼(450), 데이터 입출력부(460) 및 기입 드라이버(470)를 포함한다.

[0043] 메모리 셀 어레이(410)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 1을 참고로 하여 설명한 메모리 셀(100)일 수 있다. 어떤 실시예에서, 메모리 셀 어레이(410)는 복수의 파티션(partition)으로 구분될 수 있다.

[0044] 예를 들면, 도 5에 도시한 것처럼, 메모리 셀 어레이(410)는 행 방향으로 2등분, 열 방향으로 4등분되어 8개의



파티션(PART0-PART7)으로 구분될 수 있다. 도 5에 도시한 파티션 방법은 하나의 예이며, 셀 어레이(410)는 다양한 개수의 파티션으로 구분될 수 있다. 예를 들면, 셀 어레이(410)는 행 방향으로 m등분, 열 방향으로 n등분되어  $n*m$ 개의 파티션으로 구분될 수 있다(여기서, m과 n은 1 이상의 정수). 또한 구분된 복수의 파티션의 크기가 서로 다를 수도 있다.

- [0045] 어떤 실시예에서, 복수의 파티션(PART0-PART7) 중 어느 하나의 파티션(예를 들면, PART0)에서 쓰기 동작이 진행되는 동안, 다른 파티션(예를 들면, PART1-PART7)에서는 읽기 동작이 파티션(PART0)의 쓰기 동작과 충돌 없이 진행될 수 있다. 즉, 파티션(PART0)에 대응하는 행 디코더(430)와 기입 드라이버(470)에 의해 파티션(PART0)에 쓰기 동작이 진행되는 동안, 다른 파티션(PART1-PART7) 중 어떤 파티션(예를 들면 PART1)에서는 해당 파티션(PART1)에 대응하는 행 디코더(430)와 센스 앰프(440)에 의해 읽기 동작이 진행될 수 있다.
- [0046] 한 실시예에서, 행 디코더(430)는 파티션별로 제공되고, 센스 앰프(440)와 기입 드라이버(470)는 적어도 일부 파티션, 예를 들면 모든 파티션(PART0-PART7)에 의해 공유될 수 있다.
- [0047] 명령 버퍼(421) 및 행 주소 버퍼(422)는 메모리 컨트롤러로부터 전달되는 명령 및 주소(특히, 행 주소)를 저장한다. 어떤 실시예에서, 복수의 행 주소 버퍼(422)가 제공될 수 있다. 한 실시예에서, 행 주소 버퍼(422)는 뱅크별로 제공될 수 있으며, 메모리 컨트롤러로부터 행 주소 버퍼(422)를 어드레싱하기 위한 뱅크 주소(예를 들면, 버퍼 번호)가 입력될 수 있다. 다른 실시예에서, 각 뱅크에 복수의 행 주소 버퍼(422)가 제공될 수 있으며, 각 행 주소 버퍼(422)는 뱅크 주소 또는 그 일부에 의해 어드레싱될 수 있다.
- [0048] 행 디코더(430)는 행 주소를 복호하여서 메모리 셀 어레이(410)의 복수의 워드 라인 중 데이터를 읽거나 데이터를 쓸 워드 라인을 선택한다.
- [0049] 센스 앰프(440)는 메모리 셀 어레이(410)에 저장된 데이터를 읽는 동작을 수행한다. 센스 앰프(440)는 행 디코더(430)가 선택한 워드 라인에 연결된 복수의 메모리 셀로부터 복수의 비트 라인을 통해서 데이터를 읽을 수 있다. 행 데이터 버퍼(450)는 센스 앰프(440)가 읽은 데이터를 저장한다. 어떤 실시예에서, 복수의 행 데이터 버퍼(450)가 제공될 수 있다. 한 실시예에서, 행 데이터 버퍼(450)는 뱅크별로 제공될 수 있으며, 메모리 컨트롤러로부터 행 데이터 버퍼(450)를 어드레싱하기 위한 뱅크 주소(예를 들면, 버퍼 번호)가 입력될 수 있다. 다른 실시예에서, 각 뱅크에 복수의 행 데이터 버퍼(450)가 제공될 수 있으며, 각 행 주소 버퍼(450)는 뱅크 주소 또는 그 일부에 의해 어드레싱될 수 있다.
- [0050] 데이터 입출력부(460)는 센스 앰프(440)가 읽어서 행 데이터 버퍼(450)에 저장된 데이터를 메모리 컨트롤러로 출력한다. 또한 데이터 입출력부(460)는 메모리 컨트롤러로부터 전달되는 데이터를 입력받아서 기입 드라이버(470)로 전달한다.
- [0051] 기입 드라이버(470)는 데이터 입출력부(460)에서 입력되는 데이터를 메모리 셀 어레이(410)에 쓴다. 기입 드라이버(470)는 행 디코더(430)가 선택한 워드 라인에 연결된 복수의 메모리 셀에 복수의 비트 라인을 통해서 데이터를 쓸 수 있다.
- [0052] 어떤 실시예에서 도 4에 도시한 것처럼 메모리(400)는 오버레이 윈도우 레지스터(480)와 프로그램 버퍼(490)를 더 포함할 수 있다. 오버레이 윈도우 레지스터(480)는 데이터 입출력부(460)를 통해 입력되는 데이터를 프로그램 버퍼(490)에 먼저 저장한 후에, 이를 다시 메모리 셀 어레이(410)에 쓸 수 있다. 어떤 실시예에서, 프로그램 버퍼로 쓰기 속도가 빠른 메모리, 예를 들면 SRAM(static random access memory)이 사용될 수 있다. 오버레이 윈도우 레지스터(480)는 데이터를 쓸 주소와 프로그램 버퍼(490)에서의 저장 위치를 매핑할 수 있다.
- [0053] 다음 도 6을 참고로 하여 오버레이 윈도우 레지스터(480)의 한 예에 대해서 설명한다.
- [0054] 도 6을 참고하면, 오버레이 윈도우 레지스터(480)는 명령 코드 레지스터(481), 명령 주소 레지스터(482), 명령 데이터 레지스터(483), 명령 실행 레지스터(484) 및 상태 레지스터(485)를 포함한다. 이러한 레지스터(481-485)는 메모리 매핑된 레지스터(memory-mapped register)로 구현될 수 있다.
- [0055] 명령 코드 레지스터(481)는 명령 코드가 기록되며, 명령 코드는 쓰기 명령이다. 명령 주소 레지스터(482)는 프로그램 버퍼(490)에 저장된 데이터가 쓰여질 메모리 셀 어레이(410)의 주소가 기록된다. 명령 데이터 레지스터(483)는 데이터를 쓰기 위한 쓰기 명령의 데이터가 기록된다. 프로그램 버퍼(490)에 데이터가 저장되고 레지스터(481, 482, 483)에 값이 기록되어서 오버레이 윈도우 레지스터(480)가 준비된 상태에서, 실제 데이터 쓰기를 실행하기 위한 값이 명령 실행 레지스터(484)에 기록된다.
- [0056] 상태 레지스터(485)는 메모리(400)의 상태를 지시하며, 메모리 셀 어레이(410)에 쓰기가 완료하는 경우 이 상태

를 지시할 수 있다. 따라서 메모리 컨트롤러는 상태 레지스터(485)의 상태를 확인해서 쓰기가 완료되었는지를 확인할 수 있다.

- [0057] 이제 본 발명의 실시예에 따른 메모리 컨트롤러에 대해서 설명한다.
- [0058] 도 7 및 도 8은 각각 본 발명의 한 실시예에 따른 메모리 컨트롤러를 개략적으로 나타내는 도면이다.
- [0059] 도 7을 참고하면, 메모리 컨트롤러(700)는 중앙 처리 유닛(central processing unit, CPU)(도시하지 않음)과 메모리 장치(800)에 연결되며, CPU로부터의 요청에 응답하여서 메모리 장치(800)에 액세스한다. 예를 들면, 메모리 컨트롤러(700)는 메모리 장치(800)의 읽기 또는 쓰기 동작을 제어할 수 있다. 어떤 실시예에서 메모리 장치(800)는 복수의 메모리 칩을 포함할 수 있다.
- [0060] 메모리 컨트롤러(700)는 메모리 컨트롤러 인터페이스(도시하지 않음)를 통해 CPU와 통신한다. 메모리 컨트롤러(700)는 메모리 컨트롤러 인터페이스를 통해 CPU로부터 읽기/쓰기 명령 및 어드레스를 수신하고, 데이터를 교환할 수 있다. 어떤 실시예에서, 메모리 컨트롤러 인터페이스는 시스템 버스를 수 있다. 시스템 버스는 예를 들면 FSB(front side bus), AXI(advanced extensible interface) 또는 아발론(Avalon) 버스 등일 수 있다.
- [0061] 또한 메모리 컨트롤러(700)는 메모리 장치(800)에 포함된 복수의 메모리 칩이 공통으로 연결된 버스("채널"이라고도 함)(710)를 통해 메모리 장치(800)와 통신할 수 있다. 메모리 컨트롤러(700)는 채널(710)을 통해 메모리 장치(800)에 읽기/쓰기 명령 및 어드레스를 전달하고, 데이터를 교환할 수 있다.
- [0062] 도 8을 참고하면, 어떤 실시예에서 복수의 채널(710a)에 각각 연결되는 복수의 메모리 장치(800a)가 제공될 수 있다. 이 경우, 메모리 컨트롤러(700a)는 복수의 채널에 각각 연결되는 복수의 채널 컨트롤러(701a)를 포함할 수 있다. 따라서 각 메모리 장치(800a)에 포함되는 복수의 메모리 칩은 대응하는 채널(710a)을 통해 대응하는 채널 컨트롤러(701a)와 통신할 수 있다.
- [0063] 도 9는 본 발명의 한 실시예에 따른 메모리 컨트롤러의 개략적인 블록도이다.
- [0064] 도 9를 참고하면, 메모리 컨트롤러(900)는 요청 큐(910), 스케줄러(920) 및 명령 큐(930)를 포함한다. 도 8을 참고로 하여 설명한 것처럼, 메모리 컨트롤러가 복수의 채널 컨트롤러를 포함하는 경우, 도 9에 도시한 메모리 컨트롤러(900)는 채널 컨트롤러에 대응할 수 있다.
- [0065] 요청 큐(910)는 CPU로부터 입력된 요청, 즉 쓰기 요청 및 읽기 요청을 저장한다. 어떤 실시예에서, 요청 큐(910)는 연결형 리스트(linked list) 또는 원형 버퍼(circular buffer)로 구현될 수 있다. 어떤 실시예에서, 요청 큐(910)는 읽기 요청을 저장하는 읽기 요청 큐와 쓰기 요청을 저장하는 쓰기 요청 큐를 포함할 수 있다.
- [0066] 스케줄러(920)는 요청 큐(910)에 저장된 요청, 즉 읽기 요청과 쓰기 요청에 따라 메모리 장치로부터 데이터를 읽기 위한 읽기 명령 및 메모리 장치에 데이터를 쓰기 위한 쓰기 명령을 생성하고, 읽기 요청 및 쓰기 요청에 대한 완료를 반환한다. 스케줄러(920)는 메모리 셀 어레이의 어떤 파티션에서 쓰기 동작이 수행되는 동안, 해당 파티션과 충돌이 일어나지 않는 파티션에서 읽기 동작이 수행되도록 쓰기 요청과 읽기 요청을 처리할 수 있다.
- [0067] 출력 명령 큐(930)는 스케줄러에서 생성된 읽기 명령 및 쓰기 명령을 저장한다.
- [0068] 도 10a 및 도 10b는 각각 본 발명의 한 실시예에 따른 메모리 컨트롤러에서의 요청 스케줄링 방법을 나타내는 흐름도이다.
- [0069] 도 10a를 참고하면, 스케줄러(920)는 출력 명령 큐(930)가 비었는지 확인한다(S1005). 한 실시예에서, 스케줄러(920)는 출력 명령 큐(930)가 완전히 비었는지를 확인할 수 있다. 다른 실시예에서, 스케줄러(920)는 출력 명령 큐(930)가 일정 수준 이상으로 비었는지를 확인할 수 있다.
- [0070] 출력 명령 큐(930)가 비었으면(S1005: 예), 스케줄러(920)는 새로운 명령 시퀀스를 생성하기 위해서 스케줄링 동작을 수행한다(S1010-S1070). 먼저, 스케줄러(920)는 요청 큐(910)가 비어 있는지 확인한다(S1010). 요청 큐(910)가 비어 있으면(S1010: 예), 스케줄러(920)는 다음 스케줄링을 위해서 다시 요청 큐(910)의 상태를 확인한다.
- [0071] 요청 큐(910)가 비어 있지 않으면(S1010: 아니요), 스케줄러(920)는 메모리 장치에서 쓰기 요청에 따라 쓰기 동작이 진행되고 있는지를 확인한다(S1020). 쓰기 동작이 진행되고 있지 않으면(S1020: 아니요), 스케줄러(920)는 요청 큐(910)에 입력된 요청 중 소정의 조건을 만족하는 요청이 읽기 요청인지 쓰기 요청인지 확인한다(S1030). 어떤 실시예에서, 스케줄러(920)는 요청 큐(910)에 입력된 요청 중에서 가장 오래된 요청이 읽기 요청인지 쓰기

요청인지 확인할 수 있다(S1030). 아래에서는 가장 오래된 요청을 조건 판단에 사용하는 것으로 가정하여서 설명한다.

[0072] 가장 오래된 요청이 쓰기 요청이면(S1030: 아니요), 스케줄러(920)는 가장 오래된 쓰기 요청을 선택하고, 선택한 쓰기 요청으로 쓰기 명령을 생성하여 출력 명령 큐(930)에 입력한다(S1040). 어떤 실시예에서, 스케줄러(920)는 선택한 쓰기 요청과 동시에 쓰기가 가능한 메모리 셀의 그룹을 타깃으로 하는 모든 쓰기 요청과 선택한 쓰기 요청으로 쓰기 명령을 생성할 수 있다(S1040). 이 경우, 메모리 셀 그룹에 대한 쓰기 요청을 동시에 처리할 수 있으므로, 메모리 셀 그룹의 데이터를 한번에 쓸 수 있어서 빈번한 메모리 접근에 따른 전력 소모 및 시간을 줄일 수 있다. 어떤 실시예에서, 이러한 메모리 셀 그룹을 "페이지(page)"라 부를 수 있다. 어떤 실시예에서, 페이지는 동시에 쓰기가 가능한 인접한 비트 라인에 위치한 메모리 셀의 그룹일 수 있다. 한 실시예에서, 동시에 쓰기가 가능한 인접한 비트 라인에 위치한 메모리 셀 그룹은 동일한 워드 라인을 공유할 수 있다. 어떤 실시예에서, 페이지의 크기는 도 4에 도시한 프로그램 버퍼(490)의 크기와 동일할 수 있다. 다른 실시예에서, 페이지의 크기는 도 4에 도시한 프로그램 버퍼(490)의 크기보다 크거나 작을 수 있다.

[0073] 쓰기 동작이 진행되고 있거나(S1020: 예), 가장 오래된 요청이 읽기 요청인 경우(S1030: 예), 스케줄러(920)는 파티션의 충돌을 확인하기 전에 메모리 셀 어레이(410)의 열린 행(open row)에 히트하는 읽기 요청이 있는지를 확인할 수 있다(S1050). 즉, 열린 행에 히트하는 읽기 요청에 대응하는 데이터가 행 데이터 버퍼(450)에 저장되어 있다면, 스케줄러(920)는 행 데이터 버퍼(450)로부터 데이터를 읽을 수 있다. 이를 위해, 메모리 셀 어레이(410)의 복수의 워드 라인 중에서 읽기 동작을 위해 열린 워드 라인이 있다면, 스케줄러(920)는 선택되어 있는 워드 라인에 연결된 메모리 셀에 해당하는 읽기 요청이 있는지를 확인할 수 있다(S1050). 열린 행에 히트하는 읽기 요청이 있다면(S1050: 예), 스케줄러(920)는 열린 행에 히트하는 읽기 요청을 선택하고, 선택한 읽기 요청으로 읽기 명령을 생성하여 출력 명령 큐(930)에 입력한다(S1055). 한 실시예에서, 스케줄러(920)는 열린 행에 히트하는 읽기 요청 중에서 가장 오래된 읽기 요청을 선택할 수 있다.

[0074] 열린 행에 히트하는 읽기 요청이 없다면(S1050: 아니요), 스케줄러(920)는 현재 쓰기 동작이 진행되고 있는 파티션과 읽기 동작이 충돌하지 않는 파티션에 해당하는 읽기 요청이 있는지 확인한다(S1060). 예를 들면, 도 5에 도시한 예에서, 파티션(PART0)에 쓰기 동작이 진행되고 있는 동안, 파티션(PART1-PART7)에서 읽기 동작은 쓰기 동작과 충돌 없이 진행될 수 있다. 현재 쓰기 동작이 진행되고 있는 파티션과 읽기 동작이 충돌하지 않는 파티션에 해당하는 읽기 요청이 있으면, 스케줄러(920)는 해당 읽기 요청을 선택하고, 선택한 읽기 요청으로 읽기 명령을 생성하여 출력 명령 큐(930)에 입력한다(S1070). 현재 쓰기 요청이 처리되고 있는 파티션과 충돌이 없는 파티션에 해당하는 읽기 요청이 없다면, 스케줄러(920)는 스케줄링을 수행할 수 없으며, 쓰기 동작이 완료되거나 다른 읽기 요청이 요청 큐(910)에 입력될 때까지 대기할 수 있다.

[0075] 어떤 실시예에서, 메모리(400)에 복수의 행 데이터 버퍼(450)가 제공되는 경우, 스케줄러(920)는 읽기 요청을 처리할 때 행 데이터 버퍼(450)를 선택할 수 있다(S1070). 한 실시예에서, 스케줄러(920)는 복수의 행 데이터 버퍼(450) 중에서 읽기 요청에 해당하는 데이터에 히트하는 행 데이터 버퍼(450)가 존재하면, 히트하는 행 데이터 버퍼(450)를 선택할 수 있다. 그러면 스케줄러(920)는 히트하는 행 데이터 버퍼(450)에 저장된 데이터를 읽을 수 있다. 다른 실시예에서, 스케줄러(920)는 복수의 행 데이터 버퍼(450) 중에서 읽기 요청에 해당하는 데이터에 히트하는 행 데이터 버퍼(450)가 존재하지 않으면, 가장 오래된 데이터가 저장되어 있는 행 데이터 버퍼(450)를 선택할 수 있다. 그러면 스케줄러(920)는 선택한 행 데이터 버퍼(450)에서 가장 오래된 데이터를 버리고, 메모리 셀 어레이(410)에서 읽은 데이터를 선택한 데이터 버퍼(450)에 저장할 수 있다.

[0076] 어떤 실시예에서, 도 10b에 도시한 것처럼, 쓰기 동작이 진행되고 있거나(S1020: 예), 가장 오래된 요청이 읽기 요청인 경우(S1030: 예), 스케줄러(920)는 열린 행에 히트하는 읽기 요청이 있는지를 확인하는 과정(S1050) 없이, 파티션 충돌을 확인할 수 있다(S1060).

[0077] 도 10a 및 도 10b에서는 쓰기 동작이 진행되고 있는 경우(S1020: 예)와 쓰기 동작이 진행되지 않고 가장 오래된 요청이 읽기 요청인 경우(S1030: 예)에 단계 S1060의 처리가 수행되는 것으로 도시하였지만, 어떤 실시예에서, 쓰기 동작이 진행되고 있지 않고 가장 오래된 요청이 읽기 요청인 경우(S1030: 예)에는 파티션의 충돌 여부와는 관계없이 요청 큐(910)에 읽기 요청이 존재하는지만을 확인하고(S1060), 해당하는 읽기 요청으로 읽기 명령을 생성할 수 있다(S1070). 어떤 실시예에서, 선택된 읽기 요청은 가장 오래된 읽기 요청일 수 있다.

[0078] 스케줄러는 이와 같이 읽기 요청과 쓰기 요청을 스케줄링하는 과정(S1005-S1070)을 반복할 수 있다.

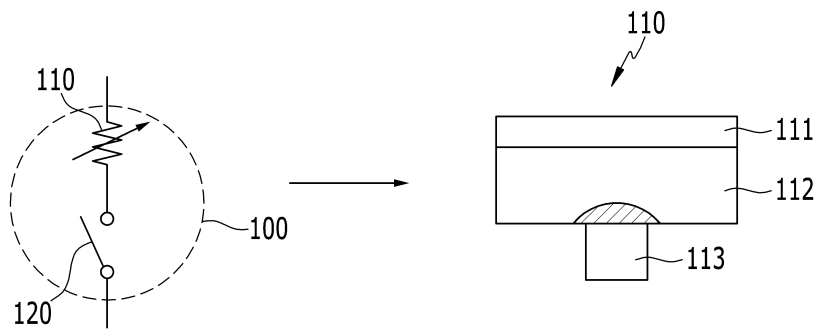
[0079] 이와 같이, 본 발명의 한 실시예에 따르면, 쓰기 요청에 따라 쓰기 동작이 진행되는 동안 쓰기 동작과 충돌 없

이 다른 파티션에서 읽기 동작이 동시에 수행될 수 있다. 즉, 파티션을 고려한 스케줄링이 가능하다. 예를 들면, 메모리 셀 어레이(도 4의 410)가 8개의 파티션으로 구분되고 파티션들 사이에 충돌이 없다면, 특정파티션에 쓰기 동작이 진행되는 경우, 7/8의 확률로 읽기 요청이 처리될 수 있다.

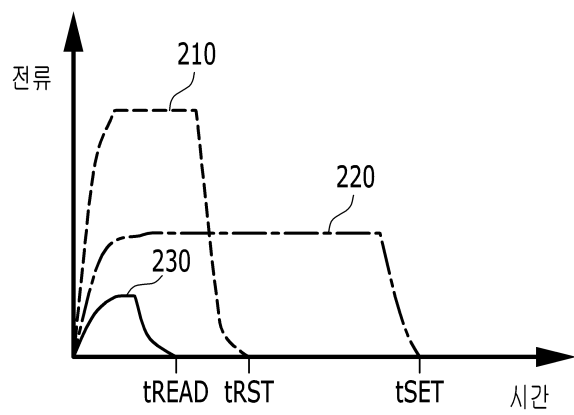
- [0080] 도 1 내지 도 3을 참고로 하여 설명한 것처럼 쓰기 시간이 읽기 시간보다 훨씬 길기 때문에, 어떤 실시예에서 요청 큐(도 9의 910)의 깊이(depth)를 깊게 할 수 있다. 그러면 많은 읽기 요청이 요청 큐(910)에 입력되어 대기하고 있으므로 쓰기 동작 동안 많은 읽기 요청이 처리될 수 있다.
- [0081] 도 11은 본 발명의 다른 실시예에 따른 메모리 컨트롤러의 개략적인 블록도이다.
- [0082] 도 11을 참고하면, 메모리 컨트롤러(1100)는 주소 매핑기(1110), 복수의 랭크 컨트롤러(1120), 중재기(arbiter)(1130) 및 명령 순서화기(1140)를 포함한다.
- [0083] 메모리 장치가 복수의 랭크를 포함할 수 있다. 어떤 실시예에서, 랭크는 공유하는 채널을 통해 독립적으로 접근이 가능한 메모리 칩의 집합일 수 있다.
- [0084] 복수의 랭크는 독립적으로 동작할 수 있으며, 명령, 주소 및 데이터를 위한 채널을 공유할 수 있다. 이 경우, 메모리 컨트롤러는 복수의 랭크에 각각 대응하는 복수의 랭크 컨트롤러(1120)를 포함할 수 있다. 각 랭크 컨트롤러(1120)는 도 7 내지 도 10을 참고로 하여 설명한 메모리 컨트롤러처럼 형성될 수 있다.
- [0085] 주소 매핑기(1110)는 CPU로부터의 명령(쓰기 요청 또는 읽기 요청), 주소 및 데이터를 복수의 랭크 중에서 주소에 해당하는 랭크에 대응하는 랭크 컨트롤러로 매핑한다.
- [0086] 중재기(1130)는 채널 상태를 참조해서 채널에 대한 접근을 중재한다. 중재기(1130)는 복수의 랭크 컨트롤러(1120)로부터의 명령에 대한 타이밍을 조절할 수 있다. 어떤 실시예에서, 중재기(1130)는 열 주소에 대한 행 주소 지연(row address to column address delay), 전하 펌핑 시간 또는 CAS(column address strobe) 지연 시간 등을 고려할 수 있다.
- [0087] 어떤 실시예에서, 중재기(1130)는 채널에 대한 접근을 중재하기 위한 정책으로 라운드 로빈(round robin) 방식 또는 우선 순위 기반(priority-based) 방식 등을 사용할 수 있다.
- [0088] 어떤 실시예에서, 메모리 컨트롤러는 별도의 칩(컨트롤러)이거나 다른 칩(컨트롤러)에 통합될 수 있다. 예를 들면, 메모리 컨트롤러는 CPU와 메모리 장치와 같은 마더보드(motherboard)의 다른 부분 사이의 통신을 관리하는 노스브릿지(northbridge)에 통합될 수 있다.
- [0089] 어떤 실시예에서, 메모리 컨트롤러(1210)는 도 12에 도시한 것처럼 메모리 장치(1220)와 함께 메모리 모듈(1200)에 통합될 수 있다. 어떤 실시예에서 메모리 시스템(1200)은 복수의 메모리 칩이 통합되어 있는 메모리 모듈일 수 있다. 한 실시예에서 메모리 모듈은 DIMM(dual in-line memory module)일 수 있다.
- [0090] 어떤 실시예에서, 메모리 컨트롤러(1311)는 도 13에 도시한 것처럼 CPU 등의 프로세서(1310)에 통합될 수 있다. 한 실시예에서, 메모리 컨트롤러(1311)와 프로세서(1310)는 시스템 버스(도시하지 않음)에 연결되고, 메모리 컨트롤러(1311)는 메모리 장치(1320)에 버스(채널)(1330)을 통해 연결될 수 있다.
- [0091] 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

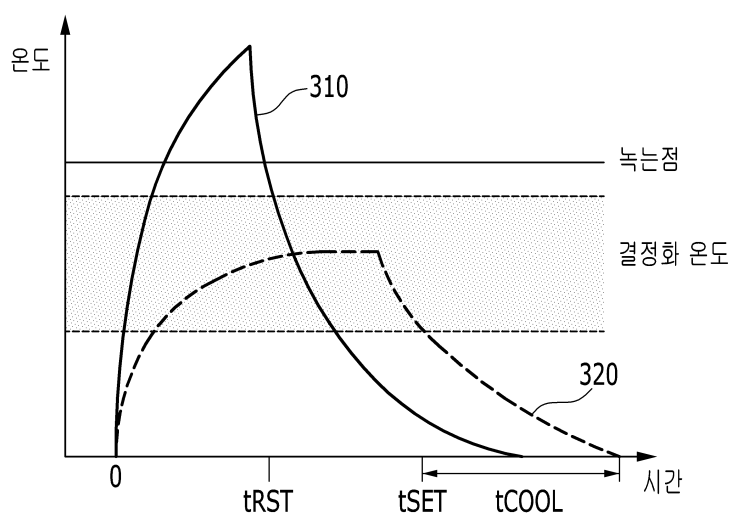
도면1



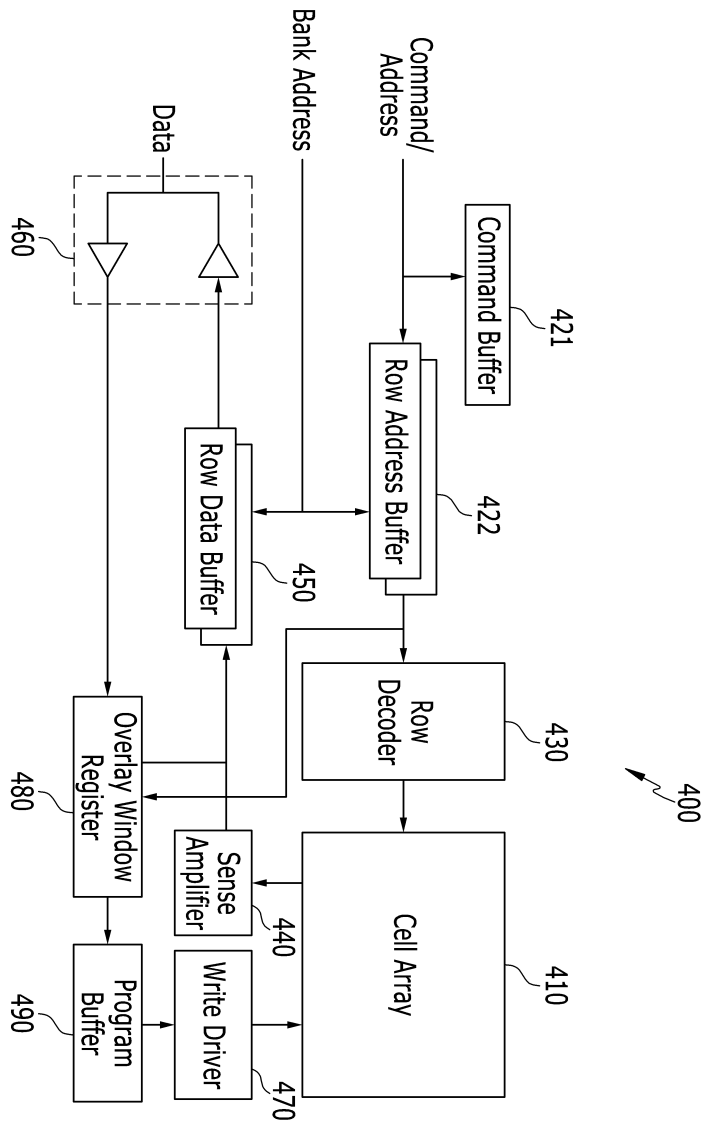
도면2



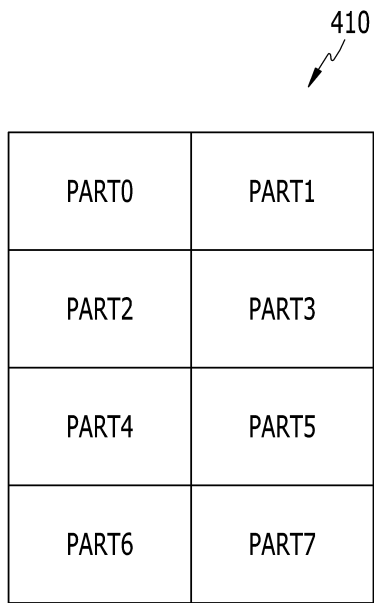
도면3



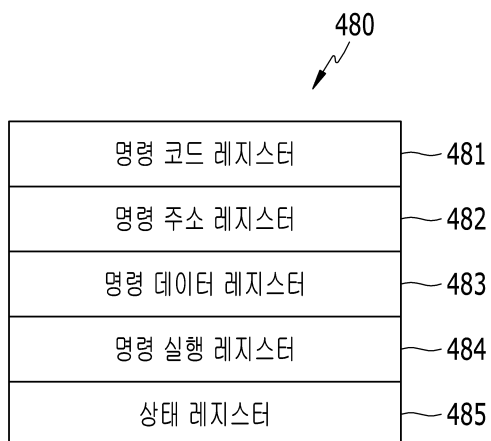
도면4



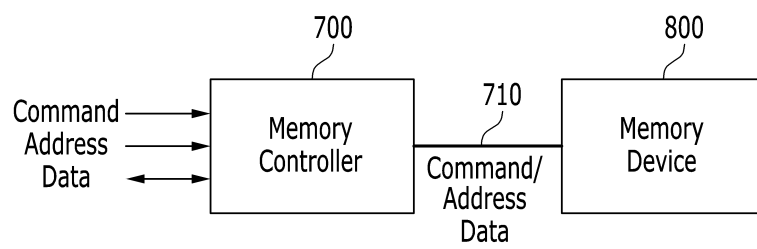
도면5



도면6

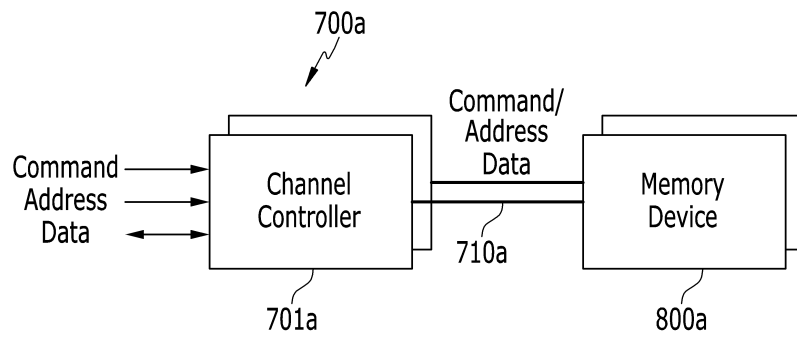


도면7

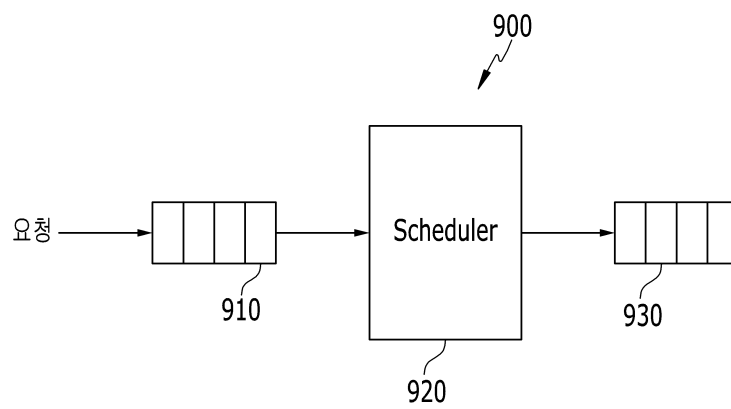




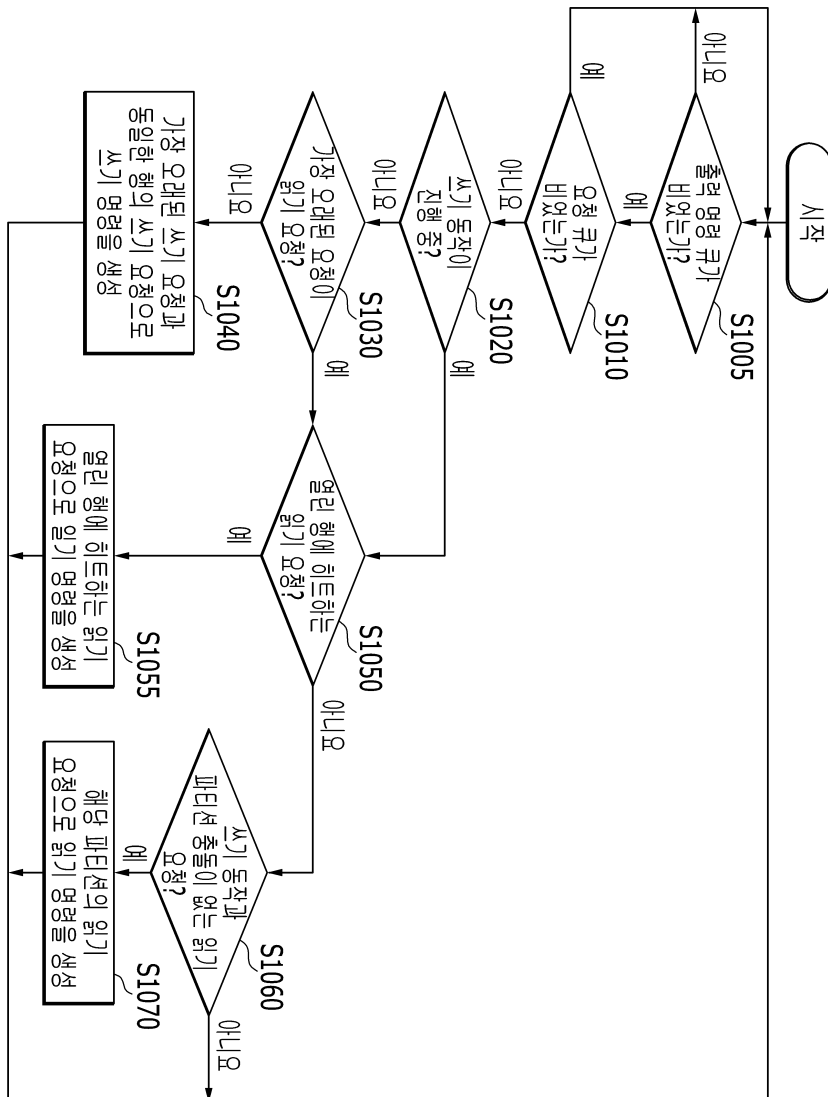
도면8



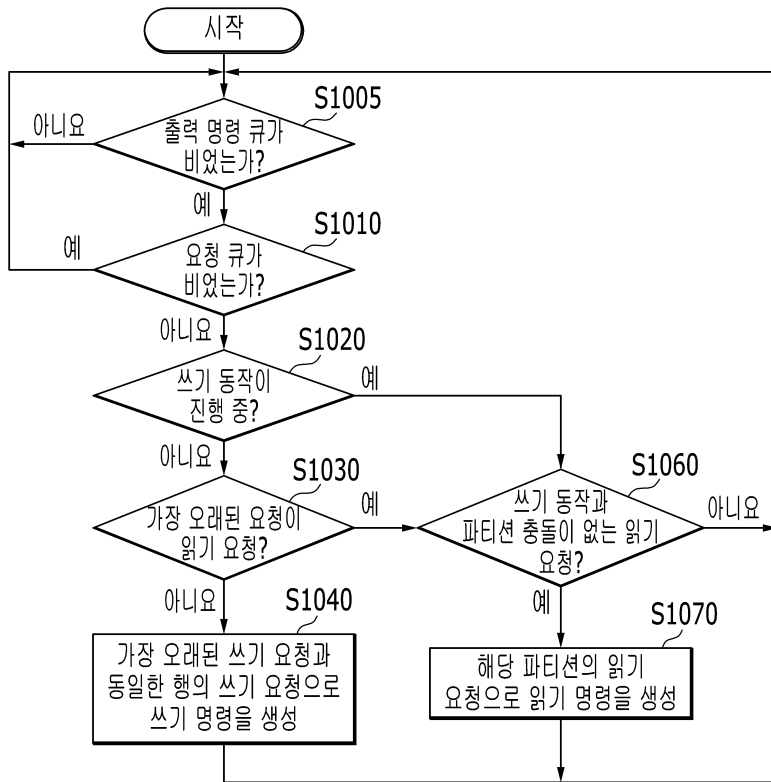
도면9



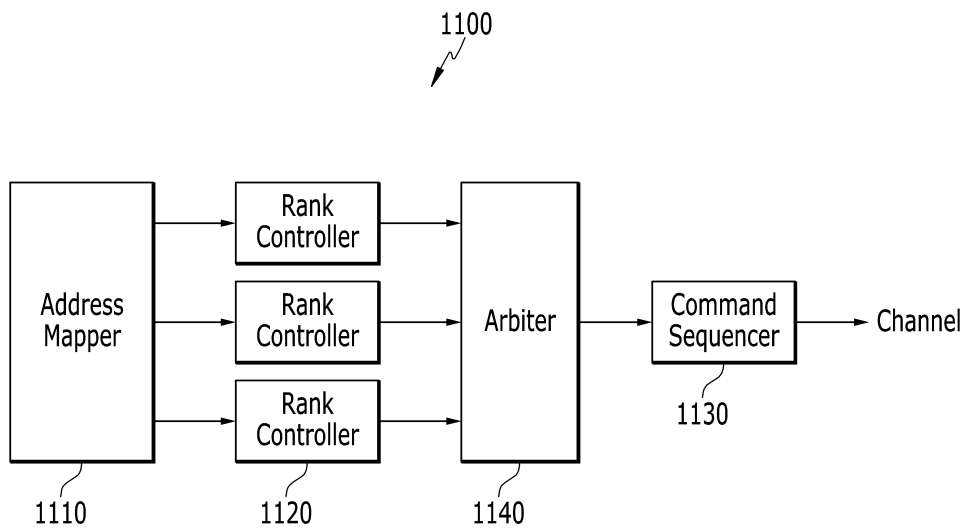
도면10a



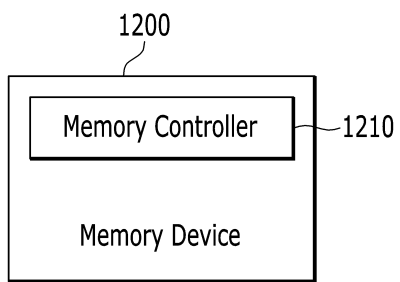
도면10b



도면11



도면12



도면13

