(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl.[6]
G06F 9/30

(45) 공고일자 2001년02월01일
(11) 등록번호 10-0279784
(24) 등록일자 2000년11월03일

(21) 출원번호 10-1997-0031776
(65) 공개번호 특1998-0010763
(22) 출원일자 1997년07월09일
(43) 공개일자 1998년04월30일
(30) 우선권주장 96-182141 1996년07월11일 일본(JP)

(73) 특허권자 가부시키가이샤 히다치 세이사꾸쇼 가나이 쓰도무
일본국 도쿄도 지요다구 간다 스루가다이 4-6
(72) 발명자 데라다 고이치
일본국 가나가와켄 요코하마시 가나자와구 도미오카니시 5-18-15
고지마 게이지
일본국 가나가와켄 사가미하라시 미나미하시모토 1-20-11-405
후지카와 요시후미
일본국 히로시마켄 다카다군 요시다쵸 오아자타케하라 793번지
노지리 도오루
일본국 도쿄도 신쥬쿠구 니시신쥬쿠 3-4-4
니시오카 기요카즈
일본국 가나가와켄 오다와라시 나카무라하라 417-15
(74) 대리인 백남기

심사관 : 오홍수

(54) 연산처리장치

요약

1명령어 레지스터내에 여러개의 명령필드를 갖고 이들 명령을 병렬로 실행하는 처리장치에 관한 것으로서, 상호간에서의 연산결과를 반영(전송)하는데 필요한 시간을 삭감하고 처리프로그램의 스텝수를 저감하기 위해서, 여러개의 처리회로를 갖는 처리장치로서, 각각의 처리회로는 명령을 실행하는 동안에 하나의 처리회로의 명령의 결과에 따른 정보를 적어도 1개의 다른 처리회로로 송신하는 송신회로와 하나의 처리회로에서 송신된 정보를 검토하는 명령을 실행하기 위한 ALU를 포함하는 구성으로 하였다.

이러한 구성으로 하는 것에 의해, 처리프로그램의 스텝수를 저감하는 것이 가능하게 되고, 또 더욱 좁은 면적에서의 회로실장이 가능하고 처리장치 상호간의 데이타 전송을 더욱더 고속으로 실행하는 것이 가능하다는 효과가 얻어진다.

대표도

도1

명세서

도면의 간단한 설명

제1도는 본 발명의 다른 실시예의 처리장치의 개략구성을 도시한 블럭도.

제2도는 C언어를 사용하는 프로그램예를 도시한 설명도.

제3도는 관련 종래기술의 처리장치에 인가하는 형식으로 컴파일한 예를 도시한 설명도.

제4도는 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 예를 도시한 설명도.

제5도는 본 발명의 바람직한 실시예의 처리장치의 개략구성을 도시한 블럭도.

제6도는 본 발명의 바람직한 실시예에서 사용되는 명령어를 도시한 설명도.

제7도는 C언어를 사용하는 프로그램예를 도시한 설명도.

제8도는 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 설명도.

제9도는 바람직한 실시예의 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 설명도.

제10도는 C언어를 사용하는 프로그램예를 도시한 설명도.

제11도는 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 설명도.

제12도는 알림기능을 갖는 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 설명도.

제13도는 프레디키트 레지스터 갱신수단의 개략구성을 도시한 블럭도.

제14도는 본 발명의 또 다른 실시예의 처리장치의 개략구성을 도시한 블럭도.

제15도는 리드값을 알릴 수 있는 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 설명도.

제16도는 본 발명의 응용으로서 정보처리시스템의 개략구성을 도시한 블럭도.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 1명령어 레지스터내에 여러개의 명령필드를 갖고 이들 명령을 병렬로 실행하는 처리장치에 관한 것이다.

전통적인 처리장치는 1명령어로 1개의 처리를 실행하고 또 명령어의 열에 대해서는 직렬적으로 차례로 처리를 실행하도록 구성되어 있는 것이 일반적이었다.

한편, 최근에는 실행속도의 향상을 위해 1개의 명령어만으로 여러개의 명령을 처리할 수 있는 명령체계를 갖고, 이들 명령을 병렬로 실행하는 처리장치가 개발되었다. 이 처리장치는 일반적으로 VLIW(Very Long Instruction Word)형 처리장치라 불리고 있다.

이러한 유형의 처리장치는 병렬로 여러개의 명령을 실행하도록 여러개의 처리장치를 갖는다. 또, 이 처리장치는 각각의 처리장치가 독립해서 처리를 실행할 수 있도록, 여러개의 처리장치에 대응한 여러개의 레지스터 파일을 갖는다. 이와 같은 여러개의 처리장치를 사용해서 특정의 처리를 실행하는 경우, 일반적으로 처리장치 상호간의 데이타통신이 불가결하다. 따라서, 이러한 목적을 위해, 이러한 유형의 처리장치는 예를 들면 여러개의 처리장치 사이에서의 레지스터값 전송수단 또는 여러개의 처리장치에서 액세스할 수 있는 공용레지스터와 같은 수단을 갖는다. 이와 같은 처리장치로서는 예를 들면 일본국 특허공개공보 평성5-233281호에 개시되어 있는 기술이 있다.

또, 상술한 바와 같은 고속화를 실현하기 위한 수단에 부가해서, 처리 자체를 시계열로 여러개의 스테이지로 분할하고, 여러개의 독립된 스테이지가 직렬적으로 처리를 실행하는 처리장치가 있다. 이들 처리장치는 파이프라인형 처리장치라 불리고 있다.

이러한 유형의 처리장치는 명령어가 직렬적으로 배열되어 있는 경우에 최대의 성능을 발휘할 수 있는 것으로 알려져 있다. 한편, 명령어가 직렬적으로 배열되지 않고 예를 들면 조건분기명령이 포함되는 처리인 경우에는 파이프라인의 제어가 방해(혼란)되어 일시적인 성능의 저하가 발생한다.

이와 같은 문제를 극복하기 위해서, 이러한 유형의 처리장치는 조건분기처리를 저감하도록 개량이 실시되고 있다. 그 전형적인 방법은 프레디키트 레지스터(Predicate Register)를 사용하는 방법이다.

프레디키트 레지스터는 명령어를 수식하고, 해당(관련) 명령어를 실행할지 실행하지 않을지를 결정하기 위한 레지스터이다. 프레디키트 레지스터의 사용에 의해서 조건분기명령의 사용빈도를 대폭으로 저감할 수 있게 된다. 후술하는 본 발명의 이해를 위해, 이 성능에 대해서 도면을 사용하여 간단히 설명한다.

제2도는 C언어를 사용하는 프로그램예를 도시한 것이다. 제3도는 제2도의 프로그램을 관련 종래기술의 처리장치에 인가하는 형식으로 컴파일한 예를 도시한 것이다. 제4도는 제2도의 프로그램을 프레디키트 레지스터를 사용하는 처리장치에 인가하는 형식으로 컴파일한 예를 도시한 것이다. 이들 도면에 도시된 바와 같이, 제3도에 있어서 조건분기에 의해 실현되는 산술 또는 논리 처리를 제4도에서는 조건분기처리를 필요로 하지 않고 실현할 수 있다. 제4도의 2행째는 프레디키트 레지스터를 사용하는 명령어를 나타내고 있다. 1행째의 비교명령에 의해서 1번째의 프레디키트 레지스터(p0)에 비교결과의 값을 라이트한다. 2행째의 감산명령은 명령어에 전치(前置)되어 있는 기술 「(p0)」에 의해서, p0에 저장되어 있는 값이 「참」인 경우에만 실행된다. p0에 저장되어 있는 값이 「거짓」인 경우에는 2행째의 감산명령은 리드되더라도 감산처리는 실행되지 않는다. 이와 같은 실행방법에 의해 조건분기처리를 저감할 수 있다.

발명이 이루고자 하는 기술적 과제

그러나, 상기한 바와 같은 고속처리를 실현하기 위한 수단을 병용해서 사용하는 경우, 즉 VLIW형 처리장치의 특징과 파이프라인형 처리장치의 특징을 겸비한 처리장치를 구성하는 경우에는 다음과 같은 문제가 있다.

병렬로 처리를 실행하고 있는 여러개의 처리장치 상호간에서 처리결과의 참조를 실행하기 위해 처리장치 상호간에서의 레지스터 파일의 전송처리가 빈번하게 발생하므로, 병렬처리에 의한 고속화 또는 파이프라인처리에 의한 고속화의 효과를 충분히 얻을 수 없는 경우가 있었다.

또, 전송처리를 위한 프로그램스텝수가 증가하기 때문에 고속처리를 실현할 수 없었다.

본 발명의 목적은 여러개의 처리장치가 상호간에서의 연산결과를 사용하는 처리를 병렬로 실행하는 처리

장치에 있어서, 상호간에서의 연산결과를 반영하는데 필요한 시간을 삭감할 수 있어 고속화가 가능한 처리장치를 제공하는 것이다.

또, 본 발명의 다른 목적은 여러개의 처리장치가 상호간에서의 연산결과를 사용하는 처리를 병렬로 실행하는 처리장치에 있어서, 처리프로그램의 스텝수를 저감할 수 있는 처리장치를 제공하는 것이다.

본 발명의 또 다른 목적은 여러개의 처리장치가 상호간에서의 연산결과를 사용하는 처리를 병렬로 실행하는 처리장치에 있어서, 상호간에서의 연산결과를 반영하는데 필요한 시간을 삭감할 수 있고 또 처리프로그램의 스텝수를 저감하기 위한 명령형식을 생성하고 이 명령을 실행하는 처리장치를 제공하는 것이다.

발명의 구성 및 작용

하나의 관점에 따르면, 본 발명에 의한 처리장치는 여러개의 처리회로를 갖는 처리장치로서, 각각의 처리회로는 명령을 실행하는 동안에 처리회로중의 하나의 처리회로의 명령 결과에 따른 정보를 적어도 1개의 다른 처리회로로 송신하는 송신회로와 상기 하나의 처리회로에서 송신된 정보를 검토(considering)하는 명령을 실행하는 ALU를 포함하는 것을 특징으로 한다.

본 발명에 의한 컴파일러소프트웨어는 기억매체에 저장되어 여러개의 처리회로를 갖는 처리장치에 사용되는 명령열을 생성하는 컴파일러 소프트웨어로서, 컴퓨터에 의해 실행하는 경우에는 상기 컴퓨터가 상기 처리회로에 의해 병렬로 실행될 명령열에 명령을 배열하는 스텝과 상기 처리회로중의 하나의 처리회로에 사용되는 명령필드형식을 각 명령마다 생성하는 스텝을 실행하도록 하고, 상기 명령필드형식은 명령필드의 결과를 적어도 1개의 다른 처리회로로 송신하기 위한 송신필드를 구비하는 것을 특징으로 한다.

본 발명의 다른 관점에 의하면, 여러개의 처리회로를 갖는 처리장치에서 명령을 실행하는 방법으로서, 상기 명령을 하나의 처리회로에서 실행하는 동안에 상기 처리회로중의 하나의 처리회로의 명령 결과를 적어도 1개의 다른 처리회로로 송신하는 스텝과 상기 하나의 처리회로에서 송신된 상기 결과를 검토하는 상기 적어도 1개의 다른 처리회로내의 ALU에서 명령을 실행하는 스텝을 포함하는 것이다.

본 발명의 다른 관점에 의하면, 여러개의 처리회로를 갖는 처리장치를 포함하는 정보처리시스템으로서, 각각의 처리회로는 명령을 실행하는 동안에 상기 처리회로중의 하나의 처리회로의 명령 결과를 적어도 1개의 다른 처리회로로 송신하는 송신회로, 상기 적어도 1개의 다른 처리회로에서 송신된 상기 결과를 검토하는 명령을 실행하기 위한 ALU 및 상기 처리장치용의 명령을 저장하는 메모리를 포함하며, 상기 명령은 상기 하나의 처리회로에 의해 실행된 상기 명령 결과를 상기 적어도 1개의 다른 처리회로에 보고하기 위한 필드를 구비하는 것을 특징으로 한다.

본 발명의 또 다른 관점에 의하면, 1개의 명령어중의 여러개의 명령필드를 병렬로 실행하는 처리장치로서, 여러개의 명령필드군에 대응해서 마련된 여러개의 레지스터, 연산결과에 따라서 여러개의 레지스터에 값을 저장하는 회로 및 상기 여러개의 레지스터에 저장된 값의 평가에 따라서 연산을 실행할지 실행하지 않을지를 선택하는 연산회로를 포함하고, 각 명령필드군은 적어도 1개의 명령필드를 구비하는 처리장치를 마련하는 것이다.

본 발명의 또 다른 관점에 의하면, 1개의 명령어중의 여러개의 명령필드를 병렬로 실행하는 처리장치로서, 상기 여러개의 명령필드중 적어도 1개의 명령필드에 대응해서 마련된 여러개의 연산장치를 포함하고, 각각의 연산장치는 상기 적어도 1개의 명령필드를 유지하는 명령레지스터, 상기 적어도 1개의 명령필드에 대응해서 연산을 실행하는 연산회로 및 상기 연산회로의 연산을 실행할지 실행하지 않을지를 결정하기 위한 값을 저장하는 레지스터를 구비하고, 상기 연산회로는 상기 레지스터에 라이트된 값에 따라서 상기 명령을 실행할지 실행하지 않을지를 결정하는 것을 특징으로 한다.

상기한 각 구성으로 하는 것에 의해, 여러개의 연산을 실행하기 위한 여러개의 처리장치 상호간에서의 연산결과의 전송처리의 회수를 삭감할 수 있고, 그 결과 처리프로그램의 스텝수를 저감하는 것이 가능하게 된다.

이와 같이, 본 발명에 의한 처리장치에 있어서는 고속연산을 가능하게 하는 관점에서 여러개의 처리장치가 상호간에서의 연산결과를 사용하여 연산을 병렬로 실행하는 것이다.

또, 본 발명에 의한 처리장치에 있어서는 처리프로그램의 스텝수를 저감하는 관점에서 여러개의 처리장치가 상호간에서의 연산결과를 사용하여 연산을 병렬로 실행하는 것이다.

또, 본 발명에 의한 처리장치에 있어서는 고속화를 실현할 수 있는 명령형식을 실행하고 또 처리프로그램의 스텝수를 저감하는 관점에서 상호간에서의 연산결과를 사용하여 연산을 병렬로 실행하는 것이다.

상기 이점에 의해, 본 발명에서는 처리장치 전체의 실제성능을 현저하게 향상시킬 수 있다는 효과가 얻어진다.

본 발명은 첨부도면과 함께 이하의 상세한 설명으로부터 명확하게 될 것이다.

본 발명의 그 밖의 목적과 새로운 특징은 첨부도면과 함께 고려될 실시예에 관한 이하의 상세한 설명으로부터 명확하게 될 것이다.

이하, 도면을 참조해서 본 발명의 바람직한 실시예를 상세하게 설명한다.

제5도는 본 발명의 바람직한 실시예의 처리장치의 개략구성을 도시한 것이다. 제5도에 있어서 (1) 및 (2)는 처리장치, (3)은 명령캐시, (4)는 데이타캐시이다. 도면에는 도시하지 않지만, 본 발명의 처리장치, 명령캐시, 데이타캐시는 1칩 LSI에 적용하는 것이 바람직하다. 1칩LSI는 도면에 도시한 구성요소 이외에도 입출력부 등의 구성요소를 더 갖지만, 이들은 본 발명의 내용과는 실질적으로 관계없고 또 주지의 기술로도 구성가능하기 때문에 여기서는 생략하고 있다.

본 실시예에서 사용하는 명령어는 제6도에 도시한 바와 같이, 명령1필드와 명령2필드로 구성된다. 명령1필드와 명령2필드는 각각 프레디키트 레지스터 선택필드(P필드), 프레디키트 레지스터 알림필드

(broadcasting field)(B필드) 및 명령필드(I필드)로 구성된다. 제6도에 있어서 (601)은 명령어, (611)은 명령1필드, (612)는 명령2필드, (621) 및 (631)은 P필드, (622) 및 (632)는 B필드, (623) 및 (633)은 I필드를 각각 나타낸다. 이들 필드중 명령1필드는 처리장치(1)로 공급되고, 명령2필드는 처리장치(2)로 공급된다.

또, 여기에서 나타낸 명령어의 형식(구성)은 어디까지나 1예일 뿐, 동일한 요지이면 구성을 변경해서 실시해도 관계없다. 예를 들면, 명령필드는 항상 2개일 필요는 없고 더욱 많게 예를 들면 4개이어도 좋다.

제5도로 되돌아가서, 처리장치(1)은 다음의 구성요소로 이루어진다. (101)은 명령레지스터로서, 명령캐시(3)에서 리드된 명령중 처리장치(1)에 관한 명령1필드를 유지한다. (102)는 데이타레지스터로서, 처리장치(1)에 있어서 처리를 실행하기 위한 오퍼랜드 데이타를 유지한다. (103)은 명령1필드에 의해 지정된 명령을 실행하기 위한 ALU이다. (105)는 ALU(103)이 명령을 실행할지 실행하지 않을지에 관한 정보를 유지하는 프레디키트 레지스터이다. (106)은 프레디키트 레지스터(105)의 값을 갱신할 때에 입력데이타를 선택하기 위한 셀렉터이다.

처리장치(2)는 처리장치(1)과 동일한 구성요소로 이루어진다. 즉, 처리장치(2)는 명령레지스터(201), 데이타 레지스터(202), ALU(203), 프레디키트 레지스터(205), 셀렉터(206)으로 구성된다.

여기서, 데이타 레지스터(102) 및 (202), 프레디키트 레지스터(105) 및 (205)는 선택적인 사용을 위해 각각이 여러개의 레지스터영역을 갖고 있다. 예를들면, 32개의 데이타레지스터영역을 갖는 경우, 그들 영역중에서 1개의 데이타레지스터영역만을 선택하기 위해서, 5개의 데이타레지스터영역 선택신호를 데이타레지스터(102) 및 (202)에 각각 입력하게 된다. 이 경우, 각 데이타레지스터영역은 예를 들면 r0, rl, r2와 같이 데이타레지스터영역에 번호를 할당하는 것에 의해 일반적으로 식별할 수 있다. 또한, 이상 설명한 내용은 데이타레지스터 뿐만 아니라 프레디키트 레지스터에도 적용할 수 있다.

다음에, 본 실시예의 동작에 대해서 설명한다. 또한, 여기에서 설명하는 것 이외의 통상의 ALU에 의한 연산에 대해서는 관련 종래기술 및 그의 상세한 설명에 의해 실현할 수 있으므로 여기에서는 설명을 생략한다.

먼저, 제4도에 도시한 명령열을 실행할 때의 연산에 대해서 설명한다. 제4도는 1개의 처리장치만으로 처리가능한 3명령어로 이루어지는 명령열의 예를 도시한 것이다.

제1명령어는 비교명령(cmp.gt)으로서, 데이타레지스터영역1(rl)과 즉시값 “2” 를 비교하고, 데이타레지스터영역1 쪽이 큰 경우에 프레디키트 레지스터영역0(p0)에 「참」을, 그렇지 않은 경우에는 「거짓」을 라이트한다. 여기에서, 프레디키트 레지스터(105)에 포함되는 여러개의 프레디키트 레지스터영역중 값이 라이트될 프레디키트 레지스터영역은 명령1필드의 프레디키트 레지스터 선택필드(P필드)의 일부를 사용해서 지정되고 있다. 또, 값을 라이트하기 위한 프레디키트 레지스터영역은 예를 들면 명령필드(I필드)의 일부를 사용하는 것에 의해 지정할 수도 있다.

제2명령어는 감산명령(sub)이다. 이 명령에서는 rl에서 즉시값 “1” 을 감하고, 그 결과를 rl에 라이트백(write back)한다 단, 이 명령어는 수식어 “(p0)” 의 뒤에 배치되므로, 만약 p0이 「참」이면 이 감산은 실행되지만, 만약 p0이 「거짓」이면 이 감산은 실행되지 않는다.

제3명령어는 가산명령(add)이다. 이 명령에서는 데이타 레지스터영역 2(r2)와 rl을 가산하고, 그 결과를 r2에 라이트백한다.

다음에, 본 실시예를 제7도, 제8도 및 제9도를 참조해서 설명한다. 제7도는 C언어를 사용하는 프로그램의 다른 예를 도시한 것, 제8도는 제7도의 프로그램을 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 것이다. 제8도에 있어서 (501)은 처리장치(1)에 인가하는 명령어열, (502)는 처리장치(2)에 인가하는 명령어열이다. 여기에서는, 설명을 간단하게 하기 위해서 처리장치(1)과 처리장치(2)가 동일한 처리속도로 명령어를 처리하는 것으로서 설명한다.

우선, 처리장치(2)는 제1명령어(add)에 의해 r3과 r4를 가산하고, 그 결과를 r5에 저장한다. 동시에, 처리장치(1)은 제1명령어(add)에 의해 rl과 r2를 가산하고, 그 결과를 r7에 저장한다.

다음의 스텝에서, 처리장치(1)은 제2명령어(mul)에 의해 r7과 r6의 승산을 실행하고, 그 결과를 r7에 저장한다. 동시에, 처리장치(2)는 제2명령어(xfer)에 의해 데이타레지스터의 값을 전송한다. 이 예에서는 데이타레지스터(202)에 포함되는 데이타레지스터영역5(r5)가 유지하고 있는 값을 데이타 레지스터(102)에 포함되는 데이타레지스터영역5(r5)에 라이트한다.

다음의 스텝에서, 처리장치(1)과 처리장치(2)는 모두 제3명령어(cmp.gt)에 의해 비교명령을 실행하고, 처리의 결과로서 구해진 값인 비교 결과를 프레디키트 레지스터영역0(p0)에 각각 저장한다.

또, 다음의 스텝에서는 처리장치(1)은 p0의 값이 「참」인 경우에 제4명령어(add)에 의해 r7과 즉시값1의 가산을 실행하고, 그 결과를 r7에 저장한다. 동시에, 처리장치(2)는 p0의 값이 「참」인 경우에 제4명령어(add)에 의해 r5와 즉시값2의 가산을 실행하고, 그 결과를 r5에 저장한다.

다음에, 제9도는 제7도의 프로그램을 본 실시예의 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 것이다. 제9도에 있어서, (511)은 처리장치(1)에 인가하는 명령어열, (512)는 처리장치(2)에 인가하는 명령어열이다.

이 예의 특징은 명령어열(512)의 제2명령어(cmp.gt)에 부가되어 있는 B파라미터이다. 이 파라미터는 명령필드중의 프레디키트 레지스터 알림필드(B필드)에 대응하고 있고, 다른 처리장치에 대해서 프레디키트 레지스터의 갱신을 알릴것인지 알리지 않을 것인지를 결정한다. 물론, 알림동작은 그 결과를 여러개의 ALU 중 1개 이상의 ALU로 통신, 송신 또는 보고하는 경우에 실행할 수 있다. 이 제2명령어를 실행하면, 상기 제2명령어의 실행주기의 종료에 의해 처리장치(2)의 프레디키트 레지스터영역(0)(p0)에 유지되어 있는 값이 제2명령어의 결과에 따른 값으로 갱신될 뿐만 아니라, 다른 처리장치, 이 예에서는 처리장치(1)의 프레디키트 레지스터영역0(p0)에 유지되어 있는 값도 같은 방식으로 갱신된다. 이러한 동작(연산)은 처리장

치(1) 및 처리장치(2)의 셀렉터(106) 및 (206)(제5도)에 의해 실현된다.

명령어열(512)의 B파라미터는 처리장치(1)의 셀렉터(106)과 처리장치(2)의 셀렉터(206)을 모두 제어한다.

이들 명령어열(511) 및 (512)를 발생시키는 컴파일러는 B파라미터를 나타내는 명령어(511) 및 (512)를 동시에 생성하지 않는다.

통상동작의 경우, 프레디키트 레지스터 알림필드(B필드)의 값은 「거짓」이다. 이 경우 셀렉터는 동일한 처리장치내의 ALU에 의해 생성된 프레디키트 레지스터로의 라이트데이타를 선택하고, 이 데이타를 프레디키트 레지스터에 저장한다. 그러나, 명령어내에 B파라미터가 지정된 것에 의해서 프레디키트 레지스터 알림필드(B필드)의 값이 「참」인 경우, 셀렉터는 B필드가 「참」인 처리장치에서 출력되는 라이트데이타를 선택하고, 이 데이타를 프레디키트 레지스터에 저장한다.

이와 같은 다른 처리장치에 대해서 갱신을 알리는 것이 가능한 특징을 갖는 프레디키트 레지스터 갱신수단을 마련하는 것에 의해서, 데이타레지스터의 불필요한 전송처리를 삭감하는 것이 가능하게 된다.

계속해서, 본 실시예의 다른 예를 제10도, 제11도 및 제12도에 도시한다. 제10도는 C언어를 사용하는 프로그램예를 도시한 것이다. 제11도는 제10도의 프로그램을 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 것이다. 제12도는 제10도의 프로그램을 알림기능을 갖는 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 것이다.

이 실시예에서 처리장치는 3개 이상의 처리장치를 갖는 것을 전제로 하고 있다. 도면에서 알 수 있는 바와 같이, 상술한 예와 마찬가지로 알림기능을 유효하게 이용할 수 있는 명령어열을 생성하는 것에 의해 더욱 간단한 명령어열로 하는 것이 가능하게 된다.

본 발명의 컴파일러는 또한 다음과 같은 특징을 갖는다. 제7도의 C언어 명령열을 컴파일하는데 있어서, 컴파일러가 수정하는 “e”와 “g”가 서로 독립적이라고 검출한 경우, “e”의 연산과 “g”의 연산은 각각의 처리장치에서 실행되는 것으로 결정하고, 컴파일러가 수정하는 “e”와 “g”가 조건식의 결과에 따른다고 검출한 경우, 컴파일러는 1개의 처리장치가 조건식을 실행하고, 그 결과를 다른 처리장치로 보고한다고 결정한다. 그 결과, 제9도의 명령열이 생성된다.

다음에, 본 발명의 다른 실시예를 제1도를 참조해서 설명한다. 제1도는 제5도와 같이 처리장치의 개략구성을 도시한 것으로서, 이 처리장치는 1칩 LSI를 구성하는 것이 바람직하다. 제1도의 실시예는 제5도의 실시예와 구성면에서 기본적으로 다르다. 즉, 1개의 처리장치에 내장되어 있는 ALU의 개수가 다른 점(제5도에서는 1개, 제1도에서는 2개)과 프레디키트 레지스터로의 라이트데이타의 알림이 셀렉터가 아니라 버스(100)을 거쳐서 실행된다는 점이 구성상의 주요상이점이다. 이하의 설명에서는 제1실시예와의 구성 또는 동작상의 상이점에 대해서 설명한다.

본 실시예에서는 명령어는 명령1필드, 명령2필드, 명령3필드, 명령4필드의 4개로 구성된다. 각 명령필드의 구성요소는 제1실시예와 동일하다. 이들 명령필드중 명령1필드와 명령2필드는 제1명령필드군으로서 처리장치(1)로, 명령3필드와 명령4필드는 제2명령필드군으로서 처리장치(2)로 각각 공급된다. 각 처리장치는 2개의 ALU를 구비하고 있으므로, 명령레지스터(101) 및 (201)은 동시에 2개의 명령필드를 저장하도록 구성되어 있다.

ALU에서 출력되는 프레디키트 레지스터에 라이트하기 위한 값은 각각 처리장치 내부의 프레디키트 레지스터에 라이트될 뿐만 아니라 프레디키트 레지스터에 알리기 위해 사용되는 버스로 출력할 수도 있다. 이 알림버스를 경유해서 각 ALU에서 출력되는 프레디키트 레지스터에 라이트하기 위한 값은 모든 처리장치의 지정된 프레디키트 레지스터 영역에 동시에 알려져 저장된다.

이와 같은 구성을 채용하는 것에 의해, 처리장치의 개수나 ALU의 개수를 증가시킨 경우에도 다입력 셀렉터에 의한 신호지연이나 많은 신호선을 독립적으로 접속하기 위한 실장면적의 증대를 최소한으로 억제할 수 있다.

다음에, 제13도는 제1도에서 도시한 프레디키트 레지스터 갱신수단에 대해서 상세하게 도시한 것이다. 제13도에 있어서, (11)은 어느 ALU로 부터의 출력을 알릴 것인지를 결정하는 프레디키트 레지스터 알림(報知)조정회로, (12)는 회로(11)에 의해 결정된 출력을 선택하기 위한 프레디키트 레지스터 알림셀렉터, (13)은 알림에 관계된 예외발생회로, (14)는 예외발생회로에서 발생되는 예외신호, (15)는 예외신호(14)를 받는 예외처리회로, (16)은 예외처리회로(15)에서 출력되는 연산정지신호, (107) 및 (207)은 프레디키트 레지스터(105) 및 (205)에 저장되어 있는 값에 따라서 ALU(103) 및 (203)을 동작시킬지 동작시키지 않을지를 결정하는 명령마스크수단(논리곱회로), (108) 및 (208)은 프레디키트 레지스터 라이트값이 알려졌을 때 그것을 프레디키트 레지스터로 공급하는 프레디키트 레지스터 알림데이타 공급수단(논리합회로)이다. 각 구성요소는 1칩 LSI로서 구성되어야 한다. 또한, 제13도중의 점선은 이 설명과는 직접적으로 관계가 없지만, ALU와 데이타 레지스터를 접속하는 회로를 나타내고 있다.

프레디키트 레지스터 알림 조정회로(11)은 명령레지스터(101) 및 (201)에 포함되는 B필드의 값에 따라서 어느 ALU의 연산결과를 알릴 것인지를 결정하고, 프레디키트 레지스터 알림셀렉터(12)에 전달한다. 결정을 위한 방법은 자유롭게 선택할 수 있지만, 예를 들면 처리장치에 미리 설정한 고정의 우선순위에 의해 결정하는 방법이나 여러개의 처리장치로부터 동시에 알리기 위해 자원충돌의 에러로서 예외를 발생하는 방법 등이 고려된다. 예외발생회로(13)은 그와 같은 예외신호(14)를 발생하기 위한 회로로서, 이 예외신호(14)에 따라서 예외처리회로(15)는 연산정지신호(16)을 공급한다. 이것에 의해, 처리장치의 연산이 정지되고 리셋대기상태로 된다.

프레디키트 레지스터 알림셀렉터(12)는 조정회로(11)의 결정에 따라서 필요하다면 각 처리장치로 부터의 연산결과를 알린다.

알려진 데이타는 프레디키트 레지스터 알림데이타 공급수단(108) 및 (208)에 의해 프레디키트 레지스터(105) 및 (205)로 공급된다. 프레디키트 레지스터 알림데이타 공급수단(108) 및 (208)은 단순한 논리합을 구하는 회로에 의해 구성된다. 이 회로는 반드시 논리합을 구하는 수단일 필요는 없다. 예를 들면 알림데

이타가 공급되고 있을 때에는 알림데이타를, 공급되고 있지 않을 때에는 처리장치 내부의 ALU에서 출력된 데이타를 선택적으로 공급하는 선택회로이어도 좋다.

프레디키트 레지스터(105) 및 (205)는 여러개의 기억영역을 갖는 것으로 상정하고 있다. 프레디키트 레지스터로의 값의 저장시에 어느 프레디키트 레지스터영역에 값을 저장할지는 명령레지스터(101) 및 (201)에 저장되어 있는 오퍼랜드의 값에 의해서 결정된다. 또, 프레디키트 레지스터로 부터의 값의 리드시에 어느 프레디키트 레지스터영역에서 값을 리드할 것인지도 명령레지스터(101)및 (201)에 저장되어 있는 오퍼랜드의 값에 의해 결정된다. 이들 프레디키트 레지스터영역의 선택에 관한 정보는 명령레지스터의 P필드 또는 I필드에 저장된다.

이상의 실시예에서는 처리장치 상호간에서의 프레디키트 레지스터의 값의 선택은 프레디키트 레지스터로의 라이트데이타를 선택하는 방법에 의해 실현하고 있었다. 즉, 어느 ALU의 출력을 프레디키트 레지스터에 라이트할지를 전환하는 것에 의해서 프레디키트 레지스터의 값의 선택을 실행하고 있었다. 그러나, 프레디키트 레지스터의 값의 선택은 프레디키트 레지스터로부터의 리드데이타를 선택하는 방법을 사용해서 실행할 수도 있다.

이 방법을 사용하는 실시예를 제14도, 제15도를 참조해서 설명한다. 제15도는 제10도의 프로그램을 리드값을 알리는 기능을 갖는 프레디키트 레지스터형 처리장치에 인가하는 형식으로 컴파일한 명령어열의 예를 도시한 것이다. 여기에서는, 참조할 프레디키트 레지스터를 나타내기 위해서, 프레디키트 레지스터를 나타내는 명령어에 처리장치의 번호를 지정하고 있다. 예를 들면, 처리장치(2)의 제3명령어의 선두부의 “(p0:1)” 은 처리장치(1)의 프레디키트 레지스터영역1(p1)을 참조해야 하는 것을 규정하고 있다.

제14도는 상술한 기능을 실현하는 본 실시예의 처리장치의 개략구성을 도시한 것이다. 즉, 이 실시예가 제1도의 실시예와 다른 구성은 프레디키트 레지스터의 라이트값을 알리는 것이 아니라, 프레디키트 레지스터로부터의 리드값을 알리는 점이다. 이하의 설명에서는 주로 상기의 실시예와 구성 또는 동작이 다른 점에 대해서 설명한다.

각 ALU에서 출력되는 프레디키트 레지스터에 라이트하기 위한 값은 처리장치 내부의 프레디키트 레지스터에 각각 라이트된다. 각 프레디키트 레지스터에서 리드되는 값은 프레디키트 알림용의 버스(140)으로 출력된다. 이 버스를 경유해서 각 ALU는 어느 처리장치에 저장된 프레디키트 레지스터의 값도 참조할 수 있다. 어느 프레디키트 레지스터의 값을 참조할지는 제15도에서 상술한 바와 같이 프레디키트 레지스터를 참조하는 명령어중에 지정된 처리장치의 번호에 의해 결정된다.

이와 같은 구성을 취하는 것에 의해, 처리장치의 개수나 ALU의 개수를 증가시킨 경우에도 다입력 셀렉터에 의한 신호지연이나 많은 신호선을 독립적으로 접속하기 위한 실장면적의 증대는 최소한으로 억제할 수 있다.

또, 본 발명은 상술한 실시예에만 한정되는 것은 아니고, 그 요지를 이탈하지 않는 범위내에서 적절히 내용을 변경해서 실시할 수 있다.

예를 들면, 실시예중에서는 처리장치를 최대 2개 사용하고 또한 1처리장치당의 ALU는 최대 2개로 해서 사용하는 것으로 기술하고 있지만, 이 값은 어떠한 제약도 없고 필요에 따라서 처리장치나 ALU의 개수를 증가시키거나 감소시킬 수 있다.

또, 실시예 중에서는 명령어의 구성이나 명령필드의 구성에 대해서 설명하고 있지만, 이 구성 자체도 한정되지 않고 자유롭게 변경가능하다.

또, 실시예 중에서는 프레디키트 레지스터는 특정의 버스에서만 데이타 라이트/리드가 실행되도록 기술하고 있지만, 이것은 설명의 간단화를 위한 것이다. 예를 들면, 임의의 데이타 레지스터와 임의의 프레디키트 레지스터 사이에서 데이타를 전송하는 구성으로 되어도 지장없다.

제16도는 멀티미디어 데이타를 처리하는 서브프로세서로서 본 발명을 사용하는 정보처리 시스템을 도시한 것이다. 제16도에 있어서 (20)은 정보처리 시스템 전체를 관리하는 메인프로세서, (21)은 메인프로그램을 저장하는 메인메모리, (22)는 제1도, 제5도, 제13도 및 제14도에 있어서 상술한 바와 같이 본 발명에 따라 동작하는 서브프로세서, (23)은 명령어열이 저장된 서브메모리, (24)는 서브프로세서(22)용의 멀티미디어 데이타를 관리하는 멀티미디어 I/O회로, (25)는 결과를 표시하거나 연산을 지정하는 표시회로, (26)은 음성데이타를 입출력하는 음성회로, (27)은 접속된 네트워크를 거쳐서 데이타를 통신하는 네트워크회로, (28)은 주변장치를 제어하는 주변회로, (29)는 연산기의 명령 등을 입력하는 키보드장치, (30)은 서브프로세서에서 실행되는 명령어열 또는 프로그램용으로 사용되는 데이타 및 상기 메인프로그램을 기억하는 디스크기억장치, (31)은 정보가 흐르는 버스이다.

본 발명의 시스템에 의하면, 스텝수를 저감할 수 있으므로 고속처리가 가능할 뿐만 아니라 서브 메모리와 기억장치의 용량을 저감할 수가 있다.

발명의 효과

이상 설명한 바와 같이 본 발명에 의하면, 여러개의 처리장치를 갖고 여러개의 연산을 병렬로 실행하는 처리장치에 있어서, 여러개의 처리장치 상호간에서의 연산결과의 전송처리의 회수를 삭감할 수 있는 명령포맷을 사용하고 있다. 이것에 의해, 처리프로그램의 스텝수를 저감하는 것이 가능하게 된다.

또, 처리장치 상호간에서 프레디키트 레지스터의 갱신에 필요로 되는 신호선수는 데이타 레지스터의 내용을 전송하는데 필요한 비트폭보다 작으므로, 더욱 좁은 면적에서의 회로실장이 가능하고 처리장치 상호간의 데이타 전송을 더욱더 고속으로 실행하는 것이 가능하다.

이상, 본 발명을 첨부도면에 따라서 구체적으로 설명했지만, 본 발명은 상기 실시예에 한정되는 것은 아니고, 그 요지를 이탈하지 않는 범위에서 여러가지 변경가능한 것은 물론이다.

(57) 청구의 범위

청구항 1

여러개의 처리회로를 갖는 처리장치로서, 각각의 처리회로는 명령을 실행하는 동안에 상기 처리회로중의 하나의 처리회로의 명령의 결과에 따른 정보를 적어도 1개의 다른 처리회로로 송신하는 송신회로와; 상기 하나의 처리회로에서 송신된 정보를 검토하는 명령을 실행하기 위한 ALU를 포함하는 것을 특징으로 하는 처리장치.

청구항 2

제1항에 있어서, 각각의 처리회로는 상기 적어도 1개의 다른 처리회로로부터의 정보와 상기 ALU로부터의 정보에서 상기 명령중의 하나의 정보를 상기 ALU에서 선택적으로 이용할 수 있도록 하는 셀렉터를 더 포함하는 것을 특징으로 하는 처리장치.

청구항 3

제1항에 있어서, 상기 명령의 형식은 상기 하나의 처리회로의 결과를 상기 적어도 1개의 다른 처리회로에 보고하기 위한 필드를 구비하는 것을 특징으로 하는 처리장치.

청구항 4

여러개의 처리회로를 갖는 처리장치에서 명령을 실행하는 방법으로서, 상기 명령을 하나의 처리회로에서 실행하는 동안에 상기 처리회로중의 하나의 처리회로의 명령 결과를 적어도 1개의 다른 처리회로로 송신하는 스텝과; 상기 하나의 처리회로에서 송신된 상기 결과를 검토하는 상기 적어도 1개의 다른 처리회로내의 ALU에서 명령을 실행하는 스텝을 포함하는 것을 특징으로 하는 명령실행방법.

청구항 5

제4항에 있어서, 상기 적어도 1개의 다른 처리회로의 결과와 상기 ALU의 결과에서 상기 명령중의 하나의 결과를 상기 ALU에서 선택적으로 이용할 수 있도록 선택하는 스텝을 더 포함하는 것을 특징으로 하는 명령실행방법.

청구항 6

여러개의 처리회로를 갖는 처리장치를 포함하는 정보처리시스템으로서, 각각의 처리회로는 명령을 실행하는 동안에 상기 처리회로중의 하나의 처리회로의 명령 결과를 적어도 1개의 다른 처리회로로 송신하는 송신회로; 상기 적어도 1개의 다른 처리회로에서 송신된 상기 결과를 검토하는 명령을 실행하기 위한 ALU 및; 상기 처리장치용의 명령을 저장하는 메모리를 포함하며, 상기 명령은 상기 하나의 처리회로에 의해 실행된 상기 명령 결과를 상기 적어도 1개의 다른 처리회로에 보고하기 위한 필드를 구비하는 것을 특징으로 하는 정보처리시스템.

청구항 7

제6항에 있어서, 상기 처리장치는 상기 적어도 1개의 다른 처리회로의 결과와 상기 ALU의 결과에서 상기 명령중의 하나의 결과를 상기 ALU에서 선택적으로 이용할 수 있도록 하는 셀렉터를 더 포함하는 것을 특징으로 하는 정보처리시스템.

청구항 8

1개의 명령어중의 여러개의 명령필드를 병렬로 실행하는 처리장치로서, 여러개의 명령필드군에 대응해서 마련된 여러개의 레지스터; 연산결과에 따라서 여러개의 레지스터에 값을 저장하는 회로 및; 상기 여러개의 레지스터에 저장된 값의 평가에 따라서 연산을 실행할지 실행하지 않을지를 선택하는 연산회로를 포함하고, 각 명령필드군은 적어도 1개의 명령필드를 구비하는 것을 특징으로 하는 처리장치.

청구항 9

제8항에 있어서, 대응하는 명령필드군의 연산 결과를 평가하는 것에 의해 얻어진 값과 상기 대응하는 명령필드군 이외의 군의 연산 결과를 평가하는 것에 의해 얻어진 값을 여러개의 레지스터에 선택적으로 저장하는 회로를 더 포함하고, 상기 여러개의 레지스터는 각각 상기 대응하는 명령필드군에 의해서만 평가되는 것을 특징으로 하는 처리장치.

청구항 10

제8항에 있어서, 상기 대응하는 명령필드군의 상기 레지스터에 저장된 값과 상기 대응하는 명령필드군 이외의 군의 상기 레지스터에 저장된 값을 선택적으로 평가하는 회로를 더 포함하고, 상기 여러개의 레지스터는 어느 하나의 명령필드군에 의해 평가되는 것을 특징으로 하는 처리장치.

청구항 11

제9항에 있어서, 상기 여러개의 레지스터에 값을 저장하는 회로는 상기 다른 명령필드군의 연산결과를 평가하는 것에 의해 얻어진 값이 송신되어 있는 경우에는 이 송신되어 있는 값을 선택해서 저장하고, 그렇지 않은 경우에는 상기 대응하는 명령필드군의 연산 결과를 평가하는 것에 의해 얻어진 값을 선택해서 저장하는 것을 특징으로 하는 처리장치.

청구항 12

제10항에 있어서, 상기 여러개의 레지스터에 저장되어 있는 값을 평가하는 회로는 상기 여러개의 명령필

드군의 모두를 향해 상기 레지스터중의 어느 하나에 저장되어 있는 값이 알려져 있는 경우에는 이 알려져 있는 값을 선택해서 평가하고, 그렇지 않은 경우에는 상기 대응하는 명령필드군의 상기 레지스터에 저장되어 있는 값을 선택해서 평가하는 것을 특징으로 하는 처리장치.

청구항 13

제11항에 있어서, 상기 명령필드는 상기 적어도 1개의 다른 명령필드군을 향해 연산결과를 평가한 값을 송신할 것인지 송신하지 않을 것인지를 결정하기 위한 명령비트열을 내포하고 있는 것을 특징으로 하는 처리장치.

청구항 14

제12항에 있어서, 상기 명령필드는 상기 적어도 1개의 다른 명령필드군을 향해 상기 여러개의 레지스터중의 어느 하나에 저장되어 있는 값을 송신할 것인지 송신하지 않을 것인지를 결정하기 위한 명령비트열을 내포하고 있는 것을 특징으로 하는 처리장치.

청구항 15

제11항에 있어서, 상기 연산 결과를 평가하는 것에 의해 얻어진 값이 2개 이상의 명령필드군에서 송출되고 있는 것을 검출하는 회로 및; 상기 검출결과에 따라서 예외신호를 발생시키는 회로를 더 포함하는 것을 특징으로 하는 처리장치.

청구항 16

제12항에 있어서, 상기 레지스터중의 어느 하나에 저장되어 있는 값이 2개 이상의 명령필드군에서 송출되고 있는 것을 검출하는 회로 및; 상기 검출결과에 따라서 예외신호를 발생시키는 회로를 더 포함하는 것을 특징으로 하는 처리장치.

청구항 17

제16항에 있어서, 상기 연산 결과를 평가하는 것에 의해 얻어진 값은 공유버스선을 거쳐 송신되는 것을 특징으로 하는 처리장치.

청구항 18

1개의 명령어중의 여러개의 명령필드를 병렬로 실행하는 처리장치로서, 상기 여러개의 명령필드중 적어도 1개의 명령필드에 대응해서 마련된 여러개의 연산장치를 포함하고, 각각의 연산장치는 상기 적어도 1개의 명령필드를 유지하는 명령레지스터; 상기 적어도 1개의 명령필드에 대응해서 연산을 실행하는 연산회로 및; 상기 연산회로의 연산을 실행할지 실행하지 않을지를 결정하기 위한 값을 저장하는 레지스터를 구비하고, 상기 연산회로는 상기 레지스터에 라이트된 값에 따라서 상기 명령을 실행할지 실행하지 않을지를 결정하는 것을 특징으로 하는 처리장치.

청구항 19

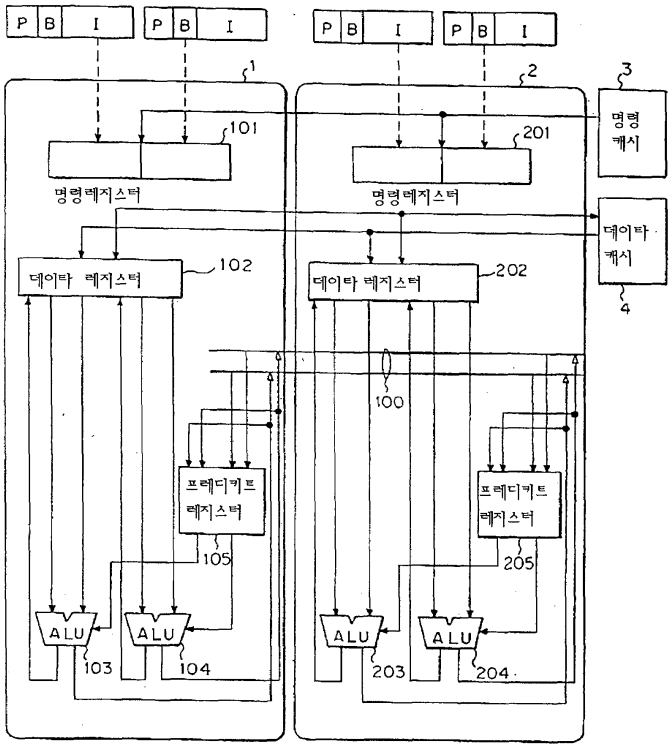
제18항에 있어서, 미리 정해진 명령의 연산 결과를 평가하는 것에 의해 얻어진 값을 적어도 1개의 연산장치내의 레지스터에 라이트하는 회로를 더 포함하고, 상기 연산회로는 상기 레지스터를 지정받은 상기 명령을 실행할지 실행하지 않을지를 결정하는 것을 특징으로 하는 처리장치.

청구항 20

제18항에 있어서, 미리 정해진 명령의 연산 결과를 평가하는 것에 의해 얻어진 값을 자신의 연산장치내의 레지스터에 라이트하는 회로를 더 포함하고, 상기 연산회로는 어느 하나의 연산장치내의 상기 레지스터를 지정받은 명령을 실행할지 실행하지 않을지를 결정하는 것을 특징으로 하는 처리장치.

도면

도면1



도면2

```
001   if ( a > 2 )
002   {
003         a = a - 1;
004   }
005   b = b + a;
```

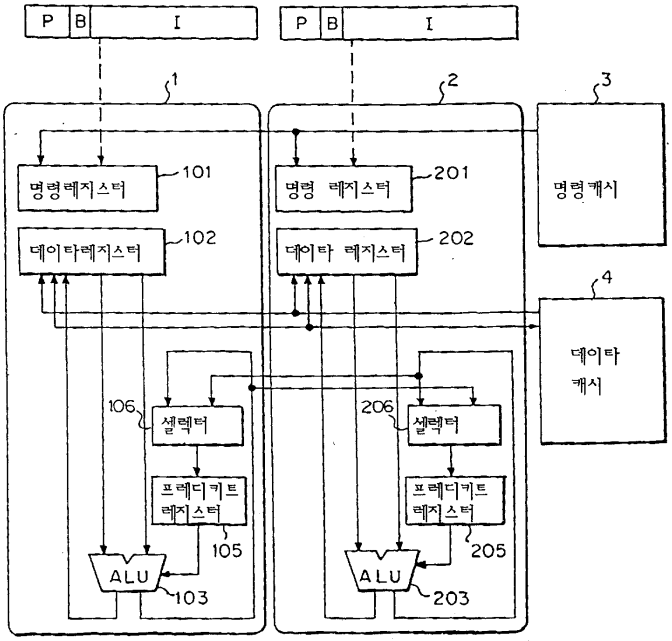
도면3

```
001         cmp  r1, #2
002         ble  $1
003         sub  r1, #1, r1
004   $1:   add  r2,  r1, r2
```

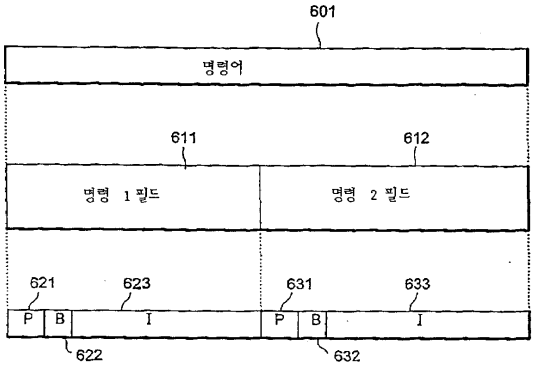
도면4

```
001         cmp.gt r1, #2, p0
002   (p0)  sub    r1, #1, r1
003         add    r2,  r1, r2
```

도면5



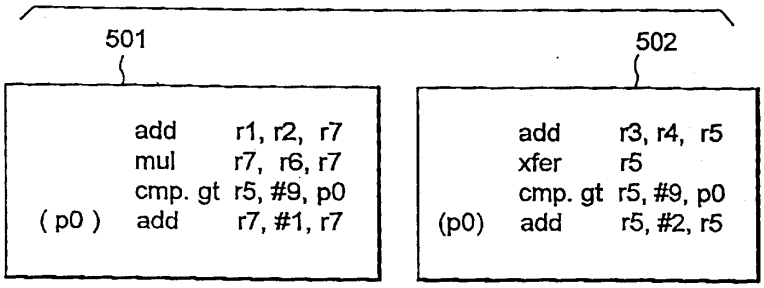
도면6



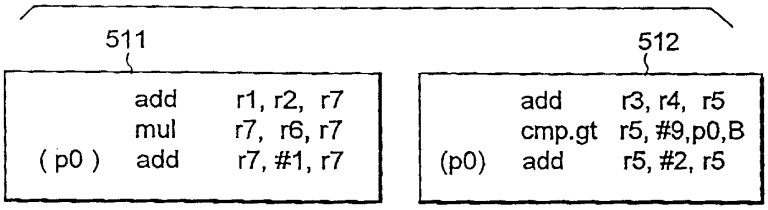
도면7

```
e = c + d;
g = (a + b) * f;
if ( e > 9 )
{
        g ++;
        e = e + 2;
}
```

도면8



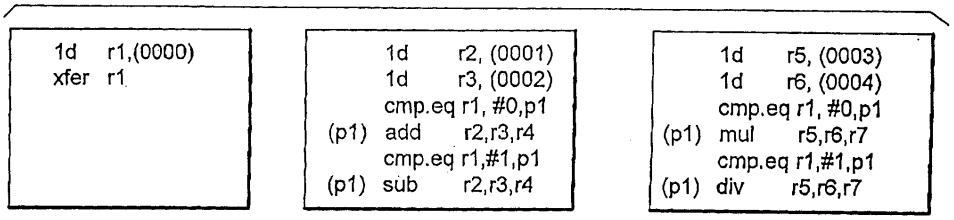
도면9



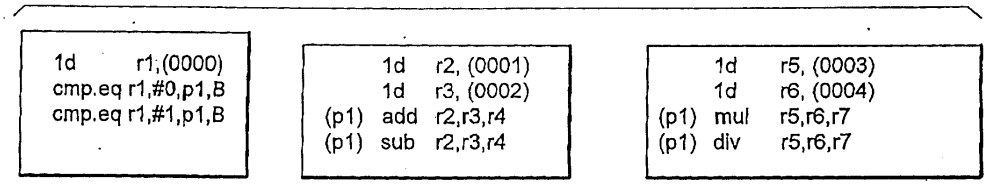
도면10

```
switch (a)
  {
case 0:  d = b + c;
         g = a + f;
         break;
case 1:  d = b - c ;
         g = e / f ;
         break;
}
```

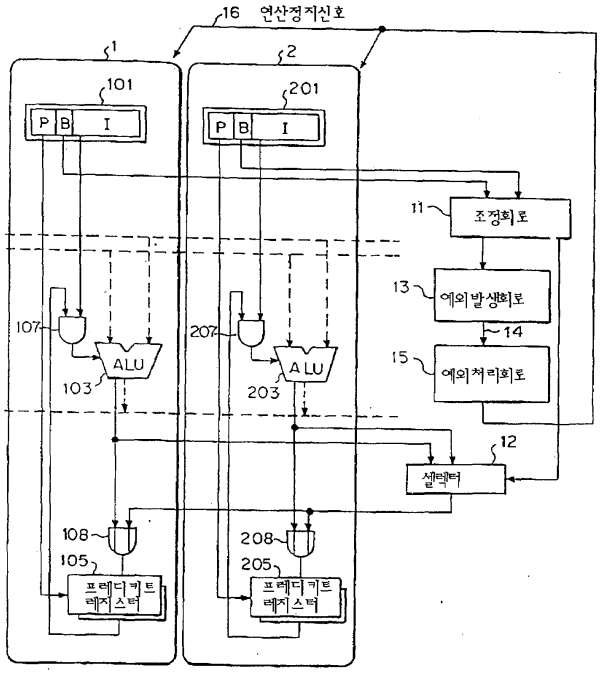
도면11



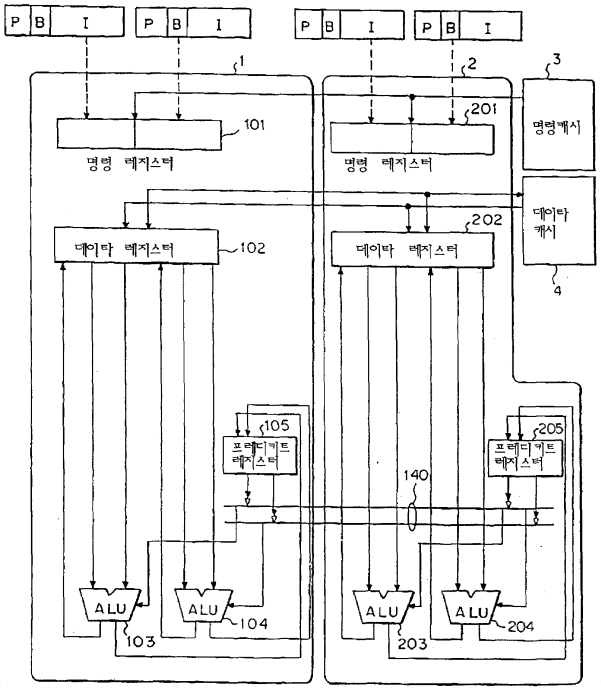
도면12



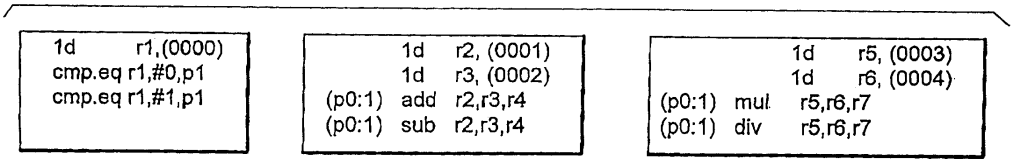
도면13



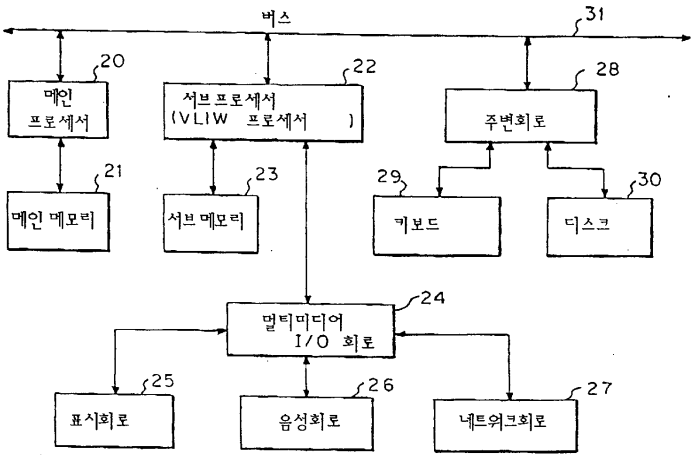
도면14



도면15



도면16


버스
31
20
메인
프로세서
22
서브프로세서
(VLIW 프로세서)
28
주변회로
21
메인 메모리
23
서브메모리
29
키보드
30
디스크
24
멀티미디어
I/O 회로
25
표시회로
26
음성회로
27
네트워크회로