

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.⁷
G06F 12/00

(11) 공개번호 특2001-0007418
(43) 공개일자 2001년01월26일

(21) 출원번호	10-2000-0033403
(22) 출원일자	2000년06월17일
(30) 우선권주장	09/340,376 1999년06월28일 미국(US)
(71) 출원인	인터내셔널 비지네스 머신즈 코포레이션 포만 제프리 엘 미국 10504 뉴욕주 아몬크
(72) 발명자	스틸찰스리 미국80027콜로라도주루시빌웨스트하퍼스트리트333
(74) 대리인	김창세, 김원준, 장성구

심사청구 : 있음

(54) 패딩 데이터 추가 및 추출 회로

요약

논리 섹터 포맷과 물리 섹터 포맷 사이의 데이터 변환을 관리하는 회로가 개시된다. 패딩 모드에서, 이 회로는 논리 길이를 갖는 논리 섹터 포맷의 데이터 스트림에 패딩값을 추가함으로써, 물리 길이를 갖는 물리 섹터 포맷의 데이터 스트림을 발생시킨다. 추출 모드에서, 이 회로는 물리 섹터 포맷의 데이터 스트림으로부터 논리 섹터 포맷의 데이터 스트림을 추출해 낸다.

대표도

도4

명세서

도면의 간단한 설명

도 1은 종래의 일치되지 않는 논리 및 물리 섹터를 예시하는 도면,
도 2는 종래 기술에 따라 소프트웨어로 논리 섹터를 패딩하는 것을 예시하는 도면,
도 3a 및 도 3b는 본 발명에 따라 하드웨어로 논리 섹터를 패딩하는 것을 보여주는 도면,
도 4는 본 발명에 따라 논리 섹터에 대해 패딩값을 추가 및 추출하는 논리 회로의 개략도,
도 5는 본 발명에 따라 논리 섹터에 패딩값을 추가하는 하드웨어에 의한 카운터의 이용을 도시하는 도면,
도 6은 본 발명에 따라 물리 섹터로부터 논리 섹터를 추출하는 하드웨어에 의한 카운터의 이용을 도시하는 도면.

도면의 주요 부분에 대한 부호의 설명

404 : 논리 섹터 길이 레지스터	406 : 물리 섹터 길이 레지스터
408 : 섹터 레지스터/카운터	410 : 패드값 레지스터
412 : 논리 길이 카운터	414 : 물리 길이 카운터
426, 428, 430 : 0 검출기	424 : 제어 로직
448, 458, 452, 460 : 데이터 래치	450 : 패드/데이터 다중화기
454 : 메모리 인터페이스	

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 디스크 저장 시스템에 관한 것으로, 더 구체적으로는 물리 섹터의 경계(boundary of physical sector)에 대해 논리 섹터의 경계(boundary of logical sector)를 맞추기(align) 위해 논리 섹터에 대해

패딩 데이터(padding data)를 추가(appending) 및 추출(extracting)하는 회로에 관한 것이다.

전형적인 디스크 저장 시스템상에서 데이터는 물리 섹터로 편성되는데, 이 때 각각의 물리 섹터는 사전 정의된 개수의 연속적인 바이트를 포함하고 있다. 데이터가 디스크 드라이브에 기록되거나 또는 디스크 드라이브로부터 판독될 때, 전송되는 데이터의 양은 반드시 물리 섹터의 단위로 전송되어야 하고, 모든 데이터 전송은 반드시 물리 섹터의 경계에서 시작 및 종료해야 한다. 예를 들어, 물리 섹터가 514 바이트로 이루어진다면, 데이터 전송은 514 바이트, 1028 바이트, 1542 바이트로 이루어질 수 있다.

프로세서가 디스크 드라이브를 액세스할 때, 예를 들면, 프로세서가 파일을 기록할 때, 프로세서는 아마 정확히 하나의 물리 섹터 또는 그 정수 배의 물리 섹터에 대해 동작할 필요가 없을 것이다. 대신에, 프로세서는 아마도 예를 들어 514 바이트, 1028 바이트, 1542 바이트 등이 아닌 양의 데이터를 필요로 할 것이다.

프로세서와 데이터 저장 시스템간에 데이터를 전송하는 상황에서, 프로세서에 의해 조작되는 데이터는 논리 섹터로 불린다. 전술한 바와 같이, 논리 섹터의 경계는 통상적으로 물리 섹터의 경계와 일치하지 않는다. 따라서, 데이터 전송은 종종 실제로 프로세서에 의해 이용되는 것보다 더 많은 데이터의 조작을 수반한다.

디스크 저장 시스템의 설계 목적은 기록 밀도(recording density)와 처리량(throughput)을 최대화하는 것을 포함한다. 기록 밀도는 디스크 저장 시스템의 전체 메모리 용량에 대해 유효 데이터가 점유하는 메모리의 비율(ratio)이다. 처리량은 특정 시간 동안 시스템에 의해 처리되는 전체 작업량의 측정치이다.

다음의 특허들은 기록 밀도와 처리량을 최대화하려고 시도한 대표적인 종래 기술이다. 일반적으로, 이들은 디스크 드라이브상의 데이터를 편성하거나 또는 논리 및 물리 섹터 경계들의 불일치(misalignment)를 처리하는 시스템 또는 방법 측면을 다룬다.

발명의 명칭이 "Variable Zone Layout For Information Storage Disk Drive"인 에모 등(Emo et al.)의 미국 특허 제 5,596,458호는 다수의 디스크 표면을 갖는 디스크에 대한 각각의 판독/기록 헤드의 성능 특성을 측정함으로써 기록 밀도를 개선하는 방법 및 시스템을 기술한다. 존(zone) 경계는 성능 특성을 근거하여, 그리고 데이터가 특정 존에 대해 판독 및 기록되는 빈도(frequency)에 근거하여 각각의 디스크 표면에 대해 설정된다. 디스크의 표면상의 존 경계가 반드시 수직적으로 정렬될 필요는 없을 것이다. 기록 밀도는 사전 정의된 수준보다 더 낮은 성능 특성을 갖는 판독/기록 헤드의 판독/기록 빈도를 줄임으로써 개선된다.

발명의 명칭이 "Multi-Drive Virtual Mass Storage Device And Method Of Operating Same"인 클라인 등(Klein et al.)의 미국 특허 제 5,671,439호는 상이한 물리 장치에 대해 할당되는 블록들로 편성되는 가상의 대용량 저장 장치(virtual mass storage device)를 기술한다. 물리 장치는 가상 장치의 총 전송율을 증가시키기 위해 병렬로 동작한다.

발명의 명칭이 "Changing Page Size In Storage Media Of Computer System"인 맨지스 등(Manges et al.)의 미국 특허 제 5,765,201호는 데이터가 헤더에 의해 정의되는 블록 혹은 섹터에 저장되는 시스템에 적용되는 발명을 기술한다. 메모리 대상과 세그먼트를 정의하고, 물리 메모리에 가상 메모리 어드레스를 지정하기 위해 테이블이 이용된다. 헤더와 테이블은 물리 메모리의 섹터 혹은 페이지내의 모든 데이터를 재기록함이 없이 변경될 수 있다. 따라서, 페이지 사이즈 역시 새로운 시스템 구성 요소를 수용하도록 변경될 수 있다.

발명의 명칭이 "Hardware Alignment In A Headerless Disk Drive Architecture"인 쿠울 등(Kool et al.)의 미국 특허 제 5,802,584호는 디스크 드라이브의 물리 섹터로/로부터의 데이터 전송 및 물리 섹터로부터의 데이터 전송을 제어하는 시스템을 개시한다. 이 시스템은 이벤트 제어 워드(event control word)를 디코딩하는 디스크 컨트롤러(disk controller)와, 판독/기록 헤드가 디스크상에서 섹터를 통과할 때마다 증가하는 카운터(counter)를 포함한다. 카운터는 헤드가 위치되어 있는 데이터 프레임을 표시함으로써 판독/기록 헤드의 위치를 나타낸다. 정렬 프로세서(alignment processor)는 카운터에 의해 표시되는 데이터 프레임에 대응하는 제어 워드를 찾기 위해 제어 워드를 주사(scan)한다. 주사하는 동안, 정렬 프로세서는 또한 판독/기록 헤드의 위치에 의해 표시되는 논리 및 물리 섹터 카운터를 정렬한다. 그 이후, 디스크 컨트롤러는 정렬된 값들을 이용하여 제어 워드를 디코딩하고 데이터 전송을 제어하기 시작한다.

발명의 명칭이 "System And Method For Providing Variable Sector-Format Operation To A Disk Access System"인 머피 등(Murphy et al.)의 미국 특허 제 5,813,025호는 사전 정의된 논리 섹터 포맷과 가변 물리 섹터 포맷 사이에서 I/O 요구를 처리하는 시스템 및 방법을 기술한다. 판독 동작이 물리 섹터의 중앙에서 시작한 경우, 첫 번째 논리 섹터의 앞에 있는 데이터는 폐기되고, 사용자 요청 데이터(user request data)는 사용자 버퍼에 저장된다. 기록 동작이 물리 섹터의 중앙에 위치된 논리 섹터에 대한 것일 경우, 물리 섹터는 논리 섹터에 대한 기록 요구에 따라 판독 및 수정된 뒤, 완전한 물리 섹터로서 디스크에 다시 기록된다. 버퍼는, 디스크에 다시 기록되기 이전에 판독된 뒤 수정되는 데이터를 유지하기 위해 보존된다.

도 1은 종래 기술에 따른 논리 섹터와 물리 섹터 사이의 관계를 예시한다. 디스크 드라이브상의 데이터 스트림(data stream)은 각각 514 바이트인 물리 섹터 1~128로 편성(organized)되는데, 이것은 각각 2054 바이트인 논리 섹터 #1~#32를 갖고 있다.

논리 섹터 #1~#32는 연속적인 데이터 스트림으로서 프로세서로부터 디스크 드라이브에 기록된 것으로, 즉, 논리 섹터 사이에 갭(gap)이나 충전 문자(filler)가 없이 기록된 것이다. 이것은 기록 밀도를 개선하기 위해 이용되는 기법으로, 유효한 데이터(valid occurrence of data) 사이에 무효(invalid) 혹은 초과(excess) 데이터의 존재와 같은 요인에 의해 악영향을 받는다.

4개의 물리 섹터는 총 2056 바이트의 길이이고, 각각의 2054 바이트의 논리 섹터를 수용하기 위해서는 4개의 물리 섹터가 요구됨을 유의하자. 예를 들어, 논리 섹터 #1이 물리 섹터 1~4에 존재한다. 2056 바이트 길이의 4개의 물리 섹터는 2054 바이트 길이의 논리 섹터보다 2 바이트 더 크므로, 물리 섹터 4의 끝(end)으로부터 2 바이트 이전에 논리 섹터 #1이 종료하고 논리 섹터 #2가 시작한다. 다음, 논리 섹터 #2는 물리 섹터 8의 끝보다 4 바이트 앞서 종료한다. 즉, 논리 섹터 #2는 물리 섹터 4의 바이트 512에서 시작해서 물리 섹터 8의 바이트 510에서 종료한다. 이러한 2 바이트 차이는 누적되어, 논리 섹터 #32는 물리 섹터 128의 종료 경계로부터 64 바이트(즉, 32×2 바이트) 이전에 종료한다.

32개의 논리 섹터의 데이터 스트림을 디스크 드라이브에 기록할 때, 프로세서는 논리 섹터 #32의 끝에 64 바이트의 패딩 데이터를 추가해야 한다. 기록 동작이 끝나면, 디스크 드라이브는 인터럽트(105)를 프로세서에게 제기한다. 인터럽트(105)를 수신한 후, 프로세서는 32개 논리 섹터로 이루어진 또 다른 집합을 디스크 드라이브로 보낸다.

논리 및 물리 섹터간의 관계의 미세한 차이는 디스크 드라이브로부터 데이터를 판독할 때 더욱 더 명확해진다. 예를 들면, 프로세서가 논리 섹터 #2를 판독하고자 하는 것으로 가정하자. 프로세서는 논리 섹터 #2의 어드레스를 결정하고 이것이 물리 섹터 4의 바이트 512에서 시작하여 물리 섹터 8의 바이트 510에서 종료하는 것을 인지해야 한다. 디스크 드라이브는 오로지 물리 섹터의 단위로만 데이터를 전송할 수 있고 모든 데이터 전송은 물리 섹터의 경계에서 시작 및 종료해야 하므로, 디스크 드라이브는 물리 섹터 4~8에 포함된 모든 데이터를 판독해야만 한다. 그 다음에, 프로세서는 초과 데이터(excess data)를 폐기해야 한다. 특히, 물리 섹터 4의 바이트 1~511의 데이터와 물리 섹터 8의 바이트 513 및 514의 데이터를 폐기해야 한다. 논리 섹터가 물리 섹터와 정렬되지 않은 도 1에 도시된 것과 같은 시스템이 갖는 문제중 하나는 초과 데이터의 조작이 시스템의 총 처리량을 감소시킨다는 것이다.

논리 및 물리 섹터 경계의 불일치를 다루는 또 다른 종래 기법은 논리 섹터의 길이가 물리 섹터의 길이 까지 확장되도록 각각의 논리 섹터를 패딩하는 것이다. 이런 조작이 비교적 간단하기는 해도, 통상적으로 소프트웨어로 구현되며, 이러한 구현은 이후에 논의되는 결과를 동반한다.

도 2는 소프트웨어를 이용해 논리 섹터를 패딩하는 것을 예시한다. 디스크 드라이브상의 데이터 스트림은 각각 514 바이트인 물리 섹터 1~128로 편성되고, 이것은 각각 2054 바이트인 논리 섹터 #1~#32를 갖는다. 각각 2054 바이트인 논리 섹터를 수용하기 위해서는 총 2056 바이트 길이인 4개의 물리 섹터가 필요하다. 예를 들어, 논리 섹터 #1은 물리 섹터 1~4에 존재한다. 2056 바이트 길이인 물리 섹터 1~4는 2054 바이트 길이인 논리 섹터 #1보다 2 바이트 더 크므로, 논리 섹터 #1은 물리 섹터 4의 끝에서부터 2 바이트 앞서 종료한다. 이 2 바이트는 논리 섹터 #1의 끝에 소프트웨어에 의해 추가되는 패딩 데이터(205)로 채워진다. 논리 섹터 #2의 시작은 물리 섹터 5의 시작과 일치한다.

도 2에 표시된 데이터 스트림은 연속적인 데이터 스트림이 아니다. 기록 동작 동안, 프로세서는 논리 섹터 #1을 위해 2054 바이트의 데이터를 준비하고, 2 바이트의 패딩 데이터(205)를 추가한 뒤, 이 데이터를 디스크 드라이브로 보낸다. 기록 동작이 종료하면, 디스크 드라이브는 인터럽트(210)를 프로세서에게 제기한다. 인터럽트(210)를 수신한 후, 프로세서는 데이터(215)가 추가된 논리 섹터 #2를 디스크 드라이브로 보낸다. 이러한 기록 동작을 마치면, 디스크 드라이브는 프로세서에게 인터럽트(220)를 제기한다.

인터럽트 처리 루틴(interrupt handlers)은 다른 태스크로부터 프로세서를 전용(divert)하며, 이것은 종종 인터럽트를 처리하도록 준비하기 위한 상당한 오버헤드(overhead)와 연관된다. 도 2에 도시한 것처럼, 소프트웨어로 논리 섹터의 끝에 패딩 데이터를 추가하는 시스템이 갖는 문제중 하나는 기록 동작이 종료할 때마다 디스크 드라이브가 프로세서를 인터럽트한다는 것이다.

발명이 이루고자하는 기술적 과제

따라서, 본 발명의 목적은 디스크 드라이브에 기록할 때 논리 섹터의 끝 경계를 물리 섹터의 끝 경계에 맞추기 위해 논리 섹터에 패딩 데이터를 추가하되, 이것을 하드웨어로 구현하는 개선된 시스템을 제공하는 것이다.

본 발명의 다른 목적은 하드 드라이브로부터 데이터를 판독할 때 논리 섹터를 산출하기 위해 물리 섹터로부터 패딩 데이터를 제거하되, 이것을 하드웨어로 구현하는 개선된 시스템을 제공하는 것이다.

본 발명의 또 다른 목적은 가변 길이의 패딩 데이터를 추가 및 추출함으로써 논리 및 물리 섹터가 가변 길이를 갖도록 하되, 이것을 하드웨어로 구현하는 개선된 시스템을 제공하는 것이다.

본 발명에 따르면, 소정의 회로가 논리 섹터 포맷과 물리 섹터 포맷 사이에 데이터의 변환을 관리한다. 패딩 모드에서, 이 회로는 논리 데이터 스트림(이 스트림의 논리 섹터는 소정의 논리 길이를 가짐)에 패딩값을 추가하여, 물리 데이터 스트림(이 스트림의 물리 섹터는 소정의 물리 길이를 가짐)을 만든다. 주출 모드에서는, 이 회로가 물리 데이터 스트림으로부터 논리 데이터 스트림을 추출한다.

패딩 모드에서, 이 회로는 스위치, 논리 길이 카운터(logical length counter), 물리 길이 카운터(physical length counter), 콘트롤러를 포함한다. 스위치는 논리 데이터 스트림을 수신하는 입력과, 패딩값을 수신하는 입력과, 물리 데이터 스트림을 전송하는 출력을 갖는다. 논리 길이 카운터는 논리 데이터 스트림의 바이트를 계수하고, 물리 길이 카운터는 물리 데이터 스트림의 바이트를 계수한다. 콘트롤러는 논리 길이가 계수될 때까지 논리 데이터 스트림을 선택하도록 스위치를 제어하고, 물리 길이가 계수될 때까지는 패딩값을 선택하도록 스위치를 제어한다.

주출 모드에서, 이 회로는 데이터 래치(data latch), 논리 길이 카운터, 물리 길이 카운터를 포함한다. 데이터 래치는 물리 데이터 스트림을 수신하는 입력과, 논리 데이터 스트림을 전송하는 출력을 갖는다. 논리 길이 카운터는 논리 데이터 스트림의 바이트를 계수하고, 물리 길이 카운터는 물리 데이터 스트림

의 바이트를 계수한다. 콘트롤러는 논리 길이가 계수될 때까지 데이터 래치가 물리 데이터 스트림을 수신하도록 만들고, 물리 길이가 계수될 때까지는 데이터 래치를 동작 불능으로 만든다.

발명의 구성 및 작용

본 발명은 물리 섹터의 경계에 대해 논리 섹터의 경계를 정렬하기 위해 논리 섹터로/로부터 패딩 데이터를 추가/추출하는 회로를 개시한다. 먼저, 회로의 개념적인 동작이 도 3a 및 도 3b의 논의에서 설명된다. 그 뒤, 회로 그 자체에 대해서는 도 4 내지 도 6에 대한 논의에서 설명된다.

도 3a 및 도 3b는 본 발명에 따라 하드웨어로 논리 섹터를 패딩하는 것을 도시한다. 도 3a는 일반적인 경우를, 도 3b는 본 발명의 예시적인 적용을 도시한다.

도 3a에서, 디스크 드라이브상의 데이터 스트림은 물리 섹터 1~32로 편성되고, 이것은 프로세서로부터 제기된 데이터의 논리 섹터 1~32를 보유한다. 패딩 데이터는 각각의 논리 섹터의 끝에 추가되어, 다음 논리 섹터의 시작을 다음 물리 섹터의 시작에 맞춘다. 예를 들어, 패딩 데이터(310)가 논리 섹터 #1에 추가되어, 논리 섹터 #2의 시작을 물리 섹터 #2의 시작에 맞춘다. 패딩 데이터는 프로세서와 디스크 드라이브 사이에 동작적으로 위치된 회로에 의해 각각의 논리 섹터에 추가된다.

기록 동작 동안, 프로세서는 논리 섹터 1~32를 연속 데이터 스트림으로서 준비한다. 프로세서는 이 데이터 스트림을 전술한 회로에 제기한다. 이 회로는 데이터 스트림을 디스크 드라이브로 전송하고, 각각의 논리 섹터의 끝에 패딩 데이터를 추가한다. 디스크 드라이브는 회로로부터 데이터 스트림을 수신하면 이 데이터 스트림을 기록한다. 모든 데이터 스트림을 기록한 후, 디스크 드라이브는 인터럽트(350)를 프로세서에게 제기한다. 그 이후에 프로세서는 회로에 대해 다른 데이터 스트림을 발행할 수도 있다.

본 발명은 프로세서의 소프트웨어가 아닌 회로에 의해 구현되므로, 여러 번의 인터럽트를 처리하는 종래 기술의 문제가 다소 해결되어, 시스템 처리량이 향상된다.

도 3b는 특정 응용에서 본 발명을 실행하는 것을 예시한다. 디스크 드라이브상의 데이터 스트림은 각각 514 바이트인 물리 섹터 1~128로 편성되고, 이것은 각각 2054 바이트인 논리 섹터 #1~#32를 포함한다. 총 2056 바이트 길이인 4개의 물리 섹터는 각각 2054 바이트인 논리 섹터를 수용하는 것이 요구된다. 예를 들어, 논리 섹터 #1은 물리 섹터 1~4에 존재한다. 2056 바이트 길이인 물리 섹터 1~4는 2054 바이트 길이인 논리 섹터 #1보다 2 바이트 더 크므로, 논리 섹터 #1은 물리 섹터 4의 끝에서부터 2 바이트 앞서 종료한다. 본 발명에 따른 회로는 논리 섹터 #1의 끝에 2 바이트의 패딩 데이터를 추가하여, 논리 섹터 #2의 시작이 물리 섹터 #5의 시작과 일치하도록 만든다. 마찬가지로, 논리 섹터 #2는 물리 섹터 8의 끝에서부터 2 바이트 앞서 종료한다. 회로는 논리 섹터 #2의 끝에 2 바이트의 패딩 데이터를 추가하여, 논리 섹터 #3의 시작이 물리 섹터 #9의 시작과 일치하도록 만든다.

기록 동작 동안, 프로세서는 논리 섹터 #1~#32를 연속 데이터 스트림으로서 마련한다. 프로세서는 이 회로에 데이터 스트림을 발행한다. 회로는 이 데이터 스트림을 디스크 드라이브로 보내고, 각각의 논리 섹터의 끝에 2 바이트의 패딩 데이터를 추가한다. 디스크 드라이브는 회로로부터 수신되는 대로 데이터 스트림을 기록한다. 모든 데이터 스트림을 기록한 후, 디스크 드라이브는 프로세서에게 인터럽트(350a)를 제기한다. 프로세서는 그 이후에 다른 데이터 스트림을 회로로 보낼 수도 있다.

판독 동작은 예시를 통해 가장 잘 설명된다. 프로세서가 논리 섹터 #2를 판독하고자 한다고 가정하자. 프로세서는 논리 섹터 #2의 어드레스를 판정하여, 이 논리 섹터가 물리 섹터 5에서 시작하여, 물리 섹터 8에서 종료한다는 것을 인식해야 한다. 디스크 드라이브는 오로지 물리 섹터의 단위로만 데이터를 전송할 수 있고, 모든 데이터 전송은 물리 섹터의 경계에서 시작하고 종료해야 하므로, 디스크 드라이브는 물리 섹터 4~8까지 그 속에 포함된 모두를 판독해야 한다.

디스크 드라이브는 물리 섹터 4~8을 회로로 보낸다. 물리 섹터 8의 마지막 2 바이트의 데이터는 기록 기록 동안 이 회로에 의해 추가된 패딩 데이터를 포함하고 있다. 판독 동작 동안에는, 회로가 이 2 바이트의 패딩 데이터를 폐기하여, 어떠한 초과 데이터도 갖지 않는 논리 섹터 #2를 프로세서에게 전송할 것이다.

도 4는 본 발명에 따라 논리 섹터에 대해 패딩값을 추가 및 추출하는 논리 회로(400)의 개략도이다. 이 회로의 주요 시스템 구성 요소는 제어 로직(424), 논리 길이 카운터(412), 물리 길이 카운터(414), 패드/데이터 다중화기(pad/data multiplexer)(450) 및 데이터 래치(458)이다. 또한, 논리 섹터 길이 레지스터(404), 물리 섹터 길이 레지스터(406), 섹터 레지스터/카운터(408), 패드값 레지스터(410), 0 검출기(zero detects)(426, 428, 430), 버퍼(446, 456, 462), 데이터 래치(448, 452, 460) 및 메모리 인터페이스(454)를 포함한다.

프로세서 인터페이스 버스(402)는 프로세서(도시 안됨)로부터 논리 섹터 길이 레지스터(404), 물리 섹터 길이 레지스터(406), 섹터 레지스터/카운터(408) 및 패드값 레지스터(410)로의 신호에 대한 경로를 제공한다. 논리 섹터 레지스터(404)의 출력은 논리 길이 카운터(412)의 입력에 접속된다. 물리 섹터 길이 레지스터(406)의 출력은 물리 길이 카운터(414)의 입력에 접속된다. 패드값 레지스터(410)의 출력은 패드/데이터 다중화기(450)의 입력에 접속된다.

논리 길이 카운터(412), 물리 길이 카운터(414), 섹터 레지스터/카운터(408)의 출력들은 0 검출기(zero detects)(426, 428, 430)의 입력에 각각 접속된다. 0 검출기(426, 428, 430)의 출력은 제어 로직(424)의 입력에 접속된다.

프로세서의 데이터 스트림은 버퍼(446)에 의해 수신되고, 데이터 래치(448)를 통해 패드/데이터 다중화기(450)에 접속된다. 데이터는 패드/데이터 다중화기(450)의 출력으로부터 데이터 래치(452)를 통해 메모리 인터페이스(454)로 진행하는데, 메모리 인터페이스는 데이터를 디스크 드라이브 버퍼(도시 안됨)로

보낸다.

디스크 드라이브 버퍼로부터의 데이터 스트림은 메모리 인터페이스(454)에 의해 수신되어, 데이터 래치(460)로 보내진다. 데이터 래치(460)의 출력은 데이터 래치(458)에 접속된다. 데이터 래치(458)의 출력은 버퍼(456)에 접속되고, 버퍼는 데이터를 프로세서로 보낸다.

제어 로직(424)은 동작 회로(400)를 제어하기 위한 몇 개의 신호를 발생한다. 인에이블 신호(enable signal)(416)는 섹터 레지스터/카운터(408)의 계수 기능을 동작 가능 상태로 만든다. 로드 신호(load signal)(418)는 논리 섹터 길이 레지스터(404)와 물리 섹터 길이 레지스터(406)로부터 각각 값들을 로딩하도록 논리 길이 카운터(412)와 물리 길이 카운터(414)를 제어한다. 인에이블 신호(420, 422)는 논리 길이 카운터(412)와 물리 길이 카운터(414)의 계수 기능을 각각 동작 가능 상태로 만든다. 메모리 제어 신호(444)는 메모리 인터페이스(454)의 동작을 지시한다. 인에이블 신호(432, 434, 440, 442)는 각각 데이터 래치(458, 448, 460, 452)로의 데이터 로딩을 제어한다. 선택 신호(436)는 데이터 래치(448) 또는 패드값 레지스터(410)로부터 데이터를 입력하기 위해 패드/데이터 다중화기(450)를 제어한다. 제어 로직(424)은 또한 데이터 대기 신호(409)를 발생시키기 위해 버퍼(411)를 통해 전송되는 데이터 대기 신호(431)를 발생한다. 데이터 유효 신호(data validation signal)(438)는 버퍼(462)를 통해 전송된다.

회로(400)는 패딩 모드(padding mode)와 추출 모드(extracting mode)를 갖는다. 패딩 모드에서, 이 회로(400)는 논리 섹터의 데이터 스트림(즉, 논리 데이터 스트림)을 수신하고, 각각의 논리 섹터의 끝에 패딩 데이터를 추가하여 물리 섹터의 데이터 스트림(즉, 물리 데이터 스트림)을 만든다. 추출 모드에서는, 회로(400)가 물리 섹터의 데이터 스트림을 수신하고, 패딩 데이터를 억제하여, 논리 섹터의 데이터 스트림을 추출한다. 이 두 가지 모드의 각각의 동작에 대한 세부 내용은 이하에 설명된다.

어떤 시스템 구성 요소의 기능은 패딩 모드의 동작과도 관련이 있고 추출 모드의 동작과도 관련 있다. 각 모드의 동작을 검사하기 전에 이런 시스템 구성 요소의 동작을 먼저 논의한다.

논리 섹터는 사전 정의된 개수의 바이트의 논리 길이를 갖고, 물리 섹터는 사전 정의된 개수의 바이트의 물리 길이를 갖는다. 논리 길이 및 물리 길이는 프로세서에 의해 프로그램 가능하다. 프로세서 인터페이스 버스(402)를 통해, 프로세서는 논리 길이를 논리 섹터 길이 레지스터(404)에 로딩하고, 물리 길이를 물리 섹터 길이 레지스터(406)에 로딩한다. 프로세서는 섹터의 개수를 섹터 레지스터/카운터(408)에 로딩함으로써 디스크 드라이브로로부터 전송될 섹터의 개수를 지정한다. 또한 프로세서는 패딩값을 패드값 레지스터(410)에 로딩함으로써 패딩 데이터에 대한 값을 지정한다.

어떠한 동작 모드에서도든, 각각의 데이터 섹터를 전송하기 위해서는, 제어 로직(424)이 논리 길이 카운터(412)를 제어하여 논리 섹터 길이 레지스터(404)로부터 논리 길이를 로딩하고, 물리 길이 카운터(414)를 제어하여 물리 섹터 길이 레지스터(406)로부터 물리 길이를 로딩한다. 데이터 전송 동안에 제어 로직(424)은 논리 길이 카운터(412), 물리 길이 카운터(414), 섹터 레지스터/카운터(408)를 동작 가능 상태로 만든다. 논리 길이 카운터(412)는 1 바이트의 데이터가 전송될 때마다 계수를 1씩 줄여나가서, 논리 길이가 모두 계수되면 0에 도달한다. 물리 길이 카운터(414)는 1 바이트의 데이터가 전송될 때마다 계수를 1씩 줄여나가, 물리 길이가 모두 계수되면 0에 도달한다. 마찬가지로, 섹터 레지스터/카운터(408)도 각 섹터의 데이터가 전송될 때마다 계수를 1씩 줄여나가 모든 섹터가 전송 완료되면 0에 도달한다. 0 검출기(426, 428, 430)는 논리 길이 카운터(412), 물리 길이 카운터(414), 섹터 레지스터/카운터(408)의 계수를 각각 감시하여, 각각의 카운터의 계수가 0에 도달하면, 이 사실을 제어 로직(424)에게 통지한다.

제어 로직(424)은 물리 길이 카운터(414)와 섹터 레지스터/카운터(408)의 상태를 각각의 0 검출기(428, 430)를 통해 간접적으로 감시함으로써, 다중 섹터의 전송을 관리한다. 물리 길이 카운터(414)가 0에 도달하면, 하나의 섹터가 전부 전송된 것이다. 그 응답으로, 제어 로직(424)은 인에이블 신호(416)를 통해 섹터 레지스터/카운터(408)의 계수를 1 감소시킨다. 섹터 레지스터/카운터(408)가 아직 0으로 감소되지 않았다면, 더 많은 섹터들이 전송되어야 하며, 로드 신호(418)를 통해 제어 로직(424)은 논리 길이 섹터 레지스터(404)로부터 논리 길이 카운터(412)를 재로딩(reload)하고, 물리 섹터 레지스터(406)로부터 물리 길이 카운터(414)를 재로딩한다.

패딩 모드에서, 제어 로직(424)은 버퍼(411)를 통해 접속되어 있는 데이터 대기 신호(431)를 관리하여, 프로세서로부터 논리 데이터 스트림의 전송을 제어한다. 처음에는 데이터 대기 신호(431)가 설정되지 않는다. 그러므로, 프로세서가 논리 데이터 스트림을 버퍼(446)로 전송하는 것이 허가되고, 데이터는 데이터 래치(448)에 래치된다. 논리 데이터 스트림은 데이터 래치(448)로부터 패드/데이터 다중화기(450)의 입력으로 전송된다.

제어 로직(424)은 0 검출기(426)를 통해 논리 길이 카운터(412)를 간접적으로 감시하고, 논리 길이 카운터(412)가 0이 될 때까지 데이터 래치(448)로부터의 논리 데이터 스트림을 선택하도록 선택 신호(436)를 통해 패드/데이터 다중화기(450)를 제어한다. 그 이후, 제어 로직(424)은 0 검출기(428)를 통해 물리 길이 카운터(414)를 간접적으로 감시한다. 제어 로직(424)은 데이터 대기 신호(431)를 발행함으로써 프로세서로부터의 데이터 전송을 금지하고, 선택 신호(436)를 통해 패드/데이터 다중화기(450)를 제어함으로써 이 패드/데이터 다중화기가 물리 길이 카운터(414)가 0에 도달할 때까지 패드값 레지스터(410)로부터 패드값을 선택하도록 만든다. 패드/데이터 다중화기(450)의 출력은 물리 데이터 스트림인데, 다시 말해서, 논리 섹터에 패딩 데이터가 부가되어 이루어진 데이터 스트림으로서, 논리 섹터와 그의 패딩 데이터가 조합된 길이는 물리 섹터의 길이이기도 하다.

패드/데이터 다중화기(450)의 물리 데이터 스트림 출력은 데이터 래치(452)로 송출되고, 그 이후에 메모리 인터페이스(454)로 송출된다. 메모리 제어 신호(444)를 통해 제어 로직(424)은 메모리 인터페이스(454)에게 물리 데이터 스트림을 디스크 드라이브 버퍼에게 보낼 것을 지시한다.

추출 모드에서는, 물리 데이터 스트림이 메모리 인터페이스(454)를 통해 디스크 드라이브 버퍼로부터 수신되어, 데이터 래치(460)에 일시 저장된다. 다음, 물리 데이터 스트림은 데이터 래치(460)로부터 대이

터 래치(458)로 송출된다.

제어 로직(424)은 0 검출기(426)를 통해 논리 길이 카운터(412)를 간접적으로 감시하고, 논리 길이 카운터(412)가 0이 될 때까지 인에이블 신호(432)를 통해 데이터 래치(460)로부터 데이터를 입력하도록 데이터 래치(458)를 제어한다. 그 이후에, 제어 로직(424)은 0 검출기(428)를 통해 간접적으로 물리 길이 카운터(414)를 감시하고, 선택 신호(436)를 통해 물리 길이 카운터(414)가 0에 도달할 때까지 데이터 래치(458)를 동작 불능 상태로 만든다. 데이터 래치(458)의 출력은 논리 데이터 스트림인데, 다시 말해서, 물리 섹터가 갖고 있는 패딩값을 억제시킨 이 물리 섹터로부터 추출한 논리 섹터로 이루어진 데이터 스트림이다.

데이터 래치(458)의 논리 데이터 스트림 출력은 버퍼(456)를 통해 프로세서로 송출된다. 제어 로직(424)은 데이터 래치(458)의 출력이 유효한 논리 데이터 스트림을 전송하는 시간 동안에 데이터 유효 신호(438)를 발생한다. 데이터 유효 신호(438)는 버퍼(462)를 통해 접속되어 전송 데이터 유효 신호(461)로서 프로세서에게 전송된다.

도 5는 논리 섹터에 패딩값을 추가하는 하드웨어 카운터의 이용을 도시한다. 이것은 2054 바이트 길이의 논리 섹터가 프로세서로부터 수신되고 2056 바이트 길이의 물리 섹터가 디스크 드라이브로 전송되는 경우를 도시한다. 논리 길이 카운터(도 4의 도면 부호(412) 참조)와 물리 길이 카운터(도 4의 도면 부호(414) 참조)는 각각 전송중인 데이터 스트림의 각 바이트를 1(one count)로 간주하여 0이 될 때까지 거꾸로 계수(counting down)한다. 단계(505)에서, 논리 길이 카운터(412)는 0이 되어, 하나의 논리 섹터를 계수 완료했음을 나타낸다. 물리 길이 카운터(414)는 카운터의 계수가 0이 되는 단계(515)가 될 때까지 계수를 계속한다. 2 바이트의 패딩 데이터가 데이터 스트림에 추가되어 있음을 유의하자.

단계(520)에서, 섹터 레지스터/카운터(도 4의 도면 부호(408) 참조)는 그 값이 1 감소하고, 논리 길이 카운터(412)와 물리 길이 카운터(414)는 다시 로딩된 뒤, 후속 섹터가 전송되는 동안 다시 거꾸로 계수될 수 있게 된다. 당업자라면, 카운터가 최종적으로 0의 값으로 감소하므로, 카운터에는 실제 계수될 바이트 수보다 101 작은 값이 로딩됨을 이해할 것이다.

도 6은 본 발명에 따른 하드웨어로 물리 섹터로부터 논리 섹터를 추출하기 위한 카운터의 이용을 도시한다. 2056 바이트 길이의 물리 섹터가 디스크드라이브로부터 수신되어, 2054 바이트 길이의 논리 섹터가 프로세서로 전송된다. 논리길이 카운터(도 4의 도면 부호(412) 참조)와 물리 길이 카운터(도 4의 도면 부호(414) 참조)는 각각 전송중인 데이터 스트림의 각 바이트를 1(one count)로 간주하여 0이 될 때까지 거꾸로 센다. 전송 데이터 유효 신호(도 4의 도면 부호(461) 참조)는 유효한 데이터가 프로세서로 전송되는 동안에는 논리 "1"이다. 단계(605)에서, 논리 길이 카운터(412)는 0에 도달하여, 하나의 논리 섹터가 전송되었음을 나타낸다. 단계(610)에서, 데이터 스트림의 패딩값이 디스크 드라이브로부터 수신되는 동안 전송 데이터 유효 신호(461)는 논리 "0"으로 변한다. 물리 길이 카운터(414)는 그 값이 0이 되는 단계(615)까지 계속해서 거꾸로 계수한다.

단계(620)에서, 섹터 레지스터/카운터(도 4의 도면 부호(408) 참조)는 그 값이 1 감소하고, 논리 길이 카운터(412)와 물리 길이 카운터(414)는 다시 로딩된 뒤, 후속 섹터가 전송되는 동안 다시 거꾸로 계수될 수 있게 된다. 전송 유효 데이터 신호(461)는 다시 한번 더 유효한 데이터가 전송되는 것을 나타내기 위해 논리 "1"로 변경된다.

전술한 설명은 단지 본 발명을 예시한 것임을 이해해야 한다. 본 발명을 벗어나지 않고도 당업자에 의해 다양한 대체나 수정이 유도될 수 있다. 예를 들면, 데이터 래치나 버퍼가 회로를 통해 전송되는 데이터의 순서를 변경하도록 구성될 수도 있고, 여러 가지 제어 신호의 타이밍 관계가 수정될 수도 있다. 또한, 전송중인 데이터 스트림이 바이트가 아닌 다른 데이터 단위일 수도 있고, 데이터 전송선이 직렬링크나 병렬 버스로 구성될 수도 있다. 따라서, 본 발명은 첨부된 특허청구범위의 범주에 속하는 이러한 대체, 수정 및 변형을 모두 포함하도록 의도된다.

발명의 효과

따라서, 본 발명에 따르면, 물리 섹터의 경계에 대해 논리 섹터의 경계를 정렬하기 위해 논리 섹터에 대해 패딩 데이터를 추가/추출하되, 이것을 하드웨어로 구현함으로써, 기록 밀도(recording density)와 처리량(throughput)을 개선할 수 있게 된다.

(57) 청구의 범위

청구항 1

논리 길이(logical length)를 갖는 제 1 데이터 스트림에 패딩값(padding value)을 추가하여, 제 2 데이터 스트림—상기 제 2 데이터 스트림은 기록 장치에 기록될 때 상기 기록 장치상에서 상기 기록 장치의 동작 요구에 맞는 정해진 물리 길이(determined physical length)를 점유하고, 상기 논리 길이와 상기 물리 길이는 데이터 단위(data units)로 측정됨—를 발생시키는 회로에 있어서,

① 상기 제 1 데이터 스트림을 수신하는 제 1 입력과, 상기 패딩값을 수신하는 제 2 입력과, 상기 제 2 데이터 스트림을 전송하는 출력을 구비하며, 상기 제 1 입력과 상기 제 2 입력 중 어느 하나를 상기 출력으로 송출하는 것을 제어할 수 있는 스위치(switch)와,

② 상기 제 1 데이터 스트림의 데이터 단위를 계수(count)하며, 상기 논리 길이가 계수 완료된 시점을 나타내는 출력을 구비하는 논리 길이 카운터(logical length counter)와,

③ 상기 제 2 데이터 스트림의 데이터 단위를 계수(count)하며, 상기 물리 길이가 계수 완료된 시점을 나타내는 출력을 구비하는 물리 길이 카운터(physical length counter)와,

④ 상기 논리 길이 카운터와 상기 물리 길이 카운터로부터의 상기 출력에 응답하여, 상기 논리 길이가 계수 완료될 때까지 상기 제 1 데이터 스트림을 선택하도록 상기 스위치를 제어한 뒤, 상기 물리 길이가 계수 완료될 때까지 상기 패딩값을 선택하도록 상기 스위치를 제어하는 콘트롤러(controller)를 포함하는

패딩값 추가 회로.

청구항 2

제 1 항에 있어서,

상기 스위치의 상기 출력으로부터 상기 제 2 데이터 스트림을 수신하는 입력과, 상기 제 2 데이터 스트림을 전송하는 출력을 구비하며, 상기 제 2 데이터 스트림을 수신하는 상기 입력을 동작 가능 상태(enable)나 동작 불능 상태(disable)로 제어할 수 있는 데이터 래치(data latch)를 더 포함하는 패딩값 추가 회로.

청구항 3

제 1 항에 있어서,

상기 제 2 데이터 스트림의 섹터를 계수(count)하며, 상기 섹터가 계수 완료된 시점을 나타내는 출력을 구비하는 섹터 카운터(sector counter)를 더 포함하는 패딩값 추가 회로.

청구항 4

제 1 항에 있어서,

상기 논리 길이를 표시하는 값을 보유하는 논리 길이 레지스터(logical length register)를 더 포함하여, 상기 논리 길이 카운터는 상기 논리 길이 레지스터로부터 로딩(load)되는 패딩값 추가 회로.

청구항 5

제 1 항에 있어서,

상기 물리 길이를 표시하는 값을 보유하는 물리 길이 레지스터(physical length register)를 더 포함하여, 상기 물리 길이 카운터는 상기 물리 길이 레지스터로부터 로딩(load)되는 패딩값 추가 회로.

청구항 6

제 1 항에 있어서,

상기 패딩값을 보유하는 패딩값 레지스터(padding value register)를 더 포함하여, 상기 스위치의 상기 제 2 입력이 상기 패딩값 레지스터로부터 상기 패딩값을 수신하는 패딩값 추가 회로.

청구항 7

제 1 항에 있어서,

상기 데이터 단위는 바이트(byte)인 패딩값 추가 회로.

청구항 8

기록 장치의 동작 요구에 맞는 정해진 물리 길이를 갖는 제 2 데이터 스트림으로부터 논리 길이를 갖는 제 1 데이터 스트림—상기 논리 길이와 상기 물리 길이는 데이터 단위로 측정됨—을 추출하는 회로에 있어서,

① 상기 제 2 데이터 스트림을 수신하는 입력과, 상기 제 1 데이터 스트림을 전송하는 출력을 구비하며, 상기 제 2 데이터 스트림을 수신하는 상기 입력을 동작 가능 상태(enable) 또는 동작 불능 상태(disable)로 제어할 수 있는 데이터 래치(data latch)와,

② 상기 제 1 데이터 스트림의 상기 데이터 단위를 계수(count)하며, 상기 논리 길이가 계수 완료된 시점을 나타내는 출력을 구비하는 논리 길이 카운터(logical length counter)와,

③ 상기 제 2 데이터 스트림의 상기 데이터 단위를 계수(count)하며, 상기 물리 길이가 계수 완료된 시점을 나타내는 출력을 구비하는 물리 길이 카운터(physical length counter)와,

④ 상기 논리 길이 카운터와 상기 물리 길이 카운터로부터의 상기 출력에 응답하여, 상기 논리 길이가 계수 완료될 때까지 상기 제 2 데이터 스트림을 수신하는 상기 입력을 동작 가능 상태로 만들도록 상기 데이터 래치를 제어한 뒤, 상기 물리 길이가 계수 완료될 때까지 상기 제 2 데이터 스트림을 수신하는 상기 입력을 동작 불능 상태로 만들도록 상기 데이터 래치를 제어하는 콘트롤러(controller)를 포함하는 논리 길이를 갖는 데이터 스트림을 추출하는 회로.

청구항 9

제 8 항에 있어서,

상기 제 2 데이터 스트림의 섹터를 계수(count)하며, 상기 섹터가 계수 완료된 시점을 나타내는 출력을 구비하는 섹터 카운터(sector counter)를 더 포함하는 논리 길이를 갖는 데이터 스트림을 추출하는 회로.

청구항 10

제 8 항에 있어서,

상기 논리 길이를 표시하는 값을 보유하는 논리 길이 레지스터(logical length register)를 더 포함하여, 상기 논리 길이 카운터가 상기 논리 길이 레지스터로부터 로딩(load)되는 논리 길이를 갖는 데이터 스트림을 추출하는 회로.

청구항 11

제 8 항에 있어서,

상기 물리 길이를 표시하는 값을 보유하는 물리 길이 레지스터(physical length register)를 더 포함하여, 상기 물리 길이 카운터가 상기 물리 길이 레지스터로부터 로딩(load)되는 논리 길이를 갖는 데이터 스트림을 추출하는 회로.

청구항 12

제 8 항에 있어서,

상기 콘트롤러는 상기 데이터 래치의 상기 출력이 상기 제 1 데이터 스트림을 전송중임을 나타내는 출력을 제공하는 논리 길이를 갖는 데이터 스트림을 추출하는 회로.

청구항 13

제 8 항에 있어서,

상기 데이터 단위는 바이트(byte)인 논리 길이를 갖는 데이터 스트림을 추출하는 회로.

청구항 14

논리 길이(logical length)를 갖는 제 1 데이터 스트림에 패딩값(padding value)을 추가하여, 기록 장치에 기록될 때 상기 기록 장치상에서 상기 기록 장치의 동적 요구에 맞는 정해진 물리 길이(physical length)를 점유할 제 2 데이터 스트림을 발생시키는 패딩 모드(padding mode), 또는 상기 제 2 데이터 스트림으로부터 상기 제 1 데이터 스트림을 추출하는 추출 모드(extracting mode)에서 동작할 수 있는 회로—상기 논리 길이와 상기 물리 길이는 데이터 단위(data units)로 측정됨—에 있어서,

- ① 상기 제 1 데이터 스트림을 수신하는 제 1 입력과, 상기 패딩값을 수신하는 제 2 입력과, 상기 제 2 데이터 스트림을 전송하는 출력을 구비하며, 상기 제 1 입력과 상기 제 2 입력중 어느 하나를 상기 출력으로 송출하는 것을 제어할 수 있는 스위치(switch)와,
- ② 상기 제 2 데이터 스트림을 수신하는 입력과, 상기 제 1 데이터 스트림을 전송하는 출력을 구비하며, 상기 제 2 데이터 스트림을 수신하는 상기 입력을 동작 가능 상태(enable) 혹은 동작 불능 상태(disable)로 제어할 수 있는 제 1 데이터 래치(data latch)와,
- ③ 상기 제 1 데이터 스트림의 상기 데이터 단위를 계수(count)하며, 상기 논리 길이가 계수 완료된 시점을 나타내는 출력을 구비하는 논리 길이 카운터(logical length counter)와,
- ④ 상기 제 2 데이터 스트림의 상기 데이터 단위를 계수(count)하며, 상기 물리 길이가 계수 완료된 시점을 나타내는 출력을 구비하는 물리 길이 카운터(physical length counter)와,
- ⑤ 상기 논리 길이 카운터와 상기 물리 길이 카운터로부터의 상기 출력에 응답하는 콘트롤러(controller)를 포함하되,

상기 콘트롤러는,

- ⑦ 상기 패딩 모드에서, 상기 논리 길이가 계수 완료될 때까지 상기 제 1 데이터 스트림을 선택하도록 상기 스위치를 제어한 뒤, 상기 물리 길이가 계수 완료될 때까지 상기 패딩값을 선택하도록 상기 스위치를 제어하고,
- ⑧ 상기 추출 모드에서는, 상기 논리 길이가 계수 완료될 때까지 상기 제 2 데이터 스트림을 수신하는 상기 입력을 동작 가능 상태(enable)로 하도록 상기 제 1 데이터 래치를 제어한 뒤, 상기 물리 길이가 계수 완료될 때까지 상기 제 2 데이터 스트림을 수신하는 상기 입력을 동작 불능 상태(disable)로 하도록 상기 제 1 데이터 래치를 제어하는 회로.

청구항 15

제 14 항에 있어서,

상기 스위치의 상기 출력으로부터 상기 제 2 데이터 스트림을 수신하는 입력과, 상기 제 2 데이터 스트림을 전송하는 출력을 구비하며, 상기 제 2 데이터 스트림을 수신하는 상기 입력을 동작 가능 상태(enable) 혹은 동작 불능 상태(disable)로 제어할 수 있는 제 2 데이터 래치(data latch)를 더 포함하는 회로.

청구항 16

제 14 항에 있어서,

상기 제 2 데이터 스트림의 섹터를 계수(count)하며, 상기 섹터가 계수 완료된 시점을 나타내는 출력을 구비하는 섹터 카운터(sector counter)를 더 포함하는 회로.

청구항 17

제 14 항에 있어서,

상기 논리 길이를 표시하는 값을 보유하는 논리 길이 레지스터(logical length register)를 더 포함하여, 상기 논리 길이 카운터는 상기 논리 길이 레지스터로부터 로딩(load)되는 회로.

청구항 18

제 14 항에 있어서,

상기 물리 길이를 표시하는 값을 보유하는 물리 길이 레지스터(physical length register)를 더 포함하여, 상기 물리 길이 카운터는 상기 물리 길이 레지스터로부터 로딩(load)되는 회로.

청구항 19

제 14 항에 있어서,

상기 패딩값을 보유하는 패딩값 레지스터(padding value register)를 더 포함하여, 상기 스위치의 상기 제 2 입력이 상기 패딩값 레지스터로부터 상기 패딩값을 수신하는 회로.

청구항 20

제 14 항에 있어서,

상기 콘트롤러는 상기 제 1 데이터 랙치의 상기 출력이 상기 제 1 데이터 스트림을 전송종임을 나타내는 출력을 제공하는 회로.

청구항 21

제 14 항에 있어서,

상기 데이터 단위는 바이트(byte)인 회로.

도면

도면1

(종래 기술)

물리 섹터에 정렬되지 않은 논리 섹터

논리 섹터 #1	논리 섹터 #2	논리 섹터 #3	논리 섹터 #31	논리 섹터 #32
2054 바이트	2054 바이트	2054 바이트	2054 바이트	2054 바이트
2056 바이트	2056 바이트	2056 바이트	2056 바이트	2056 바이트

105, 인더럽트

도면2

(종래 기술)

소프트웨어로 논리 섹터에 패딩 데이터를 추가

논리 섹터 #1	논리 섹터 #2	논리 섹터 #3
2054 바이트	2054 바이트	2054 바이트

205

215

논리 섹터 #31	논리 섹터 #32
2054 바이트	2054 바이트

2056 바이트	2056 바이트	2056 바이트
514 514 514 514 514 514 514 514 514 514 514 514	514 514 514 514 514 514 514 514 514 514 514 514	514 514 514 514 514 514 514 514 514 514 514 514

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

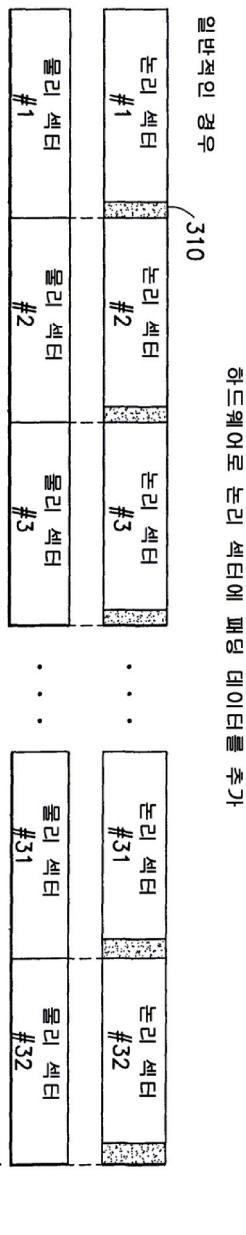
↑

↑

↑

↑

도면3a



350, 인터럽트

도면3b

예시적인 적용

논리 셋터 #1	논리 셋터 #2	논리 셋터 #3
2054 바이트	2054 바이트	2054 바이트

.. .

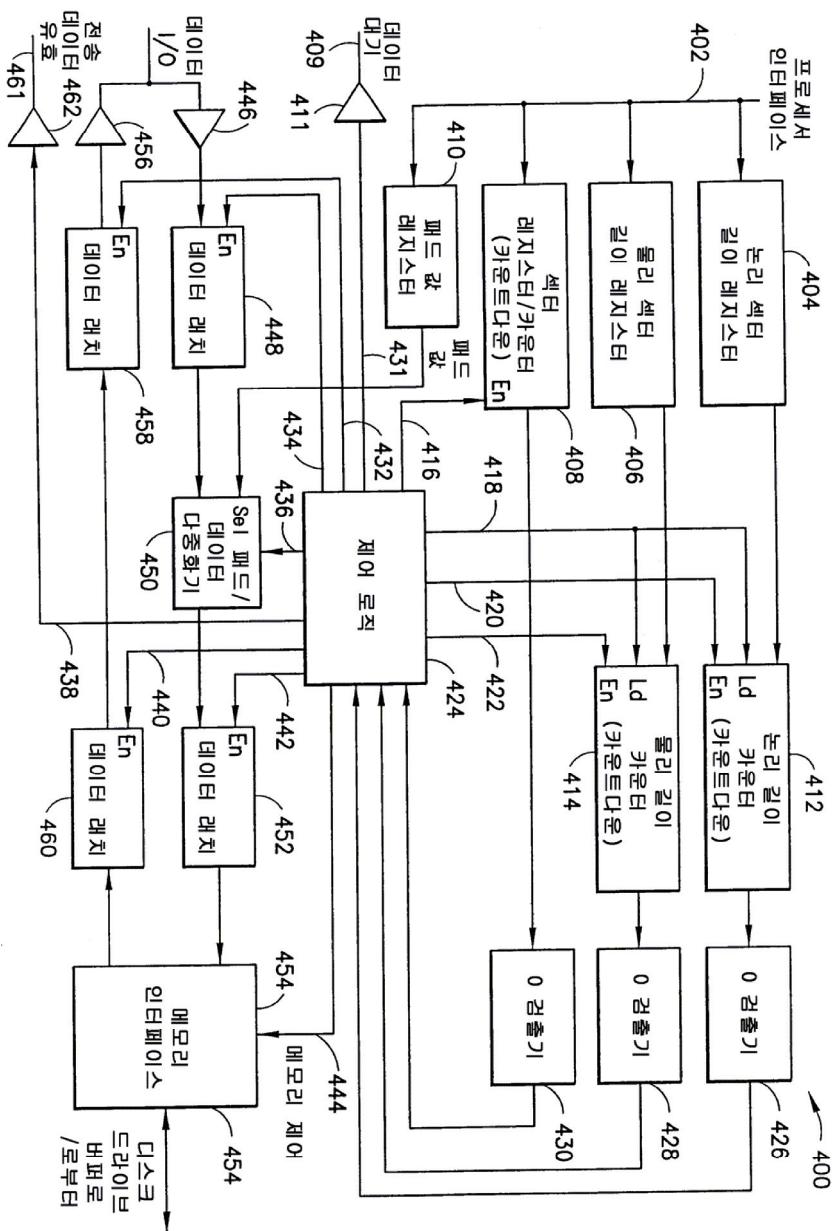
논리 셋터 #31	논리 셋터 #32
2054 바이트	2054 바이트

.. .

2056 바이트	2056 바이트	2056 바이트
514 514 514 514 514 514 514 514 514 514 514 514	514 514 514 514 514 514 514 514 514 514 514 514	514 514 514 514 514 514 514 514 514 514 514 514

350A, 인더 험트

도면4



도면5

섹터	카운트
논리	길이	카운트
물리	길이	카운트
...

논리 회로에 패딩 데이터를 추가하기 위해
하드웨어에 의한 카운터의 이용

31

30

3	2	1	0	0	2053	2052	2051
5	4	3	2	1	0	2055	2054
데이터	데이터	데이터	데이터	데이터	데이터	데이터	데이터
...

505

515

520

도면6

