(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0136767
(43) 공개일자 2020년12월08일

(51) 국제특허분류(Int. Cl.)
*G11C 11/406* (2006.01)
(52) CPC특허분류
*G11C 11/40615* (2013.01)
*G11C 2211/4061* (2013.01)
(21) 출원번호 10-2019-0062777
(22) 출원일자 2019년05월28일
심사청구일자 없음

(71) 출원인
**서울대학교산학협력단**
서울특별시 관악구 관악로 1 (신림동)
**경희대학교 산학협력단**
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
**이혁재**
경기도 성남시 분당구 서판교로 147 ,1106동103호(판교동,판교원마을)
**김현**
서울특별시 노원구 공릉로 232 서울과학기술대학교 미래관 526호
*(뒷면에 계속)*
(74) 대리인
**특허법인한남**

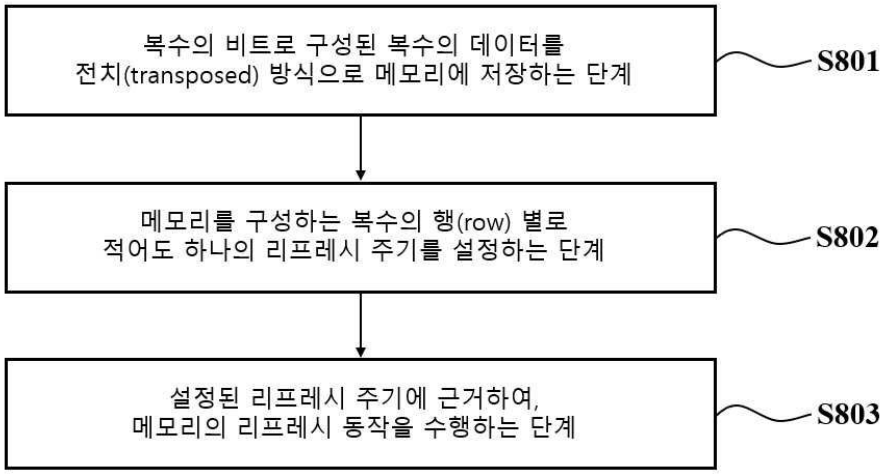
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 **근사적 메모리 아키텍처 및 그를 이용하는 데이터 처리장치**

## (57) 요 약

본 발명에 따른 디램 디바이스의 제어방법은, 복수의 비트로 구성된 복수의 데이터를 전치(transposed) 방식으로 메모리에 저장하는 단계와, 상기 메모리를 구성하는 복수의 행(row) 별로 적어도 하나의 리프레시 주기를 설정하는 단계 및 설정된 리프레시 주기에 근거하여, 상기 메모리의 리프레시 동작을 수행하는 단계를 포함하는 것을 특징으로 한다.

*대 표 도* - 도8



(72) 발명자

**응우옌 두이탄**

서울특시 관악구 관악로 1 905동 246호 A

**김보열**

서울특별시 동작구 동작대로29길 91, 202동 1005호

**장익준**

경기도 용인시 덕영대로 1732 경희대학교국제캠퍼스

# 명 세 서

## 청구범위

### 청구항 1

복수의 비트로 구성된 복수의 데이터를 전치(transposed) 방식으로 메모리에 저장하는 단계;

상기 메모리를 구성하는 복수의 행(row) 별로 적어도 하나의 리프레시 주기를 설정하는 단계;

설정된 리프레시 주기에 근거하여, 상기 메모리의 리프레시 동작을 수행하는 단계를 포함하는 것을 특징으로 하는 디램 디바이스의 제어방법.

### 청구항 2

제1항에 있어서,

상기 데이터를 전치 방식으로 메모리에 저장하는 단계는,

상기 메모리를 구성하는 복수의 행(Row)에, 상기 복수의 데이터의 최상위비트(MSB)부터 최하위비트(LSB)까지 순차적으로 저장하는 과정을 포함하는 것을 특징으로 하는 디램 디바이스의 제어방법.

### 청구항 3

제1항에 있어서,

상기 데이터를 구성하는 복수의 비트는, 부호 비트와, 지수 비트 및 가수 비트를 포함하고,

상기 데이터를 전치 방식으로 메모리에 저장하는 단계는,

상기 복수의 데이터 각각의 부호 비트 및 지수 비트를 상기 메모리의 일부 행에 저장하는 과정과,

상기 복수의 데이터 각각의 가수 비트를 상기 메모리의 나머지 행에 저장하는 과정을 포함하는 것을 특징으로 하는 디램 디바이스의 제어방법.

### 청구항 4

제3항에 있어서,

상기 데이터는,

1비트의 부호 비트(sign bit)와, 8비트의 지수 비트(exponent bits)와, 23비트의 가수 비트(mantissa bits)로 구성되고,

상기 메모리의 제1 행에는,

상기 복수의 데이터의 부호 비트들이 저장되고,

상기 메모리의 제2 행 내지 제9 행에는, 상기 복수의 데이터의 지수 비트들이 저장되고,

상기 메모리의 제10 행 내지 제32행에는, 상기 복수의 데이터의 가수 비트들이 저장되는 것을 특징으로 하는 디램 디바이스의 제어방법.

### 청구항 5

제4항에 있어서,

상기 리프레시 주기를 설정하는 단계는,

상기 메모리의 복수의 행 중 상기 부호 비트 또는 상기 지수 비트가 저장되는 일부의 행에 대해, 제1 주기 값을 설정하는 과정과,

상기 메모리의 복수의 행 중 상기 가수 비트가 저장되는, 나머지의 행에 대해, 제2 주기 값을 설정하는 과정을

포함하는 것을 특징으로 하는 디램 디바이스의 제어방법.

**청구항 6**

제5항에 있어서,

상기 제2 주기 값은 상기 제1 주기 값보다 길게 설정되는 것을 특징으로 디램 디바이스의 제어방법.

**청구항 7**

제6항에 있어서,

상기 리프레시 동작을 수행하는 단계는,

상기 제1 주기마다 카운터를 증가시키는 과정과,

상기 제1 주기마다 상기 카운터에 근거하여, 상기 메모리의 각 행마다 리프레시의 수행 여부를 결정하는 과정을 포함하는 것을 특징으로 하는 디램 디바이스의 제어방법.

**청구항 8**

제1항에 있어서,

상기 리프레시 동작을 수행하는 단계는,

미리 설정된 리프레시 주기마다 카운터를 증가시키는 과정과,

상기 카운터의 값과, 상기 복수의 행 별로 설정된 리프레시 주기에 근거하여, 상기 행 별로 리프레시의 수행 여부를 결정하는 과정을 포함하는 것을 특징으로 하는 디램 디바이스의 제어 방법.

**청구항 9**

제8항에 있어서,

상기 리프레시 동작을 수행하는 단계는,

상기 복수의 행 중 일부분에 대해, 상기 카운터가 증가될 때마다 리프레시 동작을 수행하는 과정과,

상기 복수의 행 중 나머지 부분에 대해, 상기 카운터가 소정의 값에 대응될 때, 리프레시 동작을 수행하는 과정을 포함하는 것을 특징으로 하는 디램 디바이스의 제어 방법.

**청구항 10**

딥 러닝 애플리케이션을 동작시키는 프로세서;

상기 딥 러닝 애플리케이션과 관련된 데이터를 저장하는 메모리 디바이스; 및

상기 메모리 디바이스의 데이터 저장 방식을 제어하는 메모리 컨트롤러;를 포함하고,

상기 메모리 디바이스는,

복수의 행으로 구성되며, 복수의 데이터를 전치(transposed) 방식으로 저장하고,

상기 복수의 행 중 적어도 일부에 대해, 미리 설정된 제1 주기마다 리프레시 동작을 수행하고, 나머지 일부에 대해, 상기 제1 주기와 상이한 제2 주기마다 리프레시 동작을 수행하도록 구성되는 것을 특징으로 하는 데이터 처리 장치.

**청구항 11**

제10항에 있어서,

상기 메모리 컨트롤러는,

상기 복수의 데이터로부터 추출한 복수의 부호 비트를 상기 메모리 디바이스의 제1 어드레스 영역에 저장시키고,

상기 복수의 데이터로부터 추출한 복수의 지수 비트를 상기 메모리 디바이스의 제2 어드레스 영역에 저장시키고,

상기 복수의 데이터로부터 추출한 복수의 가수 비트를 상기 메모리 디바이스의 제3 어드레스 영역에 저장시키는 것을 특징으로 하는 데이터 처리 장치.

**청구항 12**

제11항에 있어서,

상기 복수의 데이터가 32-비트 플로팅 포인트 데이터인 경우, 상기 제1 어드레스 영역은 상기 메모리 디바이스의 제1 행에 대응되고, 상기 제2 어드레스 영역은 상기 메모리 디바이스의 제2 행 내지 제9 행에 대응되며, 상기 제3 어드레스 영역은 상기 메모리 디바이스의 제10 행 내지 제32 행에 대응되는 것을 특징으로 하는 데이터 처리 장치.

**청구항 13**

제10항에 있어서,

상기 메모리 디바이스는,

상기 복수의 데이터가 전치 방식으로 저장된 후, 미리 설정된 주기마다 리프레시 동작을 수행하되, 상기 복수의 행 중 적어도 일부에 대해 상기 리프레시 동작을 스킵하도록 구성되는 것을 특징으로 하는 데이터 처리 장치.

**청구항 14**

제13항에 있어서,

상기 메모리 디바이스는,

상기 리프레시 동작이 개시되면, 상기 주기마다 소정의 카운터 변수를 증가시키고,

상기 복수의 행 중 적어도 일부에 대해, 증가된 상기 카운터 변수에 근거하여 리프레시의 수행 여부를 결정하는 것을 특징으로 하는 데이터 처리 장치.

**청구항 15**

제13항에 있어서,

상기 복수의 행 중 적어도 일부에는, 상기 복수의 데이터의 가수 비트가 저장되는 것을 특징으로 하는 데이터 처리 장치.

**청구항 16**

제15항에 있어서,

상기 메모리 디바이스는,

상기 복수의 데이터의 가수 비트가 저장된 적어도 하나의 행과, 상기 복수의 데이터의 부호 비트 또는 지수 비트가 저장된 적어도 하나의 행에 대해, 서로 다른 주기로 리프레시 동작을 수행하는 것을 특징으로 하는 데이터 처리 장치.

**청구항 17**

제16항에 있어서,

상기 메모리 디바이스는,

상기 부호 비트 또는 지수비트가 저장된 적어도 하나의 행에 대해, 미리 설정된 주기로 리프레시 동작을 수행하고,

상기 리프레시 동작이 복수 회 수행되는 동안, 상기 가수 비트가 저장된 적어도 하나의 행에 대해, 리프레시 동작을 적어도 일 회 이상 스킵하는 것을 특징으로 하는 데이터 처리 장치.

## 발명의 설명

### 기 술 분 야

[0001] 본 발명은 메모리 아키텍처와 관련된 것이며, 특히, 딥 러닝 애플리케이션에 이용되는 메모리 아키텍처와 관련된 것이다.

### 배 경 기 술

[0002] 디램(Dynamic Random-Access Memory, DRAM) 장치는, 데이터 무결성(Data integrity)을 유지하기 위하여 정기적으로 리프레시 동작을 수행하며, 이에 따라 상당한 전력 소비가 야기된다.

[0003] 리프레시(Refresh) 동작이 지연되면, 디램 셀에 저장된 데이터가 손실될 수 있고, 이는 손실된 데이터를 사용하는 계산의 정확성을 감소시킨다. 따라서, 일반적인 디램 장치는 리프레시 동작을 일정한 주기로 수행하도록 구성된다.

[0004] 한편, 딥 러닝 애플리케이션은, 데이터를 저장하기 위해 디램과 같은 메모리 디바이스를 이용한다.

[0005] 딥 러닝 애플리케이션은 메모리의 과도한 데이터 트래픽으로 많은 양의 계산을 필요로 한다. 특히, 딥 러닝 알고리즘의 복잡성은, 신경 네트워크(Neural Network)의 성능 향상을 위해 그 계층의 숫자가 증가할수록, 계속 증대된다.

[0006] 예를 들어, 딥 러닝 알고리즘인 AlexNetproposed(2012년)는 8개의 레이어를 사용하지만, 다른 딥 러닝 알고리즘인 ResNet(2015년)에 152개의 레이어로 구현된다.

[0007] 이러한 추세에 따라 메모리 대역폭은 딥 러닝 애플리케이션의 속도 향상을 제한하는 원인 중 하나가 될 수 있다. 따라서 메모리 액세스의 양을 줄이기 위한 방대한 연구가 수행되었다.

[0008] 상술한 바와 같이, 디램 셀은 저장된 데이터를 영구히 유지할 수 없으므로, 디램에 저장된 데이터를 유지하기 위해서는, 주기적으로 저장된 데이터를 읽은 다음 동일한 메모리 셀에 다시 기록하는 리프레시 동작이 수행되어야 한다.

[0009] 이러한 리프레시 동작은 중요한 데이터를 저장하는지 여부에 관계없이 디램의 모든 셀에 필요하다. 따라서, 디램의 특정 셀이, 프로세서에서 활성 프로세스에 의해 액세스되는 데이터를 저장하지 않더라도, 이 리프레시 동작은 상당한 전력 소비를 초래한다.

[0010] 한편, 디램의 밀도가 증가하는 경우에는, 상기 리프레시 동작에 의한 전력 소모도 증가한다. 즉, 디램의 밀도에 비례하여 재생 전력 소모량과 디램 총 소비 전력량의 비율이 증가한다.

[0011] 이에 따라, 향후 64 기가바이트 용량의 디램에서, 리프레시 동작은 전체 전력 소비의 50%까지 차지할 것으로 예상된다. 따라서, 용량 증가에 따라 추가적으로 발생하는 전력 소비는 컴퓨팅 시스템 설계에서 중요한 매개 변수 중 하나로 고려되어야 한다.

[0012] 종래에, 리프레시 동작에 의한 전력 소비를 줄이기 위해 다양한 방법이 제시된 바 있다. 기본적으로, 상기 방법들은 각 디램 행(Row)의 잔여 시간 정보(Retention time info)를 요구한다. 디램 컨트롤러는, 상기 잔여 시간 정보를 사용하여 각 행에 대해 선택적으로 리프레시 동작을 수행한다.

[0013] 상기 방법에 따르면, 불필요한 리프레시 동작을 생략할 수 있으므로, 디램의 리프레시 동작에 의한 전력 소비를 감소시킬 수 있다.

[0014] 그러나, 계속적으로 증가하는 디램의 모든 행의 잔여 시간 정보를 저장하는 것은 효율적이지 않으며, 확장성이 결여되는 단점이 있다.

[0015] 또한, 모든 디램 셀의 잔여 시간 정보를 프로파일링하는 것은 상당한 시간을 필요로할 뿐 아니라, VRT(Variable Retention Time) 및 DPD(Data Pattern Dependencies)를 처리해야하므로, 리프레시 동작을 생략하는 과정에서 오류가 발생할 수 있다.

[0016] 따라서, 디램의 리프레시 동작에 의한 전력 소비를 감소시키는 다른 방법으로써, 디램 셀에서 약간의 오류 발생 가능성을 허용하는 접근법을 고려해볼 수 있다.

[0017] 특히, 딥 러닝 애플리케이션에 이용되는 디램의 경우, 리프레시 동작을 생략하는 과정에서 발생할 수 있는 오류

가 딥 러닝의 계산의 정밀도를 약간 떨어뜨릴지라도, 딥 러닝의 신경망(Neural Network)의 정확성을 심각하게 저하시키지 않는다.

[0018] 종래 연구에 따르면, 딥 러닝의 계산의 정밀도가 향상되는 것은, 학습 성능의 향상을 달성하는데 있어서 무의미한 것으로 나타났다.

## 발명의 내용

### 해결하려는 과제

[0019] 상술한 문제점을 해결하기 위한 본 발명의 기술적 과제는 리프레시 동작에 의한 전력 소비를 최소화할 수 있는 디램 아키텍처 및 그를 포함하는 데이터 처리 장치를 제공하는 것이다.

[0020] 또한, 본 발명의 목적은, 딥 러닝 애플리케이션의 오류 허용성을 이용하여, 디램의 리프레시 동작 횟수를 감소시키는 디램 아키텍처 및 그를 포함하는 데이터 처리 장치를 제공하는 것이다.

[0021] 또한, 본 발명의 목적은, 디램의 행 또는 셀 별로 리프레시 동작 주기를 가변적으로 설정할 수 있는 디램 아키텍처 및 그를 포함하는 데이터 처리 장치를 제공하는 것이다.

[0022] 또한, 본 발명의 목적은, 데이터를 전치(transposed) 방식으로 저장시킴으로써, 데이터 중요도에 따라 서로 다른 리프레시 동작 주기를 적용시킬 수 있는 디램 아키텍처 및 그를 포함하는 데이터 처리 장치를 제공하는 것이다.

### 과제의 해결 수단

[0023] 위와 같은 기술적 과제를 해결하기 위해, 본 발명에서는 오류 가능성을 허용하는 근사적 디램 구조를 제안한다.

[0024] 구체적으로, 본 발명에서는 디램의 셀의 재생 빈도를 가변적으로 제어함으로써, 저장 오류를 허용하는 근사 메모리 구조를 제안한다.

[0025] 또한, 본 발명에서는 디램 장치에서 수행되는 리프레시 동작의 빈도를 감소시킴으로써, 상기 리프레시 동작에 의한 전력 소비를 감소시킬 수 있는 새로운 메모리 아키텍처를 제안한다.

[0026] 또한, 본 발명에서는 상술한 데이터 손실의 문제를 해결하기 위해, 딥 러닝 애플리케이션의 오류 허용 특성(error-tolerant property)을 이용하는 메모리 아키텍처를 제안한다. 즉, 딥 러닝 애플리케이션이 소량의 오류에 대해 내성을 갖는 특징을 이용하여, 데이터 손실에 의한 문제를 회피하고자 한다.

[0027] 구체적으로, 본 발명에 따른 디램 디바이스의 제어방법은, 복수의 비트로 구성된 복수의 데이터를 전치(transposed) 방식으로 메모리에 저장하는 단계와, 상기 메모리를 구성하는 복수의 행(row) 별로 적어도 하나의 리프레시 주기를 설정하는 단계 및 설정된 리프레시 주기에 근거하여, 상기 메모리의 리프레시 동작을 수행하는 단계를 포함하는 것을 특징으로 한다.

[0028] 일 실시예에 있어서, 상기 데이터를 전치 방식으로 메모리에 저장하는 단계는, 상기 메모리를 구성하는 복수의 행(Row)에, 상기 복수의 데이터의 최상위비트(MSB)부터 최하위비트(LSB)까지 순차적으로 저장하는 과정을 포함하는 것을 특징으로 한다.

[0029] 일 실시예에 있어서, 상기 데이터를 구성하는 복수의 비트는, 부호 비트와, 지수 비트 및 가수 비트를 포함하고, 상기 데이터를 전치 방식으로 메모리에 저장하는 단계는, 상기 복수의 데이터 각각의 부호 비트 및 지수 비트를 상기 메모리의 일부 행에 저장하는 과정과, 상기 복수의 데이터 각각의 가수 비트를 상기 메모리의 나머지 행에 저장하는 과정을 포함하는 것을 특징으로 한다.

[0030] 일 실시예에 있어서, 상기 데이터는, 1비트의 부호 비트(sign bit)와, 8비트의 지수 비트(exponent bits)와, 23비트의 가수 비트(mantissa bits)로 구성되고, 상기 메모리의 제1 행에는, 상기 복수의 데이터의 부호 비트들이 저장되고, 상기 메모리의 제2 행 내지 제9 행에는, 상기 복수의 데이터의 지수 비트들이 저장되고, 상기 메모리의 제10 행 내지 제32행에는, 상기 복수의 데이터의 가수 비트들이 저장되는 것을 특징으로 한다.

[0031] 일 실시예에 있어서, 상기 리프레시 주기를 설정하는 단계는, 상기 메모리의 복수의 행 중 상기 부호 비트 또는 상기 지수 비트가 저장되는 일부의 행에 대해, 제1 주기 값을 설정하는 과정과, 상기 메모리의 복수의 행 중 상기 가수 비트가 저장되는, 나머지의 행에 대해, 제2 주기 값을 설정하는 과정을 포함하는 것을 특징으로 한다.

[0032] 일 실시예에 있어서, 상기 제2 주기 값은 상기 제1 주기 값보다 길게 설정되는 것을 특징으로 한다.

[0033] 일 실시예에 있어서, 상기 리프레시 동작을 수행하는 단계는, 상기 제1 주기마다 카운터를 증가시키는 과정과, 상기 제1 주기마다 상기 카운터에 근거하여, 상기 메모리의 각 행마다 리프레시의 수행 여부를 결정하는 과정을 포함하는 것을 특징으로 한다.

[0034] 일 실시예에 있어서, 상기 리프레시 동작을 수행하는 단계는, 미리 설정된 리프레시 주기마다 카운터를 증가시키는 과정과, 상기 카운터의 값과, 상기 복수의 행 별로 설정된 리프레시 주기에 근거하여, 상기 행 별로 리프레시의 수행 여부를 결정하는 과정을 포함하는 것을 특징으로 한다.

[0035] 일 실시예에 있어서, 상기 리프레시 동작을 수행하는 단계는, 상기 복수의 행 중 일부분에 대해, 상기 카운터가 증가될 때마다 리프레시 동작을 수행하는 과정과, 상기 복수의 행 중 나머지 부분에 대해, 상기 카운터가 소정의 값에 대응될 때, 리프레시 동작을 수행하는 과정을 포함하는 것을 특징으로 한다.

[0036] 또한, 본 발명에서 제안하는 데이터 처리장치는, 딥 러닝 애플리케이션을 동작시키는 프로세서와, 상기 딥 러닝 애플리케이션과 관련된 데이터를 저장하는 메모리 디바이스 및 상기 메모리 디바이스의 데이터 저장 방식을 제어하는 메모리 컨트롤러를 포함하고, 상기 메모리 디바이스는, 복수의 행으로 구성되며, 복수의 데이터를 전치(transposed) 방식으로 저장하고, 상기 복수의 행 중 적어도 일부에 대해, 미리 설정된 제1 주기마다 리프레시 동작을 수행하고, 나머지 일부에 대해, 상기 제1 주기와 상이한 제2 주기마다 리프레시 동작을 수행하도록 구성되는 것을 특징으로 한다.

[0037] 일 실시예에 있어서, 상기 메모리 컨트롤러는, 상기 복수의 데이터로부터 추출한 복수의 부호 비트를 상기 메모리 디바이스의 제1 어드레스 영역에 저장시키고, 상기 복수의 데이터로부터 추출한 복수의 지수 비트를 상기 메모리 디바이스의 제2 어드레스 영역에 저장시키고, 상기 복수의 데이터로부터 추출한 복수의 가수 비트를 상기 메모리 디바이스의 제3 어드레스 영역에 저장시키는 것을 특징으로 한다.

[0038] 일 실시예에 있어서, 상기 복수의 데이터가 32-비트 플로팅 포인트 데이터인 경우, 상기 제1 어드레스 영역은 상기 메모리 디바이스의 제1 행에 대응되고, 상기 제2 어드레스 영역은 상기 메모리 디바이스의 제2 행 내지 제9 행에 대응되며, 상기 제3 어드레스 영역은 상기 메모리 디바이스의 제10 행 내지 제32 행에 대응되는 것을 특징으로 한다.

[0039] 일 실시예에 있어서, 상기 메모리 디바이스는, 상기 복수의 데이터가 전치 방식으로 저장된 후, 미리 설정된 주기마다 리프레시 동작을 수행하되, 상기 복수의 행 중 적어도 일부에 대해 상기 리프레시 동작을 스킵하도록 구성되는 것을 특징으로 한다.

[0040] 일 실시예에 있어서, 상기 메모리 디바이스는, 상기 리프레시 동작이 개시되면, 상기 주기마다 소정의 카운터 변수를 증가시키고, 상기 복수의 행 중 적어도 일부에 대해, 증가된 상기 카운터 변수에 근거하여 리프레시의 수행 여부를 결정하는 것을 특징으로 한다.

[0041] 일 실시예에 있어서, 상기 복수의 행 중 적어도 일부에는, 상기 복수의 데이터의 가수 비트가 저장되는 것을 특징으로 한다.

[0042] 일 실시예에 있어서, 상기 메모리 디바이스는, 상기 복수의 데이터의 가수 비트가 저장된 적어도 하나의 행과, 상기 복수의 데이터의 부호 비트 또는 지수 비트가 저장된 적어도 하나의 행에 대해, 서로 다른 주기로 리프레시 동작을 수행하는 것을 특징으로 한다.

[0043] 일 실시예에 있어서, 상기 메모리 디바이스는, 상기 부호 비트 또는 지수비트가 저장된 적어도 하나의 행에 대해, 미리 설정된 주기로 리프레시 동작을 수행하고, 상기 리프레시 동작이 복수 회 수행되는 동안, 상기 가수 비트가 저장된 적어도 하나의 행에 대해, 리프레시 동작을 적어도 일 회 이상 스킵하는 것을 특징으로 한다.

## 발명의 효과

[0044] 본 발명에서 제안하는 근사적 메모리 아키텍처에 의하면, 리프레시 동작 횟수를 감소시킬 수 있으므로, 리프레시 동작에 의한 전력 소비가 최소화되는 장점이 있다.

[0045] 특히, 본 발명에서 제안하는 근사적 메모리 아키텍처가 딥 러닝 애플리케이션에 이용될 때, 딥 러닝의 학습 성능을 유지하면서도, 디램의 전력 소비를 감소시킬 수 있는 효과가 도출된다.

[0046] 딥 러닝 애플리케이션의 데이터 저장을 위해, 간소화된 디램 아키텍처는 데이터를 변환된 방식으로 저장하므로,

상기 저장된 데이터가 중요성에 따라 정렬된다.

[0047] 아울러, 이러한 디램 조직은 저장된 데이터의 중요성에 따라 새로 고침 기간 제어를 지원하도록 수정된다. 이와 관련하여, GoogLeNet 및 VGG-16의 시뮬레이션 결과를 참조하면, GoogLeNet 및 VGG16 모두에 대해 분류 정확도가 상당히 감소하였고, 이에 따라 디램의 전력 소비가 69.68 % 감소한 것으로 나타났다.

## 도면의 간단한 설명

[0048] 도 1a는 종래의 메모리가 데이터를 저장하는 방식을 나타내는 개념도.

도 1b는 본 발명에 따른 메모리 아키텍처에 대응되는 데이터 저장 방식을 나타내는 개념도.

도 1c는 도 1b에 도시된 데이터 분포를 구현하기 위한 메모리 컨트롤러를 나타내는 블록도.

도 2는 본 발명에 따른 근사적 메모리의 32-비트 플로팅 포인트 데이터의 일 예를 나타내는 개념도.

도 3은 본 발명에 따른 로우-레벨 리프레시 알고리즘을 나타내는 흐름도.

도 4는 리프레시 전력 감소를 나타내는 그래프.

도 5는 두 파라미터 offset 및 incr의 다양한 조합으로 GoogLeNet의 정확도 저하를 나타내는 그래프.

도 6은 온도 증가에 따른 GogLeNet 및 VGG-16의 정확도 강하를 나타내는 그래프.

도 7은 온도와 리프레시 주기 변화에 따른 정확도 변이를 나타내는 표.

## 발명을 실시하기 위한 구체적인 내용

[0049] 도 1a에서는 종래의 메모리가 데이터를 저장하는 방식이 도시되며, 도 1b에서는 본 발명에서 제안하는 근사적 메모리의 데이터 저장 방식이 도시된다.

[0050] 구체적으로, 도 1a을 참조하면, 데이터[0](=11001010)은 어드레스 0의 메모리 위치에 저장되고, 데이터[1](=10110110)은 어드레스 1에 저장된다.

[0051] 한편, 도 1b를 참조하면, 본 발명에서 제안하는 근사적 메모리는, 데이터[0] 내지 데이터[7]의 모든 최상위비트(Most Significant Bit, MSB)가 어드레스 0의 메모리 위치에 저장되도록 전치 방식으로 데이터를 저장한다.

[0052] 마찬가지 방식으로, 데이터[0] 내지 데이터[7]의 두번째 최상위 비트는 모두 어드레스 1에 저장된다. 이 경우, 어드레스 0이 호스트 프로세서에 의해 액세스되면, 최상위비트(MSB)만이 프로세서에 전달된다.

[0053] 도 1b에 도시된 데이터 분포에 따르면, 행의 중요성에 따라 리프레시 빈도를 가변적으로 설정할 수 있다.

[0054] 도 1b에 도시된 예에서, 어드레스 0에 저장된 데이터는 첫번째 최상위비트(MSB)이므로, 일반적으로 가장 중요한 데이터 값이다.

[0055] 따라서, 첫번째 최상위비트에서 오류가 발생하면, 이러한 데이터를 사용하는 계산 결과에 심각한 영향이 발생할 수 있다.

[0056] 한편, 어드레스 7의 데이터는, 데이터의 최하위비트(LSB)로 구성된다. 최상위비트와 비교하여, 최하위비트는 상대적으로 중요도가 낮기 때문에, 이러한 최하위비트 상의 오류는 계산 결과에 실질적으로 영향을 미치지 않을 수 있다.

[0057] 예를 들면, 객체 분류를 위한 딥 러닝 어플리케이션에서, 도 2에 도시된 근사적 메모리를 이용하는 경우, 객체 분류의 정확성이 크게 저하되지 않을 수 있다. 이 경우, 어드레스 7의 리프레시 주기를 증가시킴으로써, 리프레시에 소요되는 전력의 크기를 감소시킬 수 있다.

[0058] 어드레스 7에 대한 리프레시 주기가 증가하면, 상기 어드레스 7에 대응되는 비트 데이터에 오류가 발생할 가능성이 증가한다. 그러나, 어드레스 7에 저장된 비트 데이터의 중요도는 상대적으로 낮기 때문에, 이러한 오류 가능성 증가가 데이터 전체의 계산 결과에는 큰 영향을 미치지 않을 수 있다.

[0059] 즉, 도 1b에 도시된 새로운 방식의 데이터 분포에 의하면, 데이터 중요도에 따라 리프레시 주기를 가변적으로 설정하는 것이 가능해진다.

[0060] 일 실시예에서, 첫번째 최상위비트 또는 두번째 최상위비트와 같이 중요도가 높은 데이터의 경우, 오류의 발생

을 피하기 위해 정상적인 리프레시 주기를 유지할 수 있다. 또한, 최하위비트와 같이 중요도가 낮은 데이터의 경우, 리프레시 주기를 증가시켜, 상기 최하위비트의 리프레시 동작에 의해 발생될 수 있는 전력 소비를 감소시킬 수 있다.

[0061] 도 1b에 도시된 전치된 데이터 저장 장치는, DRAM이 차단된 방식으로 액세스 될 것을 요구한다. 단일 바이트의 데이터[0]에 액세스하려면, 호스트는 어드레스0 내지 어드레스7에 저장된 데이터를 요청해야 한다. 즉, 호스트는 데이터의 단일 바이트를 액세스하기 위해 8개의 메모리 액세스가 필요하다. 따라서, 이러한 메모리 아키텍처는 단일 바이트의 데이터에 액세스 할 때 비효율적일 수 있다.

[0062] 반면, 한 블록의 데이터를 함께 액세스하는 경우 비효율성을 최소화 할 수 있다. 예를 들어, 데이터[0]에서 데이터[7]까지의 데이터 블록은 함께 액세스되며 8 바이트의 데이터에 대해 8개의 데이터 액세스가 필요하다. 따라서, 이 경우 불필요한 데이터 요청이 요구되지 않는다.

[0063] 도 1b의 데이터 저장 방법을 지원하기 위해서 컴퓨터 시스템은 메모리로부터 데이터를 가져와, 이를 호스트로 전달할 때 데이터 포맷을 변환하는 메모리 컨트롤러를 필요로 한다. 이러한 메모리 컨트롤러는 도 1c에 도시되며, 하드웨어 유닛 데이터 포맷 변환을 담당하는 비트 전이 유닛으로 정의된다.

[0064] 딥 러닝 애플리케이션은, 32-비트 플로팅 포인트 데이터(Floating Point Data)에 액세스한다.

[0065] 도 2에서는 본 발명에서 제안하는 근사적 메모리의 32 비트 플로팅 포인트 데이터의 일 예가 도시된다.

[0066] 도 2의 예에서는, 단순화를 위해 DRAM의 뱅크 당 32 행만이 설명된다. 각각의 플로팅 포인트 데이터는 1 부호 비트(sign bit), 8 지수 비트(exponent bit) 및 23 가수 비트(mantissa bit)를 갖는다. 부호 비트 및 지수 비트는 상대적으로 중요도가 높은 비트로서, 해당 비트에 오류가 발생하면 데이터 값이 크게 변경될 수 있다. 참고로, 이와 관련된 시뮬레이션 결과는 부호 비트 및 지수 비트 상의 오류가, 데이터 계산 결과의 정확도를 상당히 감소시킨다는 것을 보여준다. 시뮬레이션 결과에 대해서는 이하에서 보다 상세히 설명된다.

[0067] 따라서, 9 개의 최상위 비트(MSB)는 64ms(tRET = 64ms)의 정상 속도로 리프레시되고, 0행 내지 8행에 각각 저장된다. 반면, 23 개의 가수 비트의 중요도는 상대적으로 낮으므로, 9행 내지 31행인 근사적 행에 저장된다. 근사적 행의 비트들에 적용되는 리프레시 동작 주기는 행 번호가 증가함에 따라, 리프레시 동작 주기가 증가하므로, 오류 확률이 낮은 행에 더 적은 비트가 저장된다.

[0068] 일 예에서, 행 번호가 증가함에 따라, 리프레시 동작의 주기가 선형적으로 증가할 수 있다.

*수학식 1*

[0069]
$$RP(n)=\begin{cases}64 & for\ 0\le n\le 8\\ (n-9)*incr+offset & for\ 9\le n\le 31\end{cases}$$

[0070] 상기 수학식 1에서 RP (n)은 n 번째 행의 리프레시 주기를 나타내고, 두 개의 파라미터 incr 및 offset은 실험적으로 선택된다. 이때, RP (n)은 로우-레벨(row-level) 리프레시 알고리즘이 적용될 수 있도록 64ms의 배수로 설정된다.

[0071] 오프셋이 증가함에 따라 리프레시 주기도 증가한다. 따라서, 리프레시 동작에 의한 전력 소비는, 에러 확률 증가의 희생으로 감소한다.

[0072] 상기 두 개의 파라미터 incr 및 offset가 각각 1024ms와, 128ms로 설정될 때의 시뮬레이션 결과에 따르면, 디램 디바이스는 무시할 수 있는 정확도 감소로 69.68 %의 충분한 전력 감소를 제공한다.

[0073] 한편, 리프레시 주기를 조정하기 위해서, 본 발명에 따른 DRAM 장치의 내부 로직이 일부 변경될 수 있다. 이를 위해, 현재의 tRET 라운드 (각각의 tRET 라운드는 64ms)를 저장하기 위한 하나의 추가 카운터와, DRAM 장치가 근사적 행에 대한 리프레시 동작을 건너 뛸 수 있도록 현재의 행(row)이 리프레시 되어야하는지 여부를 결정하기 위한 추가 로직이 있다.

[0074] 본 발명에 따른 로우-레벨 리프레시 알고리즘은 도 3에 도시되어 있다. DRAM 디바이스의 통상의 리프레시 동작에서, 메모리 제어기는 8K 자동-리프레시 명령을 내린다. 뱅크 당 8K 행을 갖는 DRAM 디바이스의 경우, 하나의 자동-리프레시 명령은, 리프레시 간격(tREFI = 64ms / 8K = 7.8us)에서 뱅크 당 하나의 행을 리프레시한다.

[0075] 이하의 일 실시예에서는, 파라미터 offset이 1024ms이고, 파라미터 incr이 128ms인 경우, 어떤 논리 회로가 어떤 행을 필요로 하는지 다음과 같이 설명할 수 있다.

[0076] 행 0 내지 행 8은 현재 tRET 라운드에 관계 없이 항상 리프레시 된다.

[0077] 행 9은 tRET 라운드가 15일 때만 리프레시 된다. 참고로, 행 9의 리프레시 주기는 1024ms 또는 16 * tRET로 정의된다.

[0078] 마찬가지로 행 10 내지 31은 tRET이 17, 19, ... 59 일 때 리프레시를 수행한다. DRAM 장치가 근사적 행에 대한 리프레시 동작을 스킵하는 동안, 외부 메모리 제어기는 주기적으로 매 tRET 라운드 동안 8K 리프레시 커맨드를 주기적으로 보통 속도로 전송한다.

[0079] 전치된 유닛의 버퍼 크기는 다음과 같이 계산된다. 추가 데이터 전송을 피하기 위해, 32개의 데이터가 차단된 방식으로 함께 전송된다.

[0080] 트랜스퍼 유닛은 데이터 전송을 지원하기 위해 추가 버퍼를 사용하여 트랜스퍼될 데이터를 임시로 저장한다. DRAM의 버스트 길이가 8(BL = 8)이고 데이터 폭이 32 비트라고 가정하면, DRAM에 대한 단일 데이터 액세스는 256 비트를 전송한다.

[0081] 따라서, 버퍼의 크기는 256x32 비트로 구현된다. 이 버퍼와 DRAM 사이에 데이터를 전송하기 위해서는 DRAM에 대한 32개의 명령이 필요하므로, 대기 시간이 길어질 수 있다.

[0082] 한편, 딥 러닝 애플리케이션은 일반적으로 예측 가능한 방식으로 데이터에 액세스하기 때문에, 위와 같은 대기 시간의 증가를 방지하기 위하여 데이터 프리 패치 기술을 적용할 수 있다.

[0083] 이하에서는, 본 발명에 따른 근사적 메모리 아키텍처가 CNN(Convolution Neural Network) 애플리케이션에 사용될 때 정확성의 저하뿐만 아니라 전력 소비의 감소를 평가하는 시뮬레이션 결과를 설명한다.

[0084] Caffe 라이브러리에서 사전 훈련된 모델을 사용하여 GoogLeNet(7백만개의 파라미터) 및 VGG-16(138백만개의 파라미터)이 시뮬레이션에 사용된다. 시뮬레이션을 위하여, 상기 CNN 모델은 상술한 것과 같이 근사적 메모리에 저장되는 것으로 가정한다.

[0085] DRAM의 리프레시 전력 감소는 스킵된 리프레시 커맨드의 수를 계산함으로써 측정된다. 리프레시 전력의 감소는 다음 방정식으로부터 수학적으로 유도된다.

*수학식 2*

[0086]
$$Psave = 1 - \frac{9 + \sum_{n=0}^{22} 64/(n*incr + offset)}{32} = 0.71875 - \sum_{n=0}^{22} \frac{2}{(n*incr + offset)}$$

[0087] 상기 수학식 2에서 n, incr 및 offset은 수학식 1에서 정의된 파라미터와 같다. 도 4는 이러한 매개 변수에 대한 리프레시 전력 감소를 보여준다. 리프레시 전력 감소는 오프셋이 4,096ms 일 때 약 71%에서 거의 포화 상태이다.

[0088] 다음 시뮬레이션은 근사에 의한 정확도 감소를 추정하기 위해 수행된다.

[0089] 긴 리프레쉬 속도로 인한 에러율과 관련된 데이터는 도 7에 요약되어 있다.

[0090] 에러율은 DRAM의 동작 온도에 의존적이므로, 에러율에 대한 온도 변화의 영향이 도시된다.

[0091] CNN 모델에 오류 주입은 다음과 같이 수행된다.

[0092] 도 7의 bit errorprobability가 주어지면 사용자 정의 리프레시 주기의 오류 비율이 선형으로 보정된다. 이에

따라, 각 데이터 비트의 오류 확률이 도출된다.

[0093] 각 테스트에서, 비트 플립의 위치는 균일 분포에 따라 무작위로 선택된다.

[0094] 본 발명에 따른 근사적 메모리 아키텍처의 정확도를 측정하기 위해 ImageNet 2012 데이터 세트의 10,000 개의 테스트 이미지 세트가 사용된다. 각 추론 테스트는 수십 번 반복되어 평균 정확도가 계산된다. 각 테스트의 추론 정확도는 메모리 근사를 사용하지 않고 해당 테스트의 정확도로 정규화 된 최상위의 정확도이다.

[0095] 한편, 도 5는 두 파라미터 offset 및 incr의 다양한 조합으로 GoogLeNet의 정확도 저하를 보여준다. 수평축은 작동 온도를 나타낸다.

[0096] 시뮬레이션 결과에 따르면, 60° C보다 낮은 온도에서 정확도 손실은 모든 (offset, incr) 쌍에서 무시할 수 있다. 한편, 60° C보다 높은 온도에서는 정확도 손실이 증가한다. VGG-16의 시뮬레이션 결과는 GoogLeNet과 유사하다.

[0097] 다음 시뮬레이션은 60° C보다 높은 온도에서 수행된다.

[0098] 도 6은 고온 (80° C)에서의 GogLeNet 및 VGG-16의 정확도 강하를 보여준다.

[0099] 도 6에 도시 된 바와 같이, 1024ms의 오프셋은 큰 리프레시 전력 감소를 달성하는 동안, 많은 부정확성을 저하시키지 않는다. (1024, 128)은 (1024, 64)보다 약간 더 많은 전력을 절약하지만, (1024, 64) 및 (1024, 128)의 정확도는 99.8 %보다 높다. 이 CNN의 정확성을 유지하면서 파라미터 (1024, 128)는 전력 소비의 69.68 %를 절약한다.

[0100] 한편, 근사적 메모리에 더 많은 MSB를 저장함으로써, 전력 소비가 더 감소되는지 여부를 확인하기 위해 추가 시뮬레이션이 수행된다.

[0101] 두 파라미터 offset 및 incr는 최소 값인 (1024, 64)로 설정된다. 7개의 MSB 만 정확하게 유지되는 경우, 이 정확도는 각각 97.65%와 55.81%로 상당히 떨어진다. 이는 잘못된 지수 비트로 인해, 데이터 값이 크게 변경되는 경우이므로, 예상된 결과이다. 따라서, 모든 지수 비트는 오류없이 유지되어야 한다.

[0102] 상술한 본 발명에서는, 고온에서도 딥 러닝과 같은 오류 복원력이 있는 애플리케이션의 정확성을 유지하면서도, 비-중요(non-critical) 데이터에 대한 리프레시 동작을 건너 뛰(skip)으로써, 리프레시에 의한 전력 소비의 대략 70%를 절약 할 수 있는 새로운 근사적 메모리 아키텍처를 제안한다.

[0103] 제안된 근사적 메모리 아키텍처는 DRAM 장치의 내부 논리만 변경함으로써 구현할 수 있고, 별도의 주요한 하드웨어 요소가 추가되는 것을 필요로하지 않는다.

[0104] 상술한 본 발명의 일 실시예로서 디램 디바이스의 제어방법이 제안된다.

[0105] 도 8에 도시된 디램 디바이스의 제어방법은, 복수의 비트로 구성된 복수의 데이터를 전치(transposed) 방식으로 메모리에 저장하는 단계(S801)와, 상기 메모리를 구성하는 복수의 행(row) 별로 적어도 하나의 리프레시 주기를 설정하는 단계(S802) 및 설정된 리프레시 주기에 근거하여, 상기 메모리의 리프레시 동작을 수행하는 단계(S803)를 포함할 수 있다.

[0106] 일 예에서, 상기 데이터를 전치 방식으로 메모리에 저장하는 단계(S801)는, 상기 메모리를 구성하는 복수의 행(Row)에, 상기 복수의 데이터의 최상위비트(MSB)부터 최하위비트(LSB)까지 순차적으로 저장하는 과정을 포함할 수 있다.

[0107] 즉, 데이터를 전치 방식으로 메모리에 저장하는 단계에 의해, 도 1b에 도시된 것과 같은 방식으로, 디램 디바이스 내에서 데이터가 분포될 수 있다.

[0108] 또한, 상기 데이터를 구성하는 복수의 비트는, 부호 비트와, 지수 비트 및 가수 비트를 포함할 수 있다. 이때, 상기 데이터를 전치 방식으로 메모리에 저장하는 단계는, 상기 복수의 데이터 각각의 부호 비트 및 지수 비트를 상기 메모리의 일부 행에 저장하는 과정과, 상기 복수의 데이터 각각의 가수 비트를 상기 메모리의 나머지 행에 저장하는 과정을 포함할 수 있다.

[0109] 일 실시예에 있어서, 상기 데이터는, 1비트의 부호 비트(sign bit)와, 8비트의 지수 비트(exponent bits)와, 23비트의 가수 비트(mantissa bits)로 구성되고, 상기 메모리의 제1 행에는, 상기 복수의 데이터의 부호 비트들이 저장되고, 상기 메모리의 제2 행 내지 제9 행에는, 상기 복수의 데이터의 지수 비트들이 저장되고, 상기 메모리

의 제10 행 내지 제32행에는, 상기 복수의 데이터의 가수 비트들이 저장될 수 있다.

[0110] 일 실시예에 있어서, 상기 리프레시 주기를 설정하는 단계(S802)는, 상기 메모리의 복수의 행 중 상기 부호 비트 또는 상기 지수 비트가 저장되는 일부의 행에 대해, 제1 주기 값을 설정하는 과정과, 상기 메모리의 복수의 행 중 상기 가수 비트가 저장되는, 나머지의 행에 대해, 제2 주기 값을 설정하는 과정을 포함할 수 있다.

[0111] 이때, 상기 제2 주기 값은 상기 제1 주기 값보다 길게 설정될 수 있다. 즉, 부호 비트 또는 지수 비트가 저장되는 행보다, 가수 비트가 저장되는 행의 리프레시 주기가 더 길게 설정될 수 있다.

[0112] 한편, 상기 리프레시 동작을 수행하는 단계(S803)는, 상기 제1 주기마다 카운터를 증가시키는 과정과, 상기 제1 주기마다 상기 카운터에 근거하여, 상기 메모리의 각 행마다 리프레시의 수행 여부를 결정하는 과정을 포함할 수 있다.

[0113] 일 실시예에 있어서, 상기 리프레시 동작을 수행하는 단계(S803)는, 미리 설정된 리프레시 주기마다 카운터를 증가시키는 과정과, 상기 카운터의 값과, 상기 복수의 행 별로 설정된 리프레시 주기에 근거하여, 상기 행 별로 리프레시의 수행 여부를 결정하는 과정을 포함할 수 있다.

[0114] 상기 리프레시 동작을 수행하는 단계(S803)는, 상기 복수의 행 중 일부분에 대해, 상기 카운터가 증가될 때마다 리프레시 동작을 수행하는 과정과, 상기 복수의 행 중 나머지 부분에 대해, 상기 카운터가 소정의 값에 대응될 때, 리프레시 동작을 수행하는 과정을 포함할 수 있다.

[0115] 즉, 본 발명에 따른 디램 디바이스의 제어방법은, 가수 비트가 저장된 행에 대해, 카운터가 특정 값에 해당될 때만 리프레시 동작을 수행할 수 있다. 이로써, 가수 비트에 대한 리프레시 동작은 카운터가 특정 값에 해당 되지 않을 때 스킵되며, 본 발명에 따른 디램 디바이스는 불필요한 리프레시 동작을 생략할 수 있다.

[0116] 또한, 본 발명에서 제안하는 데이터 처리장치(100)는, 도 1c에 도시된 것과 같이, 프로세서(101)와, 근사적 메모리 디바이스(103) 및 메모리 컨트롤러(102)를 포함할 수 있다.

[0117] 구체적으로, 본 발명에 따른 데이터 처리장치(100)는, 딥 러닝 애플리케이션을 동작시키는 프로세서(101)와, 상기 딥 러닝 애플리케이션과 관련된 데이터를 저장하는 메모리 디바이스(103) 및 상기 메모리 디바이스의 데이터 저장 방식을 제어하는 메모리 컨트롤러(102)를 포함하고, 상기 메모리 디바이스는, 복수의 행으로 구성되며, 복수의 데이터를 전치(transposed) 방식으로 저장하고, 상기 복수의 행 중 적어도 일부에 대해, 미리 설정된 제1 주기마다 리프레시 동작을 수행하고, 나머지 일부에 대해, 상기 제1 주기와 상이한 제2 주기마다 리프레시 동작을 수행하도록 구성되는 것을 특징으로 한다.

[0118] 전치 방식으로 데이터를 저장하기 위하여, 메모리 컨트롤러(102)는, 비트 전이 유닛(104)을 구비할 수 있다.

[0119] 일 실시예에 있어서, 상기 메모리 컨트롤러는, 상기 복수의 데이터로부터 추출한 복수의 부호 비트를 상기 메모리 디바이스의 제1 어드레스 영역에 저장시키고, 상기 복수의 데이터로부터 추출한 복수의 지수 비트를 상기 메모리 디바이스의 제2 어드레스 영역에 저장시키고, 상기 복수의 데이터로부터 추출한 복수의 가수 비트를 상기 메모리 디바이스의 제3 어드레스 영역에 저장시키는 것을 특징으로 한다.

[0120] 일 실시예에 있어서, 상기 복수의 데이터가 32-비트 플로팅 포인트 데이터인 경우, 상기 제1 어드레스 영역은 상기 메모리 디바이스의 제1 행에 대응되고, 상기 제2 어드레스 영역은 상기 메모리 디바이스의 제2 행 내지 제9 행에 대응되며, 상기 제3 어드레스 영역은 상기 메모리 디바이스의 제10 행 내지 제32 행에 대응되는 것을 특징으로 한다.

[0121] 일 실시예에 있어서, 상기 메모리 디바이스는, 상기 복수의 데이터가 전치 방식으로 저장된 후, 미리 설정된 주기마다 리프레시 동작을 수행하되, 상기 복수의 행 중 적어도 일부에 대해 상기 리프레시 동작을 스킵하도록 구성되는 것을 특징으로 한다.

[0122] 일 실시예에 있어서, 상기 메모리 디바이스는, 상기 리프레시 동작이 개시되면, 상기 주기마다 소정의 카운터 변수를 증가시키고, 상기 복수의 행 중 적어도 일부에 대해, 증가된 상기 카운터 변수에 근거하여 리프레시의 수행 여부를 결정하는 것을 특징으로 한다.

[0123] 일 실시예에 있어서, 상기 복수의 행 중 적어도 일부에는, 상기 복수의 데이터의 가수 비트가 저장되는 것을 특징으로 한다.

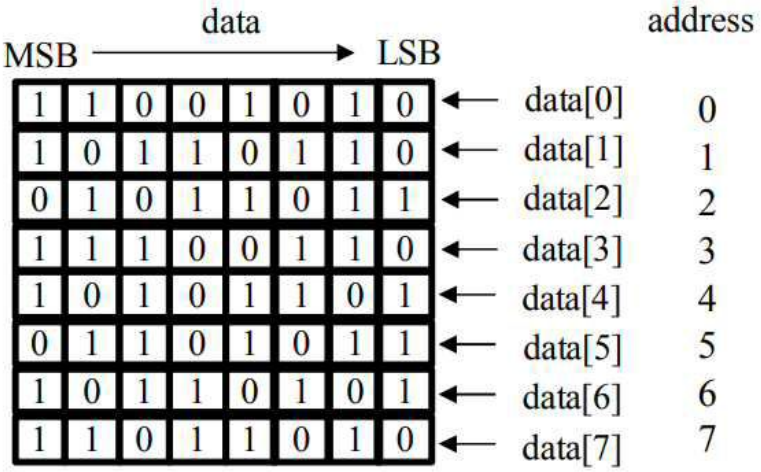
[0124] 일 실시예에 있어서, 상기 메모리 디바이스는, 상기 복수의 데이터의 가수 비트가 저장된 적어도 하나의 행과,

상기 복수의 데이터의 부호 비트 또는 지수 비트가 저장된 적어도 하나의 행에 대해, 서로 다른 주기로 리프레시 동작을 수행하는 것을 특징으로 한다.

[0125] 일 실시예에 있어서, 상기 메모리 디바이스는, 상기 부호 비트 또는 지수비트가 저장된 적어도 하나의 행에 대해, 미리 설정된 주기로 리프레시 동작을 수행하고, 상기 리프레시 동작이 복수 회 수행되는 동안, 상기 가수 비트가 저장된 적어도 하나의 행에 대해, 리프레시 동작을 적어도 일 회 이상 스킵하는 것을 특징으로 한다.

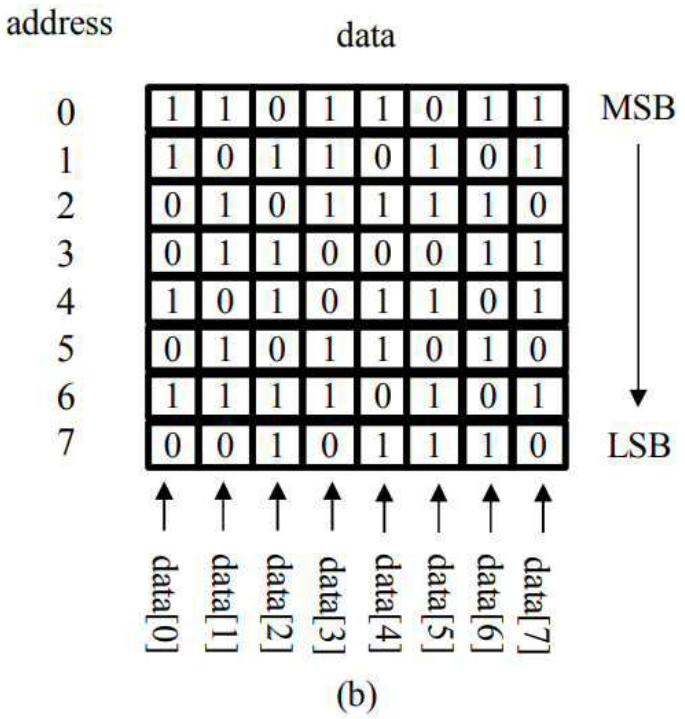
## 도면

### 도면1a

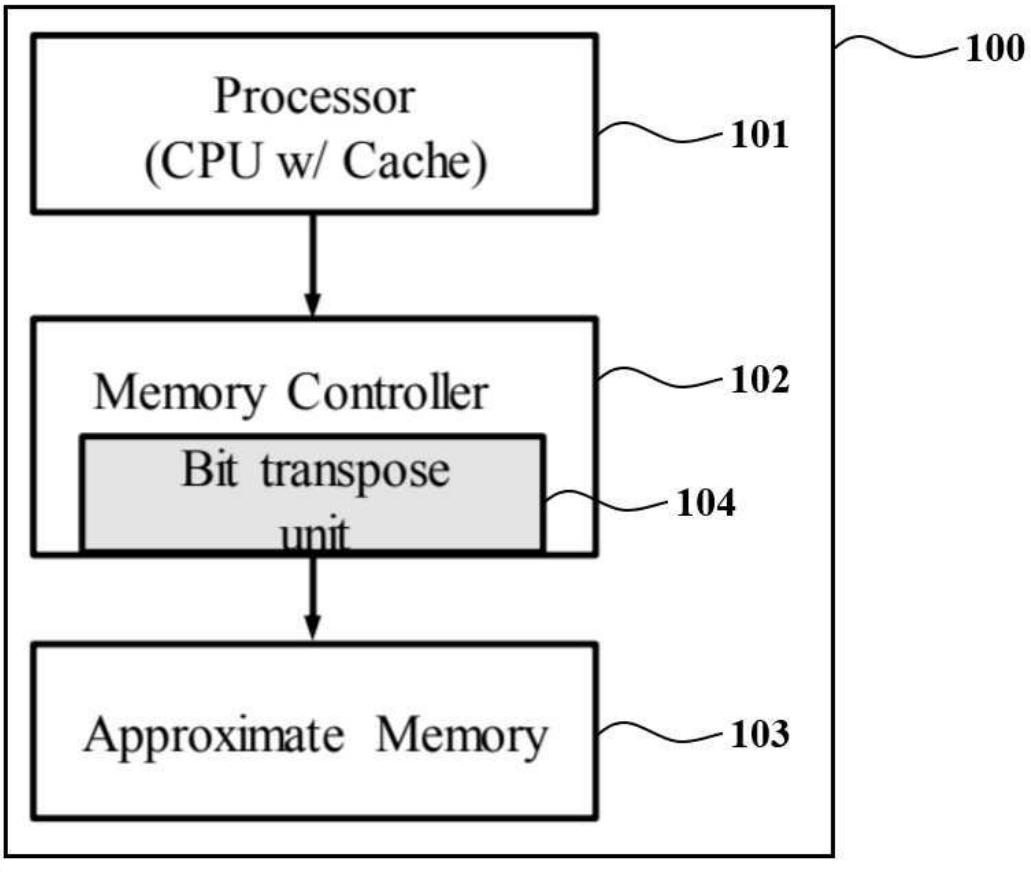



(a)

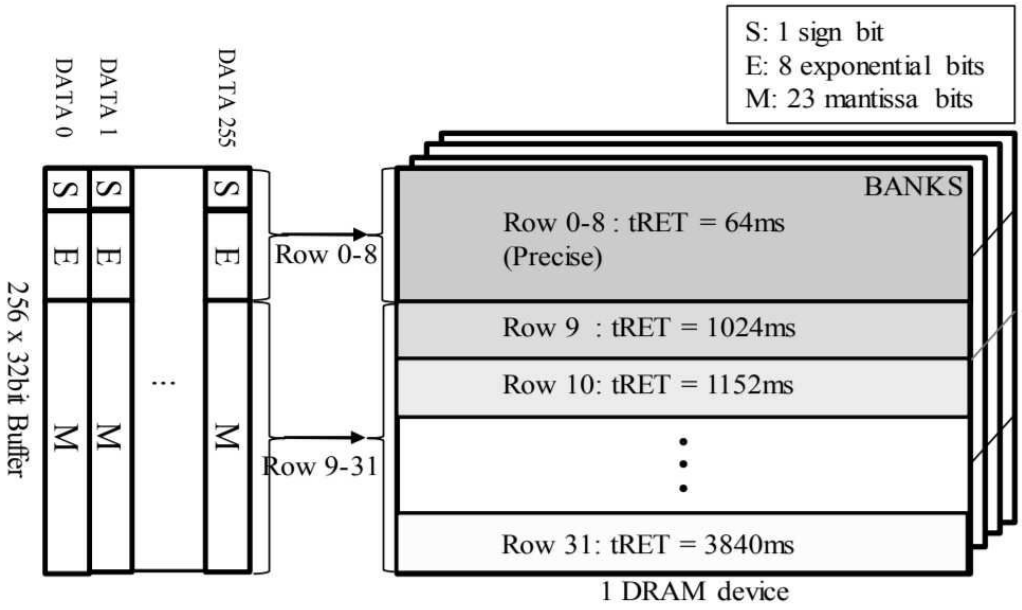
도면1b


address
data
0
1
2
3
4
5
6
7
1 1 0 1 1 0 1 1
1 0 1 1 0 1 0 1
0 1 0 1 1 1 1 0
0 1 1 0 0 0 1 1
1 0 1 0 1 1 0 1
0 1 0 1 1 0 1 0
1 1 1 1 0 1 0 1
0 0 1 0 1 1 1 0
MSB
LSB
data[0]
data[1]
data[2]
data[3]
data[4]
data[5]
data[6]
data[7]
(b)

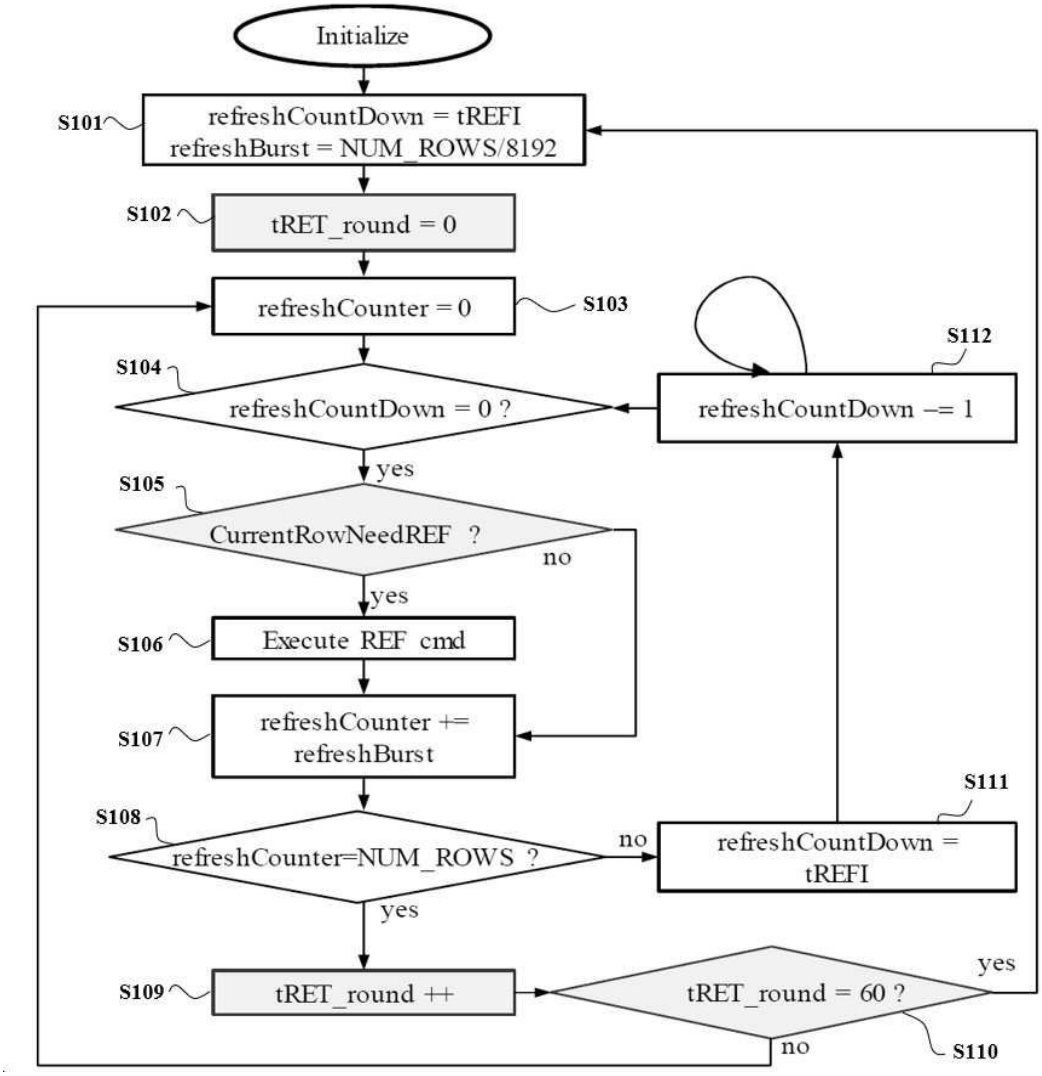
도면1c


Processor
(CPU w/ Cache)
101
100
Memory Controller
102
Bit transpose
unit
104
Approximate Memory
103

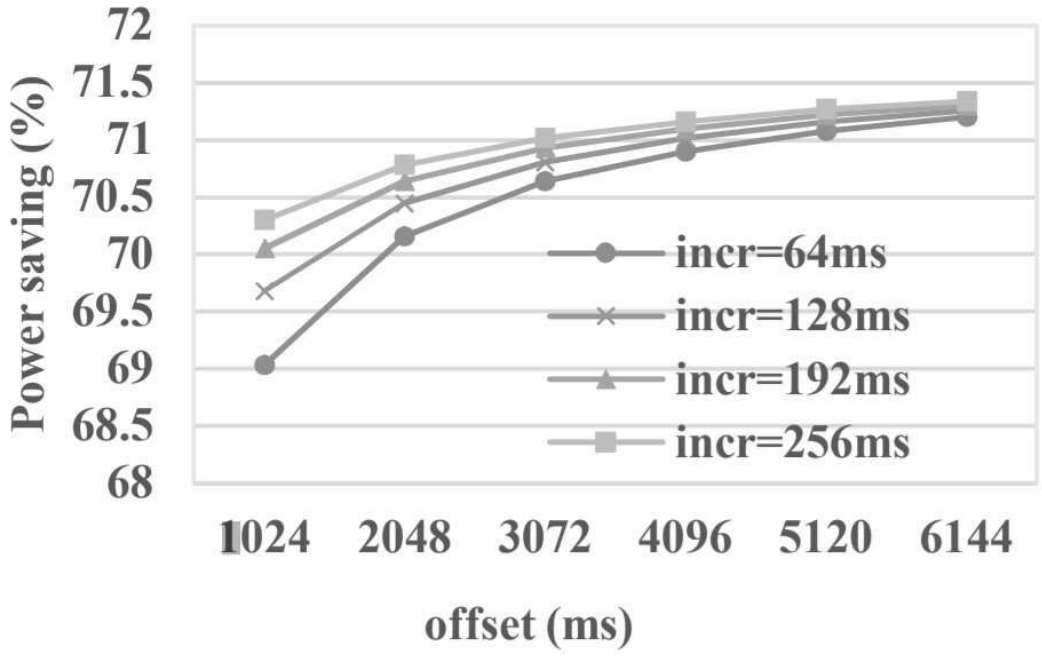
도면2


S: 1 sign bit
E: 8 exponential bits
M: 23 mantissa bits
DATA 0
DATA 1
DATA 255
S
E
M
256 x 32bit Buffer
Row 0-8
Row 9-31
BANKS
Row 0-8 : tRET = 64ms
(Precise)
Row 9 : tRET = 1024ms
Row 10: tRET = 1152ms
Row 31: tRET = 3840ms
1 DRAM device

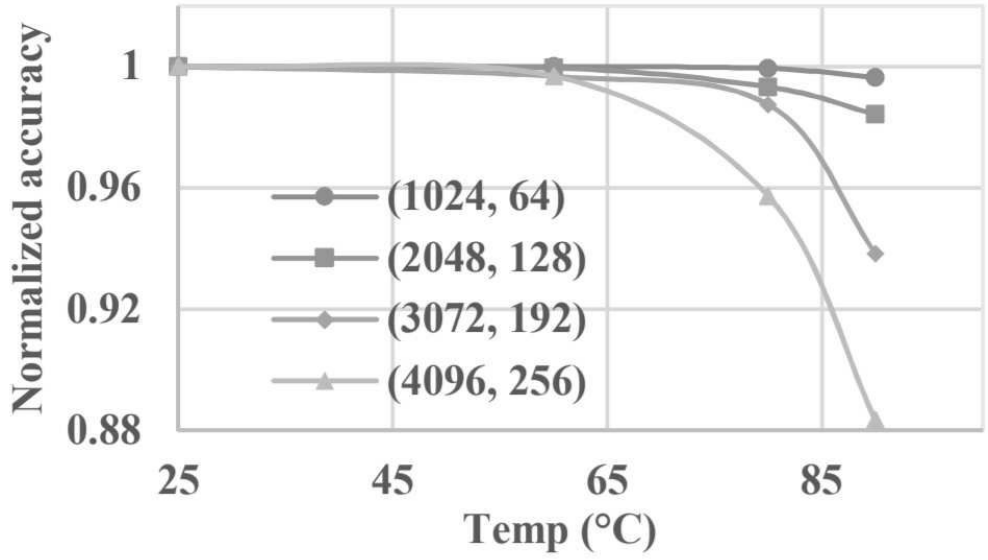
도면3


Initialize
S101
refreshCountDown = tREFI
refreshBurst = NUM_ROWS/8192
S102
tRET_round = 0
refreshCounter = 0
S103
S104
refreshCountDown = 0 ?
S112
refreshCountDown -= 1
yes
S105
CurrentRowNeedREF ?
no
yes
S106
Execute REF cmd
S107
refreshCounter +=
refreshBurst
S111
S108
refreshCounter=NUM_ROWS ?
no
refreshCountDown =
tREFI
yes
S109
tRET_round ++
tRET_round = 60 ?
yes
no
S110

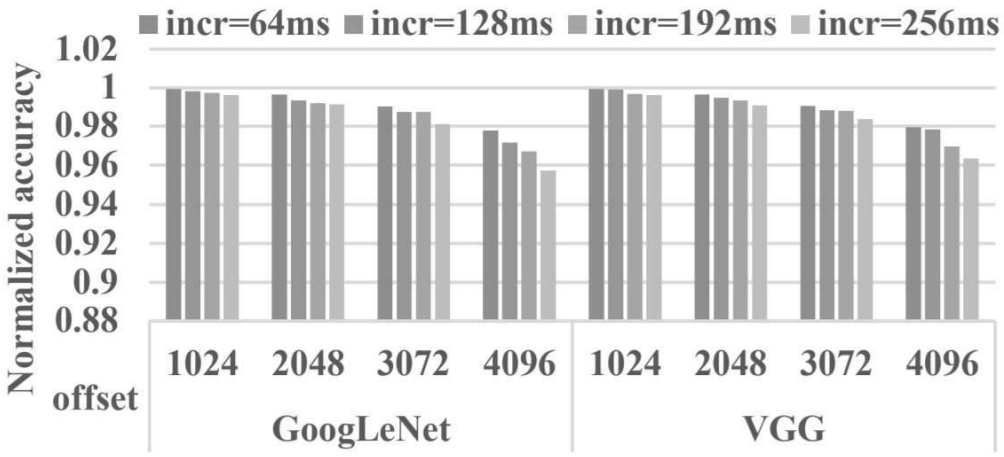
도면4


72
71.5
71
70.5
70
69.5
69
68.5
68
Power saving (%)
incr=64ms
incr=128ms
incr=192ms
incr=256ms
1024
2048
3072
4096
5120
6144
offset (ms)

도면5


1
0.96
0.92
0.88
Normalized accuracy
(1024, 64)
(2048, 128)
(3072, 192)
(4096, 256)
25
45
65
85
Temp (°C)

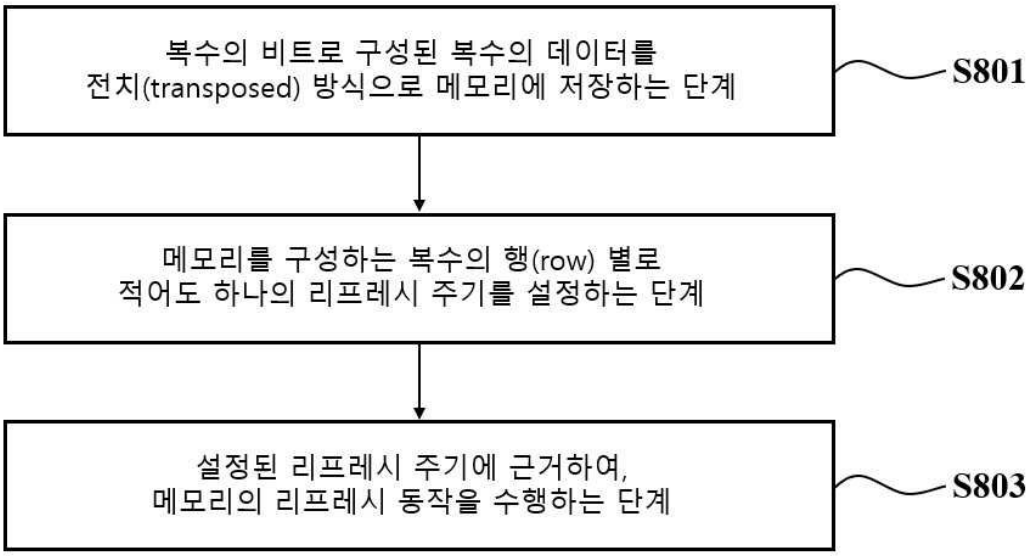
도면6




도면7

| Refresh period (s) | Bit error at 25°C | Bit error at 60°C | Bit error at 80°C | Bit error at 90°C |
|---|---|---|---|---|
| 1 | 0 | 1.87196E-7 | 0.00025795 | 0.00225826 |
| 2 | 0 | 4.00958E-6 | 0.00194832 | 0.01180268 |
| 3 | 1.16415E-9 | 2.10621E-5 | 0.00597532 | 0.02738172 |
| 4 | 4.42378E-9 | 5.73678E-5 | 0.01404861 | 0.04950775 |
| 5 | 1.39698E-8 | 0.00023114 | 0.02723729 | 0.07771726 |
| 6 | 2.39816E-8 | 0.00040858 | 0.03963425 | 0.10455954 |
| 7 | 4.02797E-8 | 0.00062755 | 0.04637795 | 0.11369069 |
| 8 | 6.61239E-8 | 0.00069538 | 0.05646876 | 0.13063926 |
| 9 | 9.33651E-8 | 0.00135385 | 0.07410479 | 0.16315238 |
| 10 | 1.35973E-7 | 0.00226014 | 0.07608483 | 0.16615006 |

도면8


복수의 비트로 구성된 복수의 데이터를
전치(transposed) 방식으로 메모리에 저장하는 단계
S801
메모리를 구성하는 복수의 행(row) 별로
적어도 하나의 리프레시 주기를 설정하는 단계
S802
설정된 리프레시 주기에 근거하여,
메모리의 리프레시 동작을 수행하는 단계
S803