(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년12월23일
(11) 등록번호 10-2480787
(24) 등록일자 2022년12월20일

(51) 국제특허분류(Int. Cl.)
G06F 12/0811 (2016.01)
(52) CPC특허분류
G06F 12/0811 (2013.01)
(21) 출원번호 10-2019-7016319
(22) 출원일자(국제) 2017년11월21일
심사청구일자 2020년11월11일
(85) 번역문제출일자 2019년06월05일
(65) 공개번호 10-2019-0087450
(43) 공개일자 2019년07월24일
(86) 국제출원번호 PCT/US2017/062889
(87) 국제공개번호 WO 2018/111515
국제공개일자 2018년06월21일
(30) 우선권주장
15/377,998 2016년12월13일 미국(US)
(56) 선행기술조사문헌
US06237067 B1*
US20130262776 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 95054 산타 클라라 어거스틴 드라이브 2485
(72) 발명자
슈나이더 다니엘
미국 캘리포니아 95054 산타 클라라 어거스틴 드라이브 2485 어드밴스드 마이크로 디바이시즈 인코포레이티드 씨/오
고드라트 파타네흐
미국매사추세츠 01719 박스버러 플로어 1, 2 & 3 센트럴 스트리트 90 어드밴스드 마이크로 디바이시즈 인코포레이티드 씨/오
(74) 대리인
박장원

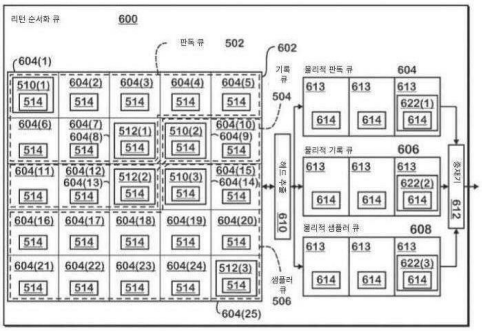
전체 청구항 수 : 총 20 항 심사관 : 안지현

(54) 발명의 명칭 **순서에 관계 없는 캐시 리턴**

(57) 요 약

순서에 관계 없는 캐시 액세스 리턴을 허용하기 위한 기술이 개시되어 있다. 리턴 순서화 큐는 여러 캐시 액세스 유형들 각각에 대해 존재하며 액세스들이 이루어진 순서대로 중요한 캐시 액세스들을 저장한다. 특정 유형에 대한 캐시 액세스 요청이 해당 유형에 대한 리턴 순서화 큐의 헤드에 있고 해당 액세스를 행한 웨이브프론트로의 리턴을 위해 캐시 액세스를 이용 가능한 경우, 캐시 시스템은 캐시 액세스를 웨이브프론트로 리턴한다. 따라서, 캐시 액세스들은 상이한 유형들의 캐시 액세스들과 관련하여 순서에 관계 없이 리턴될 수 있다. 순서에 관계 없는 리턴을 허용하면, 상대적으로 높은-레이턴시 액세스 유형(예를 들어, 텍스처 샘플러 동작) 후에 상대적으로 낮은-레이턴시 액세스 유형(예를 들어, 판독)이 발행되는 상황에서, 레이턴시를 개선하는데 도움이 된다.

대 표 도



명 세 서

청구범위

청구항 1

순서에 관계 없는 캐시 리턴(out-of-order cache return)을 수행하는 방법에 있어서,

복수의 리턴 순서화 큐(return ordering queue)들의 제1 리턴 순서화 큐의 헤드(head)에서의 제1 엔트리(entry)가 웨이브프론트(wavefront)로의 리턴을 위해 이용가능하다는 것을 결정하는 단계로서, 상기 제1 엔트리는 제1 캐시 액세스 요청에 대응되고,

상기 제1 리턴 순서화 큐는 제2 캐시 액세스 유형이 아닌 제1 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하고, 그리고

상기 복수의 리턴 순서화 큐들의 제2 리턴 순서화 큐는 상기 제1 캐시 액세스 유형이 아닌 제2 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하는, 상기 결정하는 단계; 및

상기 결정에 응답하여, 상기 제1 캐시 액세스 요청 이전의 상기 제2 리턴 순서화 큐의 엔트리들에 대응되는 캐시 액세스 요청들이 상기 웨이브프론트로의 리턴을 위해 이용가능하게 될 것을 기다리지 않고, 상기 제1 엔트리에 대응되는 캐시 리턴을 상기 웨이브프론트로 향하게 하는 단계를 포함하고,

상기 캐시 리턴은 완료된 대응하는 캐시 액세스 요청을 나타내는 데이터를 포함하는, 방법.

청구항 2

제1항에 있어서,

상기 제1 캐시 액세스 유형은 판독 유형(read type), 기록 유형(write type) 및 텍스처 샘플러 유형(texture sampler type) 중 하나를 포함하고;

상기 제2 캐시 액세스 유형은 상기 판독 유형, 상기 기록 유형 및 상기 텍스처 샘플러 유형 중 하나를 포함하고; 그리고

상기 제2 캐시 액세스 유형은 상기 제1 캐시 액세스 유형과 상이한, 방법.

청구항 3

제2항에 있어서,

상기 판독 유형은 메모리 시스템으로부터 데이터를 요청하고 리턴에서 데이터를 수신하는 액세스 유형을 포함하고;

상기 기록 유형은 상기 메모리 시스템에 데이터를 기록하고 리턴에서 확인 응답 신호(acknowledgment signal)를 수신하는 액세스 유형을 포함하고; 그리고

텍스처 샘플러 유형은 텍스처 좌표(texture coordinates)를 통해 텍스처 데이터를 요청하고 리턴에서 상기 텍스처 데이터를 수신하는 액세스 유형을 포함하는, 방법.

청구항 4

제3항에 있어서,

상기 텍스처 샘플 유형은 상기 텍스처 좌표를 하나 이상의 메모리 어드레스들로 변환하는 것, 상기 하나 이상의 메모리 어드레스들로부터 데이터를 인출하는 것, 상기 인출된 데이터를 압축 해제하는 것, 그리고 상기 인출된 데이터에 필터링을 적용하는 것 중 하나 이상을 요청하는 액세스 유형을 포함하는, 방법.

청구항 5

제1항에 있어서,

상기 복수의 리턴 순서화 큐들에 대한 모드를 선택하는 단계를 더 포함하고, 상기 모드는 상기 복수의 리턴 순서화 큐들에 어떤 리턴 순서화 큐들이 있는지를 정의하고 그리고 어떤 하나 이상의 캐시 액세스 유형들이 상기 복수의 리턴 순서화 큐들의 각각의 리턴 순서화 큐에 저장되고 순서화되는지를 정의하는, 방법.

청구항 6

제5항에 있어서,

상기 복수의 리턴 순서화 큐들은 모놀리식 메모리(monolithic memory) 내에 저장되는 가상 큐(virtual queue)들을 포함하고, 상기 가상 큐들은 상기 선택된 모드를 수용하도록 크기 조정 가능한, 방법.

청구항 7

제6항에 있어서,

각 가상 큐의 헤드로부터 대응하는 물리적 큐들의 헤드로 엔트리들을 복사하는 단계를 더 포함하고,

상기 제1 엔트리에 대응되는 상기 캐시 리턴을 상기 웨이브프론트로 향하게 하는 단계는:

상기 제1 리턴 순서화 큐에 대응되는 물리적 큐의 헤드로부터 엔트리를 제거하는 단계;

상기 물리적 큐의 다음으로 가장 오래된 엔트리가 상기 물리적 큐의 헤드에 있도록 수정하는 단계, 그리고

상기 제1 리턴 순서화 큐로부터 상기 제1 리턴 순서화 큐에 대응되는 상기 물리적 큐로 엔트리를 복사하는 단계를 포함하는, 방법.

청구항 8

제1항에 있어서,

상기 웨이브프론트의 캐시 액세스 유형-기반 배리어 명령을 실행하는 단계를 더 포함하는, 방법.

청구항 9

제8항에 있어서, 상기 캐시 액세스 유형-기반 배리어 명령을 실행하는 단계는,

특정 캐시 액세스 유형의 중요한(outstanding) 캐시 액세스가 완료될 때까지 웨이브프론트를 정지시키는 단계(stalling)를 포함하는, 방법.

청구항 10

순서에 관계 없는 캐시 리턴을 수행하기 위한 연산 유닛(compute unit)에 있어서,

웨이브프론트를 수행하도록 구성된 단일-명령-다중-데이터 유닛; 및

캐시 시스템(cache system)을 포함하고, 상기 캐시 시스템은:

제1 리턴 순서화 큐 및 제2 리턴 순서화 큐를 포함하는 복수의 리턴 순서화 큐들을 저장하고, 상기 제1 리턴 순서화 큐는 제2 캐시 액세스 유형이 아닌 제1 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하며, 상기 제2 리턴 순서화 큐는 상기 제1 캐시 액세스 유형이 아닌 제2 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하고;

상기 제1 리턴 순서화 큐의 헤드에서의 제1 엔트리가 웨이브프론트로의 리턴을 위해 이용가능하다는 것을 결정하고, 상기 제1 엔트리는 제1 캐시 액세스 요청에 대응되고; 그리고

상기 결정에 응답하여, 상기 제1 캐시 액세스 요청 이전의 상기 제2 리턴 순서화 큐의 엔트리들에 대응되는 캐시 액세스 요청들이 상기 웨이브프론트로의 리턴을 위해 이용가능하게 될 것을 기다리지 않고, 상기 제1 엔트리에 대응되는 캐시 리턴을 상기 웨이브프론트로 향하게 하도록 구성되며,

상기 캐시 리턴은 완료된 대응하는 캐시 액세스 요청을 나타내는 데이터를 포함하는, 연산 유닛.

청구항 11

제10항에 있어서,

상기 제1 캐시 액세스 유형은 판독 유형, 기록 유형 및 텍스처 샘플러 유형 중 하나를 포함하고;

상기 제2 캐시 액세스 유형은 상기 판독 유형, 상기 기록 유형 및 상기 텍스처 샘플러 유형 중 하나를 포함하고; 그리고

상기 제2 캐시 액세스 유형은 상기 제1 캐시 액세스 유형과 상이한, 연산 유닛.

청구항 12

제11항에 있어서,

상기 판독 유형은 메모리 시스템으로부터 데이터를 요청하고 리턴에서 데이터를 수신하는 액세스 유형을 포함하고;

상기 기록 유형은 상기 메모리 시스템에 데이터를 기록하고 리턴에서 확인 응답 신호를 수신하는 액세스 유형을 포함하고; 그리고

텍스처 샘플러 유형은 텍스처 좌표를 통해 텍스처 데이터를 요청하고 리턴에서 상기 텍스처 데이터를 수신하는 액세스 유형을 포함하는, 연산 유닛.

청구항 13

제12항에 있어서,

상기 텍스처 샘플 유형은 상기 텍스처 좌표를 하나 이상의 메모리 어드레스들로 변환하는 것, 상기 하나 이상의 메모리 어드레스들로부터 데이터를 인출하는 것, 상기 인출된 데이터를 압축 해제하는 것, 그리고 상기 인출된 데이터에 필터링을 적용하는 것 중 하나 이상을 요청하는 액세스 유형을 포함하는, 연산 유닛.

청구항 14

제10항에 있어서, 상기 캐시 시스템은:

상기 복수의 리턴 순서화 큐들에 대한 모드를 선택하도록 더 구성되며, 상기 모드는 상기 복수의 리턴 순서화 큐들에 어떤 리턴 순서화 큐들이 있는지를 정의하고 그리고 어떤 하나 이상의 캐시 액세스 유형들이 상기 복수의 리턴 순서화 큐들의 각각의 리턴 순서화 큐에 저장되고 순서화되는지를 정의하는, 연산 유닛.

청구항 15

제14항에 있어서,

상기 복수의 리턴 순서화 큐들은 모놀리식 메모리 내에 저장되는 가상 큐들을 포함하고, 상기 가상 큐들은 상기 선택된 모드를 수용하도록 크기 조정 가능한, 연산 유닛.

청구항 16

제15항에 있어서, 상기 캐시 시스템은:

각 가상 큐의 헤드로부터 대응되는 물리적 큐들의 헤드로 엔트리들을 복사하도록 더 구성되며,

상기 제1 엔트리에 대응되는 상기 캐시 리턴을 상기 웨이브프론트로 향하게 하는 것은:

상기 제1 리턴 순서화 큐에 대응되는 물리적 큐의 헤드로부터 엔트리를 제거하는 것;

상기 물리적 큐의 다음으로 가장 오래된 엔트리가 상기 물리적 큐의 헤드에 있도록 수정하는 것, 그리고

상기 제1 리턴 순서화 큐로부터 상기 제1 리턴 순서화 큐에 대응되는 상기 물리적 큐로 엔트리를 복사하는 것을 포함하는, 연산 유닛.

청구항 17

제10항에 있어서, 상기 웨이브프론트는,

캐시 액세스 유형-기반 배리어 명령을 실행하도록 구성된, 연산 유닛.

청구항 18

제17항에 있어서,

상기 캐시 액세스 유형-기반 배리어 명령을 실행하는 것에 응답하여, 상기 웨이브프론트는 특정 캐시 액세스 유형의 중요한 캐시 액세스가 완료될 때까지 정지되는, 연산 유닛.

청구항 19

컴퓨터 시스템이 있어서,

연산 유닛을 포함하는 가속 프로세싱 디바이스(accelerated processing device); 및

상기 가속 프로세싱 디바이스로 하여금 상기 연산 유닛의 웨이브프론트를 실행하게 하는 프로세서를 포함하고,

상기 연산 유닛은,

웨이브프론트를 수행하도록 구성된 단일-명령-다중-데이터 유닛; 및

캐시 시스템(cache system)을 포함하고, 상기 캐시 시스템은:

제1 리턴 순서화 큐 및 제2 리턴 순서화 큐를 포함하는 복수의 리턴 순서화 큐들을 저장하고, 상기 제1 리턴 순서화 큐는 제2 캐시 액세스 유형이 아닌 제1 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하며, 상기 제2 리턴 순서화 큐는 상기 제1 캐시 액세스 유형이 아닌 제2 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하고;

상기 제1 리턴 순서화 큐의 헤드에서의 제1 엔트리가 웨이브프론트로의 리턴을 위해 이용 가능하다는 것을 결정하고, 상기 제1 엔트리는 제1 캐시 액세스 요청에 대응되고; 그리고

상기 결정에 응답하여, 상기 제1 캐시 액세스 요청 이전의 상기 제2 리턴 순서화 큐의 엔트리들에 대응되는 캐시 액세스 요청들이 상기 웨이브프론트로의 리턴을 위해 이용가능하게 될 것을 기다리지 않고, 상기 제1 엔트리에 대응되는 캐시 리턴을 상기 웨이브프론트로 향하게 하도록 구성되고,

상기 캐시 리턴은 완료된 대응하는 캐시 액세스 요청을 나타내는 데이터를 포함하는, 컴퓨터 시스템.

청구항 20

제19항에 있어서,

상기 제1 캐시 액세스 유형은 판독 유형, 기록 유형 및 텍스처 샘플러 유형 중 하나를 포함하고;

상기 제2 캐시 액세스 유형은 상기 판독 유형, 상기 기록 유형 및 상기 텍스처 샘플러 유형 중 하나를 포함하고; 그리고

상기 제2 캐시 액세스 유형은 상기 제1 캐시 액세스 유형과 상이한, 컴퓨터 시스템.

발명의 설명

기 술 분 야

[0001] 관련 출원에 대한 상호 참조

[0002] 본 출원은 2016년 12월 13일자로 출원된 미국 특허 출원 제15/377,998호의 이익을 주장하며, 이는 본원에서 완전히 설명된 것처럼 참고 문헌으로 포함된다.

[0003] 기술 분야

[0004] 개시된 실시 예들은 일반적으로 그래픽 프로세싱에 관한 것이고, 특히 순서에 관계 없는(out-of-order) 캐시 리턴에 관한 것이다.

배 경 기 술

[0005] 3-차원 그래픽을 렌더링(rendering)하기 위한 하드웨어는 매우 평행하고(highly parallel), 메모리로부터 데이터를 요청하고 데이터에 대한 계산을 수행하며 프로세싱된 데이터를 스크린에 출력하기 위해 프레임 버퍼에 제

공하는 다수의 개별 프로세싱 유닛들을 포함한다. 일반적으로 메모리의 데이터에 액세스하는 데는 많은 양의 레이턴시(latency)가 필요하다. 레이턴시를 줄이기 위해 캐시 시스템(cache system)이 제공된다. 그러나, 렌더링 동작에서 일반적으로 프로세싱되는 많은 양의 데이터로 인해, 메모리 액세스 레이턴시의 추가적인 개선이 바람직하다.

발명의 내용

도면의 간단한 설명

[0006] 첨부된 도면들과 함께 예시로서 주어진 다음의 설명으로부터 더 상세한 이해가 이루어질 수 있으며, 여기서:

도 1은 하나 이상의 개시된 실시 예들이 구현될 수 있는 예시적인 디바이스의 블록도이다;

도 2는 일 예시에 따른 가속 프로세싱 디바이스를 나타내는 블록도이다;

도 3은 일 예시에 따른 그래픽 프로세싱 파이프라인을 도시하는 블록도이다;

도 4는 일 예시에 따른 캐시 시스템을 도시하는 블록도이다;

도 5는 일 예시에 따른 상이한 유형의 캐시 액세스 요청들을 순서화하기 위한 큐들을 도시하는 블록도이다;

도 6은 일 예시에 따른 리턴 순서화 큐(return ordering queue)의 예시이다;

도 7은 일 예시에 따른 액세스 유형별 리턴 순서화 큐에서 캐시 액세스 요청의 표시를 기록하는 방법의 흐름도이다; 그리고

도 8은 일 예시에 따른 워크그룹에 순서에 관계 없는 캐시 리턴을 제공하기 위한 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0007] 본 개시는 순서가 변경된(out of order) 캐시 액세스 리턴(cache access return)들을 허용하는 기술에 관한 것이다. 보다 구체적으로, 리턴 순서화 큐(return ordering queue)는 여러 캐시 액세스 유형들 각각에 대해 존재하며 액세스들이 이루어진 순서대로 중요한(outstanding) 캐시 액세스들을 저장한다. 캐시 액세스 유형들은 판독 유형, 기록 유형 및 텍스처 샘플러 유형을 포함한다. 특정 유형에 대한 캐시 액세스 요청이 해당 유형에 대한 리턴 순서화 큐의 헤드(head)에 있고 캐시 액세스가 해당 액세스를 만든 웨이브프론트(wavefront)로 리턴하기 위해 이용 가능한 경우, 캐시 시스템은 웨이브프론트에 캐시 액세스를 리턴한다. 이 리턴은 사용 가능한 캐시 액세스를 저장하는 리턴 순서화 큐와 연관된 유형 이외의 유형의 캐시 액세스 순서에 관계없이 이루어진다. 따라서, 캐시 액세스들은 다른 유형들의 캐시 액세스들과 관련하여 순서에 관계 없이 리턴될 수 있다. 순서에 관계 없는 리턴들을 허용하는 것은, 예를 들어, 상대적으로 높은-레이턴시 액세스 유형 (예를 들어, 텍스처 샘플러 동작) 후에 비교적 낮은-레이턴시 액세스 유형(예를 들어, 판독)이 발행되는 상황에서, 레이턴시를 개선하는 것을 도울 수 있다.

[0008] 도 1은 본 개시의 하나 이상의 양태들이 구현되는 예시적인 디바이스(100)의 블록도이다. 디바이스(100)는 예를 들어 컴퓨터, 게임 디바이스, 핸드헬드 디바이스, 셋탑 박스, 텔레비전, 이동 전화 또는 태블릿 컴퓨터를 포함한다. 디바이스(100)는 프로세서(102), 메모리(104), 저장 디바이스(106), 하나 이상의 입력 디바이스들(108) 및 하나 이상의 출력 디바이스들(110)을 포함한다. 디바이스(100)는 또한 입력 디바이스들(108) 및 출력 디바이스들(110)을 각각 구동시키는 입력 드라이버들(112) 및 출력 드라이버들(114)을 포함한다. 디바이스(100)는 도 1에 도시되지 않은 추가적인 구성 요소들을 포함할 수 있는 것으로 이해된다.

[0009] 프로세서(102)는 중앙 처리 장치(CPU), 그래픽 처리 장치(GPU), 동일한 다이 상에 위치한 CPU 및 GPU, 또는 하나 이상의 프로세서 코어들을 포함하며, 각 프로세서 코어는 CPU 또는 GPU일 수 있다. 메모리(104)는 프로세서(102)와 동일한 다이 상에 위치되거나 또는 프로세서(102)와 별도로 위치될 수 있다. 메모리(104)는 예를 들어 랜덤 액세스 메모리(RAM), 동적 RAM 또는 캐시와 같은 휘발성 또는 비-휘발성 메모리를 포함한다.

[0010] 저장 디바이스(106)는 예를 들어, 하드 디스크 드라이브, 솔리드 스테이트 드라이브, 광 디스크 또는 플래시 드라이브와 같은 고정식 또는 착탈식 저장소를 포함한다. 입력 디바이스들(108)은 키보드, 키패드, 터치 스크린, 터치 패드, 검출기, 마이크로폰, 가속도계, 자이로스코프, 생체 인식 스캐너 또는 네트워크 연결(예를 들어, 무선 IEEE 802 신호의 송신 및/또는 수신을 위한 무선 근거리 네트워크 카드)을 포함한다. 출력 디바이스들(110)은 디스플레이, 스피커, 프린터, 햅틱 피드백 디바이스, 하나 이상의 조명, 안테나 또는 네트워크 연결(예를 들

어, 무선 IEEE 802 신호의 송신 및/또는 수신을 위한 무선 근거리 네트워크 카드)을 포함한다.

[0011] 입력 드라이버들(112)은 프로세서(102) 및 입력 디바이스들(108)과 통신하며, 프로세서(102)가 입력 디바이스들(108)로부터 입력을 수신하도록 허용한다. 출력 드라이버들(114)은 프로세서(102) 및 출력 디바이스들(110)과 통신하며, 프로세서(102)가 출력 디바이스들(110)로 출력을 전송하도록 허용한다. 출력 드라이버들(114)은 디스플레이 디바이스(118)에 연결된 가속 프로세싱 디바이스(accelerated processing device, APD)(116)를 포함한다. APD(116)는 프로세서(102)로부터 연산 커맨드(compute command) 및 그래픽 렌더링 커맨드(graphics rendering command)를 수신하고, 이들 연산 및 그래픽 렌더링 커맨드를 처리하고, 디스플레이를 위해 디스플레이 디바이스(118)에 픽셀 출력을 제공하도록 구성된다.

[0012] APD(116)는 단일-명령-다중-데이터(ingle-instruction-multiple-data, "SIMD") 패러다임에 따라 계산을 수행하도록 구성된 하나 이상의 병렬 프로세싱 유닛들을 포함한다. 그러나, APD(116)에 의해 수행되는 것으로 기술된 기능은 SIMD 패러다임에 따라 데이터를 프로세싱하지 않는 프로세싱 디바이스들에 의해 수행될 수도 있다.

[0013] 도 2는 일 예시에 따른 가속 프로세싱 디바이스(116)의 블록도이다. 프로세서(102)는, 시스템 메모리(104)에서, 프로세서(102)에 의한 실행을 위한 하나 이상의 제어 로직 모듈들을 유지한다. 제어 논리 모듈들은 운영 시스템(120), 드라이버(122) 및 어플리케이션들(126)을 포함한다. 이들 제어 논리 모듈들은 프로세서(102) 및 APD(116)의 동작의 다양한 양태들을 제어한다. 예를 들어, 운영 시스템(120)은 하드웨어와 직접 통신하고 프로세서(102) 상에서 실행되는 다른 소프트웨어를 위한 하드웨어에 인터페이스를 제공한다. 드라이버(122)는 예를 들어 APD(116)의 다양한 기능에 액세스하기 위해 프로세서(102) 상에서 실행되는 소프트웨어(예를 들어, 어플리케이션들(126))에 어플리케이션 프로그래밍 인터페이스(application programming interface, "API")를 제공함으로써 APD(116)의 동작을 제어한다. 드라이버(122)는 또한 APD(116)의 구성 요소들(예를 들어, 이하 더 상세히 설명되는 SIMD 유닛들(138)과 같은)을 프로세싱함으로써 실행을 위해 셰이더 프로그램(shader program)들을 컴파일링하는 저스트-인-타임 컴파일러(just-in-time compiler)를 포함한다.

[0014] APD(116)는 병렬 프로세싱에 적합한 그래픽 동작 및 비-그래픽 동작과 같은 선택된 기능을 위한 커맨드 및 프로그램을 실행한다. APD(116)는 픽셀 동작, 지오메트리 계산과 같은 그래픽 파이프라인 동작들을 실행하고 프로세서(102)로부터 수신된 커맨드에 기초하여 이미지를 디스플레이 디바이스(118)에 렌더링하는데 사용될 수 있다. APD(116)는 또한 프로세서(102)로부터 수신된 커맨드에 기초하여, 비디오, 물리 시뮬레이션, 전산 유체 역학(computational fluid dynamics) 또는 다른 작업들과 관련된 동작들과 같은, 그래픽 동작들에 직접적으로 관련되지 않거나 또는 그래픽 프로세싱 파이프 라인의 "정상적인"정보 흐름의 일부가 아닌 연산 프로세싱 동작들을 또한 실행한다.

[0015] APD(116)는 SIMD 패러다임에 따라 병렬 방식(parallel manner)으로 프로세서(102)의 요청에 따라 동작들을 수행하도록 구성된 하나 이상의 SIMD 유닛들(138)을 포함하는 연산 유닛들(132)(본 명세서에서 집합적으로 "프로그램 가능 프로세싱 유닛(202)"이라 지칭될 수 있음)을 포함한다. SIMD 패러다임은 다중 프로세싱 요소들이 단일 프로그램 제어 흐름 유닛과 프로그램 카운터를 공유하여 동일한 프로그램을 실행하지만 다른 데이터로 해당 프로그램을 실행할 수 있는 패러다임이다. 일 예에서, 각 SIMD 유닛 (138)은 16 개의 레인(lane)들을 포함하고, 여기서 각각의 레인은 SIMD 유닛(138)의 다른 레인들과 동일한 시간에 동일한 명령을 실행하지만 상이한 데이터로 그 명령을 실행할 수 있다. 모든 레인들이 주어진 명령을 실행할 필요가 없다면, 레인들은 예측(predication)으로 스위치 오프(switched off)될 수 있다. 예측은 또한 다양한 제어 흐름으로 프로그램을 실행하기 위해 사용될 수 있다. 더 구체적으로, 제어 흐름이 개별 레인들에 의해 수행되는 계산에 기초한 조건부 브랜치들(conditional branches) 또는 다른 명령들을 갖는 프로그램에 대해, 현재 실행되고 있지 않은 제어 흐름 경로에 대응되는 레인들의 예측 및 상이한 제어 흐름 경로의 직렬 실행(serial execution)은 임의의(arbitrary) 제어 흐름이 뒤따르게 한다. 연산 유닛들(132)은 APD(116) 내의 APD 메모리(139)와 같은 메모리 또는 시스템 메모리(104)로부터 검색된 데이터를 캐싱하는 캐시 시스템(140)을 포함한다.

[0016] 연산 유닛들(132)에서의 실행의 기본 단위(basic unit)는 작업-항목(work-item)이다. 각 작업-항목은 특정 레인에서 병렬로 실행될 프로그램의 단일 인스턴스 생성(single instantiation)을 나타낸다. 작업 항목들은 단일 SIMD 유닛(138)에서 "웨이브 프론트"로 동시에 실행될 수 있다. 동일한 프로그램을 실행하도록 지정된 작업 항목들의 모음을 포함하는 "작업 그룹(work group)"에 여러 개의 웨이브프론트들이 포함될 수 있다. 작업 그룹은 작업 그룹을 구성하는 각각의 웨이브프론트들을 실행함으로써 실행될 수 있다. 웨이브프론트들은 단일 SIMD 유닛(138) 상에서 순차적으로 또는 상이한 SIMD 유닛들(138)상에서 부분적으로 또는 완전히 병렬로 실행될 수 있다. 웨이브프론트들은 단일 SIMD 유닛(138)에서 동시에 실행될 수 있는 작업 항목의 가장 큰 집합으로 생각될

수 있다. 따라서, 프로세서(102)로부터 수신된 커맨드가 특정 프로그램이 단일 SIMD 유닛(138) 상에서 동시에 실행될 수 없는 정도로 병렬화될 것이라는 것을 나타내면, 그 프로그램은 두 개 이상의 SIMD 유닛들(138) 상에서 병렬화되거나 동일한 SIMD 유닛(138) 상에 직렬화되는 웨이브프론트들로 나뉜다(또는 필요에 따라 병렬화되고 직렬화된다). 스케줄러(scheduler)(136)는 상이한 연산 유닛들(132) 및 SIMD 유닛(138) 상에 다양한 웨이브프론트들을 스케줄링하는 것과 관련된 동작을 수행하도록 구성된다. 스케줄링은 SIMD 유닛들(138) 상에서 실행을 위한 웨이브프론트들을 할당하는 것, 웨이브프론트이 종료된 시점을 결정하는 것, 웨이브프론트들이 언제 멈추고 다른 웨이브프론트들과 교체되어야 하는지 결정하는 것, 그리고 다른 스케줄링 작업을 포함한다.

[0017] 연산 유닛(132)에 의해 제공되는 병렬성(parallelism)은 픽셀 값 계산, 정점 변환 및 다른 그래픽 동작과 같은 그래픽 관련 동작들에 적합하다. 따라서, 프로세서(102)로부터 그래픽 프로세싱 커맨드를 받아들이는 그래픽 프로세싱 파이프라인(134)은 계산 태스크를 병렬로 실행하기 위해 컴퓨팅 유닛들(132)에 제공한다.

[0018] 연산 유닛(132)은 또한 그래픽과 관련되지 않은 계산 태스크를 수행하거나 그래픽 프로세싱 파이프라인(134)의 "일반적인" 동작(예를 들어, 그래픽 프로세싱 파이프 라인(134)의 동작을 위해 수행된 프로세싱을 보완하기 위해 수행되는 커스텀(custom) 동작)의 일부로서 수행되지 않는 연산 태스크를 수행하는 데 사용된다. 어플리케이션(126) 또는 프로세서(102) 상에서 실행되는 다른 소프트웨어는 그러한 계산 태스크를 정의하는 프로그램(종종 " 연산 셰이더 프로그램"이라고도 지칭됨)을 실행을 위해 APD(116)에 전송한다.

[0019] 도 3은 도 2에 도시된 그래픽 프로세싱 파이프라인(134)의 추가적인 세부 사항을 도시하는 블록도이다. 그래픽 프로세싱 파이프라인(134)은 각각 특정 기능을 수행하는 스테이지들을 포함한다. 스테이지들은 그래픽 프로세싱 파이프라인(134)의 기능의 서브디비전들(subdivisions)을 나타낸다. 각 스테이지는 프로그램 가능 프로세싱 유닛들(202)에서 실행되는 셰이더 프로그램으로서 부분적으로 또는 전체적으로, 또는 프로그램 가능 프로세싱 유닛들(202) 외부의 고정된-기능, 비-프로그램 가능 하드웨어로서 부분적으로 또는 전체적으로 구현된다.

[0020] 입력 어셈블러 스테이지(302)는 사용자-채움 버퍼들(user-filled buffers)(예를 들어, 어플리케이션(126)과 같은 프로세서(102)에 의해 실행되는 소프트웨어의 요청으로 채워진 버퍼들)로부터 프리미티브 데이터를 판독하고 파이프라인의 나머지에 의한 사용을 위해 데이터를 프리미티브들로 어셈블링한다. 입력 어셈블러 스테이지(302)는 사용자-채움 버퍼들에 포함된 프리미티브 데이터에 기초하여 상이한 타입의 프리미티브들을 생성할 수 있다. 입력 어셈블러 스테이지(302)는 나머지 파이프라인에 의한 사용을 위해 어셈블링된 프리미티브들을 포맷한다.

[0021] 정점 셰이더 스테이지(vertex shader stage)(304)는 입력 어셈블러 스테이지(302)에 의해 어셈블링된 프리미티브들의 정점들(vertices)을 프로세싱한다. 정점 셰이더 스테이지(304)는 변환, 스키닝, 모핑 및 정점-당 라이팅(per-vertex lighting)과 같은 다양한 정점-당 동작을 수행한다. 변환 동작은 정점들의 좌표를 변환하는 다양한 동작들을 포함할 수 있다. 이러한 작업들은 모델링 변환(modeling transformation), 보기 변환(viewing transformation), 투영 변환(projection transformation), 원근 분리(perspective division) 및 뷰포트 변환(viewport transformation) 중 하나 이상을 포함할 수 있다. 여기서, 그러한 변환들은 변환들이 수행되는 정점들의 좌표 또는 "위치(position)"를 수정하는 것으로 고려된다. 정점 셰이더 스테이지(304)의 다른 동작들은 좌표들 이외의 속성들을 수정할 수 있다.

[0022] 정점 셰이더 스테이지(304)는 하나 이상의 연산 유닛들(132) 상에서 실행될 정점 셰이더 프로그램들로서 부분적으로 또는 전체적으로 구현된다. 정점 셰이더 프로그램은 프로세서(102)에 의해 제공되고 컴퓨터 프로그래머에 의해 미리 기록된 프로그램에 기초한다. 드라이버 (122)는 연산 유닛들(132) 내에서의 실행에 적합한 포맷을 갖는 정점 셰이더 프로그램을 생성하기 위해 그러한 컴퓨터 프로그램을 컴파일링한다.

[0023] 헐 셰이더 스테이지(hull shader stage)(306), 테셀레이터 스테이지(tessellator stage)(308), 및 도메인 셰이더 스테이지(310)는 함께 동작하여 테설레이션을 구현하며, 이는 프리미티브들을 세분화하여 단순 프리미티브들을 보다 복잡한 프리미티브들로 전환한다. 헐 셰이더 스테이지(306)는 입력 프리미티브에 기초하여 테셀레이션을 위한 패치(patch)를 생성한다. 테설레이터 스테이지(308)는 패치에 대한 샘플들의 세트를 생성한다. 도메인 셰이더 스테이지(310)는 패치에 대한 샘플들에 대응되는 정점들에 대한 정점 위치들을 계산한다. 헐 셰이더 스테이지(306) 및 도메인 셰이더 스테이지(310)는 프로그램 가능 프로세싱 유닛들(202) 상에서 실행될 셰이더 프로그램으로서 구현될 수 있다.

[0024] 지오메트리 셰이더 스테이지(geometry shader stage)(312)는 프리미티브-바이-프리미티브 베이시스(primitive-by-primitive basis)에 대해 정점 동작들을 수행한다. 포인트 스프린트 확장(point sprint expansion), 동적

입자 시스템 동작(dynamic particle system operation), 퍼- 핀 생성(fur-fin generation), 쉐도우 볼륨 생성(shadow volume generation), 단일 패스 렌더-투-큐브맵(single pass render-to-cubemap), 프리미티브-당 물질 교환(per-primitive material swapping) 및 프리미티브-당 물질 설정(per-primitive material setup)과 같은 동작을 포함하는, 다양한 다른 유형들의 동작들이 지오메트리 셰이더 스테이지(312)에 의해 수행될 수 있다. 지오메트리 셰이더 스테이지(312)에 대한 동작들은 프로그램 가능 프로세싱 유닛(202) 상에서 실행되는 셰이더 프로그램에 의해 수행될 수 있다.

[0025] 래스터라이저 스테이지(rasterizer stage)(314)는 단순한 프리미티브들 및 생성된 업스트림(upstream)을 수용하고 레스터화한다. 래스터화는 어떤 스크린 픽셀들(또는 서브-픽셀 샘플들)이 특정 프리미티브로 덮여 있는지를 결정하는 것으로 구성된다. 래스터화는 고정 기능 하드웨어(fixed function hardware)로 수행된다.

[0026] 픽셀 셰이더 스테이지(316)는 프리미티브 생성된 업스트림 및 래스터화의 결과에 기초하여 스크린 픽셀들에 대한 출력 값들을 계산한다. 픽셀 셰이더 스테이지(316)는 텍스처 메모리로부터 텍스처를 적용할 수 있다. 픽셀 셰이더 스테이지(316)에 대한 동작들은 프로그램 가능 프로세싱 유닛(202) 상에서 실행되는 셰이더 프로그램에 의해 수행된다.

[0027] 출력 병합 스테이지(output merger stage)(318)는 픽셀 셰이더 스테이지(316)로부터의 출력을 수용하고, 스크린 픽셀에 대한 최종 컬러를 결정하기 위해 z-테스팅 및 알파 블렌딩과 같은 동작들을 수행하는, 이들 출력을 병합한다.

[0028] 텍스처를 정의하는 텍스처 데이터는 텍스처 유닛(320)에 의해 저장 및/또는 액세스된다. 텍스처는 그래픽 프로세싱 파이프라인(134)의 다양한 포인트들에서 사용되는 비트맵 이미지이다. 예를 들어, 일부 경우에, 픽셀 셰이더 스테이지(316)는 렌더링될 정점들의 수를 증가시키지 않으면서 픽셀에 텍스처를 적용하여 명백한 렌더링 복잡성을 개선한다(예를 들어, 보다 "사진과 동일한(photorealistic)" 외관을 제공한다).

[0029] 어떤 경우에, 정점 셰이더 스테이지(304)는 텍스처 유닛(320)으로부터의 텍스처 데이터를 사용하여 예를 들어 개선된 미학(aesthetics)을 위한 정점들을 생성하거나 수정함으로써 프리미트브들을 수정하여 복잡성을 증가시킨다. 일 예시에서, 정점 셰이더 스테이지(304)는 텍스처 유닛(320)에 저장된 높이 맵(height map)을 사용하여 정점들의 변위를 수정한다. 이러한 유형의 기술은, 예를 들어, 물을 렌더링하는데 사용되는 정점들의 위치 및 수를 변경함으로써 픽셀 셰이더 스테이지(316)에서만 사용되는 텍스처들과 비교하여 보다 현실감 있는 물을 생성하는 데 사용될 수 있다. 일부 예에서, 지오메트리 셰이더 스테이지(312)는 텍스처 유닛(320)으로부터 텍스처 데이터를 액세스한다.

[0030] 도 4는 일 예시에 따른 캐시 시스템(140)을 도시한다. 캐시 시스템(140)은 SIMD 유닛들(138)에서 실행하는 웨이브프론트들로부터 캐시 액세스에 대한 요청들을 수신하고 그러한 요청들을 프로세싱한다. 이러한 요청을 프로세싱하는 것의 일부는 연산 유닛(132)의 캐시 시스템(140)의 하나 이상의 캐시 메모리들(404)에서 요청된 데이터를 검색하는 것을 포함한다. 특히, 캐시 시스템(140)은 캐시 계층의 로우 레벨(제1 레벨과 같은)으로서 동작하는 하나 이상의 캐시 메모리들(404)을 갖는다. 웨이브프론트(412)에 의해 액세스되도록 요청된 데이터가 캐시 메모리(404)에 존재하지 않으면, 캐시 시스템(140)은 계층 인터페이스(hierarchy interface)(406)를 통해 캐시 계층 내의 다른 메모리들(예를 들어, 보다 높은 레벨 캐시 메모리, APD 메모리(139) 및/또는 시스템 메모리(104))에 액세스하여 요청된 데이터에 액세스한다.

[0031] 일부 예에서, 캐시 액세스 요청은 벡터 기반 요청(vector-based request)이다. 벡터 기반 요청은, SIMD 유닛(138) 상에서 병렬화된 동작들과 비슷한, 다수의 메모리 위치들에서 데이터를 요청할 수 있는 기능이 있다. 예를 들어, 각 작업 항목이 다른 어드레스를 지정하는, 단일 웨이브프론트의 다른 작업 항목들에 의해 실행되는, 로드 명령과 같은 단일 명령은 메모리의 여러 위치에서 판독을 초래한다. 벡터 기반 요청은 일반적으로 단일 메모리 위치에 데이터를 판독하거나 기록하는 스칼라 요청과 대조된다.

[0032] 캐시 시스템(140)의 리턴 순서화 큐(402)는 완료된 캐시 액세스가 순서대로 웨이브프론트(412)에 리턴되도록 완료된 캐시 액세스를 순서화한다. 순서대로 액세스들을 리턴한다는 것은 액세스들이 이루어진 순서로 액세스들을 한 웨이브프론트(412)에 리턴 데이터를 제공하는 것을 의미한다. 일부 예에서, 캐시 시스템(140)은 웨이브프론트-당 베이시스로 캐시 액세스 요청들을 순서화하고, 이는 특정 웨이브프론트에 의해 이루어진 캐시 액세스 요청들의 표시들이 웨이브프론트(412)가 그러한 요청들을 한 순서로 저장되지만, 다른 웨이브프론트들(412)에 의해 이루어진 캐시 액세스 요청들의 순서는 유지되지 않는다는 것을 의미한다. 이러한 캐시 액세스 요청들의 웨이브프론트별 순서화를 달성하기 위해, 각각의 웨이브프론트(412)는 캐시 액세스 요청들의 순서를 유지하기 위

해 자신의 큐 메모리 공간 세트에 할당될 수 있다.

[0033] 리턴 순서화 큐(402)는 각각의 캐시 액세스에 대한 식별자(identifier)를 저장하며, 식별자들은 웨이브프론트(412)에 의해 각각의 캐시 액세스들이 이루어지는 순서로 저장된다. 새로운 엔트리들-즉 웨이브프론트(412)로부터의 가장 최근에 이루어진 캐시 액세스 요청에 대응하는 엔트리들-이 리턴 순서화 큐(402)의 테일(tail)에 제공된다. 리턴 순서화 큐의 헤드는 가장 오래된 중요한 캐시 액세스 요청의 표시 및 그에 따라 리턴될 다음 캐시 액세스를 저장한다. 리턴 순서화 큐(402)의 헤드에서의 액세스에 대한 데이터가 캐시 메모리(404)에서 이용 가능할 때(예를 들어, 데이터의 적어도 일부가 상위 레벨 캐시 메모리, 시스템 메모리(104) 또는 APD 메모리(139)로부터 캐시 메모리(404)로 인출된(fetched)), 캐시 시스템(140)은 그 데이터를 요청 웨이브프론트(412)에 제공한다.

[0034] 캐시 리턴들을 순서화하는 한 가지 기술은 캐시 리턴의 "유형"에 관계 없이 모든 캐시 리턴들을 순서화하는 것이다. 여기서, 캐시 리턴 "유형"(이는 또한 캐시 액세스 요청의 유형이기도 함)은 판독 유형, 기록 유형 또는 텍스처 샘플러 유형 중 하나를 지칭한다. 판독 유형 액세스는 메모리에서 데이터를 요청하고 요청된 데이터가 메모리로부터 리턴되는 액세스이다. 판독 유형 액세스는 가상 또는 물리적 어드레스로 메모리 위치를 지정하며, 샘플러 동작과 달리, 텍스처 좌표를 지정하지 않는다. 판독 유형 액세스에 대한 리턴은 데이터를 요청한 웨이브프론트(412)로 요청된 데이터를 리턴하는 것을 의미한다(예를 들어, 요청된 데이터를 웨이브프론트(412)를 실행하는 SIMD 유닛(138)의 레지스터들에 배치하여 SIMD 유닛(138)이 그 데이터에 의존하는 명령들을 실행할 수 있다). 판독 유형 액세스들의 경우, 리턴 데이터는 판독 유형 액세스 요청에 의해 요청된 데이터이다.

[0035] 기록 유형 액세스에 대한 리턴은 "확인 응답된(acknowledged)" 신호를 기록을 요구한 웨이브프론트(412)를 실행하는 SIMD 유닛(138)에 리턴하는 것을 의미한다. "확인 응답된" 신호는 요청된 기록이 메모리 시스템에 의해 확인 응답되었음을 SIMD 유닛(138)에 지시하는 신호이다. "확인 응답된" 신호를 수신하면 SIMD 유닛(138)은 "확인 응답된"신호에 의존하는 동작을 진행할 수 있다. 기록 유형 액세스의 경우, 리턴 데이터는 "확인 응답된" 신호이다.

[0036] 다른 동작들과 마찬가지로, 원자 동작(atomic operation)(이는 판독-수정-기록 동작과 같은 복잡한 동작일 수 있음)은 동작 완료에 응답하여 리턴되는 신호 유형에 따라 판독 또는 기록으로 분류된다. 보다 구체적으로, 웨이브프론트에 데이터를 리턴하는 원자 동작은 판독 동작으로 분류된다. 웨이브프론트에 확인 응답된 신호를 리턴하는 원자 동작은 기록 동작으로 분류됩니다.

[0037] 메모리 동작의 텍스처 샘플러 유형은 텍스처 데이터에 대한 요청을 수신하는 것, 상기 요청에 대한 프로세싱을 수행하여 실제로 필요한 저장된 데이터를 결정하는 것, 그 저장된 데이터를 인출(fetching)하는 것, 선택적으로 저장된 데이터를 압축해제(decompressing) 및/또는 필터링하여 픽셀 또는 샘플 값을 얻는 것 및 데이터를 요청한 웨이브프론트(412)를 실행하는 SIMD 유닛(138)에 픽셀 또는 샘플 값을 리턴하는 것을 포함하는 복잡한 동작이다. 텍스처 샘플러 유형 액세스에 대한 리턴은 데이터를 요청한 웨이브프론트(412)를 실행하는 SIMD 유닛(138)에 요청된 데이터를 리턴하는 것을 의미한다. 텍스처 샘플러 유형 액세스들의 경우, 리턴 데이터는 요청 SIMD 유닛(138)에 리턴되는 픽셀 또는 샘플 값이다.

[0038] 텍스처 샘플러 동작들은 많은 수의 동작들이 포함되므로 많은 양의 레이턴시가 필요하다. 예를 들어, 텍스처 샘플 액세스 요청에는 일반적으로 데이터가 필요한 텍스처 비트맵의 위치를 식별하는 텍스처 좌표(u, v, w)가 포함된다. 텍스처 샘플 액세스에 대한 요청은 스크린 좌표의 변화율과 비교하여 텍스처 좌표의 변화율을 지정하는 하나 이상의 그래디언트 값을 포함할 수 있으며 밉맵 레벨(mipmap level)을 식별하는 데이터, 큐브맵 페이스를 식별하는 데이터 또는 다른 데이터와 같은 다른 데이터를 포함할 수 있다. 이 데이터를 기초로, 캐시 시스템(140)의 텍스처 유닛(408)은 요청된 데이터가 발견될 어드레스를 식별하고, 메모리 시스템으로부터 데이터를 가져오고(이는 연산 유닛(132)의 캐시 시스템(140)의 캐시 메모리(404)에 히트하거나 캐시 메모리(404)에 누락되어 캐시 계층 내의 다른 메모리들로부터 누락된 데이터를 캐시 메모리(404)로 가져오는 것을 포함할 수 있음), 텍스처 데이터가 압축 될 수 있고, 선택적으로 필터링 방식(예를 들어, 이중선형 필터링(bilinear filtering), 삼중선형 필터링(trilinear filtering), 이방성 필터링(anisotropic filtering))에 따라 데이터를 프로세싱할 수 있기 때문에 선택적으로 데이터를 압축 해제하고, 텍스처 샘플러 캐시 액세스 요청을 한 웨이브프론트(412)를 실행하는 SIMD 유닛(138)에 데이터를 송신한다. 이러한 동작은 수백 개의 컴퓨터 클럭 사이클들이 소요될 수 있으며, 이는 많은 양의 레이턴시를 나타낸다(여기서, 레이턴시는 SIMD 유닛(138)이 텍스처 샘플링 동작을 수행할 것을 요청한 이후부터 텍스처 샘플링 동작에 의해 얻어진 데이터가 SIMD 유닛(138)에 제공될 때까지의 시간이다).

[0039] 판독 또는 기록 동작과 비교할 때 샘플러 동작의 높은 레이턴시 때문에, 캐시 리턴들을 순서화하기 위한 다른 모드들이 본원에서 제공된다. 모든 유형의 캐시 리턴들을 순서화하는 단일 큐 대신 이러한 다른 모드들에서, 리턴 순서화 큐(402)는 상이한 유형들의 캐시 리턴들의 순서를 유지하는 둘 이상의 개별 큐들을 포함한다. 몇 가지 모드들이 가능하다. 상이한 모드들 사이에서의 선택은 프로세서(102)로부터의 요청, SIMD 유닛(138)에서 실행되는 명령, APD(116)에서 실행되는 펌웨어에 의한 알고리즘 결정, 또는 임의의 다른 기술적으로 실현 가능한 메커니즘에 응답하여 행해질 수 있다.

[0040] 하나의 모드에서, 큐는 캐시 액세스 요청들의 "샘플러", "판독" 및 "기록" 유형들 각각에 대해 유지된다. 각각의 큐는 각 유형의 캐시 액세스 표시를 저장한다. 예를 들어, 샘플러 큐는 액세스가 이루어진 순서대로 샘플러 유형의 캐시 액세스의 표시를 유지하지만 판독 유형이나 기록 유형은 표시는 유지하지 않는다. 유사하게, 판독 큐는 판독 유형의 캐시 액세스들의 순서화된 표시들을 유지하지만 기록 유형 또는 샘플러 유형의 표시는 유지하지 않으며 기록 큐는 기록 유형의 캐시 액세스의 순서화된 표시를 유지하지만 판독 유형 또는 샘플러 유형의 표시는 유지하지 않는다.

[0041] 각 큐는 헤드와 테일을 갖는다. 헤드는 해당 큐와 연관된 유형의 가장 오래된 캐시 액세스 요청을 나타내고 테일은 해당 큐와 연관된 유형의 최신 캐시 액세스 요청을 나타낸다. 특정 큐의 헤드에서의 캐시 액세스 요청에 대한 모든 데이터가 캐시 메모리(404)에 저장되고 따라서 "이용 가능"하면, 캐시 시스템(140)은 추가 프로세싱을 위해 그 데이터를 SIMD 유닛(138)으로 리턴한다. 둘 이상의 큐의 헤드에서 캐시 액세스가 이용 가능한 경우, 캐시 시스템(140)은, 서로 다른 큐들의 헤드들에서 캐시 액세스들이 서로 관련하여 발행되는 순서에 관계 없이, 임의의 기술적으로 가능한 중재 방식(예를 들면, 라운드-로빈)에 따라, 큐의 헤드에서 이용 가능한 캐시 액세스들 중 임의의 것을 선택할 수 있다.

[0042] 또 다른 모드에서, 샘플러 동작들 및 판독 동작들을 순서화하는 하나의 큐가 유지되고, 기록 동작들을 순서화하는 다른 큐가 유지된다. 다른 말로, 제1 큐는 웨이브프론트로부터 액세스 요청들이 수신된 순서로 캐시 액세스 요청들의 "판독" 유형 및 캐시 액세스의 "샘플러" 유형 모두의 표시를 저장하고, 제2 큐는 웨이브프론트로부터 액세스 요청이 수신된 순서로 캐시 액세스들의 "기록" 유형의 표시를 저장한다. 따라서, 판독 유형 캐시 액세스들 및 샘플러 유형 캐시 액세스들은 서로에 대해 순서화되지만 기록 유형 캐시 액세스들과는 순서화되지 않으며, 기록 유형 캐시 액세스들은 서로에 대해 순서화되지만 판독 유형 캐시 액세스들 또는 샘플러 유형 캐시 액세스들과 관련하여 순서화되지 않는다.

[0043] 판독 및 기록을 순서화하는 큐와 샘플러 동작들을 순서화하는 또 다른 큐를 갖는 하나의 모드 또는 기록 및 샘플러 동작을 순서화하는 큐와 판독들을 순서화하는 또 다른 큐를 갖는 다른 모드와 같은 다른 모드들도 가능하다.

[0044] 도 5는 일 예시에 따른 상이한 유형의 캐시 액세스 요청들을 순서화하기 위한 큐들을 도시하는 블록도이다. 큐들은 웨이브프론트(412)에 의해 요청들이 이루어진 순서로 판독-유형 캐시 액세스 요청들을 나타내는 엔트리들(514)을 저장하는 판독 큐(502), 웨이브프론트(412)에 의해 요청들이 이루어진 순서로 기록-유형 캐시 액세스 요청들을 나타내는 엔트리들(514)을 저장하는 기록 큐 및 웨이브프론트(412)에 의해 요청들이 이루어진 순서로 샘플러-유형 캐시 액세스 요청들을 나타내는 엔트리들(514)을 저장하는 샘플러 큐(506)를 포함한다. 각 엔트리(514)는 하나 이상의 캐시 액세스 요청들에 대한 데이터를 저장할 수 있다.

[0045] 동작 시, 웨이브프론트(412)는 캐시 액세스 요청들을 포함할 수 있는 셰이더 프로그램의 일부로서 실행한다. 그러한 요청이 이루어질 때, 캐시 시스템(140)은 그 요청을 검출하고, 그 요청이 판독 유형 요청인지, 기록 유형 요청인지 또는 샘플러 유형 요청인지 여부를 결정한다. 판독 유형 요청은 메모리로부터 웨이브프론트(412)로 데이터를 반환하는 샘플러 유형 요청이 아닌 요청이다. 기록 유형 요청은 데이터를 메모리에 저장하고 확인 응답 신호를, 메모리로부터의 데이터 없이, 웨이브 프런트(412)리턴한다. 샘플러 유형의 요청은 적어도 텍스처 좌표를 받고, 텍스처 데이터를 저장하는 메모리 내의 하나 이상의 메모리 위치들을 식별하기 위해 좌표 및 다른 데이터를 프로세싱하고, 텍스처 데이터를 검색하고 텍스처 데이터를 웨이브프론트(412)에 리턴하는 요청이다.

[0046] 캐시 액세스 요청이 어느 유형인지를 결정하는 것에 응답하여, 캐시 시스템(140)은 요청을 나타내는 엔트리(514)를 적절한 큐(예를 들어, 판독 큐(502), 기록 큐(504) 또는 샘플러 큐(506))의 테일(510)에 위치시킨다. 엔트리들(514)은 다른 엔트리들이 큐의 헤드(512)로부터 제거됨에 따라 각 큐의 헤드 (512)를 향해 이동하며, 웨이브 프론트(412)에 리턴되는 이러한 엔트리들(514)에 대한 데이터 또는 확인 응답으로 인한 것이다. 요청을 나타내는 엔트리(514)가 적절한 큐에 있는 동안, 캐시 시스템(140)은 요청을 만족 시키도록 작동한다.

[0047] 판독 요청들에 대해, 캐시 시스템(140)은 판독되도록 요청된 데이터가 캐시 메모리(404)에 있는지를 결정한다. 데이터가 캐시 메모리(404)에 있으면, 캐시 시스템(140)은 해당 엔트리(514)를, 엔트리(514)가 판독 큐(502)의 헤드(512)에 있을 때 적절한 캐시 리턴이 웨이브프론트(412)에 제공될 수 있음을 나타내는, "이용 가능한(available)"으로서 마킹한다. 데이터가 캐시 메모리(404)에 아직 없는 경우, 해당 엔트리(514)는 아직 이용 가능하다고 마킹되지 않는다. 캐시 시스템(140)은 계층 인터페이스(406)를 통해 캐시 계층의 다른 레벨들로부터 적절한 데이터를 인출한다. 인출 후, 모든 데이터가 캐시 메모리(404)에서 이용 가능할 때, 캐시 시스템(140)은 엔트리(514)가 이용 가능하다고 마킹하고, 엔트리(514)가 판독 큐(502)의 헤드(512)에 있을 때 데이터를 웨이브프론트(412)로 리턴한다.

[0048] 샘플러 요청에 대해, 캐시 시스템(140)은 샘플러 요청들을 만족시키기 위해 하나 이상의 메모리 어드레스들을 식별하기 위해 샘플러 요청에 대해 적절한 프로세싱을 수행한다. 샘플러 요청들에 대한 메모리 어드레스들을 식별하는 프로세싱이 완료된 후, 캐시 시스템(140)은 샘플러 요청에 대한 데이터가 캐시 메모리(404)에 있는지를 결정한다. 데이터가 캐시 메모리(404)에 있으면, 캐시 시스템(140)은 대응되는 엔트리(514)를 이용 가능한 것으로 마킹한다. 데이터가 캐시 메모리(404)에 있지 않으면, 대응되는 엔트리(514)는 이용 가능한 것으로 마킹되지 않고 캐시 시스템(140)은 계층 인터페이스(406)를 통해 캐시 계층으로부터 데이터를 인출한다. 캐시 메모리(404)에 있는 샘플러 요청에 대한 데이터에 응답하여, 캐시 시스템(140)은 대응되는 엔트리(514)를 이용 가능한 것으로 마킹한다. 대응되는 엔트리(514)가 이용 가능하다고 마킹되고 샘플러 큐(506)의 헤드(512)에 있을 때, 캐시 시스템(140)은 샘플러 요청에 대한 데이터를 웨이브프론트(412)로 리턴한다.

[0049] 기록 요청들에 대해, 캐시 시스템(140)은 데이터를 메모리 시스템에 기록한다. 특히, 캐시 시스템(140)은 데이터를 캐시 메모리(404)에 기록한다. 그 후, 데이터는 계층 인터페이스(406)를 통해 캐시 메모리 계층의 다른 메모리에 기록된다. 캐시 메모리(404) 또는 계층 인터페이스(406)를 통해 도달할 수 있는 외부 캐시 메모리 내의 다른 메모리들의 하나 이상에 데이터를 기록할 때(예를 들어, 캐시 메모리(404)로부터 캐시 계층에서 한 레벨 위로 있는 캐시 메모리에 데이터를 기록할 때), 캐시 시스템(140)은 기록된 데이터에 대응되는 엔트리(514)를 이용 가능한 것으로 마킹한다. 이용 가능한 엔트리(514)가 기록 큐(504)의 헤드(512)에 있는 것에 응답하여, 캐시 시스템(140)은 기록 큐(504)로부터 그 엔트리(514)를 제거하고 확인 응답 신호를 웨이브프론트(412)에 회신한다.

[0050] 도 6은 일 예시에 따른 리턴 순서화 큐(600)를 도시한다. 리턴 순서화 큐(600)는 도 4의 리턴 순서화 큐(402)의 예시이다. 리턴 순서화 큐(600)는 판독 큐(502), 기록 큐(504) 및 샘플러 큐(506)를 포함한다. 판독 큐(502), 기록 큐(504) 및 샘플러 큐(506) 각각은 모놀리식 메모리(monolithic memory)(602) 내부의 가상 큐(virtual queue)로서 구현된다. "가상 큐(virtual queue)"라는 용어는 판독 큐(502), 기록 큐(504) 및 샘플러 큐(506) 각각이 고정된 위치를 갖지 않거나 또는 모놀리식 메모리(602) 내의 길이가 모놀리식 메모리(602)의 상이한 슬롯들(604)에서 시작되거나 끝나는 것을 의미한다(용어 "슬롯(604)"은 하나의 엔트리(514)를 저장하기 위한 크기인 모놀리식 메모리(602)의 유닛을 지칭함). 서로 다른 큐들이 연속적으로(back-to-back) 표시되지만(예를 들어, 인접한 슬롯들(604)의 헤드들(512)과 테일들(510)), 큐들의 헤드들(512)은 때로는 다른 큐들의 테일들(510)에 인접하지 않을 수 있다. 일 실시 예에서, 판독 큐(502)는 모놀리식 메모리(602)에서 소정의 개수의 슬롯들(604)을 할당 받지만, 판독 큐(502)의 헤드(512(1))는 할당된 공간의 끝까지 연장되지 않는다. 이러한 예시에서, 판독 큐(502)의 헤드(512(1))는 기록 큐(504)의 테일(510(2))에 인접하지 않을 것이다. 다른 예에서, 테일(510)은 특정 큐의 시작에 있지 않다. 캐시 시스템(140)은 필요에 따라 각각의 가상 큐의 크기를 변경할 수 있고, 어떤 상황에서는 제로 슬롯(604)을 하나 이상의 큐에 할당할 수 있다. 일 예시에서, 판독 및 샘플러 동작들이 서로에 대해 순서화되고 기록들이 서로에 대해 순서화되지만 판독 및 샘플러 동작들은 기록들에 대해 순서화되지 않는 모드에서, 캐시 시스템(140)은 두 큐들(판독 및 샘플러 동작들에 대해 하나 그리고 기록들에 대해 다른 하나)에 대해 제로보다 많은 슬롯(604)을 할당하고 제3 큐에 제로 슬롯(604)을 할당한다.

[0051] 헤드 추출기(head extractor)(610)는 현재 모드에 대해 활성인 큐들의 각각의 헤드에서 데이터를 추출하고 그 데이터를 각각의 물리적 큐들에 배치한다. 물리적 판독 큐(604), 물리적 기록 큐(606) 및 물리적 샘플러 큐(608)를 포함하는 물리적 큐는 웨이브프론트(412)로의 리턴을 위해 캐시 시스템(140)의 다른 부분들에 제공하기 위해 중재기(arbiter)(612)에 의해 특정 큐의 헤드(512)를 용이하게 판독할 수 있게 한다. 예를 들어, 물리적 큐들이 없는 경우, 중재기는 먼저 큐의 헤드(512)의 어드레스를 검색한 다음 그 헤드(512)에 대응되는 엔트리(514)의 데이터를 얻는다. 헤드 추출기(610)가 각 큐의 헤드에서 엔트리들(514)을 추출하면, 중재기(612)는 고정된 메모리 위치(물리적 큐들의 헤드)를 조사하여 다음 이용 가능한 캐시 메모리 액세스에 대한 데이터를 얻는다. 물리적 큐들은 엔트리들(614)을 저장하기 위한 하나 이상의 슬롯들(613)을 포함한다. 물리적 큐들의 엔트리

들(614)은 가상 큐들(예를 들어, 모놀리식 메모리(602)의 판독 큐(502), 기록 큐(504), 및 샘플러 큐(506))의 엔트리들(514)과 동일하거나 유사한 데이터를 포함한다. 물리적 큐들 각각의 헤드(622)는 대응되는 가상 큐의 대응되는 헤드(512)의 엔트리(514)로부터의 데이터를 갖는 엔트리(614)를 저장한다.

[0052] 중재기(612)는 엔트리(614)가 이용 가능하다고 마킹될 때 물리적 큐의 헤드(622)로부터 그 엔트리(614)를 선택한다. 상이한 물리적 큐들의 헤드들(622)에서의 다중 엔트리들(614)이 이용 가능하다고 마킹되면, 중재기(612)는 웨이브프론트(412)로의 캐시 리턴을 위해 헤드들(622)에서 엔트리들(614) 중 하나를 선택하기 위해 임의의 기술적으로 가능한 중재 방식(라운드 로빈과 같은)을 적용한다.

[0053] 중재기(612)가 물리적 큐의 헤드(622)로부터 엔트리(614)를 판독할 때, 물리적 큐는 다른 엔트리들(614)을 헤드(622)를 향해 이동시키고, 헤드 추출기(610)는 모놀리식 메모리(602) 내의 대응되는 가상 큐의 대응되는 슬롯(604)으로부터 다른 엔트리(514)를 판독하고, 헤드 추출기(610)에 의해 판독된 엔트리(514)로부터의 데이터를 포함하는 새로운 엔트리(614)를 적절한 물리적 큐 내의 새로 비워진 슬롯에 위치시킨다.

[0054] 도 7은 일 예시에 따라, 액세스 유형 당 리턴 순서화 큐들에서 캐시 액세스 요청들의 표시들을 기록하는 방법(700)의 흐름도이다. 도 1 내지 도 6에 도시되고 설명된 시스템과 관련하여 설명되었지만, 임의의 기술적으로 실행 가능한 순서로 방법을 수행하도록 구성된 임의의 시스템이 본 개시의 범위 내에 있다는 것이 이해되어야 한다.

[0055] 도시된 바와 같이, 방법(700)은 캐시 시스템(140)이 웨이브프론트(412)로부터 캐시 액세스 요청을 수신하는 단계(702)에서 시작한다. 캐시 액세스 요청은 캐시 시스템(140)을 통해 메모리에 액세스하기 위한 요청이다. 단계(704)에서, 캐시 시스템(140)은 캐시 액세스 요청이 판독 유형, 기록 유형 또는 텍스처 샘플러 유형 중 어느 것인지를 결정한다. 판독 유형은 하나 이상의 위치들로부터 데이터를 판독하고 그 데이터를 상기 액세스를 행한 웨이브프론트(412)에 리턴하는 액세스 유형이다. 기록 유형은 데이터를 특정 위치에 기록하고 액세스를 행한 웨이브프론트(412)에 "확인 응답된" 신호를 리턴하는 액세스 유형이다. 텍스처 샘플러 유형은 텍스처 좌표와 가능하게는 다른 텍스처 관련 입력을 프로세싱하고, 텍스처 좌표와 가능하게는 다른 텍스처 관련 입력에 기초하여 하나 이상의 메모리 어드레스들을 식별하고, 상기 하나 이상의 메모리 어드레스들로부터 데이터를 인출하고, 데이터를 압축 해제하고 및/또는 데이터를 필터링(예를 들어, 이중선형 필터링, 삼중선형 필터링, 이방성 필터링)하기 위해 데이터를 선택적으로 프로세싱하고 인출된, 프로세싱된 텍스처 데이터를 액세스를 행한 웨이브프론트(412)에 제공하는 액세스 유형이다.

[0056] 단계(706)에서, 캐시 시스템(140)은 캐시 액세스 요청의 표시를 유형과 관련된 유형별 리턴 순서화 큐에 위치시킨다. 예를 들어, 캐시 액세스 요청이 기록 유형이면, 캐시 시스템(140)은 그 캐시 액세스 요청의 표시를 기록 유형과 관련된 리턴 순서화 큐에 위치시킨다. 각 큐는 캐시 액세스 요청들이 수신된 순서대로 캐시 액세스 요청을 유지한다. 가장 오래된 캐시 액세스 요청은 큐의 헤드에 있고 가장 최근의 캐시 액세스 요청은 큐의 테일에 있다. 각각의 큐는 캐시 시스템(140)에 대해 설정된 모드에 따라 1, 2 또는 3 가지 유형의 요청을 순서화할 수 있다.

[0057] 도 8은 일 예시에 따른 작업 그룹들에 순서에 관계 없는(out-of-order) 캐시 리턴들을 제공하기 위한 방법(800)의 흐름도이다. 비록 도 1 내지 도 6에 도시되고 설명된 시스템과 관련하여 설명되었지만, 임의의 기술적으로 실행 가능한 순서로 방법을 수행하도록 구성된 임의의 시스템이 본 개시의 범위 내에 있다는 것이 이해되어야 한다.

[0058] 도시된 바와 같이, 방법(800)은, 캐시 시스템(140)이 유형별 순서화 큐들 중 임의의 것의 헤드에서의 엔트리가 관련 캐시 액세스 요청을 행한 웨이브프론트로 리턴할 준비가 되었는지를 결정하는, 단계(802)에서 시작한다. 판독 또는 샘플러 동작에 의해 요청된 데이터가 캐시 시스템의 캐시 메모리(404)에 저장되면 판독 또는 샘플러 동작이 준비된다. 일부 예시에서, 기록될 데이터가 다른 메모리 트랜잭션들에 대해 적절히 순서화되도록 보장되었음을 나타내는 확인 응답이 수신되면, 그 시점에서 다른 메모리 트랜잭션들은 기록의 효과를 "볼"수 있으므로, 그 기록이 준비된다. 다른 예시에서, 기록될 데이터가 캐시 계층에서 캐시 시스템(140)의 캐시 메모리(404)보다 적어도 하나 이상의 레벨에 하나 이상의 메모리들에 저장되었음을 나타내는 확인 응답이 수신되면 기록이 준비된다.

[0059] 단계(804)에서, 캐시 시스템(140)은 큐의 헤드(512)에서 웨이브프론트(412)로의 리턴이 준비된 엔트리(514)를 선택한다. 다수의 큐들이 준비된 엔트리들(514)를 갖는 헤드들(512)을 갖는다면, 캐시 시스템(140)은 리턴을 위해 엔트리들(514) 중 하나를 선택한다. 이 선택은 기술적으로 가능한 임의의 중재 방식(예를 들어, 라운드

로빈)를 기반으로 할 수 있다.

[0060] 단계(806)에서, 캐시 시스템(140)은 캐시 액세스를 요청한 웨이브프론트(412)에 선택된 엔트리(514)에 대응되는 캐시 리턴을 제공한다. 판독 및 샘플러 동작들에 대해, 리턴은 요청된 데이터이다. 기록에 대해, 리턴은 확인 응답 신호이다. 단계(808)에서, 캐시 시스템(140)은 선택된 엔트리(514)를 포함하는 큐로부터 선택된 엔트리(514)를 제거하고 큐를 수정하여 헤드(512)가 다음 가장 오래된 엔트리(514)를 포인팅(pointing)한다.

[0061] 본원에 설명된 기술은 웨이브프론트(412)가 액세스 유형별 베이시스로 메모리 배리어 동작(memory barrier operation)들을 수행하는 것을 허용한다. 메모리 배리어 동작은 배리어 동작까지 모든 메모리 동작들에 대한 리턴이 수신될 때까지 웨이브프론트(412)를 정지시키는 동작이다. 액세스 유형별 베이시스 큐들 및 순서화는 웨이브프론트들(412)이 액세스 유형의 배리어 명령들을 실행할 수 있게 한다. 예를 들어, 판독-배리어는, 임의의 중요한 기록 동작 또는 샘플러 동작의 존재 여부에 관계없이, 계속 진행하기 전에 웨이브프론트(412)가 모든 중요한 판독 동작들에 대한 리턴을 대기하게 한다. 유사하게, 기록-장벽 또는 샘플러-장벽은 계속 진행하기 전에 웨이브프론트(412)가 모든 중요한 판독들 또는 중요한 샘플러들이 수신될 때까지 대기하게 한다.

[0062] 본원에서 설명된 기술은 메모리 액세스 요청들이 유형별 베이시스로 순서화되지만 캐시 액세스 유형들 사이에 순서가 변경되어 리턴될 수 있는 모드들을 제공함으로써 메모리 레이턴시를 개선한다. 따라서, 특정 유형의 캐시 액세스(예를 들어, 판독)는 다른 유형의 캐시 액세스(예를 들어, 텍스처 샘플러 동작)를 기다릴 필요가 없다. 이러한 순서화의 완화는, 예를 들어 한 유형의 레이턴시가 다른 유형의 레이턴시보다 큰 경우, 캐시 액세스 레이턴시를 줄일 수 있다. 일 예시에서, 판독 동작은 비교적 낮은 레이턴시를 가지며, 텍스처 샘플러 동작은 비교적 더 높은 레이턴시를 갖는다. 이전의 텍스처 샘플러 동작에 대한 캐시 액세스의 리턴 전에 새로운 판독 동작에 대한 캐시 리턴을 허용함으로써, 판독 동작의 레이턴시는, 새로운 판독 동작이 오래된 텍스처 샘플러 동작을 기다려야 하는 상황과 비교하여, 감소된다.

[0063] 순서가 변경된 캐시 리턴을 수행하는 방법이 제공된다. 이 방법은 복수의 리턴 순서화 큐들의 제1 리턴 순서화 큐의 헤드에서의 제1 엔트리가 웨이브프론트로의 리턴을 위해 이용 가능하다는 것을 결정하는 단계를 포함하고, 여기서 제1 엔트리는 제1 캐시 액세스 요청에 대응한다. 제1 리턴 순서화 큐는 제2 캐시 액세스 유형이 아닌 제1 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하고 복수의 리턴 순서화 큐들의 제2 리턴 순서화 큐는 제1 캐시 액세스 유형이 아닌 제2 캐시 액세스 유형의 캐시 액세스 요들청에 대한 엔트리를 저장한다. 상기 방법은 또한, 제1 캐시 액세스 요청보다 오래된 제2 순서화에서의 엔트리들에 대응되는 캐시 액세스 요청들이 웨이브프론트들로 리턴하기 위해 이용 가능하게 될 때까지 대기하지 않고서, 상기 결정에 응답하여 제1 엔트리에 대응되는 캐시 리턴을 웨이브프론트로 향하게 하는 단계를 포함한다.

[0064] 순서가 변경된 캐시 리턴을 수행하기 위한 연산 유닛이 제공된다. 연산 유닛은 캐시 시스템 및 웨이브프론트를 실행하도록 구성된 단일-명령-다중-데이터(single-instruction-multiple-data) 유닛을 포함한다. 캐시 시스템은 제1 리턴 순서화 큐 및 제2 리턴 순서화 큐를 포함하는 복수의 리턴 순서화 큐들을 저장하도록 구성되며, 여기서 제1 리턴 순서화 큐는 제2 캐시 액세스 유형이 아닌 제1 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하고, 제2 리턴 순서화 큐는 제1 캐시 액세스 유형이 아닌 제2 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하고, 캐시 시스템은 제1 리턴 순서화 큐의 헤드에서의 제1 엔트리가 웨이브프론트로의 리턴을 위해 이용 가능하다는 것을 결정하도록 구성되고, 여기서 제1 엔트리는 제1 캐시 액세스 요청에 대응되고, 캐시 시스템은, 결정에 응답하여, 제1 캐시 액세스 요청보다 오래된 제2 리턴 순서화에서의 엔트리들에 대응되는 캐시 액세스 요청들이 웨이브프론트로 리턴하기 위해 이용 가능하게 될 때까지 대기하지 않고서, 상기 제1 엔트리에 대응되는 캐시 리턴을 웨이브프론트로 향하게 하도록 구성된다.

[0065] 또한, 컴퓨터 시스템이 제공된다. 컴퓨터 시스템은 연산 유닛을 포함하는 가속 프로세싱 디바이스 및 가속 프로세싱 디바이스가 연산 유닛에서 웨이브프론트를 실행하도록 야기하는 프로세서를 포함한다. 연산 유닛은 캐시 시스템 및 웨이브프론트를 실행하도록 구성된 단일-명령-다중-데이터 유닛을 포함한다. 캐시 시스템은 제1 리턴 순서화 큐 및 제2 리턴 순서화 큐를 포함하는 복수의 리턴 순서화 큐들을 저장하도록 구성되며, 여기서 제1 리턴 순서화 큐는 제2 캐시 액세스 유형이 아닌 제1 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하고, 제2 리턴 순서화 큐는 제1 캐시 액세스 유형이 아닌 제2 캐시 액세스 유형의 캐시 액세스 요청들에 대한 엔트리들을 저장하고, 캐시 시스템은 제1 리턴 순서화 큐의 헤드에서의 제1 엔트리가 웨이브프론트로의 리턴을 위해 이용 가능하다는 것을 결정하도록 구성되고, 여기서 제1 엔트리는 제1 캐시 액세스 요청에 대응되고, 캐시 시스템은, 결정에 응답하여, 제1 캐시 액세스 요청보다 오래된 제2 리턴 순서화에서의 엔트리들에 대응되는 캐시 액세스 요청들이 웨이브프론트로 리턴하기 위해 이용 가능하게 될 때까지 대기하지 않고서, 상기 제1

엔트리에 대응되는 캐시 리턴을 웨이브프론트로 향하게 하도록 구성된다.

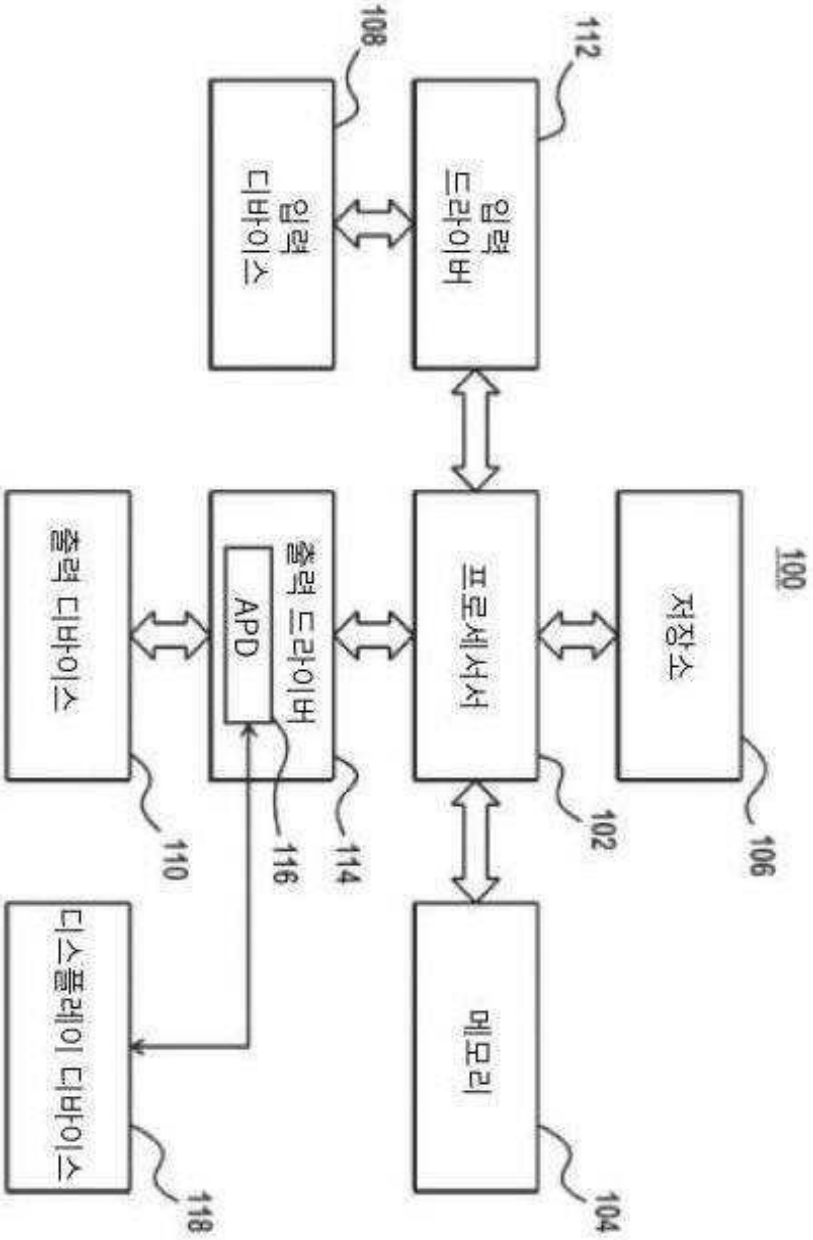
[0066] 본 명세서의 개시에 기초하여 많은 변형이 가능하다는 것을 이해해야 한다. 특징 및 요소가 특정 조합으로 상기 설명되었지만, 각 특징 또는 요소는 다른 특징 및 요소 없이 단독으로 또는 다른 특징 및 요소와 함께 또는 다른 조합 없이 다양한 조합으로 사용될 수 있다.

[0067] 제공된 방법들은 범용 컴퓨터, 프로세서 또는 프로세서 코어에서 구현될 수 있다. 적합한 프로세서는, 예로서, 범용 프로세서, 특수 목적 프로세서, 종래의 프로세서, 디지털 신호 프로세서(DSP), 복수의 마이크로 프로세서들, DSP 코어와 관련된 하나 이상의 마이크로 프로세서들, 제어기, 마이크로컨트롤러, 어플리케이션 특정 집적 회로(ASIC), 필드 프로그래머블 게이트 어레이(FPGA) 회로, 임의의 다른 유형의 집적 회로(IC), 및/또는 상태 머신을 포함한다. 이러한 프로세서는 프로세싱된 하드웨어 설명 언어(HDL) 명령어의 결과와 넷리스트(netlist)를 포함한 다른 중간 데이터를 사용하여 제조 프로세스를 구성함으로써 제조될 수 있다(그러한 명령들은 컴퓨터 판독 가능 매체 상에 저장될 수 있다). 그러한 프로세싱의 결과는 실시 예들의 양태들을 구현하는 프로세서를 제조하기 위해 반도체 제조 프로세스에서 사용되는 마스크워크(maskworks)일 수 있다.

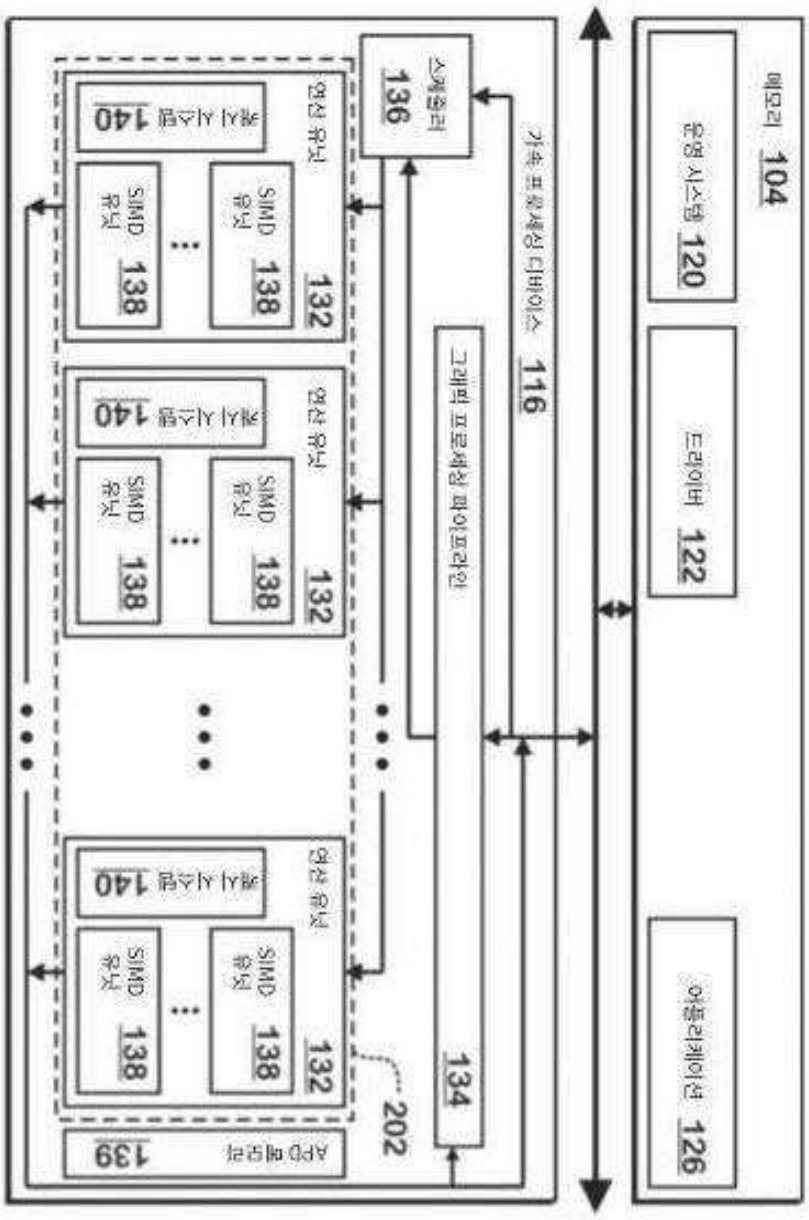
[0068] 본원에 제공된 방법 또는 흐름도는 범용 컴퓨터 또는 프로세서에 의한 실행을 위해 비-일시적 컴퓨터 판독 가능 저장 매체에 통합된 컴퓨터 프로그램, 소프트웨어 또는 펌웨어로 구현될 수 있다. 비-일시적 컴퓨터 판독 가능 저장 매체의 예시는 판독 전용 메모리(ROM), 랜덤 액세스 메모리(RAM), 레지스터, 캐시 메모리, 반도체 메모리 디바이스, 내부 하드 디스크 및 이동식 디스크와 같은 자기 매체, 광 자기 매체 및 CD-ROM 디스크 및 DVD(digital versatile disk)와 같은 광학 매체를 포함한다.

도면

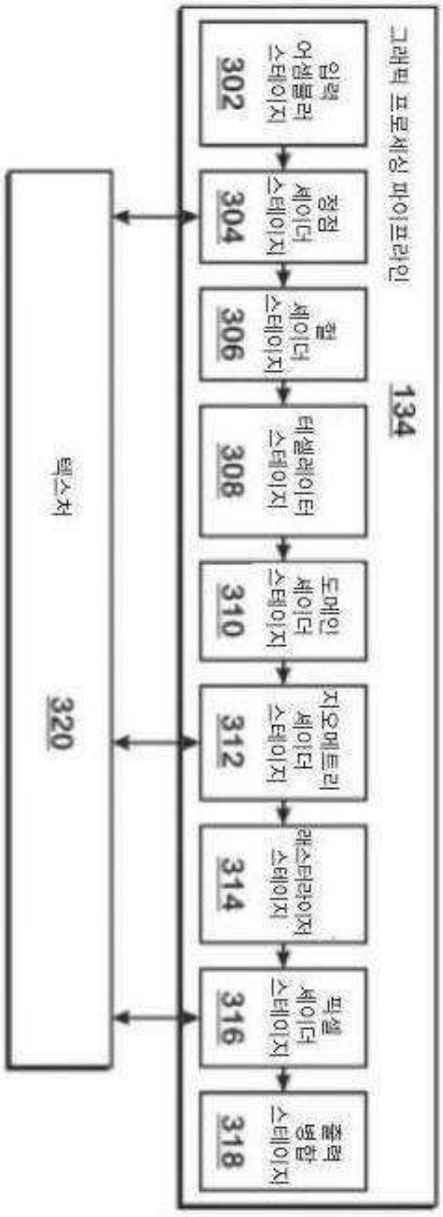
도면1


100
프로세서
102
메모리
104
저장소
106
입력 디바이스
108
입력 드라이버
112
출력 디바이스
110
출력 드라이버
114
APD
116
디스플레이 디바이스
118

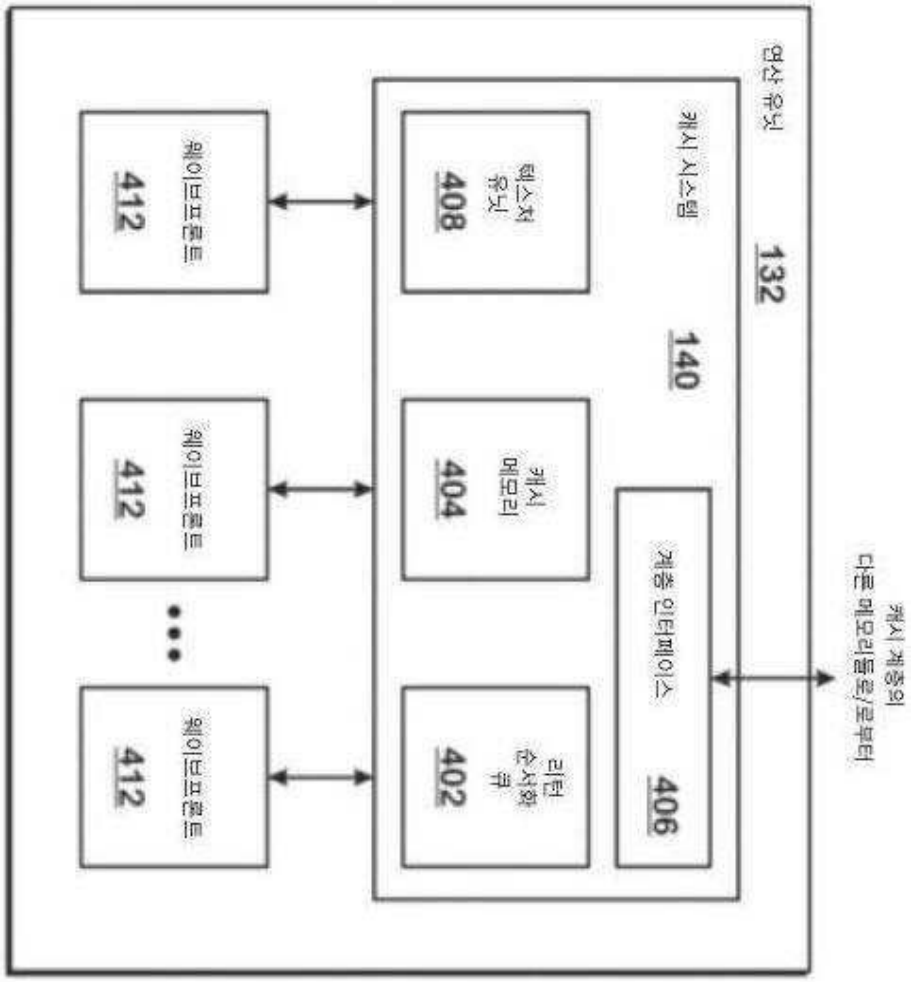
도면2


메모리 104
운영 시스템 120
드라이버 122
어플리케이션 126
가속 프로세싱 디바이스 116
스케줄러 136
그래픽 프로세싱 파이프라인 134
연산 유닛 132
SIMD 유닛 138
로컬 데이터 스토리지 140
APD 메모리 139
202

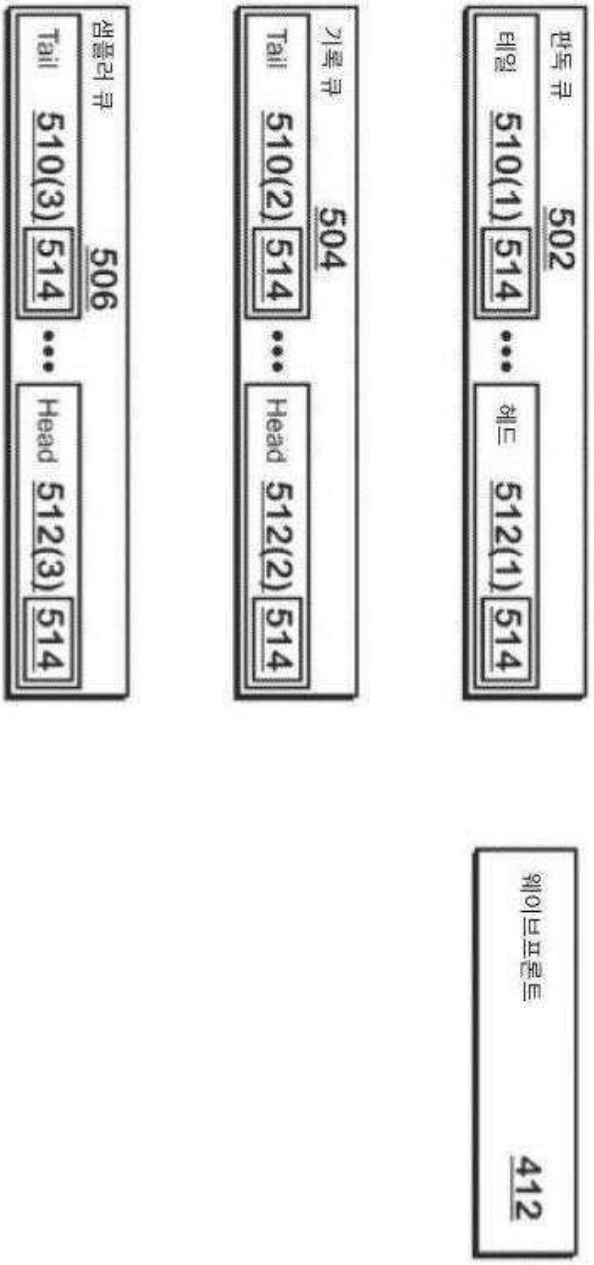
도면3


그래픽 프로세싱 파이프라인
134
입력 어셈블러 스테이지
302
정점 셰이더 스테이지
304
헐 셰이더 스테이지
306
테셀레이터 스테이지
308
도메인 셰이더 스테이지
310
지오메트리 셰이더 스테이지
312
래스터라이저 스테이지
314
픽셀 셰이더 스테이지
316
출력 병합 스테이지
318
텍스처
320

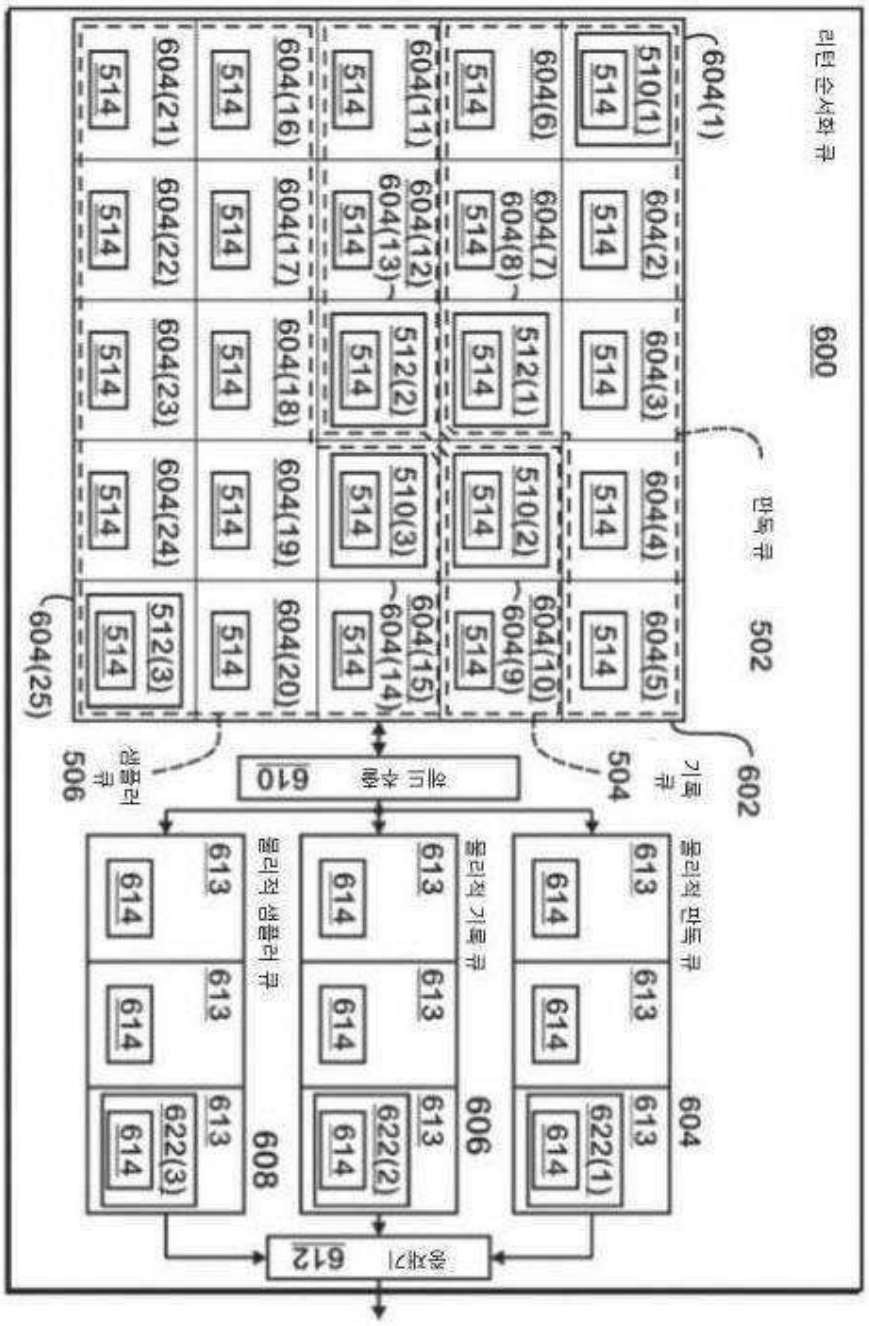
도면4


연산 유닛
132
캐시 시스템
140
텍스처 유닛
408
캐시 메모리
404
리턴 순서화 큐
402
계층 인터페이스
406
웨이브프론트
412
웨이브프론트
412
웨이브프론트
412
캐시 계층의 다른 메모리들로/로부터

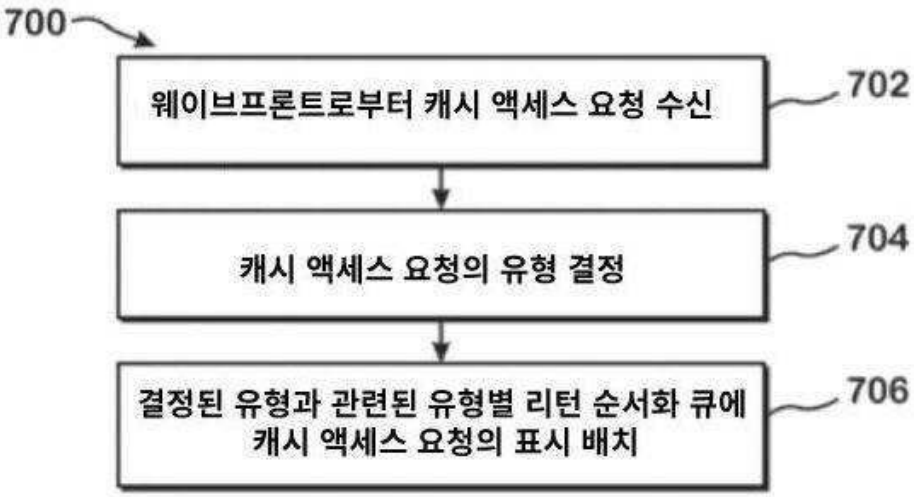
도면5


판독 큐
502
태일
510(1)
514
•••
헤드
512(1)
514
기록 큐
504
Tail
510(2)
514
•••
Head
512(2)
514
샘플러 큐
506
Tail
510(3)
514
•••
Head
512(3)
514
웨이브폼 트
412

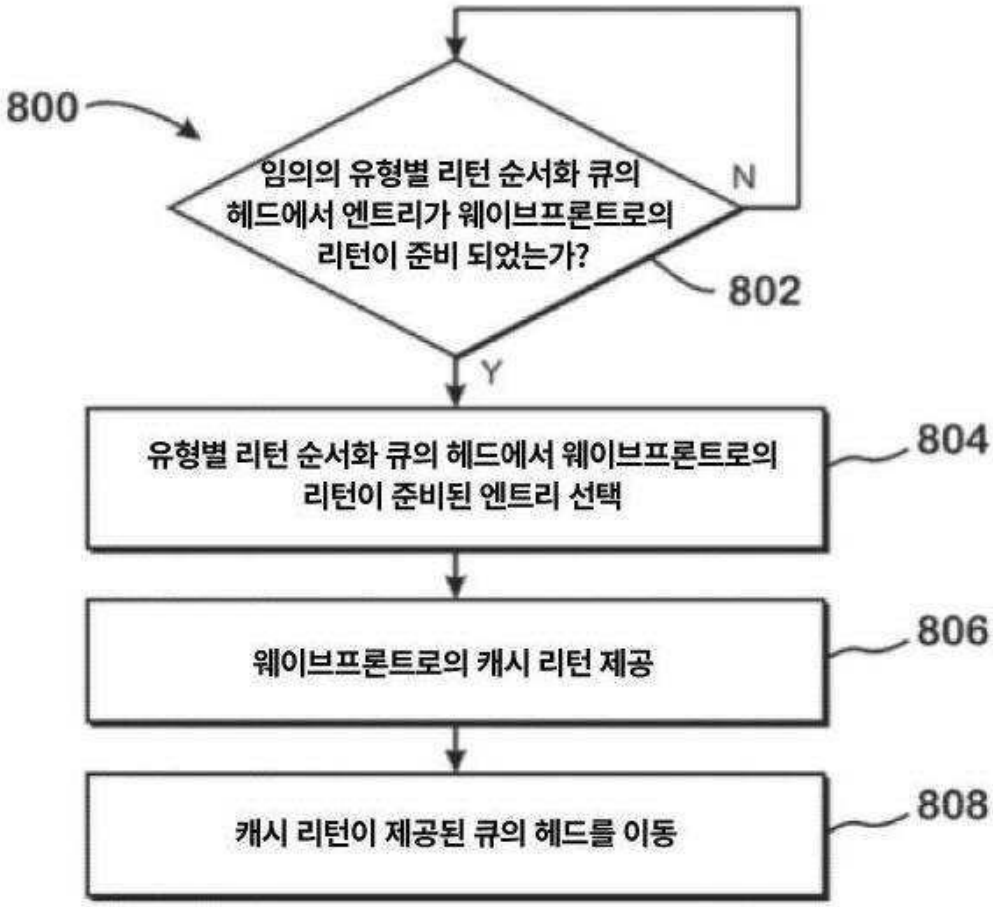
도면6


600
리턴 순서화 큐
604(1)
510(1)
514
604(2)
604(3)
604(4)
604(5)
604(6)
604(7)
604(8)
512(1)
604(9)
604(10)
604(11)
604(12)
604(13)
512(2)
604(14)
510(3)
604(15)
604(16)
604(17)
604(18)
604(19)
604(20)
604(21)
604(22)
604(23)
604(24)
512(3)
604(25)
502
펜딩 큐
602
기록 큐
504
샘플러 큐
506
610
할당 유닛
613
614
622(1)
622(2)
622(3)
604
606
608
612
할당기
물리적 펜딩 큐
물리적 기록 큐
물리적 샘플러 큐

도면7


700
웨이브프론트로부터 캐시 액세스 요청 수신
702
캐시 액세스 요청의 유형 결정
704
결정된 유형과 관련된 유형별 리턴 순서화 큐에 캐시 액세스 요청의 표시 배치
706

도면8


800
임의의 유형별 리턴 순서화 큐의 헤드에서 엔트리가 웨이브프론트로의 리턴이 준비 되었는가?
N
802
Y
유형별 리턴 순서화 큐의 헤드에서 웨이브프론트로의 리턴이 준비된 엔트리 선택
804
웨이브프론트로의 캐시 리턴 제공
806
캐시 리턴이 제공된 큐의 헤드를 이동
808