(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6542352号
(P6542352)

(45)発行日　令和1年7月10日(2019.7.10)　(24)登録日　令和1年6月21日(2019.6.21)

(51)Int.Cl.			F I		
G06F	7/483	(2006.01)	G06F	7/483	
G06F	9/30	(2018.01)	G06F	9/30	350A
G06F	9/315	(2006.01)	G06F	9/315	S
G06F	9/38	(2006.01)	G06F	9/38	370C

請求項の数 13　(全 21 頁)

(21)出願番号	特願2017-502104(P2017-502104)
(86)(22)出願日	平成27年6月22日(2015.6.22)
(65)公表番号	特表2017-527014(P2017-527014A)
(43)公表日	平成29年9月14日(2017.9.14)
(86)国際出願番号	PCT/US2015/036960
(87)国際公開番号	WO2016/010681
(87)国際公開日	平成28年1月21日(2016.1.21)
審査請求日	平成30年5月30日(2018.5.30)
(31)優先権主張番号	14/331,991
(32)優先日	平成26年7月15日(2014.7.15)
(33)優先権主張国	米国(US)

(73)特許権者　595020643
クゥアルコム・インコーポレイテッド
ＱＵＡＬＣＯＭＭ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５
(74)代理人　100108855
弁理士　蔵田　昌俊
(74)代理人　100109830
弁理士　福原　淑弘
(74)代理人　100158805
弁理士　井関　守三
(74)代理人　100112807
弁理士　岡田　貴志

最終頁に続く

(54)【発明の名称】算術論理ユニットにおいて使用するためのベクトルスケーリング命令

(57)【特許請求の範囲】

【請求項１】

ベクトルをスケーリングするための方法であって、前記方法は、

少なくとも１つのプロセッサによって、ベクトルの成分を受信することと、ここにおいて、前記ベクトルの前記成分の各々が少なくとも１つの指数を備える、

前記少なくとも１つのプロセッサによって、前記ベクトルの前記成分のそれぞれの指数の中から最大指数を決定することと、

前記少なくとも１つのプロセッサの算術論理ユニット（ＡＬＵ）によって、前記ベクトルをスケーリングすることであって、前記ベクトルの前記成分の前記それぞれの指数の各指数について、

前記それぞれの成分が０または非正規数である場合、前記指数を０に設定することと、

前記それぞれの成分が０でなく、非正規数でない場合、前記指数から前記最大指数を減算することと

を含む、前記ベクトルをスケーリングすることと

を備える、方法。

【請求項２】

前記ベクトルの前記成分の各々が浮動小数点数を備え、ここにおいて、前記浮動小数点数が、符号ビット、仮数、および前記指数として表される、請求項１に記載の方法。

【請求項３】

前記ベクトルが３次元ベクトルを備え、
前記ベクトルの前記成分が、ｘ成分と、ｙ成分と、ｚ成分とを備える、
請求項１に記載の方法。
【請求項４】
前記ベクトルをスケーリングすることが、
前記ＡＬＵによって、第１のクロックサイクルにおいて前記ベクトルの前記ｘ成分の第１の指数から前記最大指数を減算することによって、前記ベクトルの前記ｘ成分をスケーリングすることと、
前記ＡＬＵによって、第２のクロックサイクルにおいて前記ベクトルの前記ｙ成分の第２の指数から前記最大指数を減算することによって、前記ベクトルの前記ｙ成分をスケーリングすることと、
前記ＡＬＵによって、第３のクロックサイクルにおいて前記ベクトルの前記ｚ成分の第３の指数から前記最大指数を減算することによって、前記ベクトルの前記ｚ成分をスケーリングすることと
をさらに備える、請求項３に記載の方法。
【請求項５】
前記スケーリングされたｘ成分と、前記スケーリングされたｙ成分と、前記スケーリングされたｚ成分とをメモリ中の連続する記憶ロケーションに出力すること
をさらに備える、請求項４に記載の方法。
【請求項６】
前記ＡＬＵがハードウェアデジタル回路を備える、請求項１に記載の方法。
【請求項７】
ベクトルをスケーリングするための装置であって、前記装置は、
ベクトルの成分を記憶するように構成されたメモリと、ここにおいて、前記ベクトルの前記成分の各々が少なくとも１つの指数を備える、
前記ベクトルの前記成分のそれぞれの指数の中から最大指数を決定すること
を行うように構成された少なくとも１つのプロセッサと、
前記ベクトルをスケーリングするように構成された算術論理ユニット（ＡＬＵ）であって、前記ベクトルの前記成分の前記それぞれの指数の各指数について、
前記それぞれの成分が０または非正規数である場合、前記指数を０に設定することと、
前記それぞれの成分が０でなく、非正規数でない場合、前記指数から前記最大指数を減算することと
を含む、算術論理ユニット（ＡＬＵ）と
を備える、装置。
【請求項８】
前記ベクトルの前記成分の各々が浮動小数点数を備え、ここにおいて、前記浮動小数点数が、符号ビット、仮数、および前記指数として表される、請求項７に記載の装置。
【請求項９】
前記ベクトルが３次元ベクトルを備え、
前記ベクトルの前記成分が、ｘ成分と、ｙ成分と、ｚ成分とを備える、
請求項７に記載の装置。
【請求項１０】
前記ＡＬＵが、
第１のクロックサイクルにおいて前記ベクトルの前記ｘ成分の第１の指数から前記最大指数を減算することによって、前記ベクトルの前記ｘ成分をスケーリングすることと、
第２のクロックサイクルにおいて前記ベクトルの前記ｙ成分の第２の指数から前記最大指数を減算することによって、前記ベクトルの前記ｙ成分をスケーリングすることと、
第３のクロックサイクルにおいて前記ベクトルの前記ｚ成分の第３の指数から前記最大指数を減算することによって、前記ベクトルの前記ｚ成分をスケーリングすることと

を行うように構成された、請求項９に記載の装置。
【請求項１１】
前記ＡＬＵが、
前記スケーリングされたｘ成分と、前記スケーリングされたｙ成分と、前記スケーリングされたｚ成分とを前記メモリ中の連続する記憶ロケーションに出力すること
を行うように構成された、請求項１０に記載の装置。
【請求項１２】
前記ＡＬＵがハードウェアデジタル回路を備える、請求項７に記載の装置。
【請求項１３】
実行されたとき、１つまたは複数のプログラマブルプロセッサに、請求項１ないし６のいずれか一項に記載の方法を行わせる命令を記憶するコンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
[0001] 本開示は、コンピュータ処理におけるベクトルスケーリング（vector scaling）に関する。
【背景技術】
【０００２】
[0002] ベクトル正規化（vector normalization）は、ベクトルの長さを計算することと、次いで、ベクトルの各成分（component）をベクトルの計算された長さで除算することとを必要とする、ベクトルに関する演算である。３次元ベクトル（three-dimensional vector）（ｘ，ｙ，ｚ）の長さが（$x^2+y^2+z^2$）の平方根として計算される場合、そのような計算は、ベクトルの（ｘ，ｙ，ｚ）値が大きい場合、計算の中間結果を記憶するレジスタ（register）をオーバーフロー（overflow）させることがある。
【発明の概要】
【０００３】
[0003] 本開示は、コンピュータ処理におけるベクトルスケーリングのための技法を提示する。本開示の技法によれば、ベクトルが正規化される前に、ベクトルは、正規化中にベクトルの長さを計算することが、ベクトルの長さを計算することの中間結果を記憶するレジスタをオーバーフローさせることがないように、スケーリング（scale）され得る。グラフィックス処理ユニット（ＧＰＵ：graphics processing unit）の算術および論理ユニット（ＡＬＵ：arithmetic and logic unit）が、ベクトルダウンスケーリング（vector downscaling）を実行するための３サイクルスケール命令（three-cycle scale instruction）を実行するように構成され得る。ＡＬＵによって与えられるベクトルダウンスケーリングを実行するための命令（instruction）は、潜在的に、ソフトウェアベースのベクトルダウンスケーリングよりも比較的効率的にベクトルダウンスケーリングを実行し得る。
【０００４】
[0004] 本開示の一例では、ベクトルをスケーリングするための方法が、少なくとも１つのプロセッサ(processor)によって、ベクトルの成分を受信することを含み得、ここにおいて、ベクトルの成分の各々は少なくとも１つの指数（exponent）を備える。本方法は、少なくとも１つのプロセッサによって、ベクトルの成分のそれぞれの指数の中から最大指数（maximum exponent）を決定することをさらに含み得る。本方法は、少なくとも１つのプロセッサによって、最大指数に少なくとも部分的に基づいてスケーリング値（scaling value）を決定することをさらに含み得る。本方法は、少なくとも１つのプロセッサの算術論理ユニット（ＡＬＵ：arithmetic logic unit）によって、ベクトルの成分のそれぞれの指数の各々からスケーリング値を減算（subtract）することによって、ベクトルをスケーリングすることをさらに含み得る。
【０００５】
[0005] 本開示の別の例では、ベクトルをスケーリングするための装置が、ベクトルの

成分を記憶するように構成されたメモリを含み得、ここにおいて、ベクトルの成分の各々は少なくとも１つの指数を備える。本装置は、ベクトルの成分のそれぞれの指数の中から最大指数を決定することと、最大指数に少なくとも部分的に基づいてスケーリング値を決定することとを行うように構成された少なくとも１つのプロセッサをさらに含み得る。本装置は、ベクトルの成分のそれぞれの指数の各々からスケーリング値を減算することによって、ベクトルをスケーリングするように構成された算術論理ユニット（ＡＬＵ）をさらに含み得る。

【０００６】

[0006] 本開示の別の例では、ベクトルをスケーリングするための装置が、ベクトルの成分を受信するための手段を含み得、ここにおいて、ベクトルの成分の各々は少なくとも１つの指数を備える。本装置は、ベクトルの成分のそれぞれの指数の中から最大指数を決定するための手段をさらに含み得る。本装置は、最大指数に少なくとも部分的に基づいてスケーリング値を決定するための手段をさらに含み得る。本装置は、ベクトルの成分のそれぞれの指数の各々からスケーリング値を減算することによって、ベクトルをスケーリングするための手段をさらに含み得る。

【０００７】

[0007] 本開示の別の例では、コンピュータ可読記憶媒体（computer-readable storage medium）が、実行されたとき、１つまたは複数のプログラマブルプロセッサ（programmable processor）に、ベクトルの成分を受信することと、ここにおいて、ベクトルの成分の各々が少なくとも１つの指数を備える、ベクトルの成分のそれぞれの指数の中から最大指数を決定することと、最大指数に少なくとも部分的に基づいてスケーリング値を決定することと、ベクトルの成分のそれぞれの指数の各々から最大指数を減算することによって、ベクトルをスケーリングすることとを行わせる命令を記憶し得る。

【０００８】

[0008] １つまたは複数の例の詳細が添付の図面および以下の説明に記載されている。他の特徴、目的、および利点は、その説明および図面、ならびに特許請求の範囲から明らかになろう。

【図面の簡単な説明】

【０００９】

【図１】[0009] 本開示の１つまたは複数の態様を実装するように構成され得る例示的なコンピューティングデバイスを示すブロック図。

【図２】[0010] 図１のＣＰＵ、ＧＰＵ、およびシステムメモリの例示的な実装形態をさらに詳細に示すブロック図。

【図３】[0011] 本開示で開示する技法に従ってスケーリングされ得る例示的な３次元ベクトルを示す概念図。

【図４】[0012] ベクトルの各成分を表すための例示的な浮動小数点フォーマット（floating point format）を示す概念図。

【図５】[0013] ベクトルをスケーリングするための例示的なプロセスを示すフローチャート。

【発明を実施するための形態】

【００１０】

[0014] 概して、本開示は、ベクトル正規化演算が演算の中間結果を記憶するレジスタをオーバーフローさせないように、ハードウェアを介してベクトルをスケーリングするための技法について説明する。一例では、プロセッサが、ベクトルをスケーリングするためのソフトウェアコード（software code）を実行し得る。３次元ベクトルの場合、ソフトウェアコードは、３次元ベクトルの最も大きい成分を決定し、３次元ベクトルの各成分を最も大きい成分で除算するためのコードを含み得る。しかしながら、ベクトルをスケーリングするためのソフトウェアコードは、ベクトルをスケーリングするためのハードウェアベースの技法よりも遅くなり得る。したがって、ベクトルをスケーリングするためのハードウェアベースの手法が、性能を増加させ得る。

【００１１】
[0015] 中央処理ユニット（ＣＰＵ：central processing unit）またはグラフィックス処理ユニット（ＧＰＵ）などのプロセッサが、ハードウェア算術論理ユニット（ＡＬＵ）を含み得る。ＡＬＵは、整数算術（integer arithmetic）および論理演算（logical operation）を迅速に実行することが可能であるデジタル回路であり得る。したがって、ＡＬＵは、より効率的にベクトルをスケーリングするための理想的な１つのハードウェアであり得る。しかしながら、ＡＬＵは、しばしば、加算、減算、ＡＮＤ、およびＯＲ演算などの単純な演算のみを実行するように設計され得るので、ＡＬＵは、ベクトルの成分を最も大きい成分で除算することを一般的に含む、ベクトルをスケーリングするための上記で説明した技法を実装するために必要な乗算または除算演算をサポートしないことがある。
【００１２】
[0016] 本開示の態様によれば、ＧＰＵまたはＣＰＵなど、少なくとも１つのプロセッサが、ベクトルの成分を受信し得、ここにおいて、ベクトルの成分の各々が少なくとも１つの指数を備える、ベクトルの成分のそれぞれの指数の中から最大指数を決定し得る。少なくとも１つのプロセッサはさらに、最大指数に少なくとも部分的に基づいてスケーリング値を決定し得る。ＣＰＵまたはＧＰＵのＡＬＵは、ベクトルの成分のそれぞれの指数の各々からスケーリング係数（scaling factor）を減算することによって、ベクトルをスケーリングし得る。
【００１３】
[0017] 図１は、本開示の１つまたは複数の態様を実装するように構成され得る例示的なコンピューティングデバイスを示すブロック図である。図１に示されているように、コンピューティングデバイス２は、限定はしないが、ビデオデバイス、メディアプレーヤ、セットトップボックス、携帯電話およびいわゆるスマートフォンなどのワイヤレスハンドセット、携帯情報端末（ＰＤＡ）、デスクトップコンピュータ、ラップトップコンピュータ、ゲーミングコンソール、ビデオ会議ユニット、タブレットコンピューティングデバイスなどを含むコンピューティングデバイスであり得る。図１の例では、コンピューティングデバイス２は、中央演算処理装置（ＣＰＵ）６と、システムメモリ１０と、ＧＰＵ１２とを含み得る。コンピューティングデバイス２はまた、ディスプレイプロセッサ１４と、トランシーバモジュール３と、ユーザインターフェース４と、ディスプレイ８とを含み得る。トランシーバモジュール３とディスプレイプロセッサ１４とは両方とも、ＣＰＵ６および／またはＧＰＵ１２と同じ集積回路（ＩＣ）の一部であり得るか、両方ともＣＰＵ６および／またはＧＰＵ１２を含む１つまたは複数のＩＣの外部にあり得るか、あるいはＣＰＵ６および／またはＧＰＵ１２を含むＩＣの外部にあるＩＣ中に形成され得る。
【００１４】
[0018] コンピューティングデバイス２は、明快のために図１に示されていない追加のモジュールまたはユニットを含み得る。たとえば、コンピューティングデバイス２は、コンピューティングデバイス２がモバイルワイヤレス電話である例において電話通信を実現するために、そのいずれも図１に示されていないスピーカーおよびマイクロフォンを含み、またはコンピューティングデバイス２がメディアプレーヤである例においてスピーカーを含み得る。コンピューティングデバイス２はまた、ビデオカメラを含み得る。さらに、コンピューティングデバイス２に示された様々なモジュールおよびユニットは、コンピューティングデバイス２のあらゆる例において必要であるとは限らない。たとえば、ユーザインターフェース４およびディスプレイ８は、コンピューティングデバイス２が、デスクトップコンピュータ、あるいは外部ユーザインターフェースまたはディスプレイとインターフェースする能力がある他のデバイスである例において、コンピューティングデバイス２の外部にあり得る。
【００１５】
[0019] ユーザインターフェース４の例としては、限定はしないが、トラックボール、マウス、キーボード、および他のタイプの入力デバイスがある。ユーザインターフェース４はまた、タッチスクリーンであり得、ディスプレイ８の一部として組み込まれ得る。ト

ランシーバモジュール３は、コンピューティングデバイス２と別のデバイスまたはネットワークとの間のワイヤレスまたはワイヤード通信を可能にするための回路を含み得る。トランシーバモジュール３は、ワイヤードまたはワイヤレス通信のための変調器、復調器、増幅器および他のそのような回路を含み得る。

【００１６】

[0020] プロセッサ６は、実行のためのコンピュータプログラムの命令を処理するように構成された中央処理ユニット（ＣＰＵ）など、マイクロプロセッサであり得る。プロセッサ６は、コンピューティングデバイス２の動作を制御する汎用または専用プロセッサを備え得る。ユーザが、プロセッサ６に１つまたは複数のソフトウェアアプリケーションを実行させるために、コンピューティングデバイス２に入力を与え得る。プロセッサ６上で実行するそれらのソフトウェアアプリケーションは、たとえば、オペレーティングシステム、ワードプロセッサアプリケーション、電子メールアプリケーション、スプレッドシートアプリケーション、メディアプレーヤアプリケーション、ビデオゲームアプリケーション、グラフィカルユーザインターフェースアプリケーション、または別のプログラムを含み得る。さらに、プロセッサ６は、ＧＰＵ１２の動作を制御するためのＧＰＵドライバ２２を実行し得る。ユーザは、ユーザ入力インターフェース４を介してコンピューティングデバイス２に結合された、キーボード、マウス、マイクロフォン、タッチパッド、または別の入力デバイスなどの１つまたは複数の入力デバイス（図示せず）を介して、コンピューティングデバイス２に入力を与え得る。

【００１７】

[0021] プロセッサ６上で実行するソフトウェアアプリケーションは、ディスプレイ８へのグラフィックスデータのレンダリング（rendering）を引き起こすようにプロセッサ６に命令する、１つまたは複数のグラフィックスレンダリング命令を含み得る。いくつかの例では、ソフトウェア命令は、たとえば、オープングラフィックスライブラリ（ＯｐｅｎＧＬ（登録商標）：Open Graphics Library）アプリケーションプログラミングインターフェース（ＡＰＩ：application programming interface）、オープングラフィックスライブラリエンベディッドシステムズ（ＯｐｅｎＧＬ　ＥＳ：Open Graphics Library Embedded Systems）ＡＰＩ、Ｄｉｒｅｃｔ３Ｄ　ＡＰＩ、Ｘ３Ｄ　ＡＰＩ、ＲｅｎｄｅｒＭａｎ　ＡＰＩ、ＷｅｂＧＬ　ＡＰＩ、あるいは他の公開またはプロプライエタリ規格グラフィックスＡＰＩなど、グラフィックスＡＰＩに準拠し得る。グラフィックスレンダリング命令を処理するために、プロセッサ６は、ＧＰＵ１２にグラフィックスデータのレンダリングの一部または全部を実行させるために、（たとえば、ＧＰＵドライバ２２を通して）ＧＰＵ１２に１つまたは複数のグラフィックスレンダリングコマンドを発行し得る。いくつかの例では、レンダリングされるべきグラフィックスデータは、グラフィックスプリミティブ（graphics primitive）のリスト、たとえば、点、線、三角形、四角形、三角形ストリップ（triangle strip）などを含み得る。

【００１８】

[0022] ＧＰＵ１２は、１つまたは複数のグラフィックスプリミティブをディスプレイ８にレンダリングするためにグラフィックス演算を実行するように構成され得る。したがって、プロセッサ６上で実行しているソフトウェアアプリケーションのうちの１つがグラフィックス処理を必要とするとき、プロセッサ６は、ディスプレイ８にレンダリングするために、ＧＰＵ１２にグラフィックスコマンドとグラフィックスデータとを与え得る。グラフィックスデータは、たとえば、描画コマンド、状態情報、プリミティブ情報、テクスチャ情報などを含み得る。ＧＰＵ１２は、いくつかの事例では、プロセッサ６よりも、複雑なグラフィック関係演算の効率的な処理を行う、高度並列構造で構築され得る。たとえば、ＧＰＵ１２は、並列な様式で複数の頂点またはピクセルに対して演算するように構成された、シェーダユニット（shader unit）など、複数の処理要素を含み得る。ＧＰＵ１２の高度並列特性は、いくつかの事例では、ＧＰＵ１２が、プロセッサ６を使用してディスプレイ８にシーンを直接描画するよりも迅速に、グラフィックス画像（たとえば、ＧＵＩならびに２次元（２Ｄ）および／または３次元（３Ｄ）のグラフィックスシーン）をデ

ィスプレイ８上に描画することを可能にし得る。

【００１９】

[0023] ＧＰＵ１２は、いくつかの事例では、コンピューティングデバイス２のマザーボードに組み込まれ得る。他の事例では、ＧＰＵ１２は、コンピューティングデバイス２のマザーボード中のポートに取り付けられたグラフィックスカード上に存在し得、または場合によっては、コンピューティングデバイス２と相互動作するように構成された周辺デバイス内に組み込まれ得る。ＧＰＵ１２は、１つまたは複数のマイクロプロセッサ、特定用途向け集積回路（ＡＳＩＣ）、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、デジタル信号プロセッサ（ＤＳＰ）、あるいは他の等価な集積またはディスクリート論理回路など、１つまたは複数のプロセッサを含み得る。ＧＰＵ１２はまた、１つまたは複数のプロセッサコアを含み得、したがって、ＧＰＵ１２はマルチコアプロセッサと呼ばれることがある。

【００２０】

[0024] ＧＰＵ１２は、グラフィックスメモリ４０に直接結合され得る。したがって、ＧＰＵ１２は、バスを使用することなしに、グラフィックスメモリ４０からデータを読み取り、グラフィックスメモリ４０にデータを書き込み得る。言い換えれば、ＧＰＵ１２は、オフチップメモリの代わりに、ローカルストレージを使用してデータをローカルに処理し得る。そのようなグラフィックスメモリ４０は、オンチップメモリと呼ばれることがある。これは、大量のバストラフィックを経験することがある、バスを介してデータを読み取りおよび書き込みするＧＰＵ１２の必要をなくすことによって、ＧＰＵ１２がより効率的な様式で動作することを可能にする。しかしながら、いくつかの事例では、ＧＰＵ１２は、別個のメモリを含まず、代わりにバスを介してシステムメモリ１０を利用し得る。グラフィックスメモリ４０は、たとえば、ランダムアクセスメモリ（ＲＡＭ）、スタティックＲＡＭ（ＳＲＡＭ）ダイナミックＲＡＭ（ＤＲＡＭ）、消去可能プログラマブルＲＯＭ（ＥＰＲＯＭ）、電気的消去可能プログラマブルＲＯＭ（ＥＥＰＲＯＭ（登録商標））、フラッシュメモリ、磁気データ媒体または光記憶媒体など、１つまたは複数の揮発性または不揮発性メモリあるいは記憶デバイスを含み得る。

【００２１】

[0025] いくつかの例では、ＧＰＵ１２は、十分に形成された画像をシステムメモリ１０に記憶し得る。ディスプレイプロセッサ１４は、システムメモリ１０から画像を取り出し、ディスプレイ８のピクセルを、画像を表示するために点灯させる値を出力し得る。ディスプレイ８は、ＧＰＵ１２によって生成された画像コンテンツを表示するコンピューティングデバイス２のディスプレイであり得る。ディスプレイ８は、液晶ディスプレイ（ＬＣＤ）、有機発光ダイオードディスプレイ（ＯＬＥＤ）、陰極線管（ＣＲＴ）ディスプレイ、プラズマディスプレイ、または別のタイプのディスプレイデバイスであり得る。

【００２２】

[0026] 上記で説明したように、ＧＰＵ１２は、整数算術、浮動小数点、および論理演算を実行するデジタル回路であり得るＡＬＵ２４を含み得る。ＡＬＵ２４によって実行され得る演算は、加算、減算、およびビット単位演算を含み得る。いくつかの例では、ＡＬＵ２４は、乗算および除算などの演算を実行することが可能でないことがある。いくつかの例では、プロセッサ６はまた、それが算術および論理演算を実行するデジタル回路であり得るという点でＡＬＵ２４と同様に動作し得る、ＡＬＵを含み得る。

【００２３】

[0027] 図２は、図１のプロセッサ６、ＧＰＵ１２、およびシステムメモリ１０の例示的な実装形態をさらに詳細に示すブロック図である。図２に示されているように、プロセッサ６は、その各々がプロセッサ上で実行する１つまたは複数のソフトウェアアプリケーションまたはサービスであり得る、少なくとも１つのソフトウェアアプリケーション１８と、グラフィックスＡＰＩ２０と、ＧＰＵドライバ２２とを含み得る。

【００２４】

[0028] プロセッサ６およびＧＰＵ１２にとって利用可能なメモリは、システムメモリ

１０と、フレームバッファ１６とを含み得る。フレームバッファ１６は、システムメモリ１０の一部であり得るか、またはシステムメモリ１０とは別個であり得る。フレームバッファ１６は、レンダリングされた画像データを記憶し得る。

【００２５】

[0029] ソフトウェアアプリケーション１８は、ＧＰＵ１２の機能を利用する任意のアプリケーションであり得る。たとえば、ソフトウェアアプリケーション１８は、ＧＵＩアプリケーション、オペレーティングシステム、ポータブルマッピングアプリケーション、エンジニアリングまたは芸術アプリケーション（artistic application）のためのコンピュータ支援設計プログラム、ビデオゲームアプリケーション、あるいは２Ｄまたは３Ｄグラフィックスを使用する別のタイプのソフトウェアアプリケーションであり得る。

【００２６】

[0030] ソフトウェアアプリケーション１８は、グラフィカルユーザインターフェース（ＧＵＩ：graphical user interface）および／またはグラフィックスシーンをレンダリングするようにＧＰＵ１２に命令する１つまたは複数の描画命令を含み得る。たとえば、描画命令は、ＧＰＵ１２によってレンダリングされるべき１つまたは複数のグラフィックスプリミティブのセットを定義する命令を含み得る。いくつかの例では、描画命令は、ＧＵＩにおいて使用される複数のウィンドウ処理面の全部または一部を集合的に定義し得る。追加の例では、描画命令は、アプリケーションによって定義されたモデル空間内または世界空間内の１つまたは複数のグラフィックスオブジェクトを含む、グラフィックスシーンの全部または一部を集合的に定義し得る。

【００２７】

[0031] ソフトウェアアプリケーション１８は、１つまたは複数のグラフィックスプリミティブを表示可能なグラフィックス画像にレンダリングするための１つまたは複数のコマンドをＧＰＵ１２に発行するために、グラフィックスＡＰＩ２０を介してＧＰＵドライバ２２を呼び出し得る。たとえば、ソフトウェアアプリケーション１８は、ＧＰＵ１２にプリミティブ定義（primitive definition）を与えるために、グラフィックスＡＰＩ２０を介してＧＰＵドライバ２２を呼び出し得る。いくつかの事例では、プリミティブ定義は、描画プリミティブ、たとえば、三角形、長方形、三角形ファン（triangle fan）、三角形ストリップなどのリストの形式でＧＰＵ１２に与えられ得る。プリミティブ定義は、レンダリングされるべきプリミティブに関連付けられた１つまたは複数の頂点を指定する頂点仕様を含み得る。頂点仕様は、各頂点についての位置座標、ならびにいくつかの事例では、たとえば、色座標、法線ベクトル、およびテクスチャ座標など、頂点に関連付けられた他の属性を含み得る。プリミティブ定義はまた、プリミティブタイプ情報（たとえば、三角形、長方形、三角形ファン、三角形ストリップなど）、スケーリング情報、回転情報などを含み得る。

【００２８】

[0032] ソフトウェアアプリケーション１８によってＧＰＵドライバ２２に発行された命令に基づいて、ＧＰＵドライバ２２は、プリミティブをレンダリングするためにＧＰＵ１２が実行するための１つまたは複数の演算を指定する１つまたは複数のコマンドを構築し得る。ＧＰＵ１２がＣＰＵ６からコマンドを受信したとき、グラフィックス処理パイプライン（graphics processing pipeline）が、コマンドを復号し、コマンド中で指定された演算を実行するようにグラフィックス処理パイプラインを構成する。たとえば、グラフィックス処理パイプラインにおける入力アセンブラが、プリミティブデータを読み取り、グラフィックス処理パイプラインにおける他のグラフィックスパイプラインステージによる使用のためにデータをプリミティブにアセンブルし得る。指定された演算を実行した後、グラフィックス処理パイプラインは、レンダリングされたデータを、ディスプレイデバイスに関連付けられたフレームバッファ１６に出力する。

【００２９】

[0033] フレームバッファ１６は、ＧＰＵ１２のためにデスティネーションピクセル（destination pixel）を記憶する。デスティネーションピクセルは、一意のスクリーンピ

クセルロケーションに関連付けられ得る。いくつかの例では、フレームバッファ１６は、各デスティネーションピクセルについて色成分とデスティネーションアルファ値とを記憶し得る。たとえば、フレームバッファ１６は、各ピクセルについて赤、緑、青、アルファ（ＲＧＢＡ）成分を記憶し得、「ＲＧＢ」成分は色値に対応し、「Ａ」成分はデスティネーションアルファ値に対応する。フレームバッファ１６とシステムメモリ１０とは別個のメモリユニットであるものとして示されているが、他の例では、フレームバッファ１６は、システムメモリ１０の一部であり得る。

【００３０】

[0034] いくつかの例では、グラフィックス処理パイプラインは、頂点シェーダステージ、ハルシェーダステージ（hull shader stage）、ドメインシェーダステージ、ジオメトリシェーダステージ、およびピクセルシェーダステージのうちの１つまたは複数を含み得る。グラフィックス処理パイプラインのこれらのステージは、シェーダステージと見なされ得る。これらのシェーダステージは、ＧＰＵ１２においてシェーダユニット４６上で実行する１つまたは複数のシェーダプログラムとして実装され得る。シェーダユニット４６は、処理構成要素のプログラマブルパイプラインとして構成された１つまたは複数のシェーダユニットを備え得る。いくつかの例では、シェーダユニット４６は、「シェーダプロセッサ（shader processor）」または「ユニファイドシェーダ（unified shader）」と呼ばれることがあり、グラフィックスをレンダリングするために、ジオメトリ、頂点、ピクセル、または他のシェーディング演算を実行し得る。

【００３１】

[0035] ＧＰＵ１２は、グラフィックス処理パイプラインにおける頂点シェーダステージ、ハルシェーダステージ、ドメインシェーダステージ、ジオメトリシェーダステージ、およびピクセルシェーダステージのうちの１つまたは複数を実行するためにコマンドをシェーダユニット４６に送ることによって、頂点シェーディング、ハルシェーディング、ドメインシェーディング、ジオメトリシェーディング、ピクセルシェーディングなどの様々なシェーディング演算を実行するようにシェーダユニット４６を指定し得る。いくつかの例では、ＧＰＵドライバ２２が、１つまたは複数のシェーダプログラムをコンパイルし、コンパイルされたシェーダプログラムを、ＧＰＵ１２内に含まれている１つまたは複数のプログラマブルシェーダユニット上にダウンロードするように構成され得る。シェーダプログラムは、たとえば、ＯｐｅｎＧＬシェーディング言語（ＧＬＳＬ）、ハイレベルシェーディングランゲージ（ＨＬＳＬ）、グラフィックスのためのＣ（Ｃｇ）シェーディング言語など、高レベルシェーディング言語で書き込まれ得る。コンパイルされたシェーダプログラムは、ＧＰＵ１２内のシェーダユニット４６の動作を制御する１つまたは複数の命令を含み得る。たとえば、シェーダプログラムは、頂点シェーダステージの機能を実行するためにシェーダユニット４６によって実行され得る頂点シェーダプログラム、ハルシェーダステージの機能を実行するためにシェーダユニット４６によって実行され得るハルシェーダプログラム、ドメインシェーダステージの機能を実行するためにシェーダユニット４６によって実行され得るドメインシェーダプログラム、ジオメトリシェーダステージの機能を実行するためにシェーダユニット４６によって実行され得るジオメトリシェーダプログラム、および／またはピクセルシェーダの機能を実行するためにシェーダユニット４６によって実行され得るピクセルシェーダプログラムを含み得る。頂点シェーダプログラムは、プログラマブル頂点シェーダユニットまたはユニファイドシェーダユニットの実行を制御し、１つまたは複数の頂点ごとの演算を指定する命令を含み得る。

【００３２】

[0036] シェーダユニット４６はプロセッサコア４８を含み得、プロセッサコア４８の各々は、フェッチングおよび復号演算のための１つまたは複数の構成要素と、算術計算を行うための１つまたは複数の算術論理ユニットと、１つまたは複数のメモリと、キャッシュと、レジスタとを含み得る。いくつかの例では、プロセッサコア４８は、スカラー処理要素と呼ばれることもある。プロセッサコア４８の各々は汎用レジスタ２５を含み得る。汎用レジスタ２５は、プロセッサコア４８中のＡＬＵ２４によって直接アクセスされるべ

きデータを記憶し得る。たとえば、汎用レジスタ２５は、ＡＬＵ２４によってスケーリングされるべきベクトル成分を記憶し得、ＡＬＵ２４によって出力された、スケーリングされたベクトル成分をも記憶し得る。

【００３３】

[0037] プロセッサコア４８の各々は、ＡＬＵ２４など、スカラーＡＬＵを含み得る。上記で説明したように、ＡＬＵ２４は、整数算術、浮動小数点、および論理演算を実行するデジタル回路であり得る。ＡＬＵ２４によって実行され得る演算は、加算、減算、およびビット単位演算を含み得る。いくつかの例では、ＡＬＵ２４は、乗算および除算などの演算を実行することが可能でないことがある。本開示の態様によれば、ＡＬＵ２４は、ベクトルの成分をスケーリングすることによって、ベクトルをスケーリングし得る。ＡＬＵ２４はまた、ベクトルのスケーリングされた成分をグラフィックスメモリ４０または汎用レジスタ２５に出力し得る。

【００３４】

[0038] グラフィックスメモリ４０は、ＧＰＵ１２の集積回路に物理的に組み込んだオンチップストレージまたはメモリである。グラフィックスメモリ４０がオンチップであるので、ＧＰＵ１２は、システムバスを介してシステムメモリ１０から値を読み取るか、またはそれに値を書き込むよりも迅速に、グラフィックスメモリ４０から値を読み取り、またはそれに値を書き込むことが可能であり得る。グラフィックスメモリ４０は、ベクトルの成分を記憶し得、ＡＬＵ２４によるスケーリングの後のベクトルのスケーリングされた成分をも記憶し得る。グラフィックスメモリ４０は、各成分が符号付きビット、シグニフィカント（significant）、および指数としてグラフィックスメモリ４０に記憶され得るように、ベクトルの成分を浮動小数点フォーマットで記憶し得る。

【００３５】

[0039] 図３は、ＣＰＵ６またはＧＰＵ１２によってスケーリングされ得る例示的な３次元ベクトルを示すブロック図である。図３に示されているように、３次元直交座標系５２におけるベクトル５０は、ベクトル５０のそれぞれの成分５４Ａ～５４Ｃ（「成分５４」）の値を示すタプル（ｘ，ｙ，ｚ）によって表され得る。ベクトル５０の成分５４は、ｘ成分５４Ａと、ｙ成分５４Ｂと、ｚ成分５４Ｃとを含み得る。

【００３６】

[0040] 上記で説明したように、ベクトル５０の成分５４の各成分は、符号付きビット、仮数（significand）、および指数（exponent）としてグラフィックスメモリ４０または汎用レジスタ２５に記憶される浮動小数点値（floating point value）であり得る。たとえば、例示的な浮動小数点値１．２３４５は、１２３４５が仮数（significand）または仮数（mantissa）であり得、－４が底（base）１０の指数であり得るような、$12345 * 10^{-4}$に等しくなり得る。他の例では、指数は底２の指数であり得る。負の指数値を表すために、指数は、指数値が正の値に変換されるように、バイアスされるかまたはオフセットされ得る。たとえば、指数値－４が１１としてメモリに記憶され得るように、値１５が指数に加算され得る。

【００３７】

[0041] 図４は、ベクトル５０の成分５４の各成分を表すための例示的な浮動小数点フォーマットを示す概念図である。上記で説明したように、ベクトル５０の成分５４の各成分は浮動小数点値であり得る。図４に示されているように、成分５４の各成分は、浮動小数点フォーマット６０で表され得る。浮動小数点フォーマット６０は、浮動小数点フォーマット６０によって表される浮動小数点値の符号を示す符号ビット（sign bit）６２を含み得る。符号ビット６２は、浮動小数点値の符号が負の場合は１であり得、浮動小数点値の符号が正の場合は０であり得る。浮動小数点フォーマット６０は、指数６４と仮数６６とをさらに含み得る。一例では、３２ビットＩＥＥＥ浮動小数点フォーマット６０の場合、符号ビット６２は１ビットであり得、指数６４は、１２７のバイアスを用いた８ビットであり得、仮数６６は、整数が隠された２３ビットであり得る。たとえば、浮動小数点値－８２．３１２５は、$-1.0100100101_2 * 2^6$に等しくなり得る。この例では

、符号ビット６２は１に設定され得る。指数６４は、１２７のバイアスにより133_{10}である、１０００００１０１であり得、仮数６６は、整数ビットが隠され得るので、０１００１００１０１０００００００００００００であり得る。

【００３８】

[0042] 本開示の態様によれば、プロセッサ６またはＧＰＵ１２は、プロセッサ６またはＧＰＵ１２が、ベクトル正規化演算の中間結果を記憶するレジスタをオーバーフローさせることなしにベクトル５０のベクトル正規化を実行し得るように、ベクトル５０をスケーリングするためにＡＬＵ２４を使用し得る。ＡＬＵ２４はハードウェア回路であり得るので、ベクトル５０のそのようなスケーリングは、たとえば、シェーダユニット４６によって実行されるソフトウェアにおいて実行される代わりに、ハードウェアにおいて実行され得る。さらに、ＡＬＵ２４は、加算演算および減算演算を実行するための機能を含み得るが、乗算演算および／または除算演算を実行するための機能を含まないことがあるので、ＡＬＵ２４は、乗算演算または除算演算のいずれをも実行することなしにベクトル５０スケーリングすることが可能であり得る。ベクトル５０をスケーリングするために、ＧＰＵ１２は、ベクトル５０の成分５４を受信し得る。たとえば、ベクトル５０が３次元ベクトルである場合、ＧＰＵ１２は、ベクトル５０のｘ成分５４Ａ、ｙ成分５４Ｂ、およびｚ成分５４Ｃを受信し得る。ベクトル５０の成分５４は、システムメモリ１０、グラフィックスメモリ４０、シェーダユニット４６のメモリ、汎用レジスタ２５など、メモリに記憶され得る。

【００３９】

[0043] 上記で説明したように、ベクトル５０の成分５４は各々、少なくとも仮数と指数とを含む浮動小数点値であり得る。ＧＰＵ１２は、成分５４の指数の中から最大指数を決定し得る。たとえば、成分５４の指数が－１、２、および５である場合、ＧＰＵ１２は、成分５４の指数の中からの最大指数が５であると決定し得る。いくつかの例では、成分５４の指数の中から最大指数を決定することは、成分５４の指数の中から最大値指数（maximum value exponent）を決定することを含み得る。したがって、たとえば、成分５４の指数が－１、２、および５である場合、ＧＰＵ１２は、５が２または－１よりも大きいので、成分５４の中からの最大値指数が５であると決定し得る。

【００４０】

[0044] ＧＰＵ１２が成分５４の指数の中から最大指数を決定したことに応答して、ＧＰＵ１２は、成分５４の指数の各々をスケーリングするためのスケーリング値を決定し得る。一例では、スケーリング値は最大指数に等しくなり得るので、成分５４の指数のためのスケーリング値は５であり得る。別の例では、ＧＰＵ１２は、スケーリングの後のスケーリングされた指数のアンダーフローおよび／またはオーバーフローを防止するように、スケーリング値を決定し得る。この場合、ＧＰＵ１２は、最大指数に少なくとも部分的に基づいてスケーリング値を決定し得る。たとえば、スケーリング値は、最大指数＋定数であり得る。たとえば、スケーリング値は、最大指数－（ｍａｘｉｍｕｍ＿ｒｅｐｒｅｓｅｎｔａｂｌｅ＿ｅｘｐｏｎｅｎｔ－１）／２＋１であり得る。ｍａｘｉｍｕｍ＿ｒｅｐｒｅｓｅｎｔａｂｌｅ＿ｅｘｐｏｎｅｎｔは、成分５４の浮動小数点フォーマットから導出された定数である、最大表現可能指数（maximum representable exponent）であり得る。たとえば、３２ビットＩＥＥＥ浮動小数点数（floating point number）の場合、最大表現可能指数は１２８である。３２ビット浮動小数点数における指数は８ビットを用いて表されるので、指数範囲は［－１２７，１２８］（すなわち、両端値を含む－１２７から１２８まで）である。別の例では、ＧＰＵ１２は、（最大指数－１）／２－２になるようにスケーリング値を決定し得る。したがって、最大指数１５を仮定すれば、スケーリング値は、（１５－１）／２－２であり得、それは５であり得る。

【００４１】

[0045] ＧＰＵ１２がスケーリング値を決定したことに応答して、ＡＬＵ２４は、成分５４の各指数からスケーリング値を減算することによって成分５４の各成分をスケーリングするように構成され得る。たとえば、ベクトル５０のｘ成分５４Ａ、ｙ成分５４Ｂ、お

よびｚ成分５４Ｃについての指数値－１、２、および５を仮定し、ＧＰＵ１２が、５が成分５４の指数の中からの最大指数であると決定することを仮定し、ＧＰＵ１２が、最大指数の値になるようにスケーリング値を決定すること（すなわち、スケーリング値を５に設定すること）を仮定すれば、ＡＬＵ２４は、ｘ成分５４Ａについての指数値－１から５を減算し得、ＡＬＵ２４は、ｙ成分５４Ｂについての指数値２から５を減算し得、ＡＬＵ２４は、ｚ成分５４Ｃについての指数値５から５を減算し得、それぞれｘ成分５４Ａ、ｙ成分５４Ｂ、およびｚ成分５４Ｃについての指数値－６、－３、および０を有する、スケーリングされた成分５４を生じる。ＡＬＵ２４によって出力された指数を含む、得られたスケーリングされた成分は、グラフィックスメモリ４０、システムメモリ１０、汎用レジスタ２５など、メモリに記憶され得る。

【００４２】

[0046] 上記で説明したように、いくつかの例では、成分５４の指数は、バイアスされた指数（biased exponent）であり得る。ＧＰＵ１２およびＡＬＵ２４は、非バイアス指数（unbiased exponent）と同様の様式で、バイアスされた成分を処理し得る。たとえば、１５が成分５４の各指数に加算されるように、成分５４の指数の値－１、２、および５が１５だけバイアスされる場合、成分５４のバイアスされた指数の値は１４、１７、および２０であり得る。したがって、ＧＰＵ１２は、成分５４のバイアスされた指数の中からの最大指数が２０であると決定し得る。ＧＰＵ１２が成分５４のバイアスされた指数の中から最大指数を決定したことに応答して、ＧＰＵ１２は、最大指数に少なくとも部分的に基づいてスケーリング値を決定し得る。この例では、ＧＰＵ１２は、スケーリング値を最大指数の値に設定し得る。ＧＰＵ１２がスケーリング値を決定したことに応答して、ＡＬＵ２４は、成分５４の各指数からスケーリング値を減算することによって成分５４の各成分をスケーリングし得る。たとえば、ベクトル５０のｘ成分５４Ａ、ｙ成分５４Ｂ、およびｚ成分５４Ｃについてのバイアスされた指数値１４、１７、および２０を仮定し、ＧＰＵ１２が、２０が成分５４の指数の中からの最大指数であると決定することと、スケーリング値が最大指数の値に設定されることとを仮定すれば、ＡＬＵ２４は、ｘ成分５４Ａについての指数値１４から２０を減算し得、ＡＬＵ２４は、ｙ成分５４Ｂについての指数値１７から２０を減算し得、ＡＬＵ２４は、ｚ成分５４Ｃについての指数値２０から２０を減算し得る。ＡＬＵ２４は、成分５４の各指数にバイアス１５を加算し、それぞれｘ成分５４Ａ、ｙ成分５４Ｂ、およびｚ成分５４Ｃについてのバイアスされた指数値９、１２、および１５を有する、スケーリングされた成分５４を生じ得る。

【００４３】

[0047]

ＡＬＵ２４は、ＡＬＵ２４が第１のクロックサイクル（clock cycle）においてスケーリングされたｘ成分を出力し、第２のクロックサイクルにおいてスケーリングされたｙ成分を出力し、第３のクロックサイクルにおいてスケーリングされたｚ成分を出力し得るように、クロックサイクルごとに１つのスケーリングされた成分を出力するように構成され得る。ベクトル５０のスケーリングを実行するための例示的な擬似コード（pseudocode）は、次のように表され得る。

【００４４】

【数１】

```
if(src0、src1、またはsrc2がINFまたはNaNである){／／出力＝入力
        dst = src0;
        dst+1 = src1;
        dst+2 = src2;
}
else maxexp = max(src0.exp, src1.exp, src2.exp);

if(src0が0または非正規である)
        dst＝0　src0符号を維持する;
else dst.exp = src0.exp – maxexp;

if(src1が0または非正規である)
        dst＝0　src1符号を維持する;
else (dst+1).exp = src1.exp – maxexp;

if(src2が0または非正規である)
        dst＝0　src2符号を維持する;
else (dst+2).exp = src2.exp – maxexp;
```

【００４５】

[0048] 上記の擬似コードに示されているように、ｓｒｃ０、ｓｒｃ１、およびｓｒｃ２は、３次元ベクトルの成分のソースメモリロケーションであり得る。ｄｓｔは、第１のスケーリングされた成分のためのメモリ中の宛先ロケーションであり得、ｄｓｔ＋１は、第２のスケーリングされた成分のためのメモリ中の次の連続する宛先ロケーションであり得、ｄｓｔ＋２は、第３のスケーリングされた成分のためのメモリ中の次の連続する宛先ロケーションであり得る。わかるように、スケーリングされた成分は、メモリ中の連続するメモリロケーションに記憶され得る。

【００４６】

[0049] 上記に示されているように、ＧＰＵ１２は、成分のいずれかが無限（infinite）であるか、または数（number）でないかどうかを決定し得る。成分が、定義されていないかまたは表現不可能な値である場合、成分は数でないことがある。数でない成分は、指数６４が１で満たされ得、仮数６６が非０値になり得る。そうである場合、ベクトルの成分はスケーリングされない。他の場合、ＧＰＵ１２は、成分の指数の中からの最大指数に少なくとも部分的に基づいてスケーリング値を決定し得る。ｓｒｃ０、ｓｒｃ１、およびｓｒｃ２中の成分の各々について、指数が０または非正規数（denormal number）である場合、成分はスケーリングされない。非正規数は、最も小さい正規数（normal number）よりも小さい大きさをもつ非０数（non-zero number）であり得る。他の場合、ＡＬＵ２４は、成分の指数をスケーリング値だけ減算することによって、成分をスケーリングし得る。いくつかの例では、宛先メモリロケーションｄｓｔ、ｄｓｔ＋１、またはｄｓｔ＋２のいずれかが、ソースメモリロケーションｓｒｃ０、ｓｒｃ１、またはｓｒｃ２と重複し得る。

【００４７】

[0050] ＧＰＵ１２は、ベクトルのスケーリングされた成分を出力するためのスケーリング関数を（ｒｐｔ２）ｓｃａｌｅ．ｘ、（ｒ）ｘ、（ｘ，ｙ，ｚ）として与えることが

でき、（ｒｐｔ２）は、スケーリング命令が、それが３次元ベクトルのスケーリングされたｘ成分、ｙ成分、およびｚ成分を出力するために合計３回実行し得るように、初期実行の後に２回繰り返すことになることよりも示し得る。ｓｃａｌｅ．ｘは、スケーリング関数のための関数名であり得、ただし、ｓｃａｌｅ．ｘにおけるｘは、ベクトルのスケーリングされていない成分を記憶するソースメモリロケーションの開始ロケーションであり得る。（ｒ）ｘは、スケーリング命令が繰り返されることになることを示し得、（ｒ）ｘにおけるｘは、ベクトルのスケーリングされた成分を記憶するための宛先メモリロケーションの開始ロケーションであり得る。（ｘ，ｙ，ｚ）は、スケーリング関数によってスケーリングされることになるベクトルの成分であり得る。ある時間量中にＡＬＵ２４のパイプラインに発行され得るスケーリング命令の数であり得る、ＡＬＵ２４についてのスループット（throughput）は、スケーリング命令ごとに３サイクルであり得る。発行から完了までのスケーリング命令を実行するためにかかる総時間量であり得る、スケーリング命令を実行する際のＡＬＵ２４のレイテンシ（latency）は、スケーリング命令の実装形態によって異なり得る。

【００４８】

[0051] 図５は、ベクトルをスケーリングするための例示的なプロセスを示すフローチャートである。図５に示されているように、プロセスは、プロセッサ６またはＧＰＵ１２によって、ベクトルの成分を受信することを含み得、ここにおいて、ベクトルの成分の各々が少なくとも１つの指数を備える（５０２）。プロセスは、プロセッサ６またはＧＰＵ１２によって、ベクトルの成分のそれぞれの指数の中から最大指数を決定することをさらに含み得る（５０４）。プロセスは、プロセッサ６またはＧＰＵ１２によって、最大成分（maximum component）に少なくとも部分的に基づいてスケーリング値を決定することをさらに含み得る（５０６）。プロセスは、プロセッサ６またはＧＰＵ１２のＡＬＵ２４によって、ベクトルの成分のそれぞれの指数の各々からスケーリング値を減算することによってベクトルをスケーリングすることをさらに含み得る（５０８）。

【００４９】

[0052] いくつかの例では、ベクトルをスケーリングすることは、ＡＬＵ２４によって、乗算演算または除算演算を実行することなしにベクトルをスケーリングすることをさらに含み得る。いくつかの例では、ベクトルの成分の各々が浮動小数点数であり得、ここにおいて、浮動小数点数が、符号ビット、仮数、および指数として表され得る。いくつかの例では、ベクトルは３次元ベクトルを備え得、ベクトルの成分は、ｘ成分と、ｙ成分と、ｚ成分とを備え得る。いくつかの例では、ベクトルをスケーリングすることが、ＡＬＵ２４によって、第１のクロックサイクルにおいてベクトルのｘ成分の第１の指数からスケーリング値を減算することによってベクトルのｘ成分をスケーリングすることと、ＡＬＵ２４によって、第２のクロックサイクルにおいてベクトルのｙ成分の第２の指数からスケーリング値を減算することによってベクトルのｙ成分をスケーリングすることと、ＡＬＵ２４によって、第３のクロックサイクルにおいてベクトルのｚ成分の第３の指数からスケーリング値を減算することによってベクトルのｚ成分をスケーリングすることとを含み得る。いくつかの例では、プロセスは、スケーリングされたｘ成分と、スケーリングされたｙ成分と、スケーリングされたｚ成分とをメモリ中の連続する記憶ロケーション（consecutive storage locations）に出力することをさらに含み得る。いくつかの例では、ＡＬＵ２４はハードウェアデジタル回路（hardware digital circuit）であり得る。

【００５０】

[0053] いくつかの例では、最大指数に少なくとも部分的に基づいてスケーリング値を決定することは、最大指数になるようにスケーリング値を決定することを含み得る。いくつかの他の例では、最大指数に少なくとも部分的に基づいてスケーリング値を決定することは、最大指数と最大代表指数（maximum representative exponent）とに少なくとも部分的に基づいてスケーリング値を決定することを含み得る。

【００５１】

[0054] １つまたは複数の例では、説明した機能は、ハードウェア、ソフトウェア、フ

ァームウェア、またはそれらの任意の組合せで実装され得る。ソフトウェアで実装される場合、機能は、１つまたは複数の命令またはコードとして、コンピュータ可読媒体上に記憶されるか、またはコンピュータ可読媒体を介して送信され得る。コンピュータ可読媒体は、ある場所から別の場所へのコンピュータプログラムの転送を可能にする任意の媒体を含む、コンピュータデータ記憶媒体または通信媒体を含み得る。データ記憶媒体は、本開示で説明した技法の実装のための命令、コードおよび／またはデータ構造を取り出すために、１つまたは複数のコンピュータあるいは１つまたは複数のプロセッサによってアクセスされ得る、任意の利用可能な媒体であり得る。限定ではなく例として、そのようなコンピュータ可読媒体は、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、ＣＤ－ＲＯＭまたは他の光ディスクストレージ、磁気ディスクストレージまたは他の磁気ストレージデバイス、あるいは命令またはデータ構造の形態の所望のプログラムコードを搬送または記憶するために使用され得、コンピュータによってアクセスされ得る、任意の他の媒体を備えることができる。また、いかなる接続もコンピュータ可読媒体と適切に呼ばれる。たとえば、ソフトウェアが、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線（ＤＳＬ）、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、または他のリモートソースから送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、ＤＳＬ、または赤外線、無線、およびマイクロ波などのワイヤレス技術は、媒体の定義に含まれる。本明細書で使用するディスク（disk）およびディスク（disc）は、コンパクトディスク（disc）（ＣＤ）、レーザーディスク（登録商標）（disc）、光ディスク（disc）、デジタル多用途ディスク（disc）（ＤＶＤ）、フロッピー（登録商標）ディスク（disk）およびＢｌｕ－ｒａｙ（登録商標）ディスク（disc）を含み、ここで、ディスク（disk）は、通常、データを磁気的に再生し、ディスク（disc）は、データをレーザーで光学的に再生する。上記の組合せもコンピュータ可読媒体の範囲に含まれるべきである。

【００５２】

[0055] コードは、１つまたは複数のデジタル信号プロセッサ（ＤＳＰ）、汎用マイクロプロセッサ、特定用途向け集積回路（ＡＳＩＣ）、フィールドプログラマブル論理アレイ（ＦＰＧＡ）、あるいは他の等価な集積またはディスクリート論理回路など、１つまたは複数のプロセッサによって実行され得る。したがって、本明細書で使用する「プロセッサ」および「処理ユニット」という用語は、前述の構造、または本明細書で説明した技法の実装に好適な任意の他の構造のいずれかを指すことがある。さらに、いくつかの態様では、本明細書で説明した機能は、符号化および復号のために構成された専用のハードウェアモジュールおよび／またはソフトウェアモジュール内に与えられるか、あるいは複合コーデックに組み込まれ得る。また、本技法は、１つまたは複数の回路または論理要素で十分に実装され得る。

【００５３】

[0056] 本開示の技法は、ワイヤレスハンドセット、集積回路（ＩＣ）またはＩＣのセット（たとえば、チップセット）を含む、多種多様なデバイスまたは装置において実装され得る。本開示では、開示する技法を実行するように構成されたデバイスの機能的態様を強調するために様々な構成要素、モジュール、またはユニットについて説明したが、それらの構成要素、モジュール、またはユニットを、必ずしも異なるハードウェアユニットによって実現する必要があるとは限らない。むしろ、上記で説明したように、様々なユニットが、好適なソフトウェアおよび／またはファームウェアとともに、上記で説明した１つまたは複数のプロセッサを含めて、コーデックハードウェアユニットにおいて組み合わせられるか、または相互動作ハードウェアユニットの集合によって与えられ得る。

【００５４】

[0057] 様々な例について説明した。これらおよび他の例は以下の特許請求の範囲内にある。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

［Ｃ１］

ベクトルをスケーリングするための方法であって、前記方法は、
少なくとも１つのプロセッサによって、ベクトルの成分を受信することと、ここにおいて、前記ベクトルの前記成分の各々が少なくとも１つの指数を備える、
前記少なくとも１つのプロセッサによって、前記ベクトルの前記成分のそれぞれの指数の中から最大指数を決定することと、
前記最大指数に少なくとも部分的に基づいてスケーリング値を決定することと、
前記少なくとも１つのプロセッサの算術論理ユニット（ＡＬＵ）によって、前記ベクトルの前記成分の前記それぞれの指数の各々から前記スケーリング値を減算することによって、前記ベクトルをスケーリングすることと
を備える、方法。

［Ｃ２］
前記ベクトルの前記成分の各々が浮動小数点数を備え、ここにおいて、前記浮動小数点数が、符号ビット、仮数、および前記指数として表される、Ｃ１に記載の方法。

［Ｃ３］
前記ベクトルが３次元ベクトルを備え、
前記ベクトルの前記成分が、ｘ成分と、ｙ成分と、ｚ成分とを備える、
Ｃ１に記載の方法。

［Ｃ４］
前記ベクトルをスケーリングすることが、
前記ＡＬＵによって、第１のクロックサイクルにおいて前記ベクトルの前記ｘ成分の第１の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｘ成分をスケーリングすることと、
前記ＡＬＵによって、第２のクロックサイクルにおいて前記ベクトルの前記ｙ成分の第２の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｙ成分をスケーリングすることと、
前記ＡＬＵによって、第３のクロックサイクルにおいて前記ベクトルの前記ｚ成分の第３の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｚ成分をスケーリングすることと
をさらに備える、Ｃ３に記載の方法。

［Ｃ５］
前記スケーリングされたｘ成分と、前記スケーリングされたｙ成分と、前記スケーリングされたｚ成分とをメモリ中の連続する記憶ロケーションに出力すること
をさらに備える、Ｃ４に記載の方法。

［Ｃ６］
前記ＡＬＵがハードウェアデジタル回路を備える、Ｃ１に記載の方法。

［Ｃ７］
前記最大指数に少なくとも部分的に基づいてスケーリング値を決定することが、
前記最大指数になるように前記スケーリング値を決定することを備える、Ｃ１に記載の方法。

［Ｃ８］
前記最大指数に少なくとも部分的に基づいてスケーリング値を決定することが、
前記最大指数と最大代表指数とに少なくとも部分的に基づいて前記スケーリング値を決定することを備える、Ｃ１に記載の方法。

［Ｃ９］
ベクトルをスケーリングするための装置であって、前記装置は、
ベクトルの成分を記憶するように構成されたメモリと、ここにおいて、前記ベクトルの前記成分の各々が少なくとも１つの指数を備える、
前記ベクトルの前記成分のそれぞれの指数の中から最大指数を決定することと、
前記最大指数に少なくとも部分的に基づいてスケーリング値を決定することと
を行うように構成された少なくとも１つのプロセッサと、

前記ベクトルの前記成分の前記それぞれの指数の各々から前記スケーリング値を減算することによって、前記ベクトルをスケーリングするように構成された算術論理ユニット（ＡＬＵ）と
を備える、装置。
［Ｃ１０］
前記ベクトルの前記成分の各々が浮動小数点数を備え、ここにおいて、前記浮動小数点数が、符号ビット、仮数、および前記指数として表される、Ｃ９に記載の装置。
［Ｃ１１］
前記ベクトルが３次元ベクトルを備え、
前記ベクトルの前記成分が、ｘ成分と、ｙ成分と、ｚ成分とを備える、
Ｃ９に記載の装置。
［Ｃ１２］
前記ＡＬＵが、
第１のクロックサイクルにおいて前記ベクトルの前記ｘ成分の第１の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｘ成分をスケーリングすることと、
第２のクロックサイクルにおいて前記ベクトルの前記ｙ成分の第２の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｙ成分をスケーリングすることと、
第３のクロックサイクルにおいて前記ベクトルの前記ｚ成分の第３の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｚ成分をスケーリングすることと
を行うように構成された、Ｃ１１に記載の装置。
［Ｃ１３］
前記ＡＬＵが、
前記スケーリングされたｘ成分と、前記スケーリングされたｙ成分と、前記スケーリングされたｚ成分とを前記メモリ中の連続する記憶ロケーションに出力することを行うように構成された、Ｃ１２に記載の装置。
［Ｃ１４］
前記ＡＬＵがハードウェアデジタル回路を備える、Ｃ９に記載の装置。
［Ｃ１５］
前記少なくとも１つのプロセッサが、
前記最大指数になるように前記スケーリング値を決定することを行うように構成された、Ｃ９に記載の装置。
［Ｃ１６］
前記少なくとも１つのプロセッサが、
前記最大指数と最大代表指数とに少なくとも部分的に基づいて前記スケーリング値を決定することを行うように構成された、Ｃ９に記載の装置。
［Ｃ１７］
ベクトルをスケーリングするための装置であって、前記装置は、
ベクトルの成分を受信するための手段と、ここにおいて、前記ベクトルの前記成分の各々が少なくとも１つの指数を備える、
前記ベクトルの前記成分のそれぞれの指数の中から最大指数を決定するための手段と、
前記最大指数に少なくとも部分的に基づいてスケーリング値を決定するための手段と、
前記ベクトルの前記成分の前記それぞれの指数の各々から前記スケーリング値を減算することによって、前記ベクトルをスケーリングするための手段と
を備える、装置。
［Ｃ１８］
前記ベクトルの前記成分の各々が浮動小数点数を備え、ここにおいて、前記浮動小数点数が、符号ビット、仮数、および前記指数として表される、Ｃ１７に記載の装置。

［Ｃ１９］
前記ベクトルが３次元ベクトルを備え、
前記ベクトルの前記成分が、ｘ成分と、ｙ成分と、ｚ成分とを備える、
Ｃ１８に記載の装置。
［Ｃ２０］
前記ベクトルをスケーリングするための前記手段が、
第１のクロックサイクルにおいて前記ベクトルの前記ｘ成分の第１の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｘ成分をスケーリングするための手段と、
第２のクロックサイクルにおいて前記ベクトルの前記ｙ成分の第２の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｙ成分をスケーリングするための手段と、
第３のクロックサイクルにおいて前記ベクトルの前記ｚ成分の第３の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｚ成分をスケーリングするための手段と
を備える、Ｃ１９に記載の装置。
［Ｃ２１］
前記ベクトルをスケーリングするための前記手段が、
前記スケーリングされたｘ成分と、前記スケーリングされたｙ成分と、前記スケーリングされたｚ成分とをメモリ中の連続する記憶ロケーションに出力するための手段をさらに備える、Ｃ２０に記載の装置。
［Ｃ２２］
前記最大指数に少なくとも部分的に基づいてスケーリング値を決定するための前記手段が、
前記最大指数になるように前記スケーリング値を決定するための手段を備える、Ｃ１７に記載の装置。
［Ｃ２３］
前記最大指数に少なくとも部分的に基づいてスケーリング値を決定するための前記手段が、
前記最大指数と最大代表指数とに少なくとも部分的に基づいて前記スケーリング値を決定するための手段を備える、Ｃ１７に記載の装置。
［Ｃ２４］
実行されたとき、１つまたは複数のプログラマブルプロセッサに、
ベクトルの成分を受信することと、ここにおいて、前記ベクトルの前記成分の各々が少なくとも１つの指数を備える、
前記ベクトルの前記成分のそれぞれの指数の中から最大指数を決定することと、
前記最大指数に少なくとも部分的に基づいてスケーリング値を決定することと、
前記ベクトルの前記成分の前記それぞれの指数の各々から前記スケーリング値を減算することによって、前記ベクトルをスケーリングすることと
を行わせる命令を記憶するコンピュータ可読記憶媒体。
［Ｃ２５］
前記ベクトルの前記成分の各々が浮動小数点数を備え、ここにおいて、前記浮動小数点数が、符号ビット、仮数、および前記指数として表される、Ｃ２４に記載のコンピュータ可読記憶媒体。
［Ｃ２６］
前記ベクトルが３次元ベクトルを備え、
前記ベクトルの前記成分が、ｘ成分と、ｙ成分と、ｚ成分とを備える、
Ｃ２４に記載のコンピュータ可読記憶媒体。
［Ｃ２７］
前記命令が、前記１つまたは複数のプログラマブルプロセッサに、

第１のクロックサイクルにおいて前記ベクトルの前記ｘ成分の第１の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｘ成分をスケーリングすることと、
第２のクロックサイクルにおいて前記ベクトルの前記ｙ成分の第２の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｙ成分をスケーリングすることと、
第３のクロックサイクルにおいて前記ベクトルの前記ｚ成分の第３の指数から前記スケーリング値を減算することによって、前記ベクトルの前記ｚ成分をスケーリングすることと
をさらに行わせる、Ｃ２６に記載のコンピュータ可読記憶媒体。

［Ｃ２８］
前記命令が、前記１つまたは複数のプログラマブルプロセッサに、
前記スケーリングされたｘ成分と、前記スケーリングされたｙ成分と、前記スケーリングされたｚ成分とをメモリ中の連続する記憶ロケーションに出力することをさらに行わせる、Ｃ２７に記載のコンピュータ可読記憶媒体。

［Ｃ２９］
前記命令が、前記１つまたは複数のプログラマブルプロセッサに、
前記最大指数になるように前記スケーリング値を決定することをさらに行わせる、Ｃ２４に記載のコンピュータ可読記憶媒体。

［Ｃ３０］
前記命令が、前記１つまたは複数のプログラマブルプロセッサに、
前記最大指数と最大代表指数とに少なくとも部分的に基づいて前記スケーリング値を決定すること
をさらに行わせる、Ｃ２４に記載のコンピュータ可読記憶媒体。

【図１】

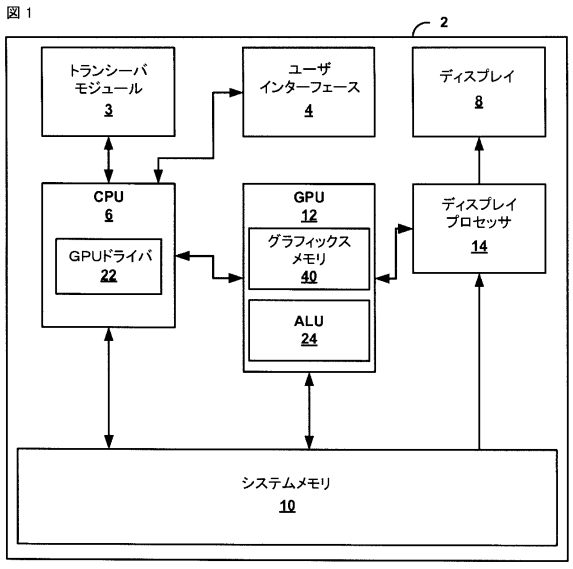


FIG. 1

【図２】

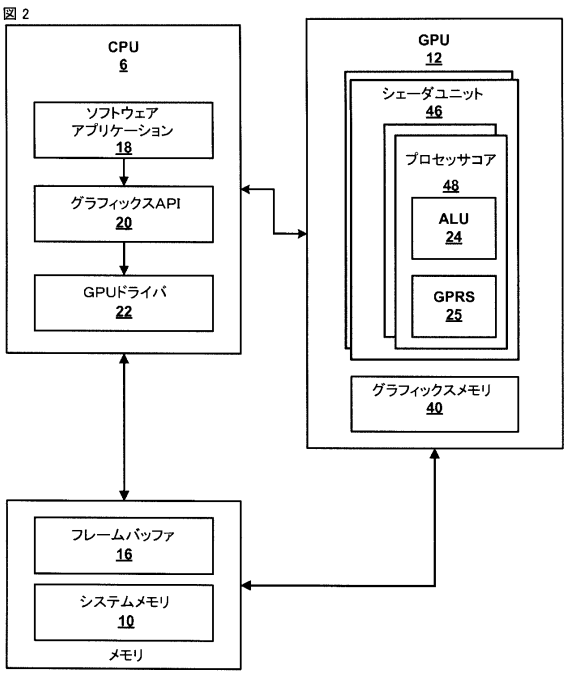


FIG. 2

【図３】

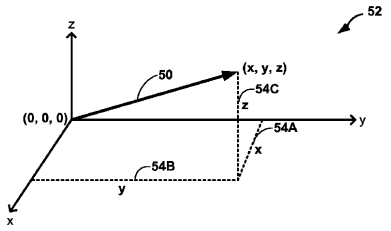


FIG. 3

【図４】

図４

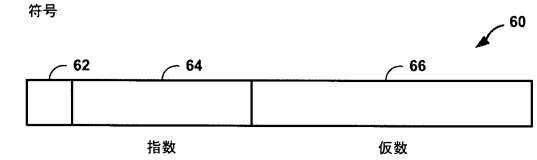


FIG. 4

【図５】

図５

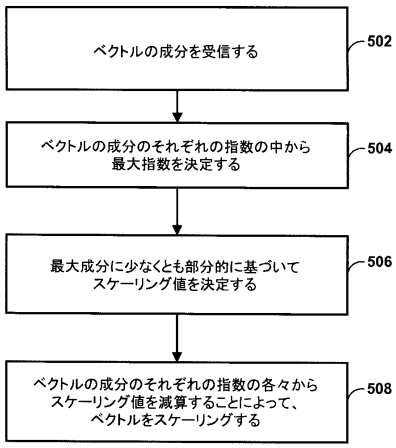


FIG. 5

フロントページの続き

(72)発明者　チェン、リン
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５
(72)発明者　グルバー、アンドリュー・イバン
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５
(72)発明者　ジャオ、グオファン
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５
(72)発明者　ホ、チェンテ
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５
(72)発明者　アルゲーデ、プラモド・バサント
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５

審査官　佐賀野　秀一

(56)参考文献　特開平１１－０２４８８７（ＪＰ，Ａ）
特開平０９－２１２３６２（ＪＰ，Ａ）
特開２００３－１７７９６０（ＪＰ，Ａ）
特開平０８－１０１９１９（ＪＰ，Ａ）
特開２００２－１４９３９７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　７／３８－　７／５３７
Ｇ０６Ｆ　　７／５７－　７／５７５
Ｇ０６Ｆ　　７／７４－　７／７８
Ｇ０６Ｆ　　９／３０－　９／３８