



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년03월06일
(11) 등록번호 10-2644276
(24) 등록일자 2024년02월29일

- (51) 국제특허분류(Int. Cl.)
G06T 1/20 (2018.01) G06F 9/38 (2024.01)
G06T 1/60 (2006.01)
- (52) CPC특허분류
G06T 1/20 (2013.01)
G06F 9/3885 (2013.01)
- (21) 출원번호 10-2016-0130832
- (22) 출원일자 2016년10월10일
심사청구일자 2021년10월07일
- (65) 공개번호 10-2018-0039479
- (43) 공개일자 2018년04월18일
- (56) 선행기술조사문헌
US20080129744 A1*
KR1020150002742 A*
US20130120413 A1*
KR1020140072097 A
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
- (72) 발명자
진승훈
경기도 수원시 장안구 율전로101번길 40, 511호
(율전동, 늘푸른벽산APT)
아브라함 산토쉬 조지
미국, 캘리포니아주 94566, 플레젠튼, 4776 아만
다 플레이스
- (74) 대리인
리엔목특허법인

전체 청구항 수 : 총 10 항

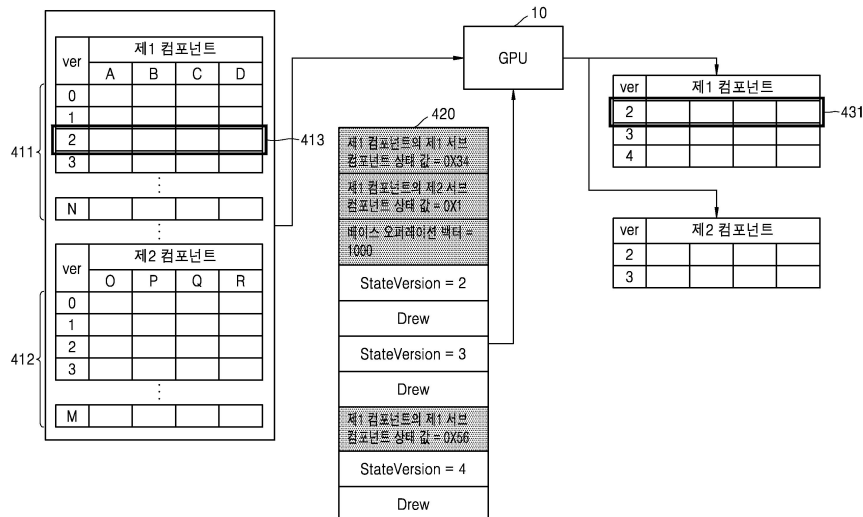
심사관 : 이후락

(54) 발명의 명칭 그래픽 처리 장치 및 방법

(57) 요약

그래픽 처리 장치는, 복수의 컴포넌트들; 컴퓨터 실행가능 명령어를 저장하는 메모리; 및 상기 컴퓨터 실행가능 명령어를 실행함으로써, 상기 복수의 컴포넌트들의 베이스 상태를 나타내는 베이스 오퍼레이션 데이터에 기초하여, 그래픽스 파이프라인을 수행하기 위한 오퍼레이션 데이터를 생성하는 적어도 하나의 프로세서를 포함하고, 상기 복수의 컴포넌트들은 상기 생성된 오퍼레이션 데이터에 따라, 상기 그래픽스 파이프라인을 수행할 수 있다.

대표도



(52) CPC특허분류

G06T 1/60 (2013.01)

G06T 2200/28 (2013.01)

명세서

청구범위

청구항 1

복수의 컴포넌트들;

컴퓨터 실행가능 명령어(computer executable instruction)를 저장하는 메모리; 및

상기 컴퓨터 실행가능 명령어를 실행함으로써,

그래픽스 파이프라인이 수행될 때 가장 사용 빈도가 높은 상기 복수의 컴포넌트들의 상태를 상기 복수의 컴포넌트들의 베이스 상태를 나타내는 베이스 오퍼레이션 데이터로 생성하고,

상기 베이스 오퍼레이션 데이터를 이용하여, 그래픽스 파이프라인을 수행하기 위한 오퍼레이션 데이터를 생성하는 적어도 하나의 프로세서를 포함하고,

상기 복수의 컴포넌트들은 상기 생성된 오퍼레이션 데이터에 따라, 상기 그래픽스 파이프라인을 수행하는, 그래픽 처리 장치.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 적어도 하나의 프로세서는,

상기 복수의 컴포넌트들의 제1 상태에서 상기 그래픽스 파이프라인을 수행하는 데에 사용되는 제1 상태 버전 코드를 포함하는 제1 오퍼레이션 데이터를 획득하고,

상기 베이스 오퍼레이션 데이터에 기초하여, 상기 제1 오퍼레이션 데이터를 상기 복수의 컴포넌트들의 제2 상태에서 사용되는 제2 상태 버전 코드를 포함하는 제2 오퍼레이션 데이터로 업데이트하는, 그래픽 처리 장치.

청구항 4

제1항에 있어서,

상기 복수의 컴포넌트들 각각은 복수의 서브 컴포넌트들을 포함하고,

상기 복수의 컴포넌트들 각각은 상기 복수의 컴포넌트들 각각의 상태에서 사용되는 코드를 포함하는 서브 오퍼레이션 데이터를 획득하고, 상기 복수의 서브 컴포넌트들 각각은 상기 서브 오퍼레이션 데이터에 따라 상기 그래픽스 파이프라인을 수행하는, 그래픽 처리 장치.

청구항 5

제1항에 있어서,

상기 적어도 하나의 프로세서는,

상기 베이스 오퍼레이션 데이터를 상기 메모리 또는 상기 복수의 컴포넌트들의 소정의 공간에 저장되도록 제어하는, 그래픽 처리 장치.

청구항 6

제3항에 있어서,

상기 적어도 하나의 프로세서는,

상기 복수의 컴포넌트들 각각의 베이스 오퍼레이션 데이터 중 상기 제2 상태를 결정하는 데에 사용되는 적어도

하나의 컴포넌트의 베이스 오퍼레이션 데이터를 획득하고,

상기 적어도 하나의 컴포넌트의 베이스 오퍼레이션 데이터에 기초하여, 상기 제1 오퍼레이션 데이터를 상기 제2 오퍼레이션 데이터로 업데이트하는, 그래픽 처리 장치.

청구항 7

제6항에 있어서,

상기 적어도 하나의 프로세서는,

상기 복수의 컴포넌트들 중에서, 상기 그래픽스 파이프라인이 수행될 때 컴포넌트에 대한 상태의 변경 빈도가 가장 높은 제1 컴포넌트를 결정하고, 상기 제2 상태에서 사용되는 제1 컴포넌트의 오퍼레이션 코드에 기초하여, 상기 베이스 오퍼레이션 데이터를 설정하는, 그래픽 처리 장치.

청구항 8

제3항에 있어서,

상기 메모리는, 상기 업데이트된 제2 오퍼레이션 데이터를 저장하고,

상기 복수의 컴포넌트들은 상기 업데이트된 제2 오퍼레이션 데이터에 따라, 상기 그래픽스 파이프라인을 수행하는, 그래픽 처리 장치.

청구항 9

삭제

청구항 10

복수의 컴포넌트들;

상기 복수의 컴포넌트들의 복수의 상태 각각에서 그래픽스 파이프라인을 수행하는 데에 사용되는 코드를 포함하는 복수의 상태 버전 데이터를 저장하는 메모리; 및

상기 복수의 상태 버전 데이터의 사용 빈도에 기초하여, 상기 복수의 상태 버전 데이터의 우선 순위를 결정하고, 상기 우선 순위에 기초하여, 상기 메모리에 저장된 복수의 상태 버전 데이터 중 우선 순위가 가장 높은 제1 상태 버전 데이터를 보존하고, 상기 제1 상태 버전 데이터보다 우선 순위가 낮은 제2 상태 버전 데이터를 그래픽스 파이프라인이 수행됨에 따라 업데이트 되도록 제어하는 적어도 하나의 프로세서를 포함하는, 그래픽 처리 장치.

청구항 11

제10항에 있어서,

상기 적어도 하나의 프로세서는, 상기 그래픽스 파이프라인에 사용될 상태 버전 데이터를 결정하고, 상기 결정된 상태 버전 데이터에 따라, 상기 그래픽스 파이프라인을 수행하는, 그래픽 처리 장치.

청구항 12

그래픽스 파이프라인을 수행하는 복수의 컴포넌트들을 포함하는 그래픽 처리 장치의 동작 방법에 있어서,

상기 그래픽스 파이프라인이 수행될 때 가장 변경 빈도가 높은 상기 복수의 컴포넌트들의 상태를 상기 복수의 컴포넌트들의 베이스 상태를 나타내는 베이스 오퍼레이션 데이터로 생성하는 단계;

상기 베이스 오퍼레이션 데이터에 기초하여, 상기 복수의 컴포넌트들의 소정 상태에서 상기 그래픽스 파이프라인을 수행하는 데에 사용되는 코드를 포함하는 오퍼레이션 데이터를 생성하는 단계; 및

상기 생성된 오퍼레이션 데이터에 따라, 상기 그래픽스 파이프라인을 수행하는 단계를 포함하는, 그래픽 처리 장치의 동작 방법.

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

발명의 설명

기술 분야

[0001] 그래픽 처리 장치 및 방법에 관한 것이다.

배경 기술

[0002] GPU(Graphic Processing Unit)와 같은 그래픽 처리 장치는 컴퓨팅 장치에서 그래픽스 데이터를 렌더링하는 역할을 담당한다. 일반적으로, 그래픽 처리 장치는 2차원 또는 3차원 객체들에 해당되는 그래픽스 데이터를 2차원 픽셀 표현으로 변환하여 디스플레이를 위한 프레임을 생성한다. 컴퓨팅 장치의 종류로는, PC (personal computers), 노트북, 비디오 게임용 콘솔뿐만 아니라, 스마트폰, 태블릿 디바이스, 웨어러블 디바이스와 같은 임베디드(embedded) 디바이스들도 포함될 수 있다. 스마트폰, 태블릿 디바이스, 웨어러블 디바이스 등과 같은 임베디드 디바이스들은 비교적 낮은 연산 처리 능력과 많은 전력 소모의 문제들로 인하여, 충분한 메모리 공간과 프로세싱 파워를 확보하고 있는 PC, 노트북, 비디오 게임용 콘솔 등과 같은 워크스테이션들과 동일한 그래픽 프로세싱 성능을 갖추기 어렵다. 하지만, 최근 전세계적으로 스마트폰 또는 태블릿 디바이스와 같은 휴대용 디바이스들이 널리 보급됨에 따라 사용자들은 스마트폰 또는 태블릿 디바이스를 통해 게임을 플레이하거나 또는 영화, 드라마 등의 콘텐츠를 감상하는 빈도가 급격히 증가하였다. 이에 따라, 그래픽 처리 장치의 제조사들은 사용자들의 수요에 발맞추어 임베디드 디바이스에서도 그래픽 처리 장치의 성능 및 처리 효율을 높이기 위한 많은 연구들을 진행하고 있다.

발명의 내용

해결하려는 과제

[0003] 그래픽 처리 장치 및 그래픽 처리 장치에서 그래픽스 파이프라인을 수행하는 방법을 제공하는 데에 있다. 또한, 그래픽을 처리하는 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체를 제공하는 데 있다. 일실시예들이 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되

지 않으며, 이하의 실시예들로부터 또 다른 기술적 과제들이 유추될 수 있다.

과제의 해결 수단

[0004] 일측에 따르면, 복수의 컴포넌트들; 컴퓨터 실행가능 명령어(computer executable instruction)를 저장하는 메모리; 및 상기 컴퓨터 실행가능 명령어를 실행함으로써, 상기 복수의 컴포넌트들의 베이스 상태를 나타내는 베이스 오퍼레이션 데이터를 이용하여, 그래픽스 파이프라인을 수행하기 위한 오퍼레이션 데이터를 생성하는 적어도 하나의 프로세서를 포함하고, 상기 복수의 컴포넌트들은 상기 생성된 오퍼레이션 데이터에 따라, 상기 그래픽스 파이프라인을 수행하는, 그래픽 처리 장치가 제공된다.

[0005] 다른 일측에 따르면, 복수의 컴포넌트들; 상기 복수의 컴포넌트들의 복수의 상태 각각에서 그래픽스 파이프라인을 수행하는 데에 사용되는 코드를 포함하는 복수의 상태 버전 데이터를 저장하는 메모리; 및 상기 복수의 상태 버전 데이터의 사용 빈도에 기초하여, 상기 복수의 상태 버전 데이터의 우선 순위를 결정하고, 상기 우선 순위에 기초하여, 상기 메모리에 저장된 복수의 상태 버전 데이터 중 우선 순위가 가장 높은 제1 상태 버전 데이터를 보존하고, 상기 제1 상태 버전 데이터보다 우선 순위가 낮은 제2 상태 버전 데이터를 그래픽스 파이프라인이 수행됨에 따라 업데이트 되도록 제어하는 적어도 하나의 프로세서를 포함하는, 그래픽 처리 장치가 제공된다.

[0006] 또 다른 일측에 따르면, 그래픽스 파이프라인을 수행하는 복수의 컴포넌트들을 포함하는 그래픽 처리 장치의 동작 방법에 있어서, 상기 복수의 컴포넌트들의 상태의 변경 빈도에 기초하여, 상기 복수의 컴포넌트들의 베이스 상태를 나타내는 베이스 오퍼레이션 데이터를 생성하는 단계; 상기 베이스 오퍼레이션 데이터에 기초하여, 상기 복수의 컴포넌트들의 소정 상태에서 상기 그래픽스 파이프라인을 수행하는 데에 사용되는 코드를 포함하는 오퍼레이션 데이터를 생성하는 단계; 및 상기 생성된 오퍼레이션 데이터에 따라, 상기 그래픽스 파이프라인을 수행하는 단계를 포함할 수 있다.

도면의 간단한 설명

- [0007] 도 1은 일실시예에 따른 컴퓨팅 시스템의 블록도이다.
- 도 2는 일실시예에 따라, 그래픽 처리 장치(Graphic Processing Unit, GPU)에 의해 처리되는 그래픽스 파이프라인을 설명하는 도면이다.
- 도 3은 일실시예에 따른 도 1의 GPU의 상세 하드웨어 구조를 도시한 블록도이다.
- 도 4는 일실시예에 따라, 그래픽 처리 장치의 동작 방법을 설명하기 위한 개념도이다.
- 도 5a 및 도 5b는 일실시예에 따라, 그래픽 처리 장치의 제1 컴포넌트가 소정 저장 공간에 저장된 오퍼레이션 데이터를 이용하여 그래픽스 파이프라인을 수행하는 과정을 설명하기 위한 도면이다.
- 도 6은 도 1에 도시된 그래픽 처리 장치의 구성을 나타낸 블록도이다.
- 도 7은 그래픽 처리 장치에서 그래픽스 파이프라인을 수행하는 복수의 컴포넌트를 설명하기 위한 도면이다.
- 도 8은 일실시예에 따라, 그래픽 처리 장치의 동작 방법을 설명하기 위한 도면이다.
- 도 9는 일실시예에 따라, 복수의 컴포넌트들 각각에 대한 상태 데이터의 사용 빈도에 따라 베이스 오퍼레이션 데이터를 설정하는 과정을 설명하기 위한 도면이다.
- 도 10은 일실시예에 따라, 복수의 컴포넌트들 각각에 대한 상태 버전 데이터의 사용 빈도에 따라, 그래픽 처리 장치의 상태 버전 데이터를 처리하는 방법을 설명하기 위한 도면이다.
- 도 11은 일실시예에 따라, 그래픽 처리 장치의 동작 방법의 흐름을 나타낸 순서도이다.
- 도 12는 일실시예에 따라, 도 13에 도시된 그래픽 처리 장치의 동작 방법을 상세하게 설명하기 위한 순서도이다.
- 도 13은 다른 일실시예에 따라, 그래픽 처리 장치의 동작 방법의 흐름을 나타낸 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0008] 이하에서, 일부 실시예들을, 첨부된 도면을 참조하여 상세하게 설명한다. 그러나, 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

- [0009] 아래 설명에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도 또는 관례, 새로운 기술의 출현 등에 따라 달라질 수 있다.
- [0010] 또한 특정한 경우는 이해를 돕거나 및/또는 설명의 편의를 위해 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 설명 부분에서 상세한 그 의미를 기재할 것이다. 따라서 아래 설명에서 사용되는 용어는 단순한 용어의 명칭이 아닌 그 용어가 가지는 의미와 명세서 전반에 걸친 내용을 토대로 이해되어야 한다.
- [0011] 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0012] 본 명세서에서 사용되는 '구성된다' 또는 '포함한다' 등의 용어는 명세서 상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다. 또한, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0013] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다.
- [0014] 이하, 실시예를 첨부된 도면을 참조하여 상세하게 설명한다. 본 실시예들은 그래픽스 파이프라인을 처리하는 방법 및 장치에 관한 것으로서 이하의 실시예들이 속하는 기술 분야에서 통상의 지식을 가진 자에게 널리 알려져 있는 사항들에 관해서는 자세한 설명을 생략한다.
- [0015] 도 1은 일실시예에 따른 컴퓨팅 시스템의 블록도이다.
- [0016] 도 1을 참고하면, 컴퓨팅 시스템(1)은 GPU(Graphic Processing Unit)(10), CPU(Central Processing Unit)(20), 메모리(30) 및 Bus(40)를 포함한다. GPU(10) 및 CPU(20)는 프로세싱 장치들에 해당되는 것들로서, 프로세싱 장치들의 범주에는 GPU(10) 및 CPU(20) 외에도 다른 종류의 프로세서들이 포함될 수 있다. 한편, 도 1에 도시된 컴퓨팅 시스템(1)에는 실시예들과 관련된 구성요소들만이 도시되어 있다. 따라서, 도 1에 도시된 구성요소들 외에 다른 범용적인 구성요소들이 더 포함될 수 있음을 당해 기술분야의 통상의 기술자라면 이해할 수 있다.
- [0017] 컴퓨팅 시스템(1)은 데스크탑 컴퓨터, 노트북 컴퓨터, 스마트폰, PDA (personal digital assistants), 휴대형 미디어 플레이어, 비디오 게임용 콘솔, 텔레비전 셋탑 박스, 태블릿 디바이스, 이북 리더, 웨어러블 디바이스 등을 예로 들 수 있지만, 이에 제한되지 않는다. 즉, 이 밖에도 컴퓨팅 시스템(1)의 범주에는 다양한 장치들이 포함될 수 있다.
- [0018] GPU(10) 및 CPU(20)와 같은 프로세싱 장치들은 각종 연산들을 처리하는 하드웨어에 해당된다. CPU(20)는 컴퓨팅 시스템(1)의 전반적인 기능들을 제어하는 하드웨어로 동작할 수 있고, 다른 구성요소들인 GPU(10) 및 메모리(30)를 제어할 수 있다. CPU(20)는 컴퓨팅 시스템(1)의 기능들을 제어하기 위한 다양한 종류의 연산들(operations)을 수행할 수 있다. GPU(10)는 3차원 또는 2차원 그래픽스 데이터를 처리하기 위하여, 컴퓨팅 시스템(1)의 그래픽 프로세싱 기능을 제어하는 하드웨어로 동작할 수 있다. 즉, GPU(10)는 OpenGL (Open Graphic(s) Library), DirectX, CUDA(Compute Unified Device Architecture) 등의 다양한 종류들의 그래픽스 파이프라인(101)을 수행하면서, 그래픽스 파이프라인(101)과 관련된 연산들(예를 들어, 셰이딩(shading) 연산, 텍스처링(texturing) 연산 등)을 수행할 수 있다.
- [0019] 메모리(30)는, DRAM(dynamic random access memory), SRAM(static random access memory) 등과 같은 RAM(random access memory)에 해당되거나, 또는 ROM(read-only memory), EEPROM(electrically erasable programmable read-only memory) 등과 같은 디바이스에 해당될 수 있다. 즉, 메모리(30)는 GPU(10) 또는 CPU(20)에서 처리가 완료된 데이터를 저장하거나, 또는 GPU(10) 또는 CPU(20)에서 실행될 데이터를 제공하는 디바이스에 해당될 수 있다. 예를 들어, 메모리(30)는 텍스처링 관련 데이터들을 저장하거나 제공할 수 있다.

- [0020] Bus(40)는 컴퓨팅 장치(1) 내의 하드웨어들 간에 데이터를 송수신할 수 있도록 하드웨어들을 연결시켜 주는 하드웨어로서, Bus(40)는 예를 들어 PCI bus, PCI Express bus 등과 같은 다양한 종류들을 포함할 수 있다.
- [0021] 도 2는 일실시예에 따라, 그래픽 처리 장치(Graphic Processing Unit, GPU)에 의해 처리되는 그래픽스 파이프라인을 설명하는 도면이다.
- [0022] 도 2를 참고하면, 그래픽스 파이프라인(도 1의 101)의 전 과정은 크게 기하(geometry) 변환, 래스터화(rasterization), 픽셀 셰이딩(pixel shading)으로 구분될 수 있다.
- [0023] 201 단계는 버텍스들(vertices)을 생성하는 단계이다. 버텍스들은 3차원 그래픽스에 포함된 오브젝트(object)들을 나타내기 위해 생성된다.
- [0024] 202 단계는 생성된 버텍스들을 버텍스 셰이딩(vertex shading)하는 단계이다. 버텍스 셰이더는 201 단계에서 생성된 버텍스들의 위치들을 정의함으로써, 버텍스 셰이딩을 수행할 수 있다.
- [0025] 203 단계는 프리미티브들을 생성하는 단계이다. 프리미티브는 하나 이상의 버텍스들을 이용하여 형성되는 점, 선, 다각형(polygon)등을 의미한다. 예를 들어, 프리미티브는 3개의 버텍스들이 연결된 삼각형일 수 있다.
- [0026] 204 단계는 프리미티브를 래스터라이징(rasterizing)하는 단계이다. 프리미티브의 래스터화(rasterization)는 프리미티브를 프래그먼트들(fragments)로 분할하는 것을 의미한다. 프래그먼트는 프리미티브에 대한 그래픽스 처리를 수행하기 위한 기본 단위일 수 있다. 프리미티브는 버텍스에 대한 정보만을 포함하므로, 래스터라이징을 통해 버텍스와 버텍스 사이의 프래그먼트들이 생성됨으로써, 3차원 그래픽스 처리가 수행될 수 있다.
- [0027] 205 단계는 픽셀들을 셰이딩하는 단계이다. 래스터라이징에 의해 생성된, 프리미티브를 구성하는 프래그먼트들은 스크린 스페이스(screen space) 상의 픽셀들에 대응될 수 있다. 당해 분야에서, 프래그먼트와 픽셀이란 용어는 경우에 따라 혼용되어 사용되기도 한다. 예를 들어, 픽셀 셰이더는 프래그먼트 셰이더라고 호칭될 수도 있다. 일반적으로, 프리미티브를 구성하는 그래픽 처리의 기본 단위를 프래그먼트라고 지칭하고, 이후에 픽셀 셰이딩에 의해 수행될 그래픽 처리의 기본 단위를 픽셀이라 지칭할 수 있다. 픽셀 셰이딩에 의하여 픽셀들의 값들, 속성들 등(예를 들어 픽셀의 컬러)이 결정될 수 있다.
- [0028] 206 단계는 픽셀의 컬러를 결정하기 위한 텍스처링 단계이다. 텍스처링은 미리 준비된 텍스처들을 이용하여 픽셀들의 컬러들을 결정하는 과정을 의미할 수 있다. 다양한 컬러들과 패턴들이 픽셀들에서 표현되기 위해서는 모든 픽셀들의 컬러들이 각각 계산되고 결정되어야 한다. GPU(10)는 미리 준비된 텍스처들을 이용하여 픽셀들의 컬러들을 결정할 수 있다. 이때, 어느 픽셀들에 그려질 오브젝트의 크기에 적응적으로 대응할 수 있도록, 서로 다른 해상도들의 텍스처들이 미리 정의되어 준비될 수 있다. 서로 다른 해상도들을 갖는, 미리 정의된 텍스처들을 밍맵(mipmap)이라고 한다.
- [0029] 207 단계는 테스트 및 믹싱(testing and mixing) 단계이다. 깊이 테스트(depth test), 컬링(curling), 클리핑(clipping) 등을 통해 최종적으로 표시될 픽셀 값들이 결정된다.
- [0030] 208 단계는 201 단계 내지 207 단계를 통해 생성된 프레임들 프레임 버퍼에 저장하고, 프레임 버퍼에 저장된 프레임들을 디스플레이 장치를 통해 표시하는 단계이다.
- [0031] 도 2에서 설명된 위와 같은 그래픽스 파이프라인(101)의 일반적인 컨셉에 대한 설명은 개괄적인 것으로서, 보다 세부적인 과정들에 대해서는 당해 기술분야의 통상의 기술자에게 자명하다.
- [0032] 도 3은 일실시예에 따른 도 1의 GPU의 상세 하드웨어 구조를 도시한 블록도이다.
- [0033] 도 3을 참고하면, GPU(10)는 그래픽스 파이프라인(100)을 수행하는 입력 어셈블러(input assembler)(110), 버텍스 셰이더(vertex shader)(120), 프리미티브 어셈블러(primitive assembler)(130), 비너(binner)(140), 래스터라이저(rasterizer)(150) 및 프래그먼트 셰이더(fragment shader)(160)를 포함할 수 있다. 또한, GPU(10)는 프로세서(170) 및 메모리(180)를 더 포함할 수 있다. 한편, GPU(10) 내에서 그래픽스 파이프라인(100)을 수행하는 위의 구성들은, 아래 설명될 기능들에 기초하여 분류된 것일 수 있다. 따라서, 그래픽스 파이프라인(100)을 수행하는 위의 구성들 각각은, 아래 설명될 기능들 각각을 실행시키는 프로그램 로직들 또는 소프트웨어 모듈들로 구현될 수 있다. 이와 다르게, 그래픽스 파이프라인(100)을 수행하는 위의 구성들은, GPU(10) 내에 구비된 서브(sub) 프로세싱 유닛들(또는 프로세서 코어들) 각각으로 구현될 수도 있다. 즉, 그래픽스 파이프라인(100)을 수행하는 위의 구성들의 구현 형태는, 어느 하나에 의해 제한되지 않는다. 나아가서, 그래픽스 파이프라인(100)을 수행하는 위의 구성들의 명칭들은 아래 설명될 기능들에 기초하여 부여된 것일 수 있으나, 이와 같

은 명칭들은 다양하게 변경될 수도 있음을 당해 기술분야의 통상의 기술자라면 이해할 수 있다.

- [0034] 한편, GPU(10)에서 그래픽스 파이프라인(100)을 수행하는 구성들의 명칭들은 설명의 편의를 위하여, OpenGL(Open Graphics Library)에서 정의된 명칭들을 사용하였으나, 이 구성들의 명칭들은 이에 제한되지 않는다. 즉, GPU(10)에서 그래픽스 파이프라인(100)을 수행하는 구성들은, Microsoft 사의 DirectX 11 (DX11), CUDA(Compute Unified Device Architecture) 6.0 등의 다른 종류의 API들에서 정의된 유사한 구성들에 대응될 수 있다.
- [0035] 입력 어셈블러(110)는, 입력된 드로우콜들(drawcalls)에 기초하여, 메모리(도 1의 30)에 저장된 객체들에 관한 버텍스들의 데이터를 그래픽스 파이프라인(100)에 공급한다. 그래픽스 파이프라인(100)에서 공급된 버텍스들은, 메쉬(mesh) 또는 표면(surface)의 표현인 패치에 관한 것일 수 있으나, 이에 제한되지 않는다. 드로우콜은 어떤 프레임에 어떤 오브젝트를 렌더링할 것인지를 나타내는 커맨드이다. 예를 들어, 드로우콜은 영상 또는 프레임에 사각형들 또는 삼각형들과 같은 프리미티브들을 그리기 위한 커맨드일 수 있다.
- [0036] 버텍스 셰이더(120)는 입력 어셈블러(110)에 의해 공급된 버텍스들을 컨트롤 셰이더(121)의 입력 컨트롤 포인트들로서 전달한다.
- [0037] 프리미티브 어셈블러(130)는 테셀레이션 파이프라인(300)이 수행된 경우, 테셀레이팅된 버텍스들을 프리미티브들(즉, 테셀레이팅된 프리미티브들(tessellated primitives))로 변환한다. 프리미티브 어셈블러(130)는 테셀레이션 파이프라인(300)이 수행되지 않은 경우에는, 버텍스 셰이더(120)로부터 출력된 버텍스들을 프리미티브들로 변환한다.
- [0038] 비너(140)는 프리미티브 어셈블러(130)로부터 출력된 프리미티브들(또는 테셀레이팅된 프리미티브들)을 이용하여, 비닝 또는 타일링을 수행한다. 즉, 비너(140)는 깊이 테스트(또는 tile Z test)를 수행하여 출력 프리미티브들 각각이 속한 타일들의 정보를 나타내는 타일 리스트를 생성(비닝)한다.
- [0039] 래스터라이저(150)는 생성된 타일 리스트에 기초하여, 출력 프리미티브들을 2차원 공간에서의 픽셀 값들로 변환한다. 프래그먼트는, 프리미티브에 의해 커버되는 픽셀들을 의미하고, 프래그먼트 셰이더(160)는, 이와 같은 프래그먼트를 생성하고, 프래그먼트에 대한 깊이 값, 스텐실 값, 컬러 값 등을 결정한다. 프래그먼트 셰이더(160)의 셰이딩 결과는 버퍼들(180)(예를 들어, 프레임 버퍼)에 저장된 후, 동영상의 프레임으로서 디스플레이될 수 있다.
- [0040] 프로세서(170)는 그래픽스 파이프라인(100)의 각 구성요소들(110 내지 160)과 메모리(180)의 전반적인 기능들 및 동작들을 제어한다.
- [0041] 도 4는 일실시예에 따라, 그래픽 처리 장치의 동작 방법을 설명하기 위한 개념도이다.
- [0042] 그래픽 처리 장치(10)는 복수의 컴포넌트들 각각에 대한 오퍼레이션 데이터(411, 412)를 복수의 상태마다 메모리(410)에 저장할 수 있다. 여기서, 메모리는 그래픽 처리 장치(10) 내부의 메모리 또는 외부의 메모리에 해당될 수 있다. 또한, 복수의 컴포넌트들 각각은 복수의 서브 컴포넌트들을 포함할 수 있다. 복수의 컴포넌트들 각각에 대한 서브 컴포넌트들의 오퍼레이션 데이터도 복수의 상태마다 메모리(410)에 저장될 수 있다. 메모리는 내부의 메모리 또는 외부의 메모리에 해당될 수 있다.
- [0043] 그래픽 처리 장치(10)는 복수의 컴포넌트들의 오퍼레이션 데이터(411, 412) 및 명령 스트림(420)을 이용하여 그래픽스 파이프라인을 수행할 수 있다. 그래픽 처리 장치 내의 복수의 컴포넌트들은 명령 스트림(420)에 기초하여, 그래픽스 파이프라인을 수행할 수 있다. 그래픽 처리 장치(10)는 복수의 컴포넌트들의 오퍼레이션 데이터(411, 412) 중에서 명령 스트림(420)에 따라 그래픽스 파이프라인을 수행하는 데에 사용되는 제1 오퍼레이션 데이터(413)를 획득할 수 있다. 그래픽 처리 장치(10)는 제1 오퍼레이션 데이터(413) 및 명령 스트림(420)에 기초하여, 그래픽스 파이프라인을 수행하는 데에 사용되는 제2 오퍼레이션 데이터(431)를 생성할 수 있다. 그래픽 처리 장치(10) 내의 복수의 컴포넌트들은 제2 오퍼레이션 데이터(431)에 따라 그래픽스 파이프라인을 수행할 수 있다.
- [0044] 도 5a 및 도 5b는 일실시예에 따라, 그래픽 처리 장치의 제1 컴포넌트가 소정 저장 공간에 저장된 오퍼레이션 데이터를 이용하여 그래픽스 파이프라인을 수행하는 과정을 설명하기 위한 도면이다.
- [0045] 그래픽 처리 장치의 제1 컴포넌트는 제1 서브 컴포넌트, 제2 서브 컴포넌트, 제3 서브 컴포넌트, 제4 서브 컴포넌트로 구성될 수 있다. 제1 컴포넌트는 드로우 명령에 따라 그래픽스 파이프라인을 수행할 수 있다.

- [0046] 드로우 명령에 대한 제1 컴포넌트의 오퍼레이션 코드를 포함하는 오퍼레이션 데이터는 소정의 할당된 공간에 저장될 수 있다. 소정의 할당된 공간은 그래픽 처리 장치의 내부 메모리 또는 외부 메모리 일 수 있다. 또한, 소정의 할당된 공간은 컴포넌트의 내부 공간일 수도 있다.
- [0047] 그래픽 처리 장치의 제1 컴포넌트는 복수의 드로우 명령들에 따라 제1 오퍼레이션 데이터, 제2 오퍼레이션 데이터, 제3 오퍼레이션 데이터, 제4 오퍼레이션 데이터, 제1 오퍼레이션 데이터, 제2 오퍼레이션 데이터, 제3 오퍼레이션 데이터, 제4 오퍼레이션 데이터의 순서로 오퍼레이션 데이터를 이용하여 그래픽스 파이프라인을 수행한다고 가정한다.
- [0048] 도 5a에 도시된 바와 같이, 그래픽스 파이프라인을 수행하는 데에 사용되는 오퍼레이션 데이터가 저장되는 소정의 할당된 공간이 4개의 상태 버전 데이터만을 저장할 수 있다고 가정한다. 소정의 할당된 공간이 4개의 상태 버전 데이터를 저장할 수 있으므로, 제1 오퍼레이션 데이터 내지 제4 오퍼레이션 데이터가 순차적으로 제1 공간 내지 제4 공간(501-504)에 저장되고, 이후에 제1 공간 내지 제 4 공간(501-504)에 저장되었던 오퍼레이션 데이터가 순차적으로 사용될 수 있다. 드로우 명령에 따라 필요한 오퍼레이션 데이터가 저장될 공간이 충분한 경우, 그래픽 처리 장치는 별도의 트래픽 없이 바로 드로우 명령에 따른 그래픽스 파이프라인을 수행할 수 있다.
- [0049] 도 5b에 도시된 바와 같이, 그래픽스 파이프라인을 수행하는 데에 사용되는 오퍼레이션 데이터가 저장되는 소정의 할당된 공간이 3개의 상태 버전 데이터만을 저장할 수 있다고 가정한다. 소정의 할당된 공간이 3개의 상태 버전 데이터만을 저장할 수 있으므로, 제1 오퍼레이션 데이터 내지 제3 오퍼레이션 데이터가 순차적으로 제1 공간 내지 제3 공간(511-513)에 저장되고, 이후에 제1 오퍼레이션 데이터가 저장되었던 제1 공간(511)에는 제4 오퍼레이션 데이터가 저장될 수 있다. 네번째 드로우 명령부터는 공간에 저장되었던 오퍼레이션 데이터가 새로운 오퍼레이션 데이터로 업데이트될 수 있다. 즉 순차적으로, 제 1 오퍼레이션 데이터는 제2 공간(512)에 저장되고, 제2 오퍼레이션 데이터는 제3 공간(513)에 저장되고, 제3 오퍼레이션 데이터는 제1 공간(511)에 저장되고, 제4 오퍼레이션 데이터는 제2 공간(512)에 저장될 수 있다. 드로우 명령에 따라 필요한 오퍼레이션 데이터가 저장될 공간이 충분하지 않은 경우, 그래픽 처리 장치는 트래픽을 최소화하기 위해 컴포넌트의 복수의 상태들 중 빈도가 높은 상태에 기초하여 생성된 베이스 오퍼레이션 데이터를 이용하여 그래픽스 파이프라인을 수행할 수 있다. 이와 관련된 그래픽 처리 장치의 동작은 도 6 내지 도 13에서 상세하게 설명한다.
- [0050] 도 6은 도 1에 도시된 그래픽 처리 장치의 구성을 나타낸 블록도이다.
- [0051] 도 6을 참고하면, 그래픽 처리 장치(600)는 복수의 컴포넌트들(610), 프로세서(620) 및 메모리(630)를 포함할 수 있다. 그러나, 도시된 구성 모두가 필수 구성 요소인 것은 아니다. 도시된 구성 요소보다 많은 구성 요소에 의해 그래픽 처리 장치(600)가 구현될 수 있고, 그보다 적은 구성 요소에 의해서도 그래픽 처리 장치(600)가 구현될 수 있다. 이하, 상기 구성 요소들에 대해 차례로 살펴본다. 도 6의 그래픽 처리 장치(600)는 도 1의 그래픽 처리 장치(10)와 대응될 수 있다.
- [0052] 그래픽 처리 장치(600)에서 복수의 컴포넌트들(610)은 그래픽스 파이프라인을 수행할 수 있다. 여기서, 복수의 컴포넌트들(610)은 도 3에 도시된 입력 어셈블러(110), 버텍스 셰이더(120), 프리미티브 어셈블러(130), 비너(140), 래스터라이저(150) 및 프래그먼트 셰이더(160)와 같은 그래픽스 파이프라인을 수행하는 컴포넌트들을 포함할 수 있고, 이에 한정되지 않는다. 복수의 컴포넌트들(610)에 상기 컴포넌트들 이외에 그래픽스 파이프라인을 수행하는 다른 컴포넌트들도 포함될 수 있음은 본 개시에 속하는 기술분야에서 통상의 지식을 가진 자라면 이해할 수 있다.
- [0053] 프로세서(620)는 메모리(630)에 저장된 컴퓨터 실행가능 명령어를 실행할 수 있다. 프로세서(620)는, 복수의 컴포넌트들(610)의 베이스 상태를 나타내는 베이스 오퍼레이션 데이터에 기초하여, 그래픽스 파이프라인을 수행하기 위한 오퍼레이션 데이터를 생성할 수 있다. 여기서, 베이스 상태는 복수의 컴포넌트들(610) 각각에 대한 복수의 상태를 고려하여 결정된 상태를 의미할 수 있다. 예를 들면, 복수의 컴포넌트들(610) 각각에 대한 복수의 상태 중 가장 빈번하게 사용되는 상태가 베이스 상태로 설정될 수 있다. 베이스 오퍼레이션 데이터는 복수의 컴포넌트들(610) 각각의 베이스 상태를 나타내는 정보를 포함할 수 있다. 예를 들면, 베이스 오퍼레이션 데이터는 복수의 컴포넌트들(610) 각각의 베이스 상태에서 사용되는 오퍼레이션 코드를 포함할 수 있다.
- [0054] 프로세서(620)는 그래픽스 파이프라인이 수행될 때 가장 빈도가 높은 복수의 컴포넌트들(610)의 상태에 기초하여, 베이스 오퍼레이션 데이터를 생성할 수 있다.
- [0055] 프로세서(620)는, 복수의 컴포넌트들(610)의 제1 상태에서 그래픽스 파이프라인을 수행하는 데에 사용되는 제1

상태 버전 코드를 포함하는 제1 오퍼레이션 데이터를 획득할 수 있다. 복수의 컴포넌트들(610)은 제1 컴포넌트, 제2 컴포넌트 및 제3 컴포넌트로 구성될 수 있다. 제1 상태 버전 코드는 제1 상태에서의 제1 컴포넌트의 상태 정보, 제2 컴포넌트의 상태 정보 및 제3 컴포넌트의 상태 정보를 나타낼 수 있다. 제1 상태 버전 코드를 포함하는 제1 오퍼레이션 데이터는 그래픽 처리 장치(600)의 내부 메모리 또는 외부 메모리로부터 획득될 수 있다.

- [0056] 프로세서(620)는, 베이스 오퍼레이션 데이터에 기초하여, 제1 오퍼레이션 데이터를 제2 오퍼레이션 데이터로 업데이트할 수 있다. 여기서, 제2 오퍼레이션 데이터는 복수의 컴포넌트들(610)의 제2 상태에서 사용되는 제2 상태 버전 코드를 포함할 수 있다. 프로세서(620)는 베이스 오퍼레이션 데이터 중에서 제2 상태에 따른 그래픽스 파이프라인에 사용될 적어도 하나의 컴포넌트의 베이스 오퍼레이션 데이터를 획득하고, 제1 오퍼레이션 데이터 중 적어도 하나의 컴포넌트에 대응되는 데이터를 적어도 하나의 컴포넌트의 베이스 오퍼레이션 데이터로 업데이트 할 수 있다. 이와 관련된 구체적인 내용은 도 9에서 설명한다.
- [0057] 또한, 복수의 컴포넌트들(610) 각각은 복수의 서브 컴포넌트들을 포함할 수 있다. 예를 들면, 그래픽 처리 장치(600)는 제1 컴포넌트, 제2 컴포넌트를 포함할 수 있다. 제1 컴포넌트는 제1 서브 컴포넌트, 제2 서브 컴포넌트 및 제3 서브 컴포넌트를 포함할 수 있고, 제2 컴포넌트는 제4 서브 컴포넌트 및 제5 서브 컴포넌트를 포함할 수 있다. 여기서, 복수의 컴포넌트들(610) 각각은 복수의 컴포넌트들(610) 각각의 상태에서 사용되는 코드를 포함하는 서브 오퍼레이션 데이터를 획득할 수 있다. 복수의 서브 컴포넌트들 각각은 서브 오퍼레이션 데이터에 따라 그래픽스 파이프라인을 수행할 수 있다.
- [0058] 프로세서(620)는 베이스 오퍼레이션 데이터를 메모리(630) 또는 복수의 컴포넌트들(610)의 소정의 공간에 저장 되도록 제어할 수 있다.
- [0059] 프로세서(620)는 복수의 컴포넌트들(610) 각각의 베이스 오퍼레이션 데이터 중 제2 상태를 결정하는 데에 사용되는 적어도 하나의 컴포넌트의 베이스 오퍼레이션 데이터를 획득할 수 있다. 프로세서(620)는 적어도 하나의 컴포넌트의 베이스 오퍼레이션 데이터에 기초하여, 제1 오퍼레이션 데이터를 제2 오퍼레이션 데이터로 업데이트 할 수 있다.
- [0060] 프로세서(620)는 복수의 컴포넌트들(610) 중에서, 그래픽스 파이프라인이 수행될 때 컴포넌트에 대한 상태의 변경 빈도가 가장 높은 제1 컴포넌트를 결정할 수 있다. 프로세서(620)는 제2 상태에서 사용되는 제1 컴포넌트의 오퍼레이션 코드에 기초하여, 베이스 오퍼레이션 데이터를 설정할 수 있다. 구체적으로, 제2 상태에서 사용되는 제1 컴포넌트의 오퍼레이션 코드가 이전의 드로우 명령에 따라 사용된 제1 컴포넌트의 오퍼레이션 코드와 동일하면, 프로세서(620)는 베이스 오퍼레이션 데이터를 그대로 사용할 수 있다. 제2 상태에서 사용되는 제1 컴포넌트의 오퍼레이션 코드가 이전의 드로우 명령에 따라 사용된 제1 컴포넌트의 오퍼레이션 코드와 동일하지 않으면, 프로세서(620)는 이전에 사용된 제1 컴포넌트의 오퍼레이션 데이터를 제2 상태에서 사용되는 제1 컴포넌트의 오퍼레이션 데이터로 업데이트할 수 있다.
- [0061] 또한, 제1 컴포넌트는 복수의 서브 컴포넌트들을 포함할 수 있다. 프로세서(620)는 복수의 서브 컴포넌트들 중에서, 그래픽스 파이프라인이 수행될 때 서브 컴포넌트에 대한 상태의 변경 빈도가 가장 높은 제1 서브 컴포넌트를 결정하고, 제2 상태에서 사용되는 제1 서브 컴포넌트의 오퍼레이션 코드에 기초하여, 베이스 오퍼레이션 데이터를 설정할 수 있다. 구체적으로, 제2 상태에서 사용되는 제1 서브 컴포넌트의 오퍼레이션 코드와 이전의 드로우 명령에 따라 사용된 제1 서브 컴포넌트의 오퍼레이션 코드가 동일한지 여부에 따라, 프로세서(620)는 베이스 오퍼레이션 데이터를 그대로 사용하거나 업데이트할 수 있다.
- [0062] 프로세서(620)는 복수의 컴포넌트들(610)의 복수의 상태 각각에서 사용되는 복수의 상태 버전 데이터의 사용 빈도에 기초하여, 복수의 상태 버전 데이터의 우선 순위를 결정할 수 있다. 여기서, 상태 버전 데이터는 그래픽스 파이프라인이 수행될 때 복수의 컴포넌트들(610) 각각에 대한 상태 값으로 구성된 데이터이다. 예를 들면, 제1 컴포넌트의 제1 상태에 대한 상태 값과 제2 컴포넌트의 제2 상태에 대한 상태 값은 제1 상태 버전 데이터로 설정될 수 있다. 복수의 상태 버전 데이터는 복수의 컴포넌트들(610)의 복수의 상태 각각에서 그래픽스 파이프라인을 수행하는 데에 사용되는 코드를 포함할 수 있다.
- [0063] 프로세서(620)는 복수의 상태 버전 데이터 중 우선 순위가 가장 높은 제1 상태 버전 데이터를 보존하고, 제1 상태 버전 데이터보다 우선 순위가 낮은 제2 상태 버전 데이터를 그래픽스 파이프라인이 수행됨에 따라 업데이트 되도록 제어할 수 있다.
- [0064] 프로세서(620)는 그래픽스 파이프라인에 사용될 상태 버전 데이터를 결정하고, 결정된 데이터에 따라 그래픽스

파이프라인을 수행할 수 있다.

- [0065] 메모리(630)는 컴퓨터 실행가능 명령어를 저장할 수 있다. 메모리(630)는 베이스 오퍼레이션 데이터 및 오퍼레이션 데이터를 저장할 수 있다. 메모리(630)의 할당된 공간에 저장된 제1 오퍼레이션 데이터, 제2 오퍼레이션 데이터, ...제N 오퍼레이션 데이터 등 전체 오퍼레이션 데이터는 그래픽스 파이프라인을 수행하는 데에 이용될 수 있다. 또한, 메모리(630)는 복수의 상태 버전 데이터를 저장할 수 있다.
- [0066] 그래픽 처리 장치(600)는 중앙 연산 프로세서를 구비하여, 복수의 컴포넌트들(610), 프로세서(620) 및 메모리(630)의 동작을 총괄적으로 제어할 수 있다. 중앙 연산 프로세서(620)는 다수의 논리 게이트들의 어레이로 구현될 수도 있고, 범용적인 마이크로 프로세서(620)와 이 마이크로 프로세서에서 실행될 수 있는 프로그램이 저장된 메모리(630)와의 조합으로 구현될 수도 있다. 또한, 다른 형태의 하드웨어로 구현될 수 있음은 본 개시에 속하는 기술분야에서 통상의 지식을 가진 자라면 이해할 수 있다.
- [0067] 이하에서는, 그래픽 처리 장치(600)가 수행하는 다양한 동작이나 응용들이 설명되는데, 상기 복수의 컴포넌트들(610), 프로세서(620) 및 메모리(630) 중 어느 구성을 특정하지 않더라도 본 개시에 속하는 기술분야에서 통상의 지식을 가진 자가 명확하게 이해하고 예상할 수 있는 정도의 내용은 통상의 구현으로 이해될 수 있으며, 본 발명의 권리범위가 특정한 구성의 명칭이나 물리적/논리적 구조에 의해 제한되는 것은 아니다.
- [0068] 도 7은 그래픽 처리 장치에서 그래픽스 파이프라인을 수행하는 복수의 컴포넌트들을 설명하기 위한 도면이다.
- [0069] 그래픽 처리 장치 내의 복수의 컴포넌트들은 명령 스트림에 따라, 그래픽스 파이프라인을 수행할 수 있다. 복수의 컴포넌트들 각각은 서로 다른 기능을 수행할 수 있다. 복수의 컴포넌트들은 도 3에서 설명한 입력 어셈블러(110), 버텍스 셰이더(120), 프리미티브 어셈블러(130), 비너(140), 래스터라이저(150) 및 프래그먼트 셰이더(160)와 같은 그래픽스 파이프라인을 수행하는 컴포넌트들에 해당될 수 있다.
- [0070] 복수의 컴포넌트들 각각은 적어도 하나의 서브 컴포넌트들로 구성될 수 있다. 도 7에 도시된 바와 같이, 제1 컴포넌트(710)는 제1 서브 컴포넌트(711), 제2 서브 컴포넌트(712), 제3 서브 컴포넌트(713) 및 제 4 서브 컴포넌트(715)로 구성될 수 있다. 제1 컴포넌트(710)의 상태에 대한 오퍼레이션 데이터는 그래픽 처리 장치 내부의 메모리(715) 또는 제1 컴포넌트의 소정의 공간에 저장될 수 있다. 오퍼레이션 데이터는 제1 컴포넌트(710)의 상태에 대한 오퍼레이션 코드를 포함할 수 있고, 오퍼레이션 코드를 나타내는 벡터 또는 값을 포함할 수 있다. 오퍼레이션 데이터는 상기 언급된 구성 이외에 다른 구성들을 포함하거나 다른 구성들로 나타낼 수 있음은 본 개시에 속하는 기술분야에서 통상의 지식을 가진 자라면 이해할 수 있다.
- [0071] 또한, 오퍼레이션 데이터는 복수의 서브 컴포넌트들 각각에 대한 오퍼레이션 데이터일 수 있다. 제1 상태 버전 데이터는 제1 컴포넌트의 제1 상태에서의 복수의 서브 컴포넌트들 각각의 오퍼레이션 데이터 전체를 의미할 수 있다. 또한, 제1 상태 버전 코드는 제1 컴포넌트의 제1 상태에서의 복수의 서브 컴포넌트들 각각의 오퍼레이션 코드 전체를 의미할 수 있다.
- [0072] 제1 컴포넌트(710)의 제1 상태에서의 제1 상태 버전 데이터(716), 제2 상태에서의 제2 상태 버전 데이터(717), 제3 상태에서의 제3 상태 버전 데이터(718) 및 제4 상태에서의 제4 상태 버전 데이터(719)는 그래픽 처리 장치 내부의 메모리(715) 또는 제1 컴포넌트의 소정의 공간에 저장될 수 있다.
- [0073] 또한, 도 7에 도시된 바와 같이, 제2 컴포넌트(720)도 제1 서브 컴포넌트(721), 제2 서브 컴포넌트(722), 제3 서브 컴포넌트(723) 및 제 4 서브 컴포넌트(724)로 구성될 수 있다. 제2 컴포넌트(720)의 복수의 상태 각각에서의 상태 버전 데이터(726-729)는 그래픽 처리 장치 내부의 메모리(725) 또는 제2 컴포넌트의 소정의 공간에 저장될 수 있다.
- [0074] 여기서, 제1 컴포넌트와 제2 컴포넌트의 서브 컴포넌트들의 개수가 같도록 도시하였지만, 서로 다른 개수의 서브 컴포넌트들로 구성될 수 있고, 서로 다른 크기의 공간에 저장될 수 있다.
- [0075] 도 8은 일실시예에 따라, 그래픽 처리 장치의 동작 방법을 설명하기 위한 도면이다.
- [0076] 도 8은 그래픽 처리 장치의 제1 컴포넌트가 명령 스트림에 따라 그래픽스 파이프라인을 수행하는 과정을 설명하기 위한 도면이다.
- [0077] 명령 스트림(810)은 베이스 오퍼레이션 데이터를 설정하는 정보, 베이스 오퍼레이션 벡터 정보, 그래픽스 파이프라인을 수행함에 있어 사용될 상태 버전 데이터 정보 및 드로우 명령 데이터에 대한 정보 등을 포함할 수 있고, 이에 한정되지 않는다.

- [0078] 베이스 오퍼레이션 데이터는 복수의 컴포넌트들의 베이스 상태를 나타내는 정보를 포함할 수 있다. 예를 들어, 그래픽 처리 장치는 그래픽스 파이프라인이 수행될 때 컴포넌트의 상태 중 가장 빈도가 높은 상태를 베이스 상태로 결정하고, 가장 빈도가 높은 상태를 나타내는 코드를 포함하는 베이스 오퍼레이션 데이터를 생성할 수 있다.
- [0079] 베이스 오퍼레이션 벡터는 베이스 오퍼레이션 데이터 중 그래픽스 파이프라인을 수행하는 데에 사용되는 데이터를 선택하는 정보를 나타낼 수 있다.
- [0080] 상태 버전 데이터는 그래픽스 파이프라인이 수행될 때 복수의 컴포넌트들 각각에 대한 상태 값을 나타내는 정보를 포함할 수 있다. 도 8에 도시된 바와 같이, 제1 컴포넌트에 대한 제1 상태 버전 데이터는 복수의 서브 컴포넌트들 각각에 대한 상태 값을 나타내는 정보를 포함할 수 있다. 구체적으로, 제1 컴포넌트에 대한 제1 상태 버전 데이터는 제1 상태에서의 제1 서브 컴포넌트 내지 제4 서브 컴포넌트 각각에 대한 상태 값을 나타내는 정보를 포함할 수 있다. 또한, 제1 컴포넌트에 대한 제2 상태 버전 데이터는 제2 상태에서의 제1 서브 컴포넌트 내지 제4 서브 컴포넌트 각각에 대한 상태 값을 나타내는 정보를 포함할 수 있다.
- [0081] 드로우 명령 데이터는 각 드로우 명령에 대한 정보를 포함한다.
- [0082] 도 8에 도시된 바와 같이, 그래픽 처리 장치는 복수의 서브 컴포넌트들의 베이스 상태를 나타내는 베이스 오퍼레이션 데이터(820)를 획득할 수 있다. 베이스 오퍼레이션 데이터(820)는 그래픽스 파이프라인이 수행될 때 가장 빈도가 높은 서브 컴포넌트의 상태의 상태 값에 기초하여 생성될 수 있다. 예를 들면, 그래픽 처리 장치는 제1 서브 컴포넌트의 베이스 상태 값을 0x34로 설정하고, 제2 서브 컴포넌트의 베이스 상태 값을 0x1로 설정할 수 있다. 제3 서브 컴포넌트의 베이스 상태 값 및 제4 서브 컴포넌트의 베이스 상태 값은 기존에 설정된 베이스 상태 값이 그대로 사용될 수 있다.
- [0083] 도 8에 도시된 바와 같이, 그래픽 처리 장치는 제1 컴포넌트의 제1 상태에서 복수의 서브 컴포넌트들에 대한 제1 상태 버전 코드를 포함하는 제1 서브 오퍼레이션 데이터(840)를 획득할 수 있다. 그래픽 처리 장치는 제1 컴포넌트의 제2 상태에서 복수의 서브 컴포넌트들에 대한 제2 상태 버전 코드를 포함하는 제2 서브 오퍼레이션 데이터를 획득할 수 있다.
- [0084] 도 8에 도시된 바와 같이, 그래픽 처리 장치는 베이스 오퍼레이션 데이터 중 제1 컴포넌트의 상태를 결정하는 데에 사용되는 적어도 하나의 서브 컴포넌트의 베이스 오퍼레이션 데이터를 획득할 수 있다. 그래픽 처리 장치는 베이스 오퍼레이션 벡터(830)를 이용하여, 제1 서브 컴포넌트의 베이스 상태 값에 대한 정보를 획득할 수 있다. 베이스 오퍼레이션 벡터(830)가 (1,0,0,0)이면, 그래픽 처리 장치는 제1 서브 컴포넌트 내지 제4 서브 컴포넌트 중 제1 서브 컴포넌트의 베이스 상태 값에 대한 정보를 획득할 수 있다.
- [0085] 도 8를 참고하면, 그래픽 처리 장치는 제1 서브 컴포넌트의 베이스 상태 값을 제1 서브 컴포넌트의 상태 값으로 업데이트하고, 제2 서브 오퍼레이션 데이터에 포함된 제2 서브 컴포넌트 내지 제4 서브 컴포넌트의 상태 값을 그대로 사용하여 제1 컴포넌트의 서브 오퍼레이션 데이터를 획득할 수 있다. 그래픽 처리 장치는 획득된 제1 컴포넌트의 서브 오퍼레이션 데이터에 따라 그래픽스 파이프라인을 수행할 수 있다.
- [0086] 제1 컴포넌트의 제1 서브 컴포넌트의 상태만 계속하여 변경되는 경우, 그래픽 처리 장치는 기존의 제2 서브 컴포넌트 내지 제4 서브 컴포넌트의 상태 값을 그대로 사용하고, 제1 서브 컴포넌트의 상태 값만을 업데이트하여, 그래픽스 파이프라인을 수행할 수 있다. 결과적으로, 그래픽 처리 장치는 불필요하게 상태 버전 데이터 전부를 업데이트 하는 동작을 방지할 수 있고, 그래픽스 파이프라인의 수행 속도를 높일 수 있다.
- [0087] 도 8은 제1 컴포넌트의 그래픽스 파이프라인 수행 과정을 설명하였는데, 그래픽 처리 장치를 구성하는 다른 컴포넌트들도 마찬가지로 방법으로 그래픽스 파이프라인을 수행할 수 있다.
- [0088] 도 9는 일실시예에 따라, 복수의 컴포넌트들 각각에 대한 상태 데이터의 사용 빈도에 따라 베이스 오퍼레이션 데이터를 설정하는 과정을 설명하기 위한 도면이다.
- [0089] 그래픽 처리 장치는 하나의 이미지를 구성하는 복수의 드로우를 분석하여 가장 빈번하게 사용된 컴포넌트의 상태 값을 베이스 오퍼레이션 데이터로 설정할 수 있다.
- [0090] 도 9에 도시된 바와 같이, 그래픽 처리 장치는 제1 이미지를 구성하는 복수의 드로우를 분석하여, 가장 빈번하게 사용된 제1 컴포넌트의 상태 값을 A로 결정할 수 있다. 마찬가지로, 그래픽 처리 장치는 제2 컴포넌트의 상태 값을 B로, 제3 컴포넌트의 상태 값을 D로 설정하여 베이스 오퍼레이션 데이터를 획득할 수 있다.

- [0091] 마찬가지로, 그래픽 처리 장치는 제2 이미지를 구성하는 복수의 드로우를 분석하여, 각 컴포넌트에서 빈번하게 사용된 상태 값을 기초하여 베이스 오퍼레이션 데이터를 획득할 수 있다. 그래픽 처리 장치는 제1 컴포넌트의 상태 값을 B로, 제2 컴포넌트의 상태 값을 B로, 제3 컴포넌트의 상태 값을 C로 설정하여 베이스 오퍼레이션 데이터를 획득할 수 있다.
- [0092] 도 10은 일실시예에 따라, 복수의 컴포넌트들 각각에 대한 상태 버전 데이터의 사용 빈도에 따라, 그래픽 처리 장치의 상태 버전 데이터를 처리하는 방법을 설명하기 위한 도면이다.
- [0093] 그래픽 처리 장치는 복수의 컴포넌트들의 복수의 상태 각각에 대한 복수의 상태 버전 데이터의 사용 빈도에 기초하여, 그래픽스 파이프라인을 수행할 수 있다.
- [0094] 도 10을 참고하면, 그래픽 처리 장치가 그래픽스 파이프라인을 수행함에 있어서, 제1 상태 버전 데이터(1010) 및 제2 상태 버전 데이터(1020)의 사용 빈도는 높고, 제3 상태 버전 데이터(1030) 및 제4 상태 버전 데이터(1040)의 사용 빈도는 낮다고 가정한다. 그래픽 처리 장치는 제1 상태 버전 데이터(1010) 및 제2 상태 버전 데이터(1020)를 보존하고, 제3 상태 버전 데이터(1030) 및 제4 상태 버전 데이터(1040)를 그래픽스 파이프라인을 수행하는 데 사용되는 다른 상태 버전 데이터로 업데이트할 수 있다.
- [0095] 그래픽 처리 장치는 사용 빈도가 낮은 상태 버전 데이터만을 새로운 상태 버전 데이터로 업데이트 함으로써, 사용 빈도가 높은 상태 버전 데이터를 재사용하여 그래픽스 파이프라인을 수행할 수 있다.
- [0096] 도 11은 일실시예에 따라, 그래픽 처리 장치의 동작 방법의 흐름을 나타낸 순서도이다.
- [0097] 도 11을 참고하면, 단계 S1110에서, 그래픽 처리 장치는 복수의 컴포넌트들의 상태의 변경 빈도에 기초하여, 복수의 컴포넌트들의 베이스 상태를 나타내는 베이스 오퍼레이션 데이터를 생성할 수 있다.
- [0098] 그래픽 처리 장치는 그래픽스 파이프라인이 수행될 때 가장 빈도가 높은 복수의 컴포넌트들의 상태에 기초하여, 베이스 오퍼레이션 데이터를 생성할 수 있다.
- [0099] 단계 S1120에서, 그래픽 처리 장치는 베이스 오퍼레이션 데이터에 기초하여, 그래픽스 파이프라인을 수행하는 데 사용되는 오퍼레이션 데이터를 생성할 수 있다.
- [0100] 복수의 컴포넌트들 각각은 복수의 서브 컴포넌트들을 포함할 수 있다. 그래픽 처리 장치는 복수의 컴포넌트들 각각이 그래픽스 파이프라인을 수행하는 데 사용되는 서브 오퍼레이션 데이터를 획득하도록 제어할 수 있다.
- [0101] 그래픽 처리 장치는 베이스 오퍼레이션 데이터를 메모리 또는 복수의 컴포넌트들의 소정의 공간에 저장할 수 있다.
- [0102] 그래픽 처리 장치는 복수의 컴포넌트들 각각의 베이스 오퍼레이션 데이터 중 제2 상태를 결정하는 데 사용되는 적어도 하나의 컴포넌트의 베이스 오퍼레이션 데이터를 획득할 수 있다.
- [0103] 단계 S1130에서, 그래픽 처리 장치는 생성된 오퍼레이션 데이터에 따라 그래픽스 파이프라인을 수행할 수 있다. 또한, 그래픽 처리 장치는 복수의 컴포넌트 각각이 서브 오퍼레이션 데이터에 따라 그래픽스 파이프라인을 수행하도록 제어할 수 있다.
- [0104] 그래픽 처리 장치는 적어도 하나의 컴포넌트의 베이스 오퍼레이션 데이터에 기초하여 제1 오퍼레이션 데이터를 제2 오퍼레이션 데이터로 업데이트할 수 있다. 그래픽 처리 장치는 업데이트된 제2 오퍼레이션 데이터를 저장할 수 있다. 그래픽 처리 장치는 업데이트된 제2 오퍼레이션 데이터에 따라 그래픽스 파이프라인을 수행할 수 있다.
- [0105] 도 12는 일실시예에 따라, 도 11에 도시된 그래픽 처리 장치의 동작 방법을 상세하게 설명하기 위한 순서도이다.
- [0106] 도 12를 참고하면, 단계 S1105에서, 그래픽 처리 장치는 복수의 컴포넌트들의 제1 상태에서 그래픽스 파이프라인을 수행하는 데 사용되는 제1 상태 버전 코드를 포함하는 제1 오퍼레이션 데이터를 획득할 수 있다.
- [0107] 단계 S1112에서, 그래픽 처리 장치는 복수의 컴포넌트들 중에서, 그래픽스 파이프라인이 수행될 때 컴포넌트에 대한 상태의 변경 빈도가 가장 높은 제1 컴포넌트를 결정할 수 있다.
- [0108] 또한, 제1 컴포넌트는 복수의 서브 컴포넌트들을 포함할 수 있다. 그래픽 처리 장치는 복수의 서브 컴포넌트들 중에서, 그래픽스 파이프라인이 수행될 때 서브 컴포넌트에 대한 상태의 변경 빈도가 가장 높은 제1 서브 컴포

년트를 결정할 수 있다.

- [0109] 단계 S1114에서, 그래픽 처리 장치는 제2 상태에 사용되는 제1 컴포넌트의 오퍼레이션에 코드에 기초하여, 베이스 오퍼레이션 데이터를 설정할 수 있다. 또한, 그래픽 처리 장치는 제2 상태에 사용되는 제1 서브 컴포넌트의 오퍼레이션 코드에 기초하여, 베이스 오퍼레이션 데이터를 설정할 수 있다.
- [0110] 단계 S1120에서, 그래픽 처리 장치는 베이스 오퍼레이션 데이터에 기초하여, 제1 오퍼레이션 데이터를 제2 상태 버전 코드를 포함하는 제2 오퍼레이션 데이터로 업데이트할 수 있다.
- [0111] 도 13은 다른 실시시에 따라, 그래픽 처리 장치의 동작 방법의 흐름을 나타낸 순서도이다.
- [0112] 도 13을 참고하면, 단계 S1310에서, 그래픽 처리 장치는 복수의 상태 버전 데이터의 사용 빈도에 기초하여, 복수의 상태 버전 데이터의 우선 순위를 결정할 수 있다.
- [0113] 단계 S1320에서, 그래픽 처리 장치는 우선 순위에 기초하여, 우선 순위가 가장 높은 제1 상태 버전 데이터를 보존하고, 제1 상태 버전 데이터보다 우선 순위가 낮은 제2 상태 버전 데이터를 그래픽스 파이프라인이 수행됨에 따라 업데이트 되도록 제어할 수 있다.
- [0114] 단계 S1330에서, 그래픽 처리 장치는 그래픽스 파이프라인이 사용될 상태 버전 데이터를 결정하고, 결정된 상태 버전 데이터에 따라 그래픽스 파이프라인을 수행할 수 있다.
- [0115] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다.
- [0116] 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다.
- [0117] 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0118] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다.
- [0119] 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0120] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다.
- [0121] 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.
- [0122] 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용

해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

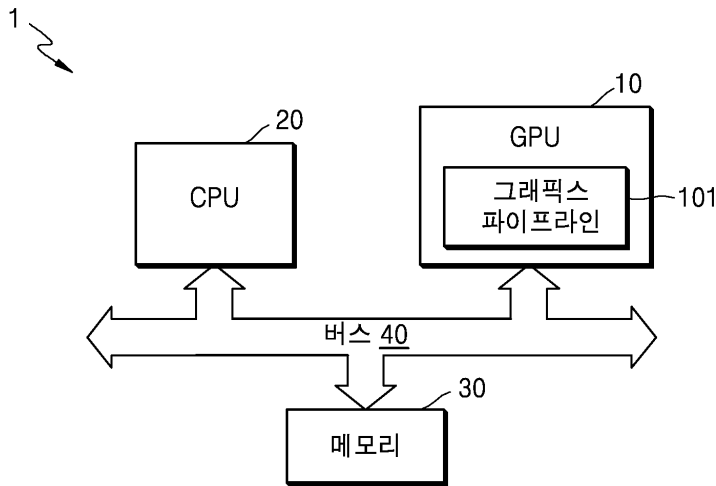
[0123] 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0124] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

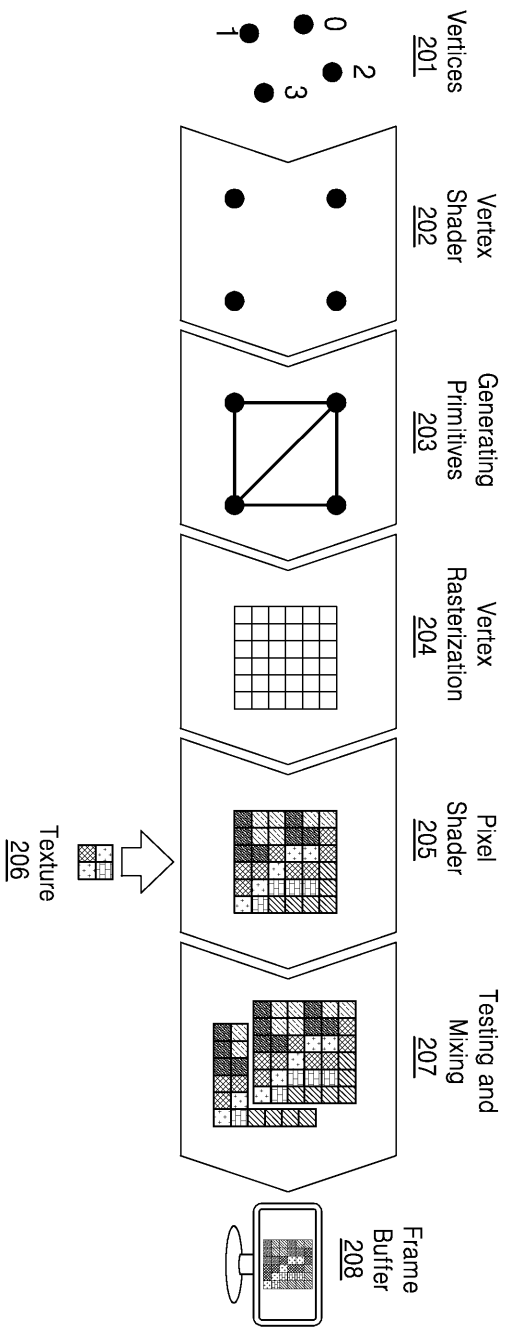
[0125] 그러므로, 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

도면

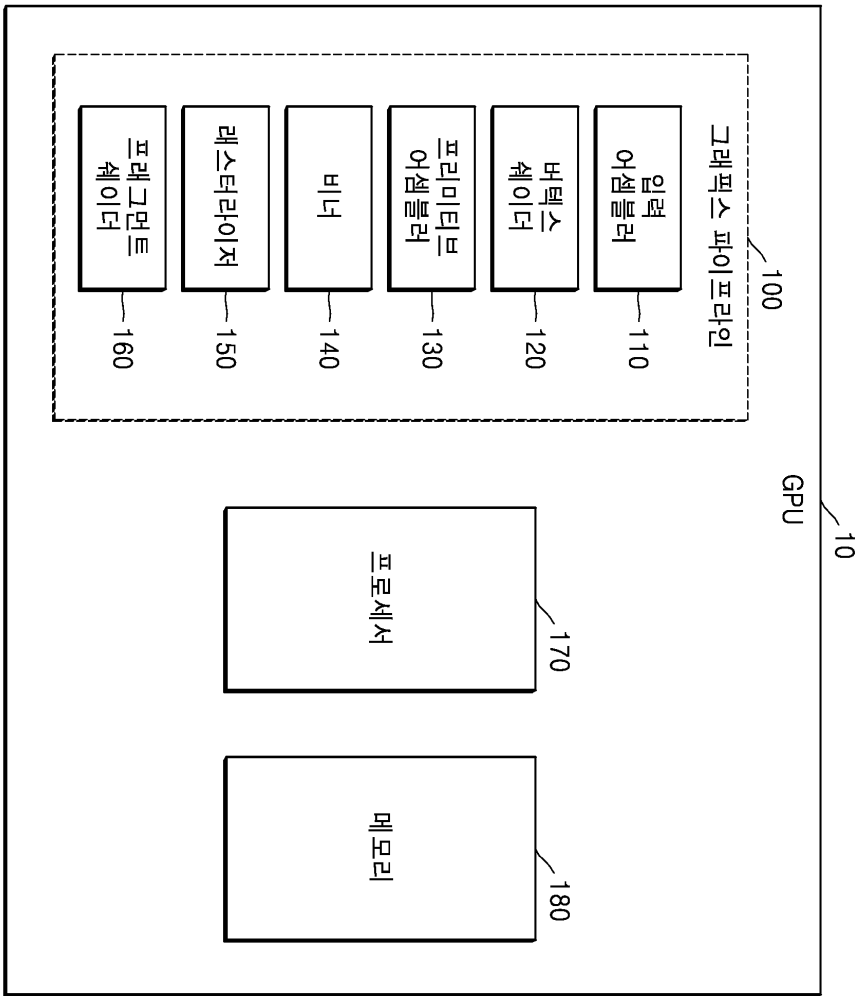
도면1



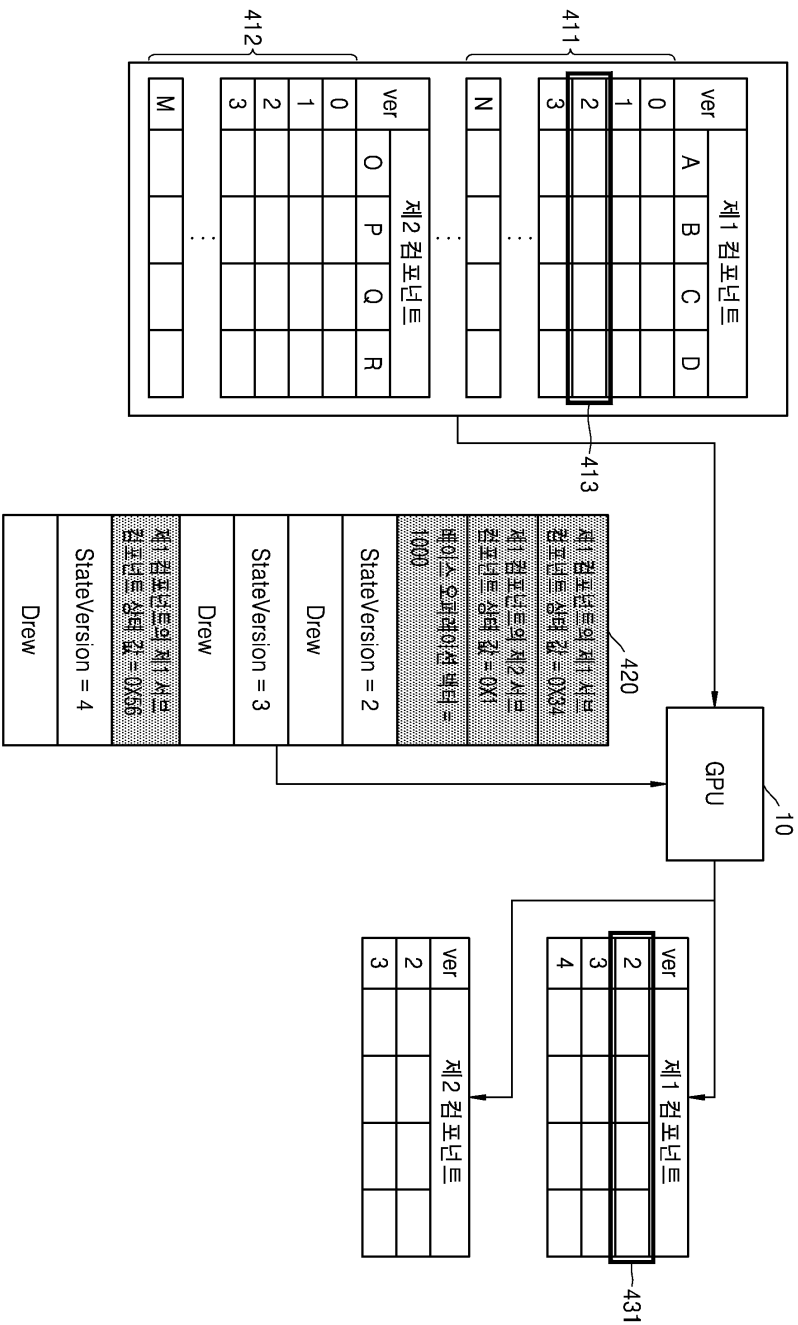
도면2



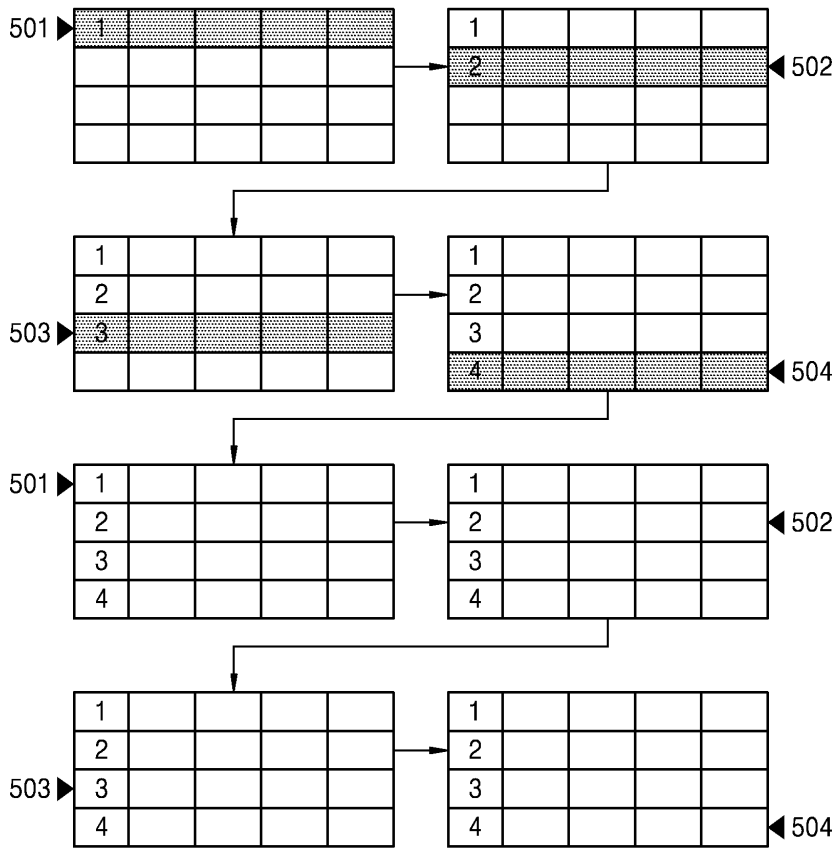
도면3



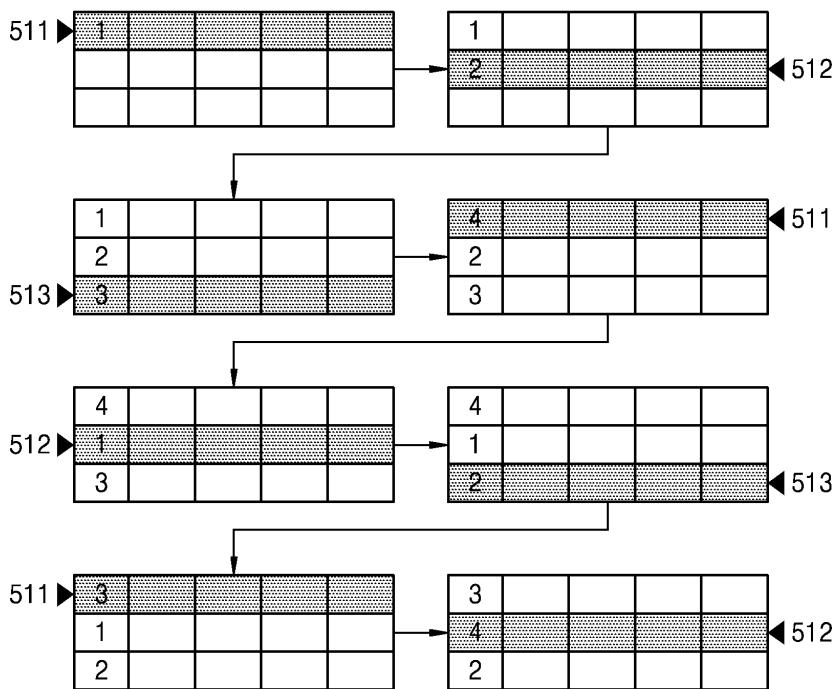
도면4



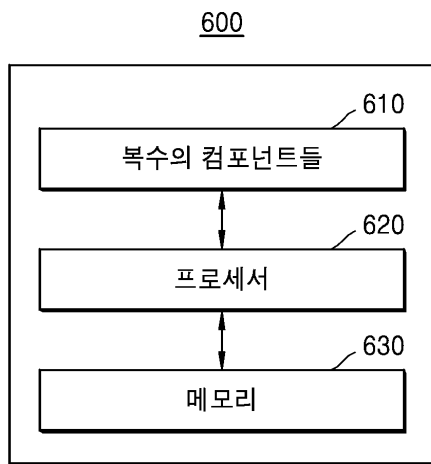
도면5a



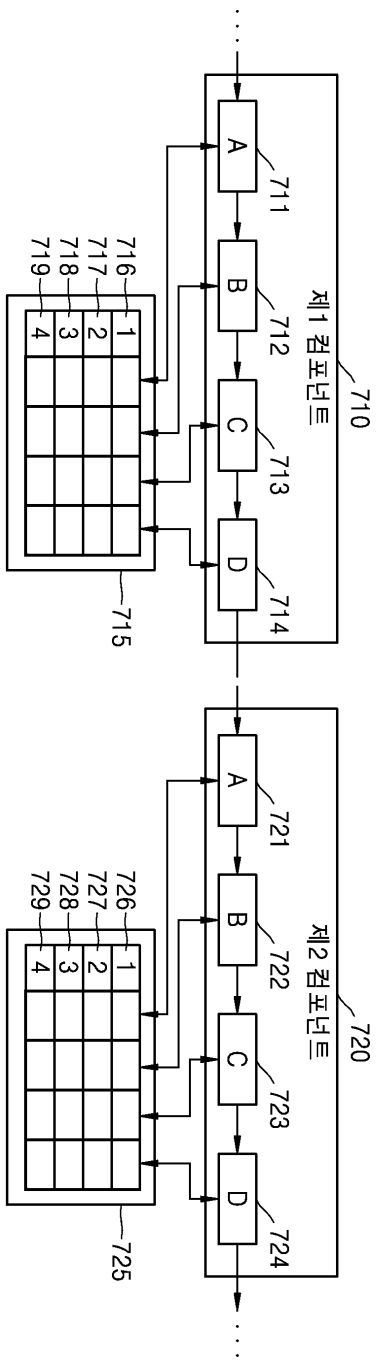
도면5b



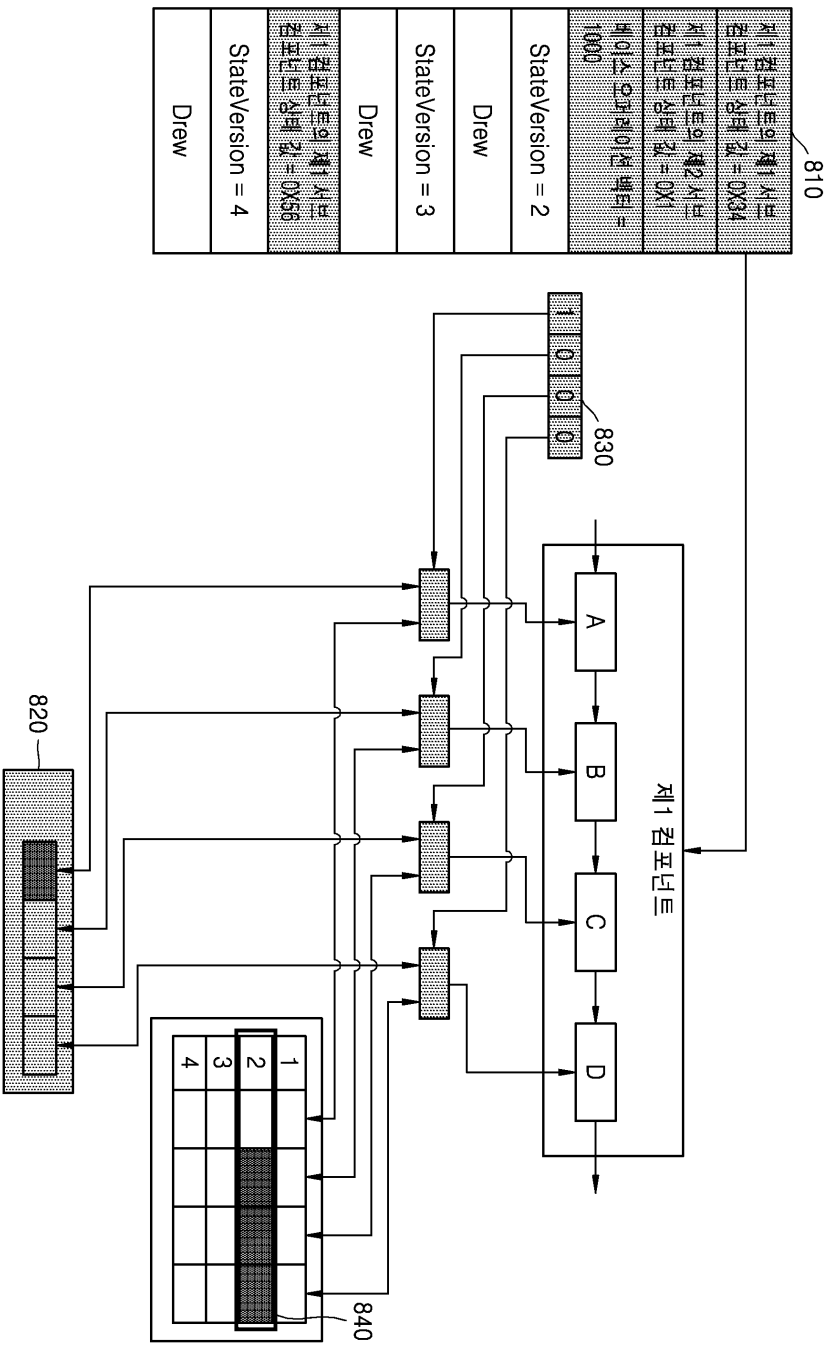
도면6



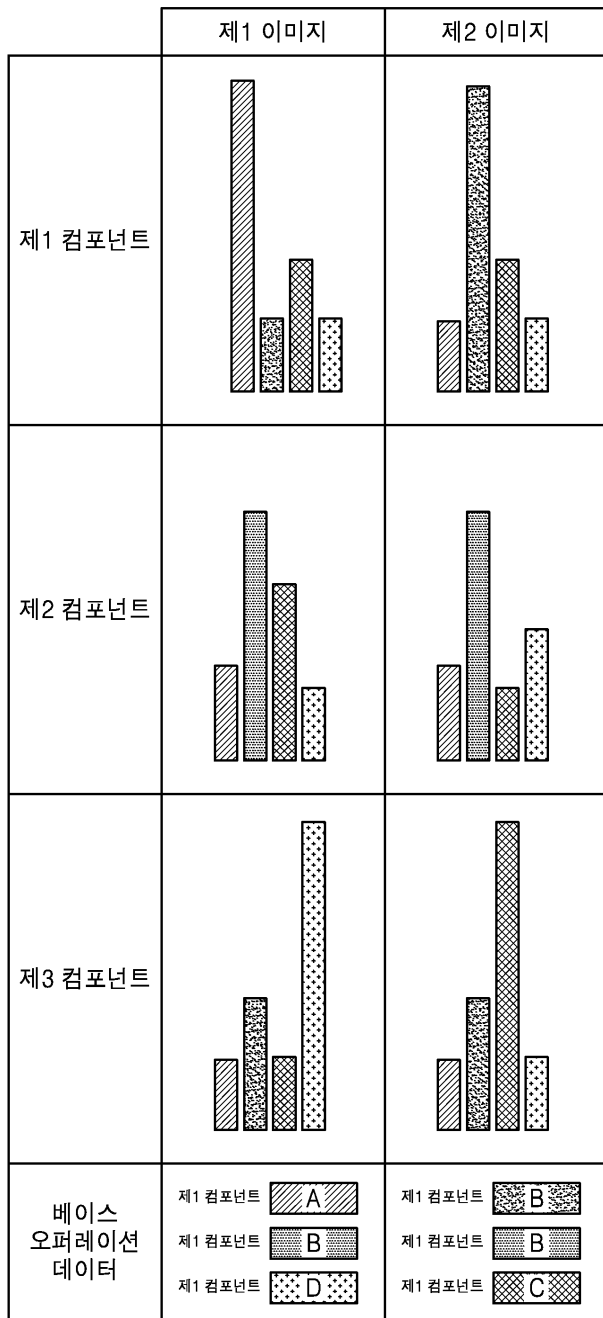
도면7



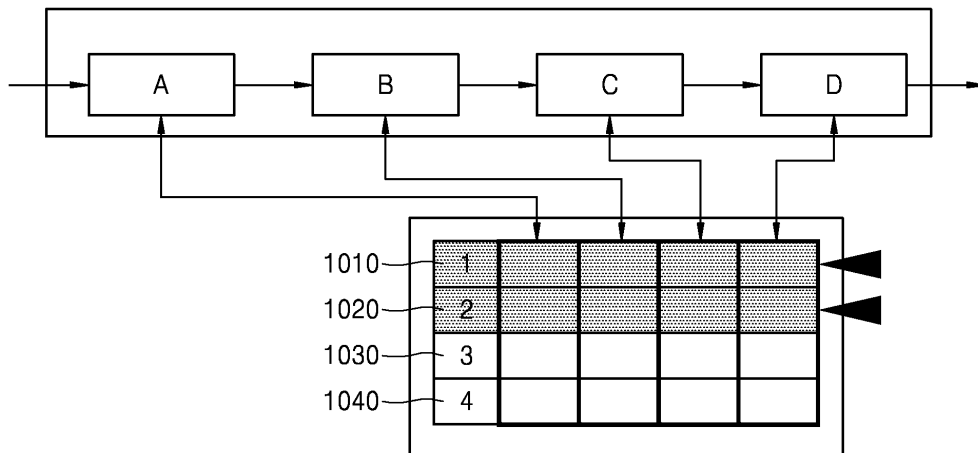
도면8



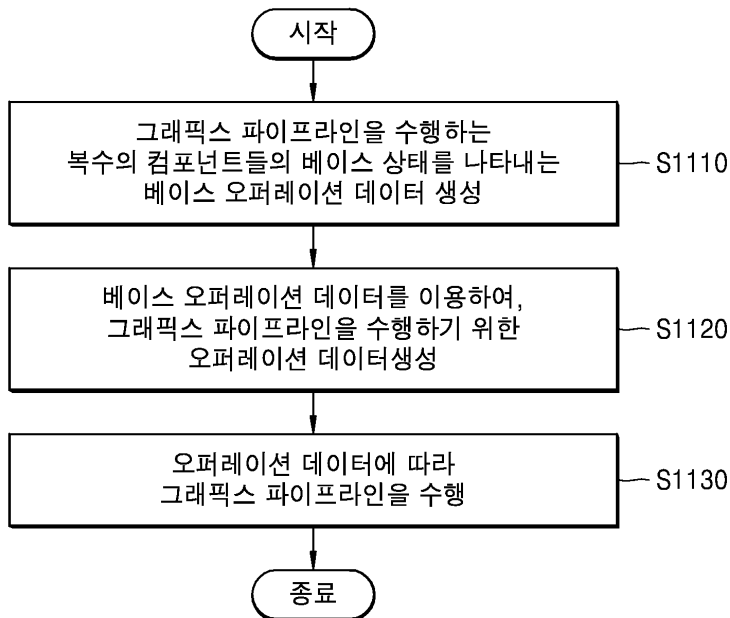
도면9



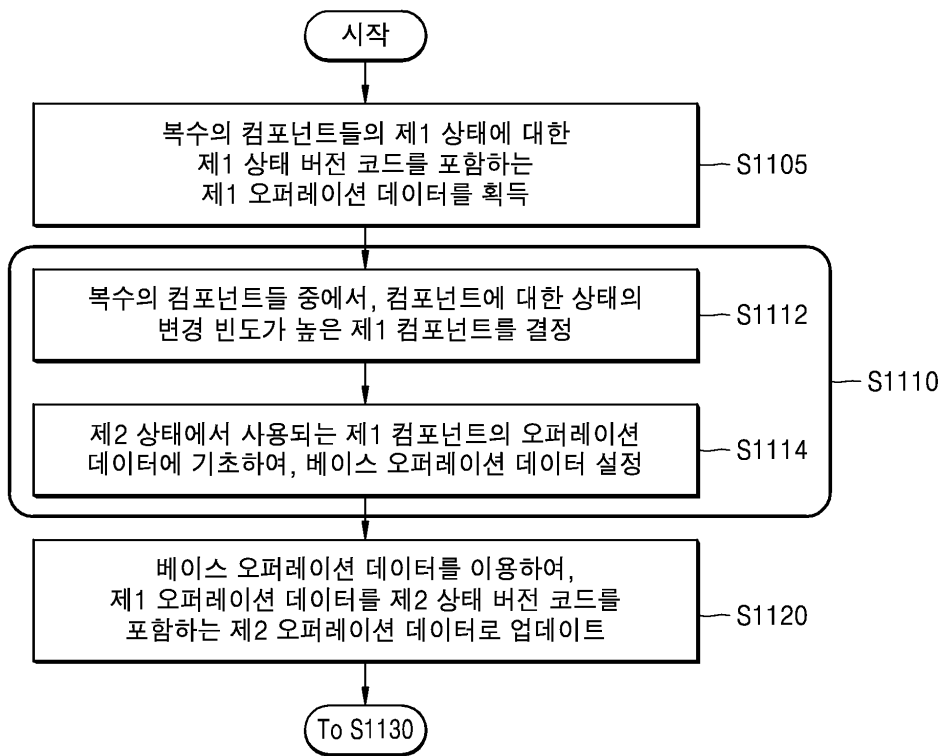
도면10



도면11



도면12



도면13

