

(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。 Int. Cl. ⁸ G06F 9/30 (2006.01)	(45) 공고일자 (11) 등록번호 (24) 등록일자	2006년01월24일 10-0545499 2006년01월17일
---	-------------------------------------	--

(21) 출원번호 (22) 출원일자 번역문 제출일자 (86) 국제출원번호 국제출원일자	10-2003-7004527 2003년03월28일 2003년03월28일 PCT/US2001/030318 2001년09월26일	(65) 공개번호 (43) 공개일자 (87) 국제공개번호 국제공개일자	10-2003-0036855 2003년05월09일 WO 2002/27473 2002년04월04일
---	---	---	--

(81) 지정국
 국내특허 : 중국, 일본, 대한민국, 싱가포르,

(30) 우선권주장 09/675,817 2000년09월28일 미국(US)

(73) 특허권자 인텔 코포레이션
 미국 캘리포니아주 95052-8119 산타클라라 피.오.박스 58119 미션 칼리지 불바드 2200

 아나로그 디바이시스 인코포레이티드
 미합중국 메사추세츠(우편번호 02062) 노우드 원 테코놀로지 웨이

(72) 발명자 토마진토마스
 미국텍사스주78750오스틴알라리아드라이브8112

 앤더슨윌리엄씨
 미국텍사스주78731오스틴에드워드마운틴드라이브3910

 로스칼레스피
 미국텍사스주78729오스틴티체스터코트13305

 칼머스케이라
 미국텍사스주78739오스틴로스트오아시스할로우3310

 레빌라주안지
 미국텍사스주78739오스틴오스트렐코브10016

 싱그레비피
 미국텍사스주78758오스틴#829매트릭보울레발드12349

(74) 대리인 김진환
 김두규

심사관 : 성경아

(54) 가변폭 명령어 정렬 엔진

요약

일 실시예에 있어서, 디지털 신호 프로세서는 그 프로세싱 파이프라인에 삽입된 복수의 버블을 감소시키는 로직에 앞서 조사하는 것을 포함한다. 이 프로세서는 복수의 버퍼 안에 명령어들을 포함하는 데이터를 수신하고, 제1 명령어의 크기를 디코딩한다. 제2 명령어의 처음은 제1 명령어의 크기를 토대로 결정된다. 제2 명령어의 크기는 디코딩되고, 프로세서는 제2 명령어를 로딩하여 복수의 버퍼 중 하나를 공핍시킬 것인지 여부를 결정한다.

대표도

도 1

명세서

기술분야

본 발명은 디지털 신호 프로세서에 관한 것으로서, 특히 디지털 신호 프로세서 내의 가변폭 명령어의 정렬에 관한 것이다.

배경기술

디지털 신호 처리는 신호들을 디지털 형태로 표시하고, 수치 계산을 이용하여 그러한 신호 표시를 변환(transformation) 또는 처리하는 것에 관한 것이다. 디지털 신호 처리는 무선 통신, 네트워킹 및 멀티미디어 등의 분야에서 많은 하이테크 제품의 핵심 기술이다. 디지털 신호 처리 기술이 널리 보급된 한 가지 이유는 하이테크 제품을 저렴하고 효율적으로 구현할 수 있고 신뢰할 수 있는 컴퓨팅 기능을 엔지니어에게 제공하는 저가의 강력한 디지털 신호 프로세서(digital signal processor)가 개발되었다는 것이다. 처음에 DSP들을 개발한 이래로, DSP 아키텍처 및 설계는 비디오 레이트 시퀀스들의 매우 복잡한 실시간 처리가 실행될 수 있는 국면까지 발전했다.

DSP들은 흔히 디지털 비디오, 이미징 및 오디오 등의 다양한 멀티미디어 어플리케이션에 이용된다. DSP들은 디지털 신호들을 조작하여 그러한 멀티미디어 파일들을 생성하고 공개할 수 있다.

MPEG-1(Motion Picture Expert Group), MPEG-2, MPEG-4 및 H.263은 디지털 비디오 압축 표준 및 파일 포맷이다. 이들 표준은 각각의 전체 프레임을 저장하는 대신에 하나의 비디오 프레임으로부터 다른 비디오 프레임까지 대부분의 변경 사항을 저장함으로써 디지털 비디오 신호의 고압축률을 달성한다. 이 비디오 정보는 그 후에 복수의 다른 기술들을 이용하여 더욱 압축될 수 있다.

이 DSP는 압축 동안에 비디오 정보에 관한 다양한 동작을 실행하는데 이용될 수 있다. 이들 동작에는 동작 검색(motion search) 및 공간 보간 알고리즘을 포함할 수 있다. 그 중요한 목적은 인접한 프레임 내의 블록사이의 왜곡을 측정하는 것이다. 이들 동작은 과도한 계산력을 요하며, 계산집약적이며, 고속의 데이터 처리량을 요구할 수 있다.

MPEG의 표준 패밀리는 멀티미디어 어플리케이션 및 파일들의 증가하는 대역폭 요건에 보조를 맞추기 위하여 개발 중이다. 이러한 표준의 새로운 버전은 MPEG 컴플라이언트 비디오 처리 장비에 이용된 DSP들에 대한 처리 요건에 더 큰 무게를 두는 보다 복잡한 알고리즘들을 제시하고 있다.

비디오 처리 장치 제조업자들은 흔히 MPEG 및 H.263 표준하에서 비디오 인코딩하는데 주문형 ASIC(application-specific integrated circuits)들에 의존한다. 그러나, ASIC들은 범용 DSP들에 비하여 설계하기 복잡하고, 생산 비용이 많이 들며, 응용 분야에 있어서도 덜 플렉시블하다.

발명의 상세한 설명

본 발명의 목적은 전술한 문제점들을 해결하기 위한 것이다.

도면의 간단한 설명

도 1은 본 발명의 일 실시예에 따른 프로세서를 이용하는 이동 비디오 장치의 블록도이다.

도 2는 본 발명의 일 실시예에 따른 신호 처리 시스템의 블록도이다.

도 3은 본 발명의 일 실시예에 따른 변경 신호 처리 시스템의 블록도이다.

도 4는 본 발명의 일 실시예에 따른 도 1의 프로세서의 전형적인 파이프라인 스테이지를 도시한다.

도 5는 본 발명의 일 실시예에 따른 정렬 멀티플렉서에서 데이터 흐름의 로직 다이어그램이다.

도 6은 본 발명의 일 실시예에 따른 명령어 요청 장치의 블록도이다.

실시예

도 1은 본 발명의 일 실시예에 따른 프로세서를 포함하는 이동 비디오 장치 (100)를 도시한다. 이 이동 비디오 장치(100)는 안테나(105) 또는 디지털 비디오 저장 매체(120), 예컨대 디지털 비디오 디스크(DVD) 또는 메모리 카드로부터 수신되는 인코딩된 비디오 신호로부터 발생된 비디오 이미지를 디스플레이하는 휴대 장치가 될 수 있다. 프로세서(110)는 이 프로세서 동작용 명령어 및 데이터를 저장할 수 있는 메모리(115)(캐시 메모리가 될 수 있음)와 통신한다. 이 프로세서(110)는 마이크로프로세서, 디지털 신호 프로세서(DSP), 슬레이브 DSP를 제어하는 마이크로프로세서, 또는 하이브리드 마이크로프로세서/DSP 아키텍처를 갖는 프로세서가 될 수 있다. 이러한 어플리케이션용으로, 프로세서(110)는 이후에 DSP(110)로 칭해질 것이다.

DSP(110)는 예컨대, 아날로그 대 디지털 변환, 복조, 필터링, 데이터 복원 및 디코딩을 포함하는 다양한 동작을 그 인코딩된 비디오 신호상에서 실행할 수 있다. 이 DSP(110)는 MPEG 패밀리 표준 및 H.263 표준 등의 다양한 디지털 비디오 압축 표준 중 하나에 따라 압축된 디지털 비디오 신호를 디코딩할 수 있다. 이 디코딩된 비디오 신호는 그 후에 디스플레이 구동기(130)로 입력되어 그 비디오 이미지를 디스플레이(125)상에 발생시킬 수 있다.

휴대 장치는 일반적으로 한정된 전원을 갖고 있다. 또한, 비디오 디코딩 동작들은 계산 집약적이다. 따라서, 이러한 장치에 이용하는 프로세서는 비교적 고속의 저전력 장치를 실현할 수 있는 이점이 있다.

DSP(110)는 깊게 파이프라인(deeply pipelined)된 로드/스토어 아키텍처를 가질 수 있다. 파이프라이닝을 적용함으로써, DSP의 퍼포먼스는 비파이프라인된 DSP에 비하여 향상될 수 있다. 제1 명령어를 인출하여, 그 제1 명령어를 실행한 다음에, 제2 명령어를 불러오는 대신에, 파이프라인된 DSP(110)는 그 제1 명령어의 실행과 동시에 제2 명령어를 인출하며, 이것에 의해 명령어 처리량을 향상시킨다. 또한, 파이프라인된 DSP의 클록 사이클은 명령어를 동일한 클록 사이클에서 인출해서 실행해야 하는 비파이프라인된 DSP의 클록 사이클보다 짧아질 수 있다.

이러한 DSP(110)는 비디오 캠코더, 원격 회의, PC 비디오 카드 및 고선명 텔레비전(HDTV)에 이용될 수 있다. 추가적으로, DSP(110)는 또한 이동 전화기, 음성 인식 및 기타 어플리케이션에 이용되는 음성 처리 등의 디지털 신호 처리를 이용하는 기타 기술과 결합하여 이용될 수도 있다.

이후, 도 2를 참조하면, 일 실시예에 따른 DSP(110)를 포함하는 신호 처리 시스템(200)의 블록도가 도시된다. 하나 이상의 아날로그 신호들은 외부 소스, 예컨대 안테나(105)에 의해 신호 조정기(202)에 제공될 수 있다. 신호 조정기(202)는 아날로그 신호에 관하여 특정 사전처리(preprocessing) 기능을 수행할 수 있다. 전형적인 사전처리 기능들은 일부의 아날로그 신호들을 함께 혼합하고, 필터링하고, 증폭하는 등의 기능들을 포함할 수 있다. 아날로그 대 디지털 변환기(ADC)(204)는 신호 조정기(202)로부터 그 사전처리된 아날로그 신호들을 수신하여, 그 사전처리된 아날로그 신호들을 샘플들로 구성하는 디지털 신호들로 변환하기 위하여 결합될 수 있다. 이 샘플들은 신호 조정기(202)에 의해 수신된 아날로그 신호들의 성질에 의해 결정된 샘플링율에 따라 얻어질 수 있다. DSP(110)는 ADC(204)의 출력에서 디지털 신호들을 수신하기 위하여 결합될 수 있고, 또 그 수신된 디지털 신호들에 관한 소정의 신호 변환을 실행하여, 하나 이상의 출력 디지털 신호들을 발생한다. 디지털 대 아날로그 변환기(DAC)(206)는 DSP(110)로부터 출력 디지털 신호들을 수신하기 위하여 결합될 수 있다. DAC(206)는 그 출력 디지털 신호들을 출력 아날로그 신호들로 변환한다. 이 출력 아날로그 신호들은 그 다음에 다른 신호 조정기(208)로 전달된다. 이 신호 조정기(208)는 그 출력 아날로그 신호들에 관한 후처리 기능을 실행한다.

전형적인 후처리 기능들은 앞서 목록된 전처리 기능들과 비슷하다. 주목할 점은 신호 조정기(202, 208), ADC(204) 및 DAC(206)의 다양한 변형예들이 널리 공지되어 있다는 것이다. 이들 장치들의 어떤 적합한 배치는 DSP(110)를 구비한 신호 처리 시스템(200)에 결합될 수 있다.

이제, 도 3을 참조하면, 다른 실시예에 따른 신호 처리 시스템(300)이 도시된다. 이러한 실시예에 있어서, 디지털 수신기(302)는 하나 이상의 디지털 신호들을 수신하고, 이 수신된 디지털 신호들을 DSP(110)로 전달하기 위하여 배치된다. 도 2에 도시된 실시예에 따르면, DSP(110)는 하나 이상의 출력 디지털 신호들을 발생하기 위하여 그 수신된 디지털 신호들에 관한 소정의 신호 변환을 실행할 수 있다. 그 출력 디지털 신호들을 수신하기 위하여 디지털 신호 전송기(304)에 결합될 수 있다. 하나의 전형적인 어플리케이션에 있어서, 신호 처리 시스템(300)은 디지털 수신기(302)가 디지털 스토리지 장치인 메모리(120)에 저장된 데이터를 나타내는 디지털 신호들을 DSP(110)로 전송한다. 이 DSP(110)는 그 다음에 그 디지털 신호들을 처리하고, 그 결과의 출력 디지털 신호들을 디지털 신호 전송기(304)로 전송한다. 이 디지털 신호 전송기(304)는 그 다음에 그 출력 디지털 신호들의 값들이 디스플레이 구동기(130)로 전송되도록 하여, 디스플레이(125) 상에 비디오 이미지를 발생시킨다.

도 4에 도시된 파이프라인은 명령어 인출(402-403), 디코드(404), 어드레스 계산(405), 실행(406-408) 및 후기입(write-back)(409) 단계들을 포함할 수 있는 8개의 단계들을 포함한다. 명령어(i)는 1 클럭 사이클에서 인출된 후에, 신규 명령어, 예컨대 i+1 및 i+2의 인출과 동시에 후속하는 클럭 사이클에서 파이프라인 상에서 동작되고 실행될 수 있다.

파이프라이닝은 추가적인 좌표 문제점 및 위험 요소들을 프로세서 퍼포먼스에 도입할 수 있다. 그 프로그램 흐름의 점프들은 그 파이프라인에서 빈 슬롯 또는 "버블(bubbles)"을 생성할 수 있다. 조건 분기가 채택되거나 예외 또는 중단이 발생하는 상황들은 명령어들의 순차적인 흐름을 변경할 수 있다. 그러한 발생 후에, 신규 명령어는 그 순차적인 프로그램 흐름의 외부로 인출되도록 하여, 그 잔류 명령어들을 상관없는 파이프라인 안에 만든다. 파이프라인에서 데이터 진행, 분기 예측 및 유효 비트를 명령어 어드레스들과 결합하는 등의 방법들은 이러한 복잡성을 처리하는데 적용될 수 있다.

도 5는 본 발명의 일 실시예에 따른 정렬 멀티플렉서(alignment mux)에서 데이터 흐름(500)의 논리도이다. 이 데이터 흐름(500)에 있어서, 명령어들은 메모리(505)로 로딩된다. 이 메모리(505)는 그 명령어들을 저장하는 복수의 버퍼(510, 515)를 포함한다. 일 실시예에 있어서, 이 버퍼(510, 515)들은 64 비트 버퍼이다. 제1 버퍼(510)는 복수의 보다 작은 16 비트 버퍼(520, 525, 530, 535)로 분할된다. 제2 버퍼(515)는 또한 복수의 보다 작은 16 비트 버퍼(540, 545, 550, 555)들로 분할된다. 각각의 그 보다 작은 16 비트 버퍼들(520-555)은 복수의 선택 멀티플렉서(560, 565, 570, 575)에 접속된다. 각각의 그 선택 멀티플렉서(560, 565, 570, 575)들은 멀티플렉서(560, 565, 570, 575)들의 출력을 선택하기 위해서 선택 라인(580)에 접속된다. 이 멀티플렉서(560, 565, 570, 575)들은 그 파이프라인에 이용하기 위하여 복수의 플롭(585, 587, 590, 592)에 저장되는 16 비트 신호를 출력한다.

이 복수의 플롭(585, 587, 590, 592)에 저장된 신호들은 그 파이프라인에 실행되는 명령어를 표시할 수 있다. 이 명령어가 16 비트 이하이면, 제1 플롭(585)만이 데이터를 포함할 수 있다. 명령어 크기가 증가함으로써, 더 많은 플롭(585-592)이 데이터를 포함할 것이다. 예컨대, 32 비트 명령어는 제1 플롭(585) 및 제2 플롭(587)에 데이터를 가질 수 있는 반면에, 64 비트 명령어는 모든 플롭(585-592)에 데이터를 가질 수 있다.

개별 명령어는 초기에 복수의 16 비트 버퍼(520-555) 사이에 분할될 수 있다. 예컨대, 64 비트 명령어는 제3 버퍼(530)에서 개시하여, 제6 버퍼(545)에서 종료할 수 있다. 도 5의 정렬 멀티플렉서는 이 데이터를 복수의 멀티플렉서(560-575)로 전송한 후에, 그 적합한 데이터를 플롭(585-592)으로 전송하기 위해 선택하는 것에 의해 파이프라인에서 처리하기 전에, 이들 명령어들이 확실하게 정렬되도록 한다. 버블들이 파이프라인에 삽입되지 않을 위험성을 줄이기 위해서, 그 정렬 멀티플렉서는 버퍼(510, 515)안의 모든 명령어 데이터가 파이프라인으로 디스패치(dispatch)될 때 메모리(505) 안에 버퍼(510, 515)들을 재로딩한다. 일 실시예에 있어서, 그 메모리(505)는 캐시 메모리가 될 수 있다.

본 발명은 64 비트 또는 이보다 작은 명령어들을 제공하는 64 비트 명령어 레지스터를 이용하여 기술된다. 물론, 본 발명은 (N 비트) 또는 보다 작은 명령어들을 제공하는 특정 크기 명령어 레지스터(N 비트) 상에서 수행될 수 있다. 또한, 본 발명이 2개의 버퍼(510, 515)들로 설명되더라도, 본 발명은 특정 수의 버퍼들을 수용하는 특정 크기로 만들어질 수 있다.

도 6은 본 발명의 일 실시예에 따른 명령어 요청 장치(600)의 블록도이다. 이 명령어 요청 장치(600)는 버퍼(510, 515)들이 비어질 때를 판정하기 위하여 수 사이클에 앞서 조사할 수 있기 때문에, 버퍼(510, 515)들은 이러한 특정 실시예에서 파이프라인으로 유입되는 버블없이 재로딩될 수 있다. 이 명령어 요청 장치(600)의 실시예는 2 사이클의 캐시 대기 시간으로 설명될 것이다. 따라서, 그 명령어 요청 장치는 버퍼(510, 515)들이 연속적으로 재로딩되는 것을 확인하기 위하여 2 사이클보다 앞서 조사해야 한다. 본 발명은 다양한 캐시 대기 시간을 갖는 시스템에 이용될 수 있다는 것을 알 수 있고, 명령어 요청 장치(600)는 버블들이 삽입되지 않는 것을 확인하기 위하여 적어도 그 캐시 대기 시간과 같은 복수의 사이클을 앞서 조사하는 것이 필요할 것이다. 물론, 그 명령어 요청 장치(600)는 그 캐시 대기 시간보다 적게 앞서서 조사할 수 있다. 이러한 실시예에 있어서, 버블들은 파이프라인에 삽입될 수 있다.

명령어 요청 장치(600)는 정렬 멀티플렉서(615)로 입력되는 버퍼(510, 515)로부터 명령어 데이터를 수신한다. 이 정렬 멀티플렉서(615)는 버퍼(510, 515)로부터 수신된 현재의 명령어 데이터를 정렬하여, 그 명령어 데이터로부터 분석된 개별 명령어들을 출력한다. 이 정렬 멀티플렉서(615)에서 데이터 흐름은 도 5를 참조로 앞서 설명되었다. 개별 명령어가 정렬 멀티플렉서(615)로부터 수신될 때, 그 명령어는 폭 비트들을 발생하기 위하여 사전 디코딩된다. 이 명령어(cur_width)의 폭은 이 명령어와 결합된 폭 비트들로부터 블록(620)에서 디코딩된다. 일 실시예에 있어서, 그 폭 비트들은 그 현재 명령어의 폭을 나타내는 2 비트 신호이다. 2 비트 신호로서는 4개의 가능한 폭값들이 있다. 예컨대, 00의 폭 비트는 그 명령어가 무효라는 것을 나타내고, 01의 폭 비트들은 16 비트 명령어를 나타내며, 10의 폭 비트들은 32 비트 명령어를 나타내고, 11의 폭 비트들은 64 비트 명령어를 나타낸다.

멀티플렉서(628)는 버퍼(510, 515)에서 명령어 위치의 분기 목표 어드레스 (branch target address) 및 현재 상태 (cstate)를 수신한다. 이 멀티플렉서(628)는 그 분기 목표 어드레스 또는 현재 상태 중 하나를 선택하여, 현재 상태가 될 수 있는 분기 목표 어드레스의 현재 상태로 플롭(630)을 로딩한다. 그 현재 상태는 그 다음에 블록(625)의 현재 명령의 폭과 결합된다. 그 현재 상태와 현재의 폭을 결합함으로써, 다음 명령어(nstate)의 초기 위치는 결정될 수 있다. 이 위치 정보는 그 후에 제2 정렬 멀티플렉서(635)에 제공되어 그 다음 명령어를 정렬하는데 이용된다. 그 다음 명령어 위치 정보는 또한 다음 클럭 사이클에서 플롭(630)으로 피드백될 수 있다. 다음 클럭 사이클에서, 그 다음 명령어 위치는 현재 명령어 위치가 되고, 이러한 정보는 플롭(630)에서 갱신된다. 그 다음 명령어 위치는 또한 트랜지션 블록(645)에 제공되어 제1 버퍼(510)로부터 제2 버퍼(515)로 트랜지션이 발생한 것을 판정하는데 도움을 준다.

제2 정렬 멀티플렉서(635)는 버퍼(510', 515')로부터의 명령어 데이터를 입력으로서 수신한다. 이 버퍼(510', 515')들은 버퍼(510, 515)와 동일해지거나, 신규 데이터를 포함할 수 있다. 이 버퍼(510', 515')들은 버퍼(510, 515)들이 소모된 경우에 다음 명령어에 대한 신규 데이터를 포함할 수 있다. 제2 정렬 버퍼(635)는 제1 정렬 버퍼(615)가 제1 명령어를 정렬하는 것과 동일한 방법으로 다음 명령어를 정렬한다. 그 다음 명령어의 폭은 그 다음에 블록(640)에서 사전 디코딩되어 그 대응하는 폭 비트들을 결정한다. 그 폭 정보(next_width)는 그 다음에 블록(645)의 트랜지션 로직에 제공된다.

이 트랜지션 로직은 버퍼(510, 515) 중 하나가 다음 명령어의 처리 후에 비워질 것인지 여부를 결정한다. 이 트랜지션 블록(645)은 다음 상태 위치 및 다음 명령어 폭을 입력으로서 포함한다. 이 트랜지션 블록(645)은 그 다음에 버퍼(510, 515) 중 하나가 그 다음 명령어 후에 소모될지 여부를 그 다음 상태 위치 및 명령어 폭을 토대로 결정한다. 예컨대, 그 표시된 다음 상태 명령어가 16 비트 버퍼(530)의 초기에 있고, 그 다음 명령어 폭이 64 비트인 경우에, 그 트랜지션 블록(645)은 그 명령어를 16 비트 버퍼(530, 535, 540, 545)로부터 제거하여, 제1 버퍼(510)을 완전히 비운다. 이 트랜지션 블록(645)은 그 다음에 제1 버퍼(510)가 재로딩될 수 있는 것을 지시하는 플롭(650)으로 신호를 보내서, 그 빈 버퍼(510)를 채우기 위해 메모리에 요청을 발생한다.

트랜지션 블록(645)은 또한 포인터의 최상위 비트(MSB)와 버퍼(520-555)들을 비교함으로써 버퍼(510, 515)들이 공핍되는지 여부를 결정할 수 있다. 예컨대, 각각의 16 비트 버퍼(520-555)들은 관련된 포인터들을 가질 수 있다. 8개의 16 비트 버퍼들이 있기 때문에, 3개의 비트 포인터들은 각각의 버퍼를 유일하게 식별하는데 이용된다. 일 실시예에 있어서, 버퍼(520)는 000의 포인터 값을 갖고, 버퍼(525)는 001의 포인터 값을 갖고, 버퍼(530)은 010의 포인터 값을 갖고, 버퍼(535)는 011의 포인터 값을 갖고, 버퍼(540)는 100의 포인터 값을 갖고, 버퍼(545)는 101의 포인터 값을 갖고, 버퍼(550)는 110의 포인터 값을 갖고, 버퍼(555)는 111의 포인터 값을 갖는다. 따라서, 대형 버퍼(510)를 포함하는 각각의 16 비트 버퍼(520, 525, 530, 535)들은 최상위 비트가 "0"인 포인터 값을 갖는다. 대형 버퍼(515)를 포함하는 각각의 16 비트 버퍼(540, 545, 550, 555)들은 최상위 비트가 "1"인 포인터 값을 갖는다.

전술한 바와 같이, 한 명령어는 복수의 16 비트 버퍼(520-555)들을 점유할 수 있다. 예컨대, 64 비트 명령어는 버퍼(525)에서 개시하여, 버퍼(540)에서 종료할 수 있다. 그 명령어의 개시에서 버퍼(525)의 포인터 값은 001이고, 그 명령어의 종료에서 버퍼(540)의 포인터 값은 100이다. 따라서, 버퍼 포인터의 최상위 비트는 "0"에서 "1"로 변하여, 버퍼(510)로부터 버퍼(515)로 트랜지션을 나타낸다. 그 포인터들의 최상위 비트들과 버퍼(520-555)들을 비교함으로써, 버퍼(510)와 버퍼(515) 사이의 트랜지션은 결정될 수 있다.

본 발명의 다른 실시예에 있어서, 카운터들은 버퍼를 로딩하는 복수의 요청을 관리하는데 이용된다. 로드 요청이 이루어질 때마다, 그 카운터는 증가될 수 있다. 그 카운터는 버퍼가 비워질 때 감소할 것이다. 카운터의 값이 버퍼의 갯수와 같을 때, 본 발명은 요청을 중지한다. 따라서, 카운터들은 버퍼가 데이터를 수용할 수 있는지 여부를 판정하기 위하여 로드 요청을 발생하기 전에 점검될 수 있다.

본 발명의 다양한 변경 및 수정은 당업자라면 쉽게 알 수 있을 것이다. 따라서, 본 발명은 그 사상 또는 필수적인 특징에 벗어남이 없이 다른 특정 형태로 구현될 수 있다.

(57) 청구의 범위

청구항 1.

프로세서에서 명령어들을 정렬하는 방법에 있어서,

제1 명령어를 정렬하는 단계와,

상기 제1 명령어의 크기를 디코딩하는 단계와,

상기 제1 명령어의 크기에 기초하여 제2 명령어의 처음을 결정하는 단계와,

상기 제2 명령어의 크기를 디코딩하는 단계와,

상기 제2 명령어를 처리함으로써 다수의 버퍼 중 하나의 버퍼가 비워지는지 여부를 결정하는 단계와,

만일 상기 제2 명령어를 처리함으로써 다수의 버퍼 중 하나의 버퍼가 비워진다면, 상기 하나의 버퍼로 하여금 추가적인 데이터를 수신하도록 명령하는 단계를 포함하는 명령어 정렬 방법.

청구항 2.

제1항에 있어서,

상기 다수의 버퍼 각각은 다수의 더 작은 보조 버퍼(sub-buffer)로 분할되고,

다수의 보조 버퍼에 다수의 명령어들을 저장하는 단계를 더 포함하는 것인 명령어 정렬 방법.

청구항 3.

제1항에 있어서,

상기 다수의 버퍼 각각은 다수의 더 작은 보조 버퍼(sub-buffer)로 분할되고,

명령어의 처음 부분에 해당하는 보조 버퍼를 가리키는 포인터의 최상위 비트와, 상기 명령어의 끝 부분에 해당하는 보조 버퍼를 가리키는 포인터의 최상위 비트를 비교하여, 다수의 명령어들 중 하나의 명령어를 처리함으로써 버퍼가 비워질 것인지 여부를 결정하는 단계를 포함하는 것인 명령어 정렬 방법.

청구항 4.

제1항에 있어서, 명령어들을 처리하기 전에 다수의 기억 장치 소자에 걸쳐서 제1 명령어를 저장하는 단계를 더 포함하는 것인 명령어 정렬 방법.

청구항 5.

제1항에 있어서, 상기 제1 명령어의 크기를 현재의 명령어 위치에 부가하여 상기 제2 명령어의 처음을 결정하는 단계를 더 포함하는 것인 명령어 정렬 방법.

청구항 6.

제1항에 있어서, 상기 제1 명령어를 정렬하는 단계는, 캐시 대기 시간과 동일한 사이클 수 만큼 미리 정렬하는 단계를 포함하는 것인 명령어 정렬 방법.

청구항 7.

제1항에 있어서, 프로세서에서 명령어들을 정렬하는 것은 디지털 신호 프로세서에서 명령어들을 정렬하는 것을 포함하는 것인 명령어 정렬 방법.

청구항 8.

제1항에 있어서, 메모리에게 상기 다수의 버퍼를 재로딩하라는 요청을 발생시키는 단계를 더 포함하는 것인 명령어 정렬 방법.

청구항 9.

프로세서 내에서 명령어들을 처리하는 방법에 있어서,

캐시 대기 시간과 적어도 동일한 사이클 수 내에, 다수의 버퍼 중 하나의 버퍼로부터 명령어 데이터가 비워질 것인지 여부를 예측하는 단계와,

만일 상기 하나의 버퍼로부터 명령어 데이터가 비워진다면, 상기 하나의 버퍼를 준비시켜 상기 하나의 버퍼에 추가적인 명령어 데이터가 로드되도록 하는 단계를 포함하는 명령어 처리 방법.

청구항 10.

제9항에 있어서, 상기 명령어 데이터에서 제1 명령어의 크기를 디코딩하는 단계와,

상기 제1 명령어의 크기 및 위치에 기초하여, 상기 명령어 데이터 내에서의 제2 명령어의 처음을 결정하는 단계와,

상기 제2 명령어의 크기를 디코딩하는 단계를 더 포함하는 것인 명령어 처리 방법.

청구항 11.

제9항에 있어서, 상기 다수의 버퍼는 다수의 보조 버퍼들로 분할되는 것인 명령어 처리 방법.

청구항 12.

제11항에 있어서, 상기 예측 단계는,

명령어의 처음 부분에 해당하는 보조 버퍼를 가리키는 포인터의 최상위 비트와, 상기 명령어의 끝 부분에 해당하는 보조 버퍼를 가리키는 포인터의 최상위 비트를 비교하여, 다수의 명령어들 중 하나의 명령어를 처리함으로써 다수의 버퍼 중 하나의 버퍼가 비워질 것인지 여부를 결정함으로써, 이루어지는 것인 명령어 처리 방법.

청구항 13.

제9항에 있어서, 상기 명령어 데이터를 정렬하는 단계를 더 포함하는 것인 명령어 처리 방법.

청구항 14.

제9항에 있어서, 디지털 신호 프로세서에서 명령어들을 처리하는 단계를 더 포함하는 것인 명령어 처리 방법.

청구항 15.

제9항에 있어서, 상기 다수의 버퍼를 재로딩하라는 요청을 발생시키는 단계를 더 포함하는 것인 명령어 처리 방법.

청구항 16.

프로세서에 있어서,

다수의 명령어를 포함하는 제1 명령어 데이터를 저장하도록 구성된 다수의 버퍼와,

실행을 위해 상기 다수의 명령어를 정렬하도록 구성된 명령어 요청 장치와,

상기 다수의 명령어의 크기를 결정하도록 구성된 폭 디코더와,

상기 다수의 버퍼 중 하나의 버퍼가 언제 비워질 것인지를 예측하고, 제2 명령어 데이터를 로드하도록 상기 다수의 버퍼 중 하나의 버퍼에게 명령하는 신호를 전송하도록 구성된 트랜지션 검출기를 포함하는 프로세서.

청구항 17.

제16항에 있어서, 상기 다수의 버퍼는 다수의 보조 버퍼로 분할되는 것인 프로세서.

청구항 18.

제16항에 있어서, 상기 트랜지션 검출기는 명령어의 처음 부분에 해당하는 보조 버퍼를 가리키는 포인터의 최상위 비트와, 상기 명령어의 끝 부분에 해당하는 보조 버퍼를 가리키는 포인터의 최상위 비트를 비교하여, 다수의 명령어 중 하나의 명령어를 처리함으로써 버퍼가 비워질 것인지 여부를 결정하는 것인 프로세서.

청구항 19.

제16항에 있어서, 상기 프로세서는 캐시 대기 시간과 동일한 사이클 수 만큼 미리 정렬하는 것인 프로세서.

청구항 20.

제16항에 있어서, 상기 프로세서는 디지털 신호 프로세서인 것인 프로세서.

청구항 21.

프로세서에서 명령어들을 정렬하기 위한 프로그램이 기록된 컴퓨터로 판독가능한 기록 매체에 있어서, 상기 프로그램은,

다수의 버퍼 내에서 명령어들을 포함하는 데이터를 수신하는 단계와,

제1 명령어의 크기를 디코딩하는 단계와,

상기 제1 명령어의 크기에 기초하여 제2 명령어의 처음을 결정하는 단계와,

상기 제2 명령어의 크기를 디코딩하는 단계와,

상기 제2 명령어를 처리함으로써, 상기 다수의 버퍼 중 하나의 버퍼가 비워질 것인지 여부를 결정하는 단계와,

상기 제2 명령어를 처리함으로써, 상기 다수의 버퍼 중 하나의 버퍼가 비워진다면, 상기 하나의 버퍼에게 추가적인 데이터를 수신하도록 명령하는 단계를 수행하는 것인 컴퓨터로 판독가능한 기록 매체.

청구항 22.

제21항에 있어서, 다수의 명령어가 다수의 보조 버퍼에 저장되는 것인 컴퓨터로 판독가능한 기록 매체.

청구항 23.

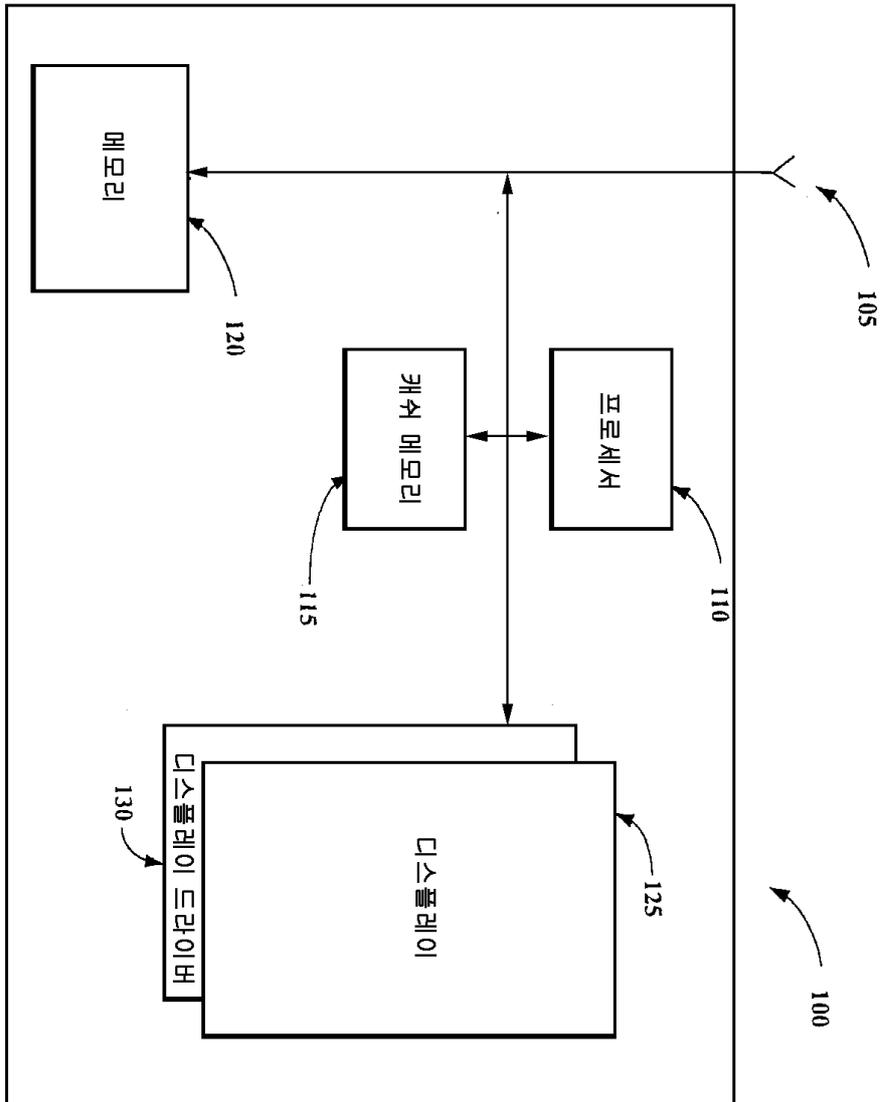
제21항에 있어서, 명령어의 처음 부분에 해당하는 보조 버퍼를 가리키는 포인터의 최상위 비트와, 상기 명령어의 끝 부분에 해당하는 보조 버퍼를 가리키는 포인터의 최상위 비트를 비교하여, 다수의 명령어 중 하나의 명령어를 처리함으로써 버퍼가 비워질 것인지 여부를 결정하는 것인 컴퓨터로 판독가능한 기록 매체.

청구항 24.

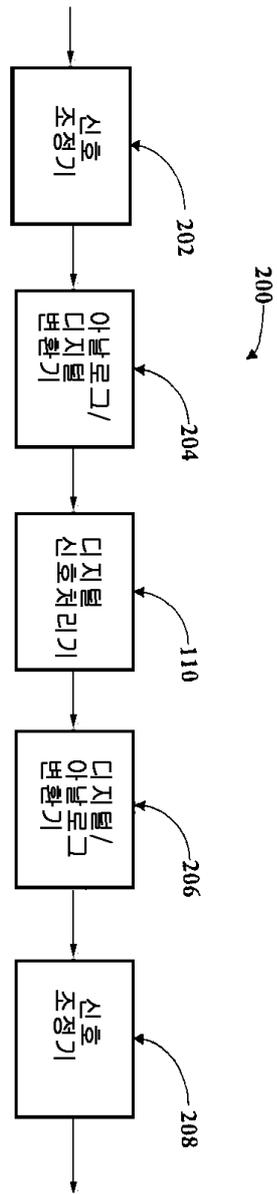
제21항에 있어서, 명령어들을 처리하기 전에 다수의 기억 장치 소자에 걸쳐 제1 명령어가 저장되는 것인 컴퓨터로 판독가능한 기록 매체.

도면

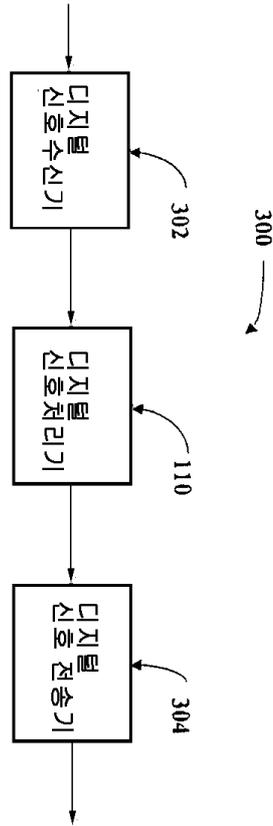
도면1



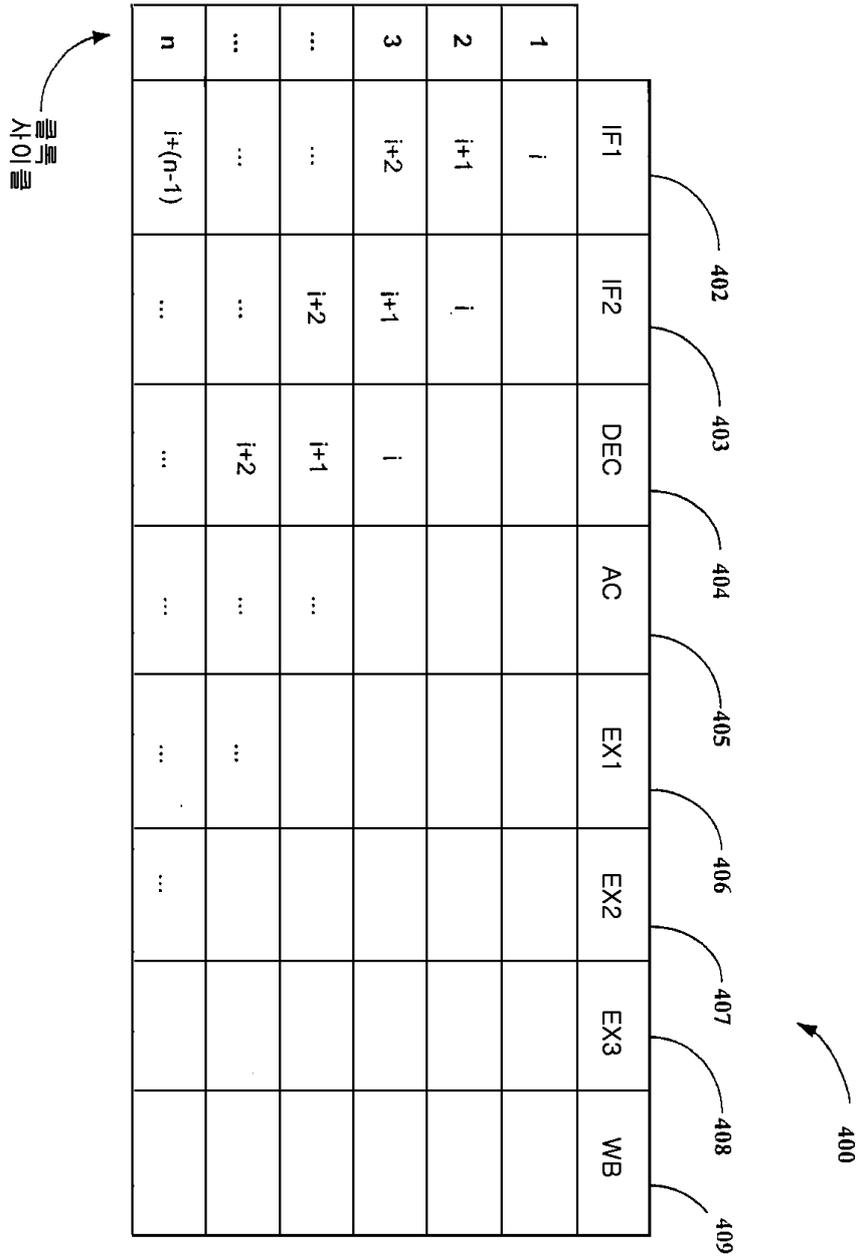
도면2



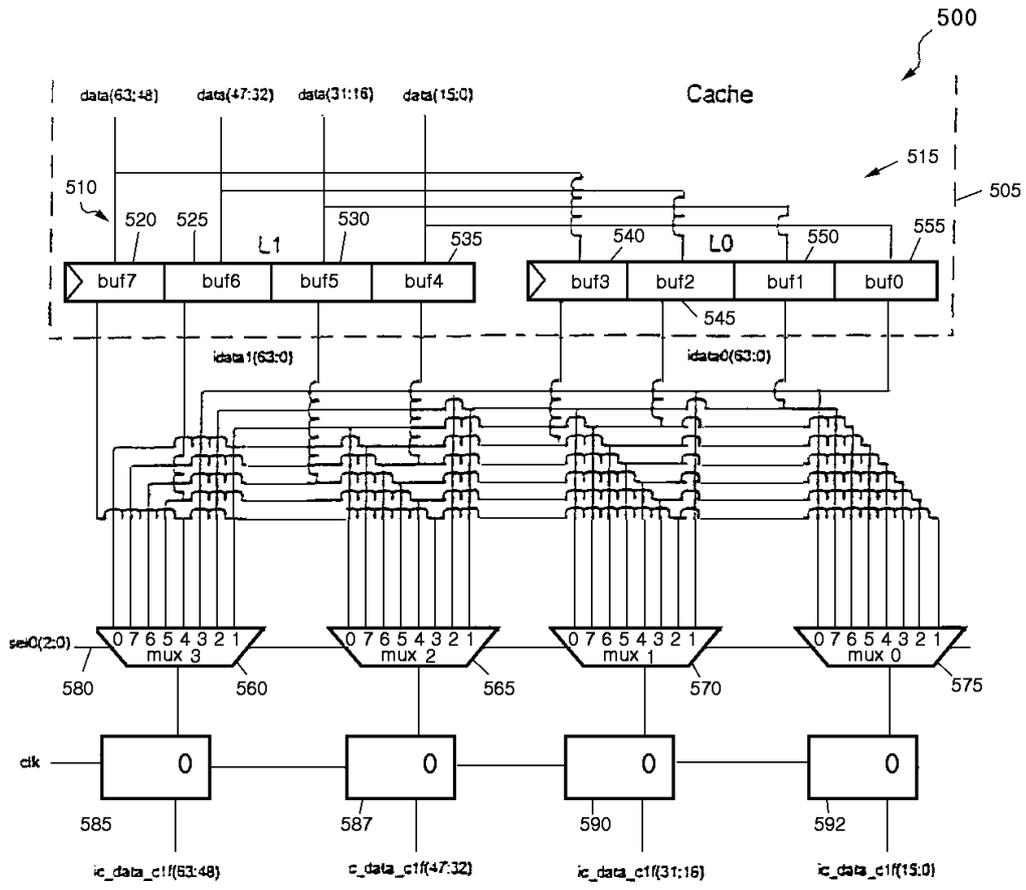
도면3



도면4



도면5



도면6

