



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년10월23일
 (11) 등록번호 10-1563070
 (24) 등록일자 2015년10월19일

(51) 국제특허분류(Int. Cl.)
G06T 1/60 (2006.01)
 (21) 출원번호 10-2013-7023665
 (22) 출원일자(국제) 2012년02월10일
 심사청구일자 2013년09월06일
 (85) 번역문제출일자 2013년09월06일
 (65) 공개번호 10-2013-0135309
 (43) 공개일자 2013년12월10일
 (86) 국제출원번호 PCT/US2012/024760
 (87) 국제공개번호 WO 2012/109619
 국제공개일자 2012년08월16일
 (30) 우선권주장
 13/024,579 2011년02월10일 미국(US)
 (56) 선행기술조사문헌
 WO1993004462 A1*
 US20060001663 A1*
 JP11167523 A
 KR1020000048739 A
 *는 심사관에 의하여 인용된 문헌

(73) 특허권자
켈컴 인코퍼레이티드
 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
 (72) 발명자
샤프 콜린
 미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
페퍼 재커리 애런
 미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
 (뒷면에 계속)
 (74) 대리인
특허법인코리아나

전체 청구항 수 : 총 18 항

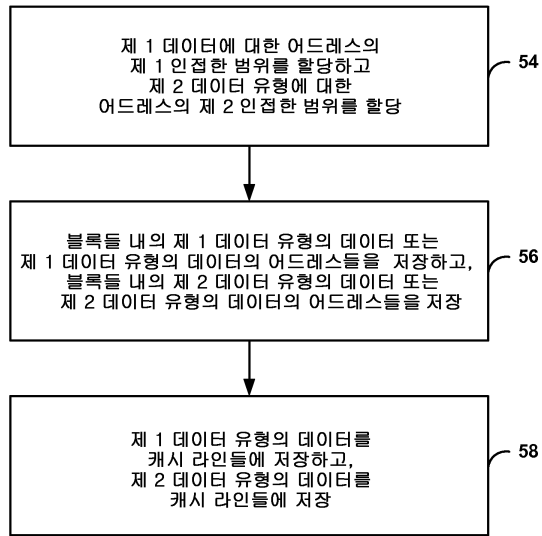
심사관 : 이상래

(54) 발명의 명칭 **그래픽 프로세싱을 위한 데이터 저장 어드레스 할당**

(57) 요약

일반적으로, 본 개시물의 양태들은 그래픽 프로세싱을 위한 각종 데이터 유형들의 데이터의 효율적인 저장을 위한 예시의 기법들을 설명한다. 일부 예들에서, 프로세싱 유닛은 제 1 및 제 2 데이터 유형에 대한 어드레스들의 제 1 및 제 2 인접한 범위를 각각 할당할 수도 있다. 프로세싱 유닛은 블록들의 어드레스들이 각각 어드레스들의 제 1 및 제 2 인접한 범위 내에 있는 상기 블록들 내의 제 1 또는 제 2 데이터 유형의 그래픽 데이터, 또는 제 1 또는 제 2 데이터 유형의 그래픽 데이터의 어드레스들을 저장할 수도 있다. 프로세싱 유닛은 제 1 데이터 유형의 그래픽 데이터 및 제 2 데이터 유형의 그래픽 데이터를 캐시의 캐시 라인들에 저장할 수도 있다.

대표도 - 도4



(72) 발명자

메츠 에두아르두스 에이

미국 92121 캘리포니아주 샌디에고 모어하우스 드
라이브 5775

리블 모리스

미국 92121 캘리포니아주 샌디에고 모어하우스 드
라이브 5775

명세서

청구범위

청구항 1

프로세싱 유닛을 이용하여, 그래픽 프로세싱을 위한 복수의 데이터 유형들 중 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위를 할당하고, 상기 그래픽 프로세싱을 위한 복수의 데이터 유형들 중 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당하는 단계로서, 상기 제 1 및 제 2 데이터 유형들은 상이한 데이터 유형들이고, 상기 복수의 데이터 유형들은 텍스처 데이터, 정점 데이터, 명령들, 상수들, 및 픽셀 데이터 중 적어도 하나를 포함하는, 상기 할당하는 단계;

상기 프로세싱 유닛을 이용하여, 블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 내의 상기 제 1 데이터 유형의 그래픽 데이터 또는 상기 제 1 데이터 유형의 상기 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하는 단계;

상기 프로세싱 유닛을 이용하여, 블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들 내의 상기 제 2 데이터 유형의 그래픽 데이터 또는 상기 제 2 데이터 유형의 상기 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하는 단계;

공통 메모리 캐시의 복수의 캐시 라인들의 제 1 세트의 각각의 제 1 필드에, 상기 어드레스들의 제 1 인접한 범위 내의 어드레스를 저장하고, 상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 상기 제 1 세트의 제 2 필드에, 상기 제 1 데이터 유형의 상기 그래픽 데이터를 저장하는 단계;

상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 제 2 세트의 각각의 제 1 필드에, 상기 어드레스들의 제 2 인접한 범위 내의 어드레스를 저장하고, 상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 상기 제 2 세트의 제 2 필드에, 상기 제 2 데이터 유형의 상기 그래픽 데이터를 저장하는 단계;

상기 제 1 데이터 유형의 상기 그래픽 데이터가 변한 경우, 상기 복수의 캐시 라인들의 상기 제 1 및 제 2 세트들 각각의 제 1 필드에 저장된 어드레스들을 상기 어드레스들의 제 1 인접한 범위에 비교하여, 어떤 캐시 라인들이 상기 제 1 데이터 유형의 그래픽 데이터를 저장하는지를 결정하는 단계; 및

상기 제 1 데이터 유형의 그래픽 데이터가 현재 데이터가 아님을 나타내기 위해 상기 제 1 데이터 유형의 그래픽 데이터를 저장하는 것으로 결정된 캐시 라인들을 무효화하는 단계

를 포함하는, 방법.

청구항 2

제 1 항에 있어서,

상기 어드레스들의 제 1 및 제 2 인접한 범위들은 저장 디바이스 어드레스들의 제 1 및 제 2 인접한 범위들 및 입/출력 메모리 관리 유닛 (input/output memory management unit; IOMMU) 어드레스들의 제 1 및 제 2 인접한 범위들 중 적어도 하나를 포함하는, 방법.

청구항 3

제 1 항에 있어서,

블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 및 블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들은 저장 디바이스 내의 저장 블록들 및 입/출력 메모리 관리 유닛 (IOMMU) 내의 어드레스 블록들 중 적어도 하나를 포함하는, 방법.

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

삭제

청구항 9

제 1 항에 있어서,

상기 제 1 데이터 유형의 그래픽 데이터 및 상기 제 2 데이터 유형의 그래픽 데이터 중 적어도 하나에 대한 요청을 수신하는 단계로서, 상기 요청은 상기 제 1 데이터 유형 또는 상기 제 2 데이터 유형에 대한 어드레스를 포함하는, 상기 요청을 수신하는 단계;

상기 제 1 데이터 유형 또는 제 2 데이터 유형에 대한 상기 어드레스가 상기 어드레스의 제 1 인접한 범위 또는 상기 어드레스의 제 2 인접한 범위 내에 있는지 여부를 결정하는 단계; 및

상기 결정에 기초하여 상기 요청을 프로세싱하는 단계를 더 포함하는, 방법.

청구항 10

복수의 캐시 라인들을 포함하는 공통 메모리 캐시; 및

프로세싱 유닛을 포함하고,

상기 프로세싱 유닛은,

그래픽 프로세싱을 위한 복수의 데이터 유형들 중 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위를 할당하고, 상기 그래픽 프로세싱을 위한 복수의 데이터 유형들 중 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당하되, 상기 제 1 및 제 2 데이터 유형들은 상이한 데이터 유형들이고, 상기 복수의 데이터 유형들은 텍스처 데이터, 정점 데이터, 명령들, 상수들, 및 픽셀 데이터 중 적어도 하나를 포함하고;

블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 내의 상기 제 1 데이터 유형의 그래픽 데이터 또는 상기 제 1 데이터 유형의 상기 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하고, 블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들 내의 상기 제 2 데이터 유형의 그래픽 데이터 또는 상기 제 2 데이터 유형의 상기 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하고;

공통 메모리 캐시의 복수의 캐시 라인들의 제 1 세트의 각각의 제 1 필드에, 상기 어드레스들의 제 1 인접한 범위 내의 어드레스를 저장하고, 상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 상기 제 1 세트의 제 2 필드에, 상기 제 1 데이터 유형의 상기 그래픽 데이터를 저장하고;

상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 제 2 세트의 각각의 제 1 필드에, 상기 어드레스들의 제 2 인접한 범위 내의 어드레스를 저장하고, 상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 상기 제 2 세트의 제 2 필드에, 상기 제 2 데이터 유형의 상기 그래픽 데이터를 저장하고;

상기 제 1 데이터 유형의 상기 그래픽 데이터가 변환 경우, 상기 복수의 캐시 라인들의 상기 제 1 및 제 2 세트들 각각의 제 1 필드에 저장된 어드레스들을 상기 어드레스들의 제 1 인접한 범위에 비교하여, 어떤 캐시 라인들이 상기 제 1 데이터 유형의 그래픽 데이터를 저장하는지를 결정하며; 그리고

상기 제 1 데이터 유형의 그래픽 데이터가 현재 데이터가 아님을 나타내기 위해 상기 제 1 데이터 유형의 그래픽 데이터를 저장하는 것으로 결정된 캐시 라인들을 무효화하도록 구성된, 장치.

청구항 11

제 10 항에 있어서,

상기 프로세싱 유닛은 프로세서 및 그래픽 프로세싱 유닛 (GPU) 중 적어도 하나를 포함하는, 장치.

청구항 12

제 10 항에 있어서,

상기 어드레스들의 제 1 및 제 2 인접한 범위들은 저장 디바이스 어드레스들의 제 1 및 제 2 인접한 범위들 및 입/출력 메모리 관리 유닛 (input/output memory management unit; IOMMU) 어드레스들의 제 1 및 제 2 인접한 범위들 중 적어도 하나를 포함하는, 장치.

청구항 13

제 10 항에 있어서,

저장 디바이스를 더 포함하고,

블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록 및 블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록은 상기 저장 디바이스 내의 저장 블록들을 포함하는, 장치.

청구항 14

제 10 항에 있어서,

입/출력 메모리 관리 유닛 (IOMMU) 을 더 포함하고,

블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 및 블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들은 상기 IOMMU 내의 어드레스 블록들을 포함하는, 장치.

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

제 10 항에 있어서,

상기 프로세싱 유닛은 상기 제 1 데이터 유형의 그래픽 데이터 및 상기 제 2 데이터 유형의 그래픽 데이터 중 적어도 하나에 대한 요청을 수신하고,

상기 요청은 상기 제 1 데이터 유형 또는 상기 제 2 데이터 유형에 대한 어드레스를 포함하고,

상기 프로세싱 유닛은 상기 제 1 데이터 유형 또는 제 2 데이터 유형에 대한 상기 어드레스가 상기 어드레스의

제 1 인접한 범위 또는 상기 어드레스의 제 2 인접한 범위 내에 있는지 여부를 결정하며;
 상기 프로세싱 유닛은 상기 결정에 기초하여 상기 요청을 프로세싱하는, 장치.

청구항 21

명령들을 포함하는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 하나 이상의 프로세싱 유닛들로 하여금,

그래픽 프로세싱을 위한 복수의 데이터 유형들 중 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위를 할당하고, 상기 그래픽 프로세싱을 위한 복수의 데이터 유형들 중 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당하게 하되, 상기 제 1 및 제 2 데이터 유형들은 상이한 데이터 유형들이고, 상기 복수의 데이터 유형들은 텍스처 데이터, 정점 데이터, 명령들, 상수들, 및 픽셀 데이터 중 적어도 하나를 포함하고;

블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 내의 상기 제 1 데이터 유형의 그래픽 데이터 또는 상기 제 1 데이터 유형의 상기 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하게 하고;

블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들 내의 상기 제 2 데이터 유형의 그래픽 데이터 또는 상기 제 2 데이터 유형의 상기 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하게 하고;

공동 메모리 캐시의 복수의 캐시 라인들의 제 1 세트의 각각의 제 1 필드에, 상기 어드레스들의 제 1 인접한 범위 내의 어드레스를 저장하고, 상기 공동 메모리 캐시의 상기 복수의 캐시 라인들의 상기 제 1 세트의 제 2 필드에, 상기 제 1 데이터 유형의 상기 그래픽 데이터를 저장하게 하고;

상기 공동 메모리 캐시의 상기 복수의 캐시 라인들의 제 2 세트의 각각의 제 1 필드에, 상기 어드레스들의 제 2 인접한 범위 내의 어드레스를 저장하고, 상기 공동 메모리 캐시의 상기 복수의 캐시 라인들의 상기 제 2 세트의 제 2 필드에, 상기 제 2 데이터 유형의 상기 그래픽 데이터를 저장하게 하고;

상기 제 1 데이터 유형의 상기 그래픽 데이터가 변환된 경우, 상기 복수의 캐시 라인들의 상기 제 1 및 제 2 세트들 각각의 제 1 필드에 저장된 어드레스들을 상기 어드레스들의 제 1 인접한 범위에 비교하여, 어떤 캐시 라인들이 상기 제 1 데이터 유형의 그래픽 데이터를 저장하는지를 결정하게 하며; 그리고

상기 제 1 데이터 유형의 그래픽 데이터가 현재 데이터가 아님을 나타내기 위해 상기 제 1 데이터 유형의 그래픽 데이터를 저장하는 것으로 결정된 캐시 라인들을 무효화하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 22

제 21 항에 있어서,

상기 어드레스들의 제 1 및 제 2 인접한 범위들은 저장 디바이스 어드레스들의 제 1 및 제 2 인접한 범위들 및 입/출력 메모리 관리 유닛 (input/output memory management unit; IOMMU) 어드레스들의 제 1 및 제 2 인접한 범위들 중 적어도 하나를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 23

제 21 항에 있어서,

블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 및 블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들은 저장 디바이스 내의 저장 블록들 및 입/출력 메모리 관리 유닛 (IOMMU) 내의 어드레스 블록들 중 적어도 하나를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

제 21 항에 있어서,

상기 하나 이상의 프로세싱 유닛들로 하여금 상기 제 1 데이터 유형의 그래픽 데이터 및 상기 제 2 데이터 유형의 그래픽 데이터 중 적어도 하나에 대한 요청을 수신하게 하는 명령들로서, 상기 요청은 상기 제 1 데이터 유형 또는 상기 제 2 데이터 유형에 대한 어드레스를 포함하는, 상기 요청을 수신하게 하는 명령들;

상기 하나 이상의 프로세싱 유닛들로 하여금 상기 제 1 데이터 유형 또는 제 2 데이터 유형에 대한 어드레스가 상기 어드레스의 제 1 인접한 범위 또는 상기 어드레스의 제 2 인접한 범위 내에 있는지 여부를 결정하게 하는 명령들; 및

상기 하나 이상의 프로세싱 유닛들로 하여금 상기 결정에 기초하여 상기 요청을 프로세싱하게 하는 명령들을 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 30

그래픽 프로세싱을 위한 복수의 데이터 유형들 중 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위를 할당하고, 상기 그래픽 프로세싱을 위한 복수의 데이터 유형들 중 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당하기 위한 수단으로서, 상기 제 1 및 제 2 데이터 유형들은 상이한 데이터 유형들이고, 상기 복수의 데이터 유형들은 텍스처 데이터, 정점 데이터, 명령들, 상수들, 및 픽셀 데이터 중 적어도 하나를 포함하는, 상기 할당하기 위한 수단;

블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 내의 상기 제 1 데이터 유형의 그래픽 데이터 또는 상기 제 1 데이터 유형의 상기 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하기 위한 수단;

블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들 내의 상기 제 2 데이터 유형의 그래픽 데이터 또는 상기 제 2 데이터 유형의 상기 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하기 위한 수단;

공통 메모리 캐시의 복수의 캐시 라인들의 제 1 세트의 각각의 제 1 필드에, 상기 어드레스들의 제 1 인접한 범위 내의 어드레스를 저장하고, 상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 상기 제 1 세트의 제 2 필드에, 상기 제 1 데이터 유형의 상기 그래픽 데이터를 저장하기 위한 수단;

상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 제 2 세트의 각각의 제 1 필드에, 상기 어드레스들의 제 2 인접한 범위 내의 어드레스를 저장하고, 상기 공통 메모리 캐시의 상기 복수의 캐시 라인들의 상기 제 2 세트의 제 2 필드에, 상기 제 2 데이터 유형의 상기 그래픽 데이터를 저장하기 위한 수단;

상기 제 1 데이터 유형의 상기 그래픽 데이터가 변환 경우, 상기 복수의 캐시 라인들의 상기 제 1 및 제 2 세트들 각각의 제 1 필드에 저장된 어드레스들을 상기 어드레스들의 제 1 인접한 범위에 비교하여, 어떤 캐시 라인들이 상기 제 1 데이터 유형의 그래픽 데이터를 저장하는지를 결정하기 위한 수단; 및

상기 제 1 데이터 유형의 그래픽 데이터가 현재 데이터가 아님을 나타내기 위해 상기 제 1 데이터 유형의 그래픽 데이터를 저장하는 것으로 결정된 캐시 라인들을 무효화하기 위한 수단을 포함하는, 장치.

청구항 31

제 30 항에 있어서,

상기 어드레스들의 제 1 및 제 2 인접한 범위들은 저장 디바이스 어드레스들의 제 1 및 제 2 인접한 범위들 및 입/출력 메모리 관리 유닛 (input/output memory management unit; IOMMU) 어드레스들의 제 1 및 제 2 인접한 범위들 중 적어도 하나를 포함하는, 장치.

청구항 32

제 30 항에 있어서,

블록들의 어드레스들이 상기 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 및 블록들의 어드레스들이 상기 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들은 저장 디바이스 내의 저장 블록들 및 입/출력 메모리 관리 유닛 (IOMMU) 내의 어드레스 블록들 중 적어도 하나를 포함하는, 장치.

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

제 30 항에 있어서,

상기 제 1 데이터 유형의 그래픽 데이터 및 상기 제 2 데이터 유형의 그래픽 데이터 중 적어도 하나에 대한 요청을 수신하기 위한 수단으로서, 상기 요청은 상기 제 1 데이터 유형 또는 상기 제 2 데이터 유형에 대한 어드레스를 포함하는, 상기 요청을 수신하기 위한 수단;

상기 제 1 데이터 유형 또는 제 2 데이터 유형에 대한 상기 어드레스가 상기 어드레스의 제 1 인접한 범위 또는 상기 어드레스의 제 2 인접한 범위 내에 있는지 여부를 결정하기 위한 수단; 및

상기 결정에 기초하여 상기 요청을 프로세싱하기 위한 수단을 더 포함하는, 장치.

발명의 설명

기술 분야

[0001] 본 개시물은 데이터 저장, 및 보다 구체적으로는 그래픽 프로세싱을 위한 데이터 저장 어드레스들의 할당에 관한 것이다.

배경 기술

[0002] 일반적으로, 시각 표현용 콘텐츠를 제공하는 디바이스는 그래픽 프로세싱 유닛 (GPU) 을 포함한다. GPU 는 디스플레이 상에 이미지 콘텐츠를 나타내는 픽셀들을 프로세싱 및 렌더링한다. 디스플레이 상의 픽셀들을 렌더링하기 위해, GPU 는 각종 데이터 유형들을 프로세싱한다. 각종 유형의 데이터는 하나 이상의 데이터 저장 디바이스들에 저장된다. GPU 는 하나 이상의 저장 디바이스들로부터 데이터의 취출하고, 이 데이터를 프로세싱하여 디스플레이 상의 픽셀들을 렌더링한다.

발명의 내용

과제의 해결 수단

- [0003] 일반적으로, 본 개시물은 그래픽 프로세싱을 위해 각종 데이터 유형들의 데이터의 효율적인 저장에 대한 기법들을 설명한다. 그래픽 프로세싱 유닛 (GPU) 은 각종 데이터 유형들의 데이터를 프로세싱하여 디스플레이를 위한 이미지를 렌더링한다. 프로세서는 각종 데이터 유형들의 데이터를 저장 디바이스에 저장하고, 각종 데이터 유형들의 데이터가 저장 디바이스에 저장되는 세그먼트들을 정의하거나 각종 데이터 유형들의 데이터에 대한 어드레스들이 저장 디바이스에 저장되는 세그먼트들을 한정할 수도 있다. 각각의 세그먼트는 인접한 어드레스들에 의해 액세스가능한 복수의 블록들을 포함할 수도 있다.
- [0004] 디스플레이는 또한, 공통 메모리 캐시를 포함할 수도 있다. 공통 메모리 캐시는 각종 데이터 유형들의 데이터를 저장할 수도 있다. GPU 는 공통 메모리 캐시로부터 데이터를 신속하게 취출할 수도 있다. 일부 예들에서, GPU 및 프로세서는 공통 메모리 캐시로부터 데이터를 신속하게 취출할 수도 있다. 본 개시물에 설명된 예시의 구현들 중 일부에서, 공통 메모리 캐시는 그래픽 프로세싱을 위해 각종 데이터 유형들 모두에 대한 데이터를 저장할 수도 있다. 예를 들어, 공통 메모리 캐시는 제 1 데이터 유형의 데이터, 및 제 2 데이터 유형의 데이터를 저장할 수도 있고, 여기서 제 1 및 제 2 데이터 유형들은 그래픽 프로세싱을 위한 상이한 데이터 유형들이다.
- [0005] 일 예에서, 본 개시물은 프로세싱 유닛을 이용하여, 그래픽 프로세싱을 위한 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위를 할당하고, 그래픽 프로세싱을 위한 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당하는 단계로서, 상기 제 1 및 제 2 데이터 유형들은 상이한 데이터 유형들인, 할당하는 단계, 프로세싱 유닛을 이용하여, 블록들의 어드레스들이 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 내의 제 1 데이터 유형의 그래픽 데이터 또는 제 1 데이터 유형의 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하는 단계, 프로세싱 유닛을 이용하여, 블록들의 어드레스들이 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들 내의 제 2 데이터 유형의 그래픽 데이터 또는 제 2 데이터 유형의 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하는 단계, 및 제 1 데이터 유형의 그래픽 데이터 및 제 2 데이터 유형의 그래픽 데이터를 공통 메모리 캐시의 복수의 캐시 라인들에 저장하는 단계를 포함한다.
- [0006] 다른 예에서, 본 개시물은 복수의 캐시 라인들을 포함하는 공통 메모리 캐시, 및 그래픽 프로세싱을 위한 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위를 할당하고, 그래픽 프로세싱을 위한 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당하고, 블록들의 어드레스들이 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 내의 제 1 데이터 유형의 그래픽 데이터 또는 제 1 데이터 유형의 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하고, 블록들의 어드레스들이 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들 내의 제 2 데이터 유형의 그래픽 데이터 또는 제 2 데이터 유형의 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하고, 제 1 데이터 유형의 그래픽 데이터 및 제 2 데이터 유형의 그래픽 데이터를 공통 메모리 캐시의 복수의 캐시 라인들에 저장하도록 구성되며, 제 1 및 제 2 데이터 유형들은 상이한 데이터 유형들인 프로세싱을 유닛을 포함하는 장치를 설명한다.
- [0007] 다른 예에서, 본 개시물은 하나 이상의 프로세싱 유닛들로 하여금, 그래픽 프로세싱을 위한 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위를 할당하고, 그래픽 프로세싱을 위한 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당하게 하고, 블록들의 어드레스들이 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 내의 제 1 데이터 유형의 그래픽 데이터 또는 제 1 데이터 유형의 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하게 하고, 블록들의 어드레스들이 어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들 내의 제 2 데이터 유형의 그래픽 데이터 또는 제 2 데이터 유형의 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하게 하고, 제 1 데이터 유형의 그래픽 데이터 및 제 2 데이터 유형의 그래픽 데이터를 공통 메모리 캐시의 복수의 캐시 라인들에 저장하게 하며, 제 1 및 제 2 데이터 유형들은 상이한 데이터 유형들인 명령들을 포함하는 컴퓨터 판독가능 저장 매체를 설명한다.
- [0008] 다른 예에서, 본 개시물은, 그래픽 프로세싱을 위한 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위를 할당하고, 그래픽 프로세싱을 위한 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당하기 위한 수단으로서, 상기 제 1 및 제 2 데이터 유형들은 상이한 데이터 유형들인, 할당하기 위한 수단, 블록들의 어드레스들이 어드레스들의 제 1 인접한 범위 내에 있는 상기 블록들 내의 제 1 데이터 유형의 그래픽 데이터 또는 제 1 데이터 유형의 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하기 위한 수단, 블록들의 어드레스들이

어드레스들의 제 2 인접한 범위 내에 있는 상기 블록들 내의 제 2 데이터 유형의 그래픽 데이터 또는 제 2 데이터 유형의 그래픽 데이터의 어드레스들 중 적어도 하나를 저장하기 위한 수단, 및 제 1 데이터 유형의 그래픽 데이터 및 제 2 데이터 유형의 그래픽 데이터를 공통 메모리 캐시의 복수의 캐시 라인들에 저장하기 위한 수단을 포함하는 장치를 설명한다.

[0009] 하나 이상의 예들의 상세들은 이하의 첨부 도면들 및 상세한 설명에서 설명된다. 다른 피쳐들, 목적들, 및 이점들이 설명 및 도면들로부터 그리고 청구항들로부터 명백해질 것이다.

도면의 간단한 설명

[0010] 도 1 은 본 개시물의 양태들을 구현하도록 구성될 수도 있는 디바이스를 예시하는 블록도이다.
 도 2 는 도 1 의 컴포넌트들 중 일부를 더 상세히 예시하는 블록도이다.
 도 3 은 도 1 의 컴포넌트들 중 일부를 더 상세히 예시하는 다른 블록도이다.
 도 4 는 본 개시물의 양태들을 구현하도록 구성될 수도 있는 디바이스의 예시의 동작을 예시하는 흐름도이다.
 도 5 는 공통 메모리 캐시의 복수의 캐시 라인들 중 어느 라인들이 특정 데이터 유형과 연관되는지를 결정하기 위한 예시의 기법을 예시하는 흐름도이다.
 도 6 은 프로세싱 유닛에 의해 수행된 예시의 기법을 예시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0011] 본 개시물의 양태들은 그래픽 프로세싱을 위한 각종 데이터 유형들의 그래픽 데이터의 효율적인 저장에 관련될 수도 있다. 예시의 목적을 위해, 본 개시물의 양태들은 그래픽 프로세싱을 위해 데이터가 사용되는 콘텍스트에서 설명된다. 그러나, 본 개시물의 양태들은 그래픽 프로세싱 시스템들 외의 시스템들로 확장 가능할 수도 있다. 본 개시물의 기법들은 일반적으로, 컴퓨터 그래픽 시스템들, 예컨대 비디오 또는 이미지 콘텐츠를 제공하는 데스크톱 컴퓨터들 및 랩톱 컴퓨터들, 디지털 미디어 플레이어들, 셋-톱 박스들, 모바일 비디오 수신 디바이스들을 포함하는 비디오 게이밍 콘솔들, 예컨대 모바일 전화기들, 개인 휴대 정보 단말 (PDA) 들, 비디오 디스플레이들, 모바일 비디오 화상회의 유닛들 등에 적용 가능할 수도 있다.

[0012] 디바이스 내의 그래픽 프로세싱 유닛 (GPU) 은 디바이스 상에 디스플레이되는 가시 가능한 콘텐츠를 생성하도록 각종 데이터 유형들의 그래픽 데이터를 프로세싱할 수도 있다. GPU 는 프로세싱 유닛의 일 예이다. 그래픽 프로세싱을 위한 각종 데이터 유형들은 텍스처 데이터, 정점 데이터, 명령들, 상수 (constant) 들, 및 픽셀 데이터를 포함할 수도 있으나, 이에 제한되지는 않는다. 각종 데이터 유형들의 그래픽 데이터는 디바이스의 저장 디바이스에 저장될 수도 있다. 상기에서 제공된 예들보다 더 많은 데이터 유형들이 그래픽 프로세싱을 위해 존재할 수도 있다.

[0013] 일부 비제한의 예들에서, 디바이스는 또한 입/출력 메모리 관리 유닛 (input/output memory management unit; IOMMU) 을 포함한다. IOMMU 는 저장 디바이스의 저장 블록들에 대한 가상화된 어드레스 공간을 GPU 에 제공할 수도 있다. IOMMU 는 복수의 어드레스 블록들을 포함할 수도 있다. 각각의 어드레스 블록은, 각종 데이터 유형들의 그래픽 데이터가 저장 디바이스에 저장되는 어드레스를 저장할 수도 있다. IOMMU 의 각각의 어드레스 블록은 개별적으로 GPU 에 의해 액세스 가능할 수도 있다.

[0014] 디바이스 내의 프로세서는 IOMMU 의 어드레스 공간을 복수의 세그먼트들로 프래그먼트화 (fragment) 할 수도 있다. 프로세서는 프로세싱 유닛의 일 예일 수도 있다. 각각의 세그먼트는 인접한 어드레스들로 주소지정 가능 (addressable) 할 수도 있는 복수의 어드레스 블록들을 포함할 수도 있다. 프로세서는 각각의 세그먼트를 할당하여, 특정 데이터 유형의 그래픽 데이터가 저장 디바이스에 저장되는 어드레스들을 저장할 수도 있다. 예를 들어, IOMMU 의 어드레스 공간의 제 1 세그먼트는 인접한 어드레스들 (0-15) 로 주소지정 가능한 어드레스 블록들을 포함할 수도 있고, IOMMU 의 어드레스 공간의 제 2 세그먼트는 인접한 어드레스들 (16-31) 을 갖고 주소지정 가능한 어드레스 블록들을 포함할 수도 있으며, 등등이다.

[0015] 이 예에서, 프로세서는 제 1 세그먼트의 인접한 어드레스들 (0-15) 에 의해 주소지정 가능한 어드레스 블록들을 할당하여, 그래픽 텍스처 데이터가 저장 디바이스에 저장되는 어드레스들을 저장할 수도 있다. 프로세서는 제 2 세그먼트의 인접한 어드레스들 (16-31) 에 의해 주소지정 가능한 어드레스 블록들을 할당하여, 그래픽 정점 데이터가 저장 디바이스 등에 저장되는 어드레스들을 저장할 수도 있다. 어드레스 블록들에 대한 인접한

어드레스들, 예를 들어 0-15 및 16-31 은 단지 예시의 목적을 위해 제공되며, 제한적이지 않다.

- [0016] 프로세서 및 GPU 의 관점에서, IOMMU 는 그래픽 프로세싱을 위해 각종 데이터 유형들의 그래픽 데이터를 저장하는 디바이스인 것으로 나타날 수도 있다. 예를 들어, 프로세서 또는 GPU 가 데이터를 판독 또는 기입하는 경우, 프로세서 또는 GPU 는 데이터가 IOMMU 로부터 판독되거나 IOMMU 에 기입되는 것처럼 데이터를 판독 또는 기입한다. IOMMU 는, 판독 또는 기입된 데이터가 실제로 저장 디바이스에 저장되는 맵을 유지할 수도 있다. 데이터가 실제로 저장 디바이스에 저장되는 맵은 가상 어드레스 공간으로 간주될 수도 있다.
- [0017] 일부 대안의 예들에서, 디바이스 내의 프로세서는 저장 디바이스의 저장 공간을 IOMMU 보다는, 복수의 세그먼트들로 프래그먼트화할 수도 있다. 이들 예들에서, IOMMU 는 필요하지 않을 수도 있으나, 본 개시물의 양태들은 이렇게 제한되는 것으로 고려되어서는 안 된다. 각각의 세그먼트는 인접한 어드레스들에 의해 주소지정 가능할 수도 있는 복수의 저장 블록들을 포함할 수도 있다. 프로세서는 각각의 세그먼트를 할당하여 특정 데이터 유형의 그래픽 데이터를 저장할 수도 있다. 예를 들어, 저장 디바이스의 제 1 세그먼트는 인접한 어드레스들 (0-15) 에 의해 주소지정 가능할 수도 있는 저장 블록들을 포함할 수도 있고, 저장 디바이스의 제 2 세그먼트는 인접한 어드레스들 (16-31) 에 의해 주소지정될 수도 있는 저장 블록들을 포함할 수도 있으며, 등등이다. 이 예에서, 프로세서는 제 1 세그먼트의 인접한 어드레스들 (0-15) 에 의해 주소지정 가능한 어드레스 블록들을 할당하여, 그래픽 픽셀 데이터를 저장할 수도 있다. 프로세서는 제 2 세그먼트의 인접한 어드레스들 (16-31) 에 의해 주소지정 가능한 어드레스 블록들을 할당하여, 그래픽 프로세싱을 위한 명령들을 저장할 수도 있으며, 등등이다. 저장 블록들에 대한 인접한 어드레스들, 예를 들어 0-15 및 16-31 은 단지 예시의 목적을 위해 제공되며, 제한되지 않는다.
- [0018] 디바이스는 또한, 공통 메모리 캐시를 포함할 수도 있다. 공통 메모리 캐시는 복수의 캐시 라인들을 포함할 수도 있는데, 각각의 캐시 라인은 그래픽 프로세싱을 위해 데이터 유형들 중 어느 하나에 대한 그래픽 데이터를 저장하도록 구성될 수도 있다. 예를 들어, 공통 메모리 캐시는 텍스처 데이터, 정점 데이터, 명령들, 상수들, 및 픽셀 데이터를 공통 메모리 캐시의 하나 이상의 캐시 라인들 내에 저장하도록 구성될 수도 있다. 캐시 라인은 저장을 위한 메모리의 고정된 사이즈의 블록으로서 간주될 수도 있다.
- [0019] 공통 메모리 캐시는 프로세서 또는 GPU 에 의한 신속한 액세스를 위해 그래픽 데이터를 저장할 수도 있다. 공통 메모리 캐시 내의 각각의 캐시 라인은 적어도 2 개의 필드들을 포함할 수도 있다. 제 1 필드는 IOMMU 의 어드레스 블록들 중 하나에 대한 어드레스 또는 저장 디바이스의 저장 블록들 중 하나에 대한 어드레스를 저장할 수도 있다. 제 1 필드가 IOMMU 의 어드레스 블록들 중 하나에 대한 어드레스를 저장하는 예에서, IOMMU 의 어드레스 블록은 데이터 유형의 그래픽 데이터가 저장되는 저장 디바이스 내의 어드레스를 포함할 수도 있다. 캐시 라인의 제 2 필드는 실제 그래픽 데이터를 저장할 수도 있다.
- [0020] 데이터 유형에 대한 그래픽 데이터가 변화, 즉 재기입되거나 소거되는 경우, 프로세서는 그 데이터 유형에 대한 그래픽 데이터를 저장하는 공통 메모리 캐시 내의 캐시 라인들 중 일부를 무효화할 필요가 있을 수도 있다. 캐시 라인을 무효화하기 위해, 프로세서는 그 캐시 라인의 제 2 필드에 널 (null) 데이터 값을 저장할 수도 있다. 공통 메모리 캐시 내의 캐시 라인들을 무효화하는 것은 무효화된 캐시 라인에 저장된 그래픽 데이터가 현재 데이터가 아님을 GPU 에 나타낼 수도 있다. 이는, 그래픽 데이터를 저장하는 캐시 라인들이 현재 데이터를 저장하지 않기 때문에 GPU 로 하여금 공통 메모리 캐시보다는 저장 디바이스로부터 데이터 유형에 대한 그래픽 데이터를 취출하게 할 수도 있다.
- [0021] 데이터 유형의 그래픽 데이터를 저장하는 캐시 라인들 중 일부를 무효화하기 위해, GPU 는 어느 캐시 라인들이 그 데이터 유형의 그래픽 데이터를 저장하는지를 결정할 수도 있다. 어느 캐시 라인이 그 데이터 유형의 그래픽 데이터를 저장하는지를 결정하기 위해, GPU 는 캐시 라인들 각각의 제 1 데이터 필드 정보를 얻어서, 캐시 라인들 각각의 제 1 데이터 필드가 그 데이터 유형에 대한 IOMMU 또는 저장 디바이스 내의 각각 어드레스 블록들 또는 저장 블록들에 대한 어드레스들의 할당된 인접한 범위 내에 있는 어드레스를 저장하는지 여부를 결정할 수도 있다.
- [0022] 예를 들어, 그래픽 데이터 유형이 그래픽 프로세싱을 위한 텍스처 데이터라고 가정한다. 추가로, 텍스처 데이터가 저장 디바이스 상에 저장되는 어드레스들, IOMMU 내의 인접한 어드레스들 (0-15) 의 저장을 위해 프로세서가 어드레스 블록들을 할당했다고 가정한다. 텍스처 데이터가 변하는 경우, 프로세서는 텍스처 데이터를 저장하는 공통 메모리 캐시에서 각각의 캐시 라인을 무효화할 수도 있다. GPU 는, 캐시 라인들 각각의 제 1 필드가 0-15 내에 있는 어드레스를 저장하는지 여부를 결정할 수도 있다. 캐시 라인이 0-15 내에 있는 어드레스를 저장하면, GPU 는 그 캐시 라인을 무효화할 수도 있다. 본 개시물의 양태들에서, GPU 는 그 특정 데

이터 유형에 대한 그래픽 데이터를 저장하는 캐시 라인들 중 하나 이상을 무효화할 수도 있다. GPU 는 공통 메모리 캐시에서의 다른 캐시 라인들 중 어느 것도 무효화하지 않을 수도 있다. 예를 들어, GPU 는 그 특정 데이터 유형에 대한 데이터를 저장하지 않는 캐시 라인들 중 어느 것도 무효화하지 않을 수도 있다.

[0023] 도 1 은 본 개시물의 양태들을 구현하도록 구성될 수도 있는 디바이스 (10) 를 예시하는 블록도이다. 디바이스 (10) 의 예들은, 모바일 무선 전화기들, 개인 휴대 정보 단말 (PDA) 들, 비디오 디스플레이들을 포함하는 비디오 게이밍 콘솔들, 모바일 비디오 화상회의 유닛들, 랩톱 컴퓨터들, 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 텔레비전 셋-톱 박스들, 디지털 미디어 플레이어들, 등을 포함할 수도 있으나, 이에 제한되지는 않는다. 디바이스 (10) 는 프로세서 (12), 그래픽 프로세싱 유닛 (GPU)(14), 디스플레이 (16), 디스플레이 버퍼 (18), 저장 디바이스 (20), 트랜시버 모듈 (22), 사용자 인터페이스 (24), 공통 메모리 캐시 (26), 및 입/출력 메모리 관리 유닛 (input/output memory management unit; IOMMU)(28) 을 포함할 수도 있다. 프로세서 (12) 및 GPU (14) 는 각각 프로세싱 유닛의 예들일 수도 있다.

[0024] 디바이스 (10) 는 명확함을 위해 도 1 에 도시되지 않은 추가의 모듈들 또는 유닛들을 포함할 수도 있다. 예를 들어, 디바이스 (10) 는 어느 것도 도 1 에는 도시되지 않은, 스피커 및 마이크를 포함하여, 디바이스 (10) 가 모바일 무선 전화기인 예들에서 전화 통신들을 달성할 수도 있다. 또한, 디바이스 (10) 에 도시된 각종 모듈들 및 유닛들은 디바이스 (10) 의 모든 예에서 필요하지 않을 수도 있다. 예를 들어, 사용자 인터페이스 (24) 및 디스플레이 (16) 는 디바이스 (10) 가 데스크톱 컴퓨터인 예들에서 디바이스 (10) 외부에 있을 수도 있다. 다른 예로써, IOMMU (28) 는 이하에서 더 상세히 설명되는 바와 같이 모든 예에서 필요하지 않을 수도 있다.

[0025] 프로세서 (12), GPU (14), 공통 메모리 캐시 (26), 및 IOMMU (28) 는 별개의 유닛들로서 도시되었으나, 본 개시물의 양태들이 이렇게 제한되지는 않는다. 일 예로써, GPU (14), 공통 메모리 캐시 (26), 및 IOMMU (28) 는 프로세서 (12) 내에 형성될 수도 있고, 예를 들어 하나의 프로세싱 유닛이 프로세서 (12) 및 GPU (14) 뿐만 아니라 공통 메모리 캐시 (26) 및 IOMMU (28) 를 포함할 수도 있다. 다른 예로써, 프로세서 (12) 는 IOMMU (28) 를 포함할 수도 있고 GPU (14) 는 공통 메모리 캐시 (26) 를 포함할 수도 있다. 프로세서 (12), GPU (14), 공통 메모리 캐시 (26), 및 IOMMU (28) 의 구성의 상이한 조합들이 가능할 수도 있고, 본 개시물의 양태들은 상이한 조합들을 고려한다.

[0026] 프로세싱 유닛으로서 각각 고려될 수도 있는 프로세서 (12) 및 GPU (14) 의 예들은 디지털 신호 프로세서 (DSP), 범용 마이크로프로세서, 주문형 집적 회로 (ASIC), 필드 프로그래머블 로직 어레이 (FPGA), 또는 다른 등가의 집적 회로 또는 이산 로직 회로를 포함하지만, 이에 제한되지는 않는다. 저장 디바이스 (20) 는 하나 이상의 컴퓨터 판독가능 저장 매체를 포함할 수도 있다. 저장 디바이스 (20) 의 예들은 랜덤 액세스 메모리 (RAM), 판독 전용 메모리 (ROM), 전기적으로 소거가능한 프로그래머블 판독 전용 메모리 (EEPROM), CD-ROM 또는 다른 광학 디스크 저장 디바이스, 자기 디스크 저장 디바이스, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령들이나 데이터 구조들의 형태로 원하는 프로그램 코드를 운반 또는 저장하는데 사용될 수 있고 컴퓨터 또는 프로세서에 의해 액세스될 수 있는 임의의 다른 매체를 포함하지만, 이에 제한되지는 않는다. 일부 양태들에서, 저장 디바이스 (20) 는, 프로세서 (12) 및/또는 GPU (14) 로 하여금 본 개시물에서 프로세서 (12) 및 GPU (14) 에 속하는 것으로 생각된 기능들을 수행하게 하는 명령들을 포함할 수도 있다.

[0027] 사용자 인터페이스 (24) 의 예들은, 트랙볼, 마우스, 키보드, 및 입력 디바이스들의 다른 유형들을 포함하지만, 이에 제한되지는 않는다. 사용자 인터페이스 (24) 는 또한 터치 스크린일 수도 있고, 디스플레이 (16) 의 일부로서 통합될 수도 있다. 트랜시버 모듈 (22) 은 디바이스 (10) 와 다른 디바이스 또는 네트워크 간의 무선 또는 유선 통신을 허용하도록 회로를 포함할 수도 있다. 트랜시버 모듈 (22) 은 변조기들, 복조기들, 증폭기들 및 유선 또는 무선 통신을 위한 다른 이러한 회로를 포함할 수도 있다.

[0028] 프로세서 (12) 는 하나 이상의 애플리케이션들을 실행할 수도 있다. 애플리케이션들의 예들은 웹 브라우저들, 이-메일 애플리케이션들, 스프레드시트들, 비디오 게임들, 또는 표시 (presentment) 를 위해 가시가능한 이미지들을 생성하는 다른 애플리케이션들을 포함한다. 하나 이상의 애플리케이션들은 저장 디바이스 (20) 내에 저장될 수도 있다. 일부 경우들에서, 프로세서 (12) 는 트랜시버 모듈 (22) 을 통해 하나 이상의 애플리케이션들을 다운로드할 수도 있다. 프로세서 (12) 는 사용자 인터페이스 (24) 를 통해 사용자에게 의한 선택에 기초하여 하나 이상의 애플리케이션들을 실행시킬 수도 있다. 일부 예들에서, 프로세서 (12) 는 사용자 상호작용 없이 하나 이상의 애플리케이션들을 실행시킬 수도 있다.

[0029] 프로세서 (12) 에 의해 생성된 가시 가능한 이미지들 각각은, 프리미티브 (primitive) 들로서 지칭될 수도

있는, 복수의 다각형들로 형성된 2 차원 (2-D) 또는 3 차원 (3-D) 이미지들일 수도 있다. 프로세서 (12) 는 다각형들의 정점들에 대한 좌표들을 결정할 수도 있다. 다각형의 일 예는 삼각형이지만, 다각형들이 삼각형 들에 한정되는 것으로 고려되지 않아야 한다. 예시의 목적을 위해, 본 개시물에서 예들은 삼각형들이 다각 형들의 관점에서 설명된다. 예를 들어, 프로세서 (12) 는 각각의 삼각형에 대해 3 개의 정점들에 대한 좌표 들을 결정할 수도 있다. 2-D 이미지들에 있어서, 각각의 삼각형에 대한 각 정점의 좌표들은 x 및 y 좌표를 포함할 수도 있다. 3-D 이미지들에 있어서, 각각의 삼각형에 대한 각 정점의 좌표들은 x, y, z, 및 w 좌표 를 포함할 수도 있는데, 여기서 w 좌표는 무한정으로 멀리 떨어져 있는 정점을 식별하기에 이룰 수도 있는 동 차 좌표계 (homogenous coordinate) 이다. 다각형들에 대해 결정된 정점 좌표들은 정점 데이터로서 지칭된 다. 프로세서 (12) 는 정점 데이터를 저장 디바이스 (20) 에 저장할 수도 있다.

[0030] 프로세서 (12) 는 또한, 결정된 정점들에 대한 각종 다른 속성들을 결정할 수도 있다. 예를 들어, 각각의 정점에 대해, 프로세서 (12) 는 픽셀 데이터로서 지칭되는 컬러 값들을 결정할 수도 있다. 각각의 컬러 값 은 3 또는 4 개의 컴포넌트들, 예를 들어 레드, 그린, 및 블루 컴포넌트들, 또는 레드, 그린, 블루, 및 투명 팩 터를 포함할 수도 있다. 일부 디바이스들에서 추가의 컬러 좌표들이 사용될 수도 있다. 프로세서 (12) 는 픽셀 데이터를 저장 디바이스 (20) 에 저장할 수도 있다.

[0031] 프로세서 (12) 는 또한, 텍스처 이미지를 저장 디바이스 (20) 에 저장할 수도 있다. 텍스처 이미지는 다각 형을 더 사실적으로 나타나게 만들도록 다각형들에 적용되는 이미지일 수도 있다. 텍스처 이미지는 일반적 으로, 텍스처 데이터의 2 차원 어레이일 수도 있다; 그러나, 텍스처 이미지는 또한 텍스처 데이터의 1 또는 3 차원 어레이일 수도 있다. 예시의 목적을 위해, 본 개시물의 양태들은 텍스처 데이터의 2 차원 어레이의 관 점에서 설명된다. 프로세서 (12) 는 어레이의 좌표들에 기초하여 2 차원 어레이 내에 텍스처 데이터를 저장 할 수도 있다. 어레이의 좌표들은 (u, v) 일 수도 있는데, 여기서 좌표 u 는 2 차원 어레이의 x-축을 따르 고, 좌표 v 는 2 차원 어레이의 y-축을 따른다. 예를 들어, 프로세서 (12) 는 다각형의 좌표들에 대응하는 어레이 내의 로케이션들에서 다각형에 적용되는 텍스처 이미지에 대한 텍스처 데이터를 저장 디바이스 (20) 에 저장할 수도 있다.

[0032] 정점 데이터, 픽셀 데이터, 및 텍스처 데이터는, GPU (14) 가 디스플레이 (16) 상의 이미지를 렌더링하는데 사 용할 수도 있는 그래픽 프로세싱을 위한 상이한 데이터 유형들의 예들이다. 정점 데이터, 픽셀 데이터, 및 텍스처 데이터에 추가하여, GPU (14) 는 또한, 디스플레이 (16) 상의 이미지를 렌더링하도록 그래픽 프로세싱을 위한 다른 데이터 유형들의 그래픽 데이터를 이용할 수도 있다. 일 예로써, GPU (14) 는 디스플레이 (16) 상의 이미지를 렌더링하도록 저장 디바이스 (20) 에 저장된 렌더링 명령들을 이용할 수도 있다. 저장 디바 이스 (20) 에 저장된 명령들은 그래픽 프로세싱을 위한 데이터 유형의 다른 예일 수도 있다. 다른 예로써, GPU (14) 는 디스플레이 (16) 상의 이미지를 렌더링하도록 저장 디바이스 (20) 에 저장된 상수들을 이용할 수도 있다. 저장 디바이스 (20) 에 저장된 상수들은 그래픽 프로세싱을 위한 데이터 유형의 다른 예일 수도 있다.

[0033] GPU (14) 는 각종 데이터 유형들로부터의 그래픽 데이터를 사용하는 그래픽 파이프라인을 구현하여 이미지를 렌 더링할 수도 있다. 그래픽 파이프라인은 적어도 일부 하드웨어를 통해 구현될 수도 있다. 예를 들어, 그래픽 파이프라인은 GPU (14) 상에서 실행하는 소프트웨어, GPU (14) 상에서 실행하는 펌웨어, GPU (14) 상에 형성된 하나 이상의 하드웨어 유닛들, 또는 이들의 조합으로서 구현될 수도 있다. 그래픽 파이프라인은 다 수의 컴포넌트들을 포함할 수도 있다. 예를 들어, GPU (14) 의 그래픽 파이프라인은 정점 데이터를 취출하 고 이 정점들의 좌표들을 다른 좌표계로 변환하며, 정점들에 대한 라이트 값들을 계산하는 정점 셰이더 (vertex shader) 를 포함할 수도 있다. GPU (14) 의 그래픽 파이프라인은 또한, 다각형 내에서의 픽셀들의 깊이 (z) 를 결정하는 원시 어셈블러 (primitive assembler) 를 포함할 수도 있다. GPU (14) 의 그래픽 파이프라인 은 또한, 픽셀들이 후속하여 다른 픽셀들에 의해 가려지는 경우 추가의 프로세싱으로부터 다각형들 내에서 이들 픽셀들을 제거하는 초기 깊이 테스트 엔진을 포함할 수도 있다. GPU (14) 의 그래픽 파이프라인은 또한, 다 각형의 정점들에 대한 컬러 값들에 기초하여 다각형 내의 픽셀들에 대한 컬러 값들을 인터포레이팅 (interpolate) 하는 픽셀 셰이더를 포함할 수도 있다.

[0034] 그래픽 파이프라인의 각종 컴포넌트들은 예시의 목적을 위해 제공되고, 제한하는 것으로 고려되지 않는다. 대안의 예들에서, 그래픽 파이프라인은 전술된 것들보다 더 많은 컴포넌트들을 포함할 수도 있다. 그래픽 파이프라인은 또한, 전술된 것들보다 더 적은 컴포넌트들을 포함할 수도 있다.

[0035] 일부 예들에서, GPU (14) 가 이미지를 렌더링함에 따라, GPU (14) 는 이미지의 렌더링된 부분들을 디스플레이

버퍼 (18) 로 출력할 수도 있다. 디스플레이 버퍼 (18) 는, 전체 이미지가 렌더링될 때까지 렌더링된 이미지를 일시적으로 저장할 수도 있다. 디스플레이 버퍼 (18) 는 프레임 버퍼로서 고려될 수도 있다. 디스플레이 버퍼 (18) 는 그 후, 렌더링된 이미지를 송신하여 디스플레이 (16) 상에 디스플레이되도록 할 수도 있다. 일부 대안의 예들에서, GPU (14) 는 이미지를 디스플레이 버퍼 (18) 에 일시적으로 저장하기 보다는 이미지의 렌더링된 부분들을 디스플레이를 위해 디스플레이 (16) 로 직접 출력할 수도 있다. 디스플레이 (16) 는 액정 디스플레이 (LCD), 음극선관 (CRT) 디스플레이, 플라즈마 디스플레이, 또는 디스플레이 디바이스의 다른 유형을 포함할 수도 있다.

[0036] 저장 디바이스 (20) 는 복수의 저장 블록들을 포함할 수도 있다. 각각의 저장 블록은 그래픽 프로세싱을 위해 각종 데이터 유형들의 그래픽 데이터를 저장할 수도 있다. 각각의 저장 블록은 개별적으로 주소지정 가능할 수도 있다. 일부 예들에서, 프로세서 (12) 는 저장 디바이스 (20) 에 제공된 저장 공간을 데이터 저장 세그먼트들로 프래그먼트화할 수도 있다. 각각의 세그먼트는 인접한 메모리 어드레스들에 의해 각각 주소지정 가능한 복수의 데이터 저장 블록들을 포함할 수도 있다. 따라서, 각각의 데이터 저장 세그먼트는 인접한 메모리 어드레스들의 범위에 의해 정의될 수도 있다.

[0037] 프로세서 (12) 는 특정 데이터 유형에 대한 각각의 데이터 저장 세그먼트를 할당할 수도 있다. 예를 들어, 프로세서 (12) 는 제 1 세그먼트를 할당하여 제 1 데이터 유형의 그래픽 데이터를 저장하고, 제 2 세그먼트를 할당하여 제 2 데이터 유형의 그래픽 데이터를 저장할 수도 있다. 이 예에서, 제 1 세그먼트의 저장 블록들은 인접한 어드레스들에 의해 주소지정 가능할 수도 있다. 예를 들어, 프로세서 (12) 는 제 1 세그먼트에서 블록들에 대한 인접한 어드레스들의 제 1 범위를 할당할 수도 있다. 제 1 세그먼트의 각각의 저장 블록의 어드레스들은 인접한 어드레스들의 제 1 범위 내에 있을 수도 있다. 유사하게, 제 2 세그먼트의 저장 블록의 어드레스들은 프로세서 (12) 에 의해 할당된 인접한 어드레스들의 제 2 범위 내에 있을 수도 있다.

[0038] 일부 예들에서, 디바이스 (10) 는 IOMMU (28) 를 포함할 수도 있다. IOMMU (28) 는 디바이스 (10) 의 모든 예에서 필요하지 않을 수도 있다. 예를 들어, IOMMU (28) 는, 프로세서 (12) 가 저장 디바이스 (20) 에 저장된 각종 데이터 유형들의 그래픽 데이터에 대한 인접한 어드레스들을 할당하는 예들에서 필요하지 않을 수도 있다. 그러나, 본 개시물의 양태들은 그렇게 제한되는 것으로 고려되어서는 안 된다. 디바이스 (10) 는, 프로세서 (12) 가 저장 디바이스 (20) 에 저장된 각종 데이터 유형들의 그래픽 데이터에 대한 인접한 어드레스들을 할당하는 예들에서도 IOMMU (28) 를 포함할 수도 있다.

[0039] IOMMU (28) 는 하나 이상의 하드웨어 유닛들, 하드웨어 유닛들 상에서 실행하는 소프트웨어, 하드웨어 유닛 상에서 실행하는 펌웨어, 또는 이들의 임의의 조합으로서 구현될 수도 있다. IOMMU (28) 가 하나 이상의 하드웨어 유닛들인 예들에서, IOMMU (28) 의 예들은 DSP, 범용 프로세서, ASIC, FPGA, 또는 다른 등가의 집적 또는 이산 로직 회로를 포함할 수도 있으나 이에 제한되지는 않는다.

[0040] IOMMU (28) 는 저장 디바이스 (20) 의 저장 블록들에 대한 가상 어드레스 공간을 GPU (14) 에 제공하므로, IOMMU (28) 는 예를 들어 GPU (14) 에 각종 데이터 유형들의 그래픽 데이터를 저장하는 디바이스인 것으로 나타난다. 예를 들어, IOMMU (28) 는 프로세서 (12) 및/또는 GPU (14) 에 의해 요청된 저장 디바이스 (20) 에 대한 액세스들을 핸들링할 책임이 있는 하드웨어 컴포넌트일 수도 있다. 예를 들어, IOMMU (28) 는 테이블 또는 다른 데이터 구조, 예를 들어 복수의 어드레스 블록들을 나타내는 레지스터들을 포함할 수도 있다. 각각의 어드레스 블록은 저장 디바이스 (20) 의 저장 블록들 중 하나의 어드레스를 저장할 수도 있다. 각각의 어드레스 블록은 개별적으로 주소지정 가능할 수도 있다.

[0041] 프로세서 (12) 또는 GPU (14) 가 데이터를 판독 또는 기입하기를 원하는 경우, 프로세서 (12) 또는 GPU (14) 상에서 실행하는 소프트웨어는 프로세서 (12) 또는 GPU (14) 가 그 어드레스에 의해 호출되는 어드레스 블록으로부터 판독되거나 어드레스 블록에 기입되는 것 처럼 IOMMU (28) 의 어드레스 블록들 중 하나의 어드레스를 호출한다. IOMMU (28) 는 그 후, 어느 저장 블록이 IOMMU (28) 의 어드레스 블록에 저장된 어드레스에 대응하는지를 결정할 수도 있다. IOMMU (28) 는 그 후, IOMMU (28) 의 어드레스 블록에 저장된 어드레스에 대응하는 저장 블록에 기입하거나 이로부터 판독할 수도 있다.

[0042] 명확함을 위해, 용어들 저장 디바이스 어드레스 또는 어드레스들은 저장 디바이스 (20) 내의 저장 블록들의 어드레스 또는 어드레스들을 지칭할 수도 있다. 또한, 용어들 IOMMU 어드레스 또는 어드레스들은 IOMMU (28) 내의 어드레스 블록들의 어드레스들 또는 어드레스를 지칭할 수도 있다. 전술된 바와 같이, IOMMU (28) 의 각각의 어드레스 블록은 저장 디바이스 어드레스를 저장할 수도 있다. 일반적으로, IOMMU (28) 는 IOMMU 어드레스들을 저장 디바이스 어드레스들에 맵핑할 수도 있다. 이 방식으로, 프로세서 (12) 또는 GPU (14) 가

IOMMU 어드레스를 호출하는 경우, IOMMU (28) 는 예를 들어, 테이블, 레지스터, 매트릭스 또는 다른 데이터 구조로 표시된 바와 같은, IOMMU 어드레스에 대응하는 저장 디바이스 어드레스를 결정함으로써 맵핑에 기초하여 그래픽 데이터를 저장하는 저장 블록에 대한 저장 디바이스 어드레스를 결정할 수도 있다.

[0043] 전술된 바와 같이, 프로세서 (12) 는 저장 디바이스 (20) 의 메모리 공간을 세그먼트들로 프래그먼트화할 수도 있다. 각각의 세그먼트는 인접한 저장 디바이스 어드레스에 의해 각각 주소지정 가능한 복수의 저장 블록들을 포함할 수도 있다. 일부 예들에서, 프로세서 (12) 는 IOMMU (28) 의 어드레스 공간을 세그먼트들로 프래그먼트화할 수도 있다. 각각의 세그먼트는 인접한 IOMMU 어드레스들에 의해 각각 주소지정 가능한 복수의 어드레스 블록들을 포함할 수도 있다.

[0044] 일부 예들에서, 프로세서 (12) 는 IOMMU (28) 내의 어드레스 블록들에 인접한 IOMMU 어드레스들의 제 1 범위를 할당할 수도 있다. 이들 예들에서, 인접한 IOMMU 어드레스들의 제 1 범위는, 제 1 데이터 유형의 그래픽 데이터가 저장 디바이스 (20) 에 저장되는 어드레스들을 저장하는 어드레스 블록들에 대한 것이다. 프로세서 (12) 는 또한, 인접한 IOMMU 어드레스들의 제 2 범위를 IOMMU (28) 내의 어드레스 블록들에 할당할 수도 있다. 이들 예들에서, 인접한 IOMMU 어드레스들의 제 2 범위는, 제 2 데이터 유형의 그래픽 데이터가 저장 디바이스 (20) 에 저장되는 어드레스들을 저장하는 어드레스 블록들에 대한 것이다. 프로세서 (12) 는 유사하게, 그래픽 프로세싱을 위해 각종 데이터 유형들에 대해 인접한 IOMMU 어드레스들의 범위들을 할당할 수도 있다.

[0045] 일부 경우들에서, 각각의 데이터 유형의 그래픽 데이터에 대한 인접한 저장 디바이스 어드레스들을 할당하는 것, 또는 각각의 데이터 유형의 그래픽 데이터에 대한 인접한 IOMMU 어드레스들을 할당하는 것이 유리할 수도 있다. 일 예로써, 프로세서 (12) 또는 GPU (14) 는 데이터 유형들 각각에 대한 그래픽 데이터가 저장 디바이스 (20) 에 저장되는 정확한 어드레스들을 추적할 필요가 없을 수도 있다. 프로세서 (12) 또는 GPU (14) 는 단지, 인접한 저장 디바이스 어드레스들 또는 IOMMU 어드레스들의 범위를 추적할 수도 있다. 인접한 저장 디바이스 어드레스들 또는 IOMMU 어드레스들 각각은 서브-히프 (sub-heap) 로서 고려될 수도 있다. 프로세서 (12) 는 저장 디바이스 (20) 내의 저장 블록들 또는 IOMMU (28) 내의 어드레스 블록들에, 서브-히프들, 예를 들어 인접한 저장 디바이스 어드레스들 또는 IOMMU 어드레스들의 범위들을 할당하는 할당기로서 고려될 수도 있다.

[0046] 예를 들어, GPU (14) 가 그래픽 프로세싱을 위해 텍스처 데이터를 취출하기를 원한다고 가정한다. 또한, 종래의 히프 할당기들에서 일어나는 바와 같이, 텍스처 데이터가 인접하게 주소지정 가능한 저장 블록들에 저장되지 않는다고 가정한다. 이 경우에서, GPU (14) 는 불충분하게 프로세싱될 수도 있는 텍스처 데이터의 각 데이터에 대한 저장 디바이스 어드레스들을 추적할 필요가 있다. 본 개시물의 양태들에서, GPU (14) 는 각각의 개별적인 저장 디바이스 어드레스를 추적할 필요가 없을 수도 있고, 저장 디바이스 어드레스들의 범위를 추적할 수도 있고, 이는 효율적인 프로세싱을 촉진할 수도 있다.

[0047] 다른 예로써, 각각의 데이터 유형의 그래픽 데이터에 대한 IOMMU 어드레스들 또는 인접한 저장 디바이스 어드레스를 할당하는 잠재적인 이점들을 위해, 프로세서 (12) 및 GPU (14) 는 부정확한 데이터를 취출할 가능성이 더 적을 수도 있다. 예를 들어, GPU (14) 상에서 실행하는 소프트웨어, 예컨대 셰이더 프로그램 (shader program) 은 GPU (14) 로 하여금 특정 데이터 유형의 데이터, 예를 들어 텍스처 데이터를 취출하게 할 수도 있다. 셰이더 프로그램은, 하나의 비제한의 예로써, OpenCL 버전 1.1 사양의 신택스를 사용하여 애플리케이션 개발자들에 의해 기입될 수도 있다. 그러나, 본 개시물의 양태들은 셰이더 프로그램이 OpenCL 버전 1.1 사양에 따라 기입되는 예들에 제한된 것으로 고려되어서는 안 된다. GPU (14) 상에서 실행하는 소프트웨어는 텍스처 데이터를 취출하기 위한 어드레스들을 GPU (14) 에 제공할 수도 있다. 이 예에서, 프로세서 (12) 는 텍스처 데이터에 대한 인접한 IOMMU 어드레스들의 범위를 갖고 GPU (14) 에 알려줄 수도 있다. GPU (14) 상에서 실행하는 소프트웨어는 그 후, 그것이 텍스처 데이터를 취출하기 위해 GPU (14) 에 제공하는 어드레스들이 인접한 IOMMU 어드레스들의 범위 내에 있다는 것을 보장할 수도 있다. 예를 들어, 셰이더 프로그램은 GPU (14) 가, 셰이더 프로그램이 요청했던 데이터의 유형에 대한 범위 밖에 있는 데이터를 액세스하고자 하지 않는다는 것을 보장할 수도 있다.

[0048] 도 1 에 예시된 바와 같이, 디바이스 (10) 는 또한, 공통 메모리 캐시 (26) 를 포함할 수도 있다. 일부 예들에서 공통 메모리 캐시 (26) 가 프로세서 (12) 또는 GPU (14) 외부에 있는 것으로서 예시되었으나, 공통 메모리 캐시 (26) 는 프로세싱 유닛, 예를 들어 프로세서 (12) 또는 GPU (14) 내에 형성될 수도 있고, 또는 프로세서 (12) 와 GPU (14) 양자에 부분적으로 형성될 수도 있다. 공통 메모리 캐시 (26) 는 프로세서 (12) 또는 GPU (14) 에 의한 신속한 액세스를 위해 그래픽 데이터를 저장할 수도 있다. 예를 들어, 프로세싱 유닛들,

예를 들어 프로세서 (12) 및 GPU (14) 는 저장 디바이스 (20) 로부터 그래픽 데이터를 추출하는 것보다 신속한 공통 메모리 캐시 (26) 로부터 그래픽 데이터를 추출할 수도 있다.

[0049] 공통 메모리 캐시 (26) 는 복수의 캐시 라인들을 포함할 수도 있고, 여기서 각각의 캐시 라인은 그래픽 프로세싱을 위한 그래픽 데이터 유형들 중 임의의 것에 대한 그래픽 데이터를 저장하도록 구성될 수도 있다. 예를 들어, 공통 메모리 캐시 (26) 는 텍스처 데이터, 정점 데이터, 명령들, 상수들, 및 픽셀 데이터를 공통 메모리 캐시 (26) 의 하나 이상의 캐시 라인들 내에 저장할 수도 있다. 공통 메모리 캐시 (26) 는, 공통 메모리 캐시 (26) 가 단일 데이터 유형에 대한 그래픽 데이터를 저장하는 다수의 캐시들이기 보다는 각종 데이터 유형들에 대한 그래픽 데이터를 저장하도록 구성되기 때문에 효율적인 저장을 촉진할 수도 있다. 예를 들어, 공통 메모리 캐시 (26) 가 텍스처 데이터, 픽셀 데이터, 정점 데이터, 명령들, 및 상수들을 동일한 캐시에 저장할 수 있기 때문에 디바이스 (10) 는 텍스처 데이터 캐시, 픽셀 데이터 캐시, 정점 데이터 캐시, 명령 캐시, 및 상수 캐시를 포함할 필요가 없을 수도 있다.

[0050] 캐시 라인은 저장을 위한 메모리의 고정된 사이즈의 블록으로서 고려될 수도 있다. 각각의 캐시 라인은 2 개의 필드들을 포함할 수도 있다. 제 1 필드는 저장 디바이스 어드레스 또는 IOMMU 어드레스 중 하나에 대한 어드레스, 예를 들어 저장 디바이스 (20) 의 저장 블록들 중 하나에 대한 어드레스 또는 IOMMU (28) 의 어드레스 블록들 중 하나에 대한 어드레스를 저장할 수도 있다. 제 2 필드는 각종 데이터 유형들의 실제 그래픽 데이터를 저장할 수도 있다.

[0051] 데이터 유형에 대한 그래픽 데이터가 변화, 예를 들어 재기입 또는 소거되는 경우, 프로세서 (12) 는 그 데이터 유형의 그래픽 데이터를 저장하는 공통 메모리 캐시 (26) 내의 캐시 라인들을 무효화할 필요가 있을 수도 있다. 예를 들어, 공통 메모리 캐시 (26) 내의 캐시 라인들 중 일부는 현재 텍스처 이미지에 대한 텍스처 데이터를 저장할 수도 있다. 프로세서 (12) 가 새로운 텍스처 이미지를 저장하는 경우, 프로세서 (12) 는, 텍스처 이미지가 변했기 때문에 텍스처 데이터를 저장하는 공통 메모리 캐시 (26) 내의 이들 캐시 라인들을 무효화할 필요가 있을 수도 있다. 다른 예로써, 공통 메모리 캐시 (26) 내의 캐시 라인들 중 일부는 프로세싱되고 있는 현재 다각형에 대한 정점 데이터를 저장할 수도 있다. GPU (14) 가 다른 다각형을 프로세싱할 필요가 있는 경우, 프로세서 (12) 는, 현재 다각형의 정점들이 더 이상 프로세싱되지 않기 때문에 정점 데이터를 저장하는 공통 메모리 캐시 (26) 내의 이들 캐시 라인들을 무효화할 필요가 있을 수도 있다.

[0052] 일 예로써, 캐시 라인을 무효화하기 위해, 프로세서 (12) 또는 GPU (14) 는 그 캐시 라인의 제 2 필드에 널 데이터 값을 저장할 수도 있다. 공통 메모리 캐시 (26) 내의 캐시 라인들을 무효화하는 것은 무효화된 캐시 라인에 저장된 데이터가 현재 데이터가 아니라는 것을 프로세서 (12) 및 GPU (14) 에 나타낼 수도 있다. "캐시 미스 (cache miss)" 는, 프로세서 (12) 또는 GPU (14) 가 무효화된 캐시 라인으로부터 데이터를 추출하는 경우 발생할 수도 있다. 캐시 미스가 발생하는 경우, 프로세서 (12) 또는 GPU (14) 는, 공통 메모리 캐시 (26) 에서의 무효화된 캐시 라인이 현재 데이터를 저장하지 않기 때문에 저장 디바이스 (20) 로부터 그래픽 데이터를 추출할 수도 있다. 일부 예들에서, 저장 디바이스 (20) 로부터 그래픽 데이터를 추출하는 것에 추가하여, 프로세서 (12) 또는 GPU (14) 는 또한 신속한 후속의 액세스를 위해 공통 메모리 캐시 (26) 의 캐시 라인들에 추출된 그래픽 데이터를 저장할 수도 있다.

[0053] 프로세서 (12) 또는 GPU (14) 는 변화된 데이터 유형에 기초하여 공통 메모리 캐시 (26) 의 어느 캐시 라인들이 무효화되는지를 결정할 수도 있다. 예를 들어, 텍스처 이미지가 변화하였으면, GPU (14) 는 공통 메모리 캐시 (26) 중 어느 캐시 라인들이 텍스처 데이터를 저장하는지를 결정할 수도 있다. 이 예에서, GPU (14) 는 텍스처 데이터를 저장하는 하나 이상의 캐시 라인들을 무효화할 수도 있고, 텍스처 데이터 외의 데이터 유형들에 대한 데이터를 저장하는 캐시 라인들 중 어느 것도 무효화하지 않을 수도 있다.

[0054] 이 방식으로, 프로세서 (12) 또는 GPU (14) 는 공통 메모리 캐시 (26) 의 캐시 라인들 중 일부만을 무효화하고, 특정 데이터 유형 변화들에 대한 매번의 그래픽 데이터마다 공통 메모리 캐시 (26) 의 전부를 무효화하지 않을 수도 있다. 공통 메모리 캐시 (26) 의 캐시 라인들의 전부를 무효화하지 않는 것은, 프로세서 (12) 또는 GPU (14) 가 각각 무효화 후에 공통 메모리 캐시 (26) 에 이전에 저장되어 있는 그래픽 데이터의 전부를 추출할 필요가 없을 수도 있기 때문에 효율적인 저장을 더욱 촉진할 수도 있다. 프로세서 (12) 또는 GPU (14) 는 단지 변화된 그래픽 데이터를 추출할 필요가 있고, 단지 변화된 데이터를 공통 메모리 캐시 (26) 안에 저장할 필요가 있을 수도 있다.

[0055] 변경된 데이터 유형의 그래픽 데이터를 저장하는 캐시 라인들을 무효화하기 위해, 프로세서 (12) 또는 GPU (14) 는 공통 메모리 캐시 (26) 중 어느 캐시 라인들이 데이터 유형의 그래픽 데이터를 저장하는지를 결정할 수도 있

다. 예를 들어, GPU (14) 는 캐시 라인들 각각의 제 1 데이터 필드의 정보를 얻어 캐시 라인들 각각의 제 1 데이터 필드가 그 데이터 유형의 저장 디바이스 어드레스 또는 IOMMU 어드레스들의 할당된 인접한 범위 내에 있는 어드레스는 저장하는지 여부를 결정할 수도 있다. 예를 들어, 그래픽 데이터 유형은 픽셀 데이터라고 가정한다. 추가로, 프로세서 (12) 가 IOMMU (28) 내의 어드레스 블록들, 인접한 IOMMU 어드레스들 (16-31) 을 할당했다고 가정한다. 인접한 IOMMU 어드레스들 (16-31) 은 각각, 이 예에서 픽셀 데이터가 저장되는 저장 디바이스 (20) 의 저장 블록들의 어드레스들을 저장할 수도 있다.

[0056]

GPU (14) 가 공통 메모리 캐시 (26) 에 저장된 픽셀 데이터 외의 픽셀 데이터를 액세스할 필요가 있는 경우, GPU (14) 는 저장 디바이스 (20) 로부터 픽셀 데이터를 추출할 필요가 있을 수도 있다. 새로운 추출된 픽셀 데이터를 공통 메모리 캐시에 저장하기 위해, 프로세서 (12) 는 프로세서 (12) 가 추출된 픽셀 데이터를 저장할 수 있도록 픽셀 데이터를 저장하는 공통 메모리 캐시 (26) 에서 충분한 수의 캐시 라인들을 무효화할 수도 있다. 프로세서는 캐시 라인들 각각의 제 1 필드가 16-31 내에 있는 어드레스를 저장하는지 여부를 결정할 수도 있다. 캐시 라인이 16-31 내에 있는 어드레스를 저장하면, 프로세서 (12) 는 그 캐시 라인을 무효화할 수도 있다.

[0057]

프로세서 (12) 는 인접한 저장 디바이스 어드레스들 또는 IOMMU 어드레스들을 할당할 수도 있기 때문에, 프로세서 (12) 는 캐시 라인이 특정 데이터 유형의 그래픽 데이터를 저장하는지 여부를 더 용이하게 결정할 수도 있다. 예를 들어, 저장 디바이스 어드레스들 또는 IOMMU 어드레스들이 인접하지 않았다면, 프로세서 (12) 는 그 특정 데이터 유형에 대한 IOMMU 어드레스 또는 모든 단일의 저장 디바이스 어드레스를 추적할 필요가 있다. 프로세서 (12) 는 그 후, 캐시 라인의 제 1 데이터 필드에 저장된 어드레스를 그 특정 데이터 유형에 대한 모든 단일 저장 디바이스 어드레스 또는 IOMMU 어드레스와 비교하여 캐시 라인이 그 특정 데이터 유형의 그래픽 데이터를 저장하는지 여부를 결정한다. 본 개시물의 양태들에서, 프로세서 (12) 는 개별적인 저장 디바이스 어드레스들 또는 IOMMU 어드레스들이 아닌 인접한 저장 디바이스 어드레스들 또는 IOMMU 어드레스들의 범위에 각각의 캐시 라인의 제 1 필드에 저장된 어드레스를 비교하여, 캐시 라인이 특정 데이터 유형의 그래픽 데이터를 저장하는지 여부를 결정할 수도 있다.

[0058]

도 2 는 도 1 의 컴포넌트들 중 일부를 더 상세히 예시하는 블록도이다. 예를 들어, 도 2 는 도 1 의 공통 메모리 캐시 (26) 및 저장 디바이스 (20) 를 예시한다. 도 2 에 예시된 바와 같이, 저장 디바이스 (20) 는 12 개의 저장 블록들 (0-11)(총괄하여, "저장 블록들" 로서 지칭됨) 을 포함한다. 저장 디바이스 (20) 는 12 개의 저장 블록들보다 더 많거나 더 적은 저장 블록들을 포함할 수도 있다. 일부 예들에서, 저장 디바이스 (20) 는 4 기가-바이트 (GB) 저장 디바이스일 수도 있다; 그러나, 이 개시물의 양태들은 이에 제한되지는 않는다.

[0059]

도 2 에 예시된 바와 같이, 저장 블록들 (0-11) 은 연속적으로 정렬되지 않는다. 일부 예들에서, 이는 저장 블록들 (0-11) 이 저장 디바이스 (20) 상에서 반드시 인접해야 할 필요는 없지만 저장 블록들 (0-11) 이 저장 디바이스 (20) 상에서 인접한 것이 가능할 수도 있음을 예시하는 것이다. 저장 블록들 (0-11) 의 각 블록은 그 어드레스에 의해 개별적으로 주소지정 가능할 수도 있다. 예를 들어, 도 2 는 저장 디바이스 어드레스들 (0-11) 을 예시한다. 저장 디바이스 어드레스들 (0-11) 은 세그먼트들로 프래그먼트화, 즉 분할될 수도 있고, 여기서 각 세그먼트는 저장 디바이스 어드레스들의 범위를 포함한다. 프로세서 (12) 는 저장 디바이스 어드레스들의 각 범위를 특정 데이터 유형에 할당할 수도 있다. 각각의 데이터 유형에 대해 저장 디바이스 어드레스들의 범위는 인접할 수도 있다.

[0060]

예를 들어, 도 2 에 예시된 바와 같이, 저장 디바이스 어드레스들의 범위 (32) 는 텍스처 데이터를 저장하는 저장 블록들, 예를 들어 저장 블록들 (4, 8 및 1) 에 대한 인접한 저장 디바이스 어드레스들일 수도 있다. 저장 디바이스 어드레스들의 범위 (32) 는 저장 어드레스들 (0-2) 을 포함할 수도 있다. 저장 디바이스 어드레스들의 범위 (34) 는 정점 데이터를 저장하는 저장 블록들, 예를 들어 저장 블록들 (10 및 5) 에 대한 인접한 저장 디바이스 어드레스들일 수도 있다. 저장 디바이스 어드레스들의 범위 (34) 는 저장 어드레스들 (3 및 4) 을 포함할 수도 있다. 저장 디바이스 어드레스들의 범위 (36) 는 명령들을 저장하는 저장 블록들, 예를 들어 저장 블록들 (2 및 9) 에 대한 인접한 저장 디바이스 어드레스들일 수도 있다. 저장 디바이스 어드레스들의 범위 (36) 는 저장 어드레스들 (5 및 6) 을 포함할 수도 있다. 저장 디바이스 어드레스들의 범위 (38) 는 상수들을 저장하는 저장 블록들, 예를 들어 저장 블록들 (0 및 6) 에 대한 인접한 저장 디바이스 어드레스들일 수도 있다. 저장 디바이스 어드레스들의 범위 (38) 는 저장 어드레스들 (7 및 8) 을 포함할 수도 있다. 저장 디바이스 어드레스들의 범위 (40) 는 픽셀 데이터를 저장하는 저장 블록들, 예를 들어 저장 블록들 (11, 3 및 7) 에 대한 인접한 저장 디바이스 어드레스들일 수도 있다. 저장 디바이스 어드레스들의 범

위 (40) 는 저장 어드레스들 (9-11) 을 포함할 수도 있다. 도 2 에 예시된 것보다 더 많거나 더 적은 텍스트 데이터, 정점 데이터, 명령들, 상수들, 및 픽셀 데이터가 존재할 수도 있다.

[0061] 도 2 에 예시된 바와 같이, 공통 메모리 캐시 (26) 는 캐시 라인들 (42A-42F)(총괄하여, "캐시 라인들 (42)" 로서 지칭됨) 을 포함할 수도 있다. 도 2 에 예시된 것보다 더 많거나 적은 캐시 라인들 (42) 이 존재할 수도 있다. 일부 예들에서, 공통 메모리 캐시 (26) 는 레벨 2 (L2) 캐시일 수도 있다. 일부 예들에서, 공통 메모리 캐시 (26) 는 빠른 범위가 무효화하는 L2 캐시와 연관된 32 킬로바이트 (KB) 8-웨이 세트일 수도 있다.

[0062] 캐시 라인들 (42) 의 각 라인 은 어드레스 필드 (30A) 및 데이터 필드 (30B) 를 포함할 수도 있다. 어드레스 필드 (30A) 는 데이터 필드 (30B) 에 저장된 그래픽 데이터에 대한 저장 디바이스 어드레스를 나타낼 수도 있다. 예를 들어, 캐시 라인 (42A) 의 어드레스 필드 (30A) 는 데이터 필드 (30B) 에 저장된 그래픽 데이터에 대한 어드레스가 1 임을 나타낸다. 도 2 에 예시된 바와 같이, 캐시 라인 (42A) 의 데이터 필드 (30B) 는, 저장 블록 (8) 의 저장 디바이스 어드레스가 1 이기 때문에 저장 블록 (8) 에 대응하는 제 2 텍스트 데이터를 저장한다.

[0063] 동일한 데이터 유형을 저장하는 공통 메모리 캐시 (26) 의 캐시 라인들 (42) 은 캐시 라인들 (42) 의 세트로서 고려될 수도 있다. 예를 들어, 도 2 에 예시된 바와 같이, 캐시 라인 (42B) 및 캐시 라인 (42E) 각각은 정점 데이터를 저장한다. 이 예에서, 캐시 라인 (42B) 및 캐시 라인 (42E) 은 정점 데이터를 저장하는 캐시 라인들의 세트를 형성할 수도 있다. 다른 예로써, 도 2 에 예시된 바와 같이, 캐시 라인 (42C), 캐시 라인 (42D), 및 캐시 라인 (42F) 각각은 픽셀 데이터를 저장한다. 이 예에서, 캐시 라인 (42C), 캐시 라인 (42D), 및 캐시 라인 (42F) 은 픽셀 데이터를 저장하는 캐시 라인들의 세트를 형성할 수도 있다.

[0064] "캐시 라인들의 세트" 는 그 세트에서의 캐시 라인들 만이 특정 그래픽 데이터 유형을 저장할 수 있음을 나타내지 않는 것으로 이해되어야 한다. 예를 들어, 도 2 에 예시된 바와 같이, 캐시 라인들 (42A 및 42E) 은 정점 데이터를 저장한다. 그러나, 대안의 예들에서, 캐시 라인들 (42A 및 42E) 은 다른 그래픽 데이터 유형들, 예를 들어 텍스트 데이터, 픽셀 데이터, 명령 데이터, 및 상수 데이터를 저장할 수도 있다. 문구 "캐시 라인들의 세트" 는 유사한 그래픽 데이터 유형들을 저장하는 캐시 라인들 (42) 의 그룹을 나타내는데 사용된다.

[0065] 일부 예들에서, 캐시 라인들의 세트는 인접하지 않을 수도 있다. 예를 들어, 캐시 라인들 (42B 및 42E) 은 캐시 라인들의 세트를 형성하지만 인접하지는 않다. 다른 예로써, 캐시 라인들 (42C, 42D, 및 42F) 은 캐시 라인들의 세트를 형성하지만 인접하지는 않다. 대안의 예들에서, 캐시 라인들의 세트가 인접한 것이 가능할 수도 있다.

[0066] 전술된 바와 같이, 일부 예들에서, 프로세서 (12) 는, 캐시 라인들 (42) 중 하나가 저장하는 데이터의 유형에 대해 그래픽 데이터가 변한다면, 캐시 라인들 (42) 중 하나를 무효화할 수도 있다. 예를 들어, 텍스트 이미지가 변한다면, 프로세서 (12) 는 텍스트 데이터를 저장하는 캐시 라인들 (42) 중 하나 이상의 캐시 라인들을 무효화할 수도 있다. 캐시 라인들 (42) 중 하나가 텍스트 데이터를 저장하는지 여부를 결정하기 위해, 프로세서 (12) 는 캐시 라인들 (42) 각각의 어드레스 필드 (30A) 를 텍스트 데이터에 할당된 저장 디바이스 어드레스들의 인접한 범위에 비교할 수도 있다.

[0067] 예를 들어, 도 2 의 예에서, 프로세서 (12) 는 인접한 저장 디바이스 어드레스들 (0-2) 을 포함하는 저장 디바이스 어드레스들의 인접한 범위 (32) 를 텍스트 데이터에 할당했다. 이 예에서, 프로세서 (12) 는 캐시 라인들 (42) 각각의 어드레스 필드 (30A) 를 비교하여 그것이 저장 디바이스 어드레스들의 인접한 범위 (32) 내에 있는지 여부를 결정할 수도 있다. 캐시 라인 (42A) 의 어드레스 필드 (30A) 는 저장 디바이스 어드레스들의 인접한 범위 (32) 내에 있다, 예를 들어 1 은 0-2 내에 있다. 이 예에서, 프로세서 (12) 는 캐시 라인 (42A) 의 데이터 필드 (30B) 에 저장된 제 2 텍스트 데이터를 널 (null) 데이터로 대체할 수도 있다. 프로세서 (12) 는 그 후, 저장 디바이스 (20) 로부터 새로운 텍스트 데이터를 추출하고, 이 텍스트 데이터를 캐시 라인 (42A) 에 저장할 수도 있다. 프로세서 (12) 는 모든 예에서 캐시 라인 (42A) 에 텍스트 데이터를 저장할 필요가 없을 수도 있다. 일부 예들에서, 프로세서 (12) 는 또한, 추출된 새로운 텍스트 데이터에 대한 저장 디바이스 어드레스가 1 과 상이한 저장 디바이스 어드레스에 저장된다면, 캐시 라인 (42A) 의 어드레스 필드 (30A) 를 업데이트할 수도 있다.

[0068] 도 3 은 도 1 의 컴포넌트들 중 일부를 더 상세히 예시하는 다른 블록도이다. 예를 들어, 도 3 은 도 1 의 공통 메모리 캐시 (26), IOMMU (28), 및 저장 디바이스 (20) 를 더 상세히 예시한다. 도 3 에 예시된 바와

같이, 저장 디바이스 (20) 는 도 2 에 예시된 저장 디바이스 (20) 와 유사할 수도 있다. 그러나, 도 3 의 예에서, 프로세서 (12) 는 인접한 저장 디바이스 어드레스들의 범위들을 할당하지 않을 수도 있다. 예를 들어, 도 3 에 예시된 바와 같이, 저장 디바이스 (20) 의 저장 블록들의 각 블록에 대한 어드레스들은 연속해서 정렬되지 않는다. 예시의 편의성을 위해, 도 3 의 예에서 저장 블록에 대한 저장 디바이스 어드레스는 저장 블록의 식별자에 대응한다. 예를 들어, 저장 디바이스 (20) 의 저장 블록 (4) 에 대한 저장 디바이스 어드레스는 4 이고, 저장 디바이스 (20) 의 저장 블록 (8) 에 대한 저장 디바이스 어드레스는 8 이고, 등등이다. 그러나, 본 개시물의 양태들은 이렇게 제한되지 않는다. 저장 블록에 대한 저장 디바이스 어드레스는 저장 블록의 식별자에 제한되지 않는다.

[0069] 도 3 은 IOMMU (28) 를 더 상세히 예시한다. 도 3 의 예에서, IOMMU (28) 는 12 개의 어드레스 블록들 0-11 을 포함한다. IOMMU (28) 는 12 개의 어드레스 블록들 보다 많거나 적은 어드레스 블록들을 포함할 수도 있다. 일부 예들에서, IOMMU (28) 는 저장 디바이스 (20) 의 저장 블록들 (0-11) 에 대한 가상화된 어드레스 공간을 GPU (14) 에 제공할 수도 있다.

[0070] 어드레스 블록들 (0-11) 중 각 블록은 그 어드레스에 의해 개별적으로 주소지정 가능할 수도 있다. 예를 들어, 도 3 은 IOMMU 어드레스들 (0-11) 을 예시한다. IOMMU 어드레스들 (0-11) 은, 각 세그먼트가 IOMMU 어드레스들의 범위를 포함하는 세그먼트들로 프래그먼트화될 수도 있다. 프로세서 (12) 는 IOMMU 어드레스들의 각 범위를 특정 데이터 유형에 할당할 수도 있다. IOMMU 어드레스들의 범위는 각 데이터 유형에 대해 인접할 수도 있다.

[0071] 예를 들어, 도 3 에 예시된 바와 같이, IOMMU 어드레스의 범위 (44) 는 텍스처 데이터를 저장하는 저장 디바이스 (20) 의 저장 블록들에 대한 저장 디바이스 어드레스들, 예를 들어 저장 디바이스 어드레스들 (4, 8 및 1) 을 저장하는 어드레스 블록들에 대한 인접한 IOMMU 어드레스들일 수도 있다. IOMMU 어드레스들의 범위 (44) 는 IOMMU 어드레스들 (0-2) 을 포함할 수도 있다. IOMMU 어드레스의 범위 (46) 는 정점 데이터를 저장하는 저장 디바이스 (20) 의 저장 블록들에 대한 저장 디바이스 어드레스들, 예를 들어 저장 디바이스 어드레스들 (10 및 5) 을 저장하는 어드레스 블록들에 대한 인접한 IOMMU 어드레스들일 수도 있다. IOMMU 어드레스들의 범위 (46) 는 IOMMU 어드레스들 (3 및 4) 을 포함할 수도 있다. IOMMU 어드레스의 범위 (48) 는 명령들을 저장하는 저장 디바이스 (20) 의 저장 블록들에 대한 저장 디바이스 어드레스들, 예를 들어 저장 디바이스 어드레스들 (2 및 9) 을 저장하는 어드레스 블록들에 대한 인접한 IOMMU 어드레스들일 수도 있다. IOMMU 어드레스들의 범위 (48) 는 IOMMU 어드레스들 (5 및 6) 을 포함할 수도 있다. IOMMU 어드레스의 범위 (50) 는 상수 (constant) 를 저장하는 저장 디바이스 (20) 의 저장 블록들에 대한 저장 디바이스 어드레스들, 예를 들어 저장 디바이스 어드레스들 (0 및 6) 을 저장하는 어드레스 블록들에 대한 인접한 IOMMU 어드레스들일 수도 있다. IOMMU 어드레스들의 범위 (50) 는 IOMMU 어드레스들 (7 및 8) 을 포함할 수도 있다. IOMMU 어드레스의 범위 (52) 는 픽셀 데이터를 저장하는 저장 디바이스 (20) 의 저장 블록들에 대한 저장 디바이스 어드레스들, 예를 들어 저장 디바이스 어드레스들 (11, 3, 및 7) 을 저장하는 어드레스 블록들에 대한 인접한 IOMMU 어드레스들일 수도 있다. IOMMU 어드레스들의 범위 (52) 는 IOMMU 어드레스들 (9-11) 을 포함할 수도 있다. 도 2 와 같이, 도 3 에 예시된 더 많거나 적은 텍스처 데이터, 정점 데이터, 명령들, 상수들, 및 픽셀 데이터가 존재할 수도 있다.

[0072] 도 2 와 유사하게, 도 3 에 예시된 바와 같이, 공통 메모리 캐시 (26) 는 캐시 라인들 (42) 을 포함할 수도 있다. 또한, 도 2 와 유사하게, 캐시 라인들 (42) 의 각 라인은 어드레스 필드 (30A) 및 데이터 필드 (30B) 를 포함할 수도 있다. 도 3 에 예시된 예에서, 어드레스 필드 (30A) 는 데이터 필드 (30B) 에 저장된 그래픽 데이터가 저장 디바이스 (20) 에 저장되는 어드레스를 저장하는 IOMMU (28) 의 어드레스 블록에 대한 IOMMU 어드레스를 나타낼 수도 있다. 예를 들어, 도 3 에서 캐시 라인 (42A) 의 어드레스 필드 (30A) 는 IOMMU (28) 의 어드레스 블록에 대한 어드레스가 1 임을 나타낸다. 도 3 에 예시된 바와 같이, IOMMU 어드레스 (1) 은 IOMMU (28) 의 어드레스 블록 (1) 에 대한 것이다. IOMMU 의 어드레스 블록 1 은 저장 디바이스 어드레스 (8) 를 저장한다. 저장 디바이스 어드레스 (8) 는 저장 디바이스 (20) 에서의 저장 블록 (8) 에 대응한다. 도 3 에 예시된 바와 같이, 저장 디바이스 (20) 의 저장 블록 (8) 은 제 2 텍스처 데이터를 저장한다. 캐시 라인 (42A) 의 데이터 필드 (30B) 는 저장 디바이스 어드레스 (8) 를 갖는 저장 블록 (8) 에 대응하는 제 2 텍스처 데이터를 저장한다. 저장 디바이스 어드레스 (8) 는 IOMMU (28) 의 어드레스 블록 (1) 에 대응하고, 어드레스 블록 (1) 에 대한 IOMMU 어드레스는 1 인데, 이는 캐시 라인 (42A) 의 데이터 필드 (30A) 에 대응한다.

[0073] 상기와 같이, 일부 예들에서, 캐시 라인들 (42) 중 하나가 저장하는 데이터의 유형에 대해 그래픽 데이터가 변

하는 경우, 프로세서 (12) 는 도 3 의 캐시 라인들 (42) 중 하나를 무효화할 수도 있다. 앞서와 같이, 텍스처 이미지가 변한다고 가정하고, 프로세서 (12) 는 텍스처 데이터를 저장하는 캐시 라인들 (42) 중 하나 이상의 캐시 라인들을 무효화할 필요가 있을 수도 있다. 캐시 라인들 (42) 중 하나가 텍스처 데이터를 저장하는지 여부를 결정하기 위해, 프로세서 (12) 는 텍스처 데이터에 할당된 IOMMU 어드레스들의 인접한 범위에 캐시 라인들 각각의 어드레스 필드 (30A) 를 비교할 수도 있다.

[0074] 예를 들어, 도 2 의 예와 유사하게, 프로세서 (12) 는 인접한 IOMMU 어드레스들 (0-2) 을 포함하는 IOMMU 어드레스들의 인접한 범위 (44) 를 텍스처 데이터에 할당했다. 이 예에서, 프로세서 (12) 는 캐시 라인들 (42) 각각의 어드레스 필드 (30A) 를 비교하여 그것이 IOMMU 어드레스들의 인접한 범위 (44) 내에 있는지 여부를 결정할 수도 있다. 도 3 에서 캐시 라인 (42A) 의 어드레스 필드 (30A) 는 IOMMU 어드레스들 (44) 의 인접한 범위 내에 있다. 이 예에서, 프로세서 (12) 는 캐시 라인 (42A) 을 무효화할 수도 있다.

[0075] 도 4 는 본 개시물의 양태들을 구현하도록 구성될 수도 있는 디바이스 (10) 의 예시의 동작을 예시하는 흐름도이다. 예시의 목적을 위해 도 1, 도 2 및 도 3 을 참조한다. 전술된 바와 같이, 프로세싱 유닛, 예를 들어 프로세서 (12) 또는 GPU (14) 는 저장 디바이스 (20) 의 저장 공간을, 각각 IOMMU (28) 의 프래그먼트 어드레스 공간 또는 복수의 저장 블록들을 포함하는 세그먼트들로 프래그먼트화하고 또는 IOMMU (28) 의 어드레스 공간을 각각 복수의 어드레스 블록들을 포함하는 세그먼트들로 프래그먼트화할 수도 있다. 예시의 목적을 위해, 저장 디바이스 (20) 의 저장 블록들 및 IOMMU (28) 의 어드레스 블록들은 일반적으로 블록들로서 지칭될 수도 있다.

[0076] 프로세싱 유닛은 제 1 데이터 유형에 대한 어드레스들의 제 1 인접한 범위, 및 제 2 데이터 유형에 대한 어드레스들의 제 2 인접한 범위를 할당할 수도 있다 (54). 제 1 및 제 2 데이터 유형들은 그래픽 프로세싱을 위해 상이한 데이터 유형들일 수도 있다. 예를 들어, 도 2 에 예시된 바와 같이, 저장 디바이스 어드레스들의 인접한 범위 (32, 34, 36, 38, 및 40) 중 하나는 어드레스들의 제 1 인접한 범위를 포함할 수도 있다. 저장 디바이스 어드레스들의 인접한 범위 (32, 34, 36, 38, 및 40) 의 다른 하나는 어드레스들의 제 2 인접한 범위를 포함할 수도 있다. 또한, 도 2 에 예시된 바와 같이, 저장 디바이스 어드레스들의 인접한 범위 (32, 34, 36, 38, 및 40) 의 각각은 특정 데이터 유형에 대해 할당된다. 저장 디바이스 어드레스들의 인접한 범위 (32, 34, 36, 38, 및 40) 의 각각에 대한 할당된 유형은 제 1 데이터 유형을 포함할 수도 있다. 저장 디바이스 어드레스들의 인접한 범위 (32, 34, 36, 38, 및 40) 의 다른 하나에 대한 할당된 데이터 유형은 제 2 데이터 유형을 포함할 수도 있다.

[0077] 다른 예로써, 도 4 에 예시된 바와 같이, IOMMU 어드레스들 (44, 46, 48, 50, 및 52) 의 인접한 범위 중 하나는 어드레스들의 제 1 인접한 범위를 포함할 수도 있다. IOMMU 어드레스들의 인접한 범위 (44, 46, 48, 50, 및 52) 의 다른 하나는 어드레스들의 제 2 인접한 범위를 포함할 수도 있다. 또한, 도 3 에 예시된 바와 같이, IOMMU 어드레스들의 인접한 범위 (44, 46, 48, 50, 및 52) 의 각각은 특정 데이터 유형에 대해 할당된다. IOMMU 어드레스들의 인접한 범위 (44, 46, 48, 50, 및 52) 의 각각에 대해 할당된 데이터 유형은 제 1 데이터 유형을 포함할 수도 있다. IOMMU 어드레스들의 연속적이 범위 (44, 46, 48, 50, 및 52) 의 다른 하나에 대해 할당된 데이터 유형은 제 2 데이터 유형을 포함할 수도 있다.

[0078] 프로세싱 유닛, 예를 들어 프로세서 (12) 또는 GPU (14) 는 그 어드레스가 어드레스들의 제 1 인접한 범위 내에 있는 블록들 내의 제 1 데이터 유형의 그래픽 데이터의 어드레스들 또는 제 1 데이터 유형의 그래픽 데이터를 저장하고, 그 어드레스들이 어드레스들의 제 2 인접한 범위 내에 있는 블록들 내의 제 2 데이터 유형의 그래픽 데이터의 어드레스들 또는 제 2 데이터 유형의 그래픽 데이터를 저장할 수도 있다 (56). 예를 들어, 도 2 에 예시된 바와 같이, 프로세서 (12) 또는 GPU (14) 는 그 어드레스들이 저장 디바이스 어드레스들의 인접한 범위 (32) 내에 있는 저장 디바이스 (20) 의 저장 블록들 내의 텍스처 데이터를 저장할 수도 있다. 도 2 에 예시된 바와 같이, 프로세서 (12) 또는 GPU (14) 는 저장 디바이스 (20) 의 저장 블록들 내의 특정 데이터 유형의 그래픽 데이터를 저장할 수도 있는데, 저장 디바이스의 어드레스들은 그 특정 데이터 유형에 대해 할당되는 어드레스들의 인접한 범위 내에 있다.

[0079] 다른 예로써, 도 3 에 예시된 바와 같이, 프로세싱 유닛, 예를 들어 프로세서 (12) 또는 GPU (14) 는 저장 디바이스 (20) 에 텍스처 데이터가 저장되는 어드레스들을 IOMMU (28) 의 어드레스 블록들 내에 저장할 수도 있는데, IOMMU 의 어드레스들은 IOMMU 어드레스들의 인접한 범위 (44) 내에 있다. 도 3 에 예시된 바와 같이, 프로세서 (12) 또는 GPU (14) 는 특정 데이터 유형의 그래픽 데이터가 저장되는 어드레스들을 저장 디바이스 (20) 의 저장 블록들 내에 저장할 수도 있는데, 저장 디바이스의 어드레스들은 그 특정 데이터 유형에 대해

할당되는 어드레스들의 인접한 범위 내에 있다.

- [0080] 프로세서 (12) 또는 GPU (14) 는 공통 메모리 캐시 (26) 의 복수의 캐시 라인들의 캐시 라인들 중 일부, 제 1 데이터 유형의 그래픽 데이터를 저장하고, 공통 메모리 캐시 (26) 의 복수의 캐시 라인들의 캐시 라인들 중 일부에 제 2 데이터 유형의 그래픽 데이터를 저장할 수도 있다 (58). 예를 들어, 도 2 및 도 3 에 예시된 바와 같이, 공통 메모리 캐시 (26) 는 캐시 라인들 (42) 을 포함한다. 또한, 도 2 및 도 3 에 예시된 바와 같이, 캐시 라인들 (42B 및 42E) 은 특정 데이터 유형의 그래픽 데이터, 예를 들어 정점 데이터를 저장하는 캐시 라인들의 그룹으로서 고려될 수도 있다. 캐시 라인들 (42C, 42D, 및 42F) 은 특정 데이터 유형의 그래픽 데이터, 예를 들어 픽셀 데이터를 저장하는 캐시 라인들의 다른 그룹으로서 고려될 수도 있다.
- [0081] 도 5 는 공통 메모리 캐시 (26) 의 복수의 캐시 라인들 중 어느 라인들이 특정 데이터 유형과 연관되는지를 결정하기 위한 예시의 기법을 예시하는 흐름도이다. 예시의 목적을 위해, 도 1, 2, 및 3 에 대한 참조가 이루어진다. 프로세싱 유닛, 예를 들어 프로세서 (12) 또는 GPU (14) 는 각 캐시 라인의 어드레스 필드를 어드레스들의 인접한 범위에 비교할 수도 있다 (60). 예를 들어, 프로세서 (12) 또는 GPU (14) 는 캐시 라인들 (42) 의 어드레스 필드 (30A) 를 도 2 의 저장 디바이스 어드레스들의 인접한 범위 (32, 34, 36, 38, 및 40) 중 각각의 범위에 비교할 수도 있다. 다른 예로써, 프로세서 (12) 또는 GPU (14) 는 캐시 라인들 (42) 의 어드레스 필드 (30A) 를 도 3 의 IOMMU 어드레스들의 인접한 범위 (44, 46, 48, 50, 및 52) 의 각 범위에 비교할 수도 있다.
- [0082] 프로세서 (12) 또는 GPU (14) 는 캐시 라인들 (42) 중 어느 캐시 라인들이 결정에 기초한 데이터 유형과 연관되는지를 결정할 수도 있다 (62). 예를 들어, 프로세서 (12) 또는 GPU (14) 는, 도 2 및 도 3 에 예시된 바와 같이 캐시 라인 (42A) 이 텍스처 데이터와 연관되고, 캐시 라인들 (42B 및 42E) 이 정점 데이터와 연관되며, 캐시 라인들 (42C, 42D, 및 42F) 이 픽셀 데이터와 연관된다는 것을 결정할 수도 있다. 도 2 의 예에서, 프로세서 (12) 또는 GPU (14) 는, 캐시 라인 (42A) 의 어드레스 필드 (30A) 가 텍스처 데이터를 저장하는 저장 디바이스 (20) 의 저장 블록에 대한 어드레스에 대응하기 때문에 캐시 라인 (42A) 이 텍스처 데이터와 연관된다는 것을 결정할 수도 있다. 프로세서 (12) 또는 GPU (14) 는, 캐시 라인들 (42B 및 42E) 의 어드레스 필드 (30A) 가 정점 데이터를 저장하는 저장 디바이스 (20) 의 저장 블록들에 대한 어드레스에 대응하기 때문에 캐시 라인들 (42B 및 42E) 이 정점 데이터와 연관된다는 것을 결정할 수도 있다. 프로세서 (12) 또는 GPU (14) 는, 캐시 라인들 (42C, 42D, 및 42F) 의 어드레스 필드 (30A) 가 픽셀 데이터를 저장하는 저장 디바이스 (20) 의 저장 블록들에 대한 어드레스에 대응하기 때문에 캐시 라인들 (42C, 42D, 및 42F) 이 픽셀 데이터와 연관된다는 것을 결정할 수도 있다.
- [0083] 도 3 의 예에서, 프로세서 (12) 또는 GPU (14) 는, 캐시 라인 (42A) 의 어드레스 필드 (30A) 가 저장 블록 (20) 에 저장되는 텍스처 데이터에 대한 어드레스를 저장하는 IOMMU (28) 의 어드레스 블록의 어드레스에 대응하기 때문에 캐시 라인 (42A) 이 텍스처 데이터와 연관된다는 것을 결정할 수도 있다. 프로세서 (12) 또는 GPU (14) 는, 캐시 라인들 (42B 및 42E) 의 어드레스 필드 (30A) 가 저장 블록 (20) 에 정점 데이터가 저장되는 어드레스를 저장하는 IOMMU (28) 의 어드레스 블록들의 어드레스에 대응하기 때문에 캐시 라인들 (42B 및 42E) 이 정점 데이터와 연관된다는 것을 결정할 수도 있다. 프로세서 (12) 또는 GPU (14) 는, 캐시 라인들 (42C, 42D, 및 42F) 의 어드레스 필드 (30A) 가 저장 블록 (20) 에 픽셀 데이터가 저장되는 어드레스를 저장하는 IOMMU (28) 의 어드레스 블록들의 어드레스에 대응하기 때문에 캐시 라인들 (42C, 42D, 및 42F) 이 픽셀 데이터와 연관된다는 것을 결정할 수도 있다.
- [0084] 도 6 은 프로세싱 유닛, 예를 들어 프로세서 (12) 또는 GPU (14) 에 의해 수행된 예시의 기법을 예시하는 흐름도이다. 예시의 목적을 위해, 도 1, 2, 및 3 에 대한 참조가 이루어진다. 프로세싱 유닛, 예를 들어 프로세서 (12) 또는 GPU (14) 는 제 1 데이터 유형의 그래픽 데이터 또는 제 2 데이터 유형의 그래픽 데이터에 대한 요청을 수신할 수도 있다 (64). 요청은 제 1 데이터 유형 또는 제 2 데이터 유형에 대한 어드레스를 포함할 수도 있다. 예를 들어, GPU (14) 상에서 실행하는 소프트웨어, 예컨대 셰이더 프로그램은 GPU (14) 로 하여금 특정 데이터 유형의 그래픽 데이터, 예를 들어 제 1 데이터 유형의 그래픽 데이터 또는 제 2 데이터 유형에 대한 그래픽 데이터를 추출하게 하는 요청을 생성할 수도 있다. GPU (14) 상에서 실행하는 소프트웨어는 요청에서 제 1 또는 제 2 데이터 유형의 그래픽 데이터를 추출하기 위한 어드레스들을 GPU (14) 에 제공할 수도 있다.
- [0085] 프로세서 (12) 또는 GPU (14) 는 제 1 또는 제 2 데이터 유형의 그래픽 데이터에 대한 요청 내에서 어드레스가 어드레스들의 제 1 인접한 범위 또는 어드레스들의 제 2 인접한 범위 내에 각각 있는지를 결정할 수도 있다

(66). 예를 들어, 세이더 프로그램이 텍스처 데이터를 요청했고 저장 디바이스 어드레스 또는 IOMMU 어드레스를 포함했다고 가정한다. 이 예에서, GPU (14)는 저장 디바이스 어드레스가 도 2의 저장 디바이스 어드레스의 인접한 범위 (32) 내에 있는지를 결정하고, 또는 IOMMU 어드레스가 도 3의 IOMMU 어드레스의 인접한 범위 (44) 내에 있는지를 결정할 수도 있다. 요청에서, 어드레스가 어드레스들의 인접한 범위 내에 있는지 여부를 결정함으로써, 프로세서 (12) 또는 GPU (14)는 프로세서 (12) 또는 GPU (14)가 부주의로 부정확한 데이터를 추출하지 않았음을 보장할 수도 있다.

[0086] 프로세서 (12) 또는 GPU (14)는 그 후, 결정에 기초하여 요청을 프로세싱할 수도 있다 (68). 예를 들어, 제 1 또는 제 2 데이터 유형의 그래픽 데이터에 대한 요청이 어드레스들의 제 1 인접한 범위 또는 어드레스들의 제 2 인접한 범위 내에 각각 있으면, 프로세서 (12) 또는 GPU (14)는 요청을 프로세싱할 수도 있다. 그러나, 제 1 또는 제 2 데이터 유형의 그래픽 데이터에 대한 요청이 어드레스들의 제 1 인접한 데이터 또는 어드레스들의 제 2 인접한 데이터 내에 있지 않으면, 프로세서 (12) 또는 GPU (14)는 요청을 프로세싱하지 않을 수도 있다.

[0087] 하나 이상의 예들에서, 전술된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합에서 구현될 수도 있다. 소프트웨어에서 구현되는 경우, 기능들은 비일시적 컴퓨터 판독가능 매체를 포함하는 제조품 상의 하나 이상의 명령들 또는 코드로서 저장될 수도 있다. 컴퓨터 판독가능 매체는 일 장소에서 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 컴퓨터 데이터 저장 매체 또는 통신 매체를 포함할 수도 있다. 데이터 저장 디바이스는 본 개시물에 설명된 기법들의 구현을 위해 명령들, 코드 및/또는 데이터 구조들을 추출하도록 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용 가능한 매체일 수도 있다. 비제한적인 예로써, 이러한 컴퓨터 판독가능 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장 디바이스, 자기 디스크 저장 디바이스, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 컴퓨터에 의해 액세스될 수 있고 명령들이나 데이터의 형태로 원하는 프로그램 코드를 운반 또는 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 본원에 사용된 바와 같은 디스크 (disk) 및 디스크 (disc)는 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루-레이 디스크를 포함하는데, 여기서 디스크 (disk)는 주로 데이터를 자기적으로 재생하는 한편, 디스크 (disc)는 레이저를 이용하여 데이터를 광학적으로 재생한다. 상기의 조합들은 또한, 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

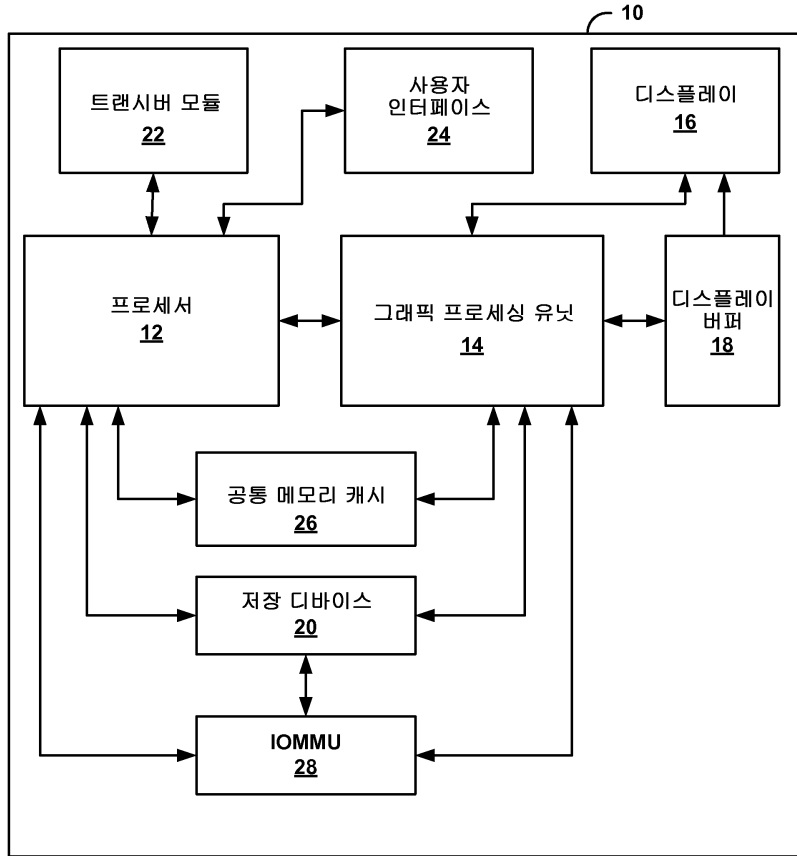
[0088] 하나 이상의 프로세서들, 예컨대 하나 이상의 DSP 들, 범용 마이크로프로세서들, ASIC 들, FPGA 들, 또는 다른 등가의 집적 또는 이산 로직 회로에 의해 코드가 실행될 수도 있다. 또한, 일부 양태들에서, 본원에 설명된 기능성은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 내에 제공되거나, 결합형 코덱으로 통합될 수도 있다. 또한, 하나 이상의 회로들 또는 로직 엘리먼트들에서 기법들이 충분히 구현될 수 있다.

[0089] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트를 포함하는 광범위한 디바이스들 또는 장치들에서 구현될 수도 있다. 각종 컴포넌트들, 모듈들, 또는 유닛들이 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에 설명되었으나, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로하지는 않는다. 차라리, 전술된 바와 같이, 각종 유닛들은 적합한 소프트웨어 및/또는 펌웨어와 관련하여 전술된 하나 이상의 프로세서들을 포함하는, 상호작용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있거나 하드웨어 유닛에 결합될 수도 있다.

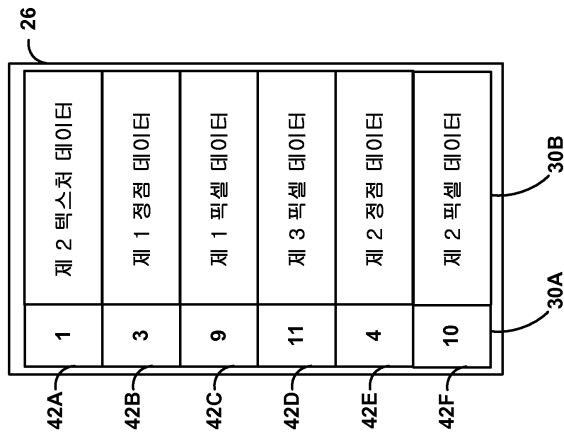
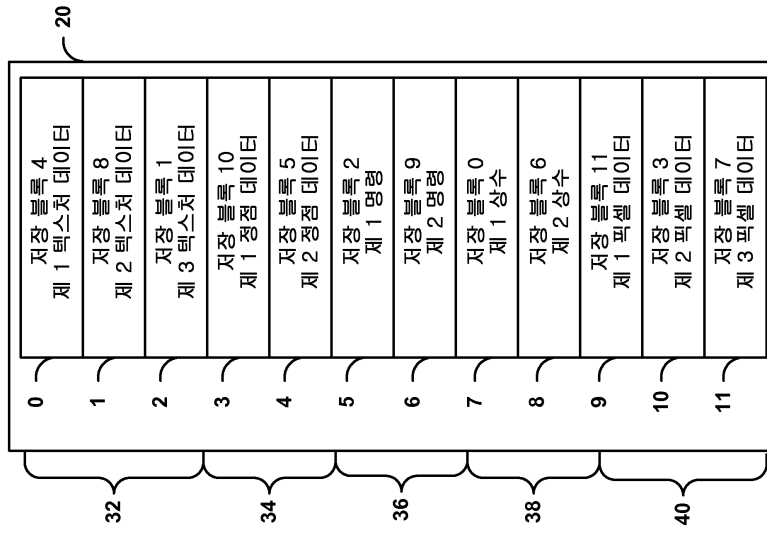
[0090] 각종 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구 범위 내에 있다.

도면

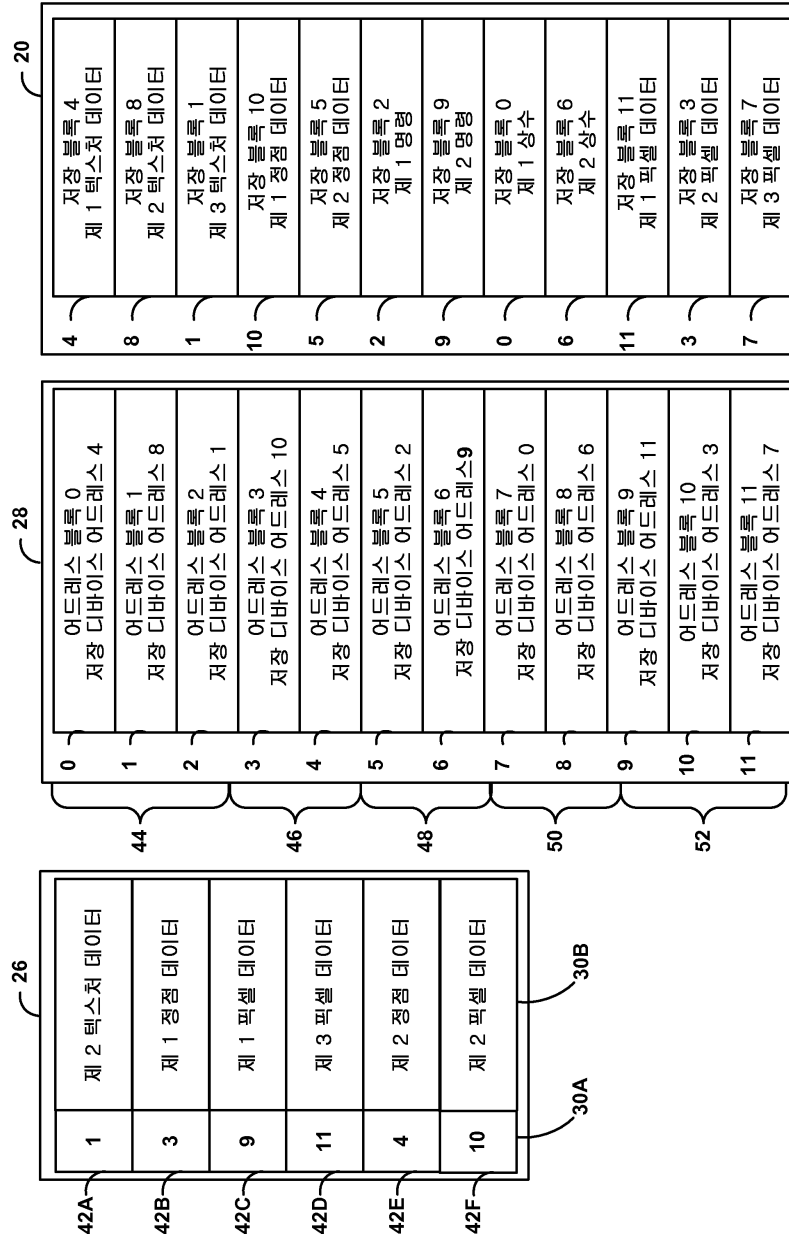
도면1



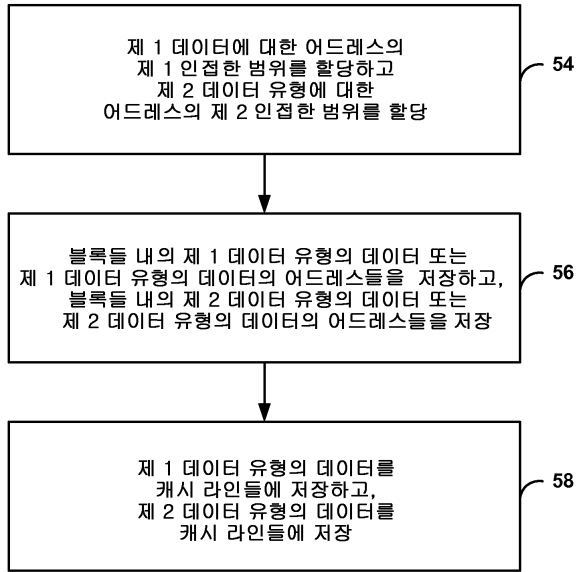
도면2



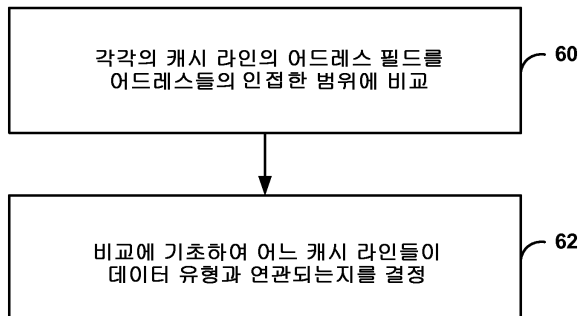
도면3



도면4



도면5



도면6

