

(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) Int. Cl. <sup>6</sup> G06F 15/16		(45) 공고일자 2000년03월 15일	
		(11) 등록번호 10-0249148	
		(24) 등록일자 1999년12월23일	
(21) 출원번호	10-1991-0015611	(65) 공개번호	특1992-0006851
(22) 출원일자	1991년09월07일	(43) 공개일자	1992년04월28일
(30) 우선권주장	90-237666 1990년09월07일	일본(JP)	
(73) 특허권자	가부시끼가이샤 히다치 세이사꾸쇼 가나이 쓰도무		
(72) 발명자	일본국 도쿄도 지요다구 간다 스루가다이 4-6 사이토 마사히코		
	일본국 히다찌 시아유가와쵸 6쵸메 20-3 유후료315 구로사와 겐이찌		
	일본국 히다찌 시미가노하라쵸 2쵸메 22-2 고바야시 요시끼		
	일본국 히다찌 시미가노하라쵸 2쵸메 24-5 반도 다타아끼		
	일본국 이바라기켄 나카군 도카이무라 미나미다이6-5 이와무라 마사히로		
	일본국 히다찌시 하나야마쵸1쵸메10-9 훗다 다까시		
	일본국 히다찌시 이시나자가쵸 1쵸메19-4-201 나가즈가 야스히로		
	일본국 히다찌시 모리야마쵸 3쵸메17-1-502 다나카 시게야		
	일본국 히다찌시 니시나루사와쵸 2쵸메16-33 다께모토 다께시		
(74) 대리인	일본국 사가미하라시 미나미다이 2쵸메8-6-202 백승남, 송재련, 한규환		

심사관 : 안철홍

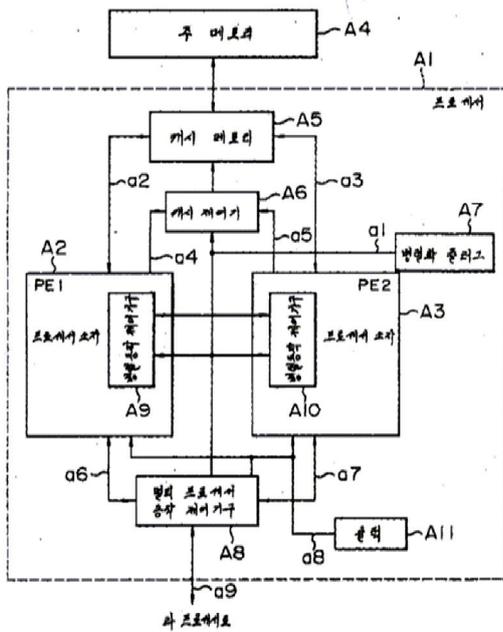
(54) 데이터처리 시스템 및 방법

요약

프로그램 병렬성으로 복사하는 데이터 처리시스템은 명령 수행을 위한 복수 프로세서 소자, 복수의 프로세서 소자에 의해 공유된 주 메모리, 복수의 프로세서 소자가 동기하여 작동 가능하게 하며 상기 복수의 프로세서 소자에 상응하여 제공하는 복수의 병렬동작 제어기구, 상기 복수의 프로세서 소자가 독자적으로 작동 가능하게 하기 위한 멀티프로세서 동작 제어기구, 기르고 병렬 작동모드 또는 멀티 프로세서 작동모드가 동작되는 것을 가르키는 값을 고정하는 플러그로 구성된다.

공유된 캐시 메모리는 공백 명령으로 수행되고 캐시 메모리에 저장된 데이터 일치가 제거되도록 캐시 제어기에 의해 제어된다.

대표도



명세서

[발명의 명칭]

데이터 처리시스템 및 방법

[도면의 간단한 설명]

제1도는 본 발명의 실시예에 따른 데이터처리시스템의 일반적인 구성을 보여주는 블록도.

제2도는 명령캐시와 데이터캐시가 분리되어 제공된 본 발명의 다른 실시예에 따른 데이터 처리시스템의 일반적인 구성을 보여주는 블록도.

제3도는 제2도에서 보여준 데이터 처리시스템내에 결합된 프로세서소자의 세부적인 내부구성을 보여주는 블록도.

제4도 및 제5도는 본 발명에 따른 데이터 처리시스템내에 각각 수요오딘 캐시제어기의 내부구성을 보여주는 블록도.

제6도는 본 발명에 따라 제고된 멀티프로세서 동작 제어기구의 기능이나 동작을 설명하기 위한 도면.

제7도는 배타적 제어기능을 실행하시 위해 설계된 멀티프로세서 동작 제어기구의 내부구성을 보여주는 구성도.

제8도는 프로세서소자 사이에서 명령을 조정하는 멀티프로세서 동작제어기구의 내부구성을 보여주는 블록도.

제9도는 프로세서 소자의 동작을 조정하기 위한 명령형식의 예를 설명하는 도면.

제10도는 동기계획 및 명령의 조합을 위한 동기기구의 내부구성도.

제11도는 버스 조정회로의 내부구성을 보여주는 블록도.

제12도는 동기계획과 명령형식의 다른 조합에 따른 동기기구의 내부구성을 보여주는 블록도.

제13도는 버스 조정회로의 내부구성을 보여주는 블록도.

제14도는 제11도 및 13도에서 보여준 하드웨어 구조에 사용되는 우선순위(시이퀀스) 결정회로의 내부구성을 보여주는 블록도.

제15도는 또다른 동기기구의 내부구성을 보여주는 블록도.

제16도 내지 20도는 레지스터 전송명령 실행시의 동작을 설명하기 위한 타이밍차트.

제21도는 정지명령과 재개명령 실행시의 동작을 설명하기 위한 타이밍차트.

제22도 및 제23도는 각각의 멀티프로세서 동작모드 및 병렬 동작모드를 변경하기위한 OS 제어를 설명하는 플로우차트.

제24도는 본 발명의 또다른 실시예에 따른 데이터 처리시스템의 일반적인 구성을 보여주는 블록도.

- 제25도는 2 판독포트와 1기록포트를 갖는 메모리의 구조를 보여주는 회로도.
- 제26도는 n 판독포트와 1기록포트를 갖는 메모리를 보여주는 블록도.
- 제27도는 명령데이터 선택고히로의 전형적인 구조를 보여주는 블록도.
- 제28도는 데이터 선택회로의 전형적인 구조를 보여주는 블록도.
- 제29도는 멀티포트 메모리를 사용하여 수행된 데이터 어드레스 변환버퍼의 구조를 보여주는 블록도.
- 제30도는 멀티플렉서를 통하여 주메모리에 종래의 멀티프로세서를 연결하는 구조를 보여주는 블록도.
- 제31도는 프로세서(i)의 전형적인 구조를 보여주는 블록도이다.

[발명의 상세한 설명]

본 발명은 캐시메모리를 구비한 컴퓨터의 고속데이터 처리에 관한 것으로, 특히 실행되는 프로그램 고유의 병렬특성(병렬성 참조)에 상응하여 프로그램 실행방법이나 모드를 변화시키는 가변구조형의 데이터 처리시스템 및 데이터 처리방법에 관한 것이다.

일반적으로, 개개의 프로세서 소자가 서로 병렬로 명령을 실행하는 복수개의 프로세서 소자를 포함하는 데이터 처리시스템과 결합하는 구성방법에는 두가지 형태가 열려져 있다. 첫번째 구성방법은 멀티프로세서형 컴퓨터시스템에 적용되고, 두번째 구성방법은 슈퍼-스칼라형 컴퓨터시스템이 적용된다.

멀티프로세서형 컴퓨터(또는 데이터처리)시스템의 경우에 있어서, 복수의 프로세서소자는 메모리기구(주메모리)나 캐시메모리를 공유하여 동작한다. 이와 관련하여 멀티프로세서 컴퓨터시스템내의 전형적거인 캐시메모리 구조설계는 JP-A-56-12726와 제임스 일, 구드먼 : "프로세서 메모리를 감소하는 캐시메모리"에 공지되어 있다. 트레픽 : 컴퓨터 구성에 대한 10차 국제 심포지움 Vol. 11, No. 3, 1983년 6월 13~17일. 이 시스템에서 개개의 프로세서소자는 다른 하나의 독립적인 독특한 프로그램을 실행한다.

멀티프로세서 컴퓨터시스템은 데이터 베이스에서의 병렬처리의 실행에서는 적합하지만, 이는 프로그램내에 포함된 복수의 명령에 대한 병렬 처리에 대해서는 고려되지 않았다. 따라서 프로세서에 의해 공유되는 데이터의 양이 증가될 때 복수의 캐시 사이에서 내장-데이터 일반성에 관한 문제가 야기된다. 예를 들면, 캐시메모리로의 데이터 기록의 무효가 자주 발생하여 캐시히트비가 낮아지는 결점을 갖는다.

대조하여, 슈퍼-스칼라 컴퓨터시스템에 있어서, 복수의 (또는 n) 프로세서소자는 병렬과 동기성이 있는 프로그램에 존재하는 복수의 명령을 실행하고, 거기에서 병렬 실행을 확실하게 완성할 수 있다. 슈퍼-스칼라 컴퓨터시스템의 일형태로서 VLIW(초장 명령(컴퓨터시스템이 알려져 있다. 멀티프로세서 시스템의 전형적인 예, 슈퍼-스칼라 시스템 및 VLIW 컴퓨터시스템은 하기에 기술된 문헌에 각각 공지되어 있다.

에스. 댄커 외 다수; "밸런스 멀티프로세서 시스템" IEEE MICRO, 1982. 2. 페이지 57~69;

케이. 무라카미 외 다수; "SIMP(단일 명령 스트림/멀티플 명령 전송); 노벨 고속 싱글-프로세서 구조", 회의록, 컴퓨터 문헌에 대한 16차 국제 심포지움, 1989. 페이지 78-85;

에이치. 하지와라 외 다수; "저레벨의 병렬성을 갖는 동적인 마이크로-프로그래머블 컴퓨터" IEEE 트랜스, C-299, No. 7, 1980. 페이지 577-595.

슈퍼-스칼라 컴퓨터시스템의 결합에 있어서, 하드웨어가 병렬로 명령을 실행하는 프로세서 소자를 사용하여 수용되어지고, 거기에서 실행되는 자체 프로그램에서 병렬성이 발견되지 않으면, 단지 하드웨어에 수용된 제한된 수의 프로세서소자만이 동작하게 되는 상황이 발생한다. 극단적으로, 연속적인 모든 명령 사이에서 병렬성이 없는 프로그램이 실행되어 질 때, n개의 프로세서소자들중 단지 하나만이 실제로 동작하게 되고, 잔여(n-1) 프로세서소자는 사용되지 않게 된다. 이러한 종류의 프로그램이 n개내에 있고, 이들 각 프로그램을 실행하는 시간이 Si(i는 1 부터 n까지의 정수를 나타냄)로 주어진다고 가정하면, 모든 프로그램을 실행하는데 필요한 시간은  $\sum S_i$  ( $\sum$ 는 총계를 나타냄)가 된다.

대조하여, 상기 언급한 형태의 프로그램이 멀티프로세서 컴퓨터 시스템으로 실행되면, 이 실행을 매우 효과적으로 실현할 수가 있다.

특히 각 프로그램의 실행 시이퀀스가 오더가 변화(작업루틴에서 흔히 발생하는 바와 같이, n개의 다른 사용자 프로세서의 실행에서와 같이)하더라도 이러한 문제가 야기되지 않는한, 또다른 하나의 독립된 n개의 프로세서 소자로 프로그램을 실행할 수가 있다. 이 경우에 각 프로그램의 각각의 실행에 필요한 최대시간 이내에 모든 프로그램의 실행이 이상적으로 완료된다. 그러나 프로세서소자 사이의 캐시메모리 및/또는 주메모리에서 경합이 발생할 수가 있기 때문에 모든 프로그램의 실행시간은 상술한 최대시간 이상이 될 것이다. 그럼에도 불구하고 멀티프로세서 컴퓨터시스템은 슈퍼-스칼라 컴퓨터시스템 보다 더 많은 이점을 갖는다.

그러나 멀티프로세서 컴퓨터시스템은 적절치 못한 상황에 직면할 수도 있다. 예를 들면, 상기 언급한 상황하에서 프로세서의 수가 실행되는 프로그램의 수보다 많아 복수의 프로그램이 동시에 실행되지 않게 될 수도 있다. 특히 멀티프로세서 컴퓨터시스템의 경우에 있어서, 실행될 프로그램의 수에 사용하는 병렬성만을 실현할 수 있다. 따라서, 프로그램의 수가 p개로 표시되고 프로세서소자의 수가 n으로 표시될 때, 프로세서 소자의 수(n)가 프로그램의 수보다 큰 경우, 프로세서소자(n-p)는 무효하게 된다. 그와반대로, 슈퍼-스칼라 컴퓨터시스템에서는 프로그램의 수가 프로세서의 수보다 더 작을 때, 병렬성이 싱글 프로그램에서 얻어지기 때문에, 프로세서소자가 무효하게 되는 것을 효과적으로 방지한다.

상기에서 이해할 수 있는 바와 같이, 최적의 컴퓨터 구조는 실행되는 프로그램에 따라 달라지게 된다.

본 발명의 목적은 그 프로그램에 본질적으로 내재하는 병렬성에 따라 최적의 프로그램을 실행하는 컴퓨터시스템을 제공하고 실현하는데 있다.

본 발명의 다른 목적은 슈퍼-스칼라 컴퓨터시스템(여기에서 병렬동작모드로서 참조됨)중의 어느 하나의 데이터 처리시스템의 동작모드를 실행할 수 있는 하드웨어를 실현하는데 있다.

본 발명의 또다른 목적은 병렬 동작모드를 실현하는데에 채택된 VLSI설계기술이 멀티프로세서 동작모드에서 효율적으로 이용될 수 있는 데이터 처리시스템을 제공하는데 있다.

본 발명의 또다른 목적은 복수의 프로세서 소자에 의해 많은 양의 데이터가 공유되는 경우에도 캐시히트를 증가할 수 있는 데이터 처리시스템 및 데이터 처리방법을 제공하는데 있다.

본 발명의 또다른 목적은 복수의 프로세서로 병렬처리를 실현함으로써 데이터 처리속도를 향상시킬 수 있는 데이터 처리시스템 및 데이터 처리방법을 제공하는데 있다.

본 발명의 또다른 목적은 캐시불일치의 문제를 해결하고, 빠른 처리속도의 실현을 가능하게 한 멀티프로세서 시스템을 제공하는데 있다.

상기 관점과 기타의 목적은 기술된 순서에 따라 하기에 나타나게 될 것이며, 본 발명의 일반적인 관점에 따른 데이터 처리시스템은 n개의 프로세서소자를 서로 독립적으로 동작시키는 멀티프로세서 동작 제어기구와, n개의 프로세서소자를 기본클럭에 동기하여 서로 병렬로 동작시키는 병렬동작 제어기구로 이루어진다.

병렬동작 제어기구는 각 프로세서 소자내에서 결합된다. 다른 한편, 프로세서소자 사이의 데이터, 어드레스, 제어신호를 전송하기 위한 통신버스와, 통신버스를 효율적으로 관리하는 버스조정회로와, 프로세서소자 사이의 동기를 행하는 동기기구가 멀티프로세서 동작 제어기구의 내부에 결합되어 제공된다.

멀티프로세서 동작 제어기구는 데이터처리시스템을 n개의 프로세서소자가 서로 독립적으로 동작하는 멀티프로세서 동작모드내에서 동작시킨다. 더구나 각 프로세서소자에 의해 실행되는 프로그램이 고속으로 통신과 동기를 허용하기 위해 멀티프로세서 동작제어기구가 제공된다.

병렬동작 제어기구는 n개의 프로세서소자 사이의 공기를 행하고, 병렬동작모드에서 레지스터는 각 프로세서소자에 의해 변환된다.

이를 위하여, 레지스터 명령을 실행할 때, 프로세서소자내에 각각 결합된 병렬동작제어기구는 관련된 프로세서소자내의 레지스터에 있는지의 여부를 결정하여, 그 결정이 긍정이면, 레지스터의 내용이 명령을 실행하는 프로세서소자로 보내진다.

프로그램 고유의 동기성이 고속인 경우, 그 프로그램은 병렬 실행모드에서 실행되는 된다. 다른 한편, 복수의 프로그램이 프로그램 고유의 낮은 동기성에도 불구하고 병렬 실행을 행하게 되면, 데이터 처리시스템은 멀티프로세서 동작모드내에서 동작하게 된다. 이를 위하여 본 발명의 데이터 처리시스템에는 멀티프로세서 동작모드나 또는 병렬 동작모드에서 동작하는 것을 나타내는 병렬화 플러그가 제공된다. 병렬화플러그가 멀티프로세서 동작모드를 지정하면, 병렬동작 제어기구의 동작은 일시 중지하게 된다. 다른 한편, 병렬화플러그가 병렬 동작모드를 지정하면 상술한 멀티프로세서 동작 제어기구의 동작이 일시 중지된다.

통신버스는 데이터의 전송과 멀티프로세서 동작모드내의 프로세서 소자의 협력을 통하여 프로그램을 실행하는데 이용된다.

버스조정회로는 통신버스를 직접 이용하는 제어나 관리와 같은 기능을 수행한다. 특히 프로세서소자중의 하나가 통신버스를 통해 데이터전송을 실행할 때, 버스조정회로는 우선순위 레벨에 따라 프로세서소자에 통신버스의 직접 사용을 부여한다.

동기기구는 전송된 데이터를 보존하고, 동기뿐 아니라 프로세서소자 사이의 전송데이터에 의해 프로세서소자의 재개 및 정지를 수행한다.

병렬 동작모드(슈퍼-스칼라 컴퓨터시스템)를 실현하기 위하여, 각 프로세서 소자를 동일 LSI칩이나 기판에 재치할 수 있다. 복수의 프로그램이 서로 협력하여 실행되는 경우에, 서로 근접한 프로세서소자의 배치에 의하여 복수의 프로그램 사이의 고속동기 및 통신에 가능하다.

본 발명의 또다른 관점에 따라, 데이터 처리시스템은 어드레스 정보를 출력하는 복수의 프로세서와, 각 프로세서로부터 지정된 어드레스로 어드레스 정보를 전송하는 적어도 복수의 판독포트를 구비한 어드레스 전송버퍼와, 복수의 캐시뱅크내에 분할된 데이터 저장영역을 갖는 캐시메모리와, 프로세서로부터의 어드레스정보에 의해 지정된 뱅크를 선택하기 위한 뱅크선택기와, 어드레스 전송버퍼의 출력정보와 프로세서로부터의 어드레스 정보에 따라 데이터의 존재여부를 결정하는 데이터 선택기로 이루어지고, 거기에서 결정결과가 긍정일 때, 지정된 프로세서로 선택된 뱅크내의 데이터를 전송한다.

본 발명의 또다른 관점에 따라, 데이터 처리시스템은 복수의 프로세서와, 프로세서로부터 지정된 명령어 어드레스로 명령어어드레스 정보를 전송하는 적어도 복수의 판독포트를 구비한 명령어어드레스 전송버퍼와, 프로세서로부터 지정된 데이터 어드레스로 데이터 어드레스 정보를 전송하는 적어도 복수의 판독포트를 구비한 데이터 어드레스 전송버퍼와, 복수의 명령뱅크내에 분할된 데이터 저장영역을 갖는 명령 캐시메모리와, 복수의 데이터 어드레스 뱅크내에 분할된 데이터 저장영역을 갖는 데이터 캐시메모리와, 프로세서로부터의 명령 어드레스정보에 의해 지정된 명령 뱅크를 선택하기 위한 명령뱅크 선택기와, 프로세서로부터의 데이터 어드레스 정보에 의해 지정된 데이터 뱅크를 선택된 뱅크내에 데이터가 존재하는지의 여부를 명령어어드레스 전송버퍼의 출력정보와 프로세서로부터의 명령어어드레스 정보에 따라 결정하는 명령결정기구

와, 데이터뱅크 선택기에 의해 선택된 뱅크내에 데이터가 존재하는지의 여부를 데이터어드레스 전송버퍼의 출력정보와 프로세서로 부터의 데이터 어드레스 정보에 따라 결정하기 위한 데이터 결정기구와, 명령 결정기구에 의한 결정결과가 긍정일 때 명령뱅크 선택기에 의해 선택된 뱅크로부터 지정된 프로세서로 데이터를 전송하기 위한 명령-오리엔트 데이터 선택기와, 데이터 결정기구에 의한 결정결과가 긍정일 때 데이터 뱅크 선택기에 의해 선택된 뱅크로부터 지정된 프로세서로 데이터를 전송하기 위한 데이터-오리엔트 데이터 선택기로 이루어진다.

상기 기술한 두 경우의 데이터 처리시스템에 있어서, 어드레스 전송버퍼의 어드레스 전송영역은 가상페이지와 물리적 페이지로 분할 되고, 캐시메모리의 메모리영역은 물리적 페이지와 데이터 영역으로 분할되며, 뱅크 선택기는 프로세서로 부터의 어드레스 정보에 의해 지정된 캐시 뱅크의 물리적 페이지의 정보를 선택하는 제1선택기로 구성된다. 더욱이, 결정기구는 어드레스 전송이 성공적으로 종료되었는지의 여부를 결정하기 위해 어드레스 전송버퍼의 가상페이지의 정보와 프로세서로 부터의 어드레스 정보를 비교하는 제1비교기와, 물리적페이지의 정보에 상응하는 데이터가 캐시내에 존재하는지의 여부를 결정하기 위해 어드레스 전송버퍼의 물리적페이지로 부터의 정보와 제1선택기의 출력정보를 비교하는 제2비교기와, 제1 및 제2비교기에 의한 두 결정의 결과가 긍정인지의 여부를 결정하는 제3 비교기를 포함한다. 부가하여, 데이터 선택기는 프로세서로 부터의 어드레스 정보에 의해 지정된 뱅크내에서 데이터를 선택하는 제2선택기와, 제3 비교기가 긍정일 때 결정의 긍정결과와 출력에 대응하여 지정된 프로세서로 제2선택기에 의해 선택된 데이터를 전송하는 제3 선택기를 포함한다.

본 발명에 따른 데이터 처리시스템의 바람직한 실시예에 있어서, 뱅크선택기는 뱅크가 복수의 프로세서에 의해 동시에 지정될 때 지정된 우선순위에 따라 뱅크를 선택하도록 구성된다.

본 발명을 실행하기 위한 바람직한 모드에 있어서, 어드레스 전송버퍼는 적어도 하나의 기록포트와 복수의 판독포트를 구비한다. 더욱이, 캐시메모리의 각 뱅크를 하나의 기록포트와 하나의 판독포트를 구비한 메모리로 구성해도 된다.

본 발명을 실행하기 위한 또다른 바람직한 모드에 있어서, 뱅크 명령 어드레스 전송버퍼와 명령 캐시메모리 각각은 하나의 기록포트와 하나이 판독포트를 구비한 반면에, 데이터 어드레스 전송버퍼는 적어도 하나의 기록포트와 복수의 판독포트를 구비한다.

더욱이, 본 발명을 실행하기 위한 모드에 따라, 각 프로세서는 복수의 연산장치와, 어드레스라인과 데이터라인에 연결된 복수의 명령 디코더와, 프로세서로 정보전송을 실행하는 복수의 명령 디코더와, 프로세서로 정보전송을 실행하는 복수의 포트를 구비한 레지스터 파일과, 레지스터 화일과 각 연산장치에 연결된 복수의 메모리 어드레스 레지스터와, 레지스터 화일과 각 연산장치에 연결된 복수의 메모리 데이터 레지스터를 포함한다.

각 구성소자는 싱글 반도체 집적회로 기판에 장착하는 것이 바람직하다.

데이터 처리시스템은 어드레스 전송버퍼와 주메모리가 서로 연결된 멀티플렉서와, 캐시메모리와 줌가 서로 연결된 멀티플렉서를 포함한다.

상술한 데이터 선택기는 결정기구에 의한 결정의 결과가 부정일때 주메모리로부터 전용메모리로 데이터를 전송한다.

어드레스 전송버퍼가 복수의 판독포트를 갖는 멀티포트 구성이고 캐시메모리가 복수의 캐시뱅크로 분할되어 있기 때문에, 각 프로세서 (또는 프로세서소자)가 다른 뱅크로 액세스하는 동안에 어드레스 전송버퍼로 동시에 액세스 할 수 있다. 이러한 특징에 의하여, 각 프로세서소자는 동시적인 액세스가 동일한 뱅크에서 발생하는 경우를 제외하고는 서로 독립적으로 동시에 액세스 할 수 있다. 이러한 특징에 의하여, 각 프로세서소자는 동시적인 액세스가 동일한 뱅크에서 발생하는 경우를 제외하고는 서로 독립적으로 동시에 캐시뱅크를 액세스 할 수 있다. 따라서 많은 양의 데이터가 각 프로세서 소자에 의해 공유되는 경우에도, 캐시히트비는 증가되므로 각 프로세서소자는 각각의 병렬처리를 실행할 수 있다.

본 발명을 첨부된 도면을 참조하여 바람직한 실시예에 따라 상세하게 기술한다.

제1도는 본 발명의 실시예에 따른 데이터 처리시스템의 일반적인 구성을 보여주는 블럭도이다. 이 도면에서 기호 A2, A3은 명령에 의한 연산동작을 실행하고 주메모리로 참조된 메모리장치나 캐시메모리에서 데이터를 인출하는 각각의 프로세서소자를 나타낸다. 제1도에 보여준 데이터 처리시스템내에 결합된 프로세서소자의 수를 두개로 가정하였지만, 본 발명에서 프로세서소자가 어떤 특정한 수로 결코 제한 되지 않는 것으로 이해되어야 할 것이다. 편의 를 도모하기 위하여, 프로세서의 수가 두개라는 가정하에서 다음에 기술한다. 메모리장치나 주메모리(A4)는 데이터와 같은 프로그램을 저장하고, 캐시메모리(A5)는 명령을 임시 저장하고 메모리(A4)에서 데이터를 인출한다. 여기에서 캐시메모리(A5)가 멀티포트 캐시로 구성된다고 가정한다. 특히 데이터 처리시스템(A1)내에 결합된 프로세서소자(즉, A2, A3)의 수를 n(1 보다 큰 정수)개로 나타낼 때, 멀티포트 캐시로 수행되는 캐시메모리(A5)는 n개의 입/출력 포트나 n개의 입력포트 및 n개의 출력포트와 부가하여 n개의 어드레스 입력포트(하기에 기술되어 지는 바와 같이, 캐시제어기(A6)에 의해 액세스됨)가 제공된다. 기호 A11은 프로세서소자(A2, A3)와 멀티프로세서 제어동작기구(A8)에서 사용되는 클럭신호를 발새하는 기본클럭을 나타낸다.

일반적으로 프로세서에 의한 전형적인 데이터 처리시스템은 두개의 동작모드를 갖는다. 이들 두 동작모드는 복수의 프로세서소자(즉, 설명된 프로세서 시스템의 두번째 경우)가 서로 독립적으로 동작하는 멀티프로세서 동작모드와 복수의 프로세서소자가 기본클럭에 동기되어 병렬로 동작하는 명령 동작모드이다. 상기 기술한 두개의 동작 모드를 제어하기 위한 기구로서, 두 형태의 제어기구, 즉 멀티프로세서 동작 제어기구(A8)와 병렬동작 제어기구(A9, A10)가 각각 제공된다. 이들 두형태의 제어기구에서 병렬동작 제어기구(A9, A10)는 이들 프로세서소자(제3도를 참조하여 하기에 기술됨)에 의해 공유되는 레지스터와 프로세서소자 사이의 동기성을 제어한다. 레지스터의 공유 결합에 있어서, 병렬동작 제어기구(A9, A10)는 관련된 프로세서소자에 의하여 실행되는 명령 사이에서 레지스터에 대한 경합(레지스터 충돌로 참조

뎁)의 발생을 체크한다. 여기세어 레지스터 충돌 장소를 찾게되면, 레지스터 충돌을 피하기 위하여 프로세서소자중의 하나에 의한 명령 실행이 지연된다. 레지스터 충돌의 체크도 제거도 실행하지 않는 그러한 하드웨어 구성이 생각될 수 있다. 그러나, 이것은 데이터 처리시스템이나 프로세서(A1)가 슈퍼-스칼라 프로세서로 동작하지 않고, VLIW(초장 명령워드)프로세서로서 동작하는 것을 의미한다.

간단하게 데이터 처리시스템인지 프로세서(A1)인지를 지시하고, 멀티프로세서 동작모드나 병렬 동작모드 내에서 명령을 실행하기 위하여 병렬화 플러그(A7)가 제공된다. 이 플러그(A7)가 멀티프로세서 동작모드를 가르키면 각 프로세서소자(A2, A3)내에 결합된 병렬동작 제어기구(A9, A10)의 동작은 정지된다. 대조하여, 플러그(A7)가 병렬 동작 모드를 가르키면, 멀티프로세서 동작 제어기구(A8)의 동작이 정지된다.

병렬화 플러그(A7)의 세팅은 간단하게 OS나 동작시스템에 의해 실행할 수 있는 특권명령에 의해 실행된다. 프로세서 시스템의 의부에 제공된 입력신호를 명령과 플러그 대신에 사용해도 된다. 동작시스템이나 OS는 후술되어지는 바와 같이, 프로그램 병렬성의 범위에서 결정되는 결과에 의거하여 병렬화 플러그(A7)를 세트할 수 있다.

제1도에서, 기호 A6은 병렬화 플러그(A7)에서 세트된 값에 따라 캐시메모리(A5)내에 기록하거나 캐시메모리로 부터의 인출데이터수와 폭을 변경하거나 변화하는 캐시 제어를 나타낸다. 병렬화 플러그(A7)가 멀티프로세서 동작모드를 가르키게 되면, n개이 프로세서소자(설명된 시스템의 경우 두개의 프로세서소자)는 데이터와 명령의 판독/기록 동작을 동시에 실행한다. 결과적으로, 데이터/명령 판독/기록 동작을 하나의 프로세서소자에 의해 실행되는 비트폭 K로 나타내면, 캐시메모리(A5)를 액세스 하기 위한 조건은 캐시메모리가 각 K비트의 n의 입/출력과 n개의 어드레스 입력을 이용하여 액세스 할 수 있도록 세트되어진다. 다른 한편, 병렬화 플러그(A7)가 병렬 동작모드를 가르키게 되면, n개의 프로세서소자는 서로 동기되어 명령/데이터 판독/기록을 실행한다. 이 경우에, 하나의 어드레스 입력을 이용하여 하나의 (n x k) 비트를 출력하는 캐시메모리(A5)로부터 명령이 인출된다. 이것은 병렬 동작모드에서 캐시메모리(A5)를 위한 데이터 판독/기록 동작이 멀티프로세서 동작모드의 경우와 유사하게 실행된다는 것을 의미한다.

각각의 병렬동작 제어기구(A9,A10)가 결합된 프로세서소자(A2,A3), 캐시메모리(A5), 캐시 메모리 제어기(A6), 병렬화 플러그(A7), 멀티프로세서 동작 제어기구(A8) 및 기본 클럭 (A11)은 기호 A1로 표시된 프로세서(또는 데이터 처리 시스템)로 구성되어 서로 작용한다.

상기 열거된 프로세서(A1)의 다양한 기능부분을 각각 다른 기판에 설치하여 실현할 수도 있지만 상술된 실시예의 경우 프로세서(A1)가 단일기판에 장착되거나 또는 단일칩 LSI로 형성되어 수용된다고 가정한다. 제1도에서 프로세서(A1)가 1:1로 상응하여 주메모리(A4)에 연결되는 것을 보여준다. 여기에서 A1에 유사한 복수의 프로세서소자는 공유메모리형 멀티프로세서 데이터 처리시스템을 실현할 수 있도록 주메모리(A4)에 연결해도 된다.

제1도를 참조하면, 기호 a1~a9는 데이터라인, 어드레스라인, 제어라인 및 신호라인을 나타낸다. 특히, 기호, a1은 멀티프로세서 동작모드나 병렬동작모드중의 하나를 가르키는 원-비트 정보가 캐시제어기(A6), 각 병렬동작제어기구(A9, A10)로 전송되는 병렬화 플러그(A7)의 출력라인을 가르킨다. 기호 a2, a3는 프로세서소자(A2, A3)와 캐시메모리(A5)사이의 데이터와 명령을 전송하는데 사용되는 데이터 라인을 가르키고, 각 데이터 라인(a2, a3)은 참조폭 즉, 32 비트~128 비트의 명령을 갖는다. 기호 a4, a5는 데이터와 명령의 어드레스를 지시하는데 사용되는 어드레스라인을 가르킨다. 특히, 캐시제어기(A6)는 병렬화 플러그(A7)에 의해 지시되는 동작모드에 의하여 프로세서소자(A2, A3)로 부터의 어드레스신호(a4, a5)를 캐시메모리(A5)에 전송한다. 기호 a6, a7은 프로세서소자(A2, A3)와 멀티프로세서 동작 제어기구(A8)를 내부적으로 연결하는 신호라인을 가르킨다. 따라서, 이들 신호라인(a6, a7)은 단지 멀티프로세서동작모드로 사용된다. 각 신호라인(a6, a7)은 각각 어드레스라인, 데이터라인 및 제어라인을 포함한다. 기호 a8은 클럭신호가 프로세서소자(A2, A3)와 멀티프로세서 동작 제어기구(A8)로 전송되는 기본클럭(A11)의 출력신호 라인을 가르킨다. 마지막 으로 기호 a9는 프로세서(A1)의 구조와 유사한 복수의 프로세서를 포함하는 멀티프로세서 데이터 처리 시스템내의 멀티프로세서 동작 제어기구 사이에서 통신을 실행하는 제어라인과 데이터 라인을 가르킨다.

제2도는 제1도에서 보여준 캐시메모리(A5)가 명령캐시와 데이터캐시로 분할된 본 발명의 다른 실시예에 따른 프로세서(또는 데이터처리시스템)의 일반적인 구성을 보여주는 블럭도이다. 제2도에서 기호 A1~A4, A6~A11, a1 및 a4~a9는 제1도에 참조된 기호와 동일한 부분이나 기구 및 신호라인을 나타낸다. 제2도에서 기호 B1,B2는 서로 분리되어 명령과 데이터를 출렁하는 명령캐시와 데이터캐시를 각각 가르킨다. 기호 B3는 명령이 전송될 때 주메모리(A4)와 명령캐시(B1)사이를 연결하기 위해 주메모리와 캐시메모리(B1, B2)사이에서 실행되는 블럭전송내에 개재된 선택기를 가르키고 데이터가 전송될 때 데이터 캐시(B2)와 주메모리(A4)사이를 연결한다.

기호 b1;b2 및 b3;b4는 데이터라인을 가르킨다. 이 데이터라인(b1,b2)은 각각 프로세서소자(A2,A3)와 명령캐시(B1)사이의 명령을 전송하는데 사용된다. 다른 한편, 데이터라인(b3,b4)은 각각 프로세서소자(A2,A3)와 데이터캐시(B2)사이의 데이터 전송에 사용된다. 기록동작이 프로세서소자(A2, A3)로부터 명령캐시(B1)로 발생하지 않기 때문에, 명령캐시(B1)에서 연장되는 데이터라인(b1,b2)은 단지 출력모드만을 갖는다.

명령캐시(B1)와 데이터캐시(B2)는 n개의 어드레스 입력포트와 n개의 입/출력포트를 갖는 것으로 가정된 멀티포트 캐시로 각각 구성된다(그러나, 명령캐시(B1)는 단지 출력포트만을 가짐). 데이터캐시로서 멀티포트캐시를 수용함으로써, 멀티프로세서 동작모드와 병렬 동작모드 양쪽 모두로부터 분리되거나 또는 독립적으로 n개의 프로세서소자에 의해 데이터의 판독과 기록이 실행된다. 결과적으로, n개의 어드레스를 데이터 캐시(B2)로 입력하기 위하여, 각 프로세서소자로 부터 연장되는 어드레스라인은 어떤 변경도 없이 사용될 수 있다. 멀티프로세서 동작모드내의 명령캐시로 부터의 명령판독은 서로 독립적으로 각 프로세서 소자에 의해 실행되고, 반면에 병렬 동작모드에서 각 프로세서소자는 실행을 위해 n개의 명령을 연속적으로 인출한다. 따라서 병렬화 플러그(A7)가 멀티프로세서 동작 모드를 지시하게 되면, n개의 프로세서소자의 어드레스는 서로 독립적으로 명령캐시(B1)에 직접 입력된다. 다른 한편, 병렬화플러그(A7)가

병렬 동작모드를 지시하면, 단지 제1프로세서소자의 어드레스 라인만이 사용된다. 상기 언급한 제어를 실행하기 위해 수용된 캐시제어기(A6)의 구조는 제4도를 참조하여 후술한다.

제3도는 프로세서소자(A2,A3)(병렬동작 제어기구(A9,A10) 각각의 배타적 구성소자)의 내부구조를 설명하는 블록도이다. 설명되는 목적을 위해 자체 프로세서소자(A2,A3)가 병렬로 두개의 명령을 실행할 수 있는 슈퍼-스칼라 프로세서로 구성된다고 가정한다. 특히, 프로세서 시스템(A1)의 멀티프로세서 동작모드에서 각 프로세서소자가 서로 독립하여 병렬로 두개의 명령을 실행하는 반면에, 병렬 동작모드에서 프로세서소자는 기본클럭(A11)에 동기되어 병렬로 네개의 명령을 실행한다. 제3도에서 기호 A1~A4, A6~A8, A11, a1 및 a4~a9 는 제1도에서 나타난 것과 같은 동일한 기능부분이나 기구 및 신호라인을 나타낸다. 더욱이 B1~B3 및 b1~b4는 제2도에서 기호로 나타난 것과 같은 동일한 기능부분과 제어신호로 나타낸다.

제3도에서 기호 D1~D4는 명령캐시(B1)로 부터 인출되는 명령을 일시적으로 홀딩하는 각각의 명령버퍼 레지스터를 가르킨다. 기호 D5~D8는 명령버퍼 레지스터(D1~D4)내에 위치하는 명령을 해독하는 각각의 디코더를 가르킨다. 기호 D9~D10은 멀티프로세서 동작모드와 병렬 동작모드내의 디코더에서 다른제어를 실행하기 위해 제공된 선택기를 가르킨다. 기호 D11, D14는 연산장치(ALU)를 나타내고, D15와, D16은 상술한 두개의 연산장치(D11;D12 및 D13;D14)에 의해 공유되는 각 레지스터화일을 나타낸다. 기호 D17은 프로세서소자(A2,A3)사이에서 레지스터의 값을 전송하는 게이트를 나타낸다.

더욱이, 기호 d1,d2는 각각의 프로세서소자(A2,A3)에서 연산장치(D11;D12 및 D13;D14)가 레지스터 값을 조회하는 판독버스를 나타내고, 기호 d3,d4는 레지스터 화일의 레지스터에 데이터를 저장하기 위해 사용되는 저장버스를 나타낸다. 부가하여 병렬 동작모드에서 프로세서소자에 의해 레지스터 값을 조회하는 글로벌 판독버스(d5)와 글로벌 저장버스(d6)가 제공된다. 마지막으로, 기호 d7~d10은 명령디코더(D5~D8)를 제어하는 제어라인을 나타낸다. 제3도에서 보여주는 바와같이 선택기(D9,D10), 게이트(D17) 및 신호라인(d5~d10)은 병렬동작 제어기구(A9,A10)를 구성한다.

각 프로세서소자(A2,A3)는 기본클럭에 동기되어 병렬로 두개의 명령을 각각 실행하는 한편, 멀티프로세서 동작모드에서 서로 독립하여 병렬로 두 명령을 실행한다. 따라서, 명령이 비트 길이 k를 가질 때, 명령버퍼 레지스터(D1;D2 및 D3;D4)에 명령캐시(B1)를 연결하는 각 신호라인(b1, b2)은 두 개의 명령에 상응하는 (2 x k)비트의 데이터폭이 필요하게 되고, 따라서 각각 k비트의 명령이 각 레지스터(D1~D4)로 보내진다. 이제부터 제1프로세서소자(A2)가 병렬로 두 명령을 실행하는 동작에 대하여 살펴본다. 제1 및 제2 명령버퍼 레지스터(D1, D2)에 의해 홀딩된 명령은 각각 제1 및 제2디코더(D5,D6)에 의하여 해독된다.

해독된 명령에 따라서 연산장치(D11,D12)와 레지스터 화일(D15)이 제어된다. 명령 오퍼랜드에 의해 지정된 레지스터 내용은 제1레지스터 파일(D15)로부터 판독버스(d1)를 통해 인출되고, 각각 제1 및 제2연산장치(D11,D12)에 의하여 레지스터화일(D15)로부터 인출된 데이터에 병렬 명령의 동작코드에 의해 지정된 연산동작이 실행된다. 그런다음 저장버스(d3)를 통하여 레지스터 화일(D15)내에 연산동작의 결과가 저장된다.

이 경우에 프로세서소자(A2)에 의해 인출되는 복수의 명령내에 병렬로 실행되는 명령의 조합이 존재함을 알수 있다. 이것이 레지스터 경합이나 충돌이 제1 및 제2 명령 사이에서 발생하는 전형적인 경우이다. 특히, 제1 명령의 종착 레지스터(즉, 제1 명령으로 실행되는 연산동작의 결과를 저장하도록 정해진 레지스터)가 제2명령의 소오스 레지스터(즉, 제2 명령의 실행으로 조회되는 값을 홀딩하도록 정해진 레지스터)로 지정될 때, 이들 두 명령을 병렬로 실행하는 것이 불가능하게 된다. 바꾸어 말하면, 단지 제1 명령만이 실행되고, 제1 명령의 실행결과가 구해질 때까지 제2명령의 실행은 대기된다. 병렬동작 제어기구(A9)(제1도 및 제3도 참조)는 프로세서소자 사이에서 레지스터가 공유되도록 할 뿐만 아니라 상기 언급한 바와 같이 레지스터 충돌을 해소하거나 클리어링 하기 위한 기능이나 룰을 구비한다. 데이터 처리시스템의 경우의 고려하에서, 프로세서소자(A2)의 내부에서 병렬처리가 실행된다고 가정하면, 레지스터 충돌의 발생이 체크된다(즉, 프로세서소자(A2)는 슈퍼-스칼라 동작을 실행한다). 종래의 프로세서에 적합시키기 위해서는 레지스터 충돌이 해소되어야 한다. 이에비해, 프로세서(A2)가 VLW 프로세서로서 병렬 처리를 실행하는 경우는, 어떤 레지스터 충돌도 발생할 수 없는 명령 문자열이 컴파일러에 의해 발생되기 때문에 해소되어야 할 레지스터 충돌문제가 야기되지 않는다.

A2가 VLW프로세서로서 동작하는 경우에는 명령캐시(B1)로부터 1개의 명령이 독출된다. 이 1개의 명령은 복수의 연산명령 필드로 구성되어 있는 것이고, 연산기(D11,D12)에서 실행되는 명령을 각각의 필드에 할당한다. 명령캐시(B1)으로부터 독출된 명령은 연산명령필드마다 각각 명령레지스터(D1,D2)에 저장되고, 디코더(D5,D6)에 의해 디코드되고, 연산기(D11,D12)에서 실행된다.

첫째, 병렬동작 제어기구(A9)에 의해 레지스터 충돌이나 경합을 클리어링 하는것에 대해 기술한다. 제1 명령(즉, 명령레지스터(D1)에 의해 홀딩된 명령)이 첫번째로 실행되기 때문에, 이 명령은 병렬로 실행되는 다른 명령의 영향을 받지 않는다. 대조하여, 제2명령(즉, 명령 레지스터(D2)에 의해 홀딩된 명령)은 제1 명령에 연속하여 실행되게 된다. 따라서 제1 명령의 종착 레지스터가 제2 명령의 소오스레지스터로 지정되는 경우가 있어서, 변경 소오스 레지스터 값이 제2 명령의 실행에 참조되지 않아, 잘못된 연산 동작을 행하게 될 수가 있으므로, 제1명령을 실행한 후에 제2 명령을 실행할 필요가 있다. 이러한 이유에서 디코더(D5)는 신호라인(d8)을 통하여 디코더(d6)로 제1명령의종착레지스터의 식별번호를 전송한다. 디코더(D6)에서 제2명령의 소오스 레지스터의 식별번호와 입력된 제1명령 식별레지스터번호가 비교되고, 두 레지스터 식별번호 사이의 일치 여부를 체크한다. 그들 사이의 비교가 일치하는 경우, 디코더(D6)는 자체의 명령 해독동작을 중지하고, 제1명령에 의해 지정된 연산동작이 연산장치(D11)(ALU1)에 의해 실행되어진 후에 제2 명령을 실행하도록 제어한다.

마찬가지로 제2프로세서소자의 병렬동작 제어기구(A10)에 의해 유사한 동작이 실행된다. 그러나 병렬화 플러그(A7)출력에 따라 병렬동작이 행해지게 되면, 하기에 기술되어지는 바와 같이 프로세서소자(A3)의 동작은 프로세서소자(A2)의 동작과는 다르게 된다. 멀티프로세서동작모드에서 각 프로세서소자(A2,A3)가 서로 독립적으로 동작하므로, 프로세서소자(A3)의 동작이 결코 프로세서소자(A2)의 동작에 영향을 미치지

지 않는다. 따라서 제3명령(즉, 명령레지스터(D3)에 의해 출당된 명령)의 실행으로 명령레지스터(D1,D2) 사이에서 레지스터 충돌이 발생하지 않는다. 제4명령(명령 레지스터(D4)에 의해 출당된 명령)을 실행하기 위하여, 제3명령의 종착 레지스터와 제4명령의 소오스레지스터 사이의 레지스터 충돌을 해소할 필요가 있다. 다른 한편, 병렬 동작모드에서 프로세서소자(A3)는 프로세서 소자(A2)에 동기되어 병렬로 명령을 실행하며, 따라서 프로세서소자(A2)에 의해 실행되는 제1 및 제2 명령의 영향을 받게 된다. 특히, 제3 명령을 실행하기 위하여, 제3 명령의 소오스레지스터와 제1 및 제2 명령의 종착 레지스터 사이에서 발생하는 레지스터 충돌을 해소해야 하고, 반면에 제4 명령을 실행하기 위해서는 제4 명령의 소오스 레지스터와 제1 내지 제3 레지스터의 종착 레지스터 사이의 레지스터 충돌을 해소해야 한다. 따라서 제3 명령의 실행으로 제1 및 제2 명령의 종착레지스터 식별번호가 신호라인(d7, d9)을 통해 디코더(D7)로 전송된다. 그런뒤 디코더(D7)에서 제3 명령의 소오스레지스터 식별번호와 입력된 제1 및 제2 명령의 종착 레지스터 번호가 비교되어, 전자와 후자 사이의 일치 여부를 체크한다. 이와 반대로, 레지스터 충돌의 발생이 검출되면, 디코더(D7)는 적어도 연산장치(D12),(ALU2)에서 실행되는 제2명령의 연산동작이 완료될 때까지 자체명령 명령의 해독동작을 중지한다. 더욱이 제4 명령을 실행하기 위하여, 제1 내지 제3 명령의 종착레지스터 식별번호가 신호라인(d7, d10)을 통해 디코더(D8)로 보내진다. 그런뒤 디코더(D8)에서 제4명령의 소오스 레지스터 식별번호와 입력된 제1 내지 제3 명령의 종착레지스터 식별번호가 비교되어 그들 사이의 일치를 체크한다. 만일 레지스터 충돌이 없으면, 연산장치(D14)에 의하여 제4명령의 연산동작이 즉시 실행된다. 이와 반대로 레지스터 충돌이 검출되면, 연산장치(D13),(ALU3)에 의해 제3 명령의 연산 동작이 완료될 때까지 자신의 명령 해독동작을 중지한다. 이 결합에서 1번째 명령의 실행이 레지스터 충돌이나 경합으로 인하여 중지하게 되면, 계속되는 명령 즉, (i+1)번째 명령의 실행을 억제하게 된다.

상기 기술에서 명백한 바와 같이, 프로세서소자(A3)내에 결합된 디코더(D7,D8)의 동작제어는 프로세서가 멀티프로세서 동작모드인지 병렬동작 모드인지에 따라 변경할 필요가 있다. 디코더 동작제어의 변경은 선택기(D9,D10)(제3도)에 의해 실현할 수 있다. 특히 멀티프로세서 동작 모드에서, 선택기(D9, D10)는 제3명령의 종착레지스터의 식별번호를 디코더(D8)에 제공하는 동안에 "0"신호(디코더(D7)에 의한 명령해독 동작을 중지할 필요가 없음)을 나타냄을 디코더(D7)에 공급하기 위하여 신호라인(a1)에 있는 병렬화 플러그(A7)의 값에 따라 제어된다. 다른 한편, 병렬 동작모드에서, 제1 및 제2 명령의 종착레지스터 식별 번호는 디코더(D8)에 보내진 제1 내지 제3명령의 종착레지스터 번호와 함께 디코더(D7)로 보내진다.

다음에 병렬동작 제어기구(A9,A10)의 룰이나 기타 기능 즉, 레지스터공유의 실현에 대하여 기술한다. 이 기능은 병렬동작 제어기구(A9,A10)에 구성되는 게이트(D17)와 신호라인(d5,d6)의 조합으로 실현된다. 멀티프로세서 동작모드에서, 각 프로세서소자는 서로 독립적으로 동작한다. 따라서 물리적으로 다른 레지스터의 값은 레지스터 식별번호 사이의 일치에 상관없이 참조된다. 특히 프로세서소자(A2)를 실행시키기 위하여 레지스터 화일(D15)의 레지스터가 사용되는 반면에, 프로세서소자(A3)를 실행시키기 위하여 레지스터화일(D16)의 레지스터가 사용된다. 이것은 각각의 판독 버스(d1,d2)와 저장버스(d3,d4)를 이용하는 판독 및 기록동작의 실행으로 성취된다. 그러나 프로세서소자가 병렬 동작모드에서 동일한 레지스터 식별번호를 지정하게 되면, 동일한 레지스터를 참조하게 된다. 이것은 상기 언급된 두개의 다른 프로세서 동작모드에 따라 기록을 위해 제공된 레지스터와 판독을 위해 제공된 레지스터를 변경할 필요가 있음을 의미한다. 이러한 문제를 해소하기 위하여, 본 발명은 레지스터 파일(D15)의 입·출력포트를 레지스터화일(D16)의 상응하는 포트의 두배로 하여 제공하고, 다른 프로세서소자와 레지스터화일(D15)내에 수용된 레지스터의 값을 전송 가능하게 하기 위해 글로벌 판독버스(d5)와 글로벌 저장버스(d6)가 제공된다. 일반적으로 프로세서소자의 수가 n개일 때, 레지스터화일(D15)의 포트수는 타 프로세서 소자내의 각 레지스터 화일의 수보다 큰 n배가 된다. 이와 같은 레지스터 화일의 구성은 프로세서소자의 수가 증가하게 되어 큰 스케일의 구조로 되므로 본 기술분야에 적합하지는 않지만, VLSI를 이용하게 되면 멀지 않아 실현 가능하게 될 것이다.

멀티프로세서 동작모드에서 각 프로세서소자는 서로 독립적으로 레지스터 화일의 참조를 행한다. 판독버스(d1,d2)를 사용하여, 두 명령의 병렬실행에 필요한 데이터를 레지스터에서 인출하고, 연산동작의 데이터 결과는 저장버스(d3,d4)를 통해 레지스터내에 기록된다. 여기에서 판독버스(d1,d2)와 저장버스(d3,d4)가 단지 내부적으로 결합된 프로세서소자에 사용되어 자기 때문에 다른 프로세서소자에 영향을 미치지 않는다.

병렬 동작모드에서 프로세서소자(A3)는 글로벌 판독버스(d5)와 글로벌 저장버스(d6)를 이용하는 레지스터 화일(D15)의 레지스터를 참조하는 반면에 레지스터 화일(D16)은 사용하지 않는다. 레지스터 공유제어의 이러한 형태는 병렬화 플러그(A7)가 신호(a1)을 사용하는 글로벌 판독버스(d5)와 글로벌 저장버스(d6)상에 제공된 게이트(D17)의 개방/폐쇄 제어에 의해 달성된다. 레지스터 화일(D15)에서 인출된 레지스터 내용중에 프로세서소자(A2)에 사용되는 이들 데이터는 판독버스(d1)를 통하여 연산장치(D11)나 연산장치(D12)로 전송된다. 더욱이 레지스터 파일(D15)내와 연산장치(D11)나 연산장치(D12)에 의해 실행되는 연산동작 결과의 기록은 저장버스(d3)를 통해 이루어진다. 이 경우에 다른 프로세서소자와의 통신은 실행되지 않는다.

프로세서소자(A2,A3)가 병렬동작 제어기구(A9,A10)를 구비하여 구성되고 프로세서가 멀티프로세서 동작모드인지 병렬 동작모드인지의 여부에 따라 명령캐시(B1)로부터 인출되는 명령의 시퀀스, 수 및 판독 폭을 변경하기 위한 캐시제어기(A6)가 제공된 것을 참조하여 설명한다.

멀티프로세서 동작모드에서 서로 독립적인 각 프로세서소자에 의해 판독명령이 수행되는 반면에, 병렬 동작모드에서 각 프로세서소자는 한번의 실행으로 n개의 명령 시퀀스를 인출한다. 특히 병렬화 플러그(A7)가 멀티프로세서 동작모드를 지시하면, n개 프로세서소자의 어드레스라인은 서로 분리되어 명령캐시(B1)의 입력에 직접 연결된다. 다른 한편 병렬화 플러그(A7)가 병렬 동작모드를 지시하면, 단지 제1프로세서소자의 어드레스라인이 제1 내지 (n-1)개의 명령길이를 갖는 어드레스값의 조건으로 부터의 결과값과 그 어드레스 라인의 어드레스값을 이용하기 위해 어드레스 입력으로 사용된다.

또, VLW의 프로세서로서 동작하는 경우에는 명령캐시로부터 1개의 명령이 독출된다. 이 1개의 명령은 복수의 연산명령 필드로부터 구성되어 있고, 프로세서(A2,A3)에서 실행되는 명령을 각각의 필드에 할당

한다.

제4도는 상기 기술한 캐시제어를 실행하는 캐시제어기의 가장 간단한 구조를 보여주는 블록도이다. 참조도면에서 각 프로세서소자의 데이터캐시(B2)와 어드레스라인(a4, a5)은 캐시(B2)의 입력에 직접 연결된다. 제1 프로세서소자(A2)의 어드레스라인의 어드레스값은 가산기(E1)에 의해 하나의 명령길이에 상응하는 워드수와 더해지고, 상기 가산의 결과값에 상응하는 신호가 어드레스라인(e1)으로 출력된다. 제4도에서 기호 E2는 선택기를 나타낸다. 병렬화 플러그(A7)가 멀티프로세서 동작모드를 지시하면 선택기(E2)는 제2프로세서소자(A3)에서 시작되는 어드레스라인(a5)을 선택하고, 반면에 병렬화 플러그(A7)가 병렬 동작모드를 지시하면, 선택기(E2)는 어드레스라인(e1)을 선택한다. 이러한 방법에서 멀티프로세서 동작모드의 경우, 각 프로세서소자가 서로 독립적으로 각각의 명령어드레스 입력을 행하는 반면에, 병렬 동작모드의 경우에는 제1 프로세서소자(A2)의 어드레스라인의 어드레스값에 따라 n개의 연속적인 명령을 지정하는 것이 가능하다. 제4도에서 보여준 구성에 추가하여, (m x k)비트의 각 출력포트와 함께(n x m x k)비트의 출력포트로 명령 캐시(B1)의 n개의 출력포트중의 하나를 선택하는 것도 고려할 수가 있다. 여기에서 k는 명령의 비트길이를 나타내고, m은 프로세서소자에 의해 동시에 인출된 명령의 수를 나타낸다.

제5도는 명령캐시와 결합되는 상기 언급한 구성의 전형적인 실현을 보여주는 블록도이다. 여기서 n과 m은 2(두개)와 동등하다고 가정한다. 이 도면에서 기호 F1은 병렬화 플러그(A7)의 출력신호(a1)에 의하여 제어되는 선택기를 나타낸다. 더욱이, f1은 명령캐시(B1)의 제1출력 포트에서 연장되는 데이터 라인을 나타내고, f2는 제2 출력포트에서 연장되는 데이터라인을 나타낸다. 이 경우에 있어서, 각 프로세서소자(A2, A3)에서 연장되는 어드레스라인(a4, a5)의 어드레스값은 명령캐시(B1)로의 어드레스 입력에 사용된다. 이것은 출력포트(f1)가 다른 출력포트(f2)의 수보다 큰 n배(이 경우는 두배)인 비트폭을 갖는다는 것을 의미한다. 이들 비트의 m x k 비트(이 경우는 2 x k 비트)는 신호라인(b1)을 통해 프로세서소자(A2)로 보내지고, 다른 비트는 신호라인(f3)를 통해 선택기(F1)에 입력된다. 멀티 프로세서 동작모드에서 (m x n)비트가 명령을 실행하기 위해 각 출력포트에서 인출된다. 그때 제1 출력포트에서 출력된 잔여((n-1) x m x k)비트는 사용되지 않는다. 병렬 동작모드의 경우에 제1출력포트(f1)에서 인출된 연속적인 명령(n x m x k 비트)은 각 프로세서소자(A2, A3)로 분배된다. 이때 잔여(n-1)출력은 사용되지 않는다. 여기에서 각각의 신호라인(f2, f3)을 통해 보내진 명령시퀀스가 프로세서소자(A3)에 의해 실행되는 선택기에 의해 결정된다. 다른 한편, 선택기(F1)는 병렬화 플러그(A7)의 출력신호(a1)에 의해 제어된다. 특히 멀티프로세서 동작모드에서 신호라인(f2)이 선택되는 반면에, 명령 동작 모드에서는 신호라인(f3)이 선택된다.

다음에 데이터 처리시스템이 멀티프로세서 동작모드에서 동작하는 본 발명에 데이터 처리시스템의 또 다른 실시예에 대하여 기술한다.

제6도는 본 발명에 따른 데이터 처리시스템의 멀티프로세서 시스템으로 동작하는데 요구되는 프로세서소자 사이의 인터페이스나 통신장치가 제공된 멀티프로세서 동작 제어기구의 기능을 설명하기 위한 차트이다. 프로세서 소자로의 인터럽션을 행하기 위한 기능으로, 식별 번호를 지정하여 프로세서 소자로 인터럽션요구를 행하는 인터럽션 기능과, 모든 프로세서소자(ALL INTERRUPT)로 인터럽션요구를 전달하는 인터럽션 기능과, 프로세서 소자중의 어느 하나가 인터럽션을 받아들임으로써 (ANYONE INTERRUPT) 인터럽션이 충분하게 되는 인터럽션 기능등의 세형태가 있다.

더욱이 제어명령의 연결에 있어서, 프로세서소자 사이에서 레지스터값을 전송하기 위한 명령과, 개시신호가 다른 프로세서소자에서 나올 때까지 관련 프로세서소자를 대기상태로 있게 하는 대기명령(WAIT)과, 관련 프로세서를 대기상태에서 해제하기 위한 재개명령(START)이 병렬처리를 정교하게 실행하도록 제공된다.

배타적 제어명령으로 주메모리의 전용어드레스에서 나오는 "비교 앤드 스왑" 명령이나 "테스트 앤드 세트" 명령이 있다. "비교 앤드 스왑" 명령에는 메모리에서 데이터를 인출하고, 인출된 데이터의 값을 체크하고, 체크결과에 따라 최초 어드레스에서 메모리내로 데이터를 기록하는 명령이 있다. 이 일련의 처리실행동안에 주메모리의 동일 어드레스에 대하여 타 프로세서소자의 "비교 앤드 스왑" 명령을 차단할 필요가 있다. 이를 위하여 "비교 앤드 스왑" 명령의 실행에 앞서서 각 프로세서소자는 멀티프로세서 동작 제어기구(A8)로 "비교 앤드 스왑"명령에 대한 실행요구를 행한다. 실행요구가 허용된 후 "비교 앤드 스왑" 명령이 실행되면, 실행요구가 소거된다. 실행요구가 복수의 프로세서소자에 의해 요구되어지는 경우, 멀티프로세서 동작 제어기구(A8)는 단지 하나의 어드레스에만 요구를 지시할것인지의 여부를 결정한다. 만일 하나의 어드레스에 실행요구를 지시하라고 결정하면, 멀티프로세서 동작 제어기구는 단지 하나의 프로세서소자에 실행권을 부여한다. 다른 한편, 복수의 요구가 타 어드레스에 직접 요구되면, 실행요구가 나오는 각 프로세서소자에 실행권이 주어진다.

제7도는 상기 기술한 "비교 앤드 스왑" 명령제어를 실행하는 멀티프로세서 동작 제어기구(A8)의 내부구조를 보여주는 개략적인 회로도이다.

제7도에서 기호 g1, g2는 각 프로세서소자에 의한 "비교 앤드 스왑" 명령의 실행에 사용되는 어드레스라인을 나타내고, 기호 g3, g4는 "비교 앤드 스왑" 명령 실행의 종료를 가르키는 신호라인을 나타내며, 기호 g7, g8는 각 프로세서소자(A2, A3)로 "비교 앤드 스왑" 명령의 실행허용이나 실행불가를 전달하는 신호라인을 나타내며, 여기에서 신호라인(g7, g8)의 "ON"상태는 "비교 앤드 스왑"명령의 실행허용을 가르키고, "OFF"상태는 실행불가를 가르킨다. 신호라인(g1~g8)은 제1도에서 보여준 신호라인(a6, a7)을 구성한다. 더욱이 멀티프로세서 동작 제어기구(A8)는 병렬화 플러그(A7)의 출력신호(a1)를 입력으로 한다.

더욱이 제7도에서 기호 G1는 현재 실행되고 있는 "비교 앤드 스왑" 명령의 어드레스를 홀딩하는 버퍼 레지스터를 나타내고, 기호 G2는 현재 프로그램실행되고 있는 "비교 앤드 스왑" 명령의 수를 조정하는 카운터를 나타내며, 기호 G3~G5는 가산 및 감산을 실행하는 연산장치를 나타내고, 기호 G6~G8는 카운터(G2)의 값과 "비교 앤드 스왑"명령의 어드레스를 비교하는 비교기를 나타내며, 기호 G9, G10는 병렬화 플러그(A7)의 값에 따라 신호라인(g7, g8)의 상태나 값을 변경하는 선택기를 나타내고, 기호 G11는

버퍼 레지스터(G1)에 입력되는 어드레스를 선택하는 선택기를 나타낸다.

다음에, 상기 기술한 하드웨어 구조의 멀티프로세서 동작 제어 기구(A8)의 전형적인 동작을 기술한다. 두 프로세서소자(A2,A3)로부터 들어오는 어드레스라인(g1,g2)의 어드레스는 각각의 비교기(G6,G7)를 통해 버퍼(G1)의 내용과 비교되어지고, 거기에서 어드레스라인(g1,g2)의 어드레스가 버퍼(G1)의 내용과 일치하는지의 여부에 따라 "ON"이나 "OFF"레벨을 신호라인(g9,g10)으로 출력한다. 더욱이 카운터(G2)의 값이 비교기(G8)에 의해 "0"(제로)값과 계속해서 비교됨으로써, 상기 비교 결과의 부정을 나타내는 신호가 신호라인(g11)으로 출력된다. 바꾸어 말하면 카운터(G2)의 값이 "0"이 아니고 신호라인(g11)이 "ON"상태라면 버퍼(G1)내에 저장된 어드레스가 존재함을 가르키고, 반면에 카운터(G2)의 값이 "0"(제로)이면, 신호라인(g11)은 "OFF"상태이다. 그때, 신호라인(g5, g9, g11)에서 신호의 논리적을 결정함으로써, 제1프로세서 소자(A2)에 의해 실행되는 "비교 앤드 스왑" 명령의 어드레스와 동일한 어드레스가 버퍼(G1)내에 존재하게 된다. 유사하게 신호라인(g6,g9,g11)에서 신호의 논리적을 결정함으로써, 제2프로세서소자(A3)에 의해 실행되는 "비교 앤드 스왑"명령의 어드레스와 동일한 어드레스가 버퍼(G1)내에 존재하게 된다. 또한 각각의 신호라인(g12,g13)으로 이들 논리적 신호의 부정을 출력하면, 상기 언급한 동일 어드레스가 버퍼(G1)내에 존재하지 않게된다. 바꾸어 말하면, 상술한 부정은 "비교 앤드 스왑" 명령이 실행되어지는 인식을 의미한다. 이것은 상술한 비교가 현재 실행되는 "비교 앤드 스왑" 명령의 어드레스 비교임을 의미한다. 따라서 타프로세서소자에서 동시에 실행되는 "비교 앤드 스왑"명령의 어떤 충돌도 해소해야 한다.

제7도에서 보여준 실시예의 경우에 있어서, 이러한 하드웨어 배열은 제1프로세서소자(A2)가 타프로세서소자 보다 항상 우선되어 부여됨을 보여준다. 따라서 제1 프로세서소자(A2)의 인식신호(g12)는 선택기(G9)를 통해 신호(g7)로 바로 보내진다. 이와 반대로 제2프로세서소자(A3)의 인식신호(g13)는 제1 프로세서소자(A2)의 인식신호(g12)와 충돌하지 않는 상태에서 선택기(G10)를 통해 신호(g8)로 보내진다. 이 두 신호(g12,g13)의 논리적 부정(g14)에 의해 이들 두 인식부호(g12,g13)의 충돌을 확실하게 방지한다. 이 경우 제2 프로세서소자의 허가인식신호(g15)는 신호(g13,g14)의 논리적으로서 구동된다. 그러면 다음 신호(g15)는 선택기(G10)로 입력된다.

선택기(G9,G10)는 병렬화 플러그(A7)가 멀티프로세서 동작모드를 지시하면 각각의 상술한 인식신호(g12,g15)를 선택한다. 다른 한편, 병렬화 플러그(A7)가 병렬 동작모드를 지시하면 이들 선택기는 병렬화 플러그(A7)의 값을 반송한다.

선택기(G11)를 통해 버퍼(G1)내에 저장하는 어드레스의 선택이 달성된다. 프로세서소자(A2 또는 A3)가 실행권을 부여함에 따라, 선택기(G11)에 의해 버퍼(G1)로 입력되는 어드레스라인이 선택된다. 실행권이 부여되면, 카운터(G2)의 값은 연산장치(G3)에 의하여 증가된다. 두 프로세서소자(A2,A3)의 인식신호(g12,g15)의 논리적(g16)에 의거하여 연산장치(G3)의 제어가 실행된다. "비교 앤드 스왑"명령의 실행이 완료되면 종료신호(g3 또는 g4)가 나오고, 이들 신호에 응답하여 연산장치(G4 또는 G5)에 의해 카운터(G2)가 감소된다. 각 비교기(G6,G7)의 출력신호(g9,g10)와 종료신호(g3,g4)의 논리적(g17,g18)에 따라 연산장치(G4,G5)의 제어가 실행되고 지정 어드레스가 버퍼(G1)내에 존재함을 나타낸다.

여기에서 버퍼(G1)의 오버플로우가 발생할 수도 있지만, 이러한 문제는 "비교 앤드 스왑" 명령의 실행이 완료되자마자 버퍼(G1)내에 있는 어드레스가 지워지기 때문에, 버퍼(G1)로서 충분한 용량을 갖는 버퍼를 사용함으로써 대처할 수가 있다.

상기에서 이해할 수 있는 바와 같이, 본 발명에 따른 멀티프로세서 동작 제어기구(A8)는 근본적으로 서로 다른 어드레스로 "비교 앤드 스왑" 명령을 실행하는 한, 복수의 "비교 앤드 스왑" 명령의 실행권을 부여할 수 있다.

제8도는 제6도에서 설명된 대기/재개명령과 레지스터 전송명령을 수행하는 멀티프로세서 동작 제어기구의 구조를 보여주는 블록도이다. 제8도에서 기호 H1,H2는 각 프로세서소자 사이의 데이터 전송, 동기 및 기타 명령을 변경하는 동기기구를 나타낸다. 기호 H3는 프로세서소자 사이의 기타 동작이나 데이터 전송을 실행하는 버스 사용권을 조정하는 버스조정회로를 나타낸다. 더욱이 기호 h1-h3는 어드레스라인, 데이터라인, 제어라인 등을 포함하는 신호라인이다. 더하여 라인(h1)은 일반적으로 통신버스로 참조된다.

이런 방법으로 멀티프로세서 동작 제어기구(A8)를 사용함으로써, 프로세서소자 사이의 협조를 프로그램을 실행하는게 가능하게 된다. 이러한 연결에서 데이터 처리시스템내에 프로세서소자의 협조명령은 제6도의 연결에서 기술된 바와 같이 프로세서소자 사이의 레지스터 내용의 전송을 지시하는 "레지스터 전송 명령"과, 어떤 어드레스를 지정하여 자체 프로세서소자의 동작을 정지하는 "대기(정지)명령"과, 지정된 어드레스에서 대기 또는 정지하고 프로세서소자를 재개하는 "재개명령"을 포함한다.

제9도는 상기 기술한 여러가지 명령의 형식예를 보여준다. 이 도면에서 보여주는 바와 같이 여기에서 두 형태의 레지스터 전송명령이 제공된다.

제9도에서 후위 "(1)"에 의해 지정된 제1 레지스터 전송명령은 두 형태의 다른 형식 즉, 송신측 프로세서소자에서 실행되는 송신명령을 위한 형식(여기에서, 송신을 위한 프로세서소자는 송신측 프로세서소자로 참조되고, 수신을 위한 프로세서소자는 수신측 프로세서소자로 참조됨)과, 수신측 프로세서소자에서 실행되는 수신명령을 위한 형식을 구비한다. 이들 두 명령형식을 조합함으로써, 송신될 값을 출력하는 레지스터의 식별신호(R1), 수신측 프로세서소자의 식별번호(PE#) 및 수신측 프로세서소자로 전송된 값을 저장하는 레지스터의 식별번호(R2)를 송신 명령 형식내에서 지정할 수 있다. 수신명령의 경우, 상기 언급한 바와 같은 그러한 지정은 필요치 않다. 명령 레지스터 전송명령(2)은 송신 및 수신 양쪽에 공유된다. 특히, 송신명령에 의하여 전송될 값을 출력하는 레지스터의 식별번호(R1)와 수신측 프로세서소자의 식별번호(PE#)가 지정된다. 유사한 방법으로 수신명령에 의하여 수신 측 프로세서소자에 전송된 값을 저장하는 레지스터의 식별번호(R1)와 송신측 프로세서소자의 식별번호(PE#)가 지정된다. 제9도에서 설명된 레지스터 전송명령의 형식구조에서 확실하게 언급된 이러한 연결에서, 1:1 레지스터 전송에 대한 프로세서소자의 최대허용수는 프로세서소자 식별번호를 지정하는 가용영역이 11비트이므로  $2^{11}$ 개까지가 된다.

1:복수 레지스터 전송을 가능하게 하기 위해서는 1비트와 하나의 프로세서소자 사이에 대응관계를 설정하는 것이 요구된다. 이 경우 데이터 처리시스템은 최대 11개의 프로세서소자를 내장할 수가 있다.

레지스터 전송의 연결에서 두 형태의 동기방식 즉, 수신측 프로세서 소자가 수신된 명령을 실행하는지의 여부에 관계없이 송신측 프로세서 소자가 직접적으로 계속해서 명령을 실행하기 위한 레지스터 내용을 보내는 동기방식(a)와, 수신측 프로세서소자가 수신된 명령을 실행할 때까지 송신측 프로세서소자가 일시정지 또는 정지상태로 세트되는 동기방식(b)가 고려된다. 따라서 상기 언급된 두 형태의 동기방식을 갖는 두 종류의 형식을 결합하여 네가지 형태의 하드웨어를 수행할 수가 있게 된다. 그러나 상기 기술을 보다 편리하고 간단하게 하기 위하여 여기에서 설명되는 데이터 처리시스템에 결합되는 두 형태의 하드웨어 즉, 명령형식(1)과 동기방식(a)을 조합하는 것과, 명령형식(2)과 동기방식(b)을 조합하는 것으로 기술한다. 다른 두 형태의 조합에 대한 하드웨어 구조는 이 기술분야에서 숙련된 사람들에 의해 하기에 기술되어지는 이 두 형태의 하드웨어를 변경함으로써 쉽게 가능하게 될 것이다.

제9도를 참조하여 각 정지(대기)명령과 재개(개시)명령에 대한 형식이 부가적으로 설명된다. 정지명령과 재개명령에서 지정한 어드레스는 레지스터(R1), 변위(d) 및 베이스레지스터(Rb)에 따라 연산된다. 종래의 컴퓨터시스템의 경우, 통상적으로 (Rb+R1+d)에 따라 어드레스 연산이 실행되었다. 그러나 본 발명에 따른 데이터 처리시스템에 있어서는 작동이 정지되는 정지어드레스가 상기 열거된 값으로부터 명확하게 결정될 수가 있다는 것이 충분히 보장된다.

제10도 내지 제14도는 레지스터 전송명령을 실행하기 위한 하드웨어 구조를 보여주는 블록도이다. 보다 구체적으로, 제10도 및 11도는 레지스터 전송명령형식(1)과 동기방식(a)의 조합에 따른 하드웨어 구조를 보여주고, 제12도 및 13도는 레지스터 전송명령형식(2)과 동기방식(b)의 조합에 대한 하드웨어 구조를 보여주며, 제14도는 상기 언급된 양쪽의 하드웨어 구조에 공유되는 우선순위(시이퀀스)결정회로를 보여준다. 다음에 이 도면번호에 대하여 기술한다.

제10도는 명령형식(1)과 동기방식(a)가 서로 조합된 경우의 동기기구(H1)의 내부구조를 보여준다. 동기기구(H2)도 유사한 구조로 실시된다. 동기방식(a)에 따라 송신명령이 실행된 프로세서소자는 수신측 프로세서소자의 수신명령을 기다리지 않는다. 이러한 상황에서 전송된 레지스터값과 전송된 레지스터값을 저장할 레지스터의 식별번호를 일시적으로 홀딩하기 위한 각 프로세서소자를 위한 버퍼레지스터화일을 제공할 필요가 있다. 이를 위하여 레지스터값과 레지스터 값이 저장되는 종착레지스터의 식별번호를 홀딩하는 동기 레지스터 화일(11)이 제공된다. 제10도에서 기호 12, 13는 각각 동기 레지스터화일의 어드레스를 지정하는 어드레스 레지스터를 나타내고, 이들은 수신측 프로세서소자가 수신된 값을 인출할 때 송신측 프로세서소자가 송신값(즉, 송신된 값)을 판독할 때 각각 사용된다. 기호 14는 동기 레지스터 화일(11)에서 판독/기록동작이 실행된 후에 어드레스 레지스터(12, 13)의 값을 증가시키는 가산기를 나타낸다. 동기레지스터화일(11)이 FIFO(선입선출)형식으로 판독/기록동작을 실행하도록 설계되어 있기 때문에, 두 어드레스 레지스터의 판독/기록 동작이 행해진 후에 어드레스를 증가시키는 가산기(14)가 제공된다. 기호 15는 기록/판독동작에 따라 어드레스 레지스터(12)나 어드레스 레지스터(13)로부터 동기레지스터 화일(11)로 입력되는 어드레스를 선택하는 선택기를 나타낸다. 기호 16 및 17은 각각의 비교회로이다. 동기 레지스터화일(11)내에 저장된 데이터가 없는 경우, 프로세서소자는 수신명령을 정지하라는 동작을 실행한다. 다른 한편, 최대용량 한도까지 동기레지스터 화일내에 데이터가 저장되는 경우, 프로세서소자는 송신명령을 정지하라는 동작을 실행한다. 이러한 실행에 있어서, 판독 어드레스 레지스터(12)의 값을 RA로 표시하고 반면에 기록 어드레스 레지스터(13)의 값을 WA로 표시한 다음 "WA = RA"를 나타내는 신호를 발생함으로써 동기레지스터 화일(11)내에 어떤 데이터도 저장되지 않은 것을 나타낼 수 있다. 더욱이 "WA+1=RA"를 나타내는 신호를 발생하여 최대용량 한도까지 동기레지스터 화일(11)내에 데이터가 저장되는 것을 알 수 있게 된다.

기호 i1~i8은 각각의 제어라인을 나타낸다. 더욱 상세하게 i1~i8은 프로세서소자(A2)로 클럭신호를 입력하는 클럭신호라인이고, 어드레스 레지스터(12, 13)는 이 입력클럭의 타이밍에 따라 동작한다. 기호 i2는 프로세서소자(A2)에 의한 수신명령의 실행이 "ON"레벨일 때의 판독신호 라인이고, 기호 i3은 타프로세서소자가 조합된 프로세서소자에 대해 송신명령을 실행할 때의 기록신호라인이다.

동기레지스터 화일(11)로 기록요구와 판독요구가 동시에 발생하면, 이들 두 요구중 하나는 대기상태가 된다. 제10도에서 보여준 하드웨어 구성의 경우에는 판독요구가 대기상태이다. 판독신호라인(i2)과 기록신호라인(i3)을 논리적으로 앤딩(ANDing)하여 형성한 판독금지신호(i4)를 프로세서소자(A2)에 요구한다. 판독금지신호(i4)가 "ON"레벨이면, 프로세서소자(A2)는 명령의 실행을 일시적으로 중지한다. 기호 i5 및 i6는 비교회로(i6, i7) 각각의 출력신호 라인이고, 여기에서 신호라인(i5)의 신호는 동기레지스터 화일(11)내에 저장된 데이터가 없음을 가르키고, 반면에 신호라인(i6)의 신호는 최대용량 한도까지 동기레지스터 화일(11)내에 데이터가 저장됨을 가르킨다. 또한 이 신호라인(i5)은 프로세서소자(A2)에 연결된다. 신호라인(i5)이 "ON"레벨일 때 프로세서소자(A2)에 의한 수신명령의 실행은 일시적으로 정지된다.

실제적으로, 기록요구가 없고 동기 레지스터 화일(11)내에 저장된 데이터가 존재하는 상황에서 동기 레지스터 화일(11)에서 판독된 데이터가 수신명령을 실행한다. 따라서 신호라인(i2, i4' 및 i5')상의 신호들의 논리적이 신호라인(i7)으로 출력되어, 판독 어드레스 레지스터(12)에 대한 제어신호로 사용된다(X'는 X의 부정을 나타냄). 신호(i7)는 동기레지스터 화일(11)에서 인출된 데이터를 받은 후에 어드레스 레지스터(12)를 증가시킨다. 이와 유사한 방법으로 동기레지스터 화일(11)로의 데이터 판독은 데이터가 최대용량 한도까지 동기 레지스터 화일내에 저장되지 않는다는 조건하에서 실행된다. 따라서 신호라인(i3, i6)상의 신호의 논리적인 신호라인(i8)으로 나와, 기록어드레스 레지스터(13)의 제어신호로 사용된다. 신호(i8)는 동기 레지스터화일(11)내에 데이터를 기록한 후 기록어드레스 레지스터(13)를 증가시킨다. 따라서 이 신호(i8)는 선택기(i5)의 제어와 동기 레지스터의 판독/기록모드의 제어에 사용된다.

기호 i9 및 i10는 종착 레지스터의 식별번호와 레지스터 값의 송수신에 사용된다. 더욱이 기호 i11는 통신버스(hi)의 사용권을 요구하는 신호라인과 수신측 프로세서소자의 식별신호를 지정하는데 사용되는 신호

호라인을 포함하는 여러 제어라인이다.

제11도는 버스조정회로(H3)의 내부구조를 보여주는 블록도이다. 도면에서 기호 J1은 기본클럭(A11)과 동일한 클럭신호를 발생하는 오실레이터이고, 기호 J2는 우선순위(시이퀀스)결정회로이다. 복수의 프로세서소자 사이의 통신버스(h1)에서 충돌이나 결합이 발생하게 되면, 우선순위 결정회로(J2)는 고정되거나 가변할 수 있는 우선순위에 따라 프로세서소자중의 하나에 통신버스(h1)의 사용권을 부여한다. 기호 J3 및 J4는 통신버스(h1)의 사용권을 지시하는 신호를 저장하는 플립플롭이다. 이들 플립플롭은 통신버스의 사용권이 관련 프로세서에 주어질때 "ON"상태로 각각 세트된다. 기호 J5는 송신명령에 의해 지정된 프로세서소자의 식별번호에 따라 출력신호 라인중의 하나인 입력신호를 송출하는 기능을 갖는 멀티플렉서이다.

기호 j1~j4는 통신버스(h1)를 구성하는 각 신호라인이다. 특히 j1은 본 발명의 전체 데이터 처리시스템의 모든 제어에 사용되는 클럭신호를 전달하는 클럭신호라인이다. 이 신호라인(i1)은 제10도에서 보여준 신호라인(i1)에 상응한다. 기호 j2 및 j3는 각각 버스요구신호 라인을 나타낸다. 하나의 프로세서소자가 레지스터 전송을 위해 통신버스(h1)를 사용하기 위해서 먼저 상응하는 신호라인에서 버스요구신호 j2 또는 j3를 출력한다. 기호 j4 와 j5는 버스 허용신호라인이다. 통신버스(h1)의 사용권이 출력되는 버스요구신호 j2 또는 j3에 대응하는 하나의 프로세서소자에 주어지면, 허용신호 j4 또는 j5는 상술한 프로세서소자로 보내진다. 버스요구신호가 복수의 프로세서소자에서 동시에 출력되면, 통신버스(h1)의 사용권은 고정되거나 가변할 수 있는 우선 순위에 따라 신호라인 j4 또는 j5를 통해 부여된다. 기호 j6 와 j7은 기록금지 신호라인이다. 제10도를 참조하여 앞에서 기술된 바와같이, 수신측 프로세서소자의 동기 레지스터 화일(i1)이 프로세서소자에 의한 송신명령의 실행으로 데이터가 최대용량 한도까지 로드되면, 송신명령을 일시적으로 정지해야 한다. 이 경우에 기록금지 신호는 신호라인 j6 또는 j7로 인출된다. 기호 j8 및 j9는 데이터가 최대용량 한도까지 동기 레지스터 화일(i1)내에 저장될 때 "풀"신호를 나타내는 신호 라인이다. 기호 j10 및 j11는 송신명령이 실행되는 것을 나타내는 신호를 프로세서소자에 보내는 기록신호라인이다. 기호 j12는 프로세서소자 식별번호의 각 신호를 전달하는 신호라인이다. 이 신호라인은 송신명령의 실행시 수신측 프로세서소자를 지정하는데 사용된다. 신호라인(j2, j4, j6 및 j12)은 제10도에서 보여준 신호라인(i11)에 상응한다. 기호 j13은 데이터 라인이고, 기호 j14는 레지스터 식별번호(REG)의 각 신호를 전달하는 신호라인이다. 이들 신호라인들은 버스요구신호(BR)(j2, j3)와 버스허용신호(BG)(j4, j5)와 함께 버스조정회로(H3)에 의해 간단하게 관리된다. 라인(j13, j14)의 신호는 버스조정회로(H3)에서 직접 사용되지 않는다.

다음에 버스조정회로(H3)의 내부에서 신호(j6, j7 및 j10, j11)를 발생하는 방법에 대하여 기술한다. 버스허용회로(j4, j5)를 발생하는 방법은 제14도를 참조하여 후술한다. 기록요구신호 j10 또는 j11은 송신명령의 실행시에 지정된 프로세서소자로 보내진다. 이 경우 프로세서소자가 식별번호 신호라인(j12)의 신호로 지시되는 반면, 프로세서소자중의 하나에 의한 송신명령의 실행은 모든 버스허용신호의 논리합신호(j15)에 의해 지시된다. 따라서 신호라인(j15)의 신호에 의해 입력되고 선택신호로서 프로세서소자 식별신호에 의해 공급된 멀티플렉서(J5)의 출력신호는 기록요구신호(J10, J11)로 바로 사용해도 된다. 수신측 프로세서소자의 동기레지스터 파일(i1)이 최대용량 한도(즉, "풀")까지 데이터를 저장하면 기록금지신호(j6, j7)는 송신명령의 실행시에 인출된다. 데이터는 최대용량 한도까지 동기레지스터 파일(i1)내에 저장되므로 신호라인(j4, j5 또는 j10, j11)의 신호에 의해 지시되므로 최대용량한도까지 동기레지스터 파일(i1)내에 저장된다(라인(j4, j5)의 신호가 송신측에서 검출되는 반면에 라인(j10, j11)의 신호는 수신측에서 검출된다). 따라서, 라인(j8, j10)상의 신호의 논리합신호(j16)나 라인(j9, j11)상의 신호의 논리합신호(j17)는 수신측 프로세서소자의 기록금지 신호를 나타낸다.

모든 신호(j16, j17)의 논리합신호(j18)를 발생하고 신호(j4, j5)와 논리합신호(j18)의 논리합신호를 발생함으로써, 송신측 프로세서소자의 기록금지신호 j6 또는 j7을 얻을 수가 있다.

다음에, 제9도에서 설명된 동기방식(b)과 레지스터 전송명령 형식(2)의 조합을 실현하기 위한 하드웨어 구조에 대하여 기술한다. 제12도는 명령형식(2)과 동기방식(b)의 조합에 상응하는 동기기구(H1)를 보여주는 블록도이다. 제12도에서 보여주는 바와 같이, 동기기구(H1)는 단지 신호라인 만으로 구성된다. 멀티프로세서 동작제어기구(A8)는 버스조정회로(H3)에 의해 달성되는 집중관리나 제어에 의해 실현된다.

제12도에서 기호(m1)는 제10도에서 보여준 신호라인(i1)과 동일한 기능을 행하는 클럭신호라인이다. 더욱이 m2와 m3는 각각 판독신호라인과 기록신호 라인이다. 레지스터 전송명령의 실행시에 수신명령이 실행되면 하나의 신호는 판독신호라인(m2)으로 인출되고, 반면에 하나의 신호는 기록명령의 실행을 위해 기록신호라인(m3)으로 인출된다. 기호(m4)는 라인으로서 뿐만 아니라 전송금지신호로도 사용된다. 레지스터 전송에 대한 송신명령과 수신명령이 상응하지 않게 되면, 전송금지신호는 프로세서소자(A2)의 동작을 일시적으로 정지하거나 멈추게 한다. 더욱이 기호(m5)는 데이터 라인이고, 기호(m6)는 여러 가지 제어라인이다.

제13도는 버스조정회로(H3)의 내부구조를 보여주는 블록도이다. 이 도면에서 기호 J1~J4와 j1~j5는 제11도에서 도시되고 동일 기호로서 표시된 것과 같은 회로와 신호라인이다. 더욱이 기호 N1, N2는 수신명령의 실행을 위해 송신측 프로세서소자에 판독신호를 전달하는 각각의 멀티플렉서이다. 기호 N3, N4는 송신명령의 실행을 위해 송신측 프로세서소자에 기록신호를 전달하는 각각의 멀티플렉서이다.

각 프로세서소자로부터의 판독신호가 선택기(N5, N6)로 입력되면, 선택기는 식별번호에 의해 지정된 수신측 프로세서소자의 판독신호라인을 선택한다. 이 신호라인은 관련 프로세서소자에 의해 실행되는 수신명령을 나타내는 신호를 전달한다. 송신명령의 실행시에 선택기 N5 또는 N6의 출력이 "ON"이면, 그 레지스터 내용은 전송된다. 기호 N7 및 N8은 수신명령의 실행시에 사용되는 선택기이다. 각 프로세서소자로부터의 기록신호는 선택기 N7 또는 N8로 입력된 다음, 수신측 프로세서소자의 식별번호에 의해 저장된 송신측 프로세서소자의 기록신호라인을 선택한다. 수신명령의 실행시에 선택기 N7 또는 N8의 출력이 "ON"이면, 송신명령의 경우에서와 같이 레지스터 내용이 전송된다.

기호 n1, n2는 기록신호라인을 나타내고, n3, n4는 판독신호라인을 나타낸다. 신호라인 n1 또는 n2는 수신

명령이 실행될 때, "ON"레벨로 된다. 유사하게, 신호선 n3 또는 n4는 기록명령이 실행될 때 "ON"레벨로 된다. 신호라인 n1 또는 n4는 기록명령이 실행될 때 "ON"레벨로 된다. 신호라인 n1 과 n3는 각각 제12도에 보인 신호라인 m2와 m3에 부합된다. 기호 n5와 n6는 전송금지 신호라인을 나타낸다. 프로세서의 소자 중 하나가 송신명령 또는 수신명령으로 실행될 때, 상대 프로세서소자가 수신명령도 실행하지 않고 송신명령도 실행하지 않게 되면 신호 n5 또는 n6의 전송이 발생될 수 있다. 전송 금지신호 n5 또는 n6는 프로세서 소자로 실행을 중지하거나 일시 정지시키기 위해 전송된다. 신호라인 n5는 제12도에 보인 신호라인 m4에 부합된다. 기호 n7과 n8는 프로세서 소자 식별신호를 지시하기 위해 각각 전송된 신호라인을 나타낸다. 특히, 신호라인 N7 또는 N8는 레지스터 전송명령의 실행의 결과로서 나타나는 프로세서 소자의 식별번호를 나타낸 신호를 전달한다. 제13도에 보인 신호라인 j2, j4와 n7은 제12도에 보인 신호라인 m6와 상응한다. 마지막으로, n9은 제12도에 보인 라인 m5에 상응하는 데이터라인을 나타낸다.

프로세서소자의 하나가 송신명령 또는 수신명령으로 실행되고 상대 프로세서 소자가 수신명령도 실행하지 않고 송신명령도 실행하지 않게 되면 버스조정회로(H3)에 발생하는 신호 n5 또는 n6가 인출된다. 상대 프로세서소자가 수신명령을 실행할 때의 정보는 선택기 N5 또는 N6의 출력신호 n10 또는 n11로 나타나고, 송신명령의 정보가 실행되는 동안 선택기 N5 또는 N6의 출력신호 n10 또는 n11로써 얻어진다. 따라서, 송신명령의 실행으로 나타나고, 상대 프로세서 소자가 대응 수신명령으로 실행되지 않도록 나타나는 정보는 신호 n3와 n10'의 논리적 신호 n14의 형태 또는 신호 n4와 n11'의 논리적신호 n15의 형태에서 유효하게 만들 수 있다. 유사하게 수신명령의 실행으로 나타나고, 일측 프로세서 소자가 송신명령으로 실행되지 않도록 나타내는 정보는 신호 n1과 n12'의 논리적 신호 n16의 형태 또는 신호 n2와 n13'의 논리적 신호 n17의 형태에서 유효하게 만들어진다. 다른 한편으로, 신호라인 n14와 n16에서 신호의 논리합은 전송금지신호 n5을 구성한다. 유사하게, 신호 n15와 n17의 논리합은 전송 금지신호 n6로써 전달된다.

제14도는 제11도와 제13도에서 보인 하드웨어 구조에 사용된 우선순위(사이퀀스)결정회로(J2)의 내부구조를 보이는 블록도이다. 제14도에 있어서, 각 프로세서 소자의 클럭신호(j1)로부터 우선순위 또는 레벨 결정회로(J2)의 버스요구신호(BR) j2와 j3이 입력되어 버스허용신호 j4와 j5가 출력된다. 도면에서, R1과 R2는 각각 플립-플롭을 나타낸다. 플립 플롭(R1)은 선행주기에서 버스허용 신호가 프로세서 소자로 나오는 프로세서 소자를 저장하기 위해 차례로 쓰이는 플립 플롭(R2)에 입력을 지연시키기 위해 쓰인다.

지금까지 각 프로세서 소자(PE1, PE2)의 버스요구신호(BR) j2와 j3사이에서 충돌 또는 경합 발생이 없었고, 버스요구신호 j2 또는 j3는 플립 플롭 J3 또는 J4에 직접 입력되고, 그로써 버스허용신호 j4 또는 j5는 버스요구신호를 내보내는 프로세서 소자에 공급할 수 있다. 그러나, 충돌 또는 경합이 버스요구신호 j2와 j3사이에서 존재할 때(즉, 라인 j2와 j3에서 신호의 논리적 "ON"되는 경우), 각 프로세서 소자의 이전 레벨은 그것에 의해 버스허용신호가 높은 (더욱 높은)이전 주기를 가진 프로세서소자에 보내는 것을 허용되도록 산출된다. 이 결합에서 프로세서 소자사이에서 고정된 우선순위를 미리 실시하는 것이 바람직하다. 그러나, 본 발명의 실시예 설명의 경우에서 제14도에 보인 회로에서 결정된 우선순위를 버스허용신호가 선행즉시 나오는 프로세서소자가 낮은 우선순위 레벨과 함께 지금 주어져지도록 수행된다.

각 프로세서소자용 버스허용신호의 논리합의 역전(전도)을 나타내는 신호(r2)는 플립 플롭(R2)의 클럭 입력단자에 입력된다. 따라서, 플립 플롭(R2)의 내용은 버스허용신호의 어느 하나가 그 레벨의 "ON"에서 "OFF"로 바뀔 때 재기록된다(전달버스 사용권의 포기를 나타냄). 플립 플롭(R1)은 제2 프로세서 소자(A3)를 위한 버스허용신호(j5)로서 입력된다. 플립 플롭(R1)은 그것에 의해 제2 프로세서소자(A3)를 위한 버스허용신호(j5)를 지연하도록 클럭신호(j1)와 동기된다. 신호라인(r3)에 나타나는 플립 플롭(R1)의 출력신호는 플립 플롭(R2)으로의 입력신호를 구성한다. 이 방법에서, 프로세서의 어느 하나라도 전송 버스의 사용권을 포기할 때, 플립 플롭(R2)은 버스 허용신호가 제2프로세서 소자(A3)에 주어지지 주어져지지 않든지 간에 기록될 수 있다. 플립 플롭(R2)의 출력신호라인은 (r4)로 나타났다. 버스요구신호 j2 및 j3사이에서 충돌이 발생할 때, 신호(r4)가 "ON"(선행 버스허용신호가 제2 프로세서소자(A3)에 주어졌을 때 즉시 나타남)으로 된다면, 버스허용신호(j4)는 제1 프로세서 소자(A2)로 주어진다. 그와 반대로, 신호(r4)가 "OFF"되었을 때, 버스허용신호(j5)는 제2 프로세서 소자(A3)에 나타난다. 그러나, 버스허용신호가 이미 프로세서 소자의 어느 하나에 나타났던 상태에서, 이 상태는 위와 같이 출당된다.

하드웨어에서 전송한 논리를 실현하기 위해 플립 플롭(J4)으로의 입력신호가( $r1 \cdot j3 + r1 \cdot (r4 \cdot r2 + j5)$ )로 주어진 상태를 만족하는 동안 플립 플롭(J3)으로의 입력신호는 ( $r1 \cdot j2 + r1 \cdot (r4 \cdot r2 + j4)$ )로 주어진 상태를 만족해야 한다. 이 경우의 결합에서 보통의 상징에서 표시 "X·Y"는 "X"와 "Y"의 논리적이고, "X+Y"는 "X"와 "Y"의 논리합을 나타낸다. 위에 언급된 논리표시에서 각 항의 의미에 관련하여 " $r1 \cdot r4 \cdot r2$ " 또는 " $r1 \cdot r4 \cdot r2$ "가 버스요구신호 사이에서 충돌 발생을 나타내는 동안(r2가 버스허용신호가 프로세서 소자의 어느것에도 아직 주어지지 않는 곳에서), 버스요구신호 사이에서 충돌을 발생하지 않는 항 " $r1 \cdot r2$ "와 " $r1 \cdot j3$ "가 추가된다. 마침내, 항 " $r1 \cdot r4$ "와 " $r1 \cdot r5$ "는 버스허용신호가 이미 프로세서 소자의 어느하나에 주어졌을 때, 허용신호가 출당됨을 유지하는 것을 나타낸다.

다음에 제9도에 설명된 대기명령과 재개 명령의 하드웨어의 현실화에 대하여 기술된다.

제15도는 재기(WAIT)명령이 재개명령을 실행하기 위한 동기기구(H1)을 보인다. 이 그림에서, 기호 T1는 프로세서 소자(A2)에 의해 대기명령의 실행에 나타난 어드레스에 저장하기 위해 적용된 대기 어드레스 레지스터를 나타낸다. 기호 T2는 배열하여 전송버스(h1)로부터의 어드레스 입력과 다른 프로세서 소자에 의한 재개명령의 실행 중의 대기 어드레스 레지스터의 내용을 비교하는 비교회로를 나타낸다.

기호 t1은 상기에 언급된 신호라인 i1 및 m1과 같은 작용을 위해 전달되는 클럭신호 라인을 나타낸다. 기호 t2는 그 자신의 라인과 마찬가지로 재개신호를 나타낸다. 재개신호는 재개명령의 실행시에 어드레스 신호(t3)와 동시에 출력된다. 어드레스 신호(t3)는 신호라인과 마찬가지로 재개명령이 실행되고, 대기명령이 실행되는 시간에 어드레스 지정을 위해 활동한다. 재개명령의 실행중에 어드레스신호(t3)는 전송버스로 출력된다. 다른 한편으로 대기명령의 실행중에 어드레스신호(t3)는 대기 어드레스 레지스터에 저장되어지는 어드레스와 상응한다. 신호라인(t6)상의 슬립신호(sleep signal)(t6)는 프로세서소자(A2)

로서 대기 명령의 실행중의 어드레스신호(t3)와 동시에 출력되어 대기 어드레스 레지스터(T1)에서 지시된 어드레스 저장을 위해 쓰인다. 기호 t4 및 t5는 각각 대기 어드레스 레지스터(T1)의 출력을 전달하고, 각각 전송버스(h1)을 통해 타프로세서 소자로부터 어드레스가 들어오는 어드레스 라인을 나타낸다. 이러한 어드레스 신호는 위에 언급된 두 어드레스 사이에 부합되는 것을 나타내는 신호가 출력되는 출력 신호라인(t8)을 가지는 비교회로(T2)에 입력된다. 신호(t8)가 "ON"될 때와 타프로세서 소자가 재개신호(t7)를 내보낼 때, 대기되었던 작동의 시작과 프로세서 소자의 회복을 위한 신호(t9)가 프로세서 소자(A2)(PE1)로 보내진다. 기호 t10는 전송버스조정을 위해 쓰이는 제어신호 라인을 포함하는 총체적인 각 제어라인을 나타낸다.

대기명령과 재개명령의 현실화를 위한 하드웨어에서, 버스조정회로(H3)의 기능, 즉 역할은 오직 어드레스라인과 제어라인과, 전송버스(h1)에 포함된 다른 것들의 사용권을 조정하고 제어하는데 있다. 따라서, 버스조정회로(H3)는 오직 제11도와 제13도에서 보이는 부품들의 오실레이터(J1)와, 우선순위 결정회로(J2)와, 플립 플롭(J3 및 J4)과, 버스허용신호라인(j4) 및 (j5)를 포함하는 이러한 내부 구조에서 수행될 수 있다. 다시 말해서, 하드웨어가 제10도 또는 제12도와 제15도에 보이는 구조의 결합에 부합하는 동시기구의 사용과 제11도 또는 제13도에 보이는 버스조정회로(H3)의 사용으로 레지스터 전송명령과, 대기명령과, 재기명령의 조정된 명령의 세가지 타입 모두의 실행이 가능하도록 현실화 될 수 있다. 더욱이 이 하드웨어와 제7도에 보인 것과의 결합으로, "비교 앤드 스왑"명령의 제어가 마찬가지로 가능하다.

제16도에서 제21도는 단지 예로서 본 발명의 실시예에 따른 데이터처리시스템의 동작을 설명하는 타이밍 차트이다. 더욱 구체적으로, 제16도에서 제20도는 명령형식(1)과 동기기구(a)가 결합되는 경우의 작동의 예를 설명하고, 한편 제21도는 대기 명령과, 재개명령의 실행에 포함된 전형적인 동작을 설명한다. 다른 한편으로, 명령형식(2)과 동기기구(b)가 결합하는 경우에 우선순위 결정회로(J2)와 플립 플롭 J3 및 J4 이외의 다른 명령회로(저장회로)는 존재하지 않는다.

따라서 그 후의 동작설명은 생략한다.

제16도에서 제20도의 타이밍 차트를 참고하여, 명령형식(1)과 동기기구(a)가 서로 결합된 경우에 데이터 처리시스템의 전형적인 동작을 설명한다.

제16도는 제1프로세서소자(PE1)에 의해 제2프로세서소자(또한 PE2로 칭함)로 송신된 명령의 수행에 포함된 작동을 설명하는 타이밍 차트이다. 먼저, 송신측 프로세서소자는 버스요구신호(BR)을 송신한다.

제1 또는 제2프로세서 소자 모두 통신버스를 이용하지 않을 때, 버스허용신호(BG)는 클럭신호(클럭)와 동기된다. 통신버스가 이용되거나 버스에 대한 충돌이나 경합이 버스요청신호 사이에 발생하면, 버스허용신호가 즉시 나온다. 버스허용신호 수신후, 수신측 프로세서소자는 송신된 명령에서 지정된 송신측 프로세서소자 식별번호, 지정레지스터의 내용이나 값(데이터) 그리고 수신측 레지스터 식별번호(REG)를 송신한다. 제16도에서 설명된 실시예의 경우에, 이런 데이터가 송신되는 시간간격은 1.5클럭 주기로서 도시된다. 그러나 이런 시간간격의 길이는 프로세서소자와 다른 것의 배치와 같은 여러 인자를 고려하여 변화될 수 있는 것으로 이해될 수 있다. 송신된 명령에 의한 프로세서 식별번호 표시를 통해, 기록신호(기록)가 수신측 프로세서로 송신된다. 데이터가 최대 용량한도까지 수신측 프로세서 소자의 동기 레지스터 파일에 저장되지 않으면, 지정된 레지스터의 값과 송신측 레지스터 식별번호가 동기 레지스터 파일에 기록되고, 여기에서 기록어드레스 레지스터(WA)가 증가된다. 데이터전송이 완성되면, 버스요구신호가 "오프"로 변함으로써 버스허용신호는 "오프"로 변하게 한다.

제17도는 또한 송신된 명령의 수행에 포함되는 작동을 설명하는 타이밍차트이다. 그러나, 이 경우에, 송신된 명령의 실행시, 수신측 프로세서소자의 동기 레지스터 파일은 이미 최대 용량 한도까지 데이터가 로드된다. 수신측 프로세서소자의 신호 "기록"과 "WA"와 송신측 프로세서소자의 신호 "BR", "BG", "PE", "데이터" 그리고 "REG"는 제16도에서 도시된 것과 같고, 같은 표시로 나타낸다. 수신측 프로세서소자의 동기레지스터화일이 상기에서 지정되듯이 최대 용량한도까지 데이터가 저장되고 "온"인 "풀"신호에 의해 지정되기 때문에, 기록금지신호 "금지"가 두 신호 "기록"과 "풀"의 논리적으로 송신된다. 이것은 수신측 프로세서소자가 통신버스의 이용권을 보유하는 동안 일시적으로 정지상태로 세트되는 결과를 가져온다. 수신측 프로세서소자가 "오프"로 변환되는 신호 "풀"의 수신명령에 응답하여 동기 레지스터 파일의 내용을 판독할 때, 수신측 프로세서 소자에 대한 기록금지신호는 "오프"가 되고, 수신측 프로세서 소자는 일시적으로 보유된 상태에서 풀려난다. 그래서, 지정 레지스터의 값과 송신측 레지스터 식별번호는 송신측 프로세서소자의 동기 레지스터화일에 저장된다. 이 경우에, 수신측 프로세서소자의 데이터 신호 "PE", "데이터" 그리고 "REG"가 출력되는 존속기간은 기록금지신호가 "온"되는 동안 각각 연장된다.

제18도는 송신된 명령의 실행에 포함되는 작동을 설명하는 다른 타이밍차트이다. 특히 이 타이밍차트는 송신측 프로세서소자의 동기레지스터 파일이 최대 용량한도까지 데이터가 로드된 경우에 송신 명령의 실행에서 통신버스의 사용권 표기에 관련된다. 이 경우에, 송신측 프로세서 소자는 기록금지 신호가 송신측 프로세서 소자에 송신될 때 일시적으로 정지된 상태로 세트되고, 동시에 버스요구 버스는 "온"에서 "오프"레벨로 변환된다. 이 결과로서, 버스허용신호는 또한 "오프"로 변환되고, 이는 통신버스 이용권이 포기되는 것을 나타낸다. 결과적으로 신호 "풀"이 수신측 프로세서소자에서 "온"에서 "오프"로 변할 때, 송신측 프로세서소자에 정지가 발생하여 일시적으로 정지된 상태에서 풀려진다. 그래서 제18도에서 설명된 계획은 통신버스에 대한 충돌이나 경합을 경감하기 위해 적합하다고 말할 수 있다.

제19도는 수신명령의 실행에 포함되는 작동을 설명하는 타이밍차트를 보여준다. 수신측 프로세서소자(PE1)는 거기에 전송된 레지스터 값 "데이터"와 데이터 저장목적 레지스터의 식별번호 "REG"를 판독하기 위해 판독신호 "판독"을 출력한다. 이 시점에서, 판독 어드레스 레지스터의 내용 "RA"는 증가된다. 판독신호가 출력되는 시간간격은 또한 수신명령을 실행하는 경우와 같이 1.5클럭 주기라고 가정하더라도, 시간간격의 길이는 프로세서소자와 다른 것들의 물리적 위치와 같은 여러인자에 따라서 변화되는 것이 가능하다.

제20도는 수신명령을 실행하는 시간에서 동기 레지스터화일에 데이터가 저장되지 않을 때 일어나는 동작을 설명하는 타이밍 차트이다. 신호 "판독", "데이터", "REG" 그리고 "RA"의 내용은 제19도에서 도시되는 것과 동일하다. 제20도를 참조하여, 수신측 프로세서소자가 수신명령을 실행하고 판독신호를 출력할 때 동기레지스터 파일에 데이터가 저장되지 않는 경우에, 신호 "공백"은 "온"레벨로 가정한다. 수신명령을 실행하는 프로세서소자는 데이터가 동기레지스터 파일에 저장되지 않는 "온" 레벨의 신호 "공백"에 근거하여 결정하고 일시적으로 정지된 상태로 가정한다. 이 시점에서, 다른 것(신호 "기록"의 "온"레벨에 의해 표시된)에 의해 프로세서 소자(PE1)에 송신명령의 실행은 신호 "공백"을 "온"에서 "오프"레벨로의 변화를 가져온다. 이 결과로, 수신측 프로세서 소자는 일시적으로 정지된 상태에서 풀러지고 제19도를 참조하여 종전에 설명하였듯이, 판독 어드레스 레지스터의 내용이 증가하는 동안 동기 레지스터 파일로부터의 데이터 저장 목적 레지스터의 식별과 거기에 전송된 레지스터 값을 판독한다.

제21도는 일시정지명령과 재개명령의 실행을 설명하는 타임차트이다. 1프로세서소자(PE1)은 일시정지명령을 수행하고 제2프로세서소자(PE2)는 재개명령을 실행한다. 일시정지명령 실행시에, 명령에 의해 나타내진 어드레스 "WAR"는 "슬립(sleep)"으로 불리는 일시정지신호와 함께 송신된다. 정지신호에 응답하여, 나타내진 어드레스가 일시정지 어드레스 레지스터에 저장된다. 일시정지신호 "슬립"을 출력한 후, 제1프로세서 소자(PE1)가 일시적으로 일시정지된(슬립)상태라고 가정한다.

재개명령은 표시된 어드레스에서 슬립하는 일시정지된 상태에 있는 모든 프로세서소자를 재개한다. 이를 위해, 표시된 어드레스는 통신버스를 경유하여 모든 프로세서 소자에 전해진다. 이 경우에, 제2 프로세서 소자는 버스허용신호 "BG"를 얻기 위해 버스요청신호 "BR"을 먼저 출력함으로써 통신버스 이용권을 보장해야 한다. 통신버스의 이용권이 보장된 후, 프로세서(PR2)는 재개신호 "회복"에 의해 표시된 어드레스신호 "어드레스"를 통신버스로 출력한다. 출력되는 어드레스는 다른 프로세서소자의 정지 어드레스 레지스터의 내용과 비교된다. 비교결과가 일치할 때 동작신호 "웨이크-업(wake-up)"이 상응하는 프로세서 소자에 보내진다. 거기에 응답하여, 일시적으로 정지된 상태에 있는 프로세서 소자는 명령의 실행을 재개한다. 일시정지된 신호와 재개신호의 출력간격이 1.5클럭주기로 가정되었지만, 이런 시간간격의 길이는 각 프로세서 소자와 다른 것들의 위치 또는 배치와 같은 여러 요소에 따라서 변한다는 것으로 이해되어야 한다.

마지막으로, 병렬동작모드와 마이크로프로세서 동작모드 사이의 데이터처리시스템을 전환하기 위해 전환기구가 설명된다.

상기 두 동작모드 전환을 위한 특정명령 같이, 두 명령, 즉 멀티프로세서 동작명령과 병렬 동작명령이 제공된다. 멀티프로세서 동작명령은 병렬성 플러그(A7)를 멀티프로세서 작동모드에 세트하는 반면에, 병렬 동작명령은 병렬성 플러그(A7)를 병렬 동작모드에 세트한다. 이들 두 명령은 동작 시스템(OS)에 의해서만 실행될 수 있는 특권 명령이다.

제22도는 예로써, 병렬 동작모드에서 멀티프로세서 동작 모드로의 전환이 동작시스템(OS)에 의해 실행되는 제어의 흐름을 설명하는 플로우차트이다. 병렬동작 모드에서 실행되는 프로그램이 종결되거나 일시적으로 정지될 때(단계 X1), OS는 동작된다(단계 X2). OS에 의해, 새로이 실행된 프로그램은 스케줄링은 거쳐 끌어내어진다(단계 X3). 이 때에, 실행될 프로그램이 존재치 않을 때, 실행될 새로운 프로그램의 출현은 "유휴"로 표시된 루프에서 대기된다. OS는 실행될 프로그램의 병렬성의 정도를 체크한다(X4). 병렬성의 정도가 높은, 즉 병렬 동작이 다소 효과적인 것을 의미할 때, 데이터 처리시스템은 OS의 제어로부터 나오는 동안 병렬 동작에 남는다. 반대로, 병렬성이 낮은 경우 멀티프로세서 동작모드에서 병렬로 복수의 프로그램을 실행하는 것은 더욱 효과적이다. 따라서, 이 경우에, 멀티프로세서 동작명령은 각 프로세서 소자가 서로 독립적으로 작동하도록 실행된다(단계 X6). 제1 프로세서 소자에 의해 실행된 프로그램이 이미 스케줄링 단계(X3)에서 선택되기 때문에, OS제어는 종결된다(단계 X7). 제2의 연속하는 프로세서 소자에 대해, 스케줄링은 이들 프로세서 소자에 의해 실행될 프로그램을 추론하는 단계(X8)에서 새로이 실행된다.

제23도는 멀티프로세서 동작에서 병렬동작 모드로의 전환에 대한 OS제어흐름을 설명하는 플로우차트이다. 다른 것들 중에서 제1프로세서 소자에 의해 실행된 프로그램이 종결되는 것이 가정된다(단계 Y1). 제22도를 참조하여 종전에 설명되었듯이, OS동작단계(Y2)와 스케줄링 단계(Y3)가 실행되고, 새로이 실행되는 프로그램의 병렬성 정도가 체크되는 단계(Y4)가 뒤이어 수행된다. 병렬성이 비교적 낮을 때, 데이터 처리시스템은 멀티프로세서 동작모드에 남고 OS제어를 빠져나간다(단계 Y5). 반면에, 병렬동작명령이 수행되는 것(단계 Y6)을 의미하는 병렬성이 높은 경우에, 동작모드는 병렬 동작모드에 전환된다. 그러나, 병렬동작 모드에서 전환은 다른 프로세서 소자가 프로그램을 실행하고 있을 때는 불가능하다. 그래서, 병렬동작 명령의 실행을 위해, 수행되고 있는 프로그램 수행의 종결이 대기되어야 한다(단계 Y7). 수행되고 있는 프로그램의 완성 또는 일시적인 정지에 대해 병렬 작동명령이 실행되도록 허용된다.

프로그램의 병렬성 정도는 소스 프로그램이 목표 프로그램으로 변환될 때의 시점에서 검출된다. 이 논리적으로, "병렬성의 정도 또는 간단한 병렬성"은 프로그램에 포함되고 병렬로 수행될 수 있는 평균갯수의 명령을 표시하기 위해 이용된다. 그래서, 병렬성의 정도는 병렬로 실행가능한 다수의 명령에 의해 프로그램에 포함된 다수의 명령을 분할함으로써 결정된다. 컴파일러 또는 다른 것에 의해 목표 프로그램에 병렬성 정도를 기록함으로써, 작동시스템(OS)이 프로그램의 병렬성을 인식하게 만드는 것이 가능하다.

제24도는 다른 것중에 멀티포트 캐시메모리의 실행과 관련된 본 발명의 다른 실시예에 따른 데이터처리 시스템의 일반적 구성을 나타낸 블록도이다. 데이터 처리시스템은 단일 LSI의 형태로 실현되고 어드레스 라인(100)과 데이터라인(102)를 포함한 버스라인을 경유하여 주메모리(A4)에 접속된다. 데이터 처리시스템(22)은 n개의 판독포트와 일개의 기록포트를 구비한 어드레스 변환버퍼 메모리(12), 각각 일개의 판독포트와 일개의 기록포트를 구비하고 연속 어드레스가 각 बैं크(1 내지 m)로 순차적으로 할당하여 데이터 저장영역이 m개의 बैं크(m개의 캐시뱅크)로 분할되는 캐시메모리를 구성하는 m개의 캐시메모리 버퍼(14-1), ..., (14-n), बैं크 선택기/판정회로(16), 데이터 선택기회로(18-1), (18-n) 및 복수의 프로세서(20-1), ..., (20-n)를 포함한다. 프로세서(20-i), 어드레스 변환 버퍼메모리(12) 그리고

선택기/결정회로(16)은 어드레스 라인(104-1), ..., (104-n)를 통해 상호 접속된다. 캐시 메모리 뱅크(14-j)(j=1 내지 m)는 차례로 데이터라인(108-i)을 경유하여 데이터 선택기회로(18-i)(i=1 내지 n)에 접속되는 뱅크선택/결정회로(16)에 접속된다. 데이터 선택회로(18-i)는 데이터라인(110-i)(i=1 내지 n)을 경유하여 프로세서(20-i)에 접속된다.

뱅크선택/결정회로(16)는 각 프로세서(20-i)로부터 들어오는 어드레스 정보에 의해 표시된 캐시메모리 뱅크(14-j)를 선택하기 위한 뱅크선택 기능으로 뿐만아니라 프로세서(20-i)로부터 공급된 어드레스 정보와 상기의 뱅크 선택기능을 통해 선택된 캐시메모리 뱅크(14-j)에 데이터 존재여부와 같은 어드레스 변환버퍼 메모리(12)로부터의 출력정보에 근거해 결정하는 판정기능으로도 작용한다. 더욱이, 상기 판정기능에 의해 유효화하는 데이터에 포함된 유효 플래그와 어드레스부분 사이의 비교가 긍정적인 대답(YES)으로 판단되면, 데이터 선택회로(18-1)는 뱅크선택 기능을 통해 선택된 캐시메모리 뱅크(4-j)에 저장된 데이터를 지정프로세서(20-i)에 전송하도록 작동한다.

이 경우에, 복수의 프로세서(20-i)에 의해 복수의 캐시뱅크(14-i)를 구비한 단일 캐시메모리의 공유는 지금까지 공지된 데이터 처리 시스템에서 캐시 결합 또는 캐시 데이터 일관성의 문제가 만족스럽게 해결되는 것을 의미한다고 언급되어야 한다.

상기 구조의 데이터 처리시스템의 작동에서, 어드레스정보가 프로세서(20-i)로부터 출력될 때, 어드레스 정보는 어드레스 변환버퍼 메모리(12)에 의해 지정된 물리적 어드레스로 변환된다. 다시 말해, 프로세서로부터의 어드레스 정보가 어드레스 변환버퍼 메모리(12)에 존재할 때, 어드레스 정보에 상응하는 물리적 어드레스를 발생한다. 더욱, 캐시메모리 뱅크에 유일하게 표시된 것(14-j)은 프로세서(20-i)로부터의 어드레스 정보와 일치하여 선택된다. 어드레스변환 버퍼메모리(12)에 의해 발생된 물리적 어드레스에 상응하는 데이터가 선택된 캐시메모리뱅크(14-j)에 존재할 때, 선택된 캐시메모리 뱅크에 위치한 데이터는 데이터 선택회로(18-i)를 거쳐 데이터가 기점을 요청한 종착 프로세서(20-i)에 보내진다.

반면에, 어드레스 변환 버퍼 메모리(12)에 의해 발생된 물리적 어드레스에 상응하는 데이터가 선택된 캐시메모리 뱅크(14-j)에 존재하지 않으면, 즉, 캐시메모리에의 액세스가 시패로 끝나는 경우에, 주메모리(A4)는 어드레스라인(100)을 거쳐 액세스되므로 관련 데이터가 데이터라인(102)를 거쳐 캐시 뱅크(14-j) 다음에 종착 프로세서(20-i)에 공급된다.

어드레스 변환 버퍼메모리(12)가 상기와 같이 복수의 캐시뱅크에 의해 구성되는 캐시메모리를 갖는 멀티 포트 형태로 구성되어 있기 때문에 복수의 프로세서(20-i)는 각 지정된 캐시메모리 뱅크(14-j)로부터 전송된 데이터를 찾기 위해 캐시메모리(14)에 동시에 액세스하는데, 이것은 차례로 캐시 적중률이 동시처리가 복수의 프로세서(20-i)에 의해 수행하도록 하는 동안 증가되는 것을 의미한다.

제25도는 두 개의 판독포트와 일개의 기록포트를 구비한 버퍼메모리(12)의 구조를 보여주는 회로도이고, 이렇게 설계된 두 프로세서가 동시에 캐시메모리로부터 데이터를 판독할 수 있다.

제25도에 있어서, 본 실시예에 따른 어드레스 변환 버퍼 메모리는, 어드레스(A0) 및 (B0)가 각각 디코더(24) 및 (26)에 입력되고, 입력 데이터(Din) 및 (Din)는 각각 인버터(28) 및 (44)에 입력되고, 데이터가 출력데이터 라인(DB0-1) 및 (DB0-2)에 각각 출력되도록, 상호 접속된 디코더(24) 및 (26), 인버터(28), 게이트(30), (32), (34) 및 (36), 메모리셀(38), 센스증폭기(40) 및 (42) 그리고 인버터(44)를 포함한다.

본 실시예의 경우에서, 메모리는 각 프로세서에 독립적으로 동작할 수 있다. 바꿔 말하면, 데이터는 프로세서에 의해 어드레스(A0)로 표시된 메모리 위치로부터 데이터라인(D<sub>2</sub>)을 경유하여 판독될 수 있다. 더욱 명확히, 프로세서가 어드레스(A0)로써 버퍼메모리에 액세스할 때, 어드레스(A0)는 디코더(24)에 의해 해독된다. 게이트(30)과 (34)가 턴온될 때, 메모리 셀(38)의 내용이 센스증폭기(42)를 통해 데이터라인(D<sub>1</sub>)과 ( $\overline{D}_1$ )을 경유하여 판독된다. 타 프로세서가 메모리 액세스를 위한 어드레스(B0)를 표시할 때, 어드레스(B0)는 디코더(26)에 의해 해독된다. 게이트(32)와 (36)이 턴온될 때, 메모리 셀(38)의 내용은 센스증폭기(40)를 통해 데이터라인(D<sub>2</sub>) 및 ( $\overline{D}_2$ )을 경유하여 판독한다.

데이터 기록시, 데이터는 역전된 분극을 갖는 데이터가 데이터라인(D<sub>1</sub>)과 ( $\overline{D}_1$ )에 보내지므로 인버터(28)과 (44)를 통해 입력된다. 데이터는 게이트(30) 및 (34)를 통해 메모리 셀(38)에 기록된다.

제25도에서 보여주는 어드레스 변환버퍼 메모리는 오직 하나의 기록포트를 갖는다. 이것은 어드레스 변환이 지정된 어드레스 데이터 히트를 실패할 때 기록요청이 발생하기 때문이다. 이 경우에, 프로세서에 의해 행해지는 액세스가 금지된다. 프로세서에 의한 메모리 액세스는 메모리 데이터가 재기록된 후에만 허가된다.

기록포트와 판독포트가 공동으로 공유되기 때문에, 본 실시예는 메모리크기의 감소에 기여하는 이점이 있다.

제26도는 본 발명의 다른 실시예에 따른 데이터 처리시스템의 일반적인 구조를 보여주는 블록도이다. 이 데이터 처리시스템의 경우에서, 복수의 일관독/일기록 포트메모리 뱅크에 의해 구성된 캐시메모리는 n개의 판독 포트와 일개의 기록포트를 갖는 어드레스 변환버퍼메모리와 두 프로세서(P1) 및 (P2)와의 결합으로 이용된다. 더욱이, 한쌍의 명령 어드레스 변환버퍼(ITLB)와 데이터 어드레스 변환버퍼(DTLB)가 제공된다. 캐시 메모리는 세 뱅크로 분할되어 각각 명령 캐시뱅크(IC<sub>1</sub>), (IC<sub>2</sub>) 및 (IC<sub>3</sub>) 그리고 데이터 캐시뱅크(DC<sub>1</sub>), (DC<sub>2</sub>) 및 (DC<sub>3</sub>)를 직접 제공한다. 데이터 변환 또는 전송은 직접 맵(map)계획과 일치하여 수행된다.

제26도를 참조하여, 프로세서(P<sub>1</sub>)와 (P<sub>2</sub>)에 의해 실행되는 명령의 어드레스(VA<sub>1</sub>) 및 (VA<sub>2</sub>)가 명령 어드레

스 변환버퍼(ITLB)와 명령캐시(IC<sub>1</sub>), (IC<sub>2</sub>) 및 (IC<sub>3</sub>)에 출력될 때, 명령 어드레스 변환 버퍼(ITLB)는 어드레스(VA<sub>1</sub>)(=A<sub>0</sub>+A<sub>1</sub>+A<sub>2</sub>) 및 (VA<sub>2</sub>)(=B<sub>0</sub>+B<sub>1</sub>+B<sub>2</sub>)의 어드레스 부분(A<sub>0</sub>) 및 (B<sub>0</sub>)을 입력으로서 주신하고 어드레스 변환이 연속으로 종결되는지를 표시하면서 신호라인(IS<sub>1</sub>) 및 (IS<sub>2</sub>)에 보내고, 이들 신호는 데이터 선택회로(IPS<sub>1</sub>) 및 (IPS<sub>2</sub>)에 공급된다. 동시에, 어드레스(VA<sub>1</sub>)의 어드레스정보(A<sub>1</sub>)와 어드레스(VA<sub>2</sub>)의 어드레스 정보(B<sub>1</sub>)는 뱅크 선택회로(ICB)에 출력된다. 더욱이, 이 시점에서, 가상 어드레스(VA<sub>1</sub>) 및 (VA<sub>2</sub>)의 어드레스포트(A<sub>2</sub>) 및 (B<sub>2</sub>)는 또한 뱅크선택을 위해 뱅크 선택회로(ICB)에 출력된다. 뱅크선택회로(ICB)는 다음에 어드레스 부분(A<sub>2</sub>)이 어드레스부분(B<sub>2</sub>)과 동일한 지를 체크한다. 둘이 같을 때, 어드레스(VA<sub>1</sub>) 및 (VA<sub>2</sub>)는 같은 캐시뱅크에 액세스하도록 결정된다. 이 경우에, 어드레스(VA<sub>2</sub>)가 저장되며 어드레스(VA<sub>1</sub>)는 우선순위 또는 우선권이 주어진다.

반면에, 어드레스정보(A<sub>2</sub>) 및 (B<sub>2</sub>)가 서로 다를 때, 액세스가 다른 캐시뱅크에 행해지도록 판정된다. 여기에서 어드레스 정보(A<sub>2</sub>)의 값에 상응하는 어드레스 라인(ICS<sub>1</sub>), (ICS<sub>2</sub>) 및 (ICS<sub>3</sub>)중 하나가 명령캐시선택 어드레스에 선택되어 선택된 어드레스 라인에 어드레스명령(A<sub>1</sub>)을 보낸다. 동일하게, 어드레스정보(B<sub>2</sub>)에 대해, 어드레스정보(B<sub>2</sub>)의 값에 상응하는 어드레스 라인(ICS<sub>1</sub>), (ICS<sub>2</sub>) 및 (ICS<sub>3</sub>)중 하나가 어드레스 정보(B<sub>1</sub>)에 공급된다. 본 실시예의 경우에서, 캐시가 세 뱅크로 분할 되기 때문에, 어드레스 정보(A<sub>2</sub>) 및 (B<sub>2</sub>)는 각각 두 비트이면 충분하다. 예로서, 어드레스 정보(A<sub>2</sub>) 및 (B<sub>2</sub>)가 제로 값일 때, 어드레스 정보(A<sub>1</sub>) 및 (B<sub>1</sub>)는 어드레스 라인(ICS<sub>1</sub>)에 공급된다. 어드레스 정보(A<sub>2</sub>) 및 (B<sub>2</sub>)가 "1" 값일 때, 어드레스 정보(A<sub>1</sub>) 및 (B<sub>1</sub>)은 어드레스 라인(ICS<sub>2</sub>)에 공급된다. 어드레스 정보(A<sub>2</sub>) 및 (B<sub>2</sub>)가 "2" 값일 때 어드레스 정보(A<sub>1</sub>) 및 (B<sub>1</sub>)은 어드레스 라인(ICS<sub>3</sub>)에 공급된다. 더욱이, 전자가 "3"값일 때, 후자는 어드레스 라인(ICS<sub>3</sub>)에 또한 공급된다. 이러한 방법으로, 언급된 캐시 뱅크는 가상어드레스(VA<sub>1</sub>)과 (VA<sub>2</sub>)의 특정비트(A<sub>2</sub>, B<sub>2</sub>)를 이용하여 결정될 수 있어, 캐시뱅크(IC<sub>1</sub>, IC<sub>2</sub>) 또는 (IC<sub>3</sub>)에 위치한 상응데이터는 정보(A<sub>1</sub>) 또는 (B<sub>1</sub>)를 이용하여 각각 데이터라인(ICD<sub>1</sub>), (ICS<sub>2</sub>) 및 (ICD<sub>3</sub>)에 출력될 수 있다.

각 프로세서(P<sub>1</sub>)과 (P<sub>2</sub>)에 상응하여 각각 제공된 명령 데이터 선택회로(IPS<sub>1</sub>) 및 (IPS<sub>2</sub>)는 어드레스 변환을 거친 정보(변역된 어드레스정보)와 신호라인(IS<sub>1</sub>) 또는 (IS<sub>2</sub>)와 각각 어드레스 정보(VA<sub>1</sub>) 또는 (VA<sub>2</sub>)에서 유도된 뱅크정보(A<sub>2</sub>) 또는 (B<sub>2</sub>)를 경유하여 공급된 정보에 근거하여 명령 데이터라인(ICD<sub>1</sub>), (ICD<sub>2</sub>) 및 (ICD<sub>3</sub>)중 하나로부터 데이터를 가져와서 데이터라인(ID<sub>1</sub>) 또는 (ID<sub>2</sub>)에 입력 데이터를 보낸다.

이와 관련하여, 변환된 어드레스정보(IS<sub>1</sub>, IS<sub>2</sub>)는 어드레스(VA<sub>1</sub>) 또는 (VA<sub>2</sub>)에 의해 표시된 데이터가 캐시 메모리에서 유효한지에 대해 결정하도록 작용한다. 데이터가 캐시에 존재하지 않으면, 프로세서(P<sub>1</sub>) 및 (P<sub>2</sub>)에 전송된 데이터는 데이터 선택회로(IPS<sub>1</sub>) 및 (IPS<sub>2</sub>)에 의해 일시정지된다. 반대로, 데이터가 캐시 메모리에서 유효할 때, 데이터선택회로(IPS<sub>1</sub>) 및 (IPS<sub>2</sub>)에 의해 선택된 데이터가 각각 데이터라인(IP<sub>1</sub>) 및 (IP<sub>2</sub>)을 경유하여 프로세서(P<sub>1</sub>) 및 (P<sub>2</sub>)에 공급된다. 따라서, 프로세서(P<sub>1</sub>) 및 (P<sub>2</sub>)는 서로 병렬로 작동된다.

다음에, 데이터 어드레스 변환 버퍼메모리(DTLB)와 데이터 캐시메모리뱅크(DC<sub>1</sub>), (DC<sub>2</sub>) 및 (DC<sub>3</sub>)의 작동과 기능에 대해 설명한다.

프로세서(P<sub>1</sub>) 및 (P<sub>2</sub>)가 데이터를 참조하기 위한 레지스터(VA<sub>3</sub>) 또는 (VA<sub>4</sub>)를 발생할 때, 어드레스(VA<sub>4</sub>)의 어드레스정보(A<sub>3</sub>)는 데이터 어드레스 변환 버퍼 메모리(DTLB)에 공급되고, 동시에 어드레스 변환의 결과는 신호라인(DS<sub>1</sub>) 및 (DS<sub>2</sub>)을 경유하여 데이터 선택회로(DPS<sub>1</sub>) 및 (DPS<sub>2</sub>)에 공급된다. 더욱이, 이 시점에서, 어드레스(VA<sub>3</sub>)의 어드레스 정보(A<sub>4</sub>) 및 (A<sub>5</sub>)와 어드레스(VA<sub>4</sub>)의 어드레스정보(B<sub>4</sub>) 및 (B<sub>5</sub>)는 또한 데이터 캐시뱅크 선택회로(DCB)에 공급된다. 뱅크선택회로(DCB)는 뱅크선택정보(A<sub>5</sub>) 및 (B<sub>5</sub>)와 일치하여 어드레스라인(DCS<sub>1</sub>) 내지 (DCS<sub>3</sub>)중 하나에 어드레스 정보(A<sub>4</sub>), (B<sub>4</sub>)를 공급하도록 구성된다. 그래서, 데이터는 데이터라인(DC<sub>1</sub>), (DC<sub>2</sub>) 및 (DC<sub>3</sub>)상에 출력된 캐시뱅크(DC<sub>1</sub>), (DC<sub>2</sub>) 및 (DC<sub>3</sub>)중 둘로부터 판독된다. 이 경우에, 어드레스정보(A<sub>5</sub>)가 어드레스정보(B<sub>5</sub>)와 같을 때, 이것은 뱅크 충돌이나 경합이 일어나는 것을 의미한다. 따라서, 어드레스정보(A<sub>4</sub>) 및 (B<sub>4</sub>)는 경합이 일어나는 뱅크에 순차적으로 공급된다. 데이터 선택회로(DPS<sub>1</sub>) 및 (DPS<sub>2</sub>)는 신호라인(DS<sub>1</sub>) 및 (DS<sub>2</sub>)의 신호와 어드레스 정보(A<sub>5</sub>) 및 (B<sub>5</sub>)에 근거하여 데이터라인(DCD<sub>1</sub>), (DCD<sub>2</sub>) 및 DCD<sub>3</sub>)중 하나의 데이터를 각각 데이터라인(DD<sub>1</sub>) 및 (DD<sub>2</sub>)을 경유하여 프로세서(P<sub>1</sub>) 및 (P<sub>2</sub>)에 전송된다. 그래서, 프로세서(P<sub>1</sub>) 및 (P<sub>2</sub>)는 서로 병렬로 작동할 수 있다.

제27도는 명령 데이터 선택회로(IPS<sub>1</sub>) 및 (IPS<sub>2</sub>)의 전형적인 구조를 보여주는 블록도이다.

도면을 참조하여, 명령 어드레스 변환 버퍼메모리(ITLB)는 가상 페이지(VPN)에 상응하는 물리적 페이지(PPN)에 의해 구성되고, 여기에서 세 캐시뱅크(IC<sub>1</sub>), (IC<sub>2</sub>) 및 (IC<sub>3</sub>)는 물리적 페이지(PPN)에 상응하는 데이터 "데이터"에 의해 구성된다. 데이터선택회로(IPS<sub>1</sub>)은 제27도에서 접속된 비교기(CMP<sub>1</sub>), (CMP<sub>2</sub>) 및 (CMP<sub>3</sub>) 그리고 선택기(AS<sub>1</sub>), (DS<sub>1</sub>) 및 (PS<sub>1</sub>)를 포함하는 반면, 데이터 선택회로(IPS<sub>2</sub>)는 비교기(CMP<sub>4</sub>), (CMP<sub>5</sub>) 및 (CMP<sub>6</sub>) 그리고 선택기(AS<sub>2</sub>), (DS<sub>2</sub>) 및 (PS<sub>2</sub>)로 구성된다.

상기 구조에서, 버퍼메모리(ITLB)는 변환된 어드레스 정보(즉, 어드레스 변환을 거친 정보)(IS<sub>1</sub>) 과

(IS<sub>2</sub>)를 출력함으로써 어드레스정보(A<sub>0</sub>) 및 (B<sub>0</sub>)의 입력에 응답한다. 변환된 어드레스정보(IS<sub>1</sub>) 및 (IS<sub>2</sub>)와 일치하여, 결합된 데이터선택회로(IPS<sub>1</sub>) 및 (IPS<sub>2</sub>)는 동시에 병렬 작동을 수행한다. 더욱 명확히, 선택회로(IPS<sub>1</sub>)에서, 변환된 어드레스 정보(IS<sub>1</sub>)의 부분을 구성하는 가상페이지(VPN)의 정보는 비교기(CMP<sub>1</sub>)에 의해 어드레스 정보(A<sub>0</sub>)와 비교되어 어드레스변환이 성공적으로 수행되었는지의 여부를 체크한다. 이 시점에서, 표시된 캐시뱅크(IC<sub>1</sub>), (IC<sub>2</sub>) 및 (IC<sub>3</sub>)는 은행 선택정보(ICS<sub>1</sub>), (ICS<sub>2</sub>) 및 (ICS<sub>3</sub>)에 근거해서 데이터(ICD<sub>1</sub>), (ICD<sub>2</sub>) 또는 (ICD<sub>3</sub>)로서 물리적페이지(PPN)의 데이터를 출력하고, 여기에서 출력된 데이터는 선택기(AS<sub>1</sub>) 및 (DS<sub>1</sub>)에 입력된다. 선택기(DS<sub>1</sub>)는 은행선택정보(A<sub>2</sub>)를 이용하여 비교기(CMP<sub>2</sub>)는 버퍼메모리(ITLB)의 물리적 페이지(PPN)의 데이터를 선택기(AS<sub>1</sub>)에 의해 선택된 캐시뱅크의 물리적 페이지(PPN)의 데이터와 비교하여 양 데이터의 내용이 서로 일치하는지를 판정한다. 판정의 결과는 비교기(CMP<sub>3</sub>)에 출력된다. 다시 말해서, 비교기(CMP<sub>2</sub>)는 어드레스(VA<sub>1</sub>)에 상응하는 데이터가 캐시에 존재하는지에 대해 판정하기 위해 판정기능으로 작동한다.

이제, 비교기(CMP<sub>3</sub>)는 어드레스 변환과 캐시 액세스가 성공적으로 수행되는지에 대해 판정한다. 판정이 긍정이면 선택기(PS<sub>1</sub>)는 선택기(DS<sub>1</sub>)로부터 출력을 데이터라인(ID<sub>1</sub>)에 출력하고, 그렇지 않으면, 주메모리(A<sub>4</sub>)로부터의 데이터가 선택기(PS<sub>1</sub>)에 의해 데이터라인(ID<sub>1</sub>)에 출력된다.

반면에, 캐시뱅크(IC<sub>1</sub>), (IC<sub>2</sub>) 및 (IC<sub>3</sub>)의 데이터 "데이터"는 선택기(DS<sub>1</sub>)에 입력되고, 여기에서 어드레스 정보와 관련하여 선택된 데이터는 어드레스(VA<sub>1</sub>)에 상응하는 데이터를 데이터라인(ID<sub>1</sub>)에 차례로 보내는 선택기(PS<sub>1</sub>)에 공급된다. 여기에서 알 수 있는 바와 같이 선택기(DS<sub>1</sub>)과 (DS<sub>2</sub>)는 데이터선택으로 작용한다.

데이터 선택회로(IPS<sub>2</sub>)는 상기의 데이터 선택회로(IPS<sub>1</sub>)와 병렬로 동일한 처리를 수행한다. 변환된 어드레스 정보(IS<sub>2</sub>)가 입력될 때, 가상페이지(VPN)의 정보는 비교기(CMP<sub>4</sub>)에 의해 어드레스정보(B<sub>0</sub>)와 비교되고, 이 비교의 결과는 비교기(CMP<sub>5</sub>)에 입력된다. 변환된 어드레스 정보(IS<sub>2</sub>)(즉, 어드레스 변환에 따른 정보)의 남은 정보인 물리적 페이지(PPN)의 정보는 비교기(CMP<sub>5</sub>)에 공급된다. 더욱이, 데이터 "데이터"의 정보와 복수의 뱅크로 분할된 구조를 갖는 캐시메모리의 캐시뱅크(IC<sub>1</sub>), (IC<sub>2</sub>) 및 (IC<sub>3</sub>)로부터 출력된 물리적 페이지(PPN)의 정보는 각각의 선택기(AS<sub>2</sub>) 및 (DS<sub>2</sub>)에 공급되고, 상응하는 물리적 페이지(PPN)와 데이터 "데이터"는 뱅크 선택정보와 관련하여 비교기(CMP<sub>5</sub>)와 선택기(PS<sub>2</sub>)에 공급된다. 이런 방법으로, 물리적 페이지(PPN)의 상호비교는 비교기(CMP<sub>5</sub>)에 의해 수행되고, 이들의 결과는 비교기(CMP<sub>6</sub>)에 출력되며, 어드레스 변환이 성공적으로 수행되는지와 데이터가 캐시에서 유효한지에 대해 판정된다. 이런 판정의 결과에 따라서, 선택기(PS<sub>2</sub>)가 제어된다. 더욱 명확하게, 비교기(CMP<sub>6</sub>)의 출력이 "온"일 때, 선택기(DS<sub>2</sub>)의 출력은 데이터라인(ID<sub>2</sub>)에 보내지고, 이 비교기의 출력이 "오프"인 경우에 데이터는 데이터라인(ID<sub>2</sub>)에 보내진 주메모리(A<sub>4</sub>)로부터 판독된다.

제28도는 데이터 선택회로(DPS<sub>1</sub>) 및 (DPS<sub>2</sub>)의 전형적인 구조를 보여주는 블록도이다.

데이터 선택회로(DPS<sub>1</sub>)는 비교기(CMP<sub>7</sub>), (CMP<sub>8</sub>) 및 (CMP<sub>9</sub>) 그리고 선택기(AS<sub>3</sub>), (OS<sub>3</sub>) 및 (PS<sub>3</sub>)로 구성되는 반면, 데이터 선택회로(DPS<sub>2</sub>)는 비교기(CMP<sub>10</sub>), (CMP<sub>11</sub>) 및 (CMP<sub>12</sub>) 그리고 선택기(AS<sub>4</sub>), (OS<sub>4</sub>) 및 (PS<sub>4</sub>)로 구성된다. 데이터 어드레스 변환버퍼 메모리(DTLB)는 가상페이지(VPN)와 상응하는 물리적 페이지(PPN)에 의해 구성된다. 캐시 메모리는 세 캐시뱅크(DC<sub>1</sub>), (DC<sub>2</sub>) 및 (DC<sub>3</sub>)로 분할되고, 캐시뱅크(DC<sub>1</sub>) 내지 (DC<sub>3</sub>)의 각각은 물리적 페이지(PPN)와 거기에 상응하는 데이터 "데이터"로 구성된다.

상기 구조에서, 어드레스 정보(A<sub>3</sub>), (B<sub>3</sub>)가 데이터 어드레스 변환 버퍼메모리(DTLB)에 입력될 때, 어드레스 변환에서 얻어진 변환된 어드레스정보(DS<sub>1</sub>), (DS<sub>2</sub>)는 버퍼메모리(DTLB)에서 출력된다. 변환된 어드레스 정보에 근거하여, 두 데이터 선택회로(DPS<sub>1</sub>) 및 (DPS<sub>2</sub>)는 서로 병렬로 작동을 수행한다. 더욱 명확히, 데이터 선택회로(DPS<sub>1</sub>)에서, 변환된 어드레스 정보(DS<sub>1</sub>)의 부분을 구성하는 가상페이지(VPN)의 정보는 비교기(CMP<sub>7</sub>)에 의해 어드레스정보(A<sub>3</sub>)의 내용과 비교되어 어드레스 변환이 성공적으로 종결되는지에 대해 판정된다. 이 시점에서, 물리적 페이지(PPN)와 데이터 "데이터"는 각각 데이터(AS<sub>3</sub>) 및 (DS<sub>3</sub>)에 입력된 뱅크 선택정보(DCS<sub>1</sub>) 내지 (DCS<sub>3</sub>)와 관련하여 캐시뱅크중 하나로부터 출력된다. 선택기(AS<sub>3</sub>)는 뱅크 선택정보(A<sub>5</sub>)와 관련된 연합 뱅크의 출력(PPN)을 비교기(CMP<sub>8</sub>)에 공급한다. 다음에 비교기(CMP<sub>8</sub>)는 버퍼메모리(DTLB)로부터 공급된 정보를 선택기(AS<sub>3</sub>)로부터의 정보와 비교하여 물리적 페이지(PPN)의 정보와 일치하는지의 여부를 판정한다. 다시말해, 비교기(CMP<sub>8</sub>)는 어드레스(VA<sub>3</sub>)에 상응하는 데이터가 캐시에 존재하는지를 체크하는 기능으로 작용한다.

비교기(CMP<sub>9</sub>)는 어드레스 변환 캐시액세스가 성공적으로 종결되는지에 대해 비교기(CMP<sub>7</sub>) 및 (CMP<sub>8</sub>)로부터의 출력에 근거해 판정하고, 판정의 결과는 선택기(PS<sub>3</sub>)에 보내진다. 비교기(CMP<sub>9</sub>)에 의한 판정의 대답이 긍정적일 때, 즉 어드레스 변환과 캐시액세스 모두 성공적으로 수행되는 것을 나타낼 때, 선택기(DS<sub>3</sub>)에 의해 선택된 데이터는 데이터라인(DD<sub>1</sub>)에 출력되는 반면, 상기 결정의 대답이 부정적일 때, 즉 상기 처가 실예로 끝나면, 데이터는 데이터 라인에 출력된 주메모리(A<sub>4</sub>)로부터 판독된다.

반면에 캐시뱅크(DC<sub>1</sub>) 내지 (DC<sub>3</sub>)중 하나가 선택될 때, 선택된 캐시뱅크의 데이터는 선택기(DS<sub>3</sub>)에 입력

되고, 어드레스정보(A<sub>3</sub>)와 관련하여 지정된 데이터는 선택기(S<sub>3</sub>)에 공급된다. 따라서, 후자는 어드레스(VA<sub>3</sub>)에 상응하는 데이터를 데이터라인(DD<sub>1</sub>)에 보낼 수 있다.

더욱이, 데이터 선택회로(DPS<sub>2</sub>)는 데이터 선택회로(DPS<sub>1</sub>)와 병렬로 동일한 처리를 수행한다. 먼저, 변환된 어드레스 정보의 부분을 구성한 물리적 페이지(PPN)의 정보는 비교기(CMP<sub>10</sub>)에 의해 어드레스정보(B<sub>3</sub>)와 비교되어 비교기(CMP<sub>12</sub>)로 출력된다. 부가하여, 변환된 어드레스 정보의 다른 부분을 이루는 물리적 페이지(PPN)의 정보는 비교기(CMP<sub>11</sub>)에 입력되어 선택기(AS<sub>4</sub>)로부터의 정보와 비교된다. 물리적 페이지(PPN)와 캐시뱅크(DC<sub>1</sub>) 내지 (DC<sub>3</sub>)로부터 출력된 데이터 "데이터"는 각각 선택기(AS<sub>4</sub>) 및 (DS<sub>4</sub>)에 입력되어, 물리적 페이지(PPN)의 정보와 뱅크 선택정보(B<sub>5</sub>)에 상응하는 데이터 "데이터"의 정보는 각각 비교기(CMP<sub>11</sub>)와 선택기(PS<sub>4</sub>)에 공급된다. 비교기(CMP<sub>11</sub>)에 의해 버퍼메모리(DTLB)의 물리적 페이지(PPN)의 정보와 캐시의 정보 사이의 일치가 검출될 때, 선택기(AS<sub>4</sub>)에 의해 선택된 데이터는 비교기(CMP<sub>12</sub>)에 공급된다. 어드레스 변환이 성공적으로 종결되는 것이 비교기(CMP<sub>11</sub>)에 의해 판정될 때, 이것은 데이터가 캐시에 존재하는 것을 의미한다. 비교기(CMP<sub>12</sub>)는 어드레스 변환의 성공을 표시하는 비교기(CMP<sub>10</sub>)의 출력과 캐시억세스의 성공을 표시하는 비교기(CMP<sub>11</sub>)의 출력으로 입력되어 둘이 성공적으로 종결되는지에 대해 판정되고, 판정결과는 선택기(DS<sub>4</sub>)로부터의 데이터에 공급된 다른 입력을 갖는 선택기(PS<sub>4</sub>)에 공급된다. 캐시뱅크(DC<sub>1</sub>) 내지 (DC<sub>3</sub>)중 하나에 존재하는 데이터 "데이터"는 뱅크 선택정보(B<sub>5</sub>)에 따라서 선택된다. 선택된 데이터는 비교기(CMP<sub>12</sub>)가 "온"일 때 선택기(DS<sub>4</sub>)를 경유하여 데이터라인(DD<sub>2</sub>)에 출력된다. 반면에, 비교기(CMP<sub>12</sub>)가 "오프"일 때, 어드레스(VA<sub>4</sub>)에 상응하는 데이터는 주메모리(A<sub>4</sub>)로부터 판독되어 데이터라인(DD<sub>2</sub>)에 출력된다. 이런 방식으로, 프로세서(P<sub>1</sub>)과 (P<sub>2</sub>)는 서로 병렬로 작동한다.

제29도는 본 발명의 다른 실시예에 따른 데이터 처리시스템의 구조를 보여주는 블록도이다. 이 도면에서 보여주듯이, 명령 어드레스 변환 버퍼와 캐시 메모리를 각각 포함한 각 메모리(50-1) 내지 (50-n)는 일개 판독포트와 일개의 기록포트를 구비한 메모리로 구성된다. 더 변환 버퍼메모리(12A)는 n개의 판독포트와 일개의 기록 포트를 구비한 메모리로 구성된다. 데이터 캐시메모리(14A)는 각 캐시뱅크가 일개의 판독포트와 일개의 기록포트를 구비하도록 m개의 뱅크(14A-1), ..., (14A-m)으로 분할된다. 메모리(50-1) 내지 (50-n)는 멀티플렉서(48)에 접속되고, 이 출력은 버퍼메모리(12A)와 캐시메모리(14-1), ..., (14-m)와 함께 멀티플렉서(46)에 접속되며, 멀티플렉서(46)은 주메모리(A4)에 접속된다.

본 실시예는 거의 재기록되지 않는 명령캐시 메모리를 값비싼 멀티포트메모리로 구성될 필요가 없기 때문에 하드웨어의 축소에 기여할 수 있다.

제30도는 본 발명에 따른 데이터 처리시스템의 다른 실시예를 보여주는 블록도이다. 도면을 참조하여, 복수의 프로세서(P<sub>1</sub>), ..., (P<sub>n</sub>)가 데이터 처리시스템에 장착될 때, 멀티플렉서(54) 및 (56)는 프로세서(P<sub>1</sub>), ..., (P<sub>n</sub>)의 입력과 출력측에 배치되고, 프로세서, 명령 캐시메모리(IC) 및 데이터 캐시메모리(DC)는 멀티플렉서(54) 및 (56)를 거쳐 상호 접속되는 반면, 명령 어드레스 변환 버퍼메모리(ITLB), 명령 캐시메모리(IC), 데이터어드레스 변환 버퍼메모리(DTLB) 그리고 데이터 캐시메모리(DC)는 멀티플렉서(52)를 거쳐 주메모리(A4)에 억세스가 멀티처리를 통해 실현되고 프로세서는 병렬로 캐시 메모리에 억세스 된다.

제31도는 프로세서(i)의 내부구조를 보여주는 블록도이다.

도면에서 보여주듯이, 프로세서(i)는 두 명령디코더(DEC<sub>1</sub>) 및 (DEC<sub>2</sub>), 두 연산장치(ALU<sub>1</sub>) 및 (ALU<sub>2</sub>), 4출력/2입력 레지스터 파일(RF), 제1 메모리 어드레스 레지스터(MAR<sub>1</sub>), 제1 메모리 데이터 레지스터(MDR<sub>1</sub>), 제2 메모리 어드레스 레지스터(MAR<sub>2</sub>) 그리고 제2 메모리 데이터 레지스터(MDR<sub>2</sub>)를 포함한다. 여기에서 레지스터(MAR<sub>1</sub>) 및 (MAR<sub>2</sub>)는 각각 어드레스 라인(A<sub>10</sub>) 및 (A<sub>11</sub>)에 접속되고, 레지스터(MDR<sub>1</sub>) 및 (MDR<sub>2</sub>)는 데이터 라인(D<sub>10</sub>) 및 (D<sub>11</sub>)에 각각 접속되고, 디코더(DEC<sub>1</sub>)는 명령버퍼 레지스터(IBR<sub>1</sub>)를 거쳐 어드레스 라인(A<sub>10</sub>)에 접속되며, 디코더(DEC<sub>2</sub>)는 명령 버퍼레지스터(IBR<sub>2</sub>)를 거쳐 데이터라인(D<sub>11</sub>)에 접속된다.

상기 프로세서의 구조로, 프로세서에 연합된 두 연산장치(ALU<sub>1</sub>) 및 (ALU<sub>2</sub>)은 각각 레지스터(MAR<sub>1</sub>) 및 (MAR<sub>2</sub>)를 이용함으로써 독자적으로 메모리 억세스를 수행할 수 있다. 그래서, 단일 프로세서는 동시에 복수의 명령을 수행하는 반면, 복수의 프로세서는 병렬 처리를 행하므로, 데이터 처리속도는 현저히 증가된다.

여기에서 알 수 있는 바와 같이, 어드레스 변환 버퍼메모리의 판독포트가 멀티포트 형태로 수행되고, 캐시메모리는 복수의 프로세서가 병렬로 캐시 메모리에 억세스될 수 있도록, 데이터 처리시스템은 복수의 캐시뱅크로 공유되도록 구성되어 있기 때문에 후자의 적중률은 복수의 프로세서에 의해 공유된 많은 데이터가 존재하는 경우에도 증가될 수 있다. 더욱이, 캐시메모리가 일개의 판독 포트와 일개의 기록포트를 가진 메모리에 의해 구성될 때, 칩크기는 상당히 감소될 수 있다.

**(57) 청구의 범위**

**청구항 1**

데이터처리시스템에 있어서, 주 메모리;

상기 주 메모리로부터 명령을 판독하는 프로그램 카운터 레지스터, 복수의 연산장치, 상기 복수의 연산

장치에서 얻어진 동작결과를 저장하는 레지스터 화일, 및 상기 프로그램 카운터와 상기 복수의 연산장치 및 상기 레지스터 화일을 서로 접속하는 접속 라인을 각각 포함하며, 명령을 실행하는 복수의 프로세서;

상기 복수의 프로세서 소자의 접속라인을 서로 접속하는 또 다른 접속라인;

상기 복수의 프로세서 중 하나의 일개의 프로그램 카운터 레지스터에 의해 지시된 적어도 하나의 명령을 판독하고, 상기 복수의 연산장치에서 각각 병렬동작을 실행하도록 상기 복수의 프로세서중 상기 하나의 상기 또 다른 접속라인 및 상기 접속라인을 통하여 각각의 상기 적어도 하나의 명령을 상기 복수의 프로세서에 공급하고, 그 결과가 상기 복수의 프로세서중 상기 하나의 상기 레지스터 화일 내에 기록되도록 하는 명령공급수단을 각각 포함하는 복수의 병렬동작 제어수단;

상기 복수의 프로세서의 각각의 프로세서 카운터 레지스터에 의해 지시된 상기 적어도 하나의 명령을 판독하고, 상기 복수의 프로세서에서 병렬동작을 각각 실행하도록 상기 복수의 프로세서중 상기 하나의 상기 접속라인을 통하여 상기 각각의 적어도 하나의 명령을 상기 복수의 프로세서에 공급하고, 그 결과가 상기 복수의 프로세서의 각각의 레지스터 화일 내에 기록되도록 하는 멀티프로세서동작 제어수단; 및

상기 복수의 병렬동작 제어수단 또는 멀티프로세싱 수단 중 어느 하나를 동작시키도록 스위칭하는 플래그를 갖는 플래그수단을 포함하는 병렬성 제어 수단을 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 2**

제1항에 있어서, 상기 병렬성 제어수단은 상기 병렬동작 제어수단과 상기 멀티프로세서동작 제어수단에 상기 플래그수단의 출력을 전송하는 출력라인을 포함하고, 상기 플래그는 실행된 소정의 명령에 따라 설정되는 것을 특징으로 하는 데이터 처리 시스템.

**청구항 3**

제1항에 있어서, 상기 복수의 프로세서에 공통적으로 접속된 캐시 메모리; 및

상기 플래그에 의해 지시되는, 병렬동작모드와 멀티프로세서동작모드 중 어느 모드에 상기 데이터 처리 시스템이 있는지의 여부에 따라 상기 캐시 메모리 및 상기 복수의 프로세서 사이의 액세스를 위한 데이터 판독/기록 폭 및 데이터 판독/기록 횟수를 변경하는 캐시제어수단을 더 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 4**

제3항에 있어서, 상기 캐시 메모리는, 복수의 캐시 뱅크; 및

상기 복수의 프로세서로부터 보내진 액세스 어드레스 정보에 의거하여 상기 복수의 캐시 뱅크 중 하나를 선택하는 선택수단을 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 5**

제4항에 있어서, 상기 선택수단은 액세스될 데이터가 상기 선택수단에 의해 선택된 상기 복수의 캐시 뱅크 중 하나에 존재하는지의 여부를 결정하는 결정수단을 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 6**

제5항에 있어서, 상기 복수의 프로세서로부터의 액세스 어드레스 정보가 로드되어 상기 액세스 어드레스 정보를 상기 주 메모리의 물리 어드레스로 변환하는 변환 록 어사이드 버퍼를 더 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 7**

데이터처리시스템에 있어서, 주 메모리;

상기 주 메모리로부터 명령을 판독하는 프로그램 카운터 레지스터, 복수의 연산장치, 상기 복수의 연산장치에서 얻어진 동작결과를 저장하는 레지스터 화일, 및 상기 프로그램 카운터와 상기 복수의 연산장치 및 상기 레지스터 화일을 서로 접속하는 접속 라인을 각각 포함하며, 명령을 실행하는 복수의 프로세서( $n$ 은 1보다 큰 정수);

상기  $n$ 개의 프로세서 소자의 접속라인을 서로 접속하는 또 다른 접속라인;

상기  $n$ 개의 프로세서 중 하나의 일개의 프로그램 카운터 레지스터에 의해 지시된 복수의  $n \times m$ 개의 명령( $m$ 은 1보다 큰 정수)을 판독하고, 상기 복수의 연산장치에서 각각 병렬동작을 실행하도록 상기  $n$ 개의 프로세서중 상기 하나의 상기 또 다른 접속라인 및 상기 접속라인을 통하여 각각의 상기  $m$ 개의 명령을  $n$ 개의 프로세서에 공급하고, 그로부터의 결과가 상기  $n$ 개의 프로세서중 상기 하나의 상기 레지스터 화일 내에 기록되도록 하는 명령공급수단을 각각 포함하는 복수의 병렬동작 제어수단;

상기  $n$ 개의 프로세서의 각각의 프로세서 카운터 레지스터에 의해 지시된 상기  $m$ 개의 명령을 판독하고, 상기  $n$ 개의 프로세서에서 병렬동작을 각각 실행하도록, 상기  $n$ 개의 프로세서중 상기 하나의 상기 접속라인을 통하여 상기  $m$ 개의 명령을 상기  $n$ 개의 프로세서에 공급하고, 그 결과가 상기  $n$ 개의 프로세서의 각각의 레지스터 화일 내에 기록되도록 하는 멀티프로세서동작 제어수단; 및

상기 복수의 병렬동작 제어수단 또는 멀티프로세싱 수단 중 어느 하나를 동작시키도록 스위칭하는 플래그를 갖는 플래그수단을 포함하는 병렬성 제어 수단을 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 8**

제7항에 있어서, 상기 병렬성 제어수단은 상기 병렬동작 제어수단과 상기 멀티프로세서동작 제어수단에 상기 플래그수단의 출력을 전송하는 출력라인을 포함하고, 상기 플래그는 실행된 소정의 명령에 따라 설정되는 것을 특징으로 하는 데이터처리시스템.

**청구항 9**

제7항에 있어서, 상기  $n$ 개의 프로세서에 공통적으로 접속된 캐시 메모리; 및 상기 플래그에 의해 지시되는, 병렬동작모드와 멀티프로세서 동작모드 중 어느 모드에 상기 데이터 처리시스템이 있는지의 여부에 따라 상기 캐시 메모리 및 상기  $n$ 개의 프로세서 사이의 액세스를 위한 데이터 판독/기록 폭 및 데이터 판독/기록 횟수를 변경하는 캐시제어수단을 더 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 10**

제9항에 있어서, 상기 캐시 메모리는, 복수의 캐시 뱅크; 및

상기 복수의 프로세서로부터 보내진 액세스 어드레스 정보에 의거하여 상기 복수의 캐시 뱅크 중 하나를 선택하는 선택수단을 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 11**

제10항에 있어서, 상기 선택수단은 액세스될 데이터가 상기 선택수단에 의해 선택된 상기 복수의 캐시 뱅크 중 하나에 존재하는지의 여부를 결정하는 결정수단을 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 12**

제11항에 있어서, 상기  $n$ 개의 프로세서로부터의 액세스 어드레스정보가 로드되어 상기 액세스 어드레스 정보를 상기 주 메모리의 물리 어드레스로 변환하는 변환 록 어사이드 버퍼를 더 포함하는 것으로 하는 데이터처리시스템.

**청구항 13**

제12항에 있어서, 상기 변환 록 어사이드 버퍼는, 상기  $n$ 개의 프로세서에 각각 접속되어 상기 주 메모리 내의 데이터와 어드레스에의 액세스를 위한 어드레스 정보를 수신하고 상기 어드레스정보를 상기 주 메모리의 물리 어드레스로 변환하는 어드레스변환 록어사이드 버퍼와 데이터변환 록어사이드 버퍼를 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 14**

제13항에 있어서, 상기 캐시 메모리는 명령 캐시 메모리와 데이터 캐시 메모리를 포함하며, 각각의 뱅크 구조는 상기  $n$ 개의 프로세서로부터의 명령 어드레스와 데이터 어드레스를 유지하고;

상기 선택 수단은, 상기 명령 캐시 메모리와 상기 데이터 캐시 메모리에 각각 접속되어 상기 명령 캐시 메모리와 데이터 캐시메모리에 액세스하는 명령선택수단 및 데이터선택수단을 포함하는 것을 특징으로 하는 데이터처리시스템.

**청구항 15**

제14항에 있어서, 상기  $n$ 개의 프로세서를 병렬동작모드에서 활성화시키는 클럭을 포함하는 것으로 하는 데이터처리시스템.

**청구항 16**

$n$ 개의 프로세서( $n$ 은 1보다 큰 정수)에 의해 공유되는 주 메모리를 포함하는 데이터 처리시스템에 있어서의 데이터 처리방법으로서, 프로그램 카운터 레지스터를 사용하여 상기 주 메모리로부터 명령을 판독하는 단계;

상기  $n$ 개의 프로세서내의 복수의 연산장치에서 각각 병렬동작을 실행하도록, 상기  $n$ 개의 프로세서중 하나의 일개의 프로그램 카운터 어드레스에 의해 지시된 복수의  $n \times m$ 개 명령( $m$ 은 1보다 큰 정수)을 판독하고 상기  $m$ 개의 명령의 각각을 상기  $n$ 개의 프로세서에 공급하고 그 결과가 병렬동작모드에서 상기  $n$ 개의 프로세서 중 상기 하나의 레지스터 화일에 기록되도록 하는 단계;

상기  $n$ 개의 프로세서내에서 각각 병렬동작을 실행하도록, 상기  $n$ 개의 프로세서의 각각의 프로그램 카운터 레지스터에 의해 지시된 상기  $m$ 개의 명령을 판독하고 상기  $m$ 개의 명령의 각각을 상기  $n$ 개의 프로세서에 공급하고 그 결과가 멀티프로세서 동작모드에서 각각 상기  $n$ 개의 프로세서의 각각의 레지스터 화일 내에 기록이 발생하는 단계 ; 및

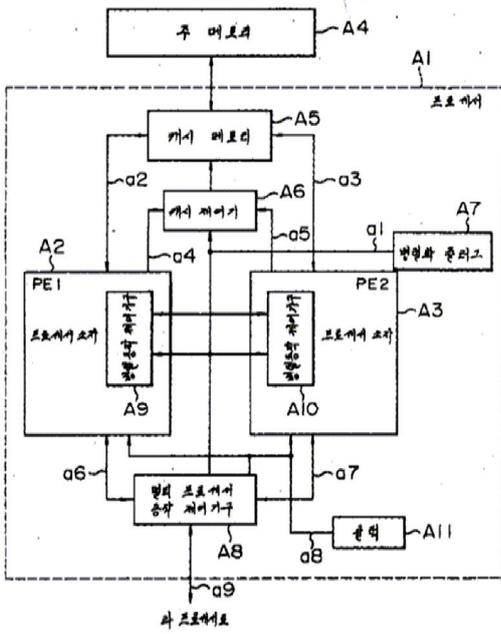
상기 병렬동작모드 또는 상기 멀티프로세서동작 제어수단 중 어느 하나에서 동작하도록 스위칭하는데 사용되는 플래그를 설정하는 단계를 포함하는 것을 특징으로 하는 데이터처리방법.

**청구항 17**

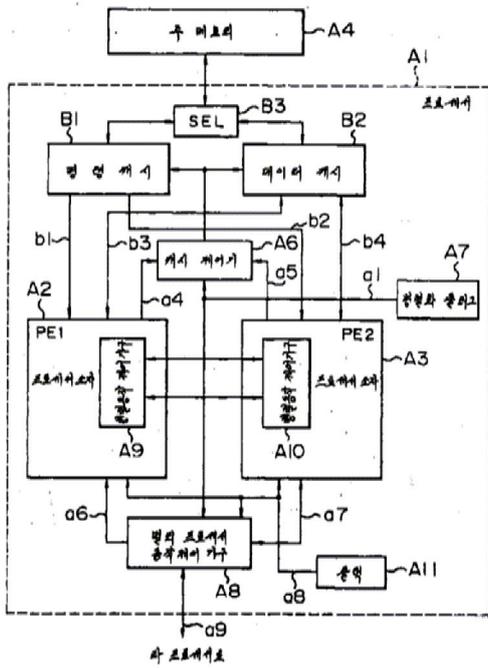
제16항에 있어서, 상기 병렬동작모드에서 상기  $n$ 개를 프로세서에 클럭을 보내는 단계를 더 포함하는 것을 특징으로 하는 데이터처리방법.

도면

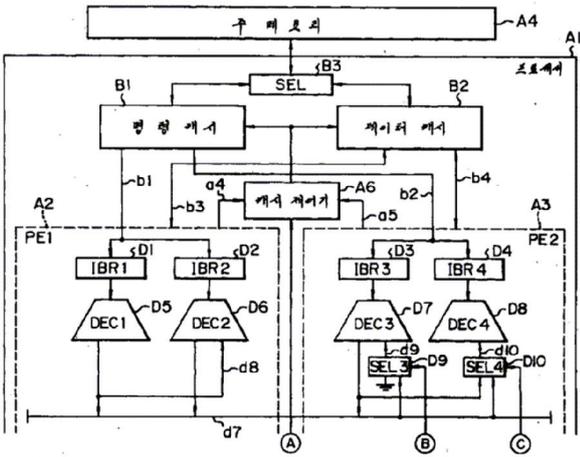
도면1



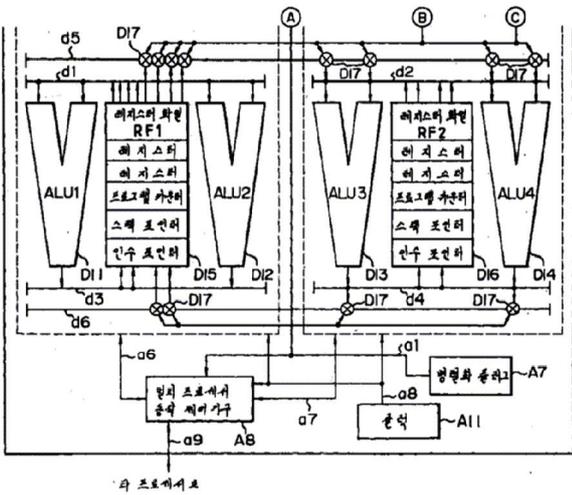
도면2



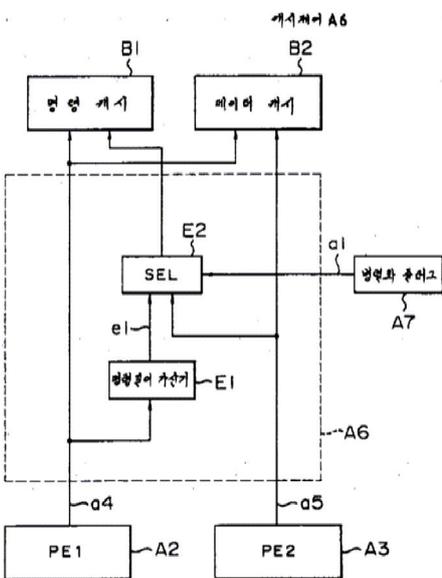
도면3a



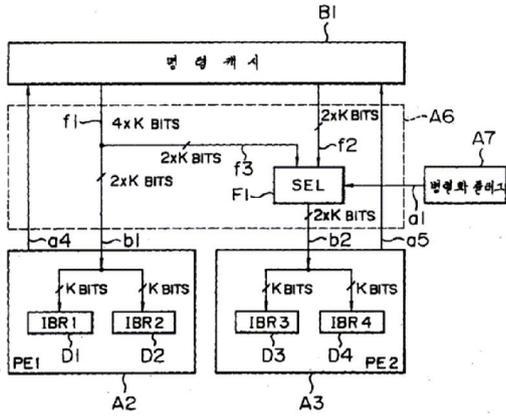
도면3b



도면4



도면5



도면6

멀티 프로세서 모드

분류	기능
인위입력	<ul style="list-style-type: none"> <li>마이크로 프로세서 ID 번호 지정</li> <li>전부</li> <li>어느 하나</li> </ul>
제어	<ul style="list-style-type: none"> <li>레지스터 전송</li> <li>대기</li> <li>캐시</li> </ul>
배와 제어	<ul style="list-style-type: none"> <li>프로세서 소자 사이의 '비교, 스왑 ('데스트, 서드')' 결합 제어</li> </ul>

도면7

