(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2010-0038462
(43) 공개일자 2010년04월14일

(51) Int. Cl.
G06T 1/00 (2006.01)
(21) 출원번호 10-2010-7004551
(22) 출원일자 2008년07월30일
심사청구일자 2010년02월26일
(85) 번역문제출일자 2010년02월26일
(86) 국제출원번호 PCT/US2008/071655
(87) 국제공개번호 WO 2009/018385
국제공개일자 2009년02월05일
(30) 우선권주장
11/830,667 2007년07월30일 미국(US)

(71) 출원인
콸콤 인코포레이티드
미국 캘리포니아 샌디에고 모어하우스 드라이브5775 (우 92121-1714)
(72) 발명자
지아오, 구오팡
미국 92121 캘리포니아 샌디에고 모어하우스 드라이브 5775
보우드, 알렉세이, 브이.
미국 92121 캘리포니아 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
남상선

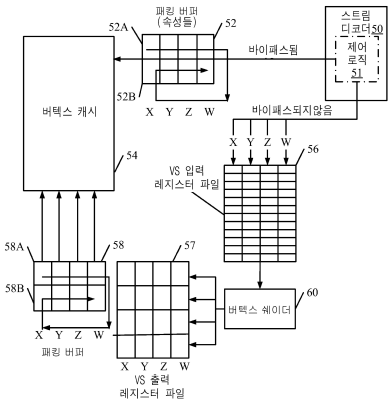
전체 청구항 수 : 총 38 항

(54) 그래픽 시스템에서 변수들의 패킹 및 링킹을 위한 방식

(57) 요 약

제 1-레벨 컴파일러 패킹 프로세스 및 제 2-레벨 하드웨어 패킹 프로세스를 가변치들에 대해 수행하는 무선 디바이스가 제공된다. 컴파일러 패킹 프로세스는 콤포넌트들의 합이 M과 같은 2개 이상의 쉐이더 변수들(가변치들 또는 속성들)을 공유형 M-차원(MD) 벡터 레지스터에 패킹한다. 하드웨어 패킹은 쉐이더 변수들(가변치들 또는 속성들)의 M 콤포넌트들 및 남아있는 임의의 변수들을 버텍스 캐시 또는 다른 저장 매체에 연이어 패킹한다.

대 표 도 - 도4



(72) 발명자

유, 천

미국 92121 캘리포니아 샌디에고 모어하우스 드라이브 5775

첸, 링준

미국 92121 캘리포니아 샌디에고 모어하우스 드라이브 5775

두, 윤

미국 92121 캘리포니아 샌디에고 모어하우스 드라이브 5775

특허청구의 범위

청구항 1

다수의 공유형 M-차원(MD) 레지스터들을 포함하는 저장 매체; 및

콤포넌트들의 합이 M과 같은 하나 이상의 쉐이더 변수들(shader variables)을 각각의 공유형 MD 레지스터에 패킹(pack)하기 위해 오퍼레이션들(operations)의 세트를 실행하는(implement) 프로세싱 유니트

를 포함하는, 디바이스.

청구항 2

제 1 항에 있어서,

상기 오퍼레이션들의 세트는 2개의 2D 벡터 쉐이더 변수들; 3D 벡터 쉐이더 변수 및 플로트(float) 쉐이더 변수; 2D 벡터 쉐이더 변수 및 2개의 상이한(distinct) 플로트 쉐이더 변수들; 및 4개의 플로트 쉐이더 변수들 중 하나를 패킹하는, 디바이스.

청구항 3

제 1 항에 있어서,

상기 쉐이더 변수들은 버텍스(vertex) 쉐이더로부터 출력 가변치들(varyings)의 세트를 포함하며, 상기 오퍼레이션들의 세트는 상기 다수의 공유형 MD 레지스터들의 상기 출력 가변치들의 세트를 패킹하고 버텍스 쉐이더 출력 파일을 생성하기 위해 임의의 나머지 가변치들을 상기 제 2 저장 매체에 상주시키기 위한(to poulate) 오퍼레이션들을 포함하는, 디바이스.

청구항 4

제 3 항에 있어서,

다수의 MD 캐시 레지스터들을 포함하는 버텍스 캐시(cache)를 더 포함하며, 상기 저장 매체는 그 내부에 상기 버텍스 쉐이더 출력 파일의 M 콤포넌트들을 연이어(consecutively) 패킹하고, 채워질 때 패킹 버퍼의 MD 레지스터의 콘텐츠들을 상기 버텍스 캐시의 각각의 MD 캐시 레지스터에 전송하기 위한 MD 레지스터를 갖는 패킹 버퍼를 포함하는, 디바이스.

청구항 5

제 4 항에 있어서,

상기 프로세싱 유니트는 프래그먼트(fragment) 쉐이더에 대한 입력 가변치들의 세트와 상기 버텍스 캐시의 상기 출력 가변치들을 링킹하기 위한 오퍼레이션들의 제 2 세트를 실행하는, 디바이스.

청구항 6

제 5 항에 있어서,

상기 오퍼레이션들의 제 2 세트는 상기 프래그먼트 쉐이더에 대한 상기 입력 가변치들의 세트에서 가변치 심볼 이름들에 대응하게 상기 버텍스 쉐이더의 상기 출력 가변치들에 대응하는 가변치 심볼 이름들을 매칭시키는 오퍼레이션들을 포함하는, 디바이스.

청구항 7

제 4 항에 있어서,

제 2 다수의 공유형 MD 레지스터들을 포함하는 제 2 저장 매체를 더 포함하며, 상기 쉐이더 변수들은 상기 버텍스 쉐이더에 대한 입력을 위해 바이패스되지 않은(non-bypassed) 입력 속성들 및 바이패싱된 입력 속성들을 더 포함하며, 상기 오퍼레이션들의 세트는 상기 제 2 다수의 공유형 MD 레지스터들에 상기 바이패스되지 않은 입력 속성들을 패킹하고 버텍스 쉐이더 입력 파일을 생성하도록 남아있는 임의의 바이패스되지않은 입력 속성들을 상

기 제 2 저장 매체에 상주시키기 위한 오퍼레이션들을 포함하는, 디바이스.

청구항 8

제 7 항에 있어서,

상기 버텍스 캐시는 제 2 다수의 MD 캐시 레지스터들을 더 포함하며, 상기 저장 매체는 상기 바이패싱된 입력 속성들의 M 콤포넌트들을 그 내부에 연이어 패킹하고 채워질 때 상기 버텍스 캐시의 상기 제 2 다수의 MD 캐시 레지스터들의 각각의 MD 캐시 레지스터로 제 2 패킹 버퍼의 상기 MD 레지스터의 콘텐츠들을 전송하기 위한 MD 레지스터를 갖는 제 2 패킹 버퍼를 포함하는, 디바이스.

청구항 9

제 4 항에 있어서,

제 2 다수의 공유형 MD 레지스터들을 포함하는 제 2 저장 매체를 더 포함하며, 상기 쉐이더 변수는 상기 버텍스 쉐이더에 대한 입력을 위한 입력 속성들의 세트를 포함하며, 패킹되는 경우, 상기 오퍼레이션들의 세트는 패킹되지 않은(unpacked) 나머지 입력 속성들의 콤포넌트들이 M을 초과할 때까지 상기 제 2 다수의 공유형 MD 레지스터들의 상기 입력 속성들의 세트를 패킹하고 버텍스 쉐이더 입력 파일이 생성되도록 상기 제 2 저장 매체에 상기 나머지 입력 속성들을 상주시키기 위한 오퍼레이션들을 포함하는, 디바이스.

청구항 10

제 1 항에 있어서,

상기 프로세싱 유니트는 셀룰러 전화, 무선 디바이스, 무선 통신 디바이스, 비디오 게임 콘솔, PDA(personal digital assistant), 랩톱 컴퓨터, 및 오디오/비디오-인에이블드(enabled) 디바이스 중 하나의 일부인, 디바이스.

청구항 11

집적 회로로서,

다수의 공유형 M-차원(MD) 레지스터들을 포함하는 저장 매체; 및

콤포넌트들의 합이 M인 하나 이상의 쉐이더 변수들을 각각의 공유형 MD 레지스터에 패킹하는 오퍼레이션들의 세트를 실행하는 프로세싱 유니트

를 포함하는, 집적 회로.

청구항 12

제 11 항에 있어서,

상기 오퍼레이션들의 세트는 2개의 2D 벡터 쉐이더 변수들; 3D 벡터 쉐이더 변수 및 플로트(float) 쉐이더 변수; 2D 벡터 쉐이더 변수 및 2개의 상이한 플로트 쉐이더 변수들; 및 4개의 플로트 쉐이더 변수들 중 하나를 패킹하는, 집적 회로.

청구항 13

제 11 항에 있어서,

상기 쉐이더 변수들은 버텍스(vertex) 쉐이더로부터 출력 가변치들(varyings)의 세트를 포함하며, 상기 오퍼레이션들의 세트는 상기 다수의 공유형 MD 레지스터들의 상기 출력 가변치들의 세트를 패킹하고 버텍스 쉐이더 출력 파일을 생성하기 위해 임의의 나머지 가변치들을 상기 저장 매체에 상주시키기 위한(to poulate) 오퍼레이션들을 포함하는, 집적 회로.

청구항 14

제 13 항에 있어서,

다수의 MD 캐시 레지스터들을 포함하는 버텍스 캐시(cache)를 더 포함하며, 상기 저장 매체는 그 내부에 상기

버텍스 쉐이더 출력 파일의 M 콤포넌트들을 연이어(consecutively) 패킹하고 채워질 때 상기 버텍스 캐시의 각각의 MD 캐시 레지스터에 패킹 버퍼의 상기 MD 레지스터의 콘텐츠들을 전송하기 위한 MD 레지스터를 갖는 패킹 버퍼를 포함하는, 집적 회로.

청구항 15

제 14 항에 있어서,

상기 프로세싱 유니트는 프래그먼트(fragment) 쉐이더에 대한 입력 가변치들의 세트와 상기 버텍스 캐시의 상기 출력 가변치들을 링킹하기 위한 오퍼레이션들의 제 2 세트를 실행하는, 집적 회로.

청구항 16

제 15 항에 있어서,

상기 오퍼레이션들의 제 2 세트는 상기 프래그먼트 쉐이더에 대한 상기 입력 가변치들의 세트에서 가변치 심볼 이름들에 대응하게 상기 버텍스 쉐이더의 상기 출력 가변치들에 대응하는 가변치 심볼 이름들을 매칭시키는 오퍼레이션들을 포함하는, 집적 회로.

청구항 17

제 14 항에 있어서,

제 2 다수의 공유형 MD 레지스터들을 포함하는 제 2 저장 매체를 더 포함하며, 상기 쉐이더 변수들은 상기 버텍스 쉐이더에 대한 입력을 위한 바이패스되지 않은(non-bypassed) 입력 속성들 및 바이패싱된 입력 속성들을 더 포함하며, 상기 오퍼레이션들의 세트는 상기 제 2 다수의 공유형 MD 레지스터들에서 상기 바이패스되지 않은 입력 속성들을 패킹하고 버텍스 쉐이더 입력 파일을 생성하도록 남아있는 임의의 바이패스되지않은 입력 속성들을 상기 제 2 저장 매체에 상주시키기 위한 오퍼레이션들을 포함하는, 집적 회로.

청구항 18

제 17 항에 있어서,

상기 버텍스 캐시는 제 2 다수의 MD 캐시 레지스터들을 더 포함하며, 상기 저장 매체는 그 내부에 상기 바이패싱된 입력 속성들의 M 콤포넌트들을 연이어 패킹하고 채워질 때 상기 버텍스 캐시의 상기 제 2 다수의 MD 캐시 레지스터들의 각각의 MD 캐시 레지스터에 제 2 패킹 버퍼의 상기 MD 레지스터의 콘텐츠들을 전송하기 위한 MD 레지스터를 갖는 제 2 패킹 버퍼를 포함하는, 집적 회로.

청구항 19

제 14 항에 있어서,

제 2 다수의 공유형 MD 레지스터들을 포함하는 제 2 저장 매체를 더 포함하며, 상기 쉐이더 변수는 상기 버텍스 쉐이더에 대한 입력을 위한 입력 속성들의 세트를 포함하며, 패킹되는 경우, 상기 오퍼레이션들의 세트는 나머지 패킹되지 않은(unpacked) 입력 속성들의 콤포넌트들이 M을 초과할 때까지 상기 제 2 다수의 공유형 MD 레지스터들의 상기 입력 속성들의 세트를 패킹하고 버텍스 쉐이더 입력 파일이 생성되도록 상기 나머지 입력 속성들을 상기 제 2 저장 매체에 상주시키기 위한 오퍼레이션들을 포함하는, 집적 회로.

청구항 20

제 11 항에 있어서,

상기 프로세싱 유니트는 셀룰러 전화, 무선 디바이스, 무선 통신 디바이스, 비디오 게임 콘솔, PDA(personal digital assistant), 랩톱 컴퓨터, 및 오디오/비디오-인에이블드(enabled) 디바이스 중 하나의 일부인, 집적 회로.

청구항 21

무선 디바이스로서,

다수의 공유형 M-차원(MD) 레지스터들을 포함하며 쉐이더 변수들의 세트를 저장하기 위한 저장 수단 ; 및

콤포넌트들의 합이 M인 상기 하나 이상의 쉐이더 변수들의 세트를 각각의 공유형 MD 레지스터에 패킹하기 위한 패킹 수단

을 포함하는, 무선 디바이스.

청구항 22

제 21 항에 있어서,

상기 패킹 수단은 2개의 2D 벡터 쉐이더 변수들; 3D 벡터 쉐이더 변수 및 플로트(float) 쉐이더 변수; 2D 벡터 쉐이더 변수 및 2개의 상이한 플로트 쉐이더 변수들; 및 4개의 플로트 쉐이더 변수들 중 적어도 하나를 패킹하기 위한 수단을 포함하는, 무선 디바이스.

청구항 23

제 21 항에 있어서,

상기 쉐이더 변수들은 버텍스(vertex) 쉐이더로부터 출력 가변치들(varyings)의 세트를 포함하며, 상기 패킹 수단은 상기 다수의 공유형 MD 레지스터들의 상기 출력 가변치들의 세트를 패킹하고 버텍스 쉐이더 출력 파일을 생성하기 위해 임의의 나머지 가변치들을 상기 저장 매체에 상주시키기 위한 가변 패킹 수단을 포함하는, 무선 디바이스.

청구항 24

제 21 항에 있어서,

상기 쉐이더 변수들의 세트는 버텍스 쉐이더에 대한 입력 속성들의 세트를 포함하며, 상기 패킹 수단은 상기 다수의 공유형 MD 레지스터들에서 상기 입력 속성들의 세트를 패킹하기 위한 속성 패킹 수단 및 버텍스 쉐이더 입력 파일을 생성하기 위해 임의의 남아있는 입력 속성들을 상기 저장 수단에 상주시키기 위한 상주 수단(populating means)을 포함하는, 무선 디바이스.

청구항 25

제 21 항에 있어서,

상기 패킹 수단은 셀룰러 전화, 무선 디바이스, 무선 통신 디바이스, 비디오 게임 콘솔, PDA(personal digital assistant), 랩톱 컴퓨터, 및 오디오/비디오-인에이블드(enabled) 디바이스 중 하나의 일부인, 무선 디바이스.

청구항 26

명령들을 포함하는 컴퓨터 판독가능 매체를 포함하는 컴퓨터 프로그램 물건으로서,

상기 명령들은 컴퓨터가 벡터들의 합이 M과 같은 쉐이더 변수들의 세트의 하나 이상의 쉐이더 변수들을 다수의 공유형 MD 벡터 레지스터들의 각각의 공유형 M-차원(MD) 벡터 레지스터에 패킹하게 하는, 컴퓨터 프로그램 물건.

청구항 27

제 26 항에 있어서,

상기 쉐이더 변수들의 세트는 버텍스 쉐이더로부터 출력 가변치들의 세트를 포함하며, 상기 명령들은 상기 컴퓨터가 상기 출력 가변치들의 세트를 패킹하고 남아있는 벡터 레지스터들을 남아있는 출력 가변치들과 상주시키게 하는, 컴퓨터 프로그램 물건.

청구항 28

제 27 항에 있어서,

상기 컴퓨터가 프래그먼트 쉐이더에 대한 입력 가변치들의 세트와 상기 패킹된 출력 가변치들의 세트를 링크시키게 하는 명령들을 더 포함하는, 컴퓨터 프로그램 물건.

청구항 29

제 28 항에 있어서,

상기 링크시키게 하는 명령은 상기 컴퓨터가 상기 프래그먼트 쉐이더에 대한 상기 입력 가변치들의 세트의 가변치 심볼 이름들에 대응하게 상기 버텍스 쉐이더로부터 상기 패킹된 출력 가변치들의 세트에 해당하는 가변치 심볼 이름들을 매칭시키게 하는 명령들을 포함하는, 컴퓨터 프로그램 물건.

청구항 30

제 26 항에 있어서,

상기 쉐이더 변수들의 세트는 버텍스 쉐이더에 대한 입력 속성들의 세트를 포함하며, 상기 명령들은 상기 컴퓨터가 상기 입력 속성들의 세트를 패킹하고 남아있는 입력 속성들과 남아있는 벡터 레지스터들을 상주시키게 하는 명령들을 포함하는, 컴퓨터 프로그램 물건.

청구항 31

벡터들의 합이 M인 쉐이더 변수들의 세트에서 하나 이상의 쉐이더 변수들을 다수의 공유형 MD 벡터 레지스터들의 각각의 공유형 M-차원(MD) 벡터 레지스터에 패킹하는 단계; 및

임의의 남아있는 쉐이더 변수들이 패킹불가능할 때까지 상기 다수의 공유형 MD 벡터 레지스터들의 패킹을 반복하는 단계

를 포함하는, 방법.

청구항 32

제 31 항에 있어서,

상기 쉐이더 변수들의 세트는 버텍스 쉐이더로부터의 출력 가변치들의 세트를 포함하며, 상기 패킹하는 단계는 상기 다수의 공유형 M-차원(MD) 레지스터들의 상기 출력 가변치들의 세트를 패킹하는 단계를 포함하는, 방법.

청구항 33

제 31 항에 있어서,

상기 패킹된 출력 가변치들의 세트 및 남아있는 출력 가변치들을 갖는 버텍스 쉐이더 출력 파일을 생성하는 단계;

패킹 버퍼의 MD 임시 레지스터에서 상기 버텍스 쉐이더 출력 파일의 M 콤포넌트들을 연이어 패킹하는 단계; 및

채워질 때 버텍스 캐시의 각각의 MD 캐시 레지스터에 상기 MD 임시 레지스터의 콘텐츠들을 전송하는 단계

를 더 포함하는, 방법.

청구항 34

제 31 항에 있어서,

상기 쉐이더 변수들의 세트는 버텍스 쉐이더에 대해 바이패스되지 않은 입력 속성들의 세트를 포함하며, 상기 패킹하는 단계는 상기 다수의 공유형 MD 레지스터들의 상기 입력 속성들의 세트를 패킹하는 단계를 포함하는, 방법.

청구항 35

제 34 항에 있어서,

패킹 버퍼의 MD 임시 레지스터에서 바이패싱된 입력 속성들의 M 콤포넌트들을 연이어 패킹하는 단계; 및

채워질 때 버텍스 캐시의 각각의 MD 캐시 레지스터에 상기 MD 임시 레지스터의 콘텐츠들을 전송하는 단계

를 더 포함하는, 방법.

청구항 36

프로세서로서,

다수의 공유형 M-차원(MD) 레지스터들을 포함하는 저장 매체; 및

콤포넌트들의 합이 M과 같은 하나 이상의 쉐이더 변수들을 각각의 공유형 MD 레지스터에 패킹하는 오퍼레이션들의 세트를 실행하는 집적 회로

를 포함하는, 프로세서.

청구항 37

제 36 항에 있어서,

상기 쉐이더 변수들은 버텍스 쉐이더로부터의 출력 가변치들의 세트를 포함하며, 상기 오퍼레이션들의 세트는 상기 다수의 공유형 M-차원(MD) 레지스터들의 상기 출력 가변치들의 세트를 패킹하고 버텍스 쉐이더 출력 파일을 생성하기 위해 남아있는 임의의 가변치들을 상기 저장 매체에 상주시키게 하는 오퍼레이션들을 포함하는, 프로세서.

청구항 38

제 37 항에 있어서,

다수의 MD 캐시 레지스터들을 포함하는 버텍스 캐시를 더 포함하며, 상기 저장 매체는 그 내부에 상기 버텍스 쉐이더 출력 파일의 M 콤포넌트들을 연이어 패킹하고 채워질 때 상기 버텍스 캐시의 각각의 MD 캐시 레지스터에 패킹 버퍼의 상기 MD 레지스터의 콘텐츠들을 전송하기 위한 MD 레지스터를 갖는 패킹 버퍼를 포함하는, 프로세서.

명 세 서

기 술 분 야

[0001] [0001] 본 발명은 일반적으로 그래픽 프로세싱 분야에 관한 것으로, 보다 특정하게는 그래픽 시스템에서의 패킹 및 링킹을 변화시키기 위한 기술들에 관한 것이다.

배 경 기 술

[0002] [0002] 퍼블릭 그래픽 규격 OpenGL 또는 OpenGL ES는 버텍스 쉐이더(vertex shader) 및 프래그먼트(fragment) 쉐이더를 사용하여 버텍스 당(per-vertex) 및 픽셀-당(per-pixel) 오퍼레이션들(operations) 동안 변경될 수 있는 고정된 기능을 갖는다. 버텍스 및 단편 쉐이더들은 디폴트 OpenGL 기능으로 달성될 수 없는 특정 효과들을 랜더링하기 위해(to render) 개발되었다.

[0003] [0003] 도 1을 참조로, 쉐이더들을 갖는 그래픽 프로세싱 유니트(GPU)에서 통상의 파이프라인 스테이지들의 일반적 흐름도가 도시된다. 여기서는 3개의 주요 파이프라인 스테이지들이 제공된다 : 블록 S10으로 표시된 버텍스 쉐이더, 블록 S12로 표시된 원시 어셈블러(primitive assembler) 및 래스터라이저(rasterizer), 및 블록 S14로 표시된 프래그먼트 쉐이더. 추가 블록(S16)은 샘플-당 오퍼레이션들(per-sample operations)을 위해 제공된다.

[0004] [0004] 버텍스 쉐이더(VS)(S10)는 기하학적 오브젝트(geometric object)의 각각의 버텍스에 대해 실행되는 프로그램 또는 컴퓨터 프로그램 물건(product)이다. VS(S10)의 입력은 블록 A2에 표시된 속성들(attribute)로 불린다. 또한, VS(S10)는 다수의 버텍스 유니폼들(vertex uniform)(0~95)(즉, 96개의 버텍스 유니폼들)을 포함할 수 있는 버텍스 유니폼들(VU2)의 입력들로서 허용된다. VS(S10)로부터의 출력들 및 원시 어셈블러 & 래스터라이저(S12)는 일반적으로 블록 V3로 표시된 가변치들(varyings)로 불리며 (VS 출력들을 저장하는) 버텍스 캐시(vertex cache) 또는 (래스터라이저 출력들을 저장하는) 다른 저장 매체 내에 제공된다. 가변치들(V3)은 기하학 오브젝트의 트라이앵글들의 픽셀들과 연관된 값들일 수 있다. 픽셀들과 연관된 값들은 기하학 오브젝트의 트라이앵글들의 버텍스들과 연관된 VS 결과치들(results)에 기초하여 계산되는 원시 어셈블러 & 래스터라이저(S12)의 결과치들이다. 버텍스들과 연관된 VS 결과치들은 및 픽셀들과 연관된 가변치들(V3)은 동일한 이름들 또는 ID들, 타입들 및 순서를 갖는다. 픽셀들과 연관된 가변치들(V3)은 프래그먼트 쉐이더(FS)(S14)에 대한 입

력들이다. 또한, FS(S14)는 일반적으로 다수의(이를 테면, 16개의) 프래그먼트 유니폼들을 포함하는 프래그먼트 유니폼들(FU3)의 입력들로서 허용된다.

[0005] [0005] 도 2는 쉐이더들을 갖는 종래의 파이프라인 스테이지들의 일반적 블록 다이어그램을 나타낸다. 그래픽 프로세싱 유니트(GPU) 내의 VS(S10)와 관련하여, 일반적으로 속성들(0~7)을 저장하기 위해 8개의 속성 레지스터들(RA2)이 제공된다. 일반적으로는 속성들(0~7)을 저장하기 위해 8개의 출력 가변 레지스터들(RV3A)가 제공된다. 통상적으로 버텍스 캐시인 가변 레지스터들(RV3A)은 VS 출력들을 저장한다. 일반적으로 가변치들(0~7)을 저장하기 위해 8개의 출력 가변 레지스터들(RV3A)가 제공된다. 가변 레지스터들(RV3B)은 픽셀들과 연관된 가변치들에 해당하는 래스터라이저 결과치들을 저장한다. 속성 레지스터들(RA2) 및 가변 레지스터들(RV3A)은 각각 속성들(0~7)로 인덱싱된 입력 레지스터들 및 가변치들(0~7)로 인덱싱된 출력 가변 레지스터들(RV3)이다. 이러한 레지스터 ID들은 버텍스 쉐이더 및 프래그먼트 쉐이더를 하이 레벨 언어에서 기계 레벨 언어로 컴파일링하는 컴파일러(compiler)에 의해 할당된다. 하이 레벨 언어로 쉐이더 프로그램에서 사용되는 레지스터들은 ID들/인덱스들을 대신하는 이름들에 의해 지명된다. 레지스터 이름들은 애플리케이션 개발자들만이 볼 수 있다. 애플리케이션들은 레지스터 이름들을 통해 레지스터들을 액세스한다. 레지스터 ID들은 GPU 하드웨어(HW)의 VS(S10) 또는 FS(S14)에 의해서만 볼 수 있다. 따라서, VS 입력 심볼 테이블, 출력 심볼 테이블 및 FS 입력 심볼 테이블과 같은 심볼 테이블이 컴파일러에 의해 생성된다. 그러나, VS 입력들 또는 입력 심볼 테이블은 콘텐츠들, ID들 및 이름들과 관련하여 출력들 또는 출력 심볼 테이블과 관련되지 않는다.

[0006] [0006] FS(S14)의 입력들 또는 입력 심볼 테이블은 VS(S10)의 출력들 또는 출력 심볼의 서브세트일 수 있지만, VS 출력들 또는 출력 심볼 테이블은 콘텐츠들 및 이름들과 관련하여 FS(S14)의 입력들 또는 입력 심볼 테이블과 매칭되야 한다.

[0007] [0007] 또한, VS(S10)은 T2로 표시된 텍스처들 뿐만 아니라 저장 매체에 저장된 버텍스 유니폼들(VU2)의 입력들로서 허용된다. 원시 어셈블러 & 래스터라이저(S12)는 파라미터 gl_Position(P) 및 가변치들(0~7)로 인덱싱된 출력 가변 레지스터들(RV3A)의 변수들을 수신한다. 원시 어셈블러 & 래스터라이저(S12)는 파라미터 gl_Position(P) 및 가변치들(0~7)로 인덱싱된 출력 가변 레지스터들(RV3A)의 가변치들을 출력한다. FS(S14)는 T3으로 표시된 텍스처들 뿐만 아니라 저장 매체에 저장된 프래그먼트 유니폼들(FU3) 및 TV3로 표시된 임시 변수들(temporary variables)을 입력으로서 허용한다. FS(S14)는 P로 표시된 파라미터 gl_Position 및 가변치들(0~7)로 인덱싱된 출력 가변 레지스터들(RV3B)의 가변치들을 수신한다. 또한, FS(S14)는 FF로 표시된 추가의 파라미터 gl_Frontfacing 및 PP로 표시된 gl_PointPosition를 수신한다. FS(S14)는 gl_FragColor(FC)를 출력한다. 또한, 속성들 및 가변치들은 쉐이더 변수들로 불린다.

발명의 내용

[0008] [0008] 본 발명에서는 그래픽 파이프라인들에서 패킹 및 링킹을 변화시키기 위한 기술들이 개시된다. 저장기 또는 메모리가 보다 효율적으로 사용되도록 하는데 있어 쉐이더 변수들의 패킹은 모바일 GPU에서 유용하다. 또한, 쉐이더 변수들의 패킹은 트래픽 대역폭을 감소시키고, 전력을 절감하며, 성능을 개선시킬 수 있다.

[0009] [0009] 일 구성으로, 디바이스는 다수의 공유형 M-차원(MD) 레지스터들을 가지는 저장 매체를 포함한다. 또한, 디바이스는 콤포넌트들의 합이 M과 같은 하나 이상의 쉐이더 변수들을 각각의 공유형 MD 레지스터에 패킹하기 위한 오퍼레이션들의 세트를 구현하기 위한 프로세싱 유니트를 포함한다.

[0010] [0010] 또 다른 구성으로, 집적 회로는 다수의 공유형 M-차원(MD) 레지스터들을 가지는 저장 매체를 포함한다. 또한, 집적 회로는 콤포넌트들의 합이 M과 같은 하나 이상의 쉐이더 변수들을 각각의 공유형 MD 레지스터에 패킹하기 위한 오퍼레이션들의 세트를 구현하기 위한 프로세싱 유니트를 포함한다.

[0011] [0011] 또 다른 구성은 컴퓨터 프로그램 물건을 포함한다. 컴퓨터 프로그램 물건은 컴퓨터가 콤포넌트들의 합이 M인 쉐이더 변수들의 세트의 하나 이상의 쉐이더 변수들을 다수의 공유형 MD 벡터 레지스터들의 각각의 공유형 M-차원(MD) 벡터 레지스터에 패킹하게 하기 위한 명령들을 포함하는 컴퓨터 판독가능 매체를 포함한다.

[0012] [0012] 또 다른 구성은 다수의 공유형 M-차원(MD) 레지스터들을 가지는 저장 매체를 포함하는 프로세서를 포함한다. 또한, 프로세서는 콤포넌트들의 합이 M인 하나 이상의 쉐이더 변수들을 각각의 공유형 MD 레지스터에 패킹하기 위한 오퍼레이션들의 세트를 구현하기 위한 집적 회로를 포함한다.

[0013] [0013] 추가의 양상들은 특히 첨부된 도면들과 함께 상세한 설명을 통해 보다 명확해질 것이다.

도면의 간단한 설명

[0014] [0014] 본 발명의 양상들 및 구성들은 도면들과 함께 하기 상세한 설명을 통해 명확해질 것이며, 도면들에서 동일한 참조 부호들은 대응되는 구성들로 참조된다.

[0015] [0015] 도 1은 쉐이더들을 갖는 그래픽 프로세싱 유니트에서 통상의 파이프라인 스테이지들의 일반적 순서도를 나타낸다.

[0016] [0016] 도 2는 쉐이더들을 갖는 통상의 파이프라인 스테이지들의 일반적 블록도를 나타낸다.

[0017] [0017] 도 3은 무선 디바이스의 블록도를 나타낸다.

[0018] [0018] 도 4는 버텍스 쉐이더 및 패킹 오퍼레이션들을 위한 그래픽 프로세싱 유니트(GPU)의 일반적 블록도를 나타낸다.

[0019] [0019] 도 5는 프래그먼트 쉐이더 및 링킹 오퍼레이션들을 이용하는 그래픽 프로세싱 유니트(GPU)의 일반적 블록도를 나타낸다.

[0020] [0020] 도 6은 드라이버의 일반적 블록도를 나타낸다.

[0021] [0021] 도 7은 2-레벨 쉐이더 변수 패킹 프로세스의 일반적 순서도를 나타낸다.

[0022] [0022] 도 8A 및 8B는 바이패싱 속성들이 제거되기 이전 및 이후의 버텍스 쉐이더 프로그램을 나타낸다.

[0023] [0023] 도 9A 및 9B는 바이패싱 속성들이 제거되기 이전 및 이후의 또 다른 버텍스 쉐이더 프로그램을 나타낸다.

[0024] [0024] 도 10A 및 10B는 바이패싱 속성들이 제거되기 이전 및 이후의 또 다른 쉐이더 프로그램을 나타낸다.

[0025] [0025] 도 11A 및 11B는 바이패싱 속성들이 제거되기 이전 및 이후의 또 다른 버텍스 쉐이더 프로그램을 나타낸다.

[0026] [0026] 도 12A 및 12B는 바이패싱 속성들이 제거되기 이전 및 이후의 또 다른 버텍스 쉐이더 프로그램을 나타낸다.

[0027] [0027] 도 13A-13C는 속성들의 바이패싱과 조합된 쉐이더 변수들의 패킹 프로세스의 일반적 순서도를 나타낸다.

[0028] [0028] 도 14는 링킹 프로세스의 일반적 흐름도를 나타낸다.

[0029] [0029] 도면들에서 이미지들은 스케일대로 도시된 것이 아니며 예시를 목적으로 간략화되었다. 이해를 돕기 위해, 가능한 동일한 참조 부호들이 사용되며, 이는 필요시 부재들을 구별하기 위해 첨자가 부가될 수 있다는 것을 제외하고, 도면들에서 공통되는 동일한 부재들을 표시하기 위한 것이다.

[0030] [0030] 첨부된 도면들은 본 발명의 예시적 구성들을 나타내며, 이는 등가적인 다른 유효한 구성들을 허용할 수 있는 본 발명의 범주를 제한하고자 하는 것은 아니다. 일 구성의 특징들 또는 단계들은 추가 설명없이 다른 구성들에서 적절히 통합될 수 있다는 것이 고려된다.

[0031] [0031] 하기 다양한 구성들에서, 순서도들은 도시된 순서로 수행되거나 또는 이들 블록들 또는 이들의 부분들은 동시적으로, 독립적으로(in parallel), 또는 상이한 순서로 수행될 수 있다.

발명을 실시하기 위한 구체적인 내용

[0015] [0032] 본 발명에서 사용되는 "예시적"이란 용어는 예, 예증 또는 예시로서 "기능한다"는 것을 의미한다. "예시적"으로 본 발명에서 개시되는 임의의 구성 또는 설계는 다른 구성들 또는 설계들에 대해 선호되는 또는 바람직한 것으로 간주될 필요는 없다.

[0016] [0033] 본 발명에 사용되는 기술들은 무선 통신들, 컴퓨팅, 퍼스널 일렉트로닉스 등에 사용될 수 있다. 무선 통신에 대한 기술들의 예시적인 사용이 하기에서 개시된다.

[0017] [0034] 도 3은 무선 통신 시스템에서 사용되는 무선 디바이스(10)의 구성에 대한 블록도를 나타낸다. 무선 디바이스(10)는 셀룰러 또는 카메라 폰, 단말, 핸드셋, PDA, 또는 소정의 다른 디바이스일 수 있다. 무선 통신 시스템은 코드 분할 멀티플 액세스(CDMA) 시스템, 모바일 통신 시스템을 위한 글로벌 시스템(GSM) 시스템, 또는

소정의 다른 시스템일 수 있다.

[0018] [0035] 무선 디바이스(10)는 수신 경로 및 전송 경로를 통해 양방향성 통신을 제공할 수 있다. 수신 경로에서, 기지국들에 의해 전송된 신호들은 안테나(12)에 의해 수신되며 수신기(RCVR)(14)에 제공된다. 수신기(14)는 수신된 신호를 조정하고 디지털화하며 추가의 프로세싱을 위한 디지털 섹션(20)에 샘플들을 제공한다. 전송 경로에서, 송신기(TMTR)(16)는 디지털 섹션(20)으로부터 전송되는 데이터를 수신하며, 데이터를 처리하고 조정하며, 안테나(12)를 통해 기지국들로 전송되는 변조된 신호를 생성한다.

[0019] [0036] 디지털 섹션(20)은 다양한 프로세싱, 인터페이스 및 메모리 유니트들, 이를 테면 예를 들어 모뎀 프로세서(22), 비디오 프로세서(24), 제어기/프로세서(26), 디스플레이 프로세서(28), ARM/DSP(32), 그래픽 프로세싱 유니트(GPU)(34), 내부 메모리(36), 및 외부 버스 인터페이스(EBI)(38)를 포함한다. 모뎀 프로세서(22)는 데이터 전송 및 수신에 대한 프로세싱(이를 테면, 인코딩, 변조, 복조, 및 디코딩)을 수행한다. 비디오 프로세서(24)는 캠코더, 비디오 플레이백, 및 비디오 컨퍼런싱과 같은 비디오 애플리케이션들에 대해 비디오 콘텐츠(이를 테면, 스틸 이미지들, 동영상들, 무빙 텍스트들)상에서 프로세싱을 수행한다. 제어기/프로세서(26)는 디지털 섹션(20) 내의 다양한 프로세싱 및 인터페이스 유니트들의 오퍼레이션을 지시할 수 있다. 디스플레이 프로세서(28)는 디스플레이 유니트(30) 상에서 비디오들, 그래픽들 및 텍스트들의 디스플레이를 용이하게 하기 위한 프로세싱을 수행한다. ARM/DSP(32)는 무선 디바이스(10)에 대한 다양한 형태들의 프로세싱을 수행할 수 있다. 그래픽 프로세싱 유니트(34)는 그래픽 파이프라인의 그래픽 프로세싱을 수행한다.

[0020] [0037] 본 발명에 개시된 기술들은 디지털 섹션(20) 내의 임의의 프로세서들 이를 테면, 그래픽 프로세싱 유니트(34)에 대해 이용될 수 있다. 내부 메모리(36)는 디지털 섹션(20) 내의 다양한 유니트들에 대한 데이터 및/또는 명령들을 저장한다. EBI(38)는 버스 또는 데이터 라인(DL)을 따라 메인 메모리(40)와 디지털 섹션(20)(이를 테면 내부 메모리(36)) 사이에서 데이터의 전송을 용이하게 한다.

[0021] [0038] 디지털 섹션(20)은 하나 이상의 DSP들, 마이크로프로세서들, RISC들 등으로 구현될 수 있다. 또한 디지털 섹션(20)은 하나 이상의 ASIC들 또는 소정의 다른 형태의 집적회로(IC)들로 제조될 수 있다.

[0022] [0039] 본 발명에 개시되는 기술들은 다양한 하드웨어 유니트들에서 구현될 수 있다. 예를 들어, 기술들은 ASIC들, DSP들, RISC들, ARM들, 디지털 신호 프로세싱 디바이스(DSPD)들, 프로그램가능 로직 디바이스(PLD)들, 필드 프로그램가능 게이트 어레이(FPGA)들, 프로세서들, 제어기들, 마이크로-제어기들, 마이크로프로세서들, 및 다른 일렉트로닉 유니트들에서 구현될 수 있다.

[0023] [0040] 또한, GPU(34)는 퍼블릭 그래픽 규격, 이를 테면 OpenGL2.0, OpenGL ES2.0, 또는 D3D9.0와 호환될 수 있다.

[0024] [0041] 도 4는 버텍스 쉐이더 및 패킹 오러페이션들을 위한 그래픽 프로세싱 유니트(GPU)(34)의 일반적 블록도를 도시한다. GPU(34)는 VS 입력 레지스터 파일(56)로 다수의 속성들을 출력하는 스트림 디코더(50)를 포함한다. 이러한 속성들은 버텍스 쉐이더(VS)(60)에 의해 허용된다. VS(60)의 출력은 VS 출력 레지스터 파일(57)에 저장되는 가변치들(varyings)을 포함한다. 인식될 수 있듯이, "레지스터(register)" 파일은 정보를 저장하는 저장 매체와 같은 하드웨어 콤포넌트이다. 일례로, "VS 입력 레지스터 파일"은 VS(6)로 전송되는 입력 파일을 저장한다. 단순화를 위해, VS 입력 레지스터 파일(56)을 참조할 때, 대부분이 경우, VS(60)에 대한 "입력 파일" 및/또는 "입력 파일"을 저장하기 위한 하드웨어가 참조된다. 마찬가지로, 단순화를 위해 VS 출력 레비스터 파일(57)이 참조될 때, 대부분의 경우, VS(60)로부터의 "출력 파일" 및/또는 "출력 파일"을 저장하기 위한 하드웨어가 참조된다. 이후 보다 상세히 개시되는 것처럼, 이러한 가변치들은 제 1-레벨 가변치 패킹을 위해 컴파일러(62)(도 6)에 의해 지능적으로(intelligently) 패킹된다. VS 출력 레지스터 파일(57)에서의 가변치들은 제 2-레벨 가변 패킹에서의 가변치들을 패킹하는 일련의(series) 또는 체인 시퀀스로 연속적으로 버퍼(58)를 패킹하도록 전송된다. 패킹 버퍼(58)가 채워짐에 따라, 패킹된 가변치들은 버텍스 캐시(cache)(57)에 저장된다.

[0025] [0042] 하기 설명에서 볼 수 있듯이, VS 출력 레지스터 파일(57) 및 VS 입력 레지스터 파일(56) 각각은 다수의 공유형 M-차원(MD) 레지스터들을 포함한다. 패킹 버퍼들(58, 52) 각각은 적어도 하나의 공유형 M-차원(MD) 레지스터를 포함한다.

[0026] [0043] 도 4의 구성에서, 스트림 디코더(50)는 2개의 스트림들, 즉 바이패싱된(bypassed) 스트림 및 바이패스되지 않은(non-bypassed) 스트림을 생성한다. 바이패스되지 않은 스트림은 VS 입력 레지스터 파일(56)로 전송되며 바람직하게는 표 1에 도시된 방식으로 패킹된다. 바이패싱된 속성들은 패킹 버퍼(52)에 패킹된다. 바이패

싱된 속성들은 이후에 도 8A, 8B, 9A, 9B, 10A, 10B, 11A, 11B, 12A 및 12B와 관련하여 상세히 개시된다.

[0027] [0044] 도 5는 프래그먼트 쉐이더 및 링킹 오퍼레이션들을 갖는 그래픽 프로세싱 유니트의 일반적 블럭도를 도시한다. 패킹된 가변치들은 버텍스 캐시(54)에 저장된다. 원시 어셈블러 및 래스터라이저(90)는 버텍스 캐시(54)의 가변치들을 입력으로서 허용한다. 원시 어셈블러 및 래스터라이저(90)는 패킹된 가변치들을 가변 버퍼(92)로 출력한다. 링킹 유니트(88)는 가변 리맵핑 및 로딩 모듈(84)에 의해 이용되는 링킹 명령들(82)의 세트를 포함한다. 도 6의 링커(80)는 드라이버(61)에 의해 도 5의 링킹 명령들(82)에 대해 저장기에 로딩되는 링킹 테이블(86)을 생성한다. 링킹 테이블(86)의 예는 하기 개시되는 표 4 및 6에 도시되며, 이는 VS 출력 심볼 테이블(표 2)의 패킹된 가변치들을 FS 입력 심볼 테이블(표 2)에 링킹한다. FS 입력 심볼 테이블은 VS 출력 심볼 테이블 보다 적은(less) 심볼들을 가질 수 있다. 링킹 유니트(88)에 의해 링킹 프로세스가 수행된 이후, 가변 리맵핑 및 로딩 모듈(84)로부터의 가변치들은 프래그먼트 쉐이더(FS)(70)에 의해 사용되도록 FS 입력 레지스터 파일(79)로 전송된다.

[0028] [0045] 도 6은 드라이버의 일반적 블록도를 도시한다. 드라이버(61)는 컴파일러(62) 및 링커(80)를 포함한다. 컴파일러(62)는 VS 입력 심볼 테이블(64) 및 VS 출력 심볼 테이블(66)을 생성한다. 예시적 VS 입력 심볼 테이블이 하기 표 1에 도시된다. 예시적 VS 출력 심볼 테이블이 하기 표 2에 도시된다. 컴파일러(62)는 FS 입력 심볼 테이블(74)에서와는 상이한 VS 출력 심볼 테이블(66)에서의 동일한 심볼 ID를 할당할 수 있으며, 이는 컴파일러(62)가 버텍스 쉐이더(60) 및 프래그먼트 쉐이더(70)를 독립적으로 컴파일할 수 있기 때문이다. 따라서, 양쪽(both) 테이블들에서 동일한 심볼을 탐색함으로써 FS 입력 심볼 테이블(74)에서의 레지스터 ID들과 VS 출력 심볼 테이블(66)에서의 레지스터 ID들 간의 맵핑을 수행하도록 드라이버(61)에 대한 링커(80)가 제공된다. 링커(80)는 동일 가변 심볼에 대해 프래그먼트 쉐이더(70)의 입력 레지스터 파일(79)에서 해당 입력 레지스터에 가변치(버텍스 캐시(54) 또는 가변 버퍼(92)의 위치에 해당)를 로딩하기 위해 GPU(34)와 통신한다.

[0029] [0046] 드라이버(61)는 명령들의 세트를 갖는 소프트웨어 드라이버이다. 컴파일러(62) 및 링커(80)는 CPU(32) 또는 제어기/프로세서(26) 상에서 동작하는 소프트웨어 드라이버(61)의 부품들(parts)인 반면, GPU(34)는 드라이버(61)에 의해 지시되는 특정 코-프로세서(co-processor)이다.

[0030] [0047] 표 1에 도시된 VS 입력 심볼 테이블은 하기의 엔트리들을 포함한다 : 속성 이름, 타입, 오리지널 할당된 속성 입력 레지스터 ID, 오리지널 마스크, 새롭게 할당된 속성 입력 레지스터 ID 및 새로운 마스크. 표 2에 도시되는 VS 출력 심볼 테이블은 하기의 엔트리들을 포함한다 : 가변치 이름, 타입, 오리지널 할당된 가변 출력 레지스터 ID, 오리지널 마스크, 새롭게 할당된 가변 출력 레지스터 ID 및 새로운 마스크. 테이블들에서의 마스크는 GPU(34)의 하드웨어(HW)에 할당된 디폴트 MD(M=4) 벡터 레지스터 저장기에 해당하는 속성 벡터들 또는 가변 벡터들에 대한 유효(valid) 콤포넌트들을 나타낸다. 오리지널 할당된 ID들 및 마스크, 그리고 새롭게 할당된 ID들 및 마스크 모두는 단지 예시를 위해 하기 테이블들에 함께 제시된다. 실제로, 오리지널 할당된 ID들 및 마스크는 일시적 결과일 수 있으며 오퍼레이션들 동안 동일한 저장 위치를 사용함으로써 새롭게 할당된 ID 및 마스크가 될 수 있다.

[0031] [0048] 컴파일러(62)는 gl_FragColor FC(도 2)로서 표시된 FS 출력(76) 및 FS 입력 심볼 테이블(74)을 생성한다. 표 2에 도시된 FS 입력 심볼 테이블(74)은 하기의 엔트리들을 포함한다: 가변치 이름, 타입, 오리지널 할당된 가변 입력 레지스터 ID, 오리지널 마스크, 새롭게 할당된 가변 입력 레지스터 ID 및 새로운 마스크.

[0032] [0049] 하기 표 1 및 표 2에서, 마지막 2개의 컬럼들은 이후 개시되는 패킹 프로세스에 따라 새롭게 업데이트된다.

표 1

[0033] VS 입력 심볼 테이블

속성 이름	타입	오리지널 할당된 속성 입력 레지스터 ID	오리지널 마스크	새롭게 할당된 속성 입력 레지스터 ID	새로운 마스크
position0	Float vector4	0	1111	0	1111
position1	Float vector3	1	0111	1	0111
Weight	Float	5	0001	2	1000
Normal	Short float vector3	2	0111	2	0111

Texcoord0	Float vector2	3	0011	3	0011
Texcoord1	Float vector2	4	0011	3	1100

표 2

[0034] VS 출력 심볼 테이블

가변치 이름	타입	오리지널 할당된 가변 출력 레지스터 ID	오리지널 마스크	새롭게 할당된 가변 출력 레지스터ID	새로운 마스크
position	Float vector4	1	1111	1	1111
color0	Float vector4	2	1111	2	1111
color1	Float vector3	3	0111	3	0111
Texcoord0	Float vector2	0	0011	0	0011
Texcoord1	Float vector2	5	0011	0	1100
Texcoord2	Float vector3	4	0111	4	0111

표 3

[0035] FS 입력 심볼 테이블

가변치 이름	타입	오리지널 할당된 가변 입력 레지스터 ID	오리지널 마스크	새롭게 할당된 가변 입력 레지스터 ID	새로운 마스크
Color0	Float vector4	0	1111	2	1111
Color1	Float vector3	1	0111	0	0111
texcoord0	Float vector2	3	0011	1	0011
texcoord1	Float vector2	2	0011	1	1100

표 4

[0036] VS 출력들 및 FS 입력들에 대한 링크 테이블

가변치 이름	오리지널 할당된 VS 가변 출력 레지스터 ID	오리지널 할당된 FS 가변 입력 레지스터 ID	오리지널 마스크	새롭게 할당된 VS 가변 출력 레지스터 ID	새롭게 할당된 FS 가변 입력 레지스터 ID	새로운 마스크
Position	1			1		
color0	2	2	1111	2	2	1111
color1	3	3	0111	3	0	0111
Texcoord0	0	1	0011	0	1	0011
Texcoord1	5	1	1100	0	1	1100
Texcoord2	4			4		

[0037] [0050] 가변치들은 플로트들(floats), 2차원(2D) 벡터들, 3차원(3D) 벡터들, 4차원(4D) 벡터들, 어레이 및 2D/3D/4D 매트릭스 등일 수 있다. OpenGL ES 쉐이딩 랭기지 스펙(shading language specification)은 모바일 GPU 34에서 지원되는 적어도 32 가변 콤포넌트들을 요구한다. 각각의 가변치들은 상이한 크기를 가지며 통상적으로 그 자신의 레지스터/버퍼 공간을 취한다. 버텍스 캐시(54)에서, 레지스터는 통상적으로 4D 벡터이다. 부가적으로, VS 입력 레지스터 파일(56)에 해당하는 레지스터들 및 VS 출력 레지스터 파일(57)에 해당하는 레지스터들은 통상적으로 4D 벡터이다. 가변 패킹은 각각의 버텍스 또는 픽셀에 대한 연속 공간(continuous space)에서 서로 엄격한(tightly) 상이한 가변치들을 구성한다. 예를 들어, 본 발명에서 개시되는 가변 패킹(varying packing)은 2개의 2D 벡터들을 4D 벡터 레지스터로 구성한다. 또 다른 예에서, 가변 패킹은 3D 벡터 및 플로트

(ID)를 4D 벡터 레지스터로 구성한다. 이들의 엄격한(tightly) 패킹 없이도, 이들은 느슨하게(loosely) 저장될 수 있다.

[0038] [0051] 상기 설명은 가변치들에 관한 것이다. 그러나, 가변치들 이외에, 속성들도 패킹될 수 있다.

[0039] [0052] 도 7은 2-레벨 쉐이더 변수들(variables) 패킹 프로세스(100)의 일반적 흐름도를 도시한다. 프로세스(100)는 블록 102에서 시작되며, 여기서는 컴파일러(62)에 의해 지시된 지능적 패킹(intelligent packing)이 이루어진다. 블록 102에서, 콤포넌트들의 합이 M과 같은 2개 이상의 쉐이더 변수들은 공유형 M-차원(MD) 벡터 레지스터에 할당된다. 예시를 위해, VS 출력 레지스터 파일(57)은 컬럼들 및 로우들로 도시된다. 각각의 로우는 X, Y, Z 및 W로 표시된 4개의 블록들을 포함한다. 블록 102는 블록 104로 이어지며, 여기서는 버텍스 캐시(54)의 NxM 저장 매체 블록에서 일련의 연속적으로 VS 출력 레지스터 파일(57)의 쉐이더 변수들을 패킹하는 패킹 버퍼(58)에서 하드웨어 가변 패킹이 이루어진다. 도 7의 쉐이더 변수들은 가변치들이다.

[0040] [0053] 하기 설명에서 볼 수 있듯이, 바이패싱된 속성들은 도 13A와 관련하여 개시되는 블록 104의 프로세스와 유사하게 패킹 버퍼(52)에서 패킹된다. 바이패스되지 않은 속성들은 블록 102과 관련하여 앞서 개시된 프로세스를 사용하여 패킹될 수 있다. 따라서, 패킹 프로세스(100)의 스테이지들이 속성들에 대해 이용될 수 있다. 따라서, 쉐이더 변수들은 가변치들 및 속성들을 포함한다.

[0041] **제 1 레벨: 컴파일러 레벨 패킹**

[0042] [0054] 컴파일러(62)에 의해 지시되는 블록 102에서의 지능적 패킹에 대한 하기 설명은 상기 표 1 및 2와 관련하여 개시된다. 지능적 패킹은 쉐이더 변수들(가변치들 및 속성들 모두)에 적용된다. 표 1은 속성 패킹의 예시이며 표 2는 가변 패킹의 예시이다. 컴파일러(62)는 레지스터 ID와 연관된 동일한 또는 공통 MD(M-차원) 벡터 레지스터를 콤포넌트들의 합이 M(M=4)과 같은 2개 이상의 가변치들에 재할당함으로써 바이패스되지 않은 속성들 또는 가변 패킹을 수행하며 이에 따라 마스크를 업데이트한다. 속성들에 대하나 MD 벡터 레지스터는 도 4의 VS 입력 레지스터 파일(56)에 대한 저장기에 해당한다. 가변치들에 대한 MD 벡터 레지스터는 도 4의 VS 출력 레지스터 파일(57)에 대한 저장기에 해당한다. 예시적 구성에서, M=4이고, 따라서 벡터들은 X, Y, Z 및 W로 표시된다. 하지만 보다 많은 또는 보다 적은 차원들을 갖는 다른 구성들도 사용될 수 있다.

[0043] [0055] 마스크는 M-비트 위치들을 갖는다. 따라서, 특정 MD 벡터 레지스터에 대해 각각 재할당된 및/또는 조합된 속성들 또는 가변치들(쉐이더 변수들)과 연관된 마스크는 공유형 MD 벡터 레지스터의 어느 부분(portion)이 이후 회수(recall) 및 사용을 위해 (조합부의) 각각의 개별 속성 또는 가변치에 할당되는지를 표시 또는 구별하기 위해 이용된다.

[0044] [0056] 예를 들어, 특히 상기 표 1을 참조로, 오리지널 할당된 속성 입력 레지스터 ID 컬럼에서, texcoord0 및 texcoord1에는 각각 ID 넘버 3과 4로 표시된 상이한 속성 입력 레지스터들이 오리지널 할당된다. 또한, texcoord0 및 texcoord1에 대한 오리지널 마스크들은 각각 0011 및 0011이다. 컴파일러(62)는 texcoord0 및 texcoord1 모두가 벡터들의 합이 4D(M=4) 벡터들인 2D 벡터들이라는 것을 결정한다. 따라서, 컴파일러(62)는 새롭게 할당된 속성 입력 레지스터 ID 컬럼에서, ID 넘버 3으로 표시된 동일한 속성 레지스터로 texcoord0 및 texcoord1의 패킹을 지시한다. 패킹 동안, texcoord0에는 마스크의 가장 작은 자릿수 비트(lowest significant bit) 0011이 할당될 수 있고 texcoord1에는 표 1의 새로운 마스크 컬럼에 표시된 마스크의 가장 높은 자릿수 비트(lowest significant bit) 1100이 할당될 수 있다. 마스크 0011은 texcoord0에 해당하는 데이터가 MD 벡터 레지스터(3)의 어느 부분에서 발견될 수 있는지를 표시한다. 마찬가지로, 마스크 1100은 texcoord1에 해당하는 데이터가 MD 벡터 레지스터(3)의 어느 부분에서 발견될 수 있는지를 표시한다. 이러한 명명법은 2개 이상의 속성들이 논-오버랩핑(non-overlapping) 방식으로 공통 레지스터를 공유하게 허용한다. 인식될 수 있듯이, 마스크에서 비트들의 수는 차원들에 따라 변할 수 있다.

[0045] [0057] 특히 VS 출력 심볼 표 66의 가변치들 texcoord0 및 texcoord1을 참조로, 이들은 표 2에서 잘 보여지는 것처럼, 새롭게 할당된 가변 레지스터 출력 ID 컬럼에서 표시된 ID 넘버 0을 갖는 동일한 가변 레지스터로 패킹된다. texcoord0에 대한 새로운 마스크는 이전(old) 마스크와 동일한 0011이다. 그러나, texcoord1에 대한 새로운 마스크는 이전 마스크와는 상이한 1100이다. 따라서, 마스크는 M-비트들을 가지며, 각각의 비트는 공유형 MD 벡터 레지스터에서의 위치를 나타낸다.

[0046] [0058] 또 다른 예에서, VS 입력 심볼 표 64의 Weight 및 Normal 속성들은 표 1의 새롭게 할당된 속성 입력 레지스터 ID에 표시된 ID 넘버 2를 갖는 동일한 속성 레지스터에 패킹된다. 컴파일러(62)가 레지스터 ID들 및 새

로운 마스크들의 재할당을 지시한 후, GPU(34)의 하드웨어(HW)는 컴파일러(62)에 의해 지시되는 제 1-레벨 패킹을 완료하는 (업데이트 마스크들을 갖는) 테이블 형태의 명령들에 따라 할당된 레지스터들에 해당 쉐이더 변수들(속성들 또는 가변치들)을 자동으로 로딩한다.

[0047] [0059] 어레이 또는 매트릭스는 논리적으로 2D/3D/4D 벡터 또는 단일 플로트로 분할될 수 있고, 컴파일러(62)에 의해 지시됨에 따라 패킹이 실행될 수 있다. 어레이는 일련의 플로트들, 2D 벡터들, 3D 벡터들 또는 4D 벡터들로 표현될 수 있다. 예를 들어, 10 플로트들의 어레이는 2개의 2D 벡터 플러스 1개의 2D 벡터, 또는 10개의 개별 플로트들로 분할될 수 있다. 각각 2x2 매트릭스는 2개의 2D 벡터들로 분할될 수 있고, 3x3 매트릭스는 3개의 3D 벡터들로 분할될 수 있고, 4x4 매트릭스는 4개의 4D 벡터들로 분할될 수 있다. 따라서, 컴파일러(62)는 2D 벡터 + 2D 벡터; 3D 벡터 + 플로트; 2D 벡터 + 벡터[+ 플로트]; 및 플로트 + 플로트[+ 플로트 [+ 플로트]]의 경우들에 대한 패킹을 지시할 수 있다. 이러한 예들은 4D 벡터 레지스터에 대한 것이다. 차원들의 수에 따라 다른 조합들도 고려된다. 입력 레지스터 파일 및 출력 레지스터 파일의 사용은 제 1 레벨 패킹에 의해 최소화될 수 있다.

[0048] [0060] 컴파일러(62)에 의해 지시되는 패킹 후에, 모든 쉐이더 변수들(가변치들)은 4D(MD) 벡터들에서는 여전히 배열되지 못할 수 있고, 이를 테면 일부 3D 벡터들, 일부 4D 벡터들 등은 존재할 수 있다. 예시적 구성에서, 가변 패킹의 제 2-레벨에 대한 가변 저장기 또는 버텍스 캐시(54)에서 엄격하게 가변치들을 패킹하는 HW에 대한 메커니즘이 수행된다.

[0049] **제 2 레벨: HW 패킹**

[0050] [0061] 가변 저장기 또는 버텍스 캐시(54)에서, 버텍스 또는 픽셀에 대한 모든 가변치들은 NxM 버퍼 블록에 저장된다. N은 가변치들의 수이다; M=4 4D 벡터들을 의미한다. 저장기 블록은 다수의(NxM) 연속적(연이은) 콤포넌트들로서 처리될 수 있다. 32 비트들/콤포넌트들 및 M=4에 대해, 콤포넌트들은 0~((Nx4)-1)로 넘버링될 수 있다. 이를 테면, N = 8, 8x4 저장 매체 블록은 0-31로 넘버링된 32개의 연속적(연이은) 콤포넌트들로서 처리된다.

[0051] [0062] 도 4에서, 패킹 버퍼(58)는 슬롯들의 2xM (M=4) 어레이로서 표현된다. 화살표는 패킹 버퍼(58)의 슬롯들을 채우는 방향을 나타낸다. 패킹 버퍼(58)의 상부 로우는 임시 버퍼(temp buffer)(58A)로 표시되는 반면, 제 2 로우는 작업 버퍼(working buffer)(58B)로 표시된다. 표 5는 HW 패킹 결과들을 나타낸다.

표 5

[0052]

VS 출력 레지스터 파일				패킹	가변 저장기 또는 버텍스 캐시			
V0.x	V0.y	V0.z			0: V0.x	1: V0.y	2:V0.z	3: V1.x
V1.x	V1.y	V1.z	V1.w		4: V1.y	5: V1.z	6: V1.w	7: V2.x
V2.x	V2.y				8: V2.y	9: V3.x	10: V3.y	11: V3.z
V3.x	V3.y	V3.z		→	12: V4.y	13: V5.y	14: V5.z	15: V5.w
	V4.y				16: V6.x	17: V6.y	18: V6.z	19: V6.w
	V5.y	V5.z	V5.w		20: V7.x	21: V7.y	22: V7.z	
V6.x	V6.y	V6.z	V76.w					
V7.x	V7.y	V7.z						

[0053] <VS 출력 레지스터 파일로부터 가변 저장기 또는 버텍스 캐시로 전환될 때의 패킹>

[0054] [0063] 제 2-레벨 패킹은 임시 버퍼(58A)(패킹 버퍼(58)의 제 1 로우)를 연이어 연속적으로 먼저 채움으로써 수행될 수 있다. 패킹 버퍼(58)의 임시 버퍼(58A)를 채운 후에, 임시 버퍼(58A)의 콘텐츠들은 버텍스 캐시(54)의 저장을 위해 전환될 수 있다. 이러한 구성에서, 패킹 버퍼(58)는 임시 버퍼(58A)로서 지정된 제 1 로우의 M 슬롯들 및 작업 버퍼로서 지정된 제 2 로우의 M 슬롯들을 포함한다.

[0055] [0064] 표 5에 개시된 예들을 이용하여, VS 출력 레지스터 파일(57)로부터 V0.x, V0.y 및 V0.z로 표시된 3개의 콤포넌트들을 가지는 가변치 V0를 판독하고 가변치들 V0.x, V0.y 및 V0.z로 연이어 임시 버퍼(58A)(상부 로우) 슬롯들 X, Y, Z 및 W를 채우는 HW 패킹이 시작된다. 알 수 있듯이, 임시 버퍼(58A)의 슬롯(W)은 프리하다. 가변치들 V0.x, V0.y 및 V0.z는 임시 버퍼(58A)가 채워질 때까지 버텍스 캐시(54)로 전송되지 않는다.

[0056] [0065] HW 패킹은 VS 출력 레지스터 파일(57)로부터 V1.x, V1.y, V1.z 및 V1.w로 표시된 4개의 콤포넌트들을 갖는 가변치 V1를 판독하고 임시 버퍼(58A)에 나머지 슬롯(들)을 채움으로써 지속된다. 이 경우, 임시 버퍼

(58A)(상부 로우) 슬롯 W은 가변치 V1.x로 채워진다. 나머지 가변 콤포넌트들(V1.y, V1.z 및 V1.w)은 제 2 로우 또는 작업 버퍼(58B)의 슬롯들 X, Y, 및 Z에 연속적으로 채워진다. 임시 버퍼(58A)가 완전히 채워짐에 따라, 임시 버퍼(58A)의 콘텐츠들은 임시 버퍼(58A)를 비우기 위해 버텍스 캐시(54)의 (제 1) 로우에 기록될 수 있다.

[0057] [0066] 임시 버퍼(58A)가 비워지면, 작업 버퍼(58B)의 슬롯들 X, Y, 및 Z에 연이어 채워지는 나머지 가변 콤포넌트들 V1.y, V1.z 및 V1.w의 콘텐츠들은 임시 버퍼(58A)로 전달된다. 다시, 임시 버퍼(58A)는 채워지지 않는다. 따라서, VS 출력 레지스터 파일(57)로부터 V2.x 및 V2.y로 표시된 2개의 콤포넌트들을 갖는 가변치 V2를 판독하고, 임시 버퍼(58A)의 나머지 슬롯(들)을 채움으로써 HW 패킹이 지속된다. 이러한 경우, 임시 버퍼(58A)(상위 로우)의 슬롯 W은 가변치 V2.x로 채워진다. 나머지 가변 콤포넌트 V2.y는 제 2 로우 또는 작업 버퍼(58B)의 슬롯 X에 채워진다. 임시 버퍼(58A)가 완전히 채워질 때, 임시 버퍼(58A)의 콘텐츠들은 임시 버퍼(58A)를 비우기 위해 버텍스 캐시(54)의 (제 2) 로우에 기록될 수 있다.

[0058] [0067] 이러한 프로세스는 VS 출력 레지스터 파일(57)에서의 가변치들에 대해 지속된다. 예를 들어, 마지막 가변치는 임시 버퍼(58A)의 X, Y 및 Z 슬롯들만을 채우기 때문에, 콘텐츠가 가변 저장기 또는 버텍스 캐시(54)에 기록되며 마스크 = xyz 또는 (111)이다.

[0059] [0068] 패킹 버퍼들(58)의 임시 버퍼(58A) 및 작업 버퍼(58B)는 성능을 위한(for performance) 것들이다. 임시 버퍼(58A)가 채워지고 가변 저장기 또는 버텍스 캐시(54)로부터의 기록이 준비될 때, 다른 버퍼(작업 버퍼(58B))가 동시적으로 채워질 수 있다. 판독 버스 및 기록 버스 모두는 시간에 따라 4(M) 콤포넌트들을 위해 이용될 수 있다. 하나의 판독 또는 기록 데이터가 4 미만 콤포넌트인 경우, 판독 또는 기록 마스크는 콤포넌트들이 판독되는 또는 기록되게 유효하다는 것을 표시하기 위해 이용된다.

[0060] [0069] 제 2-레벨 HW 패킹이 완료된 후, VS 출력 심볼 테이블(표 2)에서의 패킹된 가변치들에 해당하는 표 4의 새롭게 할당된 VS 가변 출력 레지스터 ID에 해당하는 레지스터 ID는 표 6의 새롭게 할당된 VS 가변 출력 레지스터 ID 컬럼에 표시된 가변 저장기 또는 버텍스 캐시(54)에 대응되게 변한다. 간단성 및 유연성(flexibility)을 위해, 버텍스 캐시(54)의 위치와 관련한 출력 ID는 벡터 레지스터 대신 콤포넌트 유니트에 기초하여 할당된다. 본 예에 대해, ID=0인 texcoord0, 및 ID=2인 texcoord1는 가변 저장기 또는 버텍스 캐시(54)의 제 1 로우로 패킹되며, ID=4인 color0는 제 2 로우에 ID=8인 color1는 제 3 로우에 패킹되는 것으로 가정된다. 위치(position) 및 texcoord2는 FS에 이용되지 않으며, 따라서 FS 입력 레지스터 파일(79)에서 이들에 대한 저장/패킹이 할당되지 않는다. 따라서, 새롭게 할당된 FS 가변 입력 레지스터 ID는 표 4 또는 표 6에 제공되지 않는다.

[0061] [0070] 제 2-레벨 HW 패킹은 HW에 의해 수행되나 표 6에 도시된 것과 같은 링크 테이블(86)은 드라이버(61)의 링커(80)에 의해 업데이트된다. 드라이버(61)는 동일한 패킹 메커니즘 및 도 4의 VS 입력 및 출력 심볼 테이블들(64, 66) 및 FS 입력 심볼 테이블(74) 등에 기초하여 각각의 가변 콤포넌트에 대한 가변 저장기 또는 버텍스 캐시(54)의 새로운 레지스터 ID/콤포넌트 ID를 계산할 수 있다. 표 4는 제 2-레벨 HW 패킹 없이 표현되는 링킹 테이블을 예시한다. 표 6은 HW 패킹 이후 표현되는 링킹 테이블을 예시한다.

표 6

[0062] 제 2 레벨 HW 패킹 이후 VS 출력들 및 FS 입력들에 대한 링크 테이블

가변치 이름	새롭게 할당된 VS (가변) 출력 ID	새롭게 할당된 FS (가변) 입력 ID	새로운 마스크
Position			
color0	4	2	1111
color1	8	3	0111
Texcoord0	0	1	0011
Texcoord1	2	1	1100
Texcoord2			

[0063] **바이패싱 속성들**

[0064] [0071] VS(60)과 같이 프로그램가능한 버텍스 쉐이더는 PC 게임 디바이스 및 모바일 디바이스들 모두에서의 최신 GPU의 주요 계산 유니트(key computation unit)이다. VS(60)은 계산적으로 전력을 소모하며 통상적으로 성

능 병목을 겪는다. 그러나, 일부 애플리케이션들은 쉐이더 기능을 사용하지 않을 수 있다. 다른 고려사항으로는 VS(60)에 대한 일부 입력들이 계산들을 위한 임의의 조건(need) 없이 출력들로 직접 이동할 수 있다는 것이다.

[0065] [0072] 기능들(functions)에 대한 가장 간단한 솔루션(solution)은 모든 입력을 버텍스 쉐이더로 전달(pass)하는 것이며, 버텍스 쉐이더는 이동 명령들을 실행한다. 그러나, 이러한 솔루션은 상당한 계산 전력(computation power)을 소모시켜 버텍스 쉐이더 성능을 저하시킨다. 성능 저하는 1) 데이터 입력들/출력들에 대한 불필요한 트래픽 대역폭; 및 2) 버텍스 쉐이더에서 실행되는 불필요한 이동 명령들의 결과이다.

[0066] [0073] 따라서, GPU(34)는 입력 스트림 디코더(50)로부터 입력 바이패싱 경로로 구성 및 배열된다. 입력 바이패싱 경로는 버텍스 캐시(54)로 직접 진행할 수 있다. 드라이버(61) 또는 컴파일러(62)는 버텍스 캐시(54)에 직접 바이패싱될 수 있는 입력들 및 버텍스 쉐이더(60)에 로딩되어야 하는 입력들을 지정할 수 있다. 컴파일러(62)는 바이패싱된 입력들에 대한 쉐이더 프로그램으로부터 불필요한 모든 이동 명령들을 제거한다.

[0067] [0074] 도 4에 가상으로 도시된 작은 하드웨어 제어 로직(51)은 입력 스트림 디코더(50)내에 있다. 따라서, 수신된 입력이 "바이패스"로서 지정될 때, 입력 포맷 디코딩 후, 입력은 바이패싱된 경로를 따라 전송되며 버텍스 캐시(54)에 저장된다. "바이패스"로 표시되지 않은 수신된 입력들만이 VS 입력 레지스터 파일(56) 에 패킹되고 버텍스 쉐이더(60)로 전송된다.

[0068] [0075] 예시적 실시예에서, 바이패싱된 속성들은 버텍스 캐시(54)에서의 저장 이전에 패킹 버퍼(52)에 패킹된다. 컴파일러(62)는 버퍼(58A) 패킹과 관련하여 앞서 개시된 방식으로 마스크 및/또는 레지스터 ID를 변조시킨다. 캐시 인덱스는 바이패싱된 입력들과 함께 버텍스 캐시(54)를 통과한다. 버텍스 쉐이더(60)로부터의 출력들은 동일한 버텍스에 대해 동일한 ID/인덱스를 가지며, 따라서 버텍스 캐시(54)는 버텍스 쉐이더 출력들과 바이패싱된 입력들을 쉽게 동기화시킬 수 있다.

[0069] [0076] 도 8A 및 8B는 바이패싱 속성들이 제거되기 이전 및 이후의 버텍스 쉐이더 프로그램을 도시한다. 일부 버텍스 쉐이더들은 L3 및 L4로 표시된 라인들 상에서 MOV 명령들을 갖는다. MOV 명령들은 속성들과 연관된 입력 레지스터들로부터 가변치들과 연관된 출력 레지스터들로의 이동을 야기시킨다. 이러한 속성들은 버텍스 쉐이더(60)로부터 바이패스될 수 있다. 예를 들어, 도 8A에서, 파라미터들(v0, v1, v2)은 입력 속성들이며, oPos, oFog, oT0 및 oD0는 출력 가변치들이다. 본 예에서, 라인들(L1 및 L3) 상에서의 입력 속성(v1) 및 라인들(L2 및 L4) 상에서의 입력 속성(v2)은 버텍스 쉐이더(60)에서 어떠한 계산들도 수반하지 않으며 oT0 및 oD0로만 이동된다. 따라서, 속성들(v1 및 v2)은 버텍스 쉐이더(60)의 프로그램(명령들의 세트)이 실행되기 이전에 가변 저장기 또는 버텍스 캐시(54)로 직접 바이패스될 수 있다. 속성들(v1 및 v2)이 바이패싱된 후, 이들은 도 8B의 라인들(L1, L2, L3, L4)의 삭제로 표시된 버텍스 쉐이더(60)로 전송되지 않는다. 부가적으로, 출력 가변치들(oT0 및 oD0)은 도 8B의 라인들(L3 및 L4) 없이 표시된 버텍스 쉐이더(60)로부터 출력되지 않는다. 따라서, 바이패싱 기능은 트래픽 대역폭 및 버텍스 쉐이더 계산력을 절감시킨다.

[0070] [0077] 속성 바이패싱을 위해, 제 2-레벨 HW 패킹이 하기 개시된 것처럼 조절된다. 바이패싱된 속성들은 패킹 버퍼(53)에서만 제 2-레벨 HW 패킹을 처리한다. 패킹 버퍼(52)는 스트림 디코더(50)로부터 바이패싱된 속성들을 수신한다. 스트림 디코더(50)는 메인(외부) 메모리(40)로부터 버텍스 스트림 (속성) 페칭(fetching) 및 상이한 속성 포맷들로부터 IEEE 플로트 포맷으로의 포맷 변환을 야기시킨다. 드라이버(61)는 속성들이 바이패스되고 속성들이 버텍스 쉐이더(60)를 위해 VS 입력 레지스터 파일(56)로 전송되는 것을 스트림 디코더(50)와 통신하 수 있다. 바이패싱된 속성들은 상기 임시 버퍼(58A) 및 작업 버퍼(58B)를 사용하여 앞서 개시된 것과 동일한 방식으로 패킹될 수 있다. 바이패스되지 않은 속성들은 버텍스 쉐이더(60)의 VS 입력 레지스터 파일(56)로 전송되어 패킹된다.

[0071] [0078] 바이패싱된 속성들 및 VS 출력 레지스터 파일(57)로부터의 가변치들은 전체 가변 풋프린트(footprint)로서 전체 가변 저장기 또는 버텍스 캐시(54)에 채워진다. 간단성을 위해, 바이패싱된 속성들로부터의 가변치들은 가변 저장기 또는 버텍스 캐시(54)의 제 1 소수(few) 로우들에 패킹 및 저장되고 패킹 버퍼(58)에 패킹된 VS 출력들은 이후 가변 저장 또는 버텍스 캐시(54)에 저장된다. 예를 들어, 다시 도 8A를 참조로, 가변 출력(바이패싱된 속성) oD0(v2)는 패킹 버퍼(52)의 임시 버퍼(52A)에 패킹되고 가변 저장기 또는 버텍스 캐시(54)의 제 1 로우에 저장된다. 가변 출력(바이패싱된 속성) oT0(v1)은 가변 저장기 또는 버텍스 캐시(54)의 제 2 로우에 있는 2개의 낮은 자릿수 콤포넌트들(low significant components)에 패킹 및 저장된다. VS 출력들(oPos 및 oFog)은 이후 제 2 로우의 2개의 높은 자릿수 콤포넌트들(most significant components)로부터 시작하여 패킹 또는 저장된다. 이 경우, oPos.xy는 임시 버퍼의 zw 슬롯들에 패킹된 다음 가변 저장기 또는 버텍스 캐시(54)

의 제 2 로우에 기록되며, 기록 마스크=zw 이다. 따라서, 동일 로우에 그러나 상이한 콤포넌트 위치들에서 oTo 이후 연속적으로 패킹된다. oPos.zw 및 oFog는 xyz 콤포넌트 슬롯들에서 작업 버퍼(52B)에 패킹되며 가변 저장기 또는 버텍스 캐시의 제 3 로우에 기록되고 기록 마스크 = xyz이다. 따라서, 링크 테이블(86)이 업데이트된다.

[0072] [0079] 도 9A and 9B는 바이패싱 속성들이 제거되기 이전 및 이후의 또 다른 버텍스 쉐이더 프로그램을 도시한다. 화살표들로 표시된 라인들 L5, L6, L7, L8 및 L9 상의 이동 명령들은 바이패스될 수 있다. 예를 들어, 화살표로 표시된 라인 L5 상에서, 가변 출력 oTO(v1)가 바이패스될 수 있다. 다른 가변 출력들 oT1(v1), oT2(v3), oDO(v4) 및 oD1(v5) 또한 바이패스될 수 있다. 도 9B에서, 도 9A의 화살표들로 표시된 라인들은 제거된다.

[0073] [0080] 도 10A 및 10B는 바이패싱 속성들이 제거되기 이전 및 이후 추가의 버텍스 쉐이더 프로그램을 도시한다. 화살표들로 표시된 라인들(L10 및 L11) 상에서의 이동 명령들이 바이패싱될 수 있다. 도 10B에서, 도 10A의 화살표들로 표시된 라인들(L10 및 L11)은 제거된다.

[0074] [0081] 도 11A 및 11B는 바이패싱 속성들이 제거되기 이전 및 이후의 추가의 버텍스 쉐이더 프로그램을 도시한다. 화살표들로 표시된 라인들 L1 2, L13, L14, L15 및 L16 상의 이동 명령들은 바이패스될 수 있다. 도 11B에서, 도 11A의 화살표들로 표시된 라인들 L1 2, L13, L14, L15 및 L16은 제거된다.

[0075] [0082] 도 12A 및 12B는 바이패싱 속성들이 제거되기 이전 및 이후의 추가 버텍스 쉐이더 프로그램을 도시한다. 화살표들로 표시된 라인들 L17 및 L18 상의 이동 명령들은 바이패스될 수 있다. 도 12B에서, 도 12A의 화살표들로 표시된 라인들 L17 및 L18은 제거된다. 도 8A, 8B, 9A, 9B, 10A, 10B, 11A, 11B, 12A 및 12B에 도시된 예들은 예시를 목적으로 하는 것이며 계산들을 요구하지 않는 다른 이동 명령들 또는 속성들이 "바이패스 속성들"로 표시될 수 있다.

[0076] [0083] 쉽게 알 수 있듯이, 바이패싱 속성 프로세스의 장점들로는, 1) 쉐이더 코드 크기 및 실행 명령들의 감소, 2) 입력들/출력들의 트래픽 대역폭 감소, 3) 보다 많은 버텍스들이 ALU(산술 및 논리 유니트) 대기시간(latency) 및 텍스처 로딩 대기시간을 포함하게 허용하는 레지스터 파일 크기의 감소, 4) 대기시간을 커버하기 위한 보다 적은 명령들 및 보다 많은 버텍스들로 인한 성능 개선, 5) 보다 적은 명령 실행 및 보다 적은 트래픽으로 인한 전력 절감; 6) 쉐이더 바이패스/디스에이블(disable)에 대한 제너릭(generic); 7) 8) CPU/DSP(32) 및 GPU(34) 간의 로드 밸런싱을 위해 쉐이더 프로그램의 일부를 CPU 또는 DSP로 이동시킴으로써 성능을 조정하는 드라이버(61)에 대한 옵션, 및 8) 주변의(around) 예상치못한 문제들을 처리하는 드라이버(61)에 대한 옵션이 포함된다.

[0077] [0084] 실제 게임들 및 벤치마크들로부터의 대부분의 버텍스 쉐이더들(VS)은 출력들로 직접 이동하는 소정의 입력들을 갖는 것으로 결정되었다. 표 7은 본 발명에 개시되는 바이패싱된 기능에 기초하여 절감된 입력 트래픽 및 절감된 출력 트래픽의 비교 및 상이한 쉐이더 프로그램들을 예시한다. 또한, 표 7은 절감된 명령들의 비율을 제공한다.

표 7

[0078] 트래픽 대역폭 및 계산 절감

쉐이더들	절감된 입력 트래픽 (DW)	절감된 출력 트래픽 (DW)	절감된 명령들(scalar)
VSF8 (3DMark06) (도 12A 및 12B)	4/17 = 23.5%	4/27 = 14.8%	4/97 = 4.1%
VSF12 (FarCry) (도 8A 및 8B)	6/10 = 60%	6/11 = 54.5%	4/28 = 14.3%
VSF14 (FarCry) (도 9A 및 9B)	13/21 = 61.9%	15/20 = 75%	15/70 = 21.4%
VSF17 (FarCry) (도 10A 및 10B)	0%	3/7= 42.8%	3/19 = 16.8%
VSF25 (FarCry) (도 11A 및 11B)	9/13 = 69.2%	15/21 = 71.4%	15/61 = 24.6%

[0079] [0085] 도 13A-13C는 속성들의 바이패싱과 조합된 쉐이더 변수 패킹 프로세스(200)의 일반적 순서도를 도시한다. 쉐이더 변수 패킹 프로세스(200)는 도 4의 블록도와 관련하여 개시된다. 쉐이더 변수 패킹 프로세스(200)는 블록 201에서 시작되며, 여기서 입력 속성 포맷은 스트림 디코더(50)에 의해 디코딩된다. 블록 201은 블록 202로 이어지며, 여기서 스트림 디코더(50)로부터의 소성들이 "바이패싱된 속성들"인지에 대한 결정이 이루어진다. 상기 결정이 "예(YES)"인 경우, 블록 202는 블록 204로 이어지며, 여기서 유효(바이패싱된) 속성 콤포넌트들은 임시버퍼(52A)에 누적된다. 블록 204는 블록 206으로 이어지며, 여기서 임시 버퍼(52A)가 채워졌는지 여부에 대한 결정이 이루어진다. 예를 들어, M(M=4) 한도(limit)의 바이패싱된 속성 콤포넌트들이 임시 버퍼(52A)에 채워질 수 있다. 임시 버퍼(52A)는 작업 버퍼(52B)를 채움으로써 채워진다.

[0080] [0086] 그러나, 블록 206에서의 결정이 "아니오(NO)"이면, 프로세스는 블록 211로 이동한다. 블록 211은 최종 입력 속성이 도달되었는지에 대해 평가하는 결정 블록이다. 블록 211에 대한 설명이 하기에 개시된다.

[0081] [0087] 임시 버퍼(52A)가 채워질 때, 블록 206은 블록 208로 이어지며, 여기서 임시 버퍼(52A)에 저장 또는 채워진 바이패싱된 속성 콤포넌트는 버텍스 캐시(54)로 전송되고 저장된다. 앞서 개시된 것처럼, 작업 버퍼(52B)에서의 바이패싱된 속성들은 채워지거나 혹은 재분포될(repopulated) 때까지 임시 버퍼(52A)로 전달된다. 블록 208은 이후 개시되는 블록 211로 이어진다.

[0082] [0088] 다시 블록 202을 참조로, 속성들이 미리결정된 패킹 명령들에 따라 블록 202에서의 결정이 "아니오"라는 것을 의미하는 바이패스되지 않은 속성들이라면, 바이패스되지 않은 속성들은 블록 210에서 VS 입력 레지스터에 패킹된다. 블록 210은 블록 211로 이어지며, 여기서는 최종 입력 속성이 도달되었는지에 대한 결정이 이루어진다. 상기 결정이 "아니오"라면, 블록 211은 다시 블록 201로 진행되며, 여기서 보다 많은 입력 속성들이 디코딩된다. 그렇지 않은 경우, 상기 결정이 "예"인 경우, 블록 211은 블록 212로 이어지며 여기서 임시 버퍼(52A)의 바이패싱된 나머지 속성들은 버텍스 캐시(54)로 전송된다.

[0083] [0089] 블록 212은 블록 213으로 이어지며, 여기서는 바이패스되지 않은 임의의 속성들이 이용가능한지에 대한 결정이 이루어진다. 상기 결정이 "아니오"라면, 프로세스(200)는 종료된다. 그러나, 블록 213에서의 상기 결정이 "예"인 경우, 블록 213은 도 13B의 블록 214로 이어진다. 블록 214에서, 바이패스되지 않은 속성들은 VS(60)로 전송된다. 바이패스되지 않은 속성 콤포넌트들이 VS(60)로 전송된 후, VS(60)은 블록 216에서 버텍스 쉐이딩 오퍼레이션들을 수행한다. VS(60)가 완료(done)된 후, 유효 출력 가변 콤포넌트들은 블록 218에서 제1-레벨 컴파일러 패킹을 완료하는 쉐이더 명령들의 실행 동안 VS 출력 레지스터 파일(57)에 자동적으로 패킹된다. 블록 218에서의 패킹은 도 7의 블록 102와 대응된다.

[0084] [0090] 블록 218은 도 13C의 블록 222로 이어진다. 표 5와 관련하여 앞서 개시된 것처럼, VS 출력 레지스터 파일(57)로부터의 출력 가변치들은 임시 버퍼(58A)에 누적된다. 임시 버퍼(58A)는 작업 버퍼(58B)와의 조합으로 채워진다. 블록 222는 임시 버퍼(58A)가 채워졌는지를 결정하기 위해 블록 224로 이어진다. "아니오"인 경우, 프로세스는 블록 222로 리턴된다. 상기 결정이 "예"인 경우, 블록 224는 블록 226로 이어지며, 임시 버퍼(58A)의 콘텐츠들은 버텍스 캐시(54)로 전송된다. 블록 226는 블록 228로 이어지며, 여기서는 VS 출력 레지스터 파일(57)에서의 파일이 마지막인지에 대한 결정이 이루어진다. 상기 결정이 "아니오"인 경우, 프로세스는 블록 222로 복귀된다. 상기 결정이 "예"인 경우, 블록 228은 블록 230으로 이어지며, 여기서는 임시 버퍼(58A)의 나머지 가변 콤포넌트들이 버텍스 캐시(54)로 전송된다.

[0085] [0091] 패킹 이후, 트래픽 대역폭이 감소된다. 저장기가 상당히(highly) 이용되며 또한 성능이 개선된다.

[0086] [0092] 대안적 패킹 메커니즘들이 사용될 수 있다. 예를 들어, VS 출력 레지스터 파일(57)에서의 가변치들은 제 2-레벨 HW 패킹을 사용하여 패킹되지 않는다. 대신, 파일(57)은 있는 그대로 버텍스 캐시(54)에 카피된다. 표 5와 관련하여, 표 5의 왼쪽(left hand side) 버텍스 캐시(54)에 카피된다. 이는 동일한 레이아웃 및 형상을 유지한다. 원시 어셈블러 및 래스터라이저(90)에 이어 동일한 패킹 메커니즘이 수행되며 래스터라이저 결과치들은 가변 버퍼(92)로 전송된다. 원시 어셈블러 및 래스터라이저(90)는 계산 절감을 위해 표 2에서의 마스크들에 기초하여 무효(invalid)(마스크=0) 콤포넌트들에 대한 계산들을 스킵(skip)한다.

[0087] **링커 & 링킹**

[0088] [0093] VS(60)로부터의 가변치들은 FS(70)에 입력된다. 따라서, VS(60)에대한 가변 심볼은 컴파일러(62)에 의해 생성되며 VS 출력 심볼 테이블(66)에 저장된다. FS(70)의 각각의 입력은 가변 심볼 또는 가변치 이름에 의

해 정의되는 것처럼 VS 출력 심볼 테이블(66)에서의 각각의 가변 심볼 출력으로 제한된다. 따라서, VS 출력 심볼 테이블(표 1)에서의 가변치 이름으로 표시된 가변 심볼이 FS 입력 심볼 테이블(74)(표 2)에서 FS 가변치 이름으로 표시된 하나의 엔트리와 매칭되는 경우, VS(60)로부터의 출력(가변치)은 FS(70)의 입력으로 제한된다. 링커(80)는 VS(60)에서의 순서 및 패킹 때문에 FS 입력이 통상적으로 FS(70)과 상이하다는 것을 결정한다. 링커(80)는 도 5의 가변 리맵핑 및 로딩 모듈(84)에 대해 링킹 명령들(82) 또는 링킹 테이블(86)을 생성하는 드라이버(61)의 일부이다.

[0089] [0094] 유사한 링킹 솔루션이 VS 입력 심볼 테이블(64)로 표시된 VS 입력과 버텍스 스트림 디코더(50) 간의 링킹을 위해 적용된다. 부가적으로, 링킹 솔루션은 샘플 오퍼레이션 유니트 당 입력 및 FS 출력 심볼 테이블(76)로 표시된 FS 출력의 링킹을 위해 적용된다. 링커(80)는 임의의 2개의 인접 프로그램가능 프로세싱 스테이지들에 대해 이용될 수 있다.

[0090] [0095] 도 14는 링킹 프로세스(300)의 일반적 흐름도를 도시한다. 링킹 프로세스(300)는 블록 302에서 시작되며, 여기서 링커(80)는 VS 출력 심볼 테이블(66) 및 FS 입력 심볼 테이블(76) 모두로부터 동일한 심볼을 검색하고 비교한다. 블록 304에서, 매칭 심볼과 연관된 가변치들은 가변 버퍼(92)로부터 판독된다. 블록 206은 FS 입력 레지스터 파일(79)로 가변치들을 전송한다. 따라서, 링킹이 완료된다. 예시적 결과 링킹 테이블은 표 6에 도시된다. 링킹 프로세스(300)는 FS(70)에 대해 요구되는 각각의 가변치들에 대해 반복된다.

[0091] [0096] 컴파일러(62)의 패킹으로 인해, VS 출력 심볼 테이블(66)은 FS 입력 심볼 테이블(76)과 상이하다. 따라서, 바람직한 링킹 명령은 가변 콤포넌트 당 원칙(per varying component basis)으로 규정된다.

[0092] [0097] 볼 수 있듯이, 2-레벨 패킹은 엄격한 가변 패킹을 쉽게 가능케한다. 제 1-레벨 컴파일러 패킹은 입력들 및 출력들을 적게하여 레지스터 파일 풋 프린트 크기를 감소시킨다. HW 패킹은 간단하고 효율적이다. 또한 패킹 프로세스는 입력들/출력들의 트래픽 대역폭을 감소시킨다. 패킹 프로세스(100 또는 200)는 적은 트래픽으로 인해 전력을 절감하고 캐시 저장기를 잘(highly) 이용한다.

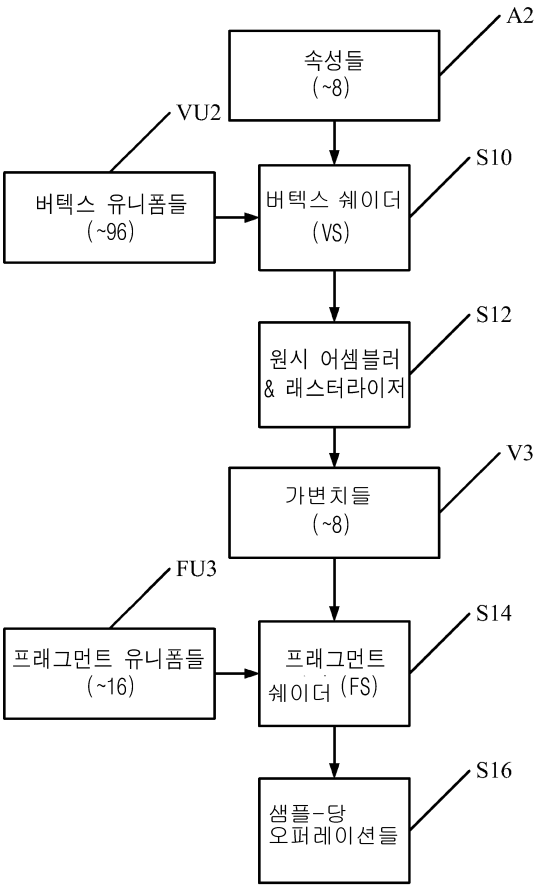
[0093] [0098] 도 4 및 도 5에 도시된 GPU 파이프라인은 임의의 2개의 인접하는 프로그램가능 프로세싱 스테이지들에 대한 일반적 링킹을 이용한다. 이는 벡터들 및 플로트들에 대한 어레이 및 매트릭스와 같이, 크고 복잡한 가변 구조들에 대한 계층적 링킹, 맵핑을 허용한다. 패킹 프로세스는 컴파일러(62)가 최적화를 위해 레지스터들을 자유롭게 재정렬(re-order) 또는 재할당(re-allocate)하는 것을 허용한다. 프로세스는 FS에 사용되지 않을 경우, 링킹 명령들 중 일부를 변조함으로써 드라이버/링커가 일부 VS 출력들을 쉽게 제거하게 허용한다.

[0094] [0099] 하나 이상의 예시적 구성들에서, 개시된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합물로 구현될 수 있다. 소프트웨어에서 구현도리 경우, 기능들은 컴퓨터-판독가능 매체 상의 하나 이상의 명령들 또는 코드로서 저장 또는 전송될 수 있다. 컴퓨터-판독가능 매체는 하나의 위치에서 다른 위치로 컴퓨터 프로그램의 전달을 원활히 하는 임의의 매체를 포함하는 통신 매체 및 컴퓨터 저장 매체 모두를 포함한다. 저장 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수 있다. 제한되는 것은 아니지만, 예로써 이러한 컴퓨터-판독가능 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학적 디스크 저장기, 자기 디스크 저장기 또는 다른 자기 저장기 디바이스들, 또는 컴퓨터에 의해 액세스될 수 있고 명령들 또는 데이터 구조들의 형태로 원하는 프로그램 코드를 전달(carry) 또는 저장하는데 이용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 결합(connection)이 컴퓨터-판독가능 매체로 불린다. 예를 들어, 소프트웨어가 웹사이트, 서버 또는 동축 케이블, 광섬유 케이블, 트위스터 페어(twisted pair), 디지털 가입자 라인(DSL), 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술들을 사용하는 다른 원격 소스로부터 전송되는 경우, 동축 케이블, 광섬유 케이블, 트위스터 페어, DSL 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술이 매체 정의에 포함된다. 본 발명에서 사용되는 디스크(disk) 및 디스크(disc)는 콤팩 디스크(disc)(CD), 레이저 디스크(disc), 광학 디스크(disc), DVD(digital versatile disc), 플로피 디스크(disk) 및 블루-레이 디스크(disc)를 포함하며, 디스크(disk)들은 통상 자기적으로 데이터를 재생하는 반면, 디스크(disc)들은 레이저를 이용하여 광학적으로 데이터를 재생한다. 이들의 조합 또한 컴퓨터-판독가능 매체의 범주에 포함되어야 한다.

[0095] [00100] 개시된 구성들의 상기 설명은 임의의 당업자들이 본 발명을 구성 또는 이용할 수 있게 하기 위해 제공된다. 이러한 구성들에 대한 다양한 변형을 당업자들은 쉽게 인식할 것이며, 본 발명에 개시되는 일반적 원리들은 본 발명의 범주 또는 사상을 이탈하지 않고 다른 구성들에 적용될 수 있다. 따라서, 본 발명은 본 발명에 도시되는 구성들로 제한되고자 의도된 것이 아니라 본 발명에 개시되는 원리들 및 신규한 특징들과 일치하는 광범위한 범주를 따르도록 의도된다.

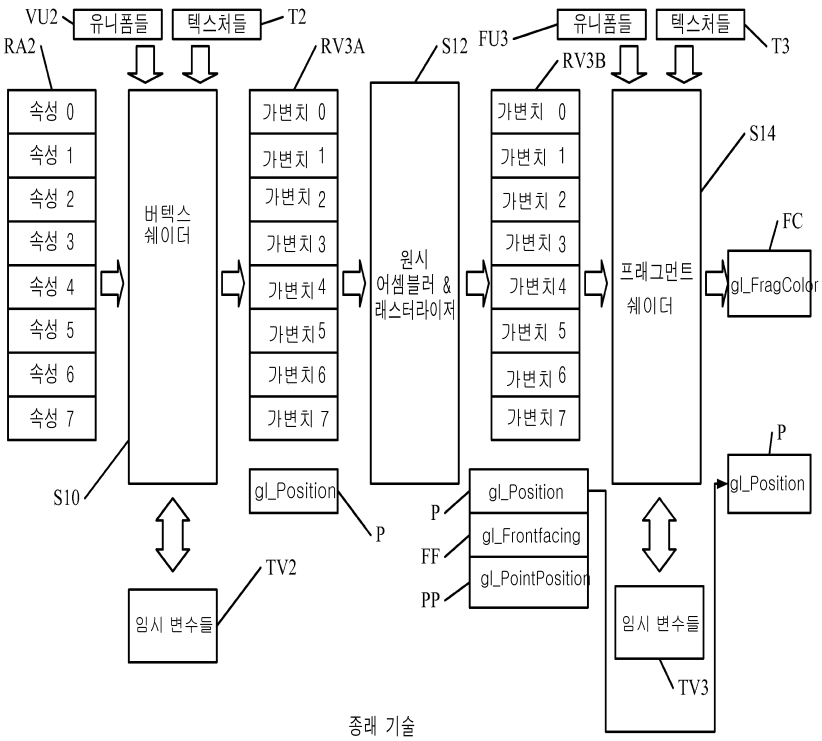
도면

도면1


A2
속성들
(~8)
VU2
버텍스 유니폼들
(~96)
S10
버텍스 쉐이더
(VS)
S12
원시 어셈블러
& 래스터라이저
V3
가변치들
(~8)
FU3
프래그먼트 유니폼들
(~16)
S14
프래그먼트
쉐이더 (FS)
S16
샘플-당
오퍼레이션들

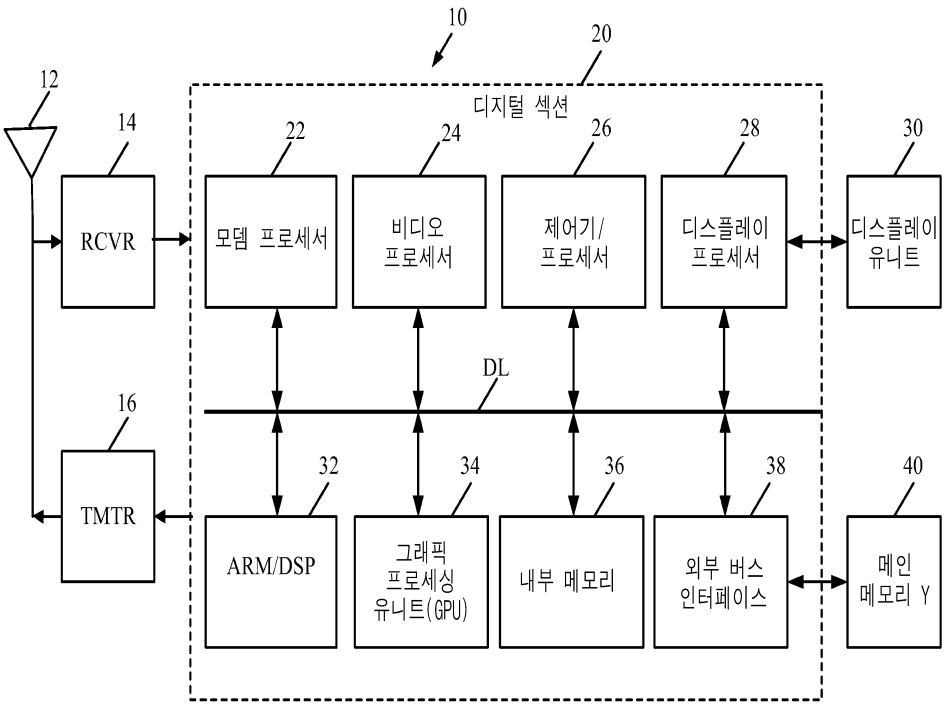
종래 기술

도면2

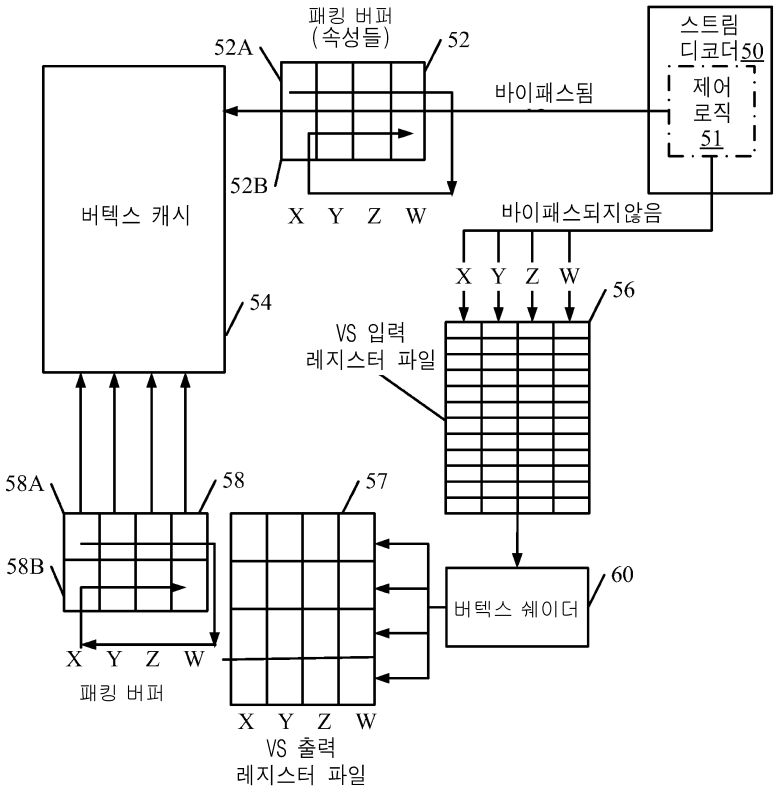

VU2
유니폼들
텍스처들
T2
RA2
RV3A
S12
FU3
유니폼들
텍스처들
T3
RV3B
속성 0
속성 1
속성 2
속성 3
속성 4
속성 5
속성 6
속성 7
버텍스 쉐이더
가변치 0
가변치 1
가변치 2
가변치 3
가변치4
가변치5
가변치6
가변치 7
원시 어셈블러 & 래스터라이저
가변치 0
가변치 1
가변치 2
가변치 3
가변치4
가변치 5
가변치 6
가변치 7
S14
프래그먼트 쉐이더
FC
gl_FragColor
P
gl_Position
S10
gl_Position
P
gl_Position
gl_Frontfacing
gl_PointPosition
P
FF
PP
TV2
임시 변수들
임시 변수들
TV3

종래 기술

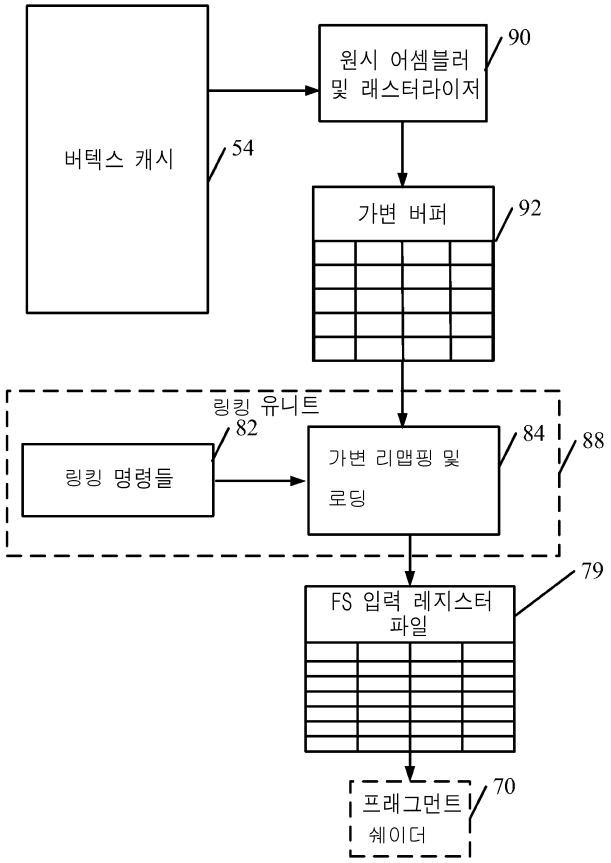
도면3


10
20
12
14
디지털 섹션
22
24
26
28
30
RCVR
모뎀 프로세서
비디오 프로세서
제어기/ 프로세서
디스플레이 프로세서
디스플레이 유니트
DL
16
32
34
36
38
40
TMTR
ARM/DSP
그래픽 프로세싱 유니트(GPU)
내부 메모리
외부 버스 인터페이스
메인 메모리 Y

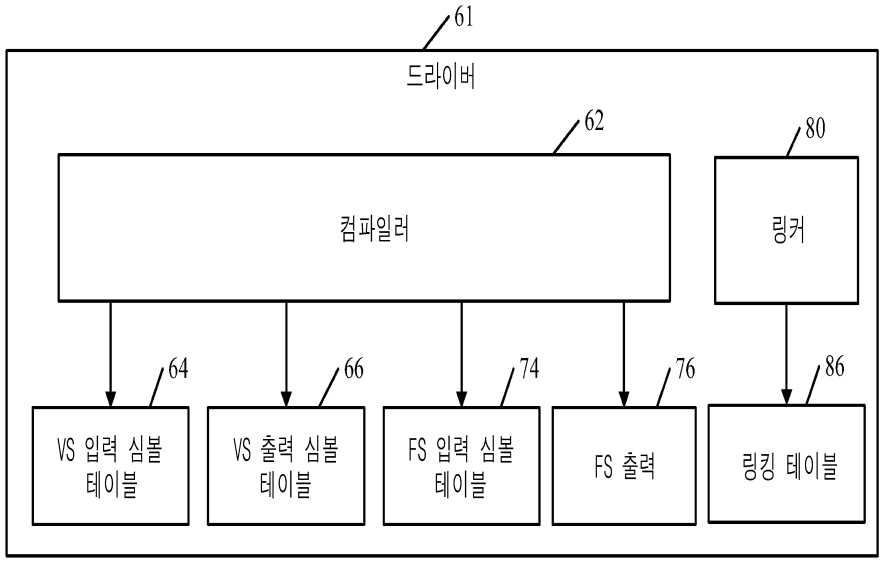
도면4


패킹 버퍼
(속성들)
52A
52
스트림
디코더50
제어
로직
51
바이패스됨
52B
X Y Z W
버텍스 캐시
바이패스되지않음
X Y Z W
56
54
VS 입력
레지스터 파일
58A
58
57
58B
60
버텍스 쉐이더
X Y Z W
패킹 버퍼
X Y Z W
VS 출력
레지스터 파일

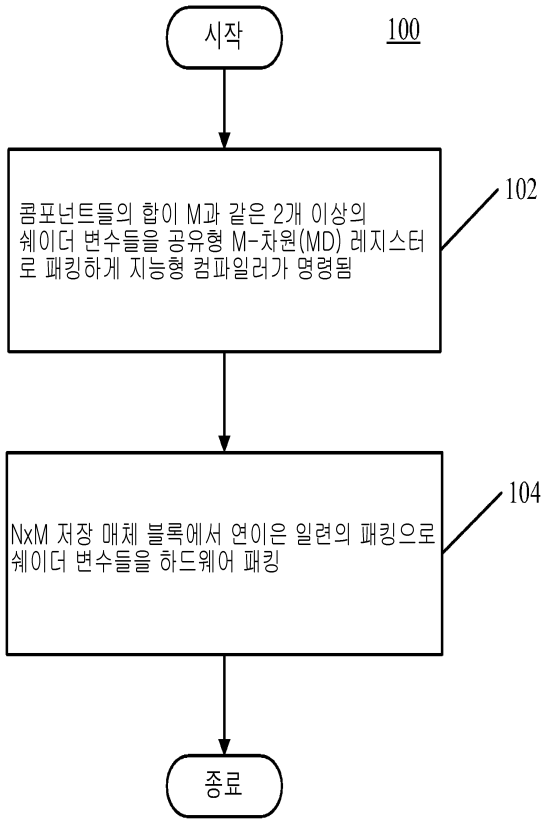
도면5


버텍스 캐시
54
원시 어셈블러 및 래스터라이저
90
가변 버퍼
92
링킹 유니트
82
링킹 명령들
가변 리맵핑 및 로딩
84
88
FS 입력 레지스터 파일
79
프래그먼트 쉐이더
70

도면6


61
드라이버
62
컴파일러
80
링커
64
VS 입력 심볼 테이블
66
VS 출력 심볼 테이블
74
FS 입력 심볼 테이블
76
FS 출력
86
링킹 테이블

도면7



도면8a

```
VERTEXSHADER_FUNCTI
ON_CREATE
"VSF12"
{
vs.1.1
dcl_position0 v0
dcl_texcoord0 v1   ◄L1—
dcl_color0 v2      ◄L2—
dp4 oPos.x, c0, v0
dp4 oPos.y, c1, v0
dp4 r0.z, c2, v0
mad r1.w, c29.x, -r0.z, c29.y
dp4 r0.w, c3, v0
max r1.w, r1.w,c28.x
add r0.w, r0.w, c4.x
min oFog, r1.w, c28.w
mov oPos.zw, r0
mov oT0.xy, v1     ◄L3—
mov oD0, v2        ◄L4—
}
```

도면8b

```
VERTEXSHADER_FUNCTI
ON_CREATE
“VSF12”
{
vs.1.1
dcl_position0 v0


dp4 oPos.x, c0, v0
dp4 oPos.y, c1, v0
dp4 r0.z, c2, v0
mad r1.w, c29.x, -r0.z, c29.y
dp4 r0.w, c3, v0
max r1.w, r1.w,c28.x
add r0.w, r0.w, c4.x
min oFog, r1.w, c28.w
mov oPos.zw, r0


}
```

도면9a

```
VERTEXSHADER_FUNCTION
_CREATE
“VSF14”
{
vs.1.1
def c4, 1,0, 0.0, 0.0, 0.0
dcl_position0 v0
dcl_texcoord0 v1
dcl_texcoord1 v2
dcl_texcoord2 v3
dcl_color0 v4
dcl_color1 v5
mad r0. v2.xyzx, c4.xxxy,
c4,yyyx
mul r1.xyz, v2.w, c0
dp4 r1.w, r0, c0
dp4 oPos.x, r1, v0
dp4 r2.w, r0, c2
mul r2.xyz, v2.w, c2
mul r1.xyz, v2.w, c1
dp4 r2.z, r2, v0
dp4  r1.w, r0, c1
mad r2.w, c29.x, -r2.z, c29.y
dp4 oPos.y, r1, v0
max r1.w, r2.w, c28.x
dp4 r0.w, r0, c3
min r1.x, r1.w, c28.w
mov oFog, r1.x
mul r0.xyz, v2.w, c3
mov oD1, r1.x
dp4 r2.w, r0, v0
mov oPos.zw, r2
mov oT0.xy, v1      ◄L5—
mov oT1.xy, v1      ◄L6—
mov oT2, v3         ◄L7—
mov oD0, v4         ◄L8—
mov oD1.xyz, v5     ◄L9—
}
```

도면9b

```
VERTEXSHADER_FUNCTION
_CREATE
“VSF14”
{
vs.1.1
def c4, 1,0, 0.0, 0.0, 0.0
dcl_position0 v0
dcl_texcoord0 v1
dcl_texcoord1 v2
dcl_texcoord2 v3
dcl_color0 v4
dcl_color1 v5
mad r0. v2.xyzx, c4.xxxy,
c4,yyyx
mul r1.xyz, v2.w, c0
dp4 r1.w, r0, c0
dp4 oPos.x, r1, v0
dp4 r2.w, r0, c2
mul r2.xyz, v2.w, c2
mul r1.xyz, v2.w, c1
dp4 r2.z, r2, v0
dp4  r1.w, r0, c1
mad r2.w, c29.x, -r2.z, c29.y
dp4 oPos.y, r1, v0
max r1.w, r2.w, c28.x
dp4 r0.w, r0, c3
min r1.x, r1.w, c28.w
mov oFog, r1.x
mul r0.xyz, v2.w, c3
mov oD1, r1.x
dp4 r2.w, r0, v0
mov oPos.zw, r2




}
```

도면10a

```
VERTEXSHADER_FUNCTION
_CREATE
“VSF17”
{
vs.1.1
dcl_position0 v0
dp4 oPos.x, c0, v0
dp4 oPos.y, c1, v0
dp4 oPos.z, c2, v0
dp4 oPos.w, c3, v0
mov oFog, v0.w      ◄L10─
mov oT0.xy.v0.w     ◄L11─


}
```

도면10b

```
VERTEXSHADER_FUNCTION
_CREATE
“VSF17”
{
vs.1.1
dcl_position0 v0
dp4 oPos.x, c0, v0
dp4 oPos.y, c1, v0
dp4 oPos.z, c2, v0
dp4 oPos.w, c3, v0




}
```

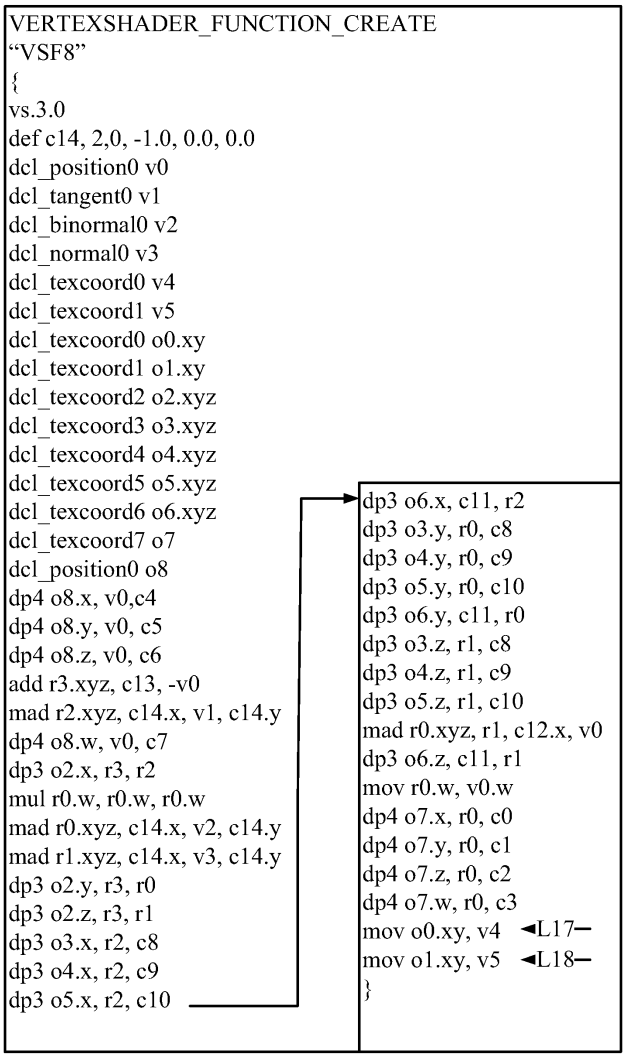
도면11a

```
VERTEXSHADER_FUNCTION
_CREATE
“VSF25”
{
vs.1.1
def c4, 0,0, 0.0, 0.0, 0.0
dcl_position0 v0
dcl_texcoord0 v1
dcl_color0 v2
dcl_color1 v3
max r0.w, v0.z, c4.w
mad r0.w, r0.w, c7.z, c7.w
mul r0.w, r0.w, r0.w
mad r0.w, r0.w, r0.w, -c7.w
mad r0.xy, c7, r0.w, v0
mov r0.z, v0.z
dp3 r2.x, r0, r0
dp3 r1.x, v0, v0
rsq r1.w. r2.x
rsq r0.w. r1.x
mul r0.xyz, r0, r1.w
rcp r0.w, r0.w
mul r0.xyz, r0, r0.w
mov r0.w, v0.w
dp4 oPos.x, c0, r0
mad r1.w, c29.x, -r1.z, c29.y
dp4 oPos.y, c1, r0
max r2.w, r1.w, c28.x
dp4 r1.w, c3, r0
min r0.x, r2.w, c28.w
mov oPos.zw, r1
mov oFog, r0.x
mov oD1.w, r0.x
mov oT0.xy, v1        ◄L12—
mov oT1.xy, v1        ◄L13—
mov oT2, c4.x         ◄L14—
mov oD0, v2           ◄L15—
mov oD1.xyz, v3       ◄L16—
}
```

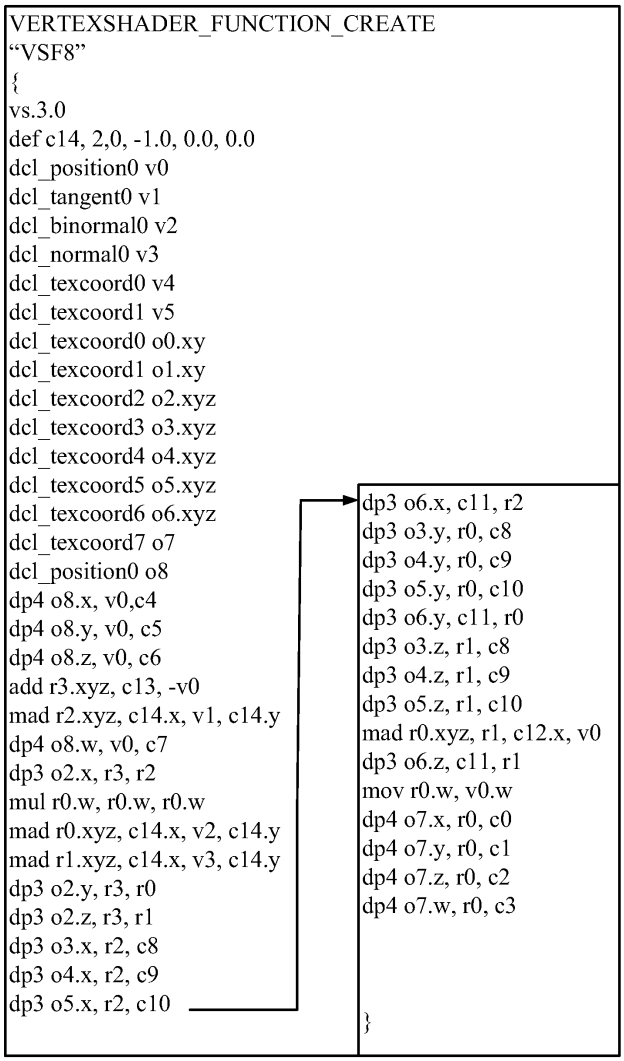
도면11b

```
VERTEXSHADER_FUNCTION
_CREATE
“VSF25”
{
vs.1.1
def c4, 0,0, 0.0, 0.0, 0.0
dcl_position0 v0
dcl_texcoord0 v1
dcl_color0 v2
dcl_color1 v3
max r0.w, v0.z, c4.w
mad r0.w, r0.w, c7.z, c7.w
mul r0.w, r0.w, r0.w
mad r0.w, r0.w, r0.w, -c7.w
mad r0.xy, c7, r0.w, v0
mov r0.z, v0.z
dp3 r2.x, r0, r0
dp3 r1.x, v0, v0
rsq r1.w. r2.x
rsq r0.w. r1.x
mul r0.xyz, r0, r1.w
rcp r0.w, r0.w
mul r0.xyz, r0, r0.w
mov r0.w, v0.w
dp4 oPos.x, c0, r0
mad r1.w, c29.x, -r1.z, c29.y
dp4 oPos.y, c1, r0
max r2.w, r1.w, c28.x
dp4 r1.w, c3, r0
min r0.x, r2.w, c28.w
mov oPos.zw, r1
mov oFog, r0.x
mov oD1.w, r0.x




}
```

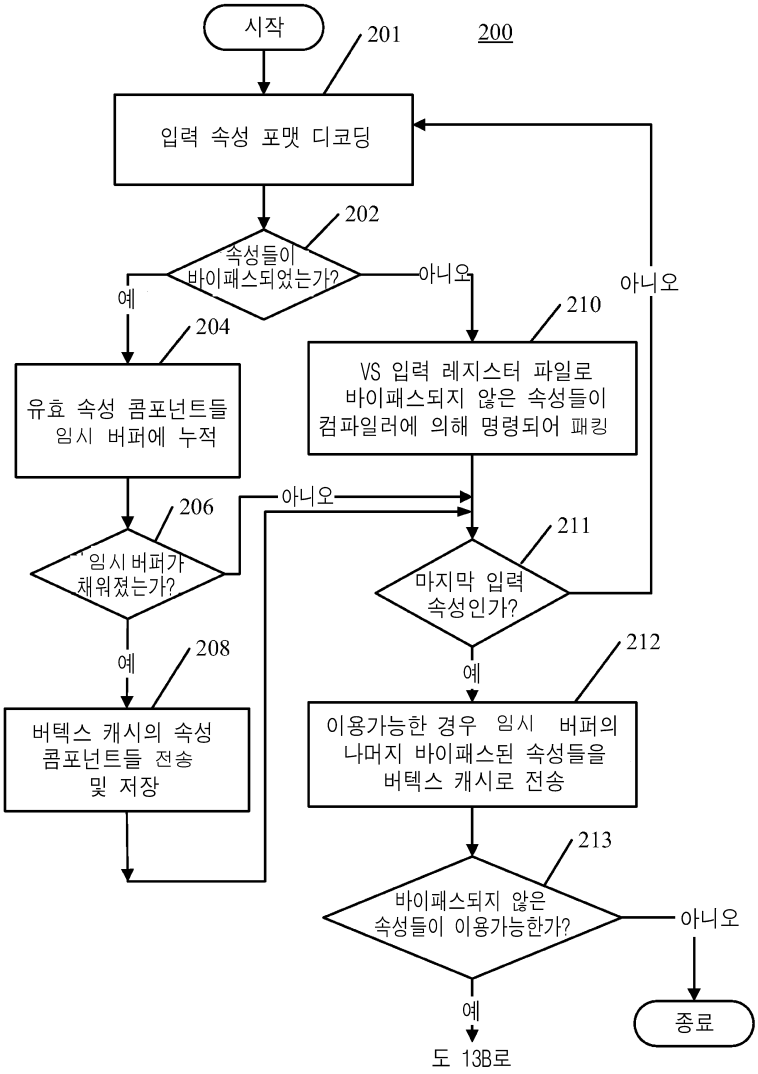
도면12a


VERTEXSHADER_FUNCTION_CREATE
“VSF8”
{
vs.3.0
def c14, 2,0, -1.0, 0.0, 0.0
dcl_position0 v0
dcl_tangent0 v1
dcl_binormal0 v2
dcl_normal0 v3
dcl_texcoord0 v4
dcl_texcoord1 v5
dcl_texcoord0 o0.xy
dcl_texcoord1 o1.xy
dcl_texcoord2 o2.xyz
dcl_texcoord3 o3.xyz
dcl_texcoord4 o4.xyz
dcl_texcoord5 o5.xyz
dcl_texcoord6 o6.xyz
dcl_texcoord7 o7
dcl_position0 o8
dp4 o8.x, v0,c4
dp4 o8.y, v0, c5
dp4 o8.z, v0, c6
add r3.xyz, c13, -v0
mad r2.xyz, c14.x, v1, c14.y
dp4 o8.w, v0, c7
dp3 o2.x, r3, r2
mul r0.w, r0.w, r0.w
mad r0.xyz, c14.x, v2, c14.y
mad r1.xyz, c14.x, v3, c14.y
dp3 o2.y, r3, r0
dp3 o2.z, r3, r1
dp3 o3.x, r2, c8
dp3 o4.x, r2, c9
dp3 o5.x, r2, c10
dp3 o6.x, c11, r2
dp3 o3.y, r0, c8
dp3 o4.y, r0, c9
dp3 o5.y, r0, c10
dp3 o6.y, c11, r0
dp3 o3.z, r1, c8
dp3 o4.z, r1, c9
dp3 o5.z, r1, c10
mad r0.xyz, r1, c12.x, v0
dp3 o6.z, c11, r1
mov r0.w, v0.w
dp4 o7.x, r0, c0
dp4 o7.y, r0, c1
dp4 o7.z, r0, c2
dp4 o7.w, r0, c3
mov o0.xy, v4 ◄L17—
mov o1.xy, v5 ◄L18—
}

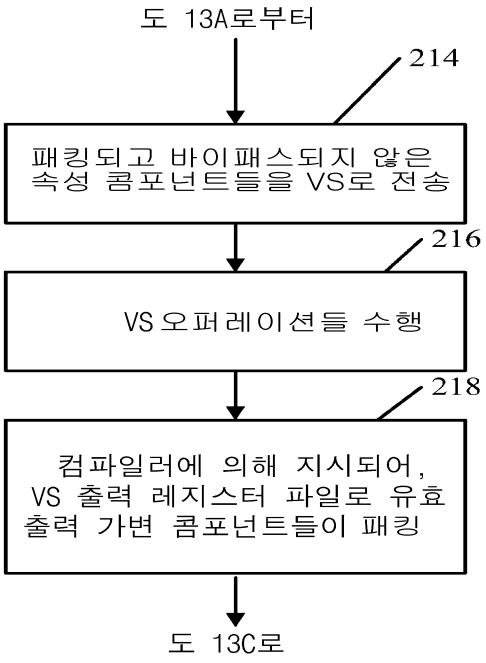
도면12b


VERTEXSHADER_FUNCTION_CREATE
“VSF8”
{
vs.3.0
def c14, 2,0, -1.0, 0.0, 0.0
dcl_position0 v0
dcl_tangent0 v1
dcl_binormal0 v2
dcl_normal0 v3
dcl_texcoord0 v4
dcl_texcoord1 v5
dcl_texcoord0 o0.xy
dcl_texcoord1 o1.xy
dcl_texcoord2 o2.xyz
dcl_texcoord3 o3.xyz
dcl_texcoord4 o4.xyz
dcl_texcoord5 o5.xyz
dcl_texcoord6 o6.xyz
dcl_texcoord7 o7
dcl_position0 o8
dp4 o8.x, v0,c4
dp4 o8.y, v0, c5
dp4 o8.z, v0, c6
add r3.xyz, c13, -v0
mad r2.xyz, c14.x, v1, c14.y
dp4 o8.w, v0, c7
dp3 o2.x, r3, r2
mul r0.w, r0.w, r0.w
mad r0.xyz, c14.x, v2, c14.y
mad r1.xyz, c14.x, v3, c14.y
dp3 o2.y, r3, r0
dp3 o2.z, r3, r1
dp3 o3.x, r2, c8
dp3 o4.x, r2, c9
dp3 o5.x, r2, c10
dp3 o6.x, c11, r2
dp3 o3.y, r0, c8
dp3 o4.y, r0, c9
dp3 o5.y, r0, c10
dp3 o6.y, c11, r0
dp3 o3.z, r1, c8
dp3 o4.z, r1, c9
dp3 o5.z, r1, c10
mad r0.xyz, r1, c12.x, v0
dp3 o6.z, c11, r1
mov r0.w, v0.w
dp4 o7.x, r0, c0
dp4 o7.y, r0, c1
dp4 o7.z, r0, c2
dp4 o7.w, r0, c3
}

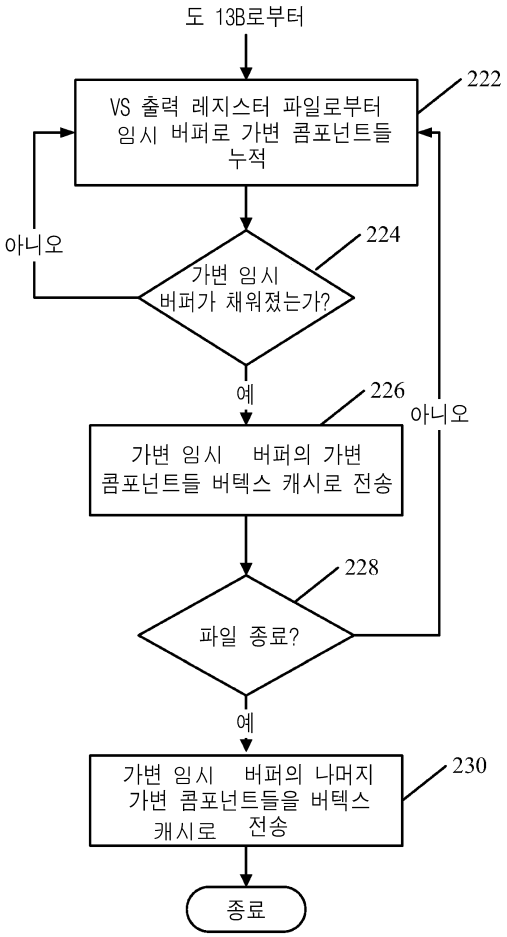
도면13a


시작
201
200
입력 속성 포맷 디코딩
202
속성들이 바이패스되었는가?
아니오
예
아니오
210
204
VS 입력 레지스터 파일로 바이패스되지 않은 속성들이 컴파일러에 의해 명령되어 패킹
유효 속성 콤포넌트들 임시 버퍼에 누적
아니오
206
211
임시 버퍼가 채워졌는가?
마지막 입력 속성인가?
예
208
212
예
버텍스 캐시의 속성 콤포넌트들 전송 및 저장
이용가능한 경우 임시 버퍼의 나머지 바이패스된 속성들을 버텍스 캐시로 전송
213
바이패스되지 않은 속성들이 이용가능한가?
아니오
종료
예
도 13B로

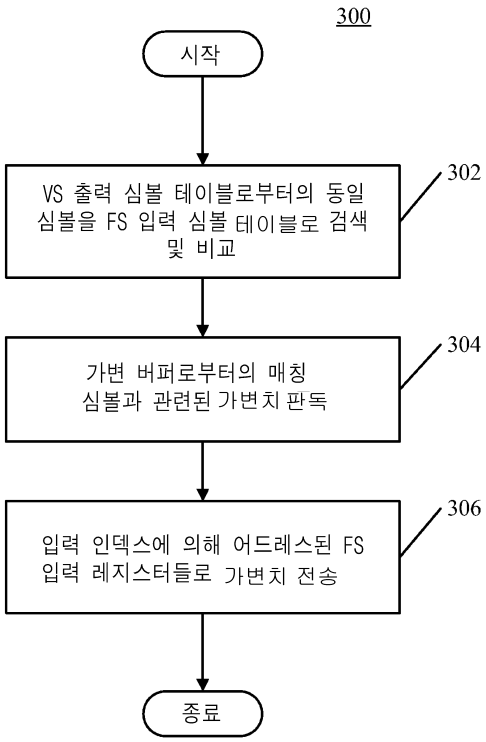
도면13b


도 13A로부터
214
패킹되고 바이패스되지 않은 속성 콤포넌트들을 VS로 전송
216
VS 오퍼레이션들 수행
218
컴파일러에 의해 지시되어, VS 출력 레지스터 파일로 유효 출력 가변 콤포넌트들이 패킹
도 13C로

도면13c


도 13B로부터
222
VS 출력 레지스터 파일로부터 임시 버퍼로 가변 콤포넌트들 누적
224
가변 임시 버퍼가 채워졌는가?
아니오
예
226
가변 임시 버퍼의 가변 콤포넌트들 버텍스 캐시로 전송
228
파일 종료?
아니오
예
230
가변 임시 버퍼의 나머지 가변 콤포넌트들을 버텍스 캐시로 전송
종료

도면14


300
시작
VS 출력 심볼 테이블로부터의 동일 심볼을 FS 입력 심볼 테이블로 검색 및 비교
302
가변 버퍼로부터의 매칭 심볼과 관련된 가변치 판독
304
입력 인덱스에 의해 어드레스된 FS 입력 레지스터들로 가변치 전송
306
종료