

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl.⁶
G06F 13/42

(45) 공고일자 1999년03월30일
(11) 등록번호 특0172310
(24) 등록일자 1998년10월23일

(21) 출원번호	특1995-064423	(65) 공개번호	특1997-049692
(22) 출원일자	1995년12월29일	(43) 공개일자	1997년07월29일

(73) 특허권자 현대전자산업주식회사 김주용
경기도 이천군 부발읍 아미리산 136-1
(72) 발명자 이석중
전북 익산시 용동면 대조리 760-3
김한홍
경기도 성남시 분당구 서현310 효자촌대창아파트 617-605
(74) 대리인 박해천, 영주석, 원석희

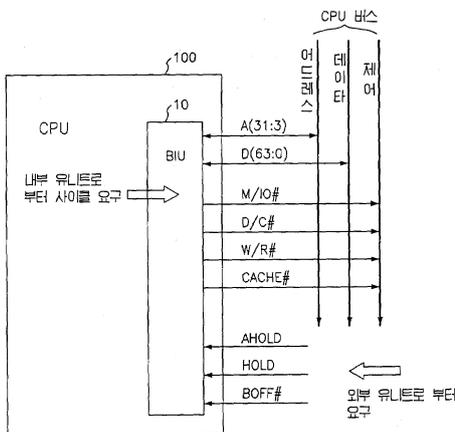
심사관 : 오홍수

(54) 교착 상태 방지를 위한 버스 유닛

요약

본 발명은 마이크로프로세서에서 외부 칩셋과 내부 유닛들과의 인터페이스(interface)를 담당하고 있는 교착 상태 방지를 위한 사이클 큐를 갖는 버스 유닛에 관한 것으로, 버스 유닛으로 들어오는 사이클 요청을 중재하여 가장 높은 우선 순위의 장치에 사이클 진행을 허가하는 사이클 요청에 대한 중재수단; 상기 중재수단으로부터 오는 사이클 승인 신호에 따라 후술할 사이클 큐에 저장할 것을 선택하는 큐 입력 다중화수단; 상기 큐 입력 다중화수단으로부터의 진행시켜야 할 사이클이 데이터 버스에 대해서 정렬이 되었는가를 검사하는 비정렬 사이클에 대한 검사수단; 상기 사이클 큐에 대한 제어를 위해 큐에 새로운 사이클을 저장하거나, 비정렬된 사이클에 대한 제어, 양보 조건이 없어지는 경우 사이클 재시작 및 사이클 큐를 비우는 큐 제어수단; 상기 큐 제어수단의 신호에 따라 상기 큐 입력 다중화수단 또는 비정렬 사이클에 대한 검사수단으로부터 오는 새로운 사이클을 받아들여 저장하거나, 사이클 진행이 완료된 큐를 지우는 사이클 큐를 구비하는 것을 특징으로 한다.

대표도



명세서

[발명의 명칭]

교착 상태 방지를 위한 버스 유닛

[도면의 간단한 설명]

제1도는 펜티움 프로세서의 버스 인터페이스 유닛의 개략도.

제2도는 버스 인터페이스 유닛중 사이클 큐와 관련한 버스 유닛의 구성도.

제3도는 사이클 큐의 각 부분 예시도.

* 도면의 주요부분에 대한 부호의 설명

11 : 큐 입력 다중화부

12 : 사이클 큐

13 : 중재기

14 : 비정렬 사이클 검사부

15 : 큐 제어기

[발명의 상세한 설명]

본 발명은 마이크로프로세서에서 외부 칩셋과 내부 유닛들과의 인터페이스(interface)를 담당하고 있는 교착 상태 방지를 위한 사이클 큐를 갖는 버스 유닛에 관한 것이다.

버스 유닛은 외부로의 사이클의 진행과, 외부로부터 사이클을 받아들이는 일을 주로 담당한다. 경우에 따라 외부 시스템이 필요에 의해 진행중이던 사이클에 대한 중지의 요구나 재시작을 요구하는 경우가 있는데, 이러한 상황에 사이클의 중지 및 재시작을 효율적으로 할 수 있도록 해준다.

또한 여러개의 마스터가 있는 경우에 흔히 발생하는 교착상태(deadlock) 조건을 벗어나기 위해서는 일종이 양보 또는 후퇴 기능(backoff)을 지원해야 하며, 이런 시스템에서 진행하던 사이클의 중단과 재시작을 보장하기 때문이다.

여기서, 상기 교착상태란 여러개의 유닛이 서로 다른 유닛이 갖고 있는 공유 자원을 원하며, 자신은 다른 유닛에 자기가 갖고 있는 공유 자원에 대해 양보를 하지 않음으로써 발생하는 상태로, 아무런 동작을 할 수 없는 상태다.

이러한 교착 상태에서 벗어나기 위해서는 유닛들 중의 하나가 양보(backoff)하여, 다른 유닛에게 자기가 갖고 있던 공유자원을 넘겨줌으로써 그것을 이용해 다른 유닛이 일을 마친 후, 그 유닛이 놓아준 공유자원을 이용해 자신의 일을 마치는 방법이 있다.

여러 개의 마스터를 갖는 시스템에서는 이 발생 가능한 교착상태 방지를 위해서 양보 기능이 필수적이다. 또한 이 양보를 하는 유닛에서는 정확하게 중지한 사이클을 재시작이 가능하도록 해주는 것이 필요하다.

상기 종래 기술에 대한 문제점을 해결하고, 여러 개의 버스 마스터가 있는 시스템, 특히 다중 프로세서(multi-processor) 시스템에서는 양보의 기능이 아주 중요하며, 이것을 자연스럽게 처리할 수 있음으로써 정확한 동작을 보장 할 수 있게 하며, 이러한 사이클 큐를 양보에 대한 저장 수단으로만 이용하는 것이 아니라, 정상적인 사이클 수행에도 이용하고, 특히 비정렬 사이클까지 처리함으로써 자원에 대한 효율성을 증대시킨 교착 상태 방지를 위한 버스 유닛을 제공하는 데 그 목적이 있다.

상기 목적을 달성하기 위하여 본 발명은, 버스 유닛으로 들어오는 사이클 요청을 중재하여 가장 높은 우선 순위의 장치에 사이클 진행을 허가하는 사이클 요청에 대한 중재수단; 상기 중재수단으로부터 오는 사이클 승인 신호에 따라 후술할 사이클 큐에 저장할 것을 선택하는 큐 입력 다중화수단; 상기 큐 입력 다중화수단으로부터의 진행시켜야할 사이클이 데이터 버스에 대해서 정렬이 되었는가를 검사하는 비정렬 사이클에 대한 검사수단; 상기 사이클 큐에 대한 제어를 위해 큐에 새로운 사이클을 저장하거나, 비정렬된 사이클에 대한 제어, 양보 조건이 없어지는 경우 사이클 재시작 및 사이클 큐를 비우는 제어수단; 상기 큐 제어수단의 신호에 따라 상기 큐 입력 다중화수단 또는 비정렬 사이클에 대한 검사수단으로부터 오는 새로운 사이클을 받아들여 저장하거나, 사이클 진행이 완료된 큐를 지우는 사이클 큐를 구비하는 것을 특징으로 한다.

이하, 첨부된 도면을 이용하여 본 발명의 일실시예를 상세히 설명한다.

먼저, 제1도는 펜티움 프로세서의 버스 인터페이스 유닛에 대한 간단하게 나타낸 것으로 실제 많은 핀들이 있으나, 버스 유닛을 통해 외부로 사이클이 진행되는 상황에서 사이클의 종류를 나타내는 핀들과, 외부 유닛이 펜티움 프로세서에게 버스 사용 중지를 요구하는데 사용되는 핀들만 나타낸다.

제2도는 버스 인터페이스 유닛 중 사이클 큐에 관련된 부분만 나타낸 것으로 실제로는 더 많은 유닛들이 사이클 큐와 관계를 맺고 있지만 그 중 중요한 블록만 나타낸다.

제3도는 실제 사이클 큐의 여러 부분에 대해 나타낸 것이다.

펜티움(Pentium) 프로세서의 경우, 양보기능을 지원하기 위해 B0FF# 라는 핀을 두고 있으며, 이 핀이 액티브(active)상태로 가능 경우는 진행 중이던 사이클을 중지하고, 다시 이 핀이 인액티브(inactive)상태로 가는 경우에 중지된 사이클을 재시작하도록 되어 있다.

펜티움 프로세서의 버스 사이클이 486프로세서에 비해 크게 달라진 것은 486프로세서의 경우는 한번에 1개의 버스 사이클만 진행이 가능했으나, 펜티움의 경우는 2개까지 가능하다. 또한 외부 칩셋이 CPU 버스에 대한 구동 중지 신호로 B0FF# 신호를 보내는 경우, 486의 경우는 사이클이 중지된 그 시점에서 다시 시작하나, 펜티움이 경우는 중지된 사이클은 처음부터 다시 시작한다.

그러므로 진행중이던(최대) 2개의 사이클에 대한 중지 및 그것들에 대한 확실한 재시작을 가능하게 해주는 수단으로 본 발명을 고안하게 되었다. 본 발명의 큐는 2개의 버스 사이클에 대한 정보를 저장하기 위해 2개의 entry로 구성되었다.

프로세서 내부의 각 유닛으로 버스 사이클 진행에 대한 요청 신호를 받아서, 이것들 중 우선순위가 높은 것을 먼저 사이클 큐로 저장한다. 이 사이클 큐는 사이클 재시작에 필요한, 사이클에 관계된 정보를 저장하는 여러 개의 부분으로 구성되어 있다. 만약에 양보 요구에 의해 사이클이 정지되는 경우는 사이클을 정보를 그대로 갖고 있다가, 양보 조건이 없어지면 사이클 큐가 버스 사이클 재 진행을 책임지게 된다.

이러한 기본적인 기능 이외에도 큐에는 데이터버스(data bus)의 경계(boundary)에 걸리는 경우에 처리를 위한 기능이 추가 되었다.

일반적으로 버스 인터페이스 유닛은 프로세서 코어(core)의 각 유닛으로부터 오는 버스 사이클에 대한 요구를 받아들여서 그것을 외부로 전하는 일을 하거나, 외부로부터 오는 여러 가지 요구를 CPU 내부로 넘겨

주는 일종의 중개 역할을 담당한다.

그러면, 상기한 도면에 대한 간단한 설명을 각각의 도면을 참조하여 상세하게 설명한다.

제1도는 본 발명의 적용 대상이 되는 펜티움 프로세서의 버스 인터페이스 유닛(BIU)에 대한 간단한 개념도이다.

펜티움 프로세서의 경우 코아로부터의 사이클 요구로는 메모리에 대한 데이터나 코드(code)를 읽거나 쓰기, I/O 장치에 대한 읽기나 쓰기, 외부 인터럽트에 대해 인터럽트 벡터를 구하기 위한 인터럽트 인정 사이클(interrupt acknowledge cycle) 및 현재 프로세서가 특정 명령어를 수행하고 있거나 특정 상태에 있음을 알리는 스페셜 사이클(special cycle)이 있다. 이러한 사이클의 종류를 나타내기 위해서 펜티움의 경우는 M/I0#, D/C#, W/R#, CACHE# 핀 등을 사용하고 있다.

외부로부터 프로세서에 대한 요구로는 다음 3가지 종류가 있다.

첫째로 'AHOLD'라는 핀을 사용하여 내부 캐쉬에 어떤 특정 캐쉬라인(cache line)이 있는가 조사하는 스누프(snoop) 사이클을 위해 프로세서의 어드레스 버스만을 플로트(float)시킨 후 외부 칩셋이 그 어드레스 버스를 구동하는 방식이 있다. 두 번째는 외부 칩셋이 CPU 버스를 사용하기 위해서 프로세서가 CPU 버스를 구동하는 것에 대한 중지를 요구하되 현재 진행 중인 사이클이 종료하면, CPU 버스를 플로트시키라는 'HOLD'핀이 있다. 나머지 하나는 버스를 플로트시키는 것은 같지만 진행하던 사이클도 중지시킨 채로 바로 다음 클럭에 버스를 플로트시키라는(양보 요구) 강력한 기능을 가진 BOFF# 핀도 있다.

펜티움 프로세서의 경우 최대 2개까지의 사이클을 진행시킬 수 있기 때문에, 양보 상황이 되는 경우 진행되던 사이클이 최대 2개까지 중단 될 수 있다. 이렇게 중단된 사이클은 양보 상황이 없어지는 경우에 재시작을 해야 하는데, 재시작의 방법으로는 사이클을 요청한 모든 유닛이 양보시 그 사이클이 중지된 것을 기억하고 있다가 다시 버스 사이클을 요청하는 방법이 있고, 다른 하나는 버스 유닛이 중지된 사이클에 대한 정보를 갖고 있다가 재시작하는 방법이 있다.

본 발명의 경우는 그 중 후자의 방법을 위한 것이다. 전자의 방법처럼 모든 유닛이 양보에 의해서 중지된 사이클을 저장하고 있는 부담을 덜기 위해서, 버스 유닛에서 일종의 버스 사이클에 대한 저장 장치 역할을 하는 사이클 큐라는 것을 두었다.

원래 사이클 큐는 양보시 정지된 사이클을 재시작하기 위한 것이나, 여기서는 양보 상태가 아닌 정상적인 사이클의 경우는 현재 진행 중인 사이클에 대한 주소(address), 오퍼랜드의 크기(os), 기타 버스 사이클이 종류를 알리는 신호등에 대한 일시적인 저장 장치(temporary latch)로서도 이용된다(위 신호들은 버스 사이클이 진행되는 동안 계속 유효한 값으로 유지 되어야 하기 때문에 래치에 저장하고 구동하는 것이 바람직하다.)

또한, 진행할 사이클이 데이터 버스의 경계(boundary)에 걸리는 경우의 처리에도 이용할 수 있다. 이 경우는 1개를 받아들여, 버스 유닛이 2개의 사이클을 만들어 주는데, 이를 위해서는 직전 사이클에 대한 정보를 저장하고 있는 것이 필요하다. 이 때 사이클 큐를 사용하게 되면 이것을 간단히 처리할 수 있다.

제2도는 본 발명에 따른 사이클 큐를 포함하는 버스 인터페이스 유닛의 구성도이다.

먼저, 사이클 요청에 대한 중재기(request arbiter)(13)는 버스 유닛으로 들어오는 사이클 요청을 중재하여 가장 높은 우선 순위의 장치에 사이클 진행을 허가하는 일을 담당한다.

큐 입력 다중화기(queue input multiplexer)(11)는 상기 중재기(13)로부터 오는 사이클 승인(acknowledge) 신호에 따라 사이클 큐에 저장할 것을 선택하는 역할을 한다.

비정렬 사이클에 대한 검사부(Misalign checker)(14)는 진행시켜야 할 사이클이 데이터 버스에 대해서 정렬이 되었는가를 살피기 위한 것으로, 입력 주소의 하위 비트와 오퍼랜드의 크기 부분을 받아들여, 사이클을 1 사이클에 처리할 수 있는지의 여부를 살핀다. 즉 펜티움 프로세서의 경우 데이터 버스의 폭이 64 비트이므로, 총 8바이트의 데이터를 1사이클에 처리가 가능하다. 여기서 데이터 버스의 최하위 바이트의 주소는 항상 8바이트 단위로 정렬되어 있는 상태이다. 그러므로 실제 처리해야 할 데이터가 8바이트보다 작더라도 시작 주소(start address)에 따라서, 1사이클에 처리할 수 없는 경우도 있다.

큐 제어부(queue control)(15)는 사이클 큐에 대한 제어를 담당하는 부분으로 큐에 새로운 사이클을 저장하거나, 비정렬된 사이클에 대한 제어, backof조건이 없어지는 경우 사이클 재시작 및 사이클 큐를 비우는 일등을 처리한다.

사이클 큐(cycle queue)(12)는 상기 큐 제어부(15)의 신호에 따라 큐입력 다중화기나 비정렬 사이클에 대한 검사부로부터 오는 새로운 사이클을 받아들여 저장하거나, 사이클 진행이 완료된 큐를 지우기도 한다.

프로세서 내부의 여러 유닛으로부터 오는 사이클 진행에 대한 요청은 이것을 중재하는 상기 중재기(request arbiter)(13)를 거쳐 가장 우선순위가 높은 유닛이 결정된다. 그러면, 상기 중재기(13)는 사이클 진행을 허락하는 승인(acknowledge)신호를 보내며, 이 신호는 큐의 입력을 결정하는 다중화기(queue input multiplexer)로 들어가서, 승인 받은 유닛으로부터의 어드레스, 오퍼랜드 크기 및 버스 사이클이 종류를 결정하는 신호로 쓰인다. 여기서 어드레스의 하위 비트와 오퍼랜드 크기는 비정렬 사이클(misaligned cycle) 여부를 결정하여 큐에 쓴다.

큐에 저장된 주소는 외부로 진행시킬 주소가 된다. 오퍼랜드의 크기를 나타내는 부분을 시작 주소의 하위 3비트와 함께 바이트 인에이블(byte enable)신호인 BE[7:0]#를 만드는데 사용된다. 또한 사이클 종류를 나타내는 부분을 사이클이 제어에 사용되며, PCD, PWT 부분은 외부 캐쉬를 제어하는데 사용된다. F와 I 비트는 비정렬 사이클에 대한 목적으로 사용된다.

제3도는 사이클 큐의 각 부분을 나타내며, 그것의 의미는 다음과 같다.

- 올더(olider) : 큐의 2개의 엔트리 중 먼저 처리를 받아야 할 것을 나타내는, 즉 다른 말로 하면 먼저

쓰여진 엔트리를 나타내는 일종의 프래그이다.

- halt(halt) : 양보 상황이 발생하여 중단된 사이클이 있음을 나타낸다. 양보 조건이 없어지는 경우 이 비트가 셋트 상태라면 사이클큐에 저장되어 있는 사이클을 우선적으로 진행시킨다.
- 어드레스(addr) : 진행시킬 사이클의 주소, 실제 외부로 나가는 주소는 상위 29비트뿐이지만, 바이트 인에이블 신호를 만들기 위해 32비트가 모두 저장된다.
- os : 진행될 사이클의 오퍼랜드의 크기를 나타낸다.
- F : 비정렬 사이클(misaligned cycle)의 첫 번째 사이클임을 나타낸다. 펜티엄 프로세서의 경우 비정렬 사이클이면, 비정렬된 상위 사이클로부터 진행시킨 것을 원칙으로 하기 때문에 이 부분을 두었으며, 정렬된 사이클의 경우 이 부분의 의미는 없다.
- I : 버스 유닛의 경계에 걸리는 비정렬 사이클임을 나타내는 부분이다.
- 버스 사이클 : 진행할 사이클의 종류를 나타내기 위한 부분으로, 총 6비트로 구성되어 있다.
- pcd, pwt : 외부 캐시를 사용하는 경우 외부 캐시를 제어하는 구실을 한다.
- V : 큐에 유효한 데이터가 들어있는지의 여부를 나타내는 valid bit 이다.

큐의 2개의 엔트리 중 먼저 처리를 받아야 할 것을 나타내는 목적으로 사용된다. 1개의 버스 사이클만 진행되는 경우는 현재 진행되는 사이클이 저장되어 있는 큐를 나타내며, 버스 사이클 파이프 라이닝에 의해 2개의 사이클이 진행되는 경우는 먼저 진행된 사이클을 나타내기 위해서 사용된다.

2개의 큐 엔트리 중 먼저 쓰여진 것을 나타내는 것은 위치에 따라 쉬프터 체인(shifter chain)을 이용해서 고정된 순서를 나타내는 대신에, 일종의 프래그 구실을 하는 이 올더의 값에 따라 결정되는 것으로 한다. 이 2개의 큐에 편의상 이름을 붙이면 F큐와 R큐라고 하고, 이 올더가 '1'이면 F큐, '0'이면 R큐가 먼저 쓰여졌다는 것을 나타내기로 한다.

상기 사이클 큐(12)에 새로운 사이클을 저장하는 것은 다음과 같이 한다. 큐가 비어 있는 상황에서 새로운 사이클이 시작되는 경우에는 무조건 F큐에 새로운 사이클에 관계된 정보를 쓰고 올더를 F큐를 가리킬 수 있도록 하며, F큐의 V(valid bit)를 셋트한다. 1개의 큐가 채워진 상태에서, 버스 사이클 파이프라이닝에 의해서 새로운 사이클을 시작하고자 하는 경우는 올더가 가리키는 엔트리가 아닌 다른 엔트리에 새로운 사이클을 저장하고 그 엔트리의 유효 비트를 셋트한다. 2개의 엔트리가 모두 채워져 있는 상황에서 더 이상 새로운 사이클을 받아들일 수 없게 된다.

진행하던 사이클이 종료되는 경우는 다음과 같이 관리한다. 2개의 엔트리에 모두 사이클이 저장되어 있던 경우는 올더가 가리키는 엔트리의 유효비트를 클리어시킨다. 그리고 나서 올더를 토글시켜서, 막사이클이 종료된 엔트리가 아닌 다른 엔트리가 이제 먼저 처리해야 할 사이클이 되었음을 나타낸다. 1개의 엔트리에만 사이클이 저장되어 있는 경우는 해당 엔트리의 유효 비트를 클리어시키고 역시 올더를 토글시킨다. 여기서 올더를 토글시키는 동작은 아무런 의미가 없으나 실제로 회로로 구현할 때 쉬워지는 이점이 있다.

사이클을 진행하던 도중에 양보 상황이 발생하는 경우에는 진행 중이던 사이클이 있는 경우 그대로 정지하고, 할트 비트를 셋트한다. 양보 조건이 없어지는 경우는 사이클 큐에 남아 있던 사이클이 모두 진행된 후에 할트 비트를 클리어시킨다.

상기 사이클 큐(12)로부터 새로운 사이클을 진행시켜야 하는 경우는 2가지 경우가 있다.

첫 번째 경우는 위에서 살펴본 바와 같이 양보 상황이 발생하는 경우는 진행중이던 사이클이 있는 경우는 할트 비트를 셋트시켜 중단된 사이클이 있음을 표시한다. 그리고 나서 양보 조건이 없어지면, '사이클 큐 제어부'는 이 할트 비트를 검사해서 셋트되어 있는 경우는 '사이클 중재기'로 신호를 보낸다. 상기 사이클 중재기(13)는 상기 사이클 큐(12)로부터의 요청을 우선적으로 처리해 준다.

두 번째 경우는 데이터 버스의 경계에 걸려서 단일 사이클로 처리할 수 없는 경우다. 즉, 'F'와 'I'비트가 셋트되어 있는 상황에서는 현재 진행되고 있는 사이클이, 비정렬 사이클의 첫 번째 사이클이라는 의미이므로, 사이클 큐 제어부가 다음 사이클 진행을 위해 사이클 중재기로 신호를 보내며, 사이클 중재기는 역시 이 요구에 대해서 우선적으로 처리를 해준다. 이렇게 진행된 두 번째 사이클의 경우는 'F'비트는 클리어시켜서 비정렬 사이클의 두 번째 사이클이 진행되고 있다는 것을 나타낸다.

본 발명은 양보가 요구되는 상황에서 사이클의 중지와 재시작이 필요한 경우에 이용이 가능하다. 즉, 펜티엄 프로세서에서만 한정되지 않고, 다른 구조를 갖는 프로세서에도 응용이 가능하다.

상기와 같은 본 발명은 여러 개의 버스 마스터가 있는 시스템, 특히 다중 프로세서(multi-processor) 시스템에서는 양보의 기능이 아주 중요하며, 이것을 자연스럽게 처리할 수 있음으로써 정확한 동작을 보장할 수 있게 되며, 이러한 사이클 큐를 양보에 대한 저장 수단으로만 이용하는 것이 아니라, 정상적인 사이클 수행에도 이용하고, 특히 비정렬 사이클까지 처리함으로써 자원에 대한 효율성을 훨씬 증대시킬 수 있는 효과가 있다.

(57) 청구의 범위

청구항 1

버스 인터페이스 유닛에 있어서, 버스 유닛으로 들어오는 사이클 요청을 중재하여 가장 높은 우선 순위의 장치에 사이클 진행을 허가하는 사이클 요청에 대한 중재수단(13); 상기 중재수단(13)으로부터 오는 사이클 승인 신호에 따라 후술할 사이클 큐(12)에 저장할 것을 선택하는 큐 입력 다중화수단(11); 상기 큐 입력 다중화수단(11)으로부터의 진행시켜야 할 사이클이 데이터 버스에 대해서 정렬이 되었는가를 검사하는 비정렬 사이클에 대한 검사수단(14); 상기 사이클 큐(12)에 대한 제어를 위해 큐에 새로운 사이클을

저장하거나, 비정렬된 사이클에 대한 제어, 양보 조건이 없어지는 경우 사이클 재시작 및 사이클 큐를 비우는 큐 제어수단(15); 상기 큐 제어수단(15)의 신호에 따라 상기 큐 입력 다중화수단(11) 또는 비정렬 사이클에 대한 검사수단(14)으로부터 오는 새로운 사이클을 받아들여 저장하거나, 사이클 진행이 완료된 큐를 지우는 사이클큐(12)를 구비하는 것을 특징으로 하는 교착 상태 방지를 위한 버스 유닛.

청구항 2

제1항에 있어서, 상기 사이클 큐(12)는, 버스 사이클 파이프라인을 적용하기 위해서 전체 파이프라인의 개수만큼의 큐 엔트리를 갖는 것을 특징으로 하는 교착 상태 방지를 위한 버스 유닛.

청구항 3

제2항에 있어서, 상기 다수의 큐 엔트리중 먼저 처리해야 할 것을 나타내기 위해 쉬프트 체인을 이용하는 것을 특징으로 하는 교착 상태 방지를 위한 버스 유닛.

청구항 4

제1항에 있어서, 상기 사이클 큐(12)는, 사이클을 진행시키고 있다가 양보에 의해서 사이클이 정지한 경우, 다시 진행시킬 사이클이 있다는 것을 나타내기 위해서 할트(halt) 부분을 갖는 것을 특징으로 하는 교착 상태 방지를 위한 버스 유닛.

청구항 5

제4항에 있어서, 상기 할트 부분이 '1'인 동안에는 새로운 사이클을 받아들이지 않는 것을 특징으로 하는 교착 상태 방지를 위한 버스 유닛.

청구항 6

제1항에 있어서, 상기 사이클 큐(12)는, 사이클 주소를 저장하여 사이클의 재시작 및 바이트 인에이블 생성에 이용하는 교착 상태 방지를 위한 버스 유닛.

청구항 7

제1항에 있어서, 상기 사이클 큐(12)는, 실제 전송이 일어나고 있는 오퍼랜드의 크기를 나타내는 부분을 갖고 있는 것을 특징으로 하는 교착 상태 방지를 위한 버스 유닛.

청구항 8

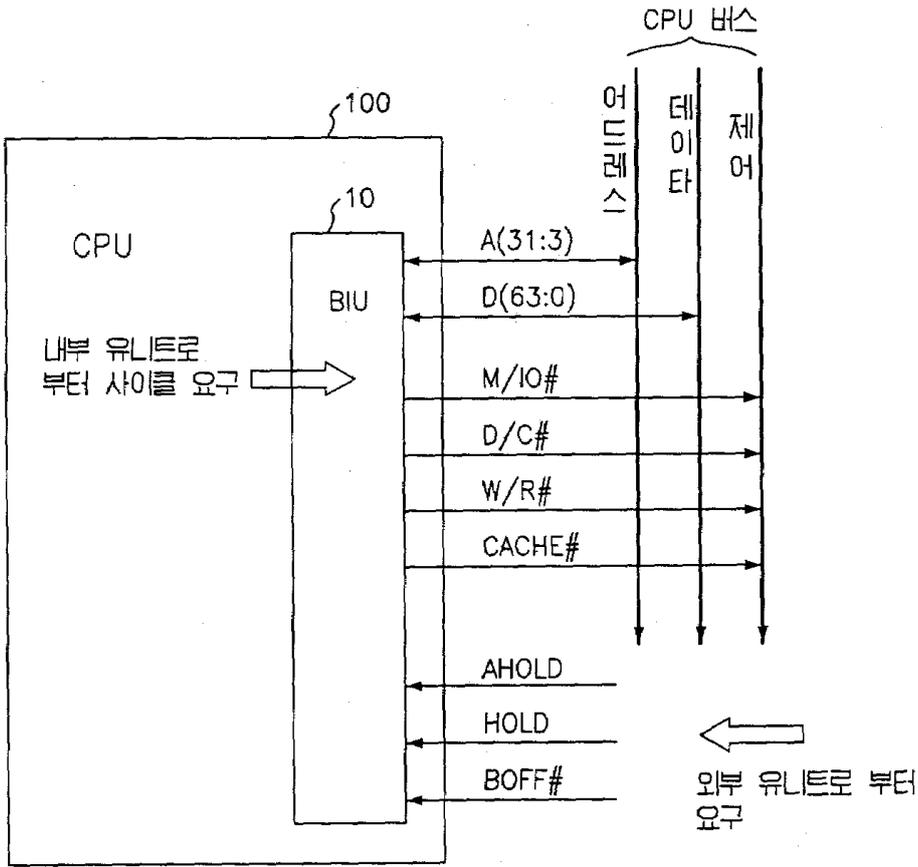
제7항에 있어서, 상기 사이클 큐(12)는, 진행되고 있는 사이클의 종류를 나타내기 위한 부분을 갖고 있는 것을 특징으로 하는 교착 상태 방지를 위한 버스 유닛.

청구항 9

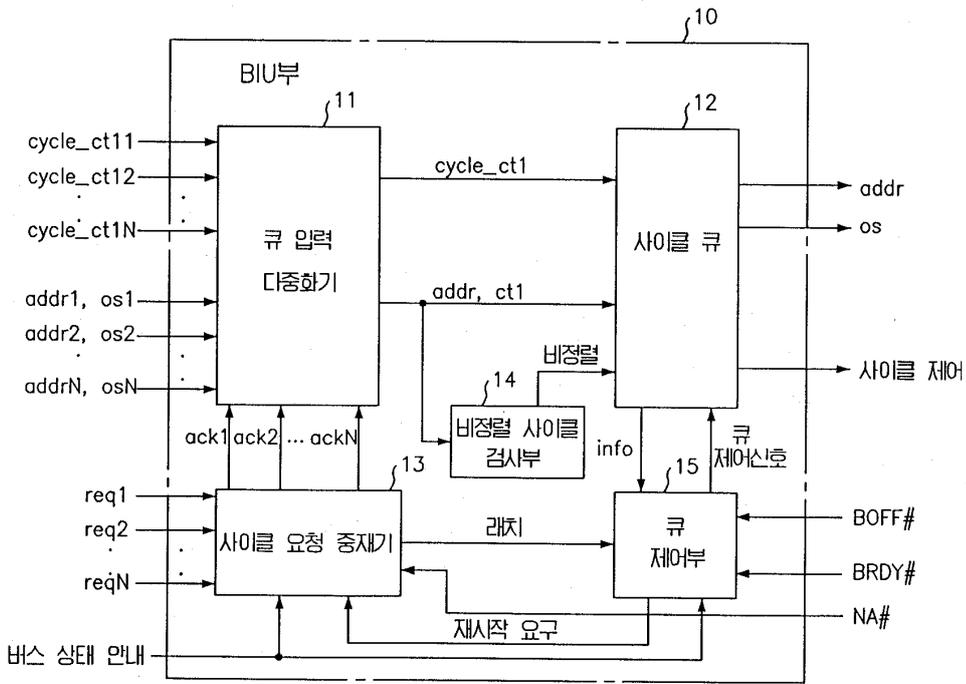
제1항에 있어서, 상기 사이클 큐(12)는, 재 사이클 큐의 엔트리에 유효한 사이클 정보가 저장되어 있다는 것을 나타내기 위해 유효비트를 갖고 있는 것을 특징으로 하는 교착상태 방지를 위한 버스 유닛.

도면

도면1



도면2



도면3

