(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.[7]
G06K 19/073
G06K 19/07
G11C 7/00

(11) 공개번호 10-2005-0103448
(43) 공개일자 2005년10월31일

(21) 출원번호 10-2004-7020514
(22) 출원일자 2004년12월17일
번역문 제출일자 2004년12월17일
(86) 국제출원번호 PCT/JP2004/001144
국제출원일자 2004년02월04일
(87) 국제공개번호 WO 2004/070728
국제공개일자 2004년08월19일

(30) 우선권주장 JP-P-2003-00027683 2003년02월04일 일본(JP)

(71) 출원인 마츠시타 덴끼 산교 가부시키가이샤
일본 오오사카후 가도마시 오오아자 가도마 1006

(72) 발명자 나카베 후토시
일본국 히로시마켄 히로시마시 아사미나미쿠 니시하라 8-22-14
가와노 신지
일본국 히로시마켄 히로시마시 사에키쿠 미노리 1-18-9-603

(74) 대리인 김영철

심사청구 : 없음

(54) 반도체 메모리 카드 및 컴퓨터 판독가능한 프로그램

요약

본 발명은 삭제된 파일의 콘텐츠를 더 이상 불법적으로 취득할 수 없게 하는 반도체 메모리 카드를 제공한다. 여기에서, 삭제 이벤트가 생성되면, 삭제 핸들러는 널-소거동작에 의해 삭제되는 파일에 대한 파일 엔티티와 FAT의 널-소거동작과 상기 파일의 다른 구성요소에 우선순위를 부여한다. 상기 파일의 파일 엔티티에 대한 널-소거동작은 삭제 핸들러를 포함하는 복수의 이벤트 핸들러에 분배된다.

대표도

도 13

명세서

기술분야

본 발명은 데이터를 파일로 저장할 수 있는 반도체 메모리 카드에 관한 것으로, 특히 파일 삭제 기술에 관한 것이다.

배경기술

IC 카드 및 SD 카드 등의 반도체 메모리 카드는 방송업 및 출판업를 포함하는 매스 미디어(mass media)로부터 금융기관과 정부 및 지방기관에 이르기까지 여러 산업분야에 실험적으로 도입되어 왔다. 반도체 메모리 카드는 그 특유의 편리함 때문에 이들 산업분야에서 큰 반향을 불러 일으켰다. 반도체 메모리 카드에서, 데이터는 물리적 계층, 파일 시스템 계층, 및 어플리케이션 계층을 갖는 계층모델(layer model)로 관리된다. 그러므로, 사용자는 반도체 메모리 카드에 파일을 생성할 수 있으며, 컴퓨터에 저장된 파일을 삭제하는 것과 동일한 절차로 생성된 파일을 삭제할 수 있다.

종래의 반도체 메모리 카드에서의 파일의 삭제는 관리정보를 오버라이트(overwrite)하는 것을 포함하며, 이 관리정보는 파일 프래그먼트(file fragment)와 파일 엔트리(file entry) 간의 링크관계를 나타내는 파일 할당표(File Allocation Table: FAT)를 포함한다. 파일의 삭제가 관리정보를 오버라이트하도록 한 이유는 다음과 같다. 파일에 대한 관리정보가 오버라이트되면, 반도체 메모리 카드에서 파일 엔티티(fiel entity)를 구성하는 프래그먼트의 위치가 손실되고, 반도체 메모리 카드에 불연속적으로 위치되는 프래그먼트들 사이의 링크관계가 손상되어 파일의 판독이 불가능하게 되기 때문이다.

종래의 반도체 메모리 카드에서의 파일 시스템에 의한 데이터 관리는 일본 특허공개공보 제2001-188701호, 미국특허 제5,479,638호 및 일본특허 제3389186호에 기재되어 있다.

파일 삭제에 관한 상술한 방법에 의하면 FAT와 파일 엔트리가 오버라이트된다. 그러나, 삭제된 파일의 파일 엔티티와 그 파일 엔티티를 구성하는 프래그먼트는 반도체 메모리 카드 상에 여전히 불연속적으로 남아있다. 이러한 삭제된 파일이 금전거래에 관한 데이터를 포함하는 경우, 은행계좌번호 및 ATM 카드번호가 불연속의 프래그먼트들로부터 판독될 위험이 있다.

또한, 제 3 자가 반도체 메모리 카드 상의 불연속 프래그먼트들을 추적하여 삭제된 파일을 재생함으로써 삭제되었어야 할 파일의 콘텐츠를 취득할 수 있다.

그러나, 파일삭제에 FAT와 파일 엔트리 등의 관리정보를 오버라이트하는 것이 포함되어야 할 이유가 있다. 그것은 오버라이트에 필요한 처리시간 때문이다. 실험적 계산에 의하면, 512바이트를 오버라이트하는데 2밀리초, 1메가바이트를 오버라이트하는데 4초, 10메가바이트를 오버라이트하는데 40초가 걸렸다. 이것은 완전한 삭제동작을 수행하는데 걸리는 시간, 즉 파일을 완전히 판독할 수 없게 하기 위해 파일의 파일 엔티티를 오버라이트하는데 걸리는 시간을 나타낸다. 또한, 이러한 완전한 삭제동작은 삭제된 파일을 구성하는 프래그먼트의 위치를 나타내는 데이터가 어디에 저장되어야 하는가에 대한 문제를 내포한다. 프래그먼트 뿐아니라 프래그먼트의 위치를 나타내는 데이터가 비휘발성 메모리에 저장되면, 제 3 자는 아마도 완전한 삭제동작이 달성되기 전에 프래그먼트의 위치를 판독할 수 있을 것이다. 따라서, 삭제되는 파일에 관한 정보는 완전한 삭제동작이 수행되는 동안 충분히 보호되지 못한다.

발명의 상세한 설명

본 발명의 목적은, 파일삭제의 명령이 있는 때로부터 그 파일이 완전히 삭제될 때까지, 삭제되는 파일의 파일 엔티티를 보호할 수 있는 반도체 메모리 카드를 제공하기 위한 것이다.

상기 목적은 엔티티 및 관리정보로 구성되는 파일을 저장하는 비휘발성 메모리; 및 처리부와 내부 메모리를 포함하는 부정변경 방지 모듈을 포함하는 반도체 모리 장치에 의해 달성된다. 처리부는 파일에 대한 삭제 이벤트가 생성되면, (i) 부정변경 방지 모듈 내의 내부 메모리에 로케이션 테이블을 생성하고, (ii)관리정보를 오버라이트하는 삭제부를 포함한다. 여기에서, 로케이션 테이블은 엔티티의 위치를 나타내고, 삭제부가 엔티티를 오버라이트할 때 삭제부에 의해 참조된다.

파일의 파일 엔티티는 부정변경 방지 모듈 내의 메모리에 저장된 로케이션 테이블을 참조하여 오버라이트된다. 그러므로, 삭제 이벤트가 생성될 때부터 파일이 완전히 삭제될 때까지 프래그먼트에 관한 정보는 충분히 보호될 수 있다.

파일 엔티티는 삭제 이벤트가 생성된 후의 유휴시간에 로케이션 테이블을 참조하여 오버라이트된다. 이러한 방식으로, 파일 엔티티 오버라이팅의 처리부하는 복수의 유휴시간 단위로 수행되도록 분할된다. 이러한 분할 때문에 반도체 메모리 카드는 곧바로 대기 상태로 복귀할 수 있어, 사용자가 파일 삭제에 오랜 시간이 걸린다는 것을 느끼지 않고 파일 엔티티의 오버라이팅을 완료할 수 있다.

또한, 파일을 구성하는 프래그먼트들 사이의 링크 관계를 파괴하는 파일 관리정보의 오버라이팅은 파일 엔티티의 오버라이팅에 대해 우선순위를 부여받는다. 오버라이팅 동작은 상술한 순서로 수행되기 때문에, 시간이 경과함에 따라 프래그먼트의 역분석은 더욱 어려워진다.

여기에서, 처리부는, (i)반도체 메모리 카드가 접속된 장치가 내린 명령에 대응하는 동작을 수행하며, (ii) 이 장치가 내린 명령을 분석하고, 그 분석결과에 대응하는 이벤트를 생성하는 분석부를 추가로 포함한다. 또한, 삭제부는, 삭제 이벤트가 생성되면 로케이션 테이블을 생성하고, 관리정보를 오버라이트하는 주삭제부; 및 삭제 이벤트와 다른 이벤트가 생성되면, 생성된 로케이션 테이블을 참조하여 엔티티를 오버라이트하는 부삭제부를 포함한다.

여기에서, 반도체 메모리 카드는 삭제 이벤트가 생성되는 시간을 측정하기 시작하는 타이머를 추가로 포함한다. 주삭제부는 타이머가 타임아웃을 알릴 때까지 관리정보에 부가하여 엔티티의 일부분을 오버라이트하고, 부삭제부는 부삭제부에 의해 오버라이트되지 않는 엔티티의 나머지 부분을 오버라이트한다. 따라서, 파일삭제의 지시에 응답하는 오버라이팅 동작은 타임아웃될 때까지 계속된다. 따라서, 예를 들어, 비휘발성 메모리에 대한 오버라이팅 동작에 필요한 기간에 기초하여 타임아웃이 조정되면, 사용자를 위한 대기시간이 최적화될 수 있다.

여기에서, 내부 메모리는 엔티티가 암호화되는 암호화키를 저장하고, 삭제 이벤트가 생성되면, 주삭제부는 파일의 관리정보를 오버라이트하기 전에 암호화키를 오버라이트한다. 암호화키의 오버라이팅은 파일 엔티티의 오버라이팅에 대해 우선순위를 부여받는다. 따라서, 파일의 해독이 불가능하게 된다. 상술한 순서로 오버라이팅 동작이 수행되기 때문에, 시간이 경과함에 따라 프래그먼트의 역분석이 더욱 어려워지게 된다.

여기에서, 반도체 메모리 카드는 임의의 장치로부터의 접촉식 또는 무접촉식의 전력공급수단에 의해 동작하지만, 처리장치는 반도체 메모리 카드가 장치로부터의 접촉식 전력공급수단에 의해 동작할 때에만 오버라이팅을 수행한다. 결국, 무접촉식 전력공급 중에 파일 엔티티의 오버라이팅은 수행되지 않는다. 이것으로 안정한 동작이 실현된다.

도면의 간단한 설명

도 1은 SDeX 메모리 카드(400)가 사용되는 환경을 도시한 도면

도 2는 본 발명에 따르는 반도체 메모리 카드의 내부 구성도

도 3은 (도 2에 도시된) TRM(1)의 하드웨어 구성도

도 4는 TRM(1)의 마스크 ROM(6)과 CPU(7)(모두 도 3에 도시됨)로 구성되는 부분의 소프트웨어 구성도

도 5는 외부 플래시 메모리(2)와 내부 EEPROM(3)(모두 도 3에 도시됨)의 논리적 포맷을 도시한 도면

도 6은 EC 클라이언트 어플리케이션에 대한 확장영역(22), 인증영역(23), 및 비인증영역(24)(도 5에 도시됨)의 내부 구성도

도 7은 파티션의 구성도

도 8의 (A)는 2중 FAT의 구성도, 도 8의 (B)는 루트 디렉토리 엔트리의 구성도, 도 8의 (C)는 사용자 영역의 구성도

도 9는 파일명 "EOB001.SE1"이 클러스터의 크기에 따라 5개의 프래그먼트로 분할되고, 그 프래그먼트들이 클러스터 003, 004, 005, 00A, 및 00C에 각각 저장되어 있는 상태를 도시한 도면

도 10은 분할되어 복수의 클러스터에 저장되는 파일명 "EOB001.SE1"에 대한 루트 디렉토리 엔트리와 FAT의 예를 도시한 도면

도 11은 API(10)(도 4에 도시됨)의 구성도

도 12의 (A)는 파일에 대한 기입동작이 판독/기입 핸들러(14)에 의해 어떻게 수행되는지를 보여주는 도면, 도 12의 (B)는 파일에 대한 판독동작이 판독/기입 이벤트 핸들러(14)에 의해 어떻게 수행되는지를 보여주는 도면

도 13은 삭제 핸들러(16)의 특성에 집중하여 API(10)의 내부 구성을 도시한 도면

도 14는 소거동작 관리 테이블의 예를 도시한 도면

도 15는 이벤트 분석 핸들러(11)(도 11에 도시됨)의 프로시져(peocedure)를 도시한 흐름도

도 16은 비명령 실행 핸들러(12)의 프로시져를 도시한 흐름도

도 17은 판독/기입 핸들러(14)의 프로시져를 도시한 흐름도

도 18은 삭제 핸들러(16)(도 11에 도시됨)의 프로시져를 도시한 흐름도

도 19는 널-소거 서브루틴(null-clear subroutine)의 프로시져를 도시한 흐름도

도 20은 삭제 이벤트가 도 10에 도시된 바와 같이 저장된 파일로 생성될 때 삭제 핸들러(16)의 동작을 도시한 도면

도 21은 비명령 실행 핸들러(12)와 삭제 핸들러(16)에 의한 오버라이팅 동작을 도시한 도면

도 22는 제 2 실시예에 관한 소거동작 관리 테이블의 예를 도시한 도면

도 23은 제 2 실시예에 관한 널-소거 서브루틴의 프로시져를 도시한 도면

도 24는 역방향 널-소거동작의 프로시져를 도시한 흐름도

도 25는 제 4 실시예에 관한 메모리 모듈의 구성도

도 26은 플래시 메모리의 성능과 FeRAM의 성능을 비교한 도면

도 27은 파일 엔트리, FAT, 및 소거동작 관리 테이블과 같은 주기적으로 갱신되는 데이터를 저장하는 FeRAM을 도시한 도면

도 28은 제 5 실시예에 관한 TRM(1)의 메모리(3)의 구성도

실시예

이하, 본 발명의 반도체 메모리 카드의 실시예에 대해 설명하기로 한다. 제 1 실시예에 관한 반도체 메모리 카드는 IC 카드의 내장 부정변경 방지 모듈을 갖는 SDeX 메모리 카드이다. SDeX 메모리 카드는 SD 메모리 카드와 유사하게 SD 휴대형 장치를 위한 기록매체로서 이용되며, IC 카드의 내장 부정변경 방지 모듈을 갖는다. 또한, 제 1 실시예에 관한 SDeX 메모리 카드(400)(도 1)는 접촉 및 무접촉 통신방식으로 외부장치와 통신할 수 있다.

우선, 본 발명에 따르는 반도체 메모리 카드(SDeX 메모리 카드(400))가 어떻게 사용되는지에 대해 설명하기로 한다. SDeX 메모리 카드(400)는 이동전화 등의 SD 휴대형 장치에 접속되며, 도 1에 도시된 환경에서 사용된다. 도 1은 SDeX 메모리 카드(400)가 사용되는 환경을 도시한다.

도 1에 도시된 환경은 전자 상거래(Electronic Commerce: EC) 서버(100), 카드 리더/라이터(200), 무선 기지국(210), 및 SD 휴대형 장치(300)를 포함한다.

EC 서버(100)는 카드 리더/라이터(reader/writer)(200), 무선 기지국(210) 및 네트워크를 통해 SDeX 메모리 카드(400)에 전자 상거래 서비스를 제공한다. EC 서버(100)에서는 다수의 EC 어플리케이션 프로그램이 동작하며, 각 어플리케이션 프로그램은 고유의 전자 상거래 서비스를 SDeX 메모리 카드(400)에 제공한다. EC 서버(100)에서 동작하는 전자 상거래 어플리케이션은 서버 어플리케이션이며, 그들 각각은 상이한 전자 상거래 서비스를 제공한다. 도 1에서, n종류의 전자 상거래 서비스에 대한 EC 서버 어플리케이션은 간략하게 S_APL1, 2, 3, ... n으로 각각 표현된다. 여기에서는, n종류의 서버 어플리케이션이 있다. EC 서버(100)는 네트워크, 카드 리더/라이터(200) 및 무선 기지국(210)을 통해 SDeX 메모리 카드(400)에 EC 명령을 내림으로써 전자 상거래 서비스를 제공한다.

카드 리더/라이터(200)는, 예를 들면, 신용카드 회사와 금융기관의 현금 자동지급기 또는 상점의 금전 등록기에 포함되는 장치이다. 카드 리더/라이터(200)는 SDeX 메모리 카드(400)에 전력을 공급하며 SDeX 메모리 카드(400)와 무접촉식으로 입출력을 수행한다. 카드 리더/라이터(200)는 네트워크에 접속된다. SDeX 메모리 카드(400)는 카드 리더/라이터(200)를 통해 EC 서버(100)가 공급하는 전자 상거래 서비스를 수신한다.

무선 기지국(210)은 빌딩 꼭대기와 전신주에 배치된다. 무선 기지국(210)은 이동 전화식 SD 휴대형 장치(300)와 무선으로 데이터의 입출력을 실행한다. 무선 기지국(210)은 네트워크에 접속된다. 또한, SD 휴대형 장치(300)는 무선 기지국(210)을 통해 EC 서버(100)가 공급하는 전자 상거래 서비스를 수신한다.

SD 휴대형 장치(300)는 SDeX 메모리 카드(400)를 SD 휴대형 장치(300)에 접속시키는 방식에 의해 SDeX 메모리 카드(400)에 액세스할 수 있다. SD 휴대형 장치(300)에는 브라우저 소프트웨어가 설치되므로 사용자는 브라우저의 사용자 인터페이스를 통해 SDeX 메모리의 파일 시스템(file system)(이하, FS라 함)에 액세스할 수 있다. 이러한 파일 시스템의 액세스는, SD 휴대형 장치(300)가 SD 메모리 카드에 의해 규정된 SD 명령을 SDeX 메모리 카드(400)에 내리고, SDeX 메모리 카드(400)로부터의 명령에 대한 응답을 수신하는 방식으로 수행된다. SD 휴대형 장치(300)가 SDeX 메모리 카드(400)를 부트스트랩(bootstrap)하면, SDeX 메모리 카드(400)는 SD 휴대형 장치(300)에 통합되어 IC 카드로서의 기능을 한다. 여기에서, SD 휴대형 장치(300)의 배면에는 헬리컬 안테나가 내장된다. SD 휴대형 장치(300)가 IC 카드로서의 기능을 하면, 헬리컬 안테나는 카드 리더/라이터(200)로부터 SDeX 메모리 카드(400)로 전력을 공급한다. 또한 SD 휴대형 장치(300)는 SDeX 메모리 카드(400)와 명령/응답을 주고 받으며, EC 서버(100)와 명령/응답을 주고 받는다. 구체적으로, SD 휴대형 장치(300)는 EC 서버(100)로부터의 전자 상거래 명령을 캡슐화하여(encapsulate) 확장된 SD 명령을 생성하고, 그 확장된 SD 명령을 SDeX 메모리 카드(400)에 출력한다. 또한, SD 휴대형 장치(300)는 SDeX 메모리 카드(400)로부터의 SD 응답으로부터 EC 응답을 취득하여, 그 EC 응답을 SDeX 메모리 카드(400)에 출력한다. SD 휴대형 장치(300)가 SDeX 메모리 카드(400)를 부트스트랩하면 SDeX 메모리 카드(400)는 IC 카드로서의 기능을 하며, "EC 모드"로 된다. SD 휴대형 장치(300)가 SDeX 메모리 카드(400)를 저장매체로서 사용하면, SDeX 메모리 카드(400)는 "SD 모드"로 된다.

SDeX 메모리 카드(400)가 SD 모드이면, 그것은 SD 메모리 카드로서 사용된다. SD 모드에서는, SD 휴대형 장치(300)가 SDeX 메모리 카드(400)의 호스트 장치이다. 이 경우, SDeX 메모리 카드(400)는 분배서버로부터 SD 휴대형 장치로 다운로드되는 오디오 데이터 및 비디오 데이터를 저장하기 위해 사용된다. 따라서, 호스트 장치는 SDeX 메모리 카드(400)에 저장된 오디오 데이터 및 비디오 데이터를 재생할 수 있다.

EC 모드에서는, SDeX 메모리 카드(400)가 IC 카드로서 사용된다. 또한, EC 모드에서는 SDeX 메모리 카드(400)가 SD 휴대형 장치(300)에 접속된다. 그러나 SDeX 메모리 카드(400)의 호스트 장치는 SD 휴대형 장치(300)가 아니라 네트워크상의 EC 서버(100)이다. SDeX 메모리 카드(400)는 카드 리더/라이터(200) 및 무선 기지국(210)과 함께 SDeX 메모리 카드(400)가 접속되는 SD 휴대형 장치(300)에 의해 EC 서버(100)와 통신한다. SDeX 메모리 카드(400)와 EC 서버(100) 사이에서는 이러한 방식으로 금융거래가 실행된다.

본 실시예에 관한 SDeX 메모리 카드(400)는 공급된 오디오 데이터와 비디오 데이터를 저장하는 기능외에도 IC카드로서의 기능을 갖기 때문에, 사용자에게 더욱 편리함을 제공한다.

여기에서, SDeX 메모리 카드(400)는, 도 1에 의하면 카드 리더/라이터(200)를 통해 EC 모드로 EC 서버(100)에 액세스한다. 이와 달리, SDeX 메모리 카드(400)는 SD 휴대형 장치(300)가 무선 기지국(210)과 네트워크를 통해 EC 서버(100)에 액세스하는 방식으로 EC 서버(100)에 액세스할 수 있다.

이하, 본 발명에 의한 반도체 메모리 카드를 제조하는 방법에 대해 설명하기로 한다. 본 발명에 따르는 반도체 메모리 카드는 도 2 및 도 3에 도시된 바와 같은 내부 구성을 가지며, 산업적으로 제조될 수 있다.

도 2에 도시된 바와 같이, 커넥터, 부정변경 방지 모듈 칩(Tamper-Resistant Module chip: TRM)(1), 및 256메가 바이트의 용량을 갖는 플래시 메모리 칩(2)이 본 발명에 따르는 반도체 메모리 카드에 패키징된다.

부정변경 방지(tamper-resistance)에 대해서는 다양한 정의가 있다. 그러나, 부정변경 방지의 일반적인 정의는 다음과 같다.

(1) TRM 칩이 물리적으로는 공개되어 있더라도 그 내부구조는 알려질 수 없다.

(2) 전자기파가 TRM 칩에 조사(照射)되더라도 그 내부구조는 알려질 수 없다.

(3) TRM에 입력되는 데이터의 길이와 그 데이터를 처리하는 시간 사이에는 비선형적 관계가 있다.

(4) 출력 데이터는 입력 데이터의 에러에 기초한 역조작(reverse operation)을 통해서는 취득되지 않는다.

이들 4가지 특성 때문에 TRM(1)은 다양한 종류의 역조작을 방어한다. 이하, TRM(1)의 하드웨어 구성에 대해 설명하기로 한다.

도 3은 TRM(1)의 하드웨어 구성을 도시한 도면이다. 도 3에 도시된 바와 같이, 내부 EEPROM(3), 외부 메모리 제어부(4), 호스트 인터페이스 모듈(HIM)(5), 마스크 ROM(6), 및 CPU(7)가 마이크로 컴퓨터 시스템을 구성하도록 TER(1)에 패키징된다.

내부 EEPROM(3)은 판독 및 기입가능한 내부 메모리이다. TRM(1)으로서 패키징된 마이크로 컴퓨터 시스템은 단위 면적당 제조비용이 높다. TRM(1)의 내부 EEPROM(3)은 32킬로 바이트의 용량을 갖는다. 이하에서는 도 2에 도시된 플래시 메모리(2)를 내부 EEPROM(3)과 구별하기 위해 때때로 외부 메모리라 언급하기도 한다.

외부 메모리 제어부(4)는 외부 플래시 메모리(2)에 액세스하기 위해 독점적으로 사용되는 회로이다. 외부 플래시 메모리(2)에 대한 액세스는 SD 휴대형 장치(300)가 내린 SD 명령에 기초하여 실행된다.

HIM(5)은 SD 휴대형 장치(300)가 내린 SD 명령을 그들의 명령의 수를 참조하여 분류한다. SD 명령의 수는 1부터 m까지의 수 또는 (m+ 1) 이상까지 확장된 수 이다. SD 명령의 수가 1과 m의 범위 안에 있으면, HIM(5)은 SD 명령을 외부 메모리 제어부(4)에 출력한다. SD 명령의 SD 명령수가 (m+ 1) 이상이면, HIM(5)은 EC 명령을 취득하여 CPU(7)에 출력한다. 이 EC 명령은 확장된 SD 명령으로 캡슐화되어 있다.

마스크 ROM(6)은 자바 가상머신(JAVA Virtual Machine)과 어플리케이션 프로그램을 미리 저장한다. SDeX 메모리 카드(400)가 SD 휴대형 장치(300)를 부트스트랩하면, 부트스트랩은 마스크 ROM(6)의 미리 정해진 어드레스로부터 시작한다. 따라서 SD 휴대형 장치(300)는 활성화되어 EC 모드로 된다.

CPU(7)는 마스크 ROM(6)에 저장된 프로그램을 실행한다.

도 4는 TRM(1)(도 3에 도시)의 마스크 ROM(6)과 CPU(7)로 구성되는 부분의 소프트웨어 구성을 도시한 도면이다. 점선 wk1로 둘러싸인 부분은 IC 카드와 등가의 모듈이다. TRM(1)의 나머지 부분은 SD 메모리 카드와 등가의 모듈이다.

SD 메모리 카드와 등가의 TRM(1)의 부분은 외부 메모리 제어부(4)와 HIM(5)을 포함한다. HIM(5)은 SD 메모리 카드로서의 기능을 가질 뿐 아니라 SD 메모리 카드 등가모듈과 IC 카드 등가모듈 사이의 제 1 접촉부로서의 기능도 갖는다.

IC 카드 호환모듈은 계층구조를 갖는다. 이 계층 구조에서, 내부 EEPROM(3)은 최하위 계층(물리적 계층)이다. 어플리케이션 인터페이스(API)(10)는 내부 EEPROM(3)이 존재하는 계층 바로 위의 계층이다. 자바 가상머신(9)은 API(10)가 존재하는 계층 바로 상부의 계층이다. EC 클라이언트 어플리케이션(8)은 최상위 계층이다. SD 메모리 카드 호환부분의 외부 메모리 제어부(4)는 내부 EEPROM(3)과 유사한 물리적 계층에 위치된다.

이하, 도 4에 도시된 소프트웨어의 구조(EC 클라이언트 어플리케이션(8), 자바 가상머신(9), 및 API(10))에 대해 설명하기로 한다.

EC 클라이언트 어플리케이션(8)은 자바(JAVA)로 쓰여진 EC 어플리케이션의 일종이며, 사용자의 지시에 기초하여 EC 서버(100)에 액세스한다. 상이한 EC 서비스에 각각 대응하는 여러 종류의 EC 서버 어플리케이션이 EC 서버(100)에서 동작하기 때문에, 상이한 EC 서비스에 각각 대응하는 여러 종류의 EC 클라이언트 어플리케이션이 SDeX 메모리 카드(400)에서 동작한다. 도 4에서, EC 클라이언트 어플리케이션 C_APL1, 2, 3, ...n은 EC 서버(100) 상의 EC 어플리케이션(S_APL 1, 2, 3, ...n)에 각각 대응하는 것으로 도시되어 있다. EC 클라이언트 어플리케이션(8)은 카드 리더/라이터(200), 무선 기지국(210), 및 네트워크를 통해 EC 서버(100) 상의 EC 서버 어플리케이션과 명령을 주고받음으로써 각종 EC 서비스를 취득하게 된다. EC 서버 어플리케이션으로부터 수신된 EC 명령이 데이터 기입 명령이면, EC 클라이언트 어플리케이션(8)은 EC 명령을 자바 가상머신(9)을 통해 API(10)에 출력한다.

또한, EC 클라이언트 어플리케이션(8)은 사용자의 지시에 기초하여 EC 모드로 외부 플래시 메모리(2)와 내부 EEPROM(3)에 액세스한다. 이 액세스는 파일을 생성하고, 그 생성된 파일에 대하여 판독동작 및 기입동작을 수행하는 등의 파일 액세스를 포함한다.

자바 가상머신(9)(도 4의 Java Card VM(등록된 상표))은 EC 클라이언트 어플리케이션(8)이 기입된 자바언어를 CPU(7)의 원시 코드로 변환하며, CPU(7)로 하여금 EC 클라이언트 어플리케이션(8)을 실행하도록 한다.

API(10)는 EC 클라이언트 어플리케이션(8)의 명령에 기초하여 외부 플래시 메모리(2)와 내부 EEPROM(3)에 판독/기입을 수행한다. 이상, SDeX 메모리 카드(400)의 소프트웨어 구조에 대해 설명하였다.

이하, 외부 플래시 메모리(2)와 내부 EEPROM(3)의 논리적 포맷에 대해 설명하기로 한다. 도 5는 외부 플래시 메모리(2)와 내부 EEPROM(3)의 논리적 포맷을 도시한 도면이다. 외부 플래시 메모리(2)와 내부 EEPROM(3)은 2개의 메모리 공간, 즉 sm1과 sm2를 갖는다. 메모리 공간 sm1은 TRM(10) 내의 CPU(7)에 의해 액세스가능하고, EC 클라이언트 어플리케이션을 위한 영역(21)과 EC 클라이언트 어플리케이션을 위한 확장된 영역(22)으로 구성된다. 메모리 공간 sm2는 TRM(1) 내의 CPU(7)를 통하지 않고도 SD 휴대형 장치(300)로부터 액세스 가능하다. 메모리 공간 sm2는 인증영역(23)과 비인증 영역(24)으로 구성된다. 인증영역(23)과 비인증영역(24)은 SD 메모리 카드의 메모리 영역이며, 일본특허 제3389186호에 상세히 설명되어 있다.

도 6은 ISO/IEC 9293에 따르는 파일 시스템 구조를 갖는, EC 클라이언트 어플리케이션을 위한 확장영역(22), 인증영역(23), 비인증 영역(24)의 구성을 도시한 도면이다. 그러나, ISO/IEC 9293 파일 시스템 구조는 하나의 예일 뿐이며 단지 편리하기 때문에 선택된 것이다. EC 클라이언트 어플리케이션을 위한 확장영역(22), 인증영역(23) 및 비인증 영역(24)은 유니버셜 디스크 포맷(Universal Disk Format: UDF) 등의 다른 파일 시스템을 가질 수도 있다. 일반적으로 말해, 프래그먼트의 길이가 변경가능하고, 시작 어드레스와 데이터 길이가 엔트리 정보에 나타나 있는 파일 시스템 구조를 이용할 수 있다.

EC 클라이언트 어플리케이션을 위한 확장영역(22)은 내부 EEPROM(3) 상의 영역(22a)과 외부 플래시 메모리(2) 상의 보안 플래시 영역(22b)으로 구성된다. 보안 플래시 영역(22b)은 파티션 1, 2, 3, ... n, 즉 파일 시스템 영역을 갖는다. 한편, 내부 EEPROM(3)내의 영역(22a)은 마스터 부트 레코더 및 파티션에 대한 참조 테이블(파티션 테이블 1, 2, 3, ...n)을 포함한다.

EC 클라이언트 어플리케이션을 위한 확장영역(22), 인증영역(23), 및 비인증 영역(24) 내의 파티션은 동일한 내부 구조를 갖는다. 도 7은 이러한 파티션의 구조를 도시한다.

파티션은 파티션 부트 섹터, 2중 파일 할당표(FAT), 루트 디렉토리 엔트리, 및 사용자 영역을 포함한다.

파티션 부트 섹터는 파티션에 관한 정보를 보여주는 테이블이다.

2중 FAT는 ISO/IEC 9293에 따르는 2개의 FAT를 포함한다. 각 FAT는 클러스터와 1대1로 대응하는 복수의 FAT 엔트리를 포함한다. 각 FAT 엔트리는 대응하는 클러스터가 사용되는지의 여부를 나타낸다. 만약 대응하는 클러스터가 사용되지 않으면 FAT 엔트리의 값은 "0"으로 설정된다. 대응하는 클러스터가 사용되면 FAT 엔트리의 값은 클러스터들 사이의 링크 관계, 즉 대응 클러스터 다음에 판독되는 다음 클러스터를 나타내는 클러스터 개수의 값으로 설정된다. 도 8의 (A)의 점선 ff1은 FAT에 포함된 복수의 FAT 엔트리(002, 003, 004, 005, ...)를 나타낸다. 각 FAT 엔트리(002, 003, 004, 005, ...)에 주어진 번호는 대응하는 클러스터, 즉 FAT 엔트리에 대응하는 클러스터의 클러스터 번호를 나타낸다.

루트 디렉토리 엔트리는 루트 디렉토리 내의 복수의 파일 엔트리를 포함하며, 각각의 파일 엔트리는 하나의 파일에 대응한다. 각각의 파일 엔트리는 파일의이름을 나타내는 "파일명", 파일의 파일 확장자를 나타내는 "파일 확장자", 파일의 시작을 저장하는 클러스터를 나타내는 "제 1 클러스터 번호", 파일의 속성을 나타내는 "파일 속성", 파일이 저장되는 시간을 나타내는 "저장시간", 파일이 저장되는 일자를 나타내는 "저장일자", 파일의 데이터 길이를 나타내는 "파일길이"의 엔트리들을 포함한다.

사용자 영역에서는 파일이 저장되고, 가장 작은 단위는 클러스터이다. 도 8의 (C)의 점선 ff2는 사용자 영역 내의 복수의 클러스터(002, 003, 004, 005, ...)를 나타낸다. 도 8의 (C)에서, 번호 002, 003, 004, 005, 006, 007, 008, ...는 각 클러스터를 식별하기 위한 목적을 갖는 16진수의 3자리 클러스터 번호이다. 데이터 영역에 대한 액세스는 가장 작은 클러스터의 단위로 실행되기 때문에 데이터 영역 내의 위치는 클러스터 번호로 표시된다.

여기에서, 파일명 EOB001.SE1이 루트 디렉토리에 저장되는 방법, 즉 파일 저장방법의 예에 대해 도 9를 참조하여 설명하기로 한다. 여기에서, "EOB001.SE1"의 "EOB"는 EC 오브젝트(Object)를 단축한 형태이며, "SE"는 "Secure EC"를 따서 명명한 확장자이다. 상술한 바와 같이, 데이터 영역에서 액세스 가능한 가장 작은 단위는 클러스터이기 때문에, 파일 EOB001.SE1은 가장 작은 단위의 클러스터로 데이터 영역에 저장될 필요가 있다. 우선, 파일 EOB001.SE1은 각각이 클러스터의 크기를 갖는 프래그먼트들로 분할되고, 분할된 각각의 프래그먼트는 클러스터에 기입된다. 도 9는 파일 EOB001.SE1이 클러스터 크기에 따라 5개의 프래그먼트로 분할되고, 그 프래그먼트들이 각각 클러스터 003, 004, 005, 00A, 및 00C로 저장된 것을 도시하고 있다.

파일 EOB001.SE1이 상술한 바와 같은 프래그먼트이면, 디렉토리 엔트리와 FAT는 도 10에 도시된 바와 같이 설정될 필요가 있다.

도 10은 EOB001.SE1이 복수의 클러스터로 분리되어 저장되어 있을 때의 디렉토리 엔트리와 FAT를 예로서 도시한 도면이다. 도 10에 따르면, 파일 EOB001.SE1의 시작부분이 클러스터 003에 저장되기 때문에, 루트 디렉토리 엔트리 내의 "제 1 클러스터 번호" 엔트리는 003, 즉 시작부분을 저장한 클러스터의 클러스터 번호를 나타낸다. 도 10으로부터 파일 EOB001.SE1의 2개의 후속 프래그먼트가 클러스터 004와 005로 각각 저장되어 있음을 알 수 있다. 파일 EOB001.SE1의 시작부분을 저장한 클러스터 003은 FAT 엔트리 003 (004)에 대응한다. 여기에서, FAT 엔트리 003은 시작부분에 계속되는 것을 저장한 클러스터 004를 지시하는 004를 나타낸다. 또한, 시작부분에 계속되는 2개의 프래그먼트를 저장한 클러스터 004 및 005는 FAT 엔트리 004 (005) 및 005 (00A)에 대응한다. 이들 FAT 엔트리는 후속 프래그먼트를 저장한 클러스터 005 및 00A를 지시하는 005와 00A를 나타낸다.

FAT 엔트리 내의 클러스터 번호를 화살표 fk1, fk2, fk3, fk4, 및 fk5를 따라 추적하면, 파일 EOB001.SE1을 구성하는 모든 프래그먼트가 판독될 수 있다. 상술한 설명으로부터, SDeX 메모리 카드(400)의 사용자 영역에 대해 액세스 가능한 가장 작은 단위는 클러스터이며, 이들 클러스터는 FAT와 1대1로 대응한다는 것을 알 수 있다. EOB 파일의 마지막을 저장하는 클러스터(도 9의 클러스터 00C)에 대응하는 FAT 엔트리는 "FFF"를 나타내며, 이것은 파일의 최종 프래그먼트를 저장하는 대응 클러스터를 나타낸다.

이하, API(10)의 구조에 대해 설명하기로 한다. 여기에서, 이벤트는, 예를 들어, EC 명령, 하드웨어의 인터럽션 발생 및 통신 데이터의 입력 등, API(10)에 대한 입력을 나타내는 포괄적인 용어이다. API(10)는 API(10)의 내외부에서 발생하는 이벤트에 응답하여 시작하는 프로그램으로 구성된다. 이러한 프로그램은 "이벤트 핸들러"라 하며, 도 11에 도시되어 있다. 도 11에 도시된 바와 같이, API(10)는 이벤트 분석 핸들러(11), 비명령실행 핸들러(12), 타이머 핸들러(13), 판독/기입 핸들러(14), 암호화키 테이블(15), 및 삭제 핸들러(16) 등의 이벤트 핸들러를 포함한다.

이벤트 분석 핸들러(11)는 API(10)의 내외부에서 발생하는 이벤트를 분석하고, 그 분석결과에 따라 API-내부 이벤트를 생성한다. API(10)의 외부에서 발생하는 대부분의 공통 이벤트 중의 하나는 EC 클라이언트 어플리케이션(8)이 내린 EC 명령이다. 이벤트 분석 핸들러(11)는 EC 명령의 콘텐츠를 분석한다. EC 명령이 파일에 대한 판독 및 기입 동작을 나타내면, 이벤트 분석 핸들러(11)는 파일 판독/기입 이벤트 및 명령 시작 이벤트의 내부 이벤트를 생성한다. EC 명령이 파일 삭제를 나타내면, 이벤트 분석 핸들러(11)는 파일 삭제 이벤트와 명령 시작 이벤트의 내부 이벤트를 생성한다. 이들 내부 이벤트의 생성에 의해 파일에 대한 판독동작 및 기입동작을 위한 이벤트 핸들러 또는 파일 삭제를 위한 이벤트 핸들러가 각각 동작되게 된다.

API 내부 이벤트는 이벤트 핸들러에 의해 완료되는 프로시져(procedure)를 나타내는 이벤트(완료 이벤트)를 포함한다. EC 명령을 실행하는 이벤트 핸들러가 완료 이벤트를 생성하면, 이벤트 분석 핸들러(11)는 EC 명령을 내린 EC 클라이언트 어플리케이션(8)에 EC 응답을 출력한다.

비명령 실행 핸들러(12)는 API(10) 내의 다른 이벤트 핸들러의 어느 것도 명령을 실행하지 않으면 동작한다. 비명령 실행 핸들러(12)의 구동기간과 EC 명령을 실행하는 다른 이벤트 핸들러의 구동기간은 상호 배타적이다. 즉, 비명령 실행 핸들러(12)는 다른 이벤트 핸들러가 전혀 동작하지 않을 경우에 동작한다.

타이머 핸들러(13)는 명령 시작 이벤트가 생성되는 시간을 측정하기 시작하여, 측정된 시간이 미리 정해진 시간의 기간과 동일해지면 타임아웃을 알린다. 이벤트 분석 핸들러(11)는 타이머 핸들러(13)가 타임아웃을 알려주는지의 여부를 감시하므로, 이벤트 핸들러에 의한 긴 동작을 강제로 중단시킬 수 있다. 이러한 강제중단은 이벤트 분석 핸들러(100)에 의한 중단 이벤트에 의해 달성된다.

판독/기입 핸들러(14)는 판독/기입 이벤트가 생성될 때 외부 플래시 메모리(2)와 내부 EEPROM(3) 상의 파일에 대하여 판독/기입을 한다. 도 12의 (A)는 파일에 대한 기입동작이 판독/기입 핸들러(14)에 의해 어떻게 수행되는지를 보여준다.

파일에 대한 기입동작은 다음의 방식으로 수행된다. 판독/기입 핸들러(14)는 데이터가 기입되는 파일명과 기입된 그 데이터를 EC 클라이언트 어플리케이션(8)(C_APL 1, 2, 3, ...n)으로부터 수신한다(도 12의 (A)의 ① 및 ②). 또한, 데이터가 기입되는 파일에 할당된 암호화키는 암호화키 테이블(15)로부터 판독/기입 핸들러(14)에 의해 취득된다(③). 그 다음에, EC 클라이언트 어플리케이션(8)(C_APL 1, 2, 3, ...n)으로부터 수신된 데이터는 암호화키 테이블(15)로부터 취득된 암호화키에 의해 암호화되어(④) 파일에 기입된다(⑤). 도 12의 (B)는 파일에 대한 판독동작이 판독/기입 핸들러(14)에 의해 어떻게 수행되는지를 보여준다.

파일에 대한 판독동작은 다음의 방식으로 수행된다. 판독/기입 핸들러(14)는 판독되는 파일명을 EC 클라이언트 어플리케이션(8)(C_APL1, 2, 3, ...n)으로부터 수신한다. 암호화된 데이터가 외부 플래시 메모리(2)로부터 판독된 후(도 12의 (B)의 ①), 판독되는 파일에 할당되는 암호화키는 암호화키 테이블(15)로부터 취득된다(②). 그리고, 판독된 데이터는 취득된 암호화키로 해독되어(③), EC 클라이언트 어플리케이션(8)(C_APL1, 2, 3, ...n)에 전달된다(④ 및 ⑤).

암호화키 테이블(15)은 파일용 암호화키와 파일명용 암호화키 간의 1대1 대응을 보여준다. 암호화키는 판독/기입 핸들러(14)에 의해 생성되며, EC 클라이언트 어플리케이션(8)이 파일을 생성할 때 암호화키 테이블(15)에 등록된다. 암호화키 테이블(15) 내의 등록된 암호화키는 파일이 열릴 때 참조되며, 판독동작 및 기입동작이 파일에 대하여 수행된다.

삭제 핸들러(16)는 삭제 이벤트가 생성될 때 동작한다. 삭제 이벤트가 생성되면, 파일을 구성하는 파일 엔트리, FAT, 및 파일 엔티티(file entity)는 소거동작 관리 테이블이 생성된 후, 널 코드(null code)와 함께 삭제 핸들러(16)에 의해 오버라이트된다. 본 실시예에서의 파일 삭제는 암호화키, 파일 엔트리, 및 파일용 FAT를 오버라이트하는 것과, 파일의 파일 엔티티를 오버라이트하는 것을 나타낸다. 여기에서, 오버라이팅은 (널-소거를 위해) 널 코드 또는 특정 방식으로 오버라이트하는 것을 나타낸다. 삭제 핸들러(16)는 전자의 오버라이팅에 우선순위를 부여한다. 후자의 오버라이트는 막대한 처리시간을 필요로 하기 때문에, 다른 이벤트 핸들러에 의해 수행될 수 있다. 그러나 본 발명에 의하면, 막대한 처리시간을 필요로 하는, 파일에 대한 파일 엔티티의 오버라이팅이, 복수의 유휴 기간(idle period)에서 신뢰성있게 분리되어 실행된다.

도 13은 상술한 본 발명의 특징에 중점을 둔 API(10)의 내부 구성도이다. 도 13의 화살표 cw1에 의해 도시된 바와 같이, 삭제 이벤트가 생성되면, 소거동작 관리 테이블이 삭제 핸들러(16)에 의해 생성되어, TRM(1) 내의 메모리(3)에 기입된다. 소거동작 관리 테이블은 오버라이팅이 어떻게 수행되는지를 보여준다. 이러한 소거동작 관리 테이블은, 널-소거되어야 하는 프래그먼트의 존재와 얼마나 많은 파일의 데이터가 널-소거되어야 하는지를 다른 이벤트 핸들러에 알려줄 수 있도록, 파일 엔트리와 FAT를 오버라이트하기 전에 생성된다. 소거동작 관리 테이블이 생성되면, 비명령 실행 핸들러(12), 판독/기입 핸들러(14) 및 삭제 핸들러(16)는 소거동작 관리 테이블을 참조하여 (화살표 nc1 및 nc2로 도시된 바와 같이) 프래그먼트의 X바이트를 오버라이트한다. 그 다음에, 소거동작 관리 테이블이 갱신된다. 화살표 nc1 및 nc2로 표시된 바와 같이, 외부 플래시 메모리(2)와 내부 EEPROM(3) 모두에 저장된 파일은 본 실시예에서는 널-소거될 수 있다. 그러나 여기에서는 설명이 복잡하게 되는 것을 피하기 위해 외부 플래시 메모리(2)에 저장된 파일에 대한 널-소거동작에 대해서만 설명한다. 본 실시예에서, 널-소거동작은 파일에 대한 파일 엔티티가 본 실시예에서 프래그먼트되었거나 프래그먼트되지 않은 경우에 수행될 수 있다. 그러나, 여기에서는 설명이 복잡하게 되는 것을 피하기 위해 파일에 대한 파일 엔티티가 프래그먼트될 때 수행되는 널-소거동작에 대해서만 설명하기로 한다. (이러한 이유로, 이하의 설명에서는 "파일 엔티티"를 일관하여 "프래그먼트"로 대체하기로 한다.)

도 14는 예로서 소거동작 관리 테이블을 도시한 도면이다. 소거동작 관리 테이블은 프래그먼트들과 1대1로 대응하는 복수의 레코드(record)로 구성된다. 각각의 레코드는 "유효 플래그", "시작 ADR", "소거된 ADR", 및 "종료 ADR"의 4개의 엔트리를 갖는다. "유효 플래그"는 대응 플래그먼트가 소거되었는지 또는 완전히 소거되었는지의 여부를 나타내고, "시작 ADR"은 프래그먼트의 시작 어드레스를 나타내며, "소거된 ADR"은 프래그먼트에 대한 소거동작이 불충분한 경우 얼마나 많은 프래그먼트의 데이터가 널-소거되었는지를 보여주는 소거완료 어드레스를 나타내고, "종료 ADR"는 프래그먼트의 종료 어드레스를 나타낸다. 삭제 핸들러(16)는 상술한 FAT 또는 파일 엔트리에 기초하여 시작 ADR 또는 종료 ADR의 값을 설정한다.

API(10) 내의 이벤트 분석 핸들러(11), 비명령 실행 핸들러(12), 판독/기입 핸들러(14), 및 삭제 핸들러(16)는 도 15 내지 도 19에 도시된 프로시져를 수행하는 프로그램을 컴퓨터 언어로 기입함으로써 생성된다. 도 15 내지 도 18은 이벤트 분석 핸들러(11), 비명령 실행 핸들러(12), 판독/기입 핸들러(14) 및 삭제 핸들러(16)의 프로시져를 각각 나타낸 흐름도이다. 도 19는 널-소거동작의 프로시져를 도시한 흐름도이다. 널-소거동작은 널 코드로 프래그먼트의 X바이트를 오버라이트하는 것을 나타낸다. 널-소거동작은 비명령 실행 핸들러(12), 판독/기입 핸들러(14) 및 삭제 핸들러(16)에 의해 실행되기 때문에 서브루틴인 것으로 간주된다.

도 15는 이벤트 분석 핸들러(11)의 프로시져를 도시한 흐름도이다.

도 15에서 단계 S101~S103은 이벤트를 스캐닝(scanning)하기 위한 루프 프로시져를 형성한다. 이 루프 프로시져는 외부 이벤트가 발생하는지의 여부를 검사하고(단계 S101), 판독/기입 핸들러(14) 및 삭제 핸들러(16)에 의한 동작이 완료되었는지의 여부를 검사하며(단계 S102), 타이머 핸들러(13)의 프로시져가 완료되었는지의 여부를 검사한다(단계 S103).

단계 S104~S109는 단계 S101에서 외부 이벤트의 발생이 검출되면 수행되는 단계이다.

단계 S104에서, 외부 이벤트가 EC 명령인지의 여부를 판정하기 위한 이벤트 분석이 행해지고, 판정결과, 외부 이벤트가 EC명령이면 EC 명령의 콘텐츠가 분석된다. 만약 외부 이벤트가 외부 플래시 메모리(2) 상의 파일에 대한 판독 및 기입 동작을 지시하는 EC 명령이면, 즉 판독/기입명령이면, 판독/기입 이벤트와 명령 시작 이벤트가 생성된다(단계 S106). 그리고, 이벤트 분석 핸들러(11)의 프로시져는 단계 S101~S103의 루프 프로시져로 복귀한다.

만약 외부 이벤트가 외부 플래시 메모리(2) 상의 파일을 삭제하라는 EC 명령이면, 즉 삭제명령이면, 삭제 이벤트와 명령 시작 이벤트가 생성된다(단계 S108). 그리고 이벤트 분석 핸들러(11)의 프로시져는 단계 S101~S103의 루프 프로시져로 복귀한다.

만약 상술한 예를 제외한 다른 외부 이벤트가 발생하면, 그 이벤트에 대응하는 동작이 수행된다(단계 S109). 그리고, 이벤트 분석 핸들러(11)의 프로시져는 단계 S101~103의 루프 프로시져로 복귀한다.

단계 S110 및 단계 S111는, 판독/기입 핸들러(14) 및 삭제 핸들러(16)에 의한 프로시져가 완료될 때 수행된다. 이들 단계에서는 명령을 내린 EC 클라이언트 어플리케이션(8)으로 EC 응답이 출력되고, API(10)에서 명령 종료 이벤트가 생성된다.

단계 S112는 타이머 핸들러(13)가 타임아웃을 알릴 때 수행된다. 이 단계에서, 중단 이벤트가 생성된다.

도 16은 비명령 실행 핸들러(12)의 프로시져를 도시한 흐름도이다. 단계 S1 및 S2는 루프 프로시져를 형성하고, 이 루프 프로시져에서 명령 시작 이벤트가 생성되었는지의 여부가 검사되고(단계 S1), 소거동작 관리 테이블이 존재하는지의 여부가 검사된다(단계 S2). 단계 S1과 S2의 매 사이클마다 단계 S2에서 소거동작 관리 테이블이 존재하는지의 여부가 판정된다. 판정결과, 소거동작 관리 테이블이 존재하면, 널-소거 서브루틴이 호출된다(단계 S4). 여기에서, 비명령 실행 핸들러(12)에 의한 프래그먼트의 오버라이팅은 다른 이벤트 핸들러가 전혀 동작하지 않는 유휴 기간에 수행된다. 그러므로, 비록 반도체 메모리 카드의 속성으로 인해 파일을 오버라이트하는데 시간이 걸리더라도 사용자는 그러한 시간 때문에 지겨워지지 않는다.

도 17은 판독/기입 핸들러(14)의 프로시져를 도시한 흐름도이다. 판독동작 및 기입동작이 수행되는 파일에 대응하는 암호화키는 단계 S20에서 암호화키 테이블(15)로부터 취득된다. 단계 S21에서는 기입동작이 수행되는지의 여부가 판정된다. 기입동작이 수행되면 EC 클라이언트 어플리케이션(8)으로부터의 데이터가 단계 S20에서 취득된 암호화키로 암호화되고(단계 S22), 암호화된 데이터는 보안 플래시 메모리(2)에 기입된다(단계 S23).

한편, 단계 S21에서의 판정이 판독동작을 수행하는 것을 나타내면, EC 클라이언트 어플리케이션(8)에 의해 요구되는 암호화된 데이터가 보안 플래시 메모리(2)로부터 판독된다(단계 S24). 보안 플래시 메모리(2)로부터 판독되는 암호화된 데이터는 단계 S20에서 취득된 암호화키를 이용하여 해독되며, EC 클라이언트 어플리케이션(8)으로 전달된다(단계 S25).

단계 S26에서는 소거동작 관리 테이블이 존재하는지의 여부가 판정된다. 만약 소거동작 관리 테이블이 존재하지 않으면 프로시져가 완료된다. 그러나, 소거동작 관리 테이블이 존재하면 단계 S27에서 널-소거동작이 수행된다.

도 18은 삭제 핸들러(16)의 프로시져를 도시한 흐름도이다. 우선, 단계 S11에서, 파일 프래그먼트와 1대1로 대응하여 레코드가 생성된다. 단계 S12에서, 각 레코드의 시작 ADR의 값과 종료 ADR의 값이, 대응 파일 프래그먼트의 시작 어드레스의 값과 종료 어드레스의 값으로 설정된다. 단계 S13에서, 각 레코드에서의 유효 플래그의 값이, 대응하는 프래그먼트가 소거중임을 나타내는 "1"로 설정된다. 단계 S14에서, 각 레코드에서의 소거된 ADR의 값이, 대응하는 파일 프래그먼트의 시작 어드레스의 값으로 설정된다.

단계 S15에서, 암호화키 테이블(15)의 복수의 암호화키 중에서 삭제되는 파일에 대응하는 암호화키가 삭제된다. 이 암호화키의 삭제는 파일의 해독을 불가능하게 만든다.

단계 S16에서 파일 엔트리가 널-소거된 후, 단계 S17에서 중단 이벤트가 생성되는지의 여부가 판정된다. 삭제 핸들러(16)의 프로시져는 중단 이벤트가 생성될 때 종료된다.

중단 이벤트가 생성되지 않으면, FAT는 단계 S18에서 널-소거된다. 이 단계에서 파일 엔트리와 FAT가 널-소거되기 때문에, 파일의 파일 엔트리를 구성하는 프래그먼트는 그들 사이의 링크관계를 손실하여 서로 연결관계가 단절되게 된다.

그 후, 단계 S19에서 중단 이벤트가 생성되는지의 여부가 판정된다. 중단 이벤트가 생성되면, 삭제 핸들러(16)의 프로시져가 종료된다. 중단 이벤트가 생성되지 않으면, 단계 S20에서 파일의 프래그먼트가 널-소거된다.

그 후, 단계 S19~S20의 프로시져가 반복된다. 이러한 방식으로, 타이머 핸들러(13)가 타임아웃을 알릴 때까지 삭제 핸들러(16)에 의한 프래그먼트의 오버라이팅이 계속된다.

도 19는 널-소거 서브루틴의 프로시져를 도시한 흐름도이다. 단계 S31에서, "1"로 설정된 유효 플래그의 값을 갖는 레코드들 중에서 첫번째 레코드(제 1 레코드)가 레코드 s로서 선택된다. 소거완료 어드레스로부터 종료 어드레스까지의 데이터 길이가 단위 길이 X보다 긴지 짧은지가 판정되어 단계 S31 이후에 수행되는 단계가 결정된다. 이 판정은 단계 S32에서 행해진다. 즉, 단계 S32에서, 소거완료 어드레스에 오버라이팅의 단위 길이 X를 가산함으로써 취득된 어드레스가 레코드 s의 종료 어드레스보다 작은지의 여부가 판정된다. 판정결과, 취득된 어드레스가 레코드 s의 종료 어드레스보다 작으면, 단계 S33~S35가 계속 수행된다. 이들 단계에서, 외부 메모리 제어부(4)는 대응 프래그먼트의 소거완료 어드레스로부터 시작하는 X바이트의 데이터를 오버라이트하라는 지시를 받는다(단계 S33). 외부 메모리 제어부(4)에 의한 오버라이팅이 완료되면(단계 S34), 소거완료 어드레스는 (소거완료 어드레스 + X)로 갱신된다(단계 S35).

소거완료 어드레스로부터 종료 어드레스까지의 데이터 길이가 단위 길이 X보다 길면, 단계 S36~S39가 수행되어 잉여 데이터 길이를 처리한다. 이들 단계에 의해, 단위 길이 X는, 종료 어드레스로부터 소거완료 어드레스를 감산하여 구한 값과 동일하게 되도록 변환된다(단계 S36). 그 후, 외부 메모리 제어부(4)는 소거완료 어드레스로부터 시작하는 X 바이트 데이터를 널 코드로 오버라이트하라는 지시를 받는다(단계 S37). 오버라이팅이 완료되면 레코드 s의 유효 플래그의 값은 "0"으로 설정된다(단계 S39). 단계 S40에서는, 유효 플래그의 값이 모든 레코드에서 "0"으로 설정되는지의 여부가 판정된다. 유효 플래그의 값이 모든 레코드에서 "0"으로 설정되면, 프로시져는 단계 S41로 진행하여 소거동작 관리 테이블이 삭제된다.

여기에서, 루트 디렉토리에 저장된 파일명이 EOB001.SE1인 파일을 삭제하기 위한 비명령 실행 핸들러(12), 판독/기입 핸들러(14), 및 삭제 핸들러(16)에 의한 동작에 대해, 예로서 도 20 및 도 21을 참조하여 설명하기로 한다. 도 20은, 도 10에 도시된 바와 같이 저장된 파일에 대한 삭제 이벤트에 응답하여, 삭제 핸들러(16)가 동작한 후의 파일에 대한 FAT와 루트 디렉토리 엔트리를 도시한 도면이다. 파일 "EOB001.SE1"에 대한 삭제 이벤트가 생성되면, 파일의 프래그먼트에 대한 소거동작 관리 테이블이 생성되고, 그 파일에 대한 파일 엔트리의 파일명 엔트리, 파일 확장자 엔트리, 및 제 1 클러스터 엔트리, 그리고 FAT 엔트리 003, 004, 005, 00A 및 00C가 널 코드로 오버라이트된다. 삭제 핸들러(16)에 의한 이러한 오버라이팅 후, 비명령 실행 핸들러(12)에 의한 동작이 시작된다.

도 21은 비명령 실행 핸들러(12)와 삭제 핸들러(16)에 의한 오버라이팅 후의 루트 디렉토리와 FAT를 도시한 도면이다. 파일의 프래그먼트에 대한 소거동작 관리 테이블은 삭제 핸들러(16)에 의해 생성되기 때문에, 프래그먼트를 저장하는 클러스터 003, 004, 005, 00A 및 00C는 널 코드로 오버라이트된다.

상술한 바와 같이, EC 클라이언트 어플리케이션(8)이 내린 EC 명령이 파일삭제를 지시하면, 널-소거동작은 타이머 핸들러(13)가 타임아웃을 알릴 때까지 삭제 핸들러(16)에 의해 연속적으로 수행된다. 삭제 핸들러(16)의 동작은 타임아웃의 발

생에 의해 종료된다. 따라서 EC 클라이언트 어플리케이션(8)이 내린 EC 명령으로부터 이벤트 분석 핸들러(11)가 한 EC 응답까지의 기간이 짧아질 수 있다. 이것으로 EC 명령을 처리하는데 필요한 시간을 단축할 수 있으므로 EC 클라이언트 어플리케이션(8)의 사용자를 실망시키지 않게 된다.

아직 널-소거되지 않은 파일의 데이터에 대한 오버라이팅 작업은, 복수의 유휴기간 내에 삭제 핸들러(16)를 제외한 비명령 실행 핸들러(12)와 판독/기입 해(14) 등의 다른 이벤트 핸들러에 의해 실행되도록 분할된다. 그러므로, 비록 EEPROM(3)에 대한 오버라이팅 동작이 긴 시간을 필요로 하고 각 프래그먼트가 큰 크기의 데이터를 갖는다 하더라도, 장시간의 삭제 프로시져로 인한 사용자의 지루함을 야기시키지 않고 파일 엔티티의 오버라이팅을 완료할 수 있다.

또한, 암호화된 파일을 판독하기 위해 중요한 암호화키는 바로 그 처음의 위치에서 널-소거된다. 그러므로, 비록 사용자가 삭제 핸들러(16)에 의한 삭제 프로시져의 도중에 SD 휴대형 장치(300)로부터 SDeX 메모리 카드(400)를 거칠게 인출해 내더라도, 암호화된 파일은 여전히 충분하게 보호된다.

(제 2 실시예)

제 1 실시예에 따라 널 코드로 프래그먼트를 오버라이팅하는 것은 시작 어드레스에서 프래그먼트의 종료 어드레스를 향해, 즉 순방향(forward direction)으로 수행된다. 제 2 실시예에 따르는 오버라이팅은 순방향일 뿐아니라 여러가지 상이한 방식으로 수행된다.

도 22는 제 2 실시예에 관한 소거동작 관리 테이블의 예를 도시한 도면이다. 도 22에 도시된 바와 같이, 테이블의 레코드는 추가로 소거방법의 엔트리를 갖는다.

소거방법 엔트리에는 순방향, 역방향, 짝수/홀수, 및 홀수/짝수 중의 하나가 설정된다. 소거방법 엔트리가 순방향으로 설정되면, 널 코드로 프래그먼트를 오버라이팅하는 것은 제 1 실시예에서 처럼 소거완료 어드레스에서 종료 어드레스를 향하여 순차적으로 수행된다. 소거방법 엔트리가 역방향으로 설정되면, 오버라이팅은 소거완료 어드레스에서 시작 어드레스를 향해 수행된다.

소거방법 엔트리가 짝수/홀수로 설정되면, 우선 짝수 번호 어드레스의 데이터가 널-소거되고, 다음으로 홀수 번호 어드레스의 데이터가 소거완료 어드레스에서 종료 어드레스를 향하여 널-소거된다.

소거방법 엔트리가 홀수/짝수로 설정되면, 우선 홀수 번호 어드레스의 데이터가 널-소거되고, 다음으로 짝수 번호 어드레스의 데이터가 소거완료 어드레스에서 종료 어드레스를 향하여 널-소거된다.

프래그먼트에 대한 소거방법은 삭제 핸들러(16)에 의해 차례로 선택될 수 있다. 보다 구체적으로, 순방향 널-소거동작은 최초의 프래그먼트에 대하여 선택되고, 역방향 널-소거동작은 두번째의 프래그먼트에 대하여 선택되며, 짝수/홀수 널-소거동작은 세번째 프래그먼트에 대하여 선택되고, 홀수/짝수 널-소거동작은 네번째 프래그먼트에 대하여 선택될 수 있다. 또한, 순방향, 역방향, 짝수/홀수, 및 홀수/짝수 동작은 다섯번째 프래그먼트와 그 이후의 프래그먼트에 대해 각각 선택될 수 있다.

이와 달리, 프래그먼트에 대한 소거방법은 랜덤하게 선택될 수도 있다. 구체적으로, 삭제 핸들러(16)에 의해 1에서 4까지의 난수가 생성되고, 그 난수 중의 하나에 의해 지시되는 소거방법으로 널-소거동작이 수행된다.

또한, 소거방법은 EC 클라이언트 어플리케이션(8)으로부터 수신된 파라미터에 따라 결정될 수 있다.

도 23은 제 2 실시예에 관한 널-소거 서브루틴 프로시져를 도시한 도면이다. 단계 S31에서, "1"로 설정된 유효 플래그의 값을 갖는 레코드 중 첫번째 레코드(제 1 레코드)가 레코드 s로서 선택된다. 레코드 s의 소거방법 엔트리는 단계 S51을 참조한다. 소거방법 엔트리가 순방향으로 설정되면, 널-소거동작은 단계 S50에서 도 19의 흐름도에 도시된 단계 S32~S40과 동일한 프로시져로 수행된다.

소거방법 엔트리가 역방향으로 설정되면(단계 S53), 널-소거동작은 도 24의 흐름도에 도시된 프로시져로 수행된다(단계 S60).

소거방법 엔트리가 짝수/홀수로 설정되면(단계 S54), 소거완료 어드레스 다음의 짝수번호 어드레스의 데이터가 도 19의 흐름도에 도시된 단계 S32~S40과 동일한 프로시져로 널-소거된다(단계 S55). 짝수번호 어드레스의 데이터가 완전히 널-소거되면(단계 S56), 소거완료 어드레스 다음의 홀수번호 어드레스의 데이터가 도 19의 흐름도에 도시된 단계 S32~S40과 동일한 프로시져로 널-소거된다(단계 S57).

소거방법 엔트리가 홀수/짝수로 설정되면(단계 S58), 소거완료 어드레스 다음의 홀수번호 어드레스의 데이터가 도 19의 흐름도에 도시된 단계 S32~S40과 동일한 프로시져로 널-소거된다(단계 S57). 홀수번호 어드레스의 데이터가 완전히 널-소거되면(단계 S59), 소거완료 어드레스 다음의 짝수번호 어드레스의 데이터가 도 19의 흐름도에 도시된 단계 S32~S40과 동일한 프로시져로 널-소거된다(단계 S55).

도 24는 역방향 널-소거동작의 프로시져를 도시한 흐름도이다. 도 19에 도시된 널-소거 서브루틴과 비교하면, 시작 어드레스와 종료 어드레스 사이의 관계가 역방향 널-소거동작과 반대이다. 단계 S61에서, 소거완료 어드레스에서 오버라이팅 단위 길이 X를 감산함으로써 구한 어드레스가 레코드 s의 시작 어드레스보다 큰지의 여부가 판정된다. 판정결과, 감산하여 구한 어드레스가 레코드 s의 시작 어드레스보다 크면, 단계 S62에서 외부 메모리 제어부(4)는 프래그먼트의 어드레스(소거완료 어드레스-X)로부터 X-바이트의 데이터를 오버라이트하라는 명령을 받는다. 외부 메모리 제어부(4)에 의한 오버라이팅이 완료되면(단계 S63), 단계 S64에서, 소거완료 어드레스는 (소거완료 어드레스-X)의 어드레스로 갱신된다.

어드레스(소거완료 어드레스-오버라이팅 단위 길이 X)가 레코드 s의 시작 어드레스보다 작으면, 단위길이 X는 소거완료 어드레스에서 시작 어드레스를 감산하여 구한 값과 동일하게 되도록 변환된다(단계 S65). 그리고, 단계 S66에서, 외부 메모리 제어부(4)는 시작 어드레스로부터의 X-바이트 데이터를 널 코드로 오버라이트하라는 명령을 받는다. 오버라이팅이 완료되면(단계 S67), 레코드 s의 유효 플래그의 값은 "0"으로 설정된다(단계 S68). 단계 S69에서, 유효 플래그의 값이 모든 레코드에서 "0"으로 설정되는지의 여부가 판정된다. 판정결과, 유효 플래그의 값이 모든 레코드에서 "0"으로 설정되면, 소거동작 관리 테이블은 단계 S70에서 삭제된다.

상술한 바와 같이, 제 2 실시예에서는 각 프래그먼트에 상이한 소거방법이 할당될 수 있다. 결국, 다양한 종류의 소거방법이 조합되므로 삭제되는 프래그먼트의 콘텐츠가 누설되지 않고 보호될 수 있다.

(제 3 실시예)

제 3 실시예에 따라, 반도체 메모리 카드가 카드 리더/라이터(200)로부터 전력을 공급받으면 제 1 실시예에 설명된 널 코드에 의한 오버라이팅이 제한된다. SDeX 메모리 카드(400)가 카드 리더/라이터(200)를 통해 EC 서버(100)에 액세스하면, SDeX 메모리 카드(400)는 제 1 실시예에 설명된 SD 휴대형 장치(300)의 배면에 있는 헬리컬 안테나를 통해 전력을 공급받는다. 전파에 의한 전력공급은 불안정하고 불충분하기 때문에 SDeX 메모리 카드(400)에 대한 불필요한 부담은 회피되는 것이 바람직하다. 제 3 실시예에 의하면 SD 휴대형 장치(300)와 삭제 핸들러(16)에 의해 다음의 동작이 수행된다.

SD 휴대형 장치(300)가 카드 리더/라이터(200)에 접근하면, 카드 리더/라이터(200)는 확장된 SD 명령을 출력하여 무접촉식으로 전력을 공급받았음을 SDeX 메모리 카드(400)에 통지한다. 반면, SD 휴대형 장치(300)가 카드 리더/라이터(200)로부터 멀어지면, 카드 리더/라이터(200)는 확장된 SD 명령을 출력하여 무접촉식으로 전력공급이 중단되었음을 SDeX 메모리 카드(400)에 통지한다. SD 휴대형 장치(300)가 카드 리더/라이터(200)에 근접하는지 멀어지는지의 여부는 SD 휴대형 장치(300)가 카드 리더/라이터(200)로부터 폴링 명령(polling command)을 수신하였는지의 여부에 의해 결정된다.

무접촉식 전력공급의 존재를 통지하기 위한 확장된 SD 명령이 SD 휴대형 장치(300)로부터 수신되면, 삭제 핸들러(16)는 무접촉식 전력공급 모드로 들어간다. 만약 삭제 핸들러(16)가 이 모드에 있으면, 파일 엔트리와 FAT의 백업 복제본이 생성되고, 삭제 이벤트의 생성에 따라 파일 엔트리와 FAT만이 오버라이트된다. 즉, 파일 엔트리와 FAT의 백업 복제본만이 생성되고, 어떠한 소거동작 관리 테이블도 생성되지 않는다. 따라서 널-소거동작은 다른 이벤트 핸들러에 의해 수행되지 않는다.

다음으로, 무접촉식 전력공급이 중단되지 않았음을 통지하기 위한 확장된 SD 명령을 SD 휴대형 장치(300)가 출력하지 않으면, 삭제 핸들러(16)는 무접촉식 전력공급 모드로부터 빠져나오고, 파일 엔트리와 FAT의 백업 복제본에 기초하여 소거동작 관리 테이블이 생성된다. 소거동작 관리 테이블이 생성되기 때문에 삭제 핸들러(16)를 제외한 다른 이벤트 핸들러에 의한 널-소거동작이 시작된다.

제 3 실시예에 따라, SDeX 메모리 카드(400)가 상술한 바와 같이 카드 리더/라이터(200)로부터의 전력공급에 의해 동작되면, 파일의 파일 엔티티에 대한 널-소거동작이 수행되지 않는다. 그러므로, SDeX 메모리 카드(400)에 불필요한 부담이 가해지지 않는다. 이것으로 SDeX 메모리 카드(400)의 안정적인 동작이 실현된다.

(제 4 실시예)

제 1 실시예, 제 2 실시예 및 제 3 실시예에서, TRM(1) 내의 메모리(3)와 외부 메모리(2)는 각각 EEPROM과 플래시 메모리로 구성된다. 그러나 제 4 실시예에 따라, TRM(1) 내의 메모리(3)와 외부 메모리(2)는 각각 2개의 메모리 모듈로 구성된다. 도 25는 제 4 실시예와 관련한 내부 메모리(3)와 외부 메모리(2)의 각각의 구성을 도시한 도면이다. 도 25에 도시된 바와 같이, EEPROM(3a)과 플래시 메모리(2a)는 주메모리 모듈이며, 제 1 내지 제 6 실시예의 EEPROM(3) 및 플래시 메모리(2)와 각각 동일하다. 제 4 실시예에서는 이들 주메모리 모듈 외에, 외부 메모리(2)와 내부 메모리(3)에 각각 보조 메모리 모듈(2b)과 보조 메모리 모듈(3b)이 구비된다. 이들 보조 메모리 모듈(2b, 3b)은 강유전체 메모리(Ferro Elelctric Random Access Memory: FeRAM)이며, 이들의 성능은 플래시 메모리의 성능과 상당히 다르다. 도 26은 플래시 메모리의 성능과 FeRAM의 성능을 비교한 도면이다. 도 26에 따르면 플래시 메모리는 값싸고 대용량 데이터를 저장하기에 적합하다(도면에서 ○으로 표시). 그러나 데이터 기입의 단위가 블록이다(※1로 표시됨). 여기에서, 이러한 블록의 크기는 플래시 메모리의 용량이 증가함에 따라 증가한다. 따라서 작은 크기의 데이터를 기입하더라도 큰 용량을 차지하게 된다. 또한 기입을 위한 시간이 길고(10,000ns), 가능한 기입동작의 횟수가 적다(1,000,000회). 또한, 데이터 기입은 저장된 데이터가 일단 삭제된 후에만 수행되므로, 기입성능이 불안정하게 된다(※2로 표시).

한편, FeRAM은 비록 비싸고 대용량 데이터를 저장하기에 적합하지는 않지만(△로 표시), 데이터 기입의 단위가 바이트이고 기입을 위한 시간이 짧다(30ns 내지 100ns). 또한, 가능한 기입동작의 횟수가 크다.

이러한 성능상의 차이점을 고려하여, 빈번하게 갱신되는 파일 엔트리, FAT 등을 저장하는 보조 메모리 모듈로서 FeRAM이 사용되면, 플래시 메모리(2a)의 기입성능이 보상될 수 있다. 도 27은 파일 엔트리, FAT 및 소거동작 관리 테이블 등의 빈번하게 갱신되는 데이터를 저장한 FeRAM을 도시한 도면이다.

제 4 실시예에 의하면, 파일 엔트리 및 FAT 등의 빈번하게 갱신되는 작은 크기의 데이터를 저장하는 보조 메모리 모듈로서 FeRAM이 사용된다. 이것으로 파일 엔트리와 FAT에 대한 고속 재기입 동작이 달성된다.

FeRAM은 파괴적 판독(destructive read out)의 특성을 갖는다(※4로 표시). 즉, 일단 저장된 데이터가 판독되면, 저장된 데이터의 콘텐츠가 저장매체로부터 삭제된다. 이 특성은 비밀을 완벽하게 보호하기 위해 바람직하다. 그러나 이 특성에 따르면 모든 데이터 판독은 판독된 데이터를 다시 기입할 것을 필요로 하며, 이것은 결국 데이터가 기록되는 횟수를 증가시킨다. 이러한 파괴적 판독의 특성을 피하기 위해서는 자기저항 RAM(Magnetoresistive Random Access Memory: MRAM)이 이용되는 것이 바람직하다.

(제 5 실시예)

제 4 실시예에서는 보조 메모리 모듈로서 FeRAM이 사용되었다. 그러나, 제 5 실시예에서는 TRM(1) 내의 메모리로 FeRAM만이 사용된다. 도 28은 제 5 실시예에 관한 내부 메모리(3)의 내부 구성을 도시한 도면이다. 여기에서 TRM(1) 내의 메모리(3)는 소형이다. 따라서, 비록 내부 메모리(3)에 FeRAM이 사용된다 하더라도 메모리(3)의 제조비용이 현저하게 증가하지는 않는다. 제 5 실시예에서는 TRM(1) 내의 메모리(3)가 FeRAM으로만 구성되어 있으나, MRAM으로만 구성될 수도 있다는 점에 유념해야 한다.

(제 6 실시예)

제 6 실시예에 의하면, 파일에 대한 관리정보는 그 파일의 파일 엔티티에 대한 암호화키와 다른 키로 암호화된다. 파일 관리정보에 대한 이러한 암호화키는 TRM(1) 내의 메모리(3)에 저장된다. 파일 삭제의 지시가 있으면, 파일 관리정보에 대한 암호화키의 오버라이팅은 파일 엔티티에 대한 암호화키의 오버라이팅에 앞서 제 1 실시예에서 설명된 프로시져와 삭제 핸들러(16)에 의해 수행된다.

파일에 대한 관리정보는 다른 암호화키로 암호화된다. 따라서, 파일의 삭제가 지시되면 관리정보의 크기에도 불구하고 파일의 판독은 즉시 행해질 수 없다.

(수정예)

상술한 설명은 본 발명의 모든 실시예를 언급한 것은 아니다. 본 발명은 다음의 수정사항 (A)~(E)를 포함하는 실시예에 의해 실현될 수 있다. 그러나, 당해 기술분야에서 통상의 기술을 갖는 자가 본 발명의 상세한 설명, 첨부도면 및 출원시의 공지의 기술에 기초하여 다음의 수정사항을 생각해 낼 수 있는 것이라면, 다음의 수정사항에는 청구항에 기재된 발명의 실시예가 포함되지 않는다.

(A) 실시예에서는 EC 클라이언트 어플리케이션을 예로 들었지만 본 발명은 다른 어플리케이션에 적용될 수 있다. 예를 들어, 본 발명은 철도, 항공, 버스 및 고속도로 회사와 같은 운송산업에 의해 소유되는 서버장치 상의 서버 어플리케이션과 대응 클라이언트 어플리케이션에 적용될 수 있다. 따라서, SDeX 메모리 카드(400)는 철도 개찰구에서 탑승절차를 위해 이용될 수 있다.

이와 달리, 본 발명은 정부기관 및 지방기관에 의해 소유되는 서버장치 상의 서버 어플리케이션과 대응 클라이언트 어플리케이션에 적용될 수 있다. 따라서, SDeX 메모리 카드(400)는 거주자 카드와 각종 증명 및 등록을 위해 이용될 수 있다.

(B) 도 15 내지 도 19, 도 23, 및 도 24에 도시된 프로그램에 의한 정보처리는 CPU 및 EEPROM 등의 하드웨어 자원에 의해 물리적으로 실현된다. 즉, 프로그램과 하드웨어 자원을 결합하여 형성된 물리적 장치가 특정 목적을 달성하기 위한 정보처리를 수행한다. 이러한 방식으로, 제 1 내지 제 6 실시예에 설명된 SDeX 메모리 카드(400)가 실현될 수 있다.

프로그램에 의한 정보처리는 하드웨어 자원에 의해 물리적으로 실현된다. 따라서, 상술한 도면에 프로시져가 도시된 프로그램은 자연법칙을 이용한 기술적 사상의 제품으로서 고려될 수 있으며 프로그램 자체도 발명으로서 간주될 수 있다. 따라서, 도 15 내지 도 19, 도 23 및 도 24는 본 발명에 따르는 프로그램의 실시예를 나타내고 있다.

제 1 내지 제 6 실시예는 본 발명에 따르는 프로그램의 이용에 관한 실시예를 설명하며, 여기에서 프로그램은 SDeX 메모리 카드(400)에 통합된다. 그러나 제 1 내지 제 6 실시예에 예시된 프로그램은 이용을 위해 SDeX 메모리 카드(400)로부터 분리될 수도 있다. 여기에서, 프로그램 자체의 이용은 (1) 프로그램의 제조 행위, (2) 프로그램의 무상양도 또는 유상양도의 행위, (3)프로그램 임대의 행위, (4) 프로그램 수입의 행위, (5) 양방향 전기통신회선을 통해 공중(公衆)에 프로그램을 공급하는 행위, 및 (6) 진열, 및 팜플렛과 카탈로그의 배포를 통해 프로그램을 광고함으로써 공중의 사용자에게 프로그램을 양도하거나 임대하는 행위를 나타낸다.

양방향 전기통신회선을 통해 프로그램을 공급하는 행위(5)의 전형적인 예로는 프로그램 다운로드 서비스와 어플리케이션 서비스 공급자(Application Service Provider: ASP) 서비스가 있다. 프로그램 다운로드 서비스는 공급자가 프로그램을 사용자에게 송신하여 사용자가 프로그램을 사용할 수 있게 하고 있다. 또, 어플리케이션 서비스 공급자 서비스는 전기통신회선을 통해 프로그램의 기능을 사용자에게 공급하지만, 프로그램 자체는 공급자가 가지고 있다.

(C) 도 15 내지 도 19, 도 23 및 도 24의 각 흐름도에서의 단계들의 순서 등의 프로시져에 대한 시간적인 순서는 본 발명을 특정하기 위한 기본적인 것으로서 고려된다. 따라서, 상술한 흐름도의 각각에 나타낸 프로시져는 제어방법이 사용되는 방법을 보여주고 있다. 그러므로, 이들 흐름도는 본 발명에 따르는 제어방법의 사용에 대한 실시예를 나타낸다. 각 흐름도에서의 단계들이 본 발명의 원래의 목적과 효과를 달성하기 위해 상술한 시간적인 순서로 실행된다면, 이들 흐름도의 프로시져는 의심할 나위없이 본 발명에 따르는 반도체 메모리 카드에 대한 제어방법의 실시예이다.

(D) 제 1 내지 제 6 실시예에서, TRM(1)의 내외부에 대하여 비휘발성 메모리로서 EEPROM이 사용되었다. 그러나, 내부 메모리 및 외부 메모리는 FeRAM 등의 다른 비휘발성 메모리일 수 있다.

(E) SD 휴대형 장치(300)의 예로서 이동전화 형식을 취하였다. 그러나 SD 휴대형 장치는 소비자의 휴대형 오디오장치, 셋톱박스(Set Top Box: STB), 또는 이동전화일 수 있다.

본 발명의 청구항에 정의된 발명은 상술한 실시예 및 그 수정예를 확장하거나 일반화한 것이다. 확장과 일반화의 정도는 출원시점에서의 관련기술의 상태에 기초한 것이다.

그러나, 청구항에 정의된 발명은 관련 기술의 기술적 문제점을 해결하기 위한 수단에 기초한 것이기 때문에, 본 발명의 범위는 관련 기술분야의 통상의 기술을 가진 자에 의해 실현되는 관련 기술의 범위로부터 벗어나지 않는다. 그러므로, 본 청구항에 정의된 각 발명은 상세한 설명에 기재된 내용에 실질적으로 대응하는 것이다.

산업상 이용 가능성

본 발명에 따르는 반도체 메모리 카드는 삭제된 파일에 대한 높은 보호를 제공하므로, 신뢰성 있는 데이터를 저장하기에 이상적이다. 따라서, 본 발명에 따르는 반도체 메모리 카드는 소비자 용품 분야와 같은 산업의 다양한 분야에 이용될 수 있다.

부호의 설명

1 : TRM 2 : 외부 EEPROM

3 : 내부 EEPROM 4 : 외부 메모리 제어부

5 : HIM 6 : 마스크 ROM

7 : CPU 8 : 클라이언트 어플리케이션

9 : 가상머신 10 : API

11: 이벤트 분석 핸들러 12 : 비명령 실행 핸들러

13: 타이머 핸들러 14 : 판독/기입 핸들러

15: 삭제 핸들러 21 : 어플리케이션 프로그램용 영역

22: 보안영역 23 : 인증영역

24: 비인증 영역 100 : 카드 리더/라이터

210: 무선 기지국 300 : 휴대형 장치

(57) 청구의 범위

청구항 1.

엔티티와 관리정보로 구성되는 파일을 저장하는 비휘발성 메모리; 및

처리부와 내부 메모리를 포함하는 부정변경 방지 모듈을 포함하며,

상기 처리부는,

상기 파일에 대한 삭제 이벤트가 생성되면, (i)상기 부정변경 방지 모듈 내의 상기 내부 메모리에 로케이션 테이블을 생성하고, (ii)상기 관리정보를 오버라이트하는 삭제부를 포함하며,

상기 로케이션 테이블은, 상기 엔티티의 위치를 나타내고, 상기 삭제부가 상기 엔티티를 오버라이트할 때 상기 삭제부에 의해 참조되는 것을 특징으로 하는 반도체 메모리 카드.

청구항 2.

제 1 항에 있어서,

상기 처리부는, (i)상기 반도체 메모리 카드가 접속된 장치가 내린 명령에 대응하는 동작을 수행하며, (ii) 상기 장치가 내린 명령을 분석하고, 그 분석결과에 대응하는 이벤트를 생성하는 분석부를 추가로 포함하며,

상기 삭제부는,

상기 삭제 이벤트가 생성되면 상기 로케이션 테이블을 생성하고, 상기 관리정보를 오버라이트하는 주삭제부; 및

상기 삭제 이벤트와 다른 이벤트가 생성되면, 상기 생성된 로케이션 테이블을 참조하여 상기 엔티티를 오버라이트하는 부삭제부를 포함하는 것을 특징으로 하는 반도체 메모리 카드.

청구항 3.

제 2 항에 있어서,

상기 삭제 이벤트가 생성되는 시간을 측정하기 시작하는 타이머를 추가로 포함하고,

상기 주삭제부는 상기 타이머가 타임아웃을 알릴 때까지 상기 관리정보에 부가하여 상기 엔티티의 일부분을 오버라이트하고,

상기 부삭제부는 상기 주삭제부에 의해 오버라이트되지 않는 상기 엔티티의 나머지 부분을 오버라이트하는 것을 특징으로 하는 반도체 메모리 카드.

청구항 4.

제 3 항에 있어서,

상기 로케이션 테이블은 (i)상기 엔티티 중 얼마나 많은 엔티티가 오버라이트되었는지를 나타내는 소거완료 어드레스와, (ii)복수의 오버라이팅 방법 중에서 선택된 엔티티를 오버라이팅하는 방법을 나타내며,

상기 부삭제부는, 상기 로케이션 테이블이 나타내는 오버라이팅 방법에 따라 상기 엔티티를 오버라이트하고, 상기 부삭제부가 상기 엔티티의 미리 정해진 길이를 오버라이트할 때마다 상기 로케이션 테이블에 나타나는 소거완료 어드레스를 갱신하는 것을 특징으로 하는 반도체 메모리 카드.

청구항 5.

제 2 항에 있어서,

상기 삭제 이벤트는, 상기 반도체 메모리 카드가 접속된 상기 장치가 상기 파일을 삭제하라는 명령을 내릴 때 상기 분석부에 의해 생성되고,

상기 다른 이벤트는, 상기 처리부가 상기 명령에 대응하는 동작을 완료할 때 상기 분석부에 의해 생성되는 명령종료 이벤트인 것을 특징으로 하는 반도체 메모리 카드.

청구항 6.

제 5 항에 있어서,

상기 주삭제부는 상기 삭제 이벤트의 생성에 응답하여 동작하기 시작하는 삭제 핸들러이며,

상기 부삭제부는 상기 명령종료 이벤트의 생성에 응답하여 동작하기 시작하는 비명령 실행 핸들러인 것을 특징으로 하는 반도체 메모리 카드.

청구항 7.

제 2 항에 있어서,

상기 삭제 이벤트는, 상기 반도체 메모리 카드가 접속된 상기 장치가 상기 파일을 삭제하라는 명령을 내릴 때 상기 분석부에 의해 생성되고,

상기 다른 이벤트는 판독 이벤트와 기입 이벤트 중의 하나이며, 상기 판독 이벤트는 상기 장치가 상기 반도체 메모리 카드 내의 다른 파일을 판독하라는 명령을 내릴 때 상기 분석부에 의해 생성되고, 상기 기입 이벤트는 상기 장치가 상기 반도체 메모리 카드 내의 상기 다른 파일을 기입하라는 명령을 내릴 때 상기 분석부에 의해 생성되는 것을 특징으로 하는 반도체 메모리 카드.

청구항 8.

제 7 항에 있어서,

상기 주삭제부는 상기 삭제 이벤트의 상기 생성에 응답하여 동작하기 시작하는 삭제 핸들러이며,

상기 부삭제부는 상기 판독 이벤트 및 상기 기입 이벤트 중의 하나의 생성에 응답하여 동작하기 시작하는 판독/기입 핸들러에 포함되는 것을 특징으로 하는 반 도체 메모리 카드.

청구항 9.

제 2 항에 있어서,

상기 내부 메모리는 상기 엔티티가 암호화되는 암호화키를 저장하고,

상기 삭제 이벤트가 생성되면, 상기 주삭제부는 상기 파일의 관리정보를 오버라이트하기 전에 상기 암호화키를 오버라이트하는 것을 특징으로 하는 반도체 메모리 카드.

청구항 10.

제 9 항에 있어서,

상기 파일의 상기 관리정보는 상기 엔티티에 대한 상기 암호화키와 다른 암호화키로 암호화되고, 상기 관리정보에 대한 상기 암호화키는 상기 내부 메모리에 저장되며,

상기 삭제 이벤트가 생성되면, 상기 주삭제부는 상기 엔티티에 대한 상기 암호화키를 오버라이트하기 전에 상기 관리정보에 대한 상기 암호화키를 오버라이트하는 것을 특징으로 하는 반도체 메모리 카드.

청구항 11.

제 9 항에 있어서,

상기 파일의 상기 엔티티는 프래그먼트들로 분할되고,

상기 로케이션 테이블은 (i)상기 엔티티를 구성하는 상기 프래그먼트의 각각의 시작 어드레스와, (ii)상기 프래그먼트의 각각이 완전히 오버라이트되지 않으면 "오프"로 설정되고, 상기 프래그먼트가 완전히 오버라이트되면 "온"으로 설정되는 플래그를 나타내는 것을 특징으로 하는 반도체 메모리 카드.

청구항 12.

제 1 항에 있어서,

임의의 장치로부터의 접촉식 또는 무접촉식의 전력공급수단에 의해 동작되며,

상기 반도체 메모리 카드가 상기 장치로부터의 상기 접촉식 전력공급수단에 의해 동작할 때에만 상기 처리부가 상기 오버라이팅을 수행하는 것을 특징으로 하는 반도체 메모리 카드.

청구항 13.

제 1 항에 있어서,

상기 비휘발성 메모리는 제 1 메모리 모듈과 제 2 메모리 모듈로 구성되며,

상기 제 2 메모리 모듈에 대한 기입의 단위는 상기 제 1 메모리 모듈에 대한 기입의 단위보다 작고,

상기 관리정보는 상기 제 2 메모리 모듈에 저장되는 것을 특징으로 하는 반도체 메모리 카드.

청구항 14.

제 13 항에 있어서,

상기 제 2 메모리 모듈은 강유전체 메모리(Ferro Electric Random Access Memory: FeRAM)와 자기저항 메모리(Magnetoresistant Random Access Memory: MRAM) 중의 하나인 것을 특징으로 하는 반도체 메모리 카드.

청구항 15.

제 1 항에 있어서,

상기 내부 메모리는 제 1 메모리 모듈과 제 2 메모리 모듈로 구성되며,

상기 제 2 메모리 모듈에 대한 기입의 단위는 상기 제 1 메모리 모듈에 대한 기입의 단위보다 작고,

상기 로케이션 테이블은 상기 제 2 메모리 모듈에 저장되는 것을 특징으로 하는 반도체 메모리 카드.

청구항 16.

제 15 항에 있어서,

상기 제 2 메모리 모듈은 강유전체 메모리(FeRAM)와 자기저항 메모리(MRAM) 중의 하나인 것을 특징으로 하는 반도체 메모리 카드.

청구항 17.

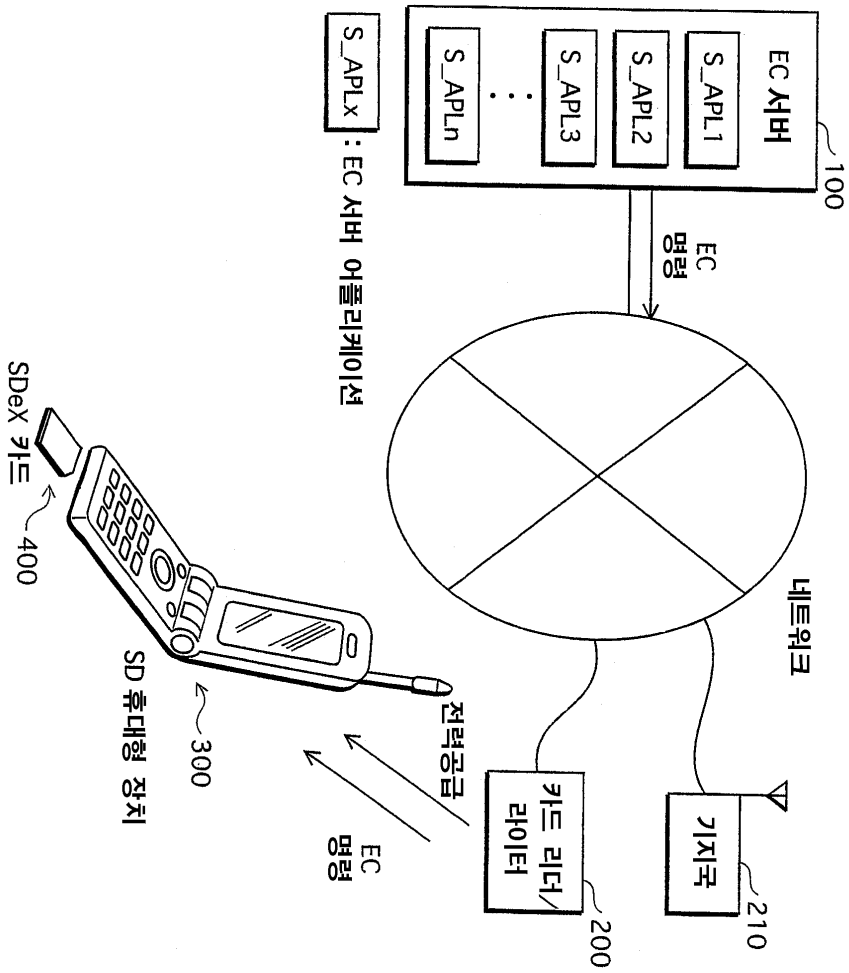
반도체 메모리 카드에 포함되는 부정변경 방지 모듈 내의 CPU에 의해 실행되는 제어 프로그램으로서, 상기 부정변경 방지 모듈은 내부 메모리를 포함하고, 상기 반도체 메모리 카드는 엔티티와 관리정보로 구성되는 파일을 저장하며,

상기 파일에 대한 삭제 이벤트가 생성되면, 상기 제어 프로그램은 (i)상기 부정변경 방지 모듈 내의 상기 내부 메모리 상에 로케이션 테이블을 생성하고, (ii)상기 관리정보를 오버라이트하며,

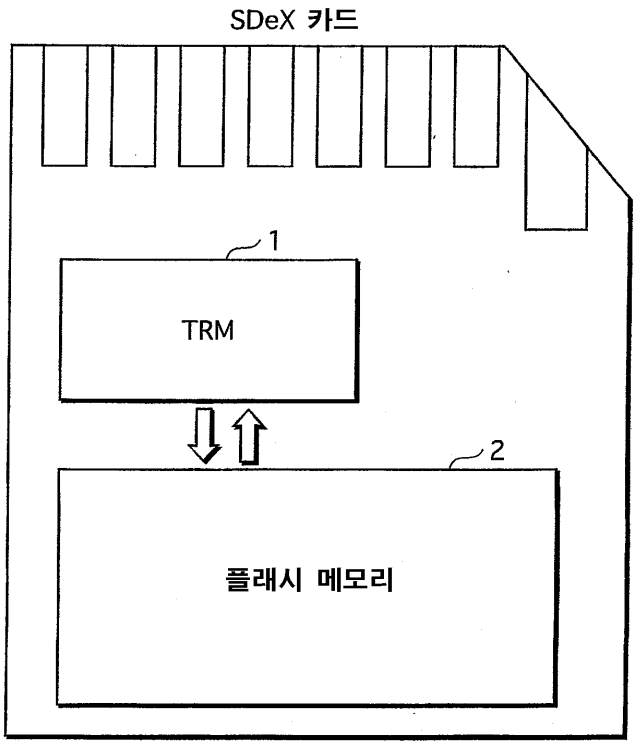
상기 로케이션 테이블은, 상기 엔티티의 위치를 표시하며, 상기 제어 프로그램이 상기 엔티티를 오버라이트할 때 상기 제어 프로그램에 의해 참조되는 것을 특징으로 하는 제어 프로그램.

도면

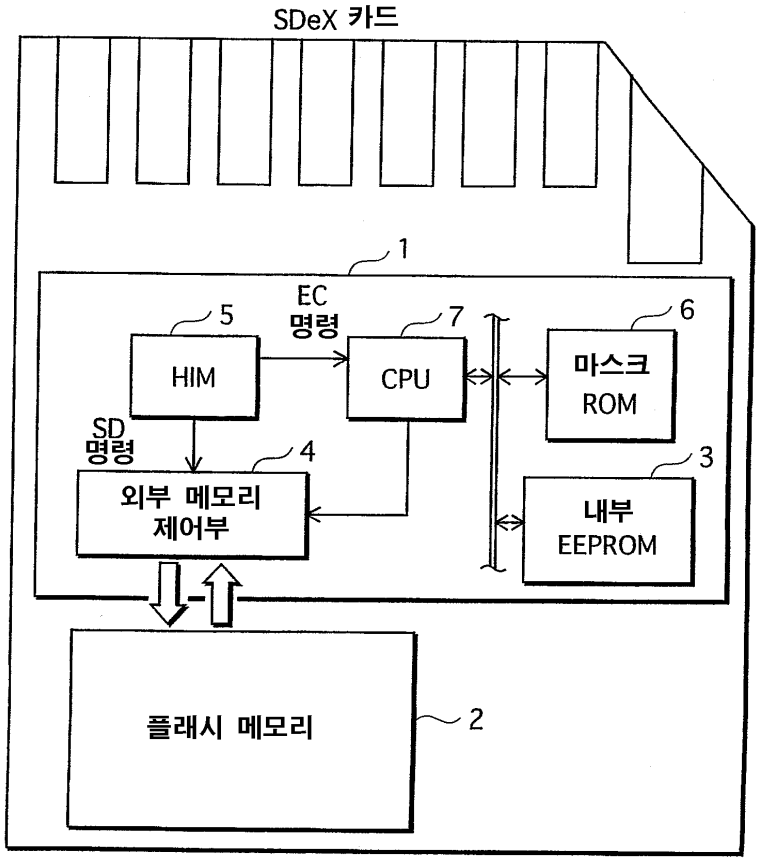
도면1



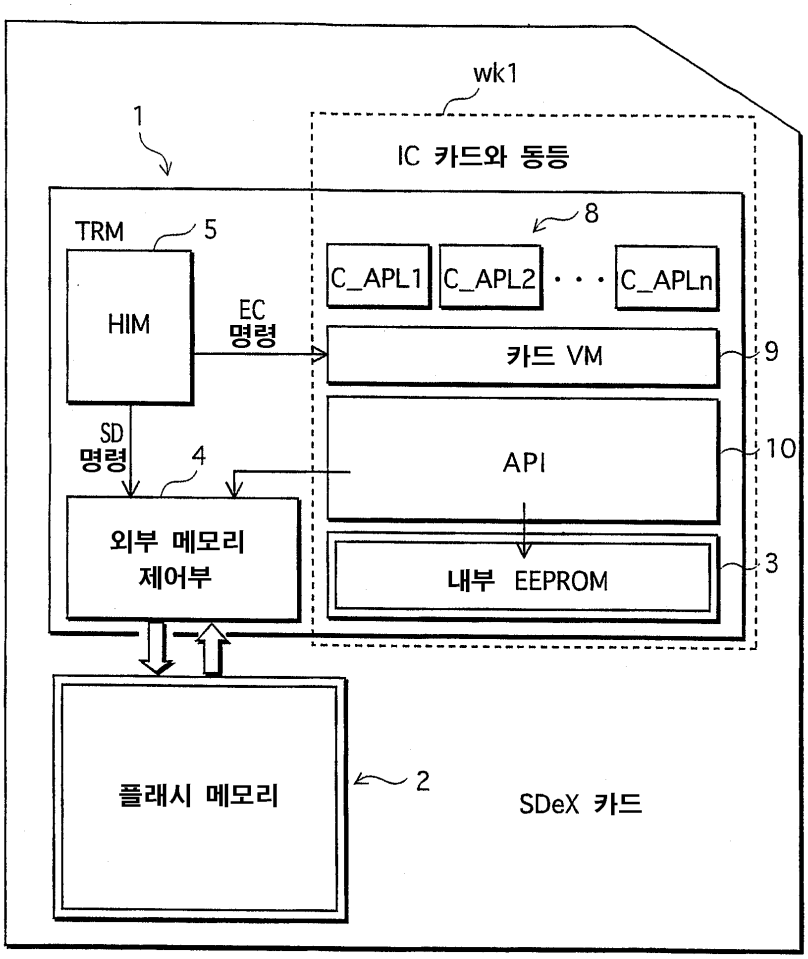
도면2



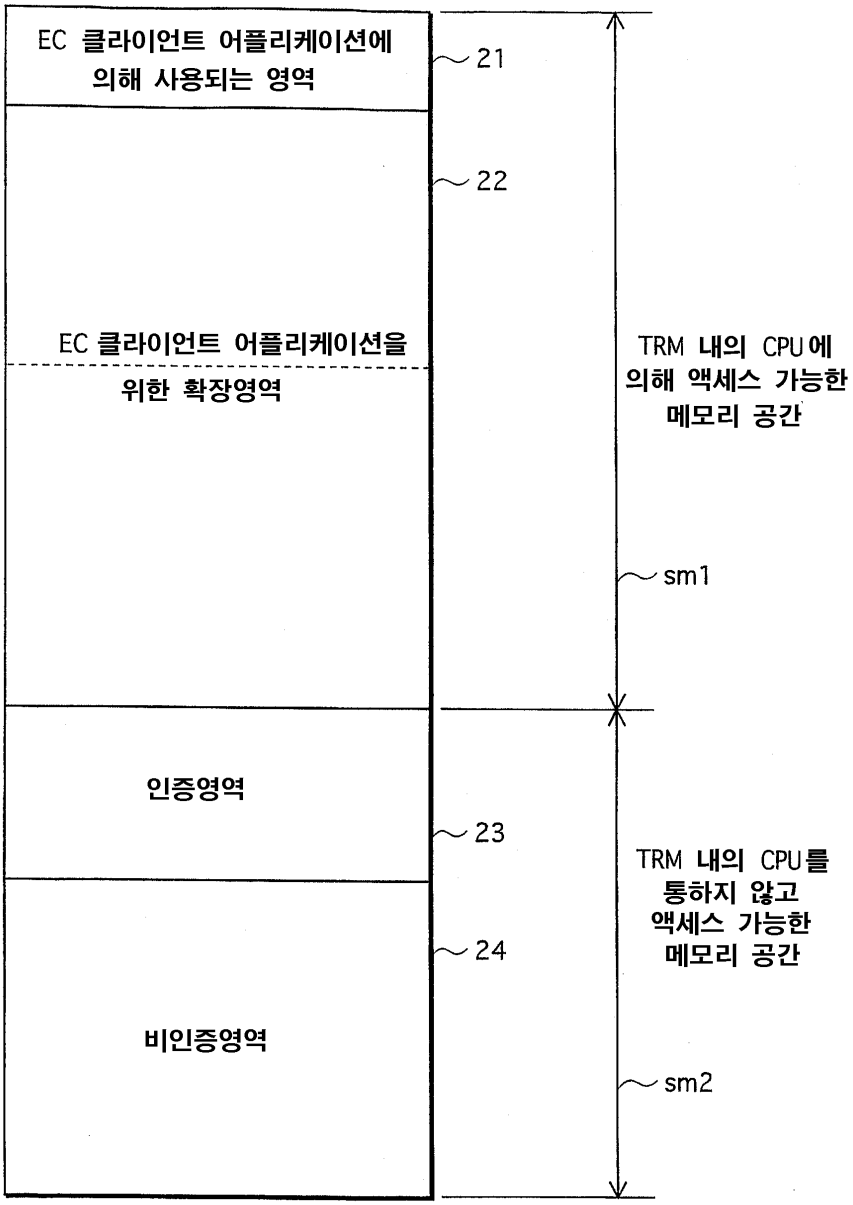
도면3


SDeX 카드
1
EC
명령
5
HIM
7
CPU
6
마스크
ROM
SD
명령
4
외부 메모리
제어부
3
내부
EEPROM
플래시 메모리
2

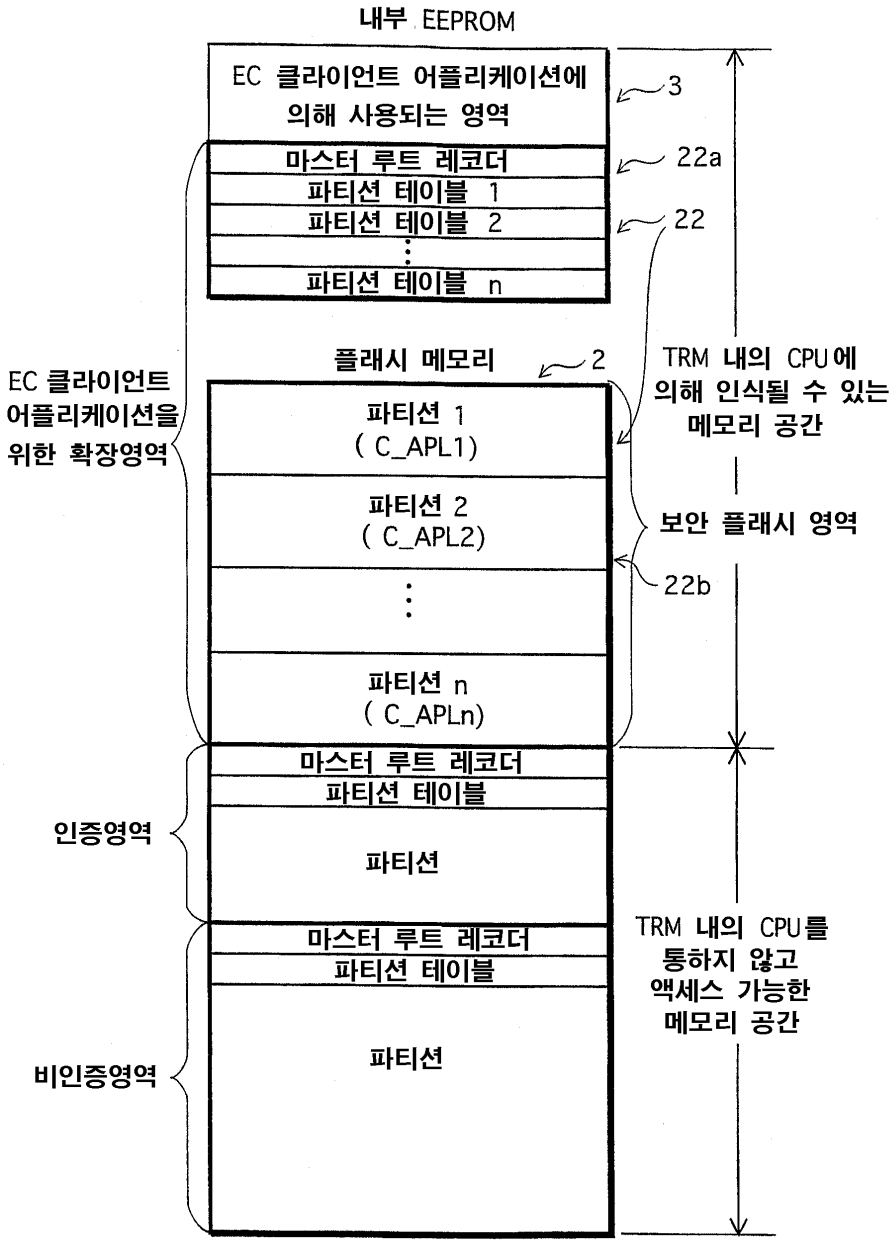
도면4


wk1
1
IC 카드와 동등
TRM
5
8
HIM
EC
명령
C_APL1
C_APL2
C_APLn
카드 VM
9
SD
명령
4
API
10
외부 메모리
제어부
내부 EEPROM
3
플래시 메모리
2
SDeX 카드

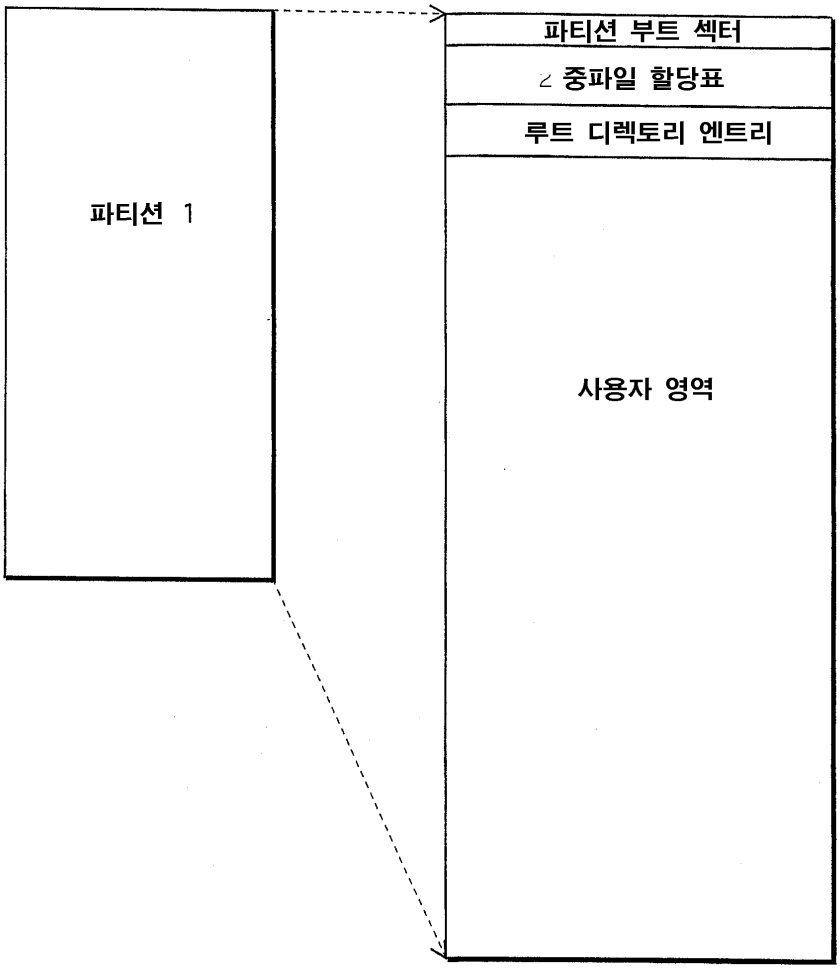
도면5


EC 클라이언트 어플리케이션에
의해 사용되는 영역
21
22
EC 클라이언트 어플리케이션을
위한 확장영역
TRM 내의 CPU에
의해 액세스 가능한
메모리 공간
sm1
인증영역
23
TRM 내의 CPU를
통하지 않고
액세스 가능한
메모리 공간
24
비인증영역
sm2

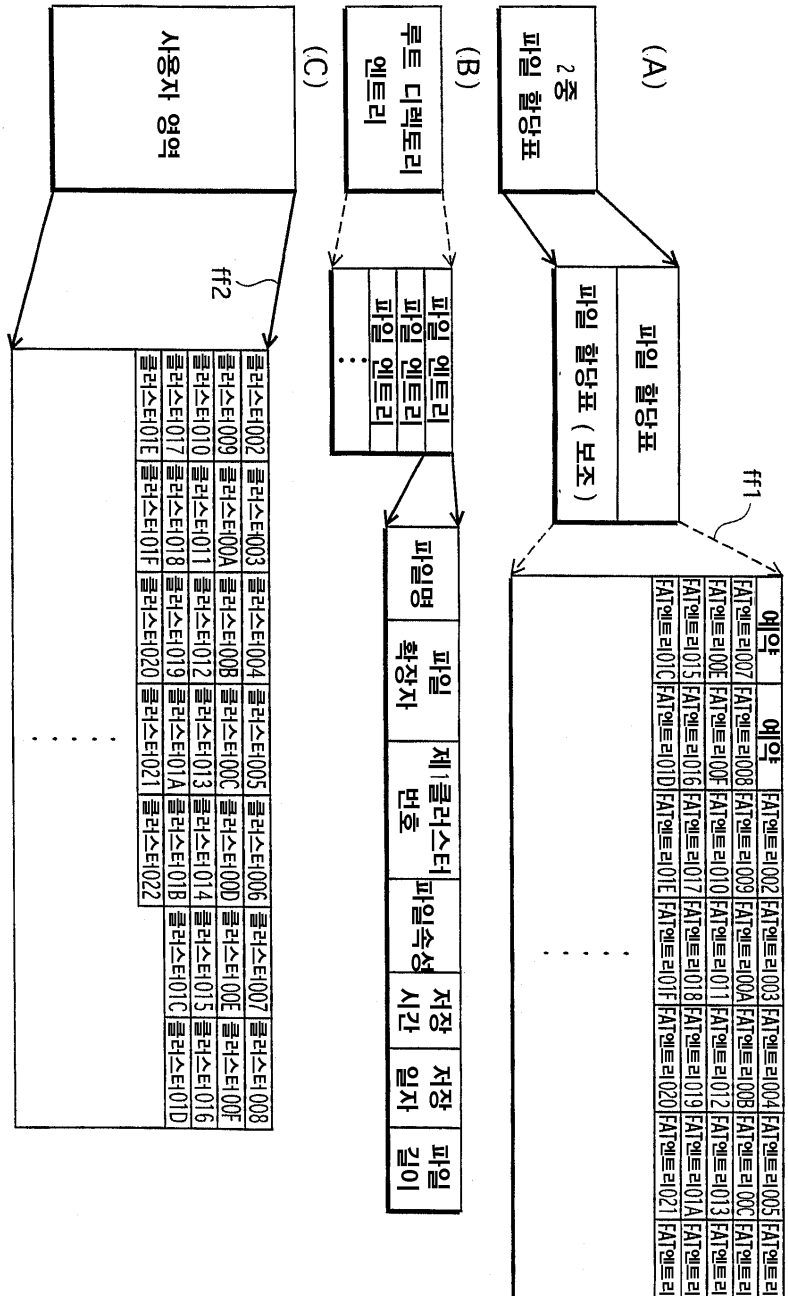
도면6


내부 EEPROM
EC 클라이언트 어플리케이션에 의해 사용되는 영역
3
마스터 루트 레코더
22a
파티션 테이블 1
파티션 테이블 2
22
파티션 테이블 n
플래시 메모리
2
TRM 내의 CPU에 의해 인식될 수 있는 메모리 공간
EC 클라이언트 어플리케이션을 위한 확장영역
파티션 1 (C_APL1)
파티션 2 (C_APL2)
보안 플래시 영역
22b
파티션 n (C_APLn)
마스터 루트 레코더
파티션 테이블
인증영역
파티션
마스터 루트 레코더
파티션 테이블
TRM 내의 CPU를 통하지 않고 액세스 가능한 메모리 공간
비인증영역
파티션

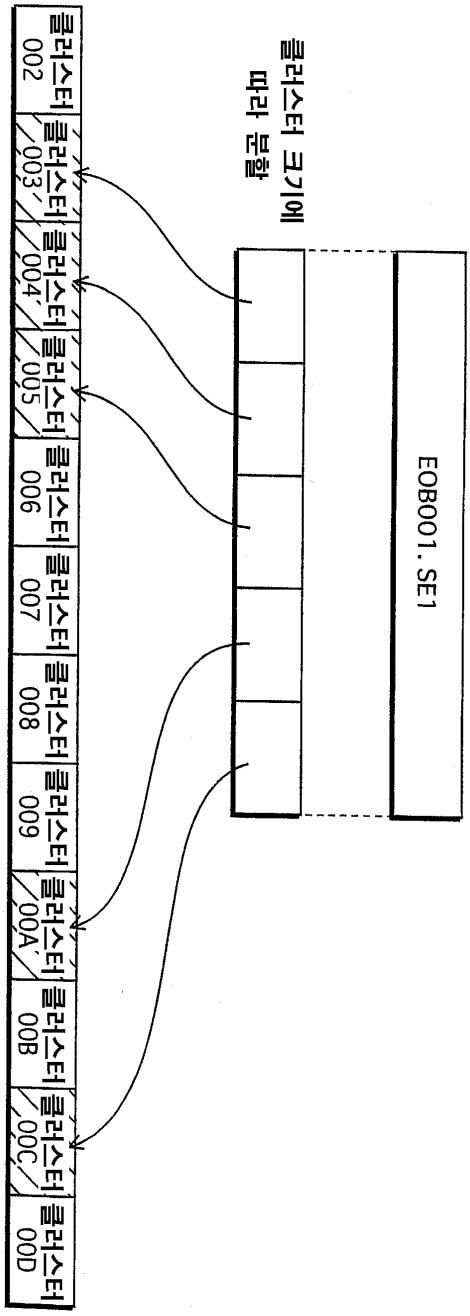
도면7



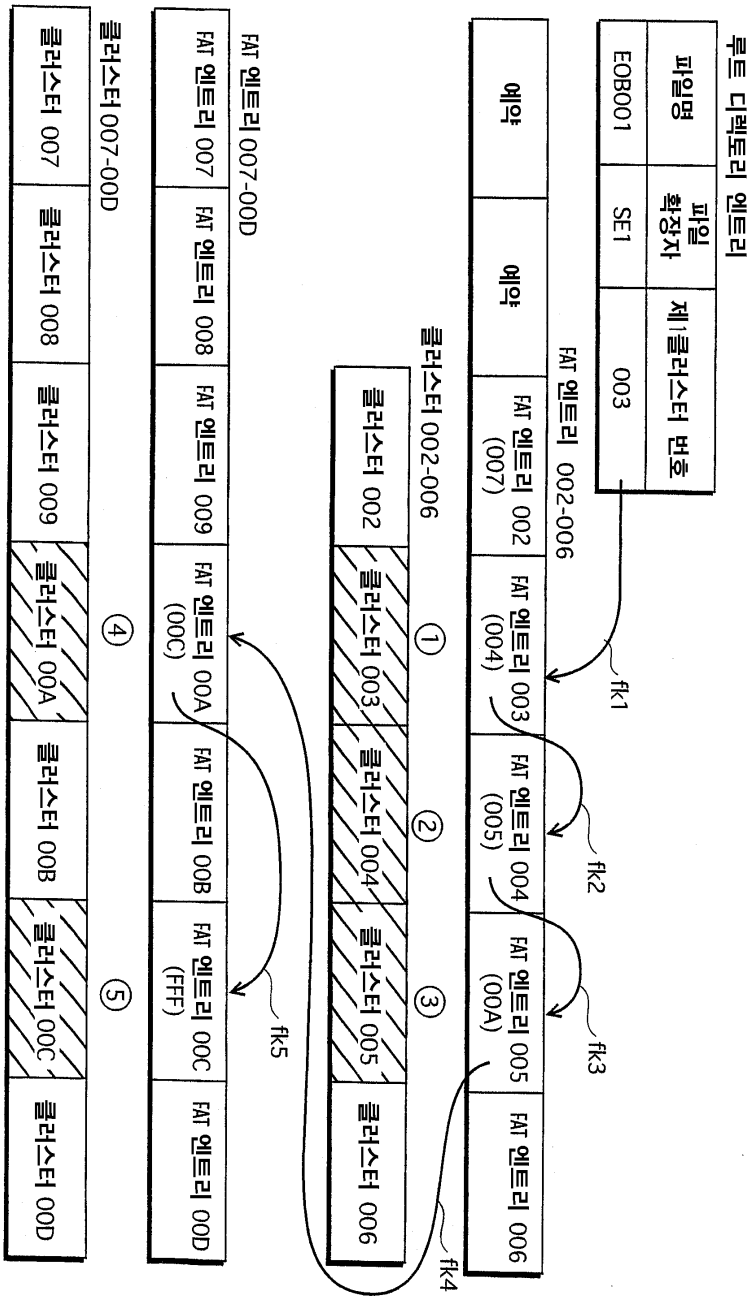
도면8


(A)
2중 파일 할당표
파일 할당표
파일 할당표 (보조)
ff1
예약
예약
FAT엔트리002
FAT엔트리003
FAT엔트리004
FAT엔트리005
FAT엔트리006
FAT엔트리007
FAT엔트리008
FAT엔트리009
FAT엔트리00A
FAT엔트리00B
FAT엔트리00C
FAT엔트리00D
FAT엔트리00E
FAT엔트리00F
FAT엔트리010
FAT엔트리011
FAT엔트리012
FAT엔트리013
FAT엔트리014
FAT엔트리015
FAT엔트리016
FAT엔트리017
FAT엔트리018
FAT엔트리019
FAT엔트리01A
FAT엔트리01R
FAT엔트리01C
FAT엔트리01D
FAT엔트리01E
FAT엔트리01F
FAT엔트리020
FAT엔트리021
FAT엔트리022
(B)
루트 디렉토리 엔트리
파일 엔트리
파일 엔트리
파일 엔트리
파일명
파일 확장자
제1클러스터 번호
파일속성
저장 시간
저장 일자
파일 길이
(C)
사용자 영역
ff2
클러스터002
클러스터003
클러스터004
클러스터005
클러스터006
클러스터007
클러스터008
클러스터009
클러스터00A
클러스터00B
클러스터00C
클러스터00D
클러스터00E
클러스터00F
클러스터010
클러스터011
클러스터012
클러스터013
클러스터014
클러스터015
클러스터016
클러스터017
클러스터018
클러스터019
클러스터01A
클러스터01B
클러스터01C
클러스터01D
클러스터01E
클러스터01F
클러스터020
클러스터021
클러스터022

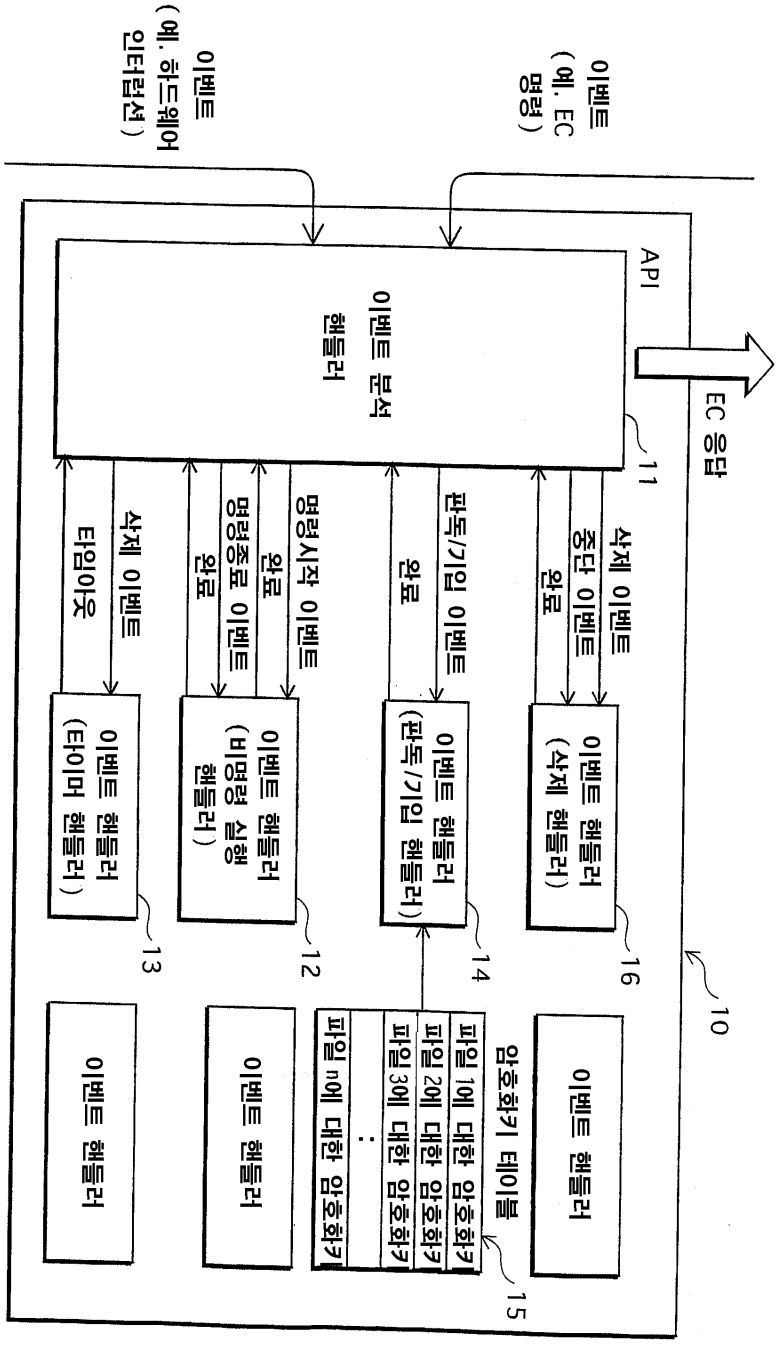
도면9


클러스터 크기에 따라 분할
EOB001.SE1
클러스터 002
클러스터 003
클러스터 004
클러스터 005
클러스터 006
클러스터 007
클러스터 008
클러스터 009
클러스터 00A
클러스터 00B
클러스터 00C
클러스터 00D

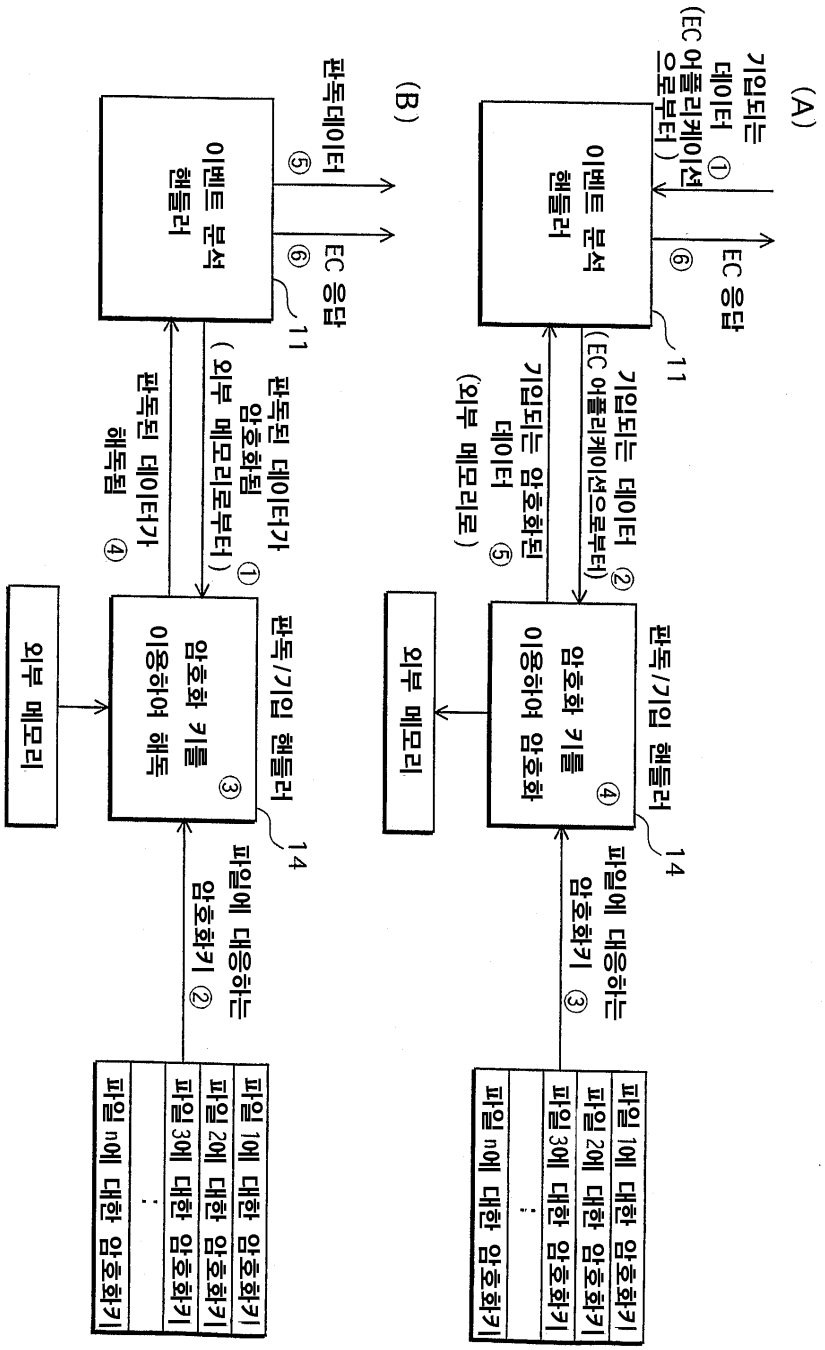
도면10


루트 디렉토리 엔트리
파일명
파일 확장자
제1클러스터 번호
EOB001
SE1
003
FAT 엔트리 002-006
예약
예약
FAT 엔트리 002 (007)
FAT 엔트리 003 (004)
FAT 엔트리 004 (005)
FAT 엔트리 005 (00A)
FAT 엔트리 006
fk1
fk2
fk3
fk4
fk5
클러스터 002-006
클러스터 002
클러스터 003
클러스터 004
클러스터 005
클러스터 006
①
②
③
FAT 엔트리 007-00D
FAT 엔트리 007
FAT 엔트리 008
FAT 엔트리 009
FAT 엔트리 00A (00C)
FAT 엔트리 00B (FFF)
FAT 엔트리 00C
FAT 엔트리 00D
클러스터 007-00D
클러스터 007
클러스터 008
클러스터 009
클러스터 00A
클러스터 00B
클러스터 00C
클러스터 00D
④
⑤

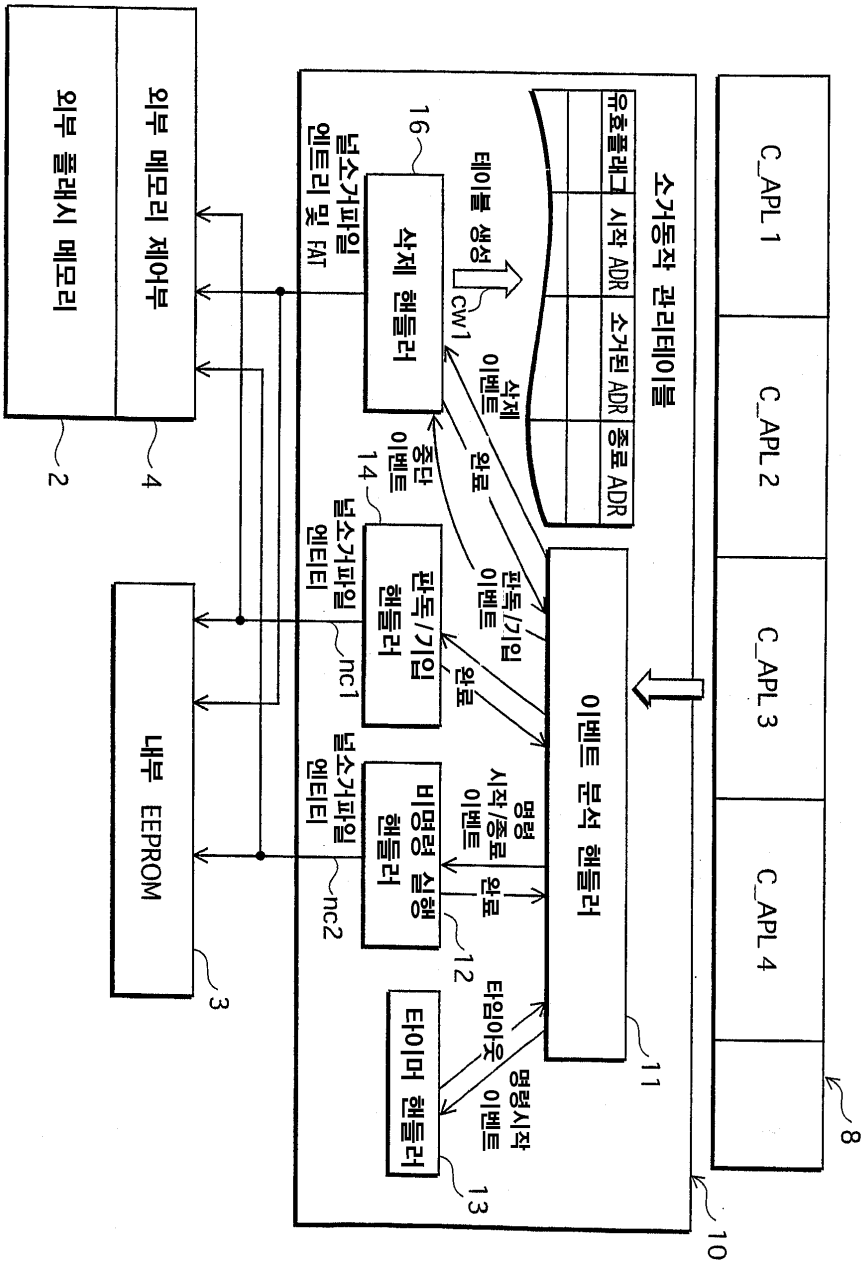
도면11


이벤트
(예. EC
명령)
이벤트
(예. 하드웨어
인터럽션)
API
EC 응답
이벤트 분석
핸들러
11
삭제 이벤트
중단 이벤트
완료
판독/기입 이벤트
완료
명령시작 이벤트
완료
명령종료 이벤트
삭제 이벤트
타임아웃
이벤트 핸들러
(삭제 핸들러)
16
이벤트 핸들러
(판독/기입 핸들러)
14
이벤트 핸들러
(비명령 실행
핸들러)
12
이벤트 핸들러
(타이머 핸들러)
13
이벤트 핸들러
이벤트 핸들러
이벤트 핸들러
암호화키 테이블
파일 1에 대한 암호화키
파일 2에 대한 암호화키
파일 3에 대한 암호화키
:
파일 n에 대한 암호화키
15
10

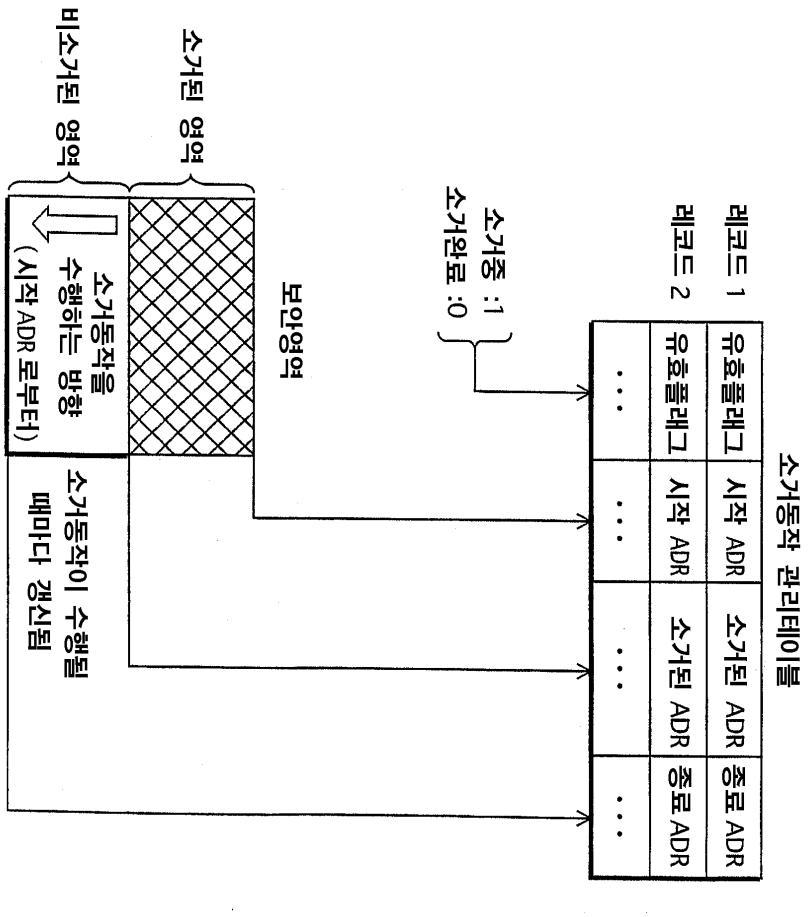
도면12


(A)
기입되는 데이터 ① (EC 어플리케이션으로부터)
이벤트 분석 핸들러
EC 응답
⑥
11
기입되는 데이터 ② (EC 어플리케이션으로부터)
기입되는 암호화된 데이터 ⑤ (외부 메모리로)
암호화 키를 이용하여 암호화 ④
판독/기입 핸들러
14
파일에 대응하는 암호화키 ③
파일 1에 대한 암호화키
파일 2에 대한 암호화키
파일 3에 대한 암호화키
파일 n에 대한 암호화키
외부 메모리
(B)
판독데이터 ⑤
이벤트 분석 핸들러
EC 응답
⑥
11
판독된 데이터가 암호화됨 ① (외부 메모리로부터)
판독된 데이터가 해독됨 ④
암호화 키를 이용하여 해독 ③
판독/기입 핸들러
14
파일에 대응하는 암호화키 ②
파일 1에 대한 암호화키
파일 2에 대한 암호화키
파일 3에 대한 암호화키
파일 n에 대한 암호화키
외부 메모리

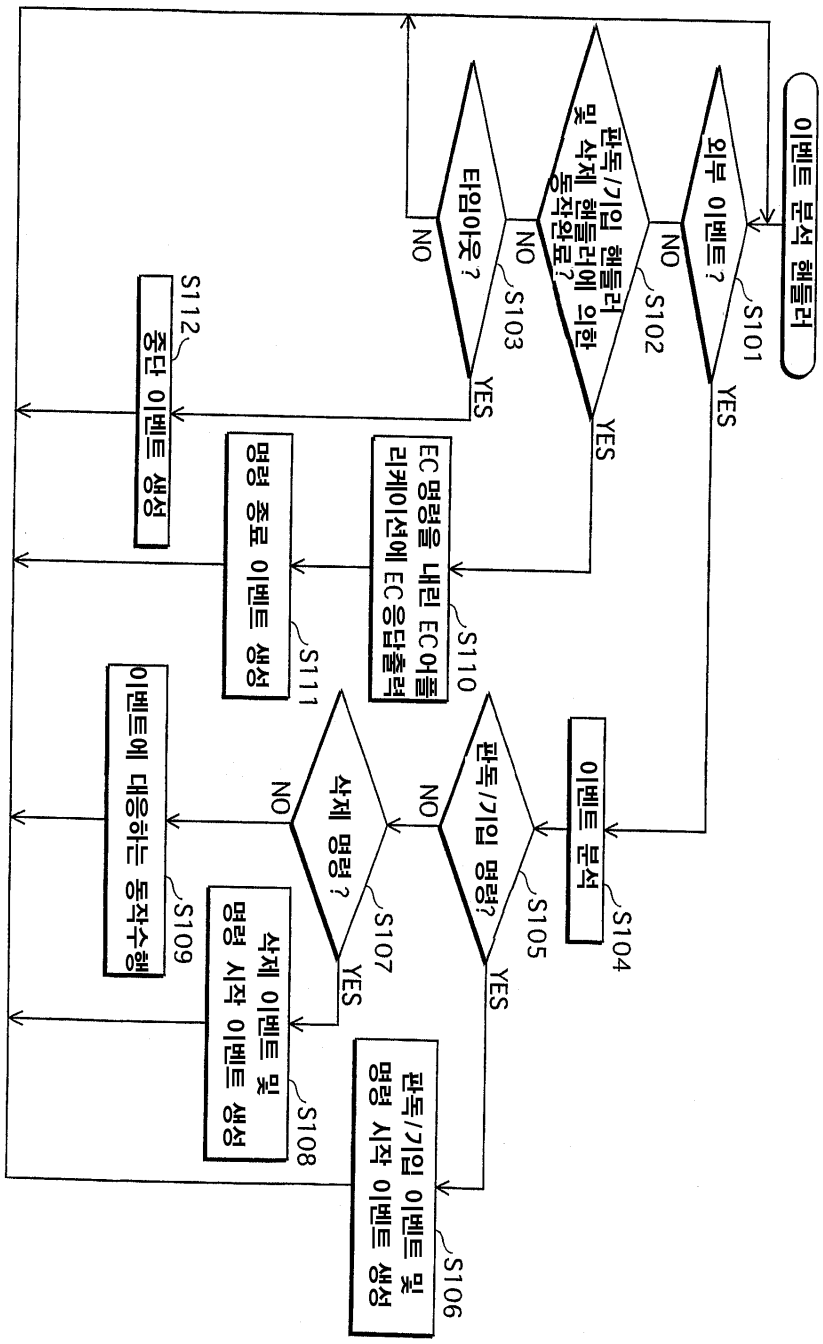
도면13


C_APL 1
C_APL 2
C_APL 3
C_APL 4
8
소거동작 관리테이블
유효플래그
시작 ADR
소거된 ADR
종료 ADR
테이블 생성
cw1
이벤트 분석 핸들러
11
삭제 이벤트
완료
중단 이벤트
14
판독/기입 이벤트
완료
명령 시작/종료 이벤트
완료
12
타임아웃 이벤트
명령시작
13
10
16
삭제 핸들러
판독/기입 핸들러
비명령 실행 핸들러
타이머 핸들러
널소거파일 엔트리 및 FAT
널소거파일 엔티티
nc1
널소거파일 엔티티
nc2
외부 메모리 제어부
외부 플래시 메모리
2
4
내부 EEPROM
3

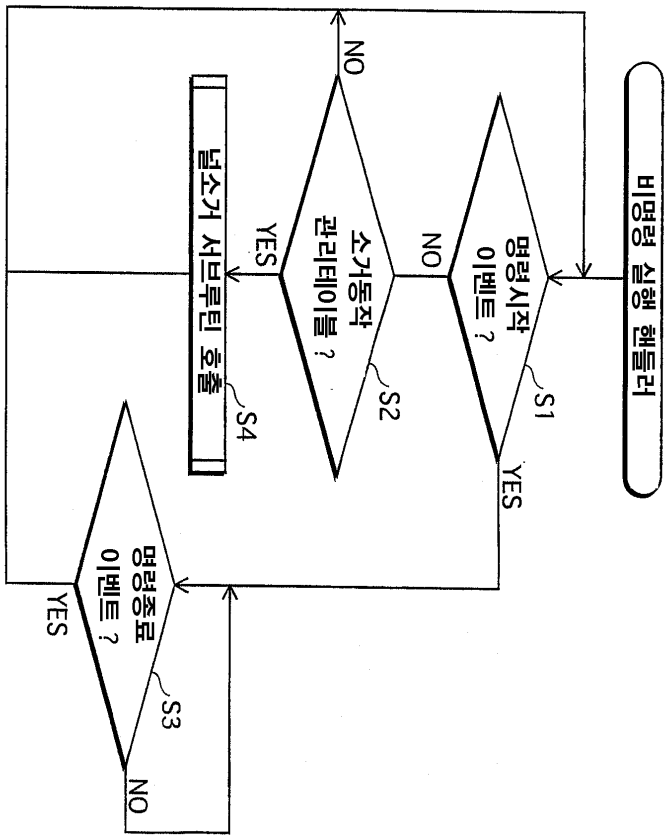
도면14


소거동작 관리테이블
레코드 1
레코드 2
유효플래그
시작 ADR
소거된 ADR
종료 ADR
유효플래그
시작 ADR
소거된 ADR
종료 ADR
소거중 :1
소거완료 :0
보안영역
소거된 영역
비소거된 영역
소거동작을 수행하는 방향 (시작 ADR 로부터)
소거동작이 수행될 때마다 갱신됨

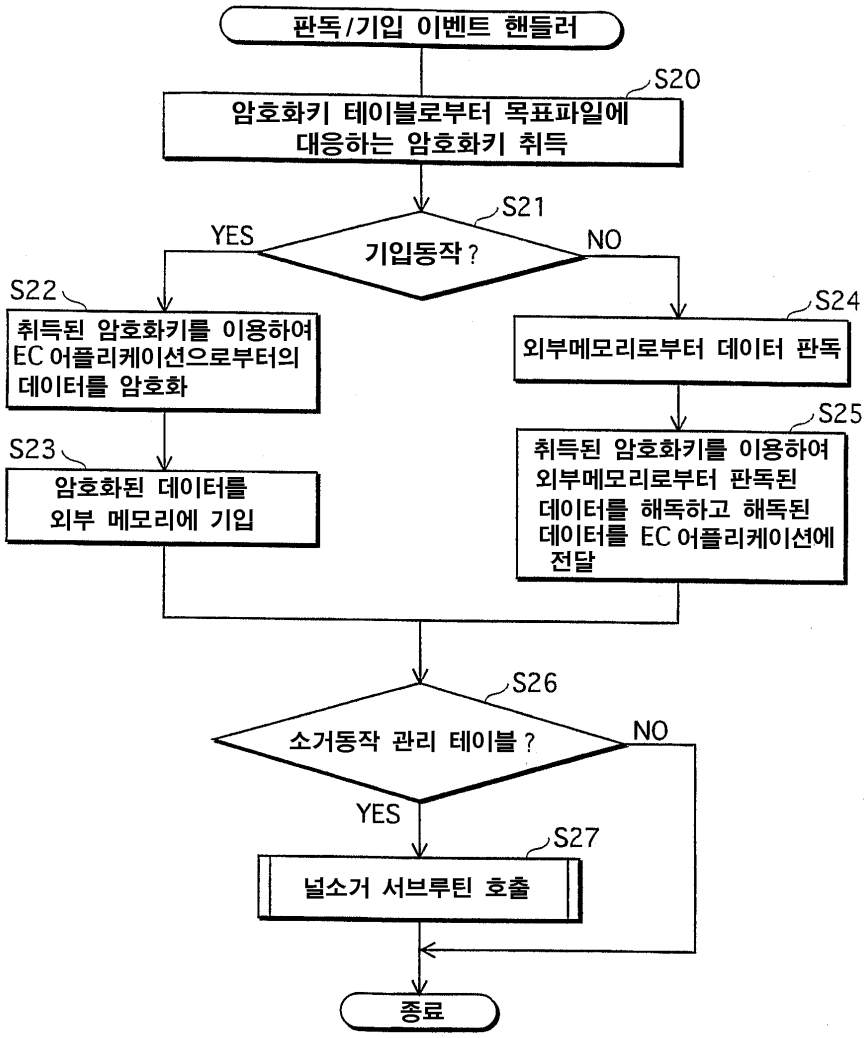
도면15


이벤트 분석 핸들러
외부 이벤트?
S101
YES
NO
판독/기입 핸들러 및 삭제 핸들러에 의한 동작완료?
S102
YES
NO
타임아웃?
S103
YES
NO
이벤트 분석
S104
판독/기입 명령?
S105
YES
NO
판독/기입 이벤트 및 명령 시작 이벤트 생성
S106
삭제 명령?
S107
YES
NO
삭제 이벤트 및 명령 시작 이벤트 생성
S108
이벤트에 대응하는 동작수행
S109
EC 명령을 내린 EC어플리케이션에 EC응답출력
S110
명령 종료 이벤트 생성
S111
중단 이벤트 생성
S112

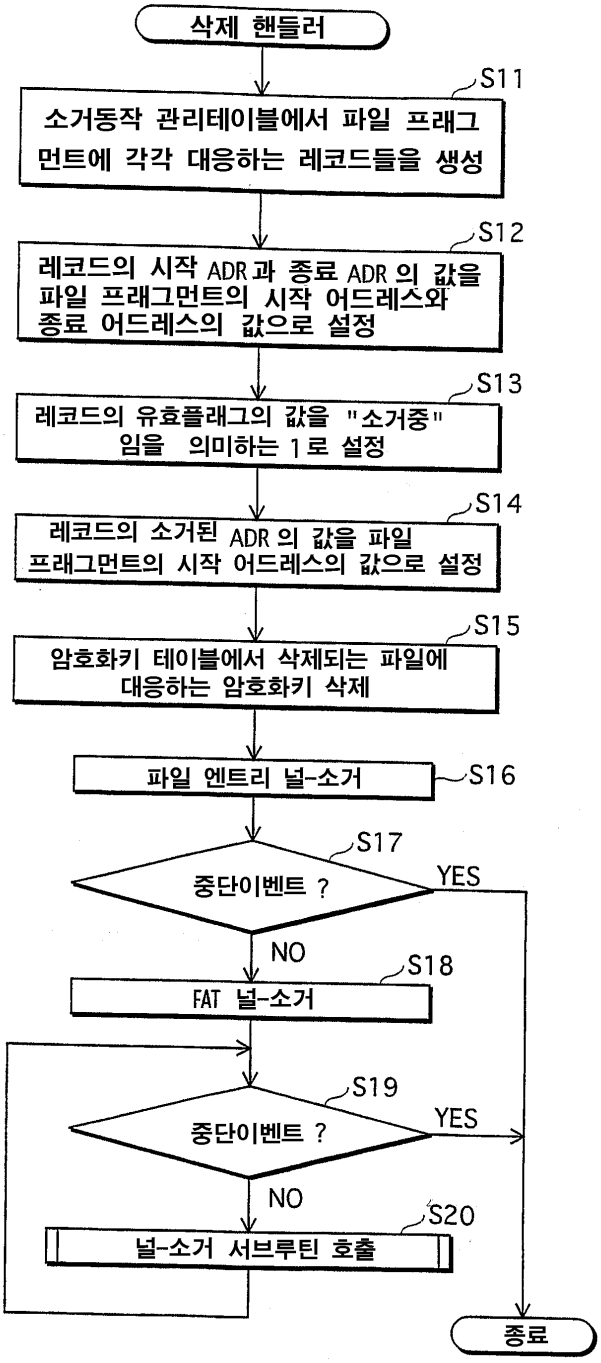
도면16


비명령 실행 핸들러
명령시작 이벤트 ?
S1
NO
YES
소거동작 관리테이블 ?
S2
NO
YES
널소거 서브루틴 호출
S4
명령종료 이벤트 ?
S3
YES
NO

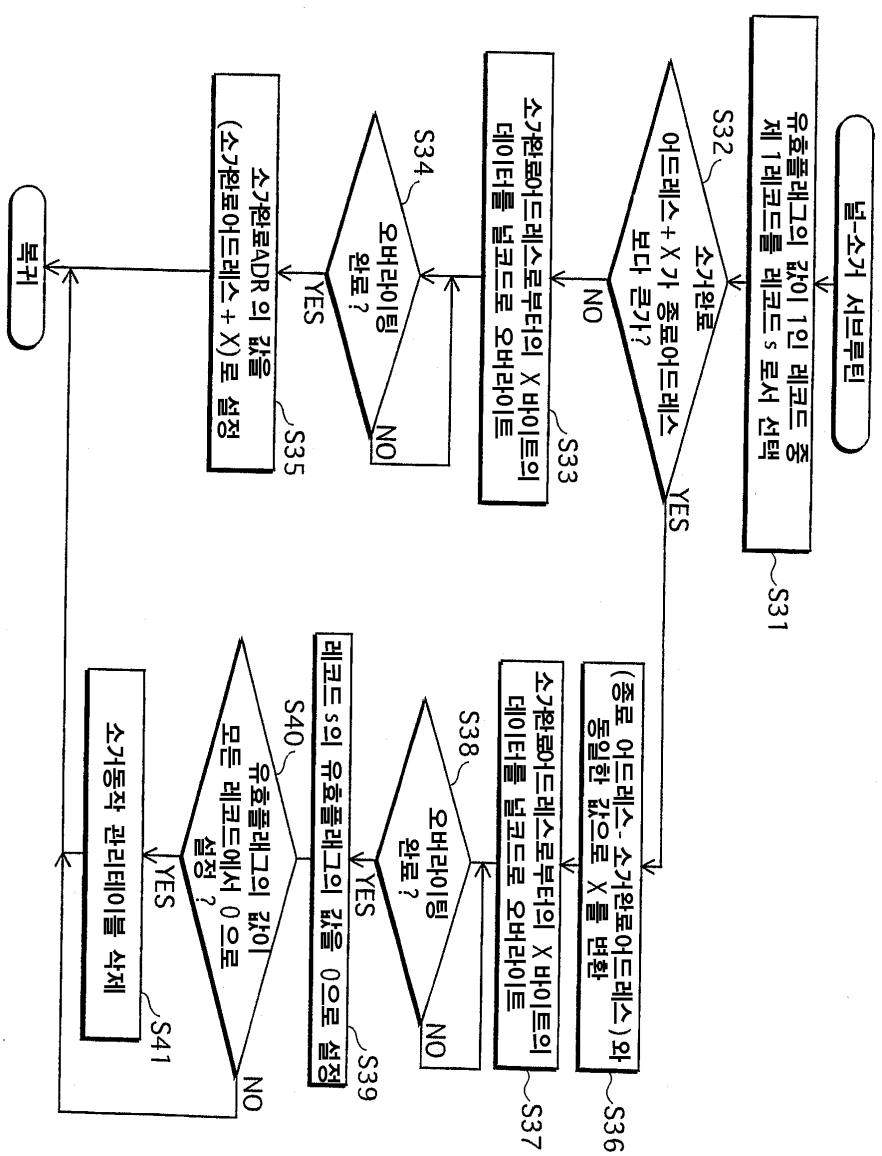
도면17


판독/기입 이벤트 핸들러
S20
암호화키 테이블로부터 목표파일에 대응하는 암호화키 취득
S21
기입동작?
YES
NO
S22
취득된 암호화키를 이용하여 EC 어플리케이션으로부터의 데이터를 암호화
S23
암호화된 데이터를 외부 메모리에 기입
S24
외부메모리로부터 데이터 판독
S25
취득된 암호화키를 이용하여 외부메모리로부터 판독된 데이터를 해독하고 해독된 데이터를 EC 어플리케이션에 전달
S26
소거동작 관리 테이블?
NO
YES
S27
널소거 서브루틴 호출
종료

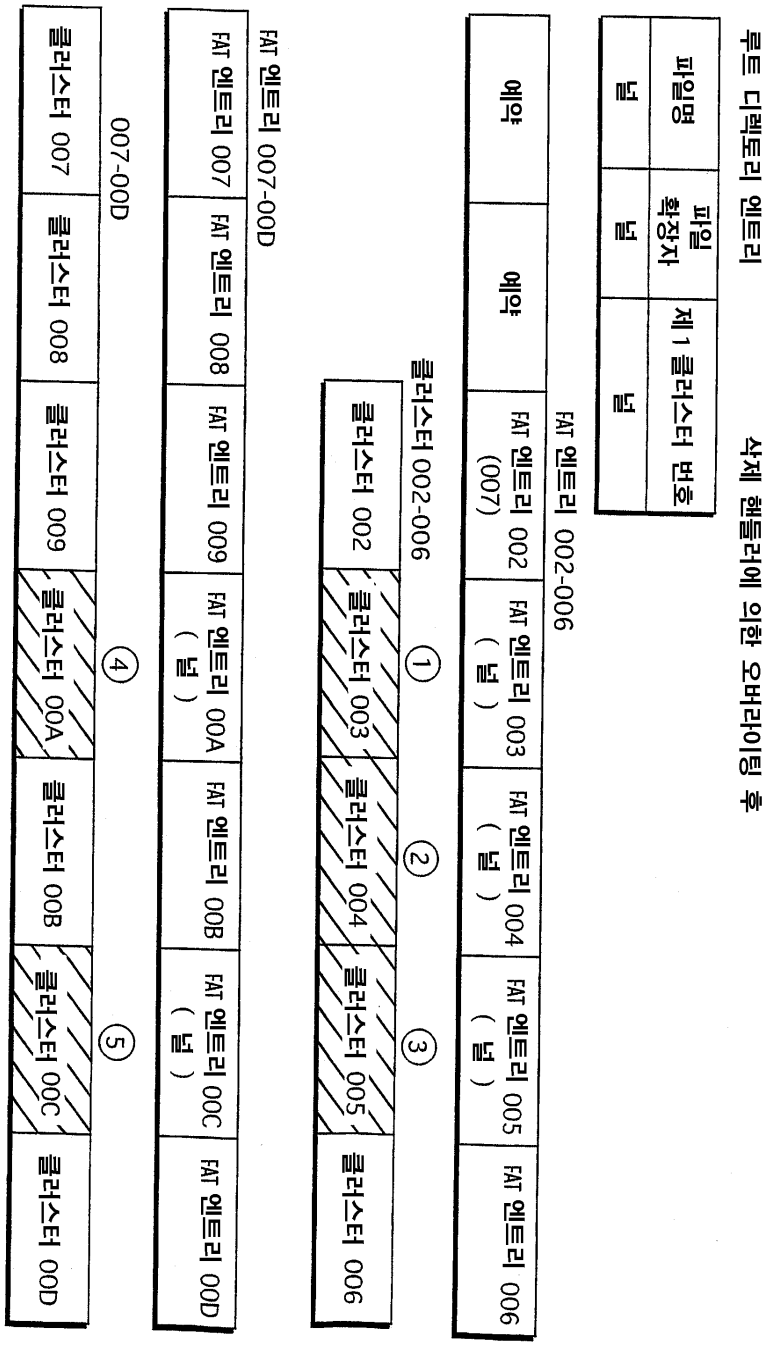
도면18


삭제 핸들러
S11
소거동작 관리테이블에서 파일 프래그먼트에 각각 대응하는 레코드들을 생성
S12
레코드의 시작 ADR 과 종료 ADR 의 값을 파일 프래그먼트의 시작 어드레스와 종료 어드레스의 값으로 설정
S13
레코드의 유효플래그의 값을 "소거중"임을 의미하는 1로 설정
S14
레코드의 소거된 ADR 의 값을 파일 프래그먼트의 시작 어드레스의 값으로 설정
S15
암호화키 테이블에서 삭제되는 파일에 대응하는 암호화키 삭제
파일 엔트리 널-소거
S16
S17
중단이벤트 ?
YES
NO
S18
FAT 널-소거
S19
중단이벤트 ?
YES
NO
S20
널-소거 서브루틴 호출
종료

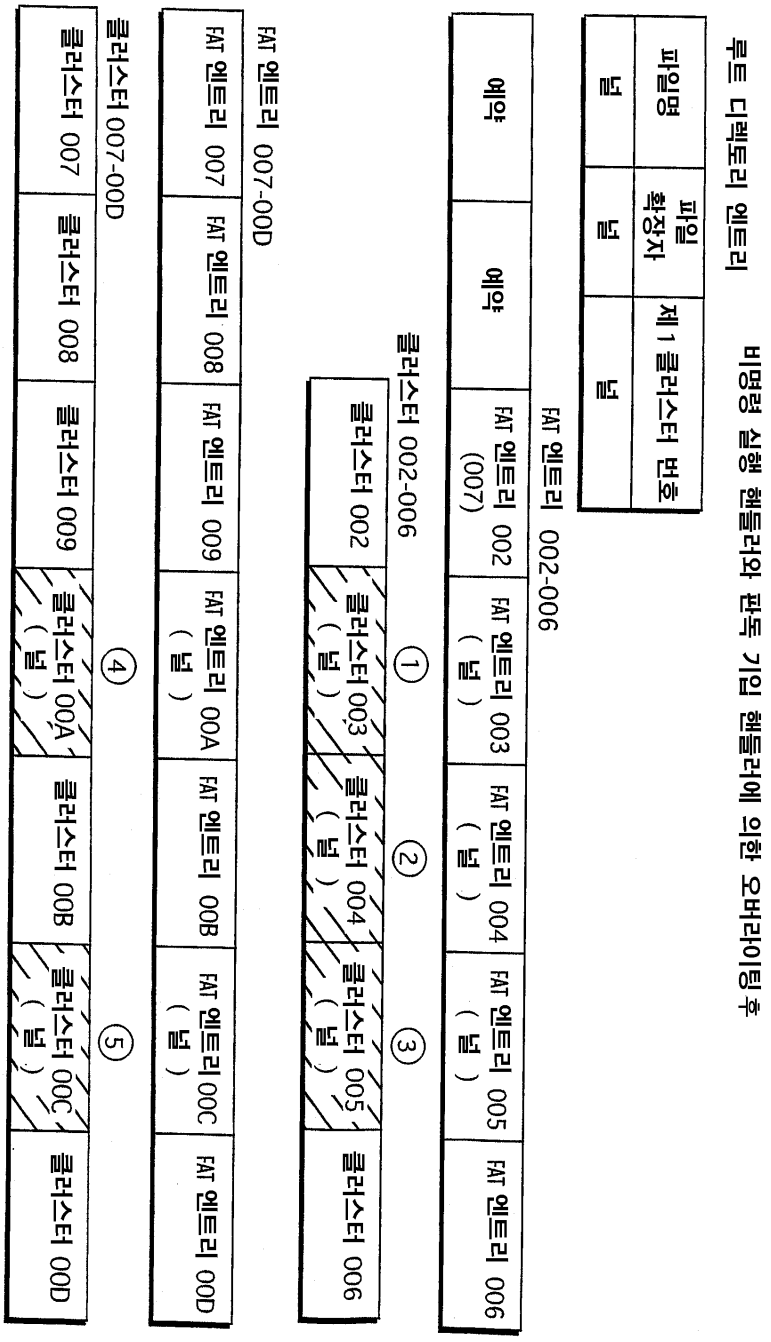
도면19


널-소거 서브루틴
유효플래그의 값이 1인 레코드 중 제 1레코드를 레코드 s 로서 선택
S31
어드레스 + X 가 종료어드레스 보다 큰가?
S32
NO
YES
소거완료
소거완료어드레스로부터의 X 바이트의 데이터를 널코드로 오버라이트
S33
오버라이팅 완료 ?
S34
NO
YES
소거완료ADR 의 값을 (소거완료어드레스 + X)로 설정
S35
복귀
(종료 어드레스- 소거완료어드레스)와 동일한 값으로 X 를 변환
S36
소거완료어드레스로부터의 X 바이트의 데이터를 널코드로 오버라이트
S37
오버라이팅 완료 ?
S38
NO
YES
레코드 s의 유효플래그의 값을 0으로 설정
S39
모든 레코드에서 유효플래그의 값이 0 으로 설정 ?
S40
NO
YES
소거동작 관리테이블 삭제
S41

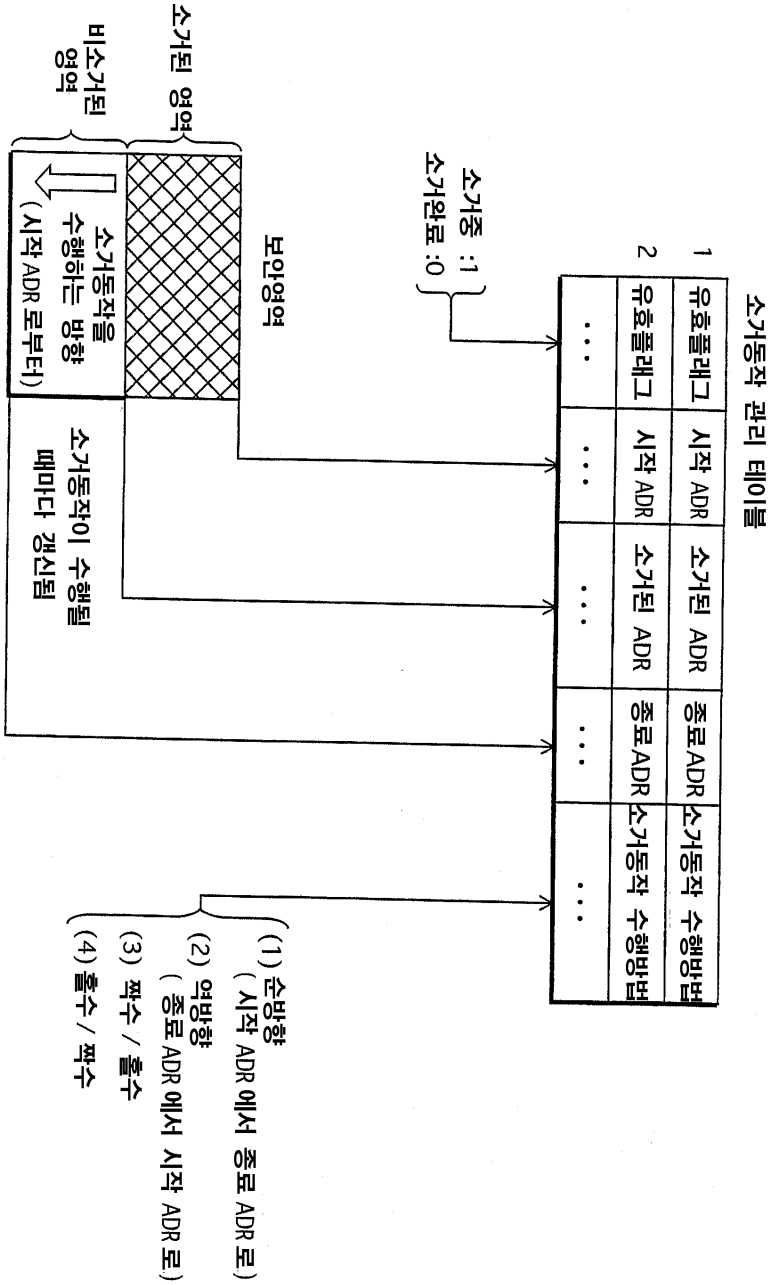
도면20


루트 디렉토리 엔트리
삭제 핸들러에 의한 오버라이팅 후
파일명
널
파일 확장자
널
제 1 클러스터 번호
널
FAT 엔트리 002-006
예약
예약
FAT 엔트리 002 (007)
FAT 엔트리 003 (널)
FAT 엔트리 004 (널)
FAT 엔트리 005 (널)
FAT 엔트리 006
클러스터 002-006
①
②
③
클러스터 002
클러스터 003
클러스터 004
클러스터 005
클러스터 006
FAT 엔트리 007-00D
FAT 엔트리 007
FAT 엔트리 008
FAT 엔트리 009
FAT 엔트리 00A (널)
FAT 엔트리 00B
FAT 엔트리 00C (널)
FAT 엔트리 00D
007-00D
④
⑤
클러스터 007
클러스터 008
클러스터 009
클러스터 00A
클러스터 00B
클러스터 00C
클러스터 00D

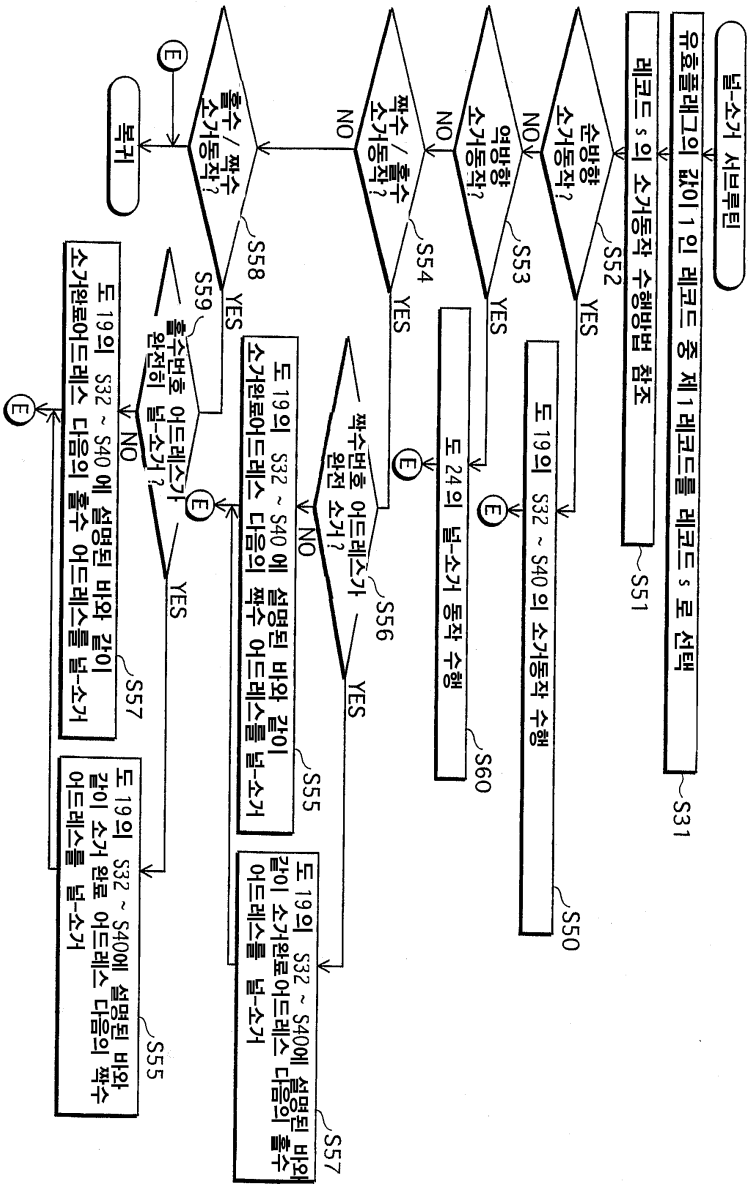
도면21


루트 디렉토리 엔트리
비명령 실행 핸들러와 판독 기입 핸들러에 의한 오버라이팅 후
파일명
파일 확장자
제 1 클러스터 번호
널
널
널
FAT 엔트리 002-006
예약
예약
FAT 엔트리 002 (007)
FAT 엔트리 003 (널)
FAT 엔트리 004 (널)
FAT 엔트리 005 (널)
FAT 엔트리 006
클러스터 002-006
①
②
③
클러스터 002
클러스터 003 (널)
클러스터 004 (널)
클러스터 005 (널)
클러스터 006
FAT 엔트리 007-00D
FAT 엔트리 007
FAT 엔트리 008
FAT 엔트리 009
FAT 엔트리 00A (널)
FAT 엔트리 00B
FAT 엔트리 00C (널)
FAT 엔트리 00D
클러스터 007-00D
④
⑤
클러스터 007
클러스터 008
클러스터 009
클러스터 00A (널)
클러스터 00B
클러스터 00C (널)
클러스터 00D

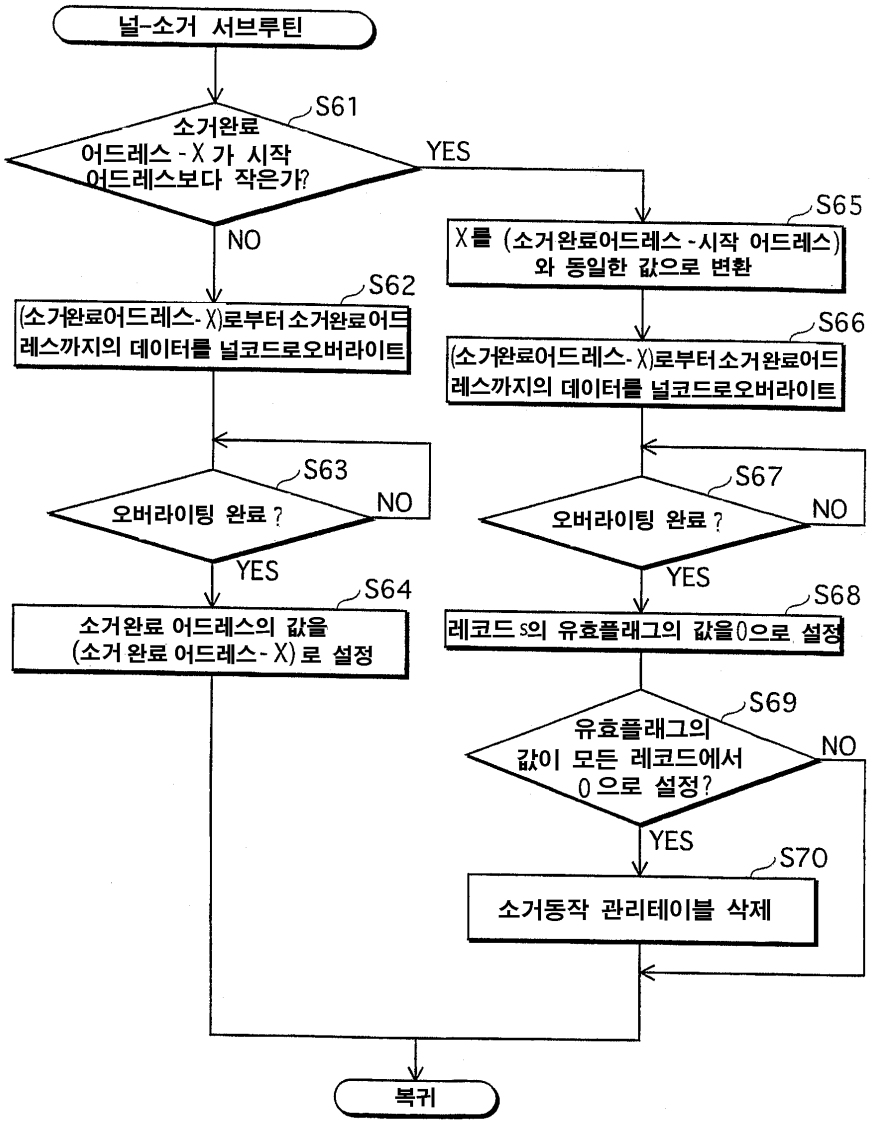
도면22


소거동작 관리 테이블
1
2
유효플래그
시작 ADR
소거된 ADR
종료ADR
소거동작 수행방법
유효플래그
시작 ADR
소거된 ADR
종료ADR
소거동작 수행방법
소거중 :1
소거완료 :0
보안영역
소거된 영역
비소거된 영역
소거동작을 수행하는 방향 (시작 ADR 로부터)
소거동작이 수행될 때마다 갱신됨
(1) 순방향 (시작 ADR 에서 종료 ADR 로)
(2) 역방향 (종료 ADR 에서 시작 ADR 로)
(3) 짝수 / 홀수
(4) 홀수 / 짝수

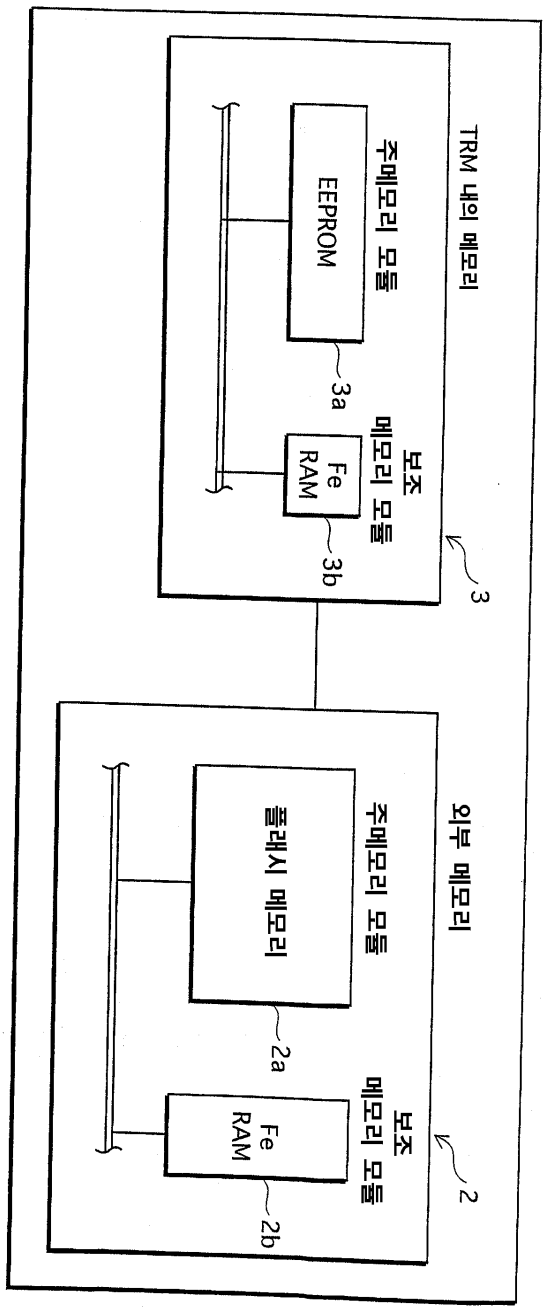
도면23


널-소거 서브루틴
유효플래그의 값이 1 인 레코드 중 제 1 레코드를 레코드 s 로 선택
S31
레코드 s 의 소거동작 수행방법 참조
S51
순방향 소거동작?
S52
YES
NO
도 19의 S32 ~ S40 의 소거동작 수행
S50
역방향 소거동작?
S53
YES
NO
도 24의 널-소거 동작 수행
S60
짝수 / 홀수 소거동작?
S54
YES
NO
짝수번호 어드레스가 완전 소거?
S56
YES
NO
도 19의 S32 ~ S40에 설명된 바와 같이 소거완료어드레스 다음의 홀수 어드레스를 널-소거
S57
도 19의 S32 ~ S40에 설명된 바와 같이 소거완료어드레스 다음의 짝수 어드레스를 널-소거
S55
홀수 / 짝수 소거동작?
S58
YES
NO
홀수번호 어드레스가 완전히 널-소거?
S59
YES
NO
도 19의 S32 ~ S40 에 설명된 바와 같이 소거완료어드레스 다음의 홀수 어드레스를 널-소거
S57
도 19의 S32 ~ S40에 설명된 바와 같이 소거 완료 어드레스 다음의 짝수 어드레스를 널-소거
S55
복귀
E

도면24


널-소거 서브루틴
S61
소거완료 어드레스-X가 시작 어드레스보다 작은가?
YES
NO
S65
X를 (소거완료어드레스-시작 어드레스)와 동일한 값으로 변환
S62
(소거완료어드레스-X)로부터 소거완료어드레스까지의 데이터를 널코드로오버라이트
S66
(소거완료어드레스-X)로부터 소거완료어드레스까지의 데이터를 널코드로오버라이트
S63
오버라이팅 완료?
NO
YES
S67
오버라이팅 완료?
NO
YES
S64
소거완료 어드레스의 값을 (소거 완료 어드레스-X)로 설정
S68
레코드s의 유효플래그의 값을0으로 설정
S69
유효플래그의 값이 모든 레코드에서 0으로 설정?
NO
YES
S70
소거동작 관리테이블 삭제
복귀

도면25

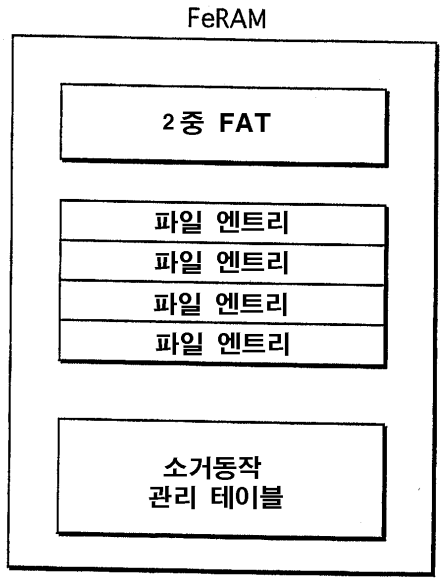

TRM 내의 메모리
주메모리 모듈
EEPROM
3a
보조
메모리 모듈
Fe
RAM
3b
3
외부 메모리
주메모리 모듈
플래시 메모리
2a
보조
메모리 모듈
Fe
RAM
2b
2

도면26

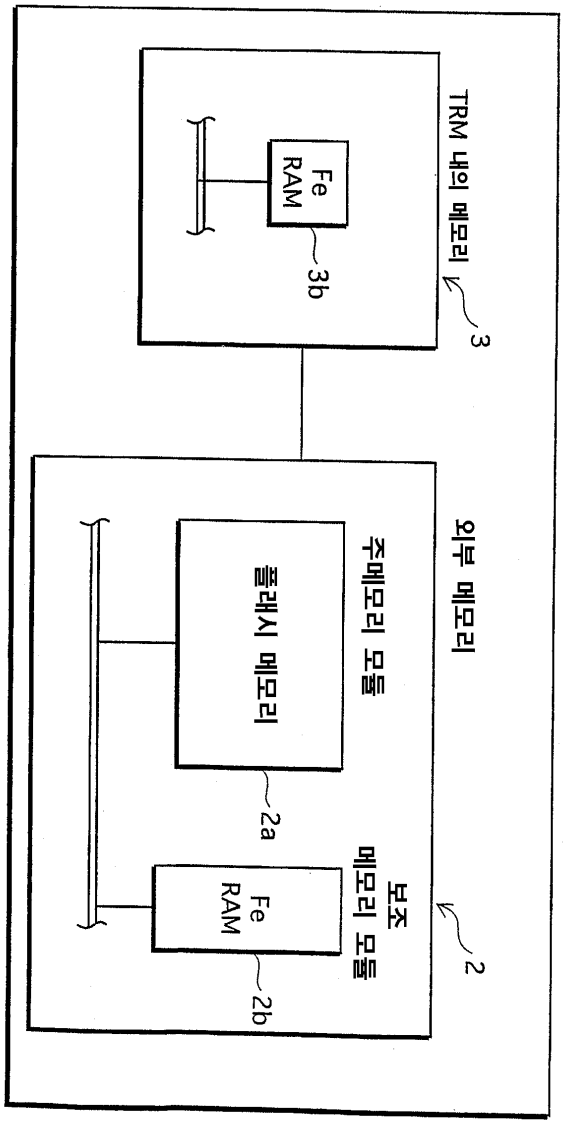
성능비교

	가격	대용량 데이터저장	기입단위	기입에 필요한 시간	가능한 기입 동작의 횟수	기입성능	판독에 필요한 시간	판독방법	전력소비
플래시	저	○	블록 ※1	10000ns	10^6	불안정 ※2	50ns	비파괴	○
FeRAM	고	△	1바이트 ※3	30~100ns	$10^{12\sim16}$	안정	30~100ns	파괴 ※4	◎

도면27



도면28


TRM 내의 메모리
3
Fe RAM
3b
외부 메모리
2
주메모리 모듈
플래시 메모리
2a
보조 메모리 모듈
Fe RAM
2b