

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(51) Int. Cl.<sup>6</sup>  
G06F 15/16

(11) 공개번호 특1999-0087119  
(43) 공개일자 1999년12월15일

(21) 출원번호	10-1998-0706509	(87) 국제공개번호	WO 1997/31318
(22) 출원일자	1998년08월21일	(87) 국제공개일자	1997년08월28일
번역문제출일자	1998년08월21일		
(86) 국제출원번호	PCT/DE1997/00216		
(86) 국제출원출원일자	1997년02월04일		
(81) 지정국	EP 유럽특허 : 오스트리아 벨기에 스위스 독일 덴마크 스페인 프랑스 영국 그리스 이탈리아 룩셈부르크 모나코 네덜란드 포르투갈 스웨덴 국내특허 : 아일랜드 일본 대한민국		
(30) 우선권 주장	196 06 629.8 1996년02월22일 독일(DE)		
(71) 출원인	지멘스 닉스도르프 인포르마치온시스템에 악티엔게젤샤프트 요아힘 부르하르트 독일 데-33106 파더보른 하인츠-닉스도르프-링 1		
(72) 발명자	루에트케, 하랄트 독일 데-33106 파더보른 하인츠-닉스도르프-링 12		
(74) 대리인	남상선		

심사청구 : 없음

(54) 다중-처리기 중앙처리장치

요약

본 발명은 각각이 처리기, 메모리 및 커플러를 포함하는 두 개 이상의 그룹을 가지며, 처리기 및 메모리는 정확하게 한 커플러에 의해 상호연결되고 커플러는 상호연결되는 중앙처리장치에 관한 것이다. 주소를 인터리빙시키므로써, 주소공간에 걸쳐 균일하게 분포된 메모리 영역은 각각의 그룹과 이산적으로 연관된다. 각각의 커플러는 자신의 커플러그룹과 연관된 메모리 영역에 대한 액세스를 수행하며 관련 커플러로의 연결을 통해 다른 액세스를 수행한다. 중앙처리장치에는 다중 시스템을 구성하기 위한 인터페이스가 구비된다.

대표도

도1

명세서

기술분야

본 발명은 모듈러 다중-처리기 시스템에 관한 것이다.

배경기술

고성능 컴퓨터 시스템은 흔히 다중-처리기 시스템으로서 구성된다. 특히 다수 처리기가 균등한 응답시간 특성을 달성하기 위해 예약 시스템등과 같은 작업단위 시스템에 사용된다. 이러한 장치에서, 공통 백플레인으로 상호연결된 모듈이 적절히 이용된다. 본 명세서의 다음 설명에서 중앙처리장치로 불리는 각각의 모듈은 백플레인상에서 버스 시스템을 통해 결합된 하나이상의 처리기를 포함한다.

한편, 처리기가 액세스할 수 있는 메모리는 별개 메모리 모듈로서 백플레인에 연결될 수 있다. 그러나, 이것은 버스 시스템의 대역폭이 전체 시스템의 효율을 제한하게 한다.

따라서, 중앙처리장치 전체에 메모리를 균등하게 분포시키고, 각각의 중앙처리장치에 처리기에 의해 직접 또는 버스 시스템을 통해 다른 처리기에 의해 액세스될 수 있는 메모리를 배열하는 것이 제안되어왔다. 각각의 메모리는 주소 변환에 의해 모든 처리기에 의해 공유된 주소공간에 매핑된다. 국부 메모리에 위치한 프로그램 또는 데이터는 특히 고속으로 효율적으로 액세스될 수 있다. 이러한 이유로, 이 메모리는 단일 연속 블록으로서 주소공간에 놓여진다.

그럼에도 불구하고, 버스 시스템은 다수 처리기를 포함하는 고성능 컴퓨터의 경우엔 병목상태에 있게된다. 국부 메모리 이외에 대한 각각의 액세스는 버스 시스템을 통해 처리된다. 메모리 액세스의 대부분이 정적으로 구성가능한 프로세스 제어 또는 트랜잭션 시스템을 제외하곤 국부 메모리에 대해 액세스되는 것을 달성하는 것은 드물다. 예를들어 4-처리기 컴퓨터에서 액세스의 절반이 국부 메모리에 대한 액세스일 지라도, 버스 시스템의 대역폭은 1-처리기 컴퓨터의 대역폭에 비해 두배를 갖는다. 버스 시스

템의 대역폭을 증가시키지 않고, 8 내지 16 처리기로 변경시키는 것은 비율면에서 상당히 작은 성능 증가와 연관된다.

또한 고속 버스 시스템과 같은 임의의 길이를 가질 수 없는 문제도 있다. 현재 사용된 빈도 및 기술에서, 버스길이는 신호 지연시간 및 임피던스에 기인하여 약 30 cm로 제한된다. 중앙처리장치에 대해, 상당히 감소될 수 없는, 3cm인 폭이 주어진다면, 이것은 최대 10개 모듈 따라서 10개 처리기를 가질 수 있게 한다. 그러나, 고성능 트랜잭션 처리 시스템은 더 많은 수의 처리기를 필요로 한다. 모듈 개수의 제한은 공유 메모리와 다수 처리기를 포함하는 각각의 중앙처리장치모듈에 의해 상쇄될 수 있다. 그러나, 각각의 중앙처리장치는 백플레인의 버스를 통해 다른 중앙처리장치의 메모리에 연결되기 때문에, 버스 시스템의 부담은 버스가 성능을 제한하는 구성요소를 나타내는 정도까지 또다시 증가한다.

버스 시스템에 의해 결합된 다중-컴퓨터 시스템의 대안은, 처리기들이 IEEE 'Standard for Scalable Coherent Interface'(SCI), IEEE STD 1596-1992(ISBN 1-55937-222-2)에 따라 연결된 멀티-컴퓨터 시스템이다. 이것은 한 처리기로부터 다음 처리기로 포인트-대-포인트(point-to-point) 연결에 관한 것이다. 따라서 각각의 처리기가 각각의 다른 처리기의 메모리를 액세스할 수 있기 때문에, SCI는 각각의 처리기가 패킷을 이웃 패킷에 전송할 것을 의도하지 않는 전송 프로토콜을 정의하여, 각각의 처리기가 적어도 하나의 데이터 포켓을 저장할 수 있는 SCI 문서의 섹션 1.3.4에 설명된 바와 같은 링 구조를 필요로 하게 한다. 이것은, 임의 개수의 처리기가 사용될 수 있게할 지라도, 성능을 손상시키는 지연을 생성한다. 여기서, 버스 시스템에 비해 큰 경우에도 대역폭 제한이 주어진다. 따라서 특히, 각각의 처리기가 단지 동일 라인 또는 열에서 다른 처리기를 액세스할 수 있는 2차원 격자 구조를 구축하는 것이 제안되었다. 기타 연결은 섹션 1.4.6에서 '에이전트'라 불리는 중개수단을 통해 처리되어야만 한다. 그러나 SCI 프로토콜은 매우 고비용을 필요로 한다.

그러므로 본 발명의 목적은 상기 단점을 방지하는 멀티컴퓨터 시스템을 위한 중앙처리장치를 제공하는 것이다.

### **발명의 상세한 설명**

본 발명은 버스 시스템과 직접연결을 결합함으로써 증대되어야 할 버스 시스템의 수가 증대되는 것을 허용하며 대응하여 증대된 대역폭의 단일 버스 시스템과 동일한 효과를 달성할 수 있게 한다.

바람직한 해결책은 각각 두 개 처리기 및 두 개의 개별 메모리를 갖는 모듈을 사용하는 것이다. 각각의 처리기는 두 개의 메모리중의 하나가 직접적으로 연결되는 커플러를 통해 주 메모리를 액세스한다. 각각의 커플러는 직접적으로 연결된 커플러 및 직접 연결된 메모리에 의해 만족되지 않는 메모리 액세스가 버스 시스템을 통해 발생하도록 메모리의 다른 커플러에 직접 연결되며 버스 시스템에 대한 포트를 갖는다. 이 해결책은 두 버스 시스템을 이용하여 간명한 방식으로 모듈사이에서 연결의 대역폭이 배가되게 한다.

본 발명의 더욱 개량된 예에서, 각각의 커플러는 다른 커플러로의 직접연결을 가지며, 이 직접연결은 모듈상의 추가포트에 연결된다. 따라서, 각각의 경우에 4개 커플러가 링 형태로 연결되는 한 중앙처리장치에 두 모듈이 연결될 수 있다. 이 경우에, 직접연결 유형은 간접연결이 대각선 방향으로 대향한 커플러에 연결되게 하는 커플러 사이에서 선택된다.

버스 시스템을 필요로 하는 메모리 액세스는 필요로 되는 메모리가 간접적으로 연결되는 커플러가 연결되는 버스 시스템만을 사용하는 결과로 된다. 기타 데이터 경로는 각각의 중앙처리장치에 위치한 연결을 통해 행해진다. 동시에, 버스 시스템의 사용은 잘 알려진 방식으로 융통성있는 확장성 및 주변장치에 대한 연결을 제공한다.

이러한 장치에서, 특히 유익한 해결책은 주소 인터리빙에 의해 가능하다. 이를 위해, 주소의 하위 유효 비트는 메모리를 액세스하게 하는 커플러를 결정한다. 중앙처리장치 당 두 개 처리기 및 캐시의 블록 사이지가 32 비트인 경우에, 6번째(또는 7번째 또는 8번째) 비트가 커플러를 선택하는 데 사용된다. 이것은 커플러의 개수가 2의 거듭제곱이라는 사실에 기초한다. 버스 시스템상에서의 주소 레졸루션은 버스상에 주소를 적재하므로써 그리고 버스 사이클을 완료시키는 연관된 메모리를 갖는 커플러에 의해 잘 알려진 방식으로 달성된다. 따라서, 관련 메모리를 포함하는 중앙처리장치를 주소지정할 필요가 없다. 결과적으로, 커플러에 대한 할당이 동일하게 존재하기 때문에, 추가 중앙처리장치가 버스 시스템의 경우에서 일반적인 경우와 같이 추가될 때 주소지정에 있어서의 어떠한 변경도 버스 시스템에 필요치 않다.

처음에, 이 해결책은 단일 처리기에서 실행된 작은 프로그램이 처리기의 커플러에 직접적으로 연결된 메모리 뿐만아니라 기타 커플러 또는 커플러에 연결된 메모리를 사용하지 않기 때문에 매우 유익한 것으로 보이지는 않는다. 그러나, 캐시가 있고 커플러 사이에 연결부가 특히 포인트-대-포인트 연결부로서 고속으로 구축될 수 있기 때문에, 이 단점은 확장성의 용이함으로 인해 상쇄된다.

따라서, 각각의 그룹이 하나의 처리기, 하나의 메모리 및 하나의 커플러로 된 둘 이상의 그룹을 포함하는 중앙처리장치가 있으며, 여기서 처리기 및 메모리는 한 커플러에 정확히 연결되고 커플러는 상호연결되며, 주소공간에 균등하게 분배된 메모리 영역은 주소 인터리빙에 의해 각각의 그룹에 이산적으로 할당되며 각각의 커플러는 그룹에 할당된 메모리 영역에 대한 액세스를 만족시키며 고려되는 커플러 연결을 통한 기타 액세스를 만족시킨다. 중앙처리장치에는 다중 시스템을 구축하기 위한 인터페이스가 구비된다.

### **도면의 간단한 설명**

도 1은 두 개의 처리기, 두 개의 메모리 및 두 개의 커플러, 두 개의 직접 연결부 및 두 개의 버스 포트를 포함하는 본 발명의 바람직한 실시예에 따른 중앙처리장치를 나타낸 도.

도 2는 대형 다중-처리기 시스템의 일부분으로서의 역할을 할 수 있는 4 버스 시스템을 갖춘 가상 4-처리기 중앙처리장치를 형성하기 위해 두 개의 중앙처리장치의 결합을 나타낸 도.

도 3은 2 버스 시스템에 의해 연결된 8중(octuple) 다중-처리기 시스템을 형성하기 위해 4개의 듀얼-처리기 중앙처리장치의 결합을 나타낸 도.

도 4는 8 버스 시스템에 의해 다른 8중 처리기 슬라이스에 연결될 수 있는 8중 처리기 슬라이스를 형성하기 위해 4개 듀얼-처리기 중앙처리장치의 결합을 나타낸 도.

### 실시예

도 1은 두 개의 처리기(11,21), 두 개의 메모리(12,22) 및 두 개의 커플러(13,23)를 갖는 중앙처리장치(10)를 나타낸다. 상부에 도시된 그룹은 처리기(11), 메모리(12) 및 커플러(13)로 이루어지고 하부에 도시된 그룹은 다른 처리기(21), 메모리(22) 및 커플러(23)로 이루어진다. 처리기는 모두 동일한 주소공간을 사용하며 각각의 경우에 다른 처리기의 메모리를 사용할 수 있다. 이를 위해, 처리기는 연결된 커플러에 액세스 요구(request)를 전달한다. 커플러는 직접적으로 연결된 메모리를 사용하거나 연결된 메모리에 의해 만족될 수 있도록 기타 커플러에 연결부(18)를 통하여 액세스 요구를 전달한다.

포인트-대-포인트 연결로 커플러를 서로 연결시키는 다양한 방법이 사용될 수 있다. 한 가능한 방법은 상기한 IEEE STD 1596-1992에 따른 SCI 연결이다. 대체로, 일반적으로 알려진 단방향 또는 양방향 데이터 및 주소라인 및 스트로브 및 승인(acknowledgment) 신호를 통한 더욱 간단한 연결로 충분하다. 간접적인 이용가능성이 필요한 경우, 의도되는 신호를 위한 커플러를 지정하는 하나 이상의 추가 주소라인이 사용된다. 이 장치에서, 신호는 저장되지 않지만 가능하다면 다음 커플러로의 연결부에 직접 전달된다. 이러한 회로 설계는 일반적으로 전문가에게 이해될 수 있으며 다음설명에서 공지될 것으로 가정된다.

이 경우 제 1 커플러(13) 및 이에 연결된 메모리(12)에 속하며 커플러(23) 및 메모리(22)에 속하는 주소공간 영역이 지정되어야만 한다. 일반적으로, 처리기의 전체 주소공간으로 부터의 연속적인 메모리 영역은 각각의 경우 메모리(12) 및 메모리(22)에 할당되며 주소 비교기에 의해 결정된다. 이 경우, 듀얼-처리기 중앙처리장치는 공유버스 시스템을 위해 인터페이스(도시되지 않음)를 갖춘 중앙처리장치를 제공하며 국부 메모리가 영향을 받는지의 여부를 결정하는 메모리 관리장치를 추가함으로써 다중 처리기 시스템을 형성하도록 결합될 수 있거나 다른 중앙처리장치의 버스 및 메모리 영역을 액세스할 것이 의도된다. 만일 주소지정된 메모리 영역이 직접적으로 연결된 메모리에 있지 않거나 다른 결합 유닛에 연결된 메모리에 있지 않다면, 메모리 액세스는 버스를 통해 처리된다. 간명한 테이블 액세스는 액세스를 위해 커플러가 사용하는 연결에 대한 결정을 위해, 특히 중앙처리장치의 두 메모리 영역이 서로 밀접하게 맞춰지고 각각의 중앙처리장치의 메모리가 동일한 사이즈인 경우 사용될 수 있다.

그러나, 본 발명의 바람직한 실시예에서, 메모리의 상이한 분할부가 사용된다. 이 장치에서, 주소영역은 인터리빙되며 물리적 메모리는 주소공간에 대해 작은 유닛으로 분배된다. 이러한 유형의 주소지정은 하나 이상의 처리기가 동일한 액세스 권한으로 즉, 모든 처리기에 의해 동일한 효율로 액세스될 수 있으며 액세스 후 회복시간을 필요로 하는 메모리 모듈을 액세스한다면 인터리빙으로서 공지되어 있다. 이것은 예를들어, '인터리빙' 이란 키워드로, F.J. Hill 및 G.R. Peterson에 의한 저서 'Digital Systems: Hardware Organization and Design', New York 1973의 섹션14.3 에 설명되어 있다. 예를들어, 제 1 메모리 뱅크에 짝수번호로 된 주소를 위치시키고 제 2 메모리 뱅크에 홀수번호로 된 주소를 위치시킴으로써, 제 2 메모리는 두 연속적인 메모리 액세스 경우에 제 1 메모리의 회복시간 동안 이미 액세스될 수 있다. 따라서, 인터리빙은 둘 이상의 메모리 뱅크의 회복시간을 보상하는 데 사용되어 왔다.

본 발명에서 사용된 기술은 인터리빙과 유사하지만 하기에서 주소 인터리빙으로 칭한다. 이 장치에서, 주소공간은, 하기에서 라인으로 불리는, 동일 사이즈로 된 다수의 블록으로 분할된다. 각각의 라인은 적어도 각각 처리기 사이에 또는 메모리와 커플러 사이에 전송된 데이터 볼륨 크기이어야 한다. 이 정보유닛은 특히 캐시가 사용될 때 '캐시라인'으로 불리운다. 대부분의 경우에 한 라인은 사용된 처리기의 다수의 워드 폭이다. 실질적으로 거의 그렇지만 캐시가 사용될 때 처리기의 명령어의 워드 폭은 중요하지 않으며; 이것은 명확성을 위해 이후에 메모리 워드로 불리워 질 처리기 사이에 또는 각각 메모리와 커플러 사이에 전송된 데이터 볼륨이 분할되지 않은 이유이다. 주소 인터리빙은 따라서 한 메모리 워드 또는 다수 메모리 워드에 기초하여 행해진다. 각각의 경우에, 2진 주소의 한 비트가 듀얼 인터리빙의 경우 메모리 뱅크를 결정하는 데 충분하다. 일반적으로, 한 비트 필드를 추출함으로써 메모리 뱅크가 결정될 수 있도록 2의 거듭제곱인 메모리 뱅크의 갯수가 사용된다. 최종 주소비트에 의해 또는 더욱 큰 유닛에서의 인터리빙이 더욱 유익한지의 여부를 결정하기 위해 측정치가 사용될 수 있다. 예를들어 한 메모리 워드를 16번까지 인터리빙하는 것은 주소라인을 재배선함으로써 문제를 발생하지 않고 달성될 수 있으므로, 간명을 위해, 최하위 비트 또는 비트들 상에서의 주소 인터리빙은 본 명세서에서 본 발명이 이에 제한되지 않는 것으로 가정된다. 대조적으로, 인터리빙은 가능한한 가장 큰 분포, 즉 항상 최하위 주소비트를 사용한다.

본 발명에서, 주소 인터리빙은 메모리의 회복시간을 메우는 공백이 아니다. 대신에, 간명한 방식으로 그리고 성능 손실없이 다중처리기 시스템을 형성하도록 결합될 수 있는 중앙처리장치를 제공하기 위해, 한 처리기에 각각의 메모리 유닛이 연관된다는 사실과 함께 사용된다. 이 장치에서, 중앙처리장치에 위치한 두 메모리가 서로 인접할 수 있게 하는 대신 이러한 두 메모리를 인터리빙시키는 것이 처음엔 유익하지 못한 것으로 보인다. 메모리가 서로 인접하는 경우에, 처리기에 국부적인 메모리의 영역에 본질적으로 한정된 프로세스는 특히 효율적으로 수행될 수 있다. 인터리빙된 메모리의 경우에, 실질적으로 모든 프로세스는 두 커플러의 연결을 통해 우회연결(detour)로 비국부적 메모리도 사용하여야 한다. 그러나, 현재 관련 운영체제로 더욱 상세한 연구에 의해 도시된 바와 같이, 임의의 프로세스도 실질적으로 일정 메모리 영역에 한정되지 않기 때문에, 서로 밀접하게 인접하는 메모리 영역의 상기 이점은 효과적으로 달성되지 못한다. 따라서, 메모리를 인터리빙하는 것은 실질적으로 비유익한 것을 의미하지 않는다. 사실상, 운영 소프트웨어가 공유 메모리 영역 특히 운영체제를 위한 공유 메모리 영역을 사용하는 경우 이점이 있는 것으로 알려져 왔다. 이전 장치에서, 국부 메모리에 이들 영역을 갖지 않는 처리기는 단점을 갖는다. 따라서, 프로세스의 실행시간은 다른 프로세스 보다 처리기상에서 더욱 느리다. 이것은 시스템의 전체 성능에 임의의 해로운 영향을 미치지 않고 본 발명에 의해 방지된다.

지금까지 설명된 듀얼-처리기 중앙처리장치는 동시에 단순하고 효율적인 방식으로 다양한 방식으로 다중 컴퓨터 시스템을 형성하도록 결합된다. 커플러의 포인트-대-포인트 연결에 대한 직접 인터페이스(14,24) 및 버스 인터페이스(15 및 25)는 이러한 목적을 위해 적절하다.

도 1로부터 알 수 있는 바와 같이, 직접 연결 인터페이스(14,24)의 각각은 각각의 커플러를 위해 제공된다. 이것은 도 2에 도시된 바와 같은 4개 처리기 및 4중(quadruple) 인터리빙을 갖는 한 중앙처리장치를 형성하기 위해 두 개 중앙처리장치를 결합할 수 있게 한다. 다른 마찬가지로의 중앙처리장치는 커플러(X3 및 X4), 처리기(CPU3 및 CPU4) 및 메모리(메모리3 및 메모리4)를 가진다. 커플러(X1, X2, X3 및 X4)는 서로 링 형태로 연결된다. 메모리는 이제 4중 인터리빙되었다. 이것은 단일 유형의 중앙처리장치가 듀얼 처리기 또는 4중 처리기로서 사용될 수 있도록 중앙처리장치의 점퍼에 의해, 또다른 주소를 추가함으로써 단순한 방식으로 행해질 수 있다. 두 주소 비트로, 각각의 커플러는 액세스가 직접적으로 연결된 메모리에, 인접한 커플러의 메모리에 또는 대향한 커플러의 메모리에 대해 발생하였는지의 여부를 결정할 수 있다. 이것은 임의의 비교기를 필요로 하지 않는다. 이는 대응하는 데이터 경로를 구동하기 위해, 커플러의 개수를 지정하는 두 비트에 대해 주소로부터 단지 2비트를 4비트로 변환하는 것이 필요하기 때문에, 이것은 게이트, 멀티플렉서 및 디코더와 같은 매체를 밀도있게 집적함으로써 달성될 수 있다. 특히, 반전된 배타적 논리합(EOR)에 의해 필요하다면, 동일한 중앙처리장치의 커플러를 주소지정하기 위해 필요한 지 또는 메모리가 직접적으로 사용될 수 없다면 인터페이스(14,24)를 통해 다른 중앙처리장치의 커플러를 주소지정하기 위해 필요한 지를 결정하기 위해 주소의 한 비트가 사용된다. 도 4에 도시된 바와 같이, 이 원리는 링 형태의 4개 듀얼 처리기 중앙처리장치를 각각의 경우에 두 개의 직접 인터페이스에 결합시키므로써 8중 슬라이스를 형성하기 위해서도 사용될 수 있다. 링 구조에 기인한 관련된 성능 손실은 직접연결이 8중 슬라이스가 버스 시스템에 의해 연결된 8개 중앙처리장치 보다 비교적 양호한 성능을 나타내도록 버스 시스템 보다 고속일 수 있게 한다는 사실에 의해 보상된다.

이를 위해 필요한 포인트-대-포인트 연결은 당업자에게 공지되어 있다. 이를 위해 필요한 연결라인은 데이터와 고려되는 주소의 관련부분을 전달한다. 양단부에서 종료되는 개방-콜렉터 라인을 사용하는 것에 대해선 어떠한 문제도 없다. 버스 시스템에서의 경우에서와 같은 어떠한 중간점 접합부도 없고 어떠한 중재도 필요하지 않으므로, 이러한 포인트-대-포인트 연결은 동시에 높은 동작속도를 달성하며 단순한 구조를 갖는다. 3개 이상의 커플러가 링 형태로 연결된다면, 한 커플러가 인접한 커플러를 통해 다른 커플러에 간접적으로 연결될 수 있어야 한다. 이를 위해, 포인트-대-포인트 연결로서 구성될 수 있는 그룹주소 및 승인 라인이 구비된다. 만일 커플러가 그룹주소 라인이 자신의 그룹번호를 나타내지 않는 요구를 수신한다면, 그룹주소 및 모든 기타 라인은 다른 포인트-대-포인트 연결에 위치된다. 주소지정된 커플러는 포인트-대-포인트 연결의 처리기가 아직 처리되지 않은 메모리 요구를 수행하였기 때문에 포인트-대-포인트 연결을 통하여 도달하는 요구를 즉시 처리할 수 없으므로 승인 신호는 통상적인 방식으로 전송될 수 있다.

버스 인터페이스(15,25)를 이용하여, 다수의 중앙처리장치가 결합된 다중처리기 시스템이 결합된다. 4개 듀얼-처리기 중앙처리장치를 갖춘 이러한 장치가 도 3에 도시되어 있다. 이 장치를 위해, 처리기 또는 캐시워드 폭 또는 다수 처리기에 기초한 2중 인터리빙은 두 커플러의 연결 또는 커플러를 통해 처리되는 처리기의 메모리 액세스가 국부적으로 또는 이에 연결된 버스(15,25)를 통해 각각 수행되는지의 여부를 결정하기 위해 사용된다. 도 3에 도시된 바와 같이, 각각의 중앙처리장치를 다른 중앙처리장치와 연결하는 두 버스 시스템이 도시되어 있다. 각각의 경우에 버스에 의해 상기와 같은 방식으로 연결된 커플러는 각각의 경우에 인터리빙에 대해 동일한 메모리 부류에 대한 액세스에 영향을 미친다. 따라서, 액세스가 직접적으로(또는 링에서 간접적으로) 연결된 커플러를 통해 발생하지 않는 인터리빙을 기초로 하여 수립된 후, 커플러는 주소가 국부 메모리 주소가 되는 지 또는 버스 시스템을 통해 요구되어야 하는지를 결정하기 위한 추가 수단으로서 메모리 관리자를 사용한다. 메모리 주소 또는 추가 조정가능한 주소 인터리빙을 위한 비교기가 이러한 목적을 위해 사용될 수 있다. 이러한 목적을 위해 필요한 이들 주소 관리자는 버스 시스템을 갖춘 다중처리기 컴퓨터 제품으로 공지되어 있다.

모든 경우에, 중앙처리장치간에 이들 중앙처리장치의 메모리 요구를 처리하기 위해 다수 버스 시스템이 존재한다. 이것은 버스 시스템을 구동하기 위한 상당한 추가비용없이 이전의 다중-처리기 시스템의 병목현상을 제거한다. 도 2 및 도 4에 나타난 바와 같이 만일 각각 바람직한 형태의 2 또는 4 중앙처리장치 4중 또는 8중 슬라이스를 형성하기 위해 결합된다면, 다중-처리기 시스템의 중앙처리장치간에 데이터소통을 처리하기 위한 4 또는 8 버스 시스템이 있게 된다. 따라서 예를들어 버스 시스템이 중앙처리장치에 대해 최대 10 플러그-인 위치까지 허용한다면, 80개 처리기의 2배로 된 처리기로 된 컴퓨터 시스템이 하나의 동일한 듀얼-처리기 중앙처리장치 모듈을 이용하여 구축될 수 있으며, 연산능력은 단일 버스 시스템이 병목현상을 나타내지 않고 처리기의 수에 따라 거의 선형으로 증가한다.

도 2 및 도 4로부터 알 수 있는 바와 같이, 바람직한 실시예의 듀얼-처리기 중앙처리장치는 상기에서 알 수 있는 바와 같이, 개별 버스 시스템이 최대길이를 가지며 교차하지 않는 방식으로 중앙처리장치를 배열할 수 있게 한다. 이것은 20, 40 및 80 처리기인 최대 구성레벨에 대해 중앙처리장치를 위한 상이한 리셉터클을 필요로 할 지라도, 중앙처리장치는 동일하게 존재한다.

도 3에 따른 장치에서 각각의 경우에 쌍으로 된 인접한 중앙처리장치의 직접연결부를 연결함으로써, 선형 배열은 또한 대응하여 연결 백플레인만을 설계하므로써 8개의 4중 슬라이스까지 결합할 수 있게 한다. 이러한 배열에선 또한 도 3에 도시된 바와 같은 두 버스 시스템으로 동작할 수 있게 하며, 각각의 경우에 4중 슬라이스의 두 커플러는 동일 버스를 이용한다. 이러한 유형의 연결에서, 처리기의 개수는 단지 4, 8, 12 또는 16으로 제한된다. 동시에, 백플레인에는 시작으로부터 우측으로 이러한 유형의 연결부와 플러그가 실장될 수 있으며, 중앙처리장치의 펌 웨어 구성은 다른 모듈로의 직접연결이 사용되었는지의 여부를 결정한다. 따라서, 소비자는 초기에 단일 중앙처리장치에 의해 즉, 처리기에 대해 2 시스템으로 자신의 시스템을 확장시킬 수 있다. 만일 짝수 개 처리기가 있다면, 이들은 각각 2 처리기로 된 4중 슬라이스로 재구성될 수 있다.

개량된 백플레인은 도 5에 도시된 바와 같이, 중앙처리장치의 각각의 버스 포트가 하나의 다른 중앙처리장치에 연결되는 4 버스 시스템을 제공한다. 따라서 전체성능은 4중 슬라이스로 도달된다. 버스 시스템

은 성단(termination)(51,52)에 의해 일단부에서 종료된다. 클램퍼(53,54)는 폐쇄된 상태로 4 버스 세그먼트를 2 버스 시스템에 연결한다. 개방 상태에서, 이들 버스 시스템은 또다른 성단에 연결되며, 따라서 4 버스 시스템을 생성한다. 따라서 버스 시스템의 수는 소비자의 전체조건으로 단순한 방식으로 증가될 수 있으며, 따라서 시스템의 성능을 설치된 중앙처리장치의 수에 거의 비례하여 증대시킨다.

이 백플레인은 또한 지금까지 설명된 듀얼-처리기 중앙처리장치를 2 모듈에 분배함으로써 달성되는 중앙처리장치의 단순한 변형을 제공한다. 이들 단일-처리기 중앙처리장치의 각각은 하나의 처리기, 하나의 메모리, 하나의 커플러 및 버스포트와 직접 포트를 갖는다. 사용된 연결 백플레인은 도 5의 상부 그룹에 따른 장치이지만 두 개의 개별 버스 시스템 역할을 하는 단일 행이 물리적으로 존재하도록 점퍼 대신 고정된 성단을 갖추고 있다. 직접 인터페이스는 서로 쌍으로 연결된다. 이제 두 처리기의 각각이 직접 인터페이스에 의해 연결되고 두 버스 시스템이 존재하므로, 공지된 버스 시스템 보다 2배의 전송 용량이 달성된다. 동일한 단일-처리기 중앙처리장치는 커플러가 직접 인터페이스를 사용하지 않고, 특별한 백플레인과 함께 각각의 경우에 두 개의 단일-처리기 중앙처리장치에 의해 확장될 수 있는 컴퓨터를 형성하는 방식으로 커플러를 구성함으로써 단일 버스 시스템을 갖는 종래의 백플레인에 대해 사용될 수 있음에도 불구하고, 버스 시스템의 최대 대역폭에 의해 지금까지 보다 덜 제한되는 고 대역폭을 달성한다.

고성능 컴퓨터에서, 캐시는 중앙처리장치의 양호한 성능을 달성하기 위해 일반적으로 제공되어야만 한다. 이들 캐시는 또한 국부 데이터 액세스의 상당부분을 부담하기 때문에, 주소 인터리빙은 더욱 유리한 것으로 여겨질 수 있다. 이것은 특히 하나의 캐시라인에 대한 그레놀러리티를 증대시킨다. 캐시는 각각의 처리기에 속하며 개념적으로는 처리기 자체와 커플러 사이에 위치된다(도시되지 않음). 캐시 제어수단은 흔히 처리기 칩에 포함되므로써 처리기는 캐시에 대한 직접 연결을 가지며 커플러로의 캐시연결은 또한 처리기에 의해 행해진다.

공유된 주소 공간을 갖는 모든 처리기에 대해, 모든 메모리 내용에 대한 가상적인 동시 이용가능성 즉, 균등한 메모리를 달성하기 위해, 캐시 시스템에 바람직한 두 방법이 있다. 관련문서로는 예를들어, P. Sweazey and A.J. Smith,에 의한 논문 'A Class of Compatible Cache Consistency Protocols and their Support by the IEEE Futurebus' Proc. 13th Int. Symp. on Computer Architecture, IEEE 1986에서 찾을 수 있다.

이들 방법중의 하나는 '스누핑'으로 불린다. 이것은 캐시의 상태를 변경시키는 적어도 버스 시스템상의 모든 동작에 대한 주소가 동시에 모든 캐시 유닛에 전달되므로써 이들은 새로운 상태를 나타내는 것에 있다. 이 방법은 또한 본 발명에 따른 중앙처리장치의 경우에도 가능한 데 이는 주소가 직접연결을 통해 다른 커플러에 전달되거나 또는 버스 시스템상에 위치되는 지의 여부를 결정하기 위해 각각의 커플러가 주소를 사용할 수 있기 때문이다. 이 경우에, 본 발명의 변형이 가능하며, 여기서 두 버스 시스템(15 및 25)이 별개 주소라인을 가질지라도, 이들은 공유 데이터 라인을 사용한다. 버스 사이클의 대부분은 캐시 정렬을 위한 순 주소 사이클이기 때문에, 두 버스 시스템은 데이터 라인을 공유한다. 주소버스는 또한 주로 32 비트 폭을 가지지만 데이터 버스는 적어도 64 비트 대부분의 경우 128 또는 256 비트 폭을 가지므로, 이것은 성능을 손상시키지 않고 상당 수의 포트를 절약할 수 있다. 또한 역으로, 본 발명에 따른 주소 인터리빙을 갖춘 듀얼 중앙처리장치는 백플레인에 단지 32개의 추가 포트로 캐시 정렬속도를 배가시킨다.

본 발명에 의한 캐시 정렬을 위한 바람직한 방법은 '태그된' 또는 '디렉토리에 기초한' 방법으로 불리는 방법이다. 이 방법에서, 각각의 메모리는 처리기가 메모리 워드를 요구하고 메모리 워드를 변경시키기 위해 메모리 워드를 수신하였다면 캐시 요구에 관하여 설정되는 추가 지시자를 포함한다. 제 2 처리기가 동일한 메모리 워드(캐시라인)를 액세스한다면, 이 액세스는 제 1 처리기의 캐시로 다시 향하게 되거나 제 1 캐시의 캐시 엔트리가 무효화되어 주 메모리로 되돌려 전송될 때 까지 연기된다. 이러한 액세스 연기는 액세스 시도를 중단시키거나 반복하므로써 수행될 수 있다. '디렉토리에 기초한' 캐시 일관성 방법에서도 상당량의 데이터 전송이 처리되어야 한다. 본 발명은 이러한 데이터 전송이 단순한 방법으로 주소 버스 시스템에 대해 균등하게 분배될 수 있게 한다. '디렉토리에 기초한' 캐시 일관성 방법의 사용에 대한 추가 문서는, 논문 'Torus with slotted rings architecture for a cache-coherent multiprocessor', Proc. 1994 Int. Conf. on Parrel and Distributed Systems, pp 76-81, IEEE Comp. Soc Press 1994 에서 찾을 수 있다.

따라서, 두 개의 처리기, 메모리, 커플러, 커플러 인터페이스 및 버스 인터페이스를 포함하는 본 발명의 바람직한 변형은 긴 버스 시스템을 필요로 하지 않고 또는 단일 버스 시스템에 의해 처리기의 성능을 제한하지 않고 단일 유형의 중앙처리장치를 사용하므로써 2 내지 80개 처리기가 시스템에 구비될 수 있게 한다.

지금까지 설명된 본 발명의 바람직한 실시예는 2의 거듭제곱 즉, 처리기 슬라이스 당 2개, 4개 또는 8개 처리기에 따른 분할을 이용한다. 그러나 3(또는 그 이상) 처리기를 사용할 수도 있다. 이것은 두 개의 가능한 변형을 포함한다. 제 1 변형은 두 커플러를 포함하는 실시예에 해당하며 직접 인터페이스 및 버스 인터페이스를 갖춘 제 3 커플러도 제공한다. 다른 변형은 중앙 커플러를 위한 버스 인터페이스 및 직접 인터페이스가 없다. 모든 비국부 메모리 액세스는 인접 커플러중의 하나로 진행된다.

메모리 뱅크의 수가 2의 거듭제곱 갯수가 아니면, 메모리 뱅크의 사이즈는 비교적 적은 수의 블록이 형성되게 하기 위해 인터리빙에 대해 비교적 크도록 선택될 것이다. 이 선택은 프로그램가능 판독 전용 메모리(PROM)에 의해 수행되며, 이것의 주소라인에는 최상위 비트가 구비되며 이것의 데이터 바이트는 인터리빙된 메모리 뱅크의 수를 표현한다. 3 그룹 및 5 주소 비트의 경우에, PROM의 내용은 0, 1, 2, 0, 1, 2, 등으로 된 즉, 주소모듈 3 으로 된 32개 값으로 이루어 진다. 16 주소라인을 갖는 PROMs는 3중 주소 인터리빙의 경우에 각각의 그룹이 65536 라인으로 분할되도록 필요 속도로 이용할 수 있다. 만일 그러한 두 중앙처리장치가 서로 연결된다면, 최하위 유효비트가 중앙처리장치를 선택하기 위해 사용되며, 따라서 단순한 방식으로 6 그룹을 갖는 주소 인터리빙이 된다.

## (57) 청구의 범위

**청구항 1**

각각이 하나의 처리기(11,21), 하나의 커플러(13,23), 하나의 메모리(12,22) 및 하나의 버스 인터페이스(15,25)로 된 적어도 두 개의 그룹을 포함하고, 처리기(11,21)의 명령은 메모리(12,22)를 포함하는 공유 주소공간을 사용하며,

커플러(13,23)의 각각은 이 커플러가 속하는 그룹의 처리기(11,21), 메모리(12,22) 및 버스 인터페이스(15,25)와 상기 커플러가 속하는 그룹 이외의 다른 그룹에 속하는 적어도 하나의 커플러에 직접 연결되며 상기 커플러(13,23)는 적어도 간접적으로 상호 연결되고,

처리기(11,21)는 공유된 주소공간을 가지며 커플러는 직접연결된 처리기의 메모리 액세스를,

처리기에 의해 전달된 주소에 기초한 주소 할당으로,

직접 연결된 메모리(12,22)를 사용하거나,

액세스가능한 메모리(12,22)를 액세스하기 위해 다른 커플러(13,23)에 상기 메모리 액세스를 전송하거나,

또는 버스 인터페이스(14,15)를 통해 상기 메모리 액세스를 실행하여,

만족시키는 것을 특징으로 하는 중앙처리장치(10).

**청구항 2**

제 1 항에 있어서, 3개 이상의 상기 그룹을 포함하며, 적어도 하나의 커플러는 커플러가 행 또는 링을 형성하는 방식으로 다른 두 커플러에 직접연결되며 한 커플러는 직접적으로 연결된 커플러로 직접적으로 연결되지 않은 다른 커플러를 주소지정할 수 있으며 상기 다른 커플러가 직접적으로 연결된 커플러와 동일한 방식으로 메모리 액세스를 만족시킬 수 있게하는 것을 특징으로 하는 중앙처리장치(10).

**청구항 3**

제 1 항 또는 제 2 항에 있어서, 각각 하나의 커플러에 연결된 두 개의 직접적으로 연결된 인터페이스를 포함하며, 커플러는, 만일 두 개의 동일 중앙처리장치의 직접적으로 연결된 인터페이스가 쌍으로 연결되면, 버추얼 중앙처리장치가 적어도 4개의 상기 그룹을 갖는 방식으로 구성될 수 있으며, 상기 커플러는 링 형태로 연결되어 4개 버스 인터페이스가 형성되는 것을 특징으로 하는 중앙처리장치(10).

**청구항 4**

하나의 처리기(11), 하나의 커플러(13), 하나의 메모리(12) 및 하나의 버스 인터페이스(15) 및 하나의 직접 인터페이스(15)를 포함하고, 처리기(11)의 명령은 메모리(12,22)를 포함하는 공유 주소공간을 사용하며,

커플러(13)는 처리기(11), 메모리(12) 및 버스 인터페이스(15)와 상기 커 직접 인터페이스(14)에 연결되며,

직접 인터페이스(14)는 마찬가지로의 다른 중앙처리장치의 정확히 하나의 직접 인터페이스에 연결될 수 있으며,

커플러는 처리기의 메모리 액세스를,

처리기에 의해 전달된 주소에 기초한 주소 할당으로,

메모리(12,22)를 사용하거나,

액세스가능한 메모리(12,22)를 액세스하기 위해 중앙처리장치의 한 커플러에 직접 인터페이스(14,15)를 통해 상기 메모리 액세스를 전송하거나,

또는 버스 인터페이스(14,15)를 통해 상기 메모리 액세스를 실행하여,

만족시키는 것을 특징으로 하는 중앙처리장치(10).

**청구항 5**

제 1 항 내지 제 4 항중 어느 한 항에 있어서, 상기 주소할당은 각각의 메모리를 공유된 주소공간에서 연속블록으로서 매핑시키는 것을 특징으로 하는 중앙처리장치(10).

**청구항 6**

제 1 항 내지 제 4 항중 어느 한 항에 있어서, 상기 주소할당은 각각의 메모리가 주소공간의 적어도 두 개의 비인접 부분에 대해 분배되는 주소 인터리빙에 의해 수행되는 것을 특징으로 하는 중앙처리장치(10).

**청구항 7**

제 1 항 내지 제 5 항중 어느 한 항에 있어서, 커플러를 위한 주소할당은 주소 인터리빙에 의해 수행되며 이렇게 주소지정된 상기 커플러는 주소가 직접연결된 메모리에 속하는 지를 결정하며, 만일 속하지 않는다면 버스 시스템에 의해 미리 정해진 주소 레졸루션을 통해 메모리 액세스를 수행하는 것을 특징으로 하는 중앙처리장치(10).

**청구항 8**

제 6 항 또는 제 7 항에 있어서, 상기 주소 인터리빙은 2진 표기법으로 나타낸 주소의 비트 필드를 추출

하므로써 수행되는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 9

제 8 항에 있어서, 상기 주소 할당은 주소할당을 포함하는 최상위 유효 주소비트를 통해 상기 주소 인터페이스에 의해 수행되는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 10

제 1 항 내지 제 9 항중 어느 한 항에 있어서, 두 개의 버스 시스템의 각각은 별개의 주소 및 제어 라인을 사용하지만, 필요한 경우엔 점퍼 및 액세스 제어를 통해 각각의 버스에 의해 사용되는 공유된 버스 라인을 사용하는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 11

제 1 항 내지 제 10 항중 어느 한 항에 있어서, 직접 인터페이스(14)는 동작될 수 없지만 동작될 수 있도록 위치한 추가 구성성분을 위치시키므로써 동작될 수 있는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 12

제 3 항 내지 제 11 항중 어느 한 항에 있어서, 상이한 중앙처리장치의 직접 인터페이스는 서로 쌍으로 연결되는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 13

제 1 항 내지 제 12 항중 어느 한 항에 있어서, 적어도 2개의 개별 버스 시스템은 중앙처리장치의 버스 인터페이스를 연결하는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 14

제 13 항에 있어서, 상기 중앙처리장치는 행으로 배열되고 상기 중앙처리장치의 버스 인터페이스는 각각의 경우에 하나의 다음 중앙처리장치에 연결되는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 15

제 14 항에 있어서, 두 버스 시스템이 일 대 일로 형성되도록 연결할 수 있게 하는 점퍼가 존재하는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 16

제 15 항에 있어서, 상기 점퍼는 터미네이션 대신에 버스 시스템의 이웃단부를 연결하는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 17

제 13 항 내지 제 16 항중 어느 한 항에 있어서, 버스 시스템은 중앙처리장치의 각각의 인터페이스가 하나의 버스 시스템에 연결되는 개수로 사용되며 어떠한 버스 시스템도 동일한 중앙처리장치의 두 버스 인터페이스를 연결하지 않는 것을 특징으로 하는 중앙처리장치(10).

#### 청구항 18

제 12 항 내지 제 17 항의 배열로 다수의 중앙처리장치를 연결하는 백플레인.

#### 청구항 19

제 1 항 내지 제 11 항중의 어느 한 항의 중앙처리장치를 동작시키는 방법에 있어서,

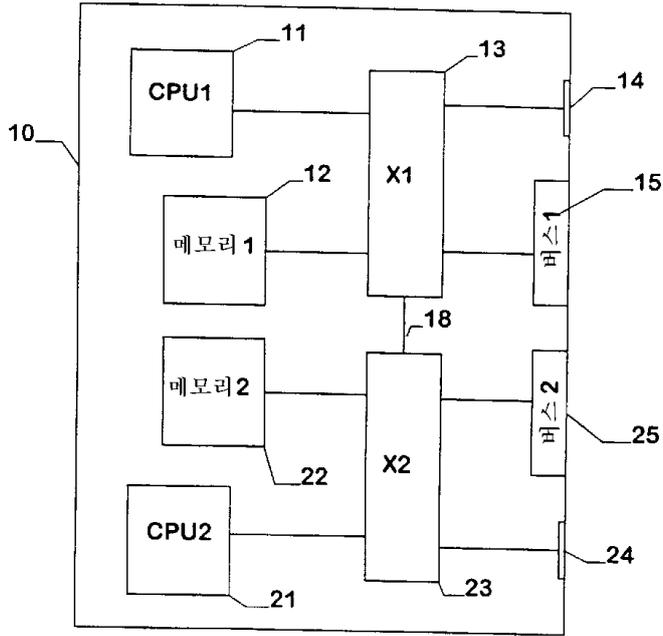
중앙처리장치는 캐시를 포함하며, 캐시의 일관성은 각각의 커플러가 직접적으로 연결된 버스에 직접적으로 연결된 메모리에 메모리 액세스를 보고하므로써 수행되고, 교대로 주소할당에 따라 버스상에 나타나는 주소를 직접적으로 연결된 프로세서에 또는 직접연결을 통하여 다른 커플러에 전송하는 스누핑에 의해 달성되는 것을 특징으로 하는 방법.

#### 청구항 20

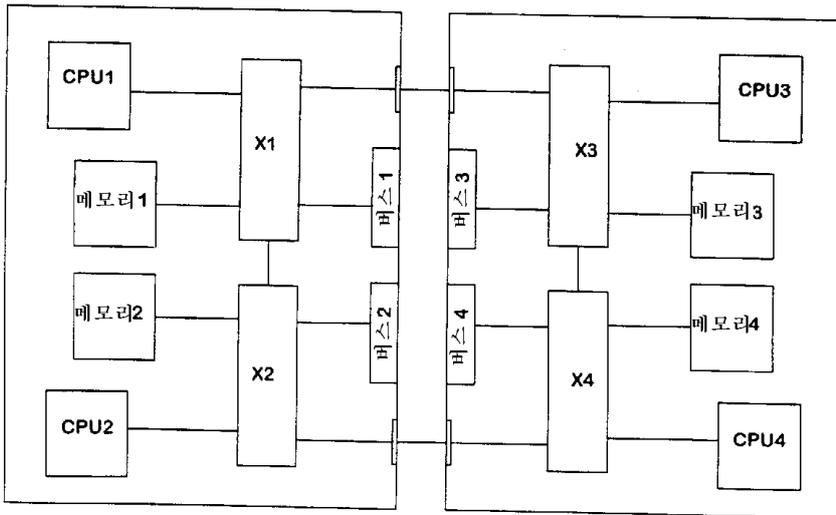
제 1 항 내지 제 11 항중의 어느 한 항의 중앙처리장치를 동작시키는 방법에 있어서, 중앙처리장치는 캐시를 포함하며, 캐시의 일관성은 디렉토리에 의해 달성되는 것을 특징으로 하는 방법.

**도면**

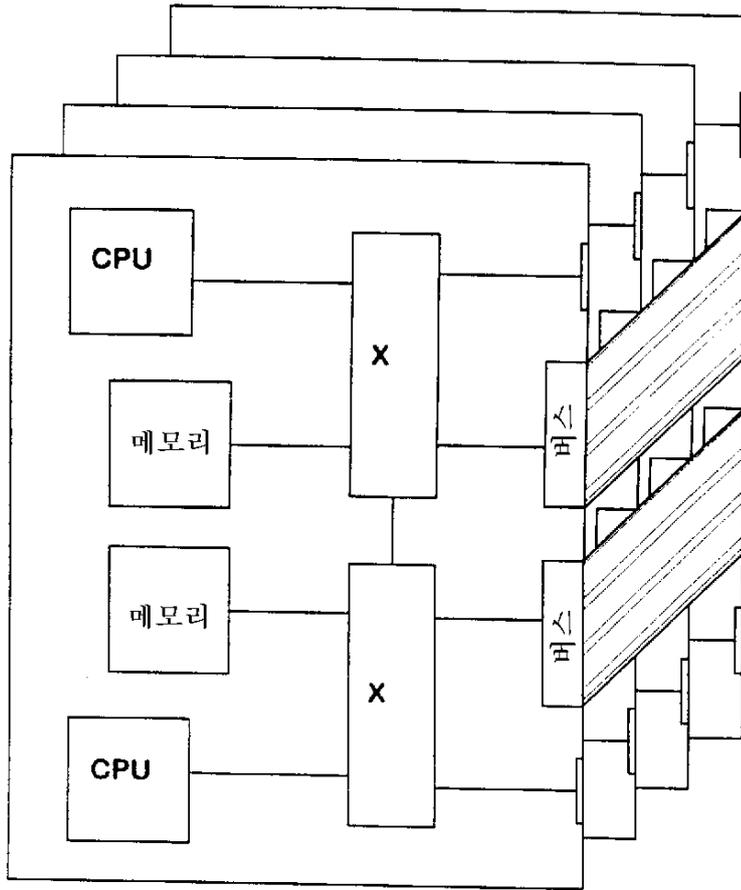
도면1



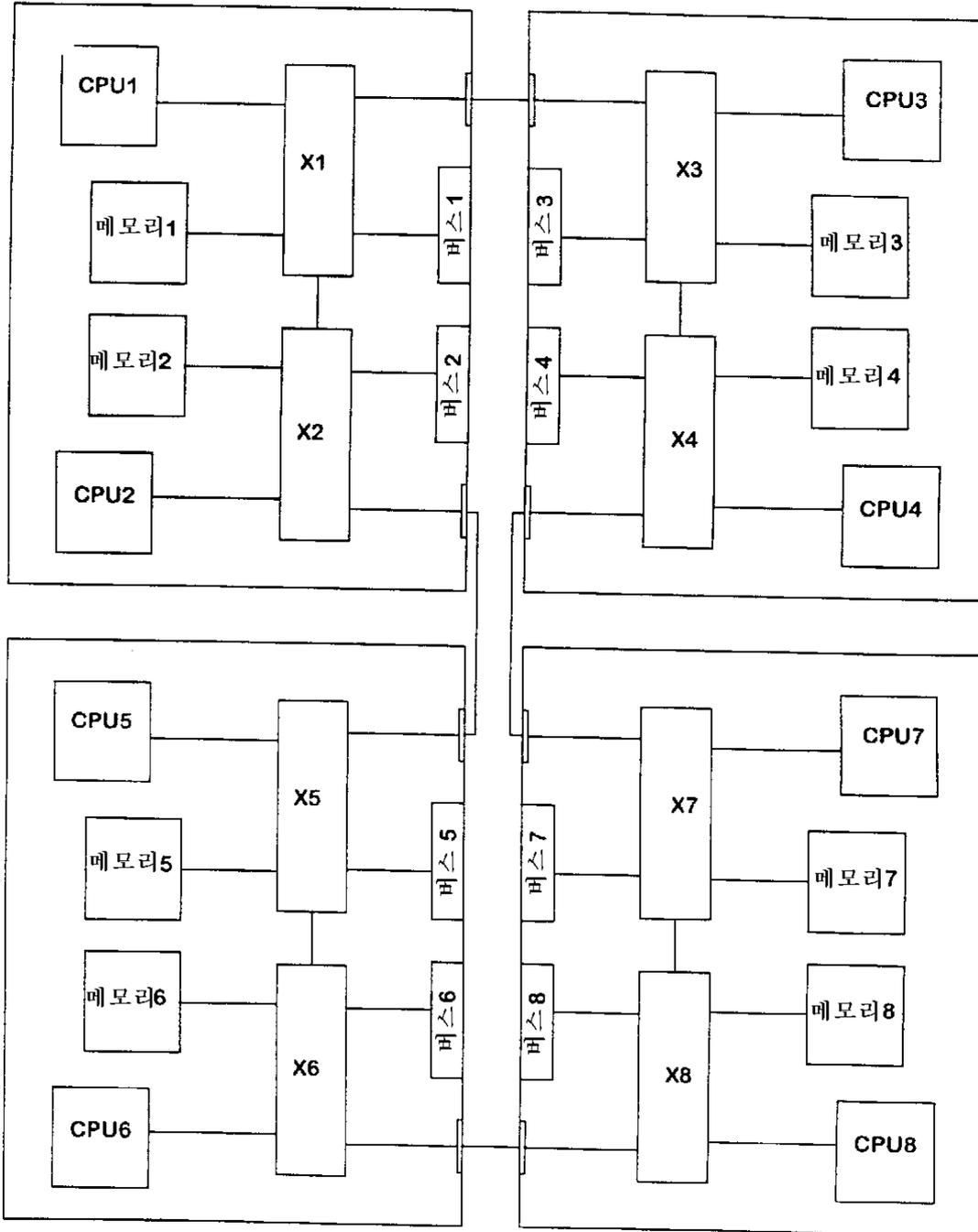
도면2



도면3



도면4



도면5

