

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4545242号
(P4545242)

(45) 発行日 平成22年9月15日(2010.9.15)

(24) 登録日 平成22年7月9日(2010.7.9)

(51) Int. Cl.		F I			
G06T	1/60	(2006.01)	G06T	1/60	450F
G06T	1/20	(2006.01)	G06T	1/20	C
G06T	15/00	(2006.01)	G06T	15/00	300
G06F	12/08	(2006.01)	G06F	12/08	501D
G09G	5/00	(2006.01)	G09G	5/00	550X

請求項の数 9 (全 12 頁)

(21) 出願番号	特願平10-154408	(73) 特許権者	398038580
(22) 出願日	平成10年6月3日(1998.6.3)		ヒューレット・パッカード・カンパニー
(65) 公開番号	特開平11-102435		HEWLETT-PACKARD COMPANY
(43) 公開日	平成11年4月13日(1999.4.13)		アメリカ合衆国カリフォルニア州パロアルト
審査請求日	平成17年5月18日(2005.5.18)		ト ハノーバー・ストリート 3000
(31) 優先権主張番号	08/870152	(74) 代理人	100059959
(32) 優先日	平成9年6月5日(1997.6.5)		弁理士 中村 稔
(33) 優先権主張国	米国 (US)	(74) 代理人	100067013
前置審査			弁理士 大塚 文昭
		(74) 代理人	100084009
			弁理士 小川 信夫

最終頁に続く

(54) 【発明の名称】 ノンブロッキング・パイプライン・キャッシュ

(57) 【特許請求の範囲】

【請求項1】

読み出し要求で指定されたメモリ・アドレスを記憶し関連するライン・アドレスを持つ複数のラインを含む、前記読み出し要求を受け取るアドレス・キャッシュ(310)と、

該アドレス・キャッシュとメモリ・コントローラ(399)に接続されて、前記アドレス・キャッシュから受け取った、前記アドレス・キャッシュ(310)における読み出し要求のミスに対応する前記メモリ・アドレスのそれぞれを記憶し、該ミスされたメモリ・アドレスのそれぞれを前記メモリ・コントローラに送るアドレス・キュー(320)であって、前記メモリ・コントローラ(399)は、前記ミスされたメモリ・アドレスのそれぞれに対応するデータのメモリ(220)からの読み出しを行わせるものであるアドレス・キュー(320)と、

前記メモリ・コントローラ(399)に接続されて、該メモリ・コントローラから受け取った、前記ミスされたメモリ・アドレスのそれぞれに対応する前記データを記憶するデータ・キュー(350)と、

前記アドレス・キャッシュ(310)に接続されて、前記アドレス・キャッシュ(310)で受け取った前記読み出し要求のそれぞれに対応する、前記アドレス・キャッシュ(310)から受け取ったライン・アドレス及びヒット/ミス情報を記憶するプローブ結果キュー(330)と、

前記データ・キューと前記プローブ結果キューに接続されて、該データ・キューから受け取った、前記ミスされたメモリ・アドレスのそれぞれに対応する前記データを記憶する

10

20

データ・キャッシュ(360)と、

前記データ・キュー、前記プローブ結果キュー、及び前記データ・キャッシュに接続された手段(370)であって、前記プローブ結果キューのヘッド・エントリに記憶された前記ヒット/ミス情報がヒットを指示したときに前記データ・キャッシュからの出力を選択し、前記プローブ結果キューのヘッド・エントリに記憶された前記ヒット/ミス情報がミスを示したときに、前記データ・キュー(350)からの出力を選択するものであり、ここで前記データは前記データ・キュー(350)から選択されたときに前記ヘッド・エントリの前記ライン・アドレスにおいて前記データ・キャッシュ(360)にも記憶されているものである、ノンブロッキング・パイプライン・キャッシュを与える手段(370)と、を備え、

10

出力を選択するための前記手段(370)によって選択された前記データは、ヒット又はミスのいずれの場合でも、前記データが前記アドレス・キャッシュ(310)において要求されたのと同じ順番であることを特徴とする、データをキャッシュして取り出す装置(300)。

【請求項2】

前記アドレス・キャッシュ(310)は完全連想型の内容参照可能メモリであり、前記メモリ・アドレスはタグとして記憶される請求項1記載の装置。

【請求項3】

前記プローブ結果キュー(330)は、さらに、
複数のエントリを持ち、各エントリはN+1ビット幅で、Nは固有のライン・アドレスを示し、1ビットはヒット/ミス情報を示す請求項1記載の装置。

20

【請求項4】

前記データ・キャッシュ(360)は、相互接続されたレジスタ・ファイルである請求項1に記載の装置。

【請求項5】

前記記憶されるデータは、テクセル・データである、請求項1に記載の装置。

【請求項6】

前記データを記憶する同期ダイナミック・アクセス・メモリ(220)を更に備える請求項1に記載の装置。

【請求項7】

前記ミスされたアドレスにあるデータは、ミスの順番で応答される、請求項1に記載の装置。

30

【請求項8】

前記複数のラインは、最後に書かれた順番で保持される、請求項1に記載の装置。

【請求項9】

関連したライン・アドレスを有する複数のラインを持つアドレス・キャッシュ(310)において受け取った、読み出し要求で指定されるメモリ・アドレスをアドレス・キャッシュ(310)に記憶する段階と、

アドレス・キュー(320)に、前記アドレス・キャッシュから受け取った、前記アドレス・キャッシュにおける読み出し要求のミスに対応する前記メモリ・アドレスのそれぞれをキューする段階と、

40

前記ミスされたメモリ・アドレスのそれぞれをメモリ・コントローラ(399)に送り、前記ミスされたメモリ・アドレスのそれぞれに対応するデータを前記メモリ・コントローラによってメモリ(220)から取り出す段階と、

データ・キュー(350)に、前記メモリ・コントローラ(399)によって取り出された前記ミスされたメモリ・アドレスのそれぞれに対応する前記データをキューする段階と、

プローブ結果キュー(330)に、前記アドレス・キャッシュ(310)から受け取った、前記読み出し要求のそれぞれに対応する前記ライン・アドレス及びヒット/ミス情報をキューする段階と、

50

データ・キャッシュ(360)に、前記データ・キューから受け取った、前記ミスされたメモリ・アドレスのそれぞれに対応する前記データを記憶する段階と、

前記データ・キュー、前記プローブ結果キュー、及び前記データ・キャッシュに接続された手段(370)によって、前記プローブ結果キューのヘッド・エントリに記憶された前記ヒット/ミス情報がヒットを指示したときに、前記データ・キャッシュ(360)からの出力を選択し、前記プローブ結果キューのヘッド・エントリに記憶された前記ヒット/ミス情報がミスを指示したときに、前記データ・キューから出力を選択するものであり、ここで前記データは前記データ・キュー(350)から選択されたときに前記ヘッド・エントリの前記キャッシュ・ライン・アドレスにおいて前記データ・キャッシュ(360)にも記憶されているものである、ノン ブロッキング・パイプライン・キャッシュを与える段階と、を含み、

選択された前記データは、ヒット又はミスのいずれの場合でも、前記データが前記アドレス・キャッシュ(310)において要求されたのと同じ順番であることを特徴とする、データをキャッシュして取り出す方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、コンピュータ・システム・メモリ、より詳細にはグラフィック・コンピュータ・システムに使われるキャッシュ・メモリに関する。

【0002】

【従来の技術】

グラフィック・コンピュータ・システムでは、テクスチャ・マッピング(すなわち「テクスチャリング」)により、システムが生成する3次元(3D)イメージの現実性を高めることができる。テクスチャの例としては、木目、れんが、カーペット、ストーンウォール、草地などが挙げられる。テクスチャは、イメージ・オブジェクトの表面に、より現実的なレンダリングを与える。テクスチャリングは、コンピュータ・メモリ内にテクスチャ・マップとして記憶される一連のテクスチャ・エレメント(テクセル)を使って実行される。テクスチャ・マップは合成されるか、またはスキャンされたイメージから得られる。

低解像度のテクチャ・マップは、64×64テクセル含んでいるのに対し、高解像度のテクスチャ・マップは、4096×4096テクセル含んでいる。典型的な場合には、テクセルは、データのワードとして記憶され、各ワードのアドレスがテクスチャ・マップの特定の座標を表す。データは、色(RBG)と、おそらく透過性情報も表す。

【0003】

表面がテクスチャ化されたオブジェクトを含むグラフィック・イメージを表示するには、グラフィック・ソフトウェアとハードウェアがその表面を、表示可能な画素(ピクセル)に関連するスクリーン座標のアレーに変換する。ピクセル座標は、テクセル・マップの対応する該当のテクセルを探し出すのに使われる。対応するテクセルの色と透過性値は、ピクセルデータに併合され、表示されたピクセルの色と透過性の最終値を決定する。表面よりも内側のテクセル座標は、オブジェクトの頂点に与えられるテクセル座標を補間することにより得られる。

「ポイントサンプリング」と呼ばれる、低品質のテクスチャ・マッピングでは、イメージのピクセルそれぞれに対し、テクセルが1つだけ使われる。その結果、ポイントサンプリングでテクスチャ化されたイメージには、そのテクスチャ化された表面に、識別できる不連続的な形の見苦しいエイリアシングアーティファクトができる傾向がある。これは、テクスチャ化される表面が、ひどく歪められている場合(例えば距離をひき延ばしてある3Dイメージの表面)に特にそうである。

【0004】

3線形のテクスチャ・マッピングなどの高品質のテクスチャリングでは、多重テクセル・マップ、例えば「Mipem In Parvo」(many in place)マップ、もしくは、Mipmapsが使われる。例えば、ある特定のテクスチャを表すのにMipmapが全部で11ある場合、テクス

10

20

30

40

50

チャの第1の高解像度Mipmapは、 1024×1024 テクセルであり、第2の高解像度Mipmapは、 512×512 テクセルである。

その次は、 256×256 テクセルで、1番の低解像度のMipmapまで下がると、 1×1 テクセルとなる。ズームを使用して、3次元感覚を与えるためにオブジェクトのサイズを増減する場合でさえも、これらの複数マップから、歪められた表面を持つテクスチャはスムーズに補間される。

高品質テクスチャ・マッピングでは、1つのピクセルに対し、8または16のテクセルのマッピングを必要とすることがある。これは、各ピクセルに対して、システムは、8または16のメモリ・アドレスのテクセル・データにアクセスしなければならないということの意味している。この場合、テクスチャリングは大量のメモリ・システム帯域幅を消費してしまうことが明白である。

【0005】

テクスチャ・マッピングで必要とされるメモリ帯域幅を減らすことは望ましい。メモリ帯域幅を減らすことによって、テクスチャ専用のメモリ・チップの数を減らすことが可能となり、テクスチャを記憶するのに、より安価な汎用低速ダイナミック・ランダム・アクセス・メモリ(DRAM)を使用することが可能となる。そして、イメージ生成の間は他のデータに使われるものと同じメモリ内にテクスチャを記憶することが可能となる。

従来技術でのテクスチャ・マッピング装置の中には、専用の高速スタティック・ランダム・アクセス・メモリ(SRAM)内にテクスチャ・マップを記憶するものもある。この方法では、連続するデータが同一のアドレスから読み出される場合でさえ、テクセル・データの読み出し要求ごとにSRAMにアクセスする。

テクスチャ・マッピング用に特別に設計されたSRAMは、高価で、グラフィック・ハードウェアに内蔵されており、限られた機能しか持たないことが多い。

現在のDRAMでは、センス・アンプはデータを「キャッシュ」するのに使用される。キャッシュを行うと、データの空間的、時間的局所性を有効に利用できる。例えば、一連のテクセル・アドレスが、すべて同じDRAMのページにある場合、データを、センス・アンプから直接アクセスすることができる。

【0006】

DRAMを使うと、同じメモリ・ページから取り出すためのメモリ帯域幅は、SRAMのメモリ帯域幅に近づくことができる。しかし、現在のページのアドレスに「ミス」があると、他のページにアクセスする必要がある。DRAMのページ間でスイッチすると、次のページのデータを取り出し、センス・アンプにラッチ(保持)する間、いくつかのプロセッサ・サイクルが必要である。これにより、アクセス待ち時間は増加する。このような待ち時間は、メモリ・システムの平均帯域幅がページ取り出しを処理するのに十分である場合は、アクセス・パス内で長いパイプラインを使用することにより、隠すことができる。メモリ・システムに実際にキャッシュを追加すると、データに良好な空間的・時間的局所性がある場合、必要な帯域幅を減少させることができる。しかし、グラフィック装置用のキャッシュ・メモリを実施することは困難である。キャッシュが従来通りのブロッキング・キャッシュとして構成されている場合、ミスは、次のアクセスを停止する。何故ならば、そのミスが、完全に処理されてからでないと、次のアクセス要求を受け入れることはできないからである。これは、取り出されたデータは、次の要求が処理される前にどこかにラッチされなければならないという事実によるものである。より多くのミスがある場合は、このアクセス停止のために、メモリ・システムが渡すテクセル取り出し用帯域幅は、パイプライン・ノンキャッシュ・メモリ・システムの場合よりも小さくなる。

【0007】

ミス・サービスされたブックキーピング論理が含まれていると、このキャッシュは、ノンブロッキングになる。しかし、ノンブロッキング・キャッシュが、例えば読み出し要求から入手可能なデータまでのパイプライン内の段階と同じ数のミスをトラックできなければ、そのキャッシュは障害物となってってしまう。

待ち時間とキャッシュ・サイズの問題は、メモリがテクセル・データ、ピクセル・データ

10

20

30

40

50

を記憶するだけでなく、他の情報も記憶する場合にはさらに難しくなる。メモリが汎用低価格DRAMから要望通りに構成されている場合、様々な異なる種類のグラフィック情報を記憶できる。しかし、この場合、様々なバッファへのアクセス要求は、ページ“スラッシング”を避けるために、一括処理されなければならない。アクセスの一括処理によって、テクセル・データに対する要求が遅れ、待ち時間やトラックの必要があるミスが増加する。

【0008】

ダイレクト・マップされたキャッシュは、使用可能と考えられる。しかし、その場合、キャッシュは、アドレスがキャッシュの全体に亘って適当に分布されるように非常に大きいものでなければならないだろう。キャッシュが数十ラインの場合、データの一部が性能を低下させるコンフリクトを頻繁に起こしている間、他のデータの一部は、かなりの時間使われないうままキャッシュの無駄になることがある。キャッシュが大きくなると、コストも増大する。

10

ノン・ブロッキング完全連想型の内容参照可能メモリ(CAM)は、データのフル・アドレスをタグとして使用するが、これはテクスチャ・マッピングに、より適していることがある。しかし、それでも、完全連想型キャッシュは適切なサイズであることが必要であり、ミスのある場合の待ち時間が、読み出し要求が出される時間と、データが使用可能になる時間との間に大きな遅れを引き起こすことがある。テクスチャ・マッピングの固有性のために、キャッシュがミスを起こす可能性が増大する。さらに悪いケースでは、各テクセルは正確に一度ずつ使われ、その結果各アクセスでミスが起こることにもなり、このキャッシュには、全く利点がない。

20

【0009】

【発明が解決しようとする課題】

したがって、従来のキャッシュの持つ欠点がなく、テクセル取り出しによって要求される帯域幅を減少できる、グラフィック・システム用のキャッシュを提供することが望まれている。

【0010】

【課題を解決するための手段】

本発明は、広義においては、本特許請求の範囲の請求項1に限定されているような、データをキャッシュする装置と、請求項10に限定されているような、データをキャッシュする方法にある。

30

以下にテクセル・データにアクセスするグラフィック・プロセッサで使用可能なノンブロッキング・パイプライン・キャッシュについて説明する。ノンブロッキング・パイプライン・キャッシュは、テクセルのアドレスとデータを、先入れ先出しキューにより相互に接続しているアドレスとデータのキャッシュに保存することが好ましい。アドレスとデータは以下に述べる通り、空間的、時間的に別個の方法で保存されると効果的である。

【0011】

アドレス・キャッシュには、複数のラインがある。各ラインは、読み出し要求で指定されたメモリ・アドレスを記憶できる。各ラインに関連するのは、ライン・アドレスである。第1のキューには、アドレス・キャッシュに接続された入力と、メモリ・コントローラに接続された出力がある。第1のキューは、読み出し要求ミスによるミス・メモリ・アドレスを記憶する。ミス・メモリ・アドレスはメモリ・コントローラに送られる。メモリ・コントローラに接続された第2のキューは、ミス・メモリ・アドレスに保存されたデータを受け取る。

40

各読み出し要求のためのアドレス・キャッシュに接続された第3のキューは、ライン・アドレスとヒット/ミス情報を保存する。ヒット/ミス情報は、特定の要求されたアドレスがアドレス・キュー内に保存されているかどうかを示す。

【0012】

データ・キャッシュは、第2と第3のキューの出力に接続されている。データ・キャッシュは、第2のキューから受け取ったミス・メモリ・アドレスにあるデータを保存する。マ

50

ルチプレクサは、第3のキューに保存されたヒット/ミス情報により、出力へのデータをデータ・キャッシュまたは第2のキューから選択する。

【0013】

【発明の実施の形態】

本発明についてのより詳細な理解は、以下の好ましい実施態様から、および添付の図を参照して得られる通りである。

図1について、本発明のノンブロッキング・パイプライン・キャッシュの好ましい実施態様について詳細を説明する。グラフィック表示システム100は、メモリ・バス115によってメイン・メモリ120に接続されている汎用中央処理装置(CPU)110を具備する。CPU110は、汎用バス125(例えばPCIバスなど)にも接続されている。バス125は、ディスク・サブシステム130、入力/出力(I/O)サブシステム140、グラフィックジェネレータ200に接続されることができる。グラフィック・ジェネレータ200は、ライン270によって、グラフィック・ディスプレイ装置150に接続されている。システム100は、グラフィック・ワークステーションとして構成されることができる。

10

【0014】

CPU110は、1つ以上の汎用プロセッサチップ、例えば米国マサチューセッツ州メイナードの(Maynard, MA) デジタル・エキップメント・コーポレーション(Digital Equipment Corporation)製のALPHA プロセッサを包含できる。

メインメモリ120は、システム100の動作中に、マシン実行可能な命令とデータを記憶するためのものである。ディスク130は、ファイル内に命令とデータを永続的に記憶する。命令とデータは、入力/出力(I/O)サブシステム140を介して得られることができる。入力/出力(I/O)システムは、システム100の動作を制御するキーボード、マウス、ジョイスティックなどの入力/出力(I/O)装置(図示していない)にも接続できる。バス115とバス125は、アドレス、データ、制御、タイミング信号を運ぶ。

20

【0015】

システム100の動作中に、ソフトウェア・プログラムの命令がCPU110によって実行される。プログラムは、ディスプレイ装置150上でグラフィック・イメージを生成するように設計されている。イメージ用のデータとイメージをレンダリングする方法についての命令は、バス115およびバス125を経由してグラフィック・ジェネレータ200に送られる。ジェネレータ200はデジタルデータを、ディスプレイ装置150が動かすことができるアナログ信号に変換する。より詳細には、グラフィックジェネレータ200は、3次元イメージを高速でレンダリングする。

30

図2に示す通り、ジェネレータ200は、図1のバス125に接続するためのバス・インタフェース205を包含する。特定目的グラフィック・プロセッサ210は、ランダム・アクセス・メモリ/デジタル・ツー・アナログ・コンバータ(RAMDAC)230に接続される。RAMDAC230は、ライン260上のデジタル・グラフィック・データを受け取り、ライン270でアナログ・ビデオ信号を生成する。

【0016】

40

さらにグラフィック・プロセッサ210は、同期ダイナミック・アクセス・メモリ(SDRAM)220およびテクセル・キャッシュ300にも接続されている。SDRAM220は、マップとバッファ内にグラフィック・データを記憶する。バッファは、フロント、バック、色、Z、ステンシル、オーバーレイ、フォーマット・バッファを含むことができる。マップは、テクスチャ・データ(テクセル)を記憶できる。エレメント205、210、300は、シングル・セミコンダクター・チップ201上に配置されることが好ましい。

動作中に、ジェネレータ200は図1に示すCPU110からグラフィック・データと命令を受け取る。データは、マップおよびバッファとしてSDRAM220内に記憶される。バッファとマップのデータは、グラフィック・プロセッサ210によって結合され、ラ

50

イン 260 上にグラフィック・データを生成する。

アクセスされるテクセルが多数あるため、またシステムの性能を良くするために、テクセル・キャッシュ 300 は、テクセル・データをキャッシュするために使われる。

【0017】

一つの実施態様では、各テクセル・キャッシュ 300 のために 1 つのメモリ・コントローラ 399 がある。例えば、8 個のテクセル・キャッシュ 300 が使われる場合、8 個のメモリ・コントローラ 399 があることになる。従来のキャッシュと比較すると、各キャッシュのライン数は、例えば、2、3、4、8、12 などの小さい整数である。

小さいサイズのキャッシュ 300 と、長いメモリ待ち時間に対処するためには、キャッシュ 300 は、パイプラインを使って空間的かつ時間的にデータ及びそのアドレス・タグを分けるように構成される。これによって、キャッシュ 300 はノンブロッキングであることが可能であり、キャッシュ 300 内にラインがあるよりも、より多くの“ミスされた”要求を保持することができる。これは、性能を向上させるだけでなく、設計を単純化する。

【0018】

図 3 は、より詳細なノンブロッキング・パイプライン・テクセル・キャッシュ 300 を示す。キャッシュ 300 は、アドレス・キャッシュ 310 を包含する。

アドレス・キャッシュ 310 は、アドレス・タグとして 22 ビット使用する完全連想型の内容参照可能メモリ (CAM) である。好ましい実施態様では、アドレス・キャッシュ 310 内のライン数は少なく、例えば 8 である。さらに、アドレス・キャッシュ 310 は、循環カウンタ内に書き込まれた最新の情報も保持している。最後に書き込まれたラインは、ミスが起こる場合、次の“犠牲”となる。キャッシュ・ラインは関連するライン・アドレス (例えば 0 から 7) を持つ。

アドレス・キャッシュ 310 は、アドレス・キュー 320 の入力エンド (テール) に結合されている。アドレス・キュー 320 は、22 ビット幅の先入れ先出し (FIFO) バッファである。アドレス・キュー 320 は、最後にミスされたアクセスのアドレス (タグ) を記憶する。

【0019】

アドレス・キュー 320 の出力エンド (ヘッド) は、図 2 のメモリ・コントローラ 399 の一つの入力に結合されている。メモリコントローラ 399 からの出力は、データキュー 350 の入力エンド (テール) に接続されている。データ・キューは、32 ビット幅の先入れ先出し (FIFO) バッファである。

アドレス・キャッシュ 310 は、ライン 302 と 303 によってプローブ結果キュー 330 の入力エンドにも接続されている。プローブ結果キュー 330 は、 $N + 1$ ビット幅の先入れ先出し (FIFO) バッファであり、ライン 303 上の N ビットがアドレス・キャッシュ 310 の特定のラインを示すのに使われ、ライン 302 上の 1 ビットはヒット/ミス情報を示すのに使われる。

データ・キャッシュ 360 とマルチプレクサ (MUX) 370 は両方共、プローブ結果キュー 330 のヘッドとデータ・キュー 350 のヘッドから入力を受け取る。データ・キャッシュ 360 は、SRAM として、相互接続されたレジスタ・ファイルとして、もしくはラッチのセットとして実施されうる。MUX 370 は、32 ビット幅でライン 309 上に複数の入力と 1 つの出力を持つ。MUX の出力はライン 302 と 303 を経由してプローブ結果キュー 330 に与えられ、最終的にライン 304 上に出される情報によって選択される。すなわち、選択にはライン 302 と 303 上の信号によって時間的遅れが発生する。この遅れは、その情報がプローブ結果キュー 330 のヘッドに達する時間である。これは、選択が、プローブ結果キューのヘッド・エントリにある情報にじかに基づいてなされるということの意味している。

【0020】

動作中に、キャッシュ 300 はライン 301 上の読み出しアクセス要求に結びついている「プローブ」アドレスを受け取る。このプローブ・アドレスは、アドレス・キャッシュ 310 のすべてのラインを同時にプローブするのに使われる。

ヒットするものがあると、ライン302上のヒット/ミス情報がロジカル「1」（高）にセットされる。ミスの場合、ライン302はロジカル「0」にセットされる。

ヒットした場合、ヒット・ライン・アドレス（Nビット）は、ライン303上で表される。ミスの場合、犠牲となるライン・アドレスが表される。これは、あるN+1ビット・エンタリー（ライン・アドレス+ヒット/ミス情報）がヒットかミスかとして、各キャッシュ・アクセスのためのプローブ結果キュー330のテールに置かれることを意味している。

【0021】

ミスの場合、プローブ・アドレスはアドレス・キュー320に置かれる。アドレス・キュー320は、メモリ・コントローラ399を介して待ち時間の長いSDRAM220に対してなされた要求を処理する。コントローラ399からの応答は、ミスしたアドレスのデータである。このデータは、データ・キュー350に置かれる。

データ・キャッシュ360は、プローブ結果キュー330のエンタリーを使ってアクセスされる。プローブ結果キューのヘッドがヒットを示す場合、ライン304上のNビットはデータ・キャッシュ360から要求されたテクセル・データを読み出すためにキャッシュ・ライン・アドレスとして使われる。ヒットの場合、MUX370は選択されたデータ・キャッシュ・アドレスにあるデータ・キャッシュ360からのデータを、ライン309上の出力として選択するようにさせられる。

【0022】

このプローブ・エンタリーがミスを示した場合、データ・キュー350の次の先入れ先出し（FIFO）バッファ・エンタリーがライン304上のプローブ・キュー・エンタリーのNビットで指定されるキャッシュ・ライン・アドレスのデータ・キャッシュ360内に記憶される。この場合、MUX370はデータ・キュー350からのデータを、ライン309上の出力として選択する。ヒットまたはミスのいずれかの場合、データは要求されたのと同一の順番でライン309上に与えられる。

選択には、他の方法も使用される。例えば、選択は、「透過性のある」ラッチによって実行される。このラッチは、サイクルの始めにデータ・キュー330からデータ・キャッシュ360にミスしたデータを「ロード」し、これによって、サイクル中にデータ・キューからのデータは、ラッチを通過して「流れる」ことが可能となる。

【0023】

アドレス・キャッシュ310にヒットがある場合、またはアドレス・キャッシュ320が一杯でない場合、新しい読み出し要求が受け付けられる。アドレス・キャッシュにミスがあり、かつ、アドレス・キュー320が一杯である場合、読み出し要求は、スペースがアドレスキュー320において使用しうるようになるまで停止されねばならない。データ・キャッシュ360にミスがある場合、またはデータ・キャッシュ350が空でない場合、応答データが利用しうる。データ・キャッシュ360にミスがあり、さらにデータ・キュー350が空の場合は、その応答はデータがデータ・キュー350内にて利用しうるようになるまで、停止されねばならない。

図4の「網掛けされた」2×2ピクセルの正方形T0からT15は、16のテクスチャ・マップ化されたピクセルがどのようにテクスチャMipmap上でマップ化され、テクセル取り出し要求を生成するかの例を示す。網掛けされた正方形はそれぞれ、その正方形が重なっている4つのテクセルの取り出しを要求する。

ラベルA0からJ3のAからJは、異なる22ビット・アドレスを示す。0、1、2、3の数は、異なるメモリ・コントローラ399、したがってテクセル・データがキャッシュされるべき異なるキャッシュを示す。

【0024】

各キャッシュ300に8ラインある場合、ヒットとミスの比率は良い。テクセルが適切な方法で取り出される場合、たとえ、ライン・キャッシュが2つであっても、性能は良い。図5の表500は、メモリ・コントローラ0のテクセル取り出し要求のための動的なヒットとミス情報520を示す。図5ではライン・キャッシュは8と想定して、縦列510が

10

20

30

40

50

テクセル取り出し要求番号を、縦列 5 1 5 がプローブ要求アドレスを、縦列 5 2 0 がヒットまたはミス情報を、縦列 5 3 0 がキャッシュ・ライン・アドレスを示す。

テクセルをキャッシュするためのノンブロッキング・パイプライン・キャッシュの有用性について説明してきた。しかし、同じ設計は他のグラフィック・アプリケーションでも、より大きい行先となる四角形上に小さい元となる四角形をタイルすることにも使用されることができる。

【 0 0 2 5 】

以上は本発明の特有の実施態様を説明したものである。しかし、本発明の範囲内で様々な変更を加えることが可能であることは、当業者にとって明らかである。

【 図面の簡単な説明 】

【 図 1 】本発明の好ましい実施態様による、ノンブロッキング・パイプライン・キャッシュを使うグラフィック表示システムのブロック図である。

【 図 2 】図 1 のシステムのグラフィック・ジェネレータのブロック図である。

【 図 3 】本発明の好ましい実施態様による、ノンブロッキング・パイプライン・キャッシュのブロック図である。

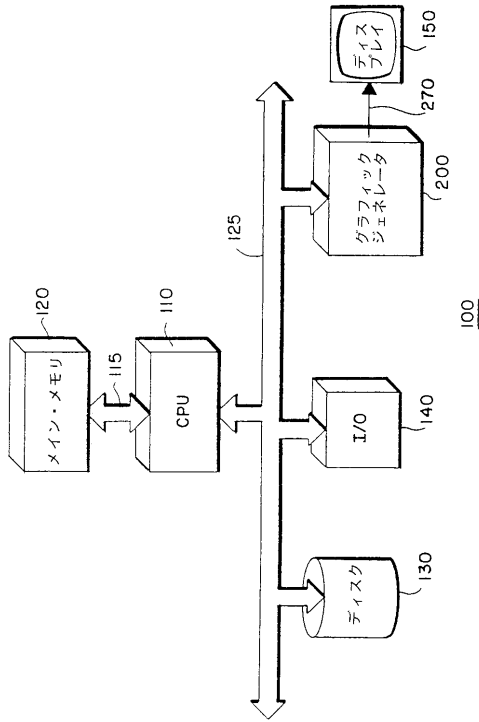
【 図 4 】テクスチャ・マップからのテクセル取り出しの線図である。

【 図 5 】 8 ライン・キャッシュに関連するヒット/ミス情報の表を示す図である。

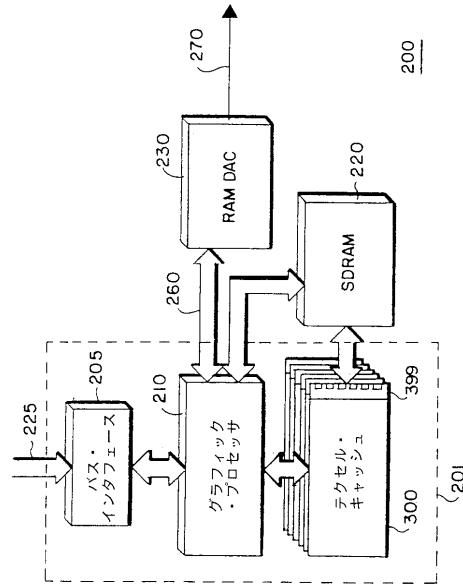
【 符号の説明 】

1 2 0	メイン・メモリ	
1 1 0	中央処理装置 (C P U)	20
1 3 0	ディスク	
1 4 0	入力/出力 (I / O)	
1 5 0	ディスプレイ	
2 0 0	グラフィック・ジェネレータ	
2 0 5	バス・インタフェース	
2 1 0	グラフィック・プロセッサ	
2 2 0	同期ダイナミック・アクセス・メモリ (S D R A M)	
2 3 0	ランダム・アクセス・メモリ/デジタル・ツー・アナログ・コンバータ (R A M D A C)	
3 0 0	テクセル・キャッシュ	30
2 2 1	メモリ・コントローラ	
3 0 1	読み出し要求	
3 0 2	ヒット/ミス	
3 0 3	エントリ/アドレス	
3 0 9	応答	
3 1 0	アドレス・キャッシュ	
3 2 0	アドレス・キュー	
3 3 0	プローブ結果キュー	
3 5 0	データ・キュー	
3 6 0	データ・キャッシュ	40
3 7 0	マルチプレクサ (M U X)	
5 1 0	取り出されたテクセル番号	
5 1 5	プローブ要求アドレス	
5 2 0	ヒット/ミス	
5 3 0	キャッシュ・ライン・アドレス	

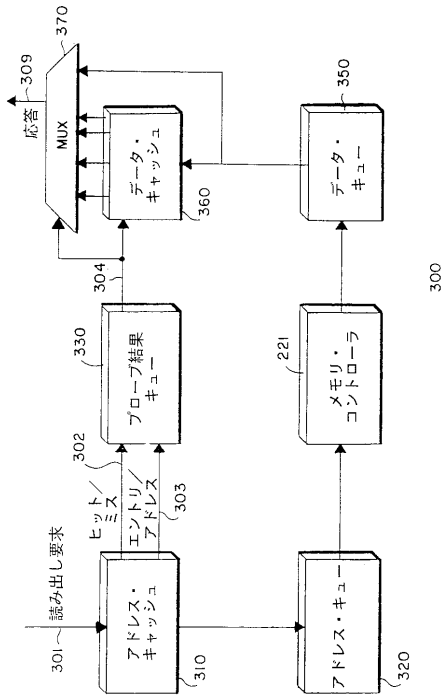
【図 1】



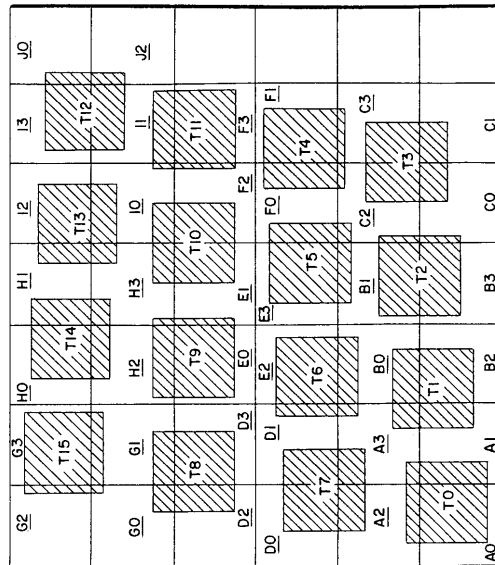
【図 2】



【図 3】



【図 4】



【図5】

500

取り出された テクセル番号	プローブ 要求 アドレス	ヒット/ ミス	キャッシュ ライン アドレス
T0	A0	ミス	0
T1	B0	ミス	1
T2	C0	ミス	2
T3	C0	ヒット	2
T4	F0	ミス	3
T5	F0	ヒット	3
T6	B0	ヒット	1
T7	D0	ミス	4
T8	G0	ミス	5
T9	E0	ミス	6
T10	I0	ミス	7
T11	I0	ヒット	7
T12	J0	ミス	0
T13	I0	ヒット	7
T14	H0	ミス	1
T15	G0	ヒット	5

510 515 520 530

フロントページの続き

- (72)発明者 ジョエル ジェイ マッコーマック
アメリカ合衆国 コロラド州 80302 ボールダー ピークビュー ロード 819
- (72)発明者 ケニス ダブリュー コーレル
アメリカ合衆国 マサチューセッツ州 01523ランカスター ノース メイン ストリート
2221
- (72)発明者 バートン ダブリュー バーコウィッツ
アメリカ合衆国 マサチューセッツ州 01701フレミングム アンジェリカ ドライヴ 84
- (72)発明者 クリストファー シー ジャーノス
アメリカ合衆国 マサチューセッツ州 01564スターリング パイクス ヒル ロード 14

審査官 菅原 道晴

- (56)参考文献 特開平04-270431(JP,A)
特開平08-328952(JP,A)
特開平07-234819(JP,A)
特開平06-012323(JP,A)
特開平02-090265(JP,A)
特開平03-217985(JP,A)
特開平06-318177(JP,A)
特開平08-314802(JP,A)
特開平08-339331(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06T 1/60
G06T 1/20
G06T 15/00
G06F 12/08