

(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl.⁷
G09G 5/36

(45) 공고일자 2005년11월15일
(11) 등록번호 10-0528955
(24) 등록일자 2005년11월09일

(21) 출원번호 10-2003-7004217
(22) 출원일자 2003년03월24일
 번역문 제출일자 2003년03월24일
(86) 국제출원번호 PCT/US2001/029605
 국제출원일자 2001년09월20일

(65) 공개번호 10-2003-0036822
(43) 공개일자 2003년05월09일
(87) 국제공개번호 WO 2002/27658
 국제공개일자 2002년04월04일

(81) 지정국

국내특허 : 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아 헤르체고비나, 바르바도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 리히텐슈타인, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀란드, 영국, 그루지야, 헝가리, 이스라엘, 아이슬란드, 일본, 케냐, 키르기스스탄, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 리베이라, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아공화국, 몽고, 말라위, 멕시코, 노르웨이, 뉴질랜드, 슬로베니아, 슬로바키아, 타지키스탄, 투르크멘, 터어키, 트리니다드토바고, 우크라이나, 우간다, 미국, 우즈베키스탄, 베트남, 폴란드, 포르투갈, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 아랍에미리트, 안티구와바부다, 코스타리카, 도미니카, 알제리, 모로코, 탄자니아, 남아프리카, 벨리제, 모잠비크, 에쿠아도르, 필리핀, 콜롬비아, 그라나다, 감비아, 인도네시아, 오만, 튀니지,

AP ARIPO특허 : 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 시에라리온, 가나, 감비아, 짐바브웨, 모잠비크, 탄자니아, 잠비아,

EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르기스스탄, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크멘,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 리히텐슈타인, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투갈, 스웨덴, 핀란드, 사이프러스, 터어키,

OA OAPI특허 : 부르키나파소, 베닌, 중앙아프리카, 콩고, 코트디부아르, 카메룬, 가봉, 기니, 말리, 모리타니, 니제르, 세네갈, 차드, 토고, 기니 비사우, 적도 기니,

(30) 우선권주장 09/671,237 2000년09월28일 미국(US)

(73) 특허권자 인텔 코퍼레이션
미국 캘리포니아주 95052-8119 산타클라라 피.오.박스 58119 미션 칼리지 불바드 2200

(72) 발명자 란젠도르프브라이언
미국캘리포니아94510베니키아샘플스크로싱272

(74) 대리인 최광호

심사관 : 박부식

(54) 싱글 블록변환의 일부를 두개의 그래픽 컨트롤러 각각이병렬로 실행하게 하는 메커니즘 및 방법

요약

컴퓨터 시스템은, 싱글 블록변환 "BLT" 동작을 병렬로 실행하여 화소데이터 블록을 그래픽면의 소스에서 목적지로 옮기는 기능을 포함해 그래픽기능과 비디오기능을 공유하도록 구성된 다수의 그래픽 컨트롤러들과, 그래픽 컨트롤러에 연결되고 지정된 패턴으로 다른 그래픽 컨트롤러에 할당된 소스의 화소데이터를 저장하도록 구성된 다수의 로컬메모리들을 구비한다. 각각의 로컬메모리는 싱글 BLT 동작을 실행하라는 요청을 받았을 때 다른 그래픽 컨트롤러가 제어하는 영역에 있으면서 다른 로컬메모리에서 복사된 소스의 모든 화소데이터를 저장하는 스크래치 패드를 포함한다.

대표도

도 7

명세서

기술분야

본 발명은 컴퓨터 시스템 아키텍처에 관한 것으로 구체적으로는 컴퓨터 시스템에서 싱글블록 변환(BLT)의 일부를 두개의 그래픽 컨트롤러들 각각이 병렬로 실행하게 하는 메커니즘과 방법에 관한 것이다.

배경기술

컴퓨터 그래픽 애플리케이션에서 대부분의 공통적인 동작들중의 하나는 디스플레이 메모리의 그래픽면(10)의 일부(소스(12))로부터 다른 부분(목적지(14))으로 화소데이터 블록을 전달하는데 사용되는 블록변환(흔히 "BLT(Block Transform)" 또는 "화소 BLT"라 한다)이다. 일련의 소스 어드레스들은 이에 대응하는 일련의 목적지 어드레스와 같이 생성된다. 소스데이터(화소)는 소스어드레스로부터 리드(read)된 다음, 목적지 어드레스에 라이트(write)된다. BLT 동작은 간단한 데이터전송 이외에도 소스데이터(화소)에 대한 논리적 동작과 기타 OPERAND(흔히 래스터(raster) 동작이나 ROP라고도 함)를 실행할 수도 있다. ROP와 BLT는 1993년 Addison-Wesley Publishing Company, Inc.에서 출판한 Foley, VanDam, Feiner, Hughes의 Computer Graphics Principles and Practice 2판 56-60페이지에 설명되어 있다. BLT 동작은 일반적으로 컬러변환, 영상의 스트레칭과 클리핑과 같이 컴퓨터 시스템에서 영상을 형성하거나 조작하는데 이용된다. BLT 동작과 함께 ROP를 실현하는 것은, 요청된 ROP 명령어에 따라 논리적 동작을 실행하는 하나 이상의 논리 회로에 소스 및/또는 목적지 데이터를 결합하여 실행된다. 소스데이터, 패턴 및 목적지 데이터를 결합하는데 사용되는 ROP의 가능한 형태는 여러가지 있다. 이에 대해서는 Addison-Wesley Publishing Company, Inc.에서 1994년 발간한 Richard F. Ferraro의 Programmer's Guide to the EGA, VGA and Super VGA Card 3판의 707-712 페이지를 참고하면 된다. 마찬가지로, 목적지 데이터외에도 브러시로 알려진 일반적인 윈도우패턴도 포함될 수 있다. 브러시 패턴은 통상 디스플레이 스크린의 윈도우에 배경화면으로 사용되는 사각형 화소들이 여러 열로 배열된 것이다. 특정된 ROP 형태에 따라, 브러시 패턴은 목적지 데이터에 복사되거나 다른식으로 목적지 데이터와 결합될 수 있다.

BLT와 관련 동작은 통상 그래픽 컨트롤러와 같은 컴퓨터 시스템의 특정 하드웨어에 의해 다른 그래픽 동작과 함께 실행된다. BLT와 관련 동작들을 수행하는 특수한 하드웨어를 일반적으로 그래픽 컨트롤러에 있는 그래픽엔진이라 한다. (ROP와 함께) 기본 BLT 동작들은, 소스데이터를 소스(SRC; 12)로부터 임시 데이터 저장장치에 리드하는 단계; 목적지 데이터나 기타 OPERAND 데이터를 그 위치로부터 리드하는 단계, 그 데이터에 대한 ROP를 실행하는 단계, 및 그 결과를 목적지(DST; 14)에 라이트하는 단계를 포함한다.

소스(12)와 목적지(14)는 오버랩 영역(16)에서 오버랩될 수도 있다(도 2 참조). 그러나, BLT 동작에 앞서 소스 화소와 목적지 화소의 값을 이용해 목적지 화소의 새로운 값을 계산해야만 한다. 즉, BLT 동작 이후의 그래픽면(10)의 상태는 그 결과를 먼저 계산하고 전체 목적지(14)를 위한 임시 데이터 저장소에 저장한 다음 목적지(14)에 복사되는 것처럼 되어야만 한다.

종래의 컴퓨터 시스템은 소스(12)의 전연(leading edge)을 목적지(14)에 복사하여 오버랩되는 소스(12)와 목적지(14)를 다룬다. 그 결과, 모든 화소는 목적지(14)로서 라이트되기 전에 소스(12)로서 리드된다. 그러나, 발전된 그래픽 애플리케이션을 위해 기존의 컴퓨터시스템의 확장카드에 추가적인 그래픽 컨트롤러를 장착하거나 플러그인하면, 비록 성능이 중요하지 않을지라도, 동일한 면에 작용하는 두개의 그래픽 컨트롤러에서는 동기화 및 일관성 문제가 일어난다. 소스와 목적지인 화소들을 목적지로 라이트하기 전에 소스로 리드하도록 동작을 시리얼화하면, 싱글 컴퓨터 시스템내의 다수의 그래픽 컨트롤러의 성능 장점이 저하될 것이다.

발명의 상세한 설명

따라서, 특히 BLT 및 관련동작중에 소스와 목적지의 오버랩 영역들을 취급할 때, 하이브리드 모델 컴퓨터 시스템내의 여러 그래픽 컨트롤러들이 적당한 동기화를 설정하고 일관성을 위해 동일한 이미지 렌더링 임무를 효과적으로 할당하고 공유하도록 할 필요가 있다.

도면의 간단한 설명

이하, 첨부 도면들을 참조하여 본 발명에 대해 자세히 설명하겠으며, 동일한 번호는 동일하거나 유사한 부분을 가르킨다.

도 1은 화소데이터 블록을 그래픽면의 소스에서 목적지로 옮기는 BLT 동작의 일례를 보여주는 도면;

도 2는 소스와 목적지가 오버랩되어 있는 그래픽면의 소스에서 목적지로 화소데이터 블록을 옮기는 BLT 동작의 일례를 보여주는 도면;

도 3은 그래픽/멀티미디어 플랫폼을 갖는 컴퓨터 시스템의 블록도;

도 4는 본 발명의 실시예에 따라 내부그래픽 컨트롤러를 갖는 호스트 칩셋을 구비한 컴퓨터 시스템의 블록도;

도 5는 본 발명의 실시예에 따라 내부그래픽 컨트롤러와 외부그래픽 컨트롤러를 갖는 하이브리드 호스트 칩셋을 구비한 컴퓨터 시스템의 블록도;

도 6은 본 발명의 실시예에 따라 내부그래픽 컨트롤러와 외부그래픽 컨트롤러 사이로 분할된 그래픽면을 보여주는 도면;

도 7은 본 발명의 일례에 따라 싱글 BLT 동작의 일부를 두개의 (내외부) 그래픽 컨트롤러들 각각이 병렬로 실행하도록 하는 메커니즘을 보여주는 도면;

도 8은 본 발명의 실시예에 따른 그래픽 컨트롤러의 블록도.

실시예

본 발명은 컴퓨터 그래픽 애플리케이션용의 컴퓨터, 서버, 주변장치, 저장장치, 소비가전장치 등의 워크스테이션에 연결하는 연속(follow-on) 칩 디자인을 포함한 모든 현대의 컴퓨터 시스템, 프로세서, 비디오 소스, 칩셋과 같이 사용하는데 응용된다. 설명의 편의상, 고성능 비디오 기능을 전달하도록 병렬로 실행하는 멀티미디어 엔진의 베이직 그래픽/멀티미디어 플랫폼 아키텍처를 갖는 컴퓨터 시스템에 관해 주로 설명하겠지만, 본 발명은 이에 한정되는 것은 아니다. 그래픽이란 말은 컴퓨터-생성 이미지, 심볼, 천연 및/또는 합성 물체와 장면의 비주얼 표지, 그림 및 텍스트를 포함할 수 있지만, 이에 한정되지는 않는다.

예를 들어, 도 3에는 BLT 동작을 실행하기 위한 베이직 그래픽/멀티미디어 플랫폼을 갖는 컴퓨터 시스템(100)이 도시되어 있다. 도 3에 도시된 바와 같이, (PC라 일컬어질 수 있는) 컴퓨터 시스템(100)은 Intel7 i386, i486, Celeron J, Pentium7 프로세서와 같은 하나 이상의 CPU 또는 프로세서(110), 프론트 사이드 버스(20)를 통해 프로세서(110)에 연결된 메모리 컨트롤러(120), 메모리버스(30)를 통해 메모리 컨트롤러(120)에 연결된 메인 메모리(130), (Advanced Graphics Port AAGP@bus와 같은) 그래픽버스(40)를 통해 메모리 컨트롤러(120)에 연결된 그래픽 컨트롤러(140), 및 PCI(Peripheral Component Interconnect) 버스(50)와 같은 다양한 I/O 장치에 접속하기 위해 메모리 컨트롤러(120)에 연결된 IO 컨트롤러 허브(ICH; IO controller hub)(170)를 포함한다. PCI 버스(50)는 자동 구성능(automatic

configurability) 및 멀티플렉스 어드레스, 컨트롤 데이터 라인들을 갖는 고성능 32/64 비트 싱크로너스 버스일 수 있는데, 이에 대해서는 새로운 비디오, 네트워킹, 또는 디스크 메모리 저장성능을 갖는 부가 구성용의 1995년 6월 1일자의 PCI Special Interest Group(SIG)에서 설명한 *PCI Local Bus Specification Revision 2.1* 최신판에 설명되어 있다.

그래픽 컨트롤러(140)는 BLT와 관련 동작을 실행하고 (음극선관, 액정 디스플레이, 평판 디스플레이와 같은) 디스플레이 모니터(150)에서의 그래픽 및/또는 비디오 영상의 비주얼 디스플레이를 제어하는데 이용될 수 있다. (프레임 버퍼와 같은) 로컬메모리(160)는 그래픽 애플리케이션 전용의 별도의 메모리일 수 있다. 이런 로컬메모리(160)는 그래픽 컨트롤러(140), 하나 이상의 프로세서(110), 또는 디스플레이 모니터(150)에 비디오 영상들을 표시하기 위한 컴퓨터 시스템(100) 내의 다른 장치로부터의 화소데이터를 저장하기 위해 그래픽 컨트롤러(140)에 연결될 수 있다.

한편, 메모리 컨트롤러(120)와 그래픽 컨트롤러(140)는 고성능 3D, 2D 및 모션 보상 비디오 성능을 제공하도록 병렬로 실행되는 전용 멀티미디어 엔진들을 구비한 하나의 그래픽/메모리 컨트롤러 허브(GMCH; graphics and memory controller hub)로 통합될 수도 있다. GMCH는 인텔사에서 제조하는 PIIX47 칩, PIIX67 칩과 같은 PCI 칩으로 구현될 수도 있다. 또, 이런 GMCH는 Intel7 810, 8XX 시리즈 칩셋과 같이 I/O 컨트롤러 허브(I/O controller hub)와 펌웨어 허브(firmware hub)와 함께 호스트 칩셋의 일부로 구현될 수도 있다.

도 4에는 호스트 칩셋(200)을 갖는 컴퓨터 시스템(100)이 도시되어 있다. 이 컴퓨터 시스템(100)은 기본적으로 도 3과 같은 요소를 갖지만, 그래픽/메모리 컨트롤러 허브(GMCH; 210), I/O 컨트롤러 허브(ICH; 220), 펌웨어 허브(FWH; 230)로 이루어지는 고집적 3-칩 솔루션을 제공하는 호스트 칩셋(200)은 예외이다.

GMCH(210)는 그래픽 애플리케이션과 비디오 기능용이면서 하나 이상의 메모리 소자들을 시스템 버스(20)에 인터페이스하기 위한 내부 그래픽 컨트롤러(212)를 내장한다. GMCH(210)의 내부 그래픽 컨트롤러(212)는 3D 물체의 표현으로부터 래스터화된 2D 디스플레이 이미지를 생성하는 기능을 포함한 다양한 3D 그래픽 기능을 실행하기 위한 3D (텍스처 매핑) 엔진(도시 안됨), 그래픽 표면에서 메모리 위치들 사이에 화소데이터를 전송하는 BLT(Block Transform) 동작을 포함한 2D 기능을 실행하는 그래픽 엔진(도시 안됨), 비디오나 그래픽 이미지를 디스플레이하는 디스플레이 엔진(도시 안됨), 및 디지털 비디오신호를 출력하고 종래의 디스플레이 모니터(150)나 새로운 공간절약형 디지털 평판 디스플레이(FPD; flat panel display)에 연결하기 위한 디지털 비디오 출력포트를 포함할 수도 있다.

GMBH(210)는 메모리 버스(30)를 통해 메인메모리(130)에, 메모리(160)와 디스플레이 모니터(150)에, 그리고 엔코더와 디지털 비디오 출력신호를 통해 텔레비전에 연결될 수 있다. GMCH(120)로는 Intel 82810, 82810-DC100 칩이 있다. GMCH(120)는 또한 ICH(220)에 연결될 수 있는 하나 이상의 I/O 소자들과 하나 이상의 프로세서(110) 사이의 통신이나 신호전송을 위한 브리지나 인터페이스 기능을 한다.

ICH(220)는 하나 이상의 I/O 소자들을 GMCH(210)에 인터페이스한다. FWH(230)는 ICH(220)에 연결되고 추가적인 시스템 제어를 위한 펌웨어를 제공한다. ICH(220)로는 Intel⁷ 82801 칩이 있고, FWH(230)로는 Intel⁷ 82802 칩이 있다.

ICH(220)는 다음과 같은 다양한 I/O 장치에 연결될 수 있는바: PCI 슬롯(194), ISA(Industry Standard Architecture) 버스 옵션(196) 및 LAN(local area network) 옵션(198)에 하나 이상의 I/O 장치들이 연결되어 있을 수 있는 PCI(Peripheral Component Interconnect) 버스(50)(PCI 로컬버스 Specification Revision 2.2); 마우스, 키보드 및 기타 주변장치(도시 안됨)에 연결하기 위한 수퍼 I/O 칩(192); 오디오 코더/디코더(코덱)와 모뎀 코덱; 다수의 USB(Universal Serial Bus) 포트(USP Specification, Revision 1.0); 및 하나 이상의 마그네틱 하드디스크 드라이브나 기타 I/O 장치들을 받아들이기 위한 다수의 Ultra/66 ATA(AT Attachment) 2 포트(X3T9.2 948D Specification; IDE(Integrated Drive Electronics) 포트로 알려지기도 함).

USB 포트와 IDE 포트는 하드디스크 드라이브(HDD)와 콤팩트디스크 롬(CD-ROM)에 인터페이스를 제공하는데 사용될 수도 있다. I/O 지원과 기능의 확장을 위해 호스트 칩셋의 ICH에 I/O 장치와 플래시 메모리(예; EPROM)를 연결할 수도 있다. 이런 I/O 장치에는, 영숫자 키보드의 동작을 제어하기 위한 키보드 컨트롤러, 마우스, 트랙볼, 터치패드, 조이스틱 등의 커서 제어장치, 마그네틱 테이프, 하드디스크 드라이브(HDD), 플로피디스크 드라이브(FDD)와 같은 대량저장장치, 프린터와 스캐너에 대한 병렬포트 등이 있다. LCD(low pin count) 버스를 통해 호스트 칩셋의 ICH에 플래시 메모리를 연결할 수도 있다. 플래시메모리는 컴퓨터 시스템(100)을 시동할 때 일련의 시스템 BIOS(basic input/output start up) 루틴을 저장할 수 있다. 수퍼 I/O 칩(192)은 다른 그룹의 I/O 장치와의 인터페이스를 제공할 수 있다.

도 3, 4에 도시된 컴퓨터 시스템의 경우, 그래픽 소스의 일부(소스)로부터 다른 곳(목적지)으로 화소 데이터 블록을 전송하기 위한 "BLT" 제어와 관련 동작들을 포함한 그래픽 애플리케이션용으로만 도 3의 그래픽 컨트롤러(140)나 도 4의 내부 그래픽 컨트롤러(212)를 사용할 수도 있다. 도 2와 관련해 설명한 것처럼 소스와 목적지가 오버랩되어 있을 경우, 도 3의 그래픽 컨트롤러(140)나 도 4의 내부 그래픽 컨트롤러(212)는 오버랩 영역의 전연(leading edge)을 먼저 복사하도록 구성된다. 예컨대, 소스(12) 우연(right edge)의 화소 컬럼이 목적지(14)의 우연에 먼저 복사된 다음, 우측으로 복사될 수 있다. 그 결과, 모든 화소들은 목적지(14)로서 라이트되기 전에 소스(12)로서 리드된다.

그러나, 발전되고 가속화된 그래픽 애플리케이션을 위해 그리고 BLT 동작을 처리하는데 필요한 시간을 단축하기 위해 도 3에 도시된 것과 같은 기존의 컴퓨터 시스템의 확장보드{예, PCI 슬롯(194)}에 추가적인 그래픽 컨트롤러(240)나 관련 로컬메모리(260)를 설치하거나 플러그인할 경우, 도 6에 도시된 것과 같은 BLT와 관련 동작을 위해 내부(호스트) 그래픽 컨트롤러(212)와 외부(원격) 그래픽 컨트롤러(240) 사이에 그래픽면(10)이 공유될 필요가 있을 뿐만 아니라, 내부(호스트) 그래픽 컨트롤러(212)와 외부(원격) 그래픽 컨트롤러(240) 사이에 동기화 및 일관성도 생긴다.

예컨대, 추가적인 그래픽 컨트롤러(240)는 플러그인플레이 장치일 수도 있지만 이것이 필수적인 것은 아니다. 본 발명을 응용하는데 필요한 것은, BLT 동작들을 서로 비동기적으로 실행시키는 두개의 그래픽 엔진들을 시스템이 구비하는 것이다. 다시말해, 두개의 그래픽 엔진들은 공통 클럭을 사용하여 클럭 레벨에서 동기적으로 동작할 수 있지만, 각각의 그래픽 엔진은 다른 그래픽 엔진이 명령어를 실행한 진행상황의 세부내용이나 심지어 명령어 리스트내의 진행상황도 알지 못한다. 동기화와 일관성 문제들은 두개의 독립적 그래픽 엔진들이 BLT 동작을 협조하여 실행하기 때문에 간단히 유도된다. 마찬가지로, 그래픽 엔진을 두개 사용하면 하나만 사용했을 때보다도 BLT 동작을 더 신속히 실행할 수 있다.

도 6은 BLT와 관련 동작을 실행하기 위해 내부(호스트) 그래픽 컨트롤러(212)와 외부(원격) 그래픽 컨트롤러(240) 사이에 바둑판 패턴으로 공유된 그래픽면(10)의 일례가 도시되어 있다. 내부(호스트) 그래픽 컨트롤러(212)와 호스트 로컬 메모리(160)는 바둑판 영역중 표시된 부분을 다루도록 할당될 수 있다. 마찬가지로, 외부(원격) 그래픽 컨트롤러(240)와 원격 로컬메모리(260)는 바둑판 영역중 표시되지 않은 부분을 다루도록 할당될 수 있지만, 그 반대로 할당될 수도 있다. 바둑판 무늬는 내부(호스트) 그래픽 컨트롤러(212)와 외부(원격) 그래픽 컨트롤러(240) 사이의 영향을 나눈 것을 보여주기 위한 역할을 할 뿐이다. 그래픽면(10)을 내부 그래픽 컨트롤러(212)와 외부 그래픽 컨트롤러(240) 사이로 분할할 수만 있으면 다른 무늬를 사용해도 된다.

주어진 소스 화소에서 BLT 동작을 실행하고자 할 때, 수평 영역은 수직 영역의 목적지 화소와 관련될 수 있고, 그 반대일 수도 있다. 이런 경우, 이 화소에 대해 어떤 그래픽 컨트롤러(212,240)가 BLT 동작을 실행할 수 있을지를 결정해야만 한다. 목적지 화소를 포함한 그래픽면(10) 영역을 담당하는 그래픽 컨트롤러가 이 화소에 대한 BLT 동작을 실행하는 책임을 질 수 있는 목적지 장악방침을 선택할 수도 있다.

화소가 외부 그래픽 컨트롤러(240)의 목적지이고 내부 그래픽 컨트롤러(212)의 소스인 BLT 동작이 있다. 이런 화소를 내부 그래픽 컨트롤러(212)가 리드할 때까지 외부 그래픽 컨트롤러(240)는 화소를 라이트할 수 없다. 내부 그래픽 컨트롤러(212)의 목적지와 외부 그래픽 컨트롤러(240)의 소스인 화소들에 대해서도 비슷한 상황이 일어난다. 소스(12)와 목적지(14) 둘다인 화소들을 목적지로 라이트하기 전에 소스로 리드하도록 동작이 연속화되면, 하이브리드 모델 컴퓨터시스템(100)내의 다수의 그래픽 컨트롤러들(212,240)의 성능 장점이 무효로 된다.

도 7에는, 본 발명에 따라 하이브리드 모델 컴퓨터시스템(100)내의 싱글 BLT 동작의 일부분을 두개(내외부) 그래픽 컨트롤러들(212,240)이 각각 병렬로 실행할 수 있는 메커니즘과 방법이 도시되어 있다. 일반적으로, 각각의 그래픽 컨트롤러(212,240)는 다른 그래픽 컨트롤러(240,212)가 제어하는 영역에 있는 모든 소스화소들을 먼저 복사하고, 복사가 이루어졌음을 다른 그래픽컨트롤러에 표시한다. 일반적으로, 그래픽 컨트롤러들(212,240)중 하나는 복사가 이루어졌음을 나머지 그래픽 컨트롤러에 신호해야만 한다. 이런 정보를 전송하는 방법으로는, 1) 다른 그래픽 컨트롤러내의 메모리 매핑 I/O 위치에 라이트하고; 2) 라이트된 위치는 정보를 전달할 수 있고 라이트된 데이터 값은 아무 의미도 갖지 않으며; 3) 라이트된 위치가 여러가지 용도를 가질 수 있고 라이트된 값은 BLT 복사 동기화가 통신되고 있는 것임을 표시하고; 4) 다른 그래픽 컨트롤러가 선택할 수 있는 실제 메모리 위치에 라이트하며; 5) 복사가 이루어졌음을 다른 그래픽 컨트롤러에 신호하는 특수 신호를 확인하며; 6) (PCI, AGP 버스와 같은) 버스를 통해 전용 특수사이클을 전송한다.

다음, 각각의 그래픽 컨트롤러(212,240)는 다른 그래픽 컨트롤러(240,212)의 소스인 목적지 화소들을 업데이트하기 전에 동기화 라이트를 기다려야만 한다. 그래픽 컨트롤러들(212,240)중 하나의 목적지이면서 나머지 그래픽 컨트롤러(240,212)의 소스는 아닌 모든 화소들은 언제라도 업데이트될 수 있다. 그 결과, 두개의 (내외부) 그래픽 컨트롤러들

(212,240)과, 하이브리드 모델 컴퓨터 시스템(100)내의 각각의 로컬메모리들(160,260)은 적절한 동기화를 설정할 수 있고 일관성을 위한 동일 영상 렌더링 임무를 효과적으로 할당 및 공유할 수 있는데, 특히 BLT 및 관련 동작을 하는 동안 오버래핑 소스와 목적지 영역들을 다룰 때 그렇다.

도 7에 도시된 바와 같이, 이 메커니즘(700)은 내부그래픽 컨트롤러(212), 외부그래픽 컨트롤러(240) 및 각각의 로컬메모리(160,260)를 포함할 수 있다. 내부 (호스트) 그래픽 컨트롤러(212) 자체의 로컬메모리(160)에는 외부 (원격) 그래픽 컨트롤러(240)로부터 복사된 화소데이터를 저장하도록 제외된 메모리 어드레스 세트인 스크래치패드(SP; scratch pad, 162)와, 소스(12)/목적지(14)용 메모리영역이 포함되어 있다. 마찬가지로, 외부 (원격) 그래픽 컨트롤러(240) 자체의 로컬메모리(260)에도 내부 (호스트) 그래픽 컨트롤러(212)로부터 복사된 화소데이터를 저장하도록 제외된 메모리 어드레스 세트인 스크래치패드(SP; 262)와, 소스(12)/목적지(14)용 메모리영역이 포함되어 있다. 한편, 스크래치 패드(162,262)는 로컬메모리(160,260)가 아닌 시스템내 어디에도 위치할 수 있다. 예컨대, 다이에, 메인메모리(130)에(도 3 참조), 다른 그래픽 컨트롤러의 로컬메모리에 스크래치 패드를 배치할 수도 있다. 단지 BLT 기간동안 이 목적으로 전용된 저장장치이기만 하면 된다. 이 저장장치는 협력적 BLT가 실행되지 않을 때는 다른 목적으로 사용될 수도 있다. 또, 그래픽 전용의 싱글 로컬메모리가 두개의(내외부) 그래픽 컨트롤러 사이에 공유될 수도 있다. 그러나, 각각의 스크래치 패드들은 독립적일 필요가 있을 수도 있다.

그래픽면(10)이 내부 (호스트) 그래픽 컨트롤러(212)와 외부 (원격) 그래픽 컨트롤러(240) 사이에 분할되기 때문에, 그래픽 컨트롤러들(212,240) 각각은 소스로부터 각각의 스크래치 패드(SP)(162,262)로 원격 화소들을 판독할 수 있다. 다시말해, 그래픽 컨트롤러들(212,240) 각각은 동일한 소스(12)를 스캔하고, 그 소스(12)에서 다른쪽 그래픽 컨트롤러로 갈 필요가 있는 모든 화소들을 결정하며, 다른쪽 그래픽 컨트롤러의 로컬메모리로부터 이들 화소들을 획득할 수 있다.

특히, BLT 동작 초기에, 각각의 그래픽 컨트롤러는 예컨대 사각형의 소스를 스캔하고, 원격 위치에 있는 이들 화소들을 결정하며, 이들 원격 소스 화소들을 원격 로컬메모리로부터 로컬 스크래치 패드(SP)로 복사한다. 또, 목적지 화소인 이들 원격 소스화소들은 협력을 위한 오버헤드를 줄이기 위해 복사될 필요가 있다. 예컨대, 소스와 목적지가 오버랩되지 않으면, 스크래치패드(SP)에 처음부터 복사하지 않고도 BLT를 진행할 수 있다. 다음, 내부 (호스트) 그래픽 컨트롤러(212)는 소스(12)를 스캔하고, 외부 (원격) 그래픽 컨트롤러(240)에 연결된 원격 로컬메모리(260)로 배치되는 모든 화소들을 포함해 소스(12)내에서 목적지(14)를 계산하는데 필요한 모든 화소들을 찾으며, 이들 모든 원격 소스화소들을 도 7에 1로 표시된 것처럼 호스트 스크래치패드(SP)(162)에 복사하라는 요청문을 보낸다.

내부 (호스트) 그래픽 컨트롤러(212)와 외부 (원격) 그래픽 컨트롤러(240)가 각각의 스크래치패드(SP)(162,262)에 원격 소스화소들을 복사한 뒤, 내외부 컨트롤러들(212,240) 각각에 동기 라이트를 보내 복사가 단계 2에서 이루어졌음을 표시할 수 있다. 예컨대, 내부 (호스트) 그래픽 컨트롤러(212)가 로컬메모리(160)의 스크래치패드(SP)(162)에 원격 소스화소들을 복사했을 때는, 내부 (호스트) 그래픽 컨트롤러(212)가 외부 (원격) 그래픽 컨트롤러(240)에 동기 라이트를 한다. 마찬가지로, 외부 (원격) 그래픽 컨트롤러(240)가 로컬메모리(260)의 스크래치패드(SP)(262)에 원격 소스화소들을 복사했을 때는, 외부 (원격) 그래픽 컨트롤러(240)가 내부 (호스트) 그래픽 컨트롤러(212)에 동기 라이트를 한다. 동기 라이트는 화소데이터를 로컬메모리에 리드 및/또는 라이트하는 메모리 사이클을 나타낼 수 있다. 동기 라이트가 일어날 때까지, 그래픽 컨트롤러들(212,240) 어떤 것도 BLT 동작을 진행할 수 없다. 그러나, 소스와 목적지가 오버랩되지 않으면 이런 동기 라이트를 생략할 수도 있다. 소스와 목적지가 오버랩되었는지의 여부를 전체 메커니즘에 호출할 필요가 있다. 불필요한 오버헤드(스크래치 패드로의 복사와 동기 라이트)로 인한 성능 비용의 절감을 위해 BLT마다 메커니즘을 호출할 수도 있다.

동기 라이트를 수신하면, 원격 소스화소들의 복사를 이미 완료한 그래픽 컨트롤러(212,240)중의 하나는 목적지(14)를 계산할 필요가 있고, 또한 나머지 그래픽 컨트롤러가 목적지(14) 계산에 필요한 원격 소스화소들의 복사를 했는지를 알게 된다. 그 결과, 그래픽 컨트롤러들(212,240)중의 하나는 나머지 그래픽 컨트롤러(240,212)에 대한 소스인 목적지 화소들을 모두 업데이트할 수 있다. 한쪽 그래픽 컨트롤러의 목적지이면서 다른쪽 그래픽 컨트롤러의 소스는 아닌 모든 화소들은 언제라도 업데이트될 수 있다.

도 7의 3 단계에서, 그래픽 컨트롤러(212,240)는 로컬메모리(160,260)에 저장된 화소이거나 각각의 로컬메모리(160,260)의 스크래치 패드(162,262)에 복사된 화소인 원격 소스화소들에 대해 사용되어, 목적지(14)의 새로운 값을 계산한 다음 이 목적지(14)를 그래픽면(10)에 라이트할 수 있다. 원격 그래픽 메모리로부터의 화소들이 목적지에 포함되어 있으면 이 화소들을 이용할 수도 있다. 예컨대, 로컬메모리(160)에 저장된 화소나 로컬메모리(160)의 스크래치패드(SP)(162)에 복사된 화소인 소스화소들에 대해 내부 (호스트) 그래픽 컨트롤러(212)를 이용해 목적지 화소들을 계산하여, 그래픽면(10)의 소스(12)로부터 목적지(14)가 움직이는 반대 방향으로 화소단위로 스캔할 수도 있다. 예를 들어, 소스(12)

가 도 6에 도시된 것처럼 우측 상방으로 목적지(14)까지 움직이면, 내부 (호스트) 그래픽 컨트롤러(212)는 상부 좌측 모서리에서 스캔을 시작한 다음 좌측하방으로 화소들을 스캔할 수 있다. 마찬가지로, 소스(12)가 목적지(14)의 우측을 지나 위로 움직이면, 내부 (호스트) 그래픽 컨트롤러(212)는 먼저 수직으로 스캔을 시작한 다음 좌측으로 움직일 수 있다.

도 2에 도시된 바와 같이, 소스(12)와 목적지(14)가 오버랩되면, 목적지(14)가 소스(12)에 대해 움직인 특정 방향으로의 스캐닝과 반대방향으로의 사각형 소스의 스캐닝에 의해 오버랩 영역 문제를 간단히 해결할 수 있다. 그 결과, 내외부 그래픽 컨트롤러들(212,240) 사이의 동기화 및 일관성 문제들을 없앨 수 있다.

도 8은 본 발명의 일 실시예에 따른 그래픽 컨트롤러(212,240)와 관련 로컬메모리(160,260)의 블록도이다. 도 8에 도시된 바와 같이, 그래픽 컨트롤러(212,240)는 로컬메모리(160,260)에 대한 접속을 제어하는 로컬메모리 컨트롤러(310), 3D 대상물의 표현으로부터 래스터화된 2D 디스플레이 영상을 생성하는 것을 포함해 다양한 3D 그래픽 기능들을 수행하는 3D (텍스처 매핑) 엔진(312), 그래픽면(10)의 메모리 위치들 사이에 화소데이터를 전송하는 BLT와 관련 동작을 포함해 2D 기능들을 수행하는 그래픽 BLT 엔진(314), 비디오나 그래픽 영상의 비주얼 디스플레이를 제어하는 디스플레이 엔진(316), 로컬메모리(160,260)의 메모리 어드레스에 요청문을 전송하여 BLT와 관련 동작들을 실행하도록 운영시스템(OS) 및 플러그인플레이 장치들과 상호작용하는 라우터(318), BLT 명령어를 포함한 유저 명령어를 해독하고 로컬메모리 컨트롤러(310)와 기타 다른 모든 엔진들(32,314,316)에 제어 스레드(thread)를 보내는 명령어 디코더(320), 및 AGP 버스(40)를 통해 하나 이상의 프로세서(10)와 신호를 송수신하기 위한 인터페이스를 제공하는 인터페이스(322)를 포함한다.

그래픽 BLT 엔진(314)는 명령어 디코더(320)의 제어하에 BLT와 관련 동작들을 요청하고 이 요청을 실행하도록 구성된 다. BLT 동작 요청문은 라우터(318)에 라우팅되고, 이 라우터는 컴퓨터 시스템(100)의 통합 어드레스 공간의 일부인 메모리 어드레스에 상기 요청문을 보낼 수 있다. 메모리 어드레스는 그래픽 컨트롤러(212,240)에 연결된 로컬메모리(160,260)내의 특정 메모리 위치이거나, 또는 컴퓨터 시스템(100)내의 다른 메모리 위치일 수 있다. 메모리 어드레스가 로컬메모리(160,260)내의 특정 메모리 위치이면, 라우터(318)가 이 메모리 어드레스를 라우팅하여 로컬메모리 컨트롤러(310)를 통해 로컬메모리(160,260)에 접근할 수 있다. 한편, 메모리 어드레스가 컴퓨터 시스템(100)내의 다른 메모리 위치이면, 라우터(318)는 인터페이스(322)를 통해 메모리 어드레스를 라우팅할 수 있다.

구체적으로, 그래픽 BLT 엔진(314)은 로컬메모리(160,260)에서 소스(12)를 스캔하고, 목적지(14)를 계산하는데 필요한 모든 소스화소들을 찾으며, 모든 소스화소를 복사하라는 요청문을 로컬메모리(160,260)로 보낼 수 있다. 다음, 도 7에서 설명한 방식대로, 그래픽 BLT 엔진(314)은 목적지 화소들을 계산하고 그래픽면(10)에 이 목적지(14)를 라이트하도록 복사가 이루어졌음을 표시하는 동기 라이트를 대기한다.

산업상 이용 가능성

이상 설명한 바와 같이, 본 발명은 특히 BLT 동작중에 오버랩되는 소스와 목적지 영역들을 취급할 때 두개의 그래픽 컨트롤러들이 각각 적당한 동기화와 일관성을 가지고 컴퓨터 시스템내의 싱글 BLT 동작의 일부분을 병렬로 실행할 수 있게 하는 메커니즘과 방법을 제공한다는 점에서 효과적이다.

이상 본 발명의 몇가지 실시예들에 대해 설명했지만, 당업자라면 본 발명의 범위를 벗어나지 않고도 다양한 변형이나 변경이 가능할 것이다. 본 발명의 범위를 벗어나지 않고 특정 상황에 대해 본 발명의 기술을 이용해 많은 변형이 가능할 수 있다. 예컨대, 두개의 그래픽 컨트롤러 각각이 싱글 BLT 동작의 일부를 병렬로 행하게 하는 메커니즘은 각각의 그래픽 컨트롤러에서 원격 소스화소의 스크래치패드 복사를 하도록 하고, 동기 라이트를 보내며, BLT와 관련 동작들을 실행하도록 구성된 드라이버 소프트웨어를 갖춘 하드웨어/소프트웨어 모듈이나 소프트웨어 모듈에 의해 구현될 수도 있다. 따라서, 본 발명은 이상 설명한 실시예들에 한정되지 않으며, 첨부된 특허청구범위에 속하는 모든 실시예들을 포함한다.

(57) 청구의 범위

청구항 1.

블록변환 "BLT" 동작의 일부를 병렬로 실행하여 화소 데이터 블록을 디스플레이 스크린의 그래픽면상의 소스로부터 목적지로 전송하는 기능을 포함해 그래픽 기능과 비디오 기능을 공유하도록 구성된 제1 및 제2 그래픽 컨트롤러들;

상기 제1 및 제2 그래픽 컨트롤러들에 연결되고, 지정된 패턴으로 이들 제1 및 제2 그래픽 컨트롤러들에 할당된 그래픽면에 상기 소스의 화소데이터를 저장하도록 구성된 메모리소자; 및

상기 BLT 동작을 실행하라는 요청이 있을 때, 다른쪽 그래픽 컨트롤러가 제어하는 영역에 있고 상기 메모리 소자로부터 복사된 소스의 모든 화소 데이터를 각각 저장하기 위한 스크래치 패드들;을 포함하는 것을 특징으로 하는 그래픽 장치.

청구항 2.

제1항에 있어서, 상기 메모리소자가,

상기 제1 그래픽 컨트롤러에 연결되고, 지정된 패턴으로 제1 그래픽 컨트롤러에 할당된 그래픽면에 소스의 화소데이터를 저장하도록 구성된 제1 로컬메모리; 및

상기 제2 그래픽 컨트롤러에 연결되고, 상기 지정된 패턴으로 제2 그래픽 컨트롤러에 할당된 그래픽면에 소스의 화소데이터를 저장하도록 구성된 제2 로컬메모리;를 포함하는 것을 특징으로 하는 그래픽 장치.

청구항 3.

제2항에 있어서, 상기 BLT 동작을 실행하라는 요청이 있을 때, 다른쪽 그래픽 컨트롤러가 제어하는 영역에 있고 다른쪽 로컬메모리로부터 복사된 상기 소스의 모든 화소데이터를 저장하도록, 상기 스크래치 패드들이 각각 제1 및 제2 로컬메모리에 포함되어 있는 것을 특징으로 하는 그래픽 장치.

청구항 4.

제1항에 있어서, 상기 BLT 동작은 그래픽면의 상기 목적지의 화소데이터를 얻기위해 상기 소스의 화소데이터에 대한 논리적 동작과 기타 OPERAND를 포함하는 것을 특징으로 하는 그래픽 장치.

청구항 5.

제2항에 있어서, 상기 BLT 동작은 그래픽면의 상기 목적지의 화소데이터를 얻기위해 상기 소스의 화소데이터에 대한 논리적 동작과 기타 OPERAND를 포함하는 것을 특징으로 하는 그래픽 장치.

청구항 6.

제1항에 있어서, 상기 제1 그래픽 컨트롤러는 칩셋에 내장되고, 상기 제2 그래픽 컨트롤러는 발전된 그래픽 애플리케이션용으로 확장카드에 플러그인되는 것을 특징으로 하는 그래픽 장치.

청구항 7.

제6항에 있어서, 상기 제1 및 제2 그래픽 컨트롤러들이 각각 BLT와 관련 동작들을 실행하도록 구성된 BLT 엔진을 포함하는 것을 특징으로 하는 그래픽 장치.

청구항 8.

제6항에 있어서, 상기 제1 및 제2 그래픽 컨트롤러들은 각각 먼저 다른쪽 그래픽 컨트롤러가 제어하는 영역에 있는 상기 소스의 모든 화소데이터를 각각의 스크래치 패드에 복사한 다음, 다른쪽 그래픽 컨트롤러에 동기 라이트를 보내 복사가 이루어졌음을 지적하며, 다른쪽 그래픽 컨트롤러로부터 동기 라이트를 수신했을 때 다른쪽 그래픽 컨트롤러의 소스인 상기 목적지의 모든 화소데이터를 업데이트하기 시작하는 것을 특징으로 하는 그래픽 장치.

청구항 9.

제8항에 있어서, 상기 제1 및 제2 그래픽 컨트롤러 각각은 다른쪽 그래픽 컨트롤러의 소스가 아닌 상기 목적지의 모든 화소데이터를 언제라도 업데이트할 수 있는 것을 특징으로 하는 그래픽 장치.

청구항 10.

제8항에 있어서, 상기 제1 및 제2 그래픽 컨트롤러들 각각은 이들 컨트롤러 각각에 상기 지정된 패턴으로 할당된 소스의 화소데이터를 이용하거나, 복사된 소스의 화소데이터를 이용해 상기 목적지의 새로운 값을 계산하고, 상기 지정된 패턴의 그래픽면에 상기 목적지를 라이트하는 것을 특징으로 하는 그래픽 장치.

청구항 11.

제8항에 있어서, 상기 제1 및 제2 그래픽 컨트롤러들 각각이,

각각의 로컬메모리에 대한 접근을 제어하는 로컬메모리 컨트롤러;

3D 대상물의 표현으로부터 래스터화된 2D 디스플레이 영상을 형성하는 것을 포함해 다양한 3D 그래픽 기능들을 실행하는 3D (텍스처 매핑) 엔진;

그래픽면의 소스로부터 목적지로 화소데이터 블록을 전송하는 상기 BLT 동작을 포함해 2D 기능들을 실행하는 그래픽 BLT 엔진;

비디오영상이나 그래픽영상의 비주얼 디스플레이를 제어하는 디스플레이 엔진;

상기 로컬메모리 컨트롤러, 3D 엔진, 그래픽 BLT 엔진 및 디스플레이 엔진에 연결되고, 운영시스템(OS)과 연계되어 상기 로컬메모리의 메모리 어드레스에 요청문들을 변환하여 상기 BLT 동작을 실행하기 위한 라우터;

BLT 명령어를 포함해 유저 명령어들을 해독하고, 상기 로컬메모리 컨트롤러, 3D 엔진, 그래픽 BLT 엔진, 디스플레이 엔진에 제어 스레드를 보내는 명령어 디코더; 및

하나 이상의 프로세서와 신호를 주고받기 위한 인터페이스를 제공하는 인터페이스;를 포함하는 것을 특징으로 하는 그래픽 장치.

청구항 12.

제1항에 있어서, 그래픽면의 상기 지정된 패턴이 바둑판 무늬에 해당하고, 바둑판 무늬중 2칸은 상기 제1 그래픽 컨트롤러에 할당되며, 바둑판 무늬중 다른 2칸은 상기 제2 그래픽 컨트롤러에 할당되는 것을 특징으로 하는 그래픽 장치.

청구항 13.

하나 이상의 프로세서;

디스플레이 스크린이 달린 디스플레이 모니터;

상기 프로세서에 연결되고, 디스플레이 모니터의 비주얼 디스플레이를 위해 비디오 데이터를 처리하는 내부 그래픽 컨트롤러와, 내부 그래픽 컨트롤러에 연결된 로컬메모리를 갖는 칩셋; 및

확장카드를 통해 상기 칩셋에 연결되고, 화소데이터 블록을 디스플레이 스크린의 그래픽면의 소스로부터 목적지로 전송하도록 블록변환 "BLT" 동작의 일부분을 병렬로 실행하는 것을 포함해, 상기 칩셋의 내부 그래픽 컨트롤러와 함께 그래픽과 비디오를 공유하도록 구성된 외부그래픽 컨트롤러 및 로컬메모리;를 포함하고,

상기 내외부 그래픽 컨트롤러들 각각의 로컬메모리는 지정된 패턴으로 각각의 그래픽 컨트롤러에 할당된 그래픽면의 소스의 화소데이터를 저장하도록 구성되고, 상기 BLT 동작을 실행하라는 요청이 있을 때 다른쪽 그래픽 컨트롤러가 제어하는 영역에 있으면서 다른쪽 로컬메모리로부터 복사된 소스의 모든 화소데이터를 저장하는 스크래치 패드를 포함하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 14.

제13항에 있어서, 상기 BLT 동작은 그래픽면의 상기 목적지의 화소데이터를 얻기위해 상기 소스의 화소데이터에 대한 논리적 동작과 기타 OPERAND를 포함하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 15.

제13항에 있어서, 상기 내외부 그래픽 컨트롤러들 각각은 BLT와 관련 동작들을 실행하도록 구성된 BLT 그래픽엔진을 포함하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 16.

제13항에 있어서, 상기 내외부 그래픽 컨트롤러들은 각각 먼저 다른쪽 그래픽 컨트롤러가 제어하는 영역에 있는 상기 소스의 모든 화소데이터를 각각의 스크래치 패드에 복사한 다음, 다른쪽 그래픽 컨트롤러에 동기 라이트를 보내 복사가 이루어졌음을 지적하며, 다른쪽 그래픽 컨트롤러로부터 동기 라이트를 수신했을 때 다른쪽 그래픽 컨트롤러의 소스인 상기 목적지의 모든 화소데이터를 업데이트하기 시작하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 17.

제16항에 있어서, 상기 내외부 그래픽 컨트롤러 각각은 다른쪽 그래픽 컨트롤러의 소스가 아닌 상기 목적지의 모든 화소데이터를 언제라도 업데이트할 수 있는 것을 특징으로 하는 컴퓨터 시스템.

청구항 18.

제17항에 있어서, 상기 내외부 그래픽 컨트롤러들 각각은 이들 컨트롤러 각각에 상기 지정된 패턴으로 할당된 소스의 화소데이터를 이용하거나, 복사된 소스의 화소데이터를 이용해 상기 목적지의 새로운 값을 계산하고, 상기 지정된 패턴의 그래픽면에 상기 목적지를 라이트하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 19.

제18항에 있어서, 상기 내외부 그래픽 컨트롤러들 각각이,

각각의 로컬메모리에 대한 접근을 제어하는 로컬메모리 컨트롤러;

3D 대상물의 표현으로부터 래스터화된 2D 디스플레이 영상을 형성하는 것을 포함해 다양한 3D 그래픽 기능들을 실행하는 3D (텍스처 매핑) 엔진;

그래픽면의 소스로부터 목적지로 화소데이터 블록을 전송하는 상기 BLT 동작을 포함해 2D 기능들을 실행하는 그래픽 BLT 엔진;

비디오영상이나 그래픽영상의 비주얼 디스플레이를 제어하는 디스플레이 엔진;

상기 로컬메모리 컨트롤러, 3D 엔진, 그래픽 BLT 엔진 및 디스플레이 엔진에 연결되고, 운영시스템(OS)과 연계되어 상기 로컬메모리의 메모리 어드레스로의 요청문들을 변환하여 상기 BLT 동작을 실행하기 위한 라우터;

BLT 명령어를 포함해 유저 명령어들을 해독하고, 상기 로컬메모리 컨트롤러, 3D 엔진, 그래픽 BLT 엔진, 디스플레이 엔진에 제어 스트림을 보내는 명령어 디코더; 및

하나 이상의 프로세서와 신호를 주고받기 위한 인터페이스를 제공하는 인터페이스;를 포함하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 20.

제13항에 있어서, 그래픽면의 상기 지정된 패턴이 바둑판 무늬에 해당하고, 바둑판 무늬중 2칸은 상기 내부그래픽 컨트롤러에 할당되며, 바둑판 무늬중 다른 2칸은 상기 외부그래픽 컨트롤러에 할당되는 것을 특징으로 하는 컴퓨터 시스템.

청구항 21.

컴퓨터 시스템내의 다수의 그래픽 컨트롤러들 각각이 블록변환 "BLT" 동작의 일부를 병렬로 실행하도록 하는 방법이 있어서:

지정된 패턴의 그래픽면의 소스로부터 목적지로 화소데이터 블록을 전송하도록 상기 BLT 동작을 실행하라는 요청을 받았을 때, 각각의 그래픽 컨트롤러로 하여금 다른쪽 그래픽 컨트롤러가 제어하는 영역에 있는 모든 화소데이터를 로컬메모리에 복사하도록 하는 단계;

각각의 그래픽 컨트롤러로 하여금 복사가 이루어졌음을 표시하도록 동기 라이트를 보내도록 하는 단계; 및

다른쪽 그래픽 컨트롤러로부터 상기 동기 라이트를 받았을 때, 각각의 그래픽 컨트롤러로 하여금 다른쪽 그래픽 컨트롤러의 소스인 목적지 화소들을 전부 업데이트하고 상기 BLT 동작을 실행하게 하는 단계;를 포함하는 것을 특징으로 하는 방법.

청구항 22.

제21항에 있어서, 상기 BLT 동작은 그래픽면의 상기 목적지의 화소데이터를 얻기 위해 상기 소스의 화소데이터에 대한 논리적 동작과 기타 OPERAND를 포함하는 것을 특징으로 하는 방법.

청구항 23.

제21항에 있어서, 상기 다수의 그래픽 컨트롤러들 각각이 다른쪽 그래픽 컨트롤러의 소스가 아닌 상기 목적지의 모든 화소데이터를 언제라도 업데이트할 수 있는 것을 특징으로 하는 방법.

청구항 24.

제21항에 있어서, 그래픽면의 상기 지정된 패턴이 바둑판 무늬에 해당하고, 바둑판 무늬중 2칸은 한쪽 그래픽 컨트롤러에 할당되며, 바둑판 무늬중 다른 2칸은 다른쪽 그래픽 컨트롤러에 할당되는 것을 특징으로 하는 방법.

청구항 25.

로컬메모리; 및

다수의 그래픽컨트롤러들에 할당된 지정된 패턴으로 디스플레이 스크린상의 소스로부터 목적지로 화소 데이터 블록을 전달하도록 블록변환 "BLT" 동작의 일부를 병렬로 각각 실행하고, 그래픽기능과 비디오기능을 공유하는 다수의 그래픽컨트롤러들;을 포함하고,

각각의 그래픽컨트롤러는, 상기 BLT 동작을 실행하라는 요청이 있을 때, 다른 그래픽 엔진이 제어하는 영역에 있는 상기 소스의 화소데이터를 각각의 로컬메모리에 먼저 복사하고, 다른 그래픽 엔진에 동기 라인을 보내 복사가 이루어졌음을 알려주며, 다른 그래픽 엔진으로부터 동기 라인을 수신했을 때 다른 그래픽 엔진의 소스인 상기 목적지의 모든 화소데이터를 업데이트하기 시작하는 것을 특징으로 하는 장치.

청구항 26.

제25항에 있어서, 상기 그래픽컨트롤러들 모두는 다른 그래픽컨트롤러의 소스가 아닌 상기 목적지의 모든 화소데이터를 언제라도 업데이트할 수 있는 것을 특징으로 하는 장치.

청구항 27.

제25항에 있어서, 상기 그래픽컨트롤러들 각각은 이들 그래픽컨트롤러 각각에 할당된 상기 지정된 패턴의 상기 소스의 화소데이터를 이용하거나 복사된 소스의 화소데이터를 이용해 목적지의 새로운 값을 계산하고, 상기 지정된 패턴의 그래픽면에 상기 목적지를 라이트하는 것을 특징으로 하는 장치.

청구항 28.

제25항에 있어서, 상기 그래픽컨트롤러들 각각이,

각각의 로컬메모리에 대한 접근을 제어하는 로컬메모리 컨트롤러;

3D 대상물의 표현으로부터 래스터화된 2D 디스플레이 영상을 형성하는 것을 포함해 다양한 3D 그래픽 기능들을 실행하는 3D (텍스처 매핑) 엔진;

그래픽면의 소스로부터 목적지로 화소데이터 블록을 전송하는 상기 BLT 동작을 포함해 2D 기능들을 실행하는 그래픽 BLT 엔진;

비디오영상이나 그래픽영상의 비주얼 디스플레이를 제어하는 디스플레이 엔진;

상기 로컬메모리 컨트롤러, 3D 엔진, 그래픽 BLT 엔진 및 디스플레이 엔진에 연결되고, 운영시스템(OS)과 연계되어 상기 로컬메모리의 메모리 어드레스로의 요청문들을 변환하여 상기 BLT 동작을 실행하기 위한 라우터;

BLT 명령어를 포함해 유저 명령어들을 해독하고, 상기 로컬메모리 컨트롤러, 3D 엔진, 그래픽 BLT 엔진, 디스플레이 엔진에 제어 스트림을 보내는 명령어 디코더; 및

하나 이상의 프로세서와 신호를 주고받기 위한 인터페이스를 제공하는 인터페이스;를 포함하는 것을 특징으로 하는 장치.

청구항 29.

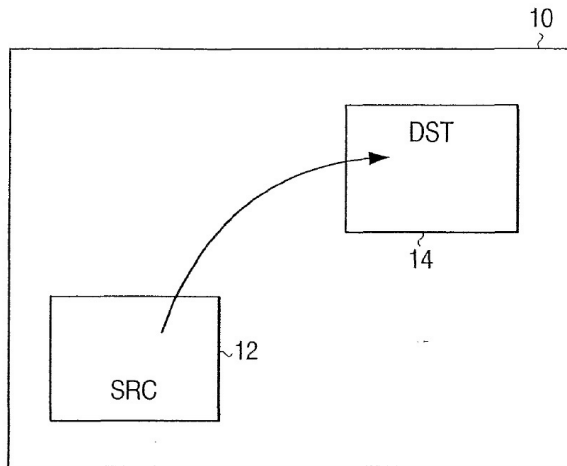
제25항에 있어서, 그래픽면의 상기 지정된 패턴이 바둑판 무늬에 해당하고, 바둑판 무늬의 ½은 한쪽 그래픽컨트롤러에 할당되며, 바둑판 무늬의 나머지 ½은 다른쪽 그래픽컨트롤러에 할당되는 것을 특징으로 하는 장치.

청구항 30.

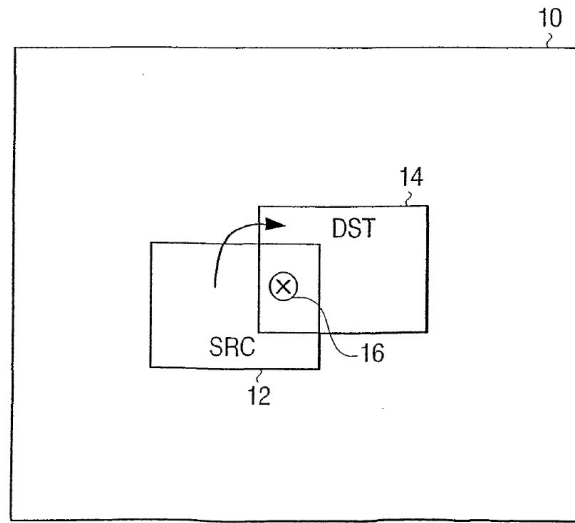
제25항에 있어서, 상기 BLT 동작은 그래픽면의 상기 목적지의 화소데이터를 얻기위해 상기 소스의 화소데이터에 대한 논리적 동작과 기타 OPERAND를 포함하는 것을 특징으로 하는 장치.

도면

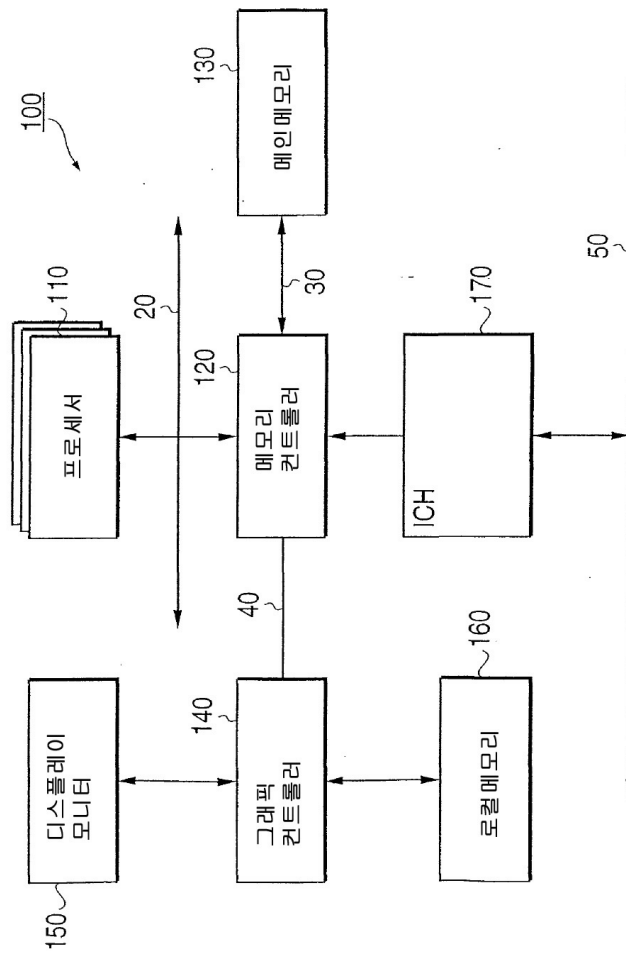
도면1



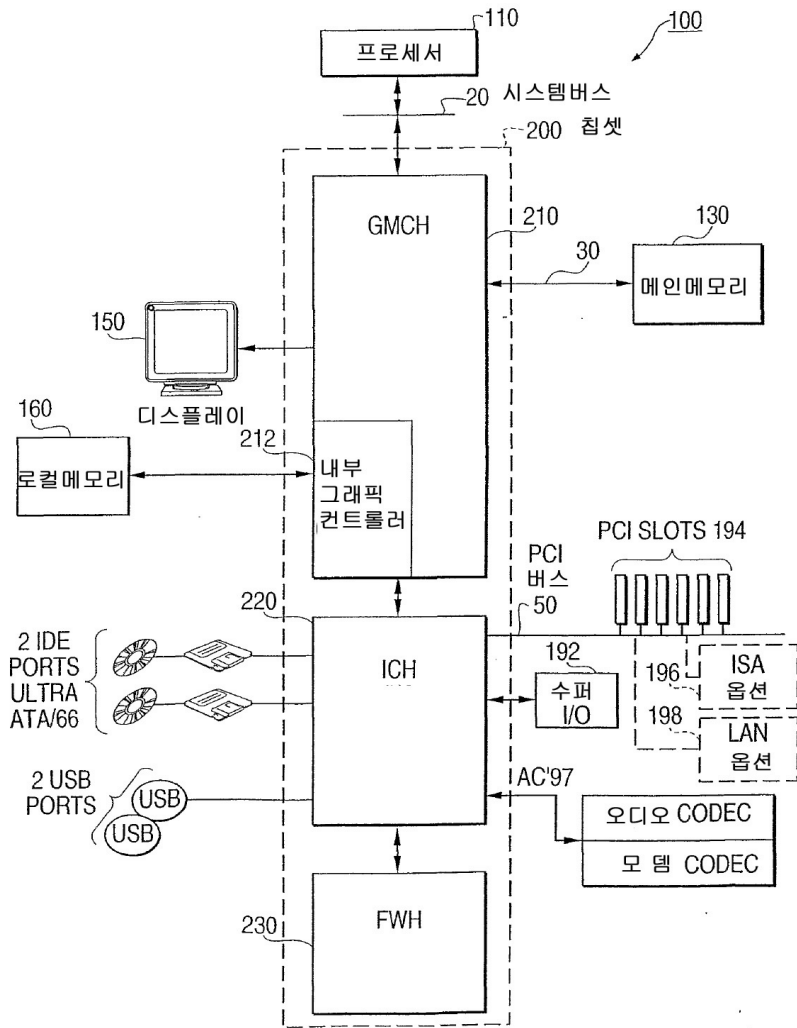
도면2



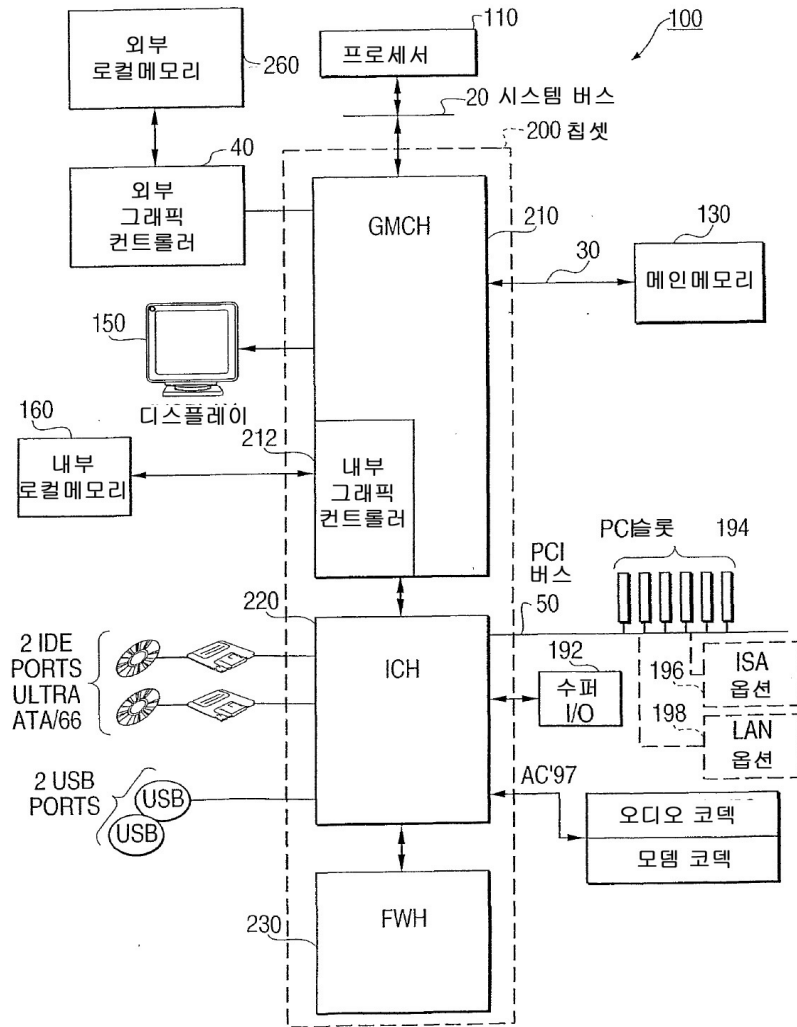
도면3



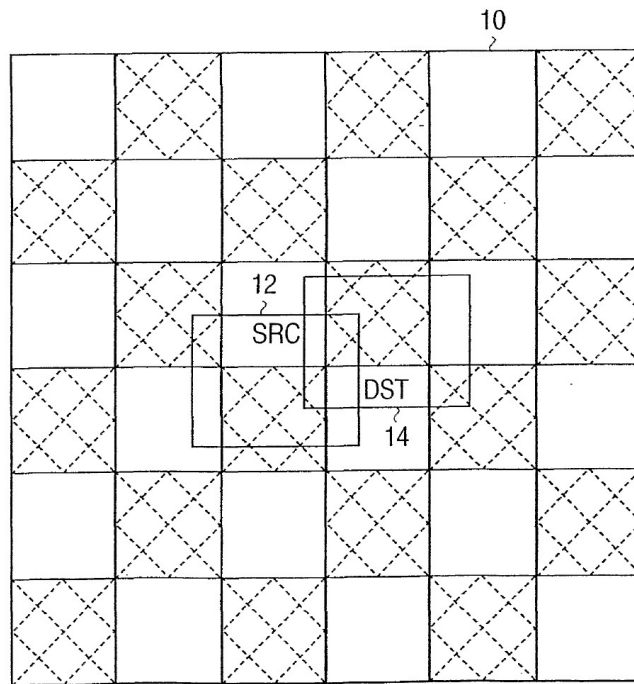
도면4



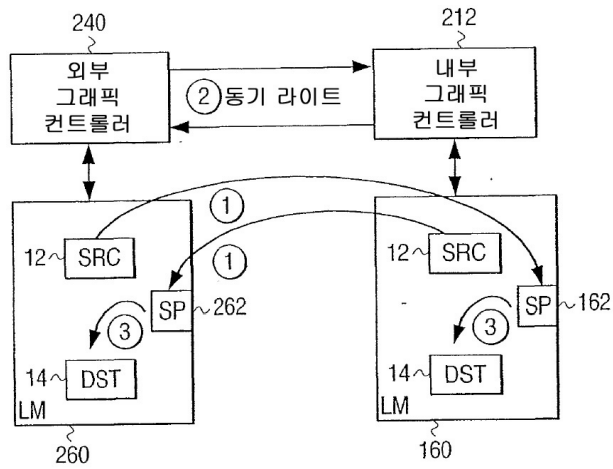
도면5



도면6



도면7



도면8

