

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4836491号  
(P4836491)

(45) 発行日 平成23年12月14日(2011.12.14)

(24) 登録日 平成23年10月7日(2011.10.7)

(51) Int. Cl.		F I			
<b>G06F</b>	<b>13/28</b>	<b>(2006.01)</b>	G06F	13/28	310J
<b>G06T</b>	<b>1/20</b>	<b>(2006.01)</b>	G06T	1/20	B
<b>G06T</b>	<b>1/60</b>	<b>(2006.01)</b>	G06T	1/60	450D
			G06T	1/60	450E

請求項の数 4 (全 27 頁)

(21) 出願番号	特願2005-148230 (P2005-148230)	(73) 特許権者	310021766
(22) 出願日	平成17年5月20日(2005.5.20)		株式会社ソニー・コンピュータエンタテインメント
(65) 公開番号	特開2006-323760 (P2006-323760A)		東京都港区港南1丁目7番1号
(43) 公開日	平成18年11月30日(2006.11.30)	(74) 代理人	100105924
審査請求日	平成20年2月28日(2008.2.28)		弁理士 森下 賢樹
		(74) 代理人	100109047
			弁理士 村田 雄祐
		(74) 代理人	100109081
			弁理士 三木 友由
		(74) 代理人	100134256
			弁理士 青木 武司

最終頁に続く

(54) 【発明の名称】 情報処理装置、システム、方法およびプロセッサ

(57) 【特許請求の範囲】

【請求項1】

装置全体を統括的に制御するメインプロセッサと、特定の処理を行う第2プロセッサとを備える情報処理装置であって、

当該装置は、

前記第2プロセッサにおける前記特定の処理のためのデータが最初に展開されるメインメモリと、

前記メインプロセッサおよび前記第2プロセッサによって実行される、複数のコマンドのセットからなる所与のコマンドリストにしたがって、前記メインメモリから同時にm個(mはm>1の整数)のデータ転送を実行可能なDMAコントローラと、

前記メインメモリからの前記データの転送先であって、前記DMAコントローラにより同時に実行可能なデータ転送数よりも多い数のバッファと、を備え、

前記コマンドリストは、前記メインメモリからのm個のデータの転送先として指定されるバッファの組み合わせを、前記第2プロセッサにおけるグラフィック処理のステージにあわせて順次切り替えることによって、データ転送先として非指定の少なくとも1つのバッファが存在するように作成されることを特徴とする情報処理装置。

【請求項2】

前記コマンドリストは、データ転送先として非指定の少なくとも1つのバッファに格納されている、前記DMAコントローラによる一ステージ前のデータ転送で書き込まれたデータを使用した処理を前記第2プロセッサに実行させるように作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 3】

装置全体を統括的に制御するメインプロセッサと、特定の処理を行う第 2 プロセッサとを備える情報処理システムであって、

当該システムは、

前記第 2 プロセッサにおける前記特定の処理のためのデータが最初に展開されるメインメモリと、

前記メインプロセッサおよび前記第 2 プロセッサによって実行される、複数のコマンドのセットからなる所与のコマンドリストにしたがって、前記メインメモリから同時に m 個 ( m は  $m > 1$  の整数 ) のデータ転送を実行可能な DMA コントローラと、

前記メインメモリからの前記データの転送先であって、前記 DMA コントローラにより同時に実行可能なデータ転送数よりも多い数のバッファと、を備え、

前記コマンドリストは、前記メインメモリからの m 個のデータの転送先として指定されるバッファの組み合わせを、前記第 2 プロセッサにおけるグラフィック処理のステージにあわせて順次切り替えることによって、データ転送先として非指定の少なくとも 1 つのバッファが存在するように作成されることを特徴とする情報処理システム。

【請求項 4】

装置全体を統括的に制御するメインプロセッサと、

同時に m 個 ( m は  $m > 1$  の整数 ) のデータ転送を実行可能な DMA コントローラと、

外部のメモリからのデータの転送先であって、前記 DMA コントローラにより同時に実行可能なデータ転送数よりも多い数のバッファを有し、特定の処理を行う第 2 プロセッサと、

を備える装置における方法であって、

前記 DMA コントローラは、前記メインプロセッサおよび前記第 2 プロセッサによって実行される、複数のコマンドのセットからなる所与のコマンドリストにしたがって、前記メインメモリからのデータ転送を実行し、

前記コマンドリストは、前記メモリからの m 個のデータの転送先として指定されるバッファの組み合わせを、前記第 2 プロセッサにおけるグラフィック処理のステージにあわせて順次切り替えることによって、データ転送先として非指定の少なくとも 1 つのバッファが存在するように作成されることを特徴とする情報処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は情報処理技術に関し、特に画像処理を含む情報処理技術に関する。

【背景技術】

【0002】

近年のコンピュータグラフィックス技術の発達に伴い、大型計算機や、パーソナルコンピュータ、ゲーム機器などの情報処理装置から出力される画像データは、ますます複雑化、高度化している。そこで、このような情報処理装置では、通常の演算処理を行うメインプロセッサとは別に、画像処理専用のグラフィックプロセッサを搭載し、メインプロセッサを画像処理から解放して、システムのオーバーヘッドの解消を図っている。

【0003】

これらの情報処理装置において高性能な演算処理を実現するためには、メインプロセッサをマルチプロセッサで構成することが有効である。マルチプロセッサでは、複数のタスクを各プロセッサに割り当て並列処理を実行して演算速度を高める。一方、グラフィックプロセッサは、メインプロセッサが処理する複数のタスクに対応した画像処理を、メインプロセッサの高速演算に対応させて行う必要がある。

【発明の開示】

【発明が解決しようとする課題】

【0004】

10

20

30

40

50

一般に、グラフィックプロセッサにおける画像処理は、メインプロセッサで実行されるアプリケーションに関わりなく動作する。しかしながら、アプリケーションの実行の上で必要となる画像処理に協調させてグラフィックプロセッサの動作を制御することができれば、情報処理装置における処理効率の向上に寄与することもあり得る。

【0005】

本発明はこのような課題に鑑みてなされたものであり、その目的はマルチプロセッサシステムにおける情報処理効率を向上させる技術の提供にある。

【課題を解決するための手段】

【0006】

本発明のある態様は、装置全体を統括的に制御するメインプロセッサと、特定の処理を行う第2プロセッサとを備える情報処理装置に関する。この装置は、同時にm個(mはm>1の整数)データ転送を制御するDMAコントローラと、特定の処理のためのデータが最初に展開されるメインメモリと、メインメモリからデータが転送されたときこれらを記憶するn個(nはn>mの整数)で一組のバッファと、を備える。そして、同時に複数のデータ転送が実行されるとき、前記一組のバッファのうち第1のバッファが一方のデータの転送先に設定され、第2のバッファが他方のデータの転送先に設定されることを特徴とする。

10

【0007】

ここで「DMAコントローラ」とは、プロセッサを介さずにメモリまたは外部の入出力装置との間でデータを転送する回路のことをいう。また、ここでいう「データ」には、テクスチャデータ、カラーデータ、値データ、Z値データ、頂点データなどのほか、グラフィックユニットにおけるシェーダプログラムなどのプログラムも含まれる。

20

【0008】

この態様によれば、メインメモリから同時にDMA転送されるデータより多数のバッファが設けられているので、メインプロセッサおよび第2プロセッサのアイドル時間が削減され、情報処理装置全体の処理効率を向上させることができる。

【発明の効果】

【0009】

本発明によれば、複数のプロセッサを有する情報処理装置において、プロセッサのアイドル時間を削減することができる。

30

【発明を実施するための最良の形態】

【0010】

図1は、本発明の実施の形態に係る情報処理装置1000の構成を示すブロック図である。情報処理装置1000は、グラフィックプロセッサ100、メインプロセッサ200、メインメモリ50およびグラフィックメモリ10を含む。情報処理装置1000は、表示装置52と接続されている。なお、図1において、様々な処理を行う機能ブロックとして記載される各要素は、ハードウェア的には、CPU、メモリ、その他のLSIで構成することができ、ソフトウェア的には、メモリにロードされたプログラムなどによって実現される。したがって、これらの機能ブロックがハードウェアのみ、ソフトウェアのみ、またはそれらの組合せによっていろいろな形で実現できることは当業者には理解されること

40

【0011】

この情報処理装置1000では、情報処理装置1000を効率よく使用するための機能、環境を提供し、装置全体を統括的に制御するオペレーティングシステム(以下、OSと略す)が実行される。そのOS上でアプリケーションソフトウェア(以下、単にアプリケーションという)が実行される。本実施の形態においては、メインプロセッサ200とグラフィックプロセッサ100は協調的に演算処理を行う。グラフィックプロセッサ100は、メインプロセッサ200により処理されるそれぞれのタスクに関連する画像処理を行い、生成した画像、映像を表示装置52に出力し、または図示しない記憶装置に記憶する。メインプロセッサ200とグラフィックプロセッサ100との協調動作については後述

50

する。

【0012】

メインプロセッサ200は、複数のサブプロセッサ30と管理プロセッサ32と制御側DMA(Direct Memory Access)コントローラ34とを含む。サブプロセッサ30、管理プロセッサ32および制御側DMAコントローラ34はバス36により相互通信が可能である。管理プロセッサ32は、アプリケーションに対応するタスクを時分割し、タイムスライス毎に各アプリケーションに対応するタスクをサブプロセッサ30に割り当て、演算処理を並列で行うように制御する。例えば、あるタイムスライスにおいて処理されるアプリケーションに対応するタスクが三次元コンピュータグラフィックスの画像処理に関するものである場合には、ジオメトリ処理を実行する。各サブプロセッサ30において画像処理の必要が発生した場合には、ジオメトリ処理後の頂点データやパラメータデータが各サブプロセッサ30からグラフィックメモリ10へ転送される。

10

【0013】

制御側DMAコントローラ34は、メインメモリ50やグラフィックプロセッサ100との間のデータ転送、データ退避などをサブプロセッサ30または管理プロセッサ32からのコマンドによって制御する。

【0014】

メインメモリ50は、主にメインプロセッサ200によって使用される記憶領域である。このメインメモリ50には、メインプロセッサ200によりコンピュータグラフィックスに関するタスクを処理して得られたデータやプログラムなどが一時的に格納される。また、このメインメモリ50には、グラフィックプロセッサ100による演算処理の結果得られたデータが退避して格納される場合もある。

20

【0015】

グラフィックメモリ10は、グラフィックプロセッサ100により使用、管理されるグラフィックに関するデータ専用のメモリ領域であって、画像フレームデータが格納されるフレームバッファ(図示せず)などに加えて、画像フレームデータを描画する際に参照される基本データである頂点データを格納するバッファ(図示せず)、および、テクスチャデータが一時的に格納されるコンテキストバッファ12を2個以上有している。

【0016】

グラフィックプロセッサ100は、画像に関する処理、例えばレンダリング処理などを専用に行う。グラフィックプロセッサ100は、ディスプレイコントローラ18、制御ユニット20、画像処理側DMAコントローラ28、グラフィック演算ユニット40、および内部レジスタ54を含む。これらのブロック同士はバス36で接続されており、各ブロック間でデータ信号の送受が行われる。

30

【0017】

グラフィック演算ユニット40は、外部から与えられる命令にしたがって、グラフィックに関する様々な演算処理を行う。その処理の一例としては、三次元モデリングデータをもとに座標変換、陰面消去、シェーディングを行って画像フレームデータを作成し、グラフィックメモリ10内のフレームバッファに書き込む一連のレンダリング処理などが挙げられる。グラフィック演算ユニット40は、特に三次元グラフィックスに関する処理を高速に行うために、ラスタライザ42、シェーダユニット44、テクスチャユニット46などの機能ブロックを含んでいる。

40

【0018】

ラスタライザ42は、描画するオブジェクトの頂点データを受け取り、三次元空間上のオブジェクトを投影変換により描画平面上の図形に変換するビュー変換を行う。さらに、描画平面上の図形を、描画平面の水平方向に沿ってスキャンしながら、一列ごとに量子化されたピクセルに変換するラスタ処理を行う。このラスタライザ42によってオブジェクトがピクセル展開されて、各ピクセル毎にピクセル情報を算出する。このピクセル情報には、RGBカラー値、透明度を表す値、視点からの奥行きを表すZ値が含まれる。

【0019】

50

ラスタライザ 4 2 は、スキャンラインに沿って所定の大きさのピクセル領域を生成し、シェーダユニット 4 4、テクスチャユニット 4 6 へと出力する。ラスタライザ 4 2 から出力されるピクセル領域は、一度キューにスタックされ、シェーダユニット 4 4 はスタックされたピクセル領域を順に処理していく。

【 0 0 2 0 】

シェーダユニット 4 4 は、ラスタライザ 4 2 により算出されたピクセル情報をもとにシェーディング処理を行い、テクスチャユニット 4 6 により得られたテクセル情報をもとに、テクスチャマッピング後のピクセル色を決定し、グラフィックメモリ 1 0 内のフレームバッファにシェーディング処理後の画像フレームデータを書き込む。さらにシェーダユニット 4 4 は、フレームバッファに書き込まれた画像フレームデータに対してフォギング、

10

アルファブレンディング等の処理を行い最終的な描画色を決定してフレームバッファの画像フレームデータを更新する。

【 0 0 2 1 】

テクスチャユニット 4 6 は、シェーダユニット 4 4 からテクスチャデータを指定するパラメータを受け取り、要求されたテクスチャデータをグラフィックメモリ 1 0 内のバッファ 1 2 から読み出し、所定の処理を行った後にシェーダユニット 4 4 に対して出力する。

【 0 0 2 2 】

制御ユニット 2 0 は、このグラフィックプロセッサ 1 0 0 全体を制御するブロックであり、ディスプレイコントローラ 1 8、グラフィック演算ユニット 4 0 を統括的に制御し、各ブロック間のデータ転送の同期管理、割り込み処理、タイマー管理等を行う。

20

【 0 0 2 3 】

ディスプレイコントローラ 1 8 は、水平および垂直同期信号を生成し、表示装置 5 2 の表示タイミングにしたがって、グラフィックメモリ 1 0 内のフレームバッファから画像フレームデータのピクセルデータをライン状に順次読み込んでいく。さらにディスプレイコントローラ 1 8 は、ライン状に読み込まれたピクセルデータを、R G B のカラー値からなるデジタルデータから表示装置 5 2 に対応したフォーマットに変換して出力する。

【 0 0 2 4 】

画像処理側 D M A コントローラ 2 8 は、メインメモリ 5 0 とグラフィックメモリ 1 0 間のデータ転送、データ退避などを制御ユニット 2 0 からのコマンドによって制御する。画像処理側 D M A コントローラ 2 8 は、テクスチャデータをメインメモリ 5 0 からグラフィックメモリ 1 0 のバッファ 1 2 に転送する。

30

【 0 0 2 5 】

次に、以上のように構成された情報処理装置 1 0 0 0 の動作について、本実施の形態の特徴であるメインプロセッサ 2 0 0 とグラフィックプロセッサ 1 0 0 との協調動作を中心に説明する。

【 0 0 2 6 】

メインプロセッサ 2 0 0 の管理プロセッサ 3 2 は、時分割されたタスクの実行順序を決定し、各タスクをサブプロセッサ 3 0 に適切に割り当てるとともに、グラフィックプロセッサ 1 0 0 との信号の送受信を行う。グラフィックプロセッサ 1 0 0 の制御ユニット 2 0 は、メインプロセッサ 2 0 0 と信号の送受信を行うほか、メインプロセッサ 2 0 0 から受け取るコマンドリストに基づいて、画像処理側 D M A コントローラ 2 8 とグラフィック演算ユニット 4 0 を統括的に制御する。グラフィック演算ユニット 4 0 は、制御ユニット 2 0 で実行されるコマンドにしたがって、上述の各種グラフィック処理を開始する。

40

【 0 0 2 7 】

制御ユニット 2 0 は、インタフェイス部 2 2 と、命令解析部 2 4 と、実行部 2 6 とを含む。インタフェイス部 2 2 は、メインプロセッサ 2 0 0 内のサブプロセッサ 3 0 やメインメモリ 5 0 などからデータを受け取る。なお、ここでいう「データ」には、コマンドリスト、頂点データ、テクスチャデータなどの狭義のデータの他、グラフィック演算ユニット 4 0 におけるシェーダプログラムなどが含まれる。命令解析部 2 4 は、インタフェイス部 2 2 が受け取ったデータの中からコマンドリストを取り出して、コマンドを解析する。実

50

行部 26 は、コマンドを順次実行する。

【0028】

グラフィック演算ユニット 40 は、シェーダプログラムなどのプログラムで規定された演算方法にしたがって、実行部 26 で実行されるコマンドにより指示されるタイミングでオブジェクトのグラフィック処理を実行する。

【0029】

タスクの処理を実行しているあるサブプロセッサ 30 に画像処理の必要が発生したとすると、サブプロセッサ 30 は、画像処理に必要な頂点データなどのオブジェクトデータをメインメモリ 50 に一時格納するとともに、これらのデータを転送するためのコンフィグデータとコマンドリストを作成する。コンフィグデータには、シェーダプログラムなどのグラフィック演算ユニット 40 で使用されるプログラムのアドレス、転送サイズなどのデータが含まれる。

10

【0030】

制御側 DMA コントローラ 34 は、メインメモリ 50 からオブジェクトデータを読み出し、透視変換などの処理を施した後、読み出したオブジェクトデータをグラフィックメモリ 10 に転送する。また、制御側 DMA コントローラ 34 は、コマンドリストやコンフィグデータを制御ユニット 20 に転送する。制御ユニット 20 に転送されるデータは、命令解析部 24 により解析され、解析されたコマンドが実行部 26 に渡されキューに入れられる。実行部 26 は、キュー内のコマンドを順次実行する。実行されたコマンドにしたがって、画像処理側 DMA コントローラ 28 は、テクスチャデータをメインメモリ 50 から読み出してバッファ 12 に転送する。

20

制御側 DMA コントローラ 34 は、管理プロセッサ 32 またはサブプロセッサ 30 の指示により実行開始する。サブプロセッサ 30 は、オブジェクトの転送準備ができた時に、後述する NOTIFY コマンドによりバッファの空きを確認して、制御側 DMA コントローラ 34 を開始させる。制御側 DMA コントローラ 34 によるサブプロセッサ 30 からグラフィックメモリ 10 へのオブジェクトデータの転送は、テクスチャデータの転送に並行して、またはグラフィック演算ユニット 40 のアイドルリング中に実行される。

【0031】

このように、グラフィックプロセッサ 100 は、内部に設けられた画像処理側 DMA コントローラ 28 を利用して、画像処理に必要な各種データを好適なタイミングでメインメモリ 50 から取得することができる。これによって、データ転送処理に係るグラフィックプロセッサ 100 およびメインプロセッサ 200 の負荷を軽減し、転送要求など最低限の処理を除いて、それらのプロセッサの演算処理と並行して転送処理を行うことができる。また、グラフィックプロセッサ 100 の構成を複雑化することなく、高速動作するメインプロセッサ 200 の処理に対応した画像処理を行うことができるため、動画の停止やコマ落ちなどが発生しにくくなる。

30

【0032】

本実施の形態において、制御ユニット 20 で解析、実行されるコマンドを「非グラフィック処理系命令」と呼ぶ。この非グラフィック処理系命令は、グラフィック演算ユニット 40 におけるグラフィック処理内容が明示されているものではなく、描画データの転送、グラフィック処理の開始や処理の待機、終了などの処理内容のみが明示されている命令系である。グラフィック演算ユニット 40 における具体的なグラフィック処理は、コンフィグデータで指定されるシェーダプログラムなどのプログラムで規定されている。

40

【0033】

このような非グラフィック処理系命令を採用したのは、以下のようなことを目的とするためである。すなわち、本実施の形態のような情報処理装置においては、グラフィックプロセッサやそこで実行されるシェーダプログラムは、ハードウェアベンダーにより提供され、情報処理装置において実行されるアプリケーションソフトウェアや映像コンテンツは、別のソフトウェアベンダーにより提供されることが多い。このような場合、ソフトウェアベンダーがグラフィックプロセッサの制御を考慮する必要はなく、処理をすべきデータ

50

のみを準備すればよい。

【0034】

しかしながら、アプリケーション毎に要求される処理の負荷などは異なるので、情報処理装置全体の効率を考えた場合、グラフィックプロセッサにおいて実行されるグラフィック処理をソフトウェアベンダー側で制御することが好ましい場合がある。また、ソフトウェアベンダー側において、グラフィックプロセッサのハードウェア構成を理解した上で、グラフィックプロセッサにおけるグラフィック処理を一定程度制御できた方が、グラフィック処理の高速化、効率化に寄与する場合もある。

そこで、グラフィック演算ユニット40で実行されるグラフィック処理の方法を明示しない命令である「非グラフィック処理系命令」を設けることで、グラフィックプロセッサの動作を簡便に制御できるようにしたのである。

10

【0035】

次に、非グラフィック処理系命令を具体的なコマンドを参照して説明する。本実施の形態では、非グラフィック処理系命令として、転送命令「DMA」、処理起動命令「KICK」、処理完了通知命令「NOTIFY」、同期化命令「SYNC」、待機命令「WAIT」、終了命令「END」、次リスト命令「NEXT」および計時命令「PERF」を有している。当然であるが、他の命令をさらに有していても、またはこれらの命令のうちいくつかの命令を含んでいない場合も、本発明の範囲に含まれる。

【0036】

上述の非グラフィック処理系命令は、全て同一のフォーマットを備えている。図2は、コマンドフォーマットの一例を示す。フォーマット120は、上述のコマンドのうちいずれのコマンドであるかを数字で指定するタイプ121、コマンド毎に設定が必要となる属性122、コマンド毎に付与されるコマンドID123、および必要な設定を明示するパラメータ124～126をそれぞれ指定するデータ領域を有する。コマンド長は、例えば16バイトである。「コマンドID」は、「1」から開始し「1」ずつ増加していき、コマンドリスト中に複数存在するDMAコマンドやKICKコマンドの識別番号として使用される。

20

【0037】

グラフィックプロセッサ100で処理されるのは、上述のKICK、DMA、NOTIFYなどを含む複数のコマンドのセットであり、このセットを本明細書においては「コマンドリスト」と呼ぶ。制御ユニット20の実行部26は、コマンドリスト内のコマンドをひとつずつ実行して、画像処理側DMAコントローラ28またはグラフィック演算ユニット40に対応する命令を実行させる。

30

【0038】

図3は、DMAコマンドのデータフォーマット130を示す。DMAコマンドは、画像処理側DMAコントローラ28に対して、メインメモリ50とグラフィックメモリ10の間の任意のデータのDMA転送を実行させる命令である。制御ユニット20の実行部26がDMAコマンドを実行すると、画像処理側DMAコントローラ28が起動し、メインメモリ50からテクスチャデータの読み込み処理を開始する。読み込まれたデータは、グラフィックメモリ10のバッファ12に格納される。

40

【0039】

タイプ131にはDMAコマンドであることを示す「1」を指定する。DMAチャンネル132はDMA転送を利用するデバイス間で衝突が発生しないように設定される2つのチャンネルのうちいずれかを指定する。DMA ID133は、DMAコマンドのIDを指定する。このDMA IDは、DMAのプロセスが終了すると直ちにグラフィックプロセッサ100の内部レジスタ54に書き込まれる。サブプロセッサ30は、DMA IDを格納した内部レジスタ54をポーリングすることで、DMAプロセスの終了を確認することもできる。タグアドレス134にはDMAタグのアドレスが指定される。DMAタグには、DMA転送に関連する情報、すなわち、アドレス、DMAサイズ、および転送方向が含まれる。タグサイズ135は、DMAタグのデータサイズが指定される。

50

## 【 0 0 4 0 】

図 4 は、K I C K コマンドのデータフォーマット 1 4 0 を示す。K I C K コマンドは、グラフィック演算ユニット 4 0 における各種グラフィック処理を開始させる命令である。制御ユニット 2 0 の実行部 2 6 が K I C K コマンドを実行すると、グラフィック演算ユニット 4 0 はグラフィックメモリ 1 0 にアクセスしてテクスチャデータとオブジェクトデータを読み出し、フレームバッファに描画する。

## 【 0 0 4 1 】

タイプ 1 4 1 には K I C K コマンドであることを示す「 2 」を指定する。コンフィグタグ 1 4 2 は、コンフィグデータを使用するか否かを指定する。K I C K I D 1 4 3 は、K I C K コマンドの I D を指定する。この K I C K I D は、K I C K プロセスが終了すると直ちにグラフィックプロセッサ 1 0 0 の内部レジスタ 5 4 に書き込まれる。サブプロセッサ 3 0 は、K I C K I D を格納した内部レジスタ 5 4 をポーリングすることで、K I C K プロセスの終了を確認することもできる。コンフィグデータアドレス 1 4 4 およびコンフィグデータサイズ 1 4 5 には、それぞれコンフィグデータのアドレスまたはデータサイズが指定される。制御ユニット 2 0 の実行部 2 6 は、K I C K コマンドを実行すると、コンフィグデータをグラフィック演算ユニット 4 0 に転送した後に、グラフィック演算ユニット 4 0 が起動される。グラフィック演算ユニット 4 0 は、コンフィグデータの内容にしたがってプログラムをロードした後、グラフィック処理を実行する。コンフィグデータは、予め決定されているか、もしくはメインプロセッサ 2 0 0 において作成される。

## 【 0 0 4 2 】

図 5 は、N O T I F Y コマンドのデータフォーマット 1 5 0 を示す。N O T I F Y コマンドは、グラフィックプロセッサ 1 0 0 からメインプロセッサ 2 0 0 に対して通知をするために使用される。制御ユニット 2 0 の実行部 2 6 が N O T I F Y コマンドを実行すると、グラフィックプロセッサ 1 0 0 内における所定のグラフィック処理の終了が、メインプロセッサ 2 0 0 内のサブプロセッサ 3 0 に通知される。具体的には、N O T I F Y コマンドは、画像処理側 D M A コントローラ 2 8 による D M A 転送が終了したことを、または、グラフィック演算ユニット 4 0 におけるグラフィック処理が終了したことをサブプロセッサ 3 0 に通知する。グラフィックプロセッサ 1 0 0 においていずれの処理の終了を通知するかは、コマンドリストにおける N O T I F Y コマンドと他のコマンドとの配置で指定することができる。

## 【 0 0 4 3 】

タイプ 1 5 1 には N O T I F Y コマンドであることを示す「 3 」を指定する。サブプロセッサ I D 1 5 4 は、信号を通知するサブプロセッサ 3 0 を指定する。全てのサブプロセッサ 3 0 を指定することも可能である。

## 【 0 0 4 4 】

例えば、グラフィックメモリには、限られた数しかコンテキストバッファ 1 2 が設けられておらず、これをグラフィック処理に必要なデータ（例えばテクスチャデータ）の格納領域として共用しなければならない。そこで、例えば、グラフィック処理に必要なデータの D M A 転送が終了したことを N O T I F Y コマンドでサブプロセッサ 3 0 に伝え、これに応じてサブプロセッサ 3 0 によりジオメトリ処理の結果、例えば頂点データなどをグラフィックメモリ 1 0 に転送するようになれば、グラフィック演算ユニット 4 0 およびサブプロセッサ 3 0 のアイドル時間が削減され、結果的に演算時間の短縮に寄与することができる。このように、N O T I F Y コマンドは、グラフィックプロセッサ 1 0 0 とメインプロセッサ 2 0 0 とを協調動作させ、情報処理装置 1 0 0 0 全体の処理効率を向上させるために必要となる命令である。N O T I F Y コマンドのより具体的な使用方法は、図 1 1 ~ 図 1 7 を参照する実施例において説明する。

## 【 0 0 4 5 】

図 6 は、S Y N C コマンドのデータフォーマット 1 6 0 を示す。S Y N C コマンドは、プロセス間に依存関係がある場合に、所定のコマンドプロセスの終了と、サブプロセッサ 3 0 における処理の終了とを同期させるコマンドである。



## 【 0 0 4 6 】

タイプ 1 6 1 には S Y N C コマンドであることを示す「 4 」を指定する。S Y N C 対象 1 6 2 には、D M A コマンド、K I C K コマンド、サブプロセッサからのオブジェクトの D M A 転送のいずれの終了を確認するかを指定する。I D 1 6 4 には、D M A コマンドまたは K I C K コマンドの I D を指定する。レジスタ 1 6 5 は、フラグレジスタを指定する。サブプロセッサ 3 0 は、自身の処理が終了すると、内部レジスタ 5 4 にフラグを立てる。後述するように、S Y N C コマンドによって、テクスチャデータとオブジェクトデータの両方の転送終了待ちをすることができる。

## 【 0 0 4 7 】

実行部 2 6 は、内部レジスタ 5 4 を参照することで、D M A コマンド、K I C K コマンドまたはサブプロセッサの終了を確認する。実行部 2 6 が S Y N C コマンドを実行すると、フォーマットで指定されたプロセス依存関係に応じて、以下に述べるような複数の動作が発生する。第 1 に、時間的に前に実行された D M A コマンドにしたがったデータの D M A 転送が完了するまで、グラフィック演算ユニット 4 0 におけるそのデータを利用するグラフィック処理の開始を保留させる。第 2 に、時間的に前に実行された K I C K コマンドによってグラフィック演算ユニット 4 0 においてグラフィック処理が実行されているときに、その処理が終了するまでコマンドリスト内の後続のコマンドの実行を保留する。第 3 に、サブプロセッサ 3 0 におけるデータ処理が終了するまで、コマンドリスト内の後続のコマンドの実行を保留する。

## 【 0 0 4 8 】

例えば、K I C K コマンドを実行しようとする場合、グラフィック処理に必要なテクスチャデータなどのデータがグラフィックメモリ上に揃わなければ、グラフィック演算ユニット 4 0 を起動することはできない。そのような場合に、テクスチャデータおよびオブジェクトデータの D M A 転送プロセスの終了を S Y N C コマンドで確認し、それに応じて K I C K コマンドを実行するようにすれば、グラフィック処理を効率的に実行することができる。また、グラフィック演算ユニット 4 0 におけるグラフィック処理の終了を S Y N C コマンドで取得し、それに応じて D M A コマンドを実行するようにすれば、グラフィック演算ユニット 4 0 のアイドル時間を削減することができる。

## 【 0 0 4 9 】

このように、S Y N C コマンドによって、D M A プロセスまたは K I C K プロセスの終了と、サブプロセッサにおける処理の終了を同期させることが可能になる。これによって、プロセス依存関係がある多数のコマンドを含む場合でも、グラフィックプロセッサ 1 0 0 の動作を適切に制御し、ひいては、グラフィックプロセッサ 1 0 0 とメインプロセッサ 2 0 0 の動作を協調させることができる。S Y N C コマンドのより具体的な使用方法については、図 1 1 ~ 図 1 7 を参照する実施例において説明する。

## 【 0 0 5 0 】

なお、S Y N C コマンドは、D M A コマンド、K I C K コマンドおよびサブプロセッサひとつずつの終了を確認することもできるし、2 個の S Y N C コマンドを連続させて記述することで、多数のプロセスの終了を同期させることもできる。

## 【 0 0 5 1 】

図示しない W A I T コマンドは、S Y N C コマンドと異なり、グラフィックプロセッサ 1 0 0 において実行中である全てのコマンドプロセスが終了するまで、以降のコマンドの実行を保留する。タイプには、W A I T コマンドであることを示す「 5 」が指定される。

## 【 0 0 5 2 】

図 7 は、N E X T コマンドのデータフォーマット 1 7 0 を示す。N E X T コマンドは、ひとかたまりのコマンドリストの終了を示すとともに、次に実行すべきコマンドリストのアドレスを示すコマンドである。タイプ 1 7 1 には、N E X T コマンドであることを示す「 6 」が指定される。次アドレス 1 7 4 には、次に実行すべきコマンドリストの先頭のアドレスが指定される。

## 【 0 0 5 3 】

10

20

30

40

50

本実施の形態では、複数のサブプロセッサ 30 が並列的に描画データの演算を実行し、そのためのグラフィック処理に必要な非グラフィック処理命令を含むコマンド列を各サブプロセッサ 30 が独立して作成する。コマンドリストは、作成された順にメインメモリ 50 に格納される。したがって、制御ユニット 20 では、それら複数のコマンドリストの実行順序を知ることができない。そこで、NEXT コマンドにより次に実行すべきコマンドリストのアドレスを指定することによって、サブプロセッサ 30 における処理の前後に関係なく、またメインメモリ 50 内のロケーションに関係なく、コマンドリストを実行することができる。

【 0 0 5 4 】

図示しない END コマンドは、コマンドリストの終了をセットする命令であり、実行部 26 が END コマンドを実行すると、コマンドリストの開始アドレスに移動する。タイプ番号は「7」である。

【 0 0 5 5 】

図 8 は、NEXT コマンドの使用方法を説明する図である。図示するように、連続して実行実施されるべきコマンドリスト 202、204、206 が存在するとする。これらのコマンドリストは、その実行順序に関係なくメインメモリ 50 に転送される。コマンドリスト 202、204 の末尾には NEXT コマンドが埋め込まれており、この NEXT コマンドには次に実行すべきコマンドリストの先頭アドレスが書き込まれている。また、コマンドリスト 206 の末尾には END コマンドが埋め込まれている。

【 0 0 5 6 】

画像処理側 DMA コントローラ 28 は、予め定められている開始アドレスから始まっているコマンドリスト 202 を第 1 のリストとしてメインメモリ 50 から取り出す。そして、コマンドリスト 202 の NEXT コマンドによって、画像処理側 DMA コントローラ 28 は次に実行すべきコマンドリスト 204 をメインメモリ 50 から取り出すことができる。同様に、コマンドリスト 204 の NEXT コマンドによって、その次に実行すべきコマンドリスト 206 を取り出すことができる。

【 0 0 5 7 】

管理プロセッサ 32 は、各サブプロセッサ 30 で作成されたコマンドリストを解析して、必要な NEXT コマンドを各コマンドリストに埋め込むように構成されている。または、サブプロセッサ 30 が互いにコマンドリストを交換し合うことにより、必要な NEXT コマンドを埋め込むように構成してもよい。つまり、あるサブプロセッサ A がメモリ上のあるアドレスを取得するとき、他のサブプロセッサによるメインメモリ 50 へのアクセスをロックする。そして、サブプロセッサ A がコマンドリストをメインメモリ 50 へ保存した後、他のプロセッサによるメインメモリへのアクセスロックを解除する。続いて、サブプロセッサ B がメインメモリの次アドレスを取得し、サブプロセッサ A の作成したコマンドリストにこのアドレスを含む NEXT コマンドを追加する。この処理を繰り返すことによって、必要な NEXT コマンドを各コマンドリストに埋め込むことができる。

【 0 0 5 8 】

図示しない PERF コマンドは、コマンドプロセスの実行サイクル数を計測するための命令である。タイプ番号は「9」が指定される。制御ユニット 20 の実行部 26 が PERF コマンドを実行すると、非グラフィック処理命令にしたがったグラフィック処理を実行したときグラフィックプロセッサにおいて要した実行サイクル数が計測される。この実行サイクル数は、例えば、制御ユニット 20 のクロックに基づいてカウントされる。この計時命令は、グラフィック処理の効率などのデータをユーザに提供するとき主に使用される。

【 0 0 5 9 】

以上述べたように、各非グラフィック処理系命令は、グラフィック処理や DMA コントローラの開始、終了に関係するものにすぎず、グラフィック演算ユニット 40 におけるグラフィック処理を直接操作している命令、すなわち処理内容を規定している命令ではないことに注意されたい。

10

20

30

40

50

## 【 0 0 6 0 】

図 9 は、情報処理装置 1 0 0 0 において D M A 転送されるデータを示す。これらのデータは、制御ユニット 2 0 により利用されるデータ 2 1 0 と、グラフィック演算ユニット 4 0 により利用されるデータ 2 2 0 とに分けることができる。

## 【 0 0 6 1 】

コンフィグデータ 2 1 2 には、シェーダプログラムなどのアドレスが指定されている。コマンドリスト 2 1 4 は、上述した非グラフィック処理系命令からなる。制御側 D M A コントローラ 3 4 は、コンフィグデータ 2 1 2、コマンドリスト 2 1 4 および頂点データ 2 2 2 の転送を制御する。コンフィグデータ 2 1 2 とコマンドリスト 2 1 4 は、グラフィックプロセッサ 1 0 0 の制御ユニット 2 0 のメモリ領域に転送される。頂点データ 2 2 2 は、グラフィックメモリ 1 0 に転送される。制御側 D M A コントローラ 3 4 によって D M A 転送される場合、転送先のメモリ確保やバッファ管理は、メインプロセッサ側で処理される。

10

画像処理側 D M A コントローラ 2 8 は、テクスチャデータ 2 2 6、パラメータデータ 2 2 4 およびシェーダプログラム 2 2 8 の転送を制御する。テクスチャデータ 2 2 6 は、グラフィックメモリ 1 0 内のバッファ 1 2 に転送される。パラメータデータ 2 2 4 とシェーダプログラム 2 2 8 は、グラフィック演算ユニット 4 0 のメモリ領域に転送される。

## 【 0 0 6 2 】

次に、非グラフィック処理系命令にしたがった情報処理装置 1 0 0 0 の動作を説明する。

20

図 1 0 は、あるオブジェクトをフレームバッファに描画するときのメインプロセッサ 2 0 0 とグラフィックプロセッサ 1 0 0 における処理とデータの流れを示すフローチャートである。以下、図 1 を参照しつつ、各処理を説明する。

## 【 0 0 6 3 】

まず、メインプロセッサ 2 0 0 は、オブジェクトの描画に必要なジオメトリデータなどのデータの作成、ロード、解析を実行する ( S 1 0 )。続いてメインプロセッサ 2 0 0 は、オブジェクトのアドレス解決を実行し、転送先であるメインメモリ 5 0 におけるデータレイアウトを決定し ( S 1 2 )、メインメモリ 5 0 に格納する。そして、メインプロセッサ 2 0 0 は、オブジェクトの描画に必要なコマンドリストとコンフィグデータを作成し ( S 1 4 )、メインメモリ 5 0 に格納する。メインプロセッサ 2 0 0 の制御側 D M A コントローラ 3 4 は、作成されたコマンドリストとコンフィグデータとを、グラフィックプロセッサ 1 0 0 の制御ユニット 2 0 に D M A 転送する ( S 1 6 )。これが終了すると、メインプロセッサ 2 0 0 は、制御ユニット 2 0 に対して、コマンドリストの実行を開始するように通知する ( S 1 8 )。

30

## 【 0 0 6 4 】

グラフィックプロセッサ 1 0 0 の制御ユニット 2 0 がメインプロセッサ 2 0 0 からコマンドリスト実行開始の通知を受け取ると、制御ユニット 2 0 内の命令解析部 2 4 がコマンドリストの解析を開始する ( S 2 2 )。実行部 2 6 がコマンドリスト内の D M A コマンドを実行すると、画像処理側 D M A コントローラ 2 8 は、D M A コマンドのタグアドレスを参照して、オブジェクトの描画に必要なテクスチャデータなどのデータをメインメモリ 5 0 からグラフィックメモリ 1 0 内のバッファ 1 2 に転送する ( S 2 4 )。実行部 2 6 がコマンドリスト内の S Y N C コマンドを実行すると、グラフィック演算ユニット 4 0 は、S 2 4 にしたがって実行中のテクスチャデータの転送が終了するまで以降のコマンドリストの実行を保留する ( S 2 6 )。

40

## 【 0 0 6 5 】

さて、この間にメインプロセッサ 2 0 0 では、オブジェクトのジオメトリ処理を実行して頂点データおよびパラメータデータを算出し ( S 2 0 )、メインメモリ 5 0 に格納しておく。メインプロセッサ 2 0 0 の制御側 D M A コントローラ 3 4 は、頂点データをグラフィックメモリ 1 0 に転送し ( S 2 8 )、ジオメトリ処理が終了したことを示すフラグを内部レジスタ 5 4 にセットする ( S 3 0 )。制御ユニット 2 0 内の実行部 2 6 は、コマンド

50

リスト内の SYNC コマンドにしたがって、頂点データの転送が終了するまで以降のコマンドリストの実行を保留する (S 3 2)。ここまでの処理によって、オブジェクトの描画に必要な各種データが、グラフィックメモリ 10 上に揃ったことになる。

【 0 0 6 6 】

この状態において、実行部 2 6 が K I C K コマンドを実行すると、制御ユニット 2 0 はコンフィグデータをグラフィック演算ユニット 4 0 に転送し、グラフィック演算ユニット 4 0 の処理が起動され、バッファ 1 2 を読み出す (S 3 4)。グラフィック演算ユニット 4 0 は、コンフィグデータに基づいてプログラムを読み出す。グラフィック演算ユニット 4 0 における処理が終了し、フレームバッファへのオブジェクトの描画が完了すると、グラフィック演算ユニット 4 0 は内部レジスタ 5 4 に K I C K プロセスが終了したことを示すフラグをセットする。グラフィック演算ユニット 4 0 は、SYNC コマンドによりオブジェクトの描画終了を確認すると (S 3 6)、NOTIFY コマンドにより、サブプロセッサ 3 0 に対してオブジェクトの描画が終了したことを通知する (S 3 8)。この通知により、メインプロセッサ 2 0 0 は、次のオブジェクトの準備を開始することができる。

【 0 0 6 7 】

情報処理装置 1 0 0 0 においては、サブプロセッサ 3 0 におけるジオメトリ処理と、オブジェクトデータおよびテクスチャデータの DMA 転送と、グラフィック演算ユニット 4 0 におけるグラフィック処理の全ての同期をとる必要があるため、制御側 DMA コントローラ 3 4、画像処理側 DMA コントローラ 2 8 両方の処理手順、処理速度を考慮したコマンドリストを作成する必要がある。以下では、コマンドリストの例を挙げて、これにした

がったメインプロセッサ 2 0 0 とグラフィックプロセッサ 1 0 0 の協調動作を説明する。  
 なお、以下の説明においては、画像処理側 DMA コントローラ 2 8 によるテクスチャデータの DMA 転送は同時に 2 つ、グラフィック演算ユニット 4 0 におけるオブジェクトの描画処理はひとつのみ実行可能であるものとする。また、グラフィックメモリ 1 0 には、テクスチャデータを格納するコンテキストバッファ 1 2 として、バッファ A、バッファ B が備えられているとする。

【 0 0 6 8 】

図 1 1 は、コマンドリストの一例であるコマンドリスト 7 0 0 を示す。なお、コマンドの解釈はリスト通りに行われるが、実際の処理は必ずしもコマンドの順序で起きるわけではなく、DMA 転送されるデータ量やグラフィック処理に要する時間により前後する。

【 0 0 6 9 】

図 1 1 において、「タイプ」はコマンド名を表し、「属性」は SYNC コマンドで同期させるべき対象を略号で示す。「D」は DMA プロセス、「F」はサブプロセッサのジオメトリ処理、「K」は K I C K プロセスを表している。「ID」は、上述した DMA I D または K I C K I D であり、この ID によって、複数ある DMA プロセスまたは K I C K プロセスのうち待機すべきプロセスを特定することができる。また、図中の「Tex」はテクスチャを、「Obj」はオブジェクトを、「SPU」はサブプロセッサをそれぞれ表す。符号 A、B は 2 つのバッファ 1 2 を表す。例えば「Obj A」は、バッファ A に DMA 転送されるオブジェクトを表し、「Tex A」は、オブジェクト A のグラフィック処理に必要であり、バッファ A に DMA 転送されるテクスチャデータを表す。なお、実際のグラフィック処理には、テクスチャデータ以外にカラーデータや値のデータなども必要となるが、ここではテクスチャデータを代表として示している。「SPU A」は、バッファ A に DMA 転送されるオブジェクト A を処理するサブプロセッサを表す。

【 0 0 7 0 】

実行部 2 6 は、コマンドリスト 7 0 0 を開始アドレスから順次実行する。まず、ID「1」を付与された DMA コマンド 7 0 1 と、ID「2」を付与された DMA コマンド 7 0 2 を実行する。これにより、画像処理側 DMA コントローラ 2 8 は、メインメモリ 5 0 からテクスチャ A (Tex A) をバッファ A に転送開始し、テクスチャ B (Tex B) をバッファ B に転送開始する。

このとき、制御側 DMA コントローラ 3 4 は、バッファ A へのテクスチャ A の DMA 転

10

20

30

40

50

送に並行して、サブプロセッサAからグラフィックメモリ10へのオブジェクトA(O b j A)の転送を実行する。また、制御側DMAコントローラ34は、バッファBへのテクスチャBのDMA転送に並行して、サブプロセッサBからグラフィックメモリ10へのオブジェクトB(O b j B)の転送を実行する。なお、制御側DMAコントローラ34は、実行部26によるコマンドの実行とは無関係に動作する。

**【0071】**

グラフィック演算ユニット40は、SYNCコマンド703にしたがって、ID「1」を付与されたDMAコマンド701に対応するバッファAへのテクスチャAの転送の終了と、制御側DMAコントローラ34によるサブプロセッサA(S P U A)からグラフィックメモリ10へのオブジェクトAの転送の終了を確認する。ID「1」を付与されたK I C Kコマンド704にしたがって、グラフィック演算ユニット40は、バッファAにリードアクセスしてオブジェクトAに対するグラフィック処理を開始する。

10

**【0072】**

続いて、グラフィック演算ユニット40は、SYNCコマンド705にしたがって、ID「2」を付与されたDMAコマンド702に対応するバッファBへのテクスチャBの転送の終了と、制御側DMAコントローラ34によるサブプロセッサB(S P U B)からグラフィックメモリ10へのオブジェクトBの転送の終了を確認する。ID「2」を付与されたK I C Kコマンド706にしたがって、グラフィック演算ユニット40は、バッファBにリードアクセスしてオブジェクトBに対するグラフィック処理を開始する。

20

**【0073】**

続いてグラフィック演算ユニット40は、SYNCコマンド707により、ID「1」を付与されたK I C Kコマンド704に対応するオブジェクトAのグラフィック処理の終了を確認する。グラフィック処理済のオブジェクトAはグラフィックメモリ10のフレームバッファに書き出される。NOTIFYコマンド708により、グラフィック演算ユニット40はサブプロセッサA(S P U A)に対してオブジェクトAのグラフィック処理が完了したことを通知する。これによってサブプロセッサAは、バッファAが空いていることを知る。

**【0074】**

画像処理側DMAコントローラ28は、ID「3」を付与されたDMAコマンド709にしたがって、空きができたバッファAに対して、新たなオブジェクトAのグラフィック処理に必要なテクスチャAの転送を開始する。

30

制御側DMAコントローラ34は、バッファAへのテクスチャAのDMA転送に並行して、サブプロセッサAからグラフィックメモリ10へのオブジェクトAのDMA転送を実行する。

**【0075】**

グラフィック演算ユニット40は、SYNCコマンド710により、ID「3」を付与されたDMAコマンド709に対応するバッファAへのテクスチャAの転送の終了と、サブプロセッサAからグラフィックメモリ10への新たなオブジェクトAの転送の終了を確認する。ID「3」を付与されたK I C Kコマンド711にしたがって、グラフィック演算ユニット40は、バッファAにリードアクセスしてオブジェクトAに対するグラフィック処理を実行する。

40

**【0076】**

グラフィック演算ユニット40は、SYNCコマンド712により、ID「2」を付与されたK I C Kコマンド706に対応するオブジェクトBのグラフィック処理の終了を確認する。グラフィック処理済のオブジェクトBはグラフィックメモリ10のフレームバッファに書き出される。NOTIFYコマンド713により、グラフィック演算ユニット40はサブプロセッサBに対してオブジェクトBのグラフィック処理が完了したことを通知する。

**【0077】**

画像処理側DMAコントローラ28は、ID「4」を付与されたDMAコマンド714

50

にしたがって、空きができたバッファBに対して、新たなオブジェクトBのグラフィック処理に必要なテクスチャBの転送を開始する。

制御側DMAコントローラ34は、バッファBへのテクスチャBのDMA転送に並行して、サブプロセッサBからグラフィックメモリ10へのオブジェクトBのDMA転送を実行する。

【0078】

グラフィック演算ユニット40は、SYNCコマンド715にしたがって、ID「4」を付与されたDMAコマンド714に対応するバッファBへのテクスチャBの転送の終了と、サブプロセッサBからバッファBへの新たなオブジェクトBの転送の終了を確認する。ID「4」を付与されたKICKコマンド716にしたがって、グラフィック演算ユニット40は、バッファBにリードアクセスしてオブジェクトBに対するグラフィック処理を開始する。

10

【0079】

実行部26がWAITコマンド717を実行すると、グラフィック演算ユニット40は、オブジェクトAとオブジェクトBの2回目のグラフィック処理の終了を待機し、NOTIFYコマンド718により、グラフィック処理の終了を全サブプロセッサに対して通知する。

【0080】

図12は、図11のコマンドリスト700にしたがったテクスチャデータの転送を模式的に示す。DMAコマンドにしたがって、テクスチャAがバッファAに転送されるのと並行して、サブプロセッサからはオブジェクトAがグラフィックメモリに転送される。テクスチャAとオブジェクトAの転送が両方とも終了すると、グラフィック演算ユニット40はオブジェクトAのグラフィック処理を実行する。また、DMAコマンドにしたがって、テクスチャBがバッファBに転送されるのと並行して、サブプロセッサからはオブジェクトBがグラフィックメモリに転送される。テクスチャBとオブジェクトBの転送が両方とも終了すると、グラフィック演算ユニット40はオブジェクトBのグラフィック処理を実行する。以降、上記の処理が繰り返される。

20

【0081】

図13は、図11のコマンドリスト700にしたがったときのバッファA、バッファBの状態を示すチャートである。図中の符号は、図11のコマンドリスト内のコマンドに与えた符号と対応する。オブジェクトAの処理に必要なテクスチャAがメインメモリ50からバッファAに転送され(701)、またオブジェクトBの処理に必要なテクスチャBがメインメモリ50からバッファBに転送される(702)。SYNCコマンドによりオブジェクトAとテクスチャAの転送の終了を確認すると、KICKコマンドによりそれらがグラフィック演算ユニット40によりバッファAから読み出され、グラフィック処理が開始される(704)。以降、DMAコマンドによるバッファA、バッファBに対するDMA転送と、KICKコマンドによるバッファA、バッファBからの読み出しが順番に繰り返される。

30

なお、図13においては、簡単のために全てのステージにおける処理が同時間で終了するように描かれているが、実際の処理においてはDMA転送されるデータ量やグラフィック処理に要する時間により前後する。

40

【0082】

情報処理装置1000においては、画像処理側DMAコントローラはテクスチャデータのDMA転送を同時に2つ実行できるが、バッファが2つであると、実装においてはグラフィック演算ユニット40およびメインプロセッサ200のアイドル時間が長くなってしまい、情報処理装置1000の処理効率の向上を妨げるおそれがある。そこで、グラフィックメモリ10にバッファ12を3つ以上設け、グラフィック演算ユニット40およびメインプロセッサ200のアイドル時間を短縮することで、情報処理装置全体の処理効率を高める方法について説明する。

【0083】

50

図14は、グラフィックメモリ10がバッファA、バッファB、バッファCの3つを備えるとした場合のコマンドリスト800の一例を示す。図15は、図14のコマンドリストにしたがったときのバッファA、バッファB、バッファCの状態を示すチャートである。図14、図15中の「Tex」「Obj」「SPU」は、図11の場合と同様にそれぞれ「テクスチャ」「オブジェクト」「サブプロセッサ」を表す。また、「IDLE」は、画像処理側DMAコントローラ28による各バッファへのDMA転送が実行されていない状態を表す。

**【0084】**

以下、図15のチャートを参照しつつ、図14のコマンドリスト800を説明する。

実行部26は、コマンドリスト800を開始アドレスから順次実行する。まず、ID「1」を付与されたDMAコマンド801と、ID「2」を付与されたDMAコマンド802を実行する。これにより、画像処理側DMAコントローラ28は、メインメモリ50からテクスチャA(Tex A)をバッファAに転送開始し、テクスチャB(Tex B)をバッファBに転送開始する。

10

このとき、制御側DMAコントローラ34は、バッファAへのテクスチャAのDMA転送に並行して、サブプロセッサAからグラフィックメモリ10へのオブジェクトA(Obj A)の転送を実行する。また、制御側DMAコントローラ34は、バッファBへのテクスチャBのDMA転送に並行して、サブプロセッサBからグラフィックメモリ10へのオブジェクトB(Obj B)の転送を実行する。

**【0085】**

20

グラフィック演算ユニット40は、SYNCコマンド803にしたがって、ID「1」を付与されたDMAコマンド801に対応するバッファAへのテクスチャAの転送の終了と、制御側DMAコントローラ34によるサブプロセッサA(SPU A)からグラフィックメモリ10へのオブジェクトAの転送の終了を確認する。ID「1」を付与されたKICKコマンド804にしたがって、グラフィック演算ユニット40は、バッファAにリードアクセスしてオブジェクトAに対するグラフィック処理を開始する。

**【0086】**

画像処理側DMAコントローラ28は、ID「3」を付与されたDMAコマンド805により、バッファCへのテクスチャC(Tex C)の転送を開始する。このとき、制御側DMAコントローラ34は、サブプロセッサCからグラフィックメモリ10へのオブジェクトC(Obj C)の転送を実行する。

30

**【0087】**

グラフィック演算ユニット40は、SYNCコマンド806により、ID「2」を付与されたDMAコマンド802に対応するバッファBへのテクスチャBの転送の終了と、制御側DMAコントローラ34によるサブプロセッサB(SPU B)からグラフィックメモリ10へのオブジェクトBの転送の終了を確認する。ID「2」を付与されたKICKコマンド807にしたがって、グラフィック演算ユニット40は、バッファBにリードアクセスしてオブジェクトBに対するグラフィック処理を開始する。

**【0088】**

グラフィック演算ユニット40は、SYNCコマンド808により、ID「1」を付与されたKICKコマンド804に対応するオブジェクトAのグラフィック処理の終了を確認する。グラフィック処理済のオブジェクトAは、グラフィックメモリ10のフレームバッファに書き出される。NOTIFYコマンド809により、グラフィック演算ユニット40はサブプロセッサA(SPU A)に対してオブジェクトAのグラフィック処理が完了したことを通知する。これによってサブプロセッサAは、バッファAが空いていることを知る。

40

**【0089】**

画像処理側DMAコントローラ28は、ID「4」を付与されたDMAコマンド810にしたがって、空きができたバッファAに対して、新たなオブジェクトAのグラフィック処理に必要なテクスチャAのDMA転送を開始する。

50

制御側DMAコントローラ34は、バッファAへのテクスチャAのDMA転送に並行して、サブプロセッサAからグラフィックメモリ10へのオブジェクトAのDMA転送を実行する。

【0090】

グラフィック演算ユニット40は、SYNCコマンド811により、ID「3」を付与されたDMAコマンド805に対応するバッファCへのテクスチャCの転送の終了と、サブプロセッサC(SPU C)からグラフィックメモリ10へのオブジェクトCの転送の終了を確認する。ID「3」を付与されたKICKコマンド812にしたがって、グラフィック演算ユニット40は、バッファCにリードアクセスしてオブジェクトCに対するグラフィック処理を実行する。

10

【0091】

以下、コマンド813~824にしたがって、上述と同様の手順が繰り返される。

実行部26がWAITコマンド825を実行すると、グラフィック演算ユニット40は、オブジェクトA~Cの2回目のグラフィック処理の終了を待機し、NOTIFYコマンド826により、グラフィック処理が終了したことを全サブプロセッサに対して通知する。

【0092】

コマンドリストは先頭から順に実行されていくので、コマンドを適切に配置することによって、グラフィック演算ユニット40のアイドル時間を削減することができる。図15を参照して説明すると、バッファAへのテクスチャAとオブジェクトAの転送終了をSYNCコマンド803により確認したのち、KICKコマンド804によりグラフィック演算ユニット40はバッファAにアクセスしてグラフィック処理を開始する。この間に、バッファBへのテクスチャBとオブジェクトBの転送と、バッファCへのテクスチャCとオブジェクトCの転送が行われる。

20

【0093】

その後も、バッファBへのテクスチャBとオブジェクトBの転送終了を確認した後、オブジェクトBのグラフィック処理を開始し、バッファCへのテクスチャCとオブジェクトCの転送終了を確認した後、オブジェクトCのグラフィック処理を開始する、といったように、グラフィック演算ユニット40を、オブジェクトA、オブジェクトB、オブジェクトCの順で演算させる。このように、SYNCコマンドとKICKコマンドを適切に配置したコマンドリストを作成することによって、グラフィック演算ユニット40のアイドル時間を短縮することができる。

30

【0094】

また、サブプロセッサは、オブジェクトを転送した時点から別のオブジェクトの演算を実行し、グラフィック処理の終了通知を受けると直ちに、そのオブジェクトを転送できる。したがって、サブプロセッサ側のアイドル時間も短縮することができる。

【0095】

図16は、グラフィックメモリ10がバッファA、バッファB、バッファC、バッファDの4つを備えるときのコマンドリスト900の一例を示す。図17は、図16のコマンドリストにしたがったときのバッファA~バッファDの状態を示すチャートである。

40

【0096】

以下、図17のチャートを参照しつつ、図16のコマンドリスト900を説明する。

実行部26は、コマンドリスト900を開始アドレスから順次実行する。まず、ID「1」を付与されたDMAコマンド901と、ID「2」を付与されたDMAコマンド902を実行する。これにより、画像処理側DMAコントローラ28は、メインメモリ50からテクスチャA(Tex A)をバッファAに転送開始し、テクスチャB(Tex B)をバッファBに転送開始する。

このとき、制御側DMAコントローラ34は、バッファAへのテクスチャAのDMA転送に並行して、サブプロセッサAからグラフィックメモリ10へのオブジェクトA(Obj A)の転送を実行する。また、制御側DMAコントローラ34は、バッファBへのテ

50



クスチャBのDMA転送に並行して、サブプロセッサBからグラフィックメモリ10へのオブジェクトB(Obj B)の転送を実行する。

【0097】

グラフィック演算ユニット40は、SYNCコマンド903にしたがって、ID「1」を付与されたDMAコマンド901に対応するバッファAへのテクスチャAの転送の終了と、制御側DMAコントローラ34によるサブプロセッサA(SPU A)からグラフィックメモリ10へのオブジェクトAの転送の終了を確認する。ID「1」を付与されたKICKコマンド904にしたがって、グラフィック演算ユニット40は、バッファAにリードアクセスしてオブジェクトAに対するグラフィック処理を開始する。

【0098】

画像処理側DMAコントローラ28は、ID「3」を付与されたDMAコマンド805により、バッファCへのテクスチャC(Tex C)の転送を開始する。このとき、制御側DMAコントローラ34は、サブプロセッサCからグラフィックメモリ10へのオブジェクトC(Obj C)の転送を実行する。

【0099】

グラフィック演算ユニット40は、SYNCコマンド906により、ID「2」を付与されたDMAコマンド902に対応するバッファBへのテクスチャBの転送の終了と、制御側DMAコントローラ34によるサブプロセッサB(SPU B)からグラフィックメモリ10へのオブジェクトBの転送の終了を確認する。ID「2」を付与されたKICKコマンド907にしたがって、グラフィック演算ユニット40は、バッファBにリードアクセスしてオブジェクトBに対するグラフィック処理を開始する。

【0100】

画像処理側DMAコントローラ28は、ID「4」を付与されたDMAコマンド908により、バッファDへのテクスチャD(Tex D)の転送が開始される。このとき、制御側DMAコントローラ34は、サブプロセッサDからグラフィックメモリ10へのオブジェクトD(Obj D)の転送を実行する。

【0101】

グラフィック演算ユニット40は、SYNCコマンド909にしたがって、ID「3」を付与されたDMAコマンド905に対応するバッファCへのテクスチャCの転送の終了と、制御側DMAコントローラ34によるサブプロセッサC(SPU C)からグラフィックメモリ10へのオブジェクトCの転送の終了を確認する。ID「3」を付与されたKICKコマンド910にしたがって、グラフィック演算ユニット40は、バッファCにリードアクセスしてオブジェクトCに対するグラフィック処理を開始する。

【0102】

グラフィック演算ユニット40は、SYNCコマンド911により、ID「1」を付与されたKICKコマンド904に対応するオブジェクトAのグラフィック処理の終了を確認する。グラフィック処理済のオブジェクトAは、グラフィックメモリ10のフレームバッファに書き出される。NOTIFYコマンド912により、グラフィック演算ユニット40はオブジェクトAのグラフィック処理が完了したことをサブプロセッサA(SPU A)に通知する。これによってサブプロセッサAは、バッファAが空いていることを知る。

【0103】

画像処理側DMAコントローラ28は、ID「5」を付与されたDMAコマンド913にしたがって、空きができたバッファAに対して、新たなオブジェクトAのグラフィック処理に必要なテクスチャAのDMA転送を開始する。このとき、制御側DMAコントローラ34は、バッファAへのテクスチャAのDMA転送に並行して、サブプロセッサAからグラフィックメモリ10へのオブジェクトAのDMA転送を実行する。

【0104】

グラフィック演算ユニット40は、SYNCコマンド914により、ID「4」を付与されたDMAコマンド908に対応するバッファDへのテクスチャDの転送の終了と、サ

10

20

30

40

50

ブプロセッサD ( S P U D ) からグラフィックメモリ10へのオブジェクトDの転送の終了を確認する。ID「4」を付与されたKICKコマンド915にしたがって、グラフィック演算ユニット40は、バッファDにリードアクセスしてオブジェクトDに対するグラフィック処理を実行する。

**【0105】**

以下、コマンド916～932にしたがって、上述と同様の手順が繰り返される。

実行部26がWAITコマンド933を実行すると、グラフィック演算ユニット40は、オブジェクトA～Dの2回目のグラフィック処理の終了を待機し、NOTIFYコマンド934により、グラフィック処理が終了したことを全サブプロセッサに対して通知する。

10

**【0106】**

このように、SYNCコマンドとKICKコマンドを適切に配置することによって、グラフィック演算ユニット40を、オブジェクトA、オブジェクトB、オブジェクトC、オブジェクトDの順で演算させている。これによって、グラフィック演算ユニット40のアイドル時間を短縮することができる。

**【0107】**

このように、本実施の形態では、非グラフィック処理系命令からなるコマンドリストを組むことによって、グラフィックプロセッサにおけるグラフィック処理の順序を制御することができる。

**【0108】**

グラフィックメモリにバッファを4つ備えることの有利な点を、図15と図17とを比較して説明する。

20

図15のバッファAを見ると、KICKコマンド(例えばコマンド804)による描画が終了してから次のKICKコマンド(コマンド817)までの間には、バッファB、Cに対応するKICKコマンド(コマンド807、812)分しか時間がない。これに対し、図17のバッファAの場合は、KICKコマンド(例えばコマンド904)による描画が終了してから次のKICKコマンド(コマンド920)までの間には、バッファB、C、Dに対応するKICKコマンド(コマンド907、910、915)分の時間を持つことができる。

**【0109】**

言い換えると、図17のバッファAにおいては、KICKコマンドによる描画開始(例えばコマンド904)から次のKICKコマンド(コマンド920)までの間に、図15のバッファAと比べて「IDLE」状態が余分に含まれている。

30

この時間的余裕は次データの準備期間として働き、グラフィック演算ユニット40、サブプロセッサ30といった演算器をストールさせることなくビジー状態に保てるという利点がある。

また、NOTIFYコマンドによるKICKの終了通知がメインプロセッサ200に到達するまでのレイテンシを、「IDLE」の時間に隠すことができるという利点もある。

**【0110】**

以上説明したように、本実施の形態によれば、非グラフィック処理系命令を通して、メインプロセッサ200とグラフィックプロセッサ100との協調動作をユーザが制御することが可能になる。

40

**【0111】**

本実施の形態における非グラフィック処理系命令は、制御ユニット20に対する簡便な少数のコマンドを組み合わせることによって、グラフィック演算ユニットの動作を制御することができる。この非グラフィック処理系命令は、グラフィック演算ユニットにおける具体的なグラフィック処理は指定しないので、シェーダプログラム等のグラフィック処理を実行するプログラムの影響を受けない。

**【0112】**

非グラフィック処理系命令の代わりに、アプリケーションソフトウェアの開発者のため

50

にグラフィック演算ユニットの処理に対する直接的で詳細なコマンドを準備しておくこともできるが、こうすると、グラフィック演算ユニットのハードウェア構成を変更した場合にコマンドのサポートをする負担が大きくなる。本実施形態による非グラフィック処理系命令は少数のコマンドしか準備されておらず、またグラフィック演算ユニットの基本的な動作に関するものに限られているので、上述のような負担を軽減することができる。

#### 【 0 1 1 3 】

また、非グラフィック処理系命令により、バッファの割り当てをユーザが指定することができる。このため、メインプロセッサがバッファの割り当てを実行する必要がなくなるので、メインプロセッサの処理負荷を軽減することができる。

#### 【 0 1 1 4 】

情報処理装置 1 0 0 0 において、メインプロセッサ 2 0 0 にはサブプロセッサ 3 0 が複数設けられているため、オブジェクトのジオメトリ処理を並列的に実行できるのに対し、グラフィックプロセッサ 1 0 0 におけるオブジェクトのグラフィック処理はひとつずつしかできない。そこで、本実施の形態にしたがって、K I C K コマンドにより起動されるグラフィック演算ユニットにおけるグラフィック処理の演算時間と、D M A コマンドにより開始されるデータの転送時間とが同程度になるようなコマンドリストを作成し、グラフィックプロセッサ 1 0 0 にとって好適なタイミングで描画データ転送を実行するようすれば、メインプロセッサとグラフィックプロセッサのロードバランスが改善されるので、情報処理装置 1 0 0 0 の処理効率が向上する。

#### 【 0 1 1 5 】

言い換えると、情報処理装置 1 0 0 0 においてはメインプロセッサ側の処理速度の方が速いので、グラフィック演算ユニットにおけるグラフィック処理の終了を待機する期間をできるだけ短縮することで、情報処理装置における処理を高速化することができる。

例えば、サイズの大きいフレームを描画する場合を考える。この場合、フレーム全体をグラフィック処理させようとする、演算量が非常に大きいため、サブプロセッサはグラフィック演算ユニットの処理の終了を待機する期間が長くなる。そこで、フレームを 4 分割して、各分割フレームを描画するのに必要なデータを D M A 転送させる時間と、各フレームのオブジェクトをジオメトリ処理する時間と、各分割フレームをグラフィック処理する時間のバランスをとるようにコマンドリストを作成すれば、画像処理時間を短縮させることができる。このように、コマンドリストの工夫次第で、情報処理の効率をアップさせることが可能となる。

#### 【 0 1 1 6 】

以上、本発明を実施の形態をもとに説明した。上記実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能で、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。以下、そのような変形例について述べる。

#### 【 0 1 1 7 】

実施の形態では、グラフィック演算ユニット 4 0 において単一のオブジェクトのグラフィック処理を行う場合について説明した。しかしながら、グラフィック演算ユニット 4 0 は、複数の異なるオブジェクトのグラフィック処理を並列に実行可能な構成にすることができる。

図 1 8 は、この場合の非グラフィック処理系命令を使用したコマンドリスト 7 5 0 の一例を示す。図 1 8 において、複数のオブジェクトの集合「O b j s e t A」のグラフィック処理に必要なテクスチャの集合を「T e x s e t A」と表記する。オブジェクトが単数である場合と同様に、テクスチャセットを D M A 転送する D M A コマンド、テクスチャセットとそれに対応するオブジェクトセットの転送終了を確認する S Y N C コマンド、転送終了後にオブジェクトセットに対する描画開始を指示する K I C K コマンドを配置する。これによって、処理対象のオブジェクトが複数存在する場合でも、単一のオブジェクトの場合と同じように、グラフィック演算ユニット 4 0 におけるグラフィック処理を制御することができる。

10

20

30

40

50

## 【 0 1 1 8 】

実施の形態に係る情報処理装置 1 0 0 0 においては、グラフィックプロセッサ 1 0 0 とメインプロセッサ 2 0 0 とが単一の L S I チップとして一体に集積化されていてもよく、または別のチップとして構成されていてもよい。実施の形態において示した各ブロックは、集積度についてまで限定するものではなく、いずれの構成要素の組合せが集積化されてもよいし、あるいは別チップとして構成されていてもよい。

## 【 0 1 1 9 】

実施の形態ではメインプロセッサ 2 0 0 をマルチプロセッサとして記述したが、本発明はシングルプロセッサにも適用できる。また、グラフィックプロセッサ 1 0 0 をマルチプロセッサとして構成し、個々のプロセッサに対して本発明を適用し、メインプロセッサ 2 0 0 との協調動作をさせることもできる。

## 【 0 1 2 0 】

実施の形態では、グラフィックプロセッサ 1 0 0 の外部にグラフィックメモリ 1 0 が設けられている構成を示したが、グラフィックメモリはグラフィックプロセッサ 1 0 0 の内部に設けられてもよい。また、グラフィックメモリを単独で設けず、メインメモリ 5 0 の一部の領域をグラフィックメモリとして使用してもよい。

## 【 0 1 2 1 】

N O T I F Y コマンドを使用してオブジェクトのグラフィック処理が終了したことをサブプロセッサに通知する代わりに、内部レジスタ 5 4 に、グラフィック処理が終了したことを示すフラグを立てるようにしてもよい。そして、メインプロセッサ 2 0 0 は、内部レジスタ 5 4 をポーリングするように構成してもよい。こうすれば、N O T I F Y コマンドを使用することなく、オブジェクトのグラフィック処理が終了したことを、メインプロセッサ 2 0 0 が知ることができる。N O T I F Y コマンドではタイムラグが生じ得るが、メインプロセッサ 2 0 0 が、このタイムラグを待たないような場合、または、プロセッサに都合のよいタイミングでグラフィック処理の終了を確認したいときは、内部レジスタを使用する。内部レジスタ 5 4 と N O T I F Y コマンドを併用してもよい。この場合、サブプロセッサ 3 0 はポーリングと N O T I F Y コマンドによる通知のいずれかを状況に応じて使い分けることができる。

## 【 0 1 2 2 】

また、N O T I F Y コマンドによりグラフィック処理の終了を通知する先は、メインプロセッサ 2 0 0 に限られず、グラフィック演算ユニット 4 0 の外部の任意のプロセッサであってよい。

## 【 0 1 2 3 】

実施の形態では、テクスチャデータやパラメータデータなどのグラフィック処理に必要なデータの D M A 転送は画像処理側 D M A コントローラ 2 8 が担当し、それ以外の、コンフィグデータやコマンドリスト、頂点データ、シェーダプログラムの D M A 転送は、制御側 D M A コントローラ 3 4 が担当するものとして説明した。しかしながら、上記は代表的な例にすぎず、これら全てのデータの D M A 転送を制御側 D M A コントローラ 3 4 で担当してもよいし、または画像処理側 D M A コントローラ 2 8 で担当してもよい。テクスチャデータやパラメータデータの転送を制御側 D M A コントローラ 3 4 が担当する場合、実行部 2 6 は、D M A コマンドを実行したとき、制御側 D M A コントローラ 3 4 に D M A 転送の指令を出す。

## 【 0 1 2 4 】

実施の形態では、制御ユニット 2 0 の実行部 2 6 がコマンドリストを上から順に実行していくことを述べたが、命令解析部 2 4 がコマンドリストを解析した際に、コマンドを再配置するようにしてもよい。例えば、命令解析部 2 4 は、適切なタイミングで P E R F コマンドを実行して、D M A コマンドと K I C K コマンドの実行サイクル数を取得する。そして、両コマンドの実行サイクル数が大きく異なる場合には、コマンドリスト内の D M A コマンドや K I C K コマンドの位置を入れ替えることによって、グラフィックプロセッサ 1 0 0 の処理効率を改善することができる。

10

20

30

40

50

## 【 0 1 2 5 】

また、命令解析部 2 4 は、コマンドリストの解析時にコマンドリストに配列ミスを見つけたときに、そのミスを修正するように構成されていてもよい。例えば、S Y N C コマンド内の I D を参照することによって、当該 S Y N C コマンドが同期の対象としている D M A コマンドまたは K I C K コマンドより前に位置しているときは、命令解析部 2 4 が S Y N C コマンドの位置を D M A コマンドまたは K I C K コマンドの後に並べ替えるように構成されていてもよい。また、D M A コマンドまたは K I C K コマンドがコマンドリスト内に記述されている場合に、それぞれに付与されている I D をもとにコマンドリストを検索して、その同期をとるための S Y N C コマンドが当該コマンドリスト内に存在しないときは、命令解析部 2 4 がその S Y N C コマンドをコマンドリストに追加するように構成されていてもよい。

10

## 【図面の簡単な説明】

## 【 0 1 2 6 】

【図 1】本発明の一実施形態に係る情報処理装置の構成を示すブロック図である。

【図 2】コマンドフォーマットの一例を示す図である。

【図 3】D M A コマンドのデータフォーマットを示す図である。

【図 4】K I C K コマンドのデータフォーマットを示す図である。

【図 5】N O T I F Y コマンドのデータフォーマットを示す図である。

【図 6】S Y N C コマンドのデータフォーマットを示す図である。

【図 7】N E X T コマンドのデータフォーマットを示す図である。

20

【図 8】N E X T コマンドを説明する図である。

【図 9】情報処理装置において D M A 転送されるデータを示す図である。

【図 10】オブジェクトを描画する際のメインプロセッサとグラフィックプロセッサの協調動作を示すフローチャートである。

【図 11】グラフィックメモリにバッファを 2 つ備えるときのコマンドリストの一例を示す図である。

【図 12】図 11 のコマンドリストにしたがったテクスチャデータの転送を模式的に示す図である。

【図 13】図 11 のコマンドリストにしたがったときのバッファの状態を示すチャートである。

30

【図 14】グラフィックメモリにバッファを 3 つ備えるときのコマンドリストの一例を示す図である。

【図 15】図 14 のコマンドリストにしたがったときのバッファの状態を示すチャートである。

【図 16】グラフィックメモリにバッファを 4 つ備えるときのコマンドリストの一例を示す図である。

【図 17】図 16 のコマンドリストにしたがったときのバッファの状態を示すチャートである。

【図 18】グラフィック演算ユニットにおいて複数のオブジェクトのグラフィック処理を並列に実行可能であるときのコマンドリストの一例を示す図である。

40

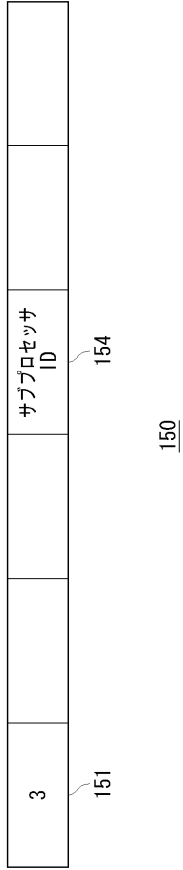
## 【符号の説明】

## 【 0 1 2 7 】

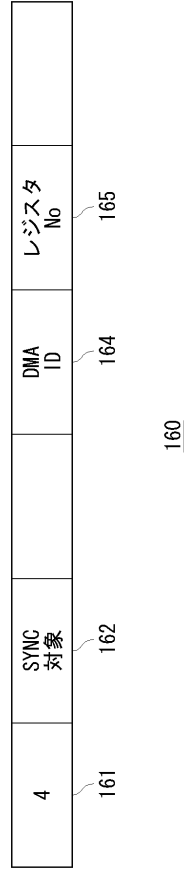
1 0 グラフィックメモリ、 1 2 コンテキストバッファ、 1 8 ディスプレイコントローラ、 2 0 制御ユニット、 2 2 インタフェイス部、 2 4 命令解析部、 2 6 実行部、 2 8 画像処理側 D M A コントローラ、 3 0 サブプロセッサ、 3 2 管理プロセッサ、 3 4 制御側 D M A コントローラ、 4 0 グラフィック演算ユニット、 4 2 ラスタライザ、 4 4 シェーダユニット、 4 6 テクスチャユニット、 5 0 メインメモリ、 5 2 表示装置、 5 4 内部レジスタ、 1 0 0 グラフィックプロセッサ、 2 0 0 メインプロセッサ、 1 0 0 0 情報処理装置。



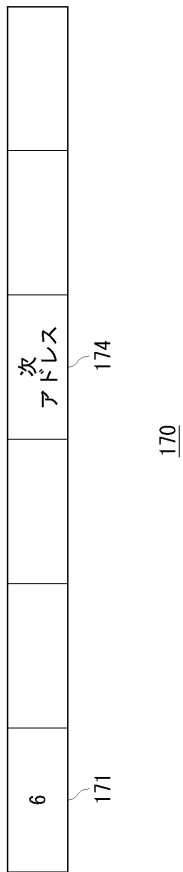
【 図 5 】



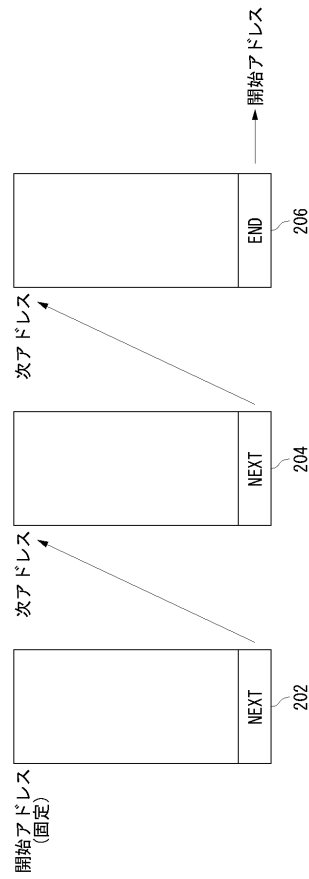
【 図 6 】



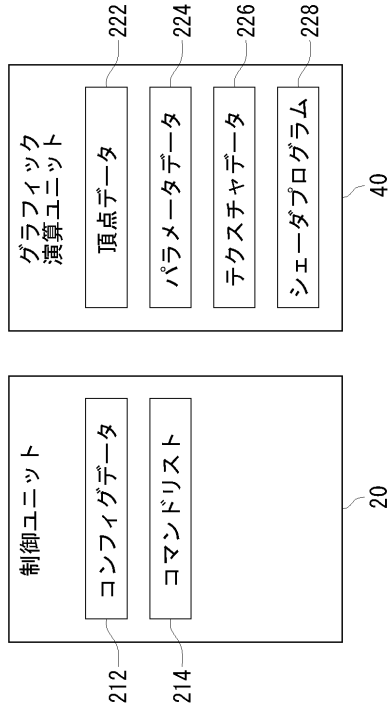
【 図 7 】



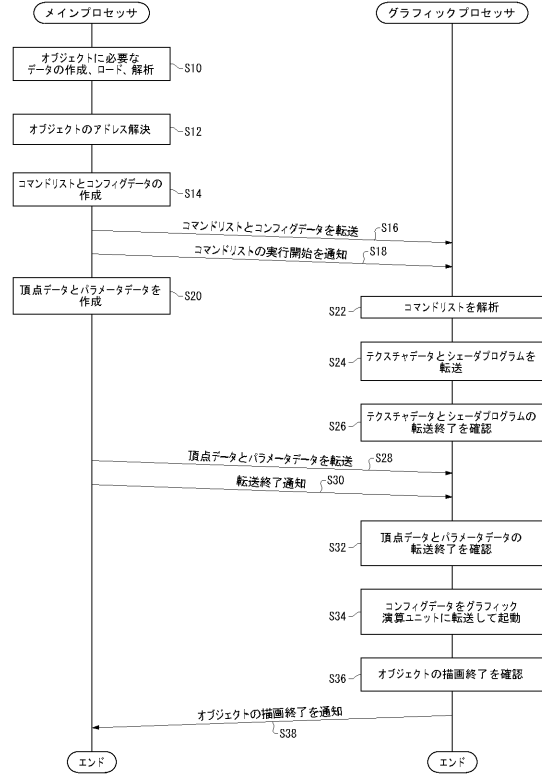
【 図 8 】



【図 9】



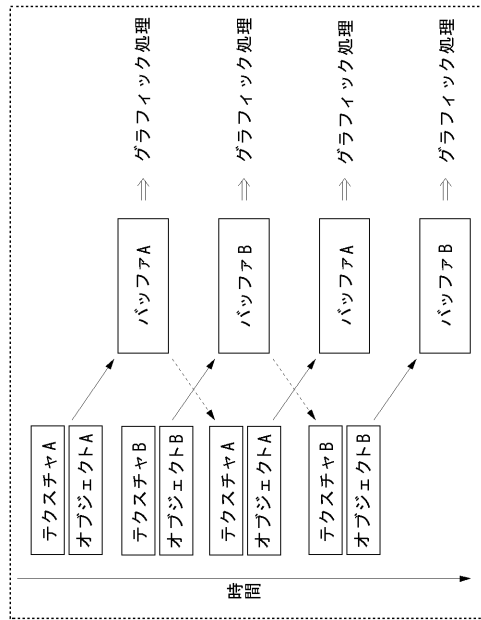
【図 10】



【図 11】

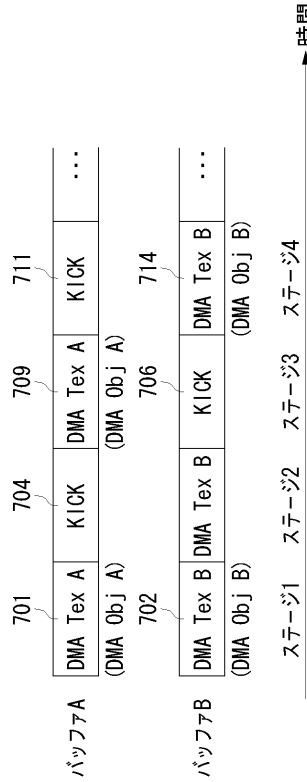
	タイプ	属性	ID	
701	DMA		1	Tex A
702	DMA		2	Tex B
703	SYNC	D F	1	SPU A
704	KICK		1	Obj A
705	SYNC	D F	2	SPU B
706	KICK		2	Obj B
707	SYNC	K	1	Obj A
708	NOTIFY		1	SPU A
709	DMA		3	Tex A
710	SYNC	D F	3	SPU A
711	KICK		3	Obj A
712	SYNC	K	2	Obj B
713	NOTIFY		2	SPU B
714	DMA		4	Tex B
715	SYNC	D F	4	SPU B
716	KICK		4	Obj B
717	WAIT			
718	NOTIFY			SPU all

【図 12】





【図 13】

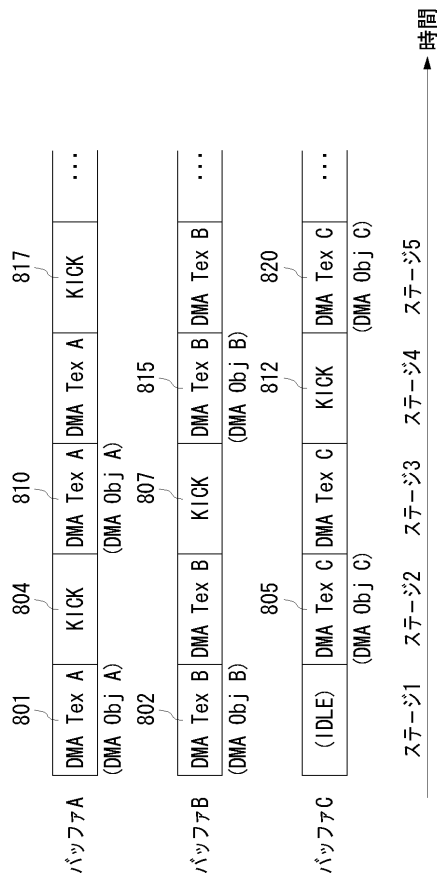


【図 14】

ID	タイプ	属性	ID	
801	DMA		1	Tex A
802	DMA		2	Tex B
803	SYNC	D F	1	SPU A
804	KICK		1	Obj A
805	DMA		3	Tex C
806	SYNC	D F	2	SPU B
807	KICK		2	Obj B
808	SYNC	K	1	Obj A
809	NOTIFY		1	SPU A
810	DMA		4	Tex A
811	SYNC	D F	3	SPU C
812	KICK		3	Obj C
813	SYNC	K	2	Obj B
814	NOTIFY		2	SPU B
815	DMA		5	Tex B
816	SYNC	D F	4	SPU A
817	KICK		4	Obj A
818	SYNC	K	3	Obj C
819	NOTIFY		3	SPU C
820	DMA		6	Tex C
821	SYNC	D F	5	SPU B
822	KICK		5	Obj B
823	SYNC	D F	6	SPU C
824	KICK		6	Obj C
825	WAIT			
826	NOTIFY			SPU all

800

【図 15】

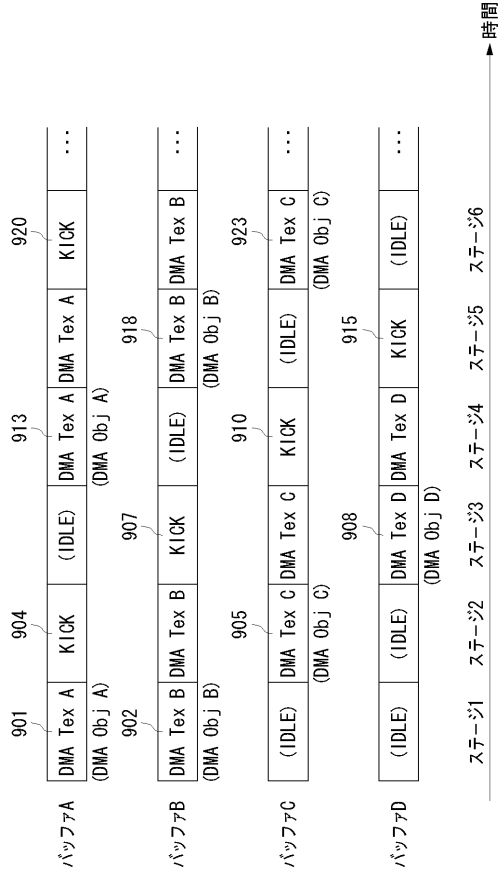


【図 16】

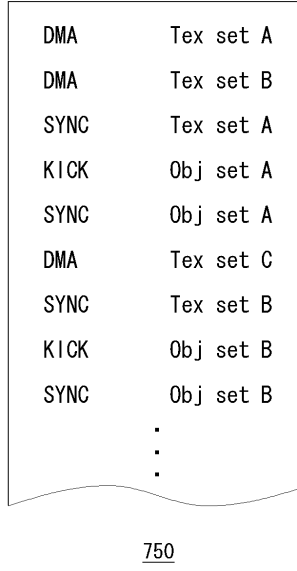
ID	タイプ	属性	ID	
901	DMA		1	Tex A
902	DMA		2	Tex B
903	SYNC	D F	1	SPU A
904	KICK		1	Obj A
905	DMA		3	Tex C
906	SYNC	D F	2	SPU B
907	KICK		2	Obj B
908	DMA		4	Tex D
909	SYNC	D F	3	SPU C
910	KICK		3	Obj C
911	SYNC	K	1	Obj A
912	NOTIFY		1	SPU A
913	DMA		5	Tex A
914	SYNC	D F	4	SPU D
915	KICK		4	Obj D
916	SYNC	K	2	Obj B
917	NOTIFY		2	SPU B
918	DMA		6	Tex B
919	SYNC	D F	5	SPU A
920	KICK		5	Obj A
921	SYNC	K	3	Obj C
922	NOTIFY		3	SPU C
923	DMA		7	Tex C
924	SYNC	D F	6	SPU B
925	KICK		6	Obj B
926	SYNC	K	4	Obj D
927	NOTIFY		4	SPU D
928	DMA		8	Tex D
929	SYNC	D F	7	SPU C
930	KICK		7	Obj C
931	SYNC	D F	8	SPU D
932	KICK		8	Obj D
933	WAIT			
934	NOTIFY			

900

【 図 17 】



【 図 18 】



---

フロントページの続き

(72)発明者 金子 基

東京都港区南青山2丁目6番21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 横山 佳弘

(56)参考文献 特開2003-196672(JP,A)

特開2002-312302(JP,A)

特開2002-259324(JP,A)

特開平10-187413(JP,A)

特開2002-259115(JP,A)

特開2002-254729(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 13/28

G06T 1/20

G06T 1/60

G06F 13/38

G06F 13/14

G06T 15/00