



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년03월09일
(11) 등록번호 10-0887526
(24) 등록일자 2009년03월02일

(51) Int. Cl.

G06F 13/20 (2006.01) G06F 13/28 (2006.01)

G06F 13/16 (2006.01)

(21) 출원번호 10-2006-7001578

(22) 출원일자 2006년01월23일

심사청구일자 2007년03월28일

번역문제출일자 2006년01월23일

(65) 공개번호 10-2006-0052870

(43) 공개일자 2006년05월19일

(86) 국제출원번호 PCT/US2004/021993

국제출원일자 2004년07월08일

(87) 국제공개번호 WO 2005/010670

국제공개일자 2005년02월03일

(30) 우선권주장

10/625,132 2003년07월22일 미국(US)

(56) 선행기술조사문헌

US6219725 B1*

*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 42 항

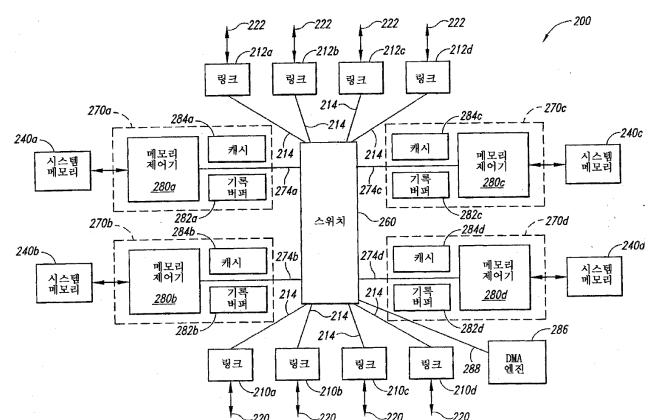
심사관 : 배경환

(54) 허브 기반 메모리 시스템에서 직접 메모리 액세스를 위한 장치 및 방법

(57) 요약

메모리 모듈을 위한 메모리 허브는 시스템 메모리에서 DMA 동작들을 수행하기 위해 DMA 엔진을 갖는다. 상기 메모리 허브는 상기 시스템 메모리의 상기 메모리 디바이스들 중 적어도 하나를 액세스하기 위해 메모리 요청들을 수신하는 링크 인터페이스를 포함하고, 상기 메모리 디바이스들에 연결하기 위해 메모리 디바이스 인터페이스를 더 포함하며, 상기 메모리 디바이스 인터페이스는 상기 메모리 디바이스들 중 적어도 하나를 액세스하기 위해 상기 메모리 디바이스들에 메모리 요청들을 연결한다. 상기 링크 인터페이스 및 상기 메모리 디바이스 인터페이스를 선택적으로 연결하는 스위치는 상기 메모리 허브에 또한 포함된다. 추가적으로, 직접 메모리 액세스(DMA) 엔진은 DMA 동작들을 수행하기 위해 상기 메모리 디바이스들 중 적어도 하나에 액세스하기 위해 메모리 요청들을 발생시키도록 상기 메모리 디바이스 인터페이스에 상기 스위치를 통해 연결된다.

대표 도



특허청구의 범위

청구항 1

복수의 메모리 디바이스들 및 메모리 허브를 포함하는 메모리 모듈에 있어서,

상기 메모리 허브는:

상기 메모리 디바이스들 중 적어도 하나로의 액세스를 위한 메모리 요청들을 수신하는 링크 인터페이스;

상기 메모리 디바이스들에 연결되고, 상기 메모리 디바이스들 중 적어도 하나로의 액세스를 위해 상기 메모리 디바이스들에 메모리 요청들을 연결하는, 메모리 디바이스 인터페이스;

상기 링크 인터페이스 및 상기 메모리 디바이스 인터페이스를 선택적으로 연결하는 스위치; 및

상기 스위치를 통해 상기 메모리 디바이스 인터페이스에 연결되는 직접 메모리 액세스(DMA: direct memory access) 엔진으로서, DMA 동작들을 수행하기 위해 상기 메모리 디바이스들 중 적어도 하나로의 액세스를 위한 메모리 요청들을 생성하는 상기 직접 메모리 액세스(DMA) 엔진을 포함하는, 메모리 모듈.

청구항 2

제 1 항에 있어서,

상기 메모리 허브는 단일 디바이스에 존재하는 상기 링크 인터페이스, 상기 메모리 디바이스 인터페이스, 상기 스위치, 및 상기 DMA 엔진을 갖는 임베딩된 시스템인, 메모리 모듈.

청구항 3

제 1 항에 있어서,

상기 메모리 디바이스 인터페이스는:

메모리 제어기 버스를 통해 상기 스위치에 연결되고 메모리 디바이스 버스를 통해 상기 메모리 디바이스들에도 연결되는 메모리 제어기;

상기 메모리 제어기에 연결되어 상기 메모리 제어기가 연결되는 상기 메모리 디바이스들 중 적어도 하나로 보내지는 메모리 요청들을 저장하는 기록 버퍼;

상기 메모리 제어기에 연결되어 상기 메모리 디바이스들에 제공되거나 상기 메모리 디바이스들로부터 검색되는 데이터를 저장하는 캐시를 포함하는, 메모리 모듈.

청구항 4

제 1 항에 있어서,

상기 스위치는 크로스 바 스위치(cross-bar switch)를 포함하는, 메모리 모듈.

청구항 5

제 1 항에 있어서,

상기 복수의 메모리 디바이스들은 메모리 동작 동안 동시에 액세스되는 메모리 디바이스들의 뱅크(bank)인, 메모리 모듈.

청구항 6

제 1 항에 있어서,

상기 복수의 메모리 디바이스들은 동기식 동적 랜덤 액세스 메모리 디바이스들을 포함하는, 메모리 모듈.

청구항 7

제 1 항에 있어서,

상기 DMA 엔진은:

DMA 동작을 위해 시작 메모리 어드레스를 저장하는 어드레스 레지스터;

데이터가 상기 DMA 동작 시 이동될 수 있는 위치의 타겟 어드레스를 저장하는 타겟 어드레스 위치;

상기 DMA 동작 시 액세스될 메모리 위치들의 수를 표시하는 카운트 값을 저장하는 카운트 레지스터;

상기 DMA 동작의 완료를 표현하거나 링크 리스트에 대응하는 메모리 어드레스를 표현하는 값을 저장하는 다음 레지스터로서, 상기 링크 리스트는 상기 어드레스 레지스터, 상기 카운트 레지스터, 및 상기 다음 레지스터로로딩될 시작 메모리 어드레스, 카운트 값, 및 다음 메모리 어드레스를 포함하는, 상기 다음 레지스터를 포함하는, 메모리 모듈.

청구항 8

복수의 메모리 디바이스들을 갖는 메모리 모듈을 위한 메모리 허브에 있어서,

상기 메모리 디바이스들 중 적어도 하나를 액세스하기 위해 메모리 요청들을 수신하는 링크 인터페이스;

상기 메모리 디바이스들에 연결하고, 상기 메모리 디바이스들 중 적어도 하나를 액세스하기 위해 상기 메모리 디바이스들에 메모리 요청들을 연결하는 메모리 디바이스 인터페이스;

상기 링크 인터페이스 및 상기 메모리 디바이스 인터페이스를 선택적으로 연결하는 스위치; 및

상기 스위치를 통해 상기 메모리 디바이스 인터페이스에 연결되는 직접 메모리 액세스(DMA) 엔진으로서, DMA 동작들을 수행하기 위해 상기 메모리 디바이스들 중 적어도 하나에 액세스하기 위해 메모리 요청들을 생성하는 상기 직접 메모리 액세스(DMA) 엔진을 포함하는, 메모리 허브.

청구항 9

제 8 항에 있어서,

상기 링크 인터페이스, 상기 메모리 디바이스 인터페이스, 상기 스위치, 및 상기 DMA 엔진은 단일 디바이스에 존재하는 임베딩된 시스템들인, 메모리 허브.

청구항 10

제 8 항에 있어서,

상기 메모리 디바이스 인터페이스는:

메모리 제어기 버스를 통해 상기 스위치에 연결되고 또한 메모리 디바이스 버스를 통해 상기 메모리 디바이스들로 연결되는 메모리 제어기;

상기 메모리 제어기에 연결되어 상기 메모리 제어기가 연결되는 상기 메모리 디바이스들 중 적어도 하나로 보내지는 메모리 요청들을 저장하는 기록 버퍼;

상기 메모리 제어기에 연결되어 상기 메모리 디바이스들에 제공되거나 상기 메모리 디바이스들로부터 검색되는 데이터를 저장하는 캐시를 포함하는, 메모리 허브.

청구항 11

제 8 항에 있어서,

상기 스위치는 크로스 바 스위치를 포함하는, 메모리 허브.

청구항 12

제 8 항에 있어서,

상기 DMA 엔진은:

DMA 동작을 위해 시작 메모리 어드레스를 저장하는 어드레스 레지스터;

데이터가 상기 DMA 동작 시 이동될 수 있는 위치의 타겟 어드레스를 저장하는 타겟 어드레스 위치;

상기 DMA 동작 시 액세스될 메모리 위치들의 수를 표시하는 카운트 값을 저장하는 카운트 레지스터; 및 상기 DMA 동작의 완료를 표현하거나 링크 리스트에 대응하는 메모리 어드레스를 표현하는 값을 저장하는 다음 레지스터로서, 상기 링크 리스트는 상기 어드레스 레지스터, 상기 카운트 레지스터, 및 상기 다음 레지스터로 로딩될 시작 메모리 어드레스, 카운트 값, 및 다음 메모리 어드레스를 포함하는, 상기 다음 레지스터를 포함하는, 메모리 허브.

청구항 13

메모리 요청들이 제공되는 메모리 버스와, 상기 메모리 버스에 연결되는 적어도 하나의 메모리 모듈로서, 상기 메모리 모듈은 복수의 메모리 디바이스들 및 메모리 허브를 갖는, 상기 적어도 하나의 메모리 모듈을 포함하는, 메모리 시스템에 있어서,

상기 메모리 허브는:

링크 인터페이스가 위치되는 상기 메모리 모듈의 상기 메모리 디바이스들 중 적어도 하나에 액세스하기 위한 메모리 요청들을 수신하도록 연결되는, 상기 링크 인터페이스;

상기 메모리 디바이스들에 연결되고, 상기 메모리 디바이스들 중 적어도 하나에 액세스하기 위해 상기 메모리 디바이스들에 메모리 요청들을 연결하는, 메모리 디바이스 인터페이스;

상기 링크 인터페이스 및 상기 메모리 디바이스 인터페이스를 선택적으로 연결하는 스위치; 및

상기 스위치를 통해 상기 메모리 디바이스 인터페이스 및 상기 링크 인터페이스에 연결되는 직접 메모리 액세스 (DMA) 엔진으로서, DMA 동작들을 수행하기 위해 상기 메모리 디바이스들 중 적어도 하나에 액세스하기 위한 메모리 요청들을 생성하는 상기 직접 메모리 액세스(DMA) 엔진을 포함하는, 메모리 시스템.

청구항 14

제 13 항에 있어서,

상기 메모리 허브는 단일 디바이스에 존재하는 상기 링크 인터페이스, 상기 메모리 디바이스 인터페이스, 상기 스위치, 및 상기 DMA 엔진을 갖는 임베딩된 시스템인, 메모리 시스템.

청구항 15

제 13 항에 있어서,

상기 메모리 버스는 고속 메모리 버스를 포함하는, 메모리 시스템.

청구항 16

제 13 항에 있어서,

상기 메모리 버스는 고속 광 메모리 버스를 포함하고, 상기 링크 인터페이스는 광 신호들 및 전기 신호들을 변환하는 광 메모리 버스 인터페이스 회로를 포함하는, 메모리 시스템.

청구항 17

제 13 항에 있어서,

복수의 메모리 모듈들은 상기 메모리 시스템에 포함되고, 상기 복수의 메모리 모듈들의 제 1 메모리 모듈은 상기 메모리 버스에 연결되고, 상기 복수의 메모리 모듈들 중 나머지 메모리 모듈들은 상기 제 1 메모리 모듈과 직렬로 연결되는, 메모리 시스템.

청구항 18

제 13 항에 있어서,

복수의 메모리 모듈들은 상기 메모리 시스템에 포함되고, 상기 복수의 메모리 모듈들 각각은 각각의 링크 인터페이스를 통해 상기 메모리 버스에 직접 연결되는, 메모리 시스템.

청구항 19

제 13 항에 있어서,

상기 메모리 허브의 상기 메모리 디바이스 인터페이스는:

메모리 제어기 버스를 통해 상기 스위치에 연결되고 메모리 디바이스 버스를 통해 상기 메모리 디바이스들에도 연결되는 메모리 제어기;

상기 메모리 제어기에 연결되어 상기 메모리 제어기가 연결되는 상기 메모리 디바이스들 중 적어도 하나로 보내지는 메모리 요청들을 저장하는 기록 버퍼; 및

상기 메모리 제어기에 연결되어 상기 메모리 디바이스들에 제공되거나 상기 메모리 디바이스들로부터 검색되는 데이터를 저장하는 캐시를 포함하는, 메모리 시스템.

청구항 20

제 13 항에 있어서,

상기 메모리 허브의 상기 스위치는 크로스 바 스위치를 포함하는, 메모리 시스템.

청구항 21

제 13 항에 있어서,

메모리 모듈의 상기 복수의 메모리 디바이스들은 메모리 동작 동안 동시에 액세스되는 메모리 디바이스들의 뱅크를 표현하는, 메모리 시스템.

청구항 22

제 13 항에 있어서,

상기 메모리 모듈들의 상기 복수의 메모리 디바이스들은 동기식 동적 랜덤 액세스 메모리 디바이스들을 포함하는, 메모리 시스템.

청구항 23

제 13 항에 있어서,

상기 메모리 허브의 상기 DMA 엔진은:

DMA 동작이 시작하는 상기 메모리 시스템에서 메모리 위치의 시작 메모리 어드레스를 저장하는 어드레스 레지스터;

데이터가 상기 DMA 동작 시 이동될 수 있는 상기 메모리 시스템에 메모리 위치의 타겟 어드레스를 저장하는 타겟 어드레스 위치;

상기 DMA 동작 시 액세스될 메모리 위치들의 수를 표시하는 카운트 값을 저장하는 카운트 레지스터; 및

상기 DMA 동작의 완료를 표현하거나 링크 리스트에 대응하는 메모리 어드레스를 표현하는 값을 저장하는 다음 레지스터로서, 상기 링크 리스트는 상기 어드레스 레지스터, 상기 카운트 레지스터, 및 상기 다음 레지스터로로딩될 시작 메모리 어드레스, 카운트 값, 및 다음 메모리 어드레스를 포함하는, 상기 다음 레지스터를 포함하는, 메모리 시스템.

청구항 24

컴퓨터 시스템에 있어서,

중앙 처리 유닛("CPU");

상기 CPU에 연결되고, 입력 포트 및 출력 포트를 갖는 시스템 제어기;

상기 시스템 제어기를 통해 상기 CPU에 연결되는 입력 디바이스;

상기 시스템 제어기를 통해 상기 CPU에 연결되는 출력 디바이스;

상기 시스템 제어기를 통해 상기 CPU에 연결되는 저장 디바이스;

적어도 하나의 메모리 모듈을 포함하고,

상기 메모리 모듈은 복수의 메모리 디바이스들 및 메모리 허브를 포함하고,

상기 메모리 허브는:

링크 인터페이스가 위치되는 상기 메모리 모듈의 상기 메모리 디바이스들 중 적어도 하나에 액세스하기 위한 메모리 요청들을 수신하도록 연결되는, 상기 링크 인터페이스;

상기 메모리 디바이스들에 연결되고, 상기 메모리 디바이스들 중 적어도 하나로의 액세스를 위해 상기 메모리 디바이스들에 메모리 요청들을 연결하는, 메모리 디바이스 인터페이스;

상기 링크 인터페이스 및 상기 메모리 디바이스 인터페이스를 선택적으로 연결하는 스위치; 및

상기 스위치를 통해 상기 메모리 디바이스 인터페이스 및 상기 링크 인터페이스에 연결되고 DMA 동작들을 수행하기 위해 상기 복수의 메모리 모듈들의 상기 메모리 디바이스들 중 적어도 하나로의 액세스를 위한 메모리 요청들을 생성하는 직접 메모리 액세스(DMA) 엔진을 포함하는, 상기 적어도 하나의 메모리 모듈; 및

상기 시스템 제어기 및 상기 메모리 모듈들 사이의 메모리 요청들 및 데이터를 연결하는 상기 복수의 메모리 모듈들 중 적어도 하나 및 상기 시스템 제어기 사이에 연결되는 통신 링크를 포함하는, 컴퓨터 시스템.

청구항 25

제 24 항에 있어서,

상기 통신 링크는 고속 메모리 버스를 포함하는, 컴퓨터 시스템.

청구항 26

제 24 항에 있어서,

상기 메모리 허브는 단일 디바이스에 존재하는 상기 링크 인터페이스, 상기 메모리 디바이스 인터페이스, 상기 스위치, 및 상기 DMA 엔진을 갖는 임베딩된 시스템인, 컴퓨터 시스템.

청구항 27

제 24 항에 있어서,

상기 통신 링크는 고속 광 메모리 버스를 포함하고, 상기 메모리 허브의 상기 링크 인터페이스는 광 신호들 및 전기 신호들을 변환하는 광 메모리 버스 인터페이스 회로를 포함하는, 컴퓨터 시스템.

청구항 28

제 24 항에 있어서,

복수의 메모리 모듈들은 상기 컴퓨터 시스템에 포함되고, 상기 복수의 메모리 모듈들의 제 1 메모리 모듈은 상기 통신 링크에 연결되고, 상기 복수의 메모리 모듈들 중 나머지 메모리 모듈들은 상기 제 1 메모리 모듈과 직렬로 연결되는, 컴퓨터 시스템.

청구항 29

제 24 항에 있어서,

복수의 메모리 모듈들은 상기 컴퓨터 시스템에 포함되고, 상기 복수의 메모리 모듈들 각각은 각각의 링크 인터페이스를 통해 상기 메모리 버스에 직접 연결되는, 컴퓨터 시스템.

청구항 30

제 24 항에 있어서,

상기 메모리 허브의 상기 메모리 디바이스 인터페이스는:

메모리 제어기 버스를 통해 상기 스위치에 연결되고 메모리 디바이스 버스를 통해 상기 메모리 디바이스들에도 연결되는 메모리 제어기;

상기 메모리 제어기에 연결되어 상기 메모리 제어기가 연결되는 상기 메모리 디바이스들 중 적어도 하나로 보내지는 메모리 요청들을 저장하는 기록 버퍼; 및

상기 메모리 제어기에 연결되어 상기 메모리 디바이스들에 제공되거나 상기 메모리 디바이스들로부터 검색되는 데이터를 저장하는 캐시를 포함하는, 컴퓨터 시스템.

청구항 31

제 24 항에 있어서,

상기 메모리 허브의 상기 스위치는 크로스 바 스위치를 포함하는, 컴퓨터 시스템.

청구항 32

제 24 항에 있어서,

메모리 모듈의 상기 복수의 메모리 디바이스들은 메모리 동작 동안 동시에 액세스되는 메모리 디바이스들의 뱅크를 나타내는, 컴퓨터 시스템.

청구항 33

제 24 항에 있어서,

상기 메모리 모듈의 상기 복수의 메모리 디바이스들은 동기식 동적 랜덤 액세스 메모리 디바이스들을 포함하는, 컴퓨터 시스템.

청구항 34

제 24 항에 있어서,

상기 메모리 허브의 상기 DMA 엔진은:

DMA 동작이 시작하는 상기 메모리 시스템에서 메모리 위치의 시작 메모리 어드레스를 저장하는 어드레스 레지스터;

데이터가 상기 DMA 동작 시 이동될 수 있는 상기 메모리 시스템에 메모리 위치의 타겟 어드레스를 저장하는 타겟 어드레스 위치;

상기 DMA 동작 시 액세스될 메모리 위치들의 수를 표시하는 카운트 값을 저장하는 카운트 레지스터; 및

상기 DMA 동작의 완료를 나타내거나 링크 리스트에 대응하는 메모리 어드레스를 나타내는 값을 저장하는 다음 레지스터로서, 상기 링크 리스트는 상기 어드레스 레지스터, 상기 카운트 레지스터, 및 상기 다음 레지스터로로 딩될 시작 메모리 어드레스, 카운트 값, 및 다음 메모리 어드레스를 포함하는, 상기 다음 레지스터를 포함하는, 컴퓨터 시스템.

청구항 35

프로세서와, 상기 프로세서에 연결되는 시스템 제어기와, 메모리 버스를 통해 상기 시스템 제어기에 연결되는 적어도 하나의 메모리 모듈을 갖는 시스템 메모리를 갖는 컴퓨터 시스템에서 메모리 동작들을 실행하는 방법에 있어서,

프로세서 개입 없이 상기 시스템 메모리에서 메모리 동작들을 실행하는 명령들을 표현하는 상기 시스템 메모리 내의 위치에 직접 메모리 액세스(direct memory access, DMA) 정보를 기록하는 단계;

상기 프로세서 및 시스템 제어기로부터 상기 메모리 버스의 제어를 획득하는 단계; 및

상기 메모리 모듈에 위치된 DMA 제어기를 이용하여, 상기 DMA 정보가 기록되는 상기 시스템 메모리 내의 상기 위치를 액세스하는 단계; 및 상기 메모리 모듈에 위치된 상기 DMA 제어기를 이용하는 상기 명령들에 의해 표현

되는 상기 메모리 동작들의 실행을 조종하는 단계를 포함하는, 메모리 동작 실행 방법.

청구항 36

제 35 항에 있어서, 상기 메모리 동작들의 실행 동안 상기 시스템 메모리를 분리시키는 단계를 더 포함하는, 메모리 동작 실행 방법.

청구항 37

제 35 항에 있어서,

DMA 정보를 기록하는 단계는:

상기 메모리 동작들이 시작하는 상기 시스템 메모리 내의 메모리 위치의 시작 메모리 어드레스를 기록하는 단계;

데이터가 상기 메모리 동작들에서 이동될 수 있는 상기 시스템 메모리 내의 메모리 위치의 타겟 어드레스를 기록하는 단계;

상기 메모리 동작들에서 액세스될 메모리 위치들의 수를 표시하는 카운트 값을 기록하는 단계; 및

상기 메모리 동작들의 종료를 나타내거나, 시작 메모리 어드레스, 카운트 값, 및 다음 메모리 어드레스 값을 포함하는 링크 리스트에 대응하는 메모리 어드레스를 나타내는 다음 메모리 어드레스 값을 기록하는 단계를 포함하는, 메모리 동작 실행 방법.

청구항 38

제 35 항에 있어서,

상기 시스템 메모리는 복수의 메모리 모듈들을 포함하고, 상기 메모리 동작들을 실행하는 단계는 상기 복수의 메모리 모듈들 중 제 1 메모리 모듈에서 메모리 위치를 액세스하여 그로부터 데이터를 판독하는 단계 및 상기 복수의 메모리 모듈들 중 제 2 메모리 모듈에서 메모리 위치를 액세스하여 상기 데이터를 기록하는 단계를 포함하는, 메모리 동작 실행 방법.

청구항 39

프로세서와, 상기 프로세서에 연결되는 시스템 제어기와, 상기 시스템 메모리에 상기 시스템 제어기를 연결하는 메모리 버스를 갖는 컴퓨터 시스템에 포함되는 시스템 메모리 내로 데이터를 전송하는 방법에 있어서,

상기 시스템 메모리 내의 위치에 DMA 명령들을 기록하는 단계로서, 상기 DMA 명령들은 상기 시스템 메모리 내의 제 1 및 제 2 위치들에 대응하는 메모리 어드레스들을 포함하는 상기 데이터를 전송하기 위해 메모리 동작들을 실행하는 명령들을 표현하는, 상기 DMA 명령들을 기록하는 단계;

상기 메모리 버스의 제어를 획득하는 단계; 및

프로세서 및 시스템 제어기 개입 없이, 상기 시스템 메모리의 메모리 모듈에 위치된 DMA 제어기를 이용하여, 상기 DMA 명령들이 기록되는 상기 시스템 메모리 내의 상기 위치를 액세스하는 단계, 상기 시스템 메모리 내의 상기 제 1 위치로부터 데이터를 판독하는 단계, 및 상기 시스템 메모리 내의 상기 제 2 위치에 상기 데이터를 기록하는 단계를 포함하는, 데이터 전송 방법.

청구항 40

제 39 항에 있어서,

상기 메모리 버스의 제어를 획득하는 단계는 상기 시스템 메모리 내에 데이터를 전송하는 동안 상기 프로세서 및 시스템 제어기로부터 상기 시스템 메모리를 분리시키는 단계를 포함하는, 데이터 전송 방법.

청구항 41

제 39 항에 있어서,

DMA 명령들을 기록하는 단계는:

상기 데이터의 전송이 시작되는 상기 시스템 메모리 내의 메모리 위치의 시작 메모리 어드레스를 기록하는 단계;

상기 데이터가 전송될 상기 시스템 메모리 내의 메모리 위치의 타겟 어드레스를 기록하는 단계;

상기 데이터의 전송 시 액세스될 메모리 위치들의 수를 표시하는 카운트 값을 기록하는 단계; 및

상기 데이터 전송의 완료를 나타내거나, 시작 메모리 어드레스, 카운트 값, 및 다음 메모리 어드레스 값을 포함하는 링크 리스트에 대응하는 메모리 어드레스를 나타내는 다음 메모리 어드레스 값을 기록하는 단계를 포함하는, 데이터 전송 방법.

청구항 42

제 39 항에 있어서,

상기 시스템 메모리는 복수의 메모리 모듈들을 포함하고, 상기 시스템 메모리 내의 상기 제 1 위치로부터 데이터를 관독하는 단계는 상기 복수의 메모리 모듈들 중 제 1 메모리 모듈 내의 메모리 위치를 액세스하여 그로부터 데이터를 관독하는 단계를 포함하고, 상기 시스템 메모리 내의 상기 제 2 위치에 상기 데이터를 기록하는 단계는 상기 복수의 메모리 모듈들 중 제 2 메모리 모듈 내의 메모리 위치를 액세스하여 상기 데이터를 기록하는 단계를 포함하는, 데이터 전송 방법.

명세서

기술 분야

<1> 본 발명은 컴퓨터 시스템들, 보다 자세하게는 메모리 허브 구조를 갖는 시스템 메모리를 포함하는 컴퓨터 시스템에 관한 것이다.

배경 기술

<2> 컴퓨터 시스템들은 프로세서에 의해 액세스되는 데이터를 저장하기 위해 동적 랜덤 액세스 메모리("DRAM")와 같은 메모리 디바이스들을 이용한다. 이러한 메모리 디바이스들은 보통 컴퓨터 시스템에서 시스템 메모리로 이용된다. 전형적인 컴퓨터 시스템에 있어서, 프로세서는 프로세서 버스 및 메모리 제어기를 통해 시스템 메모리와 통신한다. 프로세서는 관독 명령과 같은 메모리 명령을 포함하는 메모리 요청과, 데이터 또는 명령들이 관독될 수 있는 위치를 지정하는 어드레스를 발행한다(issue). 메모리 제어기는 시스템 메모리에 적용되는 행 및 열 어드레스들 뿐만 아니라 적절한 명령 신호들을 생성하기 위해 명령 및 어드레스를 이용한다. 명령들 및 어드레스들에 응답하여, 데이터는 시스템 메모리 및 프로세서 간에 전송된다. 메모리 제어기는 종종 PCI 버스와 같은 확장 버스에 프로세서 버스를 연결하는 버스 브리지 회로를 또한 포함하는 시스템 제어기의 일부이다.

<3> 메모리 디바이스들의 동작 속도가 연속적으로 증가할 지라도, 동작 속도에서 이러한 증가는 프로세서들의 동작 속도에서의 증가들과 함께 페이스를 유지하지는 않는다. 심지어 메모리 디바이스들에 프로세서들을 연결하는 메모리 제어기들의 동작 속도에서의 증가가 보다 느려진다. 메모리 제어기들 및 메모리 디바이스들의 상대적으로 느린 속도는 프로세서 및 메모리 디바이스들 사이의 데이터 대역폭을 제한한다.

<4> 프로세서들 및 메모리 디바이스들 사이의 제한된 대역폭 이외에, 컴퓨터 시스템들의 성능은 시스템 메모리 디바이스들로부터 데이터를 관독하기 위해 요구되는 시간을 증가시키는 레이턴시 문제(latency problem)들에 의해서도 제한된다. 보다 자세하게는, 메모리 디바이스 관독 명령이 동기식 DRAM("SDRAM") 디바이스와 같은 시스템 메모리 디바이스에 연결될 때, 관독 데이터는 단지 몇개의 클럭 기간들의 지연 후에 SDRAM 디바이스로부터 출력된다. 그러므로, SDRAM 디바이스들이 높은 데이터 레이트에서 버스트 데이터를 동기적으로 출력할 수 있을지라도, 상기 데이터를 초기에 제공하는 지연은 SDRAM 디바이스들과 같은 것들을 이용하여 컴퓨터 시스템의 동작 속도를 현저히 느리게 할 수 있다.

<5> 상기 메모리 레이턴시 문제를 경감시키는 한가지 접근 방법은 메모리 허브를 통해 프로세서에 연결된 다중 메모리 디바이스들을 이용하는 것이다. 메모리 허브 구조에 있어서, 시스템 제어기 또는 메모리 제어기는 고속 데이터 링크를 통해 몇개의 메모리 모듈들에 연결된다. 전형적으로, 상기 메모리 모듈들이 직렬로 서로 연결되는 것과 같은 포인트 투 포인트(point-to-point) 또는 데이터 체인(daisy-chain) 구조로 연결된다. 따라서, 상기 메모리 제어기는 제 2 고속 데이터 링크를 통해 제 2 메모리 모듈에 접속되는 제 1 메모리 모듈을 통해 제 1 고속

데이터 링크에 걸쳐 제 1 메모리 모듈에 연결되고, 제 2 메모리 모듈은 데이지 체인 방식으로 제 3 고속 데이터 링크를 통해 제 3 메모리 모듈에 연결된다.

<6> 각각의 메모리 모듈은 상기 고속 데이터 링크들에 걸쳐 상기 제어기 및 상기 메모리 디바이스들 사이의 응답들 및 메모리 요청들을 효과적으로 라우팅하는 상기 메모리 허브들을 통해 대응하는 고속 데이터 링크들 및 상기 모듈상의 다수의 메모리 디바이스들에 연결되는 메모리 허브를 포함한다. 이러한 구조를 채용하는 컴퓨터 시스템들은 또 다른 메모리 디바이스가 이전의 메모리 액세스에 응답하는 동안 프로세서가 하나의 메모리 디바이스에 액세스할 수 있기 때문에 보다 높은 대역폭을 가질 수 있다. 예를 들어, 상기 프로세서는 상기 시스템에서 또 다른 메모리 디바이스가 상기 프로세서에 판독 데이터를 제공하도록 준비하는 동안, 상기 시스템에서 상기 메모리 디바이스들 중 하나에 기록 데이터를 출력할 수 있다. 더욱이, 이러한 구조는 또한 종래의 다분기 버스 구조(multi-drop bus architecture)들에 있어서 발생하는 바와 같이, 보다 많은 메모리 모듈들이 추가됨에 따라 신호 품질에서의 저하에 대한 근심 없이 상기 시스템 메모리의 손쉬운 확장을 제공한다.

<7> 메모리 허브들을 이용하는 컴퓨터 시스템들이 우수한 성능을 제공할지라도, 그럼에도 불구하고 그것들은 종종 다양한 이유들로 최적 속도들에서 동작 실패를 할 수 있다. 예를 들어, 메모리 허브들이 보다 큰 메모리 대역폭을 갖는 컴퓨터 시스템들을 제공할 수 있을지라도, 그것들은 상술된 형태의 레이턴시 문제들로 여전히 손해를 입는다. 보다 자세하게는, 또 다른 메모리 디바이스가 데이터를 전송하기 위해 준비하는 동안 상기 프로세서가 하나의 메모리 디바이스와 통신할 수 있을지라도, 또 다른 메모리 디바이스로부터 데이터가 이용될 수 있기 전에 하나의 메모리 디바이스로부터 데이터를 때때로 수신할 필요가 있다. 그러한 경우에 있어서, 또 다른 메모리 디바이스로부터 수신된 데이터가 이용될 수 있기 전에 하나의 메모리 디바이스로부터 데이터가 수신되어야 한다. 상기 프로세서의 개입은 그러한 컴퓨터 시스템들의 동작 속도를 느리게 하기 위해 계속된다. 그 컴퓨터 시스템들이 최적 속도에서 동작 실패를 하는 이러한 이유들 중 또 다른 하나는 종래 메모리 허브들이 본질적으로 단일 채널 시스템들이라는 것이다. 이는 모든 제어, 어드레스, 및 데이터 신호들이 공통 메모리 허브 회로를 통해 통과해야 하기 때문이다. 결과적으로, 상기 메모리 허브 회로가 하나의 메모리 디바이스와 활발하게 통신하고 있을 때, 또 다른 메모리 디바이스와 통신하는 것은 자유롭지 않다.

<8> 단일 채널 병목뿐만 아니라 메모리로부터 메모리로 이동하는 데이터에서 프로세서 개입을 갖는 이슈들을 극복하기 위해 컴퓨터 시스템들에서 이용된 한가지 테크닉이 직접 메모리 액세스(DMA) 동작들의 이용이다. DMA 동작들은 상기 시스템 프로세서의 개입 없이 메모리로 그리고 메모리 외부로 이동되도록 데이터를 인에이블하는 상기 컴퓨터 시스템에 포함되는 DMA 제어기들의 이용을 통해 구현된다. 그러한 DMA 동작들 및 DMA 제어기들은 본 기술 분야에 알려져 있고, 종종 종래 컴퓨터 시스템들에서 구현된다. DMA 제어기는 상기 시스템 메모리로 그리고 메모리 외부로 요구된 데이터 전송들을 관리하고 상기 프로세서에 대해 포함되기 위한 필요성을 제거한다. 예를 들어, 엔티티가 지원되는 DMA가 상기 시스템 메모리에 데이터를 전송할 때, DMA 제어기는 상기 버스의 제어를 획득하고 상기 프로세서를 포함하지 않고 상기 시스템 메모리에 엔티티가 지원된 DMA로부터 데이터의 전송을 조정한다. 이러한 방식에 있어서, 프로세서 개입을 일으키는 레이턴시 이슈들은 데이터가 상기 시스템 버스를 통해 전송하는 동안 회피될 수 있다. 그러나, 많은 예들에 있어서, 심지어 데이터가 DMA 동작을 통해 상기 시스템 메모리에 전송된 후에도, 그럼에도 불구하고 상기 프로세서는 상기 시스템 내 하나의 위치로부터 또 다른 위치로 상기 데이터의 블록들을 이동시켜야 한다. 예를 들어, 운영체제는 상기 프로세서만을 갖도록 상기 시스템 메모리로 매스 저장 디바이스로부터 데이터를 전송하도록 DMA 동작을 지시할 것이고, 그에 따라 메모리에서 또 다른 위치로 상기 데이터를 이동시켜 상기 데이터가 이용될 수 있다. 결과적으로, DMA 동작들을 갖는 값은 상기 시스템 메모리로부터 그리고 상기 시스템 메모리로 상기 데이터 전송에서 DMA 동작의 이용에도 불구하고 상기 프로세서가 최종적으로 메모리 내 주위로 데이터를 이동시킴으로써 포함되기 때문에 다소 줄어든다.

<9> 그러므로, 메모리 허브 구조의 장점들을 제공하고 또한 그러한 시스템들에서 공통인 레이턴시 문제들을 최소화하는 컴퓨터 구조에 대한 필요성이 존재한다.

발명의 상세한 설명

<10> 본 발명은 시스템 메모리에서 DMA 동작들을 수행하는 DMA 엔진을 갖는 메모리 모듈에 대한 메모리 허브에 관한 것이다. 상기 메모리 허브는 상기 시스템 메모리의 메모리 디바이스들 중 적어도 하나로의 액세스를 위한 메모리 요청들을 수신하는 링크 인터페이스를 포함하고, 추가로 상기 메모리 디바이스들에 연결하기 위해 메모리 디바이스 인터페이스를 포함하며, 상기 메모리 디바이스 인터페이스는 상기 메모리 디바이스들 중 적어도 하나로의 액세스를 위해 메모리 디바이스들에 메모리 요청들을 연결한다. 선택적으로 링크 인터페이스 및 메모리 디바이스 인터페이스를 연결하는 스위치가 상기 메모리 허브에 추가로 포함된다. 추가적으로, 직접 메모리 액세스

(DMA)는 DMA 동작들을 수행하기 위해 상기 메모리 디바이스들 중 적어도 하나로 액세스를 위한 메모리 요청들을 생성하도록 상기 메모리 디바이스 인터페이스에 상기 스위치를 통해 연결된다.

<11> 본 발명의 한 측면에 따라, 프로세서와, 상기 프로세서에 연결되는 시스템 제어기와, 메모리 버스를 통해 상기 시스템 제어기에 연결되는 적어도 하나의 메모리 모듈을 갖는 시스템 메모리를 갖는 컴퓨터 시스템에서 메모리 동작들을 실행하기 위한 방법이 제공된다. 상기 방법은 프로세서 개입 없이 상기 시스템 메모리에서 메모리 동작들을 실행하기 위해 명령들을 표현하는 상기 시스템 메모리내 위치에 DMA 정보를 기록하는 단계와, 상기 프로세서 및 시스템 제어기로부터 상기 메모리 버스의 제어를 획득하는 단계와, DMA 정보가 기록된 상기 시스템 메모리에서 상기 위치를 액세스하는 단계와, 상기 명령들에 의해 표현된 상기 메모리 동작들을 실행하는 단계를 포함한다.

실시예

<17> 본 발명의 실시예들은 시스템 프로세서의 개입 없이 상기 시스템 메모리 내 데이터를 전송하기 위해 직접 메모리 액세스(DMA) 기능을 포함하는 메모리 허브 구조를 갖는 시스템 메모리에 관한 것이다. 특정 세부사항들이 본 발명의 충분한 이해를 제공하기 위해 이하 설명될 것이다. 그러나, 본 발명이 이러한 특정 세부사항들 없이 실행될 수 있다는 것은 본 기술 분야의 전문가들에게 명백할 것이다. 다른 예들에 있어서, 알려진 회로들, 제어 신호들, 및 타이밍 프로토콜들은 본 발명을 불필요하게 모호하지 않도록 하기 위해 자세히 제시되지는 않았다.

<18> 본 발명의 하나의 예에 따른 컴퓨터 시스템(100)은 도 1에 도시되어 있다. 컴퓨터 시스템(100)은 특정 계산들 또는 태스크들을 수행하기 위한 특정 소프트웨어를 실행하는 것과 같은, 다양한 계산 기능들을 수행하는 프로세서(104)를 포함한다. 프로세서(104)는 보통 어드레스 버스, 제어 버스, 및 데이터 버스를 포함하는 프로세서 버스(106)를 포함한다. 프로세서 버스(106)는 이전에 언급된 바와 같이 일반적으로 정적 웨이드 액세스 메모리 ("SRAM")인 캐시 메모리(108)에 전형적으로 연결된다. 마지막으로, 프로세서 버스(106)는 "노스 브리지(North Bridge)" 또는 "메모리 제어기(memory controller)"로 또한 때때로 언급되는 시스템 제어기(110)에 연결된다.

<19> 시스템 제어기(110)는 다양한 다른 구성요소들에 대해 프로세서(104)에서 통신 경로로서 작동한다. 보다 자세하게는, 시스템 제어기(110)는 차례로 비디오 터미널(114)에 연결되는 그래픽 제어기(112)에 전형적으로 연결되는 그래픽 포트를 포함한다. 시스템 제어기(110)는 오퍼레이터가 컴퓨터 시스템(100)과 인터페이스하도록 허용하기 위해 키보드 또는 마우스와 같은 하나 또는 그 이상의 입력 디바이스들(118)에 또한 연결된다. 전형적으로, 컴퓨터 시스템(100)은 시스템 제어기(110)를 통해 프로세서(104)에 연결되는 프린터와 같은 하나 또는 그 이상의 출력 디바이스들(120)을 포함한다. 하나 또는 그 이상의 데이터 저장 디바이스들(124)은 전형적으로 프로세서(104)가 내부 또는 외부 저장 매체(도시되지 않음)로부터 데이터를 저장 또는 검색하도록 허용하기 위해 시스템 제어기(110)를 통해 프로세서(104)에 또한 연결된다. 전형적인 저장 디바이스들(124)의 예들은 하드 및 플로피 디스크들, 테이프 카세트들, 및 커넥트 디스크 판독 전용 메모리들(CD-ROMs)을 포함한다.

<20> 시스템 제어기(110)는 컴퓨터 시스템(100)에 대해 시스템 메모리로 작동하는 몇개의 메모리 모듈들(130a, 130b, ... 130n)에 연결되는 메모리 허브 제어기(128)를 포함한다. 메모리 모듈들(130)은 양호하게는 광 또는 전기 통신 경로, 또는 다른 형태의 통신 경로들일 수 있는 고속 링크(134)를 통해 메모리 허브 제어기(128)에 연결된다. 고속 링크(134)가 광 통신 경로로 구현되는 경우에 있어서, 상기 광 통신 경로는 예를 들어 하나 또는 그 이상의 광 섬유들의 형태일 수 있다. 그러한 경우에 있어서, 메모리 허브 제어기(128) 및 메모리 모듈들은 광 입/출력 포트를 포함하거나 상기 광 통신 경로에 연결되는 입력 및 출력 포트들을 분리할 것이다.

<21> 메모리 모듈들(130)은 고속 링크(134)가 메모리 모듈들(130)의 메모리 허브들(140)과 함께 연결하는 단계로부터 형성되는 포인트 투 포인트 배치에서 메모리 허브 제어기(128)에 연결된 것으로 나타내고 있다. 즉, 고속 링크(134)는 직렬로 메모리 허브들(140)을 연결하는 양방향 버스이다. 따라서, 고속 링크(134) 상의 정보는 "다운스트림(downstream)" 목적지에 도달하기 위해 "업스트림(upstream)" 메모리 모듈들의 메모리 허브(140)를 통해 이동해야 한다. 예를 들어, 도 1과 특별히 관련하여, 메모리 허브 제어기(128)로부터 메모리 모듈(130c)의 메모리 허브(140)로 전달된 정보는 메모리 모듈들(130a, 130b)의 메모리 허브들(140)을 통해 통과할 것이다. 그러나, 메모리 모듈들(130)의 각각이 고속 링크에 걸쳐 메모리 허브 제어기(128)에 연결되는 연결 배치와 같은, 다른 토폴로지들이 또한 이용될 수 있다는 것을 이해할 것이다. 또한, 스위칭 토폴로지는 메모리 허브 제어기(128)가 스위치(도시되지 않음)를 통해 각각의 메모리 모듈들(130)에 선택적으로 연결되는 것으로 이용될 수 있다. 이용될 수 있는 다른 토폴로지들은 본 기술 분야의 전문가들에게 명백할 것이다.

<22> 도 1에 또한 도시된 바와 같이, 상기 메모리 허브는 각각의 버스 시스템(150)을 통해 메모리 디바이스들(148)의

4개의 세트들에 연결된다. 각각의 상기 세트들은 각각의 메모리 모듈(130)에 대한 전체 20개의 메모리 디바이스들(148)에 대해 4개의 메모리 디바이스들(148)을 포함한다. 버스 시스템들(150)은 보통 본 기술 분야에 알려진 바와 같이, 제어 버스, 어드레스 버스, 및 데이터 버스를 포함한다. 그러나, 본 기술 분야의 전문가들은 공유된 명령/어드레스 버스를 이용하는 버스 시스템과 같은 다른 버스 시스템들이 본 발명의 범위로부터 벗어나지 않으며 또한 이용될 수 있다는 것을 이해할 것이다. 추가로, 메모리 디바이스들(148)의 배치 및 메모리 디바이스(148)의 수가 본 발명의 범위로부터 벗어나지 않고 수정될 수 있다는 것은 명백할 것이다. 도 1에 예시된 예에 있어서, 메모리 디바이스들(148)은 동기식 동적 랜덤 액세스 메모리("SDRAM")이다. 그러나, SDRAM과는 다른 메모리 디바이스들이 또한 이용될 수 있다.

<23> 본 발명의 실시예에 따른 메모리 허브(200)의 실시예가 도 1의 메모리 허브(140)를 대신할 수 있는 도 2에 도시되어 있다. 메모리 허브(200)는 현재의 예에 있어서 종래의 SDRAM 디바이스들인 4개의 메모리 디바이스들(240a 내지 240d)에 연결되는 것으로 도 2에 도시되어 있다. 대안적인 실시예에 있어서, 메모리 허브(200)는 각각의 뱅크가 전형적으로 복수의 메모리 디바이스들을 갖는 오직 4개의 서로 다른 메모리 디바이스들(240a 내지 240d) 외에, 메모리 디바이스들의 4개의 서로 다른 뱅크들에 연결된다. 그러나, 제시되고 있는 예의 목적을 위해, 본 명세서에서는 4개의 메모리 디바이스들(240a 내지 240d)에 연결되는 메모리 허브(200)과 관련하여 기술될 것이다. 메모리의 다중 뱅크들을 수용하기 위해 메모리 허브(200)에서의 필요한 수정들이 본 기술 분야의 전문가들의 지식 내에 존재한다는 것을 이해할 것이다.

<24> 메모리 허브(200)에 포함되는 추가적인 것은 메모리 허브(200)가 제 1 고속 데이터 링크(220) 및 제 2 고속 데이터 링크(222) 각각에 위치되는 메모리 모듈에 연결하기 위한 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)이다. 도 1과 관련하여 이전에 논의된 바와 같이, 고속 데이터 링크들(220, 222)은 광 또는 전기 통신 경로, 또는 다른 형태의 통신 경로를 이용하여 구현될 수 있다. 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)은 종래의 것이고, 본 기술 분야에 알려진 예를 들어 송신기 및 수신기 로직과 같은 고속 데이터 링크들(220, 222)로부터 그리고 상기 링크들로 데이터, 명령, 및 어드레스 정보를 전송하도록 이용되는 회로를 포함한다. 본 기술 분야의 전문가들이 특정 형태의 통신 경로를 통해 이용되도록 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)을 수정하기 위해 충분한 이해를 가지고 있으며, 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)에서 그러한 수정들이 본 발명의 범위로부터 벗어나지 않고 만들어질 수 있다는 것은 명백할 것이다. 예를 들어, 고속 데이터 링크(220, 222)가 광 통신 경로를 이용하여 구현되는 경우에 있어서, 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)은 광 입력/출력 포트를 포함할 것이고 상기 광 통신 경로를 통해 연결된 광 신호들을 전기 신호들로 변환할 것이다.

<25> 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)은 버스들(214)에 의해 표현되는 복수의 버스 및 신호 라인들을 통해 스위치(260)에 연결된다. 단일 양방향 데이터 버스가 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)을 통해 양쪽 방향으로 데이터를 연결하도록 대안적으로 제공될 수 있을지라도, 버스들(214)은 종래의 것이고, 기록 데이터 버스 및 판독 데이터 버스를 포함한다. 본 기술 분야의 전문가들은 버스들(214)이 예의 방식으로 제공되고 버스들(214)이 캐시 일관성(cache coherency)을 유지하도록 이용될 수 있는 스누프 라인(snoop line) 및 요청 라인을 추가적으로 포함하는 것과 같이, 보다 적은 또는 보다 큰 신호 라인들을 포함할 수 있다는 것을 이해할 것이다.

<26> 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)은 메모리 허브(140)가 다양한 구성들에서 상기 시스템 메모리에 접속되도록 허용하는 회로를 포함한다. 예를 들어, 다분기 배열은 링크 인터페이스들(210a 내지 210d, 또는 212a 내지 212d) 둘 중 하나를 통해 메모리 허브 제어기(128)에 각각의 메모리 모듈을 연결함으로써 구현될 수 있다. 대안적으로, 포인트 투 포인트 또는 데이지 체인 구성은 도 1에 도시된 바와 같이, 직렬로 상기 메모리 모듈들을 연결함으로써 구현될 수 있다. 예를 들어, 링크 인터페이스들(210a 내지 210d)은 제 1 메모리 모듈에 연결하기 위해 이용될 수 있고, 링크 인터페이스들(212a 내지 212d)은 제 2 메모리 모듈에 연결하기 위해 이용될 수 있다. 프로세서 또는 시스템 제어기에 연결되는 상기 메모리 모듈은 상기 링크 인터페이스들의 하나의 세트를 통해 그에 연결될 것이고, 추가적으로 링크 인터페이스들이 다른 세트를 통해 또 다른 메모리 모듈에 연결될 것이다. 본 발명의 한가지 실시예에 있어서, 메모리 모듈의 메모리 허브(200)는 프로세서(104) 및 메모리 허브(200) 사이의 접속에 연결되는 다른 디바이스들이 존재하지 않는 포인트 투 포인트 배치로 상기 프로세서에 연결된다. 이러한 형태의 상호접속은 상대적으로 낮은 커뮤니케이션, 신호들을 반사하는 상대적으로 적은 라인 불연속, 및 상대적으로 짧은 신호 경로들을 포함하는 몇가지 이유들에 대해 프로세서(104) 및 메모리 허브(200) 사이의 신호 연결을 보다 양호하게 제공한다.

<27> 추가로, 스위치(260)는 차례로 시스템 메모리 디바이스들(240a 내지 240d)에 각각 연결되는 4개의 메모리 인터

페이스들(270a 내지 270d)에 연결된다. 각각의 시스템 메모리 디바이스(240a 내지 240d)에 대해 개별적 및 독립적 메모리 인터페이스(270a 내지 270d)를 각각 제공함으로써, 메모리 허브(200)는 전형적으로 단일 채널 메모리 구조들을 통해 발생하는 버스 또는 메모리 뱅크 충돌들을 회피한다. 스위치(260)는 버스들(274)에 의해 표현되는 복수의 버스 및 신호 라인들을 통해 각각의 메모리 인터페이스에 연결된다. 버스들(274)은 기록 데이터 버스, 판독 데이터 버스, 및 요청 라인을 포함한다. 그러나, 단일 양방향 데이터 버스가 개별적 기록 데이터 버스 및 판독 데이터 버스 대신에 대안적으로 이용될 수 있다는 것이 이해될 것이다. 더욱이, 버스들(274)은 이전에 기술된 것들 보다 더 많거나 더 적은 수의 신호 라인들을 포함할 수 있다.

<28> 본 발명의 실시예에 있어서, 각각의 메모리 인터페이스(270a 내지 270d)는 그것이 연결되는 시스템 메모리 디바이스들(240a 내지 240d)에 특별히 적응된다. 보다 자세하게는, 각각의 메모리 인터페이스(270a 내지 270d)는 그것이 연결되는 시스템 메모리 디바이스(240a 내지 240d)에 의해 각각 수신되고 발생된 특정 신호들을 수신하고 제공하기 위해 특별히 적응된다. 또한, 메모리 인터페이스들(270a 내지 270d)는 서로 다른 클럭 주파수들에서 동작하는 시스템 메모리 디바이스들(240a 내지 240d)과 함께 동작할 수 있다. 그 결과, 메모리 인터페이스들(270a 내지 270d)은 메모리 허브(200)에 연결되는 메모리 디바이스들(240a 내지 240d) 및 메모리 허브(230) 사이의 인터페이스에서 발생할 수 있는 변화들로부터 프로세서(104)를 격리시키고, 메모리 디바이스들(240a 내지 240d)이 인터페이싱할 수 있는 보다 제어된 환경을 제공한다.

<29> 링크 인터페이스들(210a 내지 210d, 212a 내지 212d) 및 메모리 인터페이스들(270a 내지 270d)를 연결하는 스위치(260)는 다양한 종래의 것 중 하나이거나, 이하 개발된 스위치들 일 수 있다. 예를 들어, 스위치(260)는 다양한 배치들에서 서로 동시에 링크 인터페이스들(210a 내지 210d, 212a 내지 212d) 및 메모리 인터페이스들(270a 내지 270d)을 연결할 수 있는 크로스 바(cross-bar) 스위치일 수 있다. 스위치(260)는 또한 크로스 바 스위치와 같이 연결성의 동일한 레벨을 제공하지는 않지만, 그럼에도 불구하고 각각의 메모리 인터페이스들(270a 내지 270d)에 링크 인터페이스들(210a 내지 210d, 212a 내지 212d)의 일부 또는 모두를 연결할 수 있는 멀티플렉서들의 세트일 수 있다. 스위치(260)는 또한 메모리 액세스들이 다른 메모리 액세스들에 걸쳐 우선권을 수신해야하는 중재 로직(도시되지 않음)을 포함할 수 있다. 이러한 기능을 수행하는 버스 중재는 본 기술 분야의 전문가들에게 알려져 있다.

<30> 도 2를 추가로 참조하면, 각각의 메모리 인터페이스들(270a 내지 270d)은 각각의 메모리 제어기(280), 각각의 기록 버퍼(282), 및 각각의 캐시 메모리 유닛(284)을 포함한다. 메모리 제어기(280)는 그것이 연결되는 시스템 메모리 디바이스(240a 내지 240d)에 제어, 어드레스, 및 데이터 신호들을 제공하고, 그것이 연결되는 시스템 메모리 디바이스(240a 내지 240d)로부터 데이터 신호들을 수신함으로써 종래의 메모리 제어기와 동일한 기능들을 수행한다. 기록 버퍼(282) 및 캐시 메모리 유닛(284)은 본 기술 분야에 알려진 바와 같은, 태그 메모리, 데이터 메모리, 비교기 등을 포함하는 버퍼 및 캐시 메모리의 일반적인 구성요소들을 포함한다. 기록 버퍼(282) 및 캐시 메모리 유닛(284)에서 이용되는 메모리 디바이스들은 DRAM 디바이스들, 정적 랜덤 액세스 메모리("SRAM") 디바이스들, 다른 형태들의 메모리 디바이스들, 또는 3개의 모든 조합일 수 있다. 추가로, 캐시 메모리 유닛(284)에서 이용되는 다른 구성요소들뿐만 아니라 이러한 메모리 디바이스들의 어느 것 또는 모든 것은 임베딩된 또는 자립형 디바이스들 둘 중 하나일 수 있다.

<31> 각각의 메모리 인터페이스(270a 내지 270d) 내의 기록 버퍼(282)는 판독 요청이 서비스되는 동안 기록 요청들을 저장하기 위해 이용될 수 있다. 그러한 시스템에 있어서, 프로세서(104)는 기록 요청이 지시되는 메모리 디바이스가 이전 기록 또는 판독 요청을 활발히 서비스하는 경우조차, 시스템 메모리 디바이스(240a 내지 240d)에 기록 요청을 내보낼 수 있다. 이러한 접근 방식을 이용하여, 메모리 요청들은 후속하는 판독 요청이 서비스되는 동안 보다 빠른 기록 요청이 기록 버퍼(282)에 저장될 수 있기 때문에, 잘못 작동될 수 있다. 판독 요청이 서비스되도록 허용하기 위해 기록 요청들을 버퍼링하기 위한 성능은 판독 요청들이 그들의 발생순 기록과 상관없이 첫번째 우선권이 주어질 수 있기 때문에, 메모리 판독 레이턴시를 매우 감소시킬 수 있다. 예를 들어, 판독 요청들과 함께 산재된 일련의 기록 요청들은 상기 판독 요청들이 파이프 라인 방식으로 저장된 기록 요청들을 서비스하기 이전에 파이프라인 방식으로 서비스되도록 허용하기 위해 기록 버퍼(282)에 저장될 수 있다. 그 결과, 기록 및 판독 요청들을 교대하기 위한 메모리 디바이스들(270a 내지 270d)에서의 연결 기록 요청 및 메모리 디바이스들(270a 내지 270d)에서의 후속하는 연결 판독 요청 사이의 긴 정착 시간들이 회피될 수 있다.

<32> 각각의 메모리 인터페이스(270a 내지 270d) 내의 캐시 메모리 유닛(284)의 이용은 데이터가 메모리 디바이스(240a 내지 240d)로부터 최근에 판독됐거나 상기 디바이스로 기록된 경우에서 그러한 데이터를 제공하기 위해 메모리 디바이스(240a 내지 240d)를 기다리지 않고 각각의 시스템 메모리 디바이스(240a 내지 240d)에 지시되는 판독 명령에 응하는 데이터를 프로세서(104)가 수신하도록 허용한다. 따라서, 캐시 메모리 유닛(284)은 상기 캡

퓨터 시스템의 메모리 대역폭을 최대화하기 위해 시스템 메모리 디바이스들(240a 내지 240d)의 판독 레이턴시를 감소시킨다. 유사하게는, 프로세서(104)는 캐시 메모리 유닛(284)에 기록 데이터를 저장할 수 있고, 그에 따라 동일한 메모리 인터페이스(270a 내지 270d)에서 메모리 제어기(280)가 캐시 메모리 유닛(284)으로부터 그것이 연결되는 시스템 메모리 디바이스(240a 내지 240d)로 상기 기록 데이터를 전송하는 동안 다른 기능들을 수행한다.

<33>

상기 시스템 메모리의 하나의 위치로부터 프로세서(104)로부터의 개입없는 상기 시스템 메모리의 또 다른 위치로 데이터의 블록들을 이동시키도록 메모리 허브(200)를 인에이블하는 버스(288)를 통해 스위치(260)에 연결되는 DMA 엔진(286)은 메모리 허브(200)에 추가로 포함된다. 버스(288)는 상기 시스템 메모리에서 데이터 전송들을 다루기 위해 어드레스, 제어, 데이터 버스들과 같은 복수의 신호 라인들 및 종래의 버스 라인들을 포함한다. 이하 상세히 기술되는 바와 같이, DMA 엔진(286)은 프로세서 개입없이 DMA 메모리 동작들을 실행하기 위해 상기 시스템 메모리에서 링크 리스트를 판독할 수 있고, 따라서 상기 메모리 동작들을 실행하는 단계로부터 대역폭 제한된 시스템 버스 및 프로세서(104)를 자유롭게 한다. 양호하게는, DMA 엔진(286)은 메모리 허브(200)에 임베딩된 회로이다. 그러나, 본 발명의 범위내에서는 메모리 허브(200)에 연결되는 개별적 DMA 디바이스가 포함된다. 추가적으로, DMA 엔진(286)은 다중 채널들 상에서 DMA 동작들을 수용하기 위한 회로를 포함할 수 있다. 그러한 다중 채널 DMA 엔진들은 본 기술 분야에 알려져 있고 종래의 테크놀로지들을 이용하여 구현될 수 있다.

<34>

본 발명의 실시예에 있어서, 프로세서(104)는 실행하기 위한 DMA 엔진(286)에 대한 시스템 메모리에서 명령들의 리스트를 기록한다. 상기 명령들은 이동하기 위한 블록의 시작 어드레스, 종료 어드레스 또는 카운트, 목적지 어드레스, 다음 명령 블록의 어드레스 등등과 같은 DMA 동작을 수행하기 위해 DMA 엔진(286)에 의해 이용되는 정보를 포함한다. DMA 엔진(286)은 일련의 연속적인 명령들을 수행하고, 그에 따라 그렇게 하도록 지시된 경우 다음 명령으로 점프한다. DMA 엔진(286)은 하나 또는 그 이상의 메모리 공간들에서 존재하는 데이터 구조를 통해 프로그램된다. 데이터 구조는 상기 시스템 메모리에서 데이터 전송 동작들을 수행하기 위해 필요한 정보를 제공하는 몇 개의 명령 블록들로 구성된다. 각각의 상기 명령 블록들은 링크된 리스트를 형성하기 위해 일련의 어드레스 포인터들을 통해 링크될 수 있다. 상기 링크된 리스트에서 제 1 명령 블록의 어드레스는 I/O 스페이스를 통해 프로그램된다. DMA 엔진(286)은 상기 I/O 스페이스 명령 레지스터를 통해 제 1 명령 블록을 폐치하고 실행하도록 지시된다. 요청된 데이터 동작을 수행한 후에, 제 1 명령 블록에서 어드레스 포인터는 DMA 엔진(286)을 다음 명령 블록에서 가리키도록 이용된다. 각각의 연속하는 명령 블록에서 어드레스 포인터는 링크된 리스트를 형성하는 다음 명령 블록을 폐치하고 실행하도록 이용된다. 상기 링크된 리스트에서 각각의 명령 블록은 NULL 포인터를 만나게 될 때까지 실행된다. NULL 포인터의 예는 모두 1로 구성되는 어드레스로서 정의된다. NULL 포인터를 검출하는 단계에 따라, 명령 블록 실행이 정지되고, 명령 스트림이 종결되었다는 것을 표시하는 상태 비트가 세팅될 것이다. 완료 상태는 메모리 허브(200)에서 I/O 레지스터에 포함될 수 있다. 추가적으로, 시작 플래그는 또한 DMA 엔진(286)이 이미 DMA 동작 실행을 시작했다는 것을 표시하도록 이용될 수 있다. 다른 상태 비트들은 상기 명령 스트림이 에러를 없이 정상적으로 종결된 경우 또는 에러들로 인해 비정상적으로 종결된 경우를 표시할 수 있다. 상태 정보는 선택적으로 호스트에서 인터럽트를 생성할 수 있다.

<35>

본 발명의 대안적인 실시예에 있어서, DMA 엔진(286)은 또한 상기 시스템에서 진단들을 운영하도록 이용될 수 있다. 알려진 양호한 데이터 패턴들로는 메모리 허브(200)의 메모리 또는 알려진 양호한 시스템 메모리에 로딩될 수 있고, 상기 시스템 메모리를 테스트하기 위해 이용된다. 애플리케이션의 이러한 형태의 보다 자세한 설명은 [출원일]에 출원되고, 참조문헌으로 본 명세서에 포함된, 명칭이 메모리 모듈들의 온 보드 진단들을 위한 시스템 및 방법(SYSTEM AND METHOD FOR ON-BOARD DIAGNOSTICS OF MEMORY MODULES)인, 현재 계류중인 미국 특허 번호_____에서 제공된다.

<36>

도 3은 DMA 엔진(300)의 일부분들을 예시하는 블록도이고, 도 4는 본 발명의 실시예들에 따른 링크된 명령 리스트 테이블(400)을 예시하는 블록도이다. DMA 엔진(300)은 메모리 허브(200)(도 2)의 DMA 엔진(286)에 대체될 수 있다. 도 3이 단지 DMA 엔진(300)의 표현이고 본 기술 분야의 전문가들은 본 발명을 행하기 위해 본 명세서에 충분한 설명이 제공되어 있다는 것을 이해할 것이다. 그러나, 대안적인 DMA 엔진들이 본 발명의 범위로부터 벗어나지 않으며 이용될 수 있다는 것 또한 이해될 것이다. DMA 엔진(300)은 DMA 동작들을 제어하기 위해 5개의 레지스터들, 어드레스 레지스터(310), 목적지 어드레스 레지스터(311), 제어 레지스터(312), 다음 레지스터(314), 및 카운트 레지스터(316)를 포함한다.

<37>

동작에 있어서, 블록 전송의 초기에, 블록에 대한 시작 어드레스는 어드레스 레지스터(310)로 로딩된다. 추가적으로, 데이터가 이동되기 위한 위치의 목적지 어드레스는 목적지 어드레스 레지스터(311)로 로딩되고, 상기 블

록의 길이는 카운트 레지스터(316)로 로딩된다. 제어 레지스터(312)는 각각의 데이터 아이템이 전송된 후에 어드레스 레지스터(310)가 증가 또는 감소되는지의 여부를 표시하는 비트로서 전송에 관한 정보를 포함한다. 현재의 예에 있어서, 데이터 아이템이 DMA 엔진(300)에 의해 전송되는 매 시간마다, 카운트 레지스터(316)가 감소되고 어드레스 레지스터(310)가 증가된다. 추가적으로, 목적지 어드레스 레지스터(311)는 증가된다(또는, 제어 세팅들에 의존하여 감소된다). 카운트 레지스터(316)의 값이 0에 도달할 때, 블록 전송은 완료되어 있다. 이 시간에, 다음 레지스터(314)의 값이 검사된다. 그것이 상기 시스템 메모리에서 유효한 위치를 가리키는 경우, 그 대상에 포함되는 상기 값들은 레지스터들(310, 312, 314, 316)로 로딩된다. 그 후에, 다음 블록 데이터 전송이 자동적으로 시작된다. 그러나, NULL 값이 이전에 기술된 바와 같이 다음 레지스터(314)에 존재하는 경우, DMA 동작이 완료된다.

<38> 도 4에 도시된 링크된 명령 리스트 테이블(400)은 각각의 레지스터들(310, 312, 314, 316)을 다시 로딩하기에 필요한 정보를 포함하는 복수의 링크 엔트리들(402, 404, 406)을 포함한다. 링크 엔트리들(402, 404, 406)은 이전에 논의된 바와 같이 상기 시스템 메모리에 저장되고, 다음 레지스터(314)에 대응하는 포인터들에 의해 함께 링크된다. 도 4에 있어서, 3개의 링크 엔트리들(402, 404, 406)이 도시되어 있다. DMA 엔진(300)의 레지스터들(310, 312, 314, 316)로 직접적으로 값들을 기록함으로써 규정되는 초기의 전송에 더하여 이러한 링크 엔트리들은 4개의 개별적 부분들을 갖는 단일 DMA 전송을 규정한다. 다음 레지스터(314)에 포함된 값(NEXT)은 제 1 링크 엔트리(402)를 가리킨다. 제 1 링크 엔트리(402)는 차례로 최종 링크 엔트리(406)를 가리키는 링크된 명령에서 다음 링크 엔트리(404)를 가리킨다. 최종 링크 엔트리(406)는 그것이 DMA 명령 리스트의 마지막 링크 엔트리라는 것을 표시하는 포인터로서 NULL 값을 포함한다. 상기 NULL 값은 유효한 메모리 위치를 가리키지 않는 예비된 포인터 값이다. 링크 엔트리들(402, 404, 406)이 예로써 제공되고, 도 4에 도시된 것 보다 더 크거나 더 적은 필드들의 정보를 포함하는 것과 같은 그에 따른 수정들이 본 발명의 범위로부터 벗어나지 않고 이루어질 수 있다는 것은 명백할 것이다.

<39> 도 5는 일련의 연속적인 블록 전송들을 위해 DMA 엔진(300)(도 3)에 의해 이용되는 제어 흐름을 예시하는 흐름도(500)이다. 단계(502)에서, DMA 레지스터들(310, 312, 314, 316)은 제 1 데이터 전송을 위해 적절한 값들로 로딩된다. 이 시간에 있어서, 상기 레지스터들을 직접적으로 로딩한 이후 또는 이전 둘 중 하나에서, 이러한 전송에 대한 링크 엔트리들을 위해 필요한 모든 정보는 링크된 명령 리스트 테이블(400)(도 4)로 로딩되어야 한다. 상기 레지스터들의 로딩은 프로세서(104)(도 1)의 명령에 있고, 상기 시스템 메모리에서 링크된 명령 리스트(400)의 로딩은 또한 프로세서(104)에 의해 달성된다.

<40> 단계(504)에서 하나의 데이터 아이템이 전송되고, 단계(506)에서 카운트 레지스터(316) 내 값은 하나의 데이터 아이템이 전송된 것을 표시하기 위해 감소된다. 단계(506)는 제어 레지스터(312)에서의 세트로서 바람직한 방향에 의존하여 어드레스 레지스터(310)의 값을 동시에 증가 또는 감소시키는 단계를 포함한다. 단계(508)에 있어서, 상기 카운트 값은 상기 카운트가 완료되는지의 여부를 결정하기 위해 검사된다. 본 발명의 하나의 실시예에 있어서, 상기 카운트가 완료되는지의 여부의 결정은 카운트 레지스터(316)로부터 캐리 아웃 비트(carry out bit)(도시되지 않음)를 검사함으로써 달성된다. 상기 데이터 전송이 완료되지 않은 것을 표시하는 상기 카운트 값의 경우에 있어서, 제어는 단계(504)로 복귀한다. 그러나, 카운트 레지스터(316)에서 카운트 값이 0와 동일한 경우, 제어는 단계(510)를 통과하고, 이전에 기술된 바와 같이 여기서 다음 레지스터(314)에서의 값은 그것이 NULL 값과 동일한 경우, 보기 위해 테스트된다. NULL 값이 존재하지 않는 경우, 단계(512)에서 다음 태그는 링크된 명령 리스트 테이블(400)로부터 DMA 제어기(300) 내 레지스터들(310, 312, 314, 316)로 로딩되고, 제어는 단계(504)로 복귀한다. 일단 마지막 링크 엔트리가 이용되면, 단계(514)에서 표시는 전송이 완료된 프로세서(104)에 만들어진다.

<41> 본 기술 분야의 전문가들에게는 DMA 엔진(300)이 상기 시스템 메모리에서 이용을 위해 "분산 수집(scatter-gather)" 능력을 구현한다는 것이 이해될 것이다. 데이터의 큰 블록이 메모리의 불연속 블록들로 판독하려 할 때, 프로세서(104)는 DMA 엔진(300)을 통해 상기 메모리를 할당하고 링크된 명령 리스트 테이블(400)을 셋 업한다. 그 후에, DMA 전송이 개시되고, DMA 엔진(300)은 그것이 완료될 때까지 전체 전송을 취급한다. 유사한 테크닉은 메모리의 연속적인 블록들에 그것들을 기록하도록 상기 시스템 메모리 내 데이터의 분산된 블록들을 수집하기 위해 이용될 수 있다. 프로세서(104)는 어느 블록들이 상기 시스템 메모리 내에서 이동되어 기록되는지와 그의 순서를 결정하고, DMA 엔진(300)을 통해 링크된 명령 리스트 테이블(400)을 셋 업한다. DMA 전송은 그 후에 개시되고, 그것이 완료될 때까지 DMA 엔진(300)에 의해 완전히 다루어진다. 링크된 명령 리스트 테이블(400)이 상기 시스템 메모리에 저장되기 때문에, 예를 들어 DMA 엔진(300)에 의해 지원되는 각각의 채널에 대해 몇 개의 링크된 리스트들을 유지하는 것이 가능하다. 더욱이, 링크된 명령 리스트 테이블(400)이 상기 시스템 메모리

리에 저장되기 때문에, 채널에 대해 하나의 보다 큰 전송으로 링크될 수 있는 개별적 전송들의 수에 대한 유일한 제한은 상기 시스템 메모리 내 남아있는 자유로운 메모리 위치들의 수이다.

<42> 본 발명의 특정 실시예들이 예시적인 목적들을 위해 본 명세서에 기술되어 있을 지라도, 다양한 수정들이 본 발명의 범위 및 취지로부터 벗어나지 않고 이루어질 수 있다는 것이 상기된 바를 통해 이해될 것이다. 또한 본 발명은 첨부된 청구항을 제외하고는 제한되지 않는다.

도면의 간단한 설명

<12> 도 1은 메모리 허브가 복수의 메모리 모듈 각각에서 포함되는 본 발명에 따른 컴퓨터 시스템을 도시한 블록도.

<13> 도 2는 도 1의 컴퓨터 시스템에서 이용되는 메모리 허브를 도시한 블록도.

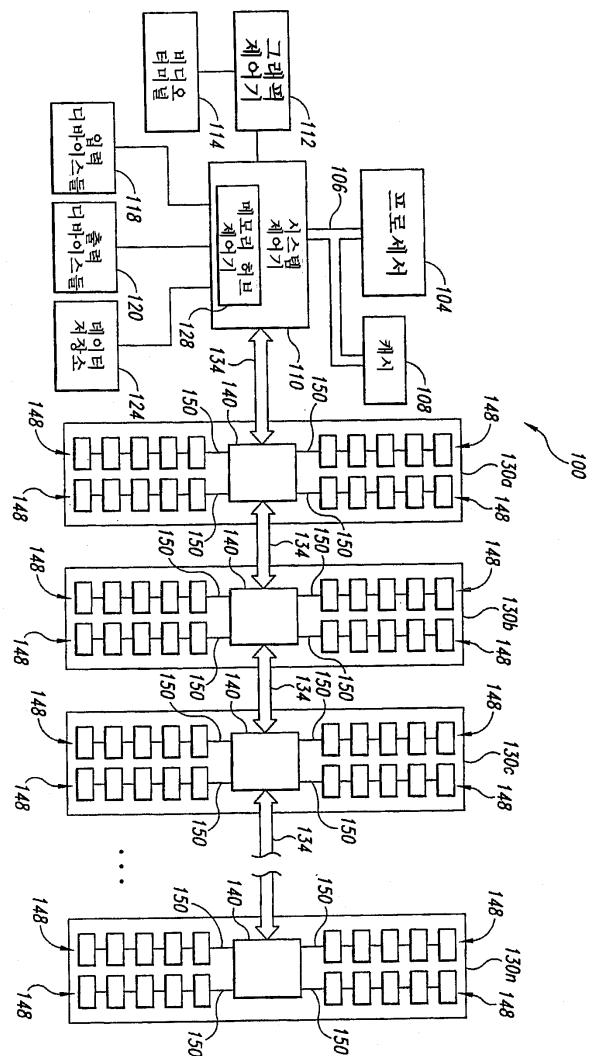
<14> 도 3은 도 2의 메모리 허브의 본 발명의 실시예에 따른 DMA 엔진의 일부를 도시한 블록도.

<15> 도 4는 도 3의 DMA 엔진에 의해 이용되는 본 발명의 실시예에 따른 태그 구조를 도시한 블록도.

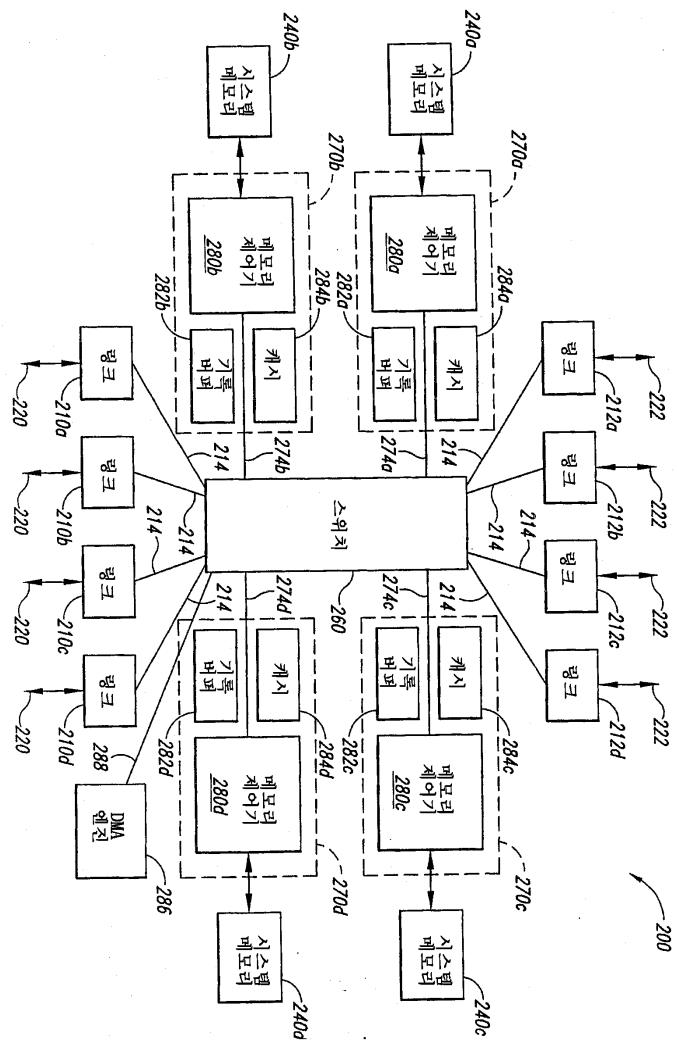
<16> 도 5는 본 발명의 실시예에 따른 도 3의 DMA 엔진의 동작을 도시한 흐름도.

도면

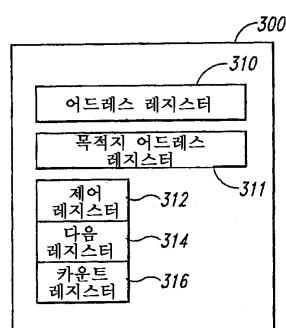
도면1



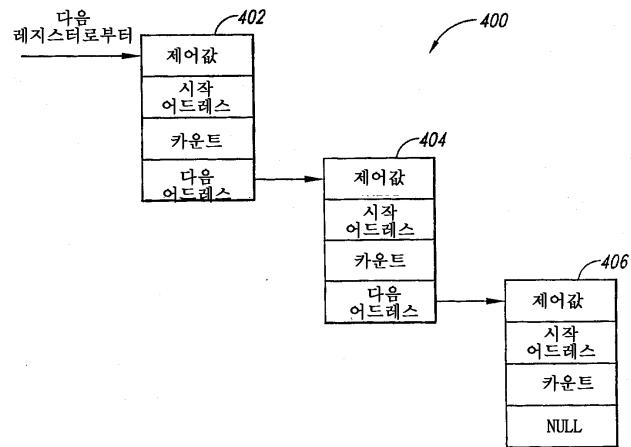
도면2



도면3



도면4



도면5

