



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년10월20일
(11) 등록번호 10-0988964
(24) 등록일자 2010년10월13일

(51) Int. Cl.

G06F 9/315 (2006.01) G06F 9/32 (2006.01)
(21) 출원번호 10-2008-7029921
(22) 출원일자(국제출원일자) 2007년05월07일
 심사청구일자 2008년12월08일
(85) 번역문제출일자 2008년12월08일
(65) 공개번호 10-2009-0009959
(43) 공개일자 2009년01월23일
(86) 국제출원번호 PCT/US2007/068394
(87) 국제공개번호 WO 2007/134013
 국제공개일자 2007년11월22일
(30) 우선권주장
 11/131,300 2006년05월10일 미국(US)

(56) 선행기술조사문항

US20040064677 A1

US20050203028 A1*

US6631460 A

050001400 A

*는 감사관에 의하여 관공관 군인

(73) 특허권자

퀄콤 인코포레이티드

미국 캘리포니아 샌디에고 모어하우스
드라이브 5775 (우 92121-1714)

(72) 발명자

정, 마오

미국 78759 텍사스 오스틴 #1022 테일러 드라퍼
레이 11250

코드레스쿠, 루씨안

미국 78726 텍사스 오스틴 글라씨어 파케 코브
12505

(74) 대리인

남상선

전체 청구항 수 : 총 33 항

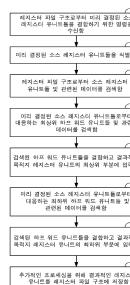
심사관 : 박지은

(54) 마이크로프로세서 내에서 다수의 헤지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 방법 및 시스템

(57) 요약

단일 명령의 실행 동안, 예를 들어, 디지털 신호 프로세서와 같은 마이크로프로세서 내에 있는 다수의 레지스터 유닛들로부터의 대응하는 하프-워드 유닛들을 결합하기 위한 방법 및 시스템이 설명된다. 레지스터 파일 구조로부터 미리 결정된 다른 소스 레지스터 유닛들을 결합하기 위한 명령이 프로세싱 유닛 내에 수신된다. 그 후에 상기 명령은 소스 레지스터 유닛들로부터 대응하는 하프-워드 유닛들을 결합하고 상기 하프-워드 유닛들을 결과적인 목적지 레지스터 유닛의 각각의 부분들로 입력하기 위해 실행된다. 상기 명령의 실행 동안, 미리 결정된 소스 레지스터 유닛들이 식별되고 식별된 레지스터 유닛들로부터 대응하는 최상위 하프-워드 유닛들 및 관련된 데이터가 검색된다. 검색된 하프-워드 유닛들은 추가적으로 결합되어 결과적인 목적지 레지스터 유닛의 각각의 최상위 부분으로 입력된다. 유사하게, 식별된 레지스터 유닛들로부터 대응하는 최하위 하프-워드 유닛들 및 관련된 데이터가 검색된다. 검색된 하프-워드 유닛들은 추가적으로 결합되어 결과적인 목적지 레지스터 유닛의 각각의 최하위 부분으로 입력된다. 최종적으로, 결과적인 목적지 레지스터 유닛은 추가적인 프로세싱을 위해 레지스터 파일 구조에 저장된다.

대표도 - 도4



특허청구의 범위

청구항 1

컴퓨터-판독가능한 매체에 있어서,

인접하지 않은 소스 레지스터들에서 동작하기 위한 명령을 포함하고,

상기 명령은, 프로세싱 시스템에서 실행될 때, 상기 프로세싱 시스템이,

결합된 최상위(most significant) 워드 유닛을 생성하기 위해 레지스터 파일 구조 내에 있는 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 대응하는 최상위 하프-워드(half-word) 유닛들을 결합하고;

상기 결합된 최상위 워드 유닛을 목적지 레지스터 유닛의 최상위 부분으로 입력하고 – 여기서 상기 목적지 레지스터 유닛은 상기 두 개의 인접하지 않은 소스 레지스터 유닛들보다 더 크고, 상기 목적지 레지스터 유닛은 단일 더블-워드 목적지 레지스터 유닛 또는 레지스터 유닛들의 정렬된 쌍을 포함함 –;

결합된 최하위(least significant) 워드 유닛을 생성하기 위해 상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 대응하는 최하위 하프-워드 유닛들을 결합하고;

상기 결합된 최하위 워드 유닛을 상기 목적지 레지스터 유닛의 최하위 부분으로 입력하고; 그리고

상기 목적지 레지스터 유닛을 상기 레지스터 파일 구조에 저장

하도록 하는, 컴퓨터-판독가능한 매체.

청구항 2

제1항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들 각각은 32-비트 폭(wide) 레지스터 유닛이고 상기 목적지 레지스터 유닛은 64-비트 폭 레지스터 유닛이며, 상기 목적지 레지스터 유닛은 4개의 하프-워드 유닛들을 저장하기 위한 크기인, 컴퓨터-판독가능한 매체.

청구항 3

제2항에 있어서,

상기 최상위 하프-워드 유닛들 각각은 16-비트 폭 유닛이고, 상기 최하위 하프-워드 유닛들 각각은 16-비트 폭 유닛인, 컴퓨터-판독가능한 매체.

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 명령의 실행은 추가적으로 상기 프로세싱 시스템이 메모리로부터 상기 두 개의 인접하지 않은 소스 레지스터 유닛들과 관련된 데이터를 검색하고, 상기 두 개의 인접하지 않은 소스 레지스터 유닛들 내에 상기 데이터를 저장하고, 상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터 상기 최상위 하프-워드 유닛들과 관련된 데이터를 검색하고, 상기 최상위 하프-워드 유닛들과 관련된 상기 데이터를 상기 목적지 레지스터 유닛의 상기 최상위 부분에 저장하도록 하는, 컴퓨터-판독가능한 매체.

청구항 6

제1항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 최상위 하프-워드 유닛들은 프로세서의 상기 레지스터 파일 구조 내에 있는 상기 목적지 레지스터 유닛의 최상위 부분으로 연결되는, 컴퓨터-판독가능한 매체.

청구항 7

제1항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들은 레지스터 유닛들의 정렬된 쌍으로서 액세스 가능하지 않은, 컴퓨터-판독가능한 매체.

청구항 8

삭제

청구항 9

삭제

청구항 10

프로세싱 시스템에 의해 수행되는 방법에 있어서,

인접하지 않은 소스 레지스터들에서 동작하기 위해 실행가능한 명령을 수신하는 단계; 및

상기 명령을,

결합된 최상위 워드 유닛을 생성하기 위해 레지스터 파일 구조 내에 있는 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 대응하는 최상위 하프-워드 유닛들을 결합하고;

상기 결합된 최상위 워드 유닛을 목적지 레지스터 유닛의 최상위 부분으로 입력하고 – 상기 목적지 레지스터 유닛은 상기 두 개의 인접하지 않은 소스 레지스터 유닛들보다 더 크고, 상기 목적지 레지스터 유닛은 단일 더블-워드 목적지 레지스터 유닛 또는 레지스터 유닛들의 정렬된 쌍을 포함함 –;

결합된 최하위 워드 유닛을 생성하기 위해 상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 대응하는 최하위 하프-워드 유닛들을 결합하고;

상기 결합된 최하위 워드 유닛을 상기 목적지 레지스터 유닛의 최하위 부분으로 입력하고; 그리고

상기 목적지 레지스터 유닛을 상기 레지스터 파일 구조에 저장

하기 위해 실행하는 단계

를 포함하는, 프로세싱 시스템에 의해 수행되는 방법.

청구항 11

제10항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들 각각은 32-비트 폭 레지스터 유닛이고 상기 목적지 레지스터 유닛은 64-비트 폭 레지스터 유닛이며, 상기 목적지 레지스터 유닛은 4개의 하프-워드 유닛들을 저장하기 위한 크기인, 프로세싱 시스템에 의해 수행되는 방법.

청구항 12

제11항에 있어서,

상기 최상위 하프-워드 유닛들 각각은 16-비트 폭 유닛이고, 상기 최하위 하프-워드 유닛들 각각은 16-비트 폭 유닛인, 프로세싱 시스템에 의해 수행되는 방법.

청구항 13

삭제

청구항 14

제10항에 있어서,

상기 실행하는 단계는,

메모리로부터 상기 두 개의 인접하지 않은 소스 레지스터 유닛들과 관련된 데이터를 검색하는 단계;

상기 두 개의 인접하지 않은 소스 레지스터 유닛들 내에 상기 데이터를 저장하는 단계;

상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터 상기 대응하는 최상위 하프-워드 유닛들과 관련된 데이터를 검색하는 단계; 및

상기 대응하는 최상위 하프-워드 유닛들과 관련된 상기 데이터를 상기 목적지 레지스터 유닛의 상기 최상위 부분에 결합하는 단계

를 더 포함하는, 프로세싱 시스템에 의해 수행되는 방법.

청구항 15

제10항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 최상위 하프-워드 유닛들은 프로세서의 상기 레지스터 파일 구조 내에 있는 상기 목적지 레지스터 유닛의 최상위 부분으로 연결되는, 프로세싱 시스템에 의해 수행되는 방법.

청구항 16

제10항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들은 레지스터 유닛들의 정렬된 쌍으로서 액세스 가능하지 않은, 프로세싱 시스템에 의해 수행되는 방법.

청구항 17

삭제

청구항 18

삭제

청구항 19

인접하지 않은 소스 레지스터들에서 동작하기 위한 명령을 포함하는 데이터를 저장하기 위한 메모리; 및

상기 메모리에 커플링(couple)되는 프로세서

를 포함하는 장치에 있어서,

상기 프로세서는, 프로세싱 유닛 및 상기 프로세싱 유닛에 커플링되는 레지스터 파일 구조를 더 포함하고;

상기 프로세싱 유닛은,

결합된 최상위 워드 유닛을 생성하기 위해 상기 레지스터 파일 구조 내에 있는 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 대응하는 최상위 하프-워드 유닛들을 결합하고;

상기 결합된 최상위 워드 유닛을 목적지 레지스터 유닛의 최상위 부분에 저장하고 – 상기 목적지 레지스터 유닛은 상기 두 개의 인접하지 않은 소스 레지스터 유닛들보다 더 크고, 상기 목적지 레지스터 유닛은 단일 더블-워드 목적지 레지스터 유닛 또는 레지스터 유닛들의 정렬된 쌍을 포함함 –;

결합된 최하위 워드 유닛을 생성하기 위해 상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 대응하는 최하위 하프-워드 유닛들을 결합하고;

상기 결합된 최하위 워드 유닛을 상기 목적지 레지스터 유닛의 최하위 부분에 저장하고; 그리고

상기 목적지 레지스터 유닛을 상기 레지스터 파일 구조에 저장

하기 위해 상기 명령을 실행하도록 적응되는, 장치.

청구항 20

제19항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들 각각은 32-비트 폭 레지스터 유닛이고 상기 목적지 레지스터 유닛은 64-비트 폭 레지스터 유닛이며, 상기 목적지 레지스터 유닛은 4개의 하프-워드 유닛들을 저장하기 위한 크기인, 장치.

청구항 21

제20항에 있어서,

상기 최상위 하프-워드 유닛들 각각은 16-비트 폭 유닛이고, 상기 최하위 하프-워드 유닛들 각각은 16-비트 폭 유닛인, 장치.

청구항 22

삭제

청구항 23

제19항에 있어서,

상기 프로세싱 유닛은,

상기 메모리로부터 상기 두 개의 인접하지 않은 소스 레지스터 유닛들과 관련된 데이터를 검색하고;

상기 두 개의 인접하지 않은 소스 레지스터 유닛들 내에 상기 데이터를 저장하고;

상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터 상기 최하위 하프-워드 유닛들과 관련된 데이터를 검색하고; 그리고

상기 최하위 하프-워드 유닛들과 관련된 상기 데이터를 상기 목적지 레지스터 유닛의 상기 최하위 부분에 결합하고 저장

하기 위해 상기 명령을 실행하도록 더 적응되는, 장치.

청구항 24

제19항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 상기 최상위 하프-워드 유닛들은 상기 프로세서의 레지스터 파일 구조 내에 있는 상기 목적지 레지스터 유닛의 최상위 부분으로 연결되는, 장치.

청구항 25

제19항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들은 레지스터 유닛들의 정렬된 쌍으로서 액세스 가능하지 않은, 장치.

청구항 26

삭제

청구항 27

삭제

청구항 28

인접하지 않은 소스 레지스터들에서 동작하기 위한 명령을 수신하기 위한 수단; 및

상기 명령을,

결합된 최상위 워드 유닛을 생성하기 위해 레지스터 파일 구조 내에 있는 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 대응하는 최상위 하프-워드 유닛들을 결합하고;

상기 결합된 최상위 워드 유닛을 목적지 레지스터 유닛의 최상위 부분으로 입력하고 – 상기 목적지 레지스터 유닛은 상기 두 개의 인접하지 않은 소스 레지스터 유닛들보다 더 크고, 상기 목적지 레지스터 유닛은 단일 더블-워드 목적지 레지스터 유닛 또는 레지스터 유닛들의 정렬된 쌍을 포함함 –;

결합된 최하위 워드 유닛을 생성하기 위해 상기 두 개의 인접하지 않은 소스 레지스터 유닛들로부터의 대응하는 최하위 하프-워드 유닛들을 결합하고;

상기 결합된 최하위 워드 유닛을 상기 목적지 레지스터 유닛의 최하위 부분으로 입력하고; 그리고

상기 목적지 레지스터 유닛을 상기 레지스터 파일 구조에 저장

하기 위해 실행하기 위한 수단

을 포함하는, 장치.

청구항 29

제28항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들 각각은 32-비트 폭 레지스터 유닛이고 상기 목적지 레지스터 유닛은 64-비트 폭 레지스터 유닛이며, 상기 목적지 레지스터 유닛은 4개의 하프-워드 유닛들을 저장하기 위한 크기인, 장치.

청구항 30

제29항에 있어서,

상기 최상위 하프-워드 유닛들 각각은 16-비트 폭 유닛이고, 상기 최하위 하프-워드 유닛들 각각은 16-비트 폭 유닛인, 장치.

청구항 31

삭제

청구항 32

제28항에 있어서,

메모리로부터 상기 두 개의 인접하지 않은 소스 레지스터 유닛들과 관련된 데이터를 검색하기 위한 수단;

상기 두 개의 인접하지 않은 소스 레지스터 유닛들 내에 상기 데이터를 저장하기 위한 수단;

상기 두 개의 인접하지 않은 개별적인 소스 레지스터 유닛들로부터 상기 최하위 하프-워드 유닛들과 관련된 데이터를 검색하기 위한 수단; 및

상기 최하위 하프-워드 유닛들과 관련된 상기 데이터를 상기 결과적인 목적지 레지스터 유닛의 상기 최하위 부분에 결합하기 위한 수단

을 더 포함하는, 장치.

청구항 33

제28항에 있어서,

상기 두 개의 인접하지 않은 소스 레지스터 유닛들은 레지스터 유닛들의 정렬된 쌍으로서 액세스 가능하지 않은, 장치.

청구항 34

삭제

청구항 35

삭제

청구항 36

제1항에 있어서,

상기 명령의 실행은 추가적으로 상기 프로세싱 시스템이 상기 결합된 최하위 워드 유닛을 생성하기 위해 상기 최하위 하프-워드 유닛들을 연결하도록 하는, 컴퓨터-판독가능한 매체.

청구항 37

제36항에 있어서,

상기 명령의 실행은 추가적으로 상기 프로세싱 시스템이 상기 결합된 최상위 워드 유닛을 생성하기 위해 상기 최상위 하프-워드 유닛들을 연결하도록 하는, 컴퓨터-판독가능한 매체.

청구항 38

제10항에 있어서,

상기 결합된 최하위 워드 유닛을 생성하기 위해 상기 최하위 하프-워드 유닛들을 연결하는 단계를 더 포함하는, 프로세싱 시스템에 의해 수행되는 방법.

청구항 39

제10항에 있어서,

상기 결합된 최상위 워드 유닛을 생성하기 위해 상기 최상위 하프-워드 유닛들을 연결하는 단계를 더 포함하는, 프로세싱 시스템에 의해 수행되는 방법.

청구항 40

제19항에 있어서,

상기 프로세서는 상기 결합된 최하위 워드 유닛을 생성하기 위해 상기 최하위 하프-워드 유닛들을 연결하는, 장치.

청구항 41

제19항에 있어서,

상기 프로세서는 상기 결합된 최상위 워드 유닛을 생성하기 위해 상기 최상위 하프-워드 유닛들을 연결하는, 장치.

청구항 42

제1항에 있어서,

상기 명령은,

상기 목적지 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 목적지-레지스터 필드;

제 1 소스 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 제 1 소스-레지스터 필드; 및

제 2 소스 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 제 2 소스-레지스터 필드

를 포함하는, 컴퓨터-판독가능한 매체.

청구항 43

제10항에 있어서,

상기 실행가능한 명령은,

상기 목적지 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 목적지-레지스터 필드;
 제 1 소스 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 제 1 소스-레지스터 필드;
 제 2 소스 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 제 2 소스-레지스터 필드; 및
 상기 소스 레지스터 유닛들 및 상기 목적지 레지스터 유닛의 레지스터 타입들에 관한 정보를 포함하는 레지스터-타입 필드
 를 포함하는, 프로세싱 시스템에 의해 수행되는 방법.

청구항 44

제19항에 있어서,
 상기 명령은,
 상기 목적지 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 목적지-레지스터 필드;
 제 1 소스 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 제 1 소스-레지스터 필드;
 제 2 소스 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 제 2 소스-레지스터 필드; 및
 명령 클래스 정보를 포함하는 명령-클래스 필드
 를 포함하는, 장치.

청구항 45

제28항에 있어서,
 상기 명령은,
 상기 목적지 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 목적지-레지스터 필드;
 제 1 소스 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 제 1 소스-레지스터 필드;
 제 2 소스 레지스터 유닛을 인코딩하기 위한 정보를 포함하는 제 2 소스-레지스터 필드;
 상기 소스 레지스터 유닛들 및 상기 목적지 레지스터 유닛의 레지스터 타입들에 관한 정보를 포함하는 레지스터-타입 필드; 및
 명령 클래스 정보를 포함하는 명령-클래스 필드
 를 포함하는, 장치.

명세서

기술 분야

[0001] 본 발명은 일반적으로 마이크로프로세서들에 관한 것이며, 더욱 상세하게는 단일 명령의 실행 동안, 예를 들어, 디지털 신호 프로세서와 같은 마이크로프로세서 내에서 다수의 레지스터 유닛들로부터의 대응하는 하프워드(half word) 유닛들을 결합하기 위한 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 전형적으로, 컴퓨터 시스템들은 하나 이상의 마이크로프로세서 디바이스들을 포함하며, 각각의 마이크로프로세서 디바이스는 컴퓨터 시스템의 메모리 내에 저장된 값들에 대한 연산들을 수행하고 컴퓨터 시스템의 전체 동작을 관리하도록 구성된다. 이러한 컴퓨터 시스템들은 또한 예컨대 사운드 카드들 및/또는 비디오 카드들과 같은 다양한 멀티미디어 디바이스들을 포함하며, 각각의 멀티미디어 디바이스는 자신 내에서 복잡한 수학적 계산들을 수행하는, 예를 들어, 디지털 신호 프로세서들(DSPs)과 같은 하나 이상의 프로세서들을 포함한다.

[0003] 디지털 신호 프로세서(DSP)는 전형적으로 예컨대 하나 이상의 산술 로직 유닛들(ALU), 하나 이상의 곱셈-및-누산 유닛들(MAC) 및 DSP 내에서 명령들의 세트에 의해 특정되는 연산들을 수행하도록 구성되는 다른 기능 유닛들과 같은 수학적 계산들을 수행하도록 구체적으로 구성된 하드웨어 실행 유닛들을 포함한다. 이러한 연산은 예

컨대 산술 연산들, 논리 연산들, 다른 데이터 프로세싱 연산들을 포함할 수 있으며, 각각의 연산은 관련된 명령들의 세트에 의해 정의된다.

[0004] 일반적으로, DSP 내에 있는 실행 유닛들은 메모리 및 실행 유닛들과 연결된 레지스터 파일로부터 데이터 및 피연산자들을 판독하고, 명령 연산들을 수행하고, 결과들을 레지스터 파일에 저장한다. 레지스터 파일은 다수의 레지스터 유닛들을 포함하며, 각각의 레지스터 유닛은 단일 레지스터로서 또는 두 개의 인접한 레지스터 유닛들의 정렬된(aligned) 쌍들로서 액세스 가능하다. 그러나, 예를 들어, 데이터를 더하거나 또는 빼기 위한 연산들과 같은 특정한 연산들은 명령들의 실행을 위해 적절하게 정렬되기 위해 레지스터 파일 내에 있는 다른 레지스터 유닛들로부터의 데이터를 요구한다. 그리하여, 레지스터 유닛들 내에 저장된 데이터의 적절한 정렬을 가능하게 하기 위해 단일 명령의 실행 동안 DSP 내에 있는 다수의 비-인접(non-adjacent) 또는 다른 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 방법 및 시스템이 요구된다.

발명의 상세한 설명

[0005] 단일 명령의 실행 동안, 예를 들어, 디지털 신호 프로세서와 같은 마이크로프로세서 내에 있는 다수의 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 방법 및 시스템이 설명된다. 일 실시예에서, 레지스터 파일 구조로부터의 미리 결정된 다른 소스 레지스터 유닛들을 결합하기 위한 명령이 프로세싱 유닛 내에 수신된다. 그 다음에 상기 명령은 소스 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하고 상기 하프워드 유닛들을 결과적인 목적지 레지스터 유닛의 개별적인 부분들로 입력하도록 실행된다.

[0006] 일 실시예에서, 미리 결정된 소스 레지스터 유닛들이 식별되고 대응하는 최상위(most significant) 하프워드 유닛들 및 관련된 데이터가 식별된 레지스터 유닛들로부터 검색된다. 검색된 하프워드 유닛들은 추가적으로 결합되어 결과적인 목적지 레지스터 유닛의 개별적인 최상위 부분에 입력된다. 유사하게, 대응하는 최하위(least significant) 하프워드 유닛들 및 관련된 데이터가 식별된 레지스터 유닛들로부터 검색된다. 검색된 하프워드 유닛들은 추가적으로 결합되어 결과적인 목적지 레지스터 유닛의 개별적인 최하위 부분에 입력된다. 최종적으로, 결과적인 목적지 레지스터 유닛은 추가적인 프로세싱을 위해 레지스터 파일 구조에 저장된다.

실시예

[0013] 단일 명령의 실행 동안, 예를 들어, 디지털 신호 프로세서와 같은 마이크로프로세서 내에 있는 다수의 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 방법 및 시스템이 설명된다. 아래에서 설명되는 시스템이 디지털 신호 프로세서로 하여금 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합할 수 있도록 하더라도, 상기 시스템은 단일 명령의 실행 동안 다수의 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결과적인 레지스터 유닛로 결합할 수 있는 마이크로프로세서 디바이스 또는 임의의 다른 프로세싱 유닛을 사용하여 구현될 수 있다는 것을 이해하도록 한다.

[0014] 일 실시예에서, 레지스터 파일 구조로부터의 미리 결정된 다른 소스 레지스터 유닛들을 결합하기 위한 명령이 프로세싱 유닛 내에 수신된다. 그 다음에 상기 명령은 소스 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하고 상기 하프워드 유닛들을 결과적인 목적지 레지스터 유닛의 개별적인 부분들로 입력하도록 실행된다.

[0015] 상기 명령의 실행 동안, 미리 결정된 소스 레지스터 유닛들이 식별되고 대응하는 최상위 하프워드 유닛들 및 관련된 데이터가 식별된 레지스터 유닛들로부터 검색된다. 검색된 하프워드 유닛들은 추가적으로 결합되어 결과적인 목적지 레지스터 유닛의 개별적인 최상위 부분에 입력된다. 유사하게, 대응하는 최하위 하프워드 유닛들 및 관련된 데이터가 식별된 레지스터 유닛들로부터 검색된다. 검색된 하프워드 유닛들은 추가적으로 결합되어 결과적인 목적지 레지스터 유닛의 개별적인 최하위 부분에 입력된다. 최종적으로, 결과적인 목적지 레지스터 유닛은 추가적인 프로세싱을 위해 레지스터 파일 구조에 저장된다.

[0016] 도 1은 명령들의 세트가 실행될 수 있는 디지털 신호 프로세싱 시스템의 블록 다이어그램이다. 도 1에 도시된 바와 같이, 디지털 신호 프로세싱 시스템(100)은 프로세싱 유닛(110), 메모리(150) 및 상기 프로세싱 유닛(110)과 상기 메모리(150)를 연결시키는 하나 이상의 버스들(160)을 포함한다.

[0017] 메모리(150)는 예컨대 VLIW 컴파일러에 의해 생성되는 VLIW 패킷들의 형태로 데이터 및 명령들을 저장하며, 각각의 VLIW 패킷은 하나 이상의 명령들을 포함한다. 패킷의 각각의 명령은 전형적으로 미리 결정된 폭(width)을 가지며, 패킷에 있는 제 1 명령이 전형적으로 패킷의 마지막 명령보다 더 낮은 메모리 주소를 가지도록 메모리(150) 내의 특정한 어드레스를 가진다. 메모리에 대한 어드레싱 방식들은 기술적으로 잘 알려져 있으므로 여기

에서 상세하게 논의되지 않는다. 메모리(150)에 있는 명령들은 버스들(160)을 통해 프로세싱 유닛(110)로 로딩된다.

[0018] 프로세싱 유닛(110)은 하나 이상의 파이프라인들(140)을 통해 하나 이상의 레지스터 파일 구조들(120)과 연결된 중앙 프로세싱 유닛 코어(130)를 더 포함한다. 프로세싱 유닛(110)은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 등을 더 포함할 수 있다.

[0019] 레지스터 파일(120)은 범용 레지스터 유닛들의 세트 및 제어 레지스터 유닛들의 세트를 더 포함하며, 범용 레지스터 유닛들의 세트는 범용 계산들을 지원하고 도 2와 관련하여 아래에서 보다 상세하게 설명되며, 제어 레지스터 유닛들의 세트는 예컨대 하드웨어 루프들, 술어(predicate) 및 다른 특별한 피연산자(operand)들과 같은 특별-목적의 기능을 지원한다.

[0020] 도 2는 디지털 신호 프로세싱 시스템 내에 있는 범용 레지스터 구조의 일 실시예를 나타내는 블록 다이어그램이다. 도 2에 도시된 바와 같이, 일 실시예에서, 레지스터 파일(120) 내에 있는 범용 레지스터 파일 구조(200)는 예컨대 32개의 32-비트 폭을 가지는 레지스터 유닛들(210)과 같은 다수의 레지스터 유닛들을 포함하며, 각각의 레지스터 유닛은 단일 레지스터로서 또는 두 개의 인접한 레지스터 유닛들(210)의 정렬된 쌍들(220)로서 액세스 가능하다.

[0021] 범용 레지스터 유닛들(210)은 적절한 명령에 기반하여 다수의 명칭들로 지칭될 수 있다. 예를 들어, 레지스터 유닛들(210)은 개별적으로 R_0, R_1, \dots, R_{30} 및 R_{31} 로서 지칭될 수 있다. 또한, 레지스터 유닛들 R_0 및 R_1 은 $R_{1:0}$ 으로 지칭되는 64-비트 레지스터 쌍(220)을 형성할 수 있다. 유사하게, 레지스터 유닛들 R_2 및 R_3 은 $R_{3:2}$ 로 지칭되는 64-비트 레지스터 쌍(220)을 형성할 수 있고, 레지스터 유닛들 R_{28} 및 R_{29} 은 $R_{29:28}$ 로 지칭되는 64-비트 레지스터 쌍(220)을 형성할 수 있고, 레지스터 유닛들 R_{30} 및 R_{31} 은 $R_{31:30}$ 로 지칭되는 64-비트 레지스터 쌍(220)을 형성할 수 있다.

[0022] 일 실시예에서, 범용 레지스터 유닛들(210)은 예컨대 어드레스 생성, 스칼라 계산 및 벡터 계산과 같은 일반적인 계산 목적들을 위해 사용되며, 로딩/저장 명령들을 위한 어드레스들, 수치 명령들을 위한 데이터 피연산자들 및 벡터 명령들을 위한 벡터 피연산자들을 포함하는 명령들을 위한 모든 피연산자들을 제공한다. 각각의 레지스터 유닛(210)은 추가적으로 레지스터 유닛(210)의 상위 부분에 위치하는 데이터를 포함하는 최상위 하프워드 유닛 및 레지스터 유닛(210)의 하위 부분에 위치하는 데이터를 포함하는 최하위 하프워드 유닛을 포함한다. 예를 들어, 32-비트 폭의 레지스터 유닛(210)을 고려하면, 레지스터 유닛(210)의 최상위 하프워드 유닛과 최하위 하프워드 유닛은 모두 16-비트 폭의 유닛들이다.

[0023] 도 3은 매우 긴 명령 워드(VLIW: Very Long Instruction Word) 디지털 신호 프로세싱 시스템 아키텍처의 일 실시예를 나타내는 블록 다이어그램이다. VLIW 시스템 아키텍처(300)는 명령 로드 버스(320), 데이터 로드 버스(322) 및 데이터 로드/저장 버스(324)를 통해 디지털 신호 프로세서(DSP)(330)로 연결되는 메모리(310)를 포함한다.

[0024] 일 실시예에서, 메모리(310)는 예컨대 1 내지 4개의 명령들을 가지는 VLIW 패킷들의 형태로 데이터 및 명령들을 저장한다. 메모리(310) 내에 저장된 명령들은 명령 로드 버스(320)를 통해 DSP(330)로 로딩된다. 일 실시예에서, 각각의 명령은 4개의 워드 폭을 가지는 128-비트 명령 로드 버스(320)를 통해 DSP(330)로 로딩되는 32-비트 워드 폭을 가진다. 일 실시예에서, 메모리(310)는 통합된 바이트-어드레싱 가능한 메모리이고, 명령들 및 데이터 모두를 저장하는 32-비트 어드레스 공간을 가지며, 리틀-엔디안(little-endian) 모드에서 동작한다.

[0025] 일 실시예에서, DSP(330)는 시퀀서(335), 4개의 프로세싱 또는 실행 유닛들(345)을 위한 4개의 파이프라인들(340), 예컨대 도 2과 관련하여 상세하게 설명되는 범용 레지스터 파일 구조(200)와 같은 (다수의 범용 레지스터 유닛들을 포함하는) 범용 레지스터 파일 구조(350) 및 제어 레지스터 파일 구조(360)를 포함한다. 시퀀서(335)는 메모리(310)로부터 명령들의 패킷들을 수신하고, 명령 내에 포함된 정보를 사용하여 각각의 수신된 패킷의 각각의 명령에 대한 적절한 파이프라인(340) 및 개별적인 실행 유닛(345)을 결정한다. 패킷의 각각의 명령에 대한 이러한 결정을 한 후에, 시퀀서(335)는 적절한 실행 유닛(345)에 의한 프로세싱을 위해 명령들을 적절한 파이프라인(340)으로 입력한다.

[0026] 일 실시예에서, 실행 유닛들(345)은 추가적으로 벡터 시프트 유닛, 벡터 MAC 유닛, 로드 유닛 및 로드/저장 유닛을 포함한다. 벡터 시프트 유닛(345)는, 예를 들어, S-타입, A64-타입, A32-타입, J-타입 및 CR-타입 명령들을 실행한다. 벡터 MAC 유닛(345)는, 예를 들어, M-타입, A64-타입, A32-타입, J-타입 및 JP-타입 명령들을 실행한다.

행한다. 로드 유닛(345)는 메모리(310)로부터 범용 레지스터 파일 구조(350)로 데이터를 로딩하고, 예를 들어, 로드-타입 및 A32-타입 명령들을 실행한다. 로드/저장 유닛(345)는 범용 레지스터 파일 구조(350)로부터 메모리(310)로 데이터를 로딩 및 저장하고, 예를 들어, 로드-타입, 저장-타입 및 A32-타입 명령들을 실행한다.

[0027] 명령을 수신하는 각각의 실행 유닛(345)은 4개의 실행 유닛들(345)에 의해 공유되는 범용 레지스터 파일 구조(350)를 사용하여 상기 명령을 수행한다. 명령에 의해 요구되는 데이터는 64-비트 데이터 로드 버스(322)를 통해 범용 레지스터 파일 구조(350)로 로딩된다. 패킷의 명령들이 실행 유닛들(345)에 의해 수행된 후에, 결과적인 데이터는 범용 레지스터 파일 구조(350)로 저장되고 그 다음에 64-비트 데이터 로드/저장 버스(324)를 통해 메모리(310)로 로딩되어 저장된다. 일 실시예에서, 예를 들어, RISC-타입 머신들에 대하여, 하나의 명령이 메모리로부터 데이터를 로딩하고, 연산을 수행하고, 그 다음에 결과들을 메모리에 저장할 수 있다. 대안적으로, DSP 유닛들에 대하여, 위의 연산들은 일반적으로 분리된다. 전형적으로, 패킷의 1 내지 4개의 명령들은 하나의 클록 사이클에서 4개의 실행 유닛들(345)에 의해 병렬적으로 수행되며, 각각의 클록 사이클 동안 최대 하나의 명령이 수신되어 파이프라인(340)에 의해 처리된다.

[0028] 일 실시예에서, 실행 유닛(345)은 또한 대응하는 명령을 실행하기 위해 제어 레지스터 파일 구조(360)를 사용할 수 있다. 제어 레지스터 파일 구조(360)는, 예를 들어, 변경자(modifier), 상태 및 술어 레지스터 유닛들과 같은 특별한 레지스터 유닛들의 세트를 포함한다.

[0029] 도 4는 디지털 신호 프로세싱 시스템(100) 내에 있는 다수의 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 방법의 일 실시예를 나타내는 플로우 다이어그램이다. 도 4의 실시예에서 도시된 바와 같이, 프로세싱 블록(410)에서, 디지털 신호 프로세싱 시스템(300) 내에 있는 레지스터 유닛들을 결합하기 위한 명령이 수신된다. 일 실시예에서, 예컨대 DSP(330) 내에 있는 실행 유닛(345)과 같은 프로세싱 유닛은 명령을 수신하고, 아래에서 설명되는 바와 같이, 범용 레지스터 파일 구조(350) 내에 저장된 미리 결정된 소스 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위해 상기 명령을 실행한다. 일 실시예에서, 미리 결정된 소스 레지스터 유닛들은 서로 다르며, 범용 레지스터 파일 구조(350) 내에서 인접하지 않게 위치하며, 그리하여 레지스터 유닛들의 지정된 쌍(200)으로서 액세스 가능하지 않다.

[0030] 프로세싱 블록(420)에서, 예컨대 제 1 32-비트 폭 소스 레지스터 유닛 및 제 2 32-비트 폭 소스 레지스터 유닛과 같은 미리 결정된 소스 레지스터 유닛들이 식별된다. 일 실시예에서, 실행 유닛(345)은 범용 레지스터 파일 구조(350)와 통신하고 결합되도록 요청되는 소스 레지스터 유닛들을 식별한다. 일 실시예에서, 그 다음에 메모리(310)는 64-비트 데이터 로드 버스(322) 및/또는 64-비트 데이터 로드/저장 버스(324)를 통해 범용 레지스터 파일 구조(350)로 명령에 의해 요구되는 데이터를 로딩한다. 대안적으로, 데이터는 이미 식별된 제 1 및 제 2 소스 레지스터 유닛들 내에 저장될 수 있다.

[0031] 프로세싱 블록(430)에서, 식별된 소스 레지스터 유닛들과 관련된 데이터가 검색된다. 일 실시예에서, 실행 유닛(345)은 범용 레지스터 파일 구조(350)로부터 식별된 소스 레지스터 유닛들 및 관련된 데이터를 검색한다.

[0032] 프로세싱 블록(440)에서, 식별된 소스 레지스터 유닛들로부터 대응하는 최상위 하프워드 유닛들이 검색된다. 일 실시예에서, 실행 유닛(345)은 추가적으로 일 실시예에서 제 1 소스 레지스터 유닛의 최상위 또는 상위 하프워드 유닛일 수 있는 예컨대 제 1 소스 레지스터 유닛로부터의 16-비트 폭 유닛과 같은 하프워드 유닛 및 일 실시예에서 제 2 소스 레지스터 유닛의 최상위 또는 상위 하프워드 유닛일 수 있는 예컨대 제 2 소스 레지스터 유닛로부터의 16-비트 폭 유닛과 같은 하프워드 유닛을 검색한다.

[0033] 프로세싱 블록(450)에서, 최상위 하프워드 유닛들은 결합되고 추가적으로 결과적인 목적지 레지스터 유닛의 최상위 또는 상위 부분으로 입력된다. 일 실시예에서, 실행 유닛(345)은 검색된 최상위 하프워드 유닛들을 결합하고 결합된 하프워드 유닛들을 결과적인 목적지 레지스터 유닛의 최상위 또는 상위 부분으로 입력한다.

[0034] 프로세싱 블록(460)에서, 식별된 소스 레지스터 유닛들로부터 대응하는 최하위 하프워드 유닛들이 검색된다. 일 실시예에서, 실행 유닛(345)은 추가적으로 일 실시예에서 제 1 소스 레지스터 유닛의 최하위 또는 하위 하프워드 유닛일 수 있는 예컨대 제 1 소스 레지스터 유닛로부터의 16-비트 폭 유닛과 같은 하프워드 유닛 및 일 실시예에서 제 2 소스 레지스터 유닛의 최하위 또는 하위 하프워드 유닛일 수 있는 예컨대 제 2 소스 레지스터 유닛로부터의 16-비트 폭 유닛과 같은 하프워드 유닛을 검색한다.

[0035] 프로세싱 블록(470)에서, 최하위 하프워드 유닛들은 결합되고 추가적으로 결과적인 목적지 레지스터 유닛의 최하위 또는 하위 부분으로 입력된다. 일 실시예에서, 실행 유닛(345)은 검색된 최하위 하프워드 유닛들을 결합하고 결합된 하프워드 유닛들을 결과적인 목적지 레지스터 유닛의 최하위 또는 하위 부분으로 입력한다.

- [0036] 최종적으로, 프로세싱 블록(480)에서, 결과적인 목적지 레지스터 유닛은 추가적인 프로세싱을 위해 저장된다. 일 실시예에서, 실행 유닛(345)는 결과적인 목적지 레지스터 유닛을 범용 레지스터 파일 구조(350)로 출력하고 추가적인 명령들의 추가적인 프로세싱을 위해 결과적인 목적지 레지스터 유닛을 저장한다.
- [0037] 도 5는 도 4와 관련하여 설명되는 레지스터 유닛들을 결합하기 위한 방법을 나타내는 블록 다이어그램이다. 도 5에 도시된 바와 같이, 소스 레지스터 유닛들 R_S 및 R_T (510 및 520)가 식별되고 범용 레지스터 파일 구조(200)로부터 검색된다.
- [0038] 일 실시예에서, 소스 레지스터 유닛들 R_S 및 R_T (510 및 520)를 결과적인 목적지 레지스터 유닛 R_D (530)로 결합하기 위한 명령은 다음과 같다:
- [0039] $R_D = \text{packHL}(R_S, R_T)$
- [0040] 여기서, R_S 와 R_T 는 개별적인 최상위 또는 상위 하프워드 유닛들 H와 최하위 또는 하위 하프워드 유닛들 L을 가지는 소스 레지스터 유닛들이다.
- [0041] 도 5에 도시된 바와 같이, 명령이 실행되면, 소스 레지스터 유닛 R_S (510)의 최상위 또는 상위 하프워드 유닛 R_{S1} 및 소스 레지스터 유닛 R_T (520)의 최상위 또는 상위 하프워드 유닛 R_{T1} 은 결합되고 추가적으로 목적지 레지스터 유닛 R_D (530)의 상위 부분으로 입력된다. 소스 레지스터 유닛 R_S (510)의 최하위 또는 하위 하프워드 유닛 R_{S2} 및 소스 레지스터 유닛 R_T (520)의 최하위 또는 하위 하프워드 유닛 R_{T2} 는 결합되고 추가적으로 목적지 레지스터 유닛 R_D (530)의 하위 부분으로 입력된다. 예를 들어, R_S 및 R_T (510 및 520)가 모두 32-비트 폭 레지스터 유닛들이라면, 일 실시예에서, 목적지 레지스터 유닛 R_D (530)는 64-비트 폭 레지스터이다.
- [0042] 도 6은 디지털 신호 프로세싱 시스템 내에 있는 다수의 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 명령의 일 실시예에 대한 블록 다이어그램이다. 도 6에 도시된 바와 같이, 도 5와 관련하여 설명된 명령은 예컨대 목적지 레지스터 유닛 R_D 를 인코딩하기 위해 필요한 정보를 포함하는 필드(610), 나중의 인코딩을 위해 예비된 빈(empty) 필드(620), 소스 레지스터 유닛 R_T 를 인코딩하기 위해 필요한 정보를 포함하는 필드(630), 나중의 인코딩을 위해 예비된 다른 빈(empty) 필드(640), 패킷/루프 분석(parse) 비트 정보를 포함하는 필드(650), 소스 레지스터 유닛 R_S 를 인코딩하기 위해 필요한 정보를 포함하는 필드(660), 동작 코드 정보를 포함하는 필드(670), 소스 및 목적지 레지스터 유닛들의 레지스터 타입과 관련된 정보를 포함하는 필드(680) 및 명령 클래스 정보를 포함하는 필드(690)와 같은 다수의 필드들을 포함한다. 그러나, 대안적으로, 본 발명의 범위를 벗어남이 없이, 다른 추가적인 필드들이 또한 위의 명령들의 인코딩 내에 포함될 수 있다.
- [0043] 본 발명의 기술 분야에서 통상의 지식을 가진 자는 정보 및 신호들이 임의의 다양한 상이한 기술들 및 기법들을 이용하여 표현될 수 있다는 것을 이해할 것이다. 예를 들어, 위의 설명에서 참조될 수 있는 데이터, 지시들, 명령들, 정보, 신호들, 비트들, 심볼들 및 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 입자들, 광학장들 또는 입자들, 또는 이들의 임의의 결합에 의해 표현될 수 있다.
- [0044] 본 발명의 기술 분야에서 통상의 지식을 가진 자는 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 회로들 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어 또는 이들 모두의 결합에 의해 구현될 수 있다는 것을 이해할 것이다. 하드웨어 및 소프트웨어의 이러한 상호 호환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 이들의 기능과 관련하여 위에서 일반적으로 설명되었다. 이러한 기능이 하드웨어 또는 소프트웨어로서 구현되는지 여부는 특정한 애플리케이션 및 전체 시스템에 대하여 부과되는 설계 제약들에 따라 좌우된다. 본 발명의 기술 분야에서 통상의 지식을 가진 자는 각각의 특정한 애플리케이션에 대하여 다양한 방식들로 설명된 기능을 구현할 수 있으나, 이러한 구현 결정들은 본 발명의 범위를 벗어나는 것으로 해석되어서는 안 될 것이다.
- [0045] 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들 및 회로들은 여기에서 설명되는 기능들을 수행하도록 설계된 범용 프로세서, 디지털 신호 프로세서(DSP), 애플리케이션 특정 집적 회로(ASIC), 필드 프로그래밍 가능한 게이트 어레이(FPGA) 또는 다른 프로그래밍 가능한 로직 장치, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트들 또는 이들의 임의의 조합을 통해 구현되거나 또는 수행될 수 있다. 범용 프로세서는 마이크로프로세서일 수 있으며, 대안적으로 범용 프로세서는 임의의 기준의 프로세서, 제어기,

마이크로콘트롤러 또는 상태 머신일 수 있다. 프로세서는 또한 컴퓨팅 장치들의 조합, 예를 들어, DSP 및 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 연결된 하나 이상의 마이크로프로세서들 또는 임의의 다른 이러한 구성으로서 구현될 수 있다.

[0046] 여기에서 제시되는 실시예들과 관련하여 설명되는 방법 또는 알고리즘의 단계들은 하드웨어로 직접 구현되거나, 프로세서에 의해 실행되는 소프트웨어 모듈로 구현되거나, 또는 이들의 결합에 의해 구현될 수 있다. 이러한 실시예들은 (컴퓨터의 CPU와 같은) 프로세서 또는 프로세싱 코어의 소정의 형태를 통해 실행되거나 또는 기계 또는 컴퓨터-판독가능한 매체 내에서 구현되거나 또는 실현되는 소프트웨어 프로그램들로서 또는 이러한 소프트웨어 프로그램들을 지원하기 위해 사용될 수 있다는 것을 이해하도록 한다. 기계 판독가능 매체는 기계(예를 들어, 컴퓨터)에 의해 판독가능한 형태로 정보를 저장하거나 또는 전송하기 위한 임의의 메커니즘을 포함한다. 예를 들어, 기계 판독가능 매체는 RAM 메모리, 플래시 메모리, ROM 메모리, EPROM 메모리, EEPROM 메모리, 레지스터들, 하드 디스크, 이동식 디스크, CD-ROM, 또는 기술적으로 공지된 임의의 다른 형태의 저장 매체를 포함할 수 있다. 예시적인 저장 매체는 프로세서와 연결되며, 그 결과 프로세서는 저장 매체로부터 정보를 판독하고 저장 매체로 정보를 기록할 수 있다. 대안적으로, 저장 매체는 프로세서로 통합될 수 있다. 프로세서 및 저장 매체는 ASIC 내에 포함될 수 있다. ASIC은 사용자 터미널 내에 포함될 수 있다. 대안적으로, 프로세서 및 저장 매체는 사용자 터미널 내에 개별적인 컴포넌트들로서 포함될 수 있다.

[0047] 제시된 실시예들에 대한 설명은 임의의 본 발명의 기술 분야에서 통상의 지식을 가진 자가 본 발명을 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 발명의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 발명의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 발명은 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

도면의 간단한 설명

[0007] 도 1은 명령들의 세트가 실행될 수 있는 디지털 신호 프로세싱 시스템의 블록 다이어그램이다.

[0008] 도 2는 디지털 신호 프로세싱 시스템 내에 있는 범용 레지스터 구조의 일 실시예를 나타내는 블록 다이어그램이다.

[0009] 도 3은 매우 긴 명령 워드(VLIW: Very Long Instruction Word) 디지털 신호 프로세싱 시스템 아키텍처의 일 실시예를 나타내는 블록 다이어그램이다.

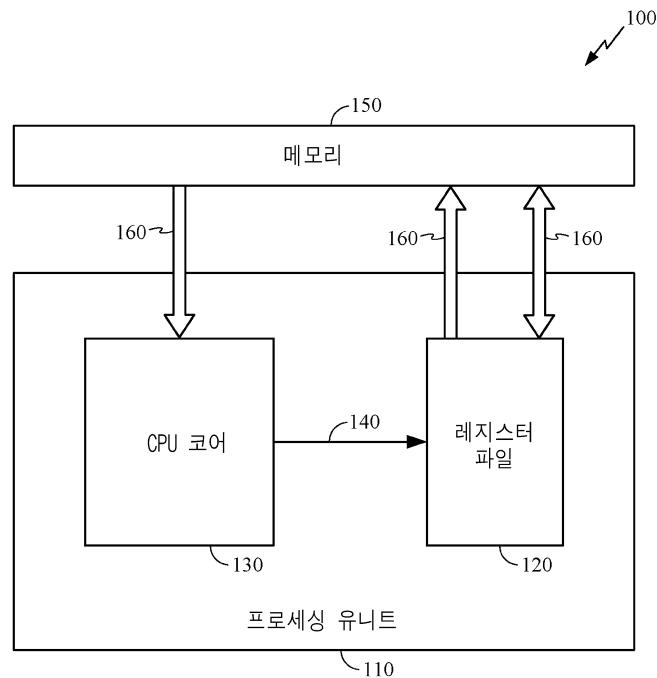
[0010] 도 4는 디지털 신호 프로세싱 시스템 내에 있는 다수의 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 방법의 일 실시예를 나타내는 플로우 다이어그램이다.

[0011] 도 5는 도 4와 관련하여 설명되는 다수의 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 방법을 나타내는 블록 다이어그램이다.

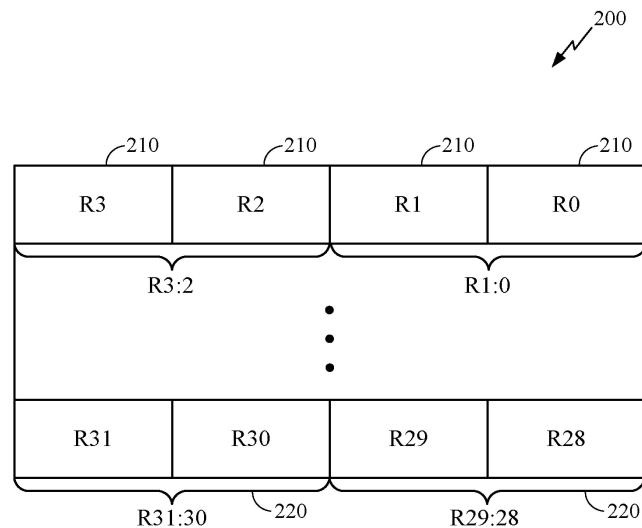
[0012] 도 6은 디지털 신호 프로세싱 시스템 내에 있는 다수의 레지스터 유닛들로부터의 대응하는 하프워드 유닛들을 결합하기 위한 명령의 일 실시예에 대한 블록 다이어그램이다.

도면

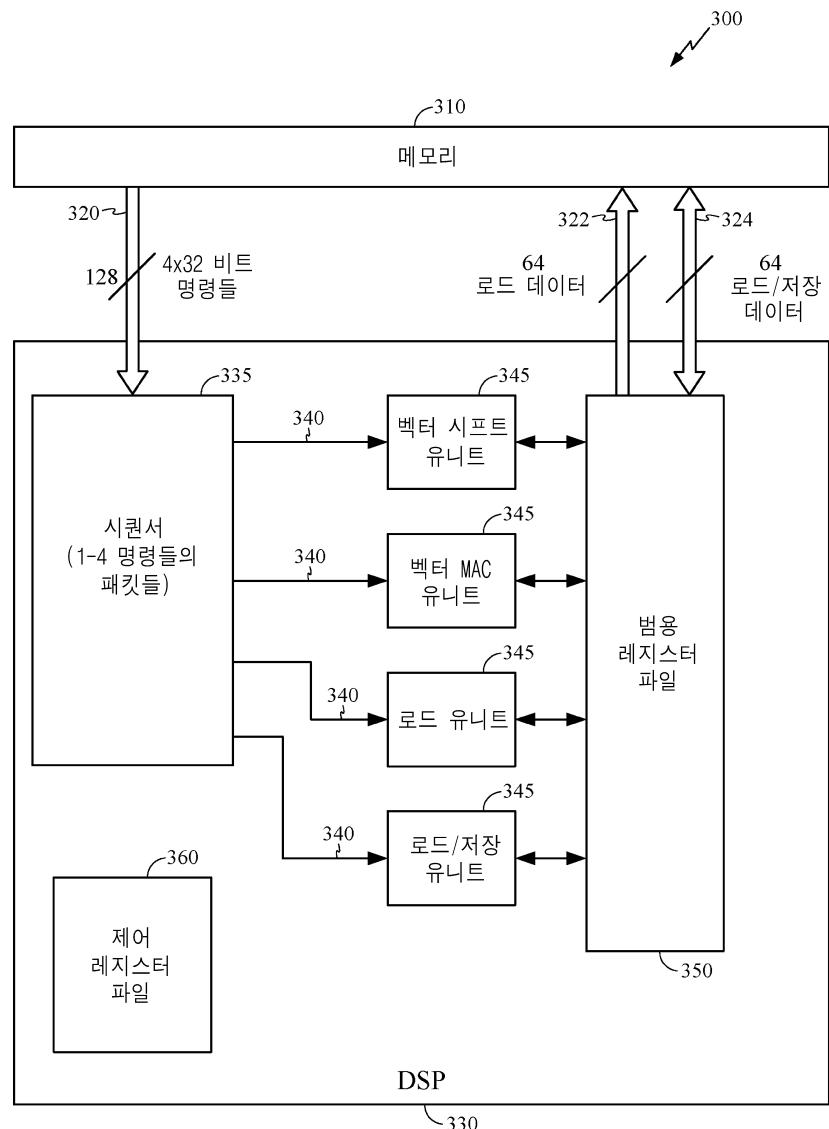
도면1



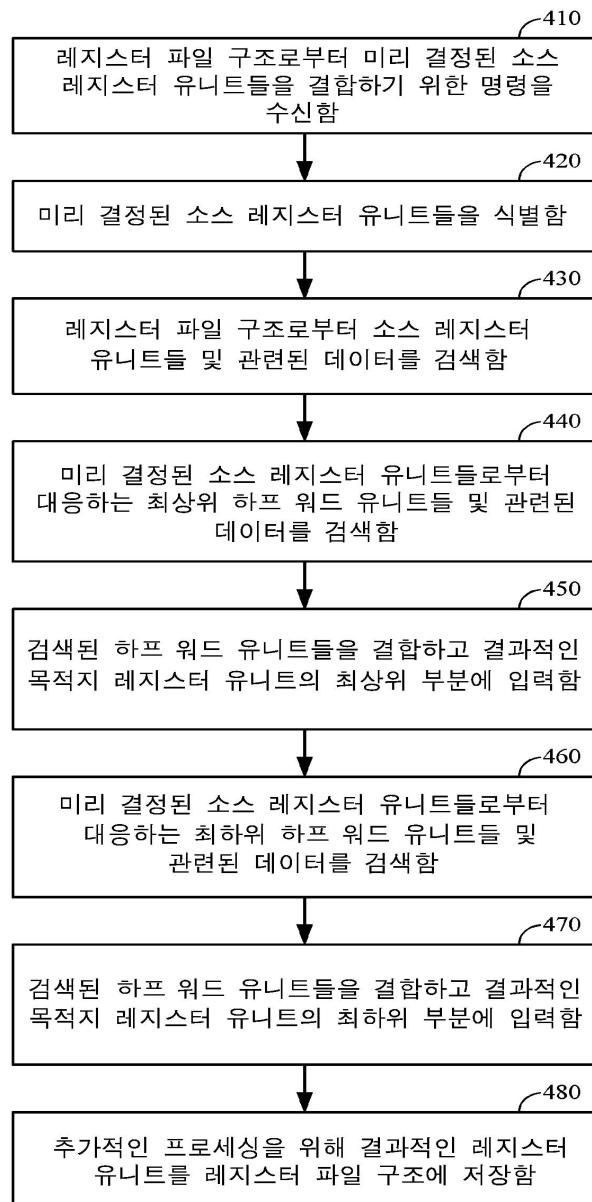
도면2



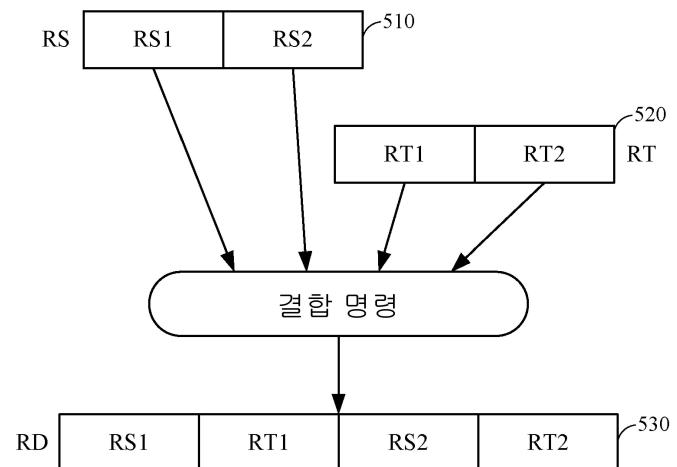
도면3



도면4



도면5



도면6

I CLASS	REG TYPE	MAJ. OP.	RS	PARSE	EMPTY	RT	EMPTY	RD
690	680	670	660	650	640	630	620	610