



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년07월08일
(11) 등록번호 10-2682383
(24) 등록일자 2024년07월02일

(51) 국제특허분류(Int. Cl.)
G06F 9/38 (2024.01) G06F 12/1018 (2016.01)
G06F 9/48 (2018.01) G06T 1/20 (2018.01)
(52) CPC특허분류
G06F 9/3851 (2023.08)
G06F 12/1018 (2013.01)
(21) 출원번호 10-2021-7042931
(22) 출원일자(국제) 2020년05월29일
심사청구일자 2023년05월23일
(85) 번역문제출일자 2021년12월28일
(65) 공개번호 10-2022-0003141
(43) 공개일자 2022년01월07일
(86) 국제출원번호 PCT/US2020/035203
(87) 국제공개번호 WO 2020/243482
국제공개일자 2020년12월03일
(30) 우선권주장
16/426,613 2019년05월30일 미국(US)
(56) 선행기술조사문헌
KR1020150130499 A*
US08139070 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 95054 산타 클라라 어거스틴 드
라이브 2485
(72) 발명자
맥크레리 렉스 엘돈
미국 플로리다 32817 올란도 스위트 375 퀴드랭글
블러바드 3501
루오 위
미국 플로리다 32817 올란도 스위트 375 퀴드랭글
블러바드 3501
(74) 대리인
(뒷면에 계속)
박장원

전체 청구항 수 : 총 23 항

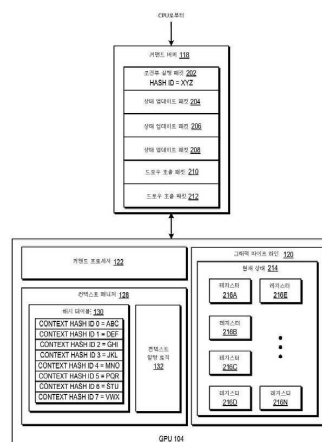
심사관 : 김평수

(54) 발명의 명칭 그래픽 컨텍스트 바운싱

(57) 요약

컨텍스트 바운싱 방법은, 그래픽 처리 유닛(GPU)[104]의 커맨드 프로세서[122]에서, 캡슐화된 상태에 대응하는 해시 식별자를 제공하는 조건부 실행 패킷[202]을 수신하는 단계를 포함한다. 캡슐화된 상태는 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷[204, 206]을 포함한다. 캡슐화된 상태 다음의 커맨드 패킷[210]은 캡슐화된 상태의 해시 식별자가 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 해시 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여 실행된다.

대표도 - 도2



(52) CPC특허분류

G06F 9/3877 (2013.01)

G06F 9/4881 (2013.01)

G06T 1/20 (2013.01)

(72) 발명자

와이즈 해리 제이.

미국 플로리다 32817 올란도 스위트 375 쿼드랭글
블러바드 3501

아쉬카르 알렉산더 푸아드

미국 플로리다 32817 올란도 스위트 375 쿼드랭글
블러바드 3501

맨터 미카엘

미국 플로리다 32817 올란도 스위트 375 쿼드랭글
블러바드 3501

명세서

청구범위

청구항 1

방법으로서,

그래픽 처리 유닛(GPU)의 커맨드 프로세서에서, 캡슐화된 상태에 대응하는 식별자를 제공하는 조건부 실행 패킷 [202]을 수신하는 단계 - 상기 캡슐화된 상태는 상기 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷을 포함함 -; 및

상기 캡슐화된 상태의 상기 식별자가 상기 GPU에 현재 저장된 활성 컨텍스트 상태의 저장된 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여, 상기 캡슐화된 상태 다음의 커맨드 패킷을 실행하는 단계를 포함하는, 방법.

청구항 2

제1항에 있어서,

상기 캡슐화된 상태의 상기 식별자가 상기 저장된 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것은,

상기 GPU에 현재 저장된 활성 컨텍스트 상태의 상기 복수의 식별자를 저장하는 식별자 테이블을 검색하기 위해 컨텍스트 매니저에 질의하는 것을 포함하는, 방법.

청구항 3

제2항에 있어서,

활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 하나와 일치하는 상기 식별자를 결정하는 것에 응답하여, 상기 캡슐화된 상태를 드레이닝(draining)하는 단계; 및

활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 상기 일치하는 하나를 상기 캡슐화된 상태 다음의 상기 커맨드 패킷에 할당하는 단계를 더 포함하는, 방법.

청구항 4

제2항에 있어서,

상기 식별자가 활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 어느 것과도 일치하지 않는다고 결정하는 것에 응답하여, 새로운 컨텍스트 상태 세트를 할당하는 단계; 및

상기 커맨드 패킷을 실행하기 전에 상기 캡슐화된 상태의 상기 하나 이상의 컨텍스트 상태 패킷을 실행하는 단계를 더 포함하는, 방법.

청구항 5

제4항에 있어서,

상기 식별자 테이블로부터, 상기 GPU에 현재 저장된 활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 하나를 회수(retiring)하는 단계; 및

식별자 테이블에서, 상기 새로운 컨텍스트 상태 세트에 대한 상기 캡슐화된 상태에 대응하는 상기 식별자를 저장하는 단계를 더 포함하는, 방법.

청구항 6

제1항에 있어서, 상기 커맨드 패킷은 드로우 호출(draw call)을 포함하는, 방법.

청구항 7

제1항에 있어서, 상기 커맨드 패킷은 상태 업데이트 패킷의 수에 대한 상기 캡슐화된 상태의 크기를 지정하는, 방법.

청구항 8

제1항에 있어서,

상기 캡슐화된 상태에 대응하는 식별자는 상기 캡슐화된 상태의 해시 표현(hashed representation)에 대응하는, 방법.

청구항 9

프로세서로서,

복수의 상태 레지스터; 및

커맨드 프로세서를 포함하며, 상기 커맨드 프로세서는,

캡슐화된 상태에 대응하는 식별자를 포함하는 조건부 실행 패킷을 수신하는 것 - 상기 캡슐화된 상태는 상기 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷을 포함함 -; 및

상기 캡슐화된 상태의 상기 식별자가 상기 프로세서에 현재 저장된 활성 컨텍스트 상태의 저장된 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여, 상기 캡슐화된 상태 다음의 커맨드 패킷을 실행하는 것을 수행하도록 구성되는, 프로세서.

청구항 10

제9항에 있어서,

상기 프로세서에 현재 저장된 활성 컨텍스트 상태의 상기 복수의 식별자를 저장하는 식별자 테이블을 포함하는 컨텍스트 매니저를 더 포함하는, 프로세서.

청구항 11

제10항에 있어서, 상기 프로세서는 상기 캡슐화된 상태의 식별자에 대해 상기 식별자 테이블을 검색하기 위해 상기 컨텍스트 매니저에 질의하도록 구성되는, 프로세서.

청구항 12

제10항에 있어서, 상기 프로세서는 추가로,

활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 하나와 일치하는 상기 식별자를 결정하는 것에 응답하여, 상기 캡슐화된 상태를 드레이닝하는 것; 및

활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 상기 일치하는 하나를 상기 캡슐화된 상태 다음의 상기 커맨드 패킷에 할당하는 것을 수행하도록 구성되는, 프로세서.

청구항 13

제10항에 있어서, 상기 프로세서는 추가로,

상기 식별자가 활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 어느 것과도 일치하지 않는다고 결정하는 것에 응답하여, 새로운 컨텍스트 상태 세트를 할당하는 것; 및

상기 커맨드 패킷을 실행하기 전에 상기 캡슐화된 상태의 상기 하나 이상의 컨텍스트 상태 패킷을 실행하는 것을 수행하도록 구성되는, 프로세서.

청구항 14

제13항에 있어서, 상기 프로세서는 추가로,

상기 식별자 테이블로부터, 상기 프로세서에 현재 저장된 활성 컨텍스트 상태의 상기 복수의 식별자 중 하나를 회수하는 것; 및

상기 식별자 테이블에서, 상기 새로운 컨텍스트 상태 세트에 대한 상기 캡슐화된 상태에 대응하는 상기 식별자를 저장하는 것을 수행하도록 구성되는, 프로세서.

청구항 15

제9항에 있어서, 상기 커맨드 패킷은 상태 업데이트 패킷의 수에 대한 상기 캡슐화된 상태의 크기를 지정하는, 프로세서.

청구항 16

제9항에 있어서,

상기 캡슐화된 상태에 대응하는 식별자는 상기 캡슐화된 상태의 해시 표현을 포함하는, 프로세서.

청구항 17

실행가능한 명령어 세트를 저장하는 비일시적 컴퓨터 판독가능 매체로서, 상기 실행가능한 명령어 세트는 적어도 하나의 프로세서를 조작하여,

상기 프로세서의 커맨드 프로세서에서, 캡슐화된 상태에 대응하는 식별자를 제공하는 조건부 실행 패킷[202]을 수신하고 - 상기 캡슐화된 상태는 상기 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷을 포함함 -;

및

상기 캡슐화된 상태의 상기 식별자가 상기 프로세서에 현재 저장된 활성 컨텍스트 상태의 저장된 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여, 상기 캡슐화된 상태 다음의 커맨드 패킷을 실행하게 하는, 비일시적 컴퓨터 판독가능 매체.

청구항 18

제17항에 있어서,

상기 실행가능한 명령어 세트는 또한, 적어도 하나의 프로세서를 조작하여,

상기 프로세서에 현재 저장된 활성 컨텍스트 상태의 상기 복수의 식별자를 저장하는 식별자 테이블을 검색하도록 컨텍스트 매니저에 질의하게 하는, 비일시적 컴퓨터 판독가능 매체.

청구항 19

제18항에 있어서,

상기 실행가능한 명령어 세트는 또한, 적어도 하나의 프로세서를 조작하여,

활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 하나와 일치하는 상기 식별자를 결정하는 것에 응답하여, 상기 캡슐화된 상태를 트레이닝하고; 및

활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 상기 일치하는 하나를 상기 캡슐화된 상태 다음의 상기 커맨드 패킷에 할당하게 하는, 비일시적 컴퓨터 판독가능 매체.

청구항 20

제18항에 있어서,

상기 실행가능한 명령어 세트는 또한, 적어도 하나의 프로세서를 조작하여,

상기 식별자가 활성 컨텍스트 상태의 상기 저장된 복수의 식별자 중 어느 것과도 일치하지 않는다고 결정하는 것에 응답하여, 새로운 컨텍스트 상태 세트를 할당하고; 및

상기 커맨드 패킷을 실행하기 전에 상기 캡슐화된 상태의 상기 하나 이상의 컨텍스트 상태 패킷을 실행하게 하는, 비일시적 컴퓨터 판독가능 매체.

청구항 21

제20항에 있어서,

상기 실행가능한 명령어 세트는 또한, 적어도 하나의 프로세서를 조작하여,

상기 식별자 테이블로부터, 상기 프로세서에 현재 저장된 활성 컨텍스트 상태의 상기 복수의 식별자 중 하나를 회수하고; 및

상기 식별자 테이블에서, 상기 새로운 컨텍스트 상태 세트에 대한 상기 캡슐화된 상태에 대응하는 상기 식별자를 저장하게 하는, 비밀시적 컴퓨터 판독가능 매체.

청구항 22

제17항에 있어서,

상기 커맨드 패킷은 상태 업데이트 패킷의 수에 대한 상기 캡슐화된 상태의 크기를 지정하는, 비밀시적 컴퓨터 판독가능 매체.

청구항 23

제17항에 있어서,

상기 캡슐화된 상태에 대응하는 식별자는 상기 캡슐화된 상태의 해시 표현에 대응하는, 비밀시적 컴퓨터 판독가능 매체.

발명의 설명

기술 분야

[0001] 본 발명은 일반적으로 그래픽 컨텍스트 바운싱에 관한 것이다.

배경 기술

[0002] 그래픽 처리 유닛(GPU)은 그래픽 처리 작업을 수행하도록 특별히 설계된 처리 유닛이다. GPU는, 예를 들어 비디오 게임 애플리케이션과 같은 최종 사용자(end-user) 애플리케이션에 필요한 그래픽 처리 작업을 실행할 수 있다. 전형적으로, 최종 사용자 애플리케이션과 GPU 사이에는 수개의 계층의 소프트웨어가 있다. 예를 들어, 일부 경우에는 최종 사용자 애플리케이션이 애플리케이션 프로그래밍 인터페이스(API)를 통해 GPU와 통신한다. API는, 최종 사용자 애플리케이션이 GPU에 의존하는 포맷이 아닌 표준화된 포맷으로 그래픽 데이터 및 커맨드를 출력할 수 있게 한다.

[0003] 많은 GPU는 그래픽 애플리케이션의 명령어를 실행하기 위한 그래픽 파이프라인을 포함한다. 그래픽 파이프라인은 명령어의 상이한 단계에서 동시에 작업하는 복수의 처리 블록을 포함한다. 파이프라이닝은 GPU가 명령어를 실행하는 데 필요한 단계들 간에 존재하는 병렬성(parallelism)의 이점을 취할 수 있게 한다. 그 결과, GPU는 더 짧은 시간 기간에 더 많은 명령어를 실행할 수 있다. 그래픽 파이프라인의 출력은 그래픽 파이프라인의 상태에 의존한다. 그래픽 파이프라인의 상태는 그래픽 파이프라인에 의해 국부적으로 저장되는 상태 패키지(예를 들어, 텍스처 핸들러, 셰이더 상수, 변환 행렬 등을 포함하는 컨텍스트 특정 상수)에 기초하여 업데이트된다. 컨텍스트 특정 상수는 국부적으로 유지되므로, 그래픽 파이프라인에 의해 신속하게 액세스될 수 있다.

도면의 간단한 설명

[0004] 본 개시는 첨부 도면을 참조함으로써 당업자에게 더 잘 이해될 수 있고 그의 많은 특징 및 이점이 명백해질 수 있다. 상이한 도면에서 동일한 참조 부호의 사용은 유사하거나 동일한 항목을 나타낸다.

도 1은 일부 실시형태에 따른 그래픽 컨텍스트 바운싱을 사용하도록 구성된 그래픽 처리 유닛을 포함하는 시스템을 도시하는 블록도이다.

도 2는 일부 실시형태에 따른 도 1의 GPU에서의 커맨드 스트림 명령어의 처리를 도시하는 블록도이다.

도 3은 일부 실시형태에 따른 도 1의 GPU에서의 캡슐화된 상태의 드레인(drain)을 도시하는 블록도이다.

도 4는 일부 실시형태에 따른 그래픽 컨텍스트 바운싱의 방법을 도시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0005] 그래픽 처리를 수행하기 위해, 시스템의 중앙 처리 유닛(CPU)은 종종 CPU의 명령어에 따라 객체를 드로잉하도록 GPU에 지시하는 일련의 커맨드를 포함하는 드로우 호출(draw call)과 같은 호출을 GPU에 발행한다. 드로우 호출이 GPU 그래픽 파이프라인을 통해 처리될 때, 드로우 호출은 다양한 구성 가능한 설정을 사용하여 메시와 텍스처가 렌더링되는 방법을 결정한다. 일반적인 GPU 워크플로우는 메모리 어레이의 상수 값을 업데이트하는 것과 그 후에 상수를 데이터로서 사용하여 드로우 동작을 수행하는 것을 포함한다. 메모리 어레이가 부여된 상수 세트를 포함하는 GPU는 특정 상태에 있는 것으로 간주될 수 있다. 컨텍스트 상태("렌더링 상태", "GPU 상태" 또는 간단히 "상태"라고도 칭함)라고 지칭되는 이러한 상수 및 설정은 렌더링의 다양한 양태에 영향을 미치며, GPU가 객체를 렌더링하는 데 필요한 정보를 포함한다. 컨텍스트 상태는 메시가 렌더링되는 방법에 대한 정의를 제공하며, 현재 버텍스/인덱스 버퍼, 현재 버텍스/픽셀 셰이더 프로그램, 셰이더 입력, 텍스처, 재료, 조명, 투명도 등과 같은 정보를 포함한다. 컨텍스트 상태는 그래픽 파이프라인에서 렌더링되는 드로우 또는 드로우 세트에 고유한 정보를 포함한다. 따라서, 컨텍스트는 무언가를 정확하게 드로잉하는 데 필요한 GPU 파이프라인 상태를 지칭한다.
- [0006] 많은 GPU는 파이프라이닝으로 알려진 기술을 사용하여 명령어를 실행한다. 파이프라이닝은, GPU가 명령어의 상이한 단계에서 동시에 작업할 수 있게 하므로, 명령어를 실행하는 데 필요한 단계들 간에 존재하는 병렬성의 이점을 취할 수 있다. 그 결과, GPU는 더 짧은 시간 기간에 더 많은 명령어를 실행할 수 있다. 그래픽 파이프라인에 의해 출력되는 비디오 데이터는 그래픽 파이프라인에 의해 국부적으로 저장되는 상태 패키지(예를 들어, 컨텍스트 특정 상수)에 의존한다. GPU에서는, GPU의 상태를 설정하고, 드로우 동작을 수행하고, 그 후에 다음의 드로우 동작 전에 소수의 상태 변경만을 행하는 것이 일반적이다. 상태 설정(예를 들어, 메모리 내의 상수 값)은 종종 하나의 드로우 동작으로부터 다음의 드로우 동작까지 동일하게 유지된다.
- [0007] 컨텍스트 특정 상수는 그래픽 파이프라인에 의한 신속한 액세스를 위해 국부적으로 유지된다. 그러나, GPU 하드웨어는 일반적으로 메모리가 제한적이며 제한된 수의 컨텍스트 상태 세트를 국부적으로만 저장(및 이에 따라 작동)한다. 따라서, GPU는, 다른 것을 드로잉하려면 그래픽 파이프라인 상태를 변경해야 하기 때문에 종종 새로운 컨텍스트 레지스터 세트에서 작업을 시작하기 위해 컨텍스트 상태를 변경할 것이다. GPU는 새롭게 제공된 컨텍스트로 컨텍스트 롤을 수행하여, 새로운 레지스터 값을 프로그래밍하여 임의의 새로운 상태 업데이트를 적용하기 전에 현재 레지스터를 새롭게 할당된 컨텍스트에 복사함으로써 현재 컨텍스트를 해제한다. 국부적으로 저장된 제한된 수의 컨텍스트 상태 세트로 인해, GPU는 때때로 컨텍스트를 소진하여, 그래픽 파이프라인은 새로운 컨텍스트가 할당될 수 있도록 컨텍스트가 자유롭게 될 때까지 대기하는 동안 스톨(stall)된다. 다양한 실시형태에서, CPU는 이전에 활성이었던 컨텍스트 상태를 재사용하고, 컨텍스트 바운싱 없이 현재 컨텍스트를 해제하고 새로운 컨텍스트를 할당한다. 컨텍스트가 사용 불가능한 경우, GPU는 컨텍스트가 자유롭게 될 때까지 대기해야 하므로 스톨을 유발한다.
- [0008] GPU 시스템 성능을 개선하기 위해, 도 1 내지 도 4는 컨텍스트 상태에 해시 기반 식별자를 할당하고 새로운 컨텍스트의 할당없이 일치하는 컨텍스트(컨텍스트 바운싱이라고 칭함)로 전환함으로써 컨텍스트 상태를 관리하기 위한 시스템 및 방법을 도시한다. 다양한 실시형태에서, 컨텍스트 바운싱 방법은 그래픽 처리 유닛(GPU)의 커맨드 프로세서에서 캡슐화된 상태에 대응하는 해시 식별자를 제공하는 조건부 실행 패킷을 수신하는 단계를 포함한다. 캡슐화된 상태는 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷을 포함한다. 캡슐화된 상태 다음의 커맨드 패킷은 캡슐화된 상태의 해시 식별자가 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 해시 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여 실행된다. 다양한 실시형태에서, 캡슐화된 상태의 해시 식별자가 복수의 해시 식별자 중 하나와 일치하는지 여부를 결정하는 것은 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 해시 식별자를 저장하는 해시 테이블을 검색하도록 컨텍스트 매니저에 질의하는 것을 포함한다.
- [0009] 일부 실시형태에서, 컨텍스트 바운싱은 또한 해시 식별자가 활성 컨텍스트 상태의 복수의 해시 식별자 중 하나와 일치한다고 결정하는 것에 응답하여 캡슐화된 상태를 드레이닝하는 것을 포함한다. 다른 실시형태에서는, 해시 식별자가 활성 컨텍스트 상태의 복수의 해시 식별자 중 어느 것과도 일치하지 않는다고 결정하는 것에 응답하여 새로운 컨텍스트 상태 세트가 할당된다. 또한, 캡슐화된 상태의 하나 이상의 컨텍스트 상태 패킷은 커맨드 패킷을 실행하기 전에 실행된다. 기존 컨텍스트로의 바운싱 백(Bouncing back)은(즉, 새로운 컨텍스트의 할당없음) 컨텍스트의 소진 및 그래픽 파이프라인의 스톨 가능성을 감소시킨다.
- [0010] 도 1은 일부 실시형태에 따른 그래픽 컨텍스트 바운싱을 사용하도록 구성된 그래픽 처리 유닛을 포함하는 시스템(100)을 도시하는 블록도이다. 도시된 실시예에서, 시스템(100)은 중앙 처리 유닛(CPU)(102), 그래픽 처리 유닛(GPU)(104), GPU(104)에 의해 이용되는 디바이스 메모리(106), 및 CPU(102)와 GPU(104)에 의해 공유되는 시

시스템 메모리(108)를 포함한다. 다양한 실시형태에서, GPU(104)는 전용 디바이스, 수개의 디바이스일 수 있거나, 또는 더 큰 디바이스(예를 들어, 노스 브리지 디바이스)에 통합될 수 있다. 메모리(106, 108)는 더블 데이터 레이트 동적 랜덤 액세스 메모리(DDR DRAM), 그래픽 DDR DRAM(GDDR DRAM) 등과 같은 임의의 다양한 랜덤 액세스 메모리(RAM) 또는 이들의 조합을 포함한다. GPU(104)는 버스(110)를 통해 CPU(102), 디바이스 메모리(106) 및 시스템 메모리(108)와 통신한다. 버스(110)는 PCI(Peripheral Component Interface) 버스, AGP(Accelerated Graphics Port) 버스, PCIE(PCI Express) 버스 등을 포함하지만 이에 한정되지 않는 컴퓨팅 시스템에서 사용되는 임의의 유형의 버스를 포함한다.

[0011] CPU(102)는 GPU(104)에서 처리하도록 의도된 명령어를 커맨드 버퍼에 전송한다. 다양한 실시형태에서, 커맨드 버퍼는, 예를 들어 버스(110)에 결합된 별도의 메모리(예를 들어, 디바이스 메모리(106)) 내의 시스템 메모리(108)에 배치될 수 있다.

[0012] 도시된 바와 같이, CPU(102)는 그래픽 커맨드를 생성하기 위한 하나 이상의 애플리케이션(112) 및 사용자 모드 드라이버(116)(또는 커널 모드 드라이버와 같은 다른 드라이버)를 실행하는 것과 같은 다수의 프로세스를 포함한다. 다양한 실시형태에서, 하나 이상의 애플리케이션(112)은 GPU(104)의 기능을 이용하는 애플리케이션을 포함한다. 애플리케이션(112)은 그래픽 사용자 인터페이스(GUI) 및/또는 그래픽 장면(graphics scene)을 렌더링하도록 GPU(104)에 지시하는 하나 이상의 그래픽 명령어를 포함할 수 있다. 예를 들어, 그래픽 명령어는 GPU(104)에 의해 렌더링될 하나 이상의 그래픽 프리미티브 세트를 정의하는 명령어를 포함할 수 있다.

[0013] 일부 실시형태에서, 애플리케이션(112)은 그래픽 애플리케이션 프로그래밍 인터페이스(API)(114)를 이용하여 사용자 모드 드라이버(116)(또는 유사한 GPU 드라이버)를 호출한다. 사용자 모드 드라이버(116)는 하나 이상의 그래픽 프리미티브를 디스플레이 가능한 그래픽 이미지로 렌더링하기 위해 GPU(104)에 하나 이상의 커맨드를 발행한다. 애플리케이션(112)에 의해 사용자 모드 드라이버(116)에 발행된 그래픽 명령어에 기초하여, 사용자 모드 드라이버(116)는 GPU(104)가 그래픽을 렌더링하기 위해 수행할 하나 이상의 동작을 지정하는 하나 이상의 그래픽 커맨드를 공식화한다. 일부 실시형태에서, 사용자 모드 드라이버(116)는 CPU(102)에서 실행하는 애플리케이션(112)의 일부이다. 예를 들어, 사용자 모드 드라이버(116)는 CPU(102)에서 실행하는 게임 애플리케이션의 일부일 수 있다. 마찬가지로, 커널 모드 드라이버(도시되지 않음)는 CPU(102)에서 실행하는 운영 체제의 일부일 수 있다. 사용자 모드 드라이버(116)에 의해 생성된 그래픽 커맨드는 디스플레이를 위한 이미지 또는 프레임을 생성하도록 의도된 그래픽 커맨드를 포함한다. 사용자 모드 드라이버(116)는 API(114)로부터 수신된 표준 코드를 GPU(104)에 의해 이해되는 명령어의 네이티브 포맷(native format)으로 변환한다. 사용자 모드 드라이버(116)는 전형적으로 GPU(104)의 제조사에 의해 기록된다. 사용자 모드 드라이버(116)에 의해 생성된 그래픽 커맨드는 실행을 위해 GPU(104)에 전송된다. 그 후, GPU(104)는 그래픽 커맨드를 실행하고 그 결과를 사용하여 디스플레이 스크린에 디스플레이되는 것을 제어할 수 있다.

[0014] 다양한 실시형태에서, CPU(102)는 GPU(104)를 위해 의도된 그래픽 커맨드를 커맨드 버퍼(118)에 전송한다. 도 1에는 도시의 편의를 위해 별도의 모듈로 도시되어 있지만, 커맨드 버퍼(118)는, 예를 들어 디바이스 메모리(106), 시스템 메모리(108), 또는 버스(110)에 통신 가능하게 결합된 별도의 메모리에 배치될 수 있다. 커맨드 버퍼(118)는 GPU(104)에 대한 입력을 포함하는 그래픽 커맨드의 스트림을 일시적으로 저장한다. 그래픽 커맨드의 스트림은, 예를 들어 하나 이상의 커맨드 패킷 및/또는 하나 이상의 상태 업데이트 패킷을 포함한다. 일부 실시형태에서, 커맨드 패킷은 디스플레이를 위해 출력될 이미지 데이터에 대한 프로세스를 실행하도록 GPU(104)에 지시하는 드로우 커맨드(호환 가능하게 "드로우 호출"이라고도 칭함)를 포함한다. 예를 들어, 드로우 커맨드는 메모리에 저장된 하나 이상의 버텍스의 그룹에 의해 정의된(예를 들어, 버텍스 버퍼에 정의된) 기하학적 구조를 렌더링하도록 GPU(104)에 지시할 수 있다. 하나 이상의 버텍스의 그룹에 의해 정의된 기하학적 구조는 일부 실시형태에서 렌더링될 복수의 프리미티브에 대응한다.

[0015] 다양한 실시형태에서, 상태 커맨드(호환 가능하게 "컨텍스트 상태 패킷" 또는 "상태 업데이트 패킷"이라고도 칭함)는 하나 이상의 상태 변수(예를 들어, 드로우 컬러)를 변경하도록 GPU(104)에 지시한다. 일부 실시형태에서, 컨텍스트 상태 패킷은 GPU(104)에서 그래픽 파이프라인(120)의 상태를 업데이트하는 상수 또는 상수 세트를 포함한다. 상태 업데이트 패킷은, 예를 들어 드로우 호출의 실행 중에 드로잉 또는 블렌딩될 컬러를 업데이트할 수 있다. 드로우 호출 및 컨텍스트 상태 패킷과 관련하여 본 명세서에 설명되었지만, 당업자는 본 개시의 범위를 벗어나지 않고 다른 커맨드가 이용될 수 있음을 인식할 것이다.

[0016] GPU(104)는, (예를 들어, 커맨드 버퍼(118) 및 버스(110)를 통해) CPU(102)로부터 실행될 커맨드 스트림 내의 커맨드를 수신하고 그래픽 파이프라인(120)에서 이러한 커맨드의 실행을 조정하는 커맨드 프로세서(122)를 포함

한다. 다양한 실시형태에서, 커맨드 프로세서(122)는 하드웨어, 펌웨어, 소프트웨어, 또는 이들의 조합으로 구현된다. 커맨드 스트림은 전술한 바와 같이 하나 이상의 드로우 호출 및 컨텍스트 상태 패킷을 포함한다. 커맨드 프로세서(122)는 또한 그래픽 파이프라인(120)의 레지스터에 기록된 컨텍스트 상태를 관리한다. 다양한 실시형태에서, 컨텍스트 상태 패킷을 수신하는 것에 응답하여, 커맨드 프로세서는 GPU(104) 내의 하나 이상의 상태 레지스터를 컨텍스트 상태 패킷에 기초한 특정 값으로 설정하고 및/또는 컨텍스트 상태 패킷에 기초하여 하나 이상의 고정 함수 처리 유닛을 구성한다.

[0017] 도 1에는 하나의 커맨드 프로세서(122)를 갖는 것으로 도시되어 있지만, 당업자는 GPU(104)가 하드웨어 큐로부터 패킷을 검색하고 실행하기 위한 임의의 수 및 유형의 커맨드 프로세서를 포함할 수 있음을 인식할 것이다. 다양한 실시형태에서, "패킷"은 단일 커맨드를 인코딩하는 메모리 버퍼를 지칭한다. 다른 유형의 패킷이 하드웨어 큐, 메모리 버퍼 등에 저장될 수 있다. 또한, 본 명세서에서 사용된 용어 "블록"은 ASIC에 포함된 처리 모듈, CPU의 실행 파이프라인, 및/또는 GPU의 그래픽 파이프라인을 지칭한다. 이러한 처리 모듈은 산술 논리 유닛, 적산/제산 유닛, 부동 소수점 유닛, 컬러 버퍼, 버텍스 셰이더, 픽셀 셰이더, 클리핑 유닛 또는 당업자에게 명백한 일부 다른 처리 모듈을 포함할 수 있지만 이에 한정되지 않는다.

[0018] 그래픽 파이프라인(120)은 스테이지 A(124a), 스테이지 B(124b) 내지 스테이지 N(124n)을 포함하는 다수의 스테이지(124)를 포함한다. 다양한 실시형태에서, 다양한 스테이지(124) 각각은 드로우 호출의 다양한 양태를 실행하는 그래픽 파이프라인(120)의 스테이지를 나타낸다. 커맨드 프로세서(122)는 동작 상태를 저장하고 업데이트하기 위해 컨텍스트 레지스터의 로컬 뱅크에 컨텍스트 상태 업데이트를 기록한다. 도 1에 도시된 바와 같이, 커맨드 프로세서는 현재 상태 레지스터(126)에 상태 업데이트를 기록한다.

[0019] 종래의 컨텍스트 관리 기술은 정보 또는 커맨드 패킷을 버퍼(때로는 큐 버퍼라고 칭함)에 디스패치한다. 이러한 큐 버퍼는 실행 중인 프로그램이 커맨드를 제출하는 FIFO(first-in-first-out)와 같은 순서로 생성된다. 그러나, GPU의 메모리 용량으로 인해, 제한된 수의 고유한 렌더링 상태 세트만이 저장될 수 있으므로 임의의 부여된 순간에 그래픽 파이프라인에서 동작된다. 따라서, 순차 프로세스는 하나의 상태 세트에서 실행될 수 있지만, 드라이버가 드로우 동작을 수행한 후에 컨텍스트 상태를 변경하는 경우 CP는 GPU가 다른 상태 세트로 변경되면 완료로 보장하기 위해 판독이 수반되는 레지스터 기록 시퀀스를 실행해야 한다. 새로운 컨텍스트에 대한 이러한 컨텍스트 롤링 프로세스는 현재 컨텍스트를 해제하기 위해 그래픽 파이프라인 아래로 이벤트를 전송하는 커맨드 프로세서(122)를 포함한다. 완료되면, 커맨드 프로세서는 통지를 받아 컨텍스트가 재사용될(또는 새로운 컨텍스트가 할당될) 수 있게 한다. 그러나, 이러한 컨텍스트 롤링 프로세스는 현재 컨텍스트와 관련된 드로우 호출이 그래픽 파이프라인의 다양한 단계를 통해 처리하는 것을 완료할 때까지 사용할 컨텍스트의 소진 및 그래픽 파이프라인의 스틀 가능성을 증가시킨다. 일반적으로, GPU 동작은 임의의 부여된 순간에 진행 중의 다양한 작업량으로 파이프라인되고; 상태를 변경하기 전에 진행 중인 작업이 회수(retire)되어야 하도록 그래픽 파이프라인을 스틀하는 것은 GPU 백엔드의 고갈 가능성을 증가시킨다.

[0020] 컨텍스트 상태를 보다 효율적으로 관리하기 위해, GPU(104)는 그래픽 컨텍스트 매니저(128)를 포함한다. 그래픽 컨텍스트 매니저(128)는 GPU(104)의 레지스터에 현재 저장된 컨텍스트 상태 세트에 대응하는 해시 기반 식별자 세트를 저장하는 해시 테이블(130)을 유지한다. 그래픽 컨텍스트 매니저(128)는 본 명세서에 설명된 해시 식별자 검색 및 컨텍스트 할당 동작을 수행하기 위한 컨텍스트 할당 로직 모듈(132)을 더 포함한다. 본 명세서에 추가로 설명되는 바와 같이, 사용자 모드 드라이버(116)는 사용자 모드 드라이버(116)가 그래픽 커맨드로 프로그래밍하는 새로운 컨텍스트 상태를 식별하기 위해 고유 해시 식별자를 제공한다. 다양한 실시형태에서, 사용자 모드 드라이버(116)는 GPU(104)에서 모든 활성 상태를 스캔하고 새로운 컨텍스트의 고유 해시 식별자가 현재 활성 컨텍스트 상태의 복수의 해시 식별자(즉, 해시 테이블(130)에 저장된 해시 식별자) 중 임의의 하나와 일치하는지 여부를 결정하기 위해 새로운 상태 패킷(또는 다른 토큰 방법)을 통해 커맨드 프로세서(122)에 표시한다.

[0021] 일부 실시형태에서, 커맨드 프로세서(122)는 해시 테이블(130)이 새로운 컨텍스트에 대응하는 요청된 고유 해시 식별자를 저장하는지 여부를 결정하기(즉, 컨텍스트가 사용 가능하다는 것을 표시하기) 위해 해시 테이블(130)을 검색하도록 그래픽 컨텍스트 매니저(128)에 질의한다. 해시 테이블(130)이 요청된 고유 해시 식별자를 포함하지 않는 경우에는, 그래픽 컨텍스트 매니저(128)는 요청된 고유 해시 식별자를 사용하여 새로운 컨텍스트를 할당한다. 그러나, 해시 테이블(130)이 요청된 고유 해시 식별자를 포함하는(이에 따라 요청된 고유 해시 식별자가 GPU(104)에서 이미 활성화된 상태에 대응함을 알려주는) 경우에는, 그래픽 컨텍스트 매니저(128)는 해당 컨텍스트를 리턴한다.

[0022] 새로운 컨텍스트를 할당하거나 컨텍스트가 해제되기를 잠재적으로 대기하는(이에 따라 그래픽 파이프라인 스틀

을 초래하는) 대신에, GPU(104)는 새로운 컨텍스트의 할당없이(예를 들어, 캐시 메모리가 캐시 라인을 공유하는 방법과 유사한 방식으로) GPU(104)에서 이미 실행 중인 기존의 일치 컨텍스트로 다시 전환함으로써 컨텍스트 바운싱을 수행한다. 따라서, GPU(104)는 상태 세트가 여전히 이용 가능하다면(즉, GPU(104)에서 로딩된다면) 다수의 상태 세트에 대해 비순차적 프로세스를 실행할 수 있다. 이 접근 방식은, 특히 애플리케이션(112)이 소수의 상태들 사이에서 신속하게 토글링(종래의 컨텍스트 관리 기술로 각 드로우 후에 상태가 변화함에 따라 컨텍스트 롤링과 새로운 상태를 획득하는 것, 및 컨텍스트의 신속한 소진을 포함함)할 때, GPU(104)의 온보드 상태를 보다 효율적으로 활용하고 사용할 컨텍스트의 소진과 관련된 스톱 수를 감소시킨다.

[0023] 도 2는 일부 실시형태에 따른 CPU(102)가 GPU(104)를 위해 의도된 명령어를 전송하는 실시형태를 도시하는 블록도이다. 도시된 바와 같이, CPU(102)는 커맨드 버퍼(118)에 명령어를 전송한다. 커맨드 버퍼(118)는 CPU(102)로부터의 커맨드 스트림의 커맨드를 일시적으로 저장한다. 커맨드 스트림은, 예를 들어 조건부 실행 패킷, 상태 업데이트 패킷 및 드로우 호출 패킷을 포함한다. 다양한 실시형태에서는, 당업자에게 명백한 바와 같이 다른 커맨드가 커맨드 스트림에 포함될 수 있다.

[0024] 조건부 실행 패킷은 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷에 대응하는 해시 식별자를 제공하는 도 1의 사용자 모드 드라이버(116)(또는 다른 소프트웨어 드라이버)에 의해 삽입된 커맨드이다. 일부 실시형태에서, 조건부 실행 패킷은 또한 dwords("double word"의 약어)의 컨텍스트 상태 패킷의 다음 그룹의 크기에 대응하는 dword 데이터를 포함한다. 컨텍스트 상태 패킷은 그 크기가 단일 dword로부터 많은 dword까지 가변하므로, dword 데이터는 조건부 실행 패킷의 해시 식별자와 관련된 컨텍스트 상태 패킷의 수(N)를 정의하고, 해당 해시 식별자 다음에 오는 다음 드로우(들)에 대한 컨텍스트 상태를 설정하는 데 필요한 모든 컨텍스트 레지스터 업데이트를 추가로 나타낸다. 또한, 본 명세서에 추가로 설명되는 바와 같이, dword 데이터는 또한 그래픽 컨텍스트 매니저(128)가 리턴하고 해당 해시 식별자와 관련된 컨텍스트가 GPU(104)에서 이미 활성화되어 있음을 나타내는 경우에 폐기할 dword의 수를 정의한다. 해당 결정에 기초하여, 커맨드 프로세서(122)는 (조건부 실행 패킷에 의해 제공되는) 해당 특정 해시 ID와 관련된 모든 상태가 폐기될 때까지 해당의 부여된 dword 데이터 크기에 기초하여 커맨드 버퍼(118)로부터 폴링(pulling)한다.

[0025] 상태 업데이트 패킷은 그래픽 파이프라인(120)의 상태를 업데이트하는 상수 또는 상수의 집합이다. 상태 업데이트 패킷은, 예를 들어 드로우 커맨드의 실행 중에 블렌딩될 컬러를 업데이트할 수 있다. 일부 실시형태에서, 상태 업데이트 패킷은 GPU(104)의 다중 컨텍스트 레지스터를 프로그래밍하는 세트 컨텍스트 패킷을 포함한다. 세트 컨텍스트 패킷은 패킷의 상태를 프로그래밍하는 데 필요한 모든 데이터를 포함한다. 다른 실시형태에서, 상태 업데이트 패킷은 또한 상태가 GPU(104)의 컨텍스트 레지스터에 기록되기 전에 메모리로부터 컨텍스트 정보를 폐기하는 것을 필요로 하는 로드 컨텍스트 패킷을 포함한다. 드로우 호출 패킷은 그래픽 파이프라인(120)이 디스플레이를 위해 출력될 데이터에 대한 프로세스를 실행하게 하는 커맨드이다.

[0026] 드로우 호출의 실행은 이전 드로우 호출 이후에 검색된 모든 상태 업데이트에 의존한다. 예를 들어, 도 2는 커맨드 스트림에 포함되는 6개의 커맨드 - (1) 조건부 실행 패킷(202), (2) 제1 상태 업데이트 패킷(204), (3) 제2 상태 업데이트 패킷(206), (4) 제3 상태 업데이트 패킷(208), (5) 제1 드로우 호출 패킷(210), 및 (6) 제2 드로우 호출 패킷(212) -를 도시한다. 드로우 호출 패킷(210)은 제1, 제2 및 제3 상태 업데이트 패킷(204-208)에 의존하는 데, 이는 그들이 다음 드로우 커맨드(즉, 드로우 호출 패킷(210)) 이전에 조건부 실행 패킷(202)에 의해 캡슐화되는 상태 업데이트이기 때문이다. 본 명세서에서 사용되는 바와 같이, 용어 "캡슐화된(encapsulated)"은 부여된 해시 식별자를 정의하는 데 사용되는 패킷의 수를 지칭한다. 따라서, 용어 "캡슐화된 상태"는 부여된 해시 식별자가 컨텍스트 바운스/전환에 이용 가능한지(예를 들어, GPU(104)의 레지스터에 저장되는지)에 따라 조건부로 실행되거나 폐기될 수 있는 컨텍스트 상태 패킷의 수를 지칭한다. 다시 말해, 캡슐화된 상태는 부여된 해시 식별자를 정의하는 데 사용되는 컨텍스트 상태 패킷의 집합이다.

[0027] 일 실시형태에서, GPU(104)에 의해 실행되는 동작의 출력은 현재 컨텍스트 상태(214)에 의존한다. 다양한 실시형태에서, 현재 컨텍스트 상태(214)는 상태 레지스터에 저장되는 텍스처 핸들러, 셰이더 상수, 변환 행렬 등과 같은 다양한 컨텍스트 특정 상수의 값에 기초한다. 도 2에 도시된 바와 같이, 현재 컨텍스트 상태(214)는 임의의 수 및 유형(예를 들어, 범용 레지스터)의 상태 레지스터 및 명령어 포인터를 나타내는 레지스터(216A 내지 216N)를 포함한다. 현재 컨텍스트 상태(214)는 또한 GPU(104)의 현재 아키텍처 상태를 정의하는 다른 변수 및 다른 값을 포함할 수 있음에 유의해야 한다. 또한, 처리 유닛의 "컨텍스트 상태"는 처리 유닛의 "컨텍스트" 또는 "상태"라고도 지칭될 수 있음에 유의해야 한다. 다양한 실시형태에서, 처리 유닛은 GPU(예를 들어, 도 1 및 도 2의 GPU(104))이다. 다른 실시형태에서, 처리 유닛은 다른 유형의 처리 유닛을 포함한다. 용어 "처리 유닛"

및 "프로세서"는 본 명세서에서 호환 가능하게 이용됨에 유의해야 한다.

- [0028] 동작 시, 커맨드 프로세서(122)는 조건부 실행 패킷(202)을 수신한다. 조건부 실행 패킷(202)은 다음 드로우 커맨드(예를 들어, 드로우 호출 패킷(210)) 이전에, 다가오는 상태 업데이트 패킷(204-208)과 관련된 고유 해시 식별자를 커맨드 프로세서(122)에 알린다. 커맨드 프로세서(122)는 조건부 실행 패킷(202)에 의해 제공되는 고유 해시 식별자가 GPU(104)에서 활성(이에 따라 해시 테이블(130)에 저장)되는지 여부를 결정하기 위해 해시 테이블(130)을 검색하도록 컨텍스트 매니저(128)에 질의한다.
- [0029] 도 2에 도시된 바와 같이, 조건부 실행 패킷(202)은 XYZ의 고유 해시 식별자를 제공한다. 커맨드 프로세서(122)는 컨텍스트 매니저(128)에 질의하고 해시 식별자 XYZ가 해시 테이블(130)에 존재하지 않는다고 결정한다. 따라서, 그 질의 미스에 기초하여, 컨텍스트 매니저(128)는 (해시 테이블(130)에서 추적된) 하드웨어의 기존 해시 식별자 중 하나를 회수하고 새로운 컨텍스트를 할당한다. 다양한 실시형태에서, 컨텍스트 매니저(128)는 해시 테이블(130)에서 가장 최근에 사용된(LRU) 비활성 해시 식별자를 회수한다. 회수된 해시 식별자는 해당의 새로운 컨텍스트에 대한 질의에 제공된 해시 식별자(예를 들어, 이 설명에서는 해시 식별자 XYZ)로 대체된다. 또한, 해시 테이블(130)에서 해시 식별자 XYZ에 대한 일치가 없다는 결정에 기초하여, 컨텍스트 매니저(128)는 질의 미스백(query miss back)을 커맨드 프로세서(122)에 표시하고 새로운 컨텍스트를 리턴한다. 그 후, 커맨드 프로세서(122)는 상태 업데이트 패킷(204-208)을 실행함으로써 캡슐화된 상태를 처리한다. 다시 말해, 질의 미스는 GPU(104)가 컨텍스트 롤링을 수행하여 새로운 컨텍스트를 할당하고, 다가오는 상태 업데이트 패킷(204-208)을 실행하고, 새로운 해시 식별자(즉, 해시 식별자 XYZ)로 다음 상태를 설정하게 한다.
- [0030] 도 3은 일부 실시형태에 따른 캡슐화된 상태의 트레이닝을 도시하는 블록도이다. 그러나, 다른 실시형태에서는, 도 2를 계속 참조하면, 커맨드 버퍼(118)에서의 커맨드 스트림(300)은 5개의 커맨드 - (1) 조건부 실행 패킷(302), (2) 제1 상태 업데이트 패킷(304), (3) 제2 상태 업데이트 패킷(306), (4) 제3 상태 업데이트 패킷(308), 및 (5) 드로우 호출 패킷(310) -를 포함한다. 드로우 호출 패킷(310)은 제1, 제2 및 제3 상태 업데이트 패킷(304-308)에 의존하는 데, 이는 그들이 다음 드로우 커맨드(즉, 드로우 호출 패킷(310)) 이전에 조건부 실행 패킷(302)에 의해 캡슐화되는 상태 업데이트이기 때문이다. 제1, 제2 및 제3 상태 업데이트 패킷(304-308)은 캡슐화된 상태(312)라고 총칭된다. 이 실시형태에서, 캡슐화된 상태(312)는 부여된 해시 식별자가 컨텍스트 바운스/전환에 이용가능한지(예를 들어, GPU(104)의 레지스터에 저장되는지)에 따라 조건부로 실행되거나 폐기될 수 있는 (즉, 조건부 실행 패킷(302)에 의해 정의된) 3개의 컨텍스트 상태 패킷을 포함한다.
- [0031] 도 3에 도시된 바와 같이, 조건부 실행 패킷(302)은 ABC의 고유 해시 식별자를 제공한다. 다시 도 2를 참조하면, 커맨드 프로세서(122)는 컨텍스트 매니저(128)에 질의하고 해시 식별자 ABC가 해시 테이블(130)에 존재한다고 결정한다. 따라서, 해당 질의 히트(query hit)에 기초하여, 컨텍스트 매니저(128)는 질의된 해시 식별자(즉, 해시 식별자 ABC)와 관련된 컨텍스트를 리턴한다. 그 후, 커맨드 프로세서(122)는 캡슐화된 상태(312)를 트레이닝하고(예를 들어, 상태 업데이트 패킷(304-308)을 스킵하고 폐기함) 기존 컨텍스트 해시 식별자를 사용하여 드로우 호출 패킷(310)의 실행을 위한 질의 히트와 일치하는 컨텍스트를 할당한다. 원하는 해시 식별자가 사용 가능하다고 결정하고, 해당 해시 식별자에 대한 컨텍스트로 전환하고, 해당 컨텍스트를 재정의했을 임의의 커맨드 패킷을 폐기함으로써, GPU(104)는 사용 가능한 컨텍스트의 부족으로부터 프로세서를 강제로 스тол하는 대신에 요청된 해시 식별자가 GPU 하드웨어에서 현재 사용 가능함을 인식하고 GPU 하드웨어로 전환함으로써 연산 효율을 개선한다.
- [0032] 도 4는 일부 실시형태에 따른 그래픽 컨텍스트 바운싱의 방법(400)을 도시하는 흐름도이다. 블록(402)에서, 그래픽 처리 유닛(GPU)의 커맨드 프로세서는 캡슐화된 상태에 대응하는 해시 식별자를 제공하는 조건부 실행 패킷을 수신한다. 예를 들어, 도 2를 참조하면, 커맨드 프로세서(122)는 커맨드 버퍼(118)로부터 조건부 실행 패킷(202)을 수신한다. 조건부 실행 패킷(202)은 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷(예를 들어, 도 2의 상태 업데이트 패킷(204-208))에 대응하는 캡슐화된 상태를 정의한다. 다양한 실시형태에서, 조건부 실행 패킷(202)은 캡슐화된 상태와 관련된 고유 해시 식별자(예를 들어, 도 2의 조건부 실행 패킷(202)에서의 해시 식별자 XYZ 또는 도 3의 조건부 실행 패킷(302)에서의 해시 식별자 ABC)를 정의한다. 또한, 일부 실시형태에서, 조건부 실행 패킷(202)은 또한 캡슐화된 상태에 대한 컨텍스트 상태 패킷의 그룹의 크기에 대응하는 dword 데이터를 포함한다.
- [0033] 블록(404)에서, 커맨드 프로세서는 캡슐화된 상태의 해시 식별자가 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 해시 식별자 중 하나와 일치하는지 여부를 결정한다. 예를 들어, 도 2를 참조하면, 커맨드 프로세서(122)는 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 해시 식별자를 저장하는 해시 테이블(130)을 검색하기 위해

컨텍스트 매니저(128)에 질의한다.

- [0034] 커맨드 프로세서가 캡슐화된 상태의 해시 식별자가 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 해시 식별자 중 어느 것과도 일치하지 않는다고 결정하는 경우, 방법(400)은 블록(406)으로 진행한다. 그러나, 커맨드 프로세서가 캡슐화된 상태의 해시 식별자가 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 해시 식별자 중 하나와 일치한다고 결정하는 경우, 방법(400)은 블록(408)으로 진행한다.
- [0035] 블록(406)에서, 그리고 도 2를 계속 참조하면, 컨텍스트 매니저(128)는 (해시 테이블(130)에서 추적된) 하드웨어의 기존의 비활성 해시 식별자 중 하나를 회수하고, 해시 식별자 XYZ가 해시 테이블(130)에 존재하지 않는다고 결정하는 컨텍스트 매니저(128)로부터의 질의 미스에 기초하여 새로운 컨텍스트를 할당한다. 일부 실시형태에서, 컨텍스트 매니저(128)는 해시 테이블(130)에서 가장 최근에 사용된(LRU) 해시 식별자를 회수한다. 블록(410)에서, 컨텍스트 매니저(128)는 회수된 해시 식별자를 해당의 새로운 컨텍스트에 대한 질의에 제공된 해시 식별자(예를 들어, 이 설명에서는 해시 식별자 XYZ)로 대체한다. 또한, 다양한 실시형태에서, 컨텍스트 매니저(128)는 질의 미스백을 커맨드 프로세서(122)에 표시하고 새로운 컨텍스트를 리턴한다.
- [0036] 블록(412)에서, 커맨드 프로세서(122)는 상태 업데이트 패킷(204-208)을 실행함으로써 캡슐화된 상태를 처리한다. 다시 말해, 질의 미스는 GPU(104)가 컨텍스트 롤링을 수행하여 새로운 컨텍스트 상태 세트를 할당하고, 다가오는 상태 업데이트 패킷(204-208)을 실행하고, 새로운 해시 식별자(즉, 해시 식별자 XYZ)로 다음 상태를 설정하게 한다. 이어서, 블록(414)에서, 커맨드 프로세서(122)는 캡슐화된 상태 다음의 커맨드 패킷(예를 들어, 도 2의 상태 드로우 호출 패킷(210))을 실행한다.
- [0037] 이제 블록(408)으로 돌아가서 도 3을 참조하면, 컨텍스트 매니저(128)는 해시 식별자 ABC가 해시 테이블(130)에 존재한다고 결정하는 컨텍스트 매니저(128)로부터의 질의 히트에 기초하여 질의된 해시 식별자(예를 들어, 도 3의 조건부 실행 패킷(302)에 의해 제공되는 해시 식별자 ABC)와 관련된 컨텍스트를 커맨드 프로세서(122)로 리턴한다. 블록(416)에서, 커맨드 프로세서(122)는 캡슐화된 상태(312)를 트레이닝하고(예를 들어, 상태 업데이트 패킷(304-308)을 스킵하고 폐기함) 기존 컨텍스트 해시 식별자를 사용하여 드로우 호출 패킷(310)의 실행을 위한 질의 히트와 일치하는 컨텍스트를 할당한다. 이어서, 블록(418)에서, 커맨드 프로세서(122)는 바이패스된 캡슐화된 상태(312) 다음의 커맨드 패킷을 실행한다. 원하는 해시 식별자가 사용 가능하다고 결정하고, 해당 해시 식별자에 대한 컨텍스트로 전환하고, 해당 컨텍스트를 재정의했을 임의의 커맨드 패킷을 폐기함으로써, GPU(104)는 사용 가능한 컨텍스트의 부족으로부터 프로세서를 강제로 스тол하는 대신에 요청된 해시 식별자가 GPU 하드웨어에서 현재 사용 가능함을 인식하고 GPU 하드웨어로 전환함으로써 연산 효율을 개선한다.
- [0038] 다양한 실시형태에서, 컨텍스트 수행 이벤트는 컨텍스트가 롤링될 때에 그래픽 파이프라인 아래로 전송된다. 컨텍스트 매니저는 미해결 컨텍스트 수행 이벤트가 있음을 나타내는 카운터를 증분시킨다. 또한, 다양한 실시형태에서, 컨텍스트는 그래픽 파이프라인에 계류 중인 컨텍스트 수행 이벤트가 없을 때에만 재활용이 허용된다.
- [0039] 제한이 아니라 예시적인 목적을 위해, 다양한 실시형태에 따른 그래픽 컨텍스트 바운싱 시스템은 GPU의 그래픽 파이프라인의 컨텍스트 상태에 기초하여 드로우 커맨드의 스트림을 실행하는 GPU와 관련하여 본 명세서에 설명되어 있다. 그러나, 이러한 그래픽 컨텍스트 바운싱 시스템은 각각의 시스템의 상태에 기초하여 커맨드의 스트림을 실행하는 다른 유형의 ASIC 또는 시스템에 적용 가능하다는 것이 이해되어야 한다. 본 명세서에 포함된 설명에 기초하여, 관련 기술(들)의 숙련자는 이러한 다른 유형의 시스템에서 본 발명의 실시형태를 구현하는 방법을 이해할 것이다. 또한, 당업자는 다수의 세부사항(예를 들어, 프로세서 구성의 특정 수 및 배치, 마이크로아키텍처 세부사항, 논리 분할/통합 세부사항, 동작 시퀀스, 유형, 시스템 구성요소의 상호관계, GPU에 저장된 고유 컨텍스트 상태 세트의 수 등)이 본 명세서에 제시됨을 인식할 것이다. 그러나, 본 명세서에 설명된 그래픽 컨텍스트 바운싱은 본 개시의 범위를 벗어나지 않고서 임의의 수의 개별 레지스터, 레지스터 뱅크, 및/또는 컨텍스트 상태의 고유 세트에 대해 수행될 수 있음이 이해된다.
- [0040] 본 명세서에 개시된 바와 같이, 일부 실시형태에서 방법은, 그래픽 처리 유닛(GPU)의 커맨드 프로세서에서, 캡슐화된 상태에 대응하는 식별자를 제공하는 조건부 실행 패킷을 수신하는 단계 - 캡슐화된 상태는 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태를 포함함 -; 및 캡슐화된 상태의 식별자가 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여, 캡슐화된 상태 다음의 커맨드 패킷을 실행하는 단계를 포함한다. 일 양태에서, 캡슐화된 상태의 식별자가 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것은 GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자를 저장하는 식별자 테이블을 검색하도록 컨텍스트 매니저에 질의하는 것을 포함한다. 다른 양태에서, 방법은, 활성 컨텍스트 상태의 복수의 식별자 중 하나와 일치하는 식별자를 결정하는 것에 응답하여, 캡슐화된 상태를 트레이닝

하는 단계; 및 활성 컨텍스트 상태의 복수의 식별자 중 일치하는 하나를 캡슐화된 상태 다음의 커맨드 패킷에 할당하는 단계를 포함한다.

[0041] 일 양태에서, 방법은, 식별자가 활성 컨텍스트 상태의 복수의 식별자 중 어느 것과도 일치하지 않는다고 결정하는 것에 응답하여, 새로운 컨텍스트 상태 세트를 할당하는 단계; 및 커맨드 패킷을 실행하기 전에 캡슐화된 상태의 하나 이상의 컨텍스트 상태 패킷을 실행하는 단계를 포함한다. 다른 양태에서, 방법은, 식별자 테이블로부터, GPU에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자 중 하나를 회수하는 단계; 및 식별자 테이블에서, 새로운 컨텍스트 상태 세트에 대한 캡슐화된 상태에 대응하는 식별자를 저장하는 단계를 포함한다. 또 다른 양태에서, 커맨드 패킷은 드로우 호출을 포함한다. 또 다른 양태에서, 커맨드 패킷은 상태 업데이트 패킷의 수에 대한 캡슐화된 상태의 크기를 지정한다.

[0042] 일부 실시형태에서, 프로세서는, 복수의 상태 레지스터; 및 커맨드 프로세서를 포함하고, 커맨드 프로세서는, 캡슐화된 상태에 대응하는 식별자를 제공하는 조건부 실행 패킷을 수신하는 것 - 캡슐화된 상태는 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태를 포함함 -; 및 캡슐화된 상태의 식별자가 프로세서에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여, 캡슐화된 상태 다음의 커맨드 패킷을 실행하는 것을 수행하도록 구성된다. 일 양태에서, 프로세서는 프로세서에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자를 저장하는 식별자 테이블을 포함하는 컨텍스트 매니저를 포함한다. 다른 양태에서, 프로세서는 제공된 식별자에 대해 식별자 테이블을 검색하기 위해 컨텍스트 매니저에 질의하도록 구성된다.

[0043] 일 양태에서, 프로세서는 추가로, 활성 컨텍스트 상태의 복수의 식별자 중 하나와 일치하는 식별자를 결정하는 것에 응답하여, 캡슐화된 상태를 트레이닝하는 것; 및 활성 컨텍스트 상태의 복수의 식별자 중 일치하는 하나를 캡슐화된 상태 다음의 커맨드 패킷에 할당하는 것을 수행하도록 구성된다. 다른 양태에서, 프로세서는 추가로, 식별자가 활성 컨텍스트 상태의 복수의 식별자 중 어느 것과도 일치하지 않는다고 결정하는 것에 응답하여, 새로운 컨텍스트 상태 세트를 할당하는 것; 및 커맨드 패킷을 실행하기 전에 캡슐화된 상태의 하나 이상의 컨텍스트 상태 패킷을 실행하는 것을 수행하도록 구성된다. 또 다른 양태에서, 프로세서는 추가로, 식별자 테이블로부터, 프로세서에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자 중 하나를 회수하는 것; 및 식별자 테이블에서, 새로운 컨텍스트 상태 세트에 대한 캡슐화된 상태에 대응하는 식별자를 저장하는 것을 수행하도록 구성된다. 또 다른 양태에서, 커맨드 패킷은 상태 업데이트 패킷의 수에 대한 캡슐화된 상태의 크기를 지정한다.

[0044] 일부 실시형태에서, 비밀시적 컴퓨터 관독 가능 매체는 실행 가능한 명령어 세트를 구현하며, 실행 가능한 명령어 세트는 적어도 하나의 프로세서를 조작하여, 프로세서의 커맨드 프로세서에서, 캡슐화된 상태에 대응하는 식별자를 제공하는 조건부 실행 패킷을 수신하는 것 - 캡슐화된 상태는 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태를 포함함 -; 및 캡슐화된 상태의 식별자가 프로세서에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여, 캡슐화된 상태 다음의 커맨드 패킷을 실행하는 것이다. 일 양태에서, 실행 가능한 명령어 세트는 적어도 하나의 프로세서를 추가로 조작하여, 프로세서에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자를 저장하는 식별자 테이블을 검색하기 위해 컨텍스트 매니저에 질의하는 것이다. 다른 양태에서, 실행 가능한 명령어 세트는 적어도 하나의 프로세서를 추가로 조작하여, 활성 컨텍스트 상태의 복수의 식별자 중 하나와 일치하는 식별자를 결정하는 것에 응답하여, 캡슐화된 상태를 트레이닝하는 것; 및 활성 컨텍스트 상태의 복수의 식별자 중 일치하는 하나를 캡슐화된 상태 다음의 커맨드 패킷에 할당하는 것이다.

[0045] 일 양태에서, 실행 가능한 명령어 세트는 적어도 하나의 프로세서를 추가로 조작하여, 식별자가 활성 컨텍스트 상태의 복수의 식별자 중 어느 것과도 일치하지 않는다고 결정하는 것에 응답하여, 새로운 컨텍스트 상태 세트를 할당하는 것; 및 커맨드 패킷을 실행하기 전에 캡슐화된 상태의 하나 이상의 컨텍스트 상태 패킷을 실행하는 것이다. 다른 양태에서, 실행 가능한 명령어 세트는 추가로 적어도 하나의 프로세서를 조작하여, 식별자 테이블로부터, 프로세서에 현재 저장된 활성 컨텍스트 상태의 복수의 식별자 중 하나를 회수하는 것; 및 식별자 테이블에서, 새로운 컨텍스트 상태 세트에 대한 캡슐화된 상태에 대응하는 식별자를 저장하는 것이다. 또 다른 양태에서, 커맨드 패킷은 상태 업데이트 패킷의 수에 대한 캡슐화된 상태의 크기를 지정한다.

[0046] 컴퓨터 관독 가능 저장 매체는 컴퓨터 시스템에 명령어 및/또는 데이터를 제공하기 위해 사용하는 동안 컴퓨터 시스템에 의해 액세스 가능한 임의의 비밀시적 저장 매체 또는 비밀시적 저장 매체의 조합을 포함할 수 있다. 이러한 저장 매체는 광학 매체(예를 들어, 콤팩트 디스크(CD), 디지털 다목적 디스크(DVD), 블루레이 디스크),

자기 매체(예를 들어, 플로피 디스크, 자기 테이프 또는 자기 하드 드라이브), 휘발성 메모리(예를 들어, 랜덤 액세스 메모리(RAM) 또는 캐시), 비휘발성 메모리(예를 들어, 판독 전용 메모리(ROM) 또는 플래시 메모리), 또는 마이크로전자기계 시스템(MEMS) 기반 저장 매체를 포함할 수 있지만, 이에 한정되지 않는다. 컴퓨터 판독 가능 저장 매체는 컴퓨팅 시스템(예를 들어, 시스템 RAM 또는 ROM)에 내장되거나, 컴퓨팅 시스템(예를 들어, 자기 하드 드라이브)에 고정식으로 부착되거나, 컴퓨팅 시스템(예를 들어, 광학 디스크 또는 유니버설 직렬 버스(USB) 기반 플래시 메모리)에 착탈식으로 부착되거나, 또는 유선 또는 무선 네트워크를 통해 컴퓨터 시스템(예를 들어, 네트워크 액세스 가능 스토리지(NAS))에 결합될 수 있다.

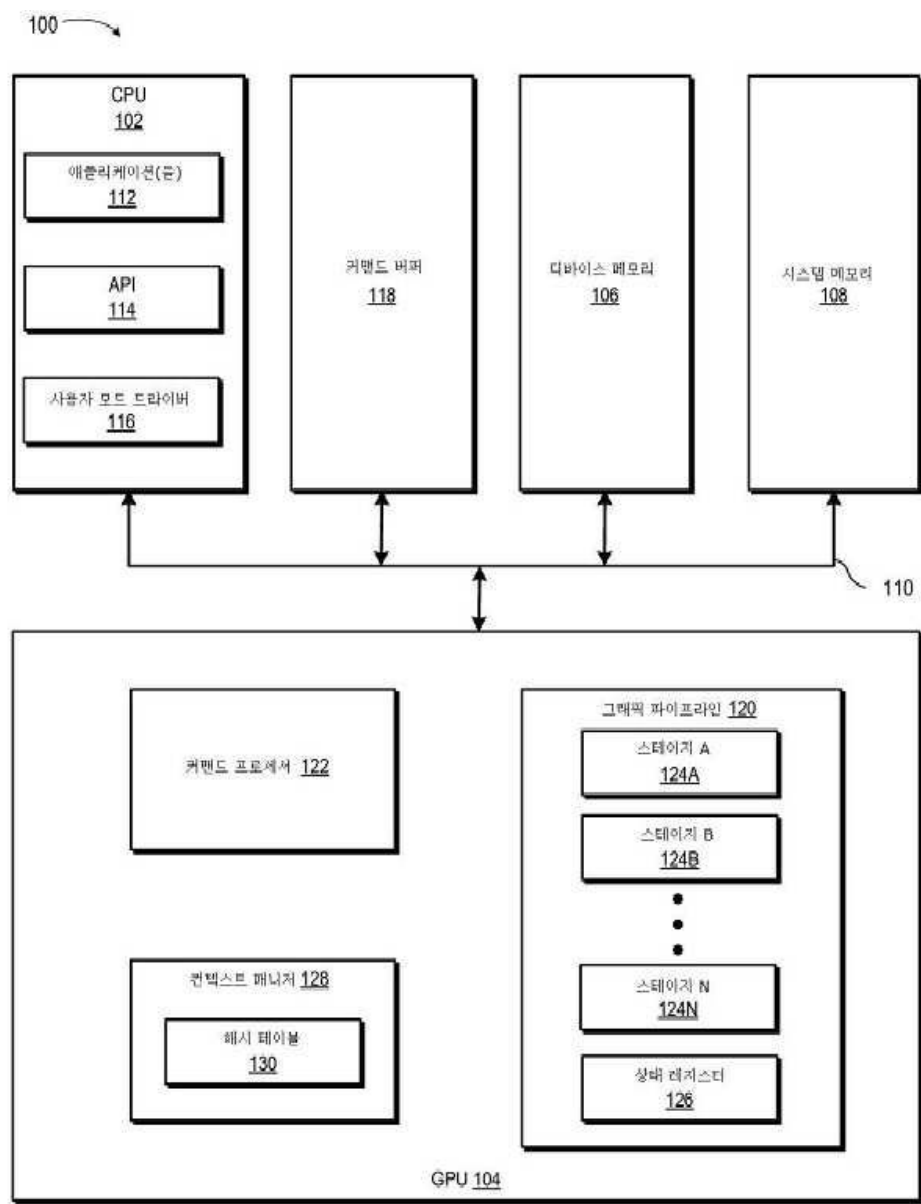
[0047] 일부 실시형태에서, 전술한 기술의 특정 양태는 소프트웨어를 실행하는 처리 시스템의 하나 이상의 프로세서에 의해 구현될 수 있다. 소프트웨어는 비일시적 컴퓨터 판독 가능 저장 매체에 저장되거나 달리 유형적으로 구현되는 하나 이상의 실행 가능한 명령어 세트를 포함한다. 소프트웨어는 하나 이상의 프로세서에 의해 실행될 때 하나 이상의 프로세서를 조작하여 전술한 기술의 하나 이상의 양태를 수행하는 명령어 및 특정 데이터를 포함할 수 있다. 비일시적 컴퓨터 판독 가능 저장 매체는, 예를 들어 자기 또는 광 디스크 저장 디바이스, 플래시 메모리, 캐시, 랜덤 액세스 메모리(RAM) 또는 다른 비휘발성 메모리 디바이스 또는 디바이스들과 같은 솔리드 스테이트 저장 디바이스 등을 포함할 수 있다. 비일시적 컴퓨터 판독 가능 저장 매체에 저장된 실행 가능한 명령어는 소스 코드, 어셈블리 언어 코드, 목적 코드, 또는 하나 이상의 프로세서에 의해 해석되거나 달리 실행 가능한 다른 명령어 포맷일 수 있다.

[0048] 일반적인 설명에서 전술한 모든 활동 또는 요소가 필요한 것은 아니며, 특정 활동 또는 디바이스의 일부가 필요하지 않을 수도 있고, 설명된 것에 추가로 하나 이상의 추가 활동 또는 요소가 수행 또는 포함될 수 있음에 유의해야 한다. 또한, 활동이 열거되는 순서가 반드시 그들이 수행되는 순서는 아니다. 또한, 구체적인 실시형태를 참조하여 개념이 설명되었다. 그러나, 당업자는 아래의 청구범위에 제시된 본 개시의 범위를 벗어나지 않고 다양한 수정 및 변경이 이루어질 수 있음을 이해한다. 따라서, 명세서 및 도면은 제한적인 의미가 아니라 예시적인 것으로 간주되어야 하며, 이러한 모든 수정은 본 개시의 범위 내에 포함되는 것으로 의도되어 있다.

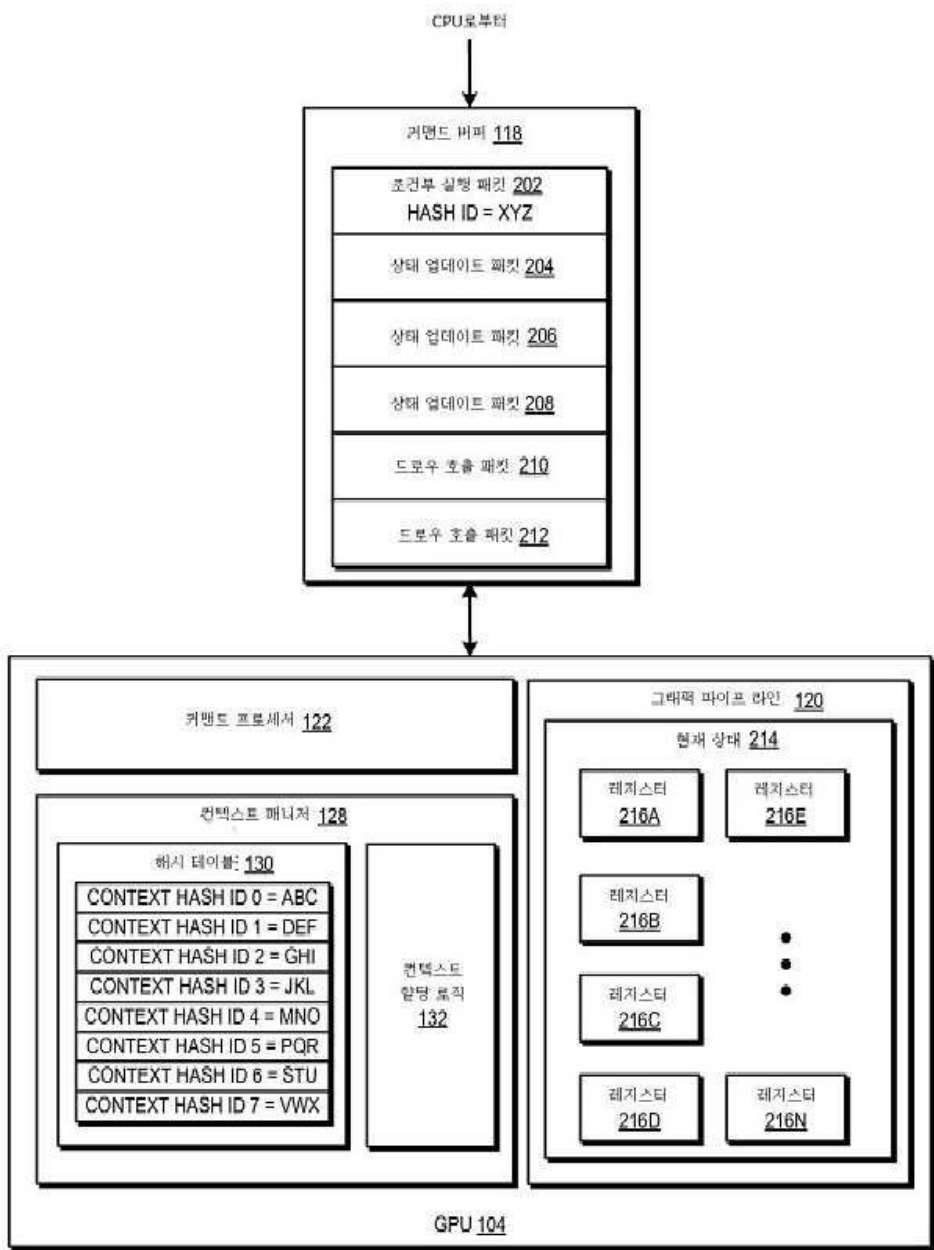
[0049] 이점, 다른 장점, 및 문제에 대한 해결책이 특정 실시형태와 관련하여 상기에 기술되어 있다. 그러나, 상기 이점, 장점, 문제에 대한 해결책, 및 임의의 이점, 장점, 또는 문제에 대한 해결책을 발생시킬 수 있거나 더욱 명확하게 할 수 있는 임의의 특징(들)은 임의의 또는 모든 청구항의 중요한, 필요한 또는 필수 특징인 것으로 해석되지 않아야 한다. 또한, 상기에 개시된 특정 실시형태는 단지 예시적인 것이며, 개시된 주제는 본 명세서의 교시의 이점을 갖는 당업자에게 명백한 상이하면서도 동등한 방식으로 수정 및 실시될 수 있다. 아래의 청구범위에 기재된 것 외에는 본 명세서에 나타난 구성 또는 설계의 세부사항의 제한이 의도되어 있지 않다. 따라서, 상기에 개시된 특정 실시형태가 변경되거나 수정될 수 있고 이러한 모든 변형은 개시된 주제의 범위 내에서 고려된다는 것이 명백하다. 따라서, 본 명세서에서 보호받고자 하는 사항은 아래의 청구범위에 제시된 바와 같다.

도면

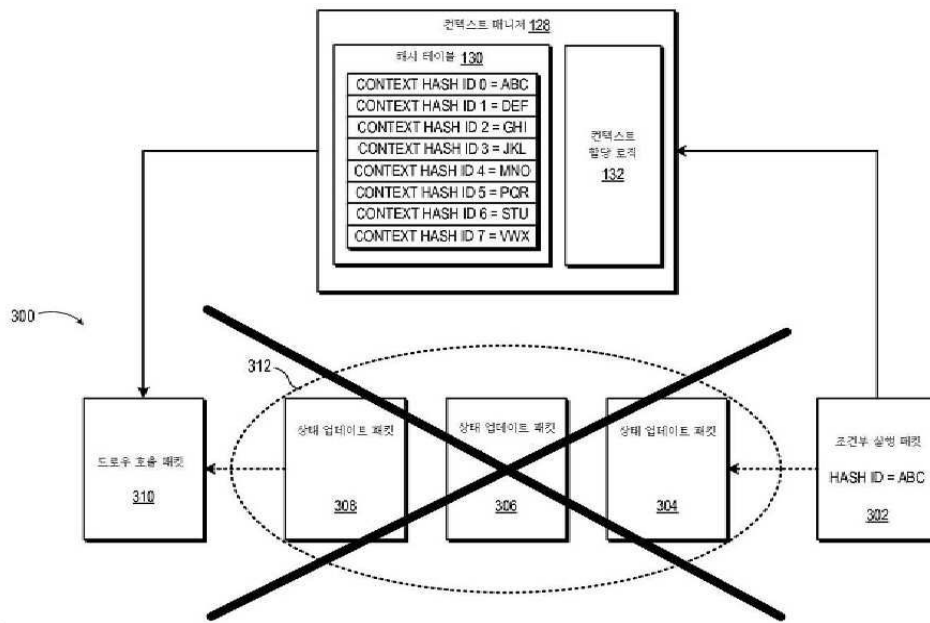
도면1



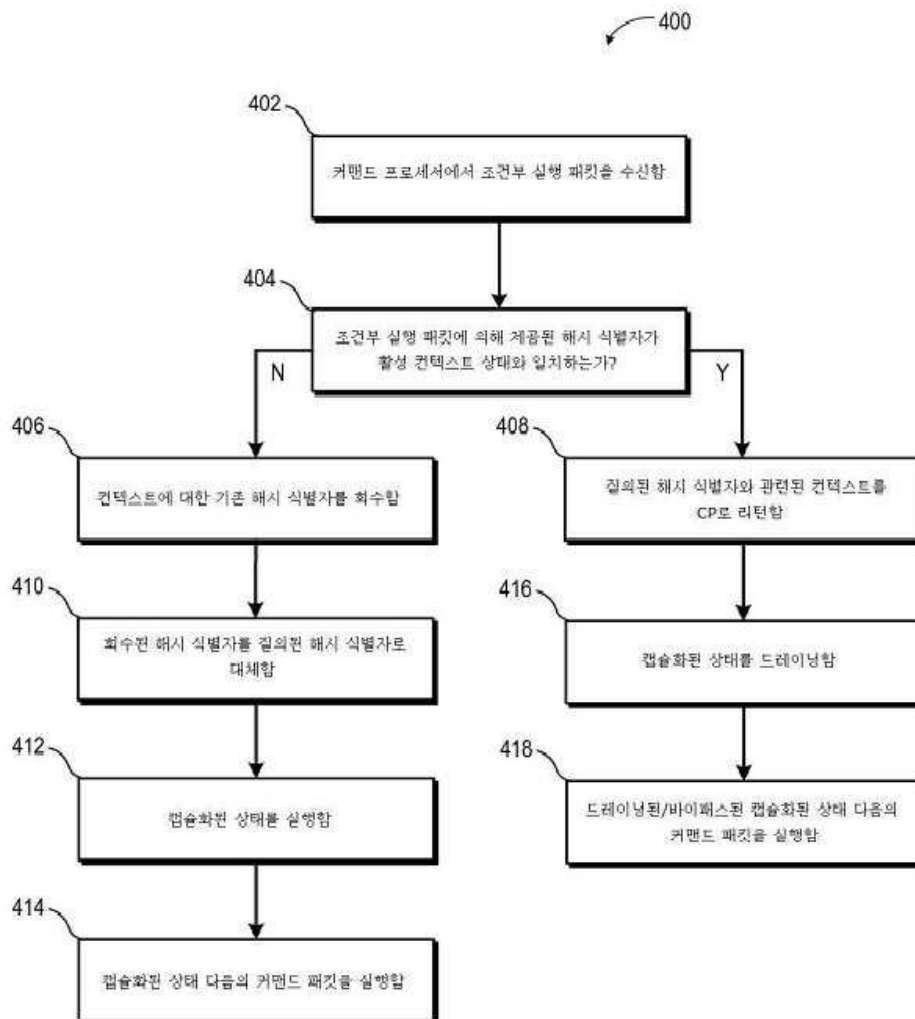
도면2



도면3



도면4



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 17

【변경전】

실행가능한 명령어 세트를 구현하는 비밀시적 컴퓨터 판독가능 매체로서, 상기 실행가능한 명령어 세트는 적어도 하나의 프로세서를 조작하여,

상기 프로세서의 커맨드 프로세서에서, 캡슐화된 상태에 대응하는 식별자를 제공하는 조건부 실행 패킷[202]을 수신하고 - 상기 캡슐화된 상태는 상기 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷을 포함함 -;

및

상기 캡슐화된 상태의 상기 식별자가 상기 프로세서에 현재 저장된 활성 컨텍스트 상태의 저장된 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여, 상기 캡슐화된 상태 다음의 커맨드 패킷을 실행하게 하는, 비밀시적 컴퓨터 판독가능 매체.

【변경후】

실행가능한 명령어 세트를 저장하는 비밀시적 컴퓨터 판독가능 매체로서, 상기 실행가능한 명령어 세트는 적어도 하나의 프로세서를 조작하여,

상기 프로세서의 커맨드 프로세서에서, 캡슐화된 상태에 대응하는 식별자를 제공하는 조건부 실행 패킷[202]을 수신하고 - 상기 캡슐화된 상태는 상기 조건부 실행 패킷 다음의 하나 이상의 컨텍스트 상태 패킷을 포함함 -;

및

상기 캡슐화된 상태의 상기 식별자가 상기 프로세서에 현재 저장된 활성 컨텍스트 상태의 저장된 복수의 식별자 중 하나와 일치하는지 여부를 결정하는 것에 적어도 부분적으로 기초하여, 상기 캡슐화된 상태 다음의 커맨드 패킷을 실행하게 하는, 비밀시적 컴퓨터 판독가능 매체.