



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년10월26일
(11) 등록번호 10-2317657
(24) 등록일자 2021년10월20일

(51) 국제특허분류(Int. Cl.)
G06F 12/06 (2006.01) G06F 12/02 (2018.01)
G06F 13/16 (2006.01) G06F 3/06 (2006.01)
(52) CPC특허분류
G06F 12/0646 (2013.01)
G06F 12/0238 (2013.01)
(21) 출원번호 10-2016-0044912
(22) 출원일자 2016년04월12일
심사청구일자 2021년04월09일
(65) 공개번호 10-2017-0041616
(43) 공개일자 2017년04월17일
(30) 우선권주장
62/238,660 2015년10월07일 미국(US)
15/060,596 2016년03월03일 미국(US)
(56) 선행기술조사문헌
US20130019048 A1
(뒷면에 계속)

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
헨슨, 크레이그
미국 캘리포니아주 95134 산호세 퍼스트 스트리트
4271 노스 24호
스와브릭, 이안
미국 캘리포니아주 95051 산타클라라 포메로이 애
비뉴 692
(뒷면에 계속)
(74) 대리인
특허법인 고려

전체 청구항 수 : 총 9 항

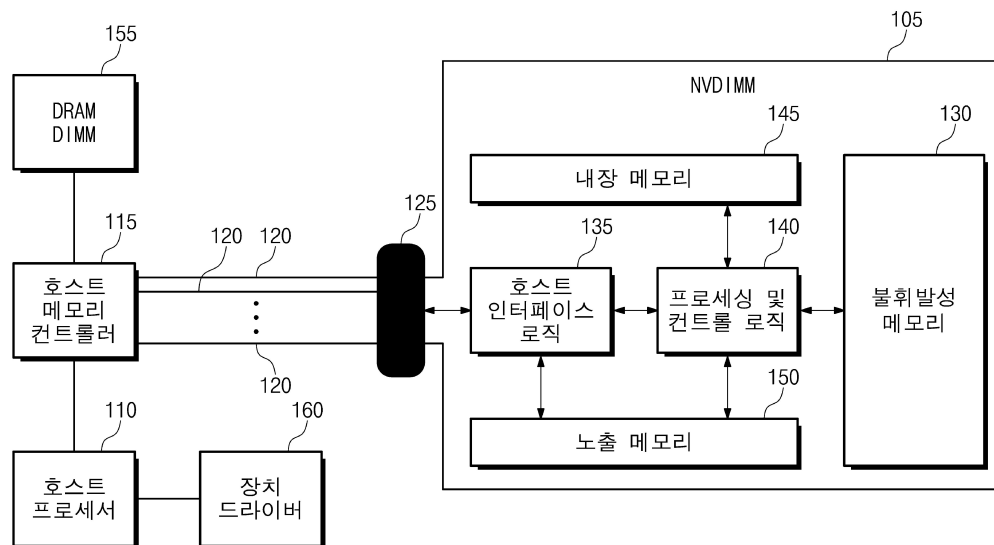
심사관 : 남윤권

(54) 발명의 명칭 NVDIMM을 포함하는 장치 및 그것의 액세스 방법

(57) 요약

본 발명의 실시 예에 따른 장치는 NVDIMM 및 장치 드라이버를 포함할 수 있다. NVDIMM(Non-Volatile Dual In-Line Memory Module)은 DIMM(Dual In-Line Memory Module) 소켓에 장착될 수 있다. 장치 드라이버는 호스트 프로세서 상에서 동작할 수 있다. NVDIMM은 메모리 및 노출 메모리를 포함하고, 노출 메모리는 노출 메모리의 제 1 사이즈 및 노출 메모리의 기본 주소를 포함할 수 있다. 장치 드라이버는 호스트 메모리 컨트롤러로 전송되는 메모리 주소를 인터셉트하고, 메모리 주소를 사전 매핑된 메모리 주소로 교체하도록 동작할 수 있다. 사전 매핑된 메모리 주소는 메모리 주소와 다를 수 있다.

대표도



- | | |
|---|--|
| <p>(52) CPC특허분류
 G06F 13/1668 (2013.01)
 G06F 3/0655 (2013.01)
 G06F 3/0683 (2013.01)</p> <p>(72) 발명자
 베커맨, 마이클
 미국 캘리포니아주 95032 로스가토스 케네디 로드
 306
 창, 치젠
 미국 캘리포니아주 94539 프리몬트 도 코트 150</p> | <p>(56) 선행기술조사문헌
 US06732249 B
 US20040215856 A1
 US20130227196 A1
 US20090154254 A1</p> |
|---|--|
-

명세서

청구범위

청구항 1

DIMM(Dual In-Line Memory Module) 소켓에 장착된 NVDIMM(Non-Volatile Dual In-Line Memory Module); 그리고 호스트 프로세서 상에서 동작하는 장치 드라이버를 포함하되,

상기 NVDIMM은 메모리 및 노출 메모리를 포함하고, 상기 노출 메모리는 상기 노출 메모리의 제 1 사이즈 및 상기 노출 메모리의 기본 주소를 포함하고,

상기 장치 드라이버는 호스트 메모리 컨트롤러로 향하는 메모리 주소를 인터셉트하고 그리고 상기 메모리 주소를 상기 호스트 메모리 컨트롤러로 전송되는 요청에서 중간 주소로 교체하도록 동작하고,

상기 중간 주소는 상기 메모리 주소와 다르고, 그리고 상기 메모리 주소에서 물리-중간(physical-to-intermediate) 매핑을 사용하여 생성되고,

상기 노출 메모리는 직접적으로 주소화 가능한(addressable) 메모리를 포함하고 그리고 상기 메모리 주소는 상기 노출 메모리에 있는 장치.

청구항 2

제 1 항에 있어,

상기 중간 주소는 상기 호스트 메모리 컨트롤러가 타겟 메모리 주소를 상기 NVDIMM에 상기 중간 주소에 응답하여 전송하도록 설계되고,

상기 타겟 메모리 주소는 상기 메모리 주소에 저장된 값의 물리적 위치를 포함하는 장치.

청구항 3

제 2 항에 있어,

상기 장치 드라이버는 상기 NVDIMM 내의 상기 노출 메모리의 상기 제 1 사이즈, 상기 NVDIMM 내의 상기 노출 메모리의 상기 기본 주소, 상기 NVDIMM 내의 상기 노출 메모리의 논리 세그먼트의 제 2 사이즈, 그리고 상기 호스트 메모리 컨트롤러의 메모리 컨트롤러 동작 모드에 응답하여 상기 메모리 주소로부터 상기 중간 주소를 생성하도록 동작하는 장치.

청구항 4

제 3 항에 있어,

상기 장치 드라이버는 상기 호스트 메모리 컨트롤러에 의한 상기 메모리 주소의 논리-물리(logical-to-physical) 매핑에 응답하여 상기 물리-중간 매핑을 사용하여 상기 메모리 주소로부터 상기 중간 주소를 생성하도록 더 동작하는 장치.

청구항 5

제 1 항에 있어,

상기 호스트 메모리 컨트롤러는 상기 중간 주소에 논리-물리 매핑을 구현하는 장치.

청구항 6

NVDIMM(Non-Volatile Dual In-Line Memory Module)의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계;

상기 메모리 주소를 물리-중간 변환(physical-to-intermediate transformation)을 사용하여 중간 주소로 사전 매핑(pre-mapping)하는 단계; 그리고

상기 중간 주소를 호스트 메모리 컨트롤러로 전송하는 단계를 포함하되,

상기 NVDIMM은 메모리 및 노출 메모리를 포함하고, 상기 노출 메모리는 상기 노출 메모리의 제 1 사이즈 및 상기 노출 메모리의 기본 주소를 포함하고, 상기 메모리 주소는 상기 노출 메모리 내에 있고,

상기 중간 주소는 상기 NVDIMM 내의 상기 노출 메모리의 상기 제 1 사이즈, 상기 NVDIMM 내의 상기 노출 메모리의 상기 기본 주소, 상기 NVDIMM 내의 상기 노출 메모리의 논리 세그먼트의 제 2 사이즈, 상기 호스트 메모리 컨트롤러의 메모리 컨트롤러 동작 모드, 그리고 상기 호스트 메모리 컨트롤러에 의해 사용되는 논리-물리 변환(logical-to-physical transformation)에 응답하는 상기 메모리 주소와 다르고,

상기 호스트 메모리 컨트롤러는 타겟 메모리 주소를 상기 NVDIMM에 전송하고,

상기 타겟 메모리 주소는 상기 중간 주소에 응답하고 그리고 상기 메모리 주소에 저장된 값의 물리적 위치를 나타내고, 그리고

상기 노출 메모리는 직접적으로 주소화 가능한(addressable) 메모리를 포함하는 방법.

청구항 7

제 6 항에 있어,

상기 호스트 메모리 컨트롤러는 상기 중간 주소에 논리-물리(logical-to-physical) 매핑을 구현하는 방법.

청구항 8

그것에 대한 명령어들을 저장하는 유형의 스토리지 매체인 비-일시적(Non-Transitory) 스토리지 매체를 포함하는 물품에 있어, 상기 명령어들은 기계에 의해 실행될 때:

NVDIMM(Non-Volatile Dual In-Line Memory Module)의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계;

물리-중간 변환(physical-to-intermediate transformation)을 사용하여 상기 메모리 주소를 중간 주소로 사전 매핑하는 단계; 그리고

상기 중간 주소를 호스트 메모리 컨트롤러로 전송하는 단계를 야기하되,

상기 NVDIMM은 메모리 및 노출 메모리를 포함하고, 상기 노출 메모리는 상기 노출 메모리의 제 1 사이즈 및 상기 노출 메모리의 기본 주소를 포함하고, 상기 메모리 주소는 상기 노출 메모리 내에 있고,

상기 중간 주소는 상기 NVDIMM 내의 상기 노출 메모리의 상기 제 1 사이즈, 상기 NVDIMM 내의 상기 노출 메모리의 상기 기본 주소, 상기 NVDIMM 내의 상기 노출 메모리의 논리 세그먼트의 제 2 사이즈, 상기 호스트 메모리 컨트롤러의 메모리 컨트롤러 동작 모드, 그리고 상기 호스트 메모리 컨트롤러에 의해 사용되는 논리-물리 변환(logical-to-physical transformation)에 응답하는 상기 메모리 주소와 다르고,

상기 호스트 메모리 컨트롤러는 타겟 메모리 주소를 상기 NVDIMM에 전송하고,

상기 타겟 메모리 주소는 상기 중간 주소에 응답하고 그리고 상기 메모리 주소에 저장된 값의 물리적 위치를 포함하고, 그리고

상기 노출 메모리는 직접적으로 주소화 가능한(addressable) 메모리를 포함하는 물품.

청구항 9

제 8 항에 있어,

상기 호스트 메모리 컨트롤러는 상기 중간 주소에 논리-물리(logical-to-physical) 매핑을 구현하는 물품.

청구항 10

삭제

청구항 11

삭제

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

발명의 설명

기술 분야

[0001] 본 발명은 메모리 장치에 관한 것으로, 좀 더 구체적으로 NVDIMM을 포함하는 장치 및 그것의 액세스 방법에 관한 것이다.

배경 기술

[0002] DIMM에 대한 요구 사항들은 수년 동안 정립되어 왔다. 1944년에 설립되어 1958년에 현재의 이름을 갖게 된 JEDEC(Joint Electron Device Engineering Council)은 DIMM 장치들에 대한 기술 사양들(Specifications)을 제공한다. DIMM 장치의 다양한 변형들이 존재하기 때문에, JEDEC은 DIMM이 동작하는 방법에 대한 다양한 표준에 대한 초안들을 작성해왔다.

[0003] DIMM들은 DIMM 슬롯(Slot) 커넥터에 연결되고, 메모리 채널의 주소 신호 라인들을 통해 호스트 프로세서와 통신한다. 또한, 호스트 프로세서(좀 더 자세히는, 메모리 컨트롤러)와 DIMM 간의 통신하는 방식은 잘 정립되어 있다.

[0004] DIMM들은 일반적으로 휘발성 메모리를 포함한다. 즉, 메모리 셀들 내의 데이터 값은 주기적으로 리프레시(Refresh)되어야 하며, 그렇지 않으면 메모리 셀들은 메모리에 저장된 데이터 값을 잃어버리게 될 것이다. 리프레시와 관련하여, DIMM은 JEDEC에 의해 정해진 최소의 리프레시 주기를 지원할 수 있어야 한다.

[0005] DIMM과 같은 휘발성 메모리와는 대조적으로, 불휘발성 메모리(NVM: Non-Volatile Memory)의 형태도 존재한다. 불휘발성 메모리의 형태는 주기적인 리프레시가 없이도 메모리에 저장된 데이터 값을 유지할 수 있다. 불휘발성

메모리의 하나의 예시적인 형태는 낸드(Nand) 플래시 메모리이다. 낸드 플래시 메모리는 USB(Universal Serial Bus) 키 및 SSD(Solid State Devices) 등에서 쉽게 찾아볼 수 있다.

[0006] DIMM 슬롯에 호환되는 불휘발성 메모리의 형태는 지속적으로 개발되어왔다. 하지만, 불휘발성 메모리는 DIMM과는 다른 원리에 따라 동작한다. 불휘발성 메모리의 형태가 DIMM 소켓(Socket)에서 동작하도록 하기 위해서는, 불휘발성 메모리는 호스트 프로세서 및 메모리 컨트롤러에게 DIMM으로 인식되도록 해야 한다. DIMM 내의 메모리 어드레싱(Addressing)은 불휘발성 메모리 형태와는 다르게 수행되기 때문에, 이는 해결되어야 할 몇 가지 문제들을 발생시킨다.

[0007] 따라서, DIMM 슬롯에서 불휘발성 메모리 형태를 사용함에 따라 발생하는 메모리 어드레싱 방법 및 다양한 이슈들을 해결하기 위한 방법이 요구된다.

발명의 내용

해결하려는 과제

[0008] 본 발명의 목적은 메모리 주소를 인터셉트하는 장치 드라이버를 이용하여 DIMM 슬롯 상의 인터페이스를 지원하는 NVDIMM을 포함하는 장치 및 그것의 액세스 방법을 제공하는데에 있다.

과제의 해결 수단

[0009] 본 발명의 실시 예에 따른 장치는 NVDIMM 및 장치 드라이버를 포함할 수 있다. NVDIMM(Non-Volatile Dual In-Line Memory Module)은 DIMM(Dual In-Line Memory Module) 소켓에 장착될 수 있다. 장치 드라이버는 호스트 프로세서 상에서 동작할 수 있다. NVDIMM은 메모리 및 노출 메모리를 포함하고, 노출 메모리는 노출 메모리의 제 1 사이즈 및 노출 메모리의 기본 주소를 포함할 수 있다. 장치 드라이버는 호스트 메모리 컨트롤러로 전송되는 메모리 주소를 인터셉트하고, 메모리 주소를 사전 매핑된 메모리 주소로 교체하도록 동작할 수 있다. 사전 매핑된 메모리 주소는 메모리 주소와 다를 수 있다.

[0010] 본 발명의 실시 예에 따른 NVDIMM을 포함하는 장치의 액세스 방법은 NVDIMM의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계, 메모리 주소를 중간 주소로 사전 매핑하는 단계, 중간 주소를 호스트 메모리 컨트롤러로 전송하는 단계를 포함할 수 있다. NVDIMM은 메모리 및 노출 메모리를 포함하고, 노출 메모리는 노출 메모리의 제 1 사이즈 및 노출 메모리의 기본 주소를 포함할 수 있다. 중간 주소는 메모리 주소와 다를 수 있다. 호스트 메모리 컨트롤러는 타겟 메모리 주소를 NVDIMM에 전송할 수 있다. 타겟 메모리 주소는 메모리 주소에 저장된 값의 물리적 위치를 의미할 수 있다.

[0011] 논 트랜지토리(Non-Transitory) 명령을 저장하는 유형의 스토리지 매체를 포함하는 장치에 있어, 논 트랜지토리 명령은 시스템에 의해 실행될 때, NVDIMM의 메모리 주소에 대해 액세스 하기 위한 요청을 수신하는 단계, 메모리 주소를 중간 주소로 사전 매핑하는 단계, 중간 주소를 호스트 메모리 컨트롤러로 전송하는 단계를 수행할 수 있다. NVDIMM은 메모리 및 노출 메모리를 포함하고, 노출 메모리는 노출 메모리의 제 1 사이즈 및 노출 메모리의 기본 주소를 포함할 수 있다. 중간 주소는 메모리 주소와 다를 수 있다. 호스트 메모리 컨트롤러는 타겟 메모리 주소를 NVDIMM에 전송할 수 있다. 타겟 메모리 주소는 메모리 주소에 저장된 값의 물리적 위치를 나타낼 수 있다.

발명의 효과

[0012] 본 발명에 따르면, 복수의 불휘발성 메모리 장치를 포함하는 NVDIMM이 DIMM 슬롯 상에서 동작할 수 있다. 이를 통해, NVDIMM은 속도가 빠른 메모리 채널을 이용할 수 있게 되고, 결과적으로 메모리의 통신 속도가 향상될 수 있다.

도면의 간단한 설명

[0013] 본 발명의 실시 예들은 제한적인 방법으로서가 아니라 예로서 도시되었으며, 첨부 도면에서 유사한 참조 번호는 유사한 요소를 참조한다.

도 1은 본 발명의 실시 예에 따른 호스트 프로세서와 통신하는 NVDIMM을 보여주는 블록도이다.

도 2는 도 1의 장치 드라이버의 설계에 있어 고려되는 다양한 사항들을 보여주는 그림이다.

도 3은 도 1의 호스트 프로세서 및 NVDIMM을 지원할 수 있는 컴퓨터 시스템을 보여주는 그림이다.

도 4는 도 1의 NVDIMM을 사용할 수 있는 시스템의 구조를 보여주는 그림이다.

도 5는 본 발명의 실시 예에 따른 도 1의 NVDIMM의 메모리 공간의 레이아웃을 예시적으로 보여주는 그림이다.

도 6은 본 발명의 실시 예에 따른 도 1의 NVDIMM 내의 메모리 주소에 액세스 하기 위한 요청을 인터셉트하고 요청된 메모리 주소를 사전에 매핑하는 도 1의 장치 드라이버를 보여주는 그림이다.

도 7은 본 발명의 실시 예에 따른 주소 에일리어스를 수행하는 도 1의 장치 드라이버를 보여주는 그림이다.

도 8은 본 발명의 다른 실시 예에 따른 도 1의 NVDIMM의 메모리 공간의 레이아웃을 예시적으로 보여주는 그림이다.

도 9는 본 발명의 실시 예에 따른 도 1의 NVDIMM 및 장치 드라이버가 도 1의 NVDIMM 내의 메모리 주소들에 대한 요청들을 처리하는 방법을 보여주는 순서도이다.

도 10은 본 발명의 실시 예에 따른 도 1의 NVDIMM이 도 1의 호스트 컴퓨터에 메모리를 보여주는 방법을 보여주는 순서도이다.

발명을 실시하기 위한 구체적인 내용

- [0014] 앞의 일반적인 설명 및 다음의 상세한 설명 모두 예시적이라는 것이 이해되어야 하며, 청구된 발명의 부가적인 설명이 제공되는 것으로 여겨져야 한다. 참조 부호들이 본 발명의 바람직한 실시 예들에 상세히 표시되어 있으며, 그것의 예들이 참조 도면들에 표시되어 있다. 가능한 어떤 경우에도, 동일한 참조 번호들이 동일한 또는 유사한 부분을 참조하기 위해서 설명 및 도면들에 사용된다.
- [0015] 이하에서는, 불휘발성 메모리 장치가 본 발명의 특징 및 기능을 설명하기 위한 예로서 사용될 것이다. 하지만, 이 기술 분야에 정통한 사람은 여기에 기재된 내용에 따라 본 발명의 다른 이점들 및 성능을 쉽게 이해할 수 있을 것이다. 본 발명은 다른 실시 예들을 통해 또한, 구현되거나 적용될 수 있을 것이다. 게다가, 상세한 설명은 본 발명의 범위, 기술적 사상 그리고 다른 목적으로부터 상당히 벗어나지 않고 관점 및 응용에 따라 수정되거나 변경될 수 있다.
- [0016] 스토리지 매체로서 낸드 플래시 메모리를 사용하는 SSD(Solid State Devices)와 같은 솔리드 스테이트 컴퓨터 스토리지 장치들은 영구적인 데이터 스토리지 서비스를 제공하고, 다양한 방법을 통해 호스트 컴퓨터들과 연결될 수 있다. 운영적 관점에서, 중요한 구성요소들은 영구적인 저장 능력을 위해 스토리지 장치들이 이용하는 낸드 플래시 메모리 회로들, 호스트 인터페이스 로직, 그리고 프로세싱 및 매체 컨트롤 로직 등이 있다.
- [0017] 한 가지 중요한 호스트 인터페이스 방법으로 NVMe(Non-Volatile Memory Express)이 있다. 호스트들은 PCIe(Peripheral Component Interconnect Express) 채널을 통해 NVMe 스토리지 장치들과 인터페이스할 수 있다. 이는 복수의 표준화된 방식으로 메모리 매핑된 서브미션 큐(Submission Queue) 및 컴플리션 큐 페어(Completion Queue pair)를 이용하여 수행될 수 있다.
- [0018] 호스트의 DRAM(Dynamic Random Access Memory) 주소 공간 내의 할당된 메모리 위치와 연관된 데이터들 및 커맨드들을 통해, 호스트 프로세서들은 NVMe 스토리지 장치들에게 데이터 I/O(Input/Output) 요청들을 전송한다. 이러한 할당된 위치 주소들은 NVMe 서브미션 큐들과 호스트 메모리 매핑된다. NVMe 장치들은 전송된 요청들을 감지하고, 장치 프로세싱 로직이 요청의 이행을 위해 요청들을 장치에 옮길 때까지 이를 보관한다.
- [0019] NVMe 장치들이 요청들을 수행한 경우, NVMe 장치들은 호스트가 요청을 배치하는 서브미션 큐와 연관된 컴플리션 큐에 응답들을 전송한다. 서브미션 큐와 같은 컴플리션 큐는 호스트 메모리 매핑된다. 호스트들은 응답의 존재를 감지하고, 메모리 매핑된 데이터의 이동 방법을 이용하여 응답에 접근하고, 그들이 응답을 수신하였는지를 나타내는 방법을 갖고 있다.
- [0020] NVMe와 대조적으로, NVDIMM(Non-Volatile Dual In-Line Memory Module)은 호스트들과 연결하기 위해 호스트 DRAM 메모리 채널을 이용한다. NVDIMM은 NVMe를 이용하는 SSD보다 현저히 빠르게 호스트 프로세서들과 통신할 수 있다. 이는 NVDIMM은 현저히 빠른 호스트 프로세서 연결 방법을 사용하기 때문이다. 메모리 채널의 전송률은 상대적으로 PCIe 연결보다 빠르다. 하지만, 이러한 형태는 추가적인 문제를 갖는다. NVDIMM이 호스트 메모리 채널에 있기 때문에, NVDIMM은 호스트 메모리 컨트롤러들에게 JEDEC(Joint Electron Device Engineering Council) 표준을 준수하는 DRAM DIMM으로 인식되어야 하고, 표준 DRAM DIMM 소켓을 이용하여 호스트 메모리 채널

널과 연결되어야 한다. 특히, NVDIMM은 완전하게 JEDEC DIMM 기술 사양들을 준수해야만 한다.

- [0021] JEDEC DIMM 기술 사양들을 준수하기 위해서, NVDIMM은 호스트에게 그것이 특정한 메모리 용량을 갖는 DRAM DIMM 이라고 보고해야만 하고, 보고될 DRAM DIMM 타입과 구별되지 않도록 동작해야만 한다. 또한, NVDIMM은 호스트에게 메모리 맵 동작을 위한 인터페이스를 제공해야만 한다. 이러한 인터페이스는 메모리 채널의 속도에서 동작하고, 시스템 메모리와 같이 구별되지 않도록 동작할 것이다. 일반적으로, NVDIMM은 적은 양의 메모리만을 메모리 매핑된 호스트 데이터의 전송에 사용되는 메모리 채널 상에서 액세스 가능하도록 노출한다. 노출되는 메모리는 DRAM, SRAM(Static Random Access Memory)와 같은 휘발성 메모리 또는 NVDIMM 내의 불휘발성 메모리의 일부분일 수 있다.
- [0022] 도 1은 본 발명의 실시 예에 따른 호스트 프로세서와 통신하는 NVDIMM을 보여주는 블록도이다. 도 1에서, 호스트 메모리 컨트롤러(115) 및 메모리 신호 라인들(120)을 통해 호스트 프로세서(110)와 통신하는 NVDIMM(105)이 도시되었다.
- [0023] 호스트 메모리 컨트롤러(115)는 NVDIMM(105)이 제공하는 메모리와의 읽기 및 쓰기 동작을 수행한다. NVDIMM(105)은 DIMM 소켓(125)을 통해 호스트 메모리 컨트롤러(115) 및 호스트 프로세서(110)와 통신한다. DIMM 소켓(125)은 호스트 프로세서(110) 및 NVDIMM(105)을 포함하는 컴퓨터 내에 장착된 표준 DIMM 소켓일 수 있다.
- [0024] NVDIMM(105)은 불휘발성 메모리(130), 호스트 인터페이스 로직(135), 프로세싱 및 컨트롤 로직(140), 내장 DRAM(145), 그리고 노출 메모리(150)를 포함할 수 있다. 불휘발성 메모리(130)는 어떠한 형태의 불휘발성 메모리도 될 수 있다. 본 발명의 일 실시 예로서, 불휘발성 메모리(130)는 낸드 플래시 스토리지를 포함할 수 있다. 다른 실시 예로서, 불휘발성 메모리(130)는 NOR 플래시 스토리지, PCM(Phase Change Memory), STT(Spin-Transfer Torque), 그리고 MRAM(Magnetic RAM)과 같은 다른 형태의 불휘발성 메모리를 포함할 수 있다. 호스트 인터페이스 로직(135)은 호스트 프로세서(110)와 NVDIMM(105) 사이의 통신의 제어를 담당한다. 프로세싱 및 컨트롤 로직(140)은 불휘발성 메모리(130)의 사용의 제어를 담당한다.
- [0025] 또한, 노출 메모리(150)는 메모리 매핑 동작을 위해 호스트 프로세서(110)에 노출된 메모리를 포함한다. 노출 메모리(150)는 내장 DRAM(145)의 일부, 불휘발성 메모리(130)의 일부, 또는 NVDIMM(105)과는 별도의 구성요소일 수 있다. 예를 들어, 본 발명의 일 실시 예로서, 불휘발성 메모리(130)는 노출 메모리(150)로서 매핑된 1Mb 및 이를 포함하는 8Gb의 전체 메모리를 포함할 수 있다. 본 발명의 다른 실시 예는 불휘발성 메모리(130)의 총 메모리의 거의 대부분이나 노출 메모리(150)의 거의 대부분, 또는 둘 모두를 포함할 수 있다.
- [0026] NVDIMM(105)을 이용하여 DIMM을 시뮬레이트(simulate) 하면서 하나의 이슈가 발생할 수 있다. 즉, 호스트 프로세서(110)는, 좀 더 구체적으로는 호스트 프로세서(110) 상에서 운영되는 프로그램들은, 메모리를 특별한 관점으로 볼 수 있다. 이러한 특별한 관점으로, 다른 어플리케이션들은 각각의 사용에 맞게 모든 메모리를 이용할 수 있는 것처럼 동작할 수 있다. 호스트 메모리 컨트롤러(115)는 메모리 주소들의 논리-물리 변환을 수행하여 이러한 메모리의 관점을 수용한다. 즉, 어플리케이션이 메모리로부터 데이터를 읽거나 쓰기 위해 메모리 위치에 액세스를 요청하는 경우, 어플리케이션은 그것이 알고 있는 메모리 주소를 제공한다. 이 메모리 주소는 원하는 메모리 셀이 위치하는 장치 상의 실제 메모리 주소가 아닌 경우가 종종 있다. 호스트 메모리 컨트롤러(115)는 이러한 논리적인 메모리 관점(어플리케이션 각각이 자체적인 논리적인 관점을 갖고 있기 때문에)과 메모리 모듈들의 물리적 주소 사이의 변환을 처리한다.
- [0027] 문제는 NVDIMM(105) 내의 노출 메모리(150)가 직접 메모리에 주소화할 수 있고, 호스트 메모리 컨트롤러(115)에 의한 이러한 논리-물리 변환을 요구하지 않는 것이다. 호스트 메모리 컨트롤러(115)가 이러한 논리-물리 변환을 수행하는 경우, 호스트 메모리 컨트롤러(115)는 원하는 메모리 셀의 정확한 위치를 막는 방법으로 주소를 컨볼브(Convolve)한다.
- [0028] 이러한 문제에 대한 해결책은 장치 드라이버(160)를 도입하는 것이다. 장치 드라이버(160)는 요청된 메모리 주소를 사전에 매핑하고, 이에 호스트 메모리 컨트롤러(115)가 그것의 논리-물리 변환을 수행한 후에 NVDIMM(105)이 예상하는 직접적인 메모리 주소를 수신한다.
- [0029] 장치 드라이버(160)가 이러한 사전 매핑을 수행하는 방법은 여러 요소들에 의존한다.
- [0030] 도 2는 도 1의 장치 드라이버의 설계에 있어 고려되는 다양한 사항들을 보여주는 그림이다. 장치 드라이버(160)의 동작에 들어가는 요소들은 노출 메모리의 사이즈(205), 노출 메모리의 기본 주소(210), 노출 메모리의 논리 세그먼트(Segment)의 사이즈(215), 메모리 컨트롤러 동작 모드(220), 그리고 메모리 컨트롤러에 의해 이용되

는 논리-물리 변환(225) 등을 포함할 수 있다. 논리 세그먼트들은 도 5 및 도 8을 참조하여 설명될 것이다.

[0031] 장치 드라이버(160)는 메모리 컨트롤러 동작 모드(220) 및 호스트 메모리 컨트롤러(115)의 논리-물리 변환(225)을 결정하기 위해 기본 시스템의 버스를 탐색할 수 있다. 다른 요소들은 도 1의 NVDIMM(105)의 설계에 있어 결정되고, 장치 드라이버(160)는 사전에 이를 알 수 있다.

[0032] 본 발명의 일 실시 예로서, 메모리 컨트롤러 동작 모드(220)는 페이지 오픈 컨트롤러 모드로 동작할 수 있다. 페이지 오픈 컨트롤러 모드는 호스트 메모리 컨트롤러(115)가 메모리 액세스 주소들이 가까이에 이격되어 있다고 가정하는 모드이다. 예를 들어, 이러한 요청들은 종종 데스크탑 컴퓨터 시스템들이나 I/O 요청들에서 발생한다. 본 발명의 다른 실시 예로서, 메모리 컨트롤러 동작 모드(220)는 페이지 클로즈(Closed) 컨트롤러 모드로 동작할 수 있다. 페이지 클로즈 컨트롤러 모드는 호스트 메모리 컨트롤러(115)가 메모리 요청들 중에 감소된 로컬리티(Locality)의 액세스가 있다고 가정하는 모드이다. 일반적으로, 비록 호스트 메모리 컨트롤러(115)가 기본 시스템이 다시 시작될 때 다른 동작 모드로 변경할 수 있다 해도, 호스트 메모리 컨트롤러(115)는 기본 시스템의 동작 중에 모드를 변경하지 않고 오직 하나의 모드로 동작한다. 논리-물리 변환(225)은 자체로 랭크 어드레스(Rank Address), 뱅크 그룹 어드레스(Bank Group Address), 뱅크 어드레스(Bank Address), 로우 어드레스(Row Address), 그리고 컬럼 어드레스(Column Address) 등의 다양한 요소들을 포함할 수 있다.

[0033] 도 1의 장치 드라이버(160)가 사전 매핑을 수행하는 방법의 예를 설명하기 전에, 추가적인 정보를 언급하는 것은 유용할 것이다.

[0034] 도 3은 도 1의 호스트 프로세서 및 NVDIMM을 지원할 수 있는 컴퓨터 시스템을 보여주는 그림이다. 도 3을 참조하면, 컴퓨터 시스템(305)은 컴퓨터(310), 모니터(315), 키보드(320), 그리고 마우스(325)를 포함할 수 있다. 당업자는 컴퓨터 시스템(305)에 다른 구성 요소들이 포함될 수 있음을 쉽게 이해할 수 있을 것이다. 예를 들어, 컴퓨터 시스템(305)은 프린터와 같은 I/O 장치 등을 더 포함할 수 있다. 더불어, 컴퓨터 시스템(305)은 도 1에 도시된 기존의 내부 구성 요소들을 포함할 수 있다. 예를 들어, 기존의 내부 구성 요소들은 도 1의 호스트 프로세서(110), 도 1의 NVDIMM(105), 그리고 추가적인 스토리지 등을 포함할 수 있다.

[0035] 도 3에 도시되지 않았지만, 당업자는 컴퓨터 시스템(305)이 그래픽 카드, 모뎀 등의 다른 내부 구성 요소들을 포함할 수 있다는 것을 쉽게 이해할 수 있을 것이다. 또한, 당업자는 컴퓨터 시스템(305)이 다른 컴퓨터 시스템과 직접적으로 혹은 다른 타입의 네트워크(미도시)를 통해 상호 작용할 수 있음을 쉽게 이해할 수 있을 것이다. 더불어, 비록 도 3은 컴퓨터 시스템(305)이 기존의 데스크 탑(Desktop) 컴퓨터로서 도시되었지만, 당업자는 컴퓨터 시스템(305)이 어떠한 형태의 머신 또는 컴퓨팅 장치가 될 수 있음을 쉽게 이해할 수 있을 것이다. 예를 들어, 상술한 머신 또는 컴퓨팅 장치는 랩 탑(Laptop) 컴퓨터, 태블릿 컴퓨터, PDA(Personal Digital Assistant), 또는 스마트 폰 등을 포함할 수 있다.

[0036] 도 4는 도 1의 NVDIMM을 사용할 수 있는 시스템의 구조를 보여주는 그림이다. 도 4를 참조하면, 일반적으로, 컴퓨터 시스템(305)은 하나 또는 하나 이상의 호스트 프로세서(110)를 포함할 수 있다. 호스트 프로세서(110)는 호스트 메모리 컨트롤러(115) 및 클록(405)을 포함할 수 있다. 클록(405)은 컴퓨터 시스템(305)의 구성 요소들의 동작을 동기화하기 위해 사용될 수 있다. 또한, 호스트 프로세서(110)는 메모리(105)와 연결될 수 있다. 예를 들어, 메모리(105)는 NVDIMM, RAM(Random Access Memory), ROM(Read-Only Memory), 또는 다른 상태 저장 매체 등을 포함할 수 있다. 호스트 프로세서(110)는 또한 스토리지 장치(410) 및 네트워크 커넥터(415)와 연결될 수 있다. 예를 들어, 네트워크 커넥터(415)는 이더넷(Ethernet) 커넥터일 수 있다. 호스트 프로세서(110)는 또한 버스(420)와 연결될 수 있다. 버스(420)는 다른 구성 요소 중에 I/O 엔진(430)을 이용하여 제어될 수 있는 I/O 인터페이스 포트들 및 유저 인터페이스(425)와 연결될 수 있다.

[0037] 도 5는 본 발명의 실시 예에 따른 도 1의 NVDIMM의 메모리 공간의 레이아웃을 예시적으로 보여주는 그림이다. 도 5에서, 불휘발성 메모리(130)가 도시되었다. 불휘발성 메모리(130)는 기본 주소(210)에서 시작하는 총 메모리 용량(505)을 포함할 수 있다. 불휘발성 메모리(130)는 논리 세그먼트들(510, 515, 520, 525, 530)과 같은 논리 세그먼트로 구분될 수 있다. 논리 세그먼트는 불휘발성 메모리(130)가 세분화된 구역이다. 비록 일반적으로는 논리 세그먼트들(510, 515, 520, 525, 530)이 2의 거듭 제곱 형태로 구성된 다수의 바이트들을 포함하지만, 논리 세그먼트들(510, 515, 520, 525, 530)은 어떠한 원하는 사이즈(215)도 될 수 있다.

[0038] 논리 세그먼트들(510, 515, 520, 525, 530)의 사이즈(215)와 불휘발성 메모리(130)의 총 메모리 용량(505) 사이에는 어떠한 관계도 요구되지 않는다. 또한, 논리 세그먼트들(510, 515, 520, 525, 530)의 사이즈(215)와 노출 메모리(150)의 사이즈(205) 사이에도 어떠한 관계도 요구되지 않는다. 하지만, 만약 논리 세그먼트들(510,

515, 520, 525, 530)의 사이즈(215)가 너무 작은 경우, 논리 세그먼트들(510, 515, 520, 525, 530) 각각은 메모리 요청에 대해 충분하지 못한 스토리지를 가질 수 있다.

[0039] 본 발명의 일 실시 예로서, 도 5는 8Gb의 불휘발성 메모리(130)의 총 메모리 용량(505), 1Mb의 노출 메모리(150)의 사이즈(205), 그리고 4Kb의 논리 세그먼트들(510, 515, 520, 525, 530) 각각의 사이즈(215)를 보여준다. 하지만, 본 발명의 다른 실시 예들은 불휘발성 메모리(130)의 총 메모리 용량(505), 노출 메모리(150)의 사이즈(205), 그리고 논리 세그먼트들(510, 515, 520, 525, 530) 각각의 사이즈(215) 중 일부 혹은 모두에 대해 어떠한 값들도 지원할 수 있다. 더불어, 도 5 및 도 8은 노출 메모리(150)를 불휘발성 메모리(130)의 일부로 도시하고 있지만, 노출 메모리(150)는 불휘발성 메모리(130)와 분리되어 구현될 수 있다.

[0040] 도 6은 본 발명의 실시 예에 따른 도 1의 NVDIMM 내의 메모리 주소에 액세스 하기 위한 요청을 인터셉트하고 요청된 메모리 주소를 사전에 매핑하는 도 1의 장치 드라이버를 보여주는 그림이다. 도 6을 참조하면, 어플리케이션 또는 다른 프로세스들은 메모리 주소(605)를 생성할 수 있다. 메모리 주소(605)는 도 1의 NVDIMM(105) 내의 메모리 주소에 해당할 수 있다. 메모리 주소(605)를 호스트 메모리 컨트롤러(115)가 처리하도록 허용하기 보다, 장치 드라이버(160)는 메모리 주소(605) 및 사전 매핑된 메모리 주소를 인터셉트할 것이다. 호스트 메모리 컨트롤러(115)는 도 2의 논리-물리 변환(225)을 수행할 것이다. 사전 매핑의 결과는 사전 매핑된 메모리 주소(610)이다. 또는, 사전 매핑된 메모리 주소(610)는 중간 주소(Intermediate Address)라고도 할 수 있다. 이어, 장치 드라이버(160)는 사전 매핑된 메모리 주소(610)를 호스트 메모리 컨트롤러(115)에 제공할 것이다. 호스트 메모리 컨트롤러(115)는 사전 매핑된 메모리 주소(610) 상의 논리-물리 변환(225)을 수행할 수 있다. 해당 결과는 타겟 메모리 주소(615)이다. 타겟 메모리 주소(615)는 도 1의 NVDIMM가 원하는 메모리 셀에 저장한 곳의 실제 주소이다.

[0041] 다시 도 1을 참조하면, 호스트 시스템 내에서, 모든 DRAM 서브 시스템(Sub-System)의 주소 신호들은 DIMM 소켓 각각에 도달한다. DIMM 소켓 각각은 NVDIMM(105)가 장착된 DIMM 소켓(125)을 포함한다. 주소 분산을 하기 위해서, 장치 드라이버(160)는 논리 메모리 세그먼트에 노출된 정확한 NVDIMM을 찾기 위해 일부 신호들을 사용한다. 비록 남은 신호들이 도 1의 NVDIMM(105)에 의해 사용될 수 있다 하더라도, 남은 신호들의 값은 매핑과 관계 없다. 상술된 바와 같이, 정확한 매핑은 이와는 다를 수 있고, 논리 세그먼트 사이즈(215) 및 호스트 메모리 컨트롤러 동작 모드(220)에 의존할 수 있다.

[0042] 장치 드라이버(160)가 사전 매핑을 수행하도록 동작하는 방법의 일 예로서, NVDIMM(105)이 도 5의 8Gb의 총 메모리 용량(505), 도 2의 1Mb의 노출 메모리(150)의 사이즈(205), 그리고 도 2의 4Kb의 논리 세그먼트들(510, 515, 520, 525, 530) 각각의 사이즈(215)를 포함한다고 가정한다. 또한, 노출 메모리(150)가 메모리 신호 라인(120) 상에서 다음의 입력 주소 비트들(row[2:0], column[9:0], ba[1:0], bg[1:0], 3'b000)과 매핑된다고 가정한다. 상술된 바는 쉽게 이해하기 위해 베릴로그(Verilog) 코드열로 기술되었을 뿐이고, 매핑 방법이 반드시 하드웨어 기술 사양의 문서로서 표현되어야 한다는 것이 아니다.

[0043] 예를 들어, 만약 메모리 컨트롤러 동작 모드(220)가 페이지 오픈 모드인 경우, 도 1의 논리-물리 변환(225)은 다음의 변환(rank[1:0], row[16:0], ba[1:0], bg[1:0], column[9:0], 3'h0)에 따라 비트들을 매핑할 수 있다. 또한, 장치 드라이버(160)는 다음의 매핑(16'hxxxx, row[2:0], ba[1:0], bg[1:0], column[9:0], 3'h0)을 이용하여 메모리 주소를 사전 매핑할 수 있다. 여기서 'x'는 'Don't care'를 의미하는 값이고, 'x' 대신에 결과에 영향을 주지 않는 어떠한 값이라도 사용될 수 있다.

[0044] 한편, 예를 들어, 만약 메모리 컨트롤러 동작 모드(220)가 페이지 클로즈 모드인 경우, 도 1의 논리-물리 변환(225)은 다음의 변환(rank[1:0], column[9:3], row[16:0], ba[1:0], bg[1:0], column[2:0], 3'h0)에 따라 비트들을 매핑할 수 있다. 또한, 장치 드라이버(160)는 다음의 매핑(2'hx, column[9:3], 14'hxxxx, row[2:0], ba[1:0], bg[1:0], column[2:0], 3'h0)을 이용하여 메모리 주소를 사전 매핑할 수 있다. 또한, 여기서 'x'는 'Don't care'를 의미하는 값이다.

[0045] 상술한 사전 매핑의 예는 현실적인 매핑이 되어야 하는 것이 아니라 구성일 뿐이다. 각각의 그룹 내의 주소 비트들은 DRAM 기술 사양으로 사용되고, 상대적으로 시스템 주소에 대한 예시적인 고장(Breakdown)으로 선택된다. 예를 들어, DRAM 기술 사양은 로우(Row) 비트의 수, 컬럼(Column) 비트의 수, 뱅크 비트의 수, 그리고 뱅크 그룹 비트의 수를 포함할 수 있다. 다만, 현실의 메모리 컨트롤러들에 대한 사전 매핑은 유사한 방법으로 수행될 수 있다.

[0046] 메모리 신호 라인(120)을 통해, NVDIMM(105)은 NVMe를 이용하여 동작하는 불휘발성 메모리 장치보다 현저히 빠

르게 호스트 프로세서(110)와 통신할 수 있다. 이는 메모리 채널의 전송률은 상대적으로 PCIe 연결보다 빠르기 때문이다. 하지만, 이에겐 잠재적인 문제가 존재한다. 모든 메모리 매핑된 I/O의 동작은 매우 좁은 주소 범위(예를 들어, 도 5에서 도시된 것과 같이 1Mb)에서 이루어지기 때문에, NVDIMM(105)은 잠재적으로 로우 햄머(Row Hammer)와 같은 DRAM DIMM 표준 결함에 의해 발생하는 심각한 성능 제약에 노출될 수 있다.

[0047] 간단히 설명하면, 특정 메모리 셀에 반복적으로 액세스가 수행되는 경우, 로우 햄머는 근처의 하나 또는 하나 이상의 메모리 셀에 변화를 유발하여 발생할 수 있다. 로우 햄머는 DRAM 표준이 처음에 개발되었을 때에는 JEDEC 표준 위원회에 알려지지 않았다. 로우 햄머 문제에 대하여, 새로운 JEDEC 기술 사양들은 메모리 컨트롤러(예를 들어, 도 1의 호스트 메모리 컨트롤러(115))에 동일한 DRAM의 로우 어드레스에 액세스하는 것을 주기를 감소시켜 억제(Throttle) 하도록 하였다. 이는 타겟 로우 리프레시(Target Row Refresh)라고 부른다.

[0048] DRAM의 특정 로우 어드레스에의 액세스를 느리게 하는 것이 로우 햄머를 방지하는 반면에, 액세스 주기를 감소시키는 것은 메모리 동작을 느리게 한다. 또한, 도 1의 NVDIMM(105)과 같이 로우 햄머 문제가 적용되지 않는 메모리가 사용되는 경우, 억제(Throttling)는 필요하지 않다. 하지만, 도 1의 NVDIMM(105)은 JEDEC 기술 사양을 준수하고 호스트 프로세서(110)와 올바르게 인터페이스하기 위하여 일반적인 DIMM으로 보여져야 하기 때문에, 호스트 메모리 컨트롤러(115)는 로우 햄머가 문제되지 않는다는 것을 알지 못한다.

[0049] 도 1의 노출 메모리(150)의 주소 범위가 좁을수록, 도 1의 호스트 메모리 컨트롤러(115)가 로우 햄머를 피하기 위해 도 1의 NVDIMM(105)에의 요청을 억제(Throttle)할 확률이 증가한다. 따라서, 호스트 메모리 컨트롤러(115)가 로우 햄머가 문제가 된다고 생각하지 않도록 하는 것이 필요할 것이다.

[0050] 도 1의 호스트 메모리 컨트롤러(115)가 로우 햄머가 문제가 된다고 생각하지 않도록 하는 한 가지 방법은 메모리 주소들이 도 1의 메모리 컨트롤러(115)에 도달하기 전에 메모리 주소들을 에일리어스(Alias)하는 것이다. 도 7을 참조하여, 장치 드라이버(160)가 에일리어스를 사용하는 방법이 설명될 것이다.

[0051] 도 7은 본 발명의 실시 예에 따른 주소 에일리어스를 수행하는 도 1의 장치 드라이버를 보여주는 그림이다. 도 7을 참조하면, 장치 드라이버(160)는 메모리 주소(605)를 사전 매핑된 메모리 주소(610)로 사전 매핑한다. 하지만, 사전 매핑의 예로서 상술된 바와 같이, 사전 매핑은 관련 없는 값을 갖는 비트들을 포함할 수 있다. 어떠한 값도 이용될 수 있다. 도 1의 호스트 메모리 컨트롤러(115) 내의 물리 로우 비트로 매핑된 하나 또는 하나 이상의 'Don't care' 비트를 변경함에 의해, 에일리어스(Alias)는 도 6의 타겟 메모리 주소(615)에 대해 생성될 수 있다. 따라서, 예를 들어, 사전 매핑 메모리 주소 에일리어스들(705, 710) 각각은 원하는 메모리 주소에 대한 에일리어스를 대표하지만, 다른 비트 패턴들을 포함한다. 특히, 'Don't care' 비트들 내의 다른 비트 패턴들을 포함한다.

[0052] 장치 드라이버(160)는 'Don't care' 비트들을 변경하여 에일리어스를 생성하기 위해 원하는 어떠한 테크닉도 사용할 수 있다. 일 실시 예로서, 장치 드라이버(160)는 'Don't care' 비트들에 대한 랜덤 값들(혹은, 슈도 랜덤(Pseudo-random) 값)을 생성할 수 있다. 다른 실시 예로서, 장치 드라이버(160)는 'Don't care' 비트들에 대한 순차적인 값들을 생성할 수 있다. 다른 실시 예는 'Don't care' 비트들에 대한 값들을 생성하는 다른 방법들을 포함할 수 있다.

[0053] 다른 실시 예들로서, 장치 드라이버(160)는 액세스된 메모리 주소들을 추적할 수 있다. 또한, 도 1의 호스트 메모리 컨트롤러(115)가 도 1의 NVDIMM(105)에 대한 액세스를 억제(Throttle)하려고 하는 경우에, 장치 드라이버(160)는 'Don't care' 비트들만을 변경할 수 있다. 다른 실시 예들로서, 도 1의 호스트 메모리 컨트롤러(115)가 도 1의 NVDIMM(105)에 대한 액세스를 억제(Throttle)하려고 하는 것과 관계 없이, 장치 드라이버(160)는 각각의 메모리 액세스에 'Don't care' 비트의 값을 변경할 수 있다.

[0054] 도 1의 호스트 메모리 컨트롤러(115)가 로우 햄머가 문제라고 생각하는 것을 방지하기 위한 다른 방법은 도 1의 노출 메모리(150)를 불 연속적인 메모리 주소들을 갖는 논리 세그먼트들로 파티션(Partition)하는 것이다.

[0055] 도 8은 도 1의 노출 메모리(150)가 불 연속적인 메모리 주소들을 갖는 논리 세그먼트들로 파티션되는 방법을 예시적으로 보여준다.

[0056] 도 8은 본 발명의 다른 실시 예에 따른 도 1의 NVDIMM의 메모리 공간의 레이아웃을 예시적으로 보여주는 그림이다. 도 8을 참조하면, 도 1의 노출 메모리(150)는 사이즈(205)의 논리 세그먼트들의 일부만을 포함한다. 사이즈(205)는 도 1의 노출 메모리(150)의 실제 사이즈보다 크다. 예를 들어, 도 1의 노출 메모리(150)는 논리 세그먼트들(805, 810, 815, 820)만을 포함할 수 있다. 반면에, 사이즈(205) 범위 내의 논리 세그먼트들(825, 830, 835)은 도 1의 노출 메모리(150)의 일부가 아니다. 노출 메모리(150) 내의 논리 세그먼트들에 대한 메모리 주소

들이 연속적이지 아니기 때문에, 로우 햄머는 더 이상 문제가 되지 않는다.

- [0057] 도 9는 본 발명의 실시 예에 따른 도 1의 NVDIMM 및 장치 드라이버가 도 1의 NVDIMM 내의 메모리 주소들에 대한 요청들을 처리하는 방법을 보여주는 순서도이다. 도 9를 참조하면, 905 단계에서, 도 1의 NVDIMM(105)은 도 1의 노출 메모리(150)를 호스트 컴퓨터(좀 더 상세하게는, 도 1의 호스트 메모리 컨트롤러(115))에게 공개(Open)할 수 있다.
- [0058] 910 단계에서, 도 1의 장치 드라이버(160)는 도 1의 호스트 메모리 컨트롤러(115)에게 갈 예정인 도 1의 메모리 주소(605)에 액세스하기 위한 요청을 인터셉트할 수 있다.
- [0059] 915 단계에서, 도 1의 장치 드라이버(160)는 도 6의 메모리 주소(605)에 액세스하기 위한 요청을 수신할 수 있다. 920 단계에서, 도 1의 장치 드라이버(160)는 도 6의 사전 매핑된 메모리 주소(610)를 생성하기 위해 도 6의 메모리 주소(605)를 사전 매핑할 수 있다.
- [0060] 925 단계에서, 메모리 주소(605)의 도 7의 사전 매핑된 메모리 주소 에일리어스(705, 710)를 생성하기 위해, 도 1의 장치 드라이버(160)는 도 6의 사전 매핑된 메모리 주소(610) 내의 하나 또는 하나 이상의 비트들을 변형할 수 있다. 도 7을 참조하여 상술된 바와 같이, 도 6의 사전 매핑된 메모리 주소(610)의 도 7의 사전 매핑된 메모리 주소 에일리어스(705, 710)를 생성하는 것은 선택적인 것이고, 따라서 925 단계 또한 선택적이다.
- [0061] 930 단계에서, 결과적으로, 도 1의 장치 드라이버(160)는 사전 매핑된 메모리 주소 에일리어스(705, 710) 또는 도 6의 사전 매핑된 메모리 주소(610)를 도 1의 호스트 메모리 컨트롤러(115)에 전송할 수 있다.
- [0062] 도 10은 본 발명의 실시 예에 따른 도 1의 NVDIMM이 도 1의 호스트 컴퓨터에 메모리를 보여주는 방법을 보여주는 순서도이다. 도 10을 참조하면, 1005 단계에서, 도 1의 NVDIMM(105)은 도 1의 불휘발성 메모리(130) 내의 연속된 메모리 블록을 선택할 수 있다. 또한, 상술한 바와 같이, 도 1의 노출 메모리(150)는 도 1의 불휘발성 메모리(130)보다 다른 곳에서 저장될 것이다. 도 1의 노출 메모리(150)가 저장된 곳을 변경하는 것을 제외하고, 도 10의 순서도는 이러한 상황들에서 변형 없이 사용될 수 있다. 도 1의 NVDIMM(105)은 어떠한 원하는 방식으로 도 메모리의 연속적인 블록을 선택할 수 있다. 예를 들어, 도 1의 NVDIMM(105)은 기본 주소 '0'에서 시작하는 연속적인 블록, 도 5의 총 메모리 용량(505)의 상단에서 끝나는 연속적인 블록, 혹은 다른 형태의 메모리의 연속적인 블록을 선택할 수 있다.
- [0063] 1010 단계에서, 도 1의 NVDIMM(105)은 메모리의 연속된 블록을 논리 세그먼트들로 분할할 수 있다. 예를 들어, 논리 세그먼트는 도 5의 논리 세그먼트들(510, 515, 520, 525, 530)일 수 있다. 1015 단계에서, 도 1의 NVDIMM(105)은 도 1의 노출 메모리(150)와 같이 논리 세그먼트들(예를 들어, 도 5의 논리 세그먼트들(510, 515, 520, 525, 530))을 노출시킬 수 있다.
- [0064] 1015 단계 대신에, 1020 단계에서, 도 1의 NVDIMM(105)은 도 1의 불휘발성 메모리(130) 내의 메모리의 불 연속적인 블록을 선택할 수 있다. 블록들 각각은 도 2의 동일한 사이즈(215)를 가질 수 있다. 1005 단계와 같이, 도 1의 NVDIMM(105)은 어떠한 원하는 방식으로 도 메모리의 불 연속적인 블록을 선택할 수 있다. 1025 단계에서, 도 1의 NVDIMM(105)은 불 연속적인 메모리의 블록들을 논리 세그먼트들(805, 810, 815, 820)로서 노출시킬 수 있다.
- [0065] 도 10은 도 1의 NVDIMM(105)가 메모리의 불 연속적인 블록 또는 연속적인 블록 중에 하나를 노출하는 것으로 제안하고 있지만, 다른 실시 예들은 두 가지 변형을 결합할 수 있다. 예를 들어, 도 1의 불휘발성 메모리(130)의 4개의 큰 부분들은 각각 연속적인 블록으로 식별될 수 있고, 각각의 부분들은 도 1의 NVDIMM(105)에 의해 노출될 수 있는 논리 세그먼트들로 분리될 수 있다.
- [0066] 도 9 및 도 10을 참조하면, 하나의 실시 예가 도시되었다. 하지만, 당업자는 상술한 단계의 순서를 변경하거나, 일부 단계를 생략하거나, 도면에서 도시되지 않은 연결을 포함하여 다른 실시 예들이 구현될 수 있음은 쉽게 이해할 수 있을 것이다. 명시적으로 설명되었는지 여부와 관계 없이, 이러한 순서도의 변형들은 본 발명의 실시 예로서 간주될 수 있다.
- [0067] 이하에서 본 발명의 특정 양상들이 적용될 수 있는 적합한 시스템 또는 시스템들에 대한 간략하고 일반적인 설명이 제공될 것이다. 시스템은 적어도 부분적으로 기존의 입력 장치들로부터의 입력에 의해 제어될 수 있다. 예를 들어, 기존의 입력 장치들은 키보드들, 마우스들을 포함할 수 있다. 혹은, 이러한 제어는 다른 머신으로부터 수신된 지침들, VR(Virtual Reality) 환경과의 상호 작용, 생체 인식 피드백, 가상 머신들, 또는 다른 입력 신호들에 의해 수행될 수 있다.

- [0068] 여기서 사용된 '시스템'이라는 용어는 광범위하게 단일 머신 혹은 시스템, 가상 머신, 또는 통신 결합 머신들의 시스템, 가상 머신들, 또는 함께 동작하는 장치들을 포함할 수 있다. 예시적인 시스템들은 퍼스널(Personal) 컴퓨터들, 워크스테이션들, 서버들, 포터블(Portable) 컴퓨터들, 포켓용(Handheld) 장치들, 휴대전화들, 태블릿들 등을 포함할 수 있다. 혹은 예시적인 시스템들은 자동차들, 기차들, 택시들과 같은 개인용 또는 대중 교통 등과 같은 교통 장치들을 포함할 수 있다.
- [0069] 시스템은 내장 컨트롤러들 등을 포함할 수 있다. 예를 들어, 내장 컨트롤러들은 프로그램이 가능하거나 혹은 불가능한 로직 장치들 또는 어레이(Array), ASIC(Application Specific Integrated Circuits), 내장 컴퓨터들, 스마트 카드들 등을 포함할 수 있다. 시스템은 하나 또는 하나 이상의 원격 머신과의 하나 또는 하나 이상의 연결을 사용할 수 있다. 예를 들어, 이러한 연결은 네트워크 인터페이스, 모뎀, 또는 다른 의사 전달 연결을 통해 이루어질 수 있다.
- [0070] 시스템은 인트라넷, 인터넷, LAN(Local Area Network), WAN(Wide Area Network) 등과 같은 물리적 및/또는 논리적 네트워크 방법에 의해 상호 연결될 수 있다. 당업자는 네트워크 통신이 다양한 유선 및/또는 무선 근거리 또는 장거리 캐리어 및 프로토콜을 이용할 수 있다는 것을 쉽게 이해할 수 있을 것이다. 예를 들어, 캐리어 및 프로토콜은 RF(Radio Frequency), 위성(Satellite), 마이크로 웨이브(Microwave), IEEE(Institute of Electrical and Electronics Engineers) 802.11, Bluetooth®, 광학, 적외선, 케이블, 레이저 등을 포함할 수 있다.
- [0071] 본 발명의 실시 예들은 기능들, 단계들, 데이터 구조들, 어플리케이션 프로그램들을 포함하는 관련 데이터를 참조하여 설명될 수 있다. 기능들, 단계들, 데이터 구조들, 어플리케이션 프로그램들은 시스템에 의해 액세스되는 경우에 시스템이 테스크를 수행하거나 추상적인 데이터 타입들 또는 로우 레벨(Low-Level)의 하드웨어 컨텍스트(Context)들을 정의하도록 한다. 예를 들어, 상술한 관련 데이터는 RAM, ROM과 같은 휘발성 및/또는 불휘발성 메모리에 저장될 수 있다. 또는, 관련 데이터는 다른 스토리지 장치들 및 그들의 관련 스토리지 매체에 저장될 수 있다. 예를 들어, 관련 스토리지 매체는 하드 드라이브들, 플로피 디스크(Floppy-Disks), 광학 스토리지(Optical Storage), 테이프(Tapes), 플래시 메모리(Flash Memory), 메모리 스틱(Memory Sticks), 디지털 비디오 디스크(Digital Video Disks), 생체 스토리지(Biological Storage) 등을 포함할 수 있다.
- [0072] 관련 데이터는 패킷, 시리얼(Serial) 데이터, 패러렐(Parallel) 데이터, 전파 신호 등의 형태로 물리적 및/또는 논리적 네트워크를 포함하는 통신 환경을 통해 전송될 수 있다. 또한, 관련 데이터는 압축되거나 암호화된 형태로 이용될 수 있다. 관련 데이터는 분산 환경에서 이용될 수 있고, 시스템 액세스에 대해 가까이 및/또는 멀리 저장될 수 있다.
- [0073] 본 발명의 실시 예들은 유형의 논 트랜지토리(Non-transitory)의 시스템 리더블(Readable) 매체를 포함할 수 있다. 시스템 리더블 매체는 하나 또는 하나 이상의 프로세서에 의해 수행되는 명령, 여기서 기술된 본 발명의 요소들을 수행하는 명령들로 구성된 명령들을 포함할 수 있다.
- [0074] 본 발명의 실시 예들은 제한 없이 다음의 예시들에 확장될 수 있다.
- [0075] 제 1 예에 따른 본 발명의 실시 예는 장치를 포함할 수 있다.
- [0076] 상기 장치는 DIMM 소켓에 장착된 NVDIMM; 그리고
- [0077] 호스트 프로세서 상에서 동작하는 장치 드라이버를 포함하되,
- [0078] NVDIMM은 메모리 및 노출 메모리를 포함하고, 노출 메모리는 노출 메모리의 제 1 사이즈 및 노출 메모리의 기본 주소를 포함하고,
- [0079] 장치 드라이버는 호스트 메모리 컨트롤러로 전송되는 메모리 주소를 인터셉트하고, 메모리 주소를 사전 매핑된 메모리 주소로 교체하도록 동작하고,
- [0080] 사전 매핑된 메모리 주소는 메모리 주소와 다른 장치이다.
- [0081] 제 2 예는 제 1 예에 따른 장치를 포함한다. NVDIMM은 SSD(Solid State Device)를 포함한다.
- [0082] 제 3 예는 제 1 예에 따른 장치를 포함한다. 사전 매핑된 메모리 주소는 호스트 메모리 컨트롤러가 타겟 메모리 주소를 NVDIMM에 전송하도록 설계되고, 타겟 메모리 주소는 메모리 주소에 저장된 값의 물리적 위치를 나타낸다.

- [0083] 제 4 예는 제 3 예에 따른 장치를 포함한다. 장치 드라이버는 NVDIMM 내의 노출 메모리의 제 1 사이즈, NVDIMM 내의 노출 메모리의 기본 주소, NVDIMM 내의 노출 메모리의 논리 세그먼트의 제 2 사이즈, 그리고 호스트 메모리 컨트롤러의 메모리 컨트롤러 동작 모드에 기초하여 메모리 주소로부터 사전 매핑된 메모리 주소를 생성하도록 동작한다.
- [0084] 제 5 예는 제 4 예에 따른 장치를 포함한다. 장치 드라이버는 호스트 메모리 컨트롤러에 의한 메모리 주소의 논리-물리 변환에 기초하여 메모리 주소로부터 사전 매핑된 메모리 주소를 생성하도록 동작한다.
- [0085] 제 6 예는 제 3 예에 따른 장치를 포함한다. 타겟 메모리 주소는 메모리 주소이다.
- [0086] 제 7 예는 제 3 예에 따른 장치를 포함한다. 타겟 메모리 주소는 메모리 주소의 에일리어스(Alias)이다.
- [0087] 제 8 예는 제 7 예에 따른 장치를 포함한다. 사전 매핑된 메모리 주소는 메모리 주소에 대한 적어도 하나의 변경된 비트를 포함한다. 적어도 하나의 변경된 비트는 NVDIMM에 의해 사용되지 않는다.
- [0088] 제 9 예는 제 3 예에 따른 장치를 포함한다. NVDIMM의 노출 메모리는 NVDIMM의 메모리 내의 불 연속적인 논리 세그먼트들로 분할된다.
- [0089] 제 10 예는 제 3 예에 따른 장치를 포함한다. 노출 메모리는 메모리의 집합이다.
- [0090] 제 11 예에 따른 본 발명의 실시 예는 방법을 포함할 수 있다. 방법은
- [0091] NVDIMM의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계;
- [0092] 메모리 주소를 중간 주소로 사전 매핑하는 단계;
- [0093] 중간 주소를 호스트 메모리 컨트롤러로 전송하는 단계를 포함하되,
- [0094] NVDIMM은 메모리 및 노출 메모리를 포함하고, 노출 메모리는 노출 메모리의 제 1 사이즈 및 노출 메모리의 기본 주소를 포함하고,
- [0095] 중간 주소는 메모리 주소와 다르고,
- [0096] 호스트 메모리 컨트롤러는 타겟 메모리 주소를 NVDIMM에 전송하고, 타겟 메모리 주소는 메모리 주소에 저장된 값의 물리적 위치를 의미하는 방법이다.
- [0097] 제 12 예는 제 11예에 따른 방법을 포함한다. NVDIMM의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계는 호스트 메모리 컨트롤러에 요청이 도착하기 전에, 메모리 주소에 액세스하기 위한 요청을 호스트 프로세서에서 호스트 메모리 컨트롤러로 인터셉트하는 단계를 포함한다.
- [0098] 제 13 예는 제 11예에 따른 방법을 포함한다. NVDIMM의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계는 SSD의 메모리 주소에 액세스하기 위한 요청을 수신하는 단계를 포함한다.
- [0099] 제 14 예는 제 11 예에 따른 방법을 포함한다. 메모리 주소를 중간 주소로 사전 매핑하는 단계는 NVDIMM 내의 노출 메모리의 제 1 사이즈, NVDIMM 내의 노출 메모리의 기본 주소, NVDIMM 내의 노출 메모리의 논리 세그먼트의 제 2 사이즈, 그리고 호스트 메모리 컨트롤러의 메모리 컨트롤러 동작 모드에 기초하여 메모리 주소를 중간 주소로 사전 매핑하는 단계를 포함한다.
- [0100] 제 15 예는 제 14 예에 따른 방법을 포함한다. 메모리 주소를 중간 주소로 사전 매핑하는 단계는 NVDIMM 내의 노출 메모리의 제 1 사이즈, NVDIMM 내의 노출 메모리의 기본 주소, NVDIMM 내의 노출 메모리의 논리 세그먼트의 제 2 사이즈, 호스트 메모리 컨트롤러의 메모리 컨트롤러 동작 모드, 그리고 호스트 메모리 컨트롤러에 의해 사용되는 논리-물리 변환에 기초하여 메모리 주소를 중간 주소로 사전 매핑하는 단계를 포함한다.
- [0101] 제 16 예는 제 11 예에 따른 방법을 포함한다. 호스트 메모리 컨트롤러는 NVDIMM에 메모리 주소를 전송한다.
- [0102] 제 17 예는 제 11 예에 따른 방법을 포함한다. 메모리 주소를 중간 주소로 사전 매핑하는 단계는 메모리 주소의 에일리어스를 메모리 주소로 사전 매핑하는 단계를 포함한다.
- [0103] 제 18 예는 제 11 예에 따른 방법을 포함한다. 메모리 주소를 중간 주소로 사전 매핑하는 단계는 메모리 주소에 대해 중간 주소 내의 적어도 하나의 비트를 변경하는 단계를 포함한다. 적어도 하나의 변경된 비트는 NVDIMM에 의해 사용되지 않는다.
- [0104] 제 19 예는 제 11 예에 따른 방법을 포함한다. 방법은 호스트 메모리 컨트롤러에 노출 메모리를 노출하는 단계

를 더 포함한다.

- [0105] 제 20 예는 제 19 예에 따른 방법을 포함한다. 호스트 메모리 컨트롤러에 노출 메모리를 노출하는 단계는,
- [0106] 제 1 사이즈를 갖는 메모리의 연속된 블록을 확인하는 단계;
- [0107] 메모리의 연속된 블록을 논리 세그먼트들로 분할하는 단계; 그리고
- [0108] 논리 세그먼트들을 호스트 메모리 컨트롤러에 노출하는 단계를 포함하되,
- [0109] 논리 세그먼트 각각은 제 2 사이즈를 갖는다.
- [0110] 제 21 예는 제 19 예에 따른 방법을 포함한다. 호스트 메모리 컨트롤러에 노출 메모리를 노출하는 단계는,
- [0111] 메모리의 불 연속적인 블록의 세트를 찾는 단계; 그리고
- [0112] 메모리의 불 연속적인 블록들을 호스트 메모리 컨트롤러에 논리 세그먼트들로서 노출하는 단계를 포함한다.
- [0113] 제 22 예에 따른 본 발명의 실시 예는 유형의 스토리지 매체를 포함하는 장치를 포함한다. 유형의 스토리지 매체는 시스템에 의해 수행되는 경우 그곳에 저장된 비 일시적인 명령을 갖는다. 비 일시적인 명령은 시스템에 의해 실행될 때,
- [0114] NVDIMM의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계;
- [0115] 메모리 주소를 중간 주소로 사전 매핑하는 단계;
- [0116] 중간 주소를 호스트 메모리 컨트롤러로 전송하는 단계를 수행하되,
- [0117] NVDIMM은 메모리 및 노출 메모리를 포함하고, 노출 메모리는 노출 메모리의 제 1 사이즈 및 노출 메모리의 기본 주소를 포함하고,
- [0118] 중간 주소는 메모리 주소와 다르고,
- [0119] 호스트 메모리 컨트롤러는 타겟 메모리 주소를 NVDIMM에 전송하고, 타겟 메모리 주소는 메모리 주소에 저장된 값의 물리적 위치를 의미하는 한다.
- [0120] 제 23 예는 제 22예에 따른 장치를 포함한다. NVDIMM의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계는 호스트 메모리 컨트롤러에 요청이 도착하기 전에, 메모리 주소에 액세스하기 위한 요청을 호스트 프로세서에서 호스트 메모리 컨트롤러로 인터셉트하는 단계를 포함한다.
- [0121] 제 24 예는 제 22 예에 따른 장치를 포함한다. NVDIMM의 메모리 주소에 대해 액세스하기 위한 요청을 수신하는 단계는 SSD의 메모리 주소에 액세스하기 위한 요청을 수신하는 단계를 포함한다.
- [0122] 제 25 예는 제 22 예에 따른 장치를 포함한다. 메모리 주소를 중간 주소로 사전 매핑하는 단계는 NVDIMM 내의 노출 메모리의 제 1 사이즈, NVDIMM 내의 노출 메모리의 기본 주소, NVDIMM 내의 노출 메모리의 논리 세그먼트의 제 2 사이즈, 그리고 호스트 메모리 컨트롤러의 메모리 컨트롤러 동작 모드에 기초하여 메모리 주소를 중간 주소로 사전 매핑하는 단계를 포함한다.
- [0123] 제 26 예는 제 25 예에 따른 장치를 포함한다. 메모리 주소를 중간 주소로 사전 매핑하는 단계는 NVDIMM 내의 노출 메모리의 제 1 사이즈, NVDIMM 내의 노출 메모리의 기본 주소, NVDIMM 내의 노출 메모리의 논리 세그먼트의 제 2 사이즈, 호스트 메모리 컨트롤러의 메모리 컨트롤러 동작 모드, 그리고 호스트 메모리 컨트롤러에 의해 사용되는 논리-물리 변환에 기초하여 메모리 주소를 중간 주소로 사전 매핑하는 단계를 포함한다.
- [0124] 제 27 예는 제 22 예에 따른 장치를 포함한다. 호스트 메모리 컨트롤러는 NVDIMM에 메모리 주소를 전송한다.
- [0125] 제 28 예는 제 22 예에 따른 장치를 포함한다. 메모리 주소를 중간 주소로 사전 매핑하는 단계는 메모리 주소의 에일리어스를 메모리 주소로 사전 매핑하는 단계를 포함한다.
- [0126] 제 29 예는 제 22 예에 따른 장치를 포함한다. 메모리 주소를 중간 주소로 사전 매핑하는 단계는 메모리 주소에 대해 중간 주소 내의 적어도 하나의 비트를 변경하는 단계를 포함한다. 적어도 하나의 변경된 비트는 NVDIMM에 의해 사용되지 않는다.
- [0127] 제 30 예는 제 22 예에 따른 장치를 포함한다. 비 일시적인 명령은 호스트 메모리 컨트롤러에 노출 메모리를 노출하는 단계를 더 수행한다.

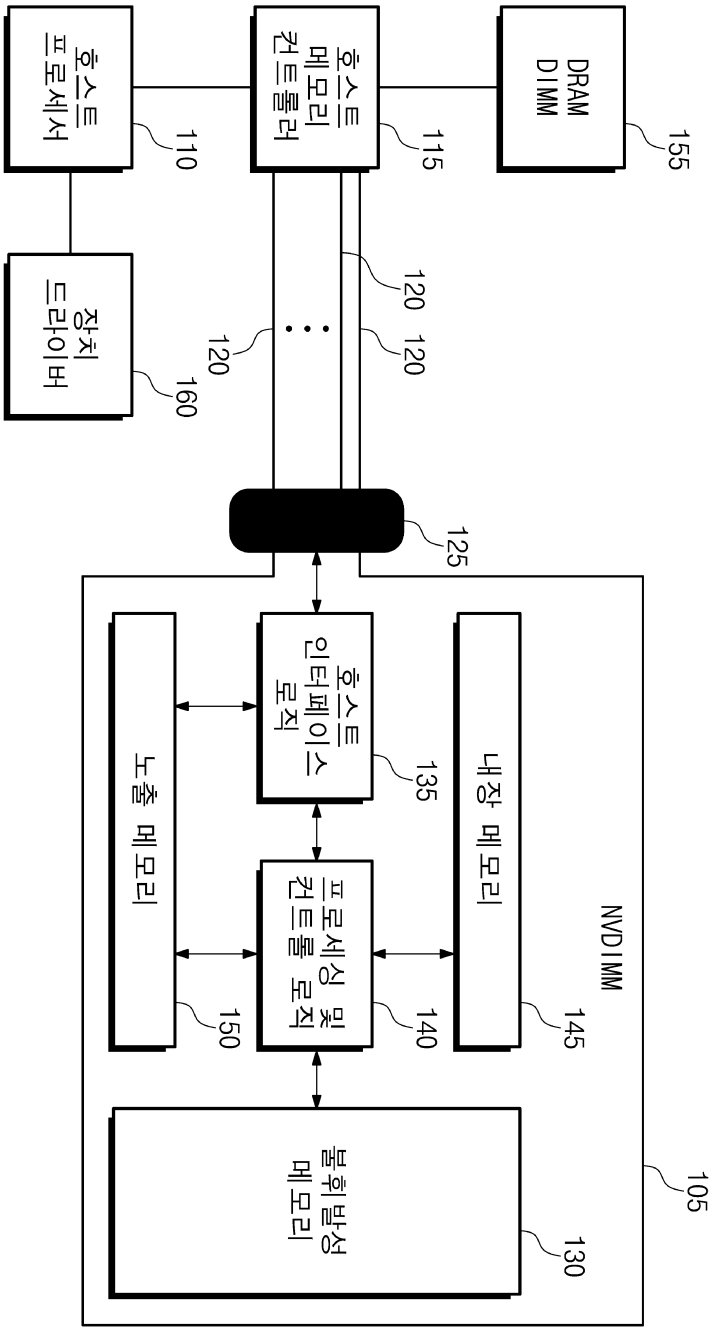
- [0128] 제 31 예는 제 30 예에 따른 장치를 포함한다. 호스트 메모리 컨트롤러에 노출 메모리를 노출하는 단계는,
- [0129] 제 1 사이즈를 갖는 메모리의 연속된 블록을 확인하는 단계;
- [0130] 메모리의 연속된 블록을 논리 세그먼트들로 분할하는 단계; 그리고
- [0131] 논리 세그먼트들을 호스트 메모리 컨트롤러에 노출하는 단계를 포함하되,
- [0132] 논리 세그먼트 각각은 제 2 사이즈를 갖는다.
- [0133] 제 32 예는 제 30 예에 따른 장치를 포함한다. 호스트 메모리 컨트롤러에 노출 메모리를 노출하는 단계는,
- [0134] 메모리의 불 연속적인 블록의 세트를 찾는 단계; 그리고
- [0135] 메모리의 불 연속적인 블록들을 호스트 메모리 컨트롤러에 논리 세그먼트들로서 노출하는 단계를 포함한다.
- [0136] 이상에서와 같이 도면과 명세서에서 최적 실시 예가 개시되었다. 여기서 특정한 용어들이 사용되었으나, 이는 단지 본 발명을 설명하기 위한 목적에서 사용된 것이지 의미 한정이나 특허청구범위에 기재된 본 발명의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 특허청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

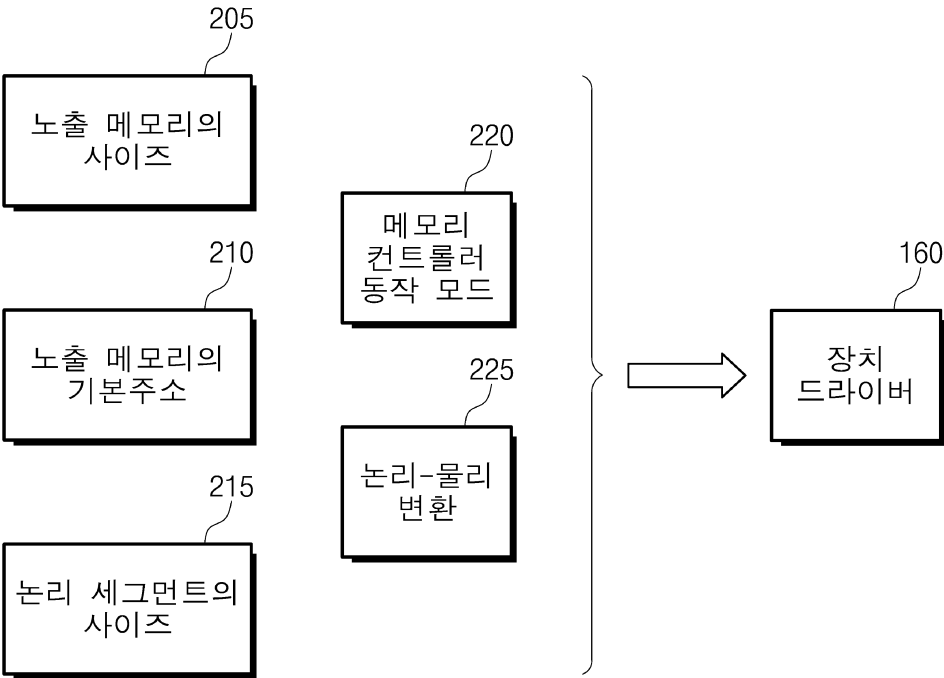
- | | | |
|--------|--------------------|---------------------|
| [0137] | 105 : NVDIMM | 110 : 호스트 프로세서 |
| | 115 : 호스트 메모리 컨트롤러 | 120 : 메모리 신호 라인 |
| | 125 : DIMM 소켓 | 130 : 불휘발성 메모리 |
| | 135 : 호스트 인터페이스 로직 | 140 : 프로세싱 및 컨트롤 로직 |
| | 145 : 내장 DRAM | 150 : 노출 메모리 |
| | 155 : DRAM DIMM | 160 : 장치 드라이버 |
| | 305 : 컴퓨터 시스템 | 310 : 컴퓨터 |
| | 315 : 모니터 | 320 : 키보드 |
| | 325 : 마우스 | 410 : 스토리지 장치 |
| | 415 : 네트워크 커넥터 | 420 : 버스 |
| | 425 : 유저 인터페이스 | 430 : I/O 엔진 |

도면

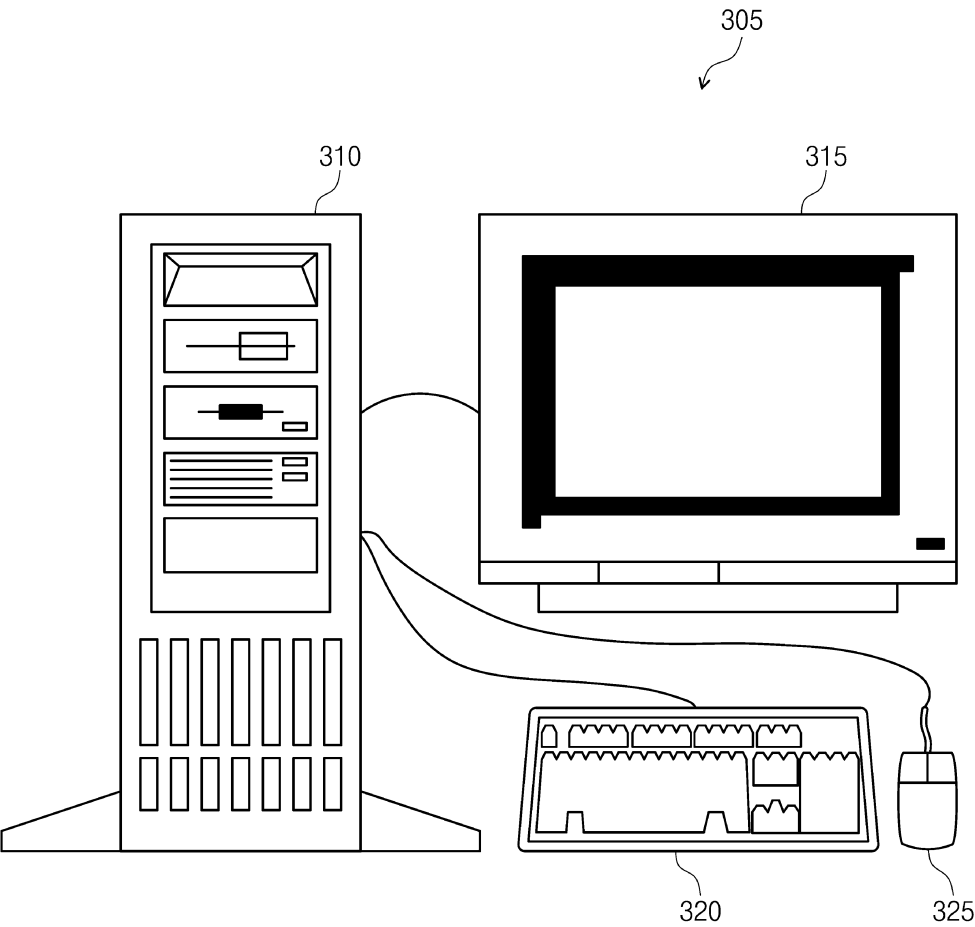
도면1



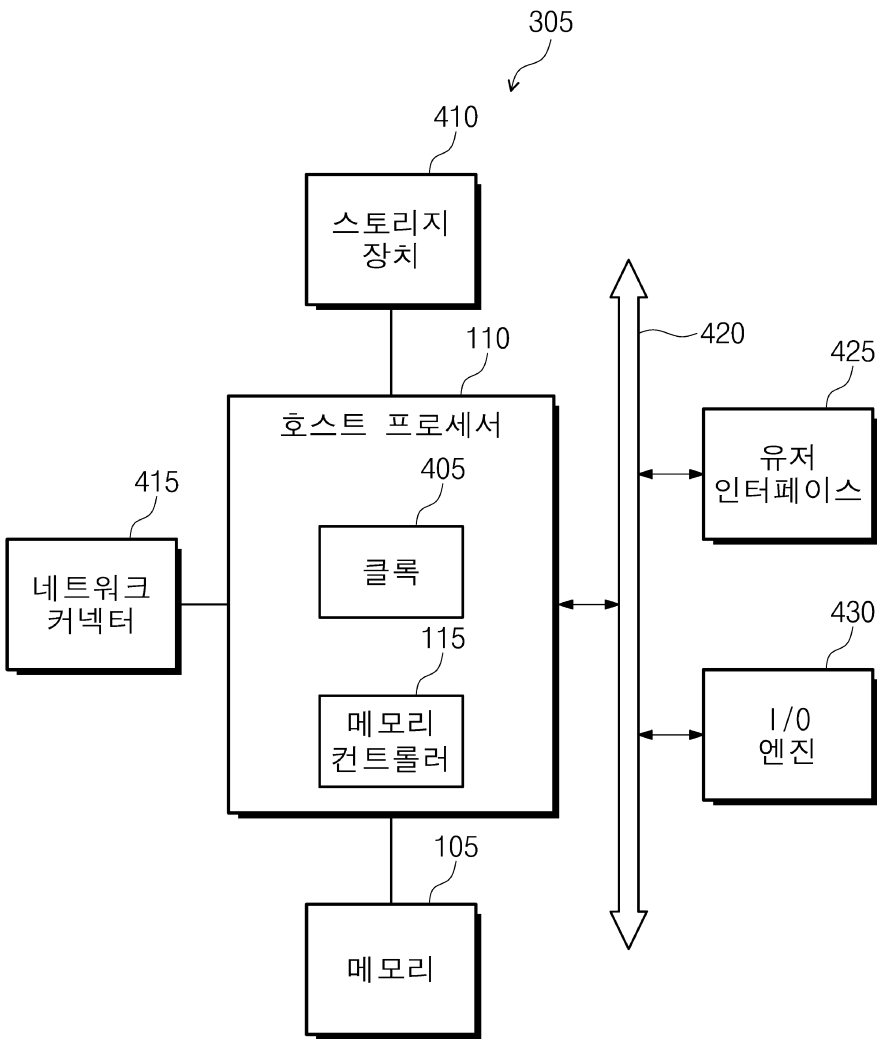
도면2



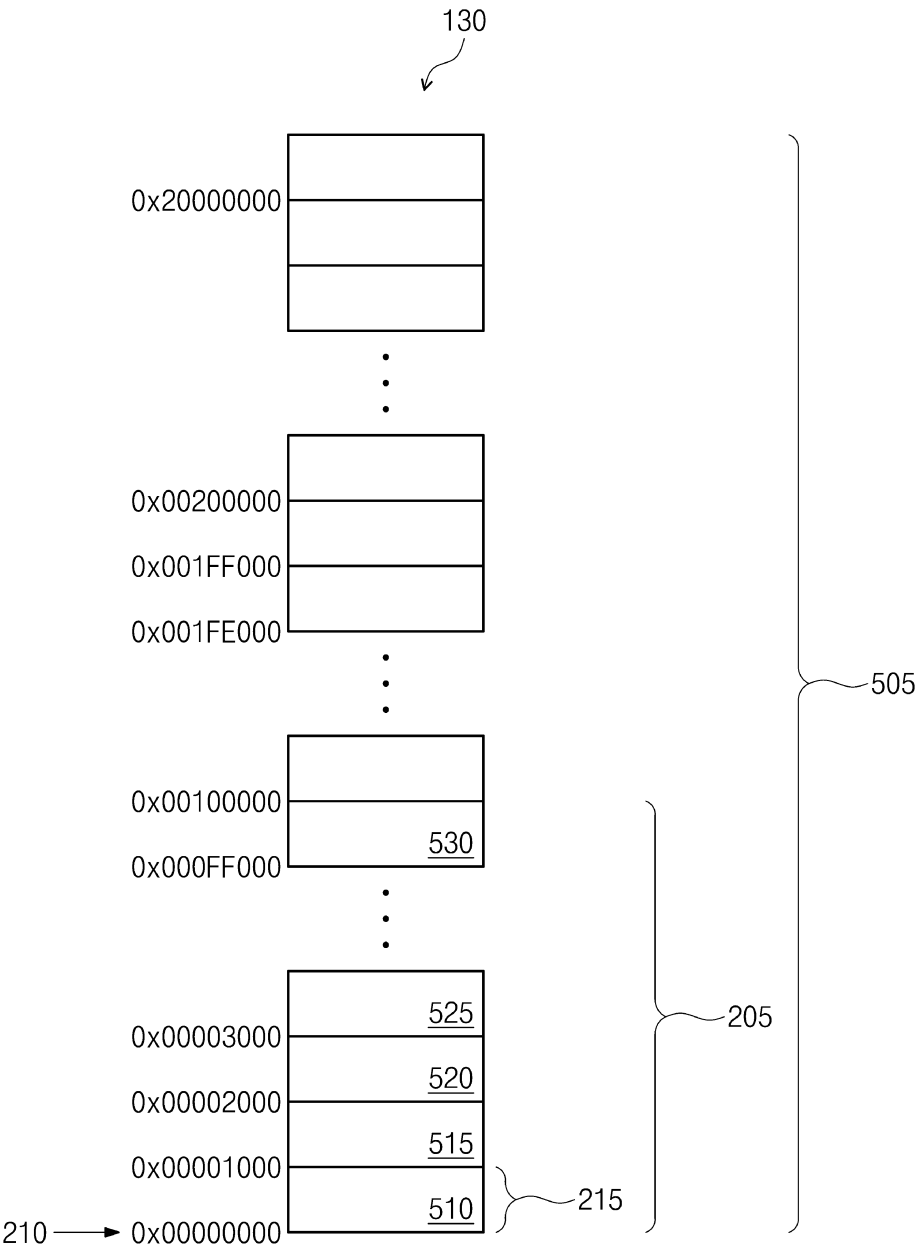
도면3



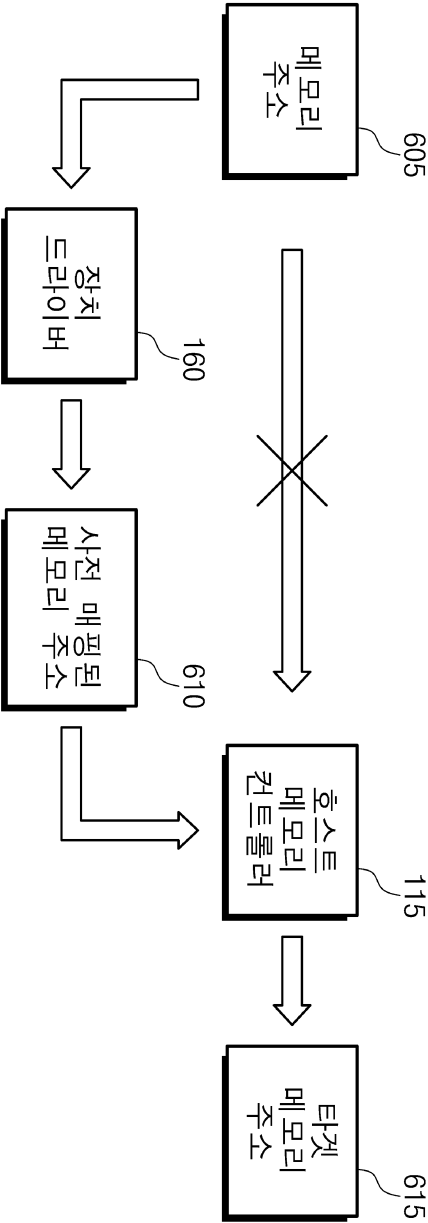
도면4



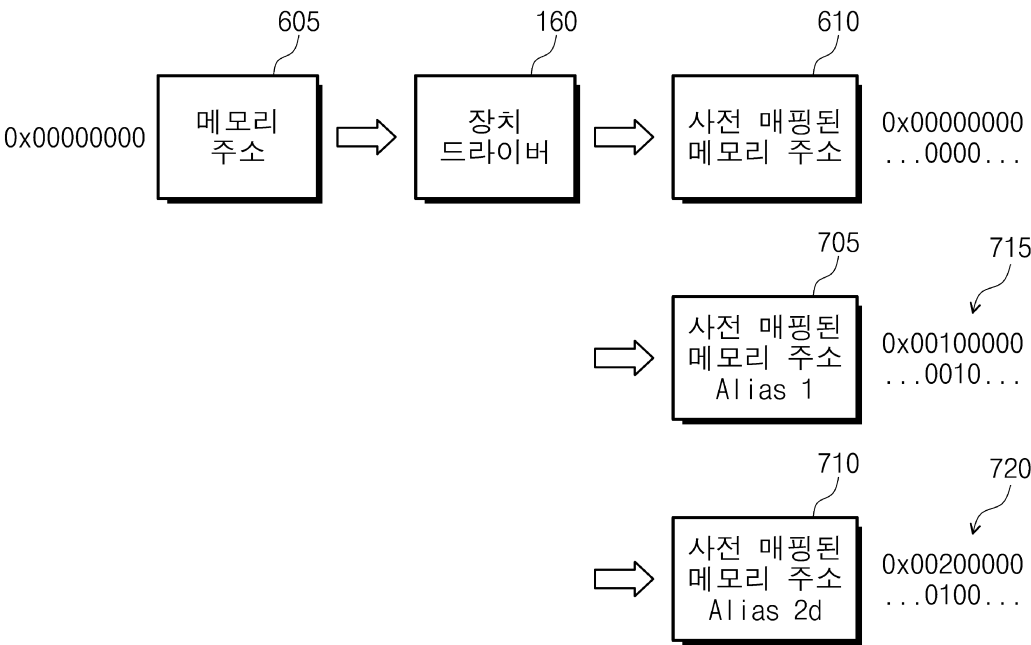
도면5



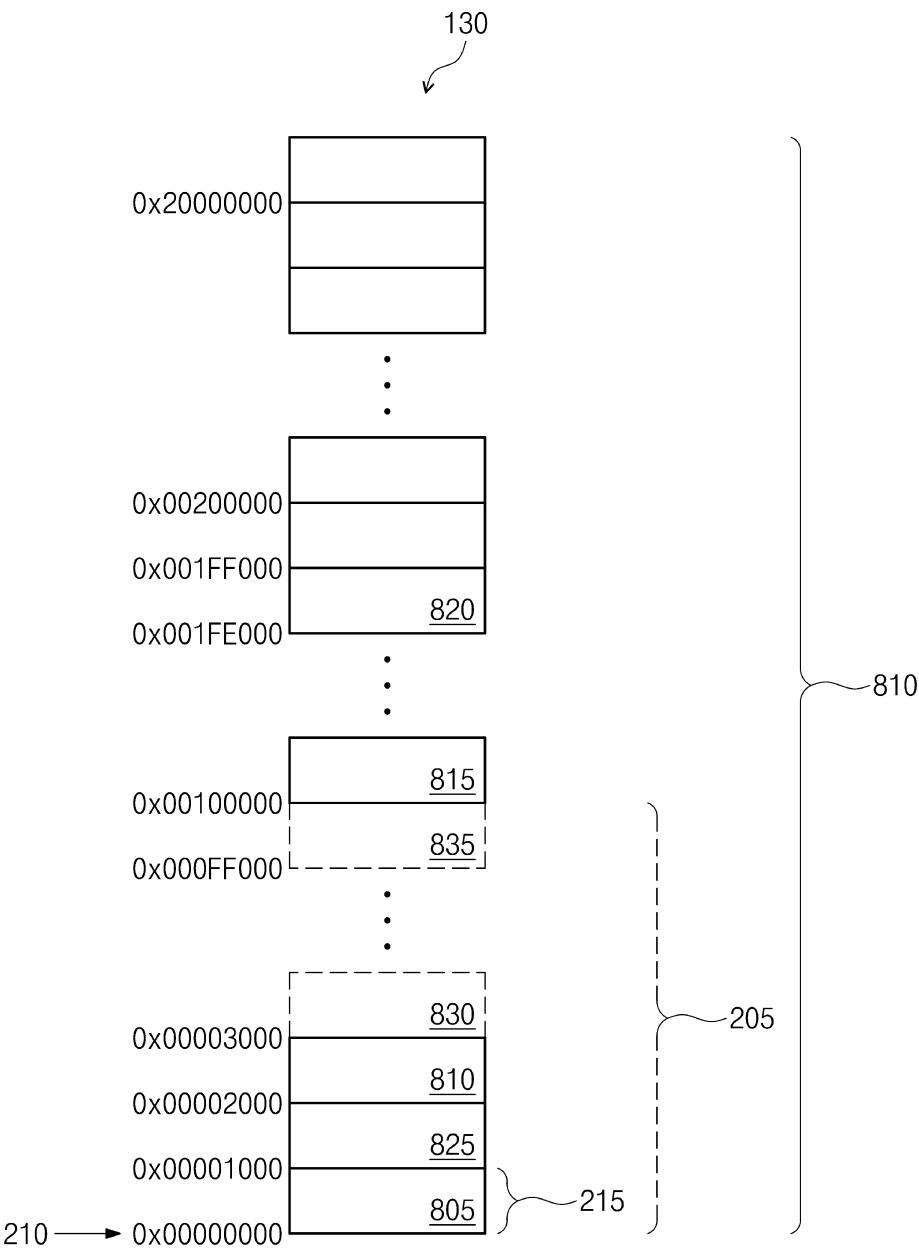
도면6



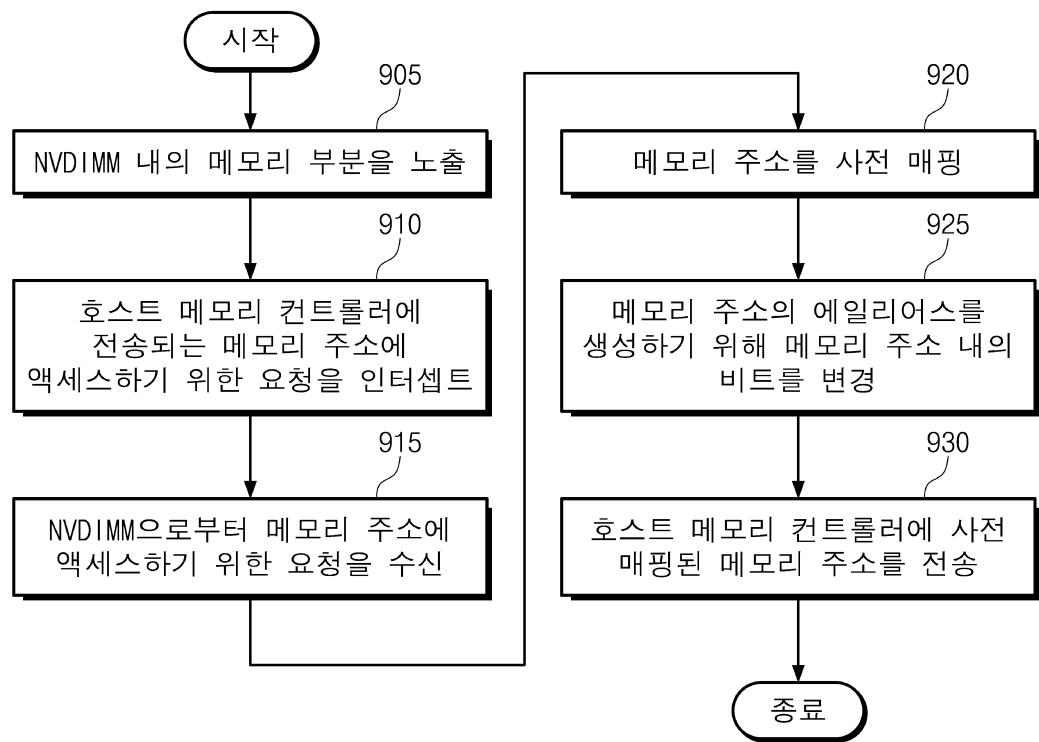
도면7



도면8



도면9



도면10

