

(19) 대한민국특허청(KR)  
(12) 특허공보(B1)

(51) Int. Cl.<sup>6</sup>  
G06F 15/16

(45) 공고일자 1996년09월23일  
(11) 공고번호 특1996-0012660

(21) 출원번호	특1992-0024699	(65) 공개번호	특1993-0016885
(22) 출원일자	1992년12월17일	(43) 공개일자	1993년08월30일
(30) 우선권 주장	816,116 1992년01월02일 미국(US) 인터내셔널 비지네스 머신즈 코포레이션 존 디. 크레인 미합중국 뉴욕 10504 아몬크		
(73) 특허권자	미합중국 뉴욕 10504 아몬크		
(72) 발명자	나더 아미니 미합중국 플로리다 33434 보카 래턴 노스 웨스트 30 스트리트 2878 베차라 포아드 보리 미합중국 플로리다 33434 보카 래턴 노스 웨스트 28 애비뉴 3008 리차드 루이스 훈 미합중국 플로리다 33437 보인턴 비치 아파트먼트 넘버 8-23 세다레이크 로드 5289 테렌스 조셉 로만		
(74) 대리인	김창세, 김영, 장성구		

**심사관 : 박제현 (책자공보 제4644호)**

**(54) 컴퓨터 시스템 및 입력/출력 장치와 중앙 처리 장치간의 중재방법**

**요약**

요약없음

**대표도**

**도1**

**명세서**

[발명의 명칭]

컴퓨터 시스템 및 입력/출력 장치와 중앙 처리 장치간의 중재방법

[도면의 간단한 설명]

제1도는 본 발명의 원리에 따라 구성한 버스 인터페이스 유닛을 갖춘 컴퓨터 시스템의 개략 블록도.

제2도는 제1도에 도시한 컴퓨터 시스템의 버스 인터페이스 유닛의 개략 블록도.

제3도는 제2도에 도시한 버스 인터페이스 유닛의 FIFO 버퍼에 대한 개략블록도.

제4도는 제1도의 CACP 회로 및 버스 인터페이스 유닛의 블록회로도.

제5도는 제1도의 버스 인터페이스 유닛에 구현된 중재 제어 논리의 일실시예의 회로도.

\* 도면의 주요부분에 대한 부호의 설명

10 : 컴퓨터 시스템 12 : 시스템 보드

14 : 프로세서 복합체 16 : 프로세서부

18 : 베이스부 20 : 로컬버스

22 : 로컬버스 접속기 24, 26 : 시스템 메모리

28 : 입/출력(I/O) 장치 30 : 메모리 버스

32 : 입/출력(I/O) 버스 34 : 프로세서 복합체 접속기

38 : 중앙 처리 장치(CPU)

40 : 정적랜덤 액세스 메모리(SRAM) 42 : 캐시 제어모듈

44 : 주파수 제어모듈 46 : 어드레스 버퍼  
 48 : 데이터 버퍼 50, 78 : 데이터 정보통로  
 52, 80 : 어드레스 정보통로 54, 82 : 제어 정보통로  
 56 : 랜덤 액세스 메모리(RAM) 58 : 메모리 제어기  
 60 : DMA 제어기 62 : 중앙 중재 제어기(CACP)  
 64 : 버스 인터페이스 유니트 66 : 버퍼/에러 정정코드(ECC)  
 68 : 구동회로 70 : 리드 온리 메모리(ROM)  
 72 : 셀프 테스트 회로 74 : 버퍼 회로  
 76 : 시스템 버스 102, 104 : 구동기/수신기  
 106 : 시스템 버스-I/O 버스 변환 논리 회로  
 108 : I/O 버스-시스템 버스 변환 논리 회로  
 110 : 메모리 어드레스 비교 논리 회로 112 : 에러 복구지원 논리 회로  
 114 : 캐시 스누핑 회로 116 : 프로그램된 I/O 회로  
 118 : 시스템 버스 어드레스 발생 회로  
 120 : I/O 버스 예상 어드레스 발생 회로  
 122 : 시스템 버스 제어기 인터페이스 123 : FIFO 제어 회로  
 124 : FIFO 버퍼 125A, 125B, 125C, 125D : 버퍼  
 126 : I/O 버스 슬레이브 인터페이스  
 128 : 버스-버스 페이싱 제어 논리 회로 148 : AND 게이트  
 150, 152 : S-R 래치

[발명의 상세한 설명]

본 발명은 컴퓨터 시스템에 있어서 버스-버스 인터페이스(bus to bus interface)에 관한 것으로서, 특히 이중 버스 구조 컴퓨터 시스템(a dual architecture computer system)의 버스 제어를 중재하기 위한 개선된 중재 제어 논리 회로(arbitration control logic) 및 중재 제어 방법에 관한 것이다.

일반적으로 컴퓨터 시스템, 특히 퍼스널 컴퓨터 시스템에 있어서는, 중앙 처리 장치(CPU), 메모리 장치, 직접 메모리 액세스(DMA) 제어기와 같은 직종 시스템 장치들간에서 데이터가 전송된다. 또한, 입/출력(I/O) 장치와 같은 확장 장치(expansion element)들 사이에서도 데이터가 전송되며, I/O 장치와 각종 시스템 장치 간에서도 데이터가 전송된다. I/O 장치들과 시스템 장치들은 컴퓨터 버스를 통해 상호통신하는데, 컴퓨터 버스는 수개의 발신지 중의 어떤 발신지로 부터 수개의 수신지 중의 어떤 수신지로 정보를 전송하는 일련의 도체들로 구성된다. 대다수의 시스템 장치와 I/O 장치는 버스 제어기(bus controllers) (즉, 컴퓨터 시스템을 제어할 수 있는 장치)와 버스 슬레이브(bus slaves) (즉, 버스 제어기에 의해 제어되는 장치)로서의 역할을 행할 수 있다.

버스를 둘 이상 갖는 퍼스널 컴퓨터 시스템은 이미 알려져 있는데, 이러한 시스템은, 통상적으로, CPU와 캐시 메모리나 메모리 제어기 간의 통신이 이루어지는 로컬(local) 버스를 갖추고 있으며, 또한 DMA 제어기나 I/O 장치와 같은 시스템 버스장치와 시스템 메모리 간의 통신이 메모리 제어기를 통해 이루어지는 시스템 I/O 버스를 갖추고 있다. 상기한 시스템 I/O 버스는 버스 인터페이스 유니트에 의해 상호 접속되는 시스템 버스와 I/O 버스를 포함한다. I/O 장치들은 I/O 버스를 통해 상호 통신한다. 이들 I/O 장치는 또한 시스템 메모리와 같은 시스템 버스장치와도 통신할 수 있어야만 하는 것이 통상적이다. 이러한 통신은 I/O 버스와 시스템 버스 및 이들 간의 버스 인터페이스 유니트를 통해 이루어져야만 한다.

중종 컴퓨터 시스템에서는 확장 장치들(expansion devices)에 대해서 이들의 I/O 버스 소유를 중재할 필요가 있다. 즉, 어떤 확장장치가 I/O 버스를 통하여 정보를 전송할 것인지 결정할 필요가 있다. 어떤 확장 장치가 I/O 버스를 통하여 정보를 전송할 수 있는가를 결정하는, 즉 버스를 할당하는 중재기(arbiter)가 알려져 있다. 그러한 중재기는 중앙 중재 제어 포인트(또는 중앙 중재 제어기)(central arbitration control point ; 이하 CACP라 함)로서 행사하며, I/O 버스에 대한 모든 중재가 그것을 통하여 이루어진다.

I/O 버스는 버스 사이클이라는 시간 분할 유니트(time divided unit)로 동작한다. I/O 버스의 버스 사이클은 중재(arbitration) 사이클과 승인(grant) 사이클로 분류된다. 중재 사이클 동안, 확장 장치들은 I/O 버스를 소유하기 위하여 다투며, 승인 사이클 동안, 그 버스의 소유권(ownership)을 획득한 장치가 정보를 전송한다.

I/O 장치는 중재기가 승인 모드일 동안에만 시스템 메모리에 대한 판독 또는 기록 사이클을 개시할 수 있다. 이러한 연속적 중재 방법은 중재와, 중재기가 승인 모드일 동안에만 가능한 메모리 판독 또는 기록 사이클을 별도로 수행하는데 필요한 시간으로 인하여 이중 버스 구조를 가진 컴퓨터 시스템의 성능을 제한하는 결과를 가져온다.

따라서, 본 발명의 목적은 (i) I/O 버스의 제어를 다루는 CPU와 I/O 장치간의 중재, 및 (ii) 시스

템 메모리에 대한 I/O 제어기 장치의 판독 또는 기록 동작, 또는 I/O 슬레이브 장치(slave device) 상의 확장 메모리(expansion memory)에 대한 CPU 기록 동작을 동시에(simultaneously) 허용하는 이중 버스 컴퓨터 시스템의 중재 시스템 및 방법을 제공하는데 있다.

본 발명에 따르면, 중재 제어 논리 회로 및 중재 방법이 이중 버스 구조를 갖는 컴퓨터에 제공된다. 중재 제어 논리는 컴퓨터 시스템의 시스템 버스와 입/출력 버스 사이에 상주하는 버스 인터페이스 유니트내의 하드웨어로 구축된 알고리즘으로써 구현된다. 이러한 중재 제어 논리로서 CACP에 의한 중재 사이클과 CPU 또는 I/O 장치에 의한 메모리 기록 및 판독 사이클을 동시에 허용함으로써 이중 버스 구조의 컴퓨터의 성능을 향상시킨다.

중재 제어 논리는 3가지 규정된 조건하에서 동작한다. 제1조건하에서는 I/O 장치가 데이터를 시스템 메모리에 기록하려 한다. 중재 사이클의 완료 후, CACP는 I/O 버스(및 버스 인터페이스 유니트를 통하여 시스템 버스)의 제어를 I/O 장치에 승인한다. I/O 장치는 시스템 메모리에 대해 1 또는 그 이상의 기록 사이클을 개시하고, 이 사이클은 버스 인터페이스 유니트의 버퍼에 일시적으로 저장되며, 그 후 시스템 메모리에 기억된다. I/O 장치는 언제든지 I/O 버스의 제어를 포기할 수도 있기 때문에, 버스 인터페이스 유니트는 아직 버퍼로부터 시스템 메모리로 기록되어질 필요가 있는 버퍼된 데이터를 보관할 수도 있다.

비록 I/O 장치가 I/O 버스의 제어를 포기할 수는 있어도, 시스템 버스를 통하여 시스템 메모리로 버퍼된 데이터의 기록 전송을 종료하기 위해서, (버스 인터페이스 유니트를 통하여)시스템 버스의 제어를 유지하여야 한다. 버스 인터페이스 유니트의 중재 제어 논리는 이 조건을 인식하여 CACP로 CACP 오버라이드 신호를 활성화시켜(버스 인터페이스 유니트를 통하여) 시스템 버스의 제어를 유지하도록 요청한다. 따라서, I/O 장치로 부터 버퍼된 기록 데이터가 시스템 메모리로 전송되는 동안에 CACP가 새로운 중재 사이클을 동시에 시작한다. 버퍼된 기록 데이터의 전송이 완료되는 즉시 CACP 오버라이드 신호가 비활성화되어 CACP가 중재를 종료하고 I/O 버스를 승인하는 것을 허용한다.

버스 인터페이스 유니트내의 중재 제어 논리 회로가 동작하는 제2조건은 I/O 장치가 시스템 메모리로부터 데이터를 판독하고자 할 때 발생한다. 중재 사이클의 종료 후, CACP는 I/O 버스(및 버스 인터페이스 유니트를 통하여 시스템 버스)의 제어를 I/O 장치에 승인한다. I/O 장치는 하나 또는 둘 이상의 판독 사이클을 시스템 메모리에 대해 개시한다. 버스 인터페이스 유니트내의 버퍼는 사전에 판독한 것에 인접한 어드레스를 가지는 프리페치된 데이터로 채워진다. I/O 장치는 일단 버퍼로부터 마지막 바이트의 데이터를 판독하면 I/O 버스의 제어를 포기할 수도 있으며, 따라서 프리페치된 데이터는 불필요하게 된다. 따라서 이런 불필요한 데이터를 효과적으로 제거하기 위하여 버스 인터페이스 유니트내의 버퍼는 반드시 리셋(reset) 되어야 한다.

비록 I/O 장치가 언제든지 I/O 버스의 제어를 포기할 수는 있지만, 시스템 메모리로부터 인터페이스 유니트로의 현 데이터(present data) 전송을 종료할 때까지(버스 인터페이스 유니트를 통하여) 시스템 버스의 제어를 유지하여야 한다. 버스 인터페이스 유니트내의 중재 제어 논리 회로는 이 조건을 인식하여 CACP로 CACP 오버라이드 신호를 활성화시켜 마지막 프리페치 작용이 종료되고 FIFO 버퍼가 리셋 될때까지(버스 인터페이스 유니트를 통하여) 시스템 버스의 제어를 유지하도록 요청한다. 따라서, 버스 인터페이스 유니트가 시스템 버스를 통하여 시스템 메모리로부터 마지막 프리페치 동작을 종료하고 FIFO 버퍼를 리셋시킬 동안에 CACP는 이와 동시에 I/O 버스상에서 새로운 중재 사이클을 시작한다.

버스 인터페이스 유니트내의 중재 제어 논리 회로가 동작하는 제3조건은 시스템 버스를 제어하는 CPU와 같은 시스템 장치가 I/O 버스상의 슬레이브로서 동작하는 I/O 장치에 데이터를 기록하고자 할 때 발생한다. 중재 사이클의 종료 후, CACP는 시스템 버스(및 버스 인터페이스 유니트를 통하여 I/O 버스)의 제어를 시스템 장치에 승인하여, 이 시스템 장치는 시스템 버스-I/O 버스 변환 논리 회로를 통하여 자신의 기록 사이클을 I/O 장치에 대해 개시한다.

시스템 버스-I/O 버스변환 논리 회로는 버스를 제공하는데, 이 버퍼내에 CPU와 같은 시스템 장치로부터 I/O 슬레이브 장치상의 확장 메모리로 기록되어질 데이터가 일시적으로 저장된 후 I/O 슬레이브 장치로 기록된다. 마지막 전송 데이터가 변환 논리 회로에 저장되면 시스템 버스는 더이상 필요치 않다. CPU 및 버스 인터페이스 유니트는 시스템 버스가 더이상 필요 없음을 표시한다. 이때 변환 논리 회로에 버퍼된 데이터가 I/O 버스를 통하여 I/O 장치에 아직 기록되어야 할지라도 CACP는 I/O 버스에 대한 중재를 시작할 수도 있다. I/O 장치와 달리, CPU는 CACP가 중재 모드이든 또는 승인 모드이든 어느 한 모드동안 I/O 장치에 기록할 수 있기 때문이다.

버스 인터페이스 유니트내의 중재 제어 논리 회로는 이 조건을 인식하여 CACP 오버라이드 신호를 CACP로 작동시켜 버퍼에 저장된 마지막 데이터가 I/O 장치에 기록될 때까지 중재 제어 논리 회로가 I/O 버스의 제어를 유지하도록 한다. 따라서, 변환 논리 회로에 저장된 데이터가 I/O 장치에 기록될 동안, 중앙 중재 제어기가 중재 사이클을 동시에 작동시킨다.

제1도에 도시된 바와 같이, 컴퓨터 시스템(10)은 시스템 보드(12)와 프로세서 복합체(processor complex) (14)로 이루어진다. 프로세서 복합체(14)는 로컬 버스 접속기(local bus connector) (22)를 통해 프로세서 로컬 버스(20)에 접속되는 프로세서부(16)와 베이스부(18)로 이루어진다. 프로세서부(16)는 50MHz에서 동작되며, 베이스부(18)는 40MHz에서 동작된다.

시스템 보드(12)는 인터리브된 시스템 메모리(interleaved system memories) (24),(26)와 입/출력(I/O) 장치(28)를 포함한다. 시스템 메모리(24),(26)와 프로세서 복합체(14)간의 통신은 메모리 버스(30)에 의해 행해지며, I/O 장치(28)와 프로세서 복합체(14)간의 통신은 I/O 버스(32)에 의해 행해진다. I/O 장치(28)와 시스템 메모리(24),(26)간의 통신은 I/O 버스(32), 시스템 버스(76)의 메모리 버스(30)에 의해 행해진다. I/O 버스(32)는 마이크로 채널(MICRO CHANNEL<sup>®</sup>) 컴퓨터 구조에 따를 수도 있다. 메모리 버스(30)와 I/O 버스(32)는 프로세서 복합체 접속기(processor complex connector) (34)에 의해 프로세서 복합체 베이스부(18) 접속된다. 메모리 확장 장치와 같은 I/O 장

치는 I/O 버스(32)를 통해 상기 컴퓨터 시스템(10)에 접속될 수도 있다. 시스템 보드(12)는 정규 동작중 상기 컴퓨터 시스템(10)에 의해 사용되는 통상적인 비디오 회로, 타이밍 회로, 키보드 제어 회로와 인터럽트 회로(도시되지 않음)을 포함할 수도 있다.

프로세서 복합체(14)의 프로세서부(16)는 중앙 처리 장치(CPU)(38)를 포함하는데, 이 CPU는, 본 바람직한 실시예의 경우, i486으로 불리우며 인텔(Intel)사로부터 구입가능한 32-비트 마이크로 프로세서를 사용하고 있다. 프로세서부(16)는 정적 랜덤 액세스 메모리(SRAM) (40), 캐시 제어 모듈(42), 주파수 제어 모듈(44), 어드레스 버퍼(46) 및 데이터 버퍼(48)를 포함한다. 로컬 버스(20)는 데이터 정보 통로(50), 어드레스 정보 통로(52)와 제어 정보 통로(54)로 구성된다. 데이터 정보 통로(50)는 CPU(38), SRAM(40)과 데이터 버퍼(48)간에 제공된다. 어드레스 정보 통로(52)는 CPU(38), 캐시 제어 모듈(42)과 어드레스 버퍼(46)간에 제공되며, 제어 정보 통로(54)는 CPU(38), 캐시 제어 모듈(42)과 주파수 제어 모듈(44)간에 제공된다. 또한, 캐시 제어 모듈(42)과 SRAM(40)간에 어드레스와 제어 정보 통로가 제공된다.

SRAM(40)은 시스템 메모리(24) 또는 (26)으로부터 또는 I/O 장치(28)상에 위치한 확장 메모리로부터의 메모리 정보를 단기간 저장하는 캐시 기능을 제공한다. 캐시 제어 모듈(42)은 상기 시스템 메모리(24),(26)의 어드레스 위치들을 저장하는 랜덤 액세스 메모리(RAM)(56)를 포함하고 있다. CPU(38)는 로컬 버스(20)를 통해 SRAM(40)에 저장된 정보를 직접 액세스 할 수 있다. 주파수 제어 모듈(44)은 50MHz 프로세서부(16)의 동작을 40MHz 베이스부(18)와 동기시키며, 또한, 버퍼(46),(48)의 동작을 제어한다. 따라서, 주파수 제어 모듈(44)은 버퍼(46),(48)에 정보가 저장되는 시간과 이러한 버퍼에 저장되는 정보가 오버라이트(overwrite)되는 시간을 결정한다. 버퍼(46),(48)는 시스템 메모리(24),(26)로부터의 두개의 기록(two writes)이 동시에 저장될 수 있게 구성된다. 버퍼(46),(48)는 양방향성 버퍼로서, 이들 버퍼는 CPU(38)에 의해 제공되는 정보와 CPU(38)로 제공되는 정보를 래치(latch)할 수 있다. 이들 버퍼(46),(48)는 양방향성이기 때문에, 프로세서 복합체(14)는 표준 베이스부(18)는 그대로 돈채로 대체 또는 개량될 수도 있다.

베이스부(18)는 메모리 제어기(58), DMA 제어기(60), 중앙 중재 제어기(central arbitration control point)(CACP) 회로(62), 버스 인터페이스 유니트(64)와 버퍼/에러 정정코드(buffer/error correction code)(ECC) 회로(66)를 포함한다. 또한, 베이스부(18)는 구동 회로(68), 리드온리 메모리(ROM)(70), 셀프 테스트 회로(72)와 버퍼(74)를 포함한다. 시스템 버스(76)는 데이터 정보 통로(78), 어드레스 정보 통로(80)와 제어 정보 통로(82)를 구비한다. 데이터 정보 통로(78)는 버퍼(74)를 버스 인터페이스 유니트(64)와 접속하며, 버스 인터페이스 유니트(64)를 DMA 제어기(60) 및 버퍼/ECC 회로(66)와 접속하고 버퍼/ECC 회로(66)를 시스템 메모리(24),(26)와 접속한다. 어드레스 정보통로와 제어 정보 통로 각각은 메모리 제어기(58)를 DMA 제어기(60) 및 버스 인터페이스 유니트(64)와 접속하고, 버스 인터페이스 유니트(64)를 버퍼(74)와 접속한다.

메모리 제어기(58)는 CPU 로컬 버스(20)와 시스템 버스(76)상에 위치하며, CPU(38), DMA 제어기(60) 또는 (I/O 장치(28) 대신) 버스 인터페이스 유니트(64)에 메모리 버스(30)를 통해 시스템 메모리(24),(26)를 액세스하도록 해준다. 메모리 제어기(58)는 메모리 버스(30)를 통해 시스템 메모리(24),(26)에 대한 시스템 메모리 사이클을 개시한다. 시스템 메모리 사이클 동안, CPU(38), DMA 제어기(60) 또는 (I/O 장치(28) 대신) 버스 인터페이스 유니트(64)는 메모리 제어기(58)를 통해 시스템 메모리(24),(26)를 액세스한다. 상기 DMA 제어기(60) 또는(I/O 장치(28) 대신) 버스 인터페이스 유니트(64)는 시스템 버스(76), 메모리 제어기(58) 및 메모리 버스(30)를 통해 시스템 메모리를 액세스하는 한편, CPU(38)는 로컬 버스(20), 메모리 제어기(58)와 메모리 버스(30)를 통해 시스템 메모리와 통신한다.

CPU(38)-I/O 버스(32) 판독 또는 기록 사이클에 있어서, 어드레스 정보는 시스템 메모리 어드레스 경계에 대해 체크된다. 이때, 어드레스 정보가 I/O 확장 메모리 어드레스 또는 I/O 포트(port) 어드레스와 일치하면, 메모리 제어기(58)는(버스 인터페이스 유니트(64)를 거쳐) I/O 버스(32)를 통해 I/O 장치(28)에 대한 I/O 메모리 사이클 또는 I/O 포트사이클을 개시한다. CPU-I/O 메모리 사이클 또는 I/O 포트사이클동안, 메모리 제어기(58)에 공급되는 어드레스는 시스템 버스(76)와 I/O 버스(32) 사이에 존재하는 버스 인터페이스 유니트(64)를 통해 상기 시스템 버스(76)로 부터 상기 I/O 버스(32)로 전송된다. 상기 어드레스에 상응하는 확장메모리를 포함하는 I/O 장치(28)는 I/O 버스(32)로부터 상기 메모리 어드레스를 수신한다. DMA 제어기(60)와 버스 인터페이스 유니트(64)는 시스템 메모리(24),(26)와 I/O 장치(28) 내의 확장 메모리 간의 정보 교환을 제어한다. DMA 제어기(60)는 또한 프로세서 복합체(14)를 대신하여 3가지 기능을 제공하는데, 첫째로, DMA 제어기(60)는 DMA 채널을 구성(configure)하기 위해 소규모 컴퓨터 서브 시스템 제어 블록(small computer subsystem control block)(SCB) 구조를 이용하므로 상기 DMA 채널을 구성하기 위해 프로그램된 I/O를 사용할 필요가 없다. 둘째로, DMA 제어기(60)는 저속도로 동작하는 메모리 확장 장치와 통상적으로 빠른 속도로 동작하는 시스템 메모리 간의 데이터 전송이 최적화되도록 버퍼링 기능을 제공한다. 셋째로, DMA 제어기(60)는 8채널의 32비트 직접 시스템 메모리 액세스 기능을 제공한다. 직접 시스템 메모리 액세스 기능을 제공할때, DMA 제어기(60)는 다음의 두 모드로 작동될 것이다. 첫째 모드에 있어서는 DMA 제어기(60)가 CPU(38)에 대해 기능적으로 슬레이브(slave)인 프로그램된 I/O 모드로 작동한다. 둘째 모드에 있어서는, DMA 제어기(60)는 시스템 버스 마스터(master)로서 동작하여, I/O 버스(32)를 중재하고 제어한다. 이러한 둘째 모드 동안, DMA 제어기(60)는 퍼스트-인-퍼스트-아웃(first in, first out)(FIFO) 레지스터 회로를 사용한다.

중앙 중재 제어기(CACP) 회로(62)는 DMA 제어기, I/O 장치 버스 제어기와 CPU (I/O 장치를 액세스하는 경우)에 대해 중재기로서 작용한다. CACP 회로(62)는 I/O 장치(28) 뿐만 아니라, DMA 제어기(60)와 메모리 제어기(58)로부터 중재 제어 신호를 수신하여 어떤 장치가 I/O 버스(32)를 제어할 것인가를 판단하고, 그 장치가 I/O 버스의 제어를 유지할 시간을 결정한다.

구동 회로(68)는 메모리 제어기(58)로부터 시스템메모리(24),(26)로 제어 정보와 어드레스 정보를 제공한다. 또한, 구동 회로(68)는 시스템 메모리(24),(26)를 구성하는데 사용되는 싱글 인-라인 메

모리 모듈들(single in-line memory modules)(SIMMS)의 수에 입각하여 이러한 정보를 구동한다. 따라서, 구동회로(68)는 시스템 메모리(24),(26)의 크기에 따라 시스템 메모리(24)와 (26)에 제공되는 제어 및 어드레스 정보의 신호 강도를 가변한다.

버퍼 회로(74)는 프로세서 복합체의 베이스부(18)와 시스템 보드(12) 사이에서 증폭작용 및 분리(isolation) 작용을 제공한다. 또한, 버퍼 회로(74)는 I/O 버스(32)와 버스 인터페이스 유니트(64) 간의 경계정보(boundary information)를 실시간으로 저장할 수 있는 버퍼를 사용한다. 따라서, 컴퓨터 시스템(10)이 고장나면, 버퍼 회로(74)는 컴퓨터 수리 기사에 의해 액세스 되어 수도 시스템 고장시에 프로세서 복합체 접속기(34)에 제공된 정보를 판단할 수도 있게 한다.

ROM(70)은 전원 투입시 확장 메모리로부터의 데이터를 시스템 메모리에 놓이게 함으로써 컴퓨터 시스템(10)을 구성한다. 셀프 테스트 회로(self test circuit)(72)는 베이스부(18)내의 다수 위치에 연결되며, 다수의 셀프 테스트 특징(features)을 제공한다. 셀프 테스트 회로(72)는 고장이 발생했는지를 판단하기 위해 버퍼 회로(74)를 액세스하며, 또한 시스템의 전원 투입시에 시스템이 작동 준비 상태인가를 판단하기 위해 베이스부(18)의 나머지 다른 주요 부분(성분)도 테스트한다.

제2도는 제1도에 도시한 시스템의 버스 인터페이스 유니트(64)에 대한 개략 블록도이다. 버스 인터페이스 유니트(64)는 시스템 버스(76)와 I/O 버스(32)간에 양방향성 고속 인터페이스를 제공함으로써 본 발명의 구현을 위한 근원(base)을 제공한다.

버스 인터페이스 유니트(64)는 시스템 버스 구동기/수신기 회로(system bus driver/receiver circuit)(102)와, I/O 버스 구동기/수신기 회로(104) 및, 이들간에 전기적으로 접속된 제어 논리 회로를 포함한다. 구동기/수신기 회로(102)는 시스템 버스(76)로 부터 신호를 수신하여 적절한 버스 인터페이스 유니트 제어 논리 회로로 보내며, 버스 인터페이스 유니트 제어 논리 회로로부터 신호를 수신하여 시스템 버스에 보내는 조향(steering) 논리 회로를 갖는다. I/O 버스 구동기/수신기 회로(104)는 I/O 버스(32)로부터 신호를 수신하여 적절한 버스 인터페이스 유니트 제어 논리 회로로 보내며, 버스 인터페이스 유니트 제어 논리 회로로 부터 신호를 수신하여 I/O 버스(32)에 보내는 조향 논리 회로를 포함한다.

버스 인터페이스 유니트의 제어 논리 회로는 시스템 버스-I/O 버스 변환 논리(system bus to I/O bus translation logic) 회로(106), I/O 버스-시스템 버스 변환 논리(I/O bus to system bus translation logic) 회로(108), 메모리 어드레스 비교 논리 회로(110), 에러 복구 지원 논리(error recovery support logic) 회로(112), 그리고 캐시 스누핑(snooping) 논리 회로(114)를 포함한다. 프로그램된 I/O 회로(programmed I/O circuit)(116)는 시스템 구동기/수신기 회로(102)와 전기적으로 결합된다.

시스템 버스-I/O 버스 변환 논리 회로(106)는 DMA 제어기(60) 또는(CPU(38)을 대신해서) 메모리 제어기(58)가 시스템 버스 제어기로서 동작하여 I/O 버스(32)를 액세스하며 이에 따라 I/O 버스상의 슬레이브 장치로서 작용하는 I/O 장치(28)와 통신하는데 필요한 수단을 제공한다. 변환 논리 회로(106)는 시스템 버스(76)의 제어, 어드레스와 데이터 라인을 I/O 버스(32)의 제어, 어드레스와 데이터 라인으로 변환한다. 대다수의 제어 신호와 모든 어드레스 신호는 시스템 버스(76)로부터 I/O 버스(32)로 전송되며, 데이터 정보는 양방향으로 전송된다. 시스템 버스 슬레이브로서 동작하는 상기 논리 회로는 상기 시스템 버스(76)를 감시하며 상기 I/O 버스(32)에 대한 사이클을 검출한다. 이러한 사이클의 검출시, 시스템 버스 슬레이브는 시스템 버스 상의 신호의 타이밍을 I/O 버스 타이밍으로 변환하고, I/O 버스(32)에 대하여 상기 사이클을 개시하고, 사이클이 완료되기까지 대기하며, 시스템 버스(76)상의 상기 사이클을 종료한다.

I/O 버스-시스템 버스 변환 논리 회로(108)는 시스템 버스 어드레스 발생 회로(118), I/O 버스 예상(expected) 어드레스 발생 회로(120), 시스템 버스 제어기 인터페이스(122), FIFO 버퍼(124), I/O 버스 슬레이브 인터페이스(126)와 버스-버스 페이싱(us to bus pacing) 제어 논리 회로(128)로 구성된다. 시스템 버스 제어기 인터페이스(122)는 40MHz에서 동작하는 고성능 32bit(4byte) i486 버스트 프로토콜을 지원한다. 버스트 모드에서는 4, 8 및 16바이트의 데이터 전송이 또한 비버스트 모드(no-burst mode)에서는 1-4 바이트의 데이터 전송이 제공된다. I/O 버스 슬레이브 인터페이스(126)는 상기 시스템 버스(76)상의 슬레이브 장치에 대한 동작에 대하여 I/O 버스(32)를 감시하며, I/O 버스(32)에 대한 동작들은 무시한다. I/O 버스 슬레이브 인터페이스(126)에 의해 픽업된(picked-up) 모든 사이클은 FIFO 버퍼(124)와 시스템 버스제어기 인터페이스(122)로 전송된다.

I/O 버스-시스템 버스 변환 논리 회로(108)는 I/O 장치(28)가 I/O 버스 제어기로서 작용하여 시스템 버스(76)를 액세스 하며, 따라서 시스템 메모리(24),(26)에 대한 판독 또는 기록하는데 필요한 수단을 제공한다. 이들 양 동작에 있어서, I/O 장치는 I/O 버스를 제어한다. I/O 장치의 속도로 동작하는 비동기식 I/O 버스 인터페이스(126)는 버스 인터페이스 유니트(64)가 I/O 버스(32)상의 I/O 장치 제어기에 대한 슬레이브로서 작용하여 메모리 어드레스를 디코드해서 판독 또는 기록 사이클이 시스템 메모리(24) 또는 (26)에 대한 것임을 판단할 수 있게 한다. 이와 동시에, 상기 시스템 버스 제어기 인터페이스(122)는 버스 인터페이스 유니트(64)가 시스템 버스(74)에 대한 제어기로서 작용할 수 있게 한다. 메모리 제어기(58)(제1도)는 버스 인터페이스 유니트(64)에 대한 슬레이브로 작동하여 버스 인터페이스 유니트(64)에 시스템 메모리로부터 판독된 데이터를 제공하거나 시스템 메모리에 데이터를 기록한다. 시스템 메모리에 대한 판독과 기록은 FIFO 버퍼(124)를 통해 이루어진다. 버퍼(124)의 블록도는 제3도에 도시한다.

제3도에 도시한 바와 같이, FIFO 버퍼(124)는 시스템 버스(76)와 I/O 버스(32)간의 데이터 정보를 일시적으로 저장하는 이중 포트의 비동기식 양방향 저장 유니트이다. FIFO 버퍼(124)는 4개의 16바이트 버퍼(125A-125D) 및 FIFO 제어회로(123)를 포함한다. 4개의 버퍼(125A-125D)는 I/O 버스 제어기와 시스템 버스 슬레이브들에 대한 데이터를 버퍼링함으로써 I/O 버스(32)및 시스템 버스(76)가 동시에 동작할 수 있도록 한다. FIFO 버퍼(124)는 물리적으로 2개의 32바이트 버퍼(125A/125B 및 125C/125D)로 구성된다. 시스템 버스 제어기 인터페이스(122) 및 I/O 버스 슬레이브 인터페이스

(126)의 각각은 하나의 32바이트 버퍼를 제어하는데, 이때 나머지 하나의 32바이트 버퍼는 그들에게 있어 명백하게 동작한다. 2개의 32바이트 버퍼는 모두 기록 및 판독 동작에 사용한다.

각각의 FIFO(125A), (125B), (125C), (125D)는 어드레스 레지스터 섹션(address register section)을 갖는데, 이 레지스터 섹션은 각각 FIFO와 물리적으로 또는 논리적으로 연관된다. 어드레스들이 서로 인접하는 경우, I/O 버스(32)로 부터 FIFO(125A)로 데이터가 전송됨에 따라, 그 데이터는 16바이트 버퍼가 16바이트의 데이터로 채워질 때까지 누적될 것이다. 비인접 어드레스(non-contiguous address)가 어드레스 섹션에 의해 검출될 경우, FIFO(125A)는 저장된 데이터를 FIFO(125C)로 전송하고, 이와 동시에 FIFO(125B)는 새로운 비인접 어드레스로 부터의 데이터를 수신하기 시작한다. FIFO(125B)는 FIFO(125A)의 경우와 같이, 16바이트의 데이터로 채워지거나, 또 다른 비인접 어드레스가 검출될 때까지 데이터를 계속 수신한다. FIFO(125B)는, 그 다음, 저장된 데이터를 FIFO(125D)로 전송하며, FIFO(125A)는 데이터를 다시 저장하기 시작한다. 따라서, 16바이트 비인접 어드레스 데이터 블록을 4개까지 저장할 수 있다.

이외에도, 두개의 32바이트 버퍼를 병렬로 하여 데이터의 판독 및 기록이 두 버퍼 사이에서 토글(toggle) 될 수 있고 따라서, 본질적으로 연속적인 판독 및 기록 기능을 수행할 수 있다.

또한, 32바이트 버퍼를 다른 I/O 버스(32) 또는 시스템 버스(76)에 연결되는 16바이트 섹션들로 분할함으로써, 저장 데이터를 레지스터의 안으로 또는 밖으로 클럭킹하는 신호의 용량성 로딩(capacitive loading)과 관련되는 FIFO의 성능에 대한 영향을 최소로 하면서, 저장 버퍼의 수를 증가시킬 수 있다. 이것은 2개의 버퍼가(병렬로) 부가될 때마다 각 버스상의 클럭신호의 로딩에 상기 용량성 로딩의 절반만이 부가되기 때문이다.

더우기, 2개의 16바이트 버퍼를 매 경로상에 직렬로 둬으로써, 판독 동작에서와 같이, 일단 2개의 16바이트 버퍼중 하나가 데이터로 채워지면 직렬로 연결된 다른 16바이트 버퍼로 그 데이터가 전송되는 한편 다른 병렬 경로에서는 데이터의 누적을 지속할 수 있다. 따라서, 데이터를 누적하거나 하나의 버스로 부터 나머지 다른 버스로 데이터를 전송함에 있어 시간적 손실이 없다.

FIFO 제어회로(123)는 FIFO(124)의 동작을 제어하는 논리를 제공한다.

개개의 I/O 디바이스(28)는 1, 2 또는 4바이트(즉 8, 16 또는 32비트)의 대역폭으로 I/O 버스를 통하여 시스템 메모리(24) 또는 (26)에 대한 기록을 행할 수도 있다. I/O 디바이스(28)가 시스템 메모리에 기록함에 있어, 제1전송 기록 데이터는 처음에 FIFO 버퍼(125A) 또는 (125B)에 저장된다. I/O 버스 예상 어드레스 발생회로(120)는 다음의 예상 어드레스 또는 인접 어드레스(contiguous address)를 계산한다. 상기 다음의 인접 어드레스를 이후의(subsequent) I/O 어드레스와 대조함으로써 이후의 전송이 인접한지의 여부를 확인할 수 있다. 인접하는 경우, 기록 데이터의 제2바이트 또는 바이트들이 동일한 FIFO 버퍼(125A) 또는 (125B)로 전송된다. FIFO는 I/O 버스(32)로 부터 최고 매초 40메가바이트의 비동기 속도(asynchronous speed)로 데이터를 수신할 수 있다.

이러한 과정은 버퍼(125A) 또는 버퍼(125B)가 16바이트 정보 패킷으로 채워지거나 또는 비인접 어드레스가 검출될 때까지 계속된다. 다음의 클럭 사이클에서, 버퍼(125A)가 채워졌다고 가정하면, 버퍼(125A)의 데이터는 버퍼(125C)로 전송된다. 이와 마찬가지로, 버퍼(125B)가 채워질 경우, 그것에 저장된 모든 내용은 단일 클럭 사이클로 버퍼(125D)에 전송된다. 버퍼(125C) 및 (125D)에 저장된 데이터는, 그 후, 시스템 버스 동작 속도의 i486 버스트 전송을 통해 시스템 메모리에 기록된다.

따라서, I/O 장치가 시스템 메모리에 기록하는 동안 버퍼(124)의 작동은 연속적이며, 버퍼(125A)와 (125B) 사이를 교번하여, 이들 버퍼의 각각은 근접한 버퍼(125C) 또는 (125D)로 데이터를 보내어 비워지고 다른 버퍼는 시스템 메모리에 기록될 데이터를 수신하게 된다. 이때, FIFO 버퍼(124)는 (i) 메모리에 기록할 다음 데이터 바이트의 어드레스를 예상하고, (ii) 시스템 버스(76)를 통하여 FIFO 버퍼로 부터 시스템 메모리로 기록할 데이터의 최대 속도를 수용함으로써 시스템 메모리로의 데이터 기록 속도를 최적화 한다.

시스템 메모리로 부터 I/O 장치(28)로의 데이터 판독 동안, FIFO 버퍼(124)는 전술한 경우와 상이하게 동작한다. 시스템 버스 어드레스 발생 회로(118)는 초기 판독 어드레스를 사용하여 이후의 판독 데이터의 판독 어드레스를 발생시키고 버퍼(125C) 또는 (125D)에 데이터를 누적한다. 시스템 버스가 16바이트 대역폭의 전송을 지원하기 때문에, 시스템 버스 제어기 인터페이스 회로(122)는 I/O 버스(32)가 실제 연속 어드레스를 제공하지 않아도 16바이트 패킷의 인접 데이터를 프리페치(prefetch)하여 그것을 버퍼(125C) 또는 (125D)에 저장할 수 있어, 전송에서의 지연(latency)이 줄어든다. 버퍼(125C)가 프리페치된 데이터로 채워질 경우, 한 클럭 사이클에 그 데이터를 버퍼(125A)로 전송한다. 마찬가지로, 버퍼(125D)가 프리페치된 데이터로 채워지면, 그 데이터를 버퍼(125B)로 이동시킨다. 그후, 버퍼(125A) 및 (125B)에 저장된 데이터는 개개의 I/O 장치 제어기에 의해 1, 2 또는 4바이트의 대역폭으로 판독될 수도 있다. 이와 같이 시스템 버스 어드레스 발생 회로(118)는 데이터의 프리페치를 중단하도록 I/O 제어기 장치에 의해 지시받을 때까지 증분계수기로서 기능한다.

버스-버스 페이싱 제어 논리 회로(128)는 고속 I/O 장치가 시스템 메모리를 좀더 빠르게 액세스하도록 해준다. 버스-버스 페이싱 제어 논리 회로(128)는 시스템(10)의 통상적인 메모리 제어기 중재 계획(normal memory controller arbitration scheme)에 우선(override)한, 즉, I/O 장치와 CPU간의 메모리 제어기(58)에 대한 교번적 액세스보다는 다중 사이클(multiple cycles)을 요구하는 고속 장치에 의한 데이터 전송 동안 I/O 버스(32)를 통제하고 있는 I/O 장치에 시스템 메모리로의 중단되지 않는 액세스가 허용된다. 따라서, I/O 장치에 의한 다중 사이클 전송 동안 CPU와 같은 로컬 장치가 메모리 버스의 제어를 요청한다고 하더라도 버스-버스 페이싱 제어 논리 회로(128)는 I/O 장치에게 연속적인 메모리 버스 제어를 부여할 것이다.

프로그램된 I/O 회로(116)는 버스 인터페이스 유니트(64) 내의 프로그래머블 레지스터 모두를 포함하는, 버스 인터페이스 유니트(64)의 일부분이다. 이들 레지스터는 특정 레지스터가 활성인지 또는 비활성인지를 결정하기 위하여 그들과 연관된 비트들을 가진다. 이들 레지스터는, 무엇보다도, 버스 인터페이스 유니트(64)가 응답할 시스템 메모리 및 확장 메모리 어드레스 범위, 캐시어블

(cacheable) 또는 비캐시어블(non-cacheable) 확장 메모리 어드레스, 시스템 메모리 또는 캐시 어드레스 범위, 및 패리티(parity) 또는 에러 검사가 버스 인터페이스 유니트에 의해 지원을 받는지 여부를 규정한다. 따라서, 프로그램된 I/O 회로(116)는 버스 인터페이스 유니트(64)에 대해서 상주 환경과 구성 옵션(configuration options)을 식별한다. I/O 버스(32)를 통하여 직접적으로 프로그램된 I/O 회로(116)내의 레지스터를 프로그램할 수는 없다. 따라서, 시스템(10)을 프로그램하려면, 사용자는 시스템 버스를 통하여 프로그램된 I/O 회로(116)와 CPU 수준으로 통신할 수 있는 I/O 장치를 액세스 해야한다.

메모리 어드레스 비교 논리 회로(110)는 메모리 어드레스가 시스템 메모리에 상응하는지 또는 I/O 버스(32)에 결합된 I/O 장치(28)상의 확장 메모리에 상응하는지를 결정한다. 확장 메모리 뿐만 아니라 시스템 메모리는 비인접 어드레스 블록일 수도 있기 때문에, 메모리 어드레스 비교 논리 회로(110)는 어떤 경계가 어떤 메모리에 상응하는가를 나타내기 위해 프로그램된 I/O 회로(116) 내의 레지스터로 부터의 경계 정보를 적재한 다수의 비교기를 포함한다. 특정한 메모리 어드레스가 메모리 어드레스 비교 논리 회로에 의해 경계정보와 비교된 후, 메모리 인터페이스 유니트는 그에 상응하게 응답하도록 마련된다. 예를 들면, I/O 버스(32)를 통제하고 있는 I/O 디바이스가 확장 메모리에 대해 판독 또는 기록을 행할 경우, 버스 인터페이스 회로는 메모리 제어기(58)로 그 어드레스를 전송할 필요가 없으며, 따라서 시간 및 메모리 대역폭을 절약할 수 있다.

에러 복구 지원 논리 회로(122)는 데이터 패리티 에러(data parity error)가 검출될지라도 시스템(10)이 동작을 계속하도록 허용한다. I/O 장치(28)의 시스템 메모리(24) 또는 (26)에 대한 모든 기록 또는 판독에 있어, 데이터의 패리티가 검사된다. 에러 복구 지원 논리 회로(112)는 프로그램된 I/O 회로(116) 내의 레지스터와 상호작용하여 검출된 패리티 에러의 어드레스 및 시간을 획득한다. 그후, 이 레지스터의 내용은 소정의 시스템 소프트웨어에 의해서 적용될 수도 있다. 예를 들면, 패리티 에러가 검출될 때마다, 상기 어드레스를 상기 레지스터에서 인출하는 상위 수준 인터럽트로 CPU(38)가 프로그램될 수도 있다. 그후, CPU는 시스템 소프트웨어 명령에 의거하여, 시스템을 계속 동작 시키든지 아니면 확인된 패리티 에러원의 동작을 단순히 종료할 것인지 결정할 것이다.

캐시 스누핑 논리(cache snooping logic) 회로(114)는 버스 인터페이스 유니트(64)가 I/O 버스(32)를 통하여 발생하는 확장 메모리에 대한 I/O 장치의 모든 기록 동작에 대하여 I/O 버스(32)를 감시할 수 있게 한다. 먼저 스누핑 논리 회로는 확장 메모리로의 기록이 SRAM(40)내에 캐시 가능한 확장 메모리에서 발생했는지를 결정한다. 캐시 가능 확장 메모리가 아니면 캐시된 데이터가 손상될 염려가 없다. 그렇지만, 캐시 가능 확장 메모리에서 기록 동작이 발생했다는 것을 나타내는 양성 비교(positive compare), 캐시 무효화(invalidation) 사이클이 시스템 버스(76)을 통하여 개시된다. 따라서, CPU는 이같은 명령을 받아 SRAM(46)의 상응 어드레스를 무효화한다. 캐시 스누핑 논리 회로(114)는 양성 비교의 어드레스를 기억하는 수단을 제공하여 I/O 버스의 스누핑이 제1양성 비교 검출후에 즉시 연속되도록 하므로써, I/O 버스(32)의 연속적인 감시가 가능하다.

본 발명은 일반적으로 상기에서 기술한 버스 인터페이스 유니트(64)에 관한 것으로서, 특히 컴퓨터 시스템(10)에서 시스템 버스(76)와 I/O 버스(32) 중간에 위치한 버스 인터페이스 유니트(64)에 내장된 시스템 버스 중재 제어 논리 회로(130)(제5도)에 관한 것이다. 중재 제어 논리 회로는 중재 사이클과 승인 사이클을 교번하는 CACP 회로(62)와 상호 작용한다. CACP 회로(62)의 동작은 CONTROLLING BUS ALLOCATION USING ARBITRAION HOLD라는 명칭으로 1991년 10월 15일 출원되어 계류중인 미합중국 특허출원 제07/777,777호에 기술되어 있다. 버스 인터페이스 유니트(64)내의 중재 제어 논리 회로(130)는 CACP 회로(62)에 의한 중재 사이클이 CPU(38) 또는 I/O 장치(28)에 의한 메모리 판독 및 기록 사이클과 동시에 중첩(overlap)하는 것을 허용함으로써 이중 버스 구조 컴퓨터 시스템의 성능을 개선한다.

제4도는 I/O 버스(32), 시스템 버스(76), CACP 회로(62) 및 버스 인터페이스 유니트(64)의 시스템 상호 연결을 도시한다.

다중 전송 요청 라인(multiple transfer request line)(144)은 이미 I/O 버스(32)를 제어하고 있는 I/O 장치(28)가 I/O 버스(32)를 통하여 데이터의 1회 이상 전송을 수행할 준비가 되어 있음을 나타내는 수단으로써 이용된다. CACP는 데이터의 다중 전송들 모두가 I/O 버스(32)를 통하여 완료될 때까지 I/O 장치에 대해 I/O 버스를 승인 모드로 유지함으로써 이 요구에 응답한다. I/O 버스를 통하여 다중 전송을 완료하면, I/O 장치는 라인(144)을 비활성화(deactivate)하고, CACP 회로(62)는 I/O 디바이스가 I/O 버스에 대해 오프되었다고 결정하여 다음 중재 사이클(132)을 개시한다. 하기에서 설명되는 바와 같이, 버스 인터페이스 유니트(64)내의 중재 제어 논리 회로(130)는 CACP(62) 회로에 CACP 오버라이드(override) 신호를 제공하여 시스템 I/O 버스를 통해 작용이 발생하는 동안 CACP 회로가 승인 모드로 재진입(re-entering)하지 못하도록 한다. 오직 CACP 오버라이드 신호(146)가 비활성(inactive)으로 될때에만 CACP 회로(62)가 다음 승인 사이클에 진입할 수 있다.

중재 제어 논리 회로(130)는 3가지 규정된 조건하에서 동작한다. 제1조건하에서, I/O 버스(32)를 제어하고 있는 I/O 장치는 I/O 및 시스템 버스(32),(76)을 통하여 시스템 메모리(24),(26)에 데이터의 다중 전송을 기록한다. 제2조건하에서, I/O 버스(32)를 제어하고 있는 I/O 장치는 I/O 및 시스템 버스(32),(76)을 통하여 시스템 메모리(24),(26)로부터 데이터의 다중 전송을 판독한다. 제3조건하에서, 시스템 버스를 제어하는 CPU(38)와 같은 시스템 장치가 I/O 버스(32)상의 슬레이브로서 동작하는 I/O 장치로 데이터를 기록한다. 이 3가지 특정 동작들 각각에서 동작이 완료되기 전에 CACP 회로(62)에 의한 중재가 진행되도록 허용함으로써 I/O 버스에 대한 CACP 회로(62)의 중재와 각 동작의 동시적 종료(completion)를 허용한다.

이제, 중재 제어 논리 회로(130)가 동작하는 3가지 조건 각각을 상세하게 설명한다. 제1조건하에서, I/O 장치(28)는 시스템 메모리(24),(26)로 데이터를 기록하려 한다. 중재 사이클이 종료된 후, CACP 회로(62)는 I/O 버스(32)의(또는, 버스 인터페이스 유니트(64)를 통해 시스템 버스의) 제어를 I/O 장치(28)에 승인한다. I/O 장치는 시스템메모리(24),(26)로 FIFO 버퍼(125A) 및/또는 (125B)에 일시적으로 저장되어 있는 하나 또는 둘 이상의 기록 사이클을 개시한다. I/O 장치(28)는 언제든지 I/O

버스(32)의 제어를 포기할 수 있기 때문에, 버스 인터페이스 유닛은 버퍼(125A) 및/또는 (125B)로부터 버퍼(125C) 및/또는 (125D)로 기록되어지고 시스템 메모리(24),(26)로 출력되어질 필요가 있는 버퍼된 데이터를 보관하고 있을 수도 있다.

비록 I/O 장치(28)가 I/O 버스(32)의 제어를 포기할 수는 있어도, 시스템 버스(76)를 통하여 시스템 메모리(24),(26)로 버퍼된 데이터의 기록 전송을 종료하기 위해서(버스 인터페이스 유닛(64)를 통하여) 시스템 버스(76)의 제어를 유지하여야 한다. 버스 인터페이스 유닛의 중재 제어 논리 회로(130)는 이 조건을 인식하여 CACP 회로(62)로 CACP 오버라이드 신호(146)를 활성화시켜(버스 인터페이스 유닛(64)를 통하여) 시스템 버스(76)의 제어를 유지하도록 요청한다. 따라서, I/O 장치로부터 FIFO 버퍼(124)에 버퍼된 기록 데이터가 시스템 메모리(24),(26)로 전송되는 동안 CACP 회로(62)는 이와 동시에 중재 사이클을 시작한다. 중재 제어 논리 회로(130)가 CACP 오버라이드 신호(146)의 활성화를 지속하는 한 CACP 회로(62)는 시스템 버스(76)에 대한 액세스를 다른 I/O 장치(28)로 승인하지 않는다. 이는 버퍼된 마지막 기록 데이터가 시스템 메모리로 전송될 때까지 지속된다. 이후, 중재 제어 논리 회로(130)는 CACP 오버라이드 신호(146)를 비활성화 하여, CACP 회로(62)가 중재를 종료하고 새로운 I/O 장치에 I/O 버스(32)를 승인하는 것을 허용한다. 이는 시스템 버스(76)에 대한 경쟁적 동작을 방지한다. I/O 버스(32)를 제어하는 I/O 장치는 오직 CACP 회로(62)가 승인 모드일 경우에만 I/O 버스상에서 사이클을 개시할 수도 있다.

버스 인터페이스 유닛(64)내의 중재 제어 논리 회로(130)가 동작하는 제2조건은 I/O 장치(28)가 시스템 메모리(24),(26)로부터 데이터를 판독하고자 할 때 발생한다. 중재 사이클이 종료된 후, CACP 회로(62)는 I/O 버스(32) (및 버스 인터페이스 유닛을 통하여 시스템 버스(76))의 제어를 I/O 장치(28)에 승인한다. I/O 장치(28)는 하나 또는 둘 이상의 판독 사이클을 시스템 메모리(24),(26)에 대해 개시한다. 상기 FIFO 버퍼(124)를 기술할 때 설명된 방식으로 FIFO 버퍼(125C) 및/또는 (125D)는 프리페치된 인접 데이터로 계속 채워진다. I/O 장치(28)는 일단 FIFO 버퍼(125A) 및/또는 (125B)로부터 마지막 바이트를 판독하면 I/O 버스(32)의 제어를 포기한다. 따라서, 버퍼(125C),(125D)로 프리페치된 데이터는 필요없게 되므로 이런 불필요한 데이터를 효과적으로 제거하기 위하여 FIFO(124)를 반드시 리셋(reset)되어야 한다.

비록 I/O 장치(28)가 언제든지 I/O 버스(32)의 제어를 포기할 수는 있지만, 메모리 제어기(58)가 시스템 메모리(24),(26)로부터 버스 인터페이스 유닛(64)로의 현재 데이터 전송을 종료한 것을 나타낼 때까지(버스 인터페이스 유닛(64)를 통하여 시스템 버스(76)의 제어를 유지하여야 한다. 버스 인터페이스 유닛내의 중재 제어 논리 회로(130)는 이 조건을 인식하여 CACP 회로(62)로 CACP 오버라이드 신호(146)를 활성화시켜 마지막 프리페치 작용이 종료되고 FIFO 버퍼(124)가 리셋될 때까지(버스 인터페이스 유닛(64)를 통하여) 시스템 버스(76)의 제어를 유지하도록 요청한다. 따라서, 버스 인터페이스 유닛(64)가 시스템 버스(76)를 통하여 시스템 메모리로부터 마지막 프리페치 동작을 종료하고 FIFO 버퍼(124)를 리셋시킬 동안에, CACP 회로(62)는 이와 동시에 I/O 버스(32)상에서 새로운 중재 사이클을 시작한다. 중재 제어 논리 회로(130)가 CACP 오버라이드 신호(146)의 활성화를 지속하는 한, CACP 회로(62)는 시스템 버스(76)에 대한 액세스를 다른 I/O 장치(28)로 승인하지 않는다. 이는 프리페치 및 리셋 동작이 시스템 버스를 통하여 완료될 때까지 지속되고, 이후에 중재 제어 논리 회로(130)는 CACP 오버라이드 신호(146)를 비활성화 하여 CACP 회로(62)가 중재를 종료하고 새로운 I/O 장치(28)에 I/O 버스(32)를 승인하는 것을 허용한다. 이는 시스템 버스에 대한 경쟁적 동작을 방지한다.

버스 인터페이스 유닛내의 중재 제어 논리 회로(130)가 동작하는 제3조건은 시스템 버스(76)를 제어하고 있는 CPU(38)와 같은 시스템 장치가 I/O 버스(32)상의 슬레이브로서 동작하는 I/O 장치에 데이터를 기록하고자 할 때 발생한다. 중재 사이클의 종료 후, CACP 회로(62)는 시스템 버스(76) (및 버스 인터페이스 유닛(64)를 통하여 I/O 버스(32))의 제어를 시스템 장치에 승인하며, 이 시스템 장치는 시스템 버스-I/O 버스 변환 논리 회로(106)를 통하여 자신의 기록 사이클을 I/O 장치에 대해 개시한다.

시스템 버스-I/O 버스 변환 논리 회로(106)는 버퍼(도시되지 않음)를 제공하는데, 이 버퍼내에 CPU(38)와 같은 시스템 장치로부터 I/O 슬레이브 장치상의 확장 메모리로 기록되어질 데이터가 일시적으로 저장된 후 I/O 슬레이브 장치로 기록된다. 마지막 전송 데이터가 변환 논리 회로(106)에 저장되면 시스템 버스(76)는 더이상 필요치 않다. CPU(38)는 버스 인터페이스 유닛(64)가 그러한 것처럼 시스템 버스(76)가 더이상 필요없음을 표시한다. 하지만, 중재 제어 논리 회로(130)는 CACP 오버라이드 신호(146)를 활성화하여 자신이 I/O 버스(32)의 제어를 유지할 필요가 있음을 CACP 회로(62)에 알린다. 이때, 변환 논리 회로(106)에 버퍼된 데이터가 I/O 버스(32)를 통하여 I/O 장치에 아직 기록되어야 할지라도 CACP 회로(62)는 I/O 버스(32)에 대한 중재를 시작할 수도 있다. I/O 장치와 달리, CPU(38)는 CACP 회로(62)가 중재 모드이든 또는 승인모드이든 어느 한 모드동안 I/O 장치에 기록할 수 있기 때문이다.

따라서, 변환 논리 회로(106)에 버퍼된 데이터가 I/O 장치에 기록될 동안, CACP 회로(62)는 이와 동시에 중재 사이클을 수행한다. 이후, 중재 제어 논리 회로(130)는 CACP 오버라이드 신호(146)를 비활성화 하여, CACP 회로(62)가 중재를 종료하고 새로운 I/O 장치(28)에 I/O 버스(32)를 승인하는 것을 허용한다. 이는 CPU(38)와 I/O 장치(28)간의 I/O 버스에 대한 경쟁적 동작을 방지한다.

제5도는 버스 인터페이스 유닛(64)내의 중재 제어 논리 회로(130)의 일실시예를 도시한다. 중재 제어 논리 회로(130)는 버스 인터페이스 유닛(64)의 하드웨어로 구축되는 알고리즘에 의해 구현된다. 중재 제어 논리 회로(130)는 AND 게이트(148) 및 S-R 래치(150),(152)를 포함한다. 전술한 바와 같이, I/O 버스(32)를 제어하고 있는 I/O 장치가 I/O 및 시스템 버스(32),(76)를 통하여 시스템 메모리(24),(26)에 대해서 데이터의 다중 전송을 판독하거나 또는 기록할 경우, 중재 제어 논리 회로(130)는 CACP 오버라이드 신호(146)를 출력한다. CACP 오버라이드 신호(146)는 음일 때 활성화(negative active)된다. 따라서, I/O 장치가 시스템 메모리(24),(26)로부터 판독할 동안, 데이터가 시스템 메모리로부터 FIFO 버퍼(124)로 프리페치되고, 판독 프리페치 시작(read prefetch begin)은



라인(154)이 하이(HIGH)로 되어 래치(150)를 세트(set)한다. 래치(150)의 보수 출력(complemented output)(156)이 로우(LOW)가 되며, 이는 CACP 오버라이드 신호(146)를 활성화시켜, 데이터가 시스템 버스(76)를 통하여 판독 프리페치 동작이 종료될 때까지(버스 인터페이스 유니트(64)를 통하여) 시스템 버스의 제어를 유지한다. 판독 프리페치 동작이 종료될 경우, 라인(158)은 하이로 되고, 래치(150)가 리셋되며, CACP 오버라이드 신호(146)가 비활성화 된다. 지금까지 중재 모드에 진입해 있었던 CACP 회로(62)는 다음 승인 모드로 진입하는 것이 허용된다.

I/O 장치가 시스템 메모리(24),(26)에 기록할 동안, 데이터는 먼저 FIFO 버퍼(124)에 기록되고, 시스템 메모리 기록 라인(write to system memory line)(160)이 하이가 되어 래치(152)를 세트한다. 래치(152)의 보수출력(162)이 로우가 되어, CACP 오버라이드 신호(146)가 활성화 되고, FIFO 버퍼(124)로부터 시스템 버스(76)를 통하여 시스템 메모리에 데이터가 기록될 동안 CACP 회로(62)가 다음 승인 사이클로 진입하지 못하게 한다. I/O 장치는 시스템 메모리로의 기록이 종료될 때까지(버스 인터페이스 유니트(64)를 통하여) 시스템 버스의 제어를 유지한다. 그렇지만 I/O 버스는 CACP 회로(62)에 의한 동시적 중재 처리에 이용할 수 있다. CACP 오버라이드 신호(146)는 CACP 회로가 승인 모드로의 재진입을 방지한다. 기록 동작이 완료되면 FIFO 버퍼(124)내의 모든 버퍼는 비게 되며, 이는 라인(164)을 하이로 하여 래치(152)를 리셋시킨다. CACP 오버라이드 신호(146)는 비활성되고, CACP 회로(62)가 승인 모드로 진입할 수 있게 된다.

시스템 장치(예를 들면 CPU(38))가 I/O 장치(28)에 기록하는 동안, 데이터는 I/O 버스 변환 논리 회로(106)에 의해 제공된 버퍼에 일시적으로 저장된다. 데이터를 저장할 때 I/O 버스 변환 논리 회로(106)는 포스트된(posted) CPU 사이클 신호(166)를 로우로 하여 CACP 오버라이드 신호(146)를 활성화시킨다. 또한 기록 사이클이 종료되었음을 시스템 버스(76)에 알린다. 이는 I/O 버스 변환 논리 회로(106)가 버퍼된 데이터의 I/O 장치(28)에로의 기록을 종료할 동안 시스템 버스(76)상에서 부가적인 동작이 발생할 수 있도록 허용한다. I/O 장치(28)로의 기록 동작이 종료되면, I/O 버스 변환 논리 회로(106)는 CPU 포스트된 사이클 신호(166)를 하이로 구동함으로써 CACP 오버라이드 신호(146)를 비활성화 하며, 이는 CACP 회로(62)가 중재를 종료하고 I/O 버스(32)를 새로운 I/O 장치(28)로 넘기는 것을 허용한다.

지금까지, 이중 버스 구조를 갖는 컴퓨터의 버스 제어 논리의 바람직한 실시예를 설명하였으나, 진술한 설명을 염두에 두고 볼 때, 그같은 설명은 하나의 예에 불과한 것으로서, 본 발명은 본 명세서 내에 설명한 특정 실시태양에 국한되지 않으며, 특허청구 범위에 의해 규정되는 본 발명의 진정한 사상을 벗어나지 않는 범위내에서 각종 재구성, 변형 및 대체가 가능함을 알아야 할 것이다.

## (57) 청구의 범위

### 청구항 1

시스템 메모리와 ; 상기 시스템 메모리에 메모리 버스에 의해서 접속되고, 상기 시스템 메모리에 대한 액세스를 제어하기 위한 메모리 제어기(memory controller)와 ; 상기 메모리 제어기와 전기적으로 접속되며, 상기 메모리 버스를 통해 상기 시스템 메모리에 대한 데이터 판독 및 기록을 행할 수 있는 중앙 처리 장치(CPU)와 ; 상기 메모리 제어기에 시스템 버스에 의해 전기적으로 접속되고 또한 입력/출력 버스를 통하여 판독 및 기록 동작(read and write operations)을 개시할 수 있는 다수의 입력/출력 장치에 상기 입력/출력 버스에 의해서 전기적으로 접속된 버스 인터페이스 유니트(bus interface unit)와 ; 상기 시스템 버스상에 위치한 중앙 중재 제어기(central arbitration control point)를 포함하고 ; 상기 버스 인터페이스 유니트는 상기 입력/출력 장치중의 하나가 상기 입력/출력 버스를 통해 판독 또는 기록 동작을 종료한 시점을 감지할 수 있고 또한 상기 버스 인터페이스 유니트를 통해 상기 시스템 버스와 상기 입력/출력 버스간에 전송되는 판독 및 기록 데이터가 전송되는 동안 일시적으로 저장되는 버퍼회로를 구비하며 ; 상기 중앙 중재 제어기는, 연속적으로 ( i ) 상기 중앙 중재 제어기가 상기 다수의 입력/출력 장치와 상기 중앙 처리 장치간을 중재하여 상기 입력/출력 장치중의 하나 또는 상기 중앙 처리 장치중 어떠한 것에 입력/출력 버스에 대한 제어를 승인할 것인지를 결정하는 중재 사이클(arbitration cycles)과 ( ii ) 상기 중앙 중재 제어기가 상기 입력/출력 버스의 제어를 승인하고 상기 시스템 버스의 제어를 상기 입력/출력 장치중의 하나 또는 상기 중앙 처리 장치로 연장 승인하는 승인 사이클(grant cycles)을 수행하며 ; 상기 중앙 중재 제어기는 사전 설정된 세트의 동작 조건에 응답하는 중재 제어 논리(arbitration control logic)에 의해 적어도 부분적으로 제어되어, 데이터 전송 동작이 상기 중앙 중재 제어기의 동작과 동시에 발생되게 허용하는 컴퓨터 시스템.

### 청구항 2

제1항에 있어서, 상기 중재 제어 논리가 상기 버스 인터페이스 유니트내의 하드웨어 구축되는 알고리즘으로 구현되는 컴퓨터 시스템.

### 청구항 3

제1항에 있어서, 상기 사전 설정된 세트의 동작 조건중의 하나는 상기 입력/출력 장치중의 하나가 상기 입력/출력 버스 및 상기 시스템 버스를 통해 상기 시스템 메모리로부터 데이터를 판독하는 경우 발생하는 컴퓨터 시스템.

### 청구항 4

제1항에 있어서, 상기 사전 설정된 세트의 동작 조건중의 하나는 상기 입력/출력 장치중의 하나가 상기 입력/출력 버스 및 상기 시스템 버스를 통해 상기 시스템 메모리로부터 데이터를 기록하는 경우 발생하는 컴퓨터 시스템.

## 청구항 5

제1항에 있어서, 상기 사전 설정된 세트의 동작 조건중의 하나는 상기 중앙 처리 장치가 입력/출력 버스상에서 슬레이브(slave)로서 작용하는 상기 입력/출력 장치중의 하나에 데이터를 기록하는 경우 발생하는 컴퓨터 시스템.

## 청구항 6

제1항에 있어서, 상기 버스 인터페이스 유닛은 상기 시스템 버스와 상기 입력/출력 버스간의 데이터 정보를 일시적으로 저장하는 이중 포트 비동기 양방향성 저장 유닛(dual ported asynchronous bidirectional storage unit)을 포함하는 컴퓨터 시스템.

## 청구항 7

제3항에 있어서, 상기 이중 포트 비동기 양방향성 저장 유닛은 판독 및 기록 동작 모드에 사용되는 적어도 두쌍의 16바이트 버퍼를 포함하는 컴퓨터 시스템.

## 청구항 8

제4항에 있어서, 상기 저장 유닛이 판독 및 기록 동작 모두에 사용되고 있는 적어도 2쌍의 16바이트 버퍼를 포함하는 컴퓨터 시스템.

## 청구항 9

제1항에 있어서, 상기 시스템 버스는 16바이트까지의 대역폭으로 상기 버스 인터페이스 유닛과 상기 시스템 메모리간의 판독 또는 기록 데이터의 버스트 전송(burst transfer)을 지원하며, 상기 입력/출력 버스는 1, 2 또는 4바이트 대역폭으로 상기 입력/출력 장치와 상기 버스 인터페이스 유닛간의 판독 또는 기록 데이터의 전송을 지원하는 컴퓨터 시스템.

## 청구항 10

컴퓨터 시스템에 있어서 입력/출력 장치와 중앙 처리 장치간을 중재하는 방법으로서, 시스템 메모리와, 상기 시스템 메모리에 메모리 버스에 의해서 접속되고 상기 시스템 메모리에 대한 액세스를 제어하기 위한 메모리 제어기를 제공하는 단계와 ; 상기 메모리 제어기와 전기적으로 접속되며, 상기 메모리 버스를 통해 상기 시스템 메모리에 대한 데이터 판독 및 기록을 행할 수 있는 중앙 처리 장치를 제공하는 단계와 ; 버스 인터페이스 유닛을 제공하는 단계로서, 상기 버스 인터페이스 유닛은 상기 메모리 제어기에 시스템 버스에 의해서 전기적으로 접속되고 또한 입력/출력 버스를 통해 판독 및 기록 동작을 개시할 수 있는 다수의 입력/출력 장치에 상기 입력/출력 버스에 의해서 전기적으로 접속되며, 상기 버스 인터페이스 유닛은 상기 입력/출력 장치중의 하나가 상기 입력/출력 버스를 통해 판독 또는 기록 동작을 종료한 시점을 감지할 수 있으며 또한 상기 버스 인터페이스 유닛을 통해 상기 시스템 버스와 상기 입력/출력 버스간에 전송되는 판독 및 기록 데이터가 전송되는 동안 일시적으로 저장되는 버퍼회로를 구비하는, 상기 버스 인터페이스 유닛을 제공하는 단계와 ; 상기 시스템 버스상에 위치한 중앙 중재 제어기를 제공하는 단계로서, 상기 중앙 중재 제어기는, 연속적으로 ( i ) 상기 중앙 중재 제어기가 상기 다수의 입력/출력 장치와 상기 중앙 처리 장치간을 중재하여 상기 입력/출력 장치중의 하나 또는 상기 중앙 처리 장치중 어떠한 것에 입력/출력 버스에 대한 제어를 승인할 것인지를 결정하는 중재 사이클과 ( ii ) 상기 중앙 중재 제어기가 상기 입력/출력 버스의 제어를 승인하고 상기 시스템 버스의 제어를 상기 입력/출력 장치중의 하나 또는 상기 중앙 처리 장치로 연장 승인하는 승인 사이클을 수행하는, 상기 중앙 중재 제어기를 제공하는 단계와 ; 사전 설정된 세트의 동작 조건에 응답하는 중재 제어 논리에 의해 상기 중앙 중재 제어기를 적어도 부분적으로 제어하여 중앙 중재와 동시에 데이터를 전송하는 단계를 포함하는 중재 방법.

## 청구항 11

제10항에 있어서, 상기 시스템 버스는 16바이트까지의 대역폭으로 상기 버스 인터페이스 유닛과 상기 시스템 메모리간의 판독 또는 기록 데이터를 전송하고, 상기 입력/출력 버스는 1, 2 또는 4바이트의 대역폭으로 상기 입력/출력 장치와 상기 인터페이스 유닛간의 판독 또는 기록 데이터를 전송하는 중재 방법.

## 청구항 12

제10항에 있어서, 상기 버스 인터페이스 유닛은 상기 시스템 버스와 상기 입력/출력 버스간에 전송되는 데이터 정보를 일시적으로 저장하는 이중 포트 비동기 양방향성 저장 유닛을 포함하는 중재 방법.

## 청구항 13

제12항에 있어서, 상기 이중 포트 비동기 양방향성 저장 유닛은 판독 및 기록 동작 모두에 사용되는 적어도 두쌍의 16바이트 버퍼를 포함하는 중재 방법.

## 청구항 14

제12항에 있어서, 상기 사전 설정된 세트의 동작 조건중의 하나는 상기 입력/출력 장치가 상기 입력/출력 버스 및 상기 시스템 버스를 통해 상기 시스템 메모리로부터 데이터를 판독하는 경우 발생하는 중재 방법.

## 청구항 15

제12항에 있어서, 상기 사전 설정된 세트의 동작 조건중의 하나는 상기 입력/출력 장치중의 하나가

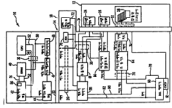
상기 입력/출력 버스 및 상기 시스템 버스를 통해 상기 시스템 메모리로부터 데이터를 기록하는 경우 발생하는 중재 방법.

## 청구항 16

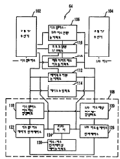
제12항에 있어서, 상기 사전 설정된 세트의 동작 조건중의 하나는 상기 중앙 처리 장치가 상기 입력/출력 버스상에서 작용하는 상기 입력/출력 장치중의 하나에 데이터를 기록하는 경우 발생하는 중재 방법.

## 도면

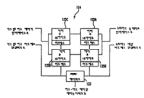
도면1



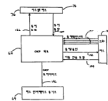
도면2



도면3



도면4



도면5

