



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년07월02일
 (11) 등록번호 10-0843136
 (24) 등록일자 2008년06월26일

(51) Int. Cl.

G11C 16/10 (2006.01) G11C 16/34 (2006.01)

(21) 출원번호 10-2006-0112411

(22) 출원일자 2006년11월14일

심사청구일자 2006년11월14일

(65) 공개번호 10-2008-0043633

(43) 공개일자 2008년05월19일

(56) 선행기술조사문헌

JP09251783 A

JP2004206855 A

KR1019980060858 A

KR1020010084392 A

전체 청구항 수 : 총 10 항

(73) 특허권자

삼성전자주식회사

경기도 수원시 영통구 매탄동 416

(72) 발명자

인지현

서울 송파구 잠실동 213-36(B01)

김진규

서울 광진구 중곡1동 615-1

(뒷면에 계속)

(74) 대리인

정상빈, 특허법인가산

심사관 : 조명관

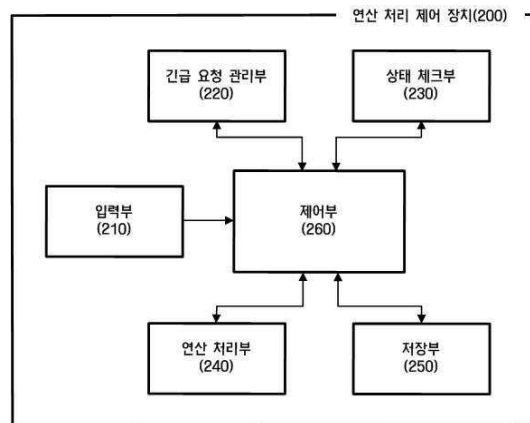
(54) 비휘발성 메모리에서 연산 처리를 제어하는 장치 및 그방법

(57) 요약

본 발명은 비휘발성 메모리에서 연산 처리를 제어하는 장치 및 그 방법에 관한 것으로서, 더욱 상세하게는 일반 요청에 대한 연산 처리 중 긴급 요청에 대한 연산 처리가 발생되면, 일반 요청에 대한 연산 처리를 중단한 후, 긴급 요청에 대한 연산 처리를 먼저 수행하는 비휘발성 메모리에서 연산 처리를 제어하는 장치 및 그 방법에 관한 것이다.

본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 장치는, 긴급 요청에 대한 연산 발생 시 프리엠프트 플래그 및 상태 백업 플래그 값을 설정하는 긴급 요청 관리부와, 상기 설정된 프리엠프트 플래그 및 상태 백업 플래그의 설정값을 체크하는 상태 체크부와, 상기 발생된 긴급 요청에 대한 연산을 처리하고, 상기 체크된 설정값을 기초로 일반 요청에 대한 연산을 처리하는 연산 처리부를 포함한다.

대표도 - 도3



(72) 발명자

신일훈

경기 수원시 영통구 영통동 황갈마을1단지아파트
151동 702호

나세욱

서울 용산구 용산동2가 18-5

김효준

경기 용인시 기흥구 상갈동 금화마을주공아파트
407동 1204호

윤송호

경기 용인시 수지구 풍덕천2동 삼성5차아파트
523-702

신동균

서울 관악구 봉천6동 1681-21

특허청구의 범위

청구항 1

긴급 요청에 대한 연산 발생 시 프리엠프트 플래그 및 상태 백업 플래그 값을 설정하는 긴급 요청 관리부;
 상기 설정된 프리엠프트 플래그 및 상태 백업 플래그의 설정값을 체크하는 상태 체크부; 및
 상기 발생된 긴급 요청에 대한 연산을 처리하고, 상기 체크된 설정값을 기초로 일반 요청에 대한 연산을 처리하는 연산 처리부를 포함하는, 비휘발성 메모리에서 연산 처리를 제어하는 장치.

청구항 2

제 1항에 있어서,
 상기 프리엠프트 플래그에 설정된 값을 통해 긴급 요청에 대한 연산 처리가 수행되었는지 여부를 확인할 수 있으며, 상기 상태 백업 플래그에 설정된 값을 통해 일반 요청에 대한 연산 처리 결과값의 저장 여부를 확인할 수 있는, 비휘발성 메모리에서 연산 처리를 제어하는 장치.

청구항 3

제 1항에 있어서,
 상기 연산 처리부는 긴급 요청에 대한 연산 처리가 발생되면 일반 요청에 대한 연산 처리를 중단한 후, 긴급 요청에 대한 연산 처리를 수행하는, 비휘발성 메모리에서 연산 처리를 제어하는 장치.

청구항 4

제 1항에 있어서,
 상기 일반 요청에 대한 연산 처리 결과값을 저장하는 저장부를 더 포함하는, 비휘발성 메모리에서 연산 처리를 제어하는 장치.

청구항 5

연산 처리를 제어하는 장치가 비휘발성 메모리에서 연산 처리를 제어하는 방법에 있어서,
 긴급 요청에 대한 연산 처리가 발생되면, 일반 요청에 대한 연산 처리의 완료를 대기하는 단계;
 상기 연산 처리 완료 후 상태 백업 플래그의 값을 설정하는 단계;
 상기 긴급 요청에 대한 연산을 처리하는 단계; 및
 상기 긴급 요청에 대한 연산 처리가 완료되면, 프리엠프트 플래그 값을 설정하는 단계를 포함하는, 비휘발성 메모리에서 연산 처리를 제어하는 방법.

청구항 6

제 5항에 있어서,
 상기 연산 처리 완료 후 연산 처리의 결과 값을 저장하는 단계를 더 포함하는, 비휘발성 메모리에서 연산 처리를 제어하는 방법.

청구항 7

제 5항에 있어서,
 상기 프리엠프트 플래그에 설정된 값을 통해 긴급 요청에 대한 연산 처리가 수행되었는지 여부를 확인할 수 있으며, 상기 상태 백업 플래그에 설정된 값을 통해 일반 요청에 대한 연산 처리 결과값의 저장 여부를 확인할 수 있는, 비휘발성 메모리에서 연산 처리를 제어하는 방법.

청구항 8

제 5항에 있어서,

상기 상태 백업 플래그 및 프리엠프트 플래그 값이 모두 설정되어 있는 경우, 일반 요청에 대한 연산 처리의 결과를 확인하는, 비휘발성 메모리에서 연산 처리를 제어하는 방법.

청구항 9

제 5항에 있어서,

상기 프리엠프트 플래그 값을 체크하는 단계;

상기 체크 결과 프리엠프트 플래그 값이 설정된 경우, 상기 프리엠프트 플래그 값을 초기화하는 단계; 및

상기 일반 요청에 대한 연산을 처리하는 단계를 포함하는, 비휘발성 메모리에서 연산 처리를 제어하는 방법.

청구항 10

제 5항에 있어서,

상기 긴급 요청에 대한 연산 처리가 발생되면, 상기 일반 요청에 대한 연산 처리를 중단하는 단계;

상기 긴급 요청에 대한 연산을 처리하는 단계; 및

상기 긴급 요청에 대한 연산 처리 완료 후, 프리엠프트 플래그 값을 설정하는 단계를 포함하는, 비휘발성 메모리에서 연산 처리를 제어하는 방법.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <12> 본 발명은 비휘발성 메모리에서 연산 처리를 제어하는 장치 및 그 방법에 관한 것으로서, 더욱 상세하게는 일반 요청에 대한 연산 처리 중 긴급 요청에 대한 연산 처리가 발생되면, 일반 요청에 대한 연산 처리를 중단한 후, 긴급 요청에 대한 연산 처리를 먼저 수행하는 비휘발성 메모리에서 연산 처리를 제어하는 장치 및 그 방법에 관한 것이다.
- <13> 최근 모바일 기기의 기능이 다양화 및 복잡화되고 멀티미디어 기능이 강화됨으로 인해 모바일 기기에 탑재된 소프트웨어의 크기 및 사용자 데이터의 저장 공간의 요구량은 점점 더 커지고 있다. 이에 따라, 코드 및 사용자 데이터를 저장하기 위한 비휘발성 메모리의 크기 및 코드를 동작시키기 위한 휘발성 메모리의 크기에 대한 요구가 급격히 증가하고 있으며, 이것으로 인해 모바일 기기의 원가가 증가하고 있다.
- <14> 종래의 모바일 기기에서는 부팅 및 코드의 저장을 위한 비휘발성 메모리로 노어(NOR) 플래시 메모리가 주로 사용되었으며, 또한 사용자 데이터를 저장하기 위해 낸드(NAND) 플래시 메모리가 사용되고, 휘발성 메모리로 디램(DRAM)이 사용되었다. 하지만, 모바일 기기 제조사들은 메모리 구성에 대한 원가를 낮추고 성능을 향상시키기 위해 NAND 플래시 메모리에 코드와 사용자 데이터를 함께 저장하는 방식을 많이 채택하고 있다. NAND 플래시 메모리는 NOR 플래시 메모리와 달리 XIP(eXecute In Place)가 되지 않아 코드 수행을 위해서는 램(RAM)으로 모든 코드를 적재한 다음 코드 수행을 해야 한다. 그런데, 최근 RAM의 크기를 줄이기 위해 NAND 플래시 메모리를 보조 기억장치(Backing Store)로 하는 요구 페이징(Demand Paging) 기법을 이용한 가상 기억장치 시스템(Virtual Memory System)을 적용한 사례가 늘고 있다. 이는 모든 코드를 RAM으로 한꺼번에 적재하는 것이 아니라 코드가 액세스될 때, 만약 액세스된 페이지가 RAM에 존재하지 않으면 페이지 단위로 RAM으로 해당 페이지를 적재시키는 것을 말한다.
- <15> 또한, 액세스된 페이지가 RAM에 존재하지 않은 경우는 예외 경우(exception case)에 해당하며, 발생한 예외 경우를 처리하지 않으면, 시스템의 어떤 동작도 진행될 수 없다. 따라서, 액세스된 페이지를 NAND 플래시 메모리로부터 읽어서 RAM으로 적재시켜 코드 수행이 계속 진행될 수 있도록 해야 한다.
- <16> 그런데, NAND 플래시 메모리에는 코드뿐만 아니라 사용자 데이터도 함께 저장되어 있으므로, 예외 경우가 발생

해서 코드 페이지를 로드하려고 할 때, 이미 NAND 플래시 메모리는 사용자 데이터의 저장 및 추출 등의 목적으로 사용 중일 수 있다. 이러한 경우, 사용중인 연산을 멈추고 상기에서 설명한 대로 예외 경우 처리가 먼저 수행되어야만 한다. 따라서, 기존에 수행 중이던 읽기/쓰기/지우기 연산을 멈추고 예외 경우를 수행하기 위한 연산에 대한 처리가 먼저 선행될 필요가 있다.

- <17> 도 1은 일반적인 NAND 플래시 메모리에서의 읽기 및 쓰기 연산을 수행하는 과정을 나타낸 도면이다.
- <18> 도시된 바와 같이, 비휘발성 메모리(예를 들어, NAND 플래시 메모리)는 페이지 단위로 읽기 쓰기 연산을 수행하며, 읽기 명령을 보내면 해당 페이지를 페이지 레지스터(21)까지 로드한다. 이때, 로딩 동작이 수행되는 동안 NAND 플래시 메모리(20)의 상태는 바쁜(Busy) 상태이다. 그 다음, 페이지 레지스터(21)까지 데이터가 모두 로딩되면, 호스트 메모리(10)로 데이터를 전송할 수 있으며, 데이터가 전송되는 동안 NAND 플래시 메모리(20)는 준비(Ready) 상태로서 다음 명령을 받을 수 있는 상태가 되지만, 다음 명령이 들어오면 페이지 레지스터(21)에 로드된 데이터 값은 변경될 수 있다. 따라서, 읽기 연산의 완료는 페이지 레지스터(21)에서 호스트 메모리(10)의 데이터 전송이 완료될 때 까지라고 할 수 있다.
- <19> 반대로, 쓰기 연산은 호스트 메모리(10)로부터 쓰기를 수행할 데이터를 페이지 레지스터(21)로 전송한다. 이때, 페이지 레지스터(21)까지 데이터를 전송하는 동안 NAND 플래시 메모리(20)는 준비(Ready) 상태가 된다. 그 다음, 페이지 레지스터(21)까지 데이터 전송을 완료한 후, 쓰기 명령을 내리면 NAND 셀(Cell)(22)로 페이지 레지스터(21)의 데이터가 쓰여지게 된다. 여기서, 페이지 레지스터(21)의 데이터가 쓰여지는 동안 NAND 플래시 메모리(20)는 바쁜(Busy) 상태로서 다른 명령을 받을 수 없는 상태가 된다. 이 후, 쓰기가 완료되면 NAND 플래시 메모리(20)의 상태 레지스터(Status Register)에는 수행된 쓰기 연산에 대한 수행 결과가 저장된다.
- <20> 한국공개특허 2001-026600(플래시 메모리의 실시간 처리방법)은 읽기/쓰기 동작을 수행하기 위해 플래시 메모리의 소정 블록에 대해 접근 요청이 들어오면, 접근 가능 여부 및 지우기 동작 중인지 등의 여부를 판단하여, 지우기 동작을 중단한 후 읽기/쓰기 동작을 수행하도록 하는 기술을 개시하고 있으나, 이는 일반 요청에 대한 연산 수행 시 긴급 요청에 대한 연산 수행이 발생되면 긴급 요청에 대한 연산 처리를 먼저 수행하는 기술에 대해서 전혀 언급하고 있지 않다.

발명이 이루고자 하는 기술적 과제

- <21> 본 발명은 일반 요청에 대한 연산 처리 중 긴급 요청에 대한 연산 처리가 발생되면, 긴급 요청에 대한 연산 처리를 먼저 수행하는데 그 목적이 있다.
- <22> 본 발명의 목적들은 이상에서 언급한 목적들로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해되어질 수 있을 것이다.

발명의 구성 및 작용

- <23> 상기 목적을 달성하기 위하여, 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 장치는, 긴급 요청에 대한 연산 발생 시 프리엠프트 플래그 및 상태 백업 플래그 값을 설정하는 긴급 요청 관리부와, 상기 설정된 프리엠프트 플래그 및 상태 백업 플래그의 설정값을 체크하는 상태 체크부와, 상기 발생된 긴급 요청에 대한 연산을 처리하고, 상기 체크된 설정값을 기초로 일반 요청에 대한 연산을 처리하는 연산 처리부를 포함한다.
- <24> 또한, 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법은, 연산 처리를 제어하는 장치가 비휘발성 메모리에서 연산 처리를 제어하는 방법에 있어서, 긴급 요청에 대한 연산 처리가 발생되면, 일반 요청에 대한 연산 처리의 완료를 대기하는 단계와, 상기 연산 처리 완료 후 상태 백업 플래그의 값을 설정하는 단계와, 상기 긴급 요청에 대한 연산을 처리하는 단계와, 상기 긴급 요청에 대한 연산을 처리가 완료되면, 프리엠프트 플래그 값을 설정하는 단계를 포함한다.
- <25> 기타 실시예들의 구체적인 사항들은 상세한 설명 및 도면들에 포함되어 있다.
- <26> 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한

다.

- <27> 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 상세히 설명하기로 한다.
- <28> 도 2는 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 장치에서 긴급 요청 및 일반 요청에 대한 연산 처리를 수행하는 과정을 나타낸 도면이다.
- <29> 도시된 바와 같이, 비휘발성 메모리를 구비한 저장 장치(100)는 디바이스 드라이버(200)와 플래시 메모리(300)를 포함할 수 있다. 디바이스 드라이버(200)는 일반 요청(Normal Request)과 긴급 요청(Urgent Request)에 대한 연산을 처리하는 것으로, 긴급 요청에 대한 연산 처리가 발생되면 일반 요청에 대한 연산 처리는 중단되고, 긴급 요청에 대한 연산 처리가 수행된다. 여기서, 디바이스 드라이버(200)는 이하 도면에서 연산 처리 제어 장치로 설명될 것이다. 플래시 메모리(300)는 디바이스 드라이버(200)에서 처리된 데이터를 저장하는 것으로, 예를 들어 NAND 플래시 메모리를 말한다.
- <30> 일반 요청은 동기화를 위해 세마포어(semaphores) 등으로 보호함으로써, 하나의 일반 요청이 끝날 때까지 다른 일반 요청들은 기다리도록 한다. 그러나, 긴급 요청은 일반 요청이 진행 중이더라도 이를 중단하고 읽기/쓰기/지우기 연산을 처리할 수 있도록 한다. 따라서, 실시간 애플리케이션을 위한 코드 액세스와 같이 긴 지연(latency)을 허용할 수 없는 요구에 대한 빠른 대응이 가능하다.
- <31> 또한, 긴급 요청 처리 루틴은 언제든지 일반 요청 처리 루틴이 동작하는 중에 끼어들어 일반 요청의 연산을 중단시킬 수 있다. 따라서, 일반 요청 처리 루틴은 연산 처리 중 긴급 요청에 의해 연산 처리가 중단된 적이 있는지 확인해서 적절한 조치를 취할 수 있어야 한다. 여기서, 긴급 요청에 의해 중단된 적이 있는지 여부는 프리엠프트 플래그(Preempted Flag)를 통해 나타낼 수 있다.
- <32> 예를 들어, 일반 요청 처리 루틴은 일반 요청을 처리하기 전에 항상 프리엠프트 플래그를 '0'으로 초기화한 후, 일반 요청에 대한 연산을 처리한다. 또한, 긴급 요청 처리 루틴은 긴급 요청을 처리하면서 항상 프리엠프트 플래그를 '1'로 설정한다. 이 후, 일반 요청 처리 루틴은 프리엠프트 플래그를 확인함으로써, 일반 읽기/쓰기/지우기 요청에 대한 연산 처리를 다시 수행할지 여부를 결정한다. 만약 비휘발성 메모리를 구비한 저장 장치(100)가 중지(Suspend) 또는 재 시작(Resume) 명령을 지원한다면, 쓰거나 지우기 연산을 처음부터 다시 수행하지 않고 재 시작 명령을 통해 쓰거나 지우기 연산을 계속해서 진행하게 할 수도 있다.
- <33> 한편, 일반 요청 처리를 위한 비휘발성 메모리를 구비한 저장 장치(100)의 연산 중에 긴급 요청이 끼여 들어 연산을 멈추고, 긴급 요청을 처리하는 경우도 있지만, 일반 요청에 대한 연산이 완료된 다음 긴급 요청에 대한 연산이 발생하는 경우도 있을 수 있다. 이 경우, 비휘발성 메모리를 구비한 저장 장치(100)는 준비 상태지만, 일반 요청 처리 루틴이 아직 상태 레지스터(Status Register) 값을 확인하지 않은 상태이다. 이에, 긴급 요청 처리 루틴에서 읽기/쓰기/지우기 연산을 수행하면, 비휘발성 메모리를 구비한 저장 장치(100)의 상태 레지스터에는 긴급 요청 처리에 대한 결과가 반영되므로, 일반 요청 처리 루틴에서는 이전에 수행했던 일반 요청에 대한 연산 결과를 알 수 없다는 문제점이 발생 된다. 따라서, 긴급 요청 처리 루틴에서는 상기와 같은 경우 상태 백업(Backup Status)이라는 변수에 이전에 진행되었던 연산에 대한 결과를 저장해두고, 상태 백업 플래그(Status-Backup Flag)를 '1'로 설정한다. 이 후, 일반 요청 처리 루틴에서는 프리엠프트 플래그가 '1'로 설정되어 있으면서 상태 백업 플래그도 '1'로 설정되어 있어 있는 경우, 저장되어 있는 상태 백업을 확인함으로써 이전에 수행시킨 연산 처리 결과를 확인할 수 있다.
- <34> 또한, 긴급 요청 처리 루틴은 진행중인 연산의 동작(operation) 종류에 따라서 다른 동작을 수행한다. 따라서, 긴급 요청 처리 루틴은 진행 중인 일반 요청의 연산이 어떤 종류였는지 알 수 있어야 한다. 플래시 메모리(300)는 그 종류에 따라 상태 레지스터를 통해 진행 중인 동작의 종류를 나타내는 것도 있으나, 하드웨어적으로 지원하지 않는 메모리도 있다. 이런 경우, 동작 타입 플래그(Operation Type Flag)를 두어 일반 요청에 대한 연산을 수행할 때 해당하는 연산 타입을 설정하여, 나중에 긴급 요청 처리 루틴에서 확인할 수 있도록 할 수 있다.
- <35> 또한, 플래시 메모리(300)는 그 종류에 따라, 진행중인 플래시 입/출력(I/O)을 중단하고 새로운 플래시 입/출력을 수행하기 위해서 비휘발성 메모리를 구비한 저장 장치(100)를 리셋(reset) 시켜야 하는 장치가 있다. 따라서, 상기와 같은 장치를 사용하는 경우, 긴급 요청 처리 루틴은 진행 중인 플래시 입/출력을 멈추고 새로운 플래시 입/출력을 수행하는 경우, 비휘발성 메모리를 구비한 저장 장치(100)에게 리셋 명령을 보내고 긴급 요청에 대한 연산을 처리한다.

- <36> 도 3은 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 장치를 나타낸 블록도이다.
- <37> 도시된 바와 같이, 연산 처리 제어 장치(200)는 입력부(210), 긴급 요청 관리부(220), 상태 체크부(230), 연산 처리부(240), 저장부(250) 및 제어부(260)를 포함하여 구성된다.
- <38> 이때, 본 실시예에서 사용되는 '~부'라는 용어는 소프트웨어 또는 FPGA 또는 ASIC과 같은 하드웨어 구성요소를 의미하며, '~부'는 어떤 역할들을 수행한다. 그렇지만 '~부'는 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. '~부'는 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다. 따라서, 일 예로서 '~부'는 소프트웨어 구성요소들, 객체지향 소프트웨어 구성요소들, 클래스 구성요소들 및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들, 및 변수들을 포함한다. 구성요소들과 '~부'들 안에서 제공되는 기능은 더 작은 수의 구성요소들 및 '~부'들로 결합되거나 추가적인 구성요소들과 '~부'들로 더 분리될 수 있다.
- <39> 입력부(210)는 사용자 입력부로부터 요청된 일반 요청에 대한 연산 처리 및 긴급 요청에 대한 연산 처리의 명령을 입력받는다. 여기서, 사용자 입력부는 호스트 메모리 및 어플리케이션으로 이해될 수 있다.
- <40> 긴급 요청 관리부(220)는 프리엠프트 플래그 및 상태 백업 플래그 값을 설정한다.
- <41> 예를 들어, 긴급 요청에 대한 연산 처리가 일반 요청에 대한 연산 처리 중에 발생되어 일반 요청에 대한 연산 처리가 중단되는 경우, 긴급 요청 관리부(220)는 프리엠프트 플래그를 '1'로 설정한다. 여기서, 일반 요청에 대한 연산 처리 시 프리엠프트 플래그를 '0'으로 초기화되어 있다.
- <42> 또한, 긴급 요청에 대한 연산 처리가 일반 요청에 대한 연산 처리가 완료된 다음에 발생되어 일반 요청에 대한 연산 처리 결과를 확인하지 못한 경우, 긴급 요청 관리부(220)는 제어부(260)에게 일반 요청에 대한 연산 처리 결과를 저장하도록 요청한다. 이에, 저장부(250)에 일반 요청에 대한 연산 처리 결과가 저장되면, 긴급 요청 관리부(220)는 일반 요청에 대한 연산 처리 결과가 저장되어 있음을 나타내기 위해 상태 백업 플래그를 '1'로 설정한다.
- <43> 상태 체크부(230)는 긴급 요청 관리부(220)를 통해 설정된 프리엠프트 플래그 및 상태 백업 플래그의 설정값을 체크하는 것으로, 프리엠프트 플래그에 설정된 값을 통해 긴급 요청에 대한 연산 처리가 수행되었는지 여부를 확인할 수 있으며, 상태 백업 플래그에 설정된 값을 통해 일반 요청에 대한 연산 처리 결과가 저장된 여부를 알 수 있다.
- <44> 예를 들어, 쓰기 연산의 처리가 종료된 다음 긴급 요청에 대한 연산 처리가 수행되었다면, 연산 처리 제어 장치(200)에는 쓰기 연산에 대한 결과가 존재하지 않으므로, 저장부(250)에 연산 결과값을 저장하며, 긴급 요청 관리부(220)는 상태 백업 플래그를 '1'로 설정한다. 이 후, 상태 체크부(230)는 상태 백업 플래그의 설정된 값을 체크하여 연산 처리 결과가 저장된 여부를 알 수 있다.
- <45> 연산 처리부(240)는 일반 요청 및 긴급 요청에 대한 연산을 처리한다. 여기서, 연산 처리부(240)는 긴급 요청에 대한 연산 처리가 발생되면 일반 요청에 대한 연산 처리를 중단시키고, 우선적으로 긴급 요청에 대한 연산 처리를 수행한다. 여기서, 연산 처리부(240)는 일반 요청에 대한 연산 및 긴급 요청에 대한 연산을 수행하는 각각의 처리부를 구비할 수도 있다. 또한, 연산 처리부(240)는 긴급 요청에 대한 연산 처리가 발생 시 연산 처리 중이던 일반 요청에 대한 연산을 바로 중단시킬 수도 있고, 진행 중이던 연산 처리가 완료될 때까지 대기할 수도 있다.
- <46> 또한, 연산 처리부(240)는 긴급 요청에 대한 연산 처리를 완료한 후, 상태 체크부(230)가 체크한 프리엠프트 플래그 값을 기초로 하여 일반 요청에 대한 연산 처리를 수행한다.
- <47> 예를 들어, 연산 처리부(240)는 상태 체크부(230)가 체크한 프리엠프트 플래그 값이 '1'인 경우, 일반 요청에 대한 연산 처리 중 긴급 요청에 대한 연산 처리가 수행되었음을 인지하고, 수행중이던 일반 요청에 대한 연산 처리를 다시 수행한다.
- <48> 또한, 연산 처리부(240)는 상태 체크부(230)가 체크한 프리엠프트 플래그 값이 '1'이고, 상태 백업 플래그 값도 '1'인 경우, 일반 요청에 대한 연산 처리가 완료된 다음에 긴급 요청에 대한 연산 처리가 수행되었음을 인지하고, 저장부(250)에 저장된 일반 요청에 대한 연산 처리 결과를 확인할 수 있다.

- <49> 저장부(250)는 연산 처리부(240)가 처리한 데이터를 저장하며, 긴급 요청에 대한 연산 처리가 발생되어 중단된 일반 요청에 대한 연산 처리의 결과를 저장한다.
- <50> 제어부(260)는 연산 처리 제어 장치(200)를 구성하는 각 기능성 블록들(210 내지 250)의 동작을 제어한다.
- <51> 도 4는 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법 중 일반 읽기 요청에 대한 연산 처리 과정을 나타낸 순서도이다. 여기서, 비휘발성 메모리는 그 종류 따라 상태를 통해 진행 중인 동작의 종류를 나타낸다고 가정한다.
- <52> 먼저, 연산 처리부(240)는 사용자의 입력부(210)로부터 요청된 읽기 연산을 처리한다(S410). 여기서, 프리엠프트 플래그는 '0'으로 초기화되어 있다. 또한, 단계 S410은 크리티컬 섹션(Critical Section)의 영역으로, 즉 크리티컬 섹션 동안에는 소프트웨어의 컨텍스트 스위치(context switch)가 일어나지 않아, 그동안에는 다른 연산이 있을 수 없으므로, 긴급한 연산의 요청이 없다고 보장되는 영역을 말한다. 이하, 단계에서도 크리티컬 섹션이 존재한다.
- <53> 그 다음, 요청된 데이터를 저장부(250)에 로드하고(S420), 상태 체크부(230)가 체크한 프리엠프트 플래그 값이 '1'인지를 확인한다(S430). 여기서, 프리엠프트 플래그 값을 확인하는 것은 긴급 요청에 대한 연산 처리가 수행되었는지 여부를 체크 하기 위해서이다.
- <54> 단계 S430을 수행한 결과, 프리엠프트 플래그 값이 '1'이 아닌 경우, 연산 처리부(240)는 로드된 데이터를 호스트 메모리로 전송한다(S440). 그 다음, 로드된 데이터의 전송이 완료되면 다시 한번 프리엠프트 플래그 값을 확인하여 긴급 요청에 대한 연산 처리에 의해 데이터 전송이 중단되지 않았는지를 확인한다.
- <55> 확인 결과 프리엠프트 플래그 값이 '1'이 아닌 경우(S450), 연산 처리부(240)는 ECC(Error Checking Code) 검사한다(S460).
- <56> 한편, 단계 S430을 수행한 결과 프리엠프트 플래그 값이 '1'인 경우, 긴급 요청에 대한 연산 처리가 수행된 것이기 때문에 연산 처리부(240)는 프리엠프트 플래그 값을 '0'으로 초기화한 후(S470), 처음부터 읽기 연산을 처리한다.
- <57> 도 5는 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법 중 일반 쓰기 요청에 대한 연산 처리 과정을 나타낸 순서도이다.
- <58> 먼저, 입력부(210)는 사용자 입력부로부터 쓰기 연산을 수행할 데이터를 전송받는다(S510). 여기서, 프리엠프트 플래그는 '0'으로 초기화되어 있다.
- <59> 그 다음, 데이터 전송이 완료되면 상태 체크부(230)는 긴급 요청에 대한 연산 처리가 수행되었는지 여부를 체크 하기 위해서 프리엠프트 플래그의 설정된 값을 확인한다(S520). 여기서, 단계 S520은 크리티컬 섹션(Critical Section)의 영역으로, 즉 크리티컬 섹션 동안에는 소프트웨어의 컨텍스트 스위치(context switch)가 일어나지 않아, 그동안에는 다른 연산이 있을 수 없으므로, 긴급한 연산의 요청이 없다고 보장되는 영역을 말한다. 이하, 단계에서도 크리티컬 섹션이 존재한다.
- <60> 단계 S520을 수행한 결과, 프리엠프트 플래그 값이 '1'인 경우, 쓰기 연산의 처리 도중 긴급 요청에 대한 연산 처리가 수행된 것이기 때문에 연산 처리부(240)는 프리엠프트 플래그 값을 '0'으로 초기화한 후(S580), 처음부터 쓰기 연산을 처리한다.
- <61> 한편, 단계 S520을 수행한 결과 프리엠프트 플래그 값이 '1'이 아닌 경우, 연산 처리부(240)는 전송된 데이터에 대한 쓰기 연산을 처리한다(S530). 그 다음, 상태 백업 플래그를 '0'으로 초기화 해둔다(S540). 여기서, 상태 백업 플래그를 '0'으로 초기화 해두는 것은, 쓰기 연산의 처리 도중 긴급 요청에 대한 연산 처리가 수행되면 쓰기 연산에 대한 현재 처리된 연산 결과를 잃어버리게 되므로, 상태 백업 플래그를 '0'으로 초기화 해둠으로써, 긴급 요청에 대한 연산 처리가 수행 시 쓰기 연산에 대한 현재 처리된 연산 결과 저장해 두라는 의미이다.
- <62> 그 다음, 쓰기 연산 처리가 완료되면(S550), 상태 체크부(230)는 프리엠프트 플래그 값과 상태 백업 플래그 값을 체크하여, 프리엠프트 플래그 값이 1이면서 상태 백업 플래그 값도 1인지 확인한다(S560).
- <63> 단계 S560을 수행한 결과, 프리엠프트 플래그 값이 1이면서 상태 백업 플래그 값도 1인 경우, 쓰기 연산의 처리가 완료된 후 긴급 요청에 대한 연산 처리가 수행된 것이므로, 저장부(250)에는 긴급 요청에 대한 연산 처리가 수행되기 전에 처리된 쓰기 연산의 처리에 대한 결과가 저장되어 있다. 이에, 연산 처리부(240)는 저장부(250)

0)에 저장된 쓰기 연산에 대한 결과를 확인한다(S590).

- <64> 한편, 단계 S560을 수행한 결과 리엠프트 플래그 값과 상태 백업 플래그 값 모두가 1이 아닌 경우, 일반 쓰기 연산을 수행하는 도중 긴급 요청에 대한 연산 처리가 없었으므로, 비휘발성 메모리에서 쓰기 연산에 대한 결과를 확인하여 처리한다(S570).
- <65> 도 6은 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법 중 일반 지우기 요청에 대한 연산 처리 과정을 나타낸 순서도이다.
- <66> 먼저, 연산 처리부(240)는 사용자의 입력부로부터 요청된 지우기 연산을 처리한다(S610). 여기서, 프리엠프트 플래그 및 상태 백업 플래그는 '0'으로 초기화 되어 있다. 또한, 단계 S610은 크리티컬 섹션(Critical Section)의 영역으로, 즉 크리티컬 섹션 동안에는 소프트웨어의 컨텍스트 스위치(context switch)가 일어나지 않아, 그 동안에는 다른 연산이 있을 수 없으므로, 긴급한 연산의 요청이 없다고 보장되는 영역을 말한다. 이하, 단계에서도 크리티컬 섹션이 존재한다.
- <67> 그 다음, 지우기 연산이 완료되면(S620), 상태 체크부(230)는 프리엠프트 플래그 값과 상태 백업 플래그 값을 체크하여, 프리엠프트 플래그 값이 1이면서 상태 백업 플래그 값도 1인지 확인한다(S630).
- <68> 단계 S630을 수행한 결과, 프리엠프트 플래그 값이 1이면서 상태 백업 플래그 값도 1인 경우, 지우기 연산의 처리가 완료된 후 긴급 요청에 대한 연산 처리가 수행된 것이므로, 저장부(250)에는 긴급 요청에 대한 연산 처리가 수행되기 전에 처리된 지우기 연산의 처리에 대한 결과가 저장되어 있다. 이에, 연산 처리부(240)는 저장부(250)에 저장된 지우기 연산의 처리에 대한 결과를 확인한다(S650).
- <69> 한편, 단계 S630을 수행한 결과 리엠프트 플래그 값과 상태 백업 플래그 값 모두가 1이 아닌 경우, 일반 지우기 연산을 수행하는 도중 긴급 요청에 대한 연산 처리가 없었으므로, 비 휘발성 메모리에서 지우기 연산에 대한 결과를 확인하여 처리한다(S640).
- <70> 도 7은 본 발명의 다른 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법 중 긴급 읽기 요청에 대한 연산 처리 과정을 나타낸 순서도이다.
- <71> 먼저, 긴급 읽기 요청에 대한 연산 처리가 발생되면, 진행 중이던 연산 처리에 대한 정보를 통해 마지막으로 수행된 연산 처리를 체크한다(S710). 여기서, 단계 S710은 크리티컬 섹션(Critical Section)의 영역으로, 즉 크리티컬 섹션 동안에는 소프트웨어의 컨텍스트 스위치(context switch)가 일어나지 않아, 그동안에는 다른 연산이 있을 수 없으므로, 긴급한 연산의 요청이 없다고 보장되는 영역을 말한다. 이하, 단계도 크리티컬 섹션 영역이 존재한다.
- <72> 단계 S710을 수행한 결과, 일반 쓰기 요청에 대한 연산을 수행한 경우(S712), 연산 처리 제어 장치(200)의 상태가 준비 상태가 될 때까지 대기한다(S714). 그 다음, 연산 처리 제어 장치(200)의 상태가 준비 상태가 되면, 상태 체크부(230)는 상태 백업 플래그의 설정된 값이 '0'인지를 체크한다(S716).
- <73> 단계 S716을 수행한 결과, 상태 백업 플래그의 값이 '0'인 경우, 연산 처리부(240)는 긴급 읽기 요청에 대한 연산을 처리하기 전에 현재 일반 쓰기 요청에 대한 연산 처리의 결과를 저장부(250)에 저장한다(S718). 그 다음, 긴급 요청 관리부(220)는 상태 백업 플래그를 '1'로 설정한다(S720).
- <74> 한편, 단계 S716을 수행한 결과 상태 백업 플래그의 값이 '0'이 아닌 경우(즉, 상태 백업 플래그의 값이 '1'인 경우), 저장부(250)에는 쓰기 연산의 처리에 대한 결과가 저장되어 있기 때문에 단계 S718 및 단계 S720 과정을 수행하지 않고 긴급 읽기 요청에 대한 연산을 처리할 수 있다.
- <75> 그 다음, 긴급 읽기 요청에 대한 연산이 처리 및 완료되면(S722), 프리엠프트 플래그 값을 '1'로 설정하고(S724), 중지되었던 일반 쓰기 요청에 대한 연산을 다시 처리한다(S726).
- <76> 한편, 단계 S710을 수행한 결과 일반 읽기 요청에 대한 연산을 수행한 경우(S730), 연산 처리 제어 장치(200)가 준비 상태가 될 때까지 대기한 후 (S732), 긴급 요청 관리부(220)는 상태 백업 플래그의 설정된 값을 '1'로 설정한다. 이 후, 단계 S720 내지 단계 S726 과정은 동일하게 수행된다.
- <77> 한편, 단계 S710을 수행한 결과 일반 지우기 요청에 대한 연산을 수행한 경우(S740), 연산 처리 제어 장치(200)가 준비 상태인지 바쁜 상태인지를 체크한다. 체크 결과 준비 상태가 아닌 경우(S742), 즉 바쁜 상태인 경우 진행중인 지우기 연산의 처리를 중단 시키고(S744), 연산 처리 제어 장치(200)가 준비 상태가 될 때까지 대기한다(S746).

- <78> 그 다음, 연산 처리 제어 장치(200)가 준비 상태가 되면, 긴급 읽기 요청에 대한 연산을 처리한다(S722). 이후, 단계 S724 및 단계 S726 과정은 동일하게 수행된다.
- <79> 한편, 도 7에서는 연산 처리 제어 장치(200)가 준비 상태가 될 때까지 대기한 후 긴급 요청에 대한 연산 처리를 수행하는 것을 설명하고 있지만, 연산 처리 제어 장치(200)가 준비 상태가 될 때까지 대기하지 않고 바로 긴급 요청에 대한 연산 처리를 수행할 수도 있다.
- <80> 또한, 본 발명에서는 긴급 읽기 요청에 대한 연산 처리 뿐만 아니라, 긴급 쓰기 요청 및 긴급 지우기 요청에 대한 연산 처리에 대하여도 상기 도 7의 과정과 동일하게 적용할 수 있다.
- <81> 이상 첨부된 도면을 참조하여 본 발명의 실시예를 설명하였지만, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다.

발명의 효과

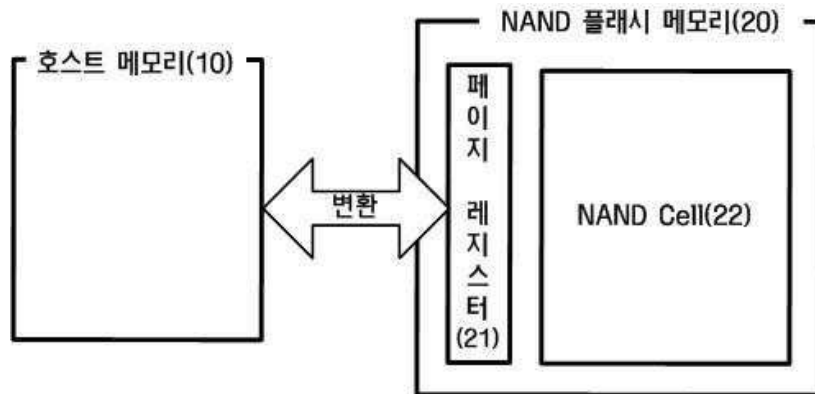
- <82> 상기한 바와 같은 본 발명의 비휘발성 메모리에서 연산 처리를 제어하는 장치 및 그 방법에 따르면 다음과 같은 효과가 하나 혹은 그 이상 있다.
- <83> 하드웨어적으로 파티션을 지원하지 않는 비휘발성 메모리에 대해서 소프트웨어를 통해 진행 중인 연산을 멈추고 긴급하게 요구되는 읽기/쓰기/지우기 연산을 수행할 수 있는 장점이 있다.

도면의 간단한 설명

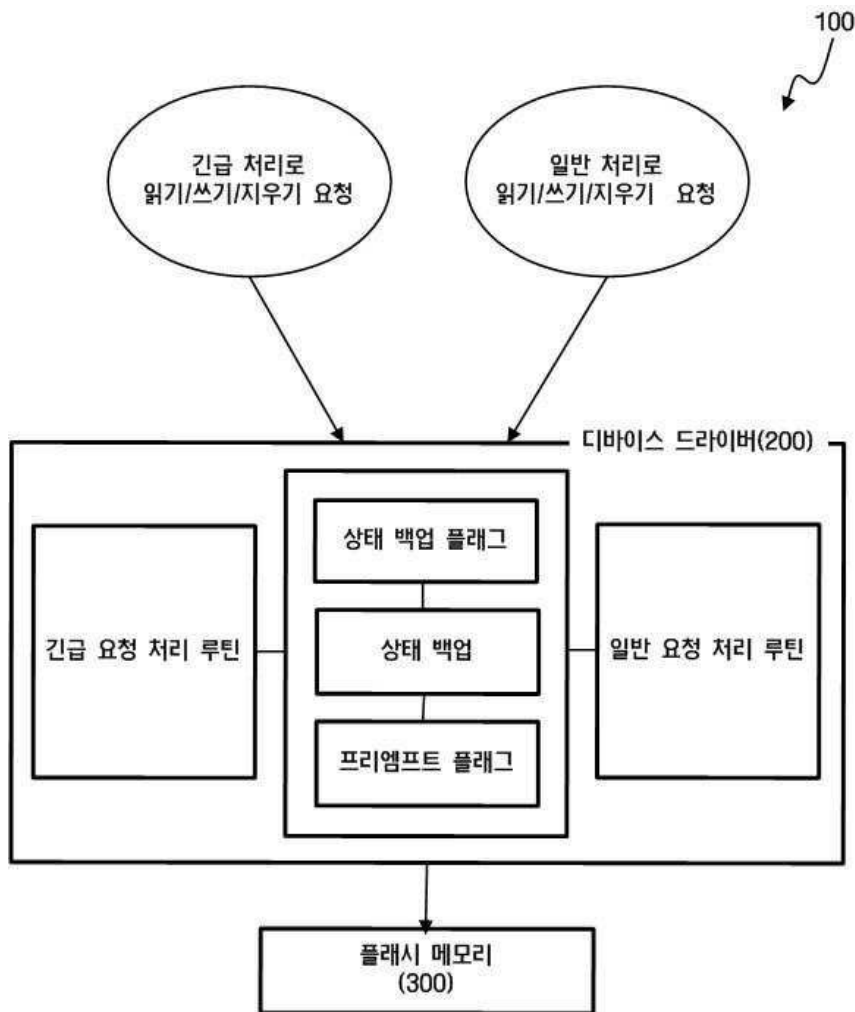
- <1> 도 1은 일반적인 NAND 플래시 메모리에서의 읽기 및 쓰기 연산을 수행하는 과정을 나타낸 도면.
- <2> 도 2는 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 장치에서 긴급 요청 및 일반 요청에 대한 연산 처리를 수행하는 과정을 나타낸 도면.
- <3> 도 3은 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 장치를 나타낸 블록도.
- <4> 도 4는 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법 중 일반 읽기 요청에 대한 연산 처리 과정을 나타낸 순서도.
- <5> 도 5는 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법 중 일반 쓰기 요청에 대한 연산 처리 과정을 나타낸 순서도.
- <6> 도 6은 본 발명의 일 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법 중 일반 지우기 요청에 대한 연산 처리 과정을 나타낸 순서도.
- <7> 도 7은 본 발명의 다른 실시예에 따른 비휘발성 메모리에서 연산 처리를 제어하는 방법 중 긴급 읽기 요청에 대한 연산 처리 과정을 나타낸 순서도.
- <8> <도면의 주요 부분에 관한 부호의 설명>
- <9> 210 : 입력부 220 : 긴급 요청 관리부
- <10> 230 : 상태 체크부 240 : 연산 처리부
- <11> 250 : 저장부 260 : 제어부

도면

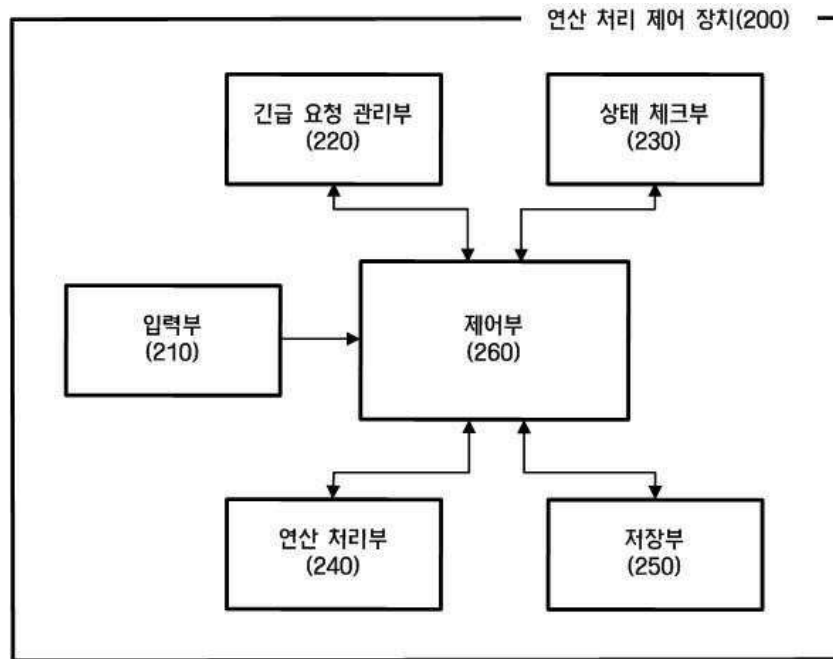
도면1



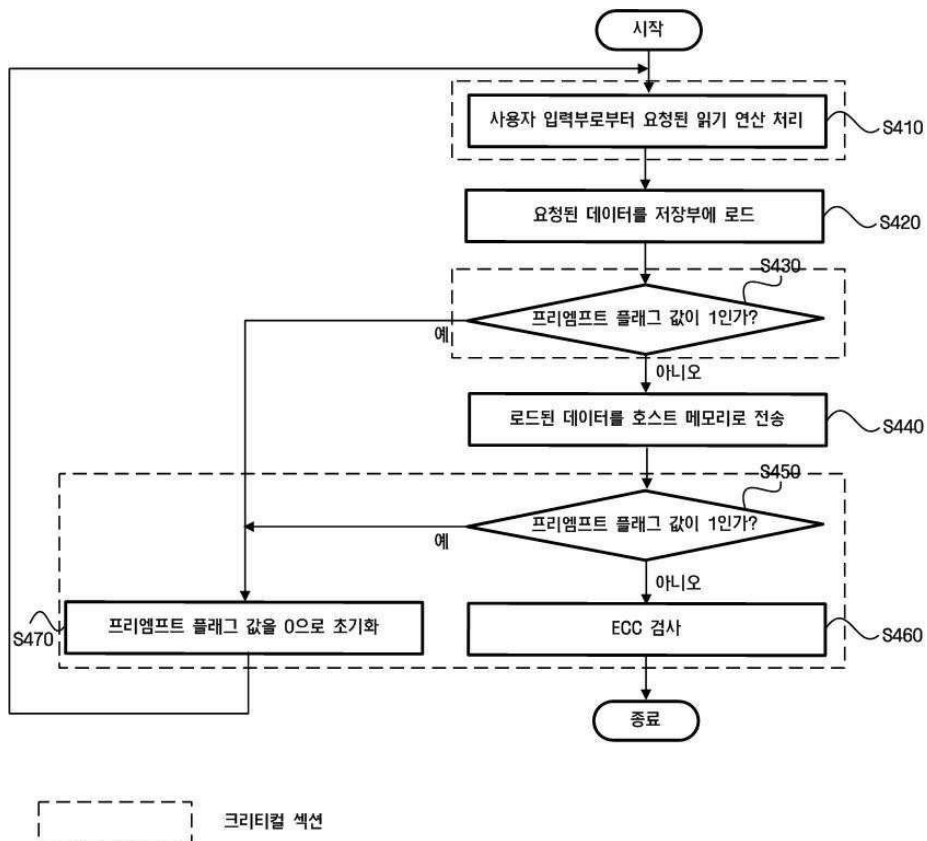
도면2



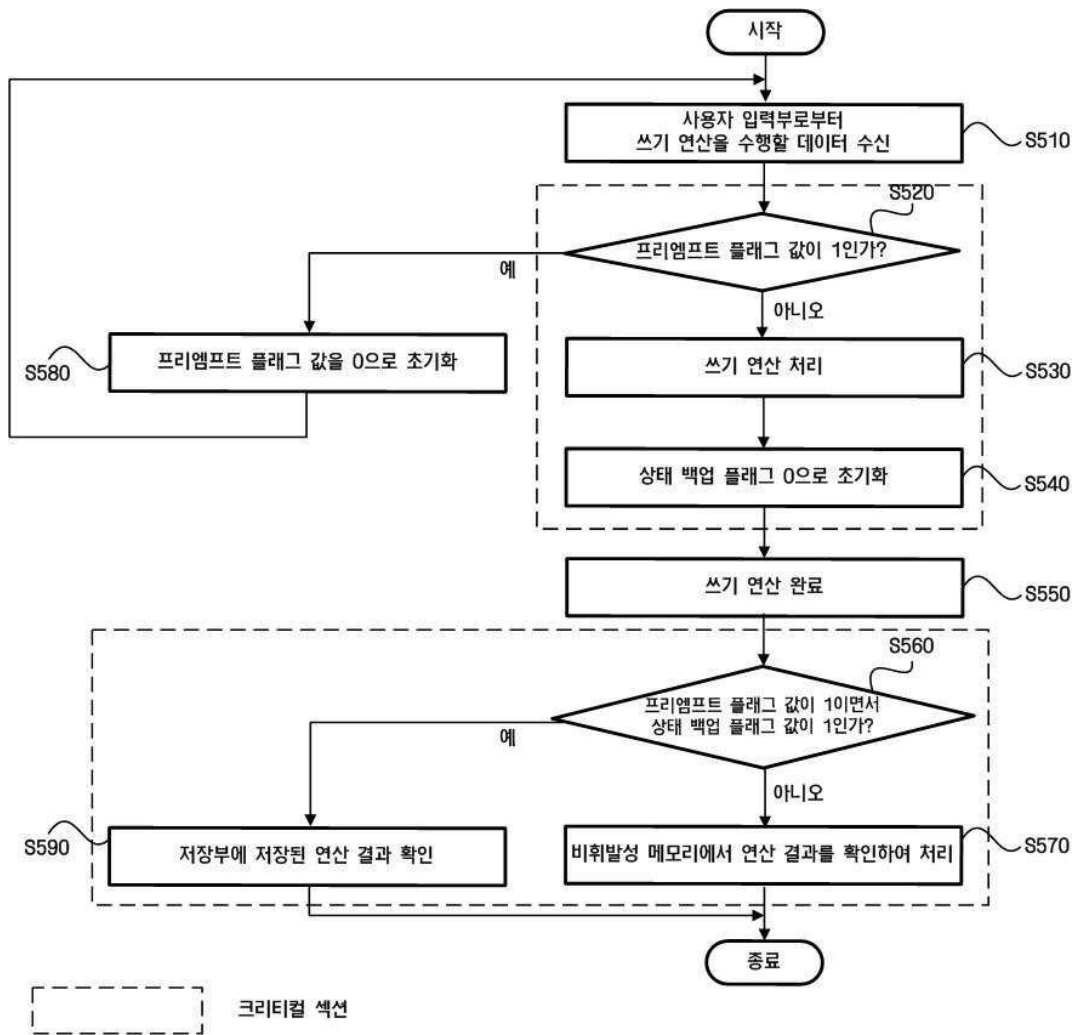
도면3



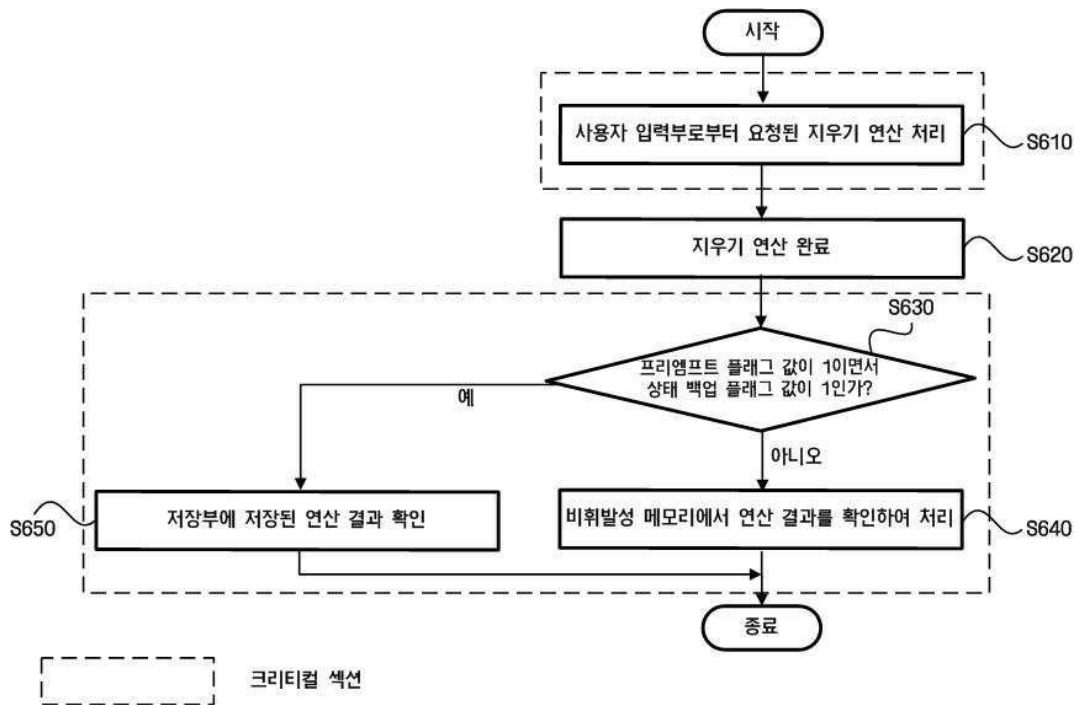
도면4



도면5



도면6



도면7

