(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0086154
(43) 공개일자 2025년06월13일

(51) 국제특허분류(Int. Cl.)
G06N 3/065 (2023.01) *G06F 15/78* (2006.01)
G06F 3/06 (2006.01) *G06F 7/544* (2017.01)
(52) CPC특허분류
G06N 3/065 (2023.01)
G06F 15/7821 (2013.01)
(21) 출원번호 10-2023-0175539
(22) 출원일자 2023년12월06일
심사청구일자 2023년12월06일

(71) 출원인
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
이규호
울산광역시 울주군 언양읍 유니스트길 50
정희창
울산광역시 울주군 언양읍 유니스트길 50
(74) 대리인
특허법인 무한

전체 청구항 수 : 총 12 항

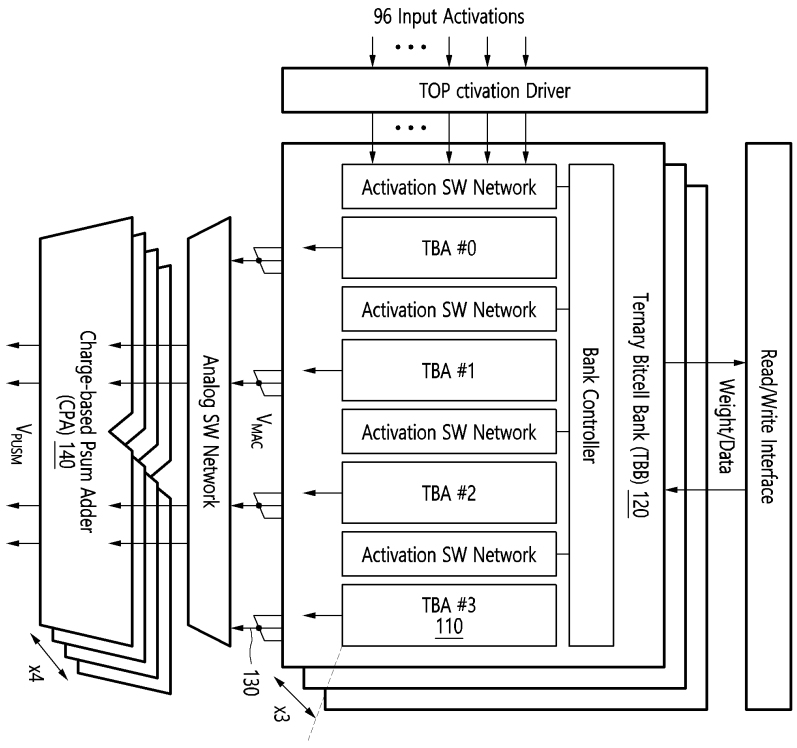
(54) 발명의 명칭 **삼진 뉴럴 네트워크 연산 가속을 위한 컴퓨팅 인 메모리 장치**

(57) 요 약

일 실시예에 따른 삼진 뉴럴 네트워크(Ternary Neural Network, TNN) 연산 가속 장치는, 제1 이진 값 및 제2 이진 값의 조합에 대응하는 삼진 가중치를 가지는 하나 이상의 삼진셀(ternary cell)을 포함하고, 상기 하나 이상의 삼진셀의 각각은, 로직 하이(logic high) 또는 로직 로우(logic low) 중 어느 하나를 지시하는 상기 제1 이진

(뒷면에 계속)

대 표 도 - 도1



값을 가지는 제1 비트 셀; 상기 제1 비트 셀의 인버터 페어(inverter pair)의 양단에 게이트를 통해 연결된 제1 트랜지스터 페어(transistor pair); 로직 하이 또는 로직 로우 중 어느 하나를 지시하는 상기 제2 이진 값을 가지는 제2 비트 셀; 상기 제2 비트 셀의 인버터 페어의 양단에 게이트를 통해 연결되고, 일단에서 상기 제1 트랜지스터 페어와 연결되며, 타단에서 미들 레벨 전압을 수신하는 제2 트랜지스터 페어; 상기 제2 트랜지스터 페어 및 출력 라인 사이에 연결되고, 상기 제1 트랜지스터 페어를 통해 수신된 입력 값과 삼진 가중치 간의 연산 결과를 상기 출력 라인에게 제공하는 연산 커패시터(computation capacitor)를 포함할 수 있다.

(52) CPC특허분류
G06F 3/0659 (2013.01)
G06F 7/5443 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711194743
과제번호	00212222
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구(과기정통부)
연구과제명	소형 인공지능을 위한 복합 다치로직 기반 초저전력, 초고효율 프로세서 구조 개발
기 여 율	70/100
과제수행기관명	울산과학기술원
연구기간	2023.03.01 ~ 2024.02.29

이 발명을 지원한 국가연구개발사업

과제고유번호	1711193271
과제번호	2022-0-01037-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	PIM인공지능반도체핵심기술개발(설계)
연구과제명	DRAM 기반 고성능 PIM 메모리 반도체 기술 개발
기 여 율	30/100
과제수행기관명	한국과학기술원
연구기간	2023.01.01 ~ 2023.12.31

명 세 서

청구범위

청구항 1

삼진 뉴럴 네트워크(Ternary Neural Network, TNN) 연산 가속 장치에 있어서,

제1 이진 값 및 제2 이진 값의 조합에 대응하는 삼진 가중치를 가지는 하나 이상의 삼진셀(ternary cell)

을 포함하고,

상기 하나 이상의 삼진셀의 각각은,

로직 하이(logic high) 또는 로직 로우(logic low) 중 어느 하나를 지시하는 상기 제1 이진 값을 가지는 제1 비트 셀;

상기 제1 비트 셀의 인버터 페어(inverter pair)의 양단에 게이트를 통해 연결된 제1 트랜지스터 페어(transistor pair);

로직 하이 또는 로직 로우 중 어느 하나를 지시하는 상기 제2 이진 값을 가지는 제2 비트 셀;

상기 제2 비트 셀의 인버터 페어의 양단에 게이트를 통해 연결되고, 일단에서 상기 제1 트랜지스터 페어와 연결되며, 타단에서 미들 레벨 전압을 수신하는 제2 트랜지스터 페어;

상기 제2 트랜지스터 페어 및 출력 라인 사이에 연결되고, 상기 제1 트랜지스터 페어를 통해 수신된 입력 값과 삼진 가중치 간의 연산 결과를 상기 출력 라인에게 제공하는 연산 커패시터(computation capacitor)

를 포함하는 TNN 연산 가속 장치.

청구항 2

제1항에 있어서,

상기 제1 비트 셀 및 상기 제2 비트 셀은,

정적 랜덤 액세스 메모리를 포함하는,

TNN 연산 가속 장치.

청구항 3

제1항에 있어서,

상기 하나 이상의 삼진셀은,

상기 제1 이진 값이 로직 하이를 지시하고 상기 제2 이진 값이 로직 로우를 지시하는 것에 기초하여, 제1 삼진 논리 값에 해당하는 삼진 가중치를 저장하는,

TNN 연산 가속 장치.

청구항 4

제1항에 있어서,

상기 하나 이상의 삼진셀은,

상기 제1 이진 값이 로직 하이를 지시하고 상기 제2 이진 값이 로직 하이를 지시하는 것에 기초하여, 제2 삼진 논리 값에 해당하는 삼진 가중치를 저장하는,

TNN 연산 가속 장치.

청구항 5

제1항에 있어서,

상기 하나 이상의 삼진셀은,

상기 제2 이진 값이 로직 로우를 지시하는 것에 기초하여,

제3 삼진 논리 값에 해당하는 삼진 가중치를 저장하는,

TNN 연산 가속 장치.

청구항 6

제1항에 있어서,

상기 하나 이상의 삼진셀의 각각은,

상기 제1 트랜지스터 페어의 트랜지스터들을 통해 수신된 입력 신호들의 전압 값들의 조합에 대응하는 삼진 입력 값 및 상기 삼진 가중치 간의 삼진 연산 결과에 기초하여,

상기 연산 커패시터를 통해 미리 결정된 전압 값에 대응하는 논리 값을 상기 출력 라인에 제공하는

TNN 연산 가속 장치.

청구항 7

제1항에 있어서,

상기 하나 이상의 삼진셀의 각각은,

상기 제2 트랜지스터 페어의 타단에서 수신된 하이 레벨 전압에 응답하여, 상기 연산 커패시터 및 상기 제2 트랜지스터 페어 간의 계산 노드(node)에서, 증가하는 전압이 미리 결정된 임계 전압 값에 도달하는 것에 기초하여,

상기 연산 결과를 출력 라인에게 제공하는

TNN 연산 가속 장치.

청구항 8

제1항에 있어서,

상기 하나 이상의 삼진셀 각각이, 행들(rows) 및 열들(columns)로 배열되는 하나 이상의 삼진 비트셀 어레이(Ternary Bitcell Array, TBA); 및

복수의 출력 라인들

을 포함하고,

상기 복수의 출력 라인들의 각각은,

상기 열들 중 해당하는 열에 배열된 삼진셀들과 연결되는

TNN 연산 가속 장치.

청구항 9

제8항에 있어서,

서로 동일한 개수의 하나 이상의 TBA를 가지는, 하나 이상의 삼진 비트셀 뱅크(Ternary Bitcell Bank, TBB)

를 포함하는,

TNN 연산 가속 장치.

청구항 10

제8항에 있어서,

상기 열들 중 해당하는 열에 배열된 삼진셀들과

출력 라인을 통해 일단에서 연결되고, 타단에서 미들 레벨 전압을 수신하는 스위칭 소자

를 포함하는

TNN 연산 가속 장치.

청구항 11

제8항에 있어서,

상기 열들 중 해당하는 열에 배열된 삼진셀들 각각을 통해 수신된 입력 값과 상기 삼진셀들 중 해당 삼진셀이 가지는 삼진 가중치 간의 연산 결과를 누적한 값에 기초하여, MAC(multiply and accumulate) 결과를 생성하는,

TNN 연산 가속 장치.

청구항 12

제11항에 있어서,

상기 생성된 MAC 결과를 수신하여, 상기 MAC 결과에 해당하는 전압 값에 기초하여 아날로그 영역(analog domain)에서 전하 기반 부분합(charge-based partial sum)을 수행하는 복수의 CPA(charge-based Partial sum Adder) 회로들

을 더 포함하고,

상기 CPA 회로들의 각각은,

MAC 결과에 해당하는 전압 값을 전하로 저장하는 복수의 전하 홀더들(charge holders); 및

상기 복수의 전하 홀더들에 저장된 전하들로부터 부분합 커패시터(partial sum capacitor)로의 순차적 전하 공유에 기초하여, 최종 MAC 결과를 생성하는 아날로그 메모리 회로(analog memory circuit)

를 포함하는,

TNN 연산 가속 장치.

발명의 설명

기 술 분 야

[0001] 이하, 삼진 뉴럴 네트워크(Ternary neural network, TNN)(이하, TNN) 연산 가속을 위한 컴퓨팅 인 메모리 장치 및 방법에 대하여 개시한다.

배 경 기 술

[0002] 최근, 컨볼루션 뉴럴 네트워크(Convolutional Neural Networks, CNNs)는 이미지 및 비디오 처리에 대한 높은 정확도로 인공지능 어플리케이션을 크게 향상했다. CNN은 대량의 데이터 트랜잭션으로 인해 기존의 폰 노이만 아키텍처로 에너지 효율성과 성능을 제한하는 대량의 연산 및 메모리 풋프린트(footprint)를 포함하는 것으로 잘 알려져 있다. 초저전력 IoT 장치들의 유망한 솔루션인 CIM(computing in memory) 아키텍처는 높은 에너지 효율성을 달성하기 위한 온칩 메모리(on-chip memory)의 내부에서 연산함으로써 기존 컴퓨터 아키텍처를 대체할 수 있게 되었다. 여러 개의 트랜지스터들은 간단한 연산들을 지원하기 위해 메모리 비트셀 근처(또는 내부)에 부착된다. CIM은 데이터 버스(data bus)를 통해 메모리로부터 연산 유닛(computation unit)으로의 데이터 가져오기 작업(data fetch operation)을 제거하기 때문에 처리량과 에너지 효율성이 크게 향상되었다. 그러나, 이전 CIM 장치들은 복잡한 작업들에 대한 이미지 분류 정확도가 떨어지는 문제가 있었다.

[0003] 일부 CIM 장치들은 멀티 비트 정확도(multi-bit precision)로 계산함으로써, 높은 분류 정확도를 달성할 수 있다. 그러나 멀티 비트 동작을 위한 반복적인 연산 사이클들 때문에 처리량과 에너지 효율성에서 엄청난 개선을

달성할 수 없었다. 한편, 이진법 신경망(Binary Neural Network, BNN)은 멀티 비트 CNN과 비교하여 분류 정확도가 떨어지지만, 에너지 효율성의 획기적인 개선을 위하여 CIM에 널리 사용된다.

[0004] 최근 연구들은 삼진 신경망(Ternary Neural Network, TNN)(이하, TNN)이 BNN과 멀티비트 CNN 사이의 절충안 중 하나임을 보여준다. TNN은 BNN 보다 이미지 분류의 정확도가 5~10% 더 높은 반면, 멀티비트 CNN에 비해 0.53%의 정확도만 손실된다. 따라서, 참고 논문 S. Jain, S. K. Gupta and A. Raghunathan, "TiM-DNN: Ternary InMemory Accelerator for Deep Neural Networks," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 7, pp. 1567-1577, July 2020에서는 비트셀 어레이에서 MAC(Multiplication and accumulation)이 수행되고 디지털 로직 유닛들(digital logic units)에서 부분합(partial sum, Psum)연산이 수행되는 TNN 가속을 위한 TiM-DNN CIM 장치를 제안했다. 그러나, Psum 연산은 모든 MAC 결과에 대해 아날로그-대-디지털 변환이 필요하기 때문에, 에너지 소비량의 63%를 차지했다. XNOR-SRAM은 전력 소비를 줄이기 위해 64개의 CIM 비트셀 열(column)이 하나의 아날로그-대-디지털 변환기를 공유했다. 그럼에도 불구하고, 처리 비트셀 어레이(processing bitcell array)는 도메인 변환이 끝날 때까지 동작을 정지시켜야 하기 때문에, 처리량이 저하되었다. 따라서, CIM 장치는 대부분의 에너지를 소비하거나 연산 중 처리량을 저하시키는 도메인 변환을 제거해야 한다. 또한, 이전 CIM 장치들은 누적된 비이상적인 아날로그 연산 변형들(cumulative non-ideal analog computing variations)로 인해 각 열(column)에서 비 선형 연산 결과가 발생하는 문제가 있었다. C3SRAM은 전하 기반 연산으로 아날로그 컴퓨팅이 특정 전압으로 수렴하는 선형성을 향상했다. 그럼에도 불구하고, 긴 MAC 연산 시간(~10ns) 때문에 시스템 처리량이 저하되었다.

[0005] 이하에서는 위에서 언급한 문제들을 해결하기 위해 TNN 연산 가속을 위한 TNN 연산 가속 장치를 제안한다. 결정 임계값(decision thresholding)을 갖는 전하 기반 컴퓨팅은 CIM 비트셀 어레이의 좋은 선형성과 높은 처리량을 달성할 수 있다. 또한, TNN 연산 가속 장치는 Psum 작업을 위한 아날로그-대-디지털 변환을 제거하여 높은 에너지 효율성을 달성할 수 있다.

발명의 내용

과제의 해결 수단

[0006] 일 실시예에 따른, 삼진 뉴럴 네트워크(Ternary Neural Network, TNN) 연산 가속 장치는, 제1 이진 값 및 제2 이진 값의 조합에 대응하는 삼진 가중치를 가지는 하나 이상의 삼진셀(ternary cell)을 포함하고, 상기 하나 이상의 삼진셀의 각각은, 로직 하이(logic high) 또는 로직 로우(logic low) 중 어느 하나를 지시하는 상기 제1 이진 값을 가지는 제1 비트 셀; 상기 제1 비트 셀의 인버터 페어(inverter pair)의 양단에 게이트를 통해 연결된 제1 트랜지스터 페어(transistor pair); 로직 하이 또는 로직 로우 중 어느 하나를 지시하는 상기 제2 이진 값을 가지는 제2 비트 셀; 상기 제2 비트 셀의 인버터 페어의 양단에 게이트를 통해 연결되고, 일단에서 상기 제1 트랜지스터 페어와 연결되며, 타단에서 미들 레벨 전압을 수신하는 제2 트랜지스터 페어; 상기 제2 트랜지스터 페어 및 출력 라인 사이에 연결되고, 상기 제1 트랜지스터 페어를 통해 수신된 입력 값과 삼진 가중치 간의 연산 결과를 상기 출력 라인에게 제공하는 연산 커패시터(computation capacitor)를 포함할 수 있다.

[0007] 상기 제1 비트 셀 및 상기 제2 비트 셀은, 정적 랜덤 액세스 메모리를 포함할 수 있다.

[0008] 상기 하나 이상의 삼진셀은, 상기 제1 이진 값이 로직 하이를 지시하고 상기 제2 이진 값이 로직 로우를 지시하는 것에 기초하여, 제1 삼진 논리 값에 해당하는 삼진 가중치를 저장할 수 있다.

[0009] 상기 하나 이상의 삼진셀은, 상기 제1 이진 값이 로직 하이를 지시하고 상기 제2 이진 값이 로직 하이를 지시하는 것에 기초하여, 제2 삼진 논리 값에 해당하는 삼진 가중치를 저장할 수 있다.

[0010] 상기 하나 이상의 삼진셀은, 상기 제2 이진 값이 로직 로우를 지시하는 것에 기초하여, 제3 삼진 논리 값에 해당하는 삼진 가중치를 저장할 수 있다.

[0011] 상기 하나 이상의 삼진셀의 각각은, 상기 제1 트랜지스터 페어의 트랜지스터들을 통해 수신된 입력 신호들의 전압 값들의 조합에 대응하는 삼진 입력 값 및 상기 삼진 가중치 간의 삼진 연산 결과에 기초하여, 상기 연산 커패시터를 통해 미리 결정된 전압 값에 대응하는 논리 값을 상기 출력 라인에 제공할 수 있다.

[0012] 상기 하나 이상의 삼진셀의 각각은, 상기 제2 트랜지스터 페어의 타단에서 수신된 하이 레벨 전압에 응답하여, 상기 연산 커패시터 및 상기 제2 트랜지스터 페어 간의 계산 노드(node)에서, 증가하는 전압이 미리 결정된 임계 전압 값에 도달하는 것에 기초하여, 상기 연산 결과를 출력 라인에게 제공할 수 있다.

[0013] 일 실시예에 따른, TNN 연산 가속 장치는, 상기 하나 이상의 삼진셀 각각이, 행들(rows) 및 열들(columns)로 배열되는 하나 이상의 삼진 비트셀 어레이(Ternary Bitcell Array, TBA); 및 복수의 출력 라인들을 포함하고, 상기 복수의 출력 라인들의 각각은, 상기 열들 중 해당하는 열에 배열된 삼진셀들과 연결될 수 있다.

[0014] 일 실시예에 따른, TNN 연산 가속 장치는, 서로 동일한 개수의 하나 이상의 TBA를 가지는, 하나 이상의 삼진 비트셀 뱅크(Ternary Bitcell Bank, TBB)를 포함할 수 있다.

[0015] 일 실시예에 따른, TNN 연산 가속 장치는, 상기 열들 중 해당하는 열에 배열된 삼진셀들과 출력 라인을 통해 일단에서 연결되고, 타단에서 미들 레벨 전압을 수신하는 스위칭 소자를 포함할 수 있다.

[0016] 일 실시예에 따른, TNN 연산 가속 장치는, 상기 열들 중 해당하는 열에 배열된 삼진셀들 각각을 통해 수신된 입력 값과 상기 삼진셀들 중 해당 삼진셀이 가지는 삼진 가중치 간의 연산 결과를 누적한 값에 기초하여, MAC(multiply and accumulate) 결과를 생성할 수 있다.

[0017] 일 실시예에 따른, TNN 연산 가속 장치는, 상기 생성된 MAC 결과를 수신하여, 상기 MAC 결과에 해당하는 전압 값에 기초하여 아날로그 영역(analog domain)에서 전하 기반 부분합(charge-based partial sum)을 수행하는 복수의 CPA(charge-based Psum Adder) 회로들을 더 포함하고, 상기 CPA 회로들의 각각은, MAC 결과에 해당하는 전압 값을 전하로 저장하는 복수의 전하 홀더들(charge holders); 및 상기 복수의 전하 홀더들에 저장된 전하들로부터 부분합 커패시터(partial sum capacitor)로의 순차적 전하 공유에 기초하여, 최종 MAC 결과를 생성하는 아날로그 메모리 회로(analog memory circuit)를 포함할 수 있다.

도면의 간단한 설명

[0018] 도 1은 일 실시예에 따른, 삼진 뉴럴 네트워크 연산 가속 장치의 전체 아키텍처를 도시한다.

도 2는 일 실시예에 따른 삼진셀의 구조를 나타낸다.

도 3a 내지 도 3c는 삼진셀이 제1 이진 값 및 제2 이진 값의 조합에 대응하는 삼진 가중치를 저장하는 것을 도시한다.

도 4a 내지 도 4c는 제1 트랜지스터 페어의 트랜지스터들을 통해 수신된 입력 신호들의 전압 값들의 조합으로 구분되는 삼진 입력 값을 도시한다.

도 5는 삼진셀에 저장된 삼진 가중치와 삼진 입력 값 간의 연산 결과에 기초하는 계산 노드의 전압 값을 나타낸다.

도 6은 일 실시예에 따른 삼진셀을 어레이 구조로 포함하는 삼진 비트셀 어레이를 도시한다.

도 7a 내지 도 7c는 TNN 연산 가속 장치가 TBA의 각 열 별로 MAC 결과를 출력하는 것을 도시한다.

도 8은 일 실시예에 따른 TNN 연산 가속 장치에 포함된 삼진셀에서 MAC 연산이 수행될 때, 결정 임계값을 적용한 결과를 비교 실시예와 그래프를 통해 비교한 도면이다.

도 9a 내지 도 9i는 CPA 회로들의 구조와 동작 방법을 설명하는 도면이다.

도 10a 내지 10h는 일 실시예에 따른 TNN 연산 가속 장치의 TNN 연산 수행을 위한 유연한 데이터 매핑 전략을 도시한다.

도 11a 내지 도 11f는 일 실시예에 따른 TNN 연산 가속 장치가 TBA 및 CPA 회로를 파이프라인 동작시키는 것을 도시한다.

도 12는 일 실시예에 따른 삼진셀의 레이아웃을 나타낸다.

도 13은 일 실시예에 따른 TNN 연산 가속 장치의 칩 레이아웃을 나타낸다.

도 14는 미리 결정된 임계 전압 값과 삼진셀에 의한 MAC 연산의 정확도 간의 상관 관계를 그래프로 나타낸다.

도 15는 CPA에 의한 부분 합 값에 따른 최종 V_{PSUM} 값의 선형성을 나타내는 그래프 및 ADC를 사용하는 비교 실시예와 비교하여 TNN 연산 가속 장치의 에너지 효율성 향상을 나타내는 그래프를 도시한다.

도 16은 CPA 회로에 의해 수행되는 부분합 연산을 통해 생성된 최종 MAC 결과에 대한 그래프다.

도 17은 일 실시예에 따른 TNN 연산 가속 장치가 동작함에 있어서 유연한 데이터 매핑 전략을 활용한 경우의 메

모리 용량을 비교 실시예의 메모리 용량과 대비하는 그래프다.

발명을 실시하기 위한 구체적인 내용

[0019] 실시예들에 대한 특정한 구조적 또는 기능적 설명들은 단지 예시를 위한 목적으로 개시된 것으로서, 다양한 형태로 변경되어 구현될 수 있다. 따라서, 실제 구현되는 형태는 개시된 특정 실시예로만 한정되는 것이 아니며, 본 명세서의 범위는 실시예들로 설명한 기술적 사상에 포함되는 변경, 균등물, 또는 대체물을 포함한다.

[0020] 제1 또는 제2 등의 용어를 다양한 구성요소들을 설명하는데 사용될 수 있지만, 이런 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 해석되어야 한다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.

[0021] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다.

[0022] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 설명된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함으로 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0023] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 해당 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0024] 이하, 실시예들을 첨부된 도면들을 참조하여 상세하게 설명한다. 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조 부호를 부여하고, 이에 대한 중복되는 설명은 생략하기로 한다.

[0026] 도 1은 일 실시예에 따른, 삼진 뉴럴 네트워크(Ternary Neural Network, TNN) 연산 가속 장치의 전체 아키텍처를 도시한다.

[0028] 일 실시예에 따른 TNN 연산 가속 장치(100)는, 삼진 가중치를 저장할 수 있다. TNN 연산 가속 장치(100)는 저장된 삼진 가중치와 삼진 입력 신호 간의 연산 결과에 기초하여, 삼진 가중치를 업데이트할 수 있다. 예를 들어, TNN 연산 가속 장치(100)는 CPU 또는 FPGA로부터 TNN에 포함된 복수의 레이어 중 제N 레이어 내의 복수의 노드에 대응하는 가중치를 로딩(load)함으로써, 삼진 가중치를 업데이트할 수 있다.

[0029] TNN 연산 가속 장치(100)는 삼진 입력 신호에 대하여 TNN 연산을 수행함으로써, TNN 연산을 가속할 수 있다. 예를 들어, TNN 연산 가속 장치(100)의 동작은 TNN에 포함된 복수의 레이어 중 적어도 하나의 레이어의 입력 액티베이션에 대한 가중합 연산 동작을 포함할 수 있다. TNN 연산 가속 장치(100)에 의해 수행되는 입력 신호에 대한 TNN 연산 동작 및 데이터 맵핑에 대하여 이하에서 구체적으로 설명한다.

[0030] 일 실시예에 따른 TNN 연산 가속 장치(100)는 복수개의 삼진 비트셀 뱅크(120)(Ternary Bitcell Bank, TBB)(이하, 'TBB'), 읽기/쓰기 인터페이스(Read/Wright Interface), 입력 신호를 생성하여 삼진 비트셀 뱅크로 송신하는 탑 액티베이션 드라이버(top activation driver), 및 복수개의 전하 기반 부분 합 덧셈기들(140)(charge-based partial sum adders, CPAs)(이하, 'CPA들')을 포함할 수 있다. 예를 들어, TNN 연산 가속 장치(100)는 3개의 TBB(120), 1개의 읽기/쓰기 인터페이스, 1개의 탑 액티베이션 드라이버, 및 4개의 CPA들(140)을 포함할 수 있다.

[0031] TBB(120)는 복수의 삼진 비트셀 어레이들(110)(Ternary Bitcell Arrays, TBAs)(이하, 'TBA들'), 복수의 액티베이션 SW 네트워크, 및 뱅크 컨트롤러(bank controller)를 포함할 수 있다. 예를 들어, TBB(120)는 4개의 TBA들(110), 4개의 액티베이션 SW 네트워크, 및 1개의 뱅크 컨트롤러를 포함할 수 있다. 예를 들어 TBB(120) 각각은 모두 4개의 TBA들(110)을 포함할 수 있고, 하나의 TBB(120)에 포함된 서로 다른 4개의 TBA들(110) 각각은 다른 TBB(120)에 포함된 서로 다른 4개의 TBA들(110) 각각과 병렬적으로 연결될 수 있다. 예를 들어, 제1 TBB에

포함된 TBA #0과 제2 TBB에 포함된 TBA #0은 병렬적으로 연결될 수 있다.

[0032] 참고로, TBA들(110) 각각은 해당 액티베이션 SW 네트워크를 통해, 탑 액티베이션 드라이버로부터 수신된 입력 신호들을 수신할 수 있다. TBA들(110)은 각각 16T1C의 삼진셀(미도시)을 포함할 수 있다. TBA들(110) 각각에 포함된 삼진셀에 대하여 이하 도 2에서 구체적으로 설명한다.

[0033] TBB(120)에 포함된 복수의 TBA들(110)은 파이프라인 방식으로 동작할 수 있다. 예를 들어, 3개의 TBA들(110)(예: TBA #0, TBA #1, TBA #2)이 [-96, +96]의 범위에서 32개의 MAC(Multiply and Accumulate) 연산들을 수행하는 동안, 나머지 TBA(예: TBA #3)는 TBA에 저장된 가중치를 업데이트 할 수 있다. 또한, TBB(120)는 각각 병렬적으로 연결될 수 있다. 예를 들어, TBB(120)는 [-288, +288] 범위에서 MAC 연산 결과(예: V_{MAC})를 계산(예: 합산)하기 위해 연결될 수 있다. TBA들(110)에서 수행되는 MAC 연산 및 가중치의 업데이트에 대하여 이하 도 7a 내지 7c에서 구체적으로 설명한다.

[0034] CPA들(140)은 TBB(120)로부터 수신한 MAC 연산 결과(130)(예: 전압)에 기초하여, 아날로그 영역에서 전하 기반 부분 합 연산을 수행할 수 있다. 예를 들어, CPA들(140)은 제1 TBB에 포함된 TBA #0으로부터 생성된 제1 연산 결과, 제2 TBB에 포함된 TBA #0으로부터 생성된 제2 연산 결과 및 제3 TBB에 포함된 TBA #0으로부터 생성된 제3 연산 결과를 합한 MAC 연산 결과(130)를 수신하여, 커패시터 및 스위치를 이용한 전하 공유를 수행함으로써 최종 MAC 결과(예: V_{PSUM})를 출력할 수 있다. CPA들(140)에 의해 최종 MAC 결과를 출력하는 방법에 대하여 이하 도 9a 내지 9i에서 구체적으로 설명한다.

[0035] 다시 말해, 일 실시예에 따른 TNN 연산 가속 장치는, TNN 연산을 위한 에너지 효율적인 아키텍처, 빠른 처리 속도 및 좋은 선형성을 위한 컴팩트한 16T1C 구조의 삼진 비트셀 어레이(ternary bitcell array) 아키텍처, 아날로그 영역에서 연산을 수행하는 전하 기반 부분 합(charge based partial sum) 회로, 및 제한된 크기의 가중치 메모리로 전체 컨볼루션 레이어에 대한 유연한 데이터 매핑을 포함할 수 있다.

[0037] 도 2는 일 실시예에 따른 삼진셀(Ternary cell)의 구조를 나타낸다.

[0039] 일 실시예에 따른 TNN 연산 가속 장치는 제1 이진 값 및 제2 이진 값의 조합에 대응하는 삼진 가중치를 가지는 하나 이상의 삼진셀(ternary cell)(200)을 포함할 수 있다. 제1 이진 값 및 제2 이진 값의 조합에 대응하는 삼진 가중치는 예를 들어, [0, 1, 2]의 값에 대응할 수 있다. 다른 예를 들어 삼진 가중치는 [-1, 0 1]의 값에 대응할 수 있다. 이하에서는 삼진 가중치가 [-1, 0 1]에 대응하는 경우를 중점적으로 설명하나, 삼진 가중치 값은 이에 제한되는 것은 아니다.

[0040] 하나 이상의 삼진셀(200)의 각각은, 로직 하이(logic high) 또는 로직 로우(logic low) 중 어느 하나를 지시하는 제1 이진 값을 가지는 제1 비트 셀(210)을 포함할 수 있다. 예를 들어, 제1 비트 셀(210)은 정적 랜덤 액세스 메모리(Static Random Access Memory, SRAM)를 포함할 수 있다. 예를 들어, 제1 비트 셀(210)이 SRAM에 해당하는 경우, 제1 이진 값은, SRAM에 포함된 한 쌍의 인버터 양단(211, 212)에 저장되는 값에 대응할 수 있다. 예를 들어, 로직 하이는 제1 단(211)(예: QM 노드) 및 제2 단(212)(예: QMB 노드) 각각에 (1, 0)의 값이 저장되는 경우를 지시할 수 있다. 동시에 로직 로우는 제1 단(211) 및 제2 단(212) 각각에 (0, 1)의 값이 저장되는 경우를 지시할 수 있다. 다른 예를 들어, 로직 하이는 제1 단(211) 및 제2 단 각각에 (0, 1)의 값이 저장되는 경우를 지시할 수 있고, 동시에 로직 로우는 제1 단(211) 및 제2 단(212) 각각에 (1, 0)의 값이 저장되는 경우를 지시할 수 있다.

[0041] 삼진셀(200)은 제1 비트 셀(210)의 인버터 페어(inverter pair)의 양단(211, 212)에 게이트를 통해 연결된 제1 트랜지스터 페어(transistor pair)(230)를 포함할 수 있다. 예를 들어, 제1 트랜지스터 페어(230)는 제1 트랜지스터(예: T0) 및 제2 트랜지스터(예: T1)를 포함할 수 있다. 제1 트랜지스터 페어(230)는 제1 트랜지스터의 일단으로 제1 입력 신호(216)(예: CWLB)를 수신할 수 있다. 제1트랜지스터의 타단은 제2 트랜지스터의 일단과 연결될 수 있다. 제1 트랜지스터 페어(230)는 제2 트랜지스터의 타단으로 제2 입력 신호(215)(예: CWL)을 수신할 수 있다. 예를 들어, 제1 트랜지스터 및 제2 트랜지스터는 각각 저-문턱전압 NMOS(low-VT NMOS)를 포함할 수 있다.

[0042] 삼진셀(200)은 로직 하이 또는 로직 로우 중 어느 하나를 지시하는 제2 이진 값을 가지는 제2 비트 셀(220)을 포함할 수 있다. 예를 들어, 제2 비트 셀(220)은 SRAM을 포함할 수 있다. 예를 들어, 제2 비트 셀(220)이

SRAM에 해당하는 경우, 로직 하이 및 로직 로우를 지시하는 제2 이진 값에 대한 설명은 위에서 설명한 제1 비트 셀(210)의 로직 하이 및 로직 로우를 지시하는 제1 이진 값에 대한 설명과 중복되므로 생략한다.

[0043] 삼진셀(200)은 제2 비트 셀(220)의 인버터 페어의 양단(221, 222)에 게이트를 통해 연결되고, 일단에서 제1 트랜지스터 페어(230)와 연결되며, 타단에서 미들 레벨 전압을 수신하는 제2 트랜지스터 페어(240)를 포함할 수 있다. 예를 들어, 제2 트랜지스터 페어(230)는 제3 트랜지스터(예: T2) 및 제4 트랜지스터(예: T3)를 포함할 수 있다. 제2 트랜지스터 페어(240)의 제3 트랜지스터의 일단은 제1 트랜지스터 페어(230)의 제1 트랜지스터 및 제2 트랜지스터가 연결됨으로써 형성된 공통 노드와 연결될 수 있다. 제2 트랜지스터 페어(240)의 제3 트랜지스터의 타단은 제4 트랜지스터의 일단과 연결될 수 있다. 제2 트랜지스터 페어(240)는 제3 트랜지스터의 타단과 제4 트랜지스터의 일단 간의 연결에 기초하여 컴퓨테이션 노드(computation node)(예: CN)를 형성할 수 있다. 제2 트랜지스터 페어(240)의 제4 트랜지스터의 타단은 미들 레벨 전압을 수신할 수 있다. 예를 들어, 미들 레벨 전압은 325mV에 대응하는 전압 값을 지시할 수 있다. 참고로, 하이 레벨 전압(V_H), 미들 레벨 전압(V_M), 및 로우 레벨 전압(V_L) 각각은, 예를 들어, 620mV, 325mV, 및 0V에 대응하는 전압 값을 지시할 수 있다.

[0044] 삼진셀(200)은 제2 트랜지스터 페어(240) 및 출력 라인(260) 사이에 연결되고, 제1 트랜지스터 페어(230)를 통해 수신된 입력 값(215, 216)과 삼진 가중치 간의 연산 결과를 출력 라인(260)에게 제공하는 연산 커패시터(computation capacitor)(250)를 포함할 수 있다. 예를 들어, 연산 커패시터(250)는 일단에서 제2 트랜지스터 페어(240)의 제3 트랜지스터 및 제4 트랜지스터에 의해 형성된 컴퓨테이션 노드와 일단에서 연결되고, 타단에서 출력 라인과 연결될 수 있다. 예를 들어, 연산 커패시터(250)는 제1 입력 신호(216) 및 제2 입력 신호(215)에 기초하는 삼진 입력 신호와 삼진 가중치 간의 곱셈 결과에 대응하는 전하를 저장하고, 저장된 전하를 출력 라인(260)에 제공할 수 있다. 삼진셀(200)의 구체적인 동작 방법에 관하여, 이하 도 3a 내지 도 5를 참조한다.

[0046] 도 3a 내지 도 3c는 삼진셀이 제1 이진 값 및 제2 이진 값의 조합에 대응하는 삼진 가중치를 저장하는 것을 도시한다.

[0047] 아래의 표 1을 참조하여, 도 3a 내지 도 3c 각각 마다 삼진셀이 저장하는 삼진 가중치에 대하여 설명한다.

표 1

[0049]

Weight	+1	-1	0
QL/QLB	V_{DD}/GND	V_{DD}/GND	GND/V_{DD}
QM/QMB	GND/V_{DD}	V_{DD}/GND	Don't care

[0050] 도 3a는 삼진셀(300a)이 제1 삼진 논리 값에 해당하는 삼진 가중치를 저장하는 것을 나타낸다. TNN 연산 가속 장치에 포함된, 하나 이상의 삼진셀(300a)은, 제1 이진 값이 로직 하이를 지시하고 제2 이진 값이 로직 로우를 지시하는 것에 기초하여, 제1 삼진 논리 값에 해당하는 삼진 가중치를 저장할 수 있다. 예를 들어, 도 3a 및 위의 표 1을 참조하면, 삼진셀(300a)은 QM 노드(310a)를 접지(ground)시키고, QMB 노드(320a)에 공급 전원(예: V_{DD})을 인가함으로써 로직 하이를 지시하는 제1 이진 값을 설정할 수 있다. 또한, 삼진셀(300a)은 QL 노드(330a)에 공급 전원을 인가하고, QLB 노드(340a)를 접지시킴으로써, 로직 로우를 지시하는 제2 이진 값을 설정할 수 있다. 삼진셀(300a)은 로직 하이를 지시하는 제1 이진 값 및 로직 로우를 지시하는 제2 이진 값에 기초하여, 제1 트랜지스터 페어의 제2 트랜지스터(예: T1) 및 제2 트랜지스터 페어의 제3 트랜지스터(예: T2)를 동작할 수 있다. 삼진셀(300a)은 로직 하이를 지시하는 제1 이진 값 및 로직 로우를 지시하는 제2 이진 값에 기초하여, 제1 트랜지스터 페어의 제2 트랜지스터(예: T1) 일단을 통해 수신되는 제2 입력 신호(예: CWL)를 제2 트랜지스터 페어의 제3 트랜지스터(예: T2)를 거쳐 연산 커패시터가 연결된 계산 노드(computation node, CN)로 전달할 수 있다. 따라서, 삼진셀(300a)에 의해 제2 입력 신호(예: CWL)가 계산 노드(예: CN)로 전달될 수 있는 경우(350a)를 삼진셀(300a)이 제1 삼진 논리 값에 해당하는 삼진 가중치를 저장하고 있는 것으로 볼 수 있다. 예를 들어, 제1 삼진 논리 값은 '+1'을 지시하는 가중치 값을 포함할 수 있다.

[0051] 다만, 이상에서는 QM 노드(310a) 및 QL 노드(330a)를 기준으로 로드 하이 및 로드 로우를 설명하였으나, 로직

레벨이 반대로 설계되는 경우 로드 하이 및 로드 로우가 반전되어 표현될 수 있으며, 이하의 설명에서는 QM 노드(310a) 및 QL 노드(330a)를 기준으로 로드 하이, 로드 로우를 설명한다.

[0053] 도 3b는 삼진셀(300b)이 제2 삼진 논리 값에 해당하는 삼진 가중치를 저장하는 것을 나타낸다. TNN 연산 가속 장치에 포함된, 하나 이상의 삼진셀(300b)은 제1 이진 값이 로직 하이를 지시하고 제2 이진 값이 로직 하이를 지시하는 것에 기초하여, 제2 삼진 논리 값에 해당하는 삼진 가중치를 저장할 수 있다. 예를 들어, 도 3b 및 위의 표 1을 참조하면, 삼진셀(300b)은 QM 노드(310b)에 공급 전원을 인가하고, QLB 노드(320b)를 접지시킴으로써, 로직 하이를 지시하는 제1 이진 값을 설정할 수 있다. 동시에 삼진셀(300b)은 QL 노드(330b)에 공급 전원을 인가하고, QLB 노드(340b)를 접지시킴으로써, 로직 하이를 지시하는 제2 이진 값을 설정할 수 있다. 삼진셀(300b)은 로직 하이를 지시하는 제1 이진 값 및 로직 하이를 지시하는 제2 이진 값에 기초하여, 제1 트랜지스터 페어의 제1 트랜지스터(예: T0) 및 제2 트랜지스터 페어의 제3 트랜지스터(예: T2)를 동작할 수 있다. 삼진셀(300b)은 로직 하이를 지시하는 제1 이진 값 및 로직 하이를 지시하는 제2 이진 값에 기초하여, 제1 트랜지스터 페어의 제1 트랜지스터(예: T0) 일단을 통해 수신되는 제1 입력 신호(예: CWLB)를 제2 트랜지스터 페어의 제3 트랜지스터(예: T2)를 거쳐 연산 커패시터가 연결된 계산 노드(CN)로 전달할 수 있다. 따라서, 삼진셀(300b)에 의해 제1 입력 신호가 계산 노드로 전달될 수 있는 경우(350b)를 삼진셀(300b)이 제2 삼진 논리 값에 해당하는 삼진 가중치를 저장하고 있는 것으로 볼 수 있다. 예를 들어, 제2 삼진 논리 값은 '-1'을 지시하는 가중치 값을 포함할 수 있다.

[0055] 도 3c는 삼진셀(300c)이 제3 삼진 논리 값에 해당하는 삼진 가중치를 저장하는 것을 나타낸다. TNN 연산 가속 장치에 포함된, 하나 이상의 삼진셀(300c)은, 제2 이진 값이 로직 로우를 지시하는 것에 기초하여, 제3 삼진 논리 값에 해당하는 삼진 가중치를 저장할 수 있다. 예를 들어, 도 3c 및 위의 표 1을 참조하면, 삼진셀(300c)은 QL 노드(310c)를 접지시키고, QLB 노드(320c)에 공급 전원을 인가함으로써, 로직 로우를 지시하는 제2 이진 값을 설정할 수 있다. 삼진셀(300c)은 로직 로우를 지시하는 제2 이진 값에 기초하여, 제2 트랜지스터 페어의 제4 트랜지스터(예: T3)를 동작할 수 있다. 삼진셀(300c)은 로직 로우를 지시하는 제2 이진 값에 기초하여, 제2 트랜지스터 페어의 제4 트랜지스터(예: T3) 일단을 통해 수신되는 미들 레벨 전압 값에 대응하는 신호(예: V_M)를 연산 커패시터가 연결된 계산 노드로 전달할 수 있다. 따라서, 삼진셀(300c)에 의해 미들 레벨 전압 값에 대응하는 신호가 계산 노드로 전달될 수 있는 경우(330c)를 삼진셀(300c)이 제3 삼진 논리 값에 해당하는 삼진 가중치를 저장하고 있는 것으로 볼 수 있다. 예를 들어, 제3 삼진 논리 값은 '0'을 지시하는 가중치 값을 포함할 수 있다.

[0057] 도 4a 내지 도 4c는 제1 트랜지스터 페어의 트랜지스터들을 통해 수신된 입력 신호들의 전압 값들의 조합으로 구분되는 삼진 입력 값을 도시한다.

[0058] 아래의 표 2를 참조하여, 도 4a 내지 도 4c 각각 마다 입력 신호들의 전압 값들의 조합으로 구분되는 삼진 입력 값에 대하여 설명한다.

표 2

[0059]

Activation	+1	-1	0
CWL	V_H	V_L	V_M
CWLB	V_L	V_H	V_M

[0060] 표 2에서 activation +1, -1, 0은 삼진셀에 수신된 입력 신호들의 전압 값들의 조합으로 구분되는 삼진 입력 값들을 나타낸다. 표 2에서 CWL, CWLB 각각은 제1 트랜지스터 페어의 제2 트랜지스터(예: T1)의 일단 및 제1트랜지스터(예: T0)의 일단을 통해 인가되는 입력 신호들을 나타낼 수 있다.

[0061] 도 4a는 제1 트랜지스터 페어의 트랜지스터들을 통해 수신된 입력 신호들의 전압 값들의 조합으로 제1 삼진 입력 값을 구현한 도면이다. 예를 들어, 삼진셀(400a)은 제1 트랜지스터 페어의 제1 트랜지스터(예: T0)의 일단에 인가되는 제1 입력 신호(420a)의 전압 값이 로우 레벨 전압(예: V_L)에 해당하고, 제1 트랜지스터 페어의 제2

트랜지스터(예: T1)의 일단에 인가되는 제2 입력 신호(410a)의 전압 값이 하이 레벨 전압(예: V_H)에 해당하는 경우(430a)를 제1 삼진 입력 값으로 구분할 수 있다. 예를 들어, 제1 삼진 입력 값은 '+1'의 값을 지시하는 입력 액티베이션 값으로 구분될 수 있다.

[0062] 도 4b는 제1 트랜지스터 페어의 트랜지스터들을 통해 수신된 입력 신호들의 전압 값들의 조합으로 제2 삼진 입력 값을 구현한 도면이다. 예를 들어, 삼진셀(400b)은 제1 트랜지스터 페어의 제1 트랜지스터(예: T0)의 일단에 인가되는 제1 입력 신호(420b)의 전압 값이 하이 레벨 전압(예: V_H)에 해당하고, 제1 트랜지스터 페어의 제2 트랜지스터(예: T1)의 일단에 인가되는 제2 입력 신호(410b)의 전압 값이 로우 레벨 전압(예: V_L)에 해당하는 경우(430b)를 제2 삼진 입력 값으로 구분할 수 있다. 예를 들어 제2 삼진 입력 값은 '-1'의 값을 지시하는 입력 액티베이션 값으로 구분될 수 있다.

[0063] 도 4c는 제1 트랜지스터 페어의 트랜지스터들을 통해 수신된 입력 신호들의 전압 값들의 조합으로 제3 삼진 입력 값을 구현한 도면이다. 예를 들어, 삼진셀(400c)은 제1 트랜지스터 페어의 제1 트랜지스터(예: T0)의 일단에 인가되는 제1 입력 신호(420c)의 전압 값이 미들 레벨 전압(예: V_M)에 해당하고, 제1 트랜지스터 페어의 제2 트랜지스터(예: T1)의 일단에 인가되는 제2 입력 신호(410c)의 전압 값이 미들 레벨 전압(예: V_M)에 해당하는 경우(430c)를 제3 삼진 입력 값으로 구분할 수 있다. 예를 들어 제3 삼진 입력 값은 '0'의 값을 지시하는 입력 액티베이션 값으로 구분될 수 있다.

[0065] 도 5는 삼진셀에 저장된 삼진 가중치와 삼진 입력 값 간의 연산 결과에 기초하는 계산 노드의 전압 값을 나타낸다.

[0066] 일 실시예에 따른 하나 이상의 삼진셀(500)의 각각은, 제1 트랜지스터 페어의 트랜지스터들을 통해 수신된 입력 신호들의 전압 값들의 조합에 대응하는 삼진 입력 값 및 상기 삼진 가중치 간의 삼진 연산 결과에 기초하여, 연산 커패시터를 통해 미리 결정된 전압 값에 대응하는 논리 값을 출력 라인에 제공할 수 있다.

[0067] 도 3a 내지 도 3c에서 설명한 바와 같이, 삼진셀(300a, 300b, 300c)은 각각 제1 삼진 논리 값(예: +1), 제2 삼진 논리 값(예: -1), 제3 삼진 논리 값(예: 0)에 해당하는 삼진 가중치를 저장할 수 있다. 또한, 도 4a 내지 도 4c에서 설명한 바와 같이 삼진셀(400a, 400b, 400c)은 각각 제1 삼진 입력 값(예: +1), 제2 삼진 입력 값(예: -1), 제3 삼진 입력 값(예: 0)을 구분할 수 있다. 삼진셀(500)은 서로 다른 3개의 삼진 가중치와 서로 다른 3개의 삼진 입력 값에 기초하여, MULT 연산 결과를 생성할 수 있다. 아래의 표 3을 참조하면, 도 5에서 삼진셀(500)은 삼진 가중치와 삼진 입력 값 간의 연산에 기초하는 삼진 MULT 연산 결과에 따라 계산 노드의 전압 값을 하이 레벨 전압(V_H), 로우 레벨 전압(V_L), 및 미들 레벨 전압(V_M) 중 하나로 나타낼 수 있다.

표 3

[0068]

Activation × Weight	+1	-1	0
Computation node	V_H	V_L	V_M

[0069] 회로(510)는 삼진셀(500)에서의 삼진 가중치와 삼진 입력 값 간의 연산에 기초하는 삼진 MULT 연산 결과가 '+1'인 경우를 나타낸다. 예를 들어, 삼진셀(500)에 저장된 삼진 가중치가 '+1'이고, 삼진 입력 값이 '+1'인 때, 삼진 가중치와 삼진 입력 값을 곱한 삼진 MULT 연산 결과가 '+1'일 수 있다. 다른 예를 들어, 삼진셀(500)에 저장된 삼진 가중치가 '-1'이고, 삼진 입력 값이 '-1'인 때, 삼진 MULT 연산 결과는 '+1'일 수 있다. 다시 말해, 삼진셀(500)의 (삼진 가중치, 삼진 입력 값)이 각각 (1, 1) 또는 (-1, -1)인 경우 MULT 연산 결과는 +1일 수 있다. MULT 연산 결과가 +1인 경우, 삼진셀(500)의 계산 노드 전압은 회로(510)에서 하이 레벨 전압 값에 해당할 수 있다. 회로(520)는 삼진셀(500)에서의 삼진 MULT 연산 결과가 '-1'인 경우를 나타낸다. 예를 들어, 삼진셀(500)에 저장된 삼진 가중치가 '+1'이고, 삼진 입력 값이 '-1'인 때, 삼진 가중치와 삼진 입력 값을 곱한 삼진 MULT 연산 결과는 '-1'일 수 있다. 다른 예를 들어, 삼진셀(500)에 저장된 삼진 가중치가 '-1'이고, 삼진 입력 값이 '+1'인 때, 삼진 MULT 연산 결과는 '-1'일 수 있다. 다시 말해, 삼진셀(500)의 (삼진 가중치, 삼진 입력 값)이 각각 (1, -1) 또는 (-1, 1)인 경우 MULT 연산 결과는 +1일 수 있다. MULT 연산 결과가 -1인 경우, 삼진셀(500)의 계산 노드 전압은 회로(520)에서 로우 레벨 전압 값에 해당할 수 있다.

[0070] 회로(530)는 삼진셀(500)에서 삼진 MULT 연산 결과가 0인 경우를 나타낸다. 예를 들어, 삼진셀(500)의 (삼진 가중치, 삼진 입력 값)이 각각 (1, 0), (-1, 0), (0, 1), (0, -1) 또는 (0, 0)인 경우 MULT 연산 결과는 0일 수 있다. MULT 연산 결과가 0인 경우, 삼진셀(500)의 계산 노드 전압은 회로(530)에서 미들 레벨 전압 값에 해당할 수 있다.

[0071] 삼진셀(500)은 회로(510, 520, 530)의 계산 노드 전압 값에 따라, 대응하는 논리 값을 출력 라인에 제공할 수 있다.

[0073] 도 6은 일 실시예에 따른 삼진셀을 어레이 구조로 포함하는 삼진 비트셀 어레이를 도시한다.

[0074] 일 실시예에 따른 TNN 연산 가속 장치는, 하나 이상의 삼진셀(601) 각각이, 행들(rows) 및 열들(columns)로 배열되는 하나 이상의 삼진 비트셀 어레이(Ternary Bitcell Array, TBA)(이하 'TBA')(600), 및 복수의 출력 라인들을 포함할 수 있다. 예를 들어, TBA(600)는 96×32의 삼진셀(601) 어레이, 비트라인/워드라인 드라이버, MAC 프리 차지 회로(MAC pre charge circuit) 및 어레이 컨트롤러(array ctrl)를 포함할 수 있다. 복수의 출력 라인들의 각각은 TBA(600)의 열들 중 해당하는 열에 배열된 삼진셀들(601)과 연결될 수 있다. 복수의 출력 라인들의 각각은 연결된 삼진셀들(601)의 삼진 가중치와 삼진 입력 신호 간의 연산 결과에 대응하는 출력 신호를 TBA(600)의 열 별로 누적할 수 있다. 예를 들어, 96×32의 삼진셀(601) 어레이를 포함하는 TBA(600)는 각 열 별로 병렬적으로 처리함으로써, 한 사이클에 32개의 MAC 연산들(예: |±96| MAC 연산 결과 값)을 누적한 MAC 결과(620)를 출력할 수 있다.

[0076] 도 7a 내지 도 7c는 TNN 연산 가속 장치가 TBA의 각 열(column)별로 MAC 결과를 출력하는 것을 도시한다.

[0077] 도 7a는 TBB(700a)의 열 별 MAC 연산의 타이밍도를 나타낸다. TBB(700a) 내에서 동일한 위치에 있는 TBA(701a)는 |±288| MAC 연산 결과를 획득하기 위해 연결될 수 있다. 예를 들어, TBB #0에 포함된 제1 TBA #0, TBB #1에 포함된 제2 TBA #0, 및 TBB #2에 포함된 제3 TBA #0은 병렬적으로 연결될 수 있다. TBB(700a)는 행 별로 연결된 TBA들(예: 서로 다른 TBB에 포함된 #0으로 표현되는 TBA들) 각각으로부터 MAC 연산 결과를 누적함으로써, 누적된 MAC 연산 결과(760a)(예: V_{MAC})를 CPA로 전달할 수 있다. TBB(700a)의 열 별 MAC 연산에는 프리 차지 단계(pre-charge stage) 및 연산 단계(computation stage)의 두 단계를 포함할 수 있다. 프리 차지 단계 및 연산 단계에 대하여 이하, 도 7b 및 도 7c에서 구체적으로 설명한다.

[0079] 도 7b는 TBB(700b)의 열 별 MAC 연산에서 프리 차지 단계를 도시한다.

[0080] 일 실시예에 따른 TNN 연산 가속 장치는 TBA(701b)의 열들 중 해당하는 열에 배열된 삼진셀들과 출력 라인을 통해 일단에서 연결되고, 타단에서 미들 레벨 전압을 수신하는 스위칭 소자를 포함할 수 있다. 스위칭 소자는 예를 들어 프리 차지 스위치(710b)를 포함할 수 있다. 이하에서는 프리 차지 스위치(710b)를 중점적으로 설명한다. 일 실시예에 따른 TNN 연산 가속 장치에 포함된 삼진셀은 TBA(701b)의 각 열 별로 공통 출력 라인과 연결될 수 있다. TBA(701b)의 각 열 별 출력 라인과 공통으로 연결된 삼진셀은 프리 차지 단계에서 삼진 가중치와 삼진 입력 값의 연산 결과인 MULT 값을 0으로 설정할 수 있다. 다시 말해, 0의 MULT 값은 삼진셀이 연산 커패시터와 연결된 계산 노드의 전압을 미들 레벨 전압 값으로 설정한 것을 나타낼 수 있다.

[0081] TBA(701b)의 각 열 별 출력 라인은 MAC 프리 차지 스위치(710b)와 연결될 수 있다. MAC 프리 차지 스위치(710b)의 일단은 미들 레벨 전압 값(예: V_M)을 수신할 수 있다. MAC 프리 차지 스위치(710b)가 ON되면 미들 레벨 전압 값이 출력 라인에 MAC 프리 차지 스위치(710b)를 통해 전달될 수 있다. 프리 차지 단계에서, 미들 레벨 전압 값으로 설정된 삼진셀 및 MAC 프리 차지 스위치(710b)가 ON 되어 미들 레벨 전압 값이 전달된 출력 라인 사이에 연결된 연산 커패시터들에 전위차가 발생하지 않을 수 있다. 예를 들어, TBA(701b) 내의 해당 열에 배열된 각 삼진셀들에 포함된 계산 노드와 연산 커패시터를 통해 연결된 출력 라인에 동일한 전압 값(예: 미들 레벨 전압 값)이 인가됨으로써, 연산 커패시터에 저장되는 전하량은 방정식 Q=CV에 기초하여 0의 값을 가질 수 있다. 달리 말해, 프리 차지 단계에서 TBA(701b) 내 삼진셀들의 연산 커패시터에 저장된 전체 전하량은 0의 값을 가질 수 있다. 프리 차지 단계에서 출력 라인을 통해 출력 되는 MAC 연산 결과(720b)(예: V_{MAC})는 미들 레벨 전압 값(V_M)에 대응할 수 있다.

[0083] 도 7c는 TBB(700c)의 열 별 MAC 연산에서 연산 단계를 도시한다.

[0084] 일 실시예에 따른 TBB 연산 가속 장치는, TBA(701c)의 열들 중 해당하는 열에 배열된 삼진셀들 각각을 통해 수신된 입력 값과 삼진셀들 중 해당 삼진셀이 가지는 삼진 가중치 간의 연산 결과를 누적한 값에 기초하여, MAC(multiply and accumulate) 결과를 생성할 수 있다. 연산 단계에서 삼진 입력 값이 TBA(701c)의 삼진셀에 인가될 수 있다. 삼진셀은 삼진 입력 값과 삼진 가중치에 기초하여 MULT 연산 결과를 생성할 수 있다. 예를 들어, TBA(701c)의 한 열에 배열된 삼진셀들 각각의 MULT 연산 결과는 [1, -1, -1, 0, 1, 0, 쪈 -1]에 해당할 수 있다. 참고로, 각 삼진 셀들의 계산 노드 전압은 도 5에서 설명한 바와 같이 MULT 연산 결과에 대응하는 전압 값일 수 있다. 동시에, MAC 프리 차지 스위치(도 7b의 710b)는 OFF될 수 있다.

[0085] TBB(700c)의 TBA(701c)는 출력 라인을 통해 병렬적으로 연결될 수 있다. 예를 들어, TBA #0의 해당 열에 배열된 삼진셀들은 출력 라인을 통해 연결될 수 있고, TBB #0의 TBA #0, TBB#1의 TBA #0, 및 TBB#2의 TBA #0 각각의 출력 라인(710c)이 병렬적으로 연결되어 각 출력 라인(710c)을 통한 MAC 결과(720c)를 누적할 수 있다.

[0086] 출력 라인(710c)에는 커패시티브 디바이딩(capacitive dividing)을 통해 MAC 결과(720c)가 출력될 수 있다.

[0087] 예를 들어, TBB(700c) 내의 TBA(701c)의 해당 열에 배열된 삼진셀이 4개 인 경우, 4개의 삼진셀 각각에 포함된 계산 노드의 전압 값은 입력 액티베이션 및 삼진셀 각각에 포함된 삼진 가중치 간의 연산 결과에 대응할 수 있다. 예를 들어, 4개의 삼진셀 각각에 포함된 계산 노드의 전압 값은 V_H, V_L, 또는 V_M 중 적어도 하나를 포함할 수 있다. 예를 들어, 4개의 삼진셀 중 제1 삼진셀의 계산 노드 전압 값이 V_H, 제2 삼진셀의 계산 노드 전압 값이 V_L, 제3 삼진셀의 계산 노드 전앖 값이 V_M, 제4 삼진셀의 계산 노드 전압 값이 V_H일 수 있다. 제1 삼진셀 내지 제4 삼진셀 각각의 연산 커패시터에 저장된 전하량의 합은 아래 수학식 1 내지 수학식 2에 의해 계산될 수 있다.

수학식 1

[0088] $$TotalQ=Q_1+Q_2+Q_3+Q_4$$

수학식 2

[0090] $$C\times(V_H-V_{MAC})+C\times(V_L-V_{MAC})+C\times(V_M-V_{MAC})+C\times(V_H-V_{MAC})=0$$

[0092] 수학식 1에서 C는 제1 내지 제4 삼진셀에 포함된 연산 커패시터의 커패시턴스를 나타낸다. Q1 내지 Q4는 제1 삼진셀 내지 제4 삼진셀 각각의 연산 커패시터에 저장된 전하량을 나타낸다. 참고로, 도 7b에서 설명한 바와 같이, 프리 차지 단계에서 TBA(701b) 내 삼진셀들의 연산 커패시터에 각각 저장된 전하량의 합에 대응하는 전체 전하량은 0의 값에 대응할 수 있다. 따라서, 전하량 보존 법칙에 의해 TBB(700c)의 열 별 MAC 연산에서의 연산 단계에서도 전체 전하량은 0을 유지한다. 따라서 Total Q는 0을 나타낸다. 방정식 Q=CV에 기초하여, 수학식 1에서 수학식 2를 도출할 수 있다. 수학식 2에서, V_{MAC}은 MAC 결과(720c)에 대응하는 전압 값을 나타낼 수 있다. 수학식 2에 기초하여, V_{MAC}은 $\frac{2\cdot V_H+V_M+V_L}{4}$ 를 나타낼 수 있다.

[0093] 다른 예를 들어, TBB(700c) 내에 288개의 삼진셀을 포함할 수 있다. 288개의 삼진셀 중, 입력 액티베이과 삼진 가중치 간의 MULT 연산 결과가 +1인 경우의 개수를 H, -1인 경우의 개수를 L, 0인 경우의 개수를 M이라고 나타낼 수 있다. 예를 들어, 삼진셀의 MULT 연산 결과는 출력 라인(710c)을 기준으로 2H+M, 2L+M의 경우(730c)일 수 있다. 이 때의 MAC 결과(720c)는 아래의 수학식 3에 의해 계산될 수 있다.

수학식 3

[0094]
$$V_{MAC} = \frac{2H + M}{2H + 2L + 2M} \cdot V_H$$

[0095] 도 8은 일 실시예에 따른 TNN 연산 가속 장치에 포함된 삼진셀에서 MAC 연산이 수행될 때, 결정 임계값을 적용한 결과를 비교 실시예와 그래프를 통해 비교한 도면이다.

[0097] 일 실시예에 따른 하나 이상의 삼진셀의 각각은, 제2 트랜지스터 페어의 타단에서 수신된 하이 레벨 전압에 응답하여, 연산 커패시터 및 제2 트랜지스터 페어 간의 계산 노드(node)에서, 증가하는 전압이 미리 결정된 임계 전압 값에 도달하는 것에 기초하여, 연산 결과를 출력 라인에게 제공할 수 있다.

[0098] 일 실시예에 따른 삼진셀은 전하 기반의 MAC 연산을 수행할 수 있다. 전하 기반 연산은 낮은 에너지와 좋은 선형성을 달성할 수 있으나, NMOS를 통해 구현되는 경우 느린 전압 상승을 야기할 수 있다. 일 실시예에 따른 TNN 연산 가속 장치는 제2 트랜지스터 페어의 제4 트랜지스터(예: T3)의 일단에 하이 레벨 전압이 인가되는 것에 응답하여, 계산 노드 전압(800)(예: V_{CN})을 미들 레벨 전압에서 하이 레벨 전압으로 증가시킬 수 있다.

[0099] 비교 실시예(801)는 계산 노드 전압(800)이 미들 레벨 전압에서 하이 레벨 전압까지 상승하는 시간을 그래프로 나타낸다. 비교 실시예(801)는 미들 레벨 전압에서 하이 레벨 전압까지 상승하는데 상대적으로 긴 MAC 연산 시간을 나타낼 수 있다.

[0100] 그래프(802)는 일 실시예에 따른 TNN 연산 가속 장치가 계산 노드 전압(800)이 증가함으로써 미리 결정된 임계 전압 값에 도달하면 MAC 연산 수행을 멈추는 것을 나타낸다. 예를 들어, 미리 결정된 임계 전압 값은 하이 레벨 전압과의 전압 오차(810)가 MAC 연산 결과에 미치는 영향을 최소로 하면서 연산 속도를 향상할 수 있는 값으로 결정될 수 있다. 예를 들어, TNN 연산 가속 장치는 전압 오차(810)를 0.5mV로 설정함으로써, 계산 노드 전압(800)이 하이 레벨 전압보다 0.5mV 작은 전압 값에 도달하는 경우 MAC 연산을 중단하여 MAC 연산 시간을 단축할 수 있다. 다른 예를 들어, TNN 연산 가속 장치는 MAC 연산 중단을 위한 임계 시간(820)을 결정할 수 있다. 예를 들어, 임계 시간(820)은 3.0ns로 결정되고, 해당 임계 시간(820)은 MAC 결과가 미리 결정된 임계 전압 값에 도달하기 위한 가장 긴 MAC 연산 시간일 수 있다. 그래프(802)에 따르면, 임계 전압 값 또는 임계 시간이 설정된 TNN 연산 가속 장치는 비교 실시예(801) 대비 70% 단축된 MAC 연산 시간을 달성할 수 있다. TNN 연산 가속 장치는 단축된 MAC 연산 시간에 기초하여, 비교 실시예(801)에 비해 빠르게 MAC 연산 결과를 CPA에 전달함으로써 연산 속도를 향상할 수 있다.

[0102] 도 9a 내지 도 9i는 CPA 회로들의 구조와 동작 방법을 설명하는 도면이다.

[0103] 일 실시예에 따른 CPA(Charge based Partial sum Adder) 회로는 전하 홀더들, 아날로그 메모리 회로들, 및 리셋 스위치를 포함할 수 있다. 예를 들어, CPA 회로는 4개의 전하 홀더들, 32개의 아날로그 메모리 회로들 및 1개의 리셋 스위치를 포함할 수 있다. CPA 회로는 TBB에 포함된 하나 이상의 TBA와 연결되어, TBA로부터 MAC 결과를 수신할 수 있다. CPA 회로는 스위치를 ON/OFF 함으로써 하나 이상의 TBA에서 MAC 결과의 수신 여부를 결정할 수 있다.

[0104] 리셋 스위치는 전하 홀더들 및 아날로그 메모리 회로들과 연결되어, 전하 홀더들 및 아날로그 메모리 회로를 리셋할 수 있다. 리셋 스위치에 의한 CPA의 리셋 동작은 도 9a에서 구체적으로 설명한다.

[0105] 전하 홀더들 각각은 커패시터 및 스위치를 포함할 수 있다. 전하 홀더들 각각에 포함된 커패시터의 커패시턴스 값은 다를 수 있다. 예를 들어, 제1 전하 홀더에 포함된 커패시터의 커패시턴스 값이 C에 해당하고, 제2 전하 홀더에 포함된 커패시터의 커패시턴스 값이 C/2에 해당하며, 제3 전하 홀더에 포함된 커패시터의 커패신턴스 값이 C/3에 해당하고, 제4 전하 홀더에 포함된 커패시터의 커패시턴스 값이 C/4에 해당할 수 있다. 각 전하 홀더에 포함된 커패시터의 커패시턴스 값은 이에 제한되는 것은 아니며, 이하에서는 전하 홀더들 각각에 포함된 커

패시터의 커패시턴스 값이 C, C/2/, C/3, 및 C/4인 경우를 중점적으로 설명한다. 전하 홀더들 각각은 TBA로부터 수신된 MAC 결과에 해당하는 전압 값을 전하로 저장할 수 있다.

[0106] 아날로그 메모리 회로들 각각은 부분합 커패시터 및 스위치를 포함할 수 있다. 아날로그 메모리 회로들 각각에 포함된 부분합 커패시터의 커패시턴스 값은 위에서 설명한 전하 홀더에 포함된 제1 커패시터의 커패시턴스 값과 같을 수 있다. 아날로그 메모리 회로들 각각은 복수의 전하 홀더에 저장된 전하들로부터 부분합 커패시터로의 순차적 전하 공유에 기초하여, 전하 홀더들에 저장된 전하들을 순차적으로 누적한 최종 MAC 결과를 생성할 수 있다.

[0107] 도 9a는 CPA 회로에서 전하 홀더 및 아날로그 메모리 회로를 리셋하는 동작을 설명하는 도면이다.

[0108] CPA 회로(900a)는 부분합 연산을 수행하기에 앞서, 리셋 동작을 수행할 수 있다. CPA 회로(900a)는 전하 홀더들 각각에 포함된 스위치(921a, 922a, 923a, 924a) 및 리셋 스위치(910a)와 아날로그 메모리 회로를 연결하는 스위치(930a)를 ON 함으로써, 리셋 스위치(910a)의 일단에서 인가되는 미들 레벨 전압 값에 해당하는 신호를 전하 홀더들 각각에 포함된 커패시터 및 아날로그 메모리 회로에 포함된 커패시터로 전달할 수 있다. 전하 홀더들 각각에 미들 레벨 전압 값에 해당하는 신호가 전달됨으로써, 전하 홀더들 각각은 0의 연산 결과를 의미하는 미들 레벨 전압 값을 저장할 수 있다. 전하 홀더들 각각에 미들 레벨 전압 값이 저장된 상태가 리셋 상태를 나타낼 수 있다.

[0109] 도 9b는 CPA 회로에서 리셋 동작 이후 전하 홀더에 데이터를 홀딩하는 동작을 설명하는 도면이다.

[0110] CPA 회로(900b)는, 전하 홀더들 중 제1 전하 홀더(예: charge holder #0)에 포함된 스위치(921b)는 ON하고, 나머지 전하 홀더들(예: charge holder #1, 2, 3)에 포함된 스위치는 OFF할 수 있다. CPA 회로(900b)는 스위치(930b)을 OFF 함으로써, 리셋 스위치로부터 아날로그 메모리 회로로 전달되는 미들 레벨 전압 값을 차단할 수 있다. CPA 회로(900b)는 TBA로부터 MAC 결과를 수신하여, MAC 결과에 해당하는 전압 값을 전하로 전하 홀더에 저장할 수 있다. 예를 들어, CPA 회로(900b)는 TBA #0과 CPA 회로(900b)를 연결하는 스위치(940b)를 ON함으로써, TBA #0으로부터 MAC 결과를 수신하고, TBA #0으로부터 수신된 MAC 결과에 해당하는 전압 값을 전하 형태로 전하 홀더(예: charge holder #0)에 포함된 커패시터에 저장할 수 있다. 다시 말해, CPA 회로는 전하 홀더를 통해, TBA로부터 수신된 MAC 결과를 부분합 연산 전에 홀딩(holding)할 수 있다.

[0111] 도 9c는 CPA 회로에서 홀딩된 데이터가 전하 홀더 및 아날로그 메모리 회로 간의 전하 공유를 통해 아날로그 메모리 회로에서 부분합 되는 것을 도시한다.

[0112] 일 실시예에 따른 TNN 연산 가속 장치에 포함된 CPA 회로(900c)는 TBA와 CPA 회로(900c)를 연결하는 스위치(910c)를 OFF함으로써 MAC 결과의 수신을 중단할 수 있다. CPA 회로(900c)는 스위치(930c)를 ON하고 동시에 아날로그 메모리 회로에 포함된 스위치(940c)를 ON함으로써, 전하 홀더와 아날로그 메모리 회로를 연결할 수 있다. CPA 회로(900c)는 스위치(930c) 및 스위치(940c)가 ON된 것에 응답하여, 전하 홀더에 포함된 커패시터와 아날로그 메모리 회로에 포함된 커패시터 간의 전하 공유를 수행할 수 있다. CPA 회로(900c)는 전하 공유에 기초하여, 아날로그 메모리 회로의 커패시터에 MAC 결과를 축적할 수 있다.

[0113] 도 9d는 CPA 회로에서 부분합 동작이 수행된 후, 전하 홀더를 다시 리셋하는 동작을 설명하는 도면이다.

[0114] CPA 회로(900d)는 부분 합 동작을 수행한 후, 스위치(940d) 및 스위치(950d)를 OFF 하고 스위치(930d)를 ON함으로써, 리셋 스위치(910d)와 전하 홀더(예: 전하 홀더#0)를 연결할 수 있다. CPA 회로(900d)는 미들 레벨 전압 값을 리셋 스위치(910d)로부터 전하 홀더 #0으로 전달함으로써, 전하 홀더 #0의 커패시터를 리셋할 수 있다.

[0115] 도 9e 내지 도 9i는 CPA 회로가 하나 이상의 TBA에서 순차적으로 MAC 결과들을 수신하고, 수신된 MAC 결과들을 누적함으로써 부분 합을 수행하는 것을 도시한다.

[0116] 도 9e에서, 일 실시예에 따른 CPA 회로(900)는 제1 TBA(901e)(예: TBA #0)으로부터 출력 라인(930e)를 통해 수신된 제1 MAC 결과(940e)를 제1 전하 홀더(910e)에 저장할 수 있다. CPA 회로(900)의 제1 전하 홀더(910e)에 MAC 결과를 저장하는 동작에 대하여는 도 9b에서 설명한 바와 같으므로 중복되는 내용은 생략한다. CPA 회로(900)는 제1 전하 홀더(910e)에 MAC 결과를 저장하기에 앞서, 전하 홀더들 및 아날로그 메모리 회로들을 리셋하였으므로, 아날로그 메모리 회로에 저장된 전하에 의한 전압 값(예: $V_{AM\#0}$)은 V_M과 같다.

[0117] 도 9f에서, 일 실시예에 따른 CPA 회로(900)는 스위치(910f)를 OFF 함으로써, 제1 TBA(901f)(예: TBA #0)로부

터 수신되는 MAC 결과를 차단하고, 스위치(920f)를 ON 함으로써, 제1 전하 홀더(930f)에 저장된 MAC 결과를 커패시터 간 전하 공유를 통해 아날로그 메모리 회로(940f)에 누적할 수 있다. 아날로그 메모리 회로(940f)에 누적된 전하에 대응하는 전압 값은 아래 수학식 4로 계산될 수 있다.

수학식 4

[0118] $$V_{AM\#0} = (V_M + V_{MAC,IC[0:31]})/2$$

[0119] 수학식 4에서 $V_{AM\#0}$은 아날로그 메모리 회로(940f)에 저장된 전압 값을 나타내고, $V_{MAC,IC[0:31]}$은 제1 TBA(901f)(예: TBA #0)로부터 수신된 MAC 결과에 대응하는 전압 값을 나타낼 수 있다.

[0120] 도 9g에서, 일 실시예에 따른 CPA 회로(900)는 제1 전하 홀더(910g)와 아날로그 메모리 회로(920g) 간의 전하 공유를 통한 부분합 연산이 종결된 후, 제2 TBA(901g)(예: TBA #1)로부터 수신된 제2 MAC 결과(930g)를 제2 전하 홀더(940g)에 저장할 수 있다. 이 때, 제1 전하 홀더(910g)의 스위치를 OFF 함으로써, 제1 전하 홀더(910g)와 출력 라인 간의 연결을 차단할 수 있다. CPA 회로(900)는 제2 전하 홀더(940g)에 제2 MAC 결과(930g)를 저장한 후, 제2 전하 홀더(940g)와 제2 TBA(901g) 사이의 스위치를 OFF함으로써, 제2 전하 홀더(940g)와 제2 TBA(901g)의 연결을 차단할 수 있다. CPA 회로(900)는 제2 전하 홀더(940g)와 제2 TBA(901g) 사이의 스위치를 OFF하고, 제2 전하 홀더(940g)와 아날로그 메모리 회로(920g) 사이의 스위치들을 ON함으로써, 제2 전하 홀더(940g)에 저장된 제2 MAC 결과(930g)에 대응하는 전하를 아날로그 메모리 회로(920g)와 공유할 수 있다. 전하 공유에 의해 아날로그 메모리 회로(920g)에 축적된 전하는 제1 전하 홀더(910g)와 공유된 전하 및 제2 전하 홀더(940g)와 공유된 전하가 누적된 값을 포함할 수 있다. 아날로그 메모리 회로(920g)에 누적된 전하에 대응하는 전압 값은 아래 수학식 5로 계산될 수 있다.

수학식 5

[0121] $$V_{AM\#0} = (V_M + V_{MAC,IC[0:63]})/3$$

[0122] 수학식 5에서 $V_{AM\#0}$은 아날로그 메모리 회로(920g)에 제1 전하 홀더(910g) 및 제2 전하 홀더(940g)에 의한 전하 공유에 의해 누적된 전하 값에 대응하는 전압 값을 나타내고, $V_{MAC,IC[0:63]}$은 제1 TBA(예: TBA #0) 및 제2 TBA(901g)로부터 수신된 MAC 결과의 누적 값에 대응하는 전압 값을 나타낼 수 있다.

[0124] 도 9h 에서, 일 실시예에 다른 CPA 회로(900)는 제2 전하 홀더(910h)와 아날로그 메모리 회로(920h) 간의 전하 공유를 통한 부분합 연산이 끝난 후, 제3 TBA(901h)(예: TBA #2)로부터 수신된 제3 MAC 결과(930h)를 제3 전하 홀더(940h)에 저장할 수 있다. 이 때, 제2 전하 홀더(910h)의 스위치를 OFF 함으로써, 제2 전하 홀더(910h)와 출력 라인 간의 연결을 차단할 수 있다. CPA 회로(900)는 제3 전하 홀더(940h)에 제3 MAC 결과(930h)를 저장한 후, 제3 전하 홀더(940h)와 제3 TBA(901h) 사이의 스위치를 OFF함으로써, 제3 전하 홀더(940h)와 제3 TBA(901h)의 연결을 차단할 수 있다. CPA 회로(900)는 제3 전하 홀더(940h)와 제3 TBA(901h) 사이의 스위치를 OFF하고, 제3 전하 홀더(940h)와 아날로그 메모리 회로(920h) 사이의 스위치들을 ON함으로써, 제3 전하 홀더(940h)에 저장된 제3 MAC 결과(930h)에 대응하는 전하를 아날로그 메모리 회로(920h)와 공유할 수 있다. 전하 공유에 의해 아날로그 메모리 회로(920h)에 축적된 전하는 제1 전하 홀더(예: charge holder #0) 내지 제3 전하 홀더(940h) 각각과 전하 공유함으로써 누적된 전하 값을 포함할 수 있다. 아날로그 메모리 회로(920h)에 누적된 전하에 대응하는 전압 값은 아래 수학식 6로 계산될 수 있다.

수학식 6

[0125] $$V_{AM\#0} = (V_M + V_{MAC,IC[0:95]})/4$$

[0126] 수학식 6에서 $V_{AM\#0}$은 아날로그 메모리 회로(920h)에 제1 전하 홀더(예: charge holder #0) 내지 제3 전하 홀더(940h)에 의한 전하 공유에 의해 누적된 전하 값에 대응하는 전압 값을 나타내고, $V_{MAC,IC[0:95]}$은 제1 TBA(예: TBA #0) 내지 제3 TBA(901h)로부터 수신된 MAC 결과의 누적 값에 대응하는 전압 값을 나타낼 수 있다.

[0127] 도 9i에서, 일 실시예에 따른 CPA 회로(900)는 제3 전하 홀더(910h)와 아날로그 메모리 회로 간의 전하 공유를 통한 부분합 연산이 종결된 후, 제4 TBA(901i)(예: TBA #3)로부터 수신된 제4 MAC 결과(예: MAC IC[96:127])를 제4 전하 홀더(910i)에 저장할 수 있다. 도 9f 내지 도 9h에서 설명한 CPA 회로(900)의 동작을 통해, 아날로그 메모리 회로(920i)는 부분합 연산 결과 값에 대응하는 누적된 전하 값을 가질 수 있다. 아날로그 메모리 회로(920i)에 누적된 전하에 대응하는 전압 값은 아래 수학식 7로 계산될 수 있다.

수학식 7

[0128] $$V_{AM\#0} = (V_M + V_{MAC,IC[0:127]})/5$$

[0129] 수학식 7에서 $V_{AM\#0}$은 아날로그 메모리 회로(920h)에 제1 전하 홀더(예: charge holder #0) 내지 제4 전하 홀더(910i)에 의한 전하 공유에 의해 누적된 전하 값에 대응하는 전압 값을 나타내고, $V_{MAC,IC[0:127]}$은 제1 TBA(예: TBA #0) 내지 제4 TBA(901i)로부터 수신된 MAC 결과의 누적 값에 대응하는 전압 값을 나타낼 수 있다.

[0131] 도 10a 내지 10h는 일 실시예에 따른 TNN 연산 가속 장치의 TNN 연산 수행을 위한 유연한 데이터 매핑 전략을 도시한다.

[0132] 도 10a는 일 실시예에 따른 TNN 연산 가속 장치를 동작할 때, 전체 TNN 컨볼루션 레이어들을 실행하는데 필요한 메모리 크기를 줄이기 위한 효율적인 소프트웨어-대-하드웨어 데이터 매핑 전략을 도시한다. 일 실시예에 따른 TNN 연산 가속 장치는 입력 액티베이션(1010a)에 가중치 커널(1020a)을 적용함으로써, 출력 액티베이션을 생성할 수 있다.

[0133] 입력 액티베이션(1010a)은 복수의 입력 채널 맵들을 포함할 수 있다. 입력 액티베이션(1010a)은 예를 들어, 128개의 입력 채널 맵들을 가질 수 있다.

[0134] 가중치 커널(1020a)은 입력 액티베이션(1010a)에 포함된 입력 채널 맵들의 수만큼의 채널 가중치 커널들을 포함할 수 있다. 예를 들어, 입력 액티베이션(1010a)에 포함된 입력 채널 맵들이 128개인 경우, 채널 가중치 커널들도 128개일 수 있다. 각 채널 가중치 커널은 삼진 가중치를 포함하는 3 by 3 크기일 수 있다.

[0135] 일 실시예에 따른 TNN 연산 가속 장치의 서로 다른 4개의 TBA 각각은 파이프 라인으로, 다른 TBB에 포함된 해당 TBA와 연결될 수 있다. 예를 들어 TBB #0에 포함된 TBA #0은 TBB #1에 포함된 TBA #0과 병렬적으로 연결될 수 있다. 서로 다른 TBA 각각은 32개의 입력 신호들(예: IC[0:31]- IC[32:63]- IC[64:95]- IC[96:127])에 대한 커널 연산에 대응하는 MAC 연산을 수행할 수 있다. TBA의 각 어레이에 배열된 삼진셀은 다음 32개의 MAC 연산 결과(예: OC)에 따라 삼진 가중치를 업데이트 할 수 있다.

[0136] 도 10b는 일 실시예에 따른 TNN 연산 가속 장치에 의해 수행되는 TNN 연산에 대한 데이터 매핑 전략에 대해 개괄적으로 도시한다.

[0137] 일 실시예에 따른 TNN 연산 가속 장치는 입력 신호(1010b)에 대한 TNN 연산(1000b)을 수행할 수 있다. 참고로, 입력 신호(1010b)는 RGB채널맵을 포함하는 32 by 32의 입력 이미지일 수 있다. 32 by 32의 입력 이미지는 각 픽셀 값으로 INT8(예: 0 내지 255에 대응하는 값)을 포함할 수 있다. TNN은 복수의 레이어들을 포함할 수

있다. TNN 연산(1000b)은 복수의 레이어들을 거쳐 깊어지는 방향(1020b)으로 수행될 수 있다. TNN 연산 가속 장치는 입력 신호(1010b) 및 TNN의 각 레이어의 출력(예: 피처 맵)에 대해 TNN이 깊어지는 방향(1020b)으로 3 by 3 컨볼루션 연산 또는 2 by 2 맥스 풀링 연산(max pooling operation) 중 적어도 하나를 수행함으로써, TNN 연산을 수행할 수 있다.

[0138] 도 10c는 일 실시예에 따른 TNN 연산 가속 장치에 의해 수행되는 TNN 연산 중 제2 레이어(1010c)에서 수행되는 커널 연산을 도시한다. 참고로, 입력 이미지는 R채널, G채널, 및 B채널과 같은 서로 다른 3개의 채널 각각의 32 by 32의 이미지에 대응할 수 있다. 입력 이미지의 각 픽셀 값은 INT8(예: 0 내지 255)의 비트를 포함할 수 있다. 해당 입력 이미지에 대하여, 일 실시예에 따른 TNN 연산 가속 장치와 연결된 CPU, GPU 또는 FPGA 중 적어도 하나에서 컨볼루션 연산이 수행될 수 있다. 해당 입력 이미지에 대한 컨볼루션 연산 결과는 예를 들어, +1, -1, 또는 0 중 적어도 하나의 데이터를 포함하는 삼진 액티베이션 채널 맵에 대응할 수 있다. 삼진 액티베이션 채널 맵들은 제2 레이어(1010c)에 대한 입력 액티베이션 채널 맵들에 해당할 수 있다. 예를 들어, TNN 연산 가속 장치에 포함된 각 TBA(예: TBA #0, TBA #1, TBA #2, 및 TBA #3)의 하나의 열(column)(예: 288개의 삼진셀이 배열된 열)은 3 by 3 가중치 커널에 포함된 삼진 가중치에 대응하는 값을 저장할 수 있다. 예를 들어, TBB #0, TBB #1, 및 TBB #2에 포함된 TBA들 각각이 서로 병렬적으로 연결되는 경우, 일 실시예에 따른 TNN 연산 가속 장치는 128개의 입력 액티베이션 채널 맵들의 수에 대응하는 3 by 3 가중치 커널을 저장할 수 있다. 예를 들어, 128개의 입력 액티베이션 채널 맵들의 수에 대응하여, 32개의 입력 액티베이션 채널 맵들씩 나뉘어 3 by 3 가중치 커널이 저장될 수 있다. 예를 들어, TBA #0에 0 내지 31의 입력 액티베이션 채널 맵에 대응하는 3 by 3 가중치 커널이, TBA #1에는 32 내지 63의 입력 액티베이션 채널 맵에 대응하는 3 by 3 가중치 커널이 저장될 수 있다.

[0139] TNN 연산 가속 장치에 포함된 TBA는 복수의 열들을 포함할 수 있고, 복수의 열들의 개수는 출력 채널의 개수를 나타낼 수 있다. 예를 들어, TBA에 32개의 열(column)들이 포함되는 경우, 32개의 열들은 출력 채널의 개수를 나타낼 수 있다. 제2 레이어(1010c)에서의 입력 액티베이션 채널 맵이 128개, 출력 액티베이션 채널 맵이 128개인 경우에 대응하여, TNN 연산 가속 장치는 32개의 열들을 포함하는 TBA에 기초하는 4번의 TNN 연산 동작을 수행할 수 있다. TNN 연산 가속 장치는 4번의 TNN 연산 동작을 수행함으로써, 32개의 3 by 3 크기의 가중치 커널(1020c)을 입력 액티베이션 채널 맵들에 적용하여 총 128개의 출력 액티베이션 채널 맵들을 생성할 수 있다.

[0140] 도 10d는 일 실시예에 따른 TNN 연산 가속 장치에 의해 수행되는 TNN 연산 중 제3 레이어(1010d)에서 수행되는 커널 연산을 도시한다. 제3 레이어(1010d)에서 TNN 연산 가속 장치는 3 by 3 컨볼루션 및 2 by 2 맥스 풀링 연산을 수행할 수 있다. 도 10c에서 설명한 바와 같이, TNN 연산 가속 장치는, TNN의 제3 레이어(1010d)에서의 가중치 연산에 대응하여, 128개의 입력 액티베이션 채널 맵들의 수에 대응하는 3 by 3 가중치 커널을 저장할 수 있다. 예를 들어, 제3 레이어(1010d)에서의 출력 액티베이션 채널 맵들의 수는 256개에 대응하는 경우, TNN 연산 가속 장치는, 256개의 출력 채널 맵들을 생성하도록, 128개의 입력 액티베이션 채널 맵들에 대해 가중치 커널(1020d) 연산을 총 8회 반복함으로써, 제3 레이어(1010d)에서의 TNN 연산을 수행할 수 있다.

[0141] 도 10e는 일 실시예에 따른 TNN 연산 가속 장치에 의해 수행되는 TNN 연산 중 제4 레이어(1010e)에서 수행되는 커널 연산을 도시한다. TNN 연산 가속 장치는 제4 레이어(1010e)에서 256개의 입력 액티베이션 채널 맵들에 대해 가중치 커널(1020e) 연산을 총 16회 반복함으로써, 제4 레이어(1010e)에서의 TNN 연산을 수행할 수 있다.

[0142] 도 10f는 일 실시예에 따른 TNN 연산 가속 장치의 정규화 함수 및 삼진화 함수를 포함하는 TNN 연산 블록도를 도시한다.

[0143] 블록도(1000f)는 복수의 레이어를 포함하는 TNN을 나타낸다. 복수의 레이어 각각은 TNN 연산을 위하여 컨볼루션 연산(또는, 컨볼루션 연산 및 맥스 풀링 연산) 후 컨볼루션 결과에 대한 정규화(normalization) 및 삼진화(ternarization)를 수행하는 함수를 포함할 수 있다. 예를 들어, TNN의 각 레이어는 각 레이어에서 수행되는 컨볼루션 연산의 결과값을 해당 레이어의 다음 레이어의 입력 액티베이션으로 활용하기 위한 정규화 함수(normalization function) 및 삼진화 함수(ternarize function)를 포함할 수 있다. 예를 들어, 정규화 함수는 배치 정규화 함수(batch normalization function)를 포함할 수 있다.

[0144] TNN의 각 레이어에 포함된 배치 정규화 함수 및 삼진화 함수를 구현하기 위한 회로에 대하여 도 10g에서 구체적으로 설명한다.

[0145] 도 10g는 일 실시예에 따른 TNN 연산 가속 장치에 포함된 삼진화 함수 및 배치 정규화 함수를 구현하기 위한 회

로도이다.

[0146] 회로(1000g)는 배치 정규화 함수 및 삼진화 함수의 융합을 구현한 회로(1000g)이다. 회로(1000g)는 수학식 8 내지 수학식 9에 대응하는 연산에 기초하여, 수학식 10에 대응하는 연산을 수행할 수 있다.

수학식 8

[0147]
$$Y = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta$$

수학식 9

[0149]
$$Z = f(Y) = \begin{cases} -1, & \text{if } Y < -0.5 \\ 0, & \text{if } -0.5 \leq Y \leq +0.5 \\ +1, & \text{if } Y > +0.5 \end{cases}$$

수학식 10

[0151]
$$Z = f'(X) = \begin{cases} -1, & \text{if } X < X_{ref,1} \\ 0, & \text{if } X_{ref,1} \leq X \leq +X_{ref,2} \\ +1, & \text{if } X > +X_{ref,2} \end{cases}$$

Where

$$X_{ref,2} = \frac{(0.5 - \beta) \cdot \left(\sqrt{\sigma^2 + \epsilon}\right)}{\gamma} + \mu$$

$$X_{ref,1} = -\frac{(0.5 + \beta) \cdot \left(\sqrt{\sigma^2 + \epsilon}\right)}{\gamma} + \mu$$

[0152] 수학식 8은 전통 배치 정규화 함수식을 나타내고, 수학식 9는 전통 삼진화 함수를 나타낼 수 있다. 수학식 8내지 수학식 10에서, X는 입력 값, μ는 미니 배치 평균(mini-batch mean), σ^2은 미니 배치 분산(mini-batch variance), ϵ는 정규화 파라미터, β는 시프트(shift) 파라미터, γ는 스케일(scale) 파라미터, 및 Y는 정규화된 값을 나타낼 수 있다. 수학식 10은 수학식 8을 수학식 9에 대입하여, 미분한 결과를 나타낼 수 있다. 수학식 10에서, 생성된 Xref,2 및 Xref,1을 아날로그 전압으로 변환함으로써, Vref,1 및 Vref,2를 생성할 수 있다. 수학식 10에서 생성된 Vref,1 및 Vref,2는 Ternarizer의 참조 전압(reference voltage)에 해당할 수 있다. 회로(1000g)는 참조 전압에 기초하여, CPA로부터 수신된 최종 MAC 결과(예:PSUM) 값을 +1, -1 또는 0 중 적어도 하나로 삼진화할 수 있었다.

[0154] 도 10h는 일 실시예에 따른 TNN 연산 가속 장치의 동작 타이밍도를 나타낸다.

[0155] 타이밍도(1000h)에 따르면, TNN 연산 가속 장치의 각 TBA(예: TBA #0, #1, #2, #3)는 32개의 출력 값(예: OC[0:31])을 생성하기 위해, 32개의 입력 값들(예: IC[0:31], IC[32:63] 등) 각각 마다 총 32 사이클을 소비하고, 가중치 업데이트를 위해 96 사이클을 소비할 수 있다. TNN 연산 가속 장치는 각 TBA를 파이프방식으로 동작함으로써, 하나의 TBA가 동작하는 동안, 다른 TBA의 가중치를 업데이트 할 수 있다. 예를 들어, 제1 사이클(1010h) 동안 TBA #0은 IC[0:31] 입력 값에 대하여 연산을 수행하고, 동시에 TBA #1, #2, 및 #3은 TBA의 가중치를 업데이트 할 수 있다. TNN 연산 가속 장치는 TBA의 파이프라인 동작을 통해, 제안된 유연한 데이터 매핑 전략을 수행함으로써 고정된 하드웨어 리소스 및 낮은 메모리 용량으로 전체 컨볼루션 레이어들의 실행을 가능하게 한다. 논문 "S. Jain, S. K. Gupta and A. Raghunathan, "TiM-DNN: Ternary InMemory Accelerator for Deep Neural Networks," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 7, pp. 1567-1577, July 2020." 에 개시된 비교 실시예에서는 가중치 저장을 위해 524.29KB의 메모리가 필요하나, 일 실시예에 따른 TNN 연산 가속 장치의 삼진 가중치들에 대한 메모리 용량은 9.22KB로 비교 실시예에 비해 적다. 또한, 비교 실시예에 비하여, TNN 연산 가속 장치는 신경망 추론을 위한 CIM 아키텍처 메모리 용량을 56배 감소시킨다.

[0157] 도 11a 내지 도 11f는 일 실시예에 따른 TNN 연산 가속 장치가 TBA 및 CPA 회로를 파이프라인 동작시키는 것을 도시한다.

[0158] 일 실시예에 따른 TNN 연산 가속 장치는 TBB에 포함된 하나 이상의 TBA를 병렬적으로 각각 동작 시킬 수 있고, 하나 이상의 CPA 회로들을 병렬적으로 각각 동작 시킬 수 있다.

[0159] 도 11a에서, 일 실시예에 따른 TNN 연산 가속 장치는 입력 값에 대하여 제1 TBA(1110a)에 저장된 삼진 가중치와 연산을 수행함으로써 생성된 연산 결과(예: V_{MAC})를 제1 CPA 회로(1120a)로 전달할 수 있다. TNN 연산 가속 장치는 제1 CPA 회로(1120a)에서 제1 TBA(1110a)로부터 전달받은 연산 결과에 기초하여 부분합 연산을 수행할 수 있다.

[0160] 다시 말해, TNN 연산 가속 장치는 제1 사이클(1130a)에 제1 TBA(1110a) 및 제1 CPA 회로를 동작할 뿐, 다른 TBA들 및 다른 CPA 회로들은 동작하지 않을 수 있다.

[0161] 도 11b에서, 일 실시예에 따른 TNN 연산 가속 장치는 제1 사이클 이후의 제2 사이클(1130b)에 제1 TBA(예: TBA #0) 뿐만 아니라, 제2 TBA(1110b)를 동작함으로써, 입력 값과 제2 TBA(1110b)에 저장된 삼진 가중치 간의 연산 결과를 제1 CPA 회로 또는 제2 CPA 회로(1120b)로 전달할 수 있다. 예를 들어, TNN 연산 가속 장치는 제2 사이클(1130b)에 제2 CPA 회로(1120b)와 제1 TBA를 연결할 수 있고, 제1 CPA 회로에서 제2 TBA(1110b)로부터 전달받은 연산 결과에 기초하여, 부분합 연산을 수행할 수 있다. TNN 연산 가속 장치는 제2 사이클(1130b)에 제1 TBA, 제2 TBA(1110b) 및 제1 CPA 회로 및 제2 CPA 회로(1120b)를 병렬적으로 동작할 뿐, 다른 TBA들 및 다른 CPA 회로들은 동작하지 않을 수 있다.

[0162] 도 11c에서, 일 실시예에 따른 TNN 연산 가속 장치는 도 11b의 TNN 연산 가속 장치의 동작에 이어서, 제3 사이클(1130c)에 제1 TBA 내지 제3 TBA(1110c)를 동작할 수 있다. TNN 연산 가속 장치는, 입력 값과 제3 TBA(1110c)에 저장된 삼진 가중치 간의 연산 결과를 제1 CPA 회로 또는 제3 CPA 회로(1120c) 중 적어도 하나로 전달할 수 있다. 예를 들어, TNN 연산 가속 장치는 제3 사이클(1130c)에 제3 CPA 회로(1120c)와 제1 TBA를 연결할 수 있고, 제2 CPA회로와 제2 TBA를 연결할 수 있으며, 제1 CPA회로에서 제3 TBA(1110c)로부터 전달받은 연산 결과에 기초하여, 부분합 연산을 수행할 수 있다.

[0163] 도 11d에서, 일 실시예에 따른 TNN 연산 가속 장치는 도 11c의 TNN 연산 가속 장치의 동작에 이어서, 제4 사이클(1130d)에 제1 TBA 내지 제4 TBA(1110d)를 병렬적으로 동작할 수 있다. 또한, TNN 연산 가속 장치는 제4 사이클(1130d)에서 제4 CPA 회로(1120d)도 동작할 수 있다. TNN 연산 가속 장치의 제4 사이클에서의 부분합 연산 수행 동작은 위의 도11a 내지 11c에서 설명한 바와 같으므로 생략한다.

[0164] 도 11e에서와 같이, 일 실시예에 따른 TNN 연산 가속 장치는, 제5 사이클(1130e)에서, 제1 TBA(1110e)에 저장된 삼진 가중치와 입력 신호 간의 연산을 수행함으로써 연산 결과를 CPA 회로들에 전달할 수 있다. 이와 동시에 TNN 연산 가속 장치는 제5 사이클(1130e)에서 다른 TBA들(1120e)의 삼진 가중치를 업데이트 할 수 있다. 또한, 도 11f에서와 같이, 일 실시 예에 따른 TNN 연산 가속 장치는, 제6 사이클(1130f)에서, 제1 TBA(예: TBA #0) 및 제2 TBA(예: TBA #1)를 동작함으로써, 제1 TBA에 저장된 삼진 가중치와 입력 신호 간의 연산결과 및 제2 TBA에 저장된 삼진 가중치와 입력 신호 간의 연산 결과를 CPA 회로들에 전달할 수 있다. 동시에 제6 사이클

(1130f)에서 TNN 연산 가속 장치는, 다른 TBA들(1120f)의 삼진 가중치를 업데이트 할 수 있다. TNN 연산 가속 장치가 TBA들과 CPA 회로들을 병렬적으로 동작함으로써, TNN 연삭 가속 장치에 의한 TNN 연산 및 TBA들의 삼진 가중치 업데이트가 사이클 지연 없이 수행될 수 있다.

[0166] 도 12는 일 실시예에 따른 삼진셀의 레이아웃을 나타낸다.

[0167] 일 실시예에 따른 삼진셀(1200)은 2fF 크기의 커패시턴스를 가지는 MOMCAP(Metal Oxide Metal Capacitor)(1210) 및 16T를 포함하는 칩(1220)의 적층 구조를 포함할 수 있다. 예를 들어, 칩(1220)은 두 개의 6T SRAM 및 서로 다른 4개의 트랜지스터(예: T0, T1, T2, T3)의 조합으로 형성될 수 있다. MOMCAP(1210)과 칩(1220)이 적층된 삼진셀(1200)은 가로×세로가 4㎛×0.67㎛가 되도록 제작될 수 있다.

[0168] 도 13은 일 실시예에 따른 TNN 연산 가속 장치의 칩 레이아웃을 나타낸다.

[0169] 일 실시예에 따른 TNN 연산 가속 장치(1300)는 65nm CMOS 로직 기술을 활용하여 설계될 수 있다. 예를 들어, TNN 연산 가속 장치(1300)는 3개의 TBB, 1개의 탑 액티베이션 드라이버, 1개의 읽기/쓰기 인터페이스 및 4개의 CPA회로를 포함할 수 있다. TNN 연산 가속 장치(1300)의 가로×세로는 620㎛×460㎛에 해당할 수 있다. 아래의 표 4는 도13의 일 실시예에 따른 TNN 연산 가속 장치의 특징에 대한 표이다.

표 4

[0170]

Process	65 nm CMOS	
Area	0.29 mm^2	
V_{DD}	1.0 V	
Frequency	250 MHz	
Power	1.59 mW	
Capacity (Footprint)	9.22 KB	
Accuracy (CIFAR-10)	T-CIM	91.2%
	Baseline	92.8%
Peak Perf.	1,316 GOPS	
Energy Efficiency	*823 TOPS/W*	

[0171] 표 4에 기재된 바와 같이, 일 실시예에 따른 TNN 연산 가속 장치는 공급 전원(V_{DD})이 1.0V이고, 250 MHz의 동작 진동에서 1.59 mW의 전력을 소모할 수 있다. TNN 연산 가속 장치는 1,326 GOPS의 피크 성능과 823 TOP/W의 에너지 효율을 나타낼 수 있다.

[0172] 아래의 표 5는 비교 실시예들과 일 실시예에 따른 TNN 연산 가속 장치의 성능을 비교한 표이다.

표 5

		비교실시예 1	비교실시예 2	비교실시예 3	비교실시예 4	This Work
Technology		55 nm	65 nm	65 nm	32 nm	65 nm
Operating Voltage		1 V	1 V	1/0.8/0.6 V	1 V	1 V
Cell Structure		Twin-8T	10T1C	8T1C	19T	16T1C
Precision	I	1b~4b	Binary	Ternary	Ternary	Ternary
	W	2b~5b	Binary	Binary	Ternary	Ternary
Capacity (Footprint)(KB)		0.48	295	2.05	524.29	9.22
Accuracy (CIFAR-10)		90.4%	88.8%	85.5%	N/A	91.2%
Throughput (GOPS)		21.2~67.5	9,438	1,638	114,000 (114 TOPS)	1,316
Energy Efficiency (TOPS/W)		18.37~72.03	658	671.5	114	823
TOPS/W Comparison (SOTA/This Work)		1	35.8	36.6	6.2	44.8

[0173]

[0174] 표 5에서, 비교 실시예 1은, 논문 'X. Si et al., "A Twin-8T SRAM Computation-in-Memory Unit-Macro for Multibit CNN-Based AI Edge Processors," in IEEE J. Solid-State Circuits, vol. 55, no. 1, pp. 189-202, Jan. 2020.'에 개시된 트윈 8T SRAM 셀을 포함하는 CIM 소자를 나타낸다. 비교 실시예 2는, 논문 'H. Valavi, P. J. Ramadge, E. Nestler and N. Verma, "A Mixed-Signal Binarized Convolutional-Neural-Network Accelerator Integrating Dense Weight Storage and Multiplication for Reduced Data Movement," in Proc. IEEE Symp. VLSI Circuits (VLSIC), 2018, pp. 141-142.'에 개시된 10T1C 구조의 셀을 포함하는 CNN 연산 가속 장치를 나타낸다. 비교 실시예 3은, 논문 'Z. Jiang, S. Yin, J. -S. Seo and M. Seok, "C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism," in IEEE J. Solid-State Circuits, vol. 55, no. 7, pp. 1888-1897, July 2020.'에 개시된 8T1C 구조의 셀을 포함하는 CIM 장치를 나타낸다. 비교 실시예 4는, 논문 'S. Jain, S. K. Gupta and A. Raghunathan, "TiM-DNN: Ternary In-Memory Accelerator for Deep Neural Networks," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 7, pp. 1567-1577, July 2020.'에 개시된 19T 구조의 셀을 포함하는 DNN 연산을 위한 삼진 인메모리 가속기를 나타낸다. 비교 실시예 4와 TNN 연산 가속 장치를 비교하면, 제안된 16T1C의 삼진셀은 비교 실시예 4에 비해 적은 트랜지스터를 사용하여 12 % 더 적은 면적을 차지하고, 7.22 배 높은 에너지 효율성을 달성한다. TNN 연산 가속 장치는 미리 결정된 임계 전압값에서 MAC 연산을 중단함으로써 MAC 연산 시간을 단축하고, CPA 회로를 통해 비교 실시예 3에 비해 20% 높은 에너지 효율성을 달성한다. 또한, TNN 연산 가속 장치는 유연한 데이터 매핑 전략을 통해 비교 실시예 4에 비해 56 배의 가중치 메모리 용량의 감소를 달성한다.

[0176] 도 14는 미리 결정된 임계 전압 값과 삼진셀에 의한 MAC 연산의 정확도 간의 상관 관계를 그래프로 나타낸다.

[0177] 그래프(1400)는 일 실시예에 따른 TNN 연산 가속 장치에 포함된 전체 TBA에 대한 1000번의 Monte-Carlo 시뮬레이션들을 기반으로 제안된 TBB간 MAC 연산 결과의 변화 분석을 도시한다. 그래프(1400)는 MAC 연산 결과 변화에 따른 V_{MAC} 값의 변화를 나타낸다. V_{MAC} 값은 MAC 연산 결과들과 선형적인 관계를 따른다. MAC 값이 0일 때, V_{MAC} 값의 분포 범위는 324.0mV 부터 324.9mV이다. 또한 MAC 값이 0일 때, V_{MAC} 값의 평균은 324.3mV이고, 표준 편차는 0.17mV이다.

[0178] 그래프(1410)는 일 실시예에 따른 TNN 연산 가속 장치의 MAC 값에 따른 MAC 연산 정확도를 미리 결정된 임계 전압의 값에 따라 나타낸다. 그래프(1410)에서 TNN 연산 가속 장치는 미리 결정된 임계 전압의 값을 설정하기 위해 ΔV 값을 활용한다. ΔV 가 크면 MAC 연산 정확도가 떨어지지만, 미리 결정된 임계 전압의 값까지 도달하

는 시간을 직접적으로 줄일 수 있다. 그래프(1410)에서, MAC 연산 정확도는 ΔV가 0.5mV보다 커지면(예: ΔV=0.6mV, 0.7mV 또는 1.0mV 인 경우) 눈에 띄게 감소한다. 따라서, ΔV를 0.5mV로 설정하는 경우 높은 컴퓨팅 정확도를 유지하는 플롯(1420)을 확인할 수 잇다.

[0180] 도 15는 CPA에 의한 부분 합 값에 따른 최종 V_{PSUM} 값의 선형성을 나타내는 그래프 및 ADC를 사용하는 비교 실시예와 비교하여 TNN 연산 가속 장치의 에너지 효율성 향상을 나타내는 그래프를 도시한다.

[0181] 그래프(1500)는 CPA에 의한 부분 합 값에 따른 최종 V_{PSUM} 값의 선형성을 나타낸다. 부분합 값이 0일 때, 최종 V_{PSUM} 값은 최소 323mV 내지 최대 324mV로 1mV 범위내에 존재한다. 또한 부분합 값이 0일 때, 최종 V_{PSUM} 값은 평균 323.49mV 및 0.17mV의 표준편차를 가진다.

[0182] 그래프(1510) 및 그래프(1520)는 ADC를 사용하는 비교 실시예와 CPA 회로를 사용하는 TNN 연산 가속 장치 간의 에너지 효율성을 비교하는 그래프다. TNN 연산 가속 장치는 CPA 회로를 사용함으로써, 3.7배 향상된 전력 소모를 나타내며, 7.2 배의 에너지 효율성 향상을 달성한다.

[0184] 도 16은 CPA 회로에 의해 수행되는 부분합 연산을 통해 생성된 최종 MAC 결과에 대한 그래프다.

[0185] 그래프(1600)는 각 타입 스텝에서의 CPA 회로의 동작에 따른 최종 MAC 결과(예: V_{PSUM})를 도시한다. TNN 연산 가속 장치는 CPA 회로를 리셋한 후, 전체 입력 값(예: IC[0:127])에 대한 최종 MAC 결과에 해당하는 전압 값을 방정식 $V_{PSUM} = \frac{1}{5}(V_M + \sum_{i=0}^{3} V_{MAC,IC[32\cdot i:32\cdot i+31]})$ 과 같이 전하 도메인에서 4 스텝 동안 축적한다. 예를 들어, 제1 플롯(1610)은 MULT 값이 모두 +1이고, 각 스텝에서 +288의 MAC 결과가 아날로그 메모리 회로에 더해지는 경우를 나타낸다. 제2 플롯(1620)은 제1 플롯(1610)과 반대 경우로서, MULT 값들이 -1이고, -288 MAC 결과가 각 스텝에서 아날로그 메모리 회로에 축적된다. 제3 플롯(1630)은 최종 MAC 결과가 +576인 경우를 나타낸다. 제4 플롯(1640) 및 제5 플롯(1650)은 최종 MAC 결과가 0인 두가지 경우들을 나타낸다. 제4 플롯(1640)은 각 스텝에서 0인 MAC 결과만이 아날로그 메모리 회로에 누적됨을 나타내고, 제5 플롯(1650)은 -288과 +288의 MAC 결과가 각 스텝 마다 교차적으로 누적됨을 나타낸다. 최종 MAC 결과는 제4 플롯(1640) 및 제5 플롯(1650) 모두 325mV로 수렴한다.

[0187] 도 17은 일 실시예에 따른 TNN 연산 가속 장치가 동작함에 있어서 유연한 데이터 매핑 전략을 활용한 경우의 메모리 용량을 비교 실시예의 메모리 용량과 대비하는 그래프다.

[0188] 그래프(1710) 및 그래프(1720)는 유연한 데이터 매핑 전략을 활용하는 TNN 연산 가속 장치와, 유연한 데이터 매핑 전략을 활용하지 않는 비교 실시예 간의 차지하는 메모리 용량을 나타낸다. 비교 실시예에 비하여, TNN 연산 가속 장치는, 유연한 데이터 매핑 전략을 사용하는 경우 57배 또는 62배 향상된 메모리 용량을 달성한다.

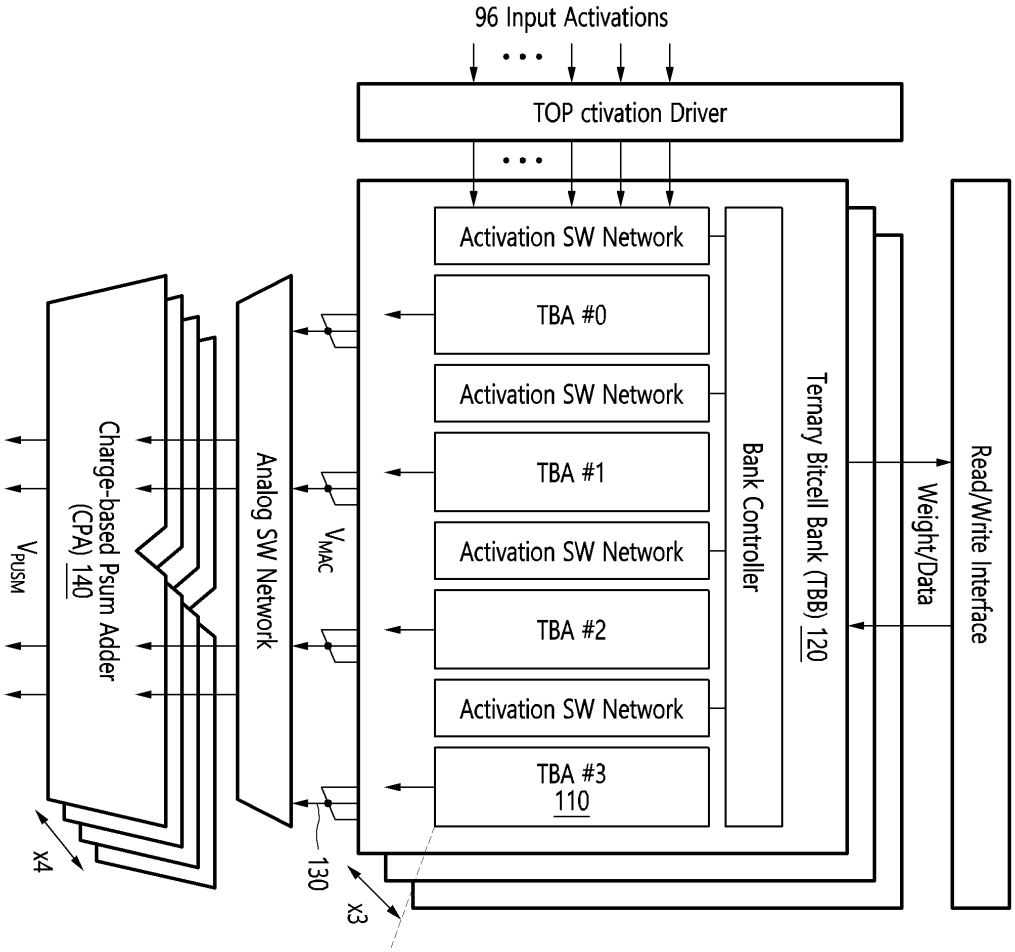
[0190] 본 문서에서, "A 또는 B", "A 및 B 중 적어도 하나", "A 또는 B 중 적어도 하나", "A, B 또는 C", "A, B 및 C 중 적어도 하나", 및 "A, B, 또는 C 중 적어도 하나"와 같은 문구들 각각은 그 문구들 중 해당하는 문구에 함께 나열된 항목들 중 어느 하나, 또는 그들의 모든 가능한 조합을 포함할 수 있다.

[0191] 이상과 같이 실시예들이 비록 한정된 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 이를 기초로 다양한 기술적 수정 및 변형을 적용할 수 있다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

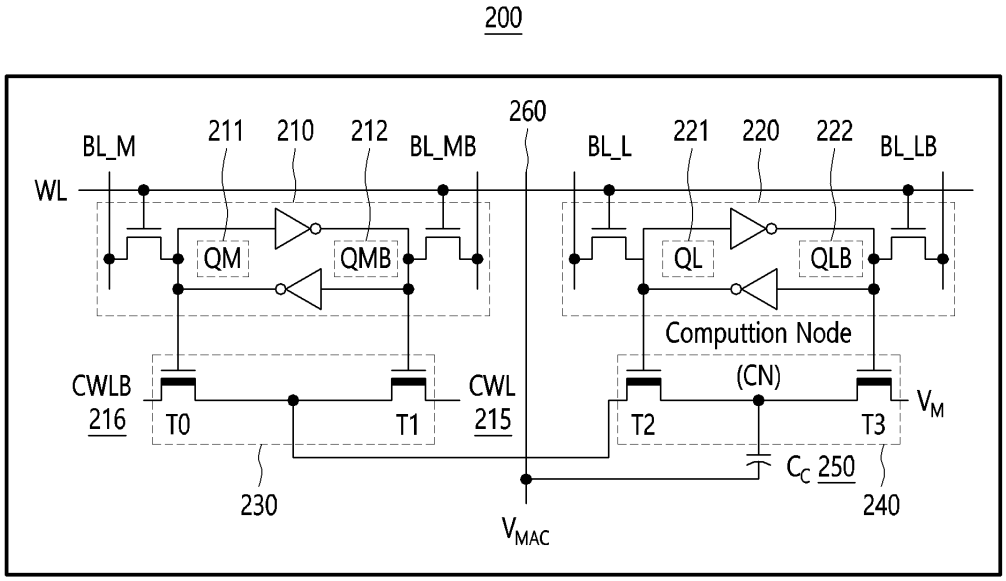
[0192] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

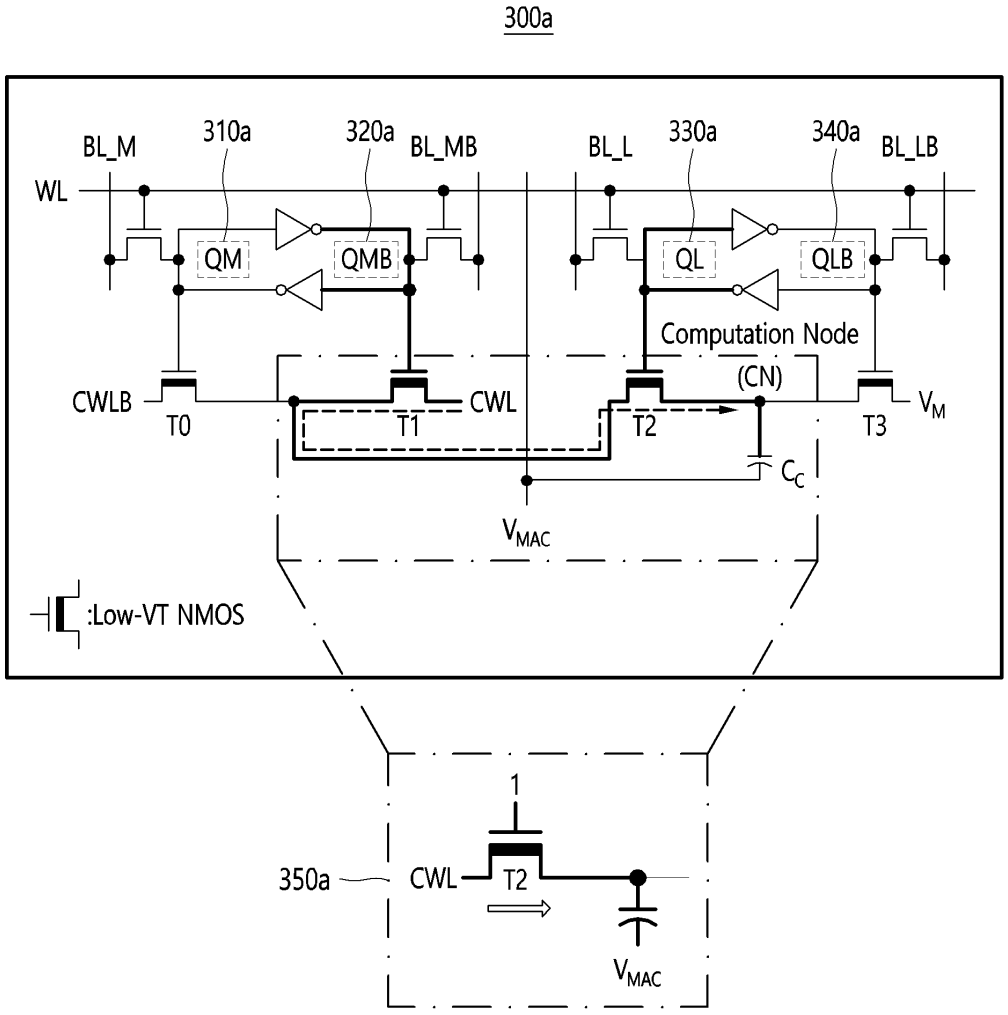
도면1


96 Input Activations
TOP ctivation Driver
Activation SW Network
TBA #0
Activation SW Network
TBA #1
Activation SW Network
TBA #2
Activation SW Network
TBA #3
110
Bank Controller
Ternary Bitcell Bank (TBB) 120
Weight/Data
Read/Write Interface
Analog SW Network
VMAC
130
x3
Charge-based Psum Adder
(CPA) 140
VPUSM
x4

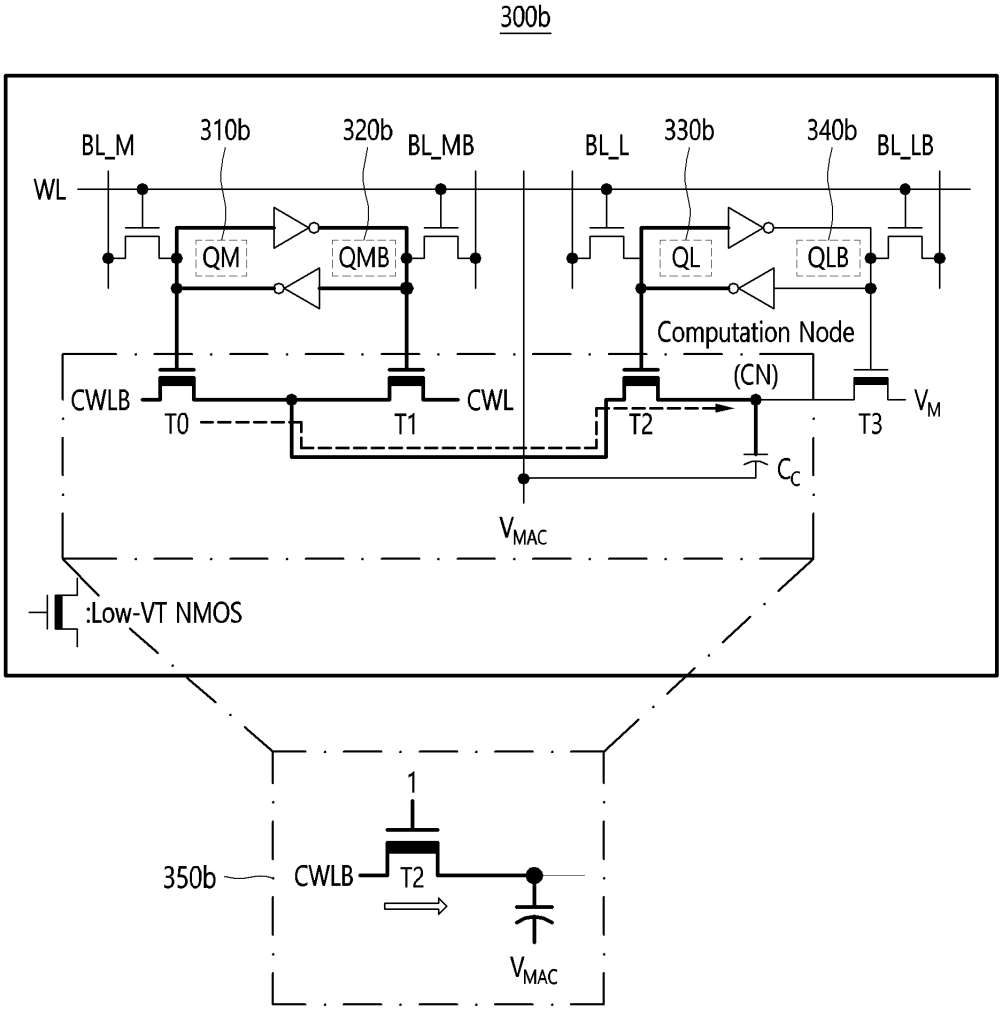
도면2


200
BL_M
211
210
212
BL_MB
260
BL_L
221
220
222
BL_LB
WL
QM
QMB
QL
QLB
Computtion Node
CWLB
216
T0
T1
CWL
215
(CN)
T2
T3
V_M
230
C_C 250
240
V_{MAC}

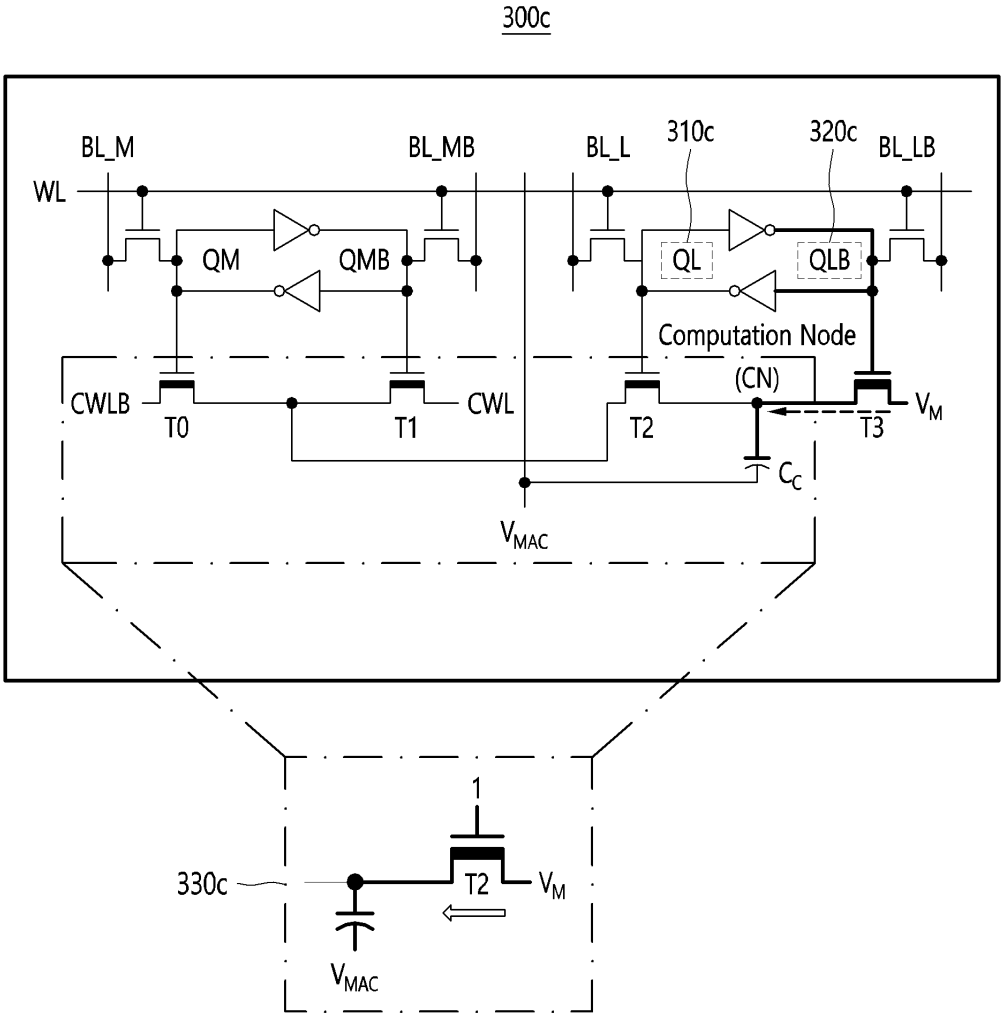
도면3a


300a
BL_M
310a
320a
BL_MB
BL_L
330a
340a
BL_LB
WL
QM
QMB
QL
QLB
Computation Node
CWLB
T0
T1
CWL
(CN)
T2
T3
V_M
C_C
V_{MAC}
:Low-VT NMOS
1
350a
CWL
T2
V_{MAC}

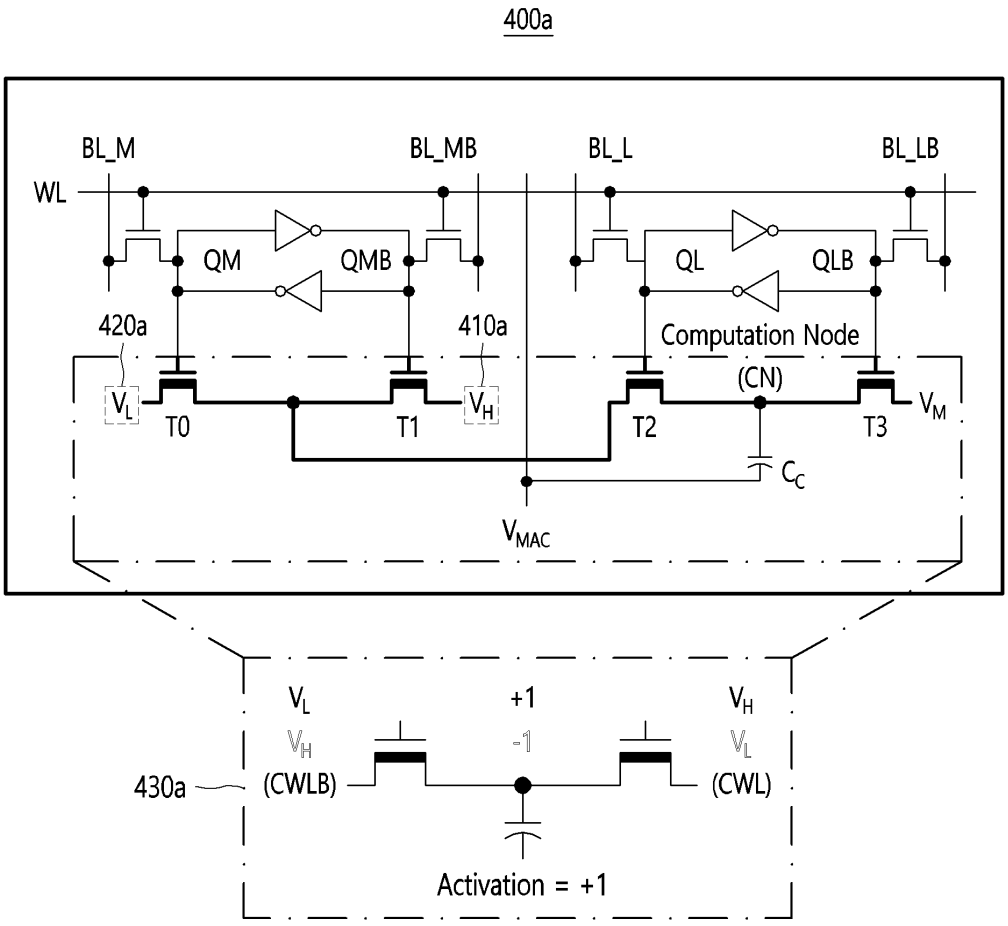
도면3b


300b
BL_M
310b
320b
BL_MB
BL_L
330b
340b
BL_LB
WL
QM
QMB
QL
QLB
Computation Node
(CN)
CWLB
CWL
T0
T1
T2
T3
V_M
C_C
V_{MAC}
:Low-VT NMOS
1
350b
CWLB
T2
V_{MAC}

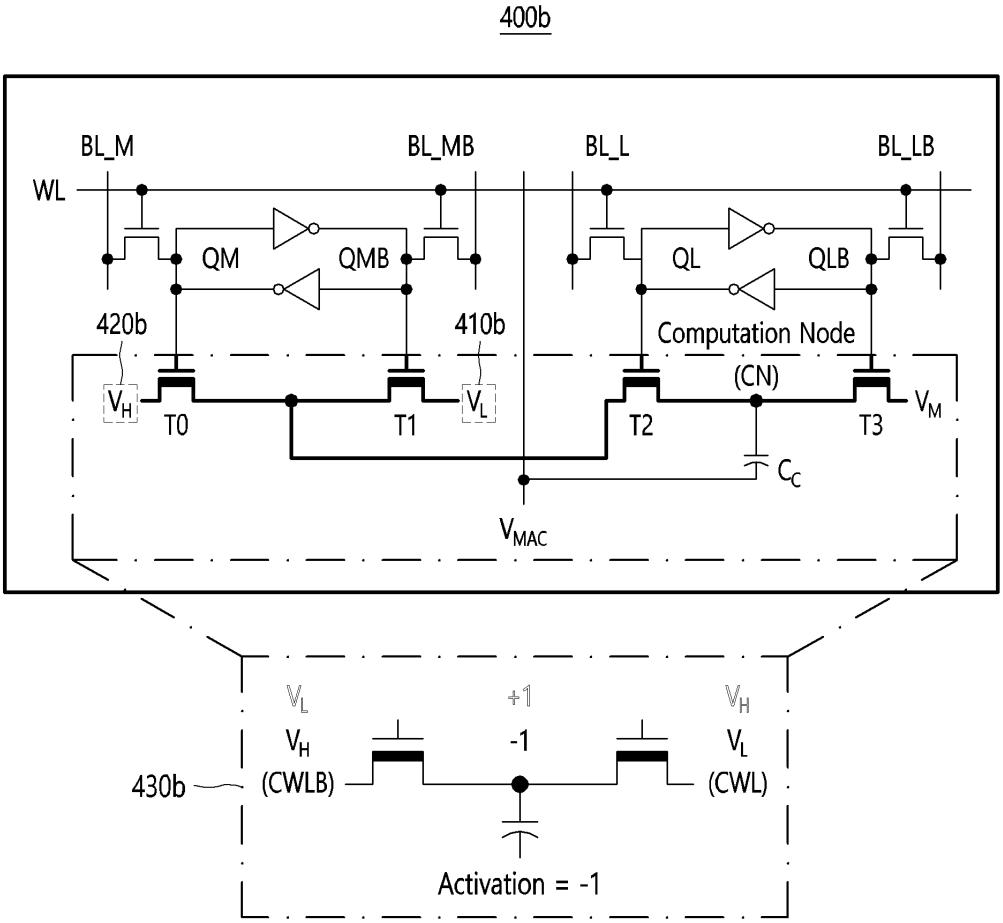
도면3c


300c
BL_M
BL_MB
BL_L
310c
320c
BL_LB
WL
QM
QMB
QL
QLB
Computation Node
(CN)
CWLB
T0
T1
CWL
T2
T3
V_M
C_C
V_{MAC}
1
330c
T2
V_M
V_{MAC}

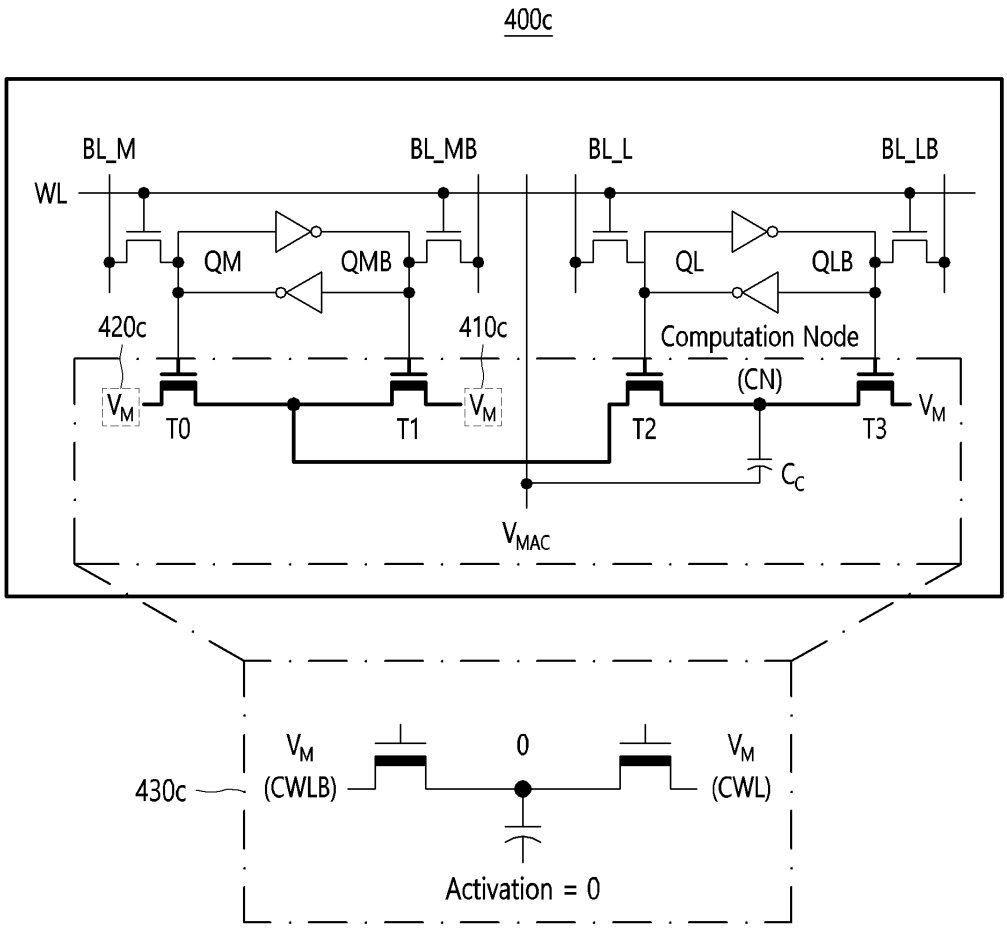
도면4a


400a
BL_M
BL_MB
BL_L
BL_LB
WL
QM
QMB
QL
QLB
420a
410a
Computation Node
(CN)
V_L
V_H
V_M
T0
T1
T2
T3
C_C
V_{MAC}
V_L
V_H
+1
V_H
V_L
-1
430a
(CWLB)
(CWL)
Activation = +1

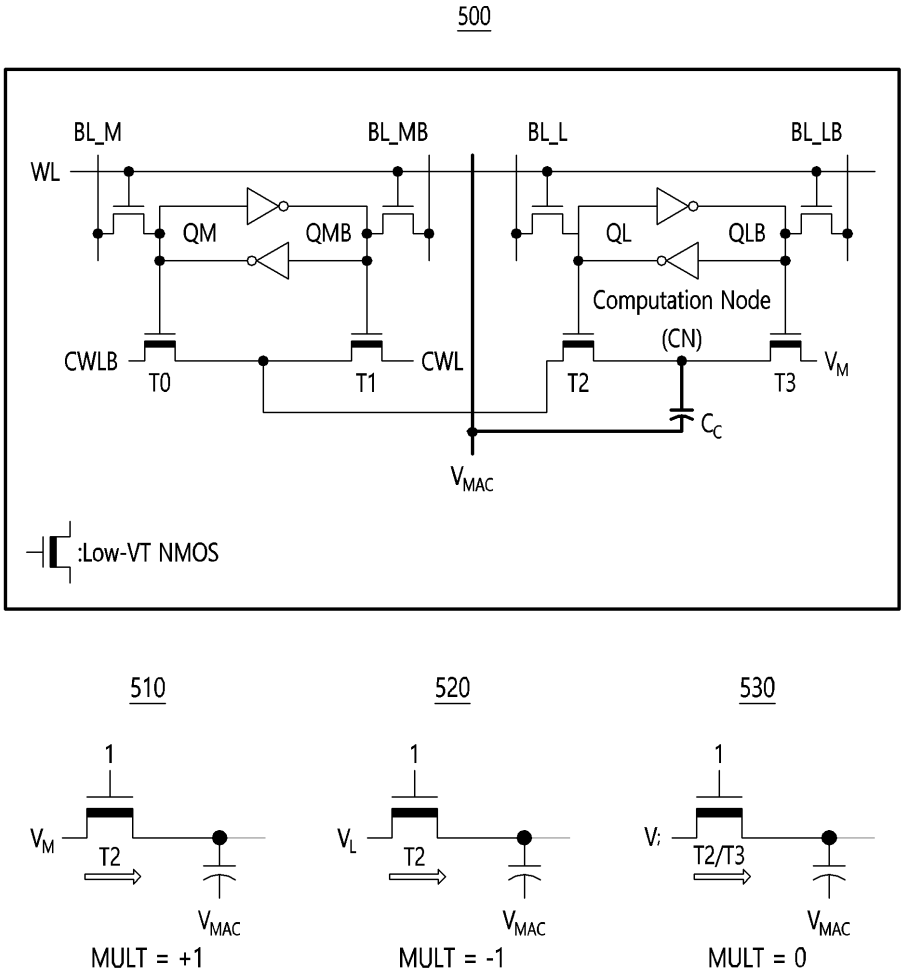
도면4b


400b
BL_M
BL_MB
BL_L
BL_LB
WL
QM
QMB
QL
QLB
420b
410b
Computation Node
(CN)
V_H
V_L
V_M
T0
T1
T2
T3
C_C
V_{MAC}
V_L
+1
V_H
V_H
-1
V_L
430b
(CWLB)
(CWL)
Activation = -1

도면4c

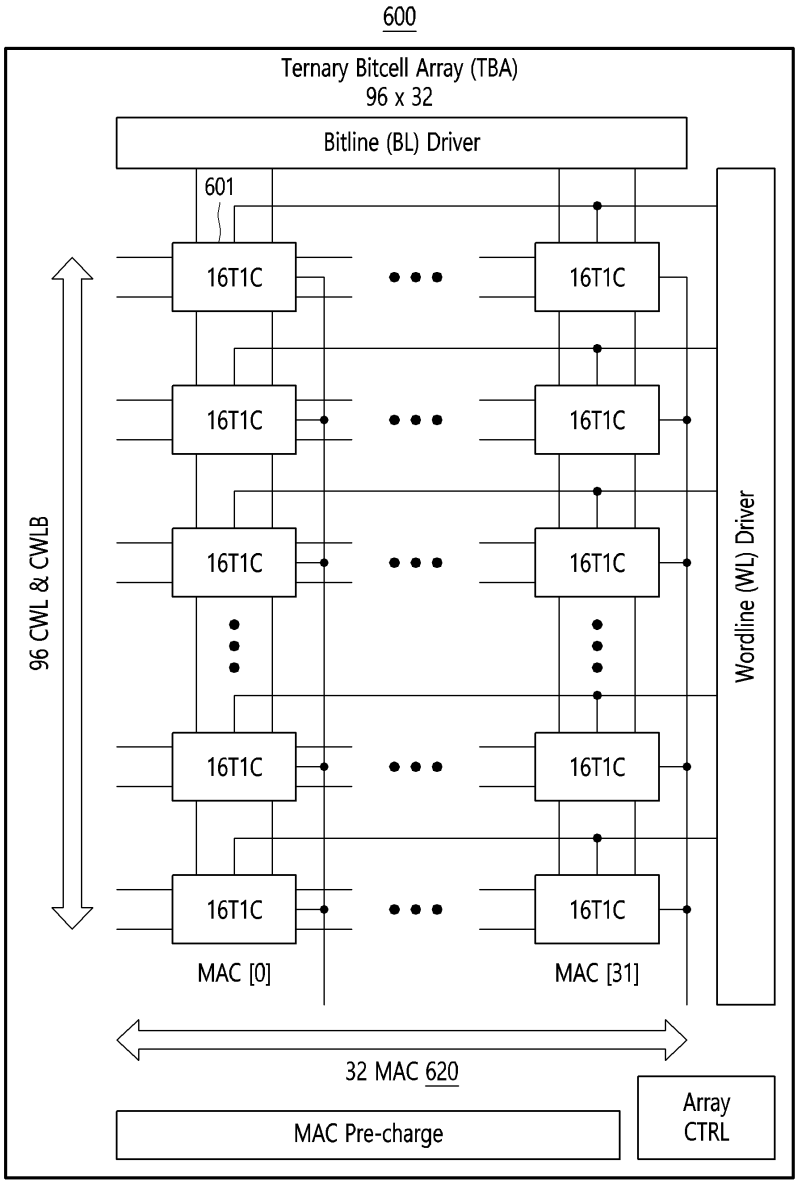

400c
BL_M
BL_MB
BL_L
BL_LB
WL
QM
QMB
QL
QLB
420c
410c
Computation Node
(CN)
VM
T0
T1
T2
T3
CC
VMAC
430c
VM
(CWLB)
0
VM
(CWL)
Activation = 0

도면5

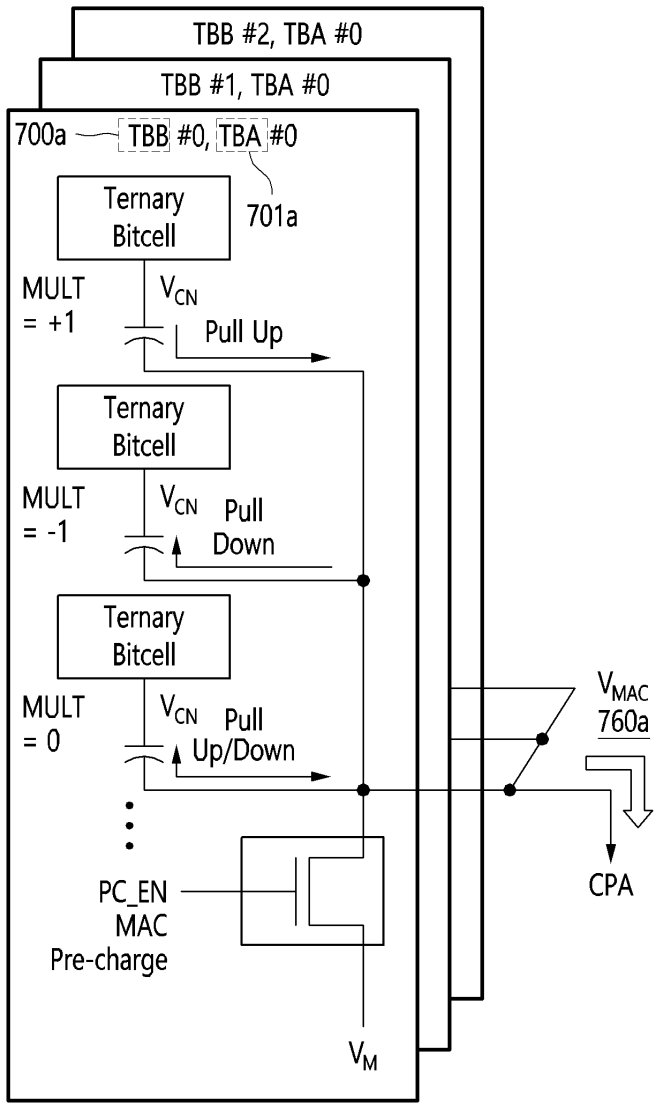

500
BL_M
BL_MB
BL_L
BL_LB
WL
QM
QMB
QL
QLB
Computation Node
(CN)
CWLB
T0
T1
CWL
T2
T3
V_M
C_C
V_{MAC}
:Low-VT NMOS

510
1
V_M
T2
V_{MAC}
MULT = +1
520
1
V_L
T2
V_{MAC}
MULT = -1
530
1
V_i
T2/T3
V_{MAC}
MULT = 0

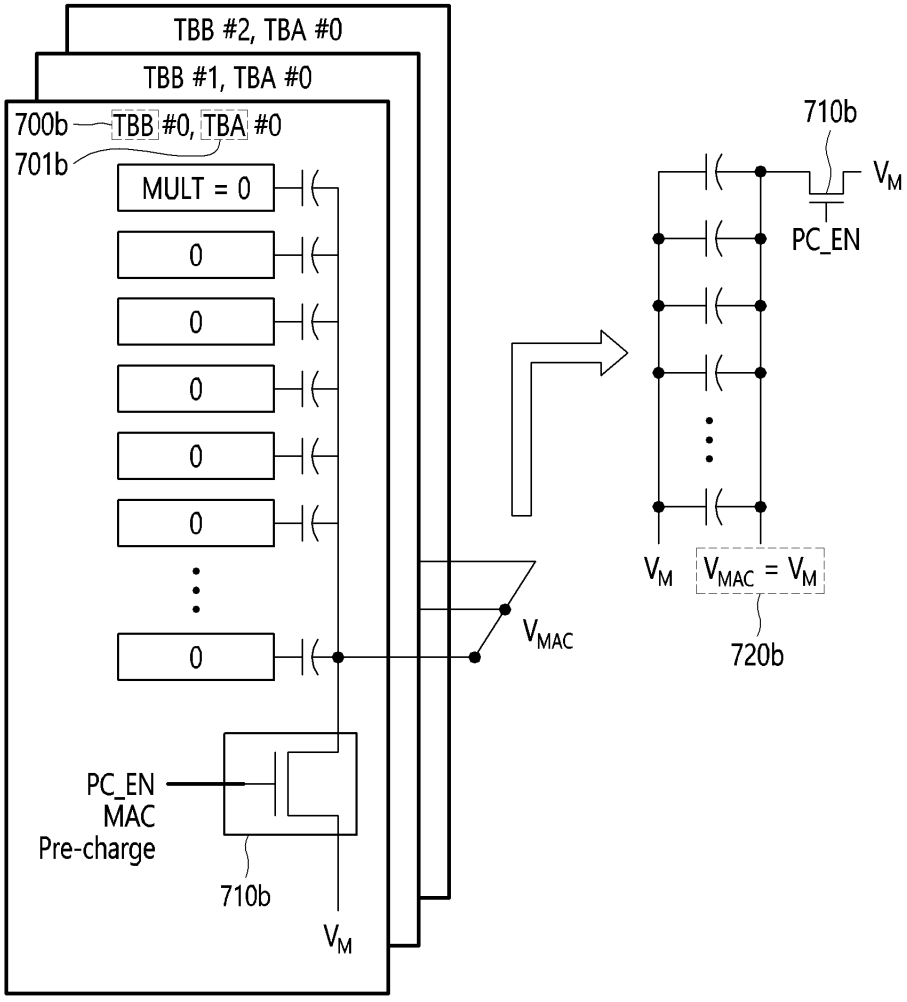
도면6


600
Ternary Bitcell Array (TBA)
96 x 32
Bitline (BL) Driver
601
16T1C
96 CWL & CWLB
Wordline (WL) Driver
MAC [0]
MAC [31]
32 MAC 620
MAC Pre-charge
Array CTRL

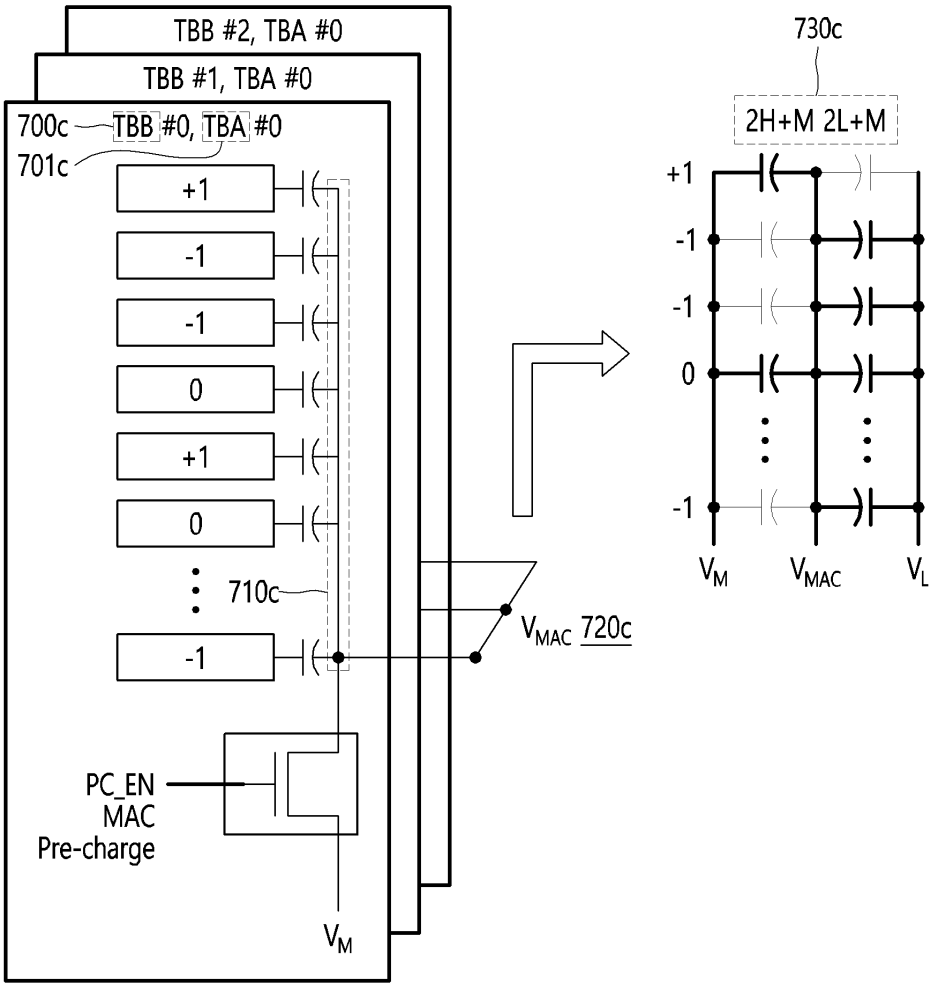
도면7a


TBB #2, TBA #0
TBB #1, TBA #0
700a
TBB #0, TBA #0
701a
Ternary Bitcell
MULT = +1
V_{CN}
Pull Up
Ternary Bitcell
MULT = -1
V_{CN}
Pull Down
Ternary Bitcell
MULT = 0
V_{CN}
Pull Up/Down
PC_EN
MAC Pre-charge
V_M
V_{MAC}
760a
CPA

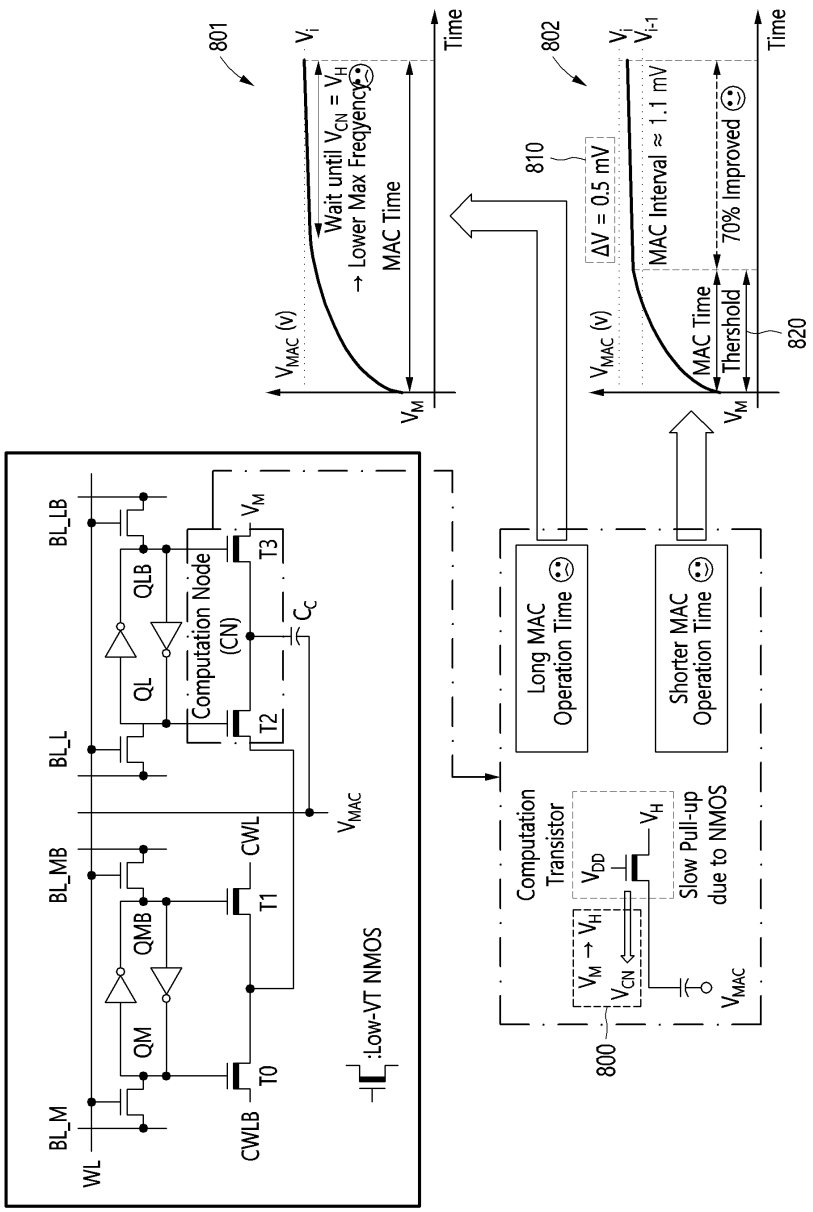
도면7b


TBB #2, TBA #0
TBB #1, TBA #0
700b
TBB #0, TBA #0
701b
MULT = 0
0
0
0
0
0
0
PC_EN
MAC
Pre-charge
710b
VM
VMAC
710b
VM
PC_EN
VM
VMAC = VM
720b

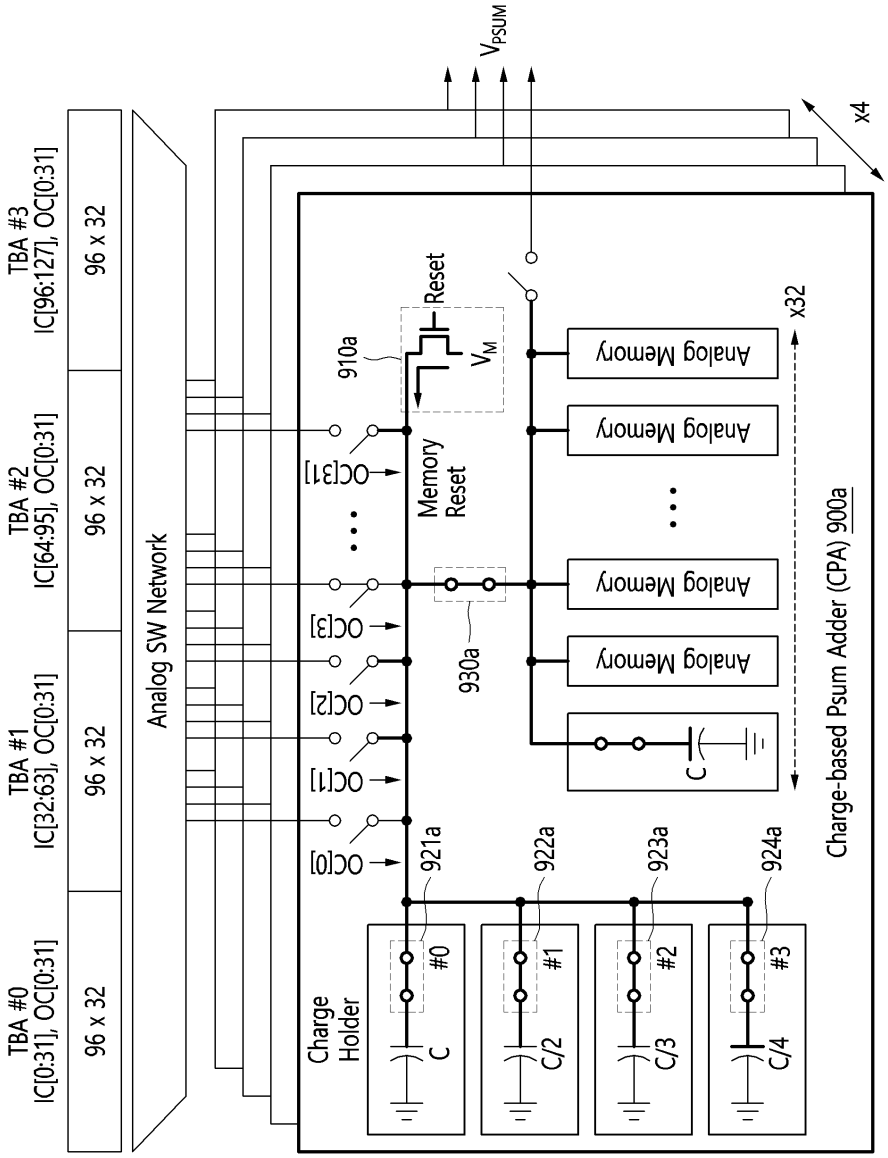
도면7c


TBB #2, TBA #0
TBB #1, TBA #0
700c
TBB #0, TBA #0
701c
+1
-1
-1
0
+1
0
710c
-1
PC_EN
MAC
Pre-charge
VM
VMAC
720c
730c
2H+M 2L+M
+1
-1
-1
0
-1
VM
VMAC
VL

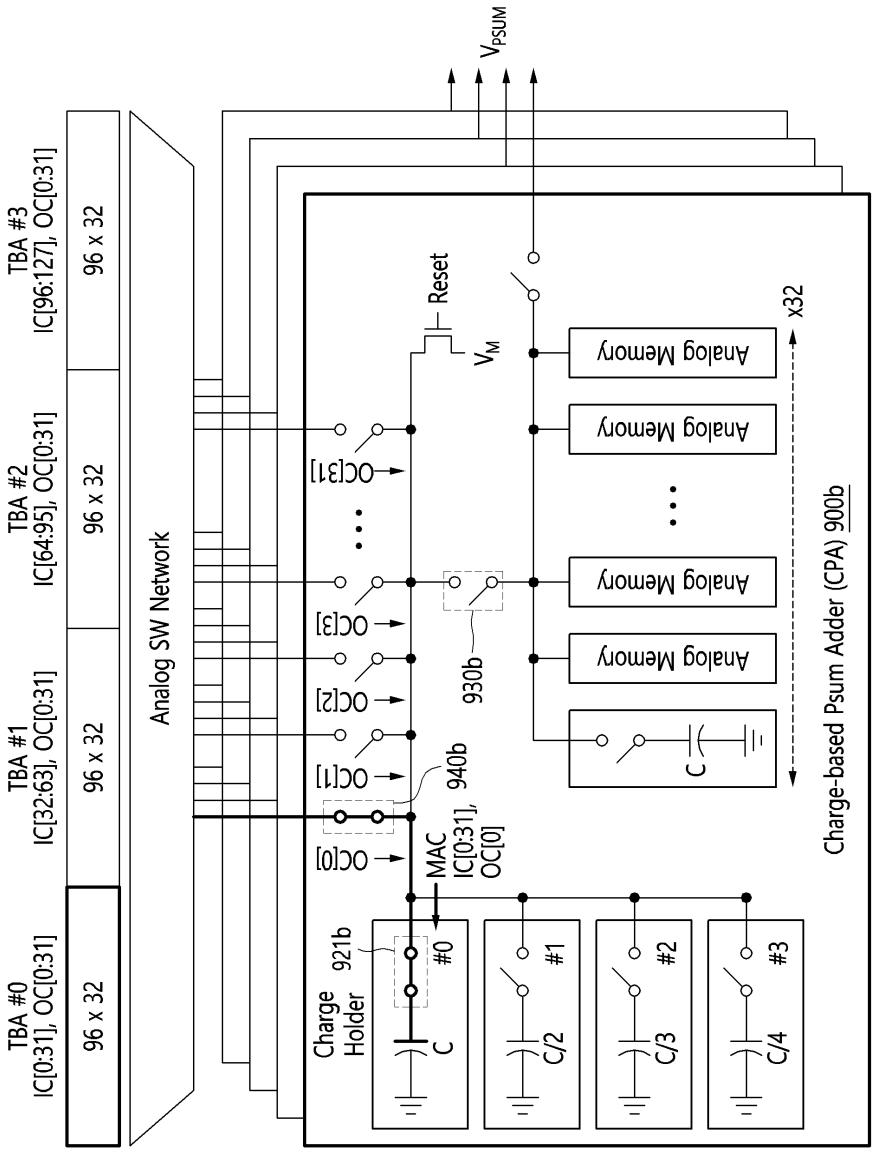
도면8


801
802
Time
Wait until $V_{CN} = V_H$
→ Lower Max Freqyency
MAC Time
V_{MAC} (v)
V_M
810
ΔV = 0.5 mV
MAC Interval ≈ 1.1 mV
70% Improved
Thershold
820
Long MAC Operation Time
Shorter MAC Operation Time
Computation Transistor
V_{DD}
V_H
$V_M \rightarrow V_H$
V_{CN}
Slow Pull-up due to NMOS
800
BL_LB
BL_L
BL_MB
BL_M
WL
QLB
QL
QMB
QM
Computation Node (CN)
T3
T2
T1
T0
CWL
CWLB
C_C
:Low-VT NMOS

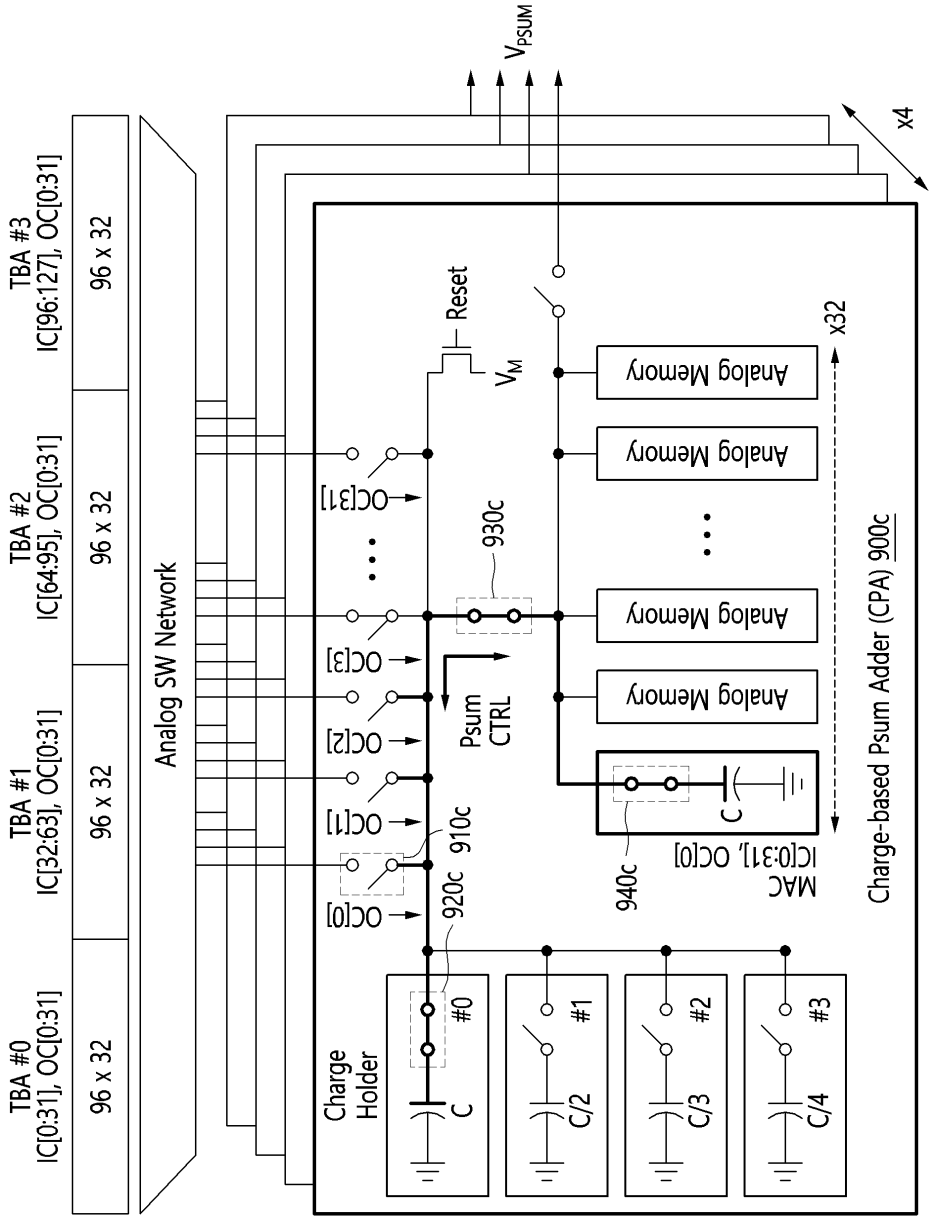
도면9a


TBA #0
IC[0:31], OC[0:31]
96 x 32
TBA #1
IC[32:63], OC[0:31]
96 x 32
TBA #2
IC[64:95], OC[0:31]
96 x 32
TBA #3
IC[96:127], OC[0:31]
96 x 32
Analog SW Network
V_{PSUM}
x4
Reset
910a
V_M
Memory Reset
930a
OC[31]
OC[3]
OC[2]
OC[1]
OC[0]
Analog Memory
Analog Memory
Analog Memory
Analog Memory
C
x32
Charge-based Psum Adder (CPA) 900a
921a
922a
923a
924a
Charge Holder
#0
#1
#2
#3
C
C/2
C/3
C/4

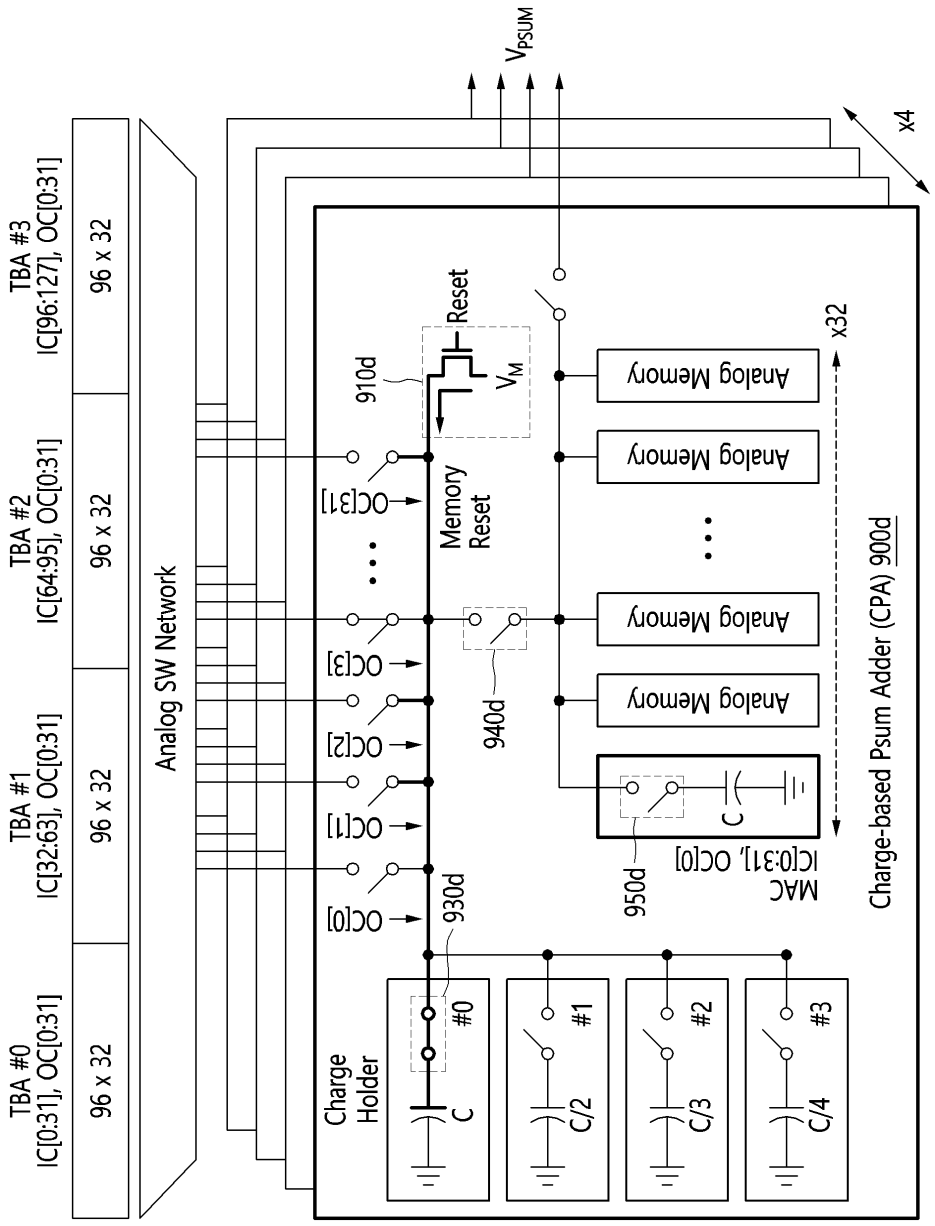
도면9b


TBA #3
IC[96:127], OC[0:31]
96 x 32
TBA #2
IC[64:95], OC[0:31]
96 x 32
TBA #1
IC[32:63], OC[0:31]
96 x 32
TBA #0
IC[0:31], OC[0:31]
96 x 32
Analog SW Network
V_{PSUM}
Reset
V_M
x32
Analog Memory
Analog Memory
Analog Memory
Analog Memory
C
930b
OC[31]
OC[3]
OC[2]
OC[1]
OC[0]
940b
MAC
IC[0:31],
OC[0]
Charge-based Psum Adder (CPA) 900b
Charge
Holder
921b
C
#0
C/2
#1
C/3
#2
C/4
#3

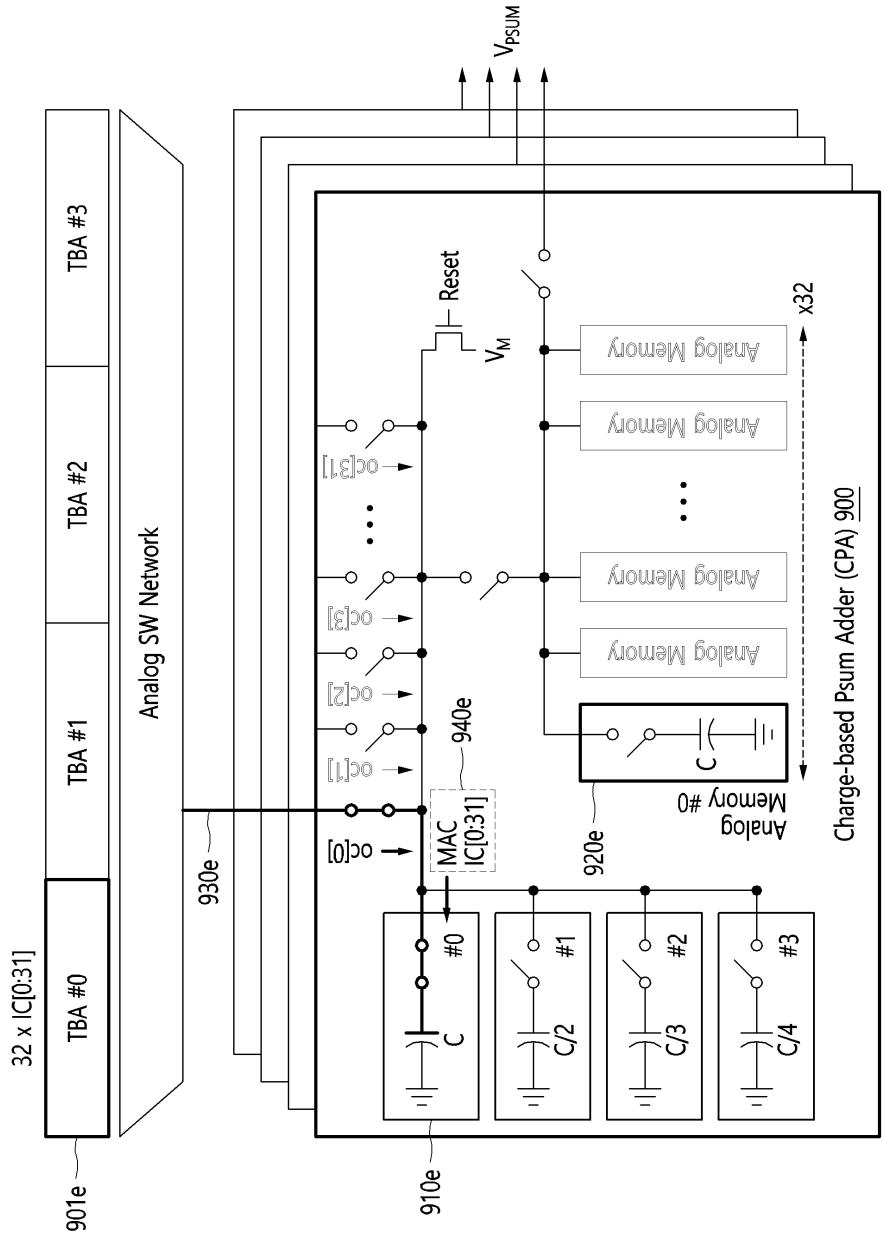
도면9c


TBA #0
IC[0:31], OC[0:31]
96 x 32
TBA #1
IC[32:63], OC[0:31]
96 x 32
TBA #2
IC[64:95], OC[0:31]
96 x 32
TBA #3
IC[96:127], OC[0:31]
96 x 32
Analog SW Network
V_{PSUM}
x4
Reset
V_M
x32
Analog Memory
Analog Memory
Analog Memory
Analog Memory
930c
Psum CTRL
OC[31]
OC[3]
OC[2]
OC[1]
OC[0]
910c
920c
940c
C
MAC IC[0:31], OC[0]
Charge-based Psum Adder (CPA) 900c
Charge Holder
#0
#1
#2
#3
C
C/2
C/3
C/4

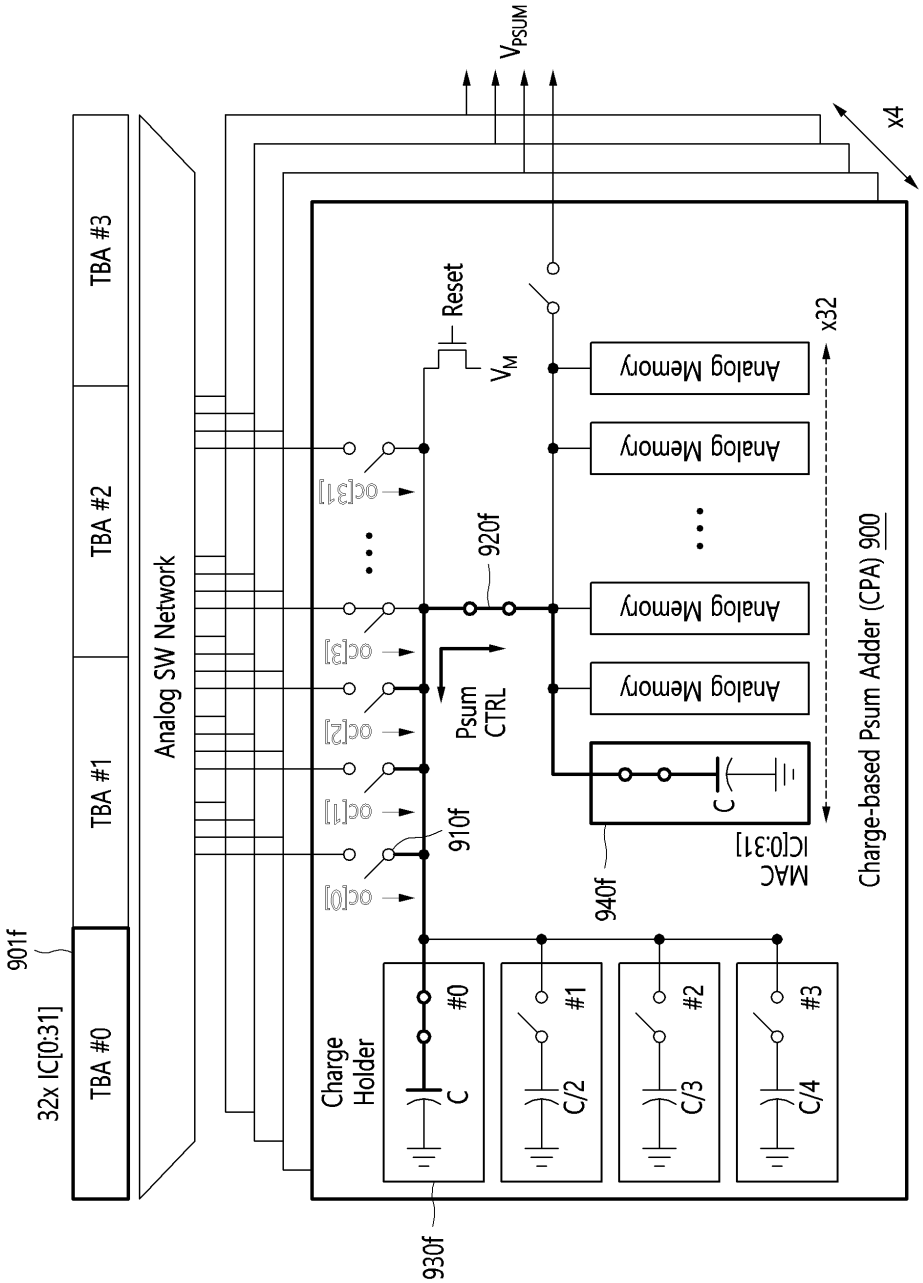
도면9d


TBA #0
IC[0:31], OC[0:31]
96 x 32
TBA #1
IC[32:63], OC[0:31]
96 x 32
TBA #2
IC[64:95], OC[0:31]
96 x 32
TBA #3
IC[96:127], OC[0:31]
96 x 32
Analog SW Network
V_{PSUM}
x4
Charge-based Psum Adder (CPA) 900d
Charge Holder
C
C/2
C/3
C/4
#0
#1
#2
#3
930d
OC[0]
OC[1]
OC[2]
OC[3]
OC[31]
Memory Reset
940d
910d
Reset
V_M
Analog Memory
x32
MAC
IC[0:31], OC[0]
950d

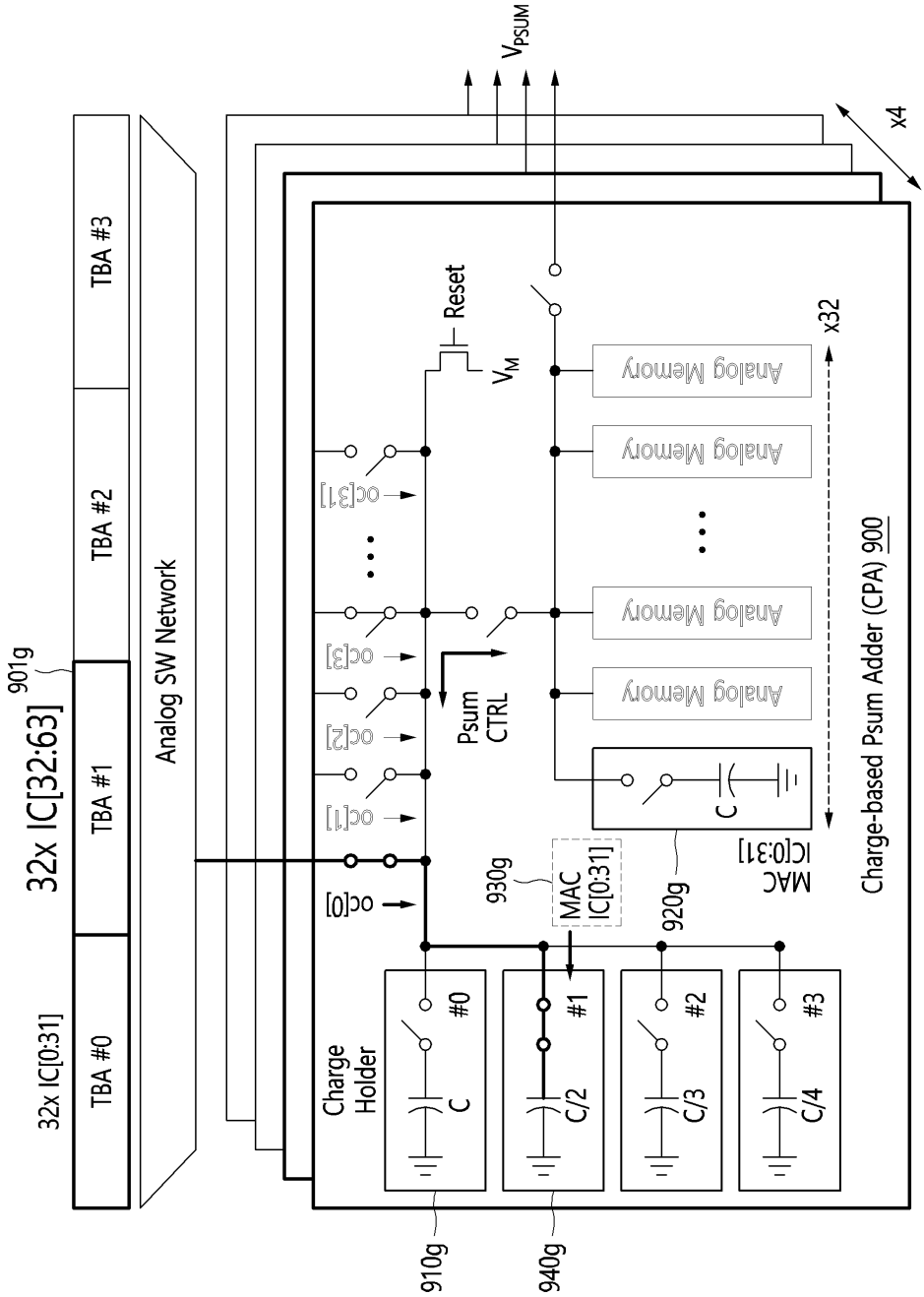
도면9e


32 x IC[0:31]
TBA #0
TBA #1
TBA #2
TBA #3
Analog SW Network
901e
930e
910e
920e
940e
V_{PSUM}
V_M
Reset
x32
Analog Memory
Analog Memory #0
Charge-based Psum Adder (CPA) 900
MAC IC[0:31]
IC[0]
IC[1]
IC[2]
IC[3]
IC[31]
C
C/2
C/3
C/4
#0
#1
#2
#3

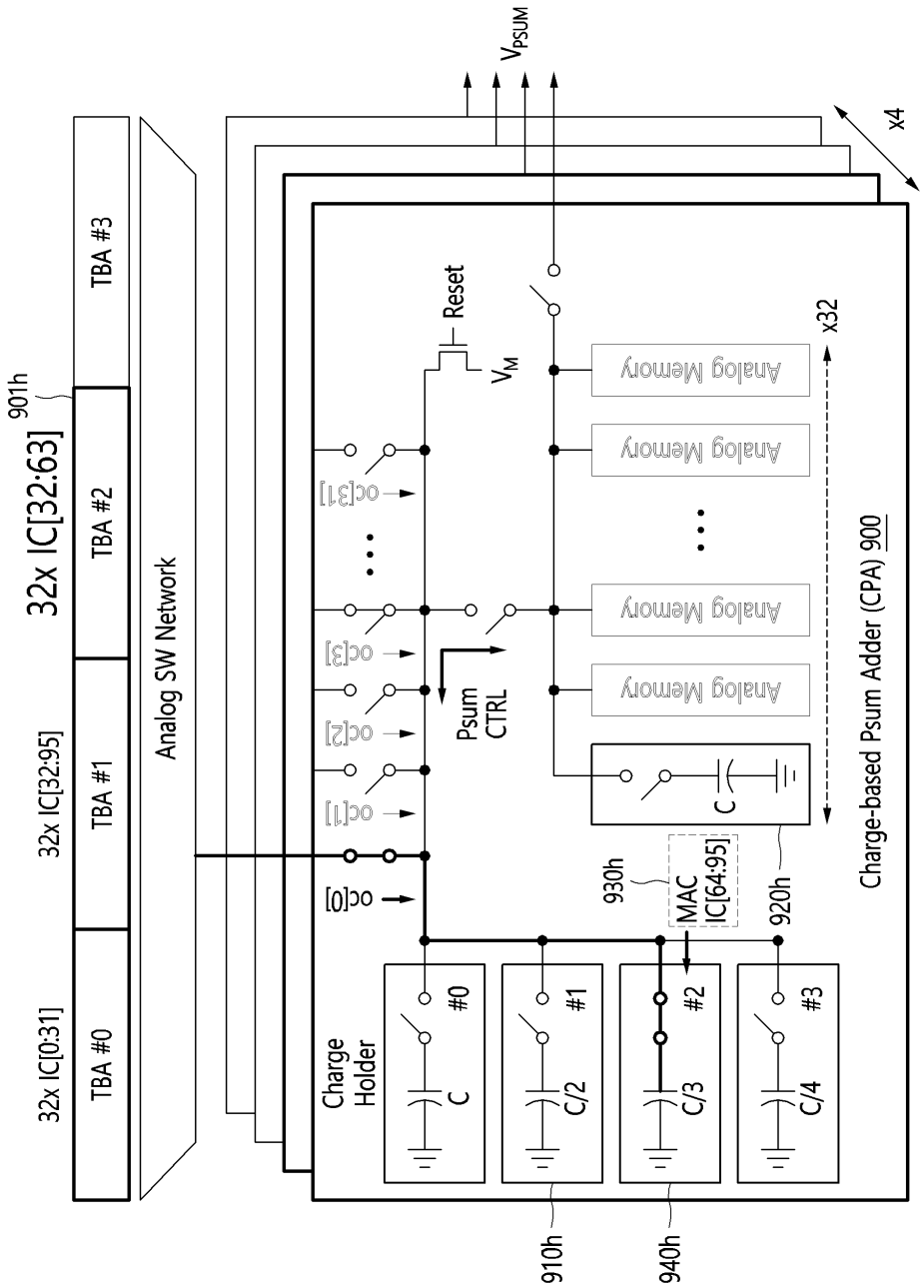
도면9f


V_{PSUM}
x4
TBA #3
TBA #2
TBA #1
TBA #0
901f
32x IC[0:31]
Analog SW Network
Reset
V_M
x32
Analog Memory
Analog Memory
Analog Memory
Analog Memory
Analog Memory
920f
Psum CTRL
IC[31]
IC[3]
IC[2]
IC[1]
IC[0]
910f
940f
MAC IC[0:31]
C
Charge-based Psum Adder (CPA) 900
Charge Holder
#0
#1
#2
#3
C
C/2
C/3
C/4
930f

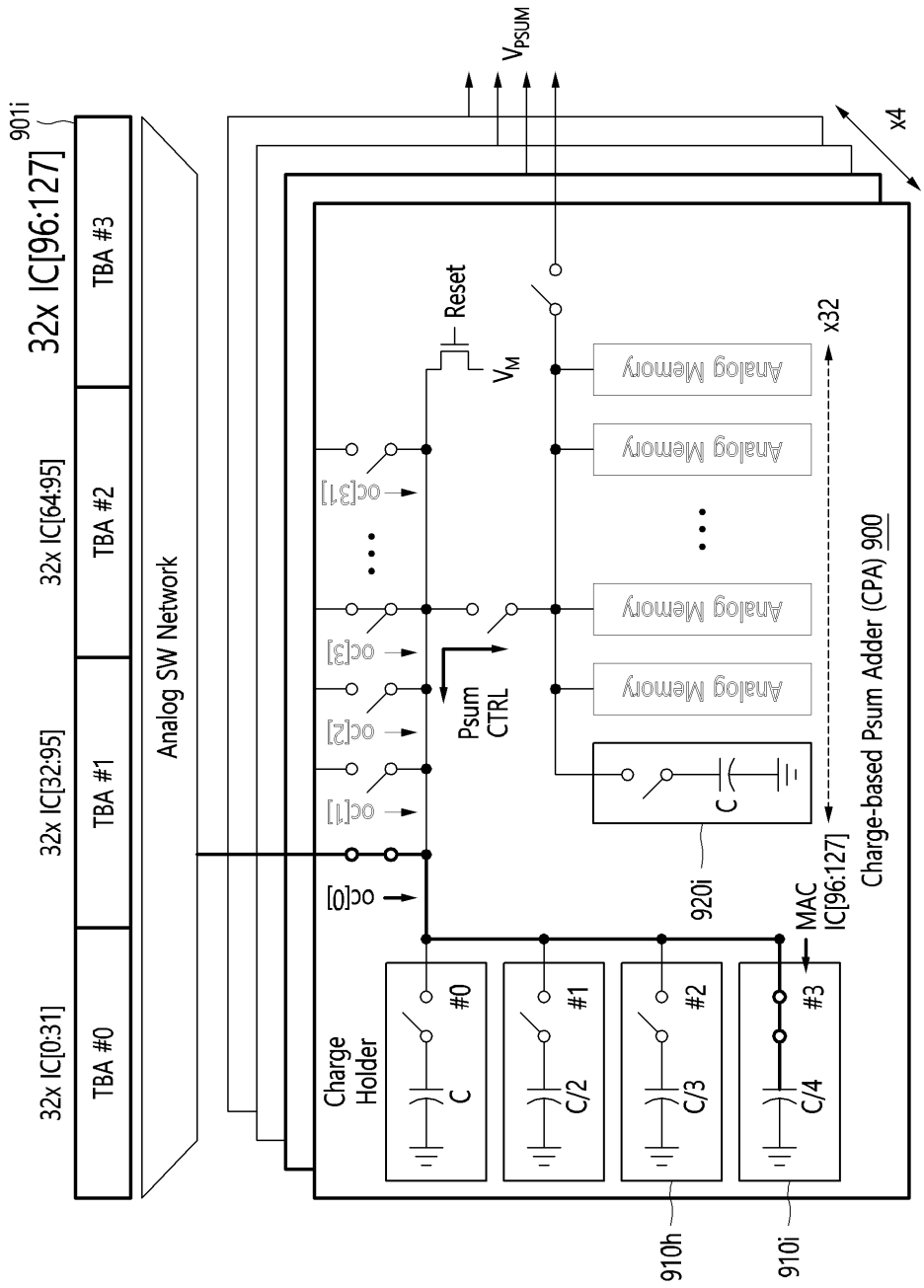
도면9g


32x IC[0:31]
32x IC[32:63]
901g
TBA #0
TBA #1
TBA #2
TBA #3
Analog SW Network
V_{PSUM}
x4
Reset
V_M
Analog Memory
x32
oc[31]
oc[3]
oc[2]
oc[1]
oc[0]
Psum CTRL
C
MAC IC[0:31]
930g
920g
Charge Holder
#0
#1
#2
#3
C
C/2
C/3
C/4
910g
940g
Charge-based Psum Adder (CPA) 900

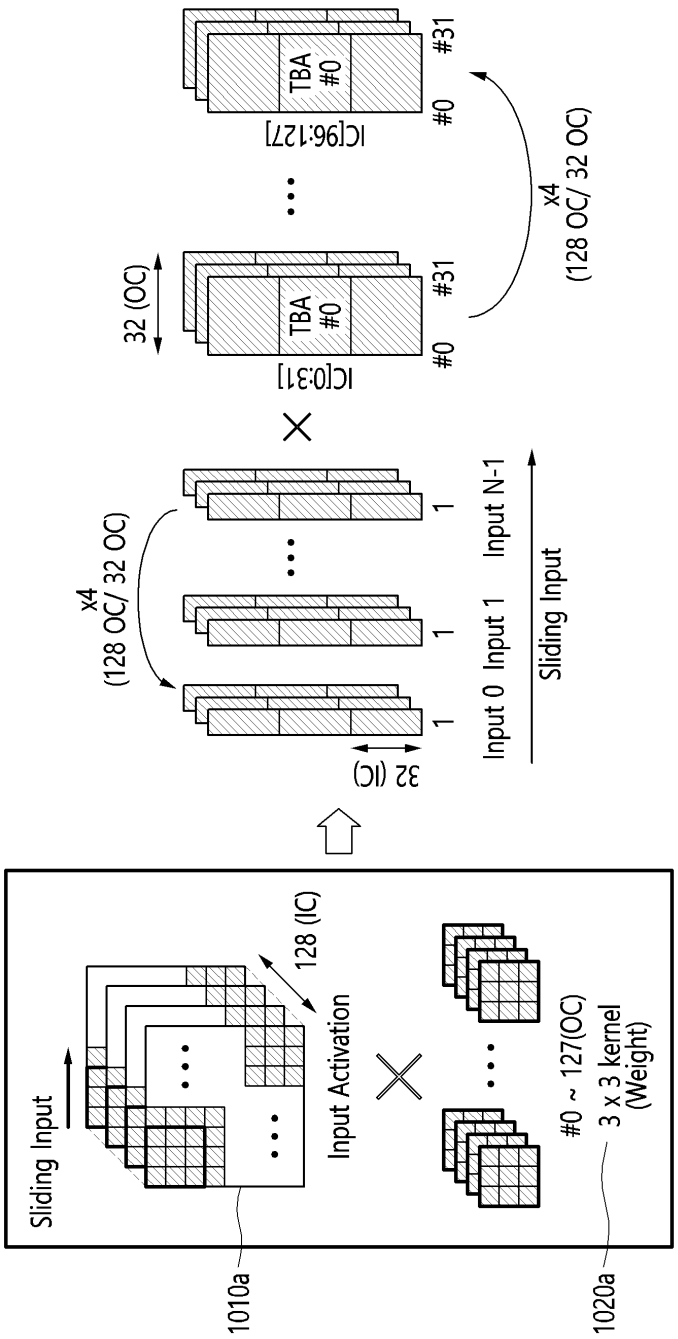
도면9h


V_{PSUM}
x4
901h
32x IC[32:63]
32x IC[32:95]
32x IC[0:31]
TBA #3
TBA #2
TBA #1
TBA #0
Analog SW Network
Reset
V_M
x32
Analog Memory
Analog Memory
Analog Memory
Analog Memory
oc[31]
oc[3]
oc[2]
oc[1]
oc[0]
Psum CTRL
C
930h
MAC IC[64:95]
920h
Charge-based Psum Adder (CPA) 900
Charge Holder
C #0
C/2 #1
C/3 #2
C/4 #3
910h
940h

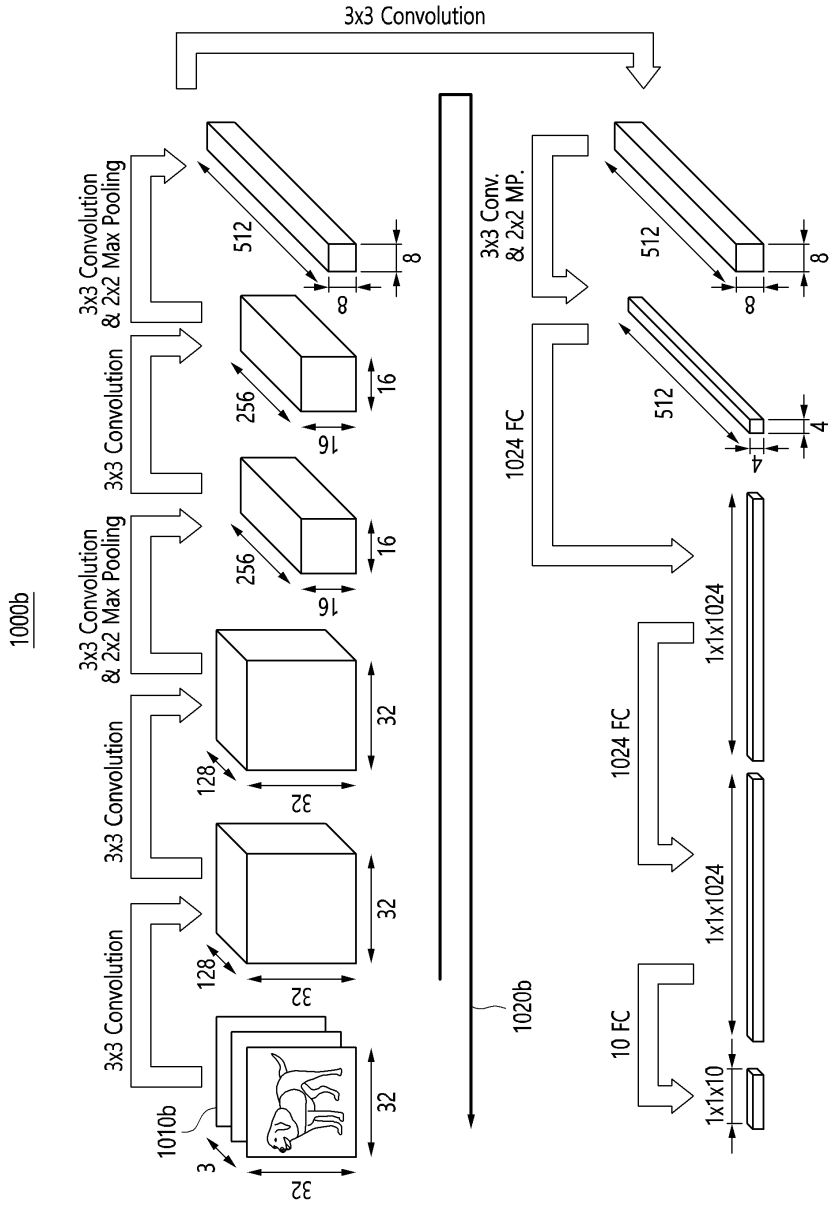
도면9i


901i
32x IC[96:127]
32x IC[64:95]
32x IC[32:95]
32x IC[0:31]
TBA #3
TBA #2
TBA #1
TBA #0
Analog SW Network
V_{PSUM}
x4
Reset
V_M
Analog Memory
x32
oc[31]
oc[3]
oc[2]
oc[1]
oc[0]
Psum CTRL
C
920i
MAC IC[96:127]
Charge-based Psum Adder (CPA) 900
Charge Holder
#0
#1
#2
#3
C
C/2
C/3
C/4
910h
910i

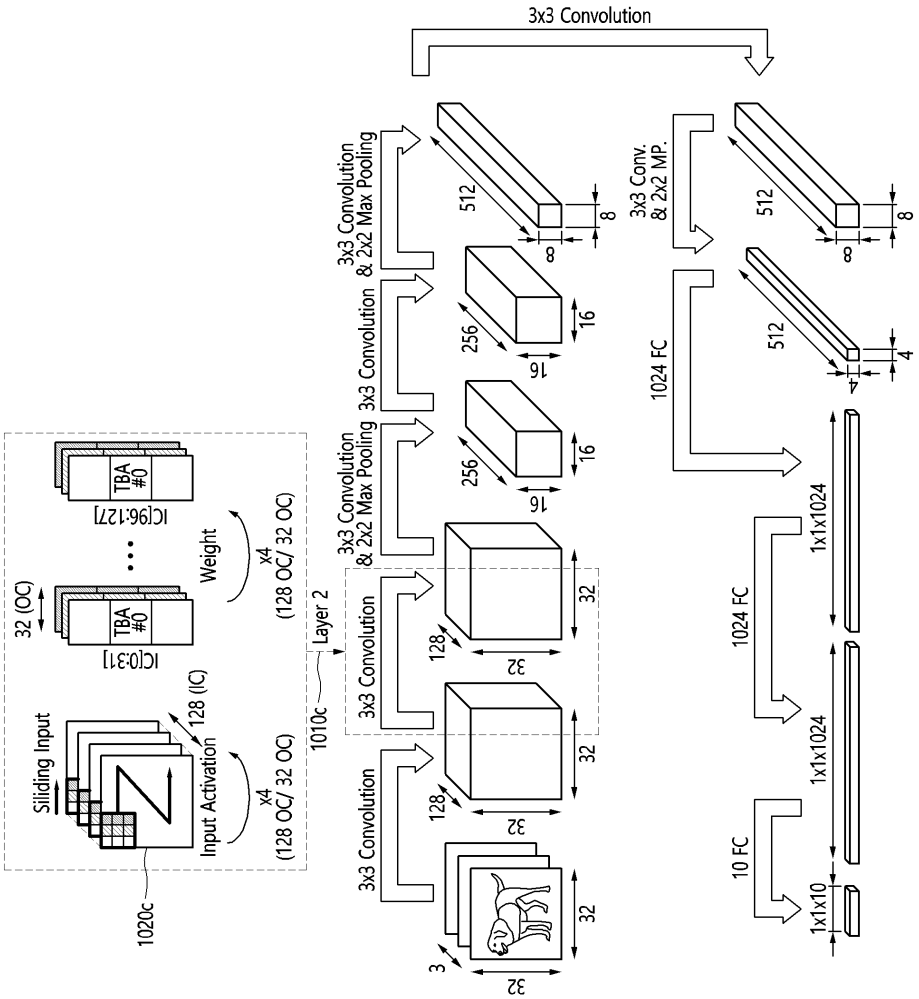
도면10a


Sliding Input
128 (IC)
Input Activation
#0 ~ 127(OC)
3 x 3 kernel
(Weight)
1010a
1020a
32 (IC)
Input 0
Input 1
Input N-1
Sliding Input
x4
(128 OC/ 32 OC)
32 (OC)
IC[0:31]
IC[96:127]
TBA
#0
#31

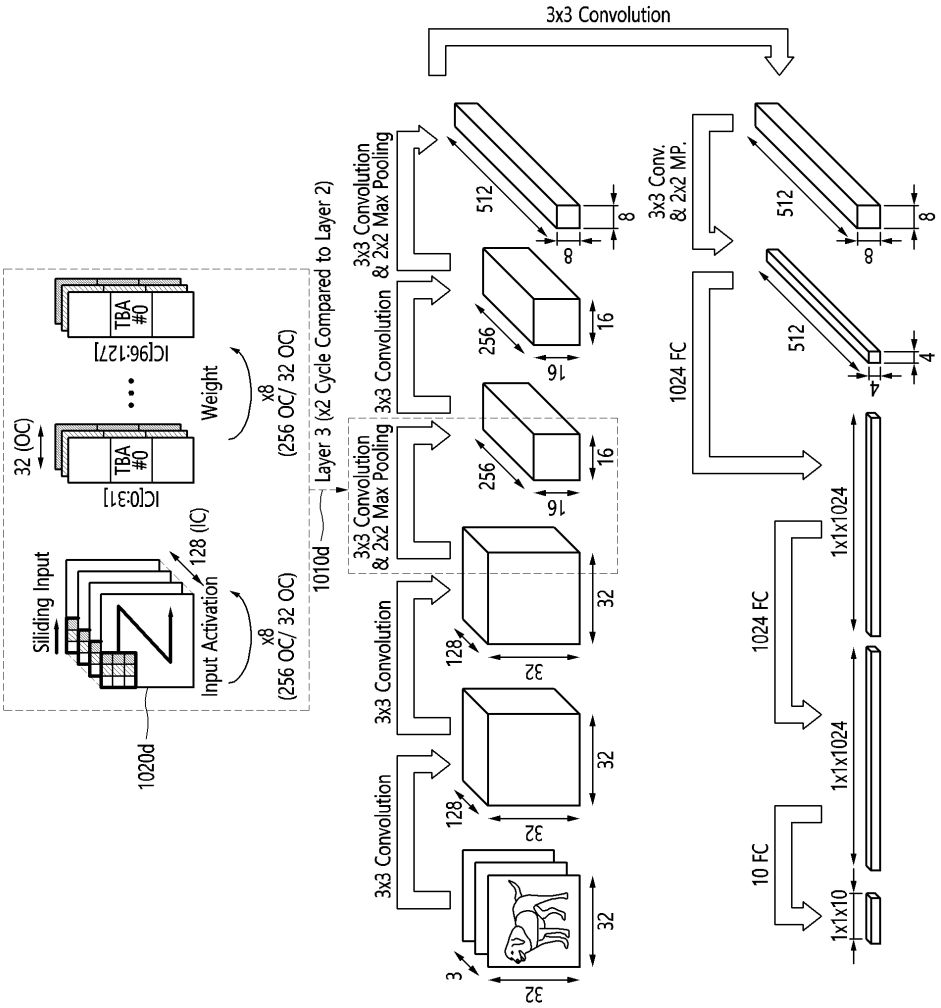
도면10b


1000b
3x3 Convolution
3x3 Convolution & 2x2 Max Pooling
3x3 Convolution
3x3 Convolution & 2x2 Max Pooling
3x3 Convolution
3x3 Conv. & 2x2 MP.
1024 FC
1024 FC
10 FC
1010b
1020b
32
3
128
32
32
128
32
32
256
16
16
256
16
8
8
512
512
8
8
512
4
1x1x1024
1x1x1024
1x1x10

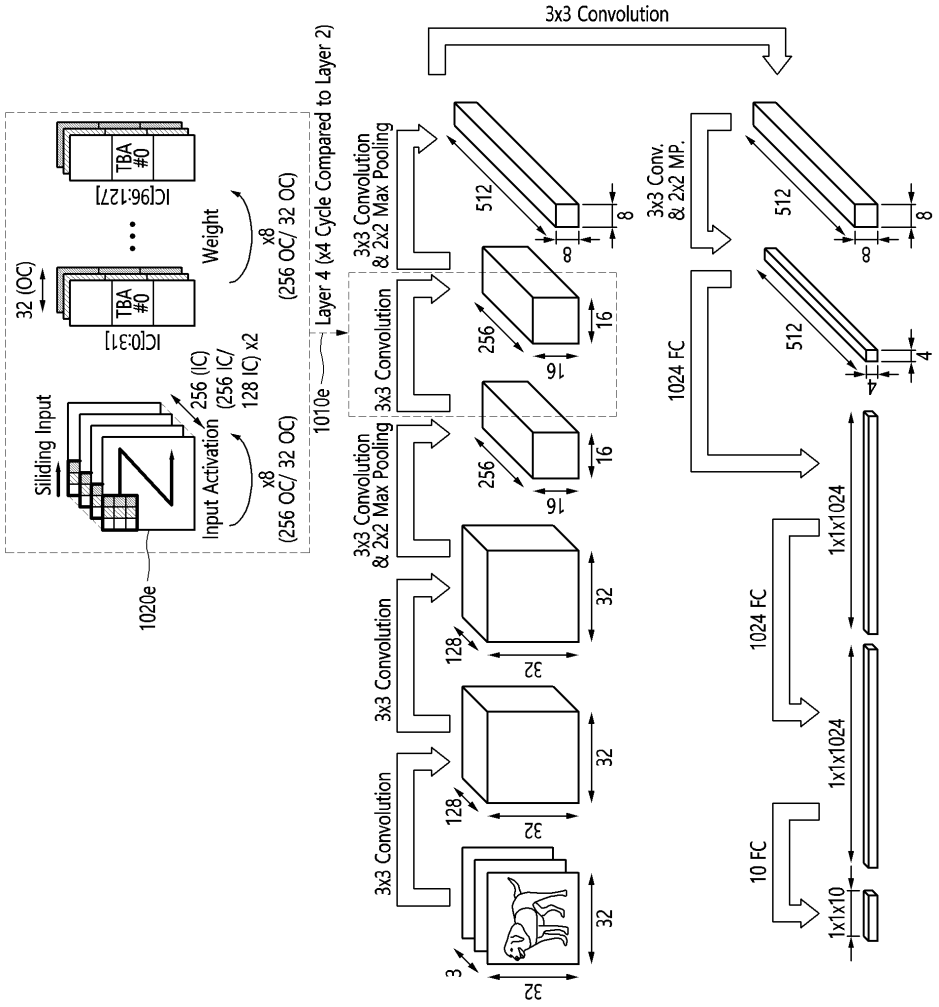
도면10c


3x3 Convolution
3x3 Convolution & 2x2 Max Pooling
3x3 Convolution
3x3 Convolution & 2x2 Max Pooling
3x3 Convolution
3x3 Conv. & 2x2 MP.
1024 FC
1024 FC
10 FC
Layer 2
1010c
1020c
Sliding Input
Input Activation
128 (IC)
32 (OC)
IC[0:31]
IC[96:127]
TBA #0
Weight
x4 (128 OC/ 32 OC)
x4 (128 OC/ 32 OC)

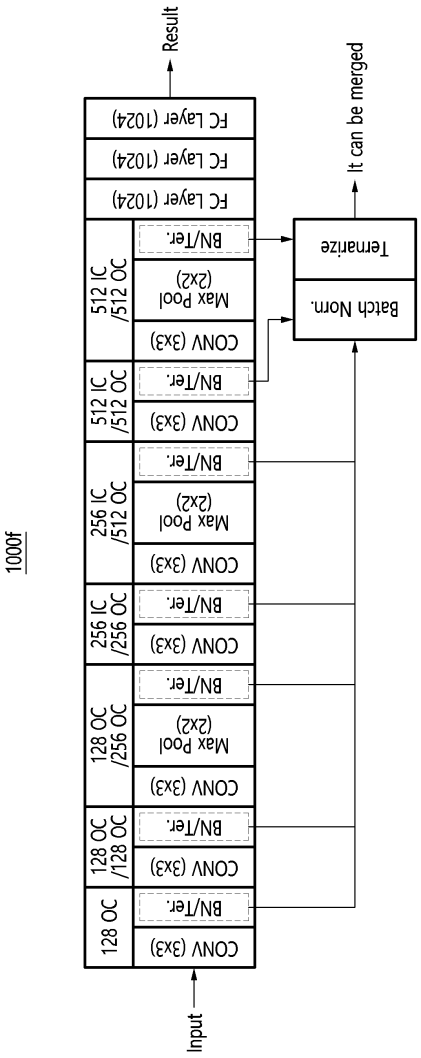
도면10d


3x3 Convolution
3x3 Convolution & 2x2 Max Pooling
3x3 Convolution
3x3 Convolution & 2x2 Max Pooling
3x3 Convolution
3x3 Convolution
3x3 Conv. & 2x2 MP.
1024 FC
1024 FC
10 FC
Layer 3 (x2 Cycle Compared to Layer 2)
1010d
1020d
Siliding Input
Input Activation
128 (IC)
32 (OC)
Weight
x8 (256 OC/ 32 OC)
x8 (256 OC/ 32 OC)
TBA #0
TBA #0
IC[0:31]
IC[96:127]
512
256
128
32
16
8
4
3
1x1x1024
1x1x1024
1x1x10

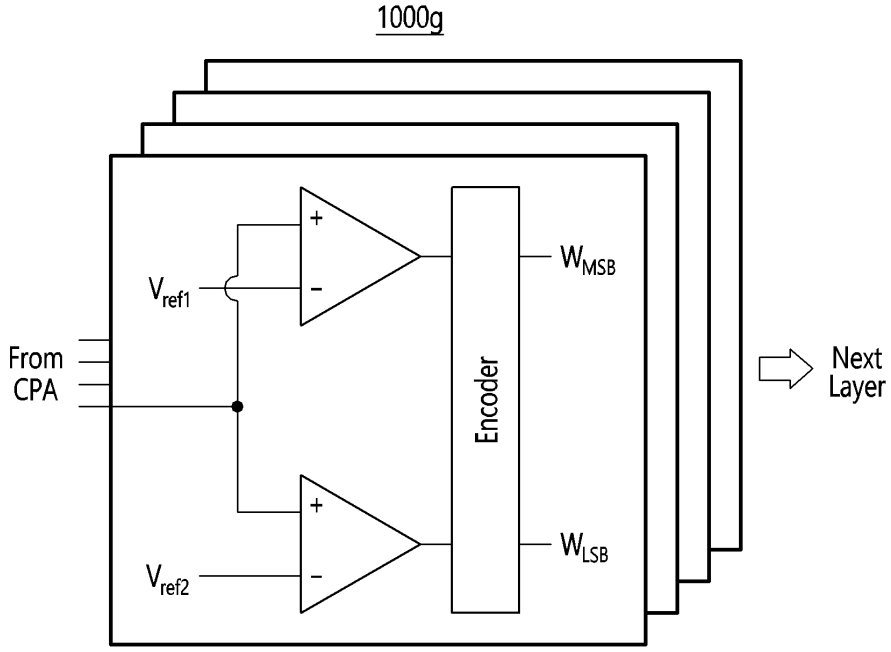
도면10e


3x3 Convolution
Layer 4 (x4 Cycle Compared to Layer 2)
1010e
1020e
Sliding Input
Input Activation
256 (IC)
(256 IC/
128 IC) x2
x8
(256 OC/ 32 OC)
32 (OC)
TBA
#0
IC[0:31]
IC[96:127]
Weight
3x3 Convolution
3x3 Convolution
& 2x2 Max Pooling
3x3 Conv.
& 2x2 MP.
1024 FC
10 FC
512
256
128
32
16
8
4
3
1x1x1024
1x1x10

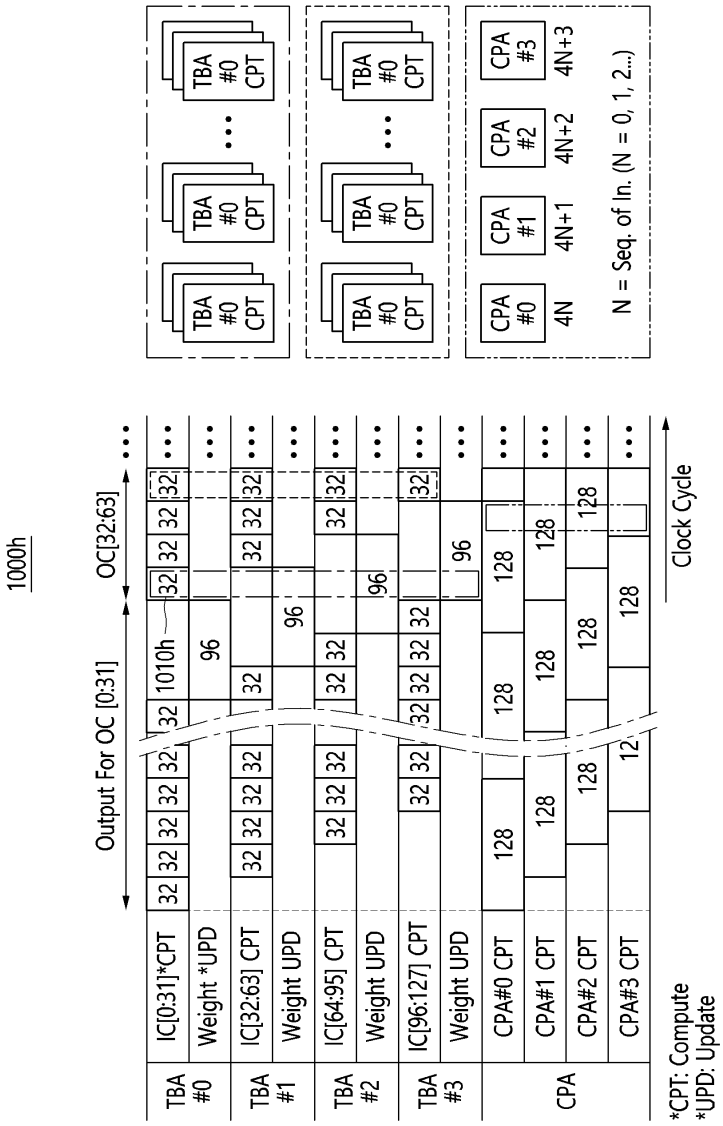
도면10f


1000f
Input
128 OC
CONV (3x3)
BN/Ter.
128 OC /128 OC
CONV (3x3)
BN/Ter.
128 OC /256 OC
CONV (3x3)
Max Pool (2x2)
BN/Ter.
256 IC /256 OC
CONV (3x3)
BN/Ter.
256 IC /512 OC
CONV (3x3)
Max Pool (2x2)
BN/Ter.
512 IC /512 OC
CONV (3x3)
BN/Ter.
512 IC /512 OC
CONV (3x3)
Max Pool (2x2)
BN/Ter.
FC Layer (1024)
FC Layer (1024)
FC Layer (1024)
Result
Batch Norm.
Ternarize
It can be merged

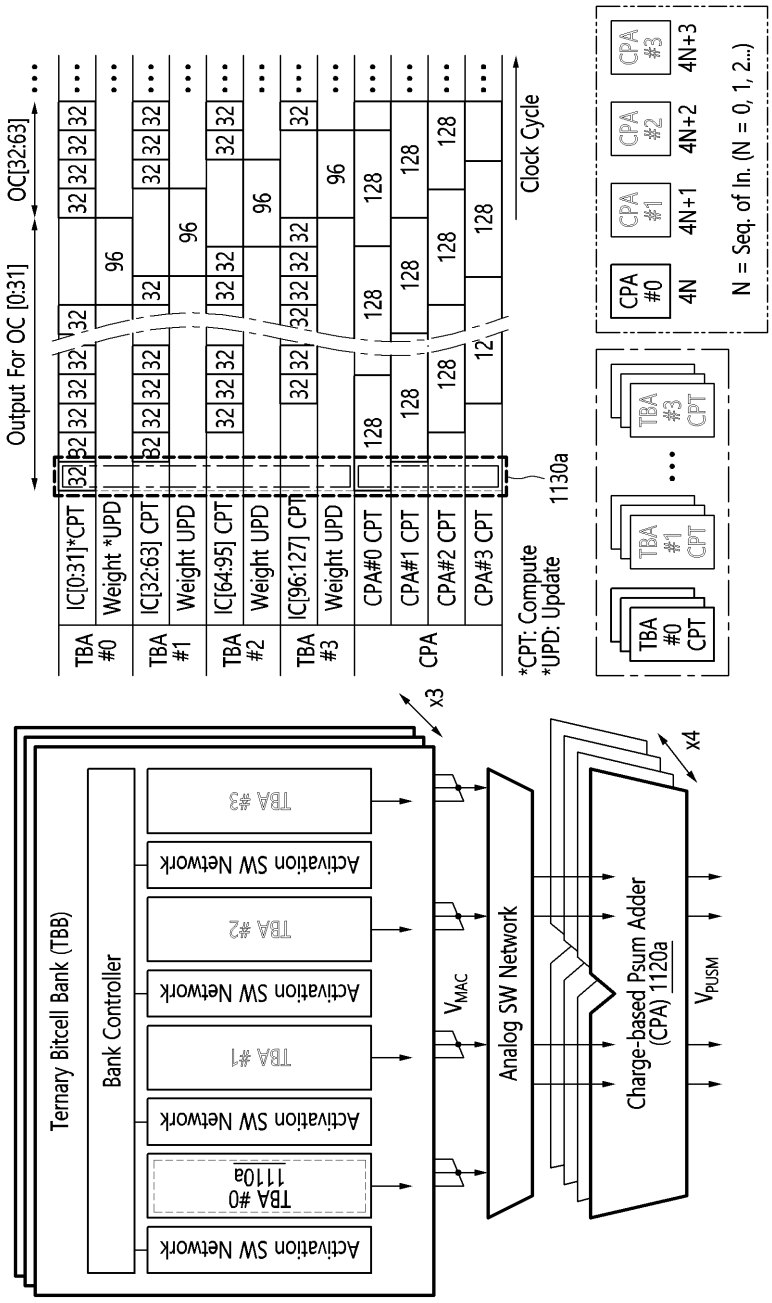
도면10g



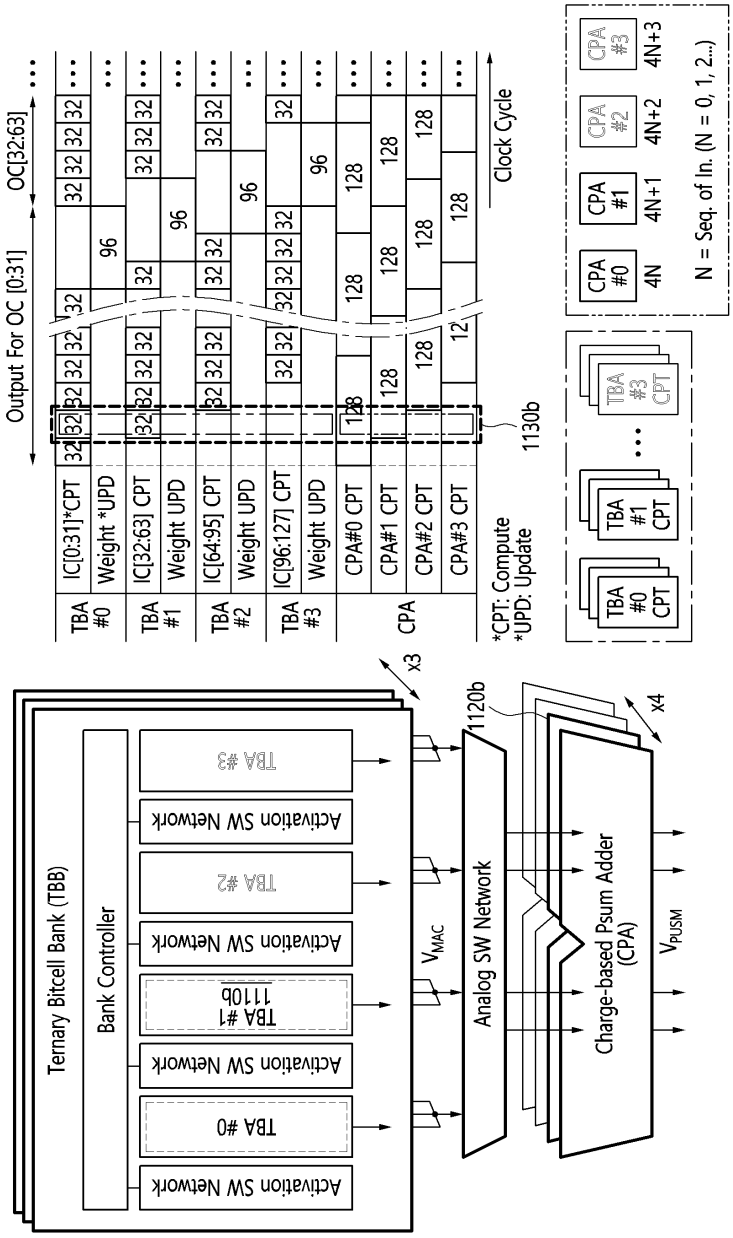
도면10h


1000h
Output For OC [0:31]
OC[32:63]
1010h
TBA #0
TBA #1
TBA #2
TBA #3
CPA
IC[0:31]*CPT
Weight *UPD
IC[32:63] CPT
Weight UPD
IC[64:95] CPT
Weight UPD
IC[96:127] CPT
Weight UPD
CPA#0 CPT
CPA#1 CPT
CPA#2 CPT
CPA#3 CPT
*CPT: Compute
*UPD: Update
Clock Cycle
TBA #0 CPT
CPA #0
CPA #1
CPA #2
CPA #3
4N
4N+1
4N+2
4N+3
N = Seq. of In. (N = 0, 1, 2...)

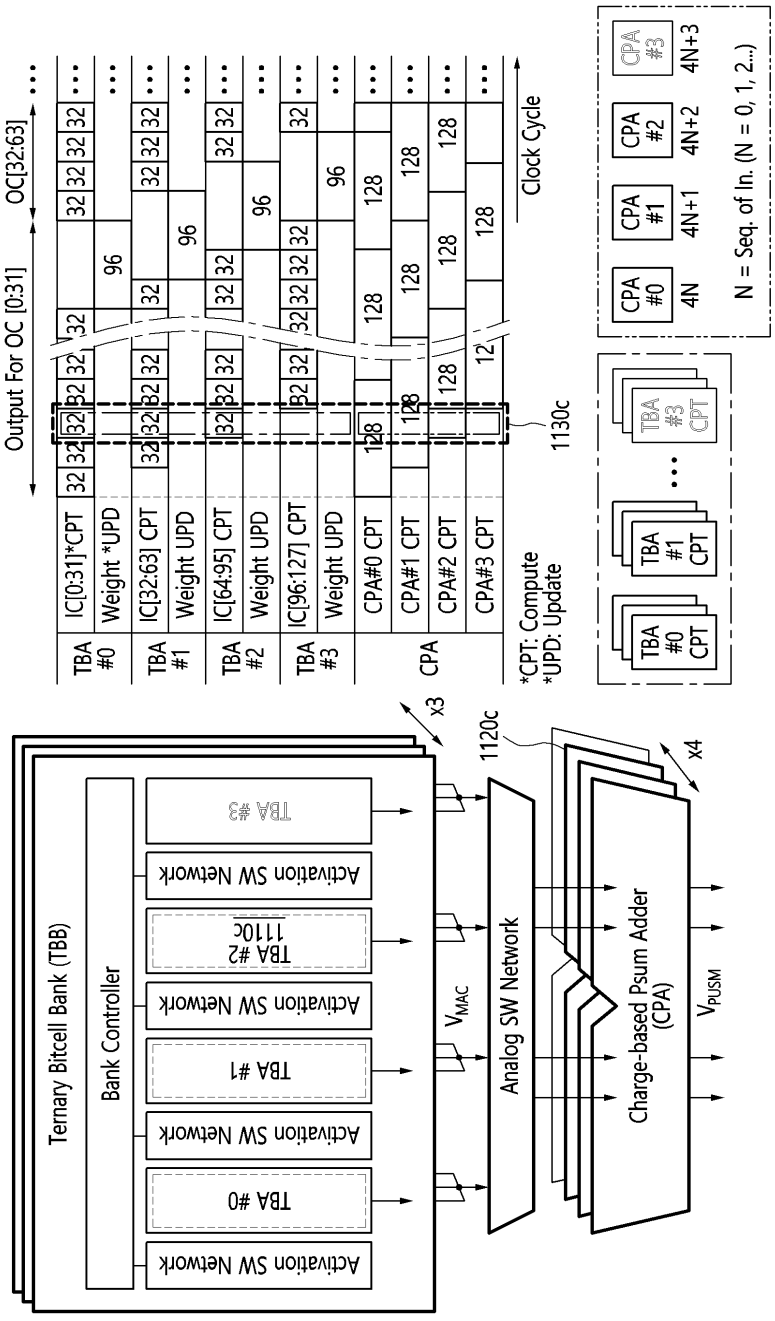
도면11a


Ternary Bitcell Bank (TBB)
Bank Controller
Activation SW Network
TBA #0 1110a
TBA #1
TBA #2
TBA #3
x3
V_{MAC}
Analog SW Network
Charge-based Psum Adder (CPA) 1120a
x4
V_{PUSM}
Output For OC [0:31]
OC[32:63]
TBA #0
TBA #1
TBA #2
TBA #3
CPA
IC[0:31]*CPT
Weight *UPD
IC[32:63] CPT
Weight UPD
IC[64:95] CPT
Weight UPD
IC[96:127] CPT
Weight UPD
CPA#0 CPT
CPA#1 CPT
CPA#2 CPT
CPA#3 CPT
Clock Cycle
*CPT: Compute
*UPD: Update
1130a
CPA #0
CPA #1
CPA #2
CPA #3
4N
4N+1
4N+2
4N+3
N = Seq. of In. (N = 0, 1, 2...)

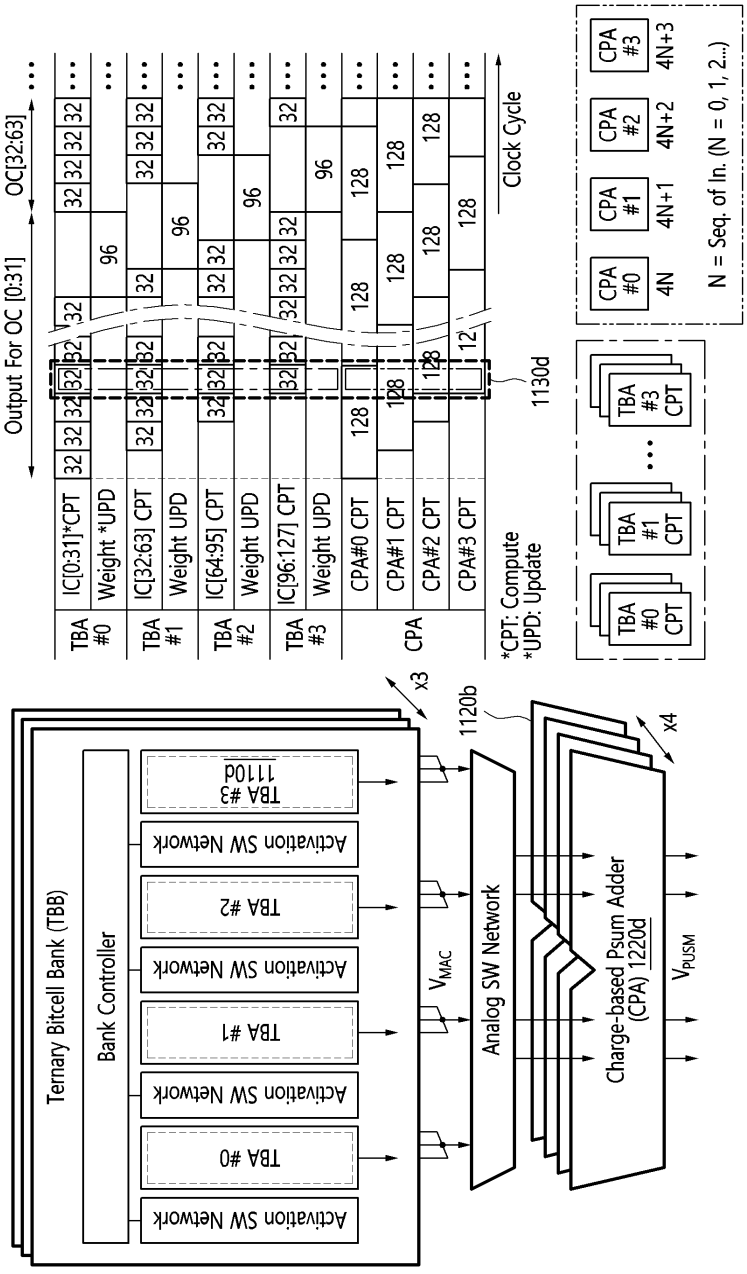
도면11b


Output For OC [0:31]
OC[32:63]
TBA #0
TBA #1
TBA #2
TBA #3
IC[0:31]*CPT
Weight *UPD
IC[32:63] CPT
Weight UPD
IC[64:95] CPT
Weight UPD
IC[96:127] CPT
Weight UPD
CPA#0 CPT
CPA#1 CPT
CPA#2 CPT
CPA#3 CPT
CPA
Clock Cycle
*CPT: Compute
*UPD: Update
1130b
CPA #0
CPA #1
CPA #2
CPA #3
4N
4N+1
4N+2
4N+3
N = Seq. of In. (N = 0, 1, 2...)
Ternary Bitcell Bank (TBB)
Bank Controller
Activation SW Network
TBA #0
TBA #1 1110b
TBA #2
TBA #3
x3
V_{MAC}
1120b
Analog SW Network
Charge-based Psum Adder (CPA)
x4
V_{PUSM}

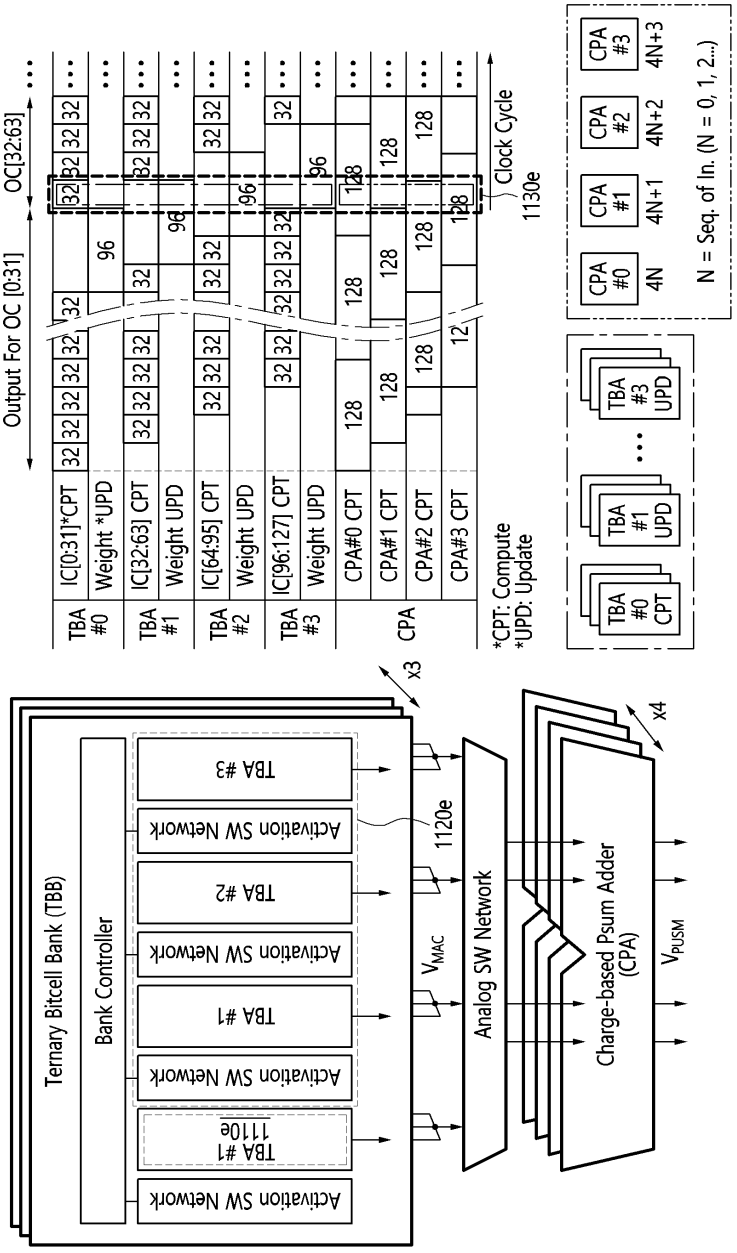
도면11c


Output For OC [0:31]
OC[32:63]
Clock Cycle
TBA #0
TBA #1
TBA #2
TBA #3
CPA
IC[0:31]*CPT
Weight *UPD
IC[32:63] CPT
Weight UPD
IC[64:95] CPT
Weight UPD
IC[96:127] CPT
Weight UPD
CPA#0 CPT
CPA#1 CPT
CPA#2 CPT
CPA#3 CPT
1130c
*CPT: Compute
*UPD: Update
CPA #0
CPA #1
CPA #2
CPA #3
4N
4N+1
4N+2
4N+3
N = Seq. of In. (N = 0, 1, 2...)
Ternary Bitcell Bank (TBB)
Bank Controller
Activation SW Network
TBA #0
TBA #1
TBA #2
TBA #3
x3
1120c
x4
Analog SW Network
Charge-based Psum Adder (CPA)
V_{MAC}
V_{PUSM}

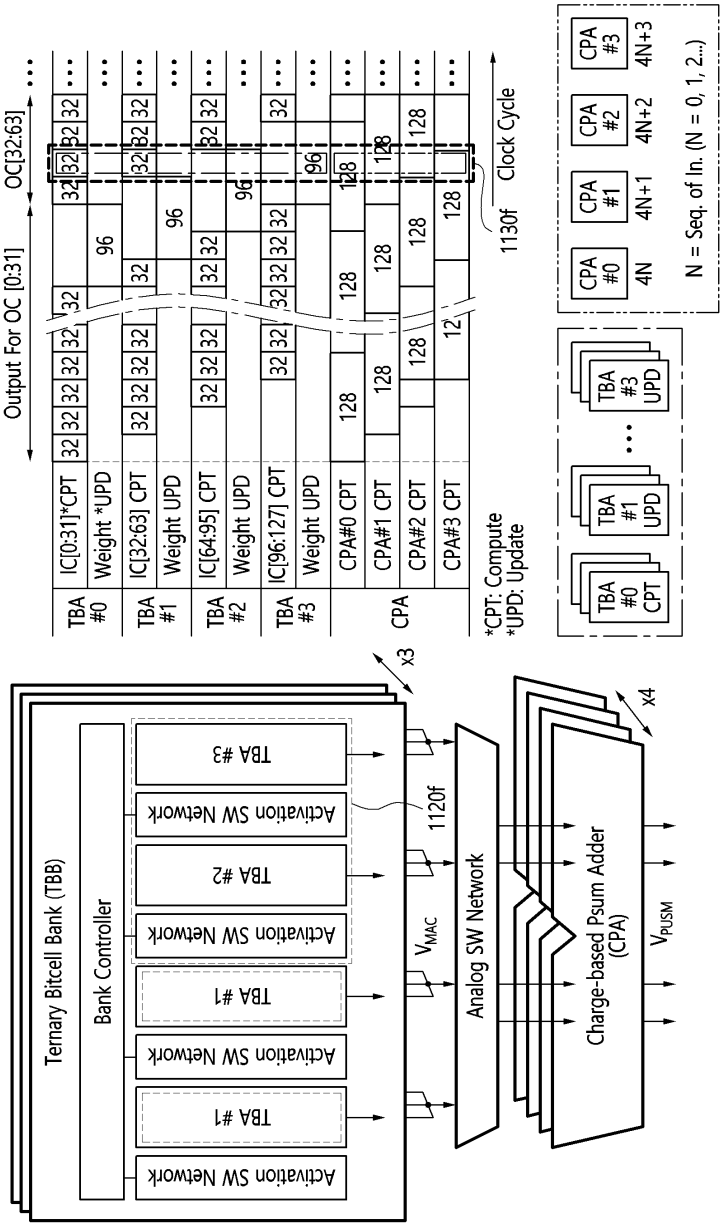
도면11d


Output For OC [0:31]
OC[32:63]
TBA #0
TBA #1
TBA #2
TBA #3
CPA
IC[0:31]*CPT
Weight *UPD
IC[32:63] CPT
Weight UPD
IC[64:95] CPT
Weight UPD
IC[96:127] CPT
Weight UPD
CPA#0 CPT
CPA#1 CPT
CPA#2 CPT
CPA#3 CPT
Clock Cycle
1130d
*CPT: Compute
*UPD: Update
CPA #0
CPA #1
CPA #2
CPA #3
4N
4N+1
4N+2
4N+3
N = Seq. of In. (N = 0, 1, 2...)
TBA #0 CPT
TBA #1 CPT
TBA #3 CPT
Ternary Bitcell Bank (TBB)
Bank Controller
Activation SW Network
TBA #0
TBA #1
TBA #2
TBA #3 1110d
x3
1120b
Analog SW Network
V_{MAC}
Charge-based Psum Adder (CPA) 1220d
x4
V_{PUSM}

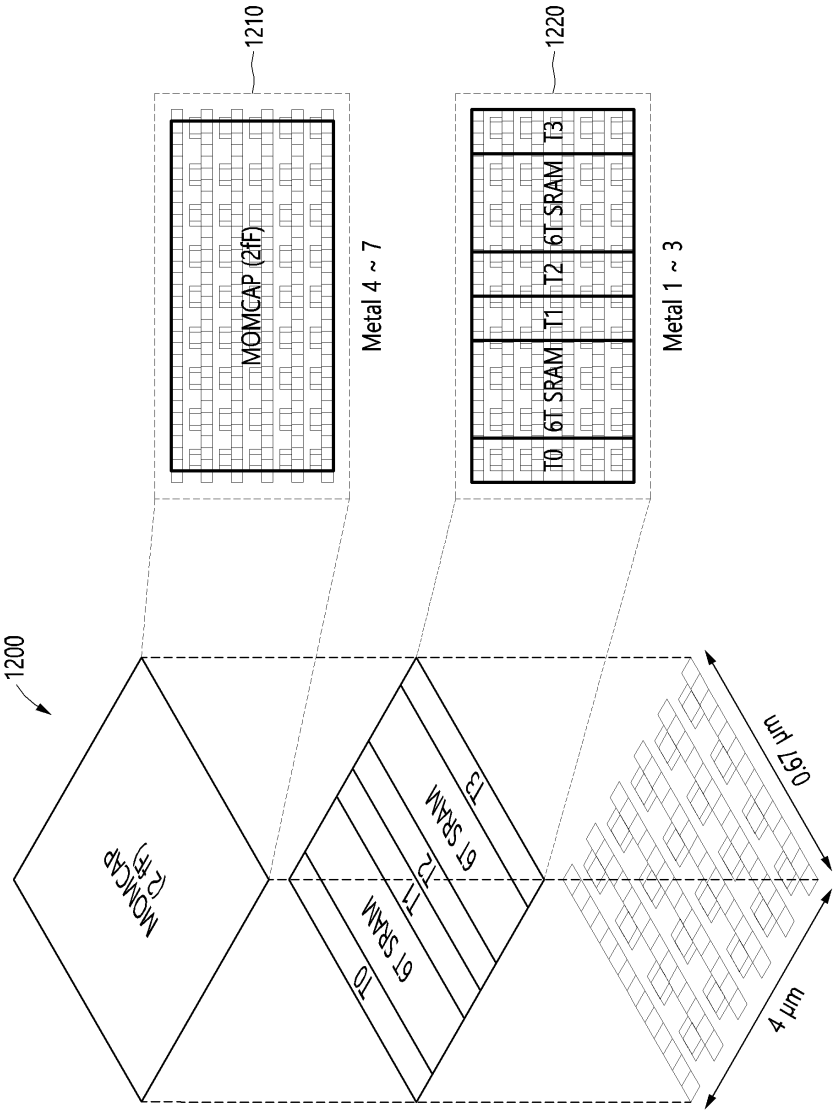
도면11e


Output For OC [0:31]
OC[32:63]
TBA #0
TBA #1
TBA #2
TBA #3
CPA
IC[0:31]*CPT
Weight *UPD
IC[32:63] CPT
Weight UPD
IC[64:95] CPT
Weight UPD
IC[96:127] CPT
Weight UPD
CPA#0 CPT
CPA#1 CPT
CPA#2 CPT
CPA#3 CPT
*CPT: Compute
*UPD: Update
Clock Cycle
1130e
CPA #0
CPA #1
CPA #2
CPA #3
4N
4N+1
4N+2
4N+3
N = Seq. of In. (N = 0, 1, 2...)
TBA #0 CPT
TBA #1 UPD
TBA #3 UPD
Ternary Bitcell Bank (TBB)
Bank Controller
Activation SW Network
TBA #1 1110e
TBA #2
TBA #3
x3
1120e
V_{MAC}
Analog SW Network
Charge-based Psum Adder (CPA)
V_{PUSM}
x4

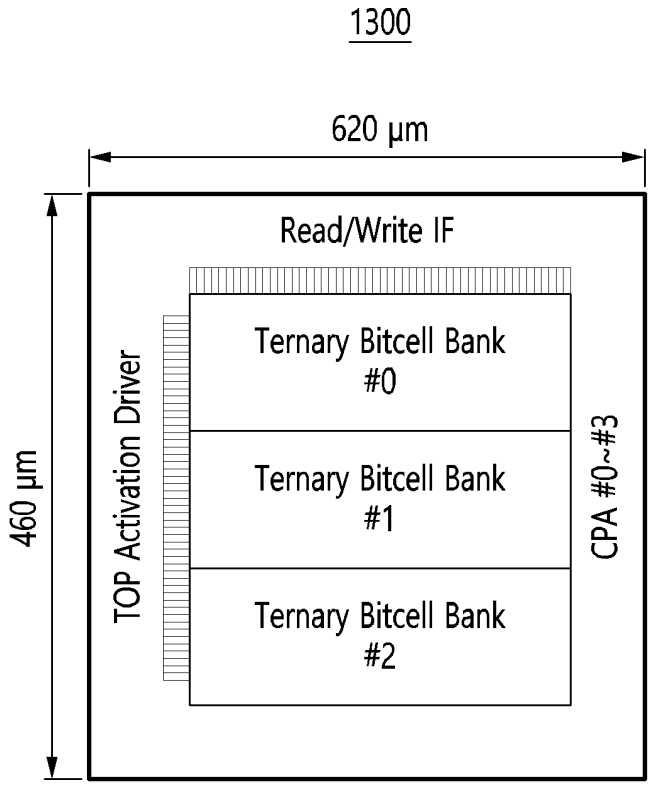
도면11f


Output For OC [0:31]
OC[32:63]
TBA #0
TBA #1
TBA #2
TBA #3
CPA
IC[0:31]*CPT
Weight *UPD
IC[32:63] CPT
Weight UPD
IC[64:95] CPT
Weight UPD
IC[96:127] CPT
Weight UPD
CPA#0 CPT
CPA#1 CPT
CPA#2 CPT
CPA#3 CPT
1130f
Clock Cycle
*CPT: Compute
*UPD: Update
TBA #0 CPT
TBA #1 UPD
TBA #3 UPD
CPA #0
CPA #1
CPA #2
CPA #3
4N
4N+1
4N+2
4N+3
N = Seq. of In. (N = 0, 1, 2...)
Ternary Bitcell Bank (TBB)
Bank Controller
Activation SW Network
TBA #1
TBA #2
TBA #3
x3
1120f
V_{MAC}
Analog SW Network
Charge-based Psum Adder (CPA)
V_{PUSM}
x4

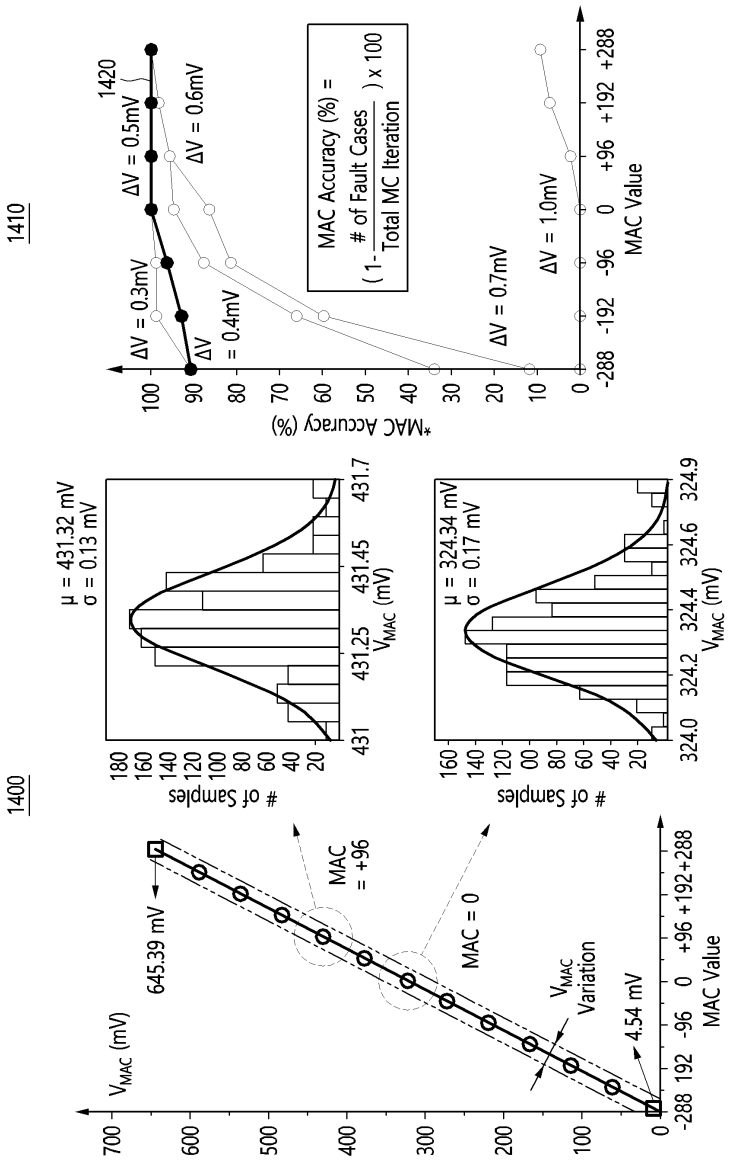
도면12


1210
1220
MOMCAP (2ff)
Metal 4 ~ 7
T0 6T SRAM T1 T2 6T SRAM T3
Metal 1 ~ 3
1200
MOMCAP (2ff)
T0 6T SRAM T1 T2 6T SRAM T3
0.67 μm
4 μm

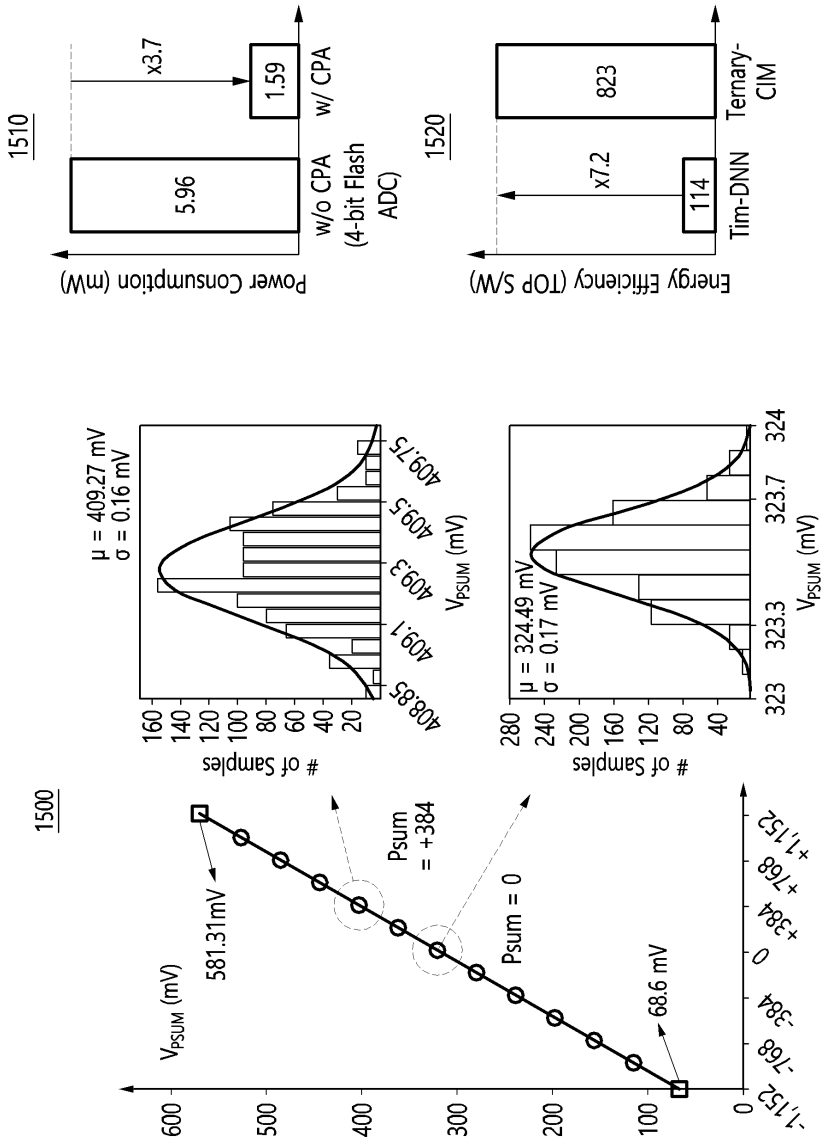
도면13


1300
620 μm
460 μm
Read/Write IF
TOP Activation Driver
Ternary Bitcell Bank #0
Ternary Bitcell Bank #1
Ternary Bitcell Bank #2
CPA #0~#3

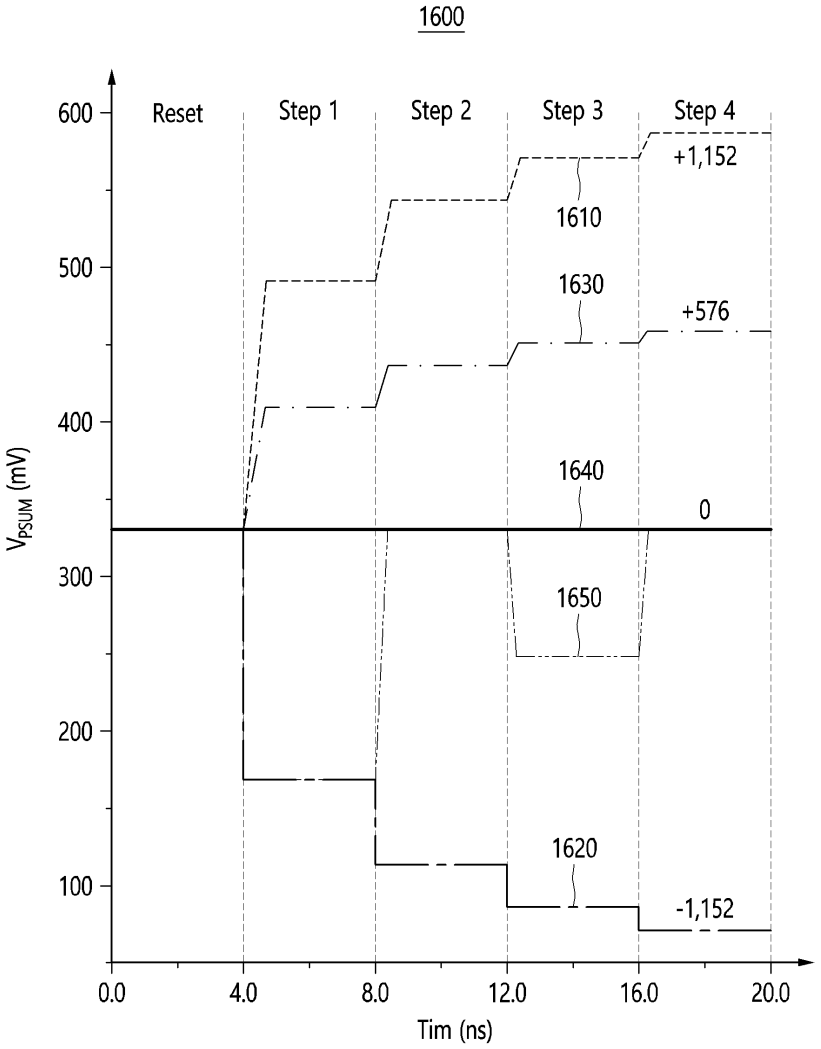
도면14


1400
1410
1420
V_{MAC} (mV)
645.39 mV
MAC = +96
MAC = 0
V_{MAC} Variation
4.54 mV
MAC Value
# of Samples
μ = 431.32 mV
σ = 0.13 mV
μ = 324.34 mV
σ = 0.17 mV
*MAC Accuracy (%)
ΔV = 0.3mV
ΔV = 0.4mV
ΔV = 0.5mV
ΔV = 0.6mV
ΔV = 0.7mV
ΔV = 1.0mV
MAC Accuracy (%) = (1- # of Fault Cases / Total MC Iteration) x 100

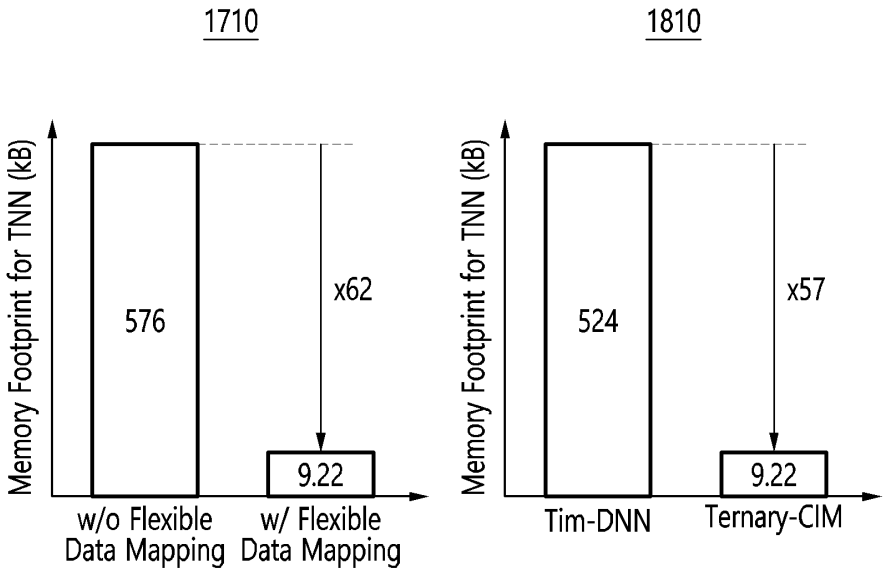
도면15


1500
1510
1520
Power Consumption (mW)
5.96
w/o CPA
(4-bit Flash ADC)
x3.7
1.59
w/ CPA
Energy Efficiency (TOP S/W)
114
Tim-DNN
x7.2
823
Ternary-CIM
μ = 409.27 mV
σ = 0.16 mV
# of Samples
V_PSUM (mV)
μ = 324.49 mV
σ = 0.17 mV
581.31mV
68.6 mV
Psum = +384
Psum = 0
V_PSUM (mV)

도면16


1600
Reset
Step 1
Step 2
Step 3
Step 4
600
500
400
300
200
100
V_{PSUM} (mV)
+1,152
1610
1630
+576
1640
0
1650
1620
-1,152
0.0
4.0
8.0
12.0
16.0
20.0
Tim (ns)

도면17


1710
1810
Memory Footprint for TNN (kB)
576
x62
9.22
w/o Flexible Data Mapping
w/ Flexible Data Mapping
Memory Footprint for TNN (kB)
524
x57
9.22
Tim-DNN
Ternary-CIM