(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0017729
(43) 공개일자 2024년02월08일

(51) 국제특허분류(Int. Cl.)
G06F 15/78 (2006.01) *G06F 17/16* (2006.01)
G06F 7/533 (2006.01) *G06F 7/57* (2006.01)
G06F 9/30 (2018.01)
(52) CPC특허분류
G06F 15/7821 (2013.01)
G06F 17/16 (2013.01)
(21) 출원번호 10-2022-0168043
(22) 출원일자 2022년12월05일
심사청구일자 2022년12월05일
(30) 우선권주장
1020220095665 2022년08월01일 대한민국(KR)

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
김선욱
경기도 남양주시 도농로 34(도농동, 부영그린타운)
백윤아
서울특별시 성북구 안암로9가길 68(안암동5가, 인하우스)
(뒷면에 계속)
(74) 대리인
특허법인주연케이알피

전체 청구항 수 : 총 10 항

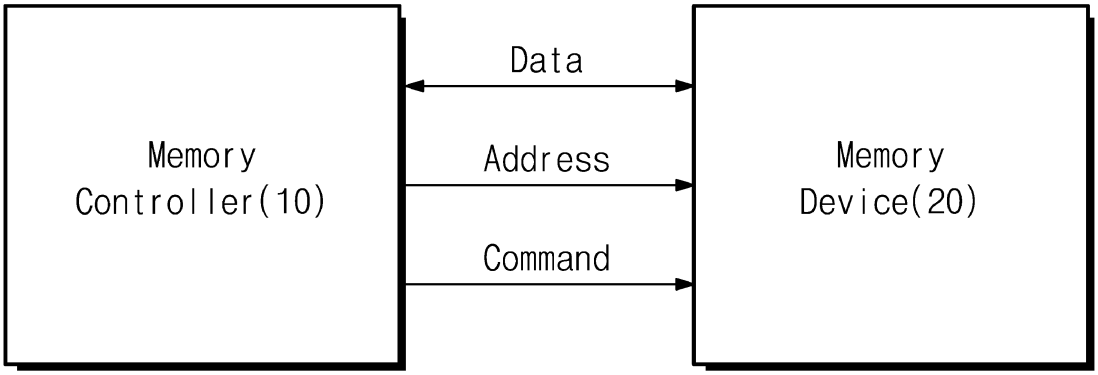
(54) 발명의 명칭 **연산 정확도가 보장된 인메모리 연산 장치 및 동작 방법**

(57) 요 약

본 발명은 인메모리 연산 아키텍처에서 메모리 뱅크 간 명령어의 순서가 바뀌는 경우에도 연산기에 입력되는 피연산자를 정확하게 지정함으로써 연산 결과의 정확도를 보장하는 인메모리 연산 장치 및 방법에 관한 것이다. 본 출원의 실시 예에 따른 메모리 장치는, 인메모리 연산을 수행하는 메모리 장치로서, 상기 입력신호를 수신하는

(뒷면에 계속)

대 표 도 - 도1



(1)

PIM IU(Processing In Memory Interface Unit); 및 상기 입력신호를 수신하고, 복수의 제1 단위 데이터를 포함하는 제1 피연산 데이터 및 복수의 제2 단위 데이터를 포함하는 제2 피연산 데이터에 기초하여 상기 인메모리 연산을 수행하는 복수의 메모리 뱅크를 포함하며, 상기 PIM IU는 상기 어드레스 신호 중 상기 제1 피연산 데이터의 인덱스 정보를 지시하는 일부 비트를 상기 복수의 메모리 뱅크 각각으로 전달하고, 상기 복수의 메모리 뱅크 각각은 상기 제1 피연산 데이터에 포함된 복수의 제1 단위 데이터를 순차적으로 읽어와서 상기 제2 피연산 데이터와 연산하되, 상기 제2 피연산 데이터의 인덱스를 상기 일부 비트를 참조하여 결정한다.

(52) CPC특허분류
G06F 7/5334 (2013.01)
G06F 7/57 (2021.08)
G06F 9/3004 (2013.01)
G06F 9/3012 (2013.01)

(72) 발명자
김창현
경기도 성남시 분당구 미금일로 136(구미동, 까치마을건영빌라)
이원준
대구광역시 동구 송라로14길 132(신천동, 신천청아람)
이주한
서울특별시 성북구 안암로9가길 25(안암동4가)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711160651
과제번호	2022-0-00441-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	PIM인공지능반도체핵심기술개발(설계)
연구과제명	재구성형 PIM 디바이스 기반의 Memory-Centric 아키텍처 개발
기 여 율	1/1
과제수행기관명	고려대학교산학협력단
연구기간	2022.04.01 ~ 2022.12.31

명 세 서

청구범위

청구항 1

인메모리 연산을 수행하는 메모리 장치로서,

메모리 컨트롤러에서 출력하는 데이터(Data), 커맨드(Command) 및 어드레스(Address) 신호를 포함하는 입력신호를 입력 받는 입출력단;

상기 입력신호를 수신하는 PIM IU(Processing In Memory Interface Unit); 및

상기 입력신호를 수신하고, 복수의 제1 단위 데이터를 포함하는 제1 피연산 데이터 및 복수의 제2 단위 데이터를 포함하는 제2 피연산 데이터에 기초하여 상기 인메모리 연산을 수행하는 복수의 메모리 뱅크를 포함하며,

상기 PIM IU는 상기 어드레스 신호 중 상기 제1 피연산 데이터의 인덱스 정보를 지시하는 일부 비트를 상기 복수의 메모리 뱅크 각각으로 전달하고,

상기 복수의 메모리 뱅크 각각은 상기 제1 피연산 데이터에 포함된 복수의 제1 단위 데이터를 순차적으로 읽어와서 상기 제2 피연산 데이터와 연산하되, 상기 제2 피연산 데이터의 인덱스를 상기 일부 비트를 참조하여 결정하는 메모리 장치.

청구항 2

제1 항에 있어서,

상기 복수의 메모리 뱅크 각각은 PIM 연산 유닛을 포함하고,

상기 PIM 연산 유닛은:

상기 복수의 제1 단위 데이터를 저장하는 제1 레지스터 및 상기 복수의 제2 단위 데이터 중 하나를 저장하는 제2 레지스터;

상기 제1 레지스터 및 상기 제2 레지스터로부터 상기 복수의 제1 단위 데이터 및 상기 복수의 제2 단위 데이터 중 하나를 전달받아 연산하고 연산 결과를 생성하는 연산기; 및

상기 연산기로부터 상기 연산 결과를 전달받아 저장하는 제3 레지스터를 포함하는 메모리 장치.

청구항 3

제2 항에 있어서,

상기 복수의 제1 단위 데이터 각각은 복수의 제1 요소 데이터를 포함하고,

상기 복수의 제2 단위 데이터 각각은 복수의 제2 요소 데이터를 포함하며,

상기 복수의 메모리 뱅크 각각은 순차적인 복수의 시간 구간 마다 상기 복수의 제2 요소 데이터 중 하나를 상기 복수의 제1 요소 데이터와 순차적으로 연산하는 메모리 장치.

청구항 4

제3 항에 있어서,

상기 복수의 시간 구간 각각에 해당하는 제1 단위 데이터의 순서가 상기 복수의 시간 구간 각각의 순서와 불일치할 때,

상기 PIM IU는 상기 복수의 시간 구간 중 임의의 n번째(n은 0 이상의 정수) 시간 구간에 해당하는 메모리 어드레스 신호의 일부 비트를 상기 PIM 연산 유닛으로 전달하고,

상기 PIM 연산 유닛은 상기 일부 비트를 참조하여 상기 제2 레지스터의 데이터 중 상기 연산기에 입력할 제2 요소 데이터를 결정하는 메모리 장치.

청구항 5

제1 항에 있어서,

상기 복수의 메모리 뱅크 각각은 복수의 메모리 셀 어레이를 포함하고,

상기 어드레스 신호는 상기 복수의 메모리 셀 어레이 각각에 대한 주소 정보, 상기 복수의 메모리 뱅크 각각에 대한 주소 정보 및 오프셋 정보를 포함하고,

상기 일부 비트는 상기 오프셋 정보에 대응되는 비트로부터 k번째(여기서, k는 자연수임) 비트까지의 비트 구간을 포함하는 메모리 장치.

청구항 6

인메모리 연산을 수행하는 메모리 장치의 동작 방법으로서,

메모리 컨트롤러에서 출력하는 데이터, 커맨드 및 어드레스 신호를 포함하는 입력신호를 입력 받는 단계;

상기 어드레스 신호 중 제1 피연산 데이터의 인덱스 정보를 지시하는 일부 비트를 복수의 메모리 뱅크 각각으로 전달하는 단계; 및

상기 제1 피연산 데이터에 포함된 복수의 제1 단위 데이터를 순차적으로 읽어와서 제2 피연산 데이터와 연산하되, 상기 제2 피연산 데이터의 인덱스를 상기 일부 비트를 참조하여 결정하는 단계; 및

상기 복수의 제1 단위 데이터를 포함하는 상기 제1 피연산 데이터 및 복수의 제2 단위 데이터를 포함하는 상기 제2 피연산 데이터에 기초하여 상기 인메모리 연산을 수행하는 단계를 포함하는 메모리 장치 동작 방법.

청구항 7

제6 항에 있어서,

제1 레지스터 및 제2 레지스터로부터 복수의 제1 단위 데이터 및 복수의 제2 단위 데이터 중 하나를 전달받아 연산하고,

제3 레지스터가 연산 결과를 전달받아 저장하는 단계를 포함하는 메모리 장치 동작 방법.

청구항 8

제7 항에 있어서,

상기 복수의 제1 단위 데이터 각각은 복수의 제1 요소 데이터를 포함하고,

상기 복수의 제2 단위 데이터 각각은 복수의 제2 요소 데이터를 포함하며,

복수의 메모리 뱅크 각각에서 순차적인 복수의 시간 구간 마다 상기 복수의 제2 요소 데이터 중 하나를 상기 복수의 제1 요소 데이터와 순차적으로 연산하는 단계를 포함하는 메모리 장치 동작 방법.

청구항 9

제8 항에 있어서,

상기 복수의 시간 구간 각각에 해당하는 제1 단위 데이터의 순서가 상기 복수의 시간 구간 각각의 순서와 불일치할 때,

임의의 n번째(n은 0 이상의 정수) 시간 구간에 해당하는 메모리 어드레스 신호의 일부 비트를 PIM 연산 유닛으로 전달하고,

상기 PIM 연산 유닛이 상기 일부 비트를 참조하여 상기 PIM 연산 유닛의 제2 레지스터의 데이터 중 연산기에 입력할 제2 요소 데이터를 결정하는 단계를 포함하는 메모리 장치 동작 방법.

청구항 10

제6 항에 있어서,

상기 어드레스 신호는 상기 복수의 메모리 뱅크 각각에 포함되는 복수의 메모리 셀 어레이 각각에 대한 주소 정

보, 상기 복수의 메모리 뱅크 각각에 대한 주소 정보 및 오프셋 정보를 포함하고,

상기 일부 비트는 상기 오프셋 정보에 대응되는 비트로부터 k번째(여기서, k는 자연수임) 비트까지의 비트 구간을 포함하는 메모리 장치 동작 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 연산 정확도가 보장된 인메모리 연산 장치 및 동작 방법에 관한 것이다.

배 경 기 술

[0002] 딥러닝(deep learning), 인공신경망(neural network) 등의 최신 기술이 적용된 응용프로그램(application)에서 사용하는 데이터는 용량이 매우 크고 지역성(locality)이 낮다. 전통적인 컴퓨팅 구조인 폰 노이만(von Nemann) 컴퓨팅 아키텍처는 연산기가 포함된 CPU(Central Processing Unit)와 연산에 필요한 데이터를 저장하는 주메모리(main memory)로 구성되어 지역성이 높은 데이터 연산에 최적화 되어있다. 따라서 기존의 컴퓨팅 아키텍처에서 위와 같은 응용프로그램을 실행할 경우 불필요한 데이터 이동과 메모리 병목현상(bottleneck)이 발생한다. 즉, 재사용되는 데이터에 빠르게 접근하기 위해 사용되는 캐시에 지역성이 낮은 데이터가 저장되어 캐시를 효율적으로 사용하지 못하고, 연산에 필요한 방대한 데이터를 대부분 캐시가 아닌, 주메모리로부터 읽어와야 하는 문제가 발생한다. 이러한 문제점을 해결하기 위해 메모리 장치 내부에 간단한 연산기를 장착하여 주메모리에서 연산을 수행하고, 연산 결과만을 CPU로 전송하는 인메모리 연산이 개발되었다.

[0003] 인메모리 연산 아키텍처는 메모리의 각 뱅크 내부의 레지스터 및 연산기로 구성된다. 연산의 실행 순서 예시는 다음과 같다. 연산에 필요한 피연산자를 준비하기 위해, 메모리 셀 또는 메모리 외부에서 하나 또는 그 이상의 피연산자를 읽어와 뱅크 내부 레지스터에 저장하고, 각 뱅크의 메모리 셀에서 나머지 피연산자를 읽어온다. 각 뱅크에서 데이터를 읽어오면 바로 연산(덧셈, 뺄셈, 곱셈 등)을 수행한다. 메모리 셀에서 데이터를 읽어와 연산을 시작하는 과정은 메모리의 RD(read) 명령어를 이용해 수행한다. 연산의 종류는 아키텍처 설계에 따라 미리 설정되어 있거나 피연산자를 읽기 위한 메모리 명령어의 일부 비트로 지정된다.

[0004] 인메모리 연산을 수행할 때 사용되는 메모리 명령어는 메모리 컨트롤러의 스케줄링에 의해 생성되며, 메모리 컨트롤러의 스케줄링 알고리즘에 의해 메모리 뱅크 간 명령어의 순서가 바뀔 수 있다. 이때 메모리 뱅크 간 명령어 순서가 바뀌는 것을 고려하지 않는다면 인메모리 연산기에 입력되는 피연산자가 잘못 지정되어 연산 결과의 정확도가 보장되지 않는 문제가 발생한다.

발명의 내용

해결하려는 과제

[0005] 본 발명의 목적은 인메모리 연산 아키텍처에서 메모리 뱅크 간 명령어의 순서가 바뀌는 경우에도 연산기에 입력되는 피연산자를 정확하게 지정함으로써 연산 결과의 정확도를 보장하는 인메모리 연산 장치 및 방법을 제공하는 것이다.

과제의 해결 수단

[0006] 본 출원의 일 실시예에 따른 인메모리 연산장치는, 메모리 컨트롤러에서 출력하는 데이터(Data), 커맨드(Command) 및 어드레스(Address) 신호를 포함하는 입력신호를 입력 받는 입출력단; 상기 입력신호를 수신하는 PIM IU(Processing In Memory Interface Unit); 및 상기 입력신호를 수신하고, 복수의 제1 단위 데이터를 포함하는 제1 피연산 데이터 및 복수의 제2 단위 데이터를 포함하는 제2 피연산 데이터에 기초하여 상기 인메모리 연산을 수행하는 복수의 메모리 뱅크를 포함하며, 상기 PIM IU는 상기 어드레스 신호 중 상기 제1 피연산 데이터의 인덱스 정보를 지시하는 일부 비트를 상기 복수의 메모리 뱅크 각각으로 전달하고, 상기 복수의 메모리 뱅크 각각은 상기 제1 피연산 데이터에 포함된 복수의 제1 단위 데이터를 순차적으로 읽어와서 상기 제2 피연산 데이터와 연산하되, 상기 제2 피연산 데이터의 인덱스를 상기 일부 비트를 참조하여 결정한다.

[0007] 본 출원의 일실시예에 따른 인메모리 연산 장치의 동작 방법은, 메모리 컨트롤러에서 출력하는 데이터, 커맨드 및 어드레스 신호를 포함하는 입력신호를 입력 받는 단계; 상기 어드레스 신호 중 제1 피연산 데이터의 인덱스 정보를 지시하는 일부 비트를 복수의 메모리 뱅크 각각으로 전달하는 단계; 및 상기 제1 피연산 데이터에 포함

된 복수의 제1 단위 데이터를 순차적으로 읽어와서 제2 피연산 데이터와 연산하되, 상기 제2 피연산 데이터의 인덱스를 상기 일부 비트를 참조하여 결정하는 단계; 및 상기 복수의 제1 단위 데이터를 포함하는 상기 제1 피연산 데이터 및 복수의 제2 단위 데이터를 포함하는 상기 제2 피연산 데이터에 기초하여 상기 인메모리 연산을 수행하는 단계를 포함한다.

발명의 효과

[0008] 본 출원의 실시예에 따른 인메모리 연산 장치는 메모리 요청에 포함된 피연산자의 메모리 주소의 일부 비트를 사용하여 인메모리 연산 장치에 입력되는 피연산자를 정확하게 지정함으로써 인메모리 연산 아키텍처에서 표준 메모리 요청으로 연산을 실행할 때, 상용 메모리 컨트롤러의 스케줄링에 의해 메모리 뱅크 간 요청의 순서가 바뀌어도 연산 결과의 정확도를 보장한다.

도면의 간단한 설명

[0009] 도 1은 본 출원의 일 실시 예에 따른 메모리 시스템의 블록도이다.

도 2는 본 출원의 일 실시예에 따른 도 1의 메모리 장치를 세부적으로 나타낸 도면이다.

도 3은 본 출원의 일 실시예에 따른 도 2의 PIM IU와 메모리 뱅크를 상세히 도시한 도면이다.

도 4는 본 출원의 일 실시예에 따른 행렬 곱 연산을 도시한 것이다.

도 5는 본 출원의 일 실시예에 따른 각 메모리 뱅크의 연산 실행 동작을 도시한 것이다.

도 6은 연산 유닛에서 어드레스 신호의 일부 비트를 참조하여 연산기에 제2 입력을 결정하는 모습을 예시적으로 나타낸 도면이다.

도 7은 어드레스 신호의 각 비트가 메모리 장치에 맵핑되는 예시적인 모습이다.

도 8은 시간 구간에서의 명령어가 바뀌었을 때 본 출원의 일 실시예에 따른 메모리 장치의 동작을 나타내는 도면이다.

도 9는 본 출원의 실시예에 따른 메모리 장치의 동작 방법의 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0010] 이하에서는, 본 발명에 따른 구현들을 첨부된 도면을 참조하여 상세하게 설명한다.

[0011] 도 1은 본 출원의 일 실시 예에 따른 메모리 시스템의 블록도이다.

[0012] 도 1을 참조하면 본 출원의 일 실시예에 따른 메모리 시스템은 메모리 컨트롤러(Memory Controller)(10) 및 메모리 장치(Memory Device)(20)를 포함한다.

[0013] 메모리 컨트롤러(10)는 메모리 장치(20)와 전기적으로 연결되어, 메모리 장치(20)에 데이터(Data)를 주고 받거나, 커맨드(Command) 및 어드레스(Address) 신호를 전송 하여 메모리 장치(20)에서 수행하는 연산의 종류 및 피연산자를 컨트롤할 수 있다.

[0014] 메모리 장치(20)는 메모리 컨트롤러(10)로부터 출력되는 데이터, 커맨드, 및 어드레스 신호에 기반하여 인메모리 연산을 수행할 수 있다. 본 출원에서 개시하는 메모리 컨트롤러(10)는 메모리 장치(20)의 종류에 따라 다양하게 구현될 수 있으며, 일 실시예에 따라 메모리 장치(20)는 DRAM(Dynamic RAM)을 포함할 수 있고, 메모리 컨트롤러(10)는 DRAM 컨트롤러로 구현될 수 있다.

[0015] 도 2는 본 출원의 일 실시예에 따른 도 1의 메모리 장치를 세부적으로 나타낸 도면이다.

[0016] 도 2를 참조하면, 메모리 장치(20)는 입출력단(100), PIM IU(Processing In Memory Interface Unit, 200) 및 메모리 뱅크(Memory Bank, 300)를 포함한다.

[0017] 입출력단(100)은 메모리 컨트롤러(10)로부터 데이터, 커맨드 및 어드레스 신호를 포함하는 입력 신호를 입력 받는다. 입력 신호는 표준 메모리 요청 신호(Standard memory request)와, PIM IU(200)로 전달되는 신호를 포함할 수 있다.

[0018] 입출력단(100)은 데이터 입출력을 위한 입출력 회로(Data I/O Circuit, 110)를 포함할 수 있다. 입출력 회로

(110)는 입력 받은 데이터를 메모리 뱅크(300) 및/또는 PIM IU(200)로 전달하거나, 또는 메모리 뱅크(300) 로부터의 데이터를 메모리 컨트롤러(10)에 출력할 수 있다.

[0019] 또는, 입출력단(100)은 입력 신호를 입력 받고 각각에 대하여 래치 신호를 생성하기 위한 어드레스 래치(Address Latch, 120) 및 커맨드 래치(Command Latch, 130)를 포함할 수 있다.

[0020] 어드레스 래치(120)는 어드레스에 기초하여 래치 신호를 생성하고, 래치 신호를 메모리 뱅크(300) 및/또는 PIM IU(200)에 전달한다.

[0021] 커맨드 래치(130)는 커맨드 디코더(Command Decoder, 140)와 연결될 수 있고, 커맨드 디코더(140)는 커맨드 래치(130)로부터의 래치 신호를 디코딩하여 디코딩 결과를 메모리 뱅크(300) 및/또는 PIM IU(200)에 전달한다.

[0022] PIM IU(200)는 입력 신호를 전달받고, 입력 신호에 기초하여 인메모리 연산 요청을 위한 신호를 생성 및 출력한다. 여기서, 인메모리 연산 요청을 위한 신호는 PIM 커맨드 신호(PIM command) 및 PIM 유효 신호(PIM valid)를 포함한다. PIM 커맨드 신호 및 PIM 유효 신호는 메모리 뱅크(300)로 전달된다. PIM IU(200)에 대한 상세한 설명은 후술한다.

[0023] 메모리 뱅크(300)는 PIM 커맨드 신호 및 PIM 유효 신호에 기초하여 피연산 데이터를 연산한다. 메모리 뱅크(300)는 복수 개가 포함될 수 있다. 복수 개의 메모리 뱅크(300)는 공유 버스(shared bus, 400)에 의해 연결될 수 있다.

[0024] 이하에서는, 메모리 뱅크(300)에 대하여 구체적으로 설명한다. 본 개시에서 메모리 뱅크(300)가 수행하는 일련의 연산은 피연산자 및 연산자에 기초하여 수행되는 것으로 가정하며, 연산자는 덧셈, 뺄셈, 곱셈 등 다양한 종류의 연산자들을 포함할 수 있다. 또한, 피연산자는 각 뱅크가 기 저장하고 있는 피연산자 및 모든 뱅크에 순차적으로 브로드캐스팅(broadcasting)되는 피연산자를 포함할 수 있다. 본 개시에서는 편의상 각 뱅크가 기 저장하고 있는 피연산자를 제2 피연산 데이터로 칭하고, 모든 뱅크에 순차적으로 브로드캐스팅 되는 피연산자를 제1 피연산 데이터로 칭한다.

[0025] 제1 피연산 데이터는 복수의 제1 단위 데이터를 포함하고, 제2 피연산 데이터는 복수의 제2 단위 데이터를 포함할 수 있다. 여기서, “단위”라는 용어는 각 피연산 데이터가 복수의 하위 요소 데이터를 포함할 때, 복수의 하위 요소 데이터가 논리적, 수학적 기준에 따라 그룹핑될 수 있을 때, 각 그룹을 칭하는 용어일 수 있다.

[0026] 예를 들어, 각 피연산 데이터가 행렬인 경우, 단위 데이터는 행(row) 또는 열(column)일 수 있고, 요소 데이터는 각 성분(element)일 수 있다.

[0027] 제1 피연산 데이터는 복수의 메모리 뱅크(300) 중 어느 하나의 메모리 셀 어레이(310)에 저장되어 있을 수 있고, 이 경우 제1 피연산 데이터는 복수의 메모리 뱅크(300) 중 어느 하나의 메모리 셀 어레이(310)로부터 읽어져 와서 나머지 메모리 뱅크(300)들에 브로드캐스팅 될 수 있다.

[0028] 도 3은 본 출원의 일 실시예에 따른 도 2의 PIM IU와 메모리 뱅크를 상세히 도시한 도면이다.

[0029] 도 3을 참조하면, PIM IU(200)는 상술한 동작을 수행하기 위하여 주소 매칭기(210) 및 PIM 커맨드 디코더(220)를 포함한다.

[0030] 주소 매칭기(210)는 입출력단으로부터 주소 정보(예, 어드레스의 래치 신호)를 전달받고, 주소 정보와 기 저장된 브로드캐스팅 주소 정보를 매칭하여 PIM 유효 신호를 생성한다. 즉, 주소 매칭기(210)는 주소 정보가 브로드캐스팅 해야 할 주소인지 여부를 판단한다.

[0031] PIM 커맨드 디코더(220)는 연산기의 연산 종류를 지시하는 PIM 커맨드 신호를 생성한다. PIM 커맨드 디코더(220)는 PIM 커맨드 신호 생성을 위해 opcode를 저장하고 있으며, 기 저장된 opcode와 입력 신호(예, 커맨드)에 기초하여 PIM 커맨드 신호를 생성한다.

[0032] 메모리 뱅크(300)는 메모리 셀 어레이(Memory cell array, 310), PIM 연산 유닛(PIM engine, 320) 및 스위치(Switch, 340)를 포함한다.

[0033] 메모리 셀 어레이(310)는 제2 피연산 데이터에 포함된 복수의 제2 단위 데이터 중 하나를 저장한다. 메모리 뱅크(300)가 복수 개가 구비될 경우, 각 메모리 셀 어레이(310)는 복수의 제2 단위 데이터 중 하나를 서로 중복되지 않도록 저장할 수 있다. 예를 들어, 제2 피연산 데이터가 특정 행렬이고, 제2 단위 데이터가 특정 행렬의 열인 경우, 각 메모리 뱅크(300)에 포함된 메모리 셀 어레이(310)는 서로 다른 열을 저장하고 있을 수 있다.

[0034] PIM 연산 유닛(320)은 메모리 셀 어레이(310)로부터 제2 피연산 데이터에 포함된 복수의 제2 단위 데이터 중 하나를 읽고, 제1 피연산 데이터에 포함된 복수의 제1 단위 데이터를 순차적으로 읽고 브로드캐스팅하면서 복수의 제2 단위 데이터 중 하나와 연산하여 연산 결과를 생성한다.

[0035] PIM 연산 유닛(320)은 연산 실행을 위하여 제1 레지스터(Register 1, 321), 제2 레지스터(Register 2, 322), 연산기(Vector ALU, 323) 및 제3 레지스터(Register 3, 324)를 포함한다.

[0036] 제2 레지스터(322)는 복수의 제2 단위 데이터를 저장한다. 제2 레지스터(322)는 메모리 셀 어레이(310)에 저장되어 있던 복수의 제2 단위 데이터를 레지스터 파일 버스 인터페이스(330)를 통해 읽어와서 저장한다. 일 실시예로, 제2 레지스터(322)는 어드레스 신호의 일부 비트를 참조하여 연산기(323)에 입력할 요소를 결정한다.

[0037] 제1 레지스터(321)는 브로드캐스팅 되는 복수의 제1 단위 데이터 중 하나를 레지스터 파일 버스 인터페이스(330)를 통해 읽어와서 저장한다.

[0038] 연산기(323)는 제1 레지스터(321) 및 제2 레지스터(322)로부터 복수의 제1 단위 데이터 및 복수의 제2 단위 데이터 중 하나를 전달받아 연산하고 연산 결과를 생성한다. 연산기(323)는 예를 들어 ALU(Arithmetic Logic Unit)일 수 있고, 여러 개가 구비될 수 있다.

[0039] 연산기(323)는 PIM 커맨드 신호에 기초하여 연산자, 즉 연산의 종류를 결정할 수 있다. 연산기(323)는 복수의 제1 단위 데이터가 순차적으로 제1 레지스터(321)에 저장될 때마다 연산을 수행할 수 있다.

[0040] 제3 레지스터(324)는 연산기(323)로부터 연산 결과를 전달받아 누적 및 저장한다.

[0041] 스위치(340)는 PIM 유효 신호에 기초하여 PIM 연산 유닛(320) 및 공유 버스(400)와의 전기적 연결을 스위칭한다. 예를 들어, 스위치(340)는 PIM 유효 신호가 주소 정보와 기 저장된 브로드캐스팅 주소 정보가 매칭됨을 지시하는 경우, 공유 버스(400)와 PIM 연산 유닛(320)을 연결하고, 이때 제1 레지스터(321)는 순차적으로 브로드캐스팅 되는 복수의 제1 단위 데이터를 저장할 수 있다.

[0042] 이하에서는, 도 4 내지 도 5를 참조하여 상술한 메모리 뱅크(300)의 구체적인 연산 실행 동작에 대하여 설명한다. 연산 실행 동작은 예시적으로 행렬 곱 연산에 대하여 설명한다.

[0043] 도 4는 본 출원의 일 실시예에 따른 행렬 곱 연산을 도시한 것이다.

[0044] 도 4를 참조하면, 행렬 곱 연산은 두 개의 행렬을 곱하여 결과 행렬을 생성한다. 여기서, 곱은 연산자이고, 곱셈의 대상이 되는 두 개의 행렬 중 A 행렬(32 x 32)은 제1 피연산 데이터이고, B 행렬(32 x 16)은 제2 피연산 데이터이고, C 행렬(32 x 16)은 연산 결과일 수 있다. A 행렬의 열은 제1 단위 데이터이고, B 행렬의 열은 제2 단위 데이터이고, A 행렬의 성분은 제1 요소 데이터이고, B 행렬의 성분은 제2 요소 데이터일 수 있다. 데이터 타입(data type)은 bfloat16으로 가정하며 각 행렬 요소의 용량은 2B로 가정한다.

[0045] A 행렬, 즉 제1 피연산 데이터가 예를 들어 복수의 메모리 뱅크(300) 중 어느 하나의 메모리 셀 어레이(310)에 저장되어 있는 경우, 열 우선(column-major) 방식으로 저장될 수 있다.

[0046] 상술한 행렬의 곱 연산을 수행하기 위한 도 4의 예시적인 소스 코드에서, 행렬 B의 요소, 즉 제2 단위 데이터는 행렬 A의 i에서 모두 공통적으로 연산된다. 또한, 행렬 A의 i는 행렬 B의 각 열, 즉 모든 메모리 뱅크(300)에서 공통적으로 연산된다.

[0047] 도 5는 본 출원의 일 실시예에 따른 각 메모리 뱅크의 연산 실행 동작을 도시한 것이다.

[0048] 도 5에서는, 행렬 B, 즉 각 뱅크의 제2 레지스터에 행렬 B의 각 열들이 저장되어 있는 것으로 가정하여 설명한다.

[0049] 각 메모리 뱅크(300)는 제2 피연산 데이터의 각 열, 즉 제2 단위 데이터를 메모리 셀 어레이(310)로부터 읽어와서 제2 레지스터(322)에 저장한다.

[0050] 이후, 복수의 시간 구간(T0 내지 T31) 내에서 행렬 A의 각 열, 즉 복수의 제1 단위 데이터는 복수의 시간 구간마다 메모리 뱅크에서 읽혀오거나 외부에서 입력되어 열 단위로 순차적으로 브로드캐스팅 된다. 행렬 A의 각 열은 모든 메모리 뱅크(300)에서 공유되어야 하므로, 모든 메모리 뱅크(300)에 브로드캐스팅 된다.

[0051] 브로드캐스팅 되는 제1 단위 데이터는 각 메모리 뱅크(300)의 제1 레지스터(321)에 저장되고, 그와 동시에 모든 메모리 뱅크(300)는 연산기(323)에서 연산을 수행한다. 이때, 연산기(323)는 복수의 시간 구간 마다 제1 단위

데이터에 포함된 복수의 제1 요소 데이터를 복수의 제2 단위 데이터 중 하나와 순차적으로 연산한다. 즉, 연산기(323)는 순차적으로 복수의 제1 단위 데이터를 읽어올 때 마다 복수의 제2 단위 데이터 중 제2 요소 데이터를 제1 단위 데이터 중 하나에 포함된 복수의 제1 요소 데이터와 연산한다.

[0052] 연산기(323)의 연산 수행 시, 도 5와 같이 연산기(323)의 일 측 입력에는 제1 레지스터(321)에 저장된 행렬 A의 각 열의 각 요소가 입력되고, 타 측 입력에는 제2 레지스터(322)에 저장된 행렬 B의 각 성분이 입력된다.

[0053] 예를 들어, 첫 번째 시간 구간(T0)에서는 각 연산기(323)의 일 측 피연산자로 제1 레지스터(321)에 저장된 행렬 A의 첫번째 열이 입력되고, 타 측 피연산자로 제2 레지스터(322)에 저장된 행렬 B의 첫 번째 열의 첫 번째 성분이 각 연산기(323)에 하나씩 입력된다. 두 번째 시간 구간(T1)에서는 일 측 피연산자로 제1 레지스터(321)에 저장된 행렬 A의 두 번째 열이 입력되고, 타 측 피연산자로 제2 레지스터(322)에 저장된 행렬 B의 첫 번째 열의 두 번째 성분이 각 연산기(323)에 하나씩 입력된다. 위 연산은 마지막 시간 구간(T31)까지 반복 수행된다.

[0054] 이후, 연산기(323)가 모든 연산을 수행하면, 각 메모리 뱅크(300)는 제3 레지스터(324)에 저장되어 있는 연산 결과, 즉 행렬 곱 결과를 행렬 C의 주소(메모리 셀 어레이(310))에 저장한다.

[0055] 도 6은 연산 유닛에서 어드레스 신호의 일부 비트를 참조하여 연산기에 제2 입력을 결정하는 모습을 예시적으로 나타낸 도면이다.

[0056] 도 6을 참조하면 일 실시예로, 본 발명은 일 실시예에 따라 PIM IU(200)에서 행렬 A의 각 열에 대한 메모리 요청이 올 때, PIM IU(200)에서 어드레스 신호 중 행렬 A에 관한 제1 피연산 데이터의 인덱스 정보를 지시하는 일부 비트(예를 들어, 10번째 비트부터 6번째 비트, ADDR[10:6])를 각 뱅크의 PIM 연산 유닛(320)으로 전달한다.

[0057] 예를 들어, 도 6에 도시된 바와 같이 0번 뱅크의 PIM 연산 유닛(320)에서 도 7의 어드레스 신호의 10번째 비트부터 6번째 비트(ADDR[10:6])를 참조하여 연산기(323)의 제2 입력을 결정한다. 각 뱅크의 PIM 연산 유닛(320)은 해당 정보를 참조하여 상기 제1 피연산 데이터와 연산 시킬 제2 피연산 데이터의 인덱스를 결정한다. 즉, PIM 연산 유닛(320)은 제2 레지스터(322)의 데이터 중 연산기(323)에 제2 입력으로 입력될 제2 요소 데이터를 결정한다.

[0058] 예를 들어, 메모리 뱅크(300)는 행렬 A의 15번째 열 $a_{0:31,15}$을 읽어 모든 뱅크의 PIM 연산 유닛(320)의 제1 레지스터(321)에 저장하고, 연산기(323)에 제1 입력으로 제1 레지스터(321)의 $a_{0:31,15}$전체를 입력한다. 상기 열에 대한 어드레스 신호 중 10번째 비트부터 6번째 비트는 01111이므로 PIM IU(200)가 해당 정보를 각 뱅크의 PIM 연산 유닛(320)으로 전달하여, 0번 뱅크와 1번 뱅크의 PIM 연산 유닛(320)의 연산기(323)에 제2 입력으로 각각 제2 레지스터의 $b_{15,0}$과 $b_{15,1}$을 입력한다. 이로써 본 출원의 실시예에 따른 메모리 장치(20)는 메모리 컨트롤러(10)의 스케줄링에 따라 뱅크 간의 메모리 요청 순서가 바뀌어도 연산 결과의 정확도를 보장한다.

[0059] 도 7은 어드레스 신호의 각 비트가 메모리 장치에 맵핑되는 예시적인 모습이다.

[0060] 도 7을 참조하면, 어드레스 신호는 복수의 메모리 셀 어레이(Row, Column) 각각에 대한 주소 정보, 복수의 메모리 뱅크(Bank) 각각에 대한 주소 정보 및 오프셋(Offset, 최소 접근 데이터 단위) 정보를 포함한다. 일 실시예로 32비트 어드레스 공간을 가정하였으며 메모리 장치(20)의 오프셋 정보를 64바이트로 가정한다. 어드레스 신호 중 5번째 비트부터 0번째 비트는 오프셋 정보인 64개 바이트를 표시한다. 본 출원의 일 실시예에 따라 PIM IU가 전달하는 일부 비트는 연산에 따라서 상기 오프셋 정보에 대응되는 비트로부터 k번째(여기서 k는 자연수임) 비트까지의 비트 구간을 포함할 수 있다.

[0061] 일 실시예에 따라 행렬 A는 각 열이 각 뱅크에 순차적으로 열 우선으로 저장되어 있다고 가정할 경우, 도 4의 행렬 A의 열 번호는 어드레스 신호의 10번째 비트부터 6번째 비트와 같다.

[0062] 도 8은 시간 구간에서의 명령어가 바뀌었을 때 본 출원의 일 실시예에 따른 메모리 장치의 동작을 나타내는 도면이다.

[0063] 도 8을 참조하면 임의의 시간 구간 중에 예를 들어, 첫 번째 시간 구간(T0)에서 어드레스 신호 중 10번째 비트부터 6번째 비트는 00001 이고, 두 번째 시간 구간(T1)에서는 00000 이므로, 제1 입력에서 명령어의 순서가 바뀌었을 때, 다시 말해 시간 구간의 순서와 시간 구간 각각에 해당하는 제1 단위 데이터의 순서가 불일치 할 때, PIM IU에서 어드레스 신호 중 제1 피연산 데이터의 인덱스 정보를 지시하는 10번째 비트부터 6번째 비트를 전달한다. 이에 따라 도 6에서 상술한대로 제2 입력이 결정되므로, 본 출원의 일 실시예에 따른 메모리 장치(20)는

시간 구간 (T0)와 (T1)에서 제1 입력의 순서가 바뀜에도 정확한 연산 결과를 산출할 수 있다.

[0064] 도 9는 본 출원의 실시예에 따른 메모리 장치의 동작 방법의 순서도이다.

[0065] S110 단계에서, 메모리 컨트롤러에서 보내는 데이터, 커맨드 및 어드레스 신호를 포함하는 입력신호를 메모리 장치에서 입력 받는다.

[0066] S120 단계에서, 상기 어드레스 신호 중 제1 피연산 데이터의 인덱스 정보를 지시하는 일부 비트를 복수의 메모리 뱅크 각각으로 전달한다. 구체적으로, PIM IU에서 어드레스 신호의 비트 중 연산기에 입력되는 제1 입력에 해당하는 데이터에 대응하는 일부 비트를 PIM 연산 유닛으로 보낸다.

[0067] S130 단계에서, 상기 제1 피연산 데이터에 포함된 복수의 제1 단위 데이터를 순차적으로 읽어와서 제2 피연산 데이터와 연산하되, 제2 피연산 데이터의 인덱스를 상기 일부 비트를 참조하여 결정한다. 구체적으로, PIM 연산 유닛은 상기 일부 비트를 참조하여 제2 레지스터의 데이터 중 연산기에 제2 입력으로 입력될 제2 요소 데이터를 결정한다.

[0068] S140 단계에서, 복수의 제1 단위 데이터를 포함하는 제1 피연산 데이터 및 복수의 제2 단위 데이터를 포함하는 제2 피연산 데이터에 기초하여 상기 인메모리 연산을 수행한다. 이로써 메모리 컨트롤러의 스케줄링에 따라 시간 구간에서 입력 데이터가 바뀌어도 입력 데이터에 대한 연산 결과가 바뀌지 않는다.

[0069] 이상에서는 본 발명에 따른 바람직한 실시 예들에 대하여 도시하고 또한 설명하였다. 그러나 본 발명은 상술한 실시 예에 한정되지 아니하며, 특허 청구의 범위에서 첨부하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시가 가능할 것이다.

부호의 설명

[0070] 10: 메모리 컨트롤러

20: 메모리 장치

100: 입출력단

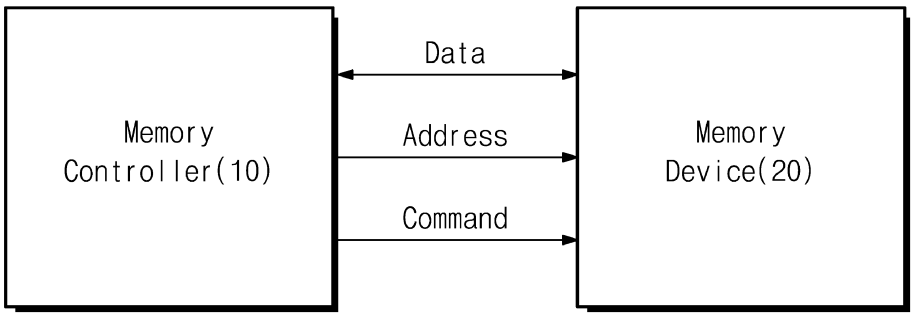
200: PIM IU

300: 메모리 뱅크

400: 공유 버스

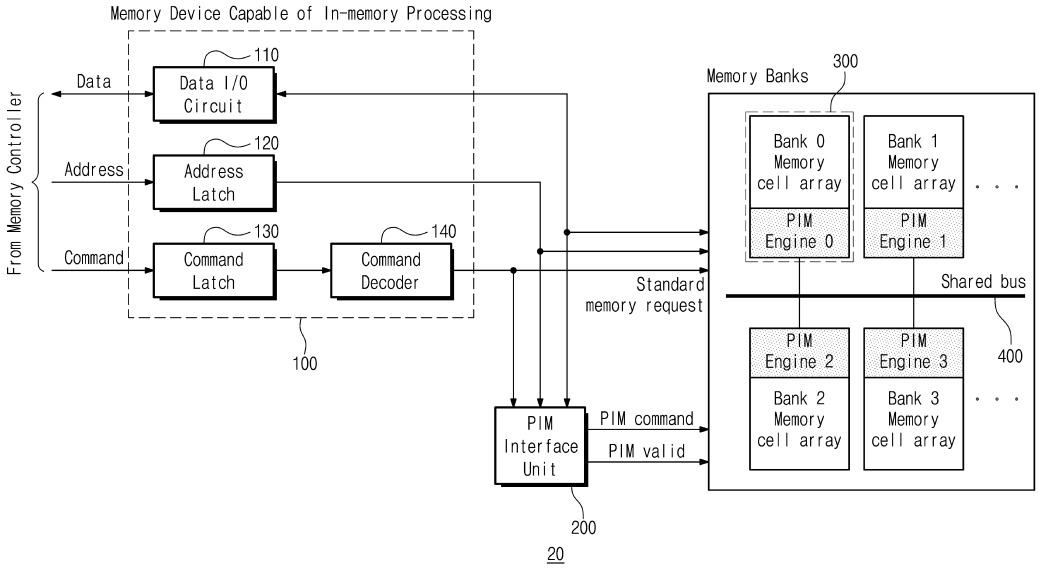
도면

도면1

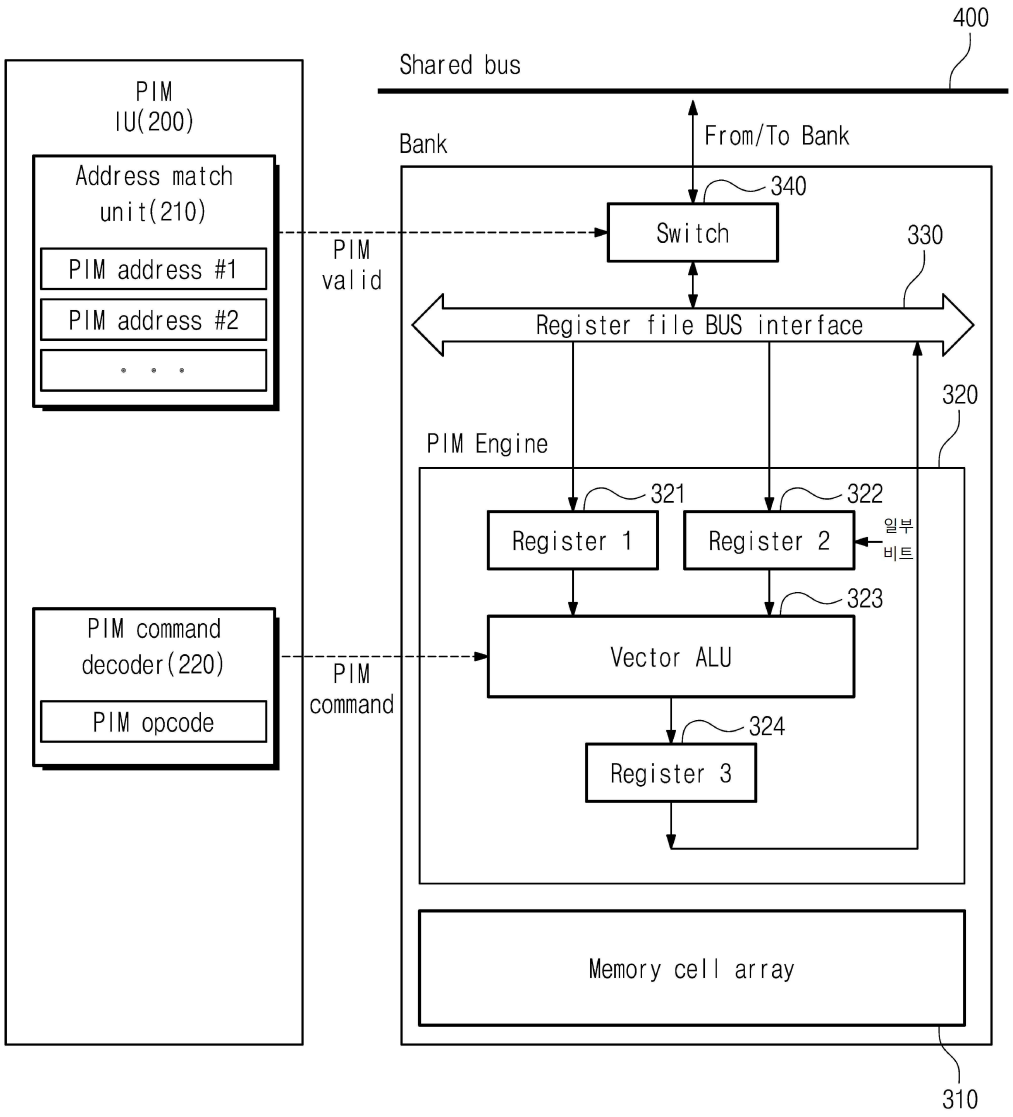


(1)

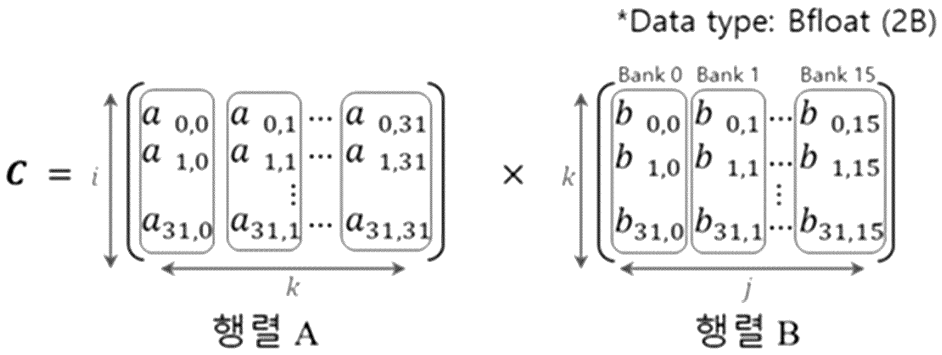
도면2


Memory Device Capable of In-memory Processing
From Memory Controller
Data
Address
Command
110
Data I/O Circuit
120
Address Latch
130
Command Latch
140
Command Decoder
100
Standard memory request
PIM Interface Unit
PIM command
PIM valid
200
20
Memory Banks
300
Bank 0 Memory cell array
PIM Engine 0
Bank 1 Memory cell array
PIM Engine 1
Shared bus
400
PIM Engine 2
Bank 2 Memory cell array
PIM Engine 3
Bank 3 Memory cell array

도면3

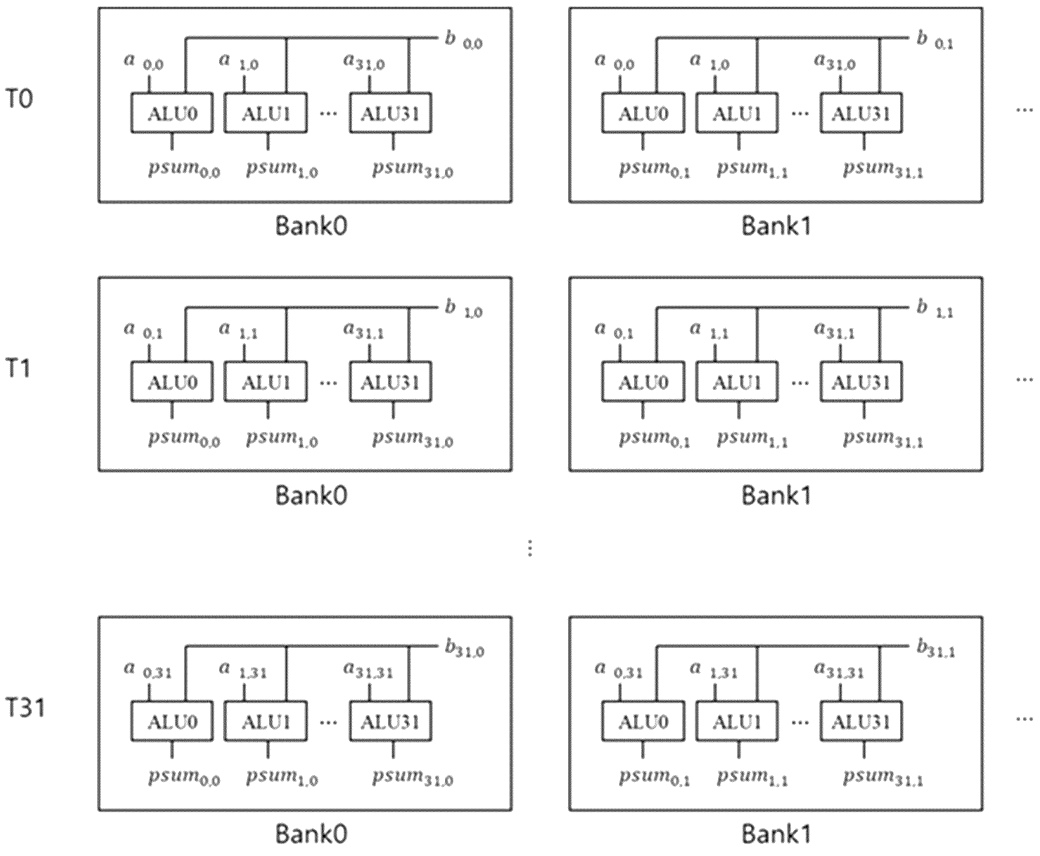

400
Shared bus
PIM
IU(200)
Address match
unit(210)
PIM address #1
PIM address #2
. . .
PIM
valid
Bank
From/To Bank
340
Switch
330
Register file BUS interface
320
PIM Engine
321
322
Register 1
Register 2
일부
비트
323
Vector ALU
324
Register 3
PIM command
decoder(220)
PIM opcode
PIM
command
Memory cell array
310

도면4

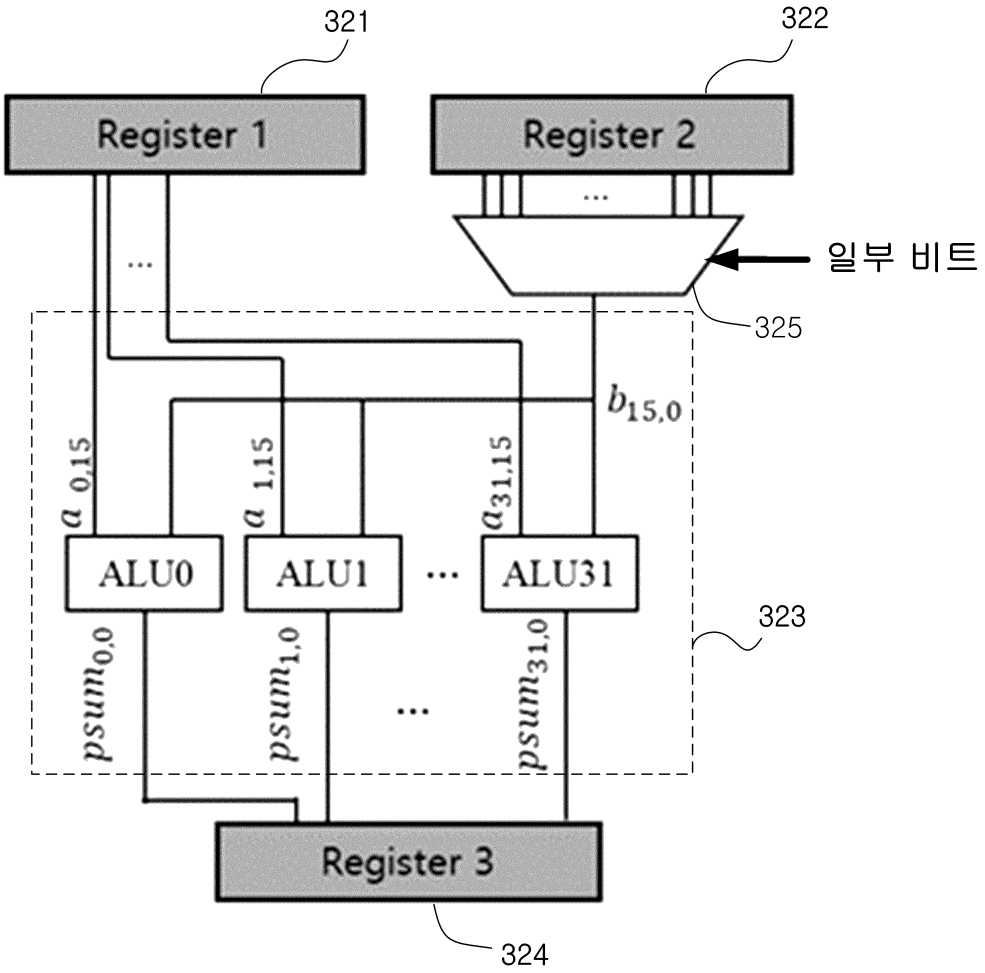



```
for(int i=0; i<32; i++)
    for(int j=0; j<16; j++)
        for(int k=0; k<32; k++)
            C[i][j]+=A[i][k]+B[k][j];
```

도면5



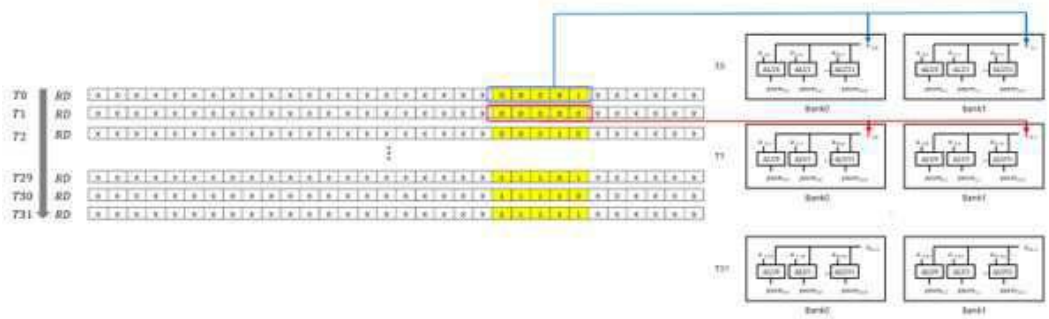
도면6


321
322
Register 1
Register 2
...
일부 비트
325
$b_{15,0}$
$a_{0,15}$
$a_{1,15}$
$a_{31,15}$
ALU0
ALU1
...
ALU31
323
$psum_{0,0}$
$psum_{1,0}$
...
$psum_{31,0}$
Register 3
324

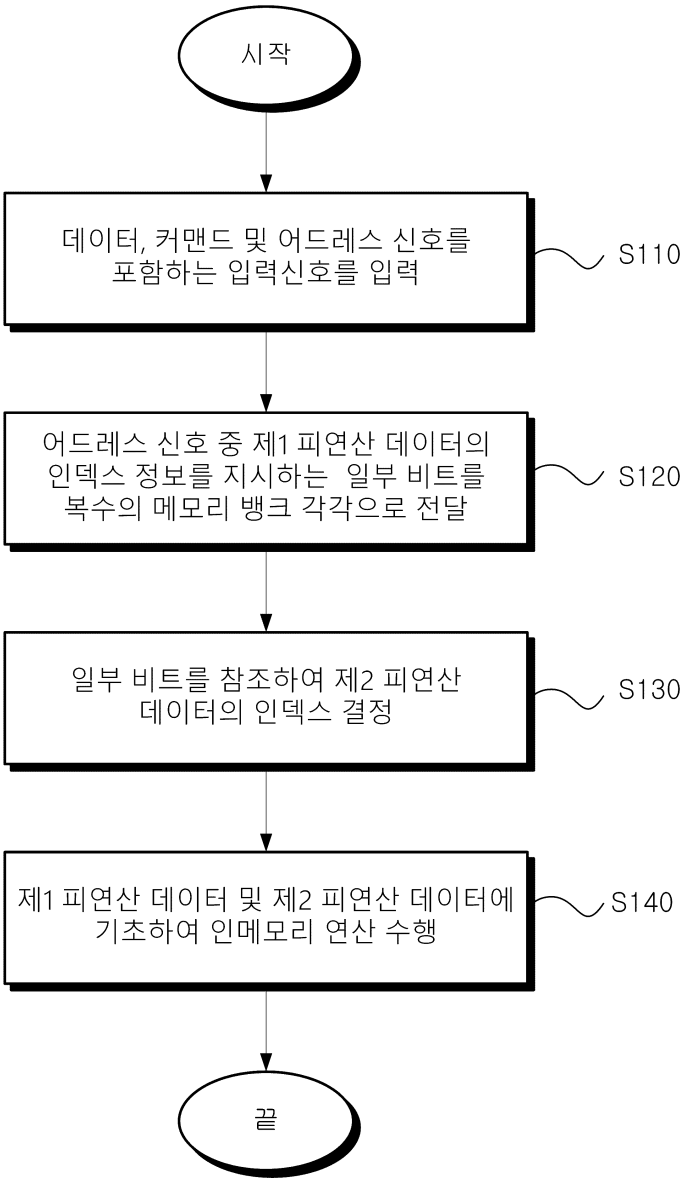
도면7


ADDR[10:6]
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
DRAM Row
DRAM Column
Bank
Offset

도면8



도면9


시작
데이터, 커맨드 및 어드레스 신호를 포함하는 입력신호를 입력
S110
어드레스 신호 중 제1 피연산 데이터의 인덱스 정보를 지시하는 일부 비트를 복수의 메모리 뱅크 각각으로 전달
S120
일부 비트를 참조하여 제2 피연산 데이터의 인덱스 결정
S130
제1 피연산 데이터 및 제2 피연산 데이터에 기초하여 인메모리 연산 수행
S140
끝