

**(19) 대한민국특허청(KR)**  
**(12) 특허공보(B1)**

(51) Int. Cl. <sup>6</sup> G06F 9/38		(45) 공고일자 1995년11월22일	
		(11) 공고번호 특1995-0014161	
		(24) 등록일자 1995년11월22일	
(21) 출원번호	특1993-0028476	(65) 공개번호	특1995-0020098
(22) 출원일자	1993년12월18일	(43) 공개일자	1995년07월24일
(71) 출원인	재단법인한국전자통신연구소 양승택 대전광역시 유성구 가정동 161번지		
(72) 발명자	김종문 대전광역시 유성구 가정동 236-1 3/3 김명원 대전광역시 유성구 신성동 한울아파트 109-301호 송윤선 대전광역시 대덕구 오정동 신동아파트 8-1203호 이영직 대전광역시 유성구 어은동 한빛아파트 102-105호		
(74) 대리인	김영길, 김명섭		

**심사관 : 오홍수 (책자공보 제4227호)**

**(54) 어레이 프로세서(array processor)의 2단계(stage) 명령어 파이프라인 처리방법**

**요약**

내용 없음.

**대표도**

**도1**

**명세서**

[발명의 명칭]

어레이 프로세서(array processor)의 2단계(stage) 명령어 파이프라인 처리방법

[도면의 간단한 설명]

제 1 도는 2단계 파이프 라인구조를 갖는 간단한 제어회로도.

제 2 도는 제어신호의 사용을 설명하기 위한 회로도.

제 3 도는 종래기술에 따른 2단계 파이프라인 명령어의 처리를 나타낸 도면.

제 4 도는 본 발명에 따른 2단계 파이프라인 명령어의 처리를 나타낸 도면.

\* 도면의 주요 부분에 대한 부호의 설명

- |                                  |                                    |
|----------------------------------|------------------------------------|
| 1 : 프로그램 메모리                     | 2 : instruction pre-fetch register |
| 3 : instruction fetch register   | 4 : read controlsignal decoder     |
| 5 : write control signal decoder | 6 : operand source                 |
| 7 : processing unit              | 8 : destination source             |

[발명의 상세한 설명]

본 발명은 VLSI 기술을 이용한 MIMD(multiple instruction multiple data)방식으로 동작할 수 있는 2클럭 명령어(명령어의 수행에 두 사이클(명령어 페치사이클과 명령어 실행사이클)이 요구되는 명령어)의 구조를 갖는 간단한 어레이 프로세서의 설계에 있어서, 명령어의 2단계(stage)파이프라인 제어회로의 명령어 처리방법에 관한 것이다.

어레이 프로세서는 병렬 시스템의 중앙처리장치로서 사용되는 주 프로세서(main processor)를 말하는데, 병렬 시스템에서는 많은 어레이 프로세서들이 사용된다.

VLSI칩 제작기술로서 현재 널리 이용되고 있는 gate array 또는 cell base 기술을 이용하여 어레이 프로

제서를 제작하면 빠르고 신뢰성 있게 칩을 제작할 수 있으나, 속도와 집적도면에서 떨어진다. 어레이 프로세서의 설계는 일반적인 마이크로 프로세서의 설계 방법을 사용하면 된다.

이러한 방법들중 명령어 축소형 프로세서(RISC:reduced instruction set computer)의 설계방법에 따라서 프로세서를 설계하면 프로세서의 구조를 간략화할 수 있다.

이 방식에 의하면 명령어 갯수가 적으므로 명령어의 동작을 단순화시킬 수 있고, 명령어 파이프라인 처리가 이루어질 수 있게 된다.

이러한 명령어 축소형 방식의 특징은 여기서 개시되는 어레이 프로세서가 갖추어야 하는 특징과 동일하나, 이 방법을 이용하면 어레이 프로세서의 속도를 향상시킬 수 있는 장점이 있다.

본 발명의 목적은 RISC의 특징인 명령어의 파이프라인 처리를 보다 간단히 구현하는 방법을 제안하는 것이다.

여기서 제안하는 설계방법은, 간단한 구조의 어레이 프로세서 여러개를 하나의 칩에 집적시키고, 이러한 어레이 프로세서를 대규모로 집적시켜서 병렬시스템의 속도향상을 얻으려는 어레이 프로세서 설계에 이용한다.

즉 어레이 프로세서는 간단하게 설계하여 가능한도 내에서 동일한 많은 어레이 프로세서를 하나의 칩에 집적시켜야 한다. 현재의 VLSI 기술로는 복잡한 구조의 어레이 프로세서를 하나의 칩에 많이 집적시킬 수는 없다.

그래서, 본 명세서에서는 구성이 단순한 일반적인 순차(sequential) 제어회로에서 추가적인 하드웨어 없이 간단한 방법으로 안정되게 명령어의 파이프라인 동작을 할 수 있는 회로를 설계할 수 있도록 하는 명령어 처리방법을 개시한다.

본 명세서에 개시되는 제어회로를 사용하기 위해서는 명령어 2사이클에 수행이 완료되도록 회로가 설계되어 있어야 하며, 본 발명에 따르면 명령어는 2단계 파이프라인 처리되고 한 사이클에 하나의 명령어가 수행된다.

제 1 도는 2단계 파이프라인 구조를 갖는 명령어의 처리를 위한 제어회로의 간단한 구조를 예시한 것이다. 1번은 프로그램이 담겨져 있는 메모리이며, 2번과 3번은 명령어가 일시적으로 저장되는 레지스터이다. 레지스터 2는 IPFR신호는 명령어를 사전에 읽기 위해서 즉, (pre-fetch)하기 위한 클럭신호이며, 레지스터 3번의 IFR신호는 프리페치된 레지스터(2)에 저장된 명령어의 파이프라인 처리를 위해 제 1 레지스터(2)의내용을 제 2 레지스터(3)에 저장하기 위한 클럭이다.

4번은 프리페치된 명령어(2의 값)를 디코딩하는 블럭이다. 이 디코더(4)는 주로 오퍼랜드가 나오는 레지스터 또는 메모리를 제어하는 신호를 만든다.

5번은 페치된 명령어(3번 레지스터 값)를 디코딩하는 블럭이다. 이 디코더(5)는 데이터가 저장되는 장소(destination) 즉, 레지스터 또는 메모리를 제어하는 신호를 만든다.

제 2 도는 제 1 도를 사용하여 제어하는 방법을 보기위한 블럭도로서, 일반적인 데이터 흐름을 나타내고 있다.

여기서, Operand control signal은 제 1 디코더(4)로부터 제공되는 신호이며, Destination control signal은 제 2 디코더(5)로부터 제공되는 신호이다.

제 2 도에서, 6번은 데이터처리를 위한 오퍼랜드를 저장하고 있는 메모리 또는 레지스터 파일(registerfile)을 나타낸 것이다.

7번은 데이터를 처리하는 부분으로 연산, 논리처리 등을 수행하는 블럭을 표시한다. 8번은 데이터 처리가 끝난 결과를 저장하는 목적지 블럭 즉, 메모리 또는 레지스터 파일이다.

제 3 도는 종래의 파이프라인 명령어의 처리과정을 나타낸 흐름도이다. 하나의 명령어는 두 사이클에 걸쳐 수행된다.

첫번째 사이클에서는 명령어를 프리페치한다. 이 단계를 일반적으로 명령어읽기(instruction fetch) 사이클이라 한다.

두번째 사이클에서는 수행된 결과를 목적지(destination)블럭에 저장하는 단계이다.

일반적으로 명령어 실행(instruction execution)사이클은 명령어에 따른 데이터 처리를 수행하고 그 결과를 메모리 또는 레지스터에 저장하는 단계이다. 이러한 두 사이클을 거치면 하나의 명령어에 대한 처리가 완료된다.

I1-A와 I1-B로 나타낸 것은 하나의 명령어가 두 사이클을 거치면서 수행되는 것을 보인다.

IPFR와 IFR은 제 1 도의 제 1 및 제 2 레지스터(2,3)의 클럭신호가 나오는 타이밍이다.

IPFR에서 명령어가 제 1 레지스터(2)에 프리페치되고, IFR에서 파이프라인 수행을 위한 제 2 레지스터(3)에 명령어가 페치된다.

I2로 표기한 두번째 명령어의 명령어 읽기 단계(I2-A)를 보면 명령어 I1 수행의 두번째 사이클인 명령어 실행사이클(I1-B)에서 명령어 I2의 읽기단계(I2-A)가 이루어진다.

이와같이 종래의 명령어 처리방식에서는 명령어들이 한 클럭씩의 간격을 두고 파이프라인 처리된다.

제 4 도는 본 발명에 따른 명령어 처리의 흐름도이다.

제 3 도와의 차이점은 명령어들의 파이프라인 처리가 반 클럭씩의 간격을 두고 이루어지는 것이다.

이렇게 동작하는 회로구조는 제 3 도에 도시된 명령어 처리방법을 사용하는 제 1 도의 그것과 유사하나, 제 1 및 제 2 디코더(4,5)에서 만들어내는 신호가 다를 수 있다.

IPFR과 IFR은 제 1 및 제 2 레지스터(2,3)의 클럭신호를 받드는 타이밍이다.

제 4 도에 도시된 바와같이 명령어의 처리를 위한 두개의 사이클이 반 사이클 겹쳐져 있으나 제 1 및 제 2 레지스터(2,3)의 신호전달은 원활하다.

그 이유는 클럭의 1레벨에서 0레벨로 가는 에지(negative edge)와 0에서 1로 가는 에지(positive edge)를 모두 사용하기 때문이다.

명령어 I1을 처리하는 첫번째 단계(I1-A)에서는, IPFR에서 명령어 I1이 제 1 레지스터(2)에 쓰여지고 제 1 디코더(4)에서는 operand control signal이 만들어지기 시작한다. IPFR이 나온후 반 사이클이 지단후에는 operand controlsignal이 A1 시간동안 제1디코더(4)에서 안정되게 나온다.

명령어 I1을 처리하는 두번째 단계(I1-B)는 IFR에서 시작된다. 이 시간에는 명령어 I1가 제 2 레지스터(3)에 쓰여지고 제 2 디코더(5)에서는 desination controlsignal이 만들어지기 시작한다.

IFR이 나온후 만들어진 desination controlsignal신호는 B1 시간동안 제 2 디코더(5)에서 안정되게 나온다. 명령어가 제 1 및 제 2 레지스터(2,3)에 저장되는 시간과, 제 1 및 제 2 디코더(4,5)가 명령어를 해석하여 제어 신호를 만드는데 필요한 시간으로서 반 사이클이므로 안정되게 신호를 만들 수 있게 된다 (IPFR이 나오는 시간과 A1이 나오는 시간, IFR이 나오는 시간과 B1이 나오는 시간).

따라서, 본 발명에 따르면 하나의 명령어가 수행되는데 1.5사이클만 필요하게 된다. 파이프라인 동작을 하지 않는 간단한 순차 제어회로에서 제 1 디코더(4)의 operand control signal과 제 2 디코더 desinationcontrol signal을 분리하여 넣어주면 본 발명에 따른 파이프라인 동작을 실현하는 회로를 만들 수 있다. 이회로는 기존의 순차 제어회로에 비교하여 명령어의 파이프라인 처리를 위해서 사용되는 하드웨어의 큰 추가없이 구현가능하다.

본 발명에 따른 방법을 이용하면 다음과 같은 장점이 있다.

1. 제안하는 명령어의 2단계 파이프라인 제어를 위한 회로구성은 일반적인 순차 제어회로에서 사용되는 하드웨어에 추가의 하드웨어 없이 간단하게 구현할 수 있으며,
2. 시스템 주파수가 높아도 제어신호를 안정하고 바르게 만들 수 있다.

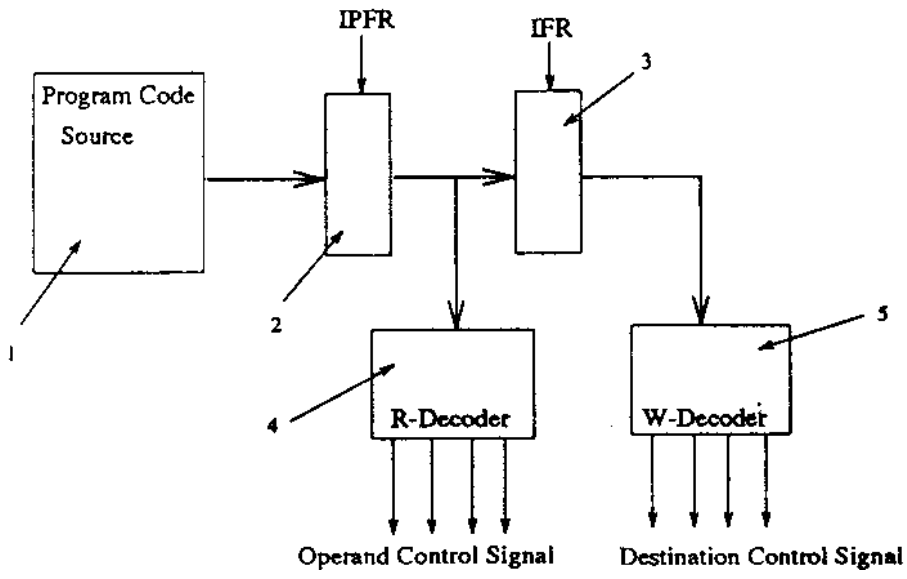
## (57) 청구의 범위

### 청구항 1

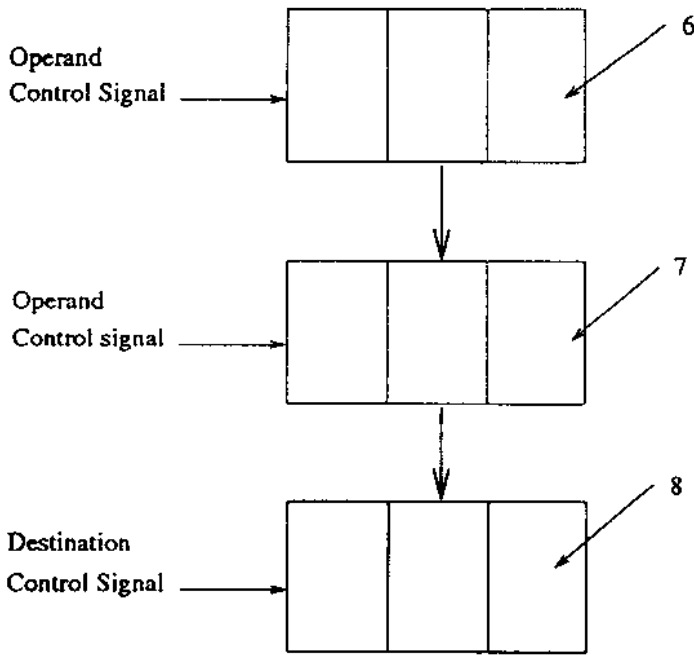
프로그램을 저장하는 메모리(1)와, IPFR신호에 따라서 상기 메모리(1)로부터 명령어를 프리페치하기 위한 제1의 레지스터(2)와, IFR신호에 따라서 명령어의 파이프라인 처리를 위한 제 2 의 레지스터(3)와, 상기 제1 및 상기 제 2 레지스터(2,3) 각각으로부터 제공되는 명령어를 각각 해독하여 오퍼랜드 및 목적지 제어 신호를 출력하는 제 1 및 제 2 디코더(4,5)를 포함하는 어레이 프로세서에서 명령어의 파이프라인 처리를 수행하는 방법에 있어서, 클럭의 네그티브 에지 및 포지티브 에지를 이용하고, 상기 IPFR신호가 제공될때, 상기 제 1 레지스터(2)에 명령어가 쓰여지게 하고, 상기 IPFR신호의 제공시점으로부터 반 사이클이 지난후 상기 제 1 디코더(4)로부터 상기 오퍼랜드 제어신호가 출력되게 하며, 상기 IPFR신호의 제공시점으로부터 반 사이클이 지난후 상기 제 2레지스터(3)로 상기 IFR신호가 제공되게 하고, 상기 IFR신호의 제공시점으로부터 반 사이클이 지난후 상기 제 2 디코더(5)로부터 상기 목적지 제어신호가 출력되게 하는 것을 특징으로 하는 어레이 프로세서의 2단계(stage)명령어 파이프라인 처리방법.

## 도면

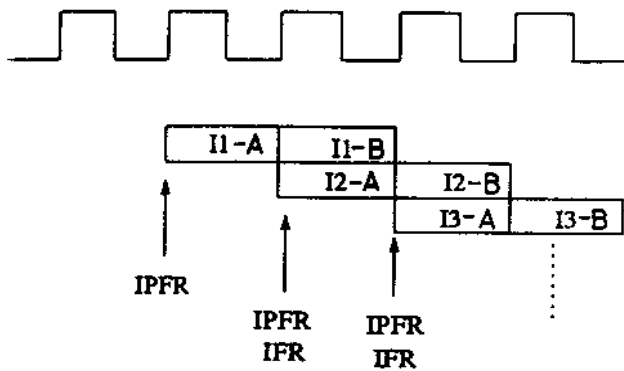
도면1



도면2



도면3



도면4

