



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0006143
(43) 공개일자 2023년01월10일

(51) 국제특허분류(Int. Cl.)
G06F 3/06 (2006.01) G06F 16/22 (2019.01)
G06F 16/25 (2019.01)
(52) CPC특허분류
G06F 3/0611 (2013.01)
G06F 16/2264 (2019.01)
(21) 출원번호 10-2021-0086972
(22) 출원일자 2021년07월02일
심사청구일자 2021년07월02일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암
동5가)
(72) 발명자
정성우
서울특별시 강남구 개포로 264, 124동 2505호(개
포동, 개포래미안포레스트)
김선영
서울특별시 동대문구 왕산로 86, 101동 2212호(용
두동, 동대문 푸르지오시티)
(74) 대리인
송인호, 윤형근, 최관락

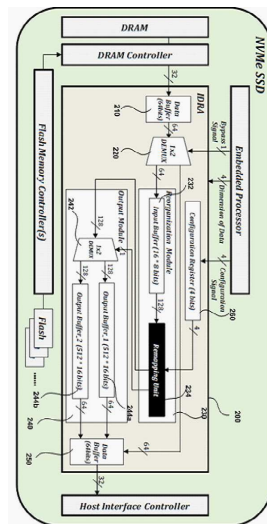
전체 청구항 수 : 총 6 항

(54) 발명의 명칭 다차원 데이터베이스를 위한 스토리지내 데이터 재구성 가속기

(57) 요약

다차원 데이터베이스를 위한 스토리지내 데이터 재구성 가속기가 개시된다. 다차원 데이터베이스를 위한 스토리
지내 데이터 재구성 가속기는, 스토리지내에서 획득된 원시 데이터를 어플리케이션의 데이터 저장 방식에 따라
재구성하는 재구성 모듈; 상기 재구성 모듈의 전단에 위치되며, 상기 원시 데이터의 재구성 여부에 따라 상기 재
구성 모듈로의 전달 또는 바이패싱을 결정하는 제1 디멀스; 및 더블 버퍼링 구조를 가지며, 상기 더블 버퍼링 구
조를 통해 상기 재구성 모듈로부터 재구성된 데이터를 지연 없이 호스트 인터페이스 컨트롤러로 출력하는 출력
모듈을 포함한다.

대표도 - 도2



(52) CPC특허분류

G06F 16/25 (2019.01)

G06F 3/0629 (2013.01)

G06F 3/0656 (2013.01)

G06F 3/0659 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126885
과제번호	2020R1A2C2003500
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구(과기정통부)(R&D)
연구과제명	온도를 고려한 3D 적층 메모리 기반 인메모리 가속기
기 여 율	1/1
과제수행기관명	고려대학교 산학협력단
연구기간	2021.03.01 ~ 2022.02.28

명세서

청구범위

청구항 1

스토리지내에서 획득된 원시 데이터를 어플리케이션의 데이터 저장 방식에 따라 재구성하는 재구성 모듈;

상기 재구성 모듈의 전단에 위치되며, 상기 원시 데이터의 재구성 여부에 따라 상기 재구성 모듈로의 전달 또는 바이패싱을 결정하는 제1 디멀스; 및

더블 버퍼링 구조를 가지며, 상기 더블 버퍼링 구조를 통해 상기 재구성 모듈로부터 재구성된 데이터를 지연 없이 호스트 인터페이스 컨트롤러로 출력하는 출력 모듈을 포함하는 스토리지내 데이터 재구성 가속기.

청구항 2

제1 항에 있어서,

상기 제1 디멀스의 전달에 위치되며, 상기 스토리지내에서 제1 데이터 폭으로 전달되는 원시 데이터를 일시적으로 저장하여 제2 데이터 폭으로 상기 디멀스로 출력하는 제1 데이터 버퍼를 포함하되,

상기 스토리지는 상기 제1 데이터 폭으로 데이터를 입출력하며,

상기 데이터 재구성 가속기는 제2 데이터 폭으로 데이터를 입출력하는 것을 특징으로 하는 스토리지내 데이터 재구성 가속기.

청구항 3

제1 항에 있어서,

상기 재구성 모듈은,

상기 제1 디멀스를 통해 전달되는 제2 데이터 폭의 원시 데이터를 일시적으로 저장하여 제3 데이터 폭으로 출력하는 입력 데이터 버퍼; 및

상기 제3 데이터 폭 단위로 상기 어플리케이션의 데이터 저장 방식에 따라 데이터를 재구성하는 리매핑 유닛을 포함하는 스토리지내 데이터 재구성 가속기.

청구항 4

제3 항에 있어서,

상기 출력 모듈은,

상기 제3 데이터 폭으로 상기 재구성 모듈로부터 전달되는 재구성된 데이터를 상기 제2 데이터 폭 단위로 출력하는 제1 출력 버퍼와 제2 출력 버퍼; 및

상기 제1 출력 버퍼와 상기 제2 출력 버퍼의 전단에 위치되며, 상기 제1 출력 버퍼와 상기 제2 출력 버퍼로 교번으로 상기 재구성된 데이터를 전달하는 제2 디멀스를 포함하되,

동일 클럭 주기에서 상기 제1 출력 버퍼와 상기 제2 출력 버퍼 중 어느 하나는 상기 제2 디멀스를 통해 재구성된 데이터를 저장하고, 다른 하나는 저장된 재구성된 데이터를 상기 제2 데이터 폭 단위로 출력하는 것을 특징으로 하는 스토리지내 데이터 재구성 가속기.

청구항 5

제1 항에 있어서,

상기 스토리지 내장 프로세서로부터 상기 어플리케이션의 데이터 저장 방식을 전달받아 저장하는 레지스터를 더 포함하되,

상기 재구성 모듈은 상기 레지스터에 저장된 어플리케이션의 데이터 저장 방식에 따라 상기 원시 데이터를 재구성하는 것을 특징으로 하는 스토리지내 데이터 재구성 가속기.

청구항 6

제1 항에 있어서,

제2 데이터 버퍼를 더 포함하되,

상기 제1 디렉스는 바이패싱시 상기 원시 데이터를 상기 제2 데이터 버퍼로 전달하되,

상기 제2 데이터 버퍼는 제1 데이터 폭 단위로 데이터를 호스트 컨트롤러를 통해 메인 메모리로 전달하는 것을 특징으로 하는 스토리지내 데이터 재구성 가속기.

발명의 설명**기술 분야**

[0001] 본 발명은 다차원 데이터베이스를 위한 스토리지내 데이터 재구성 가속기에 관한 것이다.

배경 기술

[0003] 최근 빅 데이터 시대가 도래함에 따라, 다수의 차원 (속성)을 가지는 다차원 데이터로 구성된 다차원 데이터베이스가 다방면에서 적용되고 있다. 다차원 데이터베이스가 적용된 대표적인 예시로는 OLAP(On-Line Analytical Processing)와 data warehousing등이 있다. 하지만, 기존에 널리 사용되는 행 지향 저장방식 (row-oriented storage method)을 사용하여 메인 메모리에 다차원 데이터를 저장하는 경우, 데이터 읽기 시 같은 행에 있는 불필요한 데이터가 함께 읽어진다는 단점이 있다.

[0004] 위와 같은 문제점을 해결하기 위해, 최근 열 지향 저장방식 (column-oriented storage method)을 비롯하여 다양한 메인 메모리를 위한 저장 방식들이 제안되고 있다. 이러한 저장 방식들은 탐색 작업 (scan operation)과 같은 데이터베이스 작업들을 가속하여 데이터베이스 시스템의 성능을 향상시키는 방향으로 설계된다. 이처럼 다양한 저장 방식이 적용되고 있는 메인 메모리와 달리, 도 1에 도시된 바와 같이, SSD로 대표되는 스토리지에서는 복잡한 데이터 배치 과정을 피하기 위해 여전히 행 지향 저장방식이 적용된다. 만약 어떤 데이터베이스 시스템에서 메인 메모리에 행 지향 저장 방식 외의 다른 저장 방식을 적용하는 경우, 메인 메모리와 스토리지 간 데이터를 물리적으로 저장하는 방식에 차이가 발생하게 된다. 이에 따라, 데이터를 스토리지에서 메인 메모리로 로드한 후에 해당 데이터를 메인 메모리의 저장 방식에 맞게 변환하는 데이터 재구성 작업을 필요로 한다.

[0005] 기존 CPU 기반 시스템에서는 CPU가 데이터 재구성 작업을 수행한다. CPU가 데이터 재구성 작업을 수행함에 따라, CPU와 메인 메모리 간 막대한 수의 데이터 전송이 발생한다. 이에 따라, 데이터 재구성 과정에서 잦은 메인 메모리 접근과 CPU에서의 데이터 재구성 작업으로 인해 상당한 수준의 지연 시간이 발생한다. 또한, 상당한 수준의 동적 전력 또한 CPU에서 데이터 재구성을 수행하는 동안 소모된다. 본 발명의 필요성을 확인하기 위해 수행한 사전 실험에 따르면, 상용 CPU 기반 시스템에서 TPC-H 데이터 셋을 활용하여 데이터를 재구성하여 쿼리문을 처리하는 경우, 실제로 쿼리문을 수행하는 시간 대비 데이터를 재구성하는데 걸리는 시간은 100배 이상임을 확인할 수 있었다. 이처럼 기존 CPU 기반 시스템에서 메인 메모리에 기존 행 지향 저장방식 외의 다른 저장 방식을 적용하는 경우, 데이터 재구성으로 인해 막대한 비용이 발생하는 것이 불가피하다.

발명의 내용

해결하려는 과제

- [0007] 본 발명은 다차원 데이터베이스를 위한 스토리지내 데이터 재구성 가속기를 제공하기 위한 것이다.
- [0008] 또한, 본 발명은 데이터가 스토리지에서 메인 메모리로 로드되는 동안 데이터 재구성이 가능케 할 수 있는 다차원 데이터베이스를 위한 스토리지내 데이터 재구성 가속기를 제공하기 위한 것이다.
- [0009] 또한, 본 발명은 데이터 재구성을 위한 레이턴시(latency)와 에너지를 줄일 수 있는 다차원 데이터베이스를 위한 스토리지내 데이터 재구성 가속기를 제공하기 위한 것이다.

과제의 해결 수단

- [0011] 본 발명의 일 측면에 따르면, 다차원 데이터베이스를 위한 스토리지내 데이터 재구성 가속기가 제공된다.
- [0012] 본 발명의 일 실시예에 따르면, 스토리지내에서 획득된 원시 데이터를 어플리케이션의 데이터 저장 방식에 따라 재구성하는 재구성 모듈; 상기 재구성 모듈의 전단에 위치되며, 상기 원시 데이터의 재구성 여부에 따라 상기 재구성 모듈로의 전달 또는 바이패싱을 결정하는 제1 디멀스; 및 더블 버퍼링 구조를 가지며, 상기 더블 버퍼링 구조를 통해 상기 재구성 모듈로부터 재구성된 데이터를 지연 없이 호스트 인터페이스 컨트롤러로 출력하는 출력 모듈을 포함하는 스토리지내 데이터 재구성 가속기가 제공될 수 있다.
- [0013] 상기 제1 디멀스의 전달에 위치되며, 상기 스토리지내에서 제1 데이터 폭으로 전달되는 원시 데이터를 일시적으로 저장하여 제2 데이터 폭으로 상기 디멀스로 출력하는 제1 데이터 버퍼를 포함하되, 상기 스토리지는 상기 제1 데이터 폭으로 데이터를 입출력하며, 상기 데이터 재구성 가속기는 제2 데이터 폭으로 데이터를 입출력할 수 있다.
- [0014] 상기 재구성 모듈은, 상기 제1 디멀스를 통해 전달되는 제2 데이터 폭의 원시 데이터를 일시적으로 저장하여 제3 데이터 폭으로 출력하는 입력 데이터 버퍼; 및 상기 제3 데이터 폭 단위로 상기 어플리케이션의 데이터 저장 방식에 따라 데이터를 재구성하는 리매핑 유닛을 포함할 수 있다.
- [0015] 상기 출력 모듈은, 상기 제3 데이터 폭으로 상기 재구성 모듈로부터 전달되는 재구성된 데이터를 상기 제2 데이터 폭 단위로 출력하는 제1 출력 버퍼와 제2 출력 버퍼; 및 상기 제1 출력 버퍼와 상기 제2 출력 버퍼의 전단에 위치되며, 상기 제1 출력 버퍼와 상기 제2 출력 버퍼로 교번으로 상기 재구성된 데이터를 전달하는 제2 디멀스를 포함하되, 동일 클럭 주기에서 상기 제1 출력 버퍼와 상기 제2 출력 버퍼 중 어느 하나는 상기 제2 디멀스를 통해 재구성된 데이터를 저장하고, 다른 하나는 저장된 재구성된 데이터를 상기 제2 데이터 폭 단위로 출력하는
- [0016] 상기 스토리지 내장 프로세서로부터 상기 어플리케이션의 데이터 저장 방식을 전달받아 저장하는 레지스터를 더 포함하되, 상기 재구성 모듈은 상기 레지스터에 저장된 어플리케이션의 데이터 저장 방식에 따라 상기 원시 데이터를 재구성할 수 있다.
- [0017] 제2 데이터 버퍼를 더 포함하되, 상기 제1 디멀스는 바이패싱시 상기 원시 데이터를 상기 제2 데이터 버퍼로 전달하되, 상기 제2 데이터 버퍼는 제1 데이터 폭 단위로 데이터를 호스트 컨트롤러를 통해 메인 메모리로 전달할 수 있다.

발명의 효과

- [0019] 본 발명의 일 실시예에 따른 다차원 데이터베이스를 위한 스토리지내 데이터 재구성 가속기를 제공함으로써, 데이터가 스토리지에서 메인 메모리로 로드되는 동안 데이터 재구성이 가능케 할 수 있으며, 이를 통해, 데이터 재구성을 위한 레이턴시(latency)와 에너지를 줄일 수 있는 이점이 있다.

도면의 간단한 설명

- [0021] 도 1은 스토리지의 데이터 저장 방식을 설명하기 위해 도시한 도면.

- 도 2는 본 발명의 일 실시예에 따른 스토리지내 데이터 재구성 가속기의 구조를 도시한 도면.
 도 3은 본 발명의 일 실시예에 따른 3차원 데이터를 저장하는 차원슬라이스를 예시한 도면.
 도 4는 스토리지와 메인 메모리의 데이터 저장 방식의 차이를 설명하기 위해 도시한 도면.
 도 5는 종래의 방법에 따른 데이터 재구성 과정을 설명하기 위해 도시한 도면.
 도 6은 본 발명의 일 실시예에 따른 데이터 재구성 과정을 설명하기 위해 도시한 도면.
 도 7은 종래와 본 발명의 일 실시예에 따른 데이터 재구성을 위한 실행 시간을 비교한 결과 그래프.
 도 8은 종래와 본 발명의 일 실시예에 따른 데이터 재구성을 위한 에너지 소모를 비교한 결과 그래프.

발명을 실시하기 위한 구체적인 내용

- [0022] 본 명세서에서 사용되는 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "구성된다" 또는 "포함한다" 등의 용어는 명세서상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다. 또한, 명세서에 기재된 "...부", "모듈" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어와 소프트웨어의 결합으로 구현될 수 있다.
- [0023] 이하, 첨부된 도면들을 참조하여 본 발명의 실시예를 상세히 설명한다.
- [0025] 도 2는 본 발명의 일 실시예에 따른 스토리지내 데이터 재구성 가속기의 구조를 도시한 도면이고, 도 3은 본 발명의 일 실시예에 따른 3차원 데이터를 저장하는 차원슬라이스를 예시한 도면이다. 이하에서는 스토리지가 SSD인 것을 가정하기로 한다.
- [0026] 이러한 스토리지는 복잡한 데이터 배치 과정을 피하기 위해 행 지향 저장 방식으로 데이터를 저장하며, 메인 메모리는 행 지향 저장 방식 이외의 다른 저장 방식을 적용하는 것을 가정하기로 한다.
- [0027] 또한, 스토리지와 메인 메모리는 제1 데이터 폭(width)로 데이터를 처리하는 것을 가정하기로 하며, 본 발명의 일 실시예에 따른 데이터 재구성 가속기는 제2 데이터 폭으로 데이터를 처리하는 것을 가정하기로 한다.
- [0028] 예를 들어, 제1 데이터 폭은 32비트일 수 있으며, 제2 데이터 폭은 64비트일 수 있다.
- [0029] 도 2를 참조하면, 본 발명의 일 실시예에 따른 데이터 재구성 가속기(200)는 제1 데이터 버퍼(210), 제1 디믹스(220), 레지스터(260), 재구성 모듈(230), 출력 모듈(240) 및 제2 데이터 버퍼(250)를 포함하여 구성된다.
- [0030] 제1 데이터 버퍼(210)는 DARM 컨트롤러로부터 스토리지로부터 제1 데이터 폭 단위로 원시 데이터를 전달받는다. 스토리지내의 DRAM 컨트롤러는 DRAM에서 타겟 데이터를 32비트 단위로 읽은 후 제1 데이터 버퍼(210)로 전달할 수 있다.
- [0031] 본 발명의 일 실시예에 따른 데이터 재구성 가속기(200)는 제2 데이터 폭을 가지므로, 제1 데이터 버퍼(210)는 제2 데이터 폭의 데이터가 DARM 컨트롤러로부터 전달될때까지 일시적으로 원시 데이터를 저장할 수 있다.
- [0032] 제1 데이터 버퍼(210)는 입력단과 출력단의 데이터 폭이 상이할 수 있다.
- [0033] 제1 데이터 버퍼(210)는 DARM 컨트롤러가 제1 데이터 폭을 가지므로, 입력단은 제1 데이터 폭을 가지며, 데이터 재구성 가속기(200)는 제2 데이터 폭을 가지므로 출력단은 제2 데이터 폭을 가질 수 있다.
- [0034] 제1 데이터 버퍼(210)는 DARM 컨트롤러로부터 제2 데이터 폭의 데이터가 전송될때까지 데이터를 일시적으로 저장한 후 제2 데이터 폭의 데이터가 모두 전송되며 해당 제2 데이터 폭의 데이터를 제1 디믹스(220)로 출력할 수 있다.
- [0035] 제1 디믹스(220)는 제1 데이터 버퍼(210)로부터 전달된 데이터를 재구성 모듈(230)로 전달 또는 바이패싱하기 위한 수단이다.
- [0036] 제1 디믹스(220)는 스토리지 내장 프로세서의 컨트롤 신호에 따라 제1 데이터 버퍼(210)로부터 전달된 데이터를 재구성 모듈(230)로 전달하거나 바이패스하여 제2 데이터 버퍼(250)로 출력한다.

- [0037] 제2 디믹스(242)는 스토리지 내장 프로세서의 컨트롤 신호에 따라 데이터 재구성이 필요한 경우 재구성 모듈(230)로 데이터를 전달한다. 그러나 만일 컨트롤 신호에 따라 재구성이 필요하지 않은 경우, 제1 디믹스(220)는 재구성 모듈(230)을 바이패싱하여 데이터를 제2 데이터 버퍼(250)로 출력할 수 있다.
- [0038] 레지스터(260)는 어플리케이션에 따른 재구성 방식을 저장한다.
- [0039] 즉, 레지스터(260)는 스토리지 내장 프로세서로부터 어플리케이션의 데이터 저장 방식(재구성 방식)을 전달받아 저장할 수 있다.
- [0040] 데이터 저장 방식은 어플리케이션에 따라 상이하다. 따라서, 스토리지 내장 프로세서는 호스트단의 중앙 처리 장치(CPU)로부터 어플리케이션별 데이터 저장 방식(재구성 방식)을 수신한 후 이를 레지스터(260)에 저장할 수 있다.
- [0041] 재구성 모듈(230)은 레지스터(260)에 저장된 어플리케이션의 데이터 저장 방식(재구성 방식)에 따라 데이터를 재구성하기 위한 수단이다.
- [0042] 이러한 재구성 모듈(230)은 도 2에 도시된 바와 같이, 입력 버퍼(232)와 리매핑 유닛(234)으로 구성된다.
- [0043] 입력 버퍼(232)는 제3 데이터 폭의 데이터가 수집될때까지 데이터를 일시적으로 저장한다.
- [0044] 데이터 재구성 가속기(200)의 중요 구성인 재구성 모듈(230)은 내부적으로 제3 데이터 폭 단위로 데이터를 입력하고, 처리할 수 있다.
- [0045] 따라서, 입력 버퍼(232)는 재구성 모듈(230)의 중요 구성인 리매핑 유닛(234)이 데이터를 처리할 수 있도록 제3 데이터 폭에 해당하는 데이터가 수집될때까지 데이터를 일시적으로 저장한 후 제3 데이터 폭에 해당하는 데이터 수집시 리매핑 유닛(234)으로 전달할 수 있다.
- [0046] 리매핑 유닛(234)은 레지스터(260)를 통해 스토리지 내장 프로세서로부터 어플리케이션에 따른 재구성 방식(즉, 데이터 저장 정보)를 전달받으며, 재구성 방식에 따라 데이터를 재구성하여 출력 모듈(240)로 전달할 수 있다.
- [0047] 이해와 설명의 편의를 도모하기 위해 도 3을 참조하여 보다 상세히 설명하기로 한다.
- [0048] 재구성 모듈(230)은 도 3에 도시된 바와 같이 뱅크 블록의 형태로 원시 데이터를 재구성할 수 있다.
- [0049] 하나의 뱅크 블록을 형성하는 경우, 하나의 데이터에 있는 모든 차원(속성)의 값이 필요하다. 즉, 데이터 차원 만큼 차원의 값이 필요하다. 예를 들어, 도 3에서 310으로 표시된 뱅크 블록을 형성하는 것을 가정하기로 한다.
- 이 경우, 64번째 데이터의 모든 차원의 값 $(d_{64}^1, d_{64}^2, \text{ and } d_{64}^3)$ 이 필요하다.
- [0050] 따라서, 입력 버퍼(232)는 클럭 당 데이터 버퍼에서 4개의 차원(속성)(입력 폭(input width)/ 차원의 값의 크기(size of a dimension value))(65 bits/16bits=4)의 값을 수신할 수 있다.
- [0051] 만일 뱅크 블록(데이터 차원과 동일한 수)을 형성하기 위한 충분한 차원 값이 입력 버퍼(232)에 준비되면, 입력 버퍼(232)에 준비된 데이터를 리매핑 유닛(234)으로 전달된다. 하나의 차원의 값의 크기가 16비트이고, 데이터의 차원은 최대 8이므로, 입력 버퍼(232)의 크기는 16 x 8=128 비트일 수 있다.
- [0052] 레지스터(260)에 저장된 재구성 방법과 데이터의 차원에 따라, 리매핑 유닛(234)은 차원 값을 재구성할 수 있다. 사용하지 않는 비트 필드를 하위 뱅크의 값으로 채워서 재구성된 데이터 블록의 크기는 원시 데이터 블록의 크기와 동일하게 유지될 수 있다.
- [0053] 재구성된 데이터는 리매핑 유닛(234)을 통해 출력 모듈(240)로 전달된다.
- [0054] 출력 모듈(240)은 더블 버퍼링 구조를 가지며, 더블 버퍼링 구조를 통해 재구성 모듈(230)로부터 재구성된 데이터를 지연 없이 호스트 인터페이스 컨트롤러로 출력한다.
- [0055] 출력 모듈(240)은 도 2에 도시된 바와 같이, 제2 디믹스(242), 제1 출력 버퍼(244a) 및 제2 출력 버퍼(244b)를 포함하여 구성된다.
- [0056] 제2 디믹스(242)는 재구성 모듈(230)로부터 전달되는 재구성된 데이터를 제1 및 제2 출력 버퍼(244b) 중 한 곳에 저장한다. 예를 들어, 제t 클럭 주기에서 제2 디믹스(242)는 제1 출력 버퍼(244a)에 데이터를 저장할 수 있다. 또한, 제2 출력 버퍼(244b)는 제t 클럭 주기에서 저장된 재구성된 데이터를 제2 데이터 폭 단위로 제2 데이

터 버퍼(250)로 출력할 수 있다.

- [0057] 또한, 제(t+1) 클럭 주기에서, 제2 디덱스(242)는 제2 출력 버퍼(244b)로 데이터를 저장하며, 제1 출력 버퍼(244a)에서 저장된 재구성된 데이터를 제2 데이터 폭 단위로 제2 데이터 버퍼(250)로 출력할 수 있다.
- [0058] 즉, 출력 모듈(240)은 더블 버퍼링 구조를 가지며, 한 개의 출력 버퍼(제1 출력 버퍼(244a) 및 제2 출력 버퍼(244b) 중 어느 하나)에서 데이터를 저장하는 동안 나머지 한 개의 출력 버퍼(제1 출력 버퍼(244a) 및 제2 출력 버퍼(244b) 중 다른 하나)는 데이터를 출력한다. 출력 모듈(240)에서 출력된 재구성이 완료된 데이터는 제2 데이터 버퍼(250)를 통해 호스트 인터페이스 컨트롤러를 통해 메인 메모리로 전달될 수 있다.
- [0059] 제1 출력 버퍼(244a)와 제2 출력 버퍼(244b)는 각각 데이터 출력이 준비 완료되는 경우, 제2 데이터 폭 단위(예를 들어, 64비트) 비트 출력이 클럭 주기마다 제2 데이터 버퍼(250)로 전송될 수 있다.
- [0060] 본 발명의 일 실시예에 따르면, 제1 출력 버퍼(244a)와 제2 출력 버퍼(244b)의 크기는 예를 들어, 512 x 16 비트일 수 있다. 워드 크기가 512 비트이고, 1비트 차원 값이 저장될 때 워드의 가장 큰 수가 필요하다. 예를 들어, 한 차원 값의 크기가 16 비트이므로, 요구되는 워드의 가장 큰 수는 16일 수 있다.
- [0061] 도 2에서는 데이터 재구성 가속기가 스토리지에 위치하는 경우를 가정하여 이를 중심으로 설명하였다.
- [0062] 그러나, 본 발명의 일 실시예에 따른 데이터 재구성 가속기는 PIM 형태로 메모리 컨트롤러내에 위치할 수도 있다. 데이터 재구성 가속기가 스토리지와 메모리 컨트롤러에 위치하는 경우 데이터 재구성 가속기가 동일한 데이터 패스상에 위치될 수 있다. DRAM 모듈 대비 다수의 스토리지가 있는 시스템의 경우 본 발명의 일 실시예에 따른 데이터 재구성 가속기를 스토리지에 위치시키는 것보다 PIM 형태로 가속기를 활용하는 경우 더 좋은 효율을 보일 수 있다. 본 발명의 일 실시예에 따른 데이터 재구성 가속기는 스토리지와 메모리 컨트롤러 중 한 가지에만 적용시킬 수 있는 것이 아닌 사용자의 환경에 따라 유리한 쪽으로 활용될 수 있다.
- [0064] 도 4는 스토리지와 메인 메모리의 데이터 저장 방식의 차이를 설명하기 위해 도시한 도면이고, 도 5는 종래의 방법에 따른 데이터 재구성 과정을 설명하기 위해 도시한 도면이고, 도 6은 본 발명의 일 실시예에 따른 데이터 재구성 과정을 설명하기 위해 도시한 도면이다.
- [0065] 도 4에 도시된 바와 같이, 스토리지는 행 지향 저장 방식을 사용하는 것을 가정하며, 메인 메모리는 열 지향 저장 방식으로 데이터를 저장하는 것을 가정하여 설명하기로 한다. 스토리지에서 행 지향 저장 방식으로 데이터를 저장하는 경우, 메인 메모리로 데이터를 로드한 후 데이터를 재구성해야만 한다.
- [0066] 종래의 경우, 스토리지에서 원시 데이터를 로드한 후 메인 메모리를 통해 프로세서로 전달된다. 데이터 재구성을 위해 프로세서(CPU)와 메인 메모리간 많은 데이터 전송이 발생하며, 프로세서는 데이터 재구성 과정에서 긴 지연 시간(latency)가 발생하고 상당한 수준의 에너지를 소모하게 된다.
- [0067] 도 6은 본 발명의 일 실시예에 따른 데이터 재구성 과정을 도시한 것으로, 데이터를 스토리지에서 메인 메모리로 로드하는 과정에서 스토리지 내에 있는 데이터 재구성 가속기에서 데이터를 재구성하여 메인 메모리로 전달한다.
- [0068] 이로 인해, 메인 메모리는 데이터 재구성을 위한 추가적인 접근이 필요치 않으며, 프로세서(CPU)와 메인 메모리 사이에 데이터 재구성을 위해 많은 데이터 전송이 불필요하며, 프로세서(CPU)에서 데이터 재구성을 위한 비효율적인 계산을 할 필요가 없어 효율은 높은 이점이 있다.
- [0069] 도 7은 종래와 본 발명의 일 실시예에 따른 데이터 재구성을 위한 실행 시간을 비교한 그래프이다.
- [0070] 도 7에서 보여지는 바와 같이, 본 발명의 일 실시예에 따른 스토리지내 데이터 재구성 가속기가 있는 시스템이 종래의 프로세서(CPU) 기반 시스템 대비 성능이 평균 78.6% 향상된 것을 알 수 있다.
- [0071] 도 8은 종래와 본 발명의 일 실시예에 따른 데이터 재구성을 위한 에너지 소모를 비교한 결과이다.
- [0072] 도 8에서 보여지는 바와 같이, 본 발명의 일 실시예에 따른 스토리지내 데이터 재구성 가속기가 있는 시스템이 종래의 프로세서(CPU) 기반 시스템 대비 에너지 소모가 평균 30.3% 절감되는 것을 알 수 있다.
- [0074] 본 발명의 실시 예에 따른 장치 및 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로

구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 컴퓨터 판독 가능 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야 통상의 기술자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media) 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0075] 상술한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0076] 이제까지 본 발명에 대하여 그 실시 예들을 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시 예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

부호의 설명

[0078] 200: 데이터 재구성 가속기

210: 제1 데이터 버퍼

220: 제1 디먹스

230: 재구성 모듈

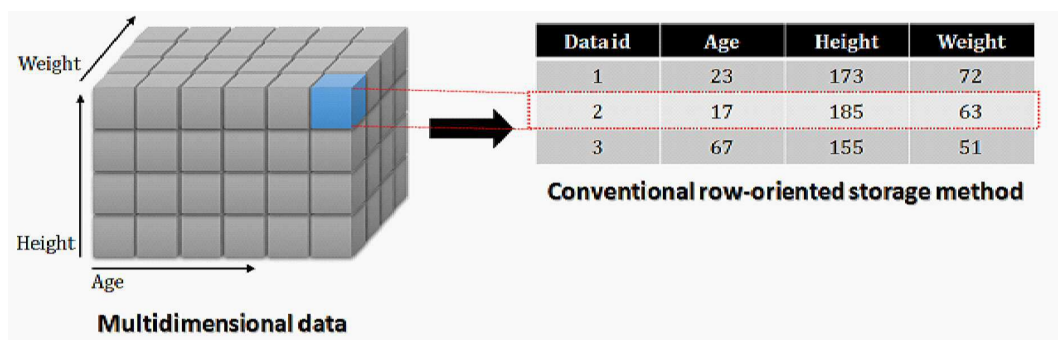
240: 출력 모듈

250: 제2 데이터 버퍼

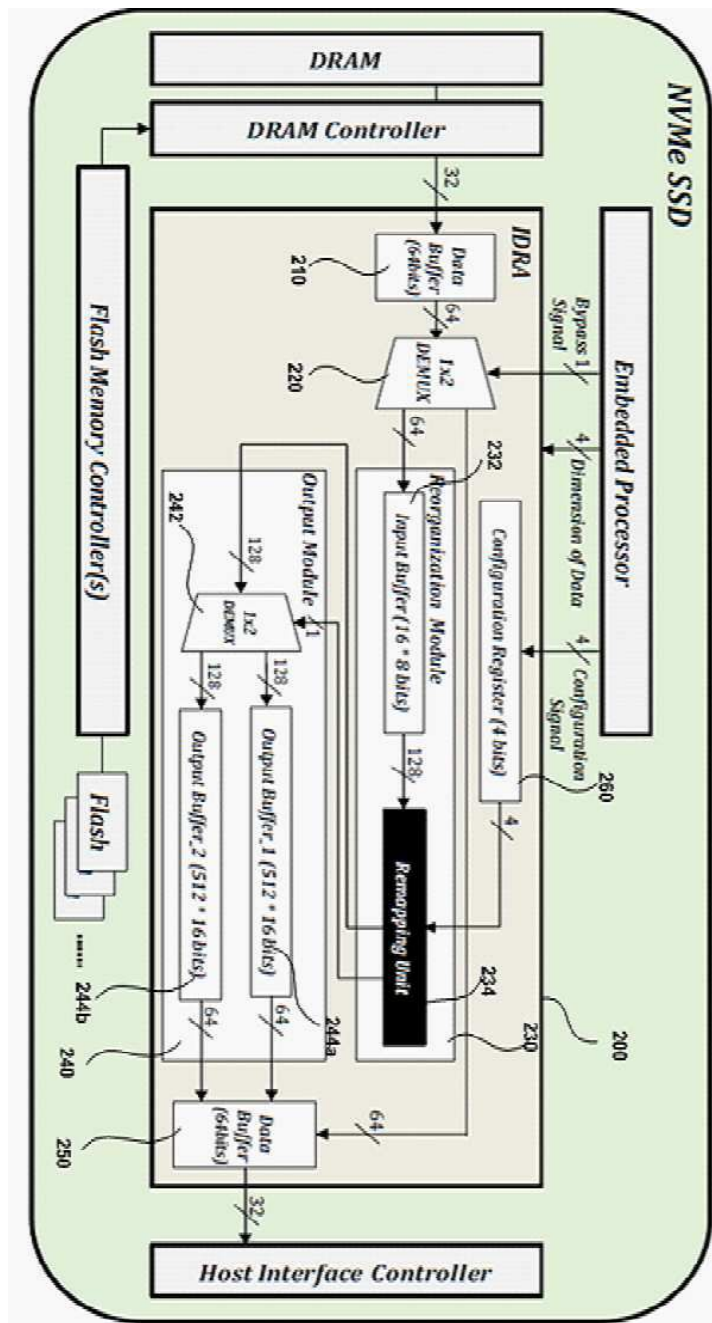
260: 레지스터

도면

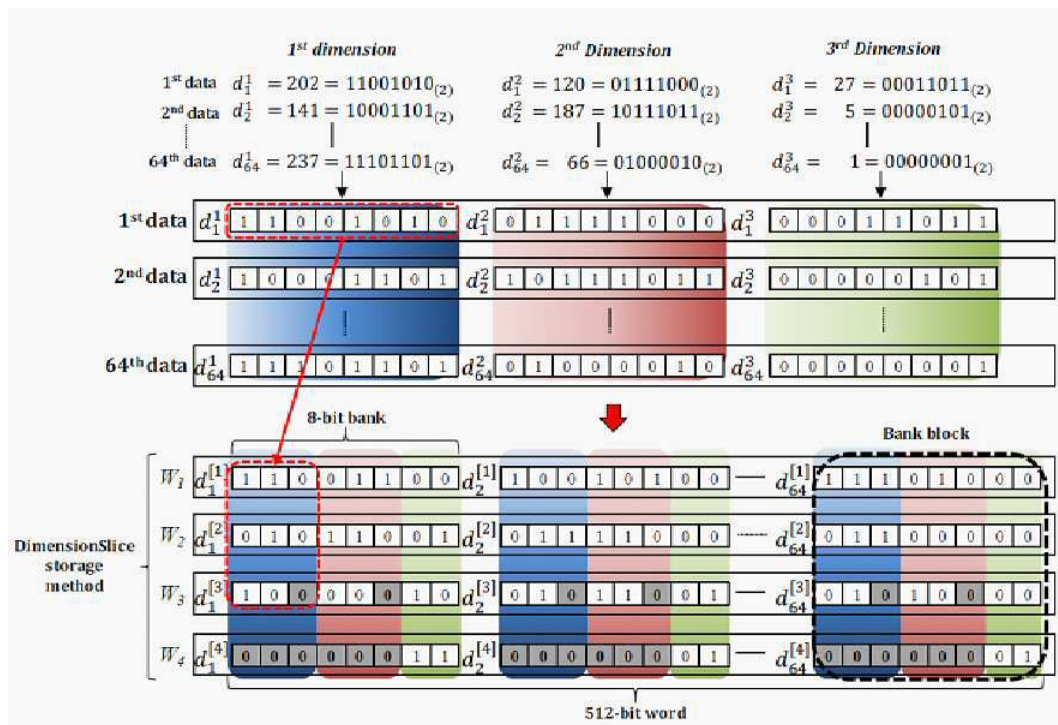
도면1



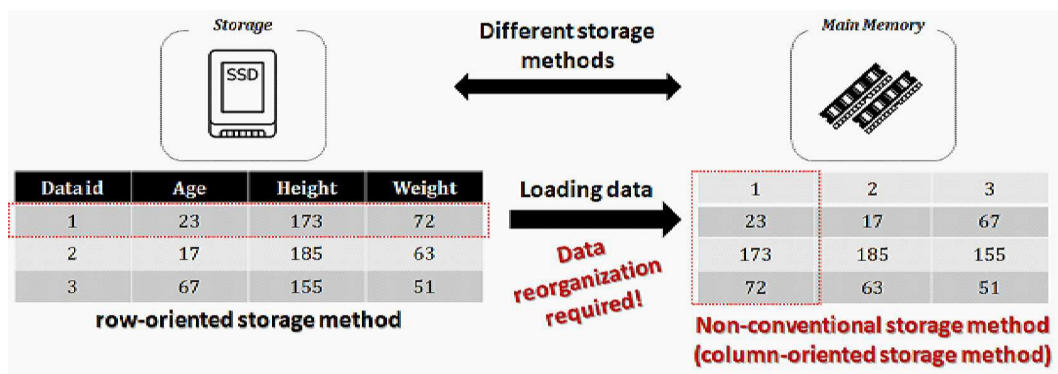
도면2



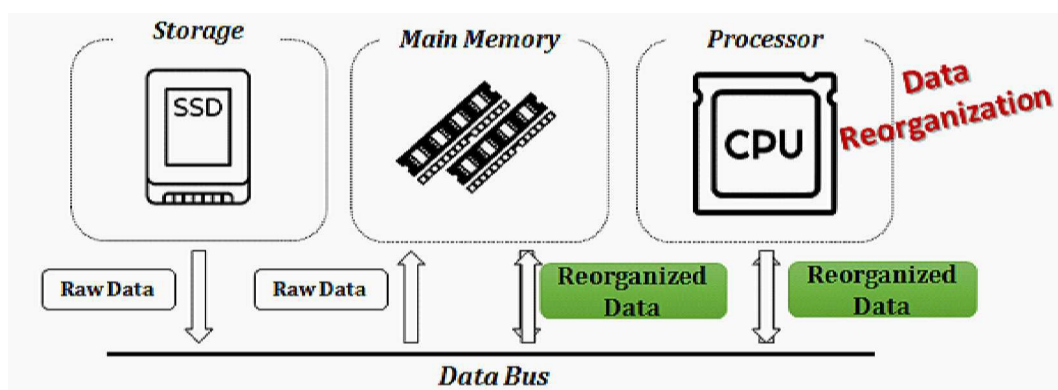
도면3



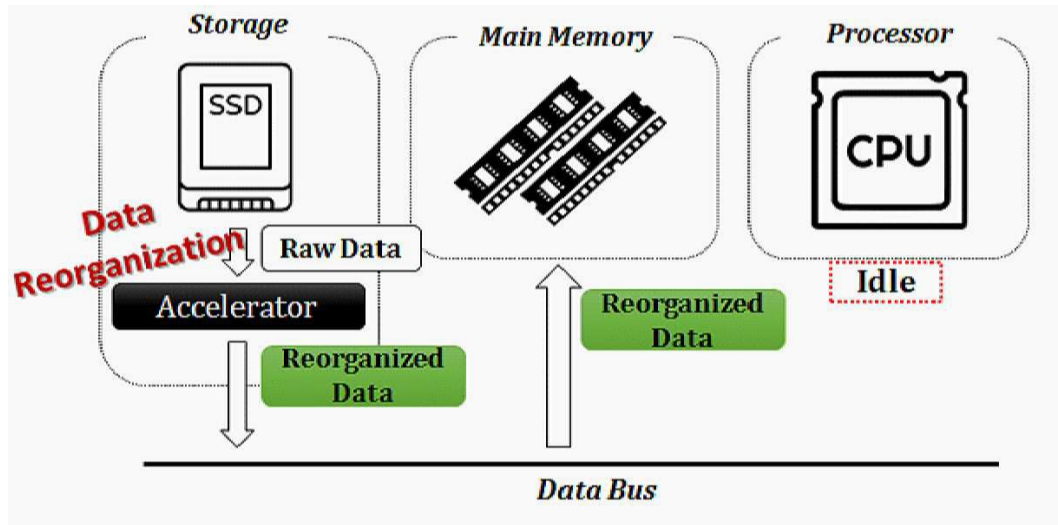
도면4



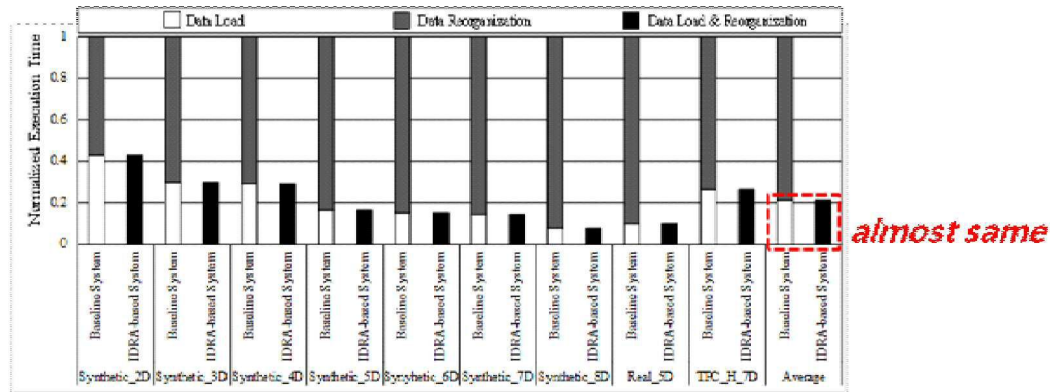
도면5



도면6



도면7



도면8

