



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년10월06일

(11) 등록번호 10-1557736

(24) 등록일자 2015년09월30일

(51) 국제특허분류(Int. Cl.)

G06F 11/00 (2006.01) G06F 11/08 (2006.01)  
G06F 11/10 (2006.01)

(21) 출원번호 10-2014-7030099(분할)

(22) 출원일자(국제) 2008년05월16일

심사청구일자 2014년11월25일

(85) 번역문제출일자 2014년10월27일

(65) 공개번호 10-2014-0141684

(43) 공개일자 2014년12월10일

(62) 원출원 특허 10-2010-7000248

원출원일자(국제) 2008년05월16일

심사청구일자 2013년05월15일

(86) 국제출원번호 PCT/US2008/063938

(87) 국제공개번호 WO 2008/154119

국제공개일자 2008년12월18일

(30) 우선권주장

11/759,585 2007년06월07일 미국(US)

(56) 선행기술조사문헌

KR1020070001871 A\*

KR1020070024504 A\*

US6256642 A

\*는 심사관에 의하여 인용된 문헌

(73) 특허권자

마이크론 테크놀로지, 임크.

미국, 아이아호, 보이세, 사우스 페더럴 웨이  
8000

(72) 발명자

리, 티에니우

미국 95124 캘리포니아주 산 호세 키로 애비뉴  
3254

(74) 대리인

양영준, 백만기

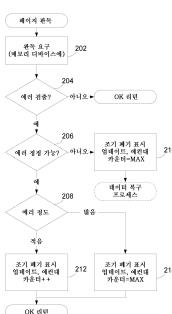
전체 청구항 수 : 총 6 항

심사관 : 정성용

(54) 발명의 명칭 새로운 불량 블록 검출

**(57) 요약**

NAND 플래시와 같은 비휘발성 집적 회로 메모리 디바이스(106)로부터 데이터를 판독하는 장치 및 방법에 관한 것이다. 예컨대, 개시된 기술은 오퍼레이팅 시스템(104)의 디바이스 드라이버(110)로 구현될 수 있다. 판독 동작 동안에 에러가 추적된다. 판독 동작 동안에 충분한 에러가 관측되면(204), 그 블록은 소거가 요구되거나 블록의 페이지에 대한 기입이 요구될 때 폐기된다(210, 212, 214, 310). 일 실시예는 정정할 수 없는 에러로부터 데이터를 복구하는 기술이다. 예컨대, 판독 모드는 데이터 복구를 시도하는 더욱 신뢰성 있는 판독 모드로 변경될 수 있다(410). 일 실시예는 또한 에러 정정 코드 데이터의 디코딩에 의해 데이터가 정정 가능한지의 여부(430)에 상관없이 메모리 디바이스로부터 데이터를 리턴시킨다(106).

**대 표 도 - 도2**

## 명세서

### 청구범위

#### 청구항 1

비휘발성 집적 회로 메모리 디바이스로부터 데이터의 페이지에 액세스하는 방법으로서, 상기 메모리 디바이스로부터 데이터의 페이지의 판독을 요구하는 단계; 상기 페이지와 연관된 에러 정정 코드들(ECC)을 검사하는 단계; 및 상기 데이터의 페이지가 어떤 정정 불가능한 에러들을 포함하고 있는지의 여부와 상관없이 상기 데이터의 페이지가 정확하게 판독된 것을 표시하는 상태를 오퍼레이팅 시스템에 제공하는 단계를 포함하는 방법.

#### 청구항 2

제1항에 있어서, 상기 디바이스로부터 데이터를 판독함에 있어서 적어도 제1 판독 동작 모드와 상기 제1 판독 동작 모드보다 느린 제2 판독 동작 모드가 선택 가능하고, 상기 페이지의 판독을 요구하는 단계는, 초기 판독의 디코딩으로 정정 불가능한 에러가 생긴 것으로 판정하는 단계 - 상기 초기 판독은 상기 제1 판독 동작 모드를 이용하여 실행됨 - ; 상기 제1 판독 동작 모드로부터 상기 제2 판독 동작 모드로 변경하는 단계; 및 상기 페이지의 또 다른 판독을 요구하는 단계를 더 포함하는 방법.

#### 청구항 3

제2항에 있어서, 상기 제1 판독 동작 모드는 직접 메모리 액세스(DMA)를 포함하고, 상기 제2 판독 동작 모드는 액세스 제어에 마이크로프로세서를 이용하는 것을 포함하는 방법.

#### 청구항 4

제2항에 있어서, 투프 카운터에 적어도 부분적으로 기초하여 상기 페이지의 또 다른 판독을 요구할지를 판정하는 단계를 더 포함하는 방법.

#### 청구항 5

제2항에 있어서, 타이머에 적어도 부분적으로 기초하여 상기 페이지의 또 다른 판독을 요구할지를 판정하는 단계를 더 포함하는 방법.

#### 청구항 6

제1항에 있어서, 오퍼레이팅 시스템의 드라이버에 의해 상기 방법을 실행하는 단계를 더 포함하는 방법.

### 발명의 설명

### 기술 분야

[0001] 본 발명의 실시예들은 일반적으로 접적 회로에 관한 것이다. 특히, 실시예들은 메모리 회로용 소프트웨어 또는 하드웨어에 관한 것이다.

### 배경기술

[0002] 플래시 메모리는 소거 가능하고 채프로그래밍 가능한 비휘발성 접적 회로 메모리의 일종이다. 플래시 메모리에서, 메모리 셀들은 소거를 위해 "블록들"로 배열된다. 블록은 소거된 후, 원한다면 프로그래밍 준비가 된다. NOR 플래시 메모리는 데이터 검색을 위해 개개의 바이트에 대한 액세스를 제공하나 비교적 밀도가 낮은 플래시 메모리의 일종이다.

[0003] NAND 플래시 메모리는 비교적 밀도가 높은 플래시 메모리의 일종이다. 직렬로 접속된 셀들의 컬럼들을 형성함으로써 부분적으로 고밀도를 달성한다. 또한, NAND 플래시 메모리의 경우, 데이터의 페이지와 같이 바이트의 비교적 큰 그룹으로 데이터를 프로그래밍 및 액세스한다. 예컨대, 페이지는 어레이의 로우 또는 로우의 일부에 대응할 수 있다.

[0004] 통상은 동시에 데이터의 "페이지"로 NAND 플래시 메모리 어레이에 대한 데이터 기입 및 판독을 행한다. 예컨대, 페이지는 2,048의 데이터 바이트와 64의 스페어 바이트로 이루어진 2,112 바이트를 가질 수 있다. 스페어 바이트는 통상은 에러 정정 코드(ECC), 웨어-레벨링(wear-leveling) 정보, 또는 다른 오버헤드 데이터용으로 이용된다. 에러 정정 코드는 저장된 데이터의 강건성을 증가시킨다. 통상적으로, 블록 코드의 형태를 이용하여 순환 잉여 검사(CRC) 체크섬, 해밍(Hamming) 코드, 리드-솔로몬(Reed-Solomon) 에러 정정 등의 에러 정정 코드를 생성한다. 이들 에러 정정 코드는 데이터 바이트의 판독 시에 에러가 있는지를 검출하고 통상은 그 에러가 에러 정정 코드의 능력을 벗어나지 않는다면 그 데이터 바이트에서 에러를 정정할 수 있다.

[0005] 플래시 메모리는 많이 이용된다. 예로는 플래시 메모리 하드 드라이브(하드 드라이브 자체), USB 플래시 드라이브 또는 썸(thumb) 드라이브, 모바일 폰, 디지털 카메라, 디지털 미디어 플레이어, 게임, 메모리 카드, 내비게이션 디바이스, 퍼스널 디지털 어시스턴트, 컴퓨터 등이 있다. 한도 내에서, 에러 정정 코드는 데이터 바이트의 데이터에서 많은 에러를 정정할 수 있다. 그러나, 그 한도를 벗어나면, 통상은 에러가 있는 데이터를 정정할 수 없다. 종래 기술의 한가지 단점은 에러를 정정할 수 없게 될 때까지 대개는 너무 늦는다는 점이다.

[0006] 데이터 저장에 플래시 메모리를 이용하는 다수의 디바이스는 또한 오퍼레이팅 시스템을 이용한다. 오퍼레이팅 시스템은 하드웨어와 다른 소프트웨어 간의 추상화 계층으로서 기능한다. 예컨대, 오퍼레이팅 시스템의 파일 시스템 및 디바이스 드라이버는 통상은 메모리 디바이스에 저장된 데이터에의 액세스를 제공한다. 오퍼레이팅 시스템 내에는 추가의 계층이 존재할 수 있다.

[0007] 도 1은 CPU(102) 형태의 마이크로프로세서, 오퍼레이팅 시스템(104), 메모리 디바이스(106)를 포함하는 처리 환경의 일부의 예를 도시한다. 메모리 디바이스(106)는 NAND 플래시 메모리 디바이스일 수 있다. 오퍼레이팅 시스템(104)은 파일 시스템(108)과 디바이스 드라이버(110)를 더 포함한다. 오퍼레이팅 시스템(104)은 1 이상의 파일 시스템과 1 이상의 디바이스 드라이버, 그리고 본 발명과는 관련없는 다른 컴포넌트들을 지원할 수 있다는 것을 알 것이다. 또한, DRAM과 같은 휘발성 메모리 디바이스(114)와 직접 메모리 액세스(DMA) 컨트롤러(116)가 도시되어 있다.

[0008] CPU(102)는 오퍼레이팅 시스템(104)의 코드를 포함하는 명령어들을 실행한다. 파일 시스템(108)의 코드는 메모리 디바이스(106)에 대한 논리적 어드레스와 같은 로우 레벨 정보와, 파일명 및 디렉토리와 같은 하이 레벨 정보 간의 추상화를 제공한다. 디바이스 드라이버(110)의 코드는 통상은 메모리 디바이스(106)에/로부터의 데이터 전송을 위한 로우 레벨 정보를 취급한다. 디바이스 드라이버(110)는 CPU(102)가 메모리 디바이스(106)에 직접 액세스(프로세서 입력/출력으로 알려짐)하기 위한 코드를 제공하거나, 메모리 디바이스(106)에/로부터의 데이터 전송을 위해 버스 제어를 취급하도록 DMA 컨트롤러(116)를 활성화하는 코드를 제공할 수 있다. DMA 컨트롤러(116)의 이용은 CPU(102)가 자유롭게 다른 태스크들을 취급할 수 있도록 한다.

[0009] 프로세서 입력/출력(PIO)과 DMA 입력/출력은 모두, 디바이스 드라이버(110)가 기입 동작 및 판독 동작과 관련된 ECC 정보를 취급할 수 있다. 다수의 오퍼레이팅 시스템에는, 각종 상이한 타입의 메모리 디바이스에 대한 판독 및 기입을 지원하는 다수의 디바이스 드라이버가 존재한다. 파일 시스템(108)에 의한 맵핑 이외에도, 다수의 NAND 플래시 메모리 디바이스는, 불량 블록 관리 및 웨어아웃(wear out) 관리에, 논리적 어드레스와 물리적 어드레스 간의, 플래시 변환 계층이라고 하는, 가상 맵핑을 이용한다.

## 도면의 간단한 설명

[0010] 이들 도면 및 관련 설명은 본 발명의 특정 실시예들을 예시하기 위한 것이며 제한하고자 하는 것은 아니다.

도 1은 CPU, 오퍼레이팅 시스템 및 메모리 디바이스를 포함하는 처리 환경을 도시한다.

도 2는 메모리 디바이스로부터 데이터의 페이지를 판독하기 위한 프로세스의 실시예를 일반적으로 예시하는 흐름도이다.

도 3은 열화된 블록의 사전 폐기(proactive retirement)를 위한 프로세스의 실시예를 일반적으로 예시하는 흐름도이다.

도 4는 정정할 수 없는 에러를 가진 것으로 판정된 데이터를 복구하기 위한 프로세스의 실시예를 예시하는 흐름도이다.

## 발명을 실시하기 위한 구체적인 내용

[0011] NAND 플래시와 같은 비휘발성 집적 회로 메모리 디바이스로부터 데이터를 판독하는 장치 및 방법에 관한 것이다. 예컨대, 이들 기술은 오퍼레이팅 시스템의 디바이스 드라이버로 구현될 수 있다. 판독 동작 동안에 에러의 적어도 일부 히스토리가 유지(예컨대, 추적)된다. 페이지에 대한 판독 동작은 통상은 대응하는 블록에 대한 소거 동작 또는 페이지의 프로그래밍보다 매우 자주 일어난다. 이들 판독 동작 동안에 만나고 추적되는 에러는 판독되는 페이지 및 블록의 신뢰성의 표시를 제공할 수 있다.

[0012] 판독 동작 동안에 충분한 에러가 판측되면, 그 블록은 폐기용으로 표시(예컨대, 지정)된다. 그 블록은 예컨대 그 블록에 대한 소거 또는 기입(프로그래밍)이 요청될 때 그 블록을 불량 블록 테이블에 부가함으로써 폐기될 수 있다.

[0013] 일 실시예는 정정할 수 없는 에러로부터 데이터를 복구하는 기술을 포함한다. 예컨대, 판독 모드는 데이터 복구를 시도하는 더욱 신뢰성 있는 판독 모드로 변경될 수 있다. 이 판독 모드는 덜 적극적인 타이밍을 가질 수 있다. 일 실시예는 또한 에러 정정 코드 데이터의 디코딩에 의해 데이터가 정정되었는지의 여부에 상관없이 메모리 디바이스로부터 데이터를 리턴시킨다. 통상적으로 크게 실패하는 하드 디스크 디바이스와는 대조적으로, 집적 회로 메모리 디바이스는 통상은 에러 정정이 실패할지라도 대부분의 데이터가 이용 가능하게 남겨지는 소량의 비트 실패를 갖는다.

[0014] 개시된 기술을 구현한 오퍼레이팅 시스템의 디바이스 드라이버는 소프트웨어로(코드로), 하드웨어로(모듈로), 또는 소프트웨어와 하드웨어의 조합으로 구현될 수 있다. 또한, NAND 플래시 메모리의 상황에서 기재하였으나, 본원에 기재한 원리 및 이점은 다른 형태의 비휘발성 집적 회로에도 적용 가능할 것이다. 본원에 기재한 이익 및 특징을 모두 제공하지 않는 실시예들을 포함한, 본 발명의 다른 실시예들은 당업자에게는 명백할 것이다.

[0015] 도 2는 메모리 디바이스(106)(도 1)로부터 데이터의 페이지를 판독하고 에러 히스토리를 유지하기 위한 프로세스를 일반적으로 예시하는 흐름도이다. 일 실시예에서, 이 프로세스는 오퍼레이팅 시스템의 디바이스 드라이버(110)(도 1)에 의해 실행되고, 메모리 디바이스(106)는 NAND 플래시 메모리 디바이스이다. 예시한 프로세스는 다양하게 변경될 수 있다는 것을 당업자는 알 것이다. 예컨대, 다른 실시예에서, 예시한 프로세스의 각종 부분들을 조합하거나, 엇갈린 순서로 재배열하거나, 제거하거나 할 수 있다. 프로세스의 시작에서, 대응하는 에러 정정 코드 데이터와 함께 메모리 디바이스에 데이터가 이미 저장되어 있는 것으로 한다.

[0016] 예시한 프로세스의 실시예는 일반적으로 메모리 디바이스(106)(도 1)의 블록의 단일 페이지를 판독하는 것과 관련하여 기재할 것이다. 카운터를 이용하여 블록에 대한 에러 히스토리를 유지한다. 메모리 디바이스(106)는 많은 블록을 가지며, 에러 추적을 위해 블록 당 하나의 카운터를 가질 수 있다. 일 실시예에서, 카운터 어레이를 이용하여 카운터들을 저장한다. 블록에 대한 카운터가 10의 값과 같은 미리 정해진 값일 수 있는 특정 임계 레벨 MAX에 도달할 때, 프로세스는 그 블록을 초기 폐기된 것으로 판정한다. 예컨대, 도 3과 관련하여 더욱 상세하게 후술하는 바와 같이 블록을 소거하거나 블록의 페이지를 기입할 때 블록을 폐기시킬 수 있다. 휘발성 또는 비휘발성 메모리를 이용하여 카운터들을 저장할 수 있다. 그러나, 강건성을 위해, 통상은 메모리 디바이스(106) 그 자체와는 상이한 디바이스에 그러한 종류의 정보를 저장하는 것이 바람직하다. 일 실시예에서, 파워업 시에 히스토리가 리셋되도록 파워업 시에 카운터들이 리셋된다. 이러한 리셋은 블록에서의 불시의 열화에 대하여 비교적 빠른 응답을 제공한다.

[0017]

프로세스는 메모리 디바이스(106)(도 1)의 특정 페이지에 대한 관독의 요구를 수신함으로써 시작한다. 그 요구는, 파일 시스템(108)(도 1)으로부터와 같이, 디바이스 드라이버(110)(도 1)보다는 오퍼레이팅 시스템(104)(도 1)의 상위 계층으로부터의 서브루틴 호출에 의한 것일 수 있다. 프로세스는 CPU(102)로 하여금 메모리 디바이스(106)와 통신하게 하는 머신 코드나 CPU(102)로 하여금 DMA 컨트롤러(116)에게 메모리 디바이스(106)와 통신하도록 지시하게 하는 명령어들과 같은 로우 레벨 명령어들을 이용하여 특정 페이지를 요구한다(202). 메모리 디바이스(106)는 메모리 어레이(들)에 내부적으로 액세스하여 페이지에 대한 데이터를 제공한다. ECC 데이터를 디코딩하여 에러가 존재하는지를 검출하고, 또한 에러가 존재하고 정정할 수 있다면 에러를 정정한다. ECC의 디코딩은 소프트웨어 기능이거나 하드웨어 기능일 수 있다. 일 실시예에서, ECC의 인코딩 및 디코딩은 디바이스 드라이버(110)에 의해 실행되는 기능이다. 다른 실시예에서, ECC의 인코딩 및 디코딩은 DMA 컨트롤러(116)의 기능이다.

[0018]

프로세스는 페이지 관독 시에 하나 이상의 에러가 검출되었는지의 여부를 판정한다(204). 에러는 정정 가능할 수도 그렇지 않을 수도 있다. 에러 검출 프로세스 동안에 에러를 만나지 않은 경우, 프로세스는 (그 페이지에 대응하는 블록에 대한 카운터를 업데이트하지 않고) 나와, 파일 시스템(108)(도 1)과 같은 디바이스 드라이버(110)(도 1)의 호출 루틴으로 "OK" 상태를 리턴시킨다. 요구된 데이터는 메모리 디바이스(106)(도 1)의 출력 버퍼에서 이용 가능하고, 데이터 버스에 대한 논리적 어드레스에서, 다른 버퍼 등의 다른 위치로 전달될 수 있다.

[0019]

에러(들)이 발생한 것으로 프로세스가 판정하면(204), 프로세스는 에러 정정 코드(ECC) 데이터의 디코딩에 의해 에러(들)을 정정할 수 있는지의 여부를 판정한다(206). ECC의 한도 내에서 에러(들)을 정정할 수 있다면, 프로세스는 정정된 에러(들)이 어느 정도인지를 판정한다(208). 예컨대, 일 실시예에서, 프로세스는 정정되지 않은 데이터를 정정된 데이터와 비교하여 에러수의 카운트를 취득한다. 이와 달리, 에러 정정 코드에 의해 정정하기에는 에러가 너무 많은 경우에는, 프로세스는 그 블록을 조기 폐기된 것으로 표시한다(210). 일 실시예에서, 카운터를 임계값 MAX로 설정하여 조기 폐기의 표시를 제공한다. 블록을 조기 폐기된 것으로 표시하는 것은 실제로 블록을 폐기시킨 것과 동일하지 않다.

[0020]

블록을 불량 블록 테이블에 부가하는 것과 같이 블록을 불량으로 마킹함으로써 블록을 폐기시킬 수 있다. 불량 블록 테이블은 통상은 메모리 디바이스(106)(도 1) 그 자체에 저장된다. 그러나, 블록으로부터의 데이터가 더 이상 요구되지 않을 때까지, 블록은 아직 불량으로 마킹되지 않을 것이다. 블록을 불량으로 마킹하는 것은 블록에 대한 관독 액세스를 방지할 것이다. 따라서, 일 실시예에서, 프로세스는 도 3과 관련하여 보다 상세히 후술하는 바와 같이 소거 동작이 요구되거나 기입 동작이 요구될 때까지 블록을 불량으로 마킹하는 것을 기다린다. 일 실시예에서, 프로세스는 파일 시스템(108)(도 1)과 같은 호출 루틴으로 에러를 리턴시키고, 블록을 폐기시키는 절차를 개시한다.

[0021]

일 실시예에서, 프로세스는 조기 폐기 표시 상태(210)로부터 데이터 복구를 시도하는 데이터 복구 프로세스로 진행한다. 그러한 데이터 복구 프로세스의 일 실시예에 대해서는 도 4와 관련하여 보다 상세히 후술할 것이다. 다른 실시예에서, 프로세스는 조기 폐기 표시 상태(210)로부터 전진하여, 파일 시스템(108)(도 1)과 같은 디바이스 드라이버의 호출 루틴으로 "관독 에러" 상태를 리턴시키고, 나간다.

[0022]

에러(들)을 정정할 수 있는지를 판정하는 단계(206)를 다시 참조하여, 에러(들)을 정정할 수 있다면, 프로세스는 ECC가 정정한 에러(들)이 어느 정도인지를 판정한다(208). 비교적 적은 것으로 또는 비교적 많은 것으로 간주되는 에러수는 특정 메모리 디바이스(106)(도 1)의 특성 및 페이지 사이즈에 의존할 수 있다. 당업자는 적절한 임계값을 쉽게 판정할 수 있을 것이다. 또한, 예시한 판정 단계(208)는 바이너리, 즉 2개의 가능한 결정이지만, 하나 이상의 중간 레벨도 이용 가능하다는 것을 알 것이다.

[0023]

일 실시예에서, 2048 바이트 페이지(ECC를 포함하지 않음)인 경우, 페이지에 정정 가능한 에러가 1개 있는 것을 "적다"라고 하고, 정정 가능한 에러가 2개 이상 있는 것을 "많다"라고 한다. 이 예에서, 페이지에 정정 가능한 에러가 1개 있는 경우, 프로세스는 판정 단계(208)로부터 카운터 증분 상태(212)로 진행하고, 정정 가능한 에러가 1개보다 많은 경우에, 프로세스는 판정 단계(208)로부터 조기 폐기 표시 상태(214)로 진행한다.

[0024]

카운터 증분 상태(212)에서, 프로세스는 판독되는 페이지의 특정 블록에 대해 카운트 1씩 카운터를 증분시킨다. 카운터 어레이는 메모리 디바이스(106)(도 1)의 다수의 블록에 대한 각종 카운트, 예컨대 메모리 디바이스(106)의 블록 당 하나의 카운터를 저장할 수 있다. 블록이 많은 페이지를 포함하면, 다수의 페이지로부터의 에러가 특정 블록에 대한 카운트에 기여할 수 있다. 프로세스는 디바이스 드라이버(110)(도 1)의 호출 루틴, 예컨대 파일 시스템(108)(도 1)으로 "OK" 상태를 리턴시키고, 나간다. 판측되는 바와 같이, 블록의 동일한 페이지

또는 다른 페이지들이 에러와 함께 반복적으로 판독되면, 블록에 대한 카운터에 저장되는 카운트는 증가할 것이다. 카운트는 블록의 페이지(들)을 판독하면서 만나는 에러 히스토리를 저장한다.

[0025] 비교적 많은 에러, 예컨대 2개 이상의 에러를 만난 경우에 이용하는 판정 단계(208)의 경로를 다시 참조하면, 조기 폐기 표시 상태(214)에서, 프로세서는 카운터를 업데이트하여 블록을 조기 폐기된 것으로 지정한다. 예컨대, 카운터는 상태(210)과 관련하여 기재하는 바와 같이 임계 레벨 MAX로 설정될 수 있다. 예시한 실시예에서, 상태(210) 및 상태(214)는 각각 카운터를 임계 레벨 MAX로 설정하여 블록을 조기 폐기된 것으로 지정한다. 다른 실시예에서, 상태(214)는 상태(212)의 증분과 크거나 같은, 그러나 상태(210)로부터의 카운터 값의 급등보다 작을 수도 있는 증분으로 카운터를 조정한다. 프로세스는 디바이스 드라이버(110)(도 1)의 호출 루틴, 예컨대 파일 시스템(108)(도 1)으로 "OK" 상태를 리턴시키고, 나간다. 도 2에 예시한 프로세스는 열화되어 폐기될 블록을 식별한다. 블록에 새로운 데이터를 기입할 때와 같이 비교적 늦은 때 보다는, 블록으로부터 데이터를 판독할 때와 같이 비교적 조기에 열화 블록을 인식하는 것이 신뢰성 면에 있어서 이로울 수 있다.

[0026] 도 3은 열화된 블록의 사전 폐기를 위한 프로세서를 일반적으로 예시하는 흐름도이다. 일 실시예에서, 이 프로세스는 오퍼레이팅 시스템의 디바이스 드라이버에 의해 실행되고, 메모리 디바이스는 NAND 플래시 메모리 디바이스이다. 예시한 프로세스는 다양하게 변경될 수 있다는 것을 당업자는 알 것이다. 예컨대, 다른 실시예에서, 예시한 프로세스의 각종 부분들을 조합하거나, 엇갈린 순서로 재배열하거나, 제거하거나 할 수 있다. NAND 플래시 메모리에서는, 페이지 레벨에서 판독 또는 기입을 행하는 반면, 블록 레벨에서 소거를 행한다.

[0027] 프로세스는 메모리 디바이스로부터 블록을 소거하는 요구나 블록의 페이지에 기입하는 요구에 의해 개시된다. 예컨대, 그 요구는 오퍼레이팅 시스템(104)(도 1)의 파일 시스템(108)(도 1)으로부터 올 수 있다. 그 요구는 데이터 업데이트, 데이터 삭제, 포맷팅 등의 상황에서 일어날 수 있다.

[0028] 프로세스는 소거되는 블록이나 기입되는 페이지의 대응 블록과 연관된 에러 히스토리를 검색한다(310). 예컨대, 에러 히스토리는 도 2와 관련하여 전술한 카운트일 수 있다.

[0029] 프로세스는 블록이 신뢰성 있는 것인지를 판정한다(320). 블록이 신뢰성 있는 것으로 간주되면, 프로세스는 메모리 디바이스(106)(도 1)에게 블록을 소거(330)하도록 지시한다. 프로세스는 소거 프로세스 또는 기입 프로세스 동안에 메모리 디바이스(106) 내에서 에러가 일어났는지의 여부를 판정한다(350). 메모리 디바이스(106)가 에러없이 블록을 소거하거나 페이지에 기입하는 것으로 한다면, 프로세스는 파일 시스템(108)(도 1)과 같은 호출 루틴에 대한 "OK" 상태로 나간다. 비트 스틱(stick)과 같이 에러 또는 기입 프로세스에서 에러가 있다면, 프로세스는 에러 상태로 나갈 수 있다.

[0030] 블록이 신뢰성 없는 것으로 간주되면, 프로세스는 요구된 소거 또는 기입 동작을 거절(360)하고 I/O 에러와 같은 에러 상태로 나간다. 예컨대, 카운터가 잠재적으로 신뢰성 없는 블록의 식별을 위한 임계값 이상인 경우에 블록을 신뢰성 없는 것으로 간주할 수 있다. 에러 상태에 응답하여, 파일 시스템(108)(도 1)과 같은 상위 계층 호출 루틴은 디바이스 드라이버(110)(도 1)에게 블록을 불량으로 마킹하도록 요구하기 전에 블록의 (있다면) 이전에 기입된 페이지를 리클레임(reclaim)할 수 있다. 이전에 기입된 페이지와 현재 기입 페이지는 상이한 블록에 기입될 수 있다. 블록을 불량으로 마킹한 후에는, 그 블록을 이용할 수 없다. 일부 메모리 디바이스에서, 블록을 불량으로 마킹하는 것은 메모리 디바이스(106)(도 1)에 있는 불량 블록 테이블에 블록을 부가함으로써 달성된다. 일 실시예에서, 불량 블록 테이블에 블록을 부가하는 것은 또한 파일 시스템(108)의 요구로 디바이스 드라이버(110)(도 1)에 의해 취급된다. 블록의 조기 폐기는 블록이 불량으로 되기 전에 블록을 폐기시킬 수 있다는 이점이 있다. 이것은 미래의 데이터 손실을 피할 수 있도록 해준다.

[0031] 도 4는 정정할 수 없는 에러로 판정된 데이터를 복구하기 위한 프로세스를 일반적으로 예시하는 흐름도이다. 일 실시예에서, 이 프로세스는 오퍼레이팅 시스템(104)(도 1)의 디바이스 드라이버(110)(도 1)에 의해 실행되고, 메모리 디바이스(106)(도 1)는 NAND 플래시 메모리 디바이스이다. 예시한 프로세스는 다양하게 변경될 수 있다는 것을 당업자는 알 것이다. 예컨대, 다른 실시예에서, 예시한 프로세스의 각종 부분들을 조합하거나, 엇갈린 순서로 재배열하거나, 제거하거나 할 수 있다.

[0032] 일 실시예에서, 프로세스는 메모리 디바이스로부터 판독된 데이터의 에러를 정정할 수 없게 된 때에 시작한다. 에러 정정 코드(ECC)가 정정할 에러가 너무 많은 경우에 에러를 정정할 수 없게 된다. 예컨대, 프로세스는 도 2의 조기 폐기 표시 상태(210)에 이어서 시작할 수 있다.

[0033] 상태(410)에서, 프로세스는 정정 가능한 데이터 판독을 시도하는 판독 동작 모드를 변경한다. 대부분의 컴퓨터

시스템은 정상 동작(고속 판독 모드) 동안에 가장 적합한 속도로 메모리 디바이스(106)(도 1)와 같은 메모리 디바이스에 액세스하도록 구성된다. 일 실시예에서, 그 판독 동작 모드는 신뢰성 있는 판독 모드로 변경된다. 이것은 예컨대 버스 속도를 늦추는 것, 레이턴시를 늘리는 것 등을 포함할 수 있다. 다른 예에서, 판독 모드는 직접 메모리 액세스(DMA) 모드로부터 액세스 제어에 마이크로프로세서를 이용하는 모드로 변경될 수 있는데, 이것은 프로세서 입력/출력, 즉 PIO로도 알려져 있다. 통상적으로, DMA 모드는 마이크로프로세서가 자유롭게 다른 태스크들을 실행할 수 있게 하기 때문에 바람직하다. 그러나, PIO의 경우, 마이크로프로세서는 DMA보다 더 많은 제어로 판독 동작을 실행할 수 있다. 예컨대, 프로세스는 셀 폰의 스피커와 같이 노이즈를 생성하는 디바이스의 일시적인 불능화를 요구할 수 있다. 다른 예에서, 프로세스는 메모리 디바이스(106)에 대한 타이밍을 신중히 늦출 수 있다. 당업자는 다른 기술들도 용이하게 결정할 수 있을 것이다. 이를 기술의 각종 특징도 조합 가능하다.

[0034] 프로세스는 메모리 디바이스로부터 이전에 판독된 페이지의 판독을 요구한다(420). 이러한 판독은 재판독에 해당한다. 예러 정정 코드 데이터도 검색하고 재적용하여 데이터가 현재 정정 가능한지를 판정한다. 많은 상황에서, 더 신뢰성 있는 모드를 이용하여 판독 후 데이터를 정정할 수 있다. 데이터가 정정 가능하다면, 프로세스는 정정된 데이터를 제공하고(440), "OK" 상태로 나간다.

[0035] 데이터가 정정 불가능하다면, 프로세스는 재시도 여부를 판정하거나(450), 예러가 있는 데이터를 제공한다(460). 재시도 판정(450) 또는 데이터 제공은 예컨대 루프 카운터, 타이머 등에 기초하여 행해질 수 있다. 프로세스는 데이터를 재판독하는 상태(410) 또는 판독 상태(420)로 리턴할 수 있다. 프로세스가 상태(410)로 리턴하면, 상이한(예컨대, 제3, 제4 등) 판독 모드도 시도할 수 있다. 예컨대, 하나 이상의 신뢰성 있는 판독 모드가 존재하면, 각종 판독 모드의 특징들을 개별적으로 및/또는 조합하여 시도할 수 있다.

[0036] 데이터가 아직 정정 불가능하다면, 일 실시예는 그럼에도 불구하고 예러가 있는 데이터를 제공한다(460). 크게 실패하는 경향이 있는 하드 디스크 디바이스와 달리, NAND 플래시와 같은 고체 상태 디바이스에서의 예러는 대개 단일 비트 실패로 특징지어진다. 페이지가 정정 불가능한 경우 예러수는 통상은 1 비트보다 많지만, 예러가 있는 데이터를 리턴하는 것이 데이터를 리턴하지 않는 것보다 나을 수 있다. 예컨대, 많은 경우에, 예러가 있는 데이터의 부분은 중요하지 않을 것이다. 예컨대, 프로그램 코드가 페이지에 저장되는 경우, 예러가 있는 부분은 반드시 실행되는 것이 아닌 코드의 부분에 해당할 수 있다. 대조적으로, 데이터를 제공하지 않는 경우, 전체 페이지의 데이터를 이용할 수 없고, 그 경우 프로그램은 거의 항상 기능을 멈춘다.

[0037] 일 실시예는 비휘발성 집적 회로 메모리 디바이스의 블록을 모니터링하는 방법으로서, 상기 방법은 예러 정정 코드들을 디코딩하여 상기 메모리 디바이스의 블록의 데이터의 페이지가 적어도 하나의 예러를 가지고 판독되었는지의 여부를 판정하는 단계; 및 적어도 상기 페이지가 적어도 하나의 예러를 가지고 판독된 경우에 판독된 페이지에 대응하여 상기 메모리 디바이스의 블록에 대한 예러 히스토리를 유지하는 단계 - 상기 예러 히스토리는 상기 블록의 초기 폐기 여부를 표시하는 데에 이용됨 - 를 포함한다.

[0038] 일 실시예는 비휘발성 집적 회로 메모리 디바이스의 블록들을 관리하는 방법으로서, 상기 방법은 메모리 블록을 소거하거나 상기 메모리 블록의 페이지에 기입하는 커맨드를 수신하는 단계; 및 상기 블록의 하나 이상의 페이지에 대한 판독 동작들과 연관된 예러 히스토리에 적어도 부분적으로 기초하여 소거 또는 기입을 불능화하는 단계를 포함한다.

[0039] 일 실시예는 비휘발성 집적 회로 메모리 디바이스로부터 데이터의 페이지에 액세스하는 방법으로서, 상기 방법은 상기 메모리 디바이스로부터 데이터의 페이지의 판독을 요구하는 단계; 상기 페이지와 연관된 예러 정정 코드들(ECC)을 검사하는 단계; 및 상기 데이터의 페이지가 어떤 정정 불가능한 예러들을 포함하고 있는지의 여부와 상관없이 상기 데이터의 페이지가 정확하게 판독된 것을 표시하는 상태를 오퍼레이팅 시스템에 제공하는 단계를 포함한다.

[0040] 일 실시예는 비휘발성 집적 회로 메모리 디바이스의 블록을 표시하기 위한 하나 이상의 표시를 유지하는 방법으로서, 상기 방법은 메모리 블록의 적어도 하나의 페이지에 대하여 실행되는 적어도 하나의 판독 동작의 일부로서 검출되는 예러가 예러 정정 코드의 디코딩에 의해 정정될 수 없다는 판정, 또는 상기 메모리 디바이스의 블록들에 대해 수집된 예러 히스토리가 상기 블록에 대한 이전 판독 동작 동안에 적어도 하나의 예러가 발생한 것을 표시한다는 판정 중 적어도 하나의 판정에 적어도 부분적으로 응답하여 상기 블록의 추가의 소거 또는 재프로그래밍을 방지하는 단계를 포함한다.

[0041] 일 실시예는 비휘발성 집적 회로 메모리 디바이스의 블록을 모니터링하는 장치로서, 상기 장치는 예러 정정 코

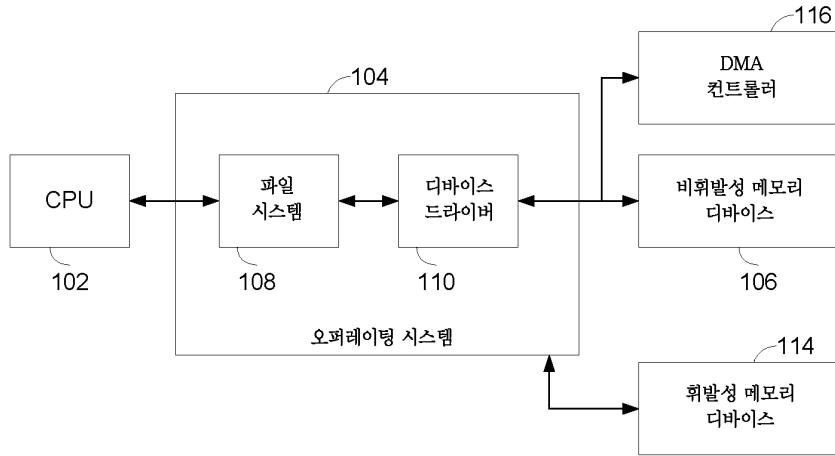
드들을 디코딩하여 상기 메모리 디바이스의 블록의 데이터의 페이지가 적어도 하나의 에러를 가지고 판독되었는지의 여부를 판정하도록 구성된 모듈; 및 적어도 상기 페이지가 적어도 하나의 에러를 가지고 판독된 경우에 판독된 페이지에 대응하여 상기 메모리 디바이스의 블록에 대한 에러 히스토리를 유지하도록 구성된 모듈 - 상기 에러 히스토리는 상기 블록의 조기 폐기 여부를 표시하는 데에 이용됨 - 을 포함한다.

[0042]

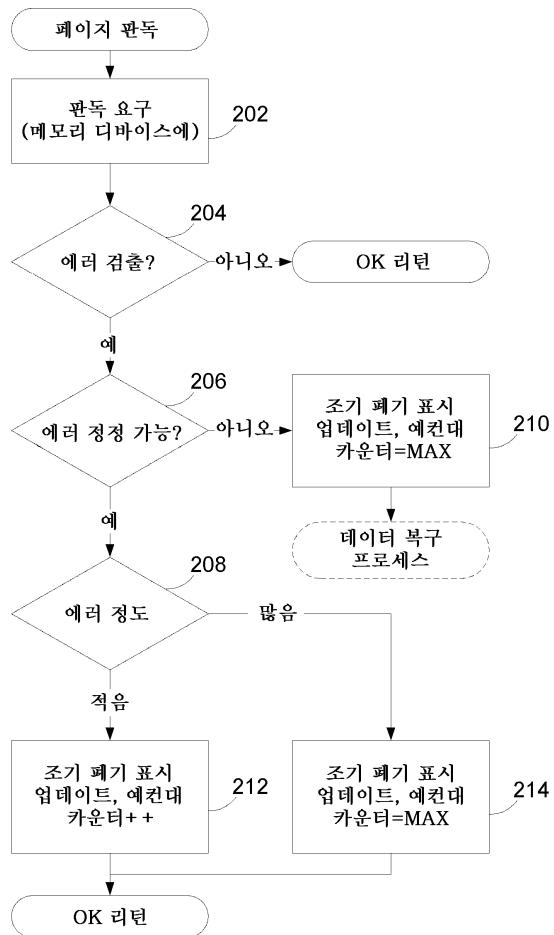
이상 각종 실시예들을 기재하였다. 이들 특정 실시예들에 관하여 기재하였으나, 본 기재는 예시적인 것이지 제한적인 것이 아니다. 당업자라면 첨부한 청구 범위에 규정된 본 발명의 기술적 사상 내에서 각종 변형 및 응용이 가능할 것이다.

## 도면

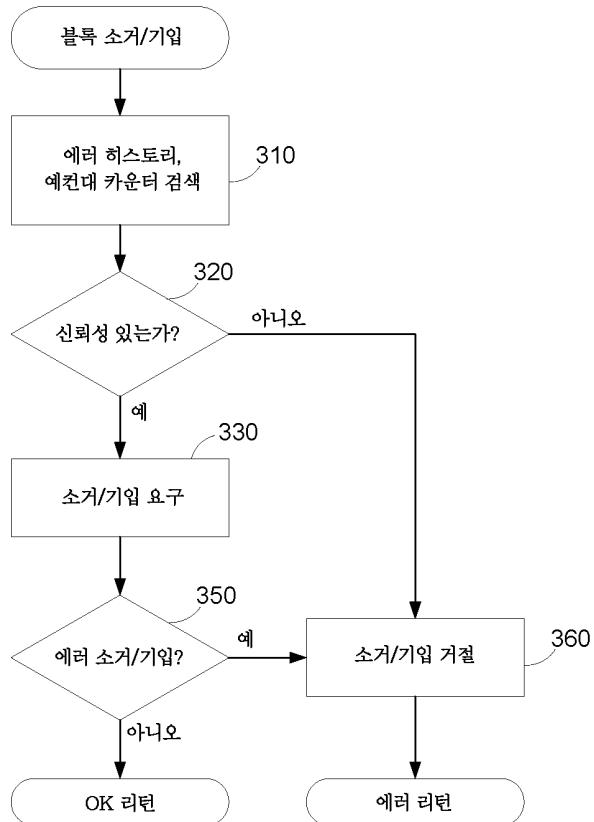
### 도면1



## 도면2



## 도면3



## 도면4

