

(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

| | | |
|--|-------------------------------------|--|
| (51) 。 Int. Cl. ⁶ G06F 12/08 | (45) 공고일자 (11) 등록번호 (24) 등록일자 | 2005년05월19일 10-0470516 2005년01월28일 |
|--|-------------------------------------|--|

| | | | |
|-----------|-----------------|-----------|-----------------|
| (21) 출원번호 | 10-1997-0059183 | (65) 공개번호 | 10-1998-0042269 |
| (22) 출원일자 | 1997년11월11일 | (43) 공개일자 | 1998년08월17일 |

(30) 우선권주장 08/748,856 1996년11월14일 미국(US)

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

(72) 발명자 모에르 윌리엄 씨.
미국, 텍사스 78620, 트리핑 스프링즈, 피에르 브랜치 로드 1005

리 레황
미국, 텍사스 78759, 오스틴 줄리비레 로드 #189

아르엔즈 존
미국, 텍사스 78748, 오스틴 샤켈포드 드라이브 10709

(74) 대리인 이병호
 신현문
 이범래
 정상구

심사관 : 최봉목

(54) 분포된태그캐시메모리시스템및그것에데이터를저장하기위한시스템

요약

루프 캐시(26)는 메인 메모리(24) 액세스를 피하도록 CPU(22)에 명령들을 공급하기 위한 데이터 처리 시스템(20)에서 이용된다. 루프 캐시에 저장되는 명령들이 CPU에 공급될 수 있는 지는 CPU에 의해 계산되는 명령 어드레스와 관련된 분포된 TAG에 의해 결정된다. 명령 어드레스는 LCACHE 인덱스 부분(42), ITAG 부분(44), 및 GTAG(46)를 포함한다. LCACHE 인덱스(42)는 각각의 ITAG 어레이(50), 명령 어레이(52), 및 유효 비트 어레이(54)에서 대응하는 장소들을 선택한다. 저장된 GTAG 값(48)은 LCACHE 인덱스(42)가 가리키는 것과 관계없이 선택된다. 명령 어드레스(40)의 GTAG 부분은 저장된 GTAG 값(48)과 비교된다. 명령 어드레스(40)의 ITAG 부분은(44)은 ITAG 어레이(50)의 인덱스된 ITAG와 비교된다. 양 GTAG 및 ITAG가 바람직하게 비교되면, 명령은 메인 메모리에서 공급되기 보다는 루프 캐시에서 CPU에 공급된다.

대표도

도 1

명세서

도면의 간단한 설명

도 1은 본 발명에 따라 메모리 부분을 포함하는 데이터 처리 시스템을 블록 형태로 도시한 도면.

도 2는 또한 본 발명에 따라 도 1의 데이터 처리 시스템에 이용되는 메모리 시스템을 블록 형태로 도시한 도면.

도 3은 도 2에 도시된 메모리 시스템의 루프 캐시에서 데이터를 저장, 이용 및 대체하는 프로세스를 도시한 순서도.

도 4는 루프 캐시 엔트리의 대체를 제어하는 상태 기계를 도시한 도면.

도 5는 도3에 도시된 방법을 개선한 순서도.

도 6은 메모리에 저장된 명령 어드레스 값들의 일부를 도시한 도면으로서, 이 메모리에서 루프 명령들은 본 발명에 이용되는 GTAG 영역 경계를 교차하지만, 도 5의 프로세스에 따라서 루프 캐시를 재로딩하는 것은 바람직하지 않음.

도 7은 도 5에 도시되고 서술된 방법을 더욱 개선한 순서도.

도 8은 메모리에 저장된 명령 어드레스의 일부를 도시한 도면으로서, 이 메모리에서 제 1 루프 명령은 GTAG 영역 경계를 교차하지만, 다음 제 2 루프 명령은 단일 영역 내에 있고, 도 6의 프로세스에 따라 루프 캐시로 로딩될 만큼 충분히 작은 크기임.

*** 도면의 주요 부분에 대한 부호의 설명***

20 : 데이터 처리 시스템 22 : 중앙 처리 장치

24 : 메인 메모리 26 : 루프 캐시

28 : 멀티플렉서 30 : 상태 기계

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 일반적으로 메모리 시스템들에 관한 것이며, 특히 중앙 처리 유닛에 의해 실행될 명령들을 저장하기 위한 캐시 메모리 시스템들에 관한 것이다.

발명이 이루고자 하는 기술적 과제

저 전력 설계(low power design)는 포터블 및 핸드헬드(portable and handheld) 응용들의 광범위한 이용 때문에 마이크로프로세서 설계에서 상당히 중요해지고 있다. 많은 휴대 및 내장형 마이크로프로세서들은 메모리에 액세스하는데 상당한 양의 에너지를 소모한다. 전형적인 마이크로프로세서에서, 명령 기준(instruction reference)들에 의해 소모되는 전력은 데이터 기준들에 의해 소모되는 전력보다 훨씬 많다. 따라서, 이러한 응용들에서 명령을 꺼내는 에너지를 감소시키는 것이 유리하다.

페이징, 팩스, 및 디지털 신호 처리와 같은 응용들은 종종 작은 프로그램 루프들을 실행시키는데 상당한 시간을 소모한다. 이들 명령들 결과들을 위해서 메인 메모리에 반복적으로 액세스하면 전력 소모 및 실행 속도에 바람직하지 않은 영향을 준다. 그러므로, 메인 메모리에 액세스할 필요성을 피하므로써, 그런 작은 루프의 실행과 관련되는 실행 속도를 증가시키고 에너지를 감소시키는 것이 바람직하다.

캐시 TAG는 캐시의 수행 성능을 증가시키는데 종종 이용된다. 캐시 TAG는 마이크로프로세서에 의해 제공되는 TAG 어드레스를 수신하여 요청된 명령들 및/또는 데이터가 캐시 메모리에 존재하는 지를 결정한다. 요청된 명령이 캐시에 위치되지 않으면, 마이크로프로세서는 그 후 메인 메모리로부터 명령을 검색해야 한다. 명령이 캐시에 기록될 때, 명령 어드레스의 보다 높은 차수의 비트가 TAG 어레이에 저장된다. 캐시 TAG는 또한 프로세서 발생 어드레스와 TAG 어드레스를 비교하는 비교기를 가진다. TAG 어드레스 및 프로세서 발생 어드레스가 동일하면, 캐시 "히트(hit)"가 발생하고, 요청된 데이터가 캐시 메모리에 위치됨을 표시하는 정합 신호(match signal)가 캐시 TAG에 의해 제공된다. 프로세서 발생 어드레스 및 TAG 어드레스가 동일하지 않으면, 캐시 "미스(miss)"가 발생하고, 정합신호는 요청된 데이터가 캐시 메모리에 위치되지 않음을 표시한다. 게다가, 유효 비트(valid bit)는 캐시의 비교 사이클 동안 저장된 TAG 어드레스의 유효 히트에 자격을 부여하기 위해 TAG 어드레스의 부분으로 설정될 수 있다.

종래의 TAG 어레이에서, 캐시의 각각의 명령 엔트리는 대응하는 TAG 어레이 엔트리를 가지며, 각각의 TAG 엔트리는 동일한 크기가 된다. 따라서, 종래의 TAG 어레이의 크기는 특히 캐시 자체가 큰 경우, 매우 크게될 수 있다. TAG 어레이의 크기를 감소시키기 위해서는, 전형적으로 보다 작은 캐시를 사용해야 한다. 그러나 반복되는 명령 루프를, 저 전력을 소모하면서 고속으로 실행할 수 있도록 충분히 큰 캐시가 매우 바람직한, 많은 장치, 특히 내장된 제어 장치가 존재한다. 이들 동일한 응용들에서, 집적회로의 크기를 가능한 작게하는 것이 바람직하다. 그러므로, 캐시 크기를 감소시킴 없이 집적회로의 전체 크기를 최소화하면서 종래의 TAG 어레이로 성취되는 유사한 목적들을 성취하는 것이 바람직하다.

발명의 구성 및 작용

일반적으로, 본 발명은 데이터 처리 시스템의 메인 메모리를 액세스하지 않고, 반복적으로 수행될 복수의 명령을 저장하기 위해 중앙 처리 장치(CPU)에 국지적인 루프 캐시를 이용하는 캐시 메모리 시스템이다. 단일 글로벌 태그값은 루프 캐시의 각각의 엔트리에 관련되는 각각의 태그 부분과 관련하여, 루프 캐시의 히트 또는 미스 상태를 결정하는데 이용된다. 루프 캐시 위 무효화 엔트리는 글로벌 태그 및 각각의 태그의 비교 논리와, 플로우(flow) 상태 변화를 검출함에 의해 제어된다. 상태 기계(state machine)는 루프 캐시 엔트리들에 대한 대체 동작(replacement operation)을 돕는데 이용된다. 본 발명은 도 1-8 및 이하의 텍스트를 참조하여 더 확실하게 이해할 수 있을 것이다.

도 1은 본 발명에 따른 데이터 처리 시스템(20)을 도시한 블록도이다. 데이터 처리 시스템(20)은 중앙 처리 장치(CPU)(22), 메인 메모리(24), 루프 캐시(26), 멀티플렉서(28) 및 상태 기계(30)를 포함한다. 일반적으로, CPU(22)는 자체에서 연속적으로 실행될 명령들에 대한 복수의 명령 어드레스들을 발생한다. 명령 어드레스들은 메인 메모리(24) 및 루프 캐시(26)에 제공된다. 각각의 명령 어드레스는 M 비트를 포함한다. 모든 M 비트들은 루프 캐시(26) 및 메인 메모리(24)에 제공된다. 명령 어드레스에 응답하여, 메인 메모리(24) 또는 루프 캐시(26)는 명령 어드레스에 대응하는 명령을 멀티플렉서(28)를 통해서 CPU에 제공한다. 루프 캐시(26)와 관련된 논리와 관련하는 상태 기계(30)는 메인 메모리(24) 또는 루프 캐시(26) 중 어느 것이 CPU(22)에 명령들을 제공하는지를 제어하기 위해 사용된다. 다른 실시예들에서, 루프 캐시(26)는 예를 들어 CPU(22) 대신에 대안적인 데이터 처리 또는 저장 유닛에 명령을 공급할 수 있다는 것을 주목하자.

상태 기계(30)는 "COF(change-of-flow)"로 라벨된 제어 신호 및 "SBBI(short backward branch instruction)"로 라벨된 제어 신호를 수신한다. 루프 캐시(26)는 세 개의 제어 비트들을 상태 기계(30)에 제공한다. 제어 비트들 중 하나는 "GTAG HIT"하고 라벨되어 있고, 다른 하나는 "ITAG HIT"로 라벨되고 또다른 비트는 "VAILD BIT"라고 라벨되어 있다. 도시된 바와 같이, "LOOP CACHE HIT" 신호는 상태 기계(30)에 의해 멀티플렉서(28)에 공급된다. 후술된 바와 같이, LOOP CACHE HIT는 GTAG HIT, ITAG HIT 및 VAILD BIT의 함수이다. LOOP CHAHE HIT가 어서트(assert)될 때, 루프 캐시(26)는 활성화되고, 데이터는 루프 캐시(26)로부터 단독으로 멀티플렉서(28)를 통해 CPU(22)에 공급된다. LOOP CACHE HIT가 부정(negate)될 때, 루프 캐시(26)는 비활성화되고 데이터는 대신에 멀티플렉서(28)를 통해서 메인 메모리(24)로부터 CPU(22)로 제공된다. 명령과 같은 데이터가 명령은 메인 메모리(24)에서 CPU(22)로 공급되고 있는 동시에, 그런 데이터는 메인 메모리(24)에서 루프 캐시(26)로 제공될 수 있다. 루프 캐시(26)가 기능하는 방식은 도 2-8을 참조하여 후술될 것이다.

루프 캐시(26)는 도 2에 상세하게 도시되어 있다. 다음 명령 어드레스는 CPU(22)에서 명령 어드레스(40)로 수신된다. 명령 어드레스(40)는 루프 캐시(LCACHE) 인덱스 부분(42), 각각의 태그(TAG) 부분(44), 및 글로벌 태그(GTAG) 부분(46)을 포함한다. LCACHE 인덱스(42)는 ITAG 어레이(50) 내, 명령 어레이(5) 내, 및 유효 비트 어레이(54) 내의 위치를 어드레스하기 위하여 사용된다. ITAG 부분(44)은 LCACHE 인덱스(42)에 의해 선택되는 ITAG 어레이(50)의 엔트리로 로딩된다. 명령 어드레스의 ITAG 부분은 또한 명령 어드레스의 ITAG 부분의 값을 LCACHE 인덱스(42)에 의해 선택되는 ITAG 어레이(50)의 엔트리에 저장되는 ITAG 값과 비교하는 비교기(62)에 결합된다. GTAG 부분(46)은 저장된 GTAG 값(48)으로서 GTAG 값을 로딩하도록 사용된다. GTAG 부분(46)은 또한 명령 인덱스의 GTAG 부분(46)을 저장된 GTAG 값(48)과 비교하는 비교기(60)에 결합된다.

무효 비트 어레이(54)는 명령 어레이(52)의 엔트리들과 관련되는 복수의 엔트리들을 포함한다. 유효 비트 어레이는 상태 기계(30)에서 갱신되거나 유지된다. 새로운 엔트리가 명령 어레이에 로딩될 때마다, 그와 관련된 유효 비트는 1과 동일하게 설정된다. ITAG 어레이(50)는 또한 각각의 명령 엔트리 자체의 특정 ITAG 엔트리를 갖도록 명령 어레이(52)의 엔트리들과 관련하는 복수의 엔트리들을 포함한다. ITAG 엔트리는 종래의 태그 어레이들에서처럼 각각의 명령의 "태그"하는 작용을 한다. 그러나, 본 발명은 명령 어드레스와 동일한 크기인 태그 부분없이 태그 기능을 수행한다. 따라서, 태그 어레이는 종래의 태그 어레이들보다 상당히 작은 크기이다. 본 발명에 따라서, 다수의 엔트리들과 공통되는 단일 글로벌 태그값을 사용하면 보다 작은 ITAG 어레이를 사용할 수 있게 된다.

루프 캐시의 엔트리들의 수는 본 발명에 의해 제한되지 않는다는 것을 알 수 있다. 최적의 크기는 데이터 처리 시스템용 특정 응용에 의해 관리된다(예를 들어, 이 응용에 대한 명령 루프의 평균 크기에 따름). 더구나, 각각의 ITAG 어레이 엔트리에서 바이트들의 수는 본 발명에 의해 제한되지 않는다. 바이트 수가 많으면 많을 수록, 어레이의 물리적인 크기는 크게 된다. 그러나, 바이트들의 수가 적으면 적을 수록, 루프 캐시 명령은 사용되지 않고, 메인 메모리가 액세스될 것이다. ITAG 어레이의 엔트리들에 다수의 바이트들을 결정하는 요인들은 데이터 처리 시스템에서 실행되는 프로그램 형태에 의존할 것이다. ITAG 어레이의 구성의 하나의 예는 다음과 같다. 32 비트들의 명령 어드레스, 16 엔트리, 단일 명령 루프 캐시의 경우에, 어드레스의 비트 0는 관계없는 반면, 비트 1 내지 비트 4(전체 4비트에 대해) LCACHE 인덱스 부분(42)으로 사용되고, 비트 5 및 비트 6(전체 2 비트에 대해)은 ITAG 부분(44)에 대해 남아있는 비트 6-31(전체 25 비트에 대해)는 전체 태그 부분(46)에 대해 이용된다.

캐시 히트 논리(64)는 비교기(60), 비교기(62) 및 유효 비트 어레이(54)에 결합된다. 캐시 히트 논리(64)는, 1)명령 어드레스(40)의 GTAG 부분(46)이 저장된 GTAG 값(48)과 정합하는 경우, 2)명령 어드레스(40)의 ITAG 부분(44)이 루프 캐시 인덱스(42)에 의해 선택되는 ITAG 부분 엔트리와 정합하는 경우 그리고 3)LCACHE 인덱스(42)에 의해 또한 선택되는 유효 비트 어레이(54)의 유효 비트가 어서트되는 경우에 캐시 히트를 표시할 것이다. 양호한 실시예에서, 비교기(60)는 또한 명령 어드레스의 GTAG 부분(46)과 저장된 GTAG 값(48)을 비교한 결과를 표시하는 GTAG 히트 신호를 제공한다.

도 3은 본 발명의 하나의 실시예를 따라서 루프 캐시(26)와 관련되는 명령을 기억, 사용 및 대체하는 프로세스(70)를 순서도 형태로 도시한 것이다. 프로세스의 제 1 단계, 즉 단계(72)에서, 유효 비트 어레이의 모든 엔트리들은 무효화되거나 제로로 설정되고, "REPLACE"로 불리는 상태 변수는 1과 동일하게 설정된다. 상태 변수 REPLACE는 메인 메모리로부터의 명령이 루프 캐시로 로딩되는지를 결정하므로써, 기존의 엔트리를 대체한다. 다음 단계(74)에서, CPU는 도 2에 도시된 다음 명령 어드레스를 명령 어드레스(40)로서 계산한다. 단계(76)에서, GTAG 히트가 있는지, 다시 말하면 명령 어드레스(40)의 GTAG 부분(46)이 저장된 GTAG 값(48)과 정합하는 지에 대한 결정이 행해진다. GTAG 히트가 없는 경우, 루프 캐시는 단계(78)에서 무효화되고 저장된 GTAG 값은 명령 어드레스의 GTAG 부분(46)의 값으로 대체된다. 도 3의 단계(78)에서, 이것은 "GTAG를 재로딩"으로서 표시된다. 또한, 단계(78) 내에서, 상태 변수 REPLACE는 다시 1과 동일하게 설정된다. GTAG가 존재하지 않았기 때문에, 인덱스된 루프 캐시 내에 저장된 명령이 사용될 수 없고 명령은 메인 메모리로부터 꺼내야 한다. 이것은 단계(80)로서 프로세스(70)로서 표시된다. 결정 단계(82)에서, REPLACE가 1과 동일한 지 또

는 엔트리가 무효한 지가 다음에 결정된다(즉, 명령 어드레스(40)의 LCACHE 인덱스(42)에 의해 인덱스된 무효 비트는 부정되거나 제로로 설정된다). 이들 두 개의 조건들 중 어느 하나가 발생하면, LCACHE 인덱스(42)에 의해 선택된 명령 어레이 엔트리는 단계(84)에서 메인 메모리로부터 수신되는 명령과 함께 로딩되고 엔트리는 무효화된다(즉, 엔트리와 관련된 무효 비트는 1과 동일하게 설정된다). 그 후 동일한 명령이 단계(86)에서 CPU에 공급된다. 그 후 다음 명령 어드레스는 단계(74)에서 계산되고 수신되며, 프로세스(70)는 계속된다.

결정 단계(76)에서, 그 결과가 GTAG 히트이면, 프로세스(70)에서 다음 단계는 플로우 변경(change-of-flow:COF)인지를 결정하는 결정 단계(88)이다. COF 신호는 CPU에 의해 어서트되며, 명령 어드레스는 바로 앞에 수신된 명령 어드레스를 순서대로 따르지 않는다. 따라서, 플로우 변경은 이용되는 분기와 유사하다. 플로우 변경이 존재하지 않으면, 다음 단계는 결정 단계(90)로 표시된 바와 같은, 루프 캐시 히트가 존재하는 지를 결정하는 것이다. 상술된 바와 같이, 명령 어드레스의 GTAG 및 ITAG 부분 둘 다가 기억된 GTAG 값 및 선택된 ITAG 어레이 엔트리 각각에 정합될 때, 그리고 유효 비트 어레이의 선택된 엔트리가 유효하다는 것을 보여줄 때, 캐시 히트 논리(64)로부터 출력되는 루프 캐시 히트 신호이다. 이들 조건이 부합하는 경우, 루프 캐시 히트가 존재하고 명령 어드레스(40)의 LCACHE 인덱스(42)에 의해 선택된 명령 어레이의 엔트리 내에 저장된 명령이 단계(92)에서 CPU로 공급된다. 명령을 위해, 메인 메모리에 액세스할 필요가 없으므로 전력 소모 및 실행 속도를 절약한다. 그 후 다음 명령 어드레스가 단계(74)에서 CPU에 의해 계산되고, 프로세스(70)는 반복된다.

대신, 단계(88)에서, 플로우 변경이 존재하는 것으로 결정하면, 이 몇가지가 결정될 필요가 없다. 첫 번째, 상태 변수 REPLACE가 1 또는 0으로 설정되어야 하는지를 결정함으로써 명령 어드레스(40)의 LCACHE 인덱스(42)에 의해 선택된 명령 어레이의 엔트리가 메인 메모리로부터 새로운 명령과 함께 로딩되는지의 여부에 영향을 미친다. 또한, 명령 어레이의 선택된 엔트리가 CPU에 공급되는지가 결정되어야 한다. 이들 결정은 다음과 같은 프로세스(70)에서 이루어진다. 단계(88)에서 플로우 변경이 존재하면, 결정 단계(94)에서 루프 캐시 히트가 존재하는 지가 다음에 결정된다. 단계(94)는 이전에 상술된 바와 같은 단계(90)와 유사하다. 루프 캐시 히트가 존재하면, 상태 변수 REPLACE는 단계(96)에서 0으로 설정되는데, 이것은 명령 엔트리의 선택 엔트리가 메인 메모리로부터 나오는 새로운 명령으로 대체되지 않는다는 것을 의미한다. 대신, 명령 어레이의 선택된 엔트리에 저장된 명령은 단계(92)에서 CPU에 공급된다. 그 후 다음 명령 어드레스가 단계(74)에서 CPU에 의해 계산되고, 프로세스(70)는 계속된다.

단계(88)에서 플로우 변경이 존재하지만, 루프 캐시 히트가 존재하지 않는다면, 그 후 상태 변수 REPLACE는 단계(98)에서 1로 설정된다. 루프 캐시 히트가 존재하지 않기 때문에, 명령은 단계(80)에서 메인 메모리로부터 꺼내져야 한다. 단계(82)에서, 그 후 REPLACE가 1과 동일한 지 또는 엔트리가 무효한 지가 결정된다. REPLACE는 단계(98)에서 1과 동일하게 설정되었기 때문에, 단계(82)의 결되는 "예"가 될 것이고, 메인 메모리로부터 꺼내지는 명령은 단계(84)에서 명령 어드레스에 의해 인덱스된 명령 어레이의 엔트리로 로딩될 것이다. 그 후 메인 메모리에서 어드레스로부터의 명령은 단계(86)에서 CPU에 제공되고, 다음 명령 어드레스는 단계(74)에서 CPU에 의해 계산된다. 그 후, 프로세스(70)는 반복된다. 프로세스(70)는 CPO가 명령 어드레스들을 공급하는 한 계속된다. 이들 명령 어드레스들은 CPU가 명령을 실행 중인 동안 CPU에 의해 공급될 것이다.

프로세스(70)에 따라서, 캐시된 루프는 CPU에 반복된 명령들의 루프를 공급함으로써 이들 명령들의 반복된 실행 동안 메인 메모리를 바이패스한다. 루프 캐시에 저장된 명령들이 CPU가 요청한 명령들이 지가 루프 캐시의 각각의 엔트리에 특정한 작은 각각의 태그 및 다수의 엔트리에 공통되는 글로벌 태그를 이용함으로써 결정된다. 단일 글로벌 태그값 또는 필드로 명령 어드레스의 최상위 비트(MSB)들을 표시함으로써, 태그 어레이의 크기는 로컬 캐시의 모든 다른 이점을 성취하는 동안 매우 작게 유지된다.

도 4는 REPLACE 상태(102) 및 FREEZE 상태(104)를 갖는 상태 기계(100)를 도시한 것이다. REPLACE 상태(102)는 REPLACE 비트가 1과 동일하게 설정될 때, 유사한 반면, FREEZE 상태는 REPLACE가 0과 동일할 때를 표시한다. 상태 기계에 의해 표시된 바와 같이, 루프 캐시 히트 및 플로우 변경이 존재할 때, 상태 변수는 REPLACE에서 FREEZE로 변경된다. FREEZE 상태에서, 상태 변수는 두 가지 조건, 즉 1)루프 캐시가 무효화되거나 2)GTAG 히트, 플로우 변경 및 루프 캐시 미스가 존재하는 조건들 중 하나의 조건이 발생할 때, REPLACE로 변경된다.

도 5는 또다른 실시예를 따른 루프 캐시를 이용하는 프로세스(110)를 또다른 순서도로 도시한 것이다. 프로세스(110)는 도 3의 프로세스(70)를 참조하여 앞서 설명된 바와 같은 많은 동일한 단계 및 플로우를 포함한다. 따라서, 공통적인 단계들 및 유사한 플로우에 관한 설명은 프로세스(110)를 설명한 것을 참조하여 생략될 것이다. 프로세스(110)는 결정 단계(76)의 결과가 "아니오"일 때(즉, GTAG 히트가 존재하지 않을 때) 프로세스 플로우(70)와 상이하다. 도 5에 도시된 바와 같이, 단계(76)에서 GTAG 히트가 존재하지 않으면, 프로세스(110)의 다음 단계(112)는 플로우 변경이 존재하는 지가 결정된다. 단계(112)는 상술된 단계(88)와 유사하다. CPU로부터 수신되는 명령 어드레스가 바로 선행하는 명령 어드레스와 비교하여 플로우 변경되었다면, 루프 캐시는 단계(78)에서 무효화된다. 저장된 GTAG 값은 리셋되고, 상태 변수 REPLACE는 1과 동일하게 설정된다. 플로우 변경과 함께 GTAG 미스가 존재하기 때문에, 명령은 단계(80)에서 메인 메모리로부터 꺼내져 설정되어야 한다. 명령은 단계(84)에서 루프 캐시의 선택된 엔트리로 로딩되는데, 왜냐하면, REPLACE가 결정 단계(82)에서 1과 동일하게 되기 때문이다. 그 후, 메인 메모리로부터 꺼내지는 명령은 CPU에 공급되고, 다음 명령 어드레스가 계산된다.

대신에, 앞서 수신된 명령 어드레스로부터 플로우 변경 없이, GTAG 미스가 존재하면, 선택된 루프 캐시 엔트리 내에 존재하는 명령이라기 보다, 명령은 메인 메모리로부터 꺼내지게, 이 명령은 CPU에 공급된다.

부가한 단계(112)의 이점은 도 6을 참조로 훨씬 잘 이해할 수 있을 것이다. 도 6은 특정 GTAG 영역과 관계되는 메모리에서 어드레스 일부를 표시한다. GTAG 영역은 동일하게 저장된 GTAG 값에 대응하는 메모리 영역이다. 도 6에 도시된 바와 같이, CPU에 의해 실행되는 특정 루프는 두 개의 서로 다른 GTAG 영역을 교차하는 어드레스들을 포함한다. 그럼에도 불구하고, 이들 명령들을 메인 메모리로부터 공급하는 것과는 대조적으로, 루프 캐시에서 그 루프의 적어도 일부만으로부터 명령을 실행하는 것이 유리할 수 있다. 따라서, 프로세스 플로우(110)의 수행에 따라서, GTAG 미스가 존재하지만 사전 계산된 명령 어드레스로부터 플로우 변경이 없다면, 명령은 GTAG 히트 또는 플로우 변경 중 어느 하나가 존재할 때까지 계속적으로 메인 메모리로부터 공급된다. 따라서, 루프 내의 명령의 적어도 일부는 에너지 보존 및 속도 효율성을 위해 루프 캐시로부터 여전히 공급될 것이다.

도 7은 루프 캐시를 활용하기 위해 본 발명을 따른 프로세스 플로우(120)를 도시한 것이다. 게다가, 많은 단계들 및 플로우들은 도 3의 플로우(70)를 참조하여 상술된 단계 및 플로우들과 유사하며, 따라서, 반복적으로 설명되지 않을 것이다. 프로세스(110)와 같이, 프로세스(120)는 GTAG 미스가 존재할 때 다시 말하면, 결정 단계(76)의 결과가 "아니오"일 때 수행되는 단계들에 관련하여 프로세스(70)와 다르다. 단계(76)에서 GTAG 미스가 존재하면, 단계(112)는 상술된 바와 같이 수행되어 플로우 변경이 존재하는지를 결정한다. 플로우 변경이 존재하지 않으면, 명령은 단계(114)에서 메인 메모리로부터 꺼내진다. 이 명령은 단계(86)에서 메인 메모리로부터 CPU에 공급되고 CPU는 단계(74)에서 다음 명령 어드레스를 계산한다. 플로우 변경이 존재하지 않을 때, 루프 캐시의 엔트리를 무효화하는 메인 메모리로 가는 것의 이점이 프로세스(110)를 참조하여 서술된 것과 동일하다.

프로세스(120)에 의해 개선했던 것은 단계(112)에서 플로우 변경이 존재하는 경우 발생한다. 결정 단계(112)의 결과가 "예"라면, 또다른 결정 단계(112)는 플로우 변경이 짧은 역방향 분기 명령(SBBI)의 결과인지를 결정하기 위해 사용된다. SBBI 신호는 분기 명령이 실행될 때, CPU에 의해 어서트되는데, 이것은 사전 명령으로부터 미리 결정된 배치 내에서 명령을 다시 분기시킨다. SBBI 신호를 생성하는 목적은 특정 명령 루프가 루프 캐시 내에서 적합할 정도로 충분히 작은 크기인지를 표시하는 것이다. 결정 단계(122)에서, 플로우 변경이 SBBI가 아니면, 명령은 단계(114)에서 메인 메모리로부터 꺼내지고, 상기 명령은 플로우 변경이 없었던 것처럼 메인 메모리에서 CPU로 공급된다. 한편, 플로우 변경이 SBBI라면, 루프 캐시는 단계(78)에서 무효화된다. 이 동일한 단계에서, 새로운 GTAG 값은 로딩되고, 상태 변수 REPLACE는 1과 동일하게 설정된다. 그 후, 명령은 단계(80)에서 메인 메모리로부터 꺼내지고, 루프 캐시에는 단계(84)에서 이 명령이 로딩되는데, 왜냐하면, REPLACE가 1과 동일하기 때문이다. 그 후, 명령은 메인 메모리에서 CPU로 공급되고, CPU는 (74)에서 다음 명령 어드레스를 계산하고, 프로세스(120)은 계속된다.

도 6과 같이, 플로우 변경이 도 8에 표시될 때, 프로세스(120)에서 결정 단계(122)를 추가하는 이점은 두 개의 다른 GTAG 영역들을 표시한다. CPU가 명령 루프(1)와 관련되는 명령 어드레스들을 계산하면, 제 1 GTAG 영역을 제 2 GTAG 영역과 교차시, 단계(76)의 결과는 "아니오"가 될 것이다(즉, GTAG 미스가 존재). 명령 루프(1) 내에서, 플로우 변경은 존재하지 않고 제 2 GTAG 영역 내에 존재하는 명령 루프(1) 부분은 메인 메모리로부터 계속해서 공급되는 반면, 제 1 GTAG 영역 내에 있는 명령 루프(1)의 부분은 도 6을 참조하여 상술된 바와 같이, 루프 캐시에 의해 공급될 것이다. 결정 단계(122)를 추가함으로써, 루프 캐시 내에서 새로운 명령 루프를 포착할 수 있다. 예를 들어, 도 8에서, 명령 루프(2)가 실행되는데, 이 루프 내에서 루프(2)로부터 최종 명령을 실행할 때, 최종 루프 명령으로부터 소정의 범위 내에 있는 역분기 또는 플로우 변경이 존재한다. 그런 예에서, 루프 캐시 엔트리를 무효화하는 것이 바람직하고, 캐시 메모리 내의 명령 루프(10)의 일부분만을 유지하는 것보다 명령 루프(2)와 관련되는 명령으로 이를 재로딩하는 것이 바람직하다. 다시 말하면, 루프 캐시는 명령들의 새로운 루프가 CPU에 의해 실행되고, 루프의 남은 일부분을 공급하도록 메인 메모리를 의지하고 루프 캐시에 루프의 일부분만을 저장하는 것보다는 루프 캐시 명령 어레이에 명령들의 이 루프를 저장하는 것이 더욱 효과적일 수 있다는 사실에서 픽업한다(pick up). 도 8에서, 명령 루프(1) 및 명령 루프(2)는 도 8을 참조하여 서술된 프로세스를 수행하는 이점을 성취하는 것이 필요하지 않는 것과 같은 명령 어드레스를 중첩하는 것이 도시된다.

앞서 말한 것으로부터, 이전에 설명에 필요성을 이행하는 동일한 것에 데이터를 저장하기 위한 방법으로 분산된 태그 캐시 메모리 시스템이 제공되는 것은 명백하다. 특히, ITAG 값(이것은 명령 어드레스의 캐시 인덱스에 의해 선택됨)과 관련하여 저장된 글로벌 태그값(이것은 명령 어드레스 캐시 인덱스로부터 선택되지 않음)을 이용하는 것은 명령을 CPU에 공급하기 위한 루프 캐시를 이용하는 수단을 제공하는 것이다. 그런 루프 캐시를 이용하는 것은 메인 메모리의 액세스들 수를 감소시키므로써 명령을 꺼내 소모되는 전력을 감소시킨다. 본 출원에서 나타내는 바와 같이, 루프 캐시를 이용하는 것은 특히 작은 명령 루프의 실행에 크게 의존하는 응용에 유용하다. 그런 응용은 디지털 신호 처리, 페이징 및 팩스 장치들을 포함한다. 본 발명에 따라, 루프 캐시를 이용하는 것은 단일 글로벌 태그값이 다수의 루프 캐시 엔트리들을 위해 이용되기 때문에 최소 영역과 관련하여 성취된다.

본 발명의 양호한 실시예 또는 실시예들의 관점에서 설명되었지만, 본 발명은 위에서 설명한 것 이외에 실시예들을 취하고 변경될 수 있음이 당업자에게 명백할 것이다. 예를 들어, 본 발명의 메인 메모리는 L2 캐시 또는 심지어 외부 메모리와 같은, 루프 캐시보다 높은 레벨에서 임의의 메모리 어레이일 수 있다. 게다가, 본 발명은 루프 캐시 어레이 내의 임의의 특정 엔트리 수 또는 바이트 수로 제한되지 않으며, 본 발명은 단일 글로벌 태그값 또는 필드의 사용에 제한되지 않는다. 소수의 글로벌 태그값은 여기에 상술된 이점들을 얻을 수 있는 동시에 저장될 수 있다. 첨부된 청구범위는 발명의 정신 및 범위 내에 속하는 모든 변경들을 포괄하도록 의도된 것이다.

(57) 청구의 범위

청구항 1.

캐시 메모리 시스템에 있어서,

복수의 엔트리들을 가진 캐시 어레이로서, 각각의 엔트리는 각각의 태그 부분 및 데이터 부분을 가진, 상기 캐시 어레이와;

단일 글로벌 태그값만을 저장하기 위한 저장 수단과;

상기 캐시 어레이 및 상기 저장 수단에 결합되는 명령 어드레스 입력으로서, 상기 명령 어드레스 입력은 인덱스 부분 및 태그 부분을 가지며, 상기 인덱스 부분은 상기 글로벌 태그값을 식별함 없이 상기 캐시 어레이의 상기 복수의 엔트리들 중 하나를 식별하고, 상기 태그 부분은 각각의 태그 부분 및 글로벌 태그 부분으로 분할되는, 상기 명령 어드레스 입력과;

상기 저장 수단, 상기 캐시 어레이, 및 상기 명령 어드레스 입력에 결합되고, 식별된 캐시 어레이 엔트리의 각각의 태그 부분과 상기 명령 어드레스 입력의 각각의 태그 부분을 비교하고, 캐시 어레이로 하여금 상기 글로벌 태그값과 상기 명령 어드레스 입력의 상기 글로벌 태그 부분을 비교하기 위한 상기 비교 수단과;

상기 비교 수단 및 캐시 어레이에 결합되고, 캐시어레이로 하여금 상기 글로벌 태그값과 상기 글로벌 태그 부분의 긍정적인 비교(favorable comparison)와 중앙 처리 유닛으로부터의 플로우 신호의 변경(change-of-flow signal) 둘 다를 검출하는 것에 응답하여 반복적으로 실행될 복수의 명령들로 다시 로딩되게하기 위한, 상태 기계(state machine)를 포함하는, 캐시 메모리 시스템.

청구항 2.

데이터 처리 및 메모리 시스템에 있어서,

명령 어드레스들을 발생하고, 명령들을 처리하기 위한 중앙 유닛으로서, 상기 명령들 중 하나는 대응하는 상기 명령 어드레스들 중 하나에 대응하는 중앙 처리 유닛과;

상기 중앙 처리 유닛에 결합되고, 데이터를 공급 및 수신하기 위한 메인 메모리와;

상기 중앙 처리 유닛과 상기 메인 메모리에 결합되고, 상기 명령들을 수신 및 공급하기 위한 캐시 메모리와;

상기 중앙 처리 유닛과 캐시 메모리에 결합되고, 상기 캐시 메모리로 하여금 상기 중앙 처리 유닛에 의해 반복적으로 실행될 명령 루프를 포함하는 복수의 명령들로 로딩되게 하는 상태 기계로서, 상기 복수의 명령들은 상기 중앙 처리 유닛에 의해 실행되는 하나의 명령이 짧은 백워드 분기 명령임을 검출하는 것에 응답하여 로딩되고, 상기 짧은 백워드 분기 명령은 상기 명령 루프가 상기 캐시 메모리 내에서 적합할 것임을 표시하기 위한 것인, 상기 상태 기계를 포함하는, 데이터 처리 및 메모리 시스템.

청구항 3.

캐시 메모리 시스템에 데이터를 저장하기 위한 방법에 있어서,

복수의 엔트리들을 가진 캐시 어레이를 제공하는 단계로서, 각각의 엔트리는 각각의 태그 부분 및 데이터 부분을 가진, 상기 복수의 엔트리를 가진 캐시 어레이를 제공하는 단계와;

인덱스 부분, 각각의 태그 부분, 및 글로벌 태그 부분을 가진 명령 어드레스를 수신하는 단계와;

상기 명령 어드레스의 인덱스 부분에 기초하여 상기 복수의 캐시 어레이의 엔트리들 중에서 엔트리를 선택하는 단계와;

메모리에 글로벌 태그값을 저장하는 단계로서, 상기 글로벌 태그값은 상기 명령 어드레스의 상기 인덱스 부분과 독립적으로 저장되는, 상기 메모리에 글로벌 태그값을 저장하는 단계와;

상기 명령 어드레스의 상기 글로벌 태그 부분과 메모리에 저장된 상기 글로벌 태그값을 비교하는 단계와;

상기 명령 어드레스의 상기 각각의 태그 부분과 상기 캐시 어레이의 선택된 엔트리의 각각의 태그 부분을 비교하는 단계와;

상기 선택된 엔트리가 유효한 지를 결정하는 단계와;

캐시 히트를 결정하는 단계로서,

상기 명령 어드레스의 상기 글로벌 태그 부분과 메모리에 저장된 글로벌 태그값을 비교하는 단계와, 상기 명령 어드레스의 상기 각각의 태그 부분과 상기 캐시 어레이의 선택된 엔트리의 상기 각각의 태그 부분을 비교하는 단계, 및 상기 선택된 엔트리가 유효한 지를 결정하는 단계에 응답하여, 캐시 히트를 결정하는 단계와;

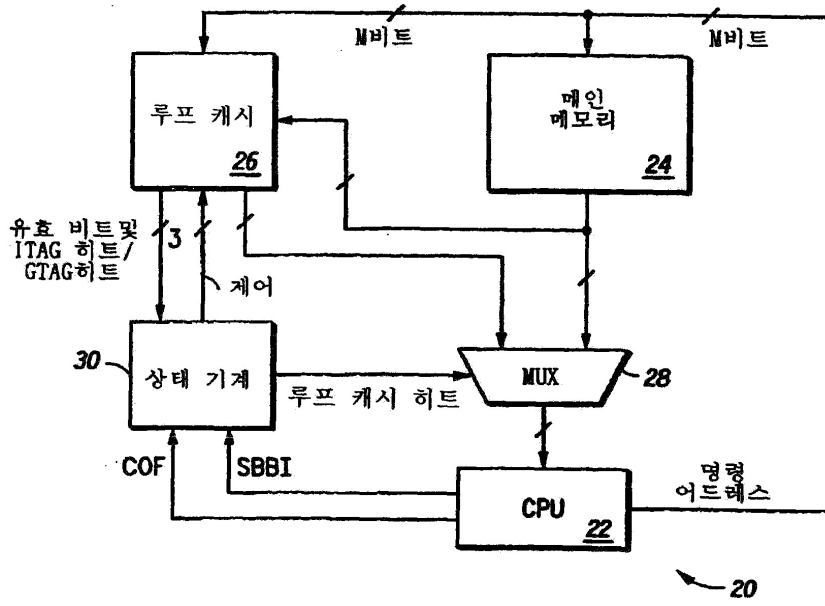
상기 캐시 히트를 결정하는 단계에 응답하여 상기 선택된 엔트리의 데이터 부분을 선택적으로 제공하는 단계와;

명령들의 시퀀스에서의 명령이 상기 검출된 명령으로부터의 미리 결정된 변위(displacement) 내의 짧은 백워드 분기로 하여금 반복적으로 실행되는 명령 루프를 정의하게 하는 것임을 검출하는 단계와;

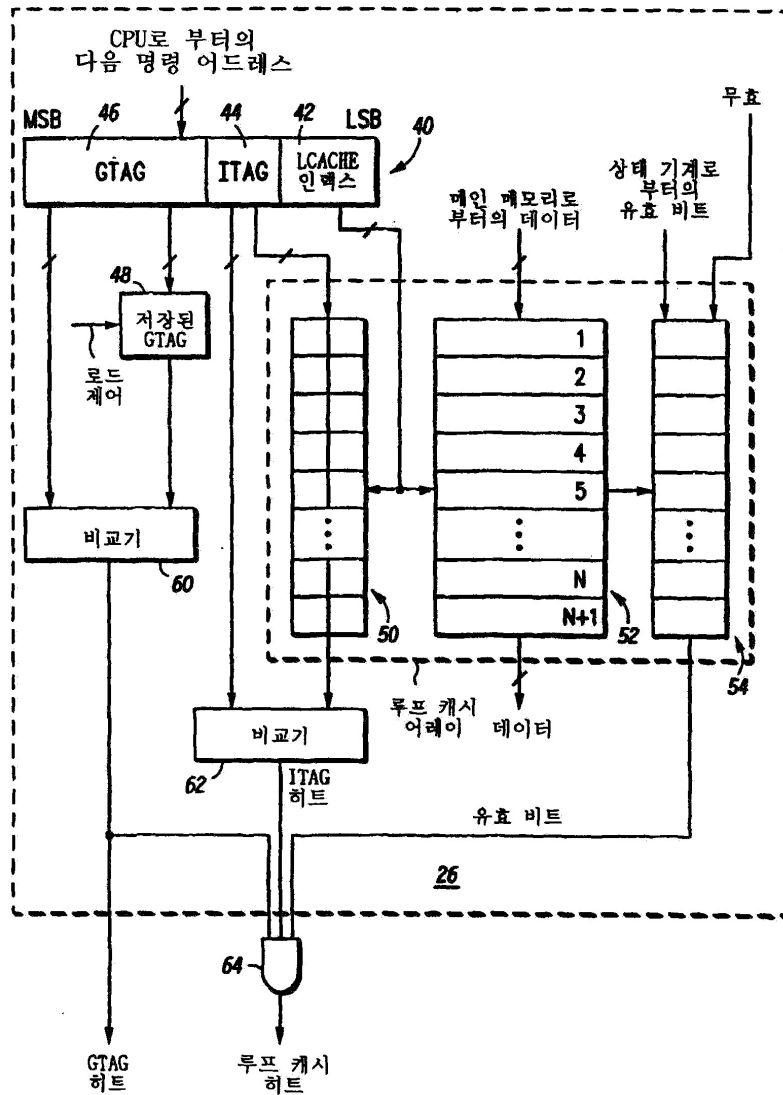
상기 명령 루프를 상기 캐시 어레이에 저장하는 단계를 포함하는, 캐시 메모리 시스템에 데이터를 저장하기 위한 방법.

도면

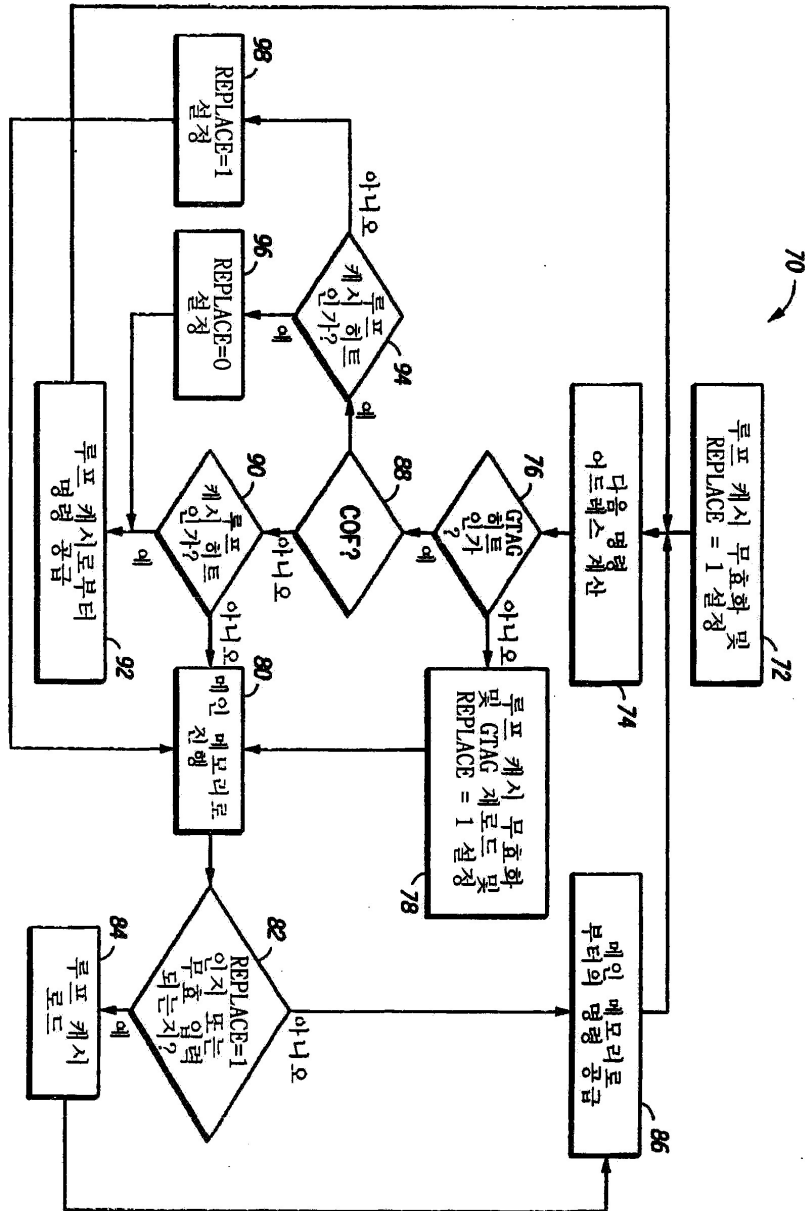
도면1



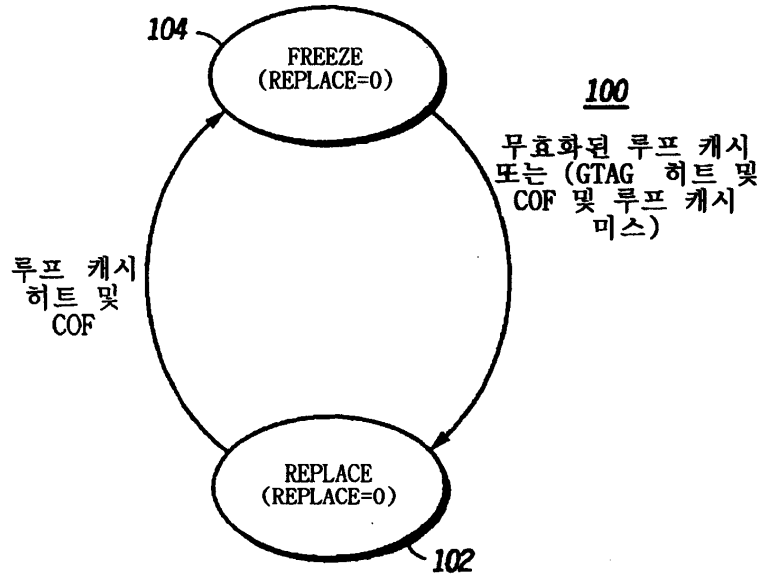
도면2



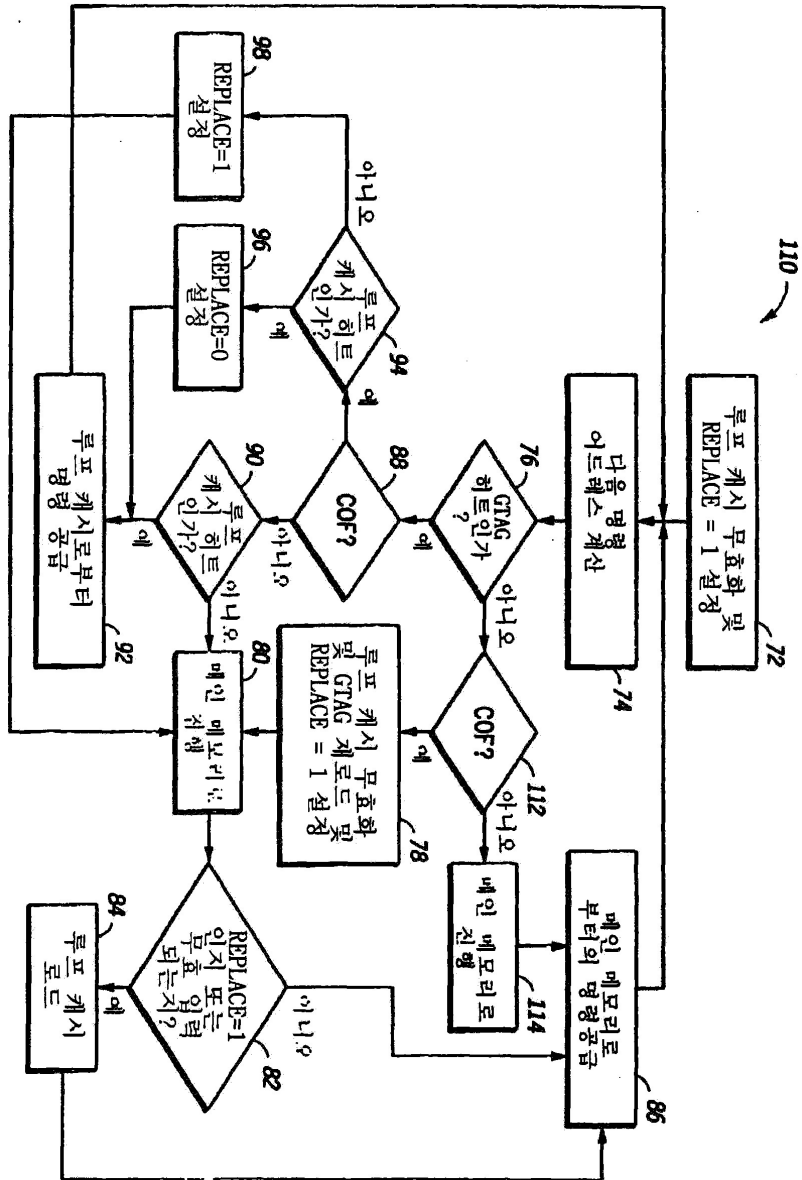
도면3



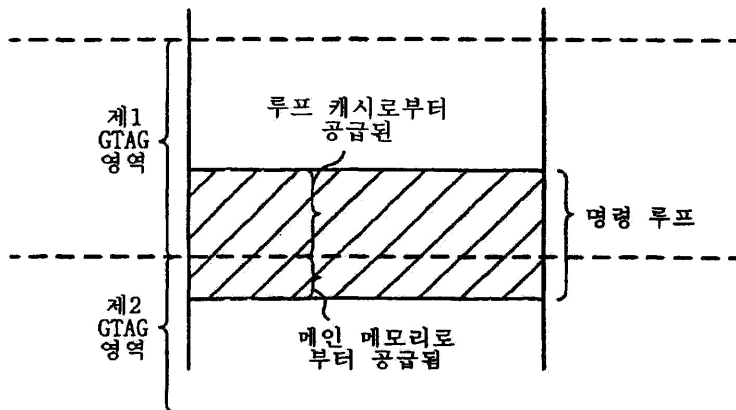
도면4



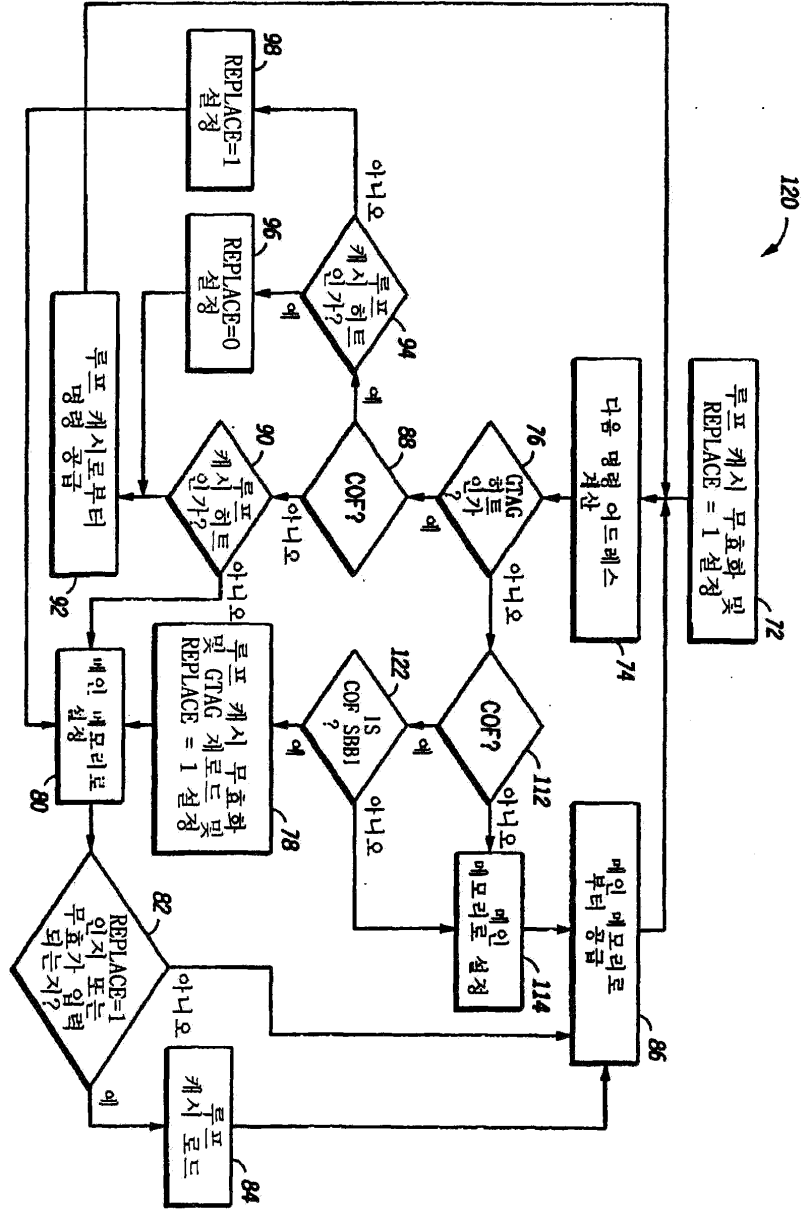
도면5



도면6



도면7



도면8

