

(19)대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) 。 Int. Cl.<sup>7</sup>  
G06F 12/12

(45) 공고일자 2005년03월16일  
(11) 등록번호 10-0476446  
(24) 등록일자 2005년03월03일

(21) 출원번호	10-2002-7008325	(65) 공개번호	10-2002-0097145
(22) 출원일자	2002년06월26일	(43) 공개일자	2002년12월31일
번역문 제출일자	2002년06월26일		
(86) 국제출원번호	PCT/US2000/042305	(87) 국제공개번호	WO 2001/50269
국제출원일자	2000년11월27일	국제공개일자	2001년07월12일

(81) 지정국

국내특허 : 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아 헤르체고비나, 바르바도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀란드, 영국, 그루지야, 헝가리, 이스라엘, 아이슬란드, 일본, 케냐, 키르기즈스탄, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 리베이라, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아 공화국, 몽고, 말라위, 멕시코, 노르웨이, 뉴질랜드, 슬로베니아, 슬로바키아, 타지키스탄, 투르크멘, 터키, 트리니다드토바고, 우크라이나, 우간다, 미국, 우즈베키스탄, 베트남, 폴란드, 포르투갈, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 아랍에미리트, 안티구와바부다, 코스타리카, 도미니카, 알제리, 모로코, 탄자니아, 남아프리카, 벨리제, 모잠비크, 그라나다, 가나, 감비아, 크로아티아, 인도네시아, 인도, 시에라리온, 세르비아 앤 몬테네그로, 짐바브웨,

AP ARIPO특허 : 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 시에라리온, 가나, 감비아, 짐바브웨, 모잠비크, 탄자니아,

EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르기즈스탄, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크멘,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투갈, 스웨덴, 핀란드, 사이프러스, 터키,

OA OAPI특허 : 부르키나파소, 베닌, 중앙아프리카, 콩고, 코트디부아르, 카메룬, 가봉, 기니, 말리, 모리타니, 니제르, 세네갈, 차드, 토고, 기니 비사우,

(30) 우선권주장 09/476,444 1999년12월30일 미국(US)

(73) 특허권자 인텔 코오퍼레이션  
미합중국 캘리포니아 산타클라라 미션 칼리지 블러바드 2200

(72) 발명자 클라크로렌스티.  
미국85048애리조나피닉스테저트윌로우로드이.3161

클라크매튜엠.  
미국85284애리조나템페칸플레이스더블유.261

(74) 대리인 유미특허법인

심사관 : 임영희

(54) 라운드 로빈 및 로킹 캐시 대체 수행 방법 및 장치

## 명세서

### 기술분야

본 발명은 캐시 대체 기법에 관한 것이다. 특히, 라운드 로빈 캐시 대체와 캐시 라인 로크 기법의 결합에 관한 것이다.

### 배경기술

새로운 데이터 셋트가 캐시에 로드(load)될 때 어떤 데이터를 대체해야 하는가를 정하는 여러 기법이 존재한다. 직접 매핑된 캐시(direct mapped cache)에서 선호하는 한 가지 기법은 LRU(least recently used) 기법이다. LRU 기법에서, 명칭이 의미하듯이, 가장 최근에 사용된 데이터 블록을 새로 들어오는 데이터로 대체한다. 컴퓨터 소프트웨어 실행 및 데이터 사용에 공통되는 지역성(locality)으로 인하여, 이는 매우 효율적인 캐시 대체 기법으로 인식되었다. 하지만, 셋트 연상성이 증가함에 따라, 하드웨어 요구의 증대로 인하여 LRU에 대한 복잡도 및 오버헤드(overhead)는 그러한 기법을 무용화시키고 있다.

또한, 별개의 디코더(decoder)가 없는 CAM(content addressable memory) 기반 캐시에 있어서, 높은 셋트 연상성(set-associativity)을 달성하고 채움 데이터(fill data)를 어디에 써야 할지를 지정하는 것이 점차 문제시되고 있다. 이용되고 있는 한 가지 가능한 해결책은 로드될 라인의 지정을 통하여 원형의 시프트 레지스터가 루핑(looping)하는 라운드 로빈 대체 기법이다. 이는 캐시에 가장 오래된 정보를 버리지만, 대체는 사용 빈도와는 완전히 독립적이다. 따라서, 이는 아주 많은 양의 캐시를 버리는 결과가 된다.

데이터가 미리 캐시에서 버려지는 것을 방지하기 위한 일부 종래의 해결책은 웨이 로킹(way locking)이었다. 이 시스템은 많은 세분성(granularity)을 주지는 못하였다. 예를 들면, 2 웨이 캐시에서, 보유될 데이터 블록이 비교적 작을지라도 캐시의 절반은 로크된다. 이는 종종 캐시 사용에 있어서 비효율적이다.

### 발명의 상세한 설명

제1 복수의 래치는 각각이 소정 캐시 라인과 결합하고 레지스터를 형성하면서 데이터 체인된다(daisy chained). 유사하게, 제2 복수의 래치는 각각이 캐시 라인과 결합하여 데이터 체인된다. 제1 레지스터는 캐시 라인의 채움 순서(fill order)를 결정하고 제2 레지스터는 캐시 라인의 로크 순서를 결정한다.

### 도면의 간단한 설명

도 1은 본 발명의 일 실시예의 블록도이다.

도 2는 본 발명의 일 실시예에서 라운드 로빈 로크 논리의 블록도이다.

도 3은 8개 사이클 구간 동안의 예시적인 일련의 로드를 도시한다.

도 4는 본 발명의 일 실시예에서 신호의 서브 셋트에 대한 타이밍도이다.

### 실시예

도 1은 본 발명의 일 실시예의 블록도이다. CAM(102) 어레이는 RAM(104)에 주소를 지정하는데 사용된다. CAM(102)과 RAM(104)의 결합은 예를 들어, 캐시 특히, 레벨 제로 캐시, 또는 다른 실시예에서는 TLB(translation lookaside buffer)일 수 있다. 어느 경우에서나, CAM(102)은 대응 엔트리(entry)에 대응하는 주소를 포함하며, 전자의 경우에는 코드 또는 데이터 라인이며 후자의 경우에는 페이지 테이블 엔트리(page table entry)이다. CAM 엔트리에 포함된 데이터는 어느 경우나 "태그(tag)" 엔트리라 불린다. 라운드 로빈 및 로크 로직(100)과 같은 제어 로직 유닛(control logic unit)은 CAM(102)에 결합된다. 제어 로직 유닛은 데이터가 CAM(102)과 RAM(104)에 쓰여질 때, 예를 들면, 라인이 캐시 채움 동작시 대체될 때 CAM이 주소 지정한 캐시에 의하여 이용되는 대체 방식을 제어한다. CAM(102)과 RAM(104) 두 회로는 당업자에게 공지된 수단을 통하여 쓰여질 수 있다. CAM(102)은 CAM의 콘텐츠 주소 지정 가능한 속성을 통하여 매칭 엔트리(matching entry)를 결정하도록 접근된다. 매치가 발견되면, 센스 증폭기(sense amp)(106)를 통하여 출력으로서 RAM(104)에서 데이터를 읽거나, 또는 상기한 채움 동작에 동안에 RAM(104)에 데이터를 쓰는 것과 같이 동일한 회로를 통하여 RAM(104)에 대한 쓰기 동작을 수행할 수 있다. 일반적으로, 이러한 형태의 캐시 회로 구조는 태그 데이터를 저장하는 RAM 셀을 이용하는 통상의 구조에 비하여 낮은 전력 특성 때문에 선호된다.

라운드 로빈 및 로크 로직(100)은 로드 요청(load request)을 수신하는 이외에, 로크 클록, 로크 선택, 로크 클리어 및 라운드 로빈 클리어를 포함하는 여러 제어 신호를 수신한다. 로크 클록이 어서팅되는 것은 로드 요청이 로크라는 것을 나타낸다. 로크 클록이 어서팅되지 않은 경우, 로드가 채워진다. 여기서 사용하는 바와 같이, 로크는 후속 로드로 의하여 덮어쓰기(overwrite)가 되지 않도록 로크되는 라인의 로드이다. 채움(fill)은 통상의 캐시 대체 방식의 일부로서 자유롭게 덮어쓰기 될 수 있는 로드이다. 로크 클리어는 로크 레지스터를 클리어하여 이전에 로드된 로크에 후속 덮어쓰기를 허용한다. 라운드 로빈 클리어는 라운드 로빈 클리어를 소정 값으로 재설정한다. 로크 선택 신호는 라운드 로빈 레지스터로부터의 신호와 워드 라인 인에이블먼트(word line enablement)에 대한 로크 레지스터 사이에서 선택하는 한 셋트의 멀티플렉서(multiplexor)에 대한 모드 선택으로 사용된다.

도 2는 본 발명의 일 실시예에서 라운드 로빈 로크 로직의 블록도이다. 복수의 라운드 로빈 래치(212, 214, 216, 210)는 한 라인의 캐시와 각각 연결된다. 도시한 실시예에서, 어레이에서는 32개 라인이 있다고 가정한다. 하지만, 더 많거나 적은 라인(따라서, 래치)은 본 발명의 범위 내이다. 따라서, 라운드 로빈 래치(200)를 포함하여 32개 라운드 로빈 래치를 마련한다. 로크 레지스터(202)를 포함하는 제2 복수의 래치(230, 232, 234)는 복수의 캐시 라인 각각과 결합된 하나의 래치를 구비한다. 도시한 실시예에서, 32개 캐시 라인 중에서 31개까지 로크시키는 31개 로크 래치가 있다. 하나 이상의 라인이 로크되는 것을 방지함으로써, 시스템은 새로운 데이터가 캐시에 로드될 필요가 있는 교착 상태를 회피하지만, 그러한 로드를 방지하여 모든 라인을 로크시킨다. 로크 래치는 라운드 로빈 래치로서 함께 데이터 체인된다. 이런 식으로, 로크 레지스터는 로크 클록에 응답하여 어레이의 위에서 아래까지 연속된 로크를 강제한다. 라운드 로빈 래치(200)는 순환 채움을 하는데, 어레이의 가장 높이 있는 언로크 라인 다음에 하부 라인으로 시작하여 하부로 진행하고 순환을 한다. 결합 로직(combination logic)(204)는 채움이 로크된 라인에 덮어쓰기가 되지 않도록 한다. 일 실시예에서, 레지스터(200, 204)는 펄스형 래치(pulsed latch)이고, 다른 실시예에서는 마스터 슬레이브 플립플롭(master slave flip-flop)이다. 펄스형 래치는 보다 영역 효율적이지만, 프로세스 스큐(process skew) 동안에 불충분한 홀드 타임(hold time)과 같은 이슈에 더욱 민감한 경향이 있다.

로크 레지스터는 동작시에 양의 전원에 연결된 입력을 갖는 어레이의 워드 라인(31)에 대응하는 시작 래치(230)를 포함한다. 따라서, 로크 클록이 어서팅될 때, 레지스터(230)의 출력에 하이(high)가 나타나고, 따라서 레지스터(230)의 입력 상에는 도시한 바와 같이 데이터 체인된다. 마스터 슬레이브 플립플롭보다 펄스형 래치를 사용할 때, 일 실시예에서 나타나는 한 이슈는 로크 클록이 어서팅될 경우 두 래치(230, 232)가 셋팅되지 않도록 회로 내에 충분한 지연이 구축되어야 한다는 것이다. 이러한 이슈를 해결하기 위하여, 본 발명의 펄스형 래치 실시예에서, 로크 클록은 로크 레지스터 내의 한 래치가 그 입력값을 수신하게 하는데 충분히 길게 되도록 짧은 펄스 주기이지만, 원하는 목적 래치의 출력에 연결된 래치가 동일한 데이터를 수신하지 않도록 충분히 짧다. 따라서, 다음 로크 클록의 어서팅시, 레지스터(230, 232) 출력 모두에서 하이로 나타난다. 이러한 방식으로, 로크 클록의 31번의 어서팅 후에 로크 래치(234)에 도달할 때까지 로크 레지스터(202)를 통하여 하이 값이 순환한다. 이러한 점에서, 로크 레지스터(202) 내의 31개의 모든 레지스터는 거기에 기억된 한 개의 논리값을 갖는다.

라운드 로빈 레지스터(200)는 라운드 로빈 리셋트 신호의 어서팅 후에 래치(210)를 제외한 모든 래치가 클리어되도록 연결된다. 래치(210)는 라운드 로빈 시작 비트(round robin start bit)이고 라운드 로빈 클리어 신호에 응답하여 설정된다. 이 신호는 풀칩 리셋(full-chip reset)에 어서팅되어 캐시를 초기화시킨다. 래치(210)가 설정되는 경우, 채움은 영의 워드 라인에서 시작할 필요가 있다. 시작(startup) 이외에, 라운드 로빈 리셋트 신호는 로크 발생에 응답하여 어서팅된다. 이는 논리적으로 허용되지 않는 상태를 구성할 수 있는 라운드 로빈 비트와 동기되지 않도록, 즉 로크 라인이 라인 채움을 위한 다음 목표로 선택되게 한다. 하부를 설정하는 이러한 방식은 로크 동작이 수행될 때 대체를 위한 언로크 위치는 논리 중 최소값을 요구한다. 캐시의 효율성에 관한 이러한 방식의 관련성을 이어서 설명한다.

따라서, 결합 로직(204)의 도움으로, 영의 워드 라인의 다음 채움 라인에 이어서, 다음 채움 라인은 로크되지 않은 최상위 라인으로 지정된다. 따라서, 워드 라인(31)이 아직 로크되지 않은 경우, 래치의 출력은 영이 되고 AND 게이트가 "1"을 출력하게 하는 인버터(310)에 의하여 반전되며, 또한 레지스터(212)를 구동시켜 높은 값을 출력시키는 OR 게이트(316)로 "1"을 출력하게 하여, 워드 라인(31)을 선택한다. 멀티플렉서(208)는 로크 선택 신호를 이용하여 라운드 로빈 레지스터 또는 로크 레지스터가 워드 라인 선택하는지 결정한다. 따라서, 로크 동작시, 워드 라인(WL) 선택은 멀티플렉서(208, 209)를 통하여 게이트(312) 출력으로부터 제거되도록 선택된다. 따라서, 로크시 채워지는 라인을 지시하는 어서팅되는 워드 라인(WL)은 논리 1로 설정되는 이전의 로크 래치와 논리 영으로 설정되는 현재의 로크 래치의 일치에 기초한다. 로직으로부터 추론할 수 있듯이, 각각의 로크 동작 워드 라인(WL) 어서팅 후에 로크 클록은 어서팅된다. (언로크된) 라인 채움에 있어서, 멀티플렉서(208)는 멀티플렉서(209)를 통하여 워드 라인 선택(WL SELECT) 노드에 대응하는 라운드 로빈 래치에 포함된 논리 1을 통과시켜 워드 라인(WL)을 어서팅한다. 높은 신호는 다음 채움이 일어나는 라인을 지시하는 포인터(pointer)로 동작하는 라운드 로빈 레지스터를 통하여 전파해간다. 멀티플렉서(209)는 멀티플렉서(208)의 출력과, 어드레싱(addressing)을 위해 CAM을 활용하는 캐시 읽기 및 쓰기 동작 동안 워드 라인 선택(WL SELECT)을 어서팅하게 하는 CAM 매치 신호 사이에서 선택을 한다. 다른 실시예에서, 디코더 출력은 매치 신호(MATCH) 대신에 멀티플렉서(209)에 제2 입력을 제공한다.

시작 레지스터(210)는 상위 팬 아웃(high fanout)을 갖는 라운드 로빈 레지스터 중에 유일한 레지스터임을 유의하여야 한다. 여기에 사용된 바와 같이, "상위" 팬 아웃은 팬 아웃을 구비한 장치의 출력으로 구동되는 3개 이상의 입력으로 간주된다. 버퍼(206)는 레지스터(210)가 다른 라운드 로빈 레지스터의 각각에 대한 신호를 구동해야 할 때 팬 아웃을 수용하는 신호를 완충시키는데 사용된다. 역으로, 다른 라운드 로빈 레지스터는 가장 가까운 이웃에 대한 입력만을 구동시킨다. 따라서, 이들은 아주 작게 유지될 수 있다. 캐시 내의 라인당 2개의 래치가 있고(펄스형 래치 실시예에서), TLB에서 통상 32개이지만 캐시의 경우에는 수 천 개가 있으므로, 이는 바람직하다. 마스터 슬레이브 플립플롭보다 펄스형 래치와 같은 래치 기억 소자를 구비한 실시예를 이용하는 주된 이유가 소형 크기가 바람직하기 때문이다. 팬 아웃을 제어할 때, 장치 크기 및 대응하는 회로의 전력 소비가 감소된다. 시작 레지스터는 하부에 있으므로, CAM 및 RAM 회로의 센스 및 쓰기 회로 영역에 대응하고 어레이의 가장 자리에서 영역을 이용할 수 있으므로, 버퍼의 배치는 단순화된다.

도 3은 8개 사이클 주기 동안 예시적인 일련의 로드를 도시한다. 사이클 1에서, 워드 라인 0에 채움 1이 로드되고, 채움 포인터는 워드 라인 31에 대한 포인트로 나아간다. 사이클 2에서, 워드 라인 31에 채움 2가 로드되고, 채움 포인터는 워드 라인 30에 대한 포인트로 나아간다. 사이클 3에서, 채움 3은 워드 라인 30에 로드된다. 이어, 사이클 4에서, 로크 1이 로크 시작 라인(워드 라인 31)에 로드되면서 채움 2를 배제하고(kicking out) 채움 포인터는 워드 라인 0에 대한 포인트로 재설정된다. 사이클 5에서, 로크 2가 다음 클록 라인(워드 라인 30)에 로드되면서 채움 3을 배제하고 채움 포인터는 워드 라인 0에 대한 포인트로 재설정된다. 사이클 6에서, 채움 4가 로드되면서 채움 1을 배제하고 채움 포인터는 가장 높은 언로크 라인, 여기서는 워드 라인 29에 대한 포인트로 나아간다. 사이클 7에서, 채움 5는 워드 라인 29를 채운다. 사이클 8에서, 로크 3은 채움 5를 배제하고 채움 포인터는 워드 라인 0에 대한 포인트로 재설정된다. 이는 순전히 예시적인 채움 및 로크 패턴이다. 이와는 달리, 로크 1, 2 및 3이 처음 3개 사이클에서 행해지면서 다음 5개 사이클에서 채움 1 내지 5가 행해지는 경우, 데이터는 캐시에서 배제되지 않는다. 따라서, 적절한 소프트웨어 설계는 각각의 로크 후 채움 부분을 재시작하는 잠재적인 위험을 제거할 수 있다. 예를 들면, 소프트웨어는 임의의 캐시 라인을 채우기 전 초기에 로크시킬 필요가 있는 모든 데이터 또는 코드를 로크시킬 수 있다.

먼저 로크를 함으로써, 캐시에 이미 로드된 코드에 라인을 로크하는 잠재적인 부정적 영향을 경감시킬 수 있다. 로크시키기에 바람직한 코드의 일례는 매우 신속히 반응하는 임계 인터럽트 조절 루틴(critical interrupt handling routine) 또는 유사 코드이다. 하지만, 상기한 기재는 본 발명의 일 실시예의 동작의 설명하기 위한 예시이다.

도 4는 본 발명의 일 실시예의 신호 서브셋의 타이밍도를 나타낸다. 펄스 클록 신호(410)는 상기 펄스 래치 실시예에 이용된다. 펄스는 인접 래치로부터의 신호가 그 이웃 래치로 전달되기 전에 로우(low)가 되기에 충분히 짧아야 한다. 따라서, 라운드 로빈 래치(도 2의 212)의 출력은 신호(412)로 도시된다. 이 출력은 펄스 클록(410)에 응답하여 하이가 된다. 신호(412)가 OR 게이트(도 1의 336)에 인가되면서 신호(414)가 나타내는 OR 게이트의 출력이 하이로 어서팅되기 전에 펄스 클록(410)은 로우가 되어야 한다.

일 실시예에서, 각각 연상 라운드 로빈 및 로크 로직 유닛을 갖는 복수의 CAM은 복수의 뱅크를 구비한 더 큰 캐시를 형성하는데 이용된다. 각 뱅크는 뱅크 내에서 그리고 다른 뱅크와는 독립적으로 라운드 로빈 및 로크 기법을 구현한다.

전술한 명세서에서, 소정 실시예를 참조하여 본 발명을 설명하였다. 하지만, 첨부된 청구범위에 기재된 것과 같이 본 발명의 사상과 범위에서 벗어나지 않는 변형과 변경이 가능함은 명백하다. 따라서, 명세서 및 도면은 제한적인 것이 아니라 예시적인 의미로 간주되어야 한다.

## (57) 청구의 범위

### 청구항 1.

각각 제1 시프트 레지스터에 연결되고 캐시 라인과 연관되어 있는 제1의 복수 래치, 그리고

각각 제2 시프트 레지스터에 연결되고 캐시 라인과 연관되어 있는 제2의 복수 래치

를 포함하며,

상기 제1 시프트 레지스터는 상기 캐시 라인에 대한 채움 순서(fill order)를 결정하고, 상기 제2 시프트 레지스터는 상기 캐시 라인에 대한 로크 순서(lock order)를 결정하는, 장치.

### 청구항 2.

제1항에서,

상기 제1의 복수 래치 내의 레지스터 상호간에 대한 입력을 구동시키기 위하여 상기 제1의 복수 래치의 제1 시작 레지스터(start register)의 출력을 증폭시키는 버퍼(buffer)를 더 포함하는 장치.

### 청구항 3.

제2항에서,

상기 제1의 복수 래치 내의 각 레지스터는 상기 복수 래치 내의 그 후속 래치에 대한 입력만을 구동시키는, 장치.

### 청구항 4.

제1항에서,

상기 제1의 복수 래치와 상기 제2의 복수 래치는 그 수가 서로 다른, 장치.

### 청구항 5.

제1항에서,

상기 제1의 복수 래치의 제1 시작 레지스터에 연관된 제1 캐시 라인 주소는, 상기 제2의 복수 래치의 제2 시작 레지스터와 연관된 제2 캐시 라인 주소와 최대 차이를 가지는, 장치.

### 청구항 6.

제5항에서,

제2 채움 라인이 다음 로크 라인과 동일한 공간에 겹치도록(coextensive) 상기 제1의 복수 래치 및 상기 제2의 복수 래치를 결합시키는 결합 로직(combinational logic)을 더 포함하는 장치.

### 청구항 7.

제1항에서,

상기 제1의 복수 래치 중 하나만이 3보다 큰 팬 아웃(fanout)을 갖는, 장치.

#### 청구항 8.

제1항에서,

상기 캐시는 TLB(translation lookaside buffer)인, 장치.

#### 청구항 9.

제1항에서,

로크되는 상기 라인은 이전 로크 레지스터 비트의 논리 1과 현재 비트의 논리 0의 동시 발생에 의하여 결정되는, 장치.

#### 청구항 10.

제1항에서,

상기 제1 시프트 레지스터는 원형 버퍼(circular buffer)로 구성되는, 장치.

#### 청구항 11.

제10항에서,

상기 제1 시프트 레지스터에서의 순환 시프트의 길이는 상기 제2 시프트 레지스터에 비트를 설정함으로써 결정되는, 장치.

#### 청구항 12.

제1항에서,

상기 각각의 래치는 마스터 슬레이브 플립플롭(master slave flipflop)을 포함하는 장치.

#### 청구항 13.

CAM(content addressable memory array),

상기 CAM에 연결된 RAM(random access memory), 그리고

상기 CAM에 연결되어, 상기 RAM의 언로크된 라인에 대하여 라운드 로빈 대체 기법을 구현하면서 상기 RAM의 소정 라인을 상기 CAM이 로크하게 하는 제어 로직(control logic)을 포함하는 시스템.

#### 청구항 14.

제13항에서,

상기 제어 로직은,

RAM 라인에 대한 채움 순서를 결정하는 데이지 체인(daisy chain)으로 함께 연결되며 각각이 상기 RAM 라인과 연관되는 제1의 복수 래치, 그리고

캐시 라인에 대한 로크 순서를 결정하는 데이지 체인으로 함께 연결되며 각각이 상기 캐시 라인과 연관되는 제2의 복수 래치를 포함하는, 시스템.

#### 청구항 15.

로크 시작 라인을 결정하는 단계,

상기 시작 라인에서 떨어진 채움 시작 라인을 결정하는 단계, 그리고

로크가 로드될 때마다 채움 포인터(fill pointer)를 상기 채움 시작 라인에 재설정하는 단계를 포함하는 방법.

## 청구항 16.

제15항에서,

로드되는 로크 라인에 응답하여 상기 채움 시작 라인에 더 가까운 라인을 지시하는 로크 지시기(lock indicator)를 변경하는 단계를 더 포함하는 방법.

## 청구항 17.

제15항에서,

제1 채움에서 상기 채움 시작 라인을 채우는 단계, 그리고

제2 채움에서 상기 로크 시작 라인에 가장 가까운 언로크된 라인을 채우는 단계를 더 포함하는 방법.

## 청구항 18.

실행시 디지털 처리 시스템이 방법을 수행하게 하는 실행 가능한 컴퓨터 프로그램 명령을 포함하는 컴퓨터로 판독 가능한 기억 매체로서,

상기 방법은,

캐시 라인에 데이터를 로드하는 캐시에 로크 모드를 초기 설정하는 단계,

상기 캐시 라인을 로크하는 단계, 그리고

상기 캐시를 채움 모드로 복귀시키는 단계

를 포함하는

컴퓨터로 판독 가능한 기억 매체.

## 요약

각각의 제1 복수 레지스터는 소정 캐시 라인과 연관되며 함께 데이터 체인된다. 유사하게, 각각의 제2 복수 레지스터는 캐시 라인과 연관되며 함께 데이터 체인된다. 제1 데이터 체인은 캐시 라인의 채움 순서를 결정하고 제2 데이터 체인은 로크 순서를 결정한다.

## 대표도

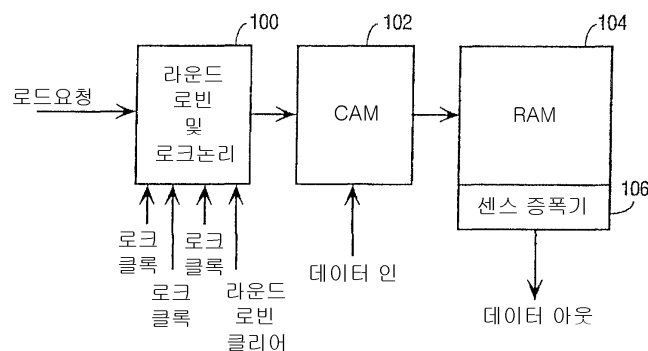
도 1

## 색인어

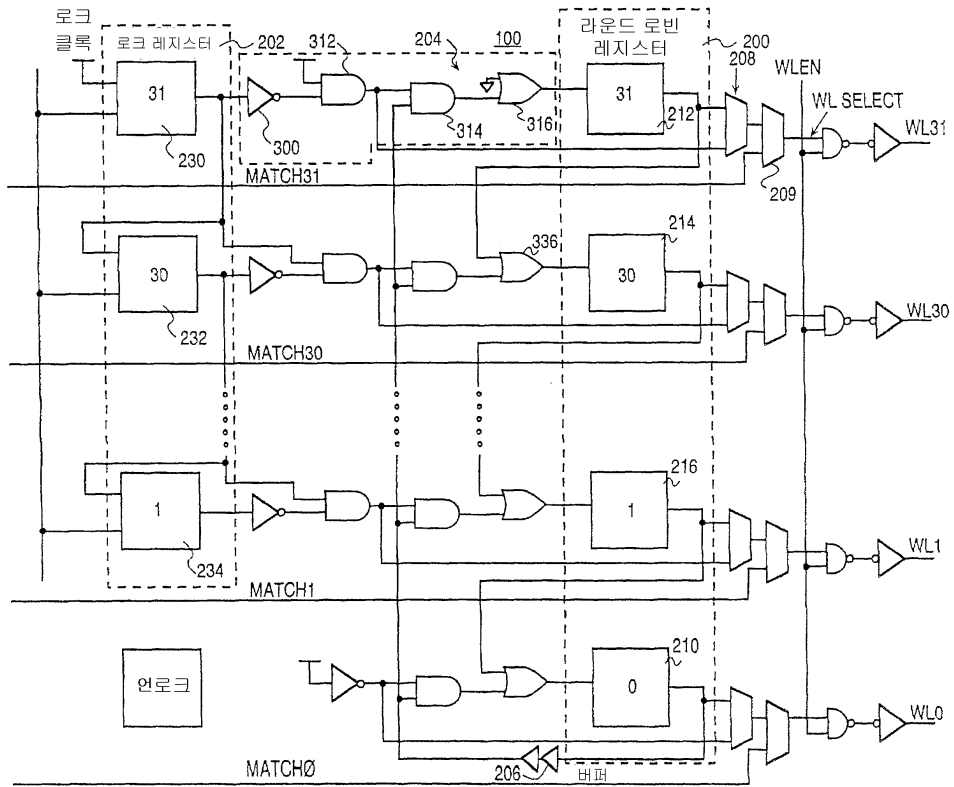
로크, 채움, 라운드로빈, 캐시, 대체, 세분성, 태그, 엔트리, CAM, 래치

## 도면

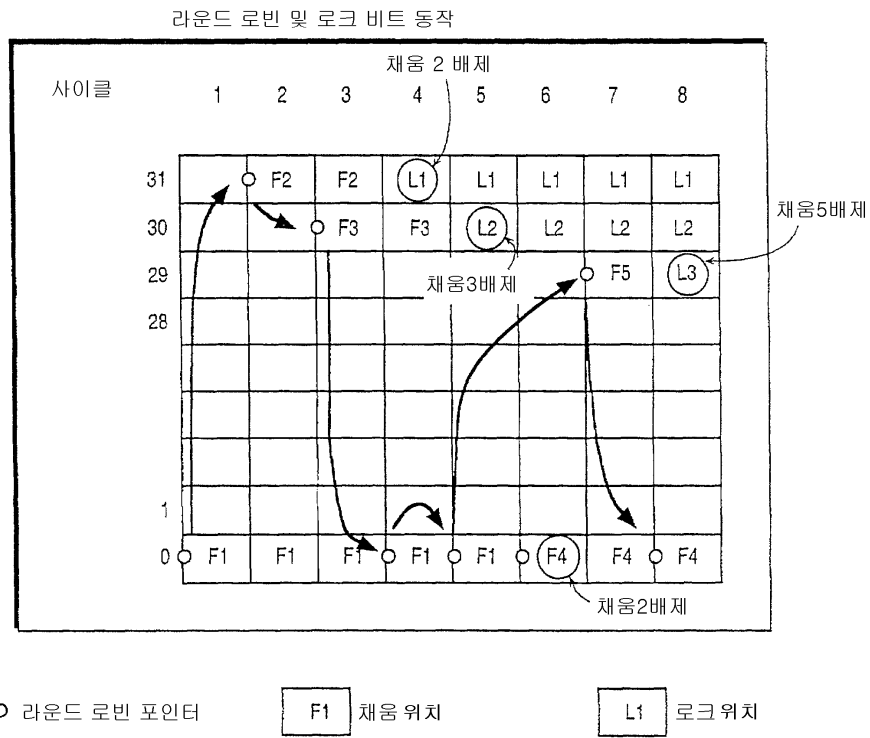
도면1



도면2



도면3



도면4

