(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년11월19일
(11) 등록번호 10-2329307
(24) 등록일자 2021년11월16일

(51) 국제특허분류(Int. Cl.)
H04L 25/03 (2006.01) *H04L 9/08* (2006.01)
H04L 9/12 (2006.01)
(52) CPC특허분류
H04L 25/03866 (2013.01)
H04L 9/0869 (2013.01)
(21) 출원번호 10-2016-0050281
(22) 출원일자 2016년04월25일
심사청구일자 2021년04월23일
(65) 공개번호 10-2016-0131892
(43) 공개일자 2016년11월16일
(30) 우선권주장
62/158,420 2015년05월07일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
KR1020090059280 A

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
창, 치젱
미국 캘리포니아주 94539 프리몬트 도 코트 150
베커맨, 마이클
미국 캘리포니아주 95032 로스가토스 케네디 로드 306
(뒷면에 계속)
(74) 대리인
특허법인 고려

전체 청구항 수 : 총 20 항 심사관 : 황철규

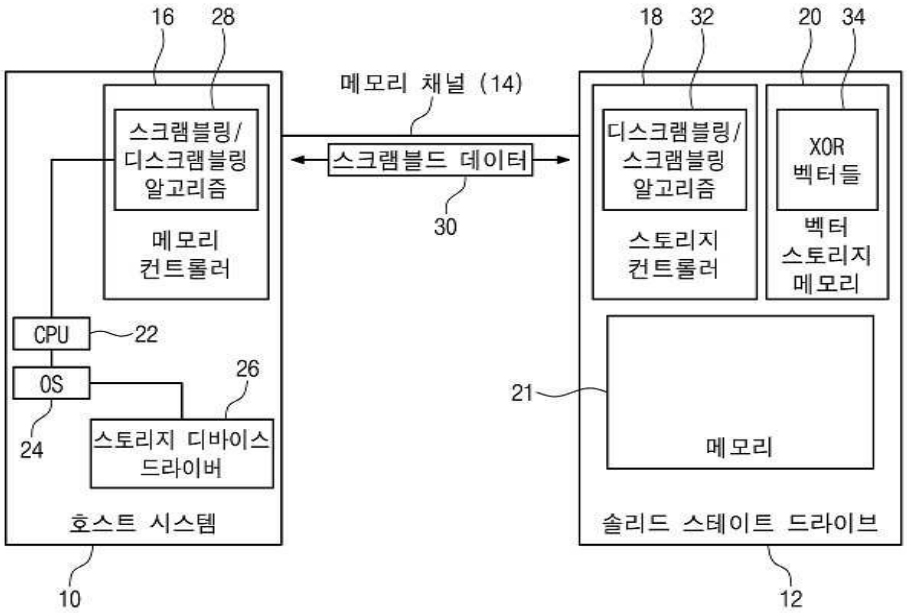
(54) 발명의 명칭 **SSD 통신 프로토콜들을 위한 XOR 기반 스크램블러 및 디스크램블러**

(57) 요 약

메모리 채널을 디스크램블링 및 스크램블링하는 예시적 실시 예는 트레이닝 모드를 실행하는 것을 포함한다. 트레이닝 모드는 데이터를 스크램블하기 위해 호스트 시스템에 의해 사용된 XOR 벡터들을 찾기 위해 상기 메모리 장치에 대하여 실행된다. 상기 XOR 벡터들의 찾음은 스크램블드 트레이닝 데이터를 생성하기 위해 메모리 장치의

(뒷면에 계속)

대 표 도 - 도1



모든 메모리 위치들에 대해 모든 제로 트레이닝 데이터를 스크램블링 알고리즘에 입력함에 의해 수행되며, 상기 스크램블드 트레이닝 데이터는 메모리 장치의 메모리 채널로 전송되며, 상기 스크램블드 트레이닝 데이터는 그 메모리 위치들에 대응하는 XOR 벡터들과 같다. 또한, 상기 찾음은 상기 메모리 장치에 의해 상기 메모리 채널의 상기 스크램블드 트레이닝 데이터를 수신하고, 상기 스크램블드 트레이닝 데이터를 상기 대응 메모리 위치들의 각각에 대한 XOR 벡터들로서 저장하는 것을 포함한다. 또한, 스크램블드 데이터를 수신하고, 상기 스크램블드 데이터를 디스크램블하기 위하여 상기 특정 메모리 위치에 대해 저장된 XOR 벡터를 사용하는 것을 포함한다.

(52) CPC특허분류
H04L 9/12 (2013.01)

(72) 발명자
스와브릭, 이안
미국 캘리포니아주 95051 산타클라라 포메로이 애비뉴 692
핸손, 크레이그
미국 캘리포니아주 95134 산호세 퍼스트 스트리트 4271 엔. 에스피씨 24

(30) 우선권주장
62/209,166 2015년08월24일 미국(US)
15/064,191 2016년03월08일 미국(US)

명 세 서

청구범위

청구항 1

호스트 시스템에 연결된 메모리 장치의 메모리 채널을 통해 전송되는 데이터를 디스크램블링 및 스크램블링하는 방법에 있어서:

데이터를 스크램블하기 위해 상기 호스트 시스템에 의해 사용된 XOR 벡터들을 찾기 위해 상기 메모리 장치에 대해 트레이닝 모드를 실행하는 단계로서:

상기 XOR 벡터들을 찾는 것은 상기 메모리 채널을 통해 상기 메모리 장치로 전송되는 스크램블드 트레이닝 데이터를 생성하기 위해 상기 메모리 장치의 모든 메모리 위치들에 대해 모든 제로 트레이닝 데이터를 스크램블링 알고리즘에 입력하는 것에 의해 수행되고, 상기 스크램블드 트레이닝 데이터는 상기 메모리 위치들에 대응하는 상기 XOR 벡터들과 동일한 것;

상기 메모리 장치에 의해 상기 메모리 채널을 통해 상기 스크램블드 트레이닝 데이터를 수신하고, 그리고 상기 스크램블드 트레이닝 데이터를 상기 대응하는 메모리 위치들의 각각에 대한 XOR 벡터들로서 저장하는 단계; 및

상기 트레이닝 모드 이후 그리고 상기 메모리 장치의 기능 모드 동안, 상기 메모리 장치에 의해, 특정 메모리 위치에 대해 상기 메모리 채널을 통해 스크램블드 데이터를 수신하고, 그리고 상기 특정 메모리 위치에 쓰기 전에 상기 스크램블드 데이터를 디스크램블하기 위해 상기 특정 메모리 위치에 저장된 상기 XOR 벡터들 중 XOR 벡터를 사용하는 단계를 포함하는 방법.

청구항 2

제 1 항에 있어서,

상기 기능 모드 동안 상기 특정 메모리 위치로부터 상기 데이터를 독출하면, 상기 스크램블드 데이터를 상기 메모리 채널을 통해 상기 호스트 시스템으로 전송하기 전에, 상기 데이터를 되찾고 그리고 상기 데이터를 스크램블하기 위해 상기 특정 메모리 위치에 저장된 상기 XOR 벡터를 사용하는 단계를 더 포함하는 방법.

청구항 3

제 1 항에 있어서,

상기 호스트 시스템 상의 스토리지 장치 드라이버는 상기 트레이닝 모드를 시작하는 방법.

청구항 4

제 1 항에 있어서,

상기 트레이닝 모드 동안 상기 메모리 장치 상의 스토리지 컨트롤러는 상기 스크램블드 트레이닝 데이터를 수신하고, 그리고 디스크램블링 및 스크램블링 알고리즘은 상기 XOR 벡터들을 저장하는 방법.

청구항 5

제 4 항에 있어서,

상기 메모리 장치는 상기 XOR 벡터들을 저장하는 벡터 스토리지 메모리를 포함하는 방법.

청구항 6

제 1 항에 있어서,

상기 기능 모드 동안 상기 스크램블드 데이터를 수신하는 것은:

상기 메모리 장치 상의 디스크램블링 및 스크램블링 알고리즘에 의해, 상기 메모리 장치의 벡터 스토리지 메모리로부터 상기 특정 메모리 위치에 대해 상기 XOR 벡터를 되찾고, 그리고 디스크램블드 데이터를 생성하기 위해

상기 XOR 벡터 및 상기 스크램블드 데이터에 대한 배타적 논리 합 연산을 수행하는 것을 더 포함하고, 상기 디스크램블드 데이터는 상기 특정 메모리 위치 내에 저장되는 방법.

청구항 7

제 1 항에 있어서,

상기 기능 모드 동안 수신되는 상기 스크램블드 데이터는: 컨트롤 및 디스크립터 데이터, 상태 데이터, 사용자 데이터, 및 컨트롤 데이터 중 적어도 하나를 포함하는 방법.

청구항 8

제 7 항에 있어서,

상기 스크램블드 데이터가 상기 컨트롤 및 디스크립터 데이터를 포함하면, 상기 메모리 장치 상의 디스크램블링 및 스크램블링 알고리즘은 상기 트레이닝 모드 동안 각기 대응하는 컨트롤 및 디스크립터 메모리 위치에 대한 상기 XOR 벡터를 저장하는 방법.

청구항 9

제 8 항에 있어서,

상기 디스크램블링 및 스크램블링 알고리즘은 상기 XOR 벡터들을 사용하여 상기 기능 모드 동안 상기 컨트롤 및 디스크립터 데이터를 디스크램블하는 방법.

청구항 10

제 7 항에 있어서,

상기 스크램블드 데이터가 상기 상태 데이터를 포함하면, 상기 메모리 장치 상의 디스크램블링 및 스크램블링 알고리즘은 상기 트레이닝 모드 동안 각기 대응하는 상태 메모리 위치에 대한 상기 XOR 벡터를 저장하는 방법.

청구항 11

제 10 항에 있어서,

상기 디스크램블링 및 스크램블링 알고리즘은 상기 XOR 벡터들을 사용하여 상기 기능 모드 동안 상기 상태 데이터를 스크램블하는 방법.

청구항 12

제 7 항에 있어서,

상기 스크램블드 데이터가 상기 사용자 데이터 및 컨트롤 데이터를 포함하면, 상기 메모리 장치 상의 디스크램블링 및 스크램블링 알고리즘은 상기 트레이닝 모드 동안 각기 대응하는 사용자 데이터 메모리 위치에 대한 상기 XOR 벡터를 저장하는 방법.

청구항 13

제 12 항에 있어서,

상기 디스크램블링 및 스크램블링 알고리즘은 상기 기능 모드 동안 상기 메모리 장치의 독출 동안에 상기 사용자 데이터 및 컨트롤 데이터를 스크램블하기 위해 상기 XOR 벡터들을 사용하고; 그리고 상기 기능 모드 동안 상기 메모리 장치에 쓰기 위해, 디스크램블링 알고리즘은 상기 메모리 장치로 쓰기 전에 상기 사용자 데이터 및 상기 컨트롤 데이터를 디스크램블하는 방법.

청구항 14

제 7 항에 있어서,

상기 메모리 장치 상의 벡터 스토리지 메모리 내에 상기 XOR 벡터들을 저장하는 단계를 더 포함하고, 상기 벡터 스토리지 메모리는 다른 영역들로 분할되고, 상기 다른 영역들은: 데이터의 대응하는 타입을 위한 XOR 벡터들을

저장하기 위해, 컨트롤 및 디스크립터 데이터 벡터 영역, 상태 데이터 벡터 영역, 사용자 데이터 벡터 영역, 및 컨트롤 데이터 벡터 영역 중 하나 이상을 포함하는 방법.

청구항 15

호스트 시스템에 연결된 메모리 장치의 메모리 채널을 통해 전송되는 데이터를 디스크램블링 및 스크램블링하는 시스템에 있어서:

상기 호스트 시스템 상에 설치된 스토리지 장치 드라이버; 및

상기 메모리 장치 상의 스토리지 컨트롤러를 포함하되,

상기 스토리지 장치 드라이버는 데이터를 스크램블하기 위해 상기 호스트 시스템에 의해 사용된 XOR 벡터들을 찾기 위해 상기 메모리 장치에 대해 트레이닝 모드를 실행하고:

상기 XOR 벡터들을 찾는 것은 상기 메모리 채널을 통해 상기 메모리 장치로 전송되는 스크램블드 트레이닝 데이터를 생성하기 위해 상기 메모리 장치의 모든 메모리 위치들에 대해 모든 제로 트레이닝 데이터를 스크램블링 알고리즘에 입력하는 것에 의해 수행되고, 상기 스크램블드 트레이닝 데이터는 상기 메모리 위치들에 대응하는 상기 XOR 벡터들과 동일하고,

상기 스토리지 컨트롤러는:

상기 메모리 채널을 통해 상기 스크램블드 트레이닝 데이터를 수신하고, 그리고 상기 스크램블드 트레이닝 데이터를 상기 대응하는 메모리 위치들의 각각에 대한 XOR 벡터들로서 저장하고; 그리고

상기 트레이닝 모드 이후 그리고 상기 메모리 장치의 기능 모드 동안, 특정 메모리 위치에 대해 상기 메모리 채널을 통해 스크램블드 데이터를 수신하고, 그리고 상기 특정 메모리 위치에 쓰기 전에 상기 스크램블드 데이터를 디스크램블하기 위해 상기 특정 메모리 위치에 저장된 상기 XOR 벡터들 중 XOR 벡터를 사용하도록 구성된 시스템.

청구항 16

제 15 항에 있어서,

상기 기능 모드 동안 상기 특정 메모리 위치로부터 상기 데이터를 독출하면, 상기 메모리 장치는, 상기 스크램블드 데이터를 상기 메모리 채널을 통해 상기 호스트 시스템으로 전송하기 전에, 상기 데이터를 되찾고 그리고 상기 데이터를 스크램블하기 위해 상기 특정 메모리 위치에 저장된 상기 XOR 벡터를 사용하는 시스템.

청구항 17

제 15 항에 있어서,

상기 트레이닝 모드 동안 상기 메모리 장치 상의 스토리지 컨트롤러는 상기 스크램블드 트레이닝 데이터를 수신하고, 그리고 디스크램블링 알고리즘은 XOR 벡터들을 저장하는 시스템.

청구항 18

제 17 항에 있어서,

상기 메모리 장치는 상기 XOR 벡터들을 저장하는 벡터 스토리지 메모리를 포함하는 시스템.

청구항 19

제 15 항에 있어서,

상기 기능 모드 동안 스크램블드 데이터를 수신하는 동안에, 상기 메모리 장치 상의 디스크램블링 및 스크램블링 알고리즘은 상기 메모리 장치의 벡터 스토리지 메모리로부터 상기 특정 메모리 위치에 대해 상기 XOR 벡터를 되찾고, 그리고 디스크램블드 데이터를 생성하기 위해 상기 XOR 벡터 및 상기 스크램블드 데이터에 대한 배타적 논리 합 연산을 수행하고, 상기 디스크램블드 데이터는 상기 특정 메모리 위치 내에 저장되는 시스템.

청구항 20

제 15 항에 있어서,

상기 기능 모드 동안 수신된 상기 스크램블드 데이터는: 컨트롤 및 디스크립터 데이터, 상태 데이터, 사용자 데이터, 및 컨트롤 데이터 중 적어도 하나를 포함하는 시스템.

발명의 설명

기 술 분 야

[0001] 본 발명은 SSD 통신 프로토콜들을 위한 XOR 기반 스크램블러 및 디스크램블러에 관한 것이다.

배 경 기 술

[0002] 솔리드 스테이트 드라이브(SSD)는 표준 다이나믹 랜덤 억세스 메모리(DRAM)와는 다르다. 다르게 되는 이유로서는 SSD의 경우 호스트가 컨트롤 정보를 SSD로 전송할 수 있게 되는 것이 필요하고 호스트가 상태 정보를 수신할 수 있게 되는 것이 필요하다는 것이다. 상태 정보는 SSD에 의해 생성된다. DRAM 채널을 통해 호스트 시스템과 SSD 간의 데이터 통신은 매우 복잡해질 수 있다. 데이터 통신이 복잡한 이유는 데이터 스크램블링의 사용 및 선택적인 에러 정정 코드(ECC)보호, 그리고 기타 데이터 보호 기법들 때문이다. 대부분의 스크램블링 기법들은 호스트 메모리 컨트롤러에 의해 수행된다. 스크램블링 기법들은 의사 랜덤 (그러나 결정됨)패턴을 생성하는 것과 패턴과 전송될 데이터를 가지고 배타적 논리 합(XOR) 연산을 수행하는 것을 수반한다. 데이터가 XOR 연산되었음을 나타내는 패턴 값은 고정된 시드(seed) 혹은 시드의 일부 조합에 근거할 수 있고, 전송되는 메모리 어드레스에 근거할 수 있다.

[0003] DDR 폼 팩터 스토리지 장치에 대해, 스토리지 장치 메모리 컨트롤러는 커맨드를 실행하고 스토리지 내에 데이터를 저장하기 위해, 메모리 버스를 통해 수신되는 데이터를 디스크램블 하는 것이 필요하다. 유사하게 스토리지로부터 출력된 데이터가 올바르게 호스트 시스템으로 다시 전송되도록 하기 위해, 스토리지 장치 메모리 컨트롤러는 데이터가 호스트 시스템에 의해 인식되도록 데이터를 스크램블 하는 것이 필요할 수 있다. 이러한 것은 스토리지 장치 메모리 컨트롤러가 동일한 스크램블링 및/또는 ECC를 적용할 수 있게 되기를 요구한다. 동일한 스크램블링 및/또는 ECC는 호스트 메모리 컨트롤러에 의해 적용된 것과 같다. 결국, 이는 스크램블링 알고리즘이 역으로 되어야 하고, DDR 채널 인 타임 상에서 데이터 의존성을 가지지 않음을 의미한다.

[0004] 호스트 시스템에 의해 사용된 스크램블링 알고리즘의 전체 역 설계(full reverse engineering)는 어렵다. 스크램블링 알고리즘이 알려져 있는 경우에도 스크램블링 알고리즘은 하나의 벤더(vendor)/플랫폼(platform)에서 다른 것으로 변할 수 있다.

[0005] SSD 통신 프로토콜들에 대해 개선된 XOR 기반 스크램블러/디스크램블러가 필요하며, 바람직하기로 호스트 메모리 컨트롤러에 의해 공급되는 스크램블링 알고리즘의 지식이나 전체 역 설계를 요구하지 않는 것이 필요하다.

발명의 내용

해결하려는 과제

[0006] 본 발명이 해결하고자 하는 기술적 과제는, SSD 통신 프로토콜들을 위한 XOR 기반 스크램블러 및 디스크램블러를 제공함에 있다.

과제의 해결 수단

[0007] 본 발명의 실시 예들은 메모리 채널을 디스크램블링 및 스크램블링하는 방법 및 장치를 제공한다. 예시적 실시예의 양상(aspects)은 데이터를 스크램블하기 위해 상기 호스트 시스템에 의해 사용된 XOR 벡터들을 찾기 위해 상기 메모리 장치에 대하여 트레이닝 모드를 실행하는 것을 포함한다. 상기 XOR 벡터들의 찾음은 스크램블드 트레이닝 데이터를 생성하기 위해 메모리 장치의 모든 메모리 위치들에 대해 모든 제로 트레이닝 데이터를 스크램블링 알고리즘에 입력함에 의해 수행되며, 상기 스크램블드 트레이닝 데이터는 메모리 장치의 메모리 채널로 전송된다. 상기 스크램블드 트레이닝 데이터는 그 메모리 위치들에 대응하는 XOR 벡터들과 같다. 상기 스크램블드 트레이닝 데이터는 상기 메모리 장치에 의해 상기 메모리 채널을 통해 수신된 후, 상기 스크램블드 트레이닝 데이터는 상기 대응 메모리 위치들의 각각에 대한 XOR 벡터들로서 저장된다. 상기 메모리 장치의 기능 모드 동안에, 상기 스크램블드 데이터는 특정 메모리 위치에 대해 상기 메모리 채널을 통해 수신되고, 상기 특정 메모리 위치 내로의 저장에 앞서, 상기 특정 메모리 위치에 대해 저장된 XOR 벡터는 상기 스크램블드 데이터를 디스크

램블 하는데 사용된다.

발명의 효과

[0008] 여기에 개시된 방법 및 시스템에 따라, 상기 예시적인 실시 예들은 호스트 메모리 컨트롤러에 의해 제공되는 스크램블링 알고리즘의 지식이나 전체 역 설계를 요구함이 없이도, 채널 데이터를 스크램블/디스크램블하는 과정을 제공한다. 상기 과정은 호스트 메모리 컨트롤러에 의해 제공되는 어떠한 에러 정정 코드(ECC) 알고리즘과는 독립적이다.

도면의 간단한 설명

[0009] 도 1은 호스트 시스템과 SSD간의 메모리 채널을 스크램블링/디스크램블링 하는 본 발명의 일 실시 예를 도시하는 블록도이다.

도 2는 호스트 시스템의 전형적인 스크램블링 알고리즘의 동작을 도시하는 도면이다.

도 3은 SSD를 위한 메모리 채널을 디스크램블링/스크램블링하는 과정을 도시하는 플로우챠트이다.

도 4a 및 4b는 SSD 내의 디스크램블링 알고리즘에 의해 수행되는 과정을 도시하는 도면들이다.

발명을 실시하기 위한 구체적인 내용

[0010] 본 발명의 개념의 실시 예들에 대한 참조가 이제 상세히 만들어질 것이다. 본 발명의 실시 예들은 첨부한 도면들 내에서 도시되며, 도면들 내에서 같은 참조부호들은 전반에 걸쳐 같은 구성요소들을 가리킨다. 실시 예들은 도면들을 참조로, 본 발명의 기본적인 개념을 설명하기 위하여 이하에서 개시된다.

[0011] 본 발명의 이점들 및 특징들과 이들을 달성하는 방법들은 이하의 실시 예 들의 상세한 설명 및 첨부 도면들을 참조함에 의해 보다 쉽게 이해될 것이다. 본 발명의 기본적인 개념은 그러나, 여러 다른 형태들로 구체화될 수 있으므로, 여기에 언급된 실시 예들에 국한되어 해석되어서는 안될 것이다. 오히려, 이들 실시 예들은, 본 분야의 통상의 지식을 가진 자에게 있어 이 개시가 철저하고 완전하게 되어 본 발명의 개념을 완전히 전달할 수 있도록 하기 위해 제공된다. 본 발명의 기본적인 개념은 첨부되는 클레임들에 의해서만 정의될 것이다. 도면들 내에서 층들의 두께 및 영역들은 명확성을 위해 과장될 수 있다.

[0012] 용어들 "하나(a)", "한(an)", 및 "그(the)"의 사용과 발명(특히 클레임들의 문맥에서)을 기술하는 문맥 내에서의 유사한 참고(referents)는 달리 표시되거나 문맥에 의해 명확히 모순되지 않는 한, 단수 및 복수 모두를 커버하는 것으로 해석된다. 용어들 "포함(comprising)", "가짐(having)", "포함(including)", 및 "보유(containing)"은 달리 기재되지 않는 한, 오픈 종결(open-ended) 용어들(예를 들어 포함하지만 그에 한정되지 않는다는 의미)로서 해석된다.

[0013] 여기에 사용된 것으로서, 용어 "알고리즘"또는 "모듈"은 소프트웨어나 하드웨어 구성요소에 제한되지 않는다. 하드웨어 구성요소에는 특정한 타스크들을 수행하는 field programmable gate array (FPGA) 또는 application specific integrated circuit (ASIC)이 포함될 수 있다. 알고리즘 또는 모듈은 어드레스가능 스토리지 매체 내에 상주하도록 적절히 구성될 수 있고 하나 이상의 프로세서들을 실행하기 위해 적절히 구성될 수 있다. 따라서, 알고리즘 또는 모듈은 예시적 방법으로, 구성요소들을 포함할 수 있다. 상기 구성요소들은 소프트웨어 구성요소들(software components), 객체 지향 소프트웨어 구성요소들(object-oriented software components), 클래스 구성요소들 및 타스크 구성요소들(class components and task components), 프로세스들(processes), 기능들(functions), 속성들(attributes), 프로시져들(procedures), 서브루틴들(subroutines), 프로그램 코드의 세그먼트들(segments of program code), 드라이버들(drivers), 펌웨어들(firmware), 마이크로코드(microcode), 회로들(circuitry), 데이터(data), 데이터 베이스들(databases), 데이터 구조들(data structures), 테이블들(tables), 어레이들(arrays), 및 변수들(variables)을 포함할 수 있다. 구성요소들에 대해 제공된 기능 및 구성요소들 또는 모듈들은 더 작은 구성요소들 또는 모듈들로 결합될 수 있거나 부가적인 구성요소들 및 구성요소들 또는 모듈들로 더 분리될 수 있다.

[0014] 달리 정의되지 않는 한, 여기에 사용되는 모든 기술적 및 과학적 용어들은 본 발명이 속하는 본 분야의 통상의 지식을 가진 자에 의해 통상적으로 이해되는 것과 같은 의미를 가진다. 어느 및 모든 예들의 사용, 또는 여기에 제공된 예시적인 용어들은 본 발명을 더 명확히 하기 위한 의도로 사용된 것이며, 달리 특정되지 않는 한 본 발명의 범위를 제한하는 것은 아니다. 더욱이, 달리 정의되지 않는 한 일반적으로 사용되는 사전들 내에 정의된

모든 용어들은 지나치게 엄격히 해석되지 않을 수 있다.

[0015] 본 발명은 메모리 장치 통신 프로토콜을 위한 XOR 기반 스크램블러/디스크램블러를 제공한다. 본 발명은 스크램블링의 지식이나 역 설계 없이, 그리고 호스트 시스템에 의해 적용되는 ECC 방법들 없이 상기 스크램블러/디스크램블러를 제공한다. 여기에 개시된 방법은 스크램블링/에러 정정이 발생한 경우에 다른 종류의 메모리 장치들이나 메모리 채널들에 대해서도 응용 가능하다. 그러나 여기에 개시된 방법은, 메모리 장치가 solid-state drive (SSD)를 포함하는 경우의 실시 예에 대하여 설명될 것이다. 상기 SSD는 예를 들면 싱글, 더블, 또는 쿼드 레이트 폼 팩터 (quad rate form-factor) SSD 일 수 있다.

[0016] SSD 트레이닝 모드 동안, 스크램블링 마스크 값이 트레이닝 시퀀스의 사용을 통해 발견되고 SSD 내에 저장된다. 상기 스크램블링 마스크 값은 배타적 OR(XOR) 벡터로서 또한 칭해지며, SSD 내의 각 메모리 위치 (each memory location)에 대해 데이터를 스크램블하기 위하여 호스트 시스템에 의해 사용된다. 결국, 상기 스크램블링 마스크 값들 혹은 XOR 벡터들은 SSD의 기능 모드 동안에 메모리 채널을 통해, 수신되는 데이터를 디스크램블하고 전송되는 데이터를 스크램블하기 위해 사용된다.

[0017] 도 1은 호스트 시스템과 DDR 폼 팩터 SSD간의 DDR 채널을 스크램블링/디스크램블링 하는 본 발명의 일 실시 예의 시스템을 도시하는 블록도이다. 도 1의 시스템은 호스트 시스템에 의해 인가되는 스크램블링 알고리즘의 선행 지식이나 역 설계를 요구하지 않는다.

[0018] 시스템은 메모리 채널 (14)을 통해 SSD (12)에 연결된 호스트 시스템 (10)을 포함한다. 잘 알려진 바와 같이 SSD는 데이터를 저장하기 위한 기계적 움직임 부분이 없으며, 데이터를 유지하기 위해 지속적인 전력을 필요로 하지도 않는다. 본 발명에 관련된 호스트 시스템 (10)의 구성 요소들은 프로세서 (22), 운영 체제 (OS) (24), 및 메모리 채널 (14)에 연결된 메모리 컨트롤러 (16)를 포함한다. 메모리 컨트롤러 (16)는 디지털 회로이다. 디지털 회로는 데이터의 흐름을 관리한다. 데이터의 흐름은 SSD (12)로서 도시된 바로서, 호스트의 메모리로부터 오는 데이터와 호스트의 메모리로 가는 데이터의 흐름을 말한다. 메모리 컨트롤러(16)는 프로세서(22)와 같은, 별도의 칩으로 구현되거나 다른 칩에 통합될 수 있다. 메모리 컨트롤러 (16)는 스크램블링/디스크램블링 알고리즘 (28)에 의해 수행되는 메모리 스크램블링 기능을 부가적으로 제공할 수 있다. 상기 알고리즘 (28)은 메모리로 전송되는 사용자 데이터를 의사 랜덤 패턴 (pseudo-random patterns)으로 변환하는 알고리즘이다.

[0019] 일 실시 예에서 SSD (12)가 더블 데이터 레이트 (DDR) 폼팩터 SSD로 구현되는 경우, 메모리 채널 (14)은 DDR 메모리 채널을 포함한다. DDR 폼 팩터 SSD는 메모리 버스 클럭 신호들의 상승 및 하강 에지 모두에서 데이터를 전송함으로써 DDR 메모리 채널의 사용을 배가 할 수 있다.

[0020] SSD (12)의 주요 구성 요소는 스토리지 컨트롤러 (18)와 호스트 시스템 (10) 으로부터 데이터를 저장하는 메모리 (21)이다. 스토리지 컨트롤러 (18)는 메모리 (21)내에 저장된 데이터를 관리한다. 스토리지 컨트롤러 (18)는 통신 프로토콜을 개재하여 메모리 채널 (14)를 통해 호스트 시스템과 통신한다. 메모리(21)는 임의의 유형의 비휘발성 랜덤 억세스 메모리(NVRAM)를 포함할 수 있다. 비 휘발성 랜덤 억세스 메모리(NVRAM)는 플래시 메모리, 강유전성 RAM (F-RAM), 자기저항 RAM (MRAM), 상 변화 메모리 (PCM), 밀리페드 메모리 (millipede memory), 기타 등을 포함할 수 있다.

[0021] 도 2는 호스트 시스템의 전형적인 스크램블링 알고리즘(28)의 동작을 도시하는 도면이다. 동작 동안에, 호스트 메모리 컨트롤러 (16)는 SSD (12)내에 저장될 데이터 (206)를 수신한다. 호스트 메모리 컨트롤러 (16)는 고정 시드 및 메모리 어드레스 (200)을 상기 데이터 (206)와 함께 수신한다. 스크램블링/디스크램블링 알고리즘 (28)은 각 DDR 채널 사이클에 대해 랜덤 시드(예를 들어 비트 패턴)을 생성한다. 상기 랜덤 시드는 본 명세서에서 XOR 벡터 (202) 로 칭해진다. 일 실시 예에서, XOR 벡터 (202)는 선형 피드백 시프트 레지스터(LFSR : 204)를 이용하여 생성될 수 있다. 혹은 XOR 벡터 (202)는 고정 시드 또는 고정 시드와 전송의 메모리 어드레스 (200)의 조합을 이용하는 기타 메커니즘을 이용하여 생성될 수 있다. 스크램블링/디스크램블링 알고리즘 (28)은 배타적 OR (XOR) 연산 (208)을 수행한다. XOR 연산은 XOR 벡터 (202)와 메모리 채널 (14)로 들어오는 데이터 (206)에 대해 수행된다. XOR 연산은 스크램블드 데이터 (30)를 생성하기 위하여 주어진 사이클 내에서 수행된다. LFSR (204)는 XOR 벡터 (202)를 생성하기 위해 피드백 다항식을 이용할 수 있다. 스크램블드 데이터 (30)는 메모리 채널 (14)을 통해 SSD (12)로 보내지고, 언스크램블(unscrambled) 되어 메모리 (21)에 저장된다.

[0022] SSD (12)로부터 사용자 데이터와 상태 정보가 호스트로 올바르게 다시 전송되도록 하기 위해, 호스트 메모리 컨트롤러 (16)에 의해 사용된 것과 같은 변환이 SSD (12)내에서 적용되어야 한다. 이 것은 다양한 호스트 시스템들 내에 사용된 스크램블링 알고리즘을 역 설계하기 위한 SSD 제조사(manufacturers)를 요구한다. 이 것은 SSD

가 그 호스트 시스템 (10)과 호환(양립)될 수 있도록 하기 위한 것이다. 또한, 호스트 스크램블링 알고리즘이 SSD 제조사에 의해 알려져 있다고 하더라도, 스크램블링 알고리즘은 호스트 업체들/플랫폼들 마다 다를 수도 있다. 스크램블링 알고리즘이 다르면 상이한 스크램블링/디스크램블링 방식들이 SSD (12)내에 구현되어야 한다.

[0023] 본 명세서에 기재된 예시적인 실시 예들은 호스트 메모리 컨트롤러에 의해 적용되는 스크램블링의 역설계(리버스 엔지니어링) 혹은 지식이 없이도, SSD 에 대한 메모리 채널을 디스크램블/스크램블하는 방법 및 시스템을 제공한다. 예시적인 실시 예들은 호스트 시스템 (10)상에 설치되고 실행되는 스토리지 디바이스 드라이버 (26)를 활용한다. 또한 실시 예들은 SSD 스토리지 컨트롤러 (18)에 의해 실행되는 디스크램블링/스크램블링 알고리즘 (32)를 이용한다. SSD 스토리지 컨트롤러 (18)는 호스트 시스템 (10)으로부터 수신된 스크램블드 데이터 (30)를 디스크램블한다. SSD 스토리지 컨트롤러 (18)는 메모리 (21)에서 호스트 시스템 (10)으로 보내질 출력 데이터를 스크램블한다. 잘 알려진 바와 같이, 디바이스 드라이버는 컴퓨터 프로그램이다. 컴퓨터 프로그램은 부착된 하드웨어 장치(이 경우 에는 SSD 12)로 소프트웨어 인터페이스를 제공한다. 하드웨어 장치는 운영 체제 (24) 및 다른 컴퓨터 프로그램을 인에이블한다. 상기 다른 컴퓨터 프로그램은 장치의 하드웨어 기능들을 억세스 하기 위한 프로그램이다.

[0024] 데이터 (206)와 XOR 된 XOR 벡터 (202)가 고정된 시드 혹은 고정된 시드와 전송의 메모리 어드레스 (200)와의 조합에 근거될 수 있으므로, SSD (12)상의 어떤 주어진 물리적 메모리 위치에 대해, 스토리지 디바이스 드라이버 (26)는 모든 메모리 위치들에 대한 제로 데이터의 트레이닝 시퀀스에 근거하여 결정되도록 구성된다. 트레이닝 시퀀스는 각 메모리 위치에 대한 호스트의 스크램블링/디스크램블링 알고리즘 (28)에 의해 사용되는 XOR 벡터 (202)에 대응한다. XOR 벡터들 (202)은 메모리 채널 (14)을 통해 스토리지 컨트롤러 (18)로 전송되고, XOR 벡터들 (34)로서 저장된다.

[0025] 예시적인 실시 예의 일 양태에 따르면, SSD (12)은 또한, 호스트 시스템 (10)으로부터 찾은 XOR 벡터들 (34)를 저장하는 벡터 스토리지 메모리 (20)를 구비한다. 주어진 메모리 어드레스에서 메모리 (21)내에 저장된 데이터는 언스크램블된다. 상기 데이터의 언스크램블은 해당 메모리 어드레스에 대해 상기 벡터 스토리지 메모리 (20)내에 저장된 XOR 벡터들 (34)를 이용하여 검색하는 동안에 수행된다.

[0026] 일 실시 예에서, 벡터 스토리지 메모리 (20)는 SRAM 메모리 또는 레지스터들의 세트로서 구현될 수 있다. 일 실시 예에서, 벡터 스토리지 메모리 (20)는 스토리지 컨트롤러 (18)의 내부에 위치될 수 있다. 혹은 벡터 스토리지 메모리 (20)는 스토리지 컨트롤러 (18) 및 메모리 (21)의 모두로부터 분리되어 위치될 수 있다. 또한, 벡터 스토리지 메모리 (20)는 메모리 (21)의 일부로서 위치될 수 있다.

[0027] 도 3은 SSD를 위한 메모리 채널을 통해 전송된 데이터를 디스크램블링/스크램블링하는 과정을 도시하는 플로우챠트이다. 과정은 XOR 벡터 (202)를 찾기 위해 SSD (12)에 대해 트레이닝 모드를 실행함에 의해 시작될 수 있다. 상기 XOR 벡터 (202)는 데이터를 스크램블하기 위해 호스트 시스템에 의해 사용된다. 상기 XOR 벡터 (202)의 찾음은 스크램블드 트레이닝 데이터 (30)를 생성하기 위해 상기 SSD (12)의 모든 메모리 위치들에 대해 모든 제로 트레이닝 데이터를 스크램블링 알고리즘 (28)에 입력함에 의해 수행된다. 상기 스크램블드 트레이닝 데이터 (30)는 메모리 채널 (14)을 통해 메모리 장치(21)로 전송되며, 상기 스크램블드 트레이닝 데이터(30)는 그 메모리 위치들에 대응하는 XOR 벡터들과 같다(블록 300). 일 실시 예에서, 트레이닝 모드는 초기화될 수 있고 시스템 (10)상의 스토리지 디바이스 드라이버 (26)에 의해 수행될 수 있다.

[0028] 상기 SSD (12)는 메모리 채널을 통해 상기 스크램블드 트레이닝 데이터를 수신하고, 상기 스크램블드 트레이닝 데이터를 상기 대응 메모리 위치들의 각각에 대한 XOR 벡터들 (34)로서 저장한다 (블록 302).

[0029] 일 실시 예에서, SSD 스토리지 컨트롤러 (18)는 상기 스크램블드 트레이닝 데이터를 수신할 수 있다. 상기 디스크램블링/스크램블링 알고리즘 (32)는 XOR 벡터들을 저장할 수 있다. 일 실시 예에 따라, SSD (12)의 벡터 스토리지 메모리 (20)는 트레이닝 모드 동안에 검색된 상기 XOR 벡터들 (34)을 저장하기 위해 사용된다. SSD (12)내에서 각각의 메모리 위치들에 대한 XOR 벡터들 (34)이 일단 검색되고 저장되면, 상기 스토리지 디바이스 드라이버 (26)는 트레이닝 모드가 완료되는 SSD (12)를 시그날링할 수 있다.

[0030] 상기 트레이닝 모드 후 상기 메모리 장치의 기능 모드 동안, 상기 메모리 장치는, 지정된 메모리 위치에 대한 메모리 채널을 통해 스크램블드 데이터를 수신한다. 상기 메모리 장치는 상기 특정 메모리 위치들로의 기입에 앞서, 상기 데이터를 디스크램블하기 위하여 상기 특정 메모리 위치에 대해 저장된 XOR 벡터를 사용한다(블록 304).

[0031] 일 실시 예에서, SSD 스토리지 컨트롤러 (18)는 상기 스크램블드 데이터를 수신할 수 있다. 상기 디스크램블링/스크램블링 알고리즘 (32)는 상기 데이터를 디스크램블 하기 위해 XOR 벡터들(34)을 사용할 수 있다.

[0032] 상기 기능 모드 동안에 상기 특정 메모리 위치로부터 데이터가 독출될 때, 상기 메모리 장치는 데이터를 출력한다. 상기 메모리 장치는 데이터를 스크램블하기 위해 상기 특정 메모리 위치에 대해 저장된 XOR 벡터를 사용한다. 상기 데이터의 사용은 상기 메모리 채널을 통해 상기 호스트 시스템으로 상기 스크램블드 데이터를 전송하기 이전이다 (블록 306).

[0033] 도 4a 및 4b는 SSD (12)내의 디스크램블링/스크램블링 알고리즘 (32)에 의해 수행되는 과정을 도시하는 도면들이다. 도 4a는 스크램블드 데이터가 호스트 시스템으로부터 수신될 때를 나타내고, 도 4b는 메모리로부터의 스크램블링 데이터가 호스트 시스템으로 보내질 때를 나타낸다. 도 4a 및 4b 모두에서, 디스크램블링/스크램블링 알고리즘 (32)은 수신된 데이터와 메모리 버스 어드레스에 대해 동작한다. 도 4a에서, 수신된 데이터 (400)는 호스트 시스템 (10)으로부터 스크램블된 데이터를 나타낸다. 디스크램블링/스크램블링 알고리즘 (32)은 상기 벡터 스토리지 메모리 (20)로부터의 상기 특정 메모리 위치에 대한 상기 XOR 벡터 (404)를 출력하기 위해 메모리 버스 어드레스 (402)를 사용한다. 디스크램블링/스크램블링 알고리즘 (32)은 디스크램블드 데이터 (412)를 생성하기 위해 상기 XOR 벡터 (404)와 상기 수신된 데이터 (400)에 대해 배타적 논리합 연산 (410)을 수행한다.

[0034] 도 4b에서, 수신된 데이터 (400')는 메모리 (21)으로부터 독출된 데이터를 나타낸다. 스크램블링/디스크램블링 알고리즘 (32)은 상기 벡터 스토리지 메모리 (20)로부터의 상기 특정 메모리 위치에 대한 상기 XOR 벡터 (404')를 출력하기 위해 메모리 버스 어드레스 (404)를 사용한다. 디스크램블링/스크램블링 알고리즘 (32)은 스크램블드 데이터 (414)를 생성하기 위해 상기 XOR 벡터 (404')와 상기 수신된 데이터 (400)에 대해 배타적 논리합 연산 (410)을 수행한다. 상기 연산은 메모리 채널 (14)을 통해 상기 호스트 시스템 (10)으로 상기 스크램블드 데이터 (414)를 전송하기 이전에 수행된다.

[0035] 예시적 실시 예들에 따라, 개시된 바와 같이 SSD (12)의 메모리 채널 (14)를 디스크램블링 및 스크램블링하는 과정은 호스트 메모리 컨트롤러 (16)에 의해 적용되는 스크램블링의 전체 역설계(리버스 엔지니어링) 혹은 지식이 없이도, 채널 데이터를 스크램블/디스크램블 한다. 또한, 상기 과정은 호스트 메모리 컨트롤러 (16)에 의해 적용되는 어떤 ECC 알고리즘과는 독립적이다.

[0036] 도 3에 대하여 상술된 과정에서, 기능적인 모드의 문맥 내에 사용된 용어 "데이터"는 호스트 시스템 (10)과 SSD (12) 간에 전송되는 다양한 종류의 데이터를 포함하는 것이다. 다양한 종류의 데이터는 컨트롤/디스크립터 데이터, 스테이터스 데이터, 유저 데이터, 및 컨트롤 데이터를 포함한다.

[0037] 컨트롤/디스크립터 데이터에서, 스크램블링/디스크램블링 알고리즘 (32)은 트레이닝 모드 동안에 각기 대응되는 컨트롤/디스크립터 메모리 위치에 대한 XOR 벡터를 저장할 수 있다. 스크램블링/디스크램블링 알고리즘 (32)은 상술한 바와 같이 기능 모드 동안에 컨트롤/디스크립터 데이터를 디스크램블링하기 위해 XOR 벡터들을 사용할 수 있다.

[0038] 스테이터스 데이터의 경우에, 스크램블링/디스크램블링 알고리즘 (32)은 트레이닝 모드 동안에 각기 대응되는 컨트롤/디스크립터 메모리 위치에 대한 XOR 벡터를 저장할 수 있다. 스크램블링/디스크램블링 알고리즘 (32)은 상술한 바와 같이 기능 모드 동안에 스테이터스 데이터를 스크램블링하기 위해 XOR 벡터들을 사용할 수 있다. ECC가 호스트 시스템 (10)상에서 인에이블되면, 스크램블링/디스크램블링 알고리즘 (32)은 스크램블드 데이터를 ECC 코드워드들에 덧붙일 수 있다.

[0039] 유저 데이터 및 컨트롤 데이터의 경우에, 스크램블링/디스크램블링 알고리즘 (32)은 트레이닝 모드 동안에 각기 대응되는 유저 데이터 메모리 위치에 대한 XOR 벡터를 저장할 수 있다. 스크램블링/디스크램블링 알고리즘 (32)은 상술한 바와 같이 기능 모드 동안 메모리 (21)의 리드 동안에 유저 데이터 및 컨트롤 데이터를 스크램블링하기 위해 XOR 벡터들을 사용할 수 있다. 기능 모드 동안 메모리 (21)로의 쓰기를 위해, 디스크램블링/스크램블링 알고리즘 (32)은 상술한 바와 같이 상기 메모리 (21)로의 쓰기 이전에, 유저 데이터 및 컨트롤 데이터를 디스크램블한다. 독출 유저 데이터 및 쓰기 유저 데이터가 동일한 메모리 위치를 공유할 수 있거나 다른 메모리 위치를 사용할 수 있음이 이해될 것이다. 독출 및 쓰기 유저 데이터에 대한 동일 메모리 위치를 공유하는 경우에, XOR 벡터들을 저장하는 스토리지 공간 요구는 감소될 수 있다.

[0040] 따라서, 벡터 스토리지 메모리 (20)는 서로 다른 종류의 데이터의 XOR 벡터들을 저장하기 위해 다른 영역들(미도시)로 구분될 수 있다. 일 실시 예에서, 벡터 스토리지 메모리 (20)는 컨트롤/디스크립터 데이터 벡터 영역, 스테이터스 데이터 벡터 영역, 유저 데이터 벡터 영역, 및 컨트롤 데이터 벡터 영역 중의 하나 이상을 포함할

수 있다.

[0041] 일 실시 예에서, 호스트 시스템 (10)이 인에이블되는 ECC 를 가진다면, 스토리지 장치 드라이버 (26)는 트레이닝 데이터로서 모든 제로 ECC 코드 워드들을 생성하도록 구성될 수 있다. 상기 트레이닝 데이터는 트레이닝 모드 동안에 스크램블링/디스크램블링 알고리즘 (28)으로 입력된다.

[0042] 대부분의 통상적인 싱글 에러 정정 및 더블 에러 검출(SECDED) ECC 구현은 모든 제로 데이터에 대해 모든 제로 코드 워드들을 초래하지만, 일부는 아닐 수 있다. 시스템 (10)의 ECC 구현이 모든 제로 데이터에 대해 모든 제로 코드 워드들을 초래하지 않는 경우의 상황에 있어서, 단계 300은 트레이닝 국면을 두 단계들로 분할할 수 있다.

[0043] 제1 단계에서, 스토리지 장치 드라이버 (26)는 호스트 메모리 컨트롤러 (16)의 스크램블링을 디세이블할 수 있다. 스토리지 장치 드라이버 (26)는, 트레이닝 데이터가 제로 또는 넌제로인지를 판정하기 위해 스크램블링/디스크램블링 알고리즘 (28)으로 모든 제로 트레이닝 데이터를 연달아 입력한다. 이에 따라 상기 디세이블이 수행된다. 다음으로, 각 데이터 사이클내의 ECC 코드워드 비트 위치들 및 모든 메모리 위치들에 대한 위치들의 값들은 스토리지 컨트롤러 (18) 및 시스템 펌웨어에 의해 기록된다.

[0044] 제2 단계에서, 스토리지 장치 드라이버 (26)는 호스트 메모리 컨트롤러 (16)의 스크램블링을 인에이블링한다. 스토리지 장치 드라이버 (26)는 제1 단계로부터의 ECC 코드워드들을 사용하여 모든 제로를 스크램블링/디스크램블링 알고리즘 (28)으로 트레이닝 데이터로서 보낸다. 디스크램블링/스크램블링 알고리즘 (32)은 벡터 스토리지 메모리 (20)내에 모든 메모리 위치들에 대한 XOR 벡터를 기록한다. 넌 제로 ECC 코드워드에 있어서, 이 과정은 트레이닝 동안 캡처된 대응 XOR 벡터 비트가 반전되어야 한다는 것을 나타내어야 한다. 반전 위치는 비 스크램블드 ECC 코드 워드가 1로 생성된 경우에 매 비트 위치로 된다.

[0045] 본 발명은 어떤 응용에도 일반적으로 적용될 수 있다. 그러한 응용은 넌-디램(non-DRAM) 메모리 장치가 어떤 스크램블드 채널에 부착된 경우 및 호스트와 정보 교환이 필요한 경우를 포함할 수 있다.

[0046] 일 실시 예에서, 디스크램블링/스크램블링 알고리즘 (32)은 소프트웨어 구성요소로 구현된다. 다른 실시 예에서, 디스크램블링/스크램블링 알고리즘 (32)은 하드웨어와 소프트웨어의 조합으로 구현된다. 비록 디스크램블링/스크램블링 알고리즘 (32), 스토리지 컨트롤러 (18), 및 벡터 스토리지 메모리 (20)가 단일 구성요소로서 도시되었으나, 각각의 기능은 작거나 큰 개수의 모듈들/구성요소들로 결합될 수 있다. 예를 들면, 일 실시 예에서, 디스크램블링/스크램블링 알고리즘 (32)은 디스크램블링 알고리즘을 위한 제1 소프트웨어 구성요소 및 스크램블링 알고리즘을 위한 제2 소프트웨어 구성요소로서 구현될 수 있다. 다른 실시 예에서, 디스크램블링/스크램블링 알고리즘 (32), 스토리지 컨트롤러 (18), 및 벡터 스토리지 메모리 (20)는 하나의 집적된 구성요소로서 구현될 수 있다.

[0047] 예시적 실시 예들의 스크램블링/디스크램블링 시스템은 고객에서 기업까지 넓은 범위의 스토리지 시장들에 적용될 수 있다. 이 시스템은 싱글 스탠드어론 머시인(single standalone machine)을 위한 디스크에 적용될 수 있다. 상기 디스크는 데스크탑 (desktop), 랩탑 (laptop), 워크스테이션 (workstation), 서버 (server) 및 기타 등등에 있는 디스크를 포함할 수 있다. 상기 시스템은 스토리지 어레이(storage array), 소프트웨어 정의 스토리지 (software-define storage: SDS), 응용 특정 스토리지 (application-specific storage), 가상 머시인 (virtual machine :VM), 가상 데스크탑 인프라스트럭쳐 (virtual desktop infrastructure :VDI), 컨텐트 분배망 (content distribution network :CDN), 및 기타 등등을 포함할 수 있다.

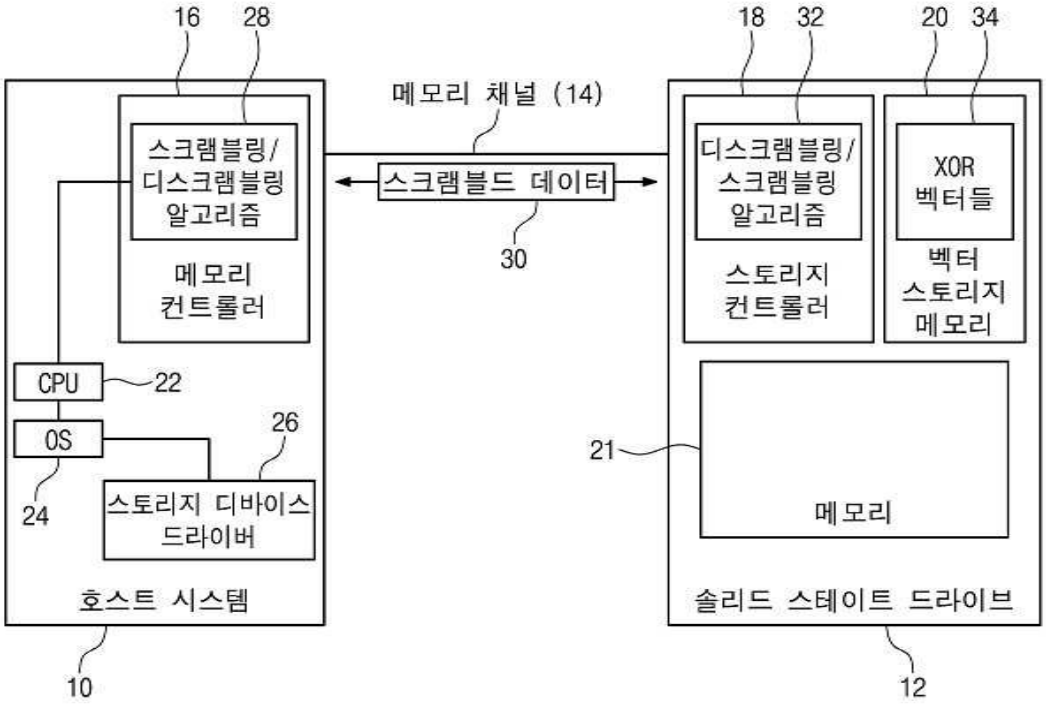
[0048] 일 실시 예에서, 예를 들면, SSD (12)의 메모리 (21)는 복수의 플래시 메모리들과 같은 복수의 비휘발성 메모리 칩들로 형성될 수 있다. 다른 예로서, 메모리 (21)는 플래시 메모리 칩들 대신에 다른 종류의 비휘발성 메모리 칩들로 형성될 수 있다. 다른 종류의 비휘발성 메모리 칩들은 예를 들어 PRAM, FRAM, MRAM, 또는 기타 메모리일 수 있다. 대안적으로 메모리 (21)는 DRAM 이나 SRAM 등과 같은 휘발성 메모리들로 형성될 수 있고, 둘 이상의 메모리들이 혼합된 경우에 하이브리드 타입을 가질 수 있다.

[0049] 메모리 장치의 메모리 채널을 경유하여 전송되는 데이터를 디스클램블링하고 스크램블링하는 방법 및 장치가 개시되었다. 본 발명은 보여지는 실시 예들에 따라 설명되었으나, 실시 예들에 대한 변형이 있을 수 있다. 본 발명의 사상과 범위 이내에서 다양한 변경들이 존재할 수 있다. 예를 들면, 예시적 실시 예가 하드웨어, 소프트웨어, 프로그램 명령들을 포함하는 컴퓨터 판독가능 매체, 또는 그들의 조합을 이용하여 구현될 수 있다. 본 발명에 따라 기록된 소프트웨어는 메모리, 하드 디스크와 같은 컴퓨터 판독가능 매체의 일부 형식으로 저장되거나 CD/DVD-ROM의 일부 형식으로 저장될 수 있다. 소프트웨어는 프로세서에 의해 실행될 수 있다. 따라서, 많은 변

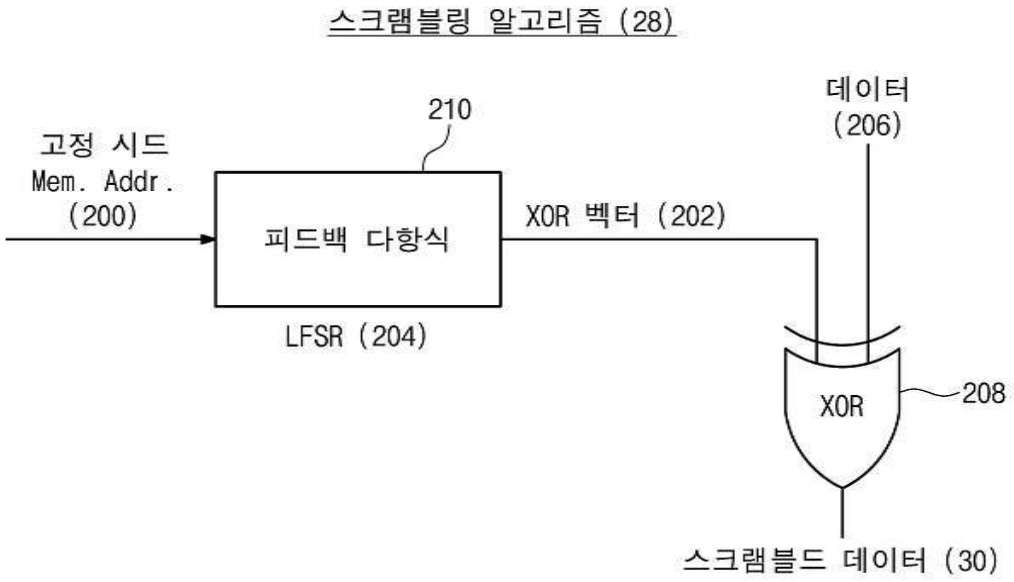
형이 첨부된 클레임들의 사상과 범위를 벗어남이 없이 본 분야의 통상의 기술을 가진 자에 의해 만들어질 수 있을 것이다.

도면

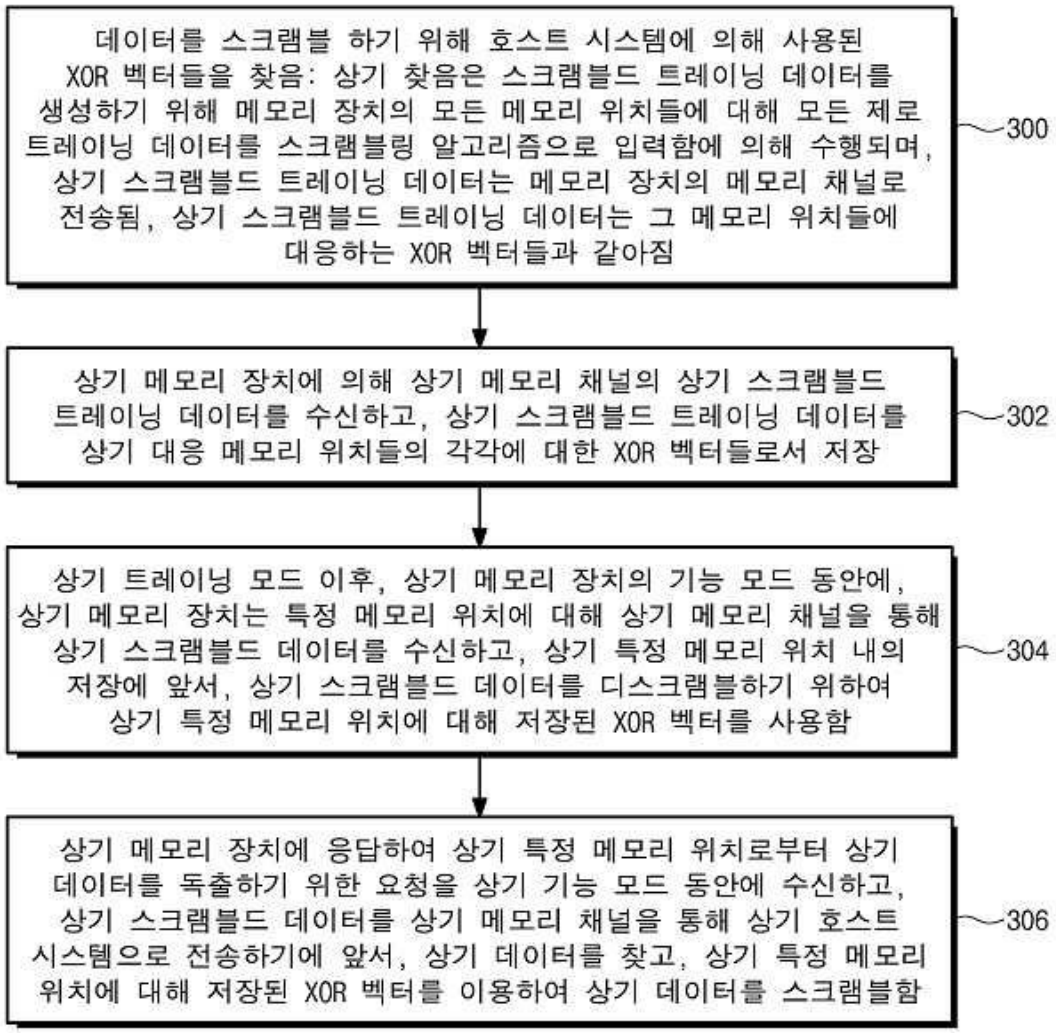
도면1



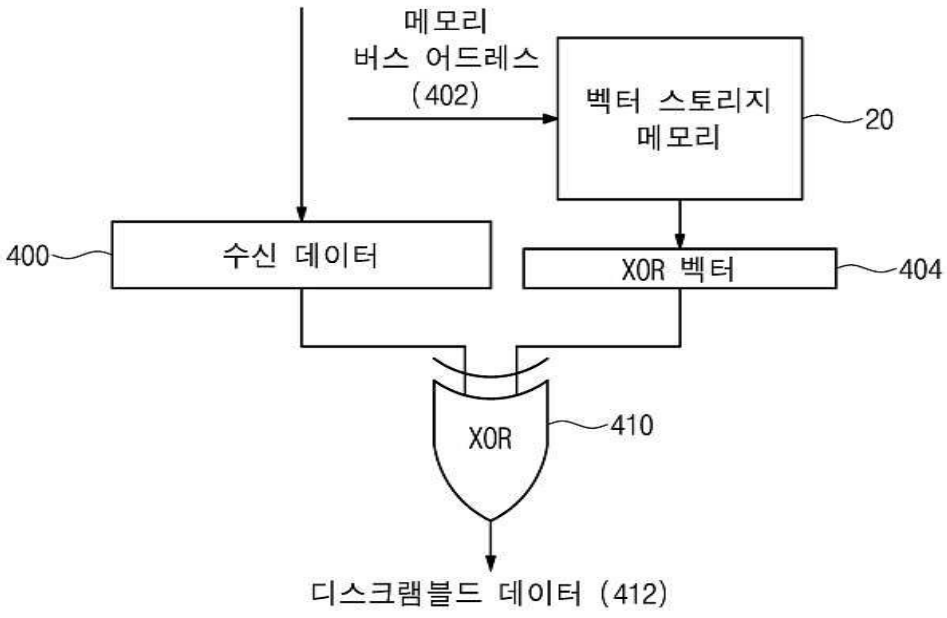
도면2



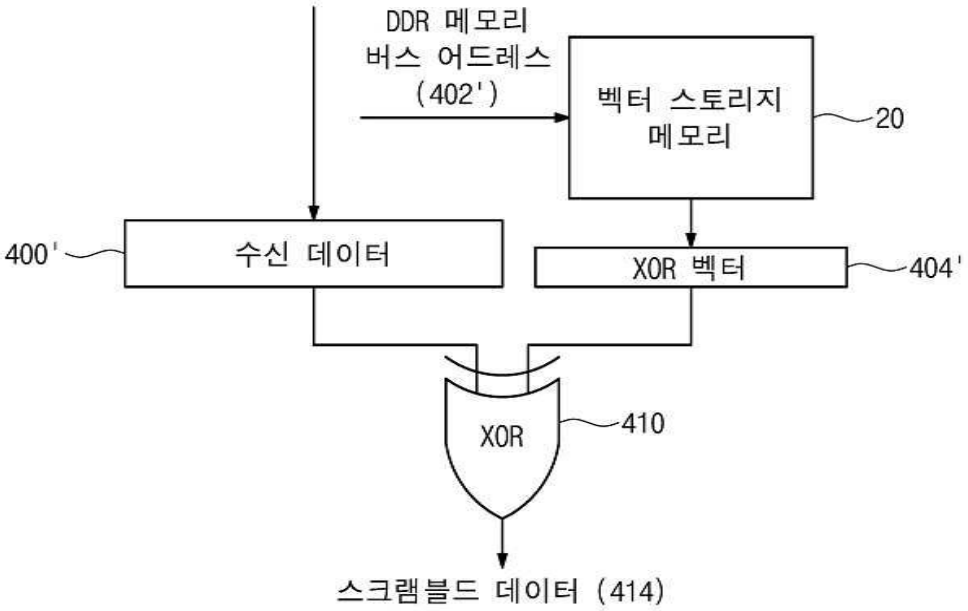
도면3


데이터를 스크램블 하기 위해 호스트 시스템에 의해 사용된 XOR 벡터들을 찾음: 상기 찾음은 스크램블드 트레이닝 데이터를 생성하기 위해 메모리 장치의 모든 메모리 위치들에 대해 모든 제로 트레이닝 데이터를 스크램블링 알고리즘으로 입력함에 의해 수행되며, 상기 스크램블드 트레이닝 데이터는 메모리 장치의 메모리 채널로 전송됨, 상기 스크램블드 트레이닝 데이터는 그 메모리 위치들에 대응하는 XOR 벡터들과 같아짐
300
상기 메모리 장치에 의해 상기 메모리 채널의 상기 스크램블드 트레이닝 데이터를 수신하고, 상기 스크램블드 트레이닝 데이터를 상기 대응 메모리 위치들의 각각에 대한 XOR 벡터들로서 저장
302
상기 트레이닝 모드 이후, 상기 메모리 장치의 기능 모드 동안에, 상기 메모리 장치는 특정 메모리 위치에 대해 상기 메모리 채널을 통해 상기 스크램블드 데이터를 수신하고, 상기 특정 메모리 위치 내의 저장에 앞서, 상기 스크램블드 데이터를 디스크램블하기 위하여 상기 특정 메모리 위치에 대해 저장된 XOR 벡터를 사용함
304
상기 메모리 장치에 응답하여 상기 특정 메모리 위치로부터 상기 데이터를 독출하기 위한 요청을 상기 기능 모드 동안에 수신하고, 상기 스크램블드 데이터를 상기 메모리 채널을 통해 상기 호스트 시스템으로 전송하기에 앞서, 상기 데이터를 찾고, 상기 특정 메모리 위치에 대해 저장된 XOR 벡터를 이용하여 상기 데이터를 스크램블함
306

도면4a


메모리 버스 어드레스 (402)
벡터 스토리지 메모리
20
400
수신 데이터
XOR 벡터
404
XOR
410
디스크램블드 데이터 (412)

도면4b


DDR 메모리
버스 어드레스
(402')
벡터 스토리지
메모리
20
400'
수신 데이터
XOR 벡터
404'
XOR
410
스크램블드 데이터 (414)