(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3971448号
(P3971448)
(45)発行日　平成19年9月5日(2007.9.5)　(24)登録日　平成19年6月15日(2007.6.15)

(51)Int.Cl.　F I
*G06T 15/00 (2006.01)*　G06T 15/00 100A
G06T 15/00 300

請求項の数 2　(全 24 頁)

(21)出願番号　特願2006-301059(P2006-301059)
(22)出願日　平成18年11月7日(2006.11.7)
(62)分割の表示　特願2004-27663(P2004-27663)の分割
原出願日　平成8年2月6日(1996.2.6)
(65)公開番号　特開2007-26473(P2007-26473A)
(43)公開日　平成19年2月1日(2007.2.1)
審査請求日　平成18年11月7日(2006.11.7)

(73)特許権者　395015319
株式会社ソニー・コンピュータエンタテインメント
東京都港区南青山二丁目６番２１号
(74)代理人　110000198
特許業務法人湘洋内外特許事務所
(72)発明者　豊　禎治
東京都港区赤坂８丁目１番２２号　株式会社ソニー・コンピュータエンタテインメント内

審査官　月野　洋一郎

最終頁に続く

(54)【発明の名称】描画装置及び描画方法

(57)【特許請求の範囲】
【請求項１】
単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、描画手段により、単位図形の全ての画素の画素データを生成して、画像メモリに描画する描画装置であって、
上記画像メモリには、テクスチャデータが格納されていて、
上記描画命令に基づいて、
単位図形を３次元空間で複数に分割する手段と、
上記分割手段により分割された各単位図形に貼り付ける必要なテクスチャのアドレスを、該単位図形毎に順次、生成する前処理手段と、
上記前処理手段が生成した上記テクスチャのアドレスに基づいて、上記画像メモリから、該テクスチャのアドレスに対応するテクスチャデータを読み出す手段と、を有し、
上記描画手段は、上記読み出したテクスチャデータを用いて上記単位図形毎にテクスチャマッピングを行い、
上記分割する手段は、
単位図形の頂点の奥行きを示すＺ値の最小値と最大値との差が所定の範囲内に収まっているか否かにより、上記単位図形の分割を行うか否かを判定する判定手段を備え、
上記分割を行なう判定をした場合、上記描画命令に基づく単位図形を３次元空間で複数に分割し、
上記読み出す手段は、上記テクスチャデータの読み出しを、上記描画手段が行なう単位

図形毎のテクスチャマッピングに先立って完了するように行なうこと
を特徴とする描画装置。
【請求項２】
単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、単位図形の全ての画素の画素データを生成して、画像メモリに描画する描画方法であって、
上記描画命令に基づいて、
単位図形を３次元空間で複数に分割するステップと、
上記分割するステップにより分割された各単位図形に貼り付ける必要なテクスチャのアドレスを、該単位図形毎に順次、生成するステップと、
上記生成した上記テクスチャのアドレスに基づいて、上記画像メモリに予め格納されているテクスチャデータの中から対応するテクスチャデータを読み出すステップと、
上記読み出したテクスチャデータを用いて上記単位図形毎にテクスチャマッピングを行なうステップと、を有し、
上記分割するステップは、単位図形の頂点の奥行きを示すＺ値の最小値と最大値との差が所定の範囲内に収まっているか否かにより、上記単位図形の分割を行うか否かを判定し、
上記分割を行なう判定をした場合、上記描画命令に基づく単位図形を３次元空間で複数に分割し、
上記読み出すステップは、上記テクスチャデータの読み出しを、上記テクスチャマッピングを行なうステップに先立って完了するように行なうこと
を特徴とする描画方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、コンピュータを用いた映像機器であるグラフィックコンピュータ、特殊効果装置、ビデオゲーム機等に用いられる描画装置及び描画方法に関する。
【背景技術】
【０００２】
従来、家庭用ＴＶゲーム機やパーソナルコンピュータあるいはグラフィックコンピュータなどにおいて、テレビジョン受像機やモニタ受像機あるいは陰極線管（ＣＲＴ：Cathode Ray Tube）ディスプレイ装置などに出力して表示する画像のデータすなわち表示出力画像データを生成する画像生成装置では、央演算処理装置（ＣＰＵ： Central Processing Unit）とフレームバッファの間に専用の描画装置を設けることにより、高速処理を可能にしている。
【０００３】
すなわち、上記画像生成装置において、ＣＰＵ側では、画像を生成する際に、直接フレームバッファをアクセスするのではなく、座標変換やクリッピング、光源計算等のジオメトリ処理を行い、３角形や４角形などの基本的な単位図形（ポリゴン）の組み合わせとして３次元モデルを定義して３次元画像を描画するための描画命令を作成し、その描画命令を描画装置に送る。例えば、３次元のオブジェクトを表示する場合は、オブジェクトを複数のポリゴンに分解して、各ポリゴン対応する描画命令をＣＰＵから描画装置に転送する。そして、描画装置は、ＣＰＵから送られてきた描画命令を解釈して、頂点の色データと奥行きを示すＺ値から、ポリゴンを構成する全ての画素の色とＺ値を考慮して、画素データをフレームバッファに書き込むレンダリング処理を行い、フレームバッファに図形を描画する。なお、上記Ｚ値は、視点からの奥行き方向の距離を示す情報である。
【０００４】
例えば、上記画像生成装置において、３次元のオブジェクトを表示する場合は、オブジェクトを複数のポリゴンに分解して、各ポリゴンに対応する描画命令をＣＰＵから描画装置に転送する。この際に、オブジェクトをより実際に近く表現するために、テクスチャマ

ッピングやミップマッピングと呼ばれる手法が採用されている。さらに、色変換データを記憶したカラールックアップテーブル（ＣＬＵＴ： Color Lock Up Table）を介して画像の色データを変換することにより、表示色を変化させる手法も広く知られている。

【０００５】

ここで、テクスチャマッピングとは、テクスチャソース画像として別に用意された２次元画像（絵柄）すなわちテクスチャパターンを物体を構成するポリゴンの表面に張り付ける技術である。また、ミップマッピングは、３次元モデルに近づいたり、それから遠ざかった場合に、ポリゴンの張り付ける絵柄が不自然にならないように画素データを補間するようにしたテクスチャマッピングの手法の１つである。

【発明の開示】

【発明が解決しようとする課題】

【０００６】

ところで、画像の描画速度は、描画エンジンにおける各ポリゴンに対するテクスチャマッピングやミップマッピング等の処理速度に依存する。また、画像の描画速度は、描画エンジンからフレームバッファへの書き込み速度に影響され、フレームバッファのアクセス速度が遅いと描画速度が低下することになる。従って、描画速度を高めるために高価な高速メモリを大容量のフレームバッファに用いることはシステムの価格の高騰につながり、安価なダイナミックランダムアクセスメモリ（ＤＲＡＭ：Dynamic Random Access Memory）等のメモリを用いるとシステムの描画速度が遅くなる、という欠点がある。

【０００７】

そこで、本発明は、上述したような実情に鑑みてなされたものであり、次のような目的を有する。

【０００８】

すなわち、本発明の目的は、安価なＤＲＡＭ等のメモリをフレームバッファとして用いても、描画速度を高速に維持できるような描画装置及び描画方法を提供することにある。

【０００９】

また、本発明に他の目的は、単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、描画手段により、単位図形の全ての画素の画素データを効率よく生成して、画像メモリに描画することができる描画装置及び描画方法を提供することにある。

【００１０】

また、本発明に他の目的は、単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、描画手段により、テクスチャキャッシュ内のテクスチャデータに基づいてテクスチャマッピグ処理を確実且つ効率よく行うことができる描画装置及び描画方法を提供することにある。

【００１１】

また、本発明に他の目的は、単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、描画手段により処理するポリゴンの大きさすなわち画素数を均等化することができる描画装置及び描画方法を提供することにある。

【００１２】

また、本発明に他の目的は、単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、描画手段により、テクスチャキャッシュ内のテクスチャデータに基づいてテクスチャの歪みの少ない状態でテクスチャマッピグ処理を行うことができる描画装置及び描画方法を提供することにある。

【００１３】

また、本発明に他の目的は、単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、描画手段により、ミップマップテクスチャデータに基づいてミップマッピング処理を効率よく行うことができる描画装置及び描画方法を提供することにある。

【００１４】

また、本発明に他の目的は、前処理手段と描画手段とをパイプラインで構成して、単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、上記描画手段により、効率よく高速描画処理を行うことができる描画装置及び描画方法を提供することにある。

【００１５】

さらに、本発明に他の目的は、描画命令に基づくポリゴンをピクセルインターリーブ処理に適した形状の複数の新たなポリゴンに分割して、描画手段いより、フレームバッファを効率よくアクセスして高速の描画処理を行うことができる描画装置及び描画方法を提供することにある。

【課題を解決するための手段】

【００１６】

本発明は、単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、描画手段により、単位図形の全ての画素の画素データを生成して、画像メモリに描画する描画装置及び描画方法であって、前処理手段によりを単位図形を複数に分割することを特徴とする。

【００１７】

本発明に係る描画装置及び描画方法では、例えば、単位図形が上記描画手段におけるテクスチャキャシュ内に収まるか否かを判定する判定手段による判定結果に基づいて、分割した新たな単位図形が上記テクスチャキャシュ内に収まるように上記描画命令に基づく単位図形を複数に分割する。

【００１８】

また、本発明に係る描画装置及び描画方法では、例えば、単位図形内の画素数が規定値以下であるか否かを判定する判定手段による判定結果に基づいて、分割した新たな単位図形内の画素数が上記規定値以下となるように上記描画命令に基づく単位図形を２次元空間で複数に分割する。

【００１９】

また、本発明に係る描画装置及び描画方法では、例えば、上記描画命令に基づく単位図形を３次元空間で複数に分割する。

【００２０】

また、本発明に係る描画装置及び描画方法では、例えば、単位図形が参照するミップマップテクスチャの参照範囲を判定する判定手段による判定結果に基づい、分割した新たな単位図形が参照するミップマップテクスチャの参照範囲が所定範囲となるように上記描画命令に基づく単位図形を３次元空間で複数に分割する。

【００２１】

また、本発明に係る描画装置及び描画方法では、例えば、単位図形内の画素数が規定値以下であるか否かを判定する判定手段による判定結果に基づいて、分割した新たな単位図形内の画素数が上記規定値以下となるように上記描画命令に基づく単位図形を上記判定手段による判定結果に基づいて３次元空間で複数に分割する。

【００２２】

また、本発明に係る描画装置及び描画方法では、例えば、単位図形に対する上記描画手段による描画処理時間を予測して判定する判定手段による判定結果に基づいて、当該前処理手段による前処理時間と上記描画手段による描画処理時間がバランスするように上記描画命令に基づく単位図形を複数に分割する。

【００２３】

さらに、本発明に係る描画装置及び描画方法では、例えば、単位図形の形状を判定する判定手段による判定結果に基づいて、分割した新たな単位図形が所定形状に近づくように上記描画命令に基づく単位図形を複数に分割する。

【発明の効果】

【００２４】

以上のように、本発明に係る描画装置及び描画方法では、単位図形を複数に分割する前

処理手段を備えるので、描画手段により処理するのに適した状態に単位図形を分割することができ、単位図形の組合せにより定義された画像モデルを描画するための描画命令に基づいて、上記描画手段により、単位図形の全ての画素の画素データを効率よく生成して、画像メモリに描画することができる。

【００２５】

また、本発明に係る描画装置及び描画方法では、単位図形が描画手段におけるテクスチャキャシュ内に収まるか否かを判定する判定手段による判定結果に基づいて、前処理手段により、分割した新たな単位図形が上記テクスチャキャシュ内に収まるように描画命令に基づく単位図形を複数に分割するので、描画手段において、テクスチャキャッシュ内のテクスチャデータに基づいてテクスチャマッピグ処理を確実且つ効率よく行うことができる。

【００２６】

また、本発明に係る描画装置及び描画方法では、単位図形内の画素数が規定値以下であるか否かを判定する判定手段による判定結果に基づいて、前処理手段により、分割した新たな単位図形内の画素数が上記規定値以下となるように上記描画命令に基づく単位図形を２次元空間で複数に分割するので、描画手段において処理するポリゴンの大きさすなわち画素数を均等化することができる。

【００２７】

また、本発明に係る描画装置及び描画方法では、単位図形内の画素数が規定値以下であるか否かを判定する判定手段による判定結果に基づいて、前処理手段により、分割した新たな単位図形内の画素数が上記規定値以下となるように上記描画命令に基づく単位図形を３次元空間で複数に分割するので、例えばテクスチャキャッシュ内のテクスチャデータに基づいてテクスチャの歪みの少ない状態でテクスチャマッピグ処理を行うことができる。

【００２８】

また、本発明に係る描画装置及び描画方法では、単位図形が参照するミップマップテクスチャの参照範囲を判定する判定手段による判定結果に基づいて、前処理手段により、分割した新たな単位図形が参照するミップマップテクスチャの参照範囲が所定範囲となるように描画命令に基づく単位図形を３次元空間で複数に分割するので、ミップマップテクスチャデータに基づいてミップマッピング処理を効率よく行うことができる。

【００２９】

また、本発明に係る描画装置及び描画方法では、単位図形に対する描画手段による描画処理時間を予測して判定する判定手段による判定結果に基づいて、前処理手段による前処理時間と上記描画手段による描画処理時間がバランスするように描画命令に基づく単位図形を複数に分割するので、上記前処理手段と描画手段の各処理時間のバランスを保つことができ、上記前処理手段と描画手段とをパイプラインで構成して効率よく高速描画処理を行うことができる。

【００３０】

さらに、本発明に係る描画装置及び描画方法では、単位図形の形状を判定する判定手段判定結果に基づいて、前処理手段により、分割した新たな単位図形が所定形状に近づくように描画命令に基づく単位図形を複数に分割するので、上記描画命令に基づくポリゴンをピクセルインターリーブ処理に適した形状の複数の新たなポリゴンに分割することができる。これにより、描画手段で、フレームバッファを効率よくアクセスして高速の描画処理を行うことができる。

【発明を実施するための最良の形態】

【００３１】

以下、本発明の好ましい実施の形態について、図面を参照しながら説明する。

【００３２】

本発明に係る描画装置は、例えば図１に示すような構成のビデオゲーム装置に適用される。本発明に係る描画方法は、このビデオゲーム装置において実施される。

【００３３】

このビデオゲーム装置は、例えば光学ディスク等の補助記憶装置に記憶されているゲームプログラムを読み出して実行することにより、使用者からの指示に応じてゲームを行うものであって、図１に示すような構成を有している。

【００３４】

すなわち、このビデオゲーム装置は、２種類のバスすなわち、メインバス１とサブバス２を備える。

【００３５】

上記メインバス１とサブバス２は、バスコントローラ１０を介して接続されている。

【００３６】

そして、上記メインバス１には、マイクロプロセッサなどからなる主中央演算処理装置（メインＣＰＵ:Central Processing Unit）１１、ランダムアクセスメモリ（ＲＡＭ：Random Access Memory）からなる主記憶装置（メインメモリ）１２、主ダイナミックメモリアクセスメモリコントローラ（メインＤＭＡＣ： Dinamic Memory Access Controller）１３、ＭＰＥＧデコーダ（ＭＤＥＣ：MPEG Decorder）１４及び画像処理装置（ＧＰＵ：Graphic Processing Unit）１５が接続されている。また、上記サブバス２には、マイクロプロセッサなどからなる副中央演算処理装置（サブＣＰＵ： Central Processing Unit）２１、ランダムアクセスメモリ（ＲＡＭ： Random Access Memory)からなる副記憶装置（サブメモリ）２２、副ダイナミックメモリアクセスメモリコントローラ（サブＤＭＡＣ：Dinamic Memory Access Controller）２３、オペレーティングシステム等のプログラムが格納されたリードオンリーメモリ（ＲＯＭ： Read Only Memory）２４、音声処理装置（ＳＰＵ： Sound Processing Unit）２５、通信制御部（ＡＴＭ： Asynchronous Transimission mode）２６、補助記憶装置２７及び入力デバイス２８が接続されている。

【００３７】

上記バスコントローラ１０は、メインバス１とサブバス２との間のスイッチングを行う上記メインバス１上のデバイスであって、初期状態ではオープンになっている。

【００３８】

また、上記メインＣＰＵ１１は、上記メインメモリ１２上のプログラムで動作する上記メインバス１上のデバイスである。このメインＣＰＵ１１は、起動時には上記バスコントローラ１０がオープンになっていることにより、上記サブバス２上のＲＯＭ２４からブートプログラムを読み込んで実行し、補助記憶装置２７からアプリケーションプログラム及び必要なデータを上記メインメモリ１２や上記サブバス２上のデバイスにロードする。このメインＣＰＵ１１には、座標変換等の処理を行うジオミトリトランスファエンジン（ＧＴＥ： Geometry TransferEngine ）１７が搭載されている。上記ＧＴＥ１７は、例えば複数の演算を並列に実行する並列演算機構を備え、上記メインＣＰＵ１１からの演算要求に応じて座標変換，光源計算，行列あるいはベクトルなどの演算を高速に行う。そして、上記メインＣＰＵ１１は、上記ＧＴＥ１７による演算結果に基づいて３角形や４角形などの基本的な単位図形（ポリゴン）の組み合わせとして３次元モデルを定義して３次元画像を描画するための各ポリゴンに対応する描画命令を作成し、この描画命令をパケット化してコマンドパケットとして上記ＧＰＵ１５に送る。

【００３９】

また、上記メインＤＭＡＣ１３は、メインバス１上のデバイスを対象とするＤＭＡ転送の制御等を行う上記メインバス１上のデバイスである。このメインＤＭＡＣ１３は、上記バスコントローラ１０がオープンになっているときにはサブバス２上のデバイスも対象とする。

【００４０】

また、上記ＧＰＵ１５は、レンダリングプロセッサとして機能する上記メインバス１上のデバイスである。このＧＰＵ１５は、メインＣＰＵ１１又はメインＤＭＡＣ１３からコマンドパケットとして送られてきた描画命令を解釈して、頂点の色データと奥行きを示すＺ値から、ポリゴンを構成する全ての画素の色とＺ値を考慮して、画素データをフレームバッファ１８すなわち画像メモリに書き込むレンダリング処理を行う。

【００４１】
また、上記ＭＤＥＣ１４は、ＣＰＵと並列に動作可能なＩ／Ｏ接続デバイスであって、画像伸張エンジンとして機能する上記メインバス１上のデバイスである。このＭＤＥＣ１４は、離散コサイン変換などの直行変換により圧縮されて符号化された画像データを復号化する。
【００４２】
また、上記サブＣＰＵ２１は、上記サブメモリ２２上のプログラムで動作する上記サブバス２上のデバイスである。
【００４３】
また、上記サブＤＭＡＣ２３は、サブバス２上のデバイスを対象とするＤＭＡ転送の制御等を行う上記サブバス２上のデバイスである。このサブＤＭＡＣ２３は、上記バスコントローラ１０がクローズなっているときにのみバス権利を獲得することができる。
【００４４】
また、上記ＳＰＵ２５は、サウンドプロセッサとして機能する上記サブバス２上のデバイスである。このＳＰＵ２５は、上記サブＣＰＵ２１又はサブＤＭＡＣ２３からコマンドパケットとして送られてくるサウンドコマンドに応じて、サウンドメモリ２９から音声データ読み出して出力する。
【００４５】
また、上記ＡＴＭ２６は、サブバス２上の通信用デバイスである。
【００４６】
また、上記補助記憶装置２７は、サブバス２上のデータ入力デバイスであって、ディスクドライブなどからなる。
【００４７】
さらに、上記入力デバイス２８は、サブバス２上のコントロールパッド、マウスなどのマンマシンインターフェースや、画像入力、音声入力などの他の機器からの入力用デバイスである。
【００４８】
すなわち、このビデオゲーム装置では、座標変換やクリッピング、光源計算等のジオメトリ処理を行い、３角形や４角形などの基本的な単位図形（ポリゴン）の組み合わせとして３次元モデルを定義して３次元画像を描画するための描画命令を作成し、各ポリゴンに対応する描画命令をコマンドパケットとしてメインバス１に送出するジオメトリ処理系が上記メインバス１上のメインＣＰＵ１１及びＧＴＵ１７などにより構成され、上記ジオメトリ処理系からの描画命令に基づいて各ポリゴンの画素データを生成してフレームバッファ１８に書き込むレンダリング処理を行い、フレームバッファ１８に図形を描画するレンダリング処理系が上記ＧＰＵ１５により構成されている。
【００４９】
以下、上述したＧＰＵ１５について具体的に説明する。
【００５０】
上記ＧＰＵ１５は、その具体的な構成を図２に示してあるように、上記メインバス１に接続されたパケットエンジン３１を備え、上記メインＣＰＵ１１又はメインＤＭＡＣ１３から上記メインバス１を介して上記パケットエンジン３１にコマンドパケットとして送られてくる描画命令に従って、プリプロセッサ３２と描画エンジン３３により各ポリゴンの画素データを上記フレームバッファ１８に書き込むレンダリング処理を行い、上記フレームバッファ１８に描画された画像の画素データを読み出して表示制御部（ＣＲＴＣ： CRT Controler）３４を介してビデオ信号として図示しないテレビジョン受像機やモニタ受像機に供給するようになっている。
【００５１】
上記パケットエンジン３１は、上記メインＣＰＵ１１又はメインＤＭＡＣ１３から上記メインバス１を介して送られてくるコマンドパケットを上記パケットエンジン３１により図示しないレジスタ上に展開する。

【００５２】
また、上記プリプロセッサ３２は、上記パケットエンジン３１にコマンドパケットとして送られてきた描画命令に従ってポリゴンデータを生成して後述するポリゴンの分割処理などの所定の前処理をポリゴンデータに施し、上記描画エンジン３３が必要とする各ポリゴンの頂点座標情報、テクスチャやミップマップテクスチャのアドレス情報、ピクセルインターリーブの制御情報などの各種データを生成する。
【００５３】
さらに、上記描画エンジン３３は、上記プリプロセッサ３２に接続されたＮ個のポリゴンエンジン３３Ａ１，３３Ａ２・・・３３ＡＮと、各ポリゴンエンジン３３Ａ１，３３Ａ２・・・３３ＡＮに接続されたＮ個のテクスチャエンジン３３Ｂ１，３３Ｂ２・・・３３ＢＮと、各テクスチャエンジン３３Ｂ１，３３Ｂ２・・・３３ＢＮに接続された第１のバススイッチャ３３Ｃと、この第１のバススイッチャ３３Ｃに接続されたＭ個のピクセルエンジン３３Ｄ１，３３Ｄ２・・・３３ＤＭと、各ピクセルエンジン３３Ｄ１，３３Ｄ２・・・３３ＤＭに接続された第２のバススイッチャ３３Ｅと、この第２のバススイッチャ３３Ｅに接続されたテクスチャキャッシュ３３Ｆと、このテクスチャキャッシュ３３Ｆに接続されたＣＬＵＴキャッシュ３３Ｇを備える。
【００５４】
この描画エンジン３３において、上記Ｎ個のポリゴンエンジン３３Ａ１，３３Ａ２・・・３３ＡＮは、上記プリプロセッサ３２により前処理が施されたポリゴンデータに基づいて、上記Ｎ個のポリゴンエンジン３３Ａ１，３３Ａ２・・・３３ＡＮは、描画命令に応じたポリゴンを順次生成してポリゴン毎にシェーディング処理などを並列処理により行う。
【００５５】
また、上記Ｎ個のテクスチャエンジン３３Ｂ１，３３Ｂ２・・・３３ＢＮは、上記ポリゴンエンジン３３Ａ１，３３Ａ２・・・３３ＡＮにより生成されたポリゴン毎に、上記テクスチャキャッシュ３３Ｆからカラールックアップテーブル（ＣＬＵＴ：Color Lock Up Table)キャッシュ３３Ｇを介して与えられるテクスチャデータに基づいて、テクスチャマッピング処理やミップマップ処理を並列処理により行う。
【００５６】
ここで、上記テクスチャキャッシュ３３Ｆには、上記Ｎ個のテクスチャエンジン３３Ｂ１，３３Ｂ２・・・３３ＢＮが処理するポリゴンに張り付けるテクスチャやミップマップテクスチャのアドレス情報が上記プリプロセッサ３２から事前に与えられ、上記アドレス情報に基づいて上記フレームバッファ１８上のテクスチャ領域からテクスチャマッピング処理に必要なテクスチャデータが転送されるとともに、該当するテクスチャデータからミップマッピング処理に必要となる解像度のデータのみが選択されてミップマップテクスチャデータとして転送される。さらに、上記ＣＬＵＴキャッシュ３３Ｇには、上記ポリゴンの描画を行なう際に参照すべきＣＬＵＴデータが上記フレームバッファ１８上のＣＬＵＴ領域から転送される。
【００５７】
上記Ｎ個のテクスチャエンジン３３Ｂ１，３３Ｂ２・・・３３ＢＮによりテクスチャマッピング処理やミップマップ処理が施されたポリゴンデータは、上記第１のバススイッチャ３３Ｃを介してＭ個のピクセルエンジン３３Ｄ１，３３Ｄ２・・・３３ＤＭに転送される。
【００５８】
上記Ｍ個のピクセルエンジン３３Ｄ１，３３Ｄ２・・・３３ＤＭは、Ｚバッファ処理やアンチエリアシング処理等の各種画像処理を並列処理により行い、Ｍ個の画素データを生成する。
【００５９】
そして、上記Ｍ個のピクセルエンジン３３Ｄ１，３３Ｄ２・・・３３ＤＭで生成されたＭ個の画素データは、この第２のバススイッチャ３３Ｅを介して上記フレームバッファ１８に書き込まれる。

【００６０】
ここで、上記第２のバススイッチャ３３Ｅは、上記プリプロセッサ３２からピクセルインターリーブの制御情報が供給されており、上記Ｍ個のピクセルエンジン３３Ｄ１，３３Ｄ２・・・３３ＤＭで生成されたＭ個の画素データのうちのＬ個の画素データを上記制御情報に基づいて選択することにより、上記フレームバッファ１８上に描画するポリゴンの形状に応じたＭ個の記憶場所をアクセス単位として画素データをＭ個づつ書き込むピクセルインターリーブ処理を行う機能を有している。
【００６１】
上記描画エンジン３３は、上記プリプロセッサ３２により前処理が施されたポリゴンデータに基づいて、各ポリゴンの全ての画素データを生成して上記フレームバッファ１８に書き込むことにより、上記描画命令によりポリゴンの組合せとして定義された画像を上記フレームバッファ１８上に描画する。そして、上記フレームバッファ１８に描画された画像の画素データを読み出してＣＲＴＣ３４を介してビデオ信号として図示しないテレビジョン受像機やモニタ受像機に供給する。
【００６２】
このような構成のＧＰＵ１５において、上記プリプロセッサ３２は、例えば、ポリゴンの頂点座標［（Ｘ０，Ｙ０），（Ｘ１，Ｙ１），（Ｘ２，Ｙ２）］やテクスチャ座標［（Ｕ０，Ｖ０），（Ｕ１，Ｖ１），（Ｕ２，Ｖ２）］に基づいて、上記Ｎ個のテクスチャエンジン３３Ｂ１，３３Ｂ２・・・３３ＢＮが処理するポリゴンに張り付けるテクスチャの先読みを行うためのアドレス情報を生成し、また、ポリゴンの辺の傾き［（Ｘ１－Ｘ０）／（Ｙ１－Ｙ０），（Ｘ２－Ｘ０）／（Ｙ２－Ｙ０），（Ｘ１－Ｘ２）／（Ｙ１－Ｙ２）］、テクスチャアドレスの傾き［（Ｕ１－Ｕ０）／（Ｙ１－Ｙ０），（Ｕ２－Ｕ０）／（Ｙ２－Ｙ０），（Ｕ１－Ｕ２）／（Ｙ１－Ｙ２）］，［（Ｖ１－Ｖ０）／（Ｙ１－Ｙ０），（Ｖ２－Ｖ０）／（Ｙ２－Ｙ０），（Ｖ１－Ｖ２）／（Ｙ１－Ｙ２）］・・・やポリゴンの面積などからミップマップの選択情報を再生して、これらの情報をテクスチャキャッシュ３３Ｆに供給する。また、ポリゴンの頂点座標［（Ｘ０，Ｙ０），（Ｘ１，Ｙ１），（Ｘ２，Ｙ２）］を左エッジの頂点順（Ｘ０，Ｙ０）→（Ｘ１，Ｙ１）→（Ｘ２，Ｙ２）又は右エッジの頂点順（Ｘ２，Ｙ２）→（Ｘ１，Ｙ１）→（Ｘ０，Ｙ０）でソーティングしたり、両端点のスキャンやテクスチャアドレスのスキャンを行う。
【００６３】
そして、上記プリプロセッサ３２は、ポリゴンデータを前処理した情報を図示しないワークメモリに蓄えておき、描画エンジン３３が次のポリゴンを処理できるようになった段階で、１ポリゴンを処理できる情報をワークメモリから上記Ｎ個のポリゴンエンジン３３Ａ１，３３Ａ２・・・３３ＡＮに転送する。これにより、上記描画エンジン３３は、新たなポリゴンの描画処理を開始する。
【００６４】
すなわち、このＧＰＵ１５では、その基本的な構成を図３に示すように、上記プリプロセッサ３２と描画エンジン３３でパイプラインにより描画処理を行い、描画命令によりポリゴンの組合せとして定義された画像を上記フレームバッファ１８上に描画する。
【００６５】
このパイプライン処理による描画処理を再度説明する。
【００６６】
上記プリプロセッサ３２は、上述のようにポリゴンデータに所定前処理を施し、上記描画エンジン３３が必要とする各ポリゴンの頂点座標情報、テクスチャやミップマップテクスチャのアドレス情報、ピクセルインターリーブの制御情報などの各種データを上記描画エンジン３３に供給する。
【００６７】
上記描画エンジン３３は、上記プリプロセッサ３２からデータを受け取り、必要とするテキスチャデータをテキスチャキャッシュ３３Ｄから読み出し、画素データを生成して上記フレームバッファ１８に書き込む。上記テキスチャキャッシュ３３Ｄは、上記プリプロ

セッサ３２における前処理により算出された必要とするテクスチャアドレスに対応するテクスチャ領域のテキスチャデータを上記フレームバッファ１８から読み出す。テキスチャデータの読み出しは、描画エンジン３３が実際に必要とする前に完了するように行われる。また、ミップマッピング処理で必要とする解像度に対応するテクスチャデータのみを上記テクスチャ領域から読み込むことにより、上記テクスチャ領域のアクセス回数を減らすことができる。

【００６８】

なお、上記テキスチャキャッシュ３３Ｆ内のデータ構造は、その一例を図４に示してあるように、テクスチャアドレスからなるタグ部ＴＡＧ、必要となるテクスチャデータが格納されておいる格納部ＤＡＴＡ、未だテクスチャデータが使用されていないことを示すフラグＬを有する。そして、上記テキスチャキャッシュ３３は、フラグＬがリセットされたエントリを使用すべく、上記フレームバッファ１８のテクスチャ領域からテクスチャデータを読み込み、そのフラグＬをセットする。描画エンジン３３は、フラグＬがセットされているエントリから該当するテクスチャデータを読み出して描画処理を行い、描画を終了してそのテクスチャデータをもはや必要としなくなった段階でそのエントリのフラグｌをリセットする。

【００６９】

このようにテクスチャマッピング処理を行う描画装置において、プリプロセッサ３２と描画エンジン３３をパイプラインで構成し、テクスチャメモリすなわち上記フレームバッファ１８上のテキスチャ領域から上記描画エンジン３３が必要とするテキスチャデータを上記プリプロセッサ３２による前処理の段階でキャッシュメモリ３３Ｆに転送することによって、上記描画エンジン３３を停止させることなく描画処理を行うことができる。また、ミップマッピング処理で必要とする解像度に対応するテクスチャデータのみを上記テクスチャ領域から読み込むことにより、上記テクスチャ領域のアクセス回数及びアクセス時間を減らすことができ、全体の描画速度を上げることができる。

【００７０】

なお、上記プリプロセッサ３２におけるポリゴンの分割処理は、例えば図５に示すフローチャートに従って行われる。

【００７１】

すなわち、ポリゴンの分割処理は、ポリゴンの数を示すポリゴンカウントＣを１に初期設定して開始される。

【００７２】

そして、第１の処理ステップＳ１では、ポリゴンを分割する必要があるか否かの判定処理を行う。この処理ステップＳ１における判定処理では、例えば、描画エンジン３３においてこれから処理するポリゴンがテクスチャキャッシュ３３Ｆ内に収まる否かを判定する。この判定処理は、例えばポリゴンの頂点のテクスチャ座標［（Ｕ０，Ｖ０），（Ｕ１，Ｖ１），（Ｕ２，Ｖ２）］を算出して、全てが１テクスチャページ内に収まっているか否かを判定すればよい。

【００７３】

そして、上記処理ステップＳ１における判定結果が「ＮＯ」すなわちポリゴンを分割する必要がある場合には、次の処理ステップＳ２に進んで、ポリゴンのＮ分割処理を行う。この処理ステップＳ２におけるポリゴンのＮ分割処理は、例えば次に示すように、ポリゴンの全ての辺を中点で分割することにより行われる。

【００７４】

Ｘ０'＝（Ｘ０＋Ｘ１）／２
Ｙ０'＝（Ｙ０＋Ｙ１）／２
Ｚ０'＝（Ｚ０＋Ｚ１）／２
Ｘ１'＝（Ｘ１＋Ｘ２）／２
Ｙ１'＝（Ｙ１＋Ｙ２）／２
Ｚ１'＝（Ｚ１＋Ｚ２）／２

Ｘ２'＝（Ｘ２＋Ｘ０）／２
Ｙ２'＝（Ｙ２＋Ｙ０）／２
Ｚ２'＝（Ｚ２＋Ｚ０）／２
Ｕ０'＝（Ｕ０＋Ｕ１）／２
Ｖ０'＝（Ｖ０＋Ｖ１）／２
Ｚ０'＝（Ｚ０＋Ｚ１）／２
Ｕ１'＝（Ｕ１＋Ｕ２）／２
Ｖ１'＝（Ｖ１＋Ｖ２）／２
Ｚ１'＝（Ｚ１＋Ｚ２）／２
Ｕ２'＝（Ｕ２＋Ｕ０）／２
Ｖ２'＝（Ｖ２＋Ｖ０）／２
Ｚ２'＝（Ｚ２＋Ｚ０）／２
Ｒ０'＝（Ｒ０＋Ｒ１）／２
Ｇ０'＝（Ｇ０＋Ｇ１）／２
Ｂ０'＝（Ｂ０＋Ｂ１）／２
Ｒ１'＝（Ｒ１＋Ｒ２）／２
Ｇ１'＝（Ｇ１＋Ｇ２）／２
Ｂ１'＝（Ｂ１＋Ｂ２）／２
Ｒ２'＝（Ｒ２＋Ｒ０）／２
Ｇ２'＝（Ｇ２＋Ｇ０）／２
Ｂ２'＝（Ｂ２＋Ｂ０）／２

すなわち、この処理ステップＳ２におけるポリゴンのＮ分割処理では、ポリゴンの全ての辺を中点で分割することにより、例えば三角形のポリゴンはＮ＝４個の新たなポリゴンに分割される。

【００７５】

次の処理ステップＳ２では、ポリゴンカウントＣをＣ＝Ｃ＋Ｎ－１としてポリゴンの数を変更する。そして、最初の処理ステップＳ１に戻り、分割された新たなポリゴンをさらに分割する必要があるか否かの判定処理を行い、分割した新たなポリゴンが上記テクスチャキャシュ内に収まるようになるまで、上記各処理ステップＳ１～Ｓ３を繰り返し行う。

【００７６】

また、上記処理ステップＳ１における判定結果が「ＹＥＳ」すなわちポリゴンを分割する必要がない場合には次の処理ステップＳ４に進む。

【００７７】

この処理ステップＳ４では、ポリゴンエンジン３３Ａ１，３３Ａ２・・・３３ＡＮＰに１ポリゴン分の前処理情報を渡して、レンダリング処理を開始させ、レンダリング処理の終了を待つことなく次の処理ステップＳ５に進む。

【００７８】

この処理ステップＳ５では、ポリゴンカウントＣをデクリメントする。

【００７９】

次の処理ステップＳ６では、ポリゴンカウントＣが「０」になったか否かの判定処理を行う。そして、この処理ステップＳ６における判定結果が「ＮＯ」すなわちＣ≠０で処理すべきポリゴンがある場合には最初の処理ステップＳ１に戻って、次のポリゴンの処理に入る。また、この処理ステップＳ６における判定結果が「ＹＥＳ」すなわち全てのポリゴンをレンダリングして分割すべきポリゴンが無くなれば、処理を終了する。

【００８０】

すなわち、上記プリプロセッサ３２では、描画エンジン３３においてこれから処理するポリゴンがテクスチャキャッシュ３３Ｆ内に収まる否か（以下、判定条件１という）を判定し、その判定結果に基づいて分割処理を行うことによって、分割した新たなポリゴンが上記テクスチャキャシュ３３Ｆ内に収まるように上記描画命令に基づくポリゴンを複数に分割する。これにより、上記描画エンジン３３において、テクスチャキャッシュ３３Ｆか

らＣＬＵＴチャッシュ３３Ｇを介して読み出されるテクスチャデータに基づいてテクスチャマッピグ処理を確実且つ効率よく行うことができる。

【００８１】

ここで、上記プリプロセッサ３２におけるポリゴンの分割処理では、上述の最初の処理ステップＳ１においてポリゴン内の画素数が規定値以下であるか否か（以下、判定条件２という）によりポリゴンを分割する必要があるか否かを判定し、その判定結果に基づいて、分割した新たなポリゴン内の画素数が上記規定値以下となるように処理ステップＳ２で上記描画命令に基づくポリゴンを２次元空間で複数に分割しても良い。これにより、上記描画エンジン３３において処理するポリゴンの大きさすなわち画素数を均等化することができる。なお、上記ポリゴン内の画素数は、例えば、そのポリゴンの頂点の外積値として面積を求め、その値が適正な値よりも小さいか否かにより判定することができる。

【００８２】

また、上記プリプロセッサ３２におけるポリゴンの分割処理では、上述の処理ステップＳ２において上記描画命令に基づくポリゴンを３次元空間で複数に分割するようにしても良い。

【００８３】

この場合、上述の処理ステップＳ１において、ポリゴンの頂点のＺ値の最小値と最大値との差が適正な範囲内に収まっているか否か（以下、判定条件３という）により、ポリゴンを分割する必要があるか否かを判定し、その判定結果に基づいて、分割した新たなポリゴン内の画素数が上記規定範囲内に収まるように上記処理ステップＳ２で上記描画命令に基づくポリゴンを３次元空間で複数に分割して、１ポリゴンの大きさを制限することによって、テクスチャキャッシュ３３ＦからＣＬＵＴチャッシュ３３Ｇを介して読み出されるテクスチャデータに基づいてテクスチャの歪みの少ない状態でテクスチャマッピグ処理を行うことができる。

【００８４】

また、この場合、上述の処理ステップＳ１において、ポリゴンの頂点のＺ値の最小値と最大値で参照するミップマップテキスチャを跨いでいるか否か（以下、判定条件４という）によりポリゴンを分割する必要があるか否かを判定し、その判定結果に基づいて、分割した新たなポリゴンがミップマップテキスチャを跨がないように、上記処理ステップＳ２で上記描画命令に基づくポリゴンを３次元空間で複数に分割して、１ポリゴンの参照するミップマップテキスチャの参照範囲を制限を制限することによって、テクスチャキャッシュ３３ＦからＣＬＵＴチャッシュ３３Ｇを介して読み出されるミップマップテクスチャデータに基づいてミップマッピング処理を効率よく行うことができる。

【００８５】

さらに、この場合、上述の処理ステップＳ１において、ポリゴン内の画素数が規定値以下であるか否かにより、ポリゴンを分割する必要があるか否かを判定し、その判定結果に基づいて、分割した新たなポリゴン内の画素数が上記規定値以下となるように上記描画命令に基づくポリゴンを上記処理ステップＳ２により３次元空間で複数に分割するようにしても良い。

【００８６】

また、上述の処理ステップＳ１において、ポリゴンに対する描画エンジン３３に描画処理時間を例えばポリゴン内の画素数に基づいて予測し、当該プリプロセッサ３２による前処理時間と上記描画エンジン３３による描画処理時間がバランスいるか否か（以下、判定条件５という）により、ポリゴンを分割する必要があるか否かを判定し、その判定結果に基づいて、当該プリプロセッサ３２による前処理時間と上記描画エンジン３３による描画処理時間がバランスするように上記描画命令に基づくポリゴンを上記処理ステップＳ２で複数に分割するようにしても良い。これにより、上記プリプロセッサ３２と描画エンジン３３の各処理時間のバランスを保ち、上記プリプロセッサ３２と描画エンジン３３とパイプラインで構成して効率よく高速描画処理を行うことができる。

【００８７】

また、上述の処理ステップＳ１において、描画エンジン３３で処理するポリゴンがピクセルインターリーブ処理に適した形状であるか否か（以下、判定条件６という）により、ポリゴンを分割する必要があるか否かを判定し、その判定結果に基づいて、上記処理ステップＳ２により上記描画命令に基づくポリゴンをピクセルインターリーブ処理に適した形状の複数の新たなポリゴンに分割するようにしても良い。これにより、描画エンジン３３でフレームバッファ１８を効率よくアクセスして高速の描画処理を行うことができる。

【００８８】

さらに、上述の処理ステップＳ１において、上述の各種判定条件を組み合わせてポリゴンを分割する必要があるか否かを判定して、その判定結果に基づいて、分割した新たなポリゴンが各種判定条件を満たすように上記描画命令に基づくポリゴンを上記処理ステップＳ２により複数に分割するようにしても良い。

【００８９】

上述の各種判定条件を組み合わせとしては、例えば、描画エンジン３３においてテキスチャマッピングを行う場合には、上記判定条件１と他の判定条件２～判定条件６との組み合わせが採用される。

【００９０】

すなわち、上述の処理ステップＳ１において、上記判定条件１と判定条件２を組み合わせてポリゴンを分割する必要があるか否かを判定して、その判定結果に基づいて、分割した新たなポリゴンが上記判定条件１と判定条件２を満たすように上記描画命令に基づくポリゴンを上記処理ステップＳ２により複数に分割することにより、上記描画エンジン３３において処理するポリゴンの大きさすなわち画素数を均等化し、上記テクスチャキャッシュ３３ＦからＣＬＵＴチャッシュ３３Ｇを介して読み出されるテクスチャデータに基づいてテクスチャマッピグ処理を確実且つ効率よく行うことができる。

【００９１】

また、上述の処理ステップＳ１において、上記判定条件１と判定条件３を組み合わせてポリゴンを分割する必要があるか否かを判定して、その判定結果に基づいて、分割した新たなポリゴンが上記判定条件１と判定条件３を満たすように上記描画命令に基づくポリゴンを上記処理ステップＳ２により複数に分割することにより、上記テクスチャキャッシュ３３ＦからＣＬＵＴチャッシュ３３Ｇを介して読み出されるテクスチャデータに基づいてテクスチャの歪みの少ない状態でテクスチャマッピグ処理を確実且つ効率よく行うことができる。さらに、上記判定条件２を組み合わせるようにすれば、上記描画エンジン３３において処理するポリゴンの大きさすなわち画素数を均等化して、テクスチャマッピグ処理を行うことができる。

【００９２】

また、上述の処理ステップＳ１において、上記判定条件１と判定条件４を組み合わせてポリゴンを分割する必要があるか否かを判定して、その判定結果に基づいて、分割した新たなポリゴンが上記判定条件１と判定条件４を満たすように上記描画命令に基づくポリゴンを上記処理ステップＳ２により複数に分割することにより、上記テクスチャキャッシュ３３ＦからＣＬＵＴチャッシュ３３Ｇを介して読み出されるテクスチャデータに基づいて、ミップマッピング処理を確実且つ効率よく行うことができる。さらに、上記判定条件２や判定条件３を組み合わせて、上記描画エンジン３３において処理するポリゴンの大きさすなわち画素数を均等化したり、テクスチャの歪みを軽減するようにしても良い。

【００９３】

また、上述の処理ステップＳ１において、上記判定条件１と判定条件５を組み合わせてポリゴンを分割する必要があるか否かを判定して、その判定結果に基づいて、分割した新たなポリゴンが上記判定条件１と判定条件５を満たすように上記描画命令に基づくポリゴンを上記処理ステップＳ２により複数に分割することにより、上記プリプロセッサ３２と描画エンジン３３の各処理時間のバランスを保ちパイプラインで効率よく高速のテクスチャマッピグ処理を行うことができる。さらに、上記判定条件２や判定条件３を組み合わせて、上記描画エンジン３３において処理するポリゴンの大きさすなわち画素数を均等化し

たり、テクスチャの歪みを軽減するようにしても良い。上記判定条件４を組み合わせて、ミップマッピング処理を行うようにしても良い。

【００９４】

さらに、上述の処理ステップＳ１において、上記判定条件１と判定条件６を組み合わせてポリゴンを分割する必要があるか否かを判定して、その判定結果に基づいて、分割した新たなポリゴンが上記判定条件１と判定条件６を満たすように上記描画命令に基づくポリゴンを上記処理ステップＳ２により複数に分割することにより、描画エンジン３３でテクスチャマッピグ処理を確実且つ効率よく行ＵＴＯともに、フレームバッファ１８を効率よくアクセスして高速の描画処理を行うことができる。さらに、上記判定条件２や判定条件３を組み合わせて、上記描画エンジン３３において処理するポリゴンの大きさすなわち画素数を均等化したり、テクスチャの歪みを軽減するようにしても良い。上記判定条件４を組み合わせてミップマッピング処理を行うようにしたり、上記判定条件５を組み合わせてパイプラインによる高速化を図るようにしても良い。

【００９５】

また、描画エンジン３３においてテキスチャマッピングを行わない場合には、上記判定条件２、判定条件５、判定条件６の組み合わせが上述の各種判定条件を組み合わせとして採用される。

【００９６】

すなわち、上述の処理ステップＳ１において、上記判定条件２と判定条件５を組み合わせてポリゴンを分割する必要があるか否かを判定して、その判定結果に基づいて、分割した新たなポリゴンが上記判定条件２と判定条件５を満たすように上記描画命令に基づくポリゴンを上記処理ステップＳ２により複数に分割することにより、上記描画エンジン３３において処理するポリゴンの大きさすなわち画素数を均等化して、上記プリプロセッサ３２と描画エンジン３３の各処理時間のバランスを保ちパイプラインで効率よく高速の描画処理を行うことができる。

【００９７】

また、上述の処理ステップＳ１において、上記判定条件２と判定条件６を組み合わせてポリゴンを分割する必要があるか否かを判定して、その判定結果に基づいて、分割した新たなポリゴンが上記判定条件２と判定条件６を満たすように上記描画命令に基づくポリゴンを上記処理ステップＳ２により複数に分割することにより、上記描画エンジン３３において処理するポリゴンの大きさすなわち画素数を均等化し、フレームバッファ１８を効率よくアクセスして高速の描画処理を行うことができる。さらに、上記判定条件５を組み合わせてパイプラインによる高速化を図るようにしても良い。

【００９８】

さらに、上述した第２のバススイッチャ３３Ｅにおけるピクセルインターリーブ処理は、次のようにして行われる。

【００９９】

すなわち、第２のバススイッチャ３３Ｅは、図６に示すように、上記図２に示したプリプロセッサ３２の出力が供給される制御回路１０１と、制御回路１０１の出力が供給されるセレクタ１０２と、セレクタ１０２の出力が各々供給される複数のマルチプレクサ／デマルチプレクサ（ＭＵＸ： Multiplexer／ＤＭＵＸ：Demultiplexer）１０３ａ，１０３ｂ，１０３ｃ，１０３ｄ，・・・とを備えている。

【０１００】

そして、ＭＵＸ／ＤＭＵＸ１０３ａ，１０３ｂ，１０３ｃ，１０３ｄ，・・・は、各々、上記図２に示したフレームバッファ１８と描画エンジン３３に接続されている。

【０１０１】

ここで、フレームバッファ１８は、複数のメモリバンク［１］，［２］，・・・，［Ｘ］，・・・，［Ｌ］からなり、複数のメモリバンク［１］，［２］，・・・，［Ｘ］，・・・，［Ｌ］は、各々、１６個のアドレスで現される短形（以下、インターリーブパターンと言う。）の各アドレスを同時にアクセスすることができるようになされている。

【０１０２】
したがって、フレームバッファ１８の、例えば、メモリバンク［Ｘ］は、アドレスＡ０～Ａ15をアクセスするための１６個の入出力ポートＰ０～Ｐ15を備えており、複数のＭＵＸ／ＤＭＵＸ１０３ａ，１０３ｂ，１０３ｃ，１０３ｄ，・・・のうちの４個のＭＵＸ／ＤＭＵＸ１０３ａ，１０３ｂ，１０３ｃ，１０３ｄは、各々、１６個の入出力ポートＰ０～Ｐ15と接続されている。
【０１０３】
また、４個のＭＵＸ／ＤＭＵＸ１０３ａ，１０３ｂ，１０３ｃ，１０３ｄは、描画エンジン３３の４個のピクセルエンジン３３ＤX1，３３ＤX2，３３ＤX3，３３ＤX4と対応して接続されている。
【０１０４】
なお、メモリバンク［Ｘ］以外の他の各メモリバンクは、上述したメモリバンク［Ｘ］と同様の構成をしているため、その詳細な説明は省略する。また、第２のバススイッチャ３３Ｅが行う上記他の各メモリバンクに対するアクセス処理についても、後述する第２のバススイッチャ３３Ｅが行うメモリバンク［Ｘ］に対するアクセス処理と同様であるため、以下の説明では、第２のバススイッチャ３３Ｅが行うメモリバンク［Ｘ］に対するアクセス処理についてのみ説明する。
【０１０５】
まず、第２のバススイッチャ３３Ｅの一連の動作について説明する。
【０１０６】
例えば、メモリバンク［Ｘ］上に描画するポリゴンの形状が図７に示すような三角形ＴABC（第１のポリゴンの形状）であった場合、先ず、プリプロセッサ３２から制御回路１０１には、ピクセルインターリーブの制御情報が供給される。
【０１０７】
制御回路１０１は、プリプロセッサ３２からのピクセルインターリーブの制御情報に基いて、三角形ＴABC 内部をアクセスする際に用いるインターリーブパターンを、例えば、（４×４）のインターリーブパターンＰに切り換える。
【０１０８】
なお、制御回路１０１におけるインターリーブパターンの切換方法についての詳細は後述する。
【０１０９】
そして、制御回路１０１は、（４×４）のインターリーブパターンＰを用いて、メモリバンク［Ｘ］上に形成される複数のインターリーブパターンのうち、アクセスすべきインターリーブパターン、すなわち三角形ＴABC 内部を全てアクセスすることができるようなインターリーブパターンを検出する。
【０１１０】
したがって、三角形ＴABC では、メモリバンク［Ｘ］上の各インターリーブパターンをＰ（ｘ方向のパターンインデックス，ｙ方向のパターンインデックス）で示した場合、図８に示すように、

Ｐ（ｘ，ｙ）＝Ｐ（３，１），Ｐ（４，１），Ｐ（１，２），Ｐ（２，２），
Ｐ（３，２），Ｐ（４，２），Ｐ（１，３），Ｐ（２，３），
Ｐ（３，３），Ｐ（４，３），Ｐ（５，３），Ｐ（２，４），
Ｐ（３，４），Ｐ（４，４），Ｐ（５，４），Ｐ（３，５），
Ｐ（４，５），Ｐ（５，５），Ｐ（４，６），Ｐ（５，６）

で示される合計２０個のインターリーブパターンが検出される。
【０１１１】
そして、制御回路１０１は、上述のようにして検出した２０個のインターリーブパターンを示すパターン情報をインターリーブパターン単位でセレクタ１０２に供給する。また、１アドレス単位でメモリアクセスを行う場合には、制御回路１０１は、三角形ＴABC の形状に基いたマスク情報をセレクタ１０２に供給する。

【０１１２】
セレクタ１０２は、制御回路１０１からインターリーブパターン単位で供給されたパターン情報に基いて、アクセスすべき（４×４）のインターリーブパターンＰに対応したアドレスをＭＵＸ／ＤＭＵＸ１０３ａ，１０３ｂ，１０３ｃ，１０３ｄに指定する。
【０１１３】
また、セレクタ１０２は、制御回路１０１からマスク情報が供給された場合には、そのマスク情報に基いて、図９に示すように、（４×４）のインターリーブパターンＰのなかでマスクを行った結果得られるアクセスすべきアドレスをＭＵＸ／ＤＭＵＸ１０３ａ，１０３ｂ，１０３ｃ，１０３ｄに指定する。したがって、例えば、図１０に示すように、上記図９に示したＰ（４，１）で示されるインターリーブパターン内のアドレスA0 ～A15において、マスクを行った結果得られるアクセスすべきアドレスは、Ａ４，Ａ５，Ａ６，Ａ８，Ａ９，Ａ１０，Ａ１３，Ａ１４，Ａ１５（斜線部分）となる。
【０１１４】
ＭＵＸ／ＤＭＵＸ１０３ａ，１０３ｂ，１０３ｃ，１０３ｄは、各々、メモリバンク［Ｘ］のアドレスA0 ～A15のうち、セレクタ１０２により指定されたアドレスをアクセスする。
【０１１５】
ここで、上述したように、ピクセルエンジン３３ＤX1，３３ＤX2，３３ＤX3，３３ＤX4からＭＵＸ／ＤＭＵＸ１０３ａ，１０３ｂ，１０３ｃ，１０３ｄには、各々、画素データが供給されるようになされている。
【０１１６】
そこで、例えば、ＭＵＸ／ＤＭＵＸ１０３ａは、セレクタ１０２により指定されたアドレスをアクセスすることにより、入出力ポートP0 ～P15のうち上記アドレスに対応した入出力ポートを介して、ピクセルエンジンＸａからの画素データをメモリバンク［Ｘ］の上記アドレスにより示される領域に書き込む。
【０１１７】
また、ＭＵＸ／ＤＭＵＸ１０３ａは、セレクタ１０２により指定されたアドレスをアクセスすることにより、入出力ポートP0 ～P15のうち上記アドレスに対応した入出力ポートを介して、メモリバンク［Ｘ］の上記アドレスにより示される領域に書き込まれているデータを読み出す。そして、ＭＵＸ／ＤＭＵＸ１０３ａは、メモリバンク［Ｘ］から読み出したデータに対して所定の処理を行う。
【０１１８】
なお、ＭＵＸ／ＤＭＵＸ１０３ｂ～１０３ｄの動作については、上述したＭＵＸ／ＤＭＵＸ１０３ａの動作と同様であるため、その詳細な説明は省略する。
【０１１９】
つぎに、上述した制御回路１０１におけるインターリーブパターンの切換方法について具体的に説明する。
【０１２０】
まず、メモリバンク［Ｘ］上に描画するポリゴンの形状が、例えば、図１１に示すようにな横長の三角形TDEF （第２のポリゴンの形状）であり、三角形TDEF を（４×４）のインターリーブパターンＰでアクセスする場合のアクセス回数について説明する。
【０１２１】
この場合、アクセスすべきインターリーブパターンの個数は、図１２に示すように、
Ｐ（ｘ，ｙ）＝Ｐ（１，１），Ｐ（２，１），Ｐ（３，１），
Ｐ（４，１），Ｐ（５，１），Ｐ（０，２），
Ｐ（１，２），Ｐ（２，２），Ｐ（３，２），
Ｐ（４，２），Ｐ（５，２），Ｐ（６，２），
Ｐ（７，２），Ｐ（８，２），Ｐ（７，３），
Ｐ（８，３），Ｐ（９，３）
の合計１７個となる。

【０１２２】
すなわち、（４×４）のインターリーブパターンＰで三角形ＴDEF をアクセスする場合、三角形ＴDEF 内部を全てアクセスするためのアクセス回数は、１７回となる。
【０１２３】
また、１アドレス単位でアクセスする場合には、上述した三角形ＴABC のアクセス時と同様に、図１３に示すように、（４×４）のインターリーブパターンＰのなかでマスクを行うことにより、必要なメモリアドレスのみをアクセスすることとなる。
【０１２４】
つぎに、図１４に示すように、三角形ＴDEF を（８×２）のインターリーブパターンＰ1でアクセスする場合、アクセスすべきインターリーブパターンの個数は、図１５に示すように、

Ｐ1（ｘ，ｙ）＝Ｐ1（１，２），Ｐ1（２，２），Ｐ1（０，３），
Ｐ1（１，３），Ｐ1（２，３），Ｐ1（０，４），
Ｐ1（１，４），Ｐ1（２，４），Ｐ1（３，４），
Ｐ1（１，５），Ｐ1（２，５），Ｐ1（３，５），
Ｐ1（４，５），Ｐ1（３，６），Ｐ1（４，６）

の合計１５個となる。
【０１２５】
すなわち、（８×２）のインターリーブパターンＰ1 で三角形ＴDEF をアクセスする場合、三角形ＴDEF 内部を全てアクセスするためのアクセス回数は、１５回となる。
【０１２６】
また、１アドレス単位でアクセスする場合には、上述した三角形ＴABC のアクセス時と同様に、図１６に示すように、（８×２）のインターリーブパターンＰ1 のなかでマスクを行うことにより、必要なメモリアドレスのみをアクセスすることとなる。
【０１２７】
つぎに、図１７に示すように、三角形ＴDEF を（１６×１）のインターリーブパターンＰ2 でアクセスする場合、アクセスすべきインターリーブパターンの個数は、図１８に示すように、

Ｐ2（ｘ，ｙ）＝Ｐ2（０，５），Ｐ2（１，５），Ｐ2（０，６），
Ｐ2（１，６），Ｐ2（０，７），Ｐ2（１，７），
Ｐ2（０，８），Ｐ2（１，８），Ｐ2（０，９），
Ｐ2（１，９），Ｐ2（０，１０），Ｐ2（１，１０），
Ｐ2（２，１０），Ｐ2（１，１１），Ｐ2（２，１１），
Ｐ2（１，１２），Ｐ2（２，１２），Ｐ2（２，１３）

の合計１８個となる。
【０１２８】
すなわち、（１６×１）のインターリーブパターンＰ2 で三角形ＴDEF をアクセスする場合、三角形ＴDEF 内部を全てアクセスするためのアクセス回数は、１８回となる。
【０１２９】
また、１アドレス単位でアクセスする場合には、上述した三角形ＴABC のアクセス時と同様に、図１９に示すように、（８×２）のインターリーブパターンＰ2 のなかでマスクを行うことにより、必要なメモリアドレスのみをアクセスすることとなる。
【０１３０】
上述のように、（４×４）のインターリーブパターンＰで三角形ＴDEF をアクセスする場合のアクセス回数は１７回、（８×２）のインターリーブパターンＰ1で三角形ＴDEF をアクセスする場合のアクセス回数は１５回、（１６×１）のインターリーブパターンＰ2 で三角形ＴDEF をアクセスする場合のアクセス回数は１８回となり、この結果、（８×２）のインターリーブパターンＰ1 で三角形ＴDEF をアクセスする場合のアクセス回数が最少のアクセス回数となる。したがって、三角形ＴDEF に対する適切なインターリーブパターンは、（８×２）のインターリーブパターンＰ1ということがわかる。

【０１３１】
そこで、制御回路１０１は、メモリバンク［Ｘ］をアクセスする際に用いるインターリーブパターンを、アクセスするポリゴンの形状に応じた適切なインターリーブパターンに切り換えるために、以下のような処理を行う。
【０１３２】
例えば、メモリバンク［Ｘ］上に描画するポリゴンの形状が図２０に示すような三角形ＴHIJ であった場合、先ず、制御回路１０１には、上述したように、プリプロセッサ３２からピクセルインターリーブの制御情報が供給される。このピクセルインターリーブの制御情報は、例えば、三角形ＴHIJの３つの頂点Ｈ，Ｉ，Ｊのｘｙ座標Ｈ（Ｘｈ，Ｙｈ），Ｉ（Ｘｉ，Ｙｉ），Ｊ（Ｘｊ，Ｙｊ）等の情報である。
【０１３３】
次に、制御回路１０１は、上記図２０に示すように、プリプロセッサ３２からのピクセルインターリーブの制御情報を用いて、三角形ＴHIJ の縦横比Ｒを、Ｘ方向の最大値ＭＡＸｘ及び最少値ＭＩＮｘ、Ｙ方向の最大値ＭＡＸｙ及び最少値ＭＩＮｙを持って、

$$R = dy/dx = (MAXx - MINx)/(MAXy - MINy)$$

なる演算により求める。
【０１３４】
なお、三角形ＴHIJ では、

$$MAXx = Xj$$
$$MINx = Xi$$
$$MAXy = Yh$$
$$MINy = Yi$$

となる。
【０１３５】
そして、制御回路１０１は、上述のようにして求めた縦横比Ｒに応じて、図２１に示すような、（１×１６）、（２×８）、（４×４）、（８×２）、（１６×１）の５種類のインターリーブパターンＰａ～Ｐｅのうち適切なインターリーブパターンを選出し、三角形ＴHIJ をアクセスする際に用いるインターリーブパターンを、選出したインターリーブパターンに切り換える。
【０１３６】
ここで、制御回路１０１は、表１に示すような、縦横比Ｒとインターリーブパターンと対応表からなるテーブルを有している。このテーブルには、縦横比Ｒに応じた適切なインターリーブパターン、すなわちアクセス回数が最小となるようなインターリーブパターンが予め設定されている。したがって、制御回路１０１は、上記テーブルを用いることにより、上述のようにして得られた縦横比Ｒに基いた適切なインターリーブパターンを選出することとなる。
【０１３７】

【表１】

| 縦横比Ｒ | インターリーブパターン |
|---|---|
| ～０．１ | Ｐａ（１６×１） |
| ０．１～０．５ | Ｐｂ（８×２） |
| ０．５～２．０ | Ｐｃ（４×４） |
| ２．０～８．０ | Ｐｄ（２×８） |
| ８．０～ | Ｐｅ（１×１６） |

【０１３８】
上述のように、第２のバススイッチャ３３Ｅでは、メモリバンク［Ｘ］上に描画するポリゴンの形状に応じて、上記図２１に示したような５種類のインターリーブパターンＰａ～Ｐｅから適切なインターリーブパターンを選出し、選出したインターリーブパターンでメモリバンク［Ｘ］をアクセスするため、最小のアクセス回数でメモリバンク［Ｘ］上に上記ポリゴンを描画することができる。したがって、第２のバススイッチャ３３Ｅは、メモリアクセスを効率良く行うことができる。

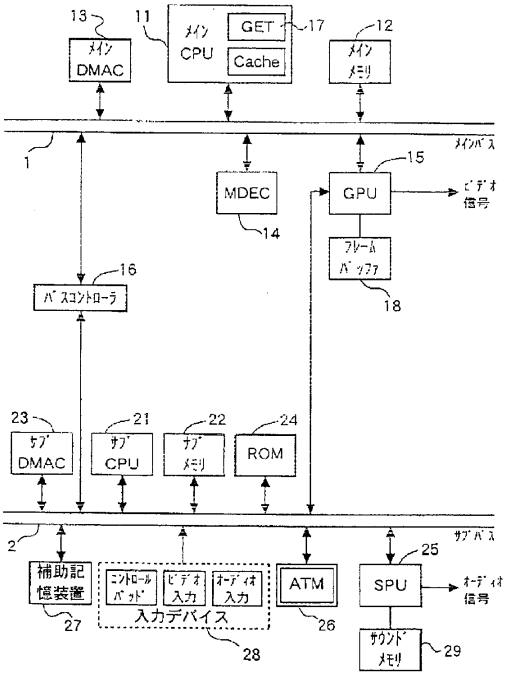
【０１３９】
また、ＧＰＵ１５は、上述のような、メモリアクセスの効率化を図った第２のバススイッチャ３３Ｅにより、フレームバッファ１８をアクセスしてデータ処理を行うため、そのデータ処理を効率良く行うことができる。

【図面の簡単な説明】

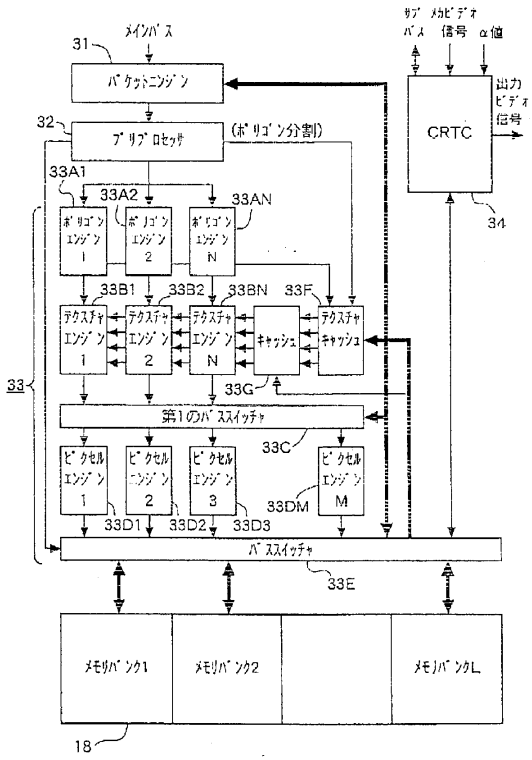
【０１４０】
【図１】本発明を適用したビデオゲーム装置の構成を示すブロック図である。
【図２】上記ビデオゲーム装置におけるＧＰＵの具体的な構成を示すブロック図である。
【図３】上記ＧＰＵの基本的な構成をブロック図である。
【図４】上記ＧＰＵにおけるテキスチャキャッシュ内のデータ構造の一例を示す図である。
【図５】上記ＧＰＵにおけるプリプロセッサによるポリゴンの分割処理を示すフローチャートである。
【図６】上記ビデオゲーム装置における第２のバススイッチャの構成を示すブロック図である。
【図７】上記ビデオゲーム装置におけるフレームバッファのメモリバンク上に描画する第１のポリゴンの形状内部をアクセスする場合について説明するための図である。
【図８】上記第１のポリゴンの形状内部をアクセスする際のアクセスすべきインターリーブパターンを説明するための図である。
【図９】上記第１のポリゴンの形状内部をアクセスする際に、１アドレス単位でアクセス

する場合のマスク処理について説明するための図である。
【図１０】上記マスク処理により得られたアクセスアドレスを説明するための図である。
【図１１】上記フレームバッファのメモリバンク上に描画する第２のポリゴンの形状内部を（４×４）のインターリーブパターンでアクセスする場合について説明するための図である。
【図１２】上記第２のポリゴンの形状内部を（４×４）のインターリーブパターンでアクセスする場合のアクセスすべきインターリーブパターンを説明するための図である。
【図１３】上記第２のポリゴンの形状内部を（４×４）のインターリーブパターン内で１アドレス単位でアクセスする場合のマスク処理について説明するための図である。
【図１４】上記第２のポリゴンの形状内部を（８×２）のインターリーブパターンでアクセスする場合について説明するための図である。
【図１５】上記第２のポリゴンの形状内部を（８×２）のインターリーブパターンでアクセスする場合のアクセスすべきインターリーブパターンを説明するための図である。
【図１６】上記第２のポリゴンの形状内部を（８×２）のインターリーブパターン内で１アドレス単位でアクセスする場合のマスク処理について説明するための図である。
【図１７】上記第２のポリゴンの形状内部を（１６×１）のインターリーブパターンでアクセスする場合について説明するための図である。
【図１８】上記第２のポリゴンの形状内部を（１６×１）のインターリーブパターンでアクセスする場合のアクセスすべきインターリーブパターンを説明するための図である。
【図１９】上記第２のポリゴンの形状内部を（１６×１）のインターリーブパターン内で１アドレス単位でアクセスする場合のマスク処理について説明するための図である。
【図２０】上記フレームバッファのメモリバンク上に描画するポリゴンの形状の縦横比を算出する処理を説明するための図である。
【図２１】１６アドレスを有する５種類のインターリーブパターンを示したパターン図である。
【符号の説明】
【０１４１】
１ メインバス、１１ メインＣＰＵ、１２ メインメモリ、１３ メインＤＭＡＣ、１５ ＧＰＵ、１７ ＧＴＥ、１８ フレームバッファ、３１ パケットエンジン、３２ プリプロセッサ、３３ 描画エンジン、３３Ａ１，３３Ａ２・・・３３ＡＮ ポリゴンエンジン、３３Ｂ１，３３Ｂ２・・・３３ＢＮ テクスチャエンジン、３３Ｃ 第１のバススイッチャ、３３Ｄ１，３３Ｄ２・・・３３ＤＭ、３３Ｅ 第２のバススイッチャ、３３Ｆ テクスチャキャッシュ

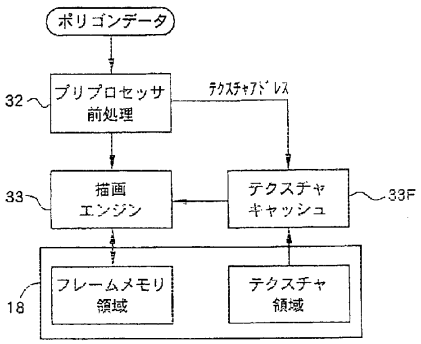
【図１】


メイン DMAC
メイン CPU
GET
Cache
メイン メモリ
メインバス
MDEC
GPU
ビデオ信号
フレームバッファ
バスコントローラ
サブ DMAC
サブ CPU
サブ メモリ
ROM
サブバス
補助記憶装置
コントロールパッド
ビデオ入力
オーディオ入力
入力デバイス
ATM
SPU
オーディオ信号
サウンドメモリ

【図２】


メインバス
パケットエンジン
プリプロセッサ
(ポリゴン分割)
サブ メインビデオ バス 信号 α値
出力ビデオ信号
CRTC
ポリゴンエンジン 1
ポリゴンエンジン 2
ポリゴンエンジン N
テクスチャエンジン 1
テクスチャエンジン 2
テクスチャエンジン N
テクスチャキャッシュ
キャッシュ
第1のバススイッチャ
ピクセルエンジン 1
ピクセルエンジン 2
ピクセルエンジン 3
ピクセルエンジン M
バススイッチャ
メモリバンク1
メモリバンク2
メモリバンクL

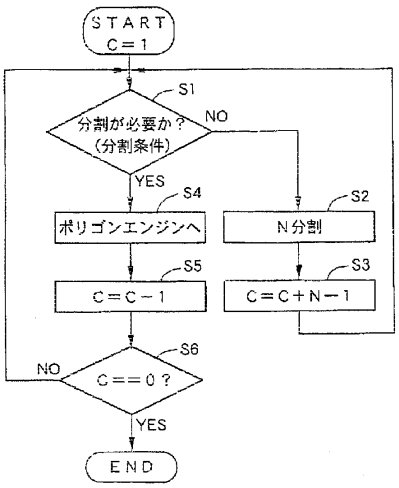
【図３】


ポリゴンデータ
プリプロセッサ前処理
テクスチャアドレス
描画エンジン
テクスチャキャッシュ
フレームメモリ領域
テクスチャ領域

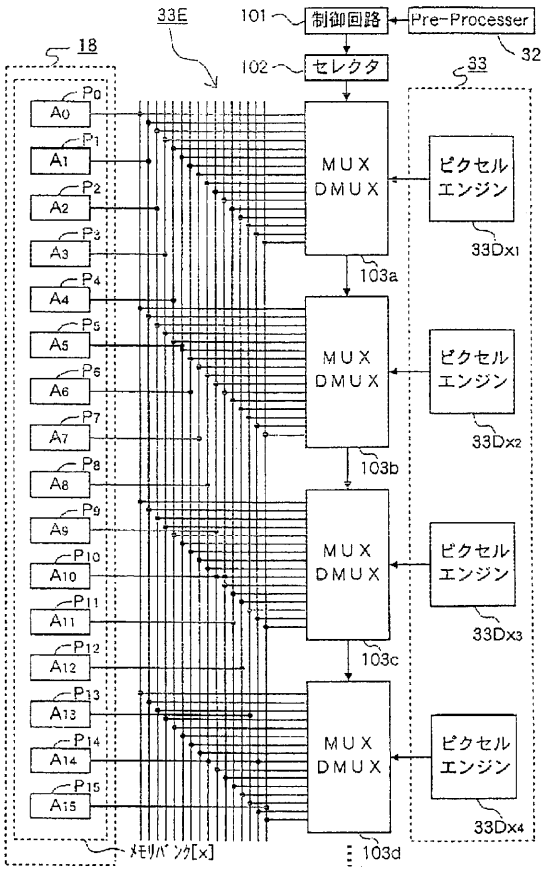
【図４】


TAG
DATA
L

【図５】

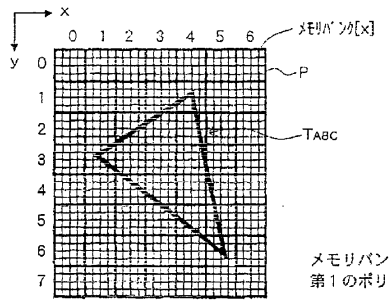

START C＝1
分割が必要か？(分割条件)
NO
YES
ポリゴンエンジンへ
N分割
C＝C－1
C＝C＋N－1
C＝＝0？
NO
YES
END

【図６】



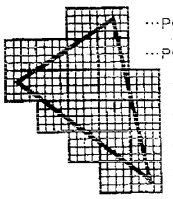

本発明の係るメモリアクセス方法を用いたバススイッチ

【図７】




メモリバンクに描画する
第１のポリゴンの形状

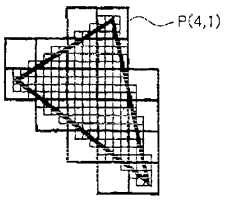
【図８】



…P(3,1),P(4,1)
…P(1,2),P(2,2),P(3,2),P(4,2)
…P(1,3),P(2,3),P(3,3),P(4,3),P(5,3)
…P(2,4),P(3,4),P(4,4),P(5,4)
…P(3,5),P(4,5),P(5,5)
…P(4,6),P(5,6)

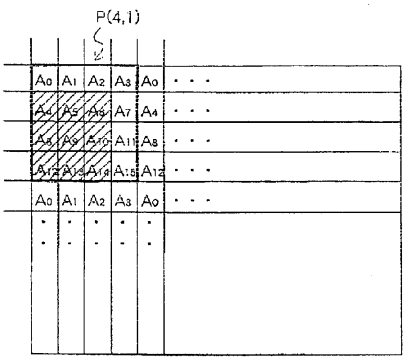
アクセスすべき
インターリーブパターン

【図９】



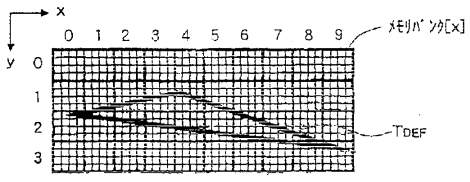

１アドレス単位でアクセス
する場合のマスク処理

【図１０】



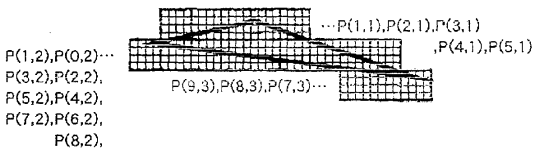

マスク処理により得られたアクセスアドレス

【図１１】




第２のポリゴンの形状を（４×４）短形でアクセスする場合

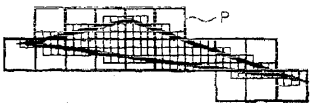
【図１２】



P(1,2),P(0,2)…
P(3,2),P(2,2),
P(5,2),P(4,2),
P(7,2),P(6,2),
P(8,2),
…P(1,1),P(2,1),P(3,1)
,P(4,1),P(5,1)
P(9,3),P(8,3),P(7,3)…

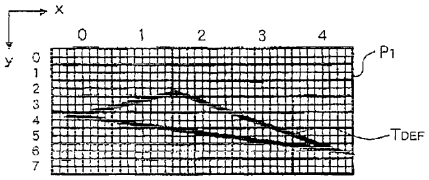
（４×４）短形でアクセスすべきインターリーブパターン

【図１３】



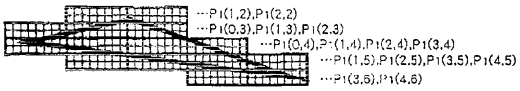

（４×４）短形内でアクセスする場合のマスク処理

【図１４】



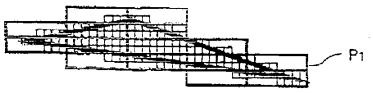

第２のポリゴンの形状を（８×２）短形でアクセスする場合

【図１５】



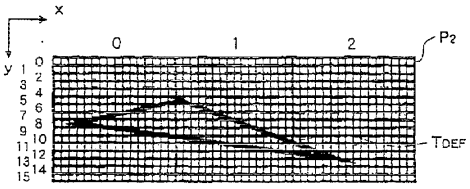

（８×２）短形でアクセスする場合の
アクセスすべきインターリーブパターン

【図１６】



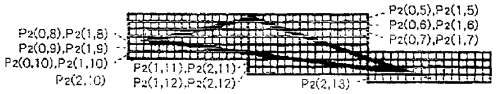

（８×２）短形内で１アドレス単位で
アクセスする場合のマスク処理

【図１７】



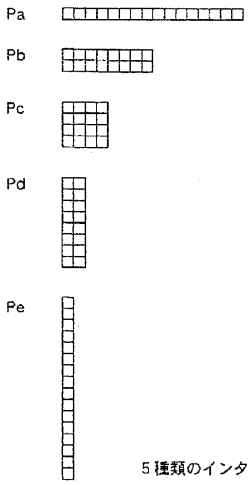

第２のポリゴンの形状を（１６×１）短形でアクセスする場合

【図１８】



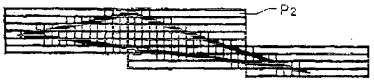

（１６×１）短形でアクセスする場合の
アクセスすべきインターリーブパターン

【図２１】



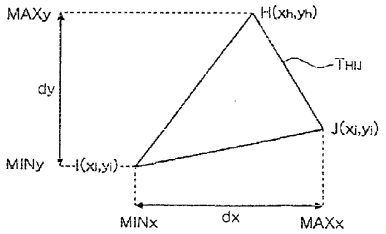

５種類のインターリーブパターン

【図１９】




（１６×１）短形内で１アドレス単位
でアクセスする場合のマスク処理

【図２０】




アクセスする形状の縦横比の算出

フロントページの続き

(56)参考文献 特開平６－３４８８５７（ＪＰ，Ａ）
特開平６－３４８８５８（ＪＰ，Ａ）
特開平７－２６２４０５（ＪＰ，Ａ）
特開平７－１８２５３７（ＪＰ，Ａ）
特開平３－２７１８７７（ＪＰ，Ａ）
特開平５－７３２５９（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｔ　１５／００