



(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。 Int. Cl.

G06F 12/16 (2006.01)

G06F 12/00 (2006.01)

(11) 공개번호 10-2007-0015074

(43) 공개일자 2007년02월01일

(21) 출원번호 10-2006-0071459

(22) 출원일자 2006년07월28일

심사청구일자 없음

(30) 우선권주장 JP-P-2005-00221714 2005년07월29일 일본(JP)

JP-P-2006-00137844 2006년05월17일 일본(JP)

(71) 출원인 소니 가부시끼 가이샤
일본국 도쿄도 시나가와쿠 키타시나가와 6쵸메 7반 35고

(72) 발명자 니시하라 도시유키
일본국 도쿄도 시나가와쿠 키타시나가와 6쵸메 7반 35고 소니가부시끼
가이샤내

(74) 대리인 신관호

전체 청구항 수 : 총 30 항

(54) 기억장치, 컴퓨터 시스템, 및 기억 시스템

(57) 요약

고속의 병렬로 고속의 플래쉬 메모리 시스템을 구축하는 경우 오버헤드를 외관상 제거하고, 항시 고속의 액세스를 가능하게 하는 새로운 저장 형태의 구성을 실현시키는 기억장치, 즉, 주기억인 플래쉬 메모리와, 제어부를 구비하며, 상기 제어부는 갠신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며, 외부로부터 액세스가 없는 장치의 대기시에, 상기 무효화 영역을 자동적으로 빈 영역으로 회복시키는 회복 처리를 실시하는 기억장치이며, 이를 이용한 컴퓨터 시스템 및 기억 시스템이다.

대표도

도 1

특허청구의 범위

청구항 1.

기억장치에 있어서,

주기적인 플래쉬 메모리와,

제어부를 구비하며,

상기 제어부는 갱신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며,

외부로부터 액세스가 없는 장치의 대기시에, 상기 무효화 영역을 자동적으로 빈 영역에 회복시키는 회복 처리를 실시하는 기억장치.

청구항 2.

제 1항에 있어서,

상기 플래쉬 메모리는 블록 소거 단위인 소거 블록을 복수개 가지며, 각 소거 블록은 각각 독립한 기입 단위인 페이지 영역을 복수개 포함하며,

상기 제어부는, 상기 회복 처리로서, 회복 대상인 무효 페이지 영역을 포함한 소거 블록에 잔존하는 유효 페이지 영역의 데이터를, 다른 소거 블록의 소거가 종료된 빈 영역을 백업시키는 처리를 포함하는 기억장치.

청구항 3.

제 1항에 있어서,

상기 기억장치에 저장된 데이터에 대해 외부로부터 액세스가 생겼을 경우에, 상기 회복 처리를 실행중이면, 상기 제어부는 상기 회복 처리를 중단하고, 액세스에의 응답 처리를 실시하는 기억장치.

청구항 4.

제 3항에 있어서,

상기 회복 처리를 중단하는 경우, 상기 제어부는 상기 회복 처리의 진행 정보를 장치내의 레지스터 혹은 메모리에 보존하고, 상기 액세스가 종료하면 상기 진행 정보를 참조하여 회복 처리를 재개하는 기억장치.

청구항 5.

제 4항에 있어서,

상기 회복 처리의 진행 정보의 적어도 일부가, 불휘발성 RAM내에 기록되어 있는 기억장치.

청구항 6.

제 5항에 있어서,

상기 불휘발성 RAM은, 적어도 강유전체 메모리(FeRAM), 강자성체 메모리(MRAM), 상변화 메모리(ovonic unified memory : OUM), 자기 저항 효과 메모리(RRAM)중 최소한 어느 하나가 되는 기억장치.

청구항 7.

제 4항에 있어서,

상기 기억장치는, 내부에 처리 유닛을 가지며, 대기시에 상기 처리 유닛은 상기 회복 처리를 실시하고, 외부로부터 액세스가 생기면 상기 처리 유닛은 인터럽트되며, 상기 인터럽트 상황에 따라 상기 처리 유닛은 회복 처리를 중단하고, 액세스 처리를 개시하는 기억장치.

청구항 8.

제 2항에 있어서,

상기 제어부는, 하나의 소거 블록내에 존재하는 하나 또는 복수의 무효 페이지 영역의 회복 처리를, 복수의 공정으로 분할하고 외부로부터 액세스가 없는 기억장치의 대기시에, 그 일부를 실행하는 기억장치.

청구항 9.

기억장치에 있어서,

주기억인 플래쉬 메모리와,

제어부를 구비하며,

상기 제어부는 갱신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며,

외부로부터 수신된 소정의 명령에 따라, 상기 무효화 영역을 빈 영역에 회복하는 회복처리를 실행하는 기억장치.

청구항 10.

제 9항에 있어서,

상기 플래쉬 메모리는 블록 소거 단위인 소거 블록을 복수개 가지며, 각 소거 블록은 각각 독립한 기입 단위인 페이지 영역을 복수개 포함하며,

상기 제어부는, 상기 회복 처리로서, 회복 대상인 무효 페이지 영역을 포함한 소거 블록에 잔존하는 유효 페이지 영역의 데이터를, 다른 소거 블록의 소거가 종료된 빈 영역을 백업시키는 처리를 포함하는 기억장치.

청구항 11.

제 9항에 있어서,

상기 제어부는 외부로부터 수신된 소정의 명령에 따라, 상기 회복 처리를 중단할 때, 회복 처리의 진행 정보를 장치내의 레지스터 혹은 메모리에 보존하고, 재차 회복 처리의 실행 명령을 받으면, 상기 진행 정보를 참조하여 회복 처리를 재개하는 기억장치.

청구항 12.

제 11항에 있어서,

상기 회복 처리의 진행 정보의 적어도 일부가, 불휘발성 RAM내에 기록되어 있는 기억장치.

청구항 13.

제 12항에 있어서,

상기 불휘발성 RAM은, 적어도 강유전체 메모리(FeRAM), 강자성체 메모리(MRAM), 상변화 메모리(OUM), 자기 저항 효과 메모리(RRAM)중 최소한 어느 하나가 되는 기억장치.

청구항 14.

제 10항에 있어서,

상기 제어부는, 하나의 소거 블록내에 존재하는 하나 또는 복수의 무효 페이지 영역의 회복 처리를 복수의 공정으로 분할하고, 외부로부터의 소정의 명령에 따라 그 일부를 실행하는 기억장치.

청구항 15.

기억장치에 있어서,

주기억인 플래쉬 메모리와,

캐시 메모리와,

제어부를 구비하며,

상기 제어부는 갱신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며,

외부로부터의 액세스가 캐시 메모리에 도달하여, 캐시가 액세스되는 동안에, 상기 무효화 데이터를 빈 영역에 회복시키는 회복처리의 적어도 일부를 병행하여 실시하는 기억장치.

청구항 16.

제 15항에 있어서,

상기 플래쉬 메모리는 블록 소거 단위인 소거 블록을 복수개 가지며, 각 소거 블록은 각각 독립한 기입 단위인 페이지 영역을 복수개 포함하며,

상기 제어부는, 상기 회복 처리로서, 회복 대상인 무효 페이지 영역을 포함한 소거 블록에 잔존하는 유효 페이지 영역의 데이터를, 다른 소거 블록의 소거가 종료된 빈 영역을 백업시키는 처리를 포함하는 기억장치.

청구항 17.

제 15항에 있어서,

외부로부터의 액세스가 캐시 메모리를 지나쳤을 경우에, 상기 회복 처리를 실행중이면, 상기 회복 처리를 중단하고, 액세스에 대한 응답 처리를 실시하는 기억장치.

상기 제어부는 외부로부터 수신된 소정의 명령에 따라, 상기 회복 처리를 중단할 때, 회복 처리의 진행 정보를 장치내의 레지스터 혹은 메모리에 보존하고, 재차 회복 처리의 실행 명령을 받으면, 상기 진행 정보를 참조하여 회복 처리를 재개하는 기억장치.

청구항 18.

제 17항에 있어서,

상기 제어부는 외부로부터 수신된 소정의 명령에 따라, 상기 회복 처리를 중단할 때, 회복 처리의 진행 정보를 장치내의 레지스터 혹은 메모리에 보존하고, 재차 회복 처리의 실행 명령을 받으면, 상기 진행 정보를 참조하여 회복 처리를 재개하는 기억장치.

청구항 19.

제 17항에 있어서,

상기 회복 처리의 진행 정보의 적어도 일부가, 불휘발성 RAM내에 기록되어 있는 기억장치.

청구항 20.

제 18항에 있어서,

상기 불휘발성 RAM은, 적어도 강유전체 메모리(FerAM), 강자성체 메모리(MRAM), 상변화 메모리(OUM), 자기 저항 효과 메모리(RRAM)중 최소한 어느 하나가 되는 기억장치.

청구항 21.

제 20항에 있어서,

상기 제어부는, 하나의 소거 블록내에 존재하는 하나 또는 복수의 무효 페이지 영역의 회복 처리를 복수의 공정으로 분할하고, 외부로부터의 액세스가 캐시 메모리에 도달하여, 캐시가 액세스되는 동안에 그 일부를 실행하는 기억장치.

청구항 22.

컴퓨터 시스템에 있어서,

호스트 컴퓨터와,

주기적인 플래쉬 메모리를 포함하고, 상기 호스트 컴퓨터내에서 취급하는 여러 가지 데이터를 기억하기 위한 불휘발성 데이터 저장부를 가지며,

상기 호스트 컴퓨터는 갱신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며,

상기 데이터 저장부에 대해서 그외의 데이터 처리 요구가 없는 대기시에 상기 무효화 영역을 빈 영역에 회복시키는 처리를 실시하는 컴퓨터 시스템.

청구항 23.

제 22항에 있어서,

상기 플래쉬 메모리는 블록 소거 단위인 소거 블록을 복수개 가지며, 각 소거 블록은 각각 독립한 기입 단위인 페이지 영역을 복수개 포함하며,

상기 회복 처리는, 회복 대상인 무효 페이지 영역을 포함한 소거 블록에 잔존하는 유효 페이지 영역의 데이터를, 다른 소거 블록의 소거가 종료된 빈 영역을 백업시키는 처리를 포함하는 컴퓨터 시스템.

청구항 24.

제 22항에 있어서,

상기 호스트 컴퓨터는, 데이터 저장부에 대한 데이터 액세스의 요구가 생겼을 경우에, 상기 회복 처리를 실행중이면, 상기 회복 처리를 일시 중단하고, 일시 중단할 때, 회복 처리의 진행 정보를 컴퓨터 시스템 내부의 레지스터 또는 메모리내에 보관하고, 재차 회복 처리를 실행할 때 상기 진행 정보를 참조하여 처리를 재개 및 계속하는 컴퓨터 시스템.

청구항 25.

제 24항에 있어서,

상기 회복 처리의 진행 정보의 적어도 일부가, 불휘발성 RAM내에 기록되어 있는 컴퓨터 시스템.

청구항 26.

제 25항에 있어서,

상기 불휘발성 RAM은, 적어도 강유전체 메모리(FeRAM), 강자성체 메모리(MRAM), 상변화 메모리(OUM), 자기 저항 효과 메모리(RRAM)중 최소한 어느 하나가 되는 컴퓨터 시스템.

청구항 27.

제 23항에 있어서,

상기 호스트 컴퓨터는, 하나의 소거 블록내에 존재하는 하나 또는 복수의 무효 페이지 영역의 회복 처리를, 복수의 공정으로 분할하고 외부로부터 데이터 저장부에 대한 액세스가 없는 대기시에, 그 일부를 실행하는 컴퓨터 시스템.

청구항 28.

기억 시스템에 있어서,

주기억인 플래쉬 메모리와,

제어 회로를 구비하며,

상기 플래쉬 메모리는, 블록 소거 단위인 소거 블록을 가지며, 각 소거 블록은 복수의 데이터 영역을 포함하고,

상기 제어 회로는, 소거 블록의 소거를 실행하기 전에, 각 데이터 영역 마다, 적어도 영역의 일부를 판독하고, 상기 판독한 값으로부터 상기 데이터 영역내에 유효한 데이터가 기록되어 있는지를 판정하고, 판정의 결과, 유효한 데이터가 기록되어 있는 경우에, 상기 데이터 영역의 데이터를 다른 소거 블록에 복사하는 기억 시스템.

청구항 29.

제 28항에 있어서,

상기 제어 회로는, 상기 데이터 영역의 일부에 기재된 플래그치를 판독하고, 상기 플래그치(flag value)가 소정의 값이면 유효하다고 판정하는 기억 시스템.

청구항 30.

제 28항에 있어서,

상기 제어 회로는, 상기 데이터 영역의 일부에 기재된 핸들치를 판독하고, 상기 데이터의 메모리상의 물리 주소와 핸들치와의 일치성을 소정의 테이블과 대조하여, 일치하고 있는 경우에 유효하다고 판정하는 기억 시스템.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 2005년 7월 29일 일본 특허청에 제출된 2005-221714호와, 2006년 5월 17일 제출된 2006-137844호에 기술된 주제와 관련되며, 그 내용은 본 발명에 참조로서 포함되어 있다.

본 발명은, 불휘발성 메모리를 포함한 기억장치 및 그것을 갖춘 컴퓨터 시스템, 및 기억 시스템에 관한 것이며, 특히 플래쉬 메모리를 포함한 기억장치에 있어서의 데이터 전송의 고속화에 관련되는 것이다.

근년, 디지털 카메라나 모바일 컴퓨터 기기의 기억 매체로서 플래쉬 메모리가 주목받고 있다.

플래쉬 메모리는, 터널링이나 핫 일렉트론 가속을 이용하고, 전자에 게이트 절연막을 통과시켜, 그것들을 부유 게이트나 트랩층에 주입해, 셀 트랜지스터의 반응을 일으키는 최소의 물리량을 변화시키는 것으로서 데이터를 기억시키는 반도체 메모리이다. 적층 게이트 구조나 MNOS 구조등을 이용한 트랜지스터 1개만으로 메모리 셀을 구성할 수 있기 때문에, 염가이고 대용량의 메모리를 실현할 수 있다. 그 대표예로서 NAND형 플래쉬 메모리를 들 수 있다.

도 1은, NAND형 플래시 메모리의 내부 구성예를 나타내는 도면이다. 도 1의 NAND형 플래시 메모리는, 비트선(BL1~BLn)에 접속된 복수의 메모리유닛트(1-1~1-n)가 어레이 상태(중형)로 배열되어 있다. 예를 들어, 선택용 트랜지스터(2)의 게이트가 선택 게이트선(SL1)에 접속되며 선택용 트랜지스터(3)의 게이트가 선택 게이트선(SL2)에 접속되어 있다. 또한, 각 메모리 셀(N0~N15)의 게이트가 워드라인(WL0~WL15)에 접속되어 있다.

각 메모리 셀(N0~N15)은 적층 게이트 구조를 가지며, 부유 게이트내에 저장된 전하 축적량에 따라서 데이터를 기억한다. 즉, 부유 게이트에 많은 전자가 축적되어 있으면, 트랜지스터의 반응을 일으키는 최소의 물리량이 상승하므로, 충전된 비트선(BL1~BLn)으로부터 메모리 유닛트(1)(-1에서 -n까지)에 이르는 전류 관통이, 센스 증폭기등을 포함한 액세스 회로(4)에서 검출되어 데이터 판정을 가능하게 한다.

이러한 NAND형 플래시 메모리는, 메모리 셀마다 비트선에 콘택트 영역을 마련할 필요도 없기 때문에, 특히 대용량으로 염가의 기억장치의 매체에 적절하다.

일반적으로 플래시 메모리의 프로그램 속도는 매우 늦고, 셀 당 수백 μ 초를 필요로 한다. 또한 데이터의 오버라이팅(overwriting)을 할 수 없기 때문에, 프로그램에 앞서 데이터 소거를 실시할 필요가 있으므로, 이에 대해 수밀리(m) 초의 시간이 걸린다. 이러한 문제에 대해서는, 많은 메모리 셀을 병렬처리 하는 것으로 대처하고 있다.

즉, 예를 들어 동일 워드라인(WL0)에 접속되어 페이지 단위를 이루는 메모리 셀군(5)내에 동시에 블록 단위로 데이터를 기입하고, 서로 메모리 유닛트를 공유하는 페이지군으로 구성되는 셀 블록(6)을 블록 단위로 모두 소거함으로써 프로그램의 전송 속도를 향상시키고 있다.

구체적으로는, 예를 들어 ISSCC2002 프리프린트 p106, 세션 6.4에는, 1Gb의 NAND형 플래시 메모리가 게재되어 있으며, 페이지 사이즈를 2k 바이트, 소거 블록 사이즈를 128kB로 하고 있다. 즉, 하나의 메모리 어레이내에서 128k바이트의 메모리 셀군을 병렬 소거하고, 거기에 메모리 셀을 2k바이트마다 병렬로 프로그래밍함으로써, 10MB / s의 프로그램 전송 속도를 실현하고 있다.

또한, 근년 플래시 메모리는 더욱 멀티값(multi-value)을 가지도록 설계되고 미세화가 진행되고 있으므로, 거기에 따르는 신호량의 저하에 대처하기 위해, 비선택 셀의 악영향이 적은 기입 수법이 검토되어 실시되고 있다.

예를 들어 NAND형 플래시에 대해서는, 소거 블록내의 페이지의 기입순서에도 제한을 두는 것이 주류가 되고 있다. 일본 특개 2002-260390호 공보에는 그 하나로서 로컬 셀프 부스트(Local Self-Boost)로 불리는 수법을 이용한 기입 순서가 기재되어 있다. 도 1의 NAND형 플래시 메모리내의 그러한 기입의 예를 이하에 설명한다.

예를 들어, 메모리 셀(N1)에 기입을 실시하여, 그 부유 게이트에 전자를 주입하는 경우, 우선, 메모리 셀(N1)을 사이에 두는 전후의 워드라인(WL0, WL2)을 각각 0V로 하고, 선택용 트랜지스터(2)를 온으로 하고, 선택용 트랜지스터(3)를 오프로 한다. 여기서 비트선(BL1)을 0V로 하고, 기입을 실시하고 싶지 않은 인접 비트선(BL2)을 3V로 하고 선택 워드라인(WL1)을 20V로 상승시키고, 그 이외의 모든 워드라인(WL3~WL15)을 10V로 상승시킨다. 이러한 기입 순서를 이용하면, 0V의 워드라인(WL0, WL2)에 삽입되어 있는 비선택 비트선에 연결되는 노드는 다른 노드로부터 분리되며, 워드라인(WL1)으로부터의 커플링을 받아 10V정도까지 전위가 상승한다.

한편, 선택된 메모리 셀(N1)에 대해서는, 인접한 셀 트랜지스터(N0)가 디프레션(depression) 상태시에 한해서, 비트선(BL1)에 인가된 0V가 셀 트랜지스터(N1)의 채널에 전달되어 기입이 실시된다. 즉, 위와 같은 기입 수법을 이용하는 경우, 기입을 실시하는 셀의 비트선측의 인접 셀은 항상 소거되어 디프레션 상태로 된다. 그 때문에, 블록 전체를 소거한 후, 기입은 메모리 셀(N15, N14, N13,,,,, N0)의 순서대로 데이터를 기입하는 것이 필수적인 조건이 된다.

이와 같이 근년의 대용량 플래시 메모리는, 그 페이지 기입에 대해서도, 랜덤인 기입은 허용하지 않고, 블록중의 상위 주소로부터 하위 주소를 향해 차례차례 기입을 실시한다.

발명이 이루고자 하는 기술적 과제

근년, 하드 디스크의 소비 전력의 크기나, 탐색 시간의 길이, 내충격성이나 휴대성등의 문제를 해소하기 위하여, 플래시 메모리가 그 대체 메모리로서 기대되고 있다. 그러나 상술한 바와 같이, 플래시 메모리에는 액세스 단위를 크게 하지 않으면

고속화할 수 없다는 결점이 있다. 또한, 데이터의 오버라이팅을 할 수 없기 때문에, 재기입(rewriting)은 반드시 소거가 필요하고, 그 때의 소거 블록은 한층 더 크다. 액세스 단위에 대해서 소거 단위가 수십배 큰 것은, 소거 시간이 길고, 일반적인 플래쉬 메모리내의 기입 동작시에 비선택 셀에 혼란이 생기게 된다. 그러나 이에 의해 그 기입 효율은 현저하게 저하한다.

예를 들어 상기 2kB단위의 전송 속도 10MB / s 플래쉬 메모리 기입 페이지와, 128kB단위의 블록 소거를 이용하여, 하드 디스크 대체용의 기억장치를 구성한다고 가정한다. 그 전송 속도를, 직렬 ATA 접속의 고속 저장부내에서 목표로 여겨지는 160MB / s까지 높이려고 했을 경우, 멀티 뱅크나 다중 칩의 구성을 이용하여, 예를 들어 16개의 메모리 어레이를 병렬 동작시킬 필요가 있다.

도 2는, 그러한 기억장치를 구성하는 플래쉬 메모리의 개념도를 나타내는 도면이다. 도 2에서, 고속 전송을 실현하기 위해, 16개의 어레이 (AR0~AR15)를 동시에 동작시킨다. 이 경우 데이터 기입시는, 예를 들어 페이지(P0, P1)내에 데이터가 동시에 기입되고, 소거시는 블록(B0~B15)을 동시에 소거한다. 이 때, 블록 기입 단위가 되는 실제 페이지 영역(21)은 32kB, 블록 소거 단위가 되는 실제 소거 블록 영역(22)은 2MB에 이르게 된다.

한편, 통상의 파일 시스템에서는, 최소의 기입 단위를 이루는 클러스터 사이즈는 4kB정도이며, 이 단위로 랜덤 액세스가 실행된다. 게다가 통상의 파일 기억 시스템은, ATA등의 인터페이스 사양에 따라서, 예를 들어 512바이트의 섹터 단위로 랜덤 액세스가 실행된다. 그 경우, 예를 들어 페이지(P0와 P1)만을 고쳐 쓰는 요구가 빈번히 발생한다. 그러나, 상술한 바와 같은 장치로 그러한 액세스를 실시하면, 결국 실제 소거 블록 영역(22) 전체를 소거해야 한다. 그 중에서 비선택 영역에 유효 파일이 있으면, 그것을 소실로부터 보호할 필요가 있다. 그 전형적인 대처에는 다음과 같이 된다.

1. 우선 플래쉬 메모리로부터, 별도로 설치된 버퍼 메모리(23)의 메모리 영역(24)에, 실제 소거 블록 영역(22) 전체의 데이터를 읽어낸다.
2. 다음에 메모리 영역(24)내에서, 페이지(P0와 P1)에 해당하는 데이터를 갱신한다.
3. 다음에 플래쉬 메모리상의 블록 영역(22)의 소거를 실행한다.
4. 마지막으로 상기 소거 종료 영역(22)내에, 메모리 영역(24)의 갱신된 블록 데이터가 재기입된다.

즉, 4 k바이트의 기입을 위해서, 실제로는 2MB의 데이터의 소거와 읽기 및 기입이 필요하다. 구체적으로는, 1페이지 분의 데이터의 읽기 및 기입에는 각각 200마이크로초(μ s)를 필요로 하며, 블록의 소거에는 2밀리초(ms)를 필요로 하므로, 약 30밀리초(ms)를 필요로 한다.

이에 대해, 미리 예비 소거 블록(25)을 준비해 두고, 소거가 끝난 블록(25)에 영역(22)의 원래의 데이터와 페이지(P0와 P1)의 갱신된 데이터를 합성시키고 기입을 실시하는 수법도 있다. 이 경우, 가상 주소 구성을 이용해 소거 블록 단위로 논리 주소와 물리 주소의 대응을 갱신하고, 액세스 대상으로 여겨진 논리 주소에 대응하는 물리 블록은 원래의 소거 블록 영역 (22)으로부터 데이터가 이동되는 블록 영역(25)으로 변화된다.

이하, 가상 주소 구성을 이용하고, 예를 들어 소거 블록내의 1페이지만을 고쳐 쓸 때의 순서의 일례를, 도 3a, 도 3b를 이용해 구체적으로 설명한다. 도 3a는, 플래쉬 메모리(30)로 된 기억 영역을, 도 3b는 주소 테이블(40)을 나타내고 있다.

이런 종류의 수법으로 현재 가장 빈번히 사용되고 있는 것은, 소거 블록 단위로 어드레스 변환을 실시하는 방식이다라는 것에 주목하자.

주소 테이블(40)을 참조하고, 논리 블록 주소(Logical Block Address=LBA)를 인덱스로 이용함으로써, 소거 블록의 플래쉬 메모리상의 주소인 물리 블록 주소(Physical Block Address=PBA)를 취득할 수 있다. 예를 들어, 호스트 제어기 또는 애플리케이션이 "0x55"의 블록 주소에 대한 액세스를 요구했을 경우, 그 주소는 테이블에 의해 "0x6B"으로 변환되어 플래쉬 메모리(30)의 소거 블록(31)에 액세스가 실시된다. 이러한 관리하에 있어서의 플래쉬 메모리상의 데이터 갱신 순서는 이하와 같다.

1. 예비 소거 블록(32)을 미리 준비해 둔다.

그 물리 주소는 "0xAA"로 한다.

2. 논리 블록 주소 "0x55"가 지정되어, 대응하는 물리 블록 주소 "0x6B"가 취득되고, 블록(31)내의 페이지(33)에 액세스가 되었다고 가정한다.

이 때, 플래쉬 메모리(30)의 재기입 대상 블록(31)으로부터, 별도 설치된 페이지 버퍼(34)를 통해, 예비 블록(32)에 데이터를 차례차례 복사해 나간다.

3. 갱신 대상이 되는 페이지(33)의 복사시에는, 페이지 버퍼(34)상에서 소망하는 부분을 갱신하고 복사를 실시하여, 복사되는 페이지(35)에 그것이 반영된다.

4. 유효한 페이지 데이터의 복사가 모두 완료하면, 주소 테이블(40)상에서, 블록(31)과 예비 블록(32)의 주소가 서로 교환된다.

5. 마지막으로 원래의 블록(31)의 소거를 실시한다.

이러한 순서에 의해서, 이후 논리 블록 주소 "0x55"에 대해서는, 물리 블록 주소 "0xAA"에 대응하는 소거 블록, 즉, 예비 블록(32)이 액세스 되어, 전체의 인테그리티(integrity)가 유지된다.

그러나 이 경우, 1페이지를 갱신하기 위해서, 블록(31)의 데이터 전체를 예비 블록(32)에 백업시키는 작업이 필요하고, 그 때문에 블록(31)내의 전체 데이터를 페이지 버퍼(34)에 판독하고 예비 블록(32)에 기입하는 일이 필요하게 된다. 따라서, 이 경우도 실제의 전송 성능은 현저하게 낮은 것이 되어 버린다.

이러한 사정 때문에, 플래쉬 메모리를 이용한 현재의 기억장치는, 특별히 기입시에 있어서는, 하드 디스크보다 더욱 낮은 전송 성능밖에 얻지 못하고 있다.

또한, 소거 블록내의 기입 순서에도 제한이 가해진 경우, 문제는 한층 더 심각하게 된다. 일반적인 파일 시스템에 대해서는, 섹터간의 기입 순서의 보증은 이루어지지 않는다. 따라서, 소거 블록 도중부터 기입이 이루어지는 것이 자주 있지만, 그 경우, 블록내에 있어서의 그것보다 상위의 주소 영역이 비어 있어도, 거기에 데이터는 쓸 수 없게 된다. 그 결과, 다음에 그러한 영역에 기입 요구가 나오게되면, 재기입 같은 처리가 필요하게 되어, 시스템은 전술한 데이터 백업이나 소거, 기입을 반복한다.

이러한 문제는 단지 전송 성능을 저하시킬 뿐만 아니라, 플래쉬 메모리의 소거 회수를 증가시켜, 그 수명까지 저하시켜 버린다.

한편, 상술한 문제를 완화하는 데이터의 보존 수법으로서 주기형의 기억 시스템이 제안되고 있다. 이러한 시스템에 대해서는, 재기입, 갱신 데이터를 빈 영역에 한 번 기입하고, 데이터를 무효화함으로써 데이터를 재기입한다. 보다 구체적으로는, 논리 주소를 물리 주소에 대응시키는 어드레스 변환 테이블을 이용하며, 관련 데이터의 물리 주소를 변경하고, 저장 매체의 빈영역에 한 번 기입함으로써 데이터를 재기입한다.

예를 들어, 특허 문헌 2(특개평 8-328762호 공보)에는, 어드레스 변환 테이블을 이용한 주기형 기억 시스템에 있어서의 관리 방법의 상세한 점들이 기재되어 있다.

도 4a, 4b는, 어드레스 변환 테이블을 이용한 주기형 기억 시스템에 있어서의 관리 방법의 일례를 설명하기 위한 도면이다. 도 4a는 플래쉬 메모리(50)는 기억 영역을, 도 4b는 주소 테이블(60)을 나타내고 있다.

주소 테이블(60)을 참조하고, 논리 페이지 주소(Logical Page Address=LPA)를 인덱스로 이용함으로써, 플래쉬 메모리상의 대응하는 페이지의 주소인 물리 페이지 주소(Physical Page Address=PPA)를 취득할 수 있다. 예를 들어, 호스트로부터 지정된 논리 페이지 주소 "0x5502"에 대한 기입을 위해, 주소 테이블을 이용해 페이지 단위로 어드레스 변환을 실시하고, 플래쉬 메모리(50)상의 물리 페이지 주소 "0x6B"를 취득한다. 이에 의해 블록(51)내의 대응하는 페이지 영역(53)에 액세스가 실시된다. 한편, 동일 페이지에 갱신을 실시할 때, 플래쉬 메모리(50)내에서 데이터를 직접 쓸 수 있는 적당한 빈 페이지 영역이 검색된다. 예를 들어 물리 블록 주소 "0xAA"에 상당하는 소거된 빈 블록(52)의 선두 페이지 영역(55)이 적절한 기입처로서 선택되었을 경우, 페이지 영역(53)의 데이터만이 페이지 버퍼(34A)를 통해 갱신되어 페이지 영역(55)에 기입된다. 이 때, 논리 페이지 주소 "0x5502"는 페이지 영역(55)의 물리 주소 "0xAA00"에 리매핑된다. 페이지 영역(53)상의 구데이터(old data)는, 그대로 남겨지게 되어, 무효화된다.

이러한 관리를 실시하면, 플래쉬 메모리내에 빈 영역이 존재하는 한, 각 페이지 데이터의 갱신에 대해서 1페이지 분의 데이터를 기입하는 것으로 충분하다. 따라서, 고속으로 재기입을 실시할 수 있다. 그 사이 데이터를 소거할 필요가 없기 때문에, 플래쉬 메모리의 재기입 회수도 큰 폭으로 저감할 수 있어 그 수명도 향상시킬 수 있다.

그러나, 이러한 주기형 기억 시스템에서는, 무효화한 원래의 데이터 영역이 소용없게 된다. 따라서, 그러한 영역은 소거하고, 재차 기입이 가능해지도록 회복 처리를 실시하는 것이 필요하다. 그러나 그 경우, 특히 소거 블록이 복수의 페이지 영역을 포함하면, 무효화된 페이지 영역과 같은 소거 블록내에 유효 데이터를 포함한 페이지 영역이 존재하는 경우가 많이 발생한다. 이러한 블록을 회복시키려면, 소거전에 유효 페이지만을, 다른 블록에 백업시킬 필요가 있다. 한편, 이러한 각 페이지의 유효 / 무효의 정보를 플래그에 의해서 RAM상에 보관했을 경우, 플래시 장치의 용량 증가에 따라, 플래그 보관에 필요한 RAM 용량이 현저하게 증가해 버린다.

본 발명의 목적은, 고속의 병렬로 고속의 플래쉬 메모리 시스템을 구축하는 경우 오버헤드를 외관상 제거하고, 항시 고속의 액세스를 가능하게 하는 새로운 저장 형태의 구성을 실현시키는 기억장치, 컴퓨터 시스템, 및 기억 시스템을 제공하는 것에 있다.

또한, 본 발명의 목적은, 특히 주기형 기억 시스템에 있어서 블록 소거를 실시할 때, RAM등의 하드웨어 자원을 가능한 한 사용하지 않고, 유효 페이지와 무효 페이지를 식별 판정하여, 유효 페이지만을 다른 블록에 신속하게 백업시키는 것이 가능한 기억 시스템을 제공하는 데에 있다.

본 발명의 제 1의 관점의 기억장치는, 주기억인 플래쉬 메모리와, 제어부를 구비하며, 상기 제어부는 갱신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며, 외부로부터 액세스가 없는 장치의 대기시에, 상기 무효화 영역을 자동적으로 빈 영역에 회복시키는 회복 처리를 실시한다.

바람직하게는, 상기 플래쉬 메모리는 블록 소거 단위인 소거 블록을 복수개 가지며, 각 소거 블록은 각각 독립한 기입 단위인 페이지 영역을 복수개 포함하며, 상기 제어부는, 상기 회복 처리로서, 회복 대상인 무효 페이지 영역을 포함한 소거 블록에 잔존하는 유효 페이지 영역의 데이터를, 다른 소거 블록의 소거가 종료된 빈 영역에 백업시키는 처리를 포함하는 기억장치.

바람직하게는, 상기 기억장치에 저장된 데이터에 대해 외부로부터 액세스가 생겼을 경우에, 상기 회복 처리를 실행중이면, 상기 제어부는 상기 회복 처리를 중단하고, 액세스에의 응답 처리를 실시한다.

바람직하게는, 상기 회복 처리를 중단하는 경우, 상기 제어부는 상기 회복 처리의 진행 정보를 장치내의 레지스터 혹은 메모리에 보존하고, 상기 액세스가 종료하면 상기 진행 정보를 참조하여 회복 처리를 재개한다.

바람직하게는, 상기 회복 처리의 진행 정보의 적어도 일부가, 불휘발성 RAM내에 기록되어 있다.

바람직하게는, 상기 불휘발성 RAM은, 적어도 강유전체 메모리(FerAM), 강자성체 메모리(MRAM), 상변화 메모리(ovonic unified memory : OUM), 자기 저항 효과 메모리(RRAM)중 최소한 어느 하나가 된다.

바람직하게는, 상기 기억장치는, 내부에 처리 유닛을 가지며, 대기시에 상기 처리 유닛은 상기 회복 처리를 실시하고, 외부로부터 액세스가 생기면 상기 처리 유닛은 인터럽트되며, 상기 인터럽트 상황에 따라 상기 처리 유닛은 회복 처리를 중단하고, 액세스 처리를 개시한다.

바람직하게는, 상기 제어부는, 하나의 소거 블록내에 존재하는 하나 또는 복수의 무효 페이지 영역의 회복 처리를, 복수의 공정으로 분할하고 외부로부터 액세스가 없는 기억장치의 대기시에, 그 일부를 실행한다.

본 발명의 제 2의 관점에 따르는 기억장치는, 주기억인 플래쉬 메모리와, 제어부를 구비하며, 상기 제어부는 상기 제어부는 갱신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며, 외부로부터 수신된 소정의 명령에 따라, 상기 무효화 영역을 빈 영역으로 회복하는 회복처리를 실행한다.

본 발명의 제 3의 관점에 따르는 기억 장치는, 주기억인 플래쉬 메모리와, 캐시 메모리와, 제어부를 구비하며, 상기 제어부는 상기 제어부는 갱신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며, 외부로부터의 액세스가 캐시 메모리에 도달하여, 캐시가 액세스되는 동안에, 상기 무효화 데이터를 빈 영역에 회복시키는 회복처리의 적어도 일부를 병행하여 실시한다.

바람직하게는, 상기 플래쉬 메모리는 블록 소거 단위인 소거 블록을 복수개 가지며, 각 소거 블록은 각각 독립한 기입 단위인 페이지 영역을 복수개 포함하며, 상기 회복 처리는, 다른 소거 블록의 소거가 종료된 빈 영역으로 회복되어지는 무효 페이지 영역을 포함한 소거 블록에 잔존하는 유효 페이지 영역의 데이터를, 백업시키는 처리를 포함한다.

바람직하게는, 외부로부터의 액세스가 백업 메모리를 지나쳤을 경우에, 상기 회복 처리를 실행중이면, 상기 회복 처리를 중단하고, 액세스에 대한 응답 처리를 실시한다.

본 발명의 제 4의 관점에 따르는 컴퓨터 시스템은, 호스트 컴퓨터와, 주기억인 플래쉬 메모리를 포함하고, 상기 호스트 컴퓨터내에서 취급하는 여러 가지 데이터를 기억하기 위한 불휘발성 데이터 저장부를 가지며, 상기 호스트 컴퓨터는 갱신 데이터를 빈 영역에 추가 기입하고, 원래의 데이터를 무효화함으로써 상기 플래쉬 메모리에 있어서의 적어도 일부 영역을 재기입하며, 상기 데이터 저장부에 대해서 그외의 데이터 처리 요구가 없는 대기시에 상기 무효화 영역을 빈 영역에 회복시키는 처리를 실시한다.

바람직하게는, 상기 호스트 컴퓨터는, 데이터 저장부에 대한 데이터 액세스의 요구가 생겼을 경우에, 상기 회복 처리를 실행중이면, 상기 회복 처리를 일시 중단하고, 일시 중단할 때, 회복 처리의 진행 정보를 컴퓨터 시스템 내부의 레지스터 또는 메모리내에 보관하고, 재차 회복 처리를 실행할 때 상기 진행 정보를 참조하여 처리를 재개 및 계속한다.

본 발명의 제 5의 관점에 따르는 기억 시스템은, 주기억인 플래쉬 메모리와, 제어 회로를 구비하며, 상기 플래쉬 메모리는, 블록 소거 단위인 소거 블록을 가지며, 각 소거 블록은 복수의 데이터 영역을 포함하고, 상기 제어 회로는, 소거 블록의 소거를 실행하기 전에, 각 데이터 영역 마다, 적어도 영역의 일부를 판독하고, 상기 판독한 값으로부터 상기 데이터 영역내에 유효한 데이터가 기록되어 있는 지를 판정하고, 판정의 결과, 유효한 데이터가 기록되어 있는 경우에, 상기 데이터 영역의 데이터를 다른 소거 블록에 복사한다.

바람직하게는, 상기 제어 회로는, 상기 데이터 영역의 일부에 기재된 플래그치를 판독하고, 상기 플래그치가 소정의 값이면 유효하다고 판정한다.

바람직하게는, 상기 제어 회로는, 상기 데이터 영역의 일부에 기재된 핸들치를 판독하고, 상기 데이터의 메모리상의 물리 주소와 핸들치와의 일치성을 소정의 테이블과 대조하여, 일치하고 있는 경우에 유효하다고 판정한다.

발명의 구성

이하, 본 발명의 실시 형태를 도면을 참조하여 설명한다.

도 5는, 본 발명의 제 1의 실시 형태와 관련되는 기억장치를 나타내는 구성도이다. 도 6a, 도 6b는, 제 1의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블 및 페이지 상태의 관리 테이블의 구성예를 나타내는 도면이다.

도 5의 기억장치(70)는, 인터페이스 회로(I/F)(71), 페이지 버퍼(72), 내부 데이터 버스(73), 제어 회로(74), 및 NAND형 플래쉬 메모리(75, 76)를, 주구성 요소로 가지고 있다.

기억장치(70)에서는, 32비트의 내부 데이터 버스(73)에, 16비트의 입출력을 가지는 2개의 칩의 NAND형 플래쉬 메모리(75, 76)가 병렬 접속되어 있다. 2개의 칩의 플래쉬 메모리(75, 76)는 판독 또는 기입시에, 동시 병렬로 액세스 된다. 페이지 버퍼(72)는, 액세스된 페이지 영역을 일시 기억하는 버퍼이며, 외부와 인터페이스 회로(71)를 통해 접속되어 있다. 제어 회로(74)는, 메모리 칩인 플래쉬 메모리(75, 76)와 페이지 버퍼(72) 및 인터페이스 회로(71) 사이에서의 데이터의 교환을 관리하는 제어기이며, 처리 유닛인 내장 CPU(85)에 의해서 제어된다. 유사한 내장된 RAM(77)에는, CPU(85)를 제어하기 위한 프로그램 영역이나 작업영역이 확보되며 그와 동시에, 어드레스 변환 테이블(78), 및 페이지 상태의 관리 테이블(86)이 구축되어 있다.

기억장치(70)의 액세스는 이하와 같이 된다. 여기에서는 간단히, 16진수의 주소는 다음과 같이 할당된다고 가정한다. 예를 들어 외부 입력 주소가 "0x5500C"일 경우, 상위의 "0x5500"는 페이지 주소, 하위 "0xC"는 페이지 영역내의 섹터 어드레스이며, 기억장치(70)는 1섹터 단위로 랜덤 액세스가 가능하다.

판독 시에, 외부로부터 "0x5500C"의 섹터 어드레스가 입력되면, 제어 회로(74)는 상위의 페이지 주소를 받아 내장 RAM(77)에 액세스하고, 어드레스 변환 테이블(78)로부터 논리 페이지 주소(LA) "0x5500"에 대응하는 물리 페이지 주소(PA) ?0x00B0?을 취득한다. 이 물리 페이지 주소 (PA)중, 상위의 "0x00"는 메모리 칩 플래쉬 메모리(75, 76)내의 소거 블록(83, 84)의 주소이다. 하위 "0xB0"는 각 소거 블록내의 페이지 영역(79, 80)의 주소이다. 이 물리 주소(PA)를 기초로 메모리 칩의 플래쉬 메모리(75, 76)가 모두 액세스되어, 각각으로부터 페이지 주소 "0x00B0"에 저장된 페이지 영역(79, 80)의 데이터가 판독되어 페이지 버퍼(72)에 저장된다. 인터페이스 회로(71)는 그 중에서, 하위 주소 "0xC"에 해당하는 섹터를 선택하여, 외부에 출력한다.

한편, 외부로부터 "0x5500C"에 기입을 실시하는 경우는, 우선 판독과 같은 순서로 페이지 영역(79, 80)의 데이터가 메모리 칩 플래쉬 메모리(75, 76)로부터 판독되어 페이지 버퍼(72)에 저장된다. 그 후 하위 주소 "0xC"에 해당하는 섹터가 선택되어 외부로부터 입력된 데이터로 갱신된다.

갱신된 페이지 데이터는, 플래쉬 메모리(75, 76)에 다시 기입되며, 그 때 판독측의 페이지 영역(79, 80)은 재기입이 안되며, 빈 영역(81, 82)에 데이터가 기입된다. 이 페이지 영역의 물리 페이지 주소는 "0x0280"는, 즉 소거 블록 "0x02"안의 "0x80"의 페이지에 상당한다. 또한 이 처리 전까지, 그 영역은 어드레스 변환 테이블(78)의 어느 물리 주소 필드에도 등록되지 않은, 미사용의 빈 페이지였다. 갱신된 데이터가 이 영역에 주기형 형태로 기입되며 동시에, 어드레스 변환 테이블(78)의 논리 주소 "0x5500"에 대응하는 물리 주소 필드에는, 빈 영역(81, 82)의 물리 페이지 주소 "0x0280"이 등록된다.

이와 같이 갱신 데이터를 한 번 기입하기 위한 빈 영역의 관리는, 예를 들어 도 6에 도시한 페이지 상태 관리 테이블(86)을 RAM(77)상에 구축해 사용한다. 테이블은(BA : Block Address)로 표현되는 각 소거 블록에 저장된 각각의 페이지 상태를 2비트의 플래그열로 기록하고 있다. "0"은 빈영역이며, "1"은 기입이 행해진 유효 영역, "2"는 데이터가 갱신되어 무효화된 무효 영역을 나타낸다.

도 6b의 상태에서는, 블록 주소 "0x02"에 있어서의 물리 페이지 "0x80"이후가 모두 빈 상태가 되어 있다. 예를 들어, 추가 재기입을 실행하기 위한 빈 영역을 결정하기 위해서, 현재 검색 대상으로 하고 있는 블록 "0x02"에 대한 포인터를 메모리 영역(87)에 보존해 둔다. 플래시 메모리에 기입요청이 있는 경우, 상위 페이지로부터 이 테이블의 플래그를 스캔 하여 빈 페이지 영역을 검색한다. 블록의 모든 빈 영역이 사용되고 있으면, 포인터를 증가시키고, 다음은 ?0x03?의 블록으로부터 빈 페이지 영역을 찾는다. 이와 같이 포인터를 블록 주소의 선두로부터 말미에 걸쳐 순환시켜 가면, 플래쉬 메모리 전역에 대해서 균일한 기입을 실시할 수 있다.

본 구성에서는 두 개의 메모리 칩인 플래쉬 메모리(75, 76)가 병렬로 배치되고, 각각으로부터 판독된 페이지 데이터가, 페이지 버퍼(72)에 일괄 저장된다. 이 때 단독 칩을 사용했을 경우에 비해 메모리 용량은 2배가 되지만, 이것은 블록 액세스 단위이며 또한 2배가 되는 어드레스 변환 테이블(78)의 관리 단위가 되는, 한 페이지 영역의 크기에 의해 실현되므로, 페이지 수는 증가하지 않는다. 따라서 어드레스 변환 테이블(78)은 변화하지 않고, 그것을 저장하는 메모리(RAM)(77)의 용량도 증가하지 않는다. 한편, 병렬화의 약점인 소거 회수의 증가나, 재기입 피로(fatigue)가 가속화하는 것은 페이지 단위의 어드레스 변환을 이용한 주기형의 기입에 의해 큰 폭으로 완화된다.

예를 들어 대형의 파일 재기입, 또는 소거 블록 전체를 고쳐 쓰는 경우에는, 유효 데이터의 백업 처리는 원래 불필요하고, 블록 소거만으로 충분하다. 따라서, 이러한 경우에는 상기 주기형 기입의 채택에 의해 큰 성능 차이는 생기지 않는다. 따라서, 일반적인 통상형 재기입 동작과 주기형 재기입 동작은 경우에 따라 조합해도 좋다.

그런데, 상술한 주기형 기입을 실시했을 경우, 갱신전의 데이터가 저장되어 있던 물리 페이지 주소 "0x00B0"에 해당하는 페이지 영역(79, 80)은 어드레스 변환 테이블(78)의 물리 주소 필드로부터 삭제되어 밖으로부터 액세스 할 수 없게 된다. 즉 무효화된다. 그러나 그들에 데이터가 기입되므로, 그대로 빈 영역으로서 사용할 수 없다. 상술한 바와 같이 재기입이 몇 번이나 반복되면, 많은 무효 페이지 영역이 발생한다. 그것들은 재차 빈 영역으로서 사용할 수 있도록, 소거하여 회복시킬 필요가 있다. 또한 그 경우, 소거 블록(83, 84)에 남겨진 유효 데이터는 백업시키지 않으면 안된다.

이러한 회복 처리에서는, 예를 들어 우선 대상 블록내의 유효 데이터를, 갱신때와 같이 일단 페이지 버퍼에 판독하고, 예비 페이지 영역에 주기형 동작에 의해 기입하고 나면 실질적인 백업을 하게 된다. 즉, 유효 페이지를 임시적으로 갱신하여, 그 원래 영역을 모두 무효화한다. 그 후 대상 블록을 소거하여, 회복 처리가 실시된다.

도 7a~도 7c는, 무효 영역의 회복 순서를 설명하기 위한 도면이다. 도 7a~도 7c에는, 소거 블록(83, 84) 내부의 유효 데이터를 백업하고, 무효 페이지 영역을 실질적으로 회복시키는, 구체적 순서 <1>, <2>, <3>을 예시하고 있다.

순서 <1>

도 6a와, 도 6b의 상태에서부터 주기형 동작에 의한 재기입이 진행되고, 소거 블록(83, 84)에는, 한 번 데이터가 써진 후, 갱신에 의해서 무효화된 페이지 영역군(94, 96)이, 유효한 페이지 영역군(93, 95, 97)과 공존하고 있다고 가정한다. 여기서 유효 페이지 영역의 데이터를 남기면서, 무효화 영역을 빈 영역으로 회복시킬 필요가 있다. 한편, 블록(91, 92)은 현재 주기형 동작의 빈 영역으로 사용되는 소거 블록이며, 페이지 영역(102)까지 기입을 하고 있다.

순서 <2>

유효한 페이지 영역군(93, 95, 97)을 소거 블록(91, 92)내의 빈 영역 (98, 99, 100)에 상부로부터 차례로 복사해 나간다. 이 때 예를 들어 도 5의 페이지 버퍼(72)에 소거 블록(83, 84)으로부터 차례차례 1페이지 영역씩을 판독하고, 소거 블록 (91, 92)에 기입하는 동시에, 어드레스 변환 테이블(78)을 갱신한다. 즉 각각의 페이지 영역의 논리 주소에 대응하는 물리 주소 필드에, 복사처(copy destination)의 물리 페이지 주소를 등록해 나간다. 이 조작은, 즉 유효 페이지 영역군(93, 95, 97)을 주기형 방법에 의해 재기입하는 동작과 동일하다. 실제로는 고쳐 쓰지 않고 복사할 뿐이지만, 이 작업에 의해서 소거 블록(83 및 84)내의 모든 페이지는 무효화되고, 유효 페이지 영역의 데이터는 소거 블록(91, 92)에 실질적으로 백업된다.

순서 <3>

소거 블록(83, 84)을 소거한다. 이에 의해서 그 내부는 모두 빈 영역이 되어, 나중의 주기형 동작을 위해 사용하는 것이 가능하게 된다. 이에 의해 무효 영역(94, 96)은 실효적으로 회복된다.

이와 같이 무효 페이지 영역의 회복 처리는, 각 유효 페이지 영역의 복사에 의한 백업 처리와, 원래 소거 블록의 소거로 구성된다. 또한, 회복 대상이 되는 소거 블록내의 유효 페이지에 통상의 갱신 처리와 같은 순서로 더미 갱신(dummy updating)을 실행하여, 그것을 가지고 페이지의 백업을 실행하면, 회복 처리시의 유효 페이지 백업을 통상의 기입 알고리즘에 통합시키는 것이 가능하다. 이 경우, 제어가 용이하게 되며, 플래쉬 메모리에 대한 기입의 균일화등, 기입시의 신뢰성 향상을 위한 각종 대책을 회복 처리에도 적용할 수 있어 기억장치의 종합적인 신뢰성을 향상시킬 수 있다.

여기에서, 본 발명의 제 1의 실시 형태는, 상기 플래쉬 메모리에 대한 주기형 기입을 채용한 반도체 기억장치의 내부에, 그 대기시에 무효화된 영역의 회복 처리를 자동적으로 실시하는 장치를 제공하는 것이다.

여기서 말하는 대기시(standby time)란, 예를 들어 기억장치에 전원이 입력되고, 그 초기화 작업이 끝나고, 언제라도 액세스 가능하면서, 외부로부터 액세스 요구가 없는 상태이다. 실제로 하드 디스크와 같은 파일 기억 시스템은, 상시 외부로부터 액세스 되는 것이 아니고, 오히려 대기 상태가 길다. 또한 예를 들어 영상의 녹화와 같이, 기억장치에 외관상 계속적으로 액세스가 되고 있는 경우에도, 호스트로부터 기억장치에 입력되는 평균적인 데이터 전송량이 기억장치 자체가 가지는 데이터 전송 능력보다 충분히 작으면, 실제로는 호스트측 버퍼가 일정량의 영상 데이터를 축적한 다음에, 모든 장치에 데이터를 입력한다. 따라서 버퍼에 영상 데이터를 축적하는 동안, 장치는 대기 상태가 된다.

도 8은, 이러한 장치가 회복 처리를 실행하는 제어 순서의 예를 나타내는 플로차트(flow chart)이다. 기억장치는 전원이 입력되고, 액세스에 필요한 초기화가 이루어진 후(ST1), 무효 페이지 영역의 회복 처리를 실행한다. 그것은 이하의 스텝 (ST2~ST5)의 처리의 반복으로부터 완성된다.

<스텝 ST2>

무효화된 페이지 영역을 검색하고, 그 회복 처리를 실시해야 할 것인가 아닌가를, 적합한 알고리즘에 의해 판정한다. 예를 들어 이하와 같은 경우의 회복이 유효하다.

동일 블록내에 일정 수 이상의 무효 페이지 영역이 존재한다. (이러한 블록은 실질적으로 회복되는 페이지 수가 많은 한편, 백업의 필요한 페이지수가 적다. 따라서 효율적인 회복이 가능하다.). 빈 영역에 대해서, 일정 비율 이상의 무효 페이지 영역이 존재한다.

빈 영역이 전체의 용량의 일정 비율 이하이다.

회복 처리가 불필요한 경우는, 처리를 종료하고, 단지 액세스를 기다리고 아무것도 실시하지 않는 통상의 대기 상태에 들어간다.

<스텝 ST3>

회복 처리가 필요하다고 판정되면, 회복 대상으로 하는 소거 블록과 다른 소거 블록내의 백업용 빈 영역을 선정하고, 회복 처리를 개시한다.

<스텝 ST4>

회복 대상이 된 소거 블록중에, 백업해야 할 유효 페이지 영역이 있는지 아닌지를 판정한다. 만약 있으면, 상위로부터 1페이지 영역씩 빈 영역에 한 페이지씩 복사해 나간다. 각 페이지의 복사가 완료하는 때마다, 어드레스 변환 테이블을 갱신한다.

<스텝 ST5>

모든 유효 페이지 영역의 백업이 완료하면, 원래 블록을 소거한다.

이러한 기억장치에는, 외부로부터 액세스 요구가 이루어졌을 때에, 상기 회복 처리를 중단하는 기구가 필요하다는 것을 주목하자. 또한 하나의 소거 블록에 대해서, 그 중의 무효화 페이지 영역을 회복 처리하려면 소거만으로 수 밀리초를 필요로 하며, 스텝(ST2~ST5)의 처리를 모두 완료하려면 수십 밀리초를 필요로 하기도 한다. 따라서 그것들을 분할해 처리시키는 장치가 필요하다.

이러한 요구에 대응하는 유효한 시책의 하나는, 스텝(ST2~ST5)을 기억장치내의 CPU(예를 들어 도 5에 있어서의 내장 CPU(85))의 제어로 처리하고, 외부로부터 액세스 명령이 입력되면 CPU를 인터럽트하고, 액세스에 대한 응답 처리 프로그램에 제어를 전달하는 것이다.

즉, 기억장치 내부를 제어하는 CPU에 인터럽트에 의해 액세스에 대한 응답 처리를 실시하면, 회복 처리가 상기 스텝(ST2~ST5)의 어느 공정에서고 회복 처리가 중단된다. 이 경우 플래쉬 메모리에 페이지 데이터의 기입이 개시되고 있으면, 그 1페이지의 기입을 완료시키고, 주소 테이블을 갱신시키고 나서 처리를 중단한다. 이에 의해 수백 마이크로(μ) 초 정도의 지연으로 안전하게 액세스 처리를 개시할 수 있다. 또한, 플래시에 메모리의 소거가 개시되고 있으면, 소거를 완료할 때에 처리를 중단한다. 이에 의해 수밀리 초의 지연으로 안전하게 액세스 처리를 개시할 수 있다.

게다가, 플래쉬 메모리 칩 자체에 중단 처리에 대응할 수 있는 사양이 탑재되고 있으면, 기입이나 소거 동작의 도중에도 즉시 플래쉬 메모리를 재설정하여, 수십 마이크로 초의 수준에서 요구된 데이터 액세스를 개시하는 것이 가능하다. 이 경우, 기입이나 소거는 어중간한 상태로 중단되지만, 기입된 페이지 영역이나 소거중에 있는 소거 블록의 소재가 파악되면, 액세스 응답 처리가 완료된 후, 재차 상기 기입 또는 소거 처리를 재개하여, 문제 없이 처리를 완료할 수 있다.

어느 경우에 대해서도, 회복 처리를 중단했을 경우에는, 그 진행 상황을 기억장치내의 메모리나 레지스터에 기록해 두어, 액세스에 대한 응답 처리가 완료되면, 그 정보에 근거해 회복 처리를 재개하는 것이 바람직하다. 예를 들어, 스텝(ST2~ST5)의 처리 중도에 중단이 이루어졌을 때는, 적어도 회복 대상의 소거 블록의 소재를 진행 정보로서 보존해 둔다. 또한 바람직한 것은, 회복 처리를 중단한 것을 가리키는 플래그를 설정하는 것이다. 그 외 필요에 따라서 유효 페이지의 백업처(backup destination)의 빈 영역이나 백업중이던 페이지의 소재들을 기록해 두어도 좋다. 이러한 정보는, 예를 들어 도 5에 있어서의 RAM(77)내의 적당한 영역에 보존한다. 예를 들어 CPU를 제어하는 펌웨어(firmware)내에, 변수로서 보존해 두면 좋다.

액세스에 대한 응답 처리가 완료되면, 재차 대기 상태로 돌아와, 예를 들어 스텝(ST2)으로부터 회복 처리가 재개된다. 그때, 스텝(ST2)에서는 상기 플래그에 근거해 중단된 처리가 있다는 것을 판정하고, 회복 처리로 진행된다. 스텝(ST3)의 처리에서 전회의 처리를 계속하도록, 회복 대상의 소거 블록과 백업처의 빈 영역을 선택한다. 게다가, 스텝(ST4)에서는, 중단시에 유효 페이지 데이터는 어디까지 백업이 완료했는지, 혹은 이미 백업은 완료하여, 원래 블록의 소거중에 중단되었는지를 검사, 판정하고, 그 정보에 따라서 미완료된 공정으로부터 회복 처리를 실시한다.

이러한 구성을 이용하여, 호스트 컴퓨터에서 완전히 알지 못하는 상태에서, 기억장치 내부에서 자동적으로 회복 처리가 실행된다. 액세스가 수신되면, 자동적으로 회복 처리를 중단하고, 고속으로 액세스에 대한 응답 처리를 실행한다.

호스트 컴퓨터로부터 통상의 하드 디스크와 같은 인터페이스 사양이나 명령을 사용하는 것이 가능하다. 예를 들어 ATA, PCI-익스프레스, USB등의 범용 인터페이스를 이용하고, 각 규격마다 준비된 표준적인 명령만으로 동작시킬 수 있으므로, 전용의 드라이버를 준비할 필요도 없다. 이와 같이 일반적인 플래쉬 메모리 카드의 대체가 용이하다.

그런데 전술의 예에서는, 주기형 재기입 동작에서 발생한 무효 페이지 영역의 회복 처리(ST2~ST5)를, 호스트로부터의 액세스 요구에 수반하는 CPU 인터럽트에 의해 분할하면서 실행했다. 즉 본래 수십 밀리 초가 걸리는 회복 처리를, 분할하면서 액세스의 사이에 실행한 것이다. 그러나, 회복 처리의 분할 방법에는 각종 바리에이션이 고려된다. 예를 들어, 상기한 8의 스텝(ST2~ST5)의 회복 처리를, 미리 보다 작은 작업으로 분할해 두어, 그것을 단위로서 실행해 가면, 인터럽트 없이도 신속히 액세스에 대응할 수 있다. 그 예를 도 9에 도시하고 있다.

도 9는, 제 1의 실시 형태의 변형예이며, 무효 페이지 영역 회복의 분할 단위 처리를 설명하기 위한 플로차트이다. 기억 장치는 전원이 입력되면, 액세스에 필요한 초기화가 이루어진 후(ST11), 무효 페이지 영역의 회복 처리를 분할 단위마다 실행한다. 분할 단위 처리는 이하의 스텝 (ST12~ST15)의 처리로 구성된다.

<스텝 ST12>

무효화된 페이지 영역을 검색하고, 그 회복 처리를 실시해야 할 것인가 아닌가를, 적합한 알고리즘으로 판정한다. 여기서, 소거 블록 레벨에서 완료되지 않는 미완료의 회복 처리가 잔존하고 있는 경우, 그 다음의 처리로 진행된다. 회복 처리가 불필요한 경우는, 처리를 종료하고, 단지 액세스를 기다리면서 아무것도 실시하지 않는 통상의 대기 상태에 들어간다.

<스텝 ST13>

회복 처리가 필요하다고 판정되면, 회복 대상으로 하는 소거 블록과 백업처의 빈 영역을 선정하고, 회복 처리를 개시한다. 미완료된 회복 처리가 잔존하고 있는 경우, 거기서 작업 대상이 되는 소거 블록이 선택된다.

<스텝 ST14>

회복 대상이 된 소거 블록중에, 백업해야 할 유효 페이지 영역이 있는지 아닌지를 판정한다. 만약 있으면, 상부로부터 가장 높은 페이지를 빈 영역으로 복사한다(ST16). 게다가, 어드레스 변환 테이블(78)을 갱신하여 분할 단위 처리를 완료한다(ST17).

<스텝 ST15>

모든 유효 페이지 영역의 백업이 완료되면, 원래 블록을 소거하고, 분할 단위 처리를 완료한다. 또한 플래쉬 메모리 칩에 소거 동작을 중단하고, 재차 같은 블록을 소거함으로써 분할에 의한 소거를 할 수 있는 기능이 있으면, 소정의 시간 소거를 실시해서, 소거 완료전이라도 그것을 중단해, 분할 단위 처리를 완료해도 좋다. 즉 분할 소거를 실시해도 좋다.

상기 분할 단위 처리가 종료하고, 또한 소거 블록 레벨에서 무효 페이지 영역의 회복 작업이 미완료된 경우는, 회복 작업중의 소거 블록, 혹은 소거의 누계 시간등, 회복 처리의 진행 정보를, 기억장치내의 메모리나 레지스터에 보존해 두는 것이 바람직하다.

이러한 분할 단위 처리는, 처리마다 일단 작업을 종료해도 좋지만, 외부(호스트)로부터 액세스 요구가 없는 경우에는, 도면과 같이 계속해 다음의 분할 단위 처리를 실행해도 좋다.

다음에, 본 발명의 제 2의 실시 형태에 대해 설명한다. 본 발명의 제 2의 실시 형태는, 외부로부터의 전용(dedicated) 회복 명령에 따라서 상기 회복 처리를 실시하는 형태이다.

도 10은, 제 2의 실시 형태와 관련되는 회복 처리를 설명하기 위한 플로차트이다. 상술한 제 1의 실시 형태에 대해서는, 장치가 대기 상태가 되면 자동적으로 회복 처리가 실시되는데 반해, 제 2의 실시 형태에서는 장치 외부로부터 입력된 회복 명령에 따라서 회복 처리를 실시한다. 도면에서, 파선으로 가리킨 무효 페이지 영역의 회복 처리(ST21~ST26)는, 제 1의 실시 형태에 있어서의 도 8의 처리(작업)(ST2~ST7)와 같다.

게다가 제 6의 실시 형태에서는, 이용자로 부터의 소정의 명령에 따라 회복 처리를 중단하는 기능을 가진다. 예를 들어 그것이 데이터에 대한 액세스 명령이면, 액세스의 응답 처리를 실시하고, 회복 처리의 중단만을 실시하는 명령이면, 장치를 액세스 대기 상태로 이동시킨다. 액세스 명령, 회복 처리의 개시 명령, 회복 처리의 중단 명령은, CPU를 인터럽트하기 위한 처리를 이벤트로서 이용하여 각각의 처리 루틴이 실행되도록 한다.

또한, 회복 처리를 중단했을 경우, 제 1의 실시 형태와 같이, 그 진행 상황을 기억장치내의 메모리나 레지스터에 기록해 두어, 재차 회복 명령이 입력되면, 그 정보에 근거해 회복 처리를 재개하는 것이 바람직하다.

예를 들어 스텝(ST21~ST24)의 처리 중도에 중단이 이루어졌을 때는, 적어도 회복 대상의 소거 블록의 소재를 진행 정보로서 보존해 둔다. 또한 바람직한 것은, 회복 처리를 중단한 것을 가리키는 플래그를 설정해 둔다. 그 외 필요에 따라서 유효 페이지의 백업처의 빈 영역이나 백업중이었던 페이지의 소재등을 기록해 두어도 좋다. 이러한 정보는, 예를 들어 도 5에 있어서의 RAM(77)내의 적당한 빈 영역에 보존한다.

재차 회복 명령이 입력되면, 다시, 스텝(ST21)으로부터 회복 처리가 재개된다. 그 때, 스텝(ST21)에서는 상기 플래그에 근거해 중단된 처리가 있는 것을 판정하고, 회복 처리로 진행된다. 스텝(ST22)에서 전회의 처리를 계속하도록, 회복 대상의 소거 블록과 백업처의 빈 영역을 선택한다. 게다가 스텝(ST23)에서는, 중단시에 유효 페이지 데이터는 어디까지 백업이 완료했는지, 혹은 이미 백업은 완료하여, 원래 블록의 소거중에 중단되었는지를 검사, 판정하고, 그 정보에 따라서 미완료된 공정으로부터 회복 처리를 실시한다.

제 2의 실시 형태에 대해서는, 호스트측이 회복 처리의 개시시기를 판단할 필요가 있으며, 기억장치의 관리에 독자적인 명령이나 작업을 필요로 하고, 이용자 입력이나 애플리케이션 프로그램의 상황에 따라, 가장 적절한 시기에 회복 처리를 실행할 수 있는 이점이 있다. 또한 장치에 대한 액세스를 사전에 예측하고, 회복 처리를 중단해, 대기 상태로 이동시키면, 중단 처리에 필요한 시간이 불필요해져, 보다 신속한 액세스에 대한 응답 처리가 가능하게 된다. 예를 들어 이용자로 부터의 키 입력이나 마우스 입력이 일정시간 없으면 회복 처리를 개시하여, 입력이 이루어진 경우에 회복 처리를 중단한다.

제 2의 실시 형태에 대해서는, 주기형 재기입시에 발생한 무효 페이지 영역을 위한 회복 처리 스텝(ST21~ST24)을, 중단 명령등에 수반하는 인터럽트로 분할했다. 그러나, 제 1의 실시 형태의 변형예와 같이, 미리 회복 처리를 보다 작은 작업으로 분할하고, 그것을 단위로서 실행해 가면, 인터럽트 없이도 신속히 액세스에 대응할 수 있다.

도 11은, 제 2의 실시 형태의 변형예이며, 무효 페이지 영역 회복의 분할 단위 처리를 설명하기 위한 플로차트이다. 도중, 파선으로 가리킨 무효 페이지 영역 회복의 분할 단위 처리(ST31~ST36)는, 제 1의 실시 형태의 변형예에 있어서의 도 9의 처리(작업)(ST12~ST17)과 같다.

상기 분할 단위 처리가 종료하거나, 또는 소거 블록 레벨에서 무효 페이지 영역의 회복 작업이 미완료인 경우는, 작업중의 소거 블록, 혹은 필요에 따라서 소거의 누계 시간등, 회복 처리의 진행 정보를, 기억장치내의 메모리나 레지스터에 보존해 두는 것이 바람직하다.

또한 이러한 분할 단위 처리는, 호스트로부터의 명령에 따라 여러 차례 그것을 반복하거나 분할 작업의 내용을 바꾸는 등, 다양한 바리에이션이 존재할 수 있다. 게다가 1회의 명령으로 한 개의 소거 블록 분의 회복 처리를 정리해 실시하는 「비분할 모드」를 추가해도 좋다.

다음에, 본 발명의 제 3의 실시 형태에 대해 설명한다. 본 발명의 제 3의 실시 형태는, 기억장치 내부에 캐시 메모리를 마련하고 거기에 도달하는 동안에 회복 처리를 실시하는 것이다.

도 12는, 본 발명의 제 3의 실시 형태와 관련되는 기억장치를 나타내는 구성도이다. 또한, 도 13은, 제 3의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블의 구성예를 나타내는 도면이다.

도 12의 기억장치(110)는, 인터페이스 회로(I/F)(111), 페이지 버퍼(112), 내부 데이터 버스(113), 제어 회로(114), NAND형 플래쉬 메모리(115, 116), 캐시 기억 장치(117), 및 제 2의 내부 데이터 버스(118)를, 주구성 요소로서 가지고 있다.

기억장치(110)에서, 32비트의 내부 데이터 버스(113)에는, 16비트의 입출력을 가지는 2칩의 NAND형 플래쉬 메모리(115, 116)가 병렬 접속되어 있다.

2개의 칩의 플래쉬 메모리(115, 116)는 판독 또는 기입 동작시에, 동시 병렬적으로 액세스 된다. 페이지 버퍼(112)는 액세스된 페이지 영역을 일시 기억하는 버퍼이며, 외부와 인터페이스 회로(111)를 통해 연결되며, 캐시 메모리(117)는 제 2의 내부 데이터 버스(118)에 접속되어 있다.

제어 회로(114)는, 메모리 칩인 플래쉬 메모리(115, 116)와 페이지 버퍼(112), 캐시 메모리(117), 및 인터페이스 회로(111)의 사이에서 데이터의 교환을 관리하는 제어기이며, 내장 CPU(119)에 의해서 제어된다. 이와 같이, 내장된 RAM(120)에는, CPU(119)를 제어하기 위한 프로그램 영역이나 작업 영역이 확보되는 동시에, 어드레스 변환 테이블(121)이 구축되어 있다. 동시에, 제어 회로(114)에는, CPU의 제어 없이 캐시 메모리(117)와 인터페이스 회로(111)간의 데이터 전송을 제어하는 제어 회로(122)가 탑재되어 있다.

기억장치(110)의 액세스는, 기본적으로는 도 5에 도시한 제 1의 실시 형태와 같다. 다만 외부 호스트와 기억장치(110)의 주기적인 플래쉬 메모리(115, 116)와의 사이에 교환되는 데이터의 일부는 캐시 메모리(117)안에 보존된다.

호스트가 기억장치(110)를 재차 액세스 하고, 캐시 메모리(117)내에 보존된 것과 같은 논리 주소의 데이터를 액세스 하려고 하면, 캐시 메모리(117)측의 데이터가 액세스 된다. 따라서 이 기간 동안에, 페이지 버퍼(112), 내부 데이터 버스(113), NAND형 플래쉬 메모리(115, 116)는 모두 액세스에 사용되지 않는다. 제 3의 실시 형태에서는 이 기간이 무효 페이지 영역의 회복 처리에 이용된다.

도 14는, 제 3의 실시 형태와 관련되는 회복 처리를 설명하기 위한 플로차트이다. 도중, 파선으로 가리킨 무효 페이지 영역의 회복 처리는, 제 1의 실시 형태에 있어서의 도 8의 처리(작업)(ST2~ST7)과 같다.

이 예에서는 제 1의 실시 형태 같이, 장치가 대기 상태가 되면 자동적으로 회복 처리가 실시되지만, 인터럽트에 의한 회복 처리의 중단되는 시간이 캐시 메모리(117)의 미스 히트(time of miss)시간이다. 장치의 개시후, 호스트로부터의 액세스가 도달하는 동안, 액세스 처리는 제어 회로(122)에 의해 실행되며, CPU(119)는 무효 페이지 영역의 회복 처리를 실시한다. 캐시가 미스하면(miss), 제어 회로(122)는 CPU(119)를 인터럽트하고, CPU(119)는 거기에 대응하는 미스 응답 처리를 실시한다. 구체적으로는 호스트의 액세스 요구에 따라 어드레스 변환 테이블(121)을 참조하여, 거기서 취득된 물리 주소로부터 플래쉬 메모리(115, 116)를 액세스하고, 페이지 버퍼(112)를 통해 페이지 데이터의 판독 또는 기입을 실시한다. 또한 필요에 따라서 호스트로부터의 입력 데이터 또는 호스트에 대한 출력 데이터를 캐시 메모리(117)에 캐시한다.

회복 처리를 중단했을 경우, 제 1의 실시 형태와 같이, 그 진행 상황을 기억장치내의 메모리나 레지스터에 기록해 두어, 재차 회복 명령이 입력되면, 그 정보에 근거해 회복 처리를 재개하는 것이 바람직하다.

예를 들어 스텝(ST42~ST45)의 처리 중도에 중단이 이루어졌을 때는, 적어도 회복 대상의 소거 블록의 소재를 진행 정보로서 보존해 둔다. 또한 바람직한 것은, 회복 처리를 중단한 것을 가리키는 플래그를 설정해 둔다. 그 외 필요에 따라서 유효 페이지의 백업처의 빈 영역이나 백업중이었던 페이지의 소재등을 기록해 두어도 좋다. 이러한 정보는, 예를 들어 도 12에 있어서의 RAM(120)내의 적당한 빈 영역에 보존한다.

미스 히트에 대한 응답 처리가 완료하면, 액세스 제어는 다시 제어 회로(122)로 전달되며 CPU(119)는 회복 처리를 스텝(ST42)으로부터 재개한다. 그 때, 스텝(ST42)에서는, 상기 플래그에 근거해 중단된 처리가 있는 것을 판정하고, 회복 처리로 진행된다. 스텝(ST43)에서 전회의 처리를 계속하도록, 회복 대상의 소거 블록과 백업처의 빈 영역을 선택한다. 게다가, 스텝(ST44)에서는, 중단시에 유효 페이지 데이터는 어디까지 백업이 완료했는지, 혹은 이미 백업은 완료하고, 원래 블록의 소거중에 중단되었는지를 검사, 판정하여, 그 정보에 따라서 미완료된 공정으로부터 회복 처리를 실시한다.

제 3의 실시 형태에서는, 캐시가 히트(hit)하고 있는 동안, 혹은 대기시에 실시되고 있던 무효 페이지 영역의 회복 처리(ST42~ST45)를, 캐시의 미스 히트에 수반하는 인터럽트에 의해 분할했다. 그러나 제 1의 실시 형태의 변형예와 같이, 미리 회복 처리를 보다 작은 작업으로 분할해 두어, 그것을 단위로서 실행해 가면, 인터럽트 없이도 신속히 액세스에 대응할 수 있다.

도 15는, 제 3의 실시 형태의 변형예이며, 무효 페이지 영역 회복의 분할 단위 처리를 설명하기 위한 플로차트이다. 도중, 파선으로 가리킨 무효 페이지 영역 회복의 분할 단위 처리는, 제 1의 실시 형태의 변형예에 있어서의 도 9의 처리(작업)(ST12~ST15)과 같다.

상기 분할 단위 처리가 종료하거나, 또는 소거 블록 레벨에서 무효 페이지 영역의 회복 작업이 미완료된 경우는, 작업중의 소거 블록의 소재, 혹은 필요에 따라서 소거의 누계 시간등, 회복 처리의 진행 정보를, 기억장치내의 메모리나 레지스터에 보존해 두는 것이 바람직하다.

이러한 분할 단위 처리는, 처리 마다 일단 작업을 종료해도 좋지만, 캐시의 미스 히트가 발생하고 있지 않는 경우에는, 도면과 같이 다음의 분할 단위 처리를 실행해도 좋다.

또한 제 3의 실시 형태에서는, 무효 페이지 영역 회복의 분할 단위 처리를 이용한 다른 변형예에 대해 설명한다.

도 16은, 제 3의 실시 형태의 제 2의 변형예이며, 무효 페이지 영역 회복의 분할 단위 처리를 설명하기 위한 플로차트이다.

여기에서는 캐시 히트를 기점으로 회복 처리를 실시하고 있다. 즉 호스트로부터의 액세스가 캐시 메모리(117)에 히트하면, CPU(119)는 액세스에 대한 응답을 제어 회로(122)에 건네주며(ST61), 무효 페이지 영역 회복의 분할 단위 처리를 실시한다. 파선으로 가리킨 무효 페이지 영역 회복의 분할 단위 처리는, 제 1의 실시 형태의 변형예에 있어서의 도 9의 처리(작업)(ST12~ST17)과 같다.

이상 3개의 제 1~제 3의 실시 형태에 대해 상세하게 말했지만, 이와 같이 무효 페이지 영역의 회복 처리를 분할해 실시할 때, 가장 분할 단위가 길어지는 것은 블록의 소거이다. 예를 들어 NAND형 플래시 메모리에서는 1페이지의 복사가 150~200 마이크로(μ) 초정도로 종료하는데 반해, 소거는 1~2밀리(m) 초 걸린다. 따라서 호스트가 액세스를 실시할 때의 지연 시간은, 소거시가 워스트(worst) 시간이 된다.

예를 들어, 호스트상의 프로그램이 기억장치를 이용하고 작업을 실시하는 경우, 그 워스트의 지연(worst delay)을 기준으로 하여 모든 작업을 처리하기 위한 시간 마진(margin)을 전망하는 경우가 자주 있다. 그 경우 프로그램의 처리 속도는 큰 영향을 받게 된다.

이러한 과제에 대처하려면, 액세스의 사이에 실시하는 회복 처리는 한 페이지 복사에만 제한을 둘 수 있으며, 블록 소거는 전원 온 또는 오프시에 모든 블록을 실행하는 것이 가능하다. 전원 온/오프시에, 이용자는 체감 속도로서 200밀리(m) 초이하의 지연은 대부분 신경쓰지 않는다. 따라서 전원 온/오프시에 100 블록 이상을 소거하는 것이 가능해진다.

또한, 미소거 블록의 소거만을 실시하는 명령이 제공될 수 있다. 예를 들어 이용자로부터의 입력이 잠시 없는 경우, 호스트가 그러한 명령을 실시하면, 이용자가 체감적인 지연을 느끼는 일없이 블록 소거를 실시할 수 있다.

제 1의 실시 형태에서, 전원 온시에 모든 소거를 실행한 예를, 제 4의 실시 형태로서 설명한다. 제 4의 실시 형태의 장치 구성은 제 1의 실시 형태의 도 5와 같다.

도 17은, 제 4의 실시 형태에 있어서의 무효 영역의 회복 처리를 설명하기 위한 플로차트이다.

기억장치(70)에 전원이 입력되어, 액세스에 필요한 초기화가 이루어지면(ST71), 우선 장치 내부에서는 미리 소거 가능하다고 표시된 백업이 끝난 블록의 소거가 개시된다(ST72). 그 완료 뒤에, 무효 페이지 영역의 회복 처리가 실행된다. 그것은 이하의 스텝(ST73~ST76)의 처리 플로우의 반복으로 구성된다.

<스텝 ST73>

무효화된 페이지 영역을 검색하고, 그 회복 처리를 실시해야 할 것인가 아닌가를, 적합한 알고리즘으로 판정한다. 회복 처리가 불필요한 경우는, 처리를 종료하고, 단지 액세스를 기다리면서 아무것도 실시하지 않는 통상의 대기 상태에 들어간다.

<스텝 ST74>

회복 처리가 필요하다고 판정되면, 회복 대상으로 하는 소거 블록과 백업처의 빈 영역을 선택해, 회복 처리를 개시한다.

<스텝 ST75>

회복 대상이 된 소거 블록중에, 백업해야 할 유효 페이지 영역이 있는지 아닌지를 판정한다. 만약 있으면, 상위로부터 1페이지 영역씩 빈 영역에 복사해 나간다(ST77). 각 페이지의 복사가 완료하는 때 마다, 어드레스 변환 테이블을 갱신한다(ST(78)).

<스텝 ST76>

모든 유효 페이지 영역의 백업이 완료하면, 원래 블록을 「백업이 끝난 블록」으로서 마크한다. 마크의 수법으로서는 예를 들어 블록의 말미 페이지에 플래그를 설정하거나, 혹은 백업이 끝난 블록의 물리 주소를 장치내의 어느 장소에 보존한다.

외부로부터 액세스 명령이 입력되면 CPU가 인터럽트된다. 그리고, 액세스에 대한 응답 처리 프로그램으로 제어가 전달된다. 이 경우, 플래쉬 메모리내에 페이지 데이터의 기입이 개시되고 있으면, 그 1페이지의 기입을 완료시키고, 주소 테이블을 갱신시키고 나서 처리를 중단한다. 이에 의해 수백 마이크로(μ) 초정도의 지연으로 안전하게 액세스 처리를 개시할 수 있다.

이상 일례를 나타냈지만, 제 1~제 3의 실시 형태 및 각각의 변형 예의 모두에 대해서 같은 방법이 가능하다는 것은 말할 것도 없다.

상술한 경우에서도, 호스트로부터의 재기입시에 소거가 끝난 블록수가 부족했을 경우는, 그 시점에서 소거를 실시하는 것이 바람직하다.

이상, 본 발명을 이용한 기억장치에 대해 구체적인 실시 형태를 말했지만, 장치 내부의 구성은 여러 가지로 수정될 있다. 예를 들어 도 5에 있어서의 페이지 버퍼(72)는, 제어 회로(74) 내부의 RAM(77)상에 구축해도 좋다. 그러한 경우에는 인터페이스 회로(71), 플래쉬 메모리(75, 76) 및 제어 회로(74)는 공통의 내부 데이터 버스(73)에 의해 접속된 구성이 된다. 또한, 페이지 버퍼(72)와 플래쉬 메모리(75, 76)와의 사이에 ECC에 의한 에러 수정 회로를 마련하고, 페이지 버퍼로부터 플래쉬 메모리에 페이지 데이터를 기입시에는 부호화를 실시해 패리티 비트를 추가하고, 플래쉬 메모리로부터 페이지 버퍼에 페이지 데이터를 판독할 때에는 복호화를 행해도 좋다. 그러한 장치 구성의 차이에 관련되지 않고, 본 발명을 적용하는 것은 가능하다. 또한 먼저 설명한 제 1~제 4의 실시 형태로서 말한 것처럼, 어드레스 변환 테이블(78, 121)에 따라서 페이지 단위의 어드레스 변환을 실시한 후, 플래쉬 메모리내의 결함 블록을 스킵하기 위해서, 블록 레벨에서 어드레스 변환을 삽입해도 좋다. 이러한 경우에도, 전술한 예와 같은 제어로, 본 발명을 문제 없이 적용하는 것이 가능하다.

그런데 여기까지는, 독립한 기억장치의 내부에 회복 처리 장치를 설치하는 경우에 대해 설명해 왔다. 그러나, 이러한 어드레스 변환 테이블의 관리와 회복 처리를, 호스트측의 제어로 실시하는 것도 가능하다. 그러한 컴퓨터 시스템을 제 5의 실시 형태에서 도 18에 도시하고 있다.

도 18은, 제 5의 실시 형태와 관련되는 컴퓨터 시스템의 구성예를 나타내는 도면이다. 또한, 도 19는, 제 5의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블의 구성예를 나타내는 도면이다.

도 18의 컴퓨터 시스템(130)은, 처리 유닛로서의 CPU(131), 시스템메모리인 RAM(132), 시스템 버스(133), 브릿지 회로(134), NAND형 플래쉬 메모리(135, 136), 데이터 버스(137), 및 페이지 버퍼(138)를, 주구성 요소로서 가지고 있다.

CPU(131)는 32비트의 시스템 버스(133)를 통해 시스템 메모리인 RAM(132)에 접속되어 있다. 게다가, 시스템 버스(133)에는 브릿지 회로(134)가 접속되어 있다. 브릿지 회로(134)에 연결되는 32비트의 데이터 버스(137)에는, 16비트의 입출력을 가지는 2칩의 NAND형 플래쉬 메모리(135, 136)가 병렬 접속되어 있다. 2칩의 플래쉬 메모리(135, 136)는 판

독 또는 기입시에, 동시 병렬적으로 액세스 된다. 페이지 버퍼(138)는 액세스된 페이지 영역을 일시 기억하는 버퍼이며, 브릿지 회로(134)에 내장되어 있다. 브릿지 회로(134)는 CPU(131)로부터 각종 명령을 받아, 플래쉬 메모리(135, 136)와 CPU(131) 또는 시스템 메모리(132)와의 사이에서의 데이터의 교환을, 페이지 버퍼(138)를 이용해 중개한다. 브릿지 회로(134)가 수신한 명령은, 예를 들어 플래쉬 메모리(135, 136)의 소정의 페이지에 대한 액세스의 이외에, 동일한 플래쉬 메모리의 소정 블록의 소거, 및 소정 페이지의 지정 주소에 대한 복사, 플래쉬 메모리의 재설정등이다.

한편, 시스템 메모리(132)내에는 상기 플래시 기억 시스템을 제어하기 위한 드라이버(139)가 있다. 이 드라이버(139)는, 오퍼레이팅 시스템(OS)이나 애플리케이션으로부터 기억장치에 대한 액세스를 수신하여 같은 메모리중에 구성된 어드레스 변환 테이블(140)을 참조하여 액세스시의 페이지 주소를 변환한다.

기억 시스템의 액세스는 이하와 같이 된다. 여기에서는 간단히, 16 진수의 주소는 다음과 같이 할당해진다고 가정한다. 예를 들어 외부 입력 주소가 "0x5500C"인 경우, 상위의 "0x5500"은 페이지 주소, 하위 "0xC"은 페이지 영역내의 섹터 드레스이며, 기억장치는 1섹터 단위로 랜덤 액세스가 가능하다.

판독시에, 예를 들어 애플리케이션으로부터 주소 "0x5500C"의 섹터의 판독이 요구되면, 드라이버(139)는 상위의 페이지 주소를 받아 어드레스 변환 테이블(140)로부터 논리 페이지 주소(LA) "0x5500"에 대응하는 물리 페이지 주소(PA) "0x00B0"을 취득한다. 이 물리 페이지 주소(PA)중, 상위의 "0x00"은 메모리 칩의 플래시 메모리(135, 136)내의 소거 블록 (141, 142)의 주소이다. 하위 "0xB0"은 각 소거 블록내의 페이지 영역 (143, 144)의 주소이다. 이 물리 주소를 기초로, 브릿지 회로(134)를 통해 메모리 칩의 플래시 메모리(135, 136)가 모두 액세스 되며, 각각으로부터 페이지 주소 "0x00B0"에 저장된 페이지 영역(143, 144)의 데이터가 판독되어 페이지 버퍼(138)에 저장된다. 브릿지 회로(134)는 그 중에서, 하위 주소 ?0xC?에 해당하는 섹터를 선택하고, CPU(131) 또는 시스템 메모리(132)에 출력한다.

한편, 애플리케이션으로부터 "0x5500C"에 기입 요구를 받았을 경우는, 우선 판독과 같은 순서로 드라이버(139)에 의해서 어드레스 변환이 실시되며, 페이지 영역(143, 144)의 데이터가 메모리 칩의 플래시 메모리 (135, 136)로부터 판독되어 페이지 버퍼(138)에 저장된다. 그 후 하위 주소 "0xC"에 해당하는 섹터가 선택되어 신규 데이터로 갱신된다. 갱신된 페이지 데이터는, 플래시 메모리(135, 136)에 재기입되며, 그 때 각각의 빈영역(145, 146)에 기입된다. 이 페이지 영역의 물리 페이지 주소는 "0x0280"이며, 즉 소거 블록 "0x02"안의 "0x80"의 페이지에 상당한다. 또한, 이 처리 전까지, 그 영역은 어드레스 변환 테이블(140)의 어느 물리 주소 필드에도 등록되지 않은, 미사용의 빈 페이지였다. 갱신된 데이터가 이 영역에 추기형 동작의 형태로 기입된다. 동시에, 어드레스 변환 테이블(140)의 논리 주소 "0x5500"에 대응하는 물리 주소 필드에는, 빈 영역 (145, 146)의 물리 페이지 주소 "0x0280"가 등록된다.

상술한 추기형 기입을 실시했을 경우, 갱신전의 데이터가 저장되어 있는 물리 페이지 주소 ?0x00B0?에 해당하는 페이지 영역(143, 144)은 어드레스 변환 테이블(140)의 물리 주소 필드로부터 삭제되어 무효화된다. 그러나 그것들에는 데이터가 기입되어 있으므로, 그대로는 빈 영역으로서 사용할 수 없다. 상술한 재기입을 몇 번이나 반복하면, 많은 무효 페이지 영역이 발생한다. 그것들은 재차 빈 영역으로서 사용할 수 있도록, 소거해 회복시킬 필요가 있다. 또한 그 경우, 소거 블록 (141, 142)에 남겨진 유효 데이터는 백업시키지 않으면 안 된다.

이러한 회복 처리는, 예를 들어 도 7a~도 7c와 관련지어 기술한 것과 같은 알고리즘으로, 빈 영역을 가지는 블록(147, 148)을 사용해 행해진다. 즉 회복 처리의 대상 블록인 소거 블록(141, 142)상의 유효한 페이지 영역군의 데이터를 백업처 블록(147, 148)내의 빈 영역에 차례로 복사하고, 거기에 대응하여 어드레스 변환 테이블(140)의 물리 주소를 갱신한다. 모두 유효 페이지 데이터가 복사되어 끝나면, 원래 블록(141, 142)을 소거한다.

제 5의 실시 형태에서는, 드라이버(139)를 이용해 호스트 CPU(131)가 브릿지 회로(134)에 명령을 전송하여, 그러한 처리(작업)를 제어한다. 예를 들어 상기 회복 작업에 있어서의 유효 페이지 데이터의 복사는, 호스트 CPU(131)로부터의 복사 명령에 따라, 브릿지 회로(134)가 소정의 페이지를 복사원(copying source)으로부터 페이지 버퍼(138)에 판독하고 그 페이지를 복사원에 기입함으로써 이루어진다. 또한, 소정의 블록을 소거한다. 이 때 시스템 버스(133)는 사용되지 않기 때문에, 만약 필요가 있으면, CPU(131)는 상기 명령을 송신한 후에 다른 처리(작업)를 병행해 실시하는 것도 가능하다.

도 20은, 프리엠프티브(preemptive)한 멀티태스킹 시스템을 이용한 회복 처리의 일례를 나타내는 플로차트이다.

호스트 CPU(131)는 타이머나 우선 순위를 이용한 스케줄링에 따라 회복 작업을 적절한 시간에 실행한다. 각 실행에는 5 밀리 초의 타임 슬라이스를 할당할 수 있다고 가정한다. 호스트는 이 할당 시간을 이용하고, 회복 처리를 분할해 실행한다.

회복 작업이 개시되면, 호스트 CPU(131)는 우선 브릿지 회로(134)에 기억 시스템 상태를 문의하고 기억 시스템이 처리(작업) 중인가 아닌가를 체크한다(ST81). 처리(작업)중이면 작업은 종료하고, 기억 시스템이 대기중이면, 드라이버(139)의 프로그램에 따라, 이하와 같은 무효 영역의 분할 회복 처리를 실시한다.

<스텝 ST82>

무효화된 페이지 영역을 검색하고, 그 회복 처리를 실시해야 할 것인가 아닌가를, 적절한 알고리즘으로 판정한다. 여기서, 소거 블록 레벨에서 완료하지 않는 미완료의 회복 처리가 잔존하고 있는 경우, 그 계속 처리로 진행된다. 회복 처리가 불필요한 경우는, 작업을 종료한다.

<스텝 ST83>

회복 처리가 필요하다고 판정되면, 회복 대상으로 하는 소거 블록과 백업처의 빈 영역을 선정하여, 회복 처리를 개시한다. 미완료의 회복 처리가 잔존하고 있는 경우, 거기서 작업 대상이 되는 소거 블록과 백업처의 빈 영역이 선택된다.

<스텝 ST84>

회복 대상이 된 소거 블록중에, 백업해야 할 유효 페이지 영역이 있는지 아닌지를 판정한다. 만약 있으면, 최상위의 1페이지를 빈 영역에 상부로부터 복사한다(ST86). 게다가, 어드레스 변환 테이블(140)을 갱신해 분할 단위 처리를 완료한다(ST87). 본 예와 같이 1회의 작업 실행에 5 밀리 초의 할당이 있는 경우, 그 사이에 복수의 페이지 복사가 가능하다. 따라서 할당할 수 있는 타임 슬라이스에 따라 처리 페이지의 상한을 마련하고(ST88), 거기에 도달할 때까지 복사 처리를 반복해도 좋다.

<스텝 ST85>

모든 유효 페이지 영역의 백업이 완료하고 있으면, 원래 블록을 소거하고, 분할 단위 처리를 완료한다.

상기 분할 단위 처리가 종료하고 소거 블록 레벨에서 무효 페이지 영역의 회복 작업이 미완료인 경우는, 작업중의 소거 블록의 소재를 포함한 회복 처리의 진행 정보를, 예를 들어 시스템 메모리(132)의 작업 영역(141)내에 보존해 두는 것이 바람직하다.

이상, 호스트가 회복 처리를 제어하는 경우에 대해서 일례를 말했지만, 기억장치의 내부에 회복 처리 장치를 설치했을 경우와 같이, 각종 바리에이션이 존재 할 수 있다. 상술한 바와같이, 상수한 시분할로 다른작업을 실행하는 것 외에, 예를 들어 이용자로부터의 키 입력이나 마우스 입력이 일정시간 없으면 회복 처리를 개시하고, 입력이 이루어지면 회복 처리를 중단해도 좋다. 또한 시스템의 온/오프시에 블록의 소거 처리를 실행해도 좋다. 중도에 ECC에 의한 에러 수정이나, 결함 블록 회피를 위한, 블록 단위의 어드레스 변환을 삽입해도 좋으며, 그 제어는 호스트측 또는 기억장치측, 예를 들어 브릿지 회로(134)의 내부에서 독립적으로 실행해도 좋다.

지금까지 설명한 무효 영역의 회복 처리를, 이용자로부터의 플래쉬 메모리에 대한 액세스가 없는 기간을 추출해 실시하는 경우, 이하와 같은 과제가 있다.

예를 들어 메모리 카드와 같은 탈착 가능한 기억장치의 경우, 대기시에서 회복 처리를 실시하여도, 이용자 본인은 액세스를 명시적으로 지시한 것은 아니다. 따라서 카드를 호스트로부터 뽑아내 버릴 가능성이 있으므로, 그 때 기억장치에 전원의 순간적인 단절이 발생한다. 또한, 내장된 저장부에서도, 호스트 기기의 전원이, 콘센트의 제거등으로 갑자기 단절되는 경우가 있다.

이 때에 회복 처리의 진행 정보가 휘발성 메모리에 저장되어 있으면, 그것들이 소실되며, 결과적으로 회복중에 있는 소거 블록이나 백업처의 영역은, 매우 불안정한 상황에 빠져 버린다. 따라서 이러한 경우에는, 회복 처리의 진행 상황의 적어도 일부는, 이른바 불휘발성 RAM에 기록되어 있는 것이 바람직하다.

불휘발성 RAM은 불휘발성이면서 1마이크로(μ) 초이하로 랜덤 액세스가 가능한 RAM의 성질을 겸비하는 반도체 메모리이며, 강유전체막을 이용한 FeRAM(Ferroelectric RAM), 강자성체를 이용한 MRAM(Magnetic RAM)이나, 상변화 재료를 이용한 OUM(Oonic Unified Memory), RRAM등이 제안되고 있다.

FeRAM은 강유전체 캐패시터의 분극 방향의 차이로 데이터를 기억하는 반도체 메모리이다. 예를 들어 미국 특허 4873664의 S. Sheffield등이 그 한 형태를 제안하고 있다. MRAM는 강자성막의 스핀 방향의 차이로 데이터를 기억하는 반도체 메모리이다. 예를 들어 ISSCC2000의 논문 다이제스트의 128 페이지에, R.Scheuerle in등이 논문을 게재하고 있다. OUM는 예를 들어 카르코게나이드막(chalcogenide film)의 상전이(phase transition)로 데이터를 기억하는 반도체 메모리이다. 예를 들어 IEDM2001의 논문 다이제스트의 803 페이지에, S.Lai등이 논문을 게재하고 있다. RRAM는 자기 저항 효과 재료의 저항 히스테리시스로 데이터를 기억하는 반도체 메모리이다. 예를 들어 IEDM2002의 논문 다이제스트의 7.5에는 W.W.Zhuang등이 논문을 게재하고 있다. 그러한 불휘발성 메모리는, 모두 셀 레벨의 액세스 속도나 재기입 회수에서, 플래쉬 메모리보다 성능이 더욱 높다.

예를 들어 도 5와 관련된 제 5 및 제 2의 실시 형태의 경우, 적어도 회복 처리중의 블록 주소는 보존될 필요가 있으며, 불휘발성 RAM에 저장되어야 한다. 게다가, 어드레스 변환 테이블(78)과 페이지 관리 테이블(86)도 진행 정보의 일부를 포함하므로, 불휘발성 RAM에 보존되는 것이 바람직하다. 예를 들어 도 5의 RAM(77)을 SRAM과 불휘발성 RAM의 혼합으로 구축하고, 상기 회복 처리의 진행 상황이나 각종 관리 테이블을 불휘발성 RAM측에, 기억장치 제어용의 펌웨어 프로그램이나 작업 영역을 SRAM측에 구축해도 좋다.

이러한 사정은, 무효 영역의 회복 처리시에 진행 정보를 보존하고 있는 다른 실시 형태에 대해서도, 모두 동일하다.

도 21은, 제 6의 실시 형태를 나타내는 도면이며, 그러한 ECC 처리와 결합 블록 관리를 도입한, 제 3의 실시 형태(도 12)의 장치 구성상의 응용 변형예를 나타내는 도면이다. 도 22a와 도 22b는, 제 6의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블 및 페이지 상태의 관리 테이블의 구성예를 나타내는 도면이다.

기억장치(110A)에서는, 32비트의 내부 데이터 버스(113A)에, 16비트의 입출력을 가지는 2칩의 NAND형 플래쉬 메모리(115A, 116A)가 병렬 접속되어 있다. 2칩의 플래쉬 메모리(115A, 116A)는 판독 또는 기입시에, 동시 병렬적으로 액세스된다. 페이지 버퍼(112A)는 액세스된 페이지 영역을 일시 기억하는 버퍼이며, 외부와 인터페이스 회로(111A)에 제 2의 내부 데이터 버스(118A)를 통해 접속되어 있다. 게다가, 제 2의 내부 데이터 버스(118A)에는 캐시 메모리(117A)가 접속되어 있다. 제어 회로(123)는 플래쉬 메모리(115A, 116A)와 페이지 버퍼(112A) 사이에서의 전송을 제어한다.

제어 회로(123)에는 플래쉬 메모리내의 결합 블록이 액세스 되지 않도록, 그 내장 RAM(124)상에, 블록 단위의 어드레스 변환으로 그것들을 스킵시키기 위한 어드레스 변환 테이블(126)이 구축되어 있다. 게다가, 제어 회로(123)에는 ECC 회로(125)가 탑재되어 있으며, 페이지 버퍼(112A)로부터 플래쉬 메모리(115A, 116A)에 데이터가 기입될 때, 부호화에 의한 패리티 비트의 추가를 실시하고, 플래쉬 메모리(115A, 116A)로부터 페이지 버퍼(112A)에 데이터가 판독될 때, 복호화에 의한 에러 수정을 실시한다.

제어 회로(114A)는 페이지 버퍼(112A), 캐시 메모리(117A) 및 인터페이스 회로(111A) 사이에서의 데이터의 교환을 관리하는 제어기이며, 내장 CPU(119A)에 의해서 제어된다. 이와같이, RAM(120A)에는, CPU(119A)를 제어하기 위한 프로그램 영역이나 작업 영역이 확보되는 동시에, 어드레스 변환 테이블(121A)이 구축되어 있다. 게다가, 제어 회로(114A)에는, CPU(119A)의 제어 없이 캐시 메모리(117A)와 인터페이스 회로(111A)간의 데이터 전송을 제어하는 제어 회로(122A)가 탑재되어 있다. 또한, 본 도면에 있어서 제어선의 기재는 생략되어 있다.

제 6의 실시 형태에 있어서의 제 3의 실시 형태(도 12)와의 주된 차이는, 페이지 버퍼(112A)와 플래쉬 메모리(115A, 116A)와의 사이의 전송에 관해서, 어드레스 변환 테이블(126)과 ECC 회로(125)를 보유하는 전용의 제어 회로(123)가 설치되어 있다는 것이다. 이에 의해 외부로부터 입력된 논리 주소는, 그것을 바탕으로 플래쉬 메모리(115A, 116A)에 대한 액세스가 이루어지는 경우, 어드레스 변환 테이블(121A, 126)에 의해 이중의 변환을 실행한다. 게다가, 입출력 데이터에도 ECC에 의한 부호화, 복호화의 변환이 실행된다. 구체적인 동작을 이하에 말한다.

판독시에, 외부로부터 0x5500C?의 섹터 드레스가 입력되면, 제어 회로(114)는 상위의 페이지 주소를 받아 내장 RAM(120A)에 액세스하고, 어드레스 변환 테이블(121A)로부터 논리 페이지 주소(LPA) "0x5500"에 대응하는 물리 페이지 주소(PPA) "0x00B0"을 취득한다. 이 물리 페이지 주소(PA)중, 상위의 "0x00"은 소거 블록의 주소를 나타내지만, 제어 회로(123)에 입력되면, 이 부분은 논리 블록 주소(LBA)로서 취급되어 어드레스 변환 테이블(126)로부터 물리 블록 주소(PBA) "0x01"이 취득된다. 이것이 메모리 칩의 플래쉬 메모리(115A, 116A)내의 소거 블록(127, 128)의 주소이다.

한편, 하위 "0xB0"은 각 소거 블록(127, 128)내의 페이지 영역 (1271, 1 281)의 주소이다. 이 물리 주소를 기초로 플래쉬 메모리(115 A, 116A)가 모두 액세스 되어 각각으로부터 페이지 주소 "0x01B0"에 저장된 페이지 영역(1271, 1281)의 데이터가 판독되고, ECC에 의한 복호화가 실행되며, 페이지 버퍼(112A)에 저장된다.

한편, 외부로부터 "0x5500C"에 기입을 실시하는 경우, 우선 판독과 같은 순서로 페이지 영역(1271, 1281)의 데이터가 플래쉬 메모리(115A, 116A)로부터 판독되어 페이지 버퍼(112A)에 저장된다. 그 후 버퍼내의 소망하는 위치가 기입 데이터로 갱신된다. 갱신된 페이지 데이터는, 플래쉬 메모리(115A, 116A)에 다시 기입되지만, 그 때 판독측의 페이지 영역 (1271, 1281)을 재기입하지 않고, 어드레스 변환 테이블(121A)의 어느 물리 주소 필드에도 등록되지 않은, 미사용의 빈 페이지에 기입된다. 그러한 빈 페이지는 제어 회로(114A)측에서 관리하는 예를 들어 도 6과 같은 관리 테이블(86)이 사용되며, 물리 페이지 주소 "0x0280"이 선택된다. 이 중 상위의 "0x02"는 제어 회로(123)에 입력되면, 어드레스 변환 테이블 (126)로부터 "0x03"으로 변환되어, 합성된 주소 "0x0380"으로부터, 메모리 칩의 플래쉬 메모리(115A, 116A)의 페이지 영역(129a, 129b)에, 페이지 버퍼(112A)의 갱신이 끝난 데이터가 기입되고 전송된다. 이 때 데이터에는, ECC 부호화에 의한 패리티 비트가 추가된다.

한편, 제어 회로(114A)는, 어드레스 변환 테이블(121A)의 논리 페이지 주소 "0x5500"에 대응하는 물리 페이지 주소의 필드에, "0x0280"을 등록하고, 원래의 물리 페이지 주소 "0x00B0"은 무효화 페이지 영역으로서 관리한다. 어느 페이지 주소도 제어 회로(123)의 변환을 거치면, 플래쉬 메모리(115A, 116A)상의 올바른 페이지 영역을 가리키게 된다.

제 6의 실시 형태에서, 제어 회로(114A)에 의해 도출되는 물리 페이지 주소는, 제어 회로(123)에 의해서 일대일의 블록 레벨의 변환을 받지만, 그 때 동일 블록내의 각 페이지의 상대 위치는 전혀 변함없다. 따라서 제어 회로(114A)로부터, 제어 회로(123)에 의해 실행된 각종 변환은 블랙 박스로 간주해도 좋다. 즉, 제어 회로(114A)는, 메모리 칩의 플래쉬 메모리 (115A, 116A)와 제어 회로(123)를 일체화한 플래쉬 메모리 장치로 간주하고, 독립적인 논리에 의해 임의의 액세스를 실시해도, 불일치성은 발생하지 않는다. 이것은 ECC에 의한 부호화, 복호화에 대해서도 동일하다. 그 결과, 제어 회로(114A)는, 제 3의 실시 형태와 완전히 같은 알고리즘에 의해 무효화 영역의 회복 처리를 실시하는 것이 가능해진다.

다음에, 소거 블록의 소거를 실행하기 전에, 각 데이터 영역마다, 적어도 영역의 일부를 판독하고, 판독한 값으로부터 데이터 영역에 유효한 데이터가 기록되어 있는지 아닌지를 판정하고, 판정의 결과, 유효한 데이터가 기록되어 있는 경우에, 데이터 영역의 데이터를 다른 소거 블록에 복사하는 기능을 가지는 기억장치에 대해 설명한다. 예를 들어, 도 5의 제 1 실시 형태는, 무효 영역을 회복시킬 때, 각 데이터 영역이 무효인가 유효인가에 대한 판정 테이블(86)의 ?PageState?플래그를 사용했다. 그러나 이러한 플래그를 데이터 영역마다 마련하면, 테이블(86)은 꽤 큰 RAM 용량을 점유 하게 된다. 따라서, 이러한 플래그는 폐지하고, 다른 방법으로 데이터 영역의 유효 판정을 행하는 것이, RAM 용량을 절약할 수 있어 비용적으로 유리하다. 본 기억장치는 그러한 요청으로부터 고안되었다.

도 23은, 제 7의 실시 형태와 관련되는 컴퓨터 시스템(기억 시스템)의 구성예를 나타내는 도면이다. 또한, 도 24는, 제 7의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블 및 주소 검색 테이블의 구성예를 나타내는 도면이다.

컴퓨터 시스템(150)은, 인터페이스 회로(I / F)(151), 제어 회로(152), 내부 버스(153), 페이지 버퍼(154), NAND형 플래쉬 메모리(155, 156), 제어 회로(157), 메모리 버스(158), 및 호스트 시스템(159)을 포함하여 구성되어 있다.

이상의 구성요소 가운데, 인터페이스 회로(I / F)(151), 제어 회로(152), 내부 버스(153), 페이지 버퍼(154), NAND형 플래쉬 메모리(155, 156), 제어 회로(157), 메모리 버스(158)에 의해 파일 기억 시스템(160)이 구성되어 있다. 또한, 파일 기억 시스템(160)에서, 페이지 버퍼(154), NAND형 플래쉬 메모리(155, 156), 제어 회로(157), 메모리 버스(158)가 플래쉬 메모리 장치(161)를 구성하고 있다. 또한, 제어 회로(152)는, RAM(162)을 내장하고 있으며, RAM(162)에는 작업 영역 (163), 어드레스 변환 테이블 (164), 및 주소 검색 테이블(165)이 형성된다.

파일 기억 시스템(160)의 내부에 있어서는, 32비트의 메모리 버스(158)에, 16비트의 입출력을 가지는 2칩의 8Gb NAND형 플래쉬 메모리(155, 156)가 병렬 접속되어 있다. 2칩의 플래쉬 메모리(155, 156)는 판독과 기입시에, 동시 병렬로 액세스 된다. 즉, 메모리 버스(158)는 16비트 버스를 2 채널 구비한 구성으로 되어 있다. 각각의 플래쉬 메모리(155, 156)는 판독 또는 기입을 위한 액세스를 예를 들어 4kB의 페이지 단위로 실시한다. 따라서, 실제 페이지 사이즈로서는 8kB가 일괄적으로 액세스 된다. 페이지 버퍼(154)는 액세스된 페이지 영역의 데이터를 일시 기억하는 페이지 버퍼이다. 플래쉬 메모리 (155, 156)와 페이지 버퍼(154)와의 사이에서의 데이터의 교환은, 제어 회로(157)에 의해 제어되고 있다. 제어 회로(157)는, 필요에 따라서 전송 데이터에 ECC 부호화에 의한 에러 보정을 실행한다. 두 개의 플래쉬 메모리(155, 156)는 페이지 버퍼(154)를 통해 파일 기억 시스템(160)의 내부 버스(153)의 사이에서 데이터를 입출력한다. 즉, 상부 회로군은 실질적

으로 하나의 플래쉬 메모리 장치(161)를 구성하며, 파일 기억 시스템(160)의 내부 버스(153)에 접속되어 있는 것으로 간주할 수 있다. 그 총용량은 16Gb(2GB)이며, 실페이지 사이즈는 8kB이다. 즉, 장치내에는 256k개의 페이지 데이터가 저장된다.

게다가, 내부 버스(153)에는 인터페이스 회로(151), 및 제어 회로 (152)가 접속되어 있다. 인터페이스 회로(151)는 ATA나 PCI 익스프레스등의 규격에 따라서 호스트 시스템(159)과의 사이에서, 데이터나 명령의 송수신을 실시한다. 제어 회로 (152)는 파일 기억 시스템(160)의 내부에서 페이지 버퍼(154)와 인터페이스 회로(151)의 사이에서 데이터의 송수신을 관리한다. 제어 회로(152)에 내장된 RAM(162)에는, 프로그램을 실행하기 위한 코드 영역이나 작업 영역(163)이 설치되어 제공되어 있다. 게다가, 페이지 단위의 가상 주소를 관리하는 어드레스 변환 테이블(164) 및 정상적인 빈 블록을 검색하는 주소 검색 테이블(165)이 구축되어 있다.

컴퓨터 시스템(150)에 대해서는, 호스트 시스템(159)은 내장하는 CPU에 의해서 제어되어 애플리케이션이나 OS의 요구에 따라, 파일 기억 시스템 (160)를 통해, 플래쉬 메모리 장치(161)에 이용자 데이터를 보존한다. 제어 회로(152)는 그 동안에 데이터 전달을 중재하며, 어드레스 변환 테이블(164)을 이용하여 어드레스 변환을 수반하는 액세스 관리를 실시한다.

파일 기억 시스템(160)는 하드 디스크와 같이 512 Byte의 섹터를 액세스 단위로 한다. 장치 내부에서는 단순화를 위해, 16 진수의 주소가 다음과 같이 할당된다고 가정한다. 예를 들어, 외부 입력 주소가 "0x05502C"인 경우, 상위 20비트의 "0x05502"는 페이지 주소이며, 최대 1 M페이지를 관리할 수 있다. 한편, 하위 4비트의 "0xC"는 페이지 영역내의 섹터 드레스이며, 1페이지중에는 16의 섹터가 포함된다. 기억장치 (160)는 페이지 버퍼(154)내의 데이터를 선택하므로, 1섹터 단위의 랜덤 액세스가 가능하다.

이하, 이러한 기억장치(160)의 내부 동작에 대해 설명한다.

제 7의 실시 형태에서는 페이지 단위의 가상 주소 관리가 채용되고 있다. 도 25는, 도 24의 어드레스 변환 테이블 및 검색 테이블을 사용한 데이터 액세스 처리를 설명하기 위한 플로차트이다. 파일 기억 시스템(160)의 내부에 있어서의 플래쉬 메모리 장치(161)에 대한 구체적 액세스는, 도 25의 플로차트에 따라서, 이하와 같은 순서로 실행된다.

<스텝 ST101>

호스트 시스템(159)으로부터 이용자 데이터의 액세스 명령과 함께 "0x05502C"의 섹터 어드레스가 입력되면, 제어 회로 (152)는 논리 페이지 주소부(LPA) "0x05502"를 인덱스에 어드레스 변환 테이블(164)을 참조하여, 액세스 대상인 이용자 데이터의 물리 페이지 주소(PPA) "0x060B0"을 취득한다.

<스텝 ST102>

상기 물리 페이지 주소에 의해, 플래쉬 메모리 장치(161)가 액세스 되어 페이지 버퍼(154)내에 이용자 데이터가 저장된 페이지 그룹이 판독된다. 그 후 섹터 어드레스 "0xC"에 해당하는 부분이 페이지 버퍼 (154)로부터 선택적으로 호스트 시스템(159)에 출력되어 판독 동작이 완료된다.

게다가, 데이터의 갱신은 이하와 같이 실시한다. "0x05502C"의 섹터를 갱신한다고 가정한다. 그 경우, 우선 페이지 버퍼 (154)에 소망하는 페이지를 판독할 필요가 있으며, 그 순서는 스텝(ST101, ST102)과 같다. 스텝(ST101, ST102)의 판독시와 같이, 플래쉬 메모리 장치(161)로부터 판독된 소망하는 데이터를 페이지 버퍼(154)에 저장한다.

<스텝 ST103>

페이지 버퍼(154)상에서 소망하는 섹터 부분을 갱신한다.

<스텝 ST104>

갱신한 이용자 데이터의 플래쉬 메모리 장치(161)에 대한 기입처(write destination)로서, RAM(162)내에 상주한 주소 검색 테이블(165) 및 레지스터(166)(도 24)로부터, 적당한 페이지 영역의 물리 페이지 주소(PPA)가 선택된다. 여기에서는 간단히 하기 위해, 물리 페이지 주소는, 상위 12비트의 물리 블록 주소부(PBA)와 하위 8비트의 페이지 오프셋부로 구성된다고 가정한다. 이 때 각 소거 블록은 256페이지(28)로 구성된다. 주소 검색 테이블(165)에는, 각 블록마다 그것들이 현재 사용되고 있는지, 혹은 소거가 끝난 빈 상태인지가, "UsedFlag"로 마크되어 있다. 게다가, 결함 블록의 경우,

"DefectFlag"가 "1"이 되고 있다. 데이터의 기입은, 소거가 끝난 블록의 선두 페이지로부터 순서대로 실행되어 간다. 이전에 기입된 페이지의 물리 주소는, 레지스터(166)에 보존되며, 기입처로서, 그것으로부터 증가된 페이지가 차례차례 선택된다. 선택이 블록의 말미에 이르면, 그 이후의 블록의 "UsedFlag"와 "DefectFlag"가 스캔 되어 다음의 미사용된 우량품의 블록이 검출되고, 그 선두 페이지가 선택된다. 레지스터(166)의 값은 현재 "0x0AA01"이며, 제어 회로(152)는 우선 사용자 데이터의 기입처로서 그것을 증가시킨 물리 페이지 주소 "0x0AA 02"를 선택한다.

<스텝 ST105>

상기 물리 페이지 주소에 의해, 플래쉬 메모리 장치(161)가 액세스 되어, 페이지 버퍼(154)내의 사용자 데이터가 플래쉬 메모리 장치(161)에 일괄 기입된다. 기입이 완료되면, 어드레스 변환 테이블(164)이 갱신되어 논리 페이지 주소(LPA) "0x05502"에 대응하는 물리 페이지 주소(PPA)는 "0x0AA02"로 갱신된다. 이에 수반하여 구 물리페이지 주소 "0x060B0"에 해당하는 페이지 영역은 무효가 된다.

<스텝 ST106>

게다가, 무효화된 구 데이터의 페이지 "0x060B0"의 예비 영역에 소정의 값을 기입하고, 무효화 시킨다. 이러한 작업은 블록내의 페이지의 기입순서가 제한된 플래시에서는 실시할 수 없다. 이러한 것을 해결하기 위한 방법은 후술한다.

그런데, 상술한 주기형 기입을 실시했을 경우, 갱신전의 데이터가 저장되어 있던 물리 페이지 주소 "0x060B0"에 해당하는 페이지 영역은 어드레스 변환 테이블(164)의 물리 주소 필드로부터 삭제되어 밖으로부터 액세스 할 수 없게 된다. 즉 무효화된다. 그러나 그것들에는 데이터가 기입되어 있으므로 그대로 빈 영역으로 사용될 수 없다. 상술한 바와같이 재기입 동작을 몇 번이나 반복하면, 많은 무효 페이지 영역이 발생한다. 그것들은 재차 빈 영역으로서 사용할 수 있도록, 소거하여 회복시킬 필요가 있다. 또한 그 경우, 소거 블록 "0x060"에 남겨진 다른 유효 데이터는 백업시키지 않으면 안 된다.

이러한 회복 처리는, 예를 들어 우선 대상 블록내의 유효 데이터를, 갱신폰과 같이 일단 페이지 버퍼에 읽어내고 나서, 다른 블록의 빈 영역에 추가형 재기입 동작에 의해 재기입하여 실질적인 백업을 행하면 좋다. 즉, 유효 페이지를 임시적으로 갱신하여, 그 원래 영역을 모두 무효화한다. 그 후, 대상 블록을 소거함으로써, 회복 처리가 실시된다.

도 26a~도 26c는, 제 7의 실시 형태와 관련되는 무효 영역의 회복 순서를 설명하기 위한 도면이다. 도 26a~도 26c에는 소거 블록(170, 171) 내부의 유효 데이터를 백업하고, 무효 페이지 영역을 실질적으로 회복시키는 순서를, 개념적으로 도해하고 있다.

순서 <11>

상술한 바와같은 주기형에 의한 재기입이 진행되고, 소거 블록(170, 171)에는, 한 번 데이터가 기입된 후, 갱신에 의해서 무효화된 페이지 영역군(175, 177)이, 유효한 페이지 영역군(174, 156, 178)과 공존하고 있다고 가정한다. 여기서 유효 페이지 영역의 데이터를 남기면서, 무효화 영역을 빈 영역으로 회복시킬 필요가 있다. 한편, 블록(172, 173)은 현재 주기형을 위한 빈 영역으로서 사용하고 있는 소거 블록이며, 페이지 영역(183)까지 기입이 이루어지고 있다.

순서 <12>

유효한 페이지 영역군(174, 176, 178)을 소거 블록(172, 173)내의 빈영역(179, 180, 181)에 차례로 복사해 나간다. 한편, 무효화 페이지 영역군(175, 177)은 복사하지 않는다. 유효 페이지와 무효 페이지의 판정은, 소거 블록(172, 173)으로부터 각 페이지의 예비 영역을 우선 도 23의 페이지 버퍼(154)에 판독하고, 그 값에 근거하여 실시하지만, 그 자세한 것은 후술한다. 판정 결과가 유효 페이지인 경우, 그 대상 페이지내의 데이터 전체를 페이지 버퍼(154)에 판독하여, 소거 블록(172, 173)에 기입하는 동시에, 어드레스 변환 테이블(164)을 갱신한다. 즉, 각각의 페이지 영역의 논리 주소에 대응하는 물리 주소 필드에, 복사처(copy destination)의 물리 페이지 주소를 등록해 나간다. 이 조작은, 즉 유효 페이지 영역군(174, 176, 178)을 주기형 방식에 의해 재기입하는 작업과 동일하다. 실제로는 고쳐 쓰지 않고 복사할 뿐이지만, 이 작업에 의해서 소거 블록 (170 및 171)내의 모든 페이지는 무효화되며, 유효 페이지 영역의 데이터는 소거 블록(172, 173)에 실질적으로 백업된다.

순서 <13>

소거 블록(170, 171)을 소거한다. 이에 의해서 그 내부는 모두 빈영역이 되어, 주기형 동작에 의해 사용하는 것이 가능하게 된다. 이에 의해, 무효화 영역(175, 177)은 효과적으로 회복된다.

이와 같이, 무효 페이지 영역의 회복 처리는, 각 유효 페이지 영역의 복사에 의한 백업 처리와, 원래 소거 블록의 소거로 구성된다. 또한, 회복 대상이 되는 소거 블록내의 유효 페이지에 통상의 갱신 처리와 같은 순서대로 더미 갱신을 실시하고, 그에 의해 페이지의 백업을 실행하면, 회복 처리시의 유효 페이지 백업을 통상의 기입 알고리즘에 통합시키는 것이 가능하다. 이 경우, 제어가 용이하게 되는 것만이 아니고, 플래쉬 메모리에 대한 기입의 표준화등, 기입시의 신뢰성 향상을 위한 각종 형태의 회복 처리에도 적용할 수 있어 기억장치의 종합적인 신뢰성을 향상시킬 수 있다.

도 27은, 제 7의 실시 형태에 있어서의 구체적인 회복 처리를 설명하기 위한 플로차트이다.

<스텝 ST111>

가비지 포인터(garbage pointer)를 참조한다. 가비지 포인터는 회복 대상이 되고 있는 소거 블록에서, 처리중인 페이지의 물리 주소(PPA)를 나타내고 있다. 회복 처리 개시는 블록의 선두를 가리키고 있다. 처리가 진행됨에 따라, 블록의 말미까지 차례차례 값이 증가되어 간다.

<스텝 ST112>

상기 가비지 포인터가 가리키는 페이지를 선택하고, 우선 그 예비 영역의 데이터를 읽어들인다. 예비 영역은, 이용자 데이터의 저장과는 별도로, 기억장치 시스템내에서 생성된 관리용 데이터를 저장하는 영역이며, 예를 들어 2kByte의 이용자 데이터에 대해서 72Byte정도의 예비 영역이 플래쉬 메모리상에 미리 설치되어 있다. 여기에 기재된 정보를 바탕으로, 선택한 페이지가 유효한가 무효인지를 판정한다.

<스텝 ST113>

선택한 페이지가 유효한가 무효인지를 판정해, 판정 결과가 유효하면, 더미 갱신 처리를 실시한다. 즉 우선 선택 페이지 전체를 페이지 버퍼에 읽어들여(ST114), 도 24의 주소 검색 테이블(166)로부터 플래쉬 메모리상의 기입처를 결정하고, 거기에 데이터를 기입한다(ST115). 즉 데이터를 다른 블록에 복사, 백업한다.

<스텝 ST116>

가비지 포인터가 블록의 말미 페이지인지 아닌지를 판정하고, 가비지 포인터가 블록의 말미 페이지이면, 가비지 포인터를 클리어하고(ST117), 대상 블록을 소거한다(ST118). 중간이라는 것을 나타내면, 가비지 포인터를 증가시키고(ST119), 스텝(ST111)으로 돌아와 다음의 페이지의 처리를 실시한다.

스텝(ST112)의 공정에서 선택 페이지의 전체 영역을 판독함으로써 스텝(ST113)의 판독 공정을 생략하는 것도 가능하다. 즉, 우선 선택 페이지의 전체 영역을 페이지 버퍼에 읽어들여, 그 예비 영역에 기재되어 있던 데이터로부터 그 페이지가 유효한가 아닌가를 판정한다. 무효이면 그것들은 모두 파기하고 스텝(ST116)으로 옮긴다. 한편, 유효하면, 이미 페이지 버퍼에는 페이지내의 전체 데이터가 판독되어 있으므로, 판독 공정을 생략 하고 더미 갱신 처리를 실시한다. 이 경우 무효 페이지에 대해서는 보다 많은 데이터를 읽어낼 필요가 있지만, 유효 페이지에 대해서는 다중의 읽기(multiple reading)를 방지할 수 있어 신속히 처리할 수 있다.

페이지내에 기재된 예비 영역의 데이터로부터, 그 페이지의 유효 / 무효를 판정하는 구체적 수법을 이하에 나타낸다.

우선, 도 25에 있어서의 스텝(ST106)과 같이, 무효가 된 페이지의 예비 영역에, 특정의 값을 플래그로서 기재해 두면, 이 판정은 즉시 실시할 수 있다. 그러나 소거 블록내의 페이지의 기입 순서에 제한이 있는 경우, 이미 한 번 데이터를 쓴 페이지 영역의 예비 영역에, 데이터를 추가로 기입할 수 있다는 보장은 없다.

이와 같이 한 번 쓴 페이지 영역의 부분 변경이 불가능한 경우의 판정 수법을, 전술한 도 24의 예에 따라 설명한다.

우선, 페이지 데이터를 쓸 때에, 예비 영역에 대응하는 논리 주소를 써 둔다. 도 24의 경우, 당초 LPA "0x05502"의 논리 주소에 대해, PPA "0x060B0"의 물리 주소가 할당되어 있었다. 즉, PPA "0x060B0"에 해당하는 플래쉬 메모리상의 페이지 영역에는, 사용자 데이터와 함께, 그 예비 영역에 논리 주소 "0x05502"가 기재되어 있다.

여기서, 호스트 제어기로부터의 명령에 의해 LPA "0x05502"의 데이터가 갱신되었을 경우, 갱신 후의 데이터는 PPA "0x0AA02"에 해당하는 페이지 영역에 기입된다. 그 때, 그 예비 영역에는 역시 논리 주소 "0x05502"가 기재된다. 한편, 어드레스 변환 테이블(164)의 LPA "0x05502"에 대응하는 물리 주소 필드는 "0x0AA02"로 갱신된다.

여기서, 원래의 데이터 영역 PPA "0x060B0"을 포함한 블록에 회복 처리를 실시해, 가비지 포인터가 후술하는 페이지로 돌아 왔다고 가정한다. 이 때, 예비 영역으로부터 논리 주소 "0x05502"가 읽힌다. 그것을 인덱스로 하여 어드레스 변환 테이블(164)을 참조하면, 대응하는 물리 주소 필드는 "0x0AA02"로 갱신되어 있기 때문에, 선택된 페이지의 물리 주소 "0x060B0"에는 일치하지 않는다. 이것을 가지고 이 페이지가 무효하다고 판정된다.

한편, 유효한 페이지이면, 그 예비 영역에 기재된 논리 주소와 물리 주소와의 조합은, 항상 어드레스 변환 테이블(164)의 조합에 일치하고 있다. 이 때에는, 취득한 논리 주소를 가지고 도 27에서 스텝(ST113)의 더미 갱신 처리와 그 이후의 처리를 실시하면 좋다. 이 경우, 통상 액세스에서는 호스트 제어기로부터 공급된 논리 주소에 데이터 갱신을 실시하는 동일한 순서에 의해, 용이하게 유효 데이터를 백업시킬 수 있다.

여기에서는 각 페이지의 논리 주소와 물리 주소와의 직접적인 대응을 판정에 사용했지만, 논리 주소에는 다양한 형태가 가능하고, 일반적으로는 호스트 제어기가 입력하는 주소 정보와 관련된 페이지 데이터의 핸들치(handle value)로 간주할 수 있다.

그런데, 여기까지는 독립한 기억장치의 내부에 회복 처리 장치를 설치하는 경우에 대해 설명해 왔다. 그러나 이러한 어드레스 변환 테이블의 관리와 회복 처리를, 호스트 제어기측의 제어로 실시하는 것도 가능하다. 그러한 컴퓨터 시스템의 예를 도 9에 도시하고 있다.

도 28은, 제 8의 실시 형태와 관련되는 컴퓨터 시스템의 구성예를 나타내는 도면이다.

도 28의 컴퓨터 시스템(190)은, 처리 유닛으로서의 CPU(191), 시스템메모리인 RAM(192), 시스템 버스(193), 브릿지 회로(194), NAND형 플래쉬 메모리(195, 196), 데이터 버스(197), 페이지 버퍼(198), 및 제어 회로(199)를, 주구성 요소로서 가지고 있다.

CPU(191)는, 32비트의 시스템 버스(93)를 통해 시스템 메모리인 RAM (192)에 접속되어 있다. 게다가, 시스템 버스(193)에는 브릿지 회로(194)가 접속되어 있다. 브릿지 회로(194)에 연결되는 32비트의 데이터 버스 (197)에는, 16비트의 입출력을 가지는 2칩의 NAND형 플래쉬 메모리(195, 196)가 병렬 접속되어 있다. 2칩의 플래쉬 메모리(195, 196)는 판독과 기입시에, 동시 병렬로 액세스 된다. 액세스된 페이지 영역을 일시 기억하는 페이지 버퍼(198) 및 제어 회로(199)는, 브릿지 회로(194)에 내장되어 있다.

브릿지 회로(194)는, CPU(191)로부터 각종 명령을 받아, 플래쉬 메모리(195, 196)와 CPU(191) 또는 시스템 메모리(192)와의 사이에서의 데이터의 교환을, 페이지 버퍼(198)를 이용해 중재한다. 또한, 필요에 따라서 전송 데이터에 ECC 부호화에 의한 에러 보정을 실시한다. 브릿지 회로 (194)가 받은 명령은, 예를 들어 플래쉬 메모리(195, 196)의 소정의 페이지에 대한 액세스 외에도, 플래쉬 메모리(195, 196)의 소정 블록의 소거, 및 소정 페이지를 지정 주소로 복사하기, 플래쉬 메모리(195, 196)의 재설정을 포함한다.

한편, 시스템 메모리(192)내에는 상기 플래시 기억 시스템을 제어하기 위한 드라이버(200)가 제공되어 있다. 이 드라이버(200)는, OS나 애플리케이션으로부터 기억장치에 대한 액세스를 수신하고, 같은 메모리내에 구성된 어드레스 변환 테이블(201)을 참조하여 액세스시의 페이지 주소를 변환한다. 또한, 데이터 갱신시에는 주소 검색 테이블(202)을 참조하여 기입처(write destination) 페이지 주소를 결정하고, 갱신용 데이터와 함께 플래쉬 메모리(195, 196)에 대한 기입 명령을 브릿지 회로(194)에 송신한다.

이러한 경우에는 CPU(191)와 시스템 메모리(102)로 구성되는 호스트 시스템(210) 자체가 도 23에 있어서의 제어 회로(152)의 역할을 수행하고, 어드레스 변환 테이블의 관리와 무효 페이지 영역의 회복 처리를 실시한다. 즉, 드라이버(200)

는 어드레스 변환 테이블(201)이나 주소 검색 테이블(202)을 위해 도 24에 도시한 테이블(164, 165)과 같은 테이블을 사용하고, OS나 애플리케이션으로부터 논리 주소를 받아 물리 주소(PPA)를 생성해 브릿지 회로(194)에 그 명령을 송신함으로써, 플래쉬 메모리(195, 196)의 각종 액세스를 실시한다.

이러한 시스템에 본 발명을 동일한 방법으로 적용하는 것이 가능하고, 호스트 시스템(210)은 플래쉬 메모리(195, 196)에 대해, 소거 블록의 소거를 실행하기 전에, 각 데이터 영역마다 이하의 처리를 실시한다.

우선, 적어도 영역의 일부를 판독하고, 그 값으로부터 데이터 영역에 유효한 데이터가 기록되어 있는지 아닌지를 판정하며, 판정의 결과 유효한 데이터가 기록되어 있으면, 데이터 영역의 데이터를 다른 소거 블록에 복사한다.

즉, 이 경우에는, 호스트 시스템(210)과 플래쉬 메모리등을, 하나의 기억 시스템을 구성하는 것으로 간주하는 것이 가능하다.

발명의 효과

이와 같이, 본 기억장치에서는, 플래쉬 메모리에 있어서의 과제 해결을 위해, 주기형의 기억 시스템에 주목하고 있다. 통상형의 재기입에서는, 유효 페이지 영역의 백업등을 시작으로 하는 장황한 처리가, 이용자의 액세스에 따라 그 때마다 발생한다. 그것은, 액세스되는 데이터 영역의 상황에 의존하는 불가피한 처리이며, 그 발생시기를 액세스와 독립적으로 제어하는 것은 곤란하다.

한편, 주기형 재기입에서는, 무효화 데이터 영역의 회복시와 같은 오버헤드가 생기지만, 그 발생시기는 이용자의 액세스 상황에 관계없이, 저장부 시스템 혹은 호스트의 형편에 따라 자유롭게 결정할 수 있다.

통상의 용도에서, 데이터 저장부는 항상 액세스 되는 것은 아니므로, 오히려 액세스가 없는 기간이 길다. 따라서, 플래쉬 메모리에 직접 액세스가 없는 기간을 추출하고, 그 기간을 이용해 상기 회복 처리를 실시하는 기능을 제공하여, 회복 처리에 수반하는 오버헤드를 외관상 소멸시킬 수 있다.

본 발명에 의하면, 플래쉬 메모리를 기억 매체로서 사용하면서, 실 사용상 통상적으로 고속의 재기입을 실현할 수 있다. 그 소거 회수도 감소시킬 수 있어 재기입 피로를 저감하고, 신뢰성이 높은 재기입이 가능하다. 게다가, 그러한 성능 향상을 얻으면서, 플래쉬 메모리에 대해 무효화된 영역의 회복도 적절히 이루어질 수 있다. 그리고, 플래쉬 메모리를 기억 매체로서 사용하면서 고속의 재기입을 실현하고, 저비용인 하드웨어로 플래쉬 메모리상의 기억 영역을 유효하게 활용할 수 있다.

본 발명의 첨부된 청구항과 그와 동등한 것들의 범위내에서 여러 가지 수정과 결합, 소결합 및 변경들이 설계요구 및 다른 인자에 따라 이루어질 수 있다는 것을 당업자들이 알 수 있다.

도면의 간단한 설명

도 1은, NAND형 플래쉬 메모리의 내부 구성예를 나타내는 도면이다.

도 2는, 기억장치를 구성하는 플래쉬 메모리의 개념도를 나타내는 도면이다.

도 3a, 및 도 3b는, 예를 들어 소거 블록내의 1페이지만을 고쳐 쓸 때의 순서의 일례를 설명하기 위한 도면이다.

도 4a, 및 도 4b는, 어드레스 변환 테이블을 이용한 주기형(additional rewriting) 기억 시스템에 있어서의 관리 방법의 일례를 설명하기 위한 도면이다.

도 5는, 제 1의 실시 형태와 관련되는 기억장치를 나타내는 구성도이다.

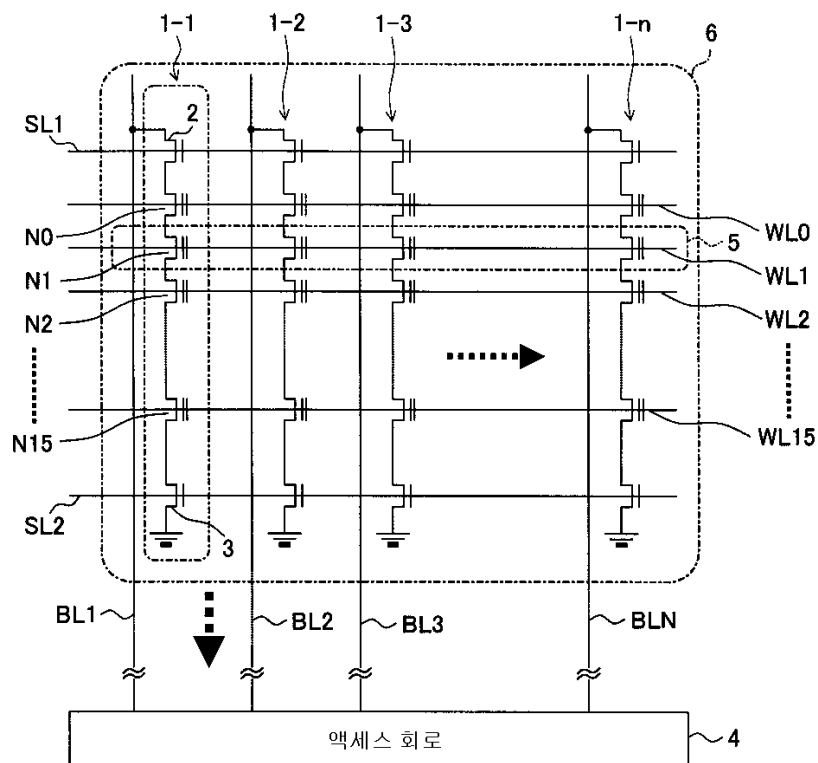
도 6a, 6b는, 제 1의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블 및 페이지 상태의 관리 테이블의 구성예를 나타내는 도면이다.

도 7a - 도 7c는 제 1의 실시 형태의 무효 영역의 회복 순서를 설명하기 위한 도면이다.

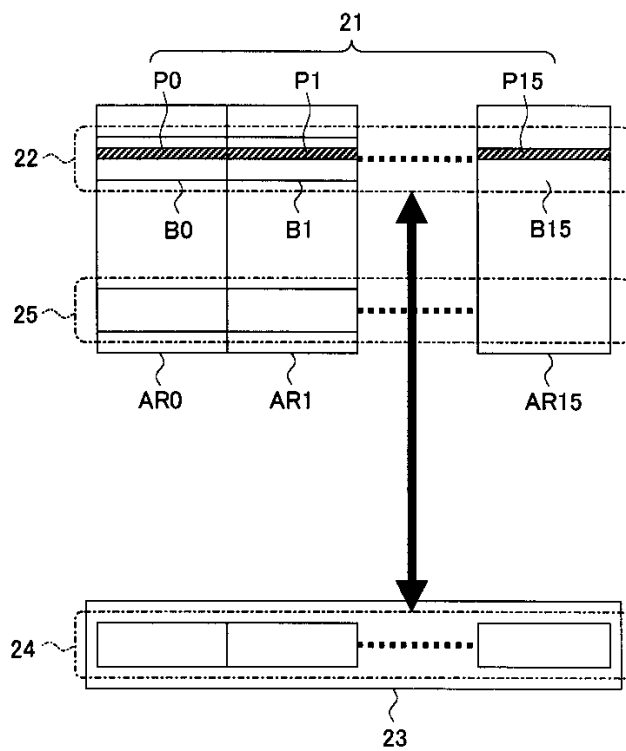
- 도 8은, 제 1의 실시 형태의 장치가 회복 처리를 실행하는 제어 순서의 예를 나타내는 플로차트(flow chart)이다.
- 도 9는, 제 1의 실시 형태의 변형예이며, 무효 페이지 영역 회복의 분할 단위 처리를 설명하기 위한 플로차트이다.
- 도 10은, 제 2의 실시 형태와 관련되는 회복 처리를 설명하기 위한 플로차트이다.
- 도 11은, 제 2의 실시 형태의 변형예이며, 무효 페이지 영역 회복의 분할 단위 처리를 설명하기 위한 플로차트이다.
- 도 12는, 제 3의 실시 형태와 관련되는 기억장치를 나타내는 구성도이다.
- 도 13은, 제 3의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블의 구성예를 나타내는 도면이다.
- 도 14는, 제 3의 실시 형태와 관련되는 회복 처리를 설명하기 위한 플로차트이다.
- 도 15는, 제 3의 실시 형태의 변형예이며, 무효 페이지 영역 회복의 분할 단위 처리를 설명하기 위한 플로차트이다.
- 도 16은, 제 3의 실시 형태의 제 2의 변형예이며, 무효 페이지 영역 회복의 분할 단위 처리를 설명하기 위한 플로차트이다.
- 도 17은, 제 4의 실시 형태에 있어서의 무효 영역의 회복 처리를 설명하기 위한 플로차트이다.
- 도 18은, 제 5의 실시 형태와 관련되는 컴퓨터 시스템의 구성예를 나타내는 도면이다.
- 도 19는, 제 5의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블의 구성예를 나타내는 도면이다.
- 도 20은, 프리엠프티브(preemptive)한 멀티태스킹 시스템을 이용한 회복 처리의 일례를 나타내는 플로차트이다.
- 도 21은, 제 6의 실시 형태를 나타내는 도이며, 그러한 ECC 처리와 결합 블록 관리를 도입한, 제 3의 실시 형태(도 12)의 장치 구성상의 응용 변형예를 나타내는 도면이다.
- 도 22a, 및 도 22b는, 제 6의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블 및 페이지 상태의 관리 테이블의 구성예를 나타내는 도면이다.
- 도 23은, 제 7의 실시 형태와 관련되는 컴퓨터 시스템(기억 시스템)의 구성예를 나타내는 도면이다.
- 도 24는, 제 7의 실시 형태와 관련되는 제어 회로의 어드레스 변환 테이블 및 주소 검색 테이블의 구성예를 나타내는 도면이다.
- 도 25는, 도 24의 어드레스 변환 테이블 및 검색 테이블을 사용한 데이터 액세스 처리를 설명하기 위한 플로차트이다.
- 도 26a - 도 26c는, 제 7의 실시 형태와 관련되는 무효 영역의 회복 순서를 설명하기 위한 도면이다.
- 도 27은, 제 7의 실시 형태에 있어서의 구체적인 회복 처리를 설명하기 위한 플로차트이다.
- 도 28은, 제 8의 실시 형태와 관련되는 컴퓨터 시스템의 구성예를 나타내는 도면이다.

도면

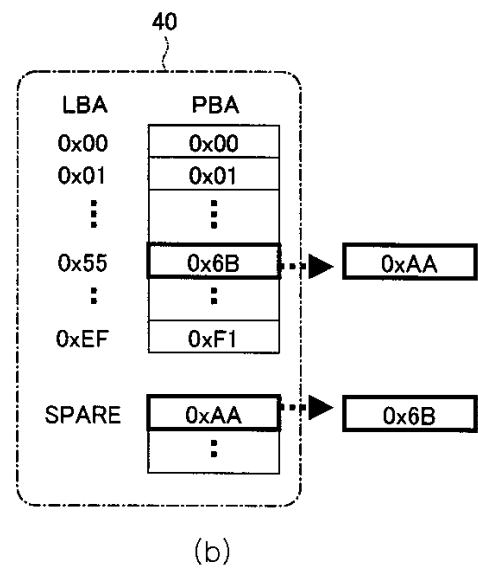
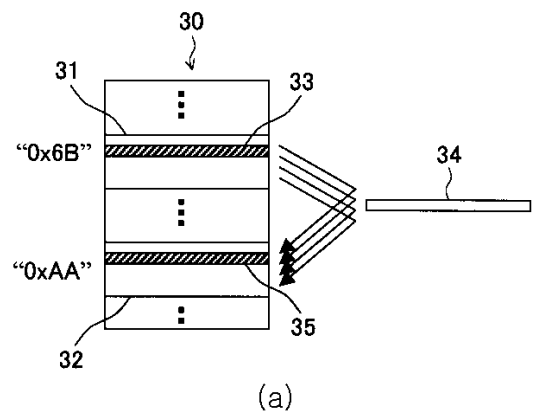
도면1



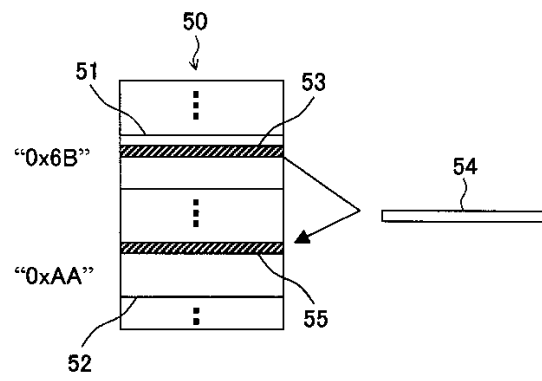
도면2



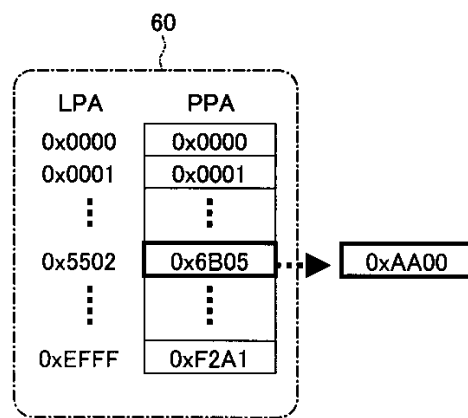
도면3



도면4

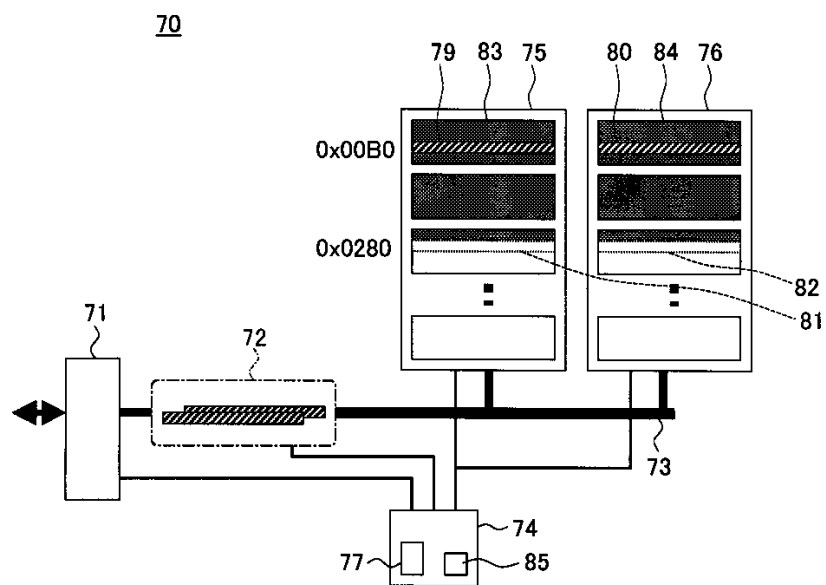


(a)

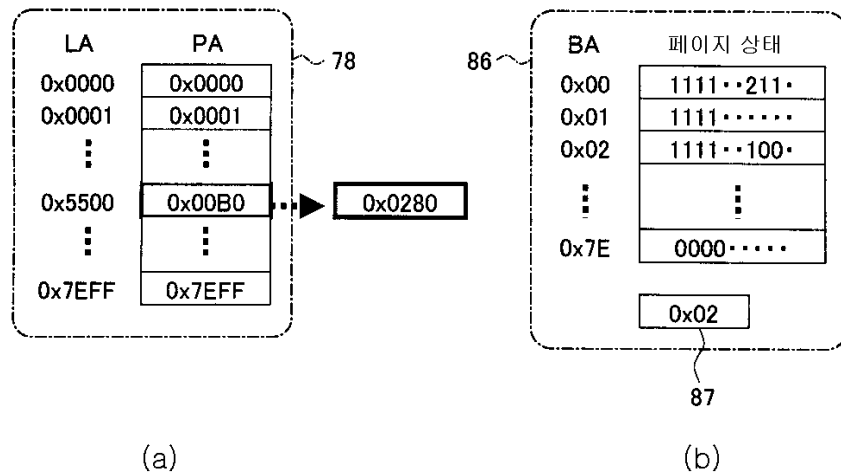


(b)

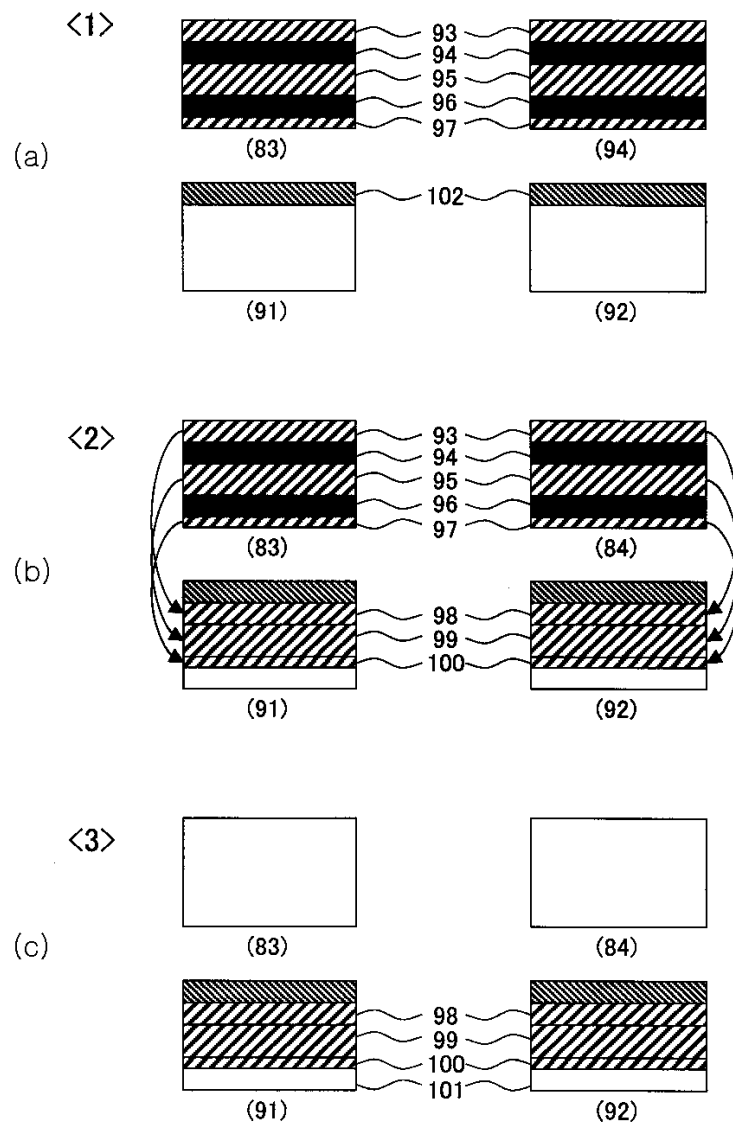
도면5



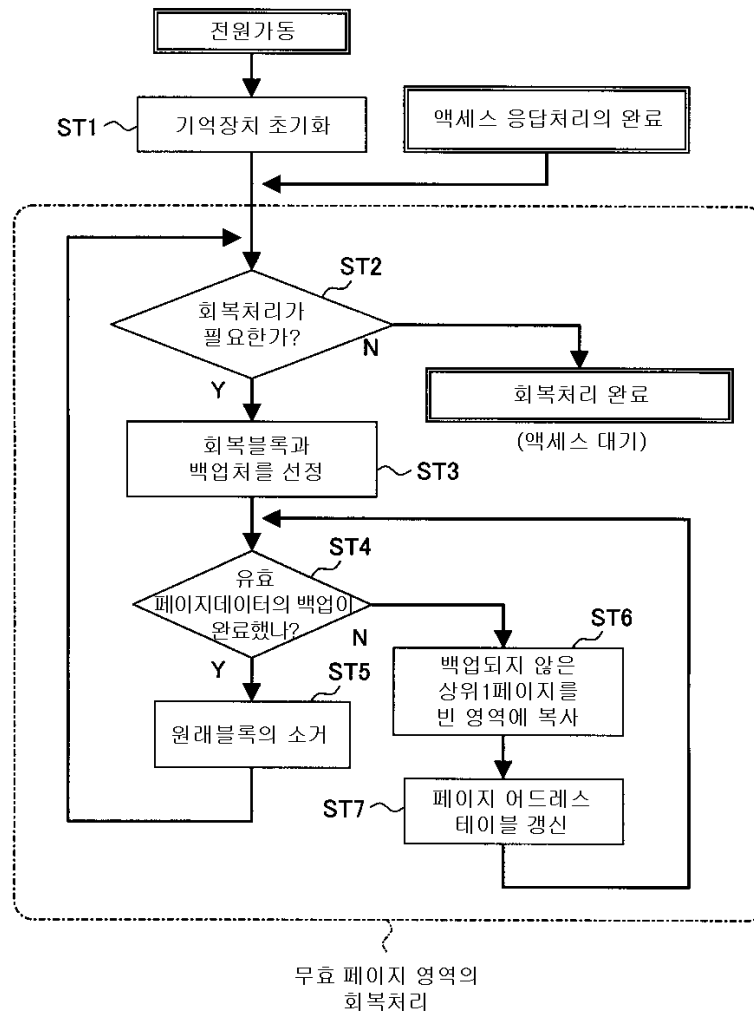
도면6



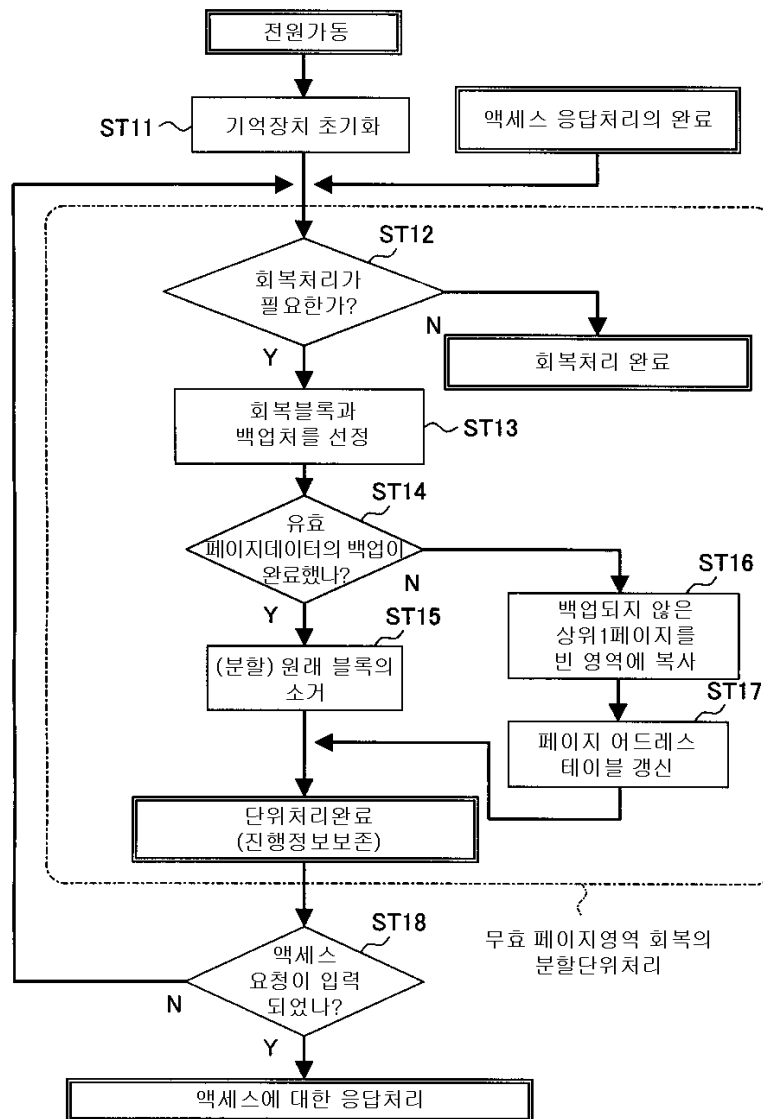
도면7



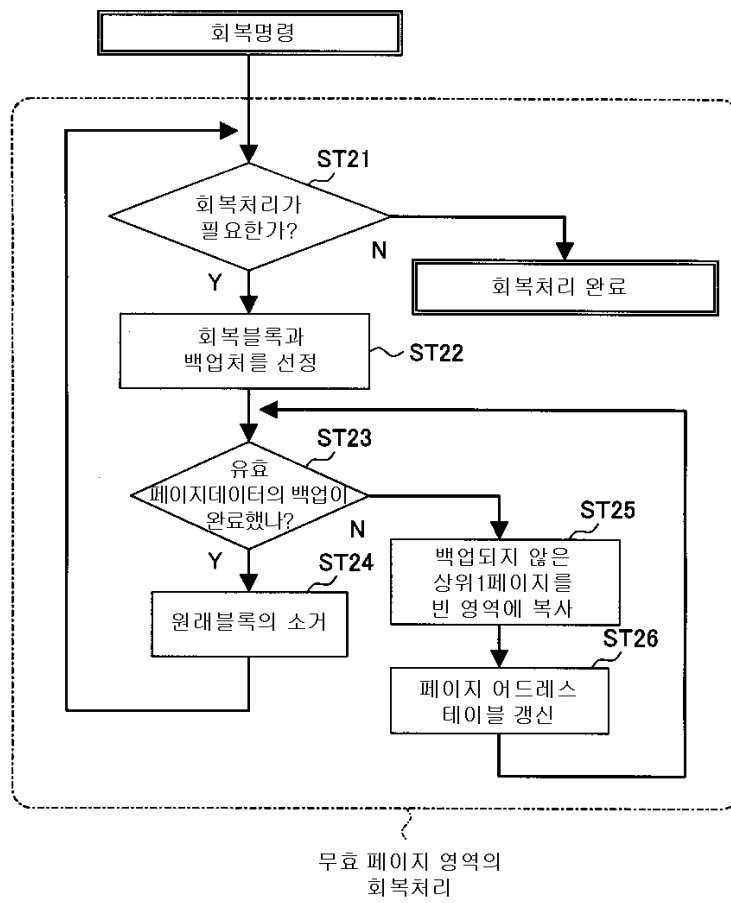
도면8



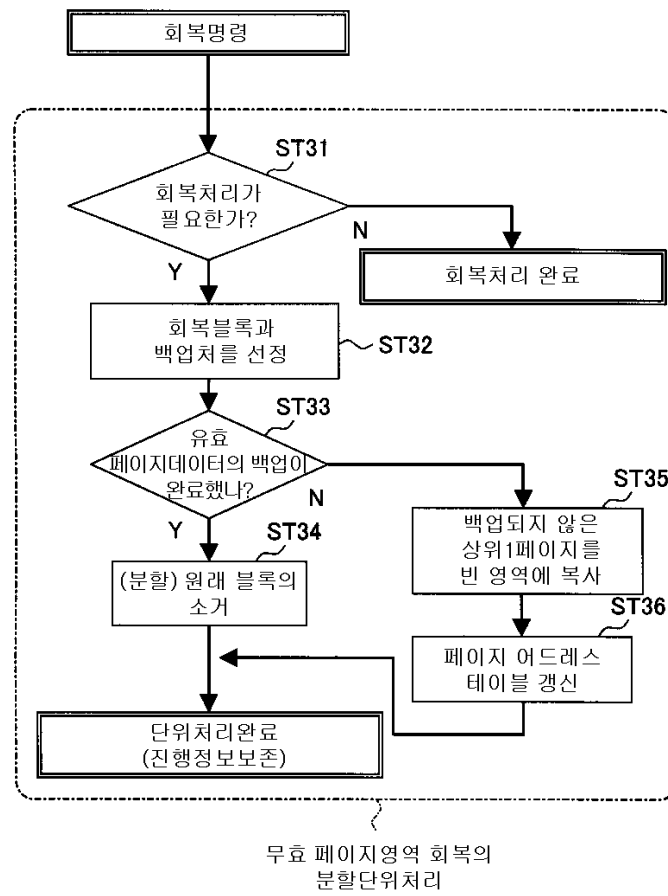
도면9



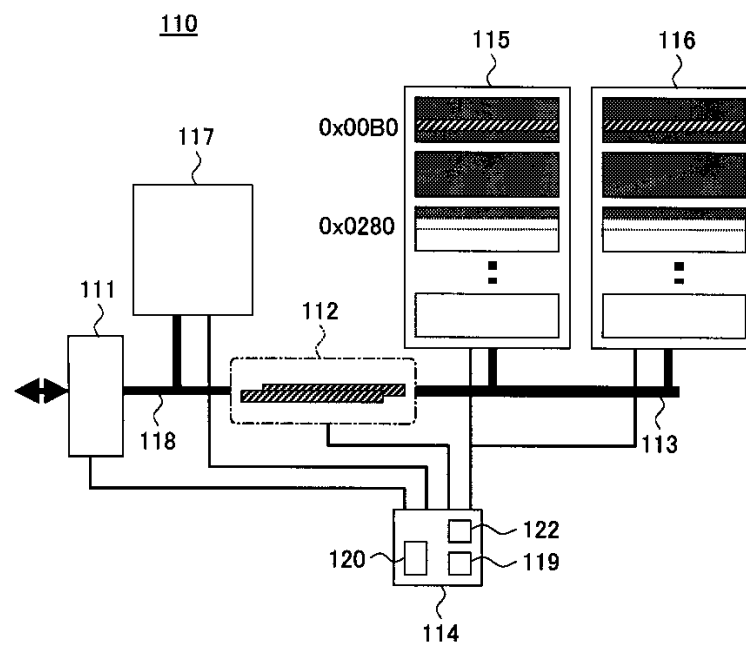
도면10



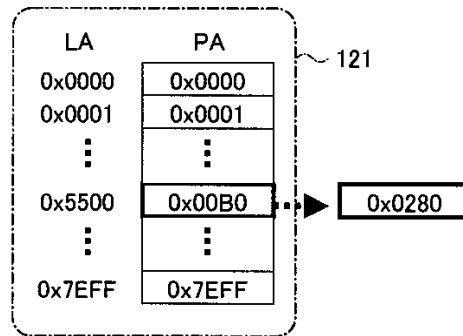
도면11



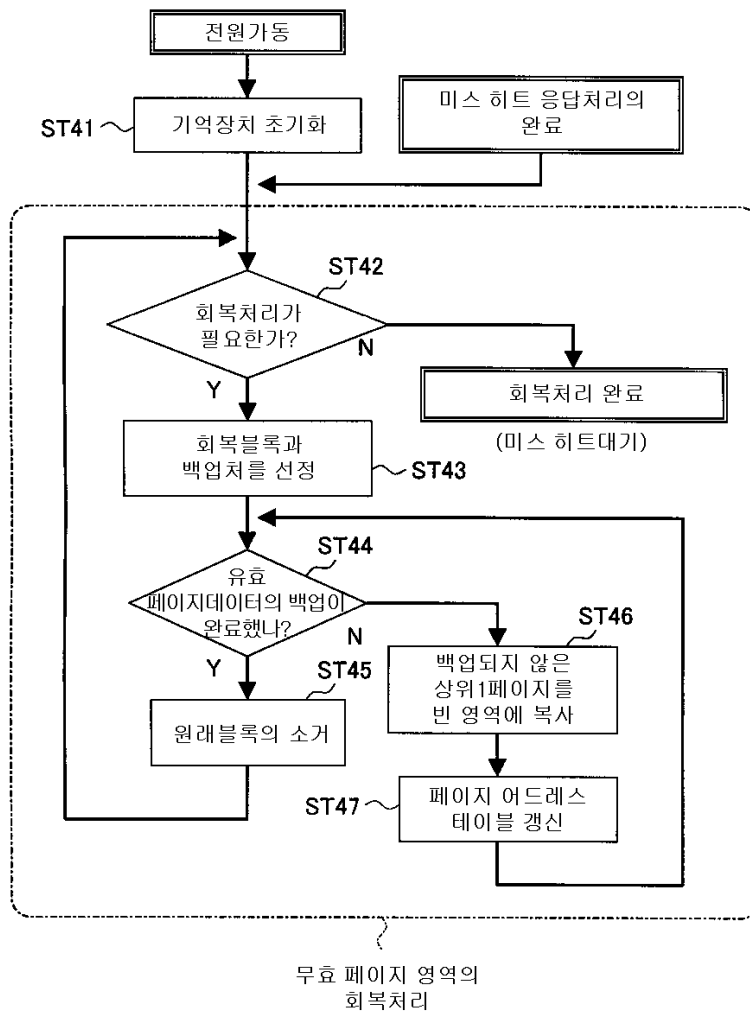
도면12



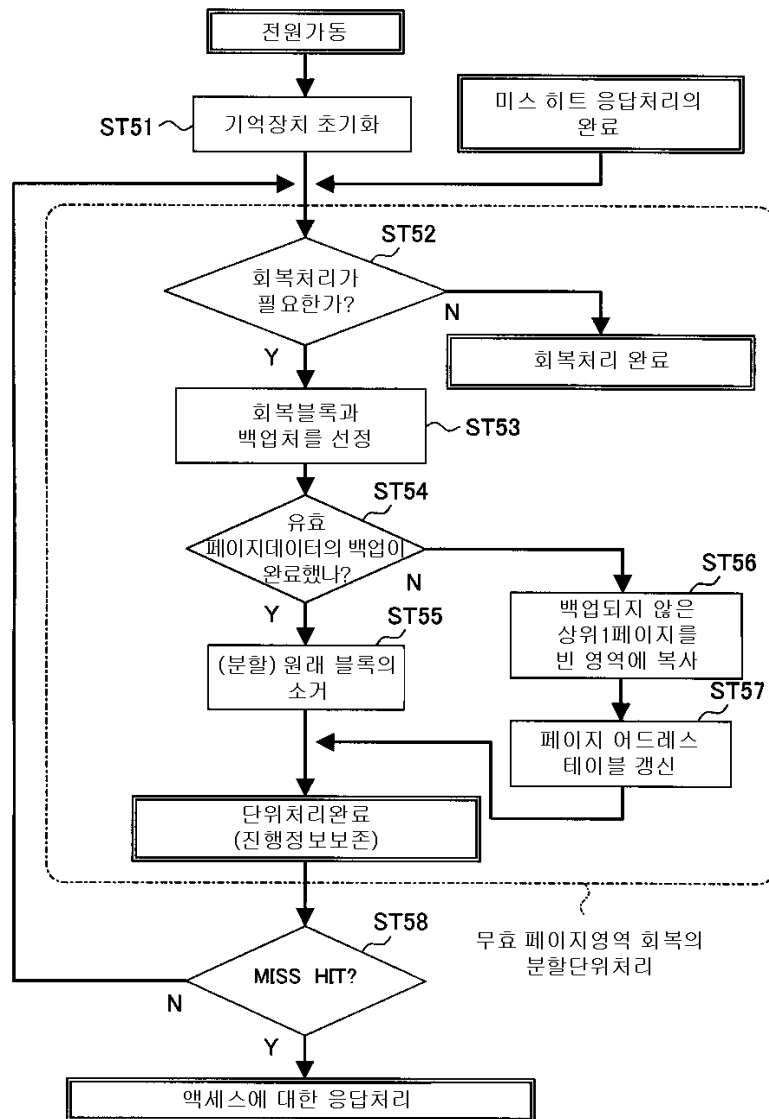
도면13



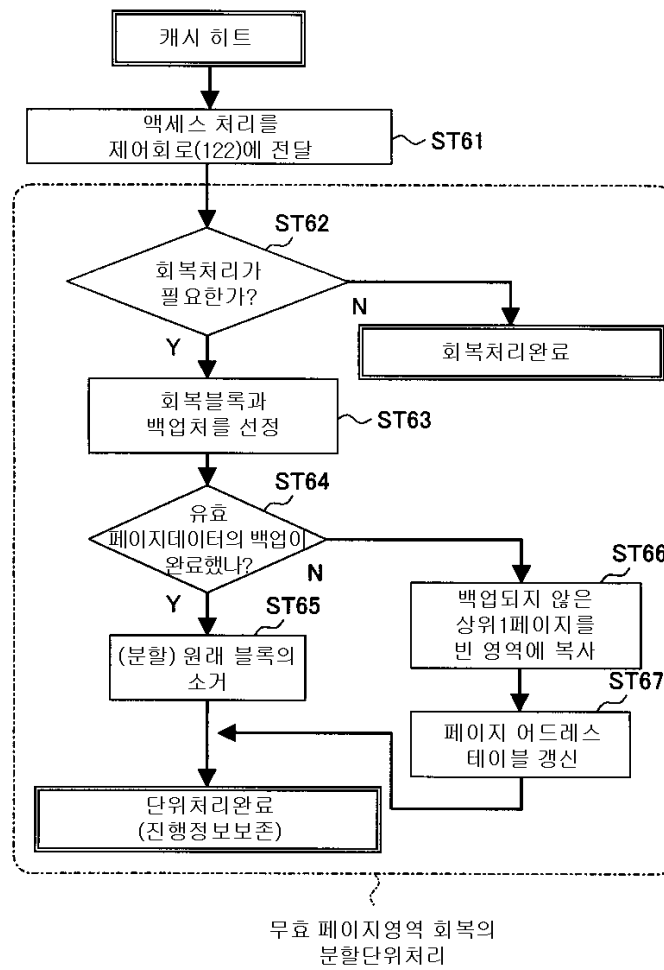
도면14



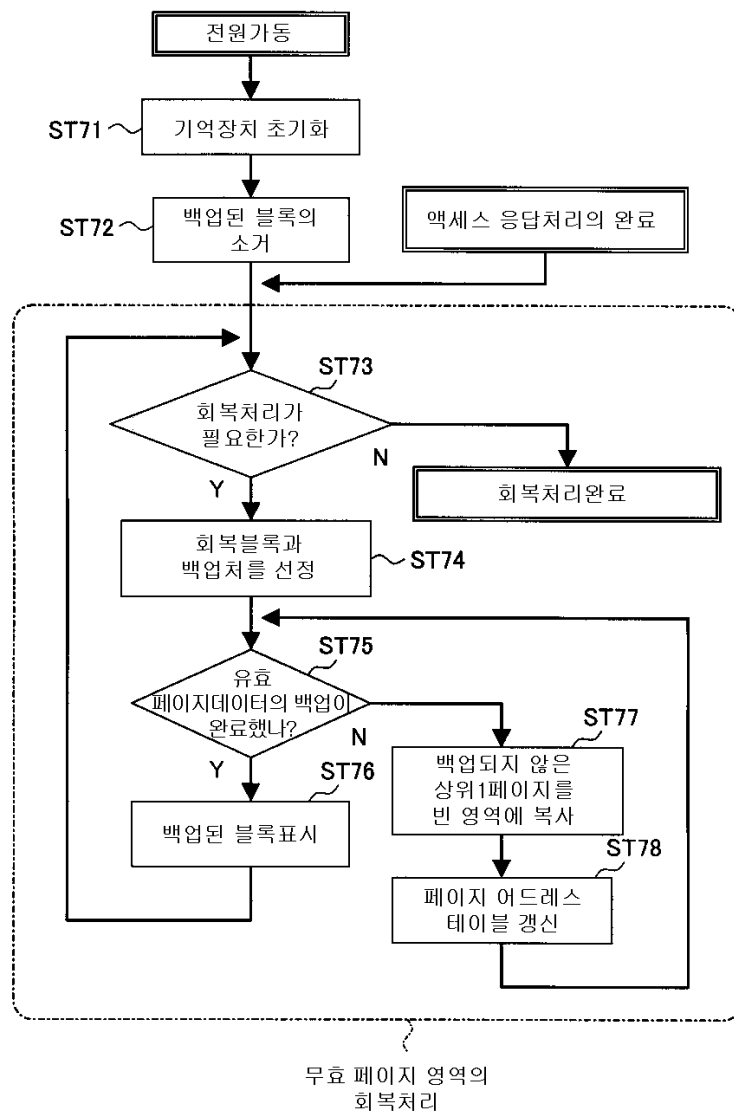
도면15



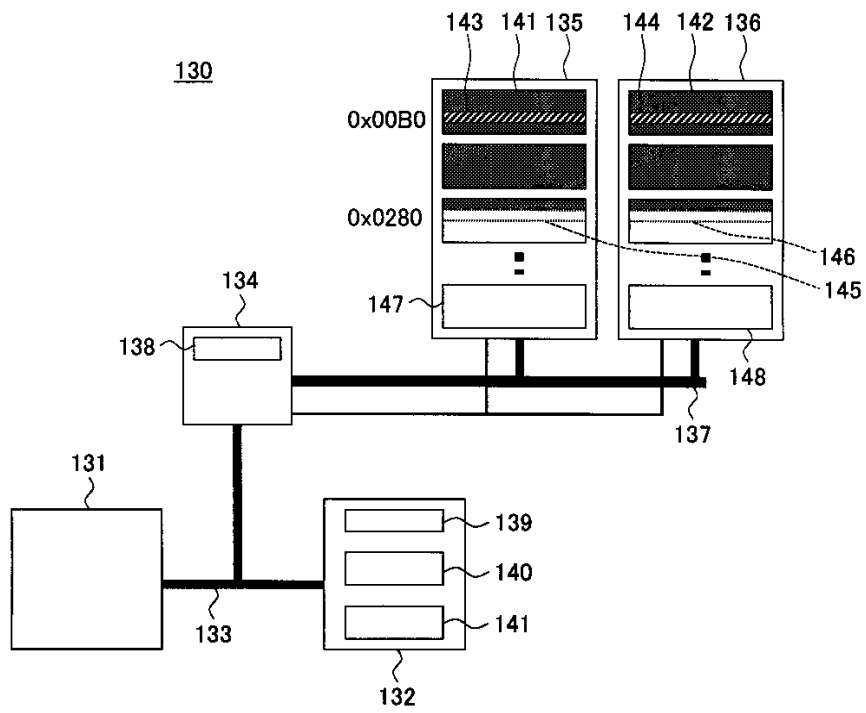
도면16



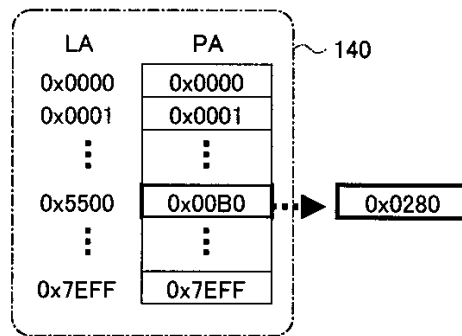
도면17



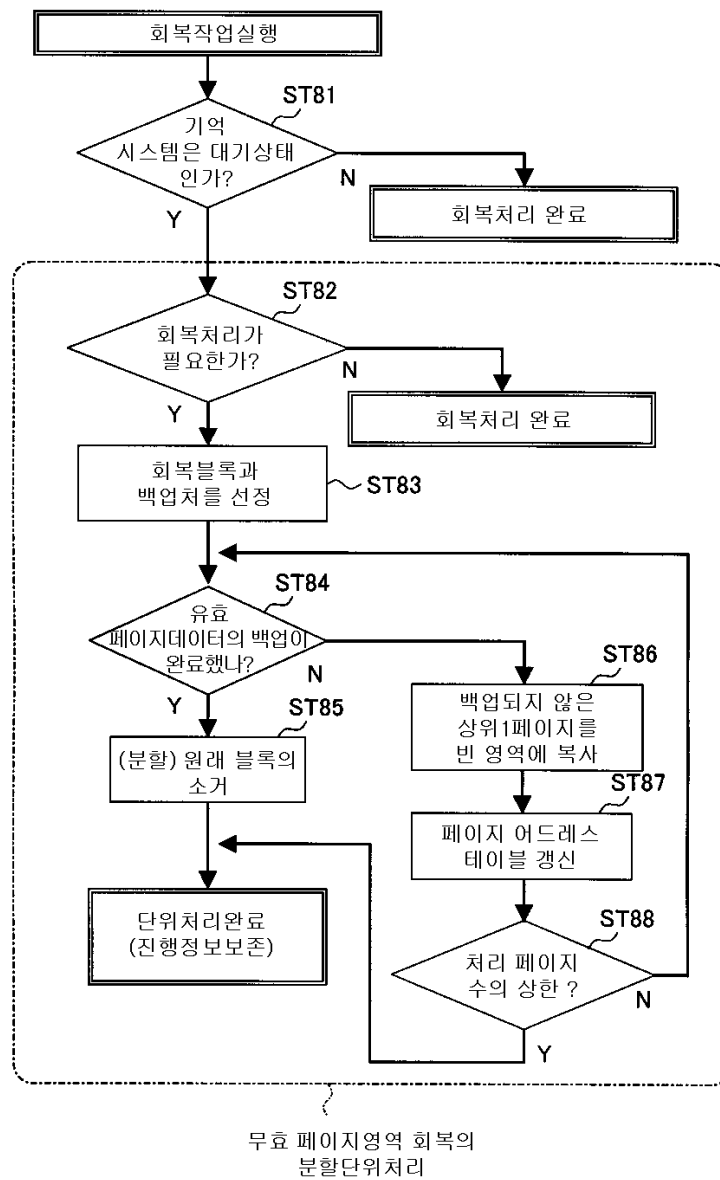
도면18



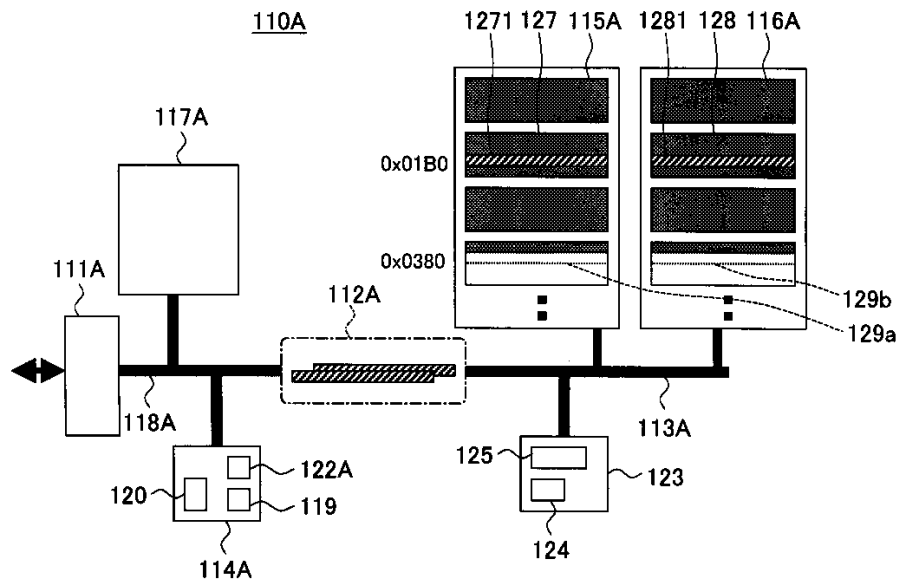
도면19



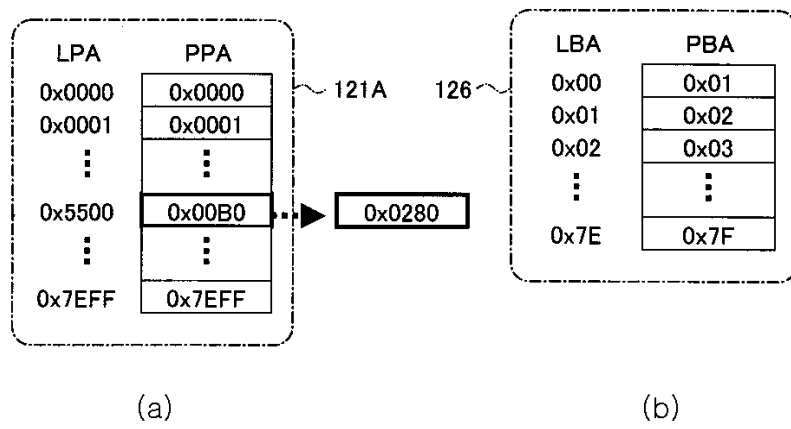
도면20



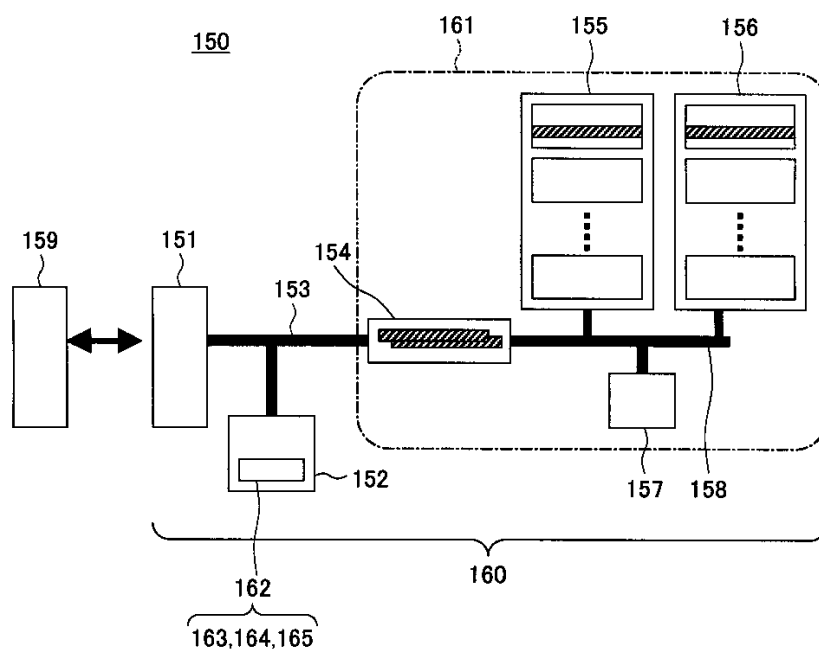
도면21



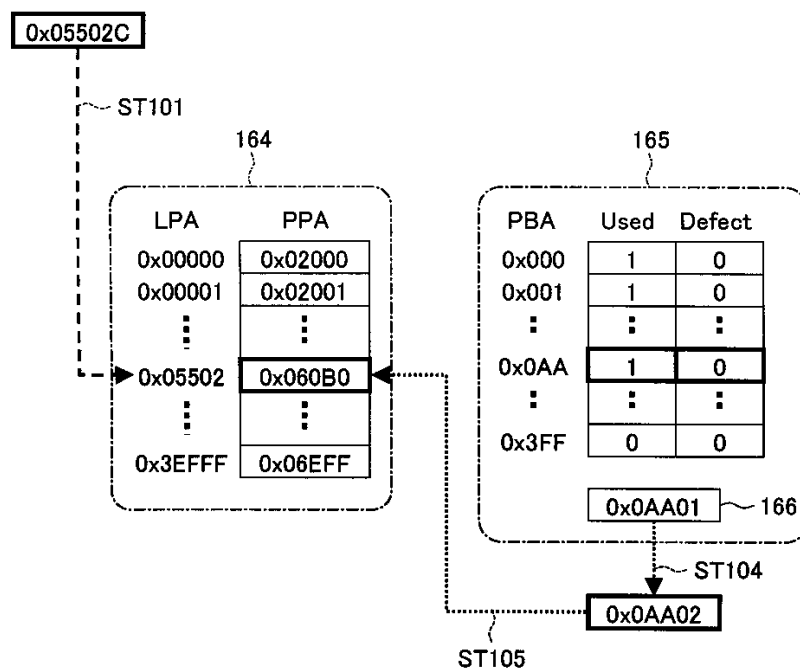
도면22



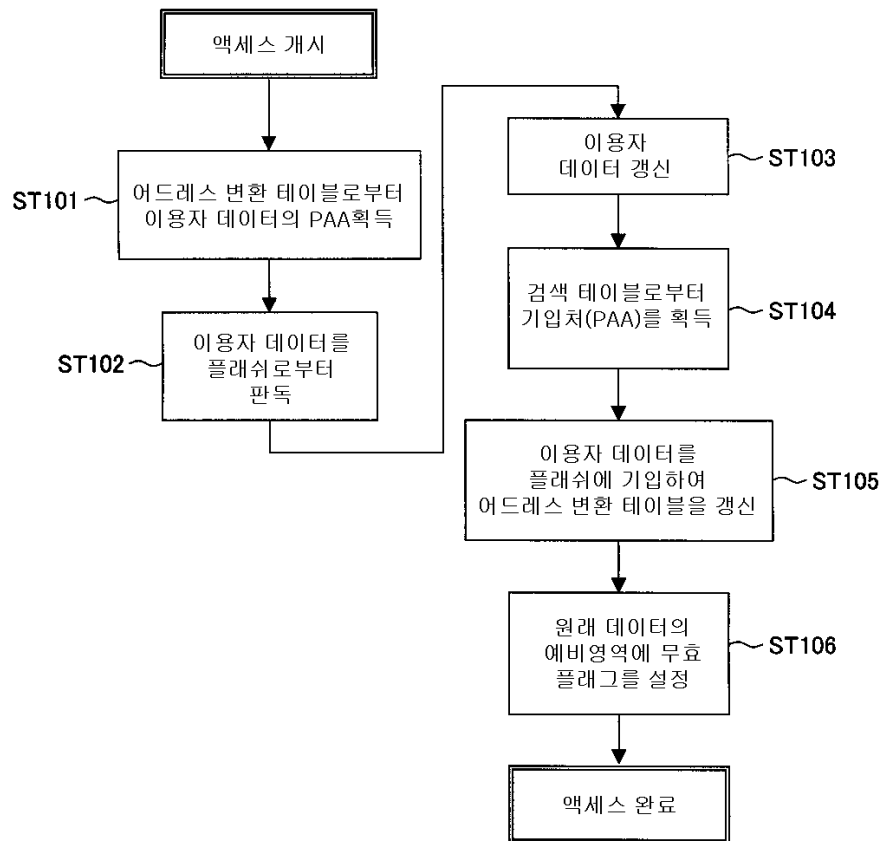
도면23



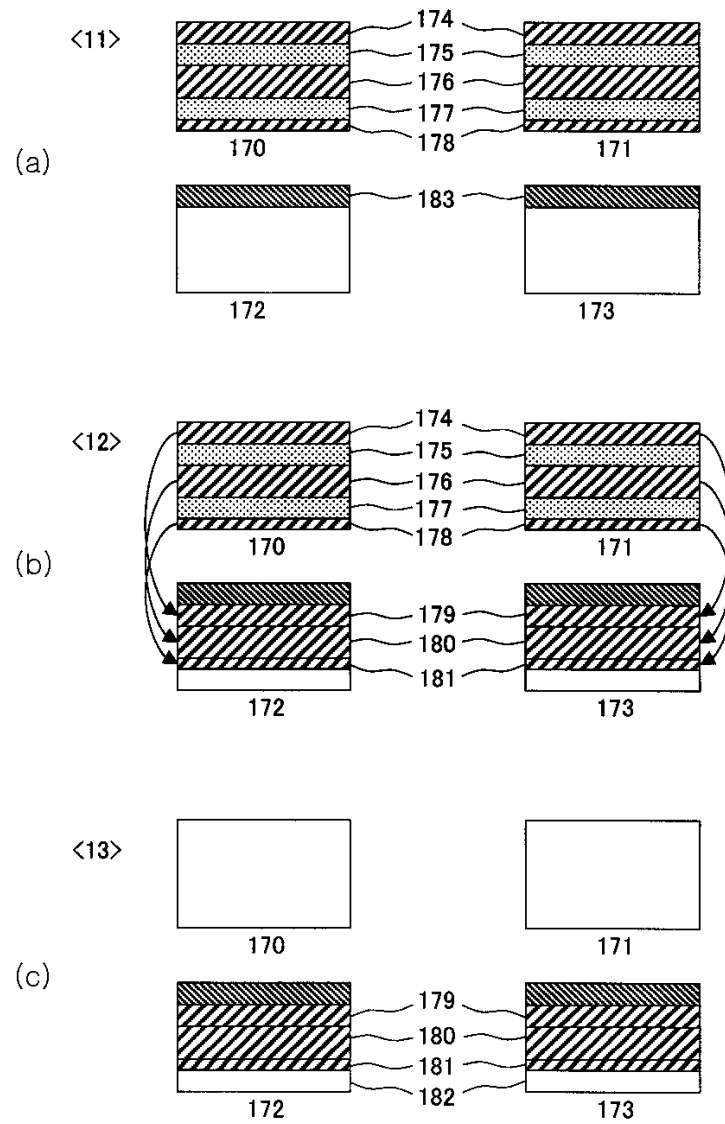
도면24



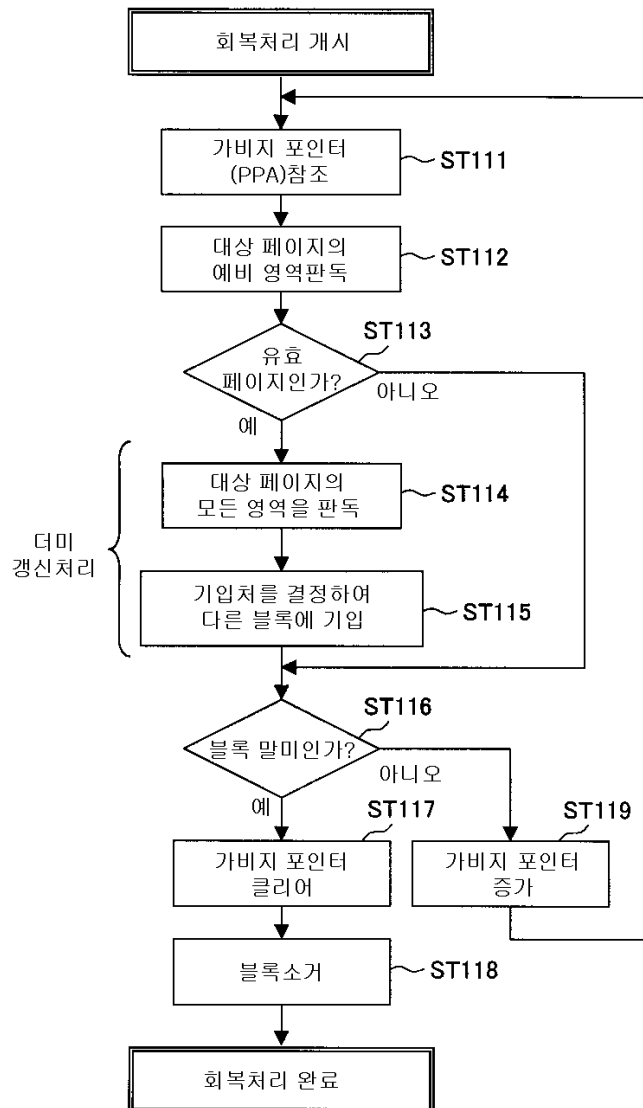
도면25



도면26



도면27



도면28

