

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6133490号  
(P6133490)

(45) 発行日 平成29年5月24日(2017.5.24)

(24) 登録日 平成29年4月28日(2017.4.28)

(51) Int.Cl.	F 1	
G 0 6 T 15/00	(2011.01)	G 0 6 T 15/00
G 0 9 G 5/00	(2006.01)	G 0 9 G 5/00
G 0 9 G 5/36	(2006.01)	G 0 9 G 5/00
G 0 9 G 5/395	(2006.01)	G 0 9 G 5/00
G 0 6 F 3/14	(2006.01)	G 0 9 G 5/36

請求項の数 42 (全 54 頁) 最終頁に続く

(21) 出願番号	特願2016-507546 (P2016-507546)	(73) 特許権者	595020643 クアアルコム・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
(86) (22) 出願日	平成26年3月17日(2014.3.17)	(74) 代理人	100108855 弁理士 蔵田 昌俊
(65) 公表番号	特表2016-522474 (P2016-522474A)	(74) 代理人	100109830 弁理士 福原 淑弘
(43) 公表日	平成28年7月28日(2016.7.28)	(74) 代理人	100158805 弁理士 井関 守三
(86) 國際出願番号	PCT/US2014/030447	(74) 代理人	100194814 弁理士 奥村 元宏
(87) 國際公開番号	W02014/168740		
(87) 國際公開日	平成26年10月16日(2014.10.16)		
審査請求日	平成28年11月1日(2016.11.1)		
(31) 優先権主張番号	61/811,056		
(32) 優先日	平成25年4月11日(2013.4.11)		
(33) 優先権主張国	米国(US)		
(31) 優先権主張番号	14/044,396		
(32) 優先日	平成25年10月2日(2013.10.2)		
(33) 優先権主張国	米国(US)		

早期審査対象出願

最終頁に続く

(54) 【発明の名称】 タイルベースのレンダリングのためのインストラフレームタイムスタンプ

## (57) 【特許請求の範囲】

## 【請求項 1】

1つまたは複数のプロセッサによって、中央処理装置(CPU)上で実行するグラフィックスアプリケーションからのタイムスタンプ要求を処理することと、

前記1つまたは複数のプロセッサのうちの少なくとも1つを用いて、前記タイムスタンプ要求に応答して複数のビンごとのタイムスタンプ要求を生成することと、

グラフィックス処理装置(GPU)を用いて、グラフィックスフレームに関するタイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復中に前記複数のビンごとのタイムスタンプ要求に応答して複数のビンごとのタイムスタンプ値を生成することと、

前記1つまたは複数のプロセッサのうちの少なくとも1つを用いて、前記複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成することと

を備え、前記タイムスタンプ値を生成することは、前記複数のビンごとのタイムスタンプ値のうちの少なくとも2つのビンごとのタイムスタンプ値の関数を実行することを備え、

前記関数を実行することは、初期のレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記GPUによって生成される初期の基準タイムスタンプを、前記グラフィックスフレームをレンダリングするために使用される2つ以上のレンダリングパス反復に関するそれぞれの基準タイムスタンプ値とそれぞれのビンごとのタイムスタンプ値との差の合計に加えることを備える、

10

20

方法。

**【請求項 2】**

前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングする、請求項 1 に記載の方法。

**【請求項 3】**

前記複数のビンごとのタイムスタンプ値を生成することは、

前記複数のレンダリングパス反復のうちの第 1 のレンダリングパス反復中に第 1 のビンごとのタイムスタンプ値を生成することと、

前記複数のレンダリングパス反復のうちの第 2 のレンダリングパス反復中に第 2 のビンごとのタイムスタンプ値を生成することと

を備え、前記第 2 のレンダリングパス反復は、前記第 1 のレンダリングパス反復とは異なり、前記少なくとも 2 つのビンごとのタイムスタンプ値は、前記第 1 および第 2 のビンごとのタイムスタンプ値を備える、

請求項 1 に記載の方法。

**【請求項 4】**

前記レンダリングパス反復のうちのそれぞれの 1 つ中に、および前記レンダリングパス反復のうちの前記それぞれの 1 つに関して何らかのプリミティブをレンダリングする前に、前記 G P U を用いて、前記第 1 または第 2 のビンごとのタイムスタンプ値とそれぞれ関連する第 1 または第 2 の基準タイムスタンプ値を生成することをさらに備える、請求項 3 に記載の方法。

**【請求項 5】**

前記関数を実行することは、次式を解くことを備え、

**【数 1】**

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y))$$

Value は、前記タイムスタンプ値であり、 TSV(y) は、第 y のレンダリングパス反復中に生成される前記それぞれのビンごとのタイムスタンプ値であり、 TSVGPU(y) は、前記第 y のレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記 G P U によって生成される前記それぞれの基準タイムスタンプ値であり、 N は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の総数である、請求項 1 に記載の方法。

**【請求項 6】**

前記タイムスタンプ要求が前記グラフィックスフレームに関して実行されるコマンドの順序付けられたシーケンス内の少なくとも 2 つの描画呼出しコマンドの間に位置決めされていることに応答して前記タイムスタンプ値を生成することをさらに備える、請求項 1 に記載の方法。

**【請求項 7】**

前記 G P U によって、前記グラフィックスフレームに関するタイルベースのレンダリングを実行している間に前記複数のレンダリングパス反復を実行することをさらに備え、

前記少なくとも 2 つのビンごとのタイムスタンプ値は、第 1 のビンごとのタイムスタンプ値と第 2 のビンごとのタイムスタンプ値とを備え、

前記第 1 のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 1 のレンダリングパス反復中に実行される少なくとも 2 つのビンごとの描画呼出しの実行の間に発生する時点を示し、前記第 1 のレンダリングパス反復中に実行される前記少なくとも 2 つのビンごとの描画呼出しの各々は、前記少なくとも 2 つの描画呼出しコマンドのうちのそれぞれの 1 つに関連し、

前記第 2 のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの

10

20

30

40

50

第2のレンダリングパス反復中に実行される少なくとも2つのビンごとの描画呼出しの実行の間に発生する時点を示し、前記第2のレンダリングパス反復中に実行される前記少なくとも2つのビンごとの描画呼出しの各々は、前記少なくとも2つの描画呼出しコマンドのうちのそれぞれの1つに関連し、前記第2のレンダリングパス反復は、前記第1のレンダリングパス反復とは異なる、

請求項6に記載の方法。

【請求項8】

複数のビンごとのタイムスタンプ要求のうちのそれぞれの1つに応答して前記複数のビンごとのタイムスタンプ値の各々を生成することと、前記ビンごとのタイムスタンプ要求の各々は、複数のコマンドストリームのうちのそれぞれの1つ内に配置され、

10

前記GPUによって、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの1つ中に前記コマンドストリームの各々を実行することと

をさらに備え、各レンダリングパス反復は、レンダターゲットの複数の副領域のうちの1つをレンダリングする、

請求項1に記載の方法。

【請求項9】

前記タイムスタンプ値を生成することは、

前記複数のビンごとのタイムスタンプ値に基づいて複数のタイムスタンプ値を生成すること

20

を備え、前記タイムスタンプ値の各々は、前記グラフィックスフレームに関して実行されるコマンドの順序付けられたシーケンス内に含まれる複数のタイムスタンプ要求のうちのそれぞれの1つに対応し、前記タイムスタンプ要求の各々は、前記グラフィックスアプリケーションによって要求される、

請求項1に記載の方法。

【請求項10】

前記タイムスタンプ要求のうちの少なくとも2つは、前記グラフィックスフレームに関して実行されるコマンドの前記順序付けられた前記順序付けられたシーケンス内で連続する描画呼出しコマンドのそれぞれの対の間に位置決めされ、

前記複数のタイムスタンプ値を生成することは、コマンドの前記順序付けられたシーケンス内の前記タイムスタンプ要求について返される前記タイムスタンプ値が、コマンドの前記順序付けられたシーケンスの始めからコマンドの前記順序付けられたシーケンスの終りまで値において単調に増加するように、前記ビンごとのタイムスタンプ値に基づいて前記複数のタイムスタンプ値を生成することを備える、

30

請求項9に記載の方法。

【請求項11】

前記複数のタイムスタンプ値は、前記グラフィックスフレームの前記レンダリング中に実行するのにコマンドの前記順序付けられたシーケンス内の描画呼出しコマンドによって費やされる時間の相対的な長さを示す、請求項9に記載の方法。

【請求項12】

前記複数のタイムスタンプ値を生成することは、次式に基づいて前記複数のタイムスタンプ値を生成することを備え、

【数2】

$$Value(x) = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(x, y) - TSVGPU(y))$$

V a l u e ( x )は、コマンドの前記順序付けられたシーケンス内の第xのタイムスタンプ要求に応答して生成されるタイムスタンプのタイムスタンプ値であり、TSV ( x , y )は、前記第yのレンダリングパス反復中に生成され、コマンドの前記順序付けられたシ

50

一ケンス内の前記第xのタイムスタンプ要求に対応する、ビンごとのタイムスタンプ値であり、T S V G P U(y)は、前記第yのレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記G P Uによって生成される基準タイムスタンプ値であり、Nは、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、請求項9に記載の方法。

#### 【請求項13】

前記タイムスタンプ値を生成することは、

前記ビンごとのタイムスタンプ要求の各々を複数のコマンドストリームのうちのそれぞれの1つ内に配置することと、

前記G P Uによって、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの1つ中に前記コマンドストリームの各々を実行することと、各レンダリングパス反復は、レンダターゲットの複数の副領域のうちの1つをレンダリングする、10

前記コマンドストリーム内に配置された前記ビンごとのタイムスタンプ要求に応答して前記G P Uによって生成された前記ビンごとのタイムスタンプ値に基づいて前記タイムスタンプ値を生成することと

を備える、請求項1に記載の方法。

#### 【請求項14】

前記1つまたは複数のプロセッサは、前記中央処理装置(C P U)を備える、請求項1に記載の方法。20

#### 【請求項15】

前記1つまたは複数のプロセッサは、前記G P Uを備える、請求項1に記載の方法。

#### 【請求項16】

中央処理装置(C P U)上で実行するグラフィックスアプリケーションからのタイムスタンプ要求を処理し、前記タイムスタンプ要求に応答して複数のビンごとのタイムスタンプ要求を生成するように構成された1つまたは複数のプロセッサと、

前記1つまたは複数のプロセッサのうちの少なくとも1つに結合されたグラフィックス処理装置(G P U)と、

を備え、

前記G P Uは、グラフィックスフレームに関するタイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復中に前記複数のビンごとのタイムスタンプ要求に応答して複数のビンごとのタイムスタンプ値を生成するように構成され、30

前記1つまたは複数のプロセッサのうちの少なくとも1つは、前記複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成するようにさらに構成され、

前記タイムスタンプ値を生成するために、前記1つまたは複数のプロセッサは、前記複数のビンごとのタイムスタンプ値のうちの少なくとも2つのビンごとのタイムスタンプ値の関数を実行するように構成され、

前記関数を実行するために、前記1つまたは複数のプロセッサは、初期のレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記G P Uによって生成される初期の基準タイムスタンプを、前記グラフィックスフレームをレンダリングするために使用される2つ以上のレンダリングパス反復に関するそれぞれの基準タイムスタンプ値とそれぞれのビンごとのタイムスタンプ値との差の合計に加えるように構成される、40

デバイス。

#### 【請求項17】

前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれの1つをレンダリングする、請求項16に記載のデバイス。

#### 【請求項18】

前記タイムスタンプを生成するために、前記1つまたは複数のプロセッサは、

前記複数のレンダリングパス反復のうちの第1のレンダリングパス反復中に第1のビン50

ごとのタイムスタンプ値を生成することと、

前記複数のレンダリングパス反復のうちの第2のレンダリングパス反復中に第2のピンごとのタイムスタンプ値を生成することと

を行うように構成され、前記第2のレンダリングパス反復は、前記第1のレンダリングパス反復とは異なり、前記少なくとも2つのピンごとのタイムスタンプ値は、前記第1および第2のピンごとのタイムスタンプ値を備える、

請求項16に記載のデバイス。

#### 【請求項19】

前記は、前記レンダリングパス反復のうちのそれぞれの1つ中に、および前記レンダリングパス反復のうちの前記それぞれの1つに関して何らかのプリミティブをレンダリングする前に、前記第1または第2のピンごとのタイムスタンプ値とそれぞれ関連する第1または第2の基準タイムスタンプ値を生成するようにさらに構成される、請求項18に記載のデバイス。10

#### 【請求項20】

前記関数を実行するために、前記1つまたは複数のプロセッサは、次式を解くようにさらに構成され、

#### 【数3】

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y))$$
20

Valueは、前記タイムスタンプ値であり、TSV(y)は、第yのレンダリングパス反復中に生成される前記それぞれのピンごとのタイムスタンプ値であり、TSVGPU(y)は、前記第yのレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記GPUによって生成される前記それぞれの基準タイムスタンプ値であり、Nは、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の総数である、請求項16に記載のデバイス。

#### 【請求項21】

前記1つまたは複数のプロセッサは、前記タイムスタンプ要求が前記グラフィックスフレームに関して実行されるコマンドの順序付けられたシーケンス内の少なくとも2つの描画呼出しコマンドの間に位置決めされたことに応答して前記タイムスタンプ値を生成するようにさらに構成される、請求項16に記載のデバイス。30

#### 【請求項22】

前記GPUは、前記グラフィックスフレームに関するタイルベースのレンダリングを実行している間に複数のレンダリングパス反復を実行するように構成され、

前記少なくとも2つのピンごとのタイムスタンプ値は、第1のピンごとのタイムスタンプ値と第2のピンごとのタイムスタンプ値とを備え、

前記第1のピンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第1のレンダリングパス反復中に実行される少なくとも2つのピンごとの描画呼出しの実行の間に発生する時点を示し、前記第1のレンダリングパス反復中に実行される前記少なくとも2つのピンごとの描画呼出しの各々は、前記少なくとも2つの描画呼出しコマンドのうちのそれぞれの1つに関連し、40

前記第2のピンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第2のレンダリングパス反復中に実行される少なくとも2つのピンごとの描画呼出しの実行の間に発生する時点を示し、前記第2のレンダリングパス反復中に実行される前記少なくとも2つのピンごとの描画呼出しの各々は、前記少なくとも2つの描画呼出しコマンドのうちのそれぞれの1つに関連し、前記第2のレンダリングパス反復は、前記第1のレンダリングパス反復とは異なる、

請求項21に記載のデバイス。50

**【請求項 2 3】**

前記 1 つまたは複数のプロセッサは、複数のビンごとのタイムスタンプ要求のうちのそれぞれの 1 つに応答して前記複数のビンごとのタイムスタンプ値の各々を生成するように構成され、前記ビンごとのタイムスタンプ要求の各々は、複数のコマンドストリームのうちのそれぞれの 1 つ内に配置され、前記 G P U は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つ中に前記コマンドストリームの各々を実行するように構成され、各レンダリングパス反復は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングするように構成される、請求項 1 6 に記載のデバイス。

**【請求項 2 4】**

10

前記 1 つまたは複数のプロセッサは、前記複数のビンごとのタイムスタンプ値に基づいて複数のタイムスタンプ値を生成するようにさらに構成され、前記タイムスタンプ値の各々は、前記グラフィックスフレームに関して実行されるコマンドの順序付けられたシーケンス内に含まれる複数のタイムスタンプ要求のうちのそれぞれの 1 つに対応し、前記タイムスタンプ要求の各々は、前記グラフィックスアプリケーションによって要求される、請求項 1 6 に記載のデバイス。

**【請求項 2 5】**

前記タイムスタンプ要求のうちの少なくとも 2 つは、前記グラフィックスフレームに関して実行されるコマンドの前記順序付けられた前記順序付けられたシーケンス内で連続する描画呼出しコマンドのそれぞれの対の間に位置決めされ、

20

前記 1 つまたは複数のプロセッサは、コマンドの前記順序付けられたシーケンス内の前記タイムスタンプ要求について返される前記タイムスタンプ値が、コマンドの前記順序付けられたシーケンスの始めからコマンドの前記順序付けられたシーケンスの終りまで値において単調に増加するように、前記ビンごとのタイムスタンプ値に基づいて前記複数のタイムスタンプ値を生成するようにさらに構成される、

請求項 2 4 に記載のデバイス。

**【請求項 2 6】**

前記複数のタイムスタンプ値は、前記グラフィックスフレームの前記レンダリング中に実行するのにコマンドの前記順序付けられたシーケンス内の描画呼出しコマンドによって費やされる時間の相対的な長さを示す、請求項 2 4 に記載のデバイス。

30

**【請求項 2 7】**

前記 1 つまたは複数のプロセッサは、次式に基づいて前記複数のタイムスタンプ値を生成するようにさらに構成され、

**【数 4】**

$$Value(x) = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(x, y) - TSVGPU(y))$$

`Value(x)` は、コマンドの前記順序付けられたシーケンス内の第  $x$  のタイムスタンプ要求に応答して生成されるタイムスタンプのタイムスタンプ値であり、`TSV(x, y)` は、前記第  $y$  のレンダリングパス反復中に生成され、コマンドの前記順序付けられたシーケンス内の前記第  $x$  のタイムスタンプ要求に対応する、ビンごとのタイムスタンプ値であり、`TSVGPU(y)` は、前記第  $y$  のレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記 G P U によって生成される基準タイムスタンプ値であり、 $N$  は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、請求項 2 4 に記載のデバイス。

40

**【請求項 2 8】**

前記 1 つまたは複数のプロセッサは、

前記ビンごとのタイムスタンプ要求の各々を複数のコマンドストリームのうちのそれぞ

50

れの 1 つ内に配置し、前記コマンドストリームの各々は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つ中に前記 G P U によって実行されるように構成され、各レンダリングパス反復は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングするように構成され、

前記 G P U に前記コマンドストリームを実行させ、

前記コマンドストリーム内に配置された前記ビンごとのタイムスタンプ要求に応答して前記 G P U によって生成された前記ビンごとのタイムスタンプ値に基づいて前記タイムスタンプ値を生成する、

ようさらに構成される、請求項 1 6 に記載のデバイス。

**【請求項 2 9】**

前記 1 つまたは複数のプロセッサは、前記 C P U の一部から少なくとも構成される、請求項 1 6 に記載のデバイス。

**【請求項 3 0】**

前記 1 つまたは複数のプロセッサは、前記 G P U を備える、請求項 1 6 に記載のデバイス。

**【請求項 3 1】**

前記デバイスは、ワイヤレス通信デバイスを備える、請求項 1 6 に記載のデバイス。

**【請求項 3 2】**

前記デバイスは、携帯電話送受話器を備える、請求項 1 6 に記載のデバイス。

**【請求項 3 3】**

中央処理装置 ( C P U ) 上で実行するグラフィックスアプリケーションからのタイムスタンプ要求を処理するための手段と、

前記タイムスタンプ要求に応答して複数のビンごとのタイムスタンプ要求を生成するための手段と、

グラフィックスフレームに関するタイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復中に前記複数のビンごとのタイムスタンプ要求に応答して複数のビンごとのタイムスタンプ値を生成するための手段と、

前記複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成するための手段と、

を備え、前記タイムスタンプ値を生成するための前記手段は、前記複数のビンごとのタイムスタンプ値のうちの少なくとも 2 つのビンごとのタイムスタンプ値の関数を実行するための手段を備え、

前記関数を実行するための前記手段は、初期のレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記 G P U によって生成される初期の基準タイムスタンプを、前記グラフィックスフレームをレンダリングするために使用される 2 つ以上のレンダリングパス反復に関するそれぞれの基準タイムスタンプ値とそれぞれのビンごとのタイムスタンプ値との差の合計に加えるための手段を備える、

装置。

**【請求項 3 4】**

前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングする、請求項 3 3 に記載の装置。

**【請求項 3 5】**

前記複数のビンごとのタイムスタンプ値を生成するための前記手段は、

前記複数のレンダリングパス反復のうちの第 1 のレンダリングパス反復中に第 1 のビンごとのタイムスタンプ値を生成するための手段と、

前記複数のレンダリングパス反復のうちの第 2 のレンダリングパス反復中に第 2 のビンごとのタイムスタンプ値を生成するための手段と

を備え、前記第 2 のレンダリングパス反復は、前記第 1 のレンダリングパス反復とは異なり、前記少なくとも 2 つのビンごとのタイムスタンプ値は、前記第 1 および第 2 のビンごとのタイムスタンプ値を備える、

10

20

30

40

50

請求項 3 3 に記載の装置。

【請求項 3 6】

前記関数を実行するための前記手段は、次式を解くための手段を備え、  
【数 5】

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y))$$

V a l u e は、前記タイムスタンプ値であり、 T S V ( y ) は、第 y のレンダリングパス  
反復中に生成される前記それぞれのビンごとのタイムスタンプ値であり、 T S V G P U ( 10  
y ) は、前記第 y のレンダリングパス反復に関する何らかのプリミティブをレンダリング  
する前に前記 G P U によって生成される前記それぞれの基準タイムスタンプ値であり、 N  
は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反  
復の回数である、請求項 3 3 に記載の装置。

【請求項 3 7】

前記タイムスタンプ値を生成するための前記手段は、前記 C P U と前記 G P U とのうち  
の少なくとも 1 つを備える、請求項 3 3 に記載の装置。

【請求項 3 8】

1 つまたは複数のプロセッサによって実行された時に、前記 1 つまたは複数のプロセッ  
サに、  
20

中央処理装置 ( C P U ) 上で実行するグラフィックスアプリケーションからのタイムス  
タンプ要求を処理させ、

前記タイムスタンプ要求に応答して複数のビンごとのタイムスタンプ要求を生成させ、  
グラフィックスフレームに関するタイルベースのレンダリングを実行している間に発生  
する複数のレンダリングパス反復中に前記複数のビンごとのタイムスタンプ要求に応答し  
て複数のビンごとのタイムスタンプ値を生成させ、

前記複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成  
させる

命令を備え、  
30

前記 1 つまたは複数のプロセッサに、前記タイムスタンプ値を生成させる命令は、実行  
された時に、前記 1 つまたは複数のプロセッサに、前記複数のビンごとのタイムスタン  
プ値のうちの少なくとも 2 つのビンごとのタイムスタンプ値の関数を実行させる命令を備  
え、

前記 1 つまたは複数のプロセッサに、前記関数を実行させる命令は、実行された時に、  
前記 1 つまたは複数のプロセッサに、初期のレンダリングパス反復に関する何らかのプリ  
ミニティブをレンダリングする前に前記 G P U によって生成される初期の基準タイムстан  
プを、前記グラフィックスフレームをレンダリングするために使用される 2 つ以上のレン  
ダリングパス反復に関するそれぞれの基準タイムスタンプ値とそれぞれのビンごとのタイ  
ムスタンプ値との差の合計に加えさせる命令を備える、  
40

非一時的コンピュータ可読記憶媒体。

【請求項 3 9】

前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれ  
ぞれの 1 つをレンダリングするように構成される、請求項 3 8 に記載の非一時的コンピュー  
タ可読記憶媒体。

【請求項 4 0】

前記 1 つまたは複数のプロセッサに、前記複数のビンごとのタイムスタンプ値を生成さ  
せる前記命令は、

実行された時に、前記 1 つまたは複数のプロセッサに、

前記複数のレンダリングパス反復のうちの第 1 のレンダリングパス反復中に第 1 のビン  
50

ごとのタイムスタンプ値を生成させる命令と、

実行された時に、前記1つまたは複数のプロセッサに、前記複数のレンダリングパス反復のうちの第2のレンダリングパス反復中に第2のビンごとのタイムスタンプ値を生成させる命令と

を備え、前記第2のレンダリングパス反復は、前記第1のレンダリングパス反復とは異なり、前記少なくとも2つのビンごとのタイムスタンプ値は、前記第1および第2のビンごとのタイムスタンプ値を備える、

請求項3 8に記載の非一時的コンピュータ可読記憶媒体。

**【請求項4 1】**

前記1つまたは複数のプロセッサに、前記関数を実行させる前記命令は、実行された時に、前記1つまたは複数のプロセッサに、次式を解かせる命令を備え、

**【数6】**

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y))$$

V a l u e は、前記タイムスタンプ値であり、 T S V ( y ) は、第 y のレンダリングパス反復中に生成される前記それぞれのビンごとのタイムスタンプ値であり、 T S V G P U ( y ) は、前記第 y のレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記G P Uによって生成される前記それぞれの基準タイムスタンプ値であり、 N は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、請求項3 8に記載の非一時的コンピュータ可読記憶媒体。

**【請求項4 2】**

前記1つまたは複数のプロセッサは、前記C P Uと前記G P Uとのうちの少なくとも1つを備える、請求項3 8に記載の非一時的コンピュータ可読記憶媒体。

**【発明の詳細な説明】**

**【技術分野】**

**【0 0 0 1】**

[0001]本願は、その内容全体が参照によって本明細書に組み込まれている、2013年4月11日に出願した米国特許仮出願第61/811056号の利益を主張するものである。

**【0 0 0 2】**

[0002]本開示は、グラフィックス処理システムに関し、より具体的には、グラフィックス処理システムでのタイムスタンプの使用に関する。

**【背景技術】**

**【0 0 0 3】**

[0003]コンピューティングデバイスは、しばしば、表示のためのグラフィックスデータのレンダリングを加速するために、グラフィックス処理ユニット ( G P U ) を利用する。そのようなコンピューティングデバイスは、たとえば、コンピュータワークステーション、携帯電話（たとえば、いわゆるスマートフォン）、組込みシステム、パーソナルコンピュータ、タブレットコンピュータ、およびビデオゲーム機を含むことができる。レンダリングは、一般に、1つまたは複数の3次元 ( 3 D ) グラフィックスオブジェクトを含むことができる3Dグラフィックスシーンを2次元 ( 2 D ) のラスタ化された画像データに変換するプロセスを指す。グラフィックスシーンは、各フレームが時間における特定の瞬間のグラフィックスシーンを描く、1つまたは複数のフレームのシーケンスとしてレンダリングされ得る。

**【0 0 0 4】**

[0004] G P U は、3 D グラフィックスシーンのレンダリングに関する少なくとも部分的なハードウェアアクセラレーションを提供するために、3 D レンダリングパイプラインを

10

20

30

40

50

含むことができる。シーン内の3Dグラフィックスオブジェクトは、グラフィックスアプリケーションによって1つまたは複数の3Dグラフィックスプリミティブ（たとえば、点、線、三角形、パッチなど）に分割され得、GPUは、レンダリングされるフレームの各々について、シーンの3Dグラフィックスプリミティブを2Dのラスタ化された画像データに変換することができる。したがって、GPUレンダリングの特定の文脈では、レンダリングは、グラフィックスシーン内の3Dオブジェクトに対応する3Dグラフィックスプリミティブを2Dのラスタ化された画像データに変換するプロセスを指すことができる。

#### 【0005】

[0005]特定のフレームに関して3Dグラフィックスプリミティブをレンダリングするために、ホスト中央処理装置（CPU）上で実行するグラフィックスアプリケーションは、レンダリングされるべきプリミティブに対応するジオメトリデータをGPUアクセス可能メモリ内に配置し、1つまたは複数のGPU状態セットアップコマンドをコマンドストリーム内に配置し、GPUにジオメトリデータに基づいてプリミティブをレンダリングさせる1つまたは複数の描画呼出しをコマンドストリーム内に配置することができる。GPUは、コマンドストリーム内に含まれるコマンドを、コマンドがコマンドストリーム内に配置されている順序で処理し、これによってシーンをレンダリングすることができる。10

#### 【発明の概要】

#### 【0006】

[0006]本開示は、タイルベースのレンダリングを実行するグラフィックス処理システム内でイントラフレームタイムスタンプ要求をサポートする技法を説明するものである。タイルベースのレンダリングは、レンダターゲット（たとえば、フレーム）を複数のビン（たとえば、副領域またはタイル）に分割することと、ビンの各々について別々のレンダリングパス反復を実行することとを含むことができる。イントラフレームタイムスタンプ要求は、レンダリングされるべきグラフィックスフレームに関連するグラフィックスコマンドストリーム内の任意の位置に配置され得るタイムスタンプ要求を指すことができる。タイムスタンプ要求は、タイムスタンプ要求を処理するデバイス（たとえば、GPUまたはCPU）によってタイムスタンプ要求が処理される時間における瞬間を示すタイムスタンプ値の要求を指すことができる。本開示のイントラフレームタイムスタンプ生成技法は、タイルベースのレンダリングを実行している間にグラフィックス処理ユニット（GPU）によって生成される1つまたは複数のビンごとのタイムスタンプ値に基づいて、アプリケーションによって要求されたタイムスタンプ値を生成することができる。アプリケーションによって要求されたタイムスタンプ値を生成するのにビンごとのタイムスタンプ値を使用することは、タイルベースのレンダリングを実行するグラフィックス処理システムによってイントラフレームタイムスタンプをサポートすることを可能にすることができる。2030

#### 【0007】

[0007]一例では、本開示は、1つまたは複数のプロセッサを用いて、グラフィックスフレームに関するタイルベースのレンダリングを実行している間にGPUによって生成される複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成することを含む方法を説明する。タイムスタンプ値は、複数のビンごとのタイムスタンプ値のうちの少なくとも2つのビンごとのタイムスタンプ値の関数とすることができる。40

#### 【0008】

[0008]別の例では、本開示は、グラフィックスフレームに関するタイルベースのレンダリングを実行している間にGPUによって生成される複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成するように構成された1つまたは複数のプロセッサを含むデバイスを説明する。タイムスタンプ値は、複数のビンごとのタイムスタンプ値のうちの少なくとも2つのビンごとのタイムスタンプ値の関数とすることができる。

#### 【0009】

[0009]別の例では、本開示は、GPUを含む装置を説明する。この装置は、グラフィッ50

クスフレームに関するタイルベースのレンダリングを実行している間に G P U によって生成される複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成するための手段をさらに含む。タイムスタンプ値は、複数のビンごとのタイムスタンプ値のうちの少なくとも 2 つのビンごとのタイムスタンプ値の関数とすることができます。

#### 【 0 0 1 0 】

[0010]別の例では、本開示は、実行された時に、1つまたは複数のプロセッサに、グラフィックスフレームに関するタイルベースのレンダリングを実行している間にグラフィックス処理ユニット ( G P U ) によって生成される複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成させる命令を記憶するコンピュータ可読記憶媒体を説明する。タイムスタンプ値は、複数のビンごとのタイムスタンプ値のうちの少なくとも 2 つのビンごとのタイムスタンプ値の関数とすることができます。10

#### 【 0 0 1 1 】

[0011]本開示の1つまたは複数の例の詳細は、添付図面と下の説明とに示される。本開示の他の特徴、目的、および利点は、その説明と図面と特許請求の範囲とから明白になる。。

#### 【 図面の簡単な説明 】

##### 【 0 0 1 2 】

【図1】[0012]本開示のイントラフレームタイムスタンプ生成技法を実施するのに使用され得る例のコンピューティングデバイスを示すブロック図。

【図2】[0013]図1に示されたコンピューティングデバイスの C P U と G P U とメモリとをさらに詳細に示すブロック図。20

【図3】[0014]複数の副領域（たとえば、タイル）に副分割される例のレンダターゲットと、副分割されたレンダターゲット上に表示されるプリミティブの例のセットとを示す概念図。

【図4】[0015]本開示に従ってグラフィックスアプリケーションによって発行される例のコマンドストリームを示す概念図。

【図5】[0016]本開示に従ってレンダリングパスを実行するための例の実行タイムラインを示す概念図。

【図6】[0017]本開示に従って複数のレンダリングパス反復を実行するための例のコマンドストリームを示す概念図。30

【図7】[0018]本開示に従ってタイルベースのレンダリングを実行するグラフィックス処理システム内でイントラフレームタイムスタンプをサポートする例の技法を示す流れ図。

【図8】[0019]本開示に従ってタイルベースのレンダリングを実行するグラフィックス処理システム内でイントラフレームタイムスタンプをサポートする別の例の技法を示す流れ図。

#### 【 発明を実施するための形態 】

##### 【 0 0 1 3 】

[0020]本開示は、タイルベースのレンダリングを実行するグラフィックス処理システム内でイントラフレームタイムスタンプ要求をサポートする技法を説明するものである。タイルベースのレンダリングは、レンダターゲット（たとえば、フレーム）を複数のビン（たとえば、副領域またはタイル）に副分割することと、ビンの各々について別々のレンダリングパス反復を実行することとを含むことができる。イントラフレームタイムスタンプ要求は、レンダリングされるべきグラフィックスフレームに関連するグラフィックスコマンドストリーム内の任意の位置に配置され得るタイムスタンプ要求を指すことができる。タイムスタンプ要求は、タイムスタンプ要求を処理するデバイス（たとえば、G P U または C P U ）によってタイムスタンプ要求が処理される時間における瞬間を示すタイムスタンプ値の要求を指すことができる。本開示のイントラフレームタイムスタンプ生成技法は、タイルベースのレンダリングを実行している間にグラフィックス処理ユニット ( G P U ) によって生成される1つまたは複数のビンごとのタイムスタンプ値に基づいて、アプリケーションによって要求されたタイムスタンプ値を生成することができる。アプリケーシ4050

ヨンによって要求されたタイムスタンプ値を生成するのにビンごとのタイムスタンプ値を使用することは、タイルベースのレンダリングを実行するグラフィックス処理システムによってイントラフレームタイムスタンプをサポートすることを可能にすることができる。

#### 【0014】

[0021] 2つの異なる描画呼出しコマンドの間に配置されたイントラフレームタイムスタンプ要求を発行するグラフィックスアプリケーションは、たとえば、要求に応答して返されるタイムスタンプ値が第1の描画呼出しコマンドの実行と第2の描画呼出しコマンドの実行との間の時刻に対応することを期待することができる。しかしながら、タイルベースのレンダリングシステムは、ある描画呼出しコマンドの実行が同一のグラフィックスフレームに関連する他の描画呼出しコマンドの実行とインターリープされるように、あるグラフィックスフレームに関する描画呼出しコマンドを不連続な形で実行する場合がある。たとえば、タイルベースのレンダリングシステムは、レンダリングされるべきグラフィックスフレームに関連する描画呼出しコマンドの実行を複数のビンごとの描画呼出しに分割し、ビンごとの描画呼出しをビンによって一緒にグループ化し、ビンごとの描画呼出しのグループの各々を、別々のレンダリングパス反復の一部として実行する場合がある。描画呼出しコマンドを実行する、この不連続なインターリープされた形は、タイルベースのレンダリングシステムがイントラフレームタイムスタンプをサポートすることを困難にすること。

#### 【0015】

[0022] 本開示で説明される技法は、タイルベースのレンダリングシステムが、不連続なインターリープされた形で描画呼出しコマンドを実行する場合であっても、タイルベースのレンダリングシステムが、イントラフレームタイムスタンプをサポートすることを可能にすることができる。たとえば、本開示のイントラフレームタイムスタンプ生成技法は、タイルベースのレンダリングを実行している間にGPUによって生成される複数のビンごとのタイムスタンプ値に基づいて、アプリケーションによって要求されたタイムスタンプ値を生成することができる。アプリケーションによって要求されたタイムスタンプ値を生成するのに使用されるビンごとのタイムスタンプ値のうちの少なくとも一部は、異なるレンダリングパス反復の一部として生成され得る。異なるレンダリングパス反復中に生成されるビンごとのタイムスタンプ値を使用することは、グラフィックスフレームのレンダリング中に実行するのに異なる描画呼出しコマンドによって費やされる時間の相対的な長さを少なくともある度合まで反映する、アプリケーションによって要求されたタイムスタンプ値をグラフィックス処理システムが生成することを可能にすることができる。この形で、有用なタイムスタンプ値が、異なる描画呼出しコマンドによって費やされる実行時間の相対的な長さに関するタイミング統計を使用するグラフィックスアプリケーションに、タイルベースのレンダリングシステムによって提供され得る。

#### 【0016】

[0023] 本明細書で使用される時に、アプリケーションによって要求されるタイムスタンプ値は、グラフィックスアプリケーション（たとえば、ソフトウェアスタック内でドライバレベルより上位にあるアプリケーション）によって生成されるタイムスタンプ要求に応答して生成されるタイムスタンプ値を指すことができる。ビンごとのタイムスタンプ値は、特定のビン（たとえば、レンダターゲットの副領域）に関するレンダリングパス反復を実行している間にGPUによって生成されるタイムスタンプ値を指すことができる。グラフィックスアプリケーションによって生成されるタイムスタンプ要求は、アプリケーションによって生成されるタイムスタンプ要求と呼ばれる場合がある。いくつかの場合に、ビンごとのタイムスタンプ要求は、GPUドライバによってまたはGPU（たとえば、ユーザアプリケーションレイヤより下位にあるソフトウェア／ハードウェアレイヤ）によって生成されるタイムスタンプ要求を指すことができる。

#### 【0017】

[0024] いくつかの例では、GPUドライバまたは他のアプリケーションは、コマンドストリーム内で受け取られるアプリケーションによって生成されたタイムスタンプ要求の各

10

20

30

40

50

々について、複数のピンごとのタイムスタンプ要求を生成することができる。ピンごとのタイムスタンプ要求は、G P Uによってサービスされ得、このG P Uは、ピンごとのタイムスタンプ要求のそれぞれを受け取ることに応答して、それぞれのピンごとのタイムスタンプ値を生成することができる。ピンごとのタイムスタンプ値は、G P Uが、タイルベースのレンダリングを実行している時に実行されるコマンドストリーム内でピンごとのタイムスタンプ要求に出会った時刻を示すことができる。アプリケーションによって生成されたタイムスタンプ要求に応答して生成されるタイムスタンプおよびタイムスタンプ値は、それぞれ、アプリケーションによって要求されたタイムスタンプおよびアプリケーションによって要求されたタイムスタンプ値と呼ばれる場合がある。同様に、ピンごとのタイムスタンプ要求に応答して生成されるタイムスタンプおよびタイムスタンプ値は、それぞれ、ピンごとのタイムスタンプおよびピンごとのタイムスタンプ値と呼ばれる場合がある。  
10

#### 【 0 0 1 8 】

[0025]グラフィックスアプリケーション（たとえば、G P Uに1つまたは複数のグラフィックスフレームをレンダリングさせる命令を含む、ホスト中央処理装置（C P U）上で実行されるアプリケーション）は、しばしば、特定のグラフィックスフレームをレンダリングするために複数の描画呼出しコマンドを発行する場合がある。たとえば、G P Uは、通常、実行されるべき描画呼出しコマンドごとにレンダ状態セッティングの単一のセットを伴って、単一のタイプのプリミティブ（たとえば、点、線、三角形、パッチなど）をレンダリングするように構成される。そのような例では、あるフレームをレンダリングするために、複数のタイプのプリミティブが必要である場合、またはそのフレームをレンダリングするために、複数のタイプのレンダ状態が必要である場合に、グラフィックスアプリケーションは、単一のグラフィックスフレームをレンダリングするために複数の描画呼出しコマンドを発行する必要がある可能性がある。  
20

#### 【 0 0 1 9 】

[0026]あるグラフィックスフレームをレンダリングするために複数の描画呼出しコマンドが使用される時に、個々の描画呼出しコマンドまたは描画呼出しコマンドのサブセットの実行に関するタイミング統計を入手するために、グラフィックスアプリケーションは、G P Uによって実行されるべきコマンドストリーム内の描画呼出しコマンドの間にタイムスタンプ要求を配置することができる。個々のグラフィックスフレームをレンダリングするのに使用される描画呼出しコマンドの間に配置されるタイムスタンプ要求は、本明細書では、イントラフレームタイムスタンプ要求と呼ばれる場合があり、そのような要求に応答して生成される対応するタイムスタンプは、イントラフレームタイムスタンプと呼ばれる場合がある。  
30

#### 【 0 0 2 0 】

[0027]グラフィックスアプリケーションは、コマンドストリーム内に配置される各タイムスタンプ要求に応答してタイムスタンプを受け取ることができる。タイムスタンプは、G P Uがタイムスタンプ要求を実行する時刻を指定するタイムスタンプ値を含むことができる。グラフィックスコマンドストリームは、通常、コマンドがコマンドストリーム内に配置される順序でG P Uによって実行されるので、グラフィックスアプリケーションは、タイムスタンプ要求がコマンドストリーム内で2つの隣接する描画呼出しの間に配置される時に、返されるタイムスタンプが、第1の描画呼出しコマンドの実行と第2の描画呼出しコマンドの実行との間に発生する時刻に対応すると期待することができる。  
40

#### 【 0 0 2 1 】

[0028]上で述べた期待を満足するタイムスタンプ値は、グラフィックスアプリケーションが、たとえば様々なタイムスタンプ処理技法を実行することを可能にすることができる。たとえば、そのようなタイムスタンプ値は、コマンドストリーム内で描画呼出しコマンドの前と後とに配置されたタイムスタンプ要求に応答して返されるタイムスタンプ値の間の差をとることによって、描画呼出しコマンドの近似実行時間を決定するのに使用され得る。

#### 【 0 0 2 2 】

[0029] タイルベースのレンダリングは、いくつかの例で、レンダターゲット（たとえば、フレーム）を複数の副領域（たとえば、ピンまたはタイル）に副分割することと、レンダターゲットの副領域の各々について、別々のレンダリングパス反復を含むレンダリングパスを実行することとを含むことができる。別々のレンダリングパス反復を実行するためには、タイルベースのレンダリングシステムは、レンダリングされるべきグラフィックスフレームに関連する描画呼出しコマンドの実行を複数のピンごとの描画呼出しに副分割し、ピンごとの描画呼出しをピンによって一緒にグループ化することができる。ピンごとの描画呼出しのグループの各々は、別々のレンダリングパス反復の一部として実行され得る。

#### 【0023】

[0030] レンダリングされるべきグラフィックスフレームが、複数の描画呼出しを含む場合には、1つの描画呼出しコマンドに関するビンごとの描画呼出しの実行は、同一のグラフィックスフレームに関する他の描画呼出しコマンドに関するビンごとの描画呼出しの実行とインターリープされ得る。しかしながら、上で議論されたように、あるタイプのタイムスタンプ処理技法は、描画呼出しコマンドが、連続的な形で、グラフィックスコマンドがコマンドストリーム内に配置される順序で実行されることを仮定する場合がある。タイルベースのレンダリングを実行する時に発生する描画呼出しコマンドのインターリープされた実行は、有用なイントラフレームタイムスタンプをそのようなタイムスタンプ処理技法に提供することをむずかしくする可能性がある。10

#### 【0024】

[0031] 本開示の技法は、描画呼出しコマンドがタイルベースのレンダリング技法の実行に起因してインターリープされた形で実行される時であっても、イントラフレームタイムスタンプを生成するのに使用され得る。いくつかの例では、本開示に従って生成されるイントラフレームタイムスタンプ値は、コマンドストリーム内の描画呼出しコマンドについて行われた実行時間の相対的な長さを示すタイムスタンプ値を提供すると同時に、連続的な順序通りの描画呼出し処理が実行された場合に入手されるはずのタイムスタンプ値を模倣したまたは近似することができる（描画呼出しが実際にはインターリープされた形で実行される可能性がある場合であっても）。この形で、タイムスタンプ値は、描画呼出しコマンドが、連続的な形で、グラフィックスコマンドがコマンドストリーム内に配置される順序で実行されることを仮定するタイムスタンプ処理技法とともに使用され得る。20

#### 【0025】

[0032] 図1は、本開示のイントラフレームタイムスタンプ生成技法を実施するのに使用され得る例のコンピューティングデバイス2を示すブロック図である。コンピューティングデバイス2は、パーソナルコンピュータ、デスクトップコンピュータ、ラップトップコンピュータ、コンピュータワークステーション、ビデオゲームプラットフォームもしくはビデオゲーム機、ワイヤレス通信デバイス（たとえば、携帯電話、セルラ電話、衛星電話、および／または携帯電話送受話器など）、陸線電話、インターネット電話、ポータブルビデオゲームデバイスもしくは携帯情報端末（PDA）などのハンドヘルドデバイス、パーソナル音楽プレイヤ、ビデオプレイヤ、ディスプレイデバイス、テレビジョン、テレビジョンセットトップボックス、サーバ、中間ネットワークデバイス、メインフレームコンピュータ、または、グラフィカルデータを処理し、および／もしくは表示する任意の他のタイプのデバイスを備えることができる。30

#### 【0026】

[0033] 図1の例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、ユーザインターフェース4、CPU6、メモリコントローラ8、メモリ10、グラフィックス処理ユニット（GPU）12、ディスプレイインターフェース14、ディスプレイ16、およびバス18を含む。ユーザインターフェース4、CPU6、メモリコントローラ8、GPU12、およびディスプレイインターフェース14は、バス18を使用して互いに通信することができる。図1に示す異なる構成要素同士の間のバスおよび通信インターフェースの特定の構成は単なる例示であり、本開示の本技法を実装するために、同じもしくは異なる構成要素を備えたコンピューティングデバイスおよび／または他のグラフィックス処理シ4050

ステムの他の構成が使用され得ることに留意されたい。

【0027】

[0034] C P U 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プロセッサまたは専用プロセッサを備えることができる。ユーザは、C P U 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の入力をコンピューティングデバイス 2 に与えることができる。C P U 6 上で実行されるソフトウェアアプリケーションは、たとえば、グラフィックスアプリケーション、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、メディアプレイヤアプリケーション、ビデオゲームアプリケーション、グラフィカルユーチュインターフェースアプリケーション、オペレーティングシステム、または任意の他のタイプのプログラムを含むことができる。  
10 ユーザは、ユーザインターフェース 4 を介してコンピューティングデバイス 2 に結合される、キーボード、マウス、マイクロホン、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 に入力を提供することができる。

【0028】

[0035] C P U 6 上で実行されるソフトウェアアプリケーションは、ディスプレイ 1 6 上での表示のためにグラフィックスデータをフレームバッファにレンダリングするように G P U 1 2 に指示する 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むことができる。いくつかの例では、グラフィックスレンダリング命令は、たとえば、Open Graphics Library (OpenGL (登録商標)) アプリケーションプログラミングインターフェース (API)、Open Graphics Library Embedded Systems (OpenGL ES) API、Direct3D API、X3D API、RenderMan API、WebGL API、または任意の他の公共のもしくはプロプライエタリな標準グラフィックス API などのグラフィックス API に準拠することができる。グラフィックスレンダリング命令を処理するために、C P U 6 は、グラフィックスデータのレンダリングの一部またはすべてを G P U 1 2 に実行するために、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 G P U 1 2 に発行することができる。一部の例では、レンダリングされることになるグラフィックスデータは、グラフィックスプリミティブのリスト、たとえば、点、線、三角形、四角形、トライアングルストリップなどを含み得る。  
20  
30

【0029】

[0036] メモリコントローラ 8 は、メモリ 1 0 に出入りするデータの転送を容易にする。たとえば、メモリコントローラ 8 は、コンピューティングデバイス 2 内のコンポーネントにメモリサービスを提供するために、メモリ読取コマンドとメモリ書き込みコマンドとを受け取り、メモリ 1 0 に関してそのようなコマンドをサービスする。メモリコントローラ 8 はメモリ 1 0 に通信可能に結合される。メモリコントローラ 8 は、図 1 の例のコンピューティングデバイス 2 内で、C P U 6 とメモリ 1 0 との両方と別々である処理モジュールであるものとして図示されているが、他の例では、メモリコントローラ 8 の機能性の一部またはすべてが、C P U 6 とメモリ 1 0 との一方または両方で実施され得る。

【0030】

[0037] メモリ 1 0 は、C P U 6 による実行のためにアクセス可能なプログラムモジュールおよび／もしくは命令ならびに／または C P U 6 上で実行されるプログラムによる使用のためのデータを記憶することができる。たとえば、メモリ 1 0 は、C P U 6 上で実行されるアプリケーションに関連するプログラムコードとグラフィックスデータとを記憶することができる。メモリ 1 0 は、さらに、コンピューティングデバイス 2 の他のコンポーネントによる使用のための情報および／またはこれによって生成された情報を記憶することができる。たとえば、メモリ 1 0 は、G P U 1 2 のデバイスマモリとして機能することができ、G P U 1 2 によって演算されることになるデータ、ならびに G P U 1 2 によって実行される演算の結果生じるデータを記憶することができる。たとえば、メモリ 1 0 は、テクスチャバッファ、深度バッファ、ステンシルバッファ、頂点バッファ、フレームバッファ  
40  
50

、レンダターゲット、または類似物の任意の組合せを記憶することができる。さらに、メモリ10は、GPU12による処理のためにコマンドストリームを記憶することができる。メモリ10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)、読み専用メモリ(ROM)、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、または光学記憶媒体など、1つまたは複数の揮発性または不揮発性のメモリまたはストレージデバイスを含むことができる。

### 【0031】

[0038] GPU12は、CPU6によってGPU12に発行されたコマンドを実行するよう構成され得る。GPU12によって実行されるコマンドは、グラフィックスコマンド、描画呼出しコマンド、GPU状態プログラミングコマンド、タイムスタンプ要求、メモリ転送コマンド、汎用コンピューティングコマンド、カーネル実行コマンドなどを含むことができる。10

### 【0032】

[0039]いくつかの例で、GPU12は、ディスプレイ16に1つまたは複数のグラフィックスプリミティブをレンダリングするためにグラフィックス動作を実行するよう構成され得る。そのような例では、CPU6上で実行され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時に、CPU6は、グラフィックスデータをGPU12に供給し、1つまたは複数のグラフィックスコマンドをGPU12に発行する。グラフィックスコマンドは、たとえば、描画呼出しコマンド、GPU状態プログラミングコマンド、メモリ転送コマンド、ブリットコマンドなどを含み得る。グラフィックスデータは、頂点バッファ、テクスチャデータ、面データなどを含み得る。いくつかの例では、CPU6は、コマンドとグラフィックスデータとをGPU12によってアクセスされ得るメモリ10に書き込むことによって、コマンドとグラフィックスデータとをGPU12に提供することができる。20

### 【0033】

[0040]さらなる例では、GPU12は、CPU6上で実行されるアプリケーションのために汎用コンピューティングを実行するよう構成され得る。そのような例では、CPU6上で実行されるソフトウェアアプリケーションの1つが計算タスクをGPU12にオフロードすることを決定するとき、CPU6は、汎用コンピューティングデータをGPU12に提供して、GPU12に1つまたは複数の汎用コンピューティングコマンドを出すことができる。汎用コンピューティングコマンドは、たとえば、カーネル実行コマンド、メモリ転送コマンドなどを含み得る。いくつかの例で、CPU6は、GPU12によってアクセスされ得るメモリ10にコマンドとデータとを書き込むことによって、コマンドおよび汎用コンピューティングデータをGPU12に供給することができる。30

### 【0034】

[0041] GPU12は、いくつかの例では、CPU6よりも効率的なベクトル演算の処理を行う高並列構造を伴って構築され得る。たとえば、GPU12は、複数の頂点、制御点、画素および/または他のデータについて並列的方式で演算するよう構成された複数の処理要素を含み得る。GPU12の高並列な性質は、いくつかの例では、GPU12が、CPU6を使用して画像をレンダリングするよりもより迅速にグラフィックス画像(たとえば、GDIおよび2次元(2D)ならびに/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16上にレンダリングするのを可能にし得る。さらに、GPU12の高並列な性質は、GPU12が、汎用コンピューティングアプリケーションに関するある種のタイプのベクトル演算と行列演算とをCPU6よりすばやく処理することを可能にすることができる。40

### 【0035】

[0042] GPU12は、一部の例では、コンピューティングデバイス2のマザーボードに統合され得る。他の例では、GPU12は、コンピューティングデバイス2のマザーボー50

ドにおけるポートに差し込まれたグラフィックスカード上に存在し得、また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。さらなる例では、GPU12は、システムオンチップ(SoC)を形成するCPU6と同じマイクロチップ上に配置され得る。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)、または他の同等の集積回路もしくはディスクリート論理回路網など、1つまたは複数のプロセッサを含むことができる。

#### 【0036】

[0043]いくつかの例では、GPU12は、メモリ10のすべてまたは一部に関するキャッシングサービスを提供することができるGPUキャッシングを含むことができる。そのような例では、GPU12は、オフチップメモリではなく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するのにキャッシングを使用することができる。これは、各読み取りコマンド中と各書き込みコマンド中に、激しいバストラフィックを経験する可能性があるバス18を介してGPU12がメモリ10にアクセスする必要を減らすことによって、GPU12がより効率的な形で動作することを可能にする。しかしながら、いくつかの例では、GPU12が、別々のキャッシングを含むのではなく、その代わりにバス18を介してメモリ10を利用する場合がある。GPUキャッシングは、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)、その他など、1つまたは複数の揮発性または不揮発性のメモリまたはストレージデバイスを含むことができる。

10

#### 【0037】

[0044]CPU6および/またはGPU12は、メモリ10内で割り振られるフレームバッファ内に、ラスタ化された画像データを記憶することができる。ディスプレイインターフェース14は、フレームバッファからデータを取り出し、ラスタ化された画像データによって表示される画像を表示するようにディスプレイ16を構成することができる。いくつかの例では、ディスプレイインターフェース14は、フレームバッファから取り出されたデジタル値をディスプレイ16によって消費可能なアナログ値に変換するように構成されたデジタル-アナログ変換器(DAC)を含むことができる。他の例では、ディスプレイインターフェース14は、処理のために、デジタル値をディスプレイ16に直接的に渡すことができる。

20

#### 【0038】

[0045]ディスプレイ16は、モニタ、テレビジョン、映写デバイス、液晶ディスプレイ(LCD)、プラズマディスプレイパネル、発光ダイオード(LED)アレイ、陰極線管(CRT)ディスプレイ、電子ペーパー、表面電界ディスプレイ(SED: surface-conduction electron-emitted display)、レーザーテレビジョンディスプレイ、ナノ結晶ディスプレイ、または別のタイプのディスプレイユニットを含むことができる。ディスプレイ16を、コンピューティングデバイス2内に一体化することができる。たとえば、ディスプレイ16を、携帯電話送受話器またはタブレットコンピュータのスクリーンとすることができる。代替案では、ディスプレイ16を、有線またはワイヤレスの通信リンクを介してコンピューティングデバイス2に結合される独立型のデバイスとすることができる。たとえば、ディスプレイ16を、ケーブルまたはワイヤレスリンクを介してパーソナルコンピュータに接続されたコンピュータモニタまたはフラットパネルディスプレイとすることができる。

30

#### 【0039】

[0046]バス18は、第1世代、第2世代、および第3世代のバス構造およびバスプロトコル、共有バス構造および共有バスプロトコル、ポイントツーポイントバス構造およびポイントツーポイントバスプロトコル、単一方向バス構造および単一方向バスプロトコル、ならびに両方向バス構造および両方向プロトコルを含む、バス構造とバスプロトコルとの任意の組合せを使用して実施され得る。バス18を実施するのに使用され得る異なるバス構造およびバスプロトコルの例は、たとえば、HyperTransportバス、In

40

50

finiBandバス、Advanced Graphics Portバス、Peripheral Component Interconnect(PCI)バス、PCI Expressバス、Advanced Microcontroller Bus Architecture(AMBA)Advanced High-performance Bus(AHB)、AMBA Advanced Peripheral Bus(APB)、およびAMBA Advanced eXtensible Interface(AXI)バスを含む。他のタイプのバス構造およびバスプロトコルも、使用され得る。

#### 【0040】

[0047]本開示によれば、コンピューティングデバイス2(たとえば、CPU6および/またはGPU12)は、本開示で説明されるイントラフレームタイムスタンプ値生成技法のいずれかを実行するように構成され得る。たとえば、コンピューティングデバイス2(たとえば、CPU6および/またはGPU12)は、グラフィックスフレームに関するタイルベースのレンダリングを実行している間にGPU12によって生成される複数のピンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成するように構成され得る。タイムスタンプ値は、複数のピンごとのタイムスタンプ値のうちの少なくとも2つのピンごとのタイムスタンプ値の関数とされ得る。アプリケーションによって要求されたタイムスタンプ値を生成するのにピンごとのタイムスタンプ値を使用することは、イントラフレームタイムスタンプ要求がタイルベースのレンダリングを実行するグラフィックス処理システムによってサポートされることを可能にすることができる。

10

#### 【0041】

[0048]動作中に、CPU6上で実行されるグラフィックスアプリケーションは、グラフィックスフレームをレンダリングするために、コマンドの順序付けられたシーケンス(たとえば、コマンドストリーム)を生成することができる。いくつかの場合に、コマンドの順序付けられたシーケンスは、複数の描画呼出しコマンドと複数のタイムスタンプ要求とを含む可能性がある。タイムスタンプ要求のうちの少なくともいくつかは、コマンドの順序付けられたシーケンス内の異なる描画呼出しコマンドの間に配置され得る。

20

#### 【0042】

[0049]タイルベースのレンダリング技法を使用してコマンドのシーケンスを実行するために、CPU6は、タイムスタンプ要求の各々について、それぞれのタイムスタンプ要求に基づいて複数のピンごとのタイムスタンプ要求を生成することができる。CPU6は、ピンごとのタイムスタンプ要求の各々を複数のコマンドストリームのうちのそれぞれのコマンドストリームに配置することができる。コマンドストリームの各々は、タイルベースのレンダリングを実行している間に、複数のレンダリングパス反復のうちのそれぞれのレンダリングパス反復中にGPU12によって実行され得る。これらのコマンドストリームは、ピンごとのコマンドストリームと呼ばれる場合がある。レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの副領域をレンダリングするよう構成され得る。

30

#### 【0043】

[0050]CPU6は、GPU12に、ピンごとのコマンドストリームを実行させることができる。ピンごとのコマンドストリームを実行している間に、GPU12は、GPU12によって受け取られたピンごとのコマンドストリーム内のピンごとのタイムスタンプ要求の実行に応答して、ピンごとのタイムスタンプ値を生成することができる。いくつかの場合に、GPU12は、ピンごとのコマンドストリーム内に含まれるピンごとのタイムスタンプ要求の各々について、それぞれのピンごとのタイムスタンプ値を生成することができる。ピンごとのタイムスタンプ値の各々は、それぞれのピンごとのタイムスタンプ値に関連するピンごとのタイムスタンプ要求がGPU12によって実行された時刻を示すことができる。いくつかの例では、ピンごとのタイムスタンプ値の各々は、GPU12によって生成されるそれぞれのピンごとのタイムスタンプ内に含まれ得る。

40

#### 【0044】

50

[0051]いくつかの例では、G P U 1 2は、ビンごとのタイムスタンプ値をC P U 6に供給することができる。ビンごとのタイムスタンプ値の受取に応答して、C P U 6は、ビンごとのタイムスタンプ値に基づいて、1つまたは複数のアプリケーションによって要求されたタイムスタンプ値を生成することができる。C P U 6は、アプリケーションによって要求されたタイムスタンプ値を生成するために、本開示で説明される技法のいずれかを使用することができる。C P U 6は、アプリケーションによって要求されたタイムスタンプ値をグラフィックスアプリケーションに供給することができる。

#### 【 0 0 4 5 】

[0052]さらなる例では、G P U 1 2は、ビンごとのタイムスタンプ値に基づいて、1つまたは複数のアプリケーションによって要求されたタイムスタンプ値を生成し、これらのアプリケーションによって要求されたタイムスタンプ値をC P U 6に供給することができる。G P U 1 2は、アプリケーションによって要求されたタイムスタンプ値を生成するために、本開示で説明される技法のいずれかを使用することができる。C P U 6は、アプリケーションによって要求されたタイムスタンプ値をグラフィックスアプリケーションに供給することができる。10

#### 【 0 0 4 6 】

[0053]追加の例では、G P U 1 2は、ビンごとのタイムスタンプ値に基づいて1つまたは複数の中間値を生成し、これらの中間値をC P U 6に供給することができる。C P U 6は、中間値に基づいてビンごとのタイムスタンプ値を生成することができる。C P U 6およびG P U 1 2は、アプリケーションによって要求されたタイムスタンプ値を生成するために、本開示で説明される技法のいずれかを使用することができる。C P U 6は、アプリケーションによって要求されたタイムスタンプ値をグラフィックスアプリケーションに供給することができる。20

#### 【 0 0 4 7 】

[0054]いくつかの例では、C P U 6および/またはG P U 1 2は、各アプリケーションによって要求されたタイムスタンプ値が少なくとも2つの異なるビンごとのタイムスタンプ値の関数になるように、アプリケーションによって要求されたタイムスタンプ値を生成することができる。そのような例では、少なくとも2つの異なるビンごとのタイムスタンプ値は、いくつかの例で、異なるレンダリングパス反復中に生成され得る。異なるレンダリングパス反復中に生成されるビンごとのタイムスタンプ値を使用することは、グラフィックスフレームのレンダリング中に実行するのに異なる描画呼出しコマンドが費やす時間の相対的な長さを少なくともある度合まで反映するアプリケーションによって要求されたタイムスタンプ値をグラフィックス処理システムが生成することを可能にすることができる。この形で、本開示で説明される技法は、タイルベースのレンダリングシステムが、不連続なインターリープされた形で描画呼出しコマンドを実行する場合であっても、タイルベースのレンダリングシステムが、イントラフレームタイムスタンプをサポートすることを可能にすることができます。30

#### 【 0 0 4 8 】

[0055]いくつかの例では、コンピューティングデバイス2(たとえば、C P U 6および/またはG P U 1 2)は、タイルベースのレンダリングを実行する時に受け取られるイントラフレームタイムスタンプ要求に応答してイントラフレームタイムスタンプを生成するように構成され得る。さらなる例では、コンピューティングデバイス2(たとえば、C P U 6および/またはG P U 1 2)は、レンダリングされるグラフィックスフレームについて実行される複数の描画呼出しコマンドと、レンダリングされるグラフィックスフレームに関連する1つまたは複数のタイムスタンプ要求とを含むコマンドストリームを受け取るよう構成され得る。そのような例では、コンピューティングデバイス2(たとえば、C P U 6および/またはG P U 1 2)は、いくつかの例で、G P U 1 2に、タイルベースのレンダリング技法を使用してグラフィックスフレームをレンダリングするために複数の描画呼出しコマンドを実行させ、1つまたは複数のタイムスタンプ要求に応答して1つまたは複数のタイムスタンプを生成させるようにさらに構成され得る。いくつかの例では、14050

つまたは複数のタイムスタンプを、複数のビンごとのタイムスタンプ値に基づいて生成することができる。

#### 【0049】

[0056]本開示で説明される技法は、いくつかの例で、たとえばCPU6、GPU12、およびシステムメモリ10を含む、図1に示されたコンピューティングデバイス2内のコンポーネントのいずれかにおいて実施され得る。たとえば、イントラフレームタイムスタンプを生成する技法を、CPU6内のグラフィックスドライバ、GPU12内の処理ユニット、またはその組合せによって実行することができる。別の例として、タイムスタンプ要求を、CPU6上で実行されるソフトウェアアプリケーション（たとえば、グラフィックスアプリケーションまたはユーザーアプリケーション）によって、CPU6上で実行されるGPUドライバに発行することができ、このタイムスタンプ要求に応答して、GPUドライバは、本開示の技法に従って生成されたタイムスタンプを返すことができる。いくつかの例では、タイムスタンプ要求および描画呼出しを、メモリ10内に記憶することができる（たとえば、1つまたは複数のコマンドキューの一部として）。さらなる例では、タイムスタンプ要求に応答して返されたタイムスタンプを、メモリ10内に記憶することができる。10

#### 【0050】

[0057]図2は、図1のコンピューティングデバイス2のCPU6、GPU12、およびメモリ10をさらに詳細に示すブロック図である。図2に示すように、CPU6はGPU12とメモリ10とに通信可能に結合され、GPU12はCPU6とメモリ10とに通信可能に結合される。いくつかの例では、GPU12は、CPU6によってマザーボードに統合され得る。追加の例では、GPU12は、CPU6を含むマザーボードのポート内にインストールされたグラフィックスカード上で実装され得る。さらなる例では、GPU12は、CPU6と相互作用するように構成された周辺デバイス内に組み込まれることが可能である。追加の例では、GPU12は、システムオンチップ（SOC）を形成するCPU6と同じマイクロチップ上に配置され得る。20

#### 【0051】

[0058]CPU6は、ソフトウェアアプリケーション24と、グラフィックスAPI26と、GPUドライバ28と、オペレーティングシステム30とのいずれかを実行するように構成された1つまたは複数のプロセッサ（たとえば、マイクロプロセッサ）を含むことができる。いくつかの例では、CPU6は、CPU6の1つまたは複数のプロセッサに本開示で説明される技法のいずれかのすべてまたは一部を実行させる命令を実行するように構成され得る。30

#### 【0052】

[0059]GPU12は、コマンドエンジン32と、1つまたは複数の処理ユニット34と、ビニングバッファ（binning buffer）36とを含む。1つまたは複数の処理ユニット34は、3Dグラフィックスレンダリングパイプラインを形成するように構成され得る。いくつかの例では、1つまたは複数の処理ユニット34は、オンチップのテッセレーション対応グラフィックスレンダリングパイプラインを実施することができる。コマンドエンジン32および処理ユニット34は、そのようなコンポーネントに帰せられる機能を実行するように構成された、専用ハードウェアユニットとファームウェアとソフトウェアとプロセッサとの任意の組合せを含むことができる。いくつかの例では、GPU12は、GPU12の1つまたは複数のプロセッサに本開示で説明される技法のいずれかのすべてまたは一部を実行させる命令を実行するように構成され得る。40

#### 【0053】

[0060]メモリ10は、1つまたは複数のコマンド38とプリミティブデータ40とタイムスタンプデータ42とを記憶することができる。いくつかの例で、メモリ10は、実行された時に1つまたは複数のプロセッサに本開示で説明される技法のいずれかのすべてまたは一部を実行させる命令をも記憶することができる。

#### 【0054】

[0061]ソフトウェアアプリケーション24は、1つまたは複数の3Dグラフィックスシーンおよび／または3Dグラフィックスオブジェクトをディスプレイ上に表示される画像にレンダリングするのにGPU12を使用するグラフィックスアプリケーションとすることができる。ソフトウェアアプリケーション24は、GPU12に3Dグラフィックスプリミティブのセットをラスタ化させ、レンダリングさせる命令を含むことができる。ソフトウェアアプリケーション24は、グラフィックスAPI26を介してGPUドライバ28に命令を発行することができる。グラフィックスAPI26は、ソフトウェアアプリケーション24から受け取られた命令をGPUドライバ28によって消費可能なフォーマットに変換するランタイムサービスとすることができます。

## 【0055】

10

[0062]GPUドライバ28は、グラフィックスAPI26を介してソフトウェアアプリケーション24から命令を受け取り、命令をサービスするためにGPU12の動作を制御する。たとえば、GPUドライバ28は、1つまたは複数のコマンド38を構築して、コマンド38をメモリ10内に配置して、コマンド38を実行するようにGPU12に命令することができる。いくつかの例では、GPUドライバ28は、コマンド38をメモリ10内に配置して、オペレーティングシステム30、たとえば、1つまたは複数のシステム呼出しを介してGPU12と通信することができる。

## 【0056】

20

[0063]オペレーティングシステム30は、ソフトウェアアプリケーション24、グラフィックスAPI26、およびGPUドライバ28がその上で実行されるソフトウェアプラットフォームを提供することができる。オペレーティングシステム30は、CPU6とメモリ10とGPU12との間でのデータの通信および転送のハードウェア詳細を管理することができる。

## 【0057】

[0064]コマンド38は、1つもしくは複数の状態コマンド、1つもしくは複数の描画呼出しコマンド、および／または1つもしくは複数のタイムスタンプ要求（たとえば、1つもしくは複数のビンごとのタイムスタンプ要求）を含むことができる。状態コマンドは、たとえばプリミティブデータなど、GPU12内の状態変数のうちの1つまたは複数を変更するようにGPU12に命令することができる。描画呼出しコマンドは、メモリ10内に記憶された（たとえば、頂点バッファ内で定義された）1つまたは複数の頂点のグループによって定義された形状をレンダリングするようにGPU12に命令することができる。1つまたは複数の頂点のグループによって定義されたジオメトリは、いくつかの例で、レンダリングされる複数のプリミティブ（たとえば、プリミティブデータ40）に対応することができる。一般に、描画呼出しコマンドは、メモリ10の定義されたセクション（たとえば、バッファ）内に記憶された頂点のすべてをレンダリングするためにGPU12を呼び出すことができる。言い換えると、GPU12が描画呼出しコマンドを受け取った後に、制御は、メモリ10の定義されたセクション（たとえば、バッファ）内の頂点によって表されるジオメトリとプリミティブとをレンダリングするために、GPU12に渡される。

30

## 【0058】

40

[0065]タイムスタンプ要求は、タイムスタンプ要求の処理に応答してタイムスタンプを生成するようにGPU12および／またはGPUドライバ28に命令することができる。GPU12および／またはGPUドライバ28は、タイムスタンプ要求の受取に応答してタイムスタンプを返すことができる。タイムスタンプは、時刻値を含むことができる。ビンごとのタイムスタンプ要求について、時刻値は、GPU12がタイムスタンプ要求を処理した時刻を示すものとすることができます。アプリケーションによって要求されたタイムスタンプ要求について、時刻値は、GPU12によってコマンドストリーム内の描画呼出しを実行するのに費やされた時間の相対的な長さを示すものとすることができます。

## 【0059】

[0066]いくつかの例では、コマンド38は、コマンドストリームの形で（たとえば、コ

50

マンドキュー、コマンドバッファなど)記憶され得る。コマンドストリームは、グラフィックスコマンドの順序付けられたシーケンスを指定することができる。いくつかの例では、コマンドの順序付けられたシーケンスは、複数の描画呼出しコマンドと、複数のタイムスタンプ要求とを含むことができる。いくつかの例では、タイムスタンプ要求のうちの少なくとも1つを、グラフィックスコマンドの順序付けられたシーケンス内で少なくとも2つの描画呼出しコマンドの間に位置決めすることができる。さらなる例では、描画呼出しコマンドのうちの少なくとも1つを、グラフィックスコマンドの順序付けられたシーケンス内で少なくとも2つのタイムスタンプ要求の間に位置決めすることができる。

#### 【0060】

[0067]コマンドエンジン32は、メモリ10内に記憶されたコマンド38を取り出し、実行するように構成される。コマンドエンジン32は、GPU12のレンダリング状態を管理し、処理ユニット34がグラフィックスレンダリングパイプラインを実施するように処理ユニット34の動作を制御し、グラフィックスレンダリングパイプラインを介してグラフィックスデータをレンダターゲットにレンダリングさせ、コマンドストリーム内のタイムスタンプ要求(たとえば、ピンごとのタイムスタンプ要求)に出会うことに応答してタイムスタンプを返すことができる。10

#### 【0061】

[0068]状態コマンドの受取に応答して、コマンドエンジン32は、状態コマンドに基づいてGPU内の1つまたは複数の状態レジスタに特定の値をセットし、および/または状態コマンドに基づいて固定機能処理ユニット34のうちの1つまたは複数を構成するように構成され得る。描画呼出しコマンドの受取に応答して、コマンドエンジン32は、メモリ10内の頂点によって表されるジオメトリ(たとえば、プリミティブデータ40によって表されるジオメトリ)を処理ユニット34にレンダリングさせるように構成され得る。コマンドエンジン32は、シェーダプログラム結合コマンドを受け取って、それらのシェーダプログラム結合コマンドに基づいて、特定のシェーダプログラムをプログラマブル処理ユニット34のうちの1つまたは複数にロードすることも可能である。タイムスタンプ要求(たとえば、ピンごとのタイムスタンプ要求)の受取に応答して、コマンドエンジン32は、タイムスタンプを生成し、そのタイムスタンプをCPU6(たとえば、GPUドライバ28)に供給することができる。20

#### 【0062】

[0069]処理ユニット34は、その各々がプログラマブル処理ユニットまたは固定機能処理ユニットであり得る、1つもしくは複数の処理ユニットを含み得る。プログラマブル処理ユニットは、たとえば、CPU6からGPU12上にダウンロードされた1つまたは複数のシェーダプログラムを実行するように構成されたプログラマブルシェーダユニットを含み得る。いくつかの例では、シェーダプログラムは、たとえば、OpenGL Shading Language(GLSL)、High Level Shading Language(HLSL)、C for Graphics(Cg)シェーディング言語など、ハイレベルシェーディング言語で書き込まれたプログラムのコンパイルバージョンであり得る。30

#### 【0063】

[0070]プログラマブルシェーダユニットは、いくつかの例で、並列に動作するように構成された複数の処理ユニット、たとえば単一命令複数データ(SIMD)パイプラインを含むことができる。プログラマブルシェーダユニットは、シェーダプログラム命令を記憶するプログラムメモリと実行状態レジスタ、たとえば、実行されつつあるプログラムメモリ内の現在の命令または取り出されるべき次の命令を示すプログラムカウンタレジスタとを有することができる。処理ユニット34内のプログラマブルシェーダユニットは、たとえば、頂点シェーダユニット、ピクセルシェーダユニット、ジオメトリシェーダユニット、ハルシェーダユニット、ドメインシェーダユニット、コンピュートシェーダユニット、および/またはユニファイドシェーダユニットを含み得る。40

#### 【0064】

50

20

50

[0071] 固定機能処理ユニットは、ある種の機能を実行するために配線接続されたハードウェアを含み得る。固定機能ハードウェアを、たとえば異なる機能を実行するために、1つまたは複数の制御信号を介して構成可能とすることができますが、固定機能ハードウェアは、通常、ユーザによってコンパイルされたプログラムを受け取ることができるプログラムメモリを含まない。いくつかの例では、処理ユニット34内の固定機能処理ユニットは、たとえば、深度テスト、シザーテスト、アルファブレンド、その他などのラスタ演算を実行する処理ユニットを含むことができる。

#### 【0065】

[0072] ビニングバッファ36は、レンダターゲットの副領域に関するラスタ化されたデータを記憶するように構成され得る。ビニングバッファ36は、レンダリングバスの実行中に実際のレンダターゲットの特定の副領域に関する一時レンダターゲットとして働くことができる。ビニングバッファ36は、たとえばランダムアクセスメモリ(RAM)、静态RAM(SRAM)、ダイナミックRAM(DRAM)、その他など、1つまたは複数の揮発性または不揮発性のメモリまたはストレージデバイスを含むことができる。いくつかの例では、ビニングバッファ36を、オンチップバッファとすることができる。オンチップバッファは、GPUが形成され、置かれ、および/または配置されるマイクロチップ、集積回路、および/またはダイと同一であるマイクロチップ、集積回路、および/またはダイ上に形成され、置かれ、および/または配置されるバッファを指すことができる。

#### 【0066】

[0073] いくつかの例では、処理ユニット34は、第1の通信インターフェースを介してビニングバッファ36にアクセスし、第1の通信インターフェースとは異なる第2の通信インターフェースを介してレンダターゲット(たとえば、メモリ10内に記憶されたフレームバッファ)にアクセスすることができる。そのような例では、第1の通信インターフェースは、いくつかの例で、第2の通信インターフェースより高い帯域幅を有することができる。第2の通信インターフェースは、いくつかの例で、図1のバス18と、図1のメモリコントローラ8とメモリ10との間の接続とに対応することができる。ビニングバッファ36がオンチップバッファである時には、第1の通信インターフェースを、GPU12の内部の通信インターフェースとすることができる。

#### 【0067】

[0074] 本明細書で使用される時に、帯域幅は、通信インターフェースが2つのコンポーネント、たとえばメモリコンポーネントとGPU12との間でデータを転送することができる速度を指すことができる。帯域幅の単位は、いくつかの例で、単位時間あたりのビット数、たとえばギガビット毎秒(Gb/s)として与えることができる。複数ビットのバス幅を有するバスが、通信インターフェースの一部として使用される時には、帯域幅は、いくつかの例で、バスの幅に、データが单一のビット線に沿って転送される速度を乗じた積と等しいものとすることができます。たとえば、バスが16ビット幅であり、このバスの各ビット線が2Gb/sの速度でデータを転送できる場合には、このバスの帯域幅は、32Gb/sと等しいものとすることができます。複数のバスが、2つのコンポーネントの間の通信インターフェースを形成する場合には、この通信インターフェースの帯域幅は、複数のバスの各々の帯域幅の関数、たとえば、個々のバスの各々の最小帯域幅とすることができます。

#### 【0068】

[0075] ビニングバッファ36が、GPU12と同一のチップ上で実施される時に、GPU12は、必ずしもシステムバスとメモリバスと(たとえば、図1のバス18と、図1のメモリコントローラ8とメモリ10との間の接続と)を介してビニングバッファ36にアクセスする必要があるのではなく、GPU12と同一のチップ上で実施される内部通信インターフェース(たとえば、バス)を介してビニングバッファ36にアクセスすることができる。そのようなインターフェースは、オンチップなので、システムバスおよびメモリバスより高い帯域幅で動作できる可能性がある。上で説明された技法は、メモリ10にア

10

20

30

40

50

クセスするのに使用される通信インターフェースの帯域幅を超える、ビニングバッファ36のための通信インターフェースを達成する1つの形であるが、他の技法が、可能であり、本開示の範囲に含まれる。

#### 【0069】

[0076]ビニングバッファ36の容量は、いくつかの例で、あるタイプのコンピューティングデバイス、たとえばモバイルデバイス上で使用可能な面積によって制限される場合がある。さらに、ビニングバッファ36が、GPU12と同一のチップ上で実施される時には、同一のチップ上でビニングバッファ36を実施するために使用可能な面積の量が、チップ上で実施される他の機能性に起因して制限される場合がある。いくつかの例では、ビニングバッファ36は、レンダターゲットのビット密度より低いビット密度を有する場合があり、これがビニングバッファ36の容量をさらに制限する可能性がある。上記および/または他の要因のゆえに、ビニングバッファ36の容量は、いくつかの場合に、レンダターゲットのサイズより小さくなる可能性がある。その結果、ビニングバッファ36の容量は、いくつかの例で、グラフィックス画像（たとえば、単一のフレーム）に関連する複数の宛先画素のすべての画素データを記憶するのに必要な最小容量未満である場合がある。メモリコンポーネントの容量は、そのメモリコンポーネント内に記憶され得るデータの最大量（たとえば、最大ビット数）を指すことができる。レンダターゲットのサイズは、レンダターゲットに割り振られたメモリ範囲内に記憶されるデータの量（たとえば、ビット数）を指すことができる。ビット密度は、特定の量の面積内に記憶できるビット数を指すことができる。

10

#### 【0070】

[0077]上で議論したように、タイルベースのレンダリングを実行する時に、GPU12は、レンダリングバスの別々の反復中にレンダターゲットの各副領域をレンダリングすることができる。たとえば、レンダターゲットの特定の副領域（たとえば、グラフィックス画像の宛先画素の特定のサブセット）の単一のレンダリングバス反復の一部として、GPU12は、レンダターゲットの特定の副領域に関するプリミティブのすべてまたはサブセットをレンダリングすることができる。ビニングバッファ36の容量は、レンダターゲットの副領域のサイズ以上になるように構成され得る。したがって、単一のレンダリングバス反復中に、レンダターゲットの副領域のうちのそれぞれの副領域に関連するすべての宛先画素データは、必ずしもメモリ10内のフレームバッファにアクセスする必要なしに、ビニングバッファ36内で使用可能にされ得る。その結果、単一のレンダリングバス反復中に、GPU12は、相対的に低帯域幅の通信インターフェースを介してメモリ10から宛先画素データを読み取る必要があるのではなく、相対的に高帯域幅の通信インターフェースを介してビニングバッファ36から宛先画素データを読み取ることができる可能性がある。

20

30

#### 【0071】

[0078]タイルベースのレンダリングを実行しないいくつかのグラフィックスシステムは、ハードウェアベースのオンチップキャッシュを使用することによってフレームバッファの一部をキャッシングすることができる場合があるが、そのようなキャッシングは、所与の画素の宛先画素値が、必要な時に使用可能になることを保証しない。これは、複数の宛先画素がハードウェアベースのキャッシング内で同一のアドレスにマッピングされる可能性があるからである。タイルベースのレンダリングが使用されない場合には、ハードウェアベースのキャッシングの現在の状態は、必ずしも、レンダターゲットの現在処理されている副領域に関連する宛先画素値を含むのではなく、レンダターゲットの他の副領域内の以前に処理されたプリミティブに関連する宛先画素値を含む可能性がある。

40

#### 【0072】

[0079]複数の宛先画素が同一のキャッシング位置にマッピングされるハードウェアベースのキャッシングとは異なって、所与のレンダリングバス反復についてビニングバッファ36内に記憶される宛先画素は、いくつかの例で、一意にアドレッシング可能とすることができます。言い換えると、所与のレンダリングバス反復について、ビニングバッファ36内の

50

アドレス可能ストレージスロットとそのレンダリングパス反復に使用される宛先画素との間に、1対1マッピングを定義することができる。その結果、タイルベースのレンダリングを実行する時に、所与のビニングパスに関するすべての宛先画素値が、いくつかの例で、相対的に低帯域幅の通信インターフェースを介してビニングバッファ36から使用可能とされ得る。さらに、ハードウェアベースのキャッシュシステムとは異なって、ビニングバッファ36内の一意にアドレス可能なデータのゆえに、キャッシュミスが発生せず、これによって、キャッシュミスの場合の帯域幅的に高価なフレームバッファアクセスに頼る必要が軽減される。

#### 【0073】

[0080]宛先画素は、特定の画素位置についてレンダターゲット（たとえば、フレームバッファまたは対応するビニングバッファのいずれか）内に記憶された画素データを指すことができる。対照的に、ソース画素は、処理ユニット34内のラスタ化処理ユニットによって生成されたが、まだレンダターゲットに記憶されおよび／またはマージされていない画素データを指すことができる。宛先画素は、異なるプリミティブに関連する複数のソース画素から合成された画素データを含む場合がある。10

#### 【0074】

[0081]タイルベースのレンダリングを実行するために、ソフトウェアアプリケーション24は、いくつかの例で、レンダリングされる1つまたは複数の3Dグラフィックスプリミティブのセットを幾何学的に定義するプリミティブデータ40をメモリ10内に配置し、1つまたは複数の描画呼出しコマンドをグラフィックスAPI26を介してGPUドライバ28に発行することができる。描画呼出しコマンドは、プリミティブデータ40によって定義されたプリミティブをGPU12によってラスタ化させ、レンダターゲット（たとえば、メモリ10内に記憶されたフレームバッファ）にレンダリングさせることができる。20

#### 【0075】

[0082]いくつかの例では、ソフトウェアアプリケーション24は、特定のタイプのプリミティブをレンダリングするようにGPU12を構成することができる。たとえば、ソフトウェアアプリケーション24は、描画呼出し中にレンダリングすべきプリミティブの特定のタイプを指定する状態コマンドをGPU12に発行することができる。追加の例では、描画呼出しコマンドを発行する前に、ソフトウェアアプリケーション24は、プリミティブをレンダリングするために1つまたは複数のテッセレーション技法を使用するようにGPU12を構成することができる。たとえば、ソフトウェアアプリケーション24は、テッセレーション技法を実施する1つまたは複数のシェーダプログラムに、描画呼出し命令中にGPU12の1つまたは複数のシェーダユニット（たとえば、ハルシェーダユニットおよび／またはドメインシェーダユニット）上で実行させることができる。30

#### 【0076】

[0083]プリミティブデータ40は、レンダリングされる1つまたは複数のプリミティブを示すデータを含むことができる。いくつかの場合に、プリミティブデータ40は、レンダリングされるプリミティブを幾何学的に定義することができる。プリミティブを幾何学的に定義することは、頂点（または制御点）のセットと対応する頂点属性とによってプリミティブを定義することを指すことができる。いくつかの例では、プリミティブデータ40は、複数の頂点、頂点リスト、および／または頂点バッファの形をとることができる。さらなる例では、プリミティブデータ40は、インデックスバッファと組み合わされた頂点バッファの形をとることができる。そのような例では、頂点バッファは、頂点を定義することができ、インデックスバッファは、どの頂点がプリミティブの各々を定義するのに使用されるのかを指定することができる。40

#### 【0077】

[0084]プリミティブデータ40内に含まれる頂点の各々は、たとえば位置座標、法線座標、テクスチャ座標、その他などの1つまたは複数の属性を含むことができる。頂点は、幾何プリミティブ（たとえば、点、線、三角形など）の頂点および／またはより高次のブ50

リミティブ（たとえば、ベジエ面などのより高次の面）の制御点に概念的に対応することができる。いくつかの場合に、頂点の各々を、1つまたは複数の頂点のグループにグループ化することができ、頂点のこれらのグループの各々は、単一のプリミティブに対応することができる。

#### 【0078】

[0085]幾何学的に定義されたプリミティブの形状を、いくつかの例で、必ずしもプリミティブデータ40内には含まれない追加のデータによって定義することができる。追加のデータは、1つもしくは複数の事前に決定されたプリミティブタイプのセットからの指定されたプリミティブタイプ、1つもしくは複数の数学関数、および/または1つもしくは複数のテッセレーション技法のうちの1つまたは複数を含むことができる。

10

#### 【0079】

[0086]いくつかの例では、指定されたプリミティブタイプは、GPU12内のレンダリング状態変数として記憶され得、ソフトウェアアプリケーション24によって構成可能とされ得る。指定されたプリミティブタイプは、いくつかの場合に、結果のレンダリングされるプリミティブの形状（たとえば、点、線、三角形など）および/またはプリミティブデータ40内に含まれる頂点の接続性（たとえば、トライアングルストリップ、トライアングルファンなど）を定義することができる。いくつかの例では、異なるプリミティブタイプは、処理ユニット34によって実施されるグラフィックスパイプラインが処理できるプリミティブトポロジのセットに対応することができる。さらなる例では、異なるプリミティブタイプは、グラフィックスAPI26によって定義され、ソフトウェアアプリケーション24による使用のために使用可能な、プリミティブトポロジのセットに対応することができる。

20

#### 【0080】

[0087]1つまたは複数の数学関数および/または1つまたは複数のテッセレーション技法は、GPU12の1つまたは複数のシェーダユニット（たとえば、ハルシェーダユニットおよび/またはドメインシェーダユニット）上で実行するように構成された1つまたは複数のシェーダプログラム内で指定され得る。数学関数は、曲線および/または曲面を有するプリミティブを定義するのに使用され得る。1つまたは複数のテッセレーション技法は、入力プリミティブの形状および/または曲率を近似する複数のテッセレートされたプリミティブによってプリミティブを定義するのに使用され得る。

30

#### 【0081】

[0088]ソフトウェアアプリケーション24からの描画呼出しコマンドの受取に応答して、GPUドライバ28は、レンダリングされる複数のプリミティブ（たとえば、プリミティブデータ40）に基づいて、GPU12にタイルベースのレンダリングを実行させることができる。たとえば、GPUドライバ28は、GPU12に、ビニングパスと複数のレンダリングパス反復を含むレンダリングパスとを実行させることができる。ビニングパス中に、GPU12は、プリミティブの各々がレンダターゲットの複数の副領域のうちのどれに画像データ（たとえば、画素データ）を与えるのかを決定し、プリミティブの各々がレンダターゲットの複数の副領域のうちのどれに画像データ（たとえば、画素データ）を与えるのかを示すビニングデータを生成することができる。ビニングデータが生成された後に、GPU12は、プリミティブの合成されたラスタ化された版を生成するために、ビニングデータとプリミティブデータ40とに基づいて、複数のレンダリングパス反復を含むレンダリングパスを実行することができる。

40

#### 【0082】

[0089]いくつかの例では、ビニングパスを実行するために、GPU12内のラスタライザは、ラスタ化されるプリミティブに対して低解像度zバッファリングおよび/または裏面カリング(back-face culling)を実行するように構成され得る。そのような例では、ビニングデータは、zバッファリングおよび/または裏面カリングの後に可視であるプリミティブに基づいて生成され得る。

#### 【0083】

50

[0090]いくつかの場合に、レンダリングされたプリミティブは、複数の画素として記憶され得る。画素の各々は、レンダターゲットの1つまたは複数の空間位置に関連付けられ得、それぞれの画素の色を示す1つまたは複数の属性を含むことができる。いくつかの場合に、画素の各々は、さらに、画素の透明度を示す1つまたは複数の属性を含むことができる。いくつかの例では、画素データは、各画素の赤、緑、青、およびアルファ（RGA）属性を含むことができ、「RGB」成分は色値に対応し「A」成分はアルファ値（すなわち、透明度またはブレンディング値）に対応する。

#### 【0084】

[0091]本開示で説明される技法は、たとえばソフトウェアアプリケーション24、グラフィックスAPI26、GPUドライバ28、コマンドエンジン32、および処理ユニット34を含む、図2に示されたコンポーネントのいずれかにおいて実施され得る。たとえば、GPUドライバ28、コマンドエンジン32、および/または処理ユニット34は、本開示で説明される技法のいずれかに従って1つまたは複数のタイムスタンプ（たとえば、イントラフレームタイムスタンプ）を生成するように構成され得る。  
10

#### 【0085】

[0092]いくつかの例では、GPUドライバ28、コマンドエンジン32、および/または処理ユニット34は、グラフィックスフレームに関するタイルベースのレンダリングを実行している間にGPU12によって生成される複数のビンごとのタイムスタンプ値に基づいて、時点を示すタイムスタンプ値を生成するように構成され得る。タイムスタンプ値は、複数のビンごとのタイムスタンプ値のうちの少なくとも2つのビンごとのタイムスタンプ値の関数とされ得る。アプリケーションによって要求されたタイムスタンプ値を生成するのにビンごとのタイムスタンプ値を使用することは、タイルベースのレンダリングを実行するグラフィックス処理システムによってイントラフレームタイムスタンプ要求をサポートすることを可能にすることができる。  
20

#### 【0086】

[0093]動作中に、CPU6上で実行されるソフトウェアアプリケーション24（たとえば、グラフィックスアプリケーション）は、グラフィックスフレームをレンダリングするために、コマンドの順序付けられたシーケンス（たとえば、コマンドストリーム）を生成することができる。いくつかの場合に、コマンドの順序付けられたシーケンスは、複数の描画呼出コマンドと複数のタイムスタンプ要求とを含むことができる。タイムスタンプ要求のうちの少なくともいくつかは、コマンドの順序付けられたシーケンス内の異なる描画呼出コマンドの間に配置され得る。ソフトウェアアプリケーション24は、グラフィックスAPI26を介してGPUドライバ28にコマンドの順序付けられたシーケンスを供給することができる。  
30

#### 【0087】

[0094]タイルベースのレンダリング技法を使用してコマンドのシーケンスを実行するために、GPUドライバ28は、タイムスタンプ要求の各々について、それぞれのタイムスタンプ要求に基づいて複数のビンごとのタイムスタンプ要求を生成することができる。GPUドライバ28は、ビンごとのタイムスタンプ要求の各々を、複数のコマンドストリームのうちのそれぞれのコマンドストリーム（たとえば、メモリ10内のコマンド38）に配置することができる。コマンドストリームの各々は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれのレンダリングパス反復中にGPU12によって実行され得る。このコマンドストリームは、ビンごとのコマンドストリームと呼ばれる場合がある。レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの副領域をレンダリングするように構成され得る。  
40

#### 【0088】

[0095]GPUドライバ28は、GPU12に、ビンごとのコマンドストリームを実行させることができる。ビンごとのコマンドストリームを実行している間に、GPU12（たとえば、コマンドエンジン32および/または処理ユニット34）は、GPU12によって受け取られたビンごとのコマンドストリーム内のビンごとのタイムスタンプ要求の実行  
50

に応答して、ピンごとのタイムスタンプ値を生成することができる。いくつかの場合に、G P U 1 2は、ピンごとのコマンドストリーム内のピンごとのタイムスタンプ要求の各々について、それぞれのピンごとのタイムスタンプ値を生成することができる。ピンごとのタイムスタンプ値の各々は、それぞれのピンごとのタイムスタンプ値に関連するピンごとのタイムスタンプ要求がG P U 1 2によって実行された時刻を示すことができる。いくつかの例では、ピンごとのタイムスタンプ値の各々は、G P U 1 2によって生成されるそれぞれのピンごとのタイムスタンプ内に含まれ得る。

#### 【 0 0 8 9 】

[0096]いくつかの例では、G P U 1 2は、ピンごとのタイムスタンプ値をG P U ドライバ2 8に供給することができる。たとえば、G P U 1 2は、ピンごとのタイムスタンプ値をメモリ1 0のタイムスタンプデータ4 2内に配置することができ、このタイムスタンプデータ4 2が、G P U ドライバ2 8によってアクセスされ得る。ピンごとのタイムスタンプ値の受取に応答して、G P U ドライバ2 8は、ピンごとのタイムスタンプ値に基づいて1つまたは複数のアプリケーションによって要求されたタイムスタンプ値を生成することができる。G P U ドライバ2 8は、アプリケーションによって要求されたタイムスタンプ値を生成するのに、本開示で説明される技法のいずれかを使用することができる。G P U ドライバ2 8は、アプリケーションによって要求されたタイムスタンプ値をソフトウェアアプリケーション2 4に供給することができる。

#### 【 0 0 9 0 】

[0097]さらなる例では、G P U 1 2は、ピンごとのタイムスタンプ値に基づいて1つまたは複数のアプリケーションによって要求されたタイムスタンプ値を生成し、アプリケーションによって要求されたタイムスタンプ値をG P U ドライバ2 8に供給することができる。G P U 1 2は、アプリケーションによって要求されたタイムスタンプ値を生成するのに、本開示で説明される技法のいずれかを使用することができる。いくつかの例では、G P U 1 2は、アプリケーションによって要求されたタイムスタンプ値をメモリ1 0のタイムスタンプデータ4 2に配置することができ、このタイムスタンプデータ4 2が、G P U ドライバ2 8によってアクセスされ得る。G P U ドライバ2 8は、アプリケーションによって要求されたタイムスタンプ値をソフトウェアアプリケーション2 4に供給することができる。

#### 【 0 0 9 1 】

[0098]追加の例では、G P U 1 2は、ピンごとのタイムスタンプ値に基づいて1つまたは複数の中間値を生成し、この中間値をG P U ドライバ2 8に供給することができる。G P U ドライバ2 8は、中間値に基づいてピンごとのタイムスタンプ値を生成することができる。G P U ドライバ2 8およびG P U 1 2は、アプリケーションによって要求されたタイムスタンプ値を生成するのに、本開示で説明される技法のいずれかを使用することができる。G P U ドライバ2 8は、アプリケーションによって要求されたタイムスタンプ値をソフトウェアアプリケーション2 4に供給することができる。

#### 【 0 0 9 2 】

[0099]いくつかの例では、C P U 6および/またはG P U 1 2は、各アプリケーションによって要求されたタイムスタンプ値が少なくとも2つの異なるピンごとのタイムスタンプ値の関数になるように、アプリケーションによって要求されたタイムスタンプ値を生成することができる。そのような例では、少なくとも2つの異なるピンごとのタイムスタンプ値は、いくつかの例で、異なるレンダリングパス反復中に生成され得る。異なるレンダリングパス反復中に生成されるピンごとのタイムスタンプ値を使用することは、グラフィックスフレームのレンダリング中に実行するのに異なる描画呼出しコマンドによって費やされる時間の相対的な長さを少なくともある度合まで反映するアプリケーションによって要求されたタイムスタンプ値をグラフィックス処理システムが生成することを可能にすることができる。この形で、本開示で説明される技法は、タイルベースのレンダリングシステムが描画呼出しコマンドを不連続なインターリープされた形で実行する場合であっても、タイルベースのレンダリングシステムがイントラフレームタイムスタンプをサポートす

10

20

30

40

50

ることを可能にすることができる。

#### 【0093】

[0100] タイルベースのレンダリングは、いくつかの例で、レンダターゲットを複数の副領域（たとえば、ピンまたはタイル）に副分割することと、レンダターゲットの副領域の各々について別々のレンダリングパス反復を含むレンダリングパスを実行することとを含むことができる。レンダリングパス中に処理される必要があるプリミティブの個数を減らすために、ビニングパスが、いくつかの例で、レンダリングパスの前に実行され得る。ビニングパスは、レンダリングされるプリミティブの各々がレンダターゲットの複数の副領域のうちのどれに画素データを与えるのかを示すビニングデータを生成するのに使用され得る。ビニングデータは、これによってレンダリングパス中に処理される必要があるプリミティブの個数を減らすために、特定のレンダリングパス反復中にアクティブであるレンダターゲットの副領域に寄与するプリミティブを選択的にレンダリングするのに、レンダリングパス反復中に使用され得る。10

#### 【0094】

[0101] レンダリングは、グラフィックスシーン内の 3D オブジェクトに対応する 3D グラフィックスプリミティブを 2D のラスタ化された画像データに変換するプロセスを指すことができる。レンダリングは、通常、レンダターゲット（たとえば、フレームバッファ）に関して行われ、このレンダターゲットは、通常、シーン内のグラフィックスプリミティブの各々がレンダリングされる時に更新される。したがって、レンダターゲットは、グラフィックスシーンの最終的な 2D のラスタ化された画像データを記憶するだけではなく、グラフィックスシーンがレンダリングされる時に中間データを記憶することもできる。レンダターゲット内に記憶される 2D のラスタ化された画像データは、複数の画素を含むことができ、画素の各々は、色データ、透明度データ、および／または深度データを含む。各新しいプリミティブがレンダターゲット内にレンダリングされる時に、新しいプリミティブの 2D のラスタ化された画像データは、以前にレンダリングされたプリミティブに関するレンダターゲット内にすでに記憶されている既存の中間データとマージされる。20

#### 【0095】

[0102] レンダターゲット内でデータをマージするために、中間データは、通常、レンダターゲットに新しいデータを書き込む前にレンダターゲットから読み取られる必要がある。したがって、レンダリングは、レンダターゲットを含むメモリに関する多数の読取動作と書込動作との実行を伴い、これによって大量のメモリ帯域幅使用をもたらす場合がある。大量のメモリ帯域幅使用のゆえに、レンダターゲットに専用の高帯域幅オンチップメモリを使用することが望ましい。しかし、たとえばモバイルアプリケーションなど、面積を制限されるアプリケーションでは、レンダターゲット内の画素の各々のデータのすべてを同時に保持することができる高帯域幅オンチップメモリを実装するのに十分な使用可能な面積がない場合がある。30

#### 【0096】

[0103] タイルベースのレンダリングは、レンダターゲットを複数の副領域（たとえば、タイルまたはピン）に副分割することと、レンダターゲットの副領域の各々について別々のレンダリングパス反復を含むレンダリングパスを実行することによって、上で述べた問題に対処することができる。副領域の各々は、レンダターゲット内の画素のサブセット（たとえば、画素の 16 × 16 タイル）に対応することができる。レンダターゲットの副領域は、その代わりにタイルまたはピンと呼ばれる場合がある。レンダリングパス反復の各々中に、対応する副領域に関連する画像データのすべてをレンダリングすることができ、これは、画素データを副領域に与えるプリミティブの各々をレンダリングすることを含む可能性がある。レンダターゲットの単一の副領域のデータを記憶するのに十分に大きい高帯域幅オンチップメモリが、レンダリングパス反復の各々のローカルレンダターゲットとして使用され得、あるレンダリングパス反復が完了した後に、そのレンダリングパス反復のローカルレンダターゲットの内容は、低帯域幅オフチップシステムメモリに記憶される一般的なレンダターゲットに転送され得る。タイルごとに別々のレンダリングパス反復4050

を実行することによって、タイルベースのレンダリング方式は、大きいオンチップメモリを可能にしない面積を制限されたアプリケーションにおいても、ラスタ化された画像データをマージするのに高帯域幅オンチップメモリを使用することを可能にすることができる可能性がある。

#### 【 0 0 9 7 】

[0104]タイルベースのレンダリングを実行する1つの手法は、レンダターゲットの副領域の各々についてレンダリングパス反復を実行し、レンダリングパス反復の各々中に、現在レンダリングされつつある特定の副領域への出力を制限するために異なるシザーセッティングを使用しながらシーン内のプリミティブのすべてをレンダリングすることである。しかしながら、そのような手法は、プリミティブがレンダリングされる副領域内で実際に可視であるか否かにかかわらず、プリミティブの各々がレンダリングパス反復の各々でレンダリングされるので、非効率的である可能性がある。  
10

#### 【 0 0 9 8 】

[0105]タイルベースのレンダリングの効率を改善するために、ビニングパスは、いくつかの例で、レンダリングパスの実行の前に実行され得る。ビニングパスは、プリミティブについてビニングデータを決定するのに使用され得る。レンダリングされるプリミティブの各々について、ビニングデータは、プリミティブの各々がレンダターゲットの副領域のどれに画素データを与えるのかに関して示すことができる。

#### 【 0 0 9 9 】

[0106]図3は、複数の副領域に副分割される例のレンダターゲット50を示す概念図である。図3は、副分割されたレンダターゲット50上に表示される複数のプリミティブをも示す。いくつかの場合に、レンダターゲット50は、フレームバッファに対応することができる。図3に示されているように、レンダターゲット50は、1~20の番号を付けられた、複数のオーバーラップしない副領域（代替案ではビンまたはタイルと呼ばれる）に分割される。副領域の各々は、レンダターゲット50内の画素のサブセット（たとえば、画素の16×16タイルなど）に対応することができる。図3に示された例のプリミティブについて、ビニングデータは、プリミティブAがタイル1と2と6と7と8と12とに画素データを与えること、プリミティブBがタイル7と8と12と13とに画素データを与えることなどを示すことができる。  
20

#### 【 0 1 0 0 】

[0107]いくつかの例では、ビニングデータは、レンダターゲットにレンダリングされるプリミティブの各々のラスタ化された版の合成物に基づいて生成され得る。いくつかの場合に、保守的なzテストおよび／または他のカリング技法が、プリミティブの各々のラスタ化された版を生成するのに使用され得る。保守的なzテストおよび／または他のカリング技法は、特定のタイルに寄与すると言われるプリミティブのリストに含まれないように、遮蔽されるプリミティブ（すなわち、他のプリミティブの背後に配置されたプリミティブ）を除去することができる。  
30

#### 【 0 1 0 1 】

[0108]特定の副領域（たとえば、タイルまたはビン）のレンダリングパス反復中に、ビニングデータは、実際に副領域に画像データ（たとえば、画素データ）を与える、レンダリングされるプリミティブを選択し、副領域に画像データを与えないプリミティブのレンダリングをバイパスするのに使用され得る。この形で、所与のレンダリングパス反復中に処理される必要があるプリミティブの個数が、いくつかの場合に減らされ得る。  
40

#### 【 0 1 0 2 】

[0109]いくつかの例では、GPU12は、ビニングパスから生成されたビニングデータに基づいて、レンダターゲットの副領域の各々のレンダリングパス反復を実行することができる。たとえば、複数のレンダリングパス反復の各々について、GPU12は、ビニングデータに基づいて、それぞれのレンダリングパス反復中に1つまたは複数の描画呼出しに関連する複数のプリミティブをレンダリングすべきかどうかを決定することができる。プリミティブがそれぞれのレンダリングパス反復に関連する副領域に画素データを与える  
50

ことをビニングデータが示す場合には、G P U 1 2 は、そのレンダリングパス反復中に、それぞれのレンダリングパス反復に関連する副領域にプリミティブをレンダリングすることができる。その一方で、プリミティブがそれぞれのレンダリングパス反復に関連する副領域に画素データを与えないことをビニングデータが示す場合には、G P U 1 2 は、それぞれのレンダリングパス反復に関連する副領域にそのプリミティブをレンダリングしないものとすることができます。

#### 【 0 1 0 3 】

[0110]図3に示された副領域は、実質的に同一のサイズと形状とであるが、他の例では、副領域が、異なるサイズおよび／または異なる形状を有することができる。さらに、副領域のサイズおよび形状は、G P U の製造時またはレンダリングの時に実質的に固定される必要があるのでなく、いくつかの例では、G P U 1 2 の動作中に動的に調整され得る。

10

#### 【 0 1 0 4 】

[0111]図4は、本開示に従ってグラフィックスアプリケーションによって発行される例のコマンドストリーム52を示す概念図である。いくつかの例では、コマンドストリーム52は、図2に示されたソフトウェアアプリケーション24によって発行され得る。さらなる例では、コマンドストリーム52は、図2に示されたG P U ドライバ28によって受け取られ、および／または処理され得る。

#### 【 0 1 0 5 】

[0112]コマンドストリーム52は、コマンドの順序付けられたシーケンス（すなわち、T S R 1 、 D R A W 1 、 T S R 2 、 D R A W 2 、 T S R 3 、 D R A W 3 、 T S R 4 ）を含む。図4では、T S R は「t i m e s t a n p r e q u e s t （タイムスタンプ要求）」を表し、D R A W は「d r a w c a l l c o m m a n d （描画呼出しコマンド）」を表す。したがって、図4に示されたコマンドのシーケンスは、第1のタイムスタンプ要求（T S R 1 ）と、それに続く第1の描画呼出しコマンド（D R A W 1 ）と、それに続く第2のタイムスタンプ要求（T S R 2 ）と、それに続く第2の描画呼出しコマンド（D R A W 2 ）と、それに続く第3のタイムスタンプ要求（T S R 3 ）と、それに続く第3の描画呼出しコマンド（D R A W 3 ）と、それに続く第4のタイムスタンプ要求（T S R 4 ）と、それに続く第4の描画呼出しコマンド（D R A W 4 ）とを含む。

20

#### 【 0 1 0 6 】

30

[0113]コマンドストリーム52内のコマンドの各々は、単一のグラフィックスフレームのレンダリングに関連付けられ得る。描画呼出しコマンドの各々は、同一のグラフィックスフレームの一部としてレンダリングされる1つまたは複数のプリミティブを指定することができる。いくつかの例では、異なる描画呼出しコマンドが、そのグラフィックスフレーム内のプリミティブをレンダリングするのに使用される異なるプリミティブタイプおよび／またはそのグラフィックスフレーム内のプリミティブをレンダリングするのに使用される異なるレンダリング状態に関連付けられ得る。

#### 【 0 1 0 7 】

[0114]図4に示されているように、タイムスタンプ要求は、描画呼出しコマンドの各々の間に配置される。図4にさらに示されているように、描画呼出しコマンドは、タイムスタンプ要求の各々の間に配置される。T S R 2 およびT S R 3 は、そのようなタイムスタンプ要求のそれぞれが2つの異なる描画呼出しコマンドの間に位置決めされるので、両方ともイントラフレームタイムスタンプ要求に対応することができる。

40

#### 【 0 1 0 8 】

[0115]図5は、本開示に従ってレンダリングパスを実行するための例の実行タイムライン54を示す概念図である。実行タイムライン54は、時間において左から右に増加する。

#### 【 0 1 0 9 】

[0116]例の実行タイムライン54は、レンダターゲットが4つのビンまたは副領域（すなわち、AとBとCとDと）に副分割される場合の、レンダリングパスの一部として実行

50

されるレンダリングパス反復のシーケンスを示す。レンダリングパス反復の各々は、レンダターゲットの特定のビンに関する実行される。たとえば、「レンダリングパス反復A」は、レンダターゲットの第1のビン（すなわち、ビン「A」）に関する実行され、「レンダリングパス反復B」は、レンダターゲットの第2のビン（すなわち、ビン「B」）に関する実行されるなどである。

#### 【0110】

[0117]いくつかの例では、図1に示されたコンピューティングシステム2が、実行タイムライン54に示されたレンダリングパス反復を実行することができる。さらなる例では、CPU12が、実行タイムライン54に示されたレンダリングパス反復を実行することができ、CPU6は、GPU12に、実行タイムライン54に示されたレンダリングパス反復を実行させることができる。10

#### 【0111】

[0118]図5に示されているように、コンピューティングシステム2（たとえば、CPU6および/またはGPU12）は、レンダターゲット内に含まれるビンの各々についてレンダリングパス反復を実行することができる。たとえば、コンピューティングシステム2（たとえば、CPU6および/またはGPU12）は、第1のビンに関する第1のレンダリングパス反復（レンダリングパス反復A）と、それに続く第2のビンに関する第2のレンダリングパス反復（レンダリングパス反復B）と、それに続く第3のビンに関する第3のレンダリングパス反復（レンダリングパス反復C）と、それに続く第4のビンに関する第4のレンダリングパス反復（レンダリングパス反復D）とを実行することができる。20

#### 【0112】

[0119]図6は、本開示に従って複数のレンダリングパス反復を実行するための例のコマンドストリーム56と58と60と62とを示す概念図である。いくつかの例では、コマンドストリーム56、58、60、および62は、図2に示されたGPUドライバ28によって発行され、および/または生成され得る。さらなる例では、コマンドストリーム56、58、60、および62は、図2に示されたGPU12および/またはコマンドエンジン32によって受け取られ、および/または処理され得る。

#### 【0113】

[0120]図6では、TSRは「`timestamp request`（タイムスタンプ要求）」を表し、DRAWは「`per-bin draw call command`（ビンごとの描画呼出しコマンド）」を表す。TSRGPUタイムスタンプ要求は、GPUドライバ28によって生成され、それぞれのレンダリングパス反復の始めにそれぞれのコマンドストリーム56と58と60と62との中に配置されるタイムスタンプ要求に対応する。いくつかの例では、図6のTSRおよびTSRGPUは、図4のコマンドストリーム52内に含まれるタイムスタンプ要求からそのようなタイムスタンプ要求を区別するために、その代わりに、ビンごとのタイムスタンプ要求と呼ばれる場合がある。ビンごとのタイムスタンプ要求に応答して生成されるタイムスタンプおよびタイムスタンプ値は、それぞれ、ビンごとのタイムスタンプおよびビンごとのタイムスタンプ値と呼ばれる場合がある。30

#### 【0114】

[0121]同様に、図4のコマンドストリーム52内に含まれるタイムスタンプ要求は、GPUドライバ28によって生成されるビンごとのタイムスタンプ要求からそのようなタイムスタンプ要求を区別するために、アプリケーションによって生成されたタイムスタンプ要求と呼ばれる場合がある。アプリケーションによって生成されたタイムスタンプ要求に応答して生成されるタイムスタンプおよびタイムスタンプ値は、それぞれ、アプリケーションによって生成されたタイムスタンプおよびアプリケーションによって生成されたタイムスタンプ値と呼ばれる場合がある。40

#### 【0115】

[0122]いくつかの例では、ビンごとのタイムスタンプ要求は、コマンドストリーム52内でソフトウェアアプリケーション24からのアプリケーションによって生成されたタイ50

ムスタンプ要求を受け取ることに応答して、GPUドライバ28によって生成され得る。たとえば、GPUドライバ28は、コマンドストリーム52内でアプリケーションによって生成されたタイムスタンプ要求TSR1に出会うことに応答して、BINごとのタイムスタンプ要求TSR1AとTSR1BとTSR1CとTSR1Dとを生成することができる。別の例として、GPUドライバ28は、コマンドストリーム52内でアプリケーションによって生成されたタイムスタンプ要求TSR3に出会うことに応答して、BINごとのタイムスタンプ要求TSR3AとTSR3BとTSR3CとTSR3Dとを生成することができる。

#### 【0116】

[0123]いくつかの例では、TSRGPUタイムスタンプ要求は、コマンドストリーム52内でアプリケーションによって生成されたタイムスタンプ要求に出会うことに基づいてまたはこれに応答して生成されるのではないものとすることができます。その代わりに、そのような例では、GPUドライバ28は、レンダリングパス中に実行されるレンダリングパスインスタンスごとにTSRGPUタイムスタンプ要求を自動的に生成することができます。

10

#### 【0117】

[0124]BINごとのタイムスタンプ要求の各々の後の数は、そのBINごとのタイムスタンプ要求に関連するアプリケーションによって生成されたタイムスタンプ要求を示す。BINごとのタイムスタンプ要求は、アプリケーションによって生成されたコマンドストリーム(たとえば、コマンドストリーム52)内でアプリケーションによって生成されたタイムスタンプ要求に出会うことに基づいてまたはこれに応答して、GPUドライバ28によって生成される内のBINごとのタイムスタンプ要求の場合に、アプリケーションによって生成されたタイムスタンプ要求に関連付けられ得る。たとえば、BINごとのタイムスタンプ要求TSR1AとTSR1BとTSR1CとTSR1Dとは、コマンドストリーム52内のアプリケーションによって生成されたタイムスタンプ要求TSR1に関連付けられる。

20

#### 【0118】

[0125]BINごとのタイムスタンプ要求の各々の後の文字は、それぞれのBINごとのタイムスタンプ要求に関連する実行タイムライン54内のレンダリングパス反復を示す。タイムスタンプ要求は、そのタイムスタンプ要求がレンダリングパス反復中に発行され、および/またはサービスされる場合に、そのレンダリングパス反復に関連付けられ得る。たとえば、TSR1Aは、アプリケーションによって生成されたタイムスタンプ要求TSR1と、実行タイムライン54内の第1のレンダリングパス反復(すなわち、レンダリングパス反復A)とに関連するBINごとのタイムスタンプ要求を示す。

30

#### 【0119】

[0126]BINごとの描画呼出しの各々の後の数は、それぞれのBINごとの描画呼出しに関連するコマンドストリーム52内の描画呼出しコマンドを示す。BINごとの描画呼出しの各々の後の文字は、それぞれのBINごとの描画呼出しに関連する実行タイムライン54内のレンダリングパス反復を示す。

#### 【0120】

[0127]たとえば、「DRAW1A」は、コマンドストリーム52内の第1の描画呼出しコマンド(すなわち、DRAW1)と、実行タイムライン54内の第1のレンダリングパス反復(すなわち、レンダリングパス反復A)とに関連するBINごとの描画呼出しを示す。別の例として、「DRAW2C」は、コマンドストリーム52内の第2の描画呼出しコマンド(すなわち、DRAW2)と、実行タイムライン54内の第3のレンダリングパス反復(すなわち、レンダリングパス反復C)とに関連するBINごとの描画呼出しを示す。

40

#### 【0121】

[0128]BINごとの描画呼出しは、そのBINごとの描画呼出しが描画呼出しコマンドを実行するために描画呼出しコマンドに基づいて生成される場合に、その描画呼出しコマンドに関連付けられ得る。BINごとの描画呼出しは、そのBINごとの描画呼出しがレンダリングパス反復中に実行される場合に、そのレンダリングパス反復に関連付けられ得る。

50

## 【0122】

[0129]図4に似て、図6のコマンドストリーム56、58、60、および62の各々は、コマンドの順序付けられたシーケンスを含む。たとえば、コマンドストリーム56は、第1のピンごとのタイムスタンプ要求(TSRGPA)と、それに続く第2のピンごとのタイムスタンプ要求(TSR1A)と、それに続く第1のピンごとの描画呼出し(DRAW1A)と、それに続く第3のピンごとのタイムスタンプ要求(TSR2A)と、それに続く第2のピンごとの描画呼出し(DRAW2A)と、それに続く第4のピンごとのタイムスタンプ要求(TSR3A)と、それに続く第3のピンごとの描画呼出し(DRAW3A)と、それに続く第5のピンごとのタイムスタンプ要求(TSR4A)とを含む。

## 【0123】

10

[0130]コマンドストリーム56、58、60、および62の各々は、レンダリングパス反復のうちのそれぞれのレンダリングパス反復中に実行される。たとえば、コマンドストリーム56は、「レンダリングパス反復A」中に実行され、コマンドストリーム58は、「レンダリングパス反復B」中に実行され、コマンドストリーム60は、「レンダリングパス反復C」中に実行され、コマンドストリーム62は、「レンダリングパス反復D」中に実行される。

## 【0124】

20

[0131]レンダリングパス反復の各々は、複数のピン(たとえば、レンダターゲットの副領域)のうちのそれぞれのピンをレンダリングするように構成される。たとえば、図6に示されているように、「レンダリングパス反復A」は、レンダターゲットの「ピンA」をレンダリングするように構成され、「レンダリングパス反復B」は、レンダターゲットの「ピンB」をレンダリングするように構成され、「レンダリングパス反復C」は、レンダターゲットの「ピンC」をレンダリングするように構成され、「レンダリングパス反復D」は、レンダターゲットの「ピンD」をレンダリングするように構成される。

## 【0125】

30

[0132]図4に示されたコマンドストリーム52の受取に応答して、GPUドライバ28は、コマンドストリーム52内に含まれる描画呼出しこmandoの各々に関する複数のピンごとの描画呼出しと、コマンドストリーム52内に含まれるタイムスタンプ要求の各々に関する複数のピンごとのタイムスタンプ要求とを生成することができる。たとえば、コマンドストリーム52内のDRAW1に出会うことに応答して、GPUドライバ28は、DRAW1AとDRAW1BとDRAW1CとDRAW1Dとを生成することができる。別の例として、コマンドストリーム52内のTSR2に出会うことに応答して、GPUドライバ28は、TSR2AとTSR2BとTSR2CとTSR2Dとを生成することができる。

## 【0126】

40

[0133]GPUドライバ28は、GPUドライバ28によって生成されたピンごとの描画呼出しとピンごとのタイムスタンプ要求とに基づいて、複数レンダリングパス反復固有のコマンドストリーム56と58と60と62とを生成することもできる。たとえば、GPUドライバ28は、ピンごとの描画呼出しとピンごとのタイムスタンプ要求とをピンによって一緒にグループ化し、ピンごとの描画呼出しとピンごとのタイムスタンプ要求との各グループをコマンドストリーム56と58と60と62とのうちの別々の1つに配置することができる。

## 【0127】

[0134]いくつかの例では、GPUドライバ28は、図4のコマンドストリーム52内のアプリケーションによって生成されたタイムスタンプ要求に関連しないコマンドストリーム56と58と60と62との各々の始めにタイムスタンプ要求を配置することもできる。これらのタイムスタンプ要求は、いくつかの例で、アプリケーションによって生成されたタイムスタンプ要求のタイムスタンプ値を生成する時に基準タイムスタンプとして働くことができる。

## 【0128】

50

[0135] G P U ドライバ 2 8 は、 G P U 1 2 に、 コマンドストリーム 5 6 と 5 8 と 6 0 と 6 2 とに基づいて複数のレンダリングパス反復を実行させることができる。たとえば、 G P U ドライバ 2 8 は、 G P U 1 2 に、 別々のレンダリングパス反復中にコマンドストリーム 5 6 と 5 8 と 6 0 と 6 2 との各々を実行させることができる。

#### 【 0 1 2 9 】

[0136] いくつかの例では、 コマンドストリーム 5 6 と 5 8 と 6 0 と 6 2 を実行する時に、 G P U 1 2 は、 コマンドストリーム 5 6 と 5 8 と 6 0 と 6 2 と内でビンごとのタイムスタンプ要求の各々に出会うことに応答して、 C P U 6 ( たとえば、 G P U ドライバ 2 8 ) にビンごとのタイムスタンプを発行することができる。 G P U 1 2 によって発行されるビンごとのタイムスタンプは、 コマンドストリーム 5 6 と 5 8 と 6 0 と 6 2 とのうちの 1 つを実行する時に G P U 1 2 がビンごとのタイムスタンプ要求に出会った時刻に対応するタイムスタンプ値を含むことができる。そのような例では、 G P U 1 2 からのビンごとのタイムスタンプおよび / またはビンごとのタイムスタンプ値の受取に応答して、 G P U ドライバ 2 8 は、 G P U 1 2 から受け取られたビンごとのタイムスタンプおよび / またはビンごとのタイムスタンプ値に基づいて 1 つまたは複数のアプリケーションによって要求されたタイムスタンプ値を生成し、 このアプリケーションによって要求されたタイムスタンプ値をソフトウェアアプリケーション 2 4 に供給することができる。  
10

#### 【 0 1 3 0 】

[0137] さらなる例では、 コマンドストリーム 5 6 と 5 8 と 6 0 と 6 2 を実行している間に、 G P U 1 2 は、 ビンごとのタイムスタンプ要求に対応するビンごとのタイムスタンプ値を内部で記録することができる。そのような例では、 G P U 1 2 は、 いくつかの例で、 G P U 1 2 によって内部で追跡されるビンごとのタイムスタンプ値に基づいて、 1 つまたは複数のアプリケーションによって要求されたタイムスタンプおよび / または 1 つまたは複数のアプリケーションによって要求されたタイムスタンプ値を生成し、 そのアプリケーションによって要求されたタイムスタンプおよび / またはタイムスタンプ値を G P U ドライバ 2 8 に供給することができる。アプリケーションによって要求されたタイムスタンプおよび / またはタイムスタンプ値の受取に応答して、 G P U ドライバ 2 8 は、 アプリケーションによって生成されたタイムスタンプを生成し、 および / またはこれをソフトウェアアプリケーション 2 4 に供給することができる。  
20

#### 【 0 1 3 1 】

[0138] G P U 1 2 がビンごとのタイムスタンプ値を内部で追跡する追加の例では、 G P U 1 2 は、 いくつかの例で、 1 つまたは複数の中間時刻値を生成し、 その 1 つまたは複数の中間時刻値を G P U ドライバ 2 8 に供給することができる。中間時刻値の受取に応答して、 G P U ドライバ 2 8 は、 アプリケーションによって生成されたタイムスタンプを生成し、 そのアプリケーションによって生成されたタイムスタンプをソフトウェアアプリケーション 2 4 に供給することができる。  
30

#### 【 0 1 3 2 】

[0139] いくつかの例では、 図 4 に示されたコマンドストリーム 5 2 について、 コマンドストリーム 5 2 内に含まれるタイムスタンプ要求の各々について G P U ドライバ 2 8 によって返されるタイムスタンプ内に含まれるタイムスタンプ値は、 表 1 にリストされた式に基づいて生成され得る。  
40

#### 【 0 1 3 3 】

【表1】

$TSV1 = TSVGPUA + (TSV1A - TSVGPUA) + (TSV1B - TSVGPUB) + (TSV1C - TSVGPUC) + (TSV1D - TSVGPUD)$	(1)
$TSV2 = TSVGPUA + (TSV2A - TSVGPUA) + (TSV2B - TSVGPUB) + (TSV2C - TSVGPUC) + (TSV2D - TSVGPUD)$	(2)
$TSV3 = TSVGPUA + (TSV3A - TSVGPUA) + (TSV3B - TSVGPUB) + (TSV3C - TSVGPUC) + (TSV3D - TSVGPUD)$	(3)
$TSV4 = TSVGPUA + (TSV4A - TSVGPUA) + (TSV4B - TSVGPUB) + (TSV4C - TSVGPUC) + (TSV4D - TSVGPUD)$	(4)

表1：タイムスタンプ値を生成するための例の式

## 【0134】

[0140]表1では、 $TSVx$ は、タイムスタンプ要求 $TSRx$ に応答して生成されるタイムスタンプ値であり、 $TSVGPUs$ は、タイムスタンプ要求 $TSRGPs$ に応答して生成されるタイムスタンプ値である。たとえば、 $TSV1$ は、タイムスタンプ要求 $TSR1$ に応答して生成されるタイムスタンプのタイムスタンプ値であり、 $TSV3B$ は、タイムスタンプ要求 $TSR3B$ に応答して生成されるタイムスタンプのタイムスタンプ値である。別の例として、 $TSVGPUA$ は、タイムスタンプ要求 $TSRGPUA$ に応答して生成されるタイムスタンプのタイムスタンプ値であり、 $TSVGPUB$ は、タイムスタンプ要求 $TSRG PUB$ に応答して生成されるタイムスタンプのタイムスタンプ値である。

## 【0135】

[0141]さらなる例では、コマンドストリーム52内に含まれるタイムスタンプ要求の各々についてGPUドライバ28によって返されるタイムスタンプ内に含まれるタイムスタンプ値は、次の包括的な式に基づいて生成され得る。

## 【0136】

## 【数1】

$$TSVx = TSVGPUA + (TSVxA - TSVGPUA) + (TSVxB - TSVGPUB) + (TSVxC - TSVGPUC) + (TSVxD - TSVGPUD) \quad (5)$$

## 【0137】

ただし、 $TSVx$ は、タイムスタンプ要求 $TSRx$ に応答して生成されるタイムスタンプのタイムスタンプ値であり、 $TSVxA$ は、タイムスタンプ要求 $TSRx A$ に応答して生成されるタイムスタンプのタイムスタンプ値であるなどである。式(5)では、 $x$ は、任意の整数とすることができます。

## 【0138】

[0142]いくつかの例では、 $TSRx A$ 、 $TSRx B$ 、 $TSRx C$ 、および $TSRx D$ は、コマンドストリーム52内で $TSRx$ を受け取ることに応答して生成されるタイムスタンプ要求とすることができます、 $TSRGPUA$ 、 $TSRG PUB$ 、 $TSRG PUC$ 、および $TSRG PUD$ は、それぞれのレンダリングパス反復の始めにGPUドライバ28によって生成されるタイムスタンプ要求とすることができます。

10

20

30

40

50

**【 0 1 3 9 】**

[0143] 追加の例では、CPU6 および / または GPU12 は、次式に基づいて 1 つまたは複数のタイムスタンプ値を生成することができる。

**【 0 1 4 0 】****【 数 2 】**

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y)) \quad (6)$$

10

**【 0 1 4 1 】**

ただし、Value は、グラフィックスアプリケーションから受け取られたタイムスタンプ要求に応答して生成されるタイムスタンプのタイムスタンプ値であり、TSV(y) は、第 y のレンダリングパス反復中に生成されるビンごとのタイムスタンプ値であり、TSVGPU(y) は、第 y のレンダリングパス反復について何らかのプリミティブをレンダリングする前に生成される基準タイムスタンプ値であり、N は、グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である。いくつかの例では、式 (6) は、表 1 の式の一般化された形とができる。

**【 0 1 4 2 】**

[0144] いくつかの場合に、TSV(y) ビンごとのタイムスタンプ値の各々は、複数のビンごとのタイムスタンプ要求のうちのそれぞれの 1 つに対応することができ、複数のビンごとのタイムスタンプ要求の各々は、Value に対応するグラフィックスアプリケーションから受け取られたタイムスタンプ要求に対応し、これに応答して生成される。

20

**【 0 1 4 3 】**

[0145]さらなる例では、CPU6 および / または GPU12 は、次式に基づいて、グラフィックスアプリケーションから受け取られたコマンドストリーム内に含まれる複数のタイムスタンプ要求のタイムスタンプ値を生成することができる。

**【 0 1 4 4 】****【 数 3 】**

30

$$Value(x) = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(x, y) - TSVGPU(y)) \quad (7)$$

**【 0 1 4 5 】**

ただし、Value(x) は、グラフィックスアプリケーションから受け取られたコマンドストリーム内の第 x のタイムスタンプ要求に応答して生成されるタイムスタンプのタイムスタンプ値であり、TSV(x, y) は、第 y のレンダリングパス反復中に生成され、コマンドストリーム内の第 x のタイムスタンプ要求に対応し、これに応答して生成される、ビンごとのタイムスタンプ値であり、TSVGPU(y) は、第 y のレンダリングパス反復について何らかのプリミティブをレンダリングする前に生成される基準タイムスタンプ値であり、N は、グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である。いくつかの場合に、TSV(x, y) ビンごとのタイムスタンプ値の各々は、複数のビンごとのタイムスタンプ要求のうちのそれぞれの 1 つに対応することができ、複数のビンごとのタイムスタンプ要求の各々は、グラフィックスアプリケーションから受け取られたコマンドストリーム内の第 x のタイムスタンプ要求に対応し、これに応答して生成される。いくつかの例では、式 (7) は、表 1 の式の一般化された形とができる。

40

**【 0 1 4 6 】**

[0146] いくつかの例では、グラフィックスアプリケーションから受け取られるコマンド

50

ストリーム内に含まれるタイムスタンプ要求の各々について、G P U ドライバ2 8は、ピンごとのコマンドストリームの各々内にピンごとのタイムスタンプ要求を配置することができる。そのような例では、ピンごとのコマンドストリームの各々内のピンごとのタイムスタンプ要求の順序は、いくつかの例で、グラフィックスアプリケーションから受け取られるコマンドストリーム内の対応するタイムスタンプ要求の順序と同一とすることができる。

#### 【 0 1 4 7 】

[0147]さらなる例では、グラフィックスアプリケーションから受け取られるコマンドストリームは、描画呼出しコマンドとタイムスタンプ要求とを含むことができる。いくつかの場合に、タイムスタンプ要求のうちの少なくとも1つは、コマンドストリーム内で描画呼出しコマンドのうちの少なくとも2つの間に位置決めされ得る。そのような例では、グラフィックスアプリケーションから受け取られるコマンドストリーム内のタイムスタンプ要求の各々について、G P U ドライバ2 8は、ピンごとのコマンドストリームの各々内にピンごとのタイムスタンプ要求を配置することができる。さらに、グラフィックスアプリケーションから受け取られるコマンドストリーム内の描画呼出しコマンドの各々について、G P U ドライバ2 8は、ピンごとのコマンドストリームの各々内にピンごとの描画呼出しを配置することができる。そのような例では、ピンごとのコマンドストリームの各々内のピンごとのタイムスタンプ要求とピンごとの描画呼出しとの順序は、いくつかの例で、グラフィックスアプリケーションから受け取られたコマンドストリーム内の対応するタイムスタンプ要求と描画呼出しコマンドとの順序と同一とすることができる。

10

20

#### 【 0 1 4 8 】

[0148]一般に、G P U ドライバ2 8とG P U 1 2との一方または両方は、ピンごとのタイムスタンプ値に基づいて、アプリケーションによって要求されたタイムスタンプおよび／またはアプリケーションによって要求されたタイムスタンプ値を生成することができる。いくつかの例では、G P U ドライバ2 8は、G P U 1 2からピンごとのタイムスタンプ値を受け取り、ピンごとのタイムスタンプ値に基づいて、アプリケーションによって要求されたタイムスタンプおよび／またはタイムスタンプ値を生成することができる。そのような例では、G P U ドライバ2 8は、いくつかの例で、式(1)～(7)のうちの1つまたは複数に基づいて、アプリケーションによって要求されたタイムスタンプおよび／またはタイムスタンプ値を生成することができる。

30

#### 【 0 1 4 9 】

[0149]さらなる例では、G P U 1 2は、ピンごとのタイムスタンプ値を生成し、ピンごとのタイムスタンプ値に基づいてアプリケーションによって要求されたタイムスタンプおよび／またはタイムスタンプ値を生成し、アプリケーションによって要求されたタイムスタンプおよび／またはタイムスタンプ値をG P U ドライバ2 8に供給することができる。そのような例では、G P U 1 2は、いくつかの例で、式(1)～(7)のうちの1つまたは複数に基づいて、アプリケーションによって要求されたタイムスタンプおよび／またはタイムスタンプ値を生成することができる。

#### 【 0 1 5 0 】

[0150]追加の例では、G P U 1 2は、ピンごとのタイムスタンプ値を生成し、ピンごとのタイムスタンプ値に基づいて1つまたは複数の中間値を生成し、中間値をG P U ドライバ2 8に供給することができる。そのような例では、G P U ドライバ2 8は、G P U 1 2から中間値を受け取り、中間値に基づいて、アプリケーションによって要求されたタイムスタンプおよび／またはタイムスタンプ値を生成することができる。

40

#### 【 0 1 5 1 】

[0151]いくつかの場合に、中間値は、式(1)～(7)内の1つまたは複数の項に対応することができる。追加の場合に、中間値は、式(1)～(7)内で指定される入力変数の任意の組合せに対応することができる。

#### 【 0 1 5 2 】

[0152]図4～6に示されているように、C P U 6および／またはG P U 1 2は、グラフ

50

イックスフレームに関するタイルベースのレンダリングを実行している間にGPU12によって生成される複数のピンごとのタイムスタンプ値（たとえば、TSR2AとTSR2BとTSR2CとTSR2Dに対応するピンごとのタイムスタンプ値）に基づいて時点を示すタイムスタンプ値（たとえば、TSR2に対応するタイムスタンプ値）を生成することができる。タイムスタンプ値（たとえば、TSR2に対応するタイムスタンプ値）は、複数のピンごとのタイムスタンプ値のうちの少なくとも2つのピンごとのタイムスタンプ値（たとえば、TSR2AとTSR2BとTSR2CとTSR2Dに対応するピンごとのタイムスタンプ値）の関数とすることができます。

【 0 1 5 3 】

[0153]いくつかの例では、複数のピンごとのタイムスタンプ値（たとえば、T S R 2 A と T S R 2 B と T S R 2 C と T S R 2 D に対応するピンごとのタイムスタンプ値）の各々は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つの間に G P U 1 2 によって生成され得る。たとえば、T S R 2 A のピンごとのタイムスタンプ値は、レンダリングパス反復 A 中に G P U 1 2 によって生成され、T S R 2 B のピンごとのタイムスタンプ値は、レンダリングパス反復 B 中に G P U 1 2 によって生成され、T S R 2 C のピンごとのタイムスタンプ値は、レンダリングパス反復 C 中に G P U 1 2 によって生成され、T S R 2 D のピンごとのタイムスタンプ値は、レンダリングパス反復 D 中に G P U 1 2 によって生成される。

【 0 1 5 4 】

[0154]レンダリングパス反復の各々は、いくつかの例で、レンダターゲットの複数の副領域のうちのそれぞれの1つをレンダリングするように構成され得る。たとえば、レンダリングパス反復Aは、レンダターゲットのBIN Aをレンダリングするように構成され、レンダリングパス反復Bは、レンダターゲットのBIN Bをレンダリングするように構成され、レンダリングパス反復Cは、レンダターゲットのBIN Cをレンダリングするように構成され、レンダリングパス反復Dは、レンダターゲットのBIN Dをレンダリングするように構成される。

〔 0 1 5 5 〕

[0155]いくつかの例では、タイムスタンプ値（たとえば、T S R 2 に対応するタイムスタンプ値）を生成するのに使用される少なくとも2つのピンごとのタイムスタンプ値は、第1のピンごとのタイムスタンプ値（たとえば、T S R 2 A に対応するタイムスタンプ値）と、第2のピンごとのタイムスタンプ値（たとえば、T S R 2 B に対応するタイムスタンプ値）とを含むことができる。第1のピンごとのタイムスタンプ値（たとえば、T S R 2 A に対応するタイムスタンプ値）は、複数のレンダリングパス反復のうちの第1のレンダリングパス反復（たとえば、レンダリングパス反復A）中にG P U 1 2 によって生成され得る。第2のピンごとのタイムスタンプ値（たとえば、T S R 2 B に対応するタイムスタンプ値）は、複数のレンダリングパス反復のうちの第2のレンダリングパス反復（たとえば、レンダリングパス反復B）中にG P U 1 2 によって生成され得る。第2のレンダリングパス反復（たとえば、レンダリングパス反復B）は、第1のレンダリングパス反復（たとえば、レンダリングパス反復A）とは異なるものとすることができます。

[ 0 1 5 6 ]

[0156]さらなる例では、タイムスタンプ値（たとえば、T S R 2 に対応するタイムスタンプ値）を生成するのに使用される少なくとも 2 つのビンごとのタイムスタンプ値は、少なくとも 2 つの基準タイムスタンプ値（たとえば、T S R G P U A と T S R G P U B と T S R G P U C と T S R G P U D とに対応するタイムスタンプ値）をさらに含むことができる。少なくとも 2 つの基準タイムスタンプ値の各々は、レンダリングパス反復のうちのそれぞれの 1 つ中に、レンダリングパス反復のうちのそれぞれの 1 つについて何らかのプリミティブをレンダリングする前に、生成され得る。たとえば、T S R G P U A に対応するタイムスタンプ値は、レンダリングパス反復 A 中に、レンダリングパス反復 A について何らかのプリミティブをレンダリングする前に（たとえば、D R A W 1 A と D R A W 2 A と D R A W 3 A とを実行する前に）生成される。

**【 0 1 5 7 】**

[0157]追加の例では、CPU6および/またはGPU12は、グラフィックスフレームに関して実行されるコマンドの順序付けられたシーケンス（コマンドストリーム52）内の少なくとも2つの描画呼出しコマンド（たとえば、DRAW1、DRAW2）の間に位置決めされたタイムスタンプ要求（たとえば、TSR2）に応答して、タイムスタンプ値（たとえば、TSR2に対応するタイムスタンプ値）を生成することができる。いくつかの例では、TSR2のタイムスタンプ値は、グラフィックスフレームのレンダリング中に実行するのにコマンドの順序付けられたシーケンス内の描画呼出しコマンド（たとえば、DRAW1、DRAW2）によって費やされる時間の相対的な長さに基づいて生成され得る。

10

**【 0 1 5 8 】**

[0158]いくつかの例では、GPU12は、グラフィックスフレームに関するタイルベースのレンダリングを実行している間に、複数のレンダリングパス反復を実行することができる。レンダリングパス反復の各々は、少なくとも2つのピンごとの描画呼出しを実行するように構成され得る。たとえば、レンダリングパス反復Aは、DRAW1AとDRAW2Aとを実行するように構成され得る。少なくとも2つのピンごとの描画呼出しの各々は、少なくとも2つの描画呼出しコマンドのうちのそれぞれの1つに関連付けられ得る。たとえば、DRAW1Aは、コマンドストリーム52内のDRAW1に関連付けられ得、DRAW2Aは、コマンドストリーム52内のDRAW2に関連付けられ得る。

**【 0 1 5 9 】**

20

[0159]そのような例では、タイムスタンプ値（たとえば、TSR2に対応するタイムスタンプ値）を生成するのに使用される少なくとも2つのピンごとのタイムスタンプ値は、第1のピンごとのタイムスタンプ値（たとえば、TSR2Aに対応するタイムスタンプ値）と、第2のピンごとのタイムスタンプ値（たとえば、TSR2Bに対応するタイムスタンプ値）とを含むことができる。第1のピンごとのタイムスタンプ値（たとえば、TSR2Aに対応するタイムスタンプ値）は、複数のレンダリングパス反復のうちの第1のレンダリングパス反復（たとえば、レンダリングパス反復A）中に少なくとも2つのピンごとの描画呼出し（たとえば、DRAW1A、DRAW2A）の間に発生する時点を示すものとすることができます。第2のピンごとのタイムスタンプ値（たとえば、TSR2Bに対応するタイムスタンプ値）は、複数のレンダリングパス反復のうちの第2のレンダリングパス反復（たとえば、レンダリングパス反復B）中に少なくとも2つのピンごとの描画呼出し（たとえば、DRAW1B、DRAW2B）の間に発生する時点を示すものとすることができます。第2のレンダリングパス反復（たとえば、レンダリングパス反復B）は、第1のレンダリングパス反復（たとえば、レンダリングパス反復A）とは異なるものとされ得る。

30

**【 0 1 6 0 】**

[0160]さらなる例では、複数のピンごとのタイムスタンプ値（たとえば、TSR2AとTSR2BとTSR2CとTSR2Dに対応するピンごとのタイムスタンプ値）の各々は、複数のピンごとのタイムスタンプ要求のそれぞれの1つ（たとえば、それぞれTSR2AとTSR2BとTSR2CとTSR2Dと）に応答して生成され得る。そのような例では、ピンごとのタイムスタンプ要求の各々（たとえば、TSR2AとTSR2BとTSR2CとTSR2Dと）は、複数のコマンドストリームのうちのそれぞれの1つ（たとえば、それぞれコマンドストリーム56と58と60と62と）内に配置され得る。コマンドストリームの各々（たとえば、それぞれコマンドストリーム56と58と60と62と）は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの1つ（たとえば、それぞれレンダリングパス反復AとBとCとDと）中にGPU12によって実行され得る。レンダリングパス反復の各々（たとえば、レンダリングパス反復AとBとCとDと）は、レンダターゲットの複数の副領域のうちのそれぞれの1つ（たとえば、それぞれピンAとBとCとDと）をレンダリングするように構成され得る。

40

50

**【 0 1 6 1 】**

[0161]いくつかの例では、CPU6および/またはGPU12は、複数のビンごとのタイムスタンプ値（たとえば、TSR1A - TSR1DとTSR2A - TSR2DとTSR3A - TSR3DとTSR4A - TSR4Dに対応するビンごとのタイムスタンプ値）に基づいて、複数のタイムスタンプ値（たとえば、TSR1とTSR2とTSR3とTSR4に対応するタイムスタンプ値）を生成することができる。タイムスタンプ値の各々（たとえば、TSR1とTSR2とTSR3とTSR4に対応するタイムスタンプ値）は、グラフィックスフレームに関して実行されるコマンドの順序付けられたシーケンス（たとえば、コマンドストリーム52）内に含まれる複数のタイムスタンプ要求のうちのそれぞれの1つ（たとえば、それぞれTSR1とTSR2とTSR3とTSR4と）に対応することができる。タイムスタンプ要求の各々は、グラフィックスアプリケーションによって要求され得る。

10

**【 0 1 6 2 】**

[0162]そのような例では、タイムスタンプ要求のうちの少なくとも2つ（たとえば、TSR2とTSR3と）は、いくつかの例で、グラフィックスフレームに関して実行されるコマンドの順序付けられたシーケンス（たとえば、コマンドストリーム52）内の連続する描画呼出しコマンドのそれぞれの対（たとえば、それぞれDRAW1 / DRAW2とDRAW2 / DRAW3と）の間に位置決めされ得る。そのような例では、CPU6および/またはGPU12は、コマンドの順序付けられたシーケンス内のタイムスタンプ要求に関する返されるタイムスタンプ値が、コマンドの順序付けられたシーケンス（たとえば、コマンドストリーム52）の始めからコマンドの順序付けられたシーケンス（たとえば、コマンドストリーム52）の終りまで値において単調に増加するように、ビンごとのタイムスタンプ値に基づいて複数のタイムスタンプ値（たとえば、TSR1とTSR2とTSR3とTSR4に対応するタイムスタンプ値）を生成することができる。たとえば、TSR1は、TSR2以下とすることができます、TSR2は、TSR3以下とすることができます、TSR3は、TSR4以下とすることができます。

20

**【 0 1 6 3 】**

[0163]そのような例では、複数のタイムスタンプ値（たとえば、TSR1とTSR2とTSR3とTSR4に対応するタイムスタンプ値）は、いくつかの例で、グラフィックスフレームのレンダリング中に実行するのにコマンドの順序付けられたシーケンス（たとえば、コマンドストリーム52）内の描画呼出しコマンド（たとえば、DRAW1とDRAW2とDRAW3と）によって費やされる時間の相対的な長さを示すものとすることができます。たとえば、グラフィックスフレームのレンダリング中に実行するのにDRAW1によって費やされる時間の長さが、グラフィックスフレームのレンダリング中に実行するのにDRAW2によって費やされる時間の長さより長い場合には、TSR2とTSR1との間の差は、いくつかの例で、TSR3とTSR2との間の差より大きいものとすることができます。

30

**【 0 1 6 4 】**

[0164]いくつかの場合に、グラフィックスフレームのレンダリング中に実行するのにDRAW1によって費やされる時間の長さは、グラフィックスフレームのレンダリング中に実行するのにDRAW1に関連するビンごとの描画呼出しコマンドの各々について費やされる時間の集計長さに対応することができ、グラフィックスフレームのレンダリング中に実行するのにDRAW2によって費やされる時間の長さは、グラフィックスフレームのレンダリング中に実行するのにDRAW2に関連するビンごとの描画呼出しコマンドの各々について費やされる時間の集計長さに対応することができる。たとえば、グラフィックスフレームのレンダリング中に実行するのにDRAW1によって費やされる時間の長さは、実行するのにDRAW1AとDRAW1BとDRAW1CとDRAW1Dによって費やされる時間の長さの合計と等しいものとすることができます、グラフィックスフレームのレンダリング中に実行するのにDRAW2によって費やされる時間の長さは、実行するのにDRAW2AとDRAW2BとDRAW2CとDRAW2Dによって費やされる時間の長

40

50

さの合計と等しいものとすることができます。

#### 【0165】

[0165]いくつかの例では、GPUドライバ28は、グラフィックスアプリケーション（たとえば、ソフトウェアアプリケーション24）からタイムスタンプ要求（たとえば、TSR2）を受け取ることができる。GPUドライバ28は、タイムスタンプ要求（たとえば、TSR2）に基づいて、複数のビンごとのタイムスタンプ要求（たとえば、TSR2AとTSR2BとTSR2CとTSR2Dと）を生成することができる。GPUドライバ28は、複数のコマンドストリームのうちのそれぞれの1つ（たとえば、それぞれコマンドストリーム56と58と60と62と）内にビンごとのタイムスタンプ要求の各々（たとえば、TSR2AとTSR2BとTSR2CとTSR2Dと）を配置することができる。  
10  
。コマンドストリームの各々（たとえば、コマンドストリーム56と58と60と62と）は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの1つ（たとえば、それぞれレンダリングパス反復AとBとCとDと）中にGPU12によって実行され得る。レンダリングパス反復の各々（たとえば、レンダリングパス反復AとBとCとDと）は、レンダターゲットの複数の副領域のうちのそれぞれの1つ（たとえば、それぞれビンAとBとCとDと）をレンダリングするように構成され得る。

#### 【0166】

[0166]GPUドライバ28は、GPU12に、コマンドストリーム（たとえば、コマンドストリーム56と58と60と62と）を実行させることができる。GPUドライバ28および／またはGPU12は、コマンドストリーム内に配置されたビンごとのタイムスタンプ要求（たとえば、TSR2AとTSR2BとTSR2CとTSR2Dと）に応答してGPU12によって生成されたビンごとのタイムスタンプ値に基づいて、タイムスタンプ値（たとえば、TSR2）を生成することができる。  
20

#### 【0167】

[0167]図4～6は、グラフィックスフレームをレンダリングするための3つの描画呼出しコマンドを含む例のコマンドストリームと、レンダターゲットを4つの異なる副領域またはビンに副分割する例のタイルベースのレンダリングシステムとを示す。しかしながら、本開示で説明される技法が、レンダリングされるグラフィックスフレームごとに同一のまたは異なる個数の描画呼出しコマンドを含むコマンドストリームと、グラフィックスフレームを同一のまたは異なる個数の副領域またはビンに副分割するタイルベースのレンダリングシステムとを用いて実施され得ることを理解されたい。  
30

#### 【0168】

[0168]いくつかの例では、ビニングパスは、図5に示されたレンダリングパス反復の前に実行され得る。そのような例では、ビニングパスは、レンダリングパス反復（たとえば、「レンダリングパス反復A」）に類似する形で扱われ得る。そのような例では、GPUドライバ28および／またはGPU12は、ビニングパス中に要求された1つまたは複数のビニングパスごとのタイムスタンプに基づき、レンダリングパスの1つまたは複数の反復中に要求された1つまたは複数のビンごとのタイムスタンプ要求に基づいて、1つまたは複数のタイムスタンプを生成することができる。  
40

#### 【0169】

[0169]いくつかの例では、GPUドライバ28および／またはGPU12は、レンダリングされるグラフィックスフレームのタイムスタンプ値のうちの少なくとも1つが、レンダリングされるグラフィックスフレームのタイムスタンプ値のうちの他の少なくとも1つと異なるように、レンダリングされるグラフィックスフレームに関する複数のタイムスタンプ要求に応答して複数のタイムスタンプ値を生成することができる。さらなる例では、GPUドライバ28および／またはGPU12は、タイムスタンプ値が、レンダリングされるグラフィックスフレームに関するコマンドストリームの始めからレンダリングされるグラフィックスフレームに関するコマンドストリームの終りまで単調に増加するよう、レンダリングされるグラフィックスフレームに関する複数のタイムスタンプ要求に応答し  
50

て複数のタイムスタンプ値を生成することができる。

#### 【0170】

[0170]図7は、本開示に従ってタイルベースのレンダリングを実行するグラフィックス処理システム内でイントラフレームタイムスタンプをサポートする例の技法を示す流れ図である。CPU6および/またはGPU12は、タイムスタンプ要求を受け取る(70)。いくつかの例では、タイムスタンプ要求は、CPU6上で実行されるグラフィックスアプリケーションから受け取られ得る。CPU6および/またはGPU12は、グラフィックスフレームに関するタイルベースのレンダリングを実行している間にGPU12によって生成される複数のピンごとのタイムスタンプ値に基づいて、時点を示すタイムスタンプ値を生成する(72)。タイムスタンプ値は、複数のピンごとのタイムスタンプ値のうちの少なくとも2つのピンごとのタイムスタンプ値の関数とすることができます。  
10

#### 【0171】

[0171]CPU6および/またはGPU12は、タイムスタンプ値を生成するために、本開示で説明される技法のいずれかを使用することができる。いくつかの例では、CPU6および/またはGPU12は、式(1)~(7)のうちの1つまたは複数に基づいてタイムスタンプ値を生成することができる。タイムスタンプ値を生成するのにピンごとのタイムスタンプ値を使用することは、タイルベースのレンダリングを実行するグラフィックス処理システムによってイントラフレームタイムスタンプ要求をサポートすることを可能にすることができる。  
20

#### 【0172】

[0172]図8は、本開示に従ってタイルベースのレンダリングを実行するグラフィックス処理システム内でイントラフレームタイムスタンプをサポートする別の例の技法を示す流れ図である。いくつかの例では、図8に示された技法の一部またはすべてが、図7に示された技法の一部またはすべてを実行するのに使用され得る。

#### 【0173】

[0173]CPU6は、タイムスタンプ要求を受け取る(74)。いくつかの例では、タイムスタンプ要求は、CPU6上で実行されるグラフィックスアプリケーションから受け取られ得る。CPU6は、タイムスタンプ要求に基づいて、複数のピンごとのタイムスタンプ要求を生成する(76)。CPU6は、複数のコマンドストリームのうちのそれぞれの1つ内にピンごとのタイムスタンプ要求の各々を配置する(78)。コマンドストリームの各々は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの1つ中にGPU12によって実行され得る。レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの1つをレンダリングするように構成され得る。  
30

#### 【0174】

[0174]CPU6は、GPU12にコマンドストリームを実行させる(80)。CPU6および/またはGPU12は、コマンドストリーム内に配置されたピンごとのタイムスタンプ要求に応答してGPU12によって生成されたピンごとのタイムスタンプ値に基づいてタイムスタンプ値を生成する(82)。CPU6および/またはGPU12は、タイムスタンプ値を生成するために本開示で説明される技法のいずれかを使用することができる。いくつかの例では、CPU6および/またはGPU12は、式(1)~(7)のうちの1つまたは複数に基づいてタイムスタンプ値を生成することができる。  
40

#### 【0175】

[0175]異なるレンダリングパス反復中に生成されるピンごとのタイムスタンプ値を使用することは、グラフィックスフレームのレンダリング中に実行するのに異なる描画呼出しコマンドによって費やされる時間の相対的な長さを少なくともある度合まで反映する、アプリケーションによって要求されたタイムスタンプ値をグラフィックス処理システムが生成することを可能にすることができます。この形で、本開示で説明される技法は、タイルベースのレンダリングシステムが不連続なインターリープされた形で描画呼出しコマンドを実行する場合であっても、タイルベースのレンダリングシステムがイントラフレームタイ  
50

ムスタンプをサポートすることを可能にすることができます。

**【0176】**

[0176] それ自体は性能ペナルティをこうむらない、正確なイントラフレームタイムスタンプは、タイルベースのレンダリングアーキテクチャ上で入手することがむずかしい可能性がある。これは、ビニングと直接レンダリングとの間で動的に切り替えることができるドライバ上ではさらによりむずかしい。

いくつかの例では、本開示の技法は、ビニングと直接レンダリングとの両方について働くことができる、適度に正確で代表的なイントラフレームタイムスタンプを実施することができる。

**【0177】**

10

[0177] いくつかの例では、各タイムスタンプ要求は、ピンごとのタイムスタンプに変更され得る。所与のレンダターゲットに関するレンダリングが処理された後に、ピン開始から各ピンのタイムスタンプまでの平均時間が、生成され、タイムスタンプとして使用され得る。これは、直接レンダリングコンポーネントによって作られるものと同等のタイムスタンプを提供することができる。本開示の技法は、タイルベースのレンダリングGPUがイントラフレームタイムスタンプをサポートすることを可能にすることができます。

**【0178】**

[0178] 本開示で説明される技法は、少なくとも部分的に、ハードウェア、ソフトウェア、ファームウェア、またはその任意の組合せで実施され得る。たとえば、説明される技法の様々な態様は、1つまたは複数のプロマイクロプロセッサ、デジタル信号プロセッサ(DSP)、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)、または任意の同等の集積回路もしくはディスクリート論理回路網を含む1つまたは複数のプロセッサならびにそのようなコンポーネントの任意の組合せ内で実施され得る。「プロセッサ」または「処理回路網」という用語は、一般に、単独でもしくは他の論理回路網と組み合わされた前述の論理回路網のいずれか、あるいは処理を実行するディスクリートハードウェアなどの任意の他の同等の回路網を指すことができる。

20

**【0179】**

[0179] そのようなハードウェア、ソフトウェア、およびファームウェアは、本開示で説明される様々な動作と機能とをサポートするために、同一のデバイス内または別々のデバイス内で実施され得る。さらに、説明されるユニット、モジュール、またはコンポーネントのいずれもが、一緒にまたは別個ではあるがインターフェラブルな論理デバイスとして別々に実施され得る。モジュールまたはユニットとしての異なる特徴の描写は、異なる機能的態様を強調することを意図されたものであって、必ずしも、そのようなモジュールまたはユニットが別々のハードウェアコンポーネントまたはソフトウェアコンポーネントによって実現されなければならないことを暗示するものではない。そうではなく、1つまたは複数のモジュールまたはユニットに関連する機能性は、別々のハードウェア、ファームウェア、および/もしくはソフトウェアコンポーネントによって実行され、または共通のもしくは別々のハードウェアもしくはソフトウェアコンポーネント内で一体化され得る。

30

**【0180】**

[0180] 本開示で説明される技法は、命令を記憶するコンピュータ可読記憶媒体などのコンピュータ可読媒体内に記憶され、実施され、または符号化されることも可能である。コンピュータ可読媒体内に組み込まれまたは符号化された命令は、たとえば命令が1つまたは複数のプロセッサによって実行される時に、その1つまたは複数のプロセッサに、本明細書で説明される技法を実行させることができる。いくつかの例では、コンピュータ可読媒体は、非一時的コンピュータ可読記憶媒体とすることができます。コンピュータ可読記憶媒体は、ランダムアクセスメモリ(RAM)、読み専用メモリ(ROM)、プログラマブル読み専用メモリ(PROM)、消去可能プログラマブル読み専用メモリ(EPROM)、電気的消去可能プログラマブル読み専用メモリ(EEROM)、フラッシュメモリ、ハードディスク、CD-ROM、フロッピ(登録商標)ディスク、カセット、磁気媒体、光学媒体、または有形である他のコンピュータ可読記憶媒体を含むことができる。

40

50

## 【0181】

[0181]コンピュータ可読媒体は、上でリストされたものなどの有形の記憶媒体に対応するコンピュータ可読記憶媒体を含むことができる。コンピュータ可読媒体は、たとえば通信プロトコルに従う、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容易にする任意の媒体を含む通信媒体を備えることもできる。この形で、「コンピュータ可読媒体」という句は、一般に、(1)非一時的である有形のコンピュータ可読記憶媒体と、(2)過渡的信号または搬送波などの非有形のコンピュータ可読通信媒体とに対応することができる。

## 【0182】

[0182]様々な態様および例が、説明された。しかしながら、下の特許請求の範囲の範囲から逸脱せずに、本開示の構造または教示に対して変更を行うことができる。10

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] 1つまたは複数のプロセッサを用いて、グラフィックスフレームに関するタイルベースのレンダリングを実行している間にグラフィックス処理ユニット(GPU)によって生成される複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成することを備え、前記タイムスタンプ値は、前記複数のビンごとのタイムスタンプ値のうちの少なくとも2つのビンごとのタイムスタンプ値の関数である、  
方法。

[C2] 前記複数のビンごとのタイムスタンプ値の各々は、前記タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの1つ中に前記GPUによって生成され、前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの1つをレンダリングするように構成される、C1に記載の方法。20

[C3] 前記少なくとも2つのビンごとのタイムスタンプ値は、第1のビンごとのタイムスタンプ値と第2のビンごとのタイムスタンプ値とを備え、

前記第1のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第1のレンダリングパス反復中に前記GPUによって生成され、

前記第2のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第2のレンダリングパス反復中に前記GPUによって生成され、前記第2のレンダリングパス反復は、前記第1のレンダリングパス反復とは異なる、30

C2に記載の方法。

[C4] 前記少なくとも2つのビンごとのタイムスタンプ値は、少なくとも2つの基準タイムスタンプ値をさらに備え、前記少なくとも2つの基準タイムスタンプ値の各々は、前記レンダリングパス反復のうちのそれぞれの1つ中に、前記レンダリングパス反復のうちの前記それぞれの1つに関する何らかのプリミティブのレンダリングの前に前記GPUによって生成される、C3に記載の方法。

[C5] 前記タイムスタンプ値を生成することは、次式に基づいて前記タイムスタンプ値を生成することを備え、

## 【数4】

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y))$$

Valueは、前記タイムスタンプ値であり、TSV(y)は、第yのレンダリングパス反復中に生成されるビンごとのタイムスタンプ値であり、TSVGPU(y)は、前記第yのレンダリングパス反復に関する何らかのプリミティブのレンダリングの前に前記GPUによって生成される基準タイムスタンプ値であり、Nは、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、C2に記載の方法。40

[C 6] 前記タイムスタンプ値を生成することは、前記グラフィックスフレームについて実行されるコマンドの順序付けられたシーケンス内の少なくとも2つの描画呼出しコマンドの間に位置決めされたタイムスタンプ要求に応答して前記タイムスタンプ値を生成することを備える、C 1に記載の方法。

[C 7] 前記GPUは、前記グラフィックスフレームに関するタイルベースのレンダリングを実行している間に複数のレンダリングパス反復を実行し、

前記少なくとも2つのビンごとのタイムスタンプ値は、第1のビンごとのタイムスタンプ値と第2のビンごとのタイムスタンプ値とを備え、

前記第1のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第1のレンダリングパス反復中に実行される少なくとも2つのビンごとの描画呼出しの実行の間に発生する時点を示し、前記第1のレンダリングパス反復中に実行される前記少なくとも2つのビンごとの描画呼出しの各々は、前記少なくとも2つの描画呼出しコマンドのうちのそれぞれの1つに関連し、

前記第2のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第2のレンダリングパス反復中に実行される少なくとも2つのビンごとの描画呼出しの実行の間に発生する時点を示し、前記第2のレンダリングパス反復中に実行される前記少なくとも2つのビンごとの描画呼出しの各々は、前記少なくとも2つの描画呼出しコマンドのうちのそれぞれの1つに関連し、前記第2のレンダリングパス反復は、前記第1のレンダリングパス反復とは異なる、

C 6に記載の方法。

10

[C 8] 前記複数のビンごとのタイムスタンプ値の各々は、複数のビンごとのタイムスタンプ要求のうちのそれぞれの1つに応答して生成され、前記ビンごとのタイムスタンプ要求の各々は、複数のコマンドストリームのうちのそれぞれの1つ内に配置され、前記コマンドストリームの各々は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの1つ中に前記GPUによって実行され、前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの1つをレンダリングするように構成される、C 1に記載の方法。

[C 9] 前記タイムスタンプ値を生成することは、

前記複数のビンごとのタイムスタンプ値に基づいて複数のタイムスタンプ値を生成すること、前記タイムスタンプ値の各々は、前記グラフィックスフレームについて実行されるコマンドの順序付けられたシーケンス内に含まれる複数のタイムスタンプ要求のうちのそれぞれの1つに対応し、前記タイムスタンプ要求の各々は、グラフィックスアプリケーションによって要求される、

30

を備える、C 1に記載の方法。

[C 10] 前記タイムスタンプ要求のうちの少なくとも2つは、前記グラフィックスフレームについて実行されるコマンドの前記順序付けられた前記順序付けられたシーケンス内で連続する描画呼出しコマンドのそれぞれの対の間に位置決めされ、

前記複数のタイムスタンプ値を生成することは、コマンドの前記順序付けられたシーケンス内の前記タイムスタンプ要求について返される前記タイムスタンプ値が、コマンドの前記順序付けられたシーケンスの始めからコマンドの前記順序付けられたシーケンスの終りまで値において単調に増加するように、前記ビンごとのタイムスタンプ値に基づいて前記複数のタイムスタンプ値を生成することを備える、

40

C 9に記載の方法。

[C 11] 前記複数のタイムスタンプ値は、前記グラフィックスフレームの前記レンダリング中に実行するのにコマンドの前記順序付けられたシーケンス内の描画呼出しコマンドによって費やされる時間の相対的な長さを示す、C 9に記載の方法。

[C 12] 前記複数のタイムスタンプ値を生成することは、次式に基づいて前記複数のタイムスタンプ値を生成することを備え、

【数5】

$$Value(x) = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(x,y) - TSVGPU(y))$$

V a l u e ( x ) は、コマンドの前記順序付けられたシーケンス内の第  $x$  のタイムスタンプ要求に応答して生成されるタイムスタンプのタイムスタンプ値であり、TSV ( x , y ) は、第  $y$  のレンダリングパス反復中に生成され、コマンドの前記順序付けられたシーケンス内の第  $x$  のタイムスタンプ要求に対応する、ビンごとのタイムスタンプ値であり、TSVGPU ( y ) は、前記第  $y$  のレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記 GPU によって生成される基準タイムスタンプ値であり、 $N$  は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、C 9 に記載の方法。

[ C 1 3 ] 前記タイムスタンプ値を生成することは、

グラフィックスアプリケーションからタイムスタンプ要求を受け取ることと、

前記タイムスタンプ要求に基づいて複数のビンごとのタイムスタンプ要求を生成することと、

前記ビンごとのタイムスタンプ要求の各々を複数のコマンドストリームのうちのそれぞれの 1 つ内に配置することと、前記コマンドストリームの各々は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つ中に前記 GPU によって実行され、前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングするように構成され、

前記 GPU に前記コマンドストリームを実行させることと、

前記コマンドストリーム内に配置された前記ビンごとのタイムスタンプ要求に応答して前記 GPU によって生成された前記ビンごとのタイムスタンプ値に基づいて前記タイムスタンプ値を生成することと、

を備える、C 1 に記載の方法。

[ C 1 4 ] 前記 1 つまたは複数のプロセッサは、中央処理装置 ( CPU ) を備える、C 1 に記載の方法。

[ C 1 5 ] 前記 1 つまたは複数のプロセッサは、前記 GPU を備える、C 1 に記載の方法。

[ C 1 6 ] グラフィックスフレームに関するタイルベースのレンダリングを実行している間にグラフィックス処理ユニット ( GPU ) によって生成される複数のビンごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成するように構成された 1 つまたは複数のプロセッサを備え、前記タイムスタンプ値は、前記複数のビンごとのタイムスタンプ値のうちの少なくとも 2 つのビンごとのタイムスタンプ値の関数である、デバイス。

[ C 1 7 ] 前記複数のビンごとのタイムスタンプ値の各々は、前記タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つ中に前記 GPU によって生成され、前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングするように構成される、C 1 6 に記載のデバイス。

[ C 1 8 ] 前記少なくとも 2 つのビンごとのタイムスタンプ値は、第 1 のビンごとのタイムスタンプ値と第 2 のビンごとのタイムスタンプ値とを備え、

前記第 1 のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 1 のレンダリングパス反復中に前記 GPU によって生成され、

前記第 2 のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 2 のレンダリングパス反復中に前記 GPU によって生成され、前記第 2 のレンダリングパス反復は、前記第 1 のレンダリングパス反復とは異なる、

10

20

30

40

50

C 1 7 に記載のデバイス。

[ C 1 9 ] 前記少なくとも 2 つのビンごとのタイムスタンプ値は、少なくとも 2 つの基準タイムスタンプ値をさらに備え、前記少なくとも 2 つの基準タイムスタンプ値の各々は、前記レンダリングパス反復のうちのそれぞれの 1 つ中に、前記レンダリングパス反復のうちの前記それぞれの 1 つに関する何らかのプリミティブのレンダリングの前に前記 GPU によって生成される、C 1 8 に記載のデバイス。

[ C 2 0 ] 前記 1 つまたは複数のプロセッサは、次式に基づいて前記タイムスタンプ値を生成するようにさらに構成され、

【数 6】

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y))$$

10

V a l u e は、前記タイムスタンプ値であり、T S V ( y ) は、第 y のレンダリングパス反復中に生成されるビンごとのタイムスタンプ値であり、T S V G P U ( y ) は、前記第 y のレンダリングパス反復に関する何らかのプリミティブのレンダリングの前に前記 GPU によって生成される基準タイムスタンプ値であり、N は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、C 1 7 に記載のデバイス。

20

[ C 2 1 ] 前記 1 つまたは複数のプロセッサは、前記グラフィックスフレームについて実行されるコマンドの順序付けられたシーケンス内の少なくとも 2 つの描画呼出しコマンドの間に位置決めされたタイムスタンプ要求に応答して前記タイムスタンプ値を生成するようにさらに構成される、C 1 6 に記載のデバイス。

[ C 2 2 ] 前記 GPU は、前記グラフィックスフレームに関するタイルベースのレンダリングを実行している間に複数のレンダリングパス反復を実行し、

前記少なくとも 2 つのビンごとのタイムスタンプ値は、第 1 のビンごとのタイムスタンプ値と第 2 のビンごとのタイムスタンプ値とを備え、

前記第 1 のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 1 のレンダリングパス反復中に実行される少なくとも 2 つのビンごとの描画呼出しの実行の間に発生する時点を示し、前記第 1 のレンダリングパス反復中に実行される前記少なくとも 2 つのビンごとの描画呼出しの各々は、前記少なくとも 2 つの描画呼出しコマンドのうちのそれぞれの 1 つに関連し、

30

前記第 2 のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 2 のレンダリングパス反復中に実行される少なくとも 2 つのビンごとの描画呼出しの実行の間に発生する時点を示し、前記第 2 のレンダリングパス反復中に実行される前記少なくとも 2 つのビンごとの描画呼出しの各々は、前記少なくとも 2 つの描画呼出しコマンドのうちのそれぞれの 1 つに関連し、前記第 2 のレンダリングパス反復は、前記第 1 のレンダリングパス反復とは異なる、

C 2 1 に記載のデバイス。

40

[ C 2 3 ] 前記複数のビンごとのタイムスタンプ値の各々は、複数のビンごとのタイムスタンプ要求のうちのそれぞれの 1 つに応答して生成され、前記ビンごとのタイムスタンプ要求の各々は、複数のコマンドストリームのうちのそれぞれの 1 つ内に配置され、前記コマンドストリームの各々は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つ中に前記 GPU によって実行され、前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングするように構成される、C 1 6 に記載のデバイス。

[ C 2 4 ] 前記 1 つまたは複数のプロセッサは、前記複数のビンごとのタイムスタンプ値に基づいて複数のタイムスタンプ値を生成するようにさらに構成され、前記タイムスタンプ値の各々は、前記グラフィックスフレームに関して実行されるコマンドの順序付けら

50

れたシーケンス内に含まれる複数のタイムスタンプ要求のうちのそれぞれの 1 つに対応し、前記タイムスタンプ要求の各々は、グラフィックスアプリケーションによって要求される、C 1 6 に記載のデバイス。

[ C 2 5 ] 前記タイムスタンプ要求のうちの少なくとも 2 つは、前記グラフィックスフレームについて実行されるコマンドの前記順序付けられた前記順序付けられたシーケンス内で連続する描画呼出しコマンドのそれぞれの対の間に位置決めされ、

前記 1 つまたは複数のプロセッサは、コマンドの前記順序付けられたシーケンス内の前記タイムスタンプ要求について返される前記タイムスタンプ値が、コマンドの前記順序付けられたシーケンスの始めからコマンドの前記順序付けられたシーケンスの終りまで値において単調に増加するように、前記ビンごとのタイムスタンプ値に基づいて前記複数のタイムスタンプ値を生成するようにさらに構成される、  
10 C 2 4 に記載のデバイス。

[ C 2 6 ] 前記複数のタイムスタンプ値は、前記グラフィックスフレームの前記レンダリング中に実行するのにコマンドの前記順序付けられたシーケンス内の描画呼出しコマンドによって費やされる時間の相対的な長さを示す、C 2 4 に記載のデバイス。

[ C 2 7 ] 前記 1 つまたは複数のプロセッサは、次式に基づいて前記複数のタイムスタンプ値を生成するようにさらに構成され、

【数 7】

$$Value(x) = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(x, y) - TSVGPU(y))$$

20

V a l u e ( x ) は、コマンドの前記順序付けられたシーケンス内の第 x のタイムスタンプ要求に応答して生成されるタイムスタンプのタイムスタンプ値であり、 T S V ( x , y ) は、第 y のレンダリングパス反復中に生成され、コマンドの前記順序付けられたシーケンス内の第 x のタイムスタンプ要求に対応する、ビンごとのタイムスタンプ値であり、 T S V G P U ( y ) は、前記第 y のレンダリングパス反復に関して何らかのプリミティブをレンダリングする前に前記 G P U によって生成される基準タイムスタンプ値であり、 N は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、C 2 4 に記載のデバイス。  
30

[ C 2 8 ] 前記 1 つまたは複数のプロセッサは、

グラフィックスアプリケーションからタイムスタンプ要求を受け取り、

前記タイムスタンプ要求に基づいて複数のビンごとのタイムスタンプ要求を生成し、

前記ビンごとのタイムスタンプ要求の各々を複数のコマンドストリームのうちのそれぞれの 1 つ内に配置し、前記コマンドストリームの各々は、タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つ中に前記 G P U によって実行され、前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングするように構成され、

前記 G P U に前記コマンドストリームを実行させ、

前記コマンドストリーム内に配置された前記ビンごとのタイムスタンプ要求に応答して前記 G P U によって生成された前記ビンごとのタイムスタンプ値に基づいて前記タイムスタンプ値を生成する、

ようさらに構成される、C 1 6 に記載のデバイス。

[ C 2 9 ] 前記 1 つまたは複数のプロセッサは、中央処理装置 ( C P U ) を備える、C 1 6 に記載のデバイス。

[ C 3 0 ] 前記 1 つまたは複数のプロセッサは、前記 G P U を備える、C 1 6 に記載のデバイス。

[ C 3 1 ] 前記デバイスは、ワイヤレス通信デバイスを備える、C 1 6 に記載のデバイス。  
50

[ C 3 2 ] 前記デバイスは、携帯電話送受話器を備える、C 1 6 に記載のデバイス。

[ C 3 3 ] グラフィックス処理ユニット ( G P U ) と、

グラフィックスフレームに関するタイルベースのレンダリングを実行している間に前記 G P U によって生成される複数のピクセルごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成するための手段と、を備え、前記タイムスタンプ値は、前記複数のピクセルごとのタイムスタンプ値のうちの少なくとも 2 つのピクセルごとのタイムスタンプ値の関数である、

装置。

[ C 3 4 ] 前記複数のピクセルごとのタイムスタンプ値の各々は、前記タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つ中に前記 G P U によって生成され、前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングするように構成される、C 3 3 に記載の装置。

10

[ C 3 5 ] 前記少なくとも 2 つのピクセルごとのタイムスタンプ値は、第 1 のピクセルごとのタイムスタンプ値と第 2 のピクセルごとのタイムスタンプ値とを備え、

前記第 1 のピクセルごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 1 のレンダリングパス反復中に前記 G P U によって生成され、

前記第 2 のピクセルごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 2 のレンダリングパス反復中に前記 G P U によって生成され、前記第 2 のレンダリングパス反復は、前記第 1 のレンダリングパス反復とは異なる、

20

C 3 4 に記載の装置。

[ C 3 6 ] 前記タイムスタンプ値を生成するための前記手段は、次式に基づいて前記タイムスタンプ値を生成するための手段を備え、

【数 8】

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y))$$

V a l u e は、前記タイムスタンプ値であり、 T S V ( y ) は、第 y のレンダリングパス反復中に生成されるピクセルごとのタイムスタンプ値であり、 T S V G P U ( y ) は、前記第 y のレンダリングパス反復に関する何らかのプリミティブのレンダリングの前に前記 G P U によって生成される基準タイムスタンプ値であり、 N は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、C 3 4 に記載の装置。

30

[ C 3 7 ] 前記タイムスタンプ値を生成するための前記手段は、中央処理装置 ( C P U ) と前記 G P U とのうちの少なくとも 1 つを備える、C 3 3 に記載の装置。

[ C 3 8 ] 1 つまたは複数のプロセッサによって実行された時に、前記 1 つまたは複数のプロセッサに、

グラフィックスフレームに関するタイルベースのレンダリングを実行している間にグラフィックス処理ユニット ( G P U ) によって生成される複数のピクセルごとのタイムスタンプ値に基づいて時点を示すタイムスタンプ値を生成させる、前記タイムスタンプ値は、前記複数のピクセルごとのタイムスタンプ値のうちの少なくとも 2 つのピクセルごとのタイムスタンプ値の関数である、

40

命令を備える非一時的コンピュータ可読記憶媒体。

[ C 3 9 ] 前記複数のピクセルごとのタイムスタンプ値の各々は、前記タイルベースのレンダリングを実行している間に発生する複数のレンダリングパス反復のうちのそれぞれの 1 つ中に前記 G P U によって生成され、前記レンダリングパス反復の各々は、レンダターゲットの複数の副領域のうちのそれぞれの 1 つをレンダリングするように構成される、C 3 8 に記載の非一時的コンピュータ可読記憶媒体。

50

【C 4 0】前記少なくとも 2 つのビンごとのタイムスタンプ値は、第 1 のビンごとのタイムスタンプ値と第 2 のビンごとのタイムスタンプ値とを備え、

前記第 1 のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 1 のレンダリングパス反復中に前記 G P U によって生成され、

前記第 2 のビンごとのタイムスタンプ値は、前記複数のレンダリングパス反復のうちの第 2 のレンダリングパス反復中に前記 G P U によって生成され、前記第 2 のレンダリングパス反復は、前記第 1 のレンダリングパス反復とは異なる、

C 3 9 に記載の非一時的コンピュータ可読記憶媒体。

【C 4 1】前記 1 つまたは複数のプロセッサに前記タイムスタンプ値を生成させる前記命令は、前記 1 つまたは複数のプロセッサに次式に基づいて前記タイムスタンプ値を生成させる命令を備え、

【数 9】

$$Value = TSVGPU(0) + \sum_{y=0}^{N-1} (TSV(y) - TSVGPU(y))$$

V a l u e は、前記タイムスタンプ値であり、T S V ( y ) は、第 y のレンダリングパス反復中に生成されるビンごとのタイムスタンプ値であり、T S V G P U ( y ) は、前記第 y のレンダリングパス反復に関する何らかのプリミティブのレンダリングの前に前記 G P U によって生成される基準タイムスタンプ値であり、N は、前記グラフィックスフレームをレンダリングするのに使用されるレンダリングパス反復の回数である、C 3 9 に記載の非一時的コンピュータ可読記憶媒体。

【C 4 2】前記 1 つまたは複数のプロセッサは、中央処理装置 ( C P U ) と前記 G P U とのうちの少なくとも 1 つを備える、C 3 8 に記載の非一時的コンピュータ可読記憶媒体。

【図1】

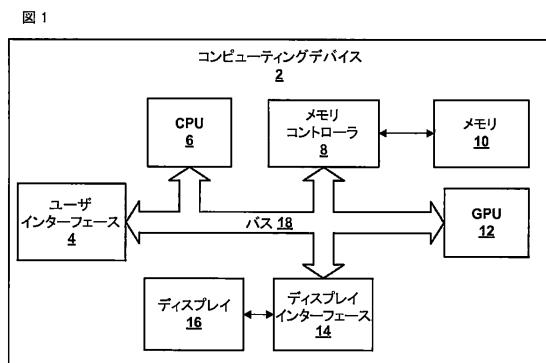


FIG. 1

【図2】

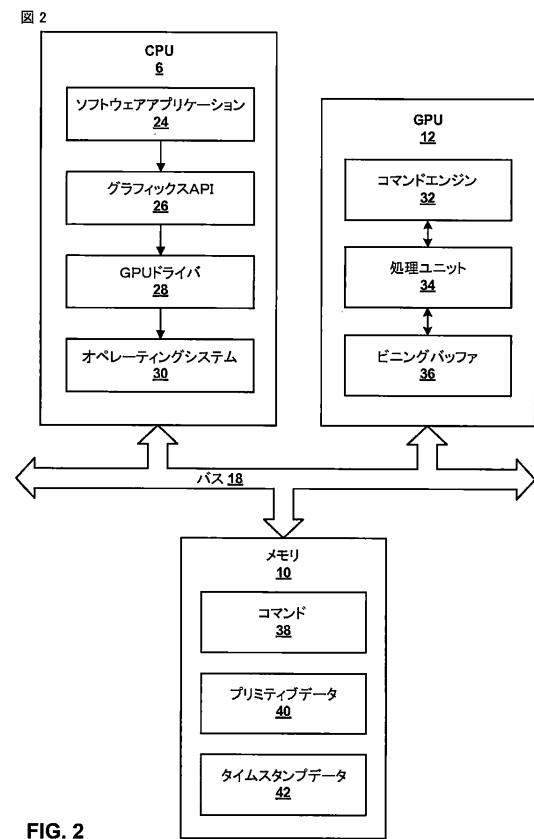


FIG. 2

【図3】

図3

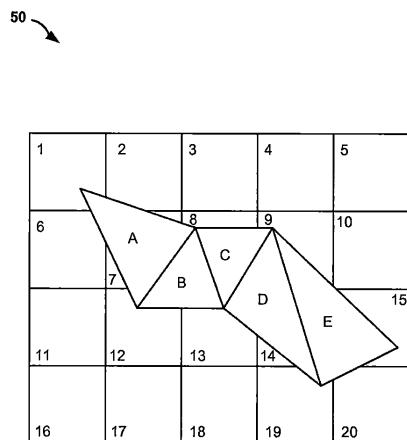


FIG. 3

【図4】

図4



52

【図5】

図5

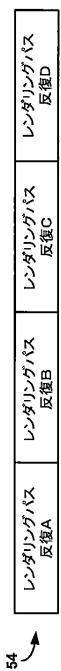


FIG. 5

【図6】

図6

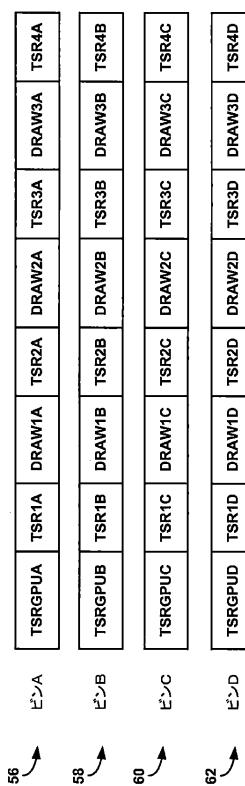


FIG. 6

【図7】

図7

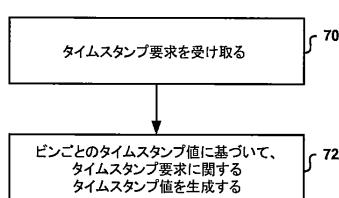


FIG. 7

【図8】

図8

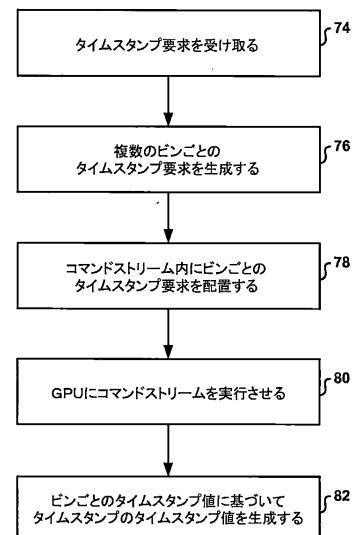


FIG. 8

---

フロントページの続き

(51)Int.Cl.

F I

G 0 9 G	5/36	5 3 0 C
G 0 9 G	5/36	5 3 0 F
G 0 6 F	3/14	A

(72)発明者 フラシャティ、クリストファー・ポール

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 ガンガニ、ヒテンドラ・モハン

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 セーサラマイアー、アビナシュ

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 千葉 久博

(56)参考文献 特開2008-90080(JP, A)

特表2006-511109(JP, A)

米国特許第7385608(US, B1)

(58)調査した分野(Int.Cl., DB名)

G 0 6 T 15 / 0 0 - 15 / 8 7

G 0 6 F 3 / 1 4 - 3 / 1 5 3

G 0 9 G 5 / 0 0 - 5 / 4 2