



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0142103
(43) 공개일자 2013년12월27일

(51) 국제특허분류(Int. Cl.)
G06F 12/02 (2006.01)
(21) 출원번호 10-2013-7002426
(22) 출원일자(국제) 2011년06월23일
심사청구일자 없음
(85) 번역문제출일자 2013년01월29일
(86) 국제출원번호 PCT/US2011/041647
(87) 국제공개번호 WO 2012/012085
국제공개일자 2012년01월26일
(30) 우선권주장
12/828,241 2010년06월30일 미국(US)

(71) 출원인
샌디스크 테크놀로지스, 인코포레이티드
미국 텍사스 75024 플라노 노스 달라스 파크웨이
6900 투 리가시 타운 센터
(72) 발명자
우, 윌리엄
미국, 캘리포니아 95014, 쿠퍼티노, 이스트 이스
테이츠 드라이브 877
트라이스터, 샤이
미국, 캘리포니아 94087, 써니배일, 털 드라이브
1454
(뒷면에 계속)
(74) 대리인
박경재

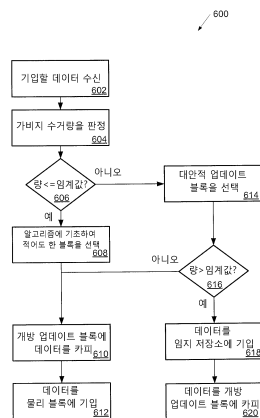
전체 청구항 수 : 총 24 항

(54) 발명의 명칭 **메모리 블록들의 우선적 가비지 수거**

(57) 요약

방법 및 시스템은 메모리 장치에서 업데이트 블록들에 대해 강제된 량의 가비지 수거 동작들을 우선적으로 수행한다. 어떤 데이터 기입에 의해 필요로 되는 가비지 수거량이 모니터되고 필요하다면 강제된 량에 부합되게 조절된다. 업데이트 블록들은 이들의 최근의 사용 또는 요구되는 가비지 수거량에 기초하여 선택될 수 있다. 또 다른 방법 및 시스템은 더 많은 수의 업데이트 블록들이 이용되도록 업데이트 블록들에 관한 제어 정보를 임시 저장 영역에 저장할 수 있다. 속도 클래스 테스트에 의해 측정된 순차 기입 수행은 이 방법 및 시스템을 사용함으로써 최적화될 수 있다.

대표도 - 도6



(72) 발명자

후양, 지안민

미국, 캘리포니아 94085, 썬니베일, 몬타라 테라스
679

허치슨, 네일, 데이비드

미국, 캘리포니아 95008, 캠벨, 트와일라 코트
2474

스프라우스, 스티븐

미국, 캘리포니아 95129, 산 호세, 호넷 드라이브
5285

특허청구의 범위

청구항 1

메모리 장치에서 가비지 수거(garbage collection) 방법에 있어서, 상기 메모리 장치 내 제어기를 사용하여:

논리 어드레스가능 유닛에 기입할 인입 데이터를 수신하는 단계;

가비지 수거 동작을 위해 수행될 가비지 수거의 량을 판정하는 단계;

상기 판정된 가비지 수거량이 소정의 임계값 미만이면, 상기 소정의 임계값을 완수하기 위해 상기 가비지 수거 동작을 개시하는 단계; 및

상기 논리 어드레스가능 유닛에 대응하는 적어도 한 물리 메타블록에 상기 인입 데이터를 기입하는 단계를 포함하는, 방법.

청구항 2

제 1 항에 있어서, 상기 가비지 수거 동작을 개시하는 단계는

상기 가비지 수거 동작을 위해 최근에 최소로 사용된 업데이트 블록을 선택하는 단계; 및

온전한 블록으로부터 상기 소정의 임계값과 동일한 데이터량을 상기 최근에 최소로 사용된 업데이트 블록에 카피하는 단계로서, 상기 소정의 임계값은 상기 판정된 가비지 수거량을 포함하는 것인 단계를 포함하는, 방법.

청구항 3

제 1 항에 있어서, 상기 논리 어드레스가능 유닛의 크기는 4 메가바이트이고;

상기 물리 메타블록의 크기는 3 메가바이트이고;

상기 소정의 임계값은 2 메가바이트를 포함하는, 방법.

청구항 4

제 1 항에 있어서, 상기 가비지 수거 동작을 개시하는 단계는

상기 가비지 수거 동작을 위해 제 1의 최근에 최소로 사용된 업데이트 블록을 선택하는 단계;

상기 제 1의 최근에 최소로 사용된 업데이트 블록으로부터 제 1의 데이터량을 제 1 개방 업데이트 블록에 카피하는 단계로서, 상기 제 1의 데이터량은 상기 소정의 임계값 미만인, 단계;

상기 가비지 수거 동작을 위해 제 2의 최근에 최소로 사용된 업데이트 블록을 선택하는 단계; 및

또 다른 온전한 블록으로부터 제 2의 데이터량을 상기 제 2의 최근에 최소로 사용된 업데이트 블록에 카피하는 단계로서, 상기 제 2의 데이터량은 상기 소정의 임계값에서 감한 상기 제 1의 데이터량과 동일한 것인 단계를 포함하고;

상기 소정의 임계값은 상기 판정된 가비지 수거량을 포함하는, 방법.

청구항 5

제 4 항에 있어서, 상기 논리 어드레스가능 유닛의 크기는 4 메가바이트이고;

상기 물리 메타블록의 크기는 3 메가바이트이고;

상기 제 1의 데이터량은 1 메가바이트를 포함하고;

상기 소정의 임계값은 2 메가바이트를 포함하는, 방법.

청구항 6

제 1 항에 있어서, 상기 메모리 장치 내 제어기를 사용하여:

상기 가비지 수거 동작을 위해 최근에 최소로 사용된 업데이트 블록을 선택하는 단계;

상기 최근에 최소로 사용된 업데이트 블록에 대한 상기 판정된 가비지 수거량이 상기 소정의 임계값보다 많으면; 상기 선택된 최근에 최소로 사용된 업데이트 블록; 대신 상기 가비지 수거 동작을 위해 대안적 업데이트 블록을 선택하는 단계로서,

(A) 상기 대안적 업데이트 블록에 대한 상기 판정된 가비지 수거량이 상기 소정의 임계값보다 많으면, (1) 상기 인입 데이터를 임시 저장 공간에 기입하고; (2) 온전한 블록으로부터 상기 소정의 임계값과 동일한 제 1의 데이터량을 상기 대안적 업데이트 블록에 카피하며;

(B) 상기 대안적 업데이트 블록에 대한 상기 판정된 가비지 수거량이 상기 소정의 임계값 미만이면, 상기 대안적 업데이트 블록으로부터 상기 소정의 임계값 미만인 제 2의 데이터량을 개방 업데이트 블록에 카피하는 단계를 더 포함하는, 방법.

청구항 7

제 6 항에 있어서, 상기 논리 어드레스가능 유닛의 크기는 4 메가바이트이고;

상기 물리 메타블록의 크기는 3 메가바이트이고;

상기 소정의 임계값은 2 메가바이트를 포함하는, 방법.

청구항 8

제 6 항에 있어서, 상기 임시 저장 공간은 바이너리 캐시를 포함하는, 방법.

청구항 9

제 6 항에 있어서, 상기 메모리 장치 내 제어기를 사용하여: 상기 임시 저장 공간으로부터 상기 인입 데이터를 개방 업데이트 블록에 카피하는 단계를 더 포함하는, 방법.

청구항 10

메모리 장치의 수행을 최적화하는 방법에 있어서, 상기 메모리 장치 내 제어기를 사용하여:

논리 어드레스가능 유닛에 기입할 인입 데이터를 수신하는 단계;

제 1 업데이트 블록 및 제 2 업데이트 블록이 제 1 상태 또는 제 2 상태에 있는지를 판정하는 단계;

상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록이 상기 제 1 상태에 있다면: 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록이 상기 제 2 상태에 놓여지게 상기 인입 데이터를 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록에 기입하는 단계; 및

상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록이 상기 제 2 상태에 있다면: 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록이 상기 제 1 상태에 놓여지게 상기 인입 데이터를 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록에 기입하는 단계를 포함하는, 방법.

청구항 11

제 10 항에 있어서, 상기 논리 어드레스가능 유닛의 크기는 4 메가바이트이고;

상기 물리 메타블록의 크기는 3 메가바이트이고;

상기 소정의 임계값은 2 메가바이트를 포함하며;

상기 제 1 상태는

데이터를 포함하지 않는 상기 제 1 업데이트 블록; 및

상기 제 2 업데이트 블록의 논리 시작부분에 1 메가바이트의 데이터를 포함하는 상기 제 2 업데이트 블록을 포함하고,

상기 제 2 상태는

상기 제 1 업데이트 블록의 논리 끝부분에 2 메가바이트의 데이터를 포함하는 상기 제 1 업데이트 블록; 및
 상기 제 2 업데이트 블록의 상기 논리 시작부분에 2 메가바이트의 데이터를 포함하는 상기 제 1 업데이트 블록
 을 포함하는, 방법.

청구항 12

제 11 항에 있어서, 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록이 상기 제 2 상태에 놓여지게 상기
 인입 데이터를 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록에 기입하는 단계는
 상기 제 2 업데이트 블록으로부터 상기 소정의 임계값을 완수하기 위해 가비지 수거 동작을 개시하는 단계; 및
 상기 인입 데이터를 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록에 기입하는 단계를 포함하는, 방법.

청구항 13

제 11 항에 있어서, 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록이 상기 제 1 상태에 놓여지게 상기
 인입 데이터를 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록에 기입하는 단계는
 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록으로부터 상기 소정의 임계값을 완수하기 위해 가비지 수
 거 동작을 개시하는 단계; 및
 상기 인입 데이터를 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록에 기입하는 단계를 포함하는, 방법.

청구항 14

제 10 항에 있어서, 상기 메모리 장치 내 제어기를 사용하여:
 (1) 가비지 수거 동작을 위해 최근에 최소로 사용된 업데이트 블록을 선택하는 단계;
 (2) 상기 최근에 최소로 사용된 업데이트 블록에 대한 상기 판정된 가비지 수거량이 소정의 임계값보다 많으면;
 상기 선택된 최근에 최소로 사용된 업데이트 블록 대신 상기 가비지 수거 동작을 위해 대안적 업데이트 블록을
 선택하는 단계로서, (A) 상기 대안적 업데이트 블록에 대한 상기 판정된 가비지 수거량이 상기 소정의 임계값보
 다 많으면, (i) 상기 인입 데이터를 임시 저장 공간에 기입하고; (ii) 상기 대안적 업데이트 블록으로부터 상기
 소정의 임계값과 동일한 제 1의 데이터량을 개방 업데이트 블록에 카피하며; (B) 상기 대안적 업데이트 블록에
 대한 상기 판정된 가비지 수거량이 상기 소정의 임계값 미만이면, 상기 대안적 업데이트 블록으로부터 상기 소
 정의 임계값 미만인 제 2의 데이터량을 개방 업데이트 블록에 카피하는 단계; 및
 (3) 상기 제 1 업데이트 블록 및 상기 제 2 업데이트 블록이 상기 제 1 상태 또는 상기 제 2 상태에 있을 때까
 지 단계(1) 및 단계(2)를 통해 반복하는 단계를 더 포함하는, 방법.

청구항 15

메모리 장치에 있어서,
 메모리; 및
 상기 메모리와 통신하는 제어기를 포함하고, 상기 제어기는
 논리 어드레스가능 유닛에 기입할 인입 데이터를 수신하고;
 가비지 수거 동작을 위해 수행될 가비지 수거의 량을 판정하고;
 상기 판정된 가비지 수거량이 소정의 임계값 미만이면, 상기 소정의 임계값을 완수하기 위해 상기 가비지 수거
 동작을 개시하고;
 상기 논리 어드레스가능 유닛에 대응하는 적어도 한 물리 메타블록에 상기 인입 데이터를 기입하게 구성된, 메
 모리 장치.

청구항 16

제 15 항에 있어서, 상기 제어기는 상기 가비지 수거 동작을 위해 최근에 최소로 사용된 업데이트 블록을 선택
 하고;

상기 판정된 량의 가비지 수거를 포함하는 상기 소정의 임계값과 동일한 데이터량을 온전한 블록으로부터 상기 최근에 최소로 사용된 업데이트 블록에 카피함으로써 상기 가비지 수거 동작을 개시하게 구성된, 메모리 장치.

청구항 17

제 15 항에 있어서, 상기 제어기는 상기 가비지 수거 동작을 위해 제 1의 최근에 최소로 사용된 업데이트 블록을 선택하고;

상기 소정의 임계값 미만인 제 1의 데이터량을 상기 제 1 최근에 최소로 사용된 업데이트 블록으로부터 제 1 개방 업데이트 블록에 카피하고;

상기 가비지 수거 동작을 위해 제 2의 최근에 최소로 사용된 업데이트 블록을 선택하고;

상기 판정된 가비지 수거량을 포함하는 상기 소정의 임계값에서 감한 상기 제 1의 데이터량과 동일한 제 2의 데이터량을 또 다른 온전한 블록으로부터 상기 제 2 최근에 최소로 사용된 업데이트 블록에 카피함으로써 상기 가비지 수거 동작을 개시하게 구성된, 메모리 장치.

청구항 18

메모리 장치에서 가비지 수거 동작을 수행할 때 동작 모드를 판정하는 방법에 있어서, 상기 메모리 장치 내 제 어기를 사용하여;

상기 가비지 수거 동작을 수행할 동작 모드를 판정하는 단계;

상기 가비지 수거 동작을 수행할 동작 모드를 판정한 것에 응하여 상기 판정된 모드에서 상기 가비지 수거 동작을 수행하는 단계를 포함하는, 방법.

청구항 19

제 18 항에 있어서, 상기 가비지 수거 동작을 수행하기 위한 동작 모드를 판정하는 단계는 적어도 한 표시자에 액세스하는 단계를 포함하는, 방법.

청구항 20

제 19 항에 있어서, 상기 적어도 한 표시자는

상기 메모리 장치로부터 데이터의 판독 기간 또는 상기 메모리 장치에 인입 데이터의 기입 기간을 포함하는 비지(busy) 기간에 수행될 가비지 수거의 제 1의 최대량; 및

논리 어드레스가능 유닛의 기입 기간 동안 가비지 수거 동작을 위해 수행될 가비지 수거의 제 2의 최대량을 포함하는, 방법.

청구항 21

제 20 항에 있어서, 상기 가비지 수거 동작을 수행할 동작 모드를 판정하는 단계는

상기 제 1 최대량을 상기 제 2 최대량과 비교하는 단계;

상기 제 1 최대량이 상기 제 2 최대량보다 크다면 상기 가비지 수거 동작을 수행하기 위해 상기 비지 기간을 판정하는 단계;

상기 제 2 최대량이 상기 제 1 최대량보다 크다면 상기 가비지 수거 동작을 수행하기 위해 상기 기입 기간을 판정하는 단계를 포함하는, 방법.

청구항 22

제 18 항에 있어서, 상기 가비지 수거 동작을 위해 수행될 가비지 수거량을 판정하는 단계를 더 포함하고;

상기 가비지 수거 동작을 수행하는 단계는 가비지 수거량이 상기 소정의 임계값 미만인 것으로 판정한 것에 응하여 상기 가비지 수거 동작을 상기 판정된 모드에서 개시하는 단계를 포함하는, 방법.

청구항 23

메모리 장치에 있어서,

메모리; 및

상기 메모리와 통신하는 제어기를 포함하고, 상기 제어기는

상기 가비지 수거 동작을 수행하기 위한 동작 모드를 판정하고;

상기 가비지 수거 동작을 수행할 동작 모드를 판정한 것에 응하여 상기 판정된 모드에서 상기 가비지 수거 동작을 수행하게 구성된, 메모리 장치.

청구항 24

제 23 항에 있어서, 상기 메모리는 적어도 한 표시자를 저장하고;

상기 제어기는 상기 적어도 한 표시자에 액세스함으로써 상기 가비지 수거 동작을 수행하기 위한 동작 모드를 판정하게 구성된, 메모리 장치.

명세서

기술분야

[0001] 관련출원에 대한 참조

[0002] 이 출원은 본원에 참조문헌으로서 포함시키는 2010년 6월 30일에 출원된 미국출원번호 12/828,241에 대한 우선권을 주장한다.

[0003] 이 출원은 일반적으로 메모리 장치들에 관한 것이다. 특히, 이 출원은 재프로그래밍가능한 비휘발성 반도체 플래시 메모리에서 블록들의 가비지 수거(garbage collection)에 관한 것이다.

배경기술

[0004] 플래시 메모리와 같은 비휘발성 메모리 시스템들은 소비자 제품들에서 사용하기 위해 널리 채택되었다. 플래시 메모리는 서로 다른 형태들, 예를 들면, 호스트 장치들 간에 옮겨질 수 있는 휴대 메모리 카드 형태 또는 호스트 장치 내 내장되는 고체상태 디스크(SSD)로서 발견될 수 있다. 데이터를 통상의 플래시 데이터 메모리 시스템에 기입할 때, 호스트는 전형적으로 데이터를 메모리 시스템의 논리 어드레스 공간 내 어드레스들에 기입하고 이들로부터 데이터를 판독한다. 이어서 메모리 시스템은 일반적으로 데이터를 논리 어드레스 공간과 메모리의 물리적 블록들 또는 메타블록들 간에 매핑하는데, 데이터는 논리 어드레스 공간 내 범위들에 대응하여 고정된 논리 그룹들로 저장된다. 일반적으로, 각각의 고정된 논리 그룹은 메모리 시스템의 별도의 물리 블록에 저장된다. 메모리 시스템은 논리 어드레스 공간이 물리 메모리에 어떻게 매핑되는가를 관리하지만 호스트는 이를 알지 못한다. 호스트는 논리 어드레스 공간 내에 자신의 데이터 파일들의 어드레스들을 관리하지만 메모리 시스템은 이 매핑을 알리지 않고 동작한다.

[0005] 논리 어드레스 공간에서 동작하는 메모리 시스템들의 결점은 단편화(fragmentation)이다. 호스트 파일 시스템에 의해 기입된 데이터는 흔히 논리 어드레스 공간에서 단편화될 수 있는데, 많은 고정된 논리 그룹들은 새로운 데이터로 단지 부분적으로만 업데이트된다. 단편화는 호스트 파일 시스템에 의해 자유 공간의 누적적 단편화의 결과로서, 그리고 아마도 심지어는 호스트 파일 시스템에 의해 개개의 파일들의 내재적 단편화의 결과로서도 발생할 수 있다. 이전에 기입된 데이터는 삭제에 기인하여 폐용(obsolete)이 되어 단편화를 더욱 야기할 수 있다. 단편화된 논리 그룹들은 다른 물리 블록에 전부 재기입될 필요가 있을 것이다. 단편화된 논리 그룹들을 재기입하는 것은 관계없는 데이터를 논리 그룹의 이전의 위치로부터 카피하는 것을 수반하는데, 이 프로세스는 가비지 수거로서 알려져 있다. 단편화가 증가함에 따라, 메모리 시스템의 전송 속도는 감소할 수 있다.

[0006] 메모리 시스템 메트릭들의 측정을 제공하기 위해서, 메모리 시스템의 기입 수행을 측정하기 위해 메모리 시스템에 테스트들이 적용될 수 있다. 예를 들면, 메모리 시스템들은 예를 들면 SD 협회로부터의 속도 클래스 명세와 같은 프로토콜들에 의해 지정될 수 있는 전송 속도 명세를 가질 수 있다. 속도 클래스 명세는 각각 2, 4, 및 6 MB/s의 메모리 시스템들에 대한 보장된 최소 전송 속도들을 나타내는 속도 클래스들 2, 4, 및 6을 포함한다. 메모리 시스템들 상에서 속도 클래스 테스트 수행은 이를테면 어드레스가능 유닛의 순차 기입 수행(Pw), 파일 할당 테이블(FAT) 기입을 위한 시간(Tfw), 및 복수의 FAT 기입들을 위한 시간의 롤링 평균(rolling average)(Tfw(avg))과 같은 메트릭들을 측정한다. 또한, 속도 클래스 테스트 수행은 단편화를 가진 메모리 시스

템의 수행(Pc)을 측정할 수 있고 이것은 대응하는 관독 수행 측정들을 포함한다. 속도 클래스 테스트 동안 일어나는 가비지 수거 활동은 테스트 결과들에 악영향을 미칠 수 있다.

발명의 내용

해결하려는 과제

[0007] 위에 언급된 문제들을 해결하기 위해서, 메모리 장치에서 가비지 수거를 제어하기 위한 방법 및 시스템이 개시된다.

과제의 해결 수단

[0008] 발명의 제 1 면에 따라서, 제어기를 포함하는 메모리 장치의 수행을 최적화하기 위한 방법이 개시된다. 방법은 논리 어드레스가능 유닛에 기입하기 위한 인입 데이터를 수신하는 단계; 가비지 수거 동작을 위해 수행될 가비지 수거의 량을 판정하는 단계; 판정된 가비지 수거량이 소정의 임계값 미만이면, 소정의 임계값을 완수하기 위해 가비지 수거 동작을 개시하는 단계를 포함한다. 또한, 방법은 논리 어드레스가능 유닛에 대응하는 적어도 한 물리 메타블록에 인입 데이터를 기입하는 단계를 포함한다. 이에 따라, 제어기는 가비지 수거가 현재 필요하지 않을지라도, 가비지 수거를 우선적으로 수행할 수 있다. 또한, 복수의 메모리 블록들 및 이 방법을 구현하는 메모리 제어기를 포함하는 시스템이 개시된다.

[0009] 또 다른 측면에 따라서, 가비지 수거 동작을 수행할 동작 모드를 판정하고; 가비지 수거 동작을 수행하고 판정된 모드에서 가비지 수거 동작을 수행하는 제어기를 포함하는 메모리 장치에서 가비지 수거 동작을 수행할 때 동작 모드를 판정하는 방법이 개시된다. 메모리 장치들은 서로 다른 동작 모드들을 가질 수 있다. 메모리 장치들은 다른 모드에서보다 한 모드에서 더 효율적으로 가비지 수거를 수행할 수 있다. 제어기는 가비지 수거 수행의 적어도 한 표시자(이를테면 동작 모드에서 가비지 수거의 최대량)를 분석함으로써 어느 동작 모드가 가비지 수거를 더 효율적으로 수행하는지를 분석할 수 있다. 분석에 따라, 제어기는 판정된 모드에서 가비지 수거를 수행할 수 있다. 예를 들면, 제어기는 가비지 수거를 효율적으로 수행하기 위해 판정된 모드에서 우선적으로 가비지 수거를 할 수 있다. 또한, 복수의 메모리 블록들 및 이 방법을 구현하는 메모리 제어기를 포함하는 시스템이 개시된다.

발명의 효과

[0010] 상술한 바와 같이 본 발명에 따르면, 메모리 장치에서 가비지 수거를 제어하기 위한 방법 및 시스템을 제공할 수 있다.

도면의 간단한 설명

- [0011] 도 1은 비휘발성 메모리를 갖는 메모리 시스템에 연결된 호스트의 블록도이다.
- 도 2는 도 1의 시스템의 물리 메모리 구성의 예를 도시한 것이다.
- 도 3은 도 2의 물리 메모리의 한 부분의 확대도이다.
- 도 4는 논리 어드레스가능 유닛들과 물리 메타블록들과의 정렬을 도시한 것이다.
- 도 5는 메모리 시스템에 데이터의 기입 후에 물리 메타블록들의 가능한 상태들을 도시한 것이다.
- 도 6은 메모리 블록들의 우선적 가비지 수거를 수반하는 메모리 장치의 수행을 최적화하는 방법을 도시한 흐름도이다.
- 도 7은 도 6에 방법에 따라 수행의 최적화가 행해진 메모리 블록들의 제 1 예를 도시한 것이다.
- 도 8은 도 6에 방법에 따라 수행의 최적화가 행해진 메모리 블록들의 제 1 예를 도시한 것이다.
- 도 9는 도 6에 방법에 따라 수행의 최적화가 행해진 메모리 블록들의 제 3 예를 도시한 것이다.
- 도 10은 도 6에 방법에 따라 수행의 최적화가 행해진 메모리 블록들의 제 4 예를 도시한 것이다.
- 도 11은 도 12의 상태머신을 포함하는 메모리 장치의 수행을 최적화하는 대안적 방법을 도시한 흐름도이다.
- 도 12는 메모리 장치의 수행을 최적화하는 도 11의 대안적 방법에 관계된 상태머신을 도시한 것이다.

도 13은 랜덤 액세스 메모리에 제어 정보의 카피를 수반하는 메모리 장치의 수행을 최적화하는 방법을 도시한 흐름도이다.

도 14는 가비지 수거의 량을 모니터하는 것을 수반하는 메모리 장치의 수행을 최적화하는 방법을 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 발명의 특징들을 구현하는데 사용하기에 적합한 플래시 메모리 시스템이 도 1 내지 도 3에 도시되었다. 도 1의 호스트 시스템(100)은 데이터를 메모리 시스템(102)에 저장하며 데이터를 플래시 메모리(102)로부터 인출한다. 플래시 메모리는 이를테면 개인용 컴퓨터에 설치된 고체상태 디스크(SSD) 드라이브 형태로 호스트 내에 내장될 수 있다. 대안적으로, 메모리(102)는 도 1에 도시된 바와 같이 기계 전기식 콘넥터의 결합부(mating part)들(104, 106)을 통해 호스트에 착탈가능하게 연결되는 카드 형태일 수도 있다. 내부 또는 내장형 SSD 드라이브로서 사용하게 구성되는 플래시 메모리는 도 1의 개요도와 유사하게 보일 수 있는데, 주 차이는 메모리 시스템(102)이 호스트 내부에 위치되어 있다는 것이다. SSD 드라이브들은 회전 자기 디스크 드라이브들의 대체용(drop-in replacements)인 개별적 모듈들의 형태일 수도 있다.

[0013] 시판되는 SSD 드라이브의 일례는 샌디스크 사에 의해 제작된 32 기가바이트 SSD이다. 시판되는 착탈가능한 플래시 메모리 카드들의 예들은 콤팩트플래시(CF), 멀티미디어카드(MMC), 보안 디지털(SD), 미니SD, 메모리 스틱, 스마트미디어, 트랜스플래시, 및 마이크로SD 카드들을 포함한다. 이들 카드들 각각이 이의 표준화된 명세들에 따른 고유한 기계 및/또는 전기적 인터페이스를 갖고 있을지라도, 각각에 포함된 플래시 메모리 시스템은 서로 유사하다. 이들 카드들은 모두가 본원의 양수인인 샌디스크 사로부터 입수될 수 있다. 샌디스크는 또한 호스트의 USB 리셉터클에 끼워넣음으로써 호스트에 연결하기 위한 USB(Universal Serial Bus) 플러그를 갖춘 소형 패키지들로 된 휴대 메모리 시스템들인 Cruzer 등록상표 하의 플래시 드라이브 제품군들을 제공한다. 이들 메모리 카드들 및 플래시 드라이브들 각각은 호스트와 인터페이스하고 이들 내에 플래시 메모리의 동작을 제어하는 제어기들을 포함한다.

[0014] SSD들, 메모리 카드들 및 플래시 드라이브들을 사용할 수 있는 호스트 시스템들은 많으며 다양하다. 이들은 데스크탑 또는 랩탑 및 그외 휴대 컴퓨터들과 같은 개인용 컴퓨터(PC)들, 셀룰라 전화들, 개인용 디지털 보조장치(personal digital assistant)들, 디지털 스틸 카메라들, 디지털 무비 카메라들 및 휴대 오디오 플레이어들을 포함한다. 휴대 메모리 카드 응용들을 위해서, 호스트는 하나 이상의 유형들의 메모리 카드들 또는 플래시 드라이브들을 위한 내장된 리셉터클을 포함할 수도 있고, 혹은 호스트는 메모리 카드를 끼우는 어댑터들을 요구할 수도 있다. 메모리 시스템은 일반적으로 자기 자신의 메모리 제어기 및 드라이버들을 내장하지만, 이 대신에 메모리가 연결된 호스트에 의해 실행되는 소프트웨어에 의해 제어되는 일부 메모리만이 있는 시스템들도 있다. 제어기를 내장하는 일부 메모리 시스템들, 특히 호스트 내에 내장된 것들에 있어서, 메모리, 제어기 및 드라이버들은 흔히 단일 집적회로 칩 상에 형성된다.

[0015] 도 1의 호스트 시스템(100)은 메모리 시스템(102)에 관한 한, 회로 및 소프트웨어의 조합으로 구성된 2개의 주요 부분들을 갖는 것으로서 볼 수도 있다. 이들은 애플리케이션 부(108), 그리고 메모리(102)와 인터페이스하는 드라이버 부(110)이다. 예를 들면, PC에서, 애플리케이션 부(108)는 워드 프로세싱, 그래픽스, 제어 또는 그외 인기있는 애플리케이션 소프트웨어를 실행하는 프로세서(이를테면 마이크로프로세서) 또는 이와 다른 유형의 제어기를 포함할 수 있다. 카메라, 셀룰라 전화 또는 단일의 한 세트의 기능들을 수행하는 데에만 주로 전용되는 그외 다른 호스트 시스템에서, 애플리케이션 부(108)는 사진들을 촬영하여 저장하기 위해 카메라를 동작시키고, 전화를 걸거나 받기 위해 셀룰라 전화를 동작시키는 등을 행하는 소프트웨어를 실행하는 프로세서 또는 이와 다른 유형의 제어기를 포함한다.

[0016] 도 1의 메모리 시스템(102)은 플래시 메모리(112), 및 회로들(114)을 포함하고 이들 둘 다는 카드가 데이터를 주고받기 위해 연결되며 메모리(112)를 제어하는 호스트와 인터페이스한다. 제어기(114)는 전형적으로 데이터 프로그램 및 관독 동안 호스트(100)에 의해 사용되는 데이터의 논리 어드레스들과 메모리(112)의 물리 어드레스들 간을 전환한다. 제어기(114)는 프로세서, 마이크로제어기, 응용특정의 집적회로(ASIC), 필드 프로그램가능 게이트 어레이, 논리 디지털 회로, 또는 그외 현재 공지되거나 나중에 개발될 논리 처리 능력을 포함할 수 있다.

[0017] 도 2는 이하 계속되는 설명에서 예로서 사용되는 플래시 메모리 셀 어레이(112)(도 1)의 구성을 개념적으로 도시한 것이다. 플래시 메모리 셀 어레이(112)는 각각 개별적으로 단일 혹은 복수의 메모리 제어기들(114)에 의해

제어되는 복수의 메모리 셀 어레이들을 포함할 수 있다. 메모리 셀들의 4개의 플레인들 혹은 서브-어레이들 (202, 204, 206, 208)은 단일 집적 메모리 셀 칩 상에, 혹은 2개의 칩들(각 칩 상에 플레인들 중 2개) 상에, 혹은 4개의 개별적 칩들 상에 있을 수 있다. 구체적 배열은 이하 논의에 중요하지 않다. 물론, 1, 2, 8, 16 혹은 그 이상과 같이, 다른 개수의 플레인들이 시스템에 존재할 수 있다. 플레인들은 이하 블록들이라 칭하는 소거의 최소 유닛을 형성하는 다수 그룹들의 메모리 셀들로 개별적으로 분할된다. 메모리 셀들의 블록들은 도 2에서 각각의 플레인들(202, 204, 206, 208)에 위치한 블록들(210, 212, 214, 216)과 같이 사각형들에 의해 도 2에 도시되었다. 각 플레인 내에 수백 혹은 수천 개의 소거 블록들이 있을 수 있다.

[0018] 위에 언급된 바와 같이, 한 블록의 메모리 셀들은 물리적으로 함께 소거될 수 있는 가장 적은 수의 메모리 셀들인 소거 유닛이다. 그러나, 증가된 병행도(parallelism)를 위해서, 블록들은 더 큰 메타블록 유닛들로 운영된다. 각 플레인으로부터 한 블록은 함께 논리적으로 연결되어 한 메타블록을 형성한다. 한 메타블록(218)을 형성하는 4개의 블록들(210, 212, 214, 216)이 도시되었다. 메타블록 내에 모든 셀들은 전형적으로 함께 소거된다. 블록들(222, 224, 226, 228)로 구성된 제 2 메타블록(220)에 도시된 바와 같이, 메타블록을 형성하기 위해 사용되는 블록들은 이들의 각각의 플레인들 내에 동일한 상대적 위치들로 제약될 필요는 없다. 높은 시스템 성능을 위해서 메타블록들을 모든 플레인들에 걸쳐 확장하는 것이 일반적으로 바람직할지라도, 메모리 시스템은 서로 다른 플레인들 내에 하나, 둘 혹은 세 개의 블록들 중 어느 하나 혹은 전부로 된 메타블록들을 동적으로 형성하는 능력을 갖고 동작될 수 있다. 이것은 한 프로그램 동작에서 메타블록의 크기가 저장을 위해 가능한 데이터 량에 더 가깝게 일치될 수 있게 한다.

[0019] 개개의 블록들은 다시 동작 목적들을 위해 도 3에 도시된 바와 같이, 메모리 셀들의 페이지들로 분할된다. 예를 들면, 블록들(210, 212, 214, 216) 각각의 메모리 셀들은 각각 8개의 페이지들(P0 ~ P7)로 분할된다. 대안적으로, 각 블록 내에 16, 32 혹은 그 이상의 페이지들의 메모리 셀들이 있을 수 있다. 페이지는 한번에 프로그램되거나 또는 판독되는 최소량의 데이터를 내포하는, 한 블록 내에 데이터 프로그램 및 판독 유닛이다. 그러나, 메모리 시스템의 동작 병행도를 증가시키기 위해서, 2 이상의 블록들 내에 이러한 페이지들은 메타페이지들로 논리적으로 연결된다. 메타페이지(302)는 4개의 블록들(210, 212, 214, 216) 각각으로부터의 한 물리 페이지로 형성되는 것으로 도 3에 도시되었다. 예를 들면, 메타페이지(302)는 4개의 블록들 각각 내에 페이지(P2)를 포함하지만 메타페이지의 페이지들은 반드시 블록들 각각 내에 동일 상대적 위치를 가질 필요는 없다. 메타페이지는 최대 프로그램 유닛이다.

[0020] 도 4는 셀 당 3비트를 가진 메모리(D3 메모리) 내 논리 어드레스가능 유닛들과 물리 메타블록들과의 정렬을 도시한 것이다. 예를 들면, 메모리는 2 ~ 32GB의 총 용량을 가진 샌디스크사에 의한 32nm 또는 43nm D3 플래시 메모리일 수 있다. 도 4는 논리 어드레스가능 유닛들과 물리 메타블록들 간에 오정렬을 예시하고자 한 것이지만 반드시 어드레스가능 유닛이 도면에 도시된 순서로 수행됨을 의미하는 것은 아니다.

[0021] 일예에서, 보안 디지털(SD) 메모리의 논리 어드레스가능 유닛 크기는 유닛들(402, 404, 406)로 나타낸 바와 같이, 4MB이다. 그러나, D3 메모리에서 물리 메타블록 크기는 메타블록들(408, 410, 412, 414)로 도시된 바와 같이, 3MB이다. 도 4에서, 점선들은 각각 1MB의 크기들을 나타낸다. 그러므로, 논리 어드레스가능 유닛들의 경계들 중 하나 또는 둘 다는 물리 메타블록 경계들과 정렬하지 않는다. 이 오정렬 때문에, 데이터가 특정 논리 어드레스가능 유닛에 기입된다면, 데이터는 도 5에 도시된 바와 같이, 2개의 메타블록들에 물리적으로 기입되어, 메타블록들 중 하나 또는 둘 다에 미사용 공간을 초래한다(도 5에서 빗금친 영역들은 데이터로 기입된 메타블록들에 공간을 나타내며 빈 영역들은 미사용 공간을 나타낸다). 미사용 공간은 미사용 공간을 회복하기 위해 복수의 가비지 수거 동작들이 필요하다면 메모리 시스템의 순차 기입 수행(Pw)이 비최적이 되게 할 수도 있을 것이다. 4MB의 논리 어드레스가능 유닛 크기 및 3MB의 물리 메타블록 크기의 본 예는 단지 예시 목적들을 위한 것이다. 다른 값들이 고려된다.

[0022] 특히, 먼저, 데이터가 어드레스가능 유닛(402)에 기입된다면, 메타블록(408) 전부와 메타블록(410)의 1MB는 인입 데이터로 기입되고 이를 도 5에서는 유형 A 어드레스가능 유닛(502)이라 칭한다. 메타블록(408)은 충만되어 닫혀지고, 메타블록(410)은 2MB의 미사용 공간을 갖는다. 두 번째로, 데이터가 어드레스가능 유닛(404)에 기입된다면, 2MB의 메타블록(410) 및 2MB의 메타블록(412)은 인입 데이터로 기입된다. 어드레스가능 유닛(404)을 유형 B 어드레스가능 유닛(504)이라 칭한다. 이 경우에, 메타블록들(410, 412)은 충만되지 않고 각 메타블록은 1MB의 미사용 공간을 갖는다. 마지막으로, 데이터가 어드레스가능 유닛(406)에 기입된다면, 1MB의 메타블록(412) 및 모든 메타블록(414) 전부는 인입 데이터로 기입된다. 어드레스가능 유닛(406)을 유형 C 어드레스가능 유닛(506)이라 칭한다. 이 경우에 메타블록(412)은 2MB의 미사용 공간을 가지며 메타블록(414)은 충만되어 닫혀

진다.

- [0023] 인입 데이터가 기입된 메타블록을 업데이트 블록이라고도 칭할 수 있다. 인입 데이터가 기입되어야 할 때 미사용 업데이트 블록들이 가용하지 않다면, 인입 데이터의 저장을 위한 업데이트 블록들을 생성하기 위해 하나 이상의 가비지 수거 동작들이 필요하다. 이하 더 상술되는 바와 같이, 유형 A 또는 유형 C 기입의 결과로서 개방 업데이트 블록은 2MB의 차후에 가비지 수거를 필요로 할 것이다. 유형 B 기입은 각각이 1MB의 차후에 가비지 수거를 필요하게 될 2개의 개방 업데이트 블록들을 유발할 것이다. 유형 B 기입의 경우에, 한 업데이트 블록만이 생성될 것이라면, 1MB의 차후에 가비지 수거만이 필요하게 된다. 따라서, 평균으로, 기입 유형들의 어느 것이든 2MB까지의 가비지 수거를 초래할 것이다.
- [0024] 그러나, 일부 상황들에서, D3 메모리에 미사용 업데이트 블록을 생성하기 위해 2MB보다 큰 가비지 수거 동작이 필요할 수도 있다. 제 1 상황은 모든 개방 업데이트 블록들이 각각 1MB의 데이터를 가질 때 일어난다. 유형 B 기입이 일어난다면, 인입 데이터를 기입하기 위해 2개의 미사용 업데이트 블록들이 필요하다. 따라서, 총 4MB의 가비지 수거를 위해 각각 2MB의 2번의 가비지 수거들이 필요할 것이다. 제 2 상황은 2MB의 데이터를 가진 하나의 개방 업데이트 블록이 있고 나머지 개방 업데이트 블록들이 각각 1MB의 데이터를 가질 때 일어난다. 다시, 유형 B 기입이 행해진다면, 인입 데이터를 기입하기 위해 2개의 미사용 업데이트 블록들이 필요하다. 이 경우에, 총 3MB의 가비지 수거를 위해서는 각각 1MB 및 2MB의 최소 2개의 가비지 수거들이 필요할 것이다.
- [0025] 속도 클래스 동작 동안, 어드레스가능 유닛의 각 기입마다, 인입 데이터의 기입들은 어드레스가능 유닛을 구성하는 기록 유닛들에 기입된다. 종래의 메모리 시스템에서, 업데이트 블록이 중단되었을 때 가비지 수거 동작이 시작된다. 그러나, 속도 클래스 동작의 측정 부분 동안에 가비지 수거 동작이 일어난다면, Pw 메트릭(어드레스가능 유닛의 순차 기입 수행)은 악영향을 받을 수 있고 속도 클래스 명세에 의해 지시된 요망되는 최소 전송 속도는 충족되지 않을 수 있다. 위에 논의된 바와 같이 최악의 경우의 시나리오는 총 4MB에 대해 각각 2번의 가비지 수거 동작들을 요구할 것이다. D3 메모리에서 4MB 가비지 수거 동작은 Pw 메트릭에 악영향을 미칠 것이다.
- [0026] 도 6은 메모리 블록들의 우선적 가비지 수거를 수반하는 메모리 장치의 수행을 최적화하는 방법(600)을 도시한 흐름도이다. 위에 기술된 최악의 경우의 시나리오를 피하기 위해서, 방법(600)은 인입 데이터의 기입이 있을 때 (이들테면 인입 데이터의 기입이 있을 때마다) 소정의 량의 우선적 가비지 수거 동작들을 수행한다. D3 메모리에서, 우선적 가비지 수거의 량은 예를 들면 2MB일 수 있다. 방법(600)은 메모리 장치가 속도 동작 동안 가비지 수거 동작과 같은 작업을 수행하기 위한 버짓(버짓)을 충족시킬 수 있게 한다. 방법(600)은 비지(busy) 시간 프로세스(이들테면 호스트 명령들 동안 동작하는) 혹은 아이들 시간 프로세스(이들테면 호스트 명령들 사이에 동작하는)로서 수행될 수 있다. 메모리 시스템(102)에서 메모리 제어기(114)는 예를 들면 방법(600)을 구현할 수 있다. 방법(600)에 대한 도 6에 기술된 단계들 각각은 단독으로 혹은 다른 단계들과 조합하여 수행될 수 있다.
- [0027] 단계(602)에서, 논리 어드레스가능 유닛에 기입될 인입 데이터는 메모리 장치에 의해 수신된다. FAT 및 비(non)-FAT 데이터를 포함한, 임의의 유형의 데이터가 메모리에 기입될 수 있다. 방법(600)은 반드시 속도 클래스 동작이 시작되었는지를 아는 것은 아니지만 대신 항상 동작할 수 있다. 단계(604)에서, 데이터 기입에 기인한 필요한 가비지 수거의 량이 판정된다. 위에 상술된 바와 같이, 데이터 기입시 미사용 업데이트 블록들이 가용하지 않다면, 하나 이상의 개방 업데이트 블록들은 미사용 업데이트 블록들을 생성하기 위해 가비지 수거 동작이 행해진다. 미사용 업데이트 블록이 가용하다면, 단계(604)에서 가비지 수거의 어떠한 량도 필요하지 않은 것이 가능하다.
- [0028] 단계(602)에서 판정된 가비지 수거량은 단계(606)에서 소정의 임계값에 비교된다. 임계값은 예를 들면, D3 메모리에 대해선 2MB일 수 있고, 혹은 다른 량일 수도 있다. 2MB 임계값은 메모리의 순차 기입 수행 Pw이 속도 클래스 동작 동안 악영향을 받지 않게 D3 메모리에 대해 선택될 수 있다. 가비지 수거량이 단계(606)에서 임계값 이하라면, 가비지 수거 동작을 위해 단계(608)에서 알고리즘에 기초하여 하나 이상의 블록들이 선택된다. 대안적으로, 단계(606)는 가비지 수거량이 제 1 임계값 이하이고 제 2 임계값 이상인지를 판정할 수 있다. 그러하다면, 하나 이상의 블록들이 가비지 수거 동작을 위해 단계(608)에서 알고리즘에 기초하여 선택될 수 있다.
- [0029] 단계(608)에서 블록을 선택하기 위해 하나 이상의 알고리즘들이 사용될 수 있다. 일예의 알고리즘은 알고리즘을 행하기 위한 최소한의 작업(혹은 최소량의 카피)을 포함할 수 있다. 알고리즘을 행하기 위한 최소한의 작업은 합체할 최소한의 데이터량을 가진 업데이트 블록을 판정하기 위해 하나 이상의 업데이트 블록들 내 데이터를 분석할 수 있다. 알고리즘을 행하기 위한 최소한의 작업은 업데이트 블록을 채우기 위해 필요한 공간량의 표시자로서 업데이트 블록 자원 내 기입 포인터를 검사할 수 있다. 또 다른 예의 알고리즘은 최근에 최소로 사용된 업

데이트 블록들을 선택하기 위한 최근에 최소 사용(LRU) 알고리즘을 포함할 수 있다. LRU 알고리즘은 하나 이상의 최근에 최소로 사용된 업데이트 블록들을 선택하기 위해 어떤 블록들이 언제 사용되었는가를 관리할 수 있다. 또 다른 예의 알고리즘은 최근에 최다 사용된 업데이트 블록들을 선택하기 위한 최근에 최다 사용(MRU) 알고리즘을 포함할 수 있다. MRU 알고리즘은 하나 이상의 최근에 최다 사용된 업데이트 블록들을 선택하기 위해 어떤 블록들이 언제 사용되었는가를 관리할 수 있다.

[0030] 가비지 수거의 임계량(이를테면 단계(606)에서 임계값)은 단계(604)에서 판정된 양이 임계량 미만일지라도 방법(600)에 의해 강제된다. 가비지 수거의 판정된 양이 임계값 미만인 상황은 메모리 장치가 정상상태(steady state)에 있을 때로서, 예를 들면, 메모리 장치 내 모든 업데이트 블록들이 데이터 기입들의 결과일 때 일어날 수 있다. 단계(610)에서, 강제된 임계량의 가비지 수거 동작은 단계(608)에서 다른 블록들로부터의 데이터를 선택된 업데이트 블록들에 카피함으로써 수행된다. 가비지 수거 동작의 결과로서 하나 이상의 미사용 업데이트 블록들이 생성된다. 마지막으로, 단계(612)에서, 인입 데이터는 생성된 미사용 업데이트 블록들에 기입되고 방법(600)은 완료된다.

[0031] 도 7은 메모리 장치가 정상상태에 있을 때 도 6에 도시된 방법(600)이 행해진 업데이트 블록들의 예를 도시한 것이다. 업데이트 블록들(702 ~ 708)의 상단 행은 유형 C 데이터 기입에 앞서 개방 업데이트 블록들을 나타내며, 업데이트 블록들(710 ~ 718)의 하단 행은 유형 C 데이터 기입 후에 업데이트 블록들을 나타낸다. 상단 행에서, 업데이트 블록들(702, 704)은 이전의 유형 B 데이터 기입의 결과들이고, 업데이트 블록(706)은 이전 유형 C 데이터 기입의 결과이고, 업데이트 블록(708)은 이전 유형 A 데이터 기입의 결과이다.

[0032] 인입 데이터의 유형 C 기입은 위에 논의된 바와 같이 데이터를 저장하기 위해 2개의 업데이트 블록들을 요구한다. D3 메모리에 유형 C 기입에 있어서, 한 업데이트 블록은 완전히 3 MB의 데이터로 채워지고 제 2 업데이트 블록은 1MB의 데이터로 채워지고 나머지는 개방된다. 예를 들면, 도 7의 하단 행에서 업데이트 블록(718)은 유형 C 기입 후에 개방 업데이트 블록을 도시한 것이다. 데이터 기입 시에 미사용 업데이트 블록들이 가용하지 않다면, 업데이트 블록(702)이 최근에 최소로 사용된 개방 업데이트 블록이기 때문에 이것이 선택된다. 업데이트 블록(702)은 1MB의 가비지 수거가 행해져 기입 후 온전한 업데이트 블록(710)이 된다. 그러나, 방법(600)을 사용할 때 가비지 수거의 임계량(2MB)이 강제되기 때문에, 가비지 수거 동작을 위해 업데이트 블록(704)이 선택된다. 업데이트 블록(704)은 1MB의 우선적 가비지 수거가 행해져 기입 후에 온전한 업데이트 블록(712)이 된다. 업데이트 블록들(714, 716)은 기입 후에 각각 업데이트 블록(706, 708)과 동일한 데이터를 가진 동일한 상태로 그대로 있다.

[0033] 도 8은 메모리 장치가 정상상태에 있을 때 도 6에 도시된 방법(600)이 행해진 업데이트 블록들의 또 다른 예를 도시한 것이다. 도 7과 유사하게, 업데이트 블록들(802 ~ 806)의 상단 행은 유형 C 데이터 기입 전에 개방 업데이트 블록들을 나타내며, 업데이트 블록들(808 ~ 814)의 하단 행은 유형 C 데이터 기입 후에 업데이트 블록들을 나타낸다. 상단 행에서, 업데이트 블록(802)은 이전 유형 B 데이터 기입의 결과이고, 업데이트 블록들(804, 806)은 이전 유형 C 데이터 기입들의 결과들이다.

[0034] 도 8에 도시된 예에서, D3 메모리를 위한 인입 데이터의 유형 C 기입은 3MB의 데이터를 가진 한 업데이트 블록 및 1MB의 데이터를 가진 제 2 업데이트 블록을 완전히 채운다. 제 2 업데이트 블록, 예를 들면, 업데이트 블록(814)은 데이터 기입 후에 개방된 그대로 있다. 데이터 기입 시에 미사용 업데이트 블록들이 가용하다면, 데이터 기입을 위한 업데이트 블록들을 생성하기 위해 종래의 메모리 장치에서는 어떠한 가비지 수거도 필요하지 않을 것이다. 그러나, 방법(600)에 따라, 업데이트 블록들(802, 804)이 최근에 최소로 사용된 업데이트 블록들이기 때문에 우선적 가비지 수거를 위해 이들이 선택된다. 각 업데이트 블록(802, 804)은 1MB의 가비지 수거가 행해진다. 따라서, 업데이트 블록(802)은 데이터 기입 후에 온전한 업데이트 블록(808)이 된다. 또 다른 1MB가 업데이트 블록(804)로부터 가비지 수거되어 업데이트 블록(810)이 된다. 업데이트 블록(810)은 아직 가용한 1MB의 공간이 있기 때문에 개방된 채로 그대로 있다. 업데이트 블록(812)은 기입 후에 업데이트 블록(806)과 동일한 데이터를 가진 동일한 상태 그대로 있다. 이 예에서, 가비지 수거의 어떠한 량도 반드시 필요한 것이 아니기 때문에 업데이트 블록들(802, 804)은 둘 다 방법(600)에 의해 우선적으로 가비지 수거된다. 예를 들면, 블록(804)은 단지 부분적으로만 가비지 수거된다.

[0035] 도 9는 메모리 장치가 정상상태에 있을 때 도 6에 도시된 방법(600)이 행해진 업데이트 블록들의 또 다른 예를 도시한 것이다. 업데이트 블록들(902 ~ 906)의 상단 행은 유형 C 데이터 기입 전에 개방 업데이트 블록들을 나타내며, 업데이트 블록들(908 ~ 914)의 하단 행은 유형 C 데이터 기입 후에 업데이트 블록들을 나타낸다. 상단 행에서, 업데이트 블록들(902, 906)은 이전의 유형 C 데이터 기입의 결과이고, 업데이트 블록(904)은 이전의 유

형 A 데이터 기입의 결과이다.

- [0036] D3 메모리를 위한 인입 데이터의 유형 C 기입은 3MB의 데이터를 가진 한 업데이트 블록 및 1MB의 데이터를 가진 제 2 업데이트 블록을 완전히 채운다. 제 2 업데이트 블록, 예를 들면, 업데이트 블록(914)은 유형 C 데이터 기입 후에 개방된 채로 그대로 있다. 미사용 업데이트 블록들이 가용하다면, 데이터 기입을 위해서 미사용 업데이트 블록들을 생성하기 위해 종래의 메모리 장치에서는 어떠한 가비지 수거도 요구되지 않을 것이다. 그러나, 방법(600)에 따라 가비지 수거를 위해 업데이트 블록(902)이 최근에 최소로 사용된 업데이트 블록이기 때문에 이것이 선택된다. 업데이트 블록(902)은 2MB의 가비지 수거가 행해져, 데이터 기입 후에 온전한 업데이트 블록(908)이 된다. 업데이트 블록들(910, 912)은 데이터 기입 후에 각각 업데이트 블록들(904, 906)과 동일한 데이터를 갖고 동일한 채로 그대로 있다. 유형 C 기입에 의해 가비지 수거의 어떠한 량도 요구되지 않기 때문에 이 예에서 업데이트 블록(902)은 우선적으로 가비지 수거된다.
- [0037] 도 6로 다시 돌아가면, 단계(604)에서 결정된 가비지 수거량이 단계(606)에서 임계값보다 크다면, 방법(600)은 단계(614)로 계속된다. 단계(614)에서, 가비지 수거 동작을 위해 하나 이상의 대안적 업데이트 블록들(반드시 최근에 최소로 사용된 업데이트 블록들은 아닌)이 선택된다. 메모리 장치가 초기 조건 상태, 예를 들면, 메모리 장치 내 업데이트 블록들이 전부 데이터 기입들의 결과가 아닐 때 일어난다. 최근에 최소로 사용된 업데이트 블록은, 선택된다면, 임계량 이상의 가비지 수거가 행해질 필요가 있어 속도 클래스 동작에 악영향을 미치게 된다. 대안적 업데이트 블록들을 선택함으로써, 필요한 가비지 수거는 속도 클래스 동작의 순차 기입 수행 측정에 악영향을 미침이 없이 여전히 수행될 수 있다.
- [0038] 단계(614)에서 선택되는 대안적 업데이트 블록은 단계(602)에서 데이터 기입의 유형에 따른다. 유형 A 또는 유형 C 기입이 행해진다면, 단계(614)에서 가비지 수거 동작을 위해 최소량의 요구되는 가비지 수거를 가진 개방 업데이트 블록이 선택된다. 단계(616)에서 선택된 개방 업데이트 블록이 임계량 이상의 가비지 수거를 요구하지 않는다면, 가비지 수거 동작은 다른 블록들로부터의 데이터를 선택된 개방 업데이트 블록에 카피함으로써 단계(610)에서 수행된다. 단계(612)에서 인입 데이터는 생성된 미사용 업데이트 블록들에 기입된다. 그러나, 선택된 개방 업데이트 블록이 단계(616)에서 임계량 이상의 가비지 수거를 요구한다면, 인입 데이터는 단계(618)에서 비휘발성 저장 영역, 이를테면 바이너리 캐시 또는 랜덤 액세스 메모리에 기입된다. 임계량의 가비지 수거 동작은 여전히 단계(620)에서 수행된다. 예를 들면, 데이터는 버퍼 랜덤 액세스 메모리(BRAM)에 기입될 수도 있다. 데이터는 업데이트 블록을 비우는 프로세스를 가비지 수거가 완료할 때까지 BRAM 내에 유지될 수 있다.
- [0039] 유형 B 기입이 단계(602)에서 행해진다면, 단계(614)에서 가비지 수거 동작을 위해 하나 이상의 대안적 개방 업데이트 블록들이 선택된다. 미사용 업데이트 블록들이 없다면, 단계(614)에서 하나 이상의 개방 업데이트 블록들이 선택된다. 예를 들면, 첫번째 2MB의 기입 동안에는 제 1 개방 업데이트 블록이 선택될 수 있고, 두번째 2MB를 기입할 때 제 2 개방 업데이트 블록이 선택될 수 있다. 또 다른 예로서, 복수의 업데이트 블록들이 동시에 선택될 수도 있다. 2개의 선택된 업데이트 블록들을 가비지 수거하기 위해 필요한 총 데이터량이 단계(616)에서 임계량보다 크다면, 단계(618)에서 인입 데이터는 임시 저장 영역에 기입된다. 임계량의 가비지 수거는 여전히 단계(620)에서 수행된다. 그러나, 단계(616)에서 2개의 선택된 업데이트 블록들을 가비지 수거하기 위해 필요한 총 데이터량이 임계량 이하이면, 가비지 수거 동작이 단계(610)에서 수행된다. 이 경우에, 임계량의 반 이하를 가진 선택된 업데이트 블록이 먼저 가비지 수거되고, 이어 다른 선택된 업데이트 블록에 대해 가비지 수거 동작이 이어진다. 인입 데이터는 단계(612)에서 생성된 미사용 업데이트 블록들에 기입된다.
- [0040] 도 10은 메모리 장치가 초기 조건 상태에 있을 때 도 6에 도시된 방법(600)이 행해진 업데이트 블록들의 예를 도시한 것이다. 업데이트 블록들(1002 ~ 1006)의 상단 행은 유형 B 데이터 기입 전에 개방 업데이트 블록들을 나타내며, 업데이트 블록들(1008 ~ 1014)의 하단 행은 유형 B 데이터 기입 후에 업데이트 블록들을 나타낸다. 상단 행에서, 업데이트 블록들(1002)은 이전의 유형 C 데이터 기입의 결과이다. 업데이트 블록(1004)은 1.5MB의 데이터로 채워지고 업데이트 블록(1006)은 2.5MB의 데이터로 채워진다. 따라서, 업데이트 블록(1004)은 1.5MB의 차후에 가비지 수거를 필요로 할 것이며 업데이트 블록(1006)은 0.5MB의 차후에 가비지 수거를 필요로 할 것이다.
- [0041] D3 메모리에 대한 인입 데이터의 유형 B 기입은 각각 2MB의 데이터로 2개의 업데이트 블록들을 채워, 두 업데이트 블록들에 데이터 기입 후에 1MB의 가용한 공간을 남긴다. 예를 들면, 업데이트 블록들(1014, 1016)은 이 경우에 유형 B 기입 후에 결과적인 개방 업데이트 블록들을 나타낸다. 유형 B 기입이 2개의 개방 업데이트 블록들이 되게 하기 때문에, 2개의 미사용 업데이트 블록들을 생성하기 위해 2개의 현재 개방된 업데이트 블록들이 가비지 수거될 필요가 있다. 이 예에서, 2개의 최근에 최소로 사용된 개방 업데이트 블록들이 선택된다면, 2MB의

임계량 이상이 가비지 수거될 것이다. 특히, 업데이트 블록(1002)은 2MB의 가비지 수거가 행해질 것이며 업데이트 블록(1004)은 1.5MB의 가비지 수거가 행하여 총 3.5MB의 가비지 수거가 되게 한다.

[0042] 가비지 수거량이 임계값을 초과하기 때문에, 최근에 최소로 사용된 업데이트 블록들 대신 가비지 수거를 위해 대안적 업데이트 블록들이 선택된다. 이 예에서, 가비지 수거를 위해 업데이트 블록들(1004, 1006)이 선택되고, 업데이트 블록(1006)은 임계값 이하, 예를 들면, 0.5MB를 가진 블록이기 때문에 이 블록이 먼저 가비지 수거되며, 업데이트 블록(1006)의 가비지 수거로 데이터 기입 후에 업데이트 블록(1012)은 온전하게 된다. 업데이트 블록(1004)은 1.5MB 량으로 가비지 수거가 행하여, 데이터 기입 후에 온전한 업데이트 블록(1010)이 된다. 업데이트 블록(1008)은 데이터 기입 후에 업데이트 블록(1002)과 동일한 채로 그대로 있다. 최근에 최소로 사용된 업데이트 블록들이 아니라 이 시나리오에서 대안적 업데이트 블록들을 선택함으로써 임계량의 가비지 수거는 여전히 우선적으로 수행된다.

[0043] 도 11은 도 12의 상태머신을 포함하는 메모리 장치의 수행을 최적화하는 대안적 방법(1100)을 도시한 흐름도이다. 방법(1100)은 이블테면 도 6의 방법(600)의 단계(608)에 관련하여 위에서 논의된 바와 같이, 가비지 수거를 위해 최근에 최소로 사용된 업데이트 블록을 선택하는 대신에, 가비지 수거를 위해 개방 업데이트 블록을 선택하는 대안적 알고리즘을 기술한다. 방법(1100)은 메모리 장치가 정상상태에 있을 때, 예를 들면, 메모리 장치 내 모든 업데이트 블록들이 데이터 기입들의 결과일 때 사용될 수 있다. 방법(1100)은 메모리 장치가 속도 클래스 동작 동안, 가비지 수거 동작과 같은 작업을 수행하기 위한 버짓을 충족할 수 있게 한다. 방법(1100)은 비지 시간 프로세스 또는 아이들 시간 프로세스로서 수행될 수도 있다. 메모리 시스템(102) 내에 메모리 제어기(114)는 예를 들면 방법(1100)을 구현할 수 있다. 방법(1100)에 대해 도 11에 기술된 단계들 각각은 단독으로 다른 단계들과 조합하여 수행될 수도 있다.

[0044] 방법(1100)에서, 2개의 업데이트 블록들은 가비지 수거 동작들을 위해 상시 선택된다. 2개의 업데이트 블록들은 도 12에 도시된 바와 같이, 2개의 상태들로서 상태 A 또는 상태 B 중 하나에 있다. 상태 A에서, 업데이트 블록(1202)은 미사용이며, 업데이트 블록(1204)은 이전 유형 A 또는 유형 C 데이터 기입의 결과로서 개방 업데이트 블록이다. 업데이트 블록(1202)은 데이터를 내포하지 않으며 차후에 가비지 수거를 요구하지 않으며 업데이트 블록(1204)은 1MB의 데이터를 가지며 2MB의 차후에 가비지 수거를 요구할 것이다. 상태 B은 업데이트 블록들(1206, 1208)을 포함하며, 이것은 이전 유형 B 데이터 기입의 결과로서 개방 업데이트 블록들이다. 각 업데이트 블록(1206, 1208)은 2MB의 데이터를 내포하며 1MB의 차후에 가비지 수거를 요구할 것이다. 메모리 장치는 일단 상태 A 또는 상태 B 중 어느 하나에서 2개의 업데이트 블록들이 발견되면 방법(1100)을 이용할 수 있다. 방법(600)의 단계들(614 ~ 618)에서 기술된 대안적 업데이트 블록 선택은 방법(1100)을 사용하기 위해 필요한 정상상태를 달성하기 위해 메모리 장치가 초기 조건 상태에 있을 때 이용될 수 있다.

[0045] 상태 A와 상태 B 간에 그리고 이 내에서 천이는 D3 메모리에 2MB의 가비지 수거를 요구한다. 2MB의 가비지 수거가 상태들 간에 그리고 이 내에서 천이기 위해서 항상 수행되기 때문에, 가비지 수거 버짓이 충족되고 속도 클래스 동작은 악영향을 받지 않는다. 방법(1100)의 단계(1102)에서, 논리 어드레스가능 유닛에 기입될 인입 데이터가 메모리 장치에 의해 수행된다. FAT 및 비-FAT 데이터를 포함하여, 임의의 유형의 데이터가 메모리에 기입될 수 있다. 방법(1100)은 속도 클래스 동작이 시작되었는지를 아는 것은 아니지만 대신 항상 동작할 수 있다.

[0046] 단계(1104)에서, 도 12에 기술된 바와 같이 2개의 선택된 업데이트 블록들이 상태 A 또는 상태 B에 있는지가 판정된다. 업데이트 블록들이 상태 A에 있다면, 방법(1100)은 단계(1102)에서 행해지는 데이터 기입이 어떤 유형인지 판정되는 단계(1106)에 계속된다. 데이터 기입이 유형 A 또는 유형 C 데이터 기입이라면, 업데이트 블록들(1202, 1204)은 업데이트 블록들이 상태 A에 머물러 있게 단계(1110)에서 가비지 수거된다. 특히, 업데이트 블록들(1202, 1204)은 유형 A 또는 유형 C 데이터 기입이 1MB의 데이터 및 2MB의 가용한 공간을 내포하는 개방 업데이트 블록이 되게 하기 때문에 상태 A에 머물러 있다. 그러나, 단계(1106)에서 데이터 기입의 유형이 유형 B 기입이라면, 업데이트 블록들(1202, 1204)은 업데이트 블록들이 상태 B로 천이하고 업데이트 블록들이 업데이트 블록들(1206, 1208)이 되게 단계(1112)에서 가비지 수거가 행해진다. 이 경우에, 유형 B 데이터 기입이 업데이트 블록들(1202, 1204)는 각각 2MB의 데이터 및 1MB의 가용한 공간을 내포하는 2개의 개방 업데이트 블록들이 되게 하기 때문에 상태 B로 천이한다. 단계(1110) 및 단계(1112)에서, 기입될 인입 데이터는 하나 이상의 업데이트 블록들에 저장된다.

[0047] 한편, 단계(1104)에서, 2개의 선택된 업데이트 블록들이 상태 B에 있는 것으로 판정되면, 방법(1100)은 단계(1108)로 계속된다. 단계(1108)에서, 단계(1102)에서 데이터 기입의 어떤 유형이 행해졌는지가 판정된다. 데이

터 기입이 유형 A 또는 유형 C 데이터 기입이라면, 업데이트 블록들(1206, 1208)은 업데이트 블록들이 상태 A로 천이하어 업데이트 블록들(1202, 1204)이 되게 단계(1114)에서 가비지 수거된다. 특히, 유형 A 또는 유형 C 데이터 기입으로 1MB의 데이터 및 2MB의 가용한 공간을 내포하는 개방 업데이트 블록이 되게 하기 때문에 업데이트 블록들(1206, 1208)은 상태 A로 천이한다. 그러나, 단계(1108)에서 데이터 기입이 유형이 유형 B 기입이라면, 업데이트 블록들(1206, 1208)은 업데이트 블록들이 상태 B에 머물러 있게 단계(1116)에서 가비지 수거된다. 이 경우에, 유형 B 데이터 기입으로 각각 2MB의 데이터 및 1MB의 가용한 공간을 내포하는 2개의 개방 업데이트 블록들이 되게 하기 때문에 업데이트 블록들(1206, 1208)은 상태 B에 머물러 있다. 전처럼, 두 단계들(1114, 1116)에서, 기입될 인입 데이터는 하나 이상의 업데이트 블록들에 저장된다.

[0048] 도 13은 가비지 수거에 관계된 데이터의 적어도 일부를 휘발성 메모리에 카피하는 것을 수반하는 메모리 장치의 수행을 최적화하는 방법(1300)을 예시한 흐름도이다. 일실시예에서, 업데이트 블록을 위한 제어 정보의 일부 또는 모두는 랜덤 액세스 메모리 또는 다른 유형의 휘발성 메모리에 카피될 수 있다. 또 다른 실시예에서, 가비지 수거가 행해진 단편화된 데이터의 일부 또는 모두는 랜덤 액세스 메모리 또는 다른 유형의 휘발성 메모리에 카피될 수 있다. 그후에, 카피된 단편화된 데이터는 비휘발성 메모리에 카피될 수 있다. 또 다른 실시예에서, 제어 정보의 일부 또는 모두 및 가비지 수거가 행해진 단편화된 데이터의 일부 또는 모두는 랜덤 액세스 메모리 또는 다른 유형의 휘발성 메모리에 카피될 수 있다.

[0049] 제어 정보와 같은 정보의 일부를 임시로 저장하기 위해 랜덤 액세스 메모리와 같은 휘발성 메모리를 사용하는 것은 속도 클래스 동작 동안 메모리 장치의 수행을 개선할 수 있다. 방법(1300)은 비지 시간 프로세스 또는 아이들 시간 프로세스로서 수행될 수 있다. 메모리 제어기(114)는 예를 들면 방법(1300)을 구현할 수 있다. 방법(1300)에 대해 도 13에 기술된 단계들 각각은 단독으로 또는 조합하여 수행될 수 있다.

[0050] 단계(1302)에서, 논리 어드레스가능 유닛에 기입될 인입 데이터는 메모리 장치에 의해 수신된다. 단계(1304)에서, 수신된 인입 데이터량은 임계값과 비교된다. 데이터량의 검출은 데이터의 시작부분이 기록 유닛들의 길이들로 수신될 때로 제약될 수 있다. 논리 어드레스가능 유닛에 기입될 데이터가 기록 유닛들의 길이들로 수신될 때, 속도 클래스 동작은 시작되었을 수도 있다. 어드레스가능 유닛이 4MB의 길이를 갖는다면, 기록 유닛은 예를 들면 16kB의 길이를 갖는다. 단계(1304)에서 검출된 데이터량이 임계값 미만이라면, 방법(1300)은 더 많은 인입 데이터를 수신하기 위해 단계(1302)로 되돌아 간다.

[0051] 그러나, 단계(1304)에서 데이터량이 임계값이거나 그 이상이라면, 방법(1300)은 단계(1306)로 계속된다. 임계값은 논리 어드레스가능 유닛의 80%이거나, 다른 량 또는 백분율일 수도 있다. 인입 데이터는 단계(1306)에서 메모리 장치의 물리 블록들에 기입된다. 단계(1308)에서, 메모리 장치에서 임의의 개방 업데이트 블록들의 가비지 수거가 수행된다. 임의의 량의 가비지 수거는 속도 클래스 동작에 악영향을 미침이 없이 필요시 단계(1308)에서 수행될 수 있다. 속도 클래스 동작의 제 1 어드레스가능 유닛 기입이 순차 기입 수행(Pw)을 측정하기 위해 사용되지 않기 때문에 속도 클래스 동작에 악영향은 없다.

[0052] 일실시예에서, 단계(1308)에서 수행되는 가비지 수거 동작에 이어, 단계(1310)에서 보인 바와 같이, 기입될 다음 데이터의 기입 유형이 확인된다. 앞에서 기술된 바와 같이 유형 A, B, 또는 C 중 어느 하나인 기입 유형은 기입되는 어드레스가능 유닛의 논리 시작 어드레스를 검사함으로써 판정될 수 있다.

[0053] 일실시예에서, 가비지 수거에 관계된 데이터의 일부 또는 부분은 자동으로 휘발성 메모리에 기입될 수 있다. 대안적 실시예에서, 가비지 수거에 관계된 데이터의 일부 또는 부분은 가비지 수거의 하나 이상의 특징들에 따라 휘발성 메모리에 기입될 수 있다. 예를 들면, 가비지 수거에 관계된 데이터의 부분을 기입할지 판정은 기입 유형에 따를 수 있다. 특히, 업데이트 블록에 관한 제어 정보는 확인된 기입 유형에 따라 랜덤 액세스 메모리 또는 플래시 메모리에 저장될 수 있다. 또 다른 예로서, 가비지 수거에 관계된 데이터의 부분을 기입할지 판정은 기입이 업데이트 블록을 채울지 여부에 따를 수 있다. 구체적으로, 휘발성 메모리에 정보의 부분을 기입할지 판정이 행해진다. 예를 들면, 판정은 다음 인입 데이터를 분석한 것에 기초할 수 있다. 특히, 단계(1312)에서, 다음 인입 데이터의 확인된 기입 유형이 업데이트 블록을 채워 온전한 블록을 생성하게 될지가 판정된다. 다음 인입 데이터의 기입 유형이 업데이트 블록을 완전히 채울 것이라면, 정보의 부분이 휘발성 메모리에 저장되는데, 이를테면 제어 정보는 단계(1316)에서 랜덤 액세스 메모리(RAM)에 저장된다. 제어 정보는 논리 어드레스가능 유닛의 정보를 물리 메타블록 또는 다른 정보에 매핑하는 것을 포함할 수 있다. 제어 정보의 예들은 논리 그룹, 메타블록 수, 기입 포인터, 및 페이지 태그를 포함하나 이들로 제한되지 않는다.

[0054] 제어 정보를 플래시 메모리 대신에 RAM에 저장함으로써, 메모리 장치 내 사용가능한 업데이트 블록들의 최대수가 늘어날 수 있어 자원들이 없음에 기인하여 강제된 가비지 수거 동작들은 감소하게 된다. 그러나, 다음 인입

데이터의 기입 유형이 단계(1312)에서 업데이트 블록을 완전히 채우지 않을 것이라면, 논리 어드레스가능 유닛에 대한 제어 정보는 단계(1314)에서 플래시 메모리에 저장된다.

[0055] 제어 정보가 RAM에 저장된다면, 논리 어드레스가능 유닛에 대응하는 파워 사이클 플래그 또는 이외 다른 표시자가 플래시 메모리 내에 설정될 수 있다. 파워 사이클 플래그를 사용함으로써, RAM에 저장된 제어 정보를 가진 업데이트 블록이 단락지기 전에 메모리 장치에의 파워 상실이 발생한다면, 제어 정보의 회복은 메모리 장치의 재초기화시 일어날 수 있다. 재초기화가 시작된 후에, 파워 사이클 플래그가 설정된다면, 제어 정보는 재구성될 수 있다(RAM에 이전에 저장된 제어 정보가 유실될지라도). 제어 정보를 재구성하는 한 방법은 자유인 업데이트 블록들의 목록인 자유 업데이트 블록 목록을 사용함에 의한다. 특히, 자유 업데이트 블록 목록은 자유인 업데이트 블록들을 판정하기 위해 사용될 수 있고, 이들 자유인 업데이트 블록들에 연관된 데이터의 적어도 일부가 분석될 수 있다. 예를 들면, 헤더 정보가 분석될 수 있고(이들테면 제 1 페이지 내 헤더 정보), 따라서 업데이트 블록 내 내포된 데이터가 판정될 수 있다. 일단 업데이트 블록 내 내포된 데이터가 판정되면, 제어 정보가 구성될 수 있다. 이어서, 이 업데이트 블록에 대해 재구성된 제어 정보는 메모리 장치의 재초기화 후에 또 다른 업데이트 블록의 가비지 수거에 이어 플래시 메모리에 저장될 수 있다. 또는, 이 업데이트 블록에 대한 재구성된 제어 정보는 다시 휘발성 메모리(이들테면 RAM)에 저장될 수 있다.

[0056] 도 14는 가비지 수거(이들테면 우선적 가비지 수거)를 수행하기 위해 어느 동작 모드를 선택할지의 흐름도이다. 특정 메모리 장치는 서로 다른 동작 모드들에서 다르게 가비지 수거를 수행할 수 있다. 한 동작 모드는 다른 동작 모드보다 가비지 수거를 수행하기가 더 효율적일 수 있다. 예를 들면, 32 나노미터 플래시 메모리는 제 1 동작 모드(이들테면 비지 기간에서 가비지 수거)에서 동작할 수 있고 제 2 동작 모드(이들테면 논리 어드레스가능 유닛의 기입동안 가비지 수거)에서 동작할 수도 있다. 제 1 동작 모드에서 수행될 수 있는 32 나노미터 플래시 메모리에 대한 가비지 수거량은 제 2 동작 모드에 대해 수행될 수 있는 가비지 수거량과는 다를 수 있다. 또 다른 예로서, 70 나노미터 플래시 메모리는 제 2 동작 모드(이들테면 논리 어드레스가능 유닛의 기입 동안 가비지 수거)와는 다르게 제 1 동작 모드(이들테면 비지 기간에서 가비지 수거)에서 가비지 수거를 수행할 수 있다.

[0057] 앞에서, 가비지 수거는 메모리 제약들에 기인하여 수행되는 것이 필요하였을 때 수행되었다. 반대로, 가비지 수거는 이들테면 도 14에 도시된 바와 같이, 플래시 메모리가 가비지 수거를 수행하기에 더 효율적인 선택된 동작 모드에서 수행될 수도 있다. 위에서 논의된 바와 같이, 가비지 수거는 우선적으로 수행될 수도 있다(도 6에 관련하여 논의된 바와 같이). 가비지 수거를 우선적으로 수행함에 있어, 플래시 메모리는 가비지 수거를 수행하는 것이 더 나은(또는 최선인) 동작 모드를 판정할 수 있다. 예를 들면, 플래시 메모리는 비지 기간 동안 수행된다면 가비지 수거가 더 효율적인 것으로 판정할 수 있다. 그러하다면, 플래시 메모리는 비지 기간 동안 우선적 가비지 수거를 수행할 수 있다. 또 다른 예로서, 플래시 메모리는 논리 어드레스가능 유닛의 기입 동안 수행된다면 가비지 수거가 더 효율적인 것으로 판정할 수 있다. 그러하다면, 플래시 메모리는 논리 어드레스가능 유닛의 기입 동안 우선적 가비지 수거를 수행할 수 있다.

[0058] 도 14는 가비지 수거를 수행하기 위해 어느 모드를 선택할 것인가 하는, 즉 가비지 수거의 량들을 모니터함으로써 메모리 장치의 수행을 최적화하는 특정한 예를 보여주는 방법(1400)을 도시한 흐름도이다. 메모리는 수거를 수행하기 위해 어느 모드를 선택할 것인가를 하나 이상의 표시자들, 이들테면 동작의 서로 다른 기간들에 최대량들의 가비지 수거에 기초한 표시자들에 기초하여 판정할 수 있다. 특히, 방법(1400)은 메모리 장치의 단일 비지 기간에 최대량의 가비지 수거 및 논리 어드레스가능 유닛의 기입 동안 카피되는 최대량의 가비지 수거를 모니터한다. 방법(1400)을 사용함으로써, 비지 기간에 최대량의 가비지 수거는, 요구되지 않을지라도 우선적으로 수행될 수 있다. 방법(1400)은 비지 시간 프로세스 또는 아이들 시간 프로세스로서 수행될 수 있다. 메모리 제어기(114)는 예를 들면 방법(1400)을 구현할 수 있다. 방법(1400)에 대해 도 14에 기술된 단계들 각각은 단독으로 또는 조합하여 수행될 수 있다.

[0059] 단계(1402)에서, 논리 어드레스가능 유닛에 기입될 인입 데이터가 메모리 장치에 의해 수신된다. 인입 데이터가 수신된 후에 2개의 파라미터들이 단계(1404)에서 액세스된다. 제 1 파라미터는 메모리 장치의 단일 비지 기간에 수행될 수 있는 최대량의 가비지 수거이다. 비지 기간은 메모리 장치로부터 판독 기간 또는 메모리 장치에 데이터의 기입 기간을 포함할 수 있다. 제 2 파라미터는 논리 어드레스가능 유닛의 기입 동안 수행될 수 있는 가비지 수거의 최대이다. 두 파라미터들이 동일할 수 있지만 제 2 파라미터가 제 1 파라미터보다 더 클 수도 있을 것임이 가능하다. 두 파라미터들은 RAM에 저장될 수 있다. 도 14에 도시된 바와 같이, 가비지 수거에서 메모리 장치가 어느 모드에서 더 효율적인지를 판정하기 위해 두 파라미터들이 서로 비교된다. 대안적 실시예에서, 메모리 장치는 가비지 수거를 위해 어느 모드가 더 효율적인지를 나타내기 위해 다른 표시자를 포함할 수 있다. 예를 들면, 메모리 장치는 가비지 수거에서 어느 모드가 더 효율적인지를 나타내는 가비지 수거 모드 플래그를

포함할 수 있다. 이에 따라, 메모리 장치는 어느 모드가 더 효율적인지를 판정할 필요가 없고(이른테면 최대량에 액세스함으로써, 더 효율적인 모드를 판정하기 위해 간단히 플래그에 액세스할 수 있다.

[0060] 단계(1406)에서, 제 2 파라미터가 제 1 파라미터보다 더 크다면, 방법(1400)은 단계(1408)로 계속된다. 단계(1408)에서, 업데이트 블록의 가비지 수거는 제 2 파라미터의 량까지 초기화된다. 가비지 수거 동작은 가비지 수거량이 제 2 파라미터의 량에 도달할 때까지 단계들(1408, 1410)을 통해 수행된다. 제 2 파라미터가 단계(1406)에서 제 1 파라미터보다 크지 않다면, 업데이트 블록의 가비지 수거는 단계(1412)에서 제 1 파라미터의 량까지 초기화된다. 이 경우에, 가비지 수거 동작은 가비지 수거량이 제 1 파라미터의 량에 도달할 때까지 단계들(1412, 1414)을 통해 수행된다. 단일 논리 어드레스가능 유닛에의 기입동안 가비지 수거 동작 동안 카피되는 데이터량은 방법(1400)을 사용할 때 최대가 된다. 특히, 논리 어드레스가능 유닛이 업데이트 블록에 완전히 기입될지라도, 가비지 수거는 제 1 또는 제 2 파라미터에 따라 최대량에 도달될 때까지 업데이트 블록에 대해 계속된다.

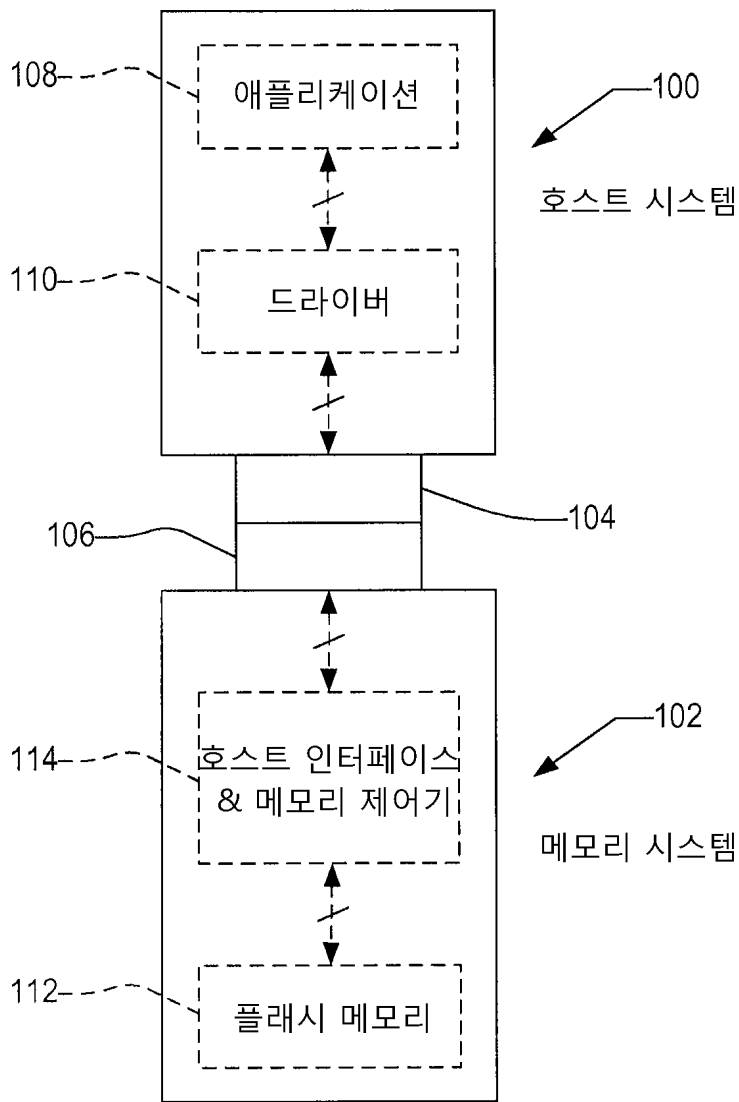
[0061] 앞에 상세한 설명은 한정이 아니라 예시적인 것으로 간주되어야 하고 이 발명의 정신 및 범위를 정의하는 것은 모든 등가물들을 포함해서 다음 청구항들임을 알아야 할 것이다.

부호의 설명

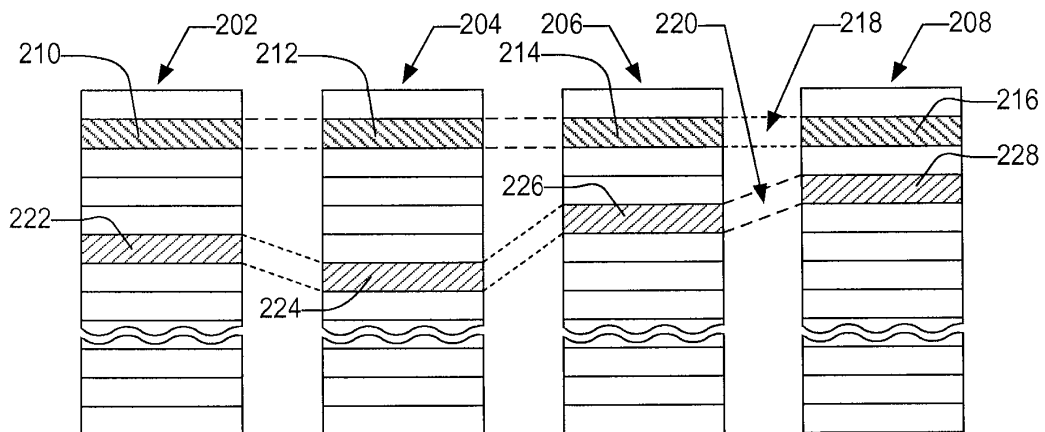
- [0062] 100: 호스트 시스템
- 102: 메모리 시스템
- 108: 애플리케이션
- 110: 구동기
- 112: 플래시 메모리
- 114: 호스트 인터페이스 & 메모리 제어기

도면

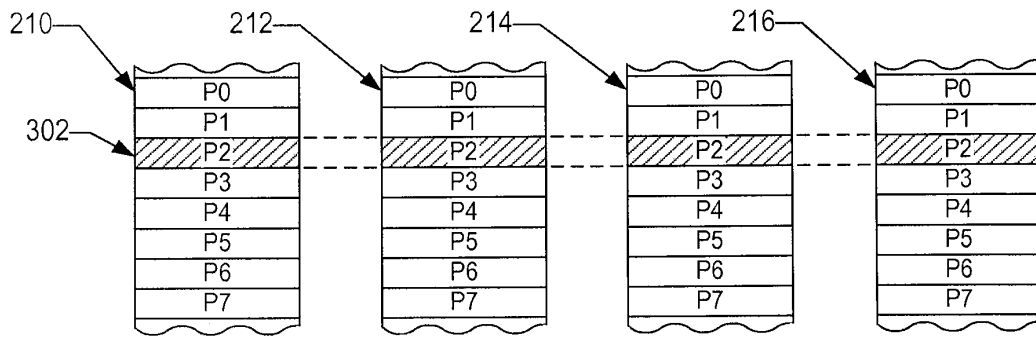
도면1



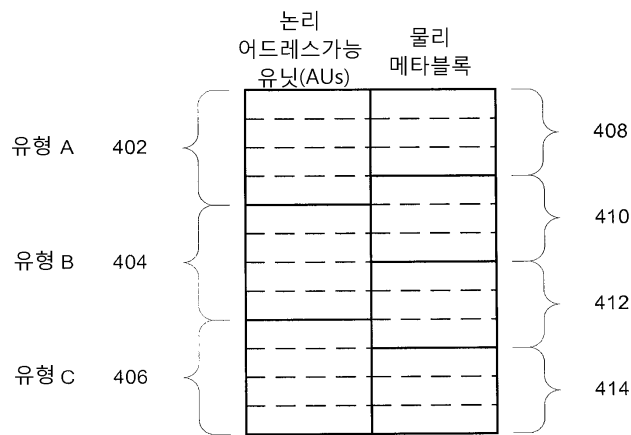
도면2



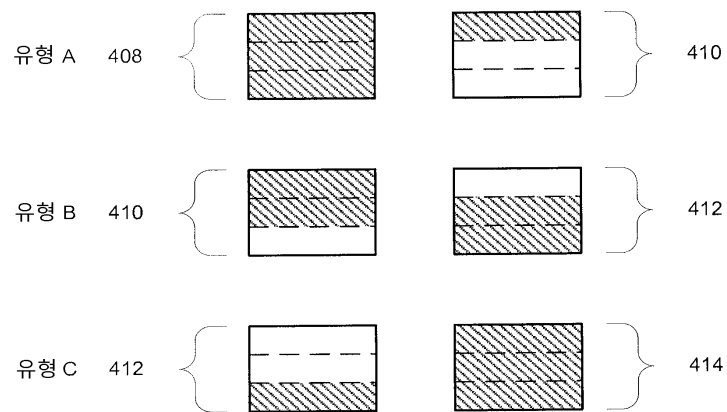
도면3



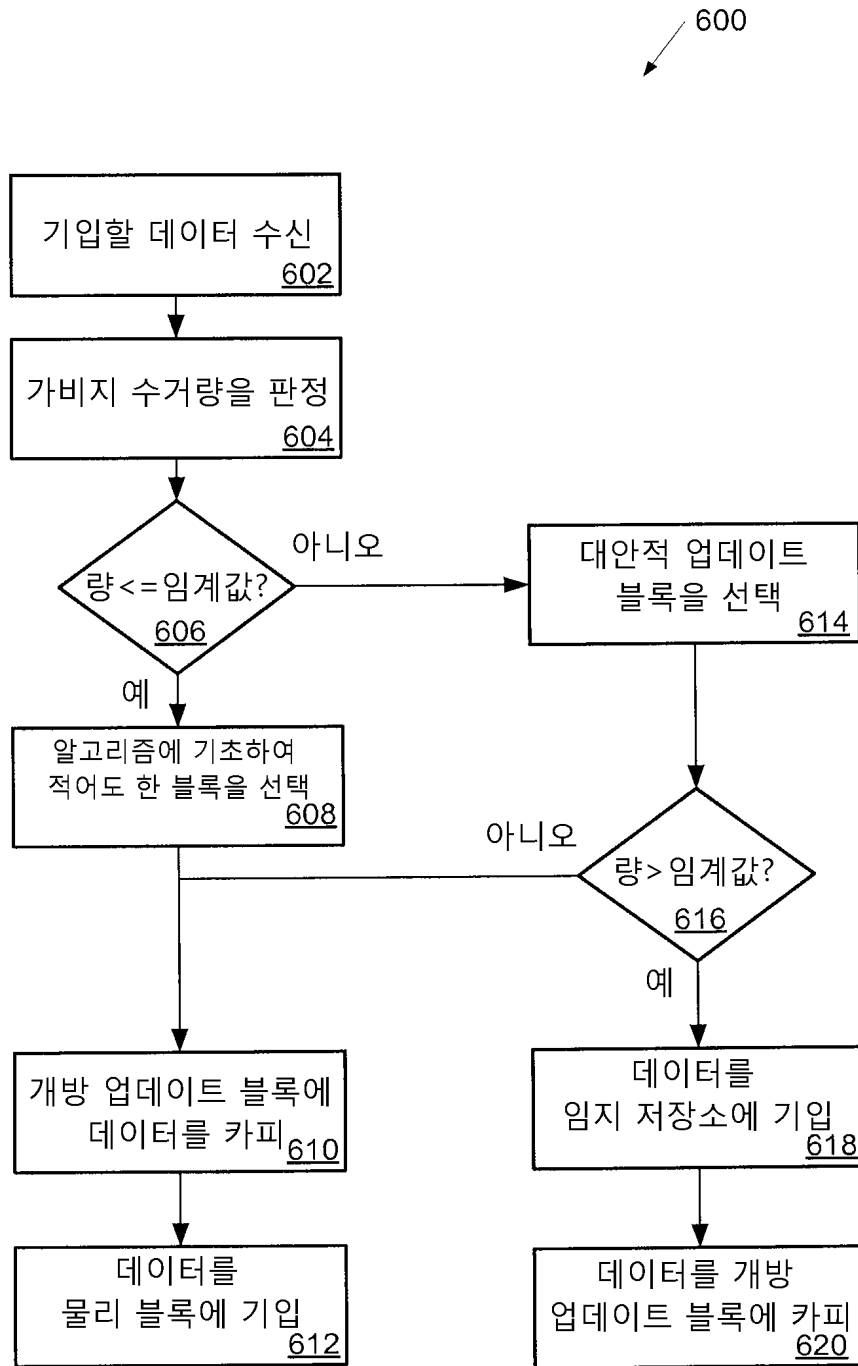
도면4



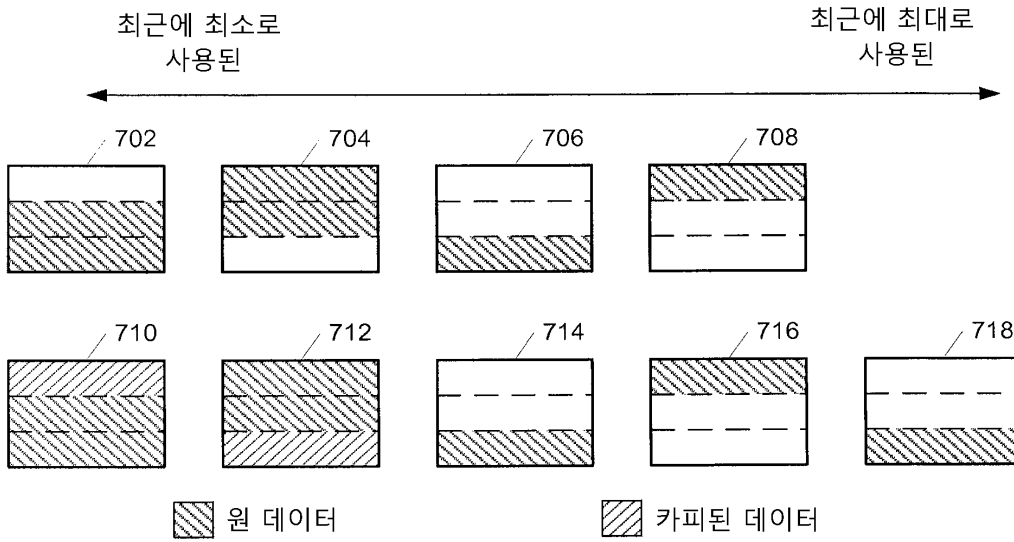
도면5



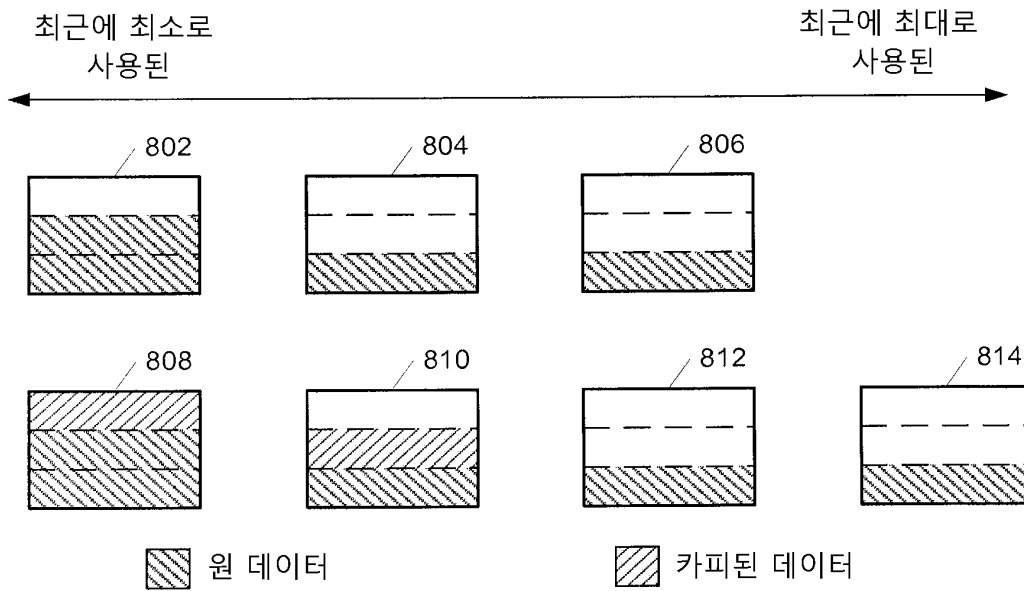
도면6



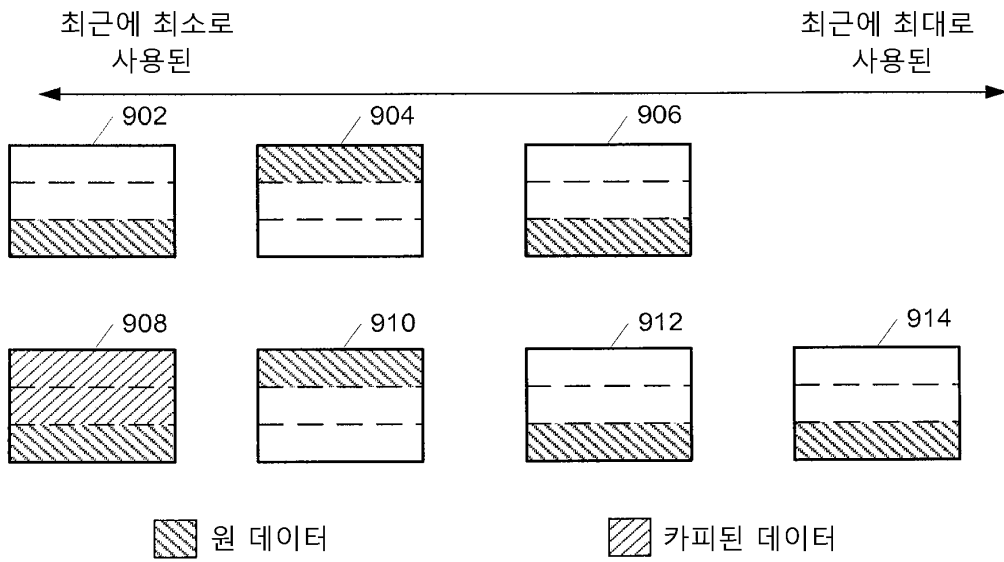
도면7



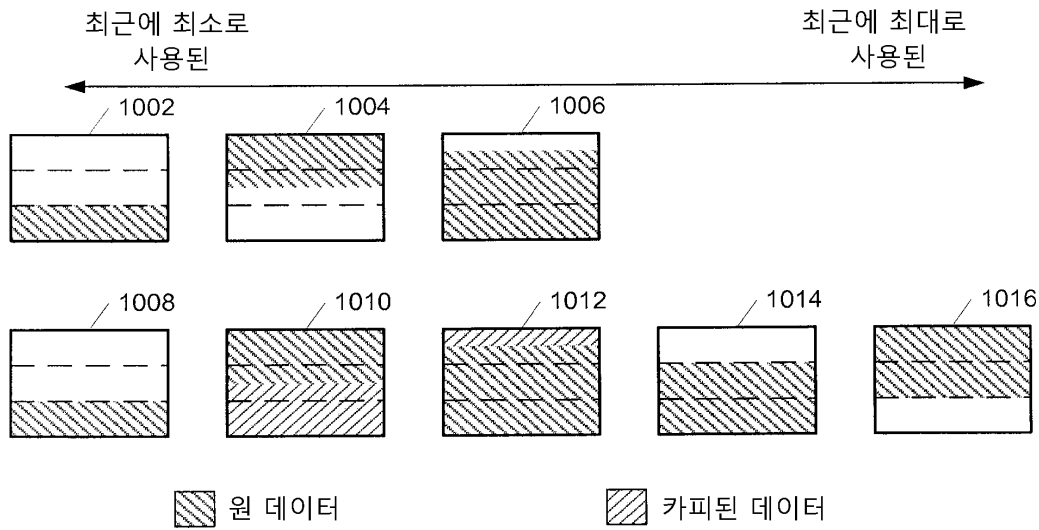
도면8



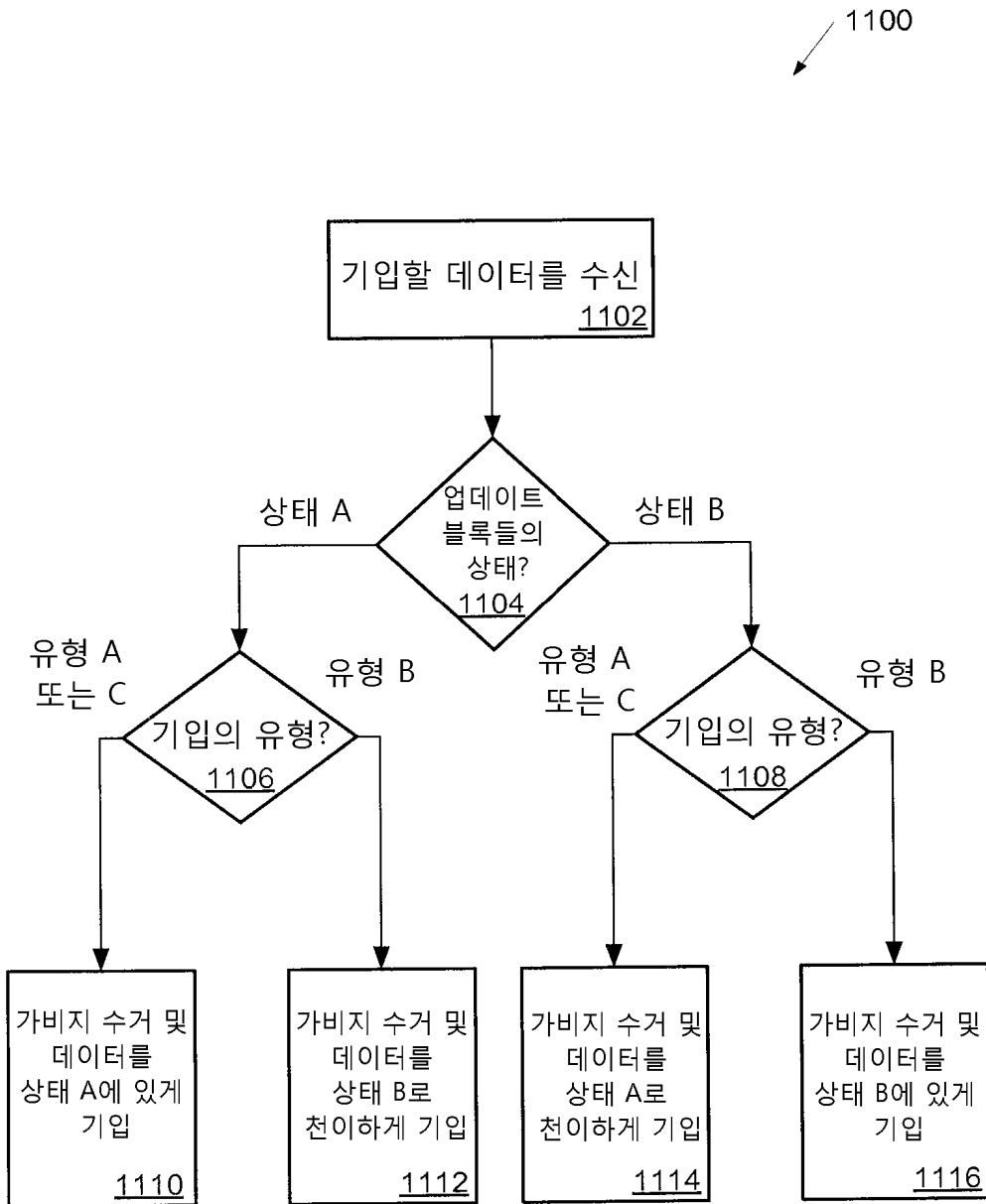
도면9



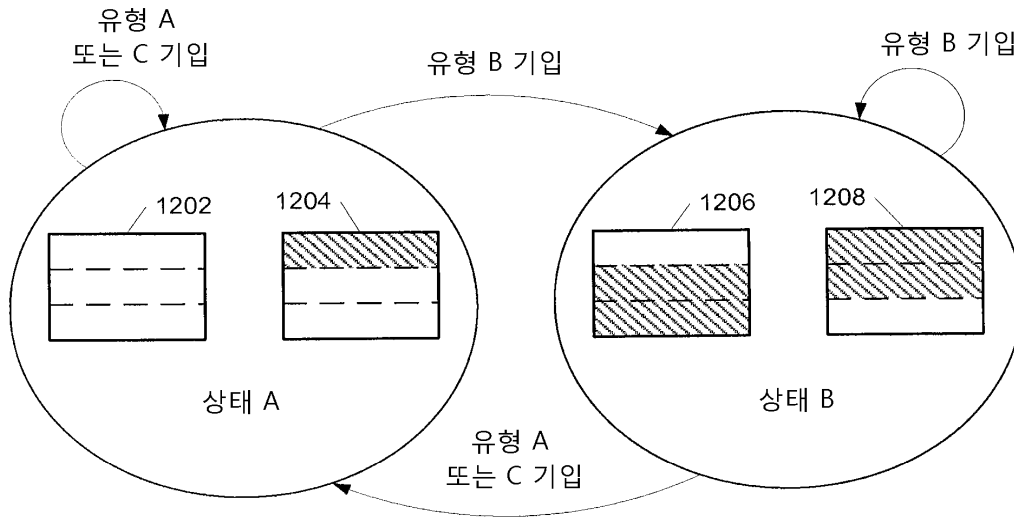
도면10



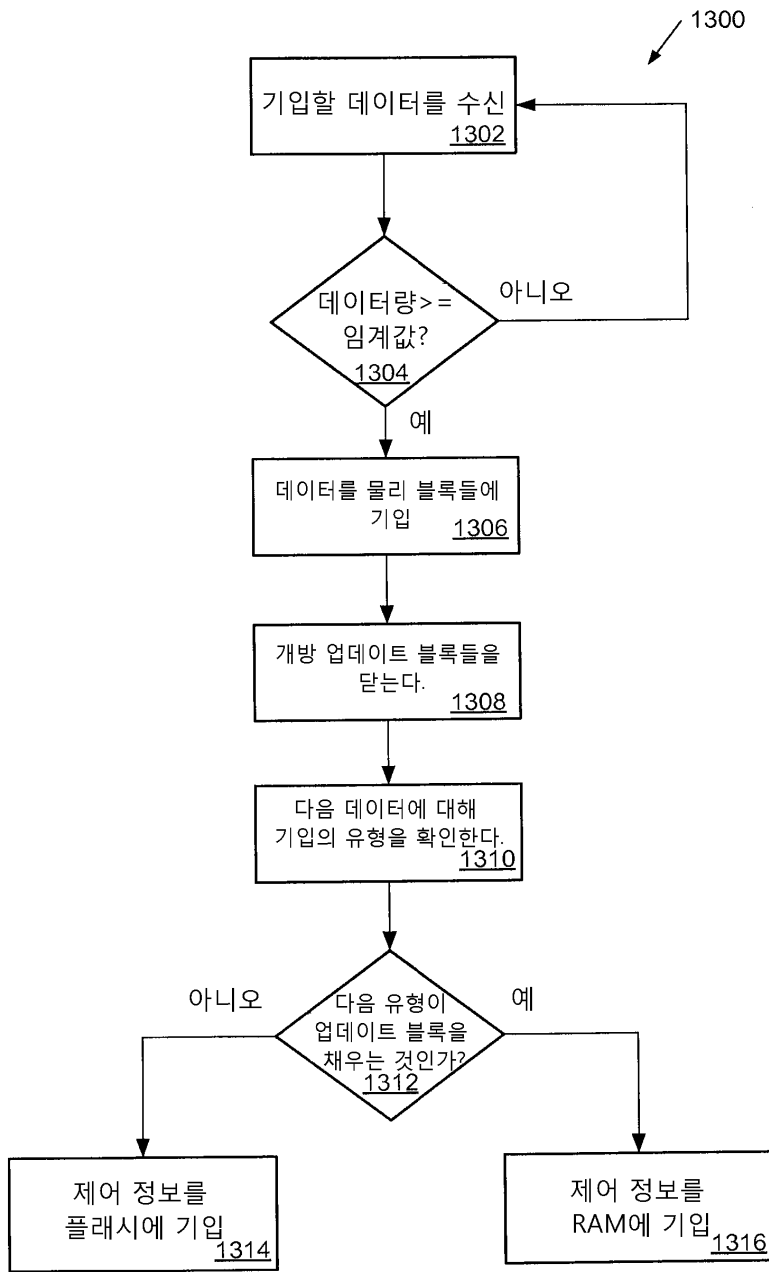
도면11



도면12



도면13



도면14

