(19) 대한민국특허청(KR)
(12) 특허공보(B1)

(51) Int. Cl.[6]
G06F 9/455

(45) 공고일자 1995년06월19일
(11) 공고번호 특1995-0006616

(21) 출원번호	특1992-0702757	(65) 공개번호	특1993-7000914
(22) 출원일자	1992년11월06일	(43) 공개일자	1993년03월16일
(86) 국제출원번호	PCT/US 92/001715	(87) 국제공개번호	WO 92/15946
(86) 국제출원일자	1992년03월03일	(87) 국제공개일자	1992년09월17일

(30) 우선권주장 666,071 1991년03월07일 미국(US)
(71) 출원인 디지탈 이큅먼트 코오포레이숀 로날드 이. 마이릭
미합중국, 매사추세츠 01754, 메이나드, 메인 스트리트 146

(72) 발명자 로빈슨, 스코트, 지이.
미합중국, 매사추세츠 01879, 팅스보로, 데이비스 로드 42
사이트스, 리챠드, 엘.
미합중국, 매사추세츠 01505, 보일스톤, 워런 스트리트 21
와이테크, 리챠드, 티이.
미합중국, 매사추세츠 01460, 리틀톤, 실버 버취 레인 8
(74) 대리인 나영환, 도두형

심사관 : 나영환 (책자공보 제4016호)

(54) 변환된 프로그램 코드에서 소스 명령 분자를 보전하기 위한 시스템 및 방법

요약

내용 없음.

대표도

도1

명세서

[발명의 명칭]

변환된 프로그램 코드에서 소스 명령 분자를 보전하기 위한 시스템 및 방법

[도면의 간단한 설명]

본 명세서의 일부를 구성하는 첨부 도면은 본 발명의 한 실시예를 예시하고 있으며, 상세한 설명과 더불어 본 발명의 목적, 장점 및 원리에 대한 설명을 제공한다.

제1도는 응용 프로그램 (a.) 제1컴퓨터 시스템(X로서 지정되고 제1명령 세트를 가짐)에 의해 실행을 위해 생성되어, (b.) 비교적 축소된 명령 세트를 갖는 다른 컴퓨터 시스템(Y로 지정됨)에 의한 실행을 위해 본 발명에 따른 X 명령 상태 분자 및 명령 입상을 갖고서 변환되는 일반적인 방식을 나타내는 전체기능 블록도이다.

제2도는 적합한 X-Y 변환 프로그램 및 X-Y 변환 프로그램이 입력 X 응용 코드로부터 실행 가능한 Y 응용 코드를 X 명령 입상(granularity)을 갖고서 발생하도록 실행되도록 하는 범용 컴퓨터 시스템에 대한 일반화된 기능 블록도이다.

제3도는 제2도에 도시된 X-Y 변환 프로그램에 대한 일반적 순서도이다.

제4도는 명령 입상 제어(IGC) 프로그램, 특권 아키텍춰 라이브러리(PAL) 코드 루우틴 및 본 발명에 따른 명령 입상 및 상태 분자의 보전성을 갖고서 Y 코드 실행을 제어하기 위해 저장된 로드록 조건부 시퀀스와 함께 변환된 또는 Y 코드가 적재된 Y 컴퓨터 시스템의 기능 블록도이다.

제5a도 및 제5b도는 IGC 프로그램에 수행된 적합한 기능 단계를 나타내는 순서도이다.

제6도는 X-Y 코드 명령 변환 및 그에 대한 비동기 사건의 관계를 심볼화한 도면이다.

제7도는 다중 메모리 기입의 경우에 상태 분자 및 명령 입상을 보전하도록 본 발명에 따라 이용되는 방식을 나타내는 제4도의 PAL 코드 루우틴을 상세히 도시한 도면이다.

제8도는 부분적 기입 및 인터록 갱신 명령의 경우에 상태 분자 및 명령 입상을 보전하도록 본 발명에 따라 이용되는 방식을 나타내는 제4도의 로드록/저장-조건부 시퀀스를 표시하는 순서도이다.

[발명의 상세한 설명]

[발명의 배경]

본 발명은 상이한 컴퓨터 시스템의 실행을 위해 프로그램 코드를 적응시키기 위한 시스템 및 방법에 관한 것으로, 특히 명령 분자 상태(instruction state-atomicity)를 보전하면서 하나의 명령 세트에 근거한 코드들을 비교적 축소된 명령 세트에 근거한 코드들로 변환시키기 위한 시스템 및 방법에 관한 것이다.

초창기의 컴퓨터 프로그래밍에서는, 컴퓨터 프로그램용 명령이 마이크로 코드 레벨로 발생되었다. 소프트웨어 기술의 진보 및 성장에 따라, 명령 복잡도(instruction complexity)를 위해 설계된 하드웨어 아키텍춰를 갖는 컴퓨터에 의해 실행될 수 있는 단일 복합 명령에 더 많은 작업(task)이 조합되었다.

명령 복잡도의 증가는 컴퓨터 하드웨어 비용 및 성능의 개발 환경면에서 가격/성능의 이익을 증가시켰다. 그 결과, 복합 명령 세트 코드(CISC)가 널리 인정받게 되었다.

그러나, 명령 복잡도의 증가에 따라, 고속 실행을 위한 시스템 하드웨어를 설계되는 것이 더욱 어려워졌다. 그대신, 서로 관계된 축소 명령 세트 코드(RISC) 컴퓨터 하드웨어 아키텍춰와 결합되는 RISC와 상당히 개선된 시스템 가격/성능을 얻게 하는 메카니즘으로서 인정받게 되었다.

RISC 시스템은 일반적으로 소정의 동작을 지시하는데에 보다 간단한 기본 명령을 사용한다. 단일 RISC 명령은 정상적으로는 기껏해야 단일 액세스 메모리를 갖는 단일 동작을 특정한다. 더우기, RISC 시스템은 정상적으로 각 기본 명령에 대한 레지스터를 제공한다. RISC 명령 세트중의 명령들은 여전히 마이크로 코드 레벨 이상으로 된다.

전형적인 CISC 시스템에서는, 단일 명령이 복잡한 동작 순서를 특정할 수 있고 또 메모리에 대한 다수의 직접 액세스를 행할 수 있다. 따라서, CISC 명령에 의해 수행된 동작은 수개의 RISC 명령을 필요로 할 수도 있다.

RISC 시스템은 일반적으로 이용 가능한 하드웨어 가격 및 성능에 대하여 보다 신속한 시스템 동작, 보다 나은 전체 시스템 성능 및 보다 낮은 시스템 가격을 제공하는 최적화된 하드웨어 및 소프트웨어의 타협(tradeoff)으로 설계된다.

CISC 시스템의 RISC 시스템으로의 변환에 있어 한가지 장애 요소는 CISC 시스템을 위해 개발되었으나 RISC 시스템에는 일반적으로 이용될 수 없는 대규모 소프트웨어 라이브러리(libraries)의 존재이다. 컴퓨터 시스템 사용자가 새로운 컴퓨터 시스템을 얻는 것을 선택한 경우, 사용자의 주요 고려사항중 하나는 사용자의 응용 프로그램 라이브러리가 새로운 컴퓨터 시스템에서의 사용을 위해 변환될 수 있는가의 여부 및 그 라이브러리를 교환하는데 드는 가격이다. 따라서, RISC 컴퓨터 시스템을 통해 보다 나은 가격/성능을 얻고자 하는 컴퓨터 시스템 사용자의 경우, RISC 컴퓨터 시스템의 실행을 위한 사용자의 응용 프로그램 라이브러리를 적응(adapting) 또는 “이동”(migrating)시키기 위한 경제적이고 효과적인 메카니즘이 제공되어야 하는 것이 대단히 중요하다.

프로그램 이동을 위해 사용자에게 몇가지 선택이 가능하다. 재번역 또는 재코드화 기술이 이용될 수 있지만, 이러한 기술은 상세한 기계 독립성을 갖지 않거나 혹은 매뉴얼 프로그래밍 수정에 의해 제거된 기존의 기계 독립성을 갖는 FORTRAN과 같은 고레벨 언어로 쓰여진 프로그램을 이동시키는데 사용된다. 또한, 재번역 또는 재코드화시, 사용자가 전형적으로 프로그램 수정 및 프로그램의 절차 보장에 관한 모든 책임을 갖게 된다.

이와는 달리 해석 절차(interpretation procedure)가 사용될 수 있지만, 이 방법의 단점은 실질적으로 프로그램 성능이 저하된다는 점이다. 구체적으로, 해석 절차는 하나의 컴퓨터를 동작시켜 주제 명령(subject instruction)의 스트림(다른 형태의 컴퓨터에 대한 명령들임)을 읽어내서 각 주제 명령에 대하여 지시된 동작을 수행하는 소프트웨어 프로그램이다. 이러한 절차는 전형적으로 단일 주제 명령을 해석하도록 하나의 컴퓨터에 대하여 10 내지 100개의 기계 명령을 실행한다. 따라서, 해석 절차는 하나의 컴퓨터에 대하여 기능적으로 동등한 코드의 실행을 지시하는 것에 비해 실질적으로 프로그램 성능이 저하되게 한다.

그러나, 가장 효율적이고 유효한 이동은 코드 변환이다. 코드 변환시, 기존의 프로그램으로부터의 각 명령은 행선 기계(destination machine)의 언어로 하나 또는 그 이상의 명령으로 번역된다. 따라서, CISC 프로그램의 RISC 프로그램으로의 변환, 또는 보다 일반적으로 변환된 코드가 비교적 축소된 명령 세트를 갖도록 하는 프로그램 변환은 변환중인 코드의 각 명령에 대하여 변환된 코드의 “다중” 또는 “다수의” 명령을 필요로 한다.

그러나, “하나 대 다수” 또는 CISC 대 RISC의 코드 변환을 행함에 있어, 원래 CISC 또는 다른 비교적 복잡한 코드로써 제공된 다수의 명령 행위 보증을 보전하는 것은 일반적으로 어려운 일이다.

변환시 어떤 어려움을 나타내는 정상적인 CISC 보증의 하나는 다른 CISC 명령 또는 그것의 일부도 단일 CISC 명령의 시작과 종료 사이에서 실행될 수 없는 요건이다. 따라서 CISC를 RISC로 변환함에 있어, 이러한 타입의 명령 전체 또는 입상이 보전되어야 하는 것이 필수적이다. 명령 입상의 보전은 상태 또는 메모리 분자가 보전되는 것을 필요로 한다. 따라서, 한 명령에서 모든 메모리 액세스가 발생되어야 하거나 혹은 어떠한 것도 변환된 코드에서 발생되지 않아야 한다.

변환 과정에서 명령 입상을 보전하기 위해서는, 각각의 보다 복잡한 명령에 대응하는 변환된 명령의

각 세트 또는 “입상”(granule)이 대응하는 보다 복잡한 명령이 생성하였던 것과 동일한 결과를 생성하도록 실행할 것이라는 보장이 있어야 한다. 이는 보다 간단한 변환된 명령의 “입상”중 어떤 것의 실행시 비동기적 사건이 발생할 수도 있더라도 그러하여야만 한다.

상기 동일자로 출원된 관련 출원 제(1870-0410)호는 코드 변환 및 변환된 코드의 실행시 명령 입상의 보전을 제공하는 발명에 관한 것이다. 또한 관련 상기 출원은 일반적으로 상태 분자를 얻으며 구체적으로 변환되어질 CISC 또는 다른 코드의 한번 기입 명령(one-write instruction)의 경우에 상태 분자를 확인하기 위한 메카니즘 및 절차에 관해 기재하고 있다.

한번 기입 명령은 예외 상황에 직면할 수 있는 최대로 하나의 상태 기입을 포함하지만, 제한되지 않은 예외 상황 없는 상태 기입의 수를 포함할 수도 있다. 두 종류의 상태 기입 사이에는 오버랩(overlap)이 존재하지 않는다.

여기서 “예외 상황”(exception)이란 용어는 명령 실행의 계속을 막는 어떤 조건을 의미한다. 전형적인 예외 상황은 다음을 포함한다.

a. 페이지 오류 또는 액세스 위반과 같은 메모리 예외 상황

b. 부동점 오우버플로우 또는 0에 의한 제산과 같은 연산 예외 상황

c. 부적법한 연산 코드 또는 정지점 명령과 같은 명령 예외 상황

상태분자는 상기 1870-0410호의 특허 출원에 따라 변환 및 실행된 한번 기입 명령 경우의 명령 입상과 동등하다. 그러나, 변환되어야 할 다른 종류의 명령들은 상태 분자 및 명령 입상의 보전을 달성하는데에 특수한 문제점을 나타내는 시퀀스들을 포함한다. 이러한 명령들은 다음의 것을 갖는 명령을 포함한다.

a. “인터록”(interlock)되는 판독-수정-기입 시퀀스 또는 부분적-메모리-워드-기입을 필요로 하고 다른 프로세서에 의한 기입을 개입시키지 않고 멀티프로세서 시스템에서 실행되어야 하는 판독-수정-기입 시퀀스

b. 예외 상황에 직면할 가능성이 있고 모두 발생되거나 모두 발생되지는 않아야 하는 다수의 상태 기입 시퀀스

이러한 특별한 경우에는, 변환된 코드의 상태 분자 및 명령 입상을 얻도록 시도하는 동안 직면하게 되는 특수 상황을 어드레스하기 위한 보다 특별한 메카니즘 및 절차가 필요하다. 다수의 상태 기입을 갖는 명령의 경우, 상태 문자 문제는 비동기적 사건이 모든 상태(메모리 또는 레지스터) 기입이 실행되기 전에 적어도 하나후의 명령 시퀀스의 실행 동안 발생된다. 따라서, 명령 입상 실행이 중지 또는 계속되면 반전 불가능한 상태 변화가 이미 실행된 한번의 상태 기입으로 이미 발생되었기 때문에 상태 에러가 생성될 수 있도록 시퀀스에서 실행되기 위해 남아있는 변환된 코드 명령들중 하나에서 예외 상황이 발생할 수 있다.

공통 메모리를 갖는 다중 프로세서를 구비한 시스템에서 변환된 코드를 실행하는 경우에, 변환된 코드가 실행중인 제1프로세서가 부분적으로 명령 입상의 판독-수정-기입 시퀀스를 실행할 수도 있지만 이어서 판독-수정-기입 시퀀스가 완료되기 전에 다른 프로세서가 판독-기입-수정 시퀀스에 의해 어드레스된 상태 기억 장소에 기입될 수도 있기 때문에 상태 분자 문제가 일어날 수 있다. 또한, 상태 입상 명령이 다른 프로세서에 의한 경합하는 상태 호출 이후에 정지 또는 계속되면, 반전 불가능한 상태 변화가 이미 발생하였기 때문에 상태 에러가 생성될 수도 있다.

따라서, 본 발명은 코드들이 다수의 또는 부분적인 기입, 인터록 명령 및 멀티프로세서 실행 환경과 같은 특수 상황과 관련된 명령들을 포함하는 경우 명령 입상과 상태 분자를 보전하면서 보다 복잡한 명령 세트를 갖는 기존의 코드들로부터의 비교적 축소된 명령 세트들을 갖는 변환 코드들을 생성 및 실행하기 위한 구조 및 절차에 관한 것이다. 따라서, 본 발명은 그러한 특수 상황이 나타나는 경우에도 응용 코드 대상을 보전하면서 컴퓨터 시스템의 가격/성능 개선이 실현될 수 있게 한다.

[발명의 개요]

제1프로그램 코드를 제2프로그램 코드로 변환하고 제1코드의 명령 상태 분자를 보전하면서 제2프로그램을 실행하기 위한 시스템 및 방법이 제공된다. 제1프로그램 코드는 제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터에서 실행될 수 있고, 제2프로그램 코드는 메모리 및 레지스터 상태와 제1명령 세트에 비해 축소된 제2명령 세트에 적합한 제2아키텍춰를 갖는 컴퓨터에서 실행될 수 있다.

제1컴퓨터는 제2코드 명령의 관점에서 제1코드 명령을 정의하는 패턴 코드에 따라서 제1코드 명령을 대응하는 제2코드 명령으로 변환한다. 각각의 제1코드 명령에 대한 제2코드 명령은 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하는 제2코드 명령을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실현하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령들을 갖는 제2그룹을 순서적으로 갖는 입상 명령 시퀀스로 조직된다.

제1특수 기입 명령은 제1시퀀스가 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의해 어떠한 인터럽트 없이 제1메모리 장소에의 기입에 어떠한 경합도 개입되지 않고 실행되어야 한다는 조건에 따라서 제1메모리 장소에의 한번의 기입을 처리하기 위해 제1시퀀스를 포함하는 구조로 된다. 또한, 제2특수 기입 명령은 어떠한 인터럽트 없이 모두 실행되어야 하는 다수의 기입을 처리하기 위해 제2시퀀스를 포함하는 구조로 된다.

제2컴퓨터 시스템은 제2프로그램 코드를 실행하도록 제2아키텍춰로써 적응된다. 제2코드 실행시, 제2코드 실행동안 각 비동기 사건의 발생을 판별하고 또 메모리 상태에 결합되면 다른 프로세서에 의한 제1메모리 장소로의 각각의 경합하는 기입의 발생을 판별하기 위한 수단이 제공된다.

어떤 입상 제2코드 명령 시퀀스는 모든 제1그룹 명령이 실행되기 전에 시퀀스 실행시 비동기 사건

인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행 이전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 상태 분자 및 제1코드 명령 입상을 보전하도록 재시도를 위해 중지되며, 그에 따라 차후의 비동기 사건 처리가 가능하게 된다.

제1시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스의 제1특수 명령 서브 시퀀스는 제1서브 시퀀스의 실행 완료전에 다른 프로세서에 의해 경합하는 기입이 행하여지는 경우 성공적인 실행이 완료될 때까지 재시도를 위해 중지된다. 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스는 제1서브 시퀀스의 의도된 실행 동안 비동기 사건 인터럽트가 발생하는 경우 재시도를 위해
중지된다.

A) 제2서브 시퀀스가 입상 명령 시퀀스에 포함되고 제2명령 서브 시퀀스의 실행동안 제1기입후 비동기 사건 인터럽트가 발생하는 경우, 혹은 B) 가능한 예외 상황으로 되는 제2그룹의 모든 상태 갱신 명령의 실행후 비동기 사건 인터럽트가 발생하는 경우, 비동기 사건 인터럽트의 처리는 지연되고 실행중인 어떤 입상 제2코드 명령 시퀀스가 완료된다.

[양호한 실시예의 설명]

제1도에 도시한 바와 같이, 소스 코드로 쓰여진 응용 프로그램(10)은 X 컴퓨터 시스템(12)에서의 실행을 위해 사용자 프로그램 라이브러리에 담겨진 다수의 응용 프로그램중 하나이다. X 컴퓨터 시스템(12)에 대한 하드웨어 아키텍춰는 프로그램(10) 또는 사용자 라이브러리내의 다른 응용 프로그램의 실행 가능한 형태를 생성해내는데 사용되는 X 명령 세트와 동작하기에 적합하다.

Y 컴퓨터 시스템(20)과 사용하기 위한 프로그램(10)을 적용시키기 위해서, 프로그램(10)의 실행 가능한 형태가 Y 컴퓨터 시스템(20)의 하드웨어 아키텍춰가 적응되어지는 Y 명령 세트를 사용하여 Y 실행 가능 코드(22)로서 제공되는 것이 필요하다.

Y 명령 세트는 일반적으로 X 명령 세트가 사용하는 것 보다 소수의 기본 명령을 사용하며 X 코드의 Y 코드로의 변환은 하나 내지 다수의 명령 변환을 필요로 한다. X 명령 세트는 CISC 명령 세트일 수 있고 Y 명령 세트는 RISC 명령 세트일 수 있다. 예를들어, 제1도에 예시 목적을 위해 상세히 나타냈지만, X 시스템은 VAX? 아키텍춰를 사용하고 Y 시스템은 VAX? RISC 아키텍춰를 사용할 수 있는데, 이 양 아키텍춰는 본 출원의 양수인인 Digital Equipment Corporaton에 의해 공급된다.

제1도에 도시한 바와 같이, 응용 프로그램(10)은 간접 통로(24) 또는 직접 통로(26)에서 Y 실행 가능 코드(22)로 이동될 수 있다. 직접 이동은 Y 컴파일러(28) 및 Y 링커(30)을 사용하여 얻어진다. 그 결과의 Y 실행 가능 코드는 참조 번호(22B)로 나타낸다.

Y 컴파일러(28) 및 Y 링커(30)가 개발되지 않았거나 이용될 수 없다면, 혹은 관련 단점 때문에 직접 이동 통로(26)를 사용하는 것을 사용자가 선택하지 않았다면, 간접 통로(24)가 프로그램 투자 절약 및 시스템 성능 이득을 달성하도록 X 응용 프로그램을 Y 시스템으로 이동시키는데 사용될 수도 있다.

간접 통로에서는, 프로그램(10)이 X 컴파일러(16) 및 X 링커(18)에 의해 X 컴퓨터 시스템(12)에 대한 실행 가능 코드(14)로 변환된다. 그 결과 X 실행 가능 코드(14)는 X 컴퓨터 시스템(12)을 동작시킬 수 있다. X-Y 변환기(32)는 참조번호(22A)로 표시된 대응하는 X 실행 가능 코드를 변환한다. 코드 변환은 Y 코드가 축소 명령 세트를 기본으로 하고 있지만 명령 입상 및 상태 분자를 갖는 X 코드 결과를 정확히 얻기 위해 실행하도록 수행된다.

[X 응용 코드의 Y 응용 코드로의 변환]

코드 변환 시스템(40)(제2도)은 제1도의 변환기(32)를 구성하는데 사용된다. 변환 시스템(40)은 프로세서(42), 메모리 시스템(44) 및 X 응용 코드(43)가 변환을 위해 입력되는 각종 입/출력 장치(도시생략)을 갖는 통상의 범용 컴퓨터를 포함한다.

변환 결과는 본 발명에 따라서 Y 코드(45)로 발생되는데 그 Y 코드(45)는 변환중인 코드의 확실한 보증을 하도록 하는 순서 및 구조로 된다. 특히 Y 코드(45)는 그것이 실제로 실행될때 X 명령의 입상 및 상태 분자의 보증된 보전을 용이하게 하도록 하는 구조로 된다. 확실한 CISC 보증의 예로서, VAX? 아키텍춰는 다음의 보증 및 표준을 포함한다.

1. 단일 명령은 완료로 동작하거나 혹은 결코 시작하지 않아야 한다. -그것은 명령을 부분적으로 실행하고, 명령을 일시 정지하고, 다른 명령을 행하고, 결과적으로 중도에 일시 정지 명령을 재개하는 것이 허용되지 않는다.

2. 어떤 메모리 액세스가 페이지 오류 또는 액세스 보호 예외 상황에 직면하여 명령이 완료되지 않게끔 되도록 메모리는 가상 메모리이다.

3. 단일 명령은 다수의 메모리 장소를 기입할 수 있으며, 모든 기입이 발생하거나 아무런 기입도 발생하지 않아야 한다. 아무런 기입도 발생하지 않으면, 명령은 잘못된 기입의 시점부터가 아니라 시작시에 재개될 것이다.

4. 메모리 오페란드는 다수의 기입중 하나를 행한 다음 중단하는 것이 소스 오페란드를 과잉 기입하게 하고 명령의 재개를 불가능하게 하게끔(부분적으로) 오버랩될 수 있다.

5. 단일 명령은 판독-수정-기입 시퀀스를 행할 수 있다.

6. 명령 오페란드는 임의의 바이트 어드레스에서 임의의 바이트로 되는 것이 허용되는 반면, 메모리 하드웨어 구성은 전형적으로 4 또는 8바이트로 구성되는 정수의 정렬된 메모리 워드만을 판독 및 기입할 수 있다. 따라서, 단일 오페란드는 2 또는 그 이상의 메모리 워드의 간격을 가질 수 있으며 단일 오페란드의 액세스가 최초 및 최종 메모리 워드의 여분의 바이트에 대한 엑세스를 포함할 수 있다.

7. 멀티프로세서 시스템에서, 상이한 프로세서에 의한 인접 바이트의 액세스가 항상 독립적인 것으로 되어야 한다. 즉, 어떤 프로세서에 대한 바이트 5의 기입은 다른 프로세서에 대한 바이트 6의 기입을 방해하지 않아야 한다(양자의 기입이 동일 메모리 워드로의 판독-수정-기입 시퀀스를 포함하는 경우에도).

8. 멀티프로세서 시스템에서, 인터록 명령을 거친 액세스가 항상 분자상으로 되어야 한다. 즉, 어떤 프로세서에 대한 바이트 4의 인터록된 판독-수정-기입은 다른 프로세서에 대한 동일 장소로의 인터록된 판독-수정-기입을 결코 방해하지 않아야 한다.

메모리 시스템(44)은 여타 섹션(section) 중에서 통상의 데이타 저장 섹션(46) 및 컴퓨터 오퍼레이팅 시스템이 저장되는 섹션(48)을 구비한다. X-Y 코드 변환에 사용된 기본 요소는 다른 메모리 섹션에 저장된 변환 프로그램이다. 입력 X 코드(43)는 X 코드 리스트(62)로서 저장된다. 또한, Y 명령의 시퀀싱을 제어하기 위해서, Y 명령 하달 기준(52)이 저장되고 X-Y 명령 코드 패턴(54)이 저장되어 명령 연산 특정자(instruction operation specifier) 및 명령 오페란드 특정자(instruction operand specifier)의 변환을 가능케 한다.

변환 프로그램(50)의 일반적 순서도가 제3도에 도시된다. 블록(60)에서, 연속 X 명령은 프로그램 루우프(64)를 통해 처리하기 위해 저장된 X 코드 리스트(62)로부터 순차적으로 입력된다.

루우프(64)에서, 기능 블록(66)은 현재 처리된 X 명령에 대응하는 Y 명령 연산 및 오페란드 특정자를 저장된 X-Y 코드 패턴(54)(제2도)에 따라서 발생한다. 다음, 기능 블록(68)으로 나타낸 바와 같이, 그 결과의 Y 코드는 이어지는 실제 Y 코드 실행시 X 명령 입상의 보전을 용이하게 하는 소정의 기준에 따라 순서를 이룬다.

X-Y 명령 변환의 그래프 표시는 제6도에 도시된다.

모든 X 명령은 일반적으로 입력을 얻고, 입력을 수정하고 그 결과를 일시 기억 장치에 두고 메모리 및 레지스터 기억 장소에 대한 상태 갱신을 제공하는 기본적 작업을 제공한다. X 명령이 "다수의" Y 명령으로 변환될때, Y 명령을 조직하는데 사용되는 순서 기준(52)은 현재 변환된 X 명령에 대한 Y 코드의 Y 명령들을 다음과 같이 그룹지어 순서를 이루도록 하는 것이다.

1. Y 코드의 명령들의 제1그룹 G1은 입력을 얻어 그 입력을 일시 기억 장치에 두는 것이다.

2. Y 코드의 명령들의 제2그룹 G2는 입력을 연산하고 수정된 결과치를 발생하여 그 결과를 일시 기억장치에 저장하는 것이다.

3. Y코드의 명령들의 제3그룹 G3는 X 상태(메모리 또는 레지스터)를 갱신하고 가능한 예외 상황으로 되게 하는 것(차후 정의됨)이다.

4. Y 코드의 명령들의 제4그룹 G4(최종 그룹)을 X 상태(메모리 또는 레지스터)를 갱신하고 가능한 예외 상황이 없도록 하는 것이다.

제4도에 참조 번호(94)로 표시한 X 메모리 상태 및 참조 번호(97)로 표시한 X 레지스터 상태는 각각 X 코드로 정의된 기억 장소로될 Y 머신의 메모리 및 레지스터 구조를 참조한다. X 메모리 및 레지스터 상태는 또한 X 아키텍춰로 비춰질 수 있는 메모리 상태 및 레지스터 상태일 수도 있다.

상술한 방식으로 변환된 코드 명령을 조직하는 장점에 대한 또 다른 정보는 특히 간단한 한번 기입 Y 명령의 경우에 적용되는 것처럼 관련 특허 출원 제1870-0410호에 기재된다.

상태 분자는 X 명령의 모든 상태 액세스가 개입없이 발생하게 되거나 혹은 아무런 액세스도 발생하지 않게 되는 것을 필수적으로 필요로 한다. 이 조건은 X 명령 입상을 제공하는데 필요하다. 후술된 특수한 경우에, X 상태 분자수는 본 발명의 동작을 통해 달성되므로 X 명령 입상이 얻어질 수 있게 된다.

다시 제3도를 참조하면, 일단 기능 블록(68)이 기재된 바와 같이 Y 명령 코드의 순서를 정하면, 테스트 블록(70)은 현재의 Y 명령이 그것이 유출되어졌던 X 명령에 대한 경계 또는 "X 입상" 표지인지의 여부를 판별한다. 연속으로 처리된 Y 명령들에 대한 테스트로부터 비롯된 Yes 및 No 비트는 기능 블록(72)에 의해 X 경계 명령 비트맵에 기록된다.

다음, 복수의 변환 처리 브랜치중 어떤 것이 뒤따르게 될 것인지 결정하도록 일련의 테스트가 행하여진다. 각 브랜치는 메모리 분자의 보전을 위해 특수 변환 처리를 요하는 특수한 경우로서 분류되는 소정의 케이스를 갖는 일반적으로 미리 정해진 변환 케이스(즉 X 명령의 종류)에 대응한다. 이 실시예에서는, 세 브랜치(65, 67, 69)중 어떤 것이 현재 실행중인 X 명령의 문자 구조에 따라 수반될 수 있다.

일반적으로, 브랜치(65, 67, 69)는 각 X명령을 처리하여 일반적으로 그 명령의 확실한 보증을 보전하고 특히 상태 분자를 보전하는 변환 코드를 생성한다.

변환 목적상, 단일 정렬된 장워드(4바이트) 또는 쿼드워드(8바이트)의 단일 RISC 저장 명령은 모든 바이트가 동시에 수정된다는 면에서 RISC 머신에 대해 분자적인 것이 되며, 어떠한 다른 바이트도 그 저장에 의해 영향받지 않는다고 가정하였다는 사실에 유의하여야 한다. 또한 모든 상태가 메모리 또는 레지스터에 유지되고 메모리 액세스가 가상 메모리 예외 상황을 생성할 수도 있으며 간단한 레지스터 이동이 결코 예외 상황을 생성하지 않는다고 가정하였다는 사실에 유의하여야 한다. 아울러, RISC 명령 시퀀스가 임의의 RISC 명령에서 외부 사건에 의해 인터럽트될 수 있다고 가정하였다.

제3도에서, 변환 브랜치(65)는 테스트 블록(120)이 현재의 X 명령이 간단한 한번 기입 명령임을 나타내는 경우 진행된다. 이 경우에, 관련 특허 출원 1870-0410에 기재된 바와 같은 처리가 실행된다. 구체적으로, 블록(122)으로 나타낸 바와 같이 특수한 변환 작업이 요구되지 않으며 블록(74)은 변환될 X 명령이 더 있는지의 여부를 판별한다. 만일 그렇다면, 프로그램 루우프(64)의 실행이

반복된다.

다시 변환 순서도를 참조하여, 만일 현재의 명령이 간단한 한번 기입 명령이 아닌 경우, 테스트 블록(124)에서 X 명령이 다수의 소정의 특수한 한번 기입 케이스중 하나인지의 여부를 판별하게 된다. 본 실시예에서는, 두개의 소정의 특수한 한번 기입 케이스, 즉 부분적 기입 명령 및 인터록 갱신 명령이 존재한다.

만일 특수한 한번 기입 케이스가 적용되면, 기능 블록(126)은 상태 분자 시퀀스를 변환 코드로 삽입함으로써 상태 분자를 보전하도록 변환 브랜치(67)에서 동작한다. 상태 분자 시퀀스는 1) 판독-수정-기입 시퀀스동안 인터럽트가 발생치 않는 경우 부분적 기입 또는 인터록 갱신의 완료, 혹은 2) 판독-수정-기입 시퀀스동안 인터럽트가 발생치 않는 경우 재시도를 위해 부분적 기입 또는 인터록 갱신의 일시 정지를(실행시간에서) 보장한다.

블록(126)에서 변환된 명령 코드에 삽입된 상태 분자 시퀀스는 로드록/저장-조건부 시퀀스라 불리는 것이 바람직하다. 이 시퀀스를 구현하기 위한 적당한 하드웨어 메카니즘은 Digital Equipment사의 관련 특허 출원 PD90-0259호에 기재된다.

제8도에 참조 부호(126A)로 표시된 로드록/저장-조건부 시퀀스에 의해 수행된 실행 시간 논리 동작을 예시한 일반화된 순서도가 도시된다. 따라서, 일단 시퀀스(126A)가 호출되면, 기능 블록(128)은 판독-수정-기입(RMW) 동작이 수행되어져야 할 메모리 워드를 적재한다.

수정 작업은 블록(130, 132, 134)에 의해 수행된다. 정해진 메모리 장소에 1바이트를 가산하는 예시적 경우에, 블록(130)은 메모리 워드의 바이트 정렬을 제공하도록 시프트 동작을 수행한다. 다음 블록(132)은 그 바이트가 0으로 마스크 한다. 마지막으로, 블록(134)은 마스크된 바이트를 메모리 워드내의 재위치에 둔다.

실행 시간에서 RMW 시퀀스의 실행 동안 다른 프로세서가 동일 메모리 장소에 기입하면, RMW 시퀀스는 두개의 독립적인 메모리 기입간의 방해를 막지 못하게 된다. 블록(136)은 RMW 시퀀스의 판독 및 수정 부분동안 동일 메모리 장소로의 또 다른 메모리 기입이 발생하였는지의 여부를 검출하도록 조건부 저장을 실행한다. 만일 다른 기입이 발생치 않으면, 다른 기입과 경합하지 않기 때문에 수정된 메모리 워드가 상태 및 메모리 분자가 유지된채 기입되게끔 조건부 저장이 실시된다. 테스트 블록(138)은 시퀀스를 종료한다. 이어서 실행시간에서 현재의 Y 명령의 추가 처리는 이하에서 보다 자세히 설명될 명령 입상 제어 프로그래밍에 따라서 제어된다.

만일 블록(136)상에서, 또 다른 프로세서가 RMW 시퀀스의 판독 및 메모리 부분 동안 수정 중인 워드의 메모리 장소로의 기입을 행하였다면, 조건부 기억이 취소되어 메모리 분자가 보전되고 RMW 시퀀스가 나중에 다시 시도된다.

다시 제3도의 변환 과정을 참조하면, 일단 시퀀스(126A)가 현재의 Y 명령으로 두어지면, 블록(24)은 X 명령이 더 존재하는지의 여부에 대하여 테스트한다. 만일 더 존재하는 경우라면, 브랜치(67)는 프로그램 사이클을 반복하도록 명령 입력 블록(60)으로 복귀할 수 있다.

브랜치(67)에 의한 단일 기입 CISC 대 RISC 변환을 처리하는 경우를 요약하면, CISC 명령은 하나의 부분적으로(1 또는 2바이트) 정렬된 상태로의 기입을 가지고 번역 코드가 멀티프로세서 시스템에서 실행될 것이며, 인터록 액세스가 요구된다. 이 경우에 상태 분자가 보전되어야 한다면 독립적인 바이트 액세스 및 판독-수정-기입 시퀀스가 적절히 처리되어야 한다.

브랜치(67)의 단일 기입의 경우에, 그룹 1 및 2명령이 갱신될 오페란드를 포함하는 단일 장워드/쿼드워드의 로드록을 포함하여 메모리 및/또는 레지스터 갱신을 제외한 CISC 명령의 모든 작업을 수행하게끔 변환은 제약받는다. 그룹 3명령은 동일 장워드/쿼드워드에 대한 조건부 저장 명령을 포함하고, 그룹 4명령은 간단한 레지스터 이동에 따른 시퀀스의 시작으로의 브랜치-온-실패 명령(branch-on-fail instruction)을 포함한다.

명령 그룹이 완료되기 전에 인터럽트된 변화된 시퀀스는 재개될때 조건부 저장 명령을 실패하므로, 시퀀스의 시작으로 다시 브랜치하게 된다. 아울러, 다른 프로세서가 로드록 명령 후이지만 조건부 저장 명령 전에 특정 장워드/쿼드워드로 기입하는 경우, 조건부 저장은 실패로 되므로 시퀀스의 시작으로 다시 브랜치하게 된다.

명령 그룹 3이 완료된후 그리고 명령 그룹 4가 완료되기 전에 인터럽트된 변환 시퀀스는 관련 특허 출원 1870-0410호에 기재된 바와 같은 간단한 레지스터 이동을 통해 해석하는 메카니즘에 대해 명령 그룹 4를 완료하게 된다. 각 변환된 시퀀스가 시작에서 종료까지 중간에 다른 변환된 시퀀스와 다른 장워드/쿼드워드의 기입없이 실행하거나, 그 시퀀스의 실행이 그룹 3의 완료전에 일시 정지되고 차후에 시작부터 재시도된다는 것이 순효과이다.

다시 제3도를 참조하면, 테스트 블록(124)이 변환중인 X 명령이 특수한 한번 기입 명령을 갖지 않음을 알게된 경우, 변환 브랜치(69)로 들어가게 되고 블록(128)은 변환중인 X 명령이 다른 특수 케이스 즉 다중기입 X 명령이라는 사실을 등록한다. 이어서 기능 블록(130)은 차후보다 상세히 설명될 다중 기입 명령에 대한 상태 분자를 제공하도록 PAL_CALL 루우틴(132)(제7도)을 변환된 명령 코드에 삽입한다.

PAL_CALL 루우틴(132)은 최고 오퍼레이팅 시스템 우선순위를 갖고서 호출되는 루우틴을 실행하기 위한 메카니즘을 제공하도록 전형적으로 다수의 컴퓨터 아키텍춰에 이용될 수 있고 컴퓨터 시스템(20)에 내장되는 특권형 아키텍춰 라이브러리(Privileged Architecture Library)로부터 실행 시간에서의 실행을 위해 호출된다. 일반적으로, PAL_CALL 루우틴(132)은 루우틴의 호출전에 비동기 사건 인터럽트가 발생하지 않고 또 현재의 Y 코드 명령 입상의 남아있는 시퀀스에서 가능한 예외 상황이 검출되지 않는 경우 상태 분자를 갖고서 모든 상태 기입을 실행한다. 그렇지 않으면, PAL_CALL 루우틴(132)은 차후의 재시도를 위해 메모리 분자가 보전된 채로 그것의 실행전에 실패로 된다.

PAL_CALL 루우틴(132)은 관련 특허 출원 PD86-0114에 기재된 하드웨어 구조에 의해 실현되는 것이 바람직하다.

이하 다중 기입의 경우에 Y 코드 명령에 대한 비동기 사건의 관계를 나타내는 선도인 제6도를 참조하기로 한다. 이 특수한 경우에, 화살표(75)는 제1기입이 처리된 후(PAL_CALL 루우틴(132)에 의해) 모든 다중 기입이 PAL_CALL 루우틴(132)에 의해 실행되기 전에, 발생한 비동기 사건을 가리킨다. 일반적으로, 이 경우에 상태 분자는 PAL_CALL 루우틴(132)과 나머지 명령 입상이 실행될 때까지 인터럽트의 실행을 일시 정지함으로써 보전될 수 있다.

구체적으로, 제7도에 도시된 PAL_CALL 루우틴(132)에 대한 순서도를 참조하면, PAL_CALL 루우틴(132)이 변환된 코드를 위한 실행 시간에서 실행중인 Y 코드 입상에 의해 호출된 후, 기능 블록(152)으로 들어가게 된다. 다음 블록(154)는 인터럽트가 RS 사이의 시퀀스동안 그리고 PAL_CALL 루우틴(132)에 대한 호출전에 발생하였는지의 여부를 테스트한다. 만일 발생하였다면, 블록(156)은 상태 분자를 보전하기 위해서 명령 시퀀스에 대한 실패 메세지를 복귀시키고, 현재 Y 코드 입상 실행시에 재시도를 위해 출구 블록(157)을 통해 호출 절차로 복귀하게 된다.

PAL_CALL 루우틴에서 앞서는 그 임계 시간 동안 인터럽트가 발생치 않으면, 테스트 블록(158)은 남아있는 모든 상태 액세스가 예외 상황없이 완료될 수 있는지의 여부를 판별한다. 만일 그렇지 않다면, 블록(156)은 다시 상태 분자를 보전하도록 행해진 명령 시퀀스에 대한 실패 메세지를 복귀시키며, 루우틴 Y 코드 시퀀스의 재시도를 가능케하도록 블록(157)을 통해 나가게 된다.

남아있는 상태 액세스가 예외 상황없이 완료될 수 있다면, PAL_CALL 루우틴(132)의 실행이 개시되고, 기능 블록(159)은 부분적 기입이 특정되는 경우 제1기입의 로드록 및 수정을 행한다. 이어서 블록(160)은 부분적 기입을 위한 조건부 저장 또는 완전 기입을 위한 저장을 수행한다.

테스트 블록(162)은 부분적 기입의 경우에 조건부 저장이 실패했는지의 여부를 판별한다. 만일 그렇다면, 루우틴은 도시한 바와 같이 부분적 기입의 재시도를 위해 블록(159)로 복귀하는 것이 바람직하다. 원한다면, 실패 메세지가 이 시점에서 대신 복귀될 수 있다.

만일 완전한 기입이 기억되거나 일단 부분적 기입이 성공적으로 저장되었다면, 다중 기입 시퀀스중 제1기입이 발생하고 Y 코드 시퀀스의 모든 기입이 기능 블록(164)로 나타낸 것처럼 상태분자를 보전하도록 완료되어야 한다.

다음, Y 코드 시퀀스의 제2기입이 예비 기입 블록(166)에 의해 처리되고 저장 또는 조건부 저장블록(168)이 실시되며, 테스트 블록(170)은 조건부 저장 실패에 대하여 실행된다. 이 동작은 블록(158, 160, 162)에 의한 제1기입의 처리에 대하여 기재한 방식과 유사하다. 동일 서브 세트의 처리(즉, 블록(158, 160, 162)에 포함된 것)는 각 선행 기입이 부분적 또는 완전 기입에 의해 성공적으로 완료됨에 따라 단일 기능 블록(172)에 의해 표시된 바와 같이 다중 기입 시퀀스의 각 후속 기입을 위해 수행된다. 모든 기입이 저장되었을때, 블록(174)은 PAL_CALL 루우틴(132)의 성공적인 완료를 나타내며 출구가 블록(157)을 통해 발생한다. 이어서 현재의 Y 명령 입상이 상태 분자를 보전한채 완료될 수 있다.

제3도의 변환 브랜치(65), (67)의 경우처럼, 브랜치(69)는 Y 명령이 추가로 변환될 필요가 있는지의 여부를 최종적으로 검사하고, 만일 그렇다면 블록(60)은 변환을 위해 다음 X 명령으로 들어가게 된다.

브랜치(69)에 의해 CISC 대 RISC 변환을 처리하는 경우의 개요로서, CISC 명령은 하나 이상의 상태 기입(다수의 행선 또는 단일 비정렬 행선 때문)을 갖는다. 인터록 상태 액세스는 이 경우에 속하지 않지만, 독립적인 바이트 액세스가 적절히 취급되어야 하고 상태 분자가 보존되어야 한다면 특정 기입의 모든 것이 수행되거나 아무것도 수행되지 않아야 한다.

명령 그룹 1 및 2가 판독 및 세트 명령으로 시작하게끔 변환이 제약받는다. 이 그룹들을 뒤따르는 것은 갱신될 각 장워드/쿼드워드의 가능한 적재를 포함하여 메모리 및/또는 레지스터 상태 갱신을 제외한 CISC 명령의 모든 작업을 행하는 명령이다. 명령 그룹(3)은 모든 저장을 특정하는 PAL_CALL 루우틴(132)을 포함하며, 명령 그룹 4는 간단한 레지스터 이동이 수반되는 시퀀스의 시작에 대한 브랜치 실패(branch-on-fail)를 포함한다.

명령 그룹 2가 완료되기 전에 인터럽트된 변환된 시퀀스는 판독 및 세트에 의해 세트된 RISC 상태 비트를 클리어하여 PAL_CALL 루우틴(132)으로 하여금 실패 메세지로 복귀되게 하므로, 시퀀스의 시작으로 다시 복귀한다. 관련 출원 PD86-0114호에 기재된 것과 같은 하드웨어 구조를 사용하면, PAL_CALL 루우틴(132)은 인터럽트 불가능한 RISC 코드의 특권 시퀀스로 들어간다.

PAL_CALL 루우틴(132)은 저장을 행하지 않고 개입 인터럽트가 발생하는 경우 즉 판독 및 세트에 의해 세트된 RISC 상태 비트가 클리어되는 경우 실패 메세지를 복귀시킨다. 그렇지 않으면, 어떤 가능한 가상메모리 예외 상황을 검사하는 모든 가능한 저장 장소에 대한 탐색이 이루어진다. 만일 어떤 예외 상황에 직면하면, PAL_CALL 루우틴(132)은 완료되지 않고 예외 상황이 취해지며, 판독 및 세트에 의해 세트된 상태 비트가 클리어된다. 이것은 PAL_CALL 루우틴(132)의 후속 재실행이 실패 메세지로 복귀되게 하므로, 시퀀스의 시작으로 다시 브랜치된다. 그렇지 않으면 PAL_CALL 루우틴(132)가 모든 지시된 저장을 수행한다. 그렇게 하는 동안, 공유 메모리내의 페이지 테이블이 또 다른 프로세서에 의해 동시에 갱신중이라고 하더라도 이전의 탐색시 사용된 가상 메모리 정보를 사용한다. 따라서, 어떠한 가상 메모리 예외 상황도 저장에 의해 발생되지 않는다.

각 부분적 메모리 워드 상태에 대하여, 로드록/수정/조건부 저장 시퀀스가 특권 코드에 의해 사용된다. 이전의 저장이 행하여지지 않고 또 그러한 제1조건부 저장이 실패로 되면(다른 프로세서가 수정시 동일 메모리 워드로 저장하기 때문), PAL_CALL 루우틴(132)으로부터 “실패”로 복귀하거나 바로 로드록/수정/조건부 저장 시퀀스를 반복할 수도 있다. 단일 저장이 발생된 후, 후속 로드록/수정 조건부 저장 시퀀스가 그들이 성공할 때까지 특권 코드내에서 반복되어야 한다. 모든 특정 기억의 완

료시, 특권 코드가 성공적으로 복귀하고, 명령 그룹 3이 완료된다.

명령 그룹 3이 완료된 후 명령 그룹 4가 완료되기 전에 인터럽트된 변환된 시퀀스는 관련 출원 1870-410호에 보다 상세히 기재된 바와 같은 간단한 레지스터 이동을 통해 해석되는 메카니즘에 의해 그룹 4를 완료하게 된다. 시퀀스가 시작부터 끝까지 중간에 다른 변환된 시퀀스 없이 그리고 본 메모리 워드의 기입을 방해하지 않고 실행하거나, 혹은 명령 그룹 3의 완료전에 일시 정지되고 이어서 시작부터 재시도되는 것이 바로 순효과이다.

모든 X 명령이 브랜치(65, 67, 69)를 통해 변환되었을때, 프로그램 루우프(64)의 순환적 실행이 종료되고 축적된 Y 코드가 기능 블록(76)으로 나타낸 것처럼 출력을 위해 이용될 수 있게 된다.

[X의 상태 분자 및 명령 입상이 보증된 결과적인 Y 코드의 실행]

제4도에 도시한 바와 같이, Y 컴퓨터 시스템(20)(제1도)에 대응하는 Y 프로세서(80)가 X 상태 분자 및 명령 입상이 보전되도록 보증된 결과 Y 코드를 실행하기 위해 제공된다. 통상의 데이타 입/출력 장치(82) 및 간격 클럭(84)가 Y 프로세서(80)에 결합되고, 때때로 이 장치들은 Y 코드 실행으로부터 프로세서 동작의 일시적 전환을 요구하는 비동기 사건을 구성하는 인터럽트를 발생한다. 본 발명에 의해 제공된 보호 보증없이, 이들 또는 다른 인터럽트에 의해 생긴 프로세서 전환은 전술한 바와 같이 특수한 종류의 명령의 실행시 X 상태 분자 및 입상의 파괴를 야기할 수 있다.

제4도에 도시한 바와 같이, 블록(86)은 입력 장치로부터 Y 프로세서(80)에 결합된 메모리 시스템(90)의 섹션(88)에 발생된 Y 코드의 입력을 나타내고, 블록(87)은 Y 코드의 실행 결과로서 출력 장치에 발생된 데이타 출력이다. 메모리 시스템(90)은 또한 조건부 데이타 섹션(92), 통상의 오퍼레이팅 시스템 섹션(94) 및 이전의 X 메모리 상태(95)를 포함한다. Y 프로세서(80)는 이전의 X 레지스터 상태(97)를 포함한다.

메모리 시스템(90)내의 명령 입상 제어 프로그램(IGC)은 명령 입상을 위한 Y 코드의 실행을 감시하도록 하는 구조로 된다. Y 코드의 실행시 IGC 프로그램의 동작은 제5a도 및 제5b도에 도시된 순서도에 보다 상세히 나타내진다.

일반적으로, Y 코드 명령 입상으로 변환된 단일 기입 및 다중 기입 X 명령은 Y 코드의 실행시 IGC 프로그램(96)에 의해 제어된다. 간단한 단일 기입 명령의 경우, 모든 상태 분자 및 명령 입상 제어는 본 명세서와 이전의 관련 출원 1870-0410호에 기재된 바와 같이 향하게 된다. 특수한 단일 기입 명령 및 다중 기입 명령의 경우, 이들 명령을 위한 변환된 코드가 제3도 및 제6도와 관련하여 여기에 기재한 바와 같이 그렇지 않으면 IGC 프로그램(96)을 통한 처리에 따라서 실행되는 동안 상태 분자가 제어된다.

구체적으로, IGC 프로그램(96)(제5a도)은 비동기 사건의 발생으로 (98)로 나타낸 바와 같이 시작한다. 일반적으로, 변환된 X 코드에 비춰질 수 있는 X 상태 변화를 잠정적으로 발생할 수 있는 인터럽트로 인해 Y 명령 스트림의 전환으로 정의된다. 다시 다중 기입을 포함하는 Y 명령의 X 입상에 대한 Y 명령의 X 입상에 대한 비동기 사건의 관계를 나타내는 제6도를 참조하기로 한다.

제5a도의 순서도를 계속 참조하면, 기능 블록(100)에 의한 비동기 사건의 처리시 일시적 홀드가 이루어지고, 기능 블록(102)에 의해 비동기 사건이 기록되는 시점에 Y 명령(PC-AE로 지정됨)의 메모리 어드레스가 처리된다.

다음, 명령 경계 비트맵이 블록(104)에 의해 Y 명령 PC-AE가 X 명령의 경계인지 판별되도록 검사된다. 만일 그렇다면, 테스트 블록(106)은 IGC 프로그램(96)을 통로(107)를 거쳐 블록(108)으로 향하게 하고 그 블록(108)은 X 코드 명령 입상을 파괴시키지 않고 비동기 사건의 처리를 위한 Y 코드의 인터럽트를 가능케 한다.

만일 Y 명령 PC-AE가 X 명령 경계가 아니라면, 기능 블록(100)은 Y 명령 카운터 PC를 X 명령 경계인 다음 백업 또는 순방향 Y 명령과 정렬시킨다. 이어서 프로그램 통로(111)는 역시 X 코드 명령 입상을 파괴시키지 않고 전술한 바와 같은 비동기 사건 처리를 위해 블록(108)으로 IGC 프로그램이 뒤따르게 된다. 이 경우에, 비동기 사건은 단지 하나 또는 그 이상이지만 모두는 아닌 Y 명령이 X 명령 입상내에서 실행되었을 시점에 발생하였고, X 명령 입상의 보전은 프로그램 블록(110)의 동작을 통해 달성된다.

구체적으로, 제5b도의 블록(110)에 대하여 도시한 것처럼, Y 명령의 순방향 스캔은 X 명령 경계에 있는 다음 Y 명령을 찾아내도록 프로그램 루우프(113)내의 기능 블록(112)에 의해 행하여진다. 테스트 블록(114)은 나머지 Y 명령이 실행전에 인터럽트를 처리하는 것이 대응하는 X 코드가 동일 비동기 사건의 도입으로 실행되었다면 생성되었을 결과와 다른 Y 코드 실행 결과를 생성할 수 있을 것인지의 여부를 판별하도록 각각의 순방향으로 스캔된 Y 명령을 검사한다.

각각의 순방향 Y 명령 테스트를 행함에 있어, 테스트 블록(114)은 비동기 사건이 처리될 수 있게 되고 이어서 Y 코드 시퀀스의 실행이 개시될 수 있다면 예외 상황 조건이 Y 명령의 의도된 실행에 의해 생성될 수 있을 것인지 판별하는 것이 바람직하다. 일반적으로 명령은 그것이 완료될 수 있다면 예외 상황을 갖는다. 다음은 순방향 Y 명령으로 식별된다면 X 경계인 다음 백업 Y 명령으로의 Y 코드 중지를 발생하는 일반적 유형의 예외 상황이다.

1) 액세스 제어 위반 또는 페이지 오류와 같은 메모리 관리 예외 상황.

2) 부동점 오우버플로우 또는 0에 의한 제산 오류와 같은 연산 예외 상황.

3) 부적법한 연산 코드 또는 정지점 연산 코드와 같은 명령 예외 상황.

본 발명의 양호한 실시예에 있어서, 변환중인 코드에 대한 적용 가능한 예외 상황의 리스트는 실행시 IGC 프로그램(96)에 액세스될 수 있는 기억 장치에 넣어진다. 따라서 예외 상황 판별은 저장된

예외 상황 리스트에 대하여 각각의 순방향으로 스캔된 Y 명령을 참조함으로써 행하여진다.

연속적인 Y 명령은 순방향 스캔에서 테스트되고, 모든 스캔된 Y 명령이 예외 상황을 보이지 않으면, 나머지 Y 명령은 비동기 사건 처리가 전술한 바와 같이 X 명령 입상을 파괴하지 않고 블록(108)(제5a도)에 의해 가능하게 된다.

한편, 만일 순방향으로 스캔된 Y 명령이 블록(114)에 의한 테스트하에서 예외 상황을 보인다면, 기능 블록(118)은 Y 프로그램 카운터를 X 명령 경계인 다음 백업 Y 명령으로 즉시 백업하며, 다시 블록(108)(제5a도)에 의해 비동기 사건 처리가 X 명령 입상을 파괴하지 않고 가능하게 된다. 이와 같은 방법으로, X 명령 입상의 파괴 가능성을 피할 수 있게 된다.

본 발명은 "하나 대 다수"의 응용 코드 변환을 달성하기 위한 효과적인 메카니즘을 제공한다. 발생된 코드는 실행 결과 뿐만 아니라 상태 문자 및 명령이 원래의 코드에 일치한다. 분자 및 입상은 단일의 한번 기입 명령 뿐만 아니라 다중 기입 및 판독-수정-기입의 단일 기입 타입을 포함하는 특수한 명령에 대하여 보증된다. 따라서, 원래의 CISC 또는 유사 코드에 대한 투자가 RISC에 대한 응용 코드 변환 또는 비교적 축소된 명령 세트를 갖는 다른 개선된 가격/성능의 컴퓨터 시스템을 사용함으로서 가격/성능의 잇점이 동시에 달성되면서 동시에 절약될 수 있다.

본 발명의 기술 사상 및 범위를 벗어나지 않고 본 기술 분야의 전문가에 의해 본 발명의 변환된 프로그램 코드에 대한 명령 상태 분자를 보전하기 위한 개선된 시스템 및 방법에 대한 다양한 변형 및 수정이 가능하다. 따라서, 본 발명은 부속 청구범위 또는 그와 등가의 범위내에 속하는 정도의 어떠한 변형 및 수정을 포함하는 것이다.

(57) 청구의 범위

청구항 1

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제1코드의 명령 분자 상태를 보전하면서 제2프로그램 코드를 실행하기 위한 방법에 있어서, 제1명령 코드를 대응하는 제2코드 명령으로 변환하도록 제1컴퓨터를 동작시키는 단계와 ; 각 제1코드 명령을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하는 단계와 ; 소정의 단일 기입 명령에 대한 상기 제2명령 그룹에 제1서브 시퀀스가 어떤 인터럽트 동작 없이 또 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 제1메모리 장소로의 경합하는 기입이 개입되지 않고 실행되어야 한다는 필요 조건에 따라서 제1메모리 장소로의 단일 기입을 처리하기 위해 제1서브 시퀀스를 갖는 제1특수 기입 명령을 포함시키는 단계와 ; 다중 기입 명령에 대한 상기 제2그룹에 어떤 인터럽트 동작없이 그리고 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 어떤 경합하는 기입 없이 모두 실행되어야 하는 다중 기입을 처리하기 위해 제2서브 시퀀스를 포함하도록 제2특수 기입 명령을 포함시키는 단계와 ; 제2프로그램 코드를 실행하도록 상기 제2컴퓨터 시스템을 동작시키는 단계와 ; 상기 제2코드의 실행시 각각의 비동기 사건의 발생을 판별하는 단계와 ; 상기 제2코드의 실행시 상기 다른 프로세서가 상기 메모리 상태에 결합되면 상기 다른 프로세서에 의한 상기 제1메모리 장소로의 각각의 경합하는 기입의 발생을 판별하는 단계와 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위해 중지하는 단계와 ; 경합 기입이 상기 제1서브 시퀀스의 실행 완료전에 상기 다른 프로세서에 의해 행해진 경우 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 제1특수 명령 시퀀스를 성공적인 실행이 완료될 때까지 중지 및 재시도하는 단계와 ; 비동기 사건 인터럽트가 상기 제1서브 시퀀스의 의도된 실행 동안 발생하는 경우 재시도를 위해 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지하는 단계와 ; A) 만일 상기 제2서브 시퀀스가 입상 명령 시퀀스에 포함되고 비동기 사건 인터럽트가 상기 제2명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료하는 단계를 포함하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 2

제1항에 있어서, 상기 제1서브 시퀀스가 판독-수정-기입 서브 시퀀스인 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 3

제2항에 있어서, 상기 제1서브 시퀀스가 부분적 기입 명령 또는 인터록 갱신 명령을 실현하는데 사용되며, 입력 데이타를 판독 및 수정하고 그 결과 데이타를 조건부로 저장하여 그것의 실행시 경합 기입이 발생한 경우 상기 제1서브 시퀀스의 실행을 실패하거나 그것의 실행시 경합 기입이 발생하지 않은 경우 상기 제1서브 시퀀스를 완료하는 로드-록 조건부 저장 시퀀스를 포함하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 4

제1항에 있어서, 상기 제2서브 시퀀스는 단일 특권 아키텍춰 라이브러리(PAL) 호출 루우틴을 포함하

는데 그 PAL 호출 루우틴은 그것이 일단 개시되면 상기 제2서브 시퀀스와 그 내에 포함된 모든 기입을 실행하며, 인터럽트가 현재의 명령 시퀀스의 실행중에 앞서 발생되지 않고 또 모든 나머지 상태 액세스가 예외 상황없이 완료될 수 있다면 상기 PAL 호출 루우틴의 개시가 가능하게 되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 5

제4항에 있어서, 상기 PAL 호출 루우틴은 제1서브 루우틴을 갖는데 제1서브 루우틴은 실행될 제1기입을 테스트하여, 그것이 부분적 기입인지의 여부를 판별하고, 부분적 기입이라면 상기 제1기입을 실행하도록 로드-록 조건부 저장을 수행하고, 조건부 저장이 상기 제1서브 루우틴의 의도된 실행시 상기 다른 프로세서에 의한 경합하는 상태 기입의 결과로서 실패로 되면 그것이 완료될 때까지 로드-록 조건부 저장을 선택적으로 재시도하고, 재시도가 선택되지 않은 경우 제1시도 동안 혹은 재시도가 선택되는 경우 후속 재시도 동안 어떠한 경합 상태 기입도 발생하지 않으면 로드-록 조건부 저장을 완료하며, 상기 PAL 호출 루우틴은 상기 제1서브 루우틴의 완료시 완료를 위해 록-인되며, 다음 제2서브 루우틴이 실행될 제2기입을 테스트하여 그것이 부분적 기입인지의 여부를 판별하고, 부분적 기입이라면 상기 제2기입을 실행하도록 로드-록 조건부 저장을 수행하고, 상기 제2서브 루우틴의 의도된 실행시 상기 다른 프로세서에 의한 경합하는 상태 기입의 결과로서 조건부 저장이 실패로 되면 그것이 완료될때까지 로드-록 조건부 저장을 재시도하며, 실행될 각각의 연속적인 기입이 모든 기입이 상기 PAL 호출을 완료하도록 실행될 때까지 상기 제2서브 루우틴과 실질적으로 동일한 서브 루우틴에 의해 처리되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 6

제5항에 있어서, 상기 제1서브 루우틴은 그것이 성공적으로 완료될 때까지 재시도되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 7

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제1코드의 명령 분자 상태를 보전하면서 제2프로그램 코드를 실행하기 위한 방법에 있어서, 제1명령 코드를 대응하는 제2코드 명령으로 제2코드 명령의 관점에서 제1코드를 정의하는 패턴 코드에 따라서 변환하도록 제1컴퓨터를 동작시키는 단계와 ; 각 제1코드 명령을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하는 단계와 ; 경합 기입이 상기 서브 시퀀스의 실행 완료 전에 상기 다른 프로세서에 의해 행해진 경우 상기 서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 특수 명령 시퀀스를 성공적인 실행이 완료될 때까지 중지 및 재시도하는 단계와 ; 비동기 사건 인터럽트가 상기 서브 시퀀스의 의도된 실행 동안 발생하는 경우 재시도를 위해 상기 서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지하는 단계와 ; 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료하는 단계를 포함하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행방법.

청구항 8

제7항에 있어서, 상기 제1서브 시퀀스가 판독-수정-기입 서브 시퀀스인 것을 특징으로 하는 프로그램 코드의 변환 및 실행방법.

청구항 9

제8항에 있어서, 상기 제1서브 시퀀스가 부분적 기입 명령 또는 인터록 갱신 명령을 실현하는데 사용되며, 입력 데이타를 판독 및 수정하고 그 결과 데이타를 조건부로 저장하여 그것의 실행시 경합 기입이 발생한 경우 상기 제1서브 시퀀스의 실행을 실패하거나 그것의 실행시 경합 기입이 발생하지 않은 경우 상기 제1서브 시퀀스를 완료하는 로드-록 조건부 저장 시퀀스를 포함하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 10

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제1코드의 명령 분자 상태를 보전하면서 제2프로그램 코드를 실행하기 위한 방법에 있어서, 제1명령 코드를 대응하는 제2코드 명령으로 제2코드 명령의 관점에서 제1코드를 정의하는 패턴 코드에 따라서 변환하도록 제1컴퓨터를 동작시키는 단계와 ; 각 제1코드 명령을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하는 단계와 ; 어떤 인터럽트 동작없이 모두 실행되어야 하는 다중 기입을 처리하기 위한 서브 시퀀스를 포함하도록 특수 기입 명령을 구성하는 단계와 ; 제2프로그램 코드를 실행하도록 제2아키텍춰에 적합한 상기 제2컴퓨터 시스템을 동작시키는 단계와 ; 상기 제2코드의 실행시 각각의 비동기 사건의 발생을 판별하는 단계와 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위

해 중지하는 단계와 ; A) 만일 상기 제2서브 시퀀스가 입상 명령 시퀀스에 포함되고 비동기 사건 인터럽트가 상기 제2명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료하는 단계를 포함하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 11

제10항에 있어서, 상기 제2서브 시퀀스는 단일 특권 아키텍춰 라이브러리(PAL) 호출 루우틴을 포함하는데 그 PAL 호출 루우틴은 그것이 일단 개시되면 상기 제2서브 시퀀스와 그 내에 포함된 모든 기입을 실행하며, 인터럽트가 현재의 명령 시퀀스의 실행중에 앞서 발생되지 않고 또 모든 나머지 상태 액세스가 예외 상황없이 완료될 수 있다면 상기 PAL 호출 루우틴의 개시가 가능하게 되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 12

제11항에 있어서, 상기 PAL 호출 루우틴은 제1서브 루우틴을 갖는데 그 제1서브 루우틴은 실행될 제1기입을 테스트하여, 그것이 부분적 기입인지의 여부를 판별하고, 부분적 기입이라면 상기 제1기입을 실행하도록 로드-록 조건부 저장을 수행하고, 조건부 저장이 상기 제1서브 루우틴의 의도된 실행시 상기 다른 프로세서에 의한 경합하는 상태 기입의 결과로서 실패로 되면 그것이 완료될 때까지 로드-록 조건부 저장을 선택적으로 재시도하고, 재시도가 선택되지 않은 경우 제1시도 동안 혹은 재시도가 선택되는 경우, 후속 재시도 동안 어떠한 경합 상태 기입도 발생하지 않으면 로드-록 조건부 저장을 완료하며, 상기 PAL 호출 루우틴은 상기 제1서브 루우틴의 완료시 완료를 위해 록-인되며, 다음 제2서브 루우틴이 실행될 제2기입을 테스트하여 그것이 부분적 기입인지의 여부를 판별하고, 부분적 기입이라면 상기 제2기입을 실행하도록 로드-록 조건부 저장을 수행하고, 상기 제2서브 루우틴의 의도된 실행시 상기 다른 프로세서에 의한 경합하는 상태 기입의 결과로서 조건부 저장이 실패로 되면 그것이 완료될 때까지 로드-록 조건부 저장을 재시도하며, 실행될 각각의 연속적인 기입이 모든 기입이 상기 PAL 호출을 완료하도록 실행될 때까지 상기 제2서브 루우틴과 실질적으로 동일한 서브 루우틴에 의해 처리되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 13

제12항에 있어서, 상기 제1서브 루우틴은 그것이 성공적으로 완료될 때까지 재시도 되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 14

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제2코드가 실행될때 제1코드의 명령 분자상태의 보전을 용이하게 하기 위한 방법에 있어서, 제1명령 코드를 대응하는 제2코드 명령으로 제2코드 명령의 관점에서 제1코드 명령을 정의하는 패턴 코드에 따라서 변환하도록 제1컴퓨터를 동작시키는 단계와 ; 각 제1코드 명령을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하는 단계와 ; 단일 기입이 어떤 인터럽트 동작 없이 또 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 제1메모리 장소로의 경합하는 기입이 개입되지 않고 실행되어야 한다는 필요 조건에 따라서 제1메모리 장소로의 단일 기입을 처리하기 위한 제1서브 시퀀스를 포함하도록 제1특수 기입 명령을 구성하는 단계와 ; 어떤 인터럽트 동작없이 모두 실행되어야 하는 다중 기입을 처리하기 위한 제2서브 시퀀스를 포함하도록 제2특수 기입 명령을 구성하는 단계와 ; 상기 제1서브 시퀀스의 실행 완료전에 상기 다른 프로세서에 의해 경합하는 기입이 행해지는 경우 성공적인 실행이 완료될 때까지 재시도를 위해 중지하도록 상기 제1서브 시퀀스를 구성하는 단계와 ; 상기 제1시퀀스의 의도된 실행동안 비동기 사건 인터럽트가 발생한다면 실패로 되게하여, 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 실행의 재시도를 가능케하도록 상기 제1서브 시퀀스를 구성하는 단계와 ; 특권 인터럽트 불가능 실행을 위한 상기 제2서브 시퀀스를 구성하여, 제2코드 실행시 상기 제2서브 시퀀스의 실행 완료 후까지 어떤 비동기 사건 인터럽트의 처리를 지연시키는 단계를 포함하는 것을 특징으로 하는 프로그램 코드의 변환 방법.

청구항 15

제2프로그램 코드가 번역되어질 제1프로그램 코드의 상태 분자를 보전하면서 제1프로그램 코드를 실행하기 위한 방법으로서, 상기 제1프로그램 코드는 제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터 상에서 실행될 수 있고 상기 제2프로그램 코드는 메모리 및 레지스터 상태에 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있으며, 각 제1코드 명령에 대한 제2코드 명령이 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직되며, 제1특수 기입 명령은 상기 단일 기입이 어떤 인터럽트 동작없이 또 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 상기 제1메모리 장소로의 경합하는 기입이 개입되지 않고 실행되어야 한다는 필요 조건에 따라서 제1메모리 장소로의 단일 기입을 처리하기 위해 제1서브 시퀀스를 포함하도록 구성되며, 제2특수 기입 명령은 어떤 인터럽트 동작 없이 모두 실행되어야 하는 다중 기입을 처리하기 위해 제2서브 시퀀스를 포함하도록 구성되는 방법에 있어서, 제2프로그램 코드를 실행하도록 제2아키텍춰에 적합한 상기 제2컴퓨터 시스템을 동작시키는 단계와 ; 상기 제2코드의 실행시 각각의 비동기 사건의

발생을 판별하는 단계와 ; 상기 제2코드의 실행시 상기 다른 프로세서가 상기 메모리 상태에 결합되면 상기 다른 프로세서에 의한 상기 제1메모리 장소로의 각각의 경합하는 기입의 발생을 판별하는 단계와 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위해 중지하는 단계와 ; 경합 기입이 상기 제1서브 시퀀스의 실행 완료 전에 상기 다른 프로세서에 의해 행해진 경우 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 제1특수 명령 시퀀스를 성공적인 실행이 완료될 때까지 재시도를 위해 중지하는 단계와 ; 비동기 사건 인터럽트가 상기 제1서브 시퀀스의 의도된 실행 동안 발생하는 경우 재시도를 위해 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지하는 단계와 ; A) 만일 상기 제2서브 시퀀스가 입상 명령 시퀀스에 포함되고 비동기 사건 인터럽트가 상기 제2명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 상기 특수 명령의 실행후 또는 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료하는 단계를 포함하는 것을 특징으로 하는 프로그램 코드의 실행 방법.

청구항 16

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제1코드의 명령 분자 상태를 보전하게끔 제2프로그램 코드를 실행하기 위한 시스템에 있어서, 제1프로그램 코드를 제2프로그램 코드로 변환하기 위한 제1프로세서 및 상기 제1프로세서에 결합된 제1메모리 시스템을 갖는 제1컴퓨터 시스템과 ; 상기 제1메모리 시스템에 제2프로그램 코드로서 저장하기 위한 상기 패턴 코드에 따라서 제1코드의 각 연속 명령을 제2코드 명령들로 변환하기 위한 수단과 ; 각 제1코드 명령에 대한 제2코드 명령들을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하기 위한 수단과 ; 단일 기입이 어떤 인터럽트 동작 없이 또 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 제1메모리 장소로의 경합하는 기입이 개입되지 않고 실행되어야 한다는 필요조건에 따라서 제1메모리 장소로의 단일 기입을 처리하기 위해 제1서브 시퀀스를 포함하도록 제1특수 기입 명령을 구성하기 위한 수단과 ; 어떤 인터럽트 동작없이 모두 실행되어야 하는 다중 기입을 처리하기 위해 제2서브 시퀀스를 포함하도록 제2특수 기입 명령을 구성하기 위한 수단과 ; 상기 제2아키텍춰를 가지고 상기 제2프로세서와 상기 제2프로세서에 결합된 제2메모리 시스템을 포함하는 메모리 및 레지스터 상태를 가지며, 상기 제1컴퓨터 시스템에 의해 출력으로서 발생된 제2코드를 실행하기 위한 제2컴퓨터 시스템과 ; 상기 제2코드의 실행시 각 비동기 사건의 발생과 상기 다른 프로세서가 상기 메모리 상태에 결합되면 상기 다른 프로세서에 의한 상기 제1메모리 장소로의 각각의 경합하는 기입의 발생을 판별하기 위한 수단과 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예의 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위해 중지시키기 위한 수단과 ; 경합 기입이 상기 제1서브 시퀀스의 실행 완료 전에 상기 다른 프로세서에 의해 행해진 경우 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 제1특수 명령 시퀀스를 성공적인 실행이 완료될 때까지 재시도를 위해 중지시키기 위한 수단과 ; 비동기 사건 인터럽트가 상기 제1서브 시퀀스의 의도된 실행 동안 발생하는 경우 재시도를 위해 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지시키기 위한 수단과 ; A) 만일 상기 제2서브 시퀀스가 입상 명령 시퀀스에 포함되고 비동기 사건 인터럽트가 상기 제2명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료하기 위한 수단을 구비하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 17

제16항에 있어서, 상기 제1서브 시퀀스가 판독-수정-기입 서브 시퀀스인 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 18

제17항에 있어서, 상기 제1서브 시퀀스가 부분적 기입 명령 또는 인터록 갱신 명령을 실행하는데 사용되며, 입력 데이타를 판독 및 수정하고 그 결과 데이타를 조건부로 저장하여 그것의 실행시 경합 기입이 발생한 경우 상기 제1서브 시퀀스의 실행을 실패하거나 그것의 실행시 경합 기입이 발생하지 않은 경우 상기 제1서브 시퀀스를 완료하는 로드-록 조건부 저장 시퀀스를 구비하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 19

제16항에 있어서, 상기 제2서브 시퀀스는 단일 특권 아키텍춰 라이브러리(PAL) 호출 루우틴을 구비하는데 그 PAL 호출 루우틴은 그것이 일단 개시되면 상기 제2서브 시퀀스와 그 내에 포함된 모든 기입을 실행하며, 인터럽트가 현재의 명령 시퀀스의 실행중에 앞서 발생되지 않고 또 모든 나머지 상태 액세스가 예외 상황없이 완료될 수 있다면 상기 PAL 호출 루우틴의 개시가 가능하게 되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 20

제19항에 있어서, 상기 PAL 호출 루우틴은 제1서브 루우틴을 갖는데 그 제1서브 루우틴은 실행될 제1기입을 테스트하여, 그것이 부분적 기입인지의 여부를 판별하고, 부분적 기입이라면 상기 제1기입을 실행하도록 로드-록 조건부 저장을 수행하고, 조건부 저장이 상기 제1서브 루우틴의 의도된 실행시 상기 다른 프로세서에 의한 경합하는 상태 기입의 결과로서 실패로 되면 그것이 완료될 때까지 로드-록 조건부 저장을 선택적으로 재시도하고, 재시도가 선택되지 않은 경우 제1시도 동안 혹은 재시도가 선택되는 경우, 후속 재시도 동안 어떠한 경합 상태 기입도 발생하지 않으면 로드-록 조건부 저장을 완료하며, 상기 PAL 호출 루우틴은 상기 제1서브 루우틴의 완료시 완료를 위해 록-인되며, 다음 제2서브 루우틴이 실행될 제2기입을 테스트하여 그것이 부분적 기입인지의 여부를 판별하고, 부분적 기입이라면 상기 제2기입을 실행하도록 로드-록 조건부 저장을 수행하고, 상기 제2서브 루우틴의 의도된 실행시 상기 다른 프로세서에 의한 경합하는 상태 기입의 결과로서 조건부 저장이 실패로 되면 그것이 완료될 때까지 로드-록 조건부 저장을 재시도하며, 실행될 각각의 연속적인 기입이 모든 기입이 상기 PAL 호출을 완료하도록 실행될 때까지 상기 제2서브 루우틴과 실질적으로 동일한 서브 루우틴에 의해 처리되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 21

제20항에 있어서, 상기 제1서브 루우틴은 그것이 성공적으로 완료될 때까지 재시도되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 22

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제1코드의 명령 분자 상태를 보전하게끔 제2프로그램 코드를 실행하기 위한 시스템에 있어서, 제1프로그램 코드를 제2프로그램 코드로 변환하기 위한 제1프로세서 및 상기 제1프로세서에 결합된 제1메모리 시스템을 갖는 제1컴퓨터 시스템과 ; 상기 제1메모리 시스템에 제2프로그램 코드로서 저장하기 위한 상기 패턴 코드에 따라서 제1코드의 각 연속 명령을 제2코드 명령들로 변환하기 위한 수단과 ; 각 제1코드 명령을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하기 위한 수단과 ; 단일 기입이 어떤 인터럽트 동작 없이 또 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 제1메모리 장소로의 경합하는 기입이 개입되지 않고 실행되어야 한다는 필요 조건에 따라서 제1메모리 장소로의 단일 기입을 처리하기 위해 서브 시퀀스를 포함하도록 특수 기입 명령을 구성하기 위한 수단과 ; 상기 제2아키텍춰를 가지고 제2프로세서와 상기 제2프로세서에 결합된 제2메모리 시스템을 포함하는 메모리 및 레지스터 상태를 가지며, 상기 제1컴퓨터 시스템에 의해 출력으로서 발생된 제2코드를 실행하기 위한 제2컴퓨터 시스템과 ; 상기 제2코드의 실행시 각 비동기 사건의 발생 및 상기 다른 프로세서가 상기 메모리 상태에 결합되면 상기 다른 프로세서에 의한 상기 제1메모리 장소로의 각각의 경합하는 기입의 발생을 판별하기 위한 수단과 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위해 중지시키기 위한 수단과 ; 경합 기입이 상기 서브 시퀀스의 실행 완료 전에 상기 다른 프로세서에 의해 행해진 경우 상기 서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 특수 명령 시퀀스를 성공적인 실행이 완료될 때까지 재시도를 위해 중지시키기 위한 수단과 ; 비동기 사건 인터럽트가 상기 서브 시퀀스의 의도된 실행 동안 발생하는 경우 재시도를 위해 상기 서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지시키기 위한 수단과 ; 상기 제2프로세서 및 상기 제2메모리 시스템을 구비하여 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료하기 위한 수단을 구비하는 것을 특징으로 하는 프로그램 코드의 변화 및 실행 시스템.

청구항 23

제22항에 있어서, 상기 제1서브 시퀀스가 판독-수정-기입 서브 시퀀스인 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 24

제23항에 있어서, 상기 제1서브 시퀀스가 부분적 기입 명령 또는 인터록 갱신 명령을 실행하는데 사용되며, 입력 데이타를 판독 및 수정하고 그 결과 데이타를 조건부로 저장하여 그것의 실행시 경합 기입이 발생한 경우 상기 제1서브 시퀀스의 실행을 실패하거나 그것의 실행시 경합 기입이 발생하지 않은 경우 상기 제1서브 시퀀스를 완료하는 로드-록 조건부 저장 시퀀스를 포함하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 25

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제1코드의 명령 분자 상태를 보전하게끔 제2프로그램 코드를 실행하기 위한 시스템에 있어서, 제1프로그램 코드를 제2프로그램 코드로 변환하기 위한 제1프로세서 및 상기 제1프로세서에 결합된 제1메모리 시스템을 갖는 제1컴퓨터 시스템과 ; 상기 제1메모리 시스템에 제2프로그램 코드로서 저장하기 위한 상기 패턴 코드에 따라서 제1코드의 각 연속 명령을 제2코드 명령들로 변환하기 위한 수단과 ; 각 제1코드 명령에 대한 제2코드

명령들을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하기 위한 수단과 ; 어떤 인터럽트 동작 없이 모두 실행되어야 하는 다중 기입을 처리하기 위한 서브 시퀀스를 포함하도록 특수 기입 명령을 구성하기 위한 수단과 ; 상기 제2아키텍춰를 가지고 제2프로세서와 상기 제2프로세서에 결합된 제2메모리 시스템을 구비하는 메모리 및 레지스터 상태를 가지며, 상기 제1컴퓨터 시스템에 의해 출력으로서 발생된 제2코드를 실행하기 위한 제2컴퓨터 시스템과 ; 상기 제2코드의 실행시 각각의 비동기 사건의 발생을 판별하기 위한 수단과 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위해 중지시키기 위한 수단과 ; 상기 제2프로세서 및 상기 제2메모리 시스템을 구비하여 A) 만일 상기 제2서브 시퀀스가 입상 명령 시퀀스에 포함되고 비동기 사건 인터럽트가 상기 제2명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료시키기 위한 수단을 구비하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 26

제25항에 있어서, 상기 제2서브 시퀀스는 단일 특권 아키텍춰 라이브러리(PAL) 호출 루우틴을 포함하는데 그 PAL 호출 루우틴은 그것이 일단 개시되면 상기 제2서브 시퀀스와 그 내에 포함된 모든 기입을 실행하며, 인터럽트가 현재의 명령 시퀀스의 실행중에 앞서 발생되지 않고 또 모든 나머지 상태 액세스가 예외 상황없이 완료될 수 있다면 상기 PAL 호출 루우틴의 개시가 가능하게 되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 27

제26항에 있어서, 상기 PAL 호출 루우틴은 제1서브 루우틴을 갖는데 그 제1서브 루우틴은 실행될 제1기입을 테스트하여, 그것이 부분적 기입인지의 여부를 판별하고, 부분적 기입이라면 상기 제1기입을 실행하도록 로드-록 조건부 저장을 수행하고, 조건부 저장이 상기 제1서브 루우틴의 의도된 실행시 상기 다른 프로세서에 의한 경합하는 상태 기입의 결과로서 실패로 되면 그것이 완료될 때까지 로드-록 조건부 저장을 선택적으로 재시도하고, 재시도가 선택되지 않은 경우 제1시도 동안 혹은 재시도가 선택되는 경우 후속 재시도 동안 어떠한 경합 상태 기입도 발생하지 않으면 로드-록 조건부 저장을 완료하며, 상기 PAL 호출 루우틴은 상기 제1서브 루우틴의 완료시 완료를 위해 록-인되며, 다음 제2서브 루우틴이 실행될 제2기입을 테스트하여 그것이 부분적 기입인지의 여부를 판별하고, 부분적 기입이라면 상기 제2기입을 실행하도록 로드-록 조건부 저장을 수행하고, 상기 제2서브 루우틴의 의도된 실행시 상기 다른 프로세서에 의한 경합하는 상태 기입의 결과로서 조건부 저장이 실패로 되면 그것이 완료될 때까지 로드-록 조건부 저장을 재시도하며, 실행될 각각의 연속적인 기입이 모든 기입이 상기 PAL 호출을 완료하도록 실행될 때까지 상기 제2서브 루우틴과 실질적으로 동일한 서브 루우틴에 의해 처리되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 28

제27항에 있어서, 상기 제1서브 루우틴은 그것이 성공적으로 완료될 때까지 재시도되는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

청구항 29

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여, 제1코드의 명령 분자 상태의 보전을 용이하게 하기 위한 시스템에 있어서, 제1프로그램 코드를 제2프로그램 코드로 변환하기 위한 제1프로세서 및 상기 제1프로세서에 결합된 제1메모리 시스템을 가지는 제1컴퓨터 시스템과 ; 상기 제1메모리 시스템에 제2프로그램 코드로서 저장하기 위한 상기 패턴 코드에 따라서 제1코드의 각 연속 명령을 제2코드 명령들로 변환하기 위한 수단과 ; 각 제1코드 명령을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하기 위한 수단과 ; 단일 기입이 어떤 인터럽트 동작 없이 또 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 제1메모리 장소로의 경합하는 기입이 개입되지 않고 실행되어야 한다는 필요 조건에 따라서 제1메모리 장소로의 단일 기입을 처리하기 위한 제1서브 시퀀스를 포함하도록 제1특수 기입 명령을 구성하는 수단과 ; 어떤 인터럽트 동작 없이 모두 실행되어야 하는 다중 기입을 처리하기 위한 제2서브 시퀀스를 포함하도록 제2특수 기입 명령을 구성하기 위한 수단과 ; 경합 기입이 상기 제1서브 시퀀스의 실행 완료 전에 상기 다른 프로세서에 의해 행해진 경우 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 제1특수 명령 시퀀스를 성공적인 실행이 완료될 때까지 재시도를 위해 중지시키기 위한 수단과 ; 상기 제2프로세서 및 상기 제2메모리 시스템을 구비하여, 비동기 사건 인터럽트가 상기 제1서브 시퀀스의 의도된 실행동안 발생하는 경우 재시도를 위해 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지시키기 위한 수단과 ; 상기 제2프로세서 및 상기 제2메모리 시스템을 구비하여, A) 만일 상기 제2서브 시퀀스가 입상 명령 시퀀스에 포함되고 비동기 사건 인터럽트가 상기 제2명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료하기 위한 수단을 구비하는 것을 특징으로 하는 프로그램

코드의 변환 시스템.

청구항 30

제2프로그램 코드가 번역되어질 제1프로그램 코드의 상태 분자를 보전하면서 제2프로그램 코드를 실행하기 위한 시스템으로서, 상기 제1프로그램 코드는 제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있고 상기 제2프로그램 코드는 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있으며, 각 제1코드 명령에 대한 제2코드 명령이 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직되며, 제1특수 기입 명령은 상기 단일 기입이 어떤 인터럽트 동작없이 또 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 상기 제1메모리 장소로의 경합하는 기입이 개입되지 않고 실행되어야 한다는 필요 조건에 따라서 제1메모리 장소로의 단일 기입을 처리하기 위해 제1서브 시퀀스를 포함하도록 구성되며, 제2특수 기입 명령은 어떤 인터럽트 동작 없이 모두 실행되어야 하는 다중 기입을 처리하기 위해 제2서브 시퀀스를 포함하도록 구성되는 시스템에 있어서, 제2프로그램 코드를 실행하도록 제2아키텍춰에 적합한 상기 제2컴퓨터 시스템을 동작시키기 위한 수단과 ; 상기 제2코드의 실행시 각각의 비동기 사건의 발생을 판별하기 위한 수단과 ; 상기 제2코드의 실행시 상기 다른 프로세서가 상기 메모리 상태에 결합되면 상기 다른 프로세서에 의한 상기 제1메모리 장소로의 각각의 경합하는 기입의 발생을 판별하기 위한 수단과 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위해 중지시키기 위한 수단과 ; 경합 기입이 상기 제1서브 시퀀스의 실행 완료 전에 상기 다른 프로세서에 의해 행해진 경우 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 제1특수 명령 시퀀스를 성공적인 실행이 완료될 때까지 재시도를 위해 중지시키기 위한 수단과 ; 비동기 사건 인터럽트가 상기 제1서브 시퀀스의 의도된 실행 동안 발생하는 경우 재시도를 위해 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지시키기 위한 수단과 ; A) 만일 상기 제2서브 시퀀스가 입상 명령 시퀀스에 포함되고 비동기 사건 인터럽트가 상기 제2명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 상기 특수 명령의 실행후 또는 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료시키기 위한 수단을 구비하는 것을 특징으로 하는 프로그램 코드의 실행 시스템.

청구항 31

제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제1코드의 명령 상태 분자를 보전하게끔 제2프로그램 코드를 실행하기 위한 방법에 있어서, 제1명령 코드를 대응하는 제2코드 명령으로 제2코드 명령의 관점에서 제1코드 명령을 정의하는 패턴 코드에 따라서 변화하도록 제1컴퓨터를 동작시키는 단계와 ; 각 제1코드 명령에 대한 제2코드 명령들을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하는 단계와 ; 경합하는 사건의 개입없이 전체적으로 처리되어야 하는 서브 시퀀스를 포함하는 적어도 하나의 제1특수 기입 명령을 상기 제2명령 그룹에 포함시키는 단계와 ; 제2프로그램 코드를 실행하도록 상기 제2아키텍춰에 적합한 상기 제2컴퓨터 시스템을 동작시키는 단계와 ; 상기 제2코드의 실행시 상기 제1특수 기입 명령의 메모리 문자와의 경합을 실제로 또는 가상적으로 생성하는 각각의 개입 사건의 발생을 판별하는 단계와 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위해 중지하는 단계와 ; 상기 제1서브 시퀀스가 다수의 단계를 포함하는 단일 기입 명령이고 경합 기입이 상기 제1서브 시퀀스의 실행 완료 전에 상기 다른 프로세서에 의해 행해졌다고 검출되면, 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 제1특수 명령 서브 시퀀스를 성공적인 실행이 완료될 때까지 중지 및 재시도하는 단계와 ; 상기 제1서브 시퀀스가 단일 기입 명령이고 비동기 사건 인터럽트가 상기 제1서브 시퀀스의 의도된 실행시 가능한 경합으로서 추정되면, 재시도를 위해 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지하는 단계와 ; A) 상기 제1서브 시퀀스가 다중 기입 명령이고 입상 명령 시퀀스에 포함되며 또 비동기 사건 인터럽트가 상기 제1명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료하는 단계를 포함하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 32

제31항에 있어서, 상기 제1특수 기입 명령은 상기 제1서브 시퀀스가 어떤 인터럽트 없이 그리고 상기 메모리 상태에 결합될 수 있는 어떤 다른 프로세서에 의한 제1메모리 장소로의 경합하는 기입의 개입없이 실행되어야 하는 필요 조건에 따라서 상기 제1메모리 장소로의 제1기입을 처리하기 위한 제1서브 시퀀스를 포함하는 제1특수 기입 명령인 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 33

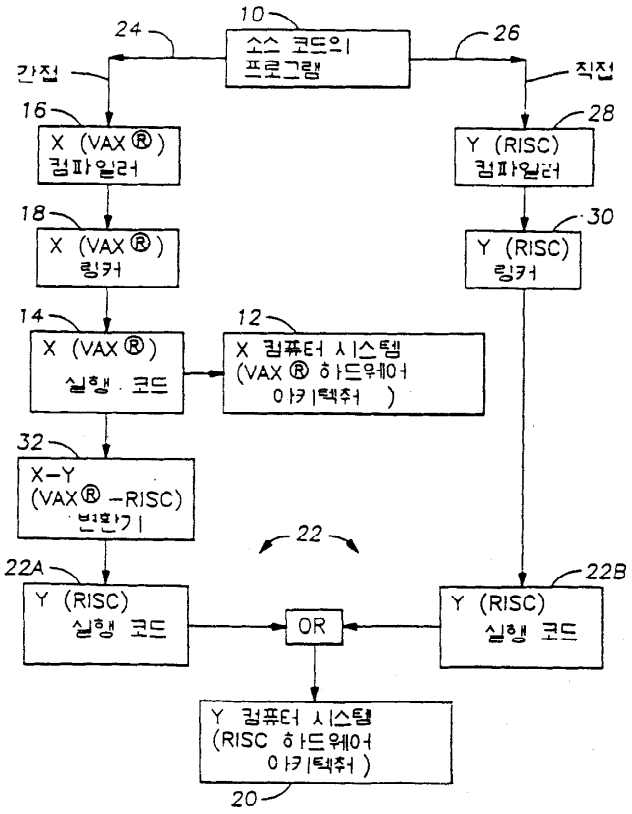
제31항에 있어서, 상기 제1특수 기입 명령은 어떤 인터럽트 없이 모두 실행되어야 하는 다중 기입을 처리하기 위한 제2서브 시퀀스를 포함하는 제2특수 기입 명령인 것을 특징으로 하는 프로그램 코드의 변환 및 실행 방법.

청구항 34

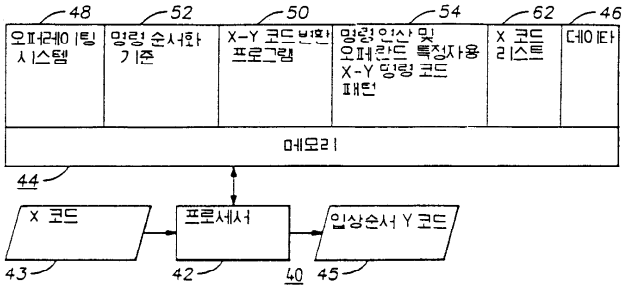
제1명령 세트에 적합한 제1아키텍춰를 갖는 컴퓨터상에서 실행될 수 있는 제1프로그램 코드를 메모리 및 레지스터 상태와 제1명령 세트에 대해 축소되는 제2명령 세트에 적합한 제2아키텍춰를 갖는 제2컴퓨터상에서 실행될 수 있는 제2프로그램 코드로 변환하여 제1코드의 명령 상태 분자를 보전하게끔 제2프로그램 코드를 실행하기 위한 시스템에 있어서, 제1프로그램 코드를 제2프로그램 코드로 변환하기 위한 제1프로세서 및 상기 제1프로세서에 결합된 제1메모리 시스템을 갖는 제1컴퓨터 시스템과 ; 제1코드의 각 연속 명령을 제2코드 명령들로 변환하기 위한 수단과 ; 각 제1코드 명령에 대한 제2코드 명령들을 순서를 이룬 적어도 두개의 그룹, 즉 상태 갱신 이외의 명령 작업을 행하고 상태 에러를 발생치 않고 실행후 중지될 수 있는 제2코드 명령들을 포함하는 제1그룹 및 변환중인 제1코드 명령을 실시하는데 필요한 어떤 특수 기입 명령을 포함하여 모든 메모리 및 레지스터 상태 갱신 명령을 갖는 제2그룹을 가지는 입상 명령 시퀀스로 조직하기 위한 수단과 ; 상기 제2아키텍춰를 가지고 제2프로세서와 상기 제2프로세서에 결합된 제2메모리 시스템을 포함하는 메모리 및 레지스터 상태를 가지며, 상기 제1컴퓨터 시스템에 의해 출력으로서 발생되는 제2코드를 실행하기 위한 제2컴퓨터 시스템과 ; 상기 제2코드의 실행시 상기 제1특수 기입 명령의 메모리 분자의 경합을 실제로 또는 가상적으로 생성하는 각 개입 사건의 발생을 판별하기 위한 수단과 ; 모든 제1그룹 명령이 실행되기 전의 시퀀스 실행 동안 비동기 사건 인터럽트가 발생하는 경우, 혹은 가능한 예외 상황으로 되는 어떤 제2그룹 명령의 실행전에 제1그룹 명령이 실행되었던 경우 제1코드 명령 분자 상태 및 제1코드 명령 입상을 보전하도록 어떤 입상 제2코드 명령 시퀀스를 재시도를 위해 중지시키기 위한 수단과 ; 상기 제1서브 시퀀스가 다수의 단계를 포함하는 단일 기입 명령이고 또 경합 기입이 상기 제1서브 시퀀스의 실행 완료 전에 상기 다른 프로세서에 의해 행하여졌다고 검출되면, 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스내의 상기 제1특수 명령 서브 시퀀스를 성공적인 실행이 완료될 때까지 재시도를 위해 중지시키기 위한 수단과 ; 상기 제1서브 시퀀스가 단일 기입 명령이고 또 비동기 사건 인터럽트가 상기 제1서브 시퀀스의 의도된 실행 동안 발생하는 경우 재시도를 위해 상기 제1서브 시퀀스를 포함하는 어떤 입상 제2코드 명령 시퀀스를 중지시키기 위한 수단과 ; A) 상기 제1서브 시퀀스가 다중 기입 명령이고 입상 명령 시퀀스에 포함되며 비동기 사건 인터럽트가 상기 제1명령 서브 시퀀스의 실행시 제1기입 이후에 발생한다면, 혹은 B) 비동기 사건 인터럽트가 가능한 예외 상황으로 되는 상기 제2그룹내의 모든 상태 갱신 명령의 실행후에 발생한다면, 비동기 사건 인터럽트의 처리를 지연시켜 실행중인 어떤 입상 제2코드 명령 시퀀스를 완료시키기 위한 수단을 구비하는 것을 특징으로 하는 프로그램 코드의 변환 및 실행 시스템.

도면

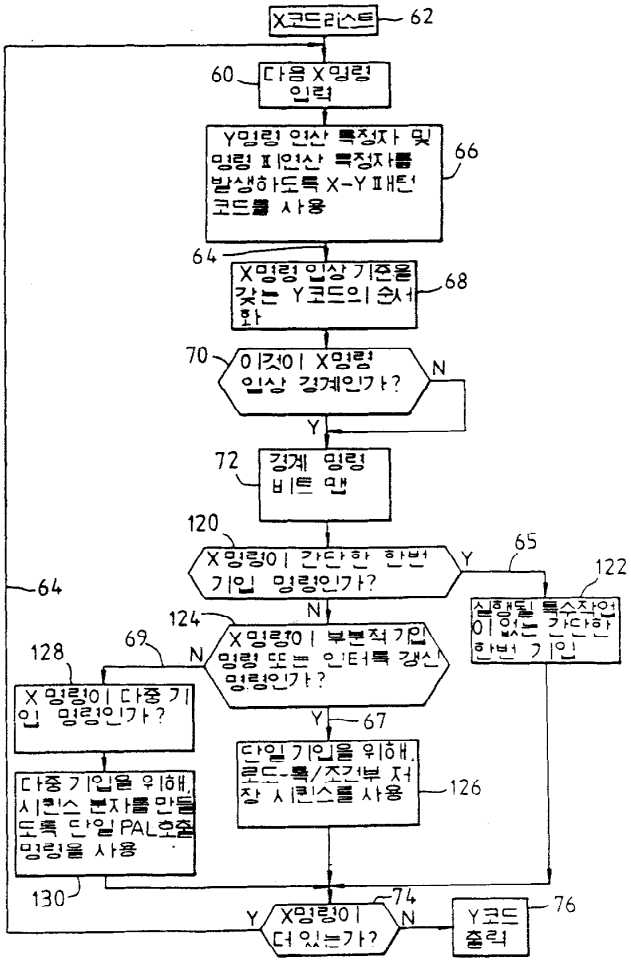
도면1



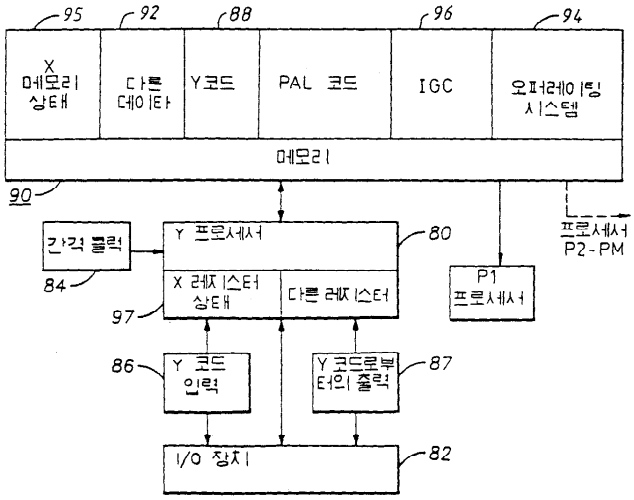
도면2


48
52
50
54
62
46
오퍼레이팅 시스템
명령 순서화 기준
X-Y 코드변환 프로그램
명령인신 및 오퍼랜드 특정사용 X-Y 명령 코드 패턴
X 코드 리스트
데이타
메모리
44
X 코드
프로세서
입상순서 Y 코드
43
42
40
45

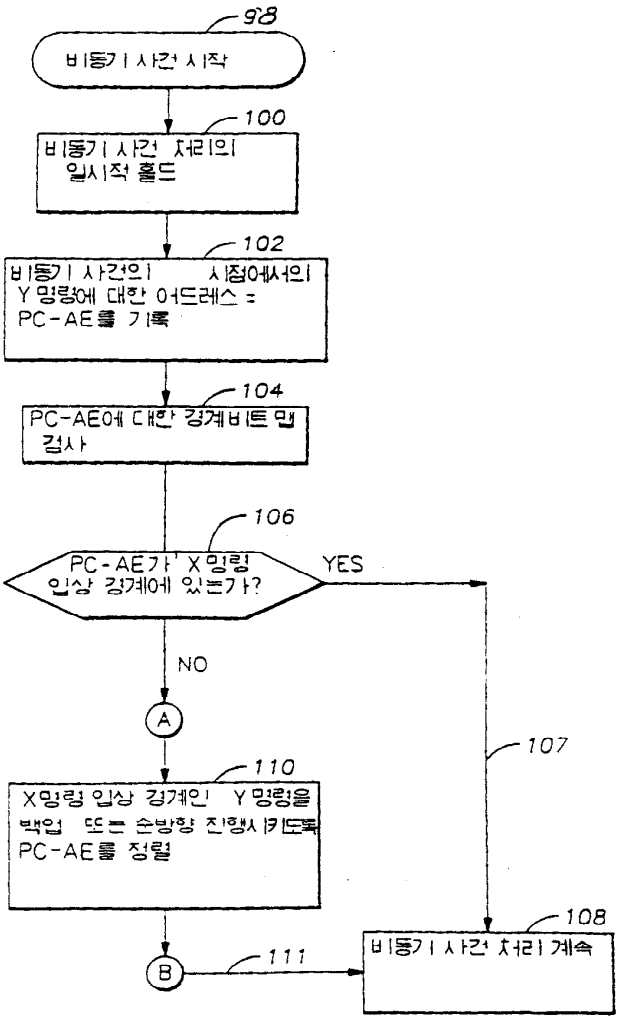
도면3


X코드리스트
62
60
다음 X명령 입력
Y명령 인신 특정자 및 명령 피인신 특정자를 발생하도록 X-Y 패턴 코드를 사용
66
64
X명령 입상 기준을 갖는 Y코드의 순서화
68
70
이것이 X명령 입상 경계인가?
N
Y
72
경계 명령 비트 맵
120
X명령이 간단한 한번 기입 명령인가?
Y
65
122
N
64
124
X명령이 부분적 기입 명령 또는 인터록 갱신 명령인가?
69
N
128
X명령이 다중 기입 명령인가?
Y
67
다중 기입을 위해 시퀀스 본자를 만들도록 단일 PAL호출 명령을 사용
다일 기입을 위해 로드-록/조건부 저장 시퀀스를 사용
126
130
74
Y
X명령이 더 있는가?
N
Y코드 출력
76

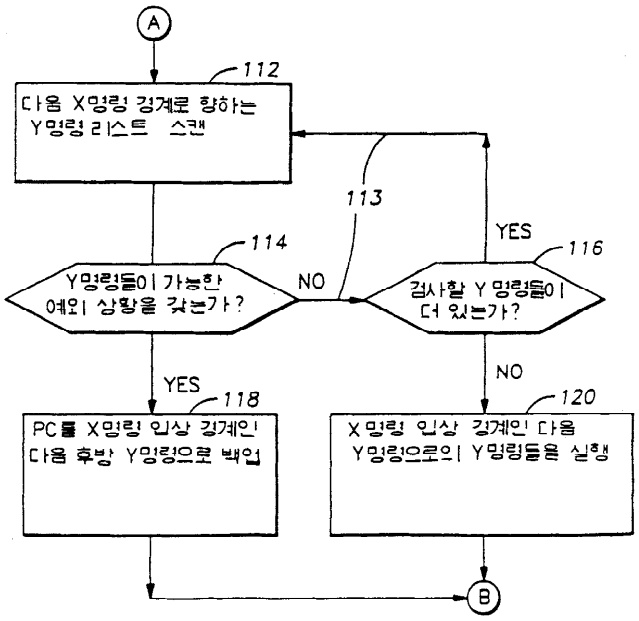
도면4


95
92
88
96
94
X 메모리 상태
다른 데이타
Y코드
PAL 코드
IGC
오퍼레이팅 시스템
메모리
90
간격 클럭
Y 프로세서
80
84
X 레지스터 상태
다른 레지스터
97
P1 프로세서
프로세서 P2-PM
86
Y 코드 입력
Y 코드로부터의 출력
87
I/O 장치
82

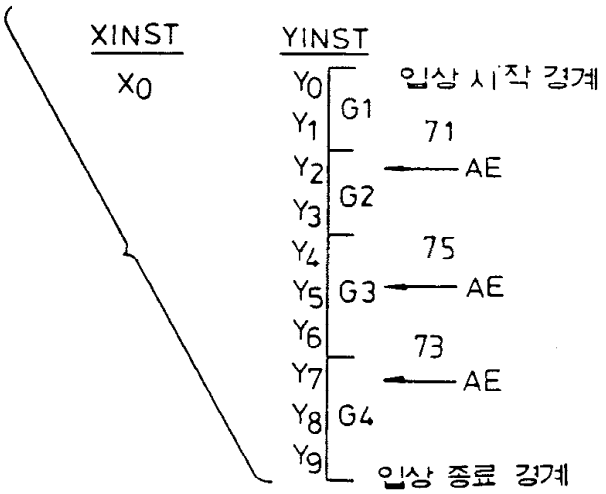
도면5A


98
비동기 사건 시작
100
비동기 사건 처리의 일시적 홀드
102
비동기 사건의 시점에서의 Y 명령에 대한 어드레스 = PC-AE를 기록
104
PC-AE에 대한 경계비트 맵 검사
106
PC-AE가 X 명령 임상 경계에 있는가?
YES
NO
A
107
110
X명령 임상 경계인 Y 명령을 백업 또는 순방향 진행시키도록 PC-AE를 정렬
B
111
108
비동기 사건 처리 계속

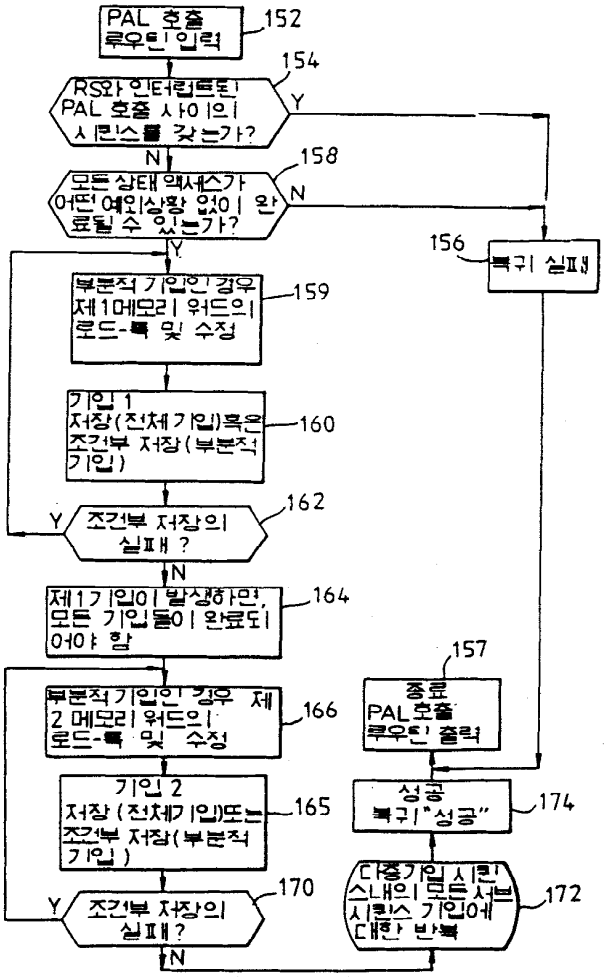
도면5B


A
112
다음 X명령 경계로 향하는 Y명령 리스트 스캔
113
114
Y명령들이 가능한 예외 상황을 갖는가?
NO
YES
116
검사할 Y명령들이 더 있는가?
YES
118
PC를 X명령 입상 경계인 다음 후방 Y명령으로 백업
NO
120
X명령 입상 경계인 다음 Y명령으로의 Y명령들을 실행
B

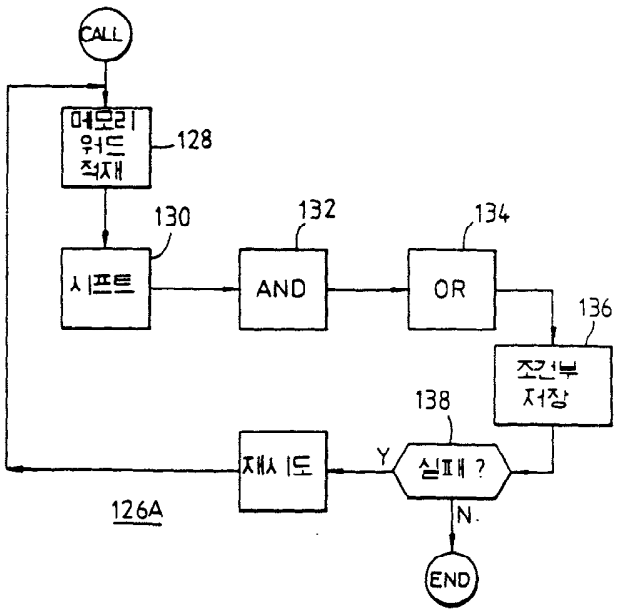
도면6


XINST
X0
YINST
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8
Y9
G1
G2
G3
G4
입상 시작 경계
71
AE
75
AE
73
AE
입상 종료 경계

도면7


PAL 호출 루우틴 입력
152
RS와 인터럽트된 PAL 호출 사이의 시퀀스를 갖는가?
154
Y
N
모든 상태 액세스가 어떤 예외상황 없이 완료될 수 있는가?
158
N
Y
156
복귀 실패
부분적 기입인 경우 제1메모리 워드의 로드-록 및 수정
159
기입 1 저장(전체기입)혹은 조건부 저장(부분적 기입)
160
조건부 저장의 실패?
162
Y
N
제1기입이 발생하면, 모든 기입들이 완료되어야 함
164
부분적 기입인 경우 제2 메모리 워드의 로드-록 및 수정
166
기입 2 저장(전체기입)또는 조건부 저장(부분적 기입)
165
조건부 저장의 실패?
170
Y
N
157
종료 PAL호출 루우틴 출력
성공 복귀 "성공"
174
다중기입 시퀀스내의 모든서브 시퀀스 기입에 대한 반복
172

도면8


CALL
메모리 워드 적재
128
130
시프트
132
AND
134
OR
136
조건부 저장
138
실패?
Y
N
재시도
126A
END