

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6097744号
(P6097744)

(45) 発行日 平成29年3月15日(2017.3.15)

(24) 登録日 平成29年2月24日(2017.2.24)

(51) Int.Cl. F I
G O 6 T 15/04 (2011.01) G O 6 T 15/04

請求項の数 20 (全 21 頁)

(21) 出願番号	特願2014-515991 (P2014-515991)	(73) 特許権者	591016172
(86) (22) 出願日	平成24年6月14日(2012.6.14)		アドバンスト・マイクロ・ディバイシズ・
(65) 公表番号	特表2014-520328 (P2014-520328A)		インコーポレイテッド
(43) 公表日	平成26年8月21日(2014.8.21)		ADVANCED MICRO DEVI
(86) 国際出願番号	PCT/US2012/042442		CES INCORPORATED
(87) 国際公開番号	W02012/174231		アメリカ合衆国、94088-3453
(87) 国際公開日	平成24年12月20日(2012.12.20)		カリフォルニア州、サニibel、ピー・
審査請求日	平成27年6月12日(2015.6.12)		オウ・ボックス・3453、ワン・エイ・
(31) 優先権主張番号	13/163,071		エム・ディ・プレイス、メイル・ストップ
(32) 優先日	平成23年6月17日(2011.6.17)		・68 (番地なし)
(33) 優先権主張国	米国 (US)	(74) 代理人	100108833
			弁理士 早川 裕司
		(74) 代理人	100111615
			弁理士 佐野 良太

最終頁に続く

(54) 【発明の名称】 シェーダプロセッサを用いたリアルタイムオンチップテクスチャ展開

(57) 【特許請求の範囲】

【請求項1】

第1のシェーダと、

前記第1のシェーダによる使用のために、テクスチャの圧縮バージョンの要求されたブロックを展開するように構成された第2のシェーダであって、前記テクスチャの圧縮バージョンは可変レート圧縮を用いて圧縮されている、第2のシェーダと、を含み、

前記テクスチャの圧縮バージョンに対して第1の仮想アドレス空間を割り当て、前記テクスチャの非圧縮バージョンに対して前記第1の仮想アドレス空間とは異なる第2の仮想アドレス空間を割り当てることと、

前記テクスチャの非圧縮バージョン内のブロックの位置を前記テクスチャの圧縮バージョン内のブロックの位置にマッピングするために、前記テクスチャの圧縮バージョンのブロックごとの位置を記憶するテーブルを利用することと、

ブロックに対する要求を、仮想アドレスと合わせて前記第1のシェーダから受信することであって、前記第1のシェーダは、前記テクスチャの非圧縮バージョン内の前記ブロックの仮想アドレスを判別するように構成されている、ことと、

前記仮想アドレスを用いてキャッシュにアクセスすることと、

前記ブロックが前記キャッシュに存在し、且つ、前記仮想アドレスが前記第1の仮想アドレス空間に対応していると判定したことに応じて、前記要求されたブロックを圧縮データとして処理することであって、前記テクスチャの圧縮バージョン内での前記要求されたブロックの位置を前記テーブルから判別し、前記要求されたブロックの圧縮バージョンを

10

20

前記第 2 のシェーダに提供し、前記要求されたブロックの非圧縮バージョンを前記第 2 のシェーダから受信し、前記要求されたブロックの非圧縮バージョンを前記第 1 のシェーダに提供する、ことと、

前記ブロックが前記キャッシュに存在し、且つ、前記仮想アドレスが前記第 2 の仮想アドレス空間に対応していると判定したことに応じて、前記ブロックを非圧縮データとして処理し、非圧縮ブロックを前記第 1 のシェーダに提供することと、を行うように構成されている、

装置。

【請求項 2】

前記テクスチャの圧縮バージョン内のブロックのサイズは可変であり、

前記テクスチャの圧縮バージョン内の前記要求されたブロックのサイズを判別するように構成されている、請求項 1 に記載の装置。

【請求項 3】

前記第 2 のシェーダは、要求されたブロックが、前記要求されたブロックのアドレスに基づいて圧縮または非圧縮されたかどうかを判別するように構成されており、

第 1 のアドレス範囲は非圧縮ブロックの仮想アドレスに対応し、第 2 のアドレス範囲は圧縮ブロックの物理アドレスに対応している、請求項 2 に記載の装置。

【請求項 4】

前記位置は、前記テクスチャの圧縮バージョンの始まりから前記要求されたブロックまでのオフセットに対応しており、

前記テクスチャの圧縮バージョンの複数のブロックごとに、D C 係数値を前記テーブルに記憶するように構成されている、請求項 2 に記載の装置。

【請求項 5】

前記テクスチャの圧縮バージョンのブロックの展開を容易にするために、前記テクスチャの圧縮バージョンのハフマン符号テーブル及び量子化テーブルを前記テーブルに記憶するように構成されている、請求項 4 に記載の装置。

【請求項 6】

前記テクスチャの圧縮バージョンを、J P E G (J o i n t P h o t o g r a p h i c E x p e r t s G r o u p) 圧縮テクスチャからハードウェア内部の J P E G フォーマットに再フォーマットするように構成されており、

前記ハードウェア内部の J P E G フォーマットは、前記テクスチャの圧縮バージョンの様々なブロックの位置及びサイズについての情報を有するテーブルを含む、請求項 1 に記載の装置。

【請求項 7】

前記位置は、前記テクスチャの圧縮バージョンの始まりから前記要求されたブロックまでのオフセットに対応しており、

前記要求されたブロックのオフセットと、前記テクスチャの圧縮バージョン内の隣接するブロックのオフセットとの差に基づいて、前記要求されたブロックのサイズを計算するように構成されている、請求項 1 に記載の装置。

【請求項 8】

前記位置は、キャッシュのページ内で前記要求されたブロックが記憶されているオフセットに対応している、請求項 1 に記載の装置。

【請求項 9】

前記第 1 のシェーダは、

前記仮想アドレスを前記テクスチャの圧縮バージョン内のアドレスにマッピングし、

前記テクスチャの圧縮バージョン内の前記アドレスをメモリに移す、

ように構成されている、請求項 1 に記載の装置。

【請求項 10】

オフチップのシステムメモリから取り出されたデータを記憶するように構成されたオンチップメモリを備え、

10

20

30

40

50

前記要求されたブロックの非圧縮バージョンが前記オンチップメモリに存在しないという判別に応じて、前記要求されたブロックの非圧縮バージョンのために前記オンチップメモリにストレージが割り当てられる、請求項 1 に記載の装置。

【請求項 1 1】

テクスチャデータを手続き的に生成するように構成された第 3 のシェーダを備え、

テクスチャデータの手続き的な生成は事前というよりむしろ動的に行われる、請求項 1 に記載の装置。

【請求項 1 2】

テクスチャデータを展開するための方法であって、

第 1 のシェーダが、テクスチャブロックを要求することと、

第 2 のシェーダが、前記第 1 のシェーダによる使用のために、可変レート圧縮テクスチャブロックを展開し、テクスチャの圧縮バージョン内での要求されたブロックの位置を判別することと、

前記テクスチャの圧縮バージョンに対して第 1 の仮想アドレス空間を割り当て、前記テクスチャの非圧縮バージョンに対して前記第 1 の仮想アドレス空間とは異なる第 2 の仮想アドレス空間を割り当てることと、

前記第 2 のシェーダが、前記テクスチャの非圧縮バージョン内のブロックの位置を前記テクスチャの圧縮バージョン内のブロックの位置にマッピングするために、前記テクスチャの圧縮バージョンのブロックごとの位置を記憶するテーブルを利用することと、

ブロックに対する要求を、仮想アドレスと合わせて前記第 1 のシェーダから受信することであって、前記第 1 のシェーダは、前記テクスチャの非圧縮バージョン内の前記ブロックの仮想アドレスを判別するように構成されている、ことと、

前記仮想アドレスを用いてキャッシュにアクセスすることと、

前記ブロックが前記キャッシュに存在し、且つ、前記仮想アドレスが前記第 1 の仮想アドレス空間に対応していると判定したことに応じて、前記要求されたブロックを圧縮データとして処理することであって、前記テクスチャの圧縮バージョン内での前記要求されたブロックの位置を前記テーブルから判別し、前記要求されたブロックの圧縮バージョンを前記第 2 のシェーダに提供し、前記要求されたブロックの非圧縮バージョンを前記第 2 のシェーダから受信し、前記要求されたブロックの非圧縮バージョンを前記第 1 のシェーダに提供する、ことと、

前記ブロックが前記キャッシュに存在し、且つ、前記仮想アドレスが前記第 2 の仮想アドレス空間に対応していると判定したことに応じて、前記ブロックを非圧縮データとして処理し、非圧縮ブロックを前記第 1 のシェーダに提供することと、を含む、

方法。

【請求項 1 3】

前記テクスチャの圧縮バージョン内のブロックのサイズは可変であり、

前記第 2 のシェーダが、前記テクスチャの圧縮バージョン内の前記要求されたブロックのサイズを判別すること、をさらに含む、請求項 1 2 に記載の方法。

【請求項 1 4】

前記第 2 のシェーダが、要求されたブロックが、前記要求されたブロックのアドレスに基づいて圧縮または非圧縮されたかどうかを判別すること、をさらに含む、

第 1 のアドレス範囲は非圧縮ブロックの仮想アドレスに対応し、第 2 のアドレス範囲は圧縮ブロックの物理アドレスに対応している、請求項 1 3 に記載の方法。

【請求項 1 5】

前記位置は、前記テクスチャの圧縮バージョンの始まりから前記ブロックまでのオフセットに対応しており、

前記テクスチャの圧縮バージョンの複数のブロックごとに、DC 係数値を前記テーブルに記憶することであって、前記テクスチャの前記複数のブロックの各々は、 8×8 のピクセルのタイルである、ことをさらに含む、請求項 1 3 に記載の方法。

【請求項 1 6】

前記テクスチャの圧縮バージョンを、J P E G (J o i n t P h o t o g r a p h i c E x p e r t s G r o u p) 圧縮テクスチャからハードウェア内部のJ P E Gフォーマットに再フォーマットすること、をさらに含み、

前記ハードウェア内部のJ P E Gフォーマットは、前記テクスチャの圧縮バージョンの様々なブロックの位置及びサイズについての情報を有するテーブルを含む、請求項12に記載の方法。

【請求項17】

前記位置は、キャッシュのページ内で前記要求されたブロックが記憶されているオフセットに対応している、請求項12に記載の方法。

【請求項18】

テクスチャデータを展開するためのプログラム命令を記憶するコンピュータ可読記憶媒体であって、前記プログラム命令は、実行されると、

第1のシェーダに、テクスチャブロックを要求させることと、

第2のシェーダに、前記第1のシェーダによる使用のために、可変レート圧縮テクスチャブロックを展開させ、テクスチャの圧縮バージョン内での要求されたブロックの位置を判別させることと、

前記テクスチャの圧縮バージョンに対して第1の仮想アドレス空間を割り当て、前記テクスチャの非圧縮バージョンに対して前記第1の仮想アドレス空間とは異なる第2の仮想アドレス空間を割り当てることと、

前記第2のシェーダに、前記テクスチャの非圧縮バージョン内のブロックの位置を前記テクスチャの圧縮バージョン内のブロックの位置にマッピングするために、前記テクスチャの圧縮バージョンのブロックごとの位置を記憶するテーブルを利用させることと、

ブロックに対する要求を、仮想アドレスと合わせて前記第1のシェーダから受信することであって、前記第1のシェーダは、前記テクスチャの非圧縮バージョン内の前記ブロックの仮想アドレスを判別するように構成されている、ことと、

前記仮想アドレスを用いてキャッシュにアクセスすることと、

前記ブロックが前記キャッシュに存在し、且つ、前記仮想アドレスが前記第1の仮想アドレス空間に対応していると判定したことに応じて、前記要求されたブロックを圧縮データとして処理することであって、前記テクスチャの圧縮バージョン内での前記要求されたブロックの位置を前記テーブルから判別し、前記要求されたブロックの圧縮バージョンを前記第2のシェーダに提供し、前記要求されたブロックの非圧縮バージョンを前記第2のシェーダから受信し、前記要求されたブロックの非圧縮バージョンを前記第1のシェーダに提供する、ことと、

前記ブロックが前記キャッシュに存在し、且つ、前記仮想アドレスが前記第2の仮想アドレス空間に対応していると判定したことに応じて、前記ブロックを非圧縮データとして処理し、非圧縮ブロックを前記第1のシェーダに提供することと、を行わせる、

コンピュータ可読記憶媒体。

【請求項19】

前記テクスチャの圧縮バージョン内のブロックのサイズは可変であり、

前記プログラム命令は、前記第2のシェーダに、前記テクスチャの圧縮バージョン内の前記要求されたブロックのサイズを判別させる、請求項18に記載のコンピュータ可読記憶媒体。

【請求項20】

前記プログラム命令は、前記第2のシェーダに、要求されたブロックが、前記要求されたブロックのアドレスに基づいて圧縮または非圧縮されたかどうかを判別させ、

第1のアドレス範囲は非圧縮ブロックの仮想アドレスに対応し、第2のアドレス範囲は圧縮ブロックの物理アドレスに対応している、請求項19に記載のコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

10

20

30

40

50

【 0 0 0 1 】

本開示は、概して、グラフィックス処理、特にテクスチャ展開の処理ユニット、方法及び媒体に関する。

【 背景技術 】

【 0 0 0 2 】

コンピュータグラフィックスの処理システムは、通常、グラフィックスプロセッシングユニット（GPU）が処理の大部分を行い、大量のデータを処理する。GPUは、複雑な集積回路であり、特に、グラフィックス処理のタスクを行うように構成されている。例えば、GPUは、ビデオゲームアプリケーションなどのエンドユーザが必要とするグラフィックス処理タスクを実行してもよい。GPUは、個別の装置であってもよいし、例えば中央処理装置（CPU）などの他のプロセッサと同じ装置に含まれてもよい。

10

【 0 0 0 3 】

GPUは、レンダリングとして知られているプロセスで、画像の構成要素の高水準記述から画像を構成するピクセルを生成する。GPUは、通常、ピクセル、テクスチャ及び幾何データを処理するために演算素子を使用することによる継続レンダリングの概念を利用する。この演算素子は、ラスタライザ、セットアップエンジン、カラーブレンダ、隠面消去及びテクスチャマッピングの機能を実行してもよい。こうした演算素子は、シェーダ、シェーダプロセッサ、シェーダアレイ、シェーダパイプ、シェーダパイプアレイ、シェーダパイプライン又はシェーダエンジンと呼ばれることが多く、コンピュータグラフィックスにおける「シェーダ」という用語は、グラフィックスリソースによって主にレンダリング効果を果たすために使用されるソフトウェア命令のセット又はプログラムを意味する。「シェーダ」は、ソフトウェア命令を実行するために使用する実際のハードウェアの構成要素又はプロセッサを意味することもある。シェーダプロセッサ又はプログラムは、データを読み取り、レンダリングし、当該データのあらゆるタイプの処理を行い得る。また、統合シェーダを備えたGPUは、ピクセル、頂点、プリミティブ及び一般化の演算処理まで多くのタイプのシェーダ処理を同時にサポートする。

20

【 0 0 0 4 】

複雑なグラフィックスのシーン生成処理の多くは、テクスチャデータを伴う。テクスチャは、色、透明度、ルックアップテーブル又は他のデータなどの様々なタイプのデータであってもよい。いくつかの実施形態では、テクスチャは、視覚的な詳細を追加する幾何学形状に描かれる、デジタル化された画像であってもよい。テクスチャの使用による大量の詳細は、グラフィカルモデルが目的の画像をつくるためにレンダリングされるに従い、グラフィカルモデルの表面にマッピングされてもよい。テクスチャマッピングは、オブジェクトの表面に現実的な外観を与えることを目的としている。テクスチャは、通常マップ又はバンプマップの形式で、色又は鏡面反射若しくは微細表面の詳細などの表面のプロパティを含む多くのプロパティを指定してもよい。テクスチャは、画像データ、カラー若しくは透明のデータ、粗さ／滑らかさのデータ又は反射率データなどであることもある。「テクセル」は、「ピクセル」がピクチャエレメントであるのと同様に、テクスチャエレメントである。「テクセル」と「ピクセル」という用語は、本明細書では相互に置き換え可能に使用され得る。

30

40

【 0 0 0 5 】

3Dコンピュータグラフィックスでは、オブジェクトの表面詳細は、一般的にテクスチャを用いて付加される。例えば、レンガの2Dビットマップイメージを、建物の3Dモデルを表すポリゴンのセットに対してテクスチャマッピングを用いて適用して、レンガで作られている外観を、オブジェクトに3Dレンダリングする。現実的なコンピュータグラフィックスを提供するには、通常多くの高品質で詳細なテクスチャを必要とする。テクスチャを用いると、大量の記憶域と帯域幅を消費する場合があります、その結果、記憶域と帯域幅の利用を減らすために、テクスチャが圧縮され得る。

【 0 0 0 6 】

従って、テクスチャ圧縮は、一般的にグラフィックスハードウェア、特に3Dグラフィ

50

ックスハードウェアの機能として広く受け入れられるようになった。テクスチャ圧縮は、オリジナルテクスチャの品質を可能な限り保持しながら、グラフィックスシステムの記憶域と帯域幅のコストを減らすことを目的としている。本明細書で説明する圧縮と展開の方法は、画像データ、ピクチャデータ、透明度情報、滑らかさ若しくは粗さのデータ又は他の類似の構造化データを含む様々なタイプのテクスチャ情報を圧縮するために使用可能である。このように、テクスチャという用語は、本明細書では、GPUの一部として圧縮又は展開されているデータを意味するために広く使用する。

【0007】

従来、テクスチャを圧縮するために、固定レート圧縮スキームが使用されており、可変レートスキームに比べていくつかの欠点の影響を受け得る。可変レート圧縮は、固定レート圧縮と異なり、より柔軟性が高く、必要に応じて品質に対する調整が可能でありうる。例えば、可変レート圧縮は、無損失圧縮を達成するために設定してもよい。いくつかの例では、可変レート圧縮スキームを使用することで、従来の固定レート圧縮スキームよりも優れた圧縮を提供でき得る。JPEG (Joint Photographic Experts Group) などの可変レート圧縮スキームは、非常に複雑であることと実装コストのために、オンザフライで展開することが望ましい場合には、通常、テクスチャ圧縮には使用されない。従って、この技術分野では、可変レート圧縮スキームの低コストのオンザフライの展開を可能にする方法及びメカニズムに対するニーズがある。

【0008】

上記を鑑みて、圧縮されたテクスチャのリアルタイムの展開を行うための改良された処理ユニット、方法及び媒体が望まれている。

【発明の概要】

【課題を解決するための手段】

【0009】

テクスチャデータを展開するための処理装置、方法及び媒体の様々な実施形態が考えられている。ある実施形態では、複数のシェーダのうち第1のシェーダは、ディスプレイデバイス又は更なる処理に用いられるデータを生成するために、テクスチャのブロックを必要とし得る。この第1のシェーダは、テクスチャの非圧縮バージョン内のブロックの仮想アドレスを計算し、ブロックに対する要求と合わせて仮想アドレスをキャッシュメモリに伝えるように構成されてもよい。複数のシェーダのうち第2のシェーダは、ブロックの非圧縮バージョンがキャッシュに保存されていないという判定に応じて、展開シェーダとして開始し、そのブロックの非圧縮バージョンの仮想アドレスが展開シェーダに渡されてもよい。また、ブロックの非圧縮バージョンがキャッシュに保存されていないという判定に応じて、要求されたブロックに対してキャッシュラインが割り当てられてもよい。

【0010】

第2のシェーダは、ブロックの圧縮バージョンをキャッシュから受け取るように構成されてもよい。キャッシュは、テクスチャの非圧縮バージョンの仮想アドレス空間を、テクスチャの圧縮バージョンのアドレス空間にマッピングするテーブルを利用するように構成されてもよい。キャッシュ及び/又は第2のシェーダは、ブロックの圧縮バージョンの場所とサイズとを、テーブルから決定するように構成されてもよい。このテーブルは、テクスチャの各ブロックの圧縮バージョンのDC係数値などの追加情報を含んでもよい。

【0011】

第2のシェーダは、キャッシュからブロックの圧縮バージョンを受け取ると、ブロックの圧縮バージョンを展開し、ブロックの展開バージョンをキャッシュに書き込むように構成されてもよい。第1のシェーダは、ブロックの展開バージョンがキャッシュに書き込まれた後に、ブロックの展開バージョンをキャッシュから受け取るように構成されてもよい。次いで、第1のシェーダは、ブロックの展開バージョンを、表示用にレンダリングされた表面に適用するように処理するように構成されてもよい。

【0012】

本明細書で説明するアプローチの以下の詳細な説明によれば、上記並びに他の機能及び

10

20

30

40

50

特徴が当業者にとって明らかになるであろう。

【0013】

以下の添付の書面と合わせて以下の説明を参照することで、システムと、方法と、機構との上記及び更なる利点の理解が深まるであろう。

【図面の簡単な説明】

【0014】

【図1】本発明のコンピュータグラフィックスシステムの一実施形態を示す図である。

【図2】本発明の1つ以上の実施形態によるGPUのブロック図である。

【図3】本発明のグラフィックスプロセッシングシステムの一実施形態を示すブロック図である。

10

【図4A】本発明のデータキャッシュの一実施形態を示すブロック図である。

【図4B】本発明の1つ以上の実施形態によるブロックマッピングテーブルを示す図である。

【図5】本発明の8×8ブロックのテクセルに対する仮想アドレス空間の一実施形態を示す図である。

【図6】本発明のデータの一部の一実施形態を示すブロック図である。

【図7】本発明のテクスチャの圧縮ブロックを展開する方法の一実施形態を示す、一般化されたフロー図である。

【0015】

以下の説明では、本明細書で示す方法及び機構を完全に理解してもらうために、多くの特定の詳細が規定されている。しかし、当業者は、こうした特定の詳細なしに様々な実施形態が実施可能であることを理解するであろう。いくつかの事例では、本明細書で説明するアプローチを曖昧にしないように、良く知られている構造、構成要素、信号、コンピュータプログラムの命令及び技術については詳細に説明していない。図を単純且つ明確にするために、図示されている要素は必ずしも縮尺に合わせて記載されていない。例えば、いくつかの要素の寸法は、他の要素と比較して拡張され得る。

20

【0016】

本明細書には、「一実施形態(“one embodiment”)」又は「ある実施形態(“an embodiment”)」という言葉が含まれる。「一実施形態では(“in one embodiment”)」又は「ある実施形態では(“in an embodiment”)」という表現は、必ずしも同じ実施形態について言及しているわけではない。特定の機能、構造又は特徴は、本開示と一貫性のある適切な方式で組み合わせられてもよい。

30

【0017】

用語について。以下の段落では、本開示(添付の請求項を含む)における用語の定義及び/又は文脈を示す。

【0018】

「備える、含む(“Comprising”)」。この用語は非限定的である。添付の請求項で使用されているように、この用語は、構造又はステップを追加することを排除しない。次のような請求項、「複数のシェーダを含むGPU(“A GPU comprising a plurality of shaders”)」を考えてみる。この請求項では、GPUが構成要素(例、テクスチャ・ユニット、入出力回路など)を追加で含むことを排除しない。

40

【0019】

「構成されている(“Configured To”)」。様々なユニット、回路又はその他の構成要素が、単一又は複数のタスクを行うように「構成されている(“configured to”)」ものとして説明又は特許請求される。こうした文脈では、「構成されている(“configured to”)」は、ユニット/回路/構成要素が、こうした単一又は複数のタスクをオペレーション中に実行する構造(例えば、回路)を含むことを示すことによって、構造を意味するために使用される。このように、ユニット/

50

回路／構成要素は、指定されたユニット／回路／構成要素がオペレーション中ではない場合（例えば、オン状態ではない）であっても、タスクを行うように構成されているとすることができる。「構成されている（“configured to”）」という表現と合わせて使用可能なユニット／回路／構成要素には、例えば、回路などのハードウェア、オペレーションを実行するために実行可能なプログラム命令を保存するメモリが含まれる。ユニット／回路／構成要素が１つ以上のタスクを行うように構成されている（“configured to”）と言及することは、そのユニット／回路／構成要素に対して３５

U.S.C. セクション 114、第６項を行使することを明示的に意図していない。さらに、「構成されている（“configured to”）」には、ソフトウェア及び／又はファームウェア（例えば、ソフトウェアを実行するＦＰＧＡまたは汎用プロセッサ）によって操作され、対象のタスクを行うことができるような方法で動作する一般的な構造（例えば、汎用回路）を含み得る。「構成されている（“configured to”）」は、１つ以上のタスクを実行又は遂行するために適合された素子（例えば、集積回路）を製造する製造工程（例えば、半導体製造施設）を適合させることも含み得る。

【００２０】

「第１の（“First”）」、「第２の（“Second”）」など。本明細書で使用されているように、こうした表現は、後に続く名詞に対するラベルとして使用され、いかなるタイプの順序（例えば、空間的、時間的、論理的など）を示唆するものではない。例えば、８つの処理要素又はコアを持つプロセッサでは、第１（“first”）及び第２（“second”）の処理要素は、８つの処理要素のうち任意の２つの処理要素を言及するために使用可能である。つまり、「第１」及び「第２」の処理要素は、論理処理要素の０及び１に制限されるものではない。

【００２１】

図１を参照すると、コンピュータグラフィックスシステムの一実施形態のブロック図が示されている。コンピュータグラフィックスシステム１００は、コンピューティングシステム１０２とディスプレイデバイス１１４とを含む。コンピューティングシステム１０２は、グラフィックスデータを処理するためのＧＰＵ（グラフィックスプロセッシングユニット）１０４を含む。いくつかの実施形態では、ＧＰＵ１０４は、コンピューティングシステム１０２内のグラフィックスカードに設けられてもよい。ＧＰＵ１０４は、ディスプレイデバイス１１４に表示するためのフレームの各ピクセルに対して色及び輝度の値を生成する、グラフィックスデータを処理してもよい。ＧＰＵ１０４は、ピクセル操作を行うための１つ以上の処理コア及び／又はシェーダのアレイを含んでもよい。

【００２２】

コンピューティングシステム１０２は、ＣＰＵ（図示せず）で実行可能なソフトウェアプログラムアプリケーション１０８と、ＡＰＩ（アプリケーションプログラミングインタフェース）１１０と、ドライバ１１２とを含んでもよい。ＡＰＩ１１０は、ＯｐｅｎＧＬ（登録商標）又はＤｉｒｅｃｔＸ（登録商標）などの業界標準の仕様に準拠してもよい。ＡＰＩ１１０は、ドライバ１１２と通信してもよい。ドライバ１１２は、ＡＰＩ１１０から受信した標準コードを、ＧＰＵ１０４が理解するネイティブ命令フォーマットに変換してもよい。ＧＰＵ１０４は、ドライバ１１２から受信した命令を実行してもよい。

【００２３】

テクスチャは、システムメモリ（図示せず）、又はコンピューティングシステム１０２の他の記憶デバイスからＧＰＵ１０４に転送されてもよい。一実施形態では、ＪＰＥＧ圧縮を用いて圧縮されてもよい。他の実施形態では、テクスチャを圧縮するために、他のタイプの可変レート圧縮を使用してもよい。本明細書の残りの部分では、様々な実施形態を説明するために、ＪＰＥＧタイプのエンコーディング例を使用する。しかし、これは例示目的に限ったものであり、本明細書で説明する方法及び機構と合わせて他のタイプの可変レート圧縮を使用してもよい。

【００２４】

ドライバ１１２は、タイリングプロセスの一部として、圧縮されたテクスチャを再フォ

10

20

30

40

50

ーマットしてもよい。この再フォーマットは、ＪＰＥＧ圧縮のテクスチャを、ハードウェア内部のＪＰＥＧ形式にトランスコーディングすることを必要とする場合がある。他の実施形態では、ＪＰＥＧ圧縮のテクスチャは、他の形式にトランスコーディングされてもよい。ハードウェア内部のＪＰＥＧ形式には、展開プロセスを行い易くするための追加情報を含んでもよい。例えば、ハードウェア内部のＪＰＥＧ形式には、ＪＰＥＧ圧縮のテクスチャの様々なブロックの場所及びサイズについての情報を有するテーブルが含まれ得る。このテーブルは、ＪＰＥＧ圧縮のテクスチャの各 8×8 ブロックのＤＣ係数についての情報を含んでもよい。さらに、このテーブルは、ハフマン符号、量子化テーブル及び圧縮テクスチャの展開を行い易くするための他の情報を含んでもよい。ドライバ１１２は、コン

10

【００２５】

コンピューティングシステム１０２は、通常、ＣＰＵ、バス、メモリ、周辺装置など図１に示されていない様々な他のデバイス／構成要素を有する。例えば、コンピューティングシステム１０２は、ディスプレイデバイス１１４以外に、キーボード、プリンタ及びマウスなどの他のデバイスに接続され得るＩ／Ｏインターフェースを含み得る。いくつかの実施形態では、コンピューティングシステム１０２は、複数のＧＰＵを含んでもよい。

【００２６】

他の実施形態では、ＧＰＵ１０４などのプロセッサは、ソフトウェアで定義されてもよい。このソフトウェア命令は、コンピュータで読み取り可能な記憶媒体に保存されてもよく、コンピューティングデバイス上で実行されると、プロセッサを定義し得る。さらなる実施形態では、プロセッサはＧＰＵ、ＣＰＵ、ビデオ・プロセッシング・ユニット（ＶＰＵ）、コプロセッサ及び／又はテクスチャデータを処理するように構成された他のタイプのプロセッサを含んでもよい。様々な実施形態では、ＧＰＵとＣＰＵとは別個の集積回路デバイス／パッケージであってもよい。様々な実施形態では、ＧＰＵとＣＰＵとは、一つの集積回路又はパッケージに含まれてもよい。

20

【００２７】

図２を参照すると、ＧＰＵ２００の一実施形態のブロック図が示されている。ＧＰＵ２００は、グラフィックス処理関連のタスク（例えば、頂点シェーダ、ジオメトリシェーダ、ピクセルシェーダなどを使用するタスク）及び一般的なコンピューティングタスク（例えば、数学的アルゴリズム、物理シミュレーションなど）を行うために利用され得る。図の例では、ＧＰＵ２００は、シェーダプロセッサアレイ２１０と、コマンドプロセッサ２１２と、テクスチャメモリ２２０と、ダイレクトメモリアクセス（ＤＭＡ）に対応するように構成され得るメモリコントローラ２２２とを含む。図２のＧＰＵ２００の実施形態は、例示目的に限ったものであり、当業者は多くの代替実施形態が可能であることを理解するであろう。こうした代替実施形態のすべてが考慮される。また、ＧＰＵ２００には、図２に示されていない他の多くの構成要素を含み得ることに留意されたい。

30

【００２８】

図示されている実施形態では、シェーダプロセッサアレイ２１０は、並列で実行可能な複数のプロセッシングユニットを含んでもよい。コマンドプロセッサ２１２は、コマンドを発行し、シェーダプロセッサアレイ２１０の個々のシェーダプロセッサに対してタスクを割り当ててもよい。いくつかの実施形態では、コマンドプロセッサ２１２は、受信した作業負荷をスレッドに分割し、そのスレッドをシェーダプロセッサアレイのプロセッシングユニット内で分散するように構成されたディスパッチプロセッサ（図示せず）を含んでもよい。シェーダプロセッサアレイ２１０は、テクスチャデータの処理と、３次元テクスチャオブジェクトを２次元画像に変換するためのレンダリングアルゴリズムの実行とを含む様々なタイプの機能を行うように構成されてもよい。上記のように、シェーダプロセッサアレイ２１０は、複数のシェーダプロセッサを含んでもよく、複数のシェーダプロセッサは、頂点及び他のテクスチャデータに対して、幅広い数学的演算及び論理演算を用いる

40

50

アルゴリズムを実行してもよい。

【0029】

いくつかの実施形態では、GPU 200は、データを一時的に保存するために、オンチップのメモリ又はオフチップのメモリを利用するように構成されてもよい。こうしたメモリは、本明細書では「キャッシュ(“caches”)」と呼ぶことがあるが、こうした用語を使用することは、こうしたメモリに対して必ずしも特定の機構、構造又はポリシーを必要とするわけではないことに留意されたい。例えば、こうしたメモリは、セット連想機構及び置換ポリシーなどの、CPU(中央演算処理装置)のキャッシュに関連する機構及びポリシーを利用することがあるが、任意の望ましい機構及び/又は記憶ポリシーを利用してもよい。様々な実施形態では、テクスチャデータを保存するために、テクスチャメモリ 220が使用される。こうした実施形態では、テクスチャメモリ 220は、頻繁に使用されるテクスチャデータなどの一部のテクスチャデータに対して、テクスチャデータがシステムメモリ 226やローカルメモリ 230のみに保存される場合に可能な速度よりも高速なアクセスを提供してもよい。システムメモリ 226は、GPU 200とCPU(中央演算処理装置、図示せず)の両方によってアクセス可能なメモリを表す場合があり、一方、ローカルメモリは、GPU 200によってのみ直接アクセス可能なメモリを表し得る。様々な実施形態では、テクスチャメモリ 220は、キャッシュ分野では一般的に知られているように、階層構造の複数のレベルを含み得る。テクスチャキャッシュシステム 220に含まれるこうしたキャッシュレベルの数は、実施形態によって異なる。テクスチャメモリ 220は、スタティックメモリ(例、SRAM)、ダイナミックメモリ(例、DRAM)を用いたスタック型メモリ又は他の方法を使用して実装してもよい。テクスチャメモリ 220は、キャッシングロジックも含んでもよい。このキャッシングロジックは、データをテクスチャメモリ 220にキャッシュし、システムメモリ 226に対するキャッシュシステム 220の相対的なレーテンシ及び/又は帯域幅を考慮するキャッシュ管理ポリシーを実行するように構成されてもよい。

【0030】

GPU 200は、メモリコントローラ 222を含んでもよい。メモリコントローラ 222は、システムメモリ 226及びローカルメモリ 230に接続されてもよい。メモリコントローラ 222は、システムメモリ 226内の圧縮テクスチャ 228などのデータにアクセスしてもよい。圧縮テクスチャ 228は、JPEGなどの様々な可変レート圧縮技術で圧縮し得る複数のテクスチャを含んでもよい。圧縮テクスチャ 228又は圧縮テクスチャ 228内の各テクスチャの一部は、最初に展開せずに、テクスチャメモリ 220及びGPU 200のシェーダプロセッサアレイ 210に(メモリコントローラ経由で)転送されてもよい。ホストドライバ 240は、コマンドとデータとを、システムメモリ 226を介してGPU 200に転送してもよい。ローカルメモリ 230は、頂点データと、GPUが使用する他のデータとを保存するために利用されてもよく、GPU 200は、フレームデータをローカルメモリ 230に書き込んでもよい。

【0031】

図3を参照すると、グラフィックスプロセッシングシステムの一実施形態のブロック図が示されている。グラフィックスプロセッシングシステム 300には、シェーダコントローラ 310を含む場合があり、シェーダコントローラ 310は、特定のグラフィックス処理タスクを、シェーダアレイ 320内の各シェーダコンピューティングユニットに割り当ててもよい。シェーダコントローラ 310は、グラフィックス処理タスク及び一般のコンピューティングタスクに対して前処理を行う場合があり、こうしたタスクをシェーダアレイ 320に発行する。シェーダコントローラ 310は、シェーダのどの処理要素が新たな作業負荷の処理に利用可能であるかを特定し、シェーダアレイ 320の利用可能な処理要素に新たな作業負荷を送信してもよい。シェーダコントローラ 310は、どの作業負荷がシェーダアレイの異なる処理要素によって処理されているかを追跡してもよく、複数のスレッドを並列で実行できるようにする。

【0032】

10

20

30

40

50

シェーダアレイ 320 は、テクスチャコンシューミングシェーダ 321 及び展開シェーダ 322 を含んでもよく、これらはシェーダアレイ 320 に含まれ得る、任意の数及びタイプのシェーダプロセッサを表す。様々な実施形態では、シェーダアレイ 320 は、テクスチャデータを手続き的に生成するように構成され得る追加のシェーダプロセッサを含んでもよい。一般的に、手続き的なテクスチャ生成とは、テクスチャをアルゴリズム的に生成するプロセスを意味することが多い。様々な実施形態では、この手続き的なテクスチャ生成は、事前というよりはむしろ動的に行われる。シェーダアレイ 320 は、他のタスクの中で、特にテクスチャマッピングと、表示デバイス用に画像データを生成するためとに使用し得る。こうしたオペレーションの一部として、テクスチャコンシューミングシェーダ 321 は、テクスチャ要求をテクスチャフィルタ 330 に対して発行する。テクスチャ要求は、テクスチャの 1 つ以上の部分（例えば、ブロック、テクセル）に対するものであってもよい。テクスチャフィルタ 330 は、要求されたテクスチャに対する仮想アドレスを生成し、要求と合わせてその仮想アドレスをキャッシュ 340 に伝えてもよい。キャッシュ 340 は、ピクセルに関連したテクセルデータ形式でテクスチャを保存してもよい。いくつかのテクスチャは圧縮され、いくつかのテクスチャは復元され得る。

【0033】

テクスチャフィルタ 330 から仮想アドレスを受信した後、キャッシュ 340 は、要求されたテクスチャがキャッシュ 340 に保存されているか否かを調べるために、既知の仮想アドレス範囲の全てに対してアドレスチェックを行い得る。要求されたテクスチャの非圧縮バージョンがキャッシュ 340 に保存されている場合には、キャッシュ 340 は、そのテクスチャの非圧縮バージョンをテクスチャフィルタ 330 に戻してもよい。テクスチャの非圧縮バージョンがキャッシュ 340 に保存されていない場合には、この試みられた要求はキャッシュミスという結果になることがある。キャッシュミスに応じて、展開シェーダ 322 は、そのテクスチャの圧縮バージョンを展開する目的で開始してもよい。様々な実施形態では、シェーダアレイ 320 は、キャッシュ 340 から要求を受信してもよいし、展開シェーダを開始してもよい。また、キャッシュミスに応じて、テクスチャコンシューミングシェーダ 321 は、テクスチャの仮想アドレスを展開シェーダ 322 に渡してもよい。展開シェーダプログラムに対するリソースは、シェーダ開始のレーテンシを減らし、リソース管理を単純化するために展開シェーダ 322 に事前に割り当ててもよい。この要求は、要求されているブロックの仮想アドレスに基づいて、シェーダアレイ 320 の特定のシェーダプロセッサにルーティングされてもよい。

【0034】

キャッシュ 340 は、テクスチャの圧縮バージョンについて照会される場合があり、そのテクスチャの圧縮バージョンがキャッシュ 340 に保存されていれば、そのテクスチャの圧縮バージョンは、展開シェーダ 322 に戻されてもよい。テクスチャの圧縮バージョンがキャッシュ 340 に保存されていない場合には、そのテクスチャの圧縮バージョンは、システムメモリ又は他の場所から取り出されてもよい。展開シェーダ 322 は、展開オペレーションを行い易くするために、追加のテーブル、テクスチャ及び / 又は定数を受信してもよい。展開シェーダ 322 は、要求されたテクスチャを展開するために必要ないくつかの追加の圧縮データを展開してもよい。JPEG 圧縮のテクスチャの場合には、テクスチャは、オリジナルのコードから新しい符号化スキームにトランスコードされ、この新しい符号化スキームは、展開の効率をさらに高めるように設計されてもよい。展開シェーダ 322 が、テクスチャの圧縮バージョンを受信して展開した後に、テクスチャコンシューミングシェーダ 321 は、適切なレンダリングの計算のために、そのテクスチャの展開バージョンを利用してもよい。このプロセスは、複数のテクスチャ及び / 又はテクスチャの一部に対して継続してもよい。他の実施形態では、テクスチャフィルタ 330 によって行われると記載されている機能は、シェーダアレイ 320 によって行われてもよく、シェーダアレイ 320 はキャッシュ 340 に直接接続されてもよい。

【0035】

キャッシュ 340 は、テーブルを利用して、テクスチャの圧縮バージョンに対する所定

10

20

30

40

50

の仮想アドレスマップがキャッシュ 340 内で保存されているアドレスを決定してもよい。様々な実施形態では、このテーブル（またはその一部）は、キャッシュ 340 又は他の場所に保存されてもよい。ある実施形態では、このテーブルは、仮想アドレスを、テクスチャの圧縮バージョンの他のアドレスにマッピングしてもよい。仮想アドレスがマッピングされたアドレスは、それ自体が仮想アドレスである場合があるし、仮想アドレスでない場合もある。利用されているアドレッシングスキームのタイプについて多くのオプションが可能であり、考えられる。このテーブルは、テクスチャの圧縮バージョンの各ブロックに対するオフセットを保存する場合があり、このオフセットは、テクスチャの圧縮バージョンの始まりからブロックまでの場所を与える。様々な実施形態では、このテーブルは、1つ以上の圧縮テクスチャのブロックに対するランダムアクセスを行い易くし得る。キャッシュ 340 のキャッシュロジックは、ブロックの圧縮バージョンに対する要求に応じて、あるブロックのアドレスを決定し得る。キャッシュロジックは、このテーブルを使用して、キャッシュのページ又はフェッチユニット内で望ましいブロックが保存されているオフセットを決定する。シェーダアレイ 320 の複数のシェーダも、このテーブルを使用して、テクスチャの要求されたブロックのオフセットを決定し得る。様々な実施形態では、キャッシュ 340 は、複数のテクスチャのマッピング情報を有する複数のテーブルを使用してもよい。

10

【0036】

テクスチャデータの処理後、シェーダアレイ 320 は、その画像データをレンダリングユニット 350 に伝えてもよい。レンダリングユニット 350 は、画像フレームの各ピクセルに対する固有の色属性を定義する特定の数値を割り当ててもよい。この数値は、フレームバッファ 360 に渡され、ディスプレイデバイス 370 にレンダリングされる場合などの適切な場合に使用するためにフレームバッファ 360 に保存される場合がある。

20

【0037】

次に続くオペレーションでは、テクスチャコンシューミングシェーダ 321 は、展開シェーダの機能を行うように構成されてもよく、展開シェーダ 322 は、テクスチャコンシューミングシェーダの機能を行うように構成されてもよい。シェーダアレイ 320 の各シェーダプロセッサは、現在のオペレーションの要件に応じて、様々な機能を行うように構成されてもよい。

【0038】

様々な実施形態では、十分に活用されていないシェーダに展開タスクを割り当てるために、ロードバランシングを利用してもよい。また、展開シェーダが多くの演算ユニットで開始できるように、一部の空間を多くの演算ユニットのために予約してもよい。さらに、複数の展開要求が、単一の SIMD（単一命令多重データ処理）に入れられてもよい。SIMD ベクトルは、1つのベクトルで複数のブロックの展開を行い易くする場合がある。ある実施形態では、4 レーンにつき 1 ブロックで、1つのベクトルで 16 ブロックを展開し得る。

30

【0039】

様々な実施形態では、グラフィックスプロセッシングシステム 300 は、テクスチャデータのオンザフライの手続き的生成を可能にし得る。第 1 のシェーダが、オンザフライのテクスチャデータを生成してもよく、第 2 のシェーダが、生成されたテクスチャデータをレンダリングオペレーションに利用してもよい。展開シェーダは、圧縮データにアクセスし、別のシェーダは、1つ以上のテーブルなどの追加のデータの展開に利用されてもよい。圧縮データの一部は、様々な圧縮技術を使用して圧縮されてもよい。様々な実施形態では、展開シェーダは、キャッシュからデータを要求してもよく、キャッシュミスに応じて、テクスチャデータを手続き的に生成するために別のシェーダが開始されてもよい。

40

【0040】

図 4A を参照すると、データキャッシュの一実施形態のブロック図が示されている。キャッシュ 410 は、キャッシュ 410 に保存可能な任意の数のテクスチャの一部を表す、テクスチャ 420 の一部とテクスチャ 430 の一部とを含み得る。テクスチャ 420 とテ

50

クスチャ４３０とは、圧縮されたテクスチャであってよい一方で、キャッシュ４１０に保存される複数のテクスチャは、圧縮テクスチャ及び非圧縮テクスチャが混在したものであってもよい。テクスチャ４２０は、ブロック４２２とブロック４２３とを含んでもよく、これらはテクスチャ４２０の任意の数のブロックを表す。テクスチャ４２０は、テクスチャ４２０の仮想アドレス空間を圧縮テクスチャ４２０のアドレス空間にマッピングするテーブル４２１を含んでもよい。テクスチャ４３０は、テクスチャ４２０と同様に編成されてもよい。他の実施形態では、テーブル４２１は、テクスチャ４２０とは別に保存されてもよい。

【００４１】

テクスチャコンシューミングシェーダが、キャッシュ４１０からテクスチャのブロックを要求し、その要求がキャッシュミスという結果になる場合には、キャッシュ４１０は、要求されたブロックに対してキャッシュラインを割り当ててもよい。キャッシュ４１０は、割り当てられたキャッシュラインのアドレスを、展開シェーダに伝えてもよい。展開シェーダが要求されたブロックに対応する圧縮ブロックの展開を完了すると、展開シェーダは、展開されたブロックをキャッシュライン４４０に書き込むように構成されてもよい。その上、展開シェーダは、展開されたブロックをキャッシュ４１０内の様々な場所に書き込んでもよい。展開シェーダが、展開したブロックをキャッシュライン４４０に書き込むことに応じて、テクスチャコンシューミングシェーダは、キャッシュ４１０から展開ブロックをフェッチするように構成されてもよい。オンザフライの圧縮ブロックの展開の結果としてのより大きなレーテンシに対応するために、対応するレーテンシ補償待ち行列を延長する必要があると得る。

【００４２】

ブロックの展開バージョンがキャッシュライン４４０に書き込まれると、キャッシュ４１０は、ブロックの圧縮バージョンとブロックの展開バージョンとを保存してもよい。様々な実施形態では、キャッシュ４１０は、両方のバージョンがキャッシュ４１０に保存されているという判定に応じて、ブロックの１つ以上のバージョンを破棄する保持ポリシーを実行してもよい。ある実施形態では、ブロックの展開バージョンは、テクスチャコンシューミングシェーダによってフェッチされた後に破棄されてもよい。他の実施形態では、ブロックの展開バージョンは、キャッシュ４１０に書き込まれた後に破棄されてもよい。さらなる実施形態では、ブロックの圧縮バージョンと展開バージョンの両方が、キャッシュ４１０で長期間維持されてもよい。

【００４３】

テクスチャのブロックの非圧縮バージョンに対する要求に応じて、キャッシュ４１０は、非圧縮バージョンがキャッシュ４１０に保存されていないと判定してもよい。様々な実施形態では、こうした判定に応じて、キャッシュ４１０は、ブロックの圧縮バージョンを自動的に検索し得る。ブロックの圧縮バージョンがキャッシュ４１０に保存されている場合には、キャッシュ４１０はシェーダに通知してもよいし、他の処理ユニット及び／又はキャッシュ４１０は、ブロックの圧縮バージョンをシェーダ又は他の処理ユニットに伝えてもよい。

【００４４】

いくつかの実施形態では、非圧縮ブロックに対する要求についてのキャッシュミスに応じて、別個のソフトウェアスレッドが開始され、そのスレッドが展開シェーダを開始し得る。テクスチャコンシューミングシェーダは、ブロックの仮想アドレスを展開シェーダに伝え得る。様々な実施形態では、シェーダが展開タスクを終了すると、展開シェーダは、非圧縮ブロックをキャッシュに伝え得る。他の実施形態では、展開シェーダが展開オペレーションを終了すると、展開シェーダは、そのシェーダ出力をテクスチャコンシューミングシェーダに伝え得る。

【００４５】

図４Ｂを参照すると、ブロックマッピングテーブルの一実施形態のブロック図が示されている。テーブル４２１は、（図４Ａの）テクスチャ４２０の複数のブロックに対するマ

10

20

30

40

50

ッピング情報を保存してもよい。様々な実施形態では、テーブル 4 2 1 は、図 4 B に図示されているものに加えて、他のタイプの情報を用いて様々な方法で編成されてもよい。例えば、ある実施形態では、テーブル 4 2 1 は、テクスチャ 4 2 0 の各ブロックに対する DC 係数値を含んでもよい。

【 0 0 4 6 】

テーブル 4 2 1 は、テクスチャ 4 2 0 の仮想アドレス空間を、(図 4 A の) 圧縮テクスチャ 4 2 0 の物理アドレス空間にマッピングしてもよい。展開シェーダ (図示せず) は、キャッシュ 4 1 0 から 1 つ以上のテクスチャ 4 2 0 のブロックをフェッチしてもよいし、受け取ってもよい。展開シェーダは、テーブル 4 2 1 から圧縮ブロックの場所及びサイズを決定してもよい。圧縮ブロックのサイズは、2 つの隣接するブロック間の開始物理アドレスの差を計算することで求められてもよい。他の実施形態では、ブロックのサイズ及び / 又は場所の情報を示すために、追加データが提供されてもよい。さらに、展開シェーダは、テーブル 4 2 1 から、各ブロックの DC 係数値などの追加情報を入手してもよい。

【 0 0 4 7 】

いくつかの実施形態では、テクスチャは、スーパーブロックに従って編成されてもよい。スーパーブロックは、16 個の 8×8 ブロックのセットであってもよく、これは 32×32 ピクセルのタイルであり、合計 1024 ピクセルである。テクスチャのインデックステーブルは、各スーパーブロックに対するテーブルエントリを含んでもよく、各テーブルエントリは、各スーパーブロックの開始アドレスを与えてもよい。ある実施形態では、このアドレスは、テクスチャ内のスーパーブロックの場所であってもよい。他の実施形態では、このアドレスは、テクスチャの開始からのオフセットであってもよい。各エントリは、スーパーブロックに属する最初の 8×8 のブロックの 4 ビットのインデックスを含んでもよい。いくつかの実施形態では、スーパーブロックは、キャッシュの 2 キロビット (Kb) の境界に整列されていない場合がある。各エントリは、16 ビットのマスクを含んでもよい。この 16 ビットのマスクには、そのブロックが次の 2 Kb のワードで始まっているか否かを示す、1 ブロックあたり 1 ビットを含んでもよい。

【 0 0 4 8 】

いくつかの実施形態では、展開シェーダは、ルックアップ目的でインデックステーブルのエントリ数を計算するために、 8×8 のブロックの仮想アドレスを、 32×32 のスーパーブロックの仮想アドレスに変換してもよい。展開シェーダは、スーパーブロックに対応するインデックステーブルのエントリをルックアップしてもよい。インデックステーブルは、他のテクスチャと同じようにシェーダによって処理されてもよい。インデックステーブルのエントリは、キャッシュされ、処理されてもよい。

【 0 0 4 9 】

シェーダは、各インデックステーブルのエントリから、仮想アドレスとなり得る基底アドレスを取得してもよい。この基底アドレスは、圧縮スーパーブロックの最初のフェッチユニットであってもよい。シェーダは、展開を必要とする要求されたブロックを含むフェッチユニットのオフセットを取得してもよい。シェーダは、ブロックが圧縮されているか否かを、ブロックのアドレスに基づき計算してもよい。あるアドレス範囲は、非圧縮ブロックの仮想アドレスに対応してもよく、他のアドレス範囲は、圧縮ブロックの物理アドレスに対応してもよい。シェーダは、この異なるアドレス範囲を区別し得る。

【 0 0 5 0 】

図 5 を参照すると、テクセルの 8×8 ブロックに対する仮想アドレス空間の一実施形態を示すブロック図が示されている。各テクセルは、仮想アドレス空間 570 内の固有のアドレスにマッピングされ得る。ブロック 500 の 8×8 の 64 テクセルの全てに対して、テクセル 1 はアドレス 501 に、テクセル 2 はアドレス 502 に、などのようにマッピングされ得る。ブロック 500 は、圧縮テクスチャ内のブロックであってもよく、仮想アドレス空間 570 は、圧縮テクスチャのブロック 500 に対して割り当てられてもよい。テクスチャは、ブロック 500 に加えて複数のブロックを含んでもよい。仮想アドレス空間 570 は、テクスチャの複数のブロックの各テクセルに対して固有のアドレスを含んでもよ

10

20

30

40

50

い。

【 0 0 5 1 】

例示目的で、非圧縮テクセルは、32ビット値（8ビット値の4セット）であると想定する。本明細書で説明する方法及び機構では、他のサイズの非圧縮テクセルを利用してもよい。例えば、24ビット値の非圧縮テクセルは、同様に処理されてもよい。様々な実施形態では、テクスチャコンシューミングシェーダは、個々のテクセルに対する要求を生成してもよい。最初に、シェーダは、テクセルの仮想アドレスを計算してもよい。次に、テクセルに対応する仮想アドレスに対してキャッシュが照会される。

【 0 0 5 2 】

図6を参照すると、圧縮データの一実施形態のブロック図が示されている。データ部分605は、圧縮データのフェッチユニットであってもよく、データ部分605のサイズは、非圧縮ブロックのサイズに基づいていてもよい。ある実施形態では、フェッチユニットは2Kbのサイズであってもよい。他の実施形態では、フェッチユニットは任意の様々なサイズであってもよい。複数の圧縮ブロックをフェッチユニットに入れてもよい。ある実施形態では、フェッチユニットに入れることができる最大ブロック数は16であると考えられ得る。他の実施形態では、フェッチユニットに他の数のブロックを入れてもよい。ある種のキャッシュアクセススキームでは、ブロックのデータは、フェッチユニットの境界を超えないと考えてもよい。

【 0 0 5 3 】

ブロックは、JPEGなどの圧縮フォーマットの最小復号可能単位であり得る。JPEGでは、ブロックは8×8のピクセルのタイル（64ピクセル）である。テクスチャが圧縮され、シェーダによって要求されたテクスチャのブロックを展開する必要がある場合には、そのブロックに対してキャッシュラインがキャッシュ内で割り当てられてもよい。ある実施形態では、キャッシュラインのサイズは、非圧縮ブロック（32ビット×64＝2Kb）全体を保存するために、2Kbであってもよい。他の実施形態では、キャッシュラインのサイズは任意の様々なサイズであってもよい。

【 0 0 5 4 】

フェッチユニットが非圧縮ブロックを含む場合には、1つのブロックのみがそのフェッチユニットに適合し得る。圧縮ブロックを含むフェッチユニットでは、176ビットのヘッダを含み得る。このフェッチユニットは、16ブロックの容量があると考えてもよい。このヘッダは、フェッチユニット内の圧縮ブロックの場所を示すために、16の11ビットオフセット値を含み得る。このオフセットは、ブロックの開始ビット位置を参照する。他の実施形態では、可変数のオフセット・インジケータが、ヘッダに存在し得る。

【 0 0 5 5 】

図6に示すように、データ部分605は、ヘッダ610及びブロック611～626を含んでもよい。ブロック611～626は、圧縮テクスチャの16の異なるブロックであってもよい。ヘッダ610は、オフセット631～646を含んでもよい。各オフセットは、データ部分605内の対応するブロックの位置に対応する11ビットのオフセット値であってもよい。他の実施形態では、他のビットサイズのオフセット値を利用してもよい。オフセット631は、ブロック611の開始アドレスを表してもよく、オフセット632は、ブロック612の開始アドレスを表してもよい。いくつかの実施形態では、キャッシュからの不要なフェッチを減らすために、最後のブロックの最後のビットを示す追加のオフセットがあってもよい。

【 0 0 5 6 】

いくつかの実施形態では、テクスチャの8×8の圧縮ブロックを入れて、フェッチユニットの境界を越えてもよい。ブロックが2つのフェッチユニットを使用していることを示す、対応する情報をインデックステーブルに保存してもよく、展開シェーダは、フェッチユニットの境界を超えるブロックに対して1つのフェッチを生成する代わりに、2つのフェッチを生成してもよい。

【 0 0 5 7 】

10

20

30

40

50

図7を参照すると、テクスチャの圧縮ブロックを展開する方法の一実施形態が示されている。説明のために、本実施形態のステップは、順番に示されている。以下で説明する方法の様々な実施形態では、説明した1つ以上の要素は、同時に行われ、示された順序と異なる順序で行われ、又は完全に省略されてもよい。他の追加の要素が必要に応じて行われてもよい。

【0058】

方法700は、ブロック705で開始し、次にブロック710にて、複数のシェーダのうち第1のシェーダは、画像に対するレンダリングオペレーションの一部として、テクスチャのブロックに対する必要性を決定する。第1のシェーダは、テクスチャコンシューミングシェーダであってもよい。次に、第1のシェーダは、ブロックの仮想アドレスを計算してもよい(ブロック715)。第1のシェーダは、テクスチャの非圧縮バージョンに対応するテクスチャの非圧縮ビューを有することがあり、その仮想アドレスは、非圧縮ビュー内の要求されたブロックの位置に対応してもよい。ブロック715の後に、第1のシェーダは、キャッシュからブロックを要求し、その要求と合わせて仮想アドレスを伝えてもよい(ブロック720)。次に、キャッシュは、ブロックの非圧縮バージョンがキャッシュに保存されているか否かを判別してもよい(条件ブロック725)。ブロックの非圧縮バージョンがキャッシュに保存されている場合には、第1のシェーダは、キャッシュからブロックの非圧縮バージョンを受信し、そのブロックを処理してもよい(ブロック770)。

【0059】

ブロックの非圧縮バージョンがキャッシュに保存されていない場合には、複数のシェーダのうち第2のシェーダは、展開シェーダとして開始してもよい(ブロック730)。シェーダが開始するレーテンシを減らし、且つ、リソース管理を単純化するために、展開シェーダのリソースを、1つ以上のシェーダプロセッサに事前に割り当ててもよい。また、要求されたブロックの仮想アドレスは、第1のシェーダから第2のシェーダに渡されてもよい。次に、要求されたブロックに対してキャッシュラインを割り当ててもよい(ブロック735)。次に、キャッシュは、ブロックの圧縮バージョンがキャッシュに保存されているか否かを判別してもよい(条件ブロック740)。様々な実施形態では、ブロックの圧縮バージョンに対する第2のシェーダによる要求に応じて、キャッシュがこの判別を行ってもよい。他の実施形態では、ブロックの非圧縮バージョンがキャッシュに保存されていないという判別に応じて、自動的にこの判別を行ってもよい(条件ブロック725)。

【0060】

ブロックの圧縮バージョンがキャッシュに保存されている場合には(条件ブロック740)、キャッシュ及び/又は第2のシェーダは、ブロックの圧縮バージョンの場所及びサイズをテーブルから決定してもよい(ブロック750)。ブロックの圧縮バージョンがキャッシュに保存されていない場合には(条件ブロック740)、ブロックの圧縮バージョンをフェッチして(例、ローカル又はシステムのメモリから)、キャッシュに保存してもよい(ブロック745)。システムメモリからブロックの圧縮バージョンをフェッチすることは、圧縮されたテクスチャ全体又はそのテクスチャの一部をフェッチすることを必要とする。キャッシュは、テクスチャの非圧縮バージョンの仮想アドレス空間を、テクスチャの圧縮バージョンのアドレス空間にマッピングするテーブルを利用するように構成されてもよい。キャッシュ及び/又は第2のシェーダは、ブロックの場所及びサイズをテーブルから決定してもよい(ブロック750)。このテーブルは、テクスチャの各ブロックの圧縮バージョンのDC係数値などの追加情報も含んでもよい。ブロック750の後に、ブロックの圧縮バージョンは、キャッシュから第2のシェーダに伝えられてもよい(ブロック755)。

【0061】

他の実施形態では、ブロックの圧縮バージョンがキャッシュに存在しない場合には(条件ブロック740)、ステップ745, 750, 755は代替のステップと置き換えられてもよい。代替ステップでは、ブロックの圧縮バージョンをシステムメモリからフェッチ

し、直接第2のシェーダに提供してもよい。こうした代替ステップは、第2のシェーダが、キャッシュからブロックの圧縮バージョンを受け取るよりも効率的であり得る。さらなる実施形態では、ブロックの圧縮バージョンは、システムメモリからフェッチされ、キャッシュに書き込まれるとともに、直接第2のシェーダに提供され得る。

【0062】

第2のシェーダがブロックの圧縮バージョンを受け取った後に(ブロック755)、第2のシェーダは、ブロックの圧縮バージョンを展開してもよい(ブロック760)。次に、ブロックの展開バージョンをキャッシュに書き込んでもよい(ブロック765)。その後、第1のシェーダは、ブロックの展開バージョンをキャッシュから受け取り、そのブロックを現在の画像に対するレンダリングオペレーションの一部として処理してもよい(ブロック770)。ブロック770の後に、この方法はブロック775で終了してもよい。方法700は、複数のテクスチャからの複数のブロックに対して繰り返されてもよい。

【0063】

実施形態の例では、機能と要素とを特定の組み合わせで説明したが、各機能または要素は、実施形態の例の他の機能及び要素なしで単独で使用可能であり、又は他の機能及び要素が有り若しくは無しのような様々な組み合わせで利用可能である。本発明は、マシン、プロセッサ及び/又は不揮発性メモリデバイスと合わせて使用する、又は不揮発性デバイスによって任意の汎用コンピュータによって実行するためのマシンで読み取り可能な命令を有する非一時的なコンピュータ可読記憶媒体で有形に実施されたコンピュータプログラム又はファームウェアで実装されてもよい。このコンピュータ可読記憶媒体は、本明細書で説明した機能、方法及びオペレーションを可能にするために動作可能なプログラム命令を含み得る。適切なプロセッサとしては、例えば、汎用プロセッサ及び特殊用途のプロセッサの両方が含まれる。

【0064】

上記の実施形態は、Verilog又はVHDLなどのHDL(ハードウェア記述言語)を使用して設計されてもよい。HDL設計は、電子システムの挙動をモデル化してもよく、この設計は合成され、最終的にはハードウェアデバイスに組み立てられてもよい。さらに、このHDL設計をコンピュータ製品に保存し、ハードウェア製造に先立ちコンピュータシステムにロードされてもよい。

【0065】

本発明で使用され、又は本発明と合わせて使用され得るハードウェアの構成要素、プロセッサ又はマシンのタイプには、ASIC(特定用途向け集積回路)、FPGA(フィールドプログラマブルゲートアレイ)、マイクロプロセッサ又は任意の集積回路が含まれてもよい。こうしたプロセッサは、処理されたHDL(ハードウェア記述言語)命令(こうした命令は、コンピュータ可読媒体に保存可能である)の結果を用いて製造工程を構成することによって製造されてもよい。こうした処理の結果は、マスクワークであってもよく、次に半導体製造工程で、本明細書で説明した方法及び機構の一態様を実装するプロセッサを製造するために使用される。

【0066】

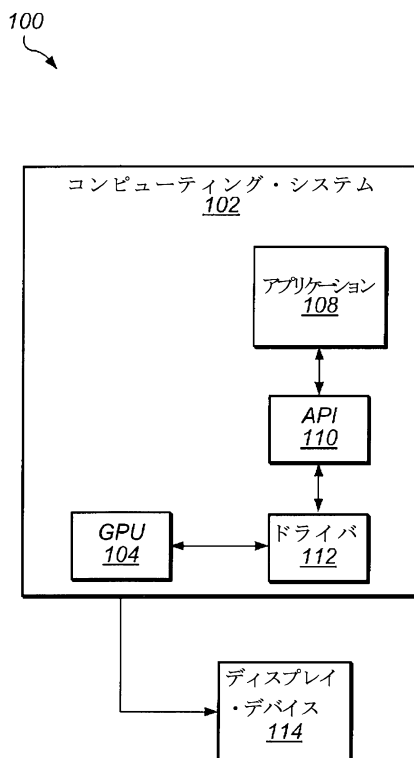
画像レンダリングの計算及びシェーダタスクを実施するために使用されるソフトウェア命令は、コンピュータで可読記憶媒体に保存されてもよい。コンピュータ可読記憶媒体は、マシン(例えば、コンピュータ)によって読み取り可能な形式(例えば、ソフトウェア、処理アプリケーション)で情報を保存するための任意の機構を含んでもよい。コンピュータ可読記憶媒体としては、以下に制限されるものではないが、磁気若しくは光学媒体(例えば、ディスク(固定若しくはリムーバブル))、テープ、CD-ROM、DVD-ROM、CD-R、CD-RW、DVD-R、DVD-RW、又は、Blu-Ray)、RAM(例えば、SDRAM(同期式動的RAM))、ダブルデータレート(DDR、DDR2、DDR3など)SDRAM、低電力DDR(LPDDR2など)SDRAM、RDRAM(Rambus DRAM)、SRAM(スタティックRAM)、ROM、USBインターフェースなどの周辺装置インターフェース経由でアクセス可能な不揮発メモリ(

例えば、フラッシュメモリ)、MEMS(微小電気機械システム)、並びにネットワーク及び/又は無線リンクなどの通信媒体経由でアクセス可能な記憶媒体が挙げられる。

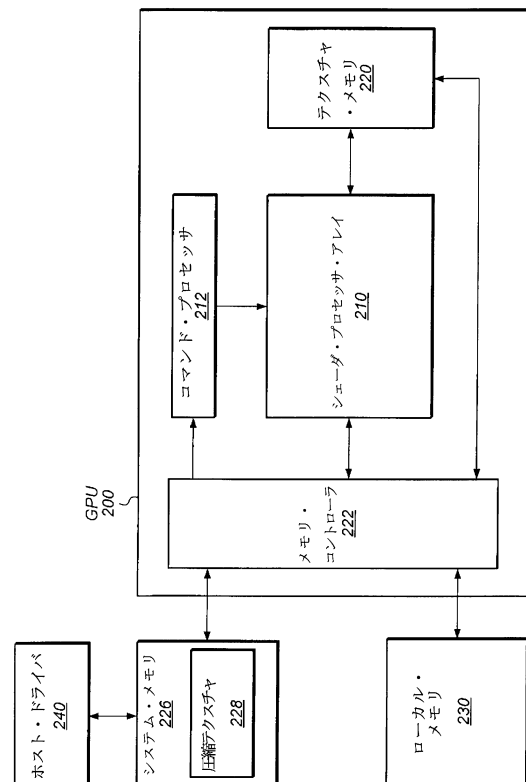
【0067】

いくつかのアプローチの実施形態を示し、説明したが、多くの変更、修正又は改良を上記のアプローチに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ろう。従って、変更、修正又は改良は、本明細書で説明する方法及び機構の範囲内であると考えられる。また、上記の実施形態は、実施の非制限的な例にすぎないことを強調する。

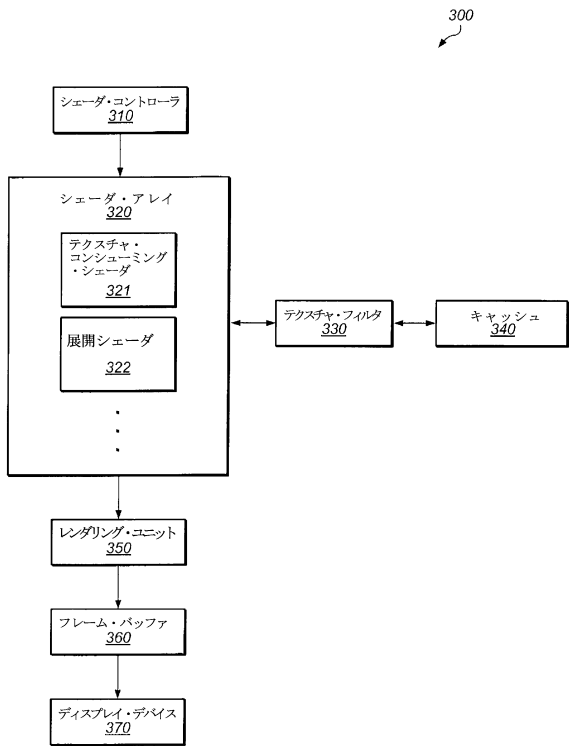
【図1】



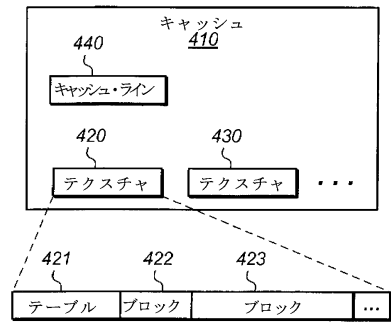
【図2】



【図 3】



【図 4 A】

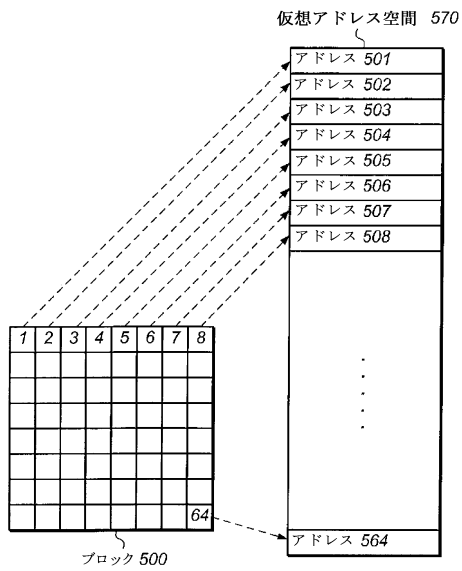


【図 4 B】

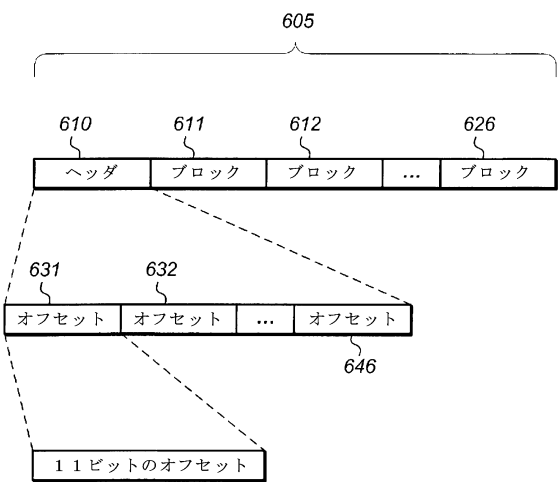
テーブル 421

仮想 アドレス	物理 アドレス	ブロック
0x000	0x100	422
0x800	0x200	423
0x2000	0x800	
⋮	⋮	⋮

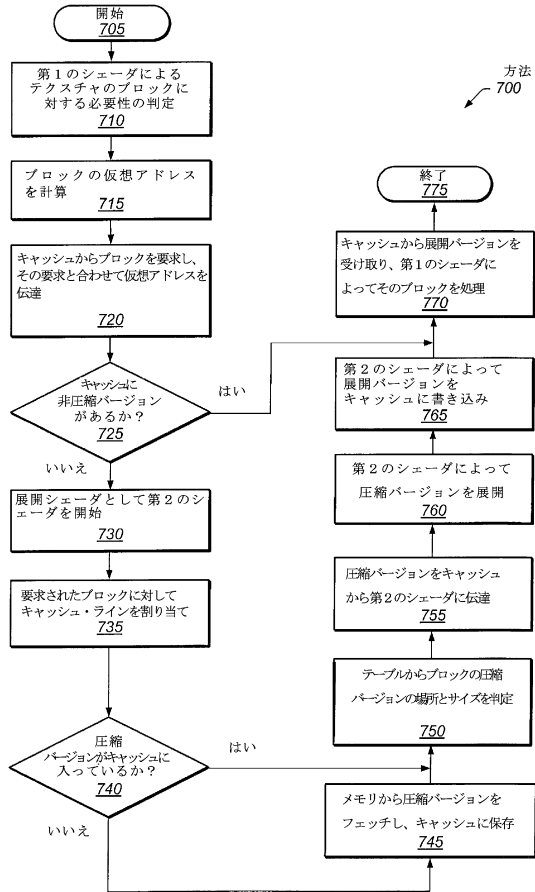
【図 5】



【図 6】



【図 7】



フロントページの続き

(74)代理人 100162156

弁理士 村雨 圭介

(72)発明者 コンスタンチン アイオールチャ

アメリカ合衆国 9 5 1 2 0 カリフォルニア州、サン ノゼ、ウッディド レイク ドライブ
7 1 8 6

(72)発明者 ジョン ダブリュ . ブラザーズ

アメリカ合衆国 9 4 0 8 5 カリフォルニア州、サニーベール、レークサイド ドライブ 1 2
5 7、# 1 2 2 6

審査官 村松 貴士

(56)参考文献 特開 2 0 0 0 - 1 0 5 8 3 9 (J P , A)

米国特許第 0 6 4 5 2 6 0 2 (U S , B 1)

米国特許第 0 6 9 5 9 1 1 0 (U S , B 1)

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 T 1 5 / 0 4