



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년08월19일

(11) 등록번호 10-1545682

(24) 등록일자 2015년08월12일

(51) 국제특허분류(Int. Cl.)

G06F 13/38 (2006.01) G06F 13/14 (2006.01)  
G09G 5/36 (2006.01) G09G 5/395 (2006.01)

(21) 출원번호 10-2010-7014602

(22) 출원일자(국제) 2008년12월01일

심사청구일자 2013년11월28일

(85) 번역문제출일자 2010년06월30일

(65) 공개번호 10-2010-0114496

(43) 공개일자 2010년10월25일

(86) 국제출원번호 PCT/US2008/085160

(87) 국제공개번호 WO 2009/073617

국제공개일자 2009년06월11일

(30) 우선권주장

11/987,559 2007년11월30일 미국(US)

(56) 선행기술조사문현

US06532019 B1

US07289125 B2

(73) 특허권자

에이티아이 테크놀로지스 유엘씨

캐나다 온타리오 엘3티 7엑스6 마크햄 커머스 벨리 드라이브 이스트 1

에이티아이 인터내셔널 에스알엘

바베이도스 크라이스트 처치 헤이스팅스 뷔몬트 하우스

(72) 발명자

쳉 제프리

캐나다 온타리오 엠2케이 2에이4 토론토 오스카 코트 8

라비올레 테리

캐나다 온타리오 엘1이 2씨3 커티스 어번 레인 41 (뒷면에 계속)

(74) 대리인

박장원

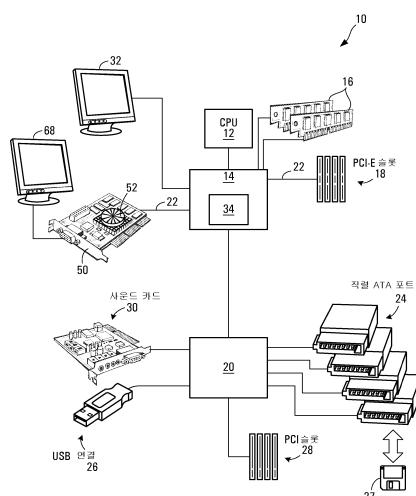
전체 청구항 수 : 총 17 항

심사관 : 고재용

(54) 발명의 명칭 고속의 주변 상호연결 버스를 통한 비디오 렌더링

**(57) 요 약**

하나의 그래픽 프로세서에 의해 생성된 그래픽들은 고속의 확장/상호연결 버스를 통해 프레임 버퍼로 전송된다. 프레임 버퍼로부터의 렌더링된 프레임들은 프레임 버퍼와 통신하는 디스플레이 인터페이스를 통해 디스플레이 기기 상에 표시된다. 기존의(예컨대, 통합된) 다른 그래픽 어댑터/서브시스템의 디스플레이 인터페이스는, 렌더링된 프레임들을 상호연결된 디스플레이 기기 상에 표시하는데 이용될 수 있다.

**대 표 도 - 도1**

(72) 발명자

**후앙 제임스**

캐나다 온타리오 엠2엔 4엠7 노스 요크 홀메스 애버뉴 171에이

**자브르지키 로버트**

캐나다 온타리오 엘5에이치 4에이1 미시사우가 레인트리 레인 1019

**톰 제이슨**

캐나다 온타리오 엘4에스 2제이5 리치몬드 힐 팜스테드 로드 185

**웡 시양쿠안**

캐나다 온타리오 엠4에이 2제이9 토론토 윈드클리프 크레센트 91

**마린코빅 샤샤**

캐나다 온타리오 엠6비 4비5 토론토 말리 애버뉴 155

**무마흐 필**

미국 캘리포니아 94063 레드우드 시티 페.오. 박스 5826

---

**치엔 링웨이**

캐나다 온타리오 엠2제이 5지5 토론토 갓스톤 로드 76

**트레시더 마이클**

캐나다 온타리오 엘3엑스 1엘6 뉴마켓 마리너 레인 660

**솔트체프 로멘**

캐나다 온타리오 엘3알 4씨4 마크햄 부차난 드라이브 73

**시 조지**

캐나다 온타리오 엠2제이 1지3 노스요크 조지 헨리 블러바드 414-130

**리츠마노프 이오우리**

캐나다 온타리오 엘4씨 5엔2 리치몬드 힐 유닛 20 캐슬 록 드라이브 141

## 명세서

### 청구범위

#### 청구항 1

컴퓨팅 디바이스를 동작시키는 방법으로서, 상기 컴퓨팅 디바이스는, 프로세서, 상기 프로세서와 통신하는 메모리, 및 제1 그래픽 서브시스템을 포함하고, 상기 제1 그래픽 서브시스템은 제1 그래픽 프로세서 및 비디오 프레임들을 디스플레이 상에 표시하는 디스플레이 인터페이스를 포함하며, 상기 프로세서와 상기 제1 그래픽 서브시스템은 인터페이스 회로에 의해 상호연결되고, 상기 컴퓨팅 디바이스는, 제2 그래픽 프로세서를 포함하고 전용 프레임 버퍼 메모리를 포함하지 않는 제2 그래픽 서브시스템, 및 주변 상호연결 버스 접속(peripheral interconnect bus connection)을 더 포함하며, 상기 제2 그래픽 서브시스템에 제공되거나 또는 상기 제2 그래픽 서브시스템으로부터 제공되는 모든 데이터가 상기 인터페이스 회로와 상기 주변 상호연결 버스 접속을 통해 제공되도록, 상기 주변 상호연결 버스 접속은 상기 제2 그래픽 서브시스템 및 상기 인터페이스 회로에서 종료(terminate)되고,

상기 방법은,

상기 디스플레이 인터페이스와 상호연결된 상기 메모리 내에 프레임 버퍼를 설정(establish)하는 단계와;

상기 주변 상호연결 버스 접속을 통해 상기 프레임 버퍼 안으로, 프리미티브 단위로(primitive by primitive), 비디오 프레임들을 렌더링하도록 상기 주변 상호연결 버스 접속을 통해 상기 제2 그래픽 프로세서에 명령어들을 제공하는 단계 - 상기 디스플레이 상에 표시되는 모든 프레임들은 상기 제2 그래픽 프로세서에 의해 렌더링되고 - 와; 그리고

상기 프레임 버퍼로부터의 상기 프레임들을 상기 디스플레이 인터페이스를 통해 상기 디스플레이 상에 표시하는 단계

를 포함하는 컴퓨팅 디바이스를 동작시키는 방법.

#### 청구항 2

제1항에 있어서,

상기 그래픽 프로세서는 제 1 그래픽 어댑터의 일부인 것을 특징으로 하는 컴퓨팅 디바이스를 동작시키는 방법.

#### 청구항 3

제1항에 있어서,

상기 프레임 버퍼 내의 상기 프레임들을 상기 디스플레이 상에서 표시하도록 상기 디스플레이 인터페이스를 프로그래밍하는 단계를

더 포함하는 것을 특징으로 하는 컴퓨팅 디바이스를 동작시키는 방법.

#### 청구항 4

제1항에 있어서,

상기 제2 그래픽 서브시스템은 상기 주변 상호연결 버스 접속에 착탈가능하게 상호연결되는 것을 특징으로 하는 컴퓨팅 디바이스를 동작시키는 방법.

#### 청구항 5

제1항에 있어서,

상기 제2 그래픽 서브시스템은 제2 그래픽 어댑터 상에 형성되는 것을 특징으로 하는 컴퓨팅 디바이스를 동작시키는 방법.

## 청구항 6

제1항에 있어서,

상기 제2 그래픽 서브시스템은 상기 제1 그래픽 서브시스템에 의해 제공되지 않는 그래픽 처리 기능을 제공하는 것을 특징으로 하는 컴퓨팅 디바이스를 동작시키는 방법.

## 청구항 7

컴퓨팅 디바이스로서,

중앙 프로세서와;

메모리와;

디스플레이 상에 비디오 프레임들을 표시하기 위해 상호연결되는 제1 그래픽 프로세서와 디스플레이 인터페이스를 포함하는 제1 그래픽 서브시스템과;

상기 중앙 프로세서, 상기 메모리 및 상기 제1 그래픽 서브시스템을 상호연결하는 인터페이스 회로와;

제2 그래픽 프로세서를 포함하고 전용 프레임 버퍼 메모리를 포함하지 않는 제2 그래픽 서브시스템과;

상기 제2 그래픽 서브시스템에 제공되거나 또는 상기 제2 그래픽 서브시스템으로부터 제공되는 모든 데이터가 상기 인터페이스 회로와 주변 상호연결 버스 접속을 통해 제공되도록 상기 제2 그래픽 서브시스템과 상기 인터페이스 회로를 상호연결하여 상기 제2 그래픽 서브시스템 및 상기 인터페이스 회로에서 종료되는 상기 주변 상호연결 버스 접속과;

렌더링하는 명령어들을 수신하도록 그리고 상기 주변 상호연결 버스 접속을 통해 그래픽들을 렌더링하도록 동작 가능한 상기 제2 그래픽 서브시스템과; 그리고

상기 메모리에 저장되는 컴퓨터 실행가능한 명령어들을 포함하고,

상기 명령어들은 상기 컴퓨팅 디바이스로 하여금,

상기 주변 상호연결 버스 접속을 통해 상기 메모리 내의 상기 프레임 버퍼 안으로, 프리미티브 단위로, 상기 제2 그래픽 프로세서가 상기 디스플레이 상에 표시되는 모든 비디오 프레임들을 렌더링하도록 상기 제2 그래픽 프로세서에 명령어들을 제공하게 하고,

상기 프레임 버퍼로부터 상기 렌더링된 프레임들을 상기 디스플레이 인터페이스를 통해 상기 디스플레이 상에 표시하게 하는 것을 특징으로 하는 컴퓨팅 디바이스.

## 청구항 8

제7항에 있어서,

상기 디스플레이 인터페이스는 주변 확장 인터페이스의 일부를 구성하는 것을 특징으로 하는 컴퓨팅 디바이스.

## 청구항 9

제8항에 있어서,

상기 그래픽 프로세서는, 상기 주변 상호연결 버스와 통신하는 주변 확장 슬롯에서 주변 확장 카드의 일부를 구성하는 것을 특징으로 하는 컴퓨팅 디바이스.

## 청구항 10

제7항에 있어서,

상기 그래픽 프로세서는, 프레임 버퍼 메모리와 디스플레이 인터페이스를 포함하지 않는 그래픽 서브시스템의 일부를 구성하는 것을 특징으로 하는 컴퓨팅 디바이스.

## 청구항 11

제7항에 있어서,

상기 제2 그래픽 서브시스템은 상기 제1 그래픽 서브시스템에 의해 제공되지 않는 그래픽 처리 기능을 제공하는 것을 특징으로 하는 컴퓨팅 디바이스.

#### 청구항 12

컴퓨팅 디바이스로서,

중앙 프로세서와;

상기 중앙 프로세서와 통신하는 메모리와;

그래픽 메모리 내의 프레임 버퍼와;

디스플레이 상에 비디오 플레이들을 표시하기 위해 상기 그래픽 메모리와 상호연결되는 디스플레이 인터페이스를 포함하는 제1 그래픽 서브시스템과;

상기 중앙 프로세서, 상기 메모리 및 상기 제1 그래픽 서브시스템을 상호연결하는 인터페이스 회로와;

주변 확장 카드상에 형성되고, 제2 그래픽 서브시스템에 로컬(local)인 프레임 버퍼를 정의하는 메모리를 갖지 않는 상기 제2 그래픽 서브시스템과;

상기 제2 그래픽 서브시스템에 제공되거나 또는 상기 제2 그래픽 서브시스템으로부터 제공되는 모든 데이터가 주변 상호연결 버스 접속을 통해 제공되도록 상기 제2 그래픽 서브시스템과 상기 인터페이스 회로를 상호연결하여 상기 제2 그래픽 서브시스템 및 상기 인터페이스에서 종료되는 상기 주변 상호연결 버스 접속과; 그리고

렌더링하는 명령어들을 수신하도록 그리고 상기 주변 상호연결 버스 접속을 통해 그래픽들을 렌더링하도록 동작 가능한 상기 제2 그래픽 서브시스템

을 포함하고,

상기 프로세서와 통신하는 상기 메모리는 컴퓨터 실행가능한 명령어들을 저장하며, 상기 명령어들은 상기 그래픽 서브시스템으로 하여금 상기 주변 상호연결 버스 접속을 통해 상기 프레임 버퍼 안으로, 프리미티브 단위로, 상기 디스플레이 상에 표시되는 모든 비디오 프레임들을 렌더링하게 하는 것을 특징으로 하는 컴퓨팅 디바이스.

#### 청구항 13

제12항에 있어서,

상기 그래픽 서브시스템은, 상기 주변 상호연결 버스에 연결된 주변 확장 카드 상에 형성되는 것을 특징으로 하는 컴퓨팅 디바이스.

#### 청구항 14

제12항에 있어서,

주변 확장 인터페이스의 일부를 구성하는 상기 디스플레이 인터페이스를 더 포함하는 것을 특징으로 하는 컴퓨팅 디바이스.

#### 청구항 15

제12항에 있어서,

상기 디스플레이 인터페이스는, 상기 중앙 프로세서와 상기 주변 상호연결 버스를 상호연결하는 주변 확장 인터페이스상의 상기 그래픽 서브시스템의 일부를 구성하는 것을 특징으로 하는 컴퓨팅 디바이스.

#### 청구항 16

그래픽 서브시스템으로서,

그래픽 프로세서와; 그리고

상기 그래픽 서브시스템에 제공되거나 또는 상기 그래픽 서브시스템으로부터 제공되는 모든 데이터가 주변 상호연결 버스 접속을 통해 전송되도록 상기 그래픽 서브시스템과 호스트 컴퓨팅 디바이스의 주변 상호연결 버스를

상호연결하기 위한 주변 상호연결 버스 인터페이스를 포함하고,

상기 호스트 컴퓨팅 디바이스는 디스플레이 상에 비디오 프레임들을 표시하기 위해 그래픽 메모리와 상호연결되는 디스플레이 인터페이스를 포함하고,

상기 그래픽 서브시스템은 상기 그래픽 서브시스템에 대해 로컬(local)인 프레임 버퍼를 정의하는 메모리를 포함하지 않고,

상기 그래픽 서브시스템은 렌더링하는 명령어들을 수신하도록 그리고 상기 주변 상호연결 버스를 통해 상기 호스트 컴퓨팅 디바이스의 상기 그래픽 메모리 내의 상기 프레임 버퍼 안으로, 프리미티브 단위로, 비디오 프레임들을 렌더링하도록 동작가능하며,

상기 그래픽 프로세서는 상기 디스플레이 상에 표시되는 모든 비디오 프레임들을 렌더링하고, 모든 그래픽 명령어들은 상기 주변 상호접속 버스 인터페이스를 통해 상기 호스트 컴퓨팅 디바이스로부터 수신되는 것을 특징으로 하는 그래픽 서브시스템.

### 청구항 17

컴퓨팅 디바이스로 하여금 청구항 제1항의 방법을 수행케하는 컴퓨터 실행가능한 명령어들을 저장하는 컴퓨터 관독가능한 매체.

### 청구항 18

삭제

### 청구항 19

삭제

### 청구항 20

삭제

## 발명의 설명

### 기술 분야

[0001] 관련 출원들에 대한 상호-참조

[0002] 본 출원은 2005년 5월 24일자 미국 특허출원(11/136,483)과 2006년 5월 30일자 미국 특허출원(11/421,005)의 일부-계속 출원(continuation-in-part)이며, 상기 2개의 미국 특허출원들은 본 발명에 대한 참조로서 그 전체 내용이 본 명세서에 통합된다.

[0003] 일반적으로 본 발명은 컴퓨팅 디바이스에 관한 것이며, 좀더 상세하게는 고속의 주변(peripheral) 버스에 의해 상호연결된 그래픽 프로세서들을 포함하는 컴퓨팅 디바이스에 관한 것이다.

### 배경 기술

[0004] 최근 몇년 동안, 컴퓨터 하드웨어는 점점 더 통합되어 왔다. 컴퓨터 주변장치들은 종종 집적회로 칩셋의 일부로서 구성되며, 상기 집적회로 칩셋은 중앙 처리장치와 협력한다. 컴퓨터 오디오, 그래픽 어댑터들 및 인터페이스들은 집적회로 칩셋의 일부를 구성한다. 이러한 통합은 컴퓨팅 디바이스들의 비용 및 사이즈를 감소시킨다.

[0005] 이와 동시에, 그래픽 프로세서는 점점 더 개량되어 향상된 성능과 특성을 제공한다. 최근의 그래픽 프로세서들은 중앙처리장치(CPU) 만큼이나 복잡하다. 따라서, 통합화 경향에도 불구하고, 고성능 그래픽 프로세서들은 주변 카드(peripheral card) 상에 형성되는 것이 일반적이다. 주변 카드 상에 형성된 그래픽 프로세서와 비교하면, 통합된 그래픽 프로세서들은 상대적으로 간단하며, 고성능을 제공하지는 못한다. 애드-온(add-on) 주변 카드의 등장으로 인해, 통합된 그래픽 구성요소들은 불필요한 것이 되었으며 따라서 디스에이블된다.

### 발명의 내용

### 해결하려는 과제

[0006] 이와 유사하게, 통합형인지 또는 외장형인지에 관계없이 그래픽 어댑터가 업그레이드됨에 따라, 성능이 떨어지는 어댑터들은 설 자리를 잃게 되었으며, 이를 그래픽 어댑터들은 폐기 혹은 디스에이블된다.

[0007] 따라서, 컴포넌트 리던던시(component redundancy)를 감소시킬 필요가 있다.

### 과제의 해결 수단

[0008] 본 발명의 일 양상에 따르면, 하나의 그래픽 프로세서에 의해 생성된 그래픽들은 고속의 확장/상호연결 버스를 통해 프레임 버퍼로 전송된다. 프레임 버퍼로부터의 렌더링된 프레임들은 프레임 버퍼와 통신하는 디스플레이 인터페이스를 통해 디스플레이 기기 상에 표시된다.

[0009] 유용하게도, 기존의(예컨대, 통합된) 다른 그래픽 어댑터/서브시스템의 디스플레이 인터페이스는, 렌더링된 프레임들을 상호연결된 디스플레이 기기 상에 표시하는데 이용될 수 있다.

[0010] 이러한 방식으로 그래픽 프로세서는, 디스플레이 인터페이스를 포함할 필요가 없는 혹은 가능하다면 프레임 버퍼조차도 포함할 필요가 없는 그래픽 어댑터의 일부를 구성할 수도 있다. 이와 유사하게, 그래픽 프로세서는 좀 더 최신의 그래픽 프로세서로 대체될 수 있다.

[0011] 그래픽 프로세서는 주변 확장 카드 상에 내장된 그래픽 어댑터의 일부를 구성할 수도 있다.

[0012] 본 발명의 일실시예에서, 컴퓨팅 디바이스는 제 1 및 제 2 그래픽 어댑터를 포함할 수도 있다. 디스플레이 기기 상에 표시될 이미지들은 메모리로 전송되는바, 상기 메모리는 제 2 그래픽 어댑터에 의해 액세스가능하다.

[0013] 제 2 그래픽 어댑터의 디스플레이 인터페이스는, 제 2 그래픽 어댑터의 메모리 내의 이미지들을 다수개의 디스플레이 기기를 중 적어도 하나 상에 표시하기 위해 이용된다. 이러한 방식으로, 제 2 어댑터의 포트를 뿐만 아니라 디스플레이 인터페이스를 구성하는 디바이스 전자장비들도 활용될 수 있다. 렌더링된 그래픽들은 프레임 단위로(frame by frame) 전송될 수도 있으며 혹은 그래픽 프리미티브 단위로(graphics primitive by graphics primitive) 전송될 수도 있다. 후자의 경우, 그래픽들은 제 2 그래픽 어댑터에 의해 액세스가능한 메모리에 직접적으로 렌더링된다. 이러한 메모리는 시스템 메모리 혹은 어댑터 메모리가 될 수 있다.

[0014] 본 발명의 다른 실시예에 따르면, 고속의 상호연결 버스를 통해 전송된 프레임들을 제 1 그래픽 프로세서가 렌더링하는 동안, 제 2 어댑터의 그래픽 프로세서는 디스에이블될 수 있으며 혹은 저전력 모드로 될 수도 있다.

[0015] 본 발명의 다른 양상에 따르면, 제 1 그래픽 어댑터의 일부를 구성하는 그래픽 프로세서는, 제 1 및 제 2 디스플레이 상에 제 1 및 제 2 이미지를 각각 드로잉(drawing)하도록 동작한다. 상기 제 1 디스플레이는 제 1 그래픽 어댑터의 제 1 디스플레이 인터페이스에 상호연결될 수 있다. 상기 제 2 디스플레이는 제 2 그래픽 어댑터의 제 2 디스플레이 인터페이스에 상호연결될 수 있다. 본 발명의 방법은, 제 1 그래픽 어댑터의 메모리 내의 제 1 및 제 2 이미지를 렌더링하는 단계, 제 2 이미지를 제 1 그래픽 어댑터의 상기 메모리로부터 제 2 그래픽 어댑터의 프레임 버퍼로 전송하는 단계, 및 제 2 그래픽 어댑터의 제 2 디스플레이 인터페이스를 이용하여, 제 2 그래픽 어댑터의 프레임 버퍼로부터의 상기 제 2 이미지를 제 2 디스플레이 상에 표시하는 단계를 포함한다.

[0016] 본 발명의 다른 양상에 따르면 컴퓨팅 디바이스가 제공되는바, 상기 컴퓨팅 디바이스는 프로세서, 상기 프로세서와 통신하는 컴퓨터 판독가능 메모리, 제 1 그래픽 어댑터, 상기 제 1 그래픽 어댑터와 통신하는 제 1 어댑터 메모리, 제 1 디스플레이 상에 이미지들을 표시하기 위한 제 1 디스플레이 인터페이스, 제 2 그래픽 어댑터, 제 2 디스플레이 상에 이미지들을 표시하기 위한 제 2 디스플레이 인터페이스, 상기 제 1 및 제 2 그래픽 어댑터를 상호연결하는 고속의 버스를 포함한다. 프로그램 코드가 컴퓨터 판독가능 메모리에 저장되며, 상기 프로그램 코드는 프로세서에 의해 실행가능하다. 프로그램 코드가 실행되는 때에 상기 프로그램 코드는, 제 1 그래픽 어댑터가 제 1 및 제 2 이미지를 제 1 어댑터 메모리 내에 렌더링하게 하며, 제 1 어댑터 메모리로부터 제 2 그래픽 어댑터의 프레임 버퍼로 제 2 이미지를 전송케하며, 그리고 제 2 그래픽 어댑터의 프레임 버퍼로부터 상기 제 2 이미지가 제 2 디스플레이에 표시되도록 제 2 그래픽 어댑터의 제 2 디스플레이 인터페이스가 프로그램되게 한다.

[0017] 본 발명의 또 다른 양상에 따르면, 제 1 어댑터는 제 1 이미지를 시스템 메모리 내에 직접 렌더링하며, 시스템 메모리는 제 2 어댑터에 의해 액세스 가능하다. 제 2 어댑터는 제 2 이미지를 동일 메모리에 렌더링한다. 제 2 어댑터는 시스템 메모리로부터의 제 1 및 제 2 이미지 둘다를 디스플레이한다.

[0018]

본 발명의 또 다른 양상에 따르면, 컴퓨팅 디바이스를 동작시키는 방법이 제공된다. 컴퓨팅 디바이스는 프로세서, 상기 프로세서와 통신하는 메모리, 상기 프로세서를 그래픽 프로세서에 상호연결하는 주변 상호연결 버스, 및 디스플레이 인터페이스를 포함한다. 상기 방법은 메모리 내에 프레임 버퍼를 설립(establish)하는 단계, 비디오 프레임들을 렌더링하도록 상기 그래픽 프로세서에게 명령하는 단계, 상기 주변 상호연결 버스를 통해 상기 프레임 버퍼 안으로 상기 비디오 프레임들을 전송하는 단계 및 상기 프레임 버퍼로부터의 상기 프레임들을 상기 디스플레이 인터페이스를 통해 디스플레이 상에 표시하는 단계를 포함한다.

[0019]

본 발명의 또 다른 양상에 따르면, 컴퓨팅 디바이스는 중앙 프로세서, 상기 중앙 프로세서와 통신하는 메모리, 상기 중앙 프로세서를 그래픽 프로세서에 상호연결하는 주변 상호연결 버스, 디스플레이 인터페이스, 상기 메모리에 저장되는 컴퓨터 실행가능한 명령들을 포함하며, 상기 명령들은 상기 컴퓨팅 디바이스로 하여금, 상기 그래픽 프로세서가 비디오 프레임들을 렌더링하게 하며, 상기 비디오 프레임들을 상기 주변 상호연결 버스를 통해 프레임 버퍼로 전송하게 하며, 그리고 상기 프레임 버퍼로부터 상기 렌더링된 프레임들을 상기 디스플레이 인터페이스를 통해 디스플레이 기기에 표시하게 한다.

[0020]

본 발명의 또 다른 양상에 따르면, 컴퓨팅 디바이스가 제공되는바, 상기 컴퓨팅 디바이스는 중앙 프로세서, 상기 중앙 프로세서와 통신하는 메모리, 제 1 프레임 버퍼, 상기 중앙 프로세서를 그래픽 서브시스템에 상호연결하는 주변 상호연결 버스, 상기 그래픽 서브시스템은 상기 그래픽 서브시스템에 로컬(local)인 프레임 버퍼를 정의하는 메모리를 갖지 않으며, 그리고 상기 메모리와 통신하는 디스플레이 인터페이스를 포함한다. 상기 메모리는 컴퓨터 실행가능한 명령들을 저장하며, 상기 명령들은 상기 그래픽 프로세서로 하여금 비디오 프레임들을 렌더링하게 하며 그리고 상기 주변 상호연결 버스를 통해 상기 비디오 프레임들을 상기 제 1 프레임 버퍼로 전송하게 한다.

[0021]

본 발명의 또 다른 양상에 따르면, 그래픽 서브시스템이 제공되는바, 상기 그래픽 서브시스템은 그래픽 프로세서, 그리고 상기 그래픽 서브시스템을 호스트 프로세서에 상호연결하기 위한 주변 상호연결 버스 인터페이스를 포함하며, 상기 그래픽 서브시스템은 상기 그래픽 서브시스템에 대해 로컬(local)인 프레임 버퍼를 정의하는 메모리를 포함하지 않으며, 그리고 상기 그래픽 서브시스템은 비디오 프레임들을 렌더링하며 그리고 상기 주변 상호연결 버스를 통해 상기 비디오 프레임들을 프레임 버퍼에 전송하도록 동작가능하다.

[0022]

본 발명의 또 다른 양상에 따르면, 제 1 및 제 2 이미지를 제 1 및 제 2 디스플레이 상에 각각 드로잉하도록 제 1 그래픽 프로세서를 동작시키는 방법이 제공된다. 제 2 디스플레이는 제 2 그래픽 어댑터의 디스플레이 인터페이스에 상호연결된다. 상기 방법은, 제 1 그래픽 프로세서와 통신하는 메모리 내의 제 1 및 제 2 이미지를 렌더링하는 단계, 제 2 이미지를 제 2 그래픽 어댑터의 프레임 버퍼로 전송하는 단계, 그리고 제 2 그래픽 어댑터의 프레임 버퍼로부터 제 2 이미지를 제 2 디스플레이 상에 표시하도록 제 2 그래픽 어댑터의 디스플레이 인터페이스를 프로그램하는 단계를 포함한다.

[0023]

첨부된 도면들과 함께 본 발명의 특정 실시예들에 대한 후술되는 설명을 참고하면 본 발명의 다른 양상들 및 피쳐들이 해당 기술분야의 당업자들에게 명백해질 것이다.

### 도면의 간단한 설명

[0024]

도면들은 본 발명의 실시예들을 단지 일례로서 예시한다.

도1은 본 발명의 일실시예에 따른 컴퓨팅 디바이스의 간략화된 블록도이다.

도2는 도1에 도시된 컴퓨팅 디바이스에서의 소프트웨어의 간략화된 논리 블록도이다.

도3은 도1에 도시된 컴퓨팅 디바이스의 일부에 대한 간략화된 블록도이다.

도4는 도1에 도시된 컴퓨팅 디바이스의 일부에 대한 또 다른 블록도이다.

도5A는 본 발명의 일실시예에 따라 도1의 컴퓨팅 디바이스에서 수행되는 단계들을 예시한 순서도이다.

도5B는 도1의 디바이스의 구성요소들에 대한 도5A의 단계들의 효과를 예시한 도면이다.

도6A는 본 발명의 일실시예에 따라 도1의 컴퓨팅 디바이스에서 수행되는 단계들을 예시한 순서도이다.

도6B는 도1의 디바이스의 구성요소들에 대한 도6A의 단계들의 효과를 예시한 도면이다.

도7 및 도8은 본 발명의 다른 일실시예에 따라 도1의 컴퓨팅 디바이스에서 수행되는 단계들을 예시한 순서도이

다.

도9는 본 발명의 다른 일실시예에 따른 컴퓨팅 디바이스의 일부에 대한 또 다른 부분 개략도이다.

도10은 본 발명의 일실시예에 따라 도9의 디바이스에서 소프트웨어에 의해 수행되는 단계들을 상세히 도시한 순서도이다.

도11A, 11B는 도9에 도시된 디바이스의 동작을 예시한 간략화된 블록도이다.

도12는 본 발명의 다른 일실시예에 따른 컴퓨팅 디바이스의 일부에 대한 또 다른 부분 개략도이다.

### 발명을 실시하기 위한 구체적인 내용

[0025] 도1은 본 발명의 일실시예에 따른 컴퓨팅 디바이스에 대한 간략화된 블록도이다. 컴퓨팅 디바이스(10)는 통상적인 인텔 x86 아키텍처를 기반으로 한다. 하지만, 본 발명은 PowerPC, AMD x86, 또는 다른 아키텍처를 갖는 컴퓨팅 디바이스에도 용이하게 적용될 수 있음을 유의해야 한다.

[0026] 컴퓨팅 디바이스(10)는 프로세서(12), 통합 인터페이스 회로(14, 20)를 통해 상호연결된 시스템 메모리(16)와 주변기기들을 포함한다.

[0027] 예시적인 컴퓨팅 디바이스(10)의 프로세서(12)는 통상적인 중앙처리유닛(CPU)이며, 예를 들면, 인텔™ x86 계열의 마이크로프로세서와 호환가능한 마이크로프로세서가 될 수도 있다.

[0028] 통합 인터페이스 회로(14, 20)는 종종 노스 브리지(north bridge) 및 사우스 브리지라고 각각 호칭되며, 주변기들 및 메모리로의 인터페이스를 프로세서(12)에게 제공한다. 예시된 바와 같이, 노스 브리지(14)는 프로세서(12), 메모리(16), 고속의 상호연결 버스(22)를 경유하는 다수의 확장 슬롯(18), 그리고 사우스 브리지(20)를 상호연결한다. 또한, 사우스 브리지(20)는, 가령, 통합된 IDE/SATA 포트(24)를 경유하는 디스크 드라이버와 같은 저속의 주변기기들 및 상호연결들, 통합된 범용 직렬 버스(USB) 포트(26)를 경유하는 범용 직렬 버스 디바이스들, 그리고 예컨대 공지된 PCI 혹은 ISA 규격을 따르는 저속의 상호연결 버스(28)를 경유하는 또 다른 주변기기들을 상호연결한다. 또한, 사우스 브리지(20)는 통합된 오디오 콤포넌트(30), 네트워크 인터페이스(미도시) 등등을 더 포함할 수도 있다.

[0029] 예시적인 상호연결 버스(22)는 PCI 익스프레스(PCIe) 버스인바, 이는 초당 기가 바이트 범위의 대역폭을 가지며, 이 대역폭에서 판독(read) 및 기입(write)의 데이터 전송이 가능하다. 예를 들면, 현재의 PCIe 버스의 속도는 266 MHz 혹은 533 MHz 정도의 속도에서 양-방향 데이터 전송을 가능케하며, 초당 2.1 기가 바이트 혹은 4.2 기가 바이트 정도의 데이터 속도(data rate)를 허용한다.

[0030] 예시된 실시예에서, 노스 브리지(14)는, 모니터, LCD 패널, 텔레비전 등등의 형태인 제 1 디스플레이 장치(32)에 컴퓨팅 디바이스(10)를 상호연결하기에 적합한 통합된 그래픽 어댑터(34)를 더 포함한다.

[0031] 아래에서 더욱 상세히 설명되는 바와 같이, 상호연결 버스(22)상의 확장 슬롯(18) 내에서 예컨대, 주변 확장 카드(50)의 일부를 구성하며, 상호연결 버스(22)에 의해서 상호연결되는 또 다른 그래픽 어댑터(52)는 컴퓨팅 디바이스(10)의 일부가 될 수도 있다.

[0032] 예시된 실시예에서, 컴퓨팅 디바이스(10)는 메모리(16) 내에 저장된 소프트웨어를 실행한다. 도2에 도시된 바와 같이, 예시적인 소프트웨어(100)는 메모리 내(16)에 저장된, 운영 시스템(102), 그래픽 라이브러리(104) 그리고 어플리케이션 소프트웨어(106)를 포함한다. 예시적인 운영 시스템은 윈도우 비스타, 윈도우 XP, 윈도우 NT 4.0, 윈도우 ME, 윈도우 2000, 윈도우 98, 윈도우 95, 혹은 리눅스 운영 시스템을 포함한다. 예시적인 그래픽 라이브러리는 마이크로소프트 DirectX 라이브러리와 OpenGL 라이브러리들을 포함할 수 있다. 컴퓨터 메모리(도1) 및 상호연결된 디스크 드라이브들(미도시)은 랜덤 액세스 메모리, 판독-전용 메모리, 디스크 저장 메모리의 적절한 조합을 포함하는바, 이들 조합은 운영 시스템과 본 발명의 일실시예에 따른 방식으로 디바이스(10)를 조정하는 그래픽 어댑터 드라이버 프로그램을 저장 및 실행하도록 디바이스(10)에 의해 이용된다. 예시적인 소프트웨어(100)(도2)는 예를 들면, 판독-전용 메모리에 저장될 수도 있으며 혹은 가령, IDE/SATA 포트(24)와 통신하는 디스크 드라이브와 같은 외부 주변기기들로부터 로딩될 수도 있다. 컴퓨터 판독가능 매체(27)는 광 저장 매체, 자기 디스크, 테이프, ROM 카트리지 등등이 될 수 있다.

[0033] 그래픽 라이브러리(104) 혹은 운영 시스템(102)은 또한, 그래픽 드라이버 소프트웨어(108)를 포함하는바, 이는 그래픽 어댑터(34, 52)와의 저-레벨 통신을 위해 이용된다. 상기 소프트웨어는 계층화되는바, 상위 레벨 계층들

과 함께 하위 계층들을 이용하여 소정의 기능들을 제공한다. 어플리케이션들은 운영 시스템(102)과 그래픽 라이브러리(104)를 이용하여 2D 혹은 3D 그래픽을 렌더링할 수 있다.

[0034] 본 명세서에서 렌더링(rendering)이라는 용어는 드로잉(drawing)을 포함하며 혹은, 표시를 위해서 그래픽 이미지를 생성하는 것을 포함한다. 그리고 렌더링은 폴리곤 렌더링(polygon rendering), 레이-트레이싱(ray-tracing), 이미지 디코딩, 라인 드로잉(line drawing) 등을 포함할 수 있다. 다음에 설명되는 바와 같이, 본 발명에 따른 예시적인 소프트웨어 일례는, 그래픽 라이브러리(104) 및/또는 드라이버 소프트웨어(108)의 일부를 구성할 수도 있다. 본 발명의 일 실시예에서, 예시적인 소프트웨어는 OpenGL 라이브러리와 관련 드라이버들의 일부를 구성할 수도 있다. OpenGL에 대해서는, The Official Guide to Learning OpenGL 버전 1.1에 상세히 설명되어 있는바, 그 내용은 본 발명에 대한 참조로서 본 명세서에 통합된다. 하지만, 해당 기술분야의 당업자라면, 본 발명의 일 실시예에 따른 예시적인 방법들이, 마이크로소프트 Direct3D 라이브러리의 일부, 커스텀 라이브러리 혹은 루틴을 이용하지 않는 어플리케이션(가령, 게임)의 일부 혹은 임의의 다른 3D 어플리케이션 혹은 라이브러리의 일부를 구성할 수도 있다는 점을 능히 이해할 수 있을 것이다.

[0035] 통합된 그래픽 어댑터(34)를 도시하고 있는, 예시적인 컴퓨팅 디바이스(10)(특히 노스 브리지 14)의 일부분에 대한 간략화된 또 다른 블록도가 도3에 예시된다. 도시된 바와 같이, 노스 브리지(14)는 고속 버스 인터페이스(36), 메모리 제어기(38), 상호연결 버스(22)로의 상호연결 브리지(14) 및 시스템 메모리(16)를 포함한다. 또한, 노스 브리지(14)는 통합된 그래픽 어댑터(34)를 포함하는바, 그래픽 어댑터(34)는 그래픽 프로세서(40), 커맨드 프로세서(42) 그리고 디스플레이 인터페이스(44)를 포함한다.

[0036] 그래픽 프로세서(40)는 시스템 메모리(16)에 할당된 프레임 버퍼(46) 내의 3차원 이미지들을 렌더링할 수 있는 3차원(3D) 그래픽 프로세서가 될 수도 있으며 그리고 그래픽들의 가속화된 렌더링을 위한 다수의 파이프라인들 및 파이프라인 스테이지를 포함할 수도 있다.

[0037] 정상적인, 싱글 모니터 동작에서는, 커맨드들은 통상적인 방식으로 커맨드 프로세서(42)에 의해서 그래픽 프로세서(40)로 전달된다. 예를 들어, 커맨드 프로세서(42)는 메모리(16) 내의 커맨드 큐(47)를 정의하는데 이용되는 레지스터를 포함할 수도 있다. 어플리케이션 소프트웨어(106) 혹은 운영 시스템(102)은 라이브러리(104)를 이용하여 그래픽 이미지들을 렌더링한다. 드라이버 소프트웨어(108)는 그래픽 프로세서에 특화된 커맨드들을 생성하며 그리고 이들을 커맨드 큐(47)에 위치시키며, 또한 상기 큐의 종료(end)를 반영하기 위해서 커맨드 프로세서(42)의 레지스터를 업데이트한다. 그래픽 프로세서(42)는 대기된(queued) 커맨드들을 비동기적으로, 파이프라인화된 방식으로 처리하는바, 이는 다른 무엇보다도 프레임 버퍼(46) 내의 이미지들을 렌더링하기 위함이다. 그래픽 프로세서(40)가 큐 내의 다음 커맨드를 처리할 수 있게되면, 그래픽 프로세서(40)는 커맨드 프로세서(42)에게 신호를 제공하며, 이는 큐(47) 내의 다음 커맨드를 폐치하고 그리고 커맨드 프로세서(42)의 레지스터에 있는 큐 포인터를 진전시킨다(advance).

[0038] 디스플레이 인터페이스(44)는 메모리 제어기(38)를 통하여 프레임 버퍼(46)를 샘플링하며 그리고 하나 이상의 비디오 출력 포트(48) 상에 이미지를 제공한다. 상기 비디오 출력 포트는 VGA 포트, 콤포지트 비디오 포트, DVI 포트, HDMI, DisplayPort 등이 될 수 있으며 이는 텔레비전, 모니터 등과 같은 도1의 디스플레이 기기(32) 상에 하나 이상의 비디오 이미지를 디스플레이하기 위한 것이다. 이러한 방식으로, 그래픽 프로세서(40)에 의해 렌더링되어 프레임 버퍼(46)에 있는 이미지들이 표시될 수 있다.

[0039] 상기 디스플레이 인터페이스(44)는, 디스플레이 기기 상에 디스플레이되도록 버퍼 내의 데이터를 변환하기에 적절한 임의의 인터페이스가 될 수 있다. 예를 들어, 디스플레이 인터페이스(44)는 RAMDAC 형태를 취할 수도 있다. 일반적으로, 디스플레이 인터페이스(44)는 예를 들면, 다수의 레지스터들을 통해서 프로그램 가능한바, 이는 드라이버 소프트웨어(108) 혹은 유사한 소프트웨어 혹은 그래픽 프로세서(40)가 그것의 디스플레이 출력에서 표시되도록 메모리(16) 내의 시작 어드레스를 설정할 수 있게한다. 마찬가지로, 인터페이스(44) 꽂셀 깊이(즉, 꽂셀 당 비트)와 스크린 해상도도 프로그램 가능하다.

[0040] 이해되는 바와 같이, 메모리(16) 내에 하나의 프레임 버퍼(46)를 할당하는 대신에, 2개의 프레임 버퍼들이 할당될 수도 있다. 이러한 버퍼들은, 렌더링된 이미지들을 좀더 부드럽게(smoothly) 표현하도록 교번적으로 이용될 수 있다. 디스플레이 인터페이스(44)는 임의의 시간에서 2개의 버퍼들 중 하나의 내용만을 디스플레이한다. 디스플레이되고 있는 버퍼는, 통상적으로 프론트(front) 버퍼라고 지칭된다. 그래픽 프로세서(40)는 소위 백(back) 버퍼라고 지칭되는 버퍼에 대해 디스플레이를 위해 이미지들을 렌더링한다. 상기 백 버퍼는 디스플레이 인터페이스(44)에 의해서 현재 디스플레이되고 있지 않은 프레임 버퍼이다. 프레임에 대한 렌더링이 완료되면, 프론트 버퍼와 백 버퍼는 뒤바뀌며(flipped), 디스플레이 인터페이스(44)는 예전의 백 버퍼를 표시하도록 프로

그랩된다. 이러한 방식으로, 그래픽 프로세서(40)에 의해 렌더링된 연속적인 프레임들 사이에서의 트랜지션(transition)은 좀더 부드럽게 보여질 수 있다.

[0041] 특히, 어댑터(34)에 의해 이용되는 레지스터들 및 프레임 버퍼 메모리(46)는 모두 글로벌 메모리 공간내의 어드레스에 매핑된다.

[0042] 하지만, 본 발명의 일실시예에서, 컴퓨팅 디바이스(10)는 또한 주변 확장 카드(50) 상에 형성된 그래픽 어댑터(52)를 포함하는바, 이는 도1에 도시된 바와 같이 노스 브리지(14)와 통신하는 고속의 상호연결 버스(22) 상의 확장 슬롯을 통해서 디바이스(10)에 삽입될 수도 있다. 노스 브리지(14)에 상호연결된 그래픽 어댑터(52)가 도4에 개략적으로 도시된다. 도시된 바와 같이, 그래픽 어댑터(52)는 그래픽 프로세서(54), 비디오 메모리(56), 커맨드 프로세서(58), 메모리 제어기(60), 그리고 디스플레이 인터페이스(62)를 포함한다. 하나 이상의 포트들(64, 66)은 도1의 디스플레이 기기(68)와 같은 하나 이상의 디스플레이 디바이스들과 어댑터(52)를 상호연결 시킬 수 있다.

[0043] 커맨드 프로세서(58) 그리고 디스플레이 인터페이스(62)는 각각 커맨드 프로세서(42)와 디스플레이 인터페이스(44)와 거의 마찬가지로 기능한다. 그래픽 프로세서(54)는 통상 동작 모드에서 그래픽 프로세서(40)와 거의 마찬가지로 동작한다. 그래픽 어댑터(52)는 그 자신의 메모리(56)를 포함하며, 메모리(56)와 시스템 메모리(16)로의 액세스를 제공하는 메모리 제어기(60)를 포함한다. 로컬 메모리(16)는 다른 것들 중에서, 그래픽 프로세서(54)에 의해 이용되는 그래픽 어댑터(52)를 위해 하나 이상의 버퍼(70)를 보유하는데 이용되며, 이는 프레임 버퍼로서 작용할 수도 있다. 그래픽 프로세서(54)를 위한 커맨드 큐(57)가 또한 시스템 메모리(16)에 설정된다.

[0044] 다시 한번 설명하면, 로컬 메모리(56)는 메모리 제어기(60)를 통해, 글로벌 시스템 메모리 공간내의 어드레스들에서 어드레스 가능하다. 마찬가지로, 커맨드 프로세서(54)에 의해 이용되는 레지스터들과 디스플레이 인터페이스(62)는 글로벌 어드레스 공간내의 어드레스들에서 어드레스 가능하다. 앞서 언급한 바와 같이, 어댑터(34)에 의해 이용되는 레지스터들과 프레임 버퍼(46)는, 글로벌 메모리 공간에 매핑된다. 따라서, 커맨드 프로세서(58)와 그래픽 프로세서(54)는, 디스플레이 인터페이스(44)와 커맨드 프로세서(42)에 의해 이용되는 프레임 버퍼(46) 및 레지스터들에게 기입할 수도 있다.

[0045] 또한, 그래픽 프로세서(54)는 메모리 제어기(60)와 함께 블록 전송 동작을 수행할 수 있어, 메모리 어드레스에서 시스템 및 어댑터 메모리(16, 56)로부터 그리고 시스템 및 어댑터 메모리(16, 56)로 데이터를 전송할 수 있다.

[0046] 통합된 구성요소들을 저렴하게 제공하고자 하는 노력으로, 통합된 그래픽 어댑터(34)는 제한된 기능을 제공한다. 예를 들어, 통합된 그래픽 제어기의 해상도, 3D 그래픽 능력 등을 비교적 제한된다. 따라서, 통합된 그래픽 어댑터(34)는 그래픽 어댑터(52) 보다 상당히 뒤떨어지는 그래픽 처리 능력을 갖는 것이 일반적이다. 예컨대, 어댑터(34)는 어댑터(52)보다 더 느리게 동작할 수 있으며 혹은 더 적은 피쳐들(features)을 제공하는 등의 저하된 능력을 가질 수 있다.

[0047] 이제, 통합된 어댑터(34) 혹은 어댑터(52)는, 포트(48)과 같은 다중 포트들을 통해 다수개의 물리적인 디스플레이 디바이스들의 상호연결을 저절로(by itself) 허용할 수도 있으며 혹은 허용하지 않을 수도 있다. 하지만, 각각의 어댑터는 적어도 하나의 물리적인 디스플레이와 상호연결될 수도 있다. 따라서, 본 발명의 일실시예에서, 메모리(16) 내에 로딩된 어댑터 드라이버 소프트웨어(108)는, 그래픽 어댑터(52)로 하여금 2개의 디스플레이(32, 68) 둘다에서 표시되도록 그래픽을 렌더링하는 마스터 그래픽 어댑터(혹은 제 1 그래픽 어댑터)로서 행동하도록 야기할 수 있으며, 그리고 그래픽 어댑터(34)로 하여금 제 2 어댑터(혹은 슬레이브 어댑터)로서 행동하도록 야기할 수 있다.

[0048] 이해되는 바와 같이, 이러한 마스터/슬레이브 동작 모드에서, 통합된 그래픽 어댑터(34)는 실질적으로 디스에이블된다. 하지만, 온-보드(on-board) 그래픽 어댑터(34)의 구성요소들 (그리고 특히 디스플레이 인터페이스 44)는, 상호연결된 모니터들 및 다른 디스플레이 디바이스들을 구동하는데 여전히 이용될 수도 있다. 이와 같이, 카드(50)는 다수의 디스플레이 포트들 및 디스플레이 드라이버들을 포함할 필요가 없다. 대신에, 디스플레이 인터페이스(44)와 그것의 포트(들)(48)이, 디스플레이 기기(68) 혹은 카드(50)에 의해 구동되는 디스플레이 기기들 이외에도 적어도 하나의 모니터 혹은 디스플레이(32)를 구동하는데 이용될 수 있다. 만일, 어댑터(52)가 다중 포트들(가령, 포트 64, 66)을 포함한다면, 프로세서(52)는 이들 다중 포트들을 통해 프리젠테이션 하기위해서 이미지들을 추가적으로 렌더링할 수 있다. 물론, 드라이버 소프트웨어(108)는 전술한 바와 같은 통상적인 동작 모드에서 어댑터(34)의 동작을 또한 허용할 수도 있다. 어댑터(34)(및 그러므로 어댑터 52)의 동작 모드를

선택하는 것은, 어플리케이션 소프트웨어(106)에 의해서 혹은 어플리케이션 소프트웨어(106)을 통해서 이루어질 수도 있다.

[0049] 특히, 본 발명의 일실시예의 마스터/슬레이브 동작 모드에서, 메모리(16) 내의 드라이버 소프트웨어(108)는 도4에 도시된 바와 같은 액티브 프레임 버퍼(70)와 어댑터(52)의 메모리(56) 내의 보조 어댑터 버퍼(72)를 할당한다. 액티브 프레임 버퍼(70)는 어댑터(52)를 위한 통상적인 프레임 버퍼로서 기능하며, 따라서 어댑터(52)의 디스플레이 기기(68) 상에서 디스플레이하기 위해 프레임들을 저장한다. 이와 같이, 그래픽 프로세서(54)는 메모리(56) 내의 액티브 프레임 버퍼(70)의 2-D 혹은 3-D 그래픽들을 통상적인 방식으로 렌더링한다. 카드(50)의 디스플레이 인터페이스(62)는 상호연결된 모니터 혹은 디스플레이 디바이스 상에서 디스플레이하도록, 액티브 프레임 버퍼(70)의 내용물들을 변환한다. 선택적으로는, 액티브 프레임 버퍼(70)는 전술한 바와 같이 프론트 및 리어 버퍼로 대체될 수도 있다.

[0050] 그래픽 프로세서(54)는 디스플레이 디바이스(32) 상에 디스플레이하기 위해서 보조 어댑터 프레임 버퍼(72)내의 이미지들을 또한 렌더링한다. 보조 어댑터 프레임 버퍼(72) 내의 프레임에 대한 렌더링이 종료하는 때, 상기 보조 어댑터 프레임 버퍼(72)의 내용물은, 그래픽 어댑터(34)의 디스플레이 인터페이스(44)에 의해 디스플레이되기 위하여 프레임 버퍼(46)로 전송된다. 이에 대해서는 후술된다.

[0051] 제 1 실시예에서, 메모리(16) 내의 그래픽 어댑터 드라이버 소프트웨어(108)는 도5A에 도시된 단계 S500을 수행한다. 특히, 드라이버 소프트웨어(108)는 커맨드들을 생성하는바, 상기 커맨드는 그래픽 프로세서(54)를 지시하여 단계 S502에서 메모리(56)의 보조 어댑터 버퍼(72)를 렌더링하게 한다. 이들 커맨드들은 어댑터(52)의 커먼드 큐(57)에 배치되며 그리고 그래픽 프로세서(54)에 의해 실행되는바, 전술한 바와 같이 큐(47)에 대기될 커맨드들과 매우 유사한 방식으로 실행된다. 일단 프레임이 렌더링되면, 드라이버 소프트웨어(108)는, 프로세서(54)로 하여금 메모리(56) 내의 보조 어댑터 프레임 버퍼(72)의 내용물을 그래픽 어댑터(34)를 위한 프레임 버퍼(46)에 할당된 메모리(16)의 그 영역으로, 비트 블록 트랜스퍼(bit block tranfer : BITBLT)하게 하는 커맨드(혹은 커맨드들)을 더 생성한다.

[0052] 또한, 그래픽 어댑터(34)의 디스플레이 인터페이스(44)는, 그것의 프레임 버퍼(46)의 내용물을 포트(48)에 상호연결된 모니터 혹은 디스플레이 기기에서 디스플레이 하도록 사전-프로그래밍되었다. 이후, 편리하게도, 포트(48)와 포트(64)에 상호연결된 디스플레이 기기(32, 68) 상에 디스플레이되는 이미지들이 그래픽 프로세서(54)에 의해서 렌더링 혹은 드로잉된다. 버퍼(72) 및 프레임 버퍼(46)에 대한 단계 S504의 효과가 도5B에 도시된다.

[0053] 제 2 실시예에서는, 도6A에 도시된 단계 S600이 수행된다. 초기에는, 프론트 프레임 버퍼와 백 프레임 버퍼가 어댑터(52)의 메모리(16) 내에 할당된다. 편의를 위해서, 이들 버퍼들은 도6에 도시된 바와 같이 버퍼(46a)와 버퍼(46b)로 지칭된다. 단계 S502와 관련하여 앞서 설명된 것처럼, 단계 S602에서 프로세서(54)는, 디스플레이 인터페이스(44)에 의해 표현되도록, 메모리(56)의 보조 어댑터 프레임 버퍼(72) 내의 임의의 프레임을 또한 렌더링한다.

[0054] 디바이스(32) 상에서 디스플레이 하기 위해 프레임을 렌더링하는 것이 끝나면, 드라이버 소프트웨어(108)는 그래픽 프로세서(54)를 프로그램하여, 메모리(56) 내의 보조 어댑터 프레임 버퍼(72)의 내용물을 그래픽 어댑터(34)를 위한 메모리(16) 내의 그 당시의 백 버퍼로 비트 블록 트랜스퍼(BITBLT)하게 한다(단계 S604). 비트 블록 트랜스퍼(BITBLT)가 완료되면, 드라이버는 프로세서(54)를 프로그램하여 어댑터(34)의 백 버퍼와 프론트 버퍼가 뒤바뀌도록(flip)(즉, 백 버퍼를 프론트 버퍼로서 이용함) 디스플레이 인터페이스(44)의 레지스터들을 프로그램한다(단계 S606). 이러한 것은 데이터의 프리젠테이션을 위해 디스플레이 인터페이스(44)에 의해 이용되는 시작 어드레스를 식별하는 디스플레이 인터페이스(44)의 레지스터를 직접 재프로그래밍함으로써 수행될 수도 있다. 이제, 그래픽 어댑터(34)의 디스플레이 인터페이스(44)는 그것의 버퍼의 내용물을 포트(48)에 상호연결된 모니터(즉, 디스플레이 68)에 제공한다. 이전의 프론트 버퍼는 이제 어댑터(34)를 위한 백 버퍼로서 이용되며, 그리고 디스플레이(32) 상에 디스플레이되기 위해 프로세서(54)에 의해 렌더링되는 다음 프레임은 상기 백 버퍼로 처음에 전송될 것이다(단계 S604). 물론, 드라이버 소프트웨어(108)는 각 버퍼(46a, 46b)들의 시작 위치 및 상태를 프론트 버퍼 및 백 버퍼로서 유지할 것이다. 버퍼들(46a, 46b)에 대한 단계 S604 및 단계 S606의 효과가 도6B에 도식적으로 도시된다.

[0055] 동기화 없는, 단일 버퍼(46)의 사용 혹은 어댑터(34)의 디스플레이 인터페이스(44)의 직접 프로그래밍은 눈에 보이는 티어링(tearing)을 야기할 수도 있다. 즉, 디스플레이 인터페이스(44)의 레지스터들이 재프로그래밍되는 수직 귀선 기간(vertical blanking interval) 바깥의 임의의 시간에서 티어링이 보일 수도 있는데, 이는 인터페이스(44)에 의해 출력되는 프레임의 중간에서 디스플레이 파라미터들이 변화하기 때문이다.

[0056]

따라서, 본 발명의 제 3 실시예에서는, 도7에 도시된 단계 S700이 수행될 수도 있다. 다시 한번, 처음에 프론트 버퍼와 백 버퍼(46a, 46b)가 어댑터(34)에 의해 이용되는 메모리(16) 내에 할당된다. 또한, 단계 S502 및 단계 S602를 참조하여 앞서 설명된 바와 같이, 프로세서(54)는 디바이스(32)에서의 디스플레이를 위해서 로컬 메모리(56)의 버퍼(72) 내에 이미지들을 렌더링한다.

[0057]

디스플레이 기기(32) 상에서의 디스플레이를 위해 프레임을 렌더링하는 것이 완료되면, 메모리(16) 내의 드라이버 소프트웨어(108)는 그래픽 프로세서(54)를 프로그램하여 메모리(56) 내의 보조 어댑터 프레임 버퍼(72)의 내용물을 그래픽 어댑터(34)의 메모리(16) 내의 그 당시 백 버퍼로 비트 블록 트랜스퍼한다(BITBLT)(단계 S704). 또한, 드라이버 소프트웨어(108)는 어댑터(34)의 커맨드 큐(47) 내에 커맨드를 위치시키 프론트 버퍼와 백 버퍼를 뒤바꾼다(단계 S706). 하지만, 커맨드 프로세서(42)의 레지스터들은, 커맨드 큐(47) 내의 웨딩 커맨드들을 반영하기 위해서 드라이버 소프트웨어(108)에 의해 업데이트되지 않는다. 대신에, 드라이버 소프트웨어(108)는, 단계 S708에서 그래픽 프로세서(54)에게 커맨드를 제공하여, 단계 S706에서 개시된 비트 블록 트랜스퍼가 완료되면 커맨드 프로세서(42)의 커맨드 큐 레지스터들을 업데이트한다. 이것은 또한, 커맨드 프로세서(42)로 하여금 명령들을 프로세서(40)에게 제공하여, 그 백 버퍼와 프론트 버퍼가 뒤바뀌게 한다. 그래픽 프로세서(40)는 대기행렬된(queued) 커맨드를 실행하여 어댑터(34)의 백 버퍼와 프론트 버퍼를 뒤바꾼다(즉, 디스플레이 인터페이스 44의 레지스터들을 재프로그래밍함으로써 이전의 백 버퍼의 내용물을 포트 48에 상호연결된 모니터 혹은 디스플레이에 제공한다).

[0058]

본 발명의 제 4 실시예에서는 도8에 도시된 단계 S800이 수행된다. 다시 한번, 처음에 프론트 버퍼와 백 버퍼가 어댑터(34)의 메모리(16) 내에 할당된다. 또한, 전술한 바와 같이, 프로세서(54)는 디스플레이(32)를 위해 로컬 메모리(56)의 버퍼(72) 내의 이미지들을 렌더링한다.

[0059]

또한, 드라이버 소프트웨어(108)는 어댑터(34)의 큐(47) 내에 커맨드를 위치시키는바, 따라서 그래픽 프로세서(40)는 메모리(56)의 보조 프레임 버퍼(72)의 내용물을, 그래픽 어댑터(34)를 위해 현재의 백 프레임 버퍼로서 할당되어 있는 시스템 메모리(16)의 바로 그 영역으로 비트 블록 트랜스퍼한다. 또한, 드라이버 소프트웨어(108)는 큐(47) 내에 커맨드를 위치시키는바, 후위(rear) 버퍼를 프론트 버퍼로 사용하도록 디스플레이 인터페이스(44)를 프로그래밍함으로써, 그래픽 프로세서(40)로 하여금 어댑터(34)의 프론트 버퍼와 백 버퍼를 뒤바뀌게 한다. 하지만, 큐 포인터를 정의하는 커맨드 프로세서(42)의 레지스터들은 프레임이 버퍼(72)에서 렌더링될 때까지 업데이트되지 않는다.

[0060]

따라서, 디스플레이(32)를 위해 프레임을 렌더링하는 것이 끝나면, 그래픽 프로세서(54)는 큐 포인터를 정의하는 커맨드 레지스터들을 업데이트하도록 커맨드를 제공하는바, 따라서 커맨드 프로세서(42)는 어댑터(34)의 커맨드 큐 내에 있는 계류중인 2개의 커맨드를 반영할 수 있다. 다음으로, 그래픽 프로세서(40)는 대기행렬된(queued) 커맨드들을 실행하여, 버퍼(72)의 내용물을 그것의 백 버퍼로 비트 블록 트랜스퍼하며 그리고 그것의 백 버퍼와 프론트 버퍼를 뒤바꾼다(즉, 이전의 백 버퍼의 내용물을 포트 48에 상호연결된 모니터 혹은 디스플레이에 제공하기 위해서 디스플레이 인터페이스 44의 레지스터들을 재프로그래밍함으로써).

[0061]

이해되는 바와 같이, 전술한 바와 같은 실시예들은, 2차원 혹은 3차원 그래픽 이미지들을 다수개의 독립적인 프레임 버퍼들 상에 드로잉(drawing)하는 것을 마스터 그래픽 프로세서가 책임질 수 있게 한다. 슬레이브 그래픽 어댑터는, 하나의 프레임 버퍼에서 생성된 이미지들을 디스플레이하는데 이용될 수 있다. 추가적인 디스플레이들을 구동시키는데 요구되는 디바이스 전자기기들 및 포트들은, 슬레이브 그래픽 어댑터(34)에 의해 제공될 수도 있다. 이러한 방식으로, 슬레이브 그래픽 제어기에 포함된 전자기기들이 효과적으로 사용되어, 다수개의 디스플레이들에 대한 구동을 허용할 수 있다.

[0062]

편리하게도, 버퍼(70, 72)들이 이용되어 단일 표면(single surface)을 정의 할 수도 있는데, 상기 단일 표면 상에서 어플리케이션 소프트웨어(106)는 싱글 그래픽 이미지를 렌더링하여, 상기 싱글 그래픽 이미지가 그래픽 어댑터(52, 34)에 상호연결된 2개 이상의 디스플레이들(48, 64)을 통해 디스플레이되도록 할 수도 있다. 따라서, 드라이버 소프트웨어(108)는, 버퍼(70) 및 버퍼(72)의 사이즈와 동일한 사이즈인 이용가능한 스크린 사이즈를 어플리케이션 소프트웨어(106)에게 알려줄 수도 있다. 만일, 버퍼(70)가  $m_1 \times n$  의 해상도를 갖고 그리고 버퍼(72)가  $m_2 \times n$  의 해상도를 갖는다면, 드라이버 소프트웨어(108)는 이용가능한 스크린 사이즈인  $(m_1 + m_2) \times n$  를 어플리케이션 소프트웨어(106)에게 알려줄 수 있다. 만일, 버퍼(70)와 버퍼(72)가 메모리(56) 내의 인접한  $(m_1 + m_2) \times n$  픽셀 위치들을 점유한다면, 드라이버 소프트웨어(108)에 의한 렌더링은 간략화될 수 있다. 어플리케이션 소프트웨어(106)는 드라이버 소프트웨어(108)를 이용하여, 그래픽 프로세서(40)가 버퍼(70) 및 버퍼(72)에

의해 정의되는 표면(surface)을 구성하는  $(m_1 + m_2) \times n$  픽셀의 해상도를 갖는 더 큰 이미지들을 렌더링하게 할 수 있다. 따라서, 버퍼(70)와 버퍼(72)는 이미지 쌍들을 저장하며, 상기 이미지 쌍의 각각의 이미지는 어플리케이션에 의해 렌더링되는 더 큰 이미지의 일부를 구성한다. 각각의 더 큰 이미지에 대한 렌더링이 끝나면, 드라이버 소프트웨어(108)는 전술한 바와 같은 단계 S500, S600, S700, 혹은 S800에 따라, 디스플레이(32) 상에서의 표시를 위해 버퍼(72) 내의 이미지를 프레임 버퍼(46)로 전송할 수 있다. 버퍼(70) 내의 이미지는 통상적인 방식으로 디스플레이(68) 상에 표시될 수도 있다. 나란히 있는 디스플레이(32)와 디스플레이(68)를 바라보는 최종 사용자는 어플리케이션(106)에 의해 렌더링된 더 큰 이미지를 인식한다. 이해되는 바와 같이, 게임과 같은 유형의 어플리케이션(106)은 수정을 필요로하지 않고서도 다중 디스플레이(multiple display)의 장점을 취할 수 있을 것이다.

[0063] 만일, 프론트 버퍼와 백 버퍼가 디스플레이(68)를 위해 이용된다면, 상이한 2개의 버퍼들이 버퍼(72) 대신에 할당 및 이용될 수 있을 것인바, 이는 디스플레이(32) 상에 표시될 표면 부분(surface portion)을 홀딩하고 있는 버퍼와 디스플레이(68) 상에 표시될 표면 부분이 렌더링 중인 소정 버퍼가 메모리(56)에서 인접하다는 것을 보장하기 위해서이다. 따라서, 방금 완료된 이미지를 홀딩하고 있는 버퍼의 내용물을 백 버퍼(46a, 혹은 46b)로 비트 블록 트랜스퍼하도록, 단계 S602, S604, S702, S704, S802, S804가 수정될 수 있다. 편리하게도, 기존의 어플리케이션 소프트웨어(106)는 수정될 필요가 없다. 대신에, 상기 어플리케이션 소프트웨어는 보고된 더 큰 스크린 사이즈에 단지 의존하기만 하면 된다. 드라이버 소프트웨어(108)는 다중 디스플레이들의 통한 표면들의 렌더링을 야기한다.

[0064] 또한, 이해되는 바와 같이, 도시된 실시예에서는 통합 그래픽 제어기가 슬레이브 그래픽 어댑터로 이용되었지만, 슬레이브 그래픽 어댑터는 충분한 대역폭을 갖는 버스를 통해 그래픽 어댑터(52)와 통신하는 또 다른 그래픽 어댑터가 될 수도 있으며, 여기서 충분한 대역폭을 갖는 상기 버스는 슬레이브 프레임 버퍼(들)로 그래픽 프레임들을 전송할 수 있게 한다. 이와 같이, 슬레이브 그래픽 프로세서는 또 다른 PCIe 호환 확장 카드로 구성될 수 있다. 따라서, 예를 들어 그래픽 카드들이 업그레이드 되는 경우, 오래된 확장 카드들도 여전히 이용될 수 있는바, 이는 상호연결된 디스플레이 기기들을 구동시킬 수 있는 이들의 능력 때문이다. 이와 유사하게, 상호연결 버스도 PCIe 상호연결 버스가 될 필요는 없으며, 디스플레이 인터페이스(44)에 의해 이용되는 리프레시 속도와 동일한 속도로, 프레임 버퍼들 사이에서 데이터를 전송할 수 있는 대역폭을 갖는 임의의 적절한 버스이기만 하면된다. 이와 유사하게, 비록 프레임 버퍼들(46)(그리고 프론트 및 백 버퍼(46a, 46b))이 시스템 메모리(16) 내에 구성된다라고 설명되었지만, 어댑터(34)에 대해 로컬인 메모리의 일부로서 용이하게 구성될 수도 있다.

[0065] 본 발명의 또 다른 대안적인 실시예에서, 버스(22)를 통한 렌더링은, 고속 버스(22)에 의해 상호연결된 그래픽 어댑터(혹은 서브시스템)로 하여금 통합 어댑터/서브시스템의 자리를 차지하게 할 수도 있다.

[0066] 이를 위한 도9는 본 발명의 또 다른 실시예에 따른 컴퓨팅 디바이스(10')의 일부를 예시한 간략화된 블록도이다. 상기 컴퓨팅 디바이스(10')는 미국특허출원(출원번호 11/421,005)에 상세히 설명되어 있다.

[0067] 간략히 말해, 컴퓨팅 디바이스(10')는 2개의 그래픽 서브시스템(30', 40')을 포함한다. 그래픽 서브시스템(30')은 그래픽 엔진/프로세서(32'), 메모리 제어기(72'), 디스플레이 인터페이스(74') 및 버스 인터페이스(78')를 포함한다. 그래픽 서브시스템(40')은 PCIe 버스와 같은 고속의 버스(22')를 통해 그래픽 서브시스템(30')과 통신한다. 제 2 그래픽 서브시스템(40')은 그 자신의 그래픽 엔진/프로세서(42'), 메모리 제어기(52'), 디스플레이 인터페이스(54')를 포함한다. 또한, 그래픽 서브시스템(40')은 그래픽 메모리(50')와도 통신한다. 이해되는 바와 같이, 서브시스템(40')은 버스(22')를 통해 메모리(14')에 그래픽들을 렌더링하도록 되어있다.

[0068] 디바이스(40')는 랩탑 혹은 더 작은 컴퓨팅 디바이스의 형태인 휴대용 컴퓨팅 디바이스로 형성될 수도 있다. 따라서, 하나의 하우징(housing)이 DC 파워 소스, 디스플레이(26') 그리로 전술한 바와 같은 마더보드 및 구성요소들을 포함할 수도 있다. 제 2 그래픽 서브시스템(40')은 상기 컴퓨팅 디바이스의 나머지들을 감싸고 있는 단일 하우징에 부가될 수도 있으며, 혹은 디바이스(10')가 거기에 물리적으로 연결되는 때에 디바이스(10')의 오직 일부만을 구성하는 도킹 스테이션의 일부를 구성할 수도 있다.

[0069] 디바이스(10')는 적어도 2개의 전력 소비 모드에서 작동할 수 있다. 높은 전력 소비 모드와 낮은 전력 소비 모드가 그것이다. 도시된 실시예에서, AC(메인) 전원에 연결된 전력원에 의해 디바이스(10')에 전원이 공급된다면 높은 전력 소비 모드일 것으로 추정되며 그리고 하나 이상의 배터리, 연료전지(fuel cell) 등을 이용하는 DC 전력원에 의해서 전원이 공급된다면 낮은 전력 소비 모드일 것으로 추정된다. 대안적으로는, 예컨대 사용자의 선호도, 실행되고 있는 어플리케이션의 유형, 배터리 레벨 등등에 기초하여 전력 소비 모드가 사용자에 의해 선

택될 수도 있고 혹은 소프트웨어에 의해 제어될 수도 있으며 또는 다른 방법으로 선택될 수도 있다.

[0070] 디바이스(10')에 대한 소프트웨어 제어 동작의 세부내용은 미국특허출원(출원번호 11/421,005)에 개시되어 있으며, 이는 전술한 소프트웨어(100)와 유사성을 공유한다.

[0071] 가령, 디바이스(10')와 같은 디바이스(10')의 소프트웨어 제어 동작의 일부분들은 높은 전력 소비 상태와 낮은 전력 소비 상태 사이에서 전환된다. 특히, 도10에는 본 발명의 예시적인 실시예에 대한 소프트웨어 블록(S80 0')이 도시되어 있는데, 이는 디바이스(10')의 시스템 메모리 내의 소프트웨어의 제어 하에서 중앙 프로세서(예컨대, 도1의 프로세서 12)에 의해 실행될 수 있다. 디바이스(10')가 상태 변화를 겪을 때마다 블록 S800'가 수행될 수도 있는데, 이를 위해서 서브시스템(30')과 서브시스템(40')은 적절히 구성되어야 한다. 예시된 바와 같이, 블록 S802'에서, 소프트웨어는 디바이스(10')가 높은 전력 소비 모드를 취할지 혹은 낮은 전력 소비 모드를 취할지를 판단한다.

[0072] 디바이스(10')가 높은 전력 소비 모드로 재개(혹은 전이)하는 경우에는 블록 S804' - S810'이 실행된다. 블록 S804에서, 그래픽 서브시스템(40')이 높은 전력 소비 모드로 이미 설정되어 있지 않았다면, 그래픽 서브시스템(40')은 완전 동작 모드(full operation mode)(높은 전력 소비 모드)에 있게 된다. 이러한 것은, 드라이버 제어 그래픽 서브시스템(40')을 통해 적절한 신호를 전력 제어기(60')에 제공함으로써 수행될 수 있다. 다음으로, 그래픽 서브시스템(40')이 블록(S806, S808)에서 인에이블된다. 다시 한번, 이러한 것은 그래픽 서브시스템(30')에 관계되어 연결된 임의의 디스플레이를 논리적으로 디스에이블시키고(블록 S804) 그리고 그래픽 서브시스템(40')에 연결된 디스플레이를 논리적으로 인에이블(블록 S808)시킴으로써 실행될 수도 있다. 가령, 미국특허출원(출원번호 11/421,005)에 기술된 바와 같은 EnumDisplayDevices() 호출 및 ChangeDisplaySettingsEX() 호출과 같은 적절한 운영 시스템 API 호출에 의해서 혹은 하드웨어와의 직접 통신을 통해 블록 S806 및 S808이 다시 수행될 수도 있다.

[0073] 주목할만한 점으로, 그 어떤 물리적인 디스플레이도 그래픽 서브시스템(40')에 연결되어 있지 않다. 그래픽 서브시스템(40')의 드라이버 소프트웨어 제어 동작은, 관련 메모리(50') 내의 이미지를 대신에 그래픽 서브시스템(30')의 버퍼(14') 내의 이미지들을 렌더링하도록 구성된다(단계 S810'). 편리하게도, 고속의 버스(22')의 존재 하에서는(예컨대, PCIe 버스로 구현됨), 이러한 렌더링이 버스(22')를 통해 가능한바, 이는 부분적으로는 상기 버스에 의해 가능해진 전송 속도 때문이다.

[0074] 상기 렌더링은 버스를 통해 프리미티브 단위로(primitive by primitive) 프레임 버퍼 안으로 향할 수 있으며, 따라서 렌더링된 이미지를 버스(22')를 통해 전송할 수 있다. 이러한 것은, 서브시스템(40')에 의해 액세스가 가능한 메모리 내에 버퍼(14')를 할당하고 그리고 서브시스템(40')을 위한 드라이버 소프트웨어에게 상기 버퍼(14')의 어드레스를 제공함으로써, 용이해질 수도 있다. 대안적으로는, 앞으로 이해되는 바와 같이, 프레임들은 하나의 버퍼에서 렌더링되고 그리고 직접 메모리 액세스 등에 의해서 버스(22')를 통해 전송될 수도 있다.

[0075] 마찬가지로, 그래픽 서브시스템(30')을 위한 드라이버는 또한, 그래픽 서브시스템(30')의 디스플레이 인터페이스(74')가 메모리(14') 내의 프레임 버퍼를 샘플링하도록 구성되는바, 따라서 그래픽 서브시스템(40')에 의해 렌더링된 메모리(14')의 프레임 버퍼내의 이미지들을 디스플레이(26')에서 표시할 수 있다. 이와 동시에, 그래픽 서브시스템(30')을 위한 드라이버는 그래픽 서브시스템(30')의 그래픽 엔진(32')이 실질적으로 수면(dormant) 혹은 아이들(idle) 상태로 남아있으라고 명령할 수 있다. 이러한 동작 모드가 도11A에 도시되어 있는 바, 그래픽 서브시스템(40') 및 그래픽 서브시스템(30')의 액티브 블록들만이 크로스해칭되어 있다.

[0076] 이해되는 바와 같이, 도11A의 실시예에서, 메모리(50') 및 디스플레이 인터페이스(54')는 필요없다. 이와 같이, 이들 기능 블록들은 서브시스템(40')에서 제거될 수 있으며 따라서 비용을 절감할 수 있다. 결과적인 서브시스템(40')은 디스플레이 인터페이스를 포함할 필요가 없으며 그리고 서브시스템(40')에 대해 로컬인 프레임 버퍼를 정의하는 메모리가 없을 것이다. 서브시스템(30')에 의해 제공되는 기능성을 보완하도록 서브시스템(40')이 제조될 수 있기 때문에, 이와 같은 그래픽 서브시스템을 제조하는 것은 유용할 것이다. 예를 들면, 서브시스템은 3D 그래픽 혹은 비디오 디코딩 능력을 제공하는 그래픽 엔진(42')을 제공할 수 있다. 그래픽 엔진(32')은 이러한 능력들을 포함하지 않을 수도 있다. 이와 동시에, 그래픽 엔진(32')에 의해 제공되는 2D 그래픽 능력들이 서브시스템(40')에 포함될 필요는 없다. 소비자들은, 추가적인 기능이 필요한 경우에만, 그래픽 서브시스템(30')을 추가할 수 있다.

[0077] 디바이스(10')가 낮은 전력 소비 모드로 전환하거나 혹은 낮은 전력 소비 모드를 재개하는 경우에는, 블록 S812' - S818'이 실행된다. 넓은 의미로, 그래픽 서브시스템(40')은 부분적으로 혹은 완전히 디스에이블되며

그리고 낮은 전력 소비 모드에 있게 된다. 그리고 렌더링은 그래픽 서브시스템(30')에 의해 다시 수행된다. 이를 위해서, 그래픽 서브시스템(30')과 관련하여 상호연결된 임의의 디스플레이가 블록 S812'에서 인에이블될 수 있으며 그리고 그래픽 서브시스템(40')에 물리적으로 연결된 임의의 디스플레이는 블록 S814'에서 논리적으로 디스에이블될 수 있다. 다음으로, 그래픽 서브시스템(30')의 드라이버 소프트웨어 제어 동작은 또한 그래픽 서브시스템(30')이 메모리(14') 내의 이미지들을 렌더링하도록 구성된다. 디스플레이 인터페이스(74')는 포트(78')에 연결된 디스플레이(26') 상에 이미지들을 표시하도록 메모리(14')를 계속 샘플링한다. 또한, 처음에 프로세서(12')는 적절한 신호를 전력 제어기(60')에게 제공하여(블록 S818'), 그래픽 서브시스템(40')을 낮은 전력 상태로 만든다. 가장 간단한 형태로, 전력 제어기(미도시)는 그래픽 서브시스템(40')에 대한 전원 공급을 끊거나 혹은 그래픽 서브시스템(40')을 저전력 수면 모드에 놓이게 한다. 다시 한번, 이러한 낮은 전력 소비 모드에서는 전압이 억제되거나(throttled) 및/또는 그래픽 서브시스템(40')의 전부 또는 일부가 파워 다운되거나 및/또는 그래픽 서브시스템(40')에 의해 사용되는 선택된 클러스터들이 느려진다. 특히, 그래픽 서브시스템(40')의 그래픽 엔진(42')은 아이들 상태 혹은 실질적으로 아이들 상태(예컨대, 느려지거나, 디스에이블되거나 혹은 파워 다운되는 상태)로 남아있다. 이러한 동작 모드는 도9B에 도시되어 있으며, 도9B에서 어댑터(40')와 그래픽 서브시스템(30')의 액티브 기능 블록들만이 크로스해칭된다. 이러한 비활성/아이들 기능 블록들은 전체적으로 디스에이블될 수도 있으며 혹은 감소된 전압 혹은 감소된 클러스터 속도에서 동작할 수 있다.

[0078] 선택적으로는, 그래픽 엔진(32')이 사용중이 아닐때, 그래픽 서브시스템(30')의 일부분들이 디스에이블될 수 있다. 이러한 것은 그래픽 엔진(32')과 다른 구성요소들을 하나 이상의 전압 아일랜드(voltage islands) 상에 위치시킴으로써 용이해질 수 있는바, 이는 그래픽 서브시스템(40')이 이미지들을 렌더링할 책임이 있는 임의의 시간에서 GPIO 혹은 유사 회로에 의해서 디스에이블될 수도 있다.

[0079] 다른 변형예들 역시 명백해질 것이다. 예를 들면, 도11A에 도시된 높은 전력 소비 모드에서, 그래픽 서브시스템(30')과 그래픽 서브시스템(40') 둘다는 메모리(14') 혹은 메모리(50')에 렌더링할 수 있다. 이와 같이, 2개의 그래픽 서브시스템들(30', 40')은 서로 협력하여 동작할 수도 있는바, 그 각각이 메모리(14') 내의 교번 프레임(alternate frame)을 렌더링할 수도 있으며 혹은 메모리(14') 내의 각 프레임의 교번 부분(alternate portion)(예컨대, 스캔-라인)을 렌더링할 수도 있다.

[0080] 본 발명의 또 다른 실시예에서는, 전술한 바와 같이, 추가적인 디스플레이 기기들이 그래픽 서브시스템들(30', 40')에 연결되어, 높은 전력 소비 모드에서 다수 디스플레이들의 동시 사용을 허용할 수도 있다. 이러한 방식으로, 디스플레이 인터페이스(54')는 제 2 디스플레이를 구동하는데 이용될 수 있다. 낮은 전력 소비 모드로 전환되면, 디바이스(10')는 도11B에 도시된 바와 같이 동작하도록 구성될 수 있다.

[0081] 이와 유사하게, 디바이스(10')(또는 10)는 버스(22')(또는 22)에 연결된 다수개의 추가 그래픽 서브시스템들을 포함할 수 있는바, 이를 모두는 높은 전력 소비 모드에서 액티브될 수 있으며, 그리고 그래픽 서브시스템(30')의 디스플레이 인터페이스(74')를 통해 그래픽들을 렌더링할 수 있다. 낮은 전력 소비 모드로 전환되면, 이들은 디스에이블될 수 있으며 그리고 렌더링은 그래픽 서브시스템(30')의 그래픽 엔진(32')에게 맡겨질 수 있다.

[0082] 도12에 도시된 본 발명의 또 다른 실시예에서, 컴퓨팅 디바이스(10')는 직접 메모리 액세스(Direct Memory Access : DMA) 제어기(90)를 포함할 수 있다. DMA 제어기(90)는 메모리(50')로부터 메모리(14')로 데이터를 전송할 수 있다. 이러한 방식으로, 디바이스(10')의 높은 전력 소비 모드에서, 그래픽 서브시스템(40')은 이미지를 메모리(50')에 렌더링할 수 있다. 렌더링된 이를 이미지들은 이후, DMA 제어기(90)에 의해서 메모리(14')의 프레임 버퍼로 전송될 수 있다. DMA 제어기(90)는 그래픽 서브시스템(30' 혹은 40')의 일부를 형성할 수 있으며(예를 들면, 그래픽 엔진 32' 혹은 42'의 DMA 엔진들처럼) 혹은 컴퓨팅 디바이스(10') 내에 다르게 위치될 수도 있다. 데이터는 버스(20')를 통해 전송될 수도 있으며 또는 메모리(50')로부터 메모리(14')로 직접 전송될 수도 있다. 디스플레이 인터페이스(74')는 전술한 바와 같이 계속해서 동작할 것인바, 렌더링된 이미지를 디스플레이(26') 상에 표시하기 위해서 메모리(14')의 프레임 버퍼를 샘플링한다. 다시 한번, 높은 전력 소비 모드에 있는, 도10의 디바이스(10')의 액티브 블록들은 도12에서 크로스해칭되어 예시된다.

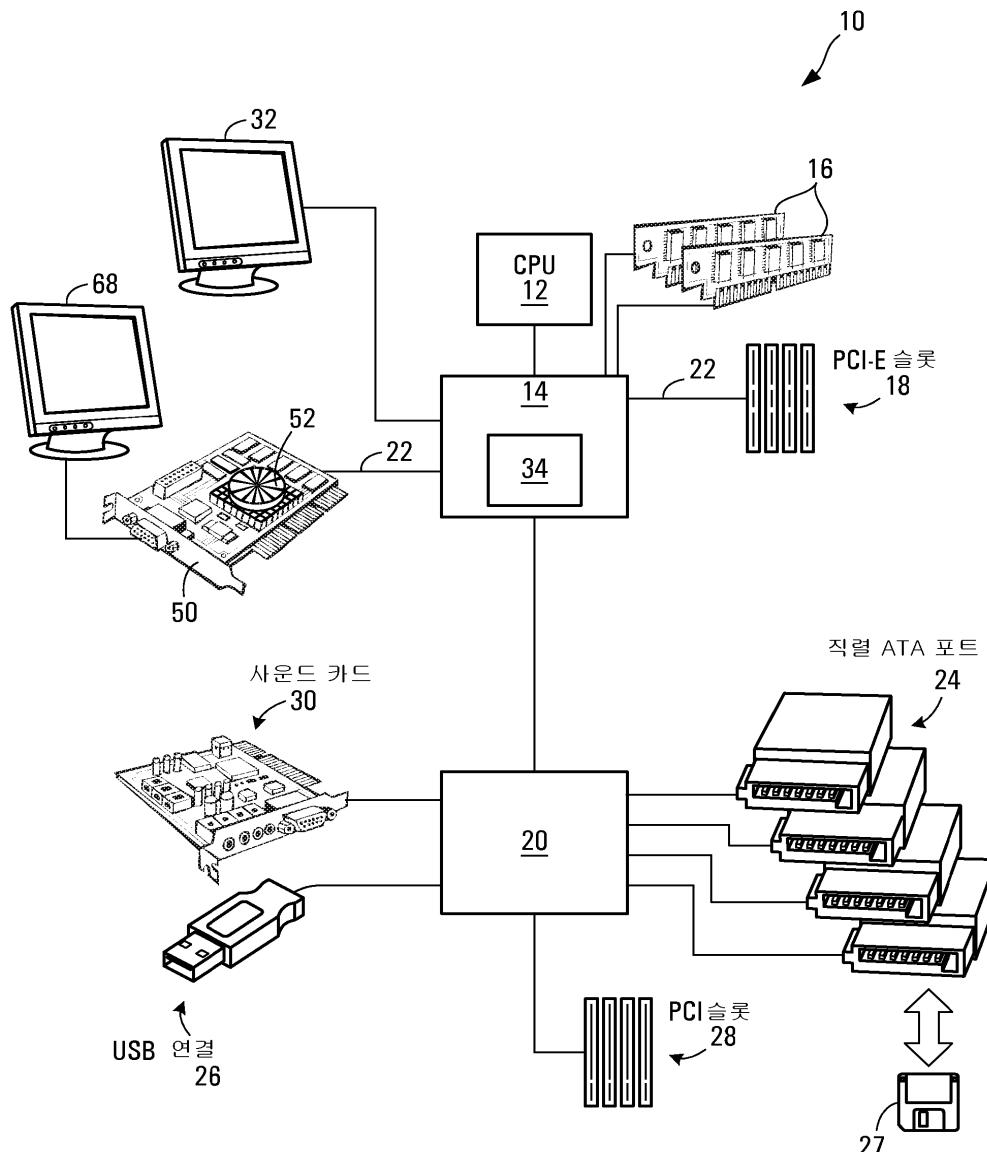
[0083] 물론, 전술한 바와 같은 본 발명의 실시예들은 예시의 목적으로만 의도된 것이며 본 발명을 제한하고자 의도된 것이 아니다. 본 발명을 구현하는 상술된 실시예들은, 형태, 부품들의 배치, 동작의 세부사항 및 순서에 있어서 수많은 변형들이 적용가능하다. 본 발명은 청구범위에 의해 정의된 바와 같은, 본 발명의 범위 내에 있는 이러한 모든 변형예들을 포함하도록 의도된 것이다.

## 부호의 설명

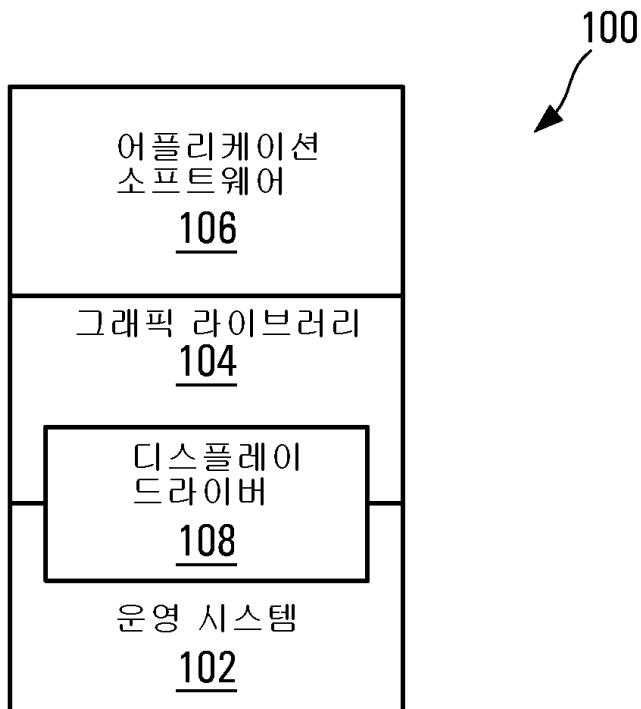
[0084]	12 : 중앙처리장치	14 : 노스 브리지
	16 : 시스템 메모리	22 : 상호연결 버스
	34 : 통합된 그래픽 어댑터	38 : 메모리제어기
	40 : 그래픽 프로세서	42 : 커맨드 프로세서

## 도면

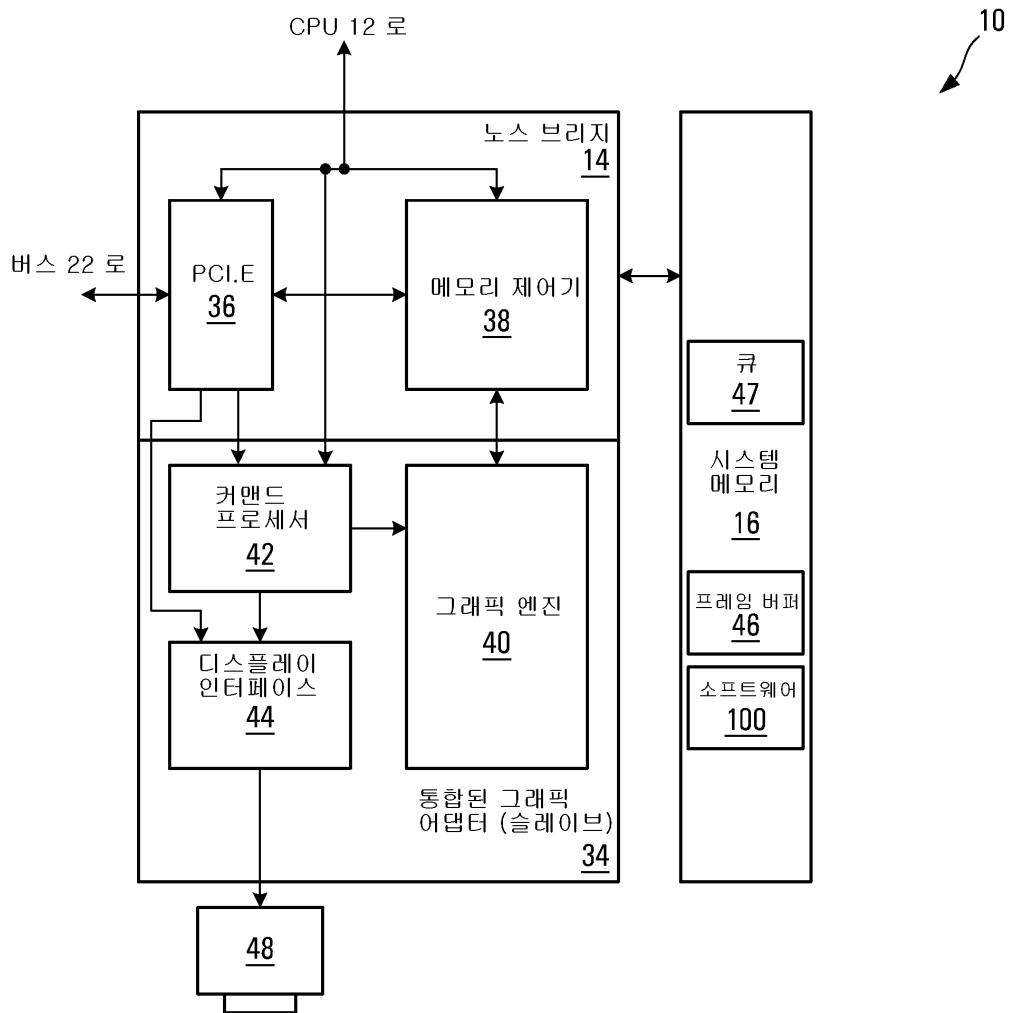
### 도면1



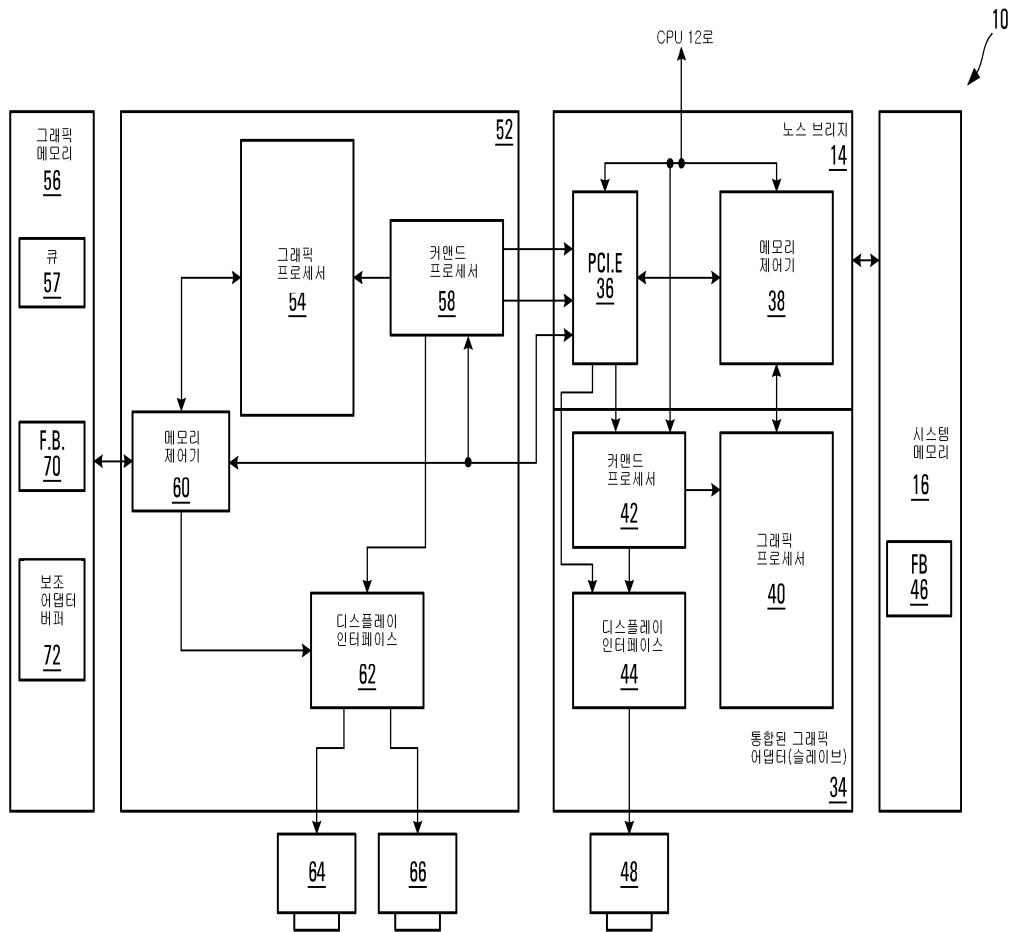
도면2



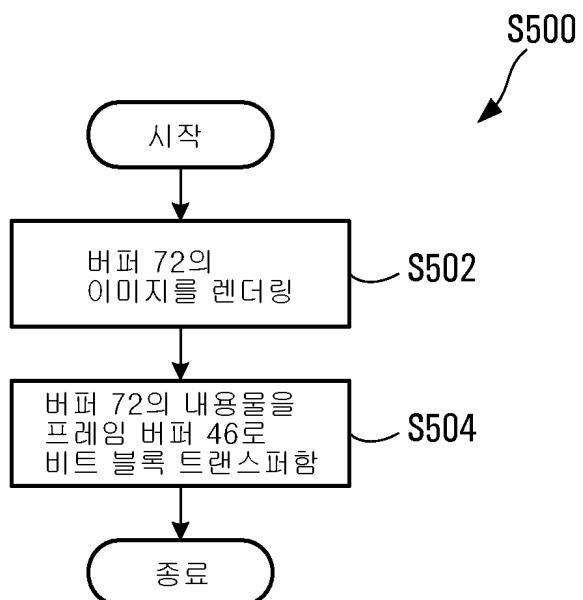
## 도면3



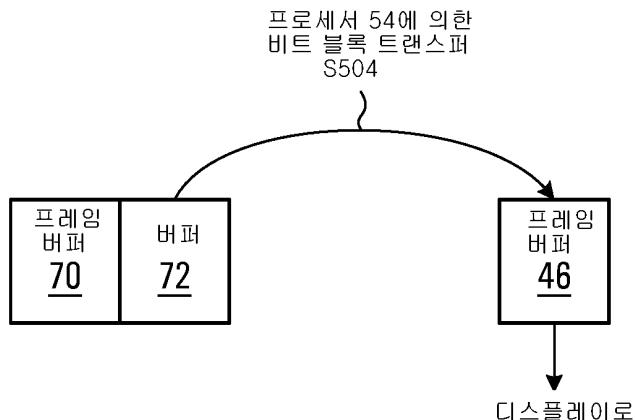
도면4



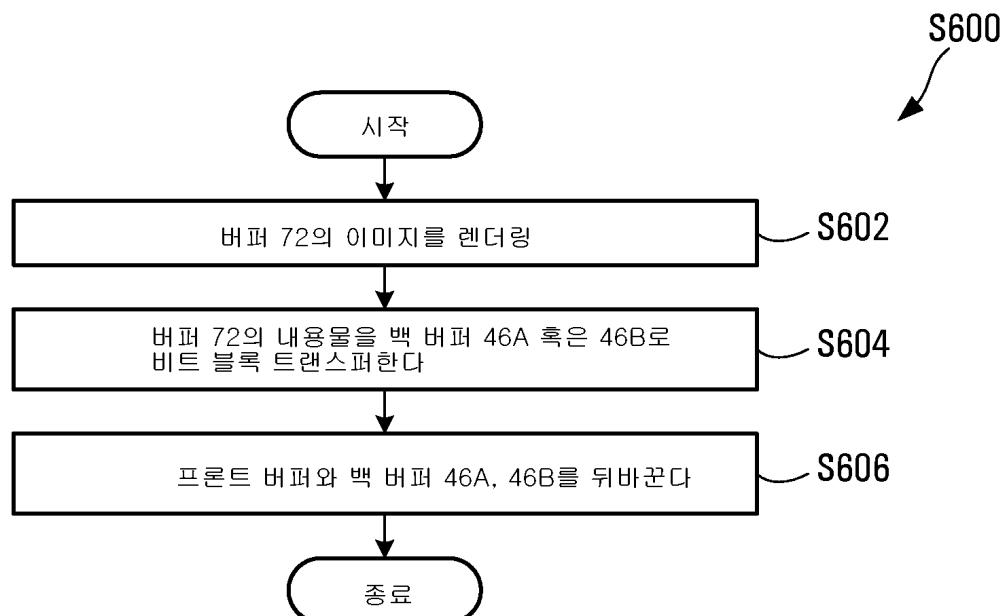
도면5a



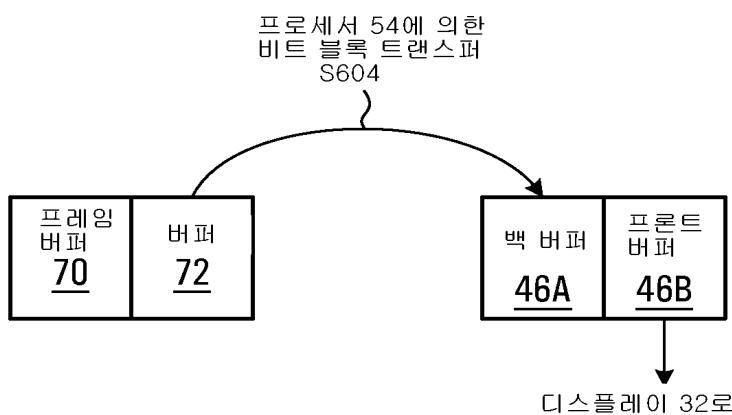
## 도면5b



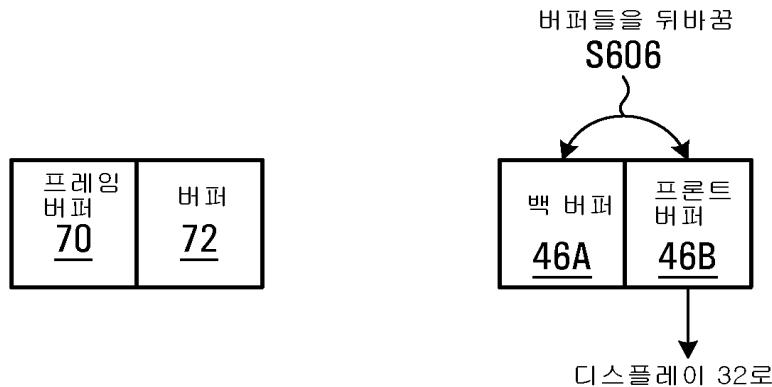
## 도면6a



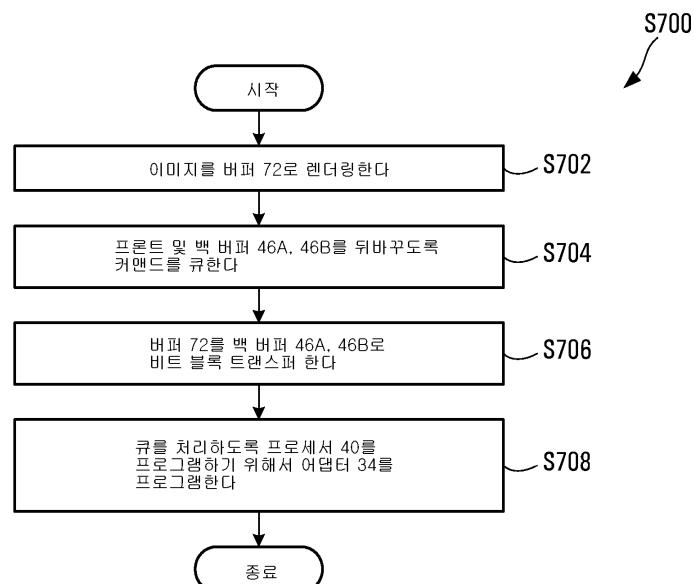
## 도면6b



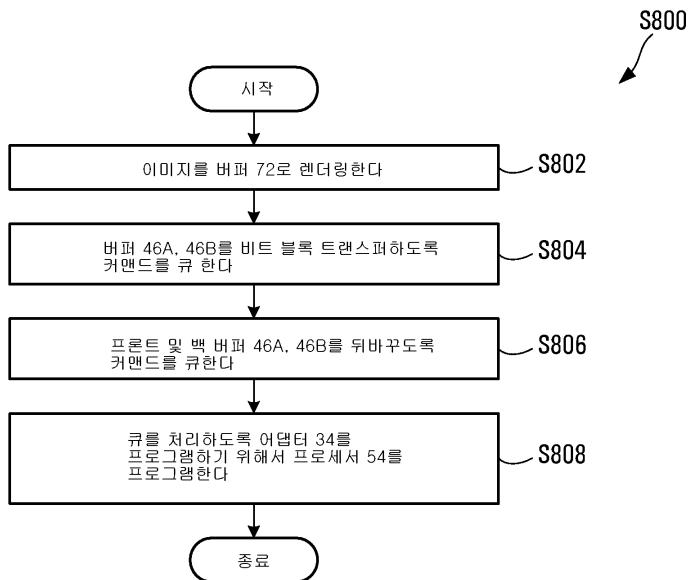
## 도면6c



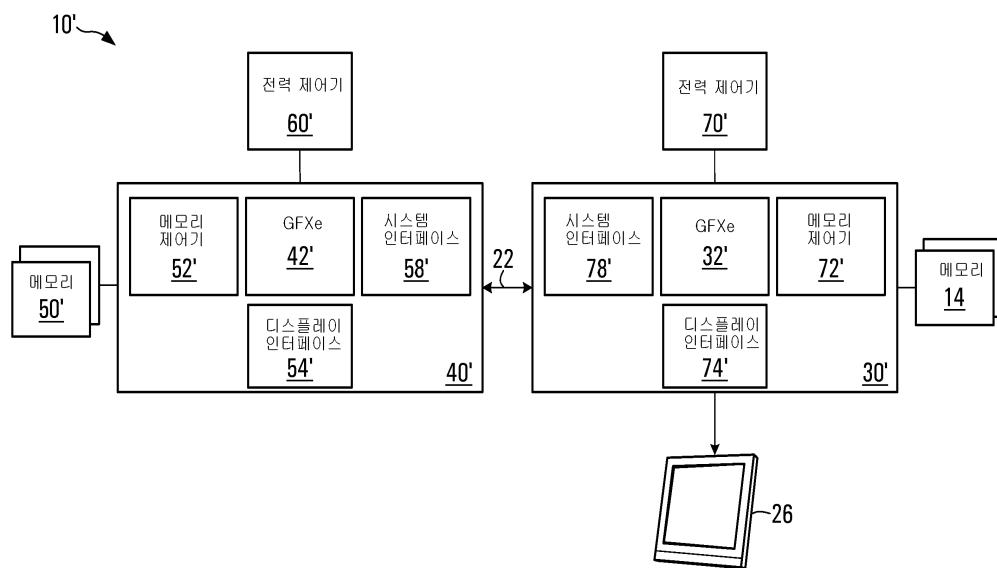
## 도면7



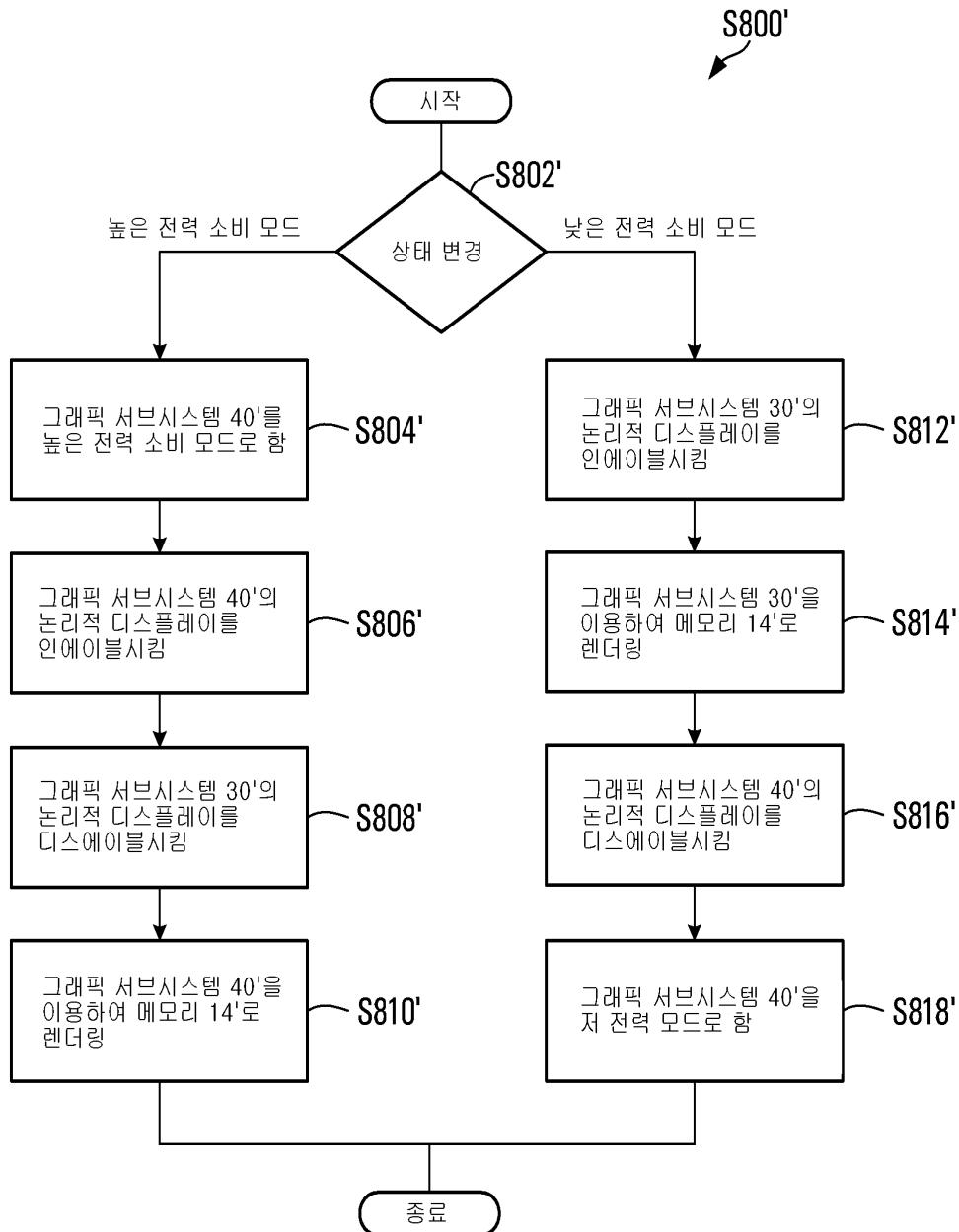
## 도면8



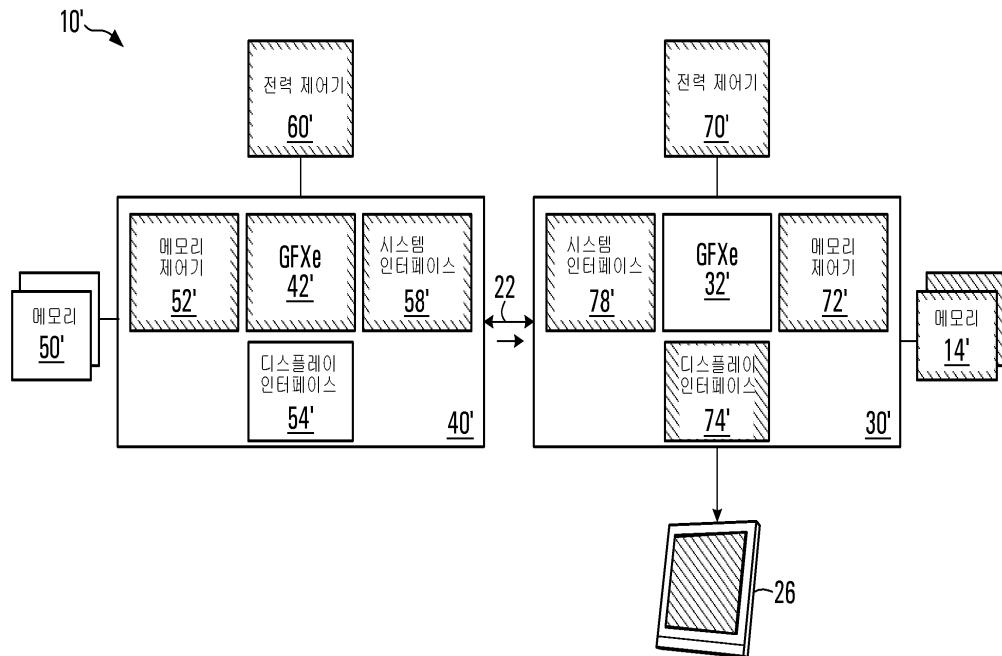
## 도면9



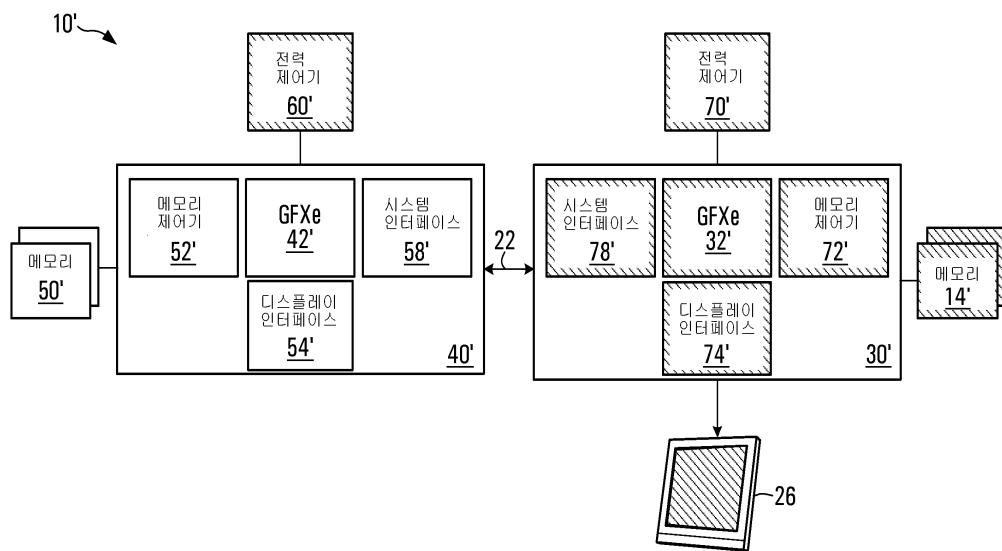
## 도면10



도면11a



도면11b



도면12

