



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년06월13일
(11) 등록번호 10-1626040
(24) 등록일자 2016년05월25일

(51) 국제특허분류(Int. Cl.)
G06F 11/10 (2006.01) G06F 13/16 (2006.01)
(52) CPC특허분류
G06F 11/1044 (2013.01)
G06F 13/1668 (2013.01)
(21) 출원번호 10-2015-7012194
(22) 출원일자(국제) 2013년10월08일
심사청구일자 2015년05월27일
(85) 번역문제출일자 2015년05월08일
(65) 공개번호 10-2015-0070252
(43) 공개일자 2015년06월24일
(86) 국제출원번호 PCT/US2013/063881
(87) 국제공개번호 WO 2014/058879
국제공개일자 2014년04월17일
(30) 우선권주장
13/649,745 2012년10월11일 미국(US)
(56) 선행기술조사문헌
JP2007115390 A
W02002091382 A2
US07620875 B1

(73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 94088-3453 서니베일 피.오.박스
3453 원 에이엠디 플레이스
(72) 발명자
로 가브리엘 에이치.
미국 워싱턴 98007 벨레뷰 엔이 12 스트리트
15115
스리드하란 빌라스 케이.
미국 매사추세츠 02446 브룩라인 웨스트본 테라스
#3 33
(74) 대리인
박장원

전체 청구항 수 : 총 11 항

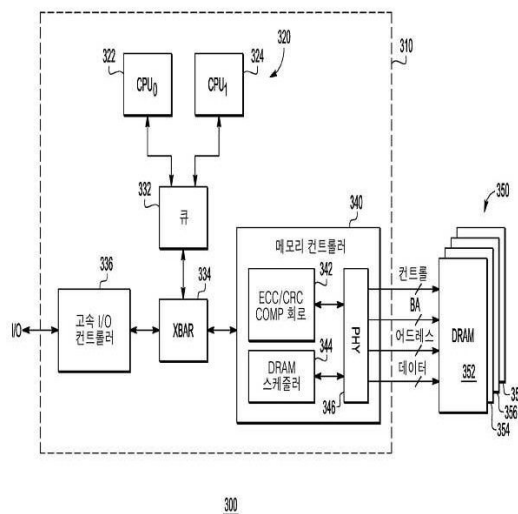
심사관 : 이동하

(54) 발명의 명칭 높은 신뢰도 메모리 컨트롤러

(57) 요약

집적 회로는 어드레스 공간을 갖는 메모리 및 메모리에 결합되어 수신된 메모리 액세스들에 응답하여 어드레스 공간에 액세스하는 메모리 컨트롤러를 포함한다. 메모리 컨트롤러는 어드레스 공간의 제 1 부분 내의 복수의 데이터 요소들, 및 어드레스 공간의 제 2 부분 내의 복수의 데이터 요소들에 대응하는 신뢰도 데이터에 더 액세스한다.

대표도



300

명세서

청구범위

청구항 1

어드레스 공간(600)을 갖는 메모리(350); 및

상기 메모리(350)에 결합되어 수신된 메모리 액세스들에 응답하여 상기 어드레스 공간(600)에 액세스하는 메모리 컨트롤러(340)로서, 상기 메모리 컨트롤러(340)는 상기 어드레스 공간(600)의 제 1 부분 내의 복수의 데이터 요소들에 더 액세스하고, 신뢰도 데이터는 상기 어드레스 공간(600)의 제 2 부분 내의 상기 복수의 데이터 요소들에 대응하는, 상기 메모리 컨트롤러를 포함하고,

상기 어드레스 공간(600)은 상기 어드레스 공간(600) 내에서 순서를 갖는 복수의 बैं크들(610, 620, 630, 640)을 포함하고;

상기 어드레스 공간(600)의 상기 제 1 부분은 상기 순서로 상기 복수의 बैं크들(610, 620, 630, 640) 중에 분배되는 미리 결정된 수의 데이터 요소들의 복수의 그룹들을 포함하고;

상기 어드레스 공간의 상기 제 2 부분은 각각의 상기 복수의 그룹들의 각 대응하는 데이터 요소에 대한 신뢰도 데이터 요소를 포함하며, 그룹에 대한 신뢰도 데이터 요소들은 상기 순서로 상기 그룹의 마지막 데이터 요소를 포함하는 제 2 बैं크(640) 후에 제 1 बैं크(610)에 위치되는 집적 회로(300).

청구항 2

청구항 1에 있어서,

상기 메모리 액세스들을 생성하고 상기 메모리 액세스들을 상기 메모리 컨트롤러(340)에 제공하는 메모리 액세스 생성 회로(320)를 더 포함하는 집적 회로(300).

청구항 3

청구항 2에 있어서,

상기 메모리 액세스 생성 회로(320)는 중앙 처리 유닛 코어를 포함하는 집적 회로(300).

청구항 4

청구항 2에 있어서,

상기 메모리 액세스 생성 회로(320) 및 상기 메모리 컨트롤러(340)는 단일 집적 회로 다이(310) 상에 결합되는 집적 회로(300).

청구항 5

청구항 1에 있어서,

상기 메모리(350)는 메모리 칩 스택(140, 240) 내에 배열되는 복수의 메모리 칩들(352, 354, 356, 358)을 포함하는 집적 회로(300).

청구항 6

청구항 1에 있어서,

상기 신뢰도 데이터는 각각의 상기 데이터 요소들에 대한 적어도 하나의 ECC를 포함하는 복수의 오류 정정 코드들(ECCs; error correcting codes)을 포함하는 집적 회로(300).

청구항 7

청구항 1에 있어서,

상기 신뢰도 데이터는 각각의 상기 데이터 요소들에 대한 적어도 하나의 CRC를 포함하는 복수의 순환 중복 검사(CRC; cyclic redundancy check) 코드들을 포함하는 집적 회로(300).

청구항 8

리퀘스터로부터 데이터 요소에 대한 기록 액세스를 수신하는 단계(810);

상기 데이터 요소에 대한 신뢰도 데이터를 산출하는 단계(820); 및

상기 데이터 요소를 어드레스 공간(600)의 제 1 부분에 저장하는 단계(830) 및 상기 신뢰도 데이터를 상기 어드레스 공간(600)의 제 2 부분에 저장하는 단계(840)를 포함하고,

상기 저장하는 단계(830, 840)는:

상기 어드레스 공간(600) 내에서 순서를 갖는 복수의 बैं크들(610, 620, 630, 640) 중에서 상기 어드레스 공간(600)을 분할하는 단계;

상기 어드레스 공간(600)의 상기 제 1 부분을 상기 순서로 상기 복수의 बैं크들(610, 620, 630, 640) 중에 분배되는 미리 결정된 수의 데이터 요소들의 복수의 그룹들로서 형성하는 단계; 및

각각의 상기 복수의 그룹들의 각 대응하는 데이터 요소에 대한 신뢰도 데이터 요소를 포함하는 상기 어드레스 공간(600)의 상기 제 2 부분을 형성하는 단계로서, 그룹에 대한 신뢰도 데이터 요소들은 상기 순서로 상기 그룹의 마지막 데이터 요소를 포함하는 제 2 बैं크(640) 후에 제 1 बैं크(610)에 위치되는, 상기 제 2 부분을 형성하는 단계를 포함하는 방법.

청구항 9

청구항 8에 있어서,

상기 리퀘스터로부터 상기 데이터 요소에 대한 판독 액세스를 수신하는 단계(900);

상기 어드레스 공간(600)의 상기 제 1 부분으로부터 상기 데이터 요소를 판독하는 단계(920); 및

상기 신뢰도 데이터가 정확하게 판독되었는지를 판단하는 단계(930)를 더 포함하는 방법.

청구항 10

청구항 9에 있어서, 상기 판단하는 단계(930)는,

산출된 신뢰도 데이터를 형성하기 위해 상기 어드레스 공간(600)의 상기 제 1 부분으로부터 판독된 상기 데이터 요소에 대한 신뢰도 데이터를 산출하는 단계(932);

저장된 신뢰도 데이터를 형성하기 위해 상기 어드레스 공간(600)의 상기 제 2 부분에 저장된 상기 신뢰도 데이터를 판독하는 단계(934); 및

상기 산출된 신뢰도 데이터를 상기 저장된 신뢰도 데이터와 비교하는 단계(936)를 포함하는 방법.

청구항 11

청구항 10에 있어서,

상기 저장된 신뢰도 데이터가 상기 산출된 신뢰도 데이터와 일치할 때, 상기 데이터 요소를 상기 리퀘스터(320)로 복귀시키는 단계(942);

상기 저장된 신뢰도 데이터가 상기 산출된 신뢰도 데이터와 일치하지 않을 때, 상기 신뢰도 데이터가 정정될 수 있는지를 판단하는 단계(944);

상기 신뢰도 데이터가 정정될 수 있을 때, 상기 데이터를 정정하고(948) 정정된 데이터를 상기 리퀘스터(320)로 복귀시키는 단계; 및

상기 신뢰도 데이터가 정정될 수 없을 때, 오류를 상기 리퀘스터(320)에 보고하는 단계(950)를 더 포함하는 방법.

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

발명의 설명

기술 분야

[0001] 본 발명은 일반적으로 집적 회로들에 관한 것으로, 보다 구체적으로 메모리 컨트롤러들을 갖는 집적 회로들에 관한 것이다.

배경 기술

[0002] 소비자들은 더 높은 성능 및 더 낮은 비용을 갖는 컴퓨터 시스템들을 계속 요구한다. 더 높은 성능 요건들을 처리하기 위해, 컴퓨터 칩 설계자들은 단일 칩 상에 다수의 프로세서 코어들을 갖는 집적 회로들을 개발해왔다. 게다가, 멀티 코어 집적 마이크로프로세서 및 연관된 메모리 칩들을 단일 구성요소로 패키징하는 다양한 다이 적층 집적 기술들이 개발되어왔다. 그러나, 메모리 칩들은 다양한 결합 상태들에 민감하다. 적층 다이 구성들에 사용되는 메모리 칩들의 경우에, 영구적 결합이 발생할 때, 스택에서 모든 다른 칩들을 교체하는 것 없이 메모리 칩을 용이하게 교체하는 것이 가능하지 않다.

발명의 내용

과제의 해결 수단

[0003] 어드레스 공간을 갖는 메모리, 및 메모리에 결합되어 수신된 메모리 액세스들에 응답하여 어드레스 공간에 액세스하는 메모리 컨트롤러를 포함하는 집적 회로가 제공된다. 메모리 컨트롤러는 상기 어드레스 공간의 제 1 부분 내의 복수의 데이터 요소들, 및 상기 어드레스 공간의 제 2 부분 내의 복수의 데이터 요소들에 대응하는 신뢰도 데이터에 더 액세스한다.

[0004] 메모리 액세스 생성 회로 및 메모리 컨트롤러를 포함하는 집적 회로가 또한 제공된다. 메모리 액세스 생성 회로는 메모리의 어드레스 공간에서 데이터 요소들에 대한 메모리 액세스들을 생성한다. 메모리 컨트롤러는 수신된 메모리 액세스들에 응답하여 어드레스 공간에 액세스하기 위해 메모리에 결합된다. 메모리 컨트롤러는 어드레스 공간의 제 1 부분 내의 복수의 데이터 요소들, 및 어드레스 공간의 제 2 부분 내의 복수의 데이터 요소들에 대응하는 신뢰도 데이터에 더 액세스한다.

[0005] 제 1 데이터 요소에 대한 기록 액세스가 리퀘스터로부터 수신되는 방법이 제공된다. 신뢰도 데이터는 데이터 요소에 대해 산출된다. 데이터 요소는 어드레스 공간의 제 1 부분에 저장되고, 신뢰도 데이터는 어드레스 공간의 제 2 부분에 저장된다.

도면의 간단한 설명

[0006] 도 1은 일부 실시예들에 따른 물리 메모리를 구현하는 제 1 멀티 칩 모듈의 사시도를 예시한다.

도 2는 일부 실시예들에 따른 물리 메모리를 구현하는 제 2 멀티 칩 모듈의 사시도를 예시한다.

도 3은 일부 실시예들에 따른 높은 신뢰도 메모리 컨트롤러를 갖는 집적 회로를 블록도 형태로 예시한다.

도 4는 일부 실시예들에 따른 도 3의 메모리에 대한 어드레스 공간의 표현을 예시한다.

도 5는 일부 실시예들에 따른 도 3의 메모리에 대한 어드레스 공간의 다른 표현을 예시한다.

도 6은 일부 실시예들에 따른 도 3의 메모리에 대한 어드레스 공간의 다른 표현을 예시한다.

도 7은 일부 실시예들에 따른 도 3의 메모리에 대한 어드레스 공간의 다른 표현을 예시한다.

도 8은 일부 실시예들에 따른 데이터를 기록하는 방법의 흐름도를 예시한다.

도 9는 일부 실시예들에 따른 데이터를 판독하는 방법의 흐름도를 예시한다.

이하의 설명에서, 상이한 도면들 내의 동일한 참조 숫자들의 사용은 유사 또는 동일한 항목들을 지시한다. 다르게 언급되지 않으면, 단어 "결합된" 및 그것의 연관된 동사 형태들은 본 기술분야에 공지된 수단에 의해 직접 연결 및 간접 전기적 연결 양자를 포함하고, 다르게 언급되지 않으면 직접 연결의 임의의 설명은 또한 간접 전기적 연결의 적절한 형태들을 사용하는 대체 실시예들을 암시한다.

발명을 실시하기 위한 구체적인 내용

[0007] 도 1은 일부 실시예들에 따른 물리 메모리를 구현하는 제 1 멀티 칩 모듈의 사시도를 예시한다. 멀티 칩 모듈(100)은 일반적으로 멀티 코어 프로세서 칩(120) 및 메모리 칩 스택(140)을 포함한다. 메모리 칩 스택(140)은 서로의 상단에 적층되는 복수의 메모리 칩들을 포함한다. 도 1에 예시된 바와 같이, 메모리 칩 스택(140)은 메모리 칩(142), 메모리 칩(144), 메모리 칩(146), 및 메모리 칩(148)을 포함한다. 일반적으로, 메모리 칩 스택(140)은 도 1에 예시된 것보다 더 많거나 더 적은 메모리 칩들을 포함할 수 있다는 점을 주목한다. 메모리 칩 스택(140)의 각각의 개별 메모리 칩은 적절한 시스템 동작을 위해 요구되는 바와 같이, 메모리 칩 스택(140)의 다른 메모리 칩들에 연결된다. 메모리 칩 스택(140)의 각각의 개별 메모리 칩은 또한 적절한 시스템 동작을 위해 요구되는 바와 같이, 멀티 코어 칩(120)에 연결된다.

[0008] 동작 시, 멀티 칩 모듈(100)의 구성요소들은 단일 집적 회로 패키지에 결합되며, 여기서 메모리 칩 스택(140) 및 멀티 코어 칩(120)은 단일 집적 회로로서 사용자에게 나타난다. 멀티 코어 칩(120)에 메모리 칩 스택(140)의 전기적 연결은 수평 인터커넥트와 결합하여, 수직 인터커넥트, 예를 들어 비아(via) 또는 실리콘 관통 구멍을 사용하여 달성된다. 멀티 코어 프로세서 다이(120)는 메모리 칩 스택(140) 내의 메모리 칩들보다 더 두껍고 메모리 칩 스택(140)을 물리적으로 지원한다. 5개의 개별 칩들과 비교될 때, 멀티 칩 모듈(100)은 일반적으로 구성요소 액세스 시간을 감소시키고 시스템 성능을 증가시킴과 동시에, 시스템 비용 및 보드 공간을 절감한다. 그러나, 메모리 칩들은 다양한 신뢰도 문제들을 받는다. 예를 들어, 환경에서 본래 발생하거나 반도체 패키징 재료로부터 방출되는 알파 입자들과 같은, 백그라운드 방사선은 비트 셀(bit cell)에 부딪칠 수 있어, 값이 손상되게 한다. 메모리의 반복된 사용은 다른 장애들을 초래할 수도 있다. 예를 들어, 임의의 중요한 디바이스들 내의 일렉트로마이그레이션(electromigration)은 디바이스들이 마모되는 것을 초래할 수 있다: 그것들은 효과적으로 더 두껍게 되며, 그것에 의해 그의 저항을 증가시키고 결국 부정확한 값들이 판독되게 하는 타이밍 오류들을 초래한다. 다른 타입들의 고장들이 또한 가능하다. 메모리 칩이 고장나면, 결합 메모리 칩을 교체하는 실제적인 방법이 없다. 그 대신에, 사용자는 고가의 옵션인, 여전히 작동하는 메모리 및 프로세서 칩들의 모두를 포함하는, 전체 패키지를 교체해야 한다.

[0009] 도 2는 일부 실시예들에 따른 물리 메모리를 구현하는 제 2 멀티 칩 모듈(200)의 사시도를 예시한다. 멀티 칩 모듈(200)은 일반적으로 인터포저(interposer)(210), 멀티 코어 프로세서 칩(220), 및 메모리 칩 스택(240)을 포함한다. 인터포저(210)는 멀티 코어 칩(220)의 활성 측면에 연결된다. 메모리 칩 스택(240)은 서로의 상단에 적층되는 복수의 메모리 칩들을 포함한다. 도 2에 예시된 바와 같이, 메모리 칩 스택(240)은 메모리 칩(242), 메모리 칩(244), 메모리 칩(246), 및 메모리 칩(248)을 포함한다. 일반적으로, 메모리 칩 스택(240)은 도 2에 예시된 것보다 더 많거나 더 적은 메모리 칩들을 포함할 수 있다는 점을 주목한다. 메모리 칩 스택(240)의 각각의 개별 메모리 칩은 적절한 시스템 동작을 위해 요구되는 바와 같이, 메모리 칩 스택(240)의 다른 메모리 칩들에 연결된다. 메모리 칩 스택(240)의 각각의 개별 메모리 칩은 또한 적절한 시스템 동작을 위해 요구되는 바와 같이, 멀티 코어 칩(220)에 연결된다. 일부 실시예들에서, 메모리 칩 스택(240)은 단일 메모리 칩을 포함한다. 일부 실시예들에서, 멀티 칩 모듈(200)은 메모리 칩 스택(240)과 같은 하나보다 더 많은 메모리 칩 스택을 포함한다.

[0010] 동작 시, 멀티 칩 모듈(200)의 구성요소들은 단일 패키지(도 2에 도시되지 않음)에 결합되고, 따라서 메모리 칩 스택(240) 및 멀티 코어 칩(220)은 단일 집적 회로로서 사용자에게 나타난다. 멀티 코어 칩(220)에 메모리 칩 스택(240)의 전기적 연결은 수평 인터커넥트와 결합하여, 수직 인터커넥트, 예를 들어 비아 또는 실리콘 관통 구멍을 사용하여 달성된다. 인터포저(210)는 메모리 칩 스택(240)의 각각의 개별 메모리 칩에 멀티 코어 칩(220)을 연결하는 것을 용이하게 하기 위해 물리적 지원 및 인터페이스 양자를 제공한다. 5개의 개별 칩들과 비교될 때, 멀티 칩 모듈(200)은 일반적으로 구성요소 액세스 시간을 감소시키고 시스템 성능을 증가시킴과 동시

에, 시스템 비용 및 보드 공간을 절약한다. 멀티 칩 모듈(200)은 멀티 코어 프로세서(220)로부터 메모리 칩 스택(240)을 분리하므로 멀티 코어 프로세서(220)의 더 양호한 냉각을 허용한다. 그러나, 멀티 칩 모듈(200)은 또한 전체 패키지를 교체하는 것 없이 결함 메모리 칩이 용이하게 교체될 수 없으므로 신뢰도 및 서비스 가용성 문제들을 겪는다.

[0011] 도 3은 일부 실시예들에 따른 높은 신뢰도 메모리 컨트롤러를 갖는 집적 회로(300)를 블록도 형태로 예시한다. 집적 회로(300)는 일반적으로 단일 집적 회로 다이 상에 구현되는 멀티 코어 프로세서(310) 및 메모리(350)를 포함한다.

[0012] 멀티 코어 프로세서(310)는 메모리 액세스 생성 회로(320), 큐(queue)(332), 크로스바 스위치(crossbar switch)(XBAR)(334), 고속 입력/출력(I/O) 컨트롤러(336), 및 메모리 컨트롤러(340)를 포함한다. 메모리 액세스 생성 회로(320)는 "CPU0"로 라벨링된 중앙 처리 유닛(CPU) 코어(322), 및 "CPU1"로 라벨링된 CPU 코어(324)를 포함한다. CPU 코어들(322 및 324)은 메모리 액세스들을 수행하고, 메모리 액세스들을 정의하는 어드레스들, 데이터, 및 제어 신호들을 송신하고 수신한다. 큐(332)는 CPU 코어(322), CPU 코어(324), 및 XBAR(334)에 연결된다. XBAR(334)은 고속 I/O 컨트롤러(336) 및 메모리 컨트롤러(340)에 연결된다. 고속 I/O 컨트롤러(336)는 도 3에 도시되지 않은, 외부 신호들의 세트를 주변 디바이스에 송신하고 수신하기 위해 "I/O"로 라벨링된 입력/출력(I/O) 포트를 갖는다.

[0013] 메모리 컨트롤러(340)는 에리 정정 코드(ECC; error correction code)/순환 중복 코드(CRC; cyclic redundancy code) 산출("comp") 회로(342), 동적 랜덤 액세스 메모리(DRAM; dynamic random-access memory) 스케줄러(344), 및 물리 인터페이스(PHY)(346)를 포함한다. ECC/CRC comp 회로(342) 및 DRAM 스케줄러(344)는 PHY(346)에 각각 연결된다. PHY(346)는 "컨트롤"로 라벨링된 신호들의 세트를 제공하는 출력, "BA"로 라벨링된 뱅크 어드레스 신호들의 세트를 제공하는 출력, "어드레스"로 라벨링된 신호들의 세트를 제공하는 출력, 및 "데이터"로 라벨링된 신호들의 세트를 송신하고 수신하는 I/O 포트를 갖는다.

[0014] 메모리(350)는 어드레스 공간을 정의하고 DRAM(352), DRAM(354), DRAM(356), 및 DRAM(358)을 포함하는, 다수의 동적 랜덤 액세스 메모리(DRAM) 칩들을 포함한다. 메모리(350)는 도 1의 메모리 칩 스택(140) 또는 도 2의 메모리 칩 스택(240)에 의해 구현될 수 있다. DRAM들(352, 354, 356, 및 358)은 JEDEC에 의해 발행된 DDR3 더블 데이터 레이트(DDR; double data rate) 표준과 호환가능하지만, 다른 실시예들에서 그들은 다른 DDR 및 비-DDR 표준들과 호환가능할 수 있다. 일반적으로, DDR 칩들은 메모리 뱅크들의 세트를 각각 갖는다. 메모리(350) 내의 각각의 DRAM 칩은 컨트롤을 수신하는 입력, BA를 수신하는 입력, 어드레스를 수신하는 입력, 및 데이터를 송신하고 수신하는 I/O 포트를 갖는다.

[0015] 동작 시, CPU 코어(322) 및 CPU 코어(324) 양자는 하나 이상의 프로그램들에 대응하는 명령들을 검색하고 실행하며 메모리 액세스 요청들을 큐(332)에 제공함으로써 명령들과 연관된 데이터에 액세스하는 능력을 갖는다. 큐(332)는 I/O 컨트롤러(336) 또는 메모리 컨트롤러(340)에 디스패치(dispatch)를 위한 액세스들을 저장한다. 큐(332)는 선입, 선출 기초로 데이터 액세스들을 우선순위화한다.

[0016] XBAR(334)은 메모리 액세스 생성 회로(320), 큐(332), 고속 I/O 컨트롤러(336), 및 메모리 컨트롤러(340)를 포함하는, 멀티 코어 프로세서(310)의 회로들 및 그 연관된 버스들을 스위칭하고 멀티플렉싱한다. 고속 I/O 컨트롤러(336)는 XBAR(334)과 이더넷 컨트롤러와 같은, 외부 회로들 사이에 연결을 제공한다.

[0017] 메모리 컨트롤러(340)는 메모리 액세스 요청들에 응답하여 메모리(350)의 어드레스 공간 내의 메모리 위치들에 액세스한다. 메모리 컨트롤러(340)는 정상 데이터 및 데이터에 관한 특수 신뢰도 정보 양자를 표준, 재고품 메모리 칩들에 저장함으로써 높은 신뢰도를 보장한다. 신뢰도 데이터 정보는 비트 오류들의 검출 및 가능한 정정을 허용한다. 신뢰도 데이터를 저비용 상품 메모리에 저장함으로써, 메모리 컨트롤러(340)는 멀티 코어 프로세서(310)가 저가의 멀티 칩 모듈들에서 적층 다이와 집적되는 것을 허용한다.

[0018] 아래에 더 상세히 설명되는 바와 같이, 메모리 컨트롤러(340)는 어드레스 공간의 제 1 부분 내의 데이터 요소들 및 어드레스 공간의 제 2 부분 내의 데이터 요소들에 대응하는 신뢰도 데이터에 액세스한다. 메모리 컨트롤러(340)는 그것이 메모리(350)에 저장하는 신뢰도 데이터를 생성한 후에, 저장된 신뢰도 데이터에 대해 검사하기 위해 신뢰도 데이터를 산출하는 ECC/CRC comp 회로(342)를 사용한다. ECC/CRC comp 회로(342)는 신뢰도 데이터를 사용하여, DRAM 스케줄러(344)에 의해 액세스되는 데이터를 검사하고, 적절하다면 데이터에서 오류들을 선택적으로 정정하고 정정된 데이터를 요청 CPU에 전송한다.

[0019] PHY(346)는 ECC/CRC comp 회로(342) 및 DRAM 스케줄러(344)에 대한 인터페이스를 멀티 뱅크 메모리(350)에 제

공한다. 데이터에 액세스하기 위해, PHY(346)는 표준 컨트롤 신호들, BA 신호들, 및 어드레스 신호들을 메모리 (350)에 제공한다. 일반적으로, 메모리 컨트롤러(340)는 어드레스 공간의 제 1 부분으로부터 데이터 요소 및 어드레스 공간의 제 2 부분으로부터 신뢰도 데이터를 판독하기 위해 PHY(346)를 제어하도록 판독 액세스 요청에 응답한다. ECC/CRC comp 회로(342)는 수신된 데이터에 기초하여 신뢰도를 생성하고 메모리 컨트롤러(340)는 데이터가 정확하게 판독되었는지를 판단하기 위해 생성된 신뢰도 데이터를 검색된 신뢰도 데이터와 비교한다. 메모리 컨트롤러(340)는 데이터 요소에 대한 신뢰도 데이터를 생성하기 위해 ECC/CRC comp 회로(342)를 제어하도록 기록 액세스 요청에 응답하고 어드레스 공간의 제 1 부분에 데이터 요소 및 어드레스 공간의 제 2 부분에 신뢰도 데이터를 기록하기 위해 PHY(346)를 제어한다. 메모리 컨트롤러(340)가 상이한 레벨들의 신뢰도 지원을 위해 메모리(350)의 어드레스 공간을 생성하고 관리하는 방법들이 이제 설명될 것이다.

[0020] 도 4는 일부 실시예들에 따른 도 3의 메모리에 대한 어드레스 공간(400)의 표현을 예시한다. 어드레스 공간 (400)은 일반적으로 "뱅크 0"으로 라벨링된 메모리 뱅크(410), "뱅크 1"로 라벨링된 메모리 뱅크(420), "뱅크 2"로 라벨링된 메모리 뱅크(430), 및 "뱅크 3"으로 라벨링된 메모리 뱅크(440)를 포함하는 연속 메모리 뱅크들 중에 어드레스들의 인접 부분을 포함한다.

[0021] 메모리 뱅크(410)는 "A"로 라벨링된 대표적인 메모리 페이지(412) 및 "B" 내지 "P"로 연속적으로 라벨링된 다수의 추가 대표적인 메모리 페이지들을 포함하는, 다수의 4 킬로바이트(KB) 메모리 페이지들을 포함한다.

[0022] 메모리 뱅크들(420, 430, 및 440)은 마찬가지로 다수의 4 KB 메모리 페이지들을 포함한다. 그러나, 메모리 뱅크 (440)는 모든 메모리 뱅크들에 대한 신뢰도 데이터를 저장하는 인접 데이터 부분(442) 및 연속 신뢰도 부분 (444)을 포함한다. 신뢰도 부분(444)은 "E0"으로 라벨링된 대표적인 메모리 페이지(446), 및 "E1"로 라벨링된 대표적인 메모리 페이지를 포함한다. 메모리 페이지(446)는 메모리 페이지들(A 내지 H) 내의 데이터 요소들에 대응하는, "EA" 내지 "EH"로 연속적으로 라벨링된, 신뢰도 데이터를 포함한다.

[0023] 동작 시, 메모리 컨트롤러(340)는 어드레스 공간(400)의 제 1(예를 들어 상단 또는 더 낮은 어드레스) 부분 내 의 데이터 요소들에 액세스하고 어드레스 공간(400)의 제 2(예를 들어 하단 또는 더 높은 어드레스) 부분, 즉 신뢰도 부분(444) 내의 데이터 요소들에 대응하는 신뢰도 데이터에 액세스한다. 예를 들어 메모리 뱅크(410)는 4 KB 메모리 페이지들로 조직된다. 도 4는 어드레스들의 인접 부분에 대표적인 페이지들(A 내지 H)을 예시한다. 메모리 뱅크(440)에서, 메모리 컨트롤러(340)는 또한 어드레스들(444)의 인접 부분에서, 페이지들(A 내지 H) 내 의 데이터 그룹들에 대응하는, 신뢰도 데이터(EA 내지 EH)를 포함하는, 신뢰도 데이터 그룹(E0)에 대한 하나의 4 KB 메모리 페이지(446)에 액세스한다.

[0024] 마찬가지로, 메모리 뱅크(410)에서, 메모리 컨트롤러(340)는 어드레스들의 인접 부분 내의 페이지들(I 내지 P) 에서 데이터 그룹들에 대한 8개의 4 KB 메모리 페이지들에 액세스한다. 메모리 뱅크(440)에서, 메모리 컨트롤러 (340)는 또한 신뢰도 부분(444)에서, 페이지들(I 내지 P) 내의 데이터 그룹들에 대응하는, 신뢰도 데이터(도 4 에 구체적으로 도시되지 않음)를 포함하는, 신뢰도 데이터 그룹(E1)에 대한 하나의 4 KB 메모리 페이지에 액세스 한다.

[0025] 어드레스 공간(400)은 인접 부분 내의 신뢰도 데이터를 어드레스 공간(400)의 단부에 배치함으로써 선형 데이터 어드레스 공간을 제공하며, 그것에 의해 어드레스 공간에서 "홀들"을 방지한다. 어드레스 공간(400)은 다양한 타입들의 신뢰도 데이터를 지원한다. 예를 들어, 임의의 표준들은 모든 64 데이터 비트들(72 전체 비트들)에 대해 8 신뢰도 비트들을 갖는 것으로, (72, 64) SECDED 코드와 같은, 유용한 단일 오류 정정, 더블 오류 검출 (SECDED) 코드를 정의한다. SECDED를 사용하면, ECC/CRC comp 회로(342)는 단일 오류를 검출하고 정정하며, 더블 오류를 검출하지만 정정하지 않는 능력을 갖는다. 다른 공지된 코드들에 대해, ECC/CRC comp 회로(342)는 2 개보다 더 많은 오류들을 검출하고/하거나 정정하는 능력을 갖는다. 어드레스 공간(400)은 신뢰도 부분(444)의 크기가 사용된 신뢰도 코드의 타입에 기초하여 변화되는 것을 허용하며, 그 자체는 시스템의 신뢰도 요구들에 기초할 수 있다.

[0026] 그러나, 모든 신뢰도 데이터를 단일 메모리 뱅크에 배치함으로써, 메모리 컨트롤러(340)는 일부 시스템들에 대한 액세스 레이턴시를 허용불가능하게 증가시킬 수 있다. 예를 들어, 페이지들을 다수의 뱅크들에서 동시에 개방하게 하는 시스템들에서, 신뢰도 부분(444)으로의 액세스들은 메모리 컨트롤러(340)가 단일 뱅크(440)로부터 상이한 뱅크들로 액세스들을 위한 신뢰도 데이터에 액세스할 때 "병목"을 야기한다. 멀티 코어 프로세서(310)는 이러한 병목을 보상하기 위해 다른 메커니즘들을 통합할 수 있다는 점을 주목한다. 예를 들어, 메모리 컨트롤러 (340) 또는 메모리 액세스 생성 회로(320)와 같은 회로들은 신뢰도 데이터를 프리페치(prefetch)하고 그것을 로컬 캐시에 저장할 수 있다. 또한, 아래에 더 완전히 설명되는 바와 같이, 메모리 컨트롤러(340)는 신뢰도 데이

터를 하나보다 더 많은 단일 뱅크에 분배함으로써 신뢰도 데이터에 액세스하기 위해 레이턴시를 보상하거나, 데이터 요소들 및 신뢰도 데이터를 메모리 뱅크들 중에 대체 형태로 저장할 수 있다.

- [0027] 도 5는 일부 실시예들에 따른 도 3의 메모리에 대한 어드레스 공간(500)의 다른 표현을 예시한다. 어드레스 공간(500)은 일반적으로 "뱅크 0"으로 라벨링된 메모리 뱅크(510), "뱅크 1"으로 라벨링된 메모리 뱅크(520), "뱅크 2"으로 라벨링된 메모리 뱅크(530), 및 "뱅크 3"으로 라벨링된 메모리 뱅크(540)를 포함하는, 연속 메모리 뱅크들 중에 어드레스들의 인접 부분을 포함한다.
- [0028] 메모리 뱅크(510)는 "A" 내지 "D"로 연속적으로 라벨링된 4개의 대표적인 메모리 페이지들(516)을 포함하는, 인접 데이터 부분(512) 내의 다수의 메모리 페이지들을 포함한다. 뱅크(510)는 또한 신뢰도 부분(514) 내의 메모리 페이지들을 포함한다. 신뢰도 부분(514) 내의 각각의 페이지는 "EA"로 라벨링된 대표적인 신뢰도 데이터(518), 및 메모리 페이지들(A 내지 D) 내의 데이터 요소들에 대응하는, "EB" 내지 "ED"로 연속적으로 라벨링된 신뢰도 데이터를 포함하는, 신뢰도 데이터를 포함한다. 마찬가지로, 메모리 뱅크들(520, 530, 및 540)은 또한 데이터 부분들(522, 532, 및 542) 및 신뢰도 부분들(524, 534, 및 544)을 각각 포함한다. 신뢰도 부분(524, 534, 및 544) 내의 각각의 페이지는 데이터 부분들(522, 532, 및 542) 내의 데이터 요소들에 대응하는 신뢰도 데이터를 각각 포함한다.
- [0029] 동작 시, 메모리 컨트롤러(340)는 각각의 메모리 뱅크의 제 1 부분 내의 데이터 요소들 및 동일한 메모리 뱅크의 제 2 부분 내의 데이터 요소들에 대응하는 신뢰도 데이터에 액세스한다. 예를 들어, 메모리 컨트롤러(340)는 데이터 부분(510)의 메모리 페이지들(516) 내의 데이터 요소들(A, B, C, 및 D), 및 신뢰도 부분(514)의 메모리 페이지(518) 내의 신뢰도 데이터(EA 내지 ED)에 액세스한다. 따라서, 메모리 컨트롤러(340)는 데이터 및 그것의 대응하는 신뢰도 데이터 양자를 단일 메모리 뱅크에 저장한다. 마찬가지로, 메모리 컨트롤러(340)는 데이터 부분들(522, 532, 및 542) 내의 데이터 요소들 각각에 대해, 메모리 뱅크들(520, 530, 및 540) 각각에 액세스한다. 메모리 컨트롤러(340)는 또한 신뢰도 부분들(524, 534, 및 544) 내의 신뢰도 데이터 각각에 대해, 메모리 뱅크들(520, 530, 및 540) 각각에 액세스한다.
- [0030] 전체, 어드레스 공간(500)은 메모리 뱅크들(510-540) 중에 분배되는 비인접 데이터 부분, 및 또한 메모리 뱅크들(510-540) 중에 분배되는 비연속 신뢰도 부분을 갖는다. 메모리 컨트롤러(340)는 제 1 (100-X)%의 메모리 뱅크로부터 데이터 요소들에 액세스하고 마지막 X%의 동일한 메모리 뱅크로부터 신뢰도 데이터에 액세스한다. 예를 들어, 메모리 컨트롤러(340)가 (64, 72) SECDED 코드를 사용할 때, $X = 12.5\%$ 이다. 어드레스 공간(500)이 신뢰도 데이터를 대응하는 데이터로서 동일한 메모리 뱅크에 배치함으로써, 단일, 선형 데이터 공간을 포함하지 않지만, 메모리 공간(500)은 도 4의 메모리 공간(400)과 연관된 병목들을 방지한다.
- [0031] 도 6은 일부 실시예들에 따른 도 3의 메모리에 대한 어드레스 공간(600)의 다른 표현을 예시한다. 어드레스 공간(600)은 일반적으로 "뱅크 0"으로 라벨링된 메모리 뱅크(610), "뱅크 1"으로 라벨링된 메모리 뱅크(620), "뱅크 2"으로 라벨링된 메모리 뱅크(630), 및 "뱅크 3"으로 라벨링된 메모리 뱅크(640)를 포함하는, 연속 메모리 뱅크들 중에 어드레스들의 인접 부분을 포함한다.
- [0032] 어드레스 공간(600)은 "A" 내지 "R"로 라벨링된 대표적인 메모리 로우들(rows)을 포함하는, 데이터를 저장하고 4개의 메모리 뱅크들 중에 분배되는 다수의 메모리 로우들을 포함한다. 어드레스 공간(600)은 또한 데이터에 대한 신뢰도 코드들을 저장하고 또한 메모리 뱅크들 중에 분배되며, 데이터 요소들을 갖는 로우들과 인터리빙(interleaving)되는 다수의 메모리 로우들을 포함한다. 이러한 로우들 각각은 다른 로우들의 데이터 요소들에 대응하는 신뢰도 데이터를 갖는다.
- [0033] 특히, 메모리 뱅크(610)는 로우들(611-615)을 포함하고; 메모리 뱅크(620)는 로우들(621-625)을 포함하고; 메모리 뱅크(630)는 로우들(631-635)을 포함하고; 메모리 뱅크(640)는 로우들(641-645)을 포함한다. 어드레스 공간(600)에서, 데이터는 메모리 뱅크들 중에 분배된다. 따라서, 데이터 요소(H)가 뱅크(640)의 로우(642)에 저장될 때까지, 데이터 요소(A)는 뱅크(610)의 로우(611)에 저장되고, 데이터 요소(B)는 뱅크(620)의 로우(621)에 저장되는 등등이다. 그러나, 8개의 데이터 요소들이 이러한 방식으로 로우들에 분배된 후에, 로우들에 대응하는 신뢰도 데이터의 세트가 저장된다. 따라서, 메모리 뱅크(610)는 로우들(A 내지 H) 내의 데이터 요소들에 대응하는 로우(613)에 "EA - EH"로 라벨링된 신뢰도 데이터를 포함한다.
- [0034] 뱅크들(610-640)은 뱅크(610) 내의 로우(615)가 데이터 요소(P)를 저장할 때까지, 데이터 요소(I)를 저장하는 뱅크(620) 내의 로우(623), 데이터 요소(J)를 저장하는 뱅크(630) 내의 로우(633) 등등으로 시작하는 연속 위치들에 8개의 후속 데이터 요소들(I 내지 P)을 저장한다. 메모리 뱅크(620)의 로우(625)는 로우들(I 내지 P)에 테

이터 요소들에 대응하는 "EI - EP"로 라벨링된 신뢰도 데이터 등을 저장하는 등등이다.

- [0035] 동작 시, 메모리 컨트롤러(340)는 데이터 요소들을 어드레스 공간(600) 내의 연속 메모리 블록들(610 내지 640) 중에서 데이터 요소들에 대응하는 신뢰도 데이터와 인터리빙시킨다. 메모리 컨트롤러(340)는 다수의 블록들 중 연속 블록들 중에 임의의 수의 연속적으로 어드레스된 데이터 요소들을 갖는, 데이터 그룹의 각각의 데이터 요소를 저장하고, 다음 연속 블록에 그룹의 모든 데이터 요소들에 대한 신뢰도 데이터를 저장한다. 예를 들어, 메모리 컨트롤러(340)는 메모리 블록들(610 내지 640) 중에, 로우들(611, 621, 631, 641, 612, 622, 632, 및 642)에서 첫번째 8개의 데이터 그룹들에 수평으로 액세스한다. 메모리 컨트롤러(340)는 여덟번째 데이터 그룹 후에 로우(613)에 위치된 신뢰도 데이터(EA - EH)에 액세스하며, 이는 로우들(A-H)에 첫번째 8개의 데이터 그룹들에 대응하는 신뢰도 데이터를 저장한다. 메모리 컨트롤러(340)는 또한 메모리 블록들(610 내지 640) 중에, 로우들(623, 633, 643, 614, 624, 634, 644, 및 615)에서 두번째 8개의 데이터 그룹들에 수평으로 액세스한다. 메모리 컨트롤러(340)는 또한 두번째 8개의 데이터 그룹들 후에 로우(625) 내의 신뢰도 데이터(EI - EP)에 액세스하며, 이는 로우들(I-P)에 두번째 8개의 데이터 그룹들에 대응하는 신뢰도 데이터를 저장하는 등등이다.
- [0036] 데이터 요소들을 연속 메모리 블록들 중에서 데이터 요소들에 대응하는 신뢰도 데이터와 인터리빙시킴으로써, 메모리 컨트롤러(340)는 특정 메모리 액세스를 위한 신뢰도 데이터가 데이터를 저장하는 동일한 메모리 블록에 저장될 기회를 감소시킨다. DDR DRAM에서, 새로운 페이지에 액세스하기 전에 이전 페이지는 프리차지 커맨드(precharge command)를 블록에 발령함으로써 폐쇄되고, 새로운 페이지는 활성화 커맨드를 발령함으로써 개방되어야 한다. 따라서, 데이터 및 그것의 대응하는 신뢰도 데이터가 동일한 블록에 저장될 확률을 감소시킴으로써, 어드레스 공간(600)은 데이터 및 대응하는 신뢰도 데이터에 액세스하는데 요구되는 시간의 평균량을 감소시킨다.
- [0037] 도 7은 일부 실시예들에 따른 도 3의 메모리에 대한 어드레스 공간(700)의 다른 표현을 예시한다. 어드레스 공간(700)은 일반적으로 "채널 0"으로 라벨링된 메모리 채널(710), "채널 1"로 라벨링된 메모리 채널(720), 데이터 요소(730), 및 신뢰도 데이터(740)를 포함한다. 메모리 채널(710)은 "뱅크 0" 내지 "뱅크 7"로 연속적으로 라벨링된, 메모리 블록들(711 내지 718)을 포함하는, 다수의 메모리 블록들을 포함한다.
- [0038] 메모리 블록(711)은 데이터 바이트들("A [7]" 내지 "A [0]")을 각각 포함하는 "A"로 라벨링된 데이터 그룹을 포함한다. 메모리 블록들(712-718)은 마찬가지로 8 바이트들을 각각 갖고 메모리 블록(711)과 유사한 방식으로 배열되는 데이터 그룹들을 포함한다. 메모리 블록(712)는 데이터 바이트들("B [7]" 내지 "B [0]")을 각각 포함하는 "B"로 라벨링된 데이터 그룹을 포함한다. 메모리 블록(713)은 데이터 바이트들("C [7]" 내지 "C [0]")을 각각 포함하는 "C"로 라벨링된 데이터 그룹을 포함한다. 메모리 블록(714)는 데이터 바이트들("D [7]" 내지 "D [0]")을 각각 포함하는 "D"로 라벨링된 데이터 그룹을 포함한다. 메모리 블록(715)는 데이터 바이트들("E [7]" 내지 "E [0]")을 각각 포함하는 "E"로 라벨링된 데이터 그룹을 포함한다. 메모리 블록(716)는 데이터 바이트들("F [7]" 내지 "F [0]")을 각각 포함하는 "F"로 라벨링된 데이터 그룹을 포함한다. 메모리 블록(717)는 데이터 바이트들("G [7]" 내지 "G [0]")을 각각 포함하는 "G"로 라벨링된 데이터 그룹을 포함한다. 메모리 블록(718)은 데이터 바이트들("H [7]" 내지 "H [0]")을 각각 포함하는 "H"로 라벨링된 데이터 그룹을 포함한다.
- [0039] 메모리 채널(720)은 "뱅크 0"으로 라벨링된 대표적인 메모리 블록(721) 및 "뱅크 1"로 라벨링된 대표적인 메모리 블록을 포함하는, 다수의 추가 메모리 블록들을 포함한다. 메모리 블록(721)은 "ECC 구성요소들"로 라벨링된 신뢰도 데이터 구성요소들을 포함한다.
- [0040] 데이터 요소(730)는 8개의 대표적인 데이터 바이트들, 구성요소([0]) 내지 구성요소([7])를 포함한다.
- [0041] 동작 시, 메모리 컨트롤러(340)는 데이터 요소(730)의 일부들을 메모리 채널(710) 중에 인터리빙하고, 데이터 요소(730)에 대한 신뢰도 데이터를 추가 메모리 블록(721)에 저장한다. 예를 들어, 메모리 컨트롤러(340)는 데이터 요소(730)의 구성요소([0])를 블록(711)의 A [0]에, 데이터 요소(730)의 구성요소([1])를 블록(712)의 B [0]에, 데이터 요소(730)의 구성요소([2])를 블록(713)의 C [0]에 등등, 내지 데이터 요소(730)의 구성요소([7])를 블록(718)의 H [0]에 저장한다. 메모리 컨트롤러(340)는 데이터 구성요소([7]) 내지 데이터 구성요소([0])에 대응하는, 신뢰도 데이터 구성요소(740)를 추가 메모리 블록(721)의 바이트 위치([0])에 더 저장한다.
- [0042] 데이터 요소의 바이트들을 메모리 채널의 블록들 중에서 인터리빙하고, 신뢰도 데이터 바이트들을 추가 메모리 채널의 추가 블록에 저장함으로써, 메모리 컨트롤러(340)는 단일 블록이 고장날 때 데이터의 복구를 허용한다.
- [0043] 그러나, 메모리 컨트롤러(340)는 결함 메모리 블록의 데이터 구성요소들을, 다른 완전한 기능 메모리 블록들에서 재생성하는 능력을 갖는다. 예를 들어, 메모리 컨트롤러(340)가 데이터 요소의 각각의 구성요소를 메모리 블록

크에 저장하고, 추가 메모리 뱅크로부터 신뢰도 데이터 구성요소를 갖는 각각의 구성요소를 커버하므로, 결합 메모리 요소의 모든 데이터 요소 구성요소는 연관된 신뢰도 데이터에 의해 추가 뱅크로부터 커버된다. 예를 들어 SECDED 코드를 사용하면, 메모리 컨트롤러(340)는 모든 데이터 요소들을 결합 뱅크에서 검출, 정정, 및 재생성하는 능력을 갖는다.

[0044] 위의 도 4-도 7에 개시된 것들과 같은 구성들을 사용함으로써, 멀티 코어 프로세서(310)는 저가의 재고품 메모리를 사용하여 메모리 칩들을 추가하는 것 없이 시스템의 신뢰도, 유용성, 및 서비스 가용성을 증대시킨다.

[0045] 도 8은 일부 실시예들에 따른 데이터를 기록하는 방법(800)의 흐름도를 예시한다. 동작 박스(810)에서, 데이터 요소에 대한 기록 액세스는 리퀘스터로부터 수신된다. 동작 박스(820)에서, 데이터 요소에 대한 신뢰도 데이터가 산출된다. 동작 박스(830)에서, 데이터 요소는 어드레스 공간의 제 1 부분에 저장된다. 동작 박스(840)에서, 신뢰도 데이터는 상기 어드레스 공간의 제 2 부분에 저장된다.

[0046] 도 9는 일부 실시예들에 따른 데이터를 판독하는 방법(900)의 흐름도를 예시한다. 예를 들어, 판독은 도 8의 방법(800)을 사용하여 이전에 기록된 데이터에 대해 수행될 수 있다. 동작 박스(910)에서, 데이터 요소에 대한 판독 액세스는 리퀘스터로부터 수신된다. 결정 박스(920)에서, 데이터 요소는 어드레스 공간의 제 1 부분으로부터 판독된다. 동작 박스들(930)의 한 세트에서, 신뢰도 데이터가 정확하게 판독되었는지가 결정된다.

[0047] 동작 박스들(930)의 세트는 산출된 신뢰도 데이터를 형성하기 위해 어드레스 공간의 제 1 부분으로부터 판독된 데이터 요소에 대한 신뢰도 데이터가 산출되는 동작 박스(932), 저장된 신뢰도 데이터를 형성하기 위해 어드레스 공간의 제 2 부분에 저장된 신뢰도 데이터가 판독되는 동작 박스(934), 및 산출된 신뢰도 데이터가 저장된 신뢰도 데이터와 비교되는 동작 박스(936)를 더 포함한다.

[0048] 방법(900)을 계속하면, 판단 박스(940)는 저장된 신뢰도 데이터가 산출된 신뢰도 데이터와 일치하는지를 판단한다. 저장된 신뢰도 데이터가 산출된 신뢰도 데이터와 일치하면, 흐름은 동작 박스(942)로 진행하며, 이는 데이터 요소를 리퀘스터로 복귀시킨다. 저장된 신뢰도가 산출된 신뢰도 데이터와 일치하지 않으면, 이때 흐름은 신뢰도 데이터가 정정될 수 있는지를 판단하는 판단 박스(944)로 진행한다. 신뢰도 데이터가 정정될 수 있으면, 이때 흐름은 데이터를 정정하는 동작 박스(946), 및 정정된 데이터를 리퀘스터로 복귀시키는 동작 박스(948)로 진행한다. 신뢰도 데이터가 정정될 수 없으면, 흐름은 오류를 리퀘스터에 보고하는 동작 박스(950)로 진행한다.

[0049] 데이터 및 대응하는 신뢰도 데이터를 저장하고 나중에 검색하는 것은 위의 도 4-도 7에 설명된 기술들 중 어느 것을 사용하여 수행될 수 있다. 따라서, 일부 실시예들에서, 어드레스 공간은 어드레스들의 제 1 인접 부분 및 어드레스들의 제 2 인접 부분으로 분할되고, 데이터 요소는 어드레스들의 제 1 인접 부분에 저장되고, 신뢰도 데이터는 어드레스들의 제 2 인접 부분에 저장된다. 일부 실시예들에서, 어드레스 공간은 뱅크의 어드레스들의 제 1 인접 부분 및 뱅크의 어드레스들의 제 2 인접 부분으로 분할되고, 데이터 요소는 뱅크의 어드레스들의 제 1 인접 부분에 저장되고 신뢰도 데이터는 뱅크의 어드레스들의 제 2 인접 부분에 저장된다. 일부 실시예들에서, 어드레스 공간은 어드레스 공간 내에서 순서를 갖는 복수의 뱅크들에 분할되고, 어드레스 공간의 제 1 부분은 그 순서로 복수의 뱅크들 중에 분배되는 미리 결정된 수의 데이터 요소들의 복수의 그룹들로 형성되고, 어드레스 공간의 제 2 부분은 복수의 그룹들 각각의 각각 대응하는 데이터 요소에 대한 신뢰도 데이터 요소를 포함하며, 여기서 그룹에 대한 신뢰도 데이터 요소들은 그 순서로 그룹의 마지막 데이터 요소를 포함하는 제 2 뱅크 후에 제 1 뱅크에 위치된다. 일부 실시예들에서, 어드레스 공간은 제 1 채널 및 제 2 채널을 사용하여 형성되며, 제 1 채널은 복수의 뱅크들을 포함하고, 데이터 요소는 제 1 채널 내의 복수의 뱅크들 중에 분배되고, 데이터 요소에 대한 신뢰도 데이터는 제 2 채널에 저장된다.

[0050] 도 3의 메모리 컨트롤러(340)는 하드웨어 및 소프트웨어의 다양한 조합들로 구현될 수 있고, 소프트웨어 구성요소는 적어도 하나의 프로세서에 의한 실행을 위한 컴퓨터 판독가능 저장 매체에 저장될 수 있다. 더욱이, 도 4-도 7에 예시된 어드레스 맵들은 컴퓨터 판독가능 저장 매체에 저장되고 메모리 컨트롤러(340)의 기능을 구현하는 적어도 하나의 프로세서에 의해 실행되는 명령들에 의해 적어도 부분적으로 구현될 수도 있다. 도 8 및 도 9에 도시된 동작들 각각은 비일시적 컴퓨터 메모리 또는 컴퓨터 판독가능 저장 매체에 저장된 명령들에 대응할 수 있다. 다양한 실시예들에서, 비일시적 컴퓨터 판독가능 저장 매체는 자기 또는 광 디스크 저장 디바이스, 플래시 메모리와 같은 고체 상태 저장 디바이스들, 또는 다른 비휘발성 메모리 디바이스 또는 디바이스들을 포함한다. 비일시적 컴퓨터 판독가능 저장 매체 상에 저장된 컴퓨터 판독가능 명령들은 하나 이상의 프로세서들에 의해 해석되고/되거나 실행가능한 소스 코드, 어셈블리 언어 코드, 목적 코드, 또는 다른 명령 포맷일 수 있다.

[0051] 더욱이, 메모리 컨트롤러(340) 및/또는 멀티 코어 프로세서(310)는 집적 회로(300)를 제조하기 위해, 프로그램

에 의해 판독되고 직접 또는 간접적으로 사용될 수 있는 데이터베이스 또는 다른 데이터 구조의 형태로 컴퓨터 액세스가능 데이터 구조에 의해 설명되거나 표현될 수 있다. 예를 들어, 이러한 데이터 구조는 베릴로그(Verilog) 또는 VHDL과 같은 고급 설계 언어(HDL; high level design language)에서 하드웨어 기능성의 동작 레벨 기술 또는 레지스터 전송 레벨(RTL; register-transfer level) 설명일 수 있다. 설명은 합성 라이브러리로부터 게이트들의 리스트를 포함하는 넷리스트(netlist)를 생성하는 설명을 합성할 수 있는 합성 툴에 의해 판독될 수 있다. 넷리스트는 또한 집적 회로(300)를 포함하는 하드웨어의 기능성을 표현하는 게이트들의 세트를 포함한다. 그 다음, 넷리스트는 마스크들에 적용될 기하학적 형상들을 설명하는 데이터 세트를 생성하기 위해 배치되고 라우팅될 수 있다. 그 다음, 마스크들은 집적 회로(300)를 제조하기 위해 다양한 반도체 제조 단계들에 사용될 수 있다. 대안적으로, 컴퓨터 액세스가능 저장 매체 상의 데이터베이스는 요구되는 바와 같이, 넷리스트(합성 라이브러리를 갖거나 갖지 않음) 또는 데이터 세트, 또는 그래픽 데이터 시스템(GDS; Graphic Data System) II 데이터일 수 있다.

[0052] 개시된 실시예들에 대한 다양한 수정들은 당해 기술에서 통상의 기술자들에게 분명할 것이다. 본 명세서에 설명된 메모리 컨트롤러는 멀티 칩 모듈들(100 및 200) 이외에 데이터 손상에 민감한 다른 직접 회로 구성들에 유용하다. 예를 들어, 프로세서 및 메모리 칩들은 플립 칩 본딩(flip-chip bonding)을 사용하여 머더보드(motherboard) 기판에 직접 부착된다. 메모리 컨트롤러 및 메모리는 동일한 다이 상에 구현되지만 다른 이유들로 고 레벨들의 전자기 간섭(EMI; electromagnetic interference)을 갖는 환경들에 사용됨으로써와 같이, 데이터 손상에 민감할 수도 있다. 메모리 칩 스택(140) 또는 메모리 칩 스택(240)은 예를 들어 개별 CPU 메모리, 개별 그래픽 처리 유닛(GPU; graphics processing unit) 메모리, 개별 APU 메모리 등으로서, 집적 회로(300) 메인 메모리로부터 분리되어 구현될 수 있다. 다이 적층 집적(100) 및 다이 적층 집적(200)은 멀티 칩 모듈(MCM; multi-chip module)로 구현될 수 있다. 대체적으로, 메모리 칩들은 공통 기판 상의 CPU, GPU, APU, 메인 메모리 등에 인접하여 배치되고 이들과 동일 평면일 수 있다. 멀티 칩 모듈들(100 및 200)이 4-칩 메모리 칩 스택들을 포함하지만, 다른 실시예들은 상이한 수들의 메모리 칩들을 포함할 수 있다는 점을 주목한다.

[0053] 메모리 컨트롤러(340)는 도 3에 도시된 바와 같은 마이크로프로세서 다이 상의 적어도 하나의 프로세서 코어와 집적될 수 있거나, 그 자체로 개별 칩일 수 있다. 일부 실시예들에서, 집적 회로(310)는 CPU를 필요로 하지 않는 로직 기능들과 같은, 컴퓨팅 기능들 이외에 다른 전체 기능들을 수행할 수 있다. 더욱이, 도 3은 CPU 코어들(322 및 324)로부터 분리된 메모리 컨트롤러(340)를 도시하지만, 그것은 CPU 코어 또는 다른 로직 블록 내부에 형성될 수도 있다.

[0054] 메모리 컨트롤러(340)의 동작은 상이한 레벨들의 신뢰도 및 오버헤드를 구현하는 다양한 어드레스 맵들에 대해 설명되었다. 도 4-도 7은 대표적인 수의 메모리 뱅크들로 이러한 개념들을 예시하지만 그 안에 설명된 기술들은 상이한 수들의 메모리 뱅크들로 스케일링될 수 있다. 예를 들어 메모리(350)가 4개의 DDR3 칩들로 구현되면, 이때 어드레스 공간 내의 메모리 뱅크들의 전체 수는 32일 것이다.

[0055] 사용될 수 있는 신뢰도 데이터의 예들은 패리티 비트들, 오류 정정 코드 비트들{예를 들어, 단일 오류 정정(SEC; single error correction), 단일 오류 정정 및 더블 오류 검출(SEC-DED; single error correction and double error detection), 더블 비트 오류 정정 및 트리플 비트 오류 검출(DEC-TED; double bit error correction and triple bit error detection), 트리플 오류 정정, 쿼드 오류 검출(TEC-QED; triple-error-correct, quad-error-detect) 및 선형 블록 코드들 예컨대 BCH(Bose Chaudhuri Hocquenghem) 코드들을 포함하지만 이들에 제한되지 않음} 및 체크섬들(예를 들어, CRC, 메시지 다이제스트(MD5))을 포함한다. 1개, 2개, 또는 그 이상의 레벨들의 ECC 보호를 위한 지원이 제공될 수 있으며, 시스템 하드웨어 또는 소프트웨어는 밸런스 성능 및 신뢰도를 선택할 수 있다.

[0056] 메모리(350)는 DRAM 기술의 맥락에서 상술되었다. 그러나, 메모리(350)는 다른 메모리 기술들, 예를 들어 정적 랜덤 액세스 메모리(SRAM; static random access memory), 상 변화 메모리(PCM; phase-change memory), 멤리스터들(memristors) 및 스핀 토크 전달 자기 RAM(STT-MRAM; spin-torque transfer magnetic RAM)과 같은 저항성 RAM 기술들, 및 플래시 메모리로 구현될 수 있다.

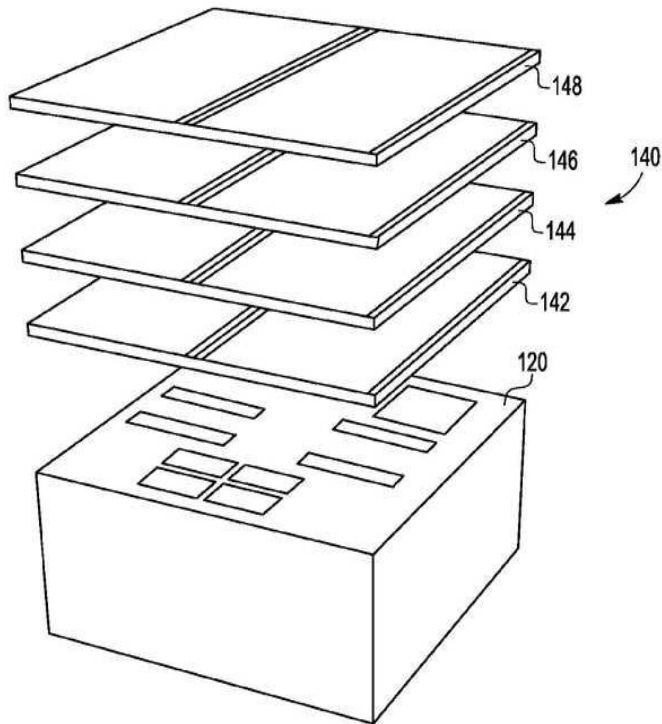
[0057] 위의 도 4-도 7에 예시된 실시예들은 8 바이트들의 데이터 당 1 바이트의 신뢰도 데이터를 사용한다. 다른 실시예들에 따르면, 주어진 수의 데이터 바이트들에 대한 신뢰도 데이터의 양은 상이할 수 있다.

[0058] 예시된 실시예들에서, 메모리 컨트롤러(340)는 임의의 메모리 뱅크들의 임의의 부분 내의 신뢰도 데이터에 액세스한다. 일부 실시예들에 따르면, 메모리 컨트롤러(340)는 대체 메모리 뱅크들의 대체 부분들에 액세스할 수 있다.

- [0059] 일부 예시된 실시예들은 다수의 뱅크들 중에서 데이터 요소들에 대응하는 신뢰도 데이터와 데이터 요소들의 인터리빙을 나타낸다. 일부 실시예들에 따르면, 인터리빙 및 매핑 알고리즘들이 수정될 수 있다.
- [0060] 일부 실시예들에서, 다수의 뱅크들에 걸친 어드레스들의 인접 부분이 도시된다. 다른 실시예들에 따르면, 어드레스들의 부분은 어드레스들의 비인접 부분일 수 있고 어드레스 홀들을 포함할 수 있다.
- [0061] 따라서, 개시된 실시예들의 범위 내에 있는 개시된 실시예들의 모든 수정들을 커버하는 것이 첨부된 특허청구범위에 의해 의도된다.

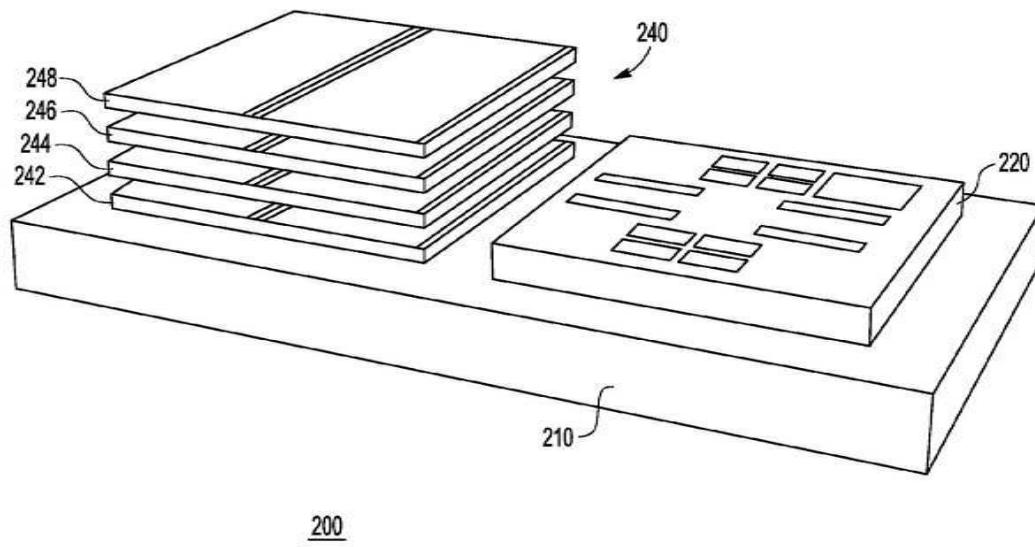
도면

도면1

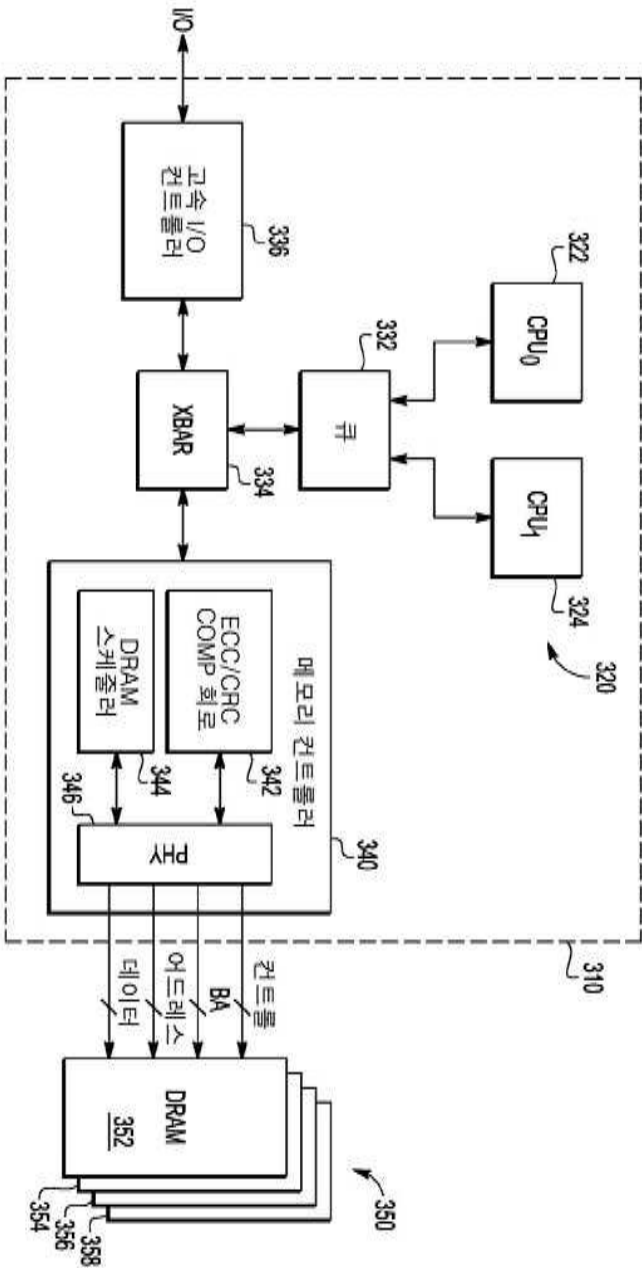


100

도면2

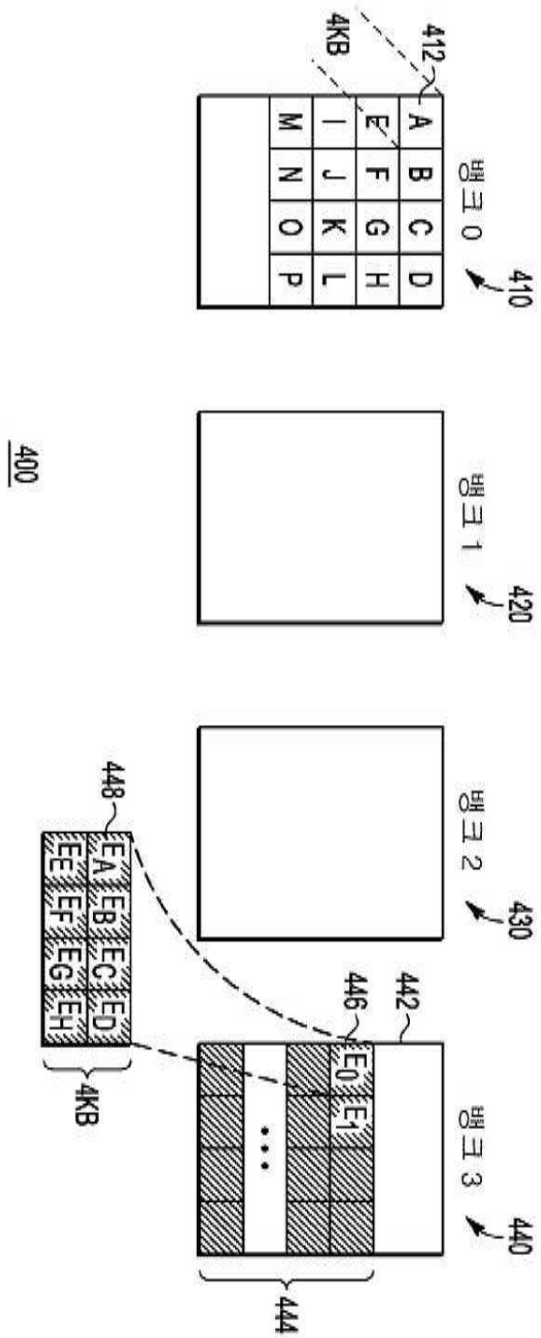


도면3

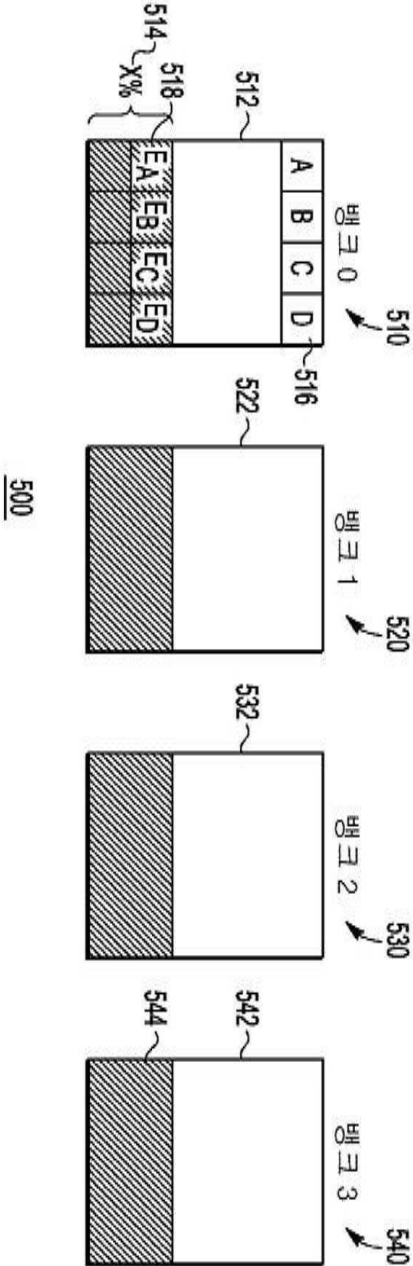


300

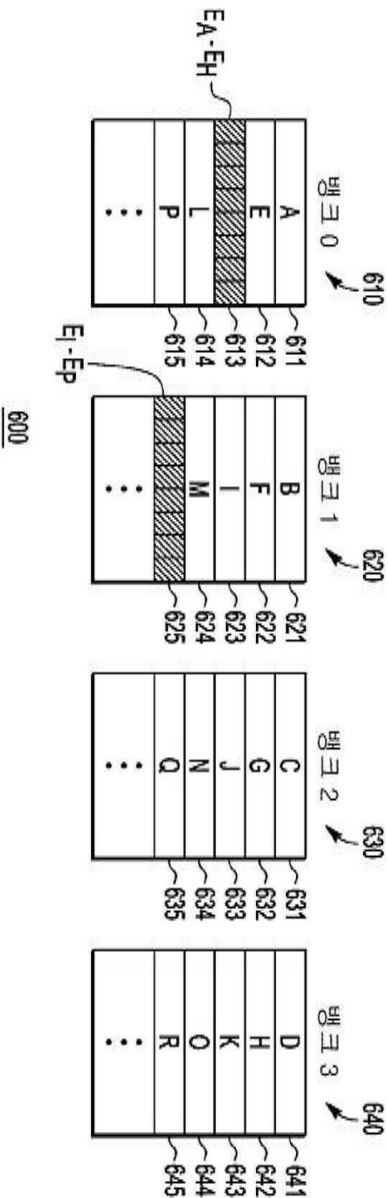
도면4



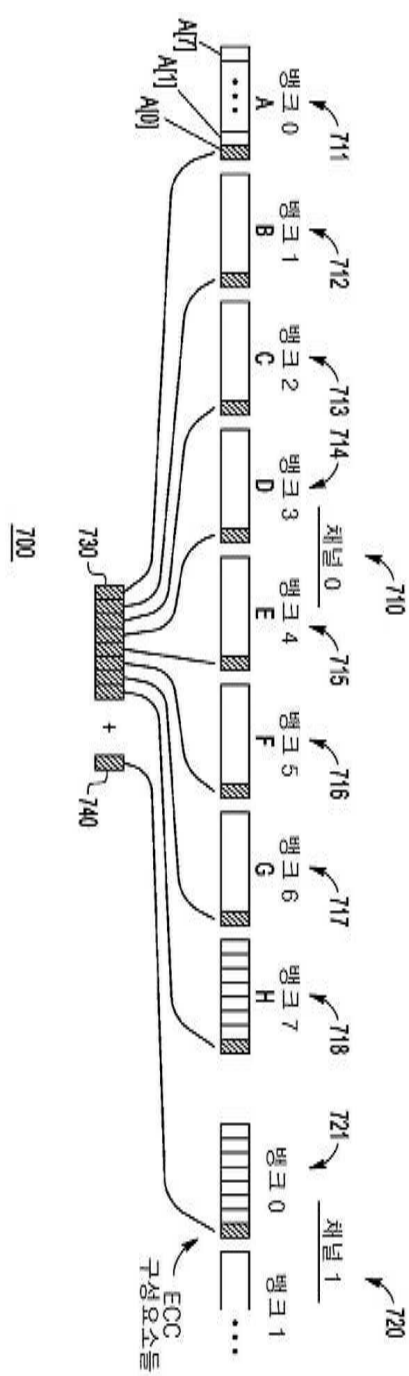
도면5



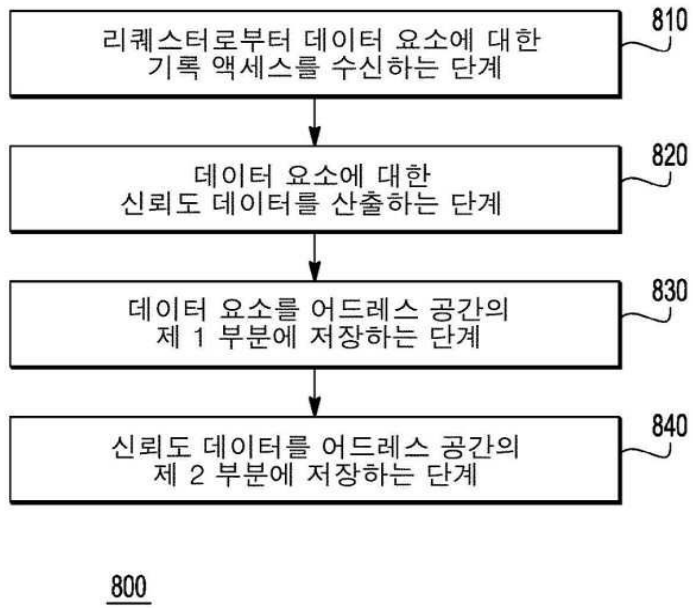
도면6



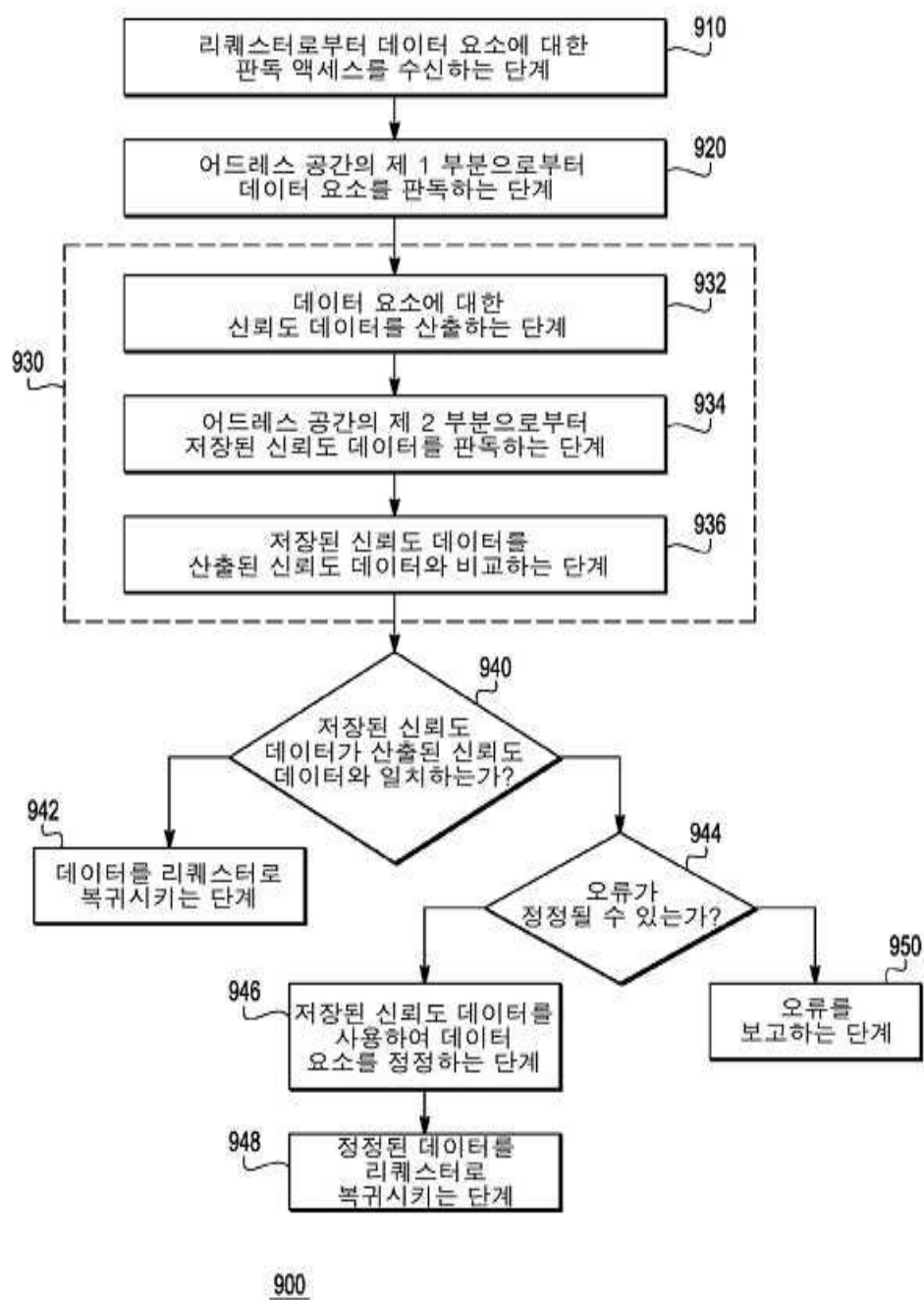
도면7



도면8



도면9



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 1

【변경전】

어드레스 공간(500)의

【변경후】

어드레스 공간(600)의

【직권보정 2】

【보정항목】 청구범위

【보정세부항목】 청구항 1

【변경전】

메모리 컨트롤러(350)는

【변경후】

메모리 컨트롤러(340)는