



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년07월28일
(11) 등록번호 10-0848687
(24) 등록일자 2008년07월21일

(51) Int. Cl.

G06T 15/00 (2006.01) G06T 15/30 (2006.01)

G06T 15/20 (2006.01) G06T 17/00 (2006.01)

(21) 출원번호 10-2007-0001518

(22) 출원일자 2007년01월05일

심사청구일자 2007년01월17일

(65) 공개번호 10-2008-0064523

(43) 공개일자 2008년07월09일

(56) 선행기술조사문헌

JP10293853 A

(뒷면에 계속)

전체 청구항 수 : 총 13 항

(73) 특허권자

삼성전자주식회사

경기도 수원시 영통구 매탄동 416

(72) 발명자

배재완

경기 수원시 영통구 영통동 1032-16 104호

최윤석

경기 수원시 영통구 망포동 늘푸른벽산아파트 109동 1802호

(74) 대리인

권혁수, 송윤호, 오세준

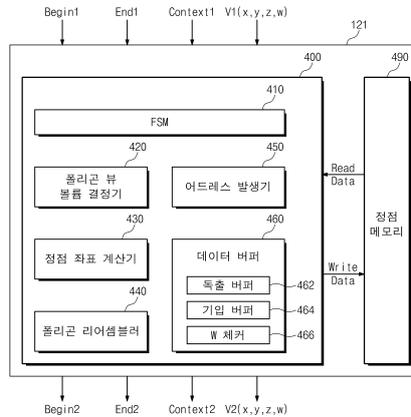
심사관 : 조우연

(54) 3차원 그래픽 처리 장치 및 그것의 동작 방법

(57) 요약

본 발명의 그래픽 처리 방법은: 제1 폴리곤을 뷰 볼륨의 전면으로 클리핑하여 제2 폴리곤을 생성하는 단계와, 상기 제2 폴리곤을 상기 뷰 볼륨의 후면으로 클리핑하여 제3 폴리곤을 생성하는 단계와, 상기 제3 폴리곤의 동치 좌표(w)가 0인지를 판별하는 단계, 그리고 상기 제3 폴리곤의 상기 동치 좌표(w)가 0일 때 상기 제3 폴리곤을 상기 뷰 볼륨의 좌면, 우면, 상면 및 하면 중 어느 하나로 클리핑하는 단계를 포함한다.

대표도 - 도4



(56) 선행기술조사문헌
JP2002015337 A
KR1019980086392 A
KR1020040036832 A
KR1020050052672 A
KR1020050052673 A

특허청구의 범위

청구항 1

제1 폴리곤을 뷰 볼륨의 전면으로 클리핑하여 제2 폴리곤을 생성하는 단계와;

상기 제2 폴리곤을 상기 뷰 볼륨의 후면으로 클리핑하여 제3 폴리곤을 생성하는 단계와;

상기 제3 폴리곤의 동치 좌표(w)가 0인지를 판별하는 단계; 그리고

상기 제3 폴리곤의 상기 동치 좌표(w)가 0일 때 상기 제3 폴리곤을 상기 뷰 볼륨의 좌면, 우면, 상면 및 하면 중 어느 하나로 클리핑하는 단계를 포함하는 것을 특징으로 하는 그래픽 처리 방법.

청구항 2

제 1 항에 있어서,

상기 제3 폴리곤의 상기 동치 좌표(w)가 0이 아니면 상기 제3 폴리곤에 대한 동치 좌표를 일반 좌표로 변환하는 단계를 더 포함하는 것을 특징으로 하는 그래픽 처리 방법.

청구항 3

제 1 항에 있어서,

상기 제1 폴리곤의 정점들이 상기 뷰 볼륨의 전면의 안쪽에 위치할 때 상기 제1 폴리곤에 대한 동치 좌표를 일반 좌표로 변환하는 단계를 더 포함하는 것을 특징으로 하는 그래픽 처리 방법.

청구항 4

제 1 항에 있어서,

상기 제2 폴리곤의 정점들이 상기 뷰 볼륨의 후면의 안쪽에 위치할 때 상기 제2 폴리곤에 대한 동치 좌표를 일반 좌표로 변환하는 단계를 더 포함하는 것을 특징으로 하는 그래픽 처리 방법.

청구항 5

제 1 항에 있어서,

상기 제1 폴리곤의 정점들 각각의 동치 좌표(w)가 음수 또는 양수인지 판별하는 단계를 더 포함하며;

상기 제1 폴리곤의 정점들 각각의 동치 좌표(w)가 모두 양수가 아니고 그리고 모두 음수가 아닐 때 상기 제1 폴리곤을 상기 뷰 볼륨의 상기 전면으로 클리핑하는 단계를 수행하는 것을 특징으로 하는 그래픽 처리 방법.

청구항 6

제어 신호들을 발생하는 제어 회로와;

상기 제어 신호들에 응답해서 폴리곤의 정점들이 뷰 볼륨 내에 위치하는 지를 판별하는 폴리곤 뷰 볼륨 결정기; 그리고

상기 제어 신호들에 응답해서 폴리곤을 상기 뷰 볼륨의 복수의 면들 중 하나로 클리핑하고, 새로운 폴리곤의 정점 좌표 데이터를 생성하는 정점 좌표 계산기를 포함하되;

상기 제어 회로는, 상기 폴리곤에 대한 적어도 두 면들의 클리핑이 수행된 후 동치 좌표(w)가 0일 때 상기 새로운 폴리곤이 상기 뷰 볼륨의 복수의 면들 중 다른 면으로 클리핑되도록 제어하는 것을 특징으로 하는 그래픽 처리 장치.

청구항 7

제 6 항에 있어서,

상기 제어 회로는,

상기 폴리곤의 정점들 각각의 동치 좌표(w)가 모두 양수가 아니고 그리고 모두 음수가 아닐 때 상기 폴리곤을

상기 뷰 볼륨의 상기 복수의 면들 중 적어도 두 면들에 대한 클리핑이 수행되도록 제어하는 것을 특징으로 하는 그래픽 처리 장치.

청구항 8

제 6 항에 있어서,

상기 정점 좌표 계산기에 의해서 계산된 폴리곤의 동치 좌표를 일반 좌표로 변환하는 원근 계산 유닛을 더 포함하는 것을 특징으로 하는 그래픽 처리 장치.

청구항 9

제 8 항에 있어서,

상기 제어 회로는,

상기 폴리곤의 상기 정점들이 상기 뷰 볼륨의 안쪽에 위치할 때 상기 정점 좌표 계산기에 의해서 계산된 폴리곤의 동치 좌표를 상기 원근 계산 유닛으로 입력하는 것을 특징으로 하는 그래픽 처리 장치.

청구항 10

제 9 항에 있어서,

상기 폴리곤 뷰 볼륨 결정기는,

상기 폴리곤의 상기 정점들이 상기 뷰 볼륨의 안쪽 또는 바깥쪽에 위치하는 지를 나타내는 위치 정보 신호를 상기 제어기로 제공하는 것을 특징으로 하는 그래픽 처리 장치.

청구항 11

제 10 항에 있어서,

상기 폴리곤 뷰 볼륨 결정기는,

상기 폴리곤의 상기 정점들의 좌표 데이터와 상기 뷰 볼륨의 좌표 데이터를 비교하고, 위치 정보 신호를 발생하는 비교 회로; 그리고

상기 위치 정보 신호를 저장하기 위한 레지스터 블록을 포함하는 것을 특징으로 하는 그래픽 처리 장치.

청구항 12

제 11 항에 있어서,

상기 폴리곤 뷰 볼륨 결정기는,

상기 제어 회로로부터의 제1 선택 신호에 응답해서 상기 뷰 볼륨의 상기 복수의 면들 중 어느 하나에 대응하는 좌표 값을 출력하는 제1 멀티플렉서와;

상기 제1 멀티플렉서로부터 출력되는 좌표 값과 상기 폴리곤의 동치 좌표(w)를 비교하고, 비교 결과를 출력하는 제1 판별기와;

상기 제어 회로로부터의 제1 선택 신호에 응답해서 상기 뷰 볼륨의 상기 복수의 면들 중 어느 하나에 대응하는 좌표 값을 출력하는 제2 멀티플렉서; 그리고

상기 제2 멀티플렉서로부터 출력되는 좌표 값과 상기 폴리곤의 상기 동치 좌표(w)를 비교하고, 비교 결과를 출력하는 제2 판별기를 포함하는 것을 특징으로 하는 그래픽 처리 장치.

청구항 13

제 11 항에 있어서,

상기 레지스터 블록은,

상기 폴리곤의 정점들 각각의 동치 좌표(w)의 부호를 더 저장하는 것을 특징으로 하는 그래픽 처리 장치.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <24> 본 발명은 그래픽 처리 장치에 관한 것이다.
- <25> 컴퓨터 시스템은 보통 디스플레이 화면 상에 그래픽 객체를 표시하기 위해 사용된다. 3차원 컴퓨터 그래픽의 용도는 일반적으로 객체 또는 객체들을 3차원으로 사실적으로 나타내는 컴퓨터 화면상에 3차원 이미지를 생성하기 위한 것이다. 현실 세계에서, 객체들은 실제 높이, 실제 폭 및 실제 깊이를 갖는 3 차원을 차지한다. 사진은 3차원 공간의 2차원 표현의 한 예이다. 3차원 컴퓨터 그래픽은 일반적으로, 하위 이미지가 3차원 기하학 및 표면 텍스처로 모델링되는 점을 제외하고는 컴퓨터 화면의 3차원 공간 상에 3차원 세계를 나타낸다는 점에서 사진과 유사하다.
- <26> 3차원 그래픽으로 생성된 이미지는 주어진 시점에서 씬(scene)의 개별 뷰(view)를 사실적인 방식으로 표현하기 위해, 비디오 오락 게임에서부터 만화 영화, 항공기 비행 시뮬레이터까지 광범위한 응용분야에서 사용된다. 최근에는 PMP(portable multimedia player), 휴대폰, PDA(personal digital[data] assistant) 등과 같은 모바일 영상 기기에서도 3차원 그래픽을 표시하기도 한다.
- <27> 최근 몇년 동안에 특히 굉장한 성장을 보이고 있는 한가지 산업은 컴퓨터 게임 산업이며, 3차원 그래픽 표시 속도는 점점 더 빨라진다. 최근에는 비교적 저렴한 시스템에서의 3차원 그래픽의 신속하고 융통성인 렌더링에 대한 요구를 불러일으키고 있다.
- <28> 3차원 그래픽 시스템은 3차원 피사체의 장면(scene)을 모델링하여 표시 장치에 표시될 수 있는 데이터 신호로 변환한다. 3차원 피사체의 장면은 그 피사체의 형상을 근사화하는 다수의 폴리곤(poligon)(또는 프리미티브(primitive))으로 표현된다. 2차원 표시 장치 상에 3차원 영상을 표시하는 프로세스는 복잡한 계산을 요한다. 따라서 이러한 프로세스는 오늘날의 마이크로프로세서 및 그래픽 처리 장치에 의해서 조차 속도가 느린 경우가 빈번하다.
- <29> 래스터화(lasterlize)는 그래픽 폴리곤의 단순한 기하학적 표현을 디스플레이용의 화소로 변환시키는 프로세스이다. 일반적으로 폴리곤은 점(dot), 선(line), 삼각형(triangle) 중 어느 한 형태를 갖는다. 일반적으로 피사체는 래스터화에 앞서 하나 이상의 폴리곤으로 변환된다. 삼각형 폴리곤은 좌표(x, y, z) 및 자신의 각 정점(vertex)에서의 그 밖의 다른 속성 예를 들면, 색상(color), 텍스처(texture) 좌표 등에 의해 표시된다. 폴리곤의 정점의 좌표(x, y)는 표시 장치 상에서의 위치를 나타낸다. 좌표(z)는 정점이 3차원 장면의 선택된 뷰 포인트(view point)로부터 떨어져 있는 거리를 나타낸다.
- <30> 3차원 그래픽 처리 속도를 향상시키기 위한 방법 가운데 하나는 뷰 볼륨 클리핑(view volume clipping)을 수행하는데 있어서, 상(top), 하(bottom), 좌(left), 우(right), 전(near) 및 후(far) 모든 면들에 대한 클리핑 대신 전면과 후면에 대해서만 클리핑을 수행하는 것이다. 나머지 상, 하, 좌 및 우면에 대해서는 클리핑을 수행하지 않고 래스터화 단계에서 뷰 포트(view port) 내에 있는 프래그먼트(fragment)는 표시되도록 하고, 뷰 포트 밖의 보여지지 않는 부분에 대한 삭제가 가능하다.
- <31> 그러나, 뷰 볼륨의 전면과 후면에 대해서만 클리핑을 수행하는 경우 뷰 볼륨의 바운더리(boundary)를 나타내는 변수(w)가 0이 되는 경우가 발생할 수 있다. 변수(w)가 0이 되는 경우 3차원 그래픽 시스템의 데이터 연산에 있어서 오류가 발생한다.

발명이 이루고자 하는 기술적 과제

- <32> 따라서 본 발명의 목적은 그래픽 처리 속도를 향상시키되 안정된 동작을 보장하는 그래픽 처리 시스템을 제공하는데 있다.

발명의 구성 및 작용

- <33> 상술한 바와 같은 목적을 달성하기 위한 본 발명의 특징에 의하면, 그래픽 처리 방법은, 제1 폴리곤을 뷰 볼륨

의 전면으로 클리핑하여 제2 폴리곤을 생성하는 단계와, 상기 제2 폴리곤을 상기 뷰 볼륨의 후면으로 클리핑하여 제3 폴리곤을 생성하는 단계와, 상기 제3 폴리곤의 동치 좌표(w)가 0인지를 판별하는 단계, 그리고 상기 제3 폴리곤의 상기 동치 좌표(w)가 0일 때 상기 제3 폴리곤을 상기 뷰 볼륨의 좌면, 우면, 상면 및 하면 중 어느 하나로 클리핑하는 단계를 포함한다.

- <34> 이 실시예에 있어서, 상기 그래픽 처리 방법은, 상기 제3 폴리곤의 상기 동치 좌표(w)가 0이 아니면 상기 제3 폴리곤에 대한 동치 좌표를 일반 좌표로 변환하는 단계를 더 포함한다.
- <35> 이 실시예에 있어서, 상기 그래픽 처리 방법은, 상기 제1 폴리곤의 정점들이 상기 뷰 볼륨의 전면의 안쪽에 위치할 때 상기 제1 폴리곤에 대한 동치 좌표를 일반 좌표로 변환하는 단계를 더 포함한다.
- <36> 이 실시예에 있어서, 상기 그래픽 처리 방법은, 상기 제2 폴리곤의 정점들이 상기 뷰 볼륨의 후면의 안쪽에 위치할 때 상기 제2 폴리곤에 대한 동치 좌표를 일반 좌표로 변환하는 단계를 더 포함한다.
- <37> 이 실시예에 있어서, 상기 그래픽 처리 방법은, 상기 제1 폴리곤의 정점들 각각의 동치 좌표(w)가 음수 또는 양수인지 판별하는 단계를 더 포함한다.
- <38> 이 실시예에 있어서, 상기 그래픽 처리 방법은, 상기 제1 폴리곤의 정점들 각각의 동치 좌표(w)가 모두 양수가 아니고 그리고 모두 음수가 아닐 때 상기 제1 폴리곤을 상기 뷰 볼륨의 상기 전면으로 클리핑하는 단계를 수행한다.
- <39> 본 발명의 다른 특징에 의하면, 그래픽 처리 장치는: 제어 신호들을 발생하는 제어 회로와, 상기 제어 신호들에 응답해서 폴리곤의 정점들이 뷰 볼륨 내에 위치하는 지를 판별하는 폴리곤 뷰 볼륨 결정기, 그리고 상기 제어 신호들에 응답해서 폴리곤을 상기 뷰 볼륨의 복수의 면들 중 하나로 클리핑하고, 새로운 폴리곤의 정점 좌표 데이터를 생성하는 정점 좌표 계산기를 포함한다. 상기 제어 회로는, 상기 폴리곤에 대한 적어도 두 면들의 클리핑이 수행된 후 동치 좌표(w)가 0일 때 상기 새로운 폴리곤이 상기 뷰 볼륨의 복수의 면들 중 다른 면으로 클리핑되도록 제어한다.
- <40> 이 실시예에 있어서, 상기 제어 회로는, 상기 폴리곤의 정점들 각각의 동치 좌표(w)가 모두 양수가 아니고 그리고 모두 음수가 아닐 때 상기 폴리곤을 상기 뷰 볼륨의 상기 복수의 면들 중 적어도 두 면들에 대한 클리핑이 수행되도록 제어한다.
- <41> 이 실시예에 있어서, 상기 정점 좌표 계산기에 의해서 계산된 폴리곤의 동치 좌표를 일반 좌표로 변환하는 원근 계산 유닛을 더 포함한다.
- <42> 이 실시예에 있어서, 상기 제어 회로는, 상기 폴리곤의 상기 정점들이 상기 뷰 볼륨의 안쪽에 위치할 때 상기 정점 좌표 계산기에 의해서 계산된 폴리곤의 동치 좌표를 상기 원근 계산 유닛으로 입력한다.
- <43> 이 실시예에 있어서, 상기 폴리곤 뷰 볼륨 결정기는, 상기 폴리곤의 상기 정점들이 상기 뷰 볼륨의 안쪽 또는 바깥쪽에 위치하는 지를 나타내는 위치 정보 신호를 상기 제어기로 제공한다.
- <44> 이 실시예에 있어서, 상기 폴리곤 뷰 볼륨 결정기는, 상기 폴리곤의 상기 정점들의 좌표 데이터와 상기 뷰 볼륨의 좌표 데이터를 비교하고, 위치 정보 신호를 발생하는 비교 회로, 그리고 상기 위치 정보 신호를 저장하기 위한 레지스터 블록을 포함한다.
- <45> 이 실시예에 있어서, 상기 폴리곤 뷰 볼륨 결정기는, 상기 제어 회로로부터의 제1 선택 신호에 응답해서 상기 뷰 볼륨의 상기 복수의 면들 중 어느 하나에 대응하는 좌표 값을 출력하는 제1 멀티플렉서와, 상기 제1 멀티플렉서로부터 출력되는 좌표 값과 상기 폴리곤의 동치 좌표(w)를 비교하고, 비교 결과를 출력하는 제1 판별기와, 상기 제어 회로로부터의 제1 선택 신호에 응답해서 상기 뷰 볼륨의 상기 복수의 면들 중 어느 하나에 대응하는 좌표 값을 출력하는 제2 멀티플렉서, 그리고 상기 제2 멀티플렉서로부터 출력되는 좌표 값과 상기 폴리곤의 상기 동치 좌표(w)를 비교하고, 비교 결과를 출력하는 제2 판별기를 포함한다.
- <46> 상기 레지스터 블록은, 상기 폴리곤의 정점들 각각의 동치 좌표(w)의 부호를 더 저장한다.
- <47> 이하 본 발명의 바람직한 실시예를 첨부된 도면을 참조하여 상세히 설명한다.
- <48> 도 1은 본 발명의 바람직한 실시예에 따른 3차원 그래픽 처리 장치의 그래픽 파이프라인을 보여주고 있다.
- <49> 그래픽 파이프 라인(100)은 정점 셰이더(vertex shader, 110), 기하학 엔진(geometry engine, 120), 셋업 및 래스터 엔진(setup and rasterizing engine, 130), 프래그먼트 셰이더(fragment shader, 140) 그리고 퍼프레그

먼트 유닛(perfragment unit, 150)을 포함한다.

- <50> 호스트(미 도시됨)로부터 입력되는 그래픽 영상 신호(IN)는 정점 셰이더(110)로 입력된다. 정점 셰이더(110)는 정점의 좌표, 색상, 반사 값 등의 다양한 정보를 이용하여 3차원 그래픽 오브젝트를 구현하는데, 정점의 위치가 변하게 됨으로써 함께 달라지는 좌표 값을 비롯한 행렬, 광원(light) 및 질감(texture) 등 수 많은 데이터에 대한 복잡한 연산을 처리한다. 정점 셰이더(110)는 좌표(x, y, z, w)를 출력한다.
- <51> 기하학 엔진(120)은 셋업 및 래스터 엔진(130)이 처리할 이미지를 생성하기 위하여 폴리곤과 기타 그래픽 데이터를 처리한다. 셋업 및 래스터 엔진(130)은 정점 셰이더(110)로부터 입력된 정점을 표시 장치 상에 뷰잉(viewing)하기 위한 형태로 변환한다. 일반적으로 기하학 엔진(120)은 조명원(lighting source)로부터 색채 컨트리뷰션(color contribution)을 생성하고, 뷰어(viewer)로부터의 거리가 멀어질수록 객체의 가시도가 낮아지도록 하는 포그(fog factor)를 생성하며, 하나의 장면(scene)을 주어진 뷰 볼륨으로 클리핑한다.
- <52> 셋업 및 래스터 엔진(130)은 스크린 좌표로 변환된 정점을 입력받고, 정점 사이의 색채를 보간하거나, 이미지를 매핑하여 객체의 정점 표현을 솔리드 객체(solid object)로 변환한다.
- <53> 셋업 및 래스터 엔진(130)으로부터 출력되는 신호는 프래그먼트 셰이더(140) 및 퍼프래그먼트 유닛(150)을 통과하여 디스플레이 장치(미 도시됨)로 제공된다.
- <54> 도 2는 도 1에 도시된 기하학 엔진(120)의 구성을 구체적으로 보여주는 블록도이다.
- <55> 도 2를 참조하면, 기하학 엔진(120)은 클리핑 유닛(clipping unit, 121), 원근 계산 유닛(perspective division unit, 122) 그리고 뷰포트 매핑 유닛(view port mapping unit, 123)을 포함한다.
- <56> 클리핑 유닛(121)은 객체를 뷰 볼륨(view volume)으로 클리핑한다. 원근 계산 유닛(122)은 제4좌표인 W를 계산한다. 본 발명은 제4 좌표인 W를 사용하여 픽셀 좌표(x, y, z)를 보정한다. 뷰포트 변환 유닛(123)은 정규화된 디바이스 좌표를 스크린 또는 윈도우 좌표로 변환한다. 디바이스 좌표는 이미지를 표시 장치에 표시하기 위해 사용되는 좌표이다.
- <57> 클리핑 유닛(121)은 어플리케이션 모델들에 대해서 뷰 볼륨을 정의하고, 뷰 볼륨 내에 있는 폴리곤의 정점 데이터(vertex data)는 그래픽 파이프 라인의 다음 단으로 전달하고, 뷰 볼륨 바깥에 있는 정점 데이터는 버림으로써 불필요 연산이 제거되도록 한다. 이와 같이 뷰 볼륨에 대해서 보여지는 부분과 보여지지 않는 부분을 구별하는 프로세스를 3차원 그래픽의 뷰 볼륨 클리핑이라고 한다.
- <58> 도 1에 도시된 정점 셰이더(110)는 모델(model)-뷰어(viewer) 변환을 위한 매트릭스(matrix) 연산을 수행하고 투시도법(perspective projection)을 통해서 사용자에게 의해 정의된 뷰볼륨에 투영(projection)시킨다. 이 과정에서 동차 좌표(homogeneous coordinate)가 사용되어 3차원이 아닌 4차원 좌표계가 사용된다. 3차원에서 한 점을 정의할 때 좌표는 (x, y, z)로 정의된다. 하지만, 3차원에서 변환(이동, 회전 등)을 수행할 때는 간편하고 빠르게 적용하기 위해 직접 계산 대신 4×4행렬을 적용한다. 이 4×4행렬을 적용하게 되면 성분 하나가 추가되어야 하며, 이 때 w라는 성분 하나를 추가해 (x, y, z, w)를 구성하는데 이를 동차 좌표라 한다. 3차원에서의 (x, y, z)를 동차좌표로 표현하면 (x, y, z, w)이 된다. 그런데, 실제 3차원의 물체를 2차원으로 투영하기 위해 원근변환을 적용한 뒤에는 좌표값이 바뀌어 출력된다. 즉, 마지막 w 성분이 다른 값을 가질 수 있다. 따라서 실제로 스크린상의 좌표 값으로 사용하기 위해서는 동차 좌표를 일반 좌표로 바꿔야 하는데, 이를 위해서는 좌표의 각 성분(x, y, z)을 w로 나눠서 구한다. 즉, (x/w, y/w, z/w, w/w) = (x/w, y/w, z/w, 1)이 되고 네번째 성분값이 1인 일반적인 좌표로 바뀐다. 원근 계산 유닛(122)은 상술한 바와 같이 동차 좌표를 일반 좌표로 변환한다.
- <59> 이와 같이 좌표의 각 성분(x, y, z)을 w로 나누기 위해서 w는 0이 아닌 다른 값을 가져야 한다. 그러나, 클리핑 유닛(121)에서 뷰 볼륨 클리핑을 수행하는데 있어서, 전면(near plane) 및 후면(far plane)에 대한 클리핑을 수행시 w가 0이 되는 경우가 발생할 수 있다.
- <60> 모델을 이루는 정점의 동차 좌표(x, y, z, w)는 클리핑 유닛(121)에 의해서 원근 투영 매트릭스(perspective projection matrix) 연산이 수행된다. 원근 투영 매트릭스 연산이 수행된 후 클리핑 유닛(121)으로부터 출력되는 동차 좌표(x', y', z', w')에서 x', y', z' 각각이 w'보다 작거나 같은 값을 가지면 정점 좌표는 뷰 볼륨 내에 존재하고, x', y', z' 각각이 w'보다 크면 정점 좌표는 뷰 볼륨 바깥에 위치한다. 결국 x'와 w', y'와 w', 그리고 z'와 w'를 비교하여 완전히 뷰 볼륨 내에 있는 데이터 값을 찾을 수 있다.

<61> 이를 정리하여 나타내면 수학적 식 1과 같다.

수학적 식 1

<62> $-w \leq x \leq w, -w \leq y \leq w, -w \leq z \leq w$ (단, $w > 0$ 일 때)

<63> $w \leq x \leq -w, w \leq y \leq -w, w \leq z \leq -w$ (단, $w < 0$ 일 때)

<64> 도 3a는 동치 좌표(z, w)에 대한 삼각형 폴리곤을 보여주고 있고, 도 3b는 도 3a에 도시된 폴리곤에 대한 전면 클리핑을 수행한 결과를 보여주며, 그리고 도 3c는 도 3b에 도시된 폴리곤에 대한 후면 클리핑을 수행한 결과를 보여주고 있다.

<65> 도 3a 내지 도 3c에 도시된 바와 같이, 폴리곤이 뷰 볼륨의 경계에 걸치게 되면 새로운 폴리곤을 형성하는 정점 좌표 데이터가 구해져야 한다. 도 3a에 도시된 폴리곤의 정점은 a, b 및 c였으나, 전면 클리핑이 수행된 후 폴리곤의 정점은 a, d 및 e로 변경되었다. 또한 후면 클리핑이 수행된 후 폴리곤의 정점은 a, e, f 및 g로 변경된다. 폴리곤의 정점(g)의 w좌표는 0이다.

<66> $w=0$ 인 경우, 원근 계산 유닛(122)에 의해서 동치 좌표가 일반 좌표로 변환되기 위하여 ($x/w, y/w, z/w, w/w$) 연산이 수행되는 경우 잘못된 연산 결과가 나온다. 본 발명은 w 가 0이 됨으로써 잘못된 연산이 실행되는 것을 막을 수 있는 스킴을 제공한다.

<67> 본 발명은 다양한 어플리케이션을 지원하기 위해 음의 w 를 가지는 폴리곤에 대해서 올바르게 클리핑하기 위한 새로운 클리핑 스킴을 제공한다. 이는 초기의 모든 클리핑 면들에 대해서 클리핑을 수행하는 것보다 동작 사이클은 감소하면서도 올바른 클리핑 결과를 다음 파이프라인 단으로 넘겨주는 것을 가능하게 한다.

<68> 도 4는 본 발명의 바람직한 실시예에 따른 클리핑 유닛을 구체적으로 보여주는 블록도이다.

<69> 도 4를 참조하면 클리핑 유닛(121)은 연산 유닛(400) 및 정점 메모리(490)를 포함한다. 연산 유닛(400)은 폴리곤에 대한 클리핑을 수행하며, 클리핑된 정점들에 대한 좌표 정보를 정점 메모리(vertex memory, 490)에 저장한다. 연산 유닛(400)은 FSM(finite state machine, 410), 폴리곤 뷰 볼륨 결정기(420), 정점 좌표 계산기(430), 폴리곤 리어셈블러(440), 어드레스 발생기(450) 그리고 데이터 버퍼(460)를 포함한다.

<70> FSM(410)은 연산 유닛(400)에 대한 전반적인 동작을 제어한다. 폴리곤 뷰 볼륨 결정기(420)는 정점 셰이퍼(110)로부터 입력되는 정점 좌표가 뷰 볼륨의 전면 및 후면 클리핑 면들의 안쪽(inside)에 위치하는지 바깥쪽(outside)에 위치하는 지를 나타내는 위치 정보를 출력한다.

<71> FSM(410)은 폴리곤 뷰 볼륨 결정기(420)로부터 출력되는 위치 정보를 입력받고, 정점 좌표들 모두가 뷰 볼륨의 전면 및 후면 클리핑 면들의 안쪽에 위치하면 정점 좌표들을 폴리곤 리어셈블러(440)로 제공한다. 만일 정점 좌표들 모두가 뷰 볼륨의 전면 및 후면 클리핑 면들의 바깥쪽에 위치하면 그 좌표를 삭제한다. 그리고 정점 좌표들이 전면 클리핑 면에 걸쳐 있거나 후면 클리핑 면에 걸쳐 있으면 폴리곤에 대한 전, 후면 클리핑뿐만 아니라 좌, 우, 상, 하 면들 각각에 대한 클리핑이 순차적으로 수행될 수 있도록 폴리곤 뷰 볼륨 결정기(420)를 제어한다.

<72> 정점 좌표 계산기(430)는 폴리곤에 대한 클리핑을 수행하고 정점들의 좌표를 계산한다.

<73> 폴리곤 리어셈블러(440)는 폴리곤에 대한 좌, 우, 상, 하 면들 각각에 대한 클리핑이 수행되었을 때 새로 생성된 폴리곤들을 연결(link)하기 위한 연결 정보를 생성한다.

<74> 어드레스 발생기(450)는 정점 메모리(490)를 액세스하기 위한 제어 신호 및 어드레스 신호를 발생한다.

<75> 데이터 버퍼(460)는 정점 메모리(490)로부터 독출된 정점 데이터를 저장하기 위한 독출 버퍼(462), 정점 메모리(490)에 저장될 정점 데이터를 저장하는 기입 버퍼(464) 그리고 w 가 0인 지를 검사하는 w 체커(466)를 포함한다.

<76> 도 5는 도 4에 도시된 폴리곤 뷰 볼륨 결정기(420)의 구체적인 구성을 보여주는 도면이다.

<77> 도 5를 참조하면, 폴리곤 뷰 볼륨 결정기(420)는 멀티플렉서들(510, 520)과 제1 및 제2 판별기들(530, 540) 그리고 레지스터 블록(550)을 포함한다. 폴리곤 뷰 볼륨 결정기(420)는 FSM(410)으로부터 점(point), 라인(line), 삼각형(triangle) 등과 같은 형태의 폴리곤 좌표($V1(x,y,z,w)$)를 입력받는다.

<78> 멀티플렉서들(510, 520) 각각은 뷰 볼륨의 좌면(left plane) 좌표값(-x), 우면(right plane) 좌표값(+x), 상면

(top plane) 좌표값(+y), 하면(bottom plane) 좌표값(-y), 전면(near plane) 좌표값(-z) 그리고 후면(far plane) 좌표값(+z)을 입력받는다. FSM(410)은 멀티플렉서들(510, 520) 각각으로 선택 신호들(SEL1, SEL2)을 입력한다.

- <79> 제1 판별기(530)는 멀티플렉서(510)로부터의 출력과 현재 폴리곤의 좌표(V1(x,y,z,w)) 입력받고, 현재 폴리곤이 뷰 볼륨 내에 위치하는지의 여부를 판별한다. 제1 판별기(530)는 폴리곤의 좌표(V1(x,y,z,w))가 멀티플렉서(510)로부터 출력되는 뷰 볼륨의 좌표값보다 안쪽에 위치하면 '1'을 출력하고, 폴리곤의 좌표(V1(x,y,z,w))가 멀티플렉서(510)로부터 출력되는 뷰 볼륨의 좌표값보다 바깥쪽에 위치하면 '0'을 출력한다.
- <80> 제2 판별기(540)는 멀티플렉서(520)로부터의 출력과 현재 폴리곤의 좌표(V1(x,y,z,w))를 입력받고, 현재 폴리곤이 뷰 볼륨 내에 위치하는지의 여부를 판별한다. 제2 판별기(540)는 폴리곤의 좌표(V1(x,y,z,w))가 멀티플렉서(520)로부터 출력되는 뷰 볼륨의 좌표값보다 안쪽에 위치하면 '1'을 출력하고, 폴리곤의 좌표(V1(x,y,z,w))가 멀티플렉서(520)로부터 출력되는 뷰 볼륨의 좌표값보다 바깥쪽에 위치하면 '0'을 출력한다.
- <81> 레지스터 블록(550)은 현재 폴리곤을 나타내는 세 개의 정점들(a, b, c) 각각의 좌표(V1(w))의 부호를 저장하기 위한 레지스터(551)를 포함한다. 레지스터(551)는 세 개의 정점들(a, b, c)에 각각 대응하는 좌표(V1(w))의 부호를 나타내는 3비트 정보(W1, W2, W3)를 저장한다. 이 실시예에서 3비트 정보(W1, W2, W3)는 좌표(V1(w))가 양수이면 '1' 그리고 음수이면 '0'이다. 레지스터 블록(550)은 3비트 정보(W1, W2, W3)가 모두 '1' 인지, 모두 '0'인지, 그리고 모두 '1' 또는 '0'이 아닌지의 여부를 나타내는 제1 신호(WC)를 출력한다. 예컨대, 3비트 정보(W1, W2, W3)가 모두 '1' 이면 제1 신호(WC)는 '10', 3비트 정보(W1, W2, W3)가 모두 '0'이면, 제1 신호(WC)는 '01' 그리고 3비트 정보(W1, W2, W3)가 모두 '1' 또는 '0'이 아니면 제1 신호(WC)는 '00'이다.
- <82> FSM(410)은 제1 신호(WC)가 '00'이면 뷰 볼륨의 전면 클리핑 및 후면 클리핑 뿐만 아니라, 좌, 우, 상, 하면에 대한 클리핑이 필요한 것으로 판단하고, 폴리곤 뷰 볼륨 결정기(420)에 의해서 정점 좌표(V1(x,y,z,w))가 뷰 볼륨 안쪽 또는 바깥쪽에 위치하는지 판별되도록 제어한다.
- <83> FSM(410)은 제1 신호(WC)가 '10'이면 정점 좌표 계산기(430)에 의해서 정점의 교차(intersection) 좌표가 계산되도록 제어한다. 한편, FSM(410)은 제1 신호(WC)가 '01'이면 폴리곤이 뷰 포트의 밖에 위치한 것으로 판단하여 폴리곤이 삭제되도록 제어한다.
- <84> 다시 도 5를 참조하면, 레지스터 블록(550)은 레지스터들(552, 553, 554)을 더 포함한다. 레지스터들(552, 553, 554)은 제1 및 제2 판별기들(530, 540)로부터 출력되는 판별 결과를 저장한다. 레지스터들(552, 553, 554)에 저장된 판별 결과들은 제2 신호(POS)로서 FSM(410)으로 제공된다. FSM(410)은 폴리곤 뷰 볼륨 결정기(420)로부터의 제2 신호(POS)를 입력받고, 클리핑 여부 및 클리핑 방식을 결정한다.
- <85> 도 6은 도 4에 도시된 FSM(410)의 클리핑 제어 수순을 보여주는 플로우차트이다.
- <86> 도 6을 참조하면, FSM(410)은 레지스터 블록(550)으로부터 입력되는 제1 신호(WC)가 '01'인지의 여부를 판별한다(610). 제1 신호(WC)가 '01'이면 FSM(410)은 폴리곤 정점들의 위치를 판별하도록 폴리곤 뷰 볼륨 결정기(420)를 제어한다. FSM(410)은 뷰 볼륨의 6면 중 하나의 면 좌표, 폴리곤 좌표(V1(x,y,z,w)), 선택 신호들(SW1, SW2)을 폴리곤 뷰 볼륨 결정기(420)로 입력한다.
- <87> 멀티플렉서(510)는 선택 신호(SEL1)에 응답해서 전면 좌표(+z)를 출력한다. 제1 판별기(530)는 현재 폴리곤 정점들의 좌표(V1(w))와 전면 좌표(+z)를 비교하여 폴리곤의 정점들 중 하나 또는 두 개의 정점들이 전면(+z) 안쪽에 위치하는지를 판별한다. 이것은 " $w \geq +z$ "(단, $w > 0$) 또는 " $w \leq -z$ "(단, $w < 0$)을 만족하는지를 판별하는 것에 의해서 가능하다. 판별 결과는 레지스터(552)에 저장된다. 도 2a에 도시된 세 개의 정점들(a, b, c) 각각의 좌표(V1(w))와 전면 좌표(+z)의 비교 결과는 레지스터(552)에 N1, N2, N3으로 저장된다. 예컨대, 정점(a)의 좌표(V1(w))가 전면 좌표(+z) 안에 위치할 때 N1은 '1'이고, 밖에 위치할 때 N1은 '0'이다. 이와 같은 방법으로 N2 및 N3의 값도 결정된다.
- <88> 멀티플렉서(520)는 선택 신호(SEL2)에 응답해서 후면 좌표(-z)를 출력한다. 제2 판별기(540)는 현재 폴리곤 정점들의 좌표(V1(w))와 후면 좌표(-z)를 비교하여 폴리곤의 정점들 중 하나 또는 두 개의 정점들이 후면(-z) 안쪽에 위치하는지를 판별한다. 이것은 " $w \geq -z$ "(단, $w > 0$) 또는 " $w \leq -z$ "(단, $w < 0$)을 만족하는지를 판별하는 것에 의해서 가능하다. 판별 결과는 레지스터(553)에 저장된다. 도 2a에 도시된 세 개의 정점들(a, b, c) 각각의 좌표(V1(w))와 후면 좌표(-z)의 비교 결과는 레지스터(553)에 F1, F2, F3으로 저장된다. 예컨대, 정점(a)의 좌표(V1(w))가 후면 좌표(-z) 안에 위치할 때 F1은 '1'이고, 밖에 위치할 때 F1은 '0'이다. 이와 같은 방법으로

F2 및 F3의 값도 결정된다.

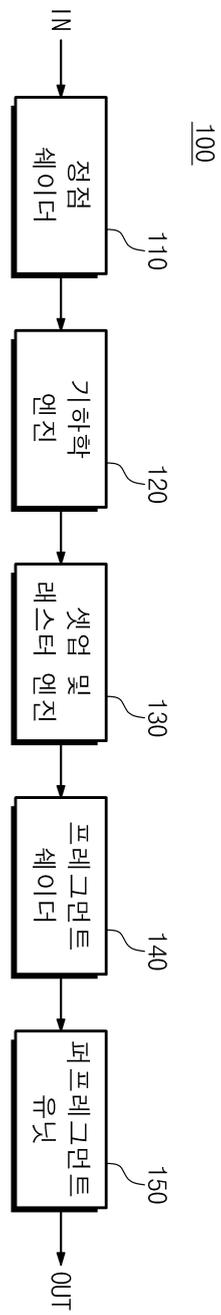
- <89> FSM(410)은 레지스터 블록(550)의 레지스터(552)에 저장된 위치 정보(N1, N2, N3)와 레지스터(553)에 저장된 위치 정보(F1, F2, F3)를 입력받고 폴리곤의 정점들이 전면(+z) 안쪽에 위치하는지의 여부를 판별한다(612). 만일 폴리곤의 정점들이 전면(+z) 안쪽에 위치하는 것으로 판별되면 FSM(410)은 폴리곤에 대한 클리핑 제어를 종료한다. 만일 폴리곤의 정점들이 전면(+z) 바깥쪽에 위치하는 것으로 판별되면 FSM(410)은 정점 좌표 계산기(430)가 폴리곤을 전면 클리핑하여 새로운 폴리곤을 생성하고, 새로운 폴리곤의 정점 좌표들을 계산하도록 제어한다(614). $w=+z$ 축을 기준으로 전면 클리핑되어서 새로 생성된 폴리곤이 도 2b에 도시되어 있다. 새로운 폴리곤은 정점들(a, d, e)을 포함한다.
- <90> FSM(410)은 폴리곤 뷰 볼륨 결정기(420)가 새로 생성된 폴리곤의 정점들이 후면(-z) 안쪽에 위치하는지를 판별하도록 제어한다(616). 폴리곤 뷰 볼륨 결정기(420) 내 멀티플렉서(510)는 선택 신호(SEL1)에 응답해서 후면 좌표(-z)를 출력한다. 제1 판별기(530)는 새로운 폴리곤의 정점들(d, e) 각각의 좌표(V1(w))와 후면 좌표(-z)를 비교하고, 비교 결과를 레지스터(553)에 F4, F5로서 저장한다.
- <91> FSM(410)은 폴리곤 뷰 볼륨 결정기(420)로부터 위치 정보(F4, F5)를 포함하는 제2 신호(POS)를 입력받고, 폴리곤의 정점들이 후면(-z) 안쪽에 위치하는가를 판별한다. 만일 폴리곤의 정점들이 후면(-z) 안쪽에 위치하면 FSM(410)은 클리핑 제어를 종료한다. 만일 폴리곤의 정점들이 후면(-z) 바깥쪽에 위치하면 FSM(410)은 정점 좌표 계산기(430)가 후면 클리핑을 수행하고 새로운 폴리곤을 생성하도록 제어한다. 정점 좌표 계산기(430)는 후면 클리핑을 수행한 후 도 2c에 도시된 바와 같이, 새로운 폴리곤의 정점들(f, g)의 좌표를 출력한다.
- <92> FSM(410)은 새로운 폴리곤의 정점들(f, g)의 좌표(V1(w))가 0인지를 판별한다(620). 새로운 폴리곤의 정점들(f, g)의 좌표(V1(w))가 0인지의 판별은 도 4에 도시된 데이터 버퍼(460) 내 W 체커(466)에 의해서 수행되며, FSM(410)은 W 체커(466)로부터 입력되는 결과 신호에 따라서 새로운 폴리곤의 정점들(f, g)의 좌표(V1(w))가 0인지를 판단한다. 좌표(V1(w))가 0이면 FSM(410)은 정점 좌표 계산기(430)가 $w=+x$ 축인 우면을 기준으로 클리핑을 수행하도록 제어한다(622).
- <93> 도 7a는 정점들(a, b, c,)을 포함하는 폴리곤을 보여주고 있다. 도 7b는 전면(+z) 및 후면(-z) 클리핑 후 정점들(a, e, f, g)을 포함하는 폴리곤을 보여준다. 도 7c는 도 7b의 정점(g)을 (z, w) 평면 상에 표시한 것이다. 도 7d는 $w=+x$ 축을 기준으로 클리핑했을 때 정점(g)의 변화를 보여주고 있다. 도 7e는 도 7d를 2차원 평면으로 보여주는 도면이다. 도 7d 및 도 7e에 도시된 바와 같이, $w=z=0$ 인 정점을 우면(+x)에 대해 클리핑을 수행하면 w는 0이 아닌 다른 값으로 변화하게 된다.
- <94> 도 7a의 폴리곤 정점들(a, b, c)의 좌표 데이터는 도 4에 도시된 데이터 버퍼(460)를 통해 정점 메모리(490)로부터 독출되며, 또한 정점 좌표 계산기(430)로부터 출력되는 새로운 폴리곤들의 정점들(a, d, e), (a, e, f, g)의 좌표 데이터는 데이터 버퍼(460)를 통해 정점 메모리(490)에 저장된다.
- <95> 상술한 바와 같이, 폴리곤을 뷰 볼륨의 전, 후, 좌, 우, 상, 하면 모두에 대한 클리핑 대신 전면 및 후면만 클리핑하여도 후속 단들에서 좌, 우, 상, 하면에 대한 클리핑과 동일한 결과를 얻을 수 있으므로 최근에는 전면 및 후면만 클리핑하는 기술이 널리 사용되고 있다. 그러나, 전면 및 후면에 대해서 클리핑을 수행했을 때 동치 좌표(V1(x,y,z,w))의 $w=0$ 인 경우가 발생할 수 있다. 이는 후속 단(stage)인 원근 계산 유닛(122)에서 x/w , y/w , z/w 와 같은 연산을 수행할 때 잘못된 연산 결과를 유발한다. 그러므로, 전면 및 후면에 대한 클리핑을 수행하는 $w=0$ 이면 좌, 우, 상, 하면 중 어느 한 면에 대한 클리핑을 더 수행하여 w가 0이 아닌 다른 값으로 변화하도록 할 수 있다. 그 결과, 그래픽 처리 장치의 처리 속도는 향상되고 안정된 동작이 보장된다.
- <96> 만일 전면(-z), 후면(+z) 그리고 우면(+x)에 대한 클리핑을 수행한 후 $w=0$ 이면 좌, 상, 하면 중 어느 한 면에 대한 클리핑을 다시 수행하여 w가 0이 아닌 다른 값이 되도록 FSM(410)은 상술한 과정을 반복적으로 수행한다.
- <97> 예시적인 바람직한 실시예들을 이용하여 본 발명을 설명하였지만, 본 발명의 범위는 개시된 실시예들에 한정되지 않는다는 것이 잘 이해될 것이다. 오히려, 본 발명의 범위에는 다양한 변형 예들 및 그 유사한 구성들이 모두 포함될 수 있도록 하려는 것이다. 따라서, 청구범위는 그러한 변형 예들 및 그 유사한 구성들 모두를 포함하는 것으로 가능한 폭넓게 해석되어야 한다.

발명의 효과

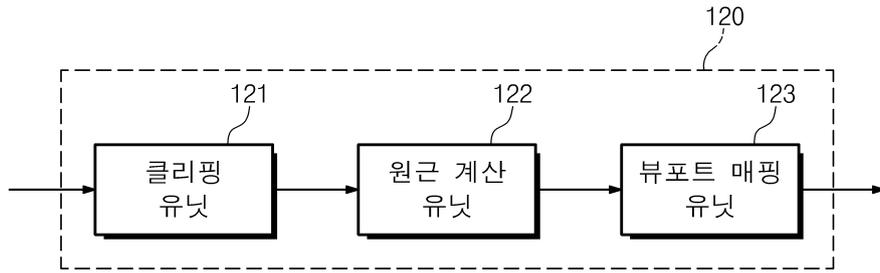
- <98> 이와 같은 본 발명에 의하면 그래픽 처리 속도를 향상시키되 안정된 동작을 수행하는 그래픽 처리 시스템이 제

도면

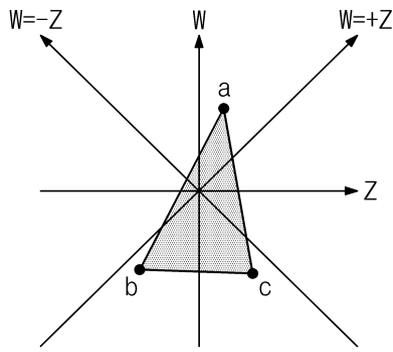
도면1



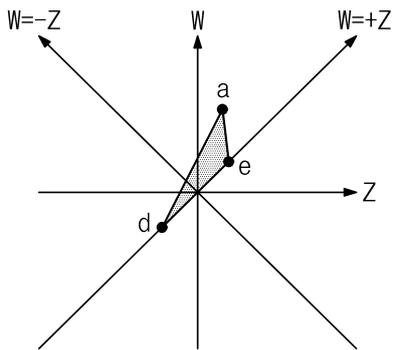
도면2



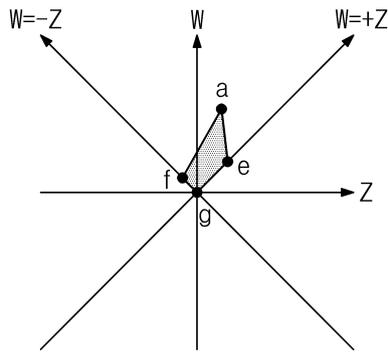
도면3a



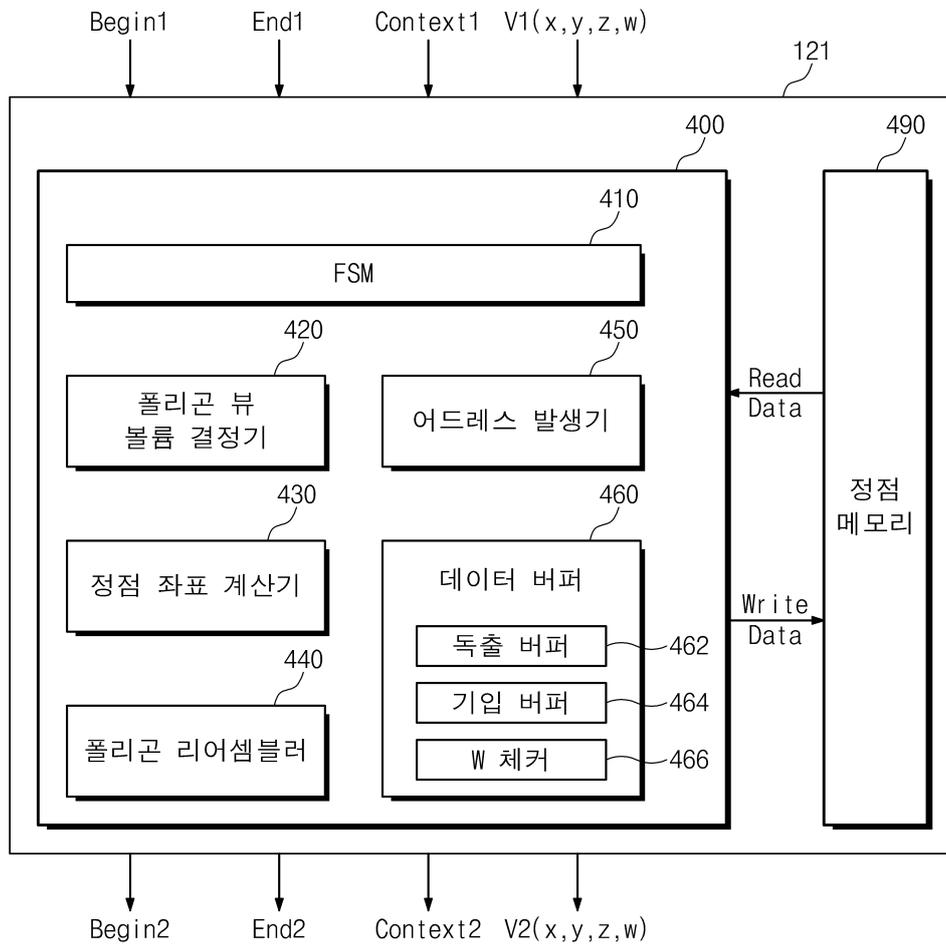
도면3b



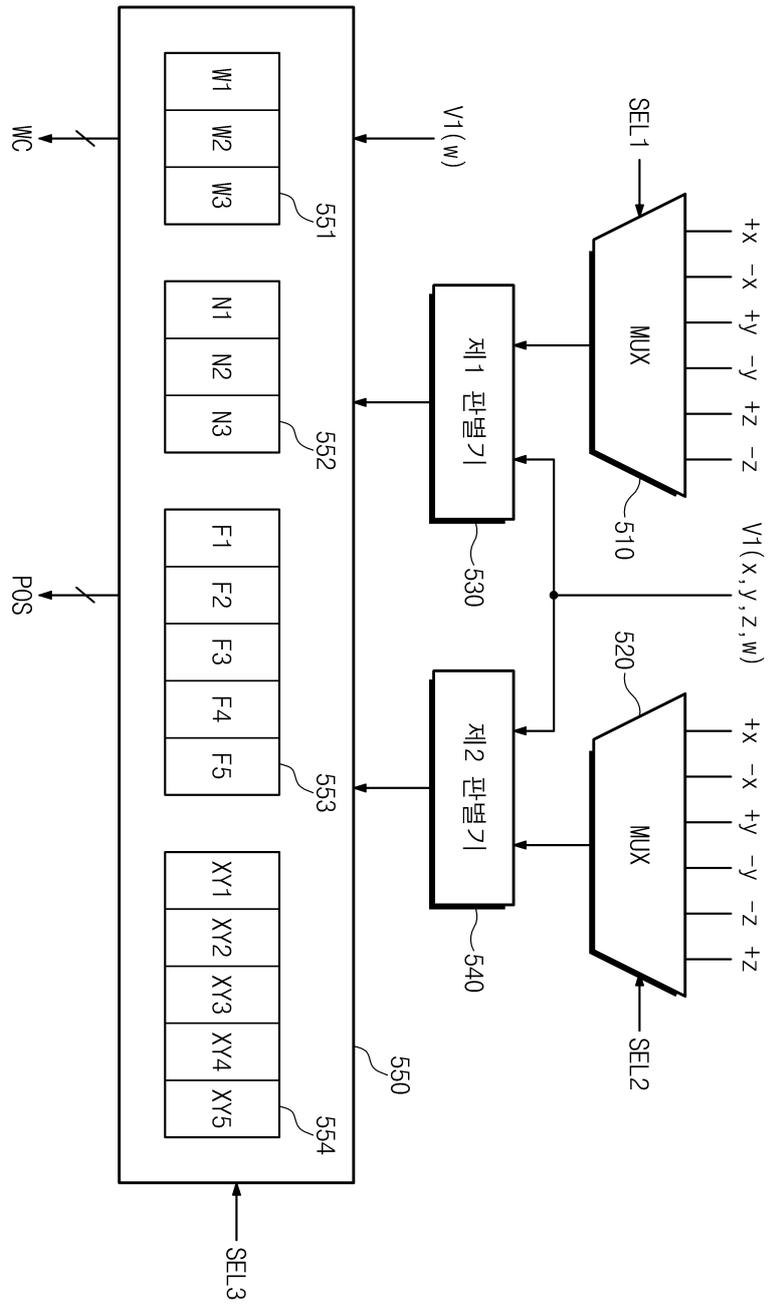
도면3c



도면4

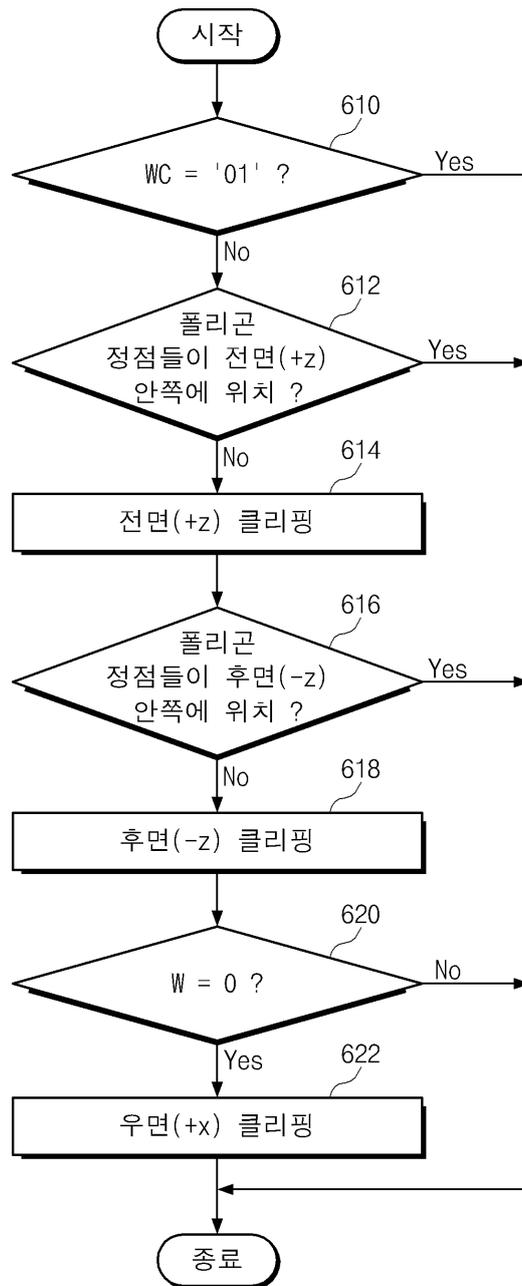


도면5

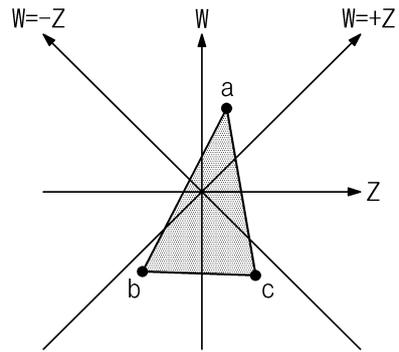


420

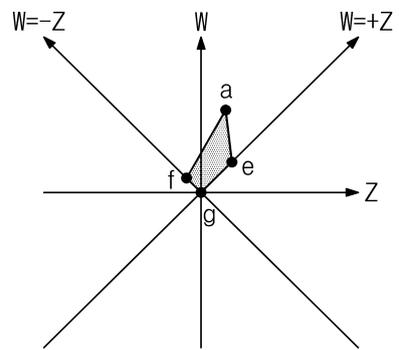
도면6



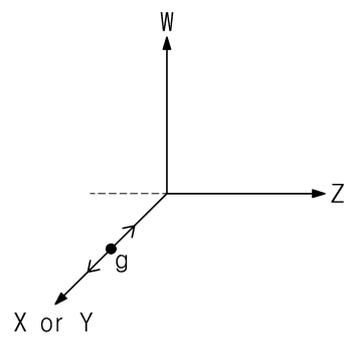
도면7a



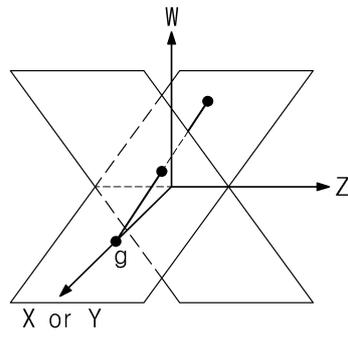
도면7b



도면7c



도면7d



도면7e

