(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl. [7]
G06F 9/38

(45) 공고일자 2001년09월17일
(11) 등록번호 10-0291602
(24) 등록일자 2001년03월14일

(21) 출원번호 10-1997-0031357
(22) 출원일자 1997년07월07일

(65) 공개번호 특1998-0010764
(43) 공개일자 1998년04월30일

(30) 우선권주장 96-177773 1996년07월08일 일본(JP)

(73) 특허권자
가부시끼가이샤 히다치 세이사꾸쇼
가나이 쓰도무
일본국 도쿄도 지요다구 간다 스루가다이 4-6

(72) 발명자
다나카 가즈히코
일본국 가나가와켄 후지사와시 구게누마가이간 2-5-1
노지리 도오루
일본국 도쿄도 신쥬쿠구 니시신쥬쿠 3-4-4
고지마 게이지
일본국 가나가와켄 사가미하라시 미나미하시모토 1-20-11-405
니시오카 기요카즈
일본국 가나가와켄 오다와라시 나카무라하라 417-15
구로카와 요시키
일본국 후쿠이켄 사카이군 하루에쵸 에도메카미신마치 232

(74) 대리인
백남기

심사관 : 오흥수

(54) 인터럽트제어장치,프로세서및계산기시스템

요약

인터럽트 제어장치, 이 장치를 사용한 프로세서 및 계산기 시스템에 관한 것으로서, 명령길이가 일정하지 않은 경우에도 인터럽트처리를 정확하게 실행할 수 있도록 하기 위해서, 인터럽트가 발생한 경우에 이 인터럽트가 발생한 명령에서 해당 명령의 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 어드레스를 저장하는 어드레스 저장수단; 인터럽트 처리가 완료된 후에 어드레스 저장수단에 저장된 각 어드레스를 인터럽트가 발생한 명령의 어드레스에서 순차 리드하는 어드레스 리드수단; 어드레스 리드수단에 의해서 최초에 리드한 어드레스로 분기해서 명령을 실행하는 제1 명령실행수단 및 ; 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 해당 어드레스가 분기명령에 의해서 지정된 명령의 어드레스일 때에는 해당 어드레스로 분기해서 명령을 실행하고, 분기명령에 의해서 지정

된 명령 이외의 명령의 어드레스일 때에는 현재 실행되고 있는 명령의 후속명령을 실행하는 제2 명령실행수단을 구비하는 구성으로 하였다.

이렇게 하는 것에 의해, CISC나 압축후의 VLIW 등 명령길이가 일정하지 않은 명령을 실행하는 경우에도 인터럽트가 발생한 명령이 지연슬롯중의 명령인지의 여부에 관계없이 인터럽트를 정확하게 처리할 수 있는 파이프라인방식의 프로세서 및 계산기 시스템을 제공할 수 있다는 효과가 얻어진다.

대표도

도 1

명세서

도면의 간단한 설명

제1도는 본 발명의 제1실시예인 인터럽트 제어장치를 사용한 프로세서와 그의 주변장치를 기능적으로 도시한 도면.

제2도는 제1도에 도시한 명령공급유닛의 개략 구성도.

제3도는 제1도에 도시한 인터럽트 제어유닛의 개략 구성도.

제4도는 제1도에 도시한 프로세서에 의해 처리되는 지연슬롯을 갖는 분기명령을 포함한 프로그램의 1예를 도시한 도면.

제5도는 제1도에 도시한 프로세서가 제4도에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면.

제6도는 제1도에 도시한 프로세서에 의해 처리되는 인터럽트 처리프로그램의 1예를 도시한 도면.

제7도는 제4도에 도시한 프로그램의 명령R에서 F스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면.

제8도는 제4도에 도시한 프로그램의 명령R에서 R스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면.

제9도는 제4도에 도시한 프로그램의 명령P에서 F스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면.

제10도는 제4도에 도시한 프로그램의 명령P에서 R스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면,

제11도는 본 발명의 제2실시예인 인터럽트 제어장치를 사용한 프로세서와 그의 주변장치를 기능적으로 도시한 도면.

제12도는 제11도에 도시한 자동복귀유닛의 개략 구성도.

제13도는 제11도에 도시한 프로세서에 의해 처리되는 인터럽트 처리프로그램의 1예를 도시한 도면.

제14도는 제4도에 도시한 프로그램의 명령R에서 F스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면.

제15도는 제4도에 도시한 프로그램의 명령R에서 R스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면.

제16도는 제4도에 도시한 프로그램의 명령P에서 F스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면.

제17도는 제4도에 도시한 프로그램의 명령P에서 R스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면.

제18도는 본 발명의 제3실시예에 사용하는 명령공급유닛의 개략 구성도.

제19도는 본 발명의 제3실시예에 사용하는 인터럽트 제어유닛의 개략 구성도.

제20도는 본 발명의 제3실시예에 사용하는 프로세서에 의해 처리되는 인터럽트 프로그램의 1예를 도시한 도면.

제21도는 본 발명의 제1실시예에서 사용한 명령공급유닛의 변형예를 도시한 도면.

제22도는 2개의 명령을 지연슬롯내에 포함하는 지연분기를 실행하는 파이프라인방식의 프로세서와 그의 주변장치를 기능적으로 도시한 도면.

제23도는 제22도에 도시한 명령공급유닛(20c)의 개략 구성도.

제24도는 제22도에 도시한 인터럽트 제어유닛(50b)의 개략 구성도.

제25도는 제22도에 도시한 프로세서가 제4도에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면.

제26도는 제22도에 도시한 프로세서에 의해 처리되는 인터럽트 처리프로그램의 1예를 도시한 도면.

제27도는 제1도에 도시된 프로세서 및 그의 주변장치를 사용한 시스템 구성예를 도시한 도면.

제28도는 연산처리장치가 실행하는 프로그램의 1예를 도시한 도면.

제29도는 파이프라인방식이 채용되고 있지 않은 연산처리장치가 제28도에 도시된 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면.

제30도는 파이프라인방식이 채용되고 있는 연산처리장치가 제28도에 도시된 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면.

제31도는 분기명령을 포함한 프로그램의 1예를 도시한 도면.

제32도는 파이프라인방식이 채용되고 있는 연산처리장치가 제31도에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면.

제33도는 지연슬롯을 갖는 분기명령을 포함한 프로그램의 1예를 도시한 도면.

제34도는 지연분기에 대응한 파이프라인방식의 연산처리장치가 제31도에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 지연슬롯을 갖는 분기명령을 실행하는 파이프라인방식의 연산처리장치에 있어서 인터럽트가 발생했을 때의 제어장치 및 이 장치를 사용한 프로세서, 계산기 시스템에 관한 것이다.

연산처리장치의 처리속도를 향상시키는 방법으로서, 처리의 파이프라인화가 있다. 이하, 연산처리장치의 파이프라인화에 대해서 도면을 참조해서 설명한다.

도 28은 연산처리장치가 실행하는 프로그램의 1예를 도시한 도면, 도 29는 파이프라인방식이 채용되고 있지 않은 연산처리장치가 도 28에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면, 도 30은 파이프라인방식이 채용되고 있는 연산처리장치가 도 28에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면이다.

또한, 통상 연산처리장치는 도 29 및 도 30에 도시한 바와 같이, 파이프라인방식이 채용되고 있는지의 여부에 관계없이 각 명령을 리드하는 F스테이지, 리드한 명령을 디코드해서 오퍼런드를 리드하는 R스테이지, 연산을 실행하는 E스테이지, 그리고 연산결과를 라이트하는 W스테이지의 4스테이지로 나누어서 실행한다. 그리고, 각 스테이지의 실행에는 1클럭 상당의 시간을 필요로 한다. 따라서, 1개의 명령을 실행하기 위해서는 4클럭 상당의 시간이 필요하게 된다.

파이프라인방식이 채용되고 있지 않은 연산처리장치에서는 도 29에 도시한 바와 같이, 임의의 명령의 전체스테이지의 실행이 종료한 후 다음의 명령을 실행한다. 따라서, 프로그램이 n개의 명령으로 구성되어 있는 경우, 상기 프로그램의 처리에는 4n프로그램 상당의 시간이 필요하다. 이에 반해, 파이프라인방식이 채용되고 있는 연산처리장치에서는 도 30에 도시한 바와 같이, 임의의 명령의 F스테이지의 실행이 종료한 후 이 명령의 R스테이지의 실행개시와 동시에 다음 명령의 실행을 개시한다. 따라서, 프로그램이 n개의 명령으로 구성되어 있는 경우, 이 프로그램의 처리는 n+3클럭 상당의 시간이면 좋다. 이와 같이, 파이프라인방식의 연산처리장치에서는 여러개의 명령을 오버랩해서 실행하므로 프로그램의 처리속도를 향상시킬 수 있다.

그런데, 파이프라인방식의 연산처리장치에서는 프로그램중에 분기명령이 있으면 프로그램 처리성능이 저하한다는 문제가 있다. 다음에, 이 문제를 도면을 참조해서 설명한다. 도 31은 분기명령을 포함한 프로그램의 1예를 도시한 도면이다. 여기서, JMP는 분기를 나타내는 커맨드이다. 명령Q는 명령X, Y로 이루어지는 루틴L1으로 분기할 것을 명령하고 있다.

도 32는 파이프라인방식이 채용되고 있는 연산처리장치가 도 31에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면이다. 통상, 연산처리장치는 명령을 디코드하여야 즉 R스테이지를 실행하여야 비로서 이 명령이 분기명령인지 아닌지를 인식할 수 있다. 바꿔 말하면, R스테이지로 되지 않으면, 상기 명령이 분기명령인지 아닌지를 판단할 수 없다. 이 때문에, 파이프라인방식의 연산처리장치에서는 도 32에 도시한 바와 같이, 명령Q의 다음에는 명령Q가 지정하는 분기지L1의 명령X가 실행되어야 함에도 불구하고 명령R이 실행되어 버린다. 그리고, 명령Q의 R스테이지가 실행되어 상기 명령이 분기명령인 것이 판명된 시점에서 명령R의 실행이 중지된다. 이와 같이, 파이프라인방식의 연산처리장치에서는 분기명령의 다음 명령이 잘못해서 실행되어 버리기 때문에, 불필요한 사이클이 발생하여 프로그램 처리성능이 저하한다.

상기 문제를 해결하는 방법으로서 지연분기가 있다. 지연분기라고 하는 것은 분기가 발생했을 때 분기명령의 후속명령을 몇개 실행하고나서 분기지의 명령을 실행하는 방법이다. 일반적으로, 분기명령의 다음에 실행되는 후속명령을 지연슬롯중의 명령이라 한다. 다음에, 지연분기에 대해서 도면을 참조해서 설명한다. 도 33은 지연슬롯을 갖는 분기명령을 포함한 프로그램의 1예를 도시한 도면, 도 34는 분기명령에 대응한 파이프라인방식의 연산처리장치가 도 31에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면이다. 또한, 여기서는 지연슬롯에 포함되는 명령이 1개인 경우에 대해서 설명한다.

지연분기에 대응한 파이프라인방식의 연산처리장치에서는 도 34에 도시한 바와 같이 명령Q의 R스테이지가 실행되어 이 명령이 분기명령인 것이 판명되었음에도 불구하고, 분기명령의 후속명령인 명령P의 실행을 중지하지 않고 실행하고 있다. 그리고, 명령P 다음에 명령Q가 지정하는 분기지인 루틴L1의 명령X를 실행하고 있다. 이와 같이, 지연분기에 대응한 파이프라인방식의 연산처리장치에서는 분기지의 명령을 실행할 때에 실행중인 분기명령의 후속명령을 중지하지 않으므로 불필요한 사이클이 발생하지 않는다.

그러나, 지연분기에 대응한 파이프라인방식의 연산처리장치에서는 지연슬롯에 포함되는 명령의 실행중에 인터럽트가 발생한 경우, 인터럽트 처리프로그램으로부터의 복귀법에 연구가 필요하다. 즉, 통상의 연산처리장치에서 사용되고 있는 바와 같은 인터럽트가 발생한 명령의 어드레스값만을 퇴피시키는 방법에서는 도 34에 있어서 지연슬롯중의 명령P로 인터럽트가 발생한 경우, 명령P의 어드레스만이 퇴피되게 된다. 이 경우, 명령Q에 지정된 분기지L1의 명령X의 어드레스가 퇴피되지 않으므로, 인터럽트 처리프로그램에서 되돌아 온 후, 명령P, 명령X, 명령Y의 순번으로 실행해야 할 것을 명령P, 명령R, 명령S의 순서로 실행해 버린다.

상기 문제를 해결하는 장치로서 일본국 특허공개공보 평성5-143363호에 기재된 인터럽트 처리장치가 있다. 이 장치에서는 인터럽트 처리프로그램으로부터의 복귀지의 어드레스를 유지하는 제1 어드레스 유지수단과 인터럽트 처리프로그램으로부터의 복귀지 어드레스의 명령의 다음에 실행할 명령의 어드레스를 유지하는 제2 어드레스 유지수단을 마련하고, 인터럽트 처리프로그램에서 되돌아올(복귀할) 때에는 제1 어드레스 유지수단에 유지되어 있는 어드레스로 분기하고, 이 명령의 실행후에는 제2 어드레스 유지수단에 유지되어 있는 어드레스로 분기하도록 한 것이다.

이와 같이 하는 것에 의해, 지연슬롯중의 명령이 1개인 경우에 지연슬롯 중의 명령에서 인터럽트가 발생했을 때에도 인터럽트 처리프로그램에서 복귀한 후 적정한 순번으로 명령을 실행할 수 있도록 하고 있다.

발명이 이루고자 하는 기술적 과제

그런데, 일본국 특허공개 공보 평성5-143363호에 기재된 장치에서는 연산처리장치로서 명령길이가 일정한 RISC(Reduced Instruction Set Computer)를 사용하는 것을 전제로 하고 있다. 즉, 지연슬롯중의 명령 이외의 명령에서 인터럽트가 발생한 경우, 명령길이가 일정하다는 것을 이용해서 제1 어드레스 유지수단에 퇴피시킨 인터럽트가 발생한 명령의 어드레스에 소정의 명령길이를 가산하는 것에 의해 제2 어드레스 유지수단에 퇴피시킬 어드레스를 산출하고 있다.

그러나, 명령길이가 일정하지 않은 CISC(Complex Instruction Set Computer)에서는 명령을 리드하는 F스테이지가 실행되어야 비로서 상기 명령의 명령길이가 판별된다. 이 때문에, 지연슬롯중의 명령 이외의 명령에서 인터럽트가 발생한 경우, 인터럽트가 이 명령의 F스테이지 실행완료전에 발생했을 때에는 상기 명령의 명령길이가 판별되어 있지 않은 경우가 있다. 이 경우, 상기 일본국 특허공개공보 평성5-143363호에 기재된 장치에서는 제2 어드레스 유지수단에 퇴피시킬 어드레스를 적정하게 산출할 수 없다는 문제가 있다.

특히, 여러개의 처리를 동시에 실행하는 차세대의 연산처리장치를 위한 장치(시스템)으로서 주목받고 있는 VLIW(Very Long Instruction Word)에서는 명령길이는 일정하지만 동시에 실행시키고자 하는 여러개의 처리를 1개의 명령으로 결합하기 때문에 명령의 용장성이 나쁘다. 이 때문에, 명령을 압축해서 메모리에 저장하는 것이 일반적이다. 그러나, 명령을 압축한 경우에는 압축후의 명령의 명령길이가 일정하지 않게 되는 경우가 있으므로, 상기 일본국 특허공개공보 평성5-143363호에 기재된 장치에서는 VLIW에 적용한 경우에도 제2 어드레스 유지수단에 퇴피시킬 어드레스를 적정하게 산출할 수 없는 경우가 있다는 문제가 있다.

본 발명의 목적은 상기 사정에 따라서 이루어지는 것으로서, 명령길이가 일정하지 않은 경우에도 인터럽트처리를 정확하게 실행할 수 있는 인터럽트 제어장치 및 이 장치를 사용한 프로세서, 계산기 시스템을 제공하는 것이다.

발명의 구성 및 작용

상기 과제를 해결하기 위해, 본 발명의 인터럽트 제어장치는 적어도 1개의 명령을 지연슬롯중에 포함하는 분기명령을 실행하는 파이프라인방식의 연산처리장치에 있어서 기억장치에 가변길이로 저장된 상기 분기명령을 포함하는 여러개의 명령을 순차 리드해서 처리할 때에 사용하는 인터럽트 제어장치로서, 인터럽트가 발생한 경우에 이 인터럽트가 발생한 명령에서 이 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 어드레스를 저장하는 어드레스 저장수단; 상기 인터럽트 처리가 완료된 후에 상기 어드레스 저장수단에 저장된 각 어드레스를 상기 인터럽트가 발생한 명령의 어드레스에서 순차 리드하는 어드레스 리드수단; 상기 어드레스 리드수단에 의해서 최초에 리드한 어드레스로 분기하여 명령을 실행하는 제1 명령실행수단 및; 상기 어드레스 리드수단에 의해서 2번째 이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스로 분기해서 명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 현재 실행되고 있는 명령의 후속명령을 실행하는 제2 명령실행수단을 구비하고 있는 것을 특징으로 한다.

여기서, 실행하고 있는 명령의 어드레스에 상기 명령의 명령길이를 가산해서 다음에 리드하여 실행할 명령의 어드레스를 생성하는 어드레스 생성수단을 구비하고 있는 경우, 상기 제2 명령실행수단은 상기 인터럽트가 발생한 시점에서 이 인터럽트가 발생한 명령의 명령길이가 판명되어 있는 경우에는 상기 어드레스 리드수단에 의해 2번째로 리드한 어드레스가 상기 분기명령에 의해서 지정된 명령이외의 명령의 어드레스이더라도 상기 어드레스로 분기해서 명령을 실행하도록 해도 좋다.

또, 본 발명의 프로세서는 명령저장용 메모리에 가변길이로 저장된 분기명령을 포함하는 여러개의 명령을 순차 리드해서 실행하는 파이프라인방식의 프로세서로서, 상기 명령저장용 메모리에서 명령을 순차 리드함과 동시에 상기 리드한 명령이 분기명령인 경우 이 분기명령의 지연슬롯에 포함되는 후속명령을 리드한 후에 상기 분기명령에 의해서 지정되는 분기지 어드레스의 명령을 리드하는 명령 공급유닛; 상기 명령공급유닛에 의해서 리드된 명령을 실행함과 동시에 상기 리드된 명령이 분기명령인 경우에 이 분기명령에 의해서 지정되는 분기지 어드레스를 상기 명령공급유닛에 통지하는 명령실행유닛 및; 인터럽트가 발생한 경우에 인터럽트처리로부터의 복귀후에 실행할 명령을 제어하는 인터럽트 제어유닛을 구비하고, 상기 인터럽트 제어유닛은 인터럽트가 발생한 경우에 이 인터럽트가 발생한 명령에서 이 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 어드레스를 저장하는 레지스터를 갖고, 상기 인터럽트처리가 완료된 후에 상기 레지스터에 저장된 각 어드레스를 상기 인터럽트가 발생한 명령의 어드레스에서 순차 리드하는 것이고, 상기 명령공급유닛은 인터럽트 처리로부터의 복귀시에 상기 인터럽트 제어유닛에 의해서 상기 레지스터에서 최초에 리드한 어드레스의 명령을 리드함과 동시에 상기 인터럽트 제어유닛에 의해서 상기 레지스터에서 2번째이후에 리드한 어드레스에 대해서는 이 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 이 어드레스로 분기해서 명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 최신에 리드한 명령의 후속명령을 리드하는 것인 것을 특징으로 한다.

또, 본 발명의 계산기 시스템은 본 발명의 프로세서; 분기명령을 포함하는 여러개의 명령이 가변길이로 저장된 명령저장용 메모리 및; 상기 프로세서를 입출력장치에 접속하는 I/O디바이스를 구비한 것을 특징으로 한다.

다음에, 본 발명의 제1 실시예에 대해서 도면을 참조해서 설명한다.

도 1은 본 발명의 제1 실시예인 인터럽트 제어장치를 사용한 프로세서와 그의 주변장치를 기능적으로 도시한 도면이다.

본 실시예인 인터럽트 제어장치를 사용한 프로세서는 분기명령을 포함하는 여러개의 명령으로 이루어지는 프로그램을 처리할 때에 1개의 명령을 지연슬롯중에 포함하는 지연분기를 실행하는 파이프라인방식의 프로세서이다.

본 실시예의 인터럽트 제어장치를 사용한 프로세서는 실행할 각 명령을 명령을 리드하는 F스테이지, 리드한 명령을 디코드해서 오퍼런드를 리드하는 R스테이지, 연산을 실행하는 E스테이지, 그리고 연산결과를 라이트하는 W스테이지의 4스테이지로 나누어서 실행하고 있다. 그리고, 각 스테이지의 실행에 1클럭 상당의 시간을 필요로 하고 있다. 따라서, 1개의 명령을 실행하기 위해서는 4클럭 상당의 시간이 필요하다.

본 실시예의 인터럽트 제어장치를 사용한 프로세서와 그의 주변장치는 도 1에 도시한 바와 같이, 명령저장용 메모리(10), 명령공급유닛(20), 명령실행유닛(30), 데이타 저장용 메모리(40)을 구비하고 있다. 여기서, 명령공급유닛(20), 명령실행유닛(30) 및 인터럽트 제어유닛(50)이 프로세서에 상당한다. 명령저장용 메모리(10) 및 데이타 저장용 메모리(40)은 주변장치이다.

명령저장용 메모리(10)은 여러개의 명령으로 구성된 소정의 프로그램이나 인터럽트처리를 실행하기 위한 인터럽트 처리프로그램 등을 저장하고 있다. 명령저장용 메모리(10)은 어드레스선(800)을 거쳐서 통지된 물리어드레스에 저장되어 있는 명령을 리드하고, 데이타선(801)을 사용해서 명령공급유닛(20)으로 전송한다.

데이타 저장용 메모리(40)은 명령공급유닛(20)에서의 어드레스변환(가상어드레스를 물리어드레스로 변환)에 사용하는 어드레스 변환테이블(41) 이외에도 명령실행유닛(30)에서의 연산처리에 사용하는 데이타 등을 저장하고 있다. 데이타 저장용 메모리(40)은 제어선(822)를 거쳐 리드가 통지된 경우, 어드레스선(820)을 거쳐서 전송된 물리어드레스에 저장되어 있는 데이타를 리드하고, 데이타선(821)을 사용해서 명령실행유닛(30)으로 전송한다. 또, 제어선(822)를 거쳐서 라이트가 통지된 경우, 데이타선(821)을 거쳐서 명령실행유닛(30)에서 전송된 데이타를 어드레스선(820)을 거쳐서 전송된 물리어드레스에 라이트한다.

명령공급유닛(20)은 실행할 명령의 가상어드레스를 생성한 후, 이 가상어드레스를 물리어드레스로 어드레스변환하고, 이 물리어드레스를 어드레스선(800)을 사용해서 명령저장용 메모리(10)으로 전송한다. 이것에 의해, 명령저장용 메모리(10)에서 실행할 명령을 리드한다. 또, 데이타선(801)에 의해서 명령저장용 메모리(10)에서 전송된 명령의 명령길이를 검출함과 동시에 필요에 따라서 소정의 처리를 실행하고, 그 후 상기 명령을 데이타선(810)을 사용해서 명령실행유닛(30)으로 전송한다. 또한, 이들 처리는 F스테이지에서 실행된다.

또, 명령공급유닛(20)은 어드레스변환 동작에 지장이 생겨 인터럽트가 발생한 경우, 이 유닛(20)에서 인터럽트가 발생한 사실을 명령공급 인터럽트 신호선(830)을 사용해서 인터럽트 제어유닛(50)에 통지한다. 또한, 본 실시예에서는 명령공급유닛(20)에서 발생한 인터럽트는 모두 F스테이지에서 발생하는 것으로 한다.

또, 명령공급유닛(20)은 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50)에서 인터럽트요구를 받은 경우, 인터럽트 처리프로그램 실행후의 복귀 가상어드레스 즉 인터럽트처리가 발생한 명령의 가상어드레스와 이 명령의 다음에 실행할 명령의 가상어드레스를 각각 어드레스선(831), (832)를 사용해서 인터럽트 제어유닛(50)으로 전송한다. 또, 어드레스 신호선(832)를 사용해서 전송한 가상어드레스 즉 인터럽트처리가 발생한 명령의 다음에 실행할 명령의 가상어드레스가 분기명령에 의해서 특정되는 분기지 어드레스인지 아닌지를 나타내는 신호를 신호선(833)을 사용해서 인터럽트 제어유닛(50)으로 전송한다.

명령실행유닛(30)은 데이타선(810)을 거쳐서 명령공급유닛(20)에서 전송된 명령을 디코드하고, 그 결과에 따라서 연산처리, 분기처리, 데이타 저장용 메모리(40)으로의 액세스, 명령공급유닛(20) 및 인터럽트 제어유닛(50)의 내부레지

스터로의 액세스 등을 실행한다. 연산처리는 R스테이지에 있어서 연산용 레지스터(31)에서 필요한 데이타를 리드하고, E스테이지에서 연산을 실행하고, W스테이지에서 연산결과를 연산용 레지스터(31)에 라이트하는 것에 의해 실행된다. 분기처리는 명령공급유닛(20)에서 전송된 명령이 분기명령인 경우, R스테이지에 있어서 분기지 어드레스선(811) 및 분기요구 신호선(812)를 사용해서 이 분기명령에 따라 특정되는 분기지의 가상어드레스 및 분기요구를 명령공급유닛(20)으로 전송하는 것에 의해 실행된다. 데이타 저장용 메모리(40)으로의 액세스는 데이타 저장지의 물리어드레스를 전송하기 위한 어드레스선(820), 데이타를 전송하기 위한 데이타선(821) 및 액세스를 제어하기 위한 제어신호선(822)를 사용해서 데이타 저장용 메모리(40)으로 액세스한다. 명령공급유닛(20) 또는 인터럽트 제어유닛(50)의 내부레지스터로의 액세스는 레지스터의 식별신호를 전송하기 위한 식별신호선(813), 데이타를 전송하기 위한 데이타 신호선(814) 및 액세스를 제어하기 위한 제어신호선(815)를 사용해서 원하는 레지스터로 액세스한다.

또, 명령실행유닛(30)은 예를 들면 디코드한 명령이 정확한 명령이 아닌 경우, 이 유닛(30)에서 인터럽트가 발생한 사실을 명령실행 인터럽트신호선(835)를 사용해서 인터럽트 제어유닛(50)에 통지한다. 또, 인터럽트지 레지스터(32)에 미리 저장되어 있는 분기지의 가상어드레스로의 분기를 실행하는 것에 의해서, 명령저장용 메모리(10)에 저장되어 있는 인터럽트 처리프로그램의 처리를 개시한다.

또한, 본 실시예에서는 명령실행유닛(30)에 있어서의 인터럽트는 모두 R스테이지에서 발생하는 것으로 한다.

인터럽트 제어유닛(50)은 명령실행 인터럽트신호선(835) 또는 명령공급 인터럽트 신호선(830)에 의해서 전송되어 온 인터럽트통지를 받고 우선순위나 인터럽트 마스크 등의 체크를 실행하고, 그 후 인터럽트 신호선(834)를 사용해서 인터럽트요구와 인터럽트가 어느 스테이지에서 발생했는지를 나타내는 신호를 명령공급유닛(20) 및 명령실행유닛(30)으로 출력한다.

다음에, 도 1에 도시한 명령공급유닛(20)에 대해서 구체적으로 설명한다.

도 2는 도 1에 도시한 명령공급유닛(20)의 개략 구성도이다.

명령공급유닛(20)은 도 2에 도시한 바와 같이, 명령변환유닛(201), 레지스터(202)~(204), 가산기(205), 셀렉터(206)~(209) 및 어드레스 변환회로(210)을 구비하고 있다.

명령변환유닛(201)은 데이타선(801)에 의해서 명령저장용 메모리(10)에서 전송된 명령을 받고, 이 명령에 명령신장 등의 변환처리를 실행한다. 그리고, 이 변환처리가 실행된 명령을 데이타선(810)을 거쳐서 명령실행유닛(30)으로 전송한다. 또, 수취한 명령의 명령길이를 검출하고, 검출한 명령길이를 신호선(710)을 거쳐서 가산기(205)로 전송한다.

레지스터(202)는 어드레스선(700)으로 출력된 가상어드레스를 유지함과 동시에 지금까지 유지하고 있던 가상어드레스를 어드레스선(701)로 출력한다. 이것에 의해, F스테이지 실행중인 명령 즉 명령공급유닛(20)에 의해서 리드 액세스되고 있는 명령의 가상어드레스를 유지한다. 레지스터(203)은 어드레스선(701)로 출력된 가상어드레스를 유지함과 동시에, 지금까지 유지하고 있던 가상어드레스를 어드레스선(702)로 출력한다. 이것에 의해, R스테이지 실행중인 명령 즉 명령실행유닛(30)에서 디코드하고 있는 명령의 가상어드레스를 유지한다. 레지스터(204)는 분기요구 신호선(812)상의 신호를 유지함과 동시에, 지금까지 유지하고 있던 신호를 신호선(721)로 출력한다. 이것에 의해, 신호선(812)로는 F스테이지 실행중인 명령의 1개전에 실행되고 있는 명령 즉 R스테이지 실행중인 명령이 분기명령인지 아닌지를 나타내는 신호가 출력되고, 신호선(721)로는 R스테이지 실행중인 명령의 1개전에 실행되고 있는 명령 즉 E스테이지 실행중인 명령이 분기명령인지 아닌지를 나타내는 신호가 출력된다.

가산기(205)는 신호선(710)에 의해서 명령변환유닛(201)에서 전송된 명령길이에 어드레스선(701)로 출력된 가상어드레스 즉 F스테이지의 실행이 완료된 명령의 가상어드레스를 가산한다. 이것에 의해, 분기가 발생되지 않은 경우에는 다음에 실행할 명령의 가상어드레스를 산출해서 어드레스선(720)으로 출력한다.

셀렉터(206)은 분기요구 신호선(812)에 의해서 명령실행유닛(30)에서 분기 요구가 전송되어 온 경우, 분기지 어드레스선(811)에 의해서 명령실행유닛(30)에서 전송된 분기지의 가상어드레스를 선택한다. 한편, 분기요구가 전송되지 않은 경우, 어드레스선(720)에 의해서 가산기(205)에서 전송된 가상어드레스를 선택한다. 그리고, 셀렉터(206)은 선택한 가상어드레스를 다음에 실행할 명령의 가상 어드레스로서 어드레스선(700)으로 출력한다.

어드레스 변환회로(210)은 셀렉터(206)에 의해서 어드레스선(700)상으로 출력된 다음에 실행할 명령의 가상어드레스를 받고, 이 가상어드레스를 물리어드레스로 변환한다. 그리고, 변환한 물리어드레스를 어드레스선(800)을 거쳐서 명령저장용 메모리(10)으로 전송한다. 이것에 의해, 명령저장용 메모리(10)은 상기 물리어드레스로 변환된 명령을 데이타선(801)을 거쳐서 명령공급유닛(20)의 명령변환유닛(201)로 전송한다.

또한, 가상어드레스에서 물리어드레스로의 어드레스변환은 어드레스 변환회로(210)이 구비하는 어드레스 변환캐시(211)을 참조해서 실행한다. 어드레스 변환캐시(211)은 어드레스변환을 고속으로 실행하기 위한 것으로서, 데이타 저장용 메모리(40)에 저장된 어드레스 변환테이블(41)의 일부분의 카피(copy)를 저장하고 있다. 어드레스 변환회로(210)은 어드레스 변환에 필요한 값이 어드레스 변환캐시(211)에 의존하지 않는 경우(이 경우, 가상어드레스는 잘못된 물리어드레스로 변환된다), 원인으로 되는 명령이 F스테이지를 완료했을 때 명령공급 인터럽트신호선(830)을 사용해서 인터럽트가 발생한 사실을 인터럽트 제어유닛(50)에 통지한다.

셀렉터(207)은 인터럽트가 발생한 경우에 인터럽트 처리프로그램에서 되돌아 왔을 때 최초에 실행할 명령의 가상어드레스 즉 상기 인터럽트가 발생한 명령의 가상어드레스를 어드레스선(831)을 거쳐서 인터럽트 제어유닛(50)으로 전송한다. 구체적으로는, 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50)에서 F스테이지에 인터럽트가 발생한 사실이 통지된 경우, 어드레스선(701)상의 가상어드레스 즉 F스테이지를 실행하고 있던 명령(상기 인터럽트가 발생한 명령)의 가상어드레스를 선택해서 어드레스선(831)로 출력한다. 한편, 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50)에서 R스테이지에 인터럽트가 발생한 사실이 통지된 경우, 어드레스선(702)상의 가상어드레스 즉 R스테이지를 실행하고 있던 명령(상기 인터럽트가 발생한 명령)의 가상어드레스를 선택해서 어드레스선(831)로 출력한다.

셀렉터(208)은 인터럽트가 발생한 경우에 인터럽트 처리프로그램에서 되돌아 왔을 때 최초에 실행할 명령의 다음에 실행할 명령의 가상어드레스를 어드레스선(832)를 거쳐서 인터럽트 제어유닛(50)으로 전송한다. 구체적으로는, 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50)에서 F스테이지에 인터럽트가 발생한 사실이 통지된 경우, 어드레스선(700)상의 가상어드레스 즉 F스테이지를 실행하고 있던 명령(상기 인터럽트가 발생한 명령)의 다음에 실행할 명령의 가상어드레스를 선택해서 어드레스선(832)로 출력한다. 한편, 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50)에서 R스테이지에 인터럽트가 발생한 사실이 통지된 경우, 어드레스선(701)상의 가상어드레스 즉 R스테이지를 실행하고 있던 명령(상기 인터럽트가 발생한 명령)의 다음에 실행할 명령의 가상어드레스를 선택해서 어드레스선(832)로 출력한다.

셀렉터(209)는 셀렉터(208)에서 선택된 가상어드레스가 분기지의 가상어드레스인지 아닌지를 나타내는 신호를 출력한다. 구체적으로는, 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50)에서 F스테이지에 인터럽트가 발생한 사실이 통지된 경우, 분기요구 신호선(812)상의 신호를 선택해서 신호선(833)으로 출력한다. 한편, 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50)에서 R스테이지에 인터럽트가 발생한 사실이 통지된 경우, 신호선(721)상의 신호를 선택해서 신호선(833)으로 출력한다.

도 2에 도시한 명령공급유닛(20)에서는 셀렉터(206)이 분기지 어드레스선(811)에 의해서 명령실행유닛(30)에서 전송된 분기지의 가상어드레스를 선택한 경우, 셀렉터(208)에 의해 선택되는 가상어드레스는 상기 분기지의 가상어드레스로 된다. 분기지의 가상어드레스는 분기명령에 의해서 특정된 가상어드레스이므로, 이 경우 어드레스선(832)로 정확한 가상어드레스의 값이 출력된 것이 보증된다.

한편, 셀렉터(206)이 어드레스선(720)에 의해서 가산기(205)에서 전송되어 온 가상어드레스를 선택한 경우, 셀렉터(208)에 의해 선택되는 가상어드레스는 이 가산기(205)에 의해 산출된 가상어드레스로 된다. 어드레스 변환캐시(211)에 필요한 값이 저장되어 있지 않기 때문에 인터럽트가 발생한 경우, 변환된 물리 어드레스에 에러가 발생하므로 명령저장용 메모리(10)에서 본래 실행할 명령과는 다른 명령이 리드되어 버린다. 그리고, 명령변환유닛(201)에서 본래 실행할 명령의 명령길이와는 다른 명령길이가 검출되어 버린다. 이와 같은 경우, 가산기(205)에서 산출된 가상어드레스에 에러가 발생하므로, 어드레스선(832)로 부정확한 가상어드레스의 값이 출력되게 된다.

이와 같이, 셀렉터(208)에 의해 선택한 가상어드레스가 분기지의 가상어드레스인 경우에 이 선택한 가상어드레스는 정확한 값인 것이 보증되지만, 분기지의 가상어드레스가 아닌 경우에는 이 선택한 가상어드레스가 정확한 값이 아닐 가능성이 있다. 이 때문에, 셀렉터(209)는 본 실시예의 방식이 적용된 프로세서에 의해 처리되는 지연슬롯중에 포함되는 명령이 1개인 것을 이용하여 인터럽트가 발생한 명령의 1개전의 명령이 분기명령인지 아닌지에 따라서 셀렉터(208)에 의해 선택한 가상어드레스가 분기지의 가상어드레스인지 아닌지를 나타내는 정보를 출력하고 있다.

또한, 어드레스 변환캐시(211)에 필요한 값이 저장되어 있지 않기 때문에 인터럽트가 발생한 경우, 명령실행유닛(30)은 인터럽트지 레지스터(32)에 미리 저장되어 있는 분기지의 가상어드레스로의 분기를 실행하는 것에 의해 명령저장용 메모리(10)에 저장된 인터럽트 처리프로그램을 실행한다. 이것에 의해, 명령실행유닛(30)은 어드레스 변환테이블(41)에서 필요한 부분을 리드하고, 어드레스 신호선(813), 데이타 신호선(814), 제어신호선(815)를 사용해서 어드레스 변환캐시(211)의 내용을 갱신한다.

다음에, 도 1에 도시한 인터럽트 제어유닛(50)에 대해서 구체적으로 설명한다.

도 3은 도 1에 도시한 인터럽트 제어유닛(50)의 개략 구성도이다.

인터럽트 제어유닛(50)은 도 3에 도시한 바와 같이, 레지스터(501)~(503), 인터럽트 조정회로(504), 셀렉터(505), 쌍방향 드라이버(506)을 구비하고 있다.

인터럽트 조정회로(504)는 여러개의 인터럽트가 동시에 발생한 경우에 우선순위에 따라서 처리할 인터럽트를 결정하거나 또는 처리하지 않을(처리하지 말아야 할) 인터럽트를 마스크한다. 상술한 바와 같이, 본 실시예에서는 명령공급유닛(20)에서 발생하는 인터럽트(F스테이지에서 발생하는 인터럽트) 및 명령실행유닛(30)에서 발생하는 인터럽트(R스테이지에서 발생하는 인터럽트)의 2종류의 인터럽트를 상정하고 있다. 명령공급유닛(20)에서 발생하는 인터럽트의 요인은 어드레스 변환캐시(211)에 필요한 값이 저장되어 있지 않는 경우 등이고, 원인으로 되는 명령이 F스테이지까지 진행했을 때에 명령공급 인터럽트 신호선(830)을 거쳐서 명령공급유닛(20)에서 인터럽트 제어유닛(50)으로 인터럽트가 발생된 사실이 통지된다. 한편, 명령실행유닛(30)에서 발생하는 인터럽트 요인은 디코드된 명령이 부정확한 명령이었던 경우 등이고, 원인으로 되는 명령이 R스테이지까지 진행했을 때에 명령실행 인터럽트 신호선(835)를 거쳐서 명령실행유닛(30)에서 인터럽트 제어유닛(50)으로 인터럽트가 발생된 사실이 통지된다.

인터럽트 조정회로(504)는 레지스터(504a)에 저장된 데이타에 따라서 명령 공급 인터럽트 신호선(830) 및 명령실행 인터럽트 신호선(835)를 거쳐서 통지된 인터럽트에 대해서 우선순위나 마스크의 유무 등을 판정하고, 처리할 인터럽트를 인터럽트 신호선(834)를 거쳐서 명령공급유닛(20), 명령실행유닛(30) 및 레지스터(501)~(503)에 통지한다.

또, 인터럽트 조정회로(504)는 제어신호선(815)를 거쳐서 명령실행유닛(30)에서 전송된 제어신호가 리드액세스이고 또한 어드레스선(813)을 거쳐서 전송된 식별신호가 레지스터(504a)를 나타내는 경우, 신호선(733)을 거쳐서 셀렉터(505)로 레지스터(504a)에 저장된 데이타(우선순위나 마스크의 유무 등을 나타내는 데이타)를 전송한다.

또, 인터럽트 조정회로(504)는 제어신호선(815)를 거쳐서 명령실행유닛(30)에서 전송된 제어신호가 라이트액세스인 경우, 데이타선(814)상의 데이타를 레지스터(504a)에 저장한다. 또한, 본 실시예에서는 명령실행유닛(30)에서 레지스터(501)~(502)로의 라이트를 실행할 수 없도록 하고 있다. 따라서, 어드레스 신호선(813)은 라이트모드인 경우, 레지스터(504a)가 여러개의 레지스터로 구성되어 있는 경우에는 이들을 선택하기 위해서만 사용된다.

레지스터(501)은 인터럽트 조정회로(504)에서 인터럽트 신호선(834)로 인터럽트요구가 출력된 것을 계기로 해서, 어드레스선(831)을 거쳐서 명령공급유닛(20)의 셀렉터(207)에서 전송된 가상어드레스 즉 인터럽트 처리프로그램에서 되돌아 왔을 때 최초에 실행할 명령의 가상어드레스를 저장한다. 레지스터(502)는 인터럽트 조정회로(504)에서 인터럽트 신호선(834)로 인터럽트요구가 출력된 것을 계기로 해서, 어드레스선(832)를 거쳐서 명령공급유닛(20)의 셀렉터(208)에서 전송된 가상어드레스 즉 인터럽트 처리프로그램에서 되돌아 왔을 때 최초에 실행할 명령의 다음에 실행할 명령의 가상어드레스를 저장한다. 레지스터(503)은 인터럽트 조정회로(504)에서 인터럽트 신호선(834)로 인터럽트요구가 출력된 것을 계기로 해서, 신호선(833)을 거쳐서 명령공급유닛(20)의 셀렉터(209)에서 전송된 정보 즉 레지스터(502)에 저장되어 있는 가상어드레스가 분기지의 가상어드레스인지 아닌지를 나타내는 정보를 저장한다.

셀렉터(505)는 레지스터(501)~(503)에 저장된 값 및 인터럽트 조정회로(504)의 레지스터(504a)에 저장된 값중에서 어드레스선(813)을 거쳐서 명령실행유닛(30)에서 전송된 식별신호가 나타내는 레지스터를 선택하고, 이 레지스터에 저장되어 있는 값을 신호선(735)상으로 출력한다.

쌍방향 드라이버(506)은 제어신호선(815)를 거쳐서 명령실행유닛(30)에서 전송된 제어신호에 따라서 데이타 전송방향을 결정한다. 제어신호가 리드액세스인 경우, 어드레스선(735)를 거쳐서 셀렉터(505)에서 전송된 값을 데이타선(814)를 거쳐서 명령실행유닛(30)으로 전송한다. 또, 제어신호가 라이트액세스인 경우에는 데이타선(814)를 거쳐서 명령공급유닛(20)에서 전송된 데이타를 데이타선(736)을 거쳐서 인터럽트 조정회로(504)로 전송한다.

다음에, 도 1에 도시한 프로세서 및 그의 주변장치의 동작에 대해서 도면을 참조해서 설명한다.

우선, 인터럽트가 발생하지 않는 경우의 동작에 대해서 설명한다.

도 4는 도 1에 도시한 프로세서에 의해 처리되는 지연슬롯을 갖는 분기명령을 포함한 프로그램의 1예를 도시한 도면이다. 여기서, LD는 레지스터 사이에서의 데이타전송을 나타내는 커맨드, JMP는 분기를 나타내는 커맨드이다.

또한, 이 프로그램을 구성하는 각 명령은 명령저장용 메모리(10)에 저장된 상태에서는 명령길이가 일정하지 않은 것으로 한다.

도 5는 도 1에 도시한 프로세서가 도 4에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면이다.

도 1에 도시한 프로세서에서는 명령공급유닛(20)은 리드할 명령의 가상어드레스를 물리어드레스로 변환하고, 이 물리어드레스에 저장되어 있는 명령을 명령저장용 메모리(10)에서 리드한다. 그리고, 리드한 명령에 신장처리 등의 변환처리를 실시해서 명령실행유닛(30)으로 전송함과 동시에, 리드한 명령의 다음에 리드할 명령의 가상어드레스를 생성한다. 명령공급유닛(20)은 상기 일련의 동작(F스테이지)을 1클럭 상당의 시간(기간)에 반복해서 실행한다.

또, 명령실행유닛(30)은 명령공급유닛(20)에서 전송된 명령을 디코드해서 오퍼런드를 리드하고(R스테이지), 연산을 실행하고(E스테이지), 그리고 연산결과를 라이트한다(W스테이지). 이 동작을 각 스테이지마다 1클럭 상당의 기간에 걸쳐서 명령공급유닛(20)에서 명령이 전송되어 올 때마다 실행한다. 따라서, 도 1에 도시한 프로세서에서는 도 5에 도시한 바와 같이, 최대 3개의 명령을 평행하게(병행해서) 실행할 수 있다. 이것에 의해, 프로그램의 처리속도를 향상시킬 수 있다.

또, 명령실행유닛(30)은 명령공급유닛(20)에서 전송된 명령이 분기명령인 경우, R스테이지 완료후에 분기지 어드레스선(811)을 사용해서 상기 분기명령에 의해서 특정되는 분기지의 가상어드레스를, 그리고 분기요구 신호선(812)를 사용해서 분기요구를 명령공급유닛(20)으로 각각 전송하고 있다. 따라서, 분기지의 가상어드레스는 상기 분기명령의 실행개시후 2클럭 상당의 시간경과후에 명령공급유닛(20)으로 전송되게 된다. 이 때문에, 도 5에 도시한 바와 같이 분기명령인 명령Q가 실행개시된 후, 상기 분기명령의 후속명령인 명령P가 실행개시된 후에 명령Q의 분기지 루틴L1(명령X, 명령Y)이 실행개시된다. 즉, 1개의 명령을 포함하는 지연슬롯을 갖는 지연분기가 실행된다.

다음에, 인터럽트가 발생한 경우의 동작에 대해서 지연슬롯중의 명령 이외에서 인터럽트가 발생한 경우와 지연슬롯중의 명령에서 인터럽트가 발생한 경우로 나누어서 설명한다.

도 6은 도 1에 도시한 프로세서에 의해 처리되는 인터럽트 처리프로그램의 1예를 도시한 도면이다. 여기서, DI는 인터럽트 금지를 나타내는 커맨드, EI는 인터럽트 해제를 나타내는 커맨드, CMP는 2개의 데이타를 비교하는 커맨드, JNE는 조건부 분기를 나타내는 커맨드, NOP(논오퍼레이션)은 아무것도 하지 않는 것을 나타내는 커맨드이다.

지연슬롯중의 명령 이외에서 인터럽트가 발생한 경우에 대해서 설명한다.

우선, F스테이지에서 명령에 인터럽트가 발생한 경우에 대해서 도 7을 참조해서 설명한다.

도 7은 도 4에 도시한 프로그램의 명령R에서 F스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면이다.

명령R의 F스테이지에서 인터럽트가 발생한 경우 즉 명령R이 명령공급유닛(20)에서 처리되고 있는 도중에 상기 명령R에 인터럽트가 발생한 경우, 명령공급유닛(20)은 명령R의 F스테이지의 완료와 동시에 명령공급 인터럽트신호선(830)을 사용해서 인터럽트가 발생한 사실을 인터럽트 제어유닛(50)에 통지한다.

인터럽트 제어유닛(50)은 상기 통지된 인터럽트의 우선순위나 마스크의 유무 등을 판정하고, 이 인터럽트가 처리해야 하는 것인 경우에 인터럽트 신호선(834)를 거쳐서 명령실행유닛(30)에 통지한다.

명령실행유닛(30)은 인터럽트 제어유닛(50)에서 인터럽트가 발생한 사실의 통지를 받으면, 데이타선(810)을 거쳐서 명령공급유닛(20)에서 전송된 명령R을 실행하지 않고 종료하며, 도 7에 도시한 바와 같이 명령R의 R스테이지 이후를 캔슬(cancel)한다. 또, 분기지 어드레스선(811)을 사용해서 인터럽트지 레지스터(32)에 미리 저장되어 있는 분기지의 가상어드레스를 명령공급유닛(20)으로 전송한다.

명령공급유닛(20)은 명령실행유닛(30)에서 전송된 분기지의 가상어드레스를 물리어드레스로 변환해서 명령저장용 메모리(10)의 상기 물리 어드레스에 저장되어 있는 명령을 리드한다. 이것에 의해, 인터럽트 처리프로그램의 선두어드레스의 명령으로의 분기가 실행되고, 도 6에 도시한 인터럽트 처리프로그램이 실행된다.

도 6에 도시한 인터럽트 처리프로그램에서는 우선 인터럽트처리 중에 또다른 인터럽트가 발생하지 않도록 인터럽트 금지를 명령한다(명령a). 이 명령을 받아 명령실행유닛(30)은 인터럽트 제어유닛(50)의 인터럽트 조정회로(504)에 내

장된 레지스터(504a)를 「마스크있음」으로 리라이트한다. 인터럽트 제어유닛(50)은 레지스터(504a)의 내용이 마스크있음인 경우에는 그 후에 명령공급 인터럽트 신호선(830) 및 명령실행 인터럽트 신호선(835)를 거쳐서 인터럽트가 통지되더라도 인터럽트요구를 인터럽트 신호선(834)를 거쳐서 다른 유닛 등에 통지하는 일은 없다.

다음에, 인터럽트 제어유닛(50)의 레지스터(501)(E-IP0)에 저장되어 있는 가상어드레스(인터럽트가 발생한 명령 즉 명령R의 가상어드레스)를 명령실행유닛(30)의 연산용 레지스터(31)내에 마련되어 있는 레지스터IA에 저장할 것을 명령한다(명령b). 이 명령을 받아 명령실행유닛(30)은 식별신호선(813) 및 제어신호선(815)를 사용해서 레지스터(501)로 액세스한다. 이것에 의해, 인터럽트 제어유닛(50)은 레지스터(501)에 저장되어 있는 가상어드레스를 어드레스선(813)을 거쳐서 명령실행유닛(30)으로 전송한다. 명령실행유닛(30)은 인터럽트 제어유닛(50)에서 전송된 가상어드레스를 연산용 레지스터(31)내의 레지스터IA에 저장한다.

다음에, 인터럽트 제어유닛(50)의 레지스터(502)(E-IP1)에 저장되어 있는 가상어드레스(인터럽트가 발생한 명령의 다음에 실행될 명령 즉 명령S의 가상어드레스)를 명령실행유닛(30)의 연산용 레지스터(31)내에 마련되어 있는 레지스터IB에 저장할 것을 명령한다(명령c). 이 명령을 받아 명령실행유닛(30)은 식별신호선(813) 및 제어신호선(815)를 사용해서 레지스터(502)로 액세스한다. 이것에 의해, 인터럽트 제어유닛(50)은 레지스터(502)에 저장되어 있는 가상어드레스를 어드레스선(813)을 거쳐서 명령실행유닛(30)으로 전송한다. 명령실행유닛(30)은 인터럽트 제어유닛(50)에서 전송된 가상어드레스를 연산용 레지스터(31)내의 레지스터IB에 저장한다.

다음에, 인터럽트 제어유닛(50)의 레지스터(503)(E-IP1v)에 저장되어 있는 값(명령c에 의해 레지스터IB에 저장한 가상어드레스가 분기지의 가상어드레스인지 아닌지를 나타내는 값)을 명령실행유닛(30)의 연산용 레지스터(31)내에 마련되어 있는 레지스터IC에 저장할 것을 명령한다(명령d). 이 명령을 받아 명령실행유닛(30)은 식별신호선(813) 및 제어신호선(815)를 사용해서 레지스터(503)으로 액세스한다. 이것에 의해, 인터럽트 제어유닛(50)은 레지스터(503)에 저장되어 있는 값을 어드레스(813)을 거쳐서 명령실행유닛(30)으로 전송한다. 명령실행유닛(30)은 인터럽트 제어유닛(50)에서 전송된 값을 연산용 레지스터(31)내의 레지스터IC에 저장한다.

또한, 본 실시예에서는 레지스터(502)에 저장되어 있는 가상어드레스가 분기지의 가상어드레스인 경우, 레지스터(503)에 저장되는 신호가 「1」로 되도록 하고 있다. 한편, 레지스터(502)에 저장되어 있는 가상어드레스가 분기지의 가상어드레스가 아닌 경우, 레지스터(503)에 저장되는 신호가 「0」으로 되도록 하고 있다.

명령e~명령o의 1개전의 명령까지의 각 명령은 소정의 인터럽트 처리를 실행하기 위한 것이다. 이들 명령은 인터럽트의 요인에 따라서 각각 다르다. 이들 명령은 통상의 인터럽트 처리프로그램과 기본적으로 동일사상으로 만들어지는 것이다. 따라서, 여기서는 이들 명령의 설명은 생략한다.

명령o에서는 레지스터IC의 내용과 0의 비교를 명령한다(명령o). 이 명령을 받아 명령실행유닛(30)은 연산용 레지스터(31)내의 레지스터IC에 저장한 값, 즉 레지스터IB에 저장한 가상어드레스가 분기지의 가상어드레스인지 아닌지를 나타내는 값이 0인지 아닌지를 비교한다. 다음에, 명령o에서 비교한 결과, 양자가 일치한 경우에는 루틴L2로 분기할 것을 명령한다(명령p). 여기서는 도 4에 도시한 명령R에 인터럽트가 발생한 것을 상정하고 있으므로, 인터럽트가 발생한 명령R의 다음에 실행될 명령S의 가상어드레스는 분기명령에 의해서 특정되는 것은 아니다. 따라서, 레지스터IC에 저장한 값은 0으로 된다. 이 때문에, 명령실행유닛(30)은 도 7에 도시한 바와 같이, 루틴L2로의 분기를 실행하지 않고 후속명령q를 실행한다.

명령q에서는 명령a에 의해 명령된 인터럽트금지를 해제하도록 명령한다. 이 명령을 받아 명령실행유닛(30)은 인터럽트 제어유닛(50)의 인터럽트 조정회로(504)에 내장된 레지스터(504a)를 「마스크없음」으로 리라이트한다. 단, 레지스터(504a)의 리라이트는 명령q의 2개의 명령 후에 유효로 되도록 하고 있다.

다음에, 레지스터IA에 저장된 가상어드레스의 명령으로의 분기를 명령한다(명령r). 이 명령을 받아 명령실행유닛(30)은 레지스터IA에 저장된 가상어드레스의 명령 즉 인터럽트가 발생한 명령R을 실행한다. 단, 본 실시예에서는 상술한 바와 같이 1개의 명령을 포함하는 지연슬롯을 갖는 지연분기를 실행하므로, 명령R의 실행은 도 7에 도시한 바와 같이 명령r의 후속명령s의 실행후에 실행된다.

또한, 명령s에는 NOP라 불리는 아무것도 하지 않는 명령이 포함되어 있으므로, 도 7에 도시한 바와 같이 명령처리에서 복귀한 후에 실행순서를 변경하지 않고 도 4에 도시한 프로그램을 정확하게 처리할 수 있다.

명령R의 F스테이지에서 인터럽트가 발생한 경우, 이 단계에서는 명령R의 정확한 명령길이가 판명되지 않으므로, 명령R의 가상어드레스에 명령R의 명령길이를 가산해도 명령R의 후속명령인 명령S의 정확한 가상어드레스를 구할 수는 없다. 그래서, 본 실시예에서는 인터럽트처리로부터의 복귀후 최초에 실행할 명령(여기서는 명령R)의 다음에 실행할 명령(여기서는 명령S)이 분기명령의 분기지가 아닌 경우, 명령r에서 레지스터(501)에 저장된 가상어드레스로의 분기를 실행한 후, 명령r의 지연슬롯에 포함되는 명령s에 의해 레지스터(502)에 저장된 가상어드레스로의 분기를 실행하지 않고 NOP명령을 실행하고 있다. 이것에 의해, 인터럽트 처리에서 복귀한 후의 프로그램의 실행순서가 변경되는 것을 방지하고 있다.

다음에, R스테이지에서 명령에 인터럽트가 발생한 경우에 대해서 도 8을 참조해서 설명한다.

도 8은 도 4에 도시한 프로그램의 명령R에서 R스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면이다.

명령R의 R스테이지에서 인터럽트가 발생한 경우 즉 명령R이 명령실행유닛(30)에서 처리되고 있는 도중에 상기 명령R에 인터럽트가 발생한 경우, 명령실행유닛(30)은 명령R의 R스테이지의 완료와 동시에 명령실행 인터럽트 신호선(835)를 사용해서 인터럽트가 발생한 사실을 인터럽트 제어유닛(50)에 통지한다.

인터럽트 제어유닛(50)은 이 통지된 인터럽트의 우선순위나 마스크의 유무등을 판정하여 상기 인터럽트가 처리해야 할 것인 경우에 인터럽트 신호선(834)를 거쳐서 명령실행유닛(30)에 통지한다.

명령실행유닛(30)은 인터럽트 제어유닛(50)에서 인터럽트가 발생한 사실의 통지를 받으면, 명령R의 실행을 중지함과 동시에 데이타선(810)을 거쳐서 명령공급유닛(20)에서 전송된 명령S의 실행을 중지하고, 도 8에 도시한 바와 같이 명령R의 E스테이지 이후 및 명령S의 R스테이지 이후를 캔슬한다. 또, 내장된 인터럽트지 레지스터(32)에 미리 저장되어 있는 분기지의 가상어드레스를 명령공급유닛(20)으로 전송한다.

명령공급유닛(20)은 명령실행유닛(30)에서 전송된 분기지의 가상어드레스를 물리어드레스로 변환해서 명령저장용 메모리(10)의 상기 물리어드레스에 저장되어 있는 명령을 리드한다. 이것에 의해, 인터럽트 처리프로그램의 선두 어드레스의 명령으로의 분기가 실행되고, 상술한 명령R의 F스테이지에서 인터럽트가 발생한 경우와 마찬가지의 요령으로 도 6에 도시한 인터럽트 처리프로그램이 실행된다. 이것에 의해, 도 8에 도시한 바와 같이 인터럽트 처리에서 복귀한 후에 실행순서를 변경하지 않고 도 4에 도시한 프로그램을 정확하게 처리할 수 있다.

또한, 명령R의 R스테이지에서 인터럽트가 발생한 경우, 명령공급유닛(20)에서의 명령R에 대한 처리(F스테이지)는 이미 완료되어 있으므로, 명령공급유닛(20)의 명령변환유닛(201)은 명령R의 정확한 명령길이를 명령공급유닛(20)의

가산기(205)로 전송하고 있다. 이 때문에, 가산기(205)에 의해 산출된 명령S의 가상어드레스가 정확한 값으로 되므로, 인터럽트 제어유닛(50)의 레지스터(502)에는 명령S의 정확한 가상어드레스가 저장되게 된다. 따라서, 레지스터(502)에 저장된 가상어드레스를 분기지로 하는 명령을 실행하는 것에 의해서 명령S를 실행하는 것도 가능하다. 그러나, 여기서는 처리를 간략화하기 위해, 명령R의 F스테이지에서 인터럽트가 발생한 경우와 마찬가지의 요령으로 도 6에 도시한 인터럽트 처리프로그램을 실행하고 있다.

지연슬롯중의 명령에서 인터럽트가 발생한 경우에 대해서 인터럽트가 F스테이지에서 발생한 경우와 R스테이지에서 발생한 경우로 나누어서 설명한다.

우선, F스테이지에서 명령에 인터럽트가 발생한 경우에 대해서 도 9를 참조해서 설명한다.

도 9는 도 4에 도시한 프로그램의 명령P에서 F스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면이다.

분기명령인 명령Q의 지연슬롯에 포함되는 명령P의 F스테이지에서 인터럽트가 발생한 경우, 즉 명령P가 명령공급유닛(20)에서 처리되고 있는 도중에 상기 명령P에 인터럽트가 발생한 경우, 명령공급유닛(20)은 명령P의 F스테이지의 완료와 동시에 명령공급 인터럽트 신호선(830)을 사용해서 인터럽트가 발생한 사실을 인터럽트 제어유닛(50)에 통지한다.

인터럽트 제어유닛(50)은 상기 통지된 인터럽트의 우선순위나 마스크의 유무를 판정하고, 상기 인터럽트가 처리해야 할 것인 경우에 인터럽트 신호선(834)를 거쳐서 명령실행유닛(30)에 통지한다.

명령실행유닛(30)은 인터럽트 제어유닛(50)에서 인터럽트가 발생한 사실의 통지를 받으면, 데이타선(810)을 거쳐서 명령공급유닛(20)에서 전송된 명령P를 실행하지 않고 종료하며, 도 9에 도시한 바와 같이 명령P의 R스테이지 이후를 캔슬(취소)한다. 또, 내장된 인터럽트지 레지스터(32)에 미리 저장되어 있는 분기지의 가상어드레스를 명령공급유닛(20)으로 전송한다.

명령공급유닛(20)은 명령실행유닛(30)에서 전송된 분기지의 가상어드레스를 물리어드레스로 변환하고, 명령저장용 메모리(10)의 상기 물리어드레스에 저장되어 있는 명령을 리드한다. 이것에 의해, 인터럽트 처리프로그램의 선두어드레스의 명령으로의 분기가 실행되고, 도 6에 도시한 인터럽트 처리프로그램이 실행된다.

또한, 지연슬롯에 포함되는 명령P에서 인터럽트가 발생한 경우, 인터럽트 처리에서 복귀한 후 최초에 실행할 명령P의 다음에 실행할 명령X는 분기명령Q에 의해서 가상어드레스가 특정된 분기지의 명령이다. 이 때문에, 인터럽트 제어유닛(50)의 레지스터(503)에 저장되는 신호는 「1」로 된다. 따라서, 도 6에 도시한 명령d에 의해서 명령실행유닛(30)의 연산용 레지스터(31)내의 레지스터IC에 저장되는 값은 「1」로 되므로, 명령p에 의해 루틴L2로의 분기가 실행된다. 단, 본 실시예에서는 상술한 바와 같이, 1개의 명령을 포함하는 지연슬롯을 갖는 지연분기를 실행하므로 명령q를 실행한 후 명령t를 실행한다.

명령t에서는 레지스터IA에 저장된 가상어드레스의 명령으로의 분기를 명령한다. 이 명령을 받아 명령실행유닛(30)은 레지스터IA에 저장된 가상어드레스의 명령 즉 인터럽트가 발생한 명령P를 실행한다. 단, 본 실시예에서는 상술한 바와 같이 1개의 명령을 포함하는 지연슬롯을 갖는 지연분기를 실행하므로, 명령P의 실행은 도 9에 도시한 바와 같이 명령t의 후속명령u의 실행후에 실행된다.

명령u에서는 레지스터IB에 저장된 가상어드레스의 명령으로의 분기를 명령한다. 이 명령을 받아 명령실행유닛(30)은 레지스터IB에 저장된 가상어드레스의 명령 즉 분기명령Q에 따라서 가상어드레스가 특정된 분기지의 명령X를 실행한다. 단, 본 실시예에서는 상술한 바와 같이 1개의 명령을 포함하는 지연슬롯을 갖는 지연분기를 실행하므로, 명령X의 실행

은 도 9에 도시한 바와 같이 명령u의 다음에 실행되는 명령 즉 명령t에 의해 지정된 분기지의 명령P의 실행후에 실행된다. 따라서, 본 실시예에서는 도 9에 도시한 바와 같이, 인터럽트 처리에서 복귀한 후 실행순서를 변경하는 일없이 도 4에 도시한 프로그램을 정확하게 처리할 수 있다.

명령P에서 인터럽트가 발생한 경우, 명령P의 다음에 실행될 명령은 분기명령Q에 의해서 가상어드레스가 특정된 분기지의 명령X이다. 따라서, 상기 인터럽트가 발생한 단계에서 명령X의 정확한 가상어드레스가 판명되고 있다. 그래서, 본 실시에에서는 인터럽트처리로부터의 복귀후 최초에 실행할 명령(여기서는 명령P)의 다음에 실행할 명령(여기서는 명령X)이 분기명령(여기서는 명령Q)의 분기지인 경우, 명령t에 의해 레지스터(501)에 저장된 가상어드레스로의 분기를 실행한 후, 명령t의 지연슬롯에 포함되는 명령u에 의해 레지스터(502)에 저장된 가상어드레스로의 분기를 실행하고 있다. 이것에 의해, 인터럽트 처리에서 복귀한 후의 프로그램의 실행순서가 변경되는 것을 방지하고 있다.

다음에, R스테이지에서 명령에 인터럽트가 발생한 경우에 대해서 도 10을 참조해서 설명한다.

도 10은 도 4에 도시한 프로그램의 명령P에서 R스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면이다.

명령P의 R스테이지에서 인터럽트가 발생한 경우 즉 명령P가 명령실행유닛(30)에서 처리되고 있는 도중에 상기 명령P로 인터럽트가 발생한 경우, 명령실행유닛(30)은 명령P의 R스테이지의 완료와 동시에 명령실행 인터럽트 신호선(835)를 사용해서 인터럽트가 발생한 사실을 인터럽트 제어유닛(50)에 통지한다.

인터럽트 제어유닛(50)은 이 통지된 인터럽트의 우선순위나 마스크의 유무 등을 판정하여 상기 인터럽트가 처리해야 할 것인 경우에 인터럽트 신호선(834)를 거쳐서 명령실행유닛(30)에 통지한다.

명령실행유닛(30)은 인터럽트 제어유닛(50)에서 인터럽트가 발생한 사실의 통지를 받으면, 명령R의 실행을 중지함과 동시에 데이타선(810)을 거쳐서 명령공급유닛(20)에서 전송된 명령X의 실행을 중지하고, 도 10에 도시한 바와 같이 명령P의 E스테이지 이후 및 명령X의 R스테이지 이후를 캔슬한다. 또, 내장된 인터럽트지 레지스터(32)에 미리 저장되어 있는 분기지의 가상어드레스를 명령공급유닛(20)으로 전송한다.

명령공급유닛(20)은 명령실행유닛(30)에서 전송된 분기지의 가상어드레스를 물리어드레스로 변환하고, 명령저장용 메모리(10)의 상기 물리어드레스에 저장되어 있는 명령을 리드한다. 이것에 의해, 인터럽트 처리프로그램의 선두어드레스의 명령으로의 분기가 실행되고, 상술한 명령P의 F스테이지에서 인터럽트가 발생한 경우와 마찬가지의 요령으로 도 6에 도시한 인터럽트 처리프로그램이 실행된다. 이것에 의해, 도 10에 도시한 바와 같이 인터럽트 처리에서 복귀한 후 실행순서를 변경하는 일 없이 도 4에 도시한 프로그램을 정확하게 처리할 수 있다.

본 실시예에서는 인터럽트 처리로부터의 복귀후 최초에 실행할 명령의 다음에 실행할 명령이 분기명령의 분기지인 경우, 명령t에 의해 레지스터(501)에 저장된 가상어드레스로의 분기를 실행한 후, 명령t의 지연슬롯에 포함되는 명령u에 의해 레지스터(502)에 저장된 가상어드레스로의 분기를 실행하고 있다. 한편, 상기 명령이 분기명령의 분기지가 아닌 경우, 명령r에 의해 레지스터(501)에 저장된 가상어드레스의 분기를 실행한 후, 명령r의 지연슬롯에 포함되는 명령s에 의해 레지스터(502)에 저장된 가상어드레스로의 분기를 실행하는 일 없이 NOP명령을 실행하고 있다. 이와 같이 하는 것에 의해, 명령저장용 메모리에 저장된 프로그램의 각 명령의 명령길이가 일정하지 않은 경우에도, 인터럽트가 발생한 명령이 지연슬롯중의 명령인지의 여부에 관계없이 상기 프로그램을 소정의 순서로 정확하게 처리할 수 있다.

다음에, 본 발명의 제2 실시예에 대해서 도면을 참조해서 설명한다.

도 11은 본 발명의 제2 실시예인 인터럽트 제어장치를 사용한 프로세서와 그의 주변장치를 기능적으로 도시한 도면, 도 12는 도 11에 도시한 자동복귀유닛의 개략 구성도이다.

본 실시예가 제1 실시예의 것과 다른 점은 인터럽트처리에서 복귀할 때 프로그램을 소정의 순서로 실행하기 위한 처리(도 6에 도시한 인터럽트 처리프로그램의 명령b~명령d, 명령o, 명령p, 명령r~명령u에 상당하는 처리)를 소프트웨어적으로 실행하는 것은 아니라 도 11에 도시한 자동복귀유닛(33)에 의해 하드웨어적으로 실행하는 점이다. 또한, 도 11에 도시한 프로세서 및 그의 주변장치에 있어서 제1 실시예가 적용된 것과 동일기능을 갖는 것에는 동일부호 또는 대응하는 부호를 붙이고 그의 상세한 설명은 생략한다.

자동복귀유닛(33)은 도 12에 도시한 바와 같이, 자동복귀용 레지스터(34), 셀렉터(331), (332), 자동복귀용 레지스터 리드회로(333), 자동복귀용 처리회로(334)를 구비하고 있다.

자동복귀용 레지스터 리드회로(333)은 인터럽트 신호선(834)를 거쳐 인터럽트 제어유닛(50)에서 인터럽트요구를 받으면, 식별신호선(870) 및 제어신호선(872)를 각각 신호선(813), (815)와 접속하도록 신호선(874)를 거쳐서 셀렉터(331), (332)를 제어한다. 그리고, 인터럽트 제어유닛(50)의 레지스터(501)~레지스터(503)으로 순차 액세스하고, 레지스터(501)~레지스터(503)에 저장된 값을 자동복귀용 레지스터(34)에 저장한다. 또한, 셀렉터(331), (332)는 각각 식별신호선(813), 제어신호선(815)를 신호선(871), (873)에 접속하도록 제어되고 있는 것으로 한다.

자동복귀용 처리회로(334)는 신호선(875)에 개시신호가 입력되면, 자동복귀용 레지스터(34)로 액세스하고 레지스터(503)에서 전송된 값을 리드한다. 상기 값이 「0」인 경우, 레지스터(502)에서 전송된 값은 분기명령에 의해서 특정된 분기지의 명령의 가상어드레스는 아니다. 이 경우, 자동복귀용 처리회로(334)는 자동복귀용 레지스터(34)로부터 레지스터(501)에서 전송된 값(인터럽트가 발생한 명령의 가상어드레스)만을 리드하고, 신호선(876)을 거쳐서 분기지 어드레스선(811)상으로 출력하고, 명령공급유닛(20)으로 전송한다. 또, 분기요구를 신호선(877)을 거쳐서 분기요구신호선(812)상으로 출력하고, 명령공급유닛(20)으로 전송한다. 한편, 이 값이 「1」인 경우, 레지스터(502)에서 전송된 값은 분기명령에 의해서 특정된 분기지의 명령의 가상어드레스이다. 이 경우, 자동복귀용 처리회로(334)는 자동복귀용 레지스터(34)로부터 레지스터(501)에서 전송된 값(인터럽트가 발생한 명령의 가상어드레스) 및 레지스터(502)에서 전송된 값(분기지의 명령의 가상어드레스)을 순차 리드하고, 신호선(876)을 거쳐서 분기지 어드레스선(811)상으로 출력하고, 명령공급유닛(20)으로 전송한다. 또, 분기요구를 신호선(877)을 거쳐서 분기요구신호선(812)상으로 출력하고, 명령공급유닛(20)으로 전송한다.

다음에, 도 11에 도시한 프로세서 및 그의 주변장치에 있어서 인터럽트가 발생한 경우의 동작을 지연슬롯중의 명령 이외에서 인터럽트가 발생한 경우와 지연슬롯중의 명령에서 인터럽트가 발생한 경우로 나누어서 설명한다. 또한, 인터럽트가 발생하지 않는 경우의 동작은 제1 실시예의 것과 동일하므로 그의 상세한 설명은 생략한다.

도 13은 도 11에 도시한 프로세서에 의해 처리되는 인터럽트 처리프로그램의 1예를 도시한 도면이다. 여기서, IRET는 자동복귀유닛(33)을 기동하기 위한 커맨드이다.

우선, 지연슬롯중의 명령 이외의 명령에서 인터럽트가 발생한 경우에 대해서 설명한다.

도 14는 도 4에 도시한 프로그램의 명령R에서 F스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면, 도 15는 도 4에 도시한 프로그램의 명령R에서 R스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면이다.

본 실시예에서는 인터럽트 제어유닛(50)에서 인터럽트가 발생한 사실의 통지를 받으면, 자동복귀유닛(33)이 식별신호선(813), 데이타신호선(814) 및 제어신호선(815)를 사용해서 인터럽트 제어유닛(50)의 레지스터(501)~레지스터(503)으로 순차 액세스하고, 레지스터(501)~레지스터(503)에 저장된 값을 자동복귀용 레지스터(34)에 저장한다. 즉, 도 6에 도시한 명령b~명령d의 처리를 자동적으로 실행한다. 또, 명령공급유닛(20) 및 명령실행유닛(30a)에 의해 도 13에 도시한 인터럽트 처리프로그램이 실행된다.

도 13에 도시한 인터럽트 처리프로그램에서는 우선 명령a를 실행해서 인터럽트를 금지하고, 다음에 명령e 이후에 소정의 인터럽트를 실행하고, 그 후 명령w를 실행한다. 이것에 의해, 명령실행유닛(30a)는 신호선(875)를 거쳐서 자동복귀유닛(33)으로 개시신호를 출력한다. 이것을 받아 자동복귀유닛(33)은 인터럽트 제어유닛(50)의 레지스터(503)에서 자동복귀용 레지스터(34)로 전송된 값을 참조하여 분기를 실행한다. 여기에서는 인터럽트 제어유닛(50)의 레지스터(503)에서 자동복귀용 레지스터(34)로 전송된 값이 「0」으로 되므로, 인터럽트 제어유닛의 레지스터(502)에서 자동복귀용 레지스터(34)로 전송된 가상어드레스를 분기지로 하는 분기는 실행하지 않는다. 인터럽트 제어유닛(50)의 레지스터(501)에서 전송된 자동복귀용 레지스터(34)로 전송된 가상어드레스(명령R의 가상어드레스)로의 분기만을 실행한다. 또한, 명령w는 인터럽트 금지해제 명령q를 포함하는 지연슬롯을 갖는다. 이 때문에, 명령R은 명령q의 실행후에 실행된다.

다음에, 지연슬롯중의 명령에서 인터럽트가 발생한 경우에 대해서 설명한다.

도 16은 도 4에 도시한 프로그램의 명령P에서 F스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면, 도 17은 도 4에 도시한 프로그램의 명령P에서 R스테이지 실행중에 인터럽트가 발생한 경우에 있어서의 프로그램의 파이프라인의 흐름을 도시한 도면이다.

지연슬롯중의 명령에서 인터럽트가 발생한 경우에도 지연슬롯 중의 명령이외의 명령에서 인터럽트가 발생한 경우와 마찬가지로, 자동복귀유닛(33)이 식별신호선(813), 데이타신호선(814) 및 제어신호선(815)를 사용해서 인터럽트 제어유닛(50)의 레지스터(501)~레지스터(503)으로 순차 액세스하고, 레지스터(501)~레지스터(503)에 저장된 값을 자동복귀용 레지스터(34)에 저장한다. 또, 명령공급유닛(20) 및 명령실행유닛(30a)에 의해 도 13에 도시한 인터럽트 처리프로그램이 실행된다.

또한, 지연슬롯 중에 포함되는 명령P에서 인터럽트가 발생한 경우, 인터럽트 처리에서 복귀한 후 최초에 실행될 명령P의 다음에 실행될 명령X는 분기명령Q에 의해서 가상어드레스가 특정된 분기지의 명령이다. 이 때문에, 인터럽트 제어유닛(50)의 레지스터(503)에서 자동복귀용 레지스터(34)로 전송되는 값은 「1」로 된다. 따라서, 자동복귀유닛(33)은 명령w를 받아 인터럽트 제어유닛(50)의 레지스터(501)에서 자동복귀용 레지스터(34)로 전송된 가상어드레스(명령P의 가상어드레스)로의 분기와 인터럽트 제어유닛의 레지스터(502)에서 자동복귀용 레지스터(34)로 전송된 가상어드레스(명령X의 가상어드레스)로의 분기를 계속해서 실행한다.

본 실시예에서는 인터럽트처리에서 복귀할 때 프로그램을 소정의 순서로 정확하게 실행하기 위한 처리(도 6에 도시한 인터럽트 처리프로그램의 명령b~명령d, 명령o, 명령p, 명령r~명령u에 상당하는 처리)를 자동복귀유닛(33)을 사용해서 자동적으로 실행하고 있다. 이와 같이 하는 것에 의해, 인터럽트 처리프로그램의 처리시간을 단축할 수 있고 또 인터럽트 처리프로그램을 작게 할 수 있다.

다음에, 본 발명의 제3 실시예에 대해서 도면을 참조해서 설명한다.

도 18은 본 발명의 제3 실시예에서 사용하는 명령공급유닛의 개략구성도, 도 19는 본 발명의 제3 실시예에서 사용하는 인터럽트 제어유닛의 개략구성도이다. 본 실시예가 적용된 프로세서 및 그의 주변장치는 제1 실시예가 적용된 것과 명령공급유닛 및 인터럽트 제어유닛이 다른 점을 제외하고 기본적으로 동일하다. 그래서, 여기서는 본 실시예를 사용한 프로세서 및 그의 주변장치를 기능적으로 도시한 도면을 생략하고 있다. 또한, 본 실시예에 있어서 제1 실시예의 것과 동일기능을 갖는 것에는 동일부호 또는 대응하는 부호를 붙이고, 그의 상세한 설명은 생략한다.

본 실시예에서 사용하는 명령공급유닛(20a)가 도 2에 도시한 제1 실시예에서 사용하는 명령공급유닛(20)과 다른 점은 도 18에 도시한 바와 같이 비교기(215), 셀렉터(216)을 마련한 점이다.

비교기(215)는 분기지 어드레스선(811)을 거쳐서 전송되어 온 분기지의 가상어드레스를 소정의 가상어드레스와 비교한다. 여기서, 소정의 가상어드레스는 단일 값일 필요는 없고, 어느 정도의 범위를 갖게 해도 좋다. 비교기(215)는 분기지 어드레스선(811)을 거쳐서 전송된 분기지의 가상어드레스와 소정의 가상어드레스가 일치했을 때에 신호를 출력한다.

셀렉터(216)은 비교기(215)에서 신호가 출력된 경우, 분기요구 신호선(812)를 거쳐서 전송된 분기요구 대신에 분기를 실행하지 않는 것을 나타내는 신호(752)를 선택하고, 이 선택한 신호를 신호선(753)으로 출력한다.

본 실시예에서 사용하는 명령공급유닛(20a)에서는 분기지 어드레스선(811)을 거쳐서 전송되어 온 분기지의 가상어드레스가 소정의 가상어드레스와 일치하는 경우, 이 가상어드레스로의 분기를 실행하지 않도록 하고 있다.

본 실시예에서 사용하는 인터럽트 제어유닛(50a)가 도 3에 도시한 제1 실시예에서 사용한 인터럽트 제어유닛(50)과 다른 점은 도 19에 도시한 바와 같이 셀렉터(507)을 마련한 점이다.

셀렉터(507)은 명령공급유닛(20a)의 셀렉터(216)에 의해 선택된 분기를 실행하지 않는 것을 나타내는 신호(752)가 레지스터(503)에 저장되어 있는 경우, 레지스터(502)에 저장된 가상어드레스 대신에 분기지로서 있을 수 없는 가상어드레스(751)을 선택하고, 이 가상어드레스를 셀렉터(505)로 전송한다. 따라서, 레지스터(503)에 분기를 실행하지 않는 것을 나타내는 신호(752)가 저장되어 있는 경우, 명령실행유닛(30)은 식별신호선(813), 데이타신호선(814) 및 제어신호선(815)를 사용해서 레지스터(502)로 액세스하면, 레지스터(502)에 저장된 가상어드레스가 아니라 분기지로서 있을 수 없는 가상어드레스(751)이 리드된다.

도 20에 본 실시예에서 사용하는 프로세서에 의해 처리되는 인터럽트 처리프로그램의 1예를 도시한다. 이 프로그램에서는 명령a에 의해 인터럽트를 금지한 후, 명령b 및 명령c에 의해 인터럽트 제어유닛(50a)의 레지스터(501), (502)에 저장된 가상어드레스를 명령실행유닛(30)의 연산용 레지스터(31)에 있는 레지스터IA, IB로 각각 전송하고 있다. 그리고, 소정의 인터럽트처리를 실행한 후, 명령q에 의해 인터럽트 금지상태를 해제하고, 그 후 인터럽트t 및 명령u에 의해 각각 레지스터IA로 전송된 가상어드레스, 레지스터IB로 전송된 가상어드레스로의 분기를 실행하고 있다.

상술한 바와 같이, 본 실시예에서는 인터럽트 제어유닛(50a)의 레지스터(503)에 분기를 실행하지 않는 것을 나타내는 신호(752)가 저장되어 있는 경우, 명령실행유닛(30)이 인터럽트 제어유닛(50a)의 레지스터(502)로 액세스하면, 분기지로서 있을 수 없는 가상어드레스(751)이 리드된다. 따라서, 본 실시예에서는 인터럽트 처리에서 복귀한 후 최초에 실행할 명령의 다음에 실행할 명령이 분기명령에 의해서 분기지의 가상어드레스가 특정된 명령인지 아닌지의 판단, 즉 제1 실시예에서 사용하는 도 6에 도시한 인터럽트 처리프로그램의 명령d, 명령o, 명령p, 명령r, 명령s에 상당하는 처리를 명령실행유닛(30)에서 소프트웨어적으로 처리할 필요가 없어진다.

본 발명은 상기의 각 실시예에 한정되는 것은 아니고 그 요지의 범위내에서 여러가지로 변형이 가능하다.

예를 들면, 상기의 각 실시예에서는 설명을 간략화하기 위해 인터럽트가 F스테이지 및 R스테이지 중 어느 하나에서 발생하는 것으로서 설명하였다. 그러나, 실제로는 이와 같은 제한은 불필요하다. 1예로서 인터럽트가 E스테이지에서 발생한 경우에도 대응할 수 있는 명령공급유닛을 도 21에 도시한다.

도 21은 제1 실시예에서 사용한 명령공급유닛의 변형예를 도시한 도면이다.

도 21에 도시한 명령공급유닛(20b)가 도 2에 도시한 제1 실시예의 명령공급유닛(20)과 다른 점은 레지스터(220), 레지스터(221)을 마련한 점이다.

레지스터(220)은 어드레스선(702)로 출력된 가상어드레스를 유지함과 동시에, 지금까지 유지하고 있던 가상어드레스를 어드레스선(703)으로 출력한다. 상술한 바와 같이, 레지스터(203)은 R스테이지 실행중인 명령 즉 명령실행유닛(30)에서 디코드되고 있는 명령의 가상어드레스를 유지하므로, 레지스터(220)은 E스테이지 실행중인 명령 즉 명령실행유닛(30)에서 연산처리가 실행되고 있는 명령의 가상어드레스를 유지하게 된다.

레지스터(221)은 신호선(721)로 출력된 신호를 유지함과 동시에, 지금까지 유지하고 있던 신호를 신호선(722)로 출력한다. 상술한 바와 같이, 레지스터(204)는 R스테이지 실행중인 명령의 1개전에 실행되고 있는 명령 즉 E스테이지 실행중인 명령이 분기명령인지 아닌지를 나타내는 신호를 유지하므로, 레지스터(221)은 E스테이지 실행중인 명령의 1개전에 실행되고 있는 명령 즉 W스테이지 실행중인 명령이 분기명령인지 아닌지를 나타내는 신호를 유지하게 된다.

상기 구성의 명령공급유닛(20b)는 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50)에서 통지된 인터럽트 요구에 따라서 셀렉터(207)~(209)를 다음과 같이 동작시킨다.

인터럽트 제어유닛(50)으로부터 인터럽트 발생이 F스테이지에서 발생한 사실이 통지된 경우, 셀렉터(207), (208), (209)는 각각 신호선(701) 상의 값, 신호선(700)상의 값, 신호선(720)상의 값을 선택한다.

또, 인터럽트 제어유닛(50)으로부터 인터럽트 발생이 R스테이지에서 발생한 것이 통지된 경우에는 셀렉터(207), (208), (209)는 각각 신호선(702)상의 값, 신호선(701)상의 값, 신호선(721)상의 값을 선택한다.

그리고, 인터럽트 제어유닛(50)으로부터 인터럽트 발생이 E스테이지에서 발생한 것이 통지된 경우에는 셀렉터(207), (208), (209)는 각각 신호선(703)상의 값, 신호선(702)상의 값, 신호선(722)상의 값을 선택한다.

따라서, 상기 구성의 명령공급유닛(20b)에 의하면, 인터럽트가 F스테이지~E스테이지 중의 하나의 스테이지에서 발생한 경우에도 어드레스선(831)상으로 인터럽트가 발생한 명령의 가상어드레스를 출력할 수 있고, 어드레스선(832)상으로 인터럽트가 발생한 명령의 다음에 실행될 명령의 가상어드레스를 출력할 수 있고, 그리고 신호선(833)상으로 인터럽트가 발생한 명령의 다음에 실행할 명령의 가상어드레스가 분기명령에 의해서 특정되는 분기지의 가상어드레스인지 아닌지를 나타내는 신호를 출력할 수 있다.

또, 상기의 각 실시예에서는 설명을 간략화하기 위해 지연슬롯에 포함되는 명령이 1개인 경우에 대해서 설명하였다. 그러나, 실제로는 이와 같은 제한은 불필요하다. 1예로서 지연슬롯이 2개의 명령을 포함하는 경우에도 대응할 수 있는 프로세서 및 그의 주변장치에 대해서 설명한다.

도 22는 2개의 명령을 지연슬롯 중에 포함하는 지연분기를 실행하는 파이프라인방식의 프로세서와 그의 주변장치를 기능적으로 도시한 도면이다.

이 프로세서는 실행할 각 명령을 명령리드 전반의 F0스테이지, 명령리드 후반의 F1스테이지, 리드한 명령을 디코드해서 오퍼런드를 리드하는 R스테이지, 연산을 실행하는 E스테이지, 그리고 연산결과를 라이트하는 W스테이지의 5스테이지로 나누어서 실행하고 있다. 그리고, 각 스테이지의 실행에 1클럭 상당의 시간을 필요로 하고 있다. 따라서, 1개의 명령을 실행하기 위해서는 5클럭 상당의 시간이 필요하다.

또, 도 22에 도시한 프로세서 및 그의 주변장치가 도 1에 도시한 것과 다른 점은 명령공급유닛(20) 대신에 명령공급유닛(20c)를 사용한 점 및 인터럽트 제어유닛(50) 대신에 인터럽트 제어유닛(50b)를 사용한 점이다. 그 밖의 구성은 도 1에 도시한 제1 실시예의 것과 동일하다.

도 23은 도 22에 도시한 명령공급유닛(20c)의 개략구성도이다. 여기서, 도 2 또는 도 21에 도시한 것과 동일기능을 갖는 것에는 동일부호를 붙이고 있다.

도 23에 도시한 명령공급유닛(20c)는 인터럽트 신호선(834)를 거쳐서 인터럽트 제어유닛(50b)에서 통지된 인터럽트 요구에 따라서 셀렉터(231)~(235)를 다음과 같이 동작시킨다.

인터럽트 제어유닛(50b)로부터 인터럽트가 F스테이지에서 발생한 사실이 통지된 경우, 셀렉터(231), (232), (233), (234), (235)는 각각 신호선(702)상의 값, 신호선(701)상의 값, 신호선(700)상의 값, 신호선(721)상의 값, 신호선(720)상의 값을 선택한다.

인터럽트 제어유닛(50b)으로부터 인터럽트가 R스테이지에서 발생한 사실이 통지된 경우, 셀렉터(231), (232), (233), (234), (235)는 각각 신호선(703)상의 값, 신호선(702)상의 값, 신호선(701)상의 값, 신호선(722)상의 값, 신호선(721)상의 값을 선택한다.

상기 구성의 명령공급유닛(20c)에서는 인터럽트가 발생한 경우에 어드레스선(851)상으로 인터럽트처리가 종료된 후 최초에 실행할 명령의 가상어드레스가 출력되고, 어드레스선(852)상으로 인터럽트처리가 종료된 후 최초에 실행할 명령의 다음에 실행할 명령의 가상어드레스가 출력되고, 어드레스선(853)상으로 인터럽트처리가 종료된 후 최초에 실행할 명령의 다음에 실행 할 명령의 가상어드레스가 출력된다. 또, 신호선(855)상으로는 어드레스선(852)상으로 출력된 가상어드레스가 분기명령에 의해서 특정되는 분기지의 명령의 가상어드레스인지 아닌지를 나타내는 값이 출력되고, 신호선(856)상으로는 어드레스선(853)상으로 출력된 가상어드레스가 분기명령에 의해서 특정되는 분기지의 명령의 가상어드레스인지 아닌지를 나타내는 값이 출력된다.

도 24는 도 22에 도시한 인터럽트 제어유닛(50b)의 개략 구성도이다. 여기서, 도 3에 도시한 것과 동일기능을 갖는 것에는 동일부호를 붙이고 있다.

도 24에 도시한 인터럽트 제어유닛(50b)는 명령공급유닛(20c)의 셀렉터(231), (232), (233), (234), (235)에서 전송된 값을 각각 레지스터(551), (552), (553), (554), (555)에 저장한다. 명령실행유닛(30)은 식별신호선(813), 데이타신호선(814) 및 제어신호선(815)를 사용해서 셀렉터(560), 쌍방향 드라이버(506)을 제어하는 것에 의해 원하는 레지스터(551)~(555)로 액세스할 수 있다.

도 25는 도 22에 도시한 프로세서가 도 4에 도시한 프로그램을 실행한 경우의 파이프라인의 흐름을 도시한 도면이다. 상기 각 실시예와 마찬가지로 실행중인 명령이 분기명령인 것을 알 수 있는 것이 R스테이지인 것으로 하면, 분기명령Q의 후속명령인 명령P, 명령R의 2명령이 분기명령Q의 지연슬롯중의 명령으로 된다.

도 26은 도 22에 도시한 프로세서에 의해 처리되는 인터럽트 처리프로그램의 1예를 도시한 도면이다. 이 인터럽트 처리프로그램은 도 6에 도시한 제1 실시예의 인터럽트 처리프로그램과 마찬가지로, 명령에 의해 인터럽트가 발생했을 때에 실행된다. 또한, 도 6에서는 인터럽트 처리에서 복귀할 때에 연속해서 발행되는 분기명령이 최대 2개이지만, 여기서는 최대 3개로 된다(도 26의 L11참조).

마지막으로, 본 발명의 인터럽트 제어장치의 프로세서를 사용한 시스템 구성예를 도 27에 도시한다. 도 27은 도 1에 도시한 프로세서 및 그의 주변장치에 키보드나 디스플레이 등의 I/O디바이스(60)을 접속하고 있다. 명령실행유닛(30)은 어드레스선(860), 데이타선(861) 및 제어신호선(862)를 사용해서 I/O디바이스(60)으로 액세스하고 그의 동작을 제어한다.

발명의 효과

이상 설명한 바와 같이, 본 발명에 의하면 CISC나 압축후의 VLIW 등 명령길이가 일정하지 않은 명령을 실행하는 경우에도 인터럽트가 발생한 명령이 지연슬롯중의 명령인지의 여부에 관계없이 인터럽트를 정확하게 처리할 수 있는 파이프라인방식의 프로세서 및 계산기 시스템을 제공할 수 있다.

(57) 청구의 범위

청구항 1.

적어도 1개의 명령을 지연슬롯중에 포함하는 분기명령을 실행하는 파이프라인방식의 연산처리장치에 있어서 기억장치에 가변길이로 저장된 상기 분기명령을 포함하는 여러개의 명령을 순차 리드해서 처리할 때에 사용하는 인터럽트 제어장치로서, 인터럽트가 발생한 경우에 이 인터럽트가 발생한 명령에서 상기 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 어드레스를 저장하는 어드레스 저장수단; 상기 인터럽트 처리가 완료된 후에 상기 어드레스 저장수단에 저장된 각 어드레스를 상기 인터럽트가 발생한 명령의 어드레스에서 순차 리드하는 어드레스 리드수단; 상기 어드레스 리드수단에 의해서 최초에 리드한 어드레스로 분기해서 명령을 실행하는 제1 명령실행수단 및; 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스로 분기해서 명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 현재 실행되고 있는 명령의 후속명령을 실행하는 제2 명령실행수단을 구비하는 것을 특징으로 하는 인터럽트 제어장치.

청구항 2.

제1항에 있어서, 실행하고 있는 명령의 어드레스에 상기 명령의 명령길이를 가산해서 다음에 리드해서 실행할 명령의 어드레스를 생성하는 어드레스 생성수단을 더 구비하고, 상기 제2 명령실행수단은 상기 인터럽트가 발생한 시점에서 이 인터럽트가 발생한 명령의 명령길이가 판명되어 있는 경우에는 상기 어드레스 리드수단에 의해 2번째로 리드한 어드레스가 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스이더라도 상기 어드레스로 분기해서 명령을 실행하는 것을 특징으로 하는 인터럽트 제어장치.

청구항 3.

제1항에 있어서, 상기 인터럽트가 발생한 명령의 다음에 실행할 명령에서 상기 인터럽트가 발생한 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 명령에 대해서 상기 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 이전에 실행한 명령이 상기 분기명령인지 아닌지를 나타내는 정보를 저장하는 분기정보 저장수단을 더 구비하고, 상기 제2 명령실행수단은 상기 분기정보 저장수단에 저장된 정보에 따라서 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스인지 아닌지를 판단하는 것을 특징으로 하는 인터럽트 제어장치.

청구항 4.

제1항에 있어서, 상기 인터럽트가 발생한 명령의 다음에 실행할 명령에서 상기 인터럽트가 발생한 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 명령에 대해서 상기 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 이전에 실행한 명령이 상기 분기명령이 아닌 경우에는 상기 명령의 어드레스로서 특수한 값의 어드레스를 상기 어드레스 저장수단에 저장시키는 어드레스저장 제어수단을 더 구비하고, 상기 제2 명령실행수단은 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 특수한 값일 때에는 상기 어드레스로의 분기를 실행하지 않는 것을 특징으로 하는 인터럽트 제어장치.

청구항 5.

제2항에 있어서, 상기 인터럽트가 발생한 명령의 다음에 실행할 명령에서 상기 인터럽트가 발생한 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 명령에 대해서 상기 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 이전에 실행한 명령이 상기 분기명령인지 아닌지를 나타내는 정보를 저장하는 분기정보 저장수단을 더 구비하고, 상기 제2 명령실행수단은 상기 분기정보 저장수단에 저장된 정보에 따라서 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스인지 아닌지를 판단하는 것을 특징으로 하는 인터럽트 제어장치.

청구항 6.

제2항에 있어서, 상기 인터럽트가 발생한 명령의 다음에 실행할 명령에서 상기 인터럽트가 발생한 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 명령에 대해서 상기 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 이전에 실행한 명령이 상기 분기명령이 아닌 경우에는 상기 명령의 어드레스로서 특수한 값의 어드레스를 상기 어드레스 저장수단에 저장시키는 어드레스저장 제어수단을 더 구비하고, 상기 제2 명령실행수단은 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 특수한 값일 때에는 상기 어드레스로의 분기를 실행하지 않는 것을 특징으로 하는 인터럽트 제어장치.

청구항 7.

제1항에 있어서, 상기 제1 명령실행수단은 상기 어드레스 리드수단에 의해서 최초에 리드한 어드레스를 점프지로 하는 분기명령을 실행하는 것이고, 상기 제2 명령실행수단은 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스를 점프지로 하는 분기명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 아무것도 하지 않는 것을 나타내는 커맨드인 논오퍼레이션 명령을 실행하는 것인 것을 특징으로 하는 인터럽트 제어장치.

청구항 8.

제2항에 있어서, 상기 제1 명령실행수단은 상기 어드레스 리드수단에 의해서 최초에 리드한 어드레스를 점프지로 하는 분기명령을 실행하는 것이고, 상기 제2 명령실행수단은 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스를 점프지로 하는 분기명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 아무것도 하지 않는 것을 나타내는 커맨드인 논오퍼레이션 명령을 실행하는 것인 것을 특징으로 하는 인터럽트 제어장치.

청구항 9.

제3항에 있어서, 상기 제1 명령실행수단은 상기 어드레스 리드수단에 의해서 최초에 리드한 어드레스를 점프지로 하는 분기명령을 실행하는 것이고, 상기 제2 명령실행수단은 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스를 점프지로 하는 분기명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 아무것도 하지 않는 것을 나타내는 커맨드인 논오퍼레이션 명령을 실행하는 것인 것을 특징으로 하는 인터럽트 제어장치.

청구항 10.

제4항에 있어서, 상기 제1 명령실행수단은 상기 어드레스 리드수단에 의해서 최초에 리드한 어드레스를 점프지로 하는 분기명령을 실행하는 것이고, 상기 제2 명령실행수단은 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스를 점프지로 하는 분기명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 아무것도 하지 않는 것을 나타내는 커맨드인 논오퍼레이션 명령을 실행하는 것인 것을 특징으로 하는 인터럽트 제어장치.

청구항 11.

제5항에 있어서, 상기 제1 명령실행수단은 상기 어드레스 리드수단에 의해서 최초에 리드한 어드레스를 점프지로 하는 분기명령을 실행하는 것이고, 상기 제2 명령실행수단은 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스를 점프지로 하는 분기명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 아무것도 하지 않는 것을 나타내는 커맨드인 논오퍼레이션 명령을 실행하는 것인 것을 특징으로 하는 인터럽트 제어장치.

청구항 12.

제6항에 있어서, 상기 제1 명령실행수단은 상기 어드레스 리드수단에 의해서 최초에 리드한 어드레스를 점프지로 하는 분기명령을 실행하는 것이고, 상기 제2 명령실행수단은 상기 어드레스 리드수단에 의해서 2번째이후에 리드한 각 어드레스에 대해서 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스를 점프지로 하는 분기명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 아무것도 하지 않는 것을 나타내는 커맨드인 논오퍼레이션 명령을 실행하는 것인 것을 특징으로 하는 인터럽트 제어장치.

청구항 13.

명령저장용 메모리에 가변길이로 저장된 분기명령을 포함하는 여러개의 명령을 순차 리드해서 실행하는 파이프라인방식의 프로세서로서, 상기 명령저장용 메모리에서 명령을 순차 리드함과 동시에, 상기 리드한 명령이 분기명령인 경우에 상기 분기명령의 지연슬롯에 포함되는 후속명령을 리드한 후에 상기 분기명령에 의해서 지정되는 분기지 어드레스의 명령을 리드하는 명령공급유닛; 상기 명령공급유닛에 의해서 리드된 명령을 실행함과 동시에, 상기 리드된 명령이 분기명령인 경우에 이 분기명령에 의해서 지정되는 분기지 어드레스를 상기 명령공급유닛에 통지하는 명령실행유닛 및; 인터럽트가 발생한 경우에 인터럽트처리로부터의 복귀후에 실행할 명령을 제어하는 인터럽트 제어유닛을 구비하고, 상기 인터럽트 제어유닛은 인터럽트가 발생한 경우에 이 인터럽트가 발생한 명령에서 상기 명령의 상기 지연슬롯에 포함되

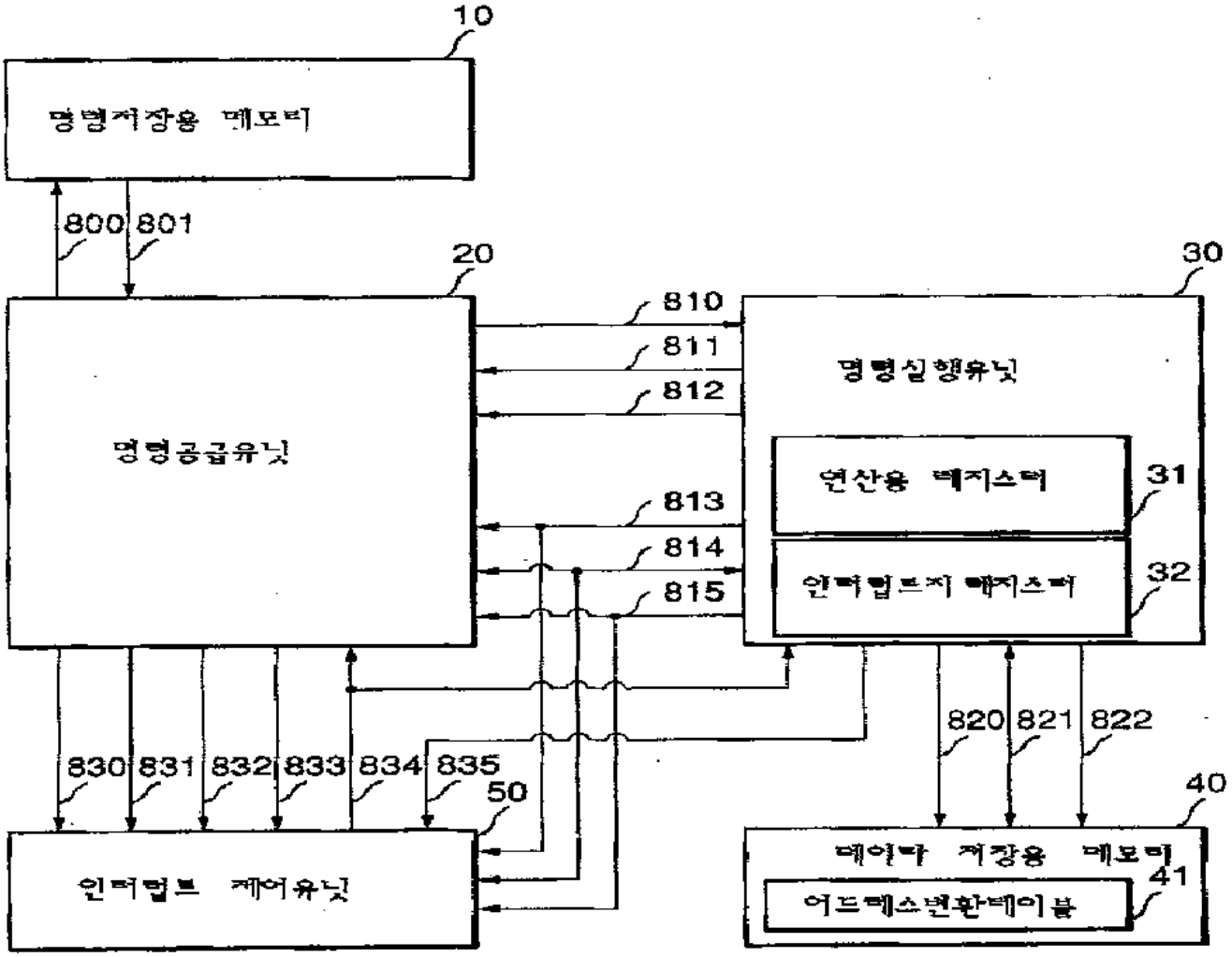
는 명령의 수만큼 나중에 실행할 명령까지의 각 어드레스를 저장하는 레지스터를 갖고, 상기 인터럽트처리가 완료된 후에 상기 레지스터에 저장된 각 어드레스를 상기 인터럽트가 발생한 명령의 어드레스에서 순차 리드하는 것이고, 상기 명령공급유닛은 인터럽트 처리로부터의 복귀시에 상기 인터럽트 제어유닛에 의해서 상기 레지스터에서 최초에 리드한 어드레스의 명령을 리드함과 동시에, 상기 인터럽트 제어유닛에 의해서 상기 레지스터에서 2번째이후에 리드한 어드레스에 대해서는 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스로 분기해서 명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 최신에 리드한 명령의 후속명령을 리드하는 것인 것을 특징으로 하는 프로세서.

청구항 14.

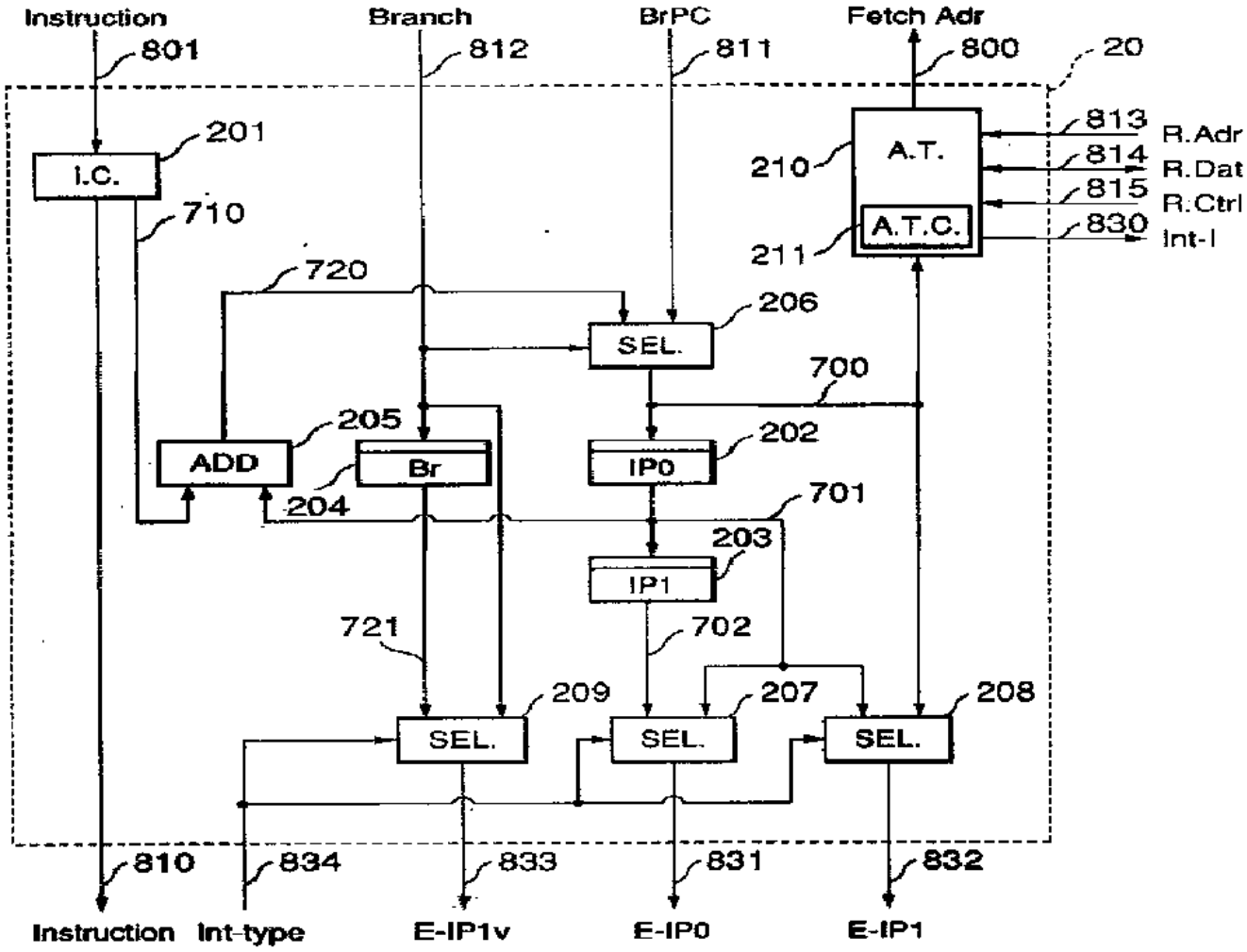
명령저장용 메모리에 가변길이로 저장된 분기명령을 포함하는 여러개의 명령을 순차 리드해서 실행하는 파이프라인방식의 프로세서로서, 상기 명령저장용 메모리에서 명령을 순차 리드함과 동시에 상기 리드한 명령이 분기명령인 경우에 상기 분기명령의 지연슬롯에 포함되는 후속명령을 리드한 후에 상기 분기명령에 의해서 지정되는 분기지 어드레스의 명령을 리드하는 명령공급유닛, 상기 명령공급유닛에 의해서 리드된 명령을 실행함과 동시에 상기 리드된 명령이 분기명령인 경우에 이 분기명령에 의해서 지정되는 분기지 어드레스를 상기 명령공급유닛에 통지하는 명령실행유닛 및, 인터럽트가 발생한 경우에 인터럽트처리로부터의 복귀후에 실행할 명령을 제어하는 인터럽트 제어유닛을 구비하고, 상기 인터럽트 제어유닛은 인터럽트가 발생한 경우에 이 인터럽트가 발생한 명령에서 상기 명령의 상기 지연슬롯에 포함되는 명령의 수만큼 나중에 실행할 명령까지의 각 어드레스를 저장하는 레지스터를 갖고, 상기 인터럽트처리가 완료된 후에 상기 레지스터에 저장된 각 어드레스를 상기 인터럽트가 발생한 명령의 어드레스에서 순차 리드하는 것이고, 상기 명령공급유닛은 인터럽트 처리로부터의 복귀시에 상기 인터럽트 제어유닛에 의해서 상기 레지스터에서 최초에 리드한 어드레스의 명령을 리드함과 동시에, 상기 인터럽트 제어유닛에 의해서 상기 레지스터에서 2번째 이후에 리드한 어드레스에 대해서는 상기 어드레스가 상기 분기명령에 의해서 지정된 명령의 어드레스일 때에는 상기 어드레스로 분기해서 명령을 실행하고, 상기 분기명령에 의해서 지정된 명령 이외의 명령의 어드레스일 때에는 최신에 리드한 명령의 후속명령을 리드하는 프로세서; 분기명령을 포함하는 여러개의 명령이 가변길이로 저장된 명령저장용 메모리 및; 상기 프로세서를 입출력장치에 접속하는 I/O디바이스를 구비한 것을 특징으로 하는 계산기 시스템.

도면

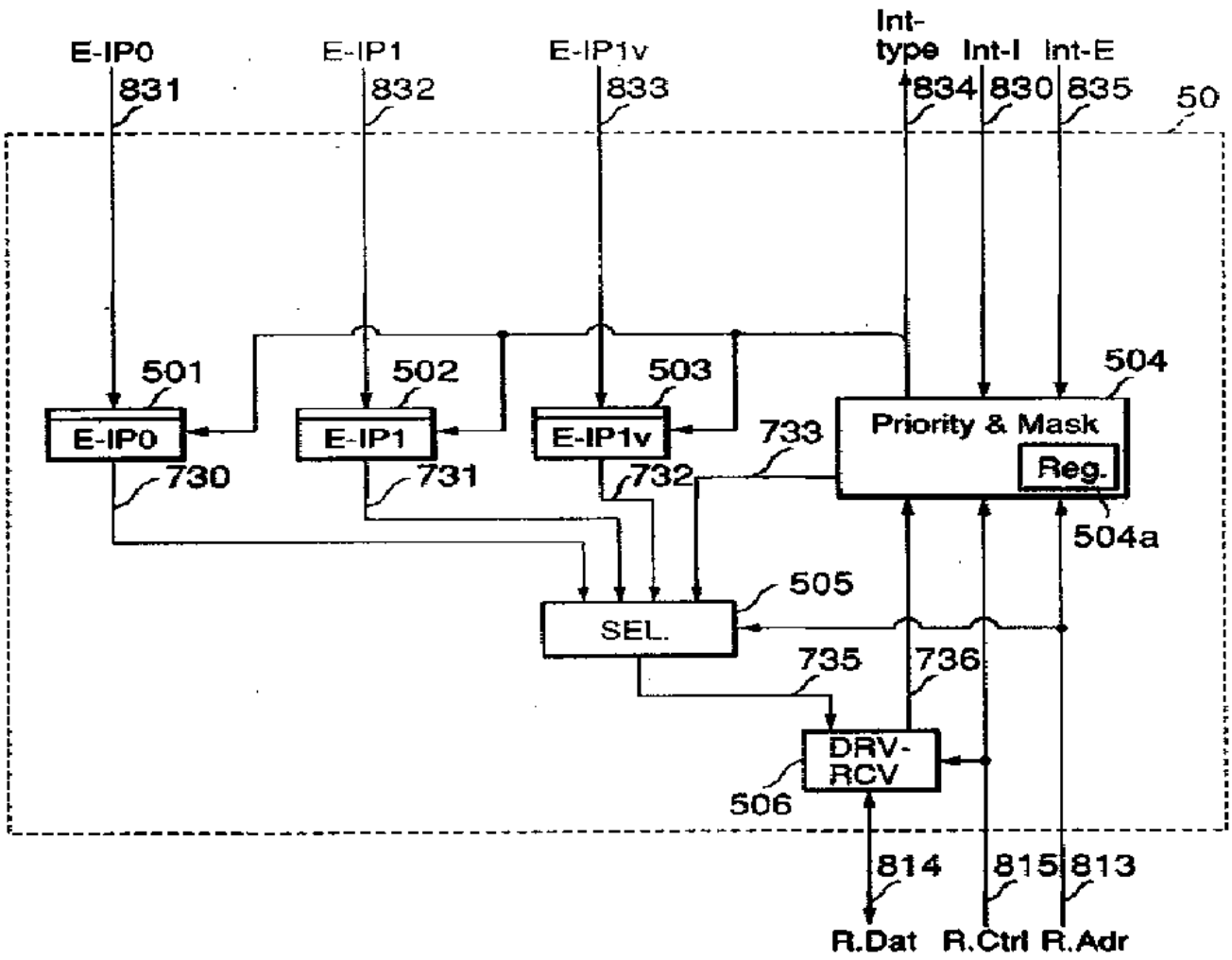
도면 1


10
명령저장용 메모리
800
801
20
명령공급유닛
30
명령실행유닛
810
811
812
813
814
815
연산용 레지스터
31
인터럽트지 레지스터
32
820
821
822
830
831
832
833
834
835
50
인터럽트 제어유닛
40
데이타 저장용 메모리
41
어드레스변환테이블

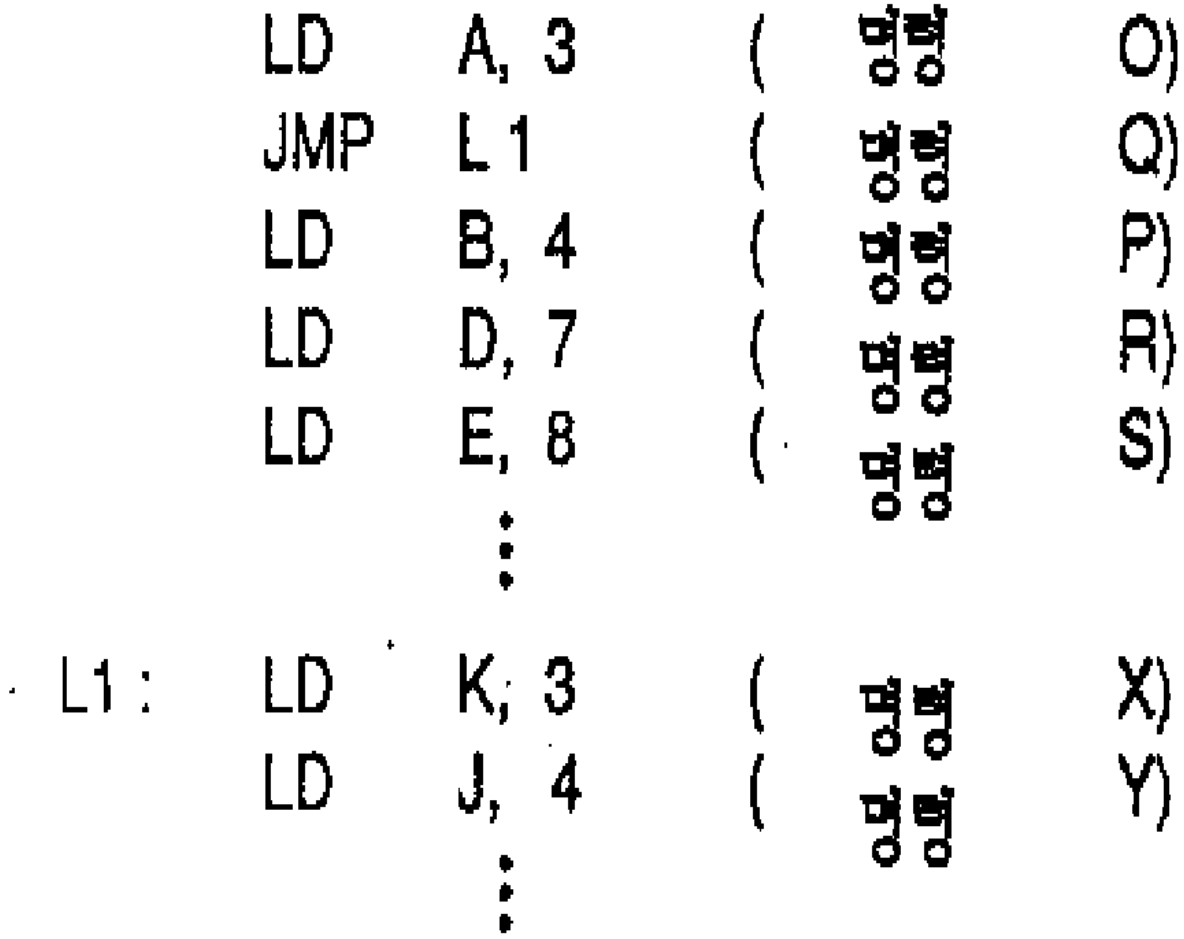
도면 2


Instruction
801
Branch
812
BrPC
811
Fetch Adr
800
20
I.C.
201
710
A.T.
210
A.T.C.
211
813
R.Adr
814
R.Dat
815
R.Ctrl
830
Int-I
720
SEL.
206
700
ADD
205
Br
204
IP0
202
701
IP1
203
721
702
SEL.
209
SEL.
207
SEL.
208
810
834
833
831
832
Instruction
Int-type
E-IP1v
E-IP0
E-IP1

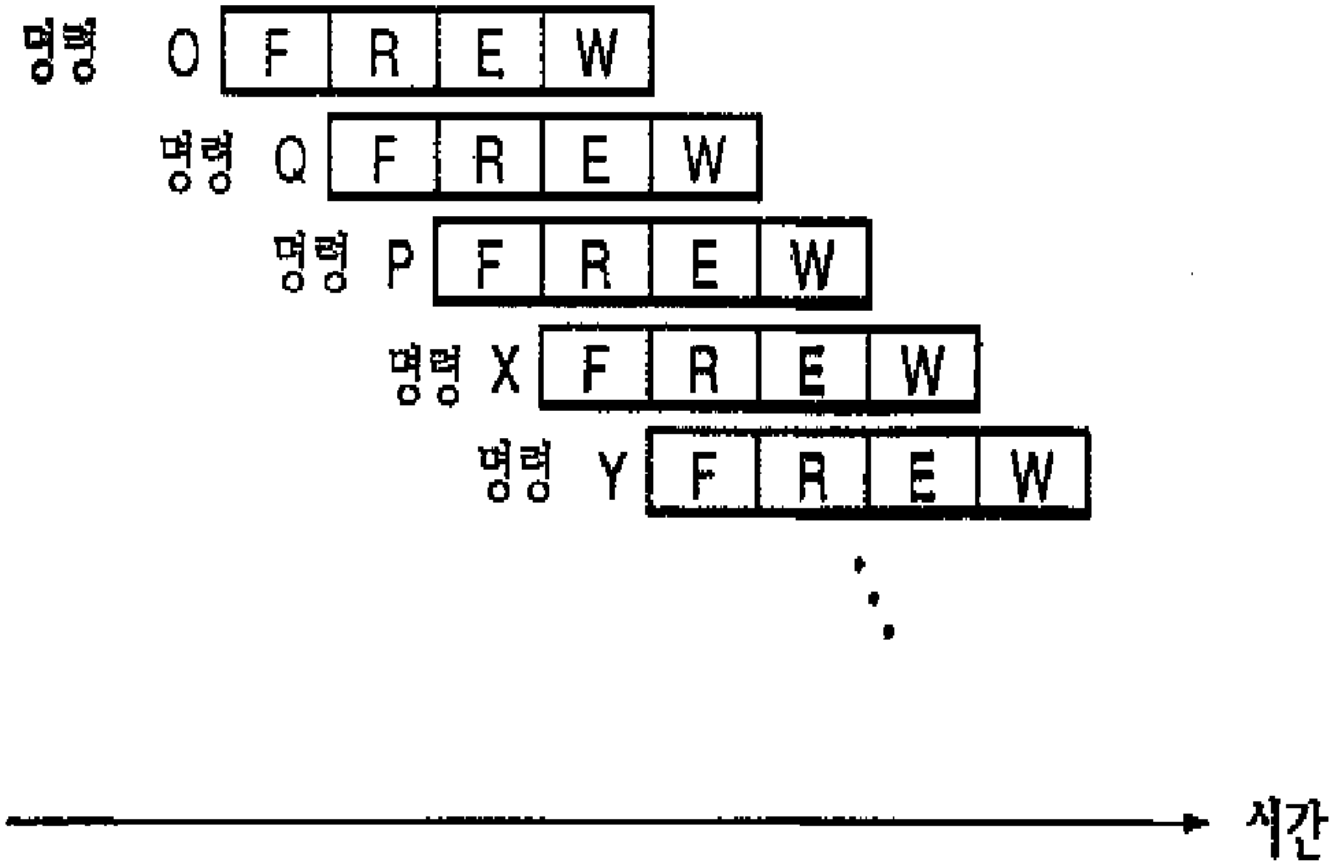
도면 3


E-IP0
831
E-IP1
832
E-IP1v
833
Int-type
834
Int-I
830
Int-E
835
50
501
E-IP0
730
502
E-IP1
731
503
E-IP1v
732
733
504
Priority & Mask
Reg.
504a
505
SEL.
735
736
DRV-RCV
506
814
815
813
R.Dat
R.Ctrl
R.Adr

도면 4



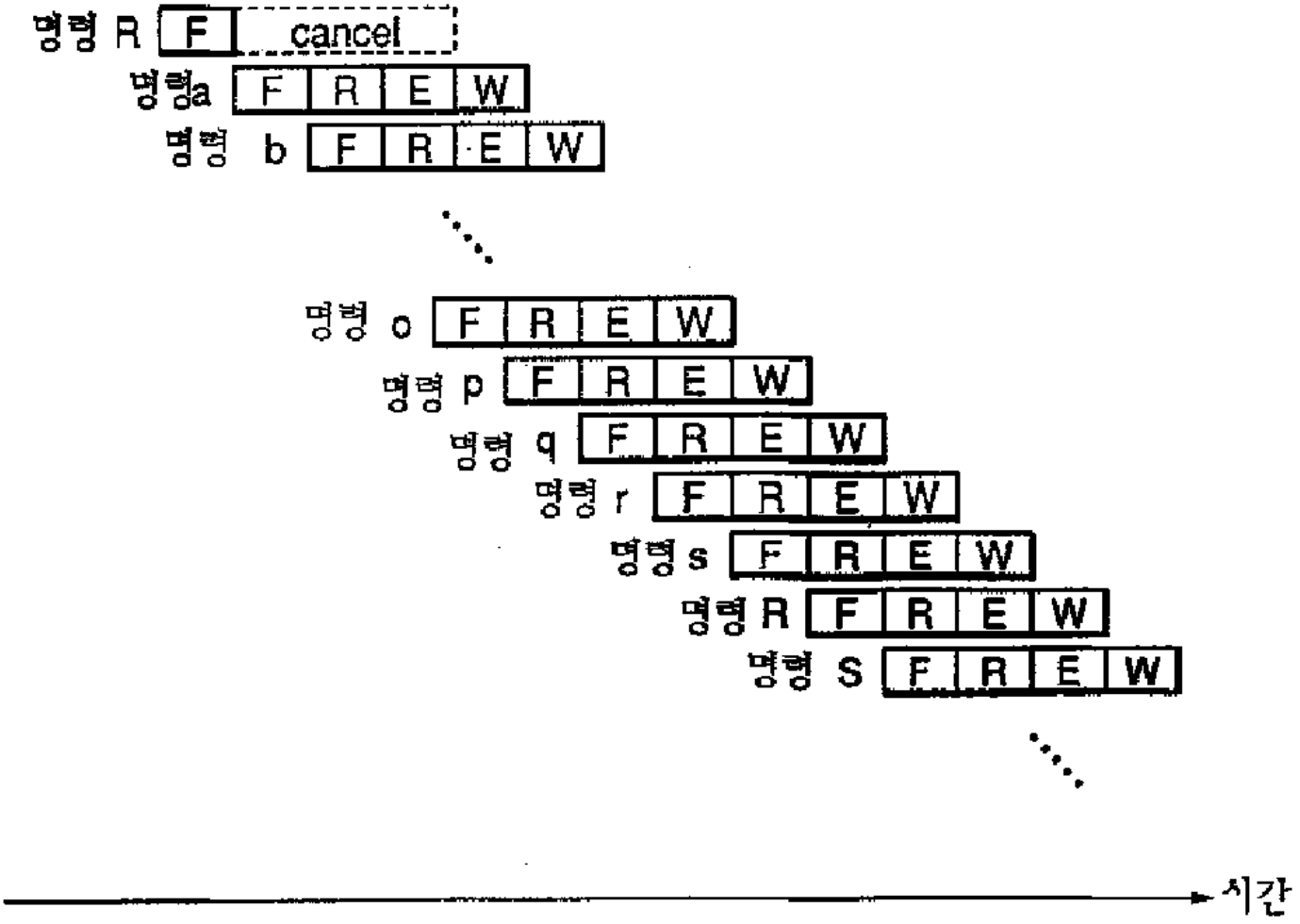
도면 5



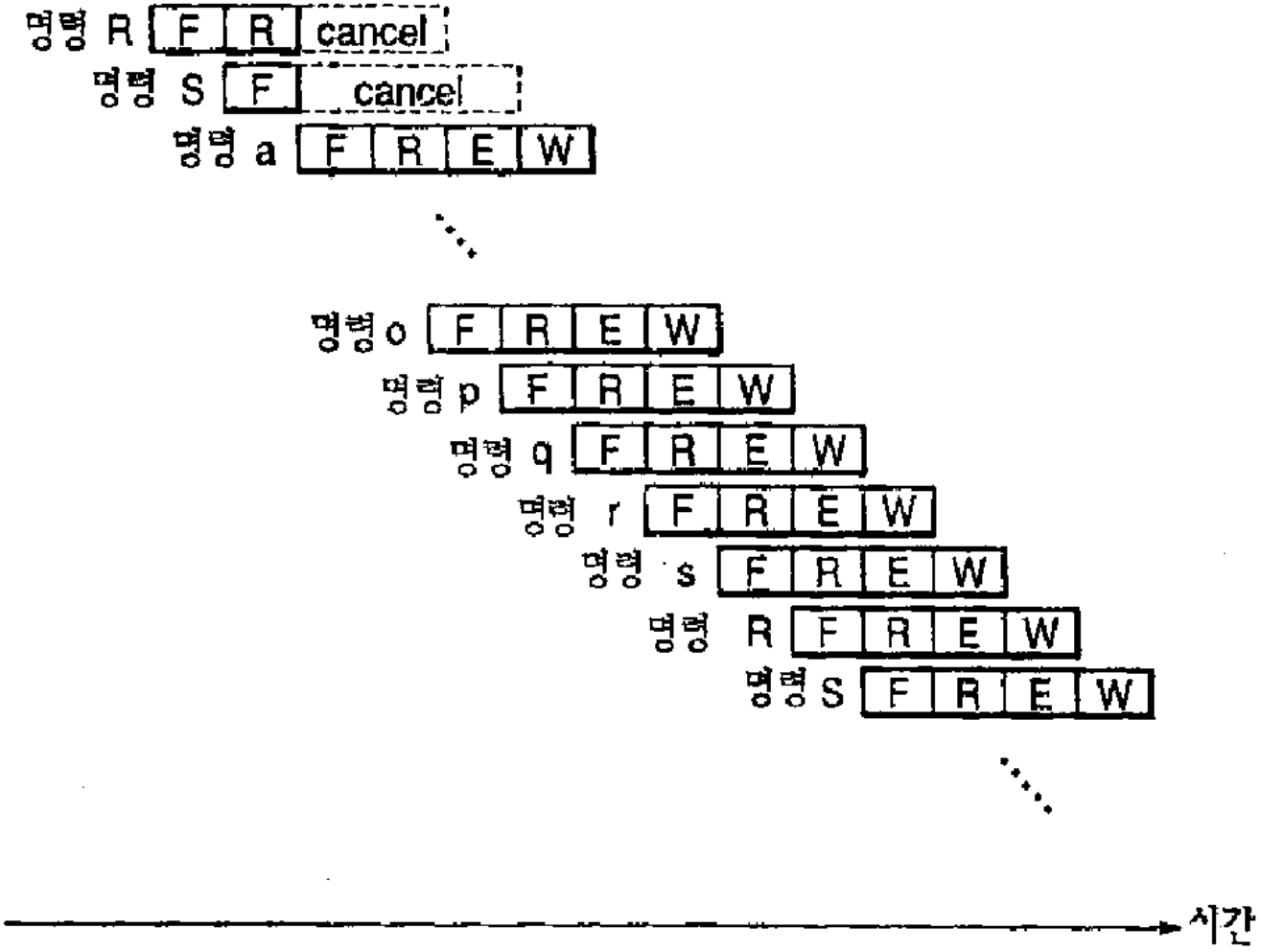
도면 6

```
        DI                    (  명령   a)
        LD    IA,  EIP0       (  명령   b)
        LD    IB,  EIP1       (  명령   c)
        LD    IC,  EIP1V      (  명령   d)
        LD    ID,  7          (  명령   e)
                ⋮
        CMP   IC,  0          (  명령   o)
        JNE   L2              (  명령   p)
        EI                    (  명령   q)
        JMP   IA              (  명령   r)
        NOP                   (  명령   s)
L2 :    JMP   IA              (  명령   t)
        JMP   IB              (  명령   u)
```

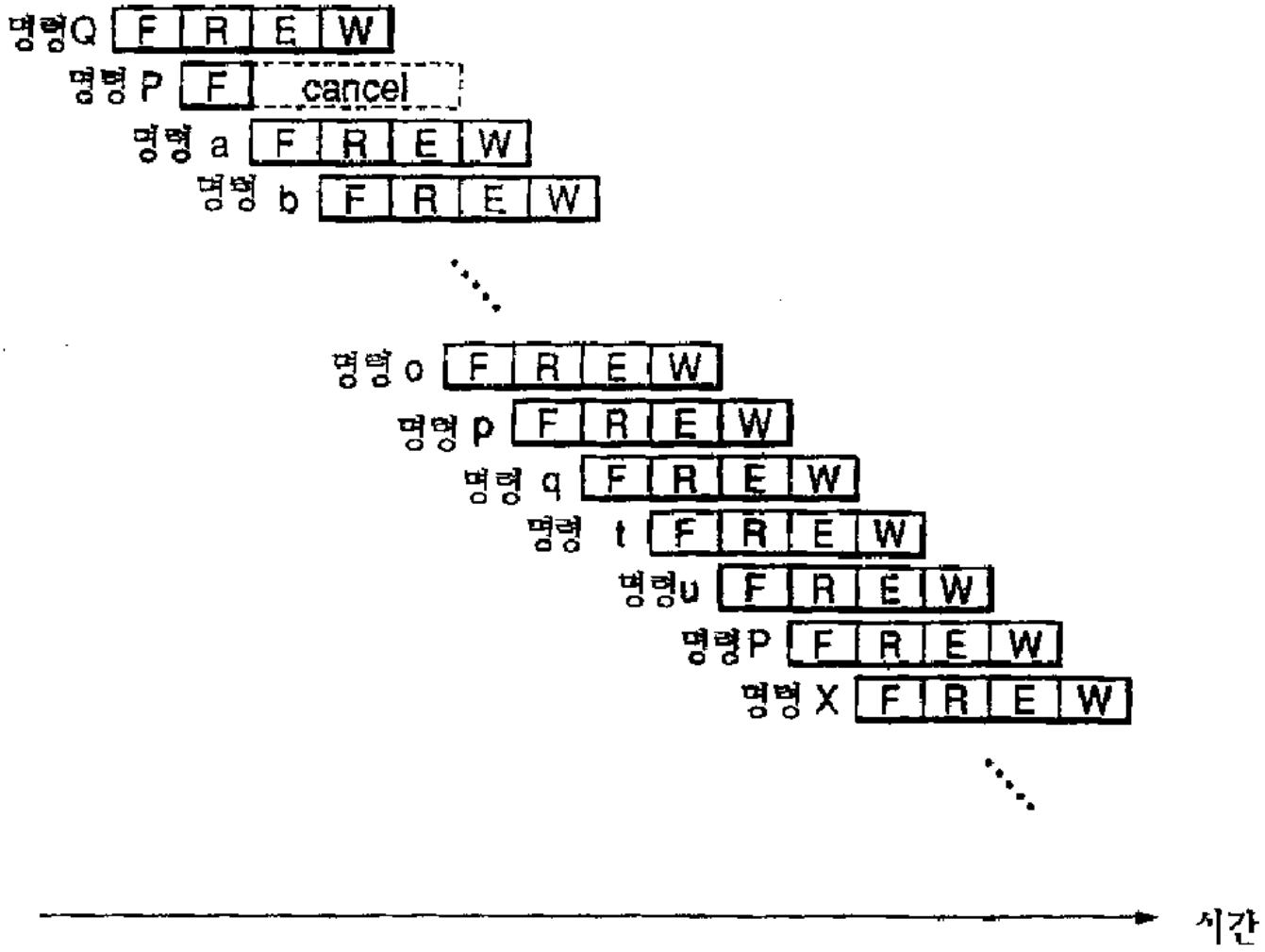
도면 7


명령 R
F
cancel
명령a
F
R
E
W
명령 b
F
R
E
W
명령 o
F
R
E
W
명령 p
F
R
E
W
명령 q
F
R
E
W
명령 r
F
R
E
W
명령 s
F
R
E
W
명령 R
F
R
E
W
명령 S
F
R
E
W
시간

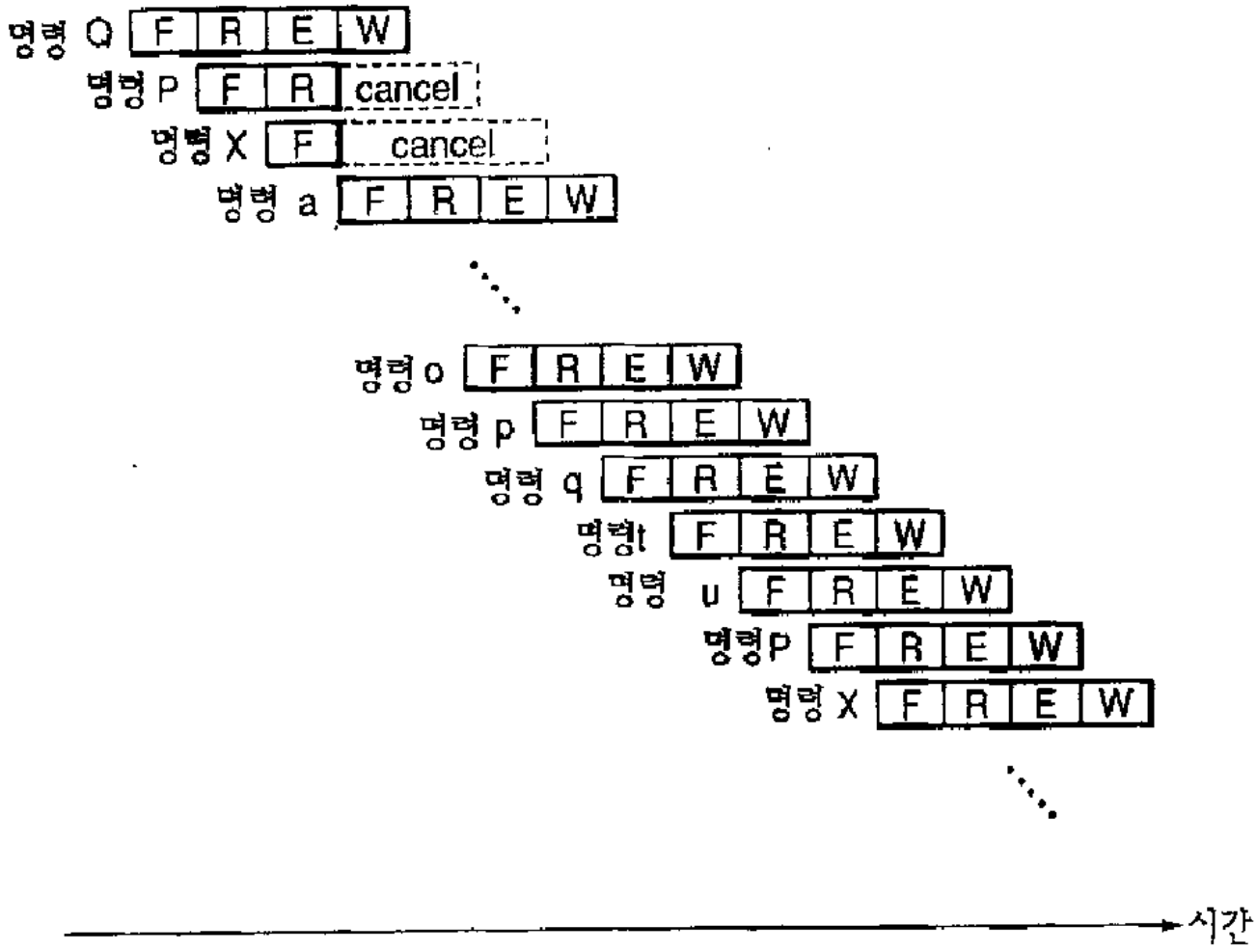
도면 8


명령 R
F
R
cancel
명령 S
F
cancel
명령 a
F
R
E
W
명령 o
F
R
E
W
명령 p
F
R
E
W
명령 q
F
R
E
W
명령 r
F
R
E
W
명령 s
F
R
E
W
명령 R
F
R
E
W
명령 S
F
R
E
W
시간

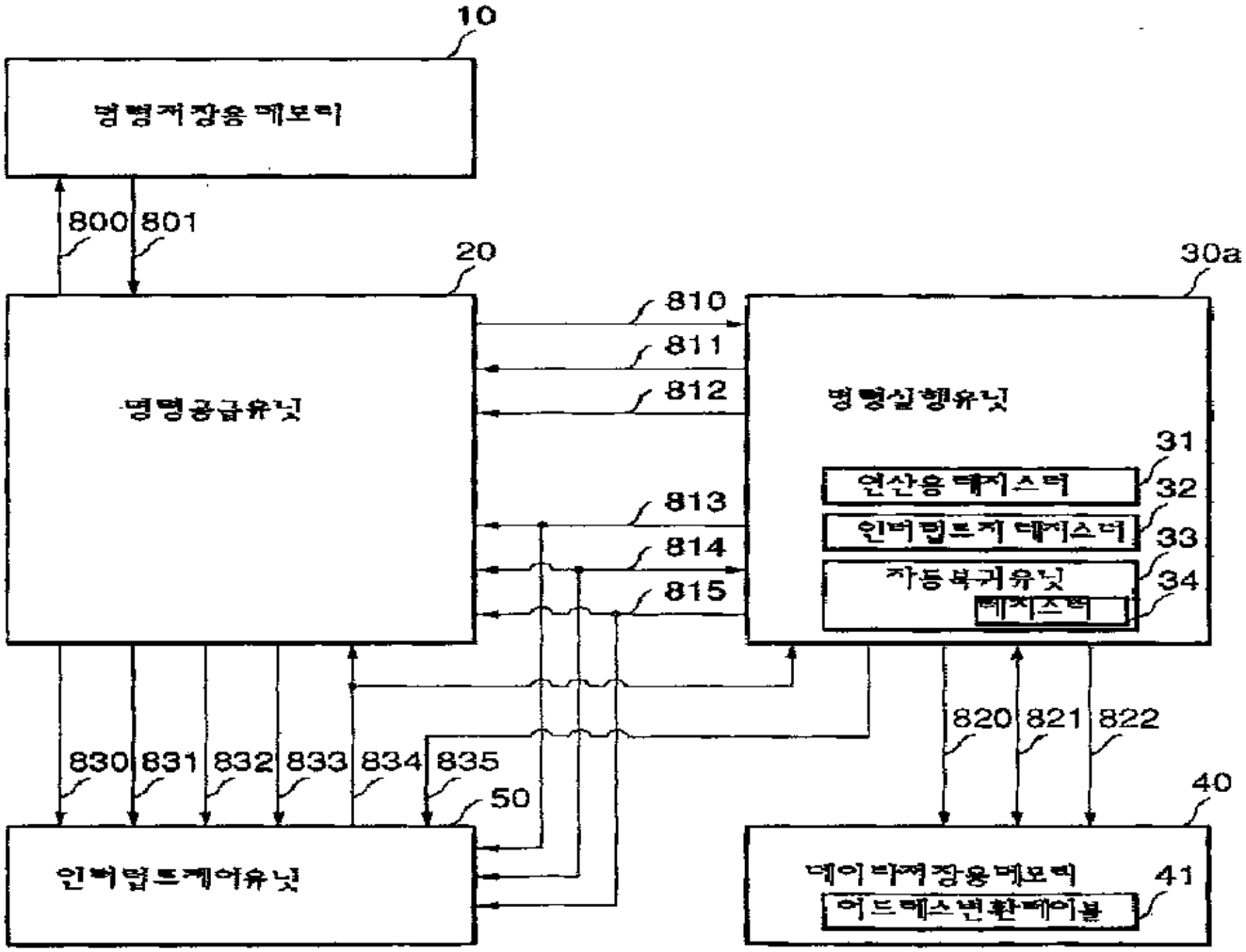
도면 9


명령Q F R E W
명령 P F cancel
명령 a F R E W
명령 b F R E W
명령 o F R E W
명령 p F R E W
명령 q F R E W
명령 t F R E W
명령u F R E W
명령P F R E W
명령 X F R E W
시간

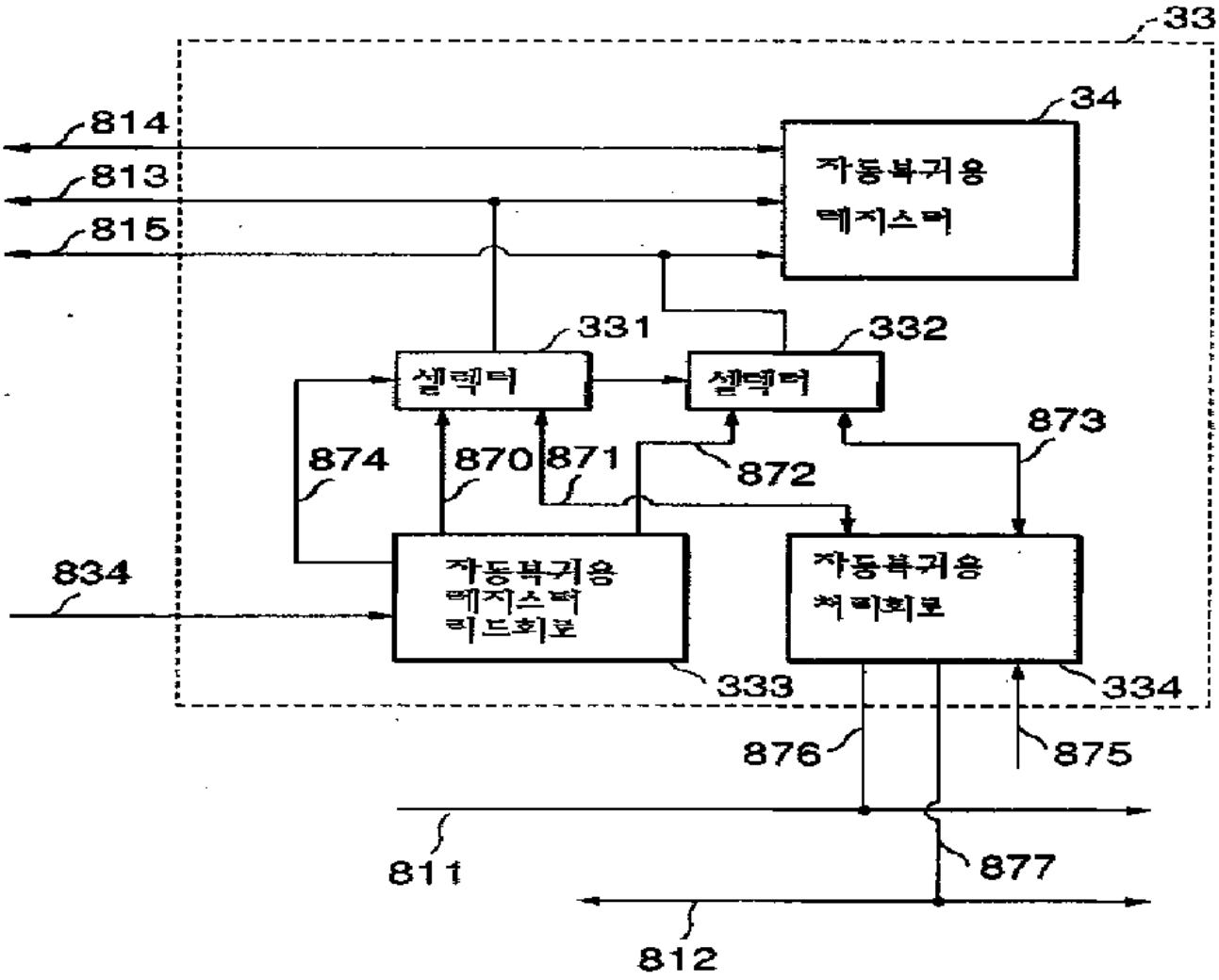
도면 10


명령 Q F R E W
명령 P F R cancel
명령 X F cancel
명령 a F R E W
명령 o F R E W
명령 p F R E W
명령 q F R E W
명령 t F R E W
명령 u F R E W
명령 P F R E W
명령 X F R E W
시간

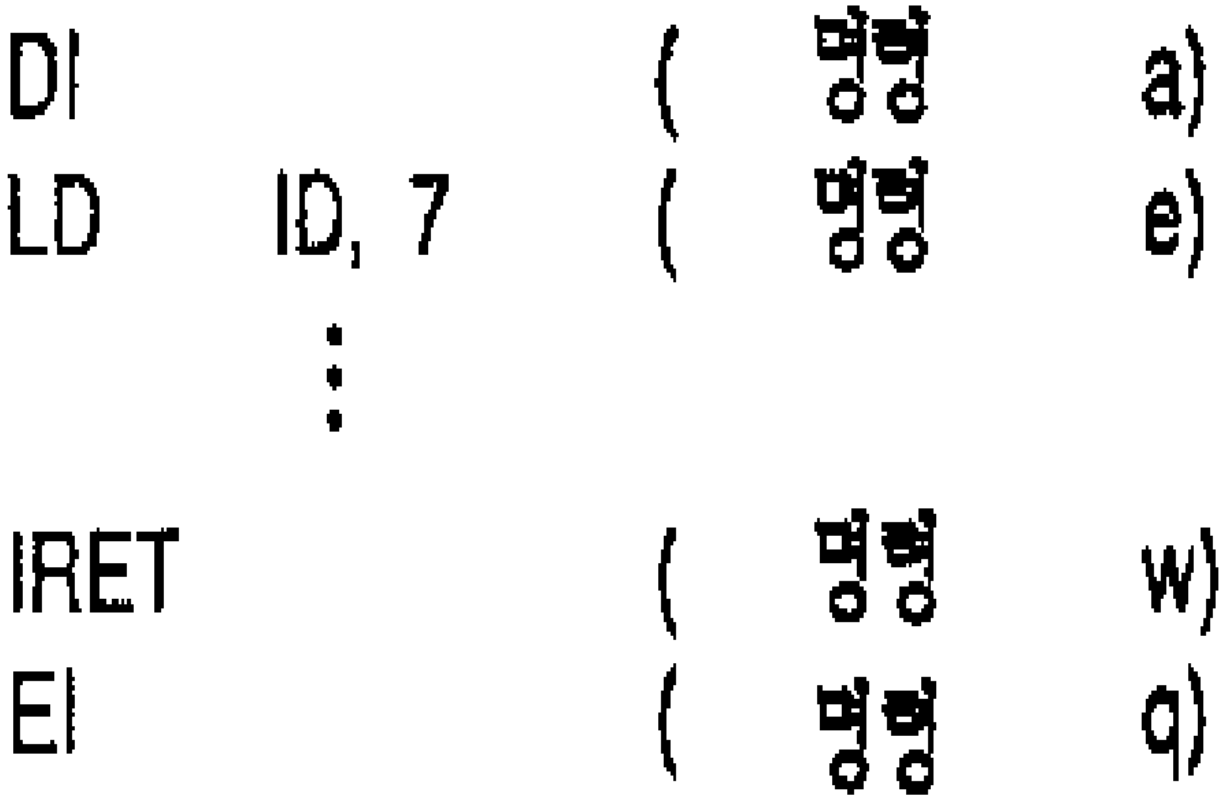
도면 11


10
명령저장용메모리
800
801
20
명령공급유닛
30a
명령실행유닛
810
811
812
813
814
815
31
연산용레지스터
32
인터럽트지 레지스터
33
자동복귀유닛
34
레지스터
820
821
822
830
831
832
833
834
835
50
인터럽트제어유닛
40
데이터저장용메모리
41
어드레스변환테이블

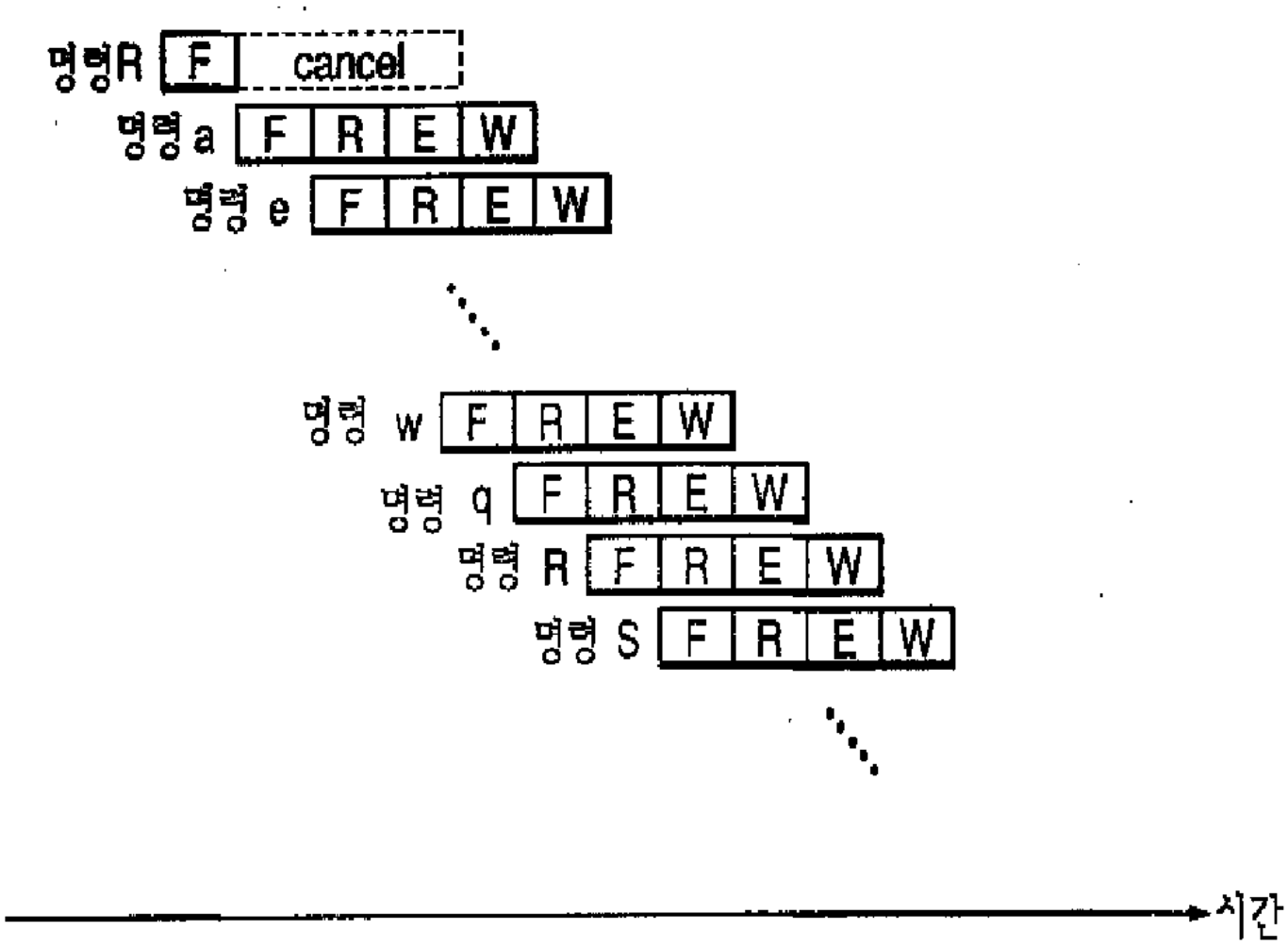
도면 12


33
34
814
813
815
자동복귀용
레지스터
331
332
셀렉터
셀렉터
873
874
870
871
872
834
자동복귀용
레지스터
리드회로
자동복귀용
처리회로
333
334
876
875
811
877
812

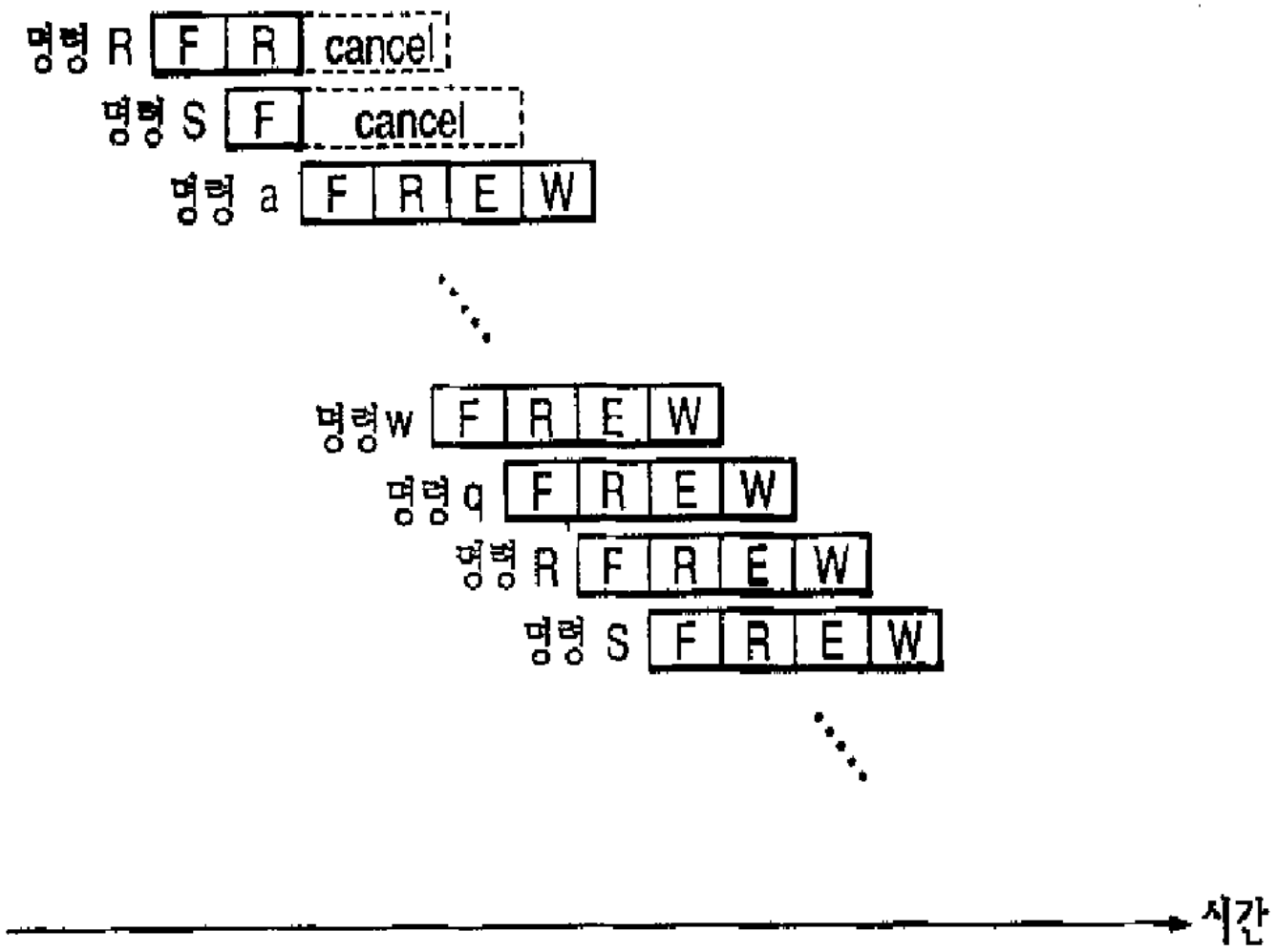
도면 13



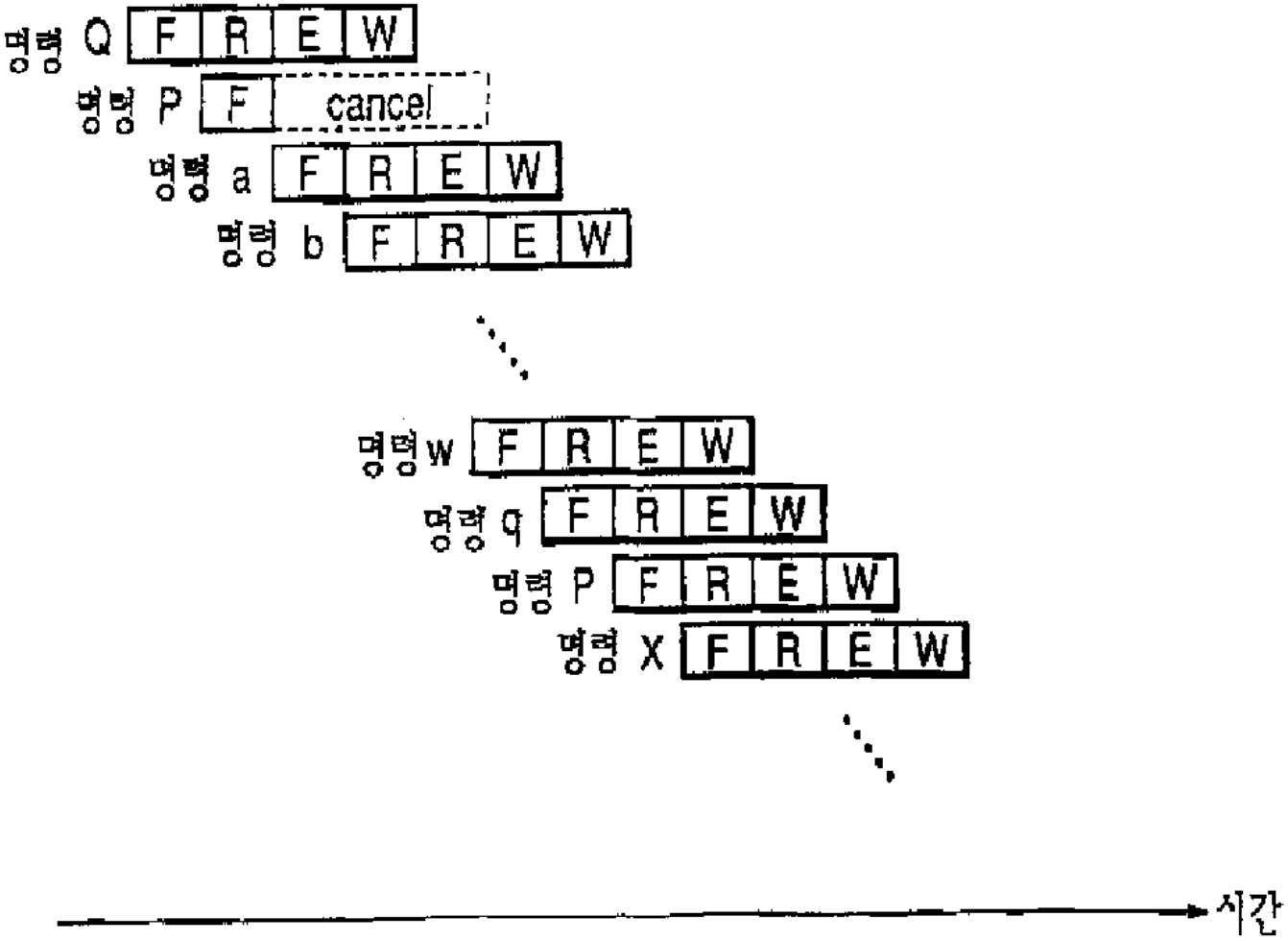
도면 14


명령R
F
cancel
명령 a
F
R
E
W
명령 e
F
R
E
W
명령 w
F
R
E
W
명령 q
F
R
E
W
명령 R
F
R
E
W
명령 S
F
R
E
W
시간

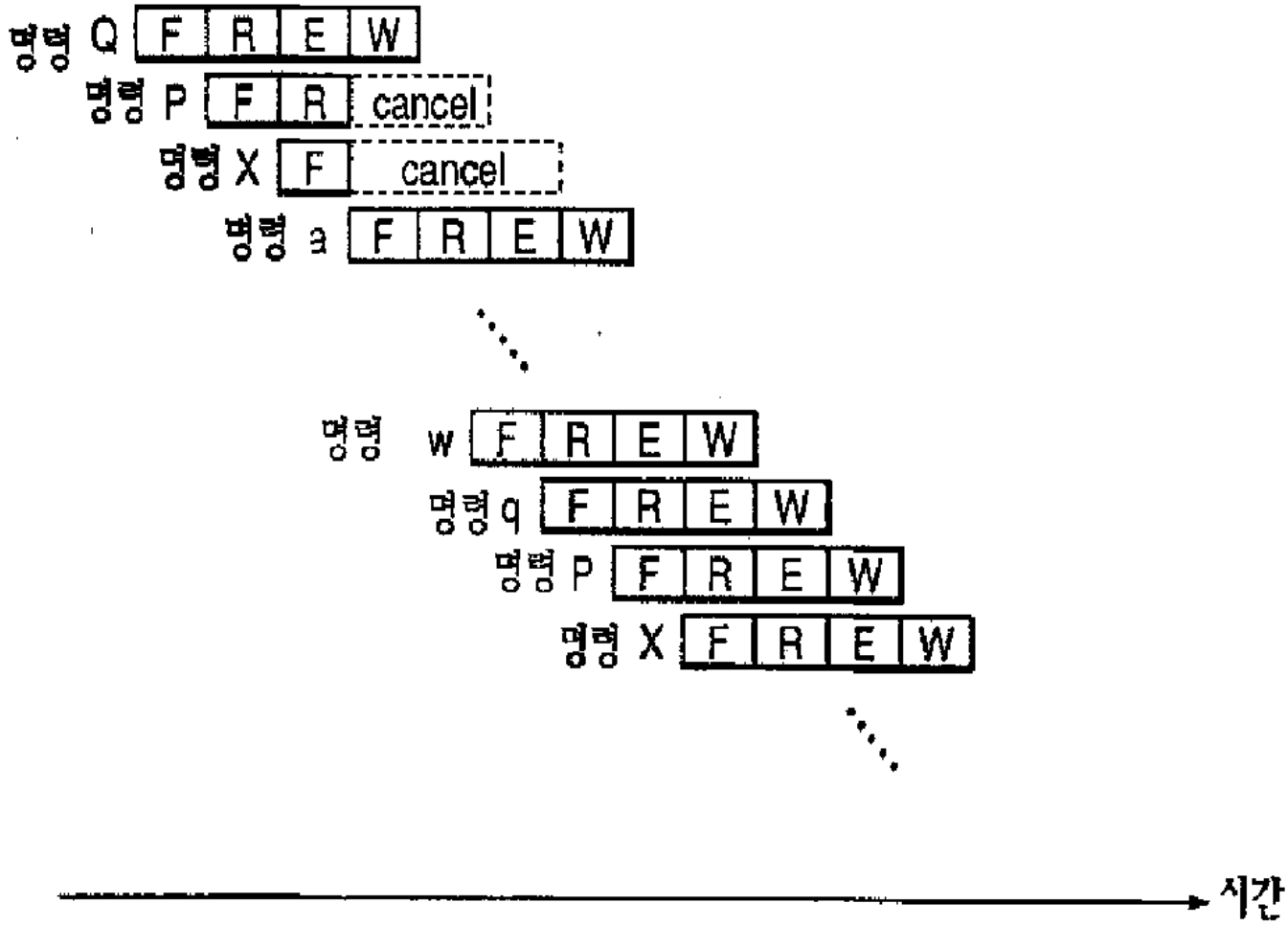
도면 15


명령 R
F
R
cancel
명령 S
F
cancel
명령 a
F
R
E
W
명령w
F
R
E
W
명령 q
F
R
E
W
명령 R
F
R
E
W
명령 S
F
R
E
W
시간

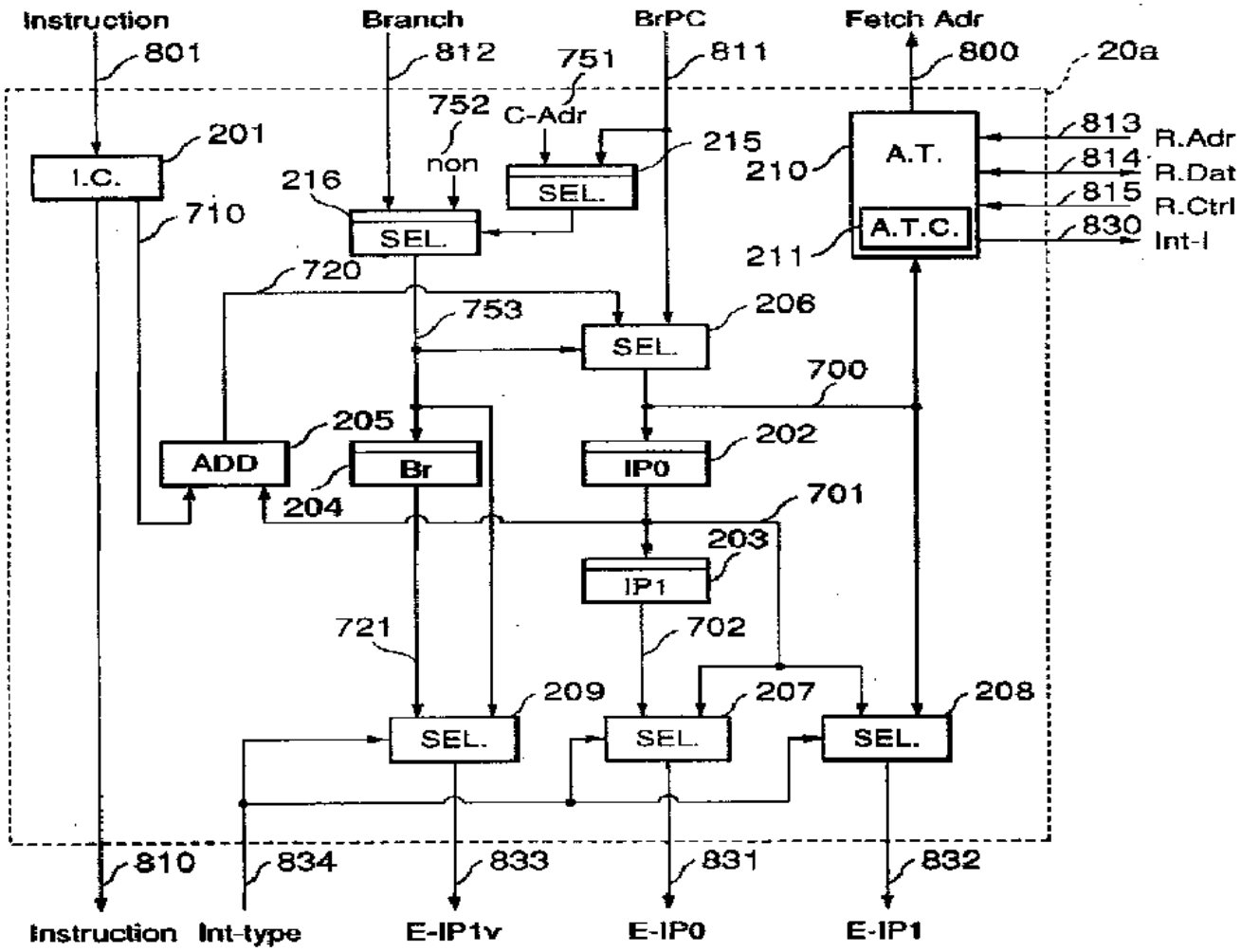
도면 16


명령 Q F R E W
명령 P F cancel
명령 a F R E W
명령 b F R E W
명령 w F R E W
명령 q F R E W
명령 P F R E W
명령 X F R E W
시간

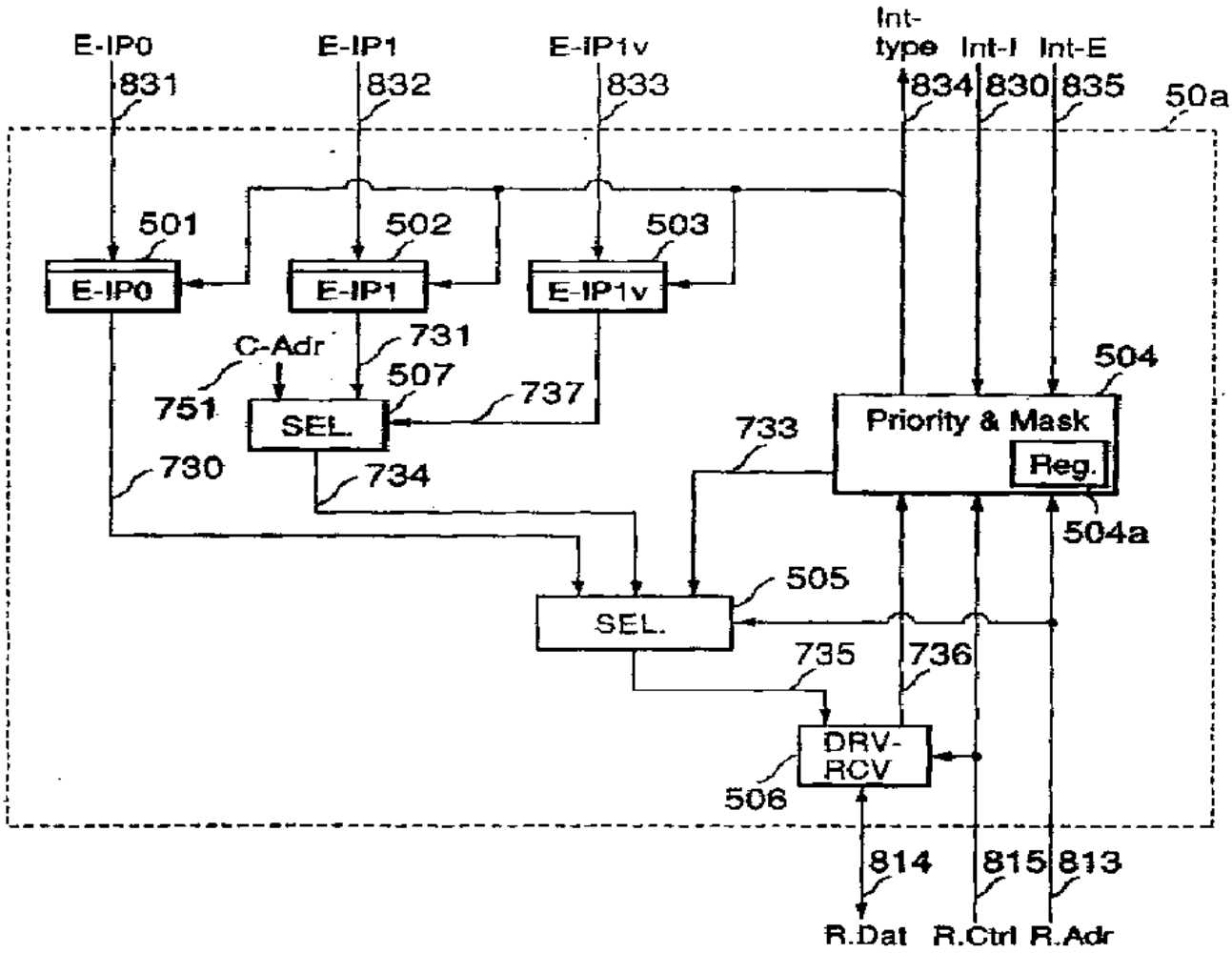
도면 17


명령 Q F R E W
명령 P F R cancel
명령 X F cancel
명령 a F R E W
명령 w F R E W
명령 q F R E W
명령 P F R E W
명령 X F R E W
시간

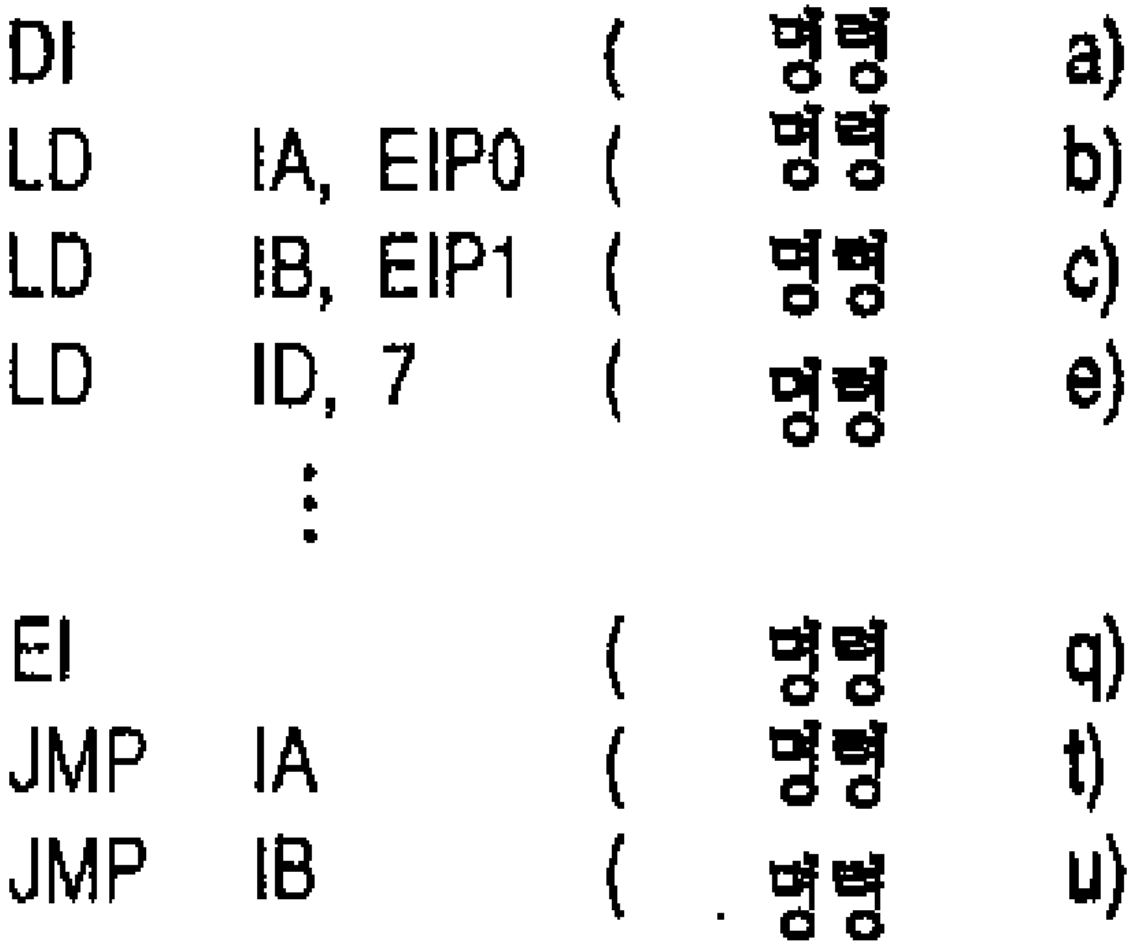
도면 18


Instruction
801
Branch
812
751
BrPC
811
Fetch Adr
800
20a
752
C-Adr
non
201
I.C.
216
SEL.
215
SEL.
210
A.T.
A.T.C.
211
813
R.Adr
814
R.Dat
815
R.Ctrl
830
Int-I
710
720
753
206
SEL.
700
205
ADD
Br
204
202
IP0
701
203
IP1
721
702
209
207
208
SEL.
SEL.
SEL.
810
834
833
831
832
Instruction
Int-type
E-IP1v
E-IP0
E-IP1

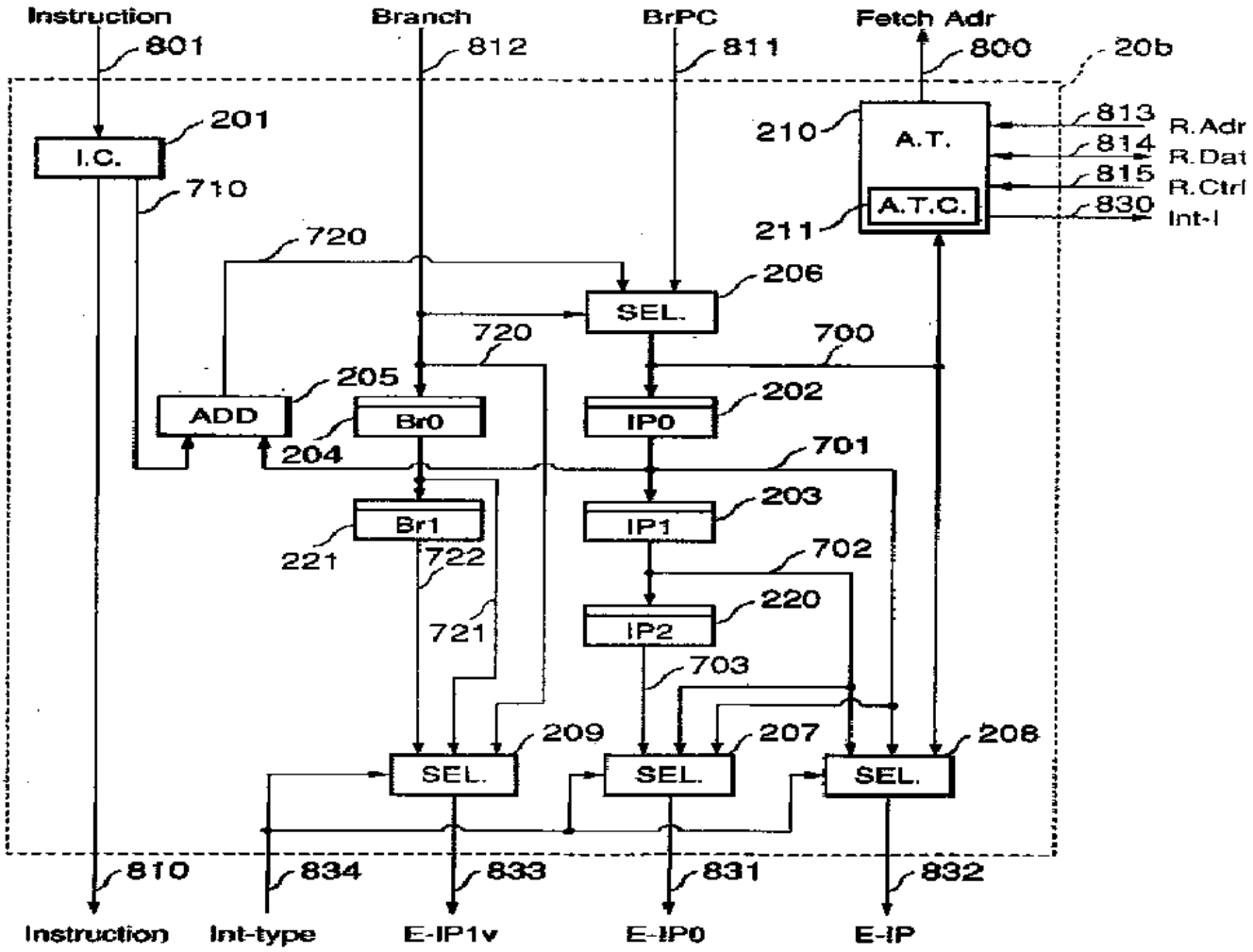
도면 19


E-IP0
831
E-IP1
832
E-IP1v
833
Int-type
834
Int-I
830
Int-E
835
50a
501
502
503
E-IP0
E-IP1
E-IP1v
C-Adr
751
731
507
737
SEL.
730
734
504
Priority & Mask
Reg.
504a
733
505
SEL.
735
736
DRV-RCV
506
814
815
813
R.Dat
R.Ctrl
R.Adr

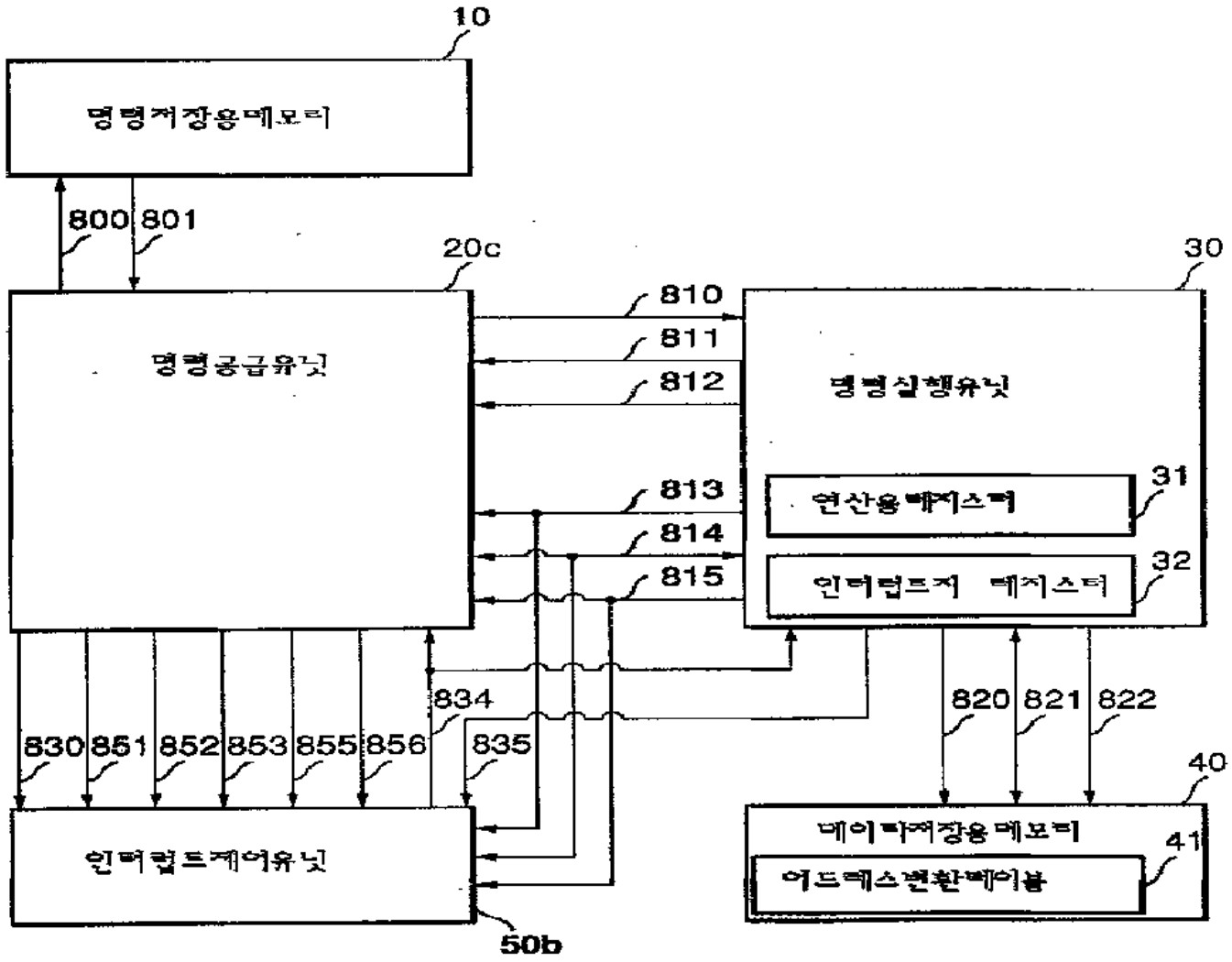
도면 20


DI (명령 a)
LD IA, EIP0 (명령 b)
LD IB, EIP1 (명령 c)
LD ID, 7 (명령 e)
EI (명령 q)
JMP IA (명령 t)
JMP IB (명령 u)

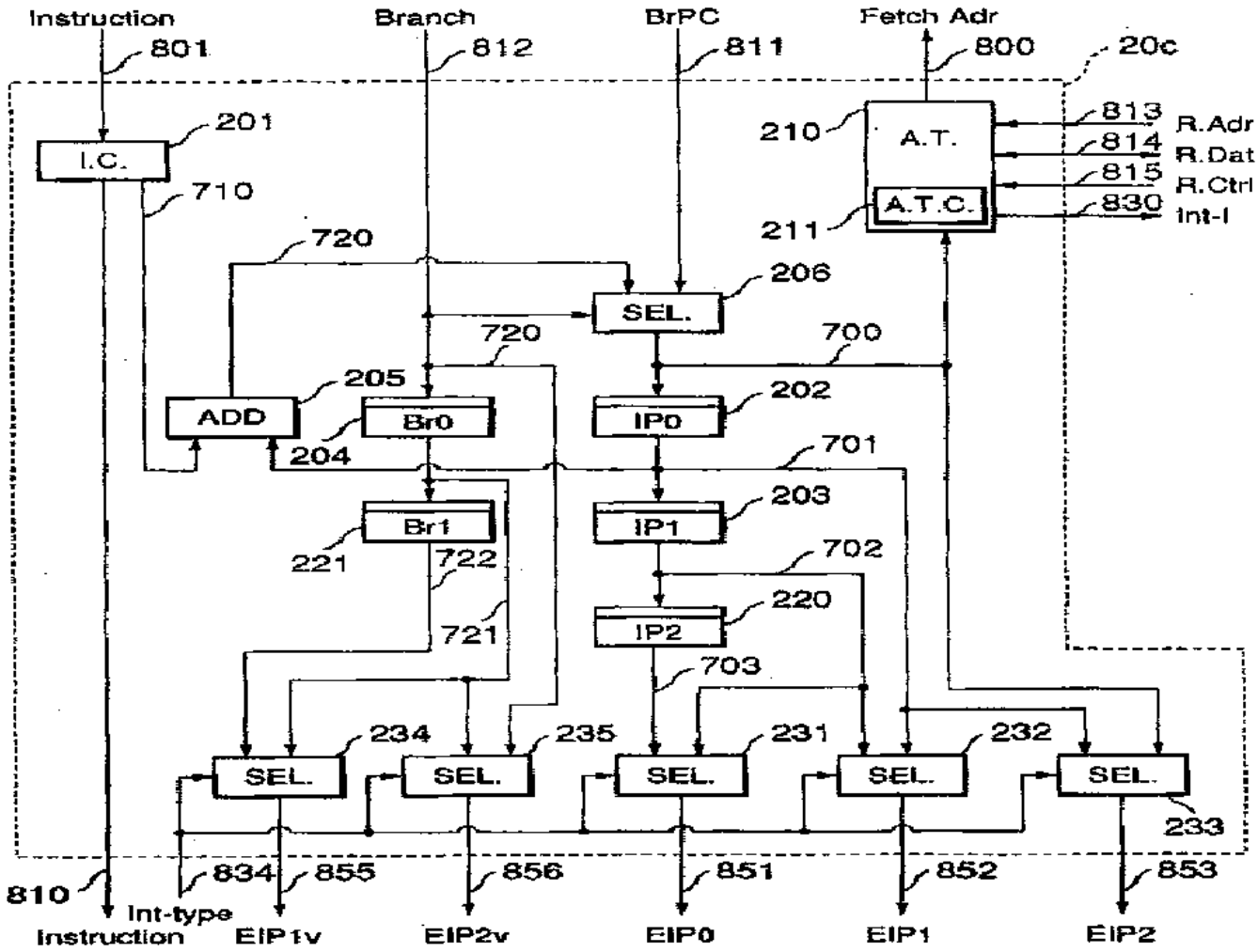
도면 21


Instruction
801
Branch
812
BrPC
811
Fetch Adr
800
20b
I.C.
201
710
210
A.T.
A.T.C.
211
813
R.Adr
814
R.Dat
815
R.Ctrl
830
Int-I
720
206
SEL.
720
700
205
ADD
Br0
204
IP0
202
701
Br1
221
722
IP1
203
702
721
IP2
220
703
209
SEL.
207
SEL.
208
SEL.
810
834
833
831
832
Instruction
Int-type
E-IP1v
E-IP0
E-IP

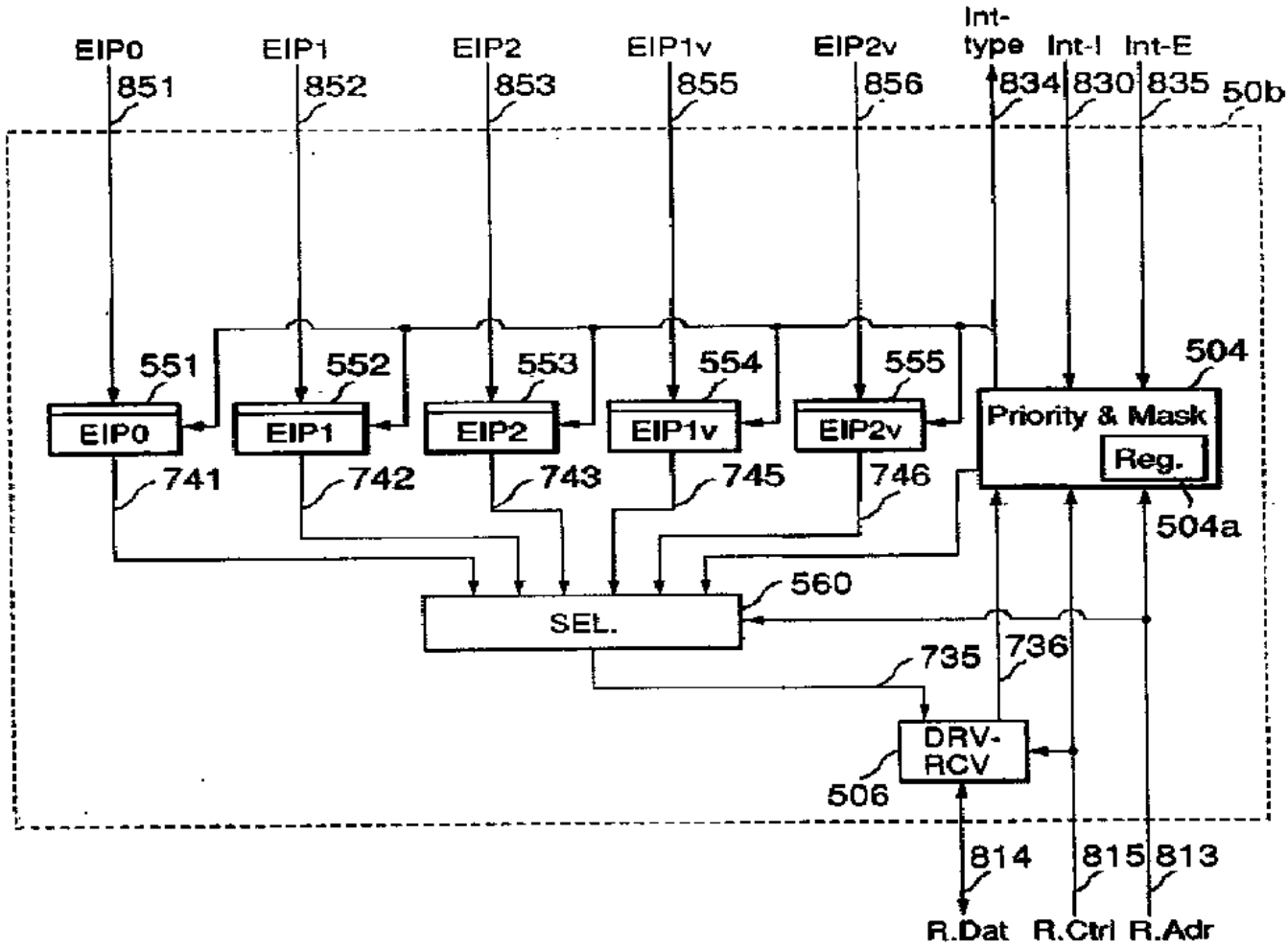
도면 22


10
명령저장용메모리
800
801
20c
명령공급유닛
30
명령실행유닛
810
811
812
813
814
815
연산용레지스터
31
인터럽트처 레지스터
32
834
835
820
821
822
830
851
852
853
855
856
40
데이터저장용메모리
41
어드레스변환테이블
인터럽트제어유닛
50b

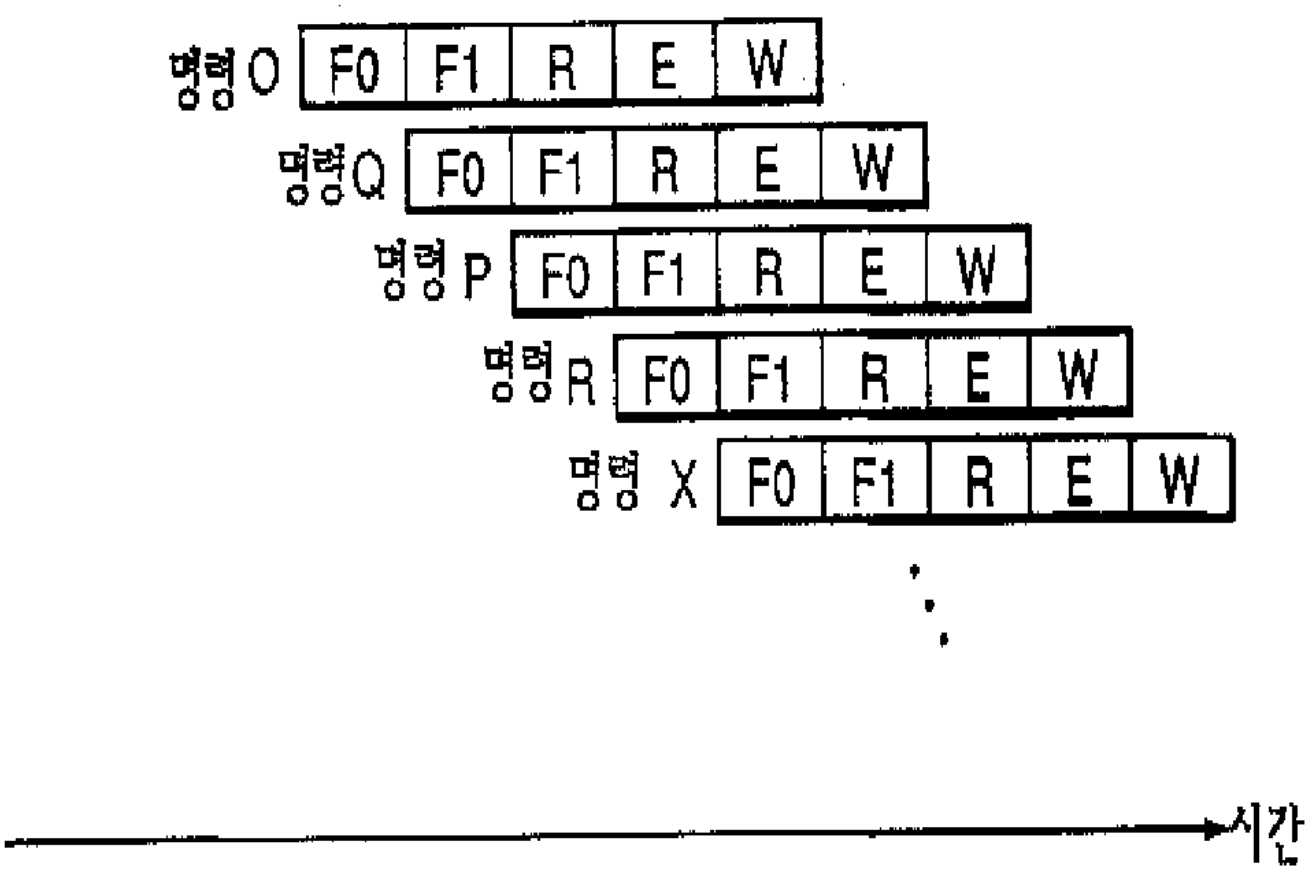
도면 23


Instruction
801
Branch
812
BrPC
811
Fetch Adr
800
20c
I.C.
201
710
720
210
A.T.
A.T.C.
211
813
814
815
830
R.Adr
R.Dat
R.Ctrl
Int-I
206
SEL.
720
700
ADD
205
204
Br0
IP0
202
701
Br1
221
722
IP1
203
702
721
IP2
220
703
234
235
231
232
233
SEL.
810
834
855
856
851
852
853
Int-type
Instruction
EIP1v
EIP2v
EIP0
EIP1
EIP2

도면 24


EIP0
851
EIP1
852
EIP2
853
EIP1v
855
EIP2v
856
Int-type
834
Int-I
830
Int-E
835
50b
551
552
553
554
555
504
EIP0
EIP1
EIP2
EIP1v
EIP2v
Priority & Mask
Reg.
741
742
743
745
746
504a
560
SEL.
735
736
DRV-RCV
506
814
815
813
R.Dat
R.Ctrl
R.Adr

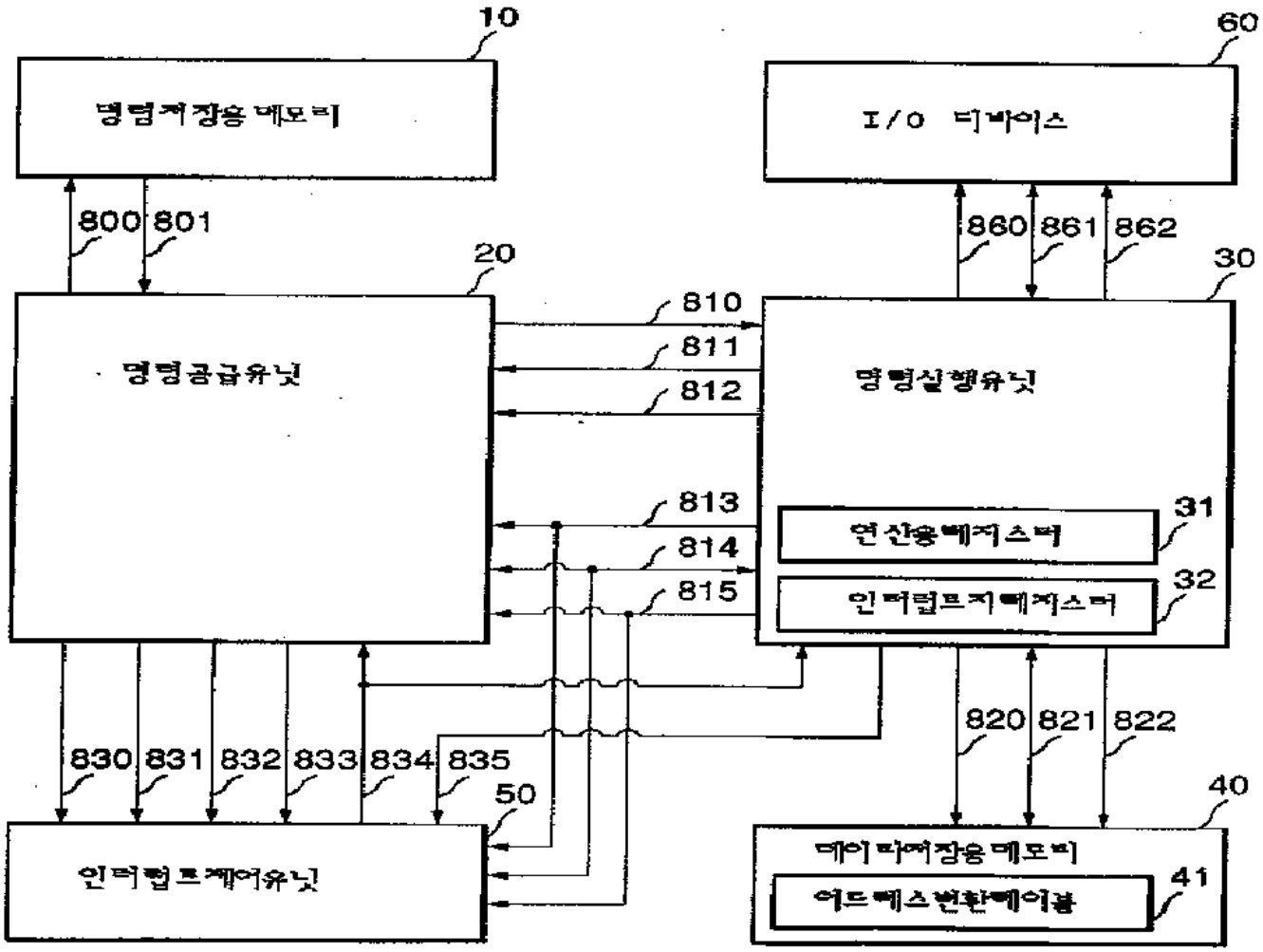
도면 25


명령 O
F0
F1
R
E
W
명령 Q
F0
F1
R
E
W
명령 P
F0
F1
R
E
W
명령 R
F0
F1
R
E
W
명령 X
F0
F1
R
E
W
시간

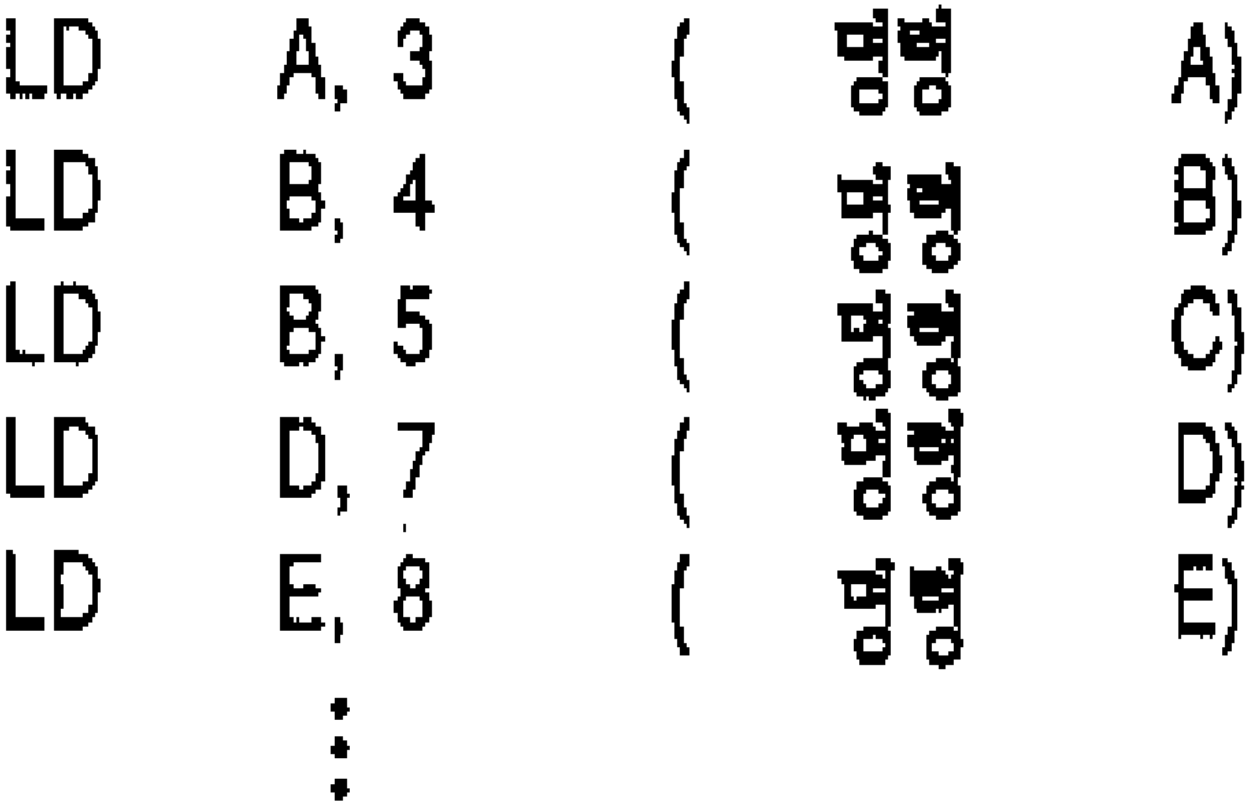
도면 26

```
        DI
        LD      IA, EIP0
        LD      IB, EIP1
        LD      IC EIP2
        LD      ID, EIP1V
        LD      IE, EIP2V
                  :
        CMP  ID, 0
        JNE   L1
        NOP
        NOP
        CMP  IE, 0
        JNE   L01
        NOP
        NOP
        JMP   L00
        NOP
        NOP
L1 :    CMP  IE,0
        JNE   L11
        NOP
        NOP
        EI                  ← ID=1, IE=0 일 때
        JMP   IA
        JMP   IB
        NOP
L11 :   EI                  ← ID=1, IE=1 일 때
        JMP   IA
        JMP   IB
        JMP   IC
L01 :   EI                  ← ID=0, IE=1 일 때
        JMP   IA
        NOP
        JMP   IC
L00 :   EI                  ← ID=0, IE=0 일 때
        JMP   IA
        NOP
        NOP
```

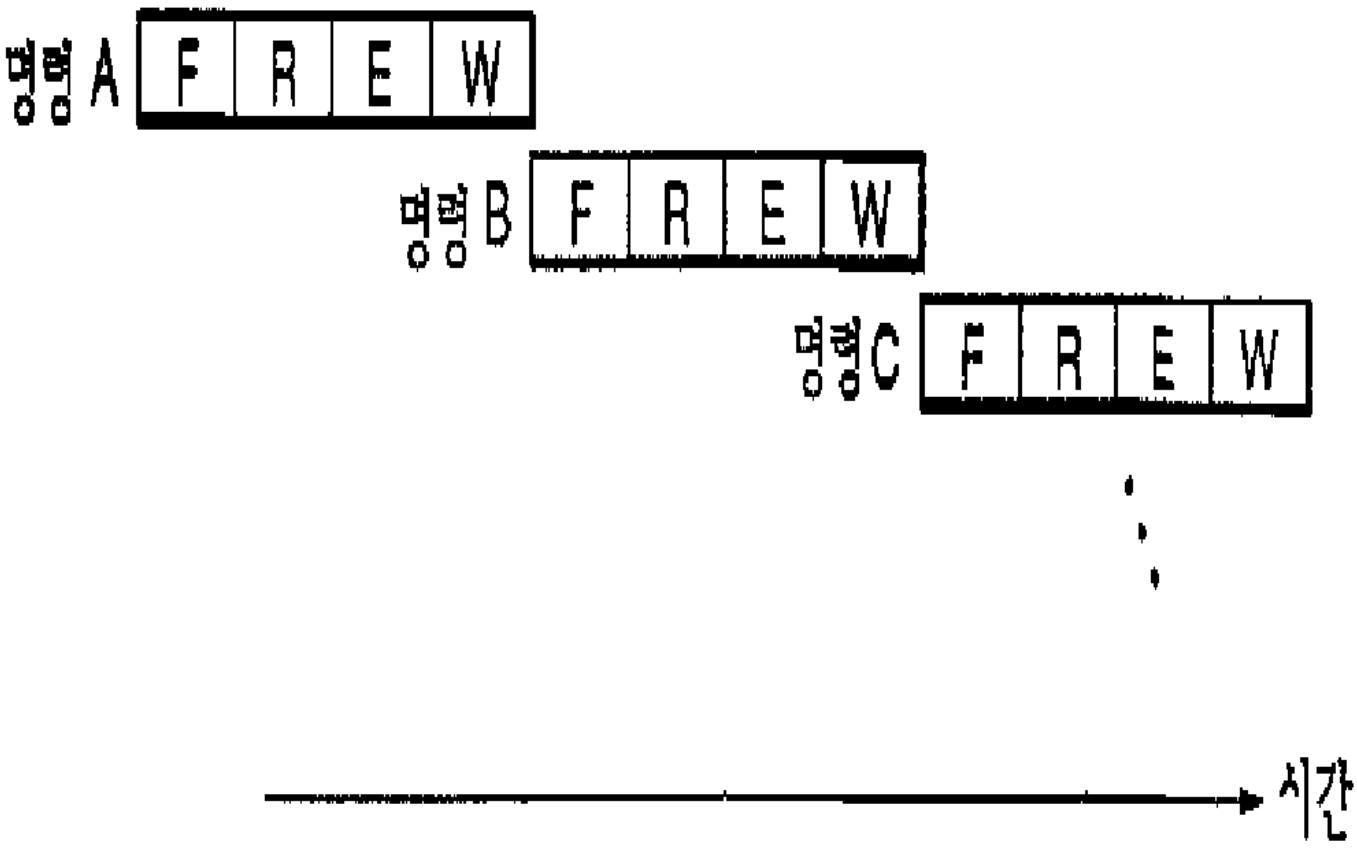
도면 27


10
명령저장용 메모리
60
I/O 디바이스
800
801
860
861
862
20
30
명령공급유닛
명령실행유닛
810
811
812
813
814
815
31
연산용 레지스터
32
인터럽트 레지스터
820
821
822
830
831
832
833
834
835
50
40
인터럽트제어유닛
데이타저장용 메모리
41
어드레스변환테이블

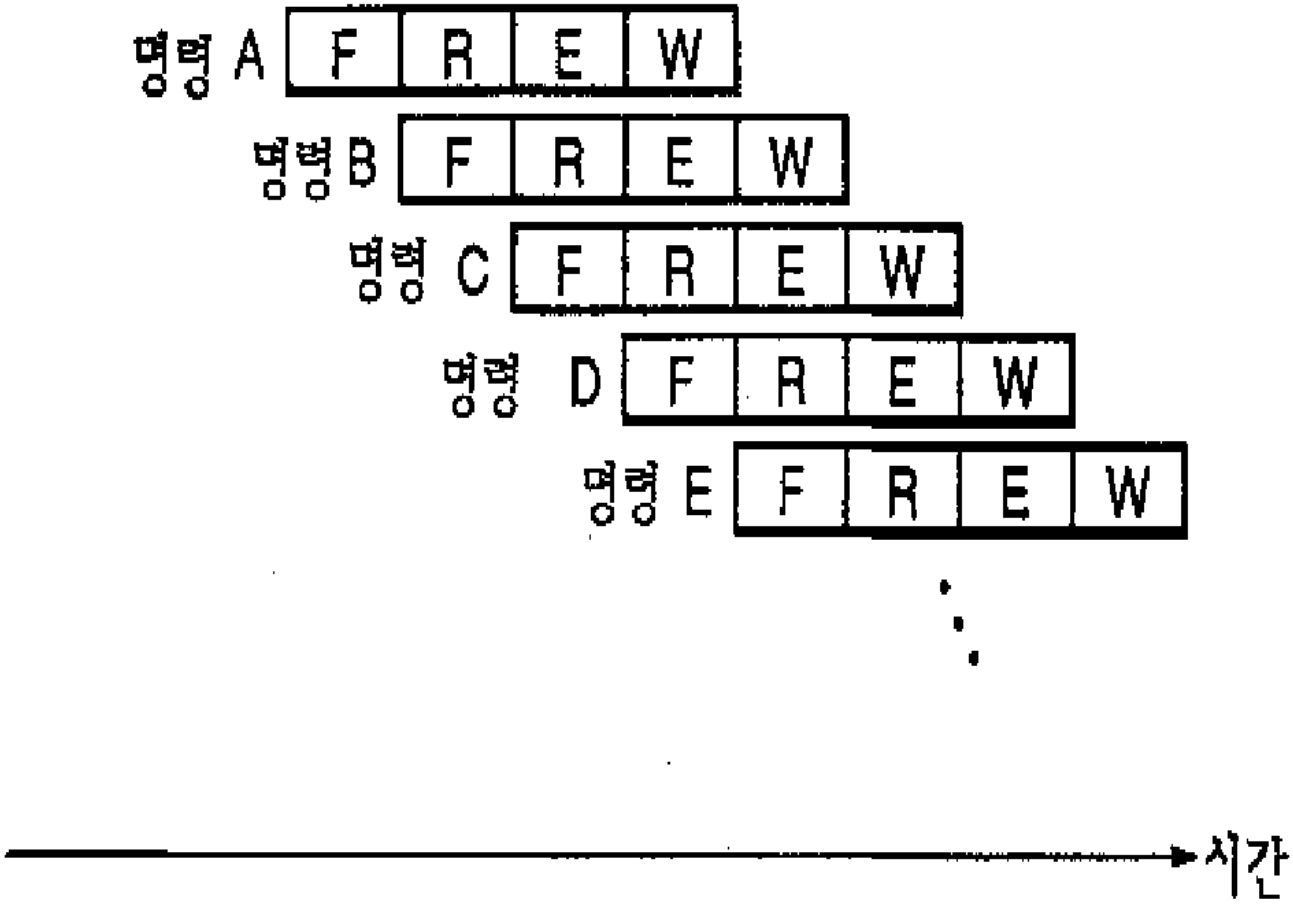
도면 28



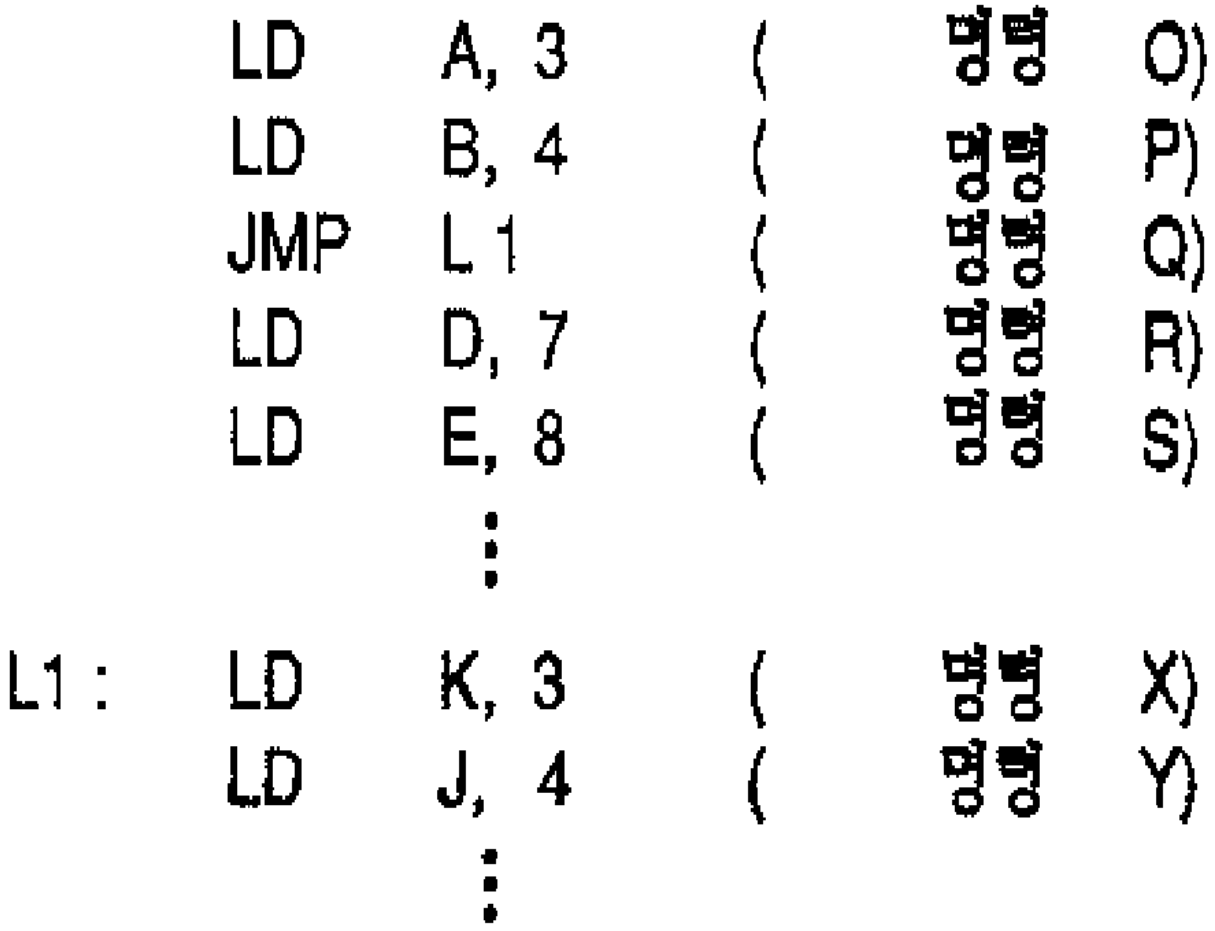
도면 29



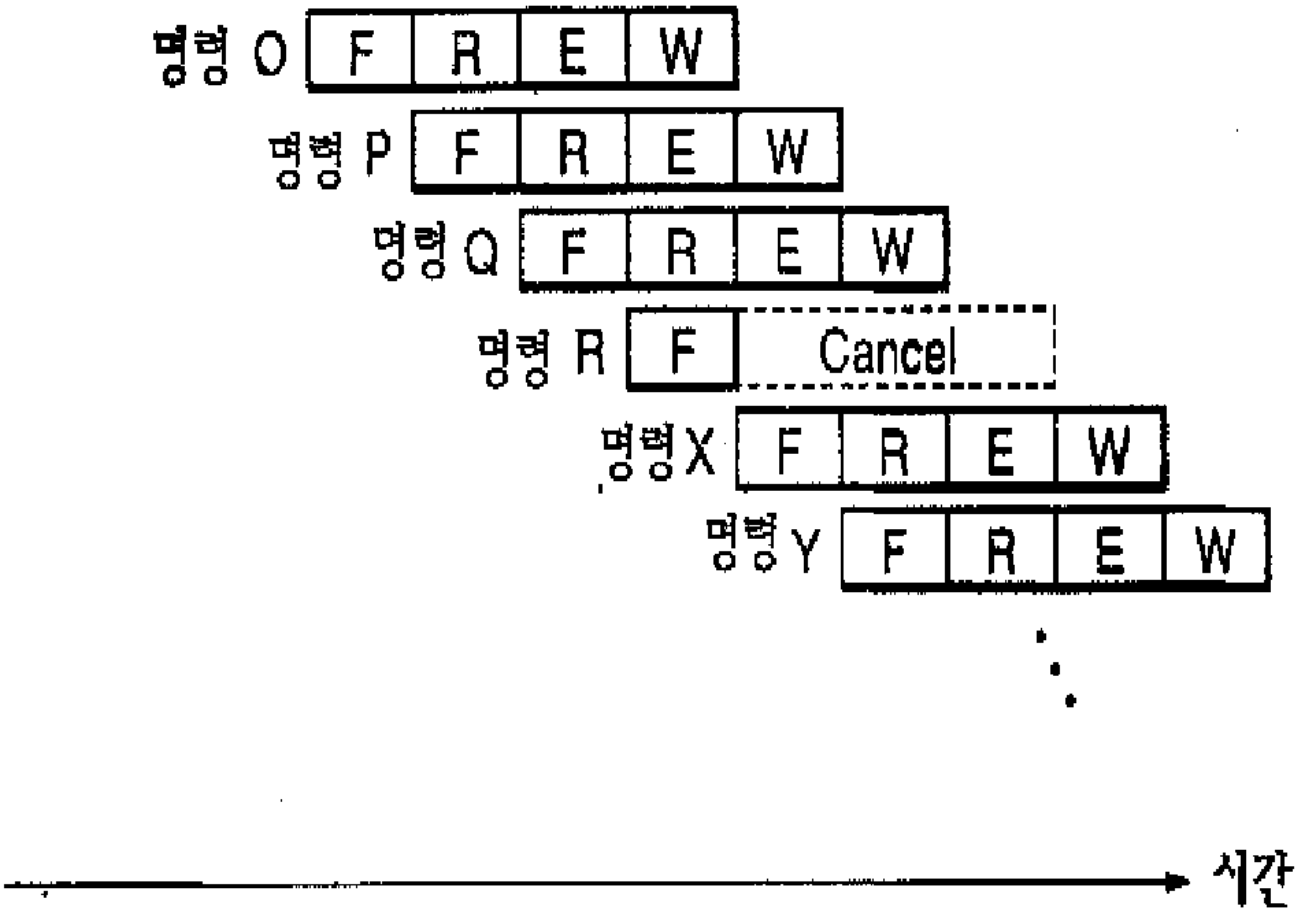
도면 30



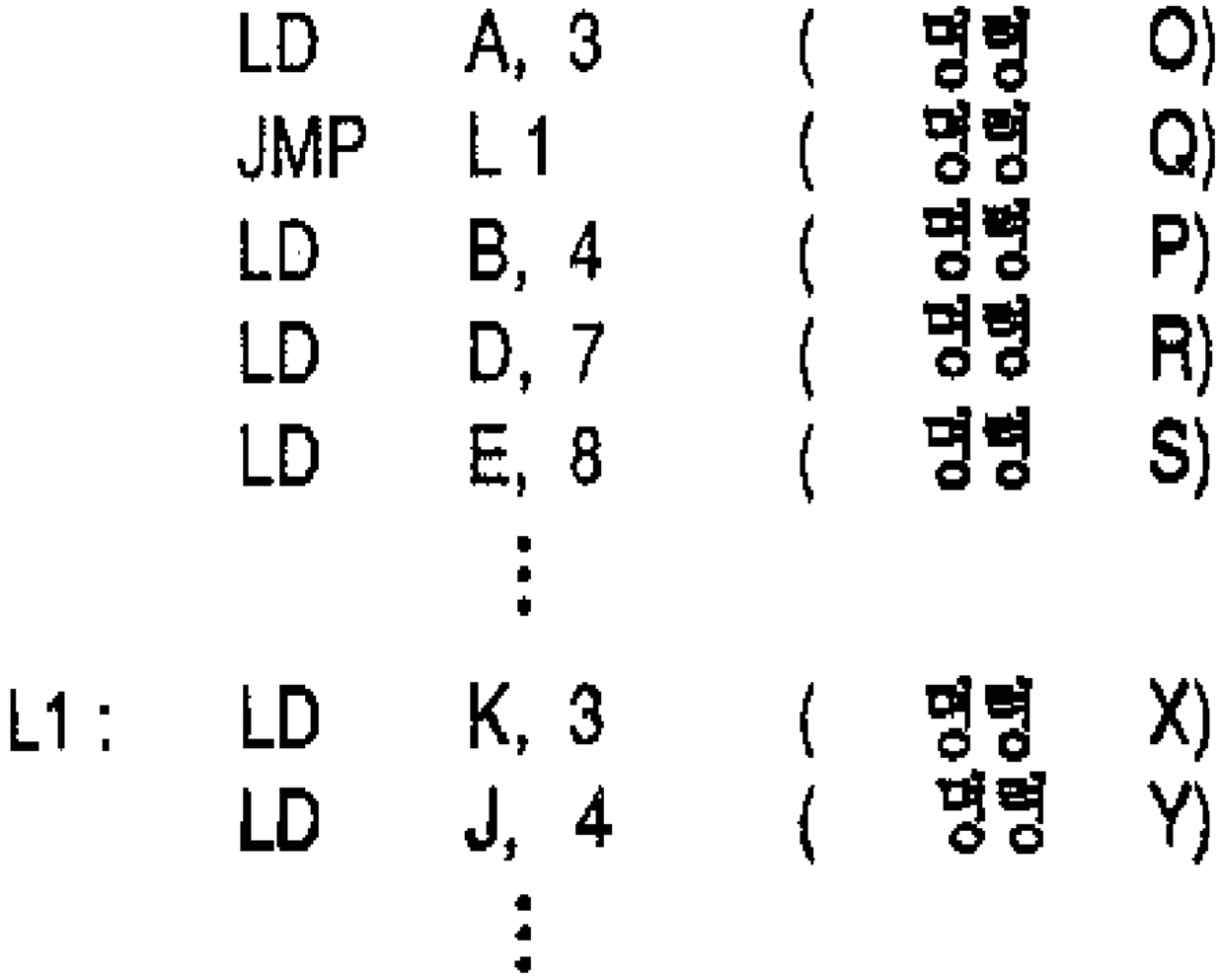
도면 31


LD A, 3 (명령 O)
LD B, 4 (명령 P)
JMP L1 (명령 Q)
LD D, 7 (명령 R)
LD E, 8 (명령 S)
L1 : LD K, 3 (명령 X)
LD J, 4 (명령 Y)

도면 32



도면 33


LD A, 3 (명령 O)
JMP L 1 (명령 Q)
LD B, 4 (명령 P)
LD D, 7 (명령 R)
LD E, 8 (명령 S)
L1 : LD K, 3 (명령 X)
LD J, 4 (명령 Y)

도면 34

