(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0116837
(43) 공개일자 2022년08월23일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2006.01) *G06F 3/06* (2006.01)
G06N 3/04 (2006.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06F 3/0656 (2013.01)
(21) 출원번호 10-2021-0020195
(22) 출원일자 2021년02월16일
심사청구일자 2021년02월16일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
정성우
서울특별시 강남구 개포로 264, 124동 2505호(개포동, 개포래미안포레스트)
이영서
서울특별시 동대문구 무학로49길 40, 502호(용두동)
공영호
서울특별시 동대문구 무학로49길 25 , 206호 (용두동,대명빌딩)
(74) 대리인
송인호, 윤형근, 최관락

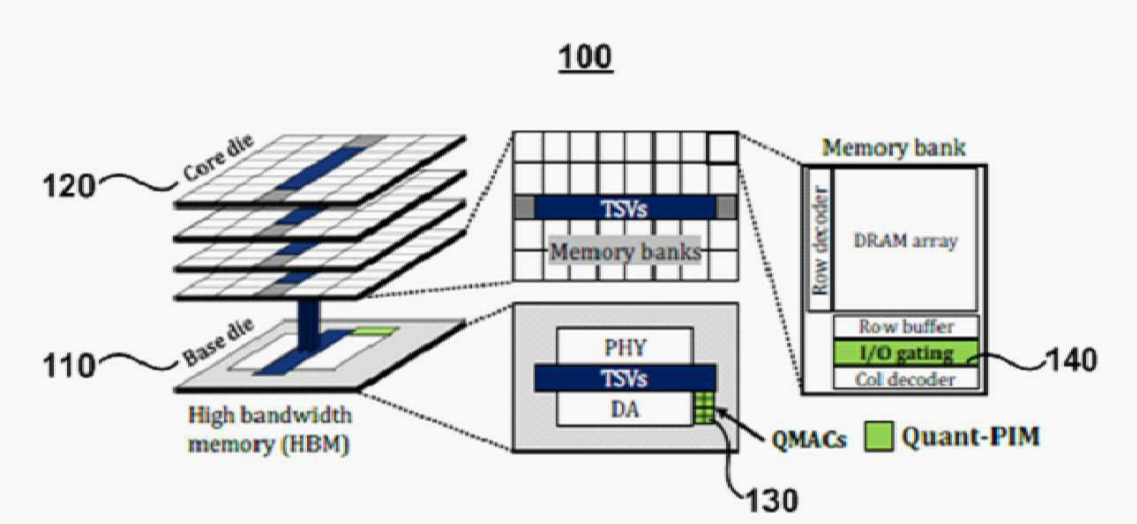
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 **레이어-단위 양자화 신경망을 위한 인-메모리 가속기 및 이의 동작 방법**

(57) 요 약

레이어-단위 양자화 신경망을 위한 인-메모리 가속기 및 이의 동작 방법이 개시된다. 고대역 메모리 내부에 형성되는 레이어-단위 양자화 신경망을 위한 인-메모리 가속기는, 상기 고대역 메모리의 베이스 다이부에 형성되며, 호스트 프로세서의 메모리 주소와 데이터 크기에 따라 메모리 접근 명령어를 생성하고, 상기 생성된 메모리 접근 명령어와 현재 연산이 수행되는 레이어의 필요 정밀도 정보를 출력하는 프로세스 모듈; 및 상기 고대역 메모리의 코어 다이부에 형성되며, 상기 프로세스 모듈의 메모리 접근 명령어에 따른 메모리 뱅크에 접근하여 데이터를 불러오되, 상기 필요 정밀도 정보에 따른 비트 단위로 상기 메모리 뱅크로부터 데이터를 불러오는 I/O 게이트부를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 3/0658 (2013.01)
G06N 3/04 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711110769
과제번호 2020R1A2C2003500
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 개인기초연구(과기정통부)(R&D)
연구과제명 온도를 고려한 3D 적층 메모리 기반 인메모리 가속기
기 여 율 1/2
과제수행기관명 고려대학교 산학협력단
연구기간 2020.03.01 ~ 2021.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1711121315
과제번호 2020M3F3A2A01082329
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 차세대지능형반도체기술개발(R&D)
연구과제명 저온 공정 기반 Si/SiGe M3D 집적 소자 및 회로 플랫폼 기술개발
기 여 율 1/2
과제수행기관명 고려대학교 산학협력단
연구기간 2020.07.01 ~ 2020.12.31

명 세 서

청구범위

청구항 1

고대역 메모리 내부에 형성되는 레이어-단위 양자화 신경망을 위한 인-메모리 가속기에 있어서,

상기 고대역 메모리의 베이스 다이부에 형성되며, 호스트 프로세서의 메모리 주소와 데이터 크기에 따라 메모리 접근 명령어를 생성하고, 상기 생성된 메모리 접근 명령어와 현재 연산이 수행되는 레이어의 필요 정밀도 정보를 출력하는 프로세스 모듈; 및

상기 고대역 메모리의 코어 다이부에 형성되며, 상기 프로세스 모듈의 메모리 접근 명령어에 따른 메모리 뱅크에 접근하여 데이터를 불러오되, 상기 필요 정밀도 정보에 따른 비트 단위로 상기 메모리 뱅크로부터 데이터를 불러오는 I/O 게이트부를 포함하는 인-메모리 가속기.

청구항 2

제1 항에 있어서,

상기 프로세스 모듈은,

상기 고대역 메모리의 메모리 채널 개수만큼 상기 베이스 다이부에 복수개 형성되되,

복수의 프로세스 모듈은 각각 독립적으로 메모리 채널에 접근 가능한 것을 특징으로 하는 인-메모리 가속기.

청구항 3

제1 항에 있어서,

상기 프로세스 모듈은,

상기 호스트 프로세서로부터 메모리 주소, 데이터 크기 및 필요 정밀도 정보를 획득하며, 상기 메모리 주소와 상기 데이터 크기에 기반하여 메모리 접근 명령어를 생성하고, 상기 생성된 메모리 접근 명령어와 상기 필요 정밀도 정보를 실리콘 비아(TSV: through silicon via)를 통해 출력하는 로컬 메모리 컨트롤러;

상기 메모리 접근 명령어와 상기 필요 정밀도 정보에 따라 상기 I/O 게이트부에서 선택적으로 불러온 데이터를 일시적으로 저장하는 입력 버퍼;

상기 입력 버퍼에 저장된 데이터를 이용하여 MAC 연산을 수행하는 이진 MAC 연산부;

상기 이진 MAC 연산부의 연산 결과를 비트별로 축적하는 축적기; 및

상기 축적기를 통해 축적된 연산 결과를 저장하는 결과 버퍼를 포함하는 것을 특징으로 하는 인-메모리 가속기.

청구항 4

제3 항에 있어서,

상기 이진 MAC 연산부는, 최대 정밀도가 N인 경우, N의 제곱개의 이진 MAC 연산 유닛을 포함하는 것을 특징으로 하는 인-메모리 가속기.

청구항 5

제4 항에 있어서,

상기 I/O 게이트부는,

상기 필요 정밀도가 N/2 이하인 경우, 두개의 데이터 블록을 읽어 상기 프로세스 모듈로 출력하되,

상기 프로세스 모듈은,

상기 두개의 데이터 블록을 이용하여 두개의 MAC 연산을 수행하는 것을 특징으로 하는 인-메모리 가속기.

청구항 6

레이어-단위 양자화 신경망을 위한 인-메모리 가속기의 동작 방법에 있어서,

(a) 베이스 다이에 형성된 프로세스 모듈이 현재 연산이 수행되는 레이어의 필요 정밀도 정보와 데이터 블록에 대한 메모리 접근 명령어를 실리콘 비아를 통해 출력하는 단계;

(b) 코어 다이부에 형성된 I/O 게이트부가 상기 메모리 접근 명령어에 따른 메모리 뱅크에 접근하여 필요 정밀도에 따른 비트 단위로 데이터 블록을 독출하여 상기 실리콘 비아를 통해 상기 프로세스 모듈로 출력하는 단계; 및

(c) 상기 프로세스 모듈이 상기 데이터 블록을 이용하여 이진 MAC 연산을 수행하여 축적한 결과값을 저장하는 단계를 포함하는 인-메모리 가속기의 동작 방법.

청구항 7

제6 항에 있어서,

상기 (b) 단계에서,

상기 필요 정밀도가 N/2 이하인 경우, 두개의 데이터 블록을 읽어 상기 프로세스 모듈로 출력하되,

상기 (c) 단계는,

상기 두개의 데이터 블록을 이용하여 두개의 MAC 연산을 수행하는 것을 특징으로 하는 인-메모리 가속기의 동작 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 레이어-단위 양자화 신경망을 위한 인-메모리 가속기 및 이의 동작방법에 관한 것이다.

배 경 기 술

[0003] 기존 신경망(deep neural network, DNN) 구조가 요구하는 많은 수의 multiply-and-accumulate(MAC) 연산과 큰 메모리 대역폭 요구량을 절감하기 위해서 양자화 신경망(quantized neural network, QNN)이 등장했다. 양자화 신경망은 기존 부동 소수점(floating-point) 기반의 데이터를 고정 소수점(fixed-point) 기반의 데이터로 변경해 MAC 연산 시에 필요한 에너지를 크게 절감하며, 기존 신경망 구조 대비 상대적으로 낮은 정밀도(precision)를 가지는 데이터를 활용하기 때문에 메모리 대역폭 요구량을 큰 폭으로 절감할 수 있다.

[0004] 하지만, 입력 데이터의 크기가 점점 커지고 신경망을 구성하는 레이어 (layer)의 수가 증가함에 따라서, 양자화 신경망은 여전히 많은 에너지를 야기한다. 양자화 신경망의 에너지를 절감하기 위해서, 최근 단일 신경망 내에서 레이어 단위로 다른 크기의 정밀도를 적용하는 레이어-단위 양자화 신경망(layer-wise quantized neural network)이 등장했다. 레이어-단위 양자화 신경망은 각 레이어 별로 필요한 크기의 정밀도만을 신경망 연산(예

를 들어, weight와 activation의 convolution 연산 등)에 활용하여 기존 양자화 신경망 대비 연산 에너지를 큰 폭으로 절감할 수 있다.

[0005] 현재까지 레이어-단위 양자화 신경망을 위한 많은 연구가 이루어져 왔지만, 지금까지의 연구들은 오직 레이어-단위 양자화 신경망을 적용했을 경우 얻을 수 있는 연산 에너지의 절감에만 초점을 맞추고, 레이어-단위 양자화 신경망에서 메모리 접근 최적화는 고려하지 않았다. 예를 들어, 기존 레이어-단위 양자화 신경망에서는 정밀도가 낮은 경우에도 한 번의 메모리 접근으로 하나의 데이터 워드(data word) 크기의 데이터를 불러와야 한다. 결과적으로, 레이어-단위 양자화 신경망은 최적화되지 않은 메모리 접근으로 인해 상당한 양의 에너지를 소모하는 문제점이 있다.

발명의 내용

해결하려는 과제

[0007] 본 발명은 레이어-단위 양자화 신경망의 연산 에너지뿐만 아니라 메모리 접근시 에너지를 절감할 수 있는 레이어-단위 양자화 신경망을 위한 인-메모리 가속기 및 이의 동작 방법을 제공하기 위한 것이다.

[0008] 또한, 본 발명은 레이어별 필요 정밀도에 따라 메인 메모리에 저장되어 있는 데이터 워드 내 필요 데이터만을 선택적으로 접근하여 메모리 접근시의 전력 소모를 큰 폭으로 절감할 수 있는 레이어-단위 양자화 신경망을 위한 인-메모리 가속기 및 이의 동작 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0010] 본 발명의 일 측면에 따르면, 레이어-단위 양자화 신경망을 위한 인-메모리 가속기가 제공된다.

[0011] 본 발명의 일 실시예에 따르면, 고대역 메모리 내부에 형성되는 레이어-단위 양자화 신경망을 위한 인-메모리 가속기에 있어서, 상기 고대역 메모리의 베이스 다이부에 형성되며, 호스트 프로세서의 메모리 주소와 데이터 크기에 따라 메모리 접근 명령어를 생성하고, 상기 생성된 메모리 접근 명령어와 현재 연산이 수행되는 레이어의 필요 정밀도 정보를 출력하는 프로세스 모듈; 및 상기 고대역 메모리의 코어 다이부에 형성되며, 상기 프로세스 모듈의 메모리 접근 명령어에 따른 메모리 뱅크에 접근하여 데이터를 불러오되, 상기 필요 정밀도 정보에 따른 비트 단위로 상기 메모리 뱅크로부터 데이터를 불러오는 I/O 게이트부를 포함하는 인-메모리 가속기가 제공될 수 있다.

[0012] 상기 프로세스 모듈은, 상기 고대역 메모리의 메모리 채널 개수만큼 상기 베이스 다이부에 복수개 형성되되, 복수의 프로세스 모듈은 각각 독립적으로 메모리 채널에 접근 가능하다.

[0013] 상기 프로세스 모듈은, 상기 호스트 프로세서로부터 메모리 주소, 데이터 크기 및 필요 정밀도 정보를 획득하며, 상기 메모리 주소와 상기 데이터 크기에 기반하여 메모리 접근 명령어를 생성하고, 상기 생성된 메모리 접근 명령어와 상기 필요 정밀도 정보를 실리콘 비아(TSV: through silicon via)를 통해 출력하는 로컬 메모리 컨트롤러; 상기 메모리 접근 명령어와 상기 필요 정밀도 정보에 따라 상기 I/O 게이트부에서 선택적으로 불러온 데이터를 일시적으로 저장하는 입력 버퍼; 상기 입력 버퍼에 저장된 데이터를 이용하여 MAC 연산을 수행하는 이진 MAC 연산부; 상기 이진 MAC 연산부의 연산 결과를 비트별로 축적하는 축적기; 및 상기 축적기를 통해 축적된 연산 결과를 저장하는 결과 버퍼를 포함할 수 있다.

[0014] 상기 이진 MAC 연산부는, 최대 정밀도가 N인 경우, N의 제곱개의 이진 MAC 연산 유닛을 포함할 수 있다.

[0015] 상기 I/O 게이트부는, 상기 필요 정밀도가 N/2 이하인 경우, 두개의 데이터 블록을 읽어 상기 프로세스 모듈로 출력하되, 상기 프로세스 모듈은, 상기 두개의 데이터 블록을 이용하여 두개의 MAC 연산을 수행할 수 있다.

[0017] 본 발명의 다른 측면에 따르면, 레이어-단위 양자화 신경망을 위한 인-메모리 가속기의 동작 방법이 제공된다.

[0018] 본 발명의 일 실시예에 따르면, 레이어-단위 양자화 신경망을 위한 인-메모리 가속기의 동작 방법에 있어서, (a) 베이스 다이에 형성된 프로세스 모듈이 현재 연산이 수행되는 레이어의 필요 정밀도 정보와 데이터 블록에 대한 메모리 접근 명령어를 실리콘 비아를 통해 출력하는 단계; (b) 코어 다이부에 형성된 I/O 게이트부가 상기

메모리 접근 명령어에 따른 메모리 뱅크에 접근하여 필요 정밀도에 따른 비트 단위로 데이터 블록을 독출하여 상기 실리콘 비아를 통해 상기 프로세스 모듈로 출력하는 단계; 및 (c) 상기 프로세스 모듈이 상기 데이터 블록을 이용하여 이진 MAC 연산을 수행하여 축적한 결과값을 저장하는 단계를 포함할 수 있다.

[0019] 상기 (b) 단계에서, 상기 필요 정밀도가 N/2 이하인 경우, 두개의 데이터 블록을 읽어 상기 프로세스 모듈로 출력하되, 상기 (c) 단계는, 상기 두개의 데이터 블록을 이용하여 두개의 MAC 연산을 수행할 수 있다.

발명의 효과

[0021] 본 발명의 일 실시예에 따른 레이어-단위 양자화 신경망을 위한 인-메모리 가속기 및 이의 동작 방법을 제공함으로써, 레이어-단위 양자화 신경망의 연산 에너지뿐만 아니라 메모리 접근시 에너지를 절감할 수 있는 이점이 있다.

[0022] 또한, 본 발명은 레이어별 필요 정밀도에 따라 메인 메모리에 저장되어 있는 데이터 워드 내 필요 데이터만을 선택적으로 접근하여 메모리 접근시의 전력 소모를 절감할 수 있는 이점도 있다.

도면의 간단한 설명

[0024] 도 1은 본 발명의 일 실시예에 따른 레이어-단위 양자화 신경망을 위한 인-메모리 가속기의 구조를 도시한 도면.

도 2는 본 발명의 일 실시예에 따른 프로세스 모듈의 세부 구조를 도시한 도면.

도 3은 본 발명의 일 실시예에 따른 인-메모리 가속기의 동작 방법을 나타낸 순서도.

도 4는 본 발명의 일 실시예에 따른 한번의 메모리 액세스로 두개의 데이터 블록을 독출하는 일 예를 설명하기 위해 도시한 도면.

도 5는 종래와 본 발명의 일 실시예에 따른 신경망에서의 실행 시간과 레이어별 실행 시간을 비교한 결과를 도시한 그래프.

도 6은 종래와 본 발명의 일 실시예에 따른 신경망 연산에 따른 전력 소비를 비교한 결과를 도시한 그래프.

도 7은 신경망의 전력 소비를 기반으로 GPU에서 발산되는 열을 고려하여 최고 온칩 온도를 분석한 결과를 나타낸 그래프.

도 8은 신경망에서 본 발명의 일 실시예에 따른 인-메모리 가속기의 시스템 에너지를 나타낸 그래프.

발명을 실시하기 위한 구체적인 내용

[0025] 본 명세서에서 사용되는 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "구성된다" 또는 "포함한다" 등의 용어는 명세서상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다. 또한, 명세서에 기재된 "...부", "모듈" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어와 소프트웨어의 결합으로 구현될 수 있다.

[0026] 이하, 첨부된 도면들을 참조하여 본 발명의 실시예를 상세히 설명한다.

[0028] 도 1은 본 발명의 일 실시예에 따른 레이어-단위 양자화 신경망을 위한 인-메모리 가속기의 구조를 도시한 도면이고, 도 2는 본 발명의 일 실시예에 따른 프로세스 모듈의 세부 구조를 도시한 도면이다.

[0029] 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 고대역폭 메모리(HBM: High Band Memory, 이하 HBM이라 칭하기로 함) 내부에 형성되는 것을 가정하기로 한다. 여기서, HBM은 베이스 다이부(110)와 복수의 코어 다이부(120)를 포함할 수 있다.

[0030] HBM의 가장 하부 층인 베이스 다이부(110)는 메모리 다이가 아니며 로직 공정으로 제작된 로직 다이로써 다양한 프로세스 모듈을 구현하는데 널리 활용되고 있다.

[0031] 따라서, 본 발명의 일 실시예에 따른 인-메모리 가속기(100) 또한 프로세스 모듈(130)을 베이스 다이부(110)에 형성하기로 한다. 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 메모리 채널별로 프로세스 모듈(130)을 생성할 수 있다. 예를 들어, HBM의 메모리 채널이 8개인 경우, 프로세스 모듈(130)은 베이스 다이부(110)에 8개 형성되며, 각 프로세스 모듈은 독립적으로 메모리 채널에 접근할 수 있다.

[0032] HBM의 가장 하부 층을 제외한 나머지 층은 코어 다이부(120)로 메모리 공정으로 제작된 메모리 다이(memory die)이다. HBM에서 각 메모리 채널은 독립적으로 동작될 수 있으므로, 효과적인 동작을 위해 각 메모리 채널별로 각각의 프로세스 모듈을 배치할 수 있다.

[0033] 코어 다이부(120)는 복수의 메모리 뱅크들로 구성된다. 코어 다이부(120)에는 I/O 게이트부(140)가 위치된다. 본 발명의 일 실시예에 따른 I/O 게이트부(140)는 워드 단위로 메모리 뱅크로부터 데이터를 불러오는 것이 아니라, 각 레이어 별 필요 정밀도에 따라 필요한 데이터만을 메모리 뱅크로부터 불러올 수 있다.

[0034] 본 발명의 일 실시예에 따르면, 인-메모리 가속기(100)의 프로세스 모듈(130)은 베이스 다이부(110)에 형성되며, I/O 게이트부(140)는 코어 다이부(120)에 형성되므로, 프로세스 모듈(130)과 I/O 게이트부(140)는 실리콘 비아(TSV: through silicon via, 이하 TSV라 칭하기로 함)를 통해 연결될 수 있다.

[0035] 이하에서는 인-메모리 가속기(100)의 각각의 구성에 대해 보다 상세히 설명하기로 한다.

[0036] 도 2에는 프로세스 모듈(130)의 세부 구조가 도시되어 있다.

[0037] 도 2를 참조하면, 본 발명의 일 실시에에 따른 프로세스 모듈(130)은 로컬 메모리 컨트롤러(210), 데이터 버퍼(220), 이진 MAC 연산부(230), 축적기(240)를 포함한다.

[0038] 로컬 메모리 컨트롤러(210)는 독립적으로 메모리에 접근하기 위해 메모리 접근 명령어를 생성하고 이를 전송하는 역할을 수행한다.

[0039] 데이터 버퍼(220)는 연산을 위해 필요한 데이터를 임시 저장하는 입력 데이터 버퍼(220a, 220b), 연산 결과를 저장하는 결과 버퍼(220c)로 구성된다.

[0040] 도 2에 도시된 바와 같이, 이진 MAC 연산부(230)에서의 연산을 위해 필요한 데이터를 임시 저장하기 위해 두개의 입력 데이터 버퍼(220a, 220b)와 결과 버퍼(220c)가 위치될 수 있다. 두개의 입력 데이터 버퍼(220a, 220b) 중 어느 하나는 이진 MAC 연산부(230)의 행 방향에 위치되며, 다른 하나는 열 방향에 위치될 수 있다.

[0041] 결과 버퍼(220c)는 이진 MAC 연산부(230)에서 이진 MAC 연산 장치의 연산 결과로 출력되는 각 비트별 연산 결과를 하나로 축적한 결과를 저장하기 위해, 축적기(240)의 후단에 위치될 수 있다.

[0042] 이진 MAC 연산부(230)는 비트별 MAC 연산을 수행한다.

[0043] 본 발명의 일 실시예에 따르면, 이진 MAC 연산부(230)는 다수개의 이진 MAC 연산 유닛을 포함할 수 있다. 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 각 레이어의 정밀도에 따라 필요한 데이터만을 메모리로부터 불러올 수 있다. 각 레이어별로 서로 다른 정밀도 데이터를 불러오기 때문에 이진 MAC 연산부(230)는 모든 정밀도에 대한 연산을 수행하기 위해 다수의 이진 MAC 연산 유닛을 배치한다. N 비트 정밀도를 가지는 MAC 연산의 경우, N의 제곱개의 이진 MAC 연산으로 변환된다. 따라서, 본 발명의 일 실시예에 따른 이진 MAC 연산부(230)는 N비트의 정밀도에 대한 모든 MAC 연산을 처리하기 위해 N의 제곱개의 이진 MAC 연산 유닛을 구비할 수 있다.

[0044] 최신 양자화 신경망에서 가장 높은 정밀도가 16 비트인 것을 고려하면, 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 16의 제곱인 256개의 이진 MAC 연산 유닛이 배치될 수 있다.

[0045] 이진 MAC 연산 유닛에 의한 MAC 연산 결과가 축적기(240)에 누적되어 최종 MAC 연산이 수행되게 된다. 즉, 이진 MAC 연산부(230)의 각 비트에 따른 이진 MAC 연산 결과는 각 비트별로 축적기(240)에서 축적될 수 있다.

[0046] 축적기(240)의 후단에는 결과 버퍼(220c)가 배치되며, 축적기(240)에 의해 축적된 연산 결과를 저장한다.

[0047] 이하, 도 3을 참조하여 인-메모리 가속기(100)의 동작 방법에 대해 보다 상세히 설명하기로 한다.

[0049] 도 3은 본 발명의 일 실시예에 따른 인-메모리 가속기의 동작 방법을 나타낸 순서도이며, 도 4는 본 발명의 일

실시예에 따른 한번의 메모리 액세스로 두개의 데이터 블록을 독출하는 일 예를 설명하기 위해 도시한 도면이다. 이하에서는 호스트 프로세서가 MAC 연산을 인-메모리 가속기(100)에 일임하는 것을 가정하여 이를 중심으로 설명하기로 한다.

[0050] 단계 310에서 프로세스 모듈(130)은 호스트 프로세서로부터 양자화 신경망 연산에 사용되는 데이터의 메모리 주소, 데이터 크기, 현재 연산이 수행되는 레이어의 필요 정밀도를 수신한다.

[0051] 단계 315에서 프로세스 모듈(130)의 로컬 메모리 컨트롤러(210)는 호스트 프로세서로부터 전송 받은 메모리 주소와 데이터 크기에 기반하여 메모리 접근 명령을 생성한다. 여기서, 메모리 접근 명령은 해당 메모리 주소와 데이터 크기에 기반하여 행과 열의 접근을 위한 명령일 수 있다.

[0052] 단계 320에서 프로세스 모듈(130)의 로컬 메모리 컨트롤러(210)는 생성된 메모리 접근 명령과 필요 정밀도 정보를 TSV를 통해 해당 메모리 주소에 따른 메모리 뱅크(DRAM 뱅크)로 출력한다.

[0053] 본 발명의 일 실시예에 따르면, 프로세스 모듈(130)은 베이스 다이부(110)에 형성되며, 실제 메모리 다이는 코어 다이부(120)로 베이스 다이부(110)와는 물리적으로 떨어져 있으므로 TSV를 통해 신호(메모리 접근 명령, 필요 정밀도 정보)가 전달되어야 한다.

[0054] 단계 325에서 메모리 뱅크 내에 위치한 I/O 게이트부(140)는 TSV를 통해 전달받은 필요 정밀도에 따라 로우 버퍼로부터 하나의 데이터 워드내의 필요한 데이터 비트만을 선택적으로 읽어온다.

[0055] 이때, I/O 게이트부(140)는 필요 정밀도 정보가 저장된 데이터 크기의 절반 이하인 경우, 하나의 메모리 접근 명령어로 두개의 데이터를 한번에 읽어올 수 있다.

[0056] 예를 들어, 하나의 데이터 워드가 16이라고 가정하기로 하며, 필요 정밀도 정보가 8비트 이하라고 가정하기로 한다.

[0057] I/O 게이트부(140)는 도 4에 도시된 바와 같이, 두개의 데이터(D_1, D_2)를 한번의 메모리 접근으로 읽어올 수 있다.

[0058] 단계 330에서 메모리 뱅크 내에 위치한 I/O 게이트부(140)는 메모리 뱅크내에서 독출한 데이터를 TSV를 통해 프로세스 모듈(130)로 전송한다.

[0059] 단계 335에서 프로세스 모듈(130)은 TSV를 통해 획득한 데이터(데이터 블록)에 기초하녀 MAC 연산을 수행한다.

[0060] n-비트 정밀도를 가지는 하나의 MAC 연산은 n^2 1-비트 MAC 연산으로 대체되기 때문에, 프로세스 모듈(130)은 다수의 이진 MAC 유닛과 축적기를 가진다. 또한, 프로세스 모듈(130)은 QNN에서의 최대 정밀도를 고려하여 256(16^2) 이진 MAC 유닛과 축적기를 구비할 수 있다. 따라서, 프로세스 모듈(130)은 16비트 이하의 정밀도를 가지는 MAC 연산을 수행할 수 있다. MAC 연산의 결과는 축적기에 축적된 후 결과 버퍼(220c)에 저장된다.

[0061] 단계 340에서 프로세스 모듈(130)은 쓰기 명령어를 생성하여 결과 버퍼에 저장된 값을 메모리 뱅크에 저장한다.

[0062] 보다 상세하게 프로세스 모듈(130)내의 로컬 메모리 컨트롤러(210)가 쓰기 명령어를 생성하여 결과 버퍼에 저장된 축적된 값을 메모리 뱅크에 저장할 수 있다. 프로세스 모듈(130)은 메모리 뱅크에 데이터를 기록할 때도, 메모리 뱅크에서 데이터를 읽어올 때와 마찬가지로 필요 정밀도에 따라 필요한 데이터의 위치만 쓰기 연산을 수행할 수 있다.

[0063] 인-메모리 가속기(100)는 단계 315 내지 단계 340까지의 과정을 MAC 연산이 종료될 때까지 반복 수행한다. MAC 연산이 수행된 이후, 인-메모리 가속기(100)는 인터럽트 신호를 통해 호스트 프로세서에게 모든 MAC 연산이 종료되었음을 알릴 수 있다.

[0064] 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 필요 정밀도가 가중치(또는 활성화) 크기의 절반(예를 들어, 8비트 이하)인 경우 레이어별 QNN의 성능을 향상시킬 수 있다. 인-메모리 가속기(100)는 선택적 메모리 액세스에서 저장된 메모리 대역폭을 활용하여 단일 메모리 작업에서 두개의 데이터 블록을 읽을 수 있다. 즉, 인-메모리 가속기(100)는 메모리 요청을 2개의 데이터 블록 주소로 통합할 수 있다.

[0065] 이는 GPU에서 널리 사용되는 메모리 통합 기술과는 차이가 있다. 메모리 통합 기술은 동일한 데이터 블록 주소에 대한 메모리 요청만을 통합한다. 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 프로세스 모듈(130) 내에 바이너리 MAC 유닛과 축적기를 배치하여 데이터 블록D_1과 D_2모두에 대한 MAC 연산을 동시에 실행할 수

있다. 단일 데이터 블록에 대한 8비트 연산은 256개의 이진 MAC 단위 중 64개의 이진 MAC 단위만 필요하므로 프로세스 모듈(130)은 2개의 데이터 블록에 대해 8비트 MAC 연산을 동시에 실행할 수 있다. 따라서, 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 더 높은 컴퓨팅 처리량을 제공하여 시스템 에너지를 절감할 수 있는 이점이 있다. 도 5는 종래와 본 발명의 일 실시예에 따른 신경망에서의 실행 시간과 레이어별 실행 시간을 비교한 결과를 도시한 그래프이다.

[0066] 도 5의 (a)는 신경망에서의 인-메모리 가속기의 실행 시간을 나타낸 것으로, 종래의 16 비트 PIM 시스템에 비해 실행 시간을 19.9 ~ 42.1 %(27.4 ~ 42.1 %) 줄이면서 상대 정확도는 100%(99%) 유지하는 것을 알 수 있다.

[0067] 도 5의 (b)는 GoogLeNet3의 실행 시간을 각 레이어로 나누어 실행 시간을 비교한 것으로, 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 종래의 16 비트 PIM에 비해 컨볼루션 계층 2, 5, 8 및 11에서 실행 시간을 크게 줄이는 것을 알 수 있다. 1% 상대 정확도 손실이 허용되는 경우 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는(99%)은 컨볼루션 계층 4에서 실행 시간을 추가로 줄일 수 있는 것을 알 수 있다. 본 발명의 일 실시예에 따른 인-메모리 가속기(100)가 필요 정밀도가 8 비트 이하인 경우 동시에 두 데이터 블록을 읽으며, 두 데이터 블록에 대한 MAC 작업을 차단하고 실행함으로써, 높은 컴퓨팅 처리량을 제공하여 레이어 별 QNN의 실행 시간을 단축할 수 있다.

[0068] 도 6은 종래와 본 발명의 일 실시예에 따른 신경망 연산에 따른 전력 소비를 비교한 결과를 도시한 그래프이다.

[0069] 도 6의 (a)는 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 종래의 16 비트 PIM 시스템에 비해 전력 소비를 15.4 ~ 18.2%(14.2 ~ 19.6%) 줄이면서 상대 정확도는 100%(99 %) 유지하는 것을 알 수 있다.

[0070] 도 6의 (b)는 GoogLeNet의 전력 소비를 각 계층으로 나눈 것으로, 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 필요한 데이터 비트만 선택적으로 액세스하기 때문에 필요 정밀도가 16 비트 이하인 경우 I / O 전력을 줄일 수 있다. 그러나 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 서로 다른 데이터 블록에서 두 개의 가중치 또는 활성화(총 16 비트)가 함께 전송되기 때문에 컨볼루션 계층 2, 5 및 8에서 전력(에너지 아님) 소비는 줄지 않는 것을 알 수 있다. 본 발명의 일 실시예에 따른 인-메모리 가속기(100)는 I/O 게이트부의 추가 회로에서 추가 전력이 발생하나 전력 오버 헤드는 전체 전력 소비의 2.2%에 불과하며 이는 I/O 전력 감소(최대 19.6 %)보다 훨씬 적은 것을 알 수 있다.

[0071] 도 7은 신경망의 전력 소비를 기반으로 GPU에서 발산되는 열을 고려하여 최고 온칩 온도를 분석한 결과를 나타낸 것으로, 종래의 16 비트 PIM 시스템의 최고 온칩 온도는 모든 신경망에서 88.2 ℃인 반면. 반면 Quant-PIM의 최고 온칩 온도는 GoogLeNet, AlexNet, NiN의 경우 각각 84.8 ℃, 85.4 ℃, 84.9 ℃로, 모든 신경망에서 본 발명의 일 실시예에 따른 인-메모리 가속기는 I / O 전력 소비를 줄이므로 16 비트 PIM 시스템보다 피크 온칩 온도가 낮은 것을 알 수 있다.

[0072] 도 8은 신경망에서 본 발명의 일 실시예에 따른 인-메모리 가속기의 시스템 에너지를 나타낸 그래프이다. 본 발명의 일 실시예에 따른 인-메모리 가속기는 종래의 16 비트 PIM 시스템에 비해 시스템 에너지를 39.1 ~ 50.4%(38.3 ~ 56.4%) 줄이면서 상대 정확도는 100%(99 %) 유지하는 것을 알 수 있다. 가속기 자체는 매우 작은 전력으로 인해 무시할 수 있는 에너지를 발생시키기 때문에, HBM(메모리) 에너지가 전체 시스템 에너지의 대부분을 차지한다. 본 발명의 일 실시예에 따른 인-메모리 가속기는 필요 정밀도가 16비트보다 낮을 때 전력 소비를 줄이며, 필요 정밀도가 8 비트 이하일 때 더 높은 컴퓨팅 처리량을 제공하여 레이어별 QNN의 실행 시간이 짧아지도록 할 수 있다. 이로 인해, 본 발명의 일 실시예에 따른 인-메모리 가속기는 종래에 비해 HBM의 동적 및 누설 에너지를 크게 줄일 수 있다.

[0074] 본 발명의 실시 예에 따른 장치 및 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 컴퓨터 판독 가능 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야 통상의 기술자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media) 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행

될 수 있는 고급 언어 코드를 포함한다.

[0075] 상술한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0076] 이제까지 본 발명에 대하여 그 실시 예들을 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시 예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

부호의 설명

[0078] 100: 인-메모리 가속기

110: 베이스 다이부

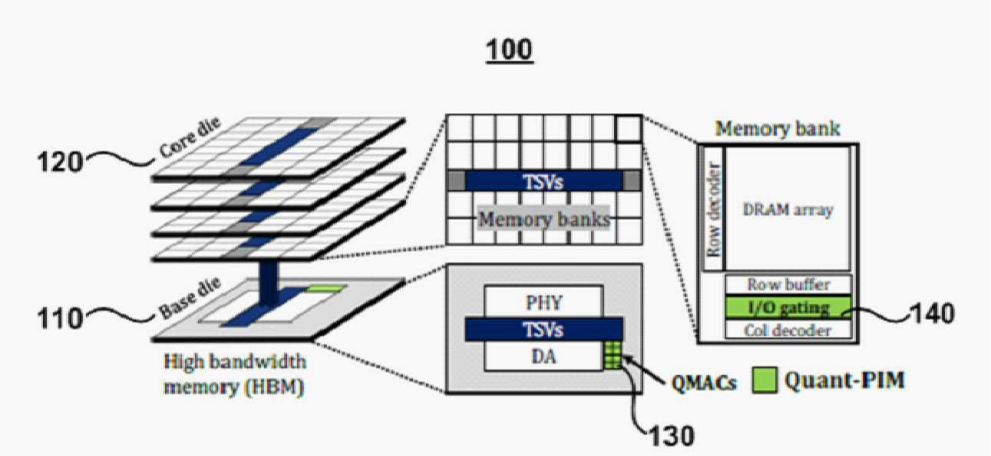
120: 코어 다이부

130: 프로세스 모듈

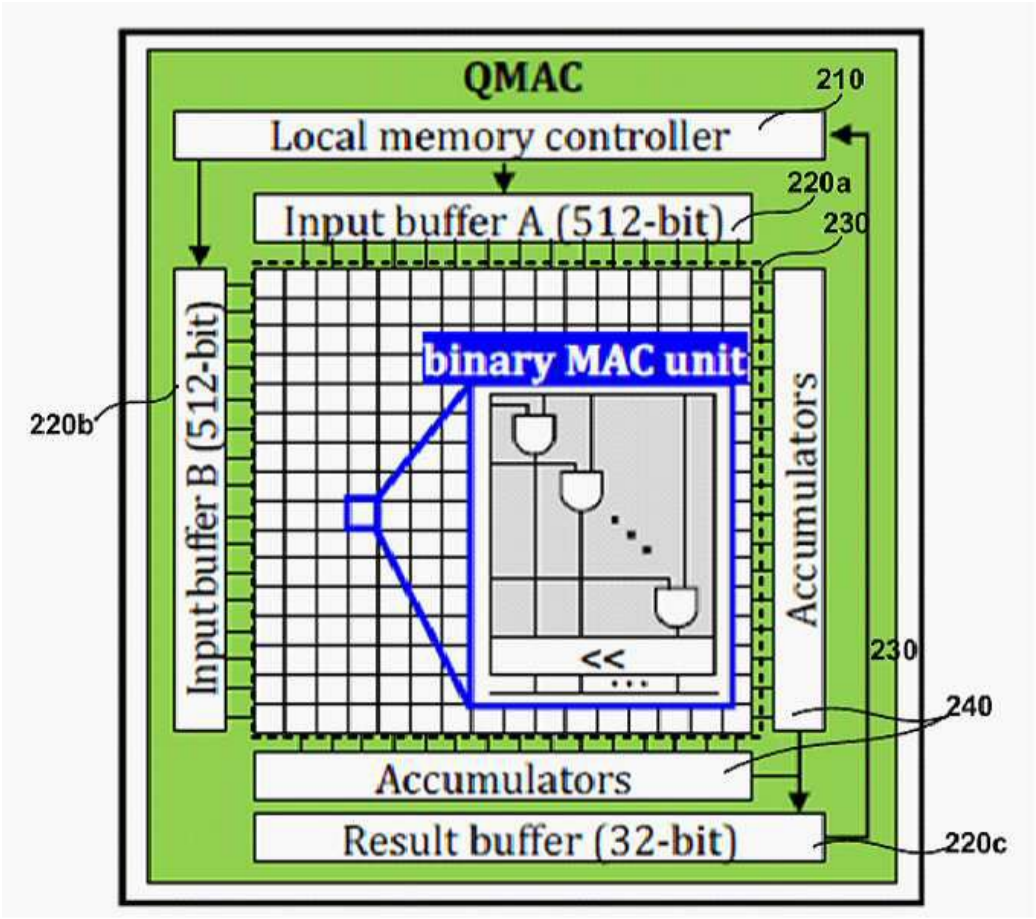
140: I/O 게이트부

도면

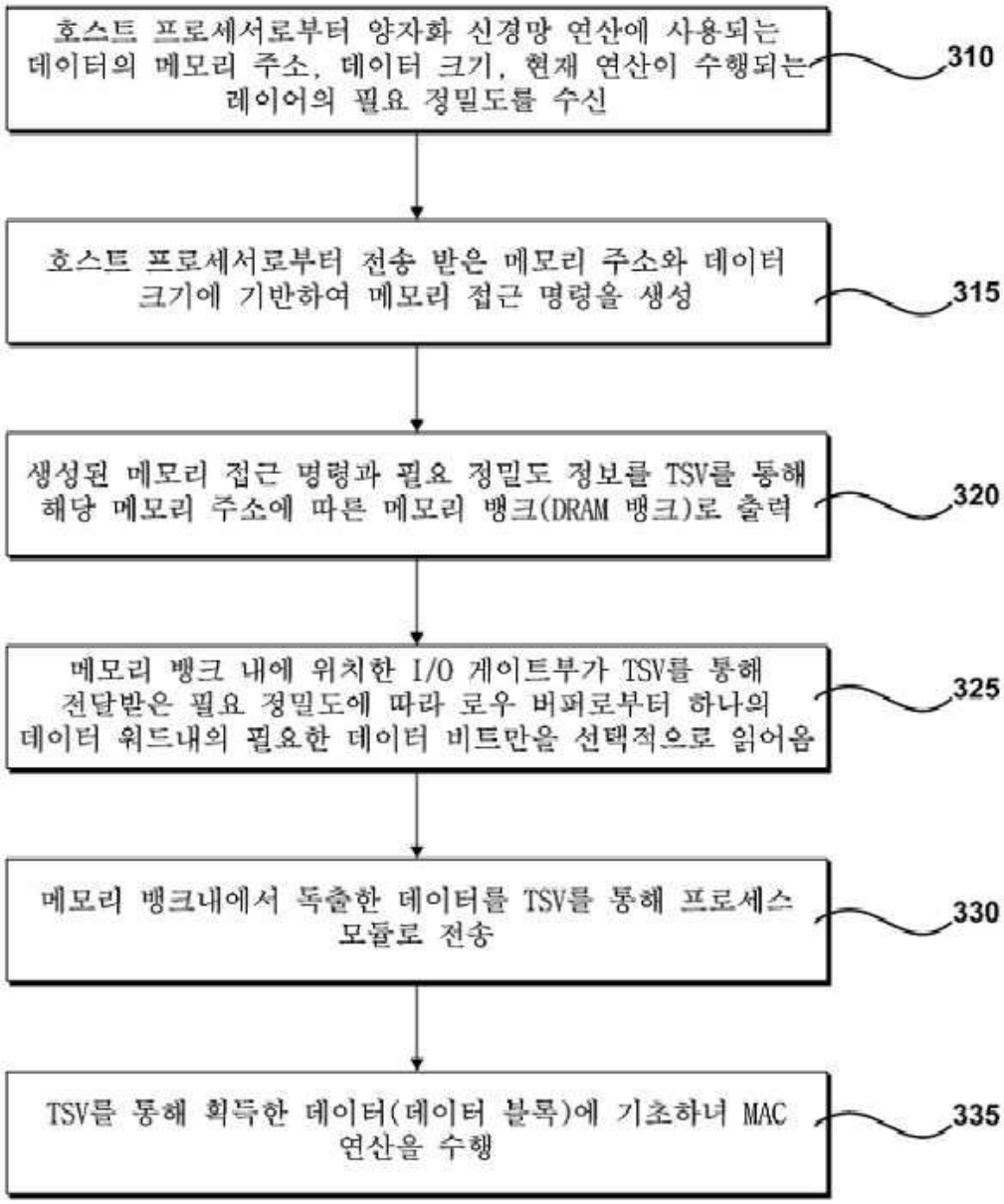
도면1



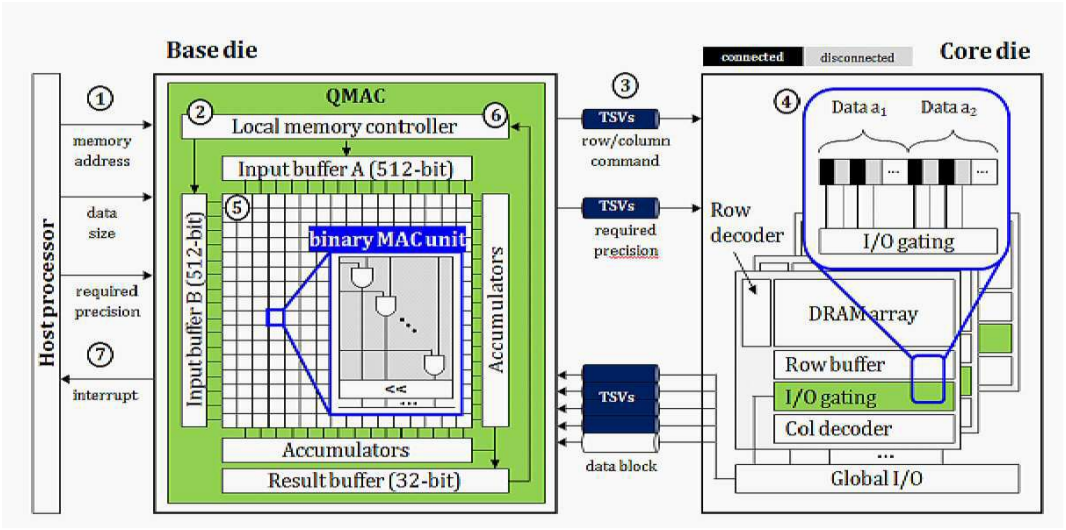
도면2


QMAC
210
Local memory controller
220a
Input buffer A (512-bit)
230
binary MAC unit
220b
Input buffer B (512-bit)
Accumulators
230
240
Accumulators
Result buffer (32-bit)
220c

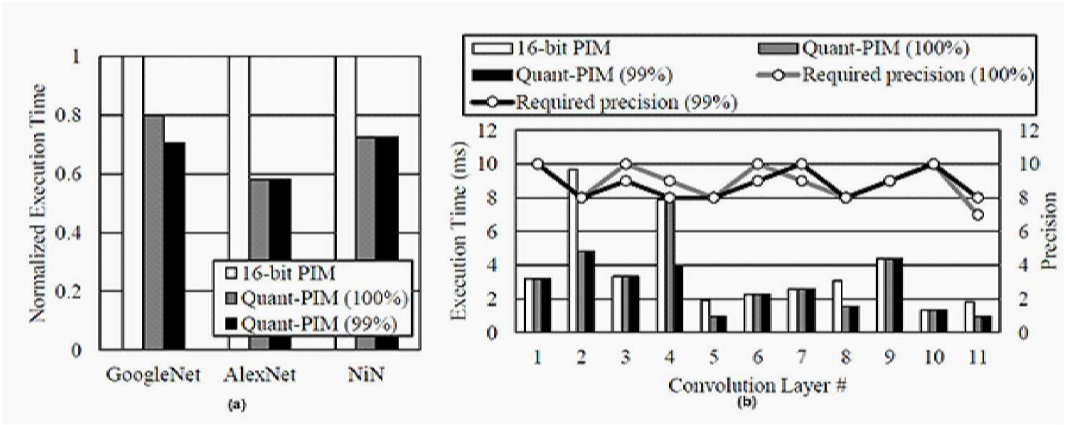
도면3


호스트 프로세서로부터 양자화 신경망 연산에 사용되는 데이터의 메모리 주소, 데이터 크기, 현재 연산이 수행되는 레이어의 필요 정밀도를 수신
310
호스트 프로세서로부터 전송 받은 메모리 주소와 데이터 크기에 기반하여 메모리 접근 명령을 생성
315
생성된 메모리 접근 명령과 필요 정밀도 정보를 TSV를 통해 해당 메모리 주소에 따른 메모리 뱅크(DRAM 뱅크)로 출력
320
메모리 뱅크 내에 위치한 I/O 게이트부가 TSV를 통해 전달받은 필요 정밀도에 따라 로우 버퍼로부터 하나의 데이터 워드내의 필요한 데이터 비트만을 선택적으로 읽어옴
325
메모리 뱅크내에서 독출한 데이터를 TSV를 통해 프로세스 모듈로 전송
330
TSV를 통해 획득한 데이터(데이터 블록)에 기초하여 MAC 연산을 수행
335

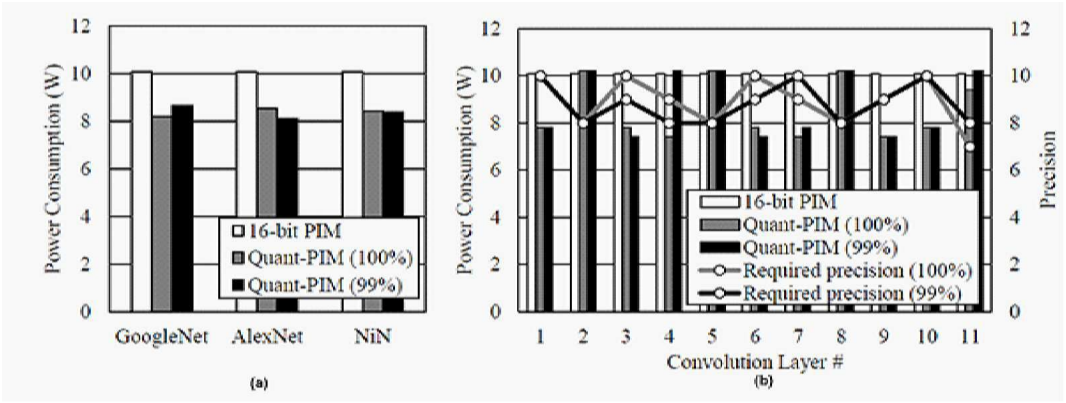
도면4


Base die
Core die
connected
disconnected
Host processor
memory address
data size
required precision
interrupt
QMAC
Local memory controller
Input buffer A (512-bit)
Input buffer B (512-bit)
binary MAC unit
Accumulators
Result buffer (32-bit)
TSVs
row/column command
required precision
data block
Data a1
Data a2
I/O gating
Row decoder
DRAM array
Row buffer
Col decoder
Global I/O

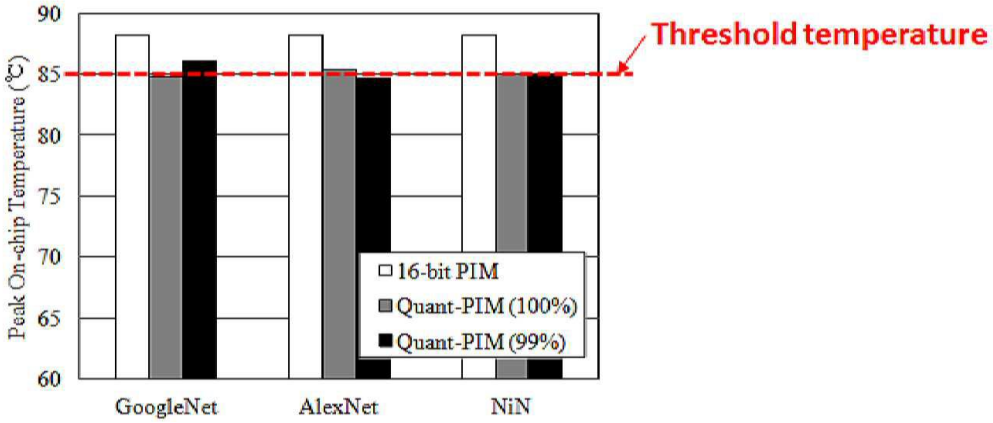
도면5


Normalized Execution Time
16-bit PIM
Quant-PIM (100%)
Quant-PIM (99%)
GoogleNet
AlexNet
NiN
(a)
Execution Time (ms)
Precision
Required precision (100%)
Required precision (99%)
Convolution Layer #
(b)

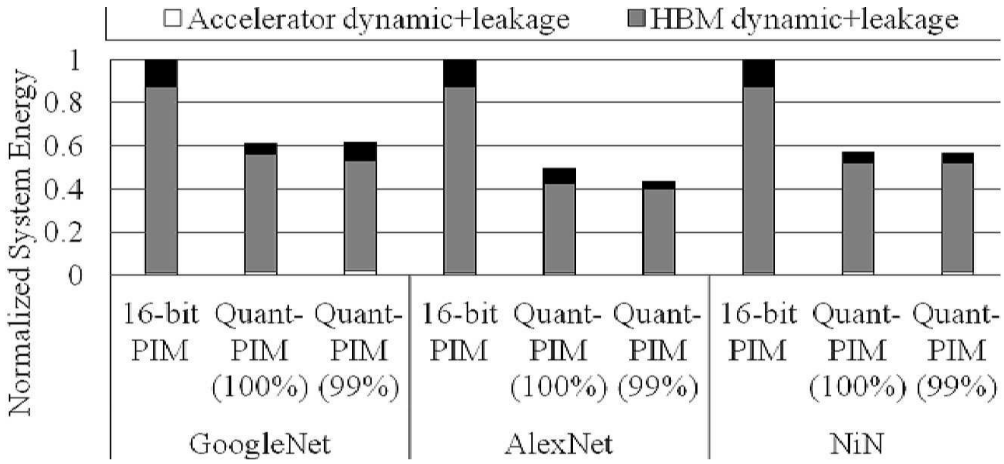
도면6


Power Consumption (W)
16-bit PIM
Quant-PIM (100%)
Quant-PIM (99%)
GoogleNet
AlexNet
NiN
(a)
Precision
Required precision (100%)
Required precision (99%)
Convolution Layer #
(b)

도면7


Threshold temperature
Peak On-chip Temperature (℃)
90
85
80
75
70
65
60
16-bit PIM
Quant-PIM (100%)
Quant-PIM (99%)
GoogleNet
AlexNet
NiN

도면8


Accelerator dynamic+leakage
HBM dynamic+leakage
Normalized System Energy
1
0.8
0.6
0.4
0.2
0
16-bit PIM
Quant-PIM (100%)
Quant-PIM (99%)
GoogleNet
AlexNet
NiN