(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.[6]
H03K 19/00

(45) 공고일자 2003년07월04일
(11) 등록번호 10-0381076
(24) 등록일자 2003년04월08일

(21) 출원번호 10-1995-0024323
(22) 출원일자 1995년08월02일

(65) 공개번호 특1996-0009411
(43) 공개일자 1996년03월22일

(30) 우선권주장 284,945 1994년08월02일 미국(US)

(73) 특허권자 모토로라 인코포레이티드
미국, 일리노이 60196, 샤움버그, 이스트 앨공퀸 로드 1303

(72) 발명자 크리스토퍼브라이언트
미국,텍사스78749,오스틴,체노코티나트레일8004

브라이언레이놀드스
미국,텍사스78681,라운드록,밥캣드라이브8710

(74) 대리인 이병호

심사관 : 권호영

(54) 인터버스버퍼

요약

인터버스 버퍼(18)는 두개의 서로 다른 크기의 버스들 사이의 데이터 트랜스퍼들(data transfers)을 조정한다. 제 1 버스(프로세서 버스(processor bus))는 빅인디언 프로토콜(a big endian protocol) 또는 '개조된' 리틀 인디언 모드(a 'munged' little endian mode)에 따라 데이터가 배열되는 것을 허용한다. 제 2 버스(로칼 버스(local bus))는 빅 인디언 프로토콜 또는 참 리틀 인디언 모드(true little endian mode)에 따라 데이터가 배열되는 것을 허용하지만 트랜잭션 크기(transaction size)를 정의하지는 않는다. 개시된 인터버스 버퍼는 다양한 서로 다른 트랜잭션 크기들 및 동작 모드들에도 불구하고 인터버스 데이터 트랜스퍼들을 조정한다.

대표도

도 1

색인어

프로세서 버스, 로칼 버스, 빅 인디언 프로토콜, 인터버스 버퍼

명세서

도면의 간단한 설명

제 1 도는 본 발명에 따라 구성된 데이터 처리 시스템의 블록도.

제 2 도, 제 3 도, 제 4 도, 제 5 도 및 제 6 도는 빅 인디언, 참 리틀 인디언 및 의사 리틀 인디언 모드들에서 동작하는 디바이스들 중에서 다양한 크기들의 데이터 트랜스퍼들의 예들을 표 형태로 나타낸 도면.
제 7 도는 빅 인디언 및 의사 리틀 인디언 모드들에서 동작하는 디바이스들 사이에서 데이터 트랜스퍼들에 이용되는 맵핑 방식을 표 형태로 나타낸 도면.
제 8 도는 제 1 도에 도시된 버스-투-버스('인터버스 버퍼')의 블록도.
제 9 도 및 10 도는 제 8 도에 도시된 인터버스 버퍼의 일부분의 동작을 나타내는 2개의 상태-전이 다이어그램들을 나타내는 도면.
도면의 주요부분에 대한 부호의 설명
12 : 데이터 프로세서 14 : 메인 메모리
16 : 하드 디스크 드라이브, 그래픽 드라이버, 확장 슬롯
18 : 버스-대-버스 브릿지 26 : 로칼 어드레스 래치
28 : 로칼 데이터 로우 래치 30 : 로칼 데이터 하이 래치

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

발명의 분야
본 발명은 디지털 컴퓨팅 시스템들(digital computing systems)에 관한 것으로, 특히, 버스-투-버스 버퍼 회로들(bus-to-bus buffer circuits)에 관한 것이다.
발명의 배경
모든 디지털 컴퓨팅 디바이스는, 멀티-유닛 트랜스퍼(multi-unit transfer)에서 개별적인 데이터 유닛들을 어떻게 배열하는가에 따라서 두 개의 상호 배타적인 클래스(classes)들중 하나로 분류될 수 있다. 이들 두 클래스들의 멤버들은 '빅 인디언(big endian)' 또는 '리틀 인디언(little endian)' 디바이스로서 공지되어 있다. 현재의 디지털 컴퓨팅 디바이스에서, 8개의 데이터 비트들, 즉 한 바이트는 통상적으로 가장 작은 개별적으로 어드레싱가능한 데이터 유닛이다. 그러나, 대부분의 디지털 컴퓨팅 디바이스들은 트랜스퍼의 단일 바이트 어드레스 및 어떤 크기의 신호에 의해 식별되는 하나 내지 여덟개의 바이트들을 한번에 트랜스퍼링한다. 그러므로, 데이터 트랜스퍼와 단일 어드레스에 대해 각각의 바이트들을 디바이스가 어떻게 배열하는지가 중요하다.
빅 인디언 디바이스들은 멀티-유닛 트랜스퍼(multi-unit transfer)의 최상위 바이트(가장 좌측의 바이트)에 최하위 어드레스를 할당하고, 트랜스퍼의 그 다음 최상위 바이트에 다음의 순차적 어드레스를 할당한다. 이러한 방법은, 멀티-유닛 트랜스퍼의 빅 엔드(big end)(이진수로서 고려됨)가 저장부의 맨처음에 오기 때문에, 빅 인디언으로 불리운다. IBM의 RISC 시스템/6000, IBM의 시스템 370 및 모토 로라의 68000 계열의 마이크로 프로세서들은 빅 인디언 디바이스들의 예들이다.
리틀 인디언 디바이스들은 멀티-유닛 트랜스퍼의 최하위 바이트(가장 우측의 바이트)에 가장 낮은 어드레스를 할당하고, 트랜스퍼의 다음의 최하위 바이트에 다음의 순차적 어드레스를 할당한다. 이러한 방법은 멀티-유닛 트랜스퍼의 리틀 엔드(little end)(이진수로 고려됨)가 저장부의 맨처음에 오기 때문에 리틀 인디언이라 칭한다. DEC의 VAX 아키텍처 및 Intel의 X86 계열의 마이크로프로세서들은 리틀 인디언 디바이스들의 예들이다.

발명이 이루고자 하는 기술적 과제

동일한 디지털 컴퓨팅 시스템에서 빅 인디언 디바이스와 리틀 인디언 디바이스를 결합하는 것이 바람직할 때가 있다. 빅 인디언 디바이스와 리틀 인디언 디바이스 사이의 멀티-유닛 데이터 트랜스퍼들은 재배열되어야 한다. 그렇지 않으면, 한 디바이스에 의한 데이터 출력이 다른 디바이스에서 이해되지 못할 수가 있다. 멀티-유닛 트랜스퍼들에서 데이터를 재배열하는데 이용되는 적어도 두가지 방법들이 있는데, 그것은 '프로그래밍된 바이트-교환(programmed byte-swapping)'과 '어드레스 개조(address munging)'이다.
프로그래밍된 바이트-교환은 두개의 서로 다른 디바이스들을 접속하는 데이터 경로에 삽입된 어떤 수의 멀티플렉서들을 내장하는 솔루션(solution)이다. 이들 멀티플렉서들은 데이터 트랜스퍼 크기(data transfer size)와 데이터 경로 크기(data path size)에 의존하여 바이트 재배열(byte reordering)을 수행한다. 예를 들어, 양쪽 디바이스들이 4바이트 양(four byte quanta)까지 데이터를 트랜스퍼 링 하면(양쪽 디바이스가 32-비트 데이터 버스들을 가지면), 각각의 바이트는, 4개의 인입하는 바이트들중 임의의 하나를 4개의 바이트 레인들중 임의의 하나에 라우팅하기 위해, 장치에 접속된 4:1 멀티플렉서를 가져야 한다.
어드레스 개조는 빅 인디언 디바이스들로부터 리틀 인디언 디바이스들로 트랜스퍼링되는 데이터를 재배열하는 솔루션이다. 이 솔루션은 두 단계를 결합한다. 첫째로, 리틀 인디언 디바이스로 향하는 각 데이터 트랜스퍼의 어드레스는 특정 수학적 함수에 따라 제 2 어드레스를 형성하도록 수정 또는 '개조'된다. 데이터 트랜스퍼는 상기 제 2 어드레스

에 의해 인덱싱된다. 둘째로, 데이터 트랜스퍼의 바이트들은 데이터 경로에 대해 역으로 된다(reversed).
본 발명의 특징들 및 이점들이 첨부된 도면들과 연결되어 취해진 다음의 상세한 설명으로부터 더 명확하게 이해될 것이며, 도면에서 같은 또는 대응하는 부분들은 같은 부호로 기재한다.

발명의 구성 및 작용

제 1 도는 본 발명에 따라 구성된 데이터 프로세싱 시스템(10)의 블록도이다. 데이터 프로세싱 시스템(10)은 빅 인디언(이후 'BE'로 칭함) 또는 개조된(munged) 빅 인디언(이후, 의사 리틀 인디언(pseudo little endian) 또는 'PLE'로 칭함) 배열 방식에 따라 멀티-바이트 데이터 트랜스퍼들을 발생시키는 데이터 프로세서(12)를 포함한다. 데이터 프로세서(12)는 메인 메모리 시스템(14)과 통신하며, 버스-투-버스 브릿지 회로(또는 '인터버스 버퍼')(18)를 통해 다양한 주변 디바이스들(16)와 통신한다. 개시된 본 발명에 따라 인터버스 버퍼(18)는 빅 인 디언 버스와 리틀 인디언(이후 'LE'로 칭함) 버스를 인터페이스하는데 필요한 바이트 재배열(byte reordering) 기능들을 제공한다.
제 1 도와 관련하여, 데이터 프로세서(12)는 제 1 입력/ 출력 프로토콜에 따르며 반면에 주변 디바이스들(16)은 제 2 입력/출력 프로토콜에 따른다. 제 1 도에 도시된 바와 같이, 데이터 프로세서(12)가 따르는 프로토콜은 '프로세서 버스'라 라벨되고 불리운다. 주변 디바이스들(16)이 따르는 프로토콜은 '로칼 버스'라 라벨되고 불리운다. 데이터 프로세싱(10)은 본 발명과 무관한 다수의 이유들에 의해 둘 또는 그 이상의 서로 다른 버스들에 따라 동작하는 디바이스들을 포함할 수도 있으며: 로칼 버스는 하드디스크 저장 시스템들, 그래픽스 드라이버들, 확장 슬롯들 등등 같은 어떤 타입들의 디바이스들에 대한 산업-표준일 수 있고, 두 버스들은 서로 다른 시점에서 개발되었을 수 있으며, 데이터 프로세싱 시스템(10)은 두개의 서로 다른 제조자들로부터의 기술을 내장할 수 있다.
도시된 실시예에서, 프로세서 버스는 Power PC(인터내셔널 비즈니스 머신즈 코포레이션(IBM)의 상표) 60X 버스 프로토콜의 64 비트 구성이다. 각 60X 버스 트랜잭션은 T-BURST 제어 신호를 각각 선언(asserting) 또는 선언해제(de-asserting) 함으로써 하나 또는 4개의 순차적인 64 비트 트랜잭션을 지정한다. 단일 64 비트 트랜잭션의 경우에, 3-비트 제어 신호인 T-SIZE는 트랜잭션의 1에서 8 바이트를 검증(validate)한다.
도시된 실시예에서, 로칼 버스는 주변 구성요소 상호접속(Peripheral Component Interconnect)('PCI') 프로토콜의 32 비트 수행이다. 제어 신호 즉 FRAME의 선언해제(de-assertion)는 단일 트랜잭션에서 최종 32 비트 트랜스퍼를 표시한다. 4개의 바이트-인에이블 제어 신호들은 4 바이트 트랜스퍼의 각각의 바이트를 개별적으로 검증한다. 하기에서 분명해지는 바와 같이, 버스 크기들의 차이 및 PCI 트랜잭션에서의 선험적 트랜잭션 크기 정보(a priori transaction size information)의 부족은, 공지된 바이트 재배열 솔루션들 즉, 프로그램 가능한 바이트 교환의 이용을 배제한다.
제 2 도 내지 제 6 도는 BE, 참 리틀 인디언(true little endian)(이후, 'LE') 및 PLE 모드들에서 동작하는 디바이스들 중에서 다양한 크기들의 데이터 트랜스퍼들의 예들을 표 형태로 도시한다. 먼저, 이들 도면들은 BE, LE 및 PLE 디바이스들의 동작들을 예시하는데 이용될 것이다. 이후에, 이들 동일한 도면들은 인터버스 버퍼(18)의 동작을 예시하는데 이용될 것이다.
제 7 도는 BE 및 PLE 모드들에서 동작하는 디바이스들 사이에서 데이터 트랜스퍼들에 이용하기 위한 맵핑 계획을 표 형태로 도시한 도면이다. 서술된 테이블은 한번에 8 바이트까지 트랜스퍼링할 수 있는 디바이스들의 동작을 예시한다. 도시된 바와 같이, BE 트랜스퍼의 시작 어드레스의 3 개의 최하위 비트들은, 트랜잭션 크기에 의존하는 어떤 비트 패턴과 배타적 OR 처리된다(exclusively-OR'd)('XOR'd'). 이러한 작용은 '개조된' 어드레스 또는 PLE 시작 어드레스를 발생시킨다. 데이터 프로세서(12)와 같은 디바이스는 PLE 트랜잭션 어드레스를 출력하기 전에 내부적으로 이러한 기능을 수행한다. 그 다음에, PLE 어드레스는 LE 또는 PLE 디바이스로부터 데이터를 기입 또는 판독하는데 이용될 수 있다. 트랜잭션이 LE 디바이스로부터 의 판독 또는 기입 동작이면, 그때 8 바이트는 트랜스퍼링될 때 데이터 경로에 대하여 개별적으로 반전된다(reversed). 트랜잭션이 다른 PLE 디바이스로부터의 판독 또는 기입 동작이면, 그때 어떤 바이트 반전도 필요하지 않다.
다시 제 2 도와 관련하여, BE 디바이스는 시작 어드레스 XX000 의해 인덱싱된 단일 8 바이트 트랜스퍼와, '정렬된(aligned)' 이중-워드 트랜스퍼를 발생시키며, 여기서, 'X'는 돈트 케어(don't care) 어드레스 비트이다. 트랜스퍼는 판독 또는 기입일 수 있다. 상기 예와 모든 다음의 예들에서, 최상위 바이트는 'A'로 표시되며, 다음의 최상위 바이트는 'B'로 표시된다. 제 7 도에 도시된 바와 같이, 어드레스의 3개의 최하위 비트들은 '000'과 XOR 처리된다. 대응하는 PLE 또는 '개조된' 어드레스는 또한 XX000이다. 그러므로, 8 바이트들(A 내지 H)은 PLE 모드('BE/PLE')에서 동작하는 BE 디바이스로부터 동일한 시작 어드레스 XX000에, 동일한 순서(즉 A 에서 H)로 출력된다.
이들 8 바이트가 LE 디바이스로 직접 트랜스퍼링되거나 또는 PLE 디바이스를 통해 LE 디바이스로 트랜스퍼링되면, 그때 8 바이트(A 내지 H)는 데이터 경로에 대해 반전된다. 특히, 첫번째의 PLE-바이트-레인(이 경우에 A)을 점유하는 바이트는 여덟번째의 LE-바이트-레인(어드레스 XX111)으로 재맵핑(re-mapped)되며, 두번째의 PLE-바이트-레인(이 경우에 B)을 점유하는 바이트는 일곱번째의 LE-바이트-레인(어드레스 XX110)으로 재맵핑되며, 세번째의 PLE-바이트-레인(이 경우에 C)을 점유하는 바이트는 여섯번째의 LE-바이트-레인(어드레스 XX101)으로 재맵핑되며, 네번째의 PLE-바이트-레인(이 경우에 D)을 점유하는 바이트는 다섯번째의 LE-바이트-레인(어드레스 XX100)으로 재맵핑되고, 다섯번째의 PLE-바이트-레인(이 경우에 E)을 점유하는 바이트는 네번째의 LE-바이트-레인(어드레스 XX011)으로 재맵핑되며, 여섯번째의 PLE-바이트-레인(이 경우에 F)을 점유하는 바이트는 세번째의 LE-바이트-레인(어드레스 XX010)으로 재맵핑되고, 일곱번째의 PLE-바이트-레인(이 경우에 G)을 점유하는 바이트는 두번째의 LE-바이트-레인(어드레스 XX001)로 재맵핑되고, 여덟번째의 PLE-바이트-레인(이 경우에 H)을 점유하는 바이트는 첫번째의 LE-바이트-레인(어드레스 XX000)으로 재맵핑된다.

다시 제 3 도를 참조하면, BE 디바이스는 2 개의 '정렬된' 워드 트랜스퍼들인 시작 어드레스들 XX000 및 XX100에 의해 인덱싱된 2 개의 4-바이트 트랜스퍼들을 발생시킨다. 이들 트랜스퍼들은 판독들 또는 기입들일 수 있다. 예를들면, 제 1 트랜스퍼는 시작 어드레스 XX000와 연관된 4개의 바이트들 A 내지 D를 메인 메모리(14)에 기입할 수 있다. 제 7 도에 도시된 바와 같이 어드레스의 3개의 최하위 비트들은 '100'과 XOR 처리된다. 대응하는 PLE 또는 '개조된' 어드레스는 XX100이다. 그러므로, 4개의 바이트들 A 내지 D는 어드레스 위치들 XX100 내지 XX111에 기입된다. 이들 4 개의 바이트들이 메인 메모리(14)로부터 LE 디바이스로 트랜스퍼링되거나 또는 직집 BE/PLE 디바이스로부터 LE 디바이스로 트랜스퍼링되면, 그때 전체 데이터 경로는 반전된다(단지 4 개의 검증된 바이트들(valid bytes)만이 인에이블링된다).

기입예와 관련하여 계속 설명하면, 제 2 의 4-바이트 트랜스퍼는 시작 어드레스 XX100와 연관된 4 개의 바이트들(E 내지 H)을 메인 메모리(14)에 기입할 것이 다. 제 7 도에 도시된 바와 같이, 어드레스의 3 개의 최하위 비트들은 '100'과 XOR처리된다. 대응하는 PLE 또는 '개조된' 어드레스는 XX000이다. 그러므로, 4개의 바이트들(E 내지 H)은 어드레스 위치들 XX000 내지 XX011에 기입된다. 이들 4개의 바이트들이 메인 메모리(14)에서 LE 디바이스로 트랜스퍼링되거나 또는 직접 BE/PLE 디바이스에서 LE 디바이스로 트랜스퍼링되면, 그때 전체 데이터 경로는 반전된다(단지 4 개의 검증된 바이트들만이 인에이블링된다).

제 4 도와 관련하여, BE 디바이스는 시작 어드레스들 XX000 및 XX101에 의해 인덱싱되는 2 개의 3-바이트 트랜스퍼들을 발생시킨다. 이들 트랜스퍼들은 판독들 또는 기입들일 수 있다. 예를들면, 제 1 의 트랜스퍼는 시작 어드레스 XX000와 연관된 3개의 바이트들 A 내지 C를 메인 메모리(14)에 기입할 수 있다. 제 7 도에 도시된 바와 같이, 어드레스의 3개의 최하위 비트들은 '101'과 XOR 처리된다. 대응하는 PLE 또는 '개조된' 어드레스는 XX101이다. 그러므로, 3개의 바이트들 A 내지 C는 어드레스 위치들 XX101 내지 XX111에 기입된다. 이들 3개의 바이트들이 메인 메모리(14)에서 LE 디바이스로 트랜스퍼링되거나 또는 직접 BE/PLE 디바이스에서 LE 디바이스로 트랜스퍼링되면, 그때 전체 데이터 경로는 반전된다(단지 3 개의 검증된 바이트들만이 인에이블링된다).

기입예와 관련하여 계속 설명하면, 제 2 트랜스퍼는 또한 시작 어드레스 XX101에 연관된 3개의 바이트들 E 내지 G를 메인 메모리(14)에 기입할 것이다. 제 7 도에 도시된 바와 같이, 어드레스의 3 개의 최하위 비트들은 '101'과 XOR된다. 대응하는 PLE 또는 '개조된' 어드레스는 XX000 이다. 그러므로, 3개의 바이트들 E 내지 G는 어드레스 위치들 XX000 내지 XX010에 기입된다. 이들 3개의 바이트들이 메인 메모리(14)에서 LE 디바이스로 트랜스퍼링되거나 또는 직접 BE/PLE 디바이스에서 LE 디바이스로 트랜스퍼링되면, 그때 전체 데이터 경로는 반전된다(단지 3개의 검증된 바이트들만이 인에이블링된다).

제 5 도에 있어서, BE 디바이스는, 4개의 '정렬된' 하프-워드 트랜스퍼(half-word transfer)들, 즉, 시작 어드레스들 XX000, XX010, XX100 및 XX110에 의해 인덱싱된 4개의 2-바이트 트랜스퍼들을 발생시킨다. 이들 트랜스퍼들은 판독들 또는 기입들일 수 있다. 예를들면, 제 1 트랜스퍼는 시작 어드레스 XX000에 연관된 2개의 바이트들 A 및 B를 메인 메모리(14)에 기입할 수 있다. 제 7 도에 도시된 바와 같이, 어드레스의 3개의 최하위 비트들은 '110'과 'XOR'된다. 대응하는 PLE 또는 '개조된' 어드레스는 XX110이다. 그러므로, 2개의 바이트들 A 및 B 는 어드레스 위치들 XX110 및 XX111에 기입된다. 이들 2개의 바이트들이 메인 메모리(14)에서 LE 디바이스로 트랜스퍼링되거나 또는 직접 BE/PLE 디바이스에서 LE 디바이스로 트랜스퍼링되면, 그때, 전체 데이터 경로는 반전된다(단지 2 개의 검증된 바이트들만이 인에이블링된다).

기입예와 관련하여 계속 설명하면, 제 2 의 2-바이트 트랜스퍼는 시작 어드레스 XX010에 연관된 바이트들 C 및 D를 메인 메모리(14)에 기입할 것이다. 제 7 도에 도시된 바와 같이, 어드레스의 3개의 최하위 비트들은 '110'과 'XOR'된다. 대응하는 PLE 또는 '개조된' 어드레스는 XX010이다. 그러므로, 2개의 바이트들 C 및 D는 어드레스 위치들 XX100 및 XX101에 기입된다. 이들 2개의 바이트들이 메인 메 모리(14)에서 LE 디바이스로 트랜스퍼링되거나 또는 직접 BE/PLE 디바이스에서 LE 디바이스로 트랜스퍼링되면, 그때 전체 데이터 경로는 반전된다(단지 2 개의 검증된 바이트들만이 인에이블링된다).

기입예와 관련하여 계속 설명하면, 제 3 의 2-바이트 트랜스퍼는 시작 어드레스 XX100과 연관된 바이트들 E 및 F를 메인 메모리(14)에 기입할 것이다. 제 7 도에 도시된 바와 같이, 어드레스의 3개의 최하위 비트들은 '100'과 XOR된다. 대응하는 PLE 또는 '개조된' 어드레스는 XX010 이다. 그러므로, 2개의 바이트들 E 및 F는 어드레스 위치들 XX010 및 XX011에 기입된다. 이들 2개의 바이트들이 메인 메모리(14)에서 LE 디바이스로 트랜스퍼링되거나 또는 직접 BE/PLE 디바이스에서 LE 디바이스로 트랜스퍼링된다면, 그때 전체 데이터 경로는 반전된다(단지 2 개의 검증된 바이트들만이 인에이블링된다).

기입예와 관련하여 계속 설명하면, 제 4 의 2-바이트 트랜스퍼는 시작 어드레스 XX100에 연관된 바이트들 G 및 H를 메인 메모리(14)에 기입할 것이다. 제 7 도에 도시된 바와 같이, 어드레스의 3개의 최하위 비트들은 '110'과 XOR된다. 대응하는 PLE 또는 '개조된' 어드레스는 XX000이다. 그러므로 2개의 바이트들 G 및 H는 어드레스 위치들 XX000 및 XX001에 기입된다. 이들 2개의 바이트들이 메인 메모리(14)에서 LE 디바이스로 트랜스퍼링되거나 또는 직접 BE/PLE 디바이스에서 LE 디바이스로 트랜스퍼링되면, 그때 전체 데이터 경로는 반전된다(2 개의 검증된 바이트들만이 인에이블링된다).

제 6 도에 있어서, BE 디바이스는 8개의 시작 어드레스들 XX000 내지 XX111 에 의해 인덱싱되는 8개의 단일-바이트 트랜스퍼들을 발생시킨다. 이들 트랜스퍼들은 판독들 또는 기입들일 수 있다. 예를들면, 8개의 단일-바이트 트랜스퍼들은 8개의 바이트들 A 내지 H를 개별적으로 메인 메모리(14)에 기입할 수 있다. 제 7 도에 도시된 바와 같이, 각 어드레스의 3개의 최하위 비트들은 '111'과 XOR된다. 8개의 시작 어드레스들 XX000 내지 XX111는 8개의 PLE 어드레스들 XX111 내지 XX000을 발생시키도록 개조된다. 그러므로, 8개의 바이트들 H 내지 A는 어드레스 위치들 XX000 및 XX111에 각각 기입된다. 이들 바이트들이 메인 메모리(14)에서 LE 디바이스로 트랜스퍼링되거나 또는 BE/P

LE 디바이스에서 LE 디바이스로 직접 트랜스퍼링되면, 그때 전체 데이터 경로는 반전된다(단지 검증된 바이트들만이 인에이블링된다). BE 바이트 배열 및 LE 바이트 배열은 이 경우에 동일하다는 것이 유의되어야 한다. 일반적으로, 데이터 트랜스퍼 크기가 가장 작은 데이터 유닛과 동일한 크기일 때, BE, LE 및 PLE 모드들사이에 차이가 없다.
제 8 도는 제 1 도에 도시된 인터버스 버퍼(18)의 블록도이다. 인터버스 버퍼(14)는 좌측부(20), 우측부(22) 및 공통 제어 유닛(24)을 가진다.
좌측부(20) 및 제어 유닛(24)은 로칼 버스에서 프로세서 버스로의 데이터 및 어드레스들의 흐름(flow)을 제어한다. 데이터 프로세서(12)가 로칼 버스에 연결된 디바이스로부터 데이터를 판독할 때 또는 로칼 버스에 연결된 디바이스가 메인 메모리(14)에 데이터를 기입할 때, 데이터는 로칼 버스에서 프로세서 버스로 흐른다. 로칼 버스에 연결된 디바이스가 데이터를 판독 또는 메인 메모리(14)에 기입할 때 어드레스들은 로칼 버스에서 프로세서 버스로 흐른다. 데이터 프로세서(12)는 로드 명령의 실행에 응답하여 로칼 버스에 연결된 디바이스로부터 데이터를 판독한다. 로칼 버스에 접속된 디바이스는 디바이스의 다양한 동작들에 응답하여 데이터를 판독 또는 메인 메모리(14)에 기입한다.
우측부(22) 및 제어 유닛(24)은 프로세서 버스에서 로칼 버스로의 데이터 및 어드레스들의 흐름을 제어한다. 데이터 프로세서(12)가 로칼 버스에 연결된 디바이스에 데이터를 기입할 때 또는 로칼 버스에 연결된 디바이스가 메인 메모리(14)로 부터 데이터를 판독할 때, 데이터는 프로세서 버스에서 로칼 버스로 흐른다. 데이터 프로세서(12)가 데이터를 판독하거나 또는 로칼 버스에 연결된 디바이스에 데이터를 기입할 때, 어드레스들은 프로세서 버스로부터 로칼 버스로 흐른다. 데이터 프로세서(12)는 저장 명령의 실행에 응답하여 그리고 데이터 코히런시 프로토콜(data coherency protocol) 즉, '스누핑이후의 데이터 캐스트-아웃(data cast-out after snooping)'에 응답하여 로칼 버스에 연결된 디바이스에 데이터를 기입한다. 로칼 버스에 연결된 디바이스는 디바이스의 다양한 동작들에 응답하여 메인 메모리(14)로부터 데이터를 판독한다.
제 8 도에 도시된 좌측부(20)와 관련하여 설명을 계속하면, 로칼 버스로 부터의 데이터 및 어드레스 정보는 아랫부분에서 들어가서 윗쪽으로 프로세서 버스로 흐른다. 특히, 32 데이터 비트는 좌측부(20)로 들어가고 서로에 대해 바이트 교환되며; 제 1 바이트는 제 4 바이트 레인으로 재라우팅되고(rerouted), 제 2 바이트는 제 3 바이트 레인으로 재라우팅되고, 제 3 바이트는 제 2 바이트 레인으로 재라우팅되고, 제 4 바이트는 제 1 바이트 레인으로 재라우팅된다. 도시된 실시예는 멀 티-층 금속 상호접속, 즉, 금속 산화물 반도체('MOS') 프로세스를 이용하여 제조된다. 서술된 바이트 반전(byte reversal)은 금속 대 금속 비어들(metal to metal vias)의 유리한 배치에 의해 달성될 수 있다.
바이트 교환된 후, 32 데이터 비트는 로칼 어드레스 래치(26), 로칼 데이터 로우 래치(28) 및 로칼 데이터 하이 래치(30)에 입력된다. 제어 신호 LATCH_ADDRESS는 로칼 어드레스 래치(26)가 입력 데이터 비트들을 샘플링하고 유지하게 한다. 그 다음에 입력 데이터 비트들은 8개의 데이터 바이트들을 선택하도록 프로세서 어드레스 버스로 라우팅된다. 제어 신호 LOCAL_LOW_SELECT는 로칼 데이터 로우 래치(28)가 입력 데이터 비트들을 샘플링하고 유지하게 한다. 제어 신호 LOCAL_HIGH_SELECT는 로칼 데이터 하이 래치(30)가 입력 데이터 비트들을 샘플링하고 유지하게 한다. 제어 유닛(24)은 제어 신호들 LATCH_ADDRESS, LOCAL_LOW_SELECT 및 LOCAL_HIGH_SELECT를 발생시킨다. 제어 유닛(24)와 동작은 제 9 도 및 10 도와 연결되어 하기에 서술된다.
로칼 데이터 로우 래치(28)에 의해 출력된 4개의 바이트들은 다시 바이트 교환되고 32비트 2:1 멀티플렉서('MUX')(32)의 한 세트의 입력들에 접속된다. 멀티플렉서(32)는 또한 바이트 교환되지 않은 로칼 데이터 로우 래치(28)에 의해 출력된 32비트들을 수신한다. 로칼 데이터 하이 래치(30)에 의해 출력된 4개의 바이트들은 다시 바이트 교환되고 32 비트 2:1 멀티플렉서('MUX')(34)의 한 세트의 입력들에 접속된다. 멀티플렉서(34)는 또한 바이트 교환되지 않은 로칼 데이터 하이 래치(30)에 의해 출력된 32 비트들을 수신한다. 제어 신호 LE_MODE는 멀티플렉서들(32 및 34)의 출력들을 선택한다. 특히, 제어 신호 LE_MODE는 데이터를 소싱(sourcing)하는 로칼 디바이스가 LE 디바이스일 때(그리고 프로세서 버스 디바이스들이 PLE 디바이스들일 때) 선언된다(asserted). 제어 신호 LE_MODE는 데이터를 소싱하는 로칼 디바이스가 BE 디바이스일 때(그리고 프로세서 버스 디바이스들이 BE 디바이스들일때) 선언해제된다(de-asserted). 선언될 때, 제어 신호 LE_MODE는 멀티플렉서들(32, 34)이 로칼 데이터 로우 래치(28)와 로칼 데이터 하이 래치(30)의 교환되지 않은 출력들을 출력하게 한다(바이트들은 입력될 때 한번만 교환되며, 바이트 재배열을 발생시킨다). 선언해제될 때, 제어 신호 LE_MODE는 멀티플렉서들(32, 34)이 로칼 데이터 로우 래치(28)와 로칼 데이터 하이 래치(30)의 바이트 교환된 출력들을 출력하게 한다(바이트들은 두번 교환되며, 어떤 바이트 재배열도 발생시키지 않는다). 멀티플렉서(32)에 의해 출력된 32 비트들은 64 비트 데이터 신호를 형성하도록 멀티플렉서(34)에 의해 출력된 32 비트들과 연결된다. 이 데이터 신호는 제 1도에 도시된 프로세서 데이터 버스에 접속된다.
제 8 도에 도시된 우측부(22)에 대한 설명을 계속해서, 프로세서 버스로부터의 데이터는 윗부분으로 들어가서 아래 방향으로 로칼 버스로 흐른다. 상세히 설명하면, 트랜잭션 어드레스 ADDRESS(27, 28)의 제 4 및 제 5 의 최하위 비트들에 의존하여 64 데이터 비트들이 우측부(22)로 들어가서 4개의 이중 워드 래치들(36, 38, 40 및 42)중 하나에 의해 래칭된다(데이터 프로세서(12)는 정렬된 이중 쿼드-워드 데이터 유닛들(aligned double quad-word data units)(256 비트들)에서 버스 트랜잭션들을 종종 실행한다. 이 방법은 어드레스 번역 거칠기(address translation granularity)와 버스 대역폭의 제한조건들(constraints)을 균형을 이루게 한다). 64 비트 4:1 멀티플렉서('MUX')는 이중 워드 래치들(36, 38, 40 및 42)의 출력들 각각을 수신한다. 트랜잭션과 연관된 어드레스, ADDRESS(27, 28)의 제 4 및 제 5 의 최하위 비트들은 멀티플렉서(44)의 하나의 특별한 입력을 선택한다. 멀티플렉서(44)의 64 비트 출력은 더 중요한 절반(more significant half)(비트들 0:31)과 덜 중요한 절반(less significant half)(비트들 32:63)으로 분리된다. 각 절반은 로칼 버스와 동일한 크기인 32 비트이다.
멀티플렉서(44)에 의해 출력된 바이트들의 더 중요한 절반은 바이트 교환되고 32 비트 2:1 멀티플렉서('MUX')(46)의 한 세트와 입력들에 접속된다. 멀티플렉서(46)는 또한 바이트 교환되지 않은 멀티플렉서(44)에 의해 출력된 바이트들의 더 중요한 절반을 수신한다. 멀티플렉서(44)에 의해 출력된 바이트들의 덜 중요한 절반은 바이트 교환되고 32 비

트 2:1 멀티플렉서('MUX')(48)의 한 세트의 입력들에 접속된다. 멀티플렉서(48)는 또한 바이트 교환되지 않은 멀티플렉서(44)에 의해 출력된 바이트들의 덜 중요한 절반을 수신한다. 제어 신호 LE_MDDE 는 멀티플렉서들(46, 48)의 출력들을 선택한다. 선언될 때, 제어 신호 LE_MODE 는 멀티플렉서들(46, 48)이 멀티플렉서(44)에 의해 출력된 교환되지 않은 더 중요한 및 덜 중요한 절반들을 출력하게 한다. 선언해제될 때, 제어 신호 LE_MODE 는 멀티플렉서들(46, 48)이 멀티플렉서(44)에 의해 출력된 바이트 교환된 더 중요한 및 덜 중요한 절반들을 출력하게 한다. 멀티플렉서들(46, 48)의 출력들은 32 비트 3:1 멀티플렉서('MUX')(50)의 두 세트의 입력들에 접속된다. 멀티플렉서(50)는 또한 프로 세서 어드레스 버스상에 있는 어드레스 신호들을 수신한다. 이러한 접속은 3개의 입력들중 하나를 선택하는 한 쌍(pair)의 제어 신호들 PROCESSOR_SELECT 및 PROCESSOR_HIGH_SELECT을 위한 경로를 멀티플렉서(50)에 제공한다. 제어 유닛(24)은 제어 신호들 PROCESSOR_LOW_SELECT 및 PROCESSOR_HIGH_SELECT를 발생시킨다. 제어 유닛(24)의 동작은 제 9 도 및 10 도를 참조하여 후술된다. 멀티플렉서(50)의 출력은 래치(52)에 의해 래칭된다. 래치(52)의 출력은 로칼 버스에 접속되기 전에 바이트 교환된다.

제 9 도 및 제 10 도는 제 8 도에 도시된 인터버스 버퍼의 일부분의 동작을 예시하는 두개의 상태-전이도들이다. 제어 유닛(24)은 동작들: DEVSEL, DATA_COUNT, LE_MODE, FRAME, IRDY, TRDY, ADDRESS(29) , DATARDY, MASTER, TARGET 및 RESET을 결정하는 11개의 입력 제어 신호들을 수신한다.

제어 신호 DEVSEL는 목표 디바이스가 특별한 트랜잭션을 받아들였다는 것을 나타낸다(PCI 버스 프로토콜은 DEVSEL이 액티브 로우 신호(active low signal)임을 지정한다. 설명의 간략화를 위해, DEVSEL은 액티브 하이 제어 신호인 것처럼 서술될 것이다).

제어 신호 DATA_COUNT는 마스터로서 특별한 트랜잭션에서 트랜스퍼들의 수를 나타낸다. DATA_COUNT 가 제로이면, 트랜잭션은 8개의 데이터 트랜스퍼들을 포함한다. 인터버스 버스(18)는 DATA_COUNT를 발생시킨다. 프로세서 버스 초기화된 트랜잭션에서, DATA_COUNT 는 트랜스퍼링될 정렬된 32 비트 양의 수를 나타낸다. DATA_COUNT는 PowerPC 60x 제어 신호들 T-SIZE 및 T-BURST로부터 계산된다. 로칼 버스 초기화된 트랜잭션에서, DATA_COUNT는 '돈케어(don't care)'이다(PCI 버스 프로토콜은 그와같은 파라미터를 정의하지 않는다).

제어 신호 LE_MODE 는 데이터 프로세싱 시스템(10)이 LE 모드(선언됨) 또는 BE 모드(선언해제됨)에서 동작하는지를 나타낸다. 상술한 바와 같이, 데이터 프로세서(12)는 데이터 프로세싱 시스템(10)이 LE 모드에서 동작할 때 PLE 모드에서 동작한다. 서술된 실시예에서, 인터버스 버퍼(18)내의 이용자 액세스가능한 레지스터의 출력은 제어 신호 LE_MODE 를 발생시킨다.

제어 신호 FRAME는 PCI 버스 프로토콜 제어 신호이다. 그것은 PCI 버스 컴플라이언트(compliant) 디바이스에 의해 개시된 특정 트랜잭션에서의 데이터 트랜스퍼들의 수를 간접적으로 나타낸다. 그와같은 디바이스는 하나 또는 그 이상의 데이터 트랜스퍼들을 요구하는한 FRAME을 선언한다(PCI 버스 프로토콜은 FRAME이 액티브 로우 신호임을 나타낸다. 설명의 용이함을 위해 FRAME은 액티브 하이 제어 신호 인처럼 서술될 것이다).

제어 신호 IRDY('I-ready'로 칭함)는 PCI 버스 프로토콜 제어 신호이다. 선언될 때, 이것은 특정 트랜잭션을 개시한 PCI 디바이스가 트랜잭션 형태가 주어지면 데이터를 적절한 것으로서 수신 또는 송신할 준비가 되어있다는 것을 나타낸다(PCI 버스 프로토콜은 IRDY가 액티브 로우 신호임을 나타낸다. 설명의 용이함을 위해, IRDY는 액티브 하이 제어 신호인 것처럼 서술될 것이다).

제어 신호 TRDY('T-ready'로 칭함)은 PCI 버스 프로토콜 제어 신호이다. 선언될 때, 이것은 특정 트랜잭션의 목표 디바이스인 PCI 디바이스가 트랜잭션 형태 가 주어지면 적절한 것으로서 데이터를 수신 또는 송신할 준비가 되어 있다는 것을 나타낸다(PCI 버스 프로토콜은 TRDY는 액티브 로우 신호임을 설명한다. 설명의 용이함을 위해, TRDY는 액티브 하이 제어 신호인 것처럼 서술될 것이다).

제어 신호 ADDRESS(29)는 특별한 버스-대-버스 트랜스퍼의 32개의 어드레스 비트들중 30 번째 비트이다. 제어 유닛(24)은 후술되는 바와 같이 뒤따르는 클록 사이클에서 이용하기 위해 각 클럭 사이클마다 제어 신호 ADDRESS(29)를 래칭한다. 이 래칭된 제어 신호는 LATCH[ADDRESS(29)]로 표기된다.

제어 신호 DATARDY('데이터 준비'로 칭함)는 인터버스 버퍼(28)에 의해 발생된다. 선언될 때, 이것은 데이터 프로세서(12) 또는 메인 메모리(14)로부터의 데이터가 로칼 버스로의 트랜스퍼를 위해 이용가능하다는 것을 표시한다.

제어 신호들 MASETR 및 TARGET는 데이터 프로세서(12) 또는 로칼 버스에 접속된 디바이스가 특별한 트랜잭션의 마스터인지를 나타낸다. 선언될 때, MASTER는 데이터 프로세서(12)가 인터버스 버퍼(18)의 마스터임을 나타낸다. 선언될 때, TARGET는 로칼 버스에 접속된 디바이스가 인터버스 버퍼(18)의 마스터임을 나타낸다.

인터버스(18)상의 핀은 제어 신호 RESET를 발생시킨다. 선언될 때, RESET는 인터버스 버퍼(18)를 재초기화한다.

제 9 도로 돌아가면, 제어 신호들 LATCH_ADDRESS, LOCAL_LOW_SELECT 및 LOCAL_HIGH_SELECT의 동작을 예시하는 상태-전이 다이아그램이 도시되어 있다. 제어 신호들 LOCAL_LOW_SELECT 및 LOCAL_HIGH_SELECT는 로칼 버스로부터 프로세서 버 스로 흐르는 덜 중요한 데이터 워드 및 더 중요한 데이터 워드로 향하여 래칭한다. 제어 유닛(24)은 그 목적을 향해 세가지 상태들, 즉 LATCH ADDRESS, LATCH LOW WORD 및 LATCH HIGH WORD를 조작한다. 제어 유닛(24)은 상태 LATCH ADDRESS에 있다면 그리고 그 경우에 한해 제어 신호 LATCH_ADDRESS를 선언한다. 제어 유닛(24)은 상태 LATCH LOW WORD에 있다면 그리고 그 경우에 한해 제어 신호 LOCAL_LOW_SELECT를 선언한다. 제어 유닛(24)은 상태 LATCH HIGH WORD에 있다면 그리고 그 경우에 한해 제어 신호 LOCAL_HIGH_SELECT를 선언한다.

제어 유닛(24)은 제어 신호 RESET가 선언된 후, LATCH ADDRESS상태로 초기화된다. 제어 유닛(24)은 함수 A가 논리적으로 참일 때 LATCH LOW WORD로 전이된다. 함수 A 는 후술된다. 역으로, 제어 유닛(24)은 함수 A가 논리적으로 거짓이면 LATCH ADDRESS에서 LATCH HIGH WORD 상태로 전이된다. 제어 유닛(24)은 함수 B가 논리적으로 참일 때, LATCH LOW WORD 상태에서 LATCH HIGH WORD 상태로 전이되거나 또는 그 역으로 전이된다. 함

수 B는 후술된다. 논리 유닛(24)은 함수 C가 논리적으로 참이면, LATCH LOW WORD 상태에서 LATCH ADDRESS 상태로 전이되거나, 또는 LATCH HIGH WORD 상태에서 LATCH ADDRESS 상태로 전이된다. 함수 C는 후술된다. 제어 유닛(24)은 함수 D가 논리적으로 참이면, LATCH LOW WORD 상태 또는 LATCH HIGH WORD 상태에서 유지된다.

제 10 도로 돌아가면, 제어 신호들 PROCESSOR_LOW_SELECT 및 PROCESSOR_HIGH_SELECT 의 동작을 예시하는 상태-전이 다이아그램이 도시되어 있다. 제어 신호들 PROCESSOR_LOW_SELECT 및 PROCESSOR_HIGH_SELECT 는 프로세서 버 스로부터 로칼 버스로 흐르는 덜 중요한 데이터 워드 및 더 중요한 데이터 워드로 향하여 출력한다. 제어 유닛(24)은 그 목적을 향하여 3 가지 상태들, 즉 MUX ADDRESS, MUX LESS SIGNIFICANT HALF 및 MUX MORE SIGNIFICANT HALF 를 조작한다. 제어 유닛(24)은 그것이 상태 MUX LESS SIGNIFICANT HALF 에 있다면 그리고 그 경우에 한해 제어 신호 PROCESSOR_LOW_SELECT를 선언한다. 제어 유닛(24)은 상태 MUX MORE SIGNIFICANT HALF에 있다면 그리고 그 경우에 한해 제어 신호 PROCESSOR_HIGH_SELECT 를 선언한다. 멀티플렉서(50)는 제어 유닛(24)이 제어 신호 PROCESSOR_LOW_SELECT도 제어 신호 PROCESSOR_HIGH_SELECT도 선언하지 않으면 ADDRESS 입력을 선택한다.

제어 유닛(24)은 제어 신호 RESET가 선언된 후에 MUX ADDRESS 상태로 초기화된다. 제어 유닛(24)은 함수 E가 논리적으로 참이면 MUX LESS SIGNIFICANT HALF 상태로 전이된다. 함수 E는 후술된다. 역으로, 제어 유닛(24)은 함수 E가 논리적으로 거짓이면 MUX ADDRESS 에서 MUX MORE SIGNIFICANT HALF 상태로 전이한다. 제어 유닛(24)은 함수 B가 논리적으로 참이면 MUX LESS SIGNIFICANT HALF 상태에서 MUX MORE SIGNIFICANT HALF 상태로 전이하며 또는 그 반대로도 된다. 제어 유닛(24)은 함수 C가 논리적으로 참이면 MUX LESS SIGNIFICANT HALF 상태에서 MUX ADDRESS 상태로 전이하거나 또는 MUX MORE SIGNIFICANT HALF 상태에서 MUX ADDRESS 상태로 전이한다. 제어 유닛(24)은 함수 D가 논리적으로 참이면, MUX MORE SIGNIFICANT HALF 상태나 또는 MUX LESS SIGNIFICANT HALF에서 유지된다.

함수 A 내지 E는 다음과 같이 정의된다.

$$A = \{LATCH[\overline{ADDRESS(29)}] \& DATA\_COUNT \neq 1 \& LE\_MODE \& MASTER\} OR$$

$$\{LATCH[ADDRESS(29)] \& [DATA\_COUNT = 1 OR\ \overline{LE\_MODE}] \& MASTER\} OR$$
$$\{[ADDRESS(29) XNOR\ LE\_MODE] \& TARGET\},$$

$$B = DEVSEL \& IRDY \& TRDY,$$

$$C = \overline{FRAME} \& \overline{DEVSEL},$$

$$D = DEVSEL \& [\overline{IRDY} OR \overline{TRDY}], \text{ and}$$

$$E = \{\overline{ADDRESS(29)} \& DATA\_COUNT \neq 1 \& LE\_MODE \& MASTER\} OR$$
$$\{ADDRESS(29) \& [DATA\_COUNT = 1 OR \overline{LE\_MODE}] \& MASTER\} OR$$
$$\{[ADDRESS(29) XOR\ LE\_MODE] \& TARGET \& DATARDY\}.$$

인터버스 버퍼(18)의 동작은 제 2 도 내지 제 6 도 및 제 9 도 및 제 10 도를 참조하여 설명된다.

예로, 제 3 도는 로칼 버스 디바이스에서 메인 메모리(14)의 어드레스 XX000으로 일 워드 기입(one word write)(바이트 ABCD) 또는 두개의 일-워드 기입들(바이트 ABCDEFGH)를 서술하는데 이용될 수 있다. 이런 트랜잭션에서, 제어 유닛(24)은 트랜잭션의 목적지 어드레스를 래칭하는 제어 신호 LATCH_ADDRESS를 먼저 선언한다. 이 어드레스는 프로세서 어드레스 버스로 출력되며, 여기서 메인 메모리(14)에 의해 유니크 저장 위치(unique storage location)를 선택하는데 사용된다. 그런 다음 제어 유닛(24)은 로칼 데이터 로우 래치(28) 또는 로칼 데이터 하이 래치(30) 중 어느 하나에 인입 데이터 워드를 래칭한다.

로칼 버스 디바이스가 LE 모드에서 동작하고 트랜잭션이 정렬된 이중-워드(ADDRESS(29)는 제로)의 제 1 워드에서 시작한다면, 제 1 의 4개의 인입 바이트들은 DCBA에 대응하고 제 2 의 4개의 인입 바이트들은 HGFE에 대응한다(이 바이트 순서는 제 3 도의 중앙부에 도시되어 있다). 이 경우에, 논리 함수 A는 참이다. 제어 유닛(24)은 로칼 데이터 로우 래치(28)가 인입 데이터를 샘플링하여 유지하게 한다. 위에서 언급한 바와 같이, 인입 데이터는 항상 적어도 한 번 바이트-교환된다(byte-swapped). 그러므로, 로칼 데이터 로우 래치(28)는 ABCD를 수신한다. 멀티플렉서(32)는 바이트 교환되지 않은 입력을 선택하여 바이트들 ABCD를 어드레스들 XX100 내지 XX111로 라우팅한다. 4개의 데이터 바이트들의 트랜스퍼후에 논리 함수 B는 참이다. 제어 유닛(24)은 로칼 데이터 하이 래치(30)가 인입 데이터를 샘플링하여 유지하게 한다. 다시, 인입 데이터는 항상 바이트-교환된다. 그러므로, 로칼 데이터 하이 래치(30)는 EFGH를 수신한다. 멀티플렉서(34)는 바이트 교환되지 않은 입력을 선택하여 어드레스들 XX000 내지 XX011로 바이트들 EFGH를 라우팅한다. 최종 바이트 배열, EFGHABCD가 제 3 도의 우측부에 도시되어 있다. 인터버스 버퍼(18)는 로칼 데이터 하이 래치(30)가 그 데이터를 래칭한 후에 적절한 바이트 인에이블 신호들을 가진 64 데이터 비트를 프로세서에 출력한다.

로칼 버스 디바이스가 상기 예의 BE 모드에서 동작한다면, 제 1 의 4개의 인입 바이트들은 ABCD 애 대응하고 제 2 의 4개의 인입 바이트들은 EFGH에 대응한다(이 바이트 배열은 제 3 도의 좌측부에 도시되어 있다). 여기서, 논리 함수 A는 거짓이며 제어 유닛(24)은 로칼 데이터 하이 래치(30)가 인입 데이터를 샘플링하여 유지하게 한다. 위에서 언급한 바와 같이, 인입 데이터는 항상 적어도 한번 바이트-교환된다. 그러므로, 로칼 데이터 하이 래치(30)는 DCBA를 수신한다. 멀티플렉서(34)는 그 바이트 교환된 입력을 선택하여 바이트들 ABCD를 어드레스들 XX000 내지 XX011로 라우팅한다. 4개의 데이터 바이트들의 트랜스퍼후에, 논리 함수 B는 참이고 제어 유닛(24)은 로칼 데이터 로우 래치(28)가 인입 데이터를 샘플링하여 유지하게 한다. 다시, 인입 데이터는 항상 바이트-교환된다. 그러므로, 로칼 데이터 로우 래치(28)는 HGFE 를 수신한다. 멀티플렉서(32)는 그 바이트 교환된 입력을 선택하여 어드레스 XX100 내지 XX111으로 바이트 EFGH를 라우팅한다. 최종 바이트 배열, ABCDEFGH는 제 3 도의 좌측부에 도시되어 있다. 인터버스 버퍼(18)는 로칼 데이터 로우 래치(28)가 그 데이터를 래칭한 후에 적절한 바이트 인에이블 신호들을 가진 64 데이터 비트를 프로세서 버스에 출력한다.

로칼 버스 디바이스가 데이터를 계속 소싱한다면, 제어 유닛(24)은 로칼 데이터 로우 래치(28)와 로칼 데이터 하이 래치(30)사이에서 계속 토글링(toggle)된다는 것을 유의해야 한다. 위에서 언급한 바와 같이, 로칼 버스 디바이스는 이전에 그 트랜잭션 크기를 표시하지 않을 것이다. 그러므로, 인입 데이터는 바이트 레인(lane) 멀티플렉싱에 의해 재배열될 수 없다. 제어 유닛(24)은 인입 데이터를 재배열해야 한다.

다른 예를 들어 설명을 계속하면, 제 3 도는 데이터 프로세서(12)로부터 로칼 버스의 어드레스를 XX000 및 XX100 각각으로의 바이트들 ABCD 및 EFGH의 두 개의 연속적 기입들을 서술하는데 이용될 수 있다. 이러한 트랜잭션에서, 제어 유닛(24)은 먼저 제어 신호 PROCESSOR_LOW_SELECT도 선언하지 않고 PROCESSOR_HIGH_SELECT도 선언하지 않으며, 멀티플렉서(50)가 트랜잭션의 목적지 어드레스를 출력하게 한다. 이 어드레스는 로칼 버스로 출력되며 디바이스들(16)중 하나에서 유니크 저장 위치를 선택하는데 사용된다. 반면에, 이중 워드 래치들(36, 38, 40 또는 42)중 하나는 정렬된 이중 쿼드-워드에서의 이중 워드의 위치에 의존하여 인입 이중 워드를 래칭한다. 정렬된 이중 쿼드-워드에서의 남아있는 3개의 이중-워드들은 '돈 케어'이다. 이중 워드 래치들(36, 38, 40 또는 42)에 의해 래칭된 데이터는 데이터 프로세서(12)와 인터버스 버퍼(18)가 BE 모드에 있을 때 BE 모드에 따라 배열된다. 이중 워드 래치들(36, 38, 40 또는 42)에 의해 래칭된 데이터는 데이터 프로세서(12)가 PLE 모드에 있고 인터버스 버퍼(18)가 LE 모드에 있을 때 PLE 모드에 따라 배열된다.

로칼 버스 디바이스가 LE 모드에서 동작하고 트랜잭션이 이중-워드(ADDRESS(29)가 제로임)의 제 1 워드에서 시작하고 DATA_COUNT 가 1이 아니면, 논리 함수 E는 참이다. 멀티플렉서(44)는 바이트들 ABCD를 멀티플렉서(48)에 출력하고 바이트들 EFGH를 멀티플렉서(46)에 출력한다(DATA 프로세서(12)는 PLE 모드에서 동작한다). 멀티플렉서들(46, 48)은 바이트 교환되지 않은 입력들을 선택하여 그 출력들을 멀티플렉서(50)로 라우팅한다. 제어 유닛(24)은 멀티플렉서 (50)가 멀티플렉서(48)의 출력, 바이트들 ABCD를 선택하게 한다. 위에서 언급한 바와 같이, 인출 데이터(outgoing data)는 멀티플렉서이후에 항상 바이트 교환된다. 그러므로, 로칼 버스 디바이스는 바이트들 DCBA를 어드레스들 XX000 내지 XX011 에 저장한다. 4개의 데이터 바이트들의 트랜스퍼후에, 논리 함수 B는 참이다. 제어 유닛(24)은 멀티플렉서(50)가 멀티플렉서(46)의 출력, 바이트들 EFGH를 선택하게 한다. 상술한 바와 같이, 인출 데이터는 바이트-교환된다. 그러므로, 로칼 버스 디바이스는 바이트 HGFE를 어드레스들 XX100 내지 XX111에 저장한다. 최종 바이트 배열, DCBAHGFE는 제 3 도의 중앙부에 도시되어 있다.

로칼 버스 디바이스가 BE 모드에서 동작하고 트랜잭션이 이중-워드(ADDRESS(29)는 제로)의 제 1 워드에서 시작한다면, 논리 함수 E는 거짓이다. 멀티플렉서(44)는 바이트들 ABCD를 멀티플렉서(46)에 출력하고 바이트들 EFGH를 멀티플렉서(48)에 출력한다(데이터 프로세서(12)는 BE 모드에서 동작한다). 멀티플렉서들(46 및 48)은 바이트 교환된 입력을 선택하여 그 출력들을 멀티플렉서 (50)에 라우팅한다. 제어 유닛(24)은 멀티플렉서(50)가 멀티플렉서(46)의 출력, 바이트들 DCBA를 선택하게 한다. 상술한 바와 같이, 인출 테이터는 멀티플렉서(50)후에 항상 바이트-교환된다. 그러므로, 로칼 버스 디바이스는 어드레스들 XX000 내지 XX011에 바이트들 ABCD를 저장한다. 4개의 데이터 바이트들의 트랜스퍼후에, 논리 함수 B는 참이다. 제어 유닛(24)은 멀티플렉서(50)가 멀티플렉서 (48)의 출력, 바이트 HGFE를 선택하게 한다. 위에서 언급한 바와 같이, 인출 데이터는 바이트-교환된다. 그러므로, 로칼 버스 디바이스는 어드레스들 XX100 내지 XX111에 바이트들 EFGH를 저장한다. 최종 바이트 배열, ABCDEFGH는 제 3 도의 좌측부에 도시되어 있다.

본 발명을 특정의 실시예를 참조하여 기술하였지만, 그 분야에 숙련된 자에게는 다른 수정들 및 개선들이 발생할 것이다. 예를 들어, 개시된 본 발명은 데이 터 프로세서(12)에 내장될 수도 있다. 그러므로, 본 발명은 첨부된 청구항들에서 정의된 바와 같이 본 발명의 정신 및 범주를 벗어나지 않는 모든 그와같은 수정들을 포함한다는 것이 이해되어야 한다.

발명의 효과

인터버스 버퍼는 다양한 서로 다른 트랜잭션 크기들 및 동작 모드들에도 불구하고 인터버스 데이터 트랜스퍼들을 조정한다.

(57) 청구의 범위

청구항 1.

인터버스 버퍼에 있어서,

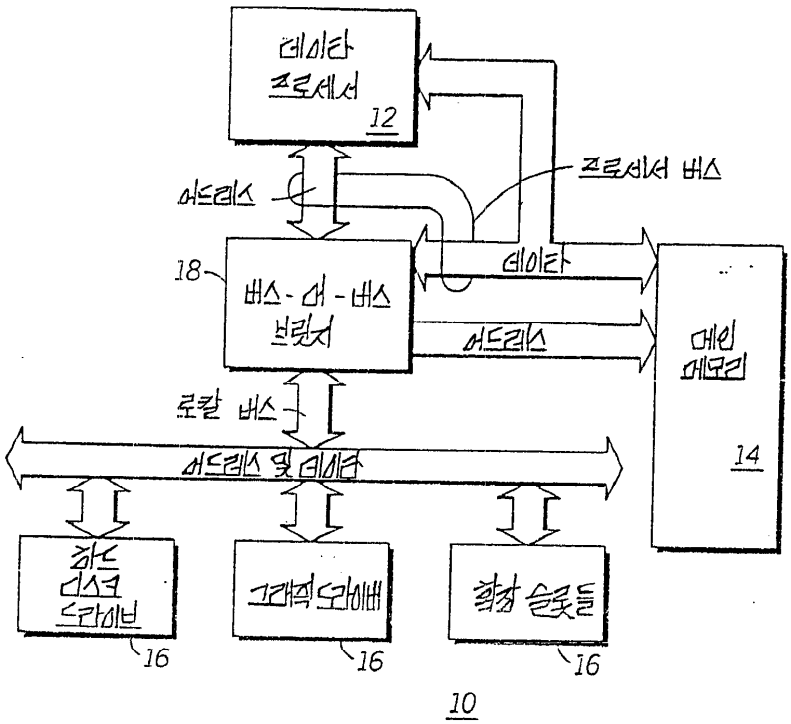
제 1 버스로부터 N개의 데이터 비트들을 수신하며 제 2 버스로 2N개의 데이터 비트들을 전송하는 제 1 데이터 경로로서, N은 정수이며, 상기 제 1 버스는 제 1 디바이스에 연결되어 빅 또는 리틀 인디언 프로토콜(a big or a little endian protocol)에 따라 데이터를 전송하고, 제 2 디바이스는 상기 제 2 버스에 연결되어 빅 인디언 프로토콜 또는 의사 또는 개조된 리틀 인디언 프로토콜(a big endian protocol or a pseudo- or munged little endian protocol)에 따라 데이터를 전송하고,
상기 제 1 버스에 연결되고, 제 1 제어 신호에 응답하여 2N개의 데이터 비트들 중 N개의 데이터 비트들을 래칭하는 제 1 래칭 수단과,
상기 제 1 래칭 수단의 출력과 상기 제 1 래칭 수단의 바이트 반전된 출력(byte reversed output)에 연결된 제 1 의 2:1 스위칭 수단으로서, 상기 제 1의 2:1 스위칭 수단의 출력은 상기 제 1 디바이스가 빅 인디언 바이트 배열 모드(a big endian byte ordering mode) 또는 리틀 인디언 바이트 배열 모드에서 동작하는지에 응답하여 선택되는, 상기 제 1 의 2:1 스위칭 수단과,
상기 제 1 버스에 연결되고, 제 2 제어 신호에 응답하여 상기 2N개의 데이터 비트들 중 서로 다른 N개의 데이터 비트들을 래칭하는, 제 2 래칭 수단과,
상기 제 2 래칭 수단의 출력과 상기 제 2 래칭 수단의 바이트 반전된 출력에 연결된 제 2 의 2:1 스위칭 수단으로서, 상기 제 2 의 2:1 스위칭 수단의 출력은 상기 제 1 디바이스가 상기 빅 인디언 바이트 배열 모드에서 또는 상기 리틀 인디언 바이트 배열 모드에서 동작하는지에 응답하여 선택되며, 상기 제 1 및 제 2 의 2:1 스위칭 수단의 출력은 상기 제 2 버스로 출력된 데이터를 형성하도록 연결되는 상기 제 2 의 2:1 스위칭 수단을 포함하는, 상기 제 1 데이터 경로와,
상기 제 2 버스로부터 2N개의 데이터 비트들을 수신하고 상기 제 1 버스로 N개의 데이터 비트들을 전송하는, 제 2 데이터 경로로서,
상기 제 2 버스로부터 2N개의 데이터 비트들 중 제 1 의 N개의 순차 비트들 및 상기 2N개의 데이터 비트들 중 상기 제 1 의 N개의 순차 비트들의 바이트 반전(a byte reversal)을 수신하는 제 3 의 2:1 스위칭 수단으로서, 상기 제 3 의 2:1 스위칭 수단의 출력은 상기 제 2 디바이스가 빅 인디언 바이트 바이트 배열 모드 또는 '개조된' 리틀 인디언 바이트 배열 모드('munged' little endian byte ordering mode)에서 동작하는지에 응답하여 선택되는, 상기 제 3 의 2:1 스위칭 수단과,
상기 제 2 버스로부터 2N개의 데이터 비트들 중 제 2 의 N개의 순차 비트들 및 상기 2N개의 데이터 비트들 중 상기 제 2 의 N개의 순차 비트들의 바이트 반전을 수신하는 제 4 의 2:1 스위칭 수단으로서, 상기 제 4 의 2:1 스위칭 수단의 출력은 상기 제 2 디바이스가 빅 인디언 바이트 배열 모드 또는 '개조된' 리틀 인디언 바이트 배열 모드에서 동작하는지에 응답하여 선택되는, 상기 제 4 의 2:1 스위칭 수단과,
상기 제 3 스위칭 수단의 출력과 제 4 스위칭 수단의 출력을 수신하는 제 5 의 2:1 스위칭 수단으로서, 상기 제 5 의 2:1 스위칭 수단의 출력은 제 3 제어 신호에 응답하여 선택되는 상기 제 5 의 2:1 스위칭 수단을 포함하는, 상기 제 2 데이터 경로와,
상기 제 1 데이터 경로와 상기 제 2 데이터 경로에 연결된 제어 유닛으로서, 상기 인터버스 버퍼에 의해 트랜스퍼링될 데이터의 메모리 어드레스를 식별하는 입력 어드레스에 응답하여 상기 제 1, 제 2 및 제 3 제어 신호들을 발생시키는 상기 제어 유닛을 포함하는 인터버스 버퍼.

청구항 2.

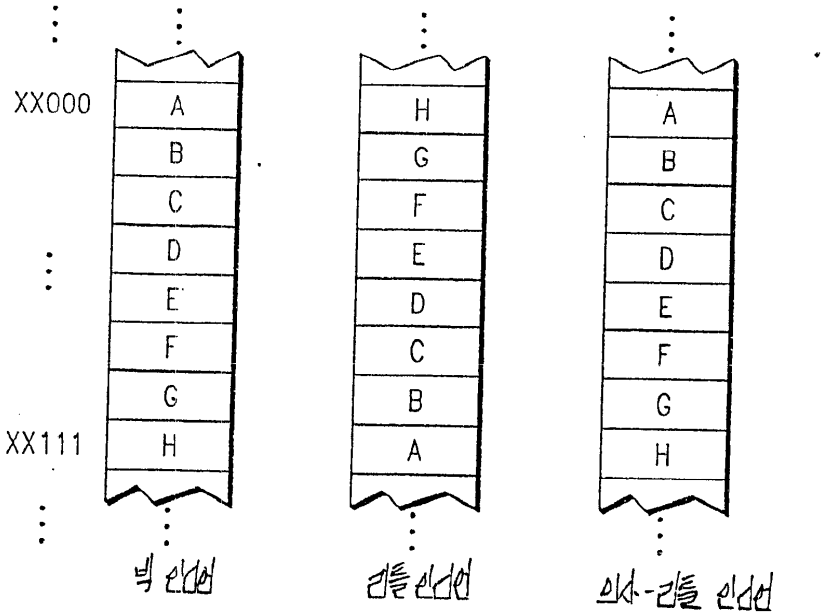
제 1 항에 있어서,
상기 제 1 및 제 2 래칭 수단들은 상기 제 1 버스의 바이트 반전에 연결되는, 인터버스 버퍼.

도면

도면1

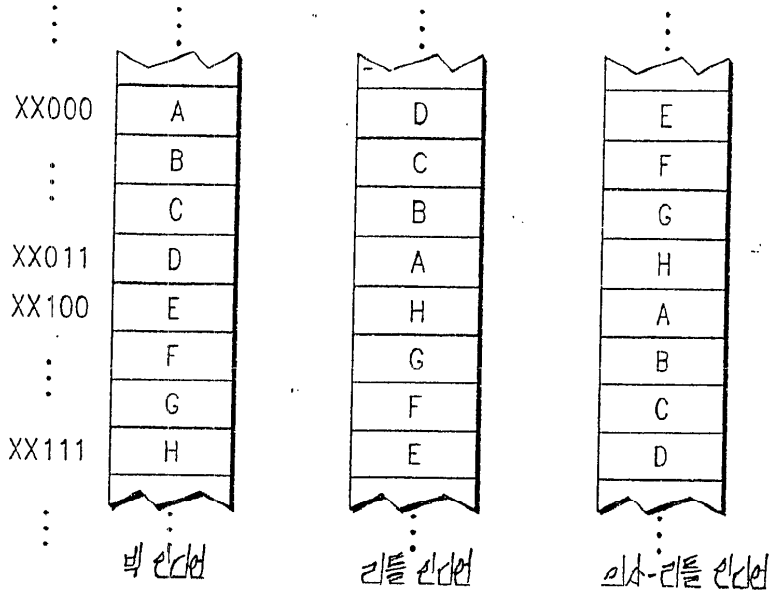


도면2



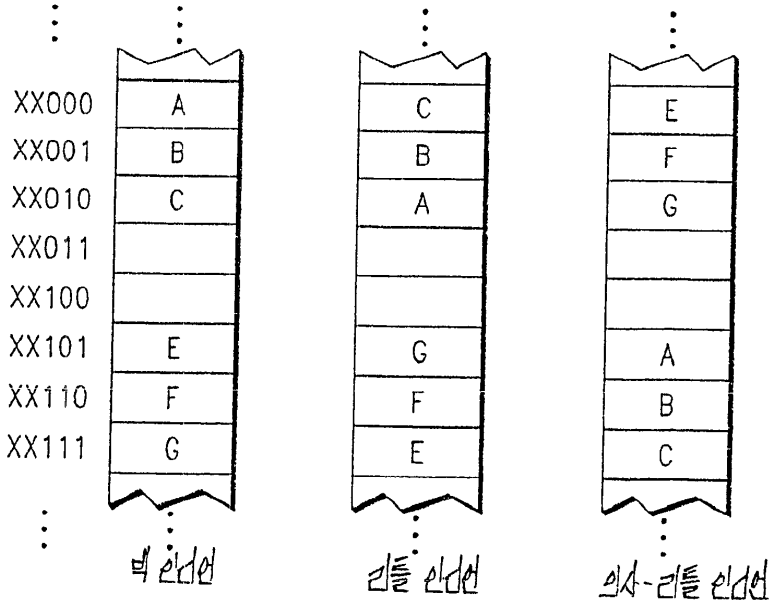
1 x 8- 바이트 트랜스퍼

도면3



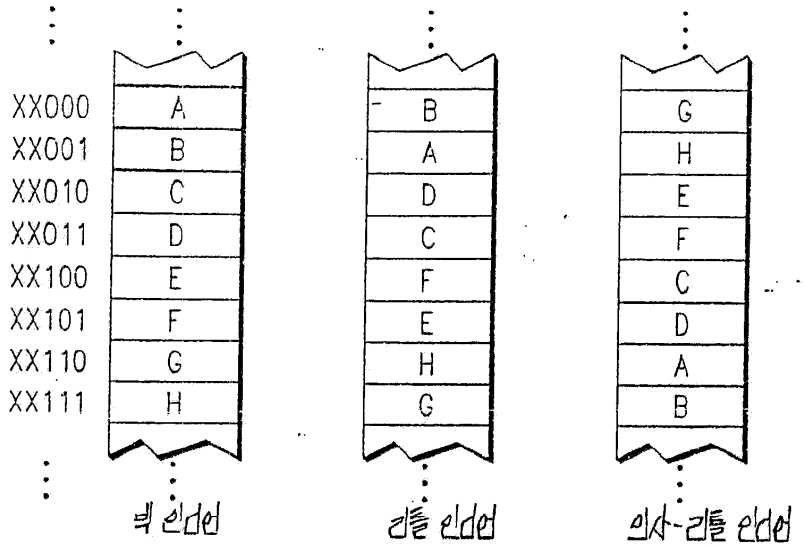
2 x 4- 바이트 트랜스퍼들

도면4



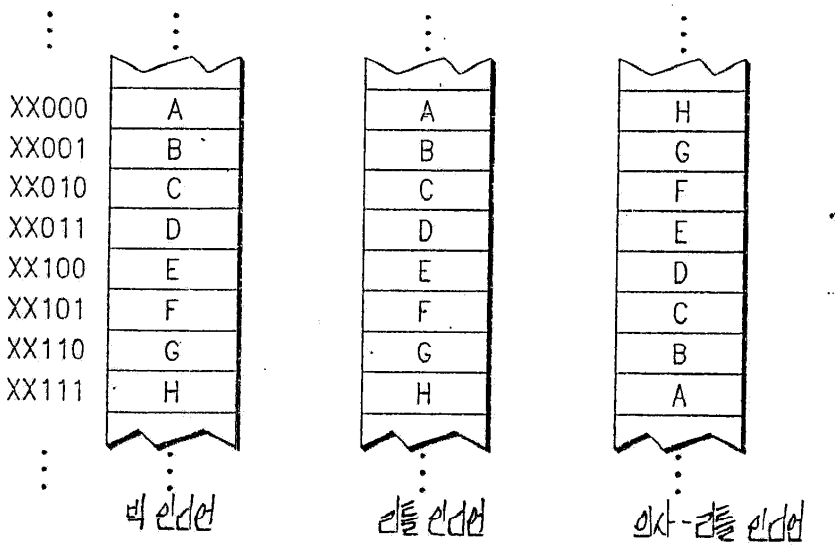
2 x 3- 바이트 트랜스퍼들

도면5



4 x 2-바이트 트랜스퍼들

도면6

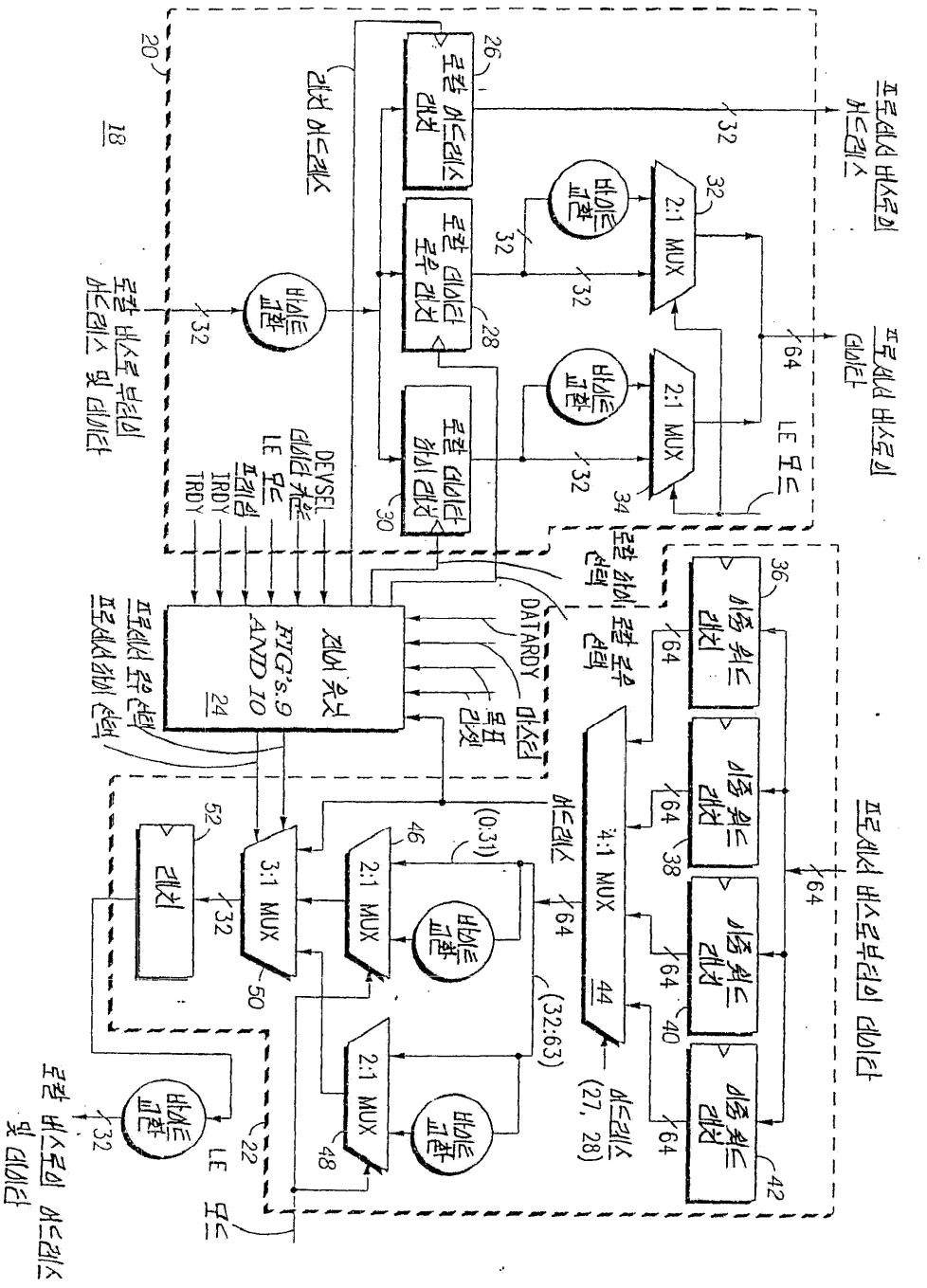


8 x 1-바이트 트랜스퍼들

도면7

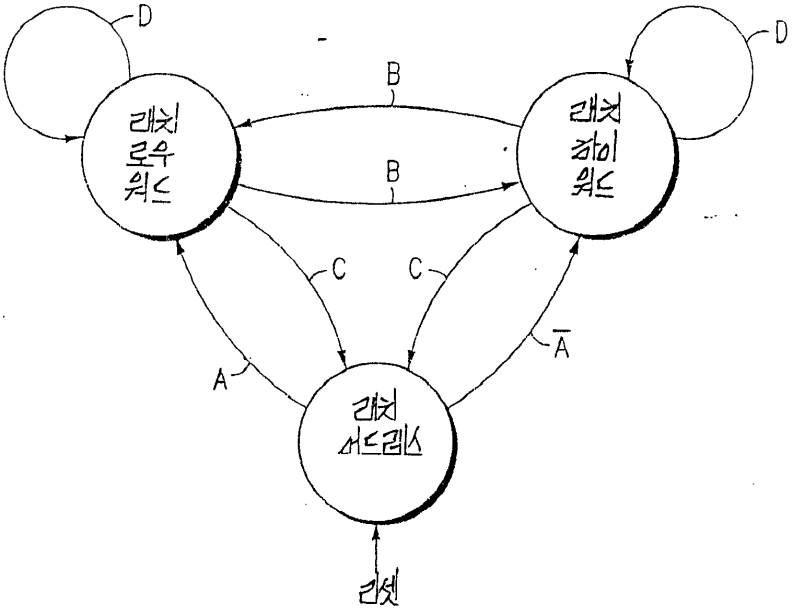
트랜스퍼 크기	BE-TO-PLE
1	ADDRESS XOR (111)
2	ADDRESS XOR (110)
3	ADDRESS XOR (101)
4	ADDRESS XOR (100)
8	ADDRESS XOR (000)

도면8

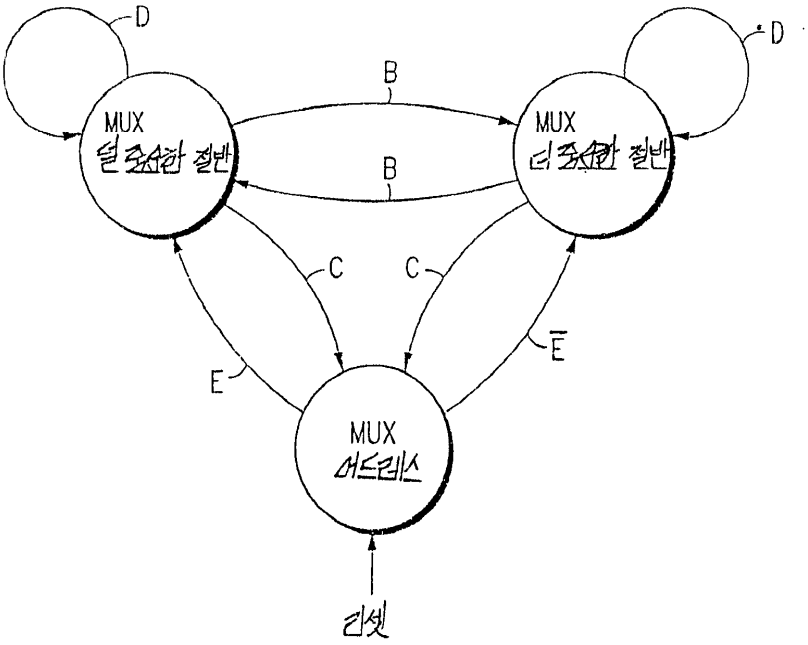


프로세서 버스로의 어드레스
프로세서 버스로의 데이타
로컬 버스로 부터의 어드레스 및 데이타
로컬 버스로의 어드레스 및 데이타
로컬 어드레스 래치
로컬 데이타 로우 래치
로컬 데이타 하이 래치
바이트 교환
2:1 MUX
LE 모드
DEVSEL
데이타 취소
LE 모드
프레임
IRDY
TRDY
제어 유닛
FIG's.9 AND 10
DATARDY
마스터
목표 리셋
프로세서 로우 선택
프로세서 하이 선택
로컬 하이 선택
로컬 로우 선택
어드레스
4:1 MUX
3:1 MUX
(0:31)
(32:63)
(27, 28)
래치

도면9


D
B
래치
로우
워드
래치
하이
워드
D
B
C
C
A
$\overline{A}$
래치
서브레스
리셋

도면10


D
B
MUX
D
B
C
C
E
$\overline{E}$
MUX
서브레스
리셋