(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.[6]
G06F 15/00

(45) 공고일자 2005년08월25일
(11) 등록번호 10-0492041
(24) 등록일자 2005년05월20일

(21) 출원번호 10-1997-0059182
(22) 출원일자 1997년11월11일

(65) 공개번호 10-1998-0042268
(43) 공개일자 1998년08월17일

(30) 우선권주장 08/748,855 1996년11월14일 미국(US)

(73) 특허권자 프리스케일 세미컨덕터, 인크.
미합중국 텍사스 (우편번호 78735) 오스틴 윌리암 캐논 드라이브 웨스트 6501

(72) 발명자 모이어, 윌리엄 씨.
미국, 텍사스, 78620, 드리핑 스프링스, 피어 브랜치로드 1005

아렌드스, 존
미국, 텍사스, 78748, 오스틴, 샤켈포드 드라이브 10709

리, 레아 황
미국, 텍사스 78759, 오스틴 #189, 졸리빌 로드 9500

(74) 대리인 이병호
이범래
정상구
신현문

심사관 : 천대녕

(54) 캐시를구비한데이터처리시스템및그방법

요약

데이터 처리 시스템(20)은 종래의 캐시에서처럼 TAG 어드레스를 저장하기 위한 TAG 배열을 이용하지 않는 캐시(26)를 구비한다. 한 실시예에 따라, 캐시(26)는 미리 결정된 변위(displacement)의 짧은 역방향 브랜치 명령(short backward branch instruction)인 흐름 변경 명령에 응답하여 캐시(26)를 활성 상태로 전이(transition)하는 상태 기계(30)를 이용한다. 미리 결정된 변위는 캐시(26) 내의 엔트리들의 수보다도 작아서, 캐시는 캐시 내에 완전히 포함될 수 있는 루프 내에 프로그램이 있는 한, 활성 상태를 유지할 수 있다. 현재 명령에 대한 유효 비트를 이전 명령과 연관시키는 유효 비트 배열에 대한 룩 어헤드 특성(look ahead feature)이 제공되어, 다음의 명령에 대한 유효 비트는 캐시를 판독하는 동안에 현재의 명령을 판독하는데 이용된 것과 동일한 인덱스로 검사된다.

대표도

도 1

색인어

메인 메모리, 캐시 메모리, 데이터 처리 시스템, 짧은 역방향 브랜치, 룩 어헤드 특성, 인덱스

명세서

도면의 간단한 설명

도 1은 본 발명에 따른 데이터 처리 시스템의 블록도.

도 2는 도 1의 루프 캐시에 대한 블록도.

도 3은 도 1의 상태 기계의 상태도.

도 4는 본 실시예에 따른 TAG 없는 캐시 메모리를 활성화하는 방법의 흐름도.

도면의 주요 부호에 대한 간단한 설명

22 : CPU 24 : 메인 메모리

26 : 루프 캐시 28 : 멀티플렉서

30 : 상태 기계

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

관련 문헌

윌리엄 씨. 모이어(William C. Moyer) 등이 저술하였고 발명의 명칭이 "DISTRIBUTED TAG CACHE MEMORY SYSTEM AND METHOD FOR STORING DATA IN THE SAME"이며, 대리인 참조 번호가 SC90334A인 관련 문헌이 동시에 본원과 출원되었으며, 양수인에게 양도되었다.

기술 분야

본 발명은 일반적으로 데이터 처리에 관한 것이며, 특히 데이터 처리 시스템에서의 캐시 및 그 방법에 관한 것이다.

발명이 이루고자 하는 기술적 과제

저전력 소비는 이동 및 휴대 장비들이 광범위하게 이용됨으로써 마이크로프로세서 및 마이크로컨트롤러 설계시 중요성을 얻고 있다. 통상적인 내장 제어 시스템은 중앙 처리 유닛(CPU) 및 다양한 상이한 형태의 메모리 및 주변 장치들을 포함한다. 상이한 형태의 메모리는 마이크로컨트롤러를 구비하는 집적 회로의 외부 및/또는 동일 집적 회로 상에 있을 수 있으며, 캐시 메모리, ROM(판독 전용 메모리) 및 다양한 SRAM(정적 랜덤 액세스 메모리) 디바이스들을 포함할 수 있다.

용량이 큰 외부 메인 메모리를 액세스하는데는 상당량의 에너지와 시간이 필요하다. 따라서, 메인 메모리에 대한 액세스들의 수를 줄이기 위해, 보다 작고, 보다 빠르고, 보다 효율적인 메모리(때때로, 캐시라고 불림)가 집적 회로 상에서 이용될 수 있다. 집적 회로의 크기를 가능한 작게 하기 위해서는, 집적 회로에 장착되는 메모리를 필요한 만큼만 포함시킬 필요가 있다.

캐시의 성능을 증가시키기 위해 흔히 캐시 TAG가 이용된다. 캐시 TAG는 마이크로프로세서가 제공하는 TAG 어드레스를 수신하여, 요청된 명령들 및/또는 데이터가 캐시 메모리 내에 있는지를 결정한다. 요청된 명령이 캐시 메모리 내에 없다면, 마이크로프로세서는 메인 메모리에서 그 명령을 검색해야 한다. 명령이 캐시에 기록되어 있을 때, 그 명령의 어드레스의 상위 비트들이 TAG 배열 내에 저장된다. 캐시 TAG는 또한 프로세서 발생 어드레스와 TAG 어드레스를 비교하는 비교기를 구비한다. TAG 어드레스와 프로세서 발생 어드레스가 동일하다면, 캐시 "히트(hit)"가 발생하며, 캐시 TAG에 의해 제공된 매치 신호(match signal)는 요청된 데이터가 캐시 메모리 내에 위치해 있음을 나타낸다. TAG 어드레스와 프로세서 발생 어드레스가 동일하지 않다면, 캐시 "미스(miss)"가 발생하며, 매치 신호는 요청된 데이터가 캐시 메모리 내에 없음을 나타낸다. 또한, 유효 비트는 캐시의 비교 사이클 동안 저장된 TAG 어드레스의 유효 히트를 한정(qualify)하기 위해, TAG 어드레스의 일부로 설정할 수도 있다.

명령 및 데이터 검색의 효율성을 향상시키기 위해, 고성능 마이크로프로세서의 메모리 시스템 내에 여러 단계들의 계층을 둘 수 있다. 명령들 및 데이터를 검색하기 위해 CPU가 소비하는 전력은 통상적으로 상기 명령들 및 데이터가 메모리 시스템의 다른 레벨로부터 페치될 때 훨씬 더 크다. 따라서, 메모리의 다른 레벨에 대한 명령 페치들의 수를 줄임으로써, 특히 배터리로 전력 공급하는 응용들에서 전력 소비를 줄이는 것이 바람직하다.

또한, 일부 응용들은 작은 프로그램 루프들에 대해 실행 시간의 상당 부분이 종종 소요된다. 그러한 응용의 한 예로 디지털 신호 처리를 위한 필터의 구현을 들 수 있다. 이들 명령들의 전부 또는 일부를 위한 메인 메모리에 반복적으로 액세스함으로써 전력 소비 및 실행 속도 모두 악영향을 가져올 수 있다. 따라서, 메인 메모리에 반복적으로 액세스할 필요성을 회피하고 캐시를 실행하는데 필요한 표면적을 줄임으로써, 그러한 작은 루프들의 실행에 연관된 전력 소비를 줄이고 실행 속도를 빠르게 하는 것이 바람직하다.

발명의 구성 및 작용

일반적으로, 본 발명은 캐시를 구비한 데이터 프로세싱 시스템을 제공한다. 캐시는 "태그 없음(TAGless)"을 의미하는데, 즉 종래의 캐시에서처럼 TAG 어드레스를 저장하기 위한 TAG 배열을 이용하지 않는다. 또한, 현재의 명령에 대한 유효 비트를 이전의 명령과 연관시켜서, 캐시를 판독하는 동안 다음 명령에 대한 유효 비트가 현재의 명령을 판독하는데 이용된 것과 동일한 인덱스로 검사되고, 따라서 TAG 없는 캐시에 간단한 룩 어헤드 특성을 제공하는 유효 비트 배열이 제공된다.

한 실시예에 따른 캐시는 미리 결정된 변위보다 작은 짧은 역방향 브랜치(SBB : short backward branch)인 흐름 변경(COF : change of flow) 명령에 응답하여 상기 캐시를 활성 상태로 전이하는 상태 기계를 포함한다. 미리 결정된 변위는 캐시(26) 내의 엔트리들의 수보다도 작아서, 캐시는 캐시 내에 완전히 포함될 수 있는 루프 내에 프로그램이 있는 한, 활성 상태를 유지할 수 있다. 본 발명에 따른 캐시 메모리는 예를 들어 통신 응용에서의 필터 기능처럼, 다수 번 반복되는, 빽빽하고(tight) 짧은 루프들을 갖는 프로그램들에 유용하다. 캐시는 "태그 없음(TAGless)"이기 때문에, TAG 배열 및 비교기가 제거되어 보다 적은 집적 회로의 표면적이 요구된다.

본 발명을 도 1 내지 도 4를 참조해서 더 상세하게 설명한다. 신호, 상태 비트, 또는 유사한 장치를 논리적으로 참 또는 논리적으로 거짓 상태로 각각 랜더링하도록 나타낼 때 "주장하다(assert)" 및 "부정하다(negate)"란 용어가 이용될 것이다. 논리적으로 참 상태가 디지털 논리 레벨 1이면, 논리적으로 거짓 상태는 디지털 논리 레벨 0을 나타낸다. 또한 논리적으로 참 상태가 디지털 논리 레벨 0이면, 논리적으로 거짓 상태는 디지털 논리 레벨 1을 나타낸다.

도 1은 본 발명에 따른 데이터 처리 시스템(20)의 블록도이다. 데이터 처리 시스템(20)은 중앙 처리 유닛(CPU;22), 메인 메모리(24), TAG 없는 루프 캐시(26), 멀티플렉서(28) 및 상태 기계(30)를 포함한다. 일반적으로, CPU(22)는 복수의 명령 어드레스들을 발생시킨다. 이 명령 어드레스들은 메인 메모리(24)와 루프 캐시(26)에 제공된다. 각 명령 어드레스는 M개 비트들로 구성되며 상기 M은 정수이다. M개 비트들 중 일부만이 M-P로 표시된 바와 같이, 루프 캐시(26)에 제공되는데, 여기서 P는 M보다 작은 정수이고 명령 어드레스들의 전역(global) TAG 부분을 나타내며, M-P는 INDEX 부분이다. 글로벌 TAG 부분은 TAG 배열을 갖는 종래의 캐시 메모리에서 이용된 TAG 어드레스와 유사하다.

상태 기계(30)는 메인 메모리(24)와 루프 캐시(26)중 어느 것이 명령들을 CPU(22)에 제공할 것인지를 결정 및 제어하는데 이용된다. 메인 메모리(24) 또는 루프 캐시(26)는 명령 어드레스에 응답하여, 명령 어드레스 M에 대응하는 명령을 멀티플렉서(28)를 통해 CPU(22)에 제공한다. 메인 메모리(24)가 명령을 CPU(22)에 제공하면, "DATA IN"이라는 라벨이 붙은 명령도 또한 캐시(26)에 제공된다. 상태 기계(30)에 의해 결정된 대로 루프 캐시(26)에 의해 명령이 제공된다면, 명령 어드레스 M-P에 대응하고 "DATA" 라벨이 붙은 명령은 CPU(22)에 제공된다. 알아두어야 할 점은 다른 실시예들에서 루프 캐시(26)는 예를 들어 CPU(22) 대신에 대안적 데이터 처리 또는 저장 유닛에 명령을 제공할 수도 있다는 점이다.

명령이 어디로부터 검색되어야 할지를 결정하기 위해, 상태 기계(30)는 "COF"(흐름 변경) 라벨이 붙은 제어 신호와 "SBBI"(짧은 역방향 브랜치 명령) 라벨이 붙은 제어 신호를 수신한다. 일련의 명령 어드레스들에서 명령 어드레스가 비연속적 명령 어드레스이면 CPU(22)는 제어 신호 COF를 상태 기계에 제공한다. 루프 캐시(26)는 두 개의 제어 비트들을 상태 기계(30)에 제공한다. 그 중 한 제어 비트는 "TRIGGERING SBB" 이라는 라벨이 붙으며, 다른 제어 비트는 "VALID BIT"라는 라벨이 붙는다. 상태 기계(30)는 또한 멀티플렉서(28)를 제어하기 위해, "LOOP CACHE HIT AND ACTIVE"이라는 라벨이 붙은 루프 캐시 히트 신호와 활성화 신호(activation signal)를 제공한다. 상태 기계(30)는 루프 캐시(26)에 대한 액세스 형태를 제어하기 위해 판독 신호 및 기록 신호와 같은 "CONTROL"이라는 라벨이 붙은 다른 제어 신호들을 제공한다.

루프 캐시(26)는 활성 및 비활성의 두 가지 상태들을 갖는다. 루프 캐시(26)가 활성 상태이면 루프 캐시(26)로부터 데이터를 판독하여 멀티플렉서(28)를 통해 CPU(22)에 제공할 수 있다. 루프 캐시(26)가 비활성 상태이면, 메인 메모리(24)로부터 멀티플렉서(28)를 통해 CPU(22)에 데이터가 제공되며 동시에 메인 메모리(24)로부터 데이터를 판독하여 루프 캐시(26)에 기록할 수 있다. 루프 캐시(26)가 다양한 신호들에 대해 기능하고 응답하는 방식에 대해서는 도 2 내지 도 4를 참조해서 상세히 후술한다.

상태 기계(30)는 TAG 어드레스들을 저장하는 TAG 배열 없이도 제어 신호들 COF 및 SBBI를 이용하여 루프 캐시(26)를 제어한다. 데이터 처리 시스템(20)은, 예를 들어 통신 시스템에서 필터링 기능이나 또는 다른 통신 집약적인 디지털 신호 처리 기능들을 제공하는 명령들과 같이, 많이 반복되는 루프들을 포함하는 명령들을 실행할 때 가장 효과적이다. 루프가 루프 캐시(26)내에서 완전하게 실행되기 위해서는 루프 캐시(26)가 루프의 모든 명령들을 포함할 수 있을 정도로 충분히 커야만 한다. CPU(22)가 루프 캐시(26)내에서 적합한 루프를 검출하면, 그 루프를 결정하는 명령이 트리거링 SBB 비트와 함께 나타나거나 표시된다. 이 때, 다음 명령이 TAG 없는 루프 캐시(26)로부터 CPU(22)에 제공된다. 명령 어드레스의 INDEX 부분(M-P)만이 루프 캐시(26)에 제공되고 CPU(22)가 루프 캐시 내로부터의 명령들을 실행하는지를 트리거링 SBB가 결정하기 때문에 데이터 처리 시스템(20)에서 TAG 어드레스와 별도의 TAG 배열은 필요 없게 된다. 또한 모든 명령 검색을 위한 TAG 비교 사이클도 필요 없게 된다. 설명된 실시예에 따라 메인 메모리(24)는 루프 내에 없는 명령들을 위해 액세스되며 따라서, 루프 캐시(26)는 루프 캐시 내에서 전체적으로 실행될 수 있는 빽빽한 루프들 내에 대부분의 실행 시간이 있는 응용들에서 가장 유용하다.

데이터 처리 시스템 내에서 캐시를 TAG 배열 없이 구현하기 때문에 집적 회로의 표면적이 줄어들며, 따라서 집적 회로의 비용과 복잡성이 감소된다. 또한, TAG 배열에 대한 액세스들이 제거되기 때문에 데이터 처리 시스템에서 전력 소비도 줄어든다.

도 2는 도 1의 루프 캐시(26)를 보다 상세히 설명한다. 도 2에 도시된 루프 캐시(26)는 "DATA IN" 라벨이 붙은 명령들을 저장하는 제 1 부분(32), 제 1 부분(32)에 포함된 명령들의 각각에 대응하는 유효 비트를 저장하기 위한 제 2 부분(34) 및 상태 기계(30)로부터 수신된 짧은 역방향 브랜치(SBB) 비트를 저장하기 위한 제 3 부분(36)의 세 부분으로 되어 있다. 루프 캐시(26)는 또한 래치(38)와 멀티플렉서(40)를 포함한다. 래치(38)는 명령 어드레스의 INDEX 부분을 수신하는 복수의 입력 단자들과, 멀티플렉서(40)의 복수의 제 1 입력 단자들에 결합되어 있는 복수의 출력 단자들을 포함한다. 멀티플렉서(40)의 복수의 제 2 입력 단자들도 또한 INDEX 부분을 수신한다. 멀티플렉서(40)의 다수의 출력 단자는 제 2 부분(34)을 어드레싱하는데 이용된다.

제 2 부분(34)은 멀티플렉서(40)의 출력으로부터 INDEX를 수신한 것에 응답하여 VALID BIT를 제공하기 위한 출력 단자를 포함한다. 제 1 부분(32)은 메인 메모리로부터 명령들을 수신하는 복수의 입력 단자들과, INDEX 수신에 응답하여 명령들을 제공하는 복수의 출력 단자들을 포함한다. 제 1 부분(32)은 1에서 N+ 1까지의 라벨이 붙은 복수의 엔트리들을 포함한다. 알아두어야 할 점은 각각의 엔트리는 하나 이상의 명령을 포함할 수 있다는 점이다. 그러나, 제 1 부분(32)의 엔트리 수는 본 발명을 설명하는데 있어서 중요하지 않으며 특정한 실시예에서 생각해 볼 수 있는 임의의 수가 될 수 있다. 제 2 부분(34)과 제 3 부분(36)은 제 1 부분의 엔트리들 수에 대응하는 다수의 엔트리들을 갖는다. 제 2 부분(34)은 유효 비트들을 저장하며 상술한 바와 같이 유효 비트들은 제 1 부분(32)에 저장된 명령들에 대한 룩-어헤드 기능을 실행하는데 이용된다. 제 2 부분(34)에는 제 1 부분(32) 내의 엔트리 각각에 대응하는 유효 비트가 있다. 대응하는 명령이 유효할 때 유효 비트는 주장되며, 반대로 대응하는 명령이 유효하지 않을 때 유효 비트는 부정된다.

룩-어헤드 기능의 실행을 위해, 각각의 INDEX는 루프 캐시(26)에 대한 기록동작 동안 래치(38)에 래치된다. 기록 신호는 멀티플렉서(40)를 제어하며, 상태 기계(30)에 의해 제공된다. 기록 동작 동안 제 2 부분(34)은 래치(38)에 저장된 INDEX에 의해 어드레싱된다. 룩 어헤드 기능을 제공하기 위해, 래치에 저장된 INDEX는 일련의 명령들에서 이전 명령이

되며, 다음 INDEX 가 제 1 부분(32)을 어드레싱하는데 이용될 때 제공된다. 예를 들어, 도 2에 도시된 바와 같이 기록 동작 동안 인덱스가 N+1 라벨의 엔트리를 어드레싱하면, 멀티플렉서(40) 출력은 제 2 부분(34)의 엔트리 N을 가리키는 "WRITE"의 라벨이 붙은 실선으로 표시된 바와 같이, 엔트리 N에 대응하는 엔트리를 어드레싱하게 된다.

루프 캐시(26)의 판독 동작 동안, 상태 기계(30)로부터의 기록 제어 신호 WRITE는 부정되며, 다음 인덱스는 제 1 부분(32), 제 2 부분(34) 및 제 3 부분(36)의 엔트리 N+1에서 실선 화살표를 이용해서 도 2에 도시된 바와 같이 제 1 부분(32), 제 2 부분(34) 및 제 3 부분(36)을 어드레싱하는데 이용된다. 판독된 명령은 현재의 명령 어드레스에 대응하며, 다음의 연속적인 명령 어드레스에 대응하는 유효 비트는 상기 INDEX와 동시에 판독된다. 따라서, 명령 유효성에 대한 룩 어헤드 특성이 데이터 처리 시스템(20)의 TAG 없는 루프 캐시에 제공된다.

알아두어야 할 점은 상기 룩 어헤드 특성은 캐시가 TAG 배열을 갖는 형태인지 여부에 상관없이 유효 비트 배열을 갖는 어떤 캐시와도 함께 이용될 수 있다는 점이다. 이것은 도 1에 도시된 메인 메모리에서와 같이, 메모리 시스템에서 계층적 구조의 다음 레벨이 선택적 활성화를 허용하여 각각의 액세스에 대한 다음 레벨을 자동으로 활성화될 필요를 회피하며, 이것은 데이터 처리 시스템의 전력 소비를 상당히 감소시키는 결과를 가져온다. 룩 어헤드 특성은 또한 룩 어헤드 특성이 부족한 종래의 제 1 캐시에 의해 야기되었을 수도 있는 명령 스트림에서 스톨들(stalls)을 제거할 수 있는데, 종래의 캐시가 메모리 시스템에서의 계층적 구조의 다음 레벨을 액세스해야 하는지를 결정하기 전에 명령이 유효한지를 알기 위해서는 기다려야 할 수도 있기 때문이다.

요청된 명령이 루프 캐시(26)에 저장되어 있지 않다면, 메인 메모리에 대한 액세스가 실행된다. 명령이 CPU(22)에 제공됨과 동시에 그 명령은 또한 루프 캐시(26)에도 기록된다. 엔트리가 제 1 부분(32)의 위치에 기록될 때, 대응하는 유효 비트가 주장된다. 이 유효 비트는 엔트리가 유효하며 제 2 부분(34)의 이전 위치에 대응하는 위치에 위치하게 됨을 나타낸다. 트리거링 SBB는 루프 캐시의 활성화를 야기하는 "취해진(taken)" SBBI이다. 트리거링 SBB가 검출되면 SBBI 명령에 대응하는 SBB 엔트리는 제 3 부분(36)에 설정된다.

제 1 부분(32) 내의 하나의 엔트리만이 트리거링 SBB 비트와 동시에 표시될 수 있다. 이 하나의 트리거링 SBB를 도 1에 도시된 바와 같이 TRIGGERING SBB 라 불린다.

도 3은 도 1의 상태 기계(30)의 동작을 나타내는 상태도이다. 도 3에 도시된 바와 같이, 상태 기계(30)는 (1) 데이터가 메인 메모리로부터 제공됨을 나타내는 비활성 상태, 및 (2) 데이터가 루프 캐시(26)에서 CPU(22)에 제공될 수 있는 활성 상태의 두 가지 동작 상태를 갖는다. 또한 상태 기계(30)가 한 상태에서 다른 상태로 전이하도록 충족시켜야 하는 조건, 또는 상태 기계(30)가 그 현재의 상태를 유지하도록 충족시켜야 하는 조건이 도 3에 도시되어 있다. 상태 기계(30)가 비활성 상태(40)에서 활성 상태(42)로 전이되기 위해, CPU(22)로부터의 제어 비트 COF가 주장되고 CPU(22)로부터의 SBBI 비트가 주장된다. 상태 기계(30)는 트리거링 SBB와 주장된 COF가 존재할 때 또는 부정된 COF가 존재할 때 루프 캐시(26)를 활성 상태(42)로 유지한다. 트리거링 SBB와 부정된 COF가 존재하거나, 또는 COF가 주장되고 제 2 부분(34)에 트리거링 SBB가 존재하지 않으면, 상태 기계(30)는 루프 캐시(26)가 활성 상태(42)에서 비활성 상태(40)로 전이하게 한다. 상태 기계(30)는 COF가 부정되거나 또는 SBBI가 부정되면 루프 캐시(26)를 비활성 상태(40)로 유지한다.

도 4는 도 1의 루프 캐시(26)를 액세스하는 흐름도이다. 단계(43)에서 루프 캐시는 제 2 부분(34)의 모든 유효 비트를 부정함으로써 무효화된다. 단계(44)에서, CPU(22)는 다음의 명령 어드레스를 계산한다. 단계(46)에서, 루프 캐시(26)가 활성 상태인지를 결정한다. 루프 캐시(26)가 활성이면 단계(46)에서 결정 단계(48)로 YES 경로가 선택된다. 결정 단계(48)에서 이전의 명령이 CPU(22)에 의해 공급된 일련의 명령에서 흐름 변경(COF)이 야기되었는지를 결정한다. 흐름 변경이 있다면 결정 단계(48)에서 결정 단계(50)로 YES 경로가 선택된다. 결정 단계(50)에서, COF가 트리거링 SBB인지를 결정한다. 미리 결정된 변위보다 작은 역방향 브랜치 명령(backward branch instruction)에 의해 흐름 변경이 야기되었다면, 결정 단계(50)에서 결정 단계(52)로 YES 경로가 선택된다. 결정 단계(52)에서, 다음 명령에 대응하는 유효 비트를 살펴봄으로써, 다음 어드레스에 대응하는 다음 명령이 유효 명령인지를 결정한다. 상기 다음 명령에 대한 유효 비트가 주장되었다면, 결정 단계(52)에서 단계(54)로 YES 경로가 선택된다. 단계(54)에서, 상기 명령이 캐시 메모리로부터 CPU(22)에 공급되어 실행된다.

그러나, 결정 단계(52)에서, 다음 명령에 대한 유효 비트가 주장되지 않았다면, 결정 단계(52)에서 단계(70)로 NO 경로가 선택된다. 단계(70)에서, 루프 캐시(26)에는 메인 메모리로부터 다음 명령이 로드되고, 다음 명령이 유효화되며, 다음 명령이 CPU(22)에 공급되어 실행된다. 주목할 점은 언제라도 상기 루프 캐시에는 명령이 로드되고 대응하는 유효 비트가 설정된다는 점이다.

결정 단계(50)를 되돌아가서, 결정 단계(48)에서 흐름 변경 명령이 트리거링 SBB가 아닌 것으로 결정되었다면, 단계(68)로 NO 경로가 선택된다. 단계(68)에서, 유효 비트 배열이 있는 제 2 부분(34)(도 2)은 무효화되고, 다음 명령이 메인 메모리로부터 CPU(22)에 공급되고, 또한 루프 캐시(26)에 로드된다. 또한, 루프 캐시(26)는 비활성 상태(40)로 전이된다(도 3). 단계(68)에서, 흐름은 단계(44)의 시작으로 되돌아가고, 여기서 다음 명령 어드레스가 CPU(22)에 의해 계산된다.

결정 단계(48)로 되돌아가서, 이전의 명령이 흐름 변경이 아니었다면, 단계(66)로 NO 경로가 선택된다. 단계(66)에서, 이전의 명령이 캐시 메모리를 활성화한 미리 결정된 변위의 역방향 브랜치(트리거링 SBB)를 야기했는지를 결정한다. 그 명령이 트리거링 SBB가 아니라면, 결정 단계(66)에서 결정 단계(52)로 NO 경로가 선택된다. 이전의 명령이 트리거링 SBB이면, 단계(66)에서 단계(68)로 YES 경로가 선택되고 그 흐름은 상술한 바와 같이 단계(68)에서 계속된다.

결정 단계(46)로 되돌아가서, 루프 캐시(26)가 비활성임을 나타내는 NO 경로가 선택되면 결정 단계(56)에서 흐름 변경이 있었는지를 결정한다. 이전의 어드레스에 대해 결정 단계(56)에서 흐름 변경이 없었다면, 단계(62)로 NO 경로가 선택된다. 단계(62)에서, 루프 캐시(26)에는 다음 명령이 로드되고 그 다음 명령은 메인 메모리로부터 CPU(22)에 공급된다. 단계(62)후, 흐름은 단계(44)에서 다시 계속된다.

결정 단계(56)로 되돌아가서, 이전의 명령이 흐름 변경이었다면 결정 단계(58)로 YES 경로가 선택된다. 결정 단계(58)에서, 짧은 역방향 브랜치 명령(SBBI)이 CPU(22)에 의해 주장되어 상태 기계(30)에 공급되었는지를 결정한다. SBBI가 주장되지 않았다면, 단계(64)로 NO 경로가 선택되고 루프 캐시(26)가 무효화되며, 루프 캐시(26)와 CPU(22)는 메인 메모리(24)로부터 다음 명령을 공급받는다. 단계(64)후, 프로그램 흐름은 단계(44)에서 다시 계속된다.

결정 단계(58)로 되돌아가서, 명령이 짧은 역방향 브랜치 명령인 것으로 표시되었다면, 단계(60)로 YES 경로가 선택된다. 단계(60)에서, 루프 캐시(26)가 활성화되고 트리거링 브랜치 명령에 대한 SBB 비트가 설정된다. 또한, 다음 명령이 루프 캐시(26)로부터 CPU(22)에 공급된다. 단계(60)후, 프로그램 흐름은 단계(44)에서 다시 계속된다. 알아두어야 할 점은 이들 단계들은 명령 어드레스들이 더 이상 없을 때까지 각각의 다음 명령 어드레스에 대해 반복되며, 명령 어드레스가 더 이상 없을 때 프로그램은 종료된다는 점이다.

발명의 효과

본 발명을 양호한 실시예의 내용으로 서술하였으나, 본 발명이 다양한 방식들로 수정될 수 있으며 상술한 것 외에 많은 실시예들을 취할 수 있다는 것은 당업자에게는 명백하다.

따라서, 첨부된 클레임에 의해 본 발명의 정신 및 범주 내에 있는 본 발명의 모든 수정들을 포함하도록 의도된다.

(57) 청구의 범위

청구항 1.

데이터 처리 시스템(20)에 있어서,

명령 어드레스들을 발생하고 명령들을 처리하는 중앙 처리 유닛(22)으로서, 상기 명령들 중 하나는 상기 명령 어드레스들 중 하나에 대응하는, 상기 중앙 처리 유닛(22);

상기 중앙 처리 유닛(22)에 결합되어, 상기 명령들을 공급 및 수신하는 제 1 메모리(24);

상기 중앙 처리 유닛(22)과 상기 제 1 메모리(24)에 결합되어, 상기 명령들을 수신 및 공급하는 제 2 메모리(26); 및

상기 중앙 처리 유닛(22)과 상기 제 2 메모리(26)에 결합되어, 상기 제 2 메모리(26)를 활성화하고 상기 명령이 유효함을 나타내는 유효 비트에만 응답하여 상기 명령을 상기 중앙 처리 유닛(22)에 공급하는 상태 기계(30)를 포함하는 데이터 처리 시스템(20).

청구항 2.

데이터 처리 시스템(20)에 있어서,

명령 어드레스들을 발생하고 상기 명령들을 처리하는 중앙 처리 유닛(22)으로서, 상기 명령들 중 하나는 상기 명령 어드레스들 중 하나에 대응하는, 중앙 처리 유닛(22);

상기 중앙 처리 유닛(22)에 결합되어, 데이터를 공급 및 수신하는 메인 메모리(24); 및

상기 중앙 처리 유닛(22)과 상기 메인 메모리(24)에 결합되어, 상기 명령들을 수신하는 캐시 메모리(26)를 포함하고, 상기 캐시 메모리(26)는,

명령들을 저장하는 제 1 부분(32)과,

유효 비트들을 저장하는 제 2 부분(34)을 포함하고,

상기 제 1 부분(32) 내에 저장된 현재 명령의 상기 유효 비트는 이전 명령의 명령 어드레스에서의 상기 제 2 부분(34) 내에 저장되는, 데이터 처리 시스템(20).

청구항 3.

데이터 처리 시스템(20)에서 명령들을 취급하는 방법으로서, 상기 데이터 처리 시스템(20)은 중앙 처리 유닛(22), 메인 메모리(24) 및 캐시 메모리(26)를 포함하는, 상기 명령들 취급 방법에 있어서,

상기 캐시 메모리(26)를 무효화하는 단계;

상기 메인 메모리(24) 또는 상기 캐시 메모리(26) 중 어느 쪽에 액세스하기 위한 다음 어드레스를 계산하는 단계;

상기 캐시 메모리(26)가 활성인지를 결정하는 단계;

상기 중앙 처리 유닛(22)에 의해 공급된 일련의 명령 어드레스들에서 이전 명령이 흐름 변경을 야기했는지를 결정하는 단계;

상기 흐름 변경을 야기한 상기 이전 명령이 상기 캐시 메모리(26)를 활성화한 미리 결정된 변위의 역방향 브랜치인지를 결정하는 단계;

상기 다음 어드레스에 대응하는 다음 명령이 유효한 명령인지를 결정하는 단계; 및

상기 캐시 메모리(26)로부터 상기 다음 명령을 공급하는 단계를 포함하는 명령들 취급 방법.

청구항 4.

데이터 처리 시스템(20)에서 명령들을 취급하는 방법으로서, 상기 데이터 처리 시스템(20)은 중앙 처리 유닛(22), 메인 메모리(24) 및 캐시 메모리(26)를 포함하는, 상기 명령들 취급 방법에 있어서,

상기 캐시 메모리(26)를 무효화하는 단계;

상기 메인 메모리(24) 또는 상기 캐시 메모리(26) 중 어느 쪽에 액세스하기 위해 다음 어드레스를 계산하는 단계;

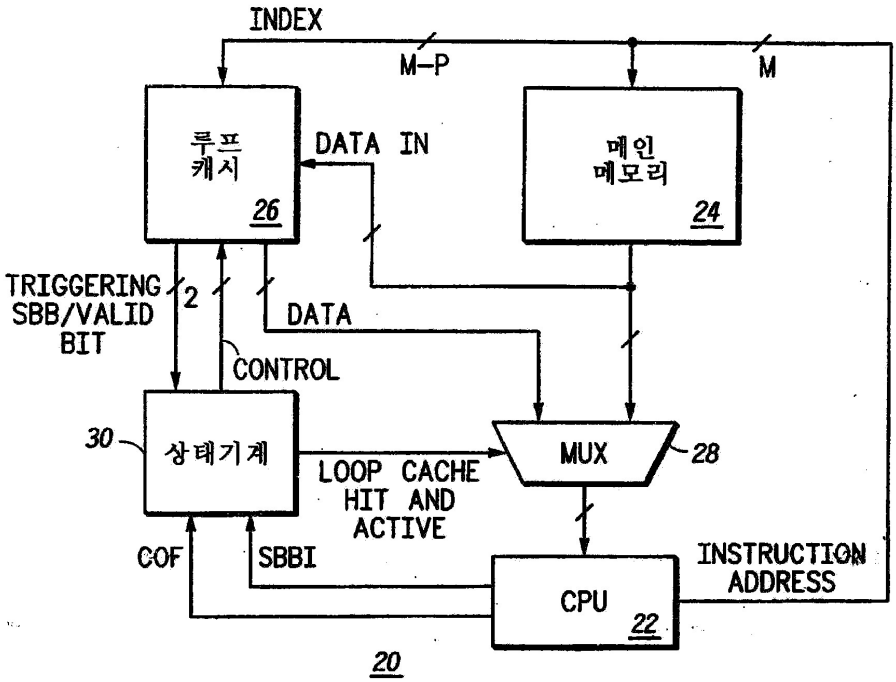
상기 캐시 메모리(26)가 비활성인지를 결정하는 단계;

상기 중앙 처리 유닛(22)에 의해 공급된 일련의 명령 어드레스들에서 흐름 변경을 야기하지 않은 이전 명령이 제공되었는지를 결정하는 단계; 및

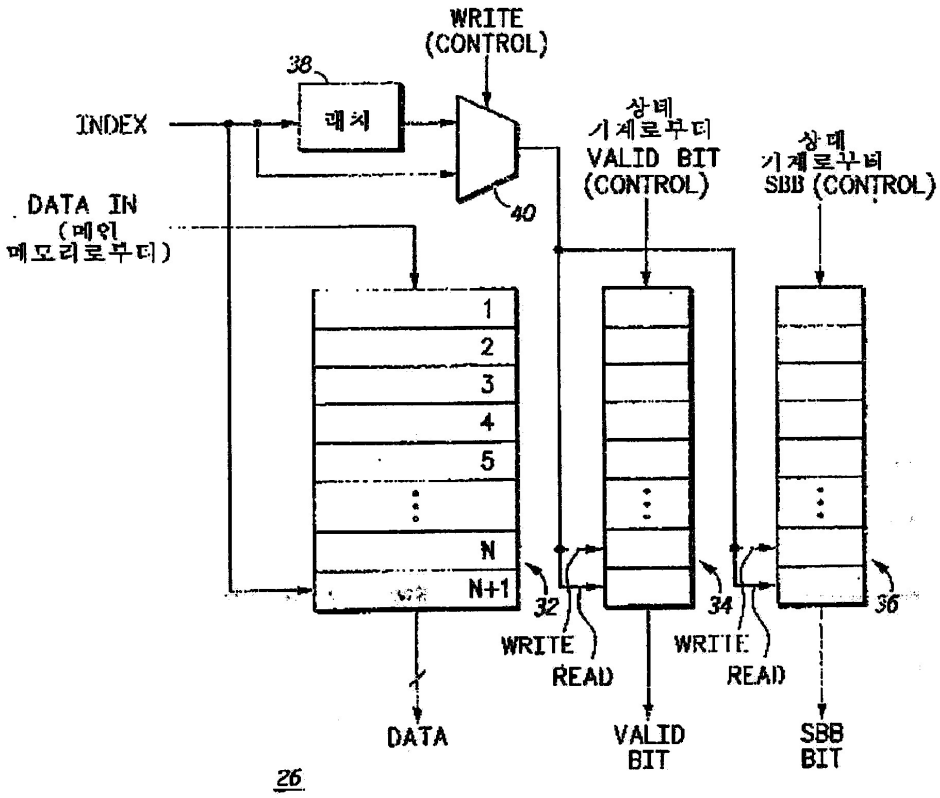
상기 메인 메모리(24)로부터 다음 명령을 상기 캐시 메모리(26)에 로드하고, 상기 다음 명령을 중앙 처리 유닛(22)에 공급하는 단계를 포함하는 명령들 취급 방법.

도면

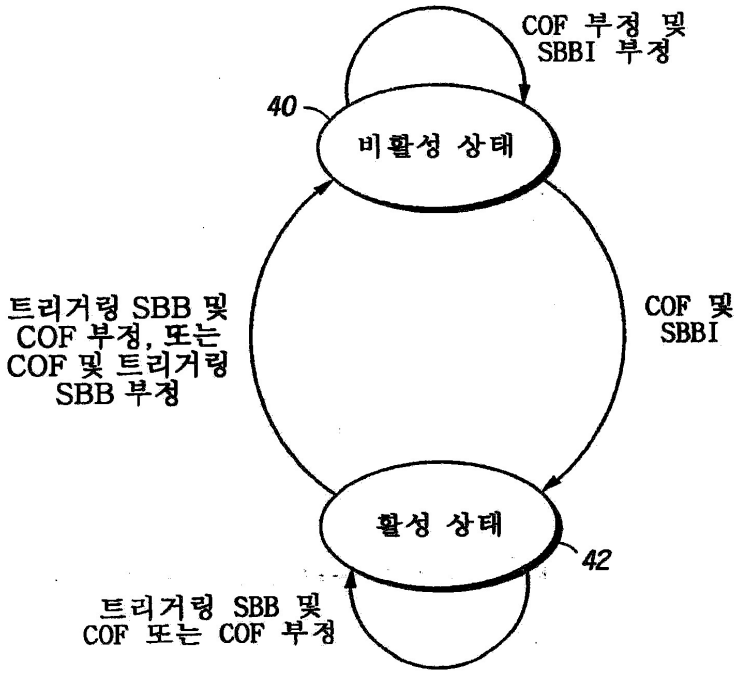
도면1



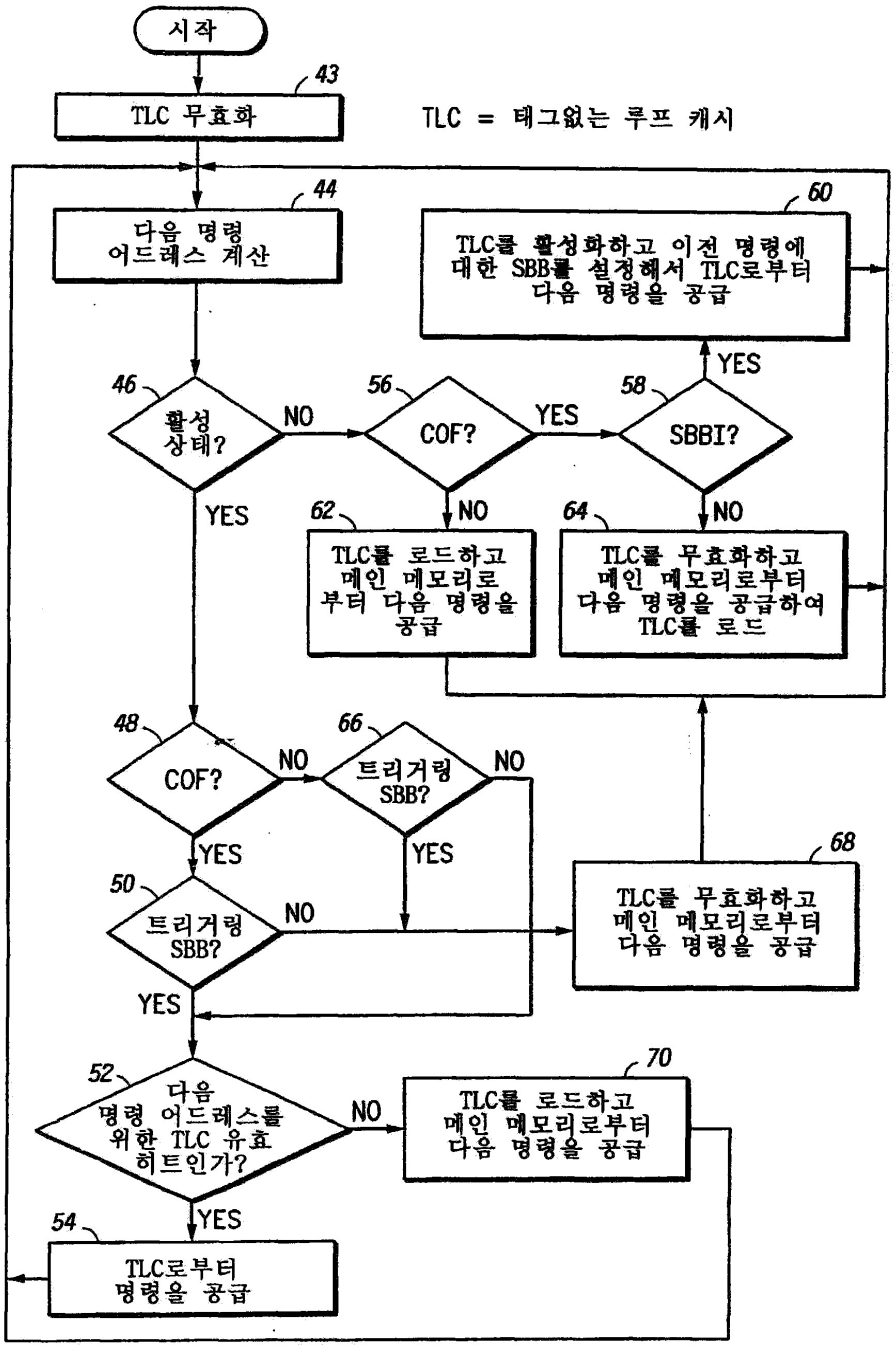
도면2


WRITE
(CONTROL)
38
래치
INDEX
DATA IN
(메인
메모리로부터)
40
상태
기계로부터
VALID BIT
(CONTROL)
상태
기계로부터
SBB (CONTROL)
1
2
3
4
5
N
N+1
32
34
36
WRITE
READ
WRITE
READ
DATA
VALID
BIT
SBB
BIT
26

도면3


COF 부정 및
SBBI 부정
40
비활성 상태
트리거링 SBB 및
COF 부정, 또는
COF 및 트리거링
SBB 부정
COF 및
SBBI
활성 상태
42
트리거링 SBB 및
COF 또는 COF 부정

도면4


시작
43
TLC 무효화
TLC = 태그없는 루프 캐시
44
다음 명령 어드레스 계산
60
TLC를 활성화하고 이전 명령에 대한 SBB를 설정해서 TLC로부터 다음 명령을 공급
46
활성 상태?
NO
56
COF?
YES
58
SBBI?
YES
YES
NO
NO
62
TLC를 로드하고 메인 메모리로부터 다음 명령을 공급
64
TLC를 무효화하고 메인 메모리로부터 다음 명령을 공급하여 TLC를 로드
48
COF?
NO
66
트리거링 SBB?
NO
YES
YES
50
트리거링 SBB?
NO
68
TLC를 무효화하고 메인 메모리로부터 다음 명령을 공급
YES
52
다음 명령 어드레스를 위한 TLC 유효 히트인가?
NO
70
TLC를 로드하고 메인 메모리로부터 다음 명령을 공급
YES
54
TLC로부터 명령을 공급