



(19)대한민국특허청(KR)  
(12) 공개특허공보(A)

(51) Int. Cl.

G06F 9/28 (2006.01)

G06F 9/38 (2006.01)

G06F 9/30 (2006.01)

G06F 13/14 (2006.01)

G06F 9/38 (2006.01)

G06F 9/30 (2006.01)

(11) 공개번호 10-2007-0022239

(43) 공개일자 2007년02월26일

(21) 출원번호 10-2006-7020245

(22) 출원일자 2006년09월28일

심사청구일자 없음

번역문 제출일자 2006년09월28일

(86) 국제출원번호 PCT/GB2005/001069

(87) 국제공개번호 WO 2005/096141

국제출원일자 2005년03월22일

국제공개일자 2005년10월13일

(30) 우선권주장 10/813,615 2004년03월31일 미국(US)

(71) 출원인 이세라 인코포레이티드  
영국 알몬즈베리 비에스32 4에이큐 아즈텍 웨스트 더 퀴드런트 2520

(72) 발명자 노우리스 시몬  
영국 베쓰 비에이1 5알유 칼리지 로드 라간 하우스 3

(74) 대리인 박장원

전체 청구항 수 : 총 21 항

(54) 비대칭 듀얼 경로 프로세싱용 장치 및 방법

(57) 요약

본 발명의 실시예들에 따르면, 컴퓨터 프로세서 아키텍처, 특히 컴퓨터 프로세서, 컴퓨터 프로세서 동작 방법, 컴퓨터를 위한 명령 세트를 이용하는 컴퓨터 프로그램 물이 개시된다. 본 발명에 따른 일 실시예에서, 컴퓨터 프로세서가 제공되는데, 상기 컴퓨터 프로세서는 명령 패킷들의 시퀀스를 보유하는 메모리로부터 폐치된 명령 패킷들을 디코딩하는 디코드 유닛과; 그리고 제 1 및 제 2 프로세싱 채널들을 포함하며, 각 채널은 복수의 기능 유닛들을 포함하며, 여기서 상기 제 1 프로세싱 채널은 제어 연산들을 수행할 수 있음과 아울러 비교적 좁은 비트 폭을 갖는 제어 레지스터 파일을 포함하며, 상기 제 2 프로세싱 채널은 적어도 하나의 입력이 벡터인 데이터 프로세싱 연산들을 수행할 수 있음과 아울러 비교적 넓은 비트 폭을 갖는 데이터 레지스터 파일을 포함한다. 상기 디코드 유닛은 각 명령 패킷에 대해 상기 명령 패킷이 (i) 상기 제 1 프로세싱 채널상에서 순차적으로 실행될 복수의 제어 명령들을 정의하는지 또는 (ii) 상기 제 2 실행 채널상에서 동시에 실행될 적어도 하나의 데이터 프로세싱 명령을 포함하는 복수의 명령들을 정의하는지를 검출함과 아울러 상기 검출에 의존하여 상기 제 1 및 제 2 채널들을 제어하도록 동작가능하다.

대표도

도 1

## 특허청구의 범위

### 청구항 1.

컴퓨터 프로세서에 있어서,

명령 패킷들 시퀀스를 보유하는 메모리로부터 폐치된 명령 패킷들을 디코딩하는 디코드 유닛과; 그리고

제 1 및 제 2 프로세싱 채널들을 포함하며, 각 채널은 복수의 기능 유닛들을 포함하며, 여기서 상기 제 1 프로세싱 채널은 제어 연산들을 수행할 수 있으며, 비교적 좁은 비트 폭을 갖는 제어 레지스터 파일을 포함하며, 그리고 상기 제 2 프로세싱 채널은 적어도 하나의 입력이 벡터인 데이터 프로세싱 연산들을 수행할 수 있으며, 비교적 넓은 비트 폭을 갖는 데이터 레지스터 파일을 포함하며;

여기서, 상기 디코드 유닛은 각 명령 패킷에 대해 상기 명령 패킷이 (i) 상기 제 1 프로세싱 채널상에서 순차적으로 실행되는 복수의 제어 명령들을 정의하는지 또는 (ii) 상기 제 2 실행 채널상에서 동시에 실행되는 적어도 하나의 데이터 프로세싱 명령을 포함하는 복수의 명령들을 정의하는지를 검출하며, 그리고 상기 검출에 의존하여 상기 제 1 및 제 2 채널들을 제어하도록 동작가능한 것을 특징으로 하는 컴퓨터 프로세서.

### 청구항 2.

제 1항에 있어서, 상기 제 1 프로세싱 채널은 브랜치 유닛 및 제어 실행 유닛을 더 포함하는 것을 특징으로 하는 컴퓨터 프로세서.

### 청구항 3.

제 1항 또는 제 2항에 있어서, 상기 제 2 프로세싱 채널은 고정 데이터 실행 유닛 및 구성가능 데이터 실행 유닛을 더 포함하는 것을 특징으로 하는 컴퓨터 프로세서.

### 청구항 4.

제 3항에 있어서, 상기 고정 데이터 실행 유닛 및 구성가능 데이터 실행 유닛 모두는 단일 명령 복수 데이터 포맷에 따라 동작하는 것을 특징으로 하는 컴퓨터 프로세서.

### 청구항 5.

선행하는 청구항들 중 어느 한 항에 있어서, 상기 제 1 및 제 2 프로세싱 채널들은 로드 저장 유닛을 공유하는 것을 특징으로 하는 컴퓨터 프로세서.

### 청구항 6.

제 5항에 있어서, 상기 로드 저장 유닛은 상기 제 1 프로세싱 채널에 의해 제공된 제어 정보 및 상기 제 2 프로세싱 채널에 의해 제공된 데이터를 사용하는 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 7.

선행하는 청구항들 중 어느 한 항에 있어서, 상기 명령 패킷들은 모두 동일한 비트 길이인 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 8.

제 7항에 있어서, 상기 명령 패킷들은 모두 64 비트 길이인 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 9.

선행하는 청구항들 중 어느 한 항에 있어서, 상기 제어 명령들은 모두 18 내지 24 비트의 비트 길이인 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 10.

제 9항에 있어서, 상기 제어 명령들은 모두 21 비트 길이인 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 11.

제 7항에 있어서, 명령 패킷의 각 명령의 특성은 적어도 제어 명령, 데이터 명령, 그리고 메모리 액세스 명령으로부터 선택되는 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 12.

제 11항에 있어서, 각 데이터 명령의 비트 길이는 34 비트인 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 13.

제 11항에 있어서, 각 메모리 액세스 명령의 비트 길이는 28 비트인 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 14.

선행하는 청구항들 중 어느 한 항에 있어서, 상기 디코드 유닛이 상기 명령 패킷이 3개의 제어 명령들을 정의하는 것으로 검출하는 때에, 상기 디코드 유닛은 상기 제 1 프로세싱 채널에 상기 3개의 제어 명령들을 제공하며, 이에 의해 상기 3개의 제어 명령들이 순차적으로 실행되도록 동작가능한 것을 특징으로 하는 컴퓨터 프로세서.

#### 청구항 15.

선행하는 청구항들 중 어느 한 항에 있어서, 상기 디코드 유닛이 상기 명령 패킷이 적어도 하나의 데이터 명령을 포함하는 2개의 명령들을 정의하는 것으로 검출하는 때에, 상기 디코드 유닛은 상기 제 2 프로세싱 채널에 적어도 상기 데이터 명령을 제공하며, 이에 의해 상기 2개의 명령들이 동시에 실행되도록 동작가능한 것을 특징으로 하는 컴퓨터 프로세서.

**청구항 16.**

선행하는 청구항들 중 어느 한 항에 있어서,

상기 디코드 유닛은 상기 시퀀스의 각 명령 패킷의 소정의 비트 위치들에서 지정된 비트들 세트의 값들을 판독하여,

- a) 상기 명령 패킷이 복수의 제어 명령들을 정의하는지 또는 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는지를 결정하며; 그리고
- b) 상기 명령 패킷이 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는 경우, 상기 2개의 명령들 각각의 특성은 제어 명령; 데이터 명령; 그리고 메모리 액세스 명령으로부터 선택되도록 동작가능한 것을 특징으로 하는 컴퓨터 프로세서.

**청구항 17.**

제 3항 내지 제 16항 중 어느 한 항에 있어서, 상기 구성가능 데이터 실행 유닛은 결과를 지정 레지스터 파일에 복귀시키기 이전에 단일 발행된 명령에 의해 제공되는 데이터 상에서 2개 이상의 연속적인 연산들을 실행할 수 있는 것을 특징으로 하는 컴퓨터 프로세서.

**청구항 18.**

각각이 복수의 기능 유닛들을 포함하는 제 1 및 제 2 프로세싱 채널들- 여기서, 상기 제 1 프로세싱 채널은 비교적 좁은 비트 폭을 갖는 제어 레지스터 파일을 포함하며, 상기 제 2 프로세싱은 비교적 넓은 비트 폭을 갖는 데이터 레지스터 파일을 포함하며 -을 포함하는 컴퓨터 프로세서 동작 방법에 있어서,

명령 패킷을 디코딩하여 상기 명령 패킷이 동일한 길이의 복수의 제어 명령들을 정의하는지 또는 적어도 하나가 벡터인 적어도 하나의 데이터 명령을 포함하는 2개의 명령들을 정의하는지를 검출하는 단계와;

상기 명령 패킷이 동일한 길이의 복수의 제어 명령들을 정의하는 때에, 상기 제어 명령들을 상기 제 1 프로세싱 채널에 제공하며, 이에 의해 상기 제어 명령들이 순차적으로 실행되는 단계와; 그리고

상기 명령 패킷이 적어도 하나의 데이터 명령을 포함하는 복수의 명령들을 정의하는 때에, 적어도 상기 데이터 명령을 상기 제 2 프로세싱 채널에 제공하며, 이에 의해 상기 복수의 명령들이 동시에 실행되는 단계를 포함하는 것을 특징으로 하는 컴퓨터 프로세서 동작 방법.

**청구항 19.**

명령 패킷들 시퀀스를 포함하는 프로그램 코드 수단을 포함하는 컴퓨터 프로그램 물(product)에 있어서,

상기 명령 패킷들은 동일한 길이의 복수의 제어 명령들을 포함하는 제 1 타입의 명령 패킷과 적어도 하나의 데이터 명령을 포함하는 복수의 명령들을 포함하는 제 2 타입의 명령 패킷을 포함하며,

여기서, 상기 컴퓨터 프로그램 물은 상기 제 1 타입의 명령 패킷이 전용 제어 프로세싱 채널에 의해 실행되며, 그리고 상기 제 2 명령 패킷의 상기 적어도 하나의 데이터 명령이 전용 데이터 프로세싱 채널에 의해 실행되며, 상기 전용 제어 프로세싱 채널이 상기 전용 데이터 프로세싱 채널보다 비교적 좁은 비트 폭을 갖도록 컴퓨터상에서 실행되는 것을 특징으로 하는 컴퓨터 프로그램 물.

**청구항 20.**

각각이 복수의 기능 유닛들을 포함하는 제 1 및 제 2 프로세싱 채널들- 여기서, 상기 제 1 프로세싱 채널은 비교적 좁은 비트 폭을 갖는 제어 레지스터 파일을 포함하며, 상기 제 2 프로세싱은 비교적 넓은 비트 폭을 갖는 데이터 레지스터 파일을 포함하며 -을 포함하는 컴퓨터 프로세서 동작 방법에 있어서,

프로그램 메모리로부터 명령 패킷들 시퀀스를 폐치하는 단계와, 상기 명령 패킷들 모두는 소정의 비트 위치들에서 지정된 비트들 세트를 포함하며;

각 명령 패킷을 디코딩하는 단계를 포함하며, 상기 디코딩 단계는 상기 지정된 비트들의 값들을 판독하여,

a) 상기 명령 패킷이 복수의 제어 명령들을 정의하는지 또는 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는지를 결정하며; 그리고

b) 상기 명령 패킷이 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는 경우에, 상기 2개의 명령들 각각의 특성은 적어도 제어 명령; 데이터 명령; 그리고 메모리 액세스 명령으로부터 선택되는 단계를 포함하는 것을 특징으로 하는 컴퓨터 프로세서 동작 방법.

## 청구항 21.

명령 패킷들 시퀀스를 포함하는 프로그램 코드 수단을 포함하는 컴퓨터 프로그램 물에 있어서,

상기 명령 패킷들은 대체적으로 동일한 길이의 복수의 제어 명령들을 포함하는 제 1 타입의 명령 패킷과 적어도 하나의 데이터 명령을 포함하는 제 1 및 제 2 명령들을 포함하는 제 2 타입의 명령 패킷을 포함하며,

상기 명령 패킷들은 상기 명령 패킷 내의 지정된 비트 위치에서 적어도 하나의 표시기 비트를 포함하며,

여기서, 상기 컴퓨터 프로그램 물은 상기 표시 비트가 상기 컴퓨터의 디코드 유닛과 협력하여,

a) 상기 명령 패킷이 복수의 제어 명령들을 정의하는지 또는 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는지를 지정하며; 그리고

b) 적어도 하나의 데이터 명령을 포함하는 복수의 명령들인 경우에, 상기 2개의 명령들 각각의 특성은 제어 명령; 데이터 명령; 그리고 메모리 액세스 명령으로부터 선택되도록 컴퓨터상에서 실행되는 것을 특징으로 하는 컴퓨터 프로그램 물.

## 명세서

### 기술분야

본 발명은 컴퓨터 프로세서, 컴퓨터 프로세서 동작 방법, 및 컴퓨터를 위한 명령 세트를 포함하는 컴퓨터 프로그램 물에 관한 것이다.

### 배경기술

컴퓨터 프로세서들의 속도를 증가시키기 위하여, 종래기술 아키텍처들은 명령들의 실행을 위해 듀얼 실행 경로들을 사용하여 왔다. 듀얼 실행 경로 프로세서들은 프로세서 속도를 증가시키기 위해 연산들의 병렬성(parallelism)을 사용하여 단일 명령 복수 데이터(SIMD) 원리에 따라 동작할 수 있다.

그러나, 듀얼 실행 경로들의 사용 및 SIMD 프로세싱에 불구하고, 프로세서 속도를 증가시킬 지속적인 필요가 있다. 전형적인 듀얼 실행 경로 프로세서들은 2개의 대체적으로 동일한 채널들을 사용하며, 이에 따라 각 채널은 제어 코드 및 데이터 경로 코드 모두를 핸들링한다. 공지된 프로세서들이 32비트 표준 엔코딩 및 16비트 "고밀도(dense)" 엔코딩의 조합을 지원하지만은, 이러한 방식들은 16비트 포맷에서 이용가능한 수(few) 비트들의 시맨틱 콘텐츠(semantic contents)의 결여를 포함하는 여러 단점들을 갖게 된다.

더욱이, 종래 범용 디지털 신호 처리기들은 컨볼루션(convolution), 고속 푸리에 변환들, Trellis/Viterbi 엔코딩, 상관관계, 유한 임펄스 응답 필터링 및 다른 연산들과 같은 특수화된 연산들의 수행을 포함하는 많은 목적들에 대해 어플리케이션 특정 알고리즘들과 부합되지 않을 수 있다.

### 발명의 상세한 설명

본 발명에 따른 일 실시예에서, 컴퓨터 프로세서가 제공된다. 상기 컴퓨터 프로세서는 명령 패킷들의 시퀀스를 보유하는 메모리로부터 폐치된 명령 패킷들을 디코딩하는 디코드 유닛과; 그리고 제 1 및 제 2 프로세싱 채널들을 포함하며, 각 채널은 복수의 기능 유닛들을 포함하며, 여기서 상기 제 1 프로세싱 채널은 제어 연산들을 수행할 수 있음과 아울러 비교적 좁은 비트 폭을 갖는 제어 레지스터 파일을 포함하며, 상기 제 2 프로세싱 채널은 적어도 하나의 입력이 벡터인 데이터 프로세싱 연산들을 수행할 수 있음과 아울러 비교적 넓은 비트 폭을 갖는 데이터 레지스터 파일을 포함한다. 상기 디코드 유닛은 각 명령 패킷에 대해 상기 명령 패킷이 (i) 상기 제 1 프로세싱 채널상에서 순차적으로 실행될 복수의 제어 명령들을 정의하는지 또는 (ii) 상기 제 2 실행 채널상에서 동시에 실행될 적어도 하나의 데이터 프로세싱 명령을 포함하는 복수의 명령들을 정의하는지를 검출함과 아울러 상기 검출에 의존하여 상기 제 1 및 제 2 채널들을 제어하도록 동작가능하다.

추가적인 관련 실시예들에서, 상기 제 1 프로세싱 채널은 브랜치 유닛 및 제어 실행 유닛을 더 포함할 수 있다. 상기 제 2 프로세싱 채널은 고정 데이터 실행 유닛 및 구성가능 데이터 실행 유닛을 더 포함할 수 있다. 상기 고정 데이터 실행 유닛 및 상기 구성가능 데이터 실행 유닛 모두는 단일 명령 복수 데이터 포맷에 따라 동작할 수 있다. 상기 제 1 및 제 2 프로세싱 채널들은 로드 저장 유닛을 공유할 수 있다. 상기 로드 저장 유닛은 상기 제 1 프로세싱 채널에 의해 제공된 제어 정보 및 상기 제 2 프로세싱 채널에 의해 제공된 데이터를 사용할 수 있다. 상기 명령 패킷들 모두는 64 비트 길이와 같은 동일한 비트 길이가 될 수 있다. 상기 제어 명령들 모두는 21 비트 길이와 같은 18 내지 24 비트의 비트 길이가 될 수 있다. 명령 패킷에서의 각 명령의 특성은 적어도 제어 명령, 데이터 명령, 그리고 메모리 액세스 명령으로부터 선택될 수 있다. 각 데이터 명령의 비트 길이는 예를 들어, 34 비트가 될 수 있으며, 각 메모리 액세스 명령의 비트 길이는 예를 들어, 28 비트가 될 수 있다.

추가적인 관련 실시예들에서, 상기 디코드 유닛이 상기 명령 패킷이 3개의 제어 명령들을 정의하는 것으로 검출하는 때에, 상기 디코드 유닛은 상기 제 1 프로세싱 채널에 상기 3개의 제어 명령들을 제공하며, 이에 의해 상기 3개의 제어 명령들이 순차적으로 실행되도록 동작가능할 수 있다. 또한, 상기 디코드 유닛이 상기 명령 패킷이 적어도 하나의 데이터 명령을 포함하는 2개의 명령들을 정의하는 것으로 검출하는 때에, 상기 디코드 유닛은 상기 제 2 프로세싱 채널에 적어도 상기 데이터 명령을 제공하며, 이에 의해 상기 2개의 명령들이 동시에 실행되도록 동작가능할 수 있다. 상기 디코드 유닛은 상기 시퀀스의 각 명령 패킷의 소정의 비트 위치들에서 지정된 비트들의 세트값들을 판독하도록 동작가능할 수 있으며, 이에 따라 a)상기 명령 패킷이 복수의 제어 명령들을 정의하는지 또는 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는지를 결정하며, b)상기 명령 패킷이 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는 경우에, 상기 2개의 명령들 각각의 특성은 제어 명령; 데이터 명령; 그리고 메모리 액세스 명령으로부터 선택된다. 상기 구성가능 데이터 실행 유닛은 결과를 지정 레지스터 파일에 복귀시키기 이전에, 단일 발행된 명령에 의해 제공된 데이터 상에서 2개 이상의 연속적인 연산들을 실행할 수 있다.

본 발명에 따른 다른 실시예에서, 제 1 및 제 2 프로세싱 채널들- 각각이 복수의 기능 유닛들을 포함하며 -을 포함하는 컴퓨터 프로세서 동작 방법이 제공되는데, 상기 제 1 프로세싱 채널은 비교적 좁은 비트 폭을 갖는 제어 레지스터 파일을 포함하며, 상기 제 2 프로세싱 채널은 비교적 넓은 비트 폭을 갖는 데이터 레지스터 파일을 포함한다. 상기 방법은 명령 패킷을 디코딩하여 상기 명령 패킷이 동일한 길이의 복수의 제어 명령들을 정의하는지 또는 적어도 하나가 벡터인 적어도 하나의 데이터 명령을 포함하는 2개의 명령들을 정의하는지를 검출하는 단계와; 상기 명령 패킷이 동일한 길이의 복수의 제어 명령들을 정의하는 때에, 상기 제어 명령들을 상기 제 1 프로세싱 채널에 제공하며, 이에 의해 상기 제어 명령들이 순차적으로 실행되는 단계와; 그리고 상기 명령 패킷이 적어도 하나의 데이터 명령을 포함하는 복수의 명령들을 정의하는 때에, 적어도 상기 데이터 명령을 상기 제 2 프로세싱 채널에 제공하며, 이에 의해 상기 복수의 명령들이 동시에 실행되는 단계를 포함한다.

본 발명에 따른 다른 실시예에서, 명령 패킷들 시퀀스를 포함하는 프로그램 코드 수단을 포함하는 컴퓨터 프로그램 물(product)이 제공되는데, 상기 명령 패킷들은 동일한 길이의 복수의 제어 명령들을 포함하는 제 1 타입의 명령 패킷과 적어도 하나의 데이터 명령을 포함하는 복수의 명령들을 포함하는 제 2 타입의 명령 패킷을 포함하며, 여기서 상기 컴퓨터



프로그램 물은 상기 제 1 타입의 명령 패킷이 전용 제어 프로세싱 채널에 의해 실행되며, 상기 제 2 명령 패킷의 상기 적어도 하나의 데이터 명령이 전용 데이터 프로세싱 채널에 의해 실행되도록 컴퓨터상에서 수행되며, 상기 전용 제어 프로세싱 채널은 상기 전용 데이터 프로세싱 채널보다 비교적 좁은 비트 폭을 갖는다.

본 발명에 따른 다른 실시예에서, 제 1 및 제 2 프로세싱 채널들- 여기서, 각각이 복수의 기능 유닛들을 포함하며 -을 포함하는 컴퓨터 프로세서 동작 방법이 제공되는데, 여기서 상기 제 1 프로세싱 채널은 비교적 좁은 비트 폭을 갖는 제어 레지스터 파일을 포함하며, 상기 제 2 프로세싱 채널은 비교적 넓은 비트 폭을 갖는 데이터 레지스터 파일을 포함한다. 상기 방법은 프로그램 메모리로부터 명령 패킷들 시퀀스를 페치하는 단계와, 상기 명령 패킷들 모두는 소정의 비트 위치들에서 지정된 비트들 세트를 포함하며; 각 명령 패킷을 디코딩하는 단계와, 상기 디코딩 단계는 상기 지정된 비트들 값들을 판독하여 a)상기 명령 패킷이 복수의 제어 명령들을 정의하는지 또는 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는지를 결정하며; 그리고 b)상기 명령 패킷이 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는 경우, 상기 2개의 명령들 각각의 특성은 적어도 제어 명령, 데이터 명령, 그리고 메모리 액세스 명령으로부터 선택되는 단계를 포함한다.

본 발명에 따른 다른 실시예에서, 명령 패킷들 시퀀스를 포함하는 프로그램 코드 수단을 포함하는 컴퓨터 프로그램 물(product)이 제공되는데, 상기 명령 패킷들은 대체적으로 동일한 길이의 복수의 제어 명령들을 포함하는 제 1 타입의 명령 패킷과 적어도 하나의 데이터 명령을 포함하는 제 1 및 제 2 명령들을 포함하는 제 2 타입의 명령 패킷을 포함하며, 상기 명령 패킷들은 상기 명령 패킷 내의 지정된 비트 위치에서 적어도 하나의 표시기 비트를 포함하며, 여기서 상기 컴퓨터 프로그램 물은 상기 표시 비트가 상기 컴퓨터의 디코드 유닛과 협력하여 a)상기 명령 패킷이 복수의 제어 명령들을 정의하는지 또는 적어도 하나가 데이터 명령인 복수의 명령들을 정의하는지를 지정하며, 그리고 b)적어도 하나의 데이터 명령을 포함하는 복수의 명령들이 있는 경우에, 상기 2개의 명령들 각각의 특성은 제어 명령; 데이터 명령; 그리고 메모리 액세스 명령으로부터 선택되도록 컴퓨터상에서 실행된다.

본 발명의 추가적인 이점들 및 신규한 특징들은 부분적으로 하기의 상세한 설명에서 제시될 것이며, 부분적으로 하기 내용 및 첨부 도면들의 검토하에서 기술분야의 당업자에게 자명하게 될 것이며, 또는 본 발명의 실시예에 의해 알게 될 수 있다.

본 발명을 더 잘 이해하기 위해, 그리고 본 발명이 어떻게 수행되는지를 보여주기 위해, 단지 예로써 첨부 도면들에 대한 참조가 있을 것이다.

## 실시예

도 1은 본 발명의 일 실시예에 따른 비대칭 듀얼 경로 컴퓨터 프로세서의 블록도이다. 도 1의 프로세서는 단일 명령 스트림(100)의 프로세싱을 2개의 서로다른 하드웨어 실행 경로들(프로세싱 제어 코드에 전용인 제어 실행 경로(102) 및 프로세싱 데이터 코드에 전용인 데이터 실행 경로(103))로 나누게 한다. 2개의 실행 경로들(102, 103)의 데이터 폭들, 연산자들 및 다른 특성들은 제어 코드와 데이터경로 코드의 서로다른 특성들에 따라 다르다. 전형적으로, 제어 코드는 보다 적은, 보다 좁은 레지스터들에 유리하기에 병행화하기 어려우므로 전형적으로(배타적이지 않음) C 코드 또는 다른 하이-레벨 언어로 기록되며, 그 코드 밀도는 일반적으로 속도 성능보다 중요하다. 대조적으로, 데이터경로 코드는 전형적으로 큰 파일의 폭넓은 레지스터들에 유리하기에 쉽게 병행화되므로 어셈블리 언어로 기록되며, 그 성능은 코드 밀도보다 더 중요하다. 도 1의 프로세서에서, 2개의 서로다른 실행 경로들(102 및 103)은 2개의 서로다른 타입들의 코드를 핸들링하는데 전용인데, 각 측은 레지스터들의 폭과 개수에 의해 차별화되는, 제어 레지스터 파일(104) 및 데이터 레지스터 파일(105)과 같은 자신의 아키텍처 레지스터 파일을 갖는다. 제어 레지스터들은 비트들 개수(일 예에서, 32비트)에 의해 보다 좁은 폭으로 되며, 데이터 레지스터들은 보다 넓은 폭(일 예에서, 64비트)로 된다. 따라서, 2개의 실행경로들 각각이 서로다른, 특수화된 기능들을 수행하는 사실로 인해 서로다른 비트폭들을 갖는다는 점에서, 프로세서는 비대칭이다.

도 1의 프로세서에서, 명령 스트림(100)은 일련의 명령 패킷들로 구성된다. 제공된 각 명령 패킷은 명령 디코드 유닛(101)에 의해 디코딩되는데, 상기 디코드 유닛은 하기에서 추가적으로 설명될 바와같이 제어 명령들과 데이터 명령들을 분리시킨다. 제어 실행 경로(102)는 본 실시예에서 데이터 실행 경로(107)와 공유되는 로드 저장 유닛(108), 실행 유닛(107), 및 브랜치 유닛(106)을 사용하여 명령 스트림에 대한 제어-흐름 연산들을 핸들링하며, 머신의 상태 레지스터들을 관리한다. 프로세서의 제어측만이 C, C++, 또는 자바 언어에 대한 컴파일러 또는 다른 하이-레벨 언어 컴파일러에 보여질 필요가 있다. 제어측 내에서, 브랜치 유닛(106)과 실행 유닛(107)의 동작은 기술분야의 당업자들에게 공지된 종래 프로세서 설계에 따른다.

데이터 실행 경로(103)는 고정 실행 유닛(109)과 구성가능 딥 실행 유닛(110) 모두에서 SIMD(단일 명령 복수 데이터) 병렬성을 이용한다. 하기에서 추가적으로 설명될 바와같이, 구성가능 딥 실행 유닛(110)은 종래 SIMD 프로세서들에 의해 사용되는 폭 차원에 부가하여 작업당 명령을 증가시키기 위해 프로세싱의 깊이 차원을 제공한다.

만일 디코딩된 명령이 제어 명령을 정의하는 경우엔, 이는 머신의 제어 실행 경로상의 적절한 기능 유닛(예를 들어, 브랜치 유닛(106), 실행 유닛(107), 및 로드/저장 유닛(108))에 인가된다. 만일 디코딩된 명령이 고정 또는 구성가능 데이터 프로세싱 연산을 갖는 명령을 정의하는 경우에, 이는 데이터 프로세싱 실행 경로에 제공된다. 명령 패킷의 데이터 명령 부분내에서 지정된 비트들은 명령이 고정 또는 구성가능 데이터 프로세싱 명령인지를 표시하며, 구성가능 명령의 경우에, 추가적으로 지정되는 비트들이 구성 정보를 정의한다. 디코딩된 데이터 프로세싱 명령의 서브-타입에 의존하여, 데이터가 머신의 데이터 프로세싱 경로의 고정 또는 구성가능 실행 서브-경로들에 제공된다.

본원에서, "구성가능"은 복수의 소정의("유사-정적인(pseudo-static)") 연산자 구성들로부터 연산자 구성을 선택하는 능력을 나타낸다. 연산자의 유사-정적인 구성은 연산자가 (i) 일정한 타입의 연산을 수행하게 하며, 또는 (ii) 일정한 방식으로 관련 요소들에 상호연결되게 하며; 또는 (iii) 상기 (i) 또는 (ii)의 조합을 가능하게 하는데 효과적이다. 실제상, 선택된 유사-정적인 구성이 한 번에 많은 연산자 요소들의 특성 및 상호연결성을 결정할 수 있다. 이는 또한 데이터 경로와 관련된 스위칭 구성들을 제어할 수 있다. 바람직한 실시예에서, 상기 복수의 유사-정적인 연산자 구성들의 적어도 일부는 하기에 추가적으로 설명될 바와같이 데이터 프로세싱 명령의 연산-코드 부분에 의해 선택가능하다. 또한 본원에서의 일 실시예에 따라, "구성가능 명령"은 다중비트 값들 레벨에서, 예를 들어 4개 이상의 비트 다중비트 값들 레벨에서 또는 워드들 레벨에서 적합화된 연산들의 수행을 허용한다.

머신의 각 서로다른 측들상에서 수행되는 제어 및 데이터 프로세싱 명령들 모두는 메모리 액세스(로드/저장) 및 기본 산술 연산들을 정의할 수 있음이 지적된다. 제어 연산들에 대한 입력들/피연산자들이 제어 레지스터 파일(104)로/파일(104)로부터 제공될 수 있는 반면에, 데이터 프로세싱 연산들에 대한 데이터/피연산자들이 레지스터 파일(105)로/파일(105)로부터 제공된다.

본 발명의 일 실시예에 따라, 각 데이터 프로세싱 연산의 적어도 하나의 입력은 벡터가 될 수 있다. 이러한 관점에서, 구성가능 연산자들 및/또는 구성가능 데이터 경로의 스위칭 회로는 수행되는 연산 특성 및/또는 이들간의 상호연결성에 의해 벡터 연산자들을 수행하도록 구성가능한 것으로 간주될 수 있다. 예를 들어, 데이터 프로세싱 연산에 대한 64비트 벡터 입력이 4개의 16비트 스칼라 피연산자들을 포함할 수 있다. 본원에서, "벡터"는 스칼라 피연산자들의 조합이다. 벡터 산술은 복수의 스칼라 피연산자들 상에서 수행될 수 있으며, 스칼라 요소들의 조정(steering), 이동 및 변경을 포함할 수 있다. 벡터 연산의 모든 피연산자들이 벡터일 필요는 없다. 예를 들어, 벡터 연산은 입력들로서 스칼라와 적어도 하나의 벡터 모두를 가질 수 있으며, 스칼라 또는 벡터인 결과를 출력할 수 있다.

본원에서, "제어 명령들"은 프로그램 흐름, 및 브랜치 및 어드레스 생성에 전용인 명령들을 포함하지만, 데이터 프로세싱 명령이 아니다. "데이터 프로세싱 명령들"은 적어도 하나의 입력이 벡터인 논리 연산들 또는 산술 연산들을 위한 명령들을 포함한다. 데이터 프로세싱 명령들은 복수의 데이터 명령들상에서, 예를 들어 SIMD 프로세싱에서, 또는 데이터 요소들의 보다 넓은, 보다 짧은 벡터들의 프로세싱에서 동작할 수 있다. 언급된 제어 명령들 및 데이터 명령들의 필수 기능들은 중첩되지 않는다. 그러나, 2개 타입의 코드가 논리 및 스칼라 산술 성능들을 갖는다는 공통점이 있다.

도 2는 도 1의 프로세서를 위한 3개의 타입의 명령 패킷을 도시한다. 각 타입의 명령 패킷은 64비트 길이이다. 명령 패킷(211)은 고밀도 제어 코드를 위한 3-스칼라 타입이며, 3개의 21비트 제어 명령들(c21)을 포함한다. 명령 패킷들(212 및 213)은 데이터경로 코드의 병행 실행을 위한 LIW(긴 명령 워드)타입이다. 본 예에서, 각 명령 패킷(212, 213)은 2개의 명령들을 포함하지만, 원하는 경우에 서로다른 수들이 포함될 수 있다. 명령 패킷(212)은 34비트 데이터 명령(d34) 및 28비트 메모리 명령(m28)을 포함하며, 데이터-측 로드-저장 동작(m28 명령)으로 데이터-측 산술(d34 명령)의 병행 실행에 사용된다. 메모리-클래스 명령들(m28)은 제어측으로부터의 어드레스들을 사용하여 프로세서의 제어 측 또는 데이터 측으로부터 관독되거나 이들에 기입될 수 있다. 명령 패킷(213)은 34비트 데이터 명령(d34)과 21비트 제어 명령(c21)을 포함하며, 제어측 산술, 브랜치 또는 로드-저장 동작과 같은 제어-측 동작(c21 명령)으로 데이터-측 산술(d34 명령)의 병행 실행에 사용된다.

도 1의 실시예의 명령 디코드 유닛(101)은 무슨 타입의 패킷이 디코딩되는지를 결정하기 위해 각 명령 패킷의 초기 식별 비트들 또는 소정의 비트 위치들에서 다른 지정된 식별 비트들을 사용한다. 예를 들어, 도 2에서 도시되는 바와같이, 초기 비트 "1"은 명령 패킷이 3개의 제어 명령들을 갖는 스칼라 제어 명령 타입임을 나타내며, 초기 비트들 "0 1" 및 "0 0"은 타입(212 및 213)의 명령 패킷들을 나타내는데, 여기서 패킷(212)에서 데이터 및 명령을 가지며, 패킷(213)에서 데이터 및 제어 명령을 갖는다. 각 명령 패킷의 초기 비트들을 디코딩한 이후에, 도 1의 디코드 유닛(101)은 명령 패킷 타입에 따라 각 패킷 명령들을 적절하게 제어 실행 경로(102) 또는 데이터 실행 경로(103)에 전달한다.



도 2의 명령 패킷들을 실행하기 위해, 도 1의 실시예의 프로세서의 명령 디코드 유닛(101)은 메모리로부터 프로그램 패킷들을 순차적으로 페치하며, 프로그램 패킷(211)은 순차적으로 실행된다. 명령 패킷 내에서, 패킷(211)의 명령들은 순차적으로 실행되는데, 21비트 제어 명령에서 64비트 워드의 최하위 단이 먼저 실행되며, 이후에 후속 21비트 제어 명령, 이후에 21비트 제어 명령의 최상위 단이 실행된다. 명령 패킷들(212 및 213) 내에서, 명령들은 동시에 실행될 수 있다(본 발명에 따른 실시예들에서 반드시 필요하지는 않다). 따라서, 도 1의 실시예의 프로세서의 프로그램 순서에서, 프로그램 패킷들은 순차적으로 실행되지만, 패킷 내의 명령들은 패킷 타입(211)에 대해 순차적으로, 또는 패킷 타입들(212 및 213)에 대해 동시에 실행될 수 있다. 하기에, 타입들(212 및 213)의 명령 패킷들은 MD 및 CD-패킷들 각각으로서(일 메모리 및 일 데이터 명령; 그리고 일 제어 명령 및 일 데이터 명령 각각을 포함함) 약칭된다.

21비트 제어 명령들을 사용함에 있어서, 도 1의 실시예는 다른 길이들의 명령들을 갖는 프로세서들에서, 특히 데이터 명령들에 대한 32비트 표준 엔코딩과 제어 코드에 대한 16비트 "고밀도" 엔코딩의 조합을 지원하는 프로세서들에서 발견된 다수의 단점들을 극복한다. 이러한 듀얼 16/32비트 프로세서들에서, 브랜치, 페치 어드레스 또는 다른 수단에 의한 엔코딩 방식들간의 스위칭 수단을 갖는 2개의 개별 디코더들의 사용 또는 각 명령에 대한 듀얼 엔코딩들의 사용으로부터 발생하는 리던던시(redundancy)가 있다. 이러한 리던던시는 본 발명의 일 실시예에 따라 모든 제어 명령들에 대해 단일 21비트 길이를 사용함으로써 제거된다. 더욱이, 21비트 제어 명령들의 사용은 16비트 "고밀도" 엔코딩 방식에서의 불충분한 시맨틱 콘텐츠(semantic content)로부터 발생하는 단점들을 제거한다. 불충분한 시맨틱 콘텐츠 때문에, 16비트 방식을 사용하는 프로세서들은 전형적으로 복사들에 대한 대응하는 코드 팽창(bloat)을 갖는 2개-피연산자 파괴 연산들; 유출(spill)/충전 또는 윈도우 포인터 조작을 위한 코드 팽창을 갖는 레지스터 파일의 서브세트에 대한 윈도우 액세스의 사용; 32비트 포맷으로의 빈번한 역전(frequent reversion)과 같은 설계 타협들의 일부 혼합을 요구하는데, 이는 모든 연산들이 16비트 포맷의 소수의 이용가능한 오퍼코드 비트들로 표현될 수 없기 때문이다. 이러한 단점들은 본 발명의 일 실시예에서 21비트 제어 명령들의 사용에 의해 경감된다.

본 발명의 일 실시예에 따라, 매우 다양한 명령들이 사용될 수 있다. 예를 들어, 명령 부호들은 하기사항들 중 하나가 될 수 있는데, 여기서 C-포맷, M-포맷 및 D-포맷은 제어, 메모리 액세스 및 데이터 포맷 각각을 나타낸다.

[표 1]

명령 부호	인수들(argument)	사용자
instr	명령이 인수를 갖지 않는다	C-포맷 온리(only)
instr dst	명령이 하나의 목적 인수를 갖는다	C-포맷 온리
instr src0	명령이 하나의 소스 인수를 갖는다	C 또는 D-포맷 온리
instr dst, src0	명령이 하나의 목적 및 하나의 소스 인수를 갖는다	D 및 M-포맷 명령들
instr dst, src0, src1	명령이 하나의 목적 인수 및 2개의 소스 인수들을 갖는다	C, D 및 M-포맷 명령들

또한 본 발명의 일 실시예에 따라, C-포맷 명령들 모두는 SISD(단일 명령 단일 데이터) 연산을 제공하는 반면에, M-포맷 및 D-포맷 명령들은 SISD 및 SIMD 연산들을 제공한다. 예를 들어, 제어 명령들은 일반적인 산술, 비교, 및 논리 명령들; 제어 흐름 명령들; 메모리 로드 및 저장 명령들; 그리고 다른 것들을 제공할 수 있다. 데이터 명령들은 일반적인 산술, 쉬프트, 논리 및 비교 명령들; 셔플(shuffle), 소트(sort), 바이트 확장, 및 변경 명령들; 선형 피드백 쉬프트 레지스터 명령들; 그리고 구성가능 딥 실행 유닛(110)(하기에 추가적으로 설명됨)을 통해 사용자-정의 명령들을 제공할 수 있다. 메모리 명령들은 메모리 로드 및 저장 명령; 선택된 데이터 레지스터들을 제어 레지스터들에 복사; 방송된 제어 레지스터들을 데이터 레지스터들에 복사; 그리고 레지스터로의 즉시 명령들을 제공할 수 있다.

본 발명의 일 실시예에 따라, 도 1의 프로세서는 제 1의, 고정 데이터 실행 경로와 제 2 구성가능 데이터 실행 경로를 특징으로 한다. 제 1 데이터 실행 경로는 종래 SIMD 프로세싱 설계들과 유사한 방식으로 레인들로 분리되는 고정 SIMD 실행 유닛을 갖는다. 제 2 데이터 실행 경로는 구성가능 딥 실행 유닛(110)을 갖는다. "딥 실행(deep configuration)"은 결과를 레지스터 파일에 복귀시키기 이전에 단일 발행된 명령에 의해 제공되는 데이터에 관하여 복수의 연속적인 연산들을 수행하는 프로세서의 능력을 가리킨다. 딥 실행의 일 예는 종래 MAC 연산(곱셈 및 덧셈)에서 발견되는데, MAC 연산은 단일 명령으로부터의 데이터에 관하여 2개의 연산들(곱셈 및 덧셈)을 수행하며, 이에 따라 2 차수의 깊이를 갖는다. 딥 실행은

결과 출력 개수와 일치하는 피연산자들 입력 개수, 또는 등가적으로, 밸런시-아웃(valency-out)과 일치하는 밸런시-인(valency-in)에 의해 특징된다. 따라서, 예를 들어, 하나의 결과를 갖는 종래 2개-피연산자 덧셈은 피연산자 개수가 결과들의 개수와 일치하지 않기 때문에 딥 실행의 예가 아닌 반면에, 컨볼루션, 고속 퓨리에 변환들, Trellis/Viterbi 엔코딩, 상관기들, 유한 임펄스 응답 필터들, 및 다른 신호 처리 알고리즘들은 딥 실행의 예가 된다. 어플리케이션-특정 디지털 신호 처리(DSP) 알고리즘들은 전형적으로 비트 레벨에서 메모리-맵핑 방식으로 딥 실행을 수행한다. 그러나, 종래 레지스터-맵핑 범용 DSP는 딥 실행을 수행하지 않으며, 대신에 MAC 연산에서 기껏해야 2 차수 깊이에서 명령들을 실행한다. 대조적으로, 도 1의 프로세서는 2보다 큰 차수들에서 동적으로 구성가능 워드-레벨 명령들을 딥 실행할 수 있는 레지스터-맵핑 범용 프로세서를 제공한다. 도 1의 프로세서에서, 딥 실행 명령의 특성(수행되는 수학적 함수의 그래프)은 명령 자체의 구성 정보에 의해 조정/적합화 될 수 있다. 바람직한 실시예에서, 포맷 명령들은 구성 정보에 할당된 비트 위치들을 포함한다. 이러한 성능을 제공하기 위해, 딥 실행 유닛(110)은 구성가능 실행 자원들을 갖는데, 이는 연산자 모드들, 상호연결들, 및 상수들이 각 어플리케이션을 적합화하도록 업로딩될 수 있음을 의미한다. 딥 실행은 실행의 병행성에 깊이 차원을 추가하는 것인데, 깊이 차원은 SIMD 및 LIW 프로세싱의 초기 개념들에 의해 제공된 폭 차원과 직교한다. 따라서, 이는 범용 프로세서의 작업당 명령을 증가시키는 추가적인 차원을 나타낸다.

도 3은 본 발명의 일 실시예에 따라 구성가능 딥 실행 유닛(310)의 구성요소들을 도시한다. 도 1에서 도시된 바와같이, 구성가능 딥 실행 유닛(110)은 데이터 실행 경로(103)의 일부이며, 이에 따라 도 2의 MD 및 CD-명령 패킷들(212 및 213)으로부터의 데이터-측 명령들에 의해 지시될 수 있다. 도 3에서, 명령(314) 및 피연산자들(315)은 도 1의 명령 디코드 유닛(101)과 데이터 레지스터 파일(105)로부터 딥 실행 유닛(310)에 제공된다. 디코드된 명령(314)의 다중-비트 구성 코드가 제어 맵(316)에 액세스하는데 사용되며, 제어 맵은 딥 실행 유닛의 연산자들을 구성하기 위해 다중-비트 코드를 비교적 복잡한 세트의 구성 신호들로 확장시킨다. 제어 맵(316)은 예를 들어, 룩-업 테이블로서 구현될 수 있는데, 여기서 명령의 서로다른 가능한 다중-비트 코드들이 딥 실행 유닛의 서로다른 가능한 연산자 구성들과 맵핑된다. 제어 맵(316)의 룩-업 테이블의 조회 결과에 기초하여, 크로스바 상호연결(317)은 연산자들(318 내지 321) 세트를 구성하는데, 여기서 임의의 구성이 다중-비트 명령 코드에 의해 표시되는 연산자 구성을 실행하는데 필요하다. 연산자들은 예를 들어, 곱셈 연산자(318), 산술 논리 유닛(ALU) 연산자(319), 상태 연산자(320) 또는 크로스-레인 변경기(321)를 포함할 수 있다. 일 실시예에서, 딥 실행 유닛은 15개 연산자들: 일 곱셈 연산자(318), 8개 ALU 연산자들(319), 4개 상태 연산자들(320), 및 2개 크로스-레인 변경기들(321)을 포함하지만, 다른 개수의 연산자들이 가능하다. 딥 실행 유닛에 제공된 피연산자들(315)은 예를 들어, 2개의 16비트 피연산자들이 될 수 있다. 이들은 피연산자들을 적절한 연산자들(318 내지 321)에 제공할 수 있는 제 2 크로스바 상호연결에 제공된다. 제 2 크로스바 상호연결(322)은 또한 연산자(318 내지 321)로부터의 즉각적인 결과들의 피드백(324)을 수신하며, 이 피드백은 또한 제 2 크로스바 상호연결(322)에 의해 적절한 연산자(318 내지 321)에 제공될 수 있다. 제 3 크로스바 상호연결(323)이 연산자들(318 내지 321)로부터의 결과를 다중화하며, 최종 결과(325)를 출력한다. 다양한 제어 신호들이 연산자들을 구성하는데 사용될 수 있다. 예를 들어, 도 3의 실시예의 제어 맵(316)은 반드시 단일 룩-업 테이블로 구현될 필요가 없으며, 일련의 2개 이상의 중속연결된 룩-업 테이블로서 구현될 수 있다. 제 1 룩-업 테이블의 일 엔트리가 소정의 다중-비트 명령 코드로부터 제 2 룩-업 테이블을 지시할 수 있으며, 이에 따라 복잡한 연산자 구성들을 위해 각 룩-업 테이블에서 요구되는 저장량을 줄일 수 있다. 예를 들어, 제 1 룩-업 테이블은 구성 카테고리들의 라이브러리들로 조직화될 수 있으며, 이에 따라 복수의 다중-비트 명령 코드들이 제 1 룩-업 테이블에서 함께 그룹화되는데, 각 그룹은 그룹의 각 다중-비트 코드에 특정 구성들을 제공하는 후속 룩-업 테이블을 가리킨다.

도 3의 실시예에 따라, 연산자들은 유익하게 다양한 연산자 클래스들로 사전-구성된다. 실제로, 이는 배선(hardwiring) 전략 레벨에 의해 달성된다. 이러한 접근의 이점은 보다 적은 소정의 구성들이 저장될 필요가 있으며, 제어 회로가 보다 단순하게 될 수 있다는 것이다. 예를 들어, 연산자들(318)은 곱셈 연산자들 클래스가 되도록 구성된다. 연산자들(319)은 ALU 연산자들로서 사전-구성되며, 연산자들(320)은 상태 연산자들로서 사전-구성되며, 연산자들(321)은 크로스-레인 변경기들로서 사전-구성되며, 그리고 다른 사전-구성 클래스들이 가능하다. 그러나, 연산자들의 클래스들이 사전-구성되지만은, 소정의 알고리즘을 수행하는데 특정 구성의 최종 배열을 위해 명령들이 적어도: (i) 각 클래스 내의 연산자들의 연결성; (ii) 다른 클래스들로부터의 연산자들과의 연결성; (iii) 임의의 관련 스위칭 수단의 연결성을 배열할 수 있는 실시간 유연성이 있다.

기술분야의 당업자는, 상기 내용이 최적 모드로서 여겨지는 것을 설명하였으며, 본 발명을 수행하는 적절한 다른 모드들의 경우에, 본 발명이 바람직한 실시예의 상세한 설명에서 개시된 특정 장치 구성들 또는 방법 단계들에 국한되지 않아야 함을 이해할 것이다. 또한, 기술분야의 당업자들은, 본 발명이 폭넓은 범위의 응용들을 가지며, 그 실시예들이 본 발명의 사상으로부터 벗어남이 없이 넓은 범위의 서로다른 구현들 및 변형들을 인정하게 됨을 인식할 것이다. 특히, 본원에서 언급된 비트 폭들은 단지 예시적인 것이지 제한적인 의미로 의도된 것이 아니며, 워드들의 절반, 워드들, 길이(long) 등으로 언급되는 임의적인 비트 폭들의 선택이 역시 그러하다.

**도면의 간단한 설명**

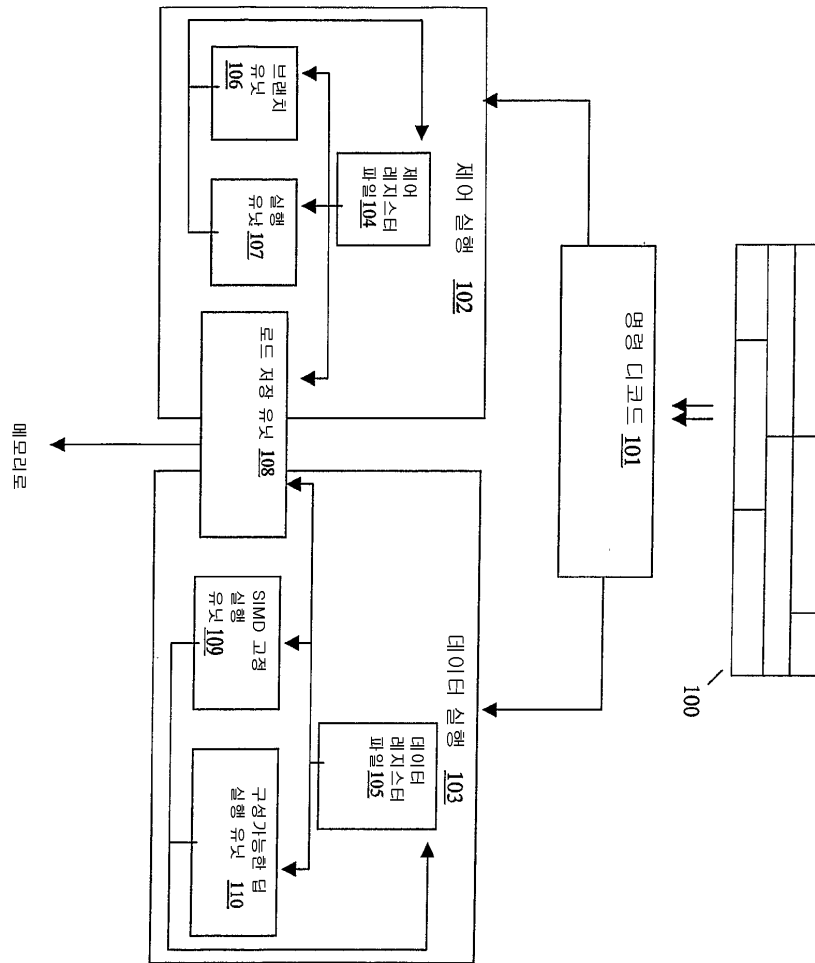
도 1은 본 발명의 일 실시예에 따른 비대칭 듀얼 실행 경로 컴퓨터 프로세서의 블록도이다.

도 2는 본 발명의 일 실시예에 따른 도 1의 프로세서를 위한 예시적인 명령 클래스들을 도시한다.

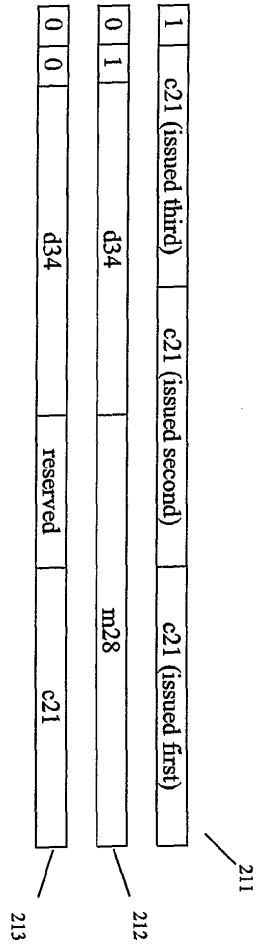
도 3은 본 발명의 일 실시예에 따라 구성가능 댕 실행 유닛의 구성요소들에 대한 개략도이다.

도면

**도면1**



도면2



도면3

