(19) 대한민국특허청(KR)
(12) 특허공보(B1)

(51) Int. Cl.[4]
G06F 9/22
G06F 11/00

(45) 공고일자 1986년09월26일
(11) 공고번호 86-001476

(21) 출원번호	특1983-0004555	(65) 공개번호	특1984-0006091
(22) 출원일자	1983년09월28일	(43) 공개일자	1984년11월21일

(30) 우선권주장 57-168912 1982년09월28일 일본(JP)
(71) 출원인 후지쓰 가부시끼 가이샤 야마모도 다꾸마
일본국 가나가와껜 가와사끼시 나까하라꾸 가미고다나까 1015반지

(72) 발명자 사또오 기요스미
일본국 가나가와껜 가와사끼시 미야마에꾸 무까이가오까 1496반지 1고오
시미즈 가즈유끼
일본국 도오꾜오도 마찌다시 오가와 2쬬오메 19반 22고오

(74) 대리인 장용식

심사관 : 허정훈 (책자공보 제1205호)

(54) 마이크로 프로그램 제어장치의 에러로부터 회복하는 방법

요약

내용 없음.

대표도

도1

명세서

[발명의 명칭]

마이크로 프로그램 제어장치의 에러로부터 회복하는 방법

[도면의 간단한 설명]

제1도는 본 발명 방법의 일실시예를 수행하기 위한 블록회로도.

제2a도는 처리장치의 통상의 동작을 설명하는 타이밍도.

제2b도는 본 발명에 따른 재시행 동작을 설명하는 타이밍도.

제2c도는 본 발명에 따른 제2 또는 더늦게 발생하는 에러로부터의 회복을 설명하는 타이밍도.

제3도는 제1도에 도시된 회로의 일부 상세예도.

* 도면의 주요부분에 대한 부호의 설명

1 : 명령레지스터 2 : 제어기억부 번지레지스터

3 : 제어기억부 4 : 마이크로 명령레지스터

5 : 태그 7 : 에러검출 및 정정회로

[발명의 상세한 설명]

본 발명은 마이크로 프로그램 제어장치, 특히 소위 파이프라인(pipeline)프로세서와 같은 마이크로 프로그램에 의해서 동작되는 장치에서 마이크로 명령에 대한 에러 검출 및 정정방법에 관한 것이다.

에러검출 코우드(error checking code : ECC)는 항상 마이크로명령 신뢰도를 증가시키기 위하여 마이크로 프로그램 제어장치의 제어기억부에 기억될 각 마이크로 명령에 가하여진다. 디코우드시 ECC에 의해 마이크로명령이 올바른지 또는 에러를 포함하고 있는지를 검사할 수 있다.

종래의 ECC 디코우딩 동작은 제어기억부로부터 판독된 마이크로명령의 실행과 병행하여 실행된다. 이것은 ECC를 디코우드하는데 비교적 긴 시간이 걸리기 때문이다. 그러므로 ECC 디코우딩 동작 및 마이크로 명령의 실행의 순차동작은 마이크로 프로그램 제어장치의 동작속도에 역으로 영향을 미친

다.

이러한 병렬동작 및 처리동안 어떠한 에러의 검출은 즉시 잘못된 마이크로명령의 실행을 정지시킨다. 그후 마이크로명령이 정정되어 "재시행"(retry) 동작이 다시 실행된다.

그러나 만약 에러가 제어기억부내의 고정비트(fixed bit)에 있을 경우에 상기 처리는 그런 에러비트를 포함하는 마이크로명령이 지정될 때마다 반복된다. 이것은 마이크로 프로그램 제어장치의 효율을 감소시킨다.

그러한 문제는 특히 파이프라인 처리에 있어서 매우 심하다. 이와 같은 프로세서에 있어서 다수의 마이크로 명령은 동시에 실행된다. 그러므로 재시행 동작은 바로 잘못된 마이크로 명령으로부터가 아니라 조금 이전의 마이크로명령 뒤로부터 시작되어야 한다.

본 발명의 목적은 마이크로 프로그램 제어장치의 감소된 효율문제를 극복하는 것, 즉 짧은 시간내에 고정에러로부터 장치를 회복시키는 데에 있다.

상기한 목적은 인터록(interlock) 모우드하에서 초기 재시행 동작을 실행하는 방법에 의해서 얻어진다. 초기 재시행동작에서 제어기억부내에 에러비트를 포함하는 마이크로명령 및 바로전의 마이크로명령의 일부는 에러를 검출 및 정정하기 위하여 처리되며 그후 전과 같은 방법으로 모두 재시행된다.

그러나 동시에 잘못된 마이크로명령을 지정하는 번지가 기록되다. 잘못된 마이크로명령을 지정하는 번지가 다시 제어기억부에 억세스될 때 프로세서는 재시행동작을 다시 하는 것이 아니라 단지 잘못된 마이크로명령에 대해서만 인터록 모우드로 동작한다.

본 발명은 첨부된 도면을 참고로 하여 다음의 설명으로부터 좀더 명확해질 것이다.

제1도는 본 발명 방법의 일실시예를 수행하기 위한 회로도이다. 제1도에서, 부재번호 1은 명령 레지스터(IR)를 표시한다. 레지스터(1)에 기억된 각 명령은 주기억장치(MM)로부터 공급된 절대명령(absolute instruction)이다.

기록된 명령, 즉 명령 코우드부의 초기값은 제어기억부 번지 레지스터(CSAR)(2)에 세트된다. 환원하면, 마이크로명령을 지정하는 번지가 레지스터(2)에 기억된다. 이것들은 마이크로명령이 제어기억부(CS)(3)로부터 판독하는 것을 가능하게 한다.

레지스터(1)에 로우드된 절대명령의 연속된 마이크로명령은 번지가 기억부(3)의 디코우더(DEC)에 공급될 때마다 레지스터(2)의 번지를 증가시키는 번지 갱신회로(+1)(6)에 의해서 제어기억부(3)로부터 순차적이고 자동적으로 발생된다.

다시 말하면 기억부(3)에 연속된 마이크로명령을 지정하는 증가된 번지는 회로(6)의 도움을 받아 레지스터(2)로부터 발생된다. 대신으로, 소위 넥스트 번지방법(next adderess method)이 사용될 수 있다.

각 마이크로명령은 기억부(3)로부터 판독되어 마이크로명령 레지스터(MIR)(4)에 직접 세트된다. 각 마이크로명령은 마이크로명령(IA,IB, 등등)으로 이용될 때까지 기억하기 위하여 파이프라인 처리부(5)에 설치된 태그(B,C, 등등)에 순차적으로 전송된다. 태그(B,C, 등등)는 레지스터로서 작용한다. 여기에 기억된 마이크로명령은 파이프라이 처리장치(5)를 실행하기 위하여 구비된다.

동시에 레지스터(4)의 내용은 에러검출 및 정정회로(ECC)(7)에 의해서 검사된다. 만약 회로(7)가 내용중에 에러가 없는 것을 지시할 경우 그 내용은 상기한 바와같이 태그(B,C, 등등)에 순차적으로 전송된다.

이와 반대로 만약 회로(7)가 에러가 있음을 지적할 경우 처리는 즉시 중단된다. 이후 곧 재시행 동작이 시작된다.

재시행동작은 잘못된 마이크로명령에 대해서뿐 아니라 약간의 바로전의 마이크로명령에 대해서도 수행되어야 한다. 즉, 재시행동작은 관련된 절대명령의 초기 마이크로명령으로 되돌아감에 의해서 수행된다.

이것은 파이프라인 처리에 있어서 잘못된 마이크로명령이 검지될 때 초기 마이크로명령이 아직 완료되지 않는 가능성이 있기 때문에 필요하다.

본 발명에 따르면 재시행동작은 인터록 모우드하에서 실행된다. 여기에서 용어 "인터록"은 회로가 ECC에 의해서 각 마이크로명령을 검사하고 검출된 잘못된 마이크로명령을 정정하여 레지스터(4)로 정정된 마이크로명령을 재기입하는 것을 의미한다. 이후, 프로세서의 동작을 동기시키기 위하여 한 스텝이 레지스터(2), 기억부(3), 레지스터(4) 및 태그(B,C, 등등)의 각 처리에 삽입된다.

또한 재시행동작동안 회로(7)에 의해 정정 마이크로명령에 대하여 동일한 정정을 수행하나 그들이 잘못되지 않은 때와 같은 마이크로명령의 요지를 유지한다.

따라서 절대명령이 인터록 모우드하에서 처리된 후 다음번 절대명령이 주기억장치(MM)로부터 레지스터(1)로 로우드된다. 새로 로우드된 절대명령은 기억부(3)에 기억된 대응하는 파이프라인 마이크로프로그램을 한정한다.

이 마이크로명령은 통상과 같이 각 ECC를 사용함에 의해서 디코우딩 동작과 병렬로 실행된다.

재시행 동작동안 또한 잘못된 마이크로명령을 지정하는 번지가 에러번지 기억레지스터(EASR)(9)에 기록된다. 일단 기록되면 에러번지는 레지스터(2)에 기억된 명령의 명령코우드부로 지시된 번지와 비교회로(10)에 의해서 계속적으로 비교된다.

만약 일치한다면, 즉 잘못된 마이크로명령의 번지가 검출되면 프로세서는 단지 그 잘못된 마이크로

명령에 대해서만 인터록 모우드로 동작한다. 인터록 모우드에 대해서는 부가스텝이 전술한 바와 같이 프로세서에 전체적으로 삽입되어야 한다. 이와같은 스텝의 삽입은 타이밍 제어회로(8) 및 클록 제어회로(11)에 의해서 수행된다.

레지스터(9)는 다수의 에러번지를 기록할 수 있다는 것을 이해해야 한다. 따라서 만약 둘 또는 그이상의 에러비트가 기억부(3)내에 존재할 경우 레지스터(9)는 그들을 기록할 수 있다. 이 경우에 하나가 다수의 비교회로에 제공되어야 한다. 비교회로들중의 임의의 어떤 하나가 일치함을 지시할 때 그 후 프로세서는 관련된 마이크로명령에 대하여 인터록 모우드로 동작하게 된다.

따라서 재시행동작은 잘못된 명령의 단지 첫번째 발생에 대해서만 수행된다. 다음번 발생에 있어서는 단지 잘못된 명령에 대하여 인터록모우드로 프로세서가 동작하는 것만이 필요하다. 이것은 매우 스므스한 파이프라인처리 진행을 가능하게 한다. 이 경우에 인터록 모우드동작에 필요한 시간은 재시행동작에 대한 시간보다 상당히 짧다는데에 유의해야 한다.

본 발명에 따른 방법이 제2a도, 제2b도 및 제2c도 참고로 하여 좀더 명확하게 설명된다. 제2a도는 프로세서에서 통상의 동작을 설명하는 타이밍도이다. 제2a도, 제2b도 및 제2c도에서 주기억장치(MM), 명령레지스터(IR)(1), 제어기억 번지레지스터(CSAR)(2), 제어기억부(CS)(3), 마이크로명령 레지스터(MIR)(4), 태그(B,C), 번지갱신 회로(+1)(6) 및 에러검출 및 정정회로(ECC)(7)는 제1도에 표시된 것과 동일하다.

동작단계는 도시된 바와 같이 단계-A, 단계-B, 단계-C 등등으로 분할된다. 단계-A는 스텝(D), 즉 디코우딩주기 및 스텝(R), 즉 레지스터 판독사이클을 포함한다. 단계-B는 스텝(A), 즉 번지계산 사이클 및 스텝(B^{I}), 즉 버퍼 판독사이클을 포함한다. 단계-C는 스텝(B_2), 즉 다른 버퍼판독 사이클을 포함한다. 이 스텝들은 소위 머신 사이클(machine cycle)과 동기하여 진행된다.

상기한 스텝들은 단지 예이며 실제는 각 머신(처리장치)의 구조에 따라서 결정된다는 것을 이해해야 한다. 주기억장치(MM)로부터 레지스터(4)로 데이타의 이동에 대하여는 이미 설명하였다. 마이크로명령의 IA부는 처리장치의 오퍼레이션에 사용된다. 동시에 마이크로명령은 ECC(7)에 의해서 검사된다. 만약 회로(7)가 어떠한 에러도 마이크로명령에 대해서 발견하지 못할 경우 신호(S)를 발생한다.

그후 회로(6)는 레지스터(2)에 대해 다음번 번지를 발생하며 동일한 동작이 그것에 의해 지정된 마이크로명령과 같이 수행된다. 만약 회로(7)가 마이크로명령내에서 에러를 발견하는 경우 신호

($\overline{S}$)를 발생한다. 이것은 즉시 파이프라인 동작을 정지시키며 재시행동작을 개시시킨다.

제2b도는 본 발명에 따른 재시행 동작을 설명하는 타이밍도이다. 재시행 동작은 인터록 모우드하에서 이루어진다. 즉, 회로(7)는 ECC를 사용함에 의해서 각 마이크로명령을 검사한다. 만약 마이크로명령이 올바른 경우는 더미 정정동작(dummy correcting operation) "CRCT"이 수행되며 그것의 요지는 레지스터(4')에 현재와 같이 유지된다.

레지스터(4')는 레지스터(4)와 동일하며 단지 시간적 순서를 명확하게 하기 위하여 표시된다는 것에 유의해야 한다.

반면에, 만약 마이크로명령이 올바르지 않을 경우 회로(7)는 신호($\overline{S}$)를 발생한다. 이때 잘못된 마이크로명령에 대한 정정동작 "CRCT"은 회로(7)로부터 발생된 소위 신드롬(SYD)에 의해서 수행된다. 이렇게 정정된 마이크로명령은 레지스터(4')에 설정된다. 이 경우에 신호($\overline{S}$)는 레지스터(9)를 활성화하여 레지스터(2)로부터 거기에 에러번지를 기록한다.

제2b도로부터 명확한 바와 같이 재시행동작은 각 마이크로명령은 한 스텝이 아니라 두개의 연속된 스텝, 즉 인터록 모우드로 인하여 중복된 스텝(R)으로 처리되기 때문에 다소 중복된다. 그러나 이로 인해 모든 마이크로명령이 확실하게 정정되는 장점이 있다.

가장 중요한 장점은 일단 재시행동작이 어떠한 잘못된 명령에 대하여 수행되면 후에 동일한 잘못된 명령이전의 명령에 대하여 다시 수행할 필요가 없다는 것이다.

제2c도는 본 발명에 따른 에러의 2번째 또는 다음번의 발생으로부터 회복을 설명하는 타이밍도이다. 제2b도는 참고하여 상기한 바와같이 일단 재시행동작이 잘못된 마이크로명령으로 인하여 실행된 경우 그 마이크로명령을 지정하는 번지가 레지스터(9)에 기록된다. 비교회로(10)는 레지스터(2)로부터의 번지를 레지스터(9)의 번지와 계속하여 비교한다.

만약 그들사이의 비교결과와 불일치 "CO"를 지시할 경우 파이프라인 동작은 통상적으로 계속된다. 만약 그들사이의 비교결과가 일치"CO"를 지시할 경우 프로세서는 단지 잘못된 마이크로명령에 대해서만 인터록 모우드로 동작한다.

즉, 정정동작 "CRCT"은 관련 ECC로부터의 신드롬(SYD)에 의해서 수행되어 잘못된 명령을 정정하고 레지스터(4')에 그것을 재기억시킨다.

이 경우에 스텝(R)은 인터록 모우드로 인하여 중복으로 된다. 그러나 이와같은 중복스텝은 통상적인 재시행동작에 필요한 시간보다 상당히 짧은 시간이 걸린다.

제3도는 제1도에 도시된 회로의 부분상세예이다. 상세회로는 주로 제1도에 표시된 회로(7), 타이밍

제어회로(8) 및 클록 제어회로(11)에 향해져 있다. 회로(7)는 소위 해밍 코우드(Hamming code)를 포함하는 데이타를 수신하여 정정회로(21) 및 에러검사회로(22)에 데이타를 인가한다.

에러검사회로(22)는 한편으로 에러지시 플립 플롭(23)을 활성화시키며 다른 한편으로는 잘못된 마이크로명령의 정정에 사용되는 신드롬을 기억하는 레지스터(24)를 활성화시킨다. 만약 에러가 존재할 경우 논리 "1"의 에러지시신호(EI)가 플립플롭(23)으로부터 AND 게이트(25 및 25')에 공급된다. 만약 재시행동작이 현단계에서 수행되지 않으면 신호(-RET)는 재시행 처리부(RP)(도시되지 않음)로부터 공급된다. 만약 재시행동작이 현재 수행중이면 신호(+RET)가 그로부터 공급된다.

AND 게이트(25')에 관해서는 신호(-RET)와 에러지시 신호(EI)가 모두 활성화 상태일 때 AND 게이트(25')로부터의 출력은 재시행 처리부(RP)를 트리거하여 재시행동작을 시작한다. AND 게이트(25)에 대해서는 신호(+RET) 및 신호(EI)가 모두 활성화상태일 때 AND 게이트(25)의 출력은 AND 게이트(26)를 열어 기록가능신호(WE)로서 클록(CLK)을 통과시키며 그 결과 레지스터(2)의 에러번지가 레지스터(9)에 기록된다. 기록된 번지는 비교회로(10)에 의해서 레지스터(2)의 번지와 비교된다.

회로(10)는 상기한 일치신호(CO) 및 불일치신호(CO)를 발생한다. 논리 "1"의 신호(CO) 및 신호(+RET), 또는 어느 하나가 OR 게이트(27)를 통하여 통과할 수 있다. 만약 상태신호(ST)가 현단계가 스텝(D)(제2b도 및 제2c도 참조)에 위치해 있는 것을 지시할 경우 AND 게이트(28)가 열려져서 제2c도의 경우는 신호(CO)가 통과하도록 하며 제2b도의 경우는 신호(+RET)가 통과하도록 한다.

AND 게이트(28)로부터의 출력은 플립플롭(29)을 세트하며 이 출력은 인터록 모우드상태(IMS)를 지시한다. 반면에 신호(ST)가 플립플롭의 세트입력(ST)에 플립플롭(30)을 세트하며 그 출력은 스텝(R)(STR)이 현재 실행중인 것을 지시한다.

플립플롭(30)은 PHA-REL 상태신호에 의해서 리세트되며 이 신호는 파이프라인 프로세서에 단계-A로부터 단계-B(제2a도 참조)로 상태전환을 하도록 항상 발생된다. 문자(REL)는 해제를 표시한다.

만약 신호(PHA-REL)가 플립플롭(30)의 리세트입력(RST)에 직접 인가되면 중복스텝(B)(제2b도 및 제2c도 참조)이 얻어지지 않게 된다. 따라서 신호(PHA-REL)는 AND 게이트(31)를 통하여 입력(RST)에 가해진다.

AND 게이트(31)는 플립플롭(29)의 반전출력이 논리 "1"일 때 열린다. 다시 말하면, 단지 인터록 모우드상태(IMS)가 발생될 경우에만 AND 게이트(31)가 열려지지 않기 때문에 그러므로 스텝(R)은 중복된다. 따라서 연장신호(PHA-REL)는 AND 게이트(31)로부터 공급되며 이 신호는 또한 AND 게이트(32)를 여는데 사용되며 더우기 다른 클록 제어수단(CCM)에 공급된다.

(57) 청구의 범위

청구항 1

마이크로명령을 기억하는 최소한 1개의 제어기억부, 마이크로명령을 지정하는 번지를 기억하는 제어기억부 번지 레지스터, 마이크로명령 레지스터, 마이크로명령 레지스터외 다단으로 구성되어 태그로 사용되는 직렬 접속된 레지스터들 및 각 마이크로프로그램의 실행과 병렬로 초기 레지스터에 기억된 각 마이크로명령의 타당성(validity)을 검사하여 마이크로명령 레지스터에 재기억될 잘못된 마이크로명령을 정정하도록 동작하는 에러검출 및 정정회로로 이루어지는 마이크로 프로그램 제어장치의 에러로부터 회복하는 방법에 있어서, 각 마이크로명령이 에러를 포함하는지 여부를 검사하는 단계와, 잘못된 마이크로명령 및 관련된 선행 마이크로명령이 연속적으로 에러 검출 및 정정을 하고 다음에 재실행되는 인터록 모우드하에서 에러가 검출된 마이크로명령을 포함하는 관련 마이크로명령에 대하여 재시행동작을 수행하는 단계와, 잘못된 마이크로명령을 지정하는 번지를 기록하는 단계 및 잘못된 마이크로명령에 대한 번지가 다시 억세스될 때 단지 잘못된 마이크로명령에 대해서만 인터록 모우드동작을 수행하는 단계로 구성되는 것을 특징으로 하는 마이크로 프로그램 제어장치의 에러로부터 회복하는 방법.

청구항 2

제1항에 있어서, 잘못된 마이크로명령에 대한 번지의 재억세싱(reaccessing)은 에러번지 기억레지스터에 기억되어서 잘못된 마이크로명령을 지정하는 상기한 기록된 번지와 상기 제어기억부 번지레지스터에 있는 각번지를 비교함에 의해서 검출되는 것을 특징으로 하는 에러로부터 회복하는 방법.

청구항 3

제2항에 이씨어서, 상기 기록될 번지는 단지 고정적으로 발생하는 잘못된 마이크로명령에 대한 것임을 특징으로 하는 에러로부터 회복하는 방법.

청구항 4

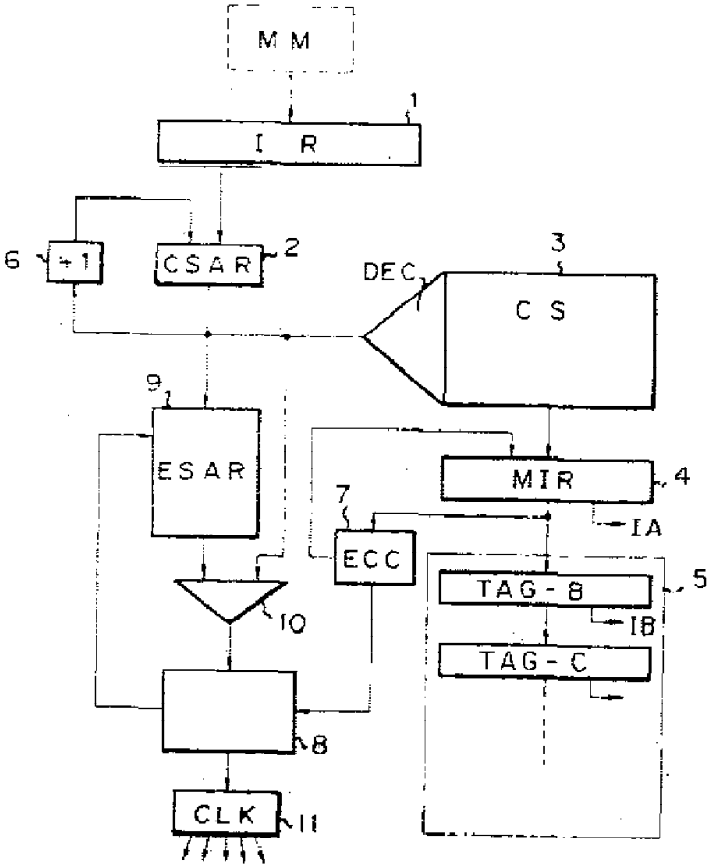
제3항에 있어서, 상기 에러번지 기억레지스터는 에러가 상기 두번째 단계에서 검출되는 바로 그 시간에 상기한 제어기억부 번지레지스터에 기억된 동일한 번지를 기록하는 것을 특징으로 하는 에러로부터 회복하는 방법.

청구항 5

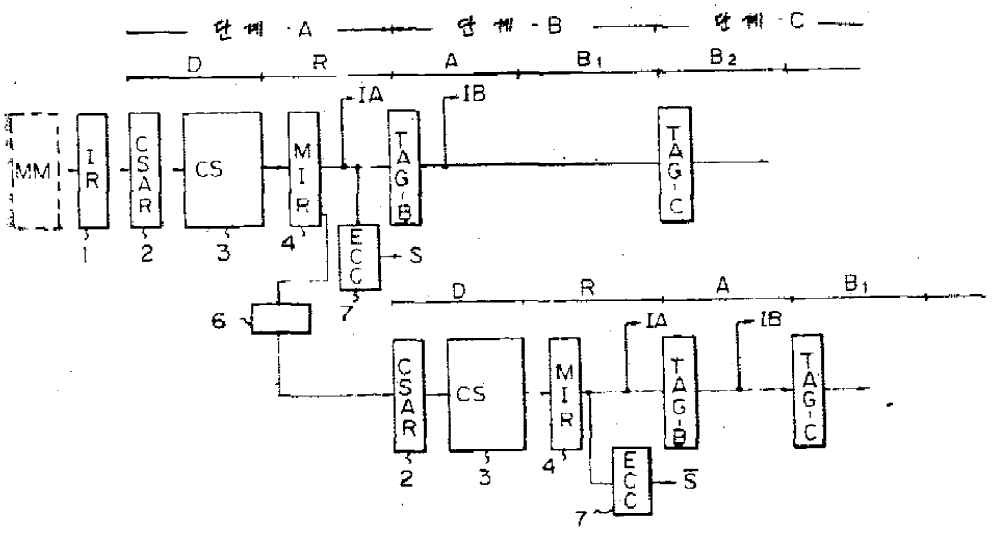
제1항에 있어서, 상기 인터록 모우드는 상기 장치의 머신 사이클에 의해 규정되는 각 스텝들중 어느 하나를 중복함에 의해서 만들어지는 것을 특징으로 하는 에러로부터 회복하는 방법.

도면

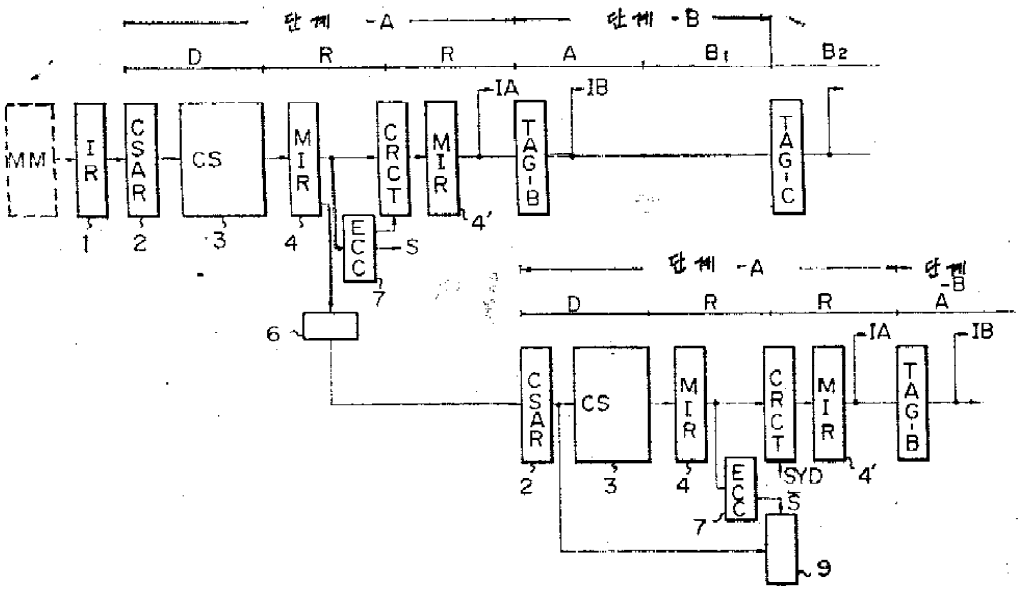
도면1


M M
1
I R
6
+1
CSAR
2
DEC
3
C S
9
ESAR
MIR
4
7
IA
ECC
TAG-B
5
IB
10
TAG-C
8
CLK
11

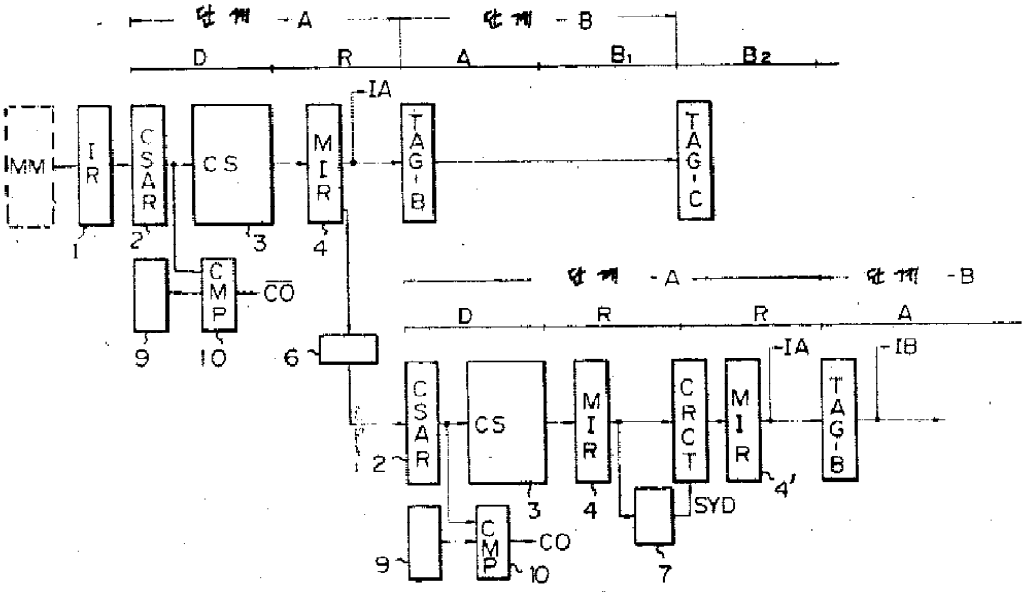
도면2A


단계-A
단계-B
단계-C
D
R
A
B1
B2
IA
IB
MM
IR
CSAR
CS
MIR
TAG-B
TAG-C
ECC
S
1
2
3
4
7
6
5

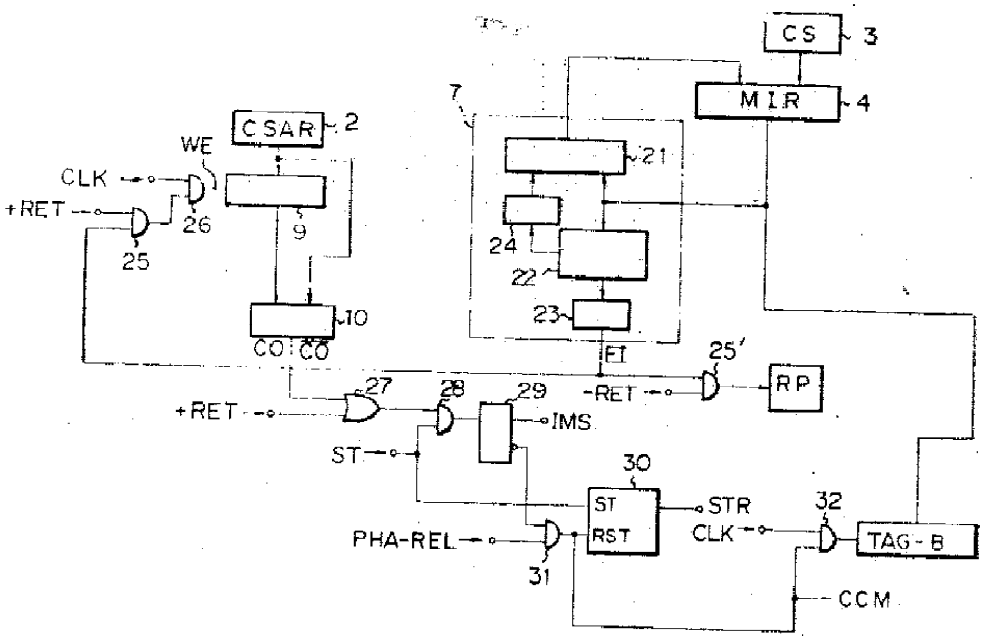
도면2B


단계 -A
단계 -B
D
R
R
A
B1
B2
MM
IR
CSAR
CS
MIR
CRCT
MIR
IA
TAG-B
IB
TAG-C
ECC
S
SYD
1
2
3
4
4'
5
6
7
9

도면2C


단계 -A
단계 -B
D
R
A
B1
B2
MM
IR
CSAR
CS
MIR
IA
TAG-B
TAG-C
CMP
CO
SYD
1
2
3
4
4'
6
7
9
10

도면3


CS
3
MIR
4
7
21
24
22
23
CSAR
2
WE
CLK
+RET
26
25
9
10
CO
CO
FI
25'
RP
-RET
27
28
29
+RET
IMS
ST
30
ST
RST
STR
CLK
32
TAG-B
PHA-REL
31
CCM