(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년12월24일
(11) 등록번호 10-2342994
(24) 등록일자 2021년12월21일

(51) 국제특허분류(Int. Cl.)
G06F 9/28 (2017.01) *G06F 15/78* (2006.01)
G06F 5/01 (2006.01) *G06F 5/06* (2006.01)
G06F 7/50 (2006.01) *G06F 7/52* (2006.01)
(52) CPC특허분류
G06F 9/28 (2013.01)
G06F 15/7839 (2013.01)
(21) 출원번호 10-2020-0137802
(22) 출원일자 2020년10월22일
심사청구일자 2020년10월22일
(30) 우선권주장
1020200090069 2020년07월21일 대한민국(KR)
(56) 선행기술조사문헌
KR1020200081623 A
KR1020170118051 A
KR1020160130700 A
KR1020170015477 A

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
박종선
서울특별시 서초구 신반포로 9
이경호
서울특별시 성북구 고려대로11길 20
최웅
경기도 부천시 소사로 170번길 81
(74) 대리인
이기성

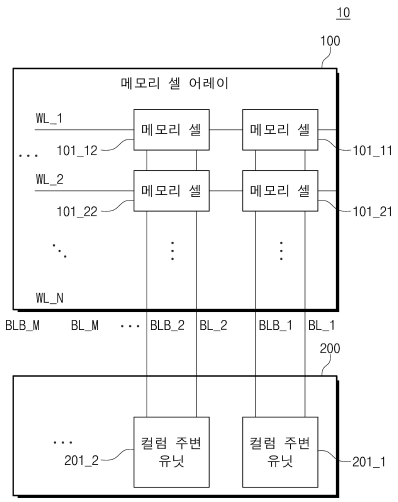
전체 청구항 수 : 총 20 항 심사관 : 지정훈

(54) 발명의 명칭 **산술 연산을 지원하는 인메모리 컴퓨팅**

(57) 요 약

본 출원의 일 실시예에 따른 인메모리 컴퓨팅은 메모리 셀 어레이 및 상기 메모리 셀 어레이에 연결된 복수의 비트라인쌍들에 연결되는 복수의 컬럼 주변 유닛들을 포함하는 컬럼 주변 회로를 포함하고, 상기 컬럼 주변 유닛들 각각은 상기 복수의 비트라인쌍들 중 하나의 비트라인 쌍을 통해 비트와이즈 데이터를 감지 증폭하는 감지증폭 및 라이트부 및 상기 비트와이즈 데이터에 기초하여, 전가산기 불 함수 기반의 산술 연산을 수행하고, 상기 산술 연산에 따라 획득되는 연산 데이터에 대해 상기 감지증폭 및 라이트부를 통해 라이트 백 동작을 수행시키는 연산 로직부를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 5/01 (2013.01)
G06F 5/06 (2021.08)
G06F 7/50 (2021.08)
G06F 7/52 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711115231
과제번호 2020R1A2C3014820
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 개인기초연구(과기정통부)(R&D)
연구과제명 온-디바이스 개인화를 위한 에너지 효율적인 딥러닝 가속기 설계
기 여 율 3/10
과제수행기관명 고려대학교
연구기간 2020.03.01 ~ 2021.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1711104070
과제번호 2015M3D1A1070465
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 미래소재디스커버리지원(R&D)
연구과제명 설계기반 Spin-Orbitronics 소재 개발
기 여 율 4/10
과제수행기관명 고려대학교
연구기간 2020.01.04 ~ 2021.01.03
이 발명을 지원한 국가연구개발사업
과제고유번호 1711121367
과제번호 2020M3F3A2A01082591
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 차세대지능형반도체기술개발(R&D)
연구과제명 저전력 스핀토크 소자 기반 보안칩 개발
기 여 율 3/10
과제수행기관명 고려대학교
연구기간 2020.07.01 ~ 2020.12.31

명 세 서

청구범위

청구항 1

메모리 셀 어레이; 및

상기 메모리 셀 어레이에 연결된 복수의 비트라인쌍들에 연결되는 복수의 컬럼 주변 유닛들을 포함하는 컬럼 주변 회로를 포함하고,

상기 컬럼 주변 유닛들 각각은

상기 복수의 비트라인쌍들 중 하나의 비트라인 쌍을 통해 비트와이즈 데이터를 감지 증폭하는 감지증폭 및 라이트부; 및

상기 비트와이즈 데이터에 기초하여, 전가산기 불 함수 기반의 산술 연산을 수행하고, 상기 산술 연산에 따라 획득되는 연산 데이터에 대해 상기 감지증폭 및 라이트부를 통해 라이트 백 동작을 수행시키는 연산 로직부를 포함하는, 인메모리 컴퓨팅.

청구항 2

제1항에 있어서,

상기 연산 데이터는 NAND, AND, NOR, OR, XNOR, XOR, NOT, Shift 연산 값을 포함하는 로직 데이터와 ADD, ADD-Shift, SUB, MULT 연산 값을 정수 데이터 중 어느 하나를 포함하는, 인메모리 컴퓨팅.

청구항 3

제2항에 있어서,

상기 연산 로직부는 하위 비트 측으로부터 제1 캐리값을 출력받는 제1 멀티플렉서;

상기 하위 비트 측으로부터 제1 합산값을 출력받는 시프트 플립플롭;

상기 제1 캐리값, 상기 제1 합산값 및 상기 감지증폭 및 라이트부를 감지된 비트와이즈 데이터에 기초하여, 제2 캐리값과 제2 합산값을 연산하는 전가산기 로직;

상기 제2 합산값을 출력받아 상위 비트측으로 전달하는 제2 멀티플렉서; 및

상기 제1 캐리값, 상기 제1 합산값, 상기 제2 합산값 및 상기 비트와이즈 데이터 중 적어도 하나를 출력받는 제3 멀티플렉서를 포함하는, 인메모리 컴퓨팅.

청구항 4

제3항에 있어서,

상기 제3 멀티플렉서는 상기 제1 캐리값, 상기 제1 합산값, 상기 제2 합산값 및 상기 비트와이즈 데이터 중 적어도 하나를 상기 감지증폭 및 라이트부를 통해 라이트 백 시키는, 인메모리 컴퓨팅.

청구항 5

제3항에 있어서,

상기 전가산기 로직은 상기 제1 멀티플렉서로부터 전송받는 선택 신호에 따라 스위칭되는 전송 게이트 기반의 회로로 구현된, 인메모리 컴퓨팅.

청구항 6

제3항에 있어서,

상기 컬럼 주변 회로는 상기 비트와이즈 데이터를 상위 비트 측 방향으로 전달하는 Shift 연산을 수행하는, 인

메모리 컴퓨팅.

청구항 7

제6항에 있어서,

상기 컬럼 주변 회로는 상기 Shift 연산이 수행되는 단일 주기에서, 상기 복수의 연산 로직부들을 통해 수행되는 모든 Shift 연산 동작을 한번에 수행하는, 인메모리 컴퓨팅.

청구항 8

제3항에 있어서,

상기 컬럼 주변 회로는 상기 제2 캐리값을 상위 비트 측 방향으로 전달하고, 상기 제2 합산값을 라이트 백 시키는 ADD 연산을 수행하는, 인메모리 컴퓨팅.

청구항 9

제8항에 있어서,

상기 컬럼 주변 회로는 상기 ADD 연산이 수행되는 단일 주기에서, 상기 복수의 연산 로직부들을 통해 수행되는 모든 ADD 연산 동작을 한번에 수행하는, 인메모리 컴퓨팅.

청구항 10

제3항에 있어서,

상기 컬럼 주변 회로는 상기 제2 합산값을 상위 비트 측 방향으로 전달함에 따라, 하위 비트 측으로부터 전달받는 상기 제1 합산값을 라이트 백 시키는 ADD-Shift 연산 동작을 수행하는, 인메모리 컴퓨팅.

청구항 11

제10항에 있어서,

상기 ADD-Shift 연산이 수행되는 주기는 상기 제2 합산값을 상위 비트 측 방향으로 전달하는 제1 주기와 상기 제1 합산값을 라이트 백 시키는 제2 주기를 포함하는, 인메모리 컴퓨팅.

청구항 12

메모리 셀 어레이;

상기 메모리 셀 어레이에 연결된 복수의 컬럼 주변 유닛들을 포함하는 컬럼 주변 회로부;

상기 메모리 셀 어레이에 저장된 피승수 데이터를 저장하는 더미셀 어레이부;

상기 더미셀 어레이와 상기 메모리 셀 어레이를 분리시키는 BL 세퍼레이터부; 및

상기 메모리 셀 어레이로부터 로드되는 승수 데이터에 기초하여, 상기 복수의 컬럼 주변 유닛들의 곱셈 연산 동작을 제어하는 시프트 레지스터 회로를 포함하는, 인메모리 컴퓨팅.

청구항 13

제12항에 있어서,

상기 더미셀 어레이부는 0의 데이터를 저장하는 제1 서브 어레이;

상기 피승수 데이터를 최하위 비트에서 최상위 비트 순서로 저장하는 제2 서브 어레이; 및

상기 복수의 컬럼 주변 유닛들을 통해 라이트 백 되는 임시 데이터를 저장하는 제3 서브 어레이를 포함하는, 인메모리 컴퓨팅.

청구항 14

제12항에 있어서,

상기 시프트 레지스터 회로는 상기 승수 데이터가 최상위 비트에서 최하위 비트 순서로 출력되는 로드 데이터를 저장하는 복수의 승수 플립플롭들; 및

상기 로드 데이터에 기초하여, 상기 복수의 컬럼 주변 유닛들을 제어하기 위한 제어 신호를 전송하는 복수의 제어 멀티플렉서들을 포함하는, 인메모리 컴퓨팅.

청구항 15

제13항에 있어서,

상기 복수의 컬럼 주변 유닛들은 상기 승수 데이터의 최상위 비트에 대응되는 제어 신호에 기초하여, 상기 제1 및 제2 서브 어레이 대한 컬럼별 ADD-Shift 연산 동작을 수행하는, 인메모리 컴퓨팅.

청구항 16

제13항에 있어서,

상기 복수의 컬럼 주변 유닛들은 상기 승수 데이터의 중간 비트가 0에 대응되는 제어 신호에 기초하여, 상기 제3 서브 어레이에 대한 Shift 연산 동작을 수행하는, 인메모리 컴퓨팅.

청구항 17

제13항에 있어서,

상기 복수의 컬럼 주변 유닛들은 상기 승수 데이터의 중간 비트가 1에 대응되는 제어 신호에 기초하여, 상기 제2 및 제3 서브 어레이에 대한 컬럼별 ADD-Shift 연산 동작을 수행하는, 인메모리 컴퓨팅.

청구항 18

제13항에 있어서,

상기 복수의 컬럼 주변 유닛들은 상기 승수 데이터의 최하위 비트에 대응되는 제어 신호에 기초하여, 상기 제2 및 제3 서브 어레이에 대한 ADD 연산 동작을 수행하는, 인메모리 컴퓨팅.

청구항 19

제13항에 있어서,

상기 복수의 컬럼 주변 유닛들 각각은 하위 비트 측으로부터 제1 캐리값을 출력받는 제1 멀티플렉서;

상기 하위 비트 측으로부터 제1 합산값을 출력받는 시프트 플립플롭;

상기 제1 캐리값, 상기 제1 합산값 및 감지증폭 및 라이트부를 통해 감지된 비트와이즈 데이터에 기초하여, 제2 캐리값과 제2 합산값을 연산하는 전가산기 로직;

상기 제2 합산값을 출력받아 상위 비트측으로 전달하는 제2 멀티플렉서; 및

상기 제1 캐리값, 상기 제1 합산값, 상기 제2 합산값 및 상기 비트와이즈 데이터 중 적어도 하나를 출력받는 제3 멀티플렉서를 포함하는, 인메모리 컴퓨팅.

청구항 20

제19항에 있어서,

상기 전가산기 로직은 상기 제1 멀티플렉서로부터 전송받는 선택 신호에 따라 스위칭되는 전송 게이트 기반의 회로로 구현된, 인메모리 컴퓨팅.

발명의 설명

기 술 분 야

[0001] 본 출원은 산술 연산을 지원하는 인메모리 컴퓨팅에 관한 것으로, 특히, 캐리 전달이 필요한 정수 연산을 고속

으로 지원하는 인메모리 컴퓨팅에 관한 것이다.

배 경 기 술

[0002] 기존의 캐시 메모리는 연산 동작에 비해 상대적으로 느린 읽기/쓰기 동작만으로 제한되어있다. 그러나, 최근 CPU에서 요구하는 데이터가 늘어날수록 메모리 접근이 늘어나며, 그로 인해 메모리의 부담이 커지고 있다. 이때 메모리에서 발생하는 선천적인 문제인 메모리 병목 현상을 줄이기 위해, 읽기/쓰기 이외에 추가적으로 메모리 내부에서 산술 연산을 수행하는 컴퓨팅 인 메모리 구조가 많이 제시되고 있다.

발명의 내용

해결하려는 과제

[0003] 본 출원의 목적은 캐리 전달이 필요한 정수 연산을 고속으로 지원하는 인메모리 컴퓨팅 구조를 제공하기 위한 것이다.

과제의 해결 수단

[0004] 본 출원의 일 실시예에 따른 인메모리 컴퓨팅은 메모리 셀 어레이 및 상기 메모리 셀 어레이에 연결된 복수의 비트라인쌍들에 연결되는 복수의 컬럼 주변 유닛들을 포함하는 컬럼 주변 회로를 포함하고, 상기 컬럼 주변 유닛들 각각은 상기 복수의 비트라인쌍들 중 하나의 비트라인 쌍을 통해 비트와이즈 데이터를 감지 증폭하는 감지 증폭 및 라이트부 및 상기 비트와이즈 데이터에 기초하여, 전가산기 불 함수 기반의 산술 연산을 수행하고, 상기 산술 연산에 따라 획득되는 연산 데이터에 대해 상기 감지증폭 및 라이트부를 통해 라이트 백 동작을 수행시키는 연산 로직부를 포함한다.

[0005] 본 출원의 다른 실시예에 따른 인메모리 컴퓨팅은 메모리 셀 어레이, 상기 메모리 셀 어레이에 연결된 복수의 컬럼 주변 유닛들을 포함하는 컬럼 주변 회로부, 상기 메모리 셀 어레이에 저장된 피승수 데이터를 저장하는 더미셀 어레이부, 상기 더미셀 어레이와 상기 메모리 셀 어레이를 분리시키는 BL 세퍼레이터부 및 상기 메모리 셀 어레이로부터 로드되는 승수 데이터에 기초하여, 상기 복수의 컬럼 주변 유닛들의 곱셈 연산 동작을 제어하는 시프트 레지스터 회로를 포함한다.

발명의 효과

[0006] 본 출원의 실시예에 따른 인메모리 컴퓨팅은 메모리 내부에서 고속의 산술 연산을 가능하게 할 수 있는 효과가 있다.

[0007] 또한, 정수 연산을 수행할 때, 캐리 전달 지연을 줄이고, 곱셈의 레이턴시를 감소시켜 연산 에너지 효율을 개선시킬 수 있는 효과가 있다.

[0008] 또한, 메모리 내부의 커패시턴스를 줄여 연산 속도를 증가시키고 에너지 소모를 감소시킬 수 있는 효과가 있다.

도면의 간단한 설명

[0009] 도 1은 본 출원의 실시예에 따른 메모리인 컴퓨팅에 대한 블록도이다.

도 2는 도 1의 컬럼 주변 유닛에 대한 블록도이다.

도 3은 도 2의 연산 로직부를 구체적으로 보여주는 회로도이다.

도 4는 도 3의 전가산기 로직에 대한 실시 예이다.

도 5는 도 1의 컬럼 주변 회로의 Shift 연산 동작을 설명하기 위한 예시 도이다.

도 6a와 도 6b는 도 1의 컬럼 주변 회로의 ADD 연산 동작을 설명하기 위한 예시 도이다.

도 7a와 도 7b는 도 1의 컬럼 주변 회로의 ADD-Shift 연산 동작을 설명하기 위한 예시 도이다.

도 8은 본 출원의 다른 실시예에 따른 메모리인 컴퓨팅에 대한 개념도이다.

도 9는 도 8의 메모리인 컴퓨팅에 대한 실시 예이다.

도 10은 레프트-쉬프트 곱셈연산 동작을 설명하기 위한 실시 예이다.

도 11과 도 12는 도 10의 제1 임시 데이터를 획득하기 위한 메모리인 컴퓨팅의 동작 예시도이다.

도 13과 도 14는 도 10의 제2 임시 데이터를 획득하기 위한 메모리인 컴퓨팅의 동작 예시도이다.

도 15와 도 16은 도 10의 제3 임시 데이터를 획득하기 위한 메모리인 컴퓨팅의 동작 예시도이다.

도 17은 도 10의 곱셈 데이터(MUL_D)를 획득하기 위한 메모리인 컴퓨팅의 동작 예시도이다.

발명을 실시하기 위한 구체적인 내용

[0010] 이하, 구체적인 실시형태 및 첨부된 도면을 참조하여 본 출원의 실시형태를 설명한다. 그러나, 본 출원의 실시형태는 여러 가지 다른 형태로 변형될 수 있으며, 본 출원의 범위가 이하 설명하는 실시형태로 한정되는 것은 아니다. 또한, 본 출원의 실시형태는 통상의 기술자에게 본 발명을 더욱 완전하게 설명하기 위해서 제공되는 것이다. 따라서, 도면에서의 요소들의 형상 및 크기 등은 보다 명확한 설명을 위해 과장될 수 있으며, 도면상의 동일한 부호로 표시되는 요소는 동일한 요소이다.

[0011] 그리고 도면에서 본 출원을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하고, 여러 층 및 영역을 명확하게 표현하기 위하여 두께를 확대하여 나타내었으며, 동일한 사상의 범위 내의 기능이 동일한 구성요소는 동일한 참조부호를 사용하여 설명한다. 나아가, 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0012] 도 1은 본 출원의 실시예에 따른 메모리인 컴퓨팅(10)에 대한 블록도이고, 도 2는 도 1의 컬럼 주변 유닛(예컨대, 200_2)에 대한 블록도이다.

[0013] 도 1과 도 2를 참조하면, 메모리인 컴퓨팅(10)은 메모리 셀 어레이(100)와 컬럼 주변 회로(200)를 포함할 수 있다.

[0014] 먼저, 메모리 셀 어레이(100)는 복수의 워드라인들(WL_1~WL_N)과 복수의 비트라인쌍(BLB_1~BLB_M, BL_1~BL_M)에 연결된 복수의 메모리 셀들(101_1M~101_NM)을 포함할 수 있다.

[0015] 다음으로, 컬럼 주변 회로(200)는 각 비트라인쌍(예컨대, BLB, BL)을 통해 메모리 셀 어레이(100)에 연결되는 복수의 컬럼 주변 유닛들(201_1~201_N)을 포함할 수 있다.

[0016] 여기서, 복수의 컬럼 주변 유닛들(201_1~201_N) 각각은 감지증폭 및 라이트부(예컨대, 210_2) 및 연산 로직부(예컨대, 220_2)를 포함할 수 있다.

[0017] 구체적으로, 감지증폭 및 라이트부(예컨대, 210_2)는 메모리 셀 어레이(100)에 연결된 비트라인쌍(예컨대, BLB, BL)을 통해 비트와이즈 데이터를 감지 증폭할 수 있다. 여기서, 비트와이즈 데이터는 하나 또는 복수의 워드라인 신호가 활성화됨에 따라 메모리 셀 어레이(100)로부터 비트라인쌍(예컨대, BLB, BL)을 통해 출력되는 데이터일 수 있다. 예를 들면, 비트와이즈 데이터는 AND 연산 값과 NOR 연산 값을 포함할 수 있다.

[0018] 다음으로, 연산 로직부(예컨대, 220_2)는 비트와이즈 데이터에 기초하여, 전가산기 불 함수(Full-Adder Boolean Equation) 기반의 산술 연산을 수행할 수 있다. 여기서, 전가산기 불 함수는 캐리 값과 합산 값을 산술 연산하기 위한 방정식일 수 있다.

[0019] 이때, 연산 로직부(예컨대, 220_2)는 산술 연산에 따라 획득되는 연산 데이터에 대해 감지증폭 및 라이트부(예컨대, 210_2)를 통해 라이트 백 동작을 수행시킬 수 있다. 여기서, 연산 데이터는 NAND, AND, NOR, OR, XNOR, XOR, NOT, Shift 연산 값을 포함하는 로직 데이터와 ADD, ADD-Shift, SUB, MULT 연산 값을 포함하는 정수 데이터 중 어느 하나일 수 있다.

[0020] 본 출원의 실시 예에 따른 메모리인 컴퓨팅(10)은 연산 로직부(예컨대, 220_2)를 통해 전가산기 불 함수 기반의 산술 연산을 수행하고, 산술 연산에 따라 획득되는 연산 데이터에 대해 감지증폭 및 라이트부(예컨대, 210_2)를 통해 라이트 백 동작을 수행시킴으로써, 메모리 내부에서 고속 산술 연산을 가능하게 할 수 있는 효과가 있다.

[0021] 이하, 도 3과 도 4를 참조하여, 연산 로직부(예컨대, 220_2)에 대해 보다 구체적으로 설명될 것이다.

[0022] 도 3은 도 2의 연산 로직부(예컨대, 220_2)를 구체적으로 보여주는 회로도이고, 도 4는 도 3의 전가산기 로직(223)에 대한 실시 예이다.

[0023] 도 3을 참조하면, 연산 로직부(예컨대, 220_2)는 제1 멀티플렉서(221), 시프트 플립플롭(222), 전가산기 로직

(223) 제2 멀티플렉서(224) 및 제3 멀티플렉서(225)를 포함할 수 있다.

[0024] 먼저, 제1 멀티플렉서(221)는 하위 비트 측으로부터 제1 캐리값(C_{N-1})을 출력받을 수 있다. 구체적으로, 제1 멀티플렉서(221)는 하위 비트라인 쌍에 연결된 컬럼 주변 유닛(예컨대, 201_1)의 연산 로직부(예컨대, 220_1)로부터 제1 캐리값(C_{N-1})을 출력받을 수 있다.

[0025] 또한, 제1 멀티플렉서(221)는 선택 제어 신호(Logic SEL)에 응답하여, 전가산기 로직(223)에 선택 신호(LSEL)를 전송함으로써, 전가산기 로직(223)을 제어할 수 있다.

[0026] 다음으로, 시프트 플립플롭(222)은 하위 비트 측으로부터 제1 합산값(S_{N-1})을 출력받을 수 있다. 구체적으로, 시프트 플립플롭(222)은 하위 비트라인 쌍에 연결된 컬럼 주변 유닛(예컨대, 201_1)의 연산 로직부(예컨대, 220_1)로부터 제1 합산값(S_{N-1})을 출력받아 저장할 수 있다.

[0027] 다음으로, 전가산기 로직(223)은 감지증폭 및 라이트부(예컨대, 210_2)를 통해 감지된 비트와이즈 데이터, 제1 캐리값(C_{N-1}), 제1 합산값(S_{N-1})에 기초하여, 전가산기 불 함수를 통해 제2 캐리값(C_N)과 제2 합산값(S_N)을 연산할 수 있다.

[0028] 실시예에 따른 전가산기 로직(223)은 도 4에 도시된 바와 같이, 제1 멀티플렉서(221)로부터 전송받는 선택 신호(LSEL)에 따라 스위칭되는 전송 게이트 기반의 회로로 구현됨으로써, 로직 게이트 기반의 전가산기보다 캐리 전달 지연을 감소시킬 수 있다.

[0029] 이때, 전가산기 로직(223)은 제2 합산값(S_N)을 제2 멀티플렉서(224)에 출력하고, 제2 캐리값(C_N)을 상위 비트라인쌍에 연결된 컬럼 주변 유닛(예컨대, 201_3)의 연산 로직부(예컨대, 220_3)로 출력할 수 있다.

[0030] 다음으로, 제2 멀티플렉서(224)는 전가산기 로직(223)을 통해 연산된 제2 합산값(S_N)을 출력받아, 상위 비트 측으로 전달할 수 있다.

[0031] 구체적으로, 제2 멀티플렉서(224)는 전가산기 로직(223)을 통해 연산된 제2 합산값(S_N)을 상위 비트라인쌍에 연결된 컬럼 주변 유닛(200_3)의 연산 로직부(예컨대, 220_3)에 전달할 수 있다.

[0032] 또한, 제2 멀티플렉서(224)는 전가산기 로직(223)을 통해 연산된 제2 합산값(S_N)을 제3 멀티플렉서(225)에 출력할 수 있다.

[0033] 다음으로, 제3 멀티플렉서(225)는 제1 캐리값(C_{N-1}), 제1 합산값(S_{N-1}), 제2 합산값(S_N) 및 로직 데이터(Logics) 중 적어도 하나를 출력받을 수 있다. 이때, 제3 멀티플렉서(225)는 제1 캐리값(C_{N-1}), 제1 합산값(S_{N-1}), 제2 합산값(S_N) 및 로직 데이터(Logics) 중 적어도 하나를 감지증폭 및 라이트부(210)를 통해 메모리 셀 어레이(100)로 라이트 백 시킬 수 있다.

[0034] 이하, 도 5를 참조하여, 컬럼 주변 회로(200)의 Shift 연산 동작에 대해 구체적으로 설명될 것이다.

[0035] 도 5는 도 1의 컬럼 주변 회로(200)의 Shift 연산 동작을 설명하기 위한 예시 도이다.

[0036] 도 5를 참조하면, 컬럼 주변 회로(200)는 비트와이즈 데이터(예컨대, A0)를 상위 비트 측 방향으로 전달하는 Shift 연산 동작을 수행할 수 있다.

[0037] 구체적으로, 컬럼 주변 회로(200)는 제1 연산 로직부(예컨대, 220_1)를 통해 출력되는 비트와이즈 데이터(예컨대, A0)를 상위 비트 측 방향에 위치한 제2 연산 로직부(예컨대, 220_2)에 전달할 수 있다.

[0038] 이때, 제1 연산 로직부(예컨대, 220_1)는 제2 연산 로직부(예컨대, 220_2)의 하위 비트 측에 위치하고, 제2 연산 로직부(예컨대, 220_2)는 제1 연산 로직부(예컨대, 220_1)의 상위 비트 측에 위치하며, 제1 및 제2 연산 로직부(예컨대, 220_1과 220_2)는 서로 전기적으로 연결될 수 있다.

[0039] 즉, 컬럼 주변 회로(200)는 제1 연산 로직부(220_1)의 전가산기 로직(223_1)를 통해 출력되는 비트와이즈 데이터(예컨대, A0)를 제2 연산 로직부(220_2)의 제3 멀티플렉서(225_2)에 전달할 수 있다. 이때, 멀티플렉서(225_2)는 비트와이즈 데이터(예컨대, A0)를 감지증폭 및 라이트부(210_2)를 통해 라이트 백 시킬 수 있다.

[0040] 또한, 컬럼 주변 회로(200)는 Shift 연산이 수행되는 단일 주기에서 복수의 연산 로직부들(220_1~220_N)을 통해

수행되는 모든 Shift 연산 동작을 한번에 수행할 수 있다.

[0041] 이하, 도 6a와 도 6b를 참조하여, 컬럼 주변 회로(200)의 ADD 연산 동작에 대해 구체적으로 설명될 것이다.

[0042] 도 6a와 도 6b는 도 1의 컬럼 주변 회로(200)의 ADD 연산 동작을 설명하기 위한 예시 도이다.

[0043] 도 6a과 도 6b를 참조하면, 컬럼 주변 회로(200)는 전가산기 불 함수를 통해 연산되는 제2 캐리값(C_N)을 상위 비트 측 방향으로 전달시키고, 전가산기 불 함수를 통해 연산되는 제2 합산값(S_N)을 라이트 백 하는 ADD 연산을 수행할 수 있다.

[0044] 예를 들면, 도 6a에 도시된 바와 같이, 컬럼 주변 회로(200)는 제1 연산 로직부(예컨대, 220_1)를 통해 연산되는 캐리값(예컨대, C0)을 제2 연산 로직부(예컨대, 220_2)로 전달할 수 있다. 이때, 컬럼 주변 회로(200)는 제1 연산 로직부(예컨대, 220_1)를 통해 연산되는 제2 합산값(예컨대, S0)을 제3 멀티플렉서(예컨대, 225_1)를 통해 라이트 백 시킬 수 있다.

[0045] 또한, 도 6b에 도시된 바와 같이, 컬럼 주변 회로(200)는 제2 연산 로직부(예컨대, 220_2)를 통해 연산되는 캐리값(예컨대, C1)을 제3 연산 로직부(예컨대, 220_3)로 전달할 수 있다. 이때, 컬럼 주변 회로(200)는 제2 연산 로직부(예컨대, 220_2)를 통해 연산되는 제2 합산값(예컨대, S1)을 제3 멀티플렉서(예컨대, 225_2)를 통해 라이트 백 시킬 수 있다.

[0046] 즉, 각 연산 로직부(220_1~220_4)를 통해 출력되는 각 캐리값(예컨대, C0)을 상위 비트라인 방향으로 전달시키고, 컬럼 주변 회로(200)는 각 연산 로직부(220_1~220_4)를 통해 출력되는 각 합산값(예컨대, S0~S3)을 라이트 백 할 수 있다.

[0047] 또한, 컬럼 주변 회로(200)는 ADD 연산이 수행되는 단일 주기에서 복수의 연산 로직부들(220_1~220_N)을 통해 수행되는 모든 ADD 연산 동작을 한번에 수행할 수 있다.

[0048] 구체적으로, 컬럼 주변 회로(200)는 ADD 연산이 수행되는 단일 주기에서, 전가산기 풀 함수 기반의 각 캐리값(예컨대, C0~C3)을 상위 비트 측으로 전달시키고, 전가산기 풀 함수 기반의 각 합산값(예컨대, S0~S3)을 라이트 백 하는 ADD 연산 동작을 한번에 수행할 수 있다.

[0049] 이하, 도 7a와 도 7b를 참조하여, 컬럼 주변 회로(200)의 ADD-Shift 연산 동작에 대해 구체적으로 설명될 것이다.

[0050] 도 7a와 도 7b는 도 1의 컬럼 주변 회로(200)의 ADD-Shift 연산 동작을 설명하기 위한 예시 도이다.

[0051] 도 7a와 도 7b를 참조하면, 컬럼 주변 회로(200)는 전가산기 불 함수를 통해 연산되는 제2 합산값(S_N)을 상위 비트 측으로 전달함에 따라, 하위 비트 측으로부터 전달받는 제1 합산값(S_{N-1})을 라이트 백 시키는 ADD-Shift 연산 동작을 수행할 수 있다.

[0052] 예를 들면, 제1 내지 제4 연산 로직부(예컨대, 220_1~220_4)는 도 7a에 도시된 바와 같이, 각 제2 합산값(예컨대, S0~S3)을 상위 비트 측 방향으로 전달할 수 있다.

[0053] 그런 다음, 도 7b에 도시된 바와 같이, 제1 내지 제5 연산 로직부(예컨대, 220_1~220_5)는 각 시프트 플립플롭(예컨대, 222_1~222_5)으로부터 각 제3 멀티플렉서(예컨대, 225_1~225_5)를 통해 각 제1 합산값(예컨대, 0~S3)을 라이트 백 시킬 수 있다.

[0054] 즉, 컬럼 주변 회로(200)는 제1 내지 제4 연산 로직부(예컨대, 220_1~220_4)로부터 상위 비트 측으로 전달된 각 제2 합산값(예컨대, 0~S3)에 기초하여, 하위 비트 측으로부터 제1 내지 제5 연산 로직부(예컨대, 220_1~220_5)를 통해 전달받는 각 제1 합산값(예컨대, 0~S3)을 라이트 백 시킴으로써, ADD-Shift 연산 동작을 수행할 수 있다.

[0055] 실시예에 따라, 컬럼 주변 회로(200)는 ADD-Shift 연산이 수행되는 단일 주기에서 복수의 연산 로직부들(220_1~220_N)을 통해 수행되는 모든 ADD-Shift 연산 동작을 한번에 수행할 수 있다.

[0056] 구체적으로, 컬럼 주변 회로(200)는 ADD-Shift 연산이 수행되는 단일 주기에서, 제2 합산값을 상위 비트 측 방향으로 전달함에 따라 하위 비트 측 방향으로부터 전달받는 제1 합산값을 라이트 백 시키는 ADD-Shift 연산 동작을 한번에 수행할 수 있다.

[0057] 도 8은 본 출원의 다른 실시예에 따른 메모리인 컴퓨팅(11)에 대한 개념도이고, 도 9는 도 8의 메모리인 컴퓨팅

(11)에 대한 실시 예이며, 도 10은 레프트-쉬프트 곱셈연산 동작을 설명하기 위한 실시 예이다.

[0058] 도 8과 도 9를 참조하면, 메모리인 컴퓨팅(10)은 메모리 셀 어레이(100), 컬럼 주변 회로(200), 더미셀 어레이(300), BL 세퍼레이터(400) 및 시프트 레지스터 회로(500)를 포함할 수 있다. 이하, 도 1 내지 도 7에서 설명된 동일한 부재번호의 메모리 셀 어레이(100)와 컬럼 주변 회로(200)에 대한 중복된 설명을 생략될 것이다.

[0059] 먼저, 메모리 셀 어레이(100)는 적어도 둘의 워드라인 신호에 따라 활성화되는 승수 데이터와 피승수 데이터(A3, A2, A1, A0)를 사전에 저장할 수 있다. 여기서, 승수 데이터와 피승수 데이터는 곱셈 연산에 사용되는 데이터일 수 있다.

[0060] 다음으로, 컬럼 주변 회로(200)는 복수의 비트라인들을 통해 메모리 셀 어레이(100)에 연결되는 복수의 컬럼 주변 유닛들(201_1~201_8)을 포함할 수 있다.

[0061] 도 3에 도시된 바와 같이, 복수의 컬럼 주변 유닛들(201_1~201_8) 각각은 감지증폭 및 라이트부(210)와 연산 로직부(220)를 포함하고, 이때, 연산 로직부(220)는 제1 멀티플렉서(221), 시프트 플립플롭(222), 전가산기 로직(223) 제2 멀티플렉서(224) 및 제3 멀티플렉서(225)를 포함할 수 있다

[0062] 다음으로, 더미셀 어레이(300)는 메모리 셀 어레이(100)와 컬럼 주변 회로(200) 사이에 위치하여, 메모리 셀 어레이(100)에 저장된 피승수 데이터(A3, A2, A1, A0)를 저장할 수 있다.

[0063] 이러한 더미셀 어레이(300)는 컬럼 방향으로 서로 이격 배치된 제1 내지 제3 서브 어레이(310~330)를 포함할 수 있다.

[0064] 구체적으로, 제1 서브 어레이(310)는 로우 방향으로 배치되고, 0의 데이터를 저장하기 위한 복수의 더미 셀들을 포함할 수 있다. 또한, 제2 서브 어레이(320)는 로우 방향으로 배치되고, 메모리 셀 어레이(100)에 저장된 피승수 데이터(A3, A2, A1, A0)를 최하위 비트에서 최상위 비트 순서로 저장하기 위한 복수의 더미 셀들을 포함할 수 있다. 또한, 제3 서브 어레이(330)는 로우 방향으로 배치되고, 컬럼 주변 회로(200)를 통해 라이트 백 되는 임시 데이터를 저장하기 위한 복수의 더미 셀들을 포함할 수 있다.

[0065] 다음으로, BL 세퍼레이터(400)는 메모리 셀 어레이(100)와 더미셀 어레이(300) 사이에 위치하여, 메모리 셀 어레이(100)와 더미셀 어레이(300)를 분리시킬 수 있다.

[0066] 구체적으로, BL 세퍼레이터(400)는 컬럼 주변 회로(200)의 곱셈 동작 여부에 기초하여, 더미셀 어레이(300)와 메모리 셀 어레이(100) 사이를 전기적으로 차단시킬 수 있다. 예를 들면, BL 세퍼레이터(400)는 메모리 셀 어레이(100)와 더미셀 어레이(300)를 연결하는 스위치들을 스위칭 오프시킴으로써, 메모리인 컴퓨팅(11)의 커패시턴스를 줄여 연산 속도를 빠르게 증가시키고 에너지 소모를 감소시킬 수 있다.

[0067] 다음으로, 시프트 레지스터 회로(500)는 메모리 셀 어레이(100)로부터 임의의 비트라인을 통해 로드되는 승수 데이터(B3, B2, B1, B0)에 기초하여, 복수의 컬럼 주변 유닛들의 곱셈 연산 동작을 제어할 수 있다.

[0068] 이러한 시프트 레지스터 회로(500)는 복수의 승수 플립플롭들(510_1~510_N)과 복수의 제어 멀티플렉서들(520_1~520_N)을 포함할 수 있다.

[0069] 구체적으로, 복수의 승수 플립플롭들(510_1~510_N)은 승수 데이터(B3, B2, B1, B0)에 기초하여, 승수 데이터(B3, B2, B1, B0)가 최상위 비트에서 최하위 비트 순서로 출력되기 위한 로드 데이터(B0, B1, B2, B3)를 비트별로 저장할 수 있다.

[0070] 여기서, 로드 데이터는 승수 데이터(B3, B2, B1, B0)가 최상위 비트에서 최하위 비트 순서로 시프트 레지스터 회로(500)에 로드된 데이터일 수 있다. 예를 들면, 승수 데이터(B3, B2, B1, B0)가 '1011'인 경우, 로드 데이터(B0, B1, B2, B3)는 '1101'일 수 있다.

[0071] 이때, 복수의 제어 멀티플렉서들(520_1~520_N)은 로드 데이터(B0, B1, B2, B3)에 기초하여, 복수의 컬럼 주변 유닛들(201_1~201_8)을 제어하기 위한 제어 신호를 전송할 수 있다.

[0072] 예를 들면, 도 9에 도시된 바와 같이, 피승수 데이터(A3, A2, A1, A0)가 '1010'이고, 승수 데이터(B3, B2, B1, B0)가 '1011'인 경우, 복수의 승수 플립플롭들(510_1~510_N)은 로드 데이터(B0, B1, B2, B3)인 '1101'를 저장할 수 있다.

[0073] 그런 다음, 복수의 승수 플립플롭들(510_1~510_N)은 로드 데이터(B0, B1, B2, B3)를 최하위 비트에서 최상위 비트 순서로 복수의 제어 멀티플렉서들(520_1~520_N)에 전송할 수 있다.

[0074] 이때, 복수의 제어 멀티플렉서들(520_1~520_N)은 로드 데이터(B0, B1, B2, B3)에 기초하여, 컬럼 주변 회로(200)를 제어하기 위한 제어 신호를 각 제2 멀티플렉서(예컨대, 224_1~224_8)에 전송할 수 있다.

[0075] 즉, 시프트 레지스터 회로(500)는 최상위 비트에서 최하위 비트 순서로 출력되기 위한 로드 데이터(B0, B1, B2, B3)에 대응되는 제어 신호를 컬럼 주변 회로(200)에 전송함으로써, 컬럼 주변 회로(200)의 곱셈 연산 동작을 제어할 수 있다.

[0076] 실시예에 따른 컬럼 주변 회로(200)는 피승수 데이터(A3, A2, A1, A0)와 로드 데이터(B0, B1, B2, B3)에 대해 곱셈 연산 동작을 수행할 수 있다.

[0077] 구체적으로, 컬럼 주변 회로(200)는 더미 어레이(300)에 저장된 피승수 데이터(A3, A2, A1, A0)와 시프트 레지스터 회로(500)를 통해 순차적으로 로드된 로드 데이터(B0, B1, B2, B3)에 대하여, 레프트-쉬프트 곱셈연산(Left-shift Multiplication) 동작을 반복적으로 수행할 수 있다.

[0078] 예를 들면, 도 10에 도시된 바와 같이, 컬럼 주변 회로(200)는 피승수 데이터(A3, A2, A1, A0)와 B3 간의 곱인 '1010'과 0의 데이터 '0000'을 더하고, 이를 Left 방향으로 시프트시켜, 제1 임시 데이터(pMult0)인 '10100'를 획득할 수 있다. 그런 다음, 컬럼 주변 회로(200)는 피승수 데이터(A3, A2, A1, A0)와 B2 의 곱인 '0000'와 제1 임시 데이터를 더하고, 이를 Left 방향으로 시프트시켜, 제2 임시 데이터인 '010100'를 획득할 수 있다. 그런 다음, 컬럼 주변 회로(200)는 피승수 데이터(A3, A2, A1, A0)와 B1 의 곱인 '1010'와 제2 임시 데이터를 더하고, 이를 Left 방향으로 시프트시켜, 제3 임시 데이터인 '0110010'를 획득할 수 있다. 이후, 컬럼 주변 회로(200)는 피승수 데이터(A3, A2, A1, A0)와 B0 의 곱인 '1010'와 제3 임시 데이터를 더하여, 피승수 데이터(A3, A2, A1, A0)와 승수 데이터(B3, B2, B1, B0) 간의 곱셈 데이터(MUL_D)인 '01101110'를 획득할 수 있다.

[0079] 이하, 도 11 내지 도 18을 참조하여, 컬럼 주변 회로(200)의 레프트-쉬프트 곱셈연산(Left-shift Multiplication) 동작에 대해 보다 구체적으로 설명될 것이다.

[0080] 도 11과 도 12는 도 10의 제1 임시 데이터(pMult1)를 획득하기 위한 메모리인 컴퓨팅(11)의 동작 예시도이다.

[0081] 도 11과 도 12를 참조하면, 컬럼 주변 회로(200)는 승수 데이터(B3, B2, B1, B0)의 최상위 비트(MSB)에 대응되는 제어 신호에 기초하여, 제1 및 제2 서브 어레이(310, 320)에 대한 컬럼별 ADD-Shift 동작을 수행할 수 있다. 이러한 컬럼별 ADD-Shift 동작은 시프트 후 덧셈 알고리즘을 이용하는 종래의 곱셈 연산 동작에 따라 발생하는 레이턴시를 개선시킬 수 있다.

[0082] 예를 들면, 도 11에 도시된 바와 같이, 컬럼 주변 회로(200)는 시프트 레지스터 회로(500)를 통해 전송받는 승수 데이터(B3, B2, B1, B0)의 최상위 비트(B3)에 대응되는 제어 신호에 기초하여, 각 전가산기 로직(예컨대, 223_1~223_8)을 통해 제1 및 제2 서브 어레이(310, 320)에 대한 컬럼별 ADD 연산 값인 '1010'을 출력할 수 있다.

[0083] 이때, 컬럼 주변 회로(200)는 각 제2 멀티플렉서(224_1~224_8)를 이용하여, 컬럼별 ADD 연산 값을 상위 비트 측으로 시프트하는 Left-shift 연산 동작을 수행할 수 있다.

[0084] 그런 다음, 컬럼 주변 회로(200)는 도 12에 도시된 바와 같이, 각 시프트 플립플롭(222_1~2228)으로부터 각 제3 멀티플렉서(225_1~225_8)를 통해 라이트 백 되는 제1 임시 데이터(pMult1)를 제3 서브 어레이(330)에 저장할 수 있다. 여기서, 제1 임시 데이터(pMult1)는 '10100'일 수 있다.

[0085] 도 13과 도 14는 도 10의 제2 임시 데이터(pMult2)를 획득하기 위한 메모리인 컴퓨팅(11)의 동작 예시도이다.

[0086] 도 13과 도 14를 참조하면, 컬럼 주변 회로(200)는 승수 데이터(B3, B2, B1, B0)의 중간 비트(B2 또는 B1)가 0에 대응되는 제어 신호에 기초하여, 제3 서브 어레이(330)에 대한 Shift 연산 동작을 수행할 수 있다.

[0087] 예를 들면, 도 13에 도시된 바와 같이, 컬럼 주변 회로(200)는 시프트 레지스터 회로(500)를 통해 전송받는 중간 비트(B2)가 0에 대응되는 제어 신호에 기초하여, 시프트 플립플롭에 저장된 제1 임시 데이터(pMult1)를 각 제2 멀티플렉서(224_1~224_8)를 통해 상위 비트 측으로 시프트하는 Left-shift 연산 동작을 수행할 수 있다.

[0088] 그런 다음, 도 14에 도시된 바와 같이, 컬럼 주변 회로(200)는 각 시프트 플립플롭(222_1~222_8)으로부터 각 제3 멀티플렉서(225_1~225_8)를 통해 라이트 백되는 제2 임시 데이터(pMult2)를 제3 서브 어레이(330)에 저장할 수 있다. 여기서, 제2 임시 데이터(pMult2)는 '101000'일 수 있다.

[0089] 도 15와 도 16은 도 10의 제3 임시 데이터(tMult)를 획득하기 위한 메모리인 컴퓨팅(11)의 동작 예시도이다.

[0090] 도 15와 도 16을 참조하면, 컬럼 주변 회로(200)는 승수 데이터(B3, B2, B1, B0)의 중간 비트(B2 또는 B1)가 1에 대응되는 제어 신호에 기초하여, 제2 및 제3 서브 어레이(320, 330)에 대해 저장된 컬럼별 ADD-Shift 연산 동작을 수행할 수 있다.

[0091] 예를 들면, 도 15에 도시된 바와 같이, 컬럼 주변 회로(200)는 시프트 레지스터 회로(500)를 통해 전송받는 중간 비트(B1) 1에 대응되는 제어 신호에 기초하여, 각 전가산기 로직(223_1~223_8)을 통해 제2 및 제3 서브 어레이(320, 330)에 대한 컬럼별 ADD 연산 값인 '110010'을 출력할 수 있다.

[0092] 이때, 컬럼 주변 회로(200)는 각 제2 멀티플렉서(224_1~224_8)를 통해 컬럼별 ADD 연산 값에 대해 상위 비트 측으로 Left-Shift 동작을 수행할 수 있다.

[0093] 그런 다음, 컬럼 주변 회로(200)는 도 16에 도시된 바와 같이, 각 시프트 플립플롭(222_1~2228)으로부터 각 제3 멀티플렉서(225_1~225_8)를 통해 라이트 백 되는 제3 임시 데이터(tMult)를 제3 서브 어레이(330)에 저장할 수 있다. 여기서, 제3 임시 데이터는 '1100100'일 수 있다.

[0094] 도 17은 도 10의 곱셈 데이터(MUL_D)를 획득하기 위한 메모리인 컴퓨팅(11)의 동작 예시도이다.

[0095] 도 17을 참조하면, 컬럼 주변 회로(200)는 승수 데이터의 최하위 비트(B0)에 대응되는 제어 신호에 기초하여, 제2 및 제3 서브 어레이(320, 330)에 대한 컬럼별 ADD 연산 동작을 수행할 수 있다.

[0096] 구체적으로, 컬럼 주변 회로(200)는 제2 서브 어레이(320)에 저장된 피승수 데이터(A3, A2, A1, A0)와 제3 서브 어레이(330)에 저장된 제3 임시 데이터(tMult)에 대해 컬럼별 ADD 연산 동작을 수행할 수 있다.

[0097] 예를 들면, 도 17에 도시된 바와 같이, 컬럼 주변 회로(200)는 시프트 레지스터 회로(500)를 통해 전송받는 최하위 비트에 대응되는 제어 신호에 기초하여, 각 전가산기 로직(223_1~223_8)을 통해 제2 및 제3 서브 어레이(320, 330)에 대한 컬럼별 ADD 연산 값인 '01101110'을 출력할 수 있다.

[0098] 그런 다음, 컬럼 주변 회로(200)는 각 제2 멀티플렉서(224_1~224_8)로부터 각 제3 멀티플렉서(225_1~225_8)를 통해 라이트 백 되는 곱셈 데이터를 제3 서브 어레이(330)에 저장할 수 있다. 여기서, 곱셈 데이터(MUL_D)는 '01101110'일 수 있다.

[0099] 본 출원은 도면에 도시된 일 실시 예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 출원의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

[0100] 10, 11: 인메모리 컴퓨팅

100: 메모리 어레이

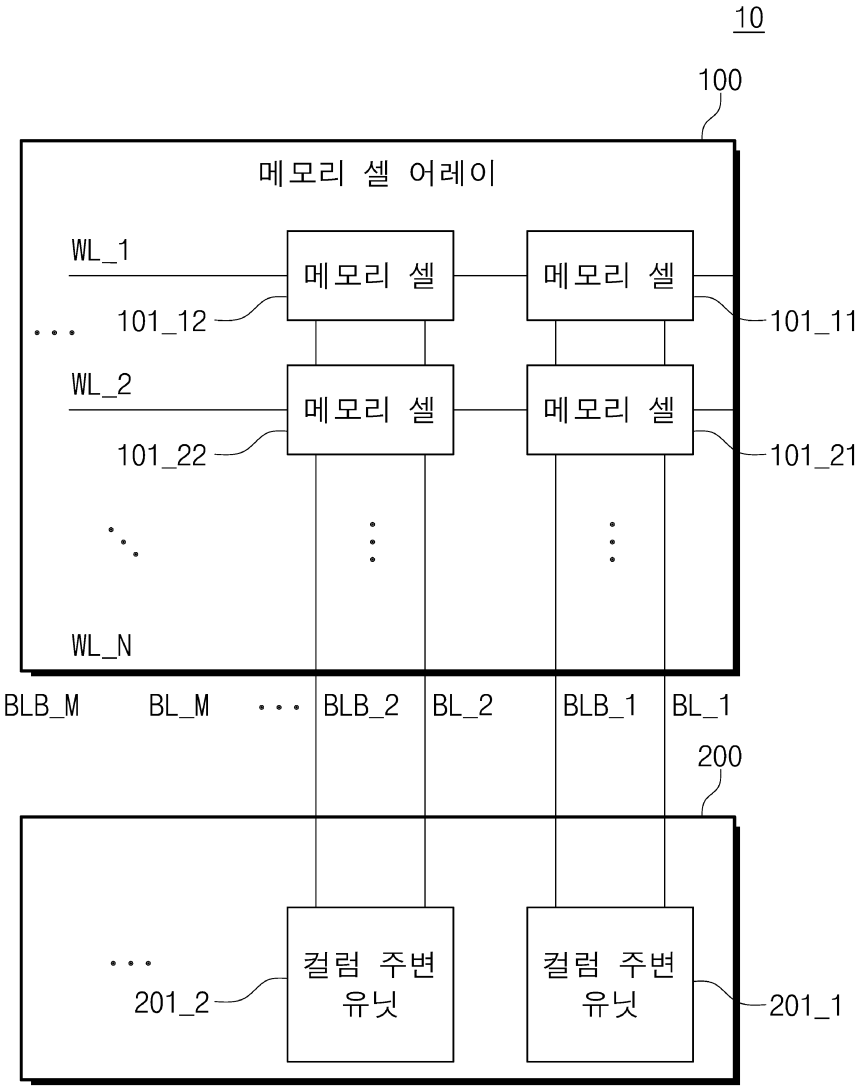
200: 컬럼 주변 회로

210: 감지증폭 및 라이트부

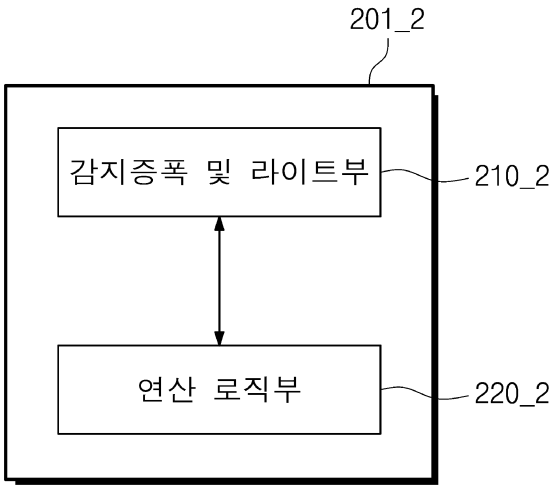
220: 연산 로직부

도면

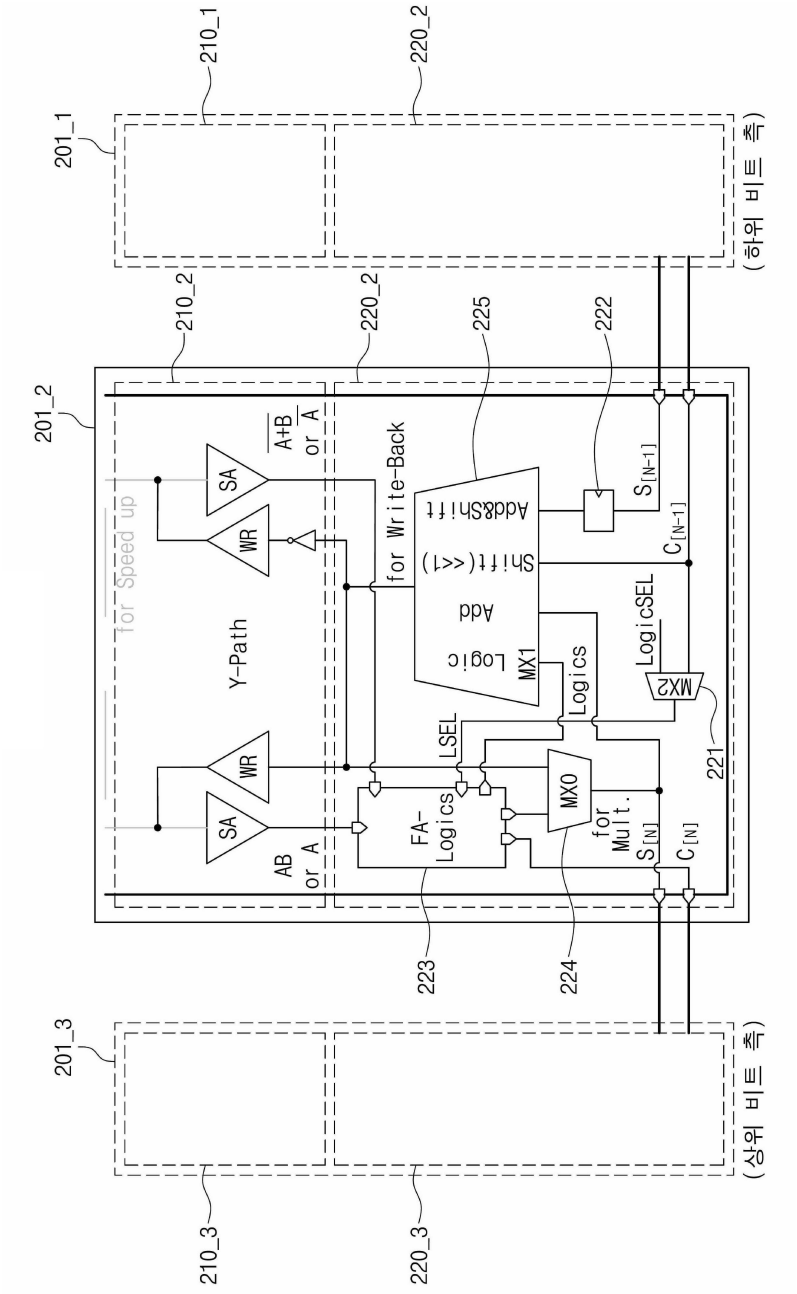
도면1


10
100
메모리 셀 어레이
WL_1
메모리 셀
메모리 셀
101_12
101_11
WL_2
메모리 셀
메모리 셀
101_22
101_21
WL_N
BLB_M
BL_M
BLB_2
BL_2
BLB_1
BL_1
200
컬럼 주변 유닛
컬럼 주변 유닛
201_2
201_1

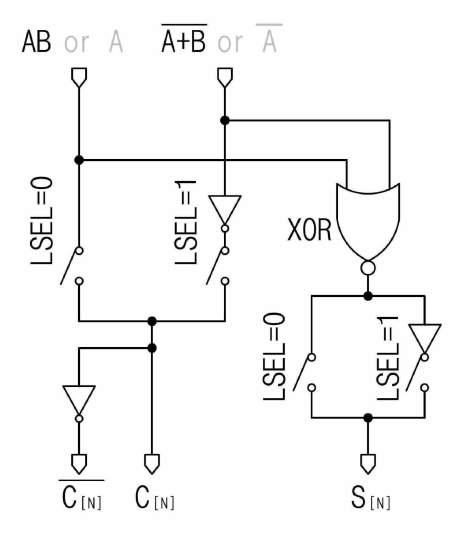
도면2



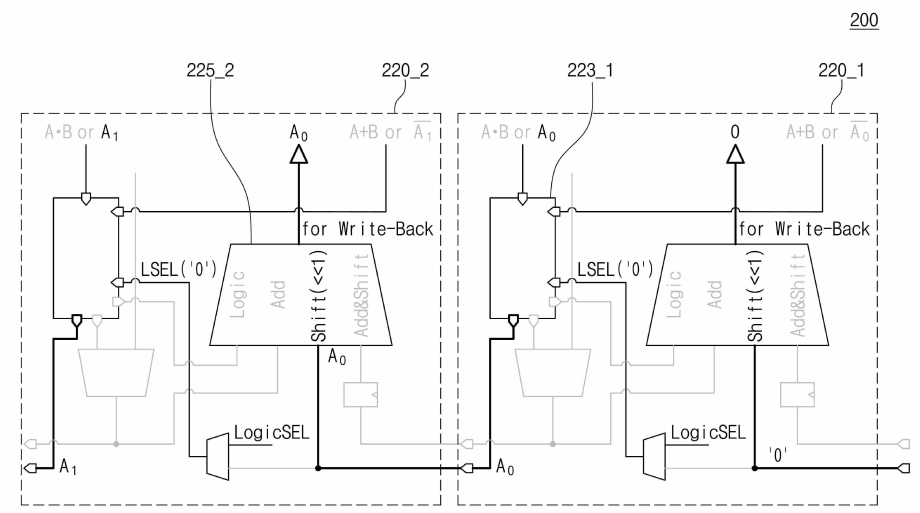
도면3


210_1
220_2
201_1
(하위 비트 측)
210_2
220_2
225
222
201_2
for Speed up
SA
WR
Y-Path
A+B or A
for Write-Back
Add&Shift
Shift(<<1)
Add
Logic
MX1
Logics
LogicSEL
MX2
221
S[N-1]
C[N-1]
LSEL
WR
SA
AB or A
FA-Logics
MX0
for Mult.
S[N]
C[N]
223
224
201_3
(상위 비트 측)
210_3
220_3

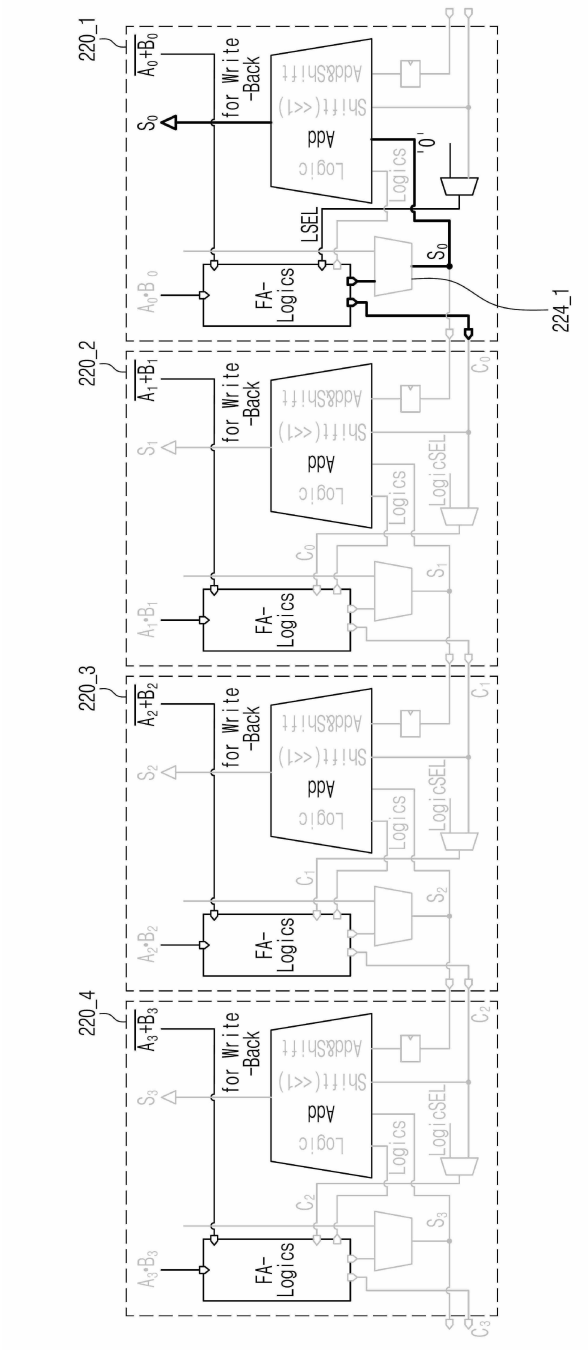
도면4


AB or A
A+B or Ā
LSEL=0
LSEL=1
XOR
LSEL=0
LSEL=1
C̄[N]
C[N]
S[N]

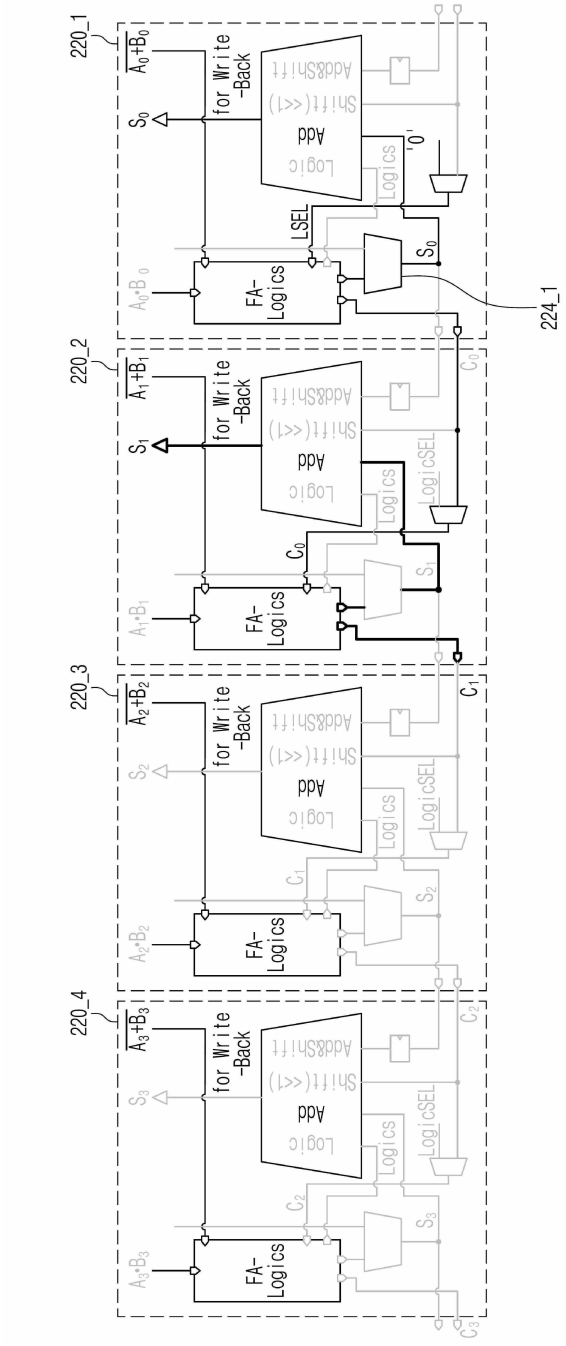
도면5


200
225_2
220_2
223_1
220_1
A·B or A1
A0
A+B or Ā1
A·B or A0
0
A+B or Ā0
for Write-Back
LSEL('0')
Logic
Add
Shift(<<1)
Add&Shift
A0
LogicSEL
A1
'0'

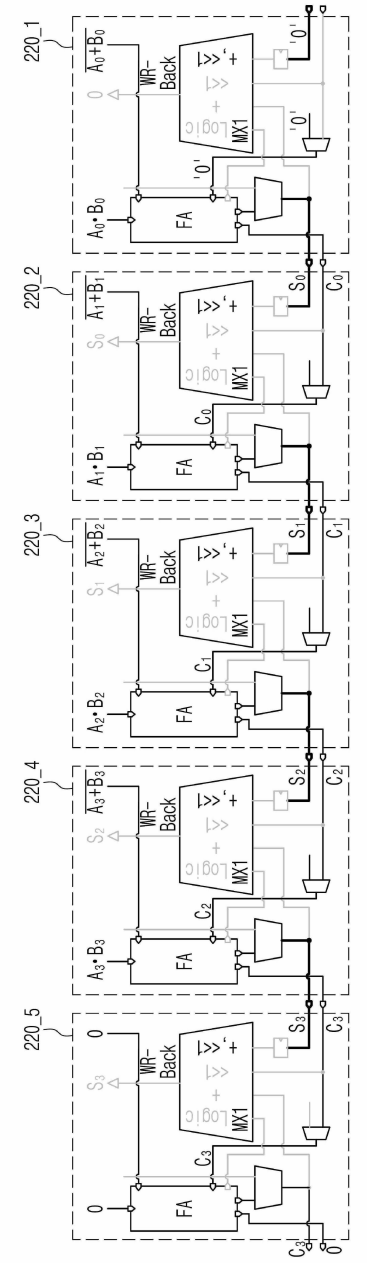
도면6a


220_1
220_2
220_3
220_4
for Write -Back
Add
Shift(<<1)
AddShift
Logic
Logics
LogicSEL
FA-Logics
LSEL
'0'
224_1

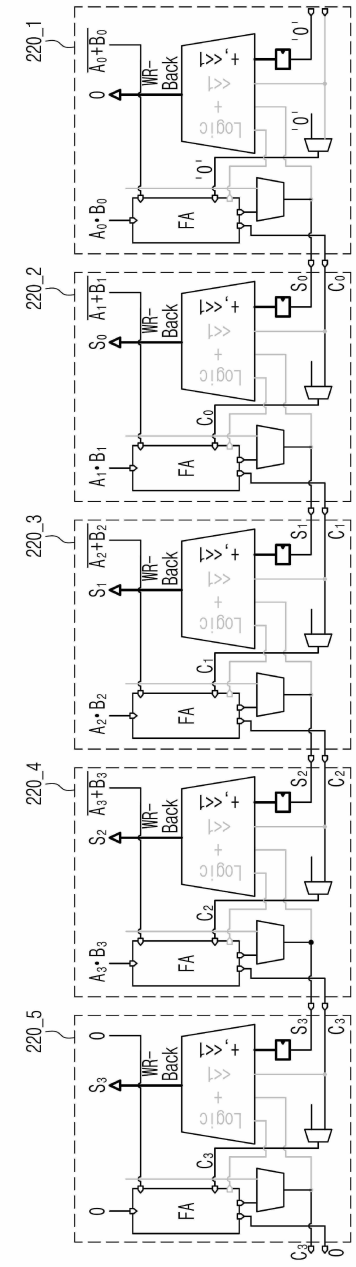
도면6b


220_1
220_2
220_3
220_4
for Write -Back
Add
Shift(<<1)
AddShift
Logic
Logics
LogicSEL
LSEL
'0'
FA-Logics
224_1

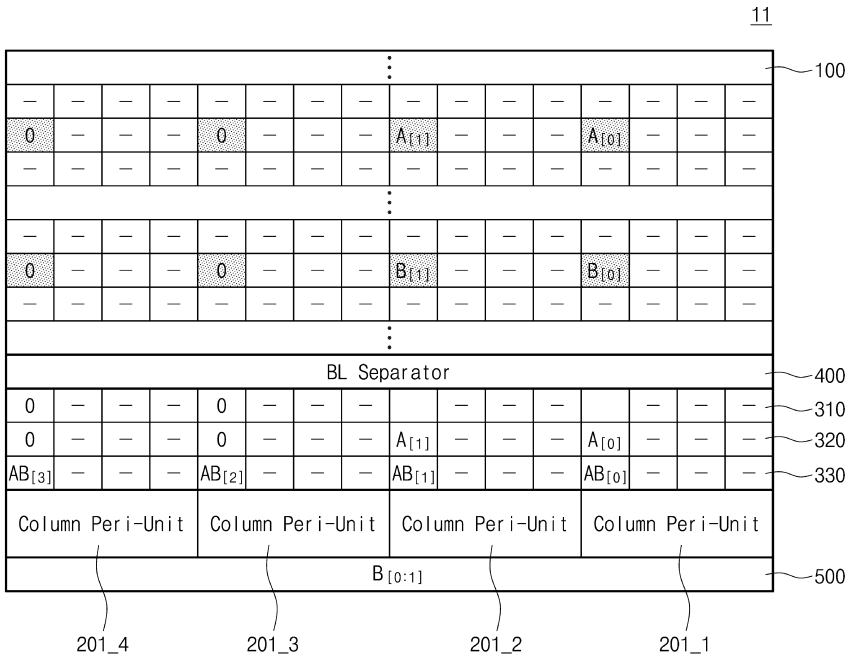
도면7a



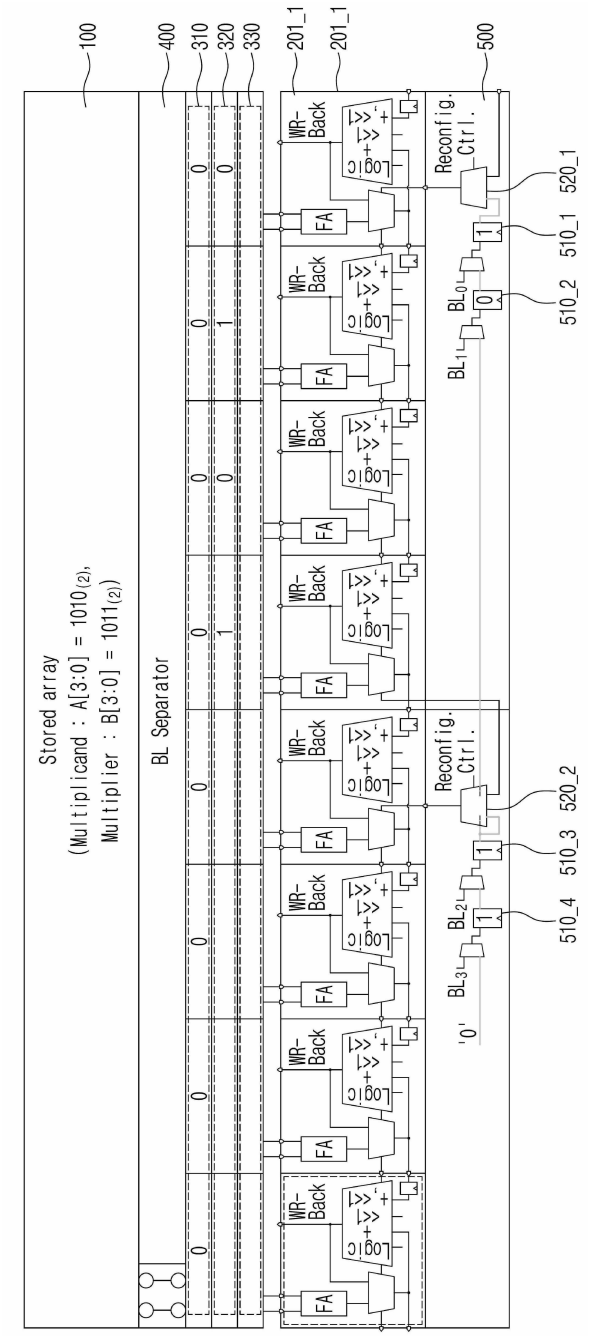
도면7b



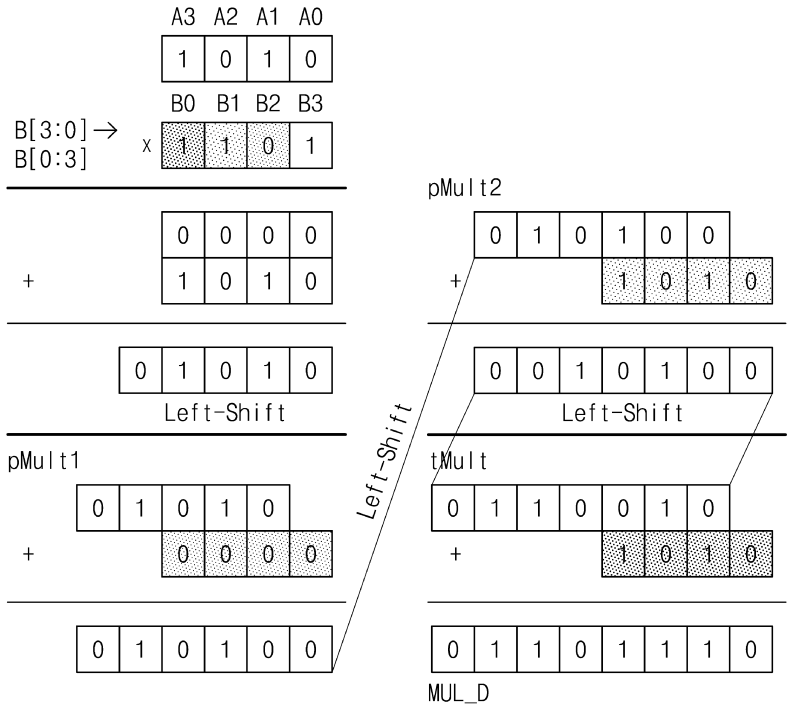
도면8


11
100
0
A[1]
A[0]
B[1]
B[0]
BL Separator
400
310
320
330
AB[3]
AB[2]
AB[1]
AB[0]
Column Peri-Unit
Column Peri-Unit
Column Peri-Unit
Column Peri-Unit
B[0:1]
500
201_4
201_3
201_2
201_1

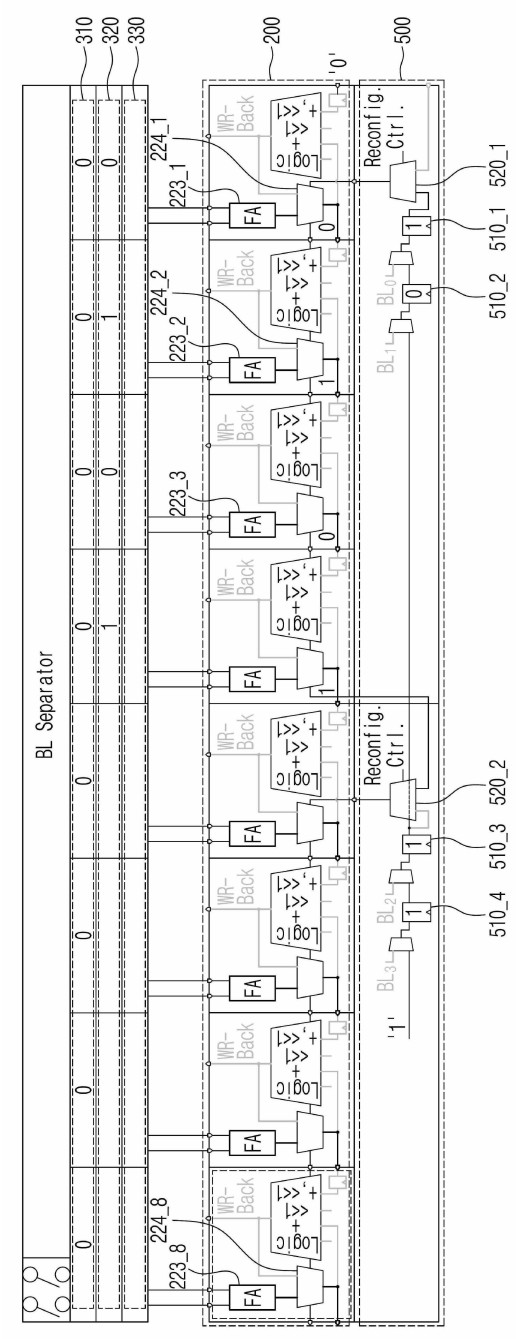
도면9


100
400
310
320
330
201_1
201_1
500
Stored array
(Multiplicand : A[3:0] = 1010(2),
Multiplier : B[3:0] = 1011(2))
BL Separator
WR-Back
FA
Reconfig. Ctrl.
520_1
520_2
510_1
510_2
510_3
510_4
BL0
BL1
BL2
BL3
'0'

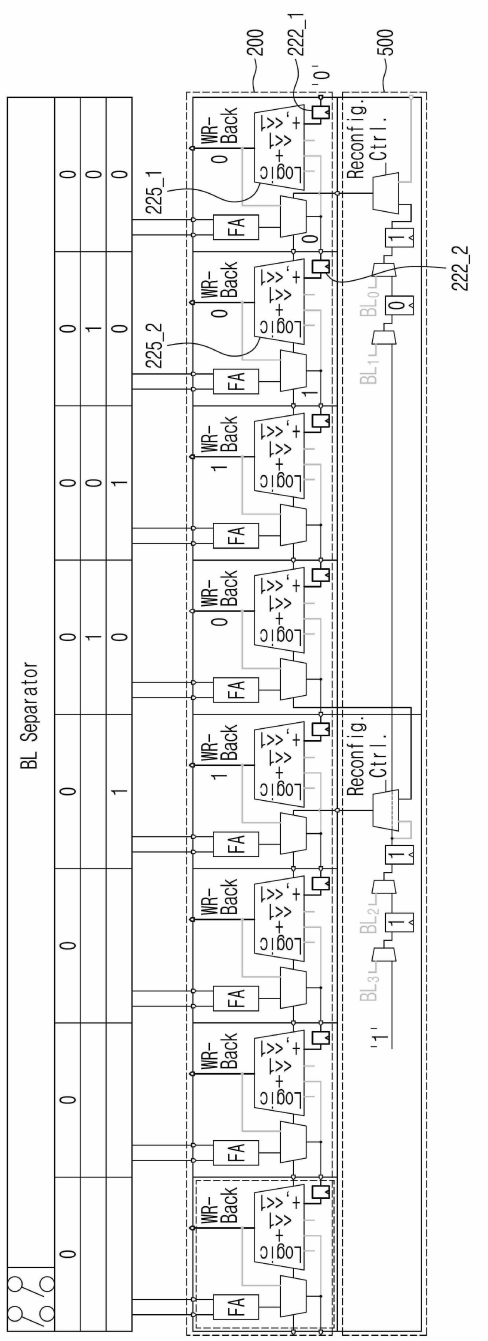
도면10


A3 A2 A1 A0
1 0 1 0
B0 B1 B2 B3
B[3:0]→
B[0:3]
x
1 1 0 1
+
0 0 0 0
1 0 1 0
0 1 0 1 0
Left-Shift
pMult1
0 1 0 1 0
+
0 0 0 0
0 1 0 1 0 0
Left-Shift
pMult2
0 1 0 1 0 0
+
1 0 1 0
0 0 1 0 1 0 0
Left-Shift
tMult
0 1 1 0 0 1 0
+
1 0 1 0
0 1 1 0 1 1 1 0
MUL_D

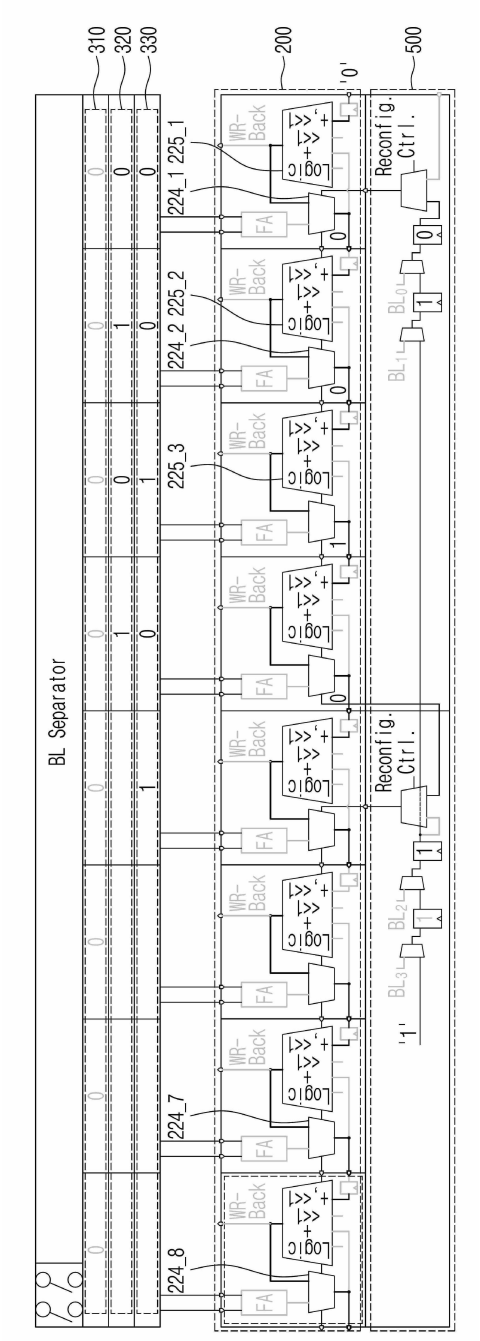
도면11


310
320
330
200
500
BL Separator
Reconfig. Ctrl.
Reconfig. Ctrl.
WR-Back
FA
520_1
520_2
510_1
510_2
510_3
510_4

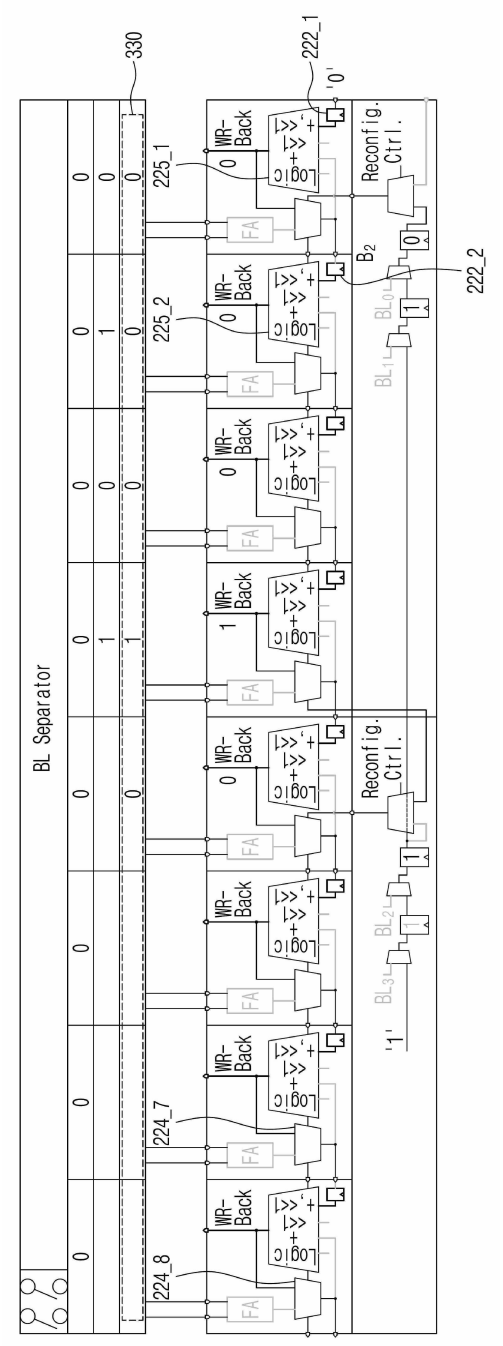
도면12


200
222_1
500
225_1
225_2
222_2
BL Separator
WR-Back
FA
Reconfig. Ctrl.
'0'
'1'

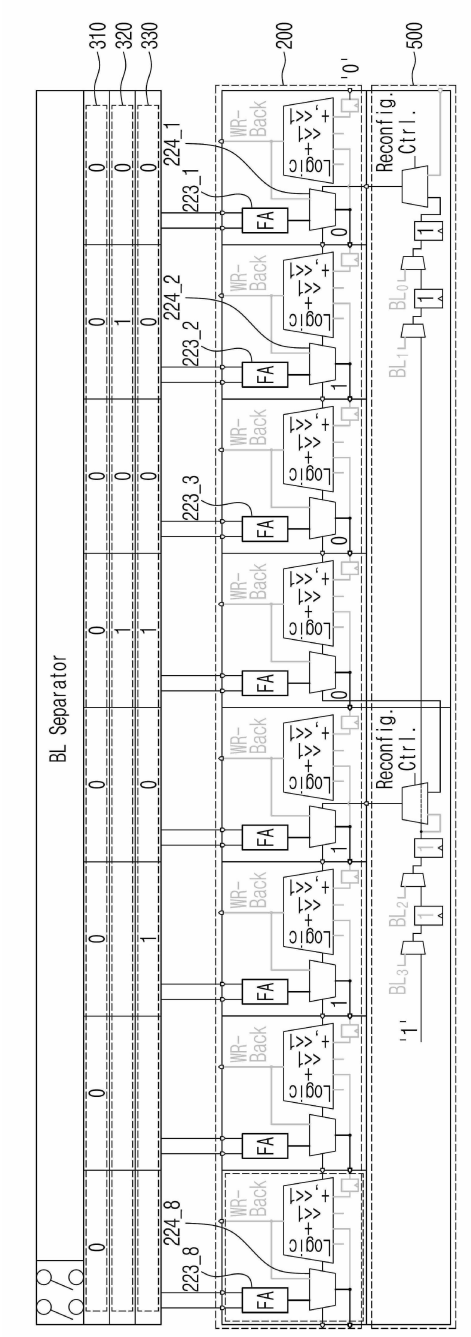
도면13


310
320
330
200
500
'0'
BL Separator
224_1
225_1
224_2
225_2
225_3
224_7
224_8
WR-Back
FA
Reconfig. Ctrl.
'1'

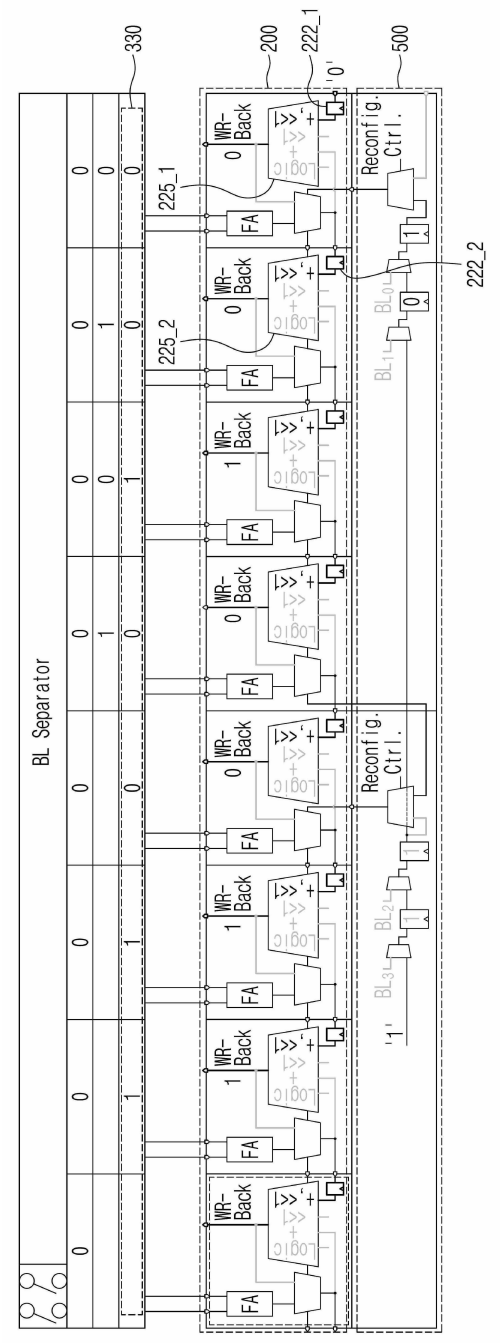
도면14


330
222_1
'0'
225_1
225_2
224_7
224_8
BL Separator
WR-Back
FA
Reconfig. Ctrl.
B2
222_2
'1'

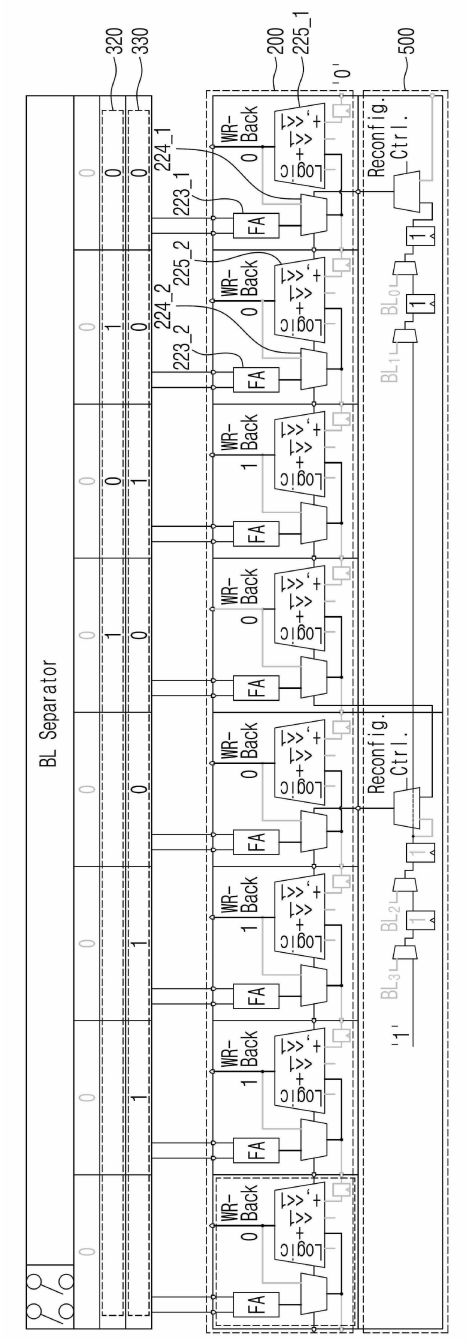
도면15


310
320
330
200
500
'0'
'1'
BL Separator
Reconfig. Ctrl.
Reconfig. Ctrl.
WR-Back
FA
224_1
223_1
224_2
223_2
223_3
224_8
223_8

도면16


330
200
222_1
500
BL Separator
WR-Back
FA
225_1
225_2
Reconfig. Ctrl.
222_2
'0'
'1'

도면17


320
330
200
225_1
'0'
500
224_1
223_1
225_2
224_2
223_2
BL Separator
WR-Back
FA
Reconfig. Ctrl.
'1'