

(19) 대한민국특허청(KR)  
(12) 특허공보(B1)

(51) Int. Cl.<sup>6</sup>  
G06F 13/28

(45) 공고일자 1997년01월08일  
(11) 공고번호 97-000274

(21) 출원번호	특1992-0024701	(65) 공개번호	특1993-0016888
(22) 출원일자	1992년12월17일	(43) 공개일자	1993년08월30일
(30) 우선권 주장	816,691 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 셔우드 브랜넨 미합중국 플로리다 33487 보카 래턴 웨스트 컨트리 클럽 블러바드 7360 리차드 루이스 혼 미합중국 플로리다 33437 보인턴 비치 아파트먼트 넘버 8-23 세다 레이 크 로드 5289 테렌스 죠셉 로만		
(74) 대리인	김창세, 김영, 장성구		

**심사관 : 오홍수 (책자공보 제4774호)**

**(54) 컴퓨터 시스템 및 그의 데이터 전송을 제어하는 방법**

**요약**

내용 없음.

**대표도**

**도1**

**명세서**

[발명의 명칭]

컴퓨터 시스템 및 그의 데이터 전송을 제어하는 방법

[도면의 간단한 설명]

제1도는 본 발명의 원리에 따라 구성된 버스 인터페이스 유닛을 구비한 컴퓨터 시스템의 개략적인 블록도.

제2도는 제1도에 도시된 컴퓨터 시스템의 버스 인터페이스 유닛의 개략적인 블록도.

제3도는 제2도에 도시된 버스 인터페이스 유닛의 FIFO형 버퍼에 대한 개략적인 블록도.

제4도는 제3도의 버스간 페이싱 로직의 일실시예를 구현하는데 사용된 제어 로직의 회로도.

제5도는 제3도의 버스간 페이싱 로직의 다른 실시예를 구현하는데 사용된 제어 로직의 회로도.

\* 도면의 주요부분에 대한 부호의 설명

10 : 컴퓨터 시스템	12 : 시스템 보드
14 : 프로세서 복합체	16 : 프로세서부
18 : 베이스부	20 : 로컬 버스
22 : 로컬 버스 접속기	24,26 : 시스템 메모리
28 : 입/출력 장치	30 : 메모리 버스
32 : 입/출력 버스	34 : 프로세서 복합체 접속기
38 : 중앙 처리 장치	40 : 스택틱 랜덤 액세스 메모리

42 : 캐쉬 제어 모듈	44 : 주파수 제어 모듈
46 : 어드레스 버퍼	48 : 데이터 버퍼
50,78 : 데이터 정보 경로	52,80 : 어드레스 정보 경로
54,82 : 제어 정보 경로	56 : 랜덤 액세스 메모리
58 : 메모리 제어기	60 : DMA 제어기
62 : 중앙 중재 제어 포인트	64 : 버스 인터페이스 유닛
66 : 버퍼/에러 정정 코드	68 : 구동 회로
70 : 판독 전용 메모리	72 : 셀프 테스트 회로
74 : 버퍼 회로	76 : 시스템 버스
102,104 : 구동기/수신기	
106 : 시스템 버스-입/출력 버스 변환 로직	
108 : 입/출력 버스-시스템 버스 변환 로직	
110 : 메모리 어드레스 비교 로직	112 : 에러 복구 지원 로직
114 : 캐쉬 스누핑 회로	116 : 프로그램된 입/출력 회로
118 : 시스템 버스 어드레스 발생 회로	
120 : 입/출력 버스 예상 어드레스 발생 회로	
122 : 시스템 버스 제어기 인터페이스	123 : FIFO 제어회로
124 : FIFO 버퍼	125A, 125B, 125C, 125D : 버퍼
126 : 입/출력 버스 슬레이브 인터페이스	
128 : 버스간 페이싱 제어 로직	

#### [발명의 상세한 설명]

본 발명은 이중 버스 구조를 갖는 컴퓨터 시스템에서의 버스간 인터페이스(bus to bus interfaces)에 관한 것으로, 특히, 두 시스템 버스간에 전송되는 데이터를 일시적으로 저장하기 위한 버스간 인터페이스 유닛 및 방법에 관한 것이다.

일반적으로 컴퓨터 시스템, 특히 퍼스널 컴퓨터 시스템내에서, 데이터는 중앙 처리 장치(CPU), 메모리 장치, 직접 메모리 액세스(DMA) 제어기와 같은 각종 시스템 장치들간에 전송된다. 또한 입/출력(I/O) 장치와 같은 확장 장치(expansion element)들간에 및 이들 입/출력 장치와 각종 시스템 장치들간에 데이터가 전송된다. 입/출력 장치들과 시스템 장치들은 컴퓨터 버스를 통해 상호간에 통신하며, 이들 컴퓨터 버스는 수개의 발신지중 어느 하나의 발신지로부터 수개의 수신지중 어느 하나의 수신지로 정보를 전송하는 일련의 도체들로 구성된다. 대다수의 시스템 장치와 입/출력 장치는 버스 제어기(bus controllers)(즉, 컴퓨터 시스템을 제어할 수 있는 장치)와 버스 슬레이브(bus slaves)(즉, 버스 제어기에 의해 제어되는 장치)로서 동작 할 수 있다.

둘 이상의 버스를 갖는 퍼스널 컴퓨터 시스템이 이미 공지되어 있으며, 통상적으로, CPU가 캐쉬 메모리 또는 메모리 제어기와 통신하는 로컬 버스(local bus)와, DMA 제어기 또는 입/출력 장치와 같은 시스템 버스 장치가 메모리 제어기를 통해 시스템 메모리와 통신하는 시스템 입/출력 버스가 제공된다. 이러한 시스템 입/출력 버스는 버스 인터페이스 유닛에 의해 상호 접속되는 시스템 버스와 입/출력 버스를 포함한다. 입/출력 장치들은 입/출력 버스를 통해 상호 통신한다. 이들 입/출력 장치는 또한 시스템 메모리와 같은 시스템 버스 장치와도 통신할 수 있어야만 하는 것이 통상적이다. 이러한 통신은 입/출력 버스와 시스템 버스 및 이들간의 버스 인터페이스 유닛을 통해 이루어져야만 한다.

시스템 버스와 입/출력 버스간의 데이터 전송시에는 흔히 상기 버스중 어느 한쪽 또는 양쪽에 결합된, 현저히 다른 속도 및 사이한 데이터 전송 모드에서 동작하는 장치를 수용할 수 있어야 할 필요가 있다. 예를 들면, 입/출력 버스에 1,2 및 4바이트의 대역 폭으로 기입 동작을 행하는 장치가 결합될 수도 있다. 한편, 시스템 버스는 매우 신속한 버스트 전송(burst transmissions)으로서 공지된 16바이트 정보 패킷을 전송할 수도 있다. 더욱이, 연속적인 어드레스로부터 비교적 많은 양의 데이터를 전송하는 것이 보통 바람직하다. 이러한 전송은, 각 어드레스 영역에 특정한 요구를 개시할 필요없이 성취될 경우에 바람직하며 시간 소모가 더욱 적다.

따라서, 시스템 버스와 입/출력 버스간의 모든 상호 접속은 상이한 속도 및 상이한 모드의 데이터 전송을 처리할 수 있어야 할 필요가 있다. 더욱이, 각각의 데이터 영역에 대해 개별적인 전송 요구를 개시하지 않고 연속적인 어드레스로부터 데이터가 전송되는 것이 바람직하다.

이에 본 발명의 목적은 상이한 전송 속도 및 상이한 모드에서도 효과적이고 효율적으로 데이터를 전송하는, 시스템 버스와 입/출력 버스간의 효율적인 데이터 버퍼를 제공하는 것이다.

본 발명에 따르면, 시스템 버스 및 입/출력 버스와 같은 이중 버스 구조를 갖는 컴퓨터 시스템용 버스간 인터페이스 유닛이 제공된다. 이 버스 인터페이스 유닛은 이들 두 버스 각각에 결합된 장치로 및 그로부터 이들 두 버스간에 전송되는 데이터를 위한 비동기 양방향성 일시적 데이터 저장 기능을

포함한다. 바람직하기로는, 이 저장 기능은 개별적인 데이터 전송, 데이터 스트림 전송 및 데이터 버스트 전송을 수용하는 모드로 동작하며, 각각의 어드레스에 대해 새로운 요구를 개시하지 않고 연속적인 어드레스로부터의 정보 전송을 수용할 수 있다.

이하 본 발명을 첨부된 도면을 참조하여 보다 상세히 설명한다.

우선 제1도를 참조하면, 시스템 보드(12)와 프로세서 복합체(processor complex)(14)로 이루어진 컴퓨터 시스템(10)이 도시된다. 프로세서 복합체(14)는 프로세서부(16)와, 로컬 버스 접속기(local bus connector)(22)를 통해 프로세서 로컬 버스(20)에 접속된 베이스부(18)로 이루어진다. 프로세서부(16)는 50MHz로 동작하며, 베이스부(18)는 40MHz로 동작한다.

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, 입/출력 장치(28)와 프로세서 복합체(14)간의 통신은 입/출력 버스(32)에 의해 행해진다. 입/출력 장치(28)와 시스템 메모리(24),(26)간의 통신은 입/출력 버스(32), 시스템 버스(76) 및 메모리 버스(30)에 의해 행해진다. 입/출력 버스(32)는 마이크로 채널(MICRO CHANNEL) 컴퓨터 구조에 따를 수도 있다. 메모리 버스(30)와 입/출력 버스(32)는 프로세서 복합체 접속기(processor complex connector)(34)에 의해 프로세서 복합체 베이스부(18)에 접속된다. 메모리 확장 장치와 같은 입/출력 장치는 입/출력 버스(32)를 통해 컴퓨터 시스템(10)에 접속될 수도 있다. 시스템 보드(12)는 정규 동작중 컴퓨터 시스템(10)에 의해 사용될 수도 있는 통상적인 비디오 회로, 타이밍 회로, 키보드 제어 회로 및 인터럽트 회로(이들 모두는 도시되지 않음)를 포함할 수도 있다.

프로세서 복합체(14)의 프로세서부(16)는 중앙 처리 장치(CPU)(38)를 포함하며, 이 CPU는, 본 바람직한 실시예의 경우, 상표명 i486으로 인텔(Intel)사로부터 구입 가능한 32-비트 마이크로 프로세서이다. 프로세서부(16)는 또한 스태틱 랜덤 액세스 메모리(static random access memory; SRAM)(40), 캐쉬 제어 모듈(cache control module)(42), 주파수 제어 모듈(frequency control module)(44), 어드레스 버퍼(address buffer)(46) 및 데이터 버퍼(data buffer)(48)를 포함한다. 로컬 버스(20)는 데이터 정보 경로(50), 어드레스 정보 경로(52) 및 제어 정보 경로(54)로 구성된다. 데이터 정보 경로(50)는 CPU(38), SRAM(40) 및 데이터 버퍼(48)간에 제공된다. 어드레스 정보 경로(52)는 CPU(38), 캐쉬 제어 모듈(42) 및 어드레스 버퍼(46)간에 제공되며, 제어 정보 경로(54)는 CPU(38), 캐쉬 제어 모듈(42) 및 주파수 제어 모듈(44)간에 제공된다. 또한, 캐쉬 제어 모듈(42)과 SRAM(40)간에는 어드레스와 제어 정보 경로가 제공된다.

SRAM(40)은 시스템 메모리(24) 또는 (26)로부터, 또는 입/출력 장치(28)상에 위치한 확장 메모리로부터의 메모리 정보를 단기간 저장하는 캐쉬 기능을 제공한다. 캐쉬 제어 모듈(42)은 이들 시스템 메모리(24),(26)의 어드레스 위치를 저장하는 랜덤 액세스 메모리(RAM)(56)를 구비한다. CPU(38)는 로컬 버스(20)를 통해 SRAM(40)에 캐싱되어 있는 정보를 직접 액세스할 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 의해 정보가 포착되는 시점과 이들 버퍼에 저장된 정보가 오버라이트(overwrite)되는 시점을 결정한다. 버퍼(46),(48)는 시스템 메모리(24),(26)로부터의 두 개의 기입 데이터가 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 양방향성이므로, 표준 베이스부(18)는 그대로 둔채 프로세서 복합체(14)의 프로세서부(16)가 대체 또는 개량될 수도 있다.

베이스부(18)는 메모리 제어기(58), DMA(direct memory access) 제어기(60), 중앙 중재 제어 포인트(central arbitration control point)(CACP) 회로(62), 버스 인터페이스 유닛(64) 및 버퍼/에러 정정 코드(buffer/error correction code)(ECC) 회로(66)를 포함한다. 또한, 베이스부(18)는 구동 회로(68), 판독 전용 메모리(ROM)(70), 셀프 테스트 회로(72) 및 버퍼(74)를 포함한다. 시스템 버스(76)는 데이터 정보 경로(78), 어드레스 정보 경로(80) 및 제어 정보 경로(82)로 이루어진다. 데이터 정보 경로(78)는, 버퍼(74)와 버스 인터페이스 유닛(64)을, 버스 인터페이스 유닛(64)과 DMA 제어기(60) 및 버퍼/ECC 회로(66)를, 버퍼/ECC 회로(66)와 시스템 메모리(24),(26)를 접속한다. 어드레스 정보 경로와 제어 정보 경로는 각각 메모리 제어기(58)와 DMA 제어기(60) 및 버스 인터페이스 유닛(64)을 접속하고, 버스 인터페이스 유닛(64)과 버퍼(74)를 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)를 대신하여 버스 인터페이스 유닛(64)에 메모리 버스(30)를 통한 시스템 메모리(24),(26)로의 액세스를 제공한다. 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)를 대신한 버스 인터페이스 유닛(64)은 메모리 제어기(58)를 통해 시스템 메모리(24),(26)를 액세스한다. DMA 제어기(60) 또는 I/O 장치(28)를 대신한 버스 인터페이스 유닛(64)은 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) 및 메모리 버스(30)를 통해 시스템 메모리와 통신한다.

CPU(38)의 입/출력 버스(32)에 대한 판독 또는 기입 사이클에 있어서, 어드레스 정보가 시스템 메모리 어드레스 경계에 대해 체크된다. 이때, 어드레스 정보가 입/출력 확장 메모리 어드레스 또는 입/출력 포트(port) 어드레스와 일치하면, 메모리 제어기(58)는(버스 인터페이스 유닛(64)을 통해) 입/출력 버스(32)상으로 입/출력 장치(28)에 대한 입/출력 메모리 사이클 또는 입/출력 포트 사이클을 개시한다. CPU의 입/출력 장치에 대한 메모리 사이클 또는 입/출력 포트 사이클 동안 메모리 제어기(58)에 제공되는 어드레스는 시스템 버스(76)와 입/출력 버스(32) 사이에 존재하는 버스 인터페이스 유닛(64)을 통해 시스템 버스(76)로부터 입/출력 버스(32)로 전송된다. 이 어드레스에 대응하는 확장 메모리를 포함하는 입/출력 장치(28)는 입/출력 버스(32)로부터 메모리 어드레스를 수신한다.

DMA 제어기(60)와 버스 인터페이스 유닛(64)은 시스템 메모리(24),(26)와 입/출력 장치(28)내 확장 메모리간의 정보 교환을 제어한다. DMA 제어기(60)는 또한 프로세서 복합체(14)를 위해 3가지 기능을 제공한다. 첫째, DMA 제어기(60)는 소규모 컴퓨터 서브시스템 제어 블록(small computer subsystem control block : SCB) 구조를 채용하여 DMA 채널을 구성함으로써, 이러한 DMA 채널을 구성하기 위해 프로그램된 입/출력 장치를 사용할 필요성을 제거한다. 둘째, DMA 제어기(60)는 저 속의 메모리 확장 장치와, 통상적으로 그보다 빠른 시스템 메모리간의 데이터 전송을 최적화 하기 위한 버퍼링 기능을 제공한다. 셋째, DMA 제어기(60)는 8채널의 32비트 직접 시스템 메모리 액세스 기능을 제공한다. 직접 시스템 메모리 액세스 기능을 제공할 때, DMA 제어기(60)는 다음의 두 모드로 동작될 것이다. 제1모드에서, DMA 제어기(60)는, CPU(38)에 대해 기능상 슬레이브(slave)가 되는 프로그램된 입/출력 모드로 동작한다. 제2모드에서, DMA 제어기(60)는 시스템 버스 마스터(master)로서 동작하여, 입/출력 버스(32)를 중재하고 제어한다. 이러한 제2모드 동안, DMA 제어기(60)는 선입 선출(first in, first out ; FIFO)형 레지스터 회로를 사용한다.

중앙 중재 제어 포인트(CACP) 회로(62)는 DMA 제어기, 입/출력 장치 버스 제어기 및 CPU(I/O 장치를 액세스하는 경우)에 대해 중재기로서 작용한다. CACP 회로(62)는, DMA 제어기(60), 메모리 제어기(58) 및 입/출력 장치(28)로부터 중재 제어 신호를 수신하여, 어느 장치가 입/출력 버스(32)를 제어할 것인가 및, 특정의 장치가 입/출력 버스의 제어를 유지하게 되는 시간의 길이를 결정한다.

구동 회로(68)는 메모리 제어기(58)로부터 시스템 메모리(24),(26)로 제어 정보와 어드레스 정보를 제공한다. 또한, 구동 회로(68)는 시스템 메모리(24),(26)를 구성하는 사용되는 싱글 인-라인 메모리 모듈들(single in-line memory modules : SIMMs)의 수에 입각하여 이러한 정보를 구동한다. 따라서, 구동 회로(68)는 시스템 메모리(24),(26)의 크기에 따라 시스템 메모리(24),(26)에 제공되는 제어 및 어드레스 정보의 신호 세기를 변환시킨다.

버퍼 회로(74)는 프로세서 복합체의 베이스부(18)와 시스템 보드(12) 사이에서 증폭 및 절연(isolation)을 제공한다. 또한, 버퍼 회로(74)는 입/출력 버스(32)와 버스 인터페이스 유닛(64)간의 경계 정보(boundary information)를 실시간으로 포착할 수 있는 버퍼를 사용한다. 따라서, 컴퓨터 시스템(10)이 고장난 경우, 버퍼 회로(74)는 컴퓨터 수리 기사에 의해 액세스되어 시스템 고장시 프로세서 복합체 접속기(34)에 제공되었던 정보를 판단할 수도 있게 한다.

ROM(70)은 전원 투입시 확장 메모리로부터의 데이터를 시스템 메모리에 위치시켜 초기화 하므로써 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 동작할 준비가 되었는지를 판단하기 위해 베이스부(18)의 다른 주요 구성 요소를 테스트 한다.

제2도를 참조하면, 제1도에 도시된 시스템의 버스 인터페이스 유닛(64)에 대한 개략적인 블록도가 도시된다. 버스 인터페이스 유닛(64)은 시스템 버스(76)와 입/출력 버스(32)간에 양방향성 고속 인터페이스를 제공함으로써 본 발명의 구현을 위한 기초(base)를 제공한다.

버스 인터페이스 유닛(64)은 시스템 버스 구동/수신 회로(system bus driver /receiver circuit)(102), 입/출력 버스 구동/수신회로(104) 및 이들간에 전기적으로 접속된 제어 로직을 포함한다. 시스템 버스 구동/수신 회로(102)는 시스템 버스(76)로부터 신호를 수신하여 적절한 버스 인터페이스 유닛 제어 로직으로 보내며, 버스 인터페이스 유닛 제어 로직으로부터 신호를 수신하여 시스템 버스(76)로 보내는 조향 로직(steering logic)을 포함한다. 입/출력 버스 구동/수신 회로(104)는 입/출력 버스(32)로부터 신호를 수신하여 적절한 버스 인터페이스 유닛 제어 로직으로 보내며, 버스 인터페이스 유닛 제어 로직으로부터 신호를 수신하여 입/출력 버스(32)로 보내는 조향 로직을 포함한다.

버스 인터페이스 유닛 제어 로직은 시스템 버스-입/출력 버스 변환 로직(system bus to I/O bus translation logic) 회로(106), 입/출력 버스-시스템 버스 변환 로직(I/O bus to system 편 translation logic) 회로(108), 메모리 어드레스 비교 로직(110), 에러 복구 지원 로직(error recovery support logic)(112) 및 캐쉬 스누핑 로직(cache snooping logic)(114)을 포함한다. 프로그램된 입/출력 회로(programmed I/O circuit)(116)는 또한 시스템 구동/수신회로(102)와 전기적으로 결합된다.

시스템 버스-입/출력 버스 변환 로직(106)은, DMA 제어기(60) 또는 (CPU( 38)를 대신해서) 메모리 제어기(58)가 시스템 버스 제어기로서 동작하여 입/출력 버스(32)를 액세스하며 이에 의해 입/출력 버스상의 슬레이브 장치로서 작용하는 입/출력 장치(28)와 통신하는데 필요한 수단을 제공한다. 변환 로직(106)은 시스템 버스(76)의 제어, 어드레스 및 데이터 라인을, 입/출력 버스(32)의 제어, 어드레스 및 데이터 라인으로 변환한다. 대부분의 제어 신호와 모든 어드레스 신호는 시스템 버스(76)로부터 입/출력 버스(32)로 전송되며, 데이터 정보는 양방향으로 전송된다. 시스템 버스 슬레이브로서 동작하는 이러한 로직은 시스템 버스(76)를 감시하여 입/출력 버스(32)에 대한 사이클을 검출한다.

이러한 사이클의 검출시, 시스템 버스 슬레이브는 시스템 버스상의 신호 타이밍을 입/출력 버스 타이밍으로 변환하고, 입/출력 버스(32)에 대한 사이클을 개시하며, 이 사이클이 완료되기를 기다려, 시스템 버스(76)상의 사이클을 종료한다.

입/출력 버스-시스템 버스 변환 로직(108)은 시스템 버스 어드레스 발생 회로(118), 입/출력 버스 예상 어드레스 발생 회로(120), 시스템 버스 제어기 인터페이스(122), FIFO 버퍼(124), 입/출력 버스 슬레이브 인터페이스(126) 및 버스간 페이싱 제어 로직(bus to bus pacing control logic)(128)으로 구성된다. 시스템 버스 제어기 인터페이스(122)는 40MHz로 동작하는 고성능 32bit(4byte) i486 버스트 프로토콜을 지원한다. 버스트 모드에서는 4, 8 및 16바이트의 데이터 전송이, 비버스트 모드(no-burst mode)에서는 1 내지 4바이트의 데이터 전송이 제공된다. 입/출력 버스 슬레이브 인터페이스(126)는 시스템 버스(76)상의 슬레이브 장치에 대한 동작에 대하여 입/출력 버스(32)를 감시하며,

입/출력 버스(32)에 대한 동작들은 무시한다. 입/출력 버스 슬레이브 인터페이스(126)에 의해 픽업된(picked-up) 모든 사이클은 FIFO 버퍼(124)와 시스템 버스 제어기 인터페이스(122)로 전송된다.

입/출력 버스-시스템 버스 변환 로직(108)은, 입/출력 장치(28)가 입/출력 제어기로서 동작하여 시스템 버스(76)를 액세스하며, 이에 의해 시스템 메모리(24),(26)에 대한 판독 또는 기입하는데 필요한 수단을 제공한다. 이들 양 동작에 있어서, 입/출력 장치는 입/출력 버스를 제어한다. 입/출력 장치의 속도로 동작하는 비동기식 입/출력 버스 인터페이스(126)는, 버스 인터페이스 유닛(64)이 입/출력 버스(32)상의 입/출력 장치 제어기에 대해 슬레이브로서 동작하여 메모리 어드레스를 디코드함으로써 시스템 메모리(24),(26)에 대한 판독 또는 기입 사이클임을 판단하도록 한다. 이와 동시에, 시스템 버스 제어기 인터페이스(122)는 버스 인터페이스 유닛(64)이 시스템 버스(74)에 대한 제어기로서 동작하도록 한다. 메모리 제어기(58)(제1도 참조)는 버스 인터페이스 유닛(64)에 대한 슬레이브로 동작하여, 버스 인터페이스 유닛(64)에 시스템 메모리로부터 판독된 데이터를 제공하거나 시스템 메모리에 데이터를 기입한다. 시스템 메모리에 대한 판독과 기입은 FIFO 버퍼(124)를 통해 성취되며, 이 버퍼(124)의 블록도가 제3도에 도시된다.

제3도에 도시된 바와 같이, FIFO 버퍼(124)는, 시스템 버스(76)와 입/출력 버스(32)간의 데이터 정보를 일시적으로 저장하는 이중 포트 비동기식 양방향 저장 유닛이다. FIFO 버퍼(124)는 4개의 16바이트 버퍼(125A-125D) 및 FIFO 제어 회로(123)를 포함한다. 4개의 버퍼(125A-125D)는 입/출력 버스 제어기와 시스템 버스 슬레이브들에 대한 데이터를 버퍼링 함으로써 입/출력 버스(32)와 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 입/출력 버스 슬레이브 인터페이스(126)는 각각 하나의 32바이트 버퍼를 제어하는데, 나머지 하나의 32바이트 버퍼는 그들에게 있어 투명하게(transparent)동작한다. 이들 두 32바이트 버퍼는 기입 및 판독 동작에 이용된다.

각각의 FIFO(125A),(125B),(125C),(125D)는 어드레스 레지스터 섹션(address register section)을 가지며, 이 레지스터 섹션은 제각기의 FIFO 버퍼와 물리적으로 또는 논리적으로 연관된다. 입/출력 버스(32)로부터 FIFO 버퍼(125A)로 데이터가 전송될 때, 어드레스들이 연속하여 제공된 경우, 16바이트 버퍼는 16개의 데이터 바이트로 채워질 때까지 데이터가 누적될 것이다. 불연속 어드레스(non-contiguous address)가 어드레스 섹션에 의해 검출되면, FIFO 버퍼(125A)는 저장된 데이터를 FIFO 버퍼(125C)로 전송하고, 이와 동시에 FIFO 버퍼(125B)는 이 새로운 불연속 어드레스로부터의 데이터를 수신하기 시작한다. FIFO 버퍼(125B)는, FIFO 버퍼(125A)의 경우와 마찬가지로, 16개의 데이터 바이트로 채워지거나, 또 다른 불연속 어드레스가 검출될 때까지 계속해서 데이터를 수신한다. 상기한 조건 발생시, FIFO 버퍼(125B)는, 저장된 데이터를 FIFO 버퍼(125D)로 전송하며, FIFO(125A)는 데이터를 다시 저장하기 시작한다. 따라서, 16바이트의 불연속 어드레스 데이터 블록을 최고 4개까지 저장할 수 있다.

또한, 두개의 32바이트 버퍼를 병렬로 구성함으로써, 데이터의 판독 및 기입이 두 버퍼 사이에서 토글(toggle)될 수 있고, 따라서, 본질적으로 연속적인 판독 및 기입 기능을 제공할 수 있다.

또한, 32바이트 버퍼를, 다른 입/출력 버스(32) 또는 시스템 버스(76)에 결합된 두 개의 16바이트 버퍼 섹션으로 분할함으로써, 데이터를 저장 레지스터의 안 또는 밖으로 클럭킹(clocking)하는 신호의 용량성 로딩(capacitive loading)과 관련되는 FIFO의 성능에 대한 영향을 최소로 하면서, 저장 버퍼의 수를 증가시킬 수 있다. 이것은 2개의 버퍼가(병렬로) 부가될 경우 각 버스상의 클럭 신호의 로딩(loading)에 용량성 로딩의 1/2만이 부가되기 때문이다.

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

FIFO 제어 회로(123)는 FIFO 버퍼(124)의 동작을 제어하기 위한 로직을 제공한다.

개개의 입/출력 장치(28)는 1, 2 또는 4바이트(즉, 8, 16 또는 32비트)의 대역 폭으로 입/출력 버스를 통해 시스템 메모리(24) 또는 (26)에 대한 기입을 행할 수도 있다. 입/출력 장치(28)가 시스템 메모리에 대한 기입을 행하는 동안, 제1전송 기입 데이터는 처음에 FIFO 버퍼(125A) 또는 (125B)에 저장된다. 입/출력 버스 예상 어드레스 발생 회로(120)는 다음의 예상 어드레스 또는 연속 어드레스(contiguous address)를 계산한다. 이러한 다음의 연속 어드레스는 이후의 전송이 연속적인 것인지의 여부를 확인하기 위해 후속하는 입/출력 어드레스와 대비하여 체크된다. 연속하는 경우, 기입 데이터의 제2바이트 또는 바이트들이 동일한 FIFO 버퍼(125A) 또는 (125B)로 전송된다. FIFO 버퍼는 입/출력 버스(32)로부터 최고 40Mbit/s의 비동기 속도(asynchronous speed)로 데이터를 수신한다.

이러한 과정은 버퍼(125A) 또는 버퍼(125B)가 16바이트 정보 패킷으로 채워지거나 또는 불연속 어드레스가 검출될 때까지 계속된다. 버퍼(125A)가 채워지면, 다음의 클럭 사이클에서, 버퍼(125A)의 데이터는 버퍼(125C)로 전송된다. 마찬가지로, 버퍼(125B)가 채워지면, 그의 저장된 모든 내용은 한 클럭 사이클내에 버퍼(125D)로 전송된다. 그리고나서, 버퍼(125C) 및 (125D)에 저장된 데이터는, 시스템 버스 동작 속도의 1486 바이트 전송을 통해 시스템 메모리에 기입된다. 따라서, 입/출력 장치에 의한 시스템 메모리로의 기입동안, 버퍼(124)의 동작은 연속적이며, 버퍼(125A)와 (125B) 사이를 교번하여, 이들 버퍼의 각각은 인접한 버퍼(125C) 또는 (125D)로 데이터를 보내어 비워지고 다른 버퍼는 시스템 메모리에 기입될 데이터를 수신하게 된다. 이때, FIFO 버퍼(124)는, (i) 메모리에 기입될 다음 데이터 바이트의 어드레스를 예상하고, (ii) 시스템 버스(76)를 통해 FIFO 버퍼로부터 시스템 메모리로 기입할 데이터의 최대 속도를 수용함으로써 시스템 메모리의 데이터 기입 속도를 최적화 한다.

시스템 메모리로부터 입/출력 장치(28)로의 데이터 판독 동안, FIFO 버퍼(124)는 전송한 경우와 상이하게 동작한다. 시스템 버스 어드레스 발생 회로(118)는 초기의 판독 어드레스를 사용하여 후속하

는 판독 데이터의 어드레스를 발생하므로써 버퍼(125C) 또는 (125D)에 데이터를 누적시킨다. 시스템 버스는 16바이트 대역 폭의 전송을 지원하므로, 시스템 버스 제어기 인터페이스 회로(122)는, 입/출력 버스(32)가 실제 연속 어드레스를 제공하지 않아도 16바이트의 연속적인 데이터 패킷을 프리페치(prefetch)하여 그것을 버퍼(125C) 또는 (125D)에 저장하므로써, 전송시의 지연(latency)을 감소시킬 수 있다. 버퍼(125C)가 프리페치된 데이터로 채워지면, 버퍼(125C)는 그 데이터를 한 클럭 사이클내에 버퍼(125A)로 전송한다. 마찬가지로, 버퍼(125D)가 프리페치된 데이터로 채워지면, 버퍼(125D)는 그 데이터를 버퍼(125B)로 이동시킨다. 그리고나서, 버퍼(125A) 및 (125B)에 저장된 데이터는 개개의 입/출력 장치 제어기에 의해 1, 2 또는 4바이트의 대역 폭으로 판독될 수 있다. 이와 같이 시스템 버스 어드레스 발생 회로(118)는, 입/출력 장치 제어기에 의해 데이터의 프리페치를 중단하도록 지시받을 때까지 증분 계수기로서 작용한다.

버스간 페이싱 제어 로직(128)은 고속 입/출력 장치의 시스템 메모리에 대한 더욱 빠른 액세스를 제공한다. 버스간 페이싱 제어 로직(128)은, 입/출력 장치와 CPU 사이에서 메모리 제어기(58)에 대한 액세스를 교번시키지 않고, 복수 사이클(multiple cycles)을 요구하는 고속 장치에 의한 데이터 전송 동안, 입/출력 버스(32)를 통제하고 있는 입/출력 장치에게 시스템 메모리에 대한 방해받지 않는 액세스를 허용하므로써, 시스템(10)의 통상적인 메모리 제어기 중재 체계(normal memory controller arbitration scheme)를 무시한다. 따라서, 입/출력 장치에 의한 복수 사이클 전송 동안에는, CPU와 같은 로컬 장치가 메모리 버스의 제어를 요청한다고 하더라도, 버스간 페이싱 제어 로직(128)은 입/출력 장치에게 연속적인 메모리 버스 제어를 허용할 것이다.

프로그램된 입/출력 회로(116)는 버스 인터페이스 유닛(64)의 일부로서, 버스 인터페이스 유닛(64)내의 프로그램 가능한 모든 레지스터를 포함한다. 이들 레지스터는 특정 레지스터가 활성화인지 또는 비활성인지를 결정하도록 제각기 비트 연관된다. 이들 레지스터는, 무엇보다도, 버스 인터페이스 유닛(64)이 응답할 시스템 메모리 및 확장 메모리 어드레스 범위, 캐쉬가능(cacheable) 또는 캐쉬 불가능(non-cacheable) 확장 메모리 어드레스, 시스템 메모리 또는 캐쉬 어드레스 범위, 및 패리티 또는 에러 검사가 버스 인터페이스 유닛에 의해 지원되는지의 여부를 규정한다. 따라서, 프로그램된 입/출력 회로(116)는 버스 인터페이스 유닛(64)이 존재하는 환경과 구성 옵션(configuration options)을 식별한다. 프로그램된 입/출력 회로(116)내의 레지스터는 입/출력 버스(32)를 통하여 직접적으로 프로그램될 수는 없다. 따라서, 시스템(10)을 프로그래밍 하려면, 사용자는 시스템 버스를 통하여 프로그램된 입/출력 회로(116)와 CPU 수준에서 통신할 수 있는 입/출력 장치를 액세스 해야 한다.

메모리 어드레스 비교 로직(110)은 메모리 어드레스가 시스템 메모리에 대응하는지 또는 입/출력 버스(32)에 결합된 입/출력 장치(28)상의 확장 메모리에 대응하는지를 결정한다. 확장 메모리 및 시스템 메모리는 불연속 어드레스 블록일 수도 있으므로, 메모리 어드레스 비교 로직(110)은, 어느 경계가 어느 메모리에 대응하는가를 표시하기 위해 프로그램된 입/출력 회로(116)내 레지스터로 부터의 경계 정보가 로드된 복수의 비교기를 포함한다. 특정한 메모리 어드레스가 메모리 어드레스 비교 로직에 의해 경계 정보와 비교된후, 메모리 인터페이스 유닛은 그에 따라 응답하도록 준비된다. 예를 들면, 입/출력 버스(32)를 제어하는 입/출력 장치가 확장 메모리에 대해 판독 또는 기입을 행하고 있는 경우에는, 버스 인터페이스 유닛이 메모리 제어기(58)로 어드레스를 전송할 필요가 없으므로, 시간 및 메모리 대역 폭이 절약된다.

에러 복구 지원 로직(112)은, 데이터 패리티 에러(data parity error)가 검출될지라도 시스템(10)이 동작을 계속하도록 허용한다. 입/출력 장치(28)의 시스템 메모리(24),(26)에 대한 모든 기입 또는 판독 액세스시, 데이터의 패리티가 검사된다. 에러 복구 지원 로직(112)은 프로그램된 입/출력 회로(116)내의 레지스터와 상호작용하여 검출된 패리티 에러의 어드레스 및 시간을 포착한다. 그리고 나서, 이 레지스터의 내용은 적절한 시스템 소프트웨어에 의해 사용될 수도 있다. 예를 들면, CPU(38)가, 패리티 에러가 검출될 때마다, 어드레스를 레지스터에서 인출하도록 하는 고급 인터럽트에 대해 프로그램될 수도 있다. 이때, CPU는, 시스템 소프트웨어 인스트럭션에 기초하여, 시스템 동작을 계속할지 아니면 패리티 에러를 발생하는 것으로 식별된 동작을 단순히 종료할 것인지를 결정할 수 있다.

캐쉬 스누핑 로직(cache snooping logic)(114)은 버스 인터페이스 유닛(64)이 입/출력 버스(32)를 통해 발생하는 입/출력 장치에 의한 확장 메모리로의 모든 기입 동작에 대하여 입/출력 버스(32)를 감시하도록 한다. 먼저 스누핑 로직은 확장 메모리의 기입이 SRAM(40)내의 캐쉬가능한 확장 메모리에서 발생했는지를 결정한다. 캐쉬가능한 확장 메모리가 아니면 캐쉬된 데이터가 손상될 염려가 없다. 그렇지만, 캐쉬가능한 확장 메모리에서 기입 동작이 발생했음을 표시하는 양성 비교(positive compare)이면, 캐쉬 무효화(invalidation) 사이클이 시스템 버스(76)를 통해 개시된다. 따라서, CPU는 이같은 지시를 받아 SRAM(46)의 대응하는 어드레스를 무효화 한다. 캐쉬 스누핑 로직(114)은 양성 비교의 어드레스를 저장하는 수단을 제공하여, 입/출력 버스의 스누핑이 제1양성 비교 검출후 즉시 갱신되도록 하므로써, 입/출력 버스(32)의 연속적인 감시를 가능하게 한다.

시스템 메모리에 대한 입/출력 장치의 액세스를 동적으로 제어하므로써 시스템 메모리(24) 및 (26)안팎으로 데이터를 전송하기 위한 입/출력 버스(32)의 능력을 개선하기 위해 버스간 페이싱 제어 로직(128)이 사용된다. 시스템 성능은 특정의 사전 결정된 조건하에서 시스템 메모리에 대한 입/출력 버스(32)를 제어하는 입/출력 장치(28)의 액세스를 고정하므로써 향상된다.

FIFO 버퍼(124)와 함께 버스간 페이싱 제어 로직(128)은 입/출력 버스 제어기의 데이터 전송 속도를 시스템 메모리의 데이터 전송 속도에 최적으로 정합시키기 위해 사용된다. 예를 들면, 고속 입/출력 제어기가, 시스템 메모리가 기입 데이터를 수신할 수 있는 속도보다 더 빨리 시스템 메모리에 데이터를 기입할 수 있으면, 데이터가 시스템 메모리에 완전히 기입되기 전에, 버퍼링된 기입 데이터가 FIFO 버퍼(125A)(125B)를 채우게 된다. 또한, 동일한 고속 입/출력 제어기가, 시스템 메모리가 판독 데이터를 제공하는 속도보다 더 빨리 시스템 메모리로부터 데이터를 판독할 수 있으면, 사전페치된 데이터는 FIFO 버퍼(125C)(125D)에서 이용가능하지 않게 된다. 이들중 어느 경우에도, 결과는 지연

을 증가시키며, 따라서 입/출력 버스(32)에 의해 관측되었을 때 데이터 전송시의 성능은 저하된다.

대표적으로, 입/출력 장치 제어기는 판독 또는 기입 동작을 개시하며, 입/출력 버스(32)를 통해 버스 인터페이스 유닛(64)으로 메모리 어드레스를 제공한다. 버스 인터페이스 유닛의 메모리 어드레스 비교 로직 회로(110)는 이 어드레스를 프로그램된 입/출력 회로(116)내에 프로그램된 어드레스 범위에 비교하여, 이 동작이 확장 메모리를 향한 것인지 또는 시스템 메모리를 향한 것인지를 결정한다. 확장 메모리에 대한 동작으로 결정되면, 이러한 조건하에서는 시스템 메모리에 대한 액세스의 우선 순위를 결정할 필요가 없으므로, 버스간 페이싱 제어 로직(128)은 아무 일도 하지 않는다. 그러나, 시스템 메모리(24) 또는 (26)에 대한 동작이고 사전결정된 조건이 만족되면, 버스간 페이싱 제어 로직(128)은 입/출력 버스(32)를 제어하는 입/출력장치(28)에게 시스템 메모리에 대한 계속적인 액세스 우선권을 승인하는 신호를 송출한다. 이 신호는, 정상적인 중재 절차동안 메모리 제어기(58)에 의해 출력된 중재 승인 신호에 대한 무효화 신호로서 작용한다.

입/출력 장치에게 시스템 메모리에 대한 액세스 우선권이 승인되는 사전결정된 조건은 다음과 같다. 즉, (1) 입/출력 버스(32)를 제어하는 입/출력 장치에 의한 판독 요구(사전폐치)가 시스템 메모리내의 사전정의된 16바이트 어드레스 경계로 정렬되지 않는 경우, (2) 입/출력 버스(32)를 제어하는 입/출력 장치가 기입 사이클에 바로 후속하여 판독 사이클을 수행하는 경우, (3) 입/출력 버스(32)를 제어하는 입/출력 장치가 데이터 전송 사이클을 완료한 경우, (4) 입/출력 버스(32)를 제어하는 입/출력 장치가 스트리밍 모드(streaming mode)(초당 40메가바이트)로 데이터를 판독하거나 기입하는 경우, 또는 (5) FIFO 버퍼(124)내의 4개의 16바이트 버퍼중에서 3개 이상이 시스템 메모리에 기입될 데이터로 이미 채워지거나 또는 시스템 메모리로부터 판독될 사전폐치된 데이터로 채워진 버퍼가 4개의 16바이트 버퍼중 아직 2개 미만인 경우이다.

본 발명의 바람직한 실시예에서, 버스간 페이싱 제어 로직(128)은 3개 모드하에서 동작한다. 제1모드는 디폴트 모드(default mode)로서, 앞서 나열된 처음 3개의 조건에 대하여 버스 제어 페이싱을 결정한다. 제2모드(선택사양 1)는 앞서 나열된 4번째 조건에 대하여 속도를 결정하며, 제3모드(선택사양 2)는 앞서 나열된 5번째 조건에 대하여 속도를 결정한다. 시스템이 자동으로 제공하는 디폴트 속도 모드에 부가하여 어느 속도 모드(선택사양 1 또는 선택사양 2)로도 시스템 사용을 선택할 수 있다. 선택은, 프로그램된 입/출력 회로(116)내의 3비트 레지스터(미도시됨)의 내용을 프로그램함으로써 이루어진다. 디폴트 속도 모드는 레지스터 내용을 0-0-0으로 세트함으로써 정의되며, 선택사양 1은 레지스터 내용을 0-0-1로 세트함으로써 정의되며, 선택사양 2는 레지스터 내용을 1-0-0로 세트함으로써 정의된다.

앞서 설명된 바와 같이, 디폴트 페이싱 로직은 처음 3개의 조건에 대하여 속도를 제어한다. 제1조건하에서, 입/출력 장치(28)는 특정의 16바이트 패킷 경계로 정렬되지 않는 시스템 메모리(24) 또는 (26)의 판독을 요구하며, 이것은 판독 요구가 16바이트 패킷 경계내의 어딘가에서 시작하는 데이터 어드레스에 대한 것임을 의미한다. 판독 데이터가 특정의 16바이트 패킷 경계내의 어딘가에서 시작하는 데이터 어드레스에 대한 것임을 의미한다. 판독 데이터가 특정의 16바이트 패킷 경계로 정렬된 경우, 시스템 버스 어드레스 발생 회로(118)는 단순히 버퍼(125C) 또는 (125D)내로 데이터의 표준 16바이트 버스트 전송을 행하게 된다. 그러나, 판독 데이터가 패킷 정렬되지 않은 경우, 데이터는 복수의 1,2,3,4바이트 사이클을 이용하여 사전폐치되어야 한다. 이 데이터가 가장 효율적인 방법으로 사전 폐치되었음을 보장하기 위해, 입/출력 장치(28)에 의한 시스템 메모리로의 액세스는, 충분한 데이터가 자동으로 사전폐치되어 16바이트 경계에 도달하는 동안 고정된다. 그리고 나서, 이러한 16바이트 미만의 판독 데이터는 입/출력 버스(32)의 타이밍에 동기되며, 입/출력 장치 제어기는 데이터를 판독하기 시작한다. 이와 같이 입/출력 장치 제어기가 데이터를 판독하는 것과 동시에, 시스템 버스 어드레스 발생 회로(118)는, 시스템 메모리에 대한 고정 신호(lock signal)를 해제하기전에 그 다음 연속적인 16바이트 패킷을 사전폐치하게 된다. 이러한 16바이트 패킷은 사전정의된 경계에서 시작하므로, 버스트 전송이 가능하다. 따라서, 두 개의 데이터 전송이 중단없이 수행된다. 이러한 조건에서 제4도에 도시된 바와 같은 고정 신호(140)의 사용은, 입/출력 버스(32)상에서 장치가 데이터를 요구한 때와 버스 인터페이스 유닛(64)이 데이터를 프리라이드(pre-ride)한 때 사이의 최소의 지연과, 페이지 형성된 시스템 메모리의 가장 효율적인 사용을 보장한다.

제2조건하에서, 입/출력 장치(28)가 기입 전송으로부터 판독요구로 변경할 때, 잉여의 버퍼링된 기입 데이터가 FIFO 버퍼(125A-125D)내에 존재할 수도 있다. 일단 입/출력 장치 버스 제어기가 기입으로부터 판독으로 변경되면, 페이싱 제어 로직(128)은 사이클의 끝(end)이 발생하였으며 FIFO 버퍼(124)내에 저장되어야 할 더 이상의 데이터가 존재하지 않는 것으로 판단한다. 연속적인 기입 데이터가 더 이상 입력되지 않는 것으로 인식되었으므로, 버퍼(125A-125D)내에 이미 버퍼링된 데이터는 시스템 메모리에 기입되고 시스템 메모리로부터 일회성의 기본 동작(one atomic operation)으로 판독요구가 사전폐치된다.

제3조건하에서, 입/출력 장치 제어기가 입/출력 버스(32)의 제어를 포기할 때, 잉여의 버퍼링된 데이터가 FIFO 버퍼(125A-125D)내에 또한 존재할 수 있다. 이 경우에, 이러한 잉여의 데이터는, 시스템 버스(76)를 해제하기 전에 시스템 메모리에 기입되어야 한다. FIFO 버퍼(125A-125D)중 2개 이상이 이러한 잉여 데이터를 포함하는 경우에는, 입/출력 장치 제어기가 입/출력 버스(32)의 제어를 포기하자마자, 페이싱제어 로직(128)은 메모리 제어기로, FIFO 버퍼(124)내의 데이터가 하나의 전송 길이를 초과하여 시스템 메모리(24) 또는 (26)에 기입되어야 할 필요가 있음을 표시하는 신호를 발생하게 된다. 이러한 조건에서는, 고정 신호(140)를 사용하여, 입/출력 제어기가 입/출력 버스(32)의 제어를 해제하는 때와 버스 인터페이스 유닛(64)이 시스템 버스(76)의 제어를 해제하는 때 사이의 지연량이 최소가 되도록 보장한다.

처음 3개의 조건에 응답하는 디폴트 로직은 버스 인터페이스 유닛(64)의 하드웨어로 구축된 알고리즘에 의해 실현된다. 대표적으로, 이러한 요구된 로직을 구현하기 위해 종래에 잘 알려진 바와 같이 상태 머신이 사용될 수 있다. 제4 및 제5조건에 응답하는 페이싱 로직도 또한 버스 인터페이스 유닛의 하드웨어내에 구축된 알고리즘에 의해 실현된다. 제4도 및 제5도는 제각기 선택사양 1 및 선택사

양 2에 대한 버스간 페이싱 제어 로직(128)의 특정 실시예를 구현하기 위해 사용된 회로도를 예시한다.

제4도를 참조하여 설명하면, 선택사양 1(앞서 설명된 제4조건에 관련됨)을 구현하기 위한 버스간 페이싱 제어 로직은 클럭 입력을 갖는 S-R 래치(132), 앤드 게이트(134) 및 (136), 및 오어 게이트(138)를 포함한다. 래치(132)가 세트(즉, S입력이 액티브)되면, 래치는 고정 신호(140)를 출력한다. 이 고정 신호(140)는, 입/출력 장치(28)가 스트리밍 모드로 시스템 메모리(24) 및 (26)에 데이터를 판독 또는 기입하고 있을 때, 입/출력 버스(932)를 제어하는 입/출력 장치(28)에게 시스템 메모리에 대한 연속적인 액세스를 허용한다. 여기서 스트리밍이라 함은 동일한 어드레스로의 데이터 전송을 의미하며, 이것은 상이한 어드레스로의 데이터 전송보다 더욱 빨리 성취될 수 있다.

제4도의 로직에 의한 스트리밍 데이터 기입 동작의 검출 및 FIFO 버퍼(125C) 또는 (125D)중 어느 하나가 16바이트 패킷의 버퍼링된 기입 데이터로 채워졌음이 검출되면, 고정 신호가 활성화되고, 시스템 메모리에 대한 버스트 기입 시퀀스(burst write sequence)가 시스템 버스(76)를 통해 기시되며, 고정 신호는, 입/출력 장치(28)가 스트리밍 시퀀스를 종료하거나 FIFO 버퍼내에 남아 있는 데이터가 존재하지 않을 때까지 액티브가 유지된다. 마찬가지로, 스트리밍 데이터 판독 동작의 검출시, 고정 신호가 활성화되고, 시스템 메모리에 대한 버스트 판독 시퀀스가 개시되며, 고정 신호는 FIFO 버퍼내에 공간이 존재하는 한 및 입/출력 장치(28)가 스트리밍 시퀀스를 종료할 때까지 액티브로 유지된다.

스트리밍 판독 또는 기입에 응답하는 버스 제어 로직(128)은 3비트 레지스터내의 내용을 0-0-1로 프로그래밍함으로써 정의된다. 따라서, 3비트 레지스터가 0-0-1로 프로그램되면, 앤드 게이트(134)는 레지스터의 내용을 디코드하여, 선택사양 1 인에이블 라인(142)를 하이(high)로 제공한다. 라인(144)은, 버스 인터페이스 유닛(64)이, 입/출력 버스(32)를 제어하는 특정 입/출력 장치(28)가 스트리밍 모드로 데이터를 기입 또는 판독하고 있는 것을 검출하는 한, 하이로 유지된다. 라인(142) 및 (144)은 이 시간동안 하이이므로, 앤드 게이트(136)가 래치(132)를 세트하여 오어 게이트(138)의 출력(146)이 하이인 때는 언제나 고정 신호(140)를 출력하게 도니다.

오어 게이트(138)의 출력(146)은, (i) 입/출력 버스(32)를 제어하는 입/출력 장치(28)가 스트리밍 데이터 판독 요구를 송출하거나 또는 (ii) FIFO 버퍼(124)내의 16바이트 버퍼(125C) 또는 (125D)중 어느 하나가 채워진 때(기입동작)에 하이이다. 이들의 경우 및 앞서 설명한 바와 같이 라인(142) 및 (144)가 하이인때에, 래치(132)가 메모리 제어기에 고정 신호(140)를 제공함으로써 메모리 제어기(58)의 중재 체계를 무효화시키게 된다. 입/출력 장치(28)가 스트리밍 모드로 더 이상 판독 또는 기입 데이터를 전송하지 않음을 표시할 때마다, 라인(148)이 하이가 되고, 이에 의해 래치(132)를 리세트함으로써 제4도에 도시된 로직을 효과적으로 디스에이블시킨다.

제5도를 참조하여 설명하면, 선택사양 2(앞서 설명된 제5조건에 관련됨)를 구현하기 위한 버스간 제어 로직은 클럭 입력을 갖는 S-R 래치(150) 및 (152)를 포함하며, 이들 각각은 판독 및 기입 사이클에 대한 분리된 로직 경로에 의해 독립적으로 제어된다. 이들 클럭 입력을 갖는 래치(150), (152)중 어느 하나가 세트되면, 고정 신호(140)가 인에이블되어 메모리 제어기(58)로 출력된다. 이들 래치는 R-입력을 활성화시키므로써 리셋되며, 이에 의해 래치는 고정 신호를 출력하는 것이 금지된다.

앞서 나열된 제5조건하에서는, (i) 데이터가 입/출력 제어기(28)에 의해, 시스템 버스(76)를 통해 시스템 메모리에 기입될 수 있는 속도보다 더 빨리 입/출력 버스(32)를 통해 버스 인터페이스 유닛(64)에 기입되거나, 또는 (ii) 데이터가 입/출력 제어기(28)에 의해, 버스 인터페이스 유닛이 시스템 메모리로부터 데이터를 사전 폐지할 수 있는 속도보다 더 빨리 버스 인터페이스 유닛(64)으로부터 판독된다. 데이터가 입/출력 제어기에 의해 너무 빨리 기입되면, FIFO 버퍼(125A-125D)내에 잉여의 데이터가 쌓이기 시작한다. 1/2을 초과하는 FIFO 버퍼내의 공간이 이러한 잉여 데이터를 포함하게 되면, 래치(152)에 의해 고정 신호(140)가 송출되어 시스템 메모리의 기본 전송(atomic transfer)을 허용한다. 데이터가 입/출력 제어기(28)에 의해 입/출력 버스(32)를 통해 버스 인터페이스 유닛(64)으로부터 너무 빨리 판독되는 경우, FIFO 버퍼(125A) 또는 (125B)가 비게 되며, 이것은 버퍼(125C) 및 (125D)에 이용가능한 사전폐지된 연속적인 데이터가 고갈 되었음을 의미한다. 따라서, 고정 신호(140)가 래치(150)에 의해 출력되어 버스 인터페이스 유닛이 버퍼(125C) 및 (125D)내로 데이터를 사전폐지할 수 있게 된다. 이와 같이 하여, FIFO 버퍼(124)의 적어도 받은 사전폐지된 데이터로 채워진 채 유지된다.

제5조건에 응답하는 버스 제어 로직은 3비트 레지스터의 내용을 1-0-0로 프로그래밍함으로써 정의된다. 그러므로, 3비트 레지스터가 1-0-0로 프로그램되면, 노어 게이트(154)가 레지스터의 내용을 디코드하여 선택사양 2 인에이블 라인(156)을 하이로 제공하게 된다. 선택사양 2 인에이블 라인(156)은, 데이터 판독 동작(제5도에 도시된 로직의 상부) 및 데이터 기입 동작(제5도에 도시된 로직의 하부)에 응답하여 제어 로직을 인에이블시키는데 사용된다.

기입 동작동안, 오어 게이트(160)의 출력(158)은 또한 버퍼(125A-125D)중 적어도 두 개가 채워진 동안 하이가 된다. 대표적으로, 이것은, 버퍼(125A) 또는 (125B)가 채워졌으며, 데이터는 버퍼(125C) 또는 (125D)로 전송되었고, 다른 버퍼(125A) 또는 (125B)가 지금 채워지고 있음을 의미한다. 이 경우, 데이터는 버퍼(125C) 또는 (125D)내에 또는 그 양쪽에 존재하며, 즉각적으로 시스템 메모리에 기입될 수 있다. 라인(156) 및 (158)이 하이인 경우, 앤드 게이트(162)는 메모리 제어기(58)로 고정 신호(140)를 출력하게 될 래치(152)를 세트시킨다. 고정 신호(140)는, 래치(152)의 R 입력이 앤드 게이트(164)의 출력에 의해 하이로 구동될 때까지 액티브로 유지된다. 앤드 게이트(164) 출력은, 버퍼(125C) 및 (125D)가 모두 비어서 이에 의해 이들 버퍼로부터 시스템 메모리로 데이터를 즉각 기입할 필요가 없음을 표시하는 경우 하이가 된다.

판독 동작 동안, 래치(150)는 앤드 게이트(166)의 하이 출력으로 세트된 때에 메모리 제어기(58)로 고정 신호를 출력하게 된다. 앤드 게이트(166)의 출력은, (i) 선택사양 2가 액티브되어 라인(156)이 하이인 때, (ii) 버스 인터페이스 유닛(64)이 시스템 메모리로부터의 판독을 검출하여 라인



(168)이 하이가 된 때 및 (iii) 버퍼(125A) 또는 (125B)가 비어 있는 경우, 버퍼(125C) 또는 (125D)의 내용이 비어 있을 수도 있고, 버퍼(125C) 또는 (125D)내에 시스템 메모리로 부터의 사전폐치된 연속적인 데이터를 위한 공간이 남아 있게 되므로, 이 경우에 고정 신호가 송출된다.

판독 동작시에 고정 신호는, 래치(150)의 R-입력이 오어 게이트(174)에 의해 활성화될 때까지 액티브로 유지된다. 오어 게이트(174)는, (i) 버스 인터페이스 유닛(64)이 시스템으로 부터의 판독 동작이 존재하지 않음을 검출한 때 또는 (ii) 버퍼(125C) 또는 (125D)중 어느 하나가 채워지고, 버퍼(125A) 및 (125B)중 어느 하나도 비어 있지 않은 때에 하이로 구동된다. 이들 두 경우중 어느 경우에도, 고정 신호는 불필요하게 되고, 이에 따라 래치(150)가 리셋된다. 그리고 나서, 시스템 메모리(24) 및 (26)에 대한 액세스는 메모리 제어기(58)에 의해 실행되는 통상의 중재 방법하에 제어된다.

이상은, 이중 버스 구조를 갖는 컴퓨터용 버스 인터페이스 유닛내의 비동기 양방향성 저장장치에 대한 바람직한 실시예가 설명되었다. 그러나, 이상의 설명은 단지 실시예로서 이루어진 것임을 이해해야 하며, 본 발명은 본 명세서에 개시된 특정 실시예에 국한되지 않으며, 특허청구 범위에 기재된 본 발명의 진정한 정신으로부터 벗어나지 않고 다양한 재구성, 변경 및 대체가 가능하다.

## (57) 청구의 범위

### 청구항 1

시스템 메모리(system memory)와 ; 상기 시스템 메모리에 버스(a memory bus)에 의해 접속되어, 상기 시스템 메모리에 대한 액세스(access)를 제어하기 위한 메모리 제어기(a memory controller)와 ; 로컬 버스(a local bus)를 통해 상기 메모리 제어기에 결합되며, 판독 및 기입 동작(read and write operations)을 수행함으로써 상기 로컬 버스 및 상기 메모리 버스를 통해 상기 시스템 메모리로 부터 데이터를 판독하거나 상기 시스템 메모리에 데이터를 기입하도록 접속되는 중앙 처리 장치(a central processing unit ; CPU)와 ; 시스템 버스(a system bus)에 의해 상기 메모리 제어기와 접속된 버스 인터페이스 유닛(a bus interface unit)과 ; 입/출력 버스(an I/O bus)에 의해 상기 버스 인터페이스 유닛에 접속되어, 상기 버스 인터페이스 유닛을 통해, 상기 시스템 메모리로부터 데이터를 판독하고 상기 시스템 메모리에 데이터를 기입하는 적어도 하나의 입/출력 장치(at least one input/output device)를 포함하며, 상기 버스 인터페이스 유닛은, 상기 적어도 하나의 입/출력 장치에 의해 수행되는 판독 및 기입 동작 동안 상기 시스템 버스와 상기 입/출력 버스 사이에 전송되는 데이터의 일시적인 저장을 제공하기 위해 양방향성 데이터 저장 유닛(a bidirectional data storage unit)을 포함하며, 상기 양방향성 저장 유닛은, (i) 상기 입/출력 버스와 상기 시스템 버스 사이에 직렬 접속된 제1버퍼 쌍(a first pair of buffers)과, (ii) 상기 제1버퍼 쌍과는 병렬로 상기 입/출력 버스와 상기 시스템 버스 사이에 직렬접속된 제2버퍼 쌍(a second pair of buffers)을 구비하여, 제1데이터 세트(first set of data)는 상기 한 쌍의 직렬 접속된 버퍼 쌍 중한 버퍼 쌍의 버퍼 사이에서 전송되게 하는 동시에, 제2데이터 세트(second set of data)는 상기 한 쌍의 직렬 접속된 버퍼 쌍중 다른 한 버퍼 쌍으로 전송되도록 하고, 상기 각 버퍼 쌍내의 각 버퍼는 상기 시스템 버스와 상기 입/출력 버스 사이의 양방향성 데이터 전송을 위해 사용되며, 상기 시스템 버스 또는 상기 입/출력 버스중 어느 하나의 버스상에서 시작하여 상기 시스템 버스 또는 상기 입/출력 버스중 다른 한 버스로의 방향으로 전송되도록 되어 있는 상기 제1 및 제2데이터 세트(first and second sets of data)는 상기 양방향성 저장 유닛으로 전송되며, 이때 (a) 상기 제1데이터 세트는 상기 직렬접속된 버퍼 쌍중 한 버퍼쌍의 한 버퍼에 저장되고, (b) (1) 상기 직렬 접속된 버퍼 쌍중 한 버퍼 쌍의 상기 한 버퍼에 저장되어 있는 상기 제1데이터 세트가 상기 직렬접속된 버퍼 쌍중 한 버퍼 쌍의 다른 버퍼로 전송되는 것과 동시에, (b) (2) 상기 제2데이터 세트가 상기 직렬접속된 버퍼 쌍중 다른 버퍼 쌍의 한 버퍼에 저장되는 컴퓨터 시스템(a computer system).

### 청구항 2

제1항에 있어서, 상기 적어도 하나의 입/출력 장치는 복수의 입/출력 장치(a plurality of I/O devices)를 포함하며, 데이터는 상기 입/출력 버스를 상기 복수의 입/출력 장치 각각과 상기 양방향성 저장 유닛 사이에서 상이한 대역폭으로 전송되는 컴퓨터 시스템.

### 청구항 3

제1항에 있어서, 상기 각 버퍼 쌍(each pair of buffers)은 32바이트 데이터를 저장하는 컴퓨터 시스템.

### 청구항 4

제3항에 있어서, 상기 32바이트의 버퍼 쌍의 각각은 직렬 접속된 두 개의 16바이트 버퍼를 포함하는 컴퓨터 시스템.

### 청구항 5

제1항에 있어서, 상기 버퍼는, 상기 입/출력 장치가 동작하는 속도로 상기 입/출력 버스를 통해 그리고 소정의 주어진 데이터 버스트 속도(a given data burst speed)로 상기 시스템 버스를 통해, 판독 및 기입하도록 구성되는 컴퓨터 시스템.

### 청구항 6

컴퓨터 시스템에서 입/출력 버스에 접속된 적어도 하나의 입/출력 장치와, 상기 입/출력 버스와 시스템 버스 사이에 직렬접속된 제1버퍼 쌍 및 상기 제1버퍼 쌍과 병렬로 상기 입/출력 버스와 상기 시스템 버스 사이에 직렬접속된 제2버퍼 쌍을 구비하는 버스 인터페이스 유닛에 의해 상기 입/출력 버스에 결합되는 시스템 버스에 접속된 시스템 메모리 사이의 데이터 전송을 제어하는 방법에 있어서, 상기 시스템 버스 또는 상기 입/출력 버스중 어느 하나의 버스상에서 시작하여 상기 시스템 버

스 또는 상기 입/출력 버스중 다른 한 버스로의 방향으로 전송되도록 되어 있는 제1 및 제2데이터 세트를 상기 버스 인터페이스 유닛을 가로질러 전송하되, 이러한 전송은 (a) 상기 한 쌍의 직렬접속된 버퍼 쌍중 한 버퍼 쌍(one of pairs of series-connected buffers)의 하나의 버퍼내에 상기 제1 데이터 세트를 일시적으로 저장하는 단계와, (b) (1) 상기 한 쌍의 직렬접속된 버퍼 쌍중 한 버퍼 쌍(one of pairs of series-connected buffers)의 상기 하나의 버퍼로 전송하는 단계와, 이와 동시에 (b) (2) 상기 한 쌍의 직렬접속된 버퍼 쌍중 다른 버퍼 쌍(the other series-connected pair of buffers)의 하나의 버퍼 내에 상기 제2데이터 세트를 전송하는 단계로 이루어지는 데이터 전송을 제어하는 방법.

#### 청구항 7

제6항에 있어서, 상기 적어도 하나의 입/출력 장치는 복수의 입/출력 장치를 포함하며, 데이터는 상기 각 입/출력 장치와 상기 버스 인터페이스 유닛 사이에 상기 입/출력 버스를 통해 상이한 대역폭으로 전송되는 데이터 전송을 제어하는 방법.

#### 청구항 8

제6항에 있어서, 상기 버퍼는, 상기 선택된 입/출력 장치가 동작하는 속도로 상기 입/출력 버스를 통해 그리고 소정의 주어진 데이터 버스트 속도로 상기 시스템 버스를 통해 판독 및 기입하도록 구성되는 데이터 전송을 제어하는 방법.

#### 청구항 9

제6항에 있어서, 상기 시스템 버스는 상기 버스 인터페이스 유닛과 상기 시스템 버스 사이에 판독 또는 기입 데이터를 최고 16바이트의 대역폭으로 전송하며, 상기 입/출력 버스는 상기 입/출력 장치와 상기 버스 인터페이스 유닛 사이에 판독 또는 기입 데이터를 최고 4바이트의 대역폭으로 전송하는 데이터 전송을 제어하는 방법.

#### 청구항 10

제6항에 있어서, 데이터는 상기 버스 인터페이스 유닛내의 상기 버퍼를 통해 선입 선출(a first-in first-out)방식으로 전송되는 데이터 전송을 제어하는 방법.

#### 청구항 11

제6항에 있어서, 상기 버스 인터페이스 유닛내의 각 버퍼 쌍의 상기 각 버퍼는 상기 시스템 버스와 상기 입/출력 버스 사이의 양방향성 데이터 전송에 사용되는 데이터 전송을 제어하는 방법.

#### 청구항 12

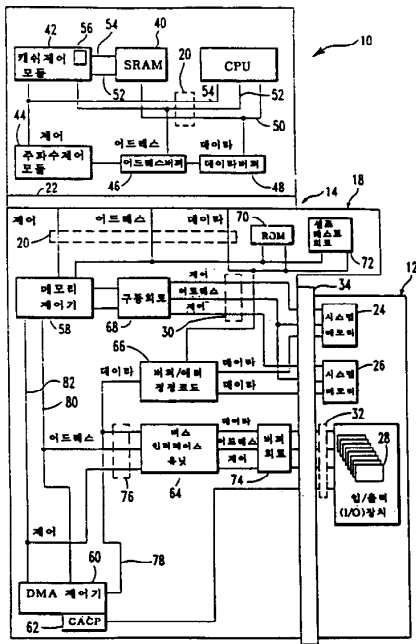
제11항에 있어서, 상기 제1버퍼 쌍중 한 버퍼(one of said first pair of buffers) 및 상기 제2버퍼 쌍중 한 버퍼(one of said second pair of buffers)는 상기 시스템 버스에 접속되며, 상기 시스템 버스에 접속되는 각 버퍼는 상기 시스템 버스를 통해 데이터를 동시에 전송할 수 있는 컴퓨터 시스템.

#### 청구항 13

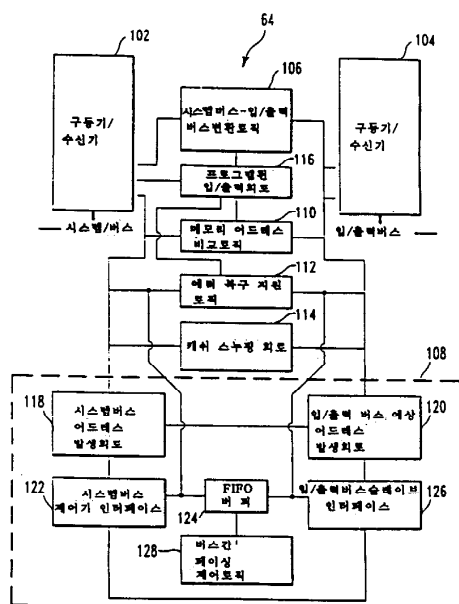
제6항에 있어서, (c) 상기 한 쌍의 직렬접속된 버퍼 쌍중 상기 다른 버퍼 쌍의 하나의 버퍼로부터 상기 한 쌍의 직렬접속된 버퍼 쌍중 상기 다른 버퍼 쌍의 다른 하나의 버퍼로 상기 제2데이터 세트를 전송하는 단계와 ; (b) (1) 상기 시스템 버스를 통해, 상기 한 쌍의 직렬접속된 버퍼 쌍중 상기 하나의 버퍼 쌍의 상기 다른 하나의 버퍼로부터 상기 제1데이터 세트를 전송하는 단계와 ; 이와 동시에 (d) (2) 상기 시스템 버스를 통해, 상기 한 쌍의 직렬접속된 버퍼 쌍중 다른 하나의 버퍼 쌍의 상기 다른 하나의 버퍼로부터 상기 제2데이터 세트를 전송하는 단계를 더 포함하는 데이터 전송을 제어하는 방법.

도면

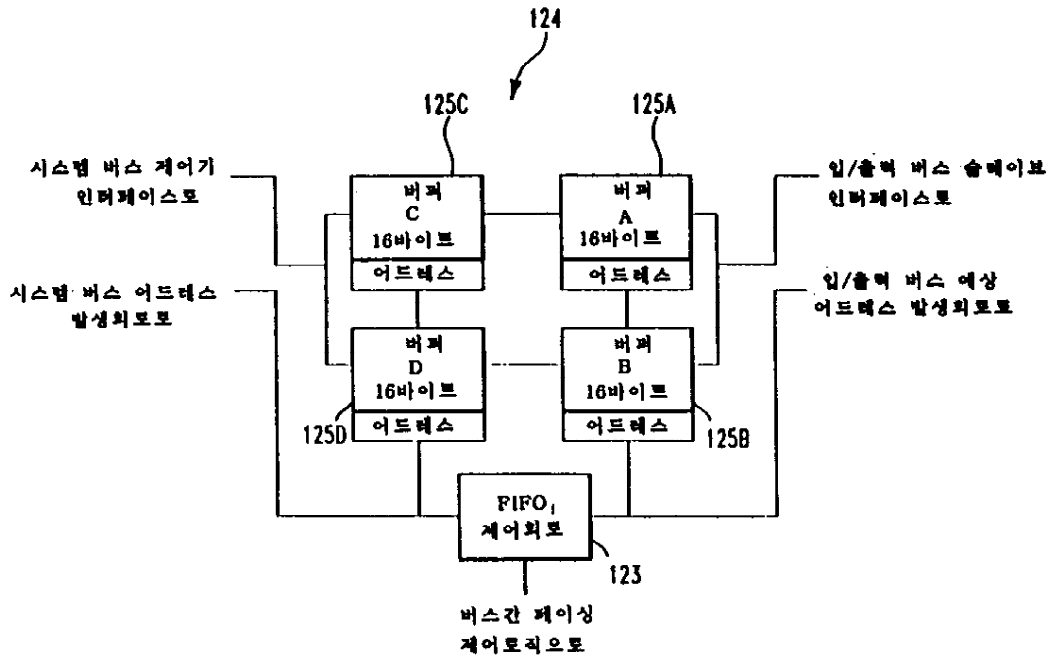
도면1



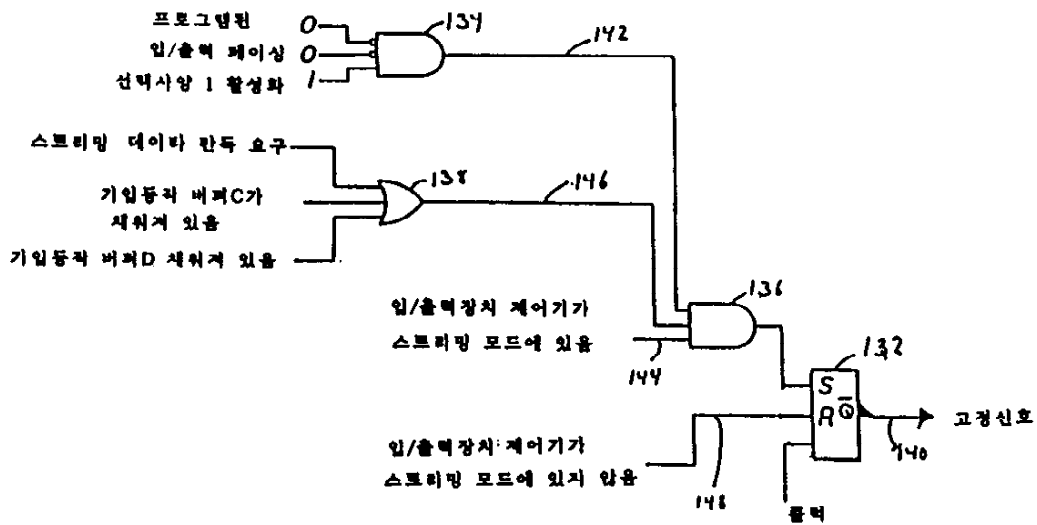
도면2



도면3



도면4



## 도면5

