

(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) Int. Cl.<sup>6</sup>  
G06F 13/00

(45) 공고일자 1999년06월 15일

(11) 등록번호 10-0194295

(24) 등록일자 1999년02월08일

(21) 출원번호	10-1996-0051982	(65) 공개번호	특1997-0062924
(22) 출원일자	1996년11월05일	(43) 공개일자	1997년09월 12일
(30) 우선권주장	8/597,092	1996년02월05일	미국(US)

(73) 특허권자 인터내셔널 비지네스 머신즈 코포레이션 포만 제프리 엘  
미국 10504 뉴욕주 아몬크 뉴오차드 로드

(72) 발명자 폴 호지스  
미국 95120 캘리포니아주 산 호세 몬테버드 드라이브 6154  
마이클 갈우드 허리  
미국 95014 캘리포니아주 쿠퍼르티노 레그나트 로드 21602  
노만 케니스 오치  
미국 95120 캘리포니아주 산 호세 뉴 크레스트 코트 20248  
미엔 시이  
미국 95070 캘리포니아주 사라토가 호웬 드라이브 13714

(74) 대리인 장수길

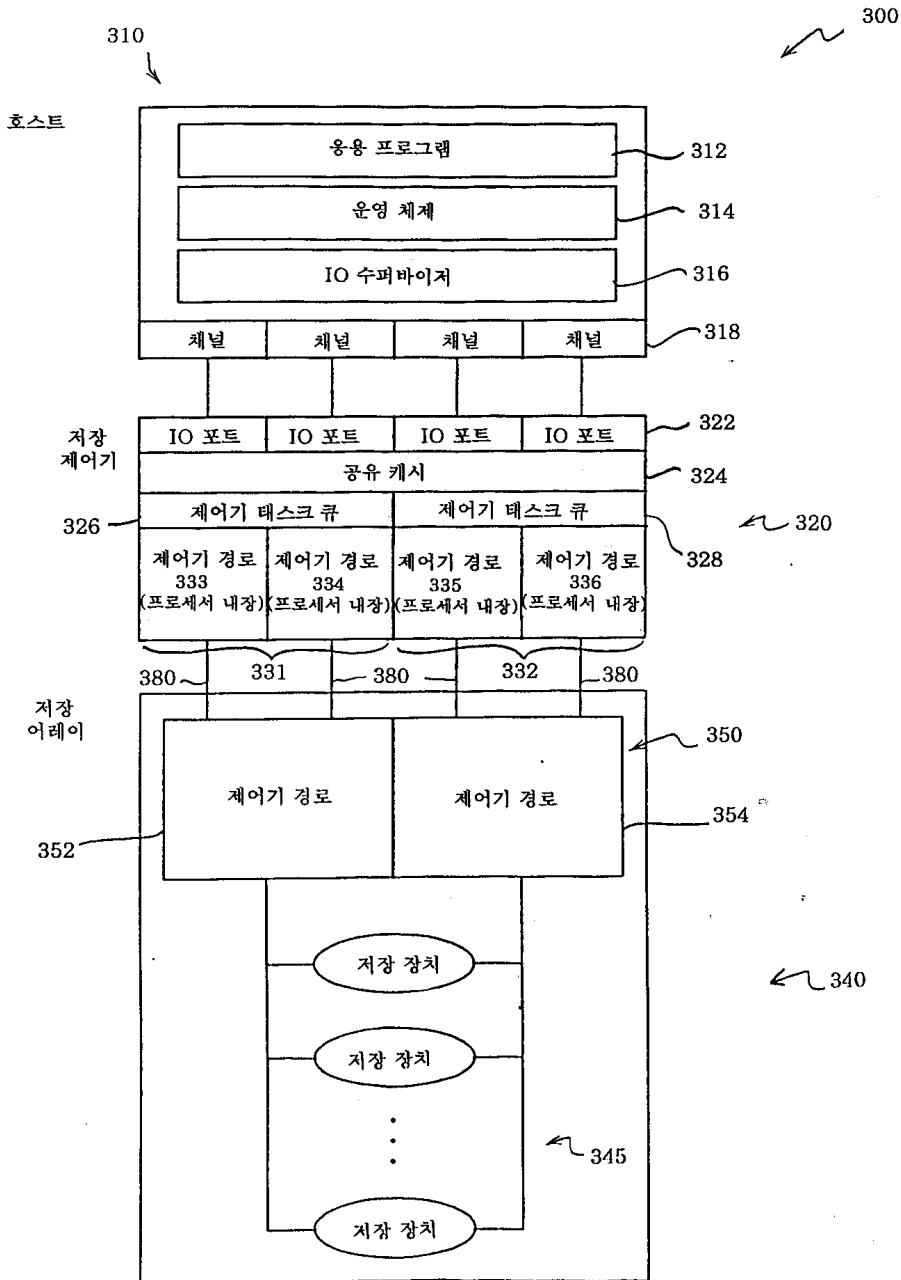
심사관 : 오홍수

(54) 데이터 처리 시스템 및 큐의 상태 갱신 방법

요약

본 발명의 데이터 처리 시스템은 호스트와 다수의 저장장치 간에 통신하기 위한 다중 독립 경로를 가지는 데, 각 경로는 저장장치를 액세스하기 위해 호스트가 발생하는 요구를 서비스하기 위하여 그 자신의 큐를 가진다. 각 요구는 모든 큐에 저장되기 전에 그의 고유 ID와 함께 고유 순차 ID가 지정된다. 각 저장장치는 수행되는 최근 요구의 ID 및 상태가 저장되는 메일박스 레지스터를 가진다. 큐가 서비스되고, 그들의 상태는 각 저장장치의 메일박스의 내용을 기반으로 갱신된다. 각 저장장치의 각 요구 및 메일박스 레지스터에 고유 태스크 ID를 지정하는 조합으로 데이터 보전성 문제의 발생, 장치 레벨에서의 요구 중복 또는 서로 동기되도록 큐를 유지하기 위한 복잡한 락킹 방안의 필요없으며 다른 경로의 같 큐와 동기되지 않아도 된다.

대표도



명세서

도면의 간단한 설명

- 제1도는 호스트가 저장 부시스템과 통신하는 데이터 처리 시스템을 도시한 도면.
- 제2도는 저장 제어기에 다수의 태스크 큐를 가지는 다른 데이터 처리 시스템을 도시하는 도면.
- 제3도는 각 장치에 태스크 큐를 가지는 또 다른 데이터 처리 시스템을 도시하는 도면.
- 제4도는 본 발명의 바람직한 실시예를 도시하는 도면.
- 제5도는 제4도에 도시된 어레이 제어기의 상세한 배선도.
- 제6도는 각 저장장치에 저장된 메일박스를 도시한 도면.
- 제7도는 본 발명의 바람직한 실시예에서 I/O 요구를 처리하는 흐름도.
- 제8도는 본 발명의 다른 바람직한 실시예를 도시한 도면.
- 제9도는 본 발명의 또 다른 바람직한 실시예를 도시한 도면.

제10도는 본 발명의 또 다른 바람직한 실시예를 도시한 도면.

\* 도면의 주요부분에 대한 부호의 설명

300 : 데이터 처리 시스템

320 : 저장 제어기

345 : 저장장치

413, 415 : 태스크큐

500 : 메일박스

## 발명의 상세한 설명

### 발명의 목적

#### 발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 다중 독립 저장 제어기와 저장장치간의 통신을 위하여 다중 독립 경로를 가지는 데이터 처리 시스템에 관한 것으로, 특히 큐들이 서로 동기되지 않아도 저장장치를 액세스하려는 요구(requests)가 수행될 수 있는 다중 독립 경로 저장 부시스템에서 큐를 효율적으로 관리하는 방법 및 수단에 관한 것이다.

다중 입력/출력(I/O) 저장 부시스템을 가지는 데이터 처리 시스템은 통상적으로 시스템내 프로세서와 각 저장장치 사이에 다중 독립 통신경로를 가진다. 제1도에는 이러한 특징을 가지는 전형적인 데이터 처리 시스템(100)이 도시되어 있다. 호스트(110)는 통상적으로 응용 프로그램(112), 운영 체제(114) 및 I/O 수퍼바이저(116)를 구비하고, 이 I/O 수퍼바이저는 호스트 발생하는 요구를 관리하는 호스트 태스크 큐(117)를 포함한다. 호스트(110)는 저장 제어기(120)와 통신하기 위하여 다수의 I/O 채널(118)을 더 구비한다. 저장 제어기(120)는 통상적으로 호스트(110)와의 통신을 위한 다수의 I/O 포트(122), 고성능을 위한 공유 캐시(124) 및 저장장치(140)를 액세스하기 위한 다수의 제어기 경로(130)를 가진다. 저장 제어기(12) 및 저장장치(140)는 통상적으로 저장 부시스템으로 불린다.

통상적으로, 호스트(110)가 발생한 I/O 요구가 캐시(124)에 미리 저장된 정보로 만족될 수 없는 경우, 저장 제어기(120)는 사용가능한 제어기 경로들(130)중의 하나를 통하여 적절한 저장장치를 액세스하여 I/O 요구를 수행할 것이다. 제1도의 데이터 처리 시스템은 통상적으로 저장 부시스템, 다중 I/O 채널, 저장 제어기에서의 다중 제어기 경로, 그리고, 저장 제어기와 저장장치 간의 다중 통신링크의 용량도로 인한 높은 가용도를 제공한다.

이러한 유형의 시스템에서, 호스트(110)가 발생하는 전형적인 I/O 요구의 큐잉은 다음과 같이 수행된다: I/O 요구는 응용 프로그램(112)에 의해 개시되어 I/O 수퍼바이저(116)로 전송된다. I/O 수퍼바이저(116)는 이 요구를 수신하여 I/O 수퍼바이저에 의해 유지관리되는 호스트 태스크 큐(117)에 이 요구를 부가한다. 통신 링크(150)중의 하나가 사용가능하게 될 때, I/O 수퍼바이저(116)는 대응한 장치가 사용가능한 큐(117)에서 첫 요구에 대한 I/O 처리를 개시한다.

이러한 유형의 시스템에서, 호스트 태스크 큐(117)가 시스템에서 유일하게 사용가능한 I/O 동작의 큐이므로, 모든 동작은 호스트(110)에서 개시되고, 단지 하나의 동작만이 소정 장치에 대하여 활성화될 수 있다. 또한, 이 동작은 I/O 수퍼바이저(116)가 이 장치에 대한 또 다른 동작을 개시할 수 있기 전에 완료된 것으로 보고되어야 한다.

이러한 유형의 구조에서, 채널(118), 통신링크(150), 장치 통신링크(160) 또는 제어기 경로(130) 중의 하나가 고장날 때 저장장치에 대한 액세스는 가능하지만, 호스트 태스크 큐(117)에 영향을 주는 고장은 호스트 태스크 큐(117)에서의 요구가 실패하도록 하여 하나 또는 그 이상의 프로그램을 중단시킬 수도 있다. 고장으로부터 복구되면 이 프로그램을 동일한 호스트 또는 상이한 호스트 시스템상에서 재실행해야 할 수도 있다.

그러나, 저장 제어기가 캐시를 가지고 있는 데이터 시스템에서, 캐시(124)와 저장장치(140) 간의 동작은 캐시(124)와 호스트(또한 중앙처리장치(CPU)로 참조됨)(11) 간의 동작과 동시에 수행되는 것이 바람직하다. 예를 들면, 호스트(110)가 발생한 저장장치(142)에 데이터를 기록하려는 요구는 데이터가 캐시(124)로 전송될 때 호스트(110)에 의해 완료된 것으로 간주될 수 있다. 그 후, 갱신된 데이터가 저장장치(142)에 기록되는 동안, 장치(142)에 대한 후속 동작이 캐시(124)로부터 수행될 수 있다.

제2도에는 저장 제어기가 다수의 태스크 큐를 가지는 데이터 시스템의 예가 도시되어 있다. 제2도는 IBM 자기 저장장치 3390[저장장치(140)]의 동작을 제어하는 IBM 저장 제어기 3990[저장 제어기(120)]과 통신하는 IBM 시스템/390[호스트(110)]을 도시한다. 이 시스템에서, 저장 제어기(120)와 저장장치(140) 간에 4개의 통신경로(데이터 경로)가 있다. 각 데이터 경로는 제어기 경로(130) 및 장치 통신링크(160)를 구비하며, 저장장치(140)중의 소정 장치상에 연산을 수행하도록 인스트럭션을 운송하는데 사용될 수 있다.

높은 가용도를 제공하기 위하여, 저장 제어기(120)는 통상적으로 두 개의 저장 부제어기(132 및 134)로 분할된다. 저장 부제어기(132)는 제어기 태스크 큐(126) 및 다수의 제어기 경로(130)(제2도에 도시된 두 제어기 경로)를 포함한다. 저장 부제어기(134)는 제어기 태스크 큐(128) 및 다수의 제어기 경로(130)(제2도에 도시된 두 제어기 경로)를 포함한다. 태스크 큐(126) 및 (128)은 서로 동일하다. 또한, 두 저장 부제어기는 전반적인 시스템 신뢰도를 향상시키도록 상이한 전력 영역에 있도록 설계된다. 따라서, 저장 부제어기들중의 하나가 고장나면, 다른 저장 부제어기가 그의 태스크 큐들로부터의 요구를 계속 수행하므로써 비록 축소된 단계이지만 연속된 수행을 제공한다.

호스트(110)가 발생한 저장장치에 대한 각 액세스 요구가 몇몇 부커맨드를 포함할 수 있고, 부커맨드 중의 하나가 하나의 부제어기에 의해 개시되고 다른 부제어기에 의해 완료될 수 있다는 것을 고려하면, 저장 제어기(120)에서 태스크 큐의 관리는 매우 중요하게 된다. 예를 들면, I/O 요구는 주로 1) 탐색/위치선정의 준비상태 커맨드와 2) 판독/기록으로 알려진 시간중속 데이터 전송 커맨드의 두 부커맨드를 포함

한다. 이러한 유형의 I/O 요구의 경우에, 저장 부제어기중의 하나는 소정시간에 부커맨드중의 하나를 속행하는데 사용될 수 있고, 규상에 다음 연산을 서비스 할 수 있다.

따라서, 판독/기록 커맨드가 부제어기중의 하나에 의해 수행되기전에, 각 저장 부제어기는 이 커맨드가 올바른 I/O 요구와 관련되도록 보장하기 위하여 각 장치에 대한 I/O 요구 상태를 계속 알아야 한다. 즉, 두 부제어기가 탐색/위치지정 부커맨드의 완료의 인식을 수신하는 메카니즘이 있어야 한다. 예를 들면, 두 부제어기가 저장장치로부터 인식을 수신할 때까지 하나의 부제어기에 의한 판독/기록 부커맨드의 수행은 지연될 수 있다. 그러나, 이러한 접근방법에서는 저장장치에 대한 액세스가 상당히 느려지며, 저장 부제어기의 독립성이 저하된다.

이러한 문제를 다루기 위한 한가지 방법은 저장 부제어기들이 높은 가용도를 보장하기 위해 독립적으로 동작하지만, 성능을 향상시키기 위해 처리 장치 액세스 요구시에 서로 상당히 밀접하게 통신하는 것이다. 이러한 구조는 제2도에 도시되어 있는데, 여기서, 저장장치를 액세스하기 위한 요구는 각 저장 부제어기에 의해 중복되어, 각 부제어기에 의해 장치로 전송된다. 이러한 구조에서, 장치를 포착하고 통신경로를 설정하는 제1 저장 부제어기는 탐색/위치선정 커맨드를 장치로 전송한다. 그 다음, 통신경로는 장치로부터 분리되고, 저장장치는 탐색/위치선정 부커맨드를 수행하기 시작한다. 일단 동작이 완료되고 장치가 데이터를 전송할 준비가 되면, 장치는 플래그 또는 인터럽트로써 저장 제어기에게 알린다. 일단, 장치가 플래그 또는 인터럽트를 발생하면, 장치로부터의 인터럽트를 감지한 제1 사용가능한 부제어기는 장치를 포착하여 장치로부터 저장 제어기(12)의 공유 캐시(124)로 데이터 전송을 완료한다.

따라서, 이러한 구조에서, 저장 부제어기는 높은 가용도를 제공하기 위해 독립적으로 동작하지만 높은 성능을 보장하기 위해 상당히 밀접하게 협동한다. 그러나, 각 저장 부제어기를 통한 통신경로는 비동기적이므로, 즉 요구가 하나의 통신경로를 통하는 것이 다른 통신경로를 통하는 것에 비해 지연될 수 있으므로, 저장 부제어기가 다른 저장 부제어기에 의해 이미 완료된 요구 또는 부커맨드를 수행하게 되는 결과를 일으키기 쉽다. 이것은 동작을 소비시켜 전반적인 저장 부시스템 성능을 저하시키기 쉽고 동시에 데이터 보전성 문제를 일으킬 수 있다.

부제어기에게 다른 부제어기에 의해 수행되는 동작을 알리기 위해 저장 부제어기들간의 제2도의 시스템에 고속 메시지 전송 구조가 사용될 수 있다. 하나의 부제어기로부터 다른 부제어기로의 메시지 전송이 지연되어 요구가 중복될 수 있고 데이터 보전성 문제를 일으킬 수 있다.

또한, 다수의 통신경로가 저장 제어기(120)와 저장장치(140) 사이에 존재하고 사용가능한 통신경로중의 하나가 태스크 큐(126, 128) 중의 하나로부터의 다음 I/O 요구를 서비스할 수 있으므로, 요구의 두 카피는 동일한 I/O 요구를 두 번 수행하지 않도록 항상 동일하게 유지되어야 한다. 이것은 작동할 준비가 된 부제어기가 큐의 두 카피에 배타적인 액세스를 가지며 큐의 두 카피는 록이 해제 되기전에 갱신되도록 보장하기 위해 정밀하고 복잡한 록킹(locking) 방안이 사용되어야 한다는 것을 의미한다.

따라서, 저장 제어기(120)의 두 저장 부제어기 하드웨어 고장에 영향을 받지 않지만, 두 부제어기들간의 통신은 큐의 보전성을 보장하는 데 상당히 중요하다. 복잡하고 정교한 록킹 방안 및 저장 제어기의 부제어기들간의 고속 통신에 의존하므로써 시스템 성능이 서서히 저하되며 성능 장애가 발생할 수 있다.

큐들간의 복잡한 록킹 방안에 의존하는 현상 및 장애를 제거하는 한 가지 방법은 저장 제어기(120)를 제거하고 필요한 기능을 각 저장장치로 이동시키는 것이다. 제3도에 이러한 구조가 도시되어 있는데, 다수의 프로세서(210) 및 (212)는 소형 컴퓨터 시스템 인터페이스(SCSI) 버스(218)를 통하여 다중 저장장치(220)와 통신한다. 이러한 구조에서, 각 저장장치(220)는 장치 제어기(224) 및 저장매체(222)를 포함한다. 각 SCSI 장치 제어기(224)는 그 자신의 장치 태스크 큐(226)를 더 포함한다.

장치 레벨 그 자체에서 저장장치에 대한 작업 큐를 유지관리하는 것은 큐잉관점으로 볼 때 효율적이지만, 이러한 장치는 부가적인 하드웨어 및 소프트웨어를 필요로 하고 정교한 장치 제어기를 가져야 한다. 정교한 장치 제어기를 가지는 장치(SCSI 장치)를 사용할 수도 있지만, 저장 제어기에서 다중 큐의 관리와 관련된 문제가 없는, 저장 제어기 또는 RAID 제어기 또는 네트워크 부착 데이터 서버에 직접 연결될 수 있는 장치 제어기를 가지지 않는 간단한 저장장치를 가지는 것이 바람직하다.

따라서, 다수의 독립 I/O 통신(데이터) 경로를 통하여 저장 제어기와 통신하는 다수의 저장장치를 가지는 저장 부시스템에서, 다중 태스크 큐에서 유지관리되는 I/O 요구를 조정하고, 저장 부제어기의 태스크 큐들을 서로 동기시킬 필요도 없고, 저장 부제어기들간에 각 저장장치의 연속된 상태 및 요구를 제공할 필요도 없고, 큐들을 서로 동기시키기 위해 부제어기들간에 고속 통신을 할 필요성도 없는 독립된 통신경로들간에 완료된 동작을 삭제시키는 방법 및 수단이 필요하다.

또한, 다수의 독립된 I/O 통신(데이터) 경로를 통하여 저장장치와 통신하는 다수의 독립된 저장 제어기를 가지는 데이터 처리 시스템에 있어서, 제어기의 다중 태스크 큐에서 유지관리되는 I/O 요구를 처리하고, 서로 완전히 동기되는 제어기에서 태스크 큐들을 서로 동기시킬 필요도 없고, 저장 제어기들간에 각 저장장치의 연속된 상태 및 요구를 제공할 필요도 없고, 큐들이 서로 동기시키기 위해 저장 제어기들간에 고속 통신을 할 필요도 없는 독립된 통신경로들간의 완료된 동작을 삭제시키는 수단 및 방법이 필요하다.

### **발명이 이루고자 하는 기술적 과제**

본 발명의 목적은 저장 제어기와 저장장치 간의 통신을 위하여 다중 독립 경로를 가지는 고성능 저장 부시스템을 제공하는 데 있다.

본 발명의 다른 목적은 저장 제어기와 저장장치 간의 통신을 위하여 다중 독립 경로를 가지는 높은 가용도의 저장 부시스템을 제공하는 데 있다.

본 발명의 또 다른 목적은 각 부제어기의 큐가 다른 부제어기의 큐와 동기될 필요가 없는 다수의 저장 부제어기를 가지는 저장 부시스템을 제공하는데 있다.

본 발명의 또 다른 목적은 각 저장 제어기의 저장 부제어기가 각 저장장치의 레지스터를 통하여 서로 통

신하는 저장 부시스템을 제공하는 데 있다.

본 발명의 또 다른 목적은 큐들이 서로 동기되도록 보장하기 위하여 큐상에 복잡한 록킹 방안이 필요없는 저장 부시스템을 제공하는 데 있다.

본 발명의 또 다른 목적은 큐 보전성을 유지관리하기 위해 저장 부제어기들간에 고속 통신이 필요없는 저장 부시스템을 제공하는 데 있다.

본 발명의 또 다른 목적은 멀티-커맨드 인스트럭션은 각 저장 부제어기의 큐를 다른 저장 부제어기의 큐와 동기시킬 필요없이, 하나의 부제어기에 의해 개시될 수 있고 다른 부제어기에 의해 완료될 수 있는 저장 부시스템을 제공하는데 있다.

본 발명의 또 다른 목적은 다수의 독립된 저장 제어기와 저장장치 간에 통신을 위한 다중 독립경로를 가지는 고성능 데이터 처리 시스템을 제공하는 데 있다.

본 발명의 또 다른 목적은 각 제어기의 큐가 다른 제어기의 큐와 동기될 필요가 없는 다중 저장 제어기를 가지는 데이터 처리 시스템을 제공하는 데 있다.

본 발명의 또 다른 목적은 저장 제어기들이 각 저장장치의 레지스터를 통하여 서로 통신하는 데이터 처리 시스템을 제공하는 데 있다.

본 발명의 또 다른 목적은 멀티-커맨드 인스트럭션이 각 저장 제어기의 큐가 다른 저장 제어기의 큐와 동기될 필요없이, 하나의 저장 제어기에 의해 개시될 수 있고 또 다른 저장 제어기에 의해 완료될 수 있는 데이터 처리 시스템을 제공하는 데 있다.

본 발명의 목적을 위해, 저장 제어기는 다수의 저장 부제어기를 포함하고, 각 저장 부제어기는 태스크 큐(큐)를 포함하는 저장 제어기와 저장장치 간의 다중 독립 I/O 경로를 가지는 데이터 처리 시스템에서, 각 큐의 상태 및 완료된 또는 진행중인 I/O 요구(작업)는 각 저장장치에서 유지관리되는 레지스터를 통하여 다른 큐로 전송되는 방법 및 수단이 개시된다. 이것은

(1) 요구가 큐에 저장되기 전에 각 I/O 요구에 고유 식별자를 배정한다. 큐에 저장된 각 I/O 요구에 대한 고유 식별자는 그 요구가 저장 부제어기들중의 하나를 통하여 완료될 때 I/O 요구가 큐의 모든 카피로부터 제거되도록 보장할 것이다 ;

(2) 그 장치와 통신경로를 설치하는 부제어기에 의해 판독 및/또는 기록될 수 있는 각 저장장치에 메일박스(mailbox)로서 알려진 레지스터를 유지관리한다. 메일박스는 고유 식별자 및 그 장치에 의해 수행되는 최근의 I/O 요구 또는 I/O 요구들의 상태를 포함한다.

(3) 저장 부제어기중의 하나가 메일박스의 내용을 판독하고 메일박스의 내용에 의존하는 적절한 활동을 취한다.

장치의 메일박스를 판독하는 저장 부제어기에 의해 취해진 적절한 행동은 저장장치의 상태를 변경시키고, 이 상태의 변경을 반영하도록 메일박스 내용을 갱신시킬 수 있다. 또한, 메일박스를 판독하는 부제어기는 판독된 내용을 기본으로 큐의 복사를 갱신시킬 수 있다.

따라서, 개시된 발명을 기반으로 큐의 복사본은 서로 동기될 필요가 없다. 큐들은 작업의 중복, 데이터 보전성 문제 또는 I/O 요구의 처리를 저하시키는 일 없이 서로 동기되지 않아도 된다.

예를 들면, 호스트가 저장장치 A로부터 데이터를 판독하려는 요구를 발생하는 경우, 요구는 태스크 큐에 저장되기 전에 저장 제어기는 고유 라벨 x를 배정한다. 일단 장치 A가 사용가능하게 되면, 장치 A를 액세스하려는 요구는 모든 사용 가능한 통신경로를 통하여 디스패치된다. 경로1이 장치 A를 제일 먼저 포착하는 경우, 탐색/위치지정 커맨드를 발생하여 데이터 전송을 위한 장치를 준비한다. 그 다음, 장치 A는 그의 메일박스에 고유 라벨 x를 저장하고, 데이터 전송을 위해 준비하기 위해 커맨드를 발생했음을 알리기 위해 I/O 동작 상태를 설정한다. 그 다음, 장치 A가 데이터 전송을 위해 그 자신을 준비하는 동안, 경로1은 장치 A를 해제시키고, 다른 동작을 수행한다. 경로2가 장치 A가 데이터 전송을 할 준비가 되었음을 감지하는 첫 경로라고 가정하면, 경로2는 장치 A의 메일박스를 판독하여 고유라벨 x를 가지는 요구가 진행 중인지를 알아내고, 그의 태스크 큐에서 x 라벨의 요구와 데이터 전송을 관련시키므로써 데이터 전송을 수행한다. 일단 데이터 전송이 완료되면, 장치 A에서 특정 I/O 요구의 상태는 고유 라벨 x와 관련된 요구가 수행되었음을 알리기 위해 완료로 설정된다. 이제, 경로1이 장치 A를 판독한다고 가정하면, 일단 장치 A의 메일박스를 판독하면, 고유 라벨 x와 관련된 요구는 이미 다른 경로를 통해 완료되었음을 알게된다. 따라서, 경로1은 그 자신의 태스크 큐로부터 고유 라벨 x와 관련된 요구를 삭제시키고, 장치 A를 해제시킨다.

각 저장장치의 메일 박스와 함께 태스크 큐에 저장된 각 I/O 요구에 대한 고유 라벨을 사용하면 동일한 요구가 저장 부제어기에 의해 한 번 이상 수행되는 것을 방지할 수 있다. 동시에, 저장 시스템의 성능에 영향을 주지 않고 태스크 큐가 비동기적으로 갱신될 수 있게 한다. 또한, 각 저장장치의 메일박스와 함께 각 I/O 요구에 대한 고유 라벨을 사용하면 각 저장 부제어기의 태스크 큐가 다른 저장 부제어기의 태스크 큐와 동기됨을 보장하기 위한 정교하고 복잡한 록킹 단계 방안이 필요없게 된다. 본 발명은 또한, 태스크 큐들이 서로 동기될 필요가 없으므로 큐의 상태를 신속하고 연속적으로 전송하기 위한 저장 부제어기들간의 고속 직통신이 필요없게 된다.

### **발명의 구성 및 작용**

본 발명의 특성 및 장점들과 바람직한 사용 형태는 첨부된 도면과 함께 다음의 상세한 설명을 참조하여 명백히 알 수 있을 것이다.

다음의 설명은 본 발명을 수행하기 위해 현재 고려되는 최상의 형태이다. 도시된 실시예의 설명 및 참조 번호는 본 발명의 원리를 설명하기 위한 것이지만 본 발명을 제한하려는 것은 아니다.

제4도를 참조하면, 본 발명의 바람직한 실시예가 도시되어 있다. 데이터 처리 시스템(300)은 저장 제어기(320)와 통신하는 호스트(310)를 구비하고, 저장 제어기는 저장 어레이(340)와 통신한다. 저장 어레이(340)는 저장장치(345)에 대한 액세스를 제어하는 어레이 제어기(350)를 포함한다. 본 실시예에서, 통상적으로 저장 제어기(320)에 부착되는 저장장치는 저장 어레이(340)로 대체된다. 그러면, 저장 어레이(340)는 논리적 IBM 3390 저장장치로 취급될 수 있다. 즉, 저장 어레이(340)는 IBM 3390 저장 부시스템과 같은 기존의 저장장치의 에뮬레이션을 제공함으로써 저장 제어기(320)에 표준 저장장치를 부착시킬 수 있다. 또한, 저장장치(345)와 결합한 어레이 제어기(350)는 신뢰할만한 저가의 디스크 어레이(RAID : reliable arrays of inexpensive and cheap disks)와 같은 구조를 통하여 상당히 높은 데이터 가용도를 제공할 수 있다.

제4도를 다시 참조하면, 호스트(310)는 응용 프로그램(312), 운영체제(314), I/O 슈퍼바이저(316) 및 저장 제어기(320)와 통신하기 위한 다수의 I/O 채널(318)을 구비한다. 저장제어기(320)는 통상적으로 호스트(310)와 통신하기 위한 다수의 I/O 포트(322), 고성능의 공유 캐시(324) 및 저장 어레이(340)를 액세스하기 위한 두 저장 부제어기(331, 332)를 포함한다. 저장 부제어기(331)는 제어기 태스크 큐(326) 및 제어기 경로(333, 334)를 포함한다. 저장 부제어기(332)는 제어기 태스크 큐(328) 및 제어기 경로(335, 336)를 포함한다. 저장 제어기(320)는 다수의 통신선(380)을 통하여 저장 어레이(340)와 통신한다.

제5도는 저장장치(345) 및 저장 제어기(320)와 통신하는 어레이 제어기(350)의 상세한 배선도이다. 어레이 제어기(350)는 두 개의 동일한 독립적으로 동작하는 제어기 경로(352) 및 (354)를 구비한다. 제어기 경로(352) 및 (354)는 함께 향상된 성능 및 고장 방지 동작을 제공한다. 바람직한 실시예에서, 어레이 제어기(350)는 다수의 통신링크(380)(본 예에서 4개가 도시됨)를 통하여 저장 제어기(320)와 통신한다. 어레이 제어기(350)는 다수의 저장 인터페이스 어댑터(400)(본 예에서는 모두 4개로 각 제어기 경로당 2개로 도시됨)를 더 구비한다. 저장 인터페이스 어댑터(400)는 저장 제어기(320)로부터 저장장치(345)를 액세스하려는 커맨드(인스트럭션, 요구)를 수신하고, 먼저, 저장 제어기(320)가 발생한 요구가 버퍼(416) 또는 버퍼(418)로부터 만족될 수 있는지의 여부를 판정한다. 버퍼(416) 및 (418)는 장치 및 저장 제어기로부터 수신한 데이터를 저장한다. 요구가 각 버퍼의 내용으로부터 만족될 수 있는 경우, 이 정보는 저장 인터페이스 어댑터(400)중의 하나를 통하여 저장 제어기(320)로 되전송된다. 각 요구가 한 버퍼의 내용으로부터 만족될 수 없는 경우, 하나의 저장 인터페이스 어댑터는 고유 요구 식별자(또한 라벨로 참조)를 생성하고, 고유 식별자를 가지는 요구를 어레이 관리기(412) 및 (414)로 전송한다. 어레이 관리기(412, 414)는 차례로 요구 및 그의 라벨을 그들 제각기의 큐(413, 415)에 부가시킨다.

어레이 관리기(412, 414)는 통상적으로 저장 제어기(320)와 저장장치(435) 사이에 데이터 처리를 관리한다. 저장장치(345) 포맷(원래의 포맷)과 에뮬레이션된 장치 포맷(저장 제어기(320)에 의해 보여지는 포맷) 간에 데이터 포맷을 변환한다. 예를 들면, 데이터는 고정 블록 포맷(원래의 장치 포맷)으로 저장장치(345)상에 저장될 수 있다. 다른 한편으로, 저장 제어기(320)는 카운트-키-데이터 포맷의 정보를 처리하도록 설정될 수 있다. 따라서, 어레이 관리기(412, 414)는 이들 두 상이한 형태의 데이터 포맷들간의 변환을 제공한다. 각 어레이 관리기는 또한 그 자신의 태스크 큐를 유지관리한다. 예를 들면, 어레이 관리기(412)는 어레이 태스크 큐(413)를 유지관리하고, 어레이 관리기(414)는 어레이 태스크 큐(415)를 유지관리한다. 각 어레이 관리기는 저장 인터페이스 어댑터(400)중의 하나로부터 수신한 메시지 및 각 저장장치의 메일박스의 내용을 기반으로 그 자신의 어레이 태스크 큐를 갱신한다. 또한, 어레이 관리기(412, 414)는 제각기 장치 인터페이스 어댑터(440, 442)를 통하여 저장장치(345)에 장치 동작을 디스패칭하는 것을 더 관리한다.

바람직한 실시예에서, 각 어레이 제어기 경로(352, 354)는 4개의 장치 인터페이스 어댑터를 포함한다. 어레이 관리기(412, 414)는 장치 또는 데이터를 액세스하려는 요구가 가능하게 될 때 제각기 장치 인터페이스 어댑터(440, 442)를 이를 통지한다. 어레이 관리기(412, 414)는 또한 통신 버스(410)를 통하여 서로 통신하여 그들 제각기의 어레이 제어기 경로(352, 354)의 동작에 영향을 줄 수 있는 사건을 서로에게 알릴 수 있다. 어레이 관리기(412, 414)는 RAID 배터적 OR(XOR) 기능을 더 포함하여 데이터의 중복 기록 및 고장 방지 기능을 보장하기 위해 저장장치(345)를 가로지르는 패리티를 유지관리한다.

전술한 바와 같이, 장치 인터페이스 어댑터(440, 442)는 장치(345)에 액세스를 제공하고, 저장장치의 동작을 제어한다. 또한, 장치 인터페이스 어댑터는 그들 제각기의 어레이 관리기에 동작 상태를 제공한다. 제5도는 다수의 저장장치(345)(본 예에서는 32개의 장치가 SD1 내지 SD32로 라벨링되어 있음)를 도시하는데, 각 장치는 메일박스(500)를 가진다. 제5도에서, 저장장치(345)는 4개의 클러스터로 배치되고, 각 클러스터는 8개의 장치를 포함한다. 장치(345)는 그들 제각기의 장치 인터페이스 어댑터(440, 442)를 통하여 어레이 제어기 경로(352, 354)에 의해 독립적으로 액세스될 수 있다. 각 장치 인터페이스 어댑터는 각 클러스터로부터 하나씩인 4개의 저장장치를 동시에 액세스할 수 있다.

제6도를 참조하면, 각 저장장치(345)에서 유지관리되는 장치 메일박스(500)가 도시되어 있다. 바람직한 실시예에서, 메일박스(500)는 고유 요구 ID(510)(태스크 ID)와, 어레이 제어기(350)로부터 저장장치로 수신된 최근의 요구 상태(520)(태스크 상태)를 저장하는 8바이트 필드를 포함한다. 고유 요구(태스크) ID(510)는 처리 및 동작을 쉽게 하기 위해 연속적인 것이 바람직하지만 필수적이지는 않다. 메일박스(500)는 또한, 요구를 수신한 장치 인터페이스 어댑터 식별과 같은 정보를 포함할 수 있다. 또한, 메일박스는 장치 충실도와 같은 정보를 포함할 수 있다.

제7도는 본 발명의 바람직한 실시예에서 어레이 관리기(412, 414)에 의해 수행되는 장치 액세스 및 큐 관리 동작의 흐름도가 도시되어 있다. 그 동작은 어레이 관리기(414)에도 적용가능하지만 어레이 관리기(412)에 관해서 기술할 것이다.

어레이 관리기(412)는 먼저 큐의 상태에 관한 메시지가 통신 버스(410)를 통하여 어레이 관리기(414)로부터 수신되었는지의 여부를 판정한다[블럭(705)]. 이러한 메시지가 수신된 경우에 어레이 태스크 큐(413)는 적절히 갱신되지만[블럭(706)], 어레이 관리기(412)는 장치(345)중의 하나로부터 인터럽트가 수신되었는지의 여부를 판정한다[블럭(710)]. 장치 n으로부터의 장치 인터럽트가 수신되는 경우, 장치 n이 선택되고 그의 메일 박스가 판독되어 태스크 큐(요구 ID) 및 장치 n에 디스패치된 최근 요구의 상태가 무엇인지가 결정된다[블럭(712, 715)]. 메일박스 레지스터의 내용이 고유 순차 ID R(태스크 R로 참조됨)를 가지



는 태스크가 진행중임을 표시한다고 가정한다. 그러면, 어레이 관리기(412)는 어레이 태스크 큐(413)를 갱신하고, R보다 작은 고유 순차 ID를 가지는 모든 요구를 삭제하고, 태스크 큐 R가 진행중임을 알리기 위하여 큐(413)를 갱신한다[블럭(716)]. 그 다음, 어레이 관리기(412)는 장치 n에 태스크 R을 완료시키기 위한 커맨드를 발생하고, 동시에 태스크 R이 완료됨을 알리기 위해 장치 n의 메일박스를 갱신한다[블럭(716)]. 그 다음, 어레이 관리기(412)는 장치 n을 선택해제시키고, 그의 어레이 태스크 큐(413)로부터 태스크 R을 삭제한다[블럭(720)].

블럭(715)으로 되돌아가면, 장치 n의 메일박스 레지스터의 내용이 태스크 R이 완료되었음을 가리키는 경우, 어레이 관리기(412)는 태스크 R과 R보다 작은 고유 순차 ID를 가지는 모든 요구들을 삭제시킴으로써 어레이 태스크 큐(413)를 갱신한다[블럭(725)]. 그 다음, 어레이 관리기는 어레이 태스크 큐(413)를 검사하여 장치 n에 대한 큐에 다른 태스크가 있는지의 여부를 판정한다[블럭(726)]. 다른 태스크가 없는 경우, 동작은 블럭(700)으로 되돌아간다. 장치 n에 대한 태스크 S가 있는 경우, 어레이 관리기(412)는 태스크 S 수행을 개시하기 위한 커맨드를 발생하고[블럭(727)], 메일박스는 태스크 S가 진행중임을 알리기 위해 갱신되고[블럭(728)], 그리고, 장치 n은 선택해제되고 큐(413)의 태스크 S의 상태가 갱신된다[블럭(729)].

블럭(710)으로 되돌아가면, 어레이 관리기(412)가 어떠한 장치로부터의 인터럽트도 수신하지 않은 경우, 큐(413)를 검사하여 수행해야 할 다음 태스크가 무엇인지와 무슨 장치인지를 결정한다[블럭(740)]. 큐(413)에 따라 어레이 관리기(412)에 의해 수행될 다음 태스크가 장치 n에 대한 태스크 T라고 가정한다. 그러면, 어레이 관리기(412)는 장치 n을 선택하는 커맨드를 발생하고 그의 메일 박스를 판독하여 ID 및 장치 n에 발생되었던 최근 태스크의 상태를 결정한다[블럭(742)]. 장치 n의 메일박스의 태스크 ID가 T보다 작고[블럭(745)] 그의 상태가 진행중인[블럭(770)] 경우라면 에러가 발생되고, 에러는 어레이 관리기(412)로 되전송된다(블럭775). 장치 n의 메일박스의 태스크 ID가 T보다 작고[블럭(745)] 그의 상태가 완료된 경우, 어레이 관리기는 태스크 T를 개시하기 위한 커맨드를 발생하고 그 태스크 T가 진행중임을 알리기 위해 장치 n에 대한 메일박스를 갱신한 다음, 어레이 태스크 큐(413)의 태스크 T의 상태를 갱신한다[블럭(771, 772, 773)].

장치 n의 메일박스의 태스크 ID가 T보다 큰 U이고[블럭(745)] 태스크 U의 상태가 완료된 경우, 어레이 관리기(412)는 큐(413)로부터의 태스크 T 및 U를 삭제시키고[블럭(755)], 장치 n에 대한 큐(413)에 다른 태스크가 있는지의 여부를 판정한다[블럭(756)]. 만약 다른 태스크가 있는 경우, 동작은 블럭(700)으로 되돌아간다. 장치 n에 대한 큐(413)에 태스크 V가 있는 경우, 어레이 관리기(412)는 태스크 V 수행을 개시하기 위한 커맨드를 발생하고[블럭(757)], 메일박스는 태스크 V가 진행중임을 알리기 위해 갱신되며, 장치 n은 선택해제되고, 큐(413)의 태스크 V의 상태가 갱신된다[블럭(759)].

장치 n의 메일박스의 태스크 ID가 T보다 큰 U이고[블럭(745)] 태스크 U의 상태가 진행중인 경우, 어레이 관리기(412)는 어레이 태스크 큐(413)를 갱신하고 U보다 작은 태스크 ID를 가지는 모든 태스크(요구)들을 삭제시키고, 태스크 U가 진행중임을 알리기 위해 큐(413)를 갱신한다[블럭(751)]. 그 다음, 어레이 관리기(412)는 장치 n에 태스크 U를 완료시키기 위한 커맨드를 발생하고[블럭(752)], 동시에 태스크 U가 완료되었음을 알리기 위해 장치 n의 메일박스를 갱신한다[블럭(753)]. 그 후, 어레이 관리기(412)는 장치 n을 선택해제시키고, 그의 어레이 태스크 큐(413)로부터 태스크 U를 삭제시킨다[블럭(754)].

이제 제4도 내지 제7도를 참조하면, 태스크 큐(413) 및 (415)의 동작과 메일 박스의 사용은 후술되는 예로써 충분히 알 수 있을 것이다. 태스크 큐(413)가 저장장치8(SD8)에 대한 고유 순차 요구(25 및 26)를 포함한다고 가정한다. 요구(25)는 큐의 최상부에 있다(이것은 이 요구가 다음에 수행될 동작이라는 것을 의미한다). 그러면, 어레이 관리기(412)는 장치 인터페이스 어댑터(449)를 통하여 커맨드를 발생하여 저장장치 8이 사용가능한지 또는 사용중인지의 여부를 알아본다. 저장장치 8이 사용중인 경우, 그 정보는 어레이 관리기(412)로 되전송되고, 어레이 관리기(412)는 저장장치 8과 통신하려는 또 다른 시도가 있기 전에 다른 장치에 대한 요구를 수행하도록 결정할 수 있다. 이와 달리, 저장장치 8이 사용가능한 경우, 어레이 관리기(412)는 장치 8을 선택하고, 즉시 저장장치 8의 메일박스(500)를 판독하여 저장장치 8에 의해 수행된 최근의 요구 상태가 무엇인지를 판정한다. 메일박스(500)가 완료된 최근 요구가 요구 번호(24)였음을 보여주는 경우, 어레이 관리기(412)는 요구(25)에 대하여 저장장치 8에 탐색/위치지정 커맨드를 발생한다. 그 다음, 요구(25)가 진행중임을 알리기 위해 메일박스를 갱신한다. 이것은 태스크 상태(520)를 p(p는 진행중임을 나타낸다)로 셋팅함으로써 행해진다. 어레이 관리기(412)는 저장장치 8을 선택해제하고, 큐(25)가 진행중임을 알리기 위해 큐(413)를 갱신한다. 또한, 어레이 관리기(412)는 요구(25)가 진행중임을 어레이 관리기(414)에게 통지하는 메시지를 통신 버스(410)를 통하여 어레이 관리기(414)로 전송한다. 그 다음, 어레이 관리기(412)는 유휴 상태로 복귀될 수 있거나 또는 다른 장치와 통신을 설정한다.

다른 한편으로, 어레이 관리기(412)가 저장장치 8을 선택하고 메일박스(500)를 판독한 후에 요구(25)에 대한 탐색/위치지정 커맨스가 이미 진행중임을 알아낸 경우, 어레이 관리기(412)는 요구(25)가 진행중임을 알리기 위해 큐(413)를 갱신하고, 그 다음, 요구(25)를 완료시키기 위해 장치 8에 커맨드를 발생하고, 상태 변경을 알리기 위해 메일박스를 갱신하며, 저장장치 8을 선택해제하고, 큐(413)로부터 요구(25)를 삭제한다.

이 대신에, 어레이 관리기(412)가 요구(25)를 수행하기 위해 장치 8과 통신링크를 설정하고, 장치 8의 메일박스를 판독한 후에 탐색/위치지정 이 이미 요구(26)에 대하여 이미 진행중임을 알아낸 경우, 어레이 관리기(412)는 요구(25)가 어레이 관리기(414)에 의해 이미 완료되었다고 결정할 것이다. 그 후, 어레이 관리기(412)는 저장장치 8을 선택해제하고, 큐(413)로부터 요구(25)를 삭제한다. 또한 어레이 관리기(412)는 요구(26)가 진행중임을 알리기 위해 큐(413)을 갱신하고, 유휴 상태로 복귀한다.

다른 한편으로, 어레이 관리기(412)가 저장장치 8에 대한 요구(15)를수행하려고, 메일박스(500)를 판독하여 요구(26)가 이미 완료되었음을 알아낸 경우, 어레이 관리기(412)를 요구((25)가 이미 완료되었고, 두 요구(25, 26)가 어레이 관리기(414)에 의해 서비스되었다고 결정할 것이다. 그 다음, 어레이 관리기(412)는 큐(413)로부터 두 요구(25, 26)를 삭제시키고, 큐(413)을 검색하여 큐(413)에 장치 8에 대한 다른 요구가 있는지의 여부를 판정한다. 이러한 요구가 발견되면, 이 요구를 수행한다. 저장장치 8에 대한 다

른 요구가 발견되지 않는 경우, 저장장치 8을 선택해제시키고, 다른 장치에 대한 요구에 대해 큐(413)를 검색한다.

따라서, 상세한 설명 및 실시예를 기반으로, 다수의 부커맨드(2, 3 또는 그 이상의 부커맨드)를 가지는 요구의 서비스는 하나의 어레이 관리기에 의해 개시될 수 있고, 동일한 요구는 또 다른 어레이 관리기에 의해 더 수행되거나 또는 완료될 수 있다는 것을 쉽게 이해할 수 있을 것이다. 또한, 두 어레이 태스크 큐에 있는 몇몇 요구는 다른 어레이 관리기가 이러한 서비스에 관여하지 않고 단지 하나의 어레이 관리기에 의해 서비스 될 수 있다는 것을 쉽게 이해할 수 있을 것이다.

그러나, 각 저장장치의 메일박스 레지스터를 사용하면 어레이 제어기의 각 어레이 관리기에 각 요구의 상태를 제공하기에 쉽고 효율적이다. 따라서, 각 제어기 경로에서의 큐들은 데이터 보존성 문제 또는 동일한 요구를 중복할 가능성 없이 서로 동기되지 않아도 된다. 각 레지스터의 메일박스와 함께 레벨로 분류되는 고유 요구를 사용함에도 불구하고, 각 요구의 상태(큐에서 완료, 진행, 대기)는 개별의 고속 통신링크를 사용하지 않거나 또는 어레이 태스크 큐가 항상 동기되게 하기 위한 복잡한 록킹 방안 없이도 어레이 관리기들간에 통신될 수 있다.

제8도는 본 발명의 다른 실시예를 도시한다. 데이터 처리 시스템(800)은 저장 제어기(820)와 통신하는 호스트(810)를 포함하고, 저장 제어기는 저장장치(840)와 직접 통신한다. 본 실시예는 제4도 및 제5도에 도시된 바람직한 실시예와 달리, 저장제어기(820)는 장치(840)에 대한 액세스를 직접 제어한다. 호스트(810)는 응용 프로그램(812), 운영 체제(814), I/O 슈퍼바이저(816) 및 다수의 통신 채널(818)을 구비한다.

저장 제어기(820)는 호스트(810)와 통신하기 위한 다수의 I/O 포트(822), 공유 캐시(824) 및 두 저장 부제어기(831, 832)를 구비한다. 저장 부제어기(831)는 태스크 큐(826) 및 제어기 경로(829)를 포함한다. 저장 부제어기(832)는 태스크 큐(828) 및 제어기 경로(830)를 포함한다. 각 제어기 경로는 프로세서 및 메모리 장치를 더 포함한다. 저장 부제어기(831, 832)는 통신 채널(846, 847)을 통하여 저장장치(840)와 통신한다. 각 저장장치(840)는 고유 ID 및 각 저장장치에 의해 수행된 최근 요구의 상태를 저장하는 메일박스를 포함한다. 저장 부제어기의 장치에 대한 액세스, 메일박스 판독 및 태스크 큐 관리의 처리는 본 발명의 바람직한 실시예와 유사하다. 이 실시예에서, 태스크 큐(826, 828)는 시스템 동작을 저하시키지 않거나, 데이터 보존성 문제를 일으키지 않거나 또는 요구를 중복할 필요가 없이 서로 동기되지 않아도 된다.

제9도는 제8도에 도시된 실시예와 유사한 본 발명의 또 다른 실시예가 도시되어 있다. 이 또 다른 실시예에서, 데이터 처리 시스템(900)은 저장 제어기(920)를 구비하고, ID 저장 제어기는 4개의 부제어기(930, 931, 932 및 933)를 포함한다. 각 저장 부제어기는 그 자신의 태스크 큐를 더 포함한다. 따라서, 이 실시예에서, 호스트(910)가 발생하는 요구를 서비스할 수 있는 저장 제어기에 4개의 태스크 큐가 있다. 개시된 본 발명을 사용하여, 태스크 큐는 데이터 보존성 문제를 일으키지 않거나 장치 레벨에서 요구를 중복하지 않거나 또는 큐에 대한 복잡한 록킹 방안없이도 서로 완전히 동기될 필요없다.

제10도는 본 발명의 또 다른 실시예를 도시하는 데, 여기서 호스트는 네트워크에서 함께 연결된 다수의 저장 제어기를 통하여 저장장치와 통신할 수 있다. 이 또 다른 실시예에서, 데이터 처리 시스템(1000)은 호스트(1010), 다수의 저장 제어기(1020A, 1020B, . . . ., 1020N) 및 저장장치(1040)를 포함한다. 각 저장 제어기는 캐시를 포함할 수 있고, 또한 하나 또는 그 이상의 저장 부제어기를 포함할 수 있으며, 이 각 저장 부제어기는 그 자신의 태스크 큐를 가진다. 따라서, 이 실시예에서, 호스트(1010) 요구는 처리를 위한 두 개 또는 그 이상의 제어기로 전송된다. 요구가 발생되어 제어기로 전송되는 동시에 고유 태스크 ID가 호스트(1010)에 의해 발생하는 것이 바람직하지만 필수적이지는 않다. 요구가 캐시에 미리 저장된 정보로부터 처리될 수 없는 경우, 태스크 큐에 저장될 것이며 본 발명의 바람직한 실시예와 유사한 방식으로 저장장치(1040)를 액세스하므로써 처리될 것이다. 개시된 본 발명을 사용하면, 상이한 제어기의 태스크 큐는 데이터 보존성 문제, 장치 레벨에서 요구의 중복 또는 큐에 대한 정교한 록킹 방안 없이 서로 동기되지 않아도 된다.

### 발명의 효과

본 발명의 다수의 실시예가 기술되었지만, 본 발명의 사상 및 범주를 벗어나지 않고서도 각종 변형이 행해질 수 있다는 것을 알 수 있을 것이다. 예를 들면, 태스크 큐의 수는 개시된 본 발명에 제한되지 않는다. 메일박스는 장치에 의해 처리되는 과거 n 요구의 히스토리 및 요구를 개시하거나 또는 완료하는 경로와 같은 부가적인 정보를 유지하도록 확장될 수 있다. 메일박스는 중단된 요구와 같은 정보 및 장치 액세스가 특정 경로로 제한되는 지의 여부와 같은 정보를 더 포함할 수 있다. 따라서, 본 발명은 도시된 특정 실시예에 의해 제한되지 않으며 단지 첨부된 특허청구의 범위의 범주에 의해서만 제한된다.

### (57) 청구의 범위

#### 청구항 1

데이터 처리 시스템에 있어서, 중앙처리장치(CPU) ; 상기 CPU와 통신하는 다수의 제어기-각 제어기는 하나의 큐를 가지며, 상기 큐는 상기 CPU로부터 수신한 각각의 고유 식별 라벨을 가지는 요구(request)를 저장함- ; 및 상기 큐들중의 하나로부터 수신한 요구를 서비스하고, 상기 서비스되는 요구의 상태 및 고유 식별 라벨을 저장하는 메일박스(mailbox)를 가지는 저장 장치를 구비한 데이터 처리 시스템.

#### 청구항 2

제1항에 있어서, 상기 요구는 다수의 부커맨드(sub-commands)를 포함하는 데이터 처리 시스템.

#### 청구항 3

제2항에 있어서, 상기 메일박스는 수행되는 각 부커맨드의 상태를 저장하는 상태 필드를 더 포함하는 데



이더 처리 시스템.

#### 청구항 4

제3항에 있어서, 상기 각 제어기는 상기 메일박스를 판독하고, 상기 메일박스의 내용을 기반으로 그의 큐에 서비스되는 요구의 상태를 갱신하는 수단을 포함하는 데이터 처리 시스템.

#### 청구항 5

데이터 처리 시스템에 있어서, 중앙처리장치(CPU); 상기 CPU와 통신하는 저장 부시스템-상기 저장 부시스템은 상기 CPU로부터 수신한 각각의 고유 식별 라벨을 가지는 요구를 동시에 저장하는 큐를 가지는 제어기를 포함함-; 및 상기 큐로부터 수신한 요구를 서비스하며, 상기 서비스되는 요구의 상태 및 고유 식별 라벨을 저장하는 메일박스를 가지는 저장 장치를 구비한 데이터 처리 시스템.

#### 청구항 6

제5항에 있어서, 상기 요구는 다수의 부커맨드를 포함하는 데이터 처리 시스템.

#### 청구항 7

제6항에 있어서, 상기 메일박스는 수행되는 각 부커맨드의 상태를 저장하는 상태 필드를 더 포함하는 데이터 처리 시스템.

#### 청구항 8

제7항에 있어서, 상기 각 제어기는 상기 메일박스를 판독하고, 상기 메일박스의 내용을 기반으로 상기 큐에 서비스되는 요구의 상태를 갱신하는 수단을 포함하는 데이터 처리 시스템.

#### 청구항 9

데이터 처리 시스템에 있어서, 중앙처리장치(CPU)와 ; 상기 CPU와 통신하는 저장 부시스템-상기 저장 부시스템은 다수의 저장 부제어기를 가지고, 상기 각 저장 부제어기는 하나의 큐를 포함하고, 상기 큐는 상기 CPU로부터 수신한 각각의 고유 식별 라벨을 가지는 요구를 동시에 저장하는 저장 제어기를 포함함- ; 및 상기 큐로부터 수신한 요구를 서비스하며, 서비스되는 각 요구의 상태 및 고유 식별 라벨을 저장하는 메일박스를 가지는 저장 장치를 구비한 데이터 처리 시스템.

#### 청구항 10

제9항에 있어서, 각 요구는 다수의 부커맨드를 포함하는 데이터 처리 시스템.

#### 청구항 11

제10항에 있어서, 상기 메일박스는 수행되는 각 부커맨드의 상태를 저장하는 상태 필드를 더 포함하는 데이터 처리 시스템.

#### 청구항 12

제11항에 있어서, 상기 메일박스를 판독하고, 상기 메일박스의 내용을 기반으로 상기 큐에 요구의 상태를 갱신하는 수단을 더 포함하는 데이터 처리 시스템.

#### 청구항 13

호스트 및 저장 부시스템을 구비한 데이터 처리 시스템으로서, 상기 저장 부시스템은 저장 제어기와 상기 호스트로부터 수신한 요구를 처리하는 저장 장치를 포함하고, 상기 저장 제어기는 다수의 큐를 포함하는 데이터 처리 시스템에서, 상기 각 큐의 상태를 갱신하는 방법에 있어서, 상기 큐에 동시에 저장된 각 요구에 고유 식별 라벨을 배정하는 단계와 ; 서비스되는 각 요구의 상태를 저장하기 위해 상기 저장 장치의 메일박스를 유지관리하는 단계와 ; 상기 메일박스를 판독하여, 서비스되는 최종 요구의 상태를 결정하는 단계와 ; 및 상기 메일박스의 내용을 기반으로 상기 큐를 갱신하는 단계를 포함하는 큐의 상태 갱신 방법.

#### 청구항 14

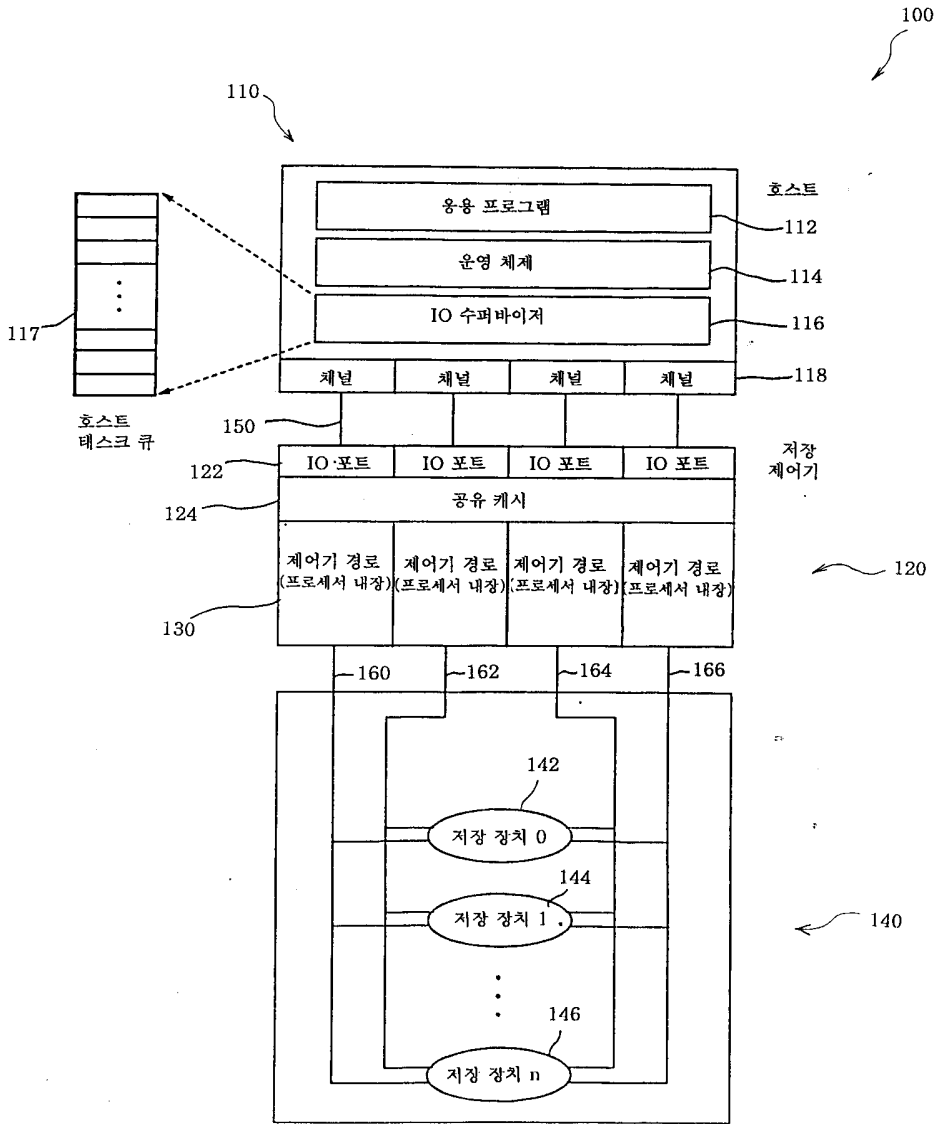
제13항에 있어서, 상기 큐를 저장된 각 요구는 다수의 부커맨드를 포함하는 큐의 갱신 방법.

#### 청구항 15

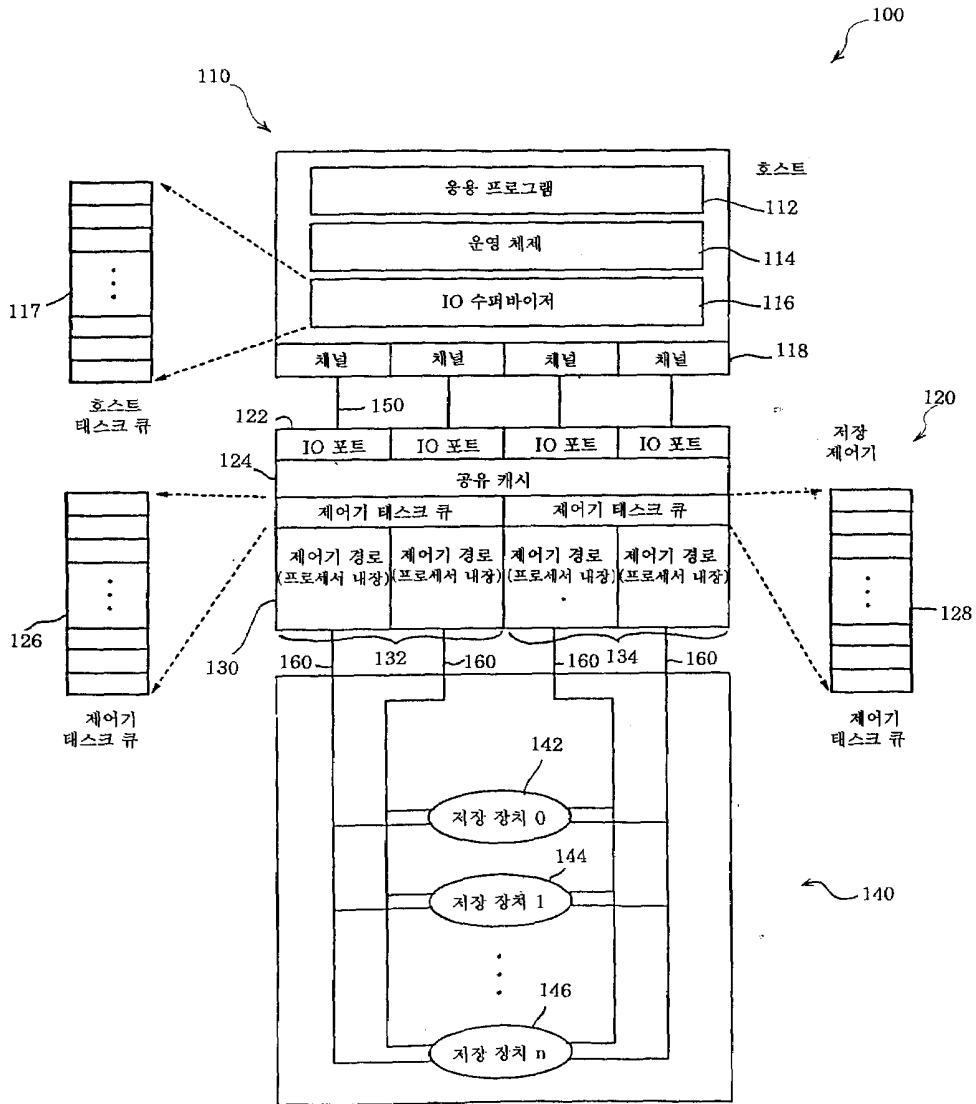
제14항에 있어서, 상기 메일박스는 수행되는 각 부커맨드의 상태를 저장하는 상태 필드를 포함하는 큐의 갱신 방법.

**도면**

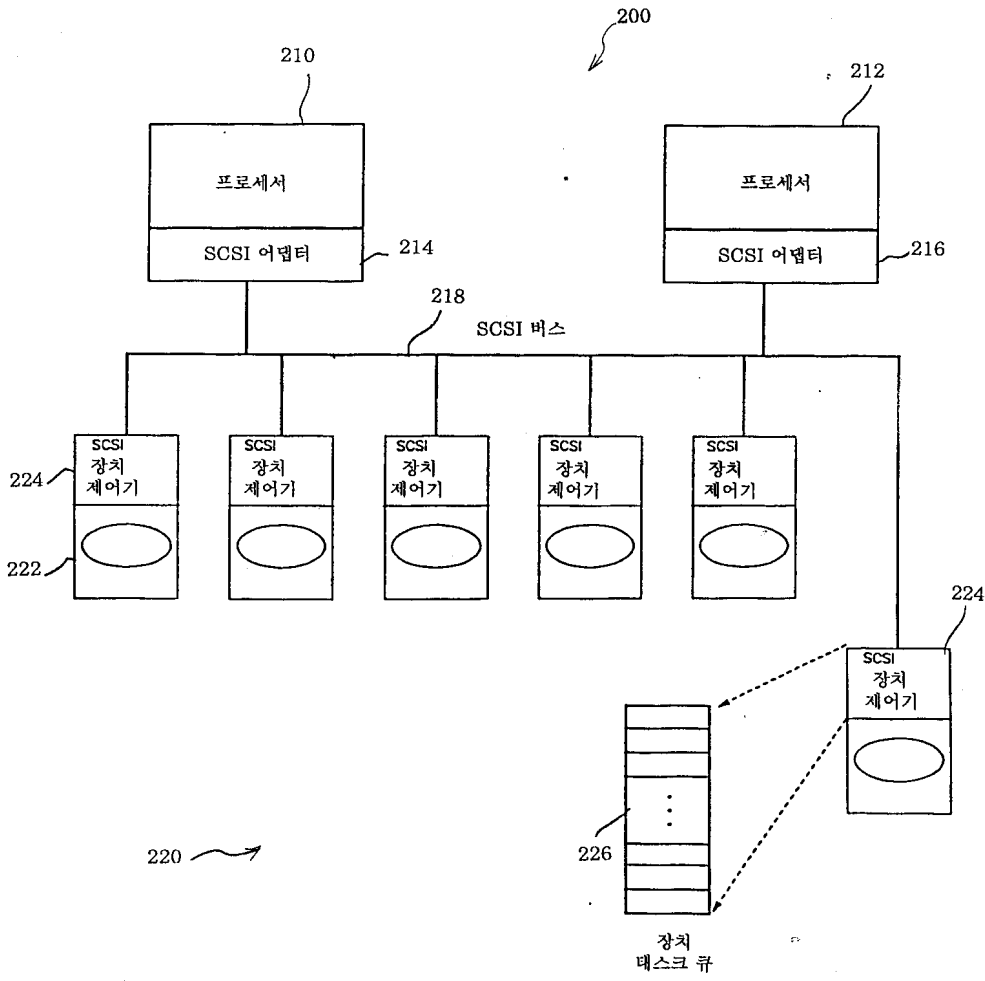
도면1



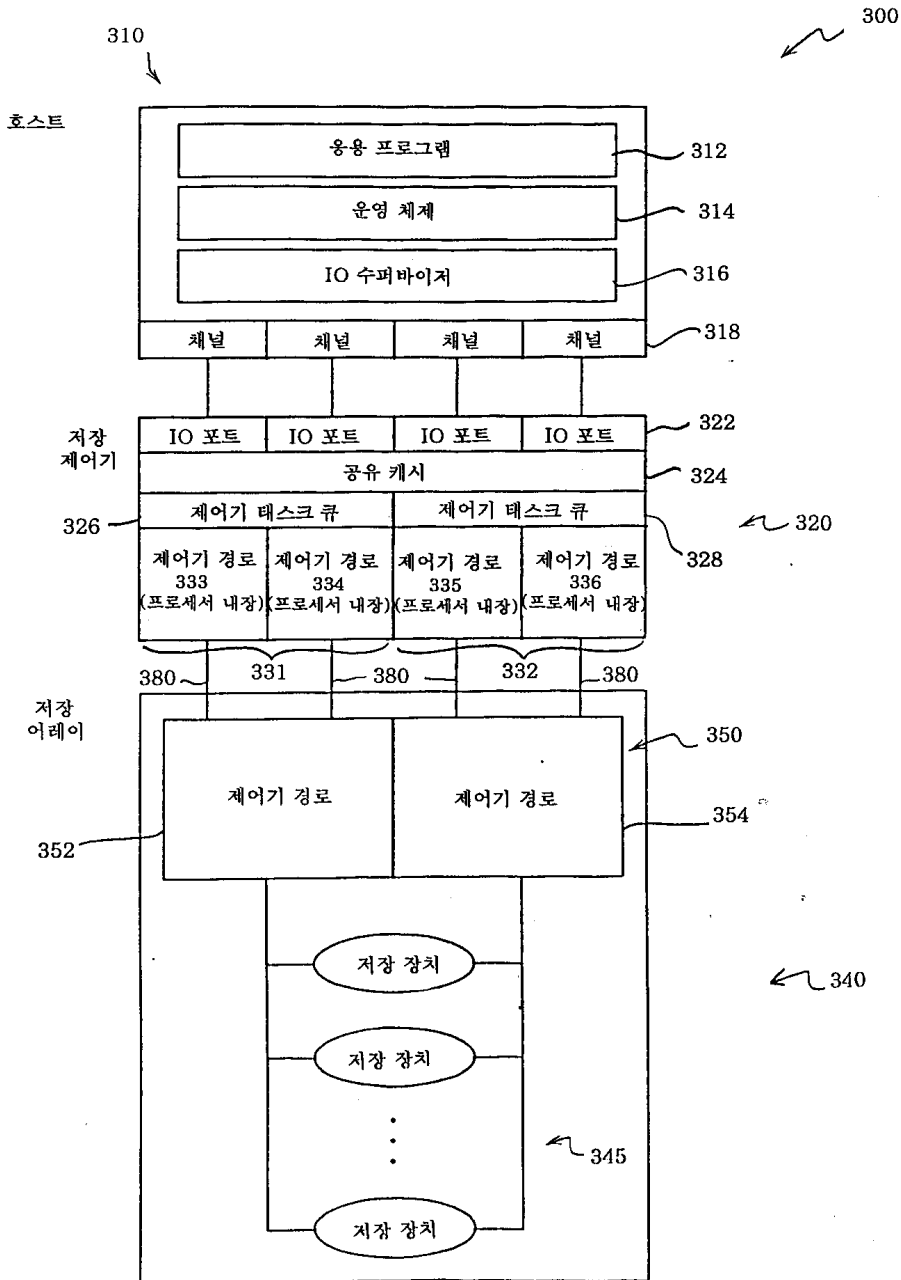
도면2



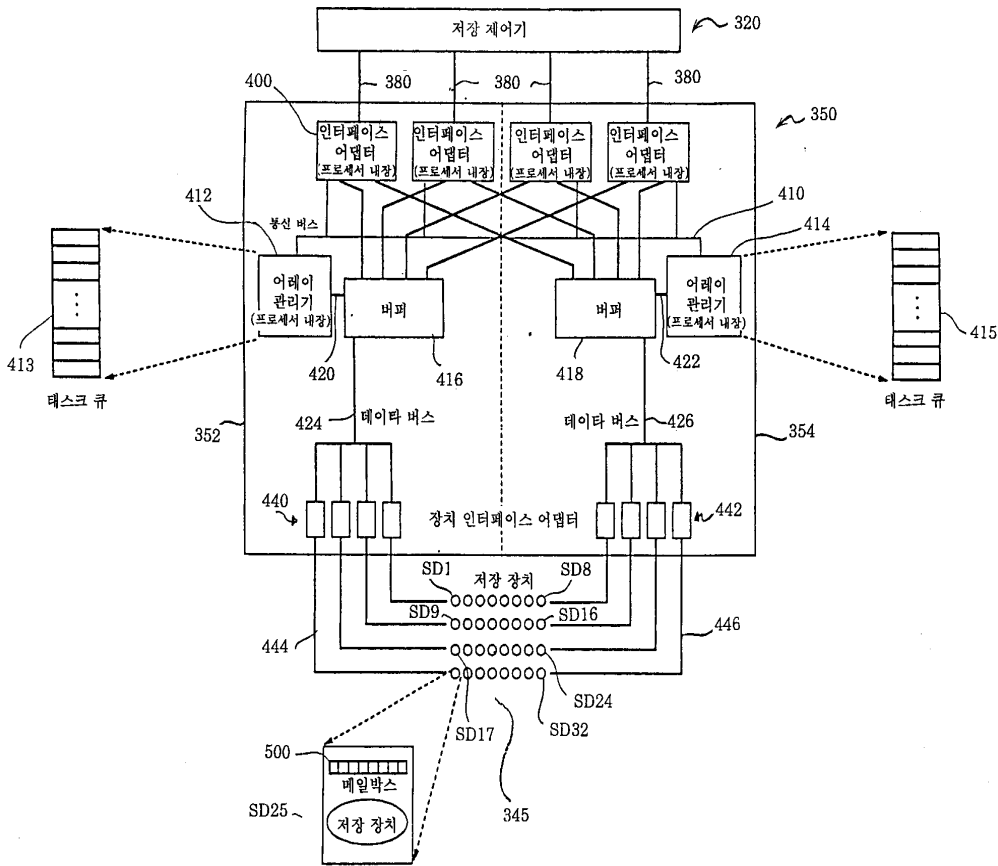
도면3



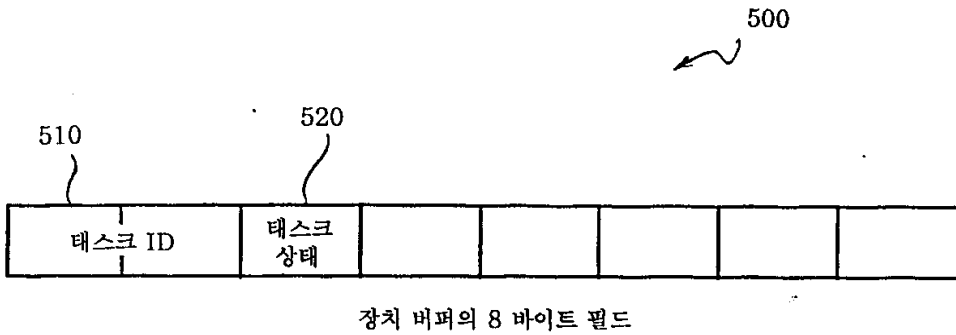
도면4



도면5

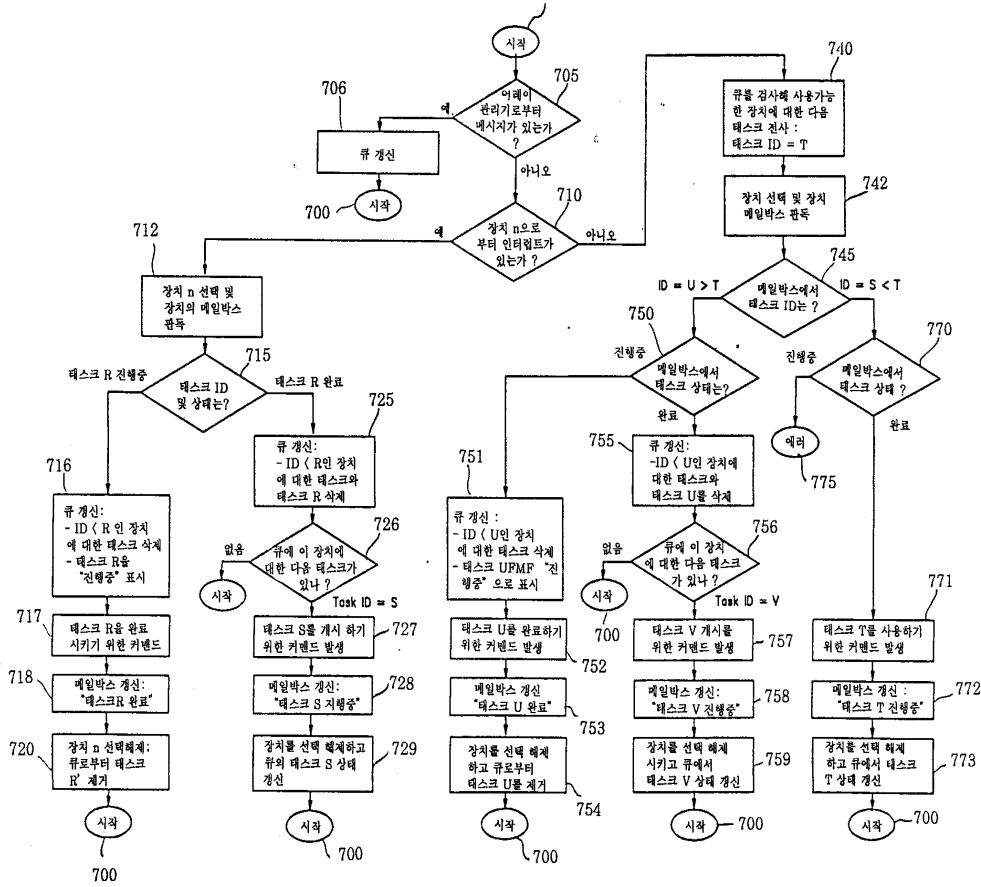


도면6

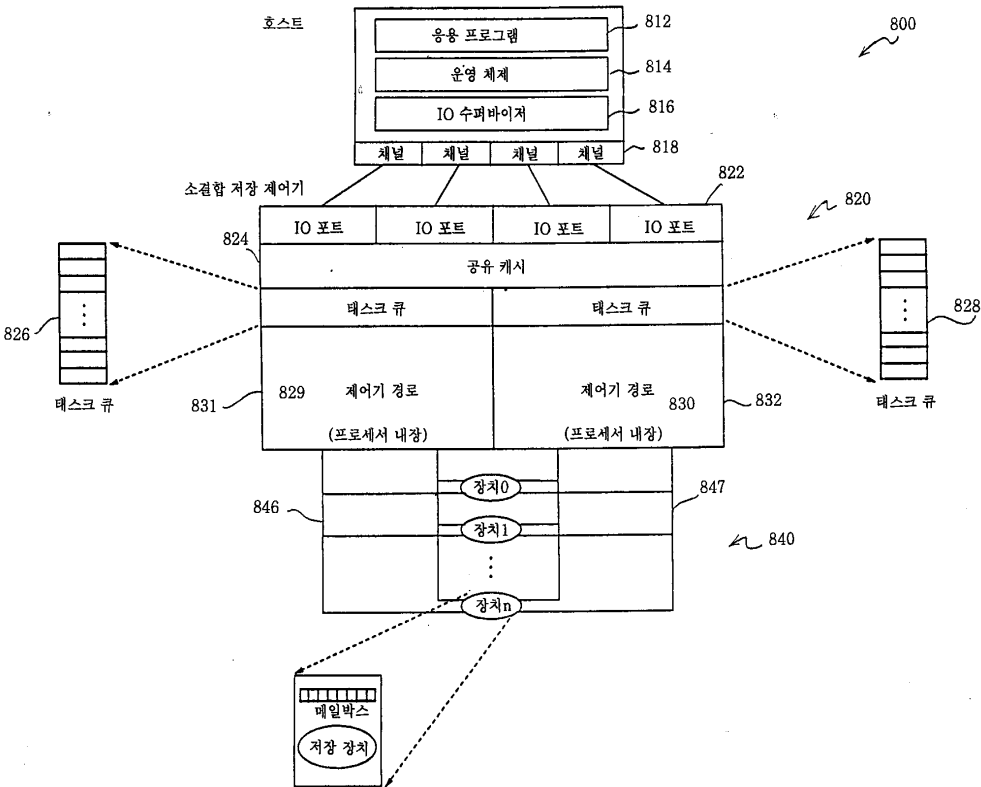




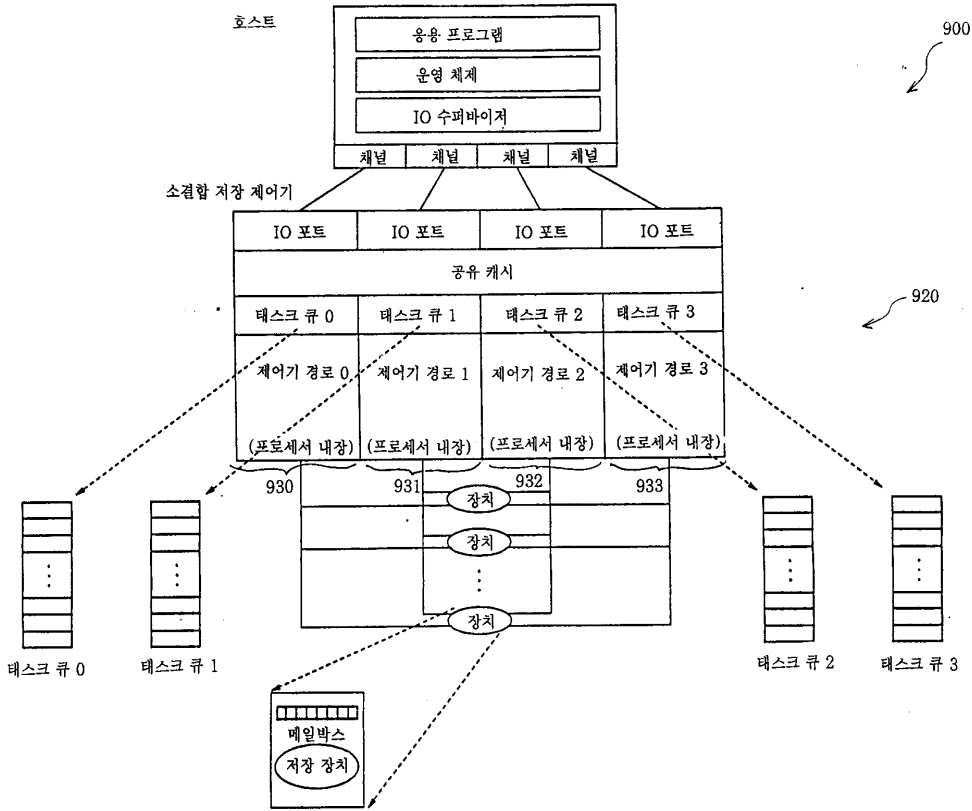
도면7



도면8



도면9



도면10

