

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4608510号  
(P4608510)

(45) 発行日 平成23年1月12日(2011.1.12)

(24) 登録日 平成22年10月15日(2010.10.15)

(51) Int.Cl. F I  
G06T 15/80 (2011.01) G06T 15/50 200

請求項の数 19 外国語出願 (全 52 頁)

(21) 出願番号	特願2007-67392 (P2007-67392)	(73) 特許権者	501261300 エヌヴィディア コーポレイション アメリカ合衆国, カリフォルニア 95050, サンタ クララ, サン トーマス エクスプレスウェイ 2701
(22) 出願日	平成19年3月15日(2007.3.15)	(74) 代理人	100094318 弁理士 山田 行一
(62) 分割の表示	特願2001-542053 (P2001-542053) の分割	(74) 代理人	100123995 弁理士 野田 雅一
原出願日	平成12年12月5日(2000.12.5)	(74) 代理人	100107456 弁理士 池田 成人
(65) 公開番号	特開2007-193835 (P2007-193835A)	(72) 発明者	ジョン・リンドホルム アメリカ合衆国, カリフォルニア州 95014 クパーティノ, バージニア・スワン・プレイス 10325
(43) 公開日	平成19年8月2日(2007.8.2)		
審査請求日	平成19年11月13日(2007.11.13)		
(31) 優先権主張番号	09/454,516		
(32) 優先日	平成11年12月6日(1999.12.6)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 単一の半導体プラットフォームで支持される変換、ライティング、ラスタ化システム

(57) 【特許請求の範囲】

【請求項1】

(a) バッファから頂点データを受信するためにバッファに結合されるように構成され、オブジェクトスペースからスクリーンスペースへ頂点データを変換する単一の半導体プラットフォーム上に位置される変換モジュールと、

(b) 変換モジュールに結合され、変換モジュールから受信された頂点データについてライティング演算を実行するために変換モジュールと同一の単一の半導体プラットフォームに位置されているライティングモジュールと、

(c) ライティングモジュールに結合され、ライティングモジュールから受信される頂点データをレンダリングするために変換モジュールとライティングモジュールと同一の単一の半導体プラットフォームに位置されているラスタ化装置とを具備し、

(d) 変換モジュールとライティングモジュールの少なくとも1つは複数の論理装置により並列して多数の演算スレッドを実行するためのシーケンサを含んでおり、

ライティングモジュールは、

(a) 頂点データを受信するように構成されている複数の入力バッファと、

(b) 入力バッファの1つの出力に結合されている第1の入力と、入力バッファの1つの出力に結合されている第2の入力とを有する乗算論理装置と、

(c) 入力バッファの1つの出力に結合されている第1の入力と、乗算論理装置の出力に結合されている第2の入力とを有する演算論理装置と、

(d) 演算論理装置の出力に結合されている入力と、乗算論理装置の第1の入力に結合さ

10

20

れている出力とを有する第 1 のレジスタ装置と、

( e ) 演算論理装置の出力に結合されている入力と、乗算論理装置の第 1 の入力および第 2 の入力に結合されている出力とを有する第 2 のレジスタ装置と、

( f ) 演算論理装置の出力に結合されている第 1 の入力と、入力バッファの 1 つの出力に結合されている第 2 の入力と、乗算論理装置の第 1 の入力に結合されている出力とを有するライティング論理装置と、

( g ) 乗算論理装置の少なくとも 1 つの入力と、演算論理装置の出力に結合されているメモリとを含んでいるグラフィック処理用のグラフィックパイプラインシステム。

【請求項 2】

入力バッファの 1 つの出力は遅延素子を介してライティングモジュールの出力に結合されている請求項 1 記載のシステム。

10

【請求項 3】

演算論理装置の出力と、入力バッファの 1 つの出力はマルチプレクサによりライティングモジュールの出力に結合されている請求項 2 記載のシステム。

【請求項 4】

乗算論理装置の出力はその第 2 の入力に結合されたフィードバックループを有している請求項 1 記載のシステム。

【請求項 5】

ライティング論理装置の第 2 の入力は遅延素子を介して入力バッファの 1 つの出力に結合されている請求項 1 記載のシステム。

20

【請求項 6】

ライティング論理装置の出力は先入れ先出しレジスタ装置を介して乗算論理装置の第 1 の入力に結合されている請求項 1 記載のシステム。

【請求項 7】

ライティング論理装置の出力はスカラー頂点データをベクトル頂点データへ変換するように構成されたコンバータモジュールを介して乗算論理装置の第 1 の入力に結合されている請求項 1 記載のシステム。

【請求項 8】

( a ) バッファから頂点データを受信するためにバッファに結合されるように構成され、オブジェクトスペースからスクリーンスペースへ頂点データを変換する単一の半導体プラットフォーム上に位置される変換モジュールと、

30

( b ) 変換モジュールに結合され、変換モジュールから受信された頂点データについてライティング演算を実行するために変換モジュールと同一の単一の半導体プラットフォーム上に位置されているライティングモジュールと、

( c ) ライティングモジュールに結合され、ライティングモジュールから受信される頂点データをレンダリングするために変換モジュールとライティングモジュールと同一の単一の半導体プラットフォーム上に位置されているラスタ化装置とを具備し、

( d ) 変換モジュールとライティングモジュールの少なくとも 1 つは複数の論理装置により並列して多数の演算スレッドを実行するためのシーケンサを含んでおり、

変換モジュールは、

40

( a ) 頂点データを受信するように構成されている入力バッファと、

( b ) 入力バッファの出力に結合された第 1 の入力を有する乗算論理装置と、

( c ) 乗算論理装置の出力に結合された第 1 の入力を有する演算論理装置と、

( d ) 演算論理装置の出力に結合する入力を有するレジスタ装置と、

( e ) 逆数または逆平方根演算を実行するために、演算論理装置またはレジスタ装置の出力に結合された入力を含んでいる反転論理装置と、

( f ) 反転論理装置の出力と、乗算論理装置の第 2 の入力との間に結合され、スカラー頂点データをベクトル頂点データに変換するように構成されているコンバータモジュールと

( g ) 乗算論理装置と演算論理装置に結合されているメモリとを含んで

50

いるグラフィック処理用のグラフィックパイプラインシステム。

【請求項 9】

メモリは乗算論理装置の第 2 の入力に結合されている請求項 8記載のシステム。

【請求項 10】

メモリは演算論理装置の出力に結合されている書込み端子を有する請求項 8記載のシステム。

【請求項 11】

乗算論理装置の出力はその第 1 の入力に結合されているフィードバックループを有している請求項 8記載のシステム。

【請求項 12】

レジスタ装置の出力は乗算論理装置の第 1 の入力に結合されている請求項 8記載のシステム。

【請求項 13】

レジスタ装置の出力は乗算論理装置の第 2 の入力に結合されている請求項 8記載のシステム。

【請求項 14】

演算論理装置の出力はその第 2 の入力に接続されたフィードバックループを有する請求項 8記載のシステム。

【請求項 15】

フィードバックループはそれに結合された遅延素子を有する請求項 14記載のシステム。

【請求項 16】

ラスタライズ装置は均質のクリップスペースで動作する請求項 1 または 8に記載のシステム。

【請求項 17】

ラスタライズ装置は均質な座標スペースにおいて動作し、ラスタライズ装置はそれぞれ W - 値を含んでいる複数の頂点により規定されるプリミティブを受信するのに適用され、W - 値に基づいてエリアを識別するように構成され、エリアはプリミティブに対応して描かれるディスプレイ部分の表示であり、W - 値は (x、y、z) 頂点をスケールするためのスケール係数である請求項 1 または 8 に記載のシステム。

【請求項 18】

(a) バッファから頂点データを受信するためにバッファに結合されるように構成され、オブジェクトスペースからスクリーンスペースへ頂点データを変換するために単一の半導体プラットフォーム上に位置されている変換手段と、

(b) 変換手段から受信される頂点データについてライティング演算を実行するために変換手段と同一の単一の半導体プラットフォーム上に位置されているライティング手段と、

(c) ライティング手段から受信される頂点データをレンダリングするために変換手段およびライティング手段と同一の単一の半導体プラットフォーム上に位置されているラスタライズ装置とを具備し、

(d) 変換手段とライティング手段の少なくとも 1 つは複数の論理装置により並列して多数の演算スレッドを実行するためのシーケンサ手段を含んでおり、ライティング手段は、

(a) 頂点データを受信するように構成されている複数の入力バッファと、

(b) 入力バッファの 1 つの出力に結合されている第 1 の入力と、入力バッファの 1 つの出力に結合されている第 2 の入力とを有する乗算論理装置と、

(c) 入力バッファの 1 つの出力に結合されている第 1 の入力と、乗算論理装置の出力に結合されている第 2 の入力とを有する演算論理装置と、

(d) 演算論理装置の出力に結合されている入力と、乗算論理装置の第 1 の入力に結合されている出力とを有する第 1 のレジスタ装置と、

(e) 演算論理装置の出力に結合されている入力と、乗算論理装置の第 1 の入力および第 2 の入力に結合されている出力とを有する第 2 のレジスタ装置と、

10

20

30

40

50

( f ) 演算論理装置の出力に結合されている第 1 の入力と、入力バッファの 1 つの出力に結合されている第 2 の入力と、乗算論理装置の第 1 の入力に結合されている出力とを有するライティング論理装置と、

( g ) 乗算論理装置の少なくとも 1 つの入力と、演算論理装置の出力に結合されているメモリとを含んでいるグラフィック処理用のグラフィックパイプラインシステム。

**【請求項 19】**

( a ) バッファから頂点データを受信するためにバッファに結合されるように構成され、オブジェクトスペースからスクリーンスペースへ頂点データを変換するために単一の半導体プラットフォーム上に位置されている変換手段と、

( b ) 変換手段から受信される頂点データについてライティング演算を実行するために変換手段と同一の単一の半導体プラットフォームに位置されているライティング手段と、

( c ) ライティング手段から受信される頂点データをレンダリングするために変換手段およびライティング手段と同一の単一の半導体プラットフォーム上に位置されているラスタライズ装置とを具備し、

( d ) 変換手段とライティング手段の少なくとも 1 つは複数の論理装置により並列して多数の演算スレッドを実行するためのシーケンサ手段を含んでおり、

変換手段は、

( a ) 頂点データを受信するように構成されている入力バッファと、

( b ) 入力バッファの出力に結合された第 1 の入力を有する乗算論理装置と、

( c ) 乗算論理装置の出力に結合された第 1 の入力を有する演算論理装置と、

( d ) 演算論理装置の出力に結合する入力を有するレジスタ装置と、

( e ) 逆数または逆平方根演算を実行するために、演算論理装置またはレジスタ装置の出力に結合された入力を含んでいる反転論理装置と、

( f ) 反転論理装置の出力と、乗算論理装置の第 2 の入力との間に結合され、スカラー頂点データをベクトル頂点データに変換するように構成されているコンバータモジュールと

、

( g ) 乗算論理装置と演算論理装置に結合されているメモリとを含んでいるグラフィック処理用のグラフィックパイプラインシステム。

**【発明の詳細な説明】**

**【技術分野】**

**【0001】**

本発明はグラフィックプロセッサ、特に変換、ライティング、ラスタライズモジュールを含んでいるグラフィックパイプラインシステムに関する。

**【背景技術】**

**【0002】**

3次元グラフィックは多数のアプリケーションの中心である。例えばコンピュータ支援設計(CAD)は、多くの産業で急激に成長しており、コンピュータ端末、カーソル、CRTおよびグラフィック端末が鉛筆および紙に置換えられ、コンピュータディスクおよびテープが画材保管場所に置換えられている。全てではないがほとんどのこれらの産業は3次元オブジェクトを操作し、表示することを非常に必要とする。このことによってコンピュータスクリーンまたは他のディスプレイ装置上に3次元オブジェクトをモデル化し、レンダリングし、表示する方法に対する関心および研究が広く普及している。しかしながら、3次元オブジェクトを現実的にレンダリングし表示することを必要とするコンピュータの計算量は非常に大きい状態であり、3次元オブジェクトの真の現実的な表示はハイエンドシステムに非常に限定される。しかしながら、3次元オブジェクトを迅速に現実的にレンダリングし表示できる廉価なシステムの必要性が非常に増加している。

**【0003】**

過去数年で大きな成長量が見られる1つの産業はコンピュータゲーム産業である。コンピュータゲームの現在の世代はこれまで増加した方法で3次元グラフィックへ移行してい

10

20

30

40

50

る。同時に、プレイ速度は益々速く駆動される。この組み合わせは比較的廉価なシステムで3次元グラフィックを急速にレンダリングする真の必要性をかきたてている。ゲームに加えて、この必要性はe-コマースアプリケーションによってもかきたてられ、これは増加したマルチメディアの能力を必要としている。

#### 【0004】

3次元グラフィックのレンダリングと表示は典型的に多数の計算とコンピュータ処理を含んでいる。例えば、3次元オブジェクトをレンダリングするため、レンダリングされるオブジェクトを規定する座標点または頂点のセットが形成されなければならない。頂点はレンダリングされ表示されるオブジェクトの表面を規定する多角形を形成するように接合されることができる。オブジェクトを規定する頂点が一度形成されると、頂点は基準の1つのオブジェクトまたはモデルフレームから基準のワールドフレームへ、最終的には平面の表示装置で表示されることができる2次元座標に変換されなければならない。このようにして、頂点は可視できるエリア外に入るので回転されスケールされ削除されまたはクリップされ、種々のライティング方式により照明され、色付けされてもよい。したがって3次元オブジェクトをレンダリングし表示するプロセスは、コンピュータ処理上集中的であり、多数の頂点を含む可能性がある。

10

#### 【0005】

このようなパイプライン化されたシステムを構成する一般的なシステムは従来技術として図1に示されている。このようなシステムでは、データソース10はプリミティブを規定する拡張された頂点ストリームを発生する。これらの頂点はパイプライン化されたグラフィックシステム12により記憶の目的で頂点メモリ13を経て一度に1つ通過される。拡張された頂点が一度頂点メモリ13からパイプライン化されたグラフィックシステム12へ受信されると、頂点は変換され、変換モジュール14とライティングモジュール16によりそれぞれライティングされ、さらにクリップされラスタライズ装置18によりレンダリングを設定され、したがって、レンダリングされたプリミティブを発生し、表示装置20で表示される。

20

#### 【0006】

動作中、変換モジュール14はスケーリング、回転、3次元頂点のセットをローカルまたはモデル座標から、レンダリングされたオブジェクトの表示に使用される2次元ウィンドウへの投影を行うために使用されてもよい。ライティングモジュール16は種々のライティング方式、光位置、周囲光レベル、材料等に基づいて頂点の色と外観を設定する。ラスタライズモジュール18は先に変換されおよび/または照明されている頂点をラスタライズまたはレンダリングする。ラスタライズモジュール18はオブジェクトをレンダリングターゲットに描写し、これはレンダリングされたデータを表示装置に移動するディスプレイ装置または中間ハードウェアまたはソフトウェア構造である。

30

#### 【発明の開示】

#### 【0007】

グラフィック処理システムを製造するとき、価格を最少にしながら、種々のグラフィック処理コンピュータポネットの速度を増加することが通常必要とされる。通常、システムの速度を増加するために集積がしばしば使用される。集積は1つの集積回路に異なる処理モジュールを含ませることを意味する。外部バスと反対に微細な半導体環境で通信するこのような処理モジュールにより、速度は非常に増加される。

40

#### 【0008】

しかしながら、集積は多数の処理モジュールを単一のチップに構成し製造する価格によりしばしば制限を受ける。グラフィック処理の領域では、速度を増加するための変換、ライティング、ラスタライズモジュールを集積する試みは価格を非常に高価にする。この価格の増加の理由は、必要な集積回路が単に高価になり過ぎる寸法になるので実行可能でないためである。

#### 【0009】

このサイズの増加は主に種々のエンジンの複雑性によるものである。高い性能の変換およびライティングエンジンは単独で非常に複雑であり、したがって任意の付加的な機能性

50

を有した構成にすることはいうまでもなく、チップ上に構成すること自体が高価である。さらに、通常のラスタ化装置はクリッピング、レンダリング等のタスクで多くの面をもち、このようなモジュールを変換およびライティングモジュールと結合するような価格の効率化の試みをほぼ不可能にする。

【 0 0 1 0 】

それ故、価格が効率的な集積を可能にする設計を有する変換、ライティング、ラスタ化モジュールが必要とされる。

【 0 0 1 1 】

グラフィックパイプラインシステムがグラフィック処理のために提供される。このようなシステムは頂点データを受信するための頂点属性バッファに結合されるように構成されている変換モジュールを含んでいる。変換モジュールはオブジェクトスペースからスクリーンスペースへ頂点データを変換する作用を行う。変換モジュールには変換モジュールから受信された頂点データについてライティング動作を行うための単一の半導体プラットフォーム上に位置されているライティングモジュールが結合されている。またライティングモジュールに結合され、ライティングモジュールから受信される頂点データをレンダリングするための単一の半導体プラットフォームに位置されているラスタ化装置も含まれている。

10

【 0 0 1 2 】

本発明の1つの特徴では、変換モジュールは頂点データを受信するため頂点属性バッファに結合されるように構成された入力バッファを含むことにより集積化を容易にするように設計されている。乗算論理装置は入力バッファの出力に結合される第1の入力を有する。また、乗算論理装置の出力に結合されている第1の入力を有する演算論理装置が設けられている。演算論理装置の出力はレジスタ装置の入力に結合されている。

20

【 0 0 1 3 】

反転論理装置も設けられており、逆数または逆平方根演算を実行するために演算論理装置の出力に結合された入力を含んでいる。さらに反転論理装置の出力と乗算論理装置の第2の入力の間に結合されている変換モジュールも含まれている。使用において、変換モジュールはスカラー頂点データをベクトル頂点データへ変換する機能を行う。

【 0 0 1 4 】

メモリは乗算論理装置と演算論理装置へ結合されている。メモリは頂点データを処理するため、入力バッファ、乗算論理装置、演算論理装置、レジスタ装置、反転論理装置、変換モジュールと関連して使用される複数の定数および変数を記憶している。最終的に、出力変換器は処理された頂点データを出力するためにライティングモジュールに結合されるため演算論理装置の出力に結合されている。

30

【 0 0 1 5 】

さらに集積を容易にするために、ライティングモジュールはそこから頂点データを受信するために変換システムに結合されるように構成された複数の入力バッファを含んでいる。入力バッファは第1の入力バッファと、第2の入力バッファと、第3の入力バッファとを含んでいる。第1の入力バッファと、第2の入力バッファと、第3の入力バッファとの入力は変換システムの出力に結合されている。

40

【 0 0 1 6 】

さらに、第1の入力バッファの出力に結合される第1の入力と、第2の入力バッファの出力に結合される第2の入力とを有する乗算論理装置が含まれている。演算論理装置は第2の入力バッファの出力に結合された第1の入力を有している。演算論理装置はさらに乗算論理装置の出力に結合される第2の入力を有する。演算論理装置の出力はライティングシステムの出力に結合されている。

【 0 0 1 7 】

次に演算論理装置の出力に結合されている入力と、演算論理装置の第1の入力に結合されている出力とを有する第1のレジスタ装置が設けられている。第2のレジスタ装置は演算論理装置の出力に結合されている入力を有する。また、このような第2のレジスタは乗

50

算論理装置の第1の入力と第2の入力に結合される出力を有する。ライティング論理装置もまた設けられており、演算論理装置の出力に結合された第1の入力と、第1の入力バッファの出力に結合された第2の入力と、乗算論理装置の第1の入力に結合された出力とを有している。

【0018】

変換モジュールに類似して、メモリは乗算論理装置の入力と演算論理装置の出力の少なくとも1つに結合される。メモリは頂点データを処理するために、入力バッファ、乗算論理装置、演算論理装置、第1のレジスタ装置、第2のレジスタ装置、ライティング論理装置と関連して使用される複数の定数および変数をそこに記憶している。

【0019】

共に、前述の変換/ライティングアーキテクチャはクリップのないラスタ化を行うために均質のクリップスペースで動作するラスタ化装置で動作してもよい。これは単一の半導体プラットフォーム上の全てのコンポーネントの配置を容易にする。均質のクリップスペースで動作するために、ラスタ化装置は隣接する設定モジュールからプリミティブを受信するときにプリミティブを規定するラインのライン方程式を決定する。その後、W-値はラインの交差点のライン方程式を使用して計算される。エリアはその後計算されたW-値に基づいて決定される。このようなエリアは描かれるディスプレイの一部分の表示である。エリア中のスペースはその後、その画素をレンダリングするためのライン方程式を使用して識別される。

【発明を実施するための最良の形態】

【0020】

本発明のこれらおよび別の利点は、以下の詳細な説明を読み、その図面の種々の図を検討することによって明らかになるであろう。上記およびその他の特徴および利点は、添付図面を参照とする以下の本発明の好ましい実施形態の詳細な説明からさらによく理解されるであろう。図1は、従来技術を示している。図1A乃至32Cは、本発明のグラフィックパイプラインシステムを示す。図1Aは、本発明の1実施形態の種々のコンポーネントを示すフロー図である。示されているように、本発明は、頂点属性バッファ(VAB)50、変換モジュール52、ライティングモジュール54、および設定モジュール57を備えたラスタ化モジュール56を含む4つの主要なモジュールに分けられる。1実施形態において、上記の各モジュールは、以下においてさらに詳細に説明するように単一の半導体プラットフォーム上に配置されている。この説明において、単一の半導体プラットフォームとはただ1つの単一の半導体ベースの集積回路またはチップのことである。

【0021】

VAB50は位置、垂線、カラー、テクスチャ座標のような複数の頂点属性状態を収集し、保持するために含まれている。完成された頂点は変換モジュール52によって処理され、その後ライティングモジュール52に送られる。変換モジュール52は、照明を行うライティングモジュール54に対してベクトル発生する。ライティングモジュール54の出力は、プリミティブを設定する設定モジュールに適したスクリーンスペースデータである。その後、ラスタ化モジュール56はプリミティブのラスタ化を行う。変換およびライティングモジュール52および54は、コマンドが一度スタートされると常に終了されるように、コマンドレベルでのみ機能停止することを認識しなければならない。

【0022】

1実施形態において、本発明は、オープン・グラフィック・ライブラリ(商標名オープンGL)およびD3D(商標名)変換およびライティングパイプラインを少なくとも部分的に使用するハードウェア構造を含んでいる。オープンGL(商標名)は2-Dおよび3-Dグラフィックイメージを既定するコンピュータ業界の標準アプリケーションプログラムインターフェース(API)である。このオープンGL(商標名)により、アプリケーションは任意のオペレーティングシステムにおいて任意のオープンGL(商標名)に従うグラフィックアダプタを使用して同じ効果を生成する。オープンGL(商標名)は、1組のコマンドまたは即時実行機能を特定する。各コマンドは描写アクションを指令するか、

10

20

30

40

50

あるいは特別な効果を生じさせる。

【 0 0 2 3 】

図 2 は、本発明の 1 実施形態による V A B 50 の概略図である。示されているように、V A B 50 はコマンドビット 200 を送り、一方において頂点の属性を表すデータビット 204 とモードビット 202 とを記憶する。使用において、V A B 50 は頂点のデータビット 204 を受取り、これを出力する。

【 0 0 2 4 】

V A B 50 は、複数の可能な頂点属性状態をデータビット 204 により受取り、記憶するように構成されている。使用において、このようなデータビット 204 すなわち頂点データが受取られ、V A B 50 中に記憶された後、頂点データは V A B 50 からグラフィック処理モジュール、すなわち変換モジュール 52 に出力される。さらに、図 2 A を参照としてさらに詳細に後述する他の処理に加えて、頂点データが V A B 50 に入力される方法を決定するコマンドビット 200 が V A B 50 によって送られる。このようなコマンドビット 200 はマイクロ制御装置、C P U、データソース、またはコマンドビット 200 を発生できる任意の他のタイプのソースのようなコマンドビットソースから受取られる。

【 0 0 2 5 】

さらに、モードビット 202 が送られ、このモードビット 202 は処理動作の複数のモードの状態を示す。したがって、モードビット 202 は、後続するグラフィック処理モジュール中で頂点データが処理される方法を決定するように適応されている。このようなモードビット 202 は、マイクロ制御装置、C P U、データソース、またはモードビット 202 を発生

【 0 0 2 6 】

V A B 50 に関連した種々の機能が専用ハードウェア、ソフトウェアまたは任意の他のタイプの論理装置によって制御されてもよい。種々の実施形態において、モードビット 202 の 6 4、1 2 8、2 5 6 または任意の他の数が使用されてもよい。

【 0 0 2 7 】

V A B 50 はまた 1 2 8 ビットフォーマットに変換される必要のある 6 4 ビットデータのための収集地点として機能する。V A B 50 の入力は 6 4 ビット / サイクルであり、その出力は 1 2 8 ビット / サイクルである。別の実施形態において、V A B 50 は 1 2 8 ビット

【 0 0 2 8 】

表 1

位置 :  $x, y, z, w$   
 拡散カラー :  $r, g, b, a$   
 反射カラー :  $r, g, b$   
 フォグ :  $f$   
 テクスチャ 0 :  $s, t, r, q$   
 テクスチャ 1 :  $s, t, r, q$   
 垂線 :  $n_x, n_y, n_z$   
 スキンウエイト :  $w$

動作中、V A B 50 は、 $x, y$  の書込み時に  $z, w$  対を  $(0.0, 1.0)$  にデフォルトすることが可能となるため、 $z, w$  データ対の前に  $x, y$  データ対が書込まれるものとして動作する。これはオープン G L ( 商標名 ) および D 3 D ( 商標名 ) におけるデフォルト成分にとって重要である可能性がある。位置、テクスチャ 0 およびテクスチャ 1 のスロットは第 3 および第 4 の成分を  $(0.0, 1.0)$  にデフォルトすることを認識しなければ



ならない。さらに、拡散カラスロットは第4の成分を(1.0)にデフォルトし、テクスチャスロットは第2の成分を(0.0)にデフォルトする。

【0029】

VAB50は、データビット204をアSEMBLするために使用されるさらに別のスロット205を含んでおり、このデータビット204は変換モジュール54に送られることができ、あるいはライティングモジュール54から受取られることができる。スロット205におけるデータビット204は浮動小数点または整数フォーマットであることができる。上述したように、各頂点のデータビット204は、データビット204の処理に影響を与えるモードを表すモードビット202の関連したセットを有している。これらのモードビット202は、以下さらに詳細に説明する理由のためにデータビット204により変換およびライティングモジュール52および54を通して送られる。

10

【0030】

1実施形態において、VAB50によって受取られる18の有効なVAB、変換およびライティングコマンドが存在することができる。図2Aは、本発明の1実施形態によるVAB50によって受取られることのできる種々のコマンドを示すチャートである。図2Aのチャートに示されている全てのロードおよび読出し文脈コマンドならびにパススルーコマンドは、128ビットまでの、またはその他のサイズの1つのデータワードを転送することを認識しなければならない。

【0031】

図2Aの各コマンドは、データビット204の各セットが1つのVABアドレスの高ダブルワードまたは低ダブルワード中に書込まれるか否かを指示する制御情報を含んでいる可能性がある。さらに、ワードレベルの制御を行う2ビットの書込みマスクが使用されてもよい。さらに、実行されるべき現在のコマンドに対するデータビット204の全てが存在していることを開始(launch)ビットがVAB制御装置に通知してもよい。

20

【0032】

各コマンドは関連した機能停止フィールドを有しており、このフィールドによってルックアップは、そのコマンドが文脈メモリを読出す読出しコマンドであるか、あるいは文脈メモリに書込む書込みコマンドであるかに関する情報を見出すことが可能になる。現在実行中のコマンドの機能停止フィールドを使用することによって、新しいコマンドは、矛盾の場合には待機させられ、あるいは進行することを可能にされることができる。

30

【0033】

動作において、VAB50はサイクル当たり128ビット(または他の任意のサイズ)までの1つの入力データワードを受取り、サイクル当たり128ビット(または他の任意のサイズ)までの1つのデータワードを出力することができる。ロードコマンドに対して、これは、データをVAB50中にロードして128ビットのクワド(quad)ワードを生成するのに2サイクル必要であり、それをドレインするために1サイクル必要であることを意味する。ライティングモジュール54内のスカラーメモリに対して、全クワドワードを累算する必要はなく、これらは1サイクル/アドレスでロードされることができる。1つの頂点に対して、7つのVABスロットをロードするのに14サイクルまで必要になる可能性があり、一方それらをドレインするには7サイクルあればよい。しかしながら、実行中の頂点コマンドを変更する頂点状態を更新するだけでよいことを認識すべきである。これは、ある場合には頂点位置が2サイクルで更新され、一方頂点データをドレインするのに7サイクルを要することを意味する。x, y位置の場合には、たった1サイクルあればよいことを認識しなければならない。

40

【0034】

図2Bは、グラフィック処理中に頂点属性をVAB50にロードし、VAB50からドレインする1つの方法を示すフローチャートである。最初に、動作210で、VAB50において頂点属性の少なくとも1つのセットが処理されるために受取られる。上述したように、頂点属性の各セットは特有で、単一の頂点に対応していることができる。

【0035】

50

使用において、頂点属性は動作212 でV A B 50が受取ったときにその中に記憶される。さらに、記憶された頂点属性の各セットは変換モジュール52の複数の入力バッファの対応した1つに転送される。受取られた頂点属性セットはまた、動作216 で示されているように、受取られた頂点属性が現在V A B 50中に記憶されている異なったセットの対応した頂点属性を有しているか否かを決定するために監視される。

【0036】

決定ブロック217 において記憶されている頂点属性が受取られた頂点属性に対応していると決定されたとき、動作218 に示されているように、記憶されている頂点属性は変換モジュール52の対応した入力バッファにばらばらな順序で出力される。記憶されている頂点属性が出力されるとすぐに、対応した入ってきた頂点属性がV A B 50中にその場所を占有してもよい。しかしながら、対応が全く見出されない場合、動作219 に示されているように、記憶されている頂点属性の各セットは規則的な予め定められたシーケンスにしたがって変換モジュール52の対応した入力バッファに転送されてもよい。

10

【0037】

記憶されている頂点属性は、それが関連した開始コマンドを有している場合には、上述した方式で転送されない可能性があることを注意すべきである。さらに、上記の方法が適切に行われるためにV A B 50の出力の帯域幅は少なくともV A B 50の入力の帯域幅でなければならない。

【0038】

図2Cは、図2Bの動作を実行するために使用される本発明のアーキテクチャを示す概略図である。示されているように、V A B 50は書込みデータ端子WDと、読出しデータ端子RDと、書込みアドレス端子WAと、および読出しアドレス端子RAとを有している。読出しデータ端子は第1のクロック制御されたバッファ230 に結合され、データビット204をV B A 50から出力する。

20

【0039】

第1のマルチプレクサ232もまた含まれており、これはV A B 50の読出しアドレス端子と第2のクロック制御されたバッファ234とに結合された出力を有している。第1のマルチプレクサ232の第1の入力はV A B 50の書込みアドレス端子に結合され、一方第1のマルチプレクサ232の第2の入力は第2のマルチプレクサ236の出力に結合されている。論理モジュール238は第1および第2のマルチプレクサ232および236と、V A B 50の書込みアドレス端子と、第2のクロック制御されたバッファ234の出力との間に結合されている。

30

【0040】

使用において、論理モジュール238は、入ってきた頂点属性が未決定でありV A B 50においてドレインするか否かを決定する作用を行う。1実施形態において、この決定は、頂点属性が未決定であるか否かを示すビットレジスタを監視することより容易に行われることができる。入ってきた頂点属性がこの時点でV A B 50中に一致したものを有していると決定された場合、論理モジュール238は、入ってきた頂点属性がすぐにその場所に記憶されるようにその一致した頂点属性をドレインするために第1のマルチプレクサ232を制御する。他方、入ってきた頂点属性がこの時点でV A B 50中に一致したものを有しないと決定された場合、論理モジュール238は、V A B 50がドレインされ、入ってきた頂点属性が、論理モジュール238によって更新される第2のマルチプレクサ236の入力によって順次または別のある予め定められた順序でロードされるように、第1のマルチプレクサ232を制御する。

40

【0041】

その結果、V A B 50は、新しく入ってきた頂点属性がロードされる前に、多数の頂点属性をドレインする必要がない。未決定の頂点属性は、可能ならば、対応したV A Bの対応したものの押し出し、それによってそれが進行することを可能にする。その結果、V A B 50は任意の順序でドレインすることができる。この能力がないと、V A B 50をドレインするのに7サイクルを要し、それをロードするのに、おそらく、さらに14サイクル要する。

50

ロードとドレインとを重複させることにより、さらに高い性能が得られる。これは、入力バッファが空であり、V A B 50が変換モジュール52の入力バッファ中にドレインできる場合にのみ可能であることを認識しなければならない。

【 0 0 4 2 】

図3は、本発明の1実施形態によるV A B 50に関連したモードビットを示している。変換/ライティングモード情報は、モードビット202によりレジスタ中に記憶される。モードビット202は、以下において明らかになるように、変換モジュール52およびライティングモジュール54のシーケンサを駆動するために使用される。各頂点は特有であることが可能である関連したモードビット202を有しており、したがって特有に構成されたプログラムシーケンスを実行することができる。モードビット202は一般にグラフィックA P Iに直接マップすることができるが、それらのあるものは導出されてもよい。

10

【 0 0 4 3 】

1実施形態において、図3のアクティブな光ビット(L I S)は隣接していてもよい。さらに、パススルービット(V P A S)は、オンにされたときに頂点データがスケールおよびバイアスと共に通過され、変換もライティングも行われまいという点で特有である。V P A Sが真(true)である場合に使用される可能なモードビット202はテクスチャ分割ビット(T D V 0, 1)およびフォグビット(商標名D 3 Dにおいてフォグ値を抽出するために使用される)である。したがって、V P A Sは予め変換されたデータに対して使用され、T D V 0, 1は商標名D 3 Dの文脈では円筒状ラップモードを処理するために使用される。

20

【 0 0 4 4 】

図4は、本発明の1実施形態の変換モジュールを示している。示されているように、変換モジュール52は6つの入力バッファ400によってV A B 50に接続されている。1実施形態において、各入力バッファ400はサイズが7 \* 1 2 8ビットである。6つの入力バッファ400はそれぞれ7つのクワドワードを記憶することができる。そのような各入力バッファ400は、パスデータが位置データと重複されていることを除いて、そのレイアウトがV A B 50と同じである。

【 0 0 4 5 】

1実施形態において、入力バッファ400がロードされた前の段階からデータが変化しているか否かを示すように各入力バッファ400の各属性に対して1つのビットが指定されていてもよい。この設計によって、各入力バッファ400は変化したデータに関してのみロードされてもよい。

30

【 0 0 4 6 】

変換モジュール52はさらに、ライティングモジュール54中の6つの出力頂点バッファ402に接続されている。出力バッファは第1のバッファ404と、第2のバッファ406と、および第3のバッファ408とを含んでいる。以下から明らかになるように、第3のバッファ408の内容、すなわち位置、テクスチャ座標データ等はライティングモジュール54では使用されない。しかしながら、第1のバッファ404および第2のバッファ406の両者は光線およびカラーデータをライティングモジュール54に入力するために使用される。ライティングモジュールは2つの読出し入力を処理するように構成されているため、2つのバッファが使用される。データは読出しコンフリクト等に関する問題を回避するように構成されていることを認識すべきである。

40

【 0 0 4 7 】

さらに、変換モジュール52には文脈メモリ410およびマイクロコードR O Mメモリ412が結合されている。変換モジュール52はオブジェクト空間頂点データをスクリーン空間に変換して、ライティングモジュール54が必要とするベクトルを生じさせる作用をする。変換モジュール52はまたスキニング(s k i n n i n g)およびテクスチャ座標を処理する。1実施形態において、変換モジュール52は平行して4つのフロートを処理する1 2 8ビット設計であってもよく、4項ドット積を行うために最適化されていてもよい。

【 0 0 4 8 】

50

図4Aは、変換モジュール52中で多数のスレッドを実行する本発明の1実施形態による方法を示すフローチャートである。動作において、変換モジュール52はインターリーブすることにより3つの頂点を平行して処理することができる。したがって、書込みおよびそれに後いて文脈メモリ410からの読出しのようなコマンドの間に機能停止状況が生じなければ、3つのコマンドが同時に平行して実行されることができる。3つの実行スレッドは互いに独立しており、任意コマンドであることができる。これは、全ての頂点が特有の対応したモードビット202を含んでいるためである。

【0049】

図4Aに示されているように、多数のスレッドを実行する方法は、動作420において実行されるべき現在のスレッドを決定することを含んでいる。この決定は、グラフィック処理モジュールがある動作の終了に必要なとするサイクルの数を識別して、そのサイクルを追跡することにより行われることができる。サイクルを追跡することによって、各スレッドはあるサイクルに割当てられ、それによって現在のサイクルに基づいて現在のスレッドを決定することができる。しかしながら、このような決定は、効果的であると思われる任意の所望の方式で行われてもよいことを認識しなければならない。

【0050】

次に動作422において、現在のサイクル中に実行されるべきスレッドに関連した命令が、対応したプログラムカウンタ番号を使用して検索される。その後、この命令は動作424においてグラフィック処理モジュールに関して実行される。

【0051】

1使用例において、この方法は、最初に、第1のプログラムカウンタによって第1の命令またはコードセグメントにアクセスすることを含む。上述のように、このようなプログラムカウンタは第1の実行スレッドと関連している。次に、第1のコードセグメントがグラフィック処理モジュールにおいて実行される。すぐに明らかになるように、このようなグラフィック処理モジュールは加算器、乗算器または任意の他の機能装置あるいはその組合せの形態をとることができる。

【0052】

グラフィック処理モジュールは実行を完了するために2以上のクロックサイクルを必要とするため、第1のコードセグメントの実行後1クロックサイクル経過してすぐに第2のコードセグメントが第2のプログラムカウンタによってアクセスされてもよい。第2のプログラムカウンタは第2の実行スレッドと関連しており、各実行スレッドが特有の頂点を処理する。

【0053】

その後、第2のコードセグメントは、グラフィック処理モジュール中での第1のコードセグメントの実行の終了前に、グラフィック処理モジュールにおいて実行を開始してもよい。使用において、グラフィック処理モジュールは出力を発生するために全てのスレッドのそれぞれに対して予め定められた数のサイクルを必要とする。したがって、全ての予め定められた数のサイクルのそれぞれに対してこの例の種々のステップが反復されてもよい。

【0054】

この技術は、従来技術より優れた多くの利点を提供する。もちろん、本発明の機能装置はさらに効率的に使用される。さらに、多数スレッド方式が使用されると仮定された場合、統御コードがより効率的に書込まれることができる。

【0055】

たとえば、答えを出力するのに3クロックサイクルを必要とする乗算器がグラフィック処理モジュールに含まれている場合、 $a = b * c$ および $d = e * a$ のような後続する動作の間に2つの無動作コマンドを含むことが必要となる。その理由は、3クロックサイクル後まで“a”が利用できないためである。しかしながら、この実施形態では、コードが $a = b * c$ の直後に $d = e * a$ を呼出すだけでもよい。それは、このようなコードは3クロックサイクルごとに1回呼出される3つの実行スレッドの1つとして実行されると仮定さ

10

20

30

40

50

れることができるからである。

【 0 0 5 6 】

図 4 B は、図 4 A の方法が行われる手順を示すフロー図である。示されているように、各実行スレッドは、命令メモリ 452 において命令またはコードセグメントにアクセスするために使用される関連したプログラムカウンタ 450 を有している。その後、このような命令は加算器 456、乗算器 454 および、または反転論理装置またはレジスタ 459 のようなグラフィック処理モジュールを動作するために使用されてもよい。

【 0 0 5 7 】

上記の処理モジュールの 2 以上のものがタンデム方式で使用される状況に適応させるために、グラフィック処理モジュール間において 1 以上のコードセグメント遅延素子 457 が使用される。3 スレッドフレームワークが使用される場合、3 クロックサイクルコードセグメント遅延素子 457 が使用される。1 実施形態において、加算命令が乗算命令に後続した場合に、コードセグメント遅延素子 457 が使用される。このような場合、乗算器 456 が出力を発生するために十分な時間が確実に経過するように、乗算命令の実行後 3 クロックサイクル経過するまで加算命令は実行されない。

【 0 0 5 8 】

各命令の実行後、現在の実行スレッドのプログラムカウンタ 450 が更新され、次の実行スレッドのプログラムカウンタが関連した命令にアクセスするためにラウンドロビンシーケンスでモジュール 458 により呼出される。プログラムカウンタは、インクリメント、ジャンプ、呼出および復帰、テーブルジャンプ、および、またはディスパッチを含む任意の方法で使用されることができ、それに限定されないことを認識しなければならない。ディスパッチとは、受取られたパラメータに基づいてコードセグメント実行の開始地点を決定することである。さらに、この多数スレッド実行フレームワークに関連した原理は、本発明のグラフィック処理パイプラインのライティングモジュール 54 にも適用可能であることを認識することが重要である。

【 0 0 5 9 】

3 スレッドフレームワークが使用される場合、任意のある時間に各スレッドが 1 つの入力バッファおよび 1 つの出力バッファに割当てられる。これによって、3 つのコマンドを処理しながら、さらに 3 つのコマンドをデータと共にロードすることが可能になる。入力バッファおよび出力バッファは、以下において図 2 7 および 2 8 を参照として説明する方式によりラウンドロビンシーケンスで割当てられる。

【 0 0 6 0 】

したがって、実行スレッドは時間的および機能的にインターリーブされる。これは、各機能装置が 3 つのステージにパイプラインされ、各スレッドがいつでも 1 つのステージを占有していることを意味する。1 実施形態において、3 つのスレッドは常に同じシーケンスで実行するように、すなわち 0、1、3 に設定されてもよい。概念上、スレッドは  $t =$  クロックモジュール 3 において機能装置に入力される。機能装置が動作し始めると、結果を出力するのに 3 サイクルを要し (6 サイクルを必要とする I L U を除いて)、その時同じスレッドは再びアクティブである。

【 0 0 6 1 】

図 5 は、本発明の 1 実施形態による図 4 の変換モジュール 52 の機能装置を示している。示されているように、頂点データを受取るために V A B 50 に結合するように構成された入力バッファ 400 が含まれている。

【 0 0 6 2 】

メモリ論理装置 ( M L U ) 500 は、入力バッファ 400 の出力に結合された第 1 の入力を有している。オプションとして、M L U 500 の出力は、その第 1 の入力に結合されたフィードバックループ 502 を有していてもよい。

【 0 0 6 3 】

演算論理装置 ( A L U ) 504 もまた設けられており、この A L U の第 1 の入力は M L U 500 の出力に結合されている。A L U 504 の出力はさらに、その第 2 の入力に接続された

10

20

30

40

50

フィードバックループ506を有している。このようなフィードバックループ502はさらに、それに結合された遅延素子508を有していてもよい。ALU504の出力には、レジスタ装置510の入力が結合されている。レジスタ装置510の出力は、MLU500の第1および第2の入力に結合されていることを認識しなければならない。

【0064】

反転論理装置(ILU)512が設けられており、このILU512は、逆数または逆平方根演算を行うためにALU504の出力に結合された入力を含んでいる。別の実施形態において、ILU512はレジスタ装置510の出力に結合された入力を含んでいてもよい。

【0065】

さらに、変換またはスメアリング(smearing)モジュール514が含まれており、このモジュール514はILU512の出力とMLU500の第2の入力との間に結合されている。使用において、この変換モジュール514はスカラー頂点データをベクトル頂点データに変換するように機能する。これはスカラーデータをベクトルと乗算して、乗算器および、または加算器が処理するベクトル演算子にすることによって行なわれる。たとえば、スカラーAは、変換後、ベクトル(A, A, A, A)になってもよい。別の実施形態では、スメアリングモジュール514はMLU500と関連したマルチプレクサまたは本発明の任意の他のコンポーネント中に含まれていてもよい。オプションとして、レジスタ516はILU512の出力と変換装置514の入力との間に結合されていてもよい。さらに、このようなレジスタ516はスレッド(thread)されてもよい。

10

【0066】

メモリ410は、MLU500の第2の入力とALU504の出力とに結合されている。とくに、メモリ410はMLU500の第2の入力に結合された読出し端子を有している。さらに、メモリ410はALU504の出力に結合された書込み端子を有している。

20

【0067】

メモリ410は、頂点データを処理するために入力バッファ400、MLU500、ALU504、レジスタ装置510、ILU512および変換モジュール514と共に使用されるために複数の定数および変数が記憶されている。このような処理には、オブジェクト空間頂点データをスクリーン空間頂点データに変換し、ベクトルを発生すること等が含まれる。

【0068】

最後に、出力コンバータ518はALU504の出力に結合されている。出力コンバータ518は、処理された頂点データがこれに出力されるように出力バッファを介してライティングモジュール54に結合されている。ILUを除く全てのデータ通路は128ビット幅であるように設計されてもよく、あるいは別のデータ通路幅が使用されてもよい。

30

【0069】

図6は、本発明の1実施形態による図5の変換モジュール52のMLU500の概略図である。示されているように、変換モジュール52のMLU500は、並列に結合された4つの乗算器600を含んでいる。

【0070】

変換モジュール52のMLU500は、3つの異なった方式で2つの4成分ベクトルを乗算するか、あるいは1つの4成分ベクトルをパス(pass)することができる。MLU500は、多重演算を行うことができる。表2は、変換モジュール52のMLU500に関連したこのような演算を示している。

40

【0071】

表2

## 【数 1】

CMLU\_MULT  $o[0] = a[0]*b[0], o[1] = a[1]*b[1], o[2] = a[2]*b[2], o[3] = a[3]*b[3]$   
 CMLU\_MULA  $o[0] = a[0]*b[0], o[1] = a[1]*b[1], o[2] = a[2]*b[2], o[3] = a[3]$   
 CMLU\_MULB  $o[0] = a[0]*b[0], o[1] = a[1]*b[1], o[2] = a[2]*b[2], o[3] = b[3]$   
 CMLU\_PASA  $o[0] = a[0], o[1] = a[1], o[2] = a[2], o[3] = a[3]$   
 CMLU\_PASB  $o[0] = b[0], o[1] = b[1], o[2] = b[2], o[3] = b[3]$

10

## 【0072】

表 3 には、可能な A および B 入力が見されている。

## 【0073】

表 3

MA M	MLU
MA V	入力バッファ
MA R	RLU (MB R と共有された)
MB I	ILU
MB C	文脈メモリ
MB R	RLU (MA R と共有された)

20

表 4 は、クロス乗積に対して使用されることのできる回転オプションを示している。

## 【0074】

表 4

MR NONE	変更なし
MR ALBR	A[XYZ]ベクトルを左に、B[XYZ]ベクトルを右に回転する
MR ARBL	A[XYZ]ベクトルを右に、B[XYZ]ベクトルを左に回転する。

## 【0075】

図 7 は、本発明の 1 実施形態による図 5 の変換モジュール 52 の ALU 504 の概略図である。示されているように、変換モジュール 52 の ALU 504 は、並列 / 直列に結合された 3 つの加算器 700 を含んでいる。使用において、変換モジュール 52 の ALU 504 は 2 つの 3 成分ベクトルを加算し、1 つの 4 成分ベクトルをパスし、あるいはベクトル成分を出力を横切ってスメア (smear) することができる。表 5 は、変換モジュール 52 の ALU 504 が行うことのできる種々の演算を示している。

30

## 【0076】

表 5

## 【数 2】

CALU\_ADDA  $o[0] = a[0]+b[0], o[1] = a[1]+b[1], o[2] = a[2]+b[2], o[3] = a[3]$   
 CALU\_ADDB  $o[0] = a[0]+b[0], o[1] = a[1]+b[1], o[2] = a[2]+b[2], o[3] = b[3]$   
 CALU\_SUM3B  $o[0123] = b[0] + b[1] + b[2]$   
 CALU\_SUM4B  $o[0123] = b[0] + b[1] + b[2] + b[3]$   
 CALU\_SMRB0  $o[0123] = b[0]$   
 CALU\_SMRB1  $o[0123] = b[1]$   
 CALU\_SMRB2  $o[0123] = b[2]$   
 CALU\_SMRB3  $o[0123] = b[3]$   
 CALU\_PASA  $o[0] = a[0], o[1] = a[1], o[2] = a[2], o[3] = a[3]$   
 CALU\_PASB  $o[0] = b[0], o[1] = b[1], o[2] = b[2], o[3] = b[3]$

40

## 【0077】

50

表 6 は、変換モジュール52の A L U 504 の A および B 入力を示している。

【 0 0 7 8 】

表 6

A A A	A L U ( 1 つの命令遅延 )
A A C	文脈メモリ
A B M	M L U

無変更、Bの否定、Aの否定を行なうことによりAおよびB入力の符号ビットを修正することもまた可能であり、ここでA、Bは絶対値である。A L U 504 がスカラー頂点データを出力した場合、このスカラー頂点データは、各出力がスカラー頂点データを表しているという意味で出力を横切ってスメアされていることを認識しなければならない。M L U 500 および A L U 504 のパス制御信号のそれぞれが演算中全ての特殊値処理をディスエーブルすることができる。

10

【 0 0 7 9 】

図 8 は、本発明の 1 実施形態による図 5 の変換モジュール52のベクトルレジスタファイル510 の概略図である。示されているように、ベクトルレジスタファイル510 は 4 組のレジスタ800 を含んでおり、各レジスタ800 は対応したマルチプレクサ802 の第 1 の入力に結合された出力と、対応したマルチプレクサ802 の第 2 の入力に結合された入力とを有している。

【 0 0 8 0 】

本発明の 1 実施形態において、ベクトルレジスタファイル510 はスレッドされている。すなわち、ベクトルレジスタファイル510 の 3 つのコピーが存在し、各スレッドがそれ自身のコピーを有している。1 実施形態では、各コピーは 8 つのレジスタを含んでおり、その各レジスタはサイズが 1 2 8 ビットであり、4 つのフロートを記憶することができる。ベクトルレジスタファイル510 は A L U 504 から書込まれ、その出力は M L U 500 にフィードバックされる。ベクトルレジスタファイル510 はサイクル当り 1 回の書込みおよび 1 回の読出しを行なう。

20

【 0 0 8 1 】

動作において、各レジスタコンポーネントへの書込み動作を個々にマスクすることもできる。ベクトルレジスタファイル510 は、書込みアドレスが読出しアドレスと同じである場合、入力から出力へのバイパス路511 によってゼロレイテンシーを示す。この場合、マスクされていないコンポーネントはレジスタから取出され、マスクされたコンポーネントはバイパスされる。このように、ベクトルレジスタファイル510 はコンポーネント単位でベクトルを生成し、あるいは A L U S M R 演算 ( 表 5 参照 ) と共にベクトル成分の順序を変更することに対して非常に有用である。一時的な結果はまたベクトルレジスタファイル510 中に記憶されることができる。

30

【 0 0 8 2 】

図 9 は、本発明の 1 実施形態による図 5 の変換モジュール52の I L U 512 の概略図である。示されているように、変換モジュール52の I L U 512 は浮動小数点の逆数 ( 1 / D ) および逆平方根 ( 1 / D ^ ( 1 / 2 ) ) を発生することができる。このような演算を行なうために、2 つの反復処理のいずれか一方が小数部に関して実行されてもよい。このような処理は任意の所望の専用ハードウェアにより実行されてもよく、以下に示されている：

40

逆数 ( 1 / D )	逆平方根 ( 1 / D ^ ( 1 / 2 ) )
$x_{n+1} = x_n ( 2 - x_n * D )$	$x_{n+1} = ( 1 / 2 ) * x_n ( 3 - x_n^2 * D )$
( 1 ) $x_n$ ( 速度 ) に対する表検索	$x_n$ ( 速度 ) に対する表検索
$x_n$	$x_n * x_n$
( 2 ) 第 1 回目の反復 : 乗算 - 加算	第 1 回目の反復 : 乗算 - 加算
$2 - x_n * D$	$3 - x_n^2 * D$
( 3 ) 第 1 回目の反復 : 乗算	第 1 回目の反復 : 乗算
$x_n ( 2 - x_n * D )$	$( 1 / 2 ) * x_n ( 3 - x_n^2 * D )$
( 4 ) 第 2 回目の反復 : 演算なし	第 2 回目の反復 : 2 乗

50



$x_n + 1$  をパス  
 (5) 第2回目の反復：乗算 - 加算  
 $2 - x_{n+1} * D$   
 (6) 第2回目の反復：乗算  
 $x_{n+1} (2 - x_{n+1} * D)$   
 D)

$x_{n+1}^2$   
 第2回目の反復：乗算 - 加算  
 $3 - x_{n+1}^2 * D$   
 第2回目の反復：乗算  
 $(1/2) * x_{n+1} (3 - x_{n+1}^2 * D)$

示されているように、2つの処理は類似しており、簡単な設計を行なっても差しつかえない。この反復は、しきい値精度が満足されるまで繰り返されることを認識しなければならない。

【0083】

10

動作において、ILU512は逆数演算および逆平方根演算を含む2つの基本的な演算を行なう。他の装置とは異なり、それは出力を発生するために6サイクルを必要とする。その入力はスカラーであり、したがって出力もそうである。前述したように、ILU512の出力におけるスレッド保持レジスタ516は、有効な結果が発生される次の回まで結果をラッチするように当てにされている。さらに、スカラー出力は、MLU500に供給される前にベクトルにスミアされる。反転装置512は、約22小数部ビット範囲内までの正確なIEEE(米国電気電子技術者協会)出力を発生するために検索表および2つのパスNewton-Raphson反復を使用する。表7は、変換モジュール52のILU512によって行なわれることのできる種々の演算を示している。

【0084】

20

表7

CILU INV	$o = 1.0 / a$ CILU
ISQ	$o = 1.0 / \text{sqrt}(a)$
CILU CINV	$o = 1.0 / a$ (レンジクランプにより)
CILU NOP	出力なし

表7の上述したレンジクランプ反転演算は、クリッピング演算がラスタ化モジュール56により処理されることを可能にするために使用されてもよい。座標はスクリーン空間に直接変換され、これは、均質のクリップスペースがほぼ0.0である場合に問題を結果的に生じさせる可能性が高い。各除算において $1.0 / 0.0$ による乗算を回避するために、 $1/w$ 計算が最小および最大ベキ指数にクランプされる。

30

【0085】

使用において、図5に示されている文脈メモリ410は、クワドワードだけを使用して読出しおよび書込みを行なう。このメモリはMLU500またはALU504によって各サイクルごとに読出されることができ、ALU504によって書込まれることができる。メモリ読出しはサイクル当たり1度だけ可能である。読出しが必要である場合には、それは命令の開始時に行なわれ、それから3サイクル後にALU504にパイプラインされる。文脈メモリ410は必ずしもスレッドされなくてよい。

【0086】

図10は、本発明の1実施形態による図5の変換モジュール52の出力コンバータ518の出力アドレスのチャートである。出力コンバータ518は出力を適切な目的地に導き、データのビット精度を変更し、性能を増加させるためにあるデータ攪拌(swizzling)を行なうことができる。ライティングモジュール54に送られる予定である全てのデータは、S1E8M13として編成された22ビット浮動小数点フォーマット(1符号、8ベキ指数、13小数部ビット)に丸められる。ライティングモジュール54における図4に示されているような目的地パツファ402はスレッドされる。

40

【0087】

データ攪拌は、ベクトルを発生しているときに有用である。このような技術により、ベクトルを生成する場合に損失を生じずに距離ベクトル(1, d, d \* d)を発生することが可能となる。距離ベクトルはフォグ、地点パラメータおよび照明減衰に対して使用される。これは、アイベクトルおよび照明方向ベクトルにより行なわれる。表8は、このよう

50

なベクトルに関連した種々の演算を示している。以下の表において、ベクトルを2乗するとは  $d^2 = d \circ t [(x, y, z), (x, y, z)]$  である  $d^2$  を  $(x, y, z)$  の  $w$  コンポーネント中に記憶することを指していることを認識しなければならない。

【0088】

表 8

(1) ベクトルを2乗する  $(x, y, z, d * d)$  ( $d * d$  を  $V B U F$  に出力し、1.0 を  $V B U F$  に出力する)

(2)  $d * d$  の逆平方根を発生する  $(1 / d)$

(3) ベクトルを正規化する  $(x / d, y / d, z / d, d)$  ( $x / d, y / d, z / d$  を  $W B U F$  に出力し、 $d$  を  $V B U F$  に出力する)

10

本発明において行なわれた数学的計算は常に  $I E E E$  方式に従ったものである必要はないことを認識しなければならない。たとえば、任意の数により乗算された“0”は“0”をレンダリングすると仮定されることができ。これは、 $d = 0$  である  $d = d^2 * 1 / (d^2)^{1/2}$  のような式を処理する場合にとくに有用である。上記の仮定を行わないと、このような式はエラーを生じ、したがって関連した計算を行なうときに問題が発生する。

【0089】

図11は、本発明の1実施形態による図5の変換モジュール52のマイクロコード編成を示す図である。変換モジュールのマイクロコードは、44ビットの総帯域幅を形成する15のフィールドに構成されてもよい。フィールドは、装置のデータフローを一致させるために遅延されてもよい。M L U 500 の演算はゼロの遅延で実行される。A L U 演算は1の遅延で実行され、R L U の出力演算は2の遅延で実行される。各遅延は3サイクルと等価である。

20

【0090】

図12は、本発明の1実施形態による図5の変換モジュール52のシーケンサ1200の概略図である。図12に示されているように、変換モジュール52のシーケンサ1200は、処理動作の複数のモードの状態を示すモードビットを  $V A B 50$  から受取るように構成されたバッファ1202を含んでいる。

【0091】

メモリ412 もまた含まれており、このメモリ412 は、モードの状態にしたがって処理動作を行なうようにそれぞれ構成されたコードセグメントを記憶することができる。シーケンシングモジュール1206はメモリ412 と制御ベクトルモジュール1205との間に結合されており、この制御ベクトルモジュール1205はバッファ1202に結合され、モードビット202 から得られた制御ベクトルに基づいてメモリ412 中の複数のアドレスを識別する。シーケンシングモジュール1206はさらに、データを出力バッファ1207に転送するように変換モジュール52を動作するために使用されることのできるコードセグメントを検索するためにメモリ412 中のアドレスにアクセスするように構成されている。

30

【0092】

図13は、図12の変換モジュール52のシーケンサ1200の使用に関連した種々の動作を詳細に示すフローチャートである。示されているように、シーケンサ1200は、変換またはライティング動作におけるグラフィック処理をシーケンス化するように構成されている。動作1320において、処理動作の複数のモードの状態を示すモードビット202 が最初に受取られる。1実施形態において、モードビット202 はソフトウェア駆動装置から受取られてもよい。

40

【0093】

その後、動作1322において、メモリ中の複数のアドレスがモードビット202 に基づいて識別される。その後、動作1324において、そのモードの状態にしたがって処理動作を行なうようにそれぞれ構成されたコードセグメントを検索するために、メモリ中のこのようなアドレスがアクセスされる。続いて、動作1326に示されているように、頂点データを処理するために変換またはライティングモジュールによりコードセグメントが実行される。

【0094】

50

図14は、図12の変換モジュール52のシーケンサ1200のシーケンシングモジュール1206の動作を詳細に示すフロー図である。示されているように、複数のモードレジスタ1430はそれぞれ、単一の頂点に対応するモードビット202の特有のセットを含んでいる。モードレジスタ1430は、図4Aおよび4Bを参照として上述した方式での多数の実行スレッドの実行を可能にするためにラウンドロビンシーケンスでポーリングされることを認識すべきである。

【0095】

現在の実行スレッドが選択されると、モードビット202の対応したグループは動作1432でデコードされる。動作1432においてモードビット202がデコードされると、対応した頂点データを処理する特定のコードセグメントがROM1404においてアクセスされたか否かをそれぞれ示す複数のビットを含む制御ベクトルが供給される。

10

【0096】

コードセグメントがROM1404でアクセスされ実行されるべきであるか否かを決定するとき、ポインタ動作1436は現在のスレッドポインタをインクリメントして、次の実行スレッドを開始し、それによって類似の動作を継続するように第2のグループモードビット202を獲得する。これはラウンドロビンシーケンスで各スレッドに対して継続される。

【0097】

制御ベクトルが一度、モードビット202の特定のグループに対して形成されると、優先度エンコーダ動作1438は次の“1”またはエネーブルされた制御ベクトルのビットを決定し、識別する。このようなビットが発見されると、優先度エンコーダ動作1438は実行のために、制御ベクトルのエネーブルビットに対応するアドレスをROM1404中に生成する。

20

【0098】

残りのスレッドを処理した後、およびモードビットがデコードされ、制御ベクトルが再度有効になった後、モードビット202の最初のグループに戻るとき、マスキング動作1434は先の“1”または前に識別されたエネーブルされたビットをマスクするために使用される。これはマスク動作1434後に全ての残りのビットの解析を可能にする。

【0099】

前述のプロセスは以下の表を使用して示されている。表9はサブジェクト頂点データについて実行される複数の式を示している。

【0100】

30

表9

【数3】

$$\begin{aligned}
 R &= (a \quad \quad \quad) \\
 R &= (a + \quad d * e \quad) \\
 R &= (a + b * c + \quad f) \\
 R &= (a + b * c + d * e \quad) \\
 R &= 1.0 / (a \quad \quad \quad) \\
 R &= 1.0 / (a + \quad d * e \quad) \\
 R &= 1.0 / (a + b * c + \quad f) \\
 R &= 1.0 / (a + b * c + d * e \quad)
 \end{aligned}$$

40

【0101】

示されているように、反転演算に加えて加算される積には4つの可能性が存在する(a, b \* c, d \* e, fおよび1 / x)。次に、モードフィールドが規定される。表10はモードフィールドの対、mode.yとmode.zを示し、それぞれ表9の演算の予め定められたセットに割当てられている。

【0102】

50

表 1 0

【数 4】

```

mode.y[4] 0: R = a
           1: R = a + d*e
           2: R = a + b*c + f
           3: R = a + b*c + d*e

mode.z[2] 0: R = R
           1: R = 1.0/R

```

10

【0 1 0 3】

その後、各演算は関連するアドレスと共にメモリに位置付けされる。表 1 1 は関連する演算をそれぞれ有する複数のメモリアドレスを示している。また制御ベクトル定義のセットも示されている。

【0 1 0 4】

表 1 1

【数 5】

```

ROM[0]: R = a
ROM[1]: R = R + b*c
ROM[2]: R = R + d*e
ROM[3]: R = R + f
ROM[4]: R = 1.0/R

```

```

cv[0] = 1;
cv[1] = (mode.y==2 || mode.y==3) ? 1 : 0;
cv[2] = (mode.y==1 || mode.y==3) ? 1 : 0;
cv[3] = (mode.y==2) ? 1 : 0;
cv[4] = (mode.z==1) ? 1 : 0;

```

20

30

【0 1 0 5】

表 1 2 は 1 例の実行を示している。

【0 1 0 6】

表 1 2

R = a + d \* e は以下に対応する：

```

mode . y = 1 ;
mode . z = 0 ;

```

40

これは以下の制御ベクトルを与える： cv [ 0 ] = 1 ;

```

cv [ 1 ] = 0 ;
cv [ 2 ] = 1 ;
cv [ 3 ] = 0 ;
cv [ 4 ] = 0 ;

```

実行

第 1 のサイクル：

cv [ 0 ] は TRUE であるので、ROM [ 0 ] を実行

制御ベクトルにさらに多くの TRUE 値が存在するので、プログラムを終了 しない

## 第2のサイクル：

c v [ 1 ] は F A L S E であるので、観察し続ける

c v [ 2 ] は T R U E であるので、R O M [ 2 ] を実行

制御ベクトルには T R U E 値がもはや存在しないので、プログラムを終了する。

## 【 0 1 0 7 】

このようにして、変換モジュール52のシーケンサ1200はスレッドされたモードビット202から得られるスレッドされた制御ベクトルをステップし、対応する制御ベクトルビットが“ T R U E ”に設定されるあらゆるR O Mアドレスを実行する。制御ベクトルはR O Mと同一の長さを有する。シーケンサ1200は1つの“ 1 ”のレート、または予め定められたサイクル数毎にエネーブルされたビットで任意の制御ベクトルをステップできる。モードビット202を使用しないコマンドはその簡潔性のためにオンザフライマイクロコードにより実行される。

10

## 【 0 1 0 8 】

このような状態をモードビット202の特有のストリングにより表示することによって、種々の動作の状態を決定するためにグラフィック処理ハードウェアの複数のイフ - ゼン ( i f - t h e n ) 節を実行することは必要ではない。改良された性能はそれによって与えられる。概念的に、これはプログラム言語のイフ節がシーケンサ1200へ移動するかのようであり、シーケンサ1200はモードビット202により示されるように“ F A L S E ”状態を即時に命令をスキップする。

## 【 0 1 0 9 】

20

前述したように、コードセグメントはR O Mに記憶され、これはモードビットにより識別される動作の種々の状態を処理することができる。1実施形態では、別々のコードセグメントはモードビットにより示される各動作を処理するために検索される。その代りとして、1つの包括性コードセグメントは可能であるそれぞれまたは幾つかの動作の組合わせを処理するために書込まれてもよい。しかしながら、各動作の組合わせでこのような大きいコードセグメントを生成することは付加的なコードスペースを必要とし、それ故、普通に使用される動作の組合わせだけでコードセグメントをモジュール化することが有効であることに注意する。

## 【 0 1 1 0 】

モードビット202は一度頂点が実行を開始すると変化しないので、制御ベクトルの生成はシーケンサに入る前に1つの頂点毎に1度実行されさえすればよい。しかしながら、これについての例外が動作が反復されるライティングのような幾つかのケースで生じる。最後の頂点命令が発見されるとき、シーケンス信号の終了 ( E O S ) が表明される。これは入力および出力バッファの状態を変更し、図 2 8 A と 2 8 B を参照して説明した方法で次のコマンドの開始を可能にするために使用される。E O S 信号は命令が処理される方法と類似の目的地バッファを解除するために遅延されるパイプラインであることに注意する。図 4 B を参照する。

30

## 【 0 1 1 1 】

図 1 4 A はグラフィック処理中のスカラーおよびベクトル頂点データの管理を一体化するために使用される本発明の種々の機能コンポーネントを示したフロー図である。示されているように、1つの機能アспект1440はベクトル頂点データを処理モジュール、即ち加算器、乗算器等へ入力し、ベクトル頂点データを出力することを含んでいる。別の機能アспект1442では、ベクトル頂点データはベクトル処理モジュール、即ち加算器、乗算器等により処理され、これは再度ベクトル頂点データへ変換されるかスミアされるスカラー頂点データを出力する。

40

## 【 0 1 1 2 】

さらに別の機能アспект1444では、ベクトル頂点データはマスクされ、それによってスカラー頂点データに変換され、その後、これはベクトル頂点データを生成する目的で、メモリ、即ちレジスタ論理装置中に記憶される。さらに別の機能アспект1446では、スカラー頂点データはベクトル処理モジュール、即ち加算器、乗算器等により抽出され、こ

50

れはスカラー処理モジュール、即ち反転論理装置により処理され、スカラー頂点データをレンダリングする。このスカラー頂点データは再度ベクトル頂点データに変換される。

【 0 1 1 3 】

図 1 4 B は図 5 の変換モジュール52に対応している図 1 4 A に示されている本発明の機能コンポーネントの1つの可能な組合せ1451を示すフロー図である。機能アスペクト1444および1446は図 4 B を参照して前述した方法と類似の方法で関連する遅延を有することに注意すべきである。図 1 4 C は図 1 4 A に示されている本発明の機能コンポーネントの別の可能な組合せ1453を示すフロー図である。

【 0 1 1 4 】

マルチプレクサは図 1 4 A - 1 4 C の機能モジュール中のベクトル頂点データからスカラー頂点データを抽出する。このようなマルチプレクサは種々の機能モジュールによる処理前に必要とされる任意のデータのスイズリングに対しても応答可能である。1実施形態では、マルチプレクサはベクトルの頂点データを通し回転することができ、他の処理用の A L U 等の他のグラフィック処理モジュールに依存する。さらに別の実施形態では、マルチプレクサはペナルティなしで独立して属性を任意選択的に再配置することができる。

【 0 1 1 5 】

図 1 4 D は特定用途向け集積回路 ( A S I C ) のようなハードウェア構造によりグラフィックパイプラインにおけるグラフィック処理中に変換システムがブレンディングまたはスキン動作を行うように構成されている方法を示している。パイプラインでの処理中に、動作1470では、複数のマトリックス、それぞれ1つのマトリックスに対応する複数の加重値および頂点データが受信される。付加的なマトリックスのセットは正規の頂点データが必要とされる可能性があることに注意すべきである。

【 0 1 1 6 】

続いて、動作1472では、複数の積の和がその後計算され、各積は頂点データと、1つのマトリックスと、そのマトリックスに対応する加重との乗算により計算される。このような積の和はその後、さらに処理を行うために動作1474で出力される。

【 0 1 1 7 】

要約すると、以下の積の和が計算される。

【 0 1 1 8 】

式 # 1

$$i = 1 \dots x \text{ に対して } v' = w_i * M_i * v$$

ここで  $v$  = 入力された頂点データ、

$w$  = 加重値、

$M$  = マトリックス、

$x$  = マトリックスの数、

$v'$  = 処理されるモジュールへ出力される頂点データ

式 # 2

$$i = 1 \dots x \text{ に対して } n' = w_i * I_i * n$$

ここで  $n$  = 入力された頂点データ ( 正規ベクトル )、

$w$  = 加重値、

$I$  = 反転マトリックス ( 逆転置マトリックス )、

$x$  = 反転マトリックスの数、

$n'$  = 処理モジュールへ出力される頂点データ ( 正規ベクトル )

式 # 3

$$V_s = [ O_x, O_y, O_z, ]' + 1 / ( v''_{we} ) * [ ( v''_x ), ( v''_y ), ( v''_z ), 1 ]$$

ここで  $v'' = C * v'$ 、

$v'$  = 式 # 1 からの積の和、

$C = [ S_x, S_y, S_z, 1 ]'$

10

20

30

40

50

\* P P = 投影マトリックス、  
 $v_s$  = 表示目的のスクリーンベクトル、  
 O = ビューポートオフセット、  
 S = ビューポートスケール

前述した加重  $w_i$  を表す方法が多数存在することに注意すべきである。例えば式 # 1 と # 2 では、 $i = 1 \dots (x - 1)$  では  $w_x$  ( $w_i$ 、ここでは  $i = x$ ) は式 1 -  $w_i$  により計算されることが言われている。このようにして加重  $w_i$  を表すことにより、全ての加重  $w$  が 1 に合計されることが確実にされる。

【 0 1 1 9 】

1 実施形態では、マトリックスはモデルビューマトリックス (M) を含み、積の和 ( $v'$ ) はライティング動作によりさらに処理されるために出力される (式 1 参照)。この積の和 ( $v'$ ) はまた合成マトリックス (C) の使用によって表示目的で別の積の和 ( $v_s$ ) を生成するためにも使用される (式 3 参照)。マトリックスは反転マトリックス (I) を含み、頂点データは正規ベクトルデータ (n) を含む。このようなケースでは、付加的な処理はライティング動作を含む (式 # 2 参照)。

【 0 1 2 0 】

図 1 5 は本発明の 1 実施形態によるライティングモジュール 54 の概略図である。示されているように、ライティングモジュール 54 は変換モジュール 52 が頂点データを出力するバッファ 402 を含んでいる。示されているように、バッファ 408 は通路 1501 によりライティングモジュール 54 をバイパスする。さらにライティングモジュール 54 には文脈メモリ 1500 とマイクロコード ROM メモリ 1502 に結合されている。

【 0 1 2 1 】

ライティングモジュール 54 はフォッグおよびポイントパラメータに加えてライティングを処理するように構成されている。使用において、ライティングモジュール 54 はバッファバイパス経路 1501 を制御し、拡散、ポイントサイズ、スペキュラー出力色およびフォッグ値を計算する。ライティングモジュール 54 は変換モジュール 52 と同一のモードビット 202 を使用することに注意すべきである。

【 0 1 2 2 】

ライティングモジュール 54 はさらに変換モジュール 52 に関してそれ程正確性を必要とせず、それ故、3 ワードで組織される 22 ビット浮動小数点値 (1.8.13 フォーマット) を処理する。第 3 のバッファ 408 のデータは 128 ビットであるので、これはライティングモジュール 54 周辺のバイパス経路 1501 を使用する。ライティングモジュール 54 は事象駆動され、同時に図 4 A と 4 B を参照して前述した変換モジュール 52 と類似の方法で 3 つのスレッドを実行する。ライティングモジュール 54 は外部ソースからコマンド発信許可を必要とすることに注意しなければならない。

【 0 1 2 3 】

図 1 6 は本発明の 1 実施形態による図 1 5 のライティングモジュール 54 の機能装置を示す概略図である。示されているように、変換システムに結合されてそこから頂点データを受信するように構成されている入力バッファ 402 が含まれている。前述したように、入力バッファ 402 は第 1 の入力バッファ 404、第 2 の入力 406、第 3 の入力バッファ 408 を含んでいる。第 1 のバッファの入力 404、第 2 の入力バッファ 406、第 3 の入力バッファ 408 の入力は変換モジュール 52 の出力に結合されている。バイパスの目的で、第 3 の入力バッファ 408 の出力は遅延素子 1608 によりライティングモジュール 54 の出力に結合されている。

【 0 1 2 4 】

さらに、第 1 の入力バッファ 404 の出力に結合されている第 1 の入力と、第 2 の入力バッファ 406 の出力に結合されている第 2 の入力を有する M L U 1610 が含まれている。M L U 1610 の出力はその第 2 の入力に結合されているフィードバックループ 1612 を有する。演算論理装置 (A L U) 1614 は第 2 の入力バッファ 406 の出力に結合されている第 1 の入力を有する。A L U 1614 はさらに M L U 1610 の出力に結合されている第 2 の入力を有する。

10

20

30

40

50

A L U 1614の出力はライティングモジュール54の出力に結合されている。A L U 1614の出力と第3の入力バッファ408の出力はマルチプレクサ1616によりライティングモジュール54の出力に結合されていることに注意すべきである。

【 0 1 2 5 】

次に、A L U 1614の出力に結合されている入力と、A L U 1614の第1の入力に結合されている出力とを有する第1のレジスタ装置1618が設けられている。第2のレジスタ装置1620はA L U 1614の出力に結合されている入力を有する。またこのような第2のレジスタ1620はM L U 1610の第1の入力と第2の入力に結合されている出力を有する。

【 0 1 2 6 】

ライティング論理装置 ( L L U ) 1622もまた設けられ、A L U 1614の出力に結合されている第1の入力と、第1の入力バッファ404の出力に結合されている第2の入力と、M L U 1610の第1の入力に結合されている出力とを有する。L L U 1622の第2の入力は遅延素子1624により第1の入力バッファ404の出力に結合されていることに注意すべきである。さらに、L L U 1622の出力は先入れ先出しレジスタ装置1626を介してM L U 1610の第1の入力に結合されている。図16に示されているように、L L U 1622の出力はまた変換モジュール1628によりM L U 1610の第1の入力にも結合されている。動作において、このような変換モジュール1628は変換モジュール52と類似の方法でスカラー頂点データをベクトル頂点データへ変換するように構成されている。

10

【 0 1 2 7 】

最後に、メモリ1500はM L U 1610の入力と演算論理装置1614の出力の少なくとも一方に結合されている。特に、メモリ1500はM L U 1610の第1および第2の入力に結合されている読取り端子を有する。さらにメモリ1500はA L U 1614の出力に結合されている書込み端子を有する。

20

【 0 1 2 8 】

メモリは頂点データを処理するため、入力バッファ402、M L U 1610、A L U 1614、第1のレジスタ装置1618、第2のレジスタ装置1620、L L U 1622と共に使用される複数の定数および変数を記憶している。

【 0 1 2 9 】

図17は本発明の1実施形態による図16のライティングモジュール54のM L U 1610の概略図である。示されているように、ライティングモジュール54のM L U 1610は並列している3つの乗算器1700を含んでいる。動作において、本発明のM L U 1610は2対3コンポーネントベクトルを乗算し、または1対3コンポーネントベクトルを通過するように構成されている。3コンポーネントベクトルの乗算はドット積または並列乗算により行われる。表13はライティングモジュール54のM L U 1610が実行できる動作を示している。

30

【 0 1 3 0 】

表 1 3

【 数 6 】

ZMLU_MULT	$o[0] = a[0]*b[0], o[1] = a[1]*b[1], o[2] = a[2]*b[2]$	40
ZMLU_PASA	$o[0] = a[0], o[1] = a[1], o[2] = a[2]$	
ZMLU_PASB	$o[0] = b[0], o[1] = b[1], o[2] = b[2]$	

【 0 1 3 1 】

表14はライティングモジュール54のM L U 1610の可能なAおよびB入力を示している。

表 1 4

M A V	V B U F F E R
M A L	L L U

50



M A R            R L U [ 2 , 3 ] ( M B R と共有 )  
 M A C            コンテキストメモリ ( M B C と共有 )  
 M B M            M L U  
 M B W            W B U F F E R  
 M B R            R L U [ 2 , 3 ] ( M A R と共有 )  
 M B C            コンテキストメモリ ( M A C と共有 )

図 1 8 は本発明の 1 実施形態による図 1 6 のライティングモジュール54の A L U 1614の概略図である。示されているように、A L U 1614は並列 / 直列の 3 つの加算器1800を含んでいる。使用において、A L U 1614は 2 対 3 コンポーネントベクトルを加算し、または 1 対 3 コンポーネントベクトルを通過するように構成されている。表 1 5 はライティングモジュール54の A L U 1614が実行できる種々の動作を示している。

10

【 0 1 3 2 】

表 1 5

【 数 7 】

ZALU\_ADD       $o[0] = a[0]+b[0], o[1] = a[1]+b[1], o[2] = a[2]+b[2]$   
 ZALU\_SUM3B     $o[012] = b[0] + b[1] + b[2]$   
 ZALU\_PASA      $o[0] = a[0], o[1] = a[1], o[2] = a[2]$   
 ZALU\_PASB      $o[0] = b[0], o[1] = b[1], o[2] = b[2]$

20

【 0 1 3 3 】

表 1 6 はライティングモジュール54の A L U 1614の可能な A および B 入力を示している。

。

【 0 1 3 4 】

表 1 6

A A W            W B U F F E  
 A A R            R L U [ 0 , 1 ]  
 A B M            M L U

30

図 1 9 は本発明の 1 実施形態による図 1 6 のライティングモジュール54のレジスタ装置 1618と1620の概略図である。示されているように、レジスタ装置1618と1620はそれぞれ 2 セットのレジスタ1900を含んでおり、レジスタ1900はそれぞれ対応するマルチプレクサ1902の第 1 の入力に接続されている出力と、マルチプレクサ1902の第 2 の入力に結合されている入力とを有する。

【 0 1 3 5 】

ライティングモジュール54のレジスタ装置1618と1620は A L U 1614の 2 つのレジスタと、M L U 1610の 2 つのレジスタに分離される。1 実施形態ではこれらのレジスタはスレッドされている。レジスタ装置1618と1620は書込みアドレスが読取りアドレスと同一であるとき、入力から出力へのバイパス通路のためにゼロの待ち時間を有する。

40

【 0 1 3 6 】

図 2 0 は本発明の 1 実施形態による図 1 6 のライティングモジュール54の L L U 1622の概略図である。L L U 1622はライティングモジュール54のライティング装置である。スカラーブロックは後に光 + マテリアルカラーを乗算するために使用されるライティング係数を計算する。L L U 1622は 2 つの M A C と、インバータと、4 つの小さいメモリとフラグレジスタを含んでいる。

【 0 1 3 7 】

フラグレジスタはライティング方程式の条件付き部分を実行するために使用される。出力は環境、拡散、スペキュラー係数である。スカラーメモリはスペキュラー近似に使用される変数と定数を含んでいる。各メモリの第 1 の位置は ( c t x 0 と c t x 2 では ) 1.0 および (

50

ctx1とctx3では) 0.0 を含んでいる。1 実施形態ではこれらはハードワイヤで結線され、ロードされる必要はない。

【 0 1 3 8 】

使用において、L L U 1622は機能的に式 ( x + L ) / ( M \* x + N ) を実行する。この式はスペキュラーライティング項を近似するために使用される。L L U 1622への入力はライティングモジュール54のA L U 1614からであり、ライティング方程式で使用されるドット積である。図 1 6 に関して前述したように、L L U 1622とM L U 1610との間に出力F I F O 1626が存在し、これはM L U 1610が係数を必要とするまで、係数をバッファする。1 実施形態ではこのようなF I F O 1626は遅延素子1608および1624、レジスタ1618および1620と共にスレッドされる。可能なカラーのマテリアル処理により、拡散およびスペキュラー出力がM L U 1610により消費されるときはわからない。

10

【 0 1 3 9 】

ライティングモジュール54はR , G , B コンポーネントのみを処理するので、拡散出力アルファコンポーネントを処理するための特別に構成されたハードウェアが存在する。このような特別に構成されたハードウェアは2つのタイプのアルファコンポーネント、即ちv t x カラー [ T b u f f e r ] および記憶されたc t x [ C t x s t o r e ] を出力できる。先のアルファコンポーネント間の選択はモードビット202 により支配される。

【 0 1 4 0 】

動作において、L L U 1622はライティングの周囲 ( C a ) 、拡散 ( C d e ) 、スペキュラー ( C s ) 係数を計算する。これらの係数は頂点のカラーに対する光の影響を生成するため周囲、拡散、スペキュラーカラーと乗算される。表 1 6 A はL L U 1622により受信された入力のリストと、ライティングの環境 ( C a ) 、拡散 ( C d e ) 、スペキュラー ( C s ) 係数を生成するために実行される計算を含んでいる。任意の所望のハードウェア構成はL L U 1622の構成に使用されることに注意する。1 実施形態では、図 2 0 で示されている特別な構成が使用される。

20

【 0 1 4 1 】

表 1 6 A

入力規定 :

- n = 正規ベクトル ( 変換エンジンから )
- e = 正規化されたアイベクトル ( 変換エンジンから )
- l = 正規化された光線ベクトル ( 変換エンジンから )
- s = スポットライトベクトル \* 光線ベクトル ( 変換エンジンから )
- D = 距離ベクトル ( 1 , d , d \* d ) ( 変換エンジンから )
- h = 半角ベクトル ( 変換エンジンから )
- K = 減衰定数ベクトル ( K 0 , K 1 , K 2 ) ( 変換エンジンから )
- L L U はその計算を実行するため以下のスカラーデータを受信する。

30

【 0 1 4 2 】

- n \* 1 ( M L U / A L U から )
- \* h ( M L U / A L U から )
- K \* D ( M L U / A L U から )
- s ( 変換エンジンから )
- パワー 0 ( ctx0-3メモリからのマテリアル指数 )
- パワー 1 ( ctx0-3メモリからのスポットライト指数 )
- 距離 ( ctx0-3メモリから )
- カットオフ ( ctx0-3メモリから )

40

無限大光

L L U 計算 :

- C a = 1 . 0
- C d = n \* l
- C s = ( n \* h ) ^ power0

50

## ローカル光

L L U 計算 :

$$\begin{aligned} a t t &= 1 . 0 / ( K * D ) \\ C a &= a t t \\ C d &= a t t * ( n * l ) \\ C s &= a t t * ( ( n * h ) ^ { p o w e r 0 } ) \end{aligned}$$

## スポットライト

L L U 計算 :

$$\begin{aligned} a t t &= ( s ^ { p o w e r 1 } ) / ( K * D ) \\ C a &= a t t \quad C d = a t t * ( n * l ) \\ C s &= a t t * ( ( n * h ) ^ { p o w e r 0 } ) \end{aligned}$$

10

前述したように、頂点シーケンサを制御するモードビットは頂点データ自体または頂点データから得られた結果により必ずしも変更されない。頂点データが頂点処理を変更することを可能にするため、L L U 1622は与えられたフラグレジスタ1623を使用する。ビットをこのフラグレジスタでT R U Eに設定することにより、フラグが計算の出力制御で特定されるならば、計算結果の0 . 0にクランプすることが可能である。フラグレジスタ1623の別の使用はレジスタ書込みのための書込みマスクを設定することである。

## 【 0 1 4 3 】

フラグレジスタ1623は性能のペナルティがなくライティング方程式で0 . 0ハイフ/ゼン/エルスクランピングを行うためにL L U 1622中に設けられる。種々のオペランドの符号ビットはフラグを設定する。表1 6 Bはフラグレジスタ1623のフラグが設定される方法と結果的なクランピングを示している。

20

## 【 0 1 4 4 】

表 1 6 B

## 無限光

L L U 計算 :

$$\begin{aligned} D f l a g &= ( n * l ) \text{ のサインビット} \\ S f l a g &= ( n * h ) \text{ のサインビット} \end{aligned}$$

クランプ :

$$\begin{aligned} C a &= ( \quad ) ? 0 : C a ; \\ C d &= ( D f l a g \quad ) ? 0 : C d ; \\ C x &= ( D f l a g | S f l a g ) ? 0 : C s ; \end{aligned}$$

30

## 局部光

L L U 計算 :

$$\begin{aligned} R f l a g &= ( r a n g e - d ) \text{ のサインビット} \\ D f l a g &= ( n * l ) \text{ のサインビット} \\ S f l a g &= ( n * h ) \text{ のサインビット} \end{aligned}$$

クランプ :

$$\begin{aligned} C a &= ( R f l a g \quad ) ? 0 : C a ; \\ C d &= ( R f l a g | D f l a g \quad ) ? 0 : C d ; \\ C x &= ( R f l a g | D f l a g | S f l a g ) ? 0 : C s ; \end{aligned}$$

40

## スポットライト

L L U 計算 :

$$\begin{aligned} C f l a g &= ( s - c u t o f f ) \text{ のサインビット} \\ R f l a g &= ( r a n g e - d ) \text{ のサインビット} \\ D f l a g &= ( n * l ) \text{ のサインビット} \\ S f l a g &= ( n * h ) \text{ のサインビット} \end{aligned}$$

クランプ :

$$\begin{aligned} C a &= ( C f l a g | R f l a g \quad ) ? 0 : C a ; \\ C d &= ( C f l a g | R f l a g | D f l a g \quad ) ? 0 : C d ; \end{aligned}$$

50

Cx = ( C f l a g | R f l a g | D f l a g | S f l a g ) ? 0 : C s ;

図21は本発明の1実施形態による図16のライティングモジュールに関連したフラグレジスタ1623の組織を示している。フラグレジスタ1623は8つの1ビットフラグを含み、ALU(IFLAG)またはMAC0(MFLAG)出力の符号ビットにより設定される。

【0145】

LLU1622が3ワードにスミアされる場合MLU1610へスカラ値を出力するとき、フラグレジスタのマスクを特定する。レジスタとマスクが真であるならば、0.0は出力を置換える。表17は出力された環境、拡散、スペキュラー属性で使用される図21の種々のフラグを示している。

10

【0146】

表17

周囲マスク：	C, R, U
拡散マスク：	D, C, R, U
スペキュラーマスク：	D, S, C, R, T, U

スペキュラー項で使用される近似は実際のcos(theta)\*\*が0.0になる場合、負になる。結果として、クランピング動作を実行する必要がある。このため、T, Uフラグが使用される。表18はLLU1622の機能論理装置(FLU)1621が行うことができる種々の動作を示している。図20に注意する。

20

【0147】

表18

ZFLU INV	o = 1 / a (仮数の正確度 - 12ビット)
ZFLU ISQ	o = 1 / sqrt(a) (仮数の正確度 - 6ビット)
ZFLU PASS	o = a
ZFLU PASS 1	o = 1.0
ZFLU MIN 1	o = (a < 1.0) ? a : 1.0
ZFLU NOP	o = 0.0

図22は本発明の1実施形態による図16のライティングモジュール54に関連したマイクロコードフィールドを示す図である。示されているように、ライティングモジュール54のマイクロコードは全体幅が85ビットである33フィールドに配置されている。フィールドは装置のデータ流を整合するように遅延される。MLU動作は遅延ゼロで行われ、ALU動作は遅延1で行われ、RLU、LLU出力動作は遅延2で行われる。各遅延は3サイクルに等しい。

30

【0148】

図23は本発明の1実施形態による図16のライティングモジュール54に関連したシーケンサ2300の概略図である。示されているように、ライティングモジュール54のシーケンサ2300はプロセス動作の複数のモードの状態を示すモードビット202を受信するように構成されている入力バッファ2302を含んでいる。また、それぞれモードの状態にしたがってプロセス動作を実行するように構成されているコードセグメントを記憶できるメモリ1502も含まれている。

40

【0149】

シーケンスモジュール2306はモードビットから得られる制御ベクトル2305に基づいてメモリ1502中の複数のアドレスを識別するためメモリ1502とバッファ2302との間に結合されている。シーケンスモジュール2306はさらに、ライティングモジュール54を動作するために使用されるコードセグメントを検索するためにメモリ1502中のアドレスをアクセスするように構成されている。

【0150】

ライティングモジュール54のシーケンサ2300は変換モジュール52のシーケンサと類似している。動作において、ライティングモジュール54のシーケンサ2300はスレッドされたモードビット202から得られるスレッドされた制御ベクトルによりステップし、それぞれの

50

ROMアドレスを実行し、その対応する制御ベクトルビットは“1”に設定される。制御ベクトルはROMが有するワードと同数のビットを有する。シーケンサ2300はスレッド毎に予め定められた数のサイクルで1つの“1”またはエネーブルビットのレートで任意の制御ベクトルをステップできる。モードビット202を使用しないコマンドはオンザフライマイクロコード発生により実行される。ライティングモジュール54のシーケンサ2300と変換モジュール52のシーケンサ1200との主な違いは、ライティングモジュール54のシーケンサ2300はループバックし8回までライティングコードを実行できることである。

【0151】

ライティングモジュール54のシーケンサ2300はそれぞれ新しい頂点ではゼロで開始し、マイクロコードシーケンスの終了時では1だけインクリメントする光カウンタを有する。モードビット202のLISフィールドが一致するビットフィールドで“1”を含んでいるならば、シーケンサ2300は戻り、ライティングマイクロコードブロックの開始時でスタートする。これはゼロがLISフィールドで発見されるか、8つの光が行われるまで継続する。カラーの累算は拡散およびスペキュラーカラーを記憶するALUレジスタを(1光線毎に)インクリメントすることによって行われる。自動メモリアドレスのインデックスは各光線で正確なパラメータをフェッチするために光カウンタを使用して実行される。

【0152】

図24は本発明の1実施形態にしたがって変換モジュール52およびライティングモジュール54のシーケンサが関連したバッファの入力および出力を制御することができる方法について詳細に説明するフローチャートである。示されているように、頂点データは動作2420でバッファの第1のセットの1つのバッファで最初に受信される。頂点データが受信されるバッファはラウンドロビンシーケンスに基づいている。

【0153】

続いて、動作2422では、バッファの第2のセットのエンプティバッファもまたラウンドロビンシーケンスに基づいて識別される。変換モジュール52は第1のセットのバッファと、第2のセットのバッファとの間に結合されている。第2のセットのバッファのエンプティバッファが識別されるとき、頂点データは変換モジュールで処理され、変換モジュールから第2のセットのバッファの識別されたエンプティバッファへ出力される。動作ステップ2424および2426を参照。

【0154】

同様に、バッファの第3のセットのエンプティバッファまたはメモリ中のスロット或いはスペースは動作2428でラウンドロビンシーケンスに基づいて識別される。ライティングモジュール54はバッファの第2のセットと第3のセットの間に結合されている。バッファの第3のセットのエンプティバッファが識別されるとき、頂点データは動作2430で示されているようにライティングモジュールで処理される。頂点データはしたがってライティングモジュール52からバッファの第3のセットの識別されたエンプティバッファへ出力される。動作2432を参照。バッファまたはメモリ中のスロットの数はフレキシブルであり、変更されてもよいことに注意すべきである。

【0155】

図25は図24の方法にしたがって変換モジュール52およびライティングモジュール54のシーケンサが関連したバッファの入力および出力を制御することができる方法の説明図である。示されているように、第1のセットのバッファまたは入力バッファ400は変換モジュール52に出力を供給し、変換モジュール52は第2のセットのバッファまたは中間バッファ404、406に出力を与える。第2のセットのバッファ404、406はメモリ2550へ出力(ドレイン)するライティングモジュール54に出力を与える。

【0156】

図25で説明されている方法を実行するため、メモリ2550のスロットと、第1および第2のセットのバッファはそれぞれ頂点データを最初に受信したときに特有の識別子をそれぞれ割当てられる。さらに、各バッファの現在の状態は追跡される。このような状態は割当てられた状態、有効な状態、アクティブ状態または行われた状態を含んでいる。

## 【 0 1 5 7 】

割当てられた状態は、バッファ/スロットが先のグラフィック処理モジュール、即ち変換モジュールまたはライティングモジュールの出力を受信するように既に割当てられていることを示している。書込みポイントがラウンドロビンシーケンスでバッファ/スロットを走査しているとき、割当てられた状態のバッファ/スロットはこのような書込みポイントを次のバッファまたはスロットにインクリメントさせる。

## 【 0 1 5 8 】

バッファ/スロットが有効な状態であるならば、そのバッファ/スロットは頂点データを受信するために使用される。他方で、アクティブ状態はバッファ/スロットが現在、実行状態であるかまたは頂点データを受信していることを示す。このアクティブ状態はスレッドが完了するまで維持され、その後読取りポイントをインクリメントし、したがってバッファ/スロットを有効状態に戻す。第1のセットのバッファ400はそれらを割当てるグラフィック処理モジュールが先に存在しないので、単に有効状態であることだけができることに注意する。

## 【 0 1 5 9 】

状態のシーケンスの1例を説明する。第1のセットのバッファ400と新しいコマンドビットのセット200の一方で頂点データを受信するとき、このようなバッファは有効状態に置かれ、その後バッファ402、404の第2のセットの1つが変換モジュール52の出力の予測において割当てられた状態に置かれる。

## 【 0 1 6 0 】

バッファ404、406の第2のセットが割当に使用可能ではないならば、第1のセットのバッファ400中の頂点データは処理されることができない。さらに実行されるコードセグメントが同時に行われる他のコードセグメントと干渉するか否かを決定するためのチェックが行われる。干渉するならば、第1のセットのバッファ400の頂点データは処理されずストール（機能停止）状態が開始される。

## 【 0 1 6 1 】

第2のセットのバッファ404、406の1つが割当状態に置かれた後、第1のセットのバッファ400はアクティブ状態に置かれる。変換モジュール52が実行を終了したとき、第2のセットのバッファ404、406は読取られ、その後有効状態に置かれる。これらの状態の変化は第2のセット404、406とメモリ2550のスロット間の頂点データの転送中も同様に行われる。

## 【 0 1 6 2 】

図25Bは設定モジュール57とトラバーサルモジュール58とを含むラスタ化モジュール56を示している。ラスタ化モジュール56は代わりの方法でエリアベースのラスタ化を実行するように構成されている。特に、複数の多角形を規定するセンスポイントがプリミティブに、またはその近くに位置され、その後一次方程式がプリミティブ中に存在する画素を決定するためにそのポイントにおいて評価される。動作中、この評価はポイントが効率的な目的で代わりの方法で移動されるときに反復される。さらに、ラスタ化モジュール56は何等クリッピングプロセスなしで動作するように構成される。

## 【 0 1 6 3 】

図26はラスタ化モジュール56の設定モジュール57の概略図である。示されているように、設定モジュール57は所望の浮動小数点計算を実行するためにデータと制御信号をそれらの適切な機能装置へ導く処理をする制御セクション61を含んでいる。プリミティブシーケンス62は頂点のシーケンスを三角形、直線または点に変える処理をする。さらに浮動小数点データパスセクション64は設定装置で必要とされる数学を実行するマルチプレクサおよび浮動小数点計算装置を含んでいる。

## 【 0 1 6 4 】

図26の参照を続けると、ラスタ化装置は整数値でのみ動作するので、出力フォーマットセクション63はエッジスロープとエッジ値の内部浮動小数点フォーマットをラスタ化装置に適している整数のフォーマットに変換する処理をする。勿論、別の実施形態では

10

20

30

40

50

、ラスタライズ装置は浮動小数点を使用し、したがって出力フォーマットセクション63の必要性をなくすることができる。

【0165】

動作において、出力フォーマットセクション63はブロック浮動小数点変換を実行する。よく知られているように、所定の数、即ち  $2.34 \times 10^1$  では、浮動小数点フォーマットは仮数 (2.34) とその指数 (10) を追跡する。ブロック浮動小数点変換は基本的に指数が同一であるように、入来するデータの仮数の小数点位置を操作する。このため、指数はラスタライズモジュール56で処理される必要はない。

【0166】

図26Aは図25Bのラスタライズモジュール56の設定モジュール57によって計算される種々のパラメータを示している。このようなパラメータは関連する機能を実行するためにラスタライズモジュール56に必要とされる。プリミティブ2600を受信するとき、設定モジュール57はプリミティブ2600のスロープ2601、スタート位置2602、スタート値2604を含む3つの値を計算する。

10

【0167】

スロープ2601はラスタライズ中に使用されるプリミティブ2600のエッジの一次方程式の係数を生成するために使用される。スロープ2601は例えば以下示す式#4および#5を使用することにより計算される。

【0168】

式#4および#5

$$\text{スロープ}_A = y_0 - y_1$$

$$\text{スロープ}_B = x_1 - x_0$$

ここで  $y_0$ 、 $y_1$  および  $x_0$ 、 $x_1$  は図26Aで示されている頂点の座標である。

20

【0169】

スロープはまた1つの回転動作等を使用することによって頂点の座標を使用して計算されることに注意する。

【0170】

スタート位置2602はさらに以下詳細に説明するようにエリアラスタライズのスタート点を示している。スタート値2604は図26Aで示されている陰影を付けられた三角形の面積に等しく、またエリアベースのラスタライズプロセス中にも使用される。このようなスタート値2604はスクリーンについてのラスタライズ位置をステップするように選択され、各ステップでスロープを付加することはラスタライズ位置がエッジにあるとき丁度ゼロに等しい。スタート値2604の計算は以下の式#6を使用して実現される。

30

【0171】

#6

$$\text{starting value} = \text{スロープ}_A * (x_s - x_0) + \text{スロープ}_B * (y_s - y_0)$$

ここで、 $x_s$ 、 $y_s$  = スタート位置2602、

スロープ<sub>A</sub>、スロープ<sub>B</sub> = 図26Aで示されている座標に基づいた1つのエッジのスロープ、

$x_0$ 、 $y_0$  = 図26Aで示されているエッジの頂点の1つの座標

40

前述の値はまた他のタイプのプリミティブに対して計算されることを理解すべきである。例えば、直線の場合、余分のスロープは4つの側面の境界のあるボックスで計算されなければならない。このようなスロープは境界のあるボックスの反対側のスロープの逆数を取ることにより容易に計算されることができる。余分のスロープの計算に加えて、別のスタート値が直線のプリミティブの場合に計算されることを必要とすることに注意すべきである。

【0172】

図27はラスタライズモジュール56が例えば三角形等の複数のプリミティブのうちの1つを処理する方法を示している。特に、最初の動作は最初にラスタライズ装置のモジュール56の設定モジュール57により実行される。プリミティブを受信するとき、一次方程式の一次

50

方程式係数は当業者によく知られた方法で図 2 6 A のスロープ 2601 を使用して動作 2700 でプリミティブを規定する直線で決定される。よく知られているように、3 つの一次方程式が三角形を規定するのに必要とされる。他方で、直線のようなプリミティブは 4 つの側面と 4 つの一次方程式により長方形または平行四辺形として描かれる。

【 0 1 7 3 】

その後、動作 2702 では、任意のプリミティブ頂点が負の W - 座標を有するならば、その一次方程式係数は変更される。このプロセスに関する付加的な情報を図 3 2 を参照してさらに詳細に説明する。

【 0 1 7 4 】

ラスタ化モジュール 56 の設定モジュール 57 もまたプリミティブの境界のあるボックスを計算することに注意しなければならない。ほとんどの三角形では、境界を有するボックスは 3 つの頂点の最小値および最大値を含んでいる。直線では、境界を有するボックスの 4 つの平行四辺形のコーナーが計算される。負の W - 座標の頂点を有する三角形または直線では、描かれるエリアは頂点の凸閉の殻を超えて延在する。

【 0 1 7 5 】

Open GL (商標名) のコマンドの 1 つは描かれない境界外を規定するシザー長方形である。ラスタ化モジュール 56 の設定モジュール 57 は境界のあるボックスとシザー長方形との交差点を計算する。シザー長方形は長方形であるので、4 つの付加的な一次方程式が与えられる。シザー長方形に関連する一次方程式は平凡な形状、即ち水平または垂直を有することに注意する。

【 0 1 7 6 】

さらに、3 - D スペースでは、近距離の平面と遠距離の平面とは平行であり、視線に対して直角である。プリミティブが三角形である場合、3 つの頂点が含まれ、任意の方位を有する平面を規定する。プリミティブの平面と、近距離および遠距離の平面との交差点は 2 つの関連する一次方程式を有する 2 つの直線を含んでいる。

【 0 1 7 7 】

したがって、各プリミティブはそれが三角形または直線の形態を取るかに応じて全部で 9 または 10 の一次方程式をそれぞれ有する。再び三角形の場合、このような一次方程式は三角形を規定する 3 つの一次方程式と、境界のあるボックスを規定する 4 つの一次方程式と、プリミティブが存在する平面と近距離の平面および遠距離の平面との交差点を規定する 2 つの一次方程式とを含んでいる。

【 0 1 7 8 】

図 2 7 を参照し続けると、プロセスは動作 2704 で進行し、プリミティブ上またはその近くの複数の点を位置付ける。スタート位置 2602 は図 2 6 A で示されているように、このような位置付けを指示している。このような点は含まれる凸形領域を規定し、凸形領域のコーナーに位置している。図 2 7 A は例えば長方形等の凸形領域 2707 を囲むこのようなセンスポイント 2705 を示している。1 実施形態では、このような長方形はサイズが 8 × 2 画素である。さらに点はプリミティブの上部の頂点を囲むように最初に位置される。選択肢として、これは切捨てを使用して実現されてもよい。

【 0 1 7 9 】

プリミティブが一度位置付けられると、プロセスは以下説明する方法でプリミティブの行を処理することにより動作 2706 で開始するトラバーサルモジュール 58 により継続される。各行の処理後、ジャンプ位置が決定 2708 で発見されるか否かを決定する。ジャンプ位置は次の行を処理するためスタート位置にあり、以下詳細に説明する。決定 2708 でジャンプ位置が発見されたことが決定されるならば、凸面領域を規定するセンスポイントは動作 2710 に移動される。しかしながら、ジャンプ位置が発見されていないことが決定されたならば、プロセスは終了される。別の実施形態では、列、対角線または任意の他のタイプのストリングが行の代わりに動作 2706 で処理されることに注意すべきである。

【 0 1 8 0 】

図 2 8 は図 2 7 の処理行動作 2706 に関連した本発明のプロセスを示すフローチャートで

10

20

30

40

50



ある。示されているように、プロセスは多角形を規定するセンスポイントが決定2801で右に移動されるか否かを決定するため、動作2800でセンスポイントを計算することにより開始する。このような決定は最も右のセンスポイントの位置に基づいて行われる。最も右のセンスポイントがプリミティブの同一エッジ外に位置されないならば、右方向の移動は許容され、現在位置の右への位置（XおよびY座標）は動作2802でスナップ位置として記憶される。しかしながら、最も右のセンスポイントがプリミティブの1以上のエッジ外に位置されるならば、右方向の移動は許容されず、動作2802はスキップされる。

【0181】

次に、一次方程式は動作2804で凸形領域、例えば長方形の点で評価される。この評価は点がプリミティブ中に存在するか否かの決定を含んでいる。ポイントがプリミティブ中に存在するか否かについてのこのような決定は、各一次方程式の評価が各センスポイントで正の値または負の値を与えるか否かを決定することを含んでいる。

10

【0182】

一次方程式はプリミティブ内では正であり、その外では負であるように公式化されることができる。画素が丁度エッジ上に存在する包含的なエッジが描かれ、ゼロに評価され、正として扱われる。描かれるべきではない排他的なエッジは開始の一次方程式の値から1の値を最初に減算することにより負にされることができる。したがって、排他的エッジ上の画素は正のゼロの代わりに負値（-1）に評価される。これはセンスポイントの移行が包含的/排他的ポリシーを無視し、単に一次方程式の符号を試験することを許容する。

【0183】

20

一次方程式が点において評価された後、決定2806でセンスポイントの現在の位置がジャンプ位置を構成するか否かが決定される。2つの下部のセンスポイントが両者ともエッジ外でなければ、ジャンプ位置は記憶されることに注意すべきである。決定2806で、ジャンプ位置が発見されたことが決定されたならば、動作2808でジャンプ位置が計算され記憶される（または存在するならば先に記憶されたジャンプ位置で置換する）。しかしながらノーであるならば、動作2808はスキップされる。

【0184】

図28の参照を続けると、決定2810で、最も左のセンスポイントが両者ともプリミティブのエッジ外であるか否かが決定される。このプロセスは再び両者の最も左のセンスポイントの一次方程式の評価が正または負値を与えるか否かを決定することを含んでいる。特に適切なセンスポイントで9または10のエッジ式の係数を計算するとき、9または10値が与えられ、それらは9または10の符号ビットを有する。現在の側面が完全にエッジ外であるか否かを決定するために、例えば本発明は2つのセンスポイントからの10の符号ビットを共に論理積（AND）処理する。任意のビットが残存するならば、両者のポイントはそのエッジ外である。

30

【0185】

最も左のセンスポイントが両者ともプリミティブエッジ外ではないことが決定されたならば、左方向にあると考えられるプリミティブの部分がさらに残留していることが結論付けされ。センスポイントは動作2812に左へ移動される。決定2810で、両者の最も左のセンスポイントがプリミティブのエッジ外であることが決定されたならば、左方向にあると考えられるプリミティブの部分がさらに残留していないことが結論付けされる。次に、決定2814で、動作2802から得られたスナップ位置が存在するか否かの決定が行われる。

40

【0186】

決定2814で、スナップ位置が存在しないことが決定されたならば、プロセスは行われる。しかしながら、スナップ位置が存在するならば、センスポイントは動作2816でスナップ位置に移動される。その後、2804 - 2812の動作に類似した動作はプリミティブの右側をマップするように行わされる。これは凸形領域の点で一次方程式を評価することにより動作2818を開始する。

【0187】

一次方程式が点で評価された後、センスポイントの現在の位置が動作2820でジャンプ位

50

置を構成するか否かが決定される。決定2820で、ジャンプ位置が発見されたことが決定されたならば、動作2822でジャンプ位置が計算され記憶される。ノーであるならば、動作2822はスキップされる。

【0188】

図28の参照を続けると、決定2824で、最も右のセンスポイントが両者ともプリミティブのエッジ外であるか否かが決定される。最も右のセンスポイントが両者ともプリミティブのエッジ外ではないことが決定されたならば、右方向にあると考えられるプリミティブ部分がさらに残留していることが結論付けされ、センスポイントは動作2826で右に移動される。決定2824で、最も右のセンスポイントが両者ともプリミティブのエッジ外にあることが決定されたならば、右方向にあると考えられるプリミティブの部分がさらに残留していないことが結論付けされ、瞬時のプロセスが実行される。

10

【0189】

図28Aと図28Bは本発明のセンスポイントがプリミティブ2850に関して移動されるシーケンスを示している。種々の代りの方法が決定2800で点が左に移動することができるか否かを決定し、最初に右に進行することを含むことに注意する。一次方程式は点が任意の所望の方法でプリミティブ内または外であることを示すために規定される。

【0190】

反復するループのステップ動作を防ぐため、本発明はしたがってラスタ化中に全体的な移動方向を使用する。最初の構成はトップ-ダウンを行い、次へステップダウンする前に1つの行の1つ1つの凸領域に行く。行のトップ-ダウンを行い、右その後左へ、または左その後右へステップしないことによりループは阻止される。

20

【0191】

前述のプロセスの1例は図27Aの多角形を規定する点P1、P2、P3、P4を参照して示されている、動作において、隣接するセンスポイントの対はそれらの方向のステップングが生産的 (productive) であるか否かを決定するため検査されることができる。例えば図27A中のP3とP4の両者が多角形のエッジ外であるが、P1および/またはP2は多角形のエッジ外ではないならば、明白に描くことのできる内部エリアは右ではなく左に位置する。したがってセンスポイントは右へ移動すべきではない。反対に、P3とP4の両者が全てのエッジ内であるならば、描くことのできる内部エリアは丁度P3とP4を越えて存在し、右へのステップが適切である。P3とP4が同じエッジの外ではないならば、右へのステップが生産的である。この同じ論理はP1とP3により誘導される上方向へのステップまたは、P1とP2により誘導される左のステップ、またはP2とP4に基づいた下方向のステップにも適用される。

30

【0192】

前述のプロセスはしたがってガイドとしてセンスポイントを使用して、プリミティブの内部周辺の点により規定される凸形区域を移動またはステップする。点によって規定される凸形領域が大きいので、多数の画素は同時に試験される。使用中、全てのセンスポイントがプリミティブの全てのエッジ内であるならば、全ての囲まれた画素は描かれることが可能でなければならない (凸形のプリミティブを想定する)。コーナー部を検査することにより多くの利点が与えられ、即ちプリミティブの任意のエリアを与える能力は内部、外部または分割である。後者のケースでのみ、点により規定される凸形領域の個々の画素が試験される必要がある。このような場合、点により規定される凸形領域の画素はこれらがプリミティブに存在するか否かを決定するために別の方法により1つずつ試験される。さらに、センスポイントはエリアを分割するエッジと分割しないエッジを規定することにより、必要な試験の量を減少する。

40

【0193】

図29は図27の処理行動作2706に関連した本発明のプロセスの別の犁耕体プロセスを示すフローチャートである。示されているように、最初に決定2900で、先の移動が第1または第2の方向であるかを決定する。実際に先の移動が存在しなかったならば、デフォルトの先の移動が仮定される。決定2900で、先の移動が第2の方向であることが決定された

50

ならば、動作2902で図28の動作2804と類似した方法で一次方程式が凸形領域、例えば長方形の点で評価される。

【0194】

図29の参照を続けると、次に、決定2904で、長方形の第1の側面のセンスポイントが両者ともプリミティブのエッジ外であるか否かに関して決定が行われる。ノーであるならば、センスポイントは動作2906で第1の方向で移動またはステップされる。長方形の第1の側面のセンスポイントが両者ともプリミティブのエッジ外であるという決定が行われると、決定2905で、点が下方向に移動できるか否か、換言すると、現在位置がジャンプ位置を構成するか否かが決定される。イエスならば、動作2908でジャンプ位置が計算され、記憶され、その後プロセスが行われる。

10

【0195】

他方で、決定2900で、先の移動が第1の方向であることが決定されたならば、動作2902 - 2908と類似の動作が実行される、特に動作2910で、動作一次方程式は凸形領域、例えば長方形の点で評価される。決定2912で、長方形の第2の側面のセンスポイントが両者ともプリミティブのエッジ外であるか否かに関する決定が行われる。ノーであるならば、センスポイントは動作2914で第2の方向で移動またはステップされる。長方形の第2の側面のセンスポイントが両者ともプリミティブのエッジ外であるという決定が行われると、決定2913で、点が下方向に移動できるか否か、換言すると、現在位置がジャンプ位置を構成するか否かが決定される。イエスならば、動作2916でジャンプ位置が計算され、記憶され、その後プロセスが行われる。

20

【0196】

図29Aは図29の犁耕体プロセスにしたがって本発明のセンスポイントがプリミティブに関して移動されるシーケンスを示している。前述の犁耕体ラスタ化はハードウェアに対してより良好な性能を与えるあるルールに従うようにシーケンスを規制する。示されているように、犁耕体ラスタ化は前後に曲がる蛇行パターンを与える。水平の犁耕体シーケンスは例えばプリミティブ三角形内に全ての画素を生成し、それらは左から右へ1つの行に存在し、その後、次の行で右から左へ画素を生成する。このような折曲がったパスは生成された画素から最近予め発生された画素までの平均距離が比較的小さいことを確実にする。

【0197】

ほぼ最近に予め発生された画素の発生は、画素および/またはそれらの対応するテキスト値が限定されたサイズのメモリ中に維持されるときに重要である。犁耕体シーケンスはこのようなメモリに既にロードされている画素またはテキストを頻繁に発見し、それ故メモリのロードの反復が行われる頻度が少なくなる。

30

【0198】

1つの選択肢として、ラスタ化の前にプリミティブを複数の部分に分離する少なくとも1つの境界が使用される。動作において、点は各部分で別々に移動される。さらに、点は第2の部分で移動される前に第1の部分の全体を移動される。

【0199】

図30は境界を使用する別の犁耕体プロセスを示しているフローチャートである。1つの選択肢として、境界を使用するか否かの決定はプリミティブの大きさに基づく。図30で示されているように、境界を処理する犁耕体プロセスは、少なくとも1つの境界が規定され、プリミティブを複数の部分またはスワス (swath) に分割する付加的な動作3000を除いて図27のプロセスと類似している。

40

【0200】

図30の参照を続けると、付加的な決定3001はプリミティブのあらゆる部分の完了にしたがう。特に、決定3001で、隣接部分のスタート位置が動作3006で発見されるか否かが決定される。イエスであるならば、センスポイントにより規定される凸形領域は動作3002でプリミティブの隣接部分のスタート点に移動され、動作3004 - 3010はプリミティブの新しい部分に対して反復される。さらに動作3006のスタート位置の決定に関する情報を図31

50

を参照してさらに詳細に説明する。

【0201】

図31Aは図30の境界ベースの犁耕体プロセスにしたがって本発明の凸形状領域がプリミティブに関して移動されるプロセスを示している。示されているように、処理される第1の部分はプリミティブの最上部の頂点を含む部分である。動作中、左の隣接部分が処理され、その後近接する左の隣接部分が処理され、以下同様に処理される。これは、左の隣接部分がなくなるまで継続される。次に第1の部分の右への隣接部分が処理され、その後、近接する右の隣接部分が処理され、全ての右の隣接部分が処理されるまで継続される。他のタイプの順序付け方式がユーザの要望にしたがって使用されてもよいことを認識すべきである。

10

【0202】

図31は図30のプロセス行の動作3006に関連したプロセスを示すフローチャートである。このようなプロセスは決定3118と3121を除いて図29の犁耕体プロセスに類似している。決定3118と3120の両者は任意のセンスポイントが任意の境界を通過しているか否かを決定する。センスポイントが境界内であることが決定されさえすれば、それぞれのループが継続される。

【0203】

動作3119と3121では、プリミティブの隣接部分のスタート位置は搜索され、決定3118と3120で、凸形領域の任意のセンスポイントがそれぞれ任意の境界を通過していることを決定したときに記憶される。図31Aで示されているように、このようなスタート位置3126は境界を越えて存在するプリミティブ部分の最上部点としてそれぞれ規定される。この位置を記憶することにより、プロセスがプリミティブにおける隣接する境界の規定された部分で反復されるときにスタート点が与えられる。

20

【0204】

動作3119と3121は両者ともプリミティブの第1の部分処理しながら実行されることに注意する。図31で明白に示していないが、部分を第1の部分の左に処理するときこのような動作の第1の動作だけが行われ、部分を第1の部分の右に処理するとき、このような動作の第2の動作だけが行われる。換言すると、部分を第1の部分の左に処理するとき、スタート位置は現在処理された部分の最も左の境界が超過されたときだけ決定される。同様に、部分を第1の部分の右に処理するとき、スタート位置は現在処理された部分の最も

30

【0205】

ラスタ化中に境界を使用することは、パイプライン処理中の非常に臨界的な問題を解決する。プリミティブが非常に広いならば、1つの行の画素に関連する記憶媒体は限定されたサイズのメモリに適合しない。境界によるラスタ化は三角形を限定された幅の行(または列)に分離し、次の部分へ移動する前に、このような部分内に全ての画素を生成する。

【0206】

例えば、三角形が100画素幅であっても、限定されたサイズの画素またはテキストチャメモリは先の20画素の情報だけを保持する。画素シーケンスを10画素幅の垂直部分内に存在するように制限することによって、以前のおよび現在の行の全ての画素はメモリに適合することが可能である。これは、境界の規定された部分内の犁耕体シーケンスが常にメモリの(存在するならば)現在の行の以前の画素と、メモリの(存在するならば)上の行の画素とを有することを意味している。

40

【0207】

ほとんどの基礎的なメモリシステムはブロック単位のあるオーバーヘッドによりデータのブロックを転送する。メモリシステムに対する小さいアクセスはこのオーバーヘッドにより重いペナルティを課される。効率的であるように、大きいアクセスが使用され、ブロックの残りは次に使用される場合のために維持される。さらに、キャッシュメモリシステムは複数のこれらの最近のブロックを維持し、メモリアクセスが避けられることができる

50

確率を増加させる。

【0208】

本発明の犁耕体シーケンスは、現在のラインの1端部のすぐ下の画素を反転し処理するときシングル-リテン-ブロックを使用する。さらに、犁耕体シーケンスはラスタ化を特定サイズの部分に限定するときキャッシュを使用する。特に部分内の2つの走査線はキャッシュに適合され、第2の走査線を通じて第1の走査線のキャッシュ記憶から利点を得られる。

【0209】

シーケンスまたは境界の規定された部分の数には制限はない。本発明は垂直部分および水平の犁耕体パターンの例を使用した。類似の原理が水平部分および垂直の犁耕体パターンまたは、対角線部分およびパターンまで拡張される。1実施形態では、ストリング(例えば、行、列、対角線等)の長さはストリングが存在するプリミティブの大きさよりも小さいようにそれぞれ限定される。

10

【0210】

図32は図27の動作2702に関連したプロセスに関連したプロセスを示すフローチャートである。瞬間的なプロセスは目の後に存在する部分でプリミティブを処理するように設計されている。これらの域外の部分はその次のラスタ化動作で問題を生じる。これを実現するため、瞬間的なプロセスは変数Wを使用し、これは投影、即ち遠近法でオブジェクトを観察するために共通して使用される。変数Wは他の座標X、Y、Zが近くのを大きく、遠くのを小さくするために割算される数字である。変数Wは投影の中心と、対応する頂点との間の距離を表す。

20

【0211】

図32で示されているように、プリミティブは最初に受信され、複数の頂点により規定される。そのような各頂点はW値を含んでいる。プリミティブを受信するとき、設定モジュールは頂点に基づいてプリミティブを特徴付けするラインを規定する役目を行う。動作3200に注意。

【0212】

W値はその後、決定3202で解析される。示されているように、1つのW-値が負であるならば、負の値を有する頂点と反対のラインの一次方程式は動作3204でフリップされる。換言すると、一次方程式の係数は-1により乗算される。さらに、2つのW-値が負であるならば、正のW-値を有する頂点と、負のW-値を有するそれぞれの頂点とを接続するラインの一次方程式は動作3206でフリップされる。3つのW-値が負であるならば、不合格(カル)状態3207が生じ、本発明は三角形を不合格とする。負であるW-値がないならば、付加的な措置は取られない。

30

【0213】

図32A-32Cはフリップ一次方程式が、処理されるスクリーンの部分に影響を与える方法を示している。図32AはW値が負のものではなく、一次方程式が変更されない状態の場合を示している。示されているように、プリミティブの内部部分はこのようなケースで満たされている。

【0214】

図32Bは1つのW-値が負であり、したがってその一次方程式がフリップされるケースを示している。示されているように、頂点と対向するプリミティブ部分は現在のケースで満たされている。特に、描かれるエリアは-W頂点を共有する2つの三角形の面と共直線性である2つのラインにより境界を与えられ、さらに、2つの+W頂点を共有する三角形の面により境界を与えられる。

40

【0215】

図32Cは2つのW-値が負であり、したがってその一次方程式がフリップされるケースを示している。示されているように、頂点と対向するプリミティブ部分は図27-32を参照して前述した方法および/またはプロセスを使用して満たされる。換言すると、描かれるエリアは+W頂点を共有する2つの三角形の面と共直線性である2つのラインによ

50

り境界を与えられ、さらに、+W頂点に近接する。

【0216】

本発明はしたがって全ての3つの前述のケースを処理することができる。三角形の部分が近距離および/遠距離の平面を越えているならば、これらの平面内にその部分だけを描く。三角形が1または2の負のZ頂点を有するならば、正確な+Z部分だけが描かれる。

【0217】

全ての頂点がオフスクリーンであり、三角形が目の後方から遠距離の平面を越えて延在しても、画素は三角形内およびスクリーン上であり、近限界と遠限界との間にZを有する。本発明は悪い画素を使用する時間の浪費を少なくすることを確実にする。スクリーンエッジまたは近距離平面と遠距離平面による全てのクリッピングが容易に使用されることが

10

【0218】

スタート点が満たされるエリア内ではないときに時によって問題が生じる。上部頂点がオフスクリーンであるか近距離の平面または遠距離の平面によりクリップされる場合にこれは生じる。この場合、トラバーサルステージは描かれる区域の上部点を検索しなければならず、上から開始する。これは三角形のエッジスロープとZスロープの符号により誘導されることにより効率的に行われる。これは三角形の一次方程式が描かれる領域外であることとその理由を発見するために、三角形の一次方程式を試験できる。外部にあるエッジおよび/またはZ限界を知ったとき、そのエッジまたは限界へ近付けるステップ方向を知る。(選択肢のあるとき)好みによって水平から垂直に移動することによって、描かれた

20

【0219】

本発明の1実施形態ではトラバーサルは三角形の上部から下部へ進行する。負のW-値をもたず頂点がシザー長方形であるならば、スタート点は三角形の上部の頂点である。トラバーサルは常にシザー長方形内で開始し、その外ではないので、エッジにより囲まれるエリアがシザー長方形を越えて延在しても、シザー長方形内の三角形部分だけが描かれる。このようにして、簡単なシザー長方形の長方形エッジクリッピングが行われる。

【0220】

種々の実施形態を前述したが、これらは技術的範囲の限定ではなく例示でのみ示されていることを理解すべきである。したがって、本発明の技術的範囲は前述の例示的な実施形態により限定されず、特許請求の範囲とそれらの均等物にしたがってのみ限定される。

30

【図面の簡単な説明】

【0221】

【図1】従来技術のグラフィック処理方法のフロー図。

【図1A】単一の半導体プラットフォーム上に構成された本発明の1実施形態の種々のコンポーネントを示すフロー図。

【図2】本発明の1実施形態による頂点属性バッファ(VAB)の概略図。

【図2A】本発明の1実施形態によるVABによって受取られることのできる種々のコマンドを示すチャート。

40

【図2B】本発明の1実施形態によるVABとの間で頂点属性をロードし、ドレインする方法を示すフローチャート。

【図2C】図2Bの動作を実行するために使用される本発明のアーキテクチャの概略図。

【図3】本発明の1実施形態によるVABに関連したモードビットの説明図。

【図4】本発明の変換モジュールを示す概略図。

【図4A】本発明の1実施形態による多数の実行スレッドを実行する方法を示すフローチャート。

【図4B】本発明の1実施形態にしたがって図4Aの方法が行われる手順を示すフロー図

50

- 【図5】本発明の1実施形態による図4の変換モジュールの機能装置の概略図。
- 【図6】図5の変換モジュールの乗算論理装置（MLU）の概略図。
- 【図7】図5の変換モジュールの演算論理装置（ALU）の概略図。
- 【図8】図5の変換モジュールのレジスタファイルの概略図。
- 【図9】図5の変換モジュールの反転論理装置（ILU）の概略図。
- 【図10】本発明の1実施形態による図5の変換モジュールの出力コンバータの出力アドレスのチャート。
- 【図11】本発明の1実施形態による図5の変換モジュールのマイクロコード編成図。
- 【図12】本発明の1実施形態による図5の変換モジュールのシーケンサの概略図。
- 【図13】図12の変換モジュールのシーケンサの使用に関連した種々の動作を詳細に示すフローチャート。 10
- 【図14】図12の変換モジュールのシーケンサのシーケンシングコンポーネントの動作を詳細に示すフロー図。
- 【図14A】グラフィック処理中にスカラーおよびベクトル成分を処理するために使用される本発明のコンポーネントを示すフロー図。
- 【図14B】図5の変換モジュールに対応している図14Aに示されている本発明の機能コンポーネントの1つの可能な組合せ1451を示すフロー図。
- 【図14C】図14Aに示されている本発明の機能コンポーネントの別の可能な組合せ1453を示すフロー図。
- 【図14D】本発明の1実施形態にしたがって図12の変換モジュールにより実施されるグラフィック処理中にブレンディング動作を行う方法を示すフロー図。 20
- 【図15】本発明の1実施形態のライティングモジュールの概略図。
- 【図16】本発明の1実施形態による図15のライティングモジュールの機能装置を示す概略図。
- 【図17】本発明の1実施形態による図16のライティングモジュールの乗算論理装置（MLU）の概略図。
- 【図18】本発明の1実施形態による図16のライティングモジュールの演算論理装置（ALU）の概略図。
- 【図19】本発明の1実施形態による図16のライティングモジュールのレジスタ装置の概略図。 30
- 【図20】本発明の1実施形態による図16のライティングモジュールのライティング論理装置（LLU）の概略図。
- 【図21】本発明の1実施形態による図16のライティングモジュールに関連したフラッグレジスタの説明図。
- 【図22】本発明の1実施形態による図16のライティングモジュールに関連したマイクロコードフィールドの説明図。
- 【図23】本発明の1実施形態による図16のライティングモジュールに関連したシーケンサの概略図。
- 【図24】本発明の1実施形態にしたがって変換およびライティングモジュールのシーケンサが関連したバッファの入力および出力をどのように制御することができるかを詳細に説明するフローチャート。 40
- 【図25】図24の方法にしたがって変換およびライティングモジュールのシーケンサが関連したバッファの入力および出力をどのように制御することができるかを示す概略図。
- 【図25B】図1Aのラスタ化装置の種々のモジュールの概略図。
- 【図26】本発明のラスタ化モジュールの設定モジュールの概略図。
- 【図26A】図26のラスタ化装置の設定モジュールによって計算される種々のパラメータを示す説明図。
- 【図27】図26に示されているラスタ化装置コンポーネントの設定およびトラバースルモジュールに関連した本発明の方法を示すフローチャート。
- 【図27A】本発明の1実施形態にしたがってプリミティブにおけるエリアを識別するた 50

めに移動される凸状領域を囲む方向ポイントを示す説明図。

【図28】図27の処理行動作2706に関連している本発明のプロセスを示すフローチャート。

【図28A】本発明の凸状領域がプリミティブに関して移動されるシーケンスを示す概略図。

【図28B】本発明の凸状領域がプリミティブに関して移動されるシーケンスの別の例を示す概略図。

【図29】図27の処理行動作2706に関連した本発明のプロセスの別の犁耕体プロセスを示すフローチャート。

【図29A】図29の犁耕体プロセスにしたがって本発明の凸状領域がプリミティブに関して移動されるシーケンスを示す概略図。

【図30】境界を使用する別の犁耕体プロセスを示すフローチャート。

【図31】図30の動作3006に関連したプロセスを示すフローチャート。

【図31A】図30および31の境界ベースの犁耕体プロセスにしたがって本発明の凸状領域がプリミティブに関して移動されるシーケンスを示す概略図。

【図32】図27の動作2702に関連したプロセスを示すフローチャート。

【図32A】図32のプロセスにおいて負のW値が1つも計算されないとき、どのようなエリアが描かれるかを示す説明図。

【図32B】図32のプロセスにおいて負のW値が1つだけ計算されたとき、どのようなエリアが描かれるかを示す説明図。

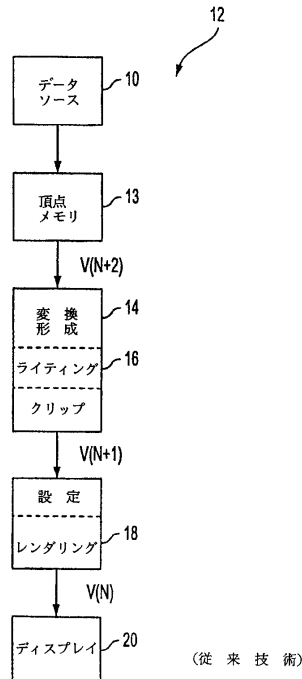
【図32C】図32のプロセスにおいて負のW値が2つだけ計算されたとき、どのようなエリアが描かれるかを示す説明図。

10

20

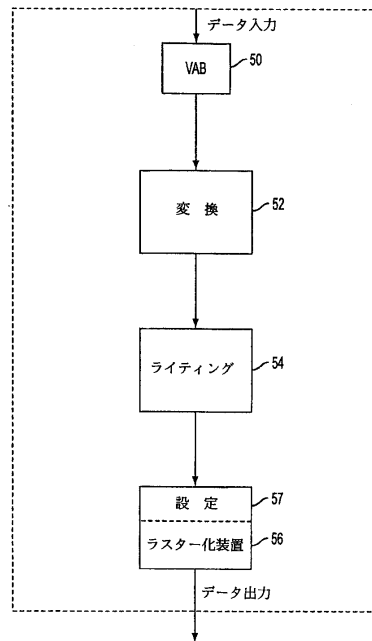
【図1】

図1



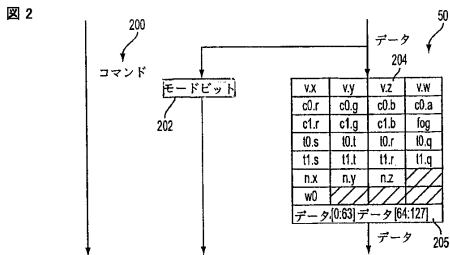
【図1A】

図1A





【図2】



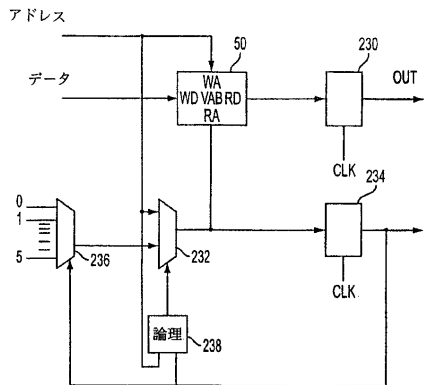
【図2A】

図2A

コマンド	変換	ライティング	記述
	機能停止	機能停止	
FE2XF_CMD_NOP			NO OPERATION. CAN BE USED AS A SPACER BETWEEN COMMANDS.
FE2XF_CMD_VERTEX	READ	READ	VERTEX DATA.
FE2XF_CMD_PASSTHR			PASSTHROUGH. TRANSFORM AND LIGHTING PASS THE DATA THROUGH.
FE2XF_CMD_RDVAB			READ THE VAB CONTENTS WHEN CONTEXT SWITCHING.
FE2XF_CMD_LDMODE			LOAD NEW MODE BITS.
FE2XF_CMD_LDXFCTX	WRITE		LOAD TRANSFORM CONTEXT MEMORY DATA.
FE2XF_CMD_RDXFCTX	READ		READ TRANSFORM CONTEXT MEMORY DATA.
FE2XF_CMD_LDLTCTX	WRITE		LOAD LIGHTING CONTEXT MEMORY DATA.
FE2XF_CMD_RDLTCTX	READ		READ LIGHTING CONTEXT MEMORY DATA.
FE2XF_CMD_LDLTCT0	WRITE		LOAD LIGHTING CONTEXT0 MEMORY DATA.
FE2XF_CMD_RDLTCT0	READ		READ LIGHTING CONTEXT0 MEMORY DATA.
FE2XF_CMD_LDLTCT1	WRITE		LOAD LIGHTING CONTEXT1 MEMORY DATA.
FE2XF_CMD_RDLTCT1	READ		READ LIGHTING CONTEXT1 MEMORY DATA.
FE2XF_CMD_LDLTCT2	WRITE		LOAD LIGHTING CONTEXT2 MEMORY DATA.
FE2XF_CMD_RDLTCT2	READ		READ LIGHTING CONTEXT2 MEMORY DATA.
FE2XF_CMD_LDLTCT3	WRITE		LOAD LIGHTING CONTEXT3 MEMORY DATA.
FE2XF_CMD_RDLTCT3	READ		READ LIGHTING CONTEXT3 MEMORY DATA.
FE2XF_CMD_SYNC	READ+WRITE	READ+WRITE	SIMILAR TO NOP, BUT IS NOT ALLOWED TO BE PROCESSED IN PARALLEL.

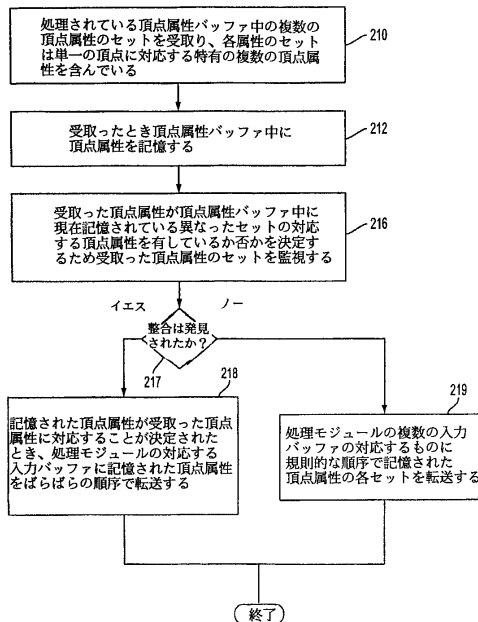
【図2C】

図2C



【図2B】

図2B



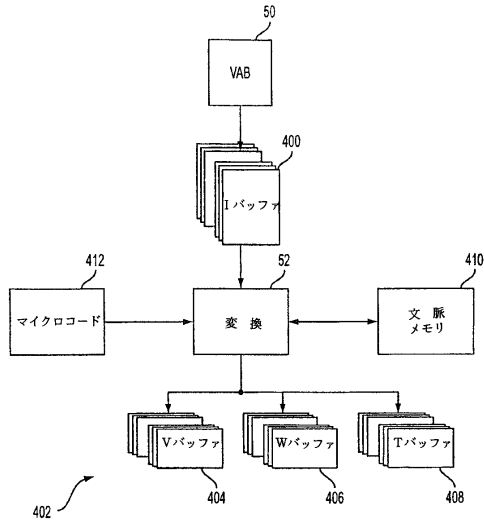
【図3】

図3

モードビット	ビット	記述
T0	1	TEXTURE 0 ENABLE
TXF0	1	TEXTURE 0 MATRIX TRANSFORM ENABLE
TDV0	1	TEXTURE 0 w/DIVIDE ENABLE
T0S	3	TEXTURE 0 TEXGEN s CONTROL
T0T	3	TEXTURE 0 TEXGEN t CONTROL
T0U	3	TEXTURE 0 TEXGEN r CONTROL
T0Q	2	TEXTURE 0 TEXGEN q CONTROL
T1	1	TEXTURE 1 ENABLE
TXF1	1	TEXTURE 1 MATRIX TRANSFORM ENABLE
TDV1	1	TEXTURE 1 w/DIVIDE ENABLE
T1S	3	TEXTURE 1 TEXGEN s CONTROL
T1T	3	TEXTURE 1 TEXGEN t CONTROL
T1U	3	TEXTURE 1 TEXGEN r CONTROL
T1Q	2	TEXTURE 1 TEXGEN q CONTROL
ETY	1	EYE TYPE INFINITE(0) OR LOCAL(1)
LIT	1	LIGHTING ENABLE
NRM	1	NORMAL NORMALIZE ENABLE
FOG	1	FOG ENABLE
LIS	16	LIGHT STATUS ( 8 LIGHTS BY 2 BITS EACH, 0:OFF,1:INFINITE,2:LOCAL,3:SPOTLIGHT)
FG	2	FOGGEN CONTROL(0: OFF, 1:RADIAL, 2: PLANE)
LAT	1	LIGHT ATTENUATION CONTROL (0: INVERT, 1: NO INVERT)
C1I	1	SPECULAR COLOR INPUT ENABLE
C1O	1	SPECULAR COLOR OUTPUT ENABLE
CM	4	COLOR MATERIAL CONTROL (1: EMISSIVE, 2: AMBIENT, 4: DIFFUSE, 8: SPECULAR)
PP	1	POINT PARAMETER ENABLE
SKIN	1	SKINNING ENABLE
VPAS	1	VERTEX PASS ENABLE

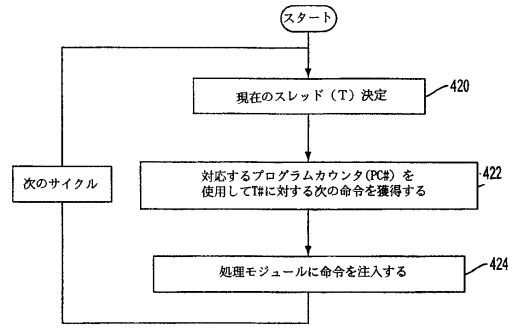
【 図 4 】

図 4



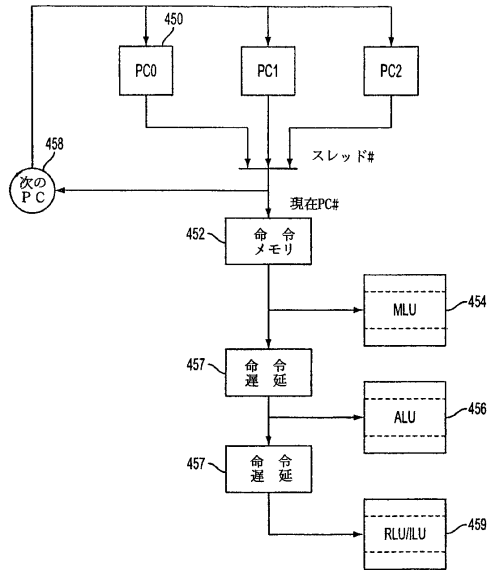
【 図 4 A 】

図 4A



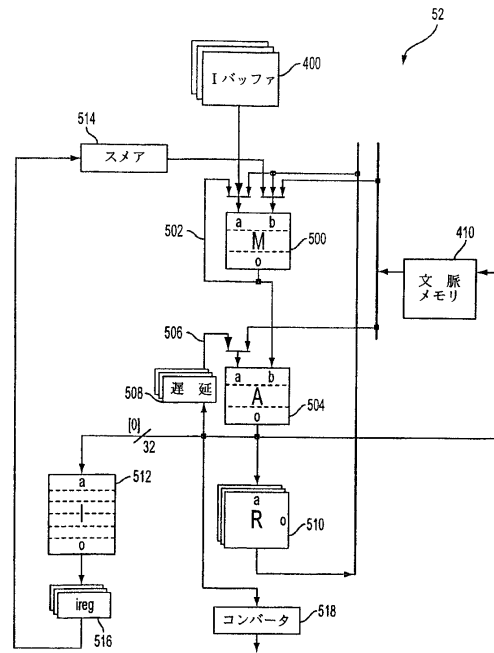
【 図 4 B 】

図 4B



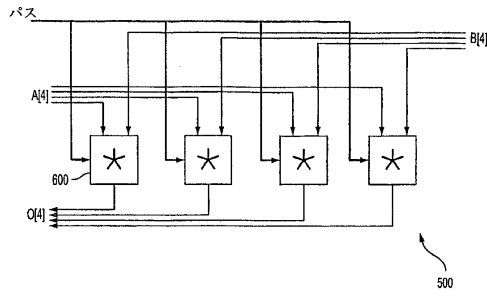
【 図 5 】

図 5



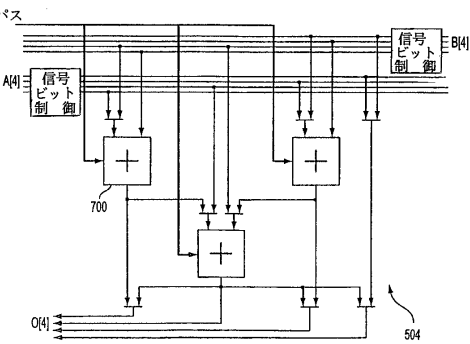
【 図 6 】

図 6



【 図 7 】

図 7



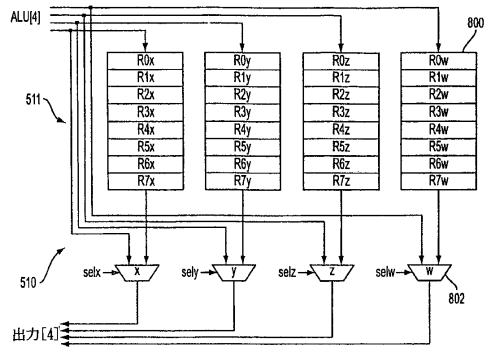
【 図 10 】

図 10

アドレス	ターゲット	作用	記述
TPOS	TBUFFER	T[0] = ALU	POSITION
TT0	TBUFFER	T[3] = ALU	TEXTURE0
TT1	TBUFFER	T[4] = ALU	TEXTURE1
WEV	WBUFFER,VBUFFER	W[0] = ALU, V[0]y = ALU.w	EYE VECTOR
WL0	WBUFFER,VBUFFER	W[1] = ALU, V[1]y = ALU.w	LIGHT0 DIRECTION VECTOR
WL1	WBUFFER,VBUFFER	W[2] = ALU, V[2]y = ALU.w	LIGHT1 DIRECTION VECTOR
WL2	WBUFFER,VBUFFER	W[3] = ALU, V[3]y = ALU.w	LIGHT2 DIRECTION VECTOR
WL3	WBUFFER,VBUFFER	W[4] = ALU, V[4]y = ALU.w	LIGHT3 DIRECTION VECTOR
WL4	WBUFFER,VBUFFER	W[5] = ALU, V[5]y = ALU.w	LIGHT4 DIRECTION VECTOR
WL5	WBUFFER,VBUFFER	W[6] = ALU, V[6]y = ALU.w	LIGHT5 DIRECTION VECTOR
WL6	WBUFFER,VBUFFER	W[7] = ALU, V[7]y = ALU.w	LIGHT6 DIRECTION VECTOR
WL7	WBUFFER,VBUFFER	W[8] = ALU, V[8]y = ALU.w	LIGHT7 DIRECTION VECTOR
WSL0	WBUFFER	W[9] = ALU	SPOTLIGHT0 cos
WSL1	WBUFFER	W[10] = ALU	SPOTLIGHT1 cos
WSL2	WBUFFER	W[11] = ALU	SPOTLIGHT2 cos
WSL3	WBUFFER	W[12] = ALU	SPOTLIGHT3 cos
WSL4	WBUFFER	W[13] = ALU	SPOTLIGHT4 cos
WSL5	WBUFFER	W[14] = ALU	SPOTLIGHT5 cos
WSL6	WBUFFER	W[15] = ALU	SPOTLIGHT6 cos
WSL7	WBUFFER	W[16] = ALU	SPOTLIGHT7 cos
VED	VBUFFER	V[0].x = 1.0, V[0].z = ALU.w	EYE RADIAL DISTANCE VECTOR
VLD0	VBUFFER	V[1].x = 1.0, V[1].z = ALU.w	LIGHT0 DISTANCE VECTOR
VLD1	VBUFFER	V[2].x = 1.0, V[2].z = ALU.w	LIGHT1 DISTANCE VECTOR
VLD2	VBUFFER	V[3].x = 1.0, V[3].z = ALU.w	LIGHT2 DISTANCE VECTOR
VLD3	VBUFFER	V[4].x = 1.0, V[4].z = ALU.w	LIGHT3 DISTANCE VECTOR
VLD4	VBUFFER	V[5].x = 1.0, V[5].z = ALU.w	LIGHT4 DISTANCE VECTOR
VLD5	VBUFFER	V[6].x = 1.0, V[6].z = ALU.w	LIGHT5 DISTANCE VECTOR
VLD6	VBUFFER	V[7].x = 1.0, V[7].z = ALU.w	LIGHT6 DISTANCE VECTOR
VLD7	VBUFFER	V[8].x = 1.0, V[8].z = ALU.w	LIGHT7 DISTANCE VECTOR
VCO	WBUFFER,TBUFFER	V[9] = ALU, T[1] = ALU	DIFFUSE COLOR
VC1	WBUFFER,TBUFFER	V[10] = ALU, T[2] = ALU	SPECULAR COLOR
VNRM	VBUFFER	V[11] = ALU	NORMAL VECTOR
VED2	VBUFFER	V[12] = ALU	EYE PLANAR DISTANCE VECTOR
TWV_NOP			NO VALID OUTPUT.

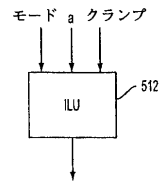
【 図 8 】

図 8



【 図 9 】

図 9

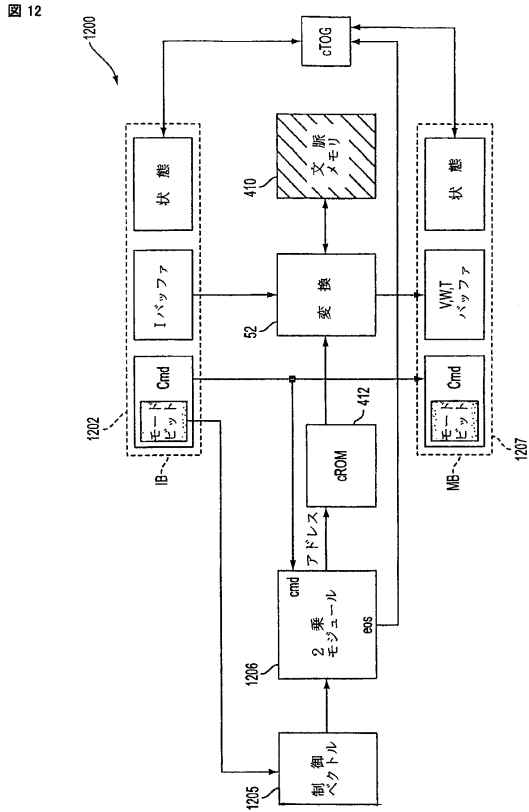


【 図 11 】

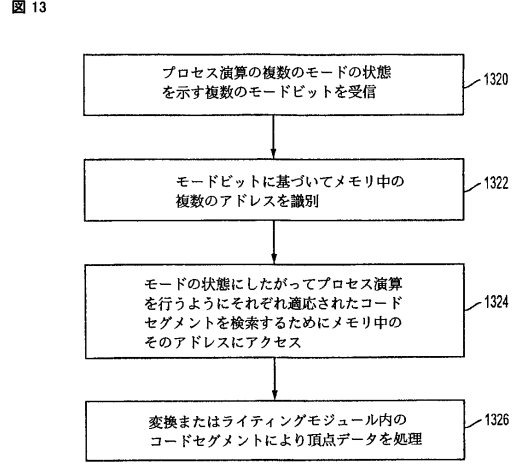
図 11

マイクロコード フィールド	ビット	位置	遅延	記述
oa	6	0:5	2	OUTPUT BUFFER WRITE ADDRESS
rta	3	6:8	0	RLU READ ADDRESS
rwm	4	9:12	2	RLU WRITE MASK
rwa	3	13:15	2	RLU WRITE ADDRESS
ilu	2	16:17	2	ILU OPERATION
alu	4	18:21	1	ALU OPERATION
ais	2	22:23	1	ALU SIGN CONTROL
ala	1	24	1	ALU INPUT A MUX
mlu	3	25:27	0	MLU OPERATION
mib	2	28:29	0	MLU INPUT B MUX
mia	2	30:31	0	MLU INPUT A MUX
va	3	32:34	0	INPUT BUFFER READ ADDRESS
ce	1	35	0,2	CONTEXT MEMORY READ/WRITE
ca	6	36:41	0,2	CONTEXT MEMORY ADDRESS
mtr	2	42:43	0	MLU INPUT VECTOR ROTATE

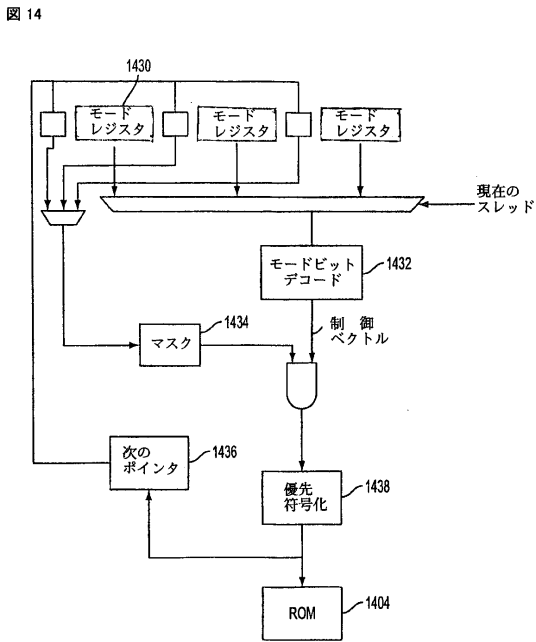
【図 12】



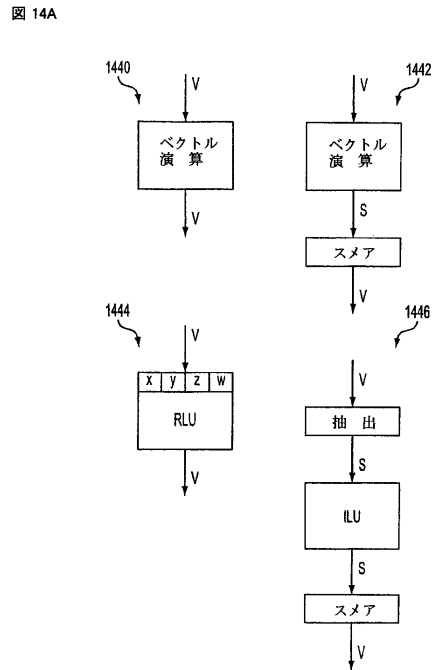
【図 13】



【図 14】

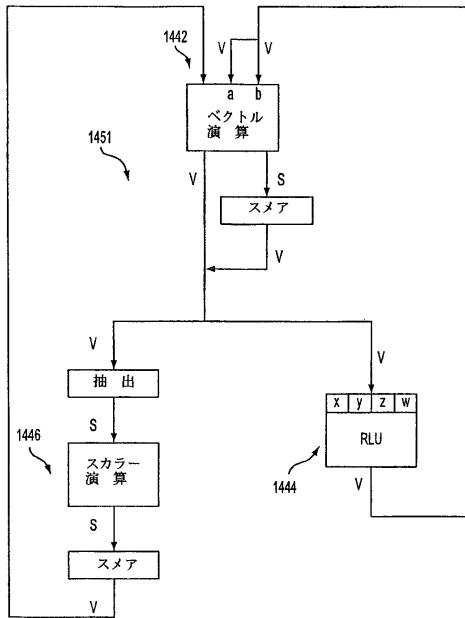


【図 14 A】



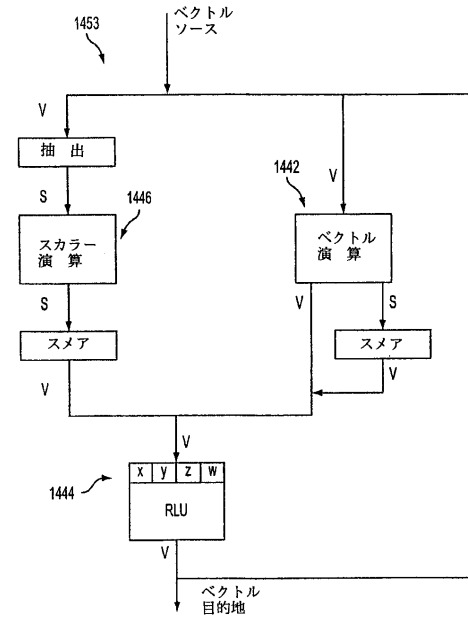
【図14B】

図14B



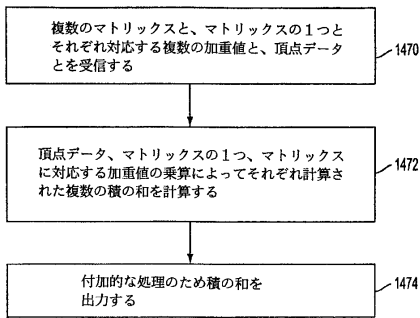
【図14C】

図14C



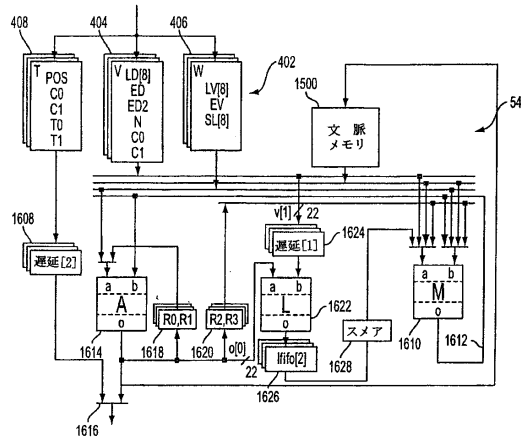
【図14D】

図14D



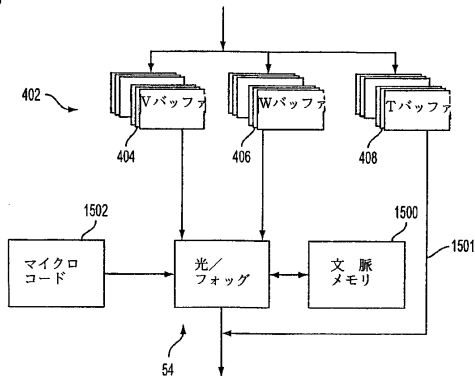
【図16】

図16

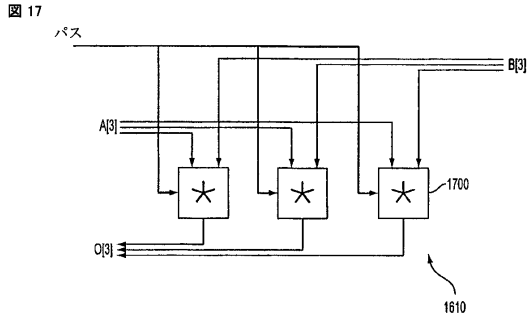


【図15】

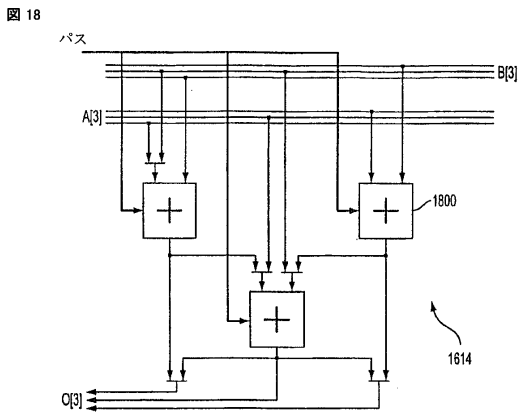
図15



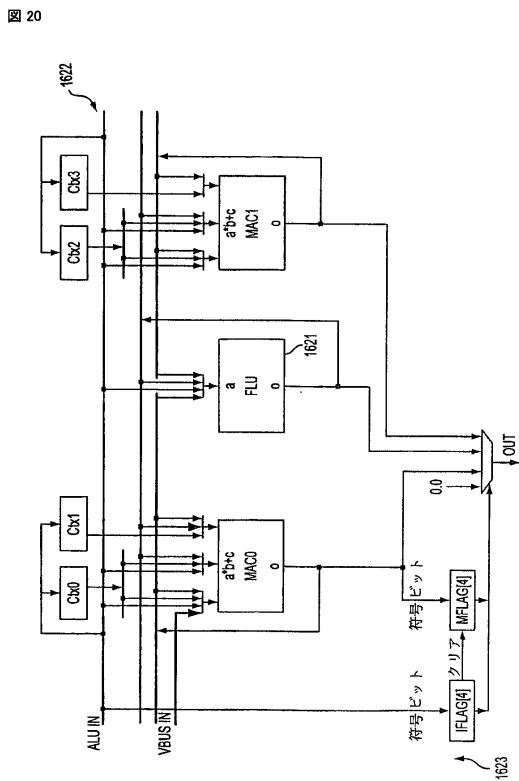
【 図 17 】



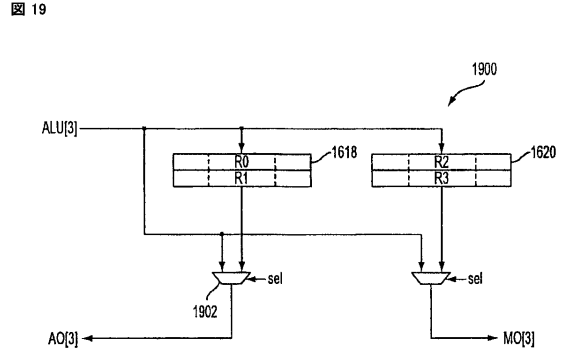
【 図 18 】



【 図 20 】



【 図 19 】



【 図 21 】

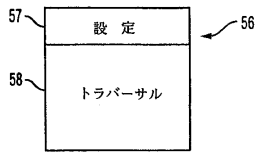
図 21

名称	レジスタ	記述
Z	IFLAG	CLEAR FLAG. SETS IFLAG AND MFLAG TO 0.
C	IFLAG	SPOTLIGHT CONE FLAG. SET IF VERTEX IS OUTSIDE SPOTLIGHT CONE.
S	IFLAG	SPECULAR2 FLAG. SET IF SPECULAR CONTRIBUTION IS NEGATIVE.
D	IFLAG	DIFFUSE FLAG. SET IF DIFFUSE TERM IS NEGATIVE.
M	MFLAG	
U	MFLAG	SPOTLIGHT CONE ATTENUATION FLAG. SET IF SPOTLIGHT CONE ATTENUATION CONTRIBUTION IS NEGATIVE.
T	MFLAG	SPECULAR FLAG. SET IF SPECULAR CONTRIBUTION IS NEGATIVE.
R	MFLAG	RANGE FLAG. SET IF VERTEX IS TOO FAR AWAY FROM THE LIGHT.



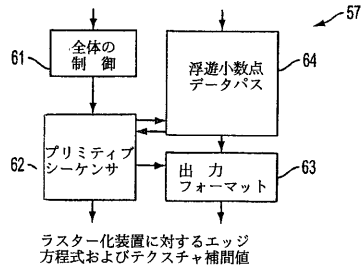
【図 25 B】

図 25B



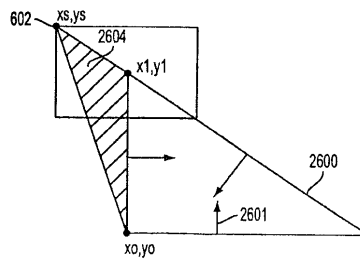
【図 26】

図 26



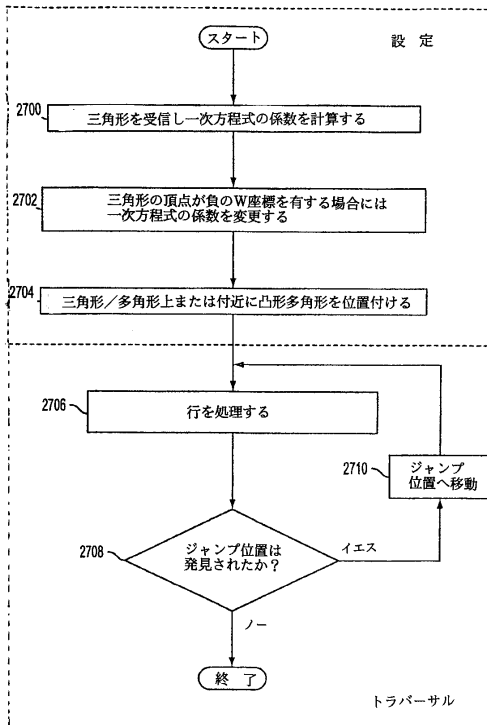
【図 26 A】

図 26A



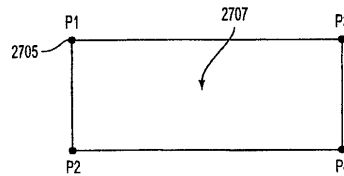
【図 27】

図 27



【図 27 A】

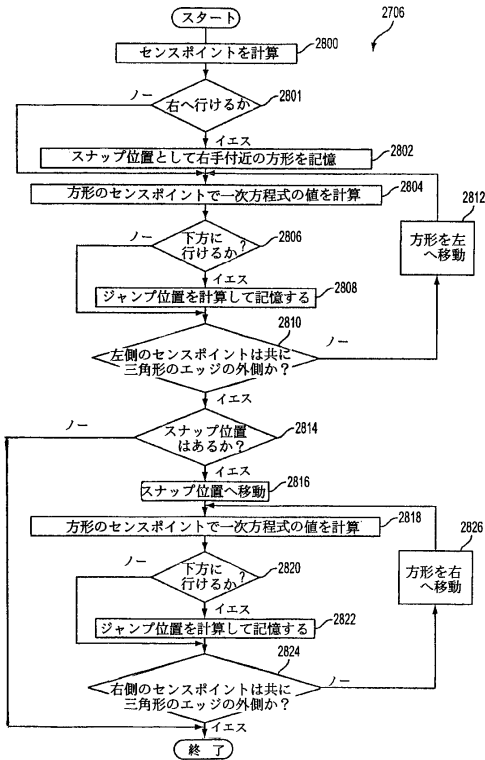
図 27A





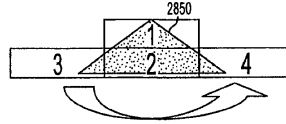
【図 28】

図 28



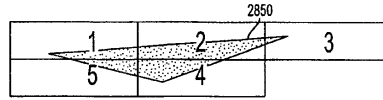
【図 28 A】

図 28A



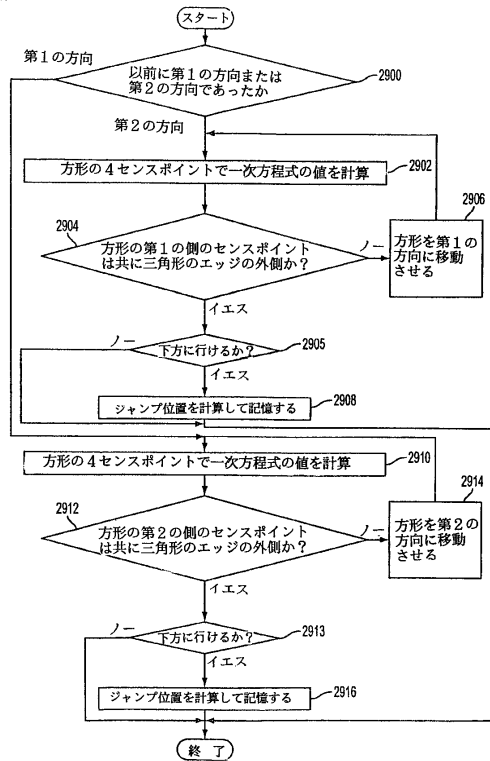
【図 28 B】

図 28B



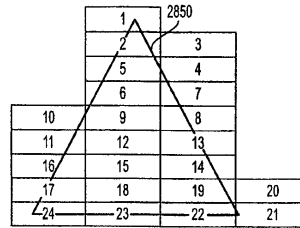
【図 29】

図 29



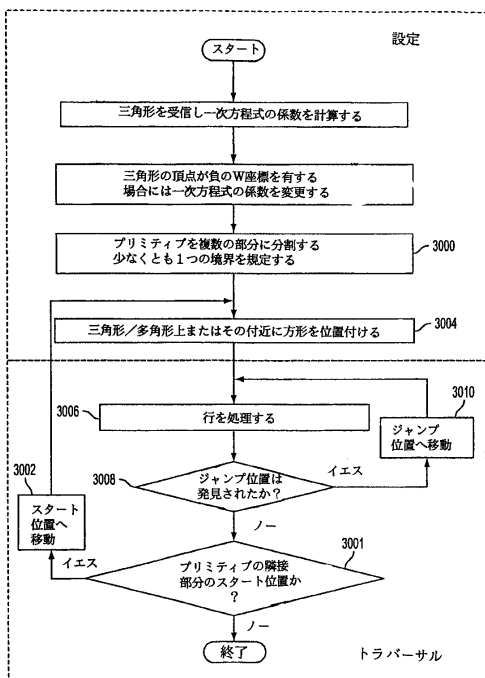
【図 29 A】

図 29A



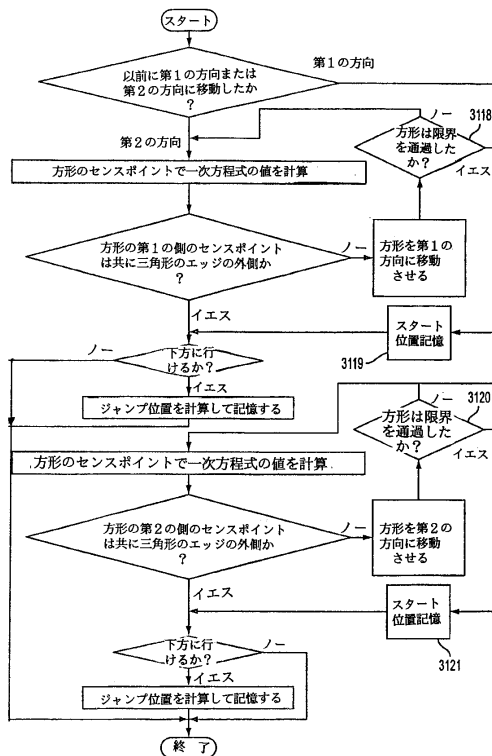
【 図 3 0 】

図 30



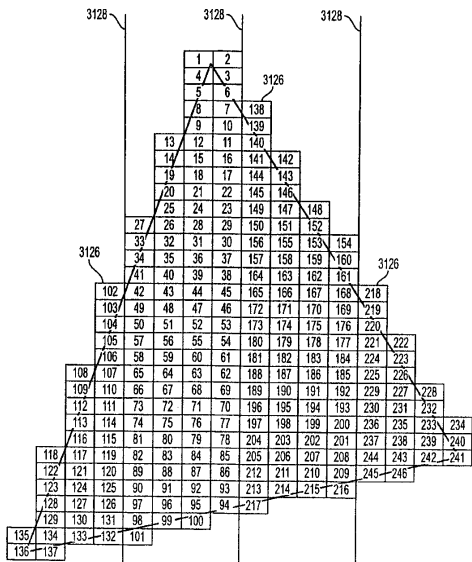
【 図 3 1 】

図 31



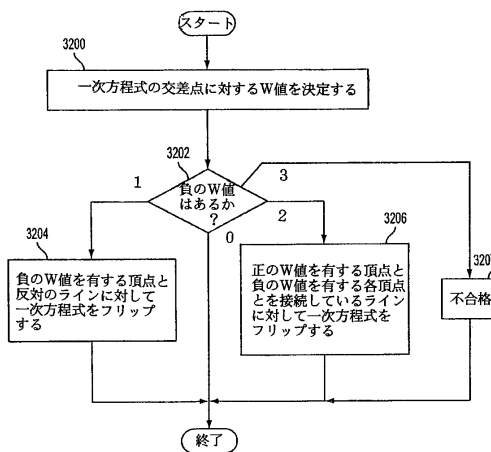
【 図 3 1 A 】

図 31A



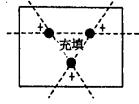
【 図 3 2 】

図 32



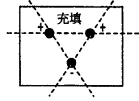
【 3 2 A 】

图 32A



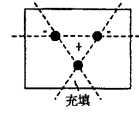
【 3 2 B 】

图 32B



【 3 2 C 】

图 32C



## フロントページの続き

- (72)発明者 サイモン・モイ  
アメリカ合衆国、カリフォルニア州 9 4 0 4 0 マウンテン・ビュー、フリーダム・レーン 1  
0 0
- (72)発明者 ケビン・ダワル  
アメリカ合衆国、カリフォルニア州 9 4 0 8 7 サニーベイル、ピーエムビー-447、イー・エル  
ル・カミノ・リアル 1 0 3 0
- (72)発明者 ミンジアン・ヤン  
アメリカ合衆国、カリフォルニア州 9 4 0 8 5 サニーベイル、ナンバー204、ダブリュ・オ  
リーブ・アベニュー 1 1 2 0
- (72)発明者 ジョン・モントリム  
アメリカ合衆国、カリフォルニア州 9 4 0 2 4 ロス・アルトス、ミラベール・アベニュー 1  
4 2 9
- (72)発明者 デビッド・カーク  
アメリカ合衆国、カリフォルニア州 9 4 1 2 3 サン・フランシスコ、プロデリック・ストリー  
ト 2 9 6 5
- (72)発明者 パオロ・サベラ  
アメリカ合衆国、カリフォルニア州 9 4 5 8 8 プレザントン、カンバーランド・ギャップ・コ  
ート 3 4 5 7
- (72)発明者 マシュー・ババキボス  
アメリカ合衆国、カリフォルニア州 9 4 3 0 3 パロ・アルト、エッジウッド・ドライブ 2 1  
2 9
- (72)発明者 ダグラス・ブーアハイズ  
アメリカ合衆国、カリフォルニア州 9 4 0 2 5 メンロ・パーク、ヒルビュー・ドライブ 1 3  
5 5
- (72)発明者 ニコラス・フォスケット  
アメリカ合衆国、カリフォルニア州 9 4 0 4 1 マウンテン・ビュー、ビュー・ストリート 2  
5 1

審査官 伊知地 和之

- (56)参考文献 特開平05-282464(JP,A)  
特開平07-249116(JP,A)  
特開平08-138060(JP,A)  
特開平10-161636(JP,A)  
特開平04-348485(JP,A)  
特開平02-183384(JP,A)

## (58)調査した分野(Int.Cl., DB名)

G 0 6 T 1 5 / 0 0 - 1 7 / 5 0  
G 0 9 G 5 / 0 0 - 5 / 4 0  
A 6 3 F 1 3 / 0 0  
C S D B (日本国特許庁)