



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

(45) 공고일자 2017년02월24일  
 (11) 등록번호 10-1710116  
 (24) 등록일자 2017년02월20일

(51) 국제특허분류(Int. Cl.)  
 G06F 9/38 (2006.01) G06F 1/32 (2006.01)  
 G06F 12/08 (2016.01) G06F 9/30 (2017.01)

(21) 출원번호 10-2010-0082694  
 (22) 출원일자 2010년08월25일  
 심사청구일자 2015년06월02일

(65) 공개번호 10-2012-0019329  
 (43) 공개일자 2012년03월06일

(56) 선행기술조사문헌  
 BINGFENG MEI 외 4명. 'ADRES: An Architecture with Tightly Coupled VLIW Processor and Coarse-Grained Reconfigurable Matrix.' Proceedings of 13th International Conference on FPLA, September 1-3, 2003.\*  
 WO2004061675 A1\*  
 EP1422618 A2  
 US20020087846 A1  
 \*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
 삼성전자주식회사  
 경기도 수원시 영통구 삼성로 129 (매탄동)

(72) 발명자  
 이강용  
 서울특별시 관악구 봉천11동 1632-9번지 401호  
 유동훈  
 서울특별시 강동구 성안로 182, 101동 302호 (길동, 노블레스)

(74) 대리인  
 특허법인 신지

전체 청구항 수 : 총 10 항

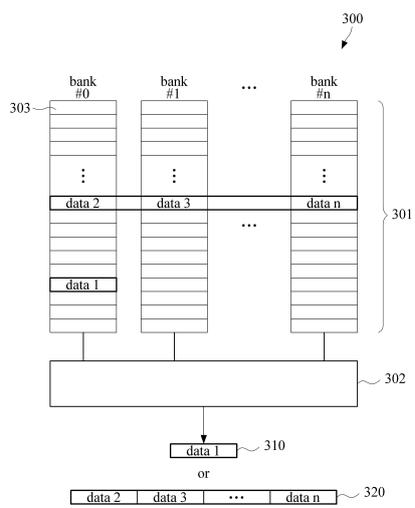
심사관 : 유진태

**(54) 발명의 명칭 프로세서, 메모리 관리 장치 및 방법**

**(57) 요약**

하나의 메모리를 VLIW instruction cache 및 CGA configuration memory로 공용화해서 사용하기 위한 장치 및 방법이 제공된다. 구체적으로, 다수의 저장 라인을 갖는 적어도 1 이상의 저장 공간을 포함하는 저장부로부터 제 1 모드 및 제 2 모드를 갖는 프로세싱 코어에 데이터를 제공할 때, 프로세싱 코어가 상기 제 1 모드인지 또는 상기 제 2 모드인지 여부를 판단하고, 프로세싱 코어가 제 1 모드인 경우, 다수의 저장 공간 중 선택된 하나의 저장 공간의 저장 라인에 대응되는 제 1 데이터를 출력하고, 프로세싱 코어가 제 2 모드인 경우, 다수의 저장 공간 중 선택된 적어도 2개의 저장 공간들의 저장 라인들에 대응되는 데이터들을 결합한 제 2 데이터를 출력한다.

**대표도 - 도3**



## 명세서

### 청구범위

#### 청구항 1

동작 모드로써 제 1 모드 또는 제 2 모드를 갖는 프로세싱 코어부;

다수의 저장 라인을 갖는 적어도 1 이상의 저장 공간을 포함하는 저장부; 및

상기 프로세싱 코어부로부터 태그(tag) 및 인덱스(index)를 포함하는 주소(address)를 전달 받고, 상기 프로세싱 코어부가 상기 제 1 모드인 경우, 상기 다수의 저장 공간 중 하나를 선택하고 선택된 저장 공간의 저장 라인에 대응되는 제 1 데이터를 출력하고, 상기 프로세싱 코어부가 상기 제 2 모드인 경우, 상기 다수의 저장 공간 중 적어도 2개를 선택하고 선택된 저장 공간들의 저장 라인들에 대응되는 데이터들을 결합한 제 2 데이터를 출력하는 출력 인터페이스부; 를 포함하고,

상기 출력 인터페이스부는

상기 프로세싱 코어부가 상기 제 1 모드인지 또는 상기 제 2 모드인지 여부를 판단하고, 상기 프로세싱 코어부에서 수행될 인스트럭션 셋 중에서 모드 전환이 이루어지는 부분을 검출하여 상기 프로세싱 코어부의 모드 전환 여부를 판단하는 모드 판단부;

상기 프로세싱 코어부가 상기 제 1 모드인 경우, 상기 주소에 포함된 태그에 따라 상기 하나의 저장 공간을 선택하고, 상기 주소에 포함된 인덱스에 따라 상기 선택된 저장 공간의 저장 라인을 선택하며, 상기 선택된 저장 라인에 대응되는 제 1 데이터를 출력하는 제 1 출력부; 및

상기 프로세싱 코어부가 상기 제 2 모드인 경우, 구성 정보의 크기에 따라 상기 다수의 저장 공간 중 적어도 2개를 선택하고, 상기 주소에 포함된 인덱스에 따라 상기 선택된 적어도 2개의 저장 공간들의 저장 라인을 선택하고, 상기 선택된 저장 라인에 대응되는 제 2 데이터를 출력하는 제 2 출력부; 를 포함하는 프로세서.

#### 청구항 2

제 1 항에 있어서, 상기 프로세싱 코어부는

재구성 가능 어레이(reconfigurable array)로 형성되고,

상기 제 1 모드에서 VLIW(very long instruction word) 아키텍처로 동작하는 프로세서.

#### 청구항 3

제 2 항에 있어서, 상기 출력 인터페이스부는

상기 제 1 데이터로써 상기 VLIW 아키텍처 상에서 처리될 VLIW 인스트럭션을 출력하는 프로세서.

#### 청구항 4

제 1 항에 있어서, 상기 프로세싱 코어부는

재구성 가능 어레이(reconfigurable array)로 형성되고,

상기 제 2 모드에서 CGA(coarse-grained array) 아키텍처로 동작하는 프로세서.

#### 청구항 5

제 4 항에 있어서, 상기 출력 인터페이스부는

상기 제 2 데이터로써 상기 CGA 아키텍처 상에서 처리될 CGA 인스트럭션 및 상기 CGA 아키텍처의 구조를 정의하기 위한 구성 정보(configuration information)를 출력하는 프로세서.

**청구항 6**

삭제

**청구항 7**

다수의 저장 라인을 갖는 적어도 1 이상의 저장 공간을 포함하는 저장부; 및

프로세싱 코어로부터 태그(tag) 및 인덱스(index)를 포함하는 주소(address)를 전달 받고, 제 1 모드에서 상기 다수의 저장 공간 중 하나를 선택하고 선택된 저장 공간의 저장 라인에 대응되는 제 1 데이터를 출력하고, 제 2 모드에서 구성 정보의 크기에 따라 상기 다수의 저장 공간 중 적어도 2개를 선택하고 선택된 저장 공간들의 저장 라인들에 대응되는 데이터들을 결합한 제 2 데이터를 출력하는 출력 인터페이스부; 를 포함하고,

상기 출력 인터페이스부는,

상기 제 1 데이터 또는 상기 제 2 데이터를 처리할 프로세싱 코어부가 상기 제 1 모드인지 또는 상기 제 2 모드인지 여부를 판단하고, 프로세싱 코어부에서 수행될 인스트럭션 셋 중에서 모드 전환이 이루어지는 부분을 검출하여 상기 프로세싱 코어부의 모드 전환 여부를 판단하는 모드 판단부;

상기 프로세싱 코어부가 상기 제 1 모드인 경우, 상기 주소에 포함된 태그에 따라 상기 하나의 저장 공간을 선택하고, 상기 주소에 포함된 인덱스에 따라 상기 선택된 저장 공간의 저장 라인을 선택하며, 상기 선택된 저장 라인에 대응되는 제 1 데이터를 출력하는 제 1 출력부; 및

상기 프로세싱 코어부가 상기 제 2 모드인 경우, 구성 정보의 크기에 따라 상기 다수의 저장 공간 중 적어도 2개를 선택하고, 상기 주소에 포함된 인덱스에 따라 상기 선택된 적어도 2개의 저장 공간들의 저장 라인을 선택하고, 상기 선택된 저장 라인에 대응되는 제 2 데이터를 출력하는 제 2 출력부; 를 포함하는 메모리 관리 장치.

**청구항 8**

삭제

**청구항 9**

다수의 저장 라인을 갖는 적어도 1 이상의 저장 공간을 포함하는 저장부로부터 제 1 모드 및 제 2 모드를 갖는 프로세싱 코어에 데이터를 제공하기 위한 메모리 관리 방법으로써,

상기 프로세싱 코어로부터 태그(tag) 및 인덱스(index)를 포함하는 주소(address)를 전달 받는 단계;

상기 프로세싱 코어에서 수행될 인스트럭션 셋 중에서 모드 전환이 이루어지는 부분을 검출하여 상기 프로세싱 코어가 상기 제 1 모드인지 또는 상기 제 2 모드인지 여부를 판단하는 단계;

상기 프로세싱 코어가 상기 제 1 모드인 경우, 상기 주소에 포함된 태그에 따라 상기 다수의 저장 공간 중 하나를 선택하고, 상기 주소에 포함된 인덱스에 따라 상기 선택된 저장 공간의 저장 라인을 선택하며, 상기 선택된 저장 공간의 저장 라인에 대응되는 제 1 데이터를 출력하는 단계; 및

상기 프로세싱 코어가 상기 제 2 모드인 경우, 구성 정보의 크기에 따라 상기 다수의 저장 공간 중 적어도 2개를 선택하고, 상기 주소에 포함된 인덱스에 따라 상기 적어도 2개의 저장 공간들의 저장 라인을 선택하고, 상기 선택된 저장 라인들에 대응되는 데이터들을 결합한 제 2 데이터를 출력하는 단계; 를 포함하는 메모리 관리 방법.

**청구항 10**

제 9 항에 있어서,  
 상기 제 1 모드는 상기 프로세싱 코어의 VLIW 모드이고,  
 상기 제 2 모드는 상기 프로세싱 코어의 CGA 모드인 메모리 관리 방법.

**청구항 11**

제 10 항에 있어서, 상기 제 1 데이터를 출력하는 단계는  
 상기 VLIW 모드에서 처리될 VLIW 인스트럭션을 출력하는 메모리 관리 방법.

**청구항 12**

제 10 항에 있어서, 상기 제 2 데이터를 출력하는 단계는  
 상기 CGA 모드에서 처리될 CGA 인스트럭션 및 상기 CGA의 구성 정보를 출력하는 메모리 관리 방법.

**발명의 설명**

**기술 분야**

[0001] 재구성 가능 어레이(reconfigurable array)의 메모리 활용 기술과 관련된다.

**배경 기술**

[0002] 재구성 가능 아키텍처(reconfigurable architecture)란 어떠한 작업을 수행하기 위한 컴퓨팅 장치의 하드웨어적 구성을 각각의 작업에 최적화되도록 변경할 수 있는 아키텍처를 의미한다.

[0003] 어떠한 작업을 하드웨어적으로만 처리하면 고정된 하드웨어의 기능으로 인해 작업 내용에 약간의 변경이 가해지면 이를 효율적으로 처리하기가 어렵다. 또한, 어떠한 작업을 소프트웨어적으로만 처리하면 그 작업 내용에 맞도록 소프트웨어를 변경하여 처리하는 것이 가능하지만 하드웨어적 처리에 비해 속도가 늦다.

[0004] 재구성 가능 아키텍처는 이러한 하드웨어/소프트웨어의 장점을 모두 만족시킬 수가 있다. 특히, 동일한 작업이 반복적으로 수행되는 디지털 신호 처리 분야에서는 이러한 재구성 가능 아키텍처가 많은 주목을 받고 있다. 또한 재구성 가능 아키텍처는 수행 중인 작업 별로 그 구조가 최적화되는 것이 가능하기 때문에, 최근에는 VLIW 모드에서 일반적인 인스트럭션을 실행하고 CGA 모드에서 루프 연산을 실행하는 혼합형 VLIW/CGA 프로세서가 등장하게 되었다.

[0005] 혼합형 VLIW/CGA 프로세서는 서로 다른 두 가지 종류의 메모리를 사용한다. 첫 번째는 VLIW 모드에서 인스트럭션을 저장하는 캐시 메모리(cache memory)이다. 두 번째는 CGA 모드에서 CGA 구성 정보를 저장하는 구성 메모리(configuration memory)이다. 그런데, 각각의 모드는 서로 배타적이기 때문에 실행 도중에 어느 하나의 메모리는 사용되지 않게 된다. 즉 VLIW 모드에서는 구성 메모리가 필요 없고, CGA 모드에서는 캐시 메모리가 필요 없게 되는데 이것은 메모리 집적 효율 및 에너지 활용 측면에서 좋지 못하다.

**발명의 내용**

**해결하려는 과제**

[0006] 캐시 메모리 및 구성 메모리로 사용할 수 있는 재구성 가능 어레이의 공용 메모리 및 이러한 공용 메모리를 이용한 메모리 관리 기술이 제공된다.

**과제의 해결 수단**

[0007] 본 발명의 일 양상에 따른 프로세서는 동작 모드로써 제 1 모드 또는 제 2 모드를 갖는 프로세싱 코어부, 다수의 저장 라인을 갖는 적어도 1 이상의 저장 공간을 포함하는 저장부, 및 프로세싱 코어부가 상기 제 1 모드인 경우, 다수의 저장 공간 중 하나를 선택하고 선택된 저장 공간의 저장 라인에 대응되는 제 1 데이터를

출력하고, 프로세싱 코어부가 제 2 모드인 경우, 다수의 저장 공간 중 적어도 2개를 선택하고 선택된 저장 공간들의 저장 라인들에 대응되는 데이터들을 결합한 제 2 데이터를 출력하는 출력 인터페이스부를 포함할 수 있다.

[0008] 본 발명의 일 양상에 따른 메모리 관리 장치는 다수의 저장 라인을 갖는 적어도 1 이상의 저장 공간을 포함하는 저장부, 및 제 1 모드에서 다수의 저장 공간 중 하나를 선택하고 선택된 저장 공간의 저장 라인에 대응되는 제 1 데이터를 출력하고, 제 2 모드에서 다수의 저장 공간 중 적어도 2개를 선택하고 선택된 저장 공간들의 저장 라인들에 대응되는 데이터들을 결합한 제 2 데이터를 출력하는 출력 인터페이스부를 포함할 수 있다.

[0009] 본 발명의 일 양상에 따른 메모리 관리 방법은 다수의 저장 라인을 갖는 적어도 1 이상의 저장 공간을 포함하는 저장부로부터 제 1 모드 및 제 2 모드를 갖는 프로세싱 코어에 데이터를 제공하기 위한 메모리 관리 방법으로써, 프로세싱 코어가 제 1 모드인지 또는 제 2 모드인지 여부를 판단하는 단계, 프로세싱 코어가 제 1 모드인 경우, 다수의 저장 공간 중 하나를 선택하고, 선택된 저장 공간의 저장 라인에 대응되는 제 1 데이터를 출력하는 단계, 및 프로세싱 코어가 제 2 모드인 경우, 다수의 저장 공간 중 적어도 2개를 선택하고, 선택된 저장 공간들의 저장 라인들에 대응되는 데이터들을 결합한 제 2 데이터를 출력하는 단계를 포함할 수 있다.

**발명의 효과**

[0010] 개시된 내용에 의하면, 하나의 메모리 장치를 인스트럭션 캐시 및 구성 메모리로 공용화해서 사용할 수 있기 때문에 메모리 칩의 집적 공간 및 에너지 소비를 줄일 수 있다.

**도면의 간단한 설명**

- [0011] 도 1은 본 발명의 일 실시예에 따른 컴퓨팅 장치를 도시한다.
- 도 2는 본 발명의 일 실시예에 따른 프로세싱 코어를 도시한다.
- 도 3은 본 발명의 일 실시예에 따른 메모리 관리 장치를 도시한다.
- 도 4는 본 발명의 일 실시예에 따른 출력 인터페이스부를 도시한다.
- 도 5a 및 도 5b는 본 발명의 일 실시예에 따른 내부 메모리의 동작을 도시한다.
- 도 6은 본 발명의 일 실시예에 따른 메모리 관리 방법을 도시한다.

**발명을 실시하기 위한 구체적인 내용**

- [0012] 이하, 첨부된 도면을 참조하여 본 발명의 실시를 위한 구체적인 예를 상세히 설명한다.
- [0013] 도 1은 본 발명의 일 실시예에 따른 컴퓨팅 장치(100)를 도시한다.
- [0014] 도 1을 참조하면, 컴퓨팅 장치(100)는 프로세서(101)와 외부 메모리(102)를 포함하고, 프로세서(101)는 프로세싱 코어(103)와 내부 메모리(104)를 포함한다.
- [0015] 프로세서(101)는 여러 가지의 데이터를 처리한다. 처리될 데이터는 외부 메모리(102)로부터 내부 메모리(103)로 패치(fetch)된다. 프로세서(101) 내부의 프로세싱 코어(103)가 어떤 태스크를 수행할 때 외부 메모리(102)에 접근하는 것보다 내부 메모리(104)에 접근하는 것이 더 빠르기 때문에 처리될 데이터를 내부 메모리(103)로 가져다 놓고 처리하는 것이 처리 속도 면에서 더 유리하다.
- [0016] 프로세싱 코어(103)는 동적 재구성 가능 어레이(dynamic reconfigurable array)에 기초하여 형성된다. 동적 재구성 가능 어레이란 시스템의 구조가 그 시스템의 사용 목적에 따라 동적으로 바뀔 수 있는 프로세서를 말한다. 따라서 프로세싱 코어(103)의 하드웨어 아키텍처는 처리 목적에 따라 변할 수 있다. 예를 들어, 프로세싱 코어(103)는 서로 배타적인 제 1 모드와 제 2 모드를 가질 수 있다. 제 1 모드는 일반적인 연산 수행에 적합한 VLIW(very long instruction word) 모드가 될 수 있다. 제 2 모드는 루프 연산 수행에 적합한 CGA(coarse-grained mode) 모드가 될 수 있다. 예컨대, 프로세싱 코어(103)는 제 1 모드에서 일반적인 연산을 처리하다가 루프 연산을 만나게 되면 제 2 모드로 전환해서 루프 연산을 처리하고 루프 연산이 끝나면 다시 제 1 모드로 전환되는 것이 가능하다.
- [0017] 각각의 모드에서 프로세싱 코어(103)의 구조는 각 모드에서 수행되는 연산에 최적화된다. 예를 들어, 제 2 모드에서 프로세싱 코어(103)의 구조는 루프 연산을 처리하기 위해 그 루프 연산에 최적화되는 것이 가능하다. 각각의 모드에서 처리되는 데이터, 인스트럭션 및 프로세싱 코어(103)의 구조를 정의하는 구성 정보

(configuration information)는 내부 메모리(104)에 저장될 수 있다.

- [0018] 내부 메모리(104)는 프로세싱 코어(103)의 각 모드마다 다른 종류의 데이터를 출력한다. 내부 메모리(104)는 프로세싱 코어(103)의 제 1 모드에 응답하여 제 1 데이터를 출력하고, 프로세싱 코어(103)의 제 2 모드에 응답하여 제 1 데이터와 상이한 제 2 데이터를 출력할 수 있다. 예컨대, 제 1 데이터는 VLIW 모드에서의 일반적인 인스트럭션이 될 수 있고, 제 2 데이터는 CGA 모드에서의 루프 인스트럭션 및 CGA 구조를 정의하는 구성 정보(configuration information)가 될 수 있다.
- [0019] 다시 말해, 내부 메모리(104)는, 프로세싱 코어(103)가 제 1 모드인 경우, 인스트럭션 캐시(instruction cache)로 동작하고, 프로세싱 코어(103)가 제 2 모드인 경우, 구성 메모리(configuration memory)로 동작하는 것이 가능하다.
- [0020] 도 2는 본 발명의 일 실시예에 따른 프로세싱 코어를 도시한다. 이것은 도 1의 프로세싱 코어(103)에 관한 일 예가 될 수 있다.
- [0021] 도 1 및 도 2를 참조하면, 프로세싱 코어(200)는 다수의 프로세싱 엘리먼트(201)와 중앙 데이터 레지스터 파일(202)을 포함한다.
- [0022] 각각의 프로세싱 엘리먼트(PE#0~PE#15)는 계산 유닛(function unit), 또는 계산 유닛 및 레지스터 파일(register file)로 구성될 수 있다. 그리고 각 프로세싱 엘리먼트(PE#0~PE#15)는 어떤 작업을 독립적으로 처리하는 것이 가능하다.
- [0023] 다수의 프로세싱 엘리먼트(201) 중에서 첫 번째 행의 프로세싱 엘리먼트들(PE#0~PE#3)은 제 1 모드에서 VLIW 프로세서로 사용될 수 있다. 즉 VLIW 모드에서 첫 번째 행의 프로세싱 엘리먼트들(PE#0~PE#3)은 일반적인 인스트럭션을 수행하는 것이 가능하다. 또는 레지스터 파일을 공유하는 별도의 프로세싱 엘리먼트가 VLIW 프로세서로 사용될 수 있다.
- [0024] 다수의 프로세싱 엘리먼트(201) 중에서 전체의 프로세싱 엘리먼트들(PE#0~PE#15)은 제 2 모드에서 CGA 프로세서로 사용될 수 있다. 즉 CGA 모드에서 전체의 프로세싱 엘리먼트들(PE#0~PE#15)은 루프 연산에 최적화되어 루프와 관련된 인스트럭션을 수행하는 것이 가능하다.
- [0025] 중앙 데이터 레지스터 파일(202)은 VLIW 모드에서 CGA 모드로의 전환, 또는 그 역으로의 전환 시에 데이터를 임시로 저장한다.
- [0026] 본 실시예에 따라, 제 1 모드 또는 VLIW 모드에서 필요한 데이터 및 인스트럭션들을 제 1 데이터라고 하고, 제 2 모드 또는 CGA 모드에서 필요한 데이터 및 인스트럭션들을 제 2 데이터라고 한다. 즉, 제 1 데이터는 VLIW 모드에서의 VLIW instruction이 될 수 있고, 제 2 데이터는 CGA 모드에서 프로세싱 엘리먼트들(201)의 연결 상태와 어떤 데이터를 어떤 프로세싱 엘리먼트가 처리할지를 정의한 configuration information이 될 수 있다.
- [0027] 도 3은 본 발명의 일 실시예에 따른 메모리 관리 장치를 도시한다. 이것은 도 1의 내부 메모리(104)에 관한 일 예가 될 수 있다.
- [0028] 도 1 및 도 3을 참조하면, 메모리 관리 장치(300)는 저장부(301)와 출력 인터페이스부(302)를 포함한다.
- [0029] 저장부(301)는 다수의 저장 공간(bank#0~bank#n)으로 구성되며, 각각의 저장 공간(bank#0~bank#n)은 다수의 저장 라인(303)으로 구분된다.
- [0030] 출력 인터페이스부(302)는 프로세싱 코어(103)의 모드에 따라 서로 다른 종류의 데이터를 프로세싱 코어(103)로 제공한다. 일 예로써, 프로세싱 코어(103)가 VLIW 모드인 경우, 출력 인터페이스부(302)는 어느 하나의 저장 공간(bank#0)을 선택하고 선택된 저장 공간(bank#0)의 저장 라인에 대응되는 데이터(data1)를 출력한다. 다른 예로써, 프로세싱 코어(103)가 CGA 모드인 경우, 출력 인터페이스부(302)는 모든 저장 공간(bank#0~bank#n)을 선택하고, 선택된 저장 공간(bank#0~bank#n)의 각 저장 라인에 대응되는 데이터(data2, data3, ..., data n)를 합쳐서 출력한다. 따라서, 본 실시예에 따라, VLIW 모드에서 출력된 제 1 데이터(310)는 VLIW instruction이 될 수 있고, CGA 모드에서 출력된 제 2 데이터(320)는 CGA configuration information이 될 수 있다.
- [0031] 출력 인터페이스부(302)가 저장부(301)의 어떤 저장 라인을 선택할지는 프로세싱 코어(103)로부터 전달 받은 주소에 따라 결정될 수 있다. 또한, 선택적으로, 제 1 모드에서 출력되는 데이터는 전달 받은 주소에 포함된 오프셋에 따라 선택된 저장 공간(bank#0)의 저장 라인에 대응되는 데이터(data1)중 일부가 출력될 수도 있다. 나아가, 제 2 모드에서 모든 저장 공간(bank#0~bank#n)의 저장 라인이 선택될 수도 있고 configuration

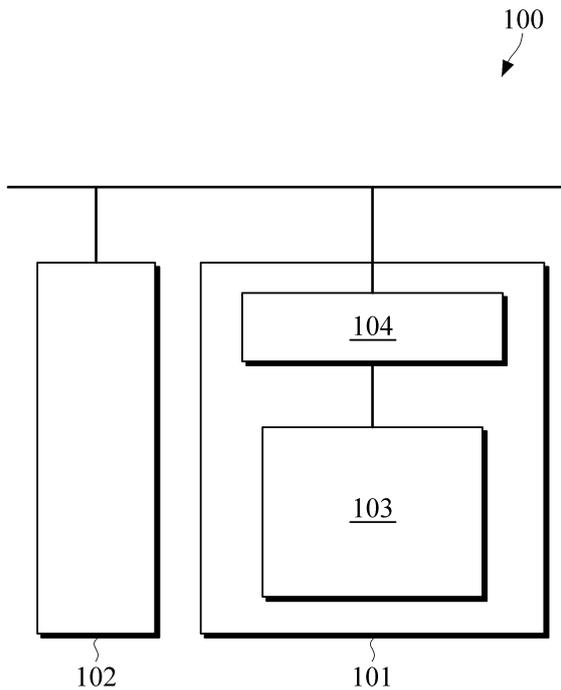
information의 크기에 따라 일부의 저장 공간(bank#0~bank#1)의 저장 라인이 선택될 수도 있다.

- [0032] 도 4는 본 발명의 일 실시예에 따른 출력 인터페이스부를 도시한다.
- [0033] 도 1, 도 3 및 도 4를 참조하면, 출력 인터페이스부(400)는 제 1 출력부(401), 제 2 출력부(402), 및 모드 판단부(403)를 포함할 수 있다.
- [0034] 제 1 출력부(401)는 다수의 저장 공간(bank#0~bank#n) 중에서 하나를 선택한다. 선택되는 저장 공간(예컨대, bank#0)은 프로세싱 코어(103)로부터 전달 받은 주소에 포함된 tag에 따라 결정될 수 있다. 제 1 출력부(401)는 선택된 저장 공간(bank#0)에서 특정한 저장 라인을 선택한다. 선택되는 저장 라인은 프로세싱 코어(103)로부터 전달 받은 주소에 포함된 index에 따라 결정될 수 있다. 제 1 출력부(401)는 선택된 저장 라인에 존재하는 데이터 중 일부 또는 전부(310)를 출력해서 프로세싱 코어(103)로 제공한다.
- [0035] 제 2 출력부(402)는 다수의 저장 공간(bank#0~bank#n) 중에서 연속적으로 2개 이상을 선택한다. 일 예로써, 제 2 출력부(402)는 모든 저장 공간(301)을 선택하는 것이 가능하다. 제 2 출력부(402)는 선택된 저장 공간(301)에서 특정한 저장 라인을 선택한다. 선택되는 저장 라인은 프로세싱 코어(103)로부터 전달 받은 주소에 포함된 index에 따라 결정될 수 있다. 제 2 출력부(402)는 선택된 저장 라인에 존재하는 데이터를 결합한 데이터(302)를 출력해서 프로세싱 코어(103)로 제공한다.
- [0036] 모드 판단부(403)는 프로세싱 코어(103)의 모드 전환을 판단한다. 예를 들어, 모드 판단부(403)는 프로세싱 코어(103)가 VLIW 모드인지 또는 CGA 모드인지 여부를 판단한다. 그리고 모드 판단부(403)는 판단 결과에 따라 제 1 출력부(401) 또는 제 2 출력부(402) 중 하나를 활성화시킨다.
- [0037] 도 5a 및 도 5b는 본 발명의 일 실시예에 따른 내부 메모리의 동작을 도시한다.
- [0038] 도 5a를 참조하면, 내부 메모리(500)는 VLIW 모드에서 VLIW instruction을 제공하는 "n-way set associative cache"로 동작할 수 있다. 예컨대, 도 5a에서, 주소가 수신되면, 주소의 index는 각각 tag set과 data set으로 전달되고 주소의 tag는 태그 비교기(501)로 전달된다. 태그 비교기(501)는 주소에 포함된 tag와 index로부터 식별된 tag를 비교해서 일치하는지 여부를 확인한다. 만약 tag가 일치하면 일치하는 tag를 데이터 선택부(502)로 전달한다. 데이터 선택부(502)는 일치하는 tag에 대응되는 데이터를 data set에서 선택하고 선택된 data를 출력한다. 또한 선택적으로, 데이터 선택부(502)는 선택된 data에서 offset을 고려해서 그 일부를 출력할 수도 있다. 데이터 선택부(502)로부터 출력되는 data에는 VLIW 모드에서 사용될 데이터 및 인스트럭션 등이 담길 수 있다.
- [0039] 도 5b를 참조하면, 내부 메모리(500)는 CGA 모드에서 CGA configuration information을 제공하는 "directed-mapped cached configuration memory"로 동작할 수 있다. 예컨대, 도 5b에서, 주소가 수신되면, 주소의 index는 각각 tag set과 data set으로 전달되고 주소의 tag는 태그 비교기(501)로 전달된다. CGA 모드에서의 configuration information은 VLIW 모드에서의 VLIW instruction 보다 그 크기가 크므로 도 5b에서는 도 5a와 같이 n-way를 구분하지 않고 tag set을 하나만 사용하고 여러 개의 data set을 하나의 data set로 취급하는 것이 가능하다. 태그 비교기(501)는 주소에 포함된 tag와 index로부터 식별된 tag를 비교해서 일치하는지 여부를 확인한다. 만약 tag가 일치하면 일치하는 tag를 데이터 결합부(503)로 전달한다. 데이터 결합부(503)는 일치하는 tag에 대응되는 데이터를 data set에서 선택하고 선택된 data를 출력한다. 즉, 도 5b에서는 data set의 하나의 라인이 configuration line이 될 수 있으며, 데이터 결합부(503)로부터 출력되는 data에는 CGA 모드에서 사용될 데이터 및 인스트럭션, 하드웨어 아키텍처의 구성 정보 등이 담길 수 있다.
- [0040] 도 6은 본 발명의 일 실시예에 따른 메모리 관리 방법을 도시한다.
- [0041] 도1, 도 3, 도 4 및 도 6을 참조하여 본 실시예에 따른 메모리 관리 방법을 설명하면 다음과 같다.
- [0042] 먼저 프로세싱 코어(103)가 VLIW 모드인지 또는 CGA 모드인지 여부를 판단한다(601). 예컨대, 모드 전환부(403)가 프로세싱 코어(103)에서 수행될 인스트럭션 셋 중에서 모드 전환이 이루어지는 부분을 검출해서 프로세싱 코어(103)의 모드 전환 여부를 판단하는 것이 가능하다.
- [0043] 프로세싱 코어(103)가 VLIW 모드인 경우, 제 1 출력부(401)를 활성화한다(602). 그리고 제 1 출력부(401)를 통해 VLIW instruction을 출력한다(603). 예를 들어, 제 1 출력부(401)가 다수의 저장 공간(bank#0~bank#n) 중에서 하나를 선택하고, 선택된 저장 공간(예컨대, bank#0)의 특정한 저장 라인에 포함된 데이터의 전부 또는 일부를 출력하는 것이 가능하다.

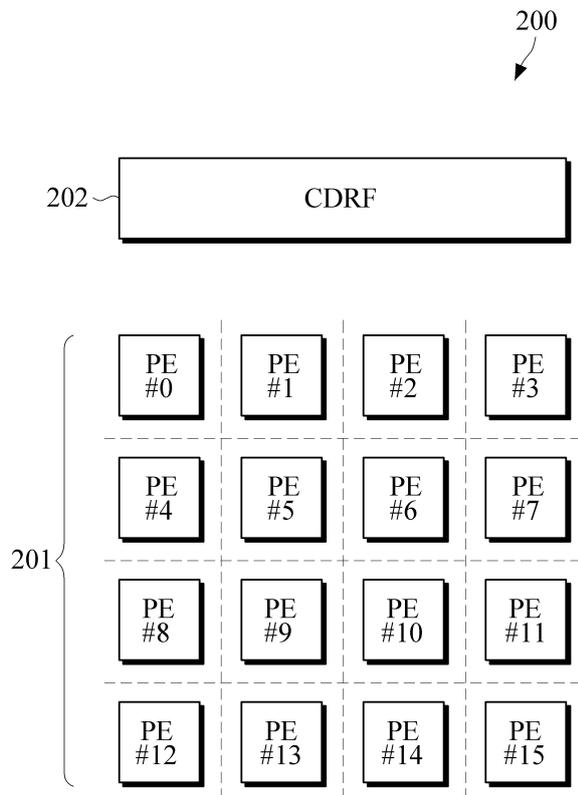
- [0044] 프로세싱 코어(103)가 CGA 모드인 경우, 제 2 출력부(402)를 활성화한다(604). 그리고 제 2 출력부(402)를 통해 CGA configuration을 출력한다(605). 예를 들어, 제 2 출력부(402)가 모든 저장 공간(301)을 선택하고 선택된 저장 공간(301)의 각 저장 라인의 데이터를 결합해서 출력하는 것이 가능하다.
- [0045] 이상에서 살펴본 것과 같이, 개시된 장치 및 방법에 의하면, VLIW/CGA mixed processor에서 processor의 각 상태에 따라 processor 내부의 하나의 메모리 장치를 "n-way set associative cache" 또는 "directed-mapped cached configuration memory"로 선택적으로 사용할 수 있게 된다.
- [0046] 한편, 본 발명의 실시 예들은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터가 읽을 수 있는 코드로 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다.
- [0047] 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현하는 것을 포함한다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고 본 발명을 구현하기 위한 기능적인(functional) 프로그램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술 분야의 프로그래머들에 의하여 용이하게 추론될 수 있다.
- [0048] 이상에서 본 발명의 실시를 위한 구체적인 예를 살펴보았다. 전술한 실시 예들은 본 발명을 예시적으로 설명하기 위한 것으로 본 발명의 권리범위가 특정 실시 예에 한정되지 아니할 것이다.

**도면**

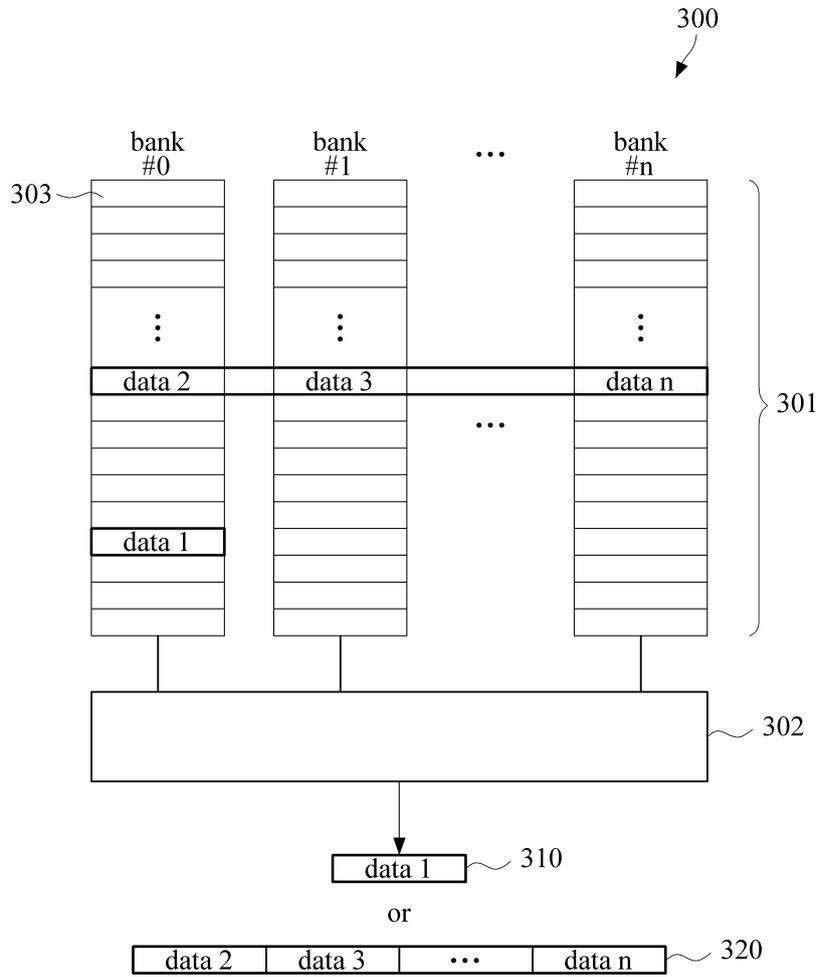
**도면1**



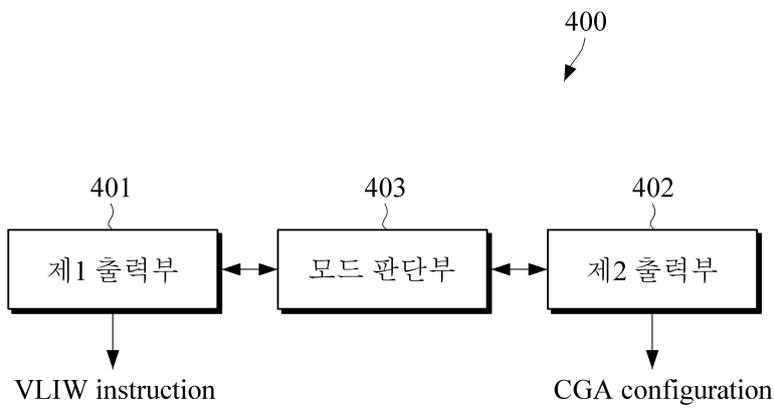
도면2



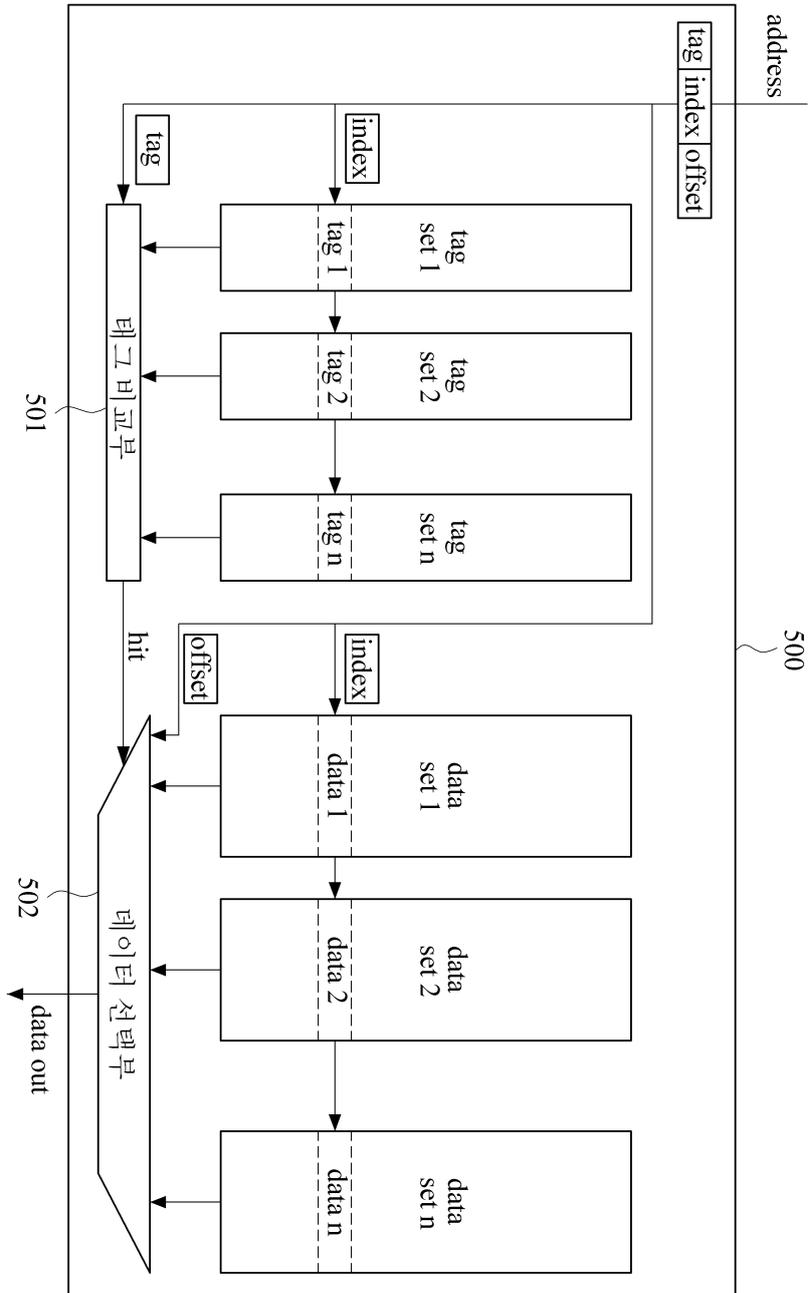
도면3



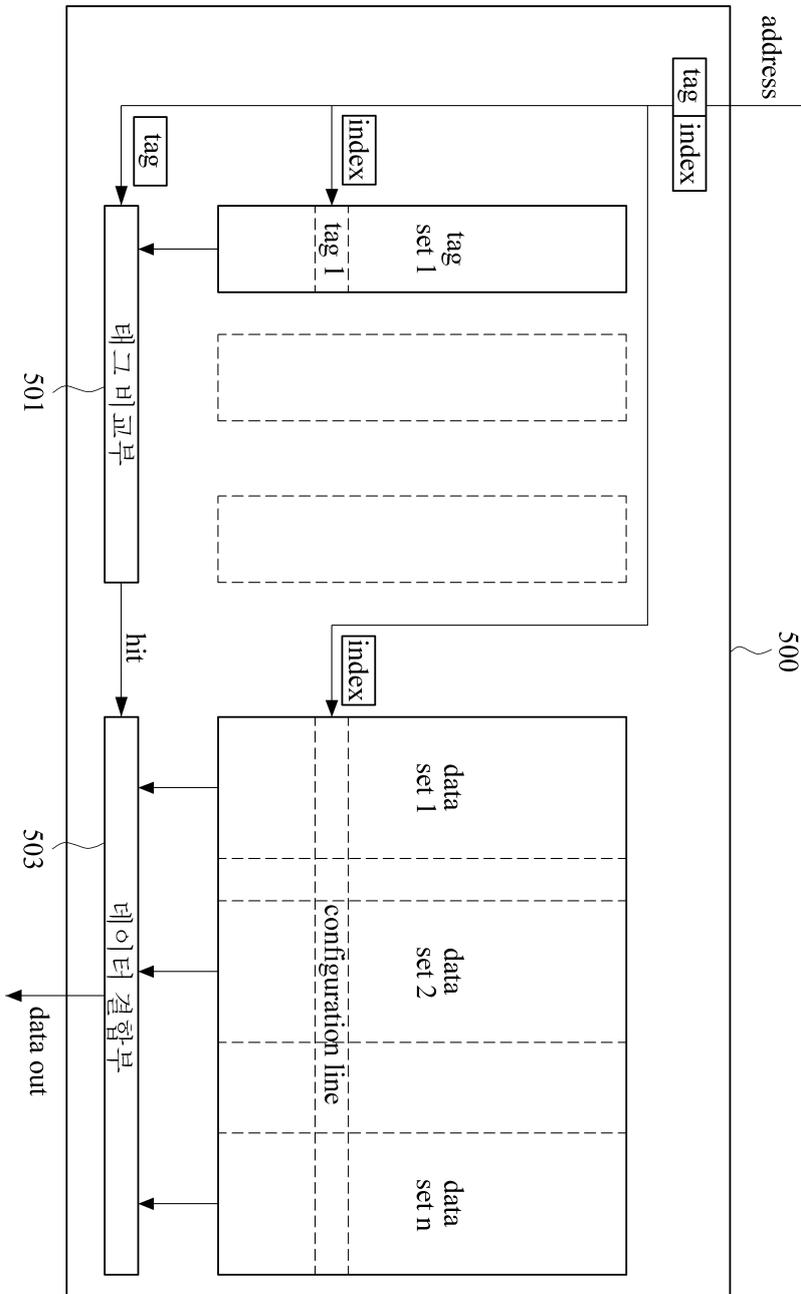
도면4



도면5a



도면5b



도면6

