

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3681026号

(P3681026)

(45) 発行日 平成17年8月10日(2005.8.10)

(24) 登録日 平成17年5月27日(2005.5.27)

(51) Int. Cl.<sup>7</sup>

F I

G06T 1/20  
G06F 15/16  
G06T 11/00  
G09G 5/36

G06T 1/20 B  
G06F 15/16 620A  
G06T 11/00 100A  
G09G 5/36 530C

請求項の数 6 (全 18 頁)

(21) 出願番号 特願平9-74931  
(22) 出願日 平成9年3月27日(1997.3.27)  
(65) 公開番号 特開平10-269351  
(43) 公開日 平成10年10月9日(1998.10.9)  
審査請求日 平成14年8月29日(2002.8.29)

(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  
(72) 発明者 鈴置 雅一  
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 真木 健彦

(56) 参考文献 特開平05-266201(JP, A)

最終頁に続く

(54) 【発明の名称】 情報処理装置および方法

(57) 【特許請求の範囲】

【請求項1】

第1の命令生成手段と、第2の命令生成手段と、供給された命令を実行する実行手段と、前記第1の命令生成手段からの命令と前記第2の命令生成手段からの命令とを調停して前記実行手段に供給する調停手段とを備え、

前記調停手段は、前記第1の命令生成手段からの命令を優先的に前記実行手段に供給し、前記第1の命令生成手段が処理中であり、かつ、前記実行手段が待機中であるとき、前記第2の命令生成手段からの命令を前記実行手段に供給することを特徴とする情報処理装置。

【請求項2】

前記実行手段は、前記第1の命令生成手段からの命令を実行する際に用いる第1のパラメータおよび前記第2の命令生成手段からの命令を実行する際に用いる第2のパラメータを保持し、

供給された命令を生成した命令生成手段に応じて、用いるパラメータを切り替えることを特徴とする請求項1に記載の情報処理装置。

【請求項3】

前記第2の命令生成手段が生成した命令を記憶する記憶手段をさらに備え、

前記調停手段は、前記記憶手段に記憶された命令を前記実行手段に供給することを特徴とする請求項1に記載の情報処理装置。

【請求項4】

10

20

前記第 1 の命令生成手段は、ベクトル処理装置で構成され、

前記第 2 の命令生成手段は、ベクトル処理装置およびマイクロプロセッサで構成されることを特徴とする請求項 1 に記載の情報処理装置。

【請求項 5】

第 1 の命令生成手段と、第 2 の命令生成手段と、供給された命令を実行する実行手段と、前記第 1 の命令生成手段からの命令と第 2 の前記命令生成手段からの命令とを調停して前記実行手段に供給する調停手段とを備えた情報処理装置における情報処理方法であって、

前記調停手段は、前記第 1 の命令生成手段からの命令を優先的に前記実行手段に供給し、前記第 1 の命令生成手段が処理中であり、かつ、前記実行手段が待機中であるとき、前記第 2 の命令生成手段からの命令を前記実行手段に供給することを特徴とする情報処理方法。

10

【請求項 6】

供給された命令を実行する実行手段に命令を供給する演算処理装置であって、

第 1 の命令生成手段と、第 2 の命令生成手段と、前記第 1 の命令生成手段からの命令と第 2 の前記命令生成手段からの命令とを調停して前記実行手段に供給する調停手段とを備え、

前記調停手段は、前記第 1 の命令生成手段からの命令を優先的に前記実行手段に供給し、前記第 1 の命令生成手段が処理中であり、かつ、前記実行手段が待機中であるとき、前記第 2 の命令生成手段からの命令を前記実行手段に供給することを特徴とする演算処理装置。

20

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、情報処理装置および方法に関し、例えば、複数の CPU がそれぞれ並列に描画命令を生成し、生成された描画命令を CPU の優先順位に従って描画装置に転送することにより、効率的に描画処理を行うようにした情報処理装置および方法に関する。

【0002】

【従来の技術】

近年の L S I の集積度の向上により、1 つの L S I に同一の回路を複数個並列に並べて、処理速度を向上させることができるようになった。また、近年のビデオゲームシステムは、中央処理装置と描画装置から構成され、CPU はコントローラからの入力データに基づいて、表示すべき所定のポリゴンを表すデータに対して 2 次元や 3 次元の座標変換処理を行い、そのポリゴンに 2 次元のフレームバッファに描画するための描画命令列（ディスプレイリスト）を作成する。そして、描画装置は、受け取った描画命令列に従って、フレームバッファへのポリゴンの描画を行う。

30

【0003】

【発明が解決しようとする課題】

しかしながら、並列化による処理能力の向上という効果は、単純操作の繰り返しが多い描画装置により多く働く。そのため、近年では、描画装置による処理に比べて、CPU による処理の方が相対的に遅くなってきている。さらに、3 次元モデルデータの容量を削減するためには、少ない頂点データ（コントロールポイント）から、曲面などの複雑な図形をその都度生成する必要があるが、その処理は CPU 側の負荷となる。そのため、描画処理を効率的に行わせることができない課題があった。

40

【0004】

本発明はこのような状況に鑑みてなされたものであり、複数の CPU に優先順位を設定し、それらを並列に動作させ、優先順位の高い CPU が生成した描画命令列を優先的に処理することにより、効率的に処理を行うことができるようにするものである。

【0005】

【課題を解決するための手段】

50

請求項 1 に記載の情報処理装置は、第 1 の命令を生成する処理を行う第 1 の生成手段と、第 2 の命令を生成する処理を行う第 2 の生成手段と、第 1 の命令、および第 2 の命令を所定のタイミングで切り替えて実行する実行手段と、第 1 の命令を実行手段に転送する第 1 の転送手段と、第 2 の命令を実行手段に転送する第 2 の転送手段とを備え、第 2 の転送手段は、第 1 の生成手段が処理中であり、かつ実行手段が待機中であるとき、第 2 の命令を実行手段に供給し、実行手段は、第 2 の命令の供給を受けたとき、第 2 の命令を実行することを特徴とする。

【 0 0 0 6 】

請求項 5 に記載の情報処理方法は、第 2 の転送手段は、第 1 の生成手段が処理中であり、かつ実行手段が待機中であるとき、第 2 の命令を実行手段に供給し、実行手段は、第 2 の命令の供給を受けたとき、第 2 の命令を実行することを特徴とする。

10

【 0 0 0 7 】

請求項 1 に記載の情報処理装置においては、第 1 の生成手段が、第 1 の命令を生成する処理を行い、第 2 の生成手段が、第 2 の命令を生成する処理を行い、実行手段が、第 1 の命令、および第 2 の命令を所定のタイミングで切り替えて実行し、第 1 の転送手段が、第 1 の命令を実行手段に転送し、第 2 の転送手段が、第 2 の命令を実行手段に転送するとき、第 2 の転送手段は、第 1 の生成手段が処理中であり、かつ実行手段が待機中であるとき、第 2 の命令を実行手段に供給し、実行手段は、第 2 の命令の供給を受けたとき、第 2 の命令を実行する。

【 0 0 0 8 】

20

請求項 5 に記載の情報処理装置においては、第 2 の転送手段は、第 1 の生成手段が処理中であり、かつ実行手段が待機中であるとき、第 2 の命令を実行手段に供給し、実行手段は、第 2 の命令の供給を受けたとき、第 2 の命令を実行する。

【 0 0 0 9 】

【 発明の実施の形態 】

図 1 乃至図 3 は、本発明の情報処理装置を応用した家庭用 TV ゲーム機の一例を示している。この家庭用 TV ゲーム機は、ゲーム機本体 2 と、このゲーム機本体 2 に接続可能な操作装置 1 7 および記録装置 3 8 で構成されている。

【 0 0 1 0 】

ゲーム機本体 2 は、図 1 乃至図 3 に示すように、略四角形状に形成され、その中央の位置に、図 4 に示すような光ディスクの一種である CD - ROM (compact disc-read only memory) 4 0 (ゲーム用媒体に相当するディスク) を装着するディスク装着部 3 と、ゲーム機本体の適宜位置にゲームを任意にリセットするリセットスイッチ 4 と、電源のオン / オフをする電源スイッチ 5 と、ディスクの装着を操作するディスク操作スイッチ 6 と、所謂ゲームにおける操作を行う操作装置 1 7 および所謂ゲームの設定などを記録しておく記録装置 3 8 を接続する接続部 7 A , 7 B を有している。

30

【 0 0 1 1 】

接続部 7 A , 7 B は、図 2 および図 3 に示すように、2 段に形成されている。接続部 7 A , 7 B の上段部には、記録装置 3 8 を接続する記録挿入部 8 が設けられ、下段部には、操作装置 1 7 を接続する接続端子挿入部 1 2 が設けられている。

40

【 0 0 1 2 】

記録挿入部 8 は、横方向に長い長方形の挿入孔と、その内部に記録装置 3 8 が挿入されるメモリ用接続端子部 (図示せず) を有している。また、図 2 に示すように、記録挿入部 8 には、記録装置 3 8 が接続されていないときに、メモリ用接続端子部を埃などから保護するシャッター 9 が設けられている。なお、記録装置 3 8 は、電氣的に書換え可能な ROM を有しており、所謂ゲームに関連するデータを記録するようになされている。

【 0 0 1 3 】

記録装置 3 8 を装着する場合、ユーザは、記録装置 3 8 の先端でシャッター 9 を内側方向に押し、さらに、記録装置 3 8 を挿入孔に押し込み、メモリ用接続端子部に接続させる。

【 0 0 1 4 】

50

接続端子挿入部 1 2 は、図 2 に示すように、横方向に長い長方形形状の挿入孔と、操作装置 1 7 の接続端子部 2 6 を接続する接続端子 1 2 A を有している。

【 0 0 1 5 】

操作装置 1 7 は、図 1 に示すように、両手の掌で挟持して 5 本の指が自由自在に動いて操作できる構造をしており、左右対称に配置された操作部 1 8 , 1 9 と、操作部 1 8 , 1 9 の中間部に設けたセレクトスイッチ 2 2 およびスタートスイッチ 2 3 と、操作部 1 8 , 1 9 の前面側に配置された操作部 2 4 , 2 5 と、ゲーム機本体 2 に接続するための接続端子部 2 6 およびケーブル 2 7 を有している。

【 0 0 1 6 】

図 5 は、上述のゲーム機本体 2 の内部の電氣的構成の一例を示している。

10

【 0 0 1 7 】

ゲーム機本体 2 は、メインバス 4 1 とサブバス 4 2 の 2 本のバスを有している。これらのバスは、サブバスインタフェース ( S B U S I F ) 4 3 を介して接続されている。

【 0 0 1 8 】

メインバス 4 1 には、マイクロプロセッサや第 1 ベクトル処理装置 ( V P E ( vector processing engine ) 0 ) など構成されるメイン C P U ( central processing unit ) 4 4 ( 第 2 の生成手段 ) と、 R A M ( random access memory ) で構成されるメインメモリ 4 5 ( 記憶手段 ) 、メインダイレクトメモリアクセスコントローラ ( メイン D M A C ( direct memory access controller ) ) 4 6 ( 第 1 の転送手段、第 2 の転送手段 ) 、 M P E G ( M 20 oving Picture Experts Group ) デコーダ ( M D E C ) 4 7 、第 2 ベクトル処理装置 ( V P E 1 ) 4 8 ( 第 1 の生成手段 ) 、および G P U I F ( graphical processing unit interface ) 7 2 を介して画像処理装置 ( G P U ) 4 9 ( 実行手段 ) が接続されている。 G P U 4 9 には、 C R T C ( CRT controller ) 8 4 が設けられている。

20

【 0 0 1 9 】

一方、サブバス 4 2 には、マイクロプロセッサなどで構成されるサブ C P U 5 0 、 R A M で構成されるサブメモリ 5 1 、サブ D M A C 5 2 、オペレーティングシステムなどのプログラムが記憶されている R O M 5 3 、音声処理装置 ( S P U ( sound processing unit ) ) 5 4 、通信制御部 ( A T M ) 5 5 、ディスク装着部 3 を兼ねる C D - R O M ドライブ 5 6 、入力部 5 7 が接続されている。そして、入力部 5 7 の接続端子 1 2 A には、操作装置 1 7 が接続される。

30

【 0 0 2 0 】

S B U S I F 4 3 は、メインバス 4 1 とサブバス 4 2 を接続し、メインバス 4 1 からのデータをサブバス 4 2 へ出力するとともに、サブバス 4 2 からのデータをメインバス 4 1 へ出力するようになされている。

【 0 0 2 1 】

メイン C P U 4 4 は、ゲーム機本体 2 の起動時に、サブバス 4 2 に接続されている R O M 5 3 から、 S B U S I F 4 3 を介して起動プログラムを読み込み、その起動プログラムを実行し、オペレーティングシステムを動作させるようになされている。

【 0 0 2 2 】

また、メイン C P U 4 4 は、 C D - R O M ドライブ 5 6 を制御し、 C D - R O M ドライブ 5 6 にセットされた C D - R O M 4 0 からアプリケーションプログラムやデータを読み出し、メインメモリ 4 5 に記憶させるようになされている。

40

【 0 0 2 3 】

また、メイン C P U 4 4 は、 C D - R O M 4 0 から読み出した、複数の基本図形 ( ポリゴン ) で構成された 3 次元オブジェクトのデータ ( ポリゴンの頂点 ( 代表点 ) の座標値など ) に対して、第 1 ベクトル処理装置 ( V P E 0 ) 7 1 ( 第 2 の生成手段 ) と共同して、非定型処理用のデータ ( ポリゴン定義情報 ) を生成するようになされている。第 1 ベクトル処理装置 ( V P E 0 ) 7 1 は、浮動小数点の実数を演算する複数の演算素子を有し、並列に浮動小数点演算を行うようになされている。

【 0 0 2 4 】

50

即ち、メインCPU44と、第1ベクトル処理装置71は、ジオメトリ処理のうちのポリゴン単位での細かな操作を必要とする処理、例えば、木の葉が風で揺れる様子や、自動車のフロントウィンドウの雨の滴等を表すポリゴンのデータを生成するような処理を行うようになされている。そして、演算された頂点情報やシェーディングモード情報等のポリゴン定義情報をパケットとして、メインバス41を介してメインメモリ45に供給するようになされている。

【0025】

ポリゴン定義情報は、描画領域設定情報とポリゴン情報とからなる。そして、描画領域設定情報は、描画領域のフレームバッファアドレスにおけるオフセット座標と、描画領域の外部にポリゴンの座標があった場合に、描画をキャンセルするための描画クリッピング領域の座標からなる。また、ポリゴン情報は、ポリゴン属性情報と頂点情報とからなり、ポリゴン属性情報は、シェーディングモード、ブレンディングモード、およびテクスチャマッピングモード等を指定する情報であり、頂点情報は、頂点描画領域内座標、頂点テクスチャ領域内座標、および頂点色等の情報である。

10

【0026】

一方、第2ベクトル処理装置(VPE1)48は、第1ベクトル処理装置71の場合と同様に、浮動小数点の実数を演算する複数の演算素子を有し、並列に浮動小数点演算を行うようになされている。そして、操作装置17の操作とマトリクスの操作で画像を生成できるもの、即ち、VPE1においてプログラムが可能な程度の比較的簡単な処理(定型処理)用のデータ(ポリゴン定義情報)を生成するようになされている。例えば、ビルや車等の簡単な形状の物体に対する透視変換、平行光源計算、2次元曲面生成等の処理を行うようになされている。そして、生成したポリゴン定義情報をGPUIF72に供給するようになされている。

20

【0027】

GPUIF72は、メインメモリ45よりメインバス41を介して供給されるメインCPU44からのポリゴン定義情報と、第2ベクトル処理装置48より供給されるポリゴン定義情報とを、衝突しないように調停しながら、GPU49に供給するようになされている。

【0028】

GPU49は、GPUIF72を介して供給されたポリゴン定義情報に基づいて、フレームメモリ58にポリゴンを描画するようになされている。GPU49は、フレームメモリ58をテクスチャメモリとしても使用できるため、フレームメモリ上のピクセルイメージをテクスチャとして、描画するポリゴンに貼り付けることができる。

30

【0029】

メインDMAC46は、メインバス41に接続されている各回路を対象として、DMA転送などの制御を行うようになされている。また、メインDMAC46は、SBUSIF43の状態に応じて、サブバス42に接続されている各回路を対象としてDMA転送などの制御を行うこともできる。また、MDEC47は、メインCPU44と並列に動作し、MPEG(Moving Picture Experts Group)方式あるいはJPEG(Joint Photographic Experts Group)方式等で圧縮されたデータを伸張するようになされている。

40

【0030】

サブCPU50は、ROM53に記憶されているプログラムに従って各種動作を行うようになされている。サブDMAC52は、SBUSIF43がメインバス41とサブバス42を切り離している状態においてのみ、サブバス42に接続されている各回路を対象としてDMA転送などの制御を行うようになされている。

【0031】

SPU54は、サブCPU50やサブDMAC52から供給されるサウンドコマンドに対応して、サウンドメモリ59から音声データを読み出してオーディオ出力として出力するようになされている。

【0032】

50

通信制御部 ( A T M ) 5 5 は、公衆回線等に接続され、その回線を介してデータの送受信を行うようになされている。

【 0 0 3 3 】

入力部 5 7 は、操作装置 1 7 を接続する接続端子部 1 2 A、他の装置 ( 図示せず ) からの画像データを受け取るビデオ入力回路 8 2、および他の装置からの音声データを受け取るオーディオ入力回路 8 3 を有している。

【 0 0 3 4 】

図 6 は、図 5 に示したメイン C P U 4 4、メインメモリ 4 5、第 2 ベクトル処理装置 ( V P E 1 ) 4 8、および G P U 4 9 の詳細な構成を示すブロック図である。

【 0 0 3 5 】

メイン C P U 4 4 を構成する C P U コア ( C O R E ) 9 4 は、所定の命令を実行するようになされている。命令 ( Instruction ) キャッシュ ( I \$ ) は、C P U コア 9 4 に供給する命令を一時的に保持するようになされている。高速なメモリである S P R ( スクラッチメモリ ( scratch pad ram ) ) 9 6 は、C P U コア 9 4 による処理結果を記憶するようになされている。データキャッシュ ( D \$ ) は、C P U コア 9 4 が処理を実行するために用いるデータを一時的に記憶するようになされている。

【 0 0 3 6 】

第 1 ベクトル処理装置 ( V P E 0 ) 7 1 を構成する浮動小数点ベクトルプロセッサユニットおよび内蔵メモリ ( V U - M E M ) 1 0 1 は、6 4 ビット長のマイクロプログラムで動作する浮動小数点ベクトルプロセッサユニット ( V U ) および内蔵メモリ ( M E M ) より構成され、後述するマイクロメモリ ( microMEM ) 9 8 に記憶されているプログラムに従って、内部レジスタと内蔵メモリに記憶されたデータを高速演算するようになされている。

【 0 0 3 7 】

P K E 1 0 2 は、後述するメイン D M A C 4 6 の制御により、転送されてくる V U のマイクロコードをマイクロメモリ 9 8 に展開したり、パックされたデータの packets を V U - M E M 1 0 1 の内蔵メモリ ( M E M ) に展開するようになされている。F M A C ( Floating Multiple Adder Calculation Unit ) 9 9 は、浮動小数点演算を実行するようになされている。D I V ( わり算器 ( Divider ) ) 1 0 0 は、わり算を実行するようになされている。そして、上述したように、メイン C P U 4 4 には、第 1 ベクトル処理装置 ( V P E 0 ) が内蔵されており、共同して非定型処理を行うようになされている。

【 0 0 3 8 】

第 2 ベクトル処理装置 4 8 を構成する浮動小数点ベクトルプロセッサユニットおよび内蔵メモリ ( V U - M E M ) 1 0 7 は、6 4 ビット長のマイクロプログラムで動作する浮動小数点ベクトルプロセッサユニット ( V U ) および内蔵メモリ ( M E M ) より構成され、後述するマイクロメモリ ( microMEM ) 1 0 3 に記憶されているプログラムに従って、内部レジスタと内蔵メモリに記憶されたデータを高速演算するようになされている。

【 0 0 3 9 】

P K E 1 0 8 は、メイン D M A C 4 6 の制御により、転送されてくる V U のマイクロコードをマイクロメモリ 1 0 3 に展開したり、パックされたデータの packets を V U - M E M 1 0 7 の内蔵メモリに展開するようになされている。F M A C ( Floating Multiple Adder Calculation Unit ) 1 0 4 は、浮動小数点演算を実行するようになされている。D I V ( わり算器 ( Divider ) ) 1 0 6 は、わり算を実行するようになされている。そして、メインメモリ 4 5 より供給されたデータに対して、定型処理を施し、処理結果を G P U I F 7 2 を介して G P U 4 9 に供給するようになされている。

【 0 0 4 0 】

メインメモリ 4 5 は、3 次元オブジェクトのデータを記憶し、必要に応じて、第 1 ベクトル処理装置 7 1、第 2 ベクトル処理装置 4 8 にそれぞれ供給するようになされている。また、メイン C P U 4 4 と第 1 ベクトル処理装置 ( V P E 0 ) 7 1 とが共同して作成したディスプレイリストを、メインメモリ 4 5 が内蔵する M F I F O ( Memory F I F O ) に一旦、記憶させ、メインバス 4 1 を介して G P U I F 7 2 に供給するようになされている。これは

10

20

30

40

50

、メインCPU 44および第1ベクトル処理装置71の方が、第2ベクトル処理装置48より優先順位が低いため、第2ベクトル処理装置48がアイドル状態になるまでディスプレイリストを保持しておく必要があるからである。

【0041】

また、CPU 44および第1ベクトル処理装置71は、共同して第2ベクトル処理装置48が処理するためのマトリクスを作成し、メインメモリ45に記憶させるようになされている。そして、第2ベクトル処理装置48は、このマトリクスを用いて、ディスプレイリストを作成するようになされている。

【0042】

GPU 49は、第1ベクトル処理装置71よりGPUIF 72を介して供給される非定型処理用のディスプレイリストと、第2ベクトル処理装置48より供給される定型処理用のディスプレイリストを処理するために、それぞれに対応する描画時のクリップ範囲や描画オフセット等のコンテキスト(描画設定条件)を保持するようになされている。CG0は非定型処理用のグラフィックコンテキストであり、CG1は、定型処理用のグラフィックコンテキストである。

10

【0043】

例えば、DMAC 109の制御により、メインメモリ45からメインバス41を介して第1ベクトル処理装置71に供給されたマイクロコードやパックされたデータ(3次元オブジェクトのデータなど)のパケットは、PKE 102により、マイクロメモリ98や内蔵メモリに展開される。そして、FMAC 99およびDIV 100により、3次元オブジェクトのデータに対して、マトリクス演算、座標変換、透視変換等の演算が施される。また、このとき、CPUコア94と共に、より複雑な処理が施される。例えば、木の葉が風に揺れる様子や、自動車のフロントウィンドウを雨の滴が垂れる様子を描画するためのディスプレイリストが作成される。

20

【0044】

このようにして作成された2次元のオブジェクトを画面に描画するためのディスプレイリスト(Complex Stream)は、メインバス41を介して一旦メインメモリ45のMFIFOに記憶され、GPUIF 72に供給される。

【0045】

一方、メインDMAC 46の制御により、メインメモリ45からメインバス41を介して第2ベクトル処理装置48に供給されたマイクロコードやパックされたデータ(3次元オブジェクトのデータなど)のパケットは、PKE 108により、マイクロメモリ103や内蔵メモリに展開される。そして、FMAC 104およびDIV 106により、メインメモリ45よりメインバス41を介して供給される、メインCPU 44および第1ベクトル処理装置71が作成したマトリクスやコンテキストに基づいて、3次元オブジェクトのデータに対して、マトリクス演算、座標変換、透視変換等の演算が施される。ここでは、比較的簡単な定型処理が施される。

30

【0046】

そして、このようにして作成された2次元のオブジェクトを画面に描画するためのディスプレイリスト(Simple Stream)は、メインバス41を介してGPUIF 72に供給される。GPUIF 72は、これら2つのストリームを調停し、それらを時分割でGPU 49に転送する。

40

【0047】

そして、GPU 49により、GPUIF 72より供給されたディスプレイリストに基づいて、描画処理が実行され、フレームメモリ58にポリゴンが描画される。ここで、ディスプレイリストがメインCPU 44および第1ベクトル処理装置71によってメインメモリ45上に作成され、メインバス41を介して供給されたものである場合、GPU 49においては、グラフィックコンテキスト(GC)0を用いて描画処理が行われ、ディスプレイリストが第2ベクトル処理装置48によって作成されたものである場合、グラフィックコンテキスト(GC)1を用いて描画処理が行われる。

50

## 【 0 0 4 8 】

フレームメモリ 5 8 に描画されたポリゴンは、C R T C 8 4 の制御により、対応するビデオ信号に変換された後、出力される。

## 【 0 0 4 9 】

図 7 は、2 つのディスプレイリストが処理されるタイミングを示す図である。図 7 における Geometry Subsystem 0 は、図 6 のブロック図においては第 2 ベクトル処理装置 4 8 に対応し、Geometry Subsystem 1 はメイン C P U 4 4 および第 1 ベクトル処理装置 7 1 に対応している。そして、Rendering Subsystem は G P U 4 9 に対応している。

## 【 0 0 5 0 】

図 7 ( A ) は、プロセッサが 1 つの場合の処理手順を示している。即ち、Geometry Subsystem 0 がディスプレイリスト ( List # 0 - 1 ) を作成すると、それを Rendering Subsystem に供給し、次のディスプレイリスト ( List # 0 - 2 以降のディスプレイリスト ) の作成を続ける。Rendering Subsystem は、供給されたディスプレイリスト ( List # 0 - 1 ) に従って、描画処理を実行する。描画処理が終了した時点で、Geometry Subsystem 0 が次のディスプレイリスト ( List # 0 - 2 ) の作成をまだ行っている場合、Rendering Subsystem は Geometry Subsystem 0 がディスプレイリスト ( List # 0 - 2 ) の作成を終了し、その供給を受けるまでの間、アイドル状態となる。

10

## 【 0 0 5 1 】

以下同様にして、Rendering Subsystem による描画処理が終了しても、Geometry Subsystem 0 による次のディスプレイリストの作成が終了していない場合、Rendering Subsystem は、次のディスプレイリストが Geometry Subsystem 0 から供給されるまでの間、アイドル状態となる。

20

## 【 0 0 5 2 】

図 7 ( B ) は、プロセッサが 2 つの場合の処理手順を示している。即ち、Geometry Subsystem 0 がディスプレイリスト ( List # 0 - 1 ) を作成する間、Rendering Subsystem はアイドル状態であるので、Geometry Subsystem 1 が既に作成し、メインメモリ 4 5 に記憶させておいたディスプレイリスト ( List # 1 - 1 ) を Rendering Subsystem に供給する。Geometry Subsystem 1 からのディスプレイリスト ( List # 1 - 1 ) の供給を受けた Rendering Subsystem は、ディスプレイリストに付加されて供給された Geometry Subsystem 1 に対応するコンテキストに基づいて、描画処理を実行する。

30

## 【 0 0 5 3 】

Geometry Subsystem 0 によるディスプレイリスト ( List # 0 - 1 ) の作成が終了すると、Geometry Subsystem 1 は、次のディスプレイリスト ( List # 1 - 2 ) の Rendering Subsystem への供給を中止する。これにより、Geometry Subsystem 0 は、作成したディスプレイリスト ( List # 0 - 1 ) を Rendering Subsystem に供給するとともに、次のディスプレイリスト ( List # 0 - 2 ) の作成を開始する。そして、Rendering Subsystem は、供給されたディスプレイリスト ( List # 0 - 1 ) に従って、描画処理を実行する。

## 【 0 0 5 4 】

Rendering Subsystem によるディスプレイリスト ( List # 0 - 1 ) の描画処理が終了したとき、Geometry Subsystem 0 は、まだ、次のディスプレイリスト ( List # 0 - 2 ) を作成中であり、Rendering Subsystem はアイドル状態であるので、Geometry Subsystem 1 は、次のディスプレイリスト ( List # 1 - 2 ) を Rendering Subsystem に供給する。Geometry Subsystem 1 からのディスプレイリスト ( List # 1 - 2 ) の供給を受けた Rendering Subsystem は、そのディスプレイリスト ( List # 1 - 2 ) に従って、描画処理を開始する。

40

## 【 0 0 5 5 】

以下同様にして、Geometry Subsystem 1 は、Geometry Subsystem 0 が処理中であり、Rendering Subsystem がアイドル中のときだけ、作成したディスプレイリストを Rendering Subsystem に供給する。これにより、複数のプロセッサが作成したディスプレイリストを、Rendering Subsystem に効率的に処理させることが可能となる。

## 【 0 0 5 6 】

50

例えば、座標変換処理を補強するために、CPUとは別のサブプロセッサや座標変換装置を設けて、複数の処理装置が共同して1つの描画装置(GPU49)にディスプレイリストを送出するようにした場合、描画装置に送出的ディスプレイリストの容量は膨大なため、短い時間間隔で、描画装置を使用するCPUを切り替えるようにしないと、各CPUのローカルメモリがオーバーフローする。そのため、図7(B)に示したように、各CPUに予め優先順位を設定し、高い優先順位のCPU(マスタCPU)(図6においては、第2ベクトル処理装置48)が描画装置へ送出的リストがなくなったら、直ちにアクセス権を次のCPU(スレーブCPU)(図6においては、メインCPU44と第1ベクトル処理装置71)へ譲るようにする。

【0057】

スレーブCPUは、マスタCPUがディスプレイリストを作成して、それを描画装置に転送する準備ができた時点で、まだ、送すべきディスプレイリストが残っていても、すぐに処理を中断し、描画装置へのアクセス権をマスタCPUに返す。

【0058】

マスタCPUには、高速に処理を実行するがローカルメモリの容量が比較的小さいものが割り当てられ、スレーブCPUにはローカルメモリの容量は比較的大きいが、処理速度が比較的遅いものが割り当てられる。

【0059】

図8に示すように、スレーブCPUのさらにスレーブとなるCPUが接続される場合もある。このような場合、特に低次のスレーブCPUは、大きなディスプレイリストを記憶するためのさらに大きな容量のローカルメモリを必要とする。このため、低次のスレーブCPUは、通常、メインメモリを抱えるメインCPUが受け持つようにしている。

【0060】

描画装置が描画を行うためには、ディスプレイリストに記述された頂点情報と同時に、上述したように、描画時のクリップ範囲や描画オフセット等のコンテキストと呼ばれる環境パラメータ(描画設定条件)が必要である。Rendering Subsystemは、各Geometry Subsystemより供給されたディスプレイリストに基づいた描画処理を、そのディスプレイリストを供給したGeometry Subsystemに対応するコンテキストに従って行う。しかしながら、ディスプレイリストを供給するGeometry Subsystemが切り替わる度に、コンテキストを再設定することは非常に面倒である。そこで、Rendering Subsystem側が、Geometry Subsystemの個数分のコンテキストを保持するようにしている。

【0061】

図7に示したように、コンテキストは、ディスプレイリストに付加され、例えば、描画するオブジェクト毎にGPU49に供給される。従って、GPU49は、オブジェクト毎に、そのオブジェクトに対応するコンテキストに基づいて、描画処理を行うことができる。

【0062】

Geometry SubsystemとRendering Subsystemは、メインバス41を構成するデータバスとアドレスバスを共有し、アドレスバスを介して現在アクセスしているGeometry SubsystemのIDを、またデータバスを介して現在アクセスしているGeometry Subsystemが作成したディスプレイリストをそれぞれRendering Subsystemに供給する。Rendering Subsystemは、供給されたIDに対応するコンテキストを選択し、このコンテキストに基づいてディスプレイリストを解釈し、フレームバッファへの描画を行う。

【0063】

以上のように、複数のプロセッサがその優先順位に従って描画装置を制御することにより、ディスプレイリストを保持するローカルメモリの容量を、プロセッサ毎に最小限にすることができる。これにより、ローカルメモリにコストをかけることなく、ディスプレイリストの作成処理を並列化することができる。また、コンテキストを各描画処理装置に保持させることにより、コンテキストの切り替え時におけるオーバーヘッドを減少させることができる。

【0064】

10

20

30

40

50

次に、複数のプロセッサがデータバスとメインメモリを時分割で共有し、データの転送を行う場合において、データの制御を、転送するデータ自身に埋め込まれたメタ命令に従って行う方法について説明する。

【0065】

図9は、メタ命令のフォーマット例を示している。メタ命令は128ビットで構成され、そのうちの64ビットのみが有効とされる。最初の16ビットには、転送すべきデータのサイズがセットされる。24ビット目乃至31ビット目には、メタ命令のIDがセットされる。そして、32ビット目乃至63ビット目には、転送すべきデータが記憶されているアドレス、または、次に読み出すべきメタ命令が記憶されているアドレスがセットされる。

10

【0066】

データの転送は、データに埋め込まれたメタ命令のIDに応じて、以下のように制御される。

【0067】

IDが「cnt」である場合、メタ命令の次のQWCワードを転送した後、パケットの次の番地のメタ命令を実行する。IDが「cnts」である場合、メタ命令の次のQWCワードをストール制御をしながら転送した後、パケットの次の番地のメタ命令を実行する。IDが「next」の場合、メタ命令に続くQWCワードを転送した後、アドレスで示される番地に記憶されているメタ命令が実行される。

【0068】

また、IDが「ref」の場合、アドレスで表される番地からQWCワードを転送した後、メタ命令の次の番地のメタ命令を実行する。IDが「refs」の場合、ADDR番地からQWCワードをストール制御を行いながら転送した後、メタ命令の次の番地のメタ命令を実行する。

20

【0069】

IDが「call」の場合、メタ命令に続くQWCワードを転送した後、パケットの次の番地をレジスタにプッシュして、アドレスで示される番地に記憶されているメタ命令を実行する。IDが「ret」の場合、メタ命令に続くQWCワードを転送した後、レジスタからポップした番地に記憶されているメタ命令を実行する。IDが「end」の場合、メタ命令に続くQWCワードを転送した後、処理を終了する。

30

【0070】

図10は、メタ命令のIDが「next」の場合の動作を示している。最初に、メインDMAC46は、レジスタDn\_TADR (Tag Address) によって指定されたアドレスから、1ワードをメタ命令ワードとして読み出す。この例の場合、そのメタ命令は「NEXT ADDR = ADDR2, LEN = 8」であるので、続く8qword (quadlet word) (1qword = 128ビット) のデータが転送された後、アドレスADDR2に記憶されているメタ命令「NEXT ADDR = ADDR1, LEN = 2」が実行される。

【0071】

これにより、続く2qwordのデータがDMACの制御により転送された後、アドレスADDR1に記憶されているメタ命令「END ADDR = -, LEN = 8」が実行される。即ち、続く8qwordのデータが転送された後、処理を終了する。

40

【0072】

図11は、メタ命令のIDが「REF」の場合の動作を示している。最初に、メインDMAC46は、レジスタDn\_TADRによって指定されたアドレスから、1ワードをメタ命令ワードとして読み出す。この例の場合、「REF ADDR = ADDR2, LEN = 2」であるので、アドレス2より以降に記憶されている2qwordのデータが転送され、次のメタ命令が実行される。

【0073】

即ち、メタ命令「REF ADDR = ADDR1, LEN = 8」が実行され、ADDR1より以降に記憶されている8qwordのデータが転送され、次のメタ命令が実行される

50

。即ち、メタ命令「END ADDR = - , LEN = 8」が実行され、続く8 wordのデータが転送される。

【0074】

図12は、メタ命令のIDが「CALL」の場合の動作を示している。最初に、メインDMAC46は、レジスタDn\_TADRによって指定されたアドレスから、1ワードをメタ命令ワードとして読み出す。この例の場合、「CALL ADDR = ADDR1 , LEN = 0」であるので、アドレス1に記憶されているメタ命令が実行される。即ち、メタ命令「CALL ADDR = ADDR2 , LEN = 8」が実行され、続く8 wordのデータが転送された後、ADDR2に記憶されているメタ命令が実行される。

【0075】

即ち、メタ命令「RET ADDR = - , LEN = 8」が実行され、続く8 wordのデータが転送された後、リターンし、メタ命令「RET ADDR = - , LEN = 0」が実行される。これにより、次のメタ命令「CALL ADDR = ADDR2 , LEN = 8」が実行され、続く8 wordのデータが転送された後、ADDR2に記憶されているメタ命令が再度実行される。

【0076】

即ち、「RET ADDR = - , LEN = 8」が実行され、続く8 wordのデータが転送された後、リターンし、次にメタ命令「END ADDR = - , LEN = 0」が実行され、処理を終了する。

【0077】

このように、データに埋め込まれたメタ命令に従って、データ転送が制御される。

【0078】

図13は、上述したメタ命令によって、データの転送が制御される様子を示す図である。CPU44がディスプレイリスト(DisplayList0)を作成している間に、1フレーム前に作成されたディスプレイリスト(DisplayList1)が第2ベクトル処理装置(VPE1)48に転送される。

【0079】

まず、メインCPU44および第1ベクトル処理装置71が、図13に示すように、IDが「NEXT」のメタ命令、コンテキスト、IDが「REF」のメタ命令、IDが「REF」のメタ命令、マトリクス、IDが「REF」のメタ命令、マトリクス、IDが「REF」のメタ命令、マトリクス、IDが「REF」のメタ命令、IDが「REF」のメタ命令、マトリクス、IDが「REF」のメタ命令、IDが「REF」のメタ命令、マトリクス、IDが「RET」のマトリクスからなるディスプレイリストを作成する。

【0080】

その間、1フレーム前に作成されたディスプレイリスト(DisplayList1)が第2ベクトル処理装置48に転送される。即ち、IDが「NEXT」のメタ命令が実行され、続くコンテキストが第2ベクトル処理装置48に転送される。次に、IDが「REF」のメタ命令が実行され、メインメモリ45内のオブジェクトデータベースの中のProgram0が参照される。このように、フレーム間で内容が変わらないデータは、ディスプレイリストからメタ命令を使用して参照され、ディスプレイリスト間で共有される。このため、ディスプレイリストの作成においては、フレーム毎に変化する位置データ(マトリクス)の更新だけとなる。

【0081】

なお、オブジェクトデータベースには、3次元物体を記述するための3次元データ(Vertex of Object)、オブジェクトデータを解釈するためのプログラム、およびオブジェクトの装飾にテクスチャマッピングを行う場合に、テクスチャとなる画像データ(Texture Image)が格納されている。

【0082】

次に、IDが「REF」の次のメタ命令が実行され、3次元座標データ(オブジェクトの頂点座標データ)(Vertex of Object0)が参照される。次に、マトリクスが第2ベクト

10

20

30

40

50

ル処理装置 48 に転送され、ID が「REF」の次のメタ命令が実行され、3次元座標データ (Vertex of Object1) が参照される。次に、マトリクスが第2ベクトル処理装置 48 に転送される。

【0083】

そして、ID が「REF」の次のメタ命令が実行され、再び3次元座標データ (Vertex of Object1) が参照され、マトリクスが第2ベクトル処理装置 48 に転送される。そして、ID が「REF」の次のメタ命令が実行され、プログラム (Program3) が参照される。次に、ID が「REF」の次のメタ命令が実行され、テクスチャとなる画像データが転送される。

【0084】

テクスチャとなる画像データがフレームメモリ 58 に記憶されていない場合、オブジェクトデータ (Vertex of Object4) の転送に先立って、テクスチャとなる画像データがフレームメモリ 58 に転送される。テクスチャとなる画像データが MDEC 47 からの解凍データやサブバス 42 からの転送データであり、フレーム毎に変化するものである場合、後述するように、ストール機能が用いられ、データ転送の同期が取られる。

【0085】

画像データが転送される間、第2ベクトル処理装置 48 の処理が一旦停止するので、この期間を最小にするために、画像データの転送期間は、他の DMA チャンネルを停止する。この指定もメタ命令の所定の制御ビットにより行うことができる。例えば、図 9 に示したメタ命令の 24 ビット目および 25 ビット目にこの制御ビットをセットすることができる。

【0086】

次に、ID が「REF」の次のメタ命令が実行され、3次元座標データ (Vertex of Object4) が参照される。次に、マトリクスが第2ベクトル処理装置 48 に転送され、ID が「RET」のメタ命令が実行され、処理を終了する。

【0087】

図 14 は、ストール機能を説明するための図である。デバイス 0 (DEV0) からメインメモリにデータの転送が行われ、メインメモリからデバイス 1 (DEV1) にデータの転送が行われる場合、メインメモリからデバイス 1 にデータが転送されるときメインメモリ上での転送アドレスが、デバイス 0 からメインメモリにデータが転送されるときメインメモリ上での転送アドレスを越える間、メインメモリからデバイス 1 へのデータの転送をストールする。

【0088】

図 13 に示した例の場合、メインメモリ 45 から第2ベクトル処理装置 48 にテクスチャとなる画像データを転送するときメインメモリ 45 上での転送アドレスが、MDEC 47 からメインメモリ 45 にテクスチャとなる画像データの転送が行われるときメインメモリ 45 上での転送アドレスを越える間、メインメモリ 45 から第2ベクトル処理装置 48 へのテクスチャとなる画像データの転送をストールする。このようにして、データの同期が取られる。

【0089】

以上のように、メイン DMAC 46 が、リストを構成するメタ命令を読み出し、それに従って、データを各プロセッサに分配する。従って、データを生成する時点で、転送の順序や形態、あるいは転送の優先順位を予めデータの中にプログラムすることにより、データの性質に依存して、最適な転送を行うことができる。また、データの転送順序をリスト形式で予め記述しておくことにより、メモリ内に無駄な作業用のコピーデータを持つ必要がなく、無駄なメモリアクセスを軽減するとともに、ディスプレイリストのサイズを削減することができる。

【0090】

また、ディスプレイリストのうち、フレーム毎に変更する部分だけを2重化して個別に記憶し、その他の部分は共通のメモリ領域に記憶させるようにすることにより、ディスプレ

10

20

30

40

50

イリストを記憶するためのメモリ容量を削減することができる。従って、少ないメモリ容量で多くのディスプレイリストを記憶することができる。

【0091】

さらに、データに埋め込まれたメタ命令に従ってデータの転送が行われるため、複数のプロセッサ間でのデータの読み出しおよび書き込みの同期を取ることを容易にすることができる、メモリ内にダブルバッファを構成することなく、複数のプロセッサがメモリを共有するようにすることができる。

【0092】

なお、上記実施の形態においては、CD-ROMにデータを記憶させるようにしたが、他の記録媒体を用いるようにすることも可能である。

10

【0093】

【発明の効果】

請求項1に記載の情報処理装置、および請求項5に記載の情報処理方法によれば、第2の転送手段は、第1の生成手段が処理中であり、かつ実行手段が待機中であるとき、第2の命令を実行手段に供給し、実行手段は、第2の命令の供給を受けたとき、第2の命令を実行するようにしたので、複数のプロセッサが効率的に実行手段を制御することができる。

【図面の簡単な説明】

【図1】本発明の情報処理装置を応用した家庭用ゲーム機の一例を示す平面図である。

【図2】図1の家庭用ゲーム機1の正面図である。

【図3】図1の家庭用ゲーム機1の側面図である。

20

【図4】図1の家庭用ゲーム機1で再生されるCD-ROMの一例を示す平面図である。

【図5】図1の家庭用ゲーム機1の内部の電氣的構成例を示すブロック図である。

【図6】図5のメインDMAC46、メインCPU44、第2ベクトル処理装置48、メインメモリ45、GPU49の詳細な構成例を示すブロック図である。

【図7】複数のプロセッサが生成するディスプレイリストの処理手順を示す図である。

【図8】3つのプロセッサがGPU49を制御するようにした場合の家庭用ゲーム機1の他の構成例を示すブロック図である。

【図9】メタ命令のフォーマット例を示す図である。

【図10】メタ命令に従って、データ転送が行われる手順を説明する図である。

【図11】メタ命令に従って、データ転送が行われる手順を説明する図である。

30

【図12】メタ命令に従って、データ転送が行われる手順を説明する図である。

【図13】ディスプレイリストに従って、データ転送が行われる手順を説明する図である。

。

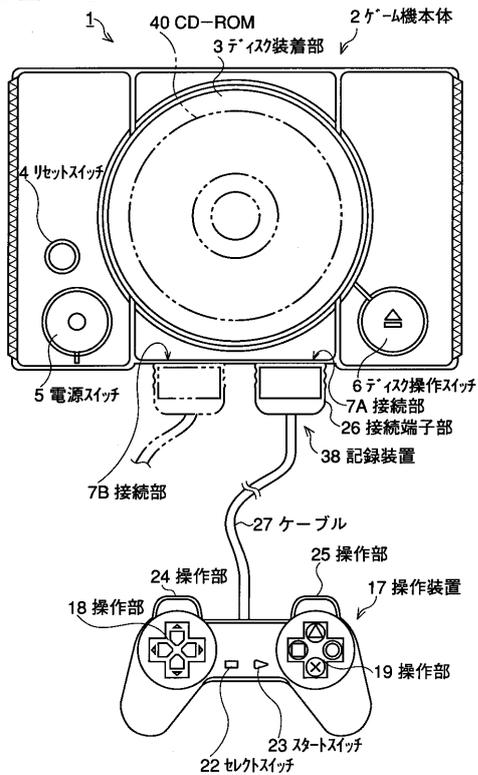
【図14】ストール制御を説明する図である。

【符号の説明】

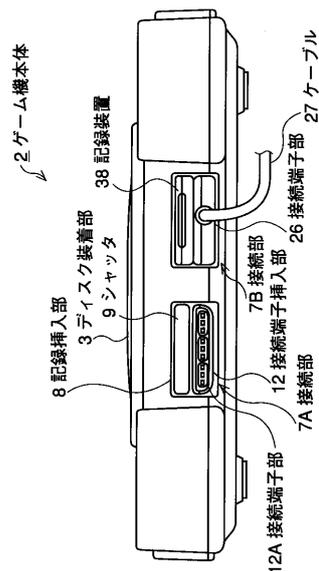
1 家庭用ゲーム機, 2 ゲーム機本体, 3 ディスク装着部, 17 操作装置,  
40 CD-ROM, 41 メインバス, 42 サブバス, 43 SBUSIF  
, 44 メインCPU, 45 メインメモリ, 46 メインDMAC, 47 M  
PEGデコーダ, 48 第2ベクトル処理装置, 49 画像処理装置(GPU), 5  
0 サブCPU, 51 サブメモリ, 52 サブDMAC, 53 ROM, 56  
CD-ROMドライブ, 58 フレームメモリ, 94 CPUコア, 98, 103  
マイクロメモリ, 99, 104 FMAC, 100, 106 DIV, 101, 1  
07 VU-MEM, 102, 108 PKE

40

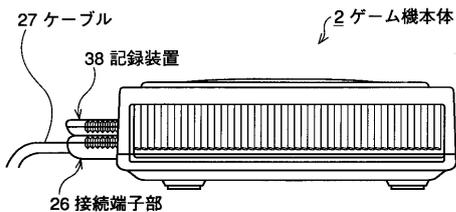
【図1】



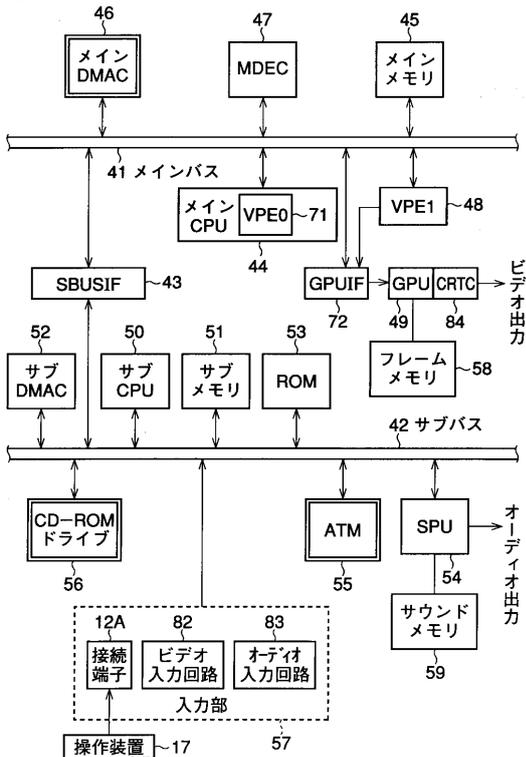
【図2】



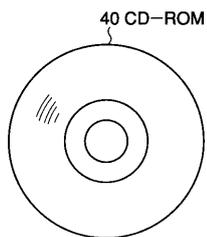
【図3】



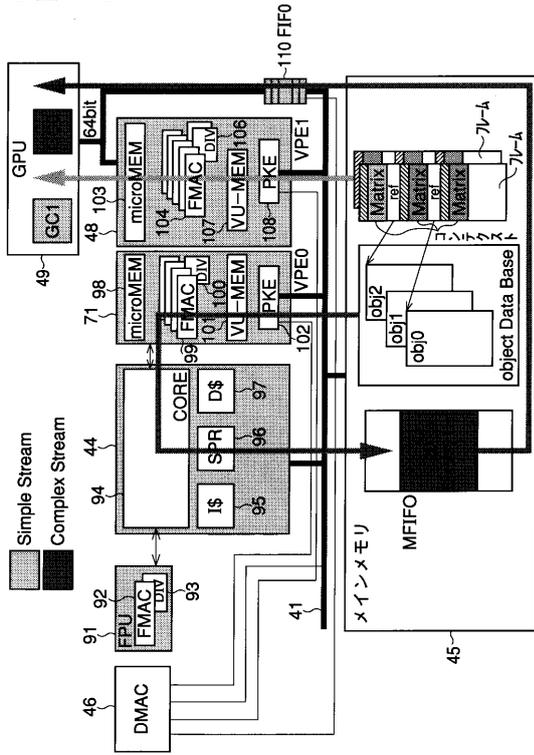
【図5】



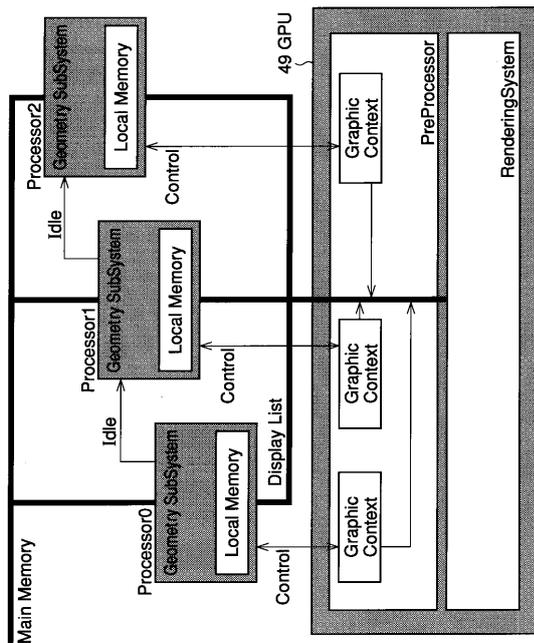
【図4】



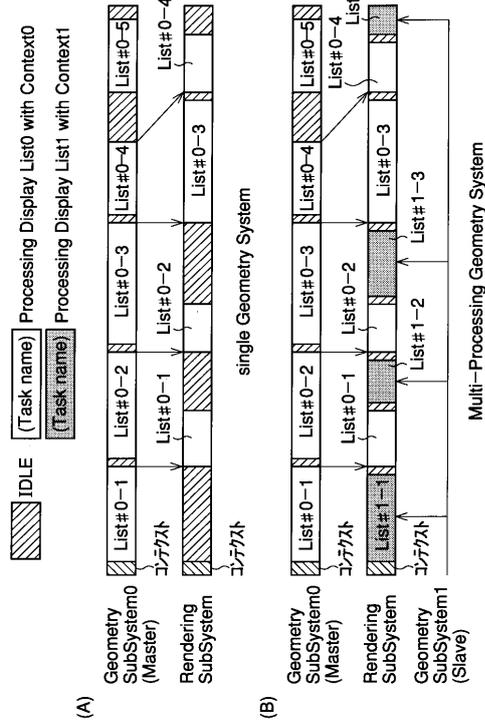
【 図 6 】



【 図 8 】



【 図 7 】

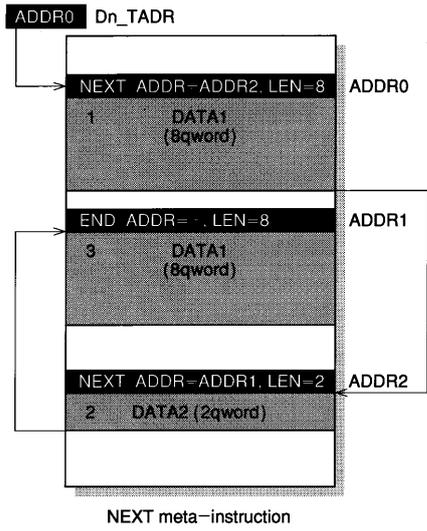


ディスプレイリストフロー

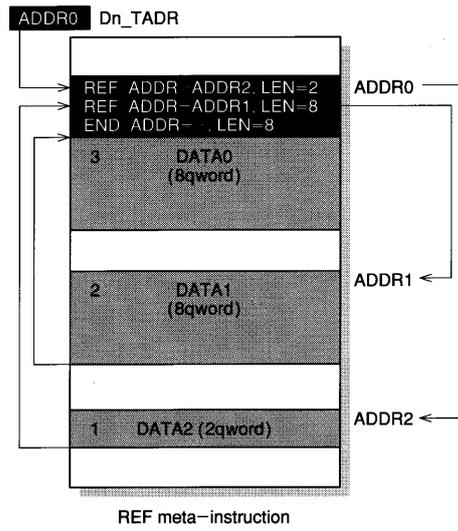
【 図 9 】

63	ADDR	32,31	24	15	0
		ID/FLG		QWC	

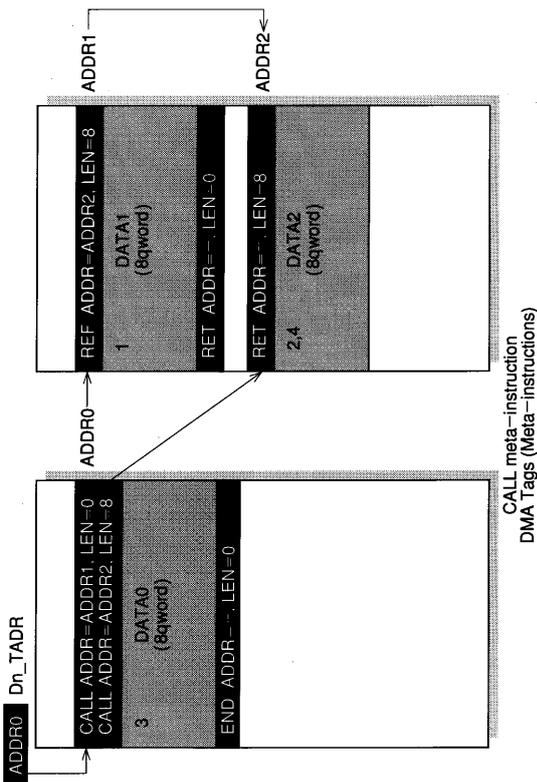
【 1 0 】



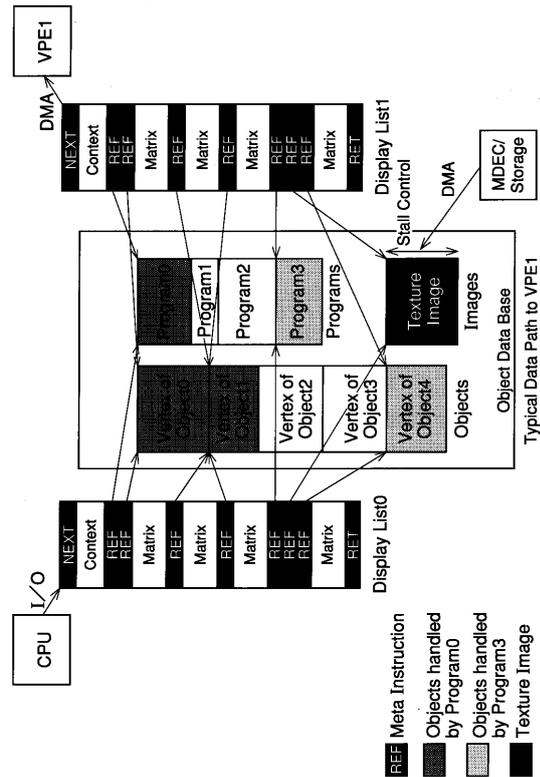
【 1 1 】



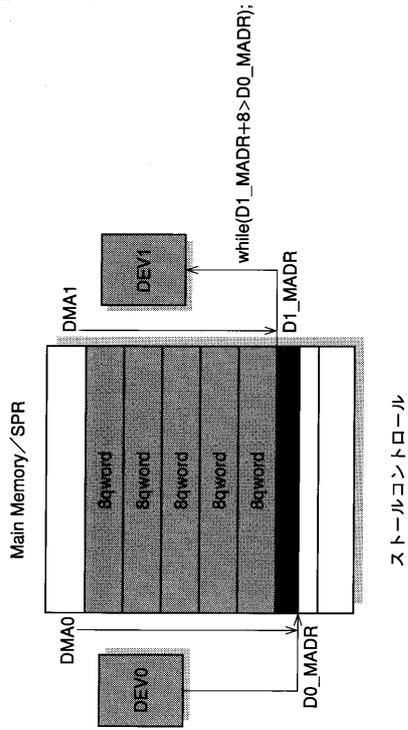
【 1 2 】



【 1 3 】



【 図 14 】



フロントページの続き

(58)調査した分野(Int.Cl.<sup>7</sup>, DB名)

G06T 1/20

G06T 15/00

G06F 15/16

G06F 9/38