

(19) 대한민국특허청(KR)
(12) 국제특허출원의 출원공개공보(A)

(51) Int. Cl.⁶
G06F 1/32
G06F 12/00

(11) 공개번호 특1998-7001105
(43) 공개일자 1998년04월30일

(21) 출원번호	특1997-7004313		
(22) 출원일자	1997년06월23일		
번역문제출일자	1997년06월23일		
(86) 국제출원번호	PCT/US 95/16601	(87) 국제공개번호	WO 96/32671
(86) 국제출원출원일자	1995년12월20일	(87) 국제공개일자	1996년10월17일
(81) 지정국	AP ARIP0특허 : 케냐 레소토 말라위 수단 스와질랜드 우간다		
	EP 유럽특허 : 오스트리아 벨기에 스위스 리히텐슈타인 독일 덴마크 스페인 프랑스 영국 그리스 아일랜드 이탈리아 룩셈부르크 모나코 네덜란드 포르투갈 스웨덴 핀란드		
	OA OAPI특허 : 부르키나파소 베냉 중앙아프리카 콩고 코트디부아르 카메룬 가봉 기니아 말리 모리타니 니제르 세네갈 차드 토고		
	국내특허 : 알바니아 아르메니아 오스트리아 오스트레일리아 바베이도스 불가리아 브라질 벨라루스 캐나다 스위스 중국 체코 독일 덴마크 에스토니아 스페인 핀란드 영국 그루지아 헝가리 아이슬란드 일본 케냐 키르기스탄 북한 한국 카자크스탄 스리랑카 라이베리아 레소토 리투아니아 룩셈부르크 라트비아 몰도바 마다가스카르 마케도니아 몽골 말라위 멕시코 노르웨이 뉴질랜드 폴란드 포르투갈 루마니아 러시아 수단 스웨덴 싱가포르 슬로베니아 슬로바키아 타지키스탄 투르크메니스탄 트리니다드투바고 우크라이나 우간다 우즈베키스탄 베트남		
(30) 우선권주장	8/363,735 1994년12월23일 미국(US)		
(71) 출원인	인텔 코퍼레이션 카알 실버맨		
(72) 발명자	미국 캘리포니아 95052 산타 클라라 미션 칼리지 불러바드 2200 칼미인 더글라스 엠. 미국 오리건 97007 비버톤 사우스 웨스트 보니 브래 14815 크로포드 존		
(74) 대리인	미국 캘리포니아 95051 산타 클라라 글로리에다 서클 2754 장용식, 정진상		

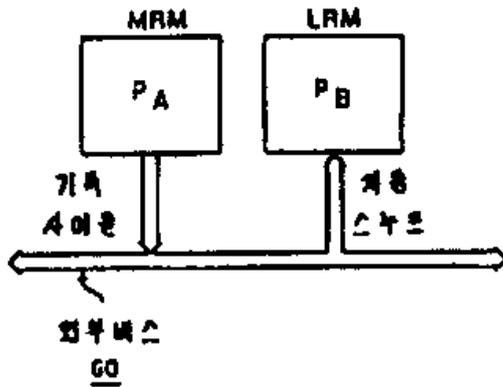
심사청구 : 없음

(54) 절전 동작 특성을 가진 캐시 코히어런트 멀티프로세싱 컴퓨터 시스템(A Cache Coherent Multiprocessing Computer System with Reduced Power Operating Features)

요약

멀티프로세싱 시스템은 절전 동작 모드 동안에 캐시 코히어런시를 유지해 준다. 상기 멀티프로세싱 시스템은 메인 메모리와 데이터 트랜잭션을 수행하기 위해 외부 버스(60)에 접속되어 있는 제1프로세서와 제2프로세서(20)를 가지고 있다. 상기 절전 동작 모드에서, 캐시 코히어런시를 유지하기 위해 상기 외부 버스(60)상에서의 트래픽을 감시 및 그 트래픽에 응답하는데 사용되는 상기 제2프로세서(20)의 내핵 로직의 일부분에 내부 물러 신호(CLK)가 접속되어 있는 동안에, 상기 제2프로세서(20)의 내핵 클럭 신호(ICK)는 상기 제2프로세서(20)의 내부 로직의 일부분으로부터 접속 해제된다. 상기 절전 동작 모드 동안에, 상기 제2프로세서(20)는 캐시 코히어런트 멀티프로세싱 시스템의 유지를 위해 계속하여 스누프를 수행하고 프로세스를 재기록 한다.

대표도



명세서

[발명의 명칭]

절전 동작 특성을 가진 캐시 코히어런트 멀티프로세싱 컴퓨터 시스템 (A Power Operating Features)Cache Coherent Multiprocessing Computer System with Reduced

[기술분야]

본 발명은 전반적으로 마이크로프로세서 기초 컴퓨터 시스템 분야에 관한 것으로, 특히 전력 소비를 조절하는 구성을 포함하고 있는 멀티프로세서 컴퓨터 시스템에 관한 것이다.

[배경기술]

반도체 프로세싱 기술의 진보에 의해 단일 반도체 기관상에 보다 많은 트랜지스터가 제조될 수 있도록 집적 회로의 구성 크기가 커져왔다. 예컨대, 최근에 제조되고 있는 가장 정교한 마이크로프로세서는 일반적으로 수백만개의 트랜지스터로 이루어진 단일의 집적 회로(IC)를 구비하고 있다. 이러한 놀라운 기술적 진보에 의해 최근의 새로운 컴퓨터 시스템의 성능과 데이터 처리 능력을 크게 증가시킬 수 있게 되었지만, 이러한 진보는 증가된 전력 소비를 수반하게 된다. 전력 소비가 증가하면 물론 상기 IC로부터 방출되어야 하는 열이 보다 많이 존재함을 의미한다.

과도한 전력 소비와 열 방출은 이제 컴퓨터 설계자가 직면하고 있는 중요한 문제가 되고 있기 때문에, 여러가지 절전 기술이 컴퓨터 시스템내의 전력 공급 전류 레벨을 최소화하기 위해 개발되어 왔다. 이들 기술의 대부분은 전력 보존이 필요할 때 상기 마이크로프로세서의 전원을 다운시키는 방법을 이용하고 있다. 하지만, 이 방법도 단점이 없는 것은 아니다.

예컨대, 협동하여 시스템 작업을 완료하는 2개 이상의 프로세서를 이용하고 있는 멀티프로세서(MP) 컴퓨터 시스템에서는 문제가 발생된다. - (예컨대, 그 현재 작업이 완료되거나 상기 마이크로프로세서가 비액티브상태이기 때문에) 하나의 마이크로프로세서의 전원이 다운되면, 상기 컴퓨터 시스템의 다른 마이크로프로세서가 시스템 버스상에서 데이터 트랜잭션(transaction)을 계속 수행할 수 있다. 문제는 전원 다운되거나 비액티브 상태인 마이크로프로세서에서 일부 버스 트랜잭션이 수정된 상태에서 저장된 데이터를 판독/기록할 수 있다는 점이다. 버스 동작을 감시하고 공유 메모리 위치를 갱신하기 위한 메카니즘이 존재하지 않으면, 데이터 코히어런스(data coherency)를 잃게 된다. 그러므로, MP 컴퓨터 시스템은 기존 데이터 액세스하려고 하는 버스 동작을 프로세서들이 인식 및 그에 응답할 수 있도록 해 주는 메카니즘을 필요로 하고 있다.

알 수 있는 바와같이, 본 발명은 절전 모드에서 동작하는 동안에 캐시 코히어런트를 유지하기 위해 개개의 프로세서가 버스 트래픽을 감시하는 멀티프로세싱 컴퓨터 시스템을 제공해 준다. 본 발명에 따라 비액티브 상태이거나 전원 다운된 프로세서는 절전 동작 모드에서 시스템 버스에 수정된 데이터를 재기록함으로써 특정 버스 트랜잭션에 응답한다. 또한, 본 발명은 운영 체제로부터의 대기시간 또는 개입없이도 기능한다. 따라서, 본 발명을 전력 소비를 최소화하면서 외부 버스와 상호작용하는 매우 투명한 방법을 제공해 준다.

[발명의 상세한 설명]

메인 메모리에 그리고 또한 제1 및 제2프로세서에 접속되어 있는 버스를 포함하고 있는 멀티프로세서 컴퓨터 시스템에 대해 설명한다. 상기 제1 및 제2프로세서는 상기 메인 메모리와 데이터 트랜잭션을 수행하기 위해 상기 버스에 접속되어 있다. 상기 제2프로세서는 상기 버스에 접속되어 있는 버스 유닛, 해당 태그 어레이(tag array)를 가지고 있는 로컬 캐시(local cache), 물리 신호를 발생하는 위상 동기 루프(PLL), 및 상기 버스 유닛, PLL 및 해당 태그 어레이가 상기 클럭 신호에 접속되어 있는 동안에 상기 제2

마이크로프로세서의 내부 로직(internal logic)의 일부분으로부터 상기 물려 신호를 접속 해제함으로써 전력 소비를 줄여 주는 수단을 가지고 있다. 상기 제2마이크로프로세서의 상기 버스 유닛은 또한 절전 동작 모드일 때 상기 버스를 스누핑(Sn70ping)하는 수단, 및 상기 제1마이크로프로세서에 의한 상기 버스 상의 기록 사이클이 상기 제7마이크로프로세서의 상기 로컬 캐시의 수정 캐시 라인에 대해 진행됨을 알리는 신호를 상기 제1마이크로프로세서7,7측으로 전송해 주는 수단을 더 포함하고 있다.

본 발명은 이하의 상세한 설명으로부터 그리고 첨부된 도면으로부터 보다 완전하게 이해할 수 있게 되지만, 본 발명은 도시된 특정 실시예에 한정되는 것으로 간주되어서는 안되며 단지 설명 및 이해를 위해 제공된 것에 불과하다.

[도면의 간단한 설명]

도 1은 본 발명에 따라 동작하는 마이크로프로세서의 일반화된 블록도. 도 2는 본 발명의 일실시예에 사용된 클럭 제어 구조의 상태 전이를 보인 도면. 도 3은 본 발명의 일측면의 동작을 보인 타이밍도. 도 4a-도 44는 본 발명의 일실시예에서 수행되는 여러 예시적 동작을 나타낸 한 세트의 도면. 도 j는 본 발명에 따른 특정 구성을 소프트웨어가 디스에이블시킬 수 있도록 해 주기 위한 비트를 포함하고 있는 테스트 레지스터를 나타낸 도면.

[실시예]

본 발명은 전력 소비를 최소화하면서 캐시 코히어런시를 유지해 주는 멀티프로세서 컴퓨터 시스템이다. 이하의 설명에서, 본 발명의 철저한 이해를 제공하기 위해 특정 신호, 프로토콜, 디바이스 종류와 같은 여러특정 상세 내용이 설명된다. 하지만, 본 발명을 실시하는데에는 이들 특정 상세 내용을 사용할 필요는 없음은 물론이다. 다른 예에서는 잘 알려진 구조, 회로, 블록, 및 구성은 본 발명이 불필요하게 불명료해지지 않도록 도시되지 않는다.

도 1은 본 발명의 여러 구성을 포함하고 있는 마이크로프로세서(70)의 블록도이다. 일실시예에서, 상기 마이크로프로세서(20)는 미국 캘리포니아, 산타 클라라 소재의 인텔 코퍼레이션에 의해 제조된, 시중에서 구할 수 있는, 개선된 Pentium'을 구비하고 있다. 본 발명이 도 1의 실시예가 더불어 절명되게 되지만, 상기 펜티엄 프로세서와 호환이 가능한 마이크로프로세서, 즉 상이한 데이터 프로세서 구조를 이용하는 마이크로프로세서를 포함해서, 단 발명의 넓은 개념이 다수의 상이한 종류의 컴퓨터 시스템에 적당가능함은 물론이다.

마이크로프로세서(20)는 위상 동기 루프(PLL) 회로(30)를 포함하고 있으며, 이 회로는 라인(46)을 따라 접속되어 있는 내부 클럭 신호(ICLK)를 상기 집적 회로의 각종 기능 유닛 블록에 제공한다. 예컨대, 상기 라인(46)상에 존재하는 상기 ICLK 신호는 도 1에 도시된 바와같이, 명령 캐시(25), 데이터 캐시(26), 게이트 ROM(21), 및 상기 프로세서의 정수 유닛(integer unit)(22)과 부동 소수점(FP) 유닛(23)을 구비하고 있는 내부 로직을 포함해서 상기 IC의 대다수의 내부 로직에 접속되어 있다. 상기 ICLK 신호는 또한 상기 인터럽트 로직 유닛(29)의 일부분에 접속되어 있다. 명령 캐시(25), 데이터 캐시(26) 및 버스 유닛(40)은 각각 64 비트 데이터 버스(42)와 32 비트 주소 버스(43)에 접속되어 있다.

상기 ICLK 신호 이외에, 상기 PLL 회로(30)는 또한 라인(45)을 따라 접속되어 있는 제2클럭 신호(CLK)를, 명령 캐시(25)와 데이터 캐시(26)로 된 태그 비교 어레이측에 제공해 준다. 상기 라인(45)상의 상기 CLK 신호는 또한 인터럽트 논리 유닛(29)과 버스 유닛(40)의 일부분에 접속되어 있다. (상기 ICLK 물려 신호와 CLK 물려 신호간의 구별에 대해서는 보다 상세히 후술한다.) 정상 동작 동안에 마이크로프로세서(20)는 대략 100 MHz의 클럭(즉, ICLK,CLK) 주파수로 동작한다. 예컨대, 다른 구현예에서는, 50MHz와 60MHz의 버스주파수가 지원된다.

일실시예에서, 버스 유닛(40)과 인터럽트 논리 유닛(29)은 MP 시스템에 마이크로프로세서(20)를 사용할 수 있도록 해주는 논리 회로를 포함하고 있다. 예컨대, 인터럽트 논리 유닛(29)에 의해 수신 및 제공된 인터럽트 신호 뿐만 아니라, 버스 유닛(40)에 의해 발생 및 수신된 버스 사이클은 잘 알려진 Pentium' 프로세서와 호환가능하다. 마이크로프로세서(20)는 또한 캐시 일관성(cache consistency)을 유지해 주고 외부 버스에 대해 중재를 행하는 로직을 포함하고 있다.

일실시예에서, 마이크로프로세서(20)는 또한 절전 모드 상태(즉, HALT 또는 STANDBY 동작 모드)로 진입할 수 있도록 해 주는 전력 관리 구성을 포함하고 있다. 마이크로프로세서(20)는 정상 동작 모드에서 피크 전력으로 동작하고 있을 때에는 10 와트 정도도 많은 전력을 소비할 수 있고, HALT 또는 STANDBY 동작 모드에서는 700 밀리와트 정도의 적은 전력을 소비할 수 있다. 운영 체제 독립 기능 뿐만 아니라 전력 할력 기능을 마이크로프로세서(20)가 구현하는 방법중의 하나로는 시스템 관리 동작 모드를 이용하는 방법을 들 수 있다. 시스템 관리 모드(SMM)는 인터럽트(SMI), 다른 주소 공간 및 명령(SRET)으로 구성되어 있다. 상기 시스템 관리 인터럽트는 시스템 관리 인터럽트 요구가 논리 유닛(29)내에 래치될 수 있도록 해 준다. 상기 래치(SMI#)가 명령 경계(instruction boundary)에서 인식될 때, 마이크로프로세서(20)는 SMM으로 진입한다.

도 1을 계속 참조하면, 일실시예에서, 마이크로프로세서(20)는 이 마이크로프로세서의 전력을 조절하는데 사용될 수 있는 외부 핀(STPCLK#로 표시되어 있음) 및 해당 회로를 포함하고 있다. 상기 STPCLK#핀에 대해서는 1952년 3월 11일에 출원되어 함께 계류중인, 발명의 명칭이 '프로세서의 클럭을 비동기적으로 정지시키는 방법 및 장치'인 미국 특허 출원 제07/970,576호에 보다 충분히 설명되어 있고, 이 출원은 본 발명의 양수인에게 양도되어 있으며, 본 명세서에서는 이를 참조 문헌으로서 포함하고 있다. 상기 STPCLK# 핀은 상기 프로세서를 저전력 STANDBY 상태로 설정해 주는 레벨 감지 르MI를 제공한다. 본 발명에 따라, 마이크로프로세서(20)는 STANDBY 동작 모드 동안에 전용 요구(즉, ADS#)와 외부 스누프(external sn77p) 요구(즉, EADS#)에 응답한다.

프로세서(20)는 또한 자동 정지(AUTO_HALT)라고 하는 메카니즘을 지원하고 있는데, 이 메카니즘은 HALT 명령이 실행될 때마다 상기 프로세서를 저전력 동작 모드로 설정한다. 마이크로프로세서(20)는 (STPCLK#핀에의 공급없이) 저전력 상태에 있는 동안에 발생된 전용 스누프 및 프로세서간 인터럽트 사건을 포함해

서 모든 정지 브레이크 사건(stop break event)에 응답한다.

마이크로프로세서(20)가 정상 동작 모드에서 동작할 때에는, 두 물려 라인(46(ICLK),45(CLK))이 인에이블되며, 이에따라 내부 코어(ore) 클럭 신호가 상기 집적 회로의 모든 유닛에 제공되게 된다. 상기 STPCLK 살 편의 제공 또는 HALT 명령의 실행의 결과로서 상기 프로세서가 저전력 상태 또는 STANDBY 상태로 진입하면, 라인(46)상의 내부 플럭 신호(ICLK)가 디스에이블된다. 하지만, 라인(46)이 디스에이블되더라도 PLL 회로(30)의 연속 동작에는 변동이 없음은 물론이다. 환인하면, PLL(권1은 마이크로프로세서(20)의 특정 부분에 접속되어 있는 라인(45)상에서 내부 코어 CLK 주파수를 계속적으로 발생한다. 동작을 유지하는(즉, CLH에 의해 액티브된) 마이크로프로세서(27)의 상기 부분은 PLL(30), 캐시 유닛(25,26)의 태그 비교 어레이, 인터럽트논리 유닛(29)의 일부분, 및 버스 유닛(40)을 포함하고 있다. 본 발명에 따라, 칸이 마이크로프로세서(27)의 상기 선택된 부분에 전원을 제공하면, MP 시스템에서 캐J.1 코히어런시를 유지하기 위해 상기 프로세서는 외부 버스 트래픽을 감시 및 그 트래픽에 응답할 수 있게 된다. 즉, 버스 중재 및 캐시 코히어런시 메카니즘은 절전 동작 모드에서 의도적으로 액티브 상태로 유지된다.

이제, 도 2를 참조하면, 본 발명의 일실시예의 중요한 5개의 특정 구성을 나타낸 상태도가 도시되어 있다.

정상 동작 상태(상태 51)에서, 마이크로프로세서(20)는 명령을 수행하기 위해 액티브 상태로 된다. 이는 ICLK 신호와 CLK 신호가 상기 IC의 내부 로직의 모든 부분에 접속되어 있는 프로세서의 전체 전력 상태를 나타낸다.

상기 마이크로프로세서(20)의 외부 STPCLK# 핀을 구동함으로써 블록(53)에 의해 표기된 STOP_GRANT상태가 될 수 있다. 상기 STOP_GRANT 상태에서, 상기 집적 회로는 절전 모드에서 동작하며, 이 모드에서 상기 프로세서의 상기 내부 기능 유닛의 대부분이 디액티브된다(즉, ICLK이 오프된다). 한편, 버스 유닛(40), 일단, 상기 STOP_GRANT 버스 사이클이 상기 버스상에 배치되고 BRDY# 신호가 복귀되면, 상기 프로세서는 상기 STOP_GBIANT 상태로 된다. 당업자는 상기 외부 시스템이 판독에 응답하여 상기 데이터 핀상에 유효 데이터를 제공할 수 있다. 즉 상기 외부 시스템(예컨대, 메인 메모리, 기타 다른 프로세서 등)이 기록 요구에 응답하여 프로세서 데이터를 받아들임에 지시하기 위한 것으로 상기 BRDY# 신호를 이해하고 있다. 일실시예에서, 상기 프로세서는 STPC17#이 공급된 후에 대략 10개의 물려 기간에 정상적인 실행 상태로 복귀한다.

또한, RESET에 의해, 상기 프로세서는 STOP_GRANT 상태에서부터 빠각나각 나와 정상 상태로 복귀하게 된다.

상기 프로세서(20)는 절전 동작 모드에서 버스 트래픽을 감시함으로써 캐시 코히어런시(예컨대, 유효화 및 외부 스누프)를 유지하기 위해 버스 유닛(40)에 대한 입력을 인식하게 된다. 예컨대, STOP_GRAOIT 상태에서, 상기 프로세서는 외부 인터럽트 신호(예컨대, SMI#, NMI, INTR, FLUSH#, R/S#, ININT)의 천이를 래치한다. 이들 인터럽트 모두는 STPCLK#의 공급 중단후에 즉 상기 정상 동작 상태로의 재진입시에 행해진다.

도 2에 블록(50)에 의해 표시되어 있는 AUTO_HALT 상태는 HALT 명령이 실행될 때마다 진입된다.

AUTO_HALT 상태에서, 상기 내부 블록(1CMK)은 대다수의 상기 내부 로직으로부터 접속 해제되는 반면에 상기 연속적인 클럭 신호(CLK)는 칸의 선택된 기능 유닛을 동작 상태로 유지한다. 상기 INTR, NMI, SMI#, RESET 또는 INIT의 발생시에, 상기 프로세서는 경상 동작 상태, 즉 상태 히로 다시 천이된다. 이전에 설명한 바와같이, 상기 프로세서의 대부분의 내부 로직으로의 강기 잘려 신호의 공급을 정지시킴으로써, 상기 AUTO_HALT 상태에 의해 전력이 크게 감소된다. 상기 버스 유닛(40), PU(30), 태그 어레이, 및 인터럽트 로직은 스누프를 지원하기 위해 그리고 고속의 재시락을 가능하게 하기 위해 액티브 상태로 유지된다. 어떤 외부 인터럽트에 의해, 상기 프로세서는 ALTT0_H.7LT 강태(교)를 라격나와 경상 동작 상태(51)로 복귀할 수 있다. AUTO_HALT 상태(50)에서 동작할 때, FLUSH# 신호(7)는 캐시(보7)의 모든 수경 라인을 개기록할 수 있고 내부 캐시를 무효화시킬 수 있다. 이때, FLUSH 확인 특수 신호가 재기록 동작 및 무효화 동작의 완료를 지시하기 위해 상기 프로세서에 의해 탄생된다.

상기 R/S# 입력(액티브 로우)의 공급에 의해, 상기 프로세서는 정상적인 실행을 정지하고 IDLE 상태로 설정될 수 있다. 도 2에는 프로브 모드(PROBB_MODE) 상태(57)로 진입하기 위해 AUTO_HALT 상태(50)로부터의 천이를 일으키는 R/S# 사건이 예시되어 있다. 특수 디버그 포트와 퍼블어 상기 프로세서의 디버깅에 사용되는 R/S# 핀이 제공되어 있다. 상기 R/S# 핀의 하이에서 로우로의 천이에 의해, 상기 프로세서는 인터럽트되고 다음 명령 경계에서 실행을 중지하게 된다. 상기 R/S# 핀의 제공 중단에 의해, PROBB_MODE 상태(56)로부터 AUTO_HALT 상태(50)로 다시 천이되게 된다.

도 2에는 또한 상기 STPCLK# 핀의 공급 또는 HALT 명령의 실행 후에 있을 수 있는 경우인 절전 모드에서 동작하고 있을 때의 프로세서의 스누핑 능력을 나타내는 상태(52)가 포함되어 있다. 전력 손실이 이들 두 상태중 어느 한 상태에서 감소되더라도, 상기 프로세서는 유닛(40)을 통해 버스 신호를 계속적으로 구동하며, 이에따라 상기 프로세서의 내부 머신 상태가 유지된다. 전용 스누프 및 프로세서간 스누프가 캐시 무효화 및 재기록 사이클을 위해 지원된다. 전력 소비는 상기 스누프가 재기록 사이클의 진행을 필요로 하는 경우에 짧은 기간동안 생성된다. 본 발명에 따라, 스누프는 마이크로코드 개입의 필요없이 하드웨어에 의해 완전 하게 처리되게 된다.

상기 STOP_GRANT 상태와 AUTO_HALT 상태에서, 마이크로프로세서(20)는 CLK에 의해 인에이블되는 캐시 유닛 태그 비교 로직을 유지함으로써 전용 스누프 및 프로세서간 스누프를 포함한 스누프를 지원한다.

이는 도 1에 예시되어 있으며, 이때, 라인(45)상의 상기 CLK 신호는 캐시(2577)의 태그 비교 로직에 접속되어 있다. 한편, 상기 ICLK 신호는 전력 소비가 최소화되도록 상기 캐시의 데이터 어레이 부분에 대해 디스에이블된다. 스누프가 발생되면, MBSI(즉, 수정, 실행, 공유, 무효) 캐시 프로토콜 비트가 필요할 때 갱신된다.

상기 태그 어레이 이외의 상기 프로세서의 부분들(예컨대, 적어도 데이터 캐시, 해당 캐시 제어 로직, 및 상기 캐시와 버스 인터페이스간의 버스)은 재기록 사이클이 필요할 때에만 동작한다. 다른 실시 예에서, 보다 적극적인 절전은 특정 상황에서 상기 태그 비교 로직을 디스에이블시킴으로써, 예컨대 모든 캐시 엔트리를 무효화 시킴으로써, 즉 스누프가능 트랜잭션이 감출될 때까지 상기 태그 어레이의 전원을 다운시킨 후에 상기 태그가 스누프 동작을 수행할 수 있도록 약간 증가된 전력 상태를 제공함으로써, 탄생된다.

도 1 내지 도 4에는 본 발명의 한가지 예시적인 실시예만이 예시되어 있지만, 상이한 각종 물러 기술을 포함해서 각종 절전 기술을 제공하기 위한 보다 복잡한 구현애가 가능함에 주의하자. 일단 스누핑이 수행되면 이용되는 특정 프로토콜에 대해서도 마찬가지이다. 예컨대, 제1프로세서로 하여금 상기 시스템 버스 상에 데이터를 재기록하도록 하는 대신에, 상기 제1프로세서는 기록을 흡수하고 더티 라인(dirty line)을 유지할 수도 있다. 따라서, 당업자는 본 발명을 구현할 수 있는 다수와 상이한 방법을 만 수 있다.

도 3에는 STPCLK와 요구와 STOP_GRANT 버스 사이 물간의 대기시간을 보인 타이밍도가 예시되어 있다.

예시된 실시예의 경우에는 STPCLK# 요구와 STOP_GRANT 버스 사이 끝간에 대략 10개의 플러의 지연이 존재함에 주의하자. 이 대기시간은 현재의 명령, CPU 기록 버퍼의 데이터량, 및 시스템 메모리 성능에 따라 좌우된다.

프로세서(20)의 데이터 캐시(26)는 캐시 일관성을 수행하는데 MBSI 프로토콜은 이용한다. 상기 데이터 캐시의 라인은 수정 상태, 독점 상태, 공유 상태, 또는 무효 상태일 수 있는 반면에, 명령 캐시(25)의 라인은 유효 상태 또는 무효 상태일 수 있다. 본 발명은 컴퓨터 시스템에서 둘 이상의 프로세서가 공통 데이터를 공유하고 있는, 발생가능한 상황을 감독하게 된다. 상기 프로세서의 로컬 캐시는 가능하면 언제나 데이터 캐시를 시도할 수 있다. 프로세서(20)의 일 실시예에서, 프로세서들간에 데이터 일관성을 보장해 주기 위해 전용캐시 코히어런트 메카니즘이 포함되어 있다. 어떤 데이터가 프로세서중 한 프로세서내로 캐시되고 다른 프로세서가 그 데이터에 대한 액세스를 시도하면, 그 데이터를 포함하고 있는 상기 프로세서는 상기 데이터가 캐시되었음을 요구 프로세서측에 통지해 준다. 상기 캐시 라인과 상기 데이터를 포함하고 있는 프로세서의 상태는 현재 상태, 및 다른 프로세서가 행한 요구의 종류에 따라 변한다.

본 발명에 따라, 기본 코히어런트 메카니즘은 사이클을 진행하고 있지 않고 버스(이하, 가장 오래된 버마스터, 즉 LRM라고 함)를 소유하고 있지 않은 프로세서가 모든 LRM 버스 동작(상기 버스를 소유하고 있지 않은 가장 최근의 버스 마스터 라고 하는 MR커)을 스누핑할 것을 요구하게 된다 이때, 버스 사이클을 진행하고 있는 상기 MRM 프로세서는 상기 데이터가 상기 LRM 캐시에 포함되어 있음을 지시하기 위해 상기 LRM 프로세서를 감시한다.

본 발명의 동작을 보다 잘 이해할 수 있도록 도 4a-도 44에 표시되어 있는 중재 인터페이스 교편예를 고려 하자, 먼저, 프로세서(Pa)가 절전 모드로 동작하고 있음을, 즉 외측 버스(60)에 대해 아이들 상태인 것으로 가정하자. 이는 프로세서(Pb)가 AUTO_HALT 상태 또는 STOP_GRANT 상태인 경우에 해당한다. 또한, 상기 컴퓨터 시스템의 프로세서(f)가 외부 버스(60)상에서 기록 사이클을 진행하고 있다고 가정하자. 또한, 프로세서(Pb)에서 상기 버스상의 기록 사이클이 수정 상태(보)인 라인에 대해 진행중인 것으로 가정하자. 이는 도 4a에 도시된 상황에 해당한다. 다른 내부 로직의 대부분의 전원이 다운됨에도 불구하고, 프로세서(Pe)에서 상기 유닛과 상기 내부 인터럽트 및 스누프 로직은 액티브 상태를 유지하고 있으므로, 프로세서(PA)에 의해 시작된 상기 기록 사이클을 감시하기 위해 프로세서(Pa)는 외측 버스(60)를 자동적으로 스누핑하게 된다.

도 4b에는 기록 트랜잭션이 수정 상태 캐시 라인을 적중하였음을 프로세서(P_A)측에 지시해 주는 프로세서(P_B)가 도시되어 있다. 이는 프로세서(P_B)에 접속되어 있는 PHIMT# 신호를 공급함으로써 행해진다. 프로세서(P_A)는 또한 자신이 버스 요구를 발생하였음을 지시해 주기 위해 전강 중재 핀, 즉 BPREQ#를 구동한다. (이 예에서, 프로세서(P_A)는 현재 외측 버스(60)를 소유하고 있는 것으로 가정함). 또한, 상기 라인이 완전하게 재기록될 때까지 다른 버스 마스터가 상기 데이터를 액세스하지 않도록 하기 위해, HITM#이 상기 버스(60)에 접속되어 있다. 이때, 프로세서(PA)는 프로세서(Pa)가 존재하지 않는 것처럼 외부 버스(60)상에서 기록 사이클을 완료하게 된다.

상기 외부 스누프는, 상기 기록 사이클이 버스(60)상에서 완료된 후에, 하지만 프로세서(Pe)가 시스템 메모리에 상기 수정 데이터를 재기록할 수 있는 기회를 얻기 전에, 바로 발생된다. 도 4b에는 상기 스누프 주소가 상기 이중 프로세서 쌍내에 캐시되어 수정 상태임을 상기 시스템측에 안리기 위해 상기 HITM갈 신호를 공급하는 프로세서(Pa)가 예시되어 있다.

도 4c에서, 중재 교환이 외부 버스(6)상에서 발생하였으므로, 프로세서(Pa)는 이제 상기 버스를 소유하게 된다. 이때, 상기 프로세서(Pa)는 M 상태 라인을 재기록한다. 상기 컴퓨터 시스템의 관점에서는, 단일의 프로세서가 상기 스누프 트랜잭션을 완료한 것처럼 보인다. 도 4c에서, 상기 두 프로세서사이에 접속되어 있는 두 전용 중재 핀은 버스 소유권이 프로세서(P7)에게 승인되었음(또는, 상기 재기록 동작의 완료 후에 프로세서(PA)가 소유권을 다시 요구함)을 지시하는데 이용된다.

마지막으로, 도 4d에서 프로세서(P_B)가 프로세서(P_A)측에 버스를 다시 승인해 준 후에 프로세서(P_A)는 원래의 기록 사이클로 복귀한다. 상기 프로세서(Pe)가 도 4a-도 44에 의해 표시된 전체 스누프 및 재기록 처리를 통해 절전 동작 모드를 유지하고 있음을 인식하는 것이 중요하다. 이는 본 발명의 중요한 특성으로 볼 수 있는데, 그 이유는 본 발명이 전력 소비 요건에 의해 제한된 컴퓨터 시스템에 상당한 이점을 제공해 주기 때문이며, 하지만 이때 캐시 코히어런스는 여전히 필요하다.

이제, 도 5를 참조하면, 마이크로프로세서(20)의 특정 구성을 소프트웨어가 디스에이블시킬 수 있도록 해 주는 비트를 포함하고 있는 특수 테스트 레지스터(12)가 도시되어 있다. 예컨대, 상기 AUTO_HALT 구성은 레지스터(12)의 비트 6의 '1' 로의 설정에 대해 디스에이블될 수 있다. 이 선정에서, HALT 명령의 실행은 상기 프로세서의 어떤 기능 유닛에 대해 내부 물러(1CLK)을 디스에이블시키지 않는다. 일 실시예에서, 상기 AUTO_HALT 구성은 디플트에 의해 인에이확되는데, 즉 레지스터(12)의 비트 6은 RESET후에 '0' 으로 설

정된다.

테스트 레지스터(12)를 통해 소프트웨어에 의해 제어된 다른 구성으로는 멀티프로세서 시스템용의 진보된 프로그래머용 인터럽트 제어(APIC)를 들 수 있다. 본 발명의 일실시예에서, 상기 프로세서는 간판한 단일 프로세서 환경에서의 인터럽트 뿐만 아니라, 복잡한 멀티프로세서 환경에서의 인터럽트달 지원하는 진보된 프로세서 SMI 제어기를 포함하고 있다. 상기 로컬 인터럽트 제어기 모듈은 3 와이어 직렬 버스상에서 I/O APIC 모듈(예컨대, 인텔 코퍼레이션에서 제조한 부품 번호 8259A)과 통신한다. 레지스터(12)의 비트 4가 '1'로 세트되면, 상기 APIC 구성은 완전하게 디스에이할된다. 이는 상기 APIC 회로가 프로세서간 인터럽트를 송수신할 수 없음을 의미한다. 상기 APIC 레지스터 공간에 대한 판독 또는 기록은 상기 외부 버스측으로 전송된다. 또한, 이중 프로세서 중재 하드웨어는 레지스터(12)의 비트 5가 세트될 때 디스에이될 수도 있다. 이 비트가 '1'로 설정되면, 전용 이중 프로세서 구성(예컨대, 만717탈 핀, PHITM# 핀, PBREQ# 핀, PBGRNT핀)은 상기 프로세서가 가장 최근의 버스 마스터(MRM)로 되자마자 디스에이할된다. 따라서, 이 비트가 '0'으로 설정되면, 상기 DP 구성이 다시 인에이올된다. 레지스터(12)에 의해 인에이블/디스에이할될 수 있는 기타 다른 구성은 SMM 입력 및 출력 메시지(비트 7), 및 고속 실행 트레이싱(tracing) 메시지(비트 8)를 포함하고 있다.

(57) 청구의 범위

청구항 1

멀티프로세서 컴퓨터 시스템에 있어서, 버스: 상기 버스에 접속되어 있는 메인 메모리: 상기 버스에 접속되어, 상기 메인 메모리와 데이터 트랜잭션을 수행하는 제1마이크로프로세서: 상기 버스에 접속되어 있는 버스 유닛, 해당 태그 어레이를 가지고 있는 로컬 캐시, 물려 신호를 발생하는 위상 동기 루프(PLL), 및 상기 버스 유닛, PLL 및 해당 태그 어레이가 상기 클럭 신호에 접속되어 있는 동안에, 상기 제2마이크로프로세서의 내부 로직의 일부분으로부터 상기 클럭 신호를 접속 해제함으로써 전력 소비를 줄여 주는 수단을 가지고 있는 제2마이크로프로세서를 구비하고 있고, 상기 제2마이크로프로세서의 상기 버스 긴닛은 절전 동작 모드일 때 상기 버스를 스누핑하는 수단, 및 상기 제1마이크로프로세서에 의한 상기 버스상의 기록 사이클이 상기 제 2마이크로프로세서의 상기 로컬 캐시내의 수정 캐시 라인에 대해 진행중임을 알리는 신호를 상기 제1마이크로프로세서측에 제공하는 수단을 더 구비하고 있는 것을 특징으로 하는 멀티프로세서 컴퓨터 시스템.

청구항 2

제 1 항에 있어서, 상기 제2마이크로프로세서는 이 제2마이크로프로세서의 기록 사이클의 완료 후에 상기 버스상의 상기 메인 메모리측에 상기 수정 캐시 라인을 재기록하는 수단을 더 구비하고 있는 것을 특징으로 하는 멀티프로세서 컴퓨터 시스템.

청구항 3

제 2 항에 있어서, 상기 제1마이크로프로세서는 이 제1마이크로프로세서의 신호 제공 수단에 응답하여 상기 제2마이크로프로세서가 상기 수정 캐시 라인의 재기록을 완료한 후에 기록 사이클을 다시 발생하는 수단을 더 구비하고 있는 것을 특징으로 하는 멀티프로세서 컴퓨터 시스템.

청구항 4

제 1 항에 있어서, 접속 해제과는 상기 수단은 마이크로코드 정지 명령을 구비하고 있는 것을 특징으로 하는 멀티프로세서 컴퓨터 시스템.

청구항 5

제 1 항에 있어서, 접속해제하는 상기 수단은 필요할 때 내부 로직의 일부분으로부터 상기 물려 신호가 접속 해제되도록 하는 외부 핀을 구비하고 있는 것을 특징으로 하는 멀티프로세서 컴퓨터 시스템.

청구항 6

멀티프로세서 컴퓨터 시스템에 있어서, 외부 버스: 상기 외부 버스에 접속되어 있는 제1 및 제2마이크로프로세서: 및 상기 제1 및 제2마이크로프로세서가 상기 외부 버스의 소유권을 중재할 수 있도록 해 주는 인터럽트 제어 유닛을 구비하고 있고, 상기 제1마이크로프로세서는 적어도 버스 유닛과 데이터 캐시의 태그 어레이에 물려 신호를 여전히 제공해 주면서 대다수의 내부 로직측으로의 클럭 신호의 공급을 중지함으로써, 상기 제1마이크로프로세서가 절전 모드에서 동작할 수 있도록 해 주는 명령 수단을 포함하고 있으며, 상기 제1마이크로프로세서는 제1비트를 가지고 있는 레지스터로서, 상기 제1비트가 세트될 때 절전 모드가 동작으로부터 디스에이올되도록 소프트웨어에 의해 판독/기록되는 레지스터를 또한 포함하고 있는 것을 특징으로 하는 멀티프로세서 컴퓨터 시스템.

청구항 7

제 6 항에 있어서, 상기 레지스터는 세트시에 중재 메카니즘을 디스에이할시키는 제 2 비트를 더 포함하고 있는 것을 특징으로 하는 멀티프로세서 컴퓨터 시스템.

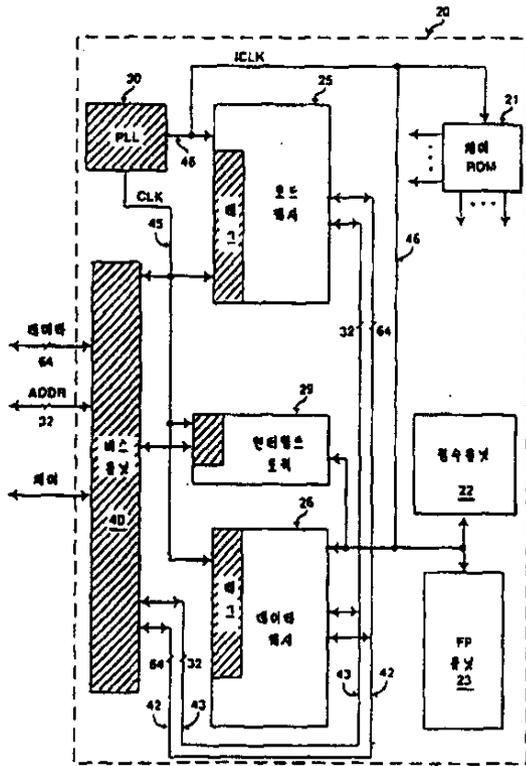
청구항 8

제 7 항에 있어서, 상기 레지스터는 세트시에 상기 인터럽트 제어 단닛을 디스에이블시키는 제 3 비트를 더 포함하고 있는 것을 특징으로 하는 멀티프로세서 컴퓨터 시스템.

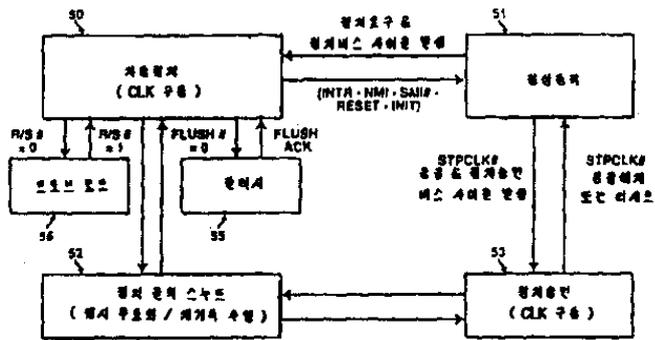
※ 참고사항 : 최초출원 내용에 의하여 공개하는 것임.

도면

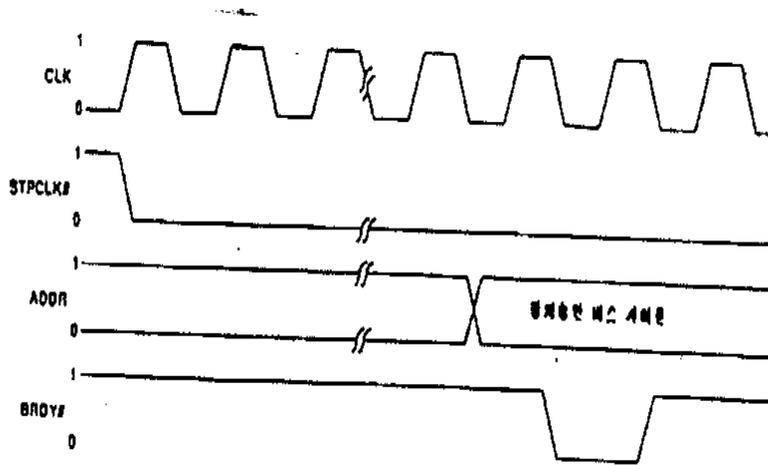
도면1



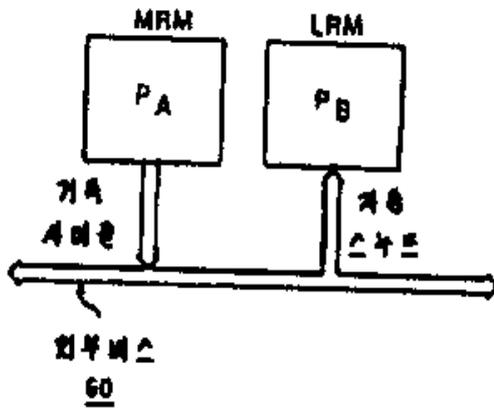
도면2



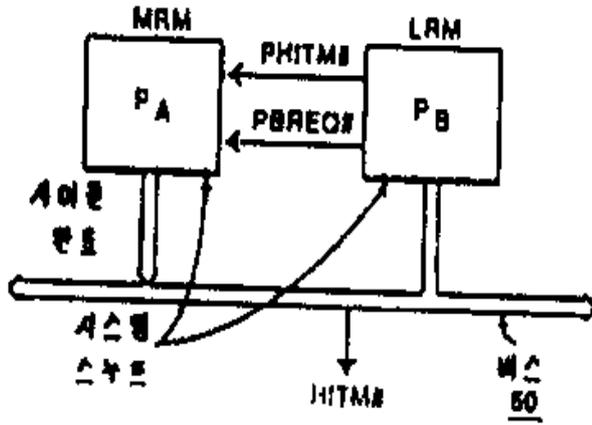
도면3



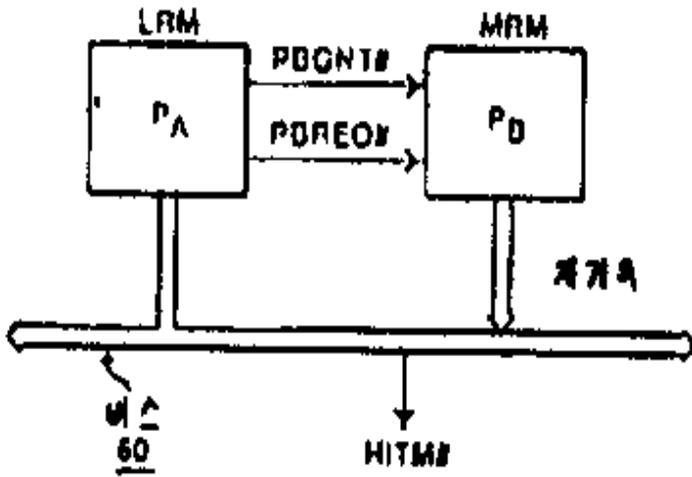
도면4a



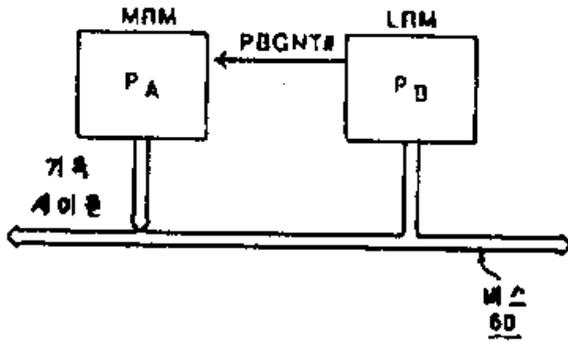
도면4b



도면4c



도면4d



도면5

