



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2013년04월12일  
(11) 등록번호 10-1254136  
(24) 등록일자 2013년04월08일

(51) 국제특허분류(Int. Cl.)  
G06F 12/14 (2006.01) H04L 9/28 (2006.01)  
(21) 출원번호 10-2007-7016699  
(22) 출원일자(국제) 2005년12월21일  
심사청구일자 2010년12월21일  
(85) 번역문제출일자 2007년07월20일  
(65) 공개번호 10-2007-0087676  
(43) 공개일자 2007년08월28일  
(86) 국제출원번호 PCT/US2005/046688  
(87) 국제공개번호 WO 2006/069273  
국제공개일자 2006년06월29일  
(30) 우선권주장  
11/313,428 2005년12월20일 미국(US)  
(뒷면에 계속)

(56) 선행기술조사문헌  
US05438575 A  
US06158004 A

전체 청구항 수 : 총 12 항

(73) 특허권자  
샌디스크 코퍼레이션  
미국 캘리포니아 95035 밀피타스 맥카시 블러바드 601  
(72) 발명자  
홀츠만 마이클  
미국 캘리포니아 95014 쿠퍼티노 7602 반하트 플레이스  
코헨 바룩 비  
이스라엘 21721 카미엘 아야 15  
(뒷면에 계속)  
(74) 대리인  
김용인, 박영복

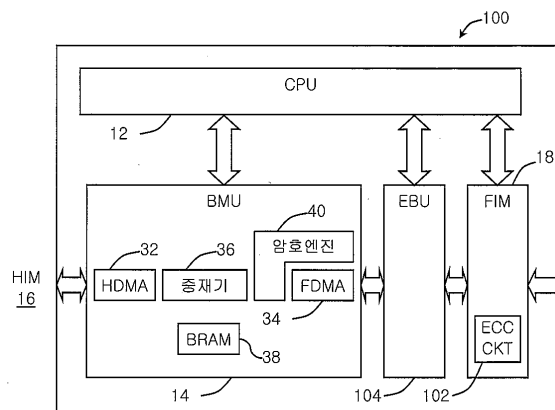
심사관 : 권오성

(54) 발명의 명칭 스트림 내 데이터 암호화/복호화 및 오류 정정 기능을 가진 메모리 시스템

(57) 요약

메모리 시스템의 처리량은 데이터 스트림 내의 데이터에 대한 오류 정정 기능이 어떤 제어기도 최소한으로 관여한 채 암호화 방법으로 처리되면 향상된다. 메모리 셀로부터 데이터가 읽혀질 때 오류를 정정하기 위해 셀들과 암호화 회로 사이를 지나는 데이터 스트림 내의 데이터에 있는 비트 오류들은 그 회로에 의해 수행되는 임의의 암호화 과정 이전에 수정된다. 복수의 버퍼를 사용하여 대기 시간을 줄일 수 있는 곳인 암호화 회로와 메모리 사이에서 데이터를 버퍼링하기 위해 채용되는 하나 또는 그 이상의 버퍼들에서 오류 정정이 일어나는 것이 바람직하다.

대표도 - 도2



(72) 발명자

**이슬람 무하메드 알**

미국 캘리포니아 95054 산타클라라 1335 1652 호프  
드라이브

**데이비슨 매튜**

미국 캘리포니아 94041 마운틴 뷰 아파트먼트 345  
1600 빌라스트리트

(30) 우선권주장

11/313,447 2005년12월20일 미국(US)

60/638,485 2004년12월21일 미국(US)

## 특허청구의 범위

### 청구항 1

비휘발성 메모리;

비휘발성 메모리로부터 판독된 데이터 안의 하나 이상의 오류(들)의 존재를 검출하도록 동작하며 데이터 내의 하나 이상의 오류(들)의 존재를 나타내는 신호를 발생하도록 또한 동작하는 회로;

데이터 상의 암호화 처리들을 수행하도록 동작하는 암호화 회로;

데이터가 암호화 회로로 보내지기 전에 비휘발성 메모리로부터 판독된 데이터를 저장하도록 동작하는 적어도 하나의 버퍼; 및

데이터 내의 하나 이상의 오류(들)의 존재를 나타내는 신호를 수신하고, 신호를 수신하는 것에 응답하여, 데이터가 적어도 하나의 버퍼로부터 암호화 회로로 보내지기 전에 적어도 하나의 버퍼에 저장된 데이터 내의 하나 이상의 오류(들)를 정정하도록 동작하는 프로세서를 포함하며,

비휘발성 메모리로부터 암호화 회로의 데이터 스트림의 한 부분은 처음에 오류 검출되고 이후 암호화 처리되며, 데이터 스트림의 한 부분이 오류 검출되고 있는 동안, 데이터 스트림의 다른 부분은 암호화 처리되는 메모리 시스템.

### 청구항 2

삭제

### 청구항 3

삭제

### 청구항 4

제 1 항에 있어서,

적어도 하나의 버퍼는 2 개의 버퍼를 포함하고, 프로세서는 비휘발성 메모리로부터 암호화 회로로 데이터를 저장하고 보내기 위해 2 개의 버퍼들을 교대로 사용하도록 또한 동작하는 메모리 시스템.

### 청구항 5

제 4 항에 있어서,

2 개의 버퍼 중 제 2 버퍼에 저장된 데이터가 암호화 회로에 보내지는 경우 프로세서는 2 개의 버퍼 중 제 1 버퍼 안의 비휘발성 메모리로부터 판독된 데이터를 저장하도록 또한 동작하는 메모리 시스템.

### 청구항 6

제 1 항에 있어서,

프로세서는 메모리 시스템이 바이패스 모드로 동작하고 있는 경우 적어도 하나의 버퍼를 바이패스하도록 또한 동작하는 메모리 시스템.

### 청구항 7

비휘발성 메모리, 데이터 내의 하나 이상의 오류(들)의 존재를 검출하도록 동작하는 회로, 암호화 회로, 적어도 하나의 버퍼, 및 프로세서를 포함하는 메모리 시스템에서 다음의 단계들을 수행하는 단계:

적어도 하나의 버퍼에서 암호화 회로와 비휘발성 메모리 사이를 통과하는 데이터를 저장하는 단계;

암호화 회로에 데이터를 제공하기 전에 적어도 하나의 버퍼에 저장된 데이터 안의 하나 이상의 오류(들)를 정정하는 단계; 및

적어도 하나의 버퍼에 저장된 데이터 안의 하나 이상의 오류(들)이 정정되고 난 후, 암호화 회로에 데이터

를 제공하는 단계를 포함하고,

데이터 안의 하나 이상의 오류(들)는 하나 이상이 오류(들)의 존재를 나타내는 회로로부터 신호를 수신하는 것에 응답하여 프로세서에 의해 정정되며,

비휘발성 메모리로부터 암호화 회로의 데이터 스트림의 한 부분은 처음에 오류 검출되고 이후 암호화 처리되며, 데이터 스트림의 한 부분이 오류 검출되고 있는 동안, 데이터 스트림의 다른 부분은 암호화 처리되는 메모리 시스템 내의 데이터 정정 방법.

#### 청구항 8

삭제

#### 청구항 9

삭제

#### 청구항 10

제 7 항에 있어서,

적어도 하나의 버퍼는 2 개의 버퍼를 포함하고 상기 방법은 비휘발성 메모리로부터 암호화 회로로 데이터를 저장하고 보내기 위해 2 개의 버퍼를 교대로 사용하는 단계를 더 포함하는 메모리 시스템 내의 데이터 정정 방법.

#### 청구항 11

제 10 항에 있어서,

2 개의 버퍼 중 제 2 버퍼에 저장된 데이터가 암호화 회로로 보내지는 경우 비휘발성 메모리로부터 판독된 데이터는 2 개의 버퍼 중 제 1 버퍼에 저장되는 메모리 시스템 내의 데이터 정정 방법.

#### 청구항 12

제 7 항에 있어서,

메모리 시스템이 바이패스 모드로 동작하고 있는 경우 적어도 하나의 버퍼를 바이패스하는 단계를 더 포함하는 메모리 시스템 내의 데이터 정정 방법.

#### 청구항 13

삭제

#### 청구항 14

제 1 항에 있어서,

비휘발성 메모리로부터 암호화 회로로 데이터 스트림은 하나 이상의 오류(들)의 존재가 검출되는 경우에만 인터럽트되는 메모리 시스템.

#### 청구항 15

제 4 항에 있어서,

비휘발성 메모리로부터 암호화 회로로 데이터 스트림은 2 개의 버퍼 중 제 1 버퍼에 저장된 데이터 안의 하나 이상의 오류(들)를 정정하는 것이 비휘발성 메모리로부터 판독된 데이터를 가지고 2 개의 버퍼 중 제 2 버퍼를 채우는 것보다 더 오래 걸리는 경우에만 인터럽트되는 메모리 시스템.

#### 청구항 16

삭제

#### 청구항 17

제 7 항에 있어서,

비휘발성 메모리로부터 암호화 회로로의 데이터 스트림은 오류(들)의 존재가 검출되는 경우에만 인터럽트되는 메모리 시스템 내의 데이터 정정 방법.

## 청구항 18

제 10 항에 있어서,

비휘발성 메모리로부터 암호화 회로로의 데이터 스트림은 2 개의 버퍼 중 제 1 버퍼에 저장된 데이터 안의 하나 이상의 오류(들)를 정정하는 것이 비휘발성 메모리로부터 판독된 데이터를 가지고 2 개의 버퍼 중 제 2 버퍼를 채우는 것보다 더 오래 걸리는 경우에만 인터럽트되는 메모리 시스템 내의 데이터 정정 방법.

## 명세서

### 기술분야

[0001] 본 발명은 일반적으로 메모리 시스템에 관한 것이며, 특히 스트림 내 데이터 암호화/복호화 및 오류 정정 기능을 가진 메모리 시스템에 관한 것이다.

### 배경기술

[0002] 모바일 기기 시장은 더욱 많은 데이터 교환을 발생함으로써 평균 수익을 증가하기 위해 콘텐츠 저장 공간을 포함하는 방향으로 발전되고 있다. 이는 콘텐츠가 모바일 기기에 저장될 때 보호되어야 한다는 것을 의미한다.

[0003] 휴대 저장 기기들은 수년 동안 상업적으로 이용되고 있다. 그것들은 하나의 컴퓨팅 기기에서 다른 컴퓨팅 기기로 데이터를 나르거나 백업 데이터를 저장하기 위해 데이터를 나른다. 휴대 하드 디스크 드라이브, 휴대 플래시 메모리 디스크, 그리고 플래시 메모리 카드 등과 같은 더욱 정교한 휴대 저장 기기들은 저장 관리를 제어하기 위해 마이크로프로세서를 포함한다.

[0004] 휴대 저장 기기에 저장된 콘텐츠를 보호하기 위해, 저장된 데이터는 일반적으로 암호화되고 승인된 사용자만이 그 데이터를 복호화하도록 허용된다.

[0005] 휴대 저장 기기에 저장된 데이터에 비트 오류가 있을 수 있으므로, 오류 정정을 이용하는 것이 바람직하다. 오류 정정을 위한 현재의 설계는 암호화 기능을 가진 휴대 저장 기기에 적합하지 않을 수 있다. 따라서 그러한 어려움을 완화할 수 있는 개선된 국부 저장 장치를 제공하는 것이 바람직하다.

### 발명의 상세한 설명

[0006] 메모리 셀들에 저장된 데이터에는 여러 가지 이유로 인해 오류가 포함될 수 있다. 따라서 메모리 셀들로부터 데이터를 읽을 때 오류 정정을 수행하는 것이 일반적이다. 오류 정정은 또한 데이터 스트림 내의 오류의 위치를 검출할 수 있다. 회로에 의해 수행되는 암호화 처리는 데이터 스트림의 비트 위치를 옮길 수가 있어서 만일 그러한 처리가 수행될 때 데이터 스트림 내의 비트 오류들이 정정되지 않는다면 그 처리 후에는 비트 오류의 위치 정보가 더 이상 정확하지 않게 되어 암호화 처리가 수행된 뒤에는 오류 정정이 더 이상 가능하지 않게 될 것이다. 따라서 발명의 일 측면은 셀들과 암호화 회로 사이를 지나는 데이터 스트림 내의 데이터에 존재하는 비트 오류들은 회로에 의해 임의의 암호화 과정이 수행되기 전에 정정되는 것이 바람직하다는 인식에 기반을 두고 있다.

[0007] 적어도 하나의 버퍼가 상기 셀들 및 상기 회로 사이를 지나는 데이터 스트림 내의 데이터를 저장하기 위하여 사용되고, 상기 버퍼에 저장되고 상기 셀들로부터 유래된 데이터 내의 오류 또는 오류들은 회로에 의해 데이터가 암호화 처리되기 전에 정정되는 것이 바람직하다.

### 실시예

[0014] 본 발명의 다양한 측면이 구현되는 메모리 시스템의 예가 도 1의 블록도에 의해 예시되어 있다. 도 1에 나타난 바와 같이, 메모리 시스템(10)은 중앙처리장치(CPU, 12), 버퍼 관리유닛(BMU, 14), 호스트 인터페이스 모듈(HIM, 16), 그리고 플래시 인터페이스 모듈(FIM, 18), 플래시 메모리(20) 및 주변장치 접속 모듈(PAM, 22)을 포함한다. 메모리 시스템(10)은 호스트 인터페이스 버스(26)와 포트 (26a)를 통해 호스트 기기(24)와 통신한다.

NAND형일 수 있는 플래시 메모리(20)는 호스트 기기(24)를 위한 데이터 저장 공간을 제공한다. CPU(12)를 위한 소프트웨어 코드는 또한 플래시 메모리(20)에 저장된다. FIM(18)은 플래시 인터페이스 버스 (28)와 포트 (28a)를 통하여 플래시 메모리(20)에 연결된다. HIM(16)은 디지털 카메라, 퍼스널 컴퓨터, 휴대용 개인 정보 단말기(PDA), 디지털 미디어 재생기, 애플-3 재생기 및 휴대폰이나 기타 다른 디지털 기기와 같은 호스트 시스템에 접속되기에 적합하다. 주변장치 접속 모듈(22)은 CPU(12)와 통신하기 위해 FIM, HIM 그리고 BMU와 같은 적합한 제어기 모듈을 선정한다. 일 실시예에서, 점선 상자 안에 있는 시스템(10)의 모든 요소는 메모리 카드 또는 스틱(10')과 같은 단일 장치 내에 담길 수 있고 바람직하게는 카드나 스틱에 포함될 수 있다.

[0015] 버퍼 관리유닛(14)은 호스트 직접 메모리 접근(HDMA, 32), 플래시 직접 메모리 접근(FDMA) 제어기(34), 중재기(36), 버퍼 임의 접근 메모리(BRAM, 38) 및 암호 엔진(40)을 포함한다. 중재기(36)는 공유 버스 중재기로 단 하나의 주장치나 시작장치(HDMA(32)나 FDMA(34) 또는 CPU(12) 등이 될 수 있다.)가 임의의 시간에 활성화되고 중속 장치나 목표 장치는 BRAM(38)이 되도록 한다. 중재기는 적절한 시작장치 요구가 BRAM(38)에 채널을 형성하는 책임을 맡는다. HDMA(32)와 FDMA(34)는 HIM(16), FIM(18) 그리고 BRAM(38) 또는 CPU 임의 접근 메모리(CPU RAM, 12a) 사이에서 데이터가 전달되도록 하는 책임을 맡는다. HDMA(32)와 FDMA(34)의 동작은 중재의 것과 동일하고 여기서 상술할 필요가 없다. BRAM(38)은 호스트 기기(24), 플래시 메모리(20) 및 CPU RAM(12a) 사이에서 통과되는 데이터를 버퍼링 하기 위해 사용된다. HDMA(32)와 FDMA(34)는 HIM(16)/FIM(18) 및 BRAM(38) 또는 CPU RAM(12a) 사이에서 데이터를 전달하고 섹터 전달의 완료를 알려 주는 책임을 맡는다. 아래에 기술되는 바와 같이 FIM(18)은 플래시 메모리(20)에서 읽혀지는 데이터 내의 오류를 검출하고 오류가 발견되면 CPU(12)에 통보하는 능력도 가지고 있다.

[0016] 최초에 플래시 메모리(20)로부터의 데이터가 호스트 기기(24)에 의해 읽혀질 때, 메모리(20) 내의 암호화된 데이터는 버스(28), FIM(18), FDMA(34), 그리고 암호화된 데이터가 복호화 되는 암호 엔진(40)을 통해 가져와지고 BRAM(38)에 저장된다. 이 후 복호화된 데이터는 BRAM(38)으로부터 HDMA(32), HIM(16), 버스(26)를 통해 호스트 기기(24)로 보내진다. BRAM(38)으로부터 가져온 데이터는 HDMA(32)에 통과되기 전에 암호 엔진에 의해 다시 암호화될 수 있는데 호스트 기기(24)로 보내진 데이터는 메모리(20)에 저장된 데이터를 복호화할 때 사용된 것과 비교하여 다른 키 값 및/또는 알고리즘으로 다시 암호화된다. 대안적 일 실시예에서, 데이터가 인가받지 않은 접근에 취약하게 될 수 있는, 전술한 처리에서 BRAM(38)내에 복호화된 데이터를 저장하는 대신, 메모리(20)로부터의 데이터가 BRAM(38)에 보내지기 전에 암호 엔진(40)에 의해 복호화되고 다시 암호화되는 것이 바람직하다. 그 후 BRAM(38) 내의 암호화된 데이터는 이전처럼 호스트 기기(24)로 보내진다. 이것은 읽기 프로세스 동안의 데이터 스트림을 예시한다.

[0017] 호스트 기기(24)에 의해 데이터가 메모리(20)에 쓰여질 때, 데이터 스트림의 방향은 반대가 된다. 예를 들어, 만약 암호화되지 않은 데이터가 호스트에 의해 버스(26), HIM(16), HDMA(32)를 통해 암호 엔진(40)으로 보내지면, 그러한 데이터는 BRAM(38)에 저장되기 전에 엔진(40)에 의해 암호화될 수 있다. 대안으로, 암호화되지 않은 데이터는 BRAM(38)에 저장될 수 있다. 그 데이터는 이 후 FDMA(34)에 보내지기 전에 메모리(20)로 전달되면서 암호화된다. 쓰여진 데이터가 다단계 암호화 처리를 거치는 경우, 엔진(40)은 처리된 데이터가 BRAM(38)에 저장되기 전에 그러한 처리를 완료하는 것이 바람직하다.

[0018] 도 1의 메모리 시스템(10)이 플래시 메모리를 포함하는 반면에, 시스템은 대신 자기 디스크, 광학 시디 등과 같은 또 다른 형태의 비휘발성 메모리뿐만 아니라, 모든 다른 형태의 재기록 가능한 비휘발성 메모리 시스템을 대안으로 포함할 수 있으며 전술된 다양한 장점들이 그러한 대안적 실시예에도 동일하게 적용될 것이다. 대안적 실시예에서는, 메모리는 또한 메모리 시스템의 나머지 구성요소들과 함께 (메모리 카드나 스틱과 같은) 동일한 물리적인 공간(body) 내에 포함되는 것이 바람직하다.

# [0019] 오류 정정

[0020] 비휘발성(예를 들어, 플래시) 메모리에 저장된 데이터는 손상되어 오류를 포함할 수 있다. 이런 이유로, FIM(18)은 비트 스트림 내의 오류 위치를 포함하는, 메모리(20)로부터 오는 데이터 스트림의 어떤 비트(들)가 오류를 포함하고 있는지를 검출하는 오류 정정(ECC) 회로(102)를 포함할 수 있다. 이는 본 발명의 또 다른 측면을 예시하기 위해 메모리 시스템(100)의 블록도인 도 2에 예시되어 있다. FIM(18)은 비트 스트림에 오류가 검출되었을 때 CPU(12)에 인터럽트 신호를 보내고, 회로(102)는 오류 내의 비트의 위치와 관련된 정보를 CPU(12)에 보낸다. 암호화 기능이 없는 기존의 메모리 시스템에서는, 오류가 CPU(12)에 의해 BRAM(38)에서 정정된다. 그러나, 만일 데이터 스트림으로부터의 데이터가 먼저 정정되기 전에 암호화 처리되면, 암호화 과정은 처리된 데이

터 스트림 내의 데이터 비트들의 위치 및/또는 값을 변하게 만들 수 있어, 암호화 처리 후의 비트 오류의 위치 및/또는 값이 회로(102)에 의해 CPU(12)로 보내지는 비트 오류의 위치 및/또는 값과는 달라지게 된다. 이것은 암호화 처리된 데이터가 BRAM(38)에 도달했을 때 오류를 정정하는 것을 불가능하게 만든다. 발명의 일 측면은 이런 문제들이 피해지도록 데이터가 암호화 처리되기 전에 검출된 오류가 정정되는 인식으로부터 비롯된다.

[0021] 오류 버퍼유닛(EBU, 104)은 BMU(14)와 FIM(18) 사이를 통과하는 데이터 스트림으로부터의 데이터를 저장하는데 사용되어, CPU(12)가 데이터 스트림내의 오류의 존재를 알리는 인터럽트를 FIM(18)로부터 받았을 때, CPU가 BRAM(38) 대신 EBU(104)에서 오류를 정정할 수 있다. 디지털 데이터를 정정하기 위해서는, 오류 내의 비트들은 회로(102)에 의해 검출된 오류(들) 위치에서 간단하게 "변환(flip)"(즉, 1은 0으로 그리고 0은 1로)된다.

[0022] 오류가 검출되었을 때 데이터 스트림 내의 인터럽트 빈도를 줄이기 위해, 도 3에 나타난 바와 같이, 두 개 또는 그 이상의 버퍼들이 EBU(104)내에 채택될 수 있다. 도 3에 나타난 바와 같이, 두 개의 버퍼들(104a, 104b)이 사용되는데, 두 버퍼 중 하나는 FIM(18)을 통해 메모리(20)로부터 데이터를 수신하고 다른 하나는 BMU(14)에 있는 FDMA(34)를 통해 암호 엔진(40)으로 데이터를 보낸다. 도 3에, 두 개의 스위치(106a, 106b)가 사용된다. 도 3에 나타난 바와 같이, 두 개의 스위치가 실선 위치에 있을 때, 버퍼 104a는 BMU(14)에 데이터를 제공하며 버퍼 104b는 FIM(18)으로부터 데이터를 수신한다. 도 3에 나타난 바와 같이, 두 개의 스위치가 점선 위치에 있을 때, 버퍼 104b가 BMU(14)에 데이터를 제공하고 버퍼 104a가 FIM(18)으로부터 데이터를 수신한다. 각각의 버퍼는 저장된 데이터가 BMU에 보내지기 전에 먼저 데이터로 채워질 수 있다. CPU는 데이터가 송수신 될 때 버퍼 104a와 104b에 있는 오류를 정정한다. 이러한 방식으로, 유일한 대기시간은 데이터 스트림이 시작될 때 두 개의 버퍼 중의 하나를 채우기 위해 필요한 시간이다. 그 이후에는, 오류가 회로(102)에 의해 검출될 때에도, 만일 CPU에 의해 오류를 정정하기 위해 소요되는 시간이 각각의 버퍼를 채우기 위해 필요한 시간에 비해 작으면, 데이터 스트림은 인터럽트되지 않는다.

[0023] 데이터를 정정하는 것이 버퍼를 채우는 것보다 오래 걸린다면, 데이터 스트림은 오류가 검출되었을 때에만 인터럽트될 것이고 오류가 검출되지 않았을 때에는 데이터 스트림은 인터럽트 없이 진행될 것이다. EBU(104)와 FDMA(34) 사이를 연결하는 버퍼가 비었다는 신호(도시되지 않음)는 데이터 스트림이 인터럽트되고 더 이상의 데이터가 쓸모없다는 것을 FDMA(34)에 알려준다. 그 다음 암호 엔진(40)과 마찬가지로 FDMA(34)는 멈추어 데이터 스트림이 다시 시작하기를 기다린다.

[0024] 데이터가 호스트 기기(24)에 의해 메모리(20)에 기록될 때, 오류 정정이 필요 없어, EBU를 바이패스(bypass)하는 것이 바람직하다. 이것은 스위치(108)에 의해 이루어진다. 스위치(108)가 닫히면, HIM(16)으로부터의 데이터(도 2에 완전히 도시되지 않음)는 두 개의 버퍼(104a 및 104b)를 간단히 바이패스한다. 스위치(108)는 데이터가 메모리(20)로부터 읽히거나 기록될 때 암호화 절차가 필요 없는 바이패스 모드에서도 닫히게 된다. 이 모드에서는, HDMA와 FDMA가 암호 엔진(40)이 시스템(10)에서 제거된 것처럼 중재기(36)에 직접 연결되고, 데이터 스트림은 EBU(104)와 암호 엔진(40)을 모두 바이패스한다. 이것 또한 스위치들을 사용하여 이루어진다. 따라서, 바이패스 모드에서는, CPU(12)의 제어를 받는 시스템(100)에 있는 논리 회로(도시되지 않음)가 데이터 스트림이 블록(40)을 바이패스하고 스위치(108)가 닫히도록 한다.

[0025] 오류 정정 절차가 도 4의 흐름도에 예시되어 있다. CPU(12)는 호스트 기기(24)로부터 읽기 명령을 받은 후에 읽기 동작을 시작한다(타원 150). 그리고 적절한 보안 설정 정보를 사용하여 암호 엔진(40)을 설정하고, BMU(14)를 읽기 동작으로 설정하며, 동작을 위해 BRAM(38)에 있는 메모리 공간의 할당과 같은 다른 변수를 설정한다(블록 152와 154). 또한 FIM(18)을 설정하는데, 예를 들면 메모리(20)에서 데이터가 읽히는 곳의 위치를 지정함으로써 설정한다(블록 156). HDMA 엔진(32)과 FDMA 엔진(34)은 그리고 나서 시작된다. 블록 158을 참조한다. CPU가 인터럽트를 받으면, FIM 인터럽트인지를 확인한다(다이아몬드 160). FIM 인터럽트가 수신되면, CPU는 그 인터럽트가 데이터 스트림에 하나 또는 그 이상의 오류가 있다는 것을 나타내는 것인지를 확인한다(블록 162). 만일 오류(들)가 나타나면, 버퍼 104a 및/또는 104b에서 오류(들) 정정(블록 164)을 진행하고 다음에 데이터가 읽히는 곳인 메모리(20) 내의 위치를 변경하기 위해 FIM(18)을 설정하기 위해 되돌아간다(블록 156). FIM 인터럽트가 데이터 스트림 내의 오류(들)를 나타내지 않는 때에는, FIM이 동작을 완료하고 CPU도 FIM을 재설정하고 재구동하기 위해 블록 156으로 되돌아간다는 것을 의미한다. 만일 CPU에 의해 검출된 인터럽트가 FIM 인터럽트가 아니라면, 데이터 인터럽트의 종료인지를 확인한다(다이아몬드 166). 만일 그렇다면, 읽기 동작은 종료된다(타원 168). 그렇지 않다면, 이 인터럽트는 데이터의 암호화 절차와 무관(예를 들면 클록 인터럽트)하고 CPU(12)는 그것을 서비스(도시되지 않음)하고 인터럽트를 확인하기 위해 다이아몬드 160으로 되돌아간다.

[0026] 도 4는 쓰기 동작을 위해서 약간만 수정되지만 하면 된다. 메모리(20)에 쓰여지는 데이터 내의 ECC 오류의 처리



가 없으므로, 쓰기 동작에서 CPU(12)는 다이아몬드 162와 블록 164의 절차들을 생략할 수 있다. 만일 FIM 인터럽트가 쓰기 동작 중에 CPU(12)에 의해 받아지면, 이것은 FIM이 동작을 완료하고 CPU도 FIM을 재설정하기 위해 블록 156으로 되돌아 갔다는 것을 의미한다. 이러한 차이를 제외하고는, 쓰기 동작은 실질적으로 읽기 동작과 비슷하다.

[0027] 본 발명이 다양한 실시예들에 대해 언급함으로써 상기 기술되는 동안, 오직 첨부된 청구항 및 이와 균등한 것들에 의해 정의됨으로써, 발명의 범위를 벗어나지 않고 변화와 수정을 할 수 있다는 것이 이해될 수 있을 것이다. 여기서 언급된 모든 참고문헌들은 참조로서 포함된다.

### 산업상 이용 가능성

[0028] 본 명세서 내에 포함되어 있음.

### 도면의 간단한 설명

[0008] 도 1은 본 발명을 예시하기 위한 호스트 장치와 통신하는 메모리 시스템의 블록도이다.

[0009] 도 2는 도 1의 메모리 시스템 블록 일부의 블록도이다.

[0010] 도 3은 도 2의 오류 정정 버퍼유닛의 바람직한 구성을 더욱 상세히 예시하는 회로도이다.

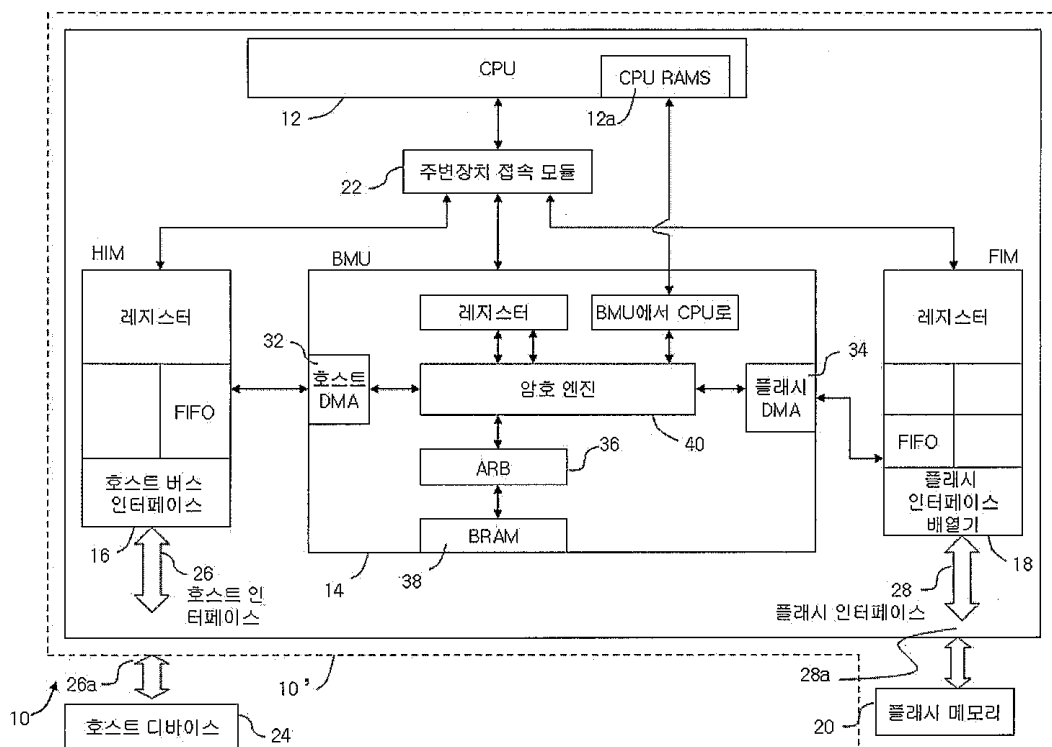
[0011] 도 4는 본 발명의 일 측면에 대한 바람직한 실시예를 예시하기 위해 도 2에 있는 시스템의 동작을 예시하는 흐름도이다.

[0012] 본 출원에서는 설명의 편의상 동일한 구성요소들은 동일한 번호로 표시된다.

[0013] 본 발명에 따른 실시예들은 첨부된 도면을 참조하여 이하에서 상세히 설명한다.

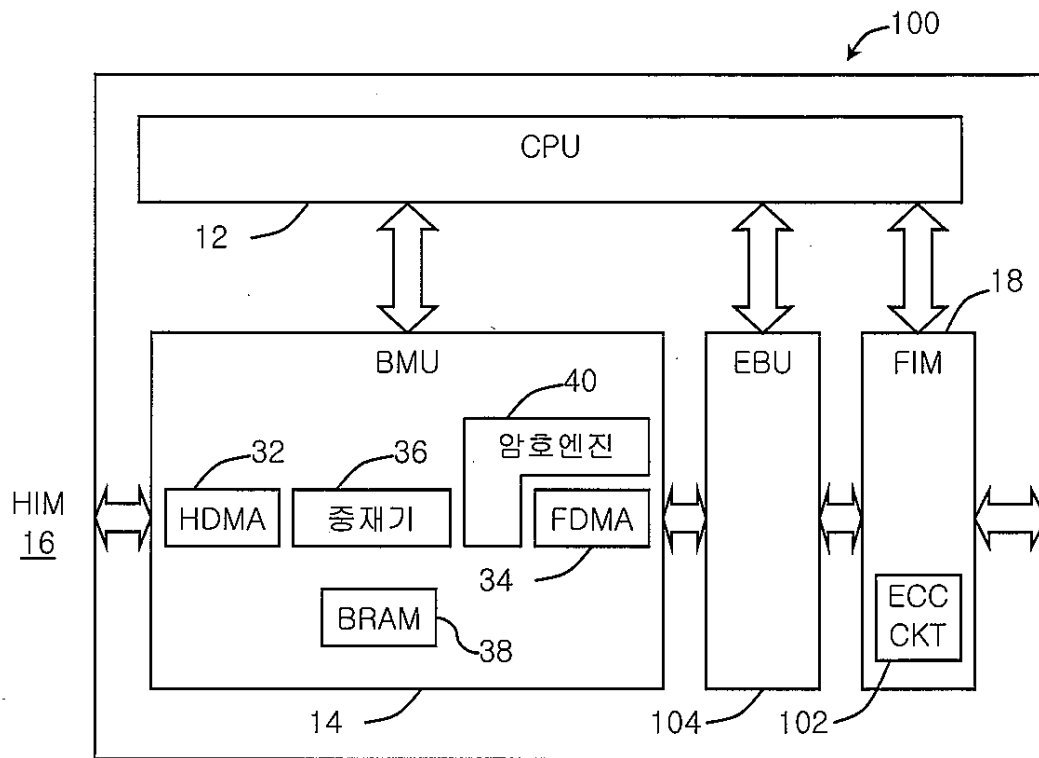
### 도면

#### 도면1

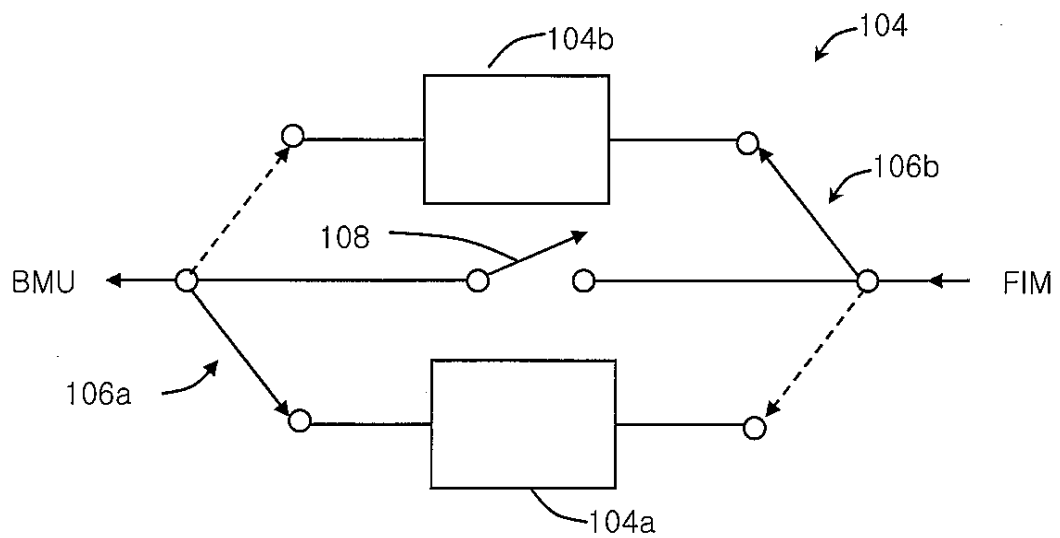




도면2



도면3



도면4

