(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0063862
(43) 공개일자 2021년06월02일

(51) 국제특허분류(Int. Cl.)
G06F 16/22 (2019.01) *G06F 12/02* (2018.01)
(52) CPC특허분류
G06F 16/2282 (2019.01)
G06F 12/0253 (2013.01)
(21) 출원번호 10-2019-0152483
(22) 출원일자 2019년11월25일
심사청구일자 없음

(71) 출원인
에스케이하이닉스 주식회사
경기도 이천시 부발읍 경충대로 2091
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
이창규
서울특별시 마포구 마포대로 73(도화동) 813
김영재
서울특별시 서초구 잠원로12길 4(잠원동, 잠원현대아파트) 101-1808
(뒷면에 계속)
(74) 대리인
김선종

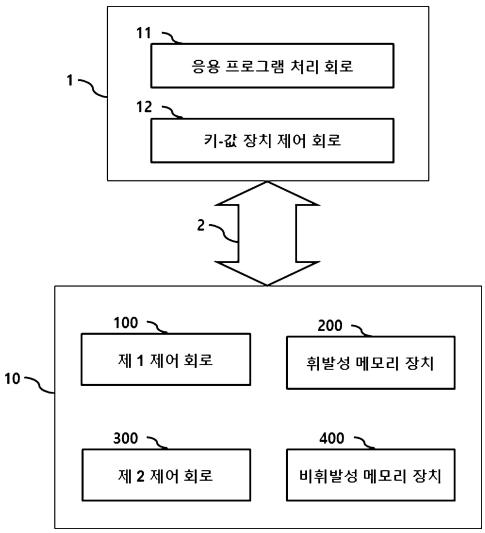
전체 청구항 수 : 총 21 항

(54) 발명의 명칭 **키-값 기반의 데이터 저장 장치 및 이를 포함하는 시스템**

(57) 요 약

본 기술에 의한 데이터 저장 장치는 키를 저장하는 영역과 값을 저장하는 영역을 포함하는 비휘발성 메모리 장치; 및 키-값 명령을 수신하여 값을 저장하는 영역에 값을 저장하고 키를 저장하는 영역에 값에 대응하는 키와 값의 주소를 함께 저장하는 동작을 제어하는 제 1 제어 회로를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 16/2246 (2019.01)
(72) 발명자
노정기
경기도 용인시 기흥구 보정로 87(보정동, 죽현마을아이파크) 208-1502
정우석
경기도 성남시 분당구 중앙공원로 53(서현동, 시범단지삼성.한신아파트) 130-1301
박경
대전광역시 서구 청사로 254(둔산동, 둥지아파트) 111-1202

명 세 서

청구범위

청구항 1

키를 저장하는 영역과 값을 저장하는 영역을 포함하는 비휘발성 메모리 장치; 및

키-값 명령을 수신하여 상기 값을 저장하는 영역에 값을 저장하고 상기 키를 저장하는 영역에 상기 값에 대응하는 키와 상기 값의 주소를 함께 저장하는 동작을 제어하는 제 1 제어 회로

를 포함하는 데이터 저장 장치.

청구항 2

청구항 1에 있어서, 키를 저장하는 영역을 포함하는 휘발성 메모리 장치를 더 포함하고,

상기 키-값 명령이 입력되는 경우 상기 제 1 제어 회로는 상기 휘발성 메모리 장치를 조회하여 대응하는 키가 존재하는지 판단하고 상기 휘발성 메모리 장치에 대응하는 키가 존재하지 않는 경우 상기 비휘발성 메모리 장치를 조회하여 대응하는 키가 존재하는지 판단하는 데이터 저장 장치.

청구항 3

청구항 2에 있어서, 상기 휘발성 메모리 장치에서 키를 저장하는 영역은 제 1 테이블을 저장하고, 상기 비휘발성 메모리 장치에서 키를 저장하는 영역은 다수의 제 2 테이블을 저장하되,

상기 제 1 테이블과 상기 다수의 제 2 테이블은 각각

값의 주소를 다수 개 저장하는 데이터 영역; 및

상기 데이터 영역에서 값의 주소가 위치하는 주소와 이에 대응하는 키의 쌍을 다수 개 저장하는 인덱스 영역

을 포함하는 데이터 저장 장치.

청구항 4

청구항 3에 있어서, 상기 제 1 테이블 및 상기 다수의 제 2 테이블은 각각 상기 인덱스 영역에 대응하는 키가 존재하는지 확인하는 필터 영역을 더 포함하는 데이터 저장 장치.

청구항 5

청구항 4에 있어서, 상기 제 1 제어 회로는 상기 제 1 테이블 및 상기 다수의 제 2 테이블은 LSM 트리 구조를 유지하도록 제어하는 데이터 저장 장치.

청구항 6

청구항 1에 있어서, 상기 비휘발성 메모리 장치에서 값을 저장하는 영역은 다수의 로그 엔트리를 포함하고, 상기 다수의 로그 엔트리 각각은 값과 값의 길이를 저장하는 데이터 저장 장치.

청구항 7

청구항 6에 있어서, 상기 비휘발성 메모리 장치의 물리 주소에 대응하는 논리 주소의 관계를 저장하는 맵핑 테이블을 더 포함하고,

상기 제 1 제어 회로는 상기 맵핑 테이블에서 프리 또는 무효 상태의 논리 주소를 포함하는 분산 로깅 연결 리스트를 관리하며,

상기 제 1 제어 회로는 로그 엔트리를 추가하는 경우 상기 분산 로깅 연결 리스트에 포함된 논리 페이지를 사용하는 데이터 저장 장치.

청구항 8

청구항 7에 있어서, 상기 비휘발성 메모리 장치에 대해서 가비지 콜렉션을 수행하는 제 2 제어 회로를 더 포함하고,

상기 가비지 콜렉션이 수행되는 경우 상기 제 1 제어 회로는 상기 분산 로깅 연결 리스트를 갱신하는 데이터 저장 장치.

청구항 9

청구항 1에 있어서,

실행 코드를 실행하여 결과를 생성하는 근접 데이터 처리 회로

를 더 포함하되,

상기 키-값 명령이 데이터 처리 명령인 경우 상기 제 1 제어 회로는 상기 키에 대응하는 값으로부터 추출되는 상기 실행 코드를 상기 근접 데이터 처리 회로에 제공하는 데이터 저장 장치.

청구항 10

청구항 9에 있어서, 상기 데이터 처리 명령은 파라미터를 더 포함하고, 상기 제 1 제어 회로는 상기 파라미터를 상기 근접 데이터 처리 회로에 추가로 제공하는 데이터 저장 장치.

청구항 11

키-값 명령을 생성하는 호스트;

상기 키-값 명령에 따라 읽기 또는 쓰기 동작을 수행하는 데이터 저장 장치; 및

상기 호스트와 상기 데이터 저장 장치 사이에서 상기 키-값 명령을 전달하는 인터페이스 회로

를 포함하되,

상기 데이터 저장 장치는

키를 저장하는 영역과 값을 저장하는 영역을 포함하는 비휘발성 메모리 장치; 및

상기 키-값 명령을 수신하여 상기 값을 저장하는 영역에 값을 저장하고 상기 키를 저장하는 영역에 상기 값에 대응하는 키와 상기 값의 주소를 함께 저장하는 동작을 제어하는 제 1 제어 회로

를 포함하는 시스템.

청구항 12

청구항 11에 있어서, 키를 저장하는 영역을 포함하는 휘발성 메모리 장치를 더 포함하고,

상기 키-값 명령이 입력되는 경우 상기 제 1 제어 회로는 상기 휘발성 메모리 장치를 조회하여 대응하는 키가 존재하는지 판단하고 상기 휘발성 메모리 장치에 대응하는 키가 존재하지 않는 경우 상기 비휘발성 메모리 장치를 조회하여 대응하는 키가 존재하는지 판단하는 시스템.

청구항 13

청구항 12에 있어서, 상기 휘발성 메모리 장치에서 키를 저장하는 영역은 제 1 테이블을 저장하고, 상기 비휘발성 메모리 장치에서 키를 저장하는 영역은 다수의 제 2 테이블을 저장하되,

상기 제 1 테이블과 상기 다수의 제 2 테이블은 각각

값의 주소를 다수 개 저장하는 데이터 영역; 및

상기 데이터 영역에서 값의 주소가 위치하는 주소와 이에 대응하는 키의 쌍을 다수 개 저장하는 인덱스 영역

을 포함하는 시스템.

청구항 14

청구항 13에 있어서, 상기 제 1 테이블 및 상기 다수의 제 2 테이블은 각각 상기 인덱스 영역에 대응하는 키가 존재하는지 확인하는 필터 영역을 더 포함하는 시스템.

청구항 15

청구항 14에 있어서, 상기 제 1 제어 회로는 상기 제 1 테이블 및 상기 다수의 제 2 테이블은 LSM 트리 구조를 유지하도록 제어하는 시스템.

청구항 16

청구항 11에 있어서, 상기 비휘발성 메모리 장치에서 값을 저장하는 영역은 다수의 로그 엔트리를 포함하고, 상기 다수의 로그 엔트리 각각은 값과 값의 길이를 저장하는 시스템.

청구항 17

청구항 16에 있어서, 상기 비휘발성 메모리 장치의 물리 주소에 대응하는 논리 주소의 관계를 저장하는 맵핑 테이블을 더 포함하고,

상기 제 1 제어 회로는 상기 맵핑 테이블에서 프리 또는 무효 상태의 논리 주소를 포함하는 분산 로깅 연결 리스트를 관리하며,

상기 제 1 제어 회로는 로그 엔트리를 추가하는 경우 상기 분산 로깅 연결 리스트에 포함된 논리 페이지를 사용하는 시스템.

청구항 18

청구항 17에 있어서, 상기 비휘발성 메모리 장치에 대해서 가비지 콜렉션을 수행하는 제 2 제어 회로를 더 포함하고,

상기 가비지 콜렉션이 수행되는 경우 상기 제 1 제어 회로는 상기 분산 로깅 연결 리스트를 갱신하는 시스템.

청구항 19

청구항 11에 있어서,

실행 코드를 실행하여 결과를 생성하는 근접 데이터 처리 회로

를 더 포함하되,

상기 키-값 명령이 데이터 처리 명령인 경우 상기 제 1 제어 회로는 상기 키에 대응하는 값으로부터 추출되는 상기 실행 코드를 상기 근접 데이터 처리 회로에 제공하는 시스템.

청구항 20

청구항 19에 있어서, 상기 데이터 처리 명령은 파라미터를 더 포함하고, 상기 제 1 제어 회로는 상기 파라미터를 상기 근접 데이터 처리 회로에 추가로 제공하는 시스템.

청구항 21

청구항 11에 있어서, 상기 호스트는

키에 대응하는 값을 읽거나 쓰는 동작을 요청하는 응용 프로그램을 처리하는 응용 프로그램 처리 회로; 및

상기 요청을 키-값 명령으로 변환하여 상기 인터페이스 회로를 통해 상기 데이터 저장 장치에 제공하는 키-값 장치 제어 회로

를 포함하는 시스템

발명의 설명

기 술 분 야

[0001] 본 기술은 키-값 기반의 데이터 저장 장치 및 이를 포함하는 시스템에 관한 것이다.

배 경 기 술

[0002] 종래의 데이터 저장 장치는 블록 단위로 데이터를 처리를 수행하므로 직접적으로 키-값 기반의 명령을 처리할 수 없는 문제가 있다.

[0003] 종래의 시스템에서는 호스트에서 운영하는 파일 시스템을 기반으로 키-값 기반의 명령을 블록 단위의 명령으로 변환하고 이를 데이터 저장 장치에 제공하여 키-값 기반의 데이터 처리 동작을 수행한다.

[0004] 이에 따라 호스트에서 키-값 기반의 응용 프로그램을 처리하는 경우 호스트는 응용 프로그램에서 제공되는 키-값 기반의 명령을 블록 단위의 명령으로 변환하는 동작이 선행되어야 하므로 처리 성능을 향상시키는데 한계가 있다.

[0005] 또한 호스트에서 운영하는 파일 시스템에 따라 데이터를 관리해야 하므로 데이터 저장 공간을 효율적으로 관리하는데 한계가 있다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) US 20180089274 A1

(특허문헌 0002) US 20180225315 A1

비특허문헌

[0007] (비특허문헌 0001) L. Lu, T. S. Pillai, A. C. Arpaci-Dusseau, and R. H. Arpaci- Dusseau, “WiscKey: Separating keys from values in SSDconscious storage,” in Proceedings of the 14th USENIX Conference on File and Storage Technologies, ser. FAST ’ 16, 2016.

발명의 내용

해결하려는 과제

[0008] 본 기술은 파일 시스템에 기초하지 않고 키-값 기반의 명령을 직접 처리할 수 있는 데이터 저장 장치와 이를 포함하는 시스템을 제공한다.

과제의 해결 수단

[0009] 본 발명의 일 실시예에 의한 데이터 저장 장치는 키를 저장하는 영역과 값을 저장하는 영역을 포함하는 비휘발성 메모리 장치; 및 키-값 명령을 수신하여 값을 저장하는 영역에 값을 저장하고 키를 저장하는 영역에 값에 대응하는 키와 값의 주소를 함께 저장하는 동작을 제어하는 제 1 제어 회로를 포함한다.

[0010] 본 발명의 일 실시예에 의한 시스템은 키-값 명령을 생성하는 호스트; 키-값 명령에 따라 읽기 또는 쓰기 동작을 수행하는 데이터 저장 장치; 및 호스트와 데이터 저장 장치 사이에서 키-값 명령을 전달하는 인터페이스 회로를 포함하되, 데이터 저장 장치는 키를 저장하는 영역과 값을 저장하는 영역을 포함하는 비휘발성 메모리 장치; 및 키-값 명령을 수신하여 값을 저장하는 영역에 값을 저장하고 키를 저장하는 영역에 상기 값에 대응하는 키와 값의 주소를 함께 저장하는 동작을 제어하는 제 1 제어 회로를 포함한다.

발명의 효과

[0011] 본 기술에 의한 데이터 저장 장치는 파일 시스템에 무관하게 키-값 기반의 명령을 독자적으로 처리하여 저장 공간을 효율적으로 사용할 수 있다.

[0012] 본 기술에 의한 시스템은 호스트에서 키-값 기반의 명령을 블록 단위의 명령으로 변환하는 동작을 수행하지 않

으므로 처리 속도를 향상시킬 수 있다.

도면의 간단한 설명

[0013] 도 1은 본 발명의 일 실시예에 의한 시스템의 블록도.

도 2는 본 발명의 일 실시예에 의한 키-값 명령의 구조를 나타내는 테이블.

도 3은 본 발명의 일 실시예에 의한 데이터 저장 장치를 나타내는 블록도.

도 4는 본 발명의 일 실시예에 의한 제 2 테이블의 데이터 구조를 나타내는 블록도.

도 5는 본 발명의 일 실시예에 의한 값 로그 영역과 로그 엔트리의 구조를 나타내는 블록도.

도 6은 본 발명의 일 실시예에 의한 데이터 저장 장치에서 PUT 명령의 처리 동작을 나타내는 설명도.

도 7은 본 발명의 일 실시예에 의한 데이터 저장 장치에서 GET 명령의 처리 동작을 나타내는 설명도.

도 8은 본 발명의 일 실시예에 의한 데이터 저장 장치에서 DELETE 명령의 처리 동작을 나타내는 설명도.

도 9는 본 발명의 일 실시예에 의한 데이터 저장 장치의 로그 클리닝 동작을 나타내는 설명도.

도 10은 본 발명의 일 실시예에 의한 데이터 저장 장치의 분산 로깅 동작을 나타내는 설명도.

도 11은 본 발명의 일 실시예에 의한 데이터 저장 장치의 분산 로깅 연결 리스트를 나타내는 설명도.

도 12는 본 발명의 일 실시예에 의한 데이터 저장 장치를 나타내는 블록도.

도 13은 본 발명의 일 실시예에 의한 근접 데이터 처리 동작을 위한 키-값 명령의 구조를 나타내는 도면.

발명을 실시하기 위한 구체적인 내용

[0014] 이하에서는 첨부한 도면을 참조하여 본 발명의 실시예를 개시한다.

[0015] 도 1은 본 발명의 일 실시예에 의한 시스템의 블록도이다.

[0016] 본 발명의 일 실시예에 의한 시스템은 호스트(1), 인터페이스 회로(2) 및 데이터 저장 장치(10)를 포함한다.

[0017] 인터페이스 회로(2)는 호스트(1)와 데이터 저장 장치(10) 사이에서 명령 신호와 데이터 신호를 송수신하기 위하여 동작하는 하드웨어, 소프트웨어 및 이들의 조합을 포함한다.

[0018] 본 실시예에서 인터페이스 회로(2)는 PCI 익스프레스(PCIe) 규격을 따르는 것이나 이에 한정되는 것은 아니다.

[0019] 데이터 저장 장치(10)는 키-값 명령을 수신하고 이에 따라 키-값 기반의 데이터 읽기/쓰기 동작을 수행한다.

[0020] 데이터 저장 장치(10)는 제 1 제어 회로(100), 휘발성 메모리 장치(200), 제 2 제어 회로(300) 및 비휘발성 메모리 장치(400)를 포함한다.

[0021] 제 1 제어 회로(100)는 키-값 명령을 처리하기 위한 데이터 구조를 관리할 수 있다.

[0022] 제 1 제어 회로(100)는 키-값 명령을 처리하기 위하여 휘발성 메모리 장치(200), 제 2 제어 회로(300) 및 비휘발성 메모리 장치(400)를 제어할 수 있다.

[0023] 본 실시예에서 휘발성 메모리 장치(200)는 디램을 포함하나 이에 한정되는 것은 아니다.

[0024] 이하에서는 휘발성 메모리 장치(200)를 디램, DRAM 등으로 지칭할 수 있다.

[0025] 또한 본 실시예에서 비휘발성 메모리 장치(400)는 낸드 플래시 메모리 장치를 포함하나 이에 한정되는 것은 아니다.

[0026] 이하에서는 비휘발성 메모리 장치(400)를 플래시 메모리 장치로 지칭할 수 있다.

[0027] 또한 본 실시예에서 제 2 제어 회로(300)는 비휘발성 메모리 장치(400)에 데이터를 읽고 쓰는 동작을 제어한다.

[0028] 예를 들어 제 2 제어 회로(300)는 종래의 SSD(Solid State Drive)에서 FTL(Flash Translation Layer)에 대응하는 것으로서 맵핑 테이블을 관리하는 동작, 가비지 콜렉션 동작을 제어하는 동작을 수행할 수 있다.

[0029] 이하에서는 제 2 제어 회로(300)를 FTL로 지칭할 수 있다.

[0030] 호스트(1)는 시스템 프로그램 또는 응용 프로그램의 동작 중에 생성되는 키-값 기반의 동작 요청에 따라 키-값 명령을 생성하여 인터페이스 회로(2)를 통해 데이터 저장 장치(10)에 제공한다.

[0031] 호스트(1)는 프로세서를 포함하여 응용 프로그램에 의한 동작을 처리하는 하드웨어, 소프트웨어 또는 이들의 조합을 포함할 수 있다.

[0032] 호스트(1)는 응용 프로그램 처리 회로(11)와 키-값 장치 제어 회로(12)를 포함한다.

[0033] 응용 프로그램 처리 회로(11)는 응용 프로그램을 수행하기 위한 하드웨어, 소프트웨어 또는 이들의 조합에 대응하는 구성이다.

[0034] 응용 프로그램은 키-값 처리를 위한 API 라이브러리와 함께 컴파일되어 기계어로 변환되고 응용 프로그램 처리 회로(11)는 컴파일된 기계어 명령을 처리할 수 있다.

[0035] 키-값 처리를 위한 API 라이브러리는 GET, PUT, DELETE 등의 키-값 요청을 지원할 수 있다.

[0036] 이때 GET은 읽기 동작, PUT은 쓰기 동작, DELETE는 삭제(소거) 동작에 대응한다.

[0037] 키-값 장치 제어 회로(12)는 응용 프로그램 처리 회로(11)에서 키-값 요청을 제공하는 경우 이에 대응하는 키-값 명령을 생성하여 인터페이스 회로(2)에 제공한다.

[0038] 본 실시예는 PCI 익스프레스(PCIe) 규격을 따르는 인터페이스 회로(1)를 통해 비휘발성 메모리를 사용하는 데이터 저장 장치(100)가 연결되어 사용되는 구조이다.

[0039] 이에 따라 본 실시예에서 키-값 장치 제어 회로(12)는 NVM 익스프레스(Nonvolatile Memory Express: NVMe) 프로토콜을 확장한 키-값 명령을 생성할 수 있다.

[0040] 또한 키-값 장치 제어 회로(12)는 생성된 키-값 명령을 포함하는 PCIe 명령을 생성하고 이를 인터페이스 회로(2)에 제공하여 키-값 명령이 데이터 저장 장치(10)에 전달되도록 할 수 있다.

[0041] 도 2는 본 발명의 일 실시예에 의한 키-값 명령의 구조를 나타내는 테이블이다.

[0042] 본 발명의 일 실시예에 의한 키-값 명령은 NVMe 명령을 확장한 형태로 볼 수 있다.

[0043] 도 2에서 (A)는 PUT 명령, (B)는 GET 명령, (C)는 DELETE 명령을 나타낸다.

[0044] 전술한 바와 같이 PUT 명령은 쓰기 동작, GET 명령은 읽기 동작, DELETE 명령은 소거 동작에 대응한다.

[0045] 도 2의 각 명령에서 CommandID, NamespaceID 필드는 NVMe 프로토콜에 존재하는 것으로서 각각 호스트에서 송신한 명령과 명령의 수신자를 식별하기 위한 정보를 나타낸다.

[0046] 나머지는 NVMe 프로토콜에 존재하는 OpCode, LBA 시작 주소, 및 유보 공간을 키-값 명령에 맞게 다시 정의한 것이다.

[0047] 나머지 필드의 의미는 다음과 같다.

[0048] OpCode 필드는 해당 명령을 받는 장치가 수행할 동작을 나타낸다.

[0049] PageList 필드는 호스트에서 송신 또는 수신할 때 데이터를 저장할 메모리 페이지 주소의 리스트를 나타낸다.

[0050] Key 필드는 PUT, GET, DELETE 명령을 수행할 대상이 되는 키를 나타낸다.

[0051] KV Length는 저장할 데이터의 총 길이를 나타낸다.

[0052] Buffer Size 필드는 GET 명령을 수행할 때 호스트가 해당 명령으로 받을 값을 저장하기 위해 준비한 메모리 크기를 나타낸다.

[0053] 도 3은 본 발명의 일 실시예에 의한 데이터 저장 장치(10)를 나타내는 블록도이다.

[0054] 본 실시예에서 데이터 저장 장치(10)는 제 1 제어 회로(100)를 더 포함하여 인터페이스 회로(2)로부터 전달된 키-값 명령을 처리하기 위하여 휘발성 메모리 장치(200), 제 2 제어 회로(300), 비휘발성 메모리 장치(400)를 제어한다.

[0055] 이때 제 1 제어 회로(100)에서 키-값 명령을 해석하여 논리 주소에 대한 읽기/쓰기 명령을 생성하고 이를 이용하여 휘발성 메모리 장치(200), 제 2 제어 회로(300) 및 비휘발성 메모리 장치(400)를 제어하여 할 수 있으며

이러한 읽기/쓰기 동작 자체는 종래의 SSD에서 수행되는 동작과 실질적으로 동일한 것이므로 이에 대해서는 구체적인 설명을 생략한다.

[0056] 본 실시예에서 제 1 제어 회로(100)는 키-값 명령을 처리하기 위하여 키와 값을 구별하여 관리한다.

[0057] 제 1 제어 회로(100)는 키를 관리하기 위하여 데이터 구조를 사용하는데 본 실시예에서는 LSM 트리(Log-Structured Merge Tree) 구조를 사용한다.

[0058] LSM 트리 구조는 제 1 테이블과 제 2 테이블을 이용하여 키를 검색하기 위한 데이터 구조이다.

[0059] 이하에서는 제 1 테이블을 MemTable, 제 2 테이블을 SSTable로 지칭할 수도 있다.

[0060] 제 1 테이블과 제 2 테이블은 기본적으로 키와 값 로그 옵셋을 저장하여 키와 키에 대응하는 값이 저장된 주소를 관리한다.

[0061] 이와는 별도로 제 1 제어 회로(100)는 값을 저장하기 위한 값 로그 영역을 관리한다.

[0062] 휘발성 메모리 장치(200)는 제 1 테이블 저장 영역(210), 제 2 테이블 캐시 영역(220), 메타 데이터 캐시 영역(230), 맵핑 테이블 저장 영역(240), 논리 주소 정보 저장 영역(250), 값 로그 버퍼 영역(260)을 포함한다.

[0063] 제 1 테이블 저장 영역(210)은 LSM 트리 구조에서 사용하는 제 1 테이블이 저장되는 영역이다.

[0064] 제 2 테이블 캐시 영역(220)은 제 1 테이블 저장 영역(210)에 저장된 제 1 테이블로부터 제 2 테이블을 생성하여 이를 임시 저장하는 영역이다.

[0065] 이에 따라 제 1 테이블은 제 2 테이블과 실질적으로 동일한 데이터 구조를 가질 수 있다.

[0066] 제 2 테이블의 데이터 구조에 대해서는 아래에서 구체적으로 개시한다.

[0067] 메타 데이터 캐시 영역(230)은 비휘발성 메모리 장치(400)에 포함된 메타 데이터 저장 영역(410)을 캐시하여 저장하는 영역이다.

[0068] 메타 데이터 저장 영역(410)에 저장되는 정보에 대해서는 아래에서 구체적으로 개시한다.

[0069] 테이블 저장 영역(240)은 논리 주소와 물리 주소 사이의 관계를 관리하는 맵핑 테이블을 저장하는 영역이다.

[0070] 본 실시예에서는 페이지 주소 단위의 맵핑을 사용하는 것으로 가정하나 반드시 이에 한정되는 아니다. 이하에서 논리 주소는 논리 페이지 주소, 물리 주소는 물리 페이지 주소로 지칭될 수 있다.

[0071] 논리 주소 정보 저장 영역(250)은 용도에 따라 전체 논리 주소를 다수의 구간으로 나누어 관리하는 경우 각 영역의 주소를 식별하기 위한 정보를 저장한다.

[0072] 본 실시예에서 전체 논리 주소는 메타 데이터 저장 구간, 제 2 테이블 저장 구간, 값 로그 저장 구간으로 구분된다.

[0073] 이들은 비휘발성 메모리 장치(400)에 포함되는 메타 데이터 저장 영역(410), 제 2 테이블 저장 영역(420), 값 로그 영역(430)에 대응한다.

[0074] 본 실시예에서 각 구간의 크기는 제 2 테이블의 최대 개수, 키와 값의 크기 등을 참조하여 미리 결정되나 이에 한정되는 것은 아니다.

[0075] 논리 주소 정보 저장 영역(250)은 이들 다수의 구간의 주소 정보를 저장할 수 있다.

[0076] 값 로그 버퍼 영역(260)은 인터페이스 회로(2)를 통해 제공된 값을 임시로 저장하는 공간이다.

[0077] 휘발성 메모리 장치(200)는 하나 또는 둘 이상의 메모리 칩을 포함할 수 있으며 이에 따라 제 1 테이블 저장 영역(210), 제 2 테이블 캐시 영역(220), 메타 데이터 캐시 영역(230), 맵핑 테이블 저장 영역(240), 논리 주소 정보 저장 영역(240), 값 로그 버퍼 영역(260)이 하나의 메모리 칩에 모두 포함될 수도 있고 서로 다른 메모리 칩에 다양한 방식으로 나뉘어 포함될 수도 있다.

[0078] 제 1 제어 회로(100)는 키-값 명령 처리 동작 과정에서 휘발성 메모리 장치(200)의 각 영역에 접근하여 정보를 추가하거나 수정하는 등의 동작을 병행할 수 있다.

[0079] 구체적인 키-값 명령 처리 동작에 대해서는 아래에서 다시 구체적으로 개시한다.

[0080] 제 2 제어 회로(300)는 맵핑 테이블 관리 회로(310)와 가비지 콜렉션 관리 회로(320)를 포함한다.

[0081] 맵핑 테이블 관리 회로(310)는 맵핑 테이블 저장 영역(240)에 저장된 논리 주소와 물리 주소 사이의 관계를 관리한다.

[0082] 가비지 콜렉션 관리 회로(320)는 가비지 콜렉션 동작을 관리하며 가비지 콜렉션 동작 과정에서 유효 페이지 데이터를 이동하여 저장할 수 있으며 이 과정에서 논리 주소와 물리 주소 사이의 관계를 조정할 수 있다.

[0083] 비휘발성 메모리 장치(400)는 메타 데이터 저장 영역(410), 제 2 테이블 저장 영역(420), 값 로그 영역(430)을 포함한다.

[0084] 제 2 테이블 저장 영역(420)은 키를 관리하기 위한 영역으로서 본 실시예에서는 LSM 트리 구조에 따라 제 2 테이블을 저장한다.

[0085] 제 1 테이블 저장 영역(210)에 저장되어 있던 제 1 테이블이 플러시 되는 경우 제 2 테이블로 변환되면서 제 2 테이블 캐시 영역(220)에 임시로 저장된 후 제 2 테이블 저장 영역(420)에 저장될 수 있다.

[0086] 본 실시예에서 키를 관리하기 위해서 휘발성 메모리 장치(200)의 제 1 테이블 저장 영역(210)과 비휘발성 메모리 장치(400)의 제 2 테이블 저장 영역(420)이 함께 활용될 수 있다.

[0087] 도 4는 제 2 테이블(421)을 나타내는 데이터 구조이다.

[0088] 제 2 테이블(421)은 블룸필터 영역(4211), 인덱스 영역(4212) 및 데이터 영역(4213)을 포함하며 각 영역은 다수의 페이지를 포함하고 페이지 단위로 정렬된다.

[0089] 블룸필터 영역(4211)은 블룸필터를 저장한다.

[0090] 인덱스 영역(4212)은 키와 옵셋으로 구성된 다수의 키-옵셋 쌍을 리스트 형태로 저장한다.

[0091] 인덱스 영역(4212)에 포함된 다수의 키-옵셋 쌍은 키 순서대로 정렬될 수 있으며 이 경우 이진 검색을 사용하여 신속한 검색을 수행할 수 있다.

[0092] 데이터 영역(4213)은 로그 옵셋 데이터를 리스트 형태로 저장한다.

[0093] 블룸필터는 입력된 키에 대응하는 키가 인덱스 영역(4212)에 존재하는지를 "false positive" 방식으로 판단한다.

[0094] 예를 들어 블룸필터에서 키가 존재하지 않는다고 판단하면 해당 키는 인덱스 영역(4212)에 확실히 존재하지 않거나 키가 존재한다고 판단하면 인덱스 영역(4212)에 해당 키가 실제로 존재하는지 검색해야 한다.

[0095] 인덱스 영역(4212)에서 키가 검색되면 키와 쌍을 이루는 옵셋을 이용하여 데이터 영역(4213)을 조회하여 대응하는 로그 옵셋 데이터를 확인할 수 있다.

[0096] 값 로그 영역(430)은 키에 대응하는 값을 저장하는 영역이다.

[0097] 도 5는 값 로그 영역(430)의 데이터 구조를 나타낸다.

[0098] 값 로그 영역(430)은 로그 테일 포인터와 로그 헤드 포인터를 이용하여 로그 엔트리(431)를 저장한다.

[0099] 로그 테일 포인터는 최초로 저장된 로그 엔트리를 지시하고 로그 헤드 포인터는 새로 입력될 로그 엔트리가 저장될 위치를 지시한다.

[0100] 새로운 로그 엔트리(431)가 로그 헤드 포인터 위치에 저장되며 로그 헤드 포인터는 그 다음 빈 곳으로 이동하여 새로운 로그 엔트리는 항상 마지막에 순차적으로 추가된다.

[0101] 로그 엔트리(431)는 값 길이 영역(4311), 값 영역(4312)을 포함한다.

[0102] 값 길이 영역(4311)은 값 데이터의 길이를 저장하고 값 영역(4312)은 값 데이터를 저장한다.

[0103] 본 실시예에서 로그 엔트리(431)는 값 정보를 저장하는 것으로 충분하나 키 정보를 추가로 저장할 수도 있다.

[0104] 키 정보를 추가로 저장하기 위하여 키 길이 영역(4313), 키 영역(4314)이 추가로 로그 엔트리(431)에 포함될 수 있다.

[0105] 키 길이 영역(4313)은 키의 길이를 저장하고 키 영역(4314)은 키 자체를 저장한다.

[0106] 로그 엔트리(431)는 페이지 단위로 형성된 후 로그 헤드 포인터 위치에 추가될 수 있다.

[0107] 메타 데이터 저장 영역(410)은 제 2 테이블에 대한 메타 데이터, 로그 헤드 포인터 등을 저장할 수 있다.

[0108] 개별 제 2 테이블에 대한 메타 데이터는 제 2 테이블의 ID, 제 2 테이블에 포함된 최대 키와 최소 키, 제 2 테이블의 시작 논리 주소, 블룸필터 영역(4211)에 포함된 페이지 수, 인덱스 영역(4212)에 포함된 페이지 수, 데이터 영역(4213)의 페이지 수 등을 포함한다.

[0109] 본 실시예에서 다수의 제 2 테이블은 멀티레벨 리스트 구조를 이용하여 LSM 트리 구조를 구현할 수 있다.

[0110] 이에 따라 멀티레벨 리스트 구조에 대한 메타 데이터로서 리스트 전체에 대한 헤드 포인터, 레벨 별 헤드 포인터를 포함할 수 있다.

[0111] 로그 헤드 포인터는 제 2 테이블 로그 헤드 포인터와 값 로그 헤드 포인터를 포함한다.

[0112] 제 2 테이블 로그 헤드 포인터는 제 2 테이블 저장 영역(420)에서 추가로 제 2 테이블을 저장하기 위한 논리 주소, 제 2 테이블 저장 영역(420)의 시작 논리 주소와 마지막 논리 주소를 포함할 수 있다.

[0113] 값 로그 헤드 포인터는 값 로그 영역(430)에 새로운 로그 엔트리(431)를 저장하기 위한 논리 주소, 값 로그 영역(430)의 시작 논리 주소와 마지막 논리 주소를 포함할 수 있다.

[0114] 도 6은 본 발명의 일 실시예에 의한 제 1 제어 회로(100)에서 PUT 명령을 처리하는 동작을 설명하는 도면이다.

[0115] PUT 명령이 입력되면 제 1 제어 회로(100)는 PUT 명령에서 키와 값을 분리하여 키 정보는 제 1 테이블, 제 2 테이블을 이용하여 관리하고 값은 값 로그 영역(430)에 저장하게 된다.

[0116] 종래에는 호스트 운영체제에서 지원하는 파일 시스템을 이용하여 키, 값 정보를 관리함에 따라 키, 값 정보가 파일로 기록되고 이에 따라 데이터 저장 공간을 낭비하였다.

[0117] 본 발명에서는 페이지 단위를 기준으로 키, 값 정보를 관리하여 데이터 저장 공간의 낭비를 방지할 수 있다.

[0118] 제 1 제어 회로(100)는 키와 값을 이용하여 새로운 로그 엔트리(431)를 생성하여 값 로그 영역(430)에 추가한다.

[0119] 이때 제 1 제어 회로(100)는 값 로그 헤드 포인터를 참조하여 새로운 값 로그 엔트리(431)가 추가된 위치에 대응하는 값 로그 옵셋 정보를 추출할 수 있다.

[0120] 값 로그 헤드 포인터는 메타 데이터 캐시 영역(230)을 참조하여 알 수 있다.

[0121] 제 1 제어 회로(100)는 입력된 키와 값 로그 옵셋을 제 1 테이블 저장 영역(210)의 제 1 테이블에 저장한다.

[0122] 이때 제 1 테이블이 가득 차면 제 1 테이블을 플러시하여 제 2 테이블(221)을 생성하고 이를 제 2 테이블 저장 영역(420)에 추가한다.

[0123] 제 2 테이블(221)을 제 2 테이블 저장 영역(420)에 추가하는 동작은 LSM 트리 구조의 규칙을 유지하면서 수행된다.

[0124] LSM 트리 구조의 규칙을 유지하면서 새로운 제 2 테이블을 추가하는 동작은 잘 알려진 것이므로 구체적인 설명을 생략한다.

[0125] 제2 테이블 저장 영역(420)에 제 2 테이블(421)을 저장하는 동작과 값 로그 영역(430)에 로그 엔트리(431)를 저장하는 동작은 제 1 제어 회로(100)에서 대응하는 주소에 대한 쓰기 명령을 통해 휘발성 메모리 장치(200)에 저장된 맵핑 테이블, 제 2 제어 회로(300)를 제어하여 수행될 수 있으며 이러한 쓰기 동작 자체는 종래의 SSD에서 수행되는 것과 실질적으로 동일하다.

[0126] PUT 동작에 따라 값 로그 헤드 포인터 등의 메타 데이터가 변경되는 것은 자명하며 이에 따라 메타 데이터 저장 영역(410)과 메타 데이터 캐시 영역(230)의 정보가 갱신되어야 하는 것 역시 자명하므로 설명을 생략한다.

[0127] 도 7은 본 발명의 일 실시예에 의한 제 1 제어 회로(100)에서 GET 명령을 처리하는 동작을 설명하는 도면이다.

[0128] GET 명령은 읽기 명령에 대응하는 동작으로서 키 데이터를 이용하여 제 1 테이블 저장 영역(210)에 저장된 제 1 테이블에 대응하는 키가 존재하는지 판단한다.

[0129] 제 1 테이블에 대응하는 키가 존재하지 않는다면 제 2 테이블 저장 영역(420)을 추가로 검색하여 대응하는 키가

존재하는지 판단한다.

[0130] 키가 존재하는지는 키와 블룸필터를 이용하여 수행할 수 있으며 키가 존재하는 것으로 판단한 경우 인덱스 영역에 키가 존재하는지 다시 검사하게 된다.

[0131] 대응하는 키가 존재하면 키에 대응하는 값 로그 옵셋을 찾고 이를 이용하여 값 로그 영역에서 값을 읽어 외부로 제공한다.

[0132] 이때 제2 테이블 저장 영역(420)에서 값 로그 옵셋을 읽는 동작이나 값 로그 영역(430)에서 값을 읽는 동작은 제 1 제어 회로(100)에서 대응하는 주소에 대한 읽기 명령을 통해 휘발성 메모리 장치(200)에 저장된 맵핑 테이블, 제 2 제어 회로(300)를 제어하여 수행될 수 있으며 이러한 읽기 동작 자체는 종래의 SSD에서 수행되는 것과 실질적으로 동일하다.

[0133] 도 8은 본 발명의 일 실시예에 의한 제 1 제어 회로(100)에서 DELETE 명령을 처리하는 동작을 설명하는 도면이다.

[0134] 본 실시예에서 DELETE 명령은 PUT 명령과 실질적으로 동일하다.

[0135] 대신 PUT 명령에서는 실제 값이 기록되지만 DELETE 명령에서는 해당 키가 삭제되었음을 표시하는 미리 정해진 형태의 소거 값이 기록된다.

[0136] 이에 따라 DELETE 명령에 대한 구체적인 동작을 반복하여 설명하지는 않는다.

[0137] DELETE 명령이 수행되더라도 이전에 기록된 키, 값이 여전히 남아 있게 된다.

[0138] 이에 따라 예를 들어 이전에 기록된 키가 제 2 테이블 저장 영역(420)에 남아 있는 경우 키에 대응하는 값 로그 옵셋을 찾아 값 로그 영역(430)에 있는 기존 값을 무효화시킨 후 값 로그 옵셋을 무효화시키는 동작을 함께 수행할 수 있다.

[0139] 값 로그 영역(430)에 무효가 된 공간이 발생하면 저장 공간이 부족해지므로 로그 클리닝 동작을 수행할 수 있다.

[0140] 도 9는 로그 클리닝 동작을 설명하는 도면이다.

[0141] 도 9에서 숫자는 논리 주소, F는 프리 페이지, I는 무효 페이지, V는 유효 페이지를 나타낸다.

[0142] 로그 클리닝 동작은 가비지 콜렉션 동작과 유사하게 유효한 페이지들을 이동하여 저장함으로써 무효 페이지들이 동일한 블록에 위치하도록 한다.

[0143] 도 9(A)는 2번 페이지를 로그 헤드 포인터가 지시하는 6번의 프리 페이지 위치로 이동하는 모습을 나타낸다.

[0144] 이후 블록을 소거하여 1, 2, 3번의 프리 페이지를 생성하게 된다.

[0145] 이와 같이 생성된 프리 페이지에 새로운 로그 엔트리를 기록할 수 있다.

[0146] 로그 클리닝 과정에서 로그 테일 포인터는 기록된 지 가장 오래된 유효 페이지인 4번 페이지를 지시하고 로그 헤드 포인터는 새로 생성된 프리 페이지인 1번 페이지를 지시하도록 갱신된다.

[0147] 다른 실시예에서는 분산 로깅 기법을 사용할 수 있다.

[0148] 분산 로깅 기법은 무효가 된 논리 페이지에 대해서도 새로운 값을 기록하는 방식으로서 로그 클리닝을 빈번하게 수행할 필요가 없다.

[0149] 분산 로깅은 로그 엔트리의 크기가 1 페이지 이하인 경우에 적절하다.

[0150] 도 10은 본 발명의 일 실시예에 의한 분산 로깅 기법을 설명하는 도면이다.

[0151] 도 10에서 숫자는 논리 주소, F는 프리 페이지, I는 무효 페이지, V는 유효 페이지를 나타낸다.

[0152] 분산 로깅 시에는 무효인 논리 페이지에 대해서도 새로운 로그 엔트리를 기록한다.

[0153] 이를 위하여 본 실시예에서는 프리이거나 무효인 논리 페이지를 연결 리스트를 이용하여 관리한다.

[0154] 맵핑 테이블에서 프리이거나 무효인 논리 페이지에 대응하는 물리 페이지는 의미 없는 정보를 포함하므로 본 실시예에서는 해당 공간을 이용하여 연결 리스트로 관리할 수 있다.

[0155] 이하에서는 이를 분산 로깅 연결 리스트로 지칭하고 프리이거나 무효인 논리 페이지를 분산 로깅 페이지로 지칭한다.

[0156] 도 11은 분산 로깅 연결 리스트를 관리하는 방법을 설명하는 도면이다.

[0157] HEAD 포인터는 첫 번째 분산 로깅 페이지를 지시하고 TAIL 포인터는 마지막 분산 로깅 페이지를 지시한다.

[0158] 분산 로깅 페이지에 대응하는 물리 페이지 번호는 다음의 분산 로깅 페이지 번호를 표시한다.

[0159] 도 11에서 첫 번째 분산 로깅 페이지는 0번이고 마지막 분산 로깅 페이지는 23번이다.

[0160] 0번 논리 페이지에 대응하는 물리 페이지 번호는 2번으로 기록되어 있는데 이는 다음 분산 로깅 페이지 번호가 2번임을 나타낸다.

[0161] 이러한 방식으로 마지막 분산 로깅 페이지까지 연결 리스트로 관리할 수 있다.

[0162] 본 실시예에서 가비지 콜렉션을 통해 새로운 프리 또는 무효 페이지가 추가되는 경우 TAIL 포인터를 갱신한다.

[0163] 다만 로그 엔트리의 크기가 한 페이지를 초과하는 경우 연속된 분산 로깅 페이지가 존재하지 않으면 로그 클리닝을 수행할 필요가 있다.

[0164] 분산 로깅 연결 리스트의 유지 관리는 제 1 제어 회로(100)에서 수행될 수 있다.

[0165] 본 발명의 다른 실시예에 의한 데이터 저장 장치(10-1)는 데이터 근접 프로세싱(NDP: Near Data Processing)을 지원할 수 있다.

[0166] 도 12는 본 발명의 일 실시예에 의한 데이터 저장 장치(10-1)를 나타내는 블록도이다.

[0167] 도 12의 데이터 저장 장치(10-1)는 도 1의 데이터 저장 장치(10)와 실질적으로 동일하나 NDP 회로(500)를 더 포함한다.

[0168] 도면에서는 NDP 회로(500)를 제 1 제어 회로(100)와 별도의 구성으로 표시하였으나 NDP 회로(500)는 제 1 제어 회로(100)에 포함될 수도 있다.

[0169] 본 실시예에서는 NDP 동작을 위한 실행 코드를 키-값 명령을 통해 미리 데이터 저장 장치(10-1)에 저장한다.

[0170] 이는 전술한 PUT 명령을 통해 수행될 수 있으며 이때 PUT 명령에 의해 지정된 키는 프로그램의 식별 번호, 값은 실행 코드에 대응한다.

[0171] 실행 코드는 NDP 회로(500)에서 처리될 수 있다.

[0172] 도 13은 NDP를 위한 키 값 명령을 나타내는 도면이다.

[0173] 도 13에서 (A)는 EXECUTE 명령이고 (E)는 STATUS 명령을 나타낸다.

[0174] EXECUTE 명령과 STATUS 명령 역시 도 2의 명령들과 마찬가지로 NVMe 프로토콜을 확장하여 구현할 수 있다.

[0175] 이들 명령에서 RUN ID 필드는 실행하고자 하는 명령의 종류를 식별하기 위한 정보를 저장하고, Parameter list는 실행 가능한 코드에서 사용할 파라미터를 저장한다.

[0176] 제 1 제어 회로(100)는 EXECUTE 명령을 수신하는 경우 대응하는 실행 코드를 읽어 온다.

[0177] 이는 GET 명령을 수행하는 것과 실질적으로 동일하다.

[0178] 실행 코드는 EXECUTE 명령에 포함된 파라미터와 함께 NDP 회로(500)에서 처리되어 그 상태 및 결과가 제 1 제어 회로(100)에 제공될 수 있다.

[0179] 제 1 제어 회로(100)는 STATUS 명령을 수신하는 경우 NDP 회로(500)에서 RUN ID에 대응하는 실행 코드의 처리 상태를 조회한다.

[0180] 처리 상태는 대기(WAITING), 실행 중(RUNNING), 완료(EXITED)를 표시한다.

[0181] 대기는 실행 코드를 실행하기 전의 상태를 나타내고, 실행 중은 실행 코드를 실행하는 중임을 나타내며 완료는 실행이 완료되어 결과가 제공되었음을 표시한다.

[0182] 실행 코드의 구조는 실시예에 따라 다양하게 결정될 수 있으며 이에 따라 이를 실행하기 위한 NDP 회로의 구조

역시 다양하게 변경될 수 있는 것이어서 이들의 상세 구조에 대해서는 구체적으로 개시하지 않는다.

[0183] 전술한 바와 같이 본 실시예는 파일 시스템에 의존하지 않고 동작하므로 NDP 동작 시 파일 시스템을 통해 파일 주소를 논리 주소로 변환하는 과정이 필요 없어 종래에 비하여 신속한 연산 동작을 수행할 수 있다.

[0184] 본 발명의 권리범위는 이상의 개시로 한정되는 것은 아니다. 본 발명의 권리범위는 청구범위에 문언적으로 기재된 범위와 그 균등범위를 기준으로 해석되어야 한다.

부호의 설명

[0185] 1: 호스트 11: 응용 프로그램 처리 회로

12: 키-값 장치 제어 회로 2: 인터페이스 회로

10: 데이터 저장 장치 100: 제 1 제어 회로

200: 휘발성 메모리 장치, 디램, DRAM

210: 제 1 테이블 저장 영역 220: 제 2 테이블 캐시 영역

230: 메타 데이터 캐시 영역 240: 맵핑 테이블 저장 영역

250: 논리 주소 정보 저장 영역 260: 값 로그 버퍼 영역

300: 제 2 제어 회로, FTL

310: 맵핑 테이블 관리 회로 320: 가비지 콜렉션 관리 회로

400: 비휘발성 메모리 장치, 플래시 메모리 장치

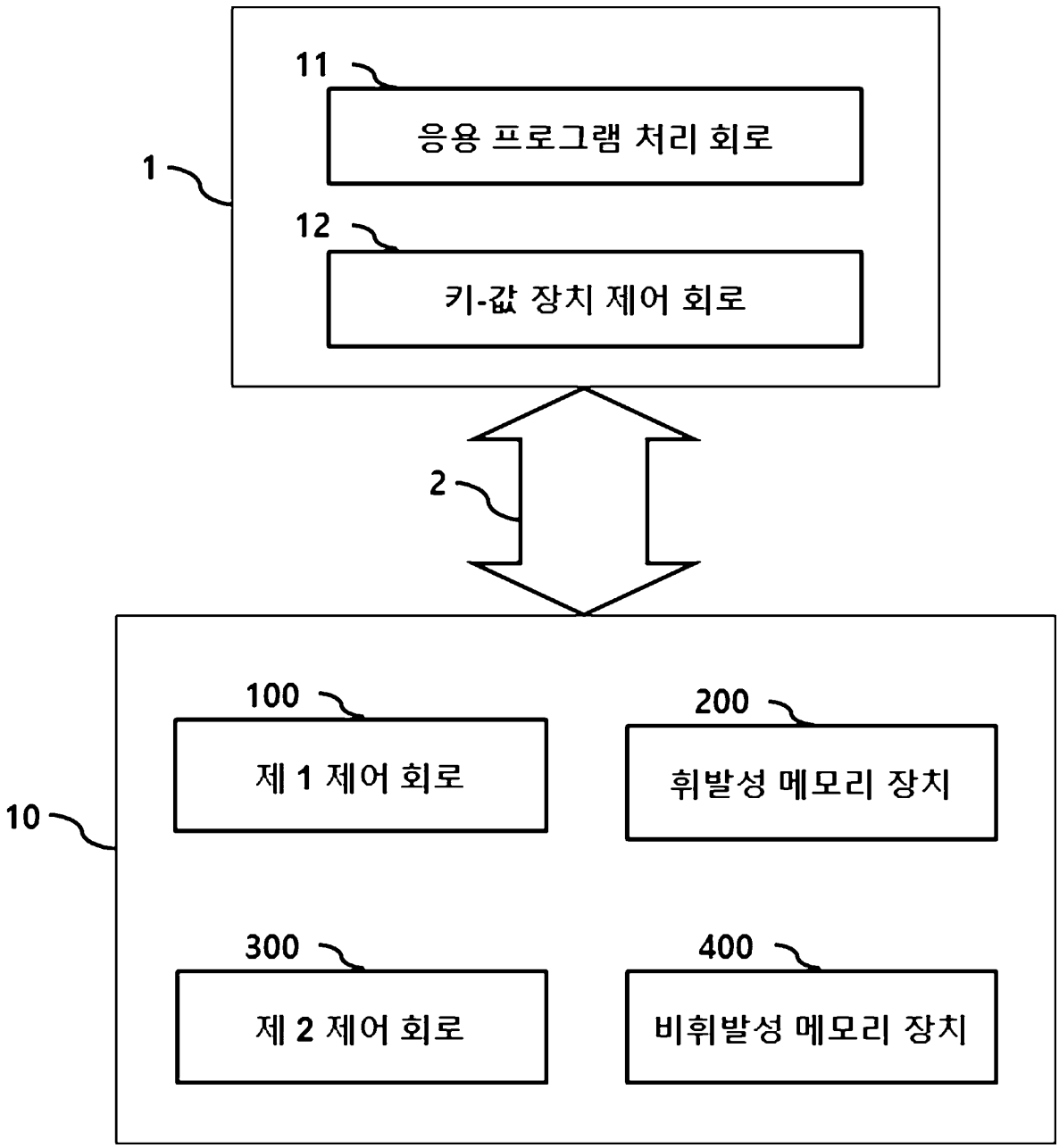
410: 메타 데이터 저장 영역 420: 제 2 테이블 저장 영역

430: 값 로그 영역

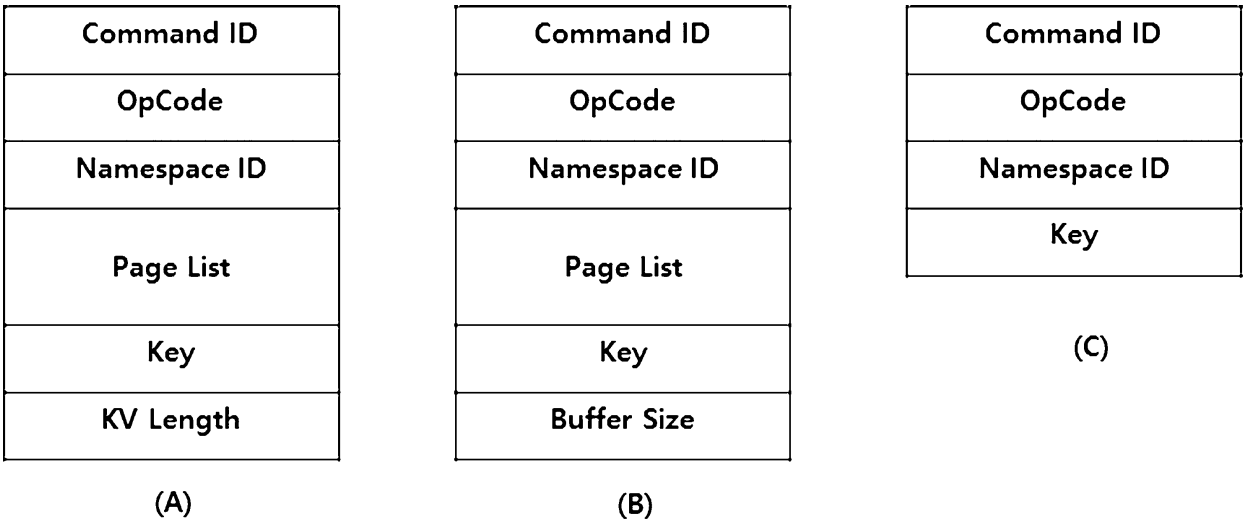
500: 근접 데이터 처리 회로, NDP 회로

도면

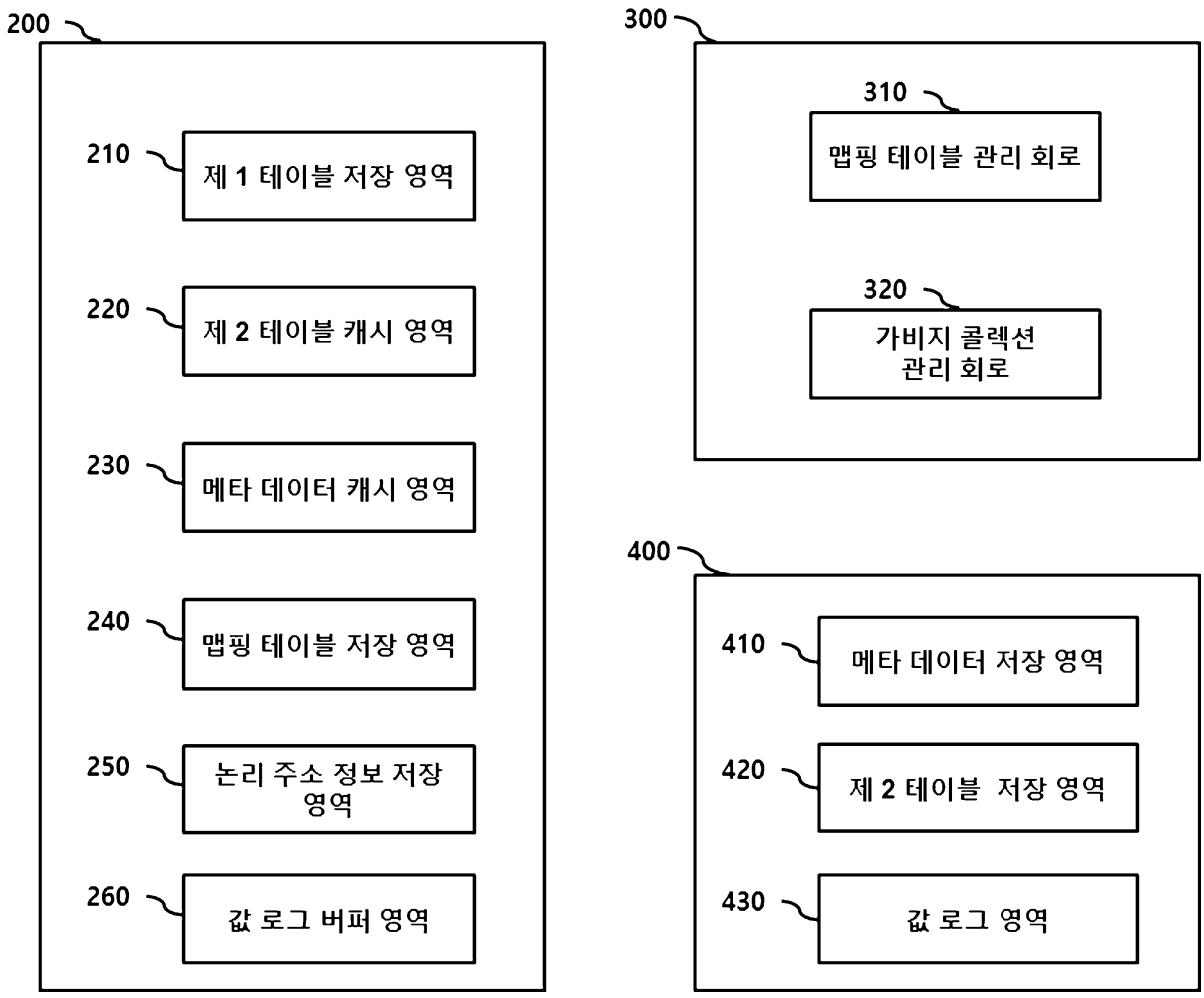
도면1


11
응용 프로그램 처리 회로
1
12
키-값 장치 제어 회로
2
100
제 1 제어 회로
200
휘발성 메모리 장치
10
300
제 2 제어 회로
400
비휘발성 메모리 장치

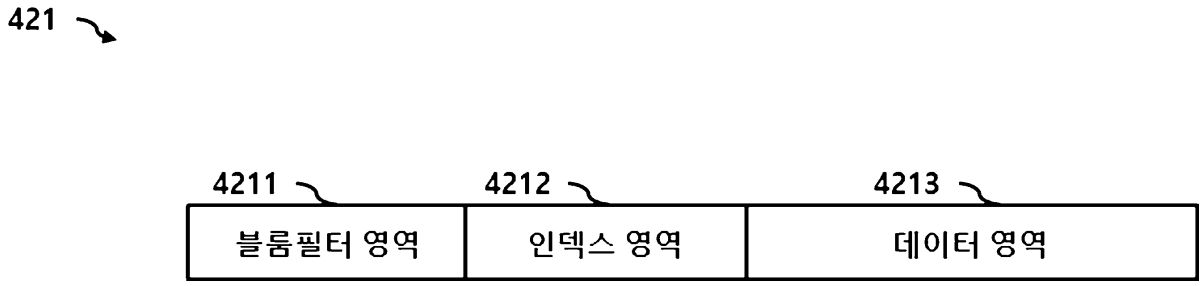
도면2


Command ID
OpCode
Namespace ID
Page List
Key
KV Length
(A)
Command ID
OpCode
Namespace ID
Page List
Key
Buffer Size
(B)
Command ID
OpCode
Namespace ID
Key
(C)

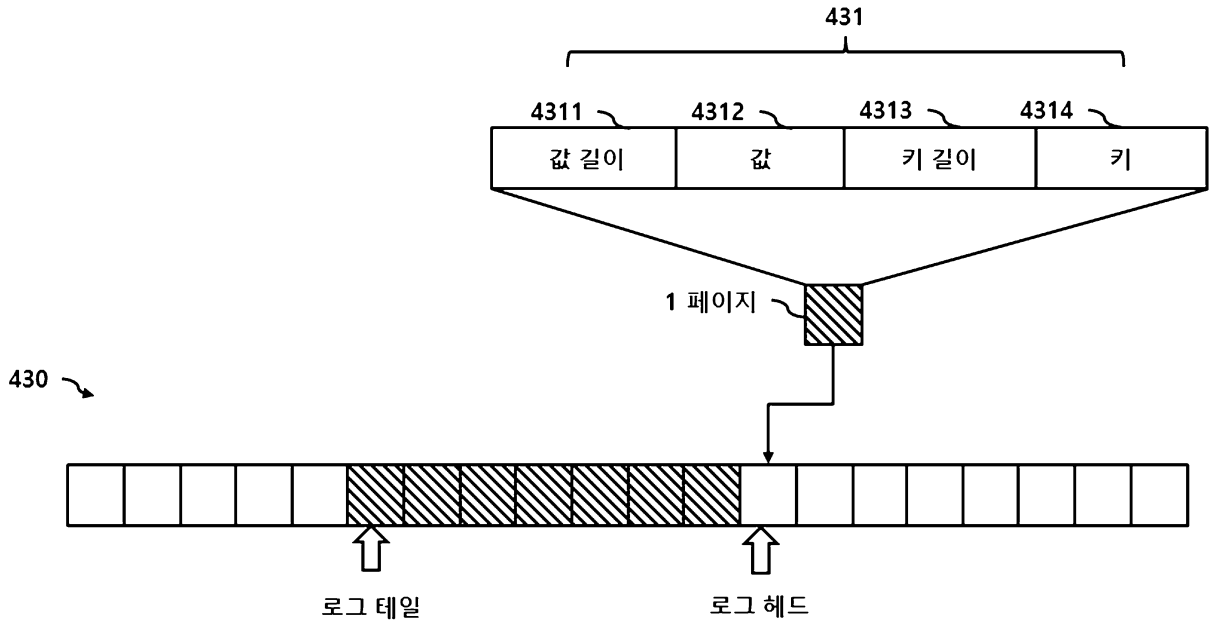
도면3


200
210
제 1 테이블 저장 영역
220
제 2 테이블 캐시 영역
230
메타 데이터 캐시 영역
240
맵핑 테이블 저장 영역
250
논리 주소 정보 저장 영역
260
값 로그 버퍼 영역
300
310
맵핑 테이블 관리 회로
320
가비지 콜렉션 관리 회로
400
410
메타 데이터 저장 영역
420
제 2 테이블 저장 영역
430
값 로그 영역

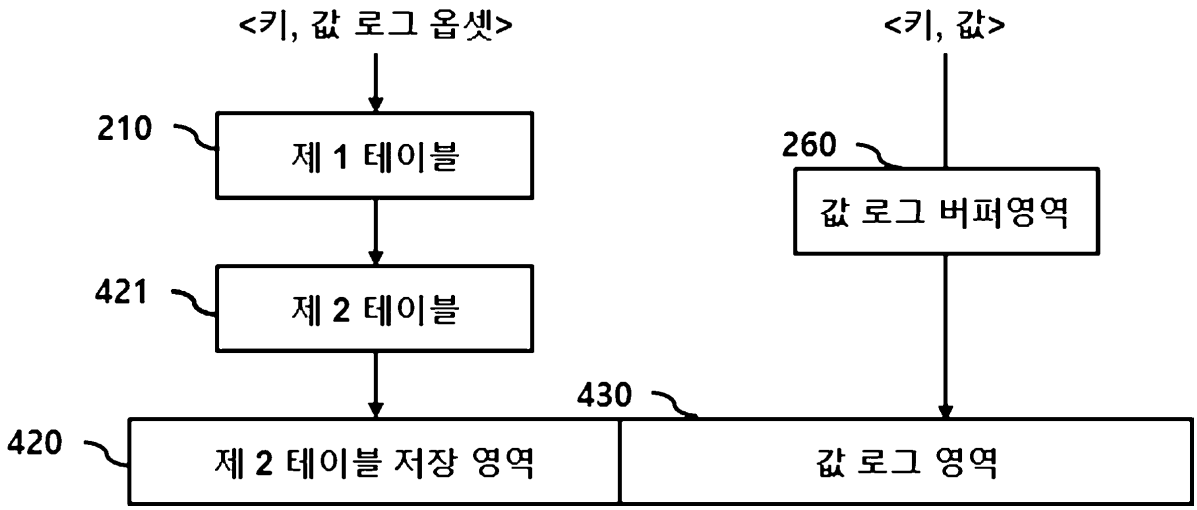
도면4


421
4211
4212
4213
블룸필터 영역
인덱스 영역
데이터 영역

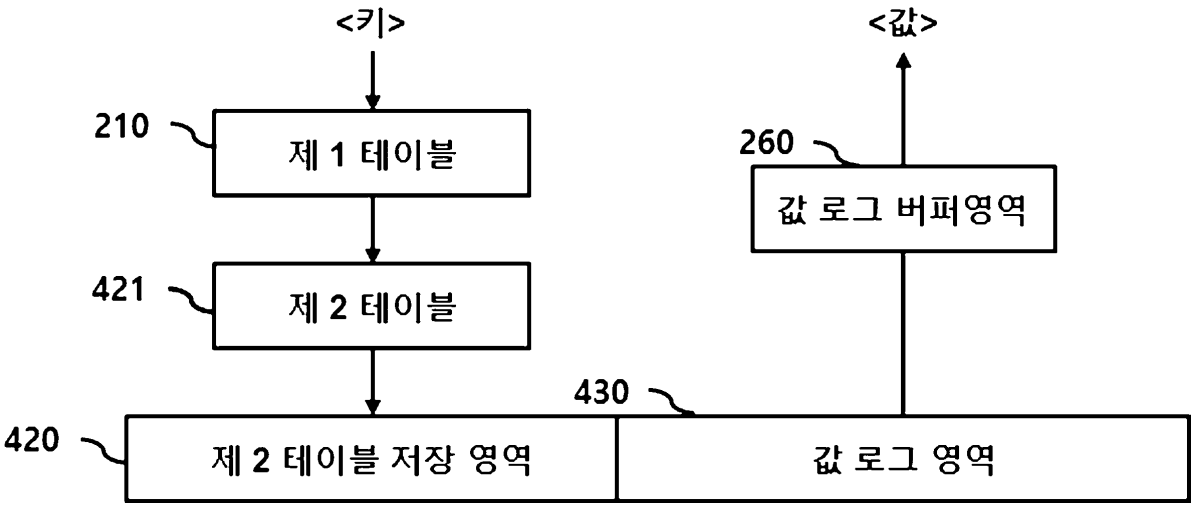
도면5


431
4311
4312
4313
4314
값 길이
값
키 길이
키
1 페이지
430
로그 테일
로그 헤드

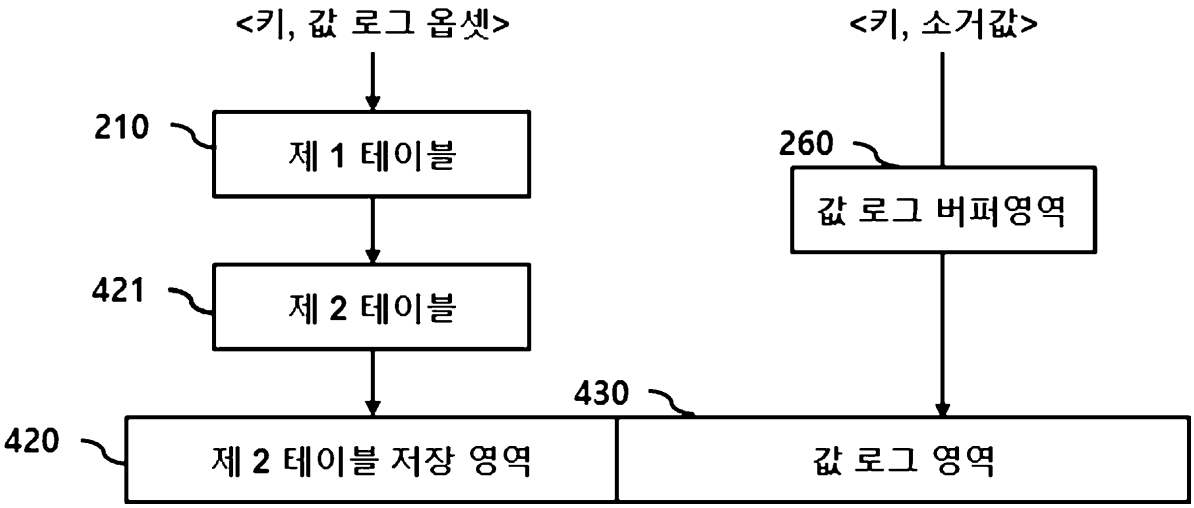
도면6


<키, 값 로그 옵셋>
<키, 값>
210
제 1 테이블
260
값 로그 버퍼영역
421
제 2 테이블
430
420
제 2 테이블 저장 영역
값 로그 영역

도면7

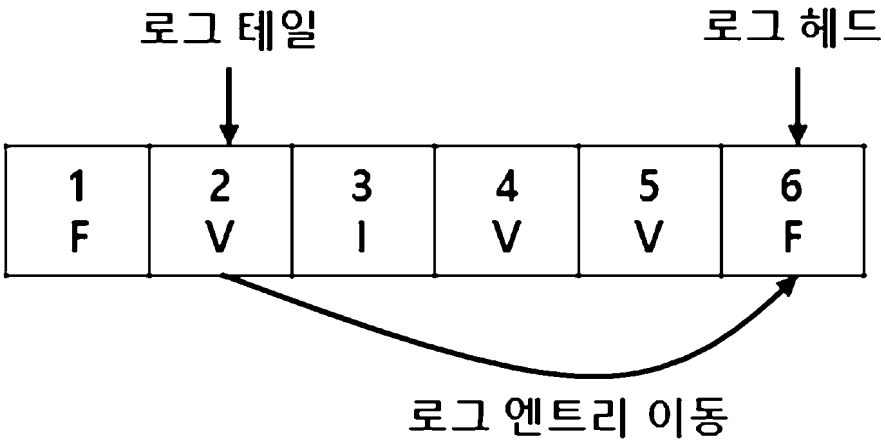

<키>
<값>
210
제 1 테이블
260
값 로그 버퍼영역
421
제 2 테이블
430
420
제 2 테이블 저장 영역
값 로그 영역

도면8

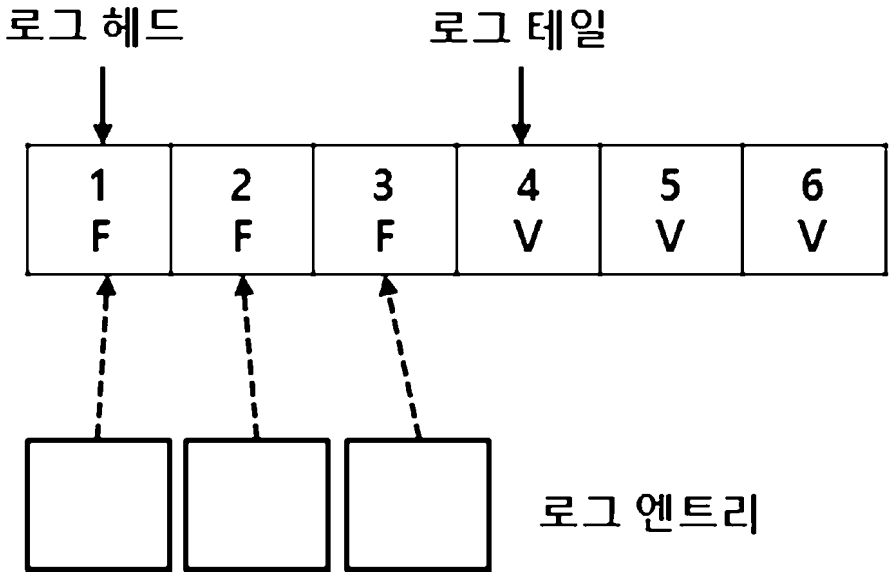

<키, 값 로그 옵셋>
<키, 소거값>
210
제 1 테이블
260
값 로그 버퍼영역
421
제 2 테이블
430
420
제 2 테이블 저장 영역
값 로그 영역

도면9

(A)

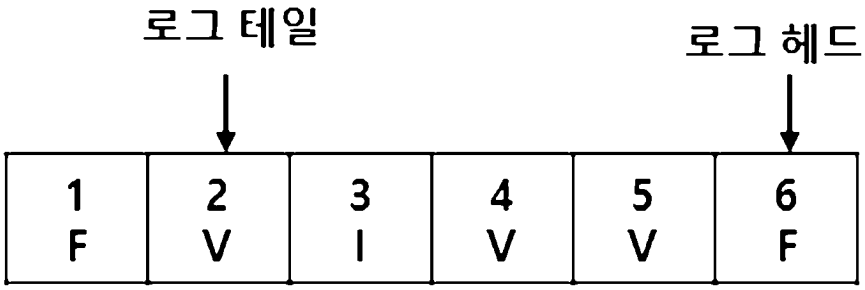

로그 테일
로그 헤드
1
F
2
V
3
I
4
V
5
V
6
F
로그 엔트리 이동

(B)

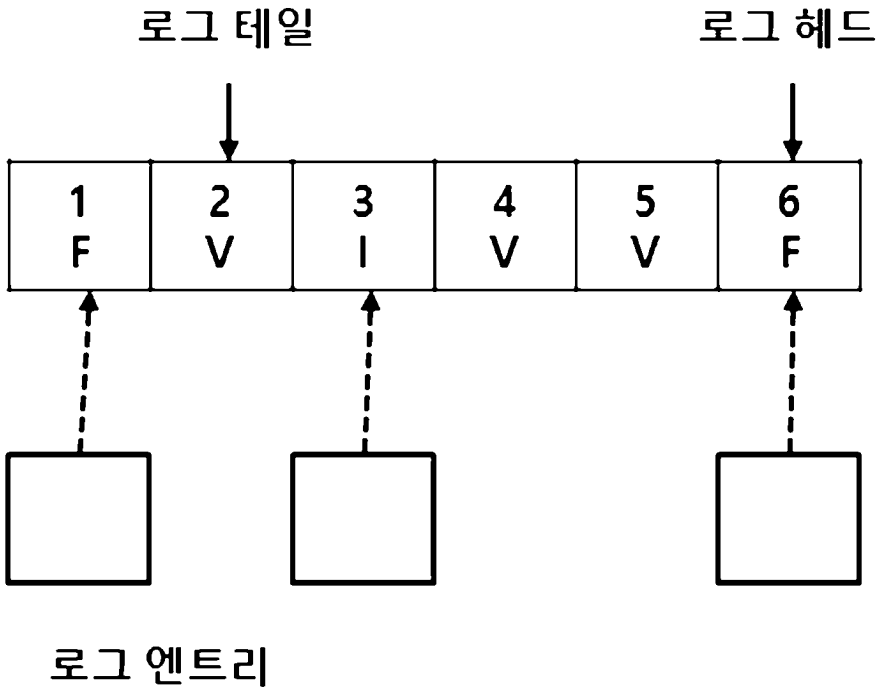

로그 헤드
로그 테일
1
F
2
F
3
F
4
V
5
V
6
V
로그 엔트리

도면10

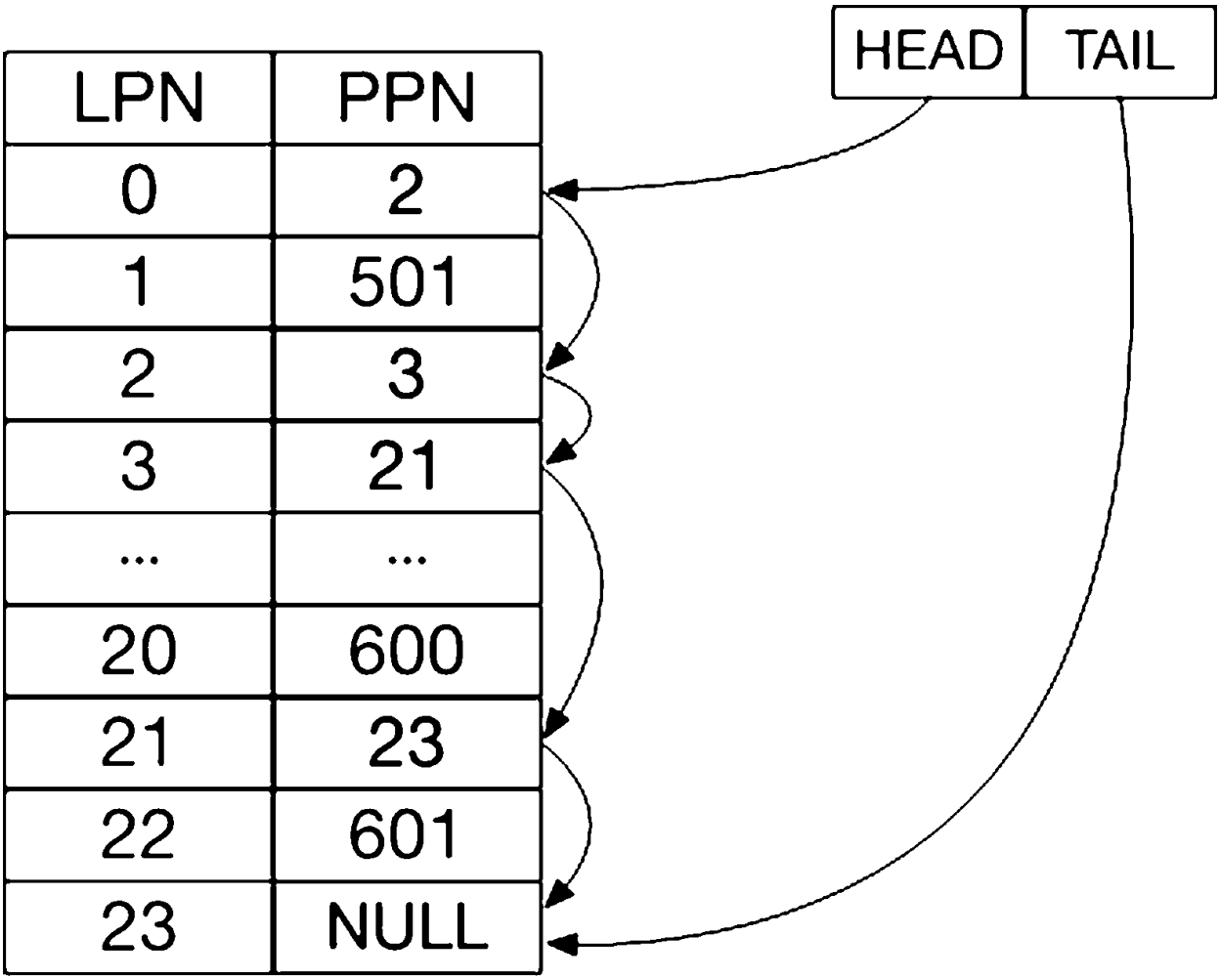
(A)



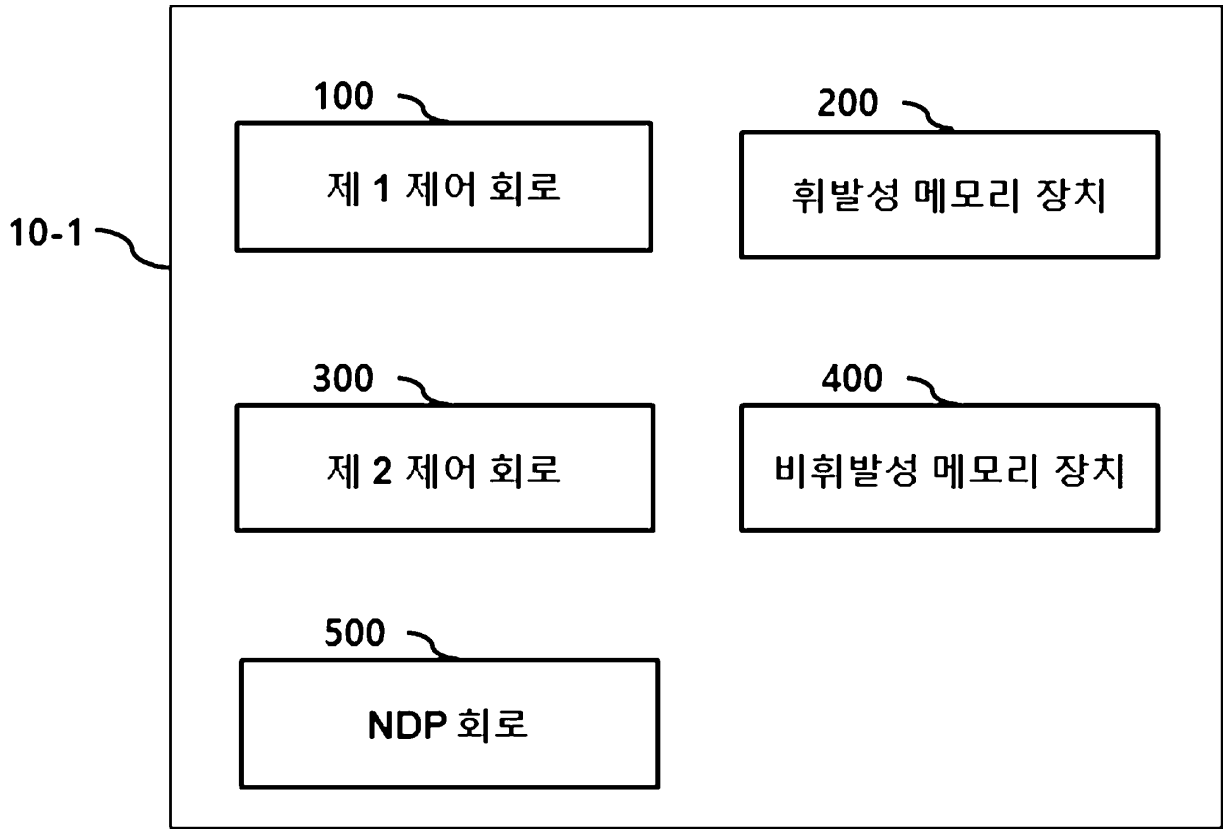
(B)



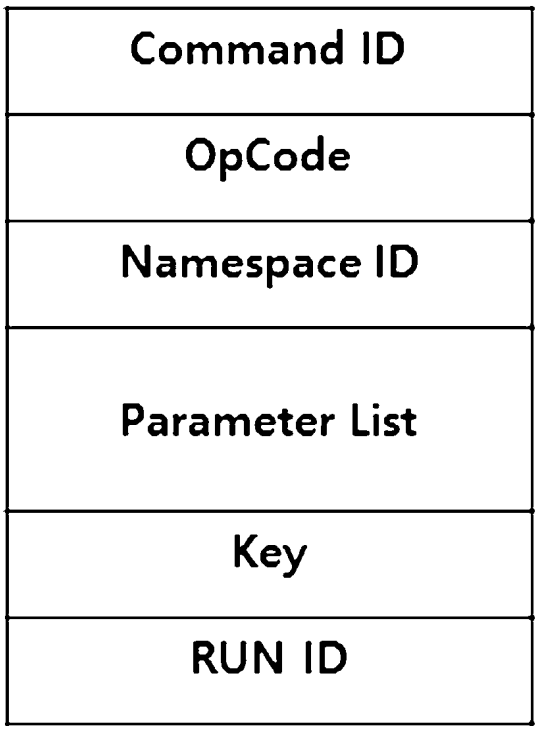
도면11



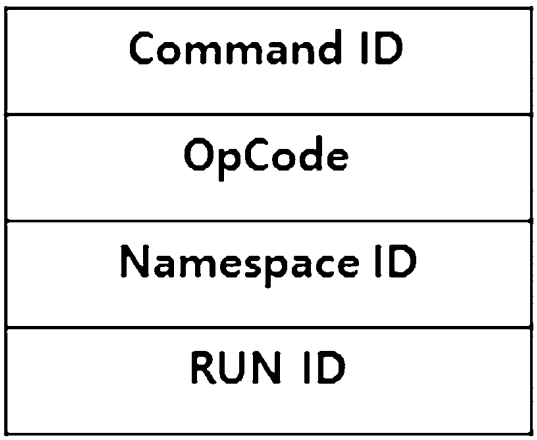
도면12


10-1
100
제 1 제어 회로
200
휘발성 메모리 장치
300
제 2 제어 회로
400
비휘발성 메모리 장치
500
NDP 회로

도면13


Command ID
OpCode
Namespace ID
Parameter List
Key
RUN ID

(A)


Command ID
OpCode
Namespace ID
RUN ID

(B)