



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0096782
(43) 공개일자 2015년08월25일

(51) 국제특허분류(Int. Cl.)
G06F 3/06 (2006.01)
(52) CPC특허분류(Coo. Cl.)
G06F 3/064 (2013.01)
G06F 12/0246 (2013.01)
(21) 출원번호 10-2015-7019521
(22) 출원일자(국제) 2013년11월07일
심사청구일자 없음
(85) 번역문제출일자 2015년07월17일
(86) 국제출원번호 PCT/US2013/068939
(87) 국제공개번호 WO 2014/099169
국제공개일자 2014년06월26일
(30) 우선권주장
13/720,532 2012년12월19일 미국(US)

(71) 출원인
켈컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
알리 다뎀 파케르
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
바부 베니쉬
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
특허법인코리아나

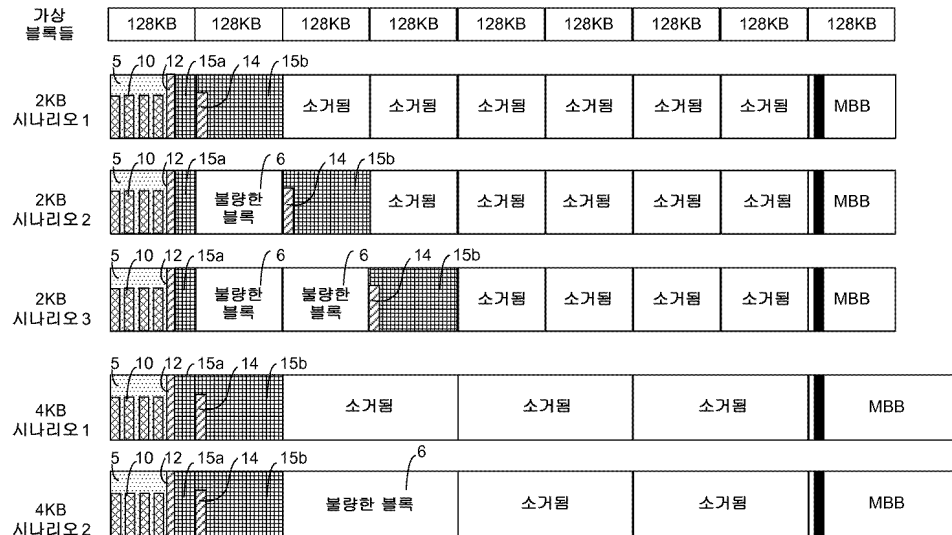
전체 청구항 수 : 총 68 항

(54) 발명의 명칭 **판독-기록 메모리 디바이스의 데이터 이미지에서 가상 경계 코드들**

(57) 요약

데이터 이미지로 판독-기록 메모리 디바이스를 구성하기 위한 방법들, 시스템들, 및 디바이스들이 제공된다. 그 방법은 판독-기록 메모리 디바이스에 대하여 지정된 가상 블록들의 시리즈의 가상 블록 사이즈에 기초하여 데이터 이미지 분배를 결정하는 것을 포함한다. 데이터 이미지는 하나 이상의 데이터 이미지 부분들로 분할되고, 여기서 가상 경계 코드가 데이터 이미지 부분들 중 적어도 하나에 부가된다. 데이터 이미지 부분들은 가상 블록들의 시리즈 중 개별 가상 블록들에 저장되고, 판독-기록 메모리 디바이스 내, 심지어 가상 블록들 간의 임의의 불량한 블록을 스킵한다.

대표도



- (52) CPC특허분류(Coo. Cl.)
G06F 3/0619 (2013.01)
G06F 3/0679 (2013.01)
G06F 3/0688 (2013.01)
G11C 29/82 (2013.01)
G06F 2212/7202 (2013.01)

- (72) 발명자

엘라라 타아라 난다키쇼레

미국 92121 캘리포니아주 샌디에고 모어하우스 드
라이브 5775

쿠마르 아쉬와니

미국 92121 캘리포니아주 샌디에고 모어하우스 드
라이브 5775

특허청구의 범위

청구항 1

관독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법으로서,

제 1 관독-기록 메모리 디바이스에 대하여 지정된 가상 블록들의 시리즈의 가상 블록 사이즈에 기초하여 데이터 이미지 분배를 결정하는 단계로서, 상기 제 1 관독-기록 메모리 디바이스는 실제 블록들의 시리즈를 포함하는, 상기 데이터 이미지 분배를 결정하는 단계;

데이터 이미지 제 1 부분에 가상 경계 코드를 추가하는 단계로서, 상기 가상 경계 코드는 상기 데이터 이미지 제 21 부분을 위치시키기 위한 마커를 나타내는, 상기 가상 경계 코드를 추가하는 단계; 및

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 1 부분을 상기 제 1 관독-기록 메모리 디바이스 상의 상기 가상 블록들의 시리즈 중의 가상 블록에 저장하는 단계를 포함하는, 관독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 2

제 1 항에 있어서,

상기 가상 경계 코드는 상기 데이터 이미지 제 1 부분의 시작부에 추가되는, 관독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 3

제 1 항에 있어서,

상기 실제 블록들의 시리즈 중 각각의 실제 블록의 사이즈는 상기 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지는, 관독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 4

제 1 항에 있어서,

데이터 이미지 제 2 부분을 상기 제 1 관독-기록 메모리 디바이스 상의 상기 가상 블록들의 시리즈 중 다른 가상 블록에 저장하는 단계를 더 포함하며,

상기 데이터 이미지는 상기 데이터 이미지 제 1 부분 및 상기 데이터 이미지 제 2 부분을 포함하는 1 초과의 부분으로 분할되는, 관독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 5

제 4 항에 있어서,

상기 제 1 관독-기록 메모리 디바이스 내의 불량한 블록은 상기 데이터 이미지 제 1 부분과 상기 데이터 이미지 제 2 부분 사이에 위치되는, 관독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 6

제 4 항에 있어서,

상기 데이터 이미지 제 1 부분 및 상기 데이터 이미지 제 2 부분은 상기 제 1 관독-기록 메모리 디바이스 상의 상기 실제 블록들의 시리즈 중 공유된 실제 블록에 저장되는, 관독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 7

제 4 항에 있어서,

상기 데이터 이미지 제 2 부분은 상기 제 1 관독-기록 메모리 디바이스의 블록 0 을 점유하는, 관독-기록 메모리

리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 8

제 4 항에 있어서,

상기 데이터 이미지 제 2 부분은 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 경계 코드의 위치와 상기 데이터 이미지의 사이즈 중 적어도 하나와 관련된 정보를 포함하는 이미지 헤더를 포함하는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 9

제 4 항에 있어서,

상기 가상 경계 코드를 데이터 이미지 제 3 부분에 추가하는 단계; 및

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 3 부분을 상기 가상 블록들의 시리즈 중 제 3 가상 블록에 저장하는 단계를 더 포함하는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 10

제 1 항에 있어서,

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 1 부분을 제 2 판독-기록 메모리 디바이스 상의 가상 블록에 저장하는 단계를 더 포함하며,

페이지, 블록 및 디바이스-특정 불량-블록 알고리즘 특징 중 적어도 하나는 상기 제 1 판독-기록 메모리 디바이스와 상기 제 2 판독-기록 메모리 디바이스 사이에서 상이한, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법.

청구항 11

판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스로서,

메모리; 및

상기 메모리에 커플링되고, 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성된 프로세서를 포함하며,

상기 동작들은,

제 1 판독-기록 메모리 디바이스에 대하여 지정된 가상 블록들의 시리즈의 가상 블록 사이즈에 기초하여 데이터 이미지 분배를 결정하는 것으로서, 상기 제 1 판독-기록 메모리 디바이스는 실제 블록들의 시리즈를 포함하는, 상기 데이터 이미지 분배를 결정하는 것;

데이터 이미지 제 1 부분에 가상 경계 코드를 추가하는 것으로서, 상기 가상 경계 코드는 상기 데이터 이미지 제 1 부분을 위치시키기 위한 마커를 나타내는, 상기 가상 경계 코드를 추가하는 것; 및

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 1 부분을 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 블록들의 시리즈 중의 가상 블록에 저장하는 것

을 포함하는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 12

제 11 항에 있어서,

상기 프로세서는, 상기 가상 경계 코드가 상기 데이터 이미지 제 1 부분의 시작부에 추가되도록 하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 13

제 11 항에 있어서,

상기 프로세서는, 상기 실제 블록들의 시리즈 중 각각의 실제 블록의 사이즈가 상기 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지도록 하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 14

제 11 항에 있어서,

상기 프로세서는,

데이터 이미지 제 2 부분을 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 블록들의 시리즈 중 다른 가상 블록에 저장하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되고,

상기 데이터 이미지는 상기 데이터 이미지 제 1 부분 및 상기 데이터 이미지 제 2 부분을 포함하는 1 초과의 부분으로 분할되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 15

제 14 항에 있어서,

상기 프로세서는, 상기 제 1 판독-기록 메모리 디바이스 내의 불량한 블록이 상기 데이터 이미지 제 1 부분과 상기 데이터 이미지 제 2 부분 사이에 위치되도록 하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 16

제 14 항에 있어서,

상기 프로세서는, 상기 데이터 이미지 제 1 부분 및 상기 데이터 이미지 제 2 부분이 상기 제 1 판독-기록 메모리 디바이스 상의 상기 실제 블록들의 시리즈 중 공유된 실제 블록에 저장되도록 하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 17

제 14 항에 있어서,

상기 프로세서는, 상기 데이터 이미지 제 2 부분이 상기 제 1 판독-기록 메모리 디바이스의 블록 0 을 점유하도록 하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 18

제 14 항에 있어서,

상기 프로세서는, 상기 데이터 이미지 제 2 부분이 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 경계 코드의 위치와 상기 데이터 이미지의 사이즈 중 적어도 하나와 관련된 정보를 포함하는 이미지 헤더를 포함하도록 하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 19

제 14 항에 있어서,

상기 프로세서는,

상기 가상 경계 코드를 데이터 이미지 제 3 부분에 추가하는 것; 및

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 3 부분을 상기 가상 블록들의 시리즈 중 제 3 가상 블록에 저장하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 20

제 11 항에 있어서,

상기 프로세서는,

상기 가상 경계 코드가 부가된 상기 데이터 이미지 제 1 부분을 제 2 판독-기록 메모리 디바이스 상의 가상 블록에 저장하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되며,

페이지, 블록 및 디바이스-특정 불량-블록 알고리즘 특징 중 적어도 하나는 상기 제 1 판독-기록 메모리 디바이스와 상기 제 2 판독-기록 메모리 디바이스 사이에서 상이한, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 21

판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스로서,

제 1 판독-기록 메모리 디바이스에 대하여 지정된 가상 블록들의 시리즈의 가상 블록 사이즈에 기초하여 데이터 이미지 분배를 결정하는 수단으로서, 상기 제 1 판독-기록 메모리 디바이스는 실제 블록들의 시리즈를 포함하는, 상기 데이터 이미지 분배를 결정하는 수단;

데이터 이미지 제 1 부분에 가상 경계 코드를 부가하는 수단으로서, 상기 가상 경계 코드는 상기 데이터 이미지 제 1 부분을 위치시키기 위한 마커를 나타내는, 상기 가상 경계 코드를 부가하는 수단; 및

상기 가상 경계 코드가 부가된 상기 데이터 이미지 제 1 부분을 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 블록들의 시리즈 중의 가상 블록에 저장하는 수단을 포함하는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 22

제 21 항에 있어서,

상기 가상 경계 코드는 상기 데이터 이미지 제 1 부분의 시작부에 부가되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 23

제 21 항에 있어서,

상기 실제 블록들의 시리즈 중 각각의 실제 블록의 사이즈는 상기 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 24

제 21 항에 있어서,

데이터 이미지 제 2 부분을 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 블록들의 시리즈 중 다른 가상 블록에 저장하는 수단을 더 포함하며,

상기 데이터 이미지는 상기 데이터 이미지 제 1 부분 및 상기 데이터 이미지 제 2 부분을 포함하는 1 초과의 부분으로 분할되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 25

제 24 항에 있어서,

상기 제 1 판독-기록 메모리 디바이스 내의 불량한 블록은 상기 데이터 이미지 제 1 부분과 상기 데이터 이미지 제 2 부분 사이에 위치되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 26

제 24 항에 있어서,

상기 데이터 이미지 제 1 부분 및 상기 데이터 이미지 제 2 부분은 상기 제 1 판독-기록 메모리 디바이스 상의 상기 실제 블록들의 시리즈 중 공유된 실제 블록에 저장되는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 27

제 24 항에 있어서,

상기 데이터 이미지 제 2 부분은 상기 제 1 판독-기록 메모리 디바이스의 블록 0 을 점유하는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 28

제 24 항에 있어서,

상기 데이터 이미지 제 2 부분은 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 경계 코드의 위치와 상기 데이터 이미지의 사이즈 중 적어도 하나와 관련된 정보를 포함하는 이미지 헤더를 포함하는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 29

제 24 항에 있어서,

상기 가상 경계 코드를 데이터 이미지 제 3 부분에 추가하는 수단; 및

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 3 부분을 상기 가상 블록들의 시리즈 중 제 3 가상 블록에 저장하는 수단을 더 포함하는, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 30

제 21 항에 있어서,

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 1 부분을 제 2 판독-기록 메모리 디바이스 상의 가상 블록에 저장하는 수단을 더 포함하며,

페이지, 블록 및 디바이스-특정 불량-블록 알고리즘 특징 중 적어도 하나는 상기 제 1 판독-기록 메모리 디바이스와 상기 제 2 판독-기록 메모리 디바이스 사이에서 상이한, 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 컴퓨팅 디바이스.

청구항 31

저장된 프로세서 실행가능 소프트웨어 명령들을 갖는 비-일시적인 컴퓨터 판독가능 저장 매체로서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금 메모리를 갖는 컴퓨팅 디바이스에서 메모리를 관리하기 위한 동작들을 수행하게 하도록 구성되고,

상기 동작들은,

제 1 판독-기록 메모리 디바이스에 대하여 지정된 가상 블록들의 시리즈의 가상 블록 사이즈에 기초하여 데이터 이미지 분배를 결정하는 것으로서, 상기 제 1 판독-기록 메모리 디바이스는 실제 블록들의 시리즈를 포함하는, 상기 데이터 이미지 분배를 결정하는 것;

데이터 이미지 제 1 부분에 가상 경계 코드를 추가하는 것으로서, 상기 가상 경계 코드는 상기 데이터 이미지 제 1 부분을 위치시키기 위한 마커를 나타내는, 상기 가상 경계 코드를 추가하는 것; 및

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 1 부분을 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 블록들의 시리즈 중의 가상 블록에 저장하는 것

을 포함하는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 32

제 31 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금, 상기 가상 경계 코드가 상기 데이터 이미지 제 1 부분의 시작부에 부가되도록 하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 33

제 31 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금, 상기 실제 블록들의 시리즈 중 각각의 실제 블록의 사이즈가 상기 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지도록 하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 34

제 31 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

데이터 이미지 제 2 부분을 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 블록들의 시리즈 중 다른 가상 블록에 저장하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되고,

상기 데이터 이미지는 상기 데이터 이미지 제 1 부분 및 상기 데이터 이미지 제 2 부분을 포함하는 1 초과의 부분으로 분할되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 35

제 34 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금, 상기 제 1 판독-기록 메모리 디바이스 내의 불량한 블록이 상기 데이터 이미지 제 1 부분과 상기 데이터 이미지 제 2 부분 사이에 위치되도록 하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 36

제 34 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금, 상기 데이터 이미지 제 1 부분 및 상기 데이터 이미지 제 2 부분이 상기 제 1 판독-기록 메모리 디바이스 상의 상기 실제 블록들의 시리즈 중 공유된 실제 블록에 저장되도록 하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 37

제 34 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금, 상기 데이터 이미지 제 2 부분이 상기 제 1 판독-기록 메모리 디바이스의 블록 0 을 점유하도록 하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 38

제 34 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금, 상기 데이터 이미지 제 2 부분이 상기 제 1 판독-기록 메모리 디바이스 상의 상기 가상 경계 코드의 위치와 상기 데이터 이미지의 사이즈 중 적어도 하나와 관련된 정보를 포함하는 이미지 헤더를 포함하도록 하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 39

제 34 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

상기 가상 경계 코드를 데이터 이미지 제 3 부분에 추가하는 것; 및

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 3 부분을 상기 가상 블록들의 시리즈 중 제 3 가상 블록에 저장하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 40

제 31 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

상기 가상 경계 코드가 추가된 상기 데이터 이미지 제 1 부분을 제 2 판독-기록 메모리 디바이스 상의 가상 블록에 저장하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되며,

페이지, 블록 및 디바이스-특정 불량-블록 알고리즘 특징 중 적어도 하나는 상기 제 1 판독-기록 메모리 디바이스와 상기 제 2 판독-기록 메모리 디바이스 사이에서 상이한, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 41

판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 방법으로서,

제 1 판독-기록 메모리 디바이스의 제 1 가상 블록 내의 제 1 데이터에 액세스하는 단계로서, 상기 제 1 판독-기록 메모리 디바이스의 실제 블록 사이즈는 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지는, 상기 제 1 데이터에 액세스하는 단계;

상기 제 1 가상 블록 다음의 제 2 가상 블록의 제 2 가상 블록 경계를 지정하는 가상 경계 코드에 대하여 상기 제 1 판독-기록 메모리 디바이스를 스캔하는 단계;

상기 제 2 가상 블록에서 상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 2 가상 블록 내의 제 2 데이터에 액세스하는 단계; 및

상기 제 1 데이터 및 상기 제 2 데이터에 기초하여 제 1 디바이스 데이터 이미지를 판독하는 단계를 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 방법.

청구항 42

제 41 항에 있어서,

상기 제 1 데이터 및 상기 제 2 데이터가 상기 데이터 이미지의 완전한 이미지를 포함하지 않다는 결정에 응답하여, 상기 제 2 가상 블록 다음의 제 3 가상 블록의 제 3 가상 블록 경계를 지정하는 가상 경계 코드에 대하여 상기 제 1 판독-기록 메모리 디바이스를 스캔하는 단계;

상기 제 3 가상 블록에서 상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 3 가상 블록 내의 제 3 데이터에 액세스하는 단계; 및

상기 제 3 데이터에 추가로 기초하여 상기 데이터 이미지를 판독하는 단계를 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 방법.

청구항 43

제 41 항에 있어서,

상기 제 1 데이터 및 상기 제 2 데이터에 기초하여 부트 로더를 실행하는 단계를 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 방법.

청구항 44

제 41 항에 있어서,

상기 가상 경계 코드가 실제 블록에서 검출되지 않는 것에 응답하여 상기 제 1 가상 블록 다음에 상기 제 1 판독-기록 메모리 디바이스의 상기 실제 블록을 스킵하는 단계를 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 방법.

청구항 45

제 41 항에 있어서,

상기 가상 경계 코드가 중간 거리에서 검출되지 않는 것에 응답하여 상기 제 1 가상 블록의 사이즈에 대응하는 상기 제 1 판독-기록 메모리 디바이스 상의 상기 중간 거리를 스킵하는 단계를 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 방법.

청구항 46

제 45 항에 있어서,

상기 제 1 가상 블록 내의 상기 제 1 데이터에 포함된 헤더 정보를 스캔하는 단계; 및

상기 헤더 정보에 기초하여 상기 데이터 이미지와 상기 가상 블록 사이즈 중 적어도 하나의 사이즈를 결정하는 단계를 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 방법.

청구항 47

제 41 항에 있어서,

제 2 판독-기록 메모리 디바이스의 제 3 가상 블록 내의 제 3 데이터에 액세스하는 단계로서, 페이지, 블록 및 디바이스-특정 불량-블록 알고리즘 특징 중 적어도 하나는 상기 제 1 판독-기록 메모리 디바이스와 상기 제 2 판독-기록 메모리 디바이스 사이에서 상이한, 상기 제 3 데이터에 액세스하는 단계;

상기 가상 경계 코드에 대하여 상기 제 2 판독-기록 메모리 디바이스를 스캔하는 단계;

상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 2 판독-기록 메모리 디바이스의 제 4 가상 블록 내의 제 4 데이터에 액세스하는 단계; 및

상기 제 3 데이터 및 상기 제 4 데이터에 기초하여 제 2 디바이스 데이터 이미지를 판독하는 단계를 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 방법.

청구항 48

컴퓨팅 디바이스로서,

메모리; 및

상기 메모리에 커플링되고, 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성된 프로세서를 포함하며,

상기 동작들은,

제 1 판독-기록 메모리 디바이스의 제 1 가상 블록 내의 제 1 데이터에 액세스하는 것으로서, 상기 제 1 판독-기록 메모리 디바이스의 실제 블록 사이즈는 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지는, 상기 제 1 데이터에 액세스하는 것;

상기 제 1 가상 블록 다음의 제 2 가상 블록의 제 2 가상 블록 경계를 지정하는 가상 경계 코드에 대하여 상기 제 1 판독-기록 메모리 디바이스를 스캔하는 것;

상기 제 2 가상 블록에서 상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 2 가상 블록 내의 제 2 데이터에 액세스하는 것; 및

상기 제 1 데이터 및 상기 제 2 데이터에 기초하여 제 1 디바이스 데이터 이미지를 판독하는 것을 포함하는, 컴퓨팅 디바이스.

청구항 49

제 48 항에 있어서,

상기 프로세서는,

상기 제 1 데이터 및 상기 제 2 데이터가 데이터 이미지의 완전한 이미지를 포함하지 않는다는 결정에 응답하여, 상기 제 2 가상 블록 다음의 제 3 가상 블록의 제 3 가상 블록 경계를 지정하는 가상 경계 코드에 대하여 상기 제 1 판독-기록 메모리 디바이스를 스캔하는 것;

상기 제 3 가상 블록에서 상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 3 가상 블록 내의 제 3 데이터에 액세스하는 것; 및

상기 제 3 데이터에 추가로 기초하여 상기 데이터 이미지를 판독하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 50

제 48 항에 있어서,

상기 프로세서는,

상기 제 1 데이터 및 상기 제 2 데이터에 기초하여 부트 로더를 실행하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 51

제 48 항에 있어서,

상기 프로세서는,

상기 가상 경계 코드가 실제 블록에서 검출되지 않는 것에 응답하여 상기 제 1 가상 블록 다음에 상기 제 1 판독-기록 메모리 디바이스의 상기 실제 블록을 스킵하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 52

제 48 항에 있어서,

상기 프로세서는,

상기 가상 경계 코드가 중간 거리에서 검출되지 않는 것에 응답하여 상기 제 1 가상 블록의 사이즈에 대응하는 상기 제 1 판독-기록 메모리 디바이스 상의 상기 중간 거리를 스킵하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 53

제 52 항에 있어서,

상기 프로세서는,

상기 제 1 가상 블록 내의 상기 제 1 데이터에 포함된 헤더 정보를 스캔하는 것; 및

상기 헤더 정보에 기초하여 상기 데이터 이미지와 상기 가상 블록 사이즈 중 적어도 하나의 사이즈를 결정하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 54

제 48 항에 있어서,

상기 프로세서는,

제 2 판독-기록 메모리 디바이스의 제 3 가상 블록 내의 제 3 데이터에 액세스하는 것으로서, 페이지, 블록 및 디바이스-특정 불량-블록 알고리즘 특징 중 적어도 하나는 상기 제 1 판독-기록 메모리 디바이스와 상기 제 2 판독-기록 메모리 디바이스 사이에서 상이한, 상기 제 3 데이터에 액세스하는 것;

상기 가상 경계 코드에 대하여 상기 제 2 판독-기록 메모리 디바이스를 스캔하는 것;

상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 2 판독-기록 메모리 디바이스의 제 4 가상 블록 내의 제 4 데이터에 액세스하는 것; 및

상기 제 3 데이터 및 상기 제 4 데이터에 기초하여 제 2 디바이스 데이터 이미지를 판독하는 것

을 더 포함하는 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 55

판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 컴퓨팅 디바이스로서,

제 1 판독-기록 메모리 디바이스의 제 1 가상 블록 내의 제 1 데이터에 액세스하는 수단으로서, 상기 제 1 판독-기록 메모리 디바이스의 실제 블록 사이즈는 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지는, 상기 제 1 데이터에 액세스하는 수단;

상기 제 1 가상 블록 다음의 제 2 가상 블록의 제 2 가상 블록 경계를 지정하는 가상 경계 코드에 대하여 상기 제 1 판독-기록 메모리 디바이스를 스캔하는 수단;

상기 제 2 가상 블록에서 상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 2 가상 블록 내의 제 2 데이터에 액세스하는 수단; 및

상기 제 1 데이터 및 상기 제 2 데이터에 기초하여 제 1 디바이스 데이터 이미지를 판독하는 수단을 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 컴퓨팅 디바이스.

청구항 56

제 55 항에 있어서,

상기 제 1 데이터 및 상기 제 2 데이터가 상기 데이터 이미지의 완전한 이미지를 포함하지 않다는 결정에 응답하여, 상기 제 2 가상 블록 다음의 제 3 가상 블록의 제 3 가상 블록 경계를 지정하는 가상 경계 코드에 대하여 상기 제 1 판독-기록 메모리 디바이스를 스캔하는 수단;

상기 제 3 가상 블록에서 상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 3 가상 블록 내의 제 3 데이터에 액세스하는 수단; 및

상기 제 3 데이터에 추가로 기초하여 상기 데이터 이미지를 판독하는 수단을 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 컴퓨팅 디바이스.

청구항 57

제 55 항에 있어서,

상기 제 1 데이터 및 상기 제 2 데이터에 기초하여 부트 로더를 실행하는 수단을 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 컴퓨팅 디바이스.

청구항 58

제 55 항에 있어서,

상기 가상 경계 코드가 실제 블록에서 검출되지 않는 것에 응답하여 상기 제 1 가상 블록 다음에 상기 제 1 판독-기록 메모리 디바이스의 상기 실제 블록을 스킵하는 수단을 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 컴퓨팅 디바이스.

청구항 59

제 55 항에 있어서,

상기 가상 경계 코드가 중간 거리에서 검출되지 않는 것에 응답하여 상기 제 1 가상 블록의 사이즈에 대응하는 상기 제 1 판독-기록 메모리 디바이스 상의 상기 중간 거리를 스킵하는 수단을 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 컴퓨팅 디바이스.

청구항 60

제 59 항에 있어서,

상기 제 1 가상 블록 내의 상기 제 1 데이터에 포함된 헤더 정보를 스캔하는 수단; 및

상기 헤더 정보에 기초하여 상기 데이터 이미지와 상기 가상 블록 사이즈 중 적어도 하나의 사이즈를 결정하는 수단을 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 컴퓨팅 디바이스.

청구항 61

제 55 항에 있어서,

제 2 판독-기록 메모리 디바이스의 제 3 가상 블록 내의 제 3 데이터에 액세스하는 수단으로서, 페이지, 블록 및 디바이스-특정 불량-블록 알고리즘 특징 중 적어도 하나는 상기 제 1 판독-기록 메모리 디바이스와 상기 제 2 판독-기록 메모리 디바이스 사이에서 상이한, 상기 제 3 데이터에 액세스하는 수단;

상기 가상 경계 코드에 대하여 상기 제 2 판독-기록 메모리 디바이스를 스캔하는 수단;

상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 2 판독-기록 메모리 디바이스의 제 4 가상 블록 내의 제 4 데이터에 액세스하는 수단; 및

상기 제 3 데이터 및 상기 제 4 데이터에 기초하여 제 2 디바이스 데이터 이미지를 판독하는 수단을 더 포함하는, 판독-기록 메모리 디바이스로부터 데이터 이미지를 판독하는 컴퓨팅 디바이스.

청구항 62

저장된 프로세서 실행가능 소프트웨어 명령들을 갖는 비-일시적인 컴퓨터 판독가능 저장 매체로서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금 메모리를 갖는 컴퓨팅 디바이스에서 메모리를 판독하기 위한 동작들을 수행하게 하도록 구성되고,

상기 동작들은,

제 1 판독-기록 메모리 디바이스의 제 1 가상 블록 내의 제 1 데이터에 액세스하는 것으로서, 상기 제 1 판독-기록 메모리 디바이스의 실제 블록 사이즈는 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지는, 상기 제 1 데이터에 액세스하는 것;

상기 제 1 가상 블록 다음의 제 2 가상 블록의 제 2 가상 블록 경계를 지정하는 가상 경계 코드에 대하여 상기 제 1 판독-기록 메모리 디바이스를 스캔하는 것;

상기 제 2 가상 블록에서 상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 2 가상 블록 내의 제 2 데이터에 액세스하는 것; 및

상기 제 1 데이터 및 상기 제 2 데이터에 기초하여 제 1 디바이스 데이터 이미지를 판독하는 것

을 포함하는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 63

제 62 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

상기 제 1 데이터 및 상기 제 2 데이터가 데이터 이미지의 완전한 이미지를 포함하지 않는다는 결정에 응답하여, 상기 제 2 가상 블록 다음의 제 3 가상 블록의 제 3 가상 블록 경계를 지정하는 가상 경계 코드에 대하여 상기 제 1 판독-기록 메모리 디바이스를 스캔하는 것;

상기 제 3 가상 블록에서 상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 3 가상 블록 내의 제 3 데이터에 액세스하는 것; 및

상기 제 3 데이터에 추가로 기초하여 상기 데이터 이미지를 판독하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 64

제 62 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

상기 제 1 데이터 및 상기 제 2 데이터에 기초하여 부트 로더를 실행하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 65

제 62 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

상기 가상 경계 코드가 실제 블록에서 검출되지 않는 것에 응답하여 상기 제 1 가상 블록 다음에 상기 제 1 판독-기록 메모리 디바이스의 상기 실제 블록을 스킵하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 66

제 62 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

상기 가상 경계 코드가 중간 거리에서 검출되지 않는 것에 응답하여 상기 제 1 가상 블록의 사이즈에 대응하는 상기 제 1 판독-기록 메모리 디바이스 상의 상기 중간 거리를 스킵하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 67

제 66 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

상기 제 1 가상 블록 내의 상기 제 1 데이터에 포함된 헤더 정보를 스캔하는 것; 및

상기 헤더 정보에 기초하여 상기 데이터 이미지와 상기 가상 블록 사이즈 중 적어도 하나의 사이즈를 결정하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

청구항 68

제 62 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금,

제 2 판독-기록 메모리 디바이스의 제 3 가상 블록 내의 제 3 데이터에 액세스하는 것으로서, 페이지, 블록 및 디바이스-특정 불량-블록 알고리즘 특징 중 적어도 하나는 상기 제 1 판독-기록 메모리 디바이스와 상기 제 2 판독-기록 메모리 디바이스 사이에서 상이한, 상기 제 3 데이터에 액세스하는 것;

상기 가상 경계 코드에 대하여 상기 제 2 판독-기록 메모리 디바이스를 스캔하는 것;

상기 가상 경계 코드를 인식하는 것에 응답하여 상기 제 2 판독-기록 메모리 디바이스의 제 4 가상 블록 내의 제 4 데이터에 액세스하는 것; 및

상기 제 3 데이터 및 상기 제 4 데이터에 기초하여 제 2 디바이스 데이터 이미지를 판독하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비-일시적인 컴퓨터 판독가능 저장 매체.

명세서

배경 기술

- [0001] 플래시 메모리는 전기적으로 소거되고 재프로그래밍될 수 있는 비-휘발성 컴퓨터 저장 칩이다. (바로 NAND 메모리로서도 지칭되는) NAND 플래시 메모리는 블록들 또는 페이지들에서 프로그래밍되고 판독될 수도 있는 고밀도 타입의 판독-기록 메모리이다. NAND 메모리는 메모리 카드들, USB 플래시 드라이브들, 솔리드-스테이트 드라이브들, 및 일반적인 데이터의 저장 및 전송을 위한 유사 제품들에서 사용될 뿐만 아니라, 다수의 디지털 디바이스들에서 구성 데이터를 저장하기 위해 사용된다.
- [0002] 판독 전용 메모리 (ROM) 는 컴퓨터들 및 다른 전자 디바이스들에서 사용되는 저장 매체의 클래스이다. 종래의 ROM 은 판독-전용 메모리이고 수정될 수 없거나, 오직 천천히 또는 어렵게 수정될 수 있고, 따라서 주로 펌웨어를 배포하는데 사용된다. 펌웨어는 특정 하드웨어를 위해 특수 설계된 소프트웨어를 지칭한다. 디바이스의 펌웨어를 변경시키는 것은 그 경제 수명 동안 거의 또는 결코 실행되지 않을 수도 있다. 펌웨어를 업데이트하는 일반적인 이유들은, 버그들을 고정시키거나 그 펌웨어가 사용되는 하드웨어에 특징들을 추가하는 것을 포함한다. 펌웨어를 업데이트하는 한 가지 방식은 ROM 과 함께 사용된 플래시 메모리를 특정 절차로 재프로그래밍하는 것이다. ROM 과 달리, NAND 메모리는 다수 회 소거되고 재프로그래밍될 수 있다.
- [0003] NAND 메모리는 통상적으로 다수의 블록들로 조직되고, 각각의 블록은 다수의 페이지들로 구성된다. NAND 메모리의 실제 페이지 또는 블록의 길이가 제조업자에 의해 세팅될 경우, 메모리는 그 길이에서 영구적으로 고정된다. 컴퓨터 메모리, 및 특히 NAND 메모리에 관련되는 것과 같은 "블록" 은 공칭 길이를 갖는, 스토리지 바이트들 또는 비트들의 시퀀스를 포함한다. 그 길이는 블록 사이즈로 지칭된다. 데이터를 블록들내로 저장하는 프로세스는 일반적으로 한 번에 전체 페이지에서 달성되지만, 데이터를 소거하는 것은 블록들의 단위들로 실행된다.
- [0004] 각각의 블록은 또한, 통상적으로 2,048 바이트 (~2KB) 또는 4,096 바이트 (~4KB) 사이즈인 다수의 페이지들로 조직될 수도 있다. 각 블록의 적어도 하나의 페이지와 연관되는 것은, 에러 정정 코드 (ECC) 검사합의 저장을 위해 사용될 수도 있는 약간의 바이트들 (통상적으로 데이터 사이즈의 1/32 임) 이다. 통상적인 블록 사이즈들은 다음을 포함한다:
- [0005] ● 각각 2,048 바이트 (~2KB) 를 가지는 64 개 페이지들 및 64 바이트의 스페어 영역, 이는 (페이지 데이터를 위한) 128KB 의 블록 사이즈 및 (에러 정정 코드들, 불량한 블록 정보, 메타 데이터 등과 같은 스페어 데이터를 위한) 4KB 의 블록 사이즈를 의미함;
 - [0006] ● 각각 4,096 바이트 (~4KB) 를 가지는 64 개 페이지들 및 128 바이트의 스페어 영역, 이는 페이지 데이터를 위한 256KB 의 블록 사이즈를 의미함;
 - [0007] ● 각각 2,048 바이트 (~2KB) 를 가지는 128 개 페이지들 및 128 바이트의 스페어 영역, 이는 페이지 데이터를 위한 256KB 의 블록 사이즈를 의미함.
- [0008] NAND 메모리들의 제작시, 제조 결함들이 종종 발생하며, 일부 메모리 셀들을 비-기능적이고 비-사용적으로 렌더링할 수 있다. 결함이 있는 메모리 셀 또는 셀들을 포함하는 이들 블록들은 "불량한 블록들" 로 지칭된다. 제조 프로세스로 인해, 다수의 NDNA 디바이스들은 일부 불량한 블록들과 함께 공장으로부터 출하된다. 예를 들어, 일부 불량한 블록들을 허용함으로써, 제조업자들은 모든 블록들이 우수하게 검증되어야할 경우에 가능한 것보다 훨씬 높은 수율들을 달성한다. 이는 NAND 플래시 비용들을 상당히 감소시키고 부품들의 저장 용량을 오직 약간 감소시킨다. 그러나, NAND 메모리의 제조업자들은 일반적으로, 최초 블록 (블록 0) 이 우수한 것을 보장하지만, 나머지 데이터 블록들은 보장되지 않는다.
- [0009] 일부 불량한 블록 관리 방식들은 칩 상의 불량한 블록들을 식별하는 정보를 NAND 메모리상에 배치한다. 그러한 불량한 블록 정보는 액세스될 이미지를 저장하는 우수한 블록들 또는 스킵될 불량한 블록들을 표시하는 리스트를 포함할 수도 있다. 그러나, NAND 메모리 자체의 특징들에서의 변수들뿐만 아니라, 재판매자 및/또는 제조업자 특정 불량 블록 검출 방식들/포맷들은, 커스터마이징된 NAND 메모리 핸들링 및 프로그래밍을 너무 자주 요구한다.

[0010] 추가로, 컴퓨팅 디바이스에서 ROM 펌웨어 설계는 소프트웨어와 비교할 때 다소 제한되며, 이는 이들의 루틴들이 개발 사이클에서 실질적으로 차후에 재기록될 수 없고, 일반적으로 컴퓨팅 디바이스가 제조되면 영구적으로 세팅되기 때문이다. 결과적으로, 플래시 메모리는, 플래시 메모리 상의 데이터 이미지의 전부 또는 부분이 재기록되고, 따라서 업데이트될 수 있기 때문에, 부트 로더 루틴들을 실행하는데 사용된다. 본 명세서에서 사용된 것과 같이, 용어 "데이터 이미지"는 NAND 메모리 디바이스와 같은 관독-기록 메모리 디바이스의 하나 이상의 메모리 셀들에 저장된 정보를 지칭한다. 현대의 부트 로더들에 대한 플래시 메모리에서 유출된 데이터 이미지들은 복잡한 경향이 있는데, 이는 그들이 전술된 것과 같이 재판매자들과 제조업자들 사이에서 상당히 변화하는 NAND 디바이스-특정 특징들을 수용하려고 시도하기 때문이다.

발명의 내용

과제의 해결 수단

[0011] 다양한 실시형태들은, 플래시 메모리와 같은 관독-기록 메모리에 데이터 이미지를 저장하는 것을 포함하며, 관독-기록 메모리는 불량한 블록들을 수용하고, 컴퓨팅 디바이스들이, 메모리 제조업자와 관계없이 단일 알고리즘을 사용하여 메모리로부터 관독할 때, 불량한 블록들을 스킵할 수 있게 한다. 그 방법은 데이터의 우수한 블록들의 경계들에서 코드들 또는 특정 값들로 관독-기록 메모리에 저장되도록 데이터 이미지를 구성하는 것을 포함한다. 그러한 코드들 또는 특정 값들은 본 명세서에서 "가상 경계 코드들"로 지칭되며, 컴퓨팅 디바이스에 의해 인식될 경우, 식별된 블록이 유효 데이터를 포함하는 것을 나타낸다. 데이터 이미지는 메모리에서, 가상 블록의 시작 또는 종료를 표시하는 가상 블록 경계에 대응하는 가상 블록 사이즈를 갖는 하나 이상의 가상 블록들에 저장될 수도 있다. 각각의 가상 블록의 사이즈는 NAND 디바이스에 대한 실제 블록 사이즈보다 작거나 동일할 수도 있다. 가상 블록 사이즈는 제품에 의해 지원되는 것으로 예상되는 최저 공통 실제 블록 사이즈로서 세팅될 수도 있다. 데이터 이미지를 NAND 메모리에 저장하기 위한 시스템은, 불량한 블록들을 식별하기 위해 메모리 제조업자 정의 알고리즘을 사용하고, 그 후에 데이터 이미지를 우수한 블록들에 저장하고 불량한 블록들은 스킵한다. 가상 경계 코드들은 데이터 이미지를 저장하는데 사용된 각각의 우수한 블록의 가상 블록 경계들에 포함된다. 따라서, 데이터 이미지는, 컴퓨팅 디바이스가 메모리 칩에서 우수 또는 불량한 블록들을 식별하기 위해 재판매자-특정 파라미터들을 뒤따르지 않고 관독-기록 메모리를 액세스할 때 관독될 데이터의 블록을 식별하기 위해 찾을 수 있는, 가상 경계 코드들을 포함한다.

[0012] 다른 양태는 컴퓨팅 디바이스들에서 구현될 수도 있는 관독-기록 메모리로부터 이미지를 관독하는 방법을 포함한다. 이러한 방법은, 메모리의 제 1 가상 블록 내의 데이터에 액세스하는 것 및 가상 경계 코드에 대하여 메모리 디바이스를 스캔하는 것을 포함할 수도 있다. 가상 경계 코드는 제 1 가상 블록을 뒤따르는 제 2 가상 블록의 경계를 지정한다. 그 방법은 또한, 가상 경계 코드를 인식하는 것에 응답하여 제 2 가상 블록 내의 데이터에 액세스하는 것을 포함할 수도 있다. 추가로, 그 방법은 적어도 제 1 가상 블록에서 액세스된 데이터와 제 2 가상 블록에서 액세스된 데이터에 기초하여 데이터 이미지를 관독하는 것을 포함할 수도 있다. 그 방법은 인식되는 가상 경계 코드들에 응답하여 추가의 가상 블록들에서 데이터에 액세스하는 것을 포함할 수도 있다.

[0013] 추가의 양태들은 위에서 논의된 방법들에 대응하는 다양한 동작들을 수행하기 위한 프로세서 실행가능 명령들로 구성된 프로세서를 갖는 컴퓨팅 디바이스를 포함한다.

[0014] 추가의 양태들은 위에서 논의된 방법 동작들에 대응하는 기능들을 수행하기 위한 다양한 수단들을 갖는 컴퓨팅 디바이스를 포함한다.

[0015] 추가의 양태들은 프로세서로 하여금 위에서 논의된 방법 동작들에 대응하는 다양한 동작들을 수행하게 하도록 구성된 저장된 프로세서 실행가능 명령들을 가지는 비-일시적인 프로세서 관독가능 저장 매체를 포함한다.

도면의 간단한 설명

[0016] 본 명세서에 첨부되고 본 명세서의 일부분을 구성하는 첨부 도면들은 본 발명의 예시적인 실시형태들을 도시하고, 위에서 주어진 일반적인 설명 및 하기에서 주어지는 상세한 설명과 함께, 본 발명의 특징들을 설명하는 역할을 한다.

도 1은 일 실시형태에 따른 관독-기록 메모리 디바이스 패키징의 다이어그램이다.

도 2는 도 1의 관독-기록 메모리 디바이스를 구성하는 프로세스 단계들을 도시한다.

도 3 은 일 실시형태에 따라, 5 개의 판독-기록 메모리 디바이스 패키징 시나리오들을 도시한다.

도 4 는 일 실시형태에 따라, 2 개의 추가의 판독-기록 메모리 디바이스 패키징 시나리오들을 도시한다.

도 5 는 판독-기록 메모리에 데이터 이미지를 저장하는 일 양태 방법을 예시하는 프로세스 흐름도이다.

도 6 은 판독-기록 메모리 디바이스로부터 부트 로더를 판독하고 실행하는 일 양태 방법의 프로세스 흐름도이다.

도 7 은 판독-기록 메모리 디바이스에 이미지를 재기록하는 프로세스 단계들을 도시한다.

도 9 은 판독-기록 메모리 디바이스에 이미지를 재기록하는 일 양태 방법의 프로세스 흐름도이다.

도 9 는 다양한 실시형태들과 함께 사용하기에 적합한 컴퓨팅 디바이스의 컴포넌트 도면이다.

도 10 은 다양한 실시형태들과 함께 사용하기에 적합한 다른 컴퓨팅 디바이스의 컴포넌트 도면이다.

도 11 은 다양한 실시형태들과 함께 사용하기에 적합한 또 다른 컴퓨팅 디바이스의 컴포넌트 도면이다.

발명을 실시하기 위한 구체적인 내용

[0017]

다양한 실시형태들은 첨부 도면들을 참조하여 상세히 설명될 것이다. 가능하면, 도면들 전체에 걸쳐 동일한 또는 유사한 부분들을 참조하기 위해, 동일한 도면 부호들이 사용될 것이다. 특정 예들 및 실시형태들에 대해 이루어진 참조들은 예시용이고, 본 발명 또는 본 청구항들의 범위를 제한하려고 의도되지는 않는다. 대안적인 실시형태들은 본 개시물의 범위로부터 벗어나지 않으면서 고안될 수도 있다. 추가로, 본 개시물의 잘 알려진 엘리먼트들은 본 개시물의 관련 세부사항들을 불명료하게 하지 않도록 상세하게 설명되지 않거나 생략될 것이다.

[0018]

단어 "예시적인" 은 "예, 예증, 또는 예시로서 기능하는" 을 의미하도록 본 명세서에서 사용된다. "예시적인" 것으로서 본 명세서에서 설명된 임의의 구현은 다른 구현들에 비해 반드시 바람직하거나 유리한 것으로서 해석되는 것은 아니다. 추가로, 단어들, "제 1", "제 2", "제 3", "일차", "이차", "삼차" 또는 유사한 용어는 다양한 설명된 엘리먼트들을 구별하기 위해 명확성을 목적으로 본 명세서에서 의도된 것이며, 본 발명을 엘리먼트들의 특정 순서 또는 계층에 제한하기 위해 의도된 것은 아니다. 본 명세서에서 사용된 것과 같은 용어들 "액세스" 또는 "액세스하는" 은 전자 저장 디바이스와 상호작용하는 행동을 지칭하거나, 특정된 경우에, 데이터 또는 정보를 스캔하거나 판독하는 목적을 위해, 그 특정 부분을 지칭한다. 또한, 본 명세서에서 사용된 것과 같은 용어들 "스캔" 또는 "스캔하는" 은 특히 전자 저장 디바이스에서 데이터 또는 정보를 검사하거나, 해석하거나, 검토하는 것을 의미한다. 대조적으로, 본 명세서에서 사용된 것과 같은 용어들 "판독" 또는 "판독하는" 은 특히 전자 저장 디바이스에서 데이터 또는 정보를 획득하거나, 추출하거나, 복사하거나, 취출하는 것을 의미한다.

[0019]

여기에 개시된 다양한 양태들은 플래시 메모리와 같은 판독-기록 메모리들에 데이터 이미지를 저장하고, 그러한 메모리로부터 커스터마이징된 및/또는 디바이스-특정된 불량 블록 검출/관리 루틴들에 대한 필요성을 방지하는 데이터 이미지들을 판독하는 메커니즘들을 제공한다. 개시된 기술들의 일 양태는 플래시 메모리 디바이스와 같은 판독-기록 메모리 디바이스에 데이터 이미지를 저장하는 방법을 포함한다. 그 방법은 판독-기록 메모리 디바이스에 대하여 지정된 가상 블록들의 시리즈의 가상 블록 사이즈에 기초하여 데이터 이미지 분배를 결정하는 것을 포함한다. 가상 블록들은 본 명세서의 양태들에 따라 생성되지만, 실제 블록들은 플래시 메모리와 같은 판독-기록 메모리의 고정된 파라미터들이다. 판독-기록 메모리 디바이스의 실제 블록 사이즈는 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어진다. 이러한 방식으로, 하나 이상의 가상 블록들은 실제 블록과 정확히 동일할 것이다. 데이터 이미지가 가상 블록 사이즈보다 크다면, 그 데이터 이미지가 저장될 판독-기록 메모리의 블록 사이즈를 수용하는 부분들로 분할된다. 가상 경계 코드는 데이터 이미지 부분들 중 적어도 하나에 부가된다. 가상 경계 코드는 가상 블록의 초기 경계를 형성하는 데이터 이미지 부분의 시작부에 부가될 수도 있다. 데이터 이미지는 부트 프로세스에서 컴퓨팅 디바이스에 의해 사용된 데이터 및 소프트웨어의 부분일 수도 있다. 데이터 이미지는 가상 블록 사이즈 및 전체 데이터 이미지의 사이즈에 의존하여, 요구되는 만큼 다수의 부분들로 분할될 수도 있다. 데이터 이미지의 전부 또는 부분은 여전히 NAND 메모리의 블록 0 에 저장될 수도 있다. 그 방법은 가상 블록들의 시리즈 중 가상 블록에 데이터 이미지 부분을 저장하는 것을 더 포함한다. 데이터 이미지 부분들을 저장할 때, 임의의 개재된 불량한 블록들은 스킵된다. 이러한 방식으로, 하나 이상의 불량한 블록들은 개별 가상 블록들에 저장된, 저장된 데이터 이미지

부분들 사이에 배치될 수도 있다. 데이터 이미지 제 1 및 제 2 부분들은 판독-기록 메모리 디바이스의 공유되는 실제 블록에 배치될 수도 있다. 데이터 이미지의 후속 부분들은 각각 그에 부가된 가상 경계 코드를 가질 수도 있다. 이는 데이터 이미지를, 우수/불량한 블록 리스트에 의존하지 않고 판독될 메모리의 가상 블록들을 로케이팅하기 위해 컴퓨팅 디바이스가 찾을 수 있는 가상 경계 코드들과 함께, 판독-기록 메모리에 저장한다. 이는 다양한 우수한 블록/불량한 블록 방식들을 매우 다양하게 수용하도록 컴퓨팅 디바이스들을 프로그래밍해야하는 필요성을 방지한다. 이러한 방식으로, 부가된 가상 경계 코드들을 갖는 동일한 데이터 이미지 부분들은 추가의 판독-기록 디바이스들에 저장될 수도 있다. 상기 추가의 판독-기록 디바이스들은 심지어, 서로 상이한, 페이지, 블록 및/또는 디바이스-특정 불량-블록 알고리즘 특징들을 가질 수도 있다.

[0020]

컴퓨팅 디바이스들은 전송된 방법을 사용하여 생성된 데이터 이미지를 활용하는 판독-기록 메모리로부터 데이터 이미지를 판독하는 일 양태 방법을 구현할 수도 있다. 이 방법은, 판독-기록 메모리가 부트 로더 이미지를 저장하기 위해 사용될 경우 컴퓨팅 디바이스를 위해 실행되는 부트 로더 루틴에서 구현될 수도 있다. 상기 부트 로드 이미지는 컴퓨팅 디바이스를 위해 실행된 독점적인 부트 로더일 필요가 없다. 이 방법은, 판독-기록 메모리의 가상 블록 내의 데이터에 액세스하는 것을 포함한다. 가상 블록 사이즈에 의해 나머지 없이 나누어 떨어지는 판독-기록 메모리 디바이스의 실제 블록 사이즈는, 실제 블록 사이즈가 가상 블록 사이즈와 동일하다는 시나리오를 포함한다. 액세스되고 있는 완전한 데이터 이미지가 가상 블록에 포함되지 않는다면, 그 방법은 다른 가상 블록의 경계 (예컨대, 시작부)를 지정하는 가상 경계 코드를 찾기 위해 판독-기록 메모리에서 전방향으로 스캔할 수도 있다. 가상 경계 코드가 인식될 경우, 부트 로더 루틴과 같은 컴퓨팅 디바이스 데이터 로딩 루틴은 다른 가상 블록 내의 데이터를 액세스할 수도 있다. 메모리가 전송된 방법에 의해 로딩되었기 때문에, 가상 경계 코드는 가상 경계 코드가 식별되었던 가상 블록이 우수하고, 판독될 수도 있는 데이터 이미지의 일부분을 포함한다고 표시한다. 데이터의 이러한 추가 블록에서 판독 이후에, 완전한 데이터 이미지가 여전히 액세스되지 않는다면, 컴퓨팅 디바이스 데이터 로딩 루틴은 전체 데이터 이미지가 판독될 때까지 추가의 가상 경계 코드들 및 데이터 블록들에 대하여 메모리를 스캔하는 것을 계속한다. 데이터 이미지의 양/사이즈는 이미지 헤더 정보로부터 결정될 수도 있다. 이러한 프로세스는, 데이터 로딩 루틴이 전체 데이터 이미지가 로딩된 것을 인식할 때, 종료할 수도 있다. 가상 경계 코드들에 대하여 메모리를 스캔함으로써, 불량한 블록들은 (그들이 가상 경계 코드를 포함하지 않을 것이기 때문에) 자동으로 스킵될 것이며, 어떤 블록들이 불량이고 우수한지를 미리 결정하는 컴퓨팅 디바이스에 대한 필요성 없이 오직 우수한 블록들로부터 데이터가 판독될 것이다. 다른 양태에 따르면, 그 방법은 제 1 블록에 (예컨대, 데이터 이미지의 프리앰블 부분에) 저장된 정보에 기초하여 데이터 이미지의 사이즈 또는 가상 블록 사이즈들을 정할 수도 있다. 판독-기록 메모리의 페이지 사이즈는 제 1 블록 또는 데이터 이미지의 프리앰블 부분으로부터 유추될 수도 있고, 결국 가상 블록 당 페이지들의 수를 결정하기 위해 사용될 수도 있다.

[0021]

추가 양태는 예컨대, 메모리에 저장된 기존의 데이터 이미지가 업데이트되거나 교체될 경우, 업데이트되거나 새로운 데이터 이미지로 플래시 메모리를 재기록하는 방법을 포함한다. 이 방법은 판독-기록 메모리의 적어도 하나의 초기 가상 블록에 초기 데이터 이미지를 포함시킨다. 또한, 판독-기록 메모리에서 사용가능한 적어도 하나의 후속 가상 블록에 대하여 적어도 하나의 초기 가상 블록을 전방향으로 스캔한다. 새로운 데이터 이미지의 제 1 부분은 후속 가상 블록에 저장될 수도 있고, 부가된 가상 경계 코드를 포함한다. 그 방법은, 전체 새로운 데이터 이미지가 저장된 후에, 적어도 하나의 초기 가상 블록으로부터 초기 이미지 데이터를 소거하는 것을 더 포함할 수도 있다. 새로운 데이터 이미지는 적어도 하나의 초기 가상 데이터 블록에 재기록될 수도 있다. 사실, 새로운 데이터의 재기록과 이전 데이터의 소거는 하나의 액션으로 수행될 수도 있다. 그러나, 새로운 데이터 이미지가 맨 처음 가상 블록 내에 기록되어야 할 필요는 없고, 뒤따르는 하나 이상의 가상 블록들에 기록될 수 있다. 전체 새로운 데이터 이미지가 적어도 하나의 초기 가상 블록 또는 맨 처음 가상 블록을 뒤따르는 하나 이상의 가상 블록들에 기록되면, 방법은 또한, 이 방법의 부분으로서 일시적으로 기록되었던, 후속하는 가상 블록(들)로부터 새로운 데이터 이미지를 소거하는 것을 포함할 수도 있다.

[0022]

최초에 파워 온되었을 때 컴퓨팅 디바이스의 프로세서는 일반적으로 ROM 또는 RAM 에 로딩된 오퍼레이팅 시스템을 갖지 않는다. 따라서, 그 프로세서는 초기에, 시스템의 부트 업 프로세스를 시작하는 부트 로더로서 지칭되는 ROM 펌웨어에 저장된 프로그램 코드를 실행한다. 부트 로더의 임무는 다른 데이터 및 프로그램들을 랜덤 액세스 메모리 (RAM) 내에 로딩하여 그 후 프로세서에 의해 실행되게 하는 것이다. 일차 부트 로더는 일반적으로 ROM 에 저장되고, 파워-업시 또는 디바이스가 리셋될 경우에 실행된다. 그 후에, CPU 에 동작가능하게 커플링된 NAND 디바이스에 저장된 하나 이상의 이차 부트 로더들은 새로운 및/또는 추가의 애플리케이션 코드가 로딩되고 실행되게 할 수도 있다. 종래의 다중-스테이지 부트 로더들은, 복잡도가 증가하는 몇몇 프로그램들이 체인 로딩 프로세스에서 교대로 로딩하는 동안 사용된다. 그러나, NAND 플래시 메모리에서, 상

이한 페이지 및/또는 블록 사이즈들 및 메모리의 불량한 블록 특징들에 걸쳐 스케일링가능한 단일 가변 사이즈 부트 로더를 가지는 것은 유리할 수도 있다. 그러한 NAND 메모리 부트 로더는 여전히 이차 부트 로더일 수도 있고 및/또는 다른 부트 로더 프로그램들과 함께 작업할 수도 있다.

[0023]

NAND 메모리 제조업자들 및 재판매자들은 그들의 메모리 디바이스들에서 불량한 블록들을 식별하기 위해 균일한 표준 포맷을 사용하지 않는다. 그러나, 컴퓨팅 디바이스는 메모리에서 불량한 블록들을 어떻게 취급할 것인지 알아야만 하며, 그렇지 않으면 적절히 또는 일관적으로 작동하지 않을 것이다. 결과적으로, 불량한 블록 소프트웨어 루틴은 통상적으로 데이터 로더 루틴들 내에 포함되며, 불량한 블록들을 수용하고 컴퓨팅 디바이스와 함께 사용되는 것으로 예측될 수도 있는 제조업자들/재판매자들의 NAND 메모리 디바이스들을 각각 핸들링하는 방법들을 제공하도록 구성된 이차 부트 로더를 포함한다. 그러나, NAND 메모리 칩들은 다수의 제조업자들 또는 재판매자들 중 어떤 것으로부터도 구매될 수도 있고, 재판매자들은 그들의 불량 블록 검출 정보/방법들을 변화시킬 수도 있다. 현재의 불량한 블록 소프트웨어 루틴들은, 다수의 상이한 루틴들을 그들의 컴퓨팅 디바이스들 내에 프로그래밍된 부트 로더(들) 내에 포함시킴으로써, 다양한 상이한 불량 블록 검출 방식들을 취급한다. 이는 재판매자들의 다양한 요건들 및 사양들을 취급하려는 시도로 실행된다. 불량한 블록 루틴들에서 그러한 가변성을 수용하는 것은, 부트 로더 ROM 펌웨어를 복잡하게 할 수 있고, 그러한 루틴들을 재판매자 변화들을 수용하도록 변경시키는 것은 일반적으로 바람직하지 않다. 또한, 제 1 부트 로더는 우수한 것이 보장되는 NAND 메모리의 최소 대표 블록 0 내에 전체가 상주하는 것으로 설계될 수도 있지만, 이는 시스템의 전체 부트-업을 복잡하게 만들 수도 있고, 종종 드라이버들과 부트 업 로드들을 복제한다. 본 명세서의 일 양태는 그러한 복잡함 및/또는 복제를 컷 다운하고, ROM 이 제 1 우수한 블록보다 많이 걸쳐 있는 부트 로더를 로딩할 수 있게 한다.

[0024]

개시된 실시형태들은 데이터 이미지를 메모리 칩들에 로딩하는 프로세스를 위해 NAND 메모리 재판매자들에 의해 구현되는 다양한 불량 블록 검출 방식들을 제한 및/또는 삭제하기 위한 방법들을 포함한다. 가상 경계 코드들을 부가함으로써, 컴퓨팅 디바이스들이 불량한 블록들을 찾기 위해 리소스들을 사용하지 않고 로딩될 데이터의 블록들을 식별할 수 있게 한다. 프로세싱되고 있는 (즉, 데이터 이미지가 로딩되고 있는) 판독-기록 메모리들에 대하여 일반적인 불량 블록 검출/핸들링 루틴을 구현하는 것은 머신 셋업의 부분으로서 구현될 수 있다.

[0025]

다양한 양태들에서, 판독-기록 메모리에 저장될 데이터 이미지는, 각각 가상 블록들에 대응하는 데이터 이미지의 포지션들에 위치되는 하나 이상의 가상 경계 코드들로 구성된다. 가상 블록들은 컴퓨팅 디바이스에서 구현될 수도 있는 판독-기록 메모리들의 블록 사이즈와 호환가능하도록 사이징된다. 가상 블록 사이즈들은 메모리들의 실제 블록 사이즈들에 매칭할 수도 있다. 그러나, 메모리들의 상이한 구성들 (즉, 상이한 사이즈의 실제 블록들을 갖는 메모리 칩들) 을 수용하기 위해, 가상 블록 사이즈는 컴퓨팅 디바이스에서 구현될 수도 있는 모든 메모리들의 실제 블록들의 최저 공통 분모 사이즈로 세팅될 수도 있다. 이는 양태 방법들이 메모리의 각 타입에 대하여 별개의 데이터 이미지를 요구하지 않고 상이한 페이지 사이즈들 (예컨대, 2K 및 4K) 의 NAND 디바이스들을 지원할 수 있게 한다. (적어도 초기 블록 이후의) 가상 블록들 내의 규칙적인 포지션들에 가상 경계 코드들을 부가함으로써, 동일한 데이터 이미지가 구현될 수도 있는 임의의 메모리의 우수한 블록들에 저장될 수도 있고, 임의의 개재하는 불량한 블록들은 스킵될 수 있다. 그러한 가상 경계 코드들은 시작부와 같은, 가상 블록의 거의 모든 곳에 부가될 수도 있다.

[0026]

데이터 이미지가 판독-기록 메모리, 예컨대 NAND 메모리 상에 저장될 경우, 재판매자-특정 불량 블록 검출 방법들은 메모리에서 불량한 블록들을 회피하고 오직 우수한 블록들만을 사용하기 위해 사용된다. 그러나, 종래의 시스템들은 재판매자-특정 불량 블록 검출 방법들을 또한 사용하기 위해 그 데이터 이미지를 차후에 판독하는 루틴들을 요구한다. 대조적으로, 개시된 실시형태들의 양태들은, 가상의 어떤 판독-기록 메모리 디바이스에도 적용가능한 더 일반적인 방법을 제공하여, 데이터 이미지를 판독할 때, 그러한 재판매자-특정 방법들을 사용해야할 필요성을 최소화하거나 제거한다. 본 명세서에 개시된 양태들은 판독-기록 메모리 상의 하나 이상의 가상 블록들에 데이터 이미지를 저장한다. 데이터 이미지는 판독-기록 메모리의 하나 이상의 가상 블록들에 저장될 수도 있다. 1 초과개의 가상 블록이 전체 데이터 이미지를 수용하기 위해 요구된다면, t 개의 다음 사용가능한 가상 블록 (또는 요구되는 경우 블록들) 은, 전체 데이터 이미지가 저장될 때까지 임의의 개재하는 불량한 블록들을 스킵하고 데이터 이미지의 나머지 부분을 저장하기 위해 사용될 수도 있다. 가상 블록들이 정의되고, 각각의 가상 블록에 배치된 데이터 이미지의 부분들에 부가된 가상 경계 코드들에 의해 경계가 정해진다. 일 실시형태는 각각의 가상 블록의 시작부에 가상 경계 코드를 배치한다. 또한, 맨 처음 가상 블록은, 특히 디바이스 상의 제 1 블록이 일반적으로 우수한 것으로 보장되기 때문에, 가상 경계 코드

요구하지 않을 수도 있다. 또한, 가상 경계 코드들이 다음 및/또는 후속하는 가상 블록들의 시작부에 배치된다면, 그 코드들은 데이터 이미지의 포지션들이 저장되는 위치들을 식별한다. 가상 경계 코드들이 부가된 데이터 이미지는 오직 메모리의 우수한 실제 블록들에만 저장되고, 따라서 가상 경계 코드들은 메모리의 우수한 실제 블록들 내의 우수한 가상 블록들에 대응한다. 결과적으로, 부트 로더는 제조업자 또는 재판매자에 관계없이 가상의 어떤 관독-기록 메모리도 수용하는 방식으로, 가상 경계 코드들을 인식하고 관독할 메모리의 블록들을 결정하기 위해 본 명세서의 양태들에 따른 단일 이미지 관독 프로세스를 사용할 수 있다. 양태 방법들은 컴퓨팅 능력들을 갖는 스마트폰 또는 다른 전자장치들과 같이, 전술된 것과 같이 프로그래밍된 NAND 메모리로부터 디바이스들을 부트 업하기 위해 사용될 수 있다. 또한, 양태 방법들은 이차 부트 로더의 페일-세이프 (fail-safe) 업데이트를 핸들링하기 위해 적용될 수도 있다.

[0027]

양태 방법들은 파워 업 또는 재시작시 로딩된 초기 소프트웨어로서, 플래시 메모리에 저장되고 스마트폰 또는 다른 모바일 컴퓨팅 디바이스와 같은 컴퓨팅 디바이스에 의해 사용되는 초기 부트 소프트웨어 이미지에 특히 적용가능할 수도 있다. 통상적으로 그러한 부트 소프트웨어는 오퍼레이팅 시스템, 프로세서 또는 디바이스 제조업자에 의해 엄격하게 제어된다. 부트 소프트웨어를 제어하는 회사는 통상적으로 일차 부트 로드 이미지를 ROM 메모리에 "버닝 (burn)" 하고, 그 후에 컴퓨팅 디바이스는 초기 소프트웨어 이미지를 로딩하기 위한 부트 루틴의 부분으로서, NAND 메모리와 같은 관독-기록 메모리에 액세스한다.

[0028]

양태 방법들, 시스템들 및 디바이스들은 가상의 어떤 NAND 디바이스로부터도 (본 명세서에서 또한 부트 로더로서 지칭되고, "BL" 로 축약되는) 단일 이차 부트 로더 아키텍처를 인에이블하는, 불량 블록 핸들링을 제공한다. 현대의 시스템들에서, 이차 부트 로더의 사이즈는 내부 SRAM 메모리의 사이즈에 의해 지시될 수도 있다. 따라서, 예를 들어, SRAM 사이즈가 NAND 디바이스에서 256KB 이면, 블록 0 에 일반적으로 배치된 프리앰블에 대하여 수 킬로바이트들을 감산하고, 증명들 (Cert) 또는 패딩 (PAD) 과 같은 기타 추가의 데이터에 대하여 추가의 킬로바이트들을 감산하는 것은 초기 실제 블록에서 약 240KB 를 남겨놓을 수도 있다. 그 240KB 사이즈는 다수의 개별 부트 로더들의 최대 사이즈에 대응하고, 오늘날 관독-기록 메모리들에서 일반적으로 구현되는 128KB 의 최소 NAND 블록 사이즈보다 크다. 본 명세서의 일 양태는 가상 블록 사이즈를 128KB 의 현대의 최소 NAND 블록 사이즈에 대응하도록 세팅한다. 따라서, 1 초과의 가상 블록이 그 SRAM 로부터 240KB BL 을 실행하기 위해 일차 부트 로더에 의해 관독되어야만 한다. 일부 현대 FLASH 메모리 디바이스들은 최소 128KB 보다 큰 블록 사이즈들을 가지고, 일부는 256KB 및 개발 중에 있는 더 큰 메모리들을 가지며, 그러한 더 큰 실제 블록들은 128KB 의 2 이상의 가상 블록들을 포함할 수도 있다. 또한, 본 명세서의 방법들의 양태들은 또한, 32KB 와 같은 더 작은 블록 사이즈들로 확장될 수도 있다.

[0029]

128KB 세그먼트 사이즈가 본 명세서에서 오직 예시의 목적으로 일 예로서 사용되며; 세그먼트들은 고려중에 있는 특정 NAND 디바이스들에 의존하여 더 커지거나 더 작아질 수도 있음이 이해되어야만 한다. 예를 들어, 128KB 가상 블록 사이즈가 사용된다면, 그 사이즈는 블록당 64 페이지 미만 또는 블록당 128KB 미만인 NAND 디바이스들을 지원하지 않을 수도 있다. 그러나, 더 작은 블록 사이즈는, 요구되는 가상 블록들의 수가 가상 블록 사이즈가 더 작아짐에 따라 증가할 수도 있기 때문에, 실질적인 제한들을 갖는다. 더 많은 가상 블록들은 유효 데이터 이미지를 발견하기 위해 요구되는 더 많은 스캔을 의미하며, 이는 증가된 부트 업 시간들로 해석할 수도 있다.

[0030]

양태 방법들은 NAND 디바이스 상에 배치된 BL 이미지를 128KB 가상 블록 사이즈 내에 배치될 수 있는 세그먼트들로 분할한다. 128KB 가상 블록 사이즈는 NAND 디바이스들에 대한 현재 최소치에 대응하기 때문에 선택되며, 상이한 가상 블록 사이즈가 원하는 바에 따라 사용될 수도 있다. 가상 블록들은 적어도 제 2 의 128KB 세그먼트 내에 마커들을 부가함으로써 생성된다. 제 1 의 128KB 세그먼트는 일반적으로 BL 프리앰블 및 다른 관독가능 코드들을 포함하며, 따라서 가상 경계 코드를 포함하지 않아도 된다. 가상 경계 코드에 의해 마킹되는 각각의 세그먼트는 가상 블록 및 그 경계들 (가상 블록의 2 개 단부들 - 또한 본 명세서에서 "가상 블록 경계" 로서 지칭됨) 을 정의한다. BL 이미지의 나머지 부분들은 전체 이미지를 수용하기 위해 요구되는 것만큼 많은 가상 블록들에 걸쳐 분할되고 분배된다. 팩킹 데이터는 데이터의 불완전한 가상 블록을 채우기 위해 부가될 수도 있다.

[0031]

다양한 양태들은 ROM 펌웨어의 개발자들이 더이상 NAND 메모리 재판매자-정의된 불량 블록 검출 방식들을 수용할 필요가 없게 한다. 다양한 양태들은 불량/우수 블록 테이블을 보유하는 사전 프로그래밍된 페이지에 의존하지 않는다. 개시된 개시물들은 상이한 NAND 디바이스들에 대하여 이전에 개별 툴들/빌드들을 요구한, 플래시 툴 및 빌드 관리의 설계를 간략화한다. 추가로, 개시된 디바이스, 시스템 및 방법들은 부트 로더의 페일-세이프 업데이트 및 그 핸들링을 가능하게 한다. 추가로, 이러한 NAND 설계는 상이한 페이지/사이즈

NAND 디바이스들을 지원하도록 스케일링가능하다.

[0032]

도 1 은 다양한 양태들에 따라 독립형 부트 로더 (15) 를 갖는 NAND 디바이스 (100) 의 패키징을 예시한다. 이 실시형태에서, 128KB 가상 블록 사이즈가 사용되며, 전술된 것과 같이 2KB 또는 4KB NAND 디바이스를 위해 작용한다. 따라서, 128KB 가상 블록을 확립하기 위해, 가상 블록 경계는 가상 경계 코드 (14) 에 의해 지정될 수도 있다. 그러한 가상 경계 코드 (14) 는 12 바이트의 데이터만큼 작을 수 있는 고유하고 인식가능한 데이터 패터 (patter) 이다. 일 예로서, 가상 경계 코드 (14) 는 16진법 숫자일 수도 있다: 844bdc5673531014 d48b54c6. 이보다 크거나 작은 가상 경계 코드 (14) 가 사용될 수도 있지만, 오직 작은 양의 데이터만이 가상 블록 경계를 정하는데 요구되기 때문에, 클 필요는 없다. 또한, 블록 0 의 시작부에서 일차 부트 로더 스택, 공유 영역들 및 다른 데이터를 위해 예비된 SRAM 의 추가 부분들이 고려될 수도 있다. 따라서, 사용가능한 SRAM 은 실제 SRAM 사이즈보다 더 작을 수도 있다.

[0033]

도 1 은 또한, 프리앰블 메모리 코드들로서 지칭될 수도 있는 하나 이상의 메모리 코드들 (10) 을 포함하는 프리앰블 (5) 을 도시한다. 몇몇 프리앰블 메모리 코드들 (10) 을 포함하고 있지만, 프리앰블은 10KB 를 초과하지 않는 것으로 가정될 수도 있다. 따라서, BL (15) 은 NAND 디바이스 (100) 의 블록 0 의 제 1 10KB 이후에 시작하여 기록될 수도 있다. 도 1 에 도시된 예에서, BL (15) 의 제 1 80 바이트들은 부트 로더 제 1 부분 (15a) 의 시작을 지정하는 이미지 헤더 (12) 를 포함하고, 가상 블록들의 사이즈 세트와 같은 추가 정보를 포함할 수도 있다. 그 후에, 가상 블록 경계는 128KB 경계에 가상 경계 코드 (14) 를 부가함으로써 마킹될 수도 있고, 다음 128KB 에서 부트 로더 제 2 부분 (15b) 이 뒤따른다. 그렇지 않으면, 가상 경계 코드 (14) 는 통상의 이차 부트 로더의 중간에 놓일 것이며, 따라서 BL (15) 는 1 초과의 부분 (15a, 15b) 으로 분할된다. 프리앰블 사이즈, 이미지 헤더 사이즈 및 가상 블록의 사이즈를 인식함으로써, 제 1 부트 로더 부분 (15a) 의 사이즈에 대하여 결정될 수 있다. 그 후에, 부트 로더의 나머지는 부트 로더 제 2 부분 (15b) 또는 필요한 경우 추가 가상 블록들에서 추가의 부분들에 포함될 수도 있다. 따라서, 일부 알려지거나 가정된 변수들은 분할된 BL (15) 의 각 부분이 얼마나 클 수 있는지의 결정에 포함될 수도 있다. 예를 들어, 다음 파라미터들을 고려한다:

[0034]

● SRAM 사이즈 = 256KB

[0035]

● 예비된/사용된 SRAM = ~16KB

[0036]

● 가상 블록 사이즈 = 128KB

[0037]

● (프리앰블 메모리 코드들 (10) 을 포함하는) 프리앰블 (5) 사이즈 = 10KB

[0038]

● 부트 로더 이미지 헤더 (12) 사이즈 = 80B

[0039]

● 패딩 (18) 을 더한 서명 및 증명들 (16) = 6KB

[0040]

● 가상 경계 코드 (VBC) 사이즈 = 12B.

[0041]

SRAM 사이즈로부터 상기 사용된 바이트들 전체를 감산하는 것은 2 개의 128KB 가상 블록들에 저장될 수 있는 NAND 메모리의 부트 로더를 위해 대략 223KB 를 남긴다.

[0042]

도 2 는 부트 로더 (15) 의 간격 및 분배를 고려하는 다른 방식을 도시한다. 먼저 20 에서, 가상 블록 사이즈는 128KB 에서 확립될 수도 있고, 따라서 2 개의 그러한 가상 블록들은 256KB SRAM 에 포함될 것이다. 22 에서, BL (15) 의 사이즈는 가상 블록들 중 적어도 2 개에 걸칠 수도 있는 것으로 고려될 수도 있다. 그 후에, 24 에서 이미지 헤더 (12) 및 증명들 (16) 이 BL (15) 이미지의 전체 길이에 추가된다. 또한, 26 에서, 프리앰블 메모리 (10) 를 포함하는 프리앰블 (5) 및 패딩 (18) 이 이미지의 전체 길이에 추가된다. 추가로, 28 에서, 가상 경계 코드는 이 실시형태에서 제1 의 128KB 가상 경계에 대응하는, BL (15) 의 2 개 부분들 (15a, 15b) 사이에 부가된다. 이들 파라미터들을 고려하여, 실제 디바이스의 페이지 사이즈들은, 얼마나 많은 실제 블록들이 사용될 것인지 뿐만 아니라 얼마나 많은 가상 블록들이 사용될 것인지를 결정할 것이다. 예를 들어, 사이즈가 4K 인 페이지들을 갖는 NAND 디바이스를 고려하자. 그러한 디바이스는 256KB 의 실제 블록들을 가지며, 블록 0 에서 부트 로더를 위해 대략 223KB 를 남겨둔다. 따라서, 4K NAND 디바이스는 이러한 BL 을 유지하기 위해 오직 하나의 우수한 블록을 필요로 할 것이다. 128KB 의 가상 블록 사이즈가 사용된다면, 그 하나의 우수한 실제 블록은 2 개의 가상 블록들을 포함할 것이다. 지금부터, 사이즈가 오직 2K 인 페이지들을 갖는 NAND 디바이스를 고려하자. 이 디바이스는 2 개의 우수한 실제 블록들을 요구할 것이다. 이들 실제 블록들의 사이즈는 128KB 의 가상 블록 사이즈와 일치한다. 따라서, 2K NAND 의 블록

0 은 오직, 제 1 BL 부분 (15a) 을 위해 대략 118KB 의 공간을 남겨둔다 (10KB 프리앰블 (5) 과 80KB 이미지 헤더 (12) 를 빼고 128KB). 그 후에, 나머지 제 2 BL 부분 (15b) 은 가상 경계 코드 (14) 에 의해 지정된 다음 우수한 블록에 위치될 수도 있다.

[0043]

도 3 은 다양한 시나리오들에서 어떻게 예시적인 부트 로더 분배가 발생될 수도 있는 지를 도시한다. 도시된 5 개 시나리오들은 모두 (5 개 시나리오들 위에 가상 블록 시리즈로서 도시된) 128KB 가상 블록 사이즈를 사용한다. 시나리오들 중 3 개는 2KB NAND 디바이스를 고려하고, 다른 2 개 시나리오들은 4KB NAND 디바이스를 고려한다. 2KB 시나리오들에서, 가상 블록 사이즈는 128KB 의 실제 블록 사이즈와 일치한다. 4KB 시나리오들에서, 가상 블록들은 실제 블록들의 사이즈의 절반이다.

[0044]

2KB 시나리오 1 에서, 부트 로더의 분할 및 분배는 간단하며, 경계 코드 (14) 가 블록 (1) 의 시작부에 위치되고, 이는 디바이스 상에 마킹된 제 1 가상 블록 경계와 일치한다. 가상 블록들은 2KB 시나리오들에서 실제 블록들과 일치하는 것에 유의하여야 한다. 전송된 NAND 메모리의 코딩과 마찬가지로, 블록 0 은 또한, 프리앰블 메모리 코드들 (10), 부트 로더 이미지 헤더 (12) 및 제 1 부트 로더 부분 (15a) 을 갖는 프리앰블 (5) 을 포함한다. 제 2 부트 로더 부분 (15b) 은, 제 2 부트 로더 부분 (15b) 에 추가되고 이 실시형태에서 가상 블록의 시작부를 표시하는, 가상 경계 코드 (14) 를 뒤따른다. 이러한 2KB 시나리오 1 는 블록 1 이 우수한 NAND 디바이스에 적용하며, 따라서 제 1 및 제 2 가상 블록들은 연속하는 우수한 블록들이다. 그러나, 제조업자들은 오직, 일반적으로 블록 0 이 우수한 것만을 보장하며, 따라서 2K NAND 디바이스에 대하여 제 2 및 제 3 시나리오들을 고려한다. 2KB 시나리오 2 에서, 블록 1 은 불량하지만 블록 2 는 우수하다. 따라서, 블록 1 이 불량하기 때문에, 가상 경계 코드 (14) 는 그 블록에 기록되지 않을 수도 있다. 오히려, 다음 우수한 블록, 즉 블록 2 에서 가상 경계 코드 (14) 가 제 2 부트 로더 부분 (15b) 과 함께 기록된다. 2KB 시나리오 3 는 블록 0 다음에 2 개의 초기 불량한 블록들 (블록 1 및 블록 2) 을 갖는다. 따라서, 이 경우, 가상 경계 코드 (14) 는 이들 2 개의 불량한 블록들 중 최종 블록 이후의 경계에 기록되며, 이 경계는 다음 우수한 블록 (블록 3) 의 시작부이다. 그 시나리오에서와 유사하게, BL 의 나머지 (BL 제 2 부분 (15b)) 가 블록 3 에 기록된다.

[0045]

4KB 시나리오들에서, 블록 1 이 우수한지 또는 불량한지의 여부는 사실상 중요하지 않으며, 이는 부트 로더가 일반적으로 2 개의 가상 블록들을 포함하기에 충분히 큰 블록 0 내에서 완전히 수용될 수도 있기 때문이다. 따라서, 4KB 시나리오 1 는, 블록 1 이 우수하지만 BL 제 1 부분 (15a) 과 BL 제 2 부분 (15b) 양자가 블록 0 내에 완전히 기록될 수 있기 때문에 요구되지 않는 상황을 도시한다. 또한, 4KB 시나리오 2 는, 그 시나리오에서 블록 1 이 불량한 것과 무관하게, BL 15a, 15b 의 양자의 부분들이 어떻게 블록 0 내에 완전히 기록될 수 있는지를 도시한다.

[0046]

도 4 는 부트 로더 분배들의 2 이상의 시나리오들을 도시한다. 도 4 의 제 1 시나리오는 4KB 시나리오 3 이고, 3 개의 부분들 (15a, 15b, 15c) 로 분할되어야만 하는 더 큰 BL 을 포함한다. 제 2 의 실제 블록, 블록 1 이 우수하기 때문에, 데이터 이미지는 연속하는 가상/실제 블록들을 넘어서 확산된다. 그러나, 블록 1 이 불량하다면, 제 2 가상 경계 코드 (14) 와 BL 제 3 부분 (15c) 은 블록 2 에 저장되었을 수 있다. 도 4 에서 제 2 시나리오는 2KB 시나리오 4 이다. 다시 한번, 3 개의 부분들 (15a, 15b, 15c) 로 분할되어야만 하는 더 큰 BL 이 사용되지만, 여기서 제 3 블록, 블록 2 은 불량하다. 따라서, 제 4 블록, 블록 3 이 가상 경계 코드 (14) 와 BL 제 3 부분 (15c) 을 저장하는데 사용된다.

[0047]

도 5 는 관독-기록 메모리 디바이스에 데이터 이미지를 저장하는 일 양태의 방법 (500) 을 도시한다. 데이터 이미지는 본 명세서의 양태들에 따라 부트 로더일 수도 있다. 블록 (510) 에서, NAND 디바이스와 같은 관독-기록 디바이스가 데이터 저장을 위해 로딩되고 준비된다. 블록 (520) 에서, 관독-기록 메모리 디바이스를 위해 지정된 가상 블록들의 시리즈의 가상 블록 사이즈에 기초하여, 데이터 이미지를 패키징하기 위한 데이터 이미지 분배가 결정된다. 이러한 결정의 부분으로서, 데이터 이미지는, 전체 데이터 이미지가 단일 가상 블록에 대하여 너무 큰 경우, 적어도 데이터 이미지 제 1 부분 및 데이터 이미지 제 2 부분으로 분할되어야만 할 수도 있다. 또한, 가상 경계 코드는 데이터 이미지 제 1 부분 및 데이터 이미지 제 2 부분 중 적어도 하나에 추가된다. 이 실시형태에서, 가상 경계 코드는 데이터 이미지 제 2 부분의 시작부에 추가되고, 따라서 블록 (530) 에서, 데이터 이미지 제 1 부분은 가상 블록들의 시리즈 중 제 1 가상 블록에 저장될 수도 있다. 블록 (540) 에서, 가상 경계 코드 (VBC) 가 저장되고, 다음 가상 블록을 마킹하며 (및 따라서 가상 블록들의 시리즈 중 제 2 가상 블록을 지정하며), 이 블록은 자연스럽게 우수한 실제 블록과 일치할 것이다. VBC 를 저장하고 가상 블록을 지정할 시, 제 1 가상 블록과 제 2 가상 블록 사이에 임의의 개재하는 불량한 블록들은 관독-기록 메모리 내에서 스킵된다. 데이터 이미지를 물리적으로 기록하는데 사용되는 프로그래밍 툴들은

공장에서 마킹된 불량한 블록들에 대하여 재판매자 특정 체크들을 수행하고, 필요한 경우에 불량한 블록들을 스킵할 수도 있다. 프로그래밍 툴들의 목적은, 데이터 이미지 내에 VBC들을 저장 및/또는 통합하는 것일 수도 있다. 따라서, 프로그래밍 툴들은 VBC들을 인식하거나 취급하지 않아야만 하는데, 이는 VBC들이 재판매자 불량한 블록 관리 방식들에 독립적으로 판독될 수 있는 패턴들을 생성하기 위한 데이터 이미지의 일부로 형성되기 때문이다. 그러한 프로그래밍 툴은 또한 NAND 드라이버를 가질 수도 있지만, ROM 에서의 NAND 드라이버와 다르게, 프로그래밍 툴 NAND 드라이버는 사이즈 또는 복잡도가 제한되지 않아야만 한다. 이는 프로그래밍 툴들이 제품에 대한 수정들 또는 업데이트들 이후 버그들을 고정하고, 심지어 새로운 NAND 재판매자-특정 불량한 블록 체크 방식들을 핸들링하게 한다. 이러한 방식으로, ROM 에서의 NAND 드라이버와 비교하여, 그러한 프로그래밍 툴들에 기능성이 부가되거나, 삭제되거나, 변화되거나, 강화될 수도 있다. 블록 (550) 에서, 데이터 이미지의 제 2 (다음) 부분은 가상 블록들의 시리즈 중 제 2 가상 블록인 VBC 와 동일한 가상 블록에 저장될 수도 있다. 결정 블록 (560) 은 완전한 데이터 이미지가 저장되었는지 여부를 결정한다. 그렇지 않다면, 방법은 블록 (540) 으로 리턴하여 다음 사용가능한 가상 블록에서 다른 VBC 를 저장하고, 다시 블록 (550) 에서 데이터 이미지의 다음 블록을 저장한다. 이러한 루프는 데이터 이미지의 모든 부분들이 저장될 때까지 진행할 것이다. 결정 블록 (560) 에서 완전한 데이터 이미지 (예컨대, 부트 로더 이미지) 가 저장되었다는 긍정적인 결정이 수행되면, 프로세스는 570 에서, 블록 (510) 의 시작에서 로딩된/준비된 판독-기록 디바이스에 대한 결론에 도달한다. 따라서, 프로세스는 추가의 판독-기록 디바이스들과 직렬로 또는 병렬로 반복될 수 있다.

[0048]

가상 블록의 시작부에 VBC 를 위치시키는 것은 우수한 블록들을 즉시 검출하는 것을 돕고, 부트-로딩 프로세스를 최적화시킬 수도 있다. 가상 블록의 단일 패스 판독은 VBC 를 검출할 수도 있고, 동시에 그 블록에 포함된 부트 로더 데이터 이미지를 판독할 수도 있다. 대안적으로, 가상 경계 코드들은 가상 블록들의 시작부에 위치될 필요가 없다. 예를 들어, VBC 는 (블록 0 을 포함하는) 각각의 가상 블록의 단부에서 또는 요구되는 가상의 어떤 포지션에도 위치될 수도 있다. 하나 이상의 VBC들과 관련된 포지셔닝 정보는 전송된 이미지 헤더에 포함될 수도 있다.

[0049]

NAND 디바이스에 하나 이상의 가상 경계 코드들이 기록되면, 일차 부트 로더는 그 디바이스를 파싱하여 가상 경계 코드들을 발견하고, 가이드포스트들과 같은 가상 경계 마커들을 사용하여 완전한 이차 부트 로더를 판독하고, 인증하고, 실행할 수 있다. 도 6 은 판독-기록 디바이스로부터 데이터 이미지를 판독하는 일 양태의 방법을 도시한다.

[0050]

도 6 은 판독-기록 메모리로부터 데이터 이미지를 판독하는 일 양태의 방법 (600) 을 도시한다. 예를 들어, 일차 부트 로더에서 NAND 파서는 이차 부트 로더를 판독하고, 로딩하고, 인증하기 위해 방법 (600) 을 구현할 수도 있다. 블록 (605) 에서, 판독-기록 디바이스가 판독되고, 그 디바이스의 페이지 0 과 일치하는 블록 0 에서의 프리앰블로 시작한다. 프리앰블은 데이터 로딩 루틴에 의해 사용될 수도 있는 디바이스 폭 및 페이지 사이즈 정보를 포함할 수도 있다. 블록 (610) 에서, 에러 정정 코딩 (ECC) 검출이 수행될 수도 있고, 이는 ECC 가 현재 방법을 위해 바람직하게 인에이블되기 때문이다. 이 동작은 4 비트 BCH 에러 정정 코딩 또는 다른 코딩으로 디폴트할 수도 있다. 프로그래밍된 페이지의 판독이 성공적이지 않아 ECC 가 검출되지 않는다면, 블록 (612) 에서 추가의 ECC 구성들이 체크될 수 있다. 예를 들어, 8 비트 BCH ECC 가 사용될 수도 있거나, 디폴트가 아닐 경우 4 비트 BCH ECC 가 사용될 수도 있다. 따라서, 하나 이상의 추가의 ECC 구성들이 체크될 수 있고, 따라서 어떤 ECC 도 블록 (614) 에서 검출되지 않는다면, 블록 (616) 에서 어떤 부트 로더 (BL) 도 발견될 수 없는 것을 의미하는 결론에 도달하고, 프로세스는 종료한다. 대안적으로, 그 방법은 AUTO_DETECT 루틴 및 가중 알고리즘 검사를 사용하여 유효 프리앰블에 대하여 체크할 수도 있다. 그렇지 않으면, ECC 가 블록 (610) 또는 블록 (614) 에서 검출된다면, 그 방법은 블록 (620) 으로 진행하고, 여기서 페이지 사이즈 검출이 수행된다. 블록 (620) 에서, 연속하는 페이지들은 실제 NAND 페이지 사이즈를 계산하기 위하여 가상 경계 코드 사이즈 및/또는 포지션을 검증하기 위해 판독될 수도 있다. 결정 블록 (620) 에서 어떤 페이지 사이즈도 검출되지 않는다면, 임의의 가상 경계 코드 (VBC) 가 검출되는지 여부에 관한 체크가 추가의 결정 블록 (622) 에서 실행된다. 622 에서 VBC 가 검출되면, 624 에서 그 오프셋 (포지션 및 사이즈) 을 표시하는 VBC 가 판독되고 다음 페이지로 이동하며, 페이지 사이즈 검출 결정 블록 (620) 으로 리턴한다. 622 에서 VBC 가 검출되지 않았다면, 블록 616 에서 어떤 부트 로더 (BL) 도 발견될 수 없다는 것을 의미하는 결론에 도달될 수도 있고, 프로세스는 종료한다. 결정 블록 (620) 에서 페이지 사이즈가 검출되면, 가상 블록 사이즈를 페이지 사이즈로 나눔으로써 가상 블록당 페이지들의 수에 관한 결정이 수행될 수도 있다. ROM 은 이러한 결정을 사용하여, 다음 가상 블록으로 진행하기 전에 얼마나 많은 고정된 페이지들이 판독되어야만 하는지를 알 수도 있다. 페이지 사이즈가 검출되면, 제 1 가상 블록의 데이터 이미지가 블록 (630) 에서 액

세스될 수도 있고, 심지어 카피될 수도 있다. 따라서, 나머지 페이지들이 그 제 1 가상 블록 밖에서 관독될 수도 있고, 그 결과 데이터 이미지의 제 1 가상 블록 내에서 모든 데이터가 액세스된다. 그 후에, 프로세스는 VBC 에 대하여 다음 가상 블록을 스캔하도록 진행할 수도 있다. 640 에서 어떤 VBC 도 검출되지 않는다면, 다음 가상 블록은 불량하고 밋/또는 손상될 수도 있다. 따라서, 블록 (642) 에서 프로세스는 가상 경계 코드에 대하여 다음 가상 블록을 스캔한다. 얼마나 많은 가상 블록들이 체크될 수도 있는지에 대한 임계치 (또한 타임-아웃 함수로서 지칭됨) 는, 프로세스가 무한 루프로 계속되지 않는 것을 보장하기 위해, 사용될 수도 있다. 따라서, 15 개 가상 블록들과 같은 임계치가 사용될 수도 있고, 644 에서 그 임계치가 도달되면, 이는 어떤 BL 발견될 수 없는 것을 의미할 수도 있고, 616 에서 프로세스는 종료한다. 가상 블록 임계치가 도달되지 않으면, 블록 (640) 에서 가상 경계 코드가 검출될 때까지, 사이클은 계속되고/블록 (640) 으로 리턴한다. 640 에서 VBC 가 검출되면, 650 에서 다음 가상 블록 내의 모든 데이터가 액세스될 수도 있다. 다음 가상 블록은, VBC 가 발견되었고 액세스될 수도 있는 가상 블록에 대응한다. 660 에서, 완전한 데이터 이미지 (예컨대, 완전한 BL) 가 관독되었는지 여부에 관한 결정이 수행될 수도 있다. 660 에서 완전한 BL 이 관독되지 않았다면, 블록 642 에서 프로세스는 가상 경계 코드에 대하여 다음 가상 블록을 스캔하고, 진술된 것과 같이 추가로 진행한다. 660 에서 완전한 BL 이 관독되었다면, 670 에서 BL 이 인증되고 실행될 수도 있으며, 따라서 제 1 및 제 2 가상 블록들에서 데이터 이미지에 의해 표현된 데이터로부터 BL 을 로딩한다. 로딩된 BL 은 오직 마커만을 의미하고 로딩될 필요가 없는 가상 경계 코드 바이트들을 스킵할 수도 있다 (포함할 필요가 없다).

[0051] 도 6 에 도시된 양태 방법은, 불량한 블록이 제 1 및 제 2 가상 블록들 사이 또는 후속 가상 블록들 사이에 배치될 때, 데이터 로딩 루틴 (예컨대, 일차 부트 로더) 가 메모리에서 우수 또는 불량한 블록들을 먼저 식별하지 않고 전체 데이터 이미지 (예컨대, 이차 부트 로더) 를 관독할 수 있도록, 어떻게 불량한 블록들이 스킵되는지를 도시한다. 따라서, 데이터 로딩 루틴은 제 1 가상 블록 다음의 불량한 블록을 스킵할 것이며, 이는 불량 가상 블록에서 어떤 가상 경계 코드도 검출되지 않을 것이기 때문이다. 또한, 제 1 가상 경계 코드는, 그들 바이트들이 오직 마커로서 요구될 수도 있기 때문에, 제 2 부트 로더의 실행에서 제외될 수도 있다.

[0052] 추가의 양태는 예컨대, 무선 디바이스들에 대한 공중 경유 (OTA) 다운로드들을 통해 펌웨어 업데이트 프로 비저닝을 포함한다. 이와 관련하여, 관독-기록 메모리를 포함하는 다수의 디바이스들, 예컨대 플래시 메모 리를 포함하는 스마트폰들은 종종 펌웨어에 대한 업그레이드 또는 업데이트를 요구한다. 개시된 불량한 블 록 관리 설계는 진술된 기술들의 확장물을 사용하여 신뢰할만한/페일 세이프 방식으로 그러한 업데이트들을 인 에이블한다. 먼저, 시스템은 현재 부트 로더가 존재하는 블록들을 검출한다. 그 후에, 더 새로운/대체 부트 로더가 기록되는데 사용가능한 하나 이상의 추가 우수한 블록으로 전진한다. 그 후에, 새로운 부트 로더는 무엇인가 업그레이드/업데이트 프로세스를 차단하는 경우, 새로운 우수한 블록 위치를 백업으로서 사용하 여 프로그래밍될 수 있다. 그 후에, 이전 부트 로더는 블록 0 에서 시작하는 초기 위치에서 소거될 수도 있 고, 새로운 이차 부트 로더가 블록 0 및 필요한 경우 후속 우수한 블록들에 재프로그래밍된다.

[0053] 도 7 은 다양한 양태들에 따라 NAND 디바이스에서 이미지를 수정하는 방법의 일 예를 도시한다. 이러한 업 그레이드 방법은 진술된 가상 경계 코드를 사용하여 가상 블록 지정들과 완전히 호환가능하다. 초기 상태 (70) 에서, 시스템은 현재 부트 로더가 존재하는 제 1 블록들 내의 데이터에 액세스한다. 이들 블록들은 프 리램블 메모리 코드들 (10) 을 갖는 프리램블 (5), 및 이미지 헤더 (12) 를 포함할 수도 있다. 도시된 예에 서, 부트 로더는 NAND 디바이스에 의존하여, 가상 블록들 또는 실제 블록들에 대응할 수도 있는 2 개의 초기 블 록들에 배치된 2 개의 부분들 (15a, 15b) 로 분할되고, 본 명세서의 양태들에 따라 가상 경계 코드 (14) 를 포 함할 수도 있다. 대안적으로, 원래의 부트 로더는 종래 기술에 따른 단일의 연속하는 부트 로더일 수도 있 고, 이는 본 명세서의 방법들에 따라 대체될 수도 있다. 이러한 스테이지에서, 임의의 부트 로더 부분들 (15a', 15b'), 새로운 프리램블 메모리 코드들 (10') 을 갖는 새로운 프리램블 (5'), 새로운 이미지 헤더 (12'), 및 임의의 필수적인 새로운 VBC들 (14') 을 포함하는 새로운 데이터 이미지를 위해 얼마나 많은 블록들 이 요구되는지를 추가로 결정하기 위해, 시스템은 실제 블록들 (또는 존재한다면 가상 블록들) 의 사이즈를 결 정할 수도 있다. 시스템은 다음의 사용가능한 우수한 블록(들)을 발견하기 위해 플래시 메모리 디바이스에 서 전방향 스캔할 수도 있다. 1 초과 1의 우수한 블록이 새로운 부트 로더를 위해 요구된다면, 시스템은 새로 운 데이터 이미지를 수용하기 위해 충분한 사용가능한 우수한 블록들이 발견될 때까지, 플래시 메모리 디바이스 에서 전방향 스캔하는 것을 계속할 수도 있다. 72 에 도시된 상태에서, 그 후에 시스템은 부트 로더 부분들 (15a', 15b'), 새로운 프리램블 메모리 코드들 (10') 을 갖는 새로운 프리램블 (5'), 새로운 이미지 헤더 (12'), 및 새로운 VBC들 (14') 을 포함하는 새로운 데이터 이미지를 식별된 사용가능한 우수한 블록들에 저장한 다. 본 발명의 양태에 따라, 사용가능한 우수한 블록들은 바로 다음에 연속적으로 사용가능한 우수한 블록

들일 필요는 없다. 도시된 예에서, 이러한 방법의 부분으로서 변화되는 이전 블록과 새로운 블록 사이에 버퍼를 제공하기 위해, 적어도 2 개의 블록들이 스킵된다. 적어도 한 쌍의 블록들을 남기는 것은 또한, 그 방법이 현대 시스템들을 위해 요구되는 1 또는 2 초과의 초기 블록들을 차지할 수도 있는, 초기 부트 로더의 사이즈에 있어서 향후 증가를 설명하게 한다. 또한, 일부 공간을 남기는 것은 또한 디바이스의 초기 블록들에서 불량한 블록들을 회피할 수 있다. 이전의 부트 로더를 소거하기 전에, 새로운 데이터 이미지에 검증 체크가 수행될 수도 있다. 72 에 도시된 스테이지에서, 시스템은 새로운 데이터 이미지가 정확히 로딩되지 않았거나, 손상되거나, 사용 불가능한 경우에 파일 세이프로서 이전 부트 로더 부분들 (15a, 15b) 을 여전히 부트 오프할 것이다. 일단, 새로운 데이터 이미지가 로딩되면 (및 옵션으로 인증되면), 74 에 도시된 상태에서 이전 부트 로더, 또는 적어도 그 초기 부분은 소거될 수도 있다. 특히, 이전 프리앰블 (5), 이전 이미지 헤더 (12), 또는 부트 로더 (15a) 의 초기 부분 중 적어도 하나가 소거되면, 이전 부트 로더는 그들의 엘리먼트들을 찾고 있는 일차 부트 로더에 의해 스킵될 것이다. 74 에 도시된 제 2 블록이 소거된 것으로 도시되지는 않지만, 그 블록은 다음 단계로 진행하기 전에 소거될 수도 있다. 그 후에, 76 에 도시된 상태에서, 시스템은 그 후, 스테이지 72 에서 이전에 저장된 새로운 데이터 이미지와 사실상 동일한 새로운 데이터 이미지를 갖는 제 1 블록들을 저장한다. 따라서, 스테이지 76 에서, 새로운 데이터 이미지는 NAND 디바이스 상의 2 개의 장소들에서 발견될 수도 있으며, 제 1 블록들을 마킹 시 에러가 발생한 경우 백업으로서 기능한다. 초기 블록들에서 데이터 이미지가 인증되면, 수정 방법은 완료한다. 더 새로운 데이터 이미지의 제 2 카피가 NAND 디바이스에 백업으로서 남겨질 수도 있다. 대안적으로, 그 방법은 그 후 추가로 78 에 도시된 상태에서, 72 에 도시된 상태에서 마킹되었던, 우수한 블록들로부터의 새로운 데이터 이미지의 제 2 카피를 소거할 수도 있다.

[0054] 도 7 에서, 종속 데이터 이미지에 속하지 않는 다양한 블록들은 "소거됨" 으로 표시된다. 그러나, 그 블록들은 비어있거나 소거되지 않아야 한다. 본 명세서의 양태들에 따라, 방법들은 프리앰블 및 가상 경계 코드들을 사용하여 부트 로더의 사이즈 및 위치를 결정할 수도 있고, 따라서 "소거됨" 으로 표시된 블록들은 데이터의 블록들이 부트 로더 이미지에 속하지 않는 것을 표시하도록 의도된다.

[0055] 도 8 에서, 판독-기록 메모리로부터 데이터 이미지를 재판독하는 일 양태의 방법 (800) 을 도시한다. 블록 (810) 에서, 초기 데이터 이미지는 판독-기록 메모리의 적어도 하나의 제 1 가상 블록에서 액세스된다. 적어도 하나의 제 1 가상 블록은 1 초과의 제 1 가상 블록을 포함할 수도 있다. 블록 (820) 에서, 방법은 판독-기록 메모리에서 사용가능한 적어도 하나의 제 2 가상 블록에 대하여 적어도 하나의 제 1 가상 블록을 전방향으로 스캔한다. 따라서, 적어도 하나의 제 1 가상 블록이 1 초과의 가상 블록을 포함했다면, 방법 (800) 에서 정의된 것과 같은 "제 2 가상 블록" 은 선행하는 제 1 가상 블록들을 뒤따를 것이다. 블록 (830) 에서, 새로운 데이터 이미지의 제 1 부분은 제 2 가상 블록에 저장된다. 블록 (840) 에서, 방법은 제 3 가상 블록까지 제 2 가상 블록을 전방향으로 스캔한다. 제 3 가상 블록이 위치되면, 850 에서 새로운 데이터 이미지의 제 2 부분은 제 3 가상 블록에 저장된다. 블록 (860) 은 적어도 하나의 제 1 가상 블록의 적어도 일부분 내의 초기 데이터 이미지를 소거하는 것을 포함한다. 전체 초기 데이터 이미지 또는 그 바로 일부분이 블록 (860) 에서 소거될 수도 있다. 예를 들어, 블록 (860) 에서 초기 데이터 이미지는 1 초과의 제 1 가상 블록으로부터 소거될 수도 있다. 870 에서, VBC 를 포함하는 새로운 데이터 이미지 제 1 부분 및 제 2 부분은 적어도 하나의 제 1 가상 블록에 저장된다. 블록 (880) 에서, 새로운 데이터 이미지 제 1 부분은 제 2 가상 블록으로부터 소거된다. 블록 (890) 에서, 새로운 데이터 이미지 제 2 부분은 제 3 가상 블록으로부터 소거된다.

[0056] 프리앰블 블록은 다양한 디바이스 특징들을 유도하기 위해 사용될 수도 있다. 유효한 프리앰블 블록이 초기에 발견되지 않으면, 프로세스는 블록들을 횡단하여 유효한 프리앰블 블록을 찾는 것을 시도할 수도 있다. 프리앰블 블록은 플래시 판독 커맨드의 부분으로서 판독된 블록의 초기 바이트들 (예컨대, 제 1 의 12 바이트들) 에서, 가상 경계 코드와 같은 특정 코드에 대하여 체크섬으로써 검출될 수도 있다. 개시된 양태에 따라, 일반적인 NAND 디바이스 폭 핸들링 기술이 바람직할 수도 있다. 이와 같이 지정된 검출된 가상 블록들을 카운팅하기 위해 알고리즘이 사용될 수도 있다. 가상 블록 카운트가 이 기술을 사용하여 획득되지 않으면, 프로세스는 어떤 2차 부트 로더도 발견되지 않은 것을 결론으로 하여 종료할 수도 있다. 그렇지 않으면, 이러한 방식으로 가상 경계 코드는 가상 블록들과 NAND 디바이스 자체의 사이즈를 결정하는데 사용될 수도 있다. 또한, ECC 검출 및 페이지 사이즈 검출은 본 명세서에 개시된 방법들의 부분으로서 구현될 수도 있다. 예를 들어, 오토-페이지 사이즈 검출 알고리즘은 디바이스 상의 페이지들의 수 및 따라서 그 페이지 사이즈를 결정하기 위해 특정 가상 경계 코드로 마킹된 페이지들의 수를 판독할 수도 있다.

[0057] 추가로, 다양한 실시형태들은 임의의 다양한 모바일 컴퓨팅 디바이스에서 및/또는 이들과 함께 구현될 수도 있으며, 그 일 예는 도 9 에 셀룰러 전화의 형태로 도시된다. 통상적인 모바일 컴퓨팅 디바이스들 (900) 은 도 9 에 도시된 컴포넌트들을 공통적으로 가질 것이다. 예를 들어, 모바일 컴퓨팅 디바이스들 (900) 은 내부 메모리 (902) 에 커플링된 프로세서 (901), 및 저항 감지 터치스크린 (904), 정전용량 감지 터치스크린, 적외선 감지 터치스크린, 음향/압전 감지 터치스크린 등과 같은 터치 표면 입력 디바이스/디스플레이 (903) 를 포함할 수도 있다. 모바일 컴퓨팅 디바이스 (900) 는 무선 데이터 링크에 접속된 전자기 방사선을 전송하고 수신하기 위한 라디오/안테나 (906), 및/또는 프로세서 (901) 에 커플링된 셀룰러 전화 트랜시버 (920) 를 포함할 수도 있다. 모바일 컴퓨팅 디바이스들 (900) 은 또한, 그 디바이스의 위치를 결정하기 위해 프로세서 (901) 에 커플링된 GPS 수신기를 포함할 수도 있다. 모바일 컴퓨팅 디바이스들 (900) 은 또한 사용자 입력들을 수신하기 위한 물리적 버튼들 (908) 을 포함할 수도 있다.

[0058] 임의의 다양한 컴퓨팅 디바이스들에서 및/또는 이들과 함께 다양한 실시형태들이 구현될 수도 있으며, 그 일 예가 도 10 에 예시된다. 예를 들면, 모바일 디바이스 (1000) 는 내부 메모리들 (1004 및 1006) 에 커플링된 프로세서 (1002) 를 포함할 수도 있다. 내부 메모리들 (1004 및 1006) 은 휘발성 또는 비휘발성 메모리들일 수도 있으며, 또한 보안 및/또는 암호화된 메모리들, 또는 미-보안 및/또는 미-암호화된 메모리들, 또는 이들의 임의의 조합일 수도 있다. 프로세서 (1002) 는, 터치 스크린 디스플레이 (1016) (예컨대, 저항 감지 터치스크린, 정전용량 감지 터치 스크린, 적외선 감지 터치 스크린 등), 또는 종래의 버튼들 (예컨대, 1012a 및 1012b) 및 비-터치 스크린 디스플레이와 같은 사용자 인터페이스에 또한 커플링될 수도 있다. 추가로, 무선 디바이스 (1000) 는 프로세서 (1002) 가 하나 이상의 유선 또는 무선 네트워크들을 통해 다른 컴퓨팅 디바이스들과 통신할 수 있게 하도록 구성된 하나 이상의 네트워크 트랜시버들을 포함할 수도 있다. 특정 예로서, 모바일 디바이스 (1000) 의 네트워크 트랜시버들은, 프로세서 (1002) 에 커플링된 셀룰러 전화기 트랜시버 (1010) 및/또는 하나 이상의 무선 데이터 링크에 접속될 수도 있으며 전자기 방사를 전송 및 수신하기 위한 하나 이상의 안테나 (1008) 를 포함할 수도 있다. 무선 디바이스 (1000) 는 또한, 사용자 입력들을 수신하기 위한 물리적 버튼들 (1012a 및 1012b) 을 포함할 수도 있다.

[0059] 전술된 다양한 실시형태들은 또한, 도 11 에 도시된 것과 같은 랩탑 컴퓨터 (1100) 와 같은 다양한 퍼스널 컴퓨팅 디바이스들 내에서 및/또는 이들과 함께 구현될 수도 있다. 다수의 랩탑 컴퓨터들은 컴퓨터의 포인팅 디바이스로서 기능하는 터치 패드 터치 표면 (1107) 을 포함하며, 따라서 터치 스크린 디스플레이가 장착되고 전술된 것과 같은 모바일 컴퓨팅 디바이스들에서 구현되는 것과 유사한 드래그, 스크롤 및 플리크 (flick) 제스처들을 수신할 수도 있다. 랩탑 컴퓨터 (1100) 는 통상적으로 플래시 메모리 디바이스 (1102) 와 같은 대용량 비휘발성 메모리 및 휘발성 메모리에 커플링된 프로세서 (1101) 를 포함할 것이다. 랩탑 컴퓨터 (1100) 는 또한 프로세서 (1101) 에 커플링된 플로피 디스크 드라이브 및 콤팩트 디스크 (CD) 드라이브를 포함할 수도 있다. 랩탑 컴퓨터 (1100) 는 또한, 프로세서 (1101) 가 하나 이상의 유선 또는 무선 네트워크들을 통해 다른 컴퓨팅 디바이스들과 통신할 수 있게 하도록 구성된, 프로세서 (1101) 에 커플링된 다수의 네트워크 트랜시버들 또는 네트워크 커넥터 포트들을 포함할 수도 있다. 특정 예로서, 랩탑 컴퓨터 (1100) 의 네트워크 트랜시버들은 이더넷, USB 또는 FireWire 커넥터 소켓들/트랜시버들, 전자기 방사를 전송하고 수신하기 위한 하나 이상의 안테나에 커플링된 와이파이 및/또는 셀룰러 데이터 네트워크 트랜시버들과 같은 하나 이상의 무선 모듈 트랜시버들을 포함할 수도 있다. 랩탑 컴퓨터 (1100) 는 또한 프로세서 (1101) 를 향후 개발될 수도 있는 네트워크에 커플링하기 위한 다른 타입의 네트워크 접속 회로들을 포함할 수도 있다. 노트북 구성에서, 컴퓨터 하우징은 터치 패드 터치 표면 (1107), 키보드 (1108), 및 디스플레이 (1109) 를 포함하며, 이들 모두는 프로세서 (1101) 에 커플링된다. 컴퓨팅 디바이스의 다른 구성들은, 잘 알려져 있는 것과 같이 (예컨대, USB 입력을 통해) 프로세서에 커플링된 컴퓨터 마우스 또는 트랙볼을 포함할 수도 있으며, 이들은 또한 다양한 실시형태들과 결합하여 사용될 수도 있다.

[0060] 본 명세서에 설명된 다양한 실시형태들에서 프로세서들은, 전술된 다양한 실시형태들의 기능들을 포함하여 다양한 기능들을 수행하기 위해 소프트웨어 명령들 (애플리케이션들) 에 의해 구성될 수 있는, 임의의 프로그래밍 가능 마이크로프로세서, 마이크로컴퓨터 또는 다중 프로세서 칩 또는 칩들일 수도 있다. 일부 디바이스들에 있어서, 무선 통신 기능들에 전용된 하나의 프로세서 및 다른 애플리케이션들을 구동하는 것에 전용된 하나의 프로세서와 같은 다중 프로세서들이 제공될 수도 있다. 통상적으로, 소프트웨어 애플리케이션들은 액세스되어 프로세서들로 로딩되기 전에 내부 메모리에 저장될 수도 있다. 프로세서들은 애플리케이션 소프트웨어 명령들을 저장하기에 충분한 내부 메모리를 포함할 수도 있다. 다수의 디바이스들에 있어서, 내부 메모리는 휘발성 메모리 또는 플래시 메모리와 같은 비휘발성 메모리, 또는 이들 양자의 혼합일 수도 있다. 이 설명

의 목적들을 위해, 메모리에 대한 일반적인 참조는, 내부 메모리 또는 디바이스에 플러징되는 착탈가능 메모리 및 프로세서 자체 내의 메모리를 포함하여 프로세서들에 의해 액세스가능한 메모리를 지칭한다.

[0061] 앞서 언급한 방법 설명들 및 프로세스 흐름 다이어그램들은 단지 예시적인 예들로서 제공되고, 다양한 실시예들의 단계들이 제시된 순서로 수행되어야 함을 요구하거나 암시하려고 의도되지는 않는다. 당업자에 의해 인식되는 것과 같이, 전술한 실시형태들에서의 블록들의 순서는 임의의 순서로 수행될 수도 있다.

[0062] "그 후에", "그리고 나서", "다음으로" 등과 같은 단어들은 단계들의 순서를 제한하려고 의도되지 않으며; 이러한 단어들은 단지 방법들의 설명을 통해 독자를 가이드하는데 이용된다. 추가로, 예를 들어, 관사들 ("a," "an" 또는 "the") 을 사용하여 단수로의 청구항 엘리먼트들에 대한 임의의 참조는 그 엘리먼트를 단수로 한정하는 것으로서 해석되지는 않는다.

[0063] 본 명세서에 개시된 실시형태들과 연계하여 설명된 다양한 예시적인 논리 블록들 및 프로세스 흐름도 블록들은 전자 하드웨어, 컴퓨터 소프트웨어, 또는 이들 양자의 조합들로서 구현될 수도 있다. 하드웨어와 소프트웨어의 이러한 상호교환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들, 및 단계들이 그들의 기능성에 대해 전반적으로 전술되었다. 그러한 기능이 하드웨어로 구현되는지 또는 소프트웨어로 구현되는지의 여부는 특정 애플리케이션 및 전체 시스템에 부과되는 설계 제약들에 따라 달라진다. 당업자라면, 상기 상술한 기능성을 각각의 특정 애플리케이션에 대해 다양한 방식으로 구현할 수도 있지만, 이러한 구현 결정은 본 발명의 범위를 벗어나게 하는 것으로 이해되어서는 안된다.

[0064] 본원에서 개시된 실시형태들과 연계하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 및 회로들은 본 범용 프로세서, 디지털 신호 프로세서 (digital signal processor; DSP), 주문형 반도체 (application specific integrated circuit; ASIC), 필드 프로그래머블 게이트 어레이 (field programmable gate array; FPGA) 또는 다른 프로그래머블 로직 디바이스, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트들, 또는 본 명세서에서 개시된 기능들을 수행하도록 설계된 것들의 임의의 조합에 의해 구현되거나 수행될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 대안에서, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예를 들면, DSP와 마이크로프로세서의 조합, 복수의 마이크로프로세서들의 조합, DSP 코어와 연계한 하나 이상의 마이크로프로세서들의 조합, 또는 임의의 다른 그러한 구성으로서 구현될 수도 있다. 대안으로, 일부 단계들 또는 방법들은 주어진 기능에 특정한 회로부로 수행될 수도 있다.

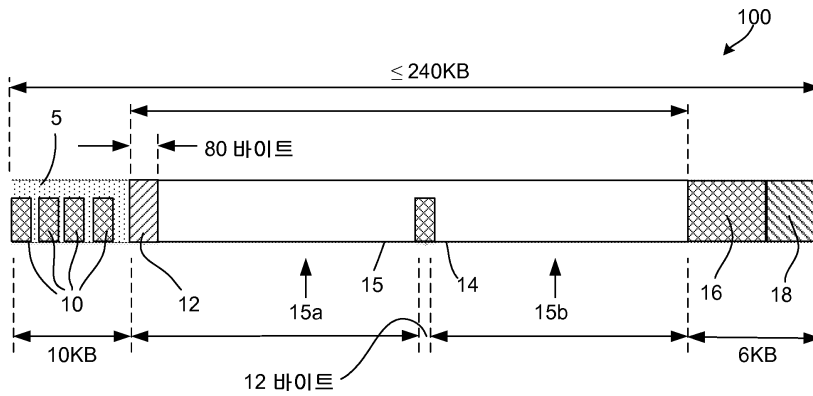
[0065] 하나 이상의 예시적인 양태들에서, 상술된 기능들은 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 비-일시적인 컴퓨터 판독가능 저장 매체 또는 비-일시적인 프로세서 판독가능 저장 매체에 저장될 수도 있다. 본 명세서에서 개시된 방법 또는 알고리즘의 단계들은, 비-일시적인 컴퓨터 판독가능 또는 프로세서 판독가능 저장 매체에 상주할 수도 있는 프로세서 실행가능 소프트웨어 모듈에서 구현될 수도 있다. 비-일시적 컴퓨터 판독가능 또는 프로세서 판독가능 저장 매체는 컴퓨터 또는 프로세서에 의해 액세스될 수도 있는 임의의 저장 매체일 수도 있다. 한정이 아닌 예로서, 그러한 비-일시적인 컴퓨터 판독가능 또는 프로세서 판독가능 매체는 RAM, ROM, EEPROM, 플래시 메모리, CD-ROM 또는 다른 광학 디스크 저장부, 자기 디스크 저장부 또는 다른 자기 저장 디바이스들, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 이용될 수도 있고 컴퓨터에 의해 액세스될 수도 있는 임의의 다른 매체를 포함할 수도 있다. 본 명세서에서 사용된 바와 같이, 디스크 (disk) 및 디스크 (disc) 는, 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피디스크 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 는 통상 자기적으로 데이터를 재생하고, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 상기의 조합들이 또한, 비-일시적인 컴퓨터 판독가능 및 프로세서 판독가능 매체의 범위 내에 포함된다. 추가로, 방법 또는 알고리즘의 동작들은 코드들 및/또는 명령들의 하나 또는 임의의 조합 또는 세트로서 비-일시적인 프로세서 판독가능 매체 및/또는 컴퓨터 판독가능 매체 상에 상주할 수도 있고, 이들은 컴퓨터 프로그램 제품에 통합될 수도 있다.

[0066] 개시된 실시형태들의 양태들의 가능한 변형들 및 조합들이 사용될 수도 있으면서, 여전히 동일한 기본적인 근본 메커니즘 및 방법들을 채용하는 것을 당업자는 인식할 것이다. 설명을 위해, 전술한 설명은 특정 실시형태들을 참조하여 기록되었다. 하지만, 상기 예시적인 논의들은, 개시된 정확한 형태들로 본 개시물을 한정하거나 포괄하는 것으로 의도되지 않는다. 다수의 변형들 및 변경들이 상기 교시들과 관련하여 가능하다. 실시형태들은 본 개시물의 원리들 및 그 실제 애플리케이션들을 설명하고, 당업자로 하여금 본 개시물, 및 고려된 특정 사용에 적합한 다양한 변형들을 갖는 다양한 실시형태들을 가장 잘 활용할 수 있도록 하기 위해 선택

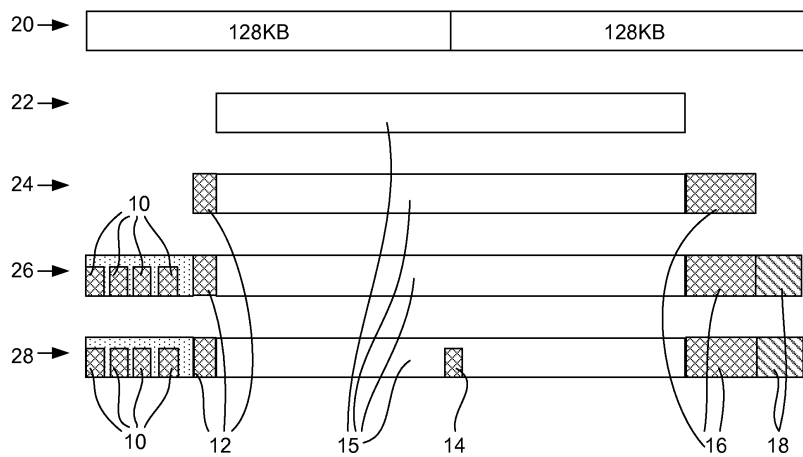
및 설명되었다. 따라서, 본 개시물은 본 명세서에 도시되고 설명된 개시된 기술들의 개별 양태들 및 실시형태들로 제한되도록 의도된 것은 아니지만, 본 명세서에 개시된 하기의 청구항들 및 원칙들 및 신규의 특징들과 일치하는 최광의 범위에 따를 것이다.

도면

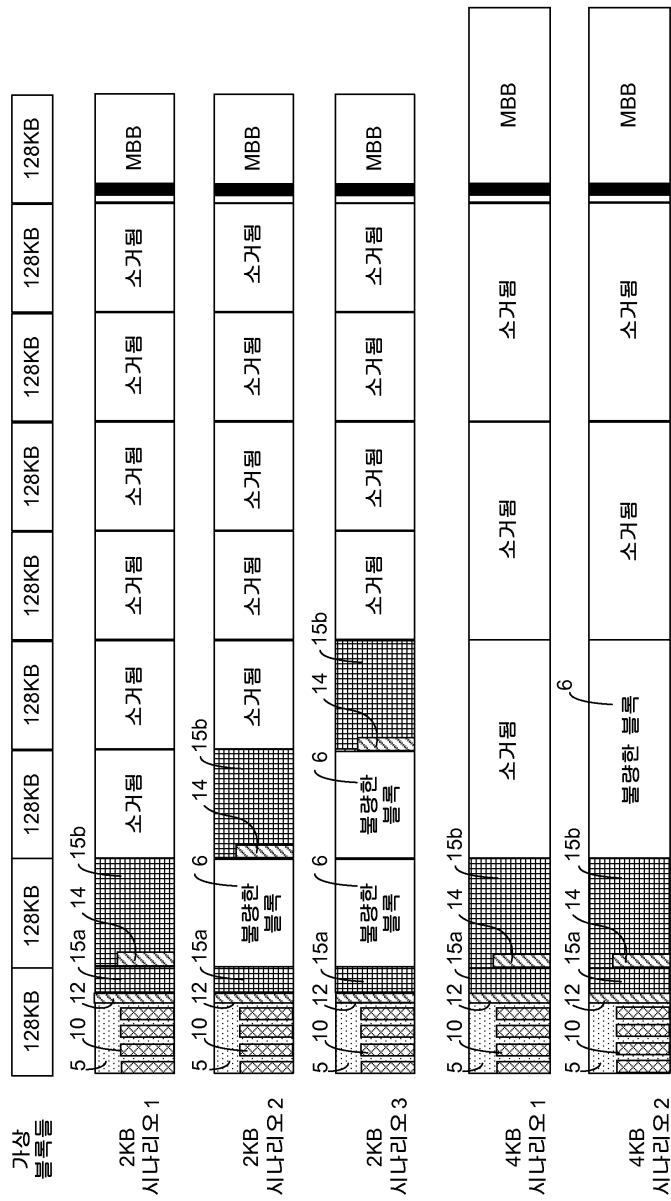
도면1



도면2

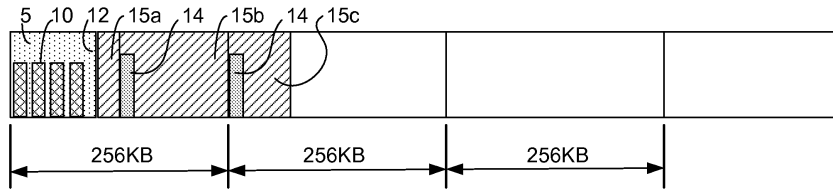


도면3

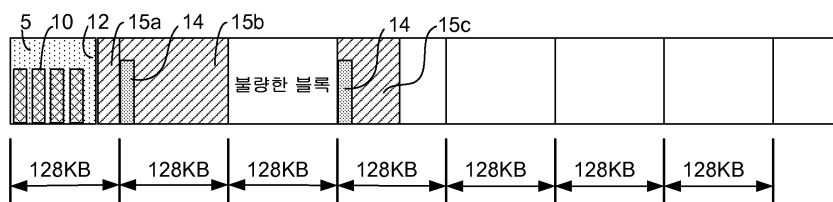


도면4

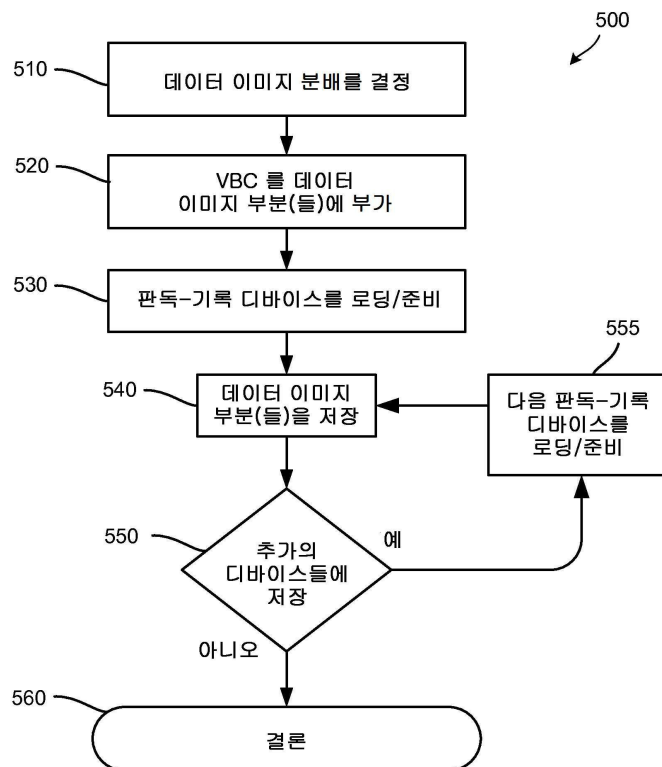
4KB
시나리오 3



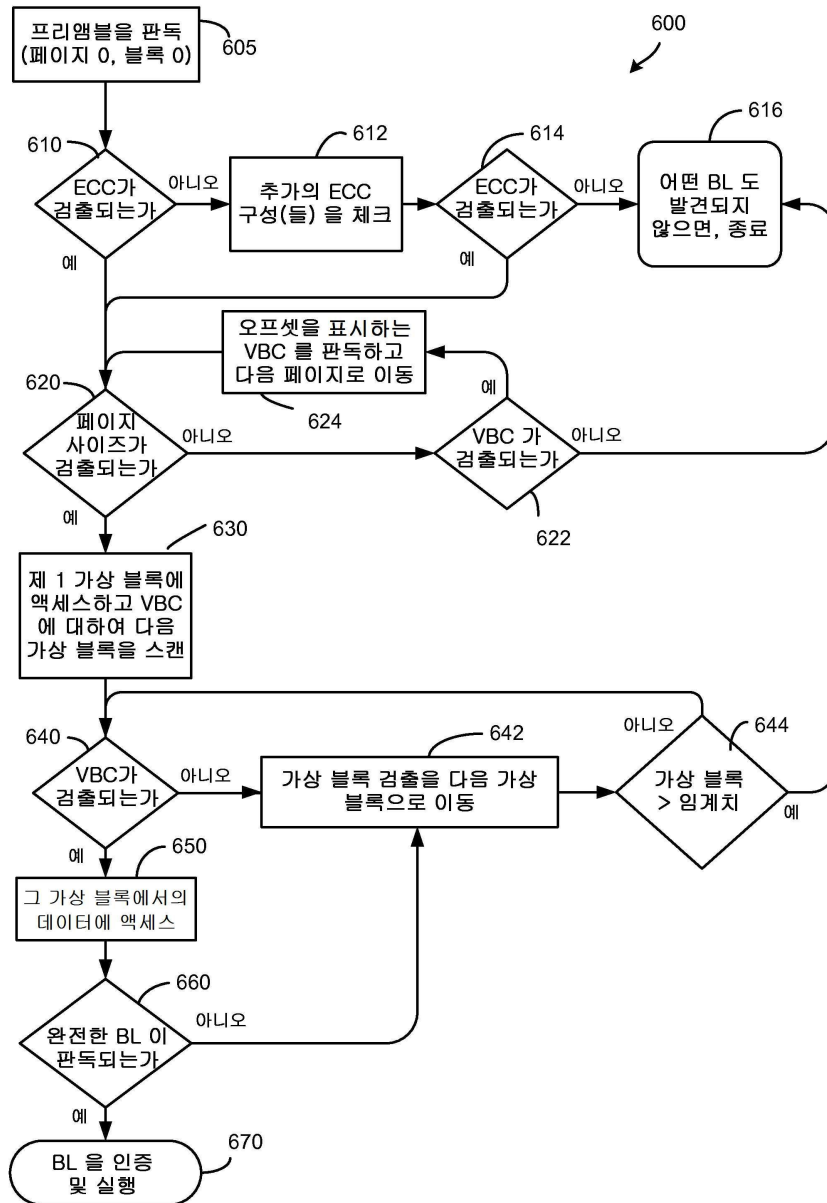
2KB
시나리오 4



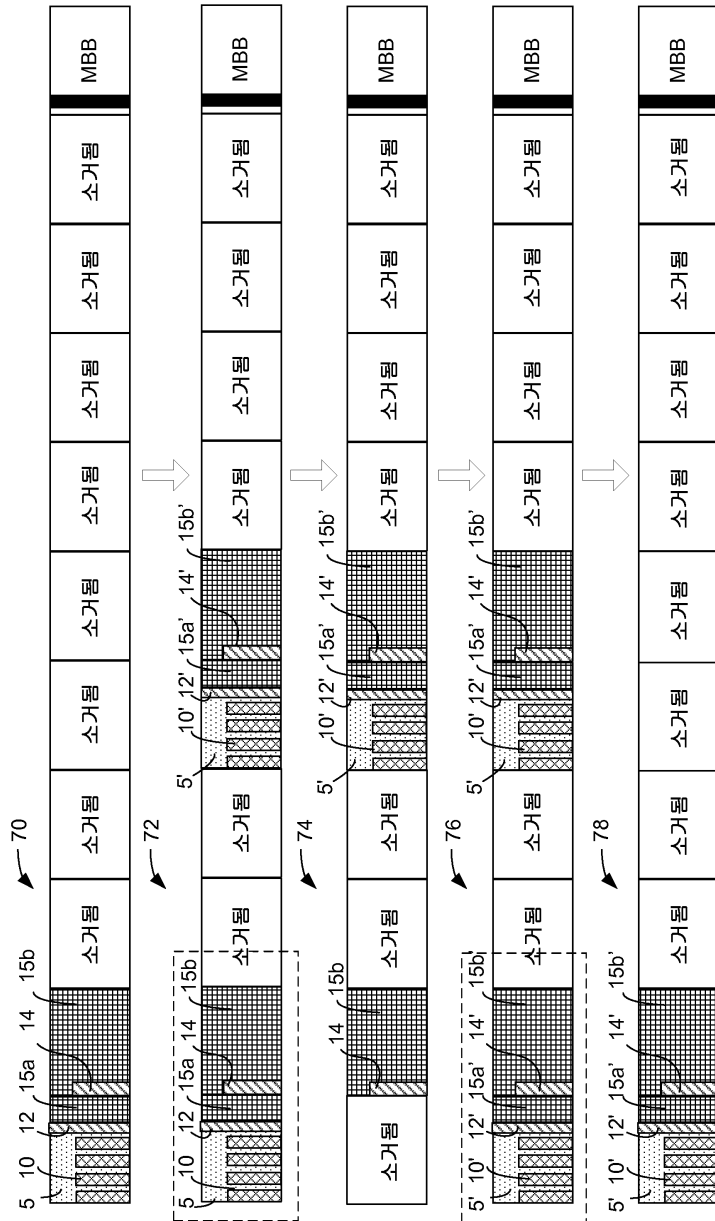
도면5



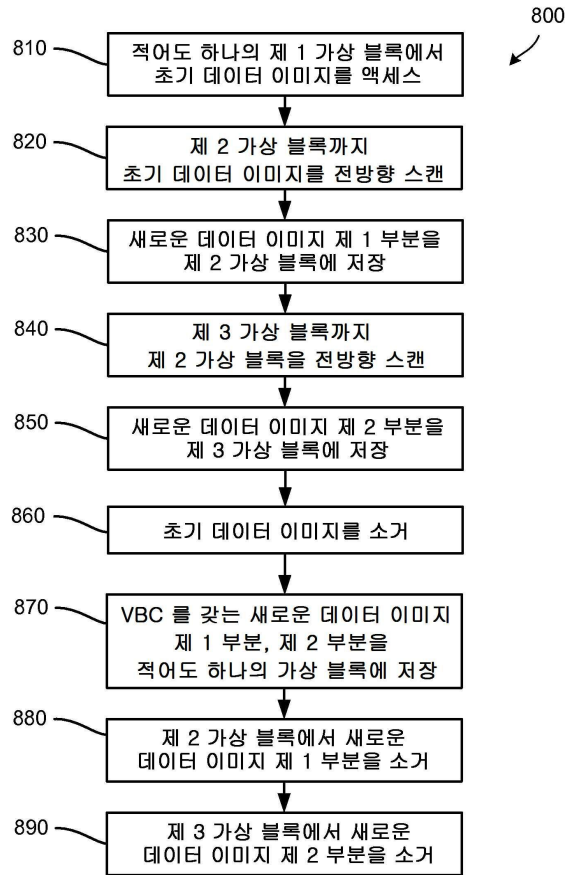
도면6



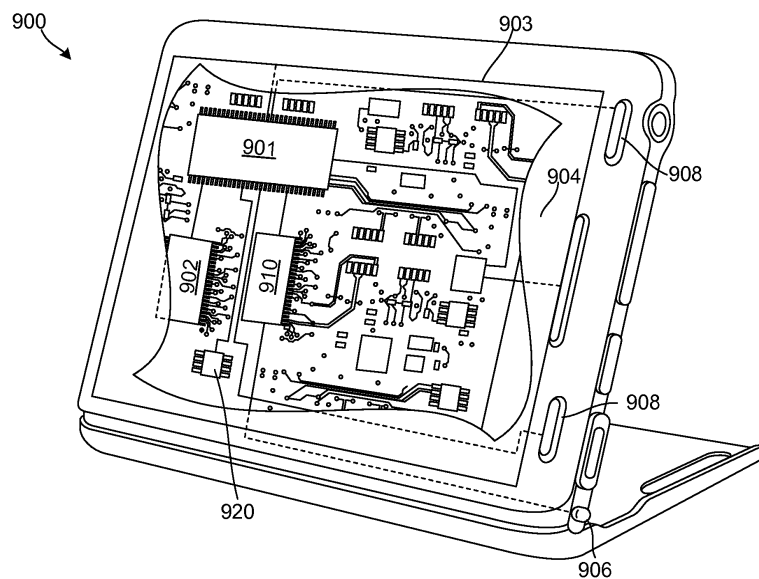
도면7



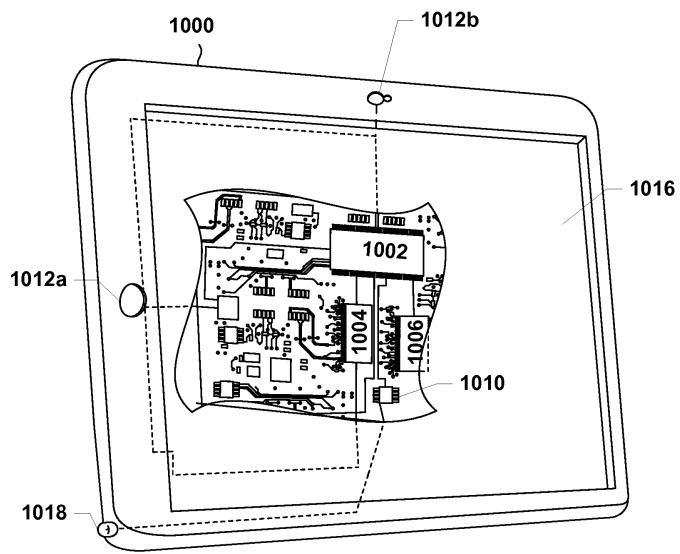
도면8



도면9



도면10



도면11

