



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0158179
(43) 공개일자 2024년11월04일

(51) 국제특허분류(Int. Cl.)
G11C 29/42 (2015.01) G06F 11/10 (2006.01)
G06N 3/08 (2023.01)
(52) CPC특허분류
G11C 29/42 (2013.01)
G06F 11/1012 (2013.01)
(21) 출원번호 10-2024-0140286
(22) 출원일자 2024년10월15일
심사청구일자 없음

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
정성우
서울특별시 노원구 동일로215길 23, 215동 504호 (상계동, 상계주공2단지아파트)
김호석
서울특별시 서초구 명달로9길 57, 방배아트자이 105동 604호
최승훈
서울특별시 성북구 보문로30가길 41-23, 302호 (동선동2가, 필하우스)
(74) 대리인
특허법인 고려

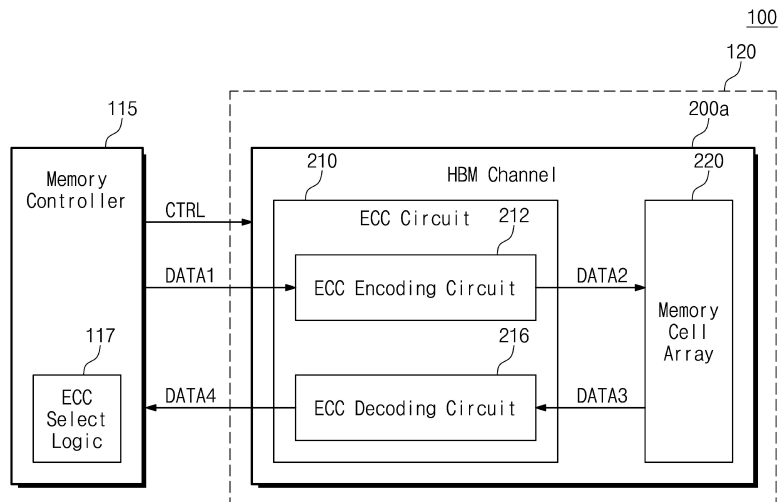
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 고-대역폭 메모리 장치의 오류 정정 코드 회로 및 그 동작 방법

(57) 요약

본 발명의 실시 예에 따른 메모리 장치의 오류 정정 코드 회로는 제1 인코더 및 제2 인코더를 포함하는 오류 정정 코드 인코딩 회로, 및 제1 디코더 및 제2 디코더를 포함하는 오류 정정 코드 디코딩 회로를 포함하되, 메모리 컨트롤러로부터 수신된 데이터가 가중치 블록인 경우, 상기 오류 정정 코드 인코딩 회로는 상기 데이터를 상기 제1 인코더로 인코딩하고, 상기 데이터가 비가중치 블록인 경우, 상기 오류 정정 코드 인코딩 회로는 상기 데이터를 상기 제2 인코더로 인코딩하고, 그리고 상기 제1 인코더에 의해 인코딩된 상기 데이터에 대한 리프레쉬 주기는 상기 제2 인코더에 의해 인코딩된 상기 데이터에 대한 리프레쉬 주기보다 길다.

대표도



(52) CPC특허분류

G06F 11/1048 (2013.01)

G06N 3/08 (2023.01)

Y02D 10/00 (2020.08)

명세서

청구범위

청구항 1

메모리 장치의 오류 정정 코드 회로에 있어서:

제1 인코더 및 제2 인코더를 포함하는 오류 정정 코드 인코딩 회로; 및

제1 디코더 및 제2 디코더를 포함하는 오류 정정 코드 디코딩 회로를 포함하되,

메모리 컨트롤러로부터 수신된 데이터가 가중치 블록인 경우, 상기 오류 정정 코드 인코딩 회로는 상기 데이터를 상기 제1 인코더로 인코딩하고,

상기 데이터가 비가중치 블록인 경우, 상기 오류 정정 코드 인코딩 회로는 상기 데이터를 상기 제2 인코더로 인코딩하고, 그리고

상기 제1 인코더에 의해 인코딩된 상기 데이터에 대한 리프레쉬 주기는 상기 제2 인코더에 의해 인코딩된 상기 데이터에 대한 리프레쉬 주기보다 긴 오류 정정 코드 회로.

청구항 2

제 1 항에 있어서,

상기 데이터가 상기 가중치 블록인 경우, 상기 오류 정정 코드 인코딩 회로의 상기 제1 인코더는:

상기 가중치 블록에 포함된 16-비트 부동 소수점 형식으로 구현된 제1 내지 제64 가중치 데이터들의 지수의 최대값 및 최소값의 차이가 16보다 작은 경우, 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하고; 그리고

상기 제1 내지 제64 가중치 데이터들의 상기 지수의 상기 최대값 및 상기 최소값의 상기 차이가 16보다 크거나 16과 같은 경우, 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하도록 구성된 오류 정정 코드 회로.

청구항 3

제 2 항에 있어서,

상기 제1 인코더가 상기 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 경우, 상기 제1 인코더는:

상기 제1 내지 제64 가중치 데이터들의 상기 지수의 상기 최대값을 7회 반복하여 저장하고; 그리고

상기 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하였음을 나타내는 1-비트 플래그를 7회 반복하여 저장하도록 구성된 오류 정정 코드 회로.

청구항 4

제 3 항에 있어서,

상기 제1 인코더가 상기 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 경우, 상기 제1 인코더는:

상기 제1 내지 제64 가중치 데이터들의 각각의 상기 지수를 저장하지 않고;

상기 제1 내지 제64 가중치 데이터들의 상기 지수의 최대값 및 상기 제1 내지 제64 가중치 데이터들의 각각의 상기 지수의 차이를 상기 제1 내지 제64 가중치 데이터들의 각각의 4-비트 델타에 저장하고; 그리고

상기 제1 내지 제64 가중치 데이터들의 각각의 상기 4-비트 델타에 대하여 단일 오류 정정 기법을 적용하도록 구성된 오류 정정 코드 회로.

청구항 5

제 4 항에 있어서,

상기 제1 인코더가 상기 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 경우, 상기 제1 인코더는 상기 제1 내지 제64 가중치 데이터들의 각각의 부호를 2회 더 저장하도록 구성된 오류 정정 코드 회로.

청구항 6

제 2 항에 있어서,

상기 제1 인코더가 상기 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 경우, 상기 제1 인코더는:

상기 제1 내지 제64 가중치 데이터들의 상기 지수의 상기 최대값을 7회 반복하여 저장하고; 그리고

상기 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하였음을 나타내는 1-비트 플래그를 7회 반복하여 저장하도록 구성된 오류 정정 코드 회로.

청구항 7

제 6 항에 있어서,

상기 제1 인코더가 상기 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 경우, 상기 제1 인코더는:

상기 제1 내지 제64 가중치 데이터들의 각각의 상기 지수를 저장하지 않고;

상기 제1 내지 제64 가중치 데이터들의 상기 지수의 최대값 및 상기 제1 내지 제64 가중치 데이터들의 각각의 상기 지수의 차이를 상기 제1 내지 제64 가중치 데이터들의 각각의 5-비트 델타에 저장하고; 그리고

상기 제1 내지 제64 가중치 데이터들의 각각의 상기 5-비트 델타의 상위 4-비트에 대하여 단일 오류 정정 기법을 적용하도록 구성된 오류 정정 코드 회로.

청구항 8

제 6 항에 있어서,

상기 1 인코더가 상기 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 경우, 상기 제1 인코더는:

상기 제1 내지 제64 가중치 데이터들의 각각의 가수의 하위 1-비트를 삭제하고; 그리고

상기 제1 내지 제64 가중치 데이터들의 각각의 부호를 2회 더 저장하도록 구성된 오류 정정 코드 회로.

청구항 9

가중치 블록에 대한 오류 정정 코드 회로의 동작 방법에 있어서:

메모리 컨트롤러가 메모리 장치로 상기 가중치 블록을 전송하는 단계;

인코더가 상기 가중치 블록에 포함된 16-비트 부동 소수점 형식으로 구현된 제1 내지 제64 가중치 데이터들의 지수의 최대값 및 최소값의 차이가 16보다 작은지 여부를 판단하는 단계; 및

상기 최대값 및 상기 최소값의 상기 차이가 16보다 작은 경우, 상기 인코더가 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하고, 그리고 상기 최대값 및 상기 최소값의 상기 차이가 16보다 크거나 16과 같은 경우, 상기 인코더가 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 단계를 포함하는 오류 정정 코드 회로의 동작 방법.

청구항 10

제 9 항에 있어서,

상기 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 단계는:

상기 제1 내지 제64 가중치 데이터들의 상기 지수의 상기 최대값을 7회 반복하여 저장하는 단계;

상기 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하였음을 나타내는 1-비트 플래그를 7회 반복하여 저장하는 단계;

상기 제1 내지 제64 가중치 데이터들의 각각의 상기 지수를 저장하지 않고, 그리고 상기 제1 내지 제64 가중치 데이터들의 상기 지수의 최대값 및 상기 제1 내지 제64 가중치 데이터들의 각각의 상기 지수의 차이를 상기 제1 내지 제64 가중치 데이터들의 각각의 4-비트 델타에 저장하는 단계;

상기 제1 내지 제64 가중치 데이터들의 각각의 상기 4-비트 델타에 대하여 단일 오류 정정 기법을 적용하는 단계; 및

상기 제1 내지 제64 가중치 데이터들의 각각의 부호를 2회 더 저장하는 단계를 포함하고, 그리고

상기 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 단계는:

상기 제1 내지 제64 가중치 데이터들의 상기 지수의 상기 최대값을 7회 반복하여 저장하는 단계;

상기 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하였음을 나타내는 1-비트 플래그를 7회 반복하여 저장하는 단계;

상기 제1 내지 제64 가중치 데이터들의 각각의 상기 지수를 저장하지 않고, 그리고 상기 제1 내지 제64 가중치 데이터들의 상기 지수의 최대값 및 상기 제1 내지 제64 가중치 데이터들의 각각의 상기 지수의 차이를 상기 제1 내지 제64 가중치 데이터들의 각각의 5-비트 델타에 저장하는 단계;

상기 제1 내지 제64 가중치 데이터들의 각각의 상기 5-비트 델타의 상위 4-비트에 대하여 단일 오류 정정 기법을 적용하는 단계;

상기 제1 내지 제64 가중치 데이터들의 각각의 가수의 하위 1-비트를 삭제하는 단계; 및

상기 제1 내지 제64 가중치 데이터들의 각각의 부호를 2회 더 저장하는 단계를 포함하는 오류 정정 코드 회로의 동작 방법.

발명의 설명

기술 분야

[0001] 본 기재는 고-대역폭 메모리 장치의 오류 정정 코드 회로에 관한 것으로, 보다 상세하게는, 가중치 데이터를 인코딩하는 고-대역폭 메모리 장치의 오류 정정 코드 회로 및 그 동작 방법에 관한 것이다.

배경 기술

[0002] CPU(Central Processing Unit)의 속도가 증가하고, GPU(Graphics Processing Unit), 또는 NPU(Neural Processing Unit) 등의 사용이 증가함에 따라, 이들의 속도에 부합할 수 있는 고-대역폭 메모리 장치가 점차 요구되고 있다. 이와 같은 고-대역폭 메모리 장치의 대표적인 예는 HBM(High-Bandwidth Memory) 장치가 있다. 고-대역폭 메모리 장치는 통상적인 메모리 장치보다 높은 대역폭을 가질 수 있고, 상술한 프로세서들의 동작 속도에 부합하도록 데이터를 제공할 수 있다.

[0003] HBM 장치는 DNN(Deep Neural Network) 추론과 같은 대규모 연산 작업에서 사용될 수 있다. 이 때, 향상된 성능을 갖는 HBM 장치를 위해, HBM의 리프레쉬 오버헤드를 줄이는 다양한 방안이 연구되고 있다. 다만, DNN 모델의 크기가 증가함에 따라, 고온(예, 섭씨 90도 이상)에서 동작하는 경우 리텐션 오류가 증가하는 어려움이 있다.

발명의 내용

해결하려는 과제

[0004] 본 기재의 목적은 DNN 추론에서 에너지 효율적인 고-대역폭 메모리 장치의 오류 정정 코드 회로를 제공하는 데에 있다.

과제의 해결 수단

[0005] 본 발명의 실시 예에 따른 메모리 장치의 오류 정정 코드 회로는 제1 인코더 및 제2 인코더를 포함하는 오류 정정 코드 인코딩 회로, 및 제1 디코더 및 제2 디코더를 포함하는 오류 정정 코드 디코딩 회로를 포함하되, 메모리 컨트롤러로부터 수신된 데이터가 가중치 블록인 경우, 상기 오류 정정 코드 인코딩 회로는 상기 데이터를 상기 제1 인코더로 인코딩하고, 상기 데이터가 비가중치 블록인 경우, 상기 오류 정정 코드 인코딩 회로는 상기 데이터를 상기 제2 인코더로 인코딩하고, 그리고 상기 제1 인코더에 의해 인코딩된 상기 데이터에 대한 리프레쉬 주기는 상기 제2 인코더에 의해 인코딩된 상기 데이터에 대한 리프레쉬 주기보다 길다.

[0006] 본 발명의 실시 예에 따른 가중치 블록에 대한 오류 정정 코드 회로의 동작 방법은 메모리 컨트롤러가 메모리 장치로 상기 가중치 블록을 전송하는 단계, 인코더가 상기 가중치 블록에 포함된 16-비트 부동 소수점 형식으로 구현된 제1 내지 제64 가중치 데이터들의 지수의 최대값 및 최소값의 차이가 16보다 작은지 여부를 판단하는 단계, 및 상기 최대값 및 상기 최소값의 상기 차이가 16보다 작은 경우, 상기 인코더가 4-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하고, 그리고 상기 최대값 및 상기 최소값의 상기 차이가 16보다 크거나 16과 같은 경우, 상기 인코더가 5-비트 델타 인코딩에 기반하여 상기 가중치 블록을 인코딩하는 단계를 포함한다.

발명의 효과

[0007] 본 기재에 따르면, DNN 추론에서 고-대역폭 메모리 장치의 오류 정정 코드 회로의 리프레쉬 주기가 통상적인 오류 정정 코드 회로에 비해 상대적으로 길 수 있다. 따라서, 에너지 효율적이고, 추론 정확도가 높은 고-대역폭 메모리 장치의 오류 정정 코드 회로 및 그 동작 방법이 제공된다.

도면의 간단한 설명

[0008] 도 1은 본 기재의 실시 예에 따른 메모리 시스템을 보여주는 블록도이다.
 도 2는 본 기재의 실시 예에 따른 메모리 장치의 동작을 설명하기 위한 도면이다.
 도 3a 및 도 3b는 본 기재의 실시 예에 따른 오류 정정 코드 인코딩 회로 및 오류 정정 코드 디코딩 회로를 보여주는 도면이다.
 도 4a 내지 도 4c는 본 기재의 실시 예에 따른 가중치 블록을 설명하기 위한 도면이다.
 도 5는 본 기재의 실시 예에 따른 4-비트 델타 인코딩에 기반하여 인코딩된 가중치 데이터를 보여주는 도면이다.
 도 6은 본 기재의 실시 예에 따른 4-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록을 보여주는 도면이다.
 도 7는 본 기재의 실시 예에 따른 5-비트 델타 인코딩에 기반하여 인코딩된 가중치 데이터를 보여주는 도면이다.
 도 8은 본 기재의 실시 예에 따른 5-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록을 보여주는 도면이다.
 도 9는 본 기재의 실시 예에 따른 오류 정정 코드 회로의 동작 방법의 예를 보여주는 순서도이다.
 도 10은 본 기재의 실시 예에 따른 4-비트 델타 인코딩 방법의 예를 보여주는 순서도이다.
 도 11은 본 기재의 실시 예에 따른 5-비트 델타 인코딩 방법의 예를 보여주는 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0009] 이하에서, 본 발명의 기술 분야에서 통상의 지식을 가진 자가 본 발명을 용이하게 실시할 수 있을 정도로, 본 발명의 실시 예들이 명확하고 상세하게 기재될 것이다.

[0010] 도 1은 본 기재의 실시 예에 따른 메모리 시스템을 보여주는 블록도이다. 도 1을 참조하면, 메모리 시스템(100)은 호스트(110) 및 메모리 장치(120)를 포함할 수 있고, 그리고 호스트(110)는 메모리 컨트롤러(115)를 포함할 수 있다.

[0011] 메모리 시스템(100)은 데이터(DATA)를 저장하고, 저장한 데이터(DATA)를 관리하여 사용자에게 필요한 정보를 제공할 수 있다. 예를 들어, 메모리 시스템(100)은 PC(Personal Computer), 랩톱 컴퓨터(Laptop Computer), 태블릿 PC(tablet PC), PDA(Personal Digital Assisatant), 웨어러블 장치(Wearable Device), 또는 카메라와 같은 전자 장치에 포함될 수 있다. 다만, 이는 예시적인 것이며, 본 발명의 범위는 이에 한정되지 않는다.

- [0012] 호스트(110)는 메모리 장치(120)와 데이터(DATA)를 주고받을 수 있다. 구체적으로, 호스트(110)는 메모리 장치(120)에 요청(Request)을 전송할 수 있고, 그리고 메모리 장치(120)로부터 요청에 대응하는 응답(Response)을 수신할 수 있다. 예를 들어, 호스트(110)는 메모리 컨트롤러(115)를 통해 요청을 전송할 수 있고, 그리고 요청에 대응하는 응답(예를 들어, 요청에 대응하는 데이터(DATA))을 수신할 수 있다. 도 1에서, 호스트(110)는 메모리 시스템(100) 내에 포함되는 것으로 도시되었지만, 이는 예시적인 것이며, 본 발명의 범위는 이에 한정되지 않는다. 예를 들어, 호스트(110)는 메모리 시스템(100) 외부에 위치할 수 있다.
- [0013] 메모리 컨트롤러(115)는 메모리 장치(120)를 제어할 수 있다. 구체적으로, 메모리 컨트롤러(115)는 복수의 신호들에 기반하여, 메모리 장치(120)를 제어할 수 있다. 예를 들어, 메모리 컨트롤러(115)는 행 어드레스 신호(RA; Row Address), 열 어드레스 신호(CA; Column Address), 또는 제어 신호(CTRL)에 기반하여 메모리 장치(120)를 제어할 수 있다. 어드레스 신호는 행 어드레스 신호(RA) 또는 열 어드레스 신호(CA)를 가리킬 수 있다. 또한, 메모리 컨트롤러(115)는 메모리 장치(120)의 동작에 요구되는 클럭 신호들(CKs)을 메모리 장치(120)에 제공할 수 있다. 예를 들어, 메모리 장치(120)는 복수의 메모리 채널들을 포함하는 다중 채널 구조에 기반하여 구현될 수 있다. 구체적으로, 다중 채널 구조에 포함되는 복수의 메모리 채널들의 각각은 데이터 채널일 수 있다. 즉, 메모리 장치(120)는 복수의 데이터 채널들을 포함하는 다중 채널 구조로 구현될 수 있다. 이 경우, 복수의 메모리 채널들의 각각이 독립적으로 동작하기 위해 메모리 장치(120)는 복수의 클럭 신호들(CKs)이 필요할 수 있다.
- [0014] 일 실시 예에서, 메모리 컨트롤러(115)는 메모리 장치(120)와 데이터(DATA)를 주고받을 수 있다. 예를 들어, 메모리 컨트롤러(115)는 호스트(110)의 요청에 응답하여 메모리 장치(120)에 데이터(DATA)를 쓰거나 또는 데이터(DATA)를 읽을 수 있다. 이 경우, 메모리 컨트롤러(115)와 메모리 장치(120)가 주고받는 데이터(DATA)는 가중치 블록 및 비가중치 블록 중 하나일 수 있다. 도 1에서, 메모리 컨트롤러(115)는 호스트(110) 내에 포함되는 것으로 도시되었지만, 이는 예시적인 것이며, 본 발명의 범위는 이에 한정되지 않는다. 예를 들어, 메모리 컨트롤러(115)는 호스트(110) 외부에 위치할 수 있다.
- [0015] 메모리 장치(120)는 메모리 시스템(100)의 데이터(DATA)를 저장할 수 있다. 구체적으로, 메모리 장치(120)는 메모리 컨트롤러(115)의 제어에 응답하여 동작할 수 있다. 예를 들어, 메모리 장치(120)는 메모리 컨트롤러(115)의 행 어드레스 신호(RA), 열 어드레스 신호(CA), 또는 제어 신호(CTRL)에 응답하여 동작할 수 있다. 또한, 메모리 장치(120)는 호스트(110) 또는 메모리 컨트롤러(115)와 동기(Synchronous)되어 동작할 수 있다. 예를 들어, 메모리 장치(120)는 메모리 컨트롤러(115)로부터 수신되는 클럭 신호들(CKs)에 동기되어 동작할 수 있다.
- [0016] 메모리 장치(120)는 상술한 바와 같이 복수의 메모리 채널들을 포함할 수 있다. 예를 들어, 메모리 장치(120)는 16개의 메모리 채널들을 포함할 수 있다. 메모리 장치(120)의 복수의 메모리 채널들의 각각은 메모리 컨트롤러(115)에 의해 병렬적 및 독립적으로 제어될 수 있다. 예를 들어, 메모리 컨트롤러(115)는 16개의 메모리 채널들의 각각을 병렬적 및 독립적으로 제어할 수 있다.
- [0017] 일 실시 예에서, 메모리 장치(120)는 고-대역폭(High-Bandwidth) 메모리 장치(120)일 수 있다. 예를 들어, 메모리 장치(120)는 HBM(High-Bandwidth Memory) 장치이거나, 또는 HBM을 포함하는 장치일 수 있다. 이하에서, 메모리 장치(120)가 HBM 장치인 것을 기준으로 설명되나, 본 발명의 범위가 이에 한정되는 것은 아니며, 후술하는 실시 예의 기술적 사상이 다른 유형의 메모리 장치(120)에도 적용될 수 있음이 이해되어야 한다.
- [0018] 도 2는 본 기재의 실시 예에 따른 메모리 장치의 동작을 설명하기 위한 도면이다. 도 1 및 도 2를 참조하면, 메모리 시스템(100)은 메모리 컨트롤러(115) 및 제a HBM 채널(200a)을 포함할 수 있다. 상술한 바와 같이, 메모리 장치(120)는 복수의 메모리 채널들을 포함할 수 있다. 도 2에서, 메모리 장치(120)에 포함된 복수의 메모리 채널들 중 제a HBM 채널(200a)만이 예시적으로 도시되었지만, 이는 예시적인 것이며, 본 발명의 범위는 이에 한정되지 않는다.
- [0019] 메모리 컨트롤러(115)는 오류 정정 코드(ECC; Error Correction Code) 선택 로직(117)을 포함할 수 있다. 메모리 컨트롤러(115)는 제a HBM 채널(200a)로 제어 신호(CTRL)를 전송하여 제a HBM 채널(200a)에 데이터(DATA)를 쓰거나 제a HBM 채널(200a)로부터 데이터(DATA)를 읽을 수 있다.
- [0020] 제a HBM 채널(200a)은 오류 정정 코드(ECC) 회로(210) 및 메모리 셀 어레이(220)를 포함할 수 있다. ECC 회로(210)는 메모리 셀 어레이(220)에 쓰일 데이터를 인코딩할 수 있다. 또한, ECC 회로(210)는 메모리 셀 어레이(220)로부터 읽힐 데이터를 디코딩할 수 있다.
- [0021] ECC 회로(210)는 ECC 인코딩 회로(212) 및 칩을 포함할 수 있다. 예를 들어, ECC 회로(210) 내의 ECC 인코딩 회로(212)는 메모리 셀 어레이(220)에 쓰일 데이터를 인코딩하고, 그리고 인코딩된 데이터는 메모리 셀 어레이

(220)에 저장될 수 있다. 또한, ECC 회로(210) 내의 ECC 디코딩 회로(216)는 메모리 셀 어레이(220)에 저장된 인코딩된 데이터를 디코딩할 수 있다.

[0022] 구체적으로, 메모리 컨트롤러(115)는 제a HBM 채널(200a)에 제1 데이터(DATA1)를 쓰기 위해, 제a HBM 채널(200a)로 제어 신호(CTRL), 어드레스 신호 및 제1 데이터(DATA1)를 전송할 수 있다. ECC 인코딩 회로(212)는 메모리 컨트롤러(115)로부터 수신된 제1 데이터(DATA1)를 제2 데이터(DATA2)로 인코딩할 수 있다. 인코딩된 제2 데이터(DATA2)는 메모리 컨트롤러(115)로부터 수신된 어드레스 신호에 대응되는 메모리 셀 어레이(220) 내의 위치에 저장될 수 있다. 즉, 제2 데이터(DATA2)는 ECC 인코딩 회로(212)에 의해 제1 데이터(DATA1)가 인코딩된 데이터일 수 있다.

[0023] 또한, 메모리 컨트롤러(115)는 제a HBM 채널(200a)로부터 제3 데이터(DATA3)를 읽기 위해, 제a HBM 채널(200a)로 제어 신호(CTRL) 및 어드레스 신호를 전송할 수 있다. ECC 디코딩 회로(216)는 메모리 컨트롤러(115)로부터 수신된 어드레스 신호에 대응되는 메모리 셀 어레이(220) 내의 위치에 저장되어 있던 제3 데이터(DATA3)를 제4 데이터(DATA4)로 디코딩할 수 있다. 디코딩된 제4 데이터(DATA4)는 메모리 컨트롤러(115)로 전송될 수 있다. 즉, 제4 데이터(DATA4)는 ECC 디코딩 회로(216)에 의해 제3 데이터(DATA3)가 디코딩된 데이터일 수 있다.

[0024] 본 기재의 실시 예에 따른 ECC 인코딩 회로(212)는 다양한 인코딩 방식에 기반하여 데이터를 인코딩할 수 있다. 구체적으로, ECC 인코딩 회로(212)는 메모리 컨트롤러(115)로부터 수신된 데이터의 유형에 기반하여 상이한 인코딩 방식을 제공할 수 있다. 예를 들어, ECC 인코딩 회로(212)는 메모리 컨트롤러(115)로부터 수신된 데이터가 가중치 블록인 경우 및 비가중치 블록인 경우의 각각에 대하여 상이한 인코딩 방식을 제공할 수 있다. 즉, ECC 인코딩 회로(212)는 인코딩할 데이터가 가중치 블록인 경우, 통상적인 인코딩 방식과 상이한 인코딩 방식에 기반하여 데이터를 인코딩할 수 있다.

[0025] 마찬가지로, 본 기재의 실시 예에 따른 ECC 디코딩 회로(216)는 다양한 디코딩 방식에 기반하여 데이터를 디코딩할 수 있다. 구체적으로 ECC 디코딩 회로(216)는 메모리 셀 어레이(220)에 저장된 데이터의 유형에 기반하여 상이한 디코딩 방식을 제공할 수 있다. 예를 들어, ECC 디코딩 회로(216)는 메모리 컨트롤러(115)로부터 수신된 데이터가 가중치 블록인 경우 및 비가중치 블록인 경우의 각각에 대하여 상이한 디코딩 방식을 제공할 수 있다. 즉, ECC 디코딩 회로(216)는 디코딩할 데이터가 인코딩된 가중치 블록인 경우, 통상적인 디코딩 방식과 상이한 디코딩 방식에 기반하여 데이터를 디코딩할 수 있다.

[0026] 메모리 컨트롤러(115)가 제a HBM 채널(200a)에 데이터를 쓰거나 또는 제a HBM 채널(200a)로부터 데이터를 읽을 때, 메모리 컨트롤러(115) 내의 ECC 선택 로직(117)은 데이터의 유형(즉, 데이터가 가중치 블록인지 여부)에 대응되는 인코딩 방식 또는 디코딩 방식을 요청하는 제어 신호(CTRL)를 생성할 수 있다. 예를 들어, 제어 신호(CTRL)에 응답하여, ECC 인코딩 회로(212)는 제1 데이터(DATA1)가 가중치 블록인 경우, 제1 데이터(DATA1)가 비가중치 블록인 경우와 상이한 인코딩 방식에 기반하여 제1 데이터(DATA1)를 제2 데이터(DATA2)로 인코딩할 수 있다. 다른 예로, 제어 신호(CTRL)에 응답하여, ECC 디코딩 회로(216)는 제3 데이터(DATA3)가 인코딩된 가중치 블록인 경우, 제3 데이터(DATA3)가 인코딩된 비가중치 블록인 경우와 상이한 디코딩 방식에 기반하여 제3 데이터(DATA3)를 제4 데이터(DATA4)로 인코딩할 수 있다.

[0027] 도 3a 및 도 3b는 본 기재의 실시 예에 따른 오류 정정 코드 인코딩 회로 및 오류 정정 코드 디코딩 회로를 보여주는 도면이다. 도 1, 도 2, 및 도 3a를 참조하면, ECC 인코딩 회로(212)는 제1 인코더(213), 제2 인코더(214), 제1 멀티플렉서(MUX1), 및 제1 디멀티플렉서(DEMUX1)를 포함할 수 있다. ECC 인코딩 회로(212)는 메모리 컨트롤러(115)가 제a HBM 채널(200a)에 데이터를 쓰기 위해 제a HBM 채널(200a)로 제어 신호(CTRL) 및 데이터를 전송함에 따라 동작할 수 있다.

[0028] 메모리 컨트롤러(115)로부터 수신된 제어 신호(CTRL)에 응답하여, 제1 디멀티플렉서(DEMUX1)는 메모리 컨트롤러(115)로부터 수신된 제1 데이터(DATA1)를 제1 인코더(213) 또는 제2 인코더(214)로 전송할 수 있다. 예를 들어, 제1 데이터(DATA1)가 가중치 블록인 경우, 제1 디멀티플렉서(DEMUX1)는 제1 데이터(DATA1)를 제1 인코더(213)로 전송할 수 있고, 그리고 제1 데이터(DATA1)가 비가중치 블록인 경우, 제1 디멀티플렉서(DEMUX1)는 제1 데이터(DATA1)를 제2 인코더(214)로 전송할 수 있다.

[0029] 제1 인코더(213)는 제1 디멀티플렉서(DEMUX1)로부터 수신된 제1 데이터(DATA1)를 4-비트 델타 인코딩 방식 및 5-비트 델타 인코딩 방식 중 하나에 기반하여 제2 데이터(DATA2)로 인코딩할 수 있다. 인코딩된 제2 데이터(DATA2)는 제1 멀티플렉서(MUX1)를 통해 메모리 셀 어레이(220)에 저장될 수 있다. 제1 인코더(213)의 동작에 대한 상세한 설명은 이하 도 4a 내지 도 8을 참조하여 후술된다.

- [0030] 제2 인코더(214)는 제1 디멀티플렉서(DEMUX1)로부터 수신된 제1 데이터(DATA1)를 통상적인 인코딩 방식에 기반하여 인코딩할 수 있다. 예를 들어, 제2 인코더(214)는 RS(Reed-Solomon) 코드 인코더, 해밍(Hamming) 코드 인코더 등에 기반하여 구현될 수 있다.
- [0031] 도 1, 도 2, 및 도 3b를 참조하면, ECC 디코딩 회로(216)는 제1 디코더(217), 제2 디코더(218), 제2 멀티플렉서(MUX2), 및 제2 디멀티플렉서(DEMUX2)를 포함할 수 있다. ECC 디코딩 회로(216)는 메모리 컨트롤러(115)가 제a HBM 채널(200a)로부터 데이터를 읽기 위해 제a HBM 채널(200a)로 제어 신호(CTRL)를 전송함에 따라 동작할 수 있다.
- [0032] 메모리 컨트롤러(115)로부터 수신된 제어 신호(CTRL)에 응답하여, 제2 디멀티플렉서(DEMUX2)는 메모리 셀 어레이(220)에 저장된 제3 데이터(DATA3)를 제1 디코더(217) 또는 제2 디코더(218)로 전송할 수 있다. 예를 들어, 제3 데이터(DATA3)가 인코딩된 가중치 블록인 경우, 제2 디멀티플렉서(DEMUX2)는 제3 데이터(DATA3)를 제1 디코더(217)로 전송할 수 있고, 그리고 제3 데이터(DATA3)가 인코딩된 비가중치 블록인 경우, 제2 디멀티플렉서(DEMUX2)는 제3 데이터(DATA3)를 제2 디코더(218)로 전송할 수 있다.
- [0033] 제1 디코더(217)는 제2 디멀티플렉서(DEMUX2)로부터 수신된 제3 데이터(DATA3)를 4-비트 델타 디코딩 방식 및 5-비트 델타 디코딩 방식 중 하나에 기반하여 제4 데이터(DATA4)로 디코딩할 수 있다. 디코딩된 제4 데이터(DATA4)는 제2 멀티플렉서(MUX2)를 통해 메모리 컨트롤러(115)로 전송될 수 있다. 제1 디코더(217)의 동작에 대한 상세한 설명은 제1 인코더(213)의 동작에 대한 상세한 설명과 함께 후술된다.
- [0034] 제2 디코더(218)는 제2 디멀티플렉서(DEMUX2)로부터 수신된 제3 데이터(DATA3)를 통상적인 디코딩 방식에 기반하여 디코딩할 수 있다. 예를 들어, 제2 디코더(218)는 RS(Reed-Solomon) 코드 디코더, 해밍(Hamming) 코드 디코더 등에 기반하여 구현될 수 있다.
- [0035] 도 4a 내지 도 4c는 본 기재의 실시 예에 따른 가중치 블록을 설명하기 위한 도면이다. 도 1, 도 2, 도 3a, 도 3b, 및 도 4a를 참조하면, 메모리 컨트롤러(115) 및 제a HBM 채널(200a)이 주고받는 데이터는 1024-비트 가중치 블록(WB; Weight Block)일 수 있다. 예를 들어, DNN 추론에서 사용되는 본 발명의 실시 예에 따른 HBM 장치가 메모리 컨트롤러(115)와 주고받는 가중치 블록(WB)은 16-비트 부동 소수점(16-bit Floating Point) 형식으로 구현된 제1 내지 제64 가중치 데이터들(WD0~WD63)을 포함하는 가중치 블록(WB)일 수 있다. 즉, 메모리 컨트롤러(115)와 메모리 장치(120)가 주고받는 데이터가 가중치 블록(WB)인 경우, 데이터는 1024-비트로 구현된 가중치 블록(WB)일 수 있다.
- [0036] 가중치 블록(WB)은 제1 내지 제64 가중치 데이터들(WD0~WD63) 및 128-비트 패리티(Parity)를 포함할 수 있다. 예를 들어, 제a HBM 채널(200a)의 ECC 회로(210)에 의해 가중치 블록(WB)이 인코딩되기 전, 가중치 블록(WB)의 128-비트 패리티는 비어있을 수 있다. 또한, 제a HBM 채널(200a)의 ECC 회로(210)에 의해 메모리 셀 어레이(220)에 저장된 인코딩된 가중치 블록(EWB)이 디코딩되면, 디코딩된 가중치 블록(WB)의 128-비트 패리티는 비어있을 수 있다.
- [0037] 다른 예로, 제a HBM 채널(200a)의 ECC 회로(210)에 의해 가중치 블록(WB)이 인코딩되면, 인코딩된 가중치 블록(EWB)의 128-비트 패리티에는 오류 검출 및 정정을 위한 정보가 저장될 수 있다. 인코딩된 가중치 블록(EWB)의 128-비트 패리티에 저장된 정보에 의해 가중치 블록(WB)에 대한 데이터 무결성이 보장될 수 있다.
- [0038] 도 1, 도 2, 도 3a, 도 3b, 도 4a, 및 도 4b를 참조하면, 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각은 16-비트 부동 소수점 형식으로 구현될 수 있다. 구체적으로, 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각은 1-비트의 부호(Sign), 8-비트의 지수(Exponent[7:0]), 및 7-비트의 가수(Mantissa[6:0])를 포함할 수 있다.
- [0039] 부호는 가중치 데이터가 양수인지 음수인지 여부를 나타낼 수 있다. 예를 들어, 부호 값이 0인 경우, 가중치 데이터는 양수이고, 그리고 부호 값이 1인 경우, 가중치 데이터는 음수일 수 있다.
- [0040] 지수는 가중치 데이터의 크기 및 범위를 나타낼 수 있다. 예를 들어, 지수 값의 바이어스(bias)를 127로 설정하는 경우, 지수 값에 기반하여 가중치 데이터의 절대값은 최대 2^{128} (즉, $2^{255-127}=2^{128}$)부터 최소 2^{-126} (즉, $2^{1-127}=2^{-126}$)까지의 값을 가질 수 있다.
- [0041] 가수는 가중치 데이터의 소수점 이하 값을 나타낼 수 있다. 예를 들어, 가수에 기반하여, 메모리 시스템(100)의 가중치 데이터에 대한 연산의 정확도가 향상될 수 있다.
- [0042] DNN 추론에서 사용되는 본 발명의 실시 예에 따른 HBM 장치에서, 가중치 데이터에서 오류가 발생하는 경우, 오

류가 발생한 위치에 따라 DNN 추론의 정확도가 저하되는 정도는 상이할 수 있다. 예를 들어, 가수에서 발생하는 비트 오류에 의해 저하되는 DNN 추론의 정확도는 부호 또는 지수에서 발생하는 비트 오류에 의해 저하되는 DNN 추론의 정확도에 비해 작을 수 있다. 따라서, 가수에서 발생하는 비트 오류에 비해 부호 또는 지수에서 발생하는 비트 오류를 보다 강력하게 정정하는 오류 정정 코드(ECC) 기법은 높은 에너지 효율 및 높은 데이터 무결성을 제공할 수 있다.

[0043] 도 1, 도 2, 도 3a, 도 3b, 도 4a, 도 4b, 및 도 4c를 참조하면, 학습된 DNN 모델들의 각각의 가중치 블록(WB)에 대하여, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이에 따른 가중치 블록(WB)들의 분포가 예시적으로 도시된다. 도 4c에서, 가로축은 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이를 가리키고, 그리고 세로축은 가중치 블록(WB)들의 개수 비율을 가리킨다.

[0044] 도 4c를 참조하면, 가중치 블록(WB)들의 각각에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이는 92% 이상이 16 미만일 수 있다. 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이가 16 미만인 경우, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이들 또한 16 미만일 수 있다. 즉, 이 경우, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이들은 4-비트로 전부 표현 가능할 수 있다.

[0045] 본 기재의 실시 예에 따른 HBM 장치의 ECC 회로(210)는 메모리 컨트롤러(115)와 주고받는 데이터가 가중치 블록(WB)이고, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이가 16 미만인 경우, 제1 인코더(213)의 4-비트 델타 인코딩 방식에 기반하여 가중치 블록(WB)을 인코딩할 수 있고, 그리고 제1 디코더(217)의 4-비트 디코딩 방식에 기반하여 인코딩된 가중치 블록(EWB)을 디코딩할 수 있다.

[0046] 또한, 가중치 블록(WB)들의 각각에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이는 97% 이상이 32 미만일 수 있다. 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이가 32 미만인 경우, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이들 또한 32 미만일 수 있다. 즉, 이 경우, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이들은 5-비트로 전부 표현 가능할 수 있다.

[0047] 본 기재의 실시 예에 따른 HBM 장치의 ECC 회로(210)는 메모리 컨트롤러(115)와 주고받는 데이터가 가중치 블록(WB)이고, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이가 16 이상인 경우, 제1 인코더(213)의 5-비트 델타 인코딩 방식에 기반하여 가중치 블록(WB)을 인코딩할 수 있고, 그리고 제1 디코더(217)의 5-비트 디코딩 방식에 기반하여 인코딩된 가중치 블록(EWB)을 디코딩할 수 있다.

[0048] 도 5는 본 기재의 실시 예에 따른 4-비트 델타 인코딩에 기반하여 인코딩된 가중치 데이터를 보여주는 도면이다. 도 5를 참조하면, 제k 인코딩된 가중치 데이터(EWDk)는 3개의 1-비트 부호, 4-비트 델타($\Delta[3:0]$), 3-비트 패리티($P[2:0]$), 7-비트 가수, 및 1-비트 잔여 비트(RM)를 포함할 수 있다.

[0049] 도 1, 도 2, 도 3a, 도 3b, 도 4a, 도 4b, 및 도 5를 참조하면, 제k 인코딩된 가중치 데이터(EWDk)는 18-비트로 구현될 수 있다. 구체적으로, 16-비트 부동 소수점 형식으로 구현된 제k 가중치 데이터(WDk)가 4-비트 델타 인코딩에 기반하여 인코딩된 제k 인코딩된 가중치 데이터(EWDk)는 가중치 블록(WB)의 128-비트 패리티 중 2-비트 패리티를 더 포함하여 18-비트로 구현될 수 있다.

[0050] 제k 인코딩된 가중치 데이터(EWDk)는 제k 가중치 데이터(WDk)의 8-비트의 지수를 포함하지 않을 수 있다. 대신, 제k 인코딩된 가중치 데이터(EWDk)는 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제k 가중치 데이터(WDk)의 지수의 차이(E_{delta})를 저장하는 4-비트 델타($\Delta[3:0]$)를 포함할 수 있다. 즉, 제1 인코더(213)는 제k 가중치 데이터(WDk)의 지수를 저장하는 대신, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제k 가중치 데이터(WDk)의 지수의 차이(E_{delta})를 4-비트 델타($\Delta[3:0]$)에 저장할 수 있다. 또한, 제1 인코더(213)는 4-비트 델타

(Delta[3:0])에 대하여 단일 오류 정정(SEC; Single Error Correction) 기법을 적용할 수 있다. 구체적으로, 제1 인코더(213)는 제k 인코딩된 가중치 데이터(EWDk)의 4-비트 델타(Delta[3:0]) 및 3-비트 패리티에 기반하여 단일 오류 정정(SEC) 기법을 적용함으로써, 제k 가중치 데이터(WDk)의 지수에 대하여 데이터 무결성을 보장할 수 있다.

[0051] 제k 인코딩된 가중치 데이터(EWDk)에는 제k 가중치 데이터(WDk)의 1-비트 부호가 2회 더 저장될 수 있다. 따라서, 제k 인코딩된 가중치 데이터(EWDk)에는 제k 가중치 데이터(WDk)의 1-비트 부호가 총 3회 저장될 수 있다. 구체적으로, 제1 인코더(213)는 제k 가중치 데이터(WDk)의 1-비트 부호를 2회 더 저장함으로써, 제k 가중치 데이터(WDk)의 부호에 대하여 데이터 무결성을 보장할 수 있다.

[0052] 제k 인코딩된 가중치 데이터(EWDk)는 잔여 비트(RM)를 1-비트 포함할 수 있다. 잔여 비트(RM)에 대한 상세한 설명은 이하 도 6을 참조하여 후술한다.

[0053] 도 6은 본 기재의 실시 예에 따른 4-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록을 보여주는 도면이다. 도 1, 도 2, 도 3a, 도 3b, 도 4a, 도 4b, 도 5, 및 도 6을 참조하면, 인코딩된 가중치 블록(EWB)은 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)을 포함할 수 있다.

[0054] 전술한 바와 같이, 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)의 각각은 잔여 비트(RM)를 1-비트 포함할 수 있다. 따라서, 인코딩된 가중치 블록(EWB)은 총 64-비트의 잔여 비트(RM)를 포함할 수 있다. 64-비트의 잔여 비트(RM)에는 1-비트 플래그(Flag) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax)이 저장될 수 있다. 예를 들어, 64-비트의 잔여 비트(RM)에는 1-비트 플래그(Flag) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax)의 각각이 여러 번 반복되어 저장될 수 있다.

[0055] 여기서, 1-비트 플래그(Flag)는 인코딩된 가중치 블록(EWB)이 제1 인코더(213)에 의해 인코딩된 방식을 나타낼 수 있다. 예를 들어, 제1 인코더(213)가 4-비트 델타 인코딩에 기반하여 가중치 블록(WB)을 인코딩한 경우, 1-비트 플래그(Flag)의 값은 0이고, 제1 인코더(213)가 5-비트 델타 인코딩에 기반하여 가중치 블록(WB)을 인코딩한 경우, 1-비트 플래그(Flag)의 값은 1일 수 있다. 다른 예로, 제1 인코더(213)가 4-비트 델타 인코딩에 기반하여 가중치 블록(WB)을 인코딩한 경우, 1-비트 플래그(Flag)의 값은 1이고, 제1 인코더(213)가 5-비트 델타 인코딩에 기반하여 가중치 블록(WB)을 인코딩한 경우, 1-비트 플래그(Flag)의 값은 0일 수 있다.

[0056] 구체적으로, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)의 64-비트의 잔여 비트(RM)에 1-비트 플래그(Flag)를 여러 번 반복하여 저장할 수 있다. 또한, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)의 64-비트의 잔여 비트(RM)에 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax)을 여러 번 반복하여 저장할 수 있다. 예를 들어, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)의 64-비트의 잔여 비트(RM)에 1-비트 플래그(Flag) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax)의 각각을 7회 반복하여 저장할 수 있다.

[0057] 이 경우, 인코딩된 가중치 블록(EWB)의 64-비트의 잔여 비트(RM) 중 7-비트에는 1-비트 플래그(Flag)가 7회 반복되어 저장되고, 그리고 64-비트의 잔여 비트(RM) 중 56-비트에는 8-비트의 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax)이 7회 반복되어 저장될 수 있다. 또한, 인코딩된 가중치 블록(EWB)의 64-비트의 잔여 비트(RM) 중 1-비트는 비어있을 수 있다.

[0058] 구체적으로, 제1 인코더(213)는 1-비트 플래그(Flag)를 7회 반복하여 저장함으로써, 인코딩된 가중치 블록(EWB)이 인코딩된 방식에 대하여 데이터 무결성을 보장할 수 있다. 또한, 제1 인코더(213)는 8-비트의 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax)을 7회 반복하여 저장함으로써 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수에 대하여 데이터 무결성을 보장할 수 있다.

[0059] 전술한 바와 같이, 인코딩된 가중치 블록(EWB)의 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)의 각각의 4-비트 델타(Delta[3:0])에는 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(Edelta)가 저장될 수 있다. 따라서, 인코딩된 가중치 블록(EWB)에 저장된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax) 및 인코딩된 가중치 블록(EWB)의 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)의 각각의 4-비트 델타(Delta[3:0])에 저장된 값에 기반하여 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수에 대하여 데이터 무결성이 보장될 수 있다.

[0060] 구체적으로, 제1 인코더(213)에 의해 4-비트 델타(Delta[3:0]) 인코딩에 기반하여 인코딩된 가중치 블록(EWB)에는 부호, 플래그(Flag), 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(Emax)이 여러 번 저장될 수

있다. 또한, 제1 인코더(213)에 의해 4-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록(EWB)에는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(E_{delta})가 저장될 수 있다. 여기서, 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(E_{delta})에 단일 오류 정정(SEC) 기법이 적용될 수 있다. 따라서, 제1 인코더(213)에 의해 4-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록(EWB)은 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 부호 및 지수에 대하여 데이터 무결성을 보장할 수 있다.

[0061] 또한, 메모리 시스템(100)은 이러한 데이터 무결성이 보장된 인코딩된 가중치 블록(EWB)이 저장된 메모리 셀 어레이(220)에 대한 리프레쉬 주기를 증가시킬 수 있다. 즉, 메모리 셀 어레이(220)에 저장된 데이터에 대한 무결성(또는, 정확도)이 보장되는 경우, 메모리 시스템(100)은 무결성(또는, 정확도)이 보장된 데이터에 대하여 리프레쉬 동작을 적게 수행함으로써, 리프레쉬 오버헤드를 줄일 수 있고, 그리고 높은 에너지 효율을 제공할 수 있다.

[0062] 도 7는 본 기재의 실시 예에 따른 5-비트 델타 인코딩에 기반하여 인코딩된 가중치 데이터를 보여주는 도면이다. 도 7을 참조하면, 제k 인코딩된 가중치 데이터(EWDk)는 3개의 1-비트 부호, 5-비트 델타($\Delta[4:0]$), 3-비트 패리티($P[2:0]$), 6-비트 가수, 및 1-비트 잔여 비트(RM)를 포함할 수 있다.

[0063] 도 1, 도 2, 도 3a, 도 3b, 도 4a, 도 4b, 및 도 7을 참조하면, 제k 인코딩된 가중치 데이터(EWDk)는 18-비트로 구현될 수 있다. 구체적으로, 16-비트 부동 소수점 형식으로 구현된 제k 가중치 데이터(WDk)가 5-비트 델타 인코딩에 기반하여 인코딩된 제k 인코딩된 가중치 데이터(EWDk)는 가중치 블록(WB)의 128-비트 패리티 중 2-비트 패리티를 더 포함하여 18-비트로 구현될 수 있다.

[0064] 제k 인코딩된 가중치 데이터(EWDk)는 제k 가중치 데이터(WDk)의 8-비트의 지수를 포함하지 않을 수 있다. 대신, 제k 인코딩된 가중치 데이터(EWDk)는 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제k 가중치 데이터(WDk)의 지수의 차이(E_{delta})를 저장하는 5-비트 델타($\Delta[4:0]$)를 포함할 수 있다. 즉, 제1 인코더(213)는 제k 가중치 데이터(WDk)의 지수를 저장하는 대신, 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제k 가중치 데이터(WDk)의 지수의 차이(E_{delta})를 5-비트 델타($\Delta[4:0]$)에 저장할 수 있다. 또한, 제1 인코더(213)는 5-비트 델타($\Delta[4:0]$)의 상위 4-비트에 대하여 단일 오류 정정(SEC) 기법을 적용할 수 있다. 구체적으로, 제1 인코더(213)는 제k 인코딩된 가중치 데이터(EWDk)의 5-비트 델타($\Delta[4:0]$)의 상위 4-비트 및 3-비트 패리티에 기반하여 단일 오류 정정(SEC) 기법을 적용함으로써, 제k 가중치 데이터(WDk)의 지수에 대하여 데이터 무결성을 보장할 수 있다.

[0065] 제k 인코딩된 가중치 데이터(EWDk)에는 제k 가중치 데이터(WDk)의 1-비트 부호가 2회 더 저장될 수 있다. 따라서, 제k 인코딩된 가중치 데이터(EWDk)에는 제k 가중치 데이터(WDk)의 1-비트 부호가 총 3회 저장될 수 있다. 구체적으로, 제1 인코더(213)는 제k 가중치 데이터(WDk)의 1-비트 부호를 2회 더 저장함으로써, 제k 가중치 데이터(WDk)의 부호에 대하여 데이터 무결성을 보장할 수 있다.

[0066] 제k 인코딩된 가중치 데이터(EWDk)는 제k 가중치 데이터(WDk)의 8-비트의 지수 중 상위 7-비트 지수를 포함할 수 있다. 즉, 제1 인코더(213)는 제k 가중치 데이터(WDk)의 8-비트의 지수 중 하위 1-비트를 삭제할 수 있다. 제k 인코딩된 가중치 데이터(EWDk)는 잔여 비트(RM)를 1-비트 포함할 수 있다. 잔여 비트(RM)에 대한 상세한 설명은 이하 도 8을 참조하여 후술한다.

[0067] 도 8은 본 기재의 실시 예에 따른 5-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록을 보여주는 도면이다. 도 1, 도 2, 도 3a, 도 3b, 도 4a, 도 4b, 도 7, 및 도 8을 참조하면, 인코딩된 가중치 블록(EWB)은 제1 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)을 포함할 수 있다.

[0068] 도 8의 인코딩된 가중치 블록(EWB)은 5-비트 델타 인코딩에 기반하여 인코딩된 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)로 구현되는 것을 제외하면 도 7의 인코딩된 가중치 블록(EWB)과 동일하게 구성될 수 있다. 즉, 도 8의 인코딩된 가중치 블록(EWB)의 잔여 비트(RM)들은 도 7의 인코딩된 가중치 블록(EWB)의 잔여 비트(RM)들과 동일하게 사용될 수 있다. 따라서, 중복되는 설명은 생략된다.

[0069] 마찬가지로, 제1 인코더(213)에 의해 5-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록(EWB)은 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 부호 및 지수에 대하여 데이터 무결성을 보장할 수 있다.

[0070] 또한, 메모리 시스템(100)은 이러한 데이터 무결성이 보장된 인코딩된 가중치 블록(EWB)이 저장된 메모리 셀 어

레이(220)에 대한 리프레쉬 주기를 증가시킬 수 있다. 즉, 메모리 셀 어레이(220)에 저장된 데이터에 대한 무결성(또는, 정확도)이 보장되는 경우, 메모리 시스템(100)은 무결성(또는, 정확도)이 보장된 데이터에 대하여 리프레쉬 동작을 적게 수행함으로써, 리프레쉬 오버헤드를 줄일 수 있고, 그리고 높은 에너지 효율을 제공할 수 있다.

[0071] 도 9는 본 기재의 실시 예에 따른 오류 정정 코드 회로의 동작 방법의 예를 보여주는 순서도이다. 도 1, 도 2, 도 3a, 도 3b, 도 4a, 도 4b, 및 도 9를 참조하면, S110 단계에서, 메모리 시스템(100)은 메모리 장치(120)로 1024-비트 가중치 블록(WB)을 입력할 수 있다. 예를 들어, 메모리 컨트롤러(115)는 메모리 장치(120)로 1024-비트 가중치 블록(WB)을 전송할 수 있다.

[0072] S120 단계에서, 메모리 시스템(100)은 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이가 16보다 작은지 여부를 판단할 수 있다. 예를 들어, 메모리 장치(120) 내의 ECC 인코딩 회로(212)의 제1 인코더(213)는 가중치 블록(WB)을 인코딩하기 위한 인코딩 방식을 결정하기 위해 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이가 16보다 작은지 여부를 판단할 수 있다.

[0073] 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이가 16보다 작은 경우, S130 단계에서, 메모리 장치(120)는 4-비트 델타 인코딩에 기반하여 1024-비트 가중치 블록(WB)을 인코딩할 수 있다. 예를 들어, 제1 인코더(213)는 4-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록(EWB)을 생성할 수 있다.

[0074] 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 최대값(E_{max}) 및 최소값(E_{min})의 차이가 16보다 큰 경우, S140 단계에서, 메모리 장치(120)는 5-비트 델타 인코딩에 기반하여 1024-비트 가중치 블록(WB)을 인코딩할 수 있다. 예를 들어, 제1 인코더(213)는 5-비트 델타 인코딩에 기반하여 인코딩된 가중치 블록(EWB)을 생성할 수 있다.

[0075] 도 10은 본 기재의 실시 예에 따른 4-비트 델타 인코딩 방법의 예를 보여주는 순서도이다. 도 1, 도 2, 도 3a, 도 3b, 도 4a, 도 4b, 도 5, 도 6, 도 9, 및 도 10을 참조하면, S131 단계에서, ECC 인코딩 회로(212)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 플래그(Flag)를 7회 반복하여 저장할 수 있다. 예를 들어, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)에 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 플래그(Flag)를 7회 반복하여 저장할 수 있다. 여기서, 플래그(Flag)는 전술한 바와 같이 제1 인코더(213)가 가중치 블록(WB)을 인코딩하는 방식에 대응되는 정보를 포함할 수 있다.

[0076] S132 단계에서, ECC 인코딩 회로(212)는 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(Edelta)를 4-비트 델타(Delta[3:0])에 저장할 수 있다. 예를 들어, 제1 인코더(213)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(Edelta)를 인코딩된 가중치 블록(EWB)의 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)의 각각의 4-비트 델타(Delta[3:0])에 저장할 수 있다.

[0077] S133 단계에서, ECC 인코딩 회로(212)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(Edelta)에 대하여 단일 오류 정정(SEC) 기법을 적용할 수 있다. 예를 들어, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)의 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)의 각각의 4-비트 델타(Delta[3:0])에 대하여 단일 오류 정정(SEC) 기법을 적용할 수 있다.

[0078] S134 단계에서, ECC 인코딩 회로(212)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 부호를 2회 더 저장할 수 있다. 예를 들어, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)에 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 1-비트 부호를 2회 더 저장할 수 있다.

[0079] 도 11은 본 기재의 실시 예에 따른 5-비트 델타 인코딩 방법의 예를 보여주는 순서도이다. 도 1, 도 2, 도 3a, 도 3b, 도 4a, 도 4b, 도 5, 도 6, 도 9, 및 도 11을 참조하면, S141 단계에서, ECC 인코딩 회로(212)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 플래그(Flag)를 7회 반복하여 저장할 수 있다. 예를 들어, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)에 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 플래그(Flag)를 7회 반복하여 저장할 수 있다. 여기서, 플래그(Flag)는 전술한 바와 같이 제1 인코더(213)가 가중치 블록(WB)을 인코딩하는 방식에 대응되는 정보를 포함할 수 있다.

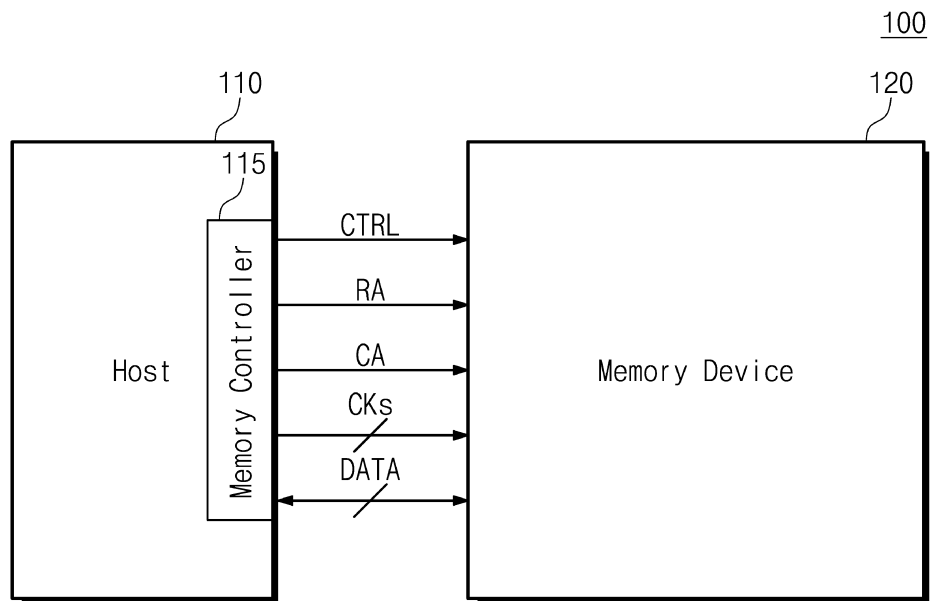
- [0080] S142 단계에서, ECC 인코딩 회로(212)는 가중치 블록(WB)에 포함된 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(E_{delta})를 5-비트 델타($\Delta[4:0]$)에 저장할 수 있다. 예를 들어, 제1 인코더(213)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(E_{delta})를 인코딩된 가중치 블록(EWB)의 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)의 각각의 5-비트 델타($\Delta[4:0]$)에 저장할 수 있다.
- [0081] S143 단계에서, ECC 인코딩 회로(212)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 지수의 최대값(E_{max}) 및 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 지수의 차이(E_{delta}) 중 상위 4-비트에 대하여 단일 오류 정정(SEC) 기법을 적용할 수 있다. 예를 들어, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)의 제0 내지 제64 인코딩된 가중치 데이터들(EWD1~EWD63)의 각각의 5-비트 델타($\Delta[4:0]$) 중 상위 4-비트에 대하여 단일 오류 정정(SEC) 기법을 적용할 수 있다.
- [0082] S144 단계에서, ECC 인코딩 회로(212)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 7-비트 가수 중 하위 1-비트를 삭제할 수 있다. 예를 들어, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)에 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 7-비트 가수 중 하위 1-비트는 삭제하고, 그리고 상위 6-비트를 저장할 수 있다.
- [0083] S145 단계에서, ECC 인코딩 회로(212)는 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 부호를 2회 더 저장할 수 있다. 예를 들어, 제1 인코더(213)는 인코딩된 가중치 블록(EWB)에 제1 내지 제64 가중치 데이터들(WD0~WD63)의 각각의 1-비트 부호를 2회 더 저장할 수 있다.
- [0084] 상술된 내용은 본 발명을 실시하기 위한 구체적인 실시 예들이다. 본 발명은 상술된 실시 예들뿐만 아니라, 단순히 설계 변경되거나 용이하게 변경할 수 있는 실시 예들 또한 포함할 것이다. 또한, 본 발명은 실시 예들을 이용하여 용이하게 변형하여 실시할 수 있는 기술들도 포함될 것이다. 따라서, 본 발명의 범위는 상술된 실시 예들에 국한되어 정해져서는 안되며 후술하는 특허청구범위뿐만 아니라 이 발명의 특허청구범위와 균등한 것들에 의해 정해져야 할 것이다.

부호의 설명

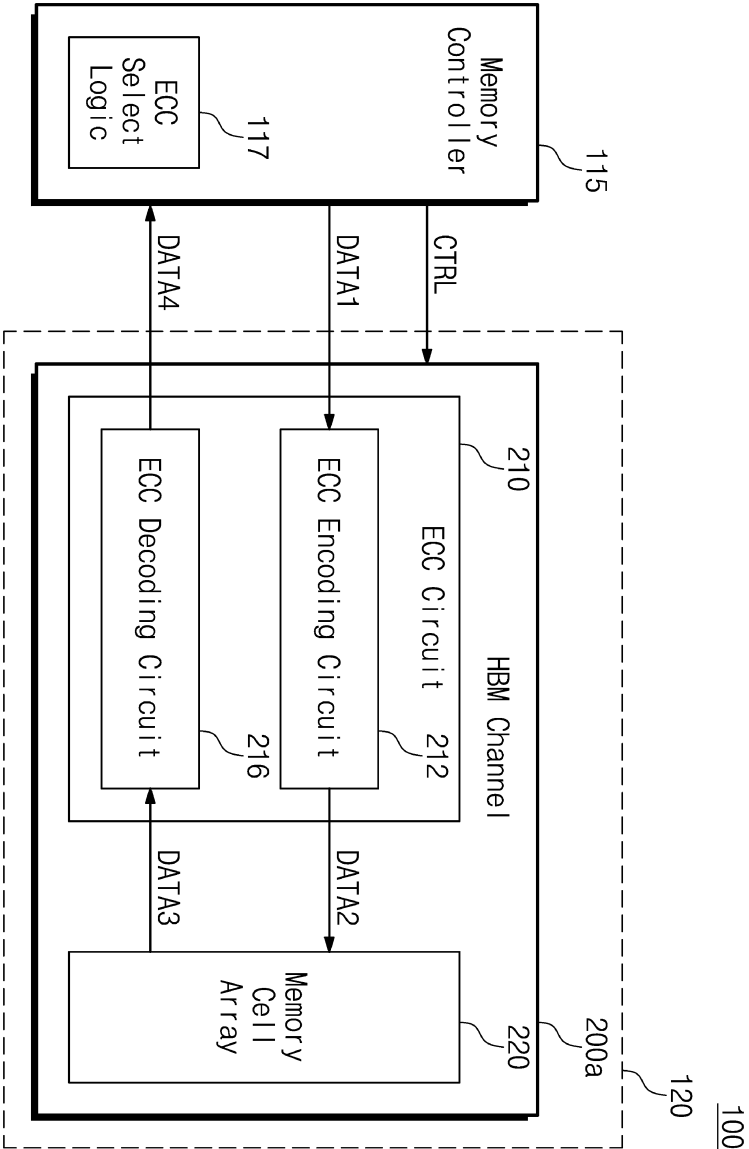
- [0085] 100: 메모리 시스템
- CTRL: 제어 신호
- RA, CA: 행 어드레스 신호 및 열 어드레스 신호
- CKs: 클럭 신호들
- 120: 메모리 장치
- 212: ECC 인코딩 회로
- 216: ECC 디코딩 회로
- WB: 가중치 블록
- WD0~WD63: 제1 내지 제64 가중치 데이터
- EWB: 인코딩된 가중치 블록
- EWD0~EWD63: 제1 내지 제64 인코딩된 가중치 데이터

도면

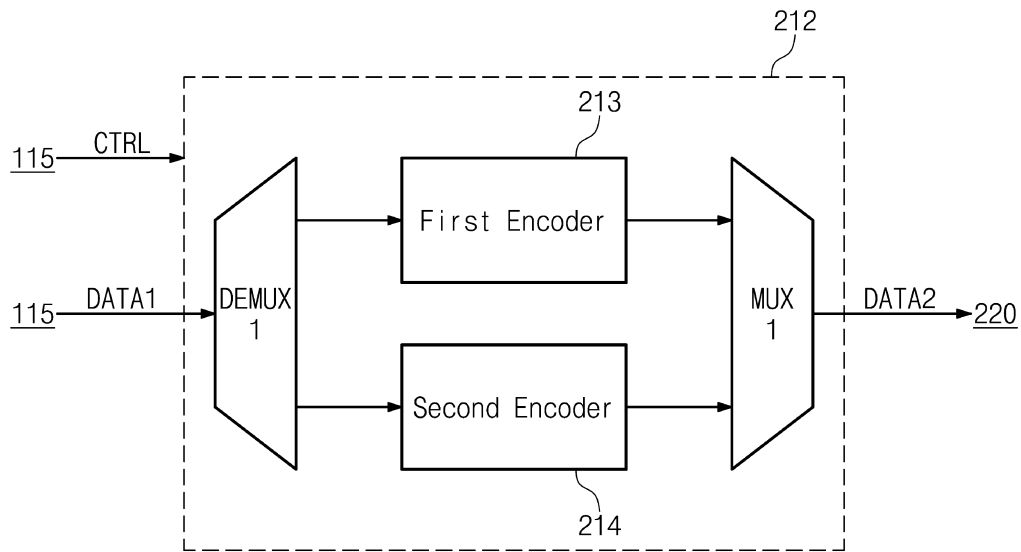
도면1



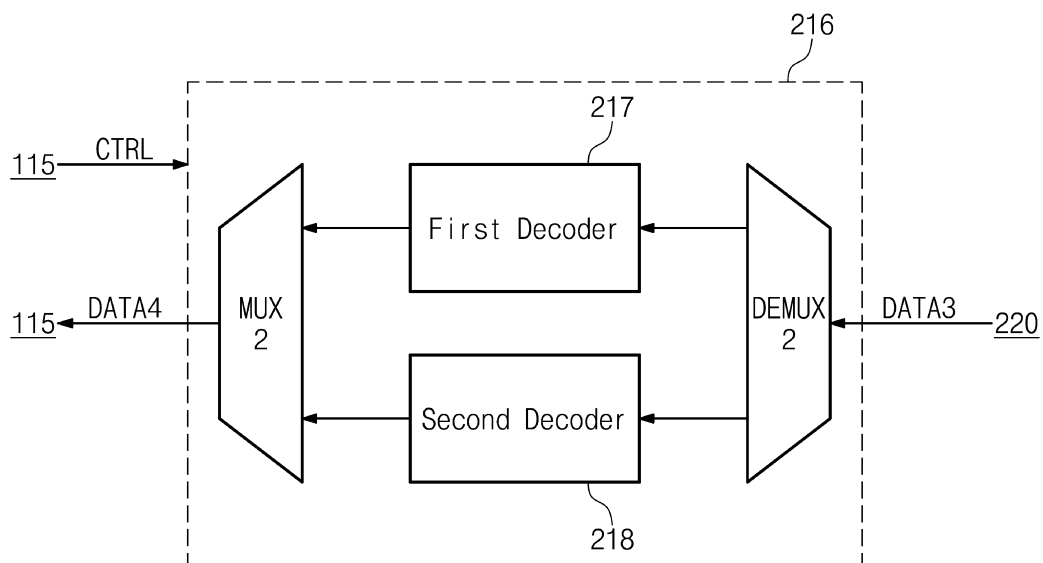
도면2



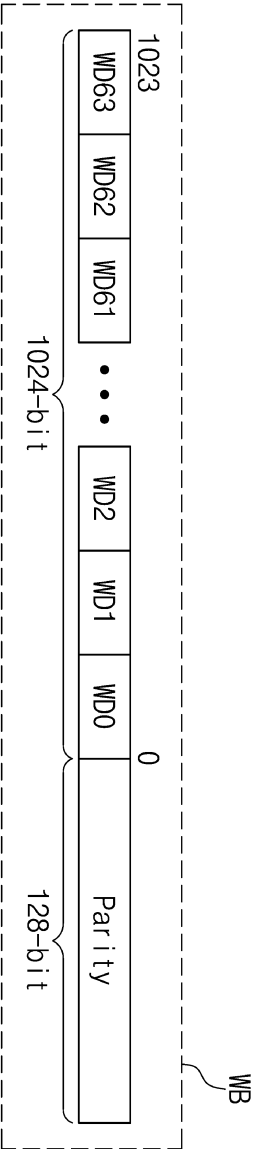
도면3a



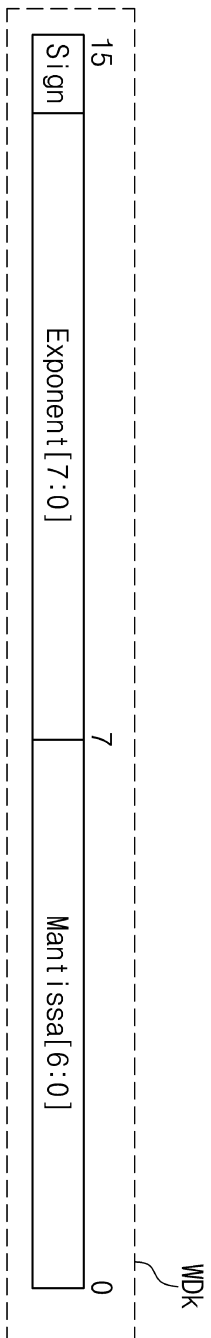
도면3b



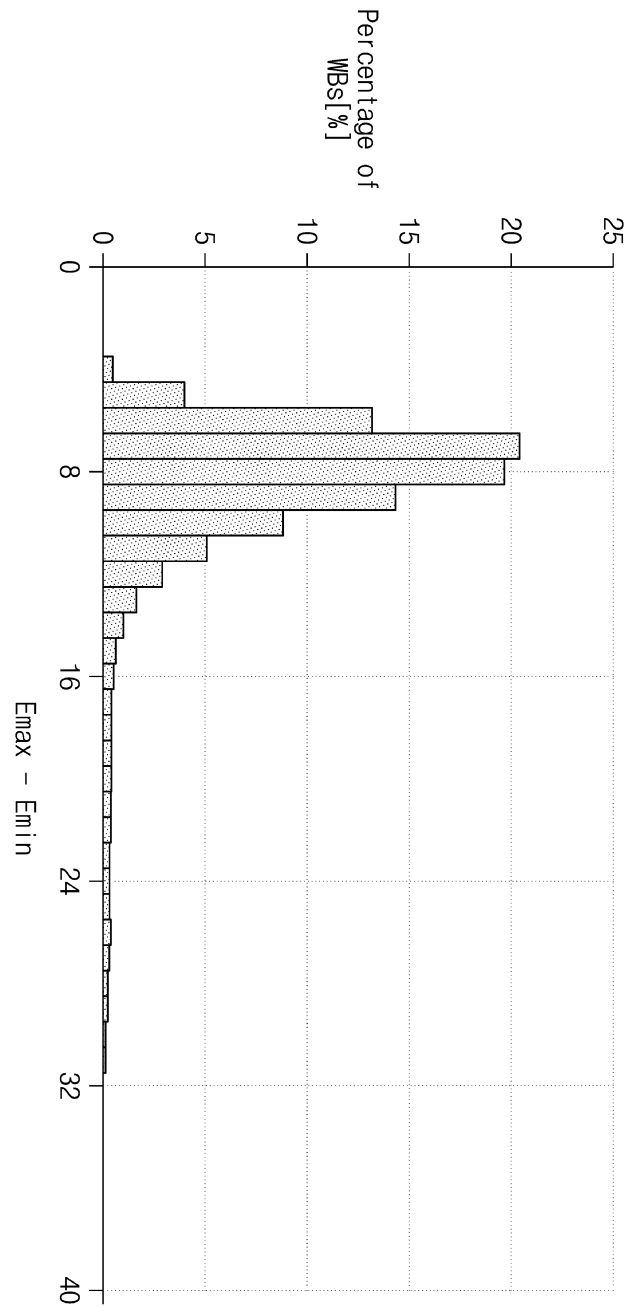
도면4a



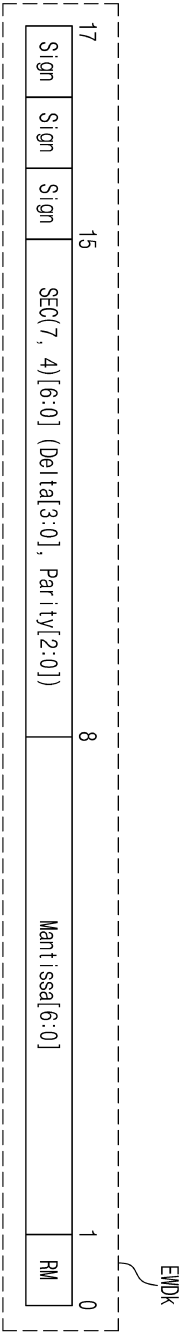
도면4b



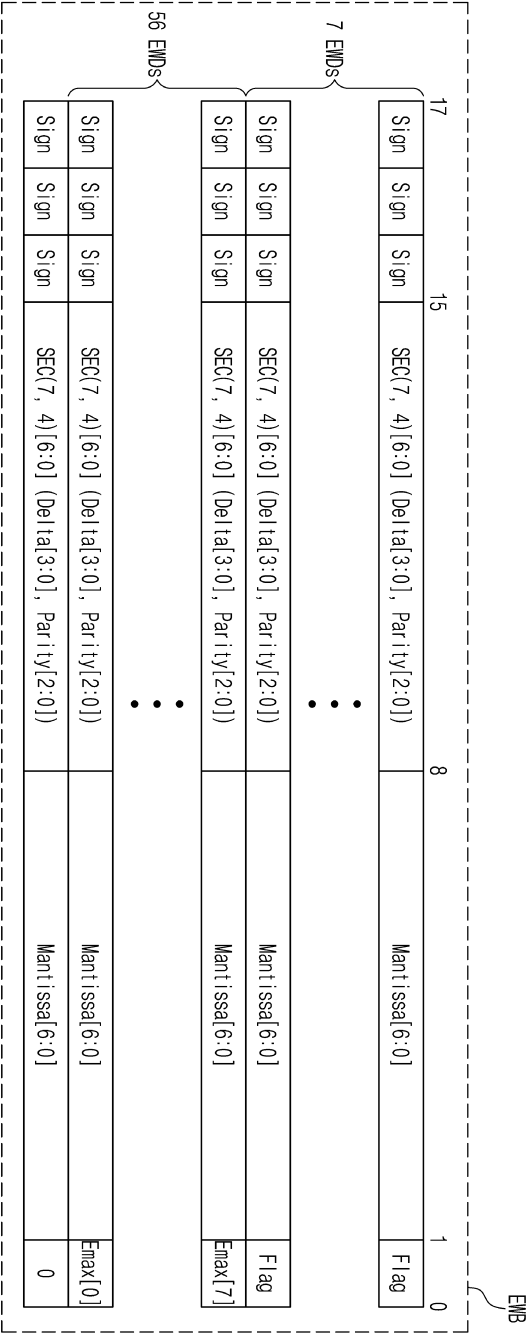
도면4c



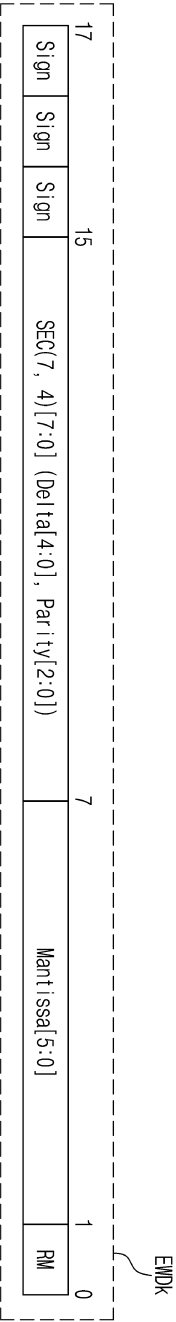
도면5



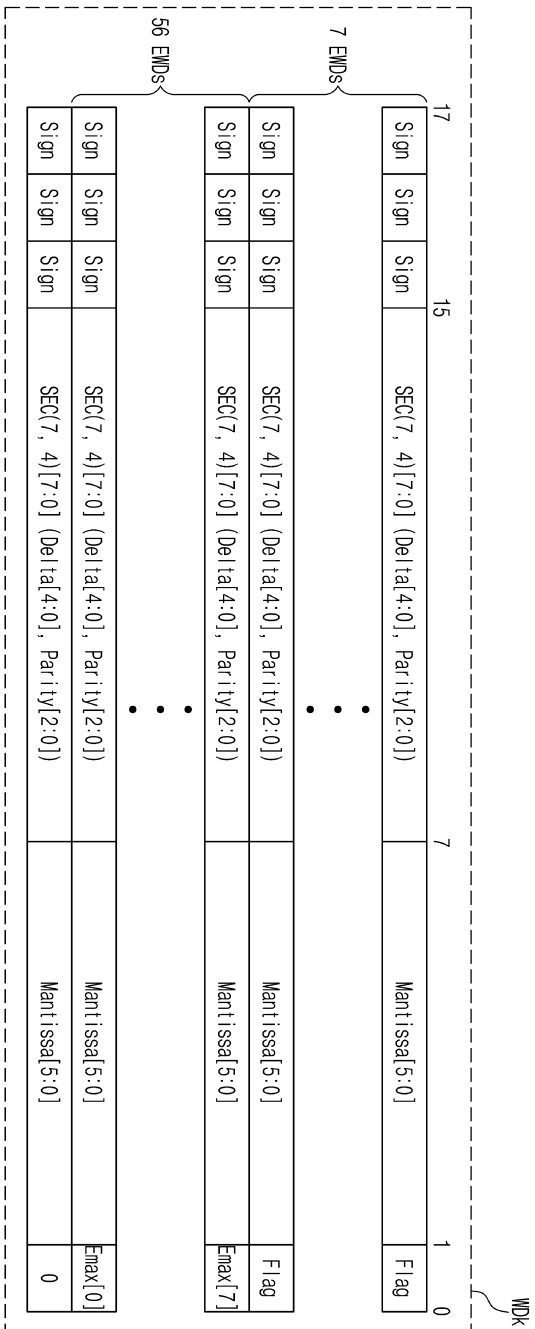
도면6



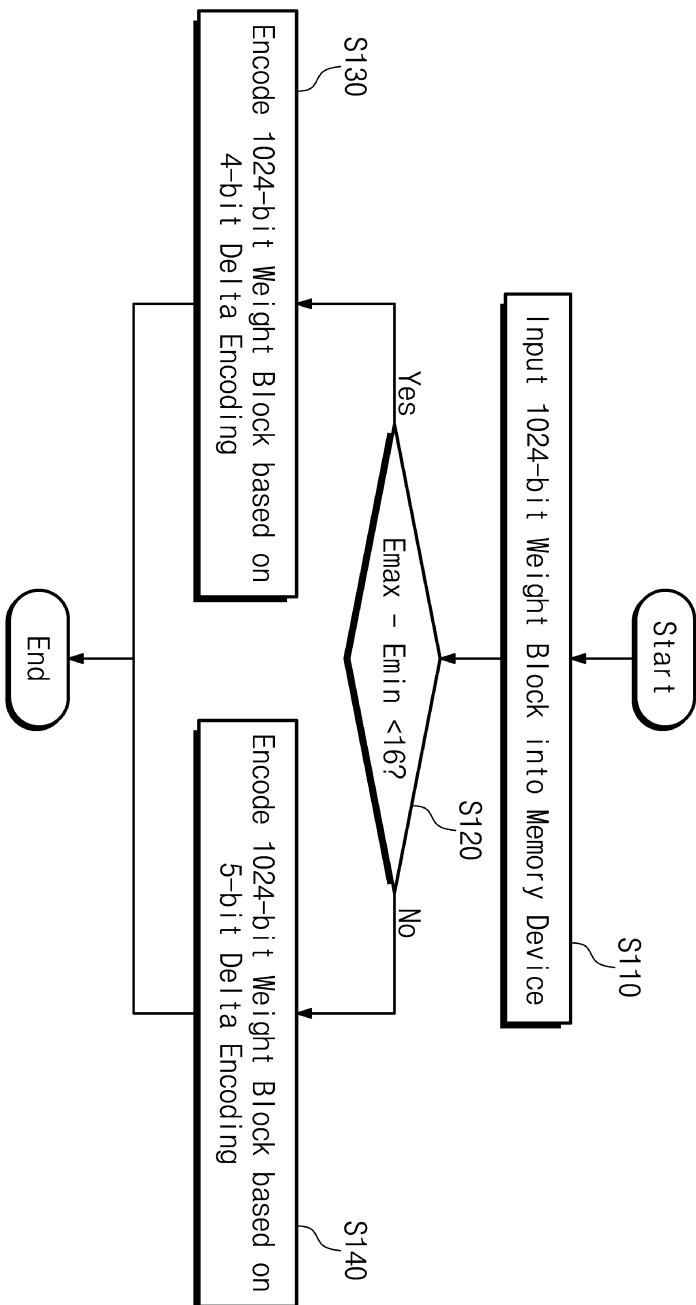
도면7



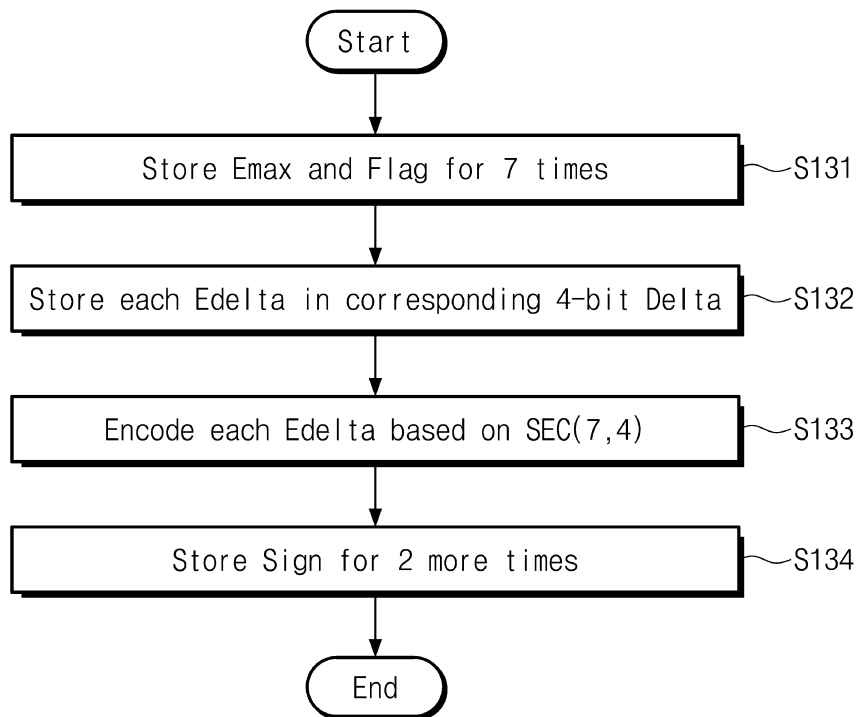
도면8



도면9



도면10



도면11

