



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년08월29일
(11) 등록번호 10-2571629
(24) 등록일자 2023년08월23일

(51) 국제특허분류(Int. Cl.)
G06F 12/02 (2018.01)

(52) CPC특허분류
G06F 12/0292 (2013.01)
G06F 2212/7201 (2013.01)

(21) 출원번호 10-2018-0058112
(22) 출원일자 2018년05월23일
심사청구일자 2021년04월09일
(65) 공개번호 10-2019-0133331
(43) 공개일자 2019년12월03일

(56) 선행기술조사문헌
KR101760144 B1

(뒷면에 계속)

전체 청구항 수 : 총 1 항

(73) 특허권자
에스케이하이닉스 주식회사
경기도 이천시 부발읍 경충대로 2091

(72) 발명자
박병규
경기도 성남시 분당구 느티로 70, 406동 1401호(정자동, 느티마을3,4단지)

(74) 대리인
신성특허법인(유한)

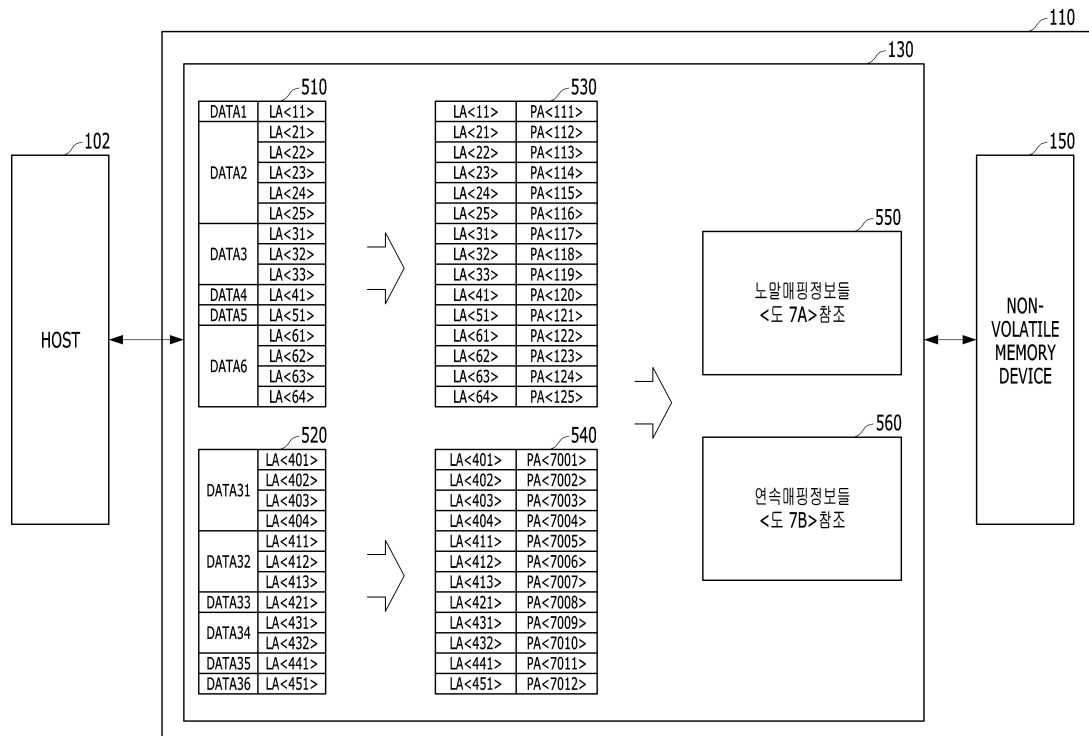
심사관 : 강민정

(54) 발명의 명칭 메모리 시스템 및 메모리 시스템의 동작방법

(57) 요약

본 기술은 비휘발성 메모리 장치를 포함하는 메모리 시스템 및 메모리 시스템의 동작방법에 관한 것으로서, 비휘발성 메모리 장치와, 호스트로부터 다수의 데이터들 및 그에 대응하는 다수의 논리주소들이 인가될 때, 논리주소들 중, N개의 제1 논리주소를 N개의 제1 물리주소에 매핑(mapping)시키기 위한 제1 매핑정보들과, M개의 제2 논리주소를 M개의 제2 물리주소에 매핑(mapping)시키기 위한 제2 매핑정보들을 포함하는 매핑정보를 생성하고, 비휘발성 메모리 장치에 생성된 매핑정보를 저장하는 메모리 시스템을 제공한다. (뒷면에 계속)

대표도



리주소를 M개의 제2 물리주소에 매핑시키기 위한 제2 매핑정보들을 관리하는 컨트롤러를 포함하며, 컨트롤러는, 연속된 값을 갖는 적어도 두 개 이상의 제1 논리주소 및 제1 물리주소를 매핑시키기 위한 제1 매핑정보들을 압축시켜 제1 연속매핑정보들로서 관리하고, 연속된 값을 갖는 적어도 두 개 이상의 제2 논리주소 및 제2 물리주소를 매핑시키기 위한 제2 매핑정보들을 압축시켜 제2 연속매핑정보들로서 관리하며, 제1 연속매핑정보들 각각을 제1 크기의 비트(bit)를 사용하여 관리하고, 제2 연속매핑정보들 각각을 제2 크기의 비트를 사용하여 관리하며, N과 M은 각각 자연수인 메모리 시스템을 포함한다.

(56) 선행기술조사문헌

KR1020140117446 A*

KR1020170109391 A*

US20160179896 A1

US20180018100 A1*

US07190284 B1*

*는 심사관에 의하여 인용된 문헌

명세서

청구범위

청구항 1

복수의 물리주소들에 각각 대응하는 복수의 저장영역들을 포함하는 비휘발성 메모리 장치;

호스트로부터 복수의 데이터들 및 그에 대응하는 복수의 논리주소들이 인가될 때, 상기 논리주소들 중 제1 논리주소들을 상기 복수의 물리주소들 중 제1 물리주소들에 매핑(mapping)시키기 위한 제1 매핑정보들과, 상기 논리주소들 중 제2 논리주소들을 상기 복수의 물리주소들 중 제2 물리주소들에 매핑시키기 위한 제2 매핑정보들을 관리하는 컨트롤러를 포함하며,

상기 컨트롤러는, 연속된 값을 갖는 적어도 두 개 이상의 제1 논리주소들 및 제1 물리주소들을 매핑시키기 위한 상기 제1 매핑정보들을 압축시켜 제1 연속매핑정보들로서 관리하고, 연속된 값을 갖는 적어도 두 개 이상의 제2 논리주소들 및 제2 물리주소들을 매핑시키기 위한 상기 제2 매핑정보들을 압축시켜 제2 연속매핑정보들로서 관리하며, 상기 저장영역들 중 상기 제1물리주소들이 나타내는 저장영역들과 상기 제2물리주소들이 나타내는 저장영역들의 위치 차이에 따라 제1비트(bit)의 크기와 제2비트의 크기를 서로 다르게 설정하고,

상기 제1비트의 크기는, 상기 제1물리주소들 중 가장 큰 값을 카운팅하기 위해 필요한 크기이며,

상기 제2비트의 크기는, 상기 제2물리주소들 중 가장 큰 값을 카운팅하기 위해 필요한 크기인 메모리 시스템.

청구항 2

◆청구항 2은(는) 설정등록료 납부시 포기되었습니다.◆

제1항에 있어서,

상기 제1 연속매핑정보들 각각은,

내부에 압축되어 포함된 상기 제1 논리주소들 중 시작논리주소 및 제1 물리주소들과의 매핑을 위한 제1 인덱스(index) 정보를 포함하는 제1 시작논리정보와,

내부에 압축되어 포함된 상기 제1 논리주소들의 개수에 대한 정보를 포함하는 제1 길이논리정보와,

내부에 압축되어 포함된 적어도 상기 제1 물리주소들 중 시작물리주소 및 제1 논리주소들과의 매핑을 위한 상기 제1 인덱스 정보를 포함하는 제1 시작물리정보, 및

내부에 압축되어 포함된 상기 제1 물리주소들의 개수에 대한 정보를 포함하는 제1 길이물리정보를 포함하는 메모리 시스템.

청구항 3

◆청구항 3은(는) 설정등록료 납부시 포기되었습니다.◆

제2항에 있어서,

상기 제2 연속매핑정보들 각각은,

내부에 압축되어 포함된 상기 제2 논리주소들 중 시작논리주소 및 제2 물리주소들과의 매핑을 위한 제2 인덱스 정보를 포함하는 제2 시작논리정보와,

내부에 압축되어 포함된 상기 제2 논리주소들의 개수에 대한 정보를 포함하는 제2 길이논리정보와,

내부에 압축되어 포함된 상기 제2 물리주소들 중 시작물리주소 및 제2 논리주소들과의 매핑을 위한 상기 제2 인덱스 정보를 포함하는 제2 시작물리정보, 및

내부에 압축되어 포함된 상기 제2 물리주소들의 개수에 대한 정보를 포함하는 제2 길이물리정보를 포함하는 메모리 시스템.

청구항 4

◆청구항 4은(는) 설정등록료 납부시 포기되었습니다.◆

제3항에 있어서,

상기 컨트롤러는,

상기 제1 연속매핑정보들 각각에 포함된 시작물리주소 중 가장 큰 값을 상기 제1 인덱스 정보와 함께 카운팅하기 위해 필요한 비트의 크기를 상기 제1비트의 크기로 결정하고,

상기 제2 연속매핑정보들 각각에 포함된 시작물리주소 중 가장 큰 값을 상기 제2 인덱스 정보와 함께 카운팅하기 위해 필요한 비트의 크기를 상기 제2비트의 크기로 결정하는 메모리 시스템.

청구항 5

◆청구항 5은(는) 설정등록료 납부시 포기되었습니다.◆

제3항에 있어서,

상기 컨트롤러는,

상기 제1 연속매핑정보들 각각에 압축되어 포함된 상기 제1 물리주소들 개수의 평균개수를 제1 평균개수로서 연산하고, 상기 제1 연속매핑정보들 각각에 포함된 상기 제1 물리주소들의 개수가 상기 제1 평균개수 이하가 되도록 상기 제1 연속매핑정보들의 압축률을 재설정하며, 상기 제1 평균개수를 카운팅하기 위해 필요한 비트의 크기를 상기 제1비트의 크기로 결정하고,

상기 제2 연속매핑정보들 각각에 압축되어 포함된 제2 물리주소 개수의 평균개수를 제2 평균개수로서 연산하고, 상기 제2 연속매핑정보들 각각에 포함된 제2 물리주소의 개수가 상기 제2 평균개수 이하가 되도록 상기 제2 연속매핑정보들의 압축률을 재설정하며, 상기 제2 평균개수를 카운팅하기 위해 필요한 비트의 크기를 상기 제2비트의 크기로 결정하는 메모리 시스템.

청구항 6

◆청구항 6은(는) 설정등록료 납부시 포기되었습니다.◆

제3항에 있어서,

상기 컨트롤러는,

상기 제1 연속매핑정보들 각각에 압축되어 포함된 상기 제1 물리주소들 개수의 최대/최소개수를 제1 최대개수 및 제1 최소개수로서 설정하고, 상기 제1 최대개수와 상기 제1 최소개수의 차이를 카운팅하기 위해 필요한 비트의 크기를 상기 제1비트의 크기로 결정하며, 결정된 상기 제1비트의 크기를 상기 제1 최소개수를 기준으로 사용하여 상기 제1 길이물리정보를 관리하고,

상기 제2 연속매핑정보들 각각에 압축되어 포함된 상기 제2 물리주소들 개수의 최대/최소개수를 제2 최대개수 및 제2 최소개수로서 설정하며, 상기 제2 최대개수와 상기 제2 최소개수의 차이를 카운팅하기 위해 필요한 비트의 크기를 상기 제2비트의 크기로 결정하며, 결정된 상기 제2비트의 크기를 상기 제2 최소개수를 기준으로 사용하여 상기 제2 길이물리정보를 관리하는 메모리 시스템.

청구항 7

◆청구항 7은(는) 설정등록료 납부시 포기되었습니다.◆

제3항에 있어서,

상기 컨트롤러는,

상기 제1 연속매핑정보들 각각에 압축되어 포함된 상기 제1 물리주소들 개수의 최대개수를 카운팅하기 위해 필요한 비트의 크기를 상기 제1비트의 크기로 결정하며,

상기 제2 연속매핑정보들 각각에 압축되어 포함된 상기 제2 물리주소들 개수의 최대개수를 카운팅하기 위해 필요한 비트의 크기를 상기 제2비트의 크기로 결정하는 메모리 시스템.

청구항 8

◆청구항 8은(는) 설정등록료 납부시 포기되었습니다.◆

제1항에 있어서,

상기 컨트롤러는,

예정된 시작값으로부터 그 값이 순차적으로 증가하는 상기 물리주소들 각각이 상기 복수의 저장영역들에 각각 대응할 때,

상기 제1물리주소들이 나타내는 저장영역의 위치가 상기 제2물리주소들이 나타내는 저장영역의 위치보다 상기 예정된 시작값이 나타내는 저장영역의 위치보다 가까운 경우, 상기 제1비트의 크기가 상기 제2비트의 크기보다 작은 크기를 갖도록 결정하는 메모리 시스템.

청구항 9

삭제

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 시스템에 관한 것으로서, 구체적으로 비휘발성 메모리 장치를 포함하는 메모리 시스템 및 메모리 시스템의 동작방법에 관한 것이다.

배경 기술

[0002] 최근 컴퓨터 환경에 대한 패러다임(paradigm)이 언제, 어디서나 컴퓨터 시스템을 사용할 수 있도록 하는 유비쿼터스 컴퓨팅(ubiquitous computing)으로 전환되고 있다. 이로 인해 휴대폰, 디지털 카메라, 노트북 컴퓨터 등과 같은 휴대용 전자 장치의 사용이 급증하고 있다. 이와 같은 휴대용 전자 장치는 일반적으로 메모리 장치를 이용하는 메모리 시스템, 다시 말해 데이터 저장 장치를 사용한다. 데이터 저장 장치는 휴대용 전자 장치의 주 기억 장치 또는 보조 기억 장치로 사용된다.

[0003] 메모리 장치를 이용한 데이터 저장 장치는 기계적인 구동부가 없어서 안정성 및 내구성이 뛰어나며, 또한 정보의 액세스 속도가 매우 빠르고 전력 소모가 적다는 장점이 있다. 이러한 장점을 갖는 메모리 시스템의 일 예로 데이터 저장 장치는, USB(Universal Serial Bus) 메모리 장치, 다양한 인터페이스를 갖는 메모리 카드, 솔리드 스테이트 드라이브(SSD: Solid State Drive) 등을 포함한다.

발명의 내용

해결하려는 과제

[0004] 본 발명의 실시예는 호스트에 대응하는 논리주소와 비휘발성 메모리 장치에 대응하는 물리주소를 매핑(mapping)하기 위한 매핑정보를 효율적으로 관리할 수 있는 메모리 시스템 및 메모리 시스템의 동작방법을 제공한다.

과제의 해결 수단

[0005] 본 발명의 실시예에 따른 메모리 시스템은, 비휘발성 메모리 장치; 호스트로부터 다수의 데이터들 및 그에 대응하는 다수의 논리주소들이 인가될 때, 상기 논리주소들 중, N개의 제1 논리주소를 N개의 제1 물리주소에 매핑(mapping)시키기 위한 제1 매핑정보들과, M개의 제2 논리주소를 M개의 제2 물리주소에 매핑시키기 위한 제2 매핑정보들을 관리하는 컨트롤러를 포함하며, 상기 컨트롤러는, 연속된 값을 갖는 적어도 두 개 이상의 제1 논리주소 및 제1 물리주소를 매핑시키기 위한 상기 제1 매핑정보들을 압축시켜 제1 연속매핑정보들로서 관리하고, 연속된 값을 갖는 적어도 두 개 이상의 제2 논리주소 및 제2 물리주소를 매핑시키기 위한 상기 제2 매핑정보들을 압축시켜 제2 연속매핑정보들로서 관리하며, 제1 연속매핑정보들 각각을 제1 크기의 비트(bit)를 사용하여 관리하고, 제2 연속매핑정보들 각각을 제2 크기의 비트를 사용하여 관리하며, N과 M은 각각 자연수일 수 있다.

[0006] 또한, 상기 컨트롤러는, 상기 호스트로부터 인가되는 상기 논리주소들이, 논리주소 값의 설정된 제1 범위에 포함되는 경우, N개의 제1 논리주소로서 판단하고, 논리주소 값의 설정된 제2 범위에 포함되는 경우, M개의 제2 논리주소로서 판단할 수 있다.

[0007] 또한, 상기 컨트롤러는, 연속되지 않은 한 개씩의 제1 논리주소 및 제1 물리주소를 매핑시키기 위한 상기 제1 매핑정보들을 제1 노말매핑정보들로서 관리하고, 연속되지 않은 한 개씩의 제2 논리주소 및 한 개의 제2 물리주소를 매핑시키기 위한 상기 제2 매핑정보들을 제2 노말매핑정보들로서 관리하며, 상기 제1 노말매핑정보들 및

상기 제2 노말매핑정보들 각각을 제3 크기의 비트를 사용하여 관리할 수 있다.

- [0008] 또한, 상기 제1 연속매핑정보들 각각은, 내부에 압축되어 포함된 적어도 두 개 이상의 제1 논리주소 중 시작논리주소 및 제1 물리주소와의 매핑을 위한 제1 인덱스(index) 정보를 포함하는 제1 시작논리정보와, 내부에 압축되어 포함된 제1 논리주소의 개수에 대한 정보를 포함하는 제1 길이논리정보와, 내부에 압축되어 포함된 적어도 두 개 이상의 제1 물리주소 중 시작물리주소 및 제1 논리주소와의 매핑을 위한 상기 제1 인덱스 정보를 포함하는 제1 시작물리정보, 및 내부에 압축되어 포함된 제1 물리주소의 개수에 대한 정보를 포함하는 제1 길이물리정보를 포함할 수 있다.
- [0009] 또한, 상기 제2 연속매핑정보들 각각은, 내부에 압축되어 포함된 적어도 두 개 이상의 제2 논리주소 중 시작논리주소 및 제2 물리주소와의 매핑을 위한 제2 인덱스 정보를 포함하는 제2 시작논리정보와, 내부에 압축되어 포함된 제2 논리주소의 개수에 대한 정보를 포함하는 제2 길이논리정보와, 내부에 압축되어 포함된 적어도 두 개 이상의 제2 물리주소 중 시작물리주소 및 제2 논리주소와의 매핑을 위한 상기 제2 인덱스 정보를 포함하는 제2 시작물리정보, 및 내부에 압축되어 포함된 제2 물리주소의 개수에 대한 정보를 포함하는 제2 길이물리정보를 포함할 수 있다.
- [0010] 또한, 상기 컨트롤러는, 상기 제1 연속매핑정보들 각각에 포함된 시작물리주소 중 가장 큰 값을 상기 제1 인덱스 정보와 함께 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 기준으로 상기 제1 시작물리정보를 관리하며, 상기 제2 연속매핑정보들 각각에 포함된 시작물리주소 중 가장 큰 값을 상기 제2 인덱스 정보와 함께 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 기준으로 상기 제2 시작물리정보를 관리할 수 있다.
- [0011] 또한, 상기 컨트롤러는, 상기 제1 연속매핑정보들 각각에 압축되어 포함된 제1 물리주소 개수의 평균개수를 제1 평균개수로서 연산한 뒤, 상기 제1 연속매핑정보들 각각에 포함된 제1 물리주소의 개수가 상기 제1 평균개수 이하가 되도록 상기 제1 연속매핑정보들의 압축률을 재설정하며, 상기 제1 평균개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 상기 제1 길이물리정보를 관리하며, 상기 제2 연속매핑정보들 각각에 압축되어 포함된 제2 물리주소 개수의 평균개수를 제2 평균개수로서 연산한 뒤, 상기 제2 연속매핑정보들 각각에 포함된 제2 물리주소의 개수가 상기 제2 평균개수 이하가 되도록 상기 제2 연속매핑정보들의 압축률을 재설정하며, 상기 제2 평균개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 상기 제2 길이물리정보를 관리할 수 있다.
- [0012] 또한, 상기 컨트롤러는, 상기 제1 연속매핑정보들 각각에 압축되어 포함된 제1 물리주소 개수의 최대/최소개수를 제1 최대개수 및 제1 최소개수로서 설정한 뒤, 상기 제1 최대개수와 상기 제1 최소개수의 차이를 카운팅하기 위해 필요한 비트크기를 결정하며, 결정된 비트크기를 상기 제1 최소개수를 기준으로 사용하여 상기 제1 길이물리정보를 관리하고, 상기 제1 연속매핑정보들을 설정된 조건에 따라 K개의 그룹으로 분류한 뒤 K개의 그룹 각각에 대해 상기 제1 최소개수를 독립적으로 설정하며, 상기 제2 연속매핑정보들 각각에 압축되어 포함된 제2 물리주소 개수의 최대/최소개수를 제2 최대개수 및 제2 최소개수로서 설정한 뒤, 상기 제2 최대개수와 상기 제2 최소개수의 차이를 카운팅하기 위해 필요한 비트크기를 결정하며, 결정된 비트크기를 상기 제2 최소개수를 기준으로 사용하여 상기 제2 길이물리정보를 관리하고, 상기 제2 연속매핑정보들을 설정된 조건에 따라 L개의 그룹으로 분류한 뒤 L개의 그룹 각각에 대해 상기 제2 최소개수를 독립적으로 설정하며, K와 L은 각각 자연수일 수 있다.
- [0013] 또한, 상기 컨트롤러는, 상기 제1 연속매핑정보들 각각에 압축되어 포함된 제1 물리주소 개수의 최대개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 상기 제1 길이물리정보를 관리하며, 상기 제2 연속매핑정보들 각각에 압축되어 포함된 제2 물리주소 개수의 최대개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 상기 제2 길이물리정보를 관리할 수 있다.
- [0014] 또한, 상기 컨트롤러는, 예정된 시작값으로부터 그 값이 순차적으로 증가하는 다수의 물리주소들 각각이 상기 비휘발성 메모리 장치에 포함된 다수의 저장영역들에 각각 대응할 때, 상기 제1 연속매핑정보들 각각에 포함된 시작물리주소가 상기 제2 연속매핑정보들 각각에 포함된 시작물리주소보다 상기 예정된 시작값으로부터 상대적으로 더 가까운 경우, 상기 제1 시작물리정보가 상기 제2 시작물리정보보다 더 작은 비트크기를 갖도록 관리하며, 상기 제1 연속매핑정보들 각각에 포함된 시작물리주소가 상기 제2 연속매핑정보들 각각에 포함된 시작물리주소보다 상기 예정된 시작값으로부터 상대적으로 더 먼 경우, 상기 제1 시작물리정보가 상기 제2 시작물리정보보다 더 큰 비트크기를 갖도록 관리할 수 있다.

- [0015] 본 발명의 또 다른 실시예에 따른 메모리 시스템의 동작방법은, 비휘발성 메모리 장치를 포함하는 메모리 시스템의 동작방법에 있어서, 호스트로부터 다수의 데이터들 및 그에 대응하는 다수의 논리주소들이 인가되는 인가 단계; 상기 인가단계의 상기 논리주소들 중, N개의 제1 논리주소를 N개의 제1 물리주소에 매핑(mapping)시키기 위한 제1 매핑정보들과, M개의 제2 논리주소를 M개의 제2 물리주소에 매핑시키기 위한 제2 매핑정보들로서 관리하는 매핑단계; 연속된 값을 갖는 적어도 두 개 이상의 제1 논리주소 및 제1 물리주소를 매핑시키기 위한 상기 제1 매핑정보들을 압축시켜 제1 연속매핑정보들로서 관리하되, 상기 제1 연속매핑정보들 각각을 제1 크기의 비트(bit)를 사용하여 관리하는 제1 관리단계; 및 연속된 값을 갖는 적어도 두 개 이상의 제2 논리주소 및 제2 물리주소를 매핑시키기 위한 상기 제2 매핑정보들을 압축시켜 제2 연속매핑정보들로서 관리하되, 상기 제2 연속매핑정보들 각각을 제2 크기의 비트를 사용하여 관리하는 제2 관리단계를 포함하며, N과 M은 각각 자연수일 수 있다.
- [0016] 또한, 상기 인가단계의 상기 논리주소들이, 논리주소 값의 설정된 제1 범위에 포함되는 경우 N개의 제1 논리주소로서 판단하고, 논리주소 값의 설정된 제2 범위에 포함되는 경우 M개의 제2 논리주소로서 판단하는 단계를 더 포함할 수 있다.
- [0017] 또한, 연속되지 않은 한 개씩의 제1 논리주소 및 제1 물리주소를 매핑시키기 위한 상기 제1 매핑정보들을 제1 노말매핑정보들로서 관리하되, 상기 제1 노말매핑정보들 각각을 제3 크기의 비트를 사용하여 관리하는 제3 관리단계; 및 연속되지 않은 한 개씩의 제2 논리주소 및 제2 물리주소를 매핑시키기 위한 상기 제2 매핑정보들을 제2 노말매핑정보들로서 관리하되, 상기 제2 노말매핑정보들 각각을 상기 제3 크기의 비트를 사용하여 관리하는 제4 관리단계를 더 포함할 수 있다.
- [0018] 또한, 상기 제1 연속매핑정보들 각각은, 내부에 압축되어 포함된 적어도 두 개 이상의 제1 논리주소 중 시작논리주소 및 제1 물리주소와의 매핑을 위한 제1 인덱스(index) 정보를 포함하는 제1 시작논리정보와, 내부에 압축되어 포함된 제1 논리주소의 개수에 대한 정보를 포함하는 제1 길이논리정보와, 내부에 압축되어 포함된 적어도 두 개 이상의 제1 물리주소 중 시작물리주소 및 제1 논리주소와의 매핑을 위한 상기 제1 인덱스 정보를 포함하는 제1 시작물리정보, 및 내부에 압축되어 포함된 제1 물리주소의 개수에 대한 정보를 포함하는 제1 길이물리정보를 포함할 수 있다.
- [0019] 또한, 상기 제2 연속매핑정보들 각각은, 내부에 압축되어 포함된 적어도 두 개 이상의 제2 논리주소 중 시작논리주소 및 제2 물리주소와의 매핑을 위한 제2 인덱스 정보를 포함하는 제2 시작논리정보와, 내부에 압축되어 포함된 제2 논리주소의 개수에 대한 정보를 포함하는 제2 길이논리정보와, 내부에 압축되어 포함된 적어도 두 개 이상의 제2 물리주소 중 시작물리주소 및 제2 논리주소와의 매핑을 위한 상기 제2 인덱스 정보를 포함하는 제2 시작물리정보, 및 내부에 압축되어 포함된 제2 물리주소의 개수에 대한 정보를 포함하는 제2 길이물리정보를 포함할 수 있다.
- [0020] 또한, 상기 제1 관리단계는, 상기 제1 연속매핑정보들 각각에 포함된 시작물리주소 중 가장 큰 값을 상기 제1 인덱스 정보와 함께 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 기준으로 상기 제1 시작물리정보를 관리하며, 상기 제2 관리단계는, 상기 제2 연속매핑정보들 각각에 포함된 시작물리주소 중 가장 큰 값을 상기 제2 인덱스 정보와 함께 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 기준으로 상기 제2 시작물리정보를 관리할 수 있다.
- [0021] 또한, 상기 제1 관리단계는, 상기 제1 연속매핑정보들 각각에 압축되어 포함된 제1 물리주소 개수의 평균개수를 제1 평균개수로서 연산한 뒤, 상기 제1 연속매핑정보들 각각에 포함된 제1 물리주소의 개수가 상기 제1 평균개수 이하가 되도록 상기 제1 연속매핑정보들의 압축률을 재설정하며, 상기 제1 평균개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 상기 제1 길이물리정보를 관리하며, 상기 제2 관리단계는, 상기 제2 연속매핑정보들 각각에 압축되어 포함된 제2 물리주소 개수의 평균개수를 제2 평균개수로서 연산한 뒤, 상기 제2 연속매핑정보들 각각에 포함된 제2 물리주소의 개수가 상기 제2 평균개수 이하가 되도록 상기 제2 연속매핑정보들의 압축률을 재설정하며, 상기 제2 평균개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 상기 제2 길이물리정보를 관리할 수 있다.
- [0022] 또한, 상기 제1 관리단계는, 상기 제1 연속매핑정보들 각각에 압축되어 포함된 제1 물리주소 개수의 최대/최소 개수를 제1 최대개수 및 제1 최소개수로서 설정한 뒤, 상기 제1 최대개수와 상기 제1 최소개수의 차이를 카운팅하기 위해 필요한 비트크기를 결정하며, 결정된 비트크기를 상기 제1 최소개수를 기준으로 사용하여 상기 제1 길이물리정보를 관리하고, 상기 제1 연속매핑정보들을 설정된 조건에 따라 K개의 그룹으로 분류한 뒤 K개의 그룹 각각에 대해 상기 제1 최소개수를 독립적으로 설정하며, 상기 제2 관리단계는, 상기 제2 연속매핑정보들 각

각에 압축되어 포함된 제2 물리주소 개수의 최대/최소개수를 제2 최대개수 및 제2 최소개수로서 설정한 뒤, 상기 제2 최대개수와 상기 제2 최소개수의 차이를 카운팅하기 위해 필요한 비트크기를 결정하며, 결정된 비트크기를 상기 제2 최소개수를 기준으로 사용하여 상기 제2 길이물리정보를 관리하고, 상기 제2 연속매핑정보들을 설정된 조건에 따라 L개의 그룹으로 분류한 뒤 L개의 그룹 각각에 대해 상기 제2 최소개수를 독립적으로 설정하며, K와 L은 각각 자연수일 수 있다.

[0023] 또한, 상기 제1 관리단계는, 상기 제1 연속매핑정보들 각각에 압축되어 포함된 제1 물리주소 개수의 최대개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 상기 제1 길이물리정보를 관리하며, 상기 제2 관리단계는, 상기 제2 연속매핑정보들 각각에 압축되어 포함된 제2 물리주소 개수의 최대개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 상기 제2 길이물리정보를 관리할 수 있다.

[0024] 또한, 예정된 시작값으로부터 그 값이 순차적으로 증가하는 다수의 물리주소들 각각이 상기 비휘발성 메모리 장치에 포함된 다수의 저장영역들에 각각 대응하는 경우, 상기 제1 연속매핑정보들 각각에 포함된 시작물리주소가 상기 제2 연속매핑정보들 각각에 포함된 시작물리주소보다 상기 예정된 시작값으로부터 상대적으로 더 가까운 경우, 상기 제1 시작물리정보가 상기 제2 시작물리정보보다 더 작은 비트크기를 갖도록 관리하는 단계; 및 상기 제1 연속매핑정보들 각각에 포함된 시작물리주소가 상기 제2 연속매핑정보들 각각에 포함된 시작물리주소보다 상기 예정된 시작값으로부터 상대적으로 더 먼 경우, 상기 제1 시작물리정보가 상기 제2 시작물리정보보다 더 큰 비트크기를 갖도록 관리하는 단계를 더 포함할 수 있다.

발명의 효과

[0025] 본 기술은 호스트에 대응하는 논리주소와 비휘발성 메모리 장치에 대응하는 물리주소를 매핑(mapping)하기 위한 매핑정보를 관리할 때, 매핑정보의 종류에 따라 크기를 가변하여 사용할 수 있다.

[0026] 이로 인해, 매핑정보가 차지하는 공간을 최소화할 수 있는 효과가 있다.

도면의 간단한 설명

[0027] 도 1은 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 일 예를 개략적으로 도시한 도면.

도 2는 본 발명의 실시 예에 따른 메모리 시스템에서 메모리 장치의 일 예를 개략적으로 도시한 도면.

도 3은 본 발명의 실시 예에 따른 메모리 장치에서 메모리 블록들의 메모리 셀 어레이 회로를 개략적으로 도시한 도면.

도 4는 본 발명의 실시 예에 따른 메모리 시스템에서 메모리 장치 구조를 개략적으로 도시한 도면.

도 5는 본 발명의 실시 예에 따른 메모리 시스템에서 메모리 장치에 데이터 처리 동작의 일 예를 개략적으로 설명하기 위한 도면.

도 6은 본 발명의 실시예에 따른 메모리 시스템에서 물리주소와 논리주소를 매핑하는 동작을 설명하기 위해 도시한 도면.

도 7a와 도 7b 및 도 8a와 도 8b 도 6에서 설명된 매핑동작 중 본 발명의 실시예에 따른 특징적인 매핑동작을 설명하기 위해 도시한 도면.

도 9 내지 도 17은 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 다른 일 예들을 개략적으로 도시한 도면.

발명을 실시하기 위한 구체적인 내용

[0028] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 설명하기로 한다. 그러나, 본 발명은 이하에서 개시되는 실시예에 한정되는 것이 아니라 서로 다른 다양한 형태로 구성될 수 있으며, 단지 본 실시예는 본 발명의 개시가 완전하도록하며 통상의 지식을 가진자에게 본 발명의 범주를 완전하게 알려주기 위해 제공되는 것이다.

[0029] 도 1은 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 일 예를 개략적으로 도시한

도면이다.

- [0030] 도 1을 참조하면, 데이터 처리 시스템(100)은, 호스트(Host)(102) 및 메모리 시스템(110)을 포함한다.
- [0031] 그리고, 호스트(102)는, 전자 장치, 예컨대 휴대폰, MP3 플레이어, 랩탑 컴퓨터 등과 같은 휴대용 전자 장치들, 또는 데스크탑 컴퓨터, 게임기, TV, 프로젝터 등과 같은 전자 장치들을 포함, 즉 유무선 전자 장치들을 포함한다.
- [0032] 또한, 호스트(102)는, 적어도 하나의 운영 시스템(OS: operating system)를 포함하며, 운영 시스템은, 호스트(102)의 기능 및 동작을 전반적으로 관리 및 제어하고, 데이터 처리 시스템(100) 또는 메모리 시스템(110)을 사용하는 사용자와 호스트(102) 간에 상호 동작을 제공한다. 여기서, 운영 시스템은, 사용자의 사용 목적 및 용도에 상응한 기능 및 동작을 지원하며, 예컨대, 호스트(102)의 이동성(mobility)에 따라 일반 운영 시스템과 모바일 운영 시스템으로 구분할 수 있다. 또한, 운영 시스템에서의 일반 운영 시스템 시스템은, 사용자의 사용 환경에 따라 개인용 운영 시스템과 기업용 운영 시스템으로 구분할 수 있으며, 일 예로, 개인용 운영 시스템은, 일반 사용자를 위한 서비스 제공 기능을 지원하도록 특성화된 시스템으로, 윈도우(windows) 및 크롬(chrome) 등을 포함하고, 기업용 운영 시스템은, 고성능을 확보 및 지원하도록 특성화된 시스템으로, 윈도우 서버(windows server), 리눅스(linux) 및 유닉스(unix) 등을 포함할 수 있다. 아울러, 운영 시스템에서의 모바일 운영 시스템은, 사용자들에게 이동성 서비스 제공 기능 및 시스템의 절전 기능을 지원하도록 특성화된 시스템으로, 안드로이드(android), iOS, 윈도우 모바일(windows mobile) 등을 포함할 수 있다. 이때, 호스트(102)는, 복수의 운영 시스템들을 포함할 수 있으며, 또한 사용자 요청(user request)에 상응한 메모리 시스템(110)과의 동작 수행을 위해 운영 시스템을 실행한다. 여기서, 호스트(102)는, 사용자 요청에 해당하는 복수의 커맨드들을 메모리 시스템(110)으로 전송하며, 그에 따라 메모리 시스템(110)에서는 커맨드들에 해당하는 동작들, 즉 사용자 요청에 상응하는 동작들을 수행한다.
- [0033] 또한, 메모리 시스템(110)은, 호스트(102)의 요청에 응답하여 동작하며, 특히 호스트(102)에 의해서 액세스되는 데이터를 저장한다. 다시 말해, 메모리 시스템(110)은, 호스트(102)의 주 기억 장치 또는 보조 기억 장치로 사용될 수 있다. 여기서, 메모리 시스템(110)은 호스트(102)와 연결되는 호스트 인터페이스 프로토콜에 따라, 다양한 종류의 저장 장치들 중 어느 하나로 구현될 수 있다. 예를 들면, 메모리 시스템(110)은, 솔리드 스테이트 드라이브(SSD: Solid State Drive), MMC, eMMC(embedded MMC), RS-MMC(Reduced Size MMC), micro-MMC 형태의 멀티 미디어 카드(MMC: Multi Media Card), SD, mini-SD, micro-SD 형태의 시큐어 디지털(SD: Secure Digital) 카드, USB(Universal Storage Bus) 저장 장치, UFS(Universal Flash Storage) 장치, CF(Compact Flash) 카드, 스마트 미디어(Smart Media) 카드, 메모리 스틱(Memory Stick) 등과 같은 다양한 종류의 저장 장치들 중 어느 하나로 구현될 수 있다.
- [0034] 아울러, 메모리 시스템(110)을 구현하는 저장 장치들은, DRAM(Dynamic Random Access Memory), SRAM(Static RAM) 등과 같은 휘발성 메모리 장치와, ROM(Read Only Memory), MROM(Mask ROM), PROM(Programmable ROM), EPROM(Erasable ROM), EEPROM(Electrically Erasable ROM), FRAM(Ferromagnetic ROM), PRAM(Phase change RAM), MRAM(Magnetic RAM), RRAM(Resistive RAM), 플래시 메모리 등과 같은 비휘발성 메모리 장치로 구현될 수 있다.
- [0035] 그리고, 메모리 시스템(110)은, 호스트(102)에 의해서 액세스되는 데이터를 저장하는 메모리 장치(150), 및 메모리 장치(150)로의 데이터 저장을 제어하는 컨트롤러(130)를 포함한다.
- [0036] 여기서, 컨트롤러(130) 및 메모리 장치(150)는 하나의 반도체 장치로 집적될 수 있다. 일 예로, 컨트롤러(130) 및 메모리 장치(150)는 하나의 반도체 장치로 집적되어 SSD를 구성할 수 있다. 메모리 시스템(110)이 SSD로 이용되는 경우, 메모리 시스템(110)에 연결되는 호스트(102)의 동작 속도는 보다 개선될 수 있다. 아울러, 컨트롤러(130) 및 메모리 장치(150)는, 하나의 반도체 장치로 집적되어 메모리 카드를 구성할 수도 있으며, 일 예로 PC 카드(PCMCIA: Personal Computer Memory Card International Association), 콤팩트 플래시 카드(CF), 스마트 미디어 카드(SM, SMC), 메모리 스틱, 멀티미디어 카드(MMC, RS-MMC, MMCmicro), SD 카드(SD, miniSD, microSD, SDHC), 유니버설 플래시 기억 장치(UFS) 등과 같은 메모리 카드를 구성할 수 있다.
- [0037] 또한, 다른 일 예로, 메모리 시스템(110)은, 컴퓨터, UMPC(Ultra Mobile PC), 워크스테이션, 넷북(net-book), PDA(Personal Digital Assistants), 포터블(portable) 컴퓨터, 웹 태블릿(web tablet), 태블릿 컴퓨터(tablet computer), 무선 전화기(wireless phone), 모바일 폰(mobile phone), 스마트폰(smart phone), e-북(e-book), PMP(portable multimedia player), 휴대용 게임기, 네비게이션(navigation) 장치, 블랙박스(black box), 디지털

털 카메라(digital camera), DMB(Digital Multimedia Broadcasting) 재생기, 3차원 텔레비전(3-dimensional television), 스마트 텔레비전(smart television), 디지털 음성 녹음기(digital audio recorder), 디지털 음성 재생기(digital audio player), 디지털 영상 녹화기(digital picture recorder), 디지털 영상 재생기(digital picture player), 디지털 동영상 녹화기(digital video recorder), 디지털 동영상 재생기(digital video player), 데이터 센터를 구성하는 스토리지, 정보를 무선 환경에서 송수신할 수 있는 장치, 홈 네트워크를 구성하는 다양한 전자 장치들 중 하나, 컴퓨터 네트워크를 구성하는 다양한 전자 장치들 중 하나, 텔레매틱스 네트워크를 구성하는 다양한 전자 장치들 중 하나, RFID(radio frequency identification) 장치, 또는 컴퓨팅 시스템을 구성하는 다양한 구성 요소들 중 하나 등을 구성할 수 있다.

[0038] 한편, 메모리 시스템(110)에서의 메모리 장치(150)는, 전원이 공급되지 않아도 저장된 데이터를 유지할 수 있으며, 특히 라이트(write) 동작을 통해 호스트(102)로부터 제공된 데이터를 저장하고, 리드(read) 동작을 통해 저장된 데이터를 호스트(102)로 제공한다. 여기서, 메모리 장치(150)는, 복수의 메모리 블록(memory block)들(152, 154, 156)을 포함하며, 각각의 메모리 블록들(152, 154, 156)은, 복수의 페이지들(pages)을 포함하며, 또한 각각의 페이지들은, 복수의 워드라인(WL: Word Line)들이 연결된 복수의 메모리 셀들을 포함한다. 또한, 메모리 장치(150)는, 복수의 메모리 블록들(152, 154, 156)이 각각 포함된 복수의 플레인들(plane)을 포함하며, 특히 복수의 플레인들이 각각 포함된 복수의 메모리 다이(memory die)들을 포함할 수 있다. 아울러, 메모리 장치(150)는, 비휘발성 메모리 장치, 일 예로 플래시 메모리가 될 수 있으며, 이때 플래시 메모리는 3차원(dimension) 입체 스택(stack) 구조가 될 수 있다.

[0039] 여기서, 메모리 장치(150)의 구조 및 메모리 장치(150)의 3차원 입체 스택 구조에 대해서는, 이하 도 2 내지 도 4에서 보다 구체적으로 설명하도록 하겠다.

[0040] 그리고, 메모리 시스템(110)에서의 컨트롤러(130)는, 호스트(102)로부터의 요청에 응답하여 메모리 장치(150)를 제어한다. 예컨대, 컨트롤러(130)는, 메모리 장치(150)로부터 리드된 데이터를 호스트(102)로 제공하고, 호스트(102)로부터 제공된 데이터를 메모리 장치(150)에 저장하며, 이를 위해 컨트롤러(130)는, 메모리 장치(150)의 리드, 라이트, 프로그램(program), 이레이즈(erase) 등의 동작을 제어한다.

[0041] 보다 구체적으로 설명하면, 컨트롤러(130)는, 호스트 인터페이스(Host I/F) 유닛(132), 프로세서(Processor)(134), 에러 정정 코드(ECC: Error Correction Code) 유닛(138), 파워 관리 유닛(PMU: Power Management Unit)(140), 메모리 인터페이스(Memory I/F) 유닛(142), 및 메모리(Memory)(144)를 포함한다.

[0042] 또한, 호스트 인터페이스 유닛(132)은, 호스트(102)의 커맨드(command) 및 데이터를 처리하며, USB(Universal Serial Bus), MMC(Multi-Media Card), PCI-E(Peripheral Component Interconnect-Express), SAS(Serial-attached SCSI), SATA(Serial Advanced Technology Attachment), PATA(Parallel Advanced Technology Attachment), SCSI(Small Computer System Interface), ESDI(Enhanced Small Disk Interface), IDE(Integrated Drive Electronics), MIPI(Mobile Industry Processor Interface) 등과 같은 다양한 인터페이스 프로토콜들 중 적어도 하나를 통해 호스트(102)와 통신하도록 구성될 수 있다. 여기서, 호스트 인터페이스 유닛(132)은, 호스트(102)와 데이터를 주고 받는 영역으로 호스트 인터페이스 계층(HIL: Host Interface Layer, 이하 'HIL'이라 칭하기로 함)이라 불리는 펌웨어(firmware)를 통해 구동될 수 있다.

[0043] 아울러, ECC 유닛(138)은, 메모리 장치(150)에서 처리되는 데이터의 에러 비트를 정정하며, ECC 인코더와 ECC 디코더를 포함할 수 있다. 여기서, ECC 인코더(ECC encoder)는 메모리 장치(150)에 프로그램될 데이터를 에러 정정 인코딩(error correction encoding)하여, 패리티(parity) 비트가 추가된 데이터를 생성하며, 패리티 비트가 추가된 데이터는, 메모리 장치(150)에 저장될 수 있다. 그리고, ECC 디코더(ECC decoder)는, 메모리 장치(150)에 저장된 데이터를 리드할 경우, 메모리 장치(150)로부터 리드된 데이터에 포함되는 에러를 검출 및 정정한다. 다시 말해, ECC 유닛(138)은, 메모리 장치(150)로부터 리드한 데이터를 에러 정정 디코딩(error correction decoding)한 후, 에러 정정 디코딩의 성공 여부를 판단하고, 판단 결과에 따라 지시 신호, 예컨대 에러 정정 성공(success)/실패(fail) 신호를 출력하며, ECC 인코딩 과정에서 생성된 패리티(parity) 비트를 사용하여 리드된 데이터의 에러 비트를 정정할 수 있다. 이때, ECC 유닛(138)은, 에러 비트 개수가 정정 가능한 에러 비트 한계치 이상 발생하면, 에러 비트를 정정할 수 없으며, 에러 비트를 정정하지 못함에 상응하는 에러 정정 실패 신호를 출력할 수 있다.

[0044] 여기서, ECC 유닛(138)은, LDPC(low density parity check) 코드(code), BCH(Bose, Chaudhri, Hocquenghem) 코드, 터보 코드(turbo code), 리드-솔로몬 코드(Reed-Solomon code), 컨벌루션 코드(convolution code), RSC(recursive systematic code), TCM(trellis-coded modulation), BCM(Block coded modulation) 등의 코디드

모듈레이션(coded modulation)을 사용하여 에러 정정을 수행할 수 있으며, 이에 한정되는 것은 아니다. 또한, ECC 유닛(138)은 오류 정정을 위한 회로, 모듈, 시스템, 또는 장치를 모두 포함할 수 있다.

- [0045] 그리고, PMU(140)는, 컨트롤러(130)의 파워, 즉 컨트롤러(130)에 포함된 구성 요소들의 파워를 제공 및 관리한다.
- [0046] 또한, 메모리 인터페이스 유닛(142)은, 컨트롤러(130)가 호스트(102)로부터의 요청에 응답하여 메모리 장치(150)를 제어하기 위해, 컨트롤러(130)와 메모리 장치(150) 간의 인터페이싱을 수행하는 메모리/스토리지(storage) 인터페이스가 된다. 여기서, 메모리 인터페이스 유닛(142)은, 메모리 장치(150)가 플래시 메모리, 특히 일 예로 메모리 장치(150)가 NAND 플래시 메모리일 경우에 NAND 플래시 컨트롤러(NFC: NAND Flash Controller)로서, 프로세서(134)의 제어에 따라, 메모리 장치(150)의 제어 신호를 생성하고 데이터를 처리한다. 그리고, 메모리 인터페이스 유닛(142)은, 컨트롤러(130)와 메모리 장치(150) 간의 커맨드 및 데이터를 처리하는 인터페이스, 일 예로 NAND 플래시 인터페이스의 동작, 특히 컨트롤러(130)와 메모리 장치(150) 간 데이터 입출력을 지원하며, 메모리 장치(150)와 데이터를 주고 받는 영역으로 플래시 인터페이스 계층(FIL: Flash Interface Layer, 이하 'FIL'이라 칭하기로 함)이라 불리는 펌웨어(firmware)를 통해 구동될 수 있다.
- [0047] 아울러, 메모리(144)는, 메모리 시스템(110) 및 컨트롤러(130)의 동작 메모리로서, 메모리 시스템(110) 및 컨트롤러(130)의 구동을 위한 데이터를 저장한다. 보다 구체적으로 설명하면, 메모리(144)는, 컨트롤러(130)가 호스트(102)로부터의 요청에 응답하여 메모리 장치(150)를 제어, 예컨대 컨트롤러(130)가, 메모리 장치(150)로부터 리드된 데이터를 호스트(102)로 제공하고, 호스트(102)로부터 제공된 데이터를 메모리 장치(150)에 저장하며, 이를 위해 컨트롤러(130)가, 메모리 장치(150)의 리드, 라이트, 프로그램, 이레이즈(erase) 등의 동작을 제어할 경우, 이러한 동작을 메모리 시스템(110), 즉 컨트롤러(130)와 메모리 장치(150) 간에 수행하기 위해 필요한 데이터를 저장한다.
- [0048] 여기서, 메모리(144)는, 휘발성 메모리로 구현될 수 있으며, 예컨대 정적 랜덤 액세스 메모리(SRAM: Static Random Access Memory), 또는 동적 랜덤 액세스 메모리(DRAM: Dynamic Random Access Memory) 등으로 구현될 수 있다. 아울러, 메모리(144)는, 도 1에서 도시한 바와 같이, 컨트롤러(130)의 내부에 존재하거나, 또는 컨트롤러(130)의 외부에 존재할 수 있으며, 이때 메모리 인터페이스를 통해 컨트롤러(130)로부터 데이터가 입출력되는 외부 휘발성 메모리로 구현될 수도 있다.
- [0049] 또한, 메모리(144)는, 전술한 바와 같이, 호스트(102)와 메모리 장치(150) 간 데이터 라이트 및 리드 등의 동작을 수행하기 위해 필요한 데이터, 및 데이터 라이트 및 리드 등의 동작 수행 시의 데이터를 저장하며, 이러한 데이터 저장을 위해, 프로그램 메모리, 데이터 메모리, 라이트 버퍼(buffer)/캐시(cache), 리드 버퍼/캐시, 데이터 버퍼/캐시, 맵(map) 버퍼/캐시 등을 포함한다.
- [0050] 그리고, 프로세서(134)는, 메모리 시스템(110)의 전체적인 동작을 제어하며, 특히 호스트(102)로부터의 라이트 요청 또는 리드 요청에 응답하여, 메모리 장치(150)에 대한 프로그램 동작 또는 리드 동작을 제어한다. 여기서, 프로세서(134)는, 메모리 시스템(110)의 제반 동작을 제어하기 위해 플래시 변환 계층(FTL: Flash Translation Layer, 이하 'FTL'이라 칭하기로 함)이라 불리는 펌웨어(firmware)를 구동한다. 또한, 프로세서(134)는, 마이크로프로세서 또는 중앙 처리 장치(CPU) 등으로 구현될 수 있다.
- [0051] 일 예로, 컨트롤러(130)는, 마이크로프로세서 또는 중앙 처리 장치(CPU) 등으로 구현된 프로세서(134)를 통해, 호스트(102)로부터 요청된 동작을 메모리 장치(150)에서 수행, 다시 말해 호스트(102)로부터 수신된 커맨드에 해당하는 커맨드 동작을, 메모리 장치(150)와 수행한다. 여기서, 컨트롤러(130)는, 호스트(102)로부터 수신된 커맨드에 해당하는 커맨드 동작으로 포그라운드(foreground) 동작을 수행, 예컨대 라이트 커맨드에 해당하는 프로그램 동작, 리드 커맨드에 해당하는 리드 동작, 이레이즈 커맨드(erase command)에 해당하는 이레이즈 동작, 셋 커맨드(set command)로 셋 파라미터 커맨드(set parameter command) 또는 셋 픽처 커맨드(set feature command)에 해당하는 파라미터 셋 동작 등을 수행할 수 있다.
- [0052] 그리고, 컨트롤러(130)는, 마이크로프로세서 또는 중앙 처리 장치(CPU) 등으로 구현된 프로세서(134)를 통해, 메모리 장치(150)에 대한 백그라운드(background) 동작을 수행할 수도 있다. 여기서, 메모리 장치(150)에 대한 백그라운드 동작은, 메모리 장치(150)의 메모리 블록들(152, 154, 156)에서 임의의 메모리 블록에 저장된 데이터를 다른 임의의 메모리 블록으로 카피(copy)하여 처리하는 동작, 일 예로 가비지 컬렉션(GC: Garbage Collection) 동작, 메모리 장치(150)의 메모리 블록들(152, 154, 156) 간 또는 메모리 블록들(152, 154, 156)에 저장된 데이터 간을 스왑(swap)하여 처리하는 동작, 일 예로 웨어 레벨링(WL: Wear Leveling) 동작, 컨트롤러

(130)에 저장된 맵 데이터를 메모리 장치(150)의 메모리 블록들(152,154,156)로 저장하는 동작, 일 예로 맵 플러시(map flush) 동작, 또는 메모리 장치(150)에 대한 배드 관리(bad management)하는 동작, 일 예로 메모리 장치(150)에 포함된 복수의 메모리 블록들(152,154,156)에서 배드 블록을 확인하여 처리하는 배드 블록 관리(bad block management) 동작 등을 포함한다.

[0053] 아울러, 컨트롤러(130)의 프로세서(134)에는, 메모리 장치(150)의 배드 관리를 수행하기 위한 관리 유닛(도시하지 않음)이 포함될 수 있으며, 관리 유닛은, 메모리 장치(150)에 포함된 복수의 메모리 블록들(152,154,156)에서 배드 블록을 확인한 후, 확인된 배드 블록을 배드 처리하는 배드 블록 관리를 수행한다. 여기서, 배드 관리는, 메모리 장치(150)가 플래시 메모리, 예컨대 낸드 플래시 메모리일 경우, 낸드의 특성으로 인해 데이터 라이트, 예컨대 데이터 프로그램(program) 시에 프로그램 실패(program fail)가 발생할 수 있으며, 프로그램 실패가 발생한 메모리 블록을 배드(bad) 처리한 후, 프로그램 실패된 데이터를 새로운 메모리 블록에 라이트, 즉 프로그램하는 것을 의미한다. 또한, 메모리 장치(150)가, 전술한 바와 같이, 3차원 입체 스택 구조를 가질 경우에는, 프로그램 실패에 따라 해당 블록을 배드 블록으로 처리하면, 메모리 장치(150)의 사용 효율 및 메모리 시스템(100)의 신뢰성이 급격하게 저하되므로, 보다 신뢰성 있는 배드 블록 관리 수행이 필요하다. 그러면 이하에서는, 도 2 내지 도 4를 참조하여 본 발명의 실시 예에 따른 메모리 시스템에서의 메모리 장치에 대해서 보다 구체적으로 설명하기로 한다.

[0054] 도 2는 본 발명의 실시 예에 따른 메모리 시스템에서 메모리 장치의 일 예를 개략적으로 도시한 도면이고, 도 3은 본 발명의 실시 예에 따른 메모리 장치에서 메모리 블록들의 메모리 셀 어레이 회로를 개략적으로 도시한 도면이며, 도 4는 본 발명의 실시 예에 따른 메모리 시스템에서 메모리 장치 구조를 개략적으로 도시한 도면으로, 메모리 장치가 3차원 비휘발성 메모리 장치로 구현될 경우의 구조를 개략적으로 도시한 도면이다.

[0055] 우선, 도 2를 참조하면, 메모리 장치(150)는, 복수의 메모리 블록들, 예컨대 블록0(BLK(Block)0)(210), 블록1(BLK1)(220), 블록2(BLK2)(230), 및 블록N-1(BLKN-1)(240)을 포함하며, 각각의 블록들(210,220,230,240)은, 복수의 페이지들(Pages), 예컨대 2M개의 페이지들(2MPages)을 포함한다. 여기서, 설명의 편의를 위해, 복수의 메모리 블록들이 각각 2M개의 페이지들을 포함하는 것을 일 예로 하여 설명하지만, 복수의 메모리들은, 각각 M개의 페이지들을 포함할 수도 있다. 그리고, 각각의 페이지들은, 복수의 워드라인(WL: Word Line)들이 연결된 복수의 메모리 셀들을 포함한다.

[0056] 또한, 메모리 장치(150)는, 복수의 메모리 블록들을, 하나의 메모리 셀에 저장 또는 표현할 수 있는 비트의 수에 따라, 단일 레벨 셀(SLC: Single Level Cell) 메모리 블록 및 멀티 레벨 셀(MLC: Multi Level Cell) 메모리 블록 등으로 포함할 수 있다. 여기서, SLC 메모리 블록은, 하나의 메모리 셀에 1 비트 데이터를 저장하는 메모리 셀들에 의해 구현된 복수의 페이지들을 포함하며, 데이터 연산 성능이 빠르며 내구성이 높다. 그리고, MLC 메모리 블록은, 하나의 메모리 셀에 멀티 비트 데이터(예를 들면, 2 비트 또는 그 이상의 비트)를 저장하는 메모리 셀들에 의해 구현된 복수의 페이지들을 포함하며, SLC 메모리 블록보다 큰 데이터 저장 공간을 가짐, 다시 말해 고집적화할 수 있다. 특히, 메모리 장치(150)는, MLC 메모리 블록으로, 하나의 메모리 셀에 2 비트 데이터를 저장할 수 있는 메모리 셀들에 의해 구현된 복수의 페이지들을 포함하는 MLC 메모리 블록뿐만 아니라, 하나의 메모리 셀에 3 비트 데이터를 저장할 수 있는 메모리 셀들에 의해 구현된 복수의 페이지들을 포함하는 트리플 레벨 셀(TLC: Triple Level Cell) 메모리 블록, 하나의 메모리 셀에 4 비트 데이터를 저장할 수 있는 메모리 셀들에 의해 구현된 복수의 페이지들을 포함하는 쿼드러플 레벨 셀(QLC: Quadruple Level Cell) 메모리 블록, 또는 하나의 메모리 셀에 5 비트 또는 그 이상의 비트 데이터를 저장할 수 있는 메모리 셀들에 의해 구현된 복수의 페이지들을 포함하는 다중 레벨 셀(multiple level cell) 메모리 블록 등을 포함할 수 있다.

[0057] 여기서, 본 발명의 실시 예에서는, 설명의 편의를 위해, 메모리 장치(150)가, 플래시 메모리, 예컨대 NAND 플래시 메모리 등과 같은 비휘발성 메모리 등으로 구현되는 것을 일 예로 설명하지만, 상변환 메모리(PCRAM: Phase Change Random Access Memory), 저항 메모리(RRAM(ReRAM): Resistive Random Access Memory), 강유전체 메모리(FRAM: Ferroelectrics Random Access Memory), 및 스핀 주입 자기 메모리(STT-RAM(STT-MRAM): Spin Transfer Torque Magnetic Random Access Memory) 등과 같은 메모리들 중 어느 하나의 메모리로 구현될 수도 있다.

[0058] 그리고, 각각의 블록들(210,220,230,240)은, 프로그램 동작을 통해 호스트(102)로부터 제공된 데이터를 저장하고, 리드 동작을 통해 저장된 데이터를 호스트(102)에게 제공한다.

[0059] 다음으로, 도 3을 참조하면, 메모리 시스템(110)의 메모리 장치(150)에 포함된 복수의 메모리 블록들(152,154,156)에서 각 메모리 블록(330), 메모리 셀 어레이로 구현되어 비트라인들(BL0 to BLm-1)에 각각 연결된 복수의 셀 스트링들(340)을 포함할 수 있다. 각 열(column)의 셀 스트링(340)은, 적어도 하나의 드레인 선택

트랜지스터(DST)와, 적어도 하나의 소스 선택 트랜지스터(SST)를 포함할 수 있다. 선택 트랜지스터들(DST, SST) 사이에는, 복수 개의 메모리 셀들, 또는 메모리 셀 트랜지스터들(MC0 to MCn-1)이 직렬로 연결될 수 있다. 각각의 메모리 셀(MC0 to MCn-1)은, 셀 당 복수의 비트들의 데이터 정보를 저장하는 MLC로 구성될 수 있다. 셀 스트링들(340)은 대응하는 비트라인들(BL0 to BLm-1)에 각각 전기적으로 연결될 수 있다.

[0060] 여기서, 도 3은, 낸드 플래시 메모리 셀로 구성된 각 메모리 블록(330)을 일 예로 도시하고 있으나, 본 발명의 실시 예에 따른 메모리 장치(150)에 포함된 복수의 메모리 블록(152, 154, 156)은, 낸드 플래시 메모리에만 국한되는 것은 아니라 노어 플래시 메모리(NOR-type Flash memory), 적어도 두 종류 이상의 메모리 셀들이 혼합된 하이브리드 플래시 메모리, 메모리 칩 내에 컨트롤러가 내장된 One-NAND 플래시 메모리 등으로도 구현될 수 있다. 아울러, 본 발명의 실시 예에 따른 메모리 장치(150)는, 전하 저장층이 전도성 부유 게이트로 구성된 플래시 메모리 장치는 물론, 전하 저장층이 절연막으로 구성된 차지 트랩형 플래시(Charge Trap Flash; CTF) 메모리 장치 등으로도 구현될 수 있다.

[0061] 그리고, 메모리 장치(150)의 전압 공급부(310)는, 동작 모드에 따라서 각각의 워드라인들로 공급될 워드라인 전압들(예를 들면, 프로그램 전압, 리드 전압, 패스 전압 등)과, 메모리 셀들이 형성된 벌크(예를 들면, 웰 영역)로 공급될 전압을 제공할 수 있으며, 이때 전압 공급 회로(310)의 전압 발생 동작은 제어 회로(도시하지 않음)의 제어에 의해 수행될 수 있다. 또한, 전압 공급부(310)는, 다수의 리드 데이터를 생성하기 위해 복수의 가변 리드 전압들을 생성할 수 있으며, 제어 회로의 제어에 응답하여 메모리 셀 어레이의 메모리 블록들(또는 섹터들) 중 하나를 선택하고, 선택된 메모리 블록의 워드라인들 중 하나를 선택할 수 있으며, 워드라인 전압을 선택된 워드라인 및 비선택된 워드라인들로 각각 제공할 수 있다.

[0062] 아울러, 메모리 장치(150)의 리드/라이트(read/write) 회로(320)는, 제어 회로에 의해서 제어되며, 동작 모드에 따라 감지 증폭기(sense amplifier)로서 또는 라이트 드라이버(write driver)로서 동작할 수 있다. 예를 들면, 검증/정상 리드 동작의 경우 리드/라이트 회로(320)는, 메모리 셀 어레이로부터 데이터를 리드하기 위한 감지 증폭기로서 동작할 수 있다. 또한, 프로그램 동작의 경우 리드/라이트 회로(320)는, 메모리 셀 어레이에 저장될 데이터에 따라 비트라인들을 구동하는 라이트 드라이버로서 동작할 수 있다. 리드/라이트 회로(320)는, 프로그램 동작 시 셀 어레이에 라이트될 데이터를 버퍼(미도시)로부터 수신하고, 입력된 데이터에 따라 비트라인들을 구동할 수 있다. 이를 위해, 리드/라이트 회로(320)는, 열(column)들(또는 비트라인들) 또는 열쌍(column pair)(또는 비트라인 쌍들)에 각각 대응되는 복수 개의 페이지 버퍼들(PB)(322, 324, 326)을 포함할 수 있으며, 각각의 페이지 버퍼(page buffer)(322, 324, 326)에는 복수의 래치들(도시하지 않음)이 포함될 수 있다.

[0063] 또한, 메모리 장치(150)는, 2차원 또는 3차원의 메모리 장치로 구현될 수 있으며, 특히 도 4에 도시한 바와 같이, 3차원 입체 스택 구조의 비휘발성 메모리 장치로 구현될 수 있으며, 3차원 구조로 구현될 경우, 복수의 메모리 블록들(BLK0 to BLKN-1)을 포함할 수 있다. 여기서, 도 4는, 도 1에 도시한 메모리 장치(150)의 메모리 블록들(152, 154, 156)을 보여주는 블록도로서, 각각의 메모리 블록들(152, 154, 156)은, 3차원 구조(또는 수직 구조)로 구현될 수 있다. 예를 들면, 각각의 메모리 블록들(152, 154, 156)은 제1방향 내지 제3방향들, 예컨대 x-축 방향, y-축 방향, 및 z-축 방향을 따라 신장된 구조물들을 포함하여, 3차원 구조로 구현될 수 있다.

[0064] 그리고, 메모리 장치(150)에 포함된 각 메모리 블록(330)은, 제2방향을 따라 신장된 복수의 낸드 스트링들(NS)을 포함할 수 있으며, 제1방향 및 제3방향들을 따라 복수의 낸드 스트링들(NS)이 제공될 수 있다. 여기서, 각 낸드 스트링(NS)은, 비트라인(BL), 적어도 하나의 스트링 선택라인(SSL), 적어도 하나의 접지 선택라인(GSL), 복수의 워드라인들(WL), 적어도 하나의 더미 워드라인(DWL), 그리고 공통 소스라인(CSL)에 연결될 수 있으며, 복수의 트랜지스터 구조들(TS)을 포함할 수 있다.

[0065] 즉, 메모리 장치(150)의 복수의 메모리 블록들(152, 154, 156)에서 각 메모리 블록(330)은, 복수의 비트라인들(BL), 복수의 스트링 선택라인들(SSL), 복수의 접지 선택라인들(GSL), 복수의 워드라인들(WL), 복수의 더미 워드라인들(DWL), 그리고 복수의 공통 소스라인(CSL)에 연결될 수 있으며, 그에 따라 복수의 낸드 스트링들(NS)을 포함할 수 있다. 또한, 각 메모리 블록(330)에서, 하나의 비트라인(BL)에 복수의 낸드 스트링들(NS)이 연결되어, 하나의 낸드 스트링(NS)에 복수의 트랜지스터들이 구현될 수 있다. 아울러, 각 낸드 스트링(NS)의 스트링 선택 트랜지스터(SST)는, 대응하는 비트라인(BL)과 연결될 수 있으며, 각 낸드 스트링(NS)의 접지 선택 트랜지스터(GST)는, 공통 소스라인(CSL)과 연결될 수 있다. 여기서, 각 낸드 스트링(NS)의 스트링 선택 트랜지스터(SST) 및 접지 선택 트랜지스터(GST) 사이에 메모리 셀들(MC)이 제공, 즉 메모리 장치(150)의 복수의 메모리 블록들(152, 154, 156)에서 각 메모리 블록(330)에는 복수의 메모리 셀들이 구현될 수 있다.

[0066] 도 5는 본 발명의 실시 예에 따른 메모리 시스템에서 메모리 장치에 데이터 처리 동작의 일 예를 개략적으로 설

명하기 위한 도면이다.

- [0067] 도 5를 참조하면, 컨트롤러(130)는, 호스트(102)로부터 프로그램 커맨드(program command) 및 프로그램 데이터(program data), 그리고 논리주소(logical address)를 수신할 수 있다. 이때, 컨트롤러(130)는, 프로그램 커맨드(program command)에 응답하여 프로그램 데이터(program data)를 메모리 장치(150)의 메모리 블록들(552, 554, 562, 564, 572, 574, 582, 584)에 포함된 다수의 페이지들에 프로그램하여 저장한다.
- [0068] 그리고, 컨트롤러(130)는, 프로그램 데이터에 대한 메타(meta) 데이터를 생성 및 업데이트한 후, 메모리 장치(150)의 메모리 블록들(552, 554, 562, 564, 572, 574, 582, 584)에 프로그램하여 저장한다. 이때, 메타 데이터에는 메모리 블록들(552, 554, 562, 564, 572, 574, 582, 584)에 저장된 프로그램 데이터에 대한 논리적/물리적(L2P: Logical to Physical) 정보 및 물리적/논리적(P2L: Physical to Logical) 정보가 포함된다. 또한, 메타 데이터에는 호스트(102)로부터 수신된 커맨드에 해당하는 커맨드 데이터에 대한 정보, 커맨드에 해당하는 커맨드 동작에 대한 정보, 커맨드 동작이 수행되는 메모리 장치(150)의 메모리 블록들에 대한 정보, 및 커맨드 동작에 상응한 맵 데이터 등에 대한 정보가 포함될 수 있다. 다시 말해, 메타 데이터에는 호스트(102)로부터 수신된 커맨드에 해당하는 프로그램 데이터를 제외한 나머지 모든 정보들 및 데이터가 포함될 수 있다.
- [0069] 여기서, 논리적/물리적(L2P: Logical to Physical) 정보 및 물리적/논리적(P2L: Physical to Logical) 정보는, 컨트롤러(130)에서 프로그램 커맨드에 응답하여 논리주소(logical address)에 대응하는 물리주소(physical address)를 맵핑(mapping)한 정보를 의미한다. 이때, 물리주소(physical address)는 호스트(102)로부터 수신한 프로그램 데이터를 저장할 메모리 장치(150)의 물리적 저장 공간에 대응하는 주소일 수 있다.
- [0070] 컨트롤러(130)는, 상술한 논리주소(logical address)와 물리주소(physical address) 간의 맵핑(mapping) 정보, 다시 말해 논리적/물리적(L2P: Logical to Physical) 정보 및 물리적/논리적(P2L: Physical to Logical)를 메모리 장치(150)의 메모리 블록들(552, 554, 562, 564, 572, 574, 582, 584) 중 적어도 하나 이상의 메모리 블록에 저장할 수 있다. 이때 논리적/물리적(L2P: Logical to Physical) 정보 및 물리적/논리적(P2L: Physical to Logical)를 저장한 적어도 하나 이상의 메모리 블록을 시스템 블록(system block)이라고 부를 수 있다.
- [0071] 예를 들어 설명하면, 컨트롤러(130)는, 호스트(102)로부터 수신된 프로그램 커맨드에 해당하는 프로그램 데이터를 컨트롤러(130)의 메모리(144)에 포함된 제1 버퍼(510)에 캐싱 및 버퍼링, 즉 사용자 데이터의 데이터 세그먼트들(512)을 데이터 버퍼/캐시인 제1 버퍼(510)에 저장한다. 이후, 컨트롤러(130)는, 제1 버퍼(510)에 저장된 데이터 세그먼트들(512)을, 메모리 장치(150)의 메모리 블록들(552, 554, 562, 564, 572, 574, 582, 584)에 포함된 페이지들에 프로그램하여 저장한다.
- [0072] 그리고, 컨트롤러(130)는, 프로그램 데이터의 데이터 세그먼트들(512)이 메모리 장치(150)의 메모리 블록들(552, 554, 562, 564, 572, 574, 582, 584)에 포함된 페이지들에 프로그램되어 저장됨에 따라, 메타 데이터인 L2P세그먼트들(522)과 P2L세그먼트들(524)들을 생성한 후, 컨트롤러(130)의 메모리(144)에 포함된 제2 버퍼(520)에 저장한다. 여기서, 컨트롤러(130)의 메모리(144)에서 제2 버퍼(520)에는, L2P세그먼트들(522)과 P2L세그먼트들(524)이 리스트 형태로 저장될 수 있다. 이후, 컨트롤러(130)는, 맵 플러시(map flush) 동작을 통해 제2 버퍼(520)에 저장된 L2P세그먼트들(522)과 P2L세그먼트들(524)을 메모리 장치(150)의 메모리 블록들(552, 554, 562, 564, 572, 574, 582, 584)에 포함된 페이지들에 프로그램하여 저장할 수 있다.
- [0073] 그리고, 컨트롤러(130)는, 호스트(102)로부터 리드 커맨드(read command) 및 논리주소(logical address)를 수신할 수 있다. 이때, 컨트롤러(130)는, 리드 커맨드에 응답하여 호스트(102)의 논리주소에 대응하는 L2P 세그먼트들(522)과 P2L 세그먼트들(524)을 메모리 장치(150)로부터 리드하여 제2버퍼(520)에 로딩(loading)할 수 있다. 이후, 컨트롤러(130)는, 제2버퍼(520)에 로딩된 L2P 세그먼트들(522)과 P2L 세그먼트들(524)로부터 호스트(102)의 논리주소에 대응하는 메모리 장치(150)의 물리주소를 확인하고, 확인을 통해 알 수 있는 저장위치, 즉, 메모리 블록들(552, 554, 562, 564, 572, 574, 582, 584) 중 특정 메모리 블록의 특정 페이지로부터 사용자 데이터의 데이터 세그먼트들(512)을 리드하여 제1 버퍼(510)에 저장한 후, 호스트(102)로 제공한다.
- [0074] 컨트롤러(130)는, 전술한 바와 같이 호스트(102)로부터 리드 커맨드 및 논리주소가 수신될 때마다 호스트(102)의 논리주소에 대응하는 L2P 세그먼트들(522)과 P2L 세그먼트들(524)를 리드하여 제2버퍼(520)에 로딩할 수 있다. 이와 같은 L2P 세그먼트들(522)과 P2L 세그먼트들(524)을 로딩하는 동작은 빈번하게 반복될 수록 메모리 시스템(110)의 리드 성능을 저하 시키는 원인이 될 수 있다.
- [0075] 이때, 컨트롤러(130)는, 제2버퍼부(520)의 공간을 많이 확보할수록 메모리 장치(150)로부터 한번에 더 많은 L2P 세그먼트들(522)과 P2L 세그먼트들(524)을 로딩할 수 있고, 그 결과 L2P 세그먼트들(522)과 P2L 세그먼트들

(524)에 대한 한번의 로딩 동작으로도 다수의 리드 커맨드들에 대응할 수 있다. 이를 통해 메모리 시스템(110)의 리드 성능이 향상 될 수 있다.

- [0076] 한편, L2P 세그먼트들(522)은, 특정한 논리주소에 대응하는 물리주소를 검색하는데 최적화 될 수 있고, 그 결과 리드 동작시 호스트(102)로부터 입력된 논리주소에 맵핑되는 물리주소를 검색하는데 효율적일 수 있다.
- [0077] 또한, P2L 세그먼트들(524)은, 프로그램 동작을 위해 최적화 될 수 있다. 컨트롤러(130)는, 호스트(102)로부터 프로그램 커맨드와 프로그램 데이터 및 논리주소를 수신할 때, 프로그램 데이터를 저장할 메모리 장치(150) 내 저장 공간을 빠르게 할당할 필요가 있다. 이때, 컨트롤러(130)는, 제2버퍼부(520)에 새롭게 할당 가능한 메모리 장치(150) 내 저장 공간에 대응하는 물리주소들의 리스트를 미리 로딩할 수 있다. 따라서, 컨트롤러(130)는, 호스트(102)로부터 프로그램 커맨드와 프로그램 데이터 및 논리주소가 수신된 시점에서, 제2버퍼부(520)에 로딩되어 있는 물리주소들의 리스트를 빠르게 검색하여 프로그램 데이터를 저장할 수 있는 저장 공간에 대응하는 물리주소를 논리주소와 맵핑한 후, 물리주소에 대응하는 저장 공간에 프로그램 데이터를 저장할 수 있다. 이때, P2L 세그먼트들(524)이 생성되어 제2버퍼부(520)에 일시 저장될 수 있고, 제2버퍼부(520)에 저장된 P2L 세그먼트들(524)는, 맵 플러시(map flush) 동작을 통해 메모리 장치(150)에 저장될 수 있다.
- [0078] 도 6은 본 발명의 실시예에 따른 메모리 시스템에서 물리주소와 논리주소를 매핑하는 동작을 설명하기 위해 도시한 도면이다.
- [0079] 도 7a와 도 7b 및 도 8a와 도 8b 도 6에서 설명된 매핑동작 중 본 발명의 실시예에 따른 특징적인 매핑동작을 설명하기 위해 도시한 도면이다.
- [0080] 먼저, 도 6을 참조하면, 도 1에 도시된 메모리 시스템(110)의 구성을 참조하여 컨트롤러(130) 및 비휘발성 메모리 장치(150)를 포함하는 메모리 시스템(110)의 구성이 도시된 것을 알 수 있다.
- [0081] 참고로, 도 6에서는 메모리 시스템(110)에 하나의 비휘발성 메모리 장치(150)만 포함되는 구성을 도시하였는데, 이는 설명의 편의를 위한 것일뿐이며, 실제로는 더 많은 개수의 비휘발성 메모리 장치가 포함될 수 있다.
- [0082] 또한, 비휘발성 메모리 장치(150)는, 도 1에서 설명한 것과 같이 다수의 메모리 블록들을 포함한다. 이때, 비휘발성 메모리 장치(150)에 포함된 메모리 블록들 각각은 도 2에서 설명한 것과 같이 다수의 페이지들을 포함한다.
- [0083] 또한, 도 1에서는 컨트롤러(130)에 포함된 것으로 도시되었던, 호스트 인터페이스(132)와, 프로세서(134)와, ECC 유닛(138)과, 파워 관리 유닛(140)과, 낸드 플래시 컨트롤러(142), 및 메모리(144)가 도 6에는 컨트롤러(130)에 포함되지 않은 것으로 도시되어 있는데, 이는, 어디까지나 설명의 편의를 위해 도면에서 생략된 것일 뿐, 실제로는 컨트롤러(130)에 포함되어 있을 것이다.
- [0084] 도 6을 참조하면, 컨트롤러(130)는, 도 5에서 설명한 것과 같이 호스트(102)로부터 프로그램 커맨드(미도시)를 통해 프로그램 데이터들(DATA<1:6, 31:36>)을 비휘발성 메모리 장치(150)에 프로그램하는 동작을 수행할 것을 요청받을 수 있다. 이를 위해, 컨트롤러(130)는, 호스트(102)로부터 프로그램 커맨드(미도시)와 함께 프로그램 데이터들(DATA<1:6, 31:36>) 및 프로그램 데이터들(DATA<1:6, 31:36>)에 대응하는 다수의 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)을 인가받는다(510, 520).
- [0085] 이때, 호스트(102)로부터부터 컨트롤러(130)로 전달된 프로그램 데이터들(DATA<1:6, 31:36>) 각각은 다양한 크기를 가질 수 있다. 즉, 프로그램 데이터들(DATA<1:6, 31:36>)는, 설정된 크기와 동일하거나 작은 크기를 갖는 노말 데이터(normal data) 및 설정된 크기보다 큰 크기를 갖는 연속 데이터(sequential data)를 포함할 수 있다. 이때, 프로그램 데이터들(DATA<1:6, 31:36>) 각각의 크기에 대한 정보는 프로그램 커맨드(미도시)와 함께 호스트(102)에서 컨트롤러(130)로 전달될 수 있다.
- [0086] 그리고, 설정된 크기는, 메모리 장치(150)에서 하나의 물리주소(PA<x>))를 통해 관리할 수 있는 저장공간에 대응하는 크기일 수 있다.
- [0087] 따라서, 컨트롤러(130)는, 호스트(102)로부터 전달받은 프로그램 데이터들(DATA<1:6, 31:36>) 각각의 크기에 따라 프로그램 데이터들(DATA<1:6, 31:36>) 각각에 대응하는 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)의 개수 및 물리주소들(PA<111:125>, PA<7001:7012>)의 개수를 달라질 수 있다.

- [0088] 즉, 컨트롤러(130)는, 호스트(102)로부터 요청된 프로그램 동작에서 노말 데이터 및 그에 대응하는 하나의 논리주소(LA<x>)가 수신될 때, 수신된 노말 데이터를 저장하기 위한 메모리 장치(150)의 저장영역을 가리키는 하나의 물리주소(PA<x>)를 선택하여 수신된 하나의 논리주소(LA<x>)에 매핑시킨다.
- [0089] 또한, 컨트롤러(130)는, 호스트(102)로부터 요청된 프로그램 동작에서 연속 데이터 및 그에 대응하는 다수의 연속된 논리주소(LA<x>)가 수신될 때, 수신된 노말 데이터를 저장하기 위한 메모리 장치(150)의 저장영역을 가리키는 다수의 연속된 물리주소(PA<x>)를 선택하여 수신된 다수의 연속된 논리주소에 매핑시킨다.
- [0090] 구체적으로, 컨트롤러(130)는, 호스트(102)로부터 전달받은 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)을 물리주소들(PA<111:125>, PA<7001:7012>)로 매핑(mapping)시키기 위한 매핑정보들을 관리한다(530, 540).
- [0091] 먼저, 컨트롤러(130)는, 호스트(102)로부터 전달된 프로그램 데이터들(DATA<1:6, 31:36>) 중 설정된 크기보다 작은 데이터를 노말 데이터(DATA<1, 4, 5, 33, 35, 36>)로 판단하고, 설정된 크기보다 큰 데이터를 연속 데이터(DATA<2, 3, 6, 31, 32, 34>)로 판단한다.
- [0092] 따라서, 컨트롤러(130)는, 노말 데이터(DATA<1, 4, 5, 33, 35, 36>)에 대응하는 노말논리주소(LA<11>, LA<41>, LA<51>, LA<421>, LA<441>, LA<451>)를 노말물리주소(PA<111, 120, 121, 7008, 7011, 7012>)에 매핑시킨다. 이때, 노말논리주소(LA<11>, LA<41>, LA<51>, LA<421>, LA<441>, LA<451>)와 노말물리주소(PA<111, 120, 121, 7008, 7011, 7012>)는 일 대 일로 매핑되는 것을 알 수 있다.
- [0093] 이렇게, 컨트롤러(130)는, 매핑정보들(530, 540) 중 노말 데이터(DATA<1, 4, 5, 33, 35, 36>)를 위한 매핑정보들, 즉, 노말논리주소(LA<11>, LA<41>, LA<51>, LA<421>, LA<441>, LA<451>)와 노말물리주소(PA<111, 120, 121, 7008, 7011, 7012>)를 매핑시킨 매핑정보들을 노말매핑정보들로서 관리한다(550).
- [0094] 그리고, 컨트롤러(130)는, 연속 데이터(DATA<2, 3, 6, 31, 32, 34>)에 대응하는 연속논리주소(LA<21:25>, LA<31:33>, LA<61:64>, LA<401:404>, LA<411:413>, LA<431:432>)를 연속물리주소(PA<112:116, 117:119, 122:125, 7001:7004, 7005:7007, 7009:7010>)에 매핑시킨다. 이때, 연속논리주소(LA<21:25>, LA<31:33>, LA<61:64>, LA<401:404>, LA<411:413>, LA<431:432>)와 연속물리주소(PA<112:116, 117:119, 122:125, 7001:7004, 7005:7007, 7009:7010>)는 다수대 대 다수로 매핑되는 것을 알 수 있다. 또한, 도 7b를 함께 참조하면, 연속논리주소(LA<21:25>, LA<31:33>, LA<61:64>, LA<401:404>, LA<411:413>, LA<431:432>) 및 연속물리주소(PA<112:116, 117:119, 122:125, 7001:7004, 7005:7007, 7009:7010>)는, 각각 압축된 형태로 매핑되는 것을 알 수 있다.
- [0095] 이렇게, 컨트롤러(130)는, 매핑정보들(530, 540) 중 연속 데이터(DATA<2, 3, 6, 31, 32, 34>)를 위한 매핑정보들, 즉, 연속논리주소(LA<21:25>, LA<31:33>, LA<61:64>, LA<401:404>, LA<411:413>, LA<431:432>)와 연속물리주소(PA<112:116, 117:119, 122:125, 7001:7004, 7005:7007, 7009:7010>)를 매핑시킨 매핑정보들을 연속매핑정보들로서 관리한다(560).
- [0096] 한편, 본 발명의 실시예에 따른 컨트롤러(130)는, 프로그램 데이터들(DATA<1:6, 31:36>) 각각에 대응하는 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)이 어떤 논리주소 범위에 포함되어 있는지에 따라 제1 논리주소(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>)와 제2 논리주소(LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)로 구분한다. 즉, 컨트롤러(130)는, 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>) 중 논리주소 값의 설정된 제1 범위에 포함되는 경우, 예컨대, '200'보다 작은 값을 갖는 경우 제1 논리주소(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>)로서 판단할 수 있다. 또한, 컨트롤러(130)는, 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>) 중 논리주소 값의 설정된 제2 범위에 포함되는 경우, 예컨대, '201'보다 큰 값을 갖는 경우 제2 논리주소(LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)로서 판단할 수 있다. 참고로, 설정된 제1 범위 및 설정된 제2 범위로 예시한 것은 어디까지나 하나의 실시예일 뿐이며, 설계자의 선택에 따라 얼마든지 다를 수 있다. 물론, 논리주소들을 두 개의 설정된 범위로 나누는 것도 어디까지나 하나의 실시예일 뿐이며, 설계자의 선택에 따라 두 개보다 더 많은 개수의 설정된 범위로 나누는 것도 얼마든지 가능하다.
- [0097] 이렇게, 본 발명의 실시예에 따른 컨트롤러(130)는, 호스트(102)로부터 인가된 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)을 물리주소들(PA<111:125>, PA<7001:7012>)로 매핑(mapping)시키기 위한 매핑정보들을 관리한다(530, 540).

LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)을 제1 논리주소(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>)와 제2 논리주소(LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)로 구분한 뒤, 제1 논리주소(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>)에 제1 물리주소(PA<111:125>)를 매핑시키고, 제2 논리주소(LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)에 제2 물리주소(PA<7001:7012>)를 매핑시킨다.

[0098] 정리하면, 컨트롤러(130)는, 호스트(102)로부터 인가된 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>) 중 N개의 제1 논리주소(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>)를 N개의 제1 물리주소(PA<111:125>)에 매핑시키기 위한 제1 매핑정보들(530)을 관리한다. 또한, 컨트롤러(130)는, 호스트(102)로부터 인가된 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>) 중 M개의 제2 논리주소(LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)를 M개의 제2 물리주소(PA<7001:7012>)에 매핑시키기 위한 제2 매핑정보들(540)을 관리한다. 여기서, N과 M은 각각 자연수이다. 또한, 도 6에서와 같이 N은 '15'이고, M은 '12'로 예시되며, 이하에서는 도 6에 예시된 값을 기준으로 설명하도록 하겠다.

[0099] 구체적으로, 컨트롤러(130)는, 호스트(102)로부터 인가된 총 27개의 논리주소들(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>, LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)을 15개의 제1 논리주소(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>)와 12개의 제2 논리주소(LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)로 구분한다. 또한, 컨트롤러(130)는, 15개의 제1 논리주소(LA<11>, LA<21:25>, LA<31:33>, LA<41>, LA<51>, LA<61:64>)를 15개의 제1 물리주소(PA<111:125>)에 매핑시키기 위한 제1 매핑정보들(530)을 관리한다. 또한, 컨트롤러(130)는, 12개의 제2 논리주소(LA<401:404>, LA<411:413>, LA<421>, LA<431:432>, LA<441>, LA<451>)를 12개의 제2 물리주소(PA<7001:7012>)에 매핑시키기 위한 제2 매핑정보들(540)을 관리한다.

[0100] 도 7a를 함께 참조하면, 컨트롤러(130)가 노말매핑정보들(550)을 어떻게 관리하는지 알 수 있다.

[0101] 구체적으로, 컨트롤러(130)는, 연속되지 않은 한 개씩의 제1 논리주소(LA<11>, LA<41>, LA<51>) 및 제1 물리주소(PA<111>, PA<120>, PA<121>)를 매핑시키기 위한 제1 매핑정보들(530)을 제1 노말매핑정보들(552)로서 관리한다.

[0102] 또한, 컨트롤러(130)는, 연속되지 않은 한 개씩의 제2 논리주소(LA<421>, LA<441>, LA<451>) 및 제1 물리주소(PA<7008>, PA<7011>, PA<7012>)를 매핑시키기 위한 제2 매핑정보들(540)을 제2 노말매핑정보들(554)로서 관리한다.

[0103] 도 7b를 함께 참조하면, 컨트롤러(130)가 연속매핑정보들(560)을 어떻게 관리하는지 알 수 있다.

[0104] 구체적으로, 컨트롤러(130)는, 연속된 값을 갖는 적어도 두 개 이상의 제1 논리주소(LA<21:25>, LA<31:33>, LA<61:64>) 및 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>)를 매핑시키기 위한 제1 매핑정보들(530)을 압축시켜 제1 연속매핑정보들(562)로서 관리한다.

[0105] 또한, 컨트롤러(130)는, 연속된 값을 갖는 적어도 두 개 이상의 제2 논리주소(LA<401:404>, LA<411:413>, LA<431:432>) 및 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>)를 매핑시키기 위한 제2 매핑정보들(540)을 압축시켜 제2 연속매핑정보들(564)로서 관리한다.

[0106] 이때, 도 8a 및 도 8b를 함께 참조하면, 컨트롤러(130)는, 제1 연속매핑정보들(562) 각각을 관리하기 위해 제1 크기의 비트(bit)를 사용하고, 제2 연속매핑정보들(564) 각각을 관리하기 위해 제2 크기의 비트를 사용한다. 즉, 컨트롤러(130)는, 제1 연속매핑정보들(562) 각각을 관리하기 위해 사용하는 비트수와 제2 연속매핑정보들(564) 각각을 관리하기 위해 사용하는 비트수를 서로 다르게 하는 것을 알 수 있다. 예컨대, 도면에서와 같이 제1 연속매핑정보들(562) 각각을 관리하기 위해 13비트를 사용하고, 제2 연속매핑정보들(564) 각각을 관리하기 위해 15비트를 사용할 수 있다.

[0107] 또한, 도면에는 직접적으로 도시되지 않았지만, 제1 노말매핑정보들(552) 및 제2 노말매핑정보들(554) 각각은 제3 크기의 비트를 사용한다. 즉, 제1 노말매핑정보들(552) 및 제2 노말매핑정보들(554) 각각을 관리하기 위해 사용하는 비트수를 동일하게 하는 것을 알 수 있다.

- [0108] 다시, 도 7b를 참조하면, 제1 연속매핑정보들(562) 각각은, 제1 시작논리정보(5621)와, 제1 길이논리정보(5622)와, 제1 시작물리정보(5623), 및 제1 길이물리정보(5624)를 포함한다.
- [0109] 여기서, 제1 시작논리정보(5621)는, 내부에 압축되어 포함된 적어도 두 개 이상의 제1 논리주소(LA<21:25>, LA<31:33>, LA<61:64>) 중 시작논리주소(START LA / LA<21>, LA<31>, LA<61>) 및 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>)와의 매핑을 위한 제1 인덱스 정보(INDEX / 1, 2, 3)를 포함한다.
- [0110] 또한, 제1 길이논리정보(5622)는, 내부에 압축되어 포함된 제1 논리주소(LA<21:25>, LA<31:33>, LA<61:64>)의 개수에 대한 정보(LA LENGTH, 5, 3, 4)를 포함한다.
- [0111] 또한, 제1 시작물리정보(5623)는, 내부에 압축되어 포함된 적어도 두 개 이상의 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>) 중 시작물리주소(PA<112>, PA<117>, PA<122>) 및 제1 논리주소(START PA / LA<21:25>, LA<31:33>, LA<61:64>)와의 매핑을 위한 제1 인덱스 정보(INDEX / 1, 2, 3)를 포함한다.
- [0112] 또한, 제1 길이물리정보(5624)는, 내부에 압축되어 포함된 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>)의 개수에 대한 정보(PA LENGTH, 5, 3, 4)를 포함한다.
- [0113] 이때, 제1 연속매핑정보들(562) 각각은, 제1 인덱스 정보(INDEX / 1, 2, 3)를 통해 제1 시작논리정보(5621)와 제1 시작물리정보(5623) 서로 매핑시킬 수 있다.
- [0114] 그리고, 제2 연속매핑정보들(564) 각각은, 제2 시작논리정보(5641)와, 제2 길이논리정보(5642)와, 제2 시작물리정보(5643), 및 제2 길이물리정보(5644)를 포함한다.
- [0115] 여기서, 제2 시작논리정보(5641)는, 내부에 압축되어 포함된 적어도 두 개 이상의 제2 논리주소(LA<401:404>, LA<411:413>, LA<431:432>) 중 시작논리주소(START LA / LA<401>, LA<411>, LA<431>) 및 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>)와의 매핑을 위한 제2 인덱스 정보(INDEX / a, b, c)를 포함한다.
- [0116] 또한, 제2 길이논리정보(5642)는, 내부에 압축되어 포함된 제2 논리주소(LA<401:404>, LA<411:413>, LA<431:432>)의 개수에 대한 정보(LA LENGTH, 4, 3, 2)를 포함한다.
- [0117] 또한, 제2 시작물리정보(5643)는, 내부에 압축되어 포함된 적어도 두 개 이상의 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>) 중 시작물리주소(PA<7001>, PA<7005>, PA<7009>) 및 제2 논리주소(LA<401:404>, LA<411:413>, LA<431:432>)와의 매핑을 위한 제2 인덱스 정보(INDEX / a, b, c)를 포함한다.
- [0118] 또한, 제2 길이물리정보(5644)는, 내부에 압축되어 포함된 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>)의 개수에 대한 정보(PA LENGTH, 4, 3, 2)를 포함한다.
- [0119] 이때, 제2 연속매핑정보들(564) 각각은, 제2 인덱스 정보(INDEX / a, b, c)를 통해 제2 시작논리정보(5641)와 제2 시작물리정보(5643) 서로 매핑시킬 수 있다.
- [0120] 한편, 컨트롤러(130)는, 전술한 설명과 같이 제1 연속매핑정보들(562) 각각을 관리하기 위해 사용하는 비트수와 제2 연속매핑정보들(564) 각각을 관리하기 위해 사용하는 비트수를 서로 다르게 할 수 있다.
- [0121] 이를 위해, 컨트롤러(130)는, 제1 연속매핑정보들(562) 각각에 포함된 시작물리주소(PA<112>, PA<117>, PA<122>) 중 가장 큰 값을(PA<122>) 제1 인덱스 정보(INDEX / 1, 2, 3)와 함께 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 기준으로 제1 시작물리정보(5623)를 관리한다.
- [0122] 또한, 컨트롤러(130)는, 제2 연속매핑정보들(564) 각각에 포함된 시작물리주소(PA<7001>, PA<7005>, PA<7009>) 중 가장 큰 값을 제2 인덱스 정보(INDEX / a, b, c)와 함께 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 기준으로 제2 시작물리정보(5643)를 관리한다.
- [0123] 예컨대, 도 8a를 참조하면, 제1 연속매핑정보들(562) 각각에 포함된 시작물리주소(PA<112>, PA<117>, PA<122>) 중 가장 큰 값을 갖는 시작물리주소를 'PA<122>'라고 가정할 수 있으며, 이를 제1 인덱스 정보(INDEX / 1, 2, 3)와 함께 카운팅하기 위해 10비트(10bit)의 데이터를 사용하여 제1 시작물리정보(5623)를 관리할 수 있다.
- [0124] 하지만, 도 8b를 참조하면, 제2 연속매핑정보들(564) 각각에 포함된 시작물리주소(PA<7001>, PA<7005>, PA<7009>) 중 가장 큰 값을 갖는 시작물리주소를 'PA<7009>'라고 가정할 수 있으며, 이를 제2 인덱스 정보(INDEX / a, b, c)와 함께 카운팅하기 위해 13비트(13bit)의 데이터를 사용하여 제2 시작물리정보(5643)를 관리할 수 있다.

- [0125] 즉, 제1 연속매핑정보들(562)에 포함된 시작물리정보의 값과, 제2 연속매핑정보들(564)에 포함된 시작물리정보의 값이 서로 다르고, 제1 인덱스 정보(INDEX / 1, 2, 3)의 개수와 제2 인덱스 정보(INDEX / a, b, c)의 개수도 서로 다를 수 있기 때문에, 제1 시작물리정보(5623)를 카운팅하기 위한 비트의 크기와 제2 시작물리정보(5643)를 카운팅하기 위한 비트의 크기가 다를 수 있다.
- [0126] 전술한 실시예에서 제1 연속매핑정보들(562) 각각에 포함된 시작물리주소(PA<112>, PA<117>, PA<122>)가 제2 연속매핑정보들(564) 각각에 포함된 시작물리주소(PA<7001>, PA<7005>, PA<7009>)보다 더 작은 비트의 데이터를 사용하여 관리하는 것을 가정하였다.
- [0127] 이는, '예정된 시작값'으로부터 그 값이 순차적으로 증가하는 다수의 물리주소(PA<x>) 각각이 비휘발성 메모리 장치(150)에 포함된 다수의 저장영역(미도시)들에 각각 대응한다고 가정할 때, 제1 연속매핑정보들(562) 각각에 포함된 시작물리주소(PA<112>, PA<117>, PA<122>)가 제2 연속매핑정보들(564) 각각에 포함된 시작물리주소(PA<7001>, PA<7005>, PA<7009>)보다 '예정된 시작값'으로부터 상대적으로 더 가까운 것으로 가정하였기 때문이다.
- [0128] 만약, 제1 연속매핑정보들(562) 각각에 포함된 시작물리주소(PA<112>, PA<117>, PA<122>)가 제2 연속매핑정보들(564) 각각에 포함된 시작물리주소(PA<7001>, PA<7005>, PA<7009>)보다 '예정된 시작값'으로부터 상대적으로 더 먼 것으로 가정하는 경우, 제1 연속매핑정보들(562) 각각에 포함된 시작물리주소(PA<112>, PA<117>, PA<122>)가 제2 연속매핑정보들(564) 각각에 포함된 시작물리주소(PA<7001>, PA<7005>, PA<7009>)보다 더 큰 비트의 데이터를 사용하여 관리하는 형태가 될 것이다.
- [0129] 정리하면, 컨트롤러(130)는, '예정된 시작값'으로부터 그 값이 순차적으로 증가하는 다수의 물리주소(PA<x>) 각각이 비휘발성 메모리 장치(150)에 포함된 다수의 저장영역(미도시)들에 각각 대응시킬 수 있다.
- [0130] 따라서, 컨트롤러(130)는, 제1 연속매핑정보들(562) 각각에 포함된 시작물리주소(PA<112>, PA<117>, PA<122>)가 제2 연속매핑정보들(564) 각각에 포함된 시작물리주소(PA<7001>, PA<7005>, PA<7009>)보다 '예정된 시작값'으로부터 상대적으로 더 가까운 경우, 제1 시작물리정보(5623)가 제2 시작물리정보(5643)보다 더 작은 비트 크기를 갖도록 관리한다.
- [0131] 반대로, 컨트롤러(130)는, 제1 연속매핑정보들(562) 각각에 포함된 시작물리주소(PA<112>, PA<117>, PA<122>)가 제2 연속매핑정보들(564) 각각에 포함된 시작물리주소(PA<7001>, PA<7005>, PA<7009>)보다 '예정된 시작값'으로부터 상대적으로 더 먼 경우, 제1 시작물리정보(5623)가 제2 시작물리정보(5643)보다 더 큰 비트크기를 갖도록 관리한다.
- [0132] 그리고, 컨트롤러(130)는, 제1 길이물리정보(5624) 및 제2 길이물리정보(5644)를 관리하기 위한 비트크기를 다음과 같은 여러 가지 방법 중 어느 하나의 방법을 사용하여 결정할 수 있다.
- [0133] 첫 번째 방법은, '최대개수'를 사용하는 방법이다.
- [0134] 구체적으로, 컨트롤러(130)는, 제1 연속매핑정보들(562) 각각에 압축되어 포함된 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>) 개수의 최대개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 제1 길이물리정보(5624)를 관리한다.
- [0135] 예컨대, 도 8a에서는, 제1 연속매핑정보들(562) 각각에 압축되어 포함된 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>)의 개수는 각각 5개, 3개, 4개이고, 최대개수는 5개이다. 따라서, 5개인 최대개수를 카운팅하기 위해 필요한 비트크기인 3비트(3bit)를 사용하여 제1 길이물리정보(5624)를 관리한다.
- [0136] 그리고, 컨트롤러(130)는, 제2 연속매핑정보들(564) 각각에 압축되어 포함된 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>) 개수의 최대개수를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트 크기를 사용하여 제2 길이물리정보(5644)를 관리한다.
- [0137] 예컨대, 도 8b에서는, 제2 연속매핑정보들(564) 각각에 압축되어 포함된 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>)의 개수는 각각 4개, 3개, 2개이고, 최대개수는 4개이다. 따라서, 4개인 최대개수를 카운팅하기 위해 필요한 비트크기인 2비트(2bit)를 사용하여 제2 길이물리정보(5644)를 관리한다.
- [0138] 두 번째 방법은, '평균개수'를 사용하는 방법이다.
- [0139] 구체적으로, 컨트롤러(130)는, 제1 연속매핑정보들(562) 각각에 압축되어 포함된 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>) 개수의 평균개수를 '제1 평균개수'로서 연산한다. 이어서, 컨트롤러(130)는, 제1

연속매핑정보들(562) 각각에 포함된 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>)의 개수가 '제1 평균개수' 이하가 되도록 제1 연속매핑정보들(562)의 압축률을 재설정한다. 이어서, 컨트롤러(130)는, '제1 평균개수'를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 제1 길이물리정보(5624)를 관리한다.

[0140] 예컨대, 도 8a에서는, 제1 연속매핑정보들(562) 각각에 압축되어 포함된 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>)의 개수는 각각 5개, 3개, 4개이고, 제1 평균개수는 4개가 될 수 있다. 여기서, 제1 평균개수가 4개이므로, 이를 카운팅하기 위해 필요한 비트크기인 2비트(2bit)를 사용하여 제1 길이물리정보(5624)를 관리한다. 이때, 두 번째 제1 물리주소(PA<117:119>)는 3개의 물리주소를 포함하고, 세 번째 제1 물리주소(PA<122:125>)는 4개의 물리주소를 포함한다. 따라서, 4개로 연산된 제1 평균개수를 카운팅하기 위해 필요한 2비트(2bit)의 크기로 두 번째 제1 물리주소(PA<117:119>)와 세 번째 제1 물리주소(PA<122:125>)를 카운팅하는 것은 얼마든지 가능하다. 반면, 첫 번째 제1 물리주소(PA<112:116>)는 5개의 물리주소를 포함하므로, 4개로 연산된 제1 평균개수를 카운팅하기 위해 필요한 2비트(2bit)의 크기로 첫 번째 제1 물리주소(PA<112:116>)를 카운팅하는 것은 불가능하다. 때문에, 컨트롤러(130)는, 첫 번째 제1 물리주소(PA<112:116>)에 포함된 5개의 물리주소(PA<112:116>)를 3개의 물리주소(PA<112:114>)와 2개의 물리주소(PA<115:116>)로 분리하여 제1 연속매핑정보들(562)에 포함시킴으로써, 제1 연속매핑정보들(562)의 압축률을 재설정하는 방식을 사용할 수 있다. 이렇게, 제1 연속매핑정보들(562)의 압축률을 재설정하여 두 개로 분리된 첫 번째 제1 물리주소(PA<112:114>, PA<115:116>)를 4개로 연산된 제1 평균개수를 카운팅하기 위해 필요한 2비트(2bit)의 크기로 카운팅하는 것이 가능하다.

[0141] 참고로, 도 8a에서와 같이 제1 연속매핑정보들(562)의 압축률을 재설정하게 되면, 도면에서와 같이 제1 인덱스 정보(INDEX / 1, 2, 3)의 값을 적절하게 변경(INDEX / 1.1, 1.2, 2, 3)하여 제1 논리주소(LA<21:25>, LA<31:33>, LA<61:64>)와의 매핑이 유지될 수 있도록 해야 할 것이다.

[0142] 그리고, 컨트롤러(130)는, 제2 연속매핑정보들(564) 각각에 압축되어 포함된 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>) 개수의 평균개수를 '제2 평균개수'로서 연산한다. 이어서, 컨트롤러(130)는, 제2 연속매핑정보들(564) 각각에 포함된 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>)의 개수가 '제2 평균개수' 이하가 되도록 제2 연속매핑정보들(564)의 압축률을 재설정한다. 이어서, 컨트롤러(130)는, '제2 평균개수'를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 제2 길이물리정보(5644)를 관리한다.

[0143] 예컨대, 도 8b에서는, 제2 연속매핑정보들(564) 각각에 압축되어 포함된 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>)의 개수는 각각 4개, 3개, 2개이고, 제2 평균개수는 3개가 될 수 있다. 여기서, 제2 평균개수가 3개이므로, 이를 카운팅하기 위해 필요한 비트크기인 2비트(2bit)를 사용하여 제2 길이물리정보(5644)를 관리한다.

[0144] 세 번째 방법은, '최대개수와 최소개수의 차이'를 사용하는 방법이다.

[0145] 구체적으로, 컨트롤러(130)는, 제1 연속매핑정보들(562) 각각에 압축되어 포함된 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>) 개수의 최대/최소개수를 '제1 최대개수' 및 '제1 최소개수'로서 설정한다. 이어서, 컨트롤러(130)는, '제1 최대개수'와 '제1 최소개수'의 차이를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 '제1 최소개수'를 기준으로 사용하여 제1 길이물리정보(5624)를 관리한다. 이때, 컨트롤러(130)는, 제1 연속매핑정보들(562)을 설정된 조건에 따라 K개의 그룹으로 분류한 뒤 K개의 그룹 각각에 대해 '제1 최소개수'를 독립적으로 설정하며, K는 자연수이다.

[0146] 예컨대, 도 8a에서는, 제1 연속매핑정보들(562) 각각에 압축되어 포함된 제1 물리주소(PA<112:116>, PA<117:119>, PA<122:125>)의 개수는 각각 5개, 3개, 4개이고, 제1 최대개수는 5개, 제1 최소개수는 3개가 될 수 있다. 여기서, 제1 최대개수와 제1 최소개수의 차이는 2개이므로, 이를 카운팅하기 위해 필요한 비트크기인 2비트(2bit)를 제1 최소개수인 3개를 기준으로 사용하여 제1 길이물리정보(5624)를 관리한다. 참고로, 도 8a에서는 제1 연속매핑정보들(562)이 한 개의 제1 최소개수를 기준으로 제1 길이물리정보(5624)를 관리하는 것을 알 수 있다. 만약, 도면과 다르게 제1 연속매핑정보들(562)의 개수가 특정개수를 넘어서고, 각각의 제1 연속매핑정보들(562)이 다양한 개수의 물리주소(PA<x>)를 포함하는 경우, 제1 연속매핑정보들(562)을 여러개의 그룹으로 구분한 뒤, 각각 그룹별로 제1 최소개수를 독립적으로 설정하는 것도 얼마든지 가능하다.

[0147] 그리고, 컨트롤러(130)는, 제2 연속매핑정보들(564) 각각에 압축되어 포함된 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>) 개수의 평균개수를 '제2 평균개수'로서 연산한다. 이어서, 컨트롤러(130)는, '제2 평균개수'를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 사용하여 제2 길이물리정보(5644)를 관리한다.

PA<7005:7007>, PA<7009:7010>) 개수의 최대/최소개수를 '제2 최대개수' 및 '제2 최소개수'로서 설정한다. 이어서, 컨트롤러(130)는, '제2 최대개수'와 '제2 최소개수'의 차이를 카운팅하기 위해 필요한 비트크기를 결정하고, 결정된 비트크기를 '제2 최소개수'를 기준으로 사용하여 제2 길이물리정보(5644)를 관리한다. 이때, 컨트롤러(130)는, 제2 연속매핑정보들(564)을 설정된 조건에 따라 L개의 그룹으로 분류한 뒤 L개의 그룹 각각에 대해 '제2 최소개수'를 독립적으로 설정하며, L는 자연수이다.

[0148] 예컨대, 도 8b에서는, 제2 연속매핑정보들(564) 각각에 압축되어 포함된 제2 물리주소(PA<7001:7004>, PA<7005:7007>, PA<7009:7010>)의 개수는 각각 4개, 3개, 2개이고, 제2 최대개수는 4개, 제2 최소개수는 2개가 될 수 있다. 여기서, 제2 최대개수와 제2 최소개수의 차이는 2개이므로, 이를 카운팅하기 위해 필요한 비트크기인 2비트(2bit)를 제2 최소개수인 2개를 기준으로 사용하여 제2 길이물리정보(5644)를 관리한다. 참고로, 도 8b에서는 제2 연속매핑정보들(564)이 한 개의 제2 최소개수를 기준으로 제2 길이물리정보(5644)를 관리하는 것을 알 수 있다. 만약, 도면과 다르게 제2 연속매핑정보들(564)의 개수가 특정개수를 넘어서고, 각각의 제2 연속매핑정보들(564)이 다양한 개수의 물리주소(PA<x>)를 포함하는 경우, 제2 연속매핑정보들(564)을 여러개의 그룹으로 구분한 뒤, 각각 그룹별로 제2 최소개수를 독립적으로 설정하는 것도 얼마든지 가능하다.

[0149] 그러면 이하에서는, 도 9 내지 도 17을 참조하여, 본 발명의 실시 예에 따라 도 1 내지 도 8b에서 설명한 메모리 장치(150) 및 컨트롤러(130)를 포함하는 메모리 시스템(110)이 적용된 데이터 처리 시스템 및 전자 기기들에 대해서 보다 구체적으로 설명하기로 한다.

[0150] 도 9는 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 다른 일 예를 개략적으로 도시한 도면이다. 여기서, 도 9는 본 발명의 실시 예에 따른 메모리 시스템이 적용된 메모리 카드 시스템을 개략적으로 도시한 도면이다.

[0151] 도 9를 참조하면, 메모리 카드 시스템(6100)은, 메모리 컨트롤러(6120), 메모리 장치(6130), 및 커넥터(6110)를 포함한다.

[0152] 보다 구체적으로 설명하면, 메모리 컨트롤러(6120)는, 비휘발성 메모리로 구현된 메모리 장치(6130)와 연결되며, 메모리 장치(6130)를 액세스하도록 구현된다. 예컨대, 메모리 컨트롤러(6120)는, 메모리 장치(6130)의 리드, 라이트, 이레이즈, 및 백그라운드(background) 동작 등을 제어하도록 구현된다. 그리고, 메모리 컨트롤러(6120)는, 메모리 장치(6130) 및 호스트(Host) 사이에 인터페이스를 제공하도록 구현되며, 메모리 장치(6130)를 제어하기 위한 펌웨어(firmware)를 구동하도록 구현된다. 즉, 메모리 컨트롤러(6120)는, 도 1에서 설명한 메모리 시스템(110)에서의 컨트롤러(130)에 대응되며, 메모리 장치(6130)는, 도 1에서 설명한 메모리 시스템(110)에서의 메모리 장치(150)에 대응될 수 있다.

[0153] 그에 따라, 메모리 컨트롤러(6120)는, 램(RAM: Random Access Memory), 프로세싱 유닛(processing unit), 호스트 인터페이스(host interface), 메모리 인터페이스(memory interface), 에러 정정부와 같은 구성 요소들을 포함할 수 있다.

[0154] 아울러, 메모리 컨트롤러(6120)는, 커넥터(6110)를 통해 외부 장치, 예컨대 도 1에서 설명한 호스트(102)와 통신할 수 있다. 예컨대, 메모리 컨트롤러(6120)는, 도 1에서 설명한 바와 같이, USB(Universal Serial Bus), MMC(multimedia card), eMMC(embedded MMC), PCI(peripheral component interconnection), PCIe(PCI express), ATA(Advanced Technology Attachment), Serial-ATA, Parallel-ATA, SCSI(small computer small interface), ESDI(enhanced small disk interface), IDE(Integrated Drive Electronics), 파이어와이어(Firewire), UFS(Universal Flash Storage), WIFI, Bluetooth 등과 같은 다양한 통신 규격들 중 적어도 하나를 통해 외부 장치와 통신하도록 구성될 수 있으며, 그에 따라 유선/무선 전자 기기들, 특히 모바일 전자 기기 등에 본 발명의 실시 예에 따른 메모리 시스템 및 데이터 처리 시스템이 적용될 수 있다.

[0155] 그리고, 메모리 장치(6130)는, 비휘발성 메모리로 구현, 예컨대 EPROM(Electrically Erasable and Programmable ROM), 낸드 플래시 메모리, 노어 플래시 메모리, PRAM(Phase-change RAM), ReRAM(Resistive RAM), FRAM(Ferroelectric RAM), STT-MRAM(Spin-Torque Magnetic RAM) 등과 같은 다양한 비휘발성 메모리들로 구현될 수 있다.

[0156] 아울러, 메모리 컨트롤러(6120) 및 메모리 장치(6130)는, 하나의 반도체 장치로 집적될 수 있으며, 일 예로 하나의 반도체 장치로 집적되어 솔리드 스테이트 드라이브(SSD: Solid State Drive)를 구성할 수 있으며, PC 카드(PCMCIA), 콤팩트 플래시 카드(CF), 스마트 미디어 카드(SM, SMC), 메모리 스틱, 멀티미디어 카드(MMC, RS-MMC, MMCmicro, eMMC), SD 카드(SD, miniSD, microSD, SDHC), 유니버설 플래시 기억장치(UFS) 등과 같은 메모

리 카드를 구성할 수 있다.

- [0157] 도 10은 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 다른 일 예를 개략적으로 도시한 도면이다.
- [0158] 도 10을 참조하면, 데이터 처리 시스템(6200)은, 적어도 하나의 비휘발성 메모리로 구현된 메모리 장치(6230), 및 메모리 장치(6230)를 제어하는 메모리 컨트롤러(6220)를 포함한다. 여기서, 도 10에 도시한 데이터 처리 시스템(6200)은, 도 1에서 설명한 바와 같이, 메모리 카드(CF, SD, microSD, 등), USB 저장 장치 등과 같은 저장 매체가 될 수 있으며, 메모리 장치(6230)는, 도 1에서 설명한 메모리 시스템(110)에서의 메모리 장치(150)에 대응되고, 메모리 컨트롤러(6220)는, 도 1에서 설명한 메모리 시스템(110)에서의 컨트롤러(130)에 대응될 수 있다.
- [0159] 그리고, 메모리 컨트롤러(6220)는, 호스트(6210)의 요청에 응답하여 메모리 장치(6230)에 대한 리드, 라이트, 이레이즈 동작 등을 제어하며, 메모리 컨트롤러(6220)는 적어도 하나의 CPU(6221), 버퍼 메모리, 예컨대 RAM(6222), ECC 회로(6223), 호스트 인터페이스(6224), 및 메모리 인터페이스, 예컨대 NVM 인터페이스(6225)를 포함한다.
- [0160] 여기서, CPU(6221)는, 메모리 장치(6230)에 대한 전반적인 동작, 예컨대 읽기, 쓰기, 파일 시스템 관리, 배드 페이지 관리 등을 제어할 수 있다. 그리고, RAM(6222)는, CPU(6221)의 제어에 따라 동작하며, 워크 메모리(work memory), 버퍼 메모리(buffer memory), 캐시 메모리(cache memory) 등으로 사용될 수 있다. 여기서, RAM(6222)이 워크 메모리로 사용되는 경우에, CPU(6221)에서 처리된 데이터가 임시 저장되며, RAM(6222)이 버퍼 메모리로 사용되는 경우에는, 호스트(6210)에서 메모리 장치(6230)로 또는 메모리 장치(6230)에서 호스트(6210)로 전송되는 데이터의 버퍼링을 위해 사용되며, RAM(6222)이 캐시 메모리로 사용되는 경우에는 저속의 메모리 장치(6230)가 고속으로 동작하도록 사용될 수 있다.
- [0161] 아울러, ECC 회로(6223)는, 도 1에서 설명한 컨트롤러(130)의 ECC 유닛(138)에 대응하며, 도 1에서 설명한 바와 같이, 메모리 장치(6230)로부터 수신된 데이터의 페일 비트(fail bit) 또는 에러 비트(error bit)를 정정하기 위한 에러 정정 코드(ECC: Error Correction Code)를 생성한다. 또한, ECC 회로(6223)는, 메모리 장치(6230)로 제공되는 데이터의 에러 정정 인코딩을 수행하여, 패리티(parity) 비트가 부가된 데이터를 형성한다. 여기서, 패리티 비트는, 메모리 장치(6230)에 저장될 수 있다. 또한, ECC 회로(6223)는, 메모리 장치(6230)로부터 출력된 데이터에 대하여 에러 정정 디코딩을 수행할 수 있으며, 이때 ECC 회로(6223)는 패리티(parity)를 사용하여 에러를 정정할 수 있다. 예컨대, ECC 회로(6223)는, 도 1에서 설명한 바와 같이, LDPC code, BCH code, turbo code, 리드-솔로몬 코드, convolution code, RSC, TCM, BCM 등의 다양한 코딩 모듈레이션(coded modulation)을 사용하여 에러를 정정할 수 있다.
- [0162] 그리고, 메모리 컨트롤러(6220)는, 호스트 인터페이스(6224)를 통해 호스트(6210)와 데이터 등을 송수신하며, NVM 인터페이스(6225)를 통해 메모리 장치(6230)와 데이터 등을 송수신한다. 여기서, 호스트 인터페이스(6224)는, PATA 버스, SATA 버스, SCSI, USB, PCIe, 낸드 인터페이스 등을 통해 호스트(6210)와 연결될 수 있다. 또한, 메모리 컨트롤러(6220)는, 무선 통신 기능, 모바일 통신 규격으로 WiFi 또는 LTE(Long Term Evolution) 등이 구현되어, 외부 장치, 예컨대 호스트(6210) 또는 호스트(6210) 이외의 다른 외부 장치와 연결된 후, 데이터 등을 송수신할 수 있으며, 특히 다양한 통신 규격들 중 적어도 하나를 통해 외부 장치와 통신하도록 구성됨에 따라, 유선/무선 전자 기기들, 특히 모바일 전자 기기 등에 본 발명의 실시 예에 따른 메모리 시스템 및 데이터 처리 시스템이 적용될 수 있다.
- [0163] 도 11은 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 다른 일 예를 개략적으로 도시한 도면이다. 여기서, 도 11은 본 발명의 실시 예에 따른 메모리 시스템이 적용된 솔리드 스테이트 드라이브(SSD: Solid State Drive)를 개략적으로 도시한 도면이다.
- [0164] 도 11을 참조하면, SSD(6300)는, 복수의 비휘발성 메모리들을 포함하는 메모리 장치(6340) 및 컨트롤러(6320)를 포함한다. 여기서, 컨트롤러(6320)는, 도 1에서 설명한 메모리 시스템(110)에서의 컨트롤러(130)에 대응되며, 메모리 장치(6340)는, 도 1에서 설명한 메모리 시스템(110)에서의 메모리 장치(150)에 대응될 수 있다.
- [0165] 보다 구체적으로 설명하면, 컨트롤러(6320)는, 복수의 채널들(CH1, CH2, CH3, ..., CHi)을 통해 메모리 장치(6340)와 연결된다. 그리고, 컨트롤러(6320)는, 적어도 하나의 프로세서(6321), 버퍼 메모리(6325), ECC 회로(6322), 호스트 인터페이스(6324), 및 메모리 인터페이스, 예컨대 비휘발성 메모리 인터페이스(6326)를 포함한다.

- [0166] 여기서, 버퍼 메모리(6325)는, 호스트(6310)로부터 수신된 데이터 또는 메모리 장치(6340)에 포함된 복수의 플래시 메모리들(NVMs)로부터 수신된 데이터를 임시 저장하거나, 복수의 플래시 메모리들(NVMs)의 메타 데이터, 예컨대 매핑 테이블을 포함하는 맵 데이터를 임시 저장한다. 또한, 버퍼 메모리(6325)는, DRAM, SDRAM, DDR SDRAM, LPDDR SDRAM, GRAM 등과 같은 휘발성 메모리 또는 FRAM, ReRAM, STT-MRAM, PRAM 등과 같은 비휘발성 메모리들로 구현될 수 있으며, 도 11에서는 설명의 편의를 위해 컨트롤러(6320) 내부에 존재하지만, 컨트롤러(6320) 외부에도 존재할 수 있다.
- [0167] 그리고, ECC 회로(6322)는, 프로그램 동작에서 메모리 장치(6340)로 프로그램될 데이터의 에러 정정 코드 값을 계산하고, 리드 동작에서 메모리 장치(6340)로부터 리드된 데이터를 에러 정정 코드 값에 근거로 하여 에러 정정 동작을 수행하며, 페일된 데이터의 복구 동작에서 메모리 장치(6340)로부터 복구된 데이터의 에러 정정 동작을 수행한다.
- [0168] 또한, 호스트 인터페이스(6324)는, 외부의 장치, 예컨대 호스트(6310)와 인터페이스 기능을 제공하며, 비휘발성 메모리 인터페이스(6326)는, 복수의 채널들을 통해 연결된 메모리 장치(6340)와 인터페이스 기능을 제공한다.
- [0169] 아울러, 도 1에서 설명한 메모리 시스템(110)이 적용된 SSD(6300)는, 복수개가 적용되어 데이터 처리 시스템, 예컨대 RAID(Redundant Array of Independent Disks) 시스템을 구현할 수 있으며, 이때 RAID 시스템에는, 복수의 SSD(6300)들과, 복수의 SSD(6300)들을 제어하는 RAID 컨트롤러가 포함될 수 있다. 여기서, RAID 컨트롤러는, 호스트(6310)로부터 라이트 커맨드를 수신하여, 프로그램 동작을 수행할 경우, 라이트 커맨드에 해당하는 데이터를, 복수의 RAID 레벨들, 즉 복수의 SSD(6300)들에서 호스트(6310)로부터 수신된 라이트 커맨드의 RAID 레벨 정보에 상응하여, 적어도 하나의 메모리 시스템, 다시 말해 SSD(6300)을 선택한 후, 선택한 SSD(6300)로 출력할 수 있다. 또한, RAID 컨트롤러는, 호스트(6310)로부터 리드 커맨드를 수신하여 리드 동작을 수행할 경우, 복수의 RAID 레벨들, 즉 복수의 SSD(6300)들에서 호스트(6310)로부터 수신된 리드 커맨드의 RAID 레벨 정보에 상응하여, 적어도 하나의 메모리 시스템, 다시 말해 SSD(6300)을 선택한 후, 선택한 SSD(6300)로부터 데이터를 호스트(6310)로 제공할 수 있다.
- [0170] 도 12는 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 다른 일 예를 개략적으로 도시한 도면이다. 여기서, 도 12는 본 발명의 실시 예에 따른 메모리 시스템이 적용된 eMMC(embedded multimedia card)를 개략적으로 도시한 도면이다.
- [0171] 도 12를 참조하면, eMMC(6400)는, 적어도 하나의 낸드 플래시 메모리로 구현된 메모리 장치(6440), 및 컨트롤러(6430)를 포함한다. 여기서, 컨트롤러(6430)는, 도 1에서 설명한 메모리 시스템(110)에서의 컨트롤러(130)에 대응되며, 메모리 장치(6440)는, 도 1에서 설명한 메모리 시스템(110)에서의 메모리 장치(150)에 대응될 수 있다.
- [0172] 보다 구체적으로 설명하면, 컨트롤러(6430)는, 복수의 채널들을 통해, 메모리 장치(2100)와 연결된다. 그리고, 컨트롤러(6430)는, 적어도 하나의 코어(6432), 호스트 인터페이스(6431), 및 메모리 인터페이스, 예컨대 낸드 인터페이스(6433)를 포함한다.
- [0173] 여기서, 코어(6432)는, eMMC(6400)의 전반적인 동작을 제어하며, 호스트 인터페이스(6431)는, 컨트롤러(6430)와 호스트(6410) 간의 인터페이스 기능을 제공하며, 낸드 인터페이스(6433)는, 메모리 장치(6440)와 컨트롤러(6430) 간의 인터페이스 기능을 제공한다. 예컨대, 호스트 인터페이스(6431)는, 도 1에서 설명한 바와 같이, 병렬 인터페이스, 일 예로 MMC 인터페이스가 될 수 있으며, 아울러 직렬 인터페이스, 일 예로 UHS(Ultra High Speed)-I/UHS-II, UFS 인터페이스가 될 수 있다.
- [0174] 도 13 내지 도 16은 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 다른 일 예를 개략적으로 도시한 도면이다. 여기서, 도 13 내지 도 16은 본 발명의 실시 예에 따른 메모리 시스템이 적용된 UFS(Universal Flash Storage)를 개략적으로 도시한 도면이다.
- [0175] 도 13 내지 도 16을 참조하면, 각각의 UFS 시스템들(6500,6600,6700,6800)은, 호스트들(6510,6610,6710,6810), UFS 장치들(6520,6620,6720,6820), 및 UFS 카드들(6530,6630,6730,6830)을 각각 포함할 수 있다. 여기서, 각각의 호스트(6510,6610,6710,6810)은, 유선/무선 전자 기기들, 특히 모바일 전자 기기 등의 어플리케이션 프로세서가 될 수 있으며, 또한 각각의 UFS 장치들(6520,6620,6720,6820)은, 임베디드 UFS(Embedded UFS) 장치들이 되고, 아울러 각각의 UFS 카드들(6530,6630,6730,6830)은, 외부 임베디드 UFS(External Embedded UFS) 장치 또는 리무벌 UFS 카드(Removable UFS Card)가 될 수 있다.
- [0176] 또한, 각 UFS 시스템들(6500,6600,6700,6800)에서, 각각의 호스트들(6510,6610,6710,6810), UFS 장치들

(6520,6620,6720,6820), 및 UFS 카드들(6530,6630,6730,6830) 간은, 각각 UFS 프로토콜을 통해 외부의 장치들, 예컨대 유선/무선 전자 기기들, 특히 모바일 전자 기기 등과 통신할 수 있으며, UFS 장치들(6520,6620,6720,6820)과 UFS 카드들(6530,6630,6730,6830)은, 도 1에서 설명한 메모리 시스템(110)으로 구현될 수 있다. 예컨대, 각 UFS 시스템들(6500,6600,6700,6800)에서, UFS 장치들(6520,6620,6720,6820)은, 도 10 내지 도 12에서 설명한 데이터 처리 시스템(6200), SSD(6300), 또는 eMMC(6400) 형태로 구현될 수 있으며, UFS 카드들(6530,6630,6730,6830)은, 도 9에서 설명한 메모리 카드 시스템(6100) 형태로 구현될 수 있다.

[0177] 아울러, 각 UFS 시스템들(6500,6600,6700,6800)에서, 각각의 호스트들(6510,6610,6710,6810), UFS 장치들(6520,6620,6720,6820), 및 UFS 카드들(6530,6630,6730,6830) 간은, UFS(Universal Flash Storage) 인터페이스, 예컨대 MIPI(Mobile Industry Processor Interface)에서의 MIPI M-PHY 및 MIPI UniPro(Unified Protocol)을 통해 통신을 수행할 수 있으며, 아울러 UFS 장치들(6520,6620,6720,6820)과 UFS 카드들(6530,6630,6730,6830) 간은, UFS 프로토콜이 아닌 다른 프로토콜을 통해 통신할 수 있으며, 예컨대 다양한 카드 프로토콜, 일 예로 UFDs, MMC, SD(secure digital), mini SD, Micro SD 등을 통해 통신할 수 있다.

[0178] 그리고, 도 13에 도시한 UFS 시스템(6500)에서, 호스트(6510), UFS 장치(6520), 및 UFS 카드(6530)에는, UniPro이 각각 존재하며, 호스트(6510)는, UFS 장치(6520) 및 UFS 카드(6530)와 각각 통신을 수행하기 위해, 스위칭(switiching) 동작을 수행하며, 특히 호스트(6510)는, UniPro에서의 링크 레이어(Link Layer) 스위칭, 예컨대 L3 스위칭을 통해, UFS 장치(6520)와 통신을 수행하거나 또는 UFS 카드(6530)와 통신을 수행한다. 이때, UFS 장치(6520)와 UFS 카드(6530) 간은, 호스트(6510)의 UniPro에서 링크 레이어 스위칭을 통해, 통신을 수행할 수도 있다. 여기서, 본 발명의 실시 예에서는, 설명의 편의를 위해, 호스트(6510)에 각각 하나의 UFS 장치(6520) 및 UFS 카드(6530)가 연결되는 것을 일 예로 하여 설명하였지만, 복수의 UFS 장치들과 UFS 카드들이, 호스트(6410)에 병렬 형태 또는 스타 형태로 연결될 수도 있으며, 또한 복수의 UFS 카드들이, UFS 장치(6520)에, 병렬 형태 또는 스타 형태로 연결되거나 직렬 형태 또는 체인 형태로 연결될 수도 있다.

[0179] 또한, 도 14에 도시한 UFS 시스템(6600)에서, 호스트(6610), UFS 장치(6620), 및 UFS 카드(6630)에는, UniPro이 각각 존재하며, 스위칭 동작을 수행하는 스위칭 모듈(6640), 특히 UniPro에서의 링크 레이어 스위칭, 예컨대 L3 스위칭 동작을 수행하는 스위칭 모듈(6640)을 통해, 호스트(6610)는, UFS 장치(6620)와 통신을 수행하거나 또는 UFS 카드(6630)와 통신을 수행한다. 이때, UFS 장치(6620)와 UFS 카드(6630) 간은, 스위칭 모듈(6640)의 UniPro에서 링크 레이어 스위칭을 통해, 통신을 수행할 수도 있다. 여기서, 본 발명의 실시 예에서는, 설명의 편의를 위해, 스위칭 모듈(6640)에 각각 하나의 UFS 장치(6620) 및 UFS 카드(6630)가 연결되는 것을 일 예로 하여 설명하였지만, 복수의 UFS 장치들과 UFS 카드들이, 스위칭 모듈(6640)에 병렬 형태 또는 스타 형태로 연결될 수도 있으며, 또한 복수의 UFS 카드들이, UFS 장치(6620)에, 병렬 형태 또는 스타 형태로 연결되거나 직렬 형태 또는 체인 형태로 연결될 수도 있다.

[0180] 아울러, 도 15에 도시한 UFS 시스템(6700)에서, 호스트(6710), UFS 장치(6720), 및 UFS 카드(6730)에는, UniPro이 각각 존재하며, 스위칭 동작을 수행하는 스위칭 모듈(6740), 특히 UniPro에서의 링크 레이어 스위칭, 예컨대 L3 스위칭 동작을 수행하는 스위칭 모듈(6740)을 통해, 호스트(6710)는, UFS 장치(6720)와 통신을 수행하거나 또는 UFS 카드(6730)와 통신을 수행한다. 이때, UFS 장치(6720)와 UFS 카드(6730) 간은, 스위칭 모듈(6740)의 UniPro에서 링크 레이어 스위칭을 통해, 통신을 수행할 수도 있으며, 스위칭 모듈(6740)은, UFS 장치(6720)의 내부 또는 외부에서 UFS 장치(6720)와 하나의 모듈로 구현될 수 있다. 여기서, 본 발명의 실시 예에서는, 설명의 편의를 위해, 스위칭 모듈(6740)에 각각 하나의 UFS 장치(6620) 및 UFS 카드(6630)가 연결되는 것을 일 예로 하여 설명하였지만, 스위칭 모듈(6740)과 UFS 장치(6720)가 각각 구현된 복수의 모듈들이, 호스트(6710)에 병렬 형태 또는 스타 형태로 연결되거나, 각각의 모듈들 간에 직렬 형태 또는 체인 형태로 연결될 수도 있으며, 또한 복수의 UFS 카드들이 스위칭 모듈(6740)에 병렬 형태 또는 스타 형태로 연결될 수도 있다.

[0181] 그리고, 도 16에 도시한 UFS 시스템(6800)에서, 호스트(6810), UFS 장치(6820), 및 UFS 카드(6830)에는, M-PHY 및 UniPro이 각각 존재하며, UFS 장치(6820)는, 호스트(6810) 및 UFS 카드(6830)와 각각 통신을 수행하기 위해, 스위칭 동작을 수행하며, 특히 UFS 장치(6820)는, 호스트(6810)와의 통신을 위한 M-PHY 및 UniPro 모듈과, UFS 카드(6830)와의 통신을 위한 M-PHY 및 UniPro 모듈 간, 스위칭, 예컨대 타겟(Target) ID(identifier) 스위칭을 통해, 호스트(6810)와 통신을 수행하거나 또는 UFS 카드(6830)와 통신을 수행한다. 이때, 호스트(6810)와 UFS 카드(6530) 간은, UFS 장치(6820)의 M-PHY 및 UniPro 모듈 간 타겟 ID 스위칭을 통해, 통신을 수행할 수도 있다. 여기서, 본 발명의 실시 예에서는, 설명의 편의를 위해, 호스트(6810)에 하나의 UFS 장치(6820)가 연결되고, 또한 하나의 UFS 장치(6820)에 하나의 UFS 카드(6830)가 연결되는 것을 일 예로 하여 설명하였지만, 호스트(6810)에 복수의 UFS 장치들이 병렬 형태 또는 스타 형태로 연결되거나 직렬 형태 또는 체

인 형태로 연결될 수도 있으며, 하나의 UFS 장치(6820)에 복수의 UFS 카드들이 병렬 형태 또는 스타 형태로 연결되거나 직렬 형태 또는 체인 형태로 연결될 수도 있다.

[0182] 도 17은 본 발명의 실시 예에 따른 메모리 시스템을 포함하는 데이터 처리 시스템의 또 다른 일 예를 개략적으로 도시한 도면이다. 여기서, 도 17은 본 발명에 따른 메모리 시스템이 적용된 사용자 시스템을 개략적으로 도시한 도면이다.

[0183] 도 17을 참조하면, 사용자 시스템(6900)은, 애플리케이션 프로세서(6930), 메모리 모듈(6920), 네트워크 모듈(6940), 스토리지 모듈(6950), 및 사용자 인터페이스(6910)를 포함한다.

[0184] 보다 구체적으로 설명하면, 애플리케이션 프로세서(6930)는, 사용자 시스템(6900)에 포함된 구성 요소들, 운영 시스템(OS: Operating System)을 구동시키며, 일 예로 사용자 시스템(6900)에 포함된 구성 요소들을 제어하는 컨트롤러들, 인터페이스들, 그래픽 엔진 등을 포함할 수 있다. 여기서, 애플리케이션 프로세서(6930)는 시스템-온-칩(SoC: System-on-Chip)으로 제공될 수 있다.

[0185] 그리고, 메모리 모듈(6920)은, 사용자 시스템(6900)의 메인 메모리, 동작 메모리, 버퍼 메모리, 또는 캐시 메모리로 동작할 수 있다. 여기서, 메모리 모듈(6920)은, DRAM, SDRAM, DDR SDRAM, DDR2 SDRAM, DDR3 SDRAM, LPDDR SDRAM, LPDDR3 SDRAM, LPDDR3 SDRAM 등과 같은 휘발성 랜덤 액세스 메모리 또는 PRAM, ReRAM, MRAM, FRAM 등과 같은 비휘발성 랜덤 액세스 메모리를 포함할 수 있다. 예컨대, 애플리케이션 프로세서(6930) 및 메모리 모듈(6920)은, POP(Package on Package)를 기반으로 패키징되어 실장될 수 있다.

[0186] 또한, 네트워크 모듈(6940)은, 외부 장치들과 통신을 수행할 수 있다. 예를 들어, 네트워크 모듈(6940)은, 유선 통신을 지원할뿐만 아니라, CDMA(Code Division Multiple Access), GSM(Global System for Mobile communication), WCDMA(wideband CDMA), CDMA-2000, TDMA(Time Division Multiple Access), LTE(Long Term Evolution), Wimax, WLAN, UWB, 블루투스, WI-DI 등과 같은 다양한 무선 통신을 지원함으로써, 유선/무선 전자 기기들, 특히 모바일 전자 기기 등과 통신을 수행할 수 있으며, 그에 따라 본 발명의 실시 예에 따른 메모리 시스템 및 데이터 처리 시스템이 유선/무선 전자 기기들에 적용될 수 있다. 여기서, 네트워크 모듈(6940)은, 애플리케이션 프로세서(6930)에 포함될 수 있다.

[0187] 아울러, 스토리지 모듈(6950)은, 데이터를 저장, 예컨대 애플리케이션 프로세서(6930)로부터 수신한 데이터를 저장한 후, 스토리지 모듈(6950)에 저장된 데이터를 애플리케이션 프로세서(6930)로 전송할 수 있다. 여기서, 스토리지 모듈(6950)은, PRAM(Phasechange RAM), MRAM(Magnetic RAM), RRAM(Resistive RAM), NAND flash, NOR flash, 3차원 구조의 NAND 플래시 등과 같은 비휘발성 메모리 등으로 구현될 수 있으며, 또한 사용자 시스템(6900)의 메모리 카드, 외장형 드라이브 등과 같은 탈착식 저장 매체(removable drive)로 제공될 수 있다. 즉, 스토리지 모듈(6950)은, 도 1에서 설명한 메모리 시스템(110)에 대응될 수 있으며, 아울러 도 11 내지 도 16에서 설명한 SSD, eMMC, UFS로 구현될 수도 있다.

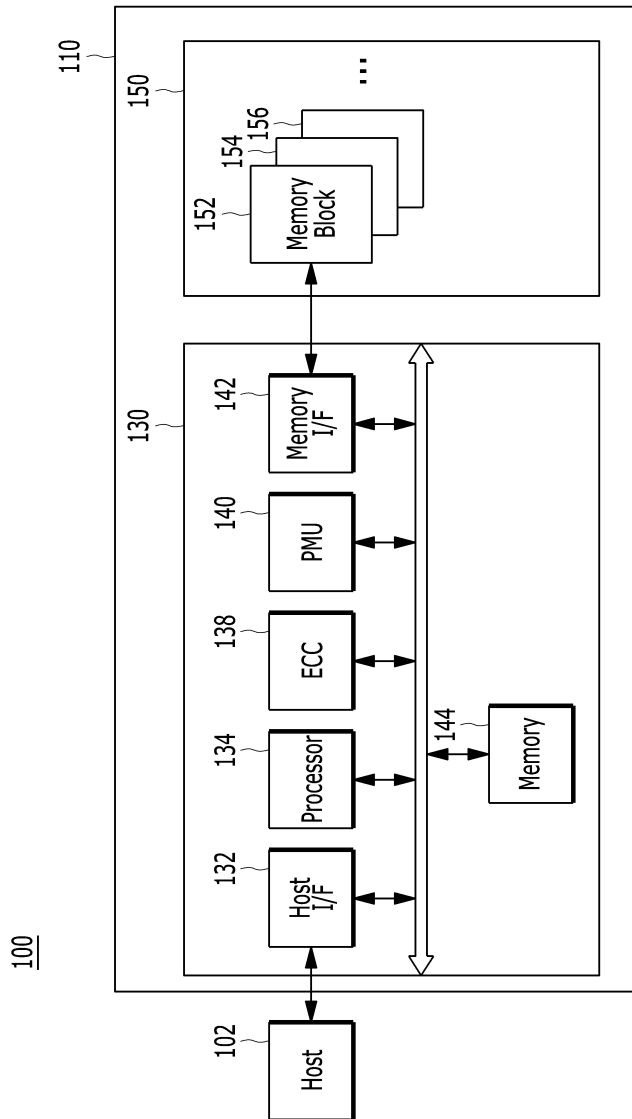
[0188] 그리고, 사용자 인터페이스(6910)는, 애플리케이션 프로세서(6930)에 데이터 또는 명령어를 입력하거나 또는 외부 장치로 데이터를 출력하는 인터페이스들을 포함할 수 있다. 예컨대, 사용자 인터페이스(6910)는, 키보드, 키패드, 버튼, 터치 패널, 터치 스크린, 터치 패드, 터치 볼, 카메라, 마이크, 자이로스코프 센서, 진동 센서, 압전 소자 등과 같은 사용자 입력 인터페이스들을 포함할 수 있으며, 아울러 LCD(Liquid Crystal Display), OLED(Organic Light Emitting Diode) 표시 장치, AMOLED(Active Matrix OLED) 표시 장치, LED, 스피커, 모터 등과 같은 사용자 출력 인터페이스들을 포함할 수 있다.

[0189] 또한, 본 발명의 실시 예에 따라 도 1에서 설명한 메모리 시스템(110)이, 사용자 시스템(6900)의 모바일 전자 기기에 적용될 경우, 애플리케이션 프로세서(6930)는, 모바일 전자 기기의 전반적인 동작을 제어하며, 네트워크 모듈(6940)은, 통신 모듈로서, 전술한 바와 같이 외부 장치와의 유선/무선 통신을 제어한다. 아울러, 사용자 인터페이스(6910)는, 모바일 전자 기기의 디스플레이/터치 모듈로 애플리케이션 프로세서(6930)에서 처리된 데이터를 디스플레이하거나, 터치 패널로부터 데이터를 입력 받도록 지원한다.

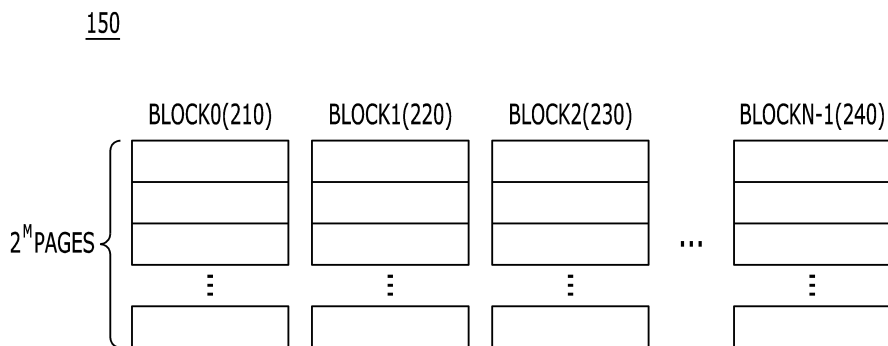
[0190] 한편, 본 발명의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로, 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 안되며 후술하는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

도면

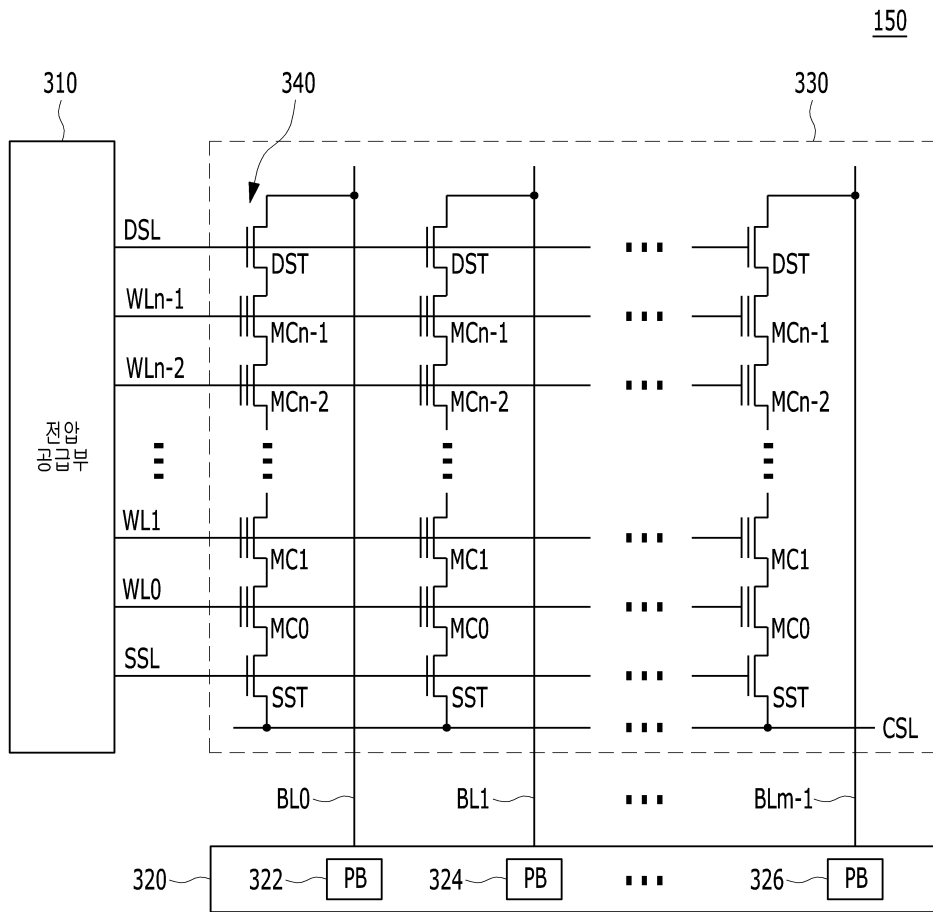
도면1



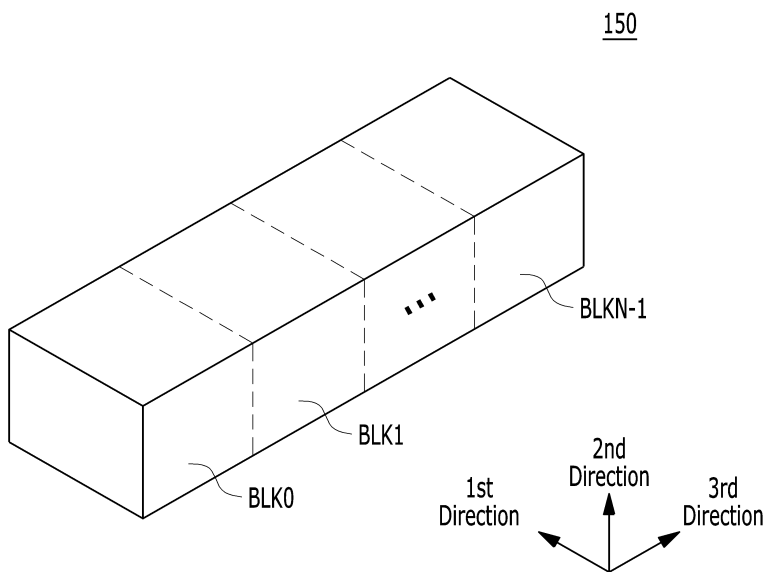
도면2

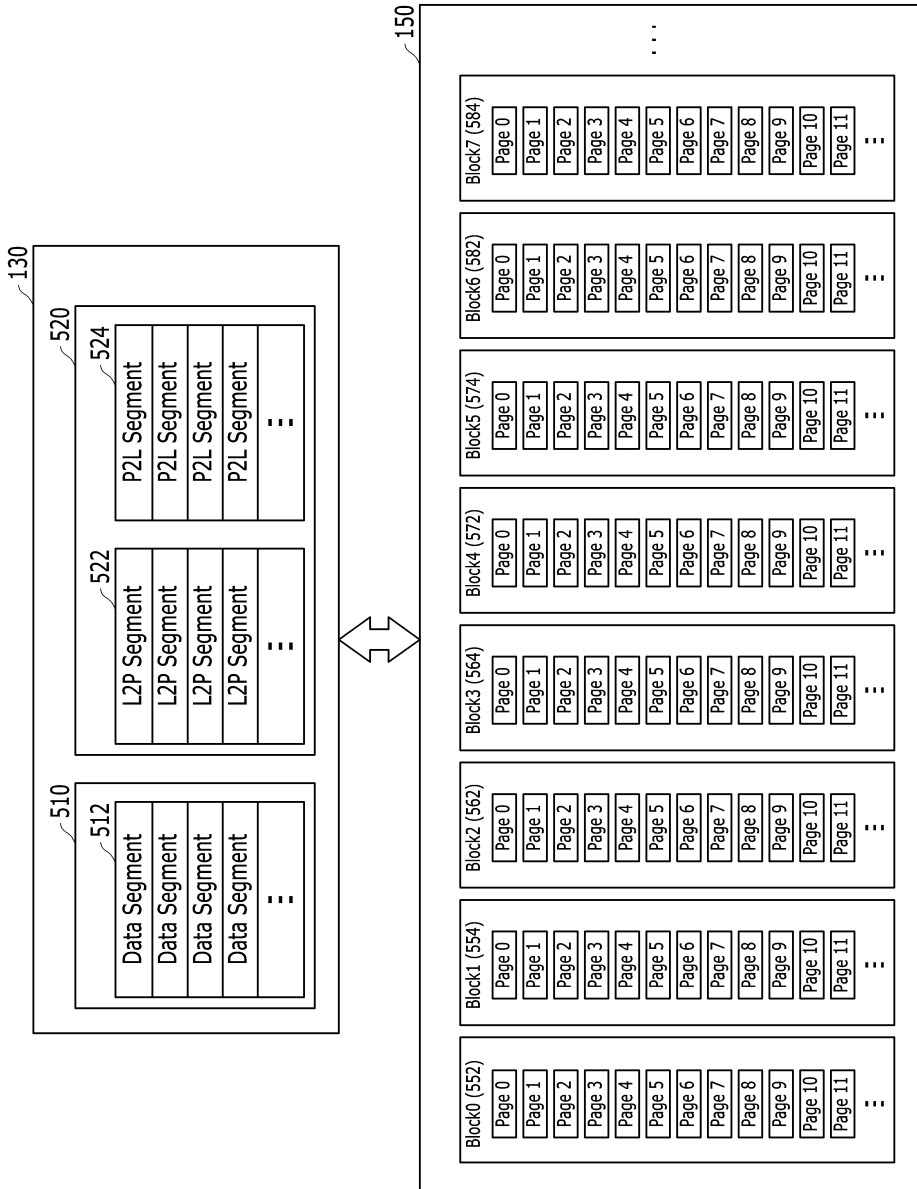


도면3

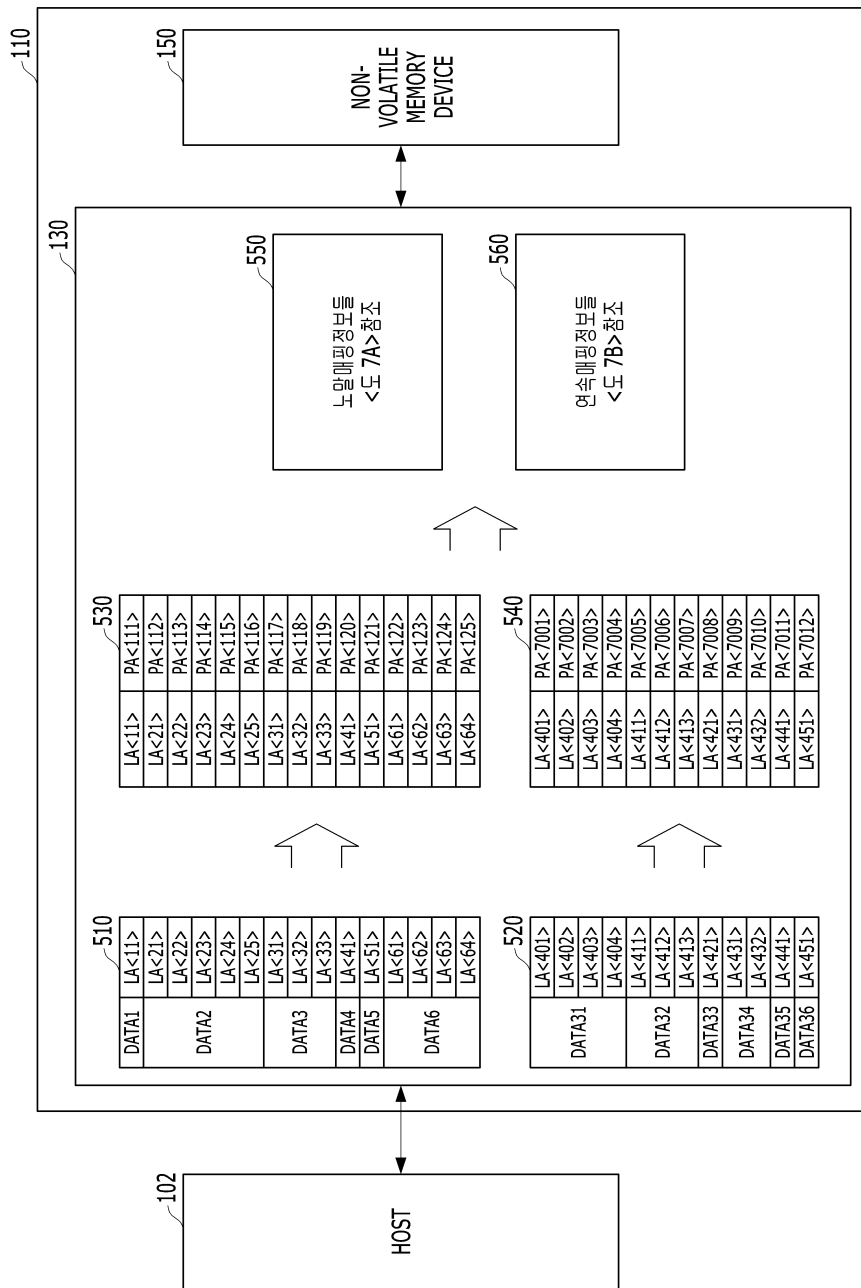


도면4





도면6

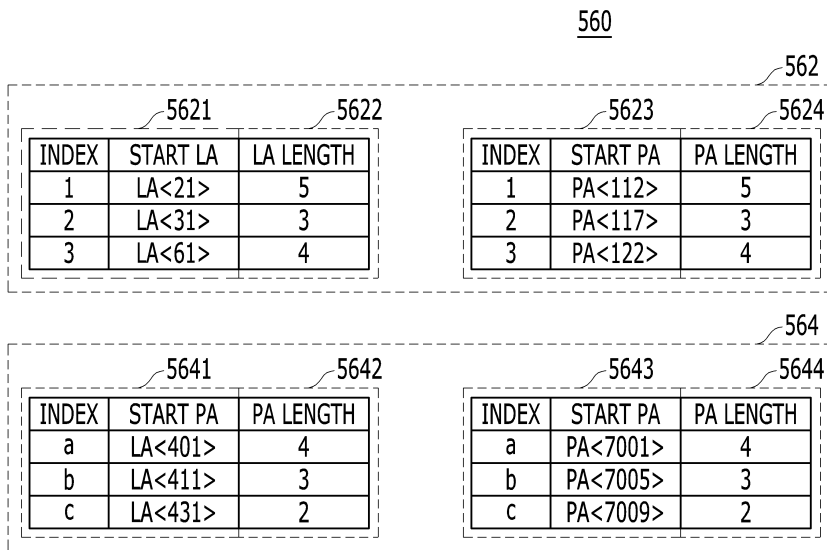


도면7a

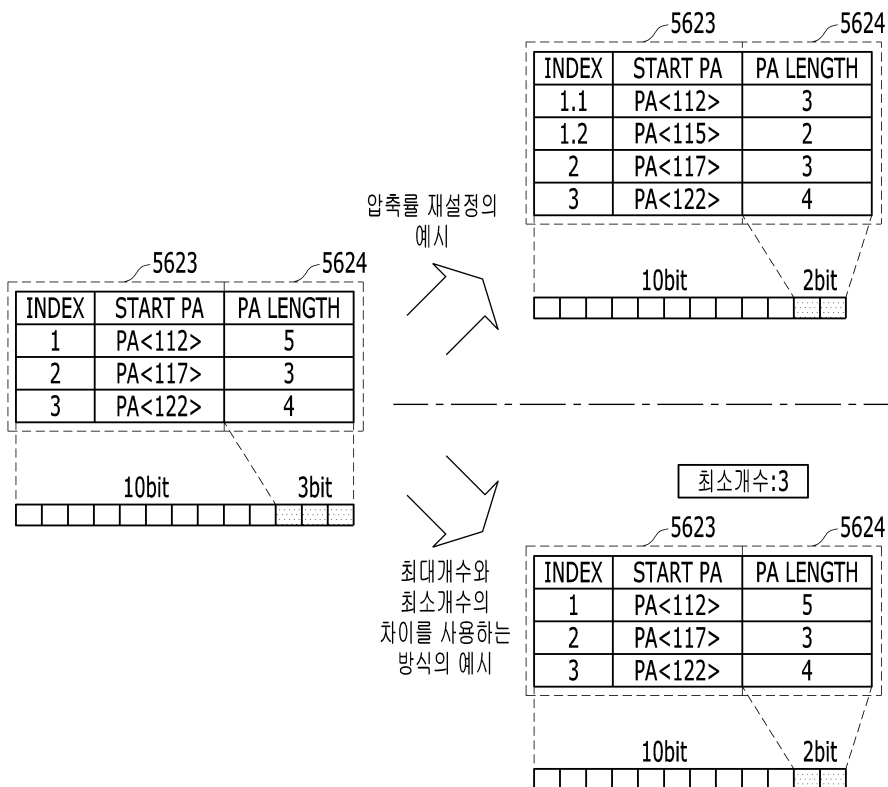
INDEX	LA	PA
1	LA<11>	PA<111>
2	LA<41>	PA<120>
3	LA<51>	PA<121>

INDEX	LA	PA
a	LA<421>	PA<7008>
b	LA<441>	PA<7011>
c	LA<451>	PA<7012>

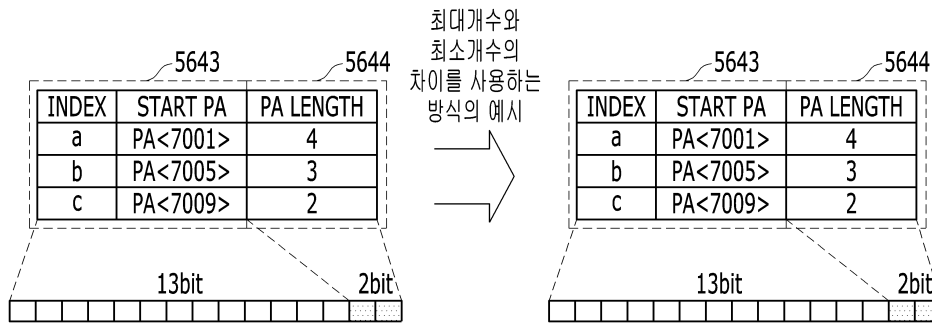
도면7b



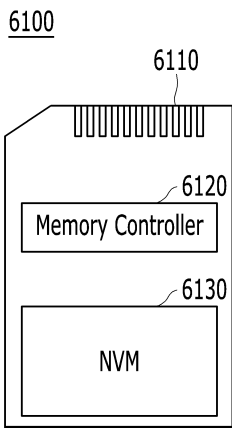
도면8a



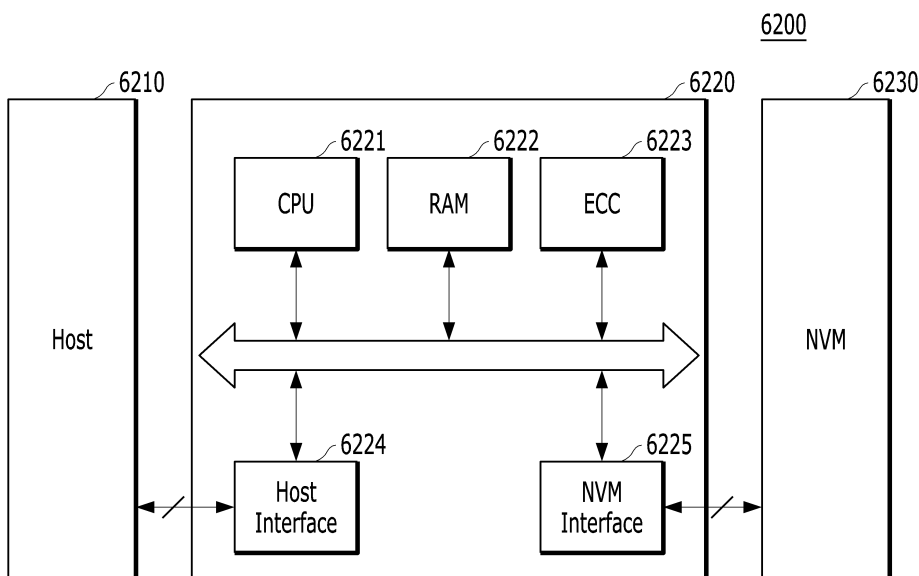
도면8b



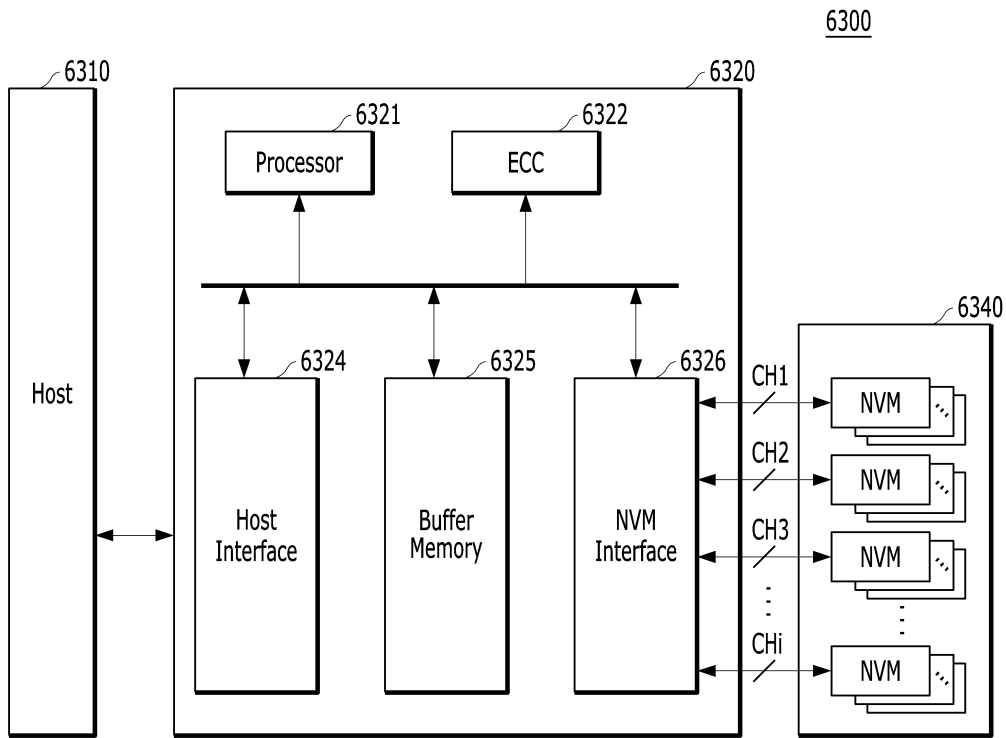
도면9



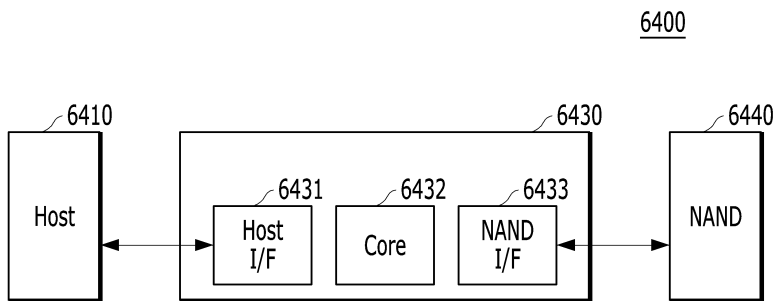
도면10



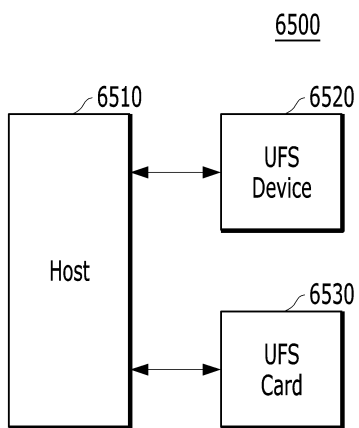
도면11



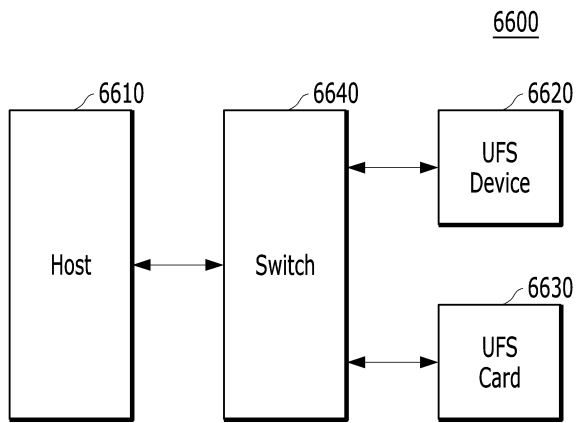
도면12



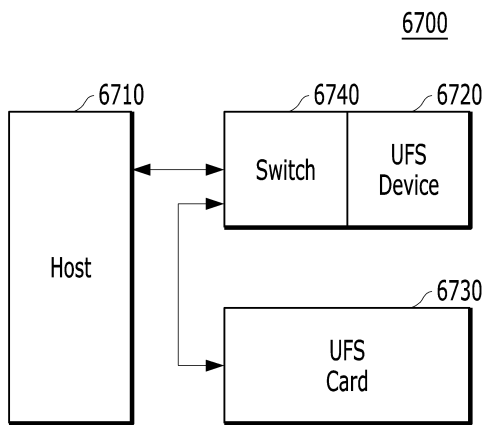
도면13



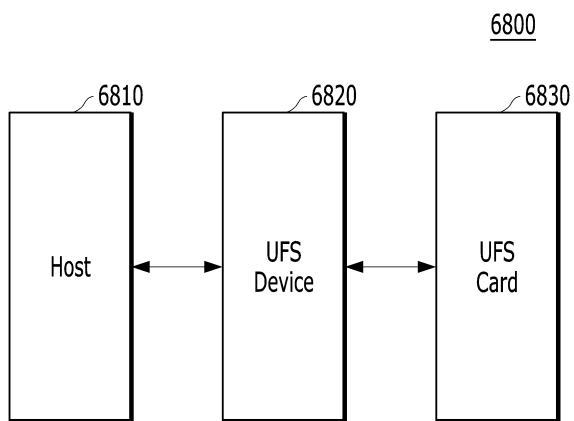
도면14



도면15



도면16



도면17

6900

