

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6352546号  
(P6352546)

(45) 発行日 平成30年7月4日(2018.7.4)

(24) 登録日 平成30年6月15日(2018.6.15)

(51) Int.Cl.

G06F 13/28 (2006.01)

F 1

G06F 13/28 310C  
G06F 13/28 310Y

請求項の数 28 (全 42 頁)

(21) 出願番号 特願2017-530746 (P2017-530746)  
 (86) (22) 出願日 平成27年11月6日 (2015.11.6)  
 (65) 公表番号 特表2017-539030 (P2017-539030A)  
 (43) 公表日 平成29年12月28日 (2017.12.28)  
 (86) 國際出願番号 PCT/US2015/059450  
 (87) 國際公開番号 WO2016/093990  
 (87) 國際公開日 平成28年6月16日 (2016.6.16)  
 審査請求日 平成29年12月21日 (2017.12.21)  
 (31) 優先権主張番号 14/566,423  
 (32) 優先日 平成26年12月10日 (2014.12.10)  
 (33) 優先権主張国 米国(US)

早期審査対象出願

(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 92  
 121-1714、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】非整列ブロック転送動作の処理

## (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を用いて、非整列ブロック転送(BLT)コマンドを複数の整列BLTコマンドに変換すること

を備え、前記非整列BLTコマンドが、前記非整列BLTコマンドに関連する複数の面のうちの第1の面のための第1の面幅と、前記第1の面内の第1のBLT領域とを指定し、前記第1の面が、ピクセルデータの2次元アレイに対応し、前記第1の面は、前記第1の面幅を有する複数の線を含み、ここにおいて、前記非整列BLTコマンドを変換することが、

前記第1の面幅に基づいて第2の面幅を決定することと、ここにおいて、前記第2の面幅が前記第1の面幅のN倍であり、ここで、Nは、2以上の整数である、

前記複数の整列BLTコマンドの各々に関連する複数の仮想面のうちの少なくとも1つのための前記第2の面幅を前記複数の整列BLTコマンドの各々が指定するように、前記複数の整列BLTコマンドを生成することと、

を備え、前記複数の整列BLTコマンドが、

前記複数の仮想面のうちのソース仮想面および前記複数の仮想面のうちの宛先仮想面と、前記ソース仮想面および前記宛先仮想面は、それぞれ前記第2の面幅を各々指定し、前記ソース仮想面は、前記第1の面中の前記複数の線のうちのN個の線のセットを組み合わせることによって定義され、

前記ソース仮想面に関連するN個のソース領域および前記宛先仮想面に関連するN

10

20

個の宛先領域と、  
を指定し、

前記N個のソース領域が、前記非整列B L Tコマンドに関連する前記第1のB L T領域を、前記複数の整列B L Tコマンドに関連するN個の不連続の領域に分割することによって作成される、方法。

【請求項2】

前記第2の面幅を決定することが、前記第1の面幅と、前記複数の整列B L Tコマンドを実行するプロセッサに関連する所定の整列幅制約とに基づいて前記第2の面幅を決定することを備える、請求項1に記載の方法。

【請求項3】

前記第1の面幅が、前記所定の整列幅制約の整数倍でない、ここにおいて、前記第1の面幅と前記所定の整列幅制約とに基づいて前記第2の面幅を決定することは、前記第2の面幅が前記所定の整列幅制約の整数倍であるように前記第2の面幅を決定することを備える、請求項2に記載の方法。

【請求項4】

Nが2に等しい、請求項1に記載の方法。

【請求項5】

前記複数の整列B L Tコマンドが、第1の整列B L Tコマンドと第2の整列B L Tコマンドとを備える、ここにおいて、前記複数の整列B L Tコマンドを生成することが、

前記第1のB L T領域に基づいて第2のB L T領域と第3のB L T領域とを決定することと、前記第2のB L T領域が、前記第3のB L T領域とは異なる、

前記第1の整列B L Tコマンドが前記第2のB L T領域を指定するように前記第1の整列B L Tコマンドを生成することと、

前記第2の整列B L Tコマンドが前記第3のB L T領域を指定するように前記第2の整列B L Tコマンドを生成することと

を備える、請求項1に記載の方法。

【請求項6】

前記第2のB L T領域が、前記第1のB L T領域の偶数線を含み、前記第3のB L T領域が、前記第1のB L T領域の奇数線を含む、請求項5に記載の方法。

【請求項7】

前記非整列B L Tコマンドが、前記非整列B L Tコマンドに関連する前記第1の面の第1の面ポインタ値を指定する、ここにおいて、前記複数の整列B L Tコマンドを生成することが、

前記複数の整列B L Tコマンドの各々が、前記複数の整列B L Tコマンドの各々に関連する前記複数の仮想面のうちの1つのための前記第1の面ポインタ値を指定するように前記複数の整列B L Tコマンドを生成すること

を備える、請求項1に記載の方法。

【請求項8】

前記複数の整列B L Tコマンドが、第1の整列B L Tコマンドと第2の整列B L Tコマンドとを備える、ここにおいて、前記複数の整列B L Tコマンドを生成することが、

前記第1のB L T領域に基づいて第2のB L T領域と第3のB L T領域とを決定することと、前記第2のB L T領域が、前記第3のB L T領域とは異なる、

前記第1の整列B L Tコマンドが、前記第2の面幅と、前記第1の面ポインタ値と、前記第2のB L T領域とを指定するように前記第1の整列B L Tコマンドを生成することと、

前記第2の整列B L Tコマンドが、前記第2の面幅と、前記第1の面ポインタ値と、前記第3のB L T領域とを指定するように前記第2の整列B L Tコマンドを生成することとを備える、請求項7に記載の方法。

【請求項9】

前記第2のB L T領域が、前記第1のB L T領域の偶数線を含み、前記第3のB L T領

10

20

30

40

50

域が、前記第1のB L T領域の奇数線を含む、請求項8に記載の方法。

【請求項10】

前記非整列B L Tコマンドが、前記非整列B L Tコマンドに関連する前記第1の面の第1の面ポインタ値を指定する、ここにおいて、前記第1の面ポインタ値が、前記複数の整列B L Tコマンドを実行するプロセッサに関連する所定の整列幅制約の整数倍ではない、ここにおいて、前記複数の整列B L Tコマンドが、第1の整列B L Tコマンドと第2の整列B L Tコマンドとを備える、ここにおいて、前記複数の整列B L Tコマンドを生成することが、

第2の面ポインタ値と第3の面ポインタ値とが前記所定の整列幅制約の異なる整数倍であるように前記第1の面ポインタ値と前記所定の整列幅制約とに基づいて前記第2の面ポインタ値と前記第3の面ポインタ値とを決定することと、

前記第1の整列B L Tコマンドが、前記第1の整列B L Tコマンドに関連する前記複数の仮想面のうちの1つの前記第2の面ポインタ値を指定するように前記第1の整列B L Tコマンドを生成することと、

前記第2の整列B L Tコマンドが、前記第2の整列B L Tコマンドに関連する前記複数の仮想面のうちの1つの前記第3の面ポインタ値を指定するように前記第2の整列B L Tコマンドを生成することと

を備える、請求項1に記載の方法。

【請求項11】

前記第2の面ポインタ値が、前記第1の面ポインタ値よりも小さく、前記第3の面ポインタ値が、前記第1の面ポインタ値よりも大きい、請求項10に記載の方法。

【請求項12】

前記複数の整列B L Tコマンドを生成することが、前記第1のB L T領域に基づいて第2のB L T領域と第3のB L T領域とを決定することを備える、前記第2のB L T領域が、前記第3のB L T領域とは異なる、

ここにおいて、前記第1の整列B L Tコマンドを生成することは、前記第1の整列B L Tコマンドが、前記第2の面幅と、前記第2の面ポインタ値と、前記第2のB L T領域とを指定するように前記第1の整列B L Tコマンドを生成することを備える、

ここにおいて、前記第2の整列B L Tコマンドを生成することは、前記第2の整列B L Tコマンドが、前記第2の面幅と、前記第3の面ポインタ値と、前記第3のB L T領域とを指定するように前記第2の整列B L Tコマンドを生成することを備える、請求項10に記載の方法。

【請求項13】

前記第2のB L T領域が、前記第1のB L T領域の偶数線を含み、前記第3のB L T領域が、前記第1のB L T領域の奇数線を含む、請求項12に記載の方法。

【請求項14】

ロック転送(B L T)コマンドを記憶するように構成されたメモリと、

非整列B L Tコマンドを複数の整列B L Tコマンドに変換することと、ここにおいて、前記非整列B L Tコマンドが、前記非整列B L Tコマンドに関連する複数の面のうちの第1の面のための第1の面幅と、前記第1の面内の第1のB L T領域とを指定し、前記第1の面が、ピクセルデータの2次元アレイに対応し、前記第1の面は、前記第1の面幅を有する複数の線を含み、

前記第1の面幅に基づいて第2の面幅を決定することと、ここにおいて、前記第2の面幅が前記第1の面幅のN倍であり、ここで、Nは、2以上の整数である、

前記複数の整列B L Tコマンドの各々に関連する複数の仮想面のうちの少なくとも1つのための前記第2の面幅を前記複数の整列B L Tコマンドの各々が指定するように、前記複数の整列B L Tコマンドを生成することと

を行うように構成された、1つまたは複数のプロセッサと

を備え、前記複数の整列B L Tコマンドが、

前記複数の仮想面のうちのソース仮想面および前記複数の仮想面のうちの宛先仮想

面と、前記ソース仮想面および前記宛先仮想面は、前記第2の面幅を各々指定し、前記ソース仮想面は、前記第1の面中の前記複数の線のうちのN個の線のセットを組み合わせることによって定義され、

前記ソース仮想面に関連するN個のソース領域および前記宛先仮想面に関連するN個の宛先領域と

を指定し、前記N個のソース領域が、前記非整列BLTコマンドに関連する前記第1のBLT領域を、前記複数の整列BLTコマンドに関連するN個の不連続の領域に分割することによって作成される、デバイス。

【請求項15】

前記1つまたは複数のプロセッサが、前記第1の面幅と、前記複数の整列BLTコマンドを実行するプロセッサに関連する所定の整列幅制約とに基づいて前記第2の面幅を決定することを行うようにさらに構成された、請求項14に記載のデバイス。 10

【請求項16】

前記第1の面幅が、前記所定の整列幅制約の整数倍でない、ここにおいて、前記1つまたは複数のプロセッサは、前記第2の面幅が前記所定の整列幅制約の整数倍であるように前記第2の面幅を決定することを行うようにさらに構成された、請求項15に記載のデバイス。

【請求項17】

Nが2に等しい、請求項14に記載のデバイス。

【請求項18】

前記複数の整列BLTコマンドが、第1の整列BLTコマンドと第2の整列BLTコマンドとを備える、ここにおいて、前記1つまたは複数のプロセッサが、

前記第1のBLT領域に基づいて第2のBLT領域と第3のBLT領域とを決定することと、前記第2のBLT領域が、前記第3のBLT領域とは異なる、

前記第1の整列BLTコマンドが前記第2のBLT領域を指定するように前記第1の整列BLTコマンドを生成することと、

前記第2の整列BLTコマンドが前記第3のBLT領域を指定するように前記第2の整列BLTコマンドを生成することと

を行うようにさらに構成された、請求項14に記載のデバイス。 30

【請求項19】

前記第2のBLT領域が、前記第1のBLT領域の偶数線を含み、前記第3のBLT領域が、前記第1のBLT領域の奇数線を含む、請求項18に記載のデバイス。

【請求項20】

前記非整列BLTコマンドが、前記非整列BLTコマンドに関連する前記第1の面の第1の面ポインタ値を指定する、ここにおいて、前記1つまたは複数のプロセッサが、

前記複数の整列BLTコマンドの各々が、前記複数の整列BLTコマンドの各々に関連する前記複数の仮想面のうちの1つのための前記第1の面ポインタ値を指定するように前記複数の整列BLTコマンドを生成すること

を行うようにさらに構成された、請求項14に記載のデバイス。 40

【請求項21】

前記複数の整列BLTコマンドが、第1の整列BLTコマンドと第2の整列BLTコマンドとを備える、ここにおいて、前記1つまたは複数のプロセッサが、

前記第1のBLT領域に基づいて第2のBLT領域と第3のBLT領域とを決定することと、前記第2のBLT領域が、前記第3のBLT領域とは異なる、

前記第1の整列BLTコマンドが、前記第2の面幅と、前記第1の面ポインタ値と、前記第2のBLT領域とを指定するように前記第1の整列BLTコマンドを生成することと、

前記第2の整列BLTコマンドが、前記第2の面幅と、前記第1の面ポインタ値と、前記第3のBLT領域とを指定するように前記第2の整列BLTコマンドを生成することと

を行うようにさらに構成された、請求項20に記載のデバイス。 50

## 【請求項 2 2】

前記第2のB L T領域が、前記第1のB L T領域の偶数線を含み、前記第3のB L T領域が、前記第1のB L T領域の奇数線を含む、請求項2 1に記載のデバイス。

## 【請求項 2 3】

前記非整列B L Tコマンドが、前記非整列B L Tコマンドに関連する前記第1の面の第1の面ポインタ値を指定する、ここにおいて、前記第1の面ポインタ値が、前記複数の整列B L Tコマンドを実行するプロセッサに関連する所定の整列幅制約の整数倍ではない、ここにおいて、前記複数の整列B L Tコマンドが、第1の整列B L Tコマンドと第2の整列B L Tコマンドとを備える、ここにおいて、前記1つまたは複数のプロセッサが、

第2の面ポインタ値と第3の面ポインタ値とが前記所定の整列幅制約の異なる整数倍であるように前記第1の面ポインタ値と前記所定の整列幅制約とに基づいて前記第2の面ポインタ値と前記第3の面ポインタ値とを決定することと、

前記第1の整列B L Tコマンドが、前記第1の整列B L Tコマンドに関連する前記複数の仮想面のうちの1つの前記第2の面ポインタ値を指定するように前記第1の整列B L Tコマンドを生成することと、

前記第2の整列B L Tコマンドが、前記第2の整列B L Tコマンドに関連する前記複数の仮想面のうちの1つの前記第3の面ポインタ値を指定するように前記第2の整列B L Tコマンドを生成することと

を行うようにさらに構成された、請求項1 4に記載のデバイス。

## 【請求項 2 4】

前記第2の面ポインタ値が、前記第1の面ポインタ値よりも小さく、前記第3の面ポインタ値が、前記第1の面ポインタ値よりも大きい、請求項2 3に記載のデバイス。

## 【請求項 2 5】

前記1つまたは複数のプロセッサが、

前記第1のB L T領域に基づいて第2のB L T領域と第3のB L T領域とを決定することと、前記第2のB L T領域が、前記第3のB L T領域とは異なる、

前記第1の整列B L Tコマンドが、前記第2の面幅と、前記第2の面ポインタ値と、前記第2のB L T領域とを指定するように前記第1の整列B L Tコマンドを生成することと、

前記第2の整列B L Tコマンドが、前記第2の面幅と、前記第3の面ポインタ値と、前記第3のB L T領域とを指定するように前記第2の整列B L Tコマンドを生成することとを行うようにさらに構成された、請求項2 3に記載のデバイス。

## 【請求項 2 6】

前記デバイスが、ワイヤレス通信デバイスとモバイルフォンハンドセットとのうちの少なくとも1つを備える、請求項1 4に記載のデバイス。

## 【請求項 2 7】

非整列ブロック転送(B L T)コマンドを受信するための手段と、

前記非整列B L Tコマンドを複数の整列B L Tコマンドに変換するための手段と、ここにおいて、前記非整列B L Tコマンドが、前記非整列B L Tコマンドに関連する複数の面のうちの第1の面のための第1の面幅と、前記第1の面内の第1のB L T領域とを指定し、前記第1の面が、ピクセルデータの2次元アレイに対応し、前記第1の面は、前記第1の面幅を有する複数の線を含み、

前記第1の面幅に基づいて第2の面幅を決定するための手段と、ここにおいて、前記第2の面幅が前記第1の面幅のN倍であり、ここで、Nは、2以上の整数である、

前記複数の整列B L Tコマンドの各々に関連する複数の仮想面のうちの少なくとも1つのための前記第2の面幅を前記複数の整列B L Tコマンドの各々が指定するように、前記複数の整列B L Tコマンドを生成するための手段と

を備え、前記複数の整列B L Tコマンドが、

前記複数の仮想面のうちのソース仮想面および前記複数の仮想面のうちの宛先仮想面と、前記ソース仮想面および前記宛先仮想面は、前記第2の面幅を各々指定し、前記ソー

10

20

30

40

50

ス仮想面は、前記第1の面中の前記複数の線のうちのN個の線のセットを組み合わせることによって定義され、

前記ソース仮想面に関連するN個のソース領域および前記宛先仮想面に関連するN個の宛先領域と

を指定し、前記N個のソース領域が、前記非整列B L Tコマンドに関連する前記第1のB L T領域を、前記複数の整列B L Tコマンドに関連するN個の不連続の領域に分割することによって作成される、装置。

#### 【請求項28】

実行されたとき、1つまたは複数のプロセッサに、

非整列ブロック転送(B L T)コマンドを複数の整列B L Tコマンドに変換することと、ここにおいて、前記非整列B L Tコマンドが、前記非整列B L Tコマンドに関連する複数の面のうちの第1の面のための第1の面幅と、前記第1の面内の第1のB L T領域とを指定し、前記第1の面が、ピクセルデータの2次元アレイに対応し、前記第1の面は、前記第1の面幅を有する複数の線を含み、

前記第1の面幅に基づいて第2の面幅を決定することと、ここにおいて、前記第2の面幅が前記第1の面幅のN倍であり、ここで、Nは、2以上の整数である、

前記複数の整列B L Tコマンドの各々に関連する複数の仮想面のうちの少なくとも1つのための前記第2の面幅を前記複数の整列B L Tコマンドの各々が指定するように、前記複数の整列B L Tコマンドを生成することと

を行わせる命令を記憶し、前記複数の整列B L Tコマンドが、

前記複数の仮想面のうちのソース仮想面および前記複数の仮想面のうちの宛先仮想面と、前記ソース仮想面および前記宛先仮想面は、前記第2の面幅を各々指定し、前記ソース仮想面は、前記第1の面中の前記複数の線のうちのN個の線のセットを組み合わせることによって定義され、

前記ソース仮想面に関連するN個複数のソース領域および前記宛先仮想面に関連するN個の宛先領域と

を指定し、前記N個のソース領域が、前記非整列B L Tコマンドに関連する前記第1のB L T領域を、前記複数の整列B L Tコマンドに関連するN個の不連続の領域に分割することによって作成される、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】

##### 【0001】

[0001]本開示は、グラフィックス処理に関し、より詳細には、グラフィックスプロセッサを用いてブロック転送(B L T)動作を実行するための技法に関する。

##### 【背景技術】

##### 【0002】

[0002]グラフィックス処理ユニット(GPU)は、しばしば、ソース面(source surface)の領域を宛先面(destination surface)の領域に複製すること、または複数のソース面の対応する領域を宛先面の単一の対応する領域に合成することを行うためにB L T動作を実行する。GPUは、B L T動作の一部としてメモリとの間で面を書き込み、読み取るための特殊なハードウェアを含み得る。面を書き込み、読み取るためのGPUハードウェアは、B L Tコマンドをうまくサービスするために満たされる必要がある様々な整列制約を有し得る。B L Tコマンドが、そのような整列制約を満たさない場合、GPUハードウェアは、B L Tコマンドをサービスすることができないことがある。

##### 【発明の概要】

##### 【0003】

[0003]本開示では、非整列B L Tコマンドを処理するための技法について説明する。非整列B L Tコマンドは、整列制約されたGPUの整列制約を満たさず、したがって、GPUによって処理されることがないことがあるB L Tコマンドを指す(refer to)ことがある。本開示の技法は、非整列B L Tコマンドを複数の整列B L Tコマンドに変換する

10

20

30

40

50

こと、ここで、複数の整列 B L T コマンドは、非整列 B L T コマンドによって生成された (produced) であろうメモリ状態と同じ結果として生じるメモリ状態を集合的に生成し得る、を伴い得る。ただし、非整列 B L T コマンドとは異なり、複数の整列 B L T コマンドは、G P U の整列制約を満たし、したがって、G P U によって処理されることが可能であり得る。

#### 【0004】

[0004] B L T コマンドを処理するために G P U を使用することは、中央処理ユニット (C P U) を使用することと比較して、B L T コマンドの実行を加速し、および / または B L T コマンドの実行に関連する電力消費量を低減し得る。したがって、非整列 B L T コマンドを複数の整列 B L T コマンドに変換することによって、比較的低電力の G P U 加速 B L T 処理の利益が、下にある (underlying) 非整列面を C P U が前処理および / または後処理する必要なしに非整列 B L T コマンドのために達成され得る。このようにして、整列制約された G P U ベースのシステムにおいて非整列 B L T コマンドを処理することに関連するパフォーマンスおよび / または電力消費量が改善され得る。

10

#### 【0005】

[0005] 一例では、本開示では、1つまたは複数のプロセッサを用いて、非整列 B L T コマンドを複数の整列 B L T コマンドに変換することを含む方法について説明する。

#### 【0006】

[0006] 別の例では、本開示では、非整列 B L T コマンドを複数の整列 B L T コマンドに変換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について説明する。

20

#### 【0007】

[0007] 別の例では、本開示では、非整列 B L T コマンドを受信するための手段を含む装置について説明する。本装置は、非整列 B L T コマンドを複数の整列 B L T コマンドに変換するための手段をさらに含む。

#### 【0008】

[0008] 別の例では、本開示では、実行されたとき、1つまたは複数のプロセッサに、非整列 B L T コマンドを複数の整列 B L T コマンドに変換することを行わせる命令を記憶したコンピュータ可読記憶媒体について説明する。

#### 【0009】

30

[0009] 本開示の1つまたは複数の例の詳細を添付の図面および以下の説明に記載する。本開示の他の特徴、目的、および利点は、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】

#### 【0010】

【図1】 [0010] 本開示の B L T 動作処理技法を実施するために使用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】 [0011] 図1のコンピューティングデバイスの部分 (portions) をさらに詳細に示すブロック図。

#### 【図3】 [0012]

40

本開示による、例示的な単一のソース B L T 動作を示す概念図。

#### 【図4】 [0013]

本開示による、例示的な複数のソース B L T 動作を示す概念図。

【図5】 [0014] 図1および図2の例示的なコンピューティングデバイスのための例示的なコマンド処理フローを示す概念図。

【図6】 [0015] 本開示による、面の線を組み合わせることによって整列面を生成するための例示的な技法を示す概念図。

【図7】 [0016] 図6に示した技法を使用した非整列 B L T 動作の2つの整列 B L T 動作への例示的な変換を示す概念図。

【図8】 [0017] 図6に示した技法を使用した非整列 B L T 動作の2つの整列 B L T 動作への例示的な変換を示す概念図。

#### 【図9】 [0017]

50

本開示による、複数の整列仮想面を定義するための例示的な技法を示す概

念図。

【図10】[0018]本開示による、図9において定義されている複数の整列仮想面を使用して非整列BLT動作の2つの整列BLT動作への例示的な変換を示す概念図。

【図11】本開示による、図9において定義されている複数の整列仮想面を使用して非整列BLT動作の2つの整列BLT動作への例示的な変換を示す概念図。

【図12】[0019]本開示による、非整列BLT動作を処理するための例示的な技法を示す流れ図。

【発明を実施するための形態】

【0011】

[0020]本開示では、非整列ブロック転送 (unaligned block transfer) (BLT) コマンドを処理するための技法について説明する。非整列BLTコマンドは、整列制約されたグラフィックス処理ユニット (GPU) の整列制約を満たさず、したがって、GPUによって処理されることができないことがあるBLTコマンドを指すことがある。本開示の技法は、非整列BLTコマンドを複数の整列BLTコマンドに変換すること、ここで、複数の整列BLTコマンドは、非整列BLTコマンドによって生成されたであろうメモリ状態と同じ結果として生じるメモリ状態を集合的に生成し得る、を伴い得る。ただし、非整列BLTコマンドとは異なり、複数の整列BLTコマンドは、GPUの整列制約を満たし、したがって、GPUによって処理され得ることが可能であり得る。

10

【0012】

[0021]BLTコマンドを処理するためにGPUを使用することは、中央処理ユニット (CPU) を使用することと比較して、BLTコマンドの実行を加速し、および/またはBLTコマンドの実行に関連する電力消費量を低減し得る。したがって、非整列BLTコマンドを複数の整列BLTコマンドに変換することによって、比較的低電力のGPU加速BLT処理の利益が、下にある非整列面をCPUが前処理および/または後処理する必要なしに非整列BLTコマンドのために達成され得る。このようにして、整列制約されたGPUベースのシステムにおいて非整列BLTコマンドを処理することに関連するパフォーマンスおよび/または電力消費量が改善され得る。

20

【0013】

[0022]BLTコマンドは、GPUにBLT動作を実行するように命令するコマンドを指すことがある。BLT動作は、ソース面の領域を宛先面の領域に複製すること、または複数のソース面の対応する領域を宛先面の単一の対応する領域に合成することを伴う動作を指すことがある。ソース面の領域は、ソース領域と呼ばれることがあり、宛先面の領域は、宛先領域と呼ばれることがある。ソース領域および宛先領域は、それらの領域内のピクセルに関して複製および合成することが行われることを示すために、代替的に、BLT領域および/または複製領域と呼ばれることがある。BLT領域の各々は、それぞれのBLT領域を含む面の連続するサブセットであり得る。いくつかの例では、BLT領域は、長方形の領域であり得る。

30

【0014】

[0023]GPUは、BLT動作の一部としてメモリとの間で面を書き込み読み取る (writing and reading surfaces) ための特殊なハードウェアを含み得る。ハードウェアは、面の各線の開始が所定の整列幅の整数倍であるメモリアドレスに対応することを必要とする整列制約を有し得る。整列制約を満たす面は、整列面であると言われ、一方、整列制約を満たさない面は、非整列面であると言われる。

40

【0015】

[0024]しばしば、コンパイラまたはアプリケーションは、BLT動作を実行しているであろう特定のGPUハードウェアの整列制約へのアクセスを有し得、したがって、結果として生じる面が整列されるように整列制約に基づいてBLT命令を生成し得る。ただし、すべてのシステムが、GPUの整列制約をコンパイラまたはアプリケーションに利用可能にするとは限らない。または、利用可能な場合であっても、いくつかのコンパイラまたはアプリケーションは、様々な理由で（たとえば、コンパイラまたはアプリケーションの複

50

雑さを低減するために) そのような整列制約を満たすために、生成されたコードを最適化しないことを決定し得る。

【0016】

[0025] BLT 動作のための 1 つまたは複数の面が GPU ハードウェアの整列制約を満たさない場合、GPU ドライバは、BLT 動作のパフォーマンスを加速するために GPU を直接使用することができないことがある。代わりに、GPU ドライバは、中央処理ユニット (CPU) を用いて BLT 動作を実行する必要があり得る。他の場合には、GPU ドライバは、非整列面を整列面に複製するために CPU を使用し、整列面に対して BLT 動作を実行するために GPU を使用し、非整列面に BLT 動作の結果を複製し戻すために CPU を使用し得る。いずれの場合も、BLT 動作自体を実行するかまたは非整列面と整列面との間で変換するために CPU を使用することは、BLT 動作の電力消費量を増加させ、BLT 動作のパフォーマンスを低減させる。

【0017】

[0026] 整列 BLT コマンドは、BLT コマンドに関連する面のすべてが整列面である BLT コマンドを指すことがある。非整列 BLT コマンドは、面のうちの少なくとも 1 つが非整列面である BLT コマンドを指すことがある。同様に、整列 BLT 動作は、BLT 動作に関連する面のすべてが整列面である BLT 動作を指すことがあり、非整列 BLT 動作は、面のうちの少なくとも 1 つが非整列面である BLT コマンドを指すことがある。

【0018】

[0027] 概念的に、整列面は、面の各線の開始が所定の整列幅の整数倍であるメモリアドレスに対応する面を指すことがある。所定の整列幅は、いくつかの例では、BLT 動作の適切な実行のために GPU メモリアクセスハードウェアによって必要とされる整列幅に対応し得る。整列境界は、所定の整列幅の整数倍であるメモリアドレスを指すことがある。

【0019】

[0028] 面は、面ポインタと、面幅パラメータと、面高さパラメータとによって定義され得る。面ポインタは、面の開始に対応するメモリアドレスを示し(または指し)得る。面幅パラメータ(またはストライドパラメータ)は、面中の各線の幅を示し得る。面高さパラメータは、面中に何本の線が含まれるかを示し得る。

【0020】

[0029] 整列面は、(1) 面ポインタが整列境界を指し、(2) 面幅が整列幅の整数倍である面を指すことがある。これらの条件のいずれかが満たされない場合、面は非整列であると言われる。

【0021】

[0030] 本開示の第 1 の態様によれば、条件(1)が満たされ、条件(2)が満たされない(すなわち、BLT コマンドの下にある面のための面幅が整列幅の整数倍でない)場合、グラフィックスドライバは、BLT コマンドに関連するソース面と宛先面との各々について、整列幅の整数倍である面幅を有する整列仮想面を形成するために、それぞれの面の複数の線を单一の線に組み合わせ得る。仮想面のための面幅が整列幅の整数倍であるので、仮想面は、整列制約された GPU によって処理され得る整列面である。

【0022】

[0031] しかしながら、仮想面を形成するために線を組み合わせることにより、BLT 領域(すなわち、コピーおよび/または合成されるべき領域)をもはや連続する領域でない(たとえば、もはや单一の連続する矩形でない)ように変換するようになり得る。たとえば、着信(incoming) BLT コマンドに関連するソース BLT 領域が、单一の連続する矩形である場合、ソース面を仮想ソース面に変換することは、ソース BLT 領域を、互いに連続しないことがある 2 つの別個の矩形に変換し得る。一般に、单一の BLT 動作は、不連続の BLT 領域に対して実行されないことがある。

【0023】

[0032] この問題に対処するために、グラフィックスドライバは、変換された BLT 領域の連続するサブセットを決定し、複数の BLT コマンドを生成し得、ここで、BLT コマ

10

20

30

40

50

ンドの各々は、G P Uに、変換されたB L T領域の連続するサブセットの各1つに対してB L T動作を実行させる。たとえば、長方形ソースB L T領域が2つの長方形ソースB L T領域に変換される場合、グラフィックスドライバは、第1に長方形ソースB L T領域のための第1のB L Tコマンドと第2の長方形ソースB L T領域のための第2のB L Tコマンドとを生成し得る。

#### 【0024】

[0033]複数のB L Tコマンドのためのソース面と宛先面との各々は、1つまたは複数の整列仮想面に対応し、それによって、複数のB L Tコマンドを整列B L Tコマンドにさせ得る。複数の整列B L Tコマンドは、ソースおよび宛先B L T領域全体を集合的に処理し、それによって、単一の非整列B L Tコマンドによって生成されたであろうメモリ状態と同じ結果として生じるメモリ状態を生成し得る。このようにして、グラフィックスドライバは、非整列B L Tコマンドを整列制約されたG P Uによってサービスされ得る複数の整列B L Tコマンドに変換し得る。

10

#### 【0025】

[0034]本開示の第2の態様によれば、条件(1)および(2)の両方が満たされない(すなわち、面ポインタが整列境界を指さず、B L Tコマンドの下にある面のための面幅が整列幅の整数倍でない)場合、グラフィックスドライバは、本開示の第1の態様に関して上記で説明したのと同様の方法で非整列面の幅を広げ得る。これにより、広げられた面幅が、整列幅に対応するようになる。ただし、広げられた面の面ポインタが整列境界において開始していないので、広げられた面は依然として整合されない。

20

#### 【0026】

[0035]この問題に対処するための1つの手法は、シフトされた面ポインタが最も近い整列境界を指すように面ポインタをシフトすることである。本開示の第1の態様に関して上記で説明したことと同様に、面を広げることは、B L T領域を複数の連続するサブセットに変換し得、グラフィックスドライバは、複数のB L Tコマンドを生成し得、ここで、B L Tコマンドの各々は、G P Uに、変換されたB L T領域の連続するサブセットの各1つに対してB L T動作を実行させる。ただし(However)、仮想面の幅が同じままである場合、面ポインタをシフトすることにより、面の境界の近くにある変換されたB L T領域サブセットが仮想面の端にラップアラウンドされるようになり得る。言い換えれば、そのような変換されたB L T領域サブセットは、もはや連続でないことがあり、単一のB L T動作によってサービスされることができないことがある。

30

#### 【0027】

[0036]この問題に対処するために、グラフィックスドライバは、ソース面と宛先面との各々のための2つの仮想面を生成し得る。2つの仮想面は、同じ広げられた面幅を有するが、別個の(separate)整列面ポインタを有し得る。仮想面の各々のための面ポインタは、非整列面ポインタのいずれかの側にある整列境界に対応し得る。たとえば、G P Uドライバは、非整列面ポインタよりも小さい整列境界を指す第1の仮想面のための面ポインタを生成し、非整列面ポインタよりも大きい整列境界を指す第2の仮想面のための面ポインタを生成し得る。

#### 【0028】

40

[0037]変換されたB L T領域サブセットは、仮想面のうちの1つの端をラップアラウンドする場合、他の仮想面の端をラップアラウンドしないことになるだろう。したがって、変換されたB L T領域サブセットの各々のためのB L Tコマンドを生成するとき、グラフィックスドライバは、各B L Tコマンドのための仮想面を選択し得、ここで、対応する変換されたB L T領域サブセットは、仮想面の端をラップアラウンドしない。言い換えれば、グラフィックスドライバは、変換されたB L T領域サブセットの各々を連続する領域(たとえば、連続する長方形領域)として表わされることを可能にする仮想面を選択し得る。

#### 【0029】

[0038]複数のB L Tコマンドのためのソース面と宛先面との各々は、複数の整列仮想面

50

のうちの1つに対応し、それによって、複数のB L Tコマンドを整列B L Tコマンドにさせ得る。複数の整列B L Tコマンドは、ソースおよび宛先B L T領域全体を集合的に処理し、それによって、単一の非整列B L Tコマンドによって生成されたであろうメモリ状態と同じ結果として生じるメモリ状態を生成し得る。

#### 【0030】

[0039]非整列B L Tコマンドとは異なり、複数の整列B L Tコマンドは、整列制約されたグラフィックス処理ユニット(G P U)によって処理されることが可能であり得、これは、中央処理ユニット(C P U)と比較して、B L Tコマンドの実行を加速すること、および/またはB L Tコマンドの実行に関連する電力消費量を低減することが可能であり得る。非整列B L Tコマンドを複数の整列B L Tコマンドに変換することによって、比較的低電力のG P U加速B L T処理の利益が、下にある非整列面をC P Uが前処理および/または後処理する必要なしに非整列B L Tコマンドのために達成され得る。このようにして、非整列B L Tコマンドを処理することに関連するパフォーマンスおよび/または電力消費量が改善され得る。

10

#### 【0031】

[0040]図1は、本開示のB L T動作処理技法を実施するために使用され得る例示的な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(P D A)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得る。

20

#### 【0032】

[0041]図1の例に示すように、コンピューティングデバイス2は、ユーザ入力インターフェース4と、C P U6と、メモリサブシステム8と、G P U10と、ディスプレイインターフェース12と、ディスプレイ14と、バス16とを含む。ユーザ入力インターフェース4、C P U6、メモリサブシステム8、G P U10およびディスプレイインターフェース12は、バス16を使用して相互に通信し得る。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ステムの他の構成が使用され得ることに留意されたい。

30

#### 【0033】

[0042]ユーザ入力インターフェース4により、1つまたは複数のユーザ入力デバイス(図示せず)をコンピューティングデバイス2に通信可能に結合することが可能になり得る。ユーザ入力デバイスにより、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に入力を与えることが可能になり得る。例示的なユーザ入力デバイスは、キーボード、マウス、トラックボール、マイクロフォン、タッチパッド、タッチセンシティブまたはプレゼンスセンシティブディスプレイ、あるいは別の入力デバイスを含む。タッチセンシティブまたはプレゼンスセンシティブディスプレイがユーザ入力デバイスとして使用される例では、ユーザ入力インターフェース4の全部または一部は、ディスプレイ14に統合され得る。

40

#### 【0034】

[0043]C P U6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特殊用途プロセッサを備え得る。C P U6は、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し得る。ソフトウェアアプリケーションは、たとえば、ビデオゲームアプリケーシ

50

ヨン、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、グラフィカルユーチューバー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、あるいは任意の他のタイプのソフトウェアアプリケーションまたはプログラムを含み得る。

#### 【0035】

[0044] C P U 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 14 へのグラフィックスデータのレンダリングを行わせるように G P U 1 0 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、Open Graphics Library (OpenGL (商標登録)) 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 System (OpenGL ES) API 、Direct3D API 、DirectX API 、RenderMan API 、WebGL API 、または任意の他の公開の、もしくはプロプライエタリの標準的なグラフィックス API などのグラフィックス 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、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全部を G P U 1 0 に実行させるために、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 0 に発行し得る。いくつかの例では、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、たとえば、点、線、三角形、四角形、トライアングルストリップ、パッチなど、グラフィックスプリミティブのリストを含み得る。

10

#### 【0036】

[0045] メモリサブシステム 8 は、コンピューティングデバイス 2 の中の他の構成要素から受信されたメモリアクセス要求にサービスするように構成され得る。たとえば、メモリサブシステム 8 は、C P U 6 、G P U 1 0 、および / またはディスプレイインターフェース 1 2 からのメモリアクセス要求にサービスし得る。メモリアクセス要求は、読み取りアクセス要求と書き込みアクセス要求とを含み得る。メモリサブシステム 8 は、バス 1 6 を介して C P U 6 、G P U 1 0 、ユーザ入力インターフェース 4 、およびディスプレイインターフェース 1 2 に通信可能に結合される。

20

#### 【0037】

[0046] メモリサブシステム 8 は、メモリコントローラ 2 0 とメモリ 2 2 とを含む。メモリコントローラ 2 0 は、メモリ 2 2 に入りメモリ 2 2 から出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 2 0 は、メモリ読み取り要求とメモリ書き込み要求とを C P U 6 および / または G P U 1 0 から受信し、コンピューティングデバイス 2 の中の構成要素にメモリサービスを提供するために、メモリ 2 2 に関するそのような要求にサービスし得る。

30

#### 【0038】

[0047] メモリコントローラ 2 0 は、専用のメモリバスを介してメモリ 2 2 に、および、バス 1 6 を介して他の構成要素 (たとえば、C P U 6 および G P U 1 0 ) に通信可能に結合される。メモリコントローラ 2 0 は、図 1 の例示的なコンピューティングデバイス 2 では、C P U 6 とメモリ 2 2 の両方とは別の処理モジュールとして示されているが、他の例では、メモリコントローラ 2 0 の機能の一部または全部は、C P U 6 、G P U 1 0 、およびメモリ 2 2 の 1 つもしくは複数で実装され得る。

40

#### 【0039】

[0048] メモリ 2 2 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が使用するためのデータを記憶し得る。たとえば、メモリ 2 2 は、ユーザアプリケーションと、それらのアプリケーションに関連するグラフィックスデータとを記憶し得る。メモリ 2 2 は、コンピューティングデバイス 2 の他の構成要素が使用するための情報、および / またはそれらによって生成された情報を記憶し得る。たとえば、メモリ 2 2 は、G P U 1 0 のデバイスメモリとして働き得、G P U 1 0 によって演算されることになるデータ、ならびに G

50

P U 1 0 によって実行される演算の結果生じるデータを記憶し得る。たとえば、メモリ 2 2 は、ソース面、宛先面、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、フレームバッファなどの、面の任意の組合せを記憶し得る。さらに、メモリ 2 2 は、G P U 1 0 によって処理するためのコマンドストリームを記憶し得る。メモリ 2 2 は、たとえば、ランダムアクセスメモリ ( R A M ) 、スタティック R A M ( S R A M ) 、ダイナミック R A M ( D R A M ) 、同期式ダイナミックランダムアクセスメモリ ( S D R A M ) 、読み取り専用メモリ ( R O M ) 、消去可能プログラマブル R O M ( E P R O M ) 、電気的消去可能プログラマブル R O M ( E E P R O M ( 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の 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み得る。いくつかの例では、メモリ 2 2 は、たとえば、D D R 1 S D R A M 、D D R 2 S D R A M 、D D R 3 S D R A M 、D D R 4 S D R A M などのダブルデータレート ( D D R ) S D R A M であり得る。

#### 【 0 0 4 0 】

[0049] G P U 1 0 は、C P U 6 によってG P U 1 0 に出されたコマンドを実行するように構成され得る。G P U 1 0 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、G P U 状態プログラミングコマンド、B L T 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み得る。B L T コマンドは、たとえば、メモリ複製コマンド、メモリ合成コマンド、およびプリットコマンド ( すなわち、ロック転送コマンド ) を含み得る。

#### 【 0 0 4 1 】

[0050] いくつかの例では、G P U 1 0 は、C P U 6 から受信されたグラフィックスデータをレンダリングし、表示するように構成され得る。そのような例では、C P U 6 は、ディスプレイ 1 4 にレンダリングするためのグラフィックスデータをG P U 1 0 に提供し、G P U 1 0 に 1 つまたは複数のグラフィックスコマンドを発行し得る。グラフィックスコマンドとしては、たとえば、描画呼出しコマンド、G P U 状態プログラミングコマンド、B L T コマンド、プリットコマンドなどがあり得る。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、C P U 6 は、コマンドとグラフィックスデータとをG P U 1 0 によってアクセスされ得るメモリ 2 2 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G P U 1 0 に提供し得る。

#### 【 0 0 4 2 】

[0051] さらなる例では、G P U 1 0 は、C P U 6 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に構成され得る。そのような例では、C P U 6 は、G P U 1 0 に汎用コンピューティングデータを提供し、G P U 1 0 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発行し得る。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、B L T コマンドなどを含み得る。いくつかの例では、C P U 6 は、コマンドとグラフィックスデータとをG P U 1 0 によってアクセスされ得るメモリ 2 2 に書き込むことによって、コマンドと汎用コンピューティングデータとをG P U 1 0 に提供し得る。

#### 【 0 0 4 3 】

[0052] G P U 1 0 は、いくつかの例では、C P U 6 よりも効率的なベクトル演算の処理を行う高度に並列な構造を伴って構築され得る。たとえば、G P U 1 0 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の方式で演算するように構成された複数の処理要素を含み得る。G P U 1 0 の高度並列特性により、いくつかの例では、G P U 1 0 が、C P U 6 を使用して画像をレンダリングするよりも迅速にグラフィックス画像 ( たとえば、G U I ならびに 2 次元 ( 2 D ) および / または 3 次元 ( 3 D ) のグラフィックスシーン ) をディスプレイ 1 4 上にレンダリングすることが可能になり得る。加えて、G P U 1 0 の高度に並列な性質により、G P U 1 0 が、C P U 6 よりも迅速に、汎用コンピューティングアプリケーションのためのいくつかのタイプのベクトル演算と行列演算とを処理することが可能になり得る。

#### 【 0 0 4 4 】

10

20

30

40

50

[0053]いくつかの例では、G P U 1 0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。他の事例では、G P U 1 0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、または場合によつては、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。さらなる例では、G P U 1 0は、システムオンチップ（S o C）を形成するC P U 6と同じマイクロチップ上に配置され得る。G P U 1 0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（A S I C）、フィールドプログラマブルゲートアレイ（F P G A）、デジタル信号プロセッサ（D S P）、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などの、1つまたは複数のプロセッサを含み得る。

10

#### 【 0 0 4 5 】

[0054]いくつかの例では、G P U 1 0は、メモリ2 2のすべてまたは一部分にキャッシングサービスを提供し得るG P U キャッシュを含み得る。そのような例では、G P U 1 0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し得る。これにより、G P U 1 0が、大量のバストラフィックを受けることがある、バス1 6を介してG P U 1 0がメモリ2 2にアクセスする必要を減らすことによって、より効率的な方法で動作することが可能になり得る。しかしながら、いくつかの例では、G P U 1 0は、別個のキャッシングを含まず、代わりに、バス1 6を介してメモリ2 2を利用し得る。G P U は、たとえば、ランダムアクセスメモリ（R A M）、スタティックR A M（S R A M）ダイナミックR A M（D R A M）、消去可能プログラマブルR O M（E P R O M）、電気的消去可能プログラマブルR O M（E E P R O M）、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ストレージデバイスを含み得る。

20

#### 【 0 0 4 6 】

[0055]C P U 6および/またはG P U 1 0は、レンダリングされた画像データをメモリ2 2内に割り振られたフレームバッファ内に記憶し得る。ディスプレイインターフェース1 2は、データをフレームバッファから取り出し、レンダリングされた画像データによって表される画像を表示するようにディスプレイ1 4を構成し得る。いくつかの例では、ディスプレイインターフェース1 2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1 4によって消費可能なアナログ信号に変換するように構成されたデジタルアナログコンバータ（D A C）を含み得る。他の例では、ディスプレイインターフェース1 2は、処理のために、デジタル値をディスプレイ1 4に直接渡し得る。

30

#### 【 0 0 4 7 】

[0056]ディスプレイ1 4は、モニタ、テレビジョン、投影デバイス、液晶ディスプレイ（L C D）、プラズマディスプレイパネル、発光ダイオード（L E D）アレイ、陰極線管（C R T）ディスプレイ、電子ペーパー、表面伝導電子放出ディスプレイ（S E D：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み得る。ディスプレイ1 4は、コンピューティングデバイス2内に統合され得る。たとえば、ディスプレイ1 4は、携帯電話ハンドセットまたはタブレットコンピュータのスクリーンであり得る。代替的に、ディスプレイ1 4は、有線またはワイヤレス通信リンクを介してコンピューティングデバイス2に結合されたスタンドアロンデバイスであり得る。たとえば、ディスプレイ1 4は、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であり得る。

40

#### 【 0 0 4 8 】

[0057]バス1 6は、第1世代、第2世代、および第3世代のバス構造ならびにバスプロトコルと、共有されたバス構造およびバスプロトコルと、ポイントツーポイントバス構造およびバスプロトコルと、一方向バス構造およびバスプロトコルと、双方向バス構造およびバスプロトコルとを含む、バス構造およびバスプロトコルの任意の組合せを使用して実装され得る。バス1 6を実装するために使用され得る様々なバス構造およびバスプロトコ

50

ルの例としては、たとえば、HyperTransportバス、InfiniBand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)バスがある。他のタイプのバス構造およびプロトコルも使用され得る。

#### 【0049】

[0058] GPU10は、BLTコマンドを受信し、対応するBLT動作を実行することによってBLTコマンドをサービスし得る。場合によっては、GPU10は、BLT動作を実行するように構成されたメモリアクセスハードウェアを含み得、このハードウェアは、BLT動作をうまく(successfully)サービスするために満たされる必要がある様々な整列制約を有し得る。たとえば、そのようなメモリアクセスハードウェアは、BLT動作のための面の一部または全部が整列されていることを必要とするように構成され得る。

#### 【0050】

[0059]整合されるべき面について、メモリアクセスハードウェアは、(1)面のための面ポインタが整列境界を指すこと、および(2)面のための面幅が整列幅の整数倍であることを必要とし得る。これらの条件のいずれかが満たされない場合、面は非整列であると言われる。

20

#### 【0051】

[0060]場合によっては、CPU6上で実行されているソフトウェアアプリケーションは、非整列BLTコマンドを発行し得る。非整列BLTコマンドは、ソフトウェアアプリケーションをコンパイルしたコンパイラまたは書き込まれたソフトウェアアプリケーションが、コンパイルおよび/またはランタイム時にコンパイラまたはアプリケーションにとって利用可能な整列制約を有しなかったか、あるいは(たとえば、コンパイラまたはアプリケーションの複雑さを低減するために)整列制約を満たすために生成されたコードを最適化することを決定しなかったときに生じ得る。BLTコマンドが非整列であるので、GPU10は、そのようなコマンドを直接サービスすることができないことがある。

#### 【0052】

30

[0061]本開示の例によれば、CPU6および/またはGPU10は、非整列BLTコマンドを複数の整列BLTコマンドに変換すること、ここで、複数の整列BLTコマンドは、非整列BLTコマンドによって生成されたであろうメモリ状態と同じ結果として生じるメモリ状態を集合的に生成し得る、を行うように構成され得る。複数の整列BLTコマンドは、GPU10の整列制約を満たし、したがって、GPU10によって処理されることが可能であり得る。このようにして、比較的低電力のGPU加速BLT処理の利益が非整列BLTコマンドについて取得され、それによって、非整列BLTコマンドを処理することに関連するパフォーマンスおよび/または電力消費量を改善し得る。

#### 【0053】

40

[0062]図2は、図1中のコンピューティングデバイス2の部分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GPU10とメモリサブシステム8とに通信可能に結合され、GPU10はCPU6とメモリサブシステム8とに通信可能に結合される。いくつかの例では、GPU10は、CPU6とともにマザーボード上に統合され得る。追加の例では、GPU10は、CPU6を含むマザーボードのポート中に設置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、GPU10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得る。追加の例では、GPU10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。

#### 【0054】

[0063]図2の例示的なシステムでは、CPU6は、ソフトウェアアプリケーション24

50

と、グラフィックス API 2 6 と、GPU ドライバ 2 8 と、オペレーティングシステム 3 0 とを実行する。ソフトウェアアプリケーション 2 4 は、グラフィックス画像を表示させる 1 つもしくは複数の命令および / または非グラフィックスタスク (たとえば、汎用コンピューティングタスク) を GPU 1 0 上で実行させる 1 つもしくは複数の命令を含み得る。いくつかの例では、ソフトウェアアプリケーション 2 4 は、1 つまたは複数の BLT 命令を含み得、これは、代替的に、BLT コマンド、ソフトウェア BLT コマンド、またはアプリケーション BLT コマンドと呼ばれることがある。ソフトウェアアプリケーション 2 4 は、グラフィックス API 2 6 に命令を発行し得る。グラフィックス API 2 6 は、ソフトウェアアプリケーション 2 4 から受信された命令を GPU ドライバ 2 8 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であり得る。

10

#### 【0055】

[0064] GPU ドライバ 2 8 は、グラフィックス API 2 6 を介してソフトウェアアプリケーション 2 4 から命令を受信し、命令をサービスするために GPU 1 0 の動作を制御する。たとえば、GPU ドライバ 2 8 は、1 つまたは複数のコマンドを編成し、コマンドをメモリ 2 2 の中に配置し、コマンドを実行するように GPU 1 0 に命令し得る。いくつかの例では、GPU ドライバ 2 8 は、コマンドをメモリ 2 2 の中に配置し、たとえば、1 つまたは複数のシステム呼出しを介してオペレーティングシステム 3 0 を介して GPU 1 0 と通信し得る。

#### 【0056】

[0065] いくつかの例では、GPU ドライバ 2 8 (または CPU 6 上の別のソフトウェア構成要素) は、本開示で説明する技法のいずれかに従って、ソフトウェアアプリケーション 2 4 から非整列 BLT コマンドを受信し、非整列 BLT コマンドを複数の整列 BLT コマンドに変換するように構成され得る。GPU ドライバ 2 8 は、実行のために GPU 1 0 に複数の整列 BLT コマンドを与え得る。

20

#### 【0057】

[0066] GPU 1 0 は、コマンドエンジン 3 2 と、ラスタ演算ユニット 3 4 と、バッファ 3 6 と、読み取りエンジン 3 8 と、書き込みエンジン 4 0 とを含む。いくつかの例では、GPU 1 0 は、3 次元 (3D) グラフィックスレンダリングパイプライン (図示せず) を含み得る。そのような例では、ラスタ演算ユニット 3 4 は、3D グラフィックスレンダリングパイプラインの一部を形成し得る。

30

#### 【0058】

[0067] いくつかの例では、GPU 1 0 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複数の処理ユニット (図示せず) を含み得る。GPU 1 0 が 3D グラフィックスレンダリングパイプラインを含む場合、複数の処理ユニットが、3D グラフィックスレンダリングパイプラインを実装するために一緒に動作し得る。いくつかの例では、3D グラフィックスレンダリングパイプラインは、Open Graphics Library (OpenGL (登録商標)) API、Open Graphics Library Embedded Systems (OpenGL ES) API、Direct3D API、DirectX API など、1 つまたは複数の公開のおよび / またはプロプライエタリの 3D グラフィックス規格の API によって指定される、3D グラフィックスパイプラインに準拠し得る。

40

#### 【0059】

[0068] プログラマブル処理ユニットは、たとえば、CPU 6 によって GPU 1 0 上にコードされた 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。いくつかの例では、プログラマブルシェーダユニットは、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、SIMD パイ

50

ブレインを含み得る。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、実行されているプログラムメモリ中の現在の命令またはフェッチされることになる次の命令を示すプログラムカウンタレジスタとを有し得る。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、計算シェーダユニットなどを含み得る。

#### 【0060】

[0069] 固定機能処理ユニットは、いくつか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は、1つまたは複数の制御信号を介して、たとえば、異なる機能を実行するように構成され得るが、固定機能ハードウェアは、通常、ユーザコンパイルプログラムを受信することが可能なプログラムメモリを含まない。GPU 10中に含まれる固定機能処理ユニットは、たとえば、ラスタ演算、深度テスト、シザーテスト、アルファブレンドなどを実行する処理ユニットを含み得る。

10

#### 【0061】

[0070] ラスタ演算ユニット34が3Dレンダリングパイプライン中に含まれる例では、ラスタ演算ユニット34は、いくつかの例では、構成可能な固定機能処理ユニットであり得る。いくつかの例では、ラスタ演算ユニット34は、3Dレンダリングパイプラインの出力統合処理ユニットとして実装され得る。

#### 【0062】

[0071] コマンドエンジン32は、CPU 6からコマンドを受信し、GPU 10にコマンドを実行せしように構成される。いくつかの例では、CPU 6は、コマンドをメモリ22に配置し得、コマンドエンジン32は、メモリ22からコマンドを取り出し得る。コマンドは、たとえば、GPU 10の状態（たとえば、レンダリング状態）を変更する状態コマンド、特定のシェーダプログラムを3Dレンダリングパイプライン中のプログラム可能処理ユニットの1つまたは複数へロードするようにGPU 10に命令するシェーダプログラム拘束コマンド、メモリ22に記憶されている特定のデータをレンダリングするようにGPU 10に命令する描画コマンド、メモリ22に記憶されている特定のデータに関して1つまたは複数の汎用コンピューティングカーネルを実行するようにGPU 10に命令する汎用コンピューティングコマンド、および／あるいは1つまたは複数のBLT動作を実行するようにGPU 10に命令するBLTコマンドを含み得る。

20

#### 【0063】

[0072] 状態コマンドを受信したことに応答して、コマンドエンジン32は、状態コマンドに基づいて、GPU 10の中の1つもしくは複数の状態レジスタを特定の値に設定するよう、および／または、状態コマンドに基づいて、GPU 10のための3Dレンダリングパイプライン（図示せず）中の固定機能処理ユニットの1つもしくは複数を構成するよう構成され得る。描画呼出しコマンドを受信したことに応答して、コマンドエンジン32は、GPU 10中の3Dレンダリングパイプライン（図示せず）に、メモリ22から3Dジオメトリを取り出させ、3Dジオメトリデータを2Dグラフィックス画像へレンダリングせしように構成され得る。シェーダプログラム拘束コマンドを受信したことに応答して、コマンドエンジン32は、3Dレンダリングパイプライン中のプログラム可能処理ユニットの1つまたは複数に、1つまたは複数の特定のシェーダプログラムをロードし得る。

30

#### 【0064】

[0073] BLTコマンドを受信したことに応答して、コマンドエンジン32は、GPU 10に、BLTコマンドに対応するBLT動作を実行させ得る。BLT動作を実行するため、コマンドエンジン32は、いくつかの例では、BLTコマンドに基づいてラスタ演算ユニット34、バッファ36、読み取りエンジン38、書き込みエンジン40のうちの1つまたは複数を構成し、構成された構成要素に、BLTコマンドに対応するBLT動作を実行させ得る。

40

#### 【0065】

50

[0074] ラスタ演算ユニット 3 4 は、宛先データを生成するために、ソースデータに基づいて 1 つまたは複数のラスタ演算を実行し得る。ソースデータは、B L T 動作の 1 つまたは複数のソース B L T 領域に対応し得る。いくつかの例では、ラスタ演算ユニット 3 4 は、バッファ 3 6 および / または書き込みエンジン 4 0 からソースデータを受信し得る。宛先データは、B L T 動作に関連する宛先 B L T 領域に書き込まれ得る。

#### 【 0 0 6 6 】

[0075] ラスタ演算ユニット 3 4 によって実行されるラスタ演算のタイプは、たとえば、コマンドエンジン 3 2 によって構成可能であり得る。ラスタ演算のタイプは、特定の B L T 動作についてのソースデータに基づいて宛先データがどのように生成されるべきかを指定し得る。ラスタ演算ユニット 3 4 が宛先データを生成するために 2 つの異なるソース領域を組み合わせる（たとえば、合成動作）場合、ラスタ化演算のタイプは、宛先データを生成するために 2 つの異なるソース領域がどのように組み合わされるべきかを指定し得る。いくつかの例では、ラスタ演算のタイプは、宛先領域中の特定のピクセル位置についての宛先データが、ソース領域中の対応するピクセルロケーションに基づいてどのように生成されるべきかを指定する、ビットごとの演算であり得る。ソース領域中の対応するピクセルロケーションは、宛先領域中のピクセルロケーションと同じ、それぞれのソース領域の相対的なロケーションにあるピクセルロケーションであり得る。ビットごとの演算は、たとえば、ビットごとの A N D 演算、O R 演算、N O T 演算、および / または X O R 演算などのブーリアン演算を含み得る。ビットごとの演算はまた、ビット複製演算、ビット反転演算、塗りつぶし演算などを含み得る。

10

20

#### 【 0 0 6 7 】

[0076] 追加の例では、ラスタ演算のタイプは、ブレンド演算（たとえば、アルファブレンド演算）を含み得る。いくつかの例では、ブレンド演算は、ブレンド演算の実行の間に使用されるべきソースピクセル乗算器と宛先ピクセル乗算器とを指定し得る。そのような例では、ブレンド演算の実行の間に、ソースピクセル値（たとえば、ソースピクセル色および / またはソースアルファ値）は、第 1 の積を生成するためにソースピクセル乗算器によって乗算され得、対応する宛先ピクセル値（たとえば、宛先ピクセル色および / または宛先アルファ値）は、第 2 の積を生成するために宛先ピクセル乗算器によって乗算され得、ピクセルロケーションに対する宛先値を生成するために、2 つの積の合計が一緒に加算され得る。他のタイプのブレンド演算も可能である。

30

#### 【 0 0 6 8 】

[0077] いくつかの例では、コマンドエンジン 3 2 は、C P U 6 から G P U 1 0 によって受信された B L T コマンド中に含まれるデータに基づいて、ラスタ演算ユニット 3 4 を構成し得る。たとえば、コマンドエンジン 3 2 は、B L T コマンドにおいて指定されているラスタ演算のタイプに基づいて、ラスタ演算ユニット 3 4 によって実行されるべきラスタ演算のタイプを選択し、選択されたタイプのラスタ化演算を使用して B L T 動作を実行するようにラスタ演算ユニット 3 4 を構成し得る。

#### 【 0 0 6 9 】

[0078] ラスタ演算ユニット 3 4 は、B L T 動作についての宛先データを生成するために異なるソース領域を組み合わせるものとして本明細書で説明されてきたが、他の例では、G P U 1 0 中の 1 つまたは複数の他の構成要素は、B L T 動作を実行するために、ラスタ演算ユニット 3 4 に加えて、またはその代わりに使用され得る。たとえば、G P U 1 0 のシェーダユニット（図示せず）上で実行されるシェーダプログラムは、B L T 動作についての宛先データを生成するために、シェーダプログラムにおいて指定されている組合せ技法に従って、異なるソース領域を組み合わせ得る。

40

#### 【 0 0 7 0 】

[0079] バッファ 3 6 は、B L T 動作の実行中にメモリ 2 2 から取り出されたソースデータを記憶し得る。いくつかの例では、バッファ 3 6 は、G P U 1 0 内の複数のレジスタとして実装され得る。バッファ 3 6 は、たとえば、先入れ先出し（F I F O）バッファリング方式を含む、様々なバッファリング方式のいずれかを利用し得る。F I F O バッファリ

50

ング方式によれば、バッファ36に記憶されているソースデータは、ソースデータがバッファ36に配置された順序で、利用者（たとえば、書き込みエンジン40）によって利用される。

【0071】

[0080] 読取りエンジン38は、メモリサブシステム8に読み取り要求を発行し、読み取り要求に応答して受信されたソースデータをバッファ36に配置するように構成され得る。読み取り要求は、CPU6から受信されたBLTコマンドにおいて指定されている1つまたは複数のソースBLT領域からデータを集合的に読み取り得る。

【0072】

[0081] 書込みエンジン40は、バッファ36からソースデータを利用し、利用されたソースデータに基づいて宛先データを生成し、宛先データをメモリ22中の宛先領域に書き込むために書き込み要求をメモリサブシステム8に発行するように構成され得る。書き込み要求は、CPU6から受信されたBLTコマンドにおいて指定されている宛先BLT領域にデータを集合的に書き込み得る。

10

【0073】

[0082] いくつかの例では、書き込みエンジン40は、宛先データがソースデータと同一であるように、宛先データを生成し得る。そのような例は、たとえば、BLT複製動作を実行するために使用され得る。さらなる例では、書き込みエンジン40は、宛先データを生成するために少なくとも2つの異なるソース領域からのソースデータを組み合わせ得る。そのような例は、たとえば、BLT合成動作を実行するために使用され得る。そのような例では、書き込みエンジン40は、いくつかの例では、ソースデータを組み合わせ、宛先データを生成するためにラスタ演算を実行するためにラスタ演算ユニット34を使用し得る。

20

【0074】

[0083] いくつかの例では、ソースデータを利用することは、書き込みエンジン40が読み取りエンジン38からソースデータを直接受信し、ソースデータを処理することを指し得る。さらなる例では、ソースデータを利用することは、書き込みエンジン40がバッファ36からソースデータを取り出し、ソースデータを処理することを指し得る。ソースデータを処理することは、ソースデータに基づいて宛先データを生成することを含み得る。データが処理された後、書き込みエンジン40は、宛先データを書き込むことを求める書き込み要求を発行し得る。

30

【0075】

[0084] いくつかの例では、読み取りエンジン38は、読み取られる面の各線の開始が所定の整列幅の整数倍であるメモリアドレスに対応することを必要とし得る。そのような要件は、本開示で説明する整列制約の一例であり得、所定の整列幅は、整列制約についての整列パラメータであり得る。そのような整列制約は、読み取りエンジン38を実装するためのハードウェアを簡略化し得る。所定の整列幅の整数倍であるメモリアドレスは、整列境界と呼ばれることがある。読み取りエンジン38が、適切な動作のためにそのような整列制約を必要とするとき、読み取りエンジン38は、整列制約された読み取りエンジン38と呼ばれることがある。

【0076】

40

[0085] 同様に、書き込みエンジン40は、さらなる例では、書き込まれるべき面の各線の開始が所定の整列幅の整数倍に対応することを必要とし得る。そのような要件は、本開示で説明する整列制約の一例をメイ(may)し、所定の整列幅は、整列制約についての整列パラメータであり得る。そのような要件は、書き込みエンジン40を実装するためのハードウェアを簡略化し得る。書き込みエンジン40が、適切な動作のためにそのような整列制約を必要とするとき、書き込みエンジン40は、整列制約された書き込みエンジン40と呼ばれることがある。

【0077】

[0086] いくつかの例では、読み取りエンジン38と書き込みエンジン40との両方が整列制約され得る。さらなる例では、読み取りエンジン38は、整列制約され得、書き込みエンジン

50

40は、整列制約されないことがある。追加の例では、書き込みエンジン40は、整列制約され得、読み取りエンジン38は、整列制約されないことがある。別の例では、読み取りエンジン38も書き込みエンジン40も整列制約されないことがある。

#### 【0078】

[0087]メモリサブシステム8はメモリ22を含む。メモリ22は、1つまたは複数のソース面42と宛先面44とを記憶し得る。ソース面42および宛先面44の各々は、ピクセルデータの2次元アレイに対応する。ピクセルデータは、面の中の各ピクセルロケーションに対する、赤、緑、青(RGB)の色値と、いくつかの場合には、アルファ値(A)とを含み得る。ソース面42および宛先面44はピクセルデータの2次元アレイに対応するので、面の中の個々のピクセルロケーションについてのピクセルデータは、行および列の座標によってアドレス指定され得る。BLT動作において使用されるソース領域の各々は、それぞれのソース面のすべてまたはサブセットに対応し得、宛先領域は、宛先面のすべてまたはサブセットに対応し得る。ソース領域および宛先領域は同じサイズまたは形状であり得、必ずしもそうではないが、通常は長方形の形状の領域である。なお、図2に、別個のソース面42と宛先面44とを示す。いくつかの例では、宛先面44が、BLT動作のためのソース面として使用され得、および/または、ソース面42の1つまたは複数が、BLT動作のための宛先面として働き得る。メモリサブシステム8は、バス16を介してCPU6、GPU10、読み取りエンジン38、および書き込みエンジン40に通信可能に結合される。

#### 【0079】

[0088]メモリアクセス要求を受信したことに応答して、メモリサブシステム8は、メモリ22に関して要求されたメモリ動作を実行させ得る。たとえば、読み取り要求を受信したことに応答して、メモリサブシステム8は、読み取り要求において指定されている1つまたは複数のメモリアドレスに対応するメモリ22中の1つまたは複数のメモリロケーションから、メモリ22に記憶されているデータを取得し(すなわち、読み取り)、取得されたデータをGPU10(たとえば、読み取りエンジン38)に提供し得る。別の例として、書き込み要求と、書き込み要求に対応する書き込みデータとを受信したことに応答して、メモリサブシステム8は、書き込みデータを、メモリ22中の書き込み要求において指定されている1つまたは複数のメモリアドレスに対応する1つまたは複数のロケーションに記憶させ得る。いくつかの例では、メモリサブシステム8は、書き込み要求の完了が成功したという確認を、書き込みエンジン40に与え得る。

#### 【0080】

[0089]コマンドエンジン32が、BLT動作を実行するようにGPU10中の他の処理構成要素を構成するものとして本明細書で説明するが、他の例では、GPU10中の1つまたは複数の他の構成要素が、コマンドエンジン32に加えてまたはその代わりにこれらの機能を実行し得る。たとえば、専用のBLT動作制御ユニットが、コマンドエンジン32の上述の機能を実行するために使用され得、または読み取りエンジン38および書き込みエンジン40が、そのような機能を実行するように構成され得る。

#### 【0081】

[0090]上記で説明したように、GPU10は、CPU6から受信されたBLTコマンドをサービスするためにBLT動作を実行し得る。BLT動作は、BLT複製動作とBLT合成動作とを含み得る。いくつかの例では、BLT動作は、代替的に、メモリ転送動作と呼ばれることがある。BLT複製動作は、ソース面のソース領域から宛先面の宛先領域にデータを複製することを伴い得る。すなわち、GPU10は、ソース面のソース領域からデータを読み取り、ソース面のソース領域に含まれるデータの厳密な複製を、宛先面の宛先領域に書き込み得る。

#### 【0082】

[0091]BLT合成動作は、同じまたは異なるソース面の少なくとも2つの異なるソース領域を何らかの方式で組み合わせることと、ソース領域の組合せを宛先面の宛先領域に書き込むこととを伴い得る。いくつかの場合には、ソース面の1つまたは複数は、BLT合

10

20

30

40

50

成動作のための宛先面と同じ面であり得る。たとえば、B L T合成動作は、単一のソース領域と単一の宛先領域とを読み取ることと、これら2つと一緒にブレンドすることと、ブレンドされた結果を宛先領域に書き込むこととを伴い得る。

#### 【0083】

[0092]いくつかの例では、B L T合成動作を実行するとき、異なるソース領域が、ラスタ演算に従って組み合わされ得、ラスタ演算はB L T合成動作の異なる例ごとに異なり得る。ラスタ演算は、特定のB L T合成動作のためにビットがどのように組み合わされるべきであるかを指定する動作であり得る。いくつかの例では、ラスタ演算は、たとえばブーリアン論理演算などのビットごとの演算を含み得る。ブーリアン論理演算は、たとえば、ビットごとのAND演算、OR演算、NOT演算、および/またはXOR演算を含み得る。追加の例では、ラスタ演算は、ブレンド演算（たとえば、アルファブレンド演算）を含み得る。

10

#### 【0084】

[0093]B L T動作において使用されるソース面および宛先面は、同じ面または異なる面であり得、ピクセルデータの2次元アレイに各々対応し得る。ピクセルデータは、面の中の各ピクセルに対する、赤、緑、青（RGB）の色値と、いくつかの場合には、アルファ値（A）とを含み得る。面はピクセルデータの2次元アレイに対応するので、面の中の個々のピクセルロケーションについてのピクセルデータは、行および列の座標によってアドレス指定され得る。B L T動作において使用されるソース領域の各々は、それぞれのソース面のすべてまたはサブセットに対応し得る。同様に、B L T動作において使用される宛先領域は、宛先面のすべてまたはサブセットに対応し得る。ソース領域および宛先領域は同じサイズまたは形状であり得、必ずしもそうではないが、通常は長方形の形状の領域である。

20

#### 【0085】

[0094]G P U 1 0がB L T動作を実行するとき、B L T動作のために使用されるソース面および宛先面は通常、オフチップメモリ2 2中に記憶される。言い換えると、メモリ2 2は、G P U 1 0が形成、配置、および/または配設される、マイクロチップ、集積回路、および/またはダイとは異なる、マイクロチップ、集積回路、および/またはダイの上に形成、配置、および/または配設され得る。メモリ2 2は、1つまたは複数のバス、場合によっては、メモリコントローラを介してG P U 1 0に通信可能に結合され得る。

30

#### 【0086】

[0095]B L T複製動作を実行するとき、G P U 1 0は、ソース領域中のピクセルロケーションの各々からデータを読み取り、ソース領域から読み取られたデータを宛先領域中の対応するピクセルロケーションに書き込み得る。同様に、B L T合成動作を実行するとき、G P U 1 0は、複数のソース領域中のロケーションの各々からデータを読み取り、データを組み合わせ、組み合わされたデータを宛先領域に書き込み得る。

#### 【0087】

[0096]図3は、本開示による、例示的な単一のソースB L T動作を示す概念図である。図3に、その各々がピクセルデータの2次元アレイに対応するソース面5 0と宛先面5 2とを示す。ピクセルデータは、面の中の各ピクセルに対する、赤、緑、青（RGB）の色値と、いくつかの場合には、アルファ値（A）とを含み得る。面5 0、5 2の各々はピクセルデータの2次元アレイに対応するので、面中の個々のピクセルロケーションについてのピクセルデータは、行および列の座標によってインデックス付けされ得る。いくつかの例では、ソース面5 0および宛先面5 2は、ビットマップであり得る。さらなる例では、ソース面5 0および宛先面5 2は、たとえばテクスチャバッファなどのバッファリソースであり得る。図3の例示的なソース面5 0および宛先面5 2は、同じ形状およびサイズであり、長方形の形状である。ただし、他の例では、ソース面5 0と宛先面5 2とは、異なる形状および/またはサイズを有し得る。

40

#### 【0088】

[0097]ソース面5 0はソースB L T領域5 4を含み、宛先面5 2は宛先B L T領域5 6

50

を含む。ソース BLT 領域 54 は、ソース面 50 中に含まれるピクセルデータのサブセットを含み、宛先 BLT 領域 56 は、宛先面 52 中に含まれるピクセルデータのサブセットを含む。面中に含まれるピクセルデータのサブセットは、面中に含まれるピクセルデータのすべてまたはすべて未満を指し得る。図 3 に示すように、ソース BLT 領域 54 と宛先 BLT 領域 56 とは、同じサイズおよび形状であり、長方形の形状である。ただし、他の例では、ソース BLT 領域 54 および宛先 BLT 領域 56 は、長方形の形状でないことがある。

#### 【0089】

[0098]図 3 に示す例示的な BLT 動作では、GPU10 は、ソース面 50 のソース BLT 領域 54 から宛先面 52 の宛先 BLT 領域 56 にデータを転送する。概して、ソース BLT 領域 54 中の各ピクセルロケーションについてのデータは、宛先 BLT 領域 56 中の対応するピクセルロケーションに転送される。異なる領域中の異なるピクセルロケーションは、それぞれの領域におけるピクセルロケーションの各々の相対的な配置が同じであれば、対応するピクセルロケーションであると言われ得る。たとえば、ソース BLT 領域 54 および宛先 BLT 領域 56 の第 2 行の第 3 列は、同じ相対的なロケーションを有する。

10

#### 【0090】

[0099] 単一のソース領域が BLT 動作のためのソースオペランドとして使用されるので、図 3 に示す BLT 動作は、単一ソースの BLT 動作である。図 3 に示す単一ソースの BLT 動作は、ソース BLT 領域 54 のデータが宛先 BLT 領域 56 に複製される BLT 複製動作を表し得る。図 3 に示す BLT 動作に別個のソース面 50 と宛先面 52 を示すが、他の例では、ソース面 50 と宛先面 52 とは同じ面であり得る。そのような例では、ソース BLT 領域 54 と宛先 BLT 領域 56 とは、いくつかの例では、BLT 複製動作によってデータが転送される同じ面内の異なる領域であり得る。

20

#### 【0091】

[0100] 図 4 は、本開示による、例示的な複数のソース BLT 動作を示す概念図である。図 4 に、第 1 のソース面 58 と、第 2 のソース面 60 と、宛先面 62 とを示す。面 58、60、62 の各々は、図 3 に関する上記で説明した面 50、52 と実質的に同様であり得る。第 1 のソース面 58 は、第 1 のソース BLT 領域 64 を含み、第 2 のソース面 60 は、第 2 のソース BLT 領域 66 を含み、宛先面 62 は、宛先 BLT 領域 68 を含む。第 1 のソース BLT 領域 64 は、第 1 のソース面 58 中に含まれるピクセルデータのサブセットを含み、第 2 のソース BLT 領域 66 は、第 2 のソース面 60 中に含まれるピクセルデータのサブセットを含み、宛先 BLT 領域 68 は、宛先面 62 中に含まれるピクセルデータのサブセットを含む。図 4 に示すように、第 1 のソース BLT 領域 64、第 2 のソース BLT 領域 66、および宛先 BLT 領域 68 は、同じサイズおよび形状であり、長方形の形状である。ただし、他の例では、領域 64、66、および 68 は、長方形の形状でないことがある。

30

#### 【0092】

[0101] 図 4 の例示的な BLT 動作では、第 1 のソース BLT 領域 64 からのソースデータおよび第 2 のソース BLT 領域 66 からのソースデータが、宛先データを生成するためには、ラスタ演算 70 を介して組み合わされ、宛先データは、宛先面 62 の宛先 BLT 領域 68 に配置される。一般に、第 1 のソース BLT 領域 64 中の各ピクセル位置のデータは、第 2 のソース BLT 領域 66 中の対応するピクセル位置からのデータと組み合わされ、宛先 BLT 領域 68 中の対応するピクセル位置に書き込まれる。異なる領域中の異なるピクセルロケーションは、それぞれの領域におけるピクセルロケーションの各々の相対的な配置が同じであれば、対応するピクセルロケーションであると言われ得る。

40

#### 【0093】

[0102] 複数のソース領域が BLT 動作のためのソースオペランドとして使用されるので、図 4 に示す BLT 動作は、複数ソースの BLT 動作である。図 4 に示す複数ソースの BLT 動作は、宛先領域に書き込まれる宛先データを生成するために GPU10 が 2 つの異なるソース領域からのデータを組み合わせるかまたは合成する BLT 合成動作を表し得る

50

。図4に示すB L T動作は、別個の第1のソース面58と、第2のソース面60と、宛先面62とを示すが、他の例では、面58、60、62のうちの1つまたは複数が同じ面であり得る。たとえば、第2のソース面60と宛先面62とが同じ面であり得、第2のソースB L T領域66と宛先B L T領域68とがその面内の同じ領域であり得る。そのような例では、B L T合成動作は、特定のラスタ演算子を使用して、第1のソースB L T領域64中のピクセルデータを、宛先面62中の既存のピクセルデータと統合し得る。

#### 【0094】

[0103]図5は、図1および図2の例示的なコンピューティングデバイスのための例示的なコマンド処理フロー72を示す概念図である。図5に示すように、ソフトウェアアプリケーション24は、GPUドライバ28にソフトウェアB L Tコマンド74を発行する。GPUドライバ28は、ソフトウェアB L Tコマンド74を受信し、ソフトウェアB L Tコマンド74に基づいて1つまたは複数のGPU B L Tコマンド76を生成し、GPU10にGPU B L Tコマンド76を発行する。GPU10は、GPU B L Tコマンド76をサービスするためにB L T動作78を実行する。たとえば、GPU B L Tコマンド76の各々について、GPU10は、それぞれの整列B L T動作78を実行し得る。

10

#### 【0095】

[0104]いくつかの例では、ソフトウェアB L Tコマンド74は、非整列B L Tコマンドであり得、GPU B L Tコマンド76は、ソフトウェアB L Tコマンド74に対応する複数の整列B L Tコマンドを含み得る。そのような例では、GPUドライバ28は、本開示で説明する技法のいずれかに従って非整列ソフトウェアB L Tコマンド74を複数の整列GPU B L Tコマンド76に変換し得る。

20

#### 【0096】

[0105]図6は、本開示による、整列仮想面を定義するための例示的な技法を示す概念図である。図6に、非整列面80と仮想面82とを示す。非整列面80は、非整列B L Tコマンドの一部としてソフトウェアアプリケーション24から受信され得る。GPUドライバ28は、非整列面80に基づいて仮想面82を定義し得る。

#### 【0097】

[0106]図6の例では、32バイトの整列制約されたGPU(たとえば、GPU10)が、B L T動作を実行するために使用される。32バイトの整列制約されたGPUは、面の線の各々の開始が32(すなわち、所定の整列幅)の整数倍であることを必要とするGPUに対応し得る。32(すなわち、整列幅)の各倍数は、整列境界と呼ばれることがある。

30

#### 【0098】

[0107]図6に示すように、非整列面80のための面幅は720バイトであり、非整列面80のための面高さは480本の線(lines)である。720は、32によって割り切れないで、非整列面80のいくつかの線は、32の倍数であるメモリアドレスにおいて開始しないことになるだろう。したがって、面80は、非整列であると言われる。

#### 【0099】

[0108]GPUドライバ28は、仮想面82中で单一の線を形成するために非整列面80からの2つの線のセットと一緒に組み合わせることによって仮想面82を定義し得る。図6に示すように、仮想面82は、非整列面80の2倍の幅があり、半分の高さがある。仮想面82の面幅は1440バイトであり、これは、32によって割り切れる。したがって、仮想面82は、32バイトの整列制約されたGPUがB L T動作を実行し得る整列面である。

40

#### 【0100】

[0109]図7および図8は、図6に示す技法に従って定義された整列仮想面を使用した非整列B L T動作の2つの整列B L T動作への例示的な変換を示す概念図である。

#### 【0101】

[0110]図7に、ソフトウェアアプリケーション24から受信されたB L Tコマンドに関する連付けられ得る単一の非整列B L T動作を示す。非整列B L Tコマンドは、B L T動作の

50

ための非整列ソース面 8 4 と非整列宛先面 8 6 とを指定する。ソース面 8 4 はソース B L T 領域 8 8 を含み、宛先面 8 6 は宛先 B L T 領域 9 0 を含む。図 7 に示すように、非整列 B L T 動作は、ソース面 8 4 のソース B L T 領域 8 8 を宛先面 8 6 の宛先 B L T 領域 9 0 に転送する。

#### 【 0 1 0 2 】

[0111]図 8 に、単一の非整列 B L T 動作に対応する整列 2 B L T 動作 (aligned two-BL T operation) を示す。整列 2 B L T 動作は、2 つの整列 B L T コマンドによって指定された 2 つの別個の B L T 動作を含み得る。整列 B L T コマンドの各々は、B L T 動作のための整列仮想ソース面 9 2 と整列仮想宛先面 9 4 とを指定し得る。仮想ソース面 9 2 はソース B L T 領域 9 6、9 8 を含み、仮想宛先面 9 4 は宛先 B L T 領域 1 0 0、1 0 2 を含む。 10

#### 【 0 1 0 3 】

[0112]仮想ソース面 9 2 は、ソース面 8 4 に基づいて定義され得、仮想宛先面 9 4 は、宛先面 8 6 に基づいて定義され得る。G P U ドライバ 2 8 は、ソース面 8 4 を仮想ソース面 9 2 に変換することと、宛先面 8 6 を仮想宛先面 9 4 に変換することとを行うために図 6 に示した技法を使用し得る。

#### 【 0 1 0 4 】

[0113]たとえば、G P U ドライバ 2 8 は、仮想ソース面 9 2 中で単一の線を形成するためにソース面 8 4 中で 2 つの線のあらゆる (every) セットと一緒に組み合わせ得る。これにより、ソース面 8 4 の偶数線のすべてが仮想ソース面 9 2 の左側にアグリゲートされ、ソース面 8 4 の奇数線のすべてが仮想ソース面 9 2 の右側にアグリゲートされることが効果的に行われるようになる。G P U ドライバ 2 8 は、仮想宛先面 9 4 を形成するために宛先面 8 6 の線と一緒に組み合わせるために同様の技法を使用し得る。 20

#### 【 0 1 0 5 】

[0114]図 7 では、領域 8 8、9 0 の 1 つおきの線が交互に (in an alternating fashion) ハッチングされる。図 8 に示すように、仮想ソース面 9 2 を形成するためにソース面 8 4 の線を組み合わせることにより、ソース B L T 領域 8 8 が、それがもはや 1 つの連続する長方形領域でないよう変換されるようになる。代わりに、ソース B L T 領域 8 8 は、ソース B L T 領域 9 6 および 9 8 に変換され、ここで、ソース B L T 領域 9 6 は、ソース B L T 領域 8 8 の偶数線を表し、ソース B L T 領域 9 8 は、ソース B L T 領域 8 8 の奇数線を表す。したがって、ソース B L T 領域 8 8 は、互いに連続しない 2 つの別個の長方形領域に変換された。宛先 B L T 領域 9 0 に関して、同様の変換が行われ、ここで、宛先 B L T 領域 9 0 は、宛先 B L T 領域 1 0 0 および 1 0 2 に変換される。 30

#### 【 0 1 0 6 】

[0115]この例では、単一の B L T 動作は、不連続の B L T 領域に対して実行されないことがある。現在、2 つの別個のソース B L T 領域 9 6、9 8 と 2 つの別個の宛先 B L T 領域 1 0 0、1 0 2 があるので、G P U ドライバ 2 8 は、非整列面 8 4、8 6 のための単一の非整列 B L T コマンドを仮想面 9 2 および 9 4 に関して実行される 2 つの整列 B L T コマンド、ハッチングされた領域 9 6、1 0 0 のための 1 つの B L T コマンドと白色領域 9 8、1 0 2 のための 1 つの B L T コマンドとに変換する。このようにして、G P U ドライバ 2 8 は、G P U 1 0 に、2 つの B L T 動作を実行させ得、ここで、第 1 の B L T 動作は、ソース B L T 領域 9 6 を宛先 B L T 領域 1 0 0 に転送し、第 2 の B L T 動作は、ソース B L T 領域 9 8 を宛先 B L T 領域 1 0 2 に転送する。 40

[0116]G P U ドライバ 2 8 によって生成された複数の B L T コマンドは、ソース B L T 領域 8 8 および宛先 B L T 領域 9 0 全体を集合的に処理し、それによって、単一の非整列 B L T コマンドによって生成されたであろうメモリ状態と同じ結果として生じるメモリ状態を生成し得る。さらに、整列 B L T コマンドの各々が、3 2 バイトの整列仮想面 (すなわち、仮想面 9 2 および 9 4) に関して実行され、これにより、B L T コマンドを 3 2 バイトの整列制約された G P U によって処理することが可能になる。このようにして、G P U ドライバ 2 8 は、非整列 B L T コマンドを整列制約された G P U によってサービスされ 50

得る複数の整列 B L T コマンドに変換し、それによって、比較的低電力の G P U 加速 B L T 处理の利益を取得し得る。

【 0 1 0 7 】

[0117] いくつかの例では、非整列 B L T コマンドを複数の整列 B L T コマンドに変換するため、G P U ドライバ 2 8 は、非整列 B L T コマンドに関連する B L T ソースまたは宛先領域の座標および次元を複数の整列 B L T コマンドのために使用されるべき B L T 領域のための座標および次元に変換し得る。いくつかの例では、図 7 および図 8 に示す 2 つの B L T 変換動作について、G P U ドライバ 2 8 は、以下の擬似コードに基づいて B L T 領域座標および次元を生成し得る。

【 0 1 0 8 】

【 数 1 】

```
BLT1:  New_SRC_X = SRC_X + (SRC_Y & 0x1)* SRC_width;
        New_SRC_Y = SRC_Y/2;
        New_DST_X = DST_X + (DST_Y & 0x1) * DST_width;
        New_DST_Y = DST_Y/2;
        New_BLT_width = BLT_width
        New_BLT_height = (BLT_Height +1)/2;
        New_SRC_SURFACE_base = SRC_SURFACE_base; // 変更なし
        New_SRC_SURFACE_WIDTH = SRC_SURFACE_WIDTH*2
        New_SRC_SURFACE_HEIGHT = SRC_SURFACE_HEIGHT +
        ((~SRC_Y)&1)/2
        New_DST_SURFACE_base = DST_SURFACE_base // 変更なし
        New_DST_SURFACE_WIDTH = DST_SURFACE_WIDTH*2
        New_DST_SURFACE_HEIGHT = DST_SURFACE_HEIGHT+
        ((~SRC_Y)&1)/2
```

10

20

30

```
BLT2:  New_SRC_X = SRC_X + ((SRC_Y+1) & 0x1)* SRC_width;
        New_SRC_Y = (SRC_Y+1)/2;
        New_DST_X = DST_X + ((DST_Y+1) & 0x1) * DST_width;
        New_DST_Y = (DST_Y+1)/2;
        New_BLT_width = BLT_width
```

```

New BLT_height = (BLT_Height)/2;
New SRC_SURFACE_base = SRC_SURFACE_base; // 変更なし
New SRC_SURFACE_WIDTH = SRC_SURFACE_WIDTH*2;

```

```

New SRC_SURFACE_HEIGHT = (SRC_SURFACE_HEIGHT +
  ((~SRC_Y+1)&1)/2;
New DST_SURFACE_base = DST_SURFACE_base; // 変更なし
New DST_SURFACE_WIDTH = DST_SURFACE_WIDTH*2;

```

10

```

New DST_SURFACE_HEIGHT = (DST_SURFACE_HEIGHT+1) +
  ((~SRC_Y+1)&1)/2;

```

### 【 0 1 0 9 】

例示的な擬似コードでは、BLT1 および BLT2 は、単一の非整列 BLT コマンドに基づいて生成される異なる整列 BLT コマンドに対応し、SRC\_X は、ソース BLT 領域 88 の左上隅の x 座標に対応し、SRC\_Y は、ソース BLT 領域 88 の左上隅の y 座標に対応し、DST\_X は、宛先 BLT 領域 90 の左上隅の x 座標に対応し、DST\_Y は、宛先 BLT 領域 90 の左上隅の y 座標に対応し、BLT\_width は、(たとえば、バイト単位での) ソース BLT 領域 88 の幅に対応し、BLT\_Height は、(たとえば、線単位での) ソース BLT 領域 88 の高さに対応し、SRC\_width は、(たとえば、バイト単位での) ソース面 84 の面幅に対応し、DST\_width は、(たとえば、バイト単位での) 宛先面 86 の面幅に対応し、SRC\_SURFACE\_base は、ソース面 84 の面ポインタに対応し、SRC\_SURFACE\_WIDTH は、ソース面 84 の面幅に対応し、SRC\_SURFACE\_HEIGHT は、ソース面 84 の高さに対応し、DST\_SURFACE\_base は、宛先面 86 の面ポインタに対応し、DST\_SURFACE\_WIDTH は、宛先面 86 の面幅に対応し、DST\_SURFACE\_HEIGHT は、宛先面 86 の高さに対応し、New\_SRC\_X は、(BLT1 の場合は) ソース BLT 領域 96 または (BLT2 の場合は) ソース BLT 領域 98 の左上隅の x 座標に対応し、New\_SRC\_Y は、(BLT1 の場合は) ソース BLT 領域 96 または (BLT2 の場合は) ソース BLT 領域 98 の左上隅の y 座標に対応し、New\_DST\_X は、(BLT1 の場合は) 宛先 BLT 領域 100 または (BLT2 の場合は) 宛先 BLT 領域 102 の左上隅の x 座標に対応し、New\_DST\_Y は、(BLT1 の場合は) 宛先 BLT 領域 100 または (BLT2 の場合は) 宛先 BLT 領域 102 の左上隅の y 座標に対応し、New\_BLT\_width は、(BLT1 の場合は) ソース BLT 領域 96 または (BLT2 の場合は) ソース BLT 領域 98 の(たとえば、バイト単位での) 幅に対応し、New\_BLT\_Height は、(BLT1 の場合は) ソース BLT 領域 96 または (BLT2 の場合は) ソース BLT 領域 98 の(たとえば、線単位での) 高さに対応し、

New\_SRC\_SURFACE\_base は、仮想ソース面 92 の面ポインタに対応し、New\_SRC\_SURFACE\_WIDTH は、仮想ソース面 92 の面幅に対応し、New\_SRC\_SURFACE\_HEIGHT は、仮想ソース面 92 の高さに対応し、New\_DST\_SURFACE\_base は、仮想宛先面 94 の面ポインタに対応し、New\_DST\_SURFACE\_WIDTH は、仮想宛先面 94 の面幅に対応し、New\_DST\_SURFACE\_HEIGHT は、仮想宛先面 94 の高さに対応する。

### 【 0 1 1 0 】

[0118] 例では、擬似コード、演算子「/」は、整数除算に対応し、演算子「&」は、ビット単位の (bit-wise) AND 演算に対応し、プレフィックス「0x」は、以下の数が 1

20

30

40

50

6進定数であることを示す。いくつかの例では、仮想ソース面92のNew\_SRC\_SSURFACE\_HEIGHTの値は、BLT1とBLT2とで異なり、その結果、BLT1とBLT2とのためにわずかに異なる仮想ソース面が使用されることに結果し得る。さらなる例では、仮想宛先面94のNew\_DST\_SURFACE\_HEIGHTの値は、BLT1とBLT2とで異なり、その結果、BLT1とBLT2とのためにわずかに異なる仮想宛先面が使用されることに結果し得る。

#### 【0111】

[0119]図9は、本開示による、複数の整列仮想面を定義するための例示的な技法を示す概念図である。図9に、非整列面110と2つの仮想面112、114とを示す。非整列面110は、非整列BLTコマンドの一部としてソフトウェアアプリケーション24から受信され得る。仮想面112、114は、非整列面110に基づいてGPUドライバ28によって定義され得る。

#### 【0112】

[0120]図9の例では、32バイトの整列制約されたGPUが、BLT動作を実行するために使用される。図9に示すように、非整列面110のための面幅は720バイトであり、非整列面110のための面高さは480本の線である。720は、32によって割り切れないで、非整列面110のためのいくつかの線は、32の倍数であるメモリアドレスにおいて開始しないことになる。したがって、面110は、非整列であると言われる。

#### 【0113】

[0121]この問題に対処するために、CPU6は、図6に関して上記で説明したのと同様の方法で非整列面110の幅を広げ得る。これにより、広げられた面幅が、整列幅制約に対応するようになる。ただし、非整列面110は、1の面ポインタを有する。1は32によって割り切れないで、広げられた面の面ポインタは非整列である。したがって、広げられた面は、依然として整列していない。

#### 【0114】

[0122]この問題に対処するための1つの手法は、シフトされた面ポインタが最も近い整列境界を指すように面ポインタをシフトすることである。図6～図8に関して上記で説明した技法と同様に、面を広げることは、長方形BLT領域を複数の別個の長方形領域に変換し得、GPUドライバ28は、複数のBLTコマンドを生成し得、ここで、BLTコマンドの各々は、GPUに、長方形BLT領域の各1つ(respective one)に対してBLT動作を実行させる。ただし、仮想面の幅が同じままである場合、面ポインタをシフトすることにより、仮想面の境界の近くにある変換された長方形BLT領域が仮想面の端にラップアラウンドされるようになり得る。言い換えれば、そのような変換されたBLT長方形領域は、もはや連続でないことがあり、単一のBLT動作によってサービスされることができないことがある。

#### 【0115】

[0123]この問題に対処するために、GPUドライバ28は、ソース面と宛先面との各々のための2つの仮想面112、114を生成し得る。仮想面112、114は、同じ広げられた面幅を有するが、異なる整列面ポインタを有し得る。いくつかの例では、GPUドライバ28は、仮想面112の非整列面ポインタに最も近い2つの整列境界を決定し、最も近い整列境界のうちの1つを仮想面112のための面ポインタとして割り当て、最も近い整列境界の他方を仮想面114のための面ポインタとして割り当て得る。図9中の例では、1の非整列面ポインタに最も近い32バイトの整列境界は、0および32である。したがって、GPUドライバ28は、仮想面112の面ポインタを0に設定し、仮想面114の面ポインタを32に設定する。

#### 【0116】

[0124]仮想面112、114の各々の面幅は1440バイトであり、これは、32によって割り切れる。さらに、仮想面のための面ポインタの各々は、メモリアドレス(すなわち、0および32)を指し、これは、32によって割り切れる。したがって、仮想面112、114の両方は、32バイトの整列制約されたGPUがBLT動作を実行し得る整列

10

20

30

40

50

面である。

【0117】

[0125]図9に示すように、非整列面110の偶数線の各々は、仮想面112の線の各1つ中に完全に含まれている。同様に、非整列面110の奇数線の各々は、仮想面114の線の各1つ中に完全に含まれている。したがって、GPUドライバ28は、GPU10に、仮想面112を使用して非整列面110の偶数線に関して第1の整列BLT動作を実行させ、仮想面114を使用して非整列面110の奇数線に関して第2の整列BLT動作を実行させ得る。このようにしてBLT動作を実行することは、仮想面112、114のいずれかの端をラップアラウンドする変換された長方形BLT領域を処理する必要を回避し得る。

10

【0118】

[0126]図10および図11は、本開示による、図9において定義されている複数の整列仮想面を使用して非整列BLT動作の2つの整列BLT動作への例示的な変換を示す概念図である。図7に示した単一の非整列BLT動作から開始して、GPUドライバ28は、単一のBLT動作を図10に示す偶数BLT動作と図11に示す奇数BLT動作とに変換し得る。

【0119】

[0127]仮想ソース面116、128は、図7中のソース面84に基づいて定義され得、仮想宛先面118、130は、図7中の宛先面86に基づいて定義され得る。GPUドライバ28は、ソース面84を仮想ソース面116、128に変換することと、宛先面86を仮想宛先面118、130に変換することを行うために図9に示した技法を使用し得る。詳細には、仮想面116、118は、図9中の仮想面112に対応し得、仮想面128、130は、図9中の仮想面114に対応し得る。仮想ソース面116、128はそれぞれ、ソースBLT領域120、122を含み、仮想宛先面118、130はそれぞれ、宛先BLT領域124、126を含む。

20

【0120】

[0128]GPUドライバ28は、GPU10に、ソースBLT領域120と宛先BLT領域124とに関して偶数BLT動作(図10)を実行させる。詳細には、偶数BLT動作は、仮想ソース面116のソースBLT領域120を仮想宛先面118の宛先BLT領域124に転送し得る。同様に、GPUドライバ28は、GPU10に、ソースBLT領域122と宛先BLT領域126とに関して奇数BLT動作(図11)を実行させる。詳細には、奇数BLT動作は、仮想ソース面128のソースBLT領域122を仮想宛先面130の宛先BLT領域126に転送し得る。

30

【0121】

[0129]図10に示すように、宛先BLT領域126は、仮想宛先面118の端をラップアラウンドする。同様に、図11において、ソースBLT領域120は、仮想ソース面128の端をラップアラウンドする。これらの領域のラップアラウンドにより、これらの領域は不連続になり、それによって、単一のBLT動作が領域を処理することが可能になることを防ぐ。ただし、上記で説明した方法で偶数および奇数BLT動作を実行することによって、これらのラップアラウンドされた領域の処理が回避され、それによって、BLT動作の効率が増加し得る。

40

【0122】

[0130]GPU10に、偶数および奇数BLT動作を実行させるために、GPUドライバ28は、図7の非整列面84、86のための単一の非整列BLTコマンドを2つの整列BLTコマンド、仮想面116、118および影つき領域120、124に関して実行される1つの偶数BLTコマンドと、仮想面128、130および白色領域122、126に関して実行される1つの奇数BLTコマンドとに変換し得る。

【0123】

[0131]GPUドライバ28によって生成された複数のBLTコマンドは、ソース領域および宛先領域全体を集合的に処理し、それによって、単一の非整列BLTコマンドによっ

50

て生成されたであろうメモリ状態と同じ結果として生じるメモリ状態を生成し得る。さらに、整列 B L T コマンドの各々が、32 バイトの整列仮想面に関して実行され、これにより、B L T コマンドを32 バイトの整列制約された G P U によって処理することが可能になる。このようにして、G P U ドライバ28は、非整列 B L T コマンドを整列制約された G P U によってサービスされ得る複数の整列 B L T コマンドに変換し、それによって、比較的低電力の G P U 加速 B L T 処理の利益を取得し得る。

【 0 1 2 4 】

[0132]いくつかの例では、非整列 B L T コマンドを複数の整列 B L T コマンドに変換するにあたり、G P U ドライバ28は、非整列 B L T コマンドに関連する宛先領域または B L T ソースの座標および次元を複数の整列 B L T コマンドのために使用されるべき B L T 領域のための座標および次元に変換し得る。いくつかの例では、図7、図10および図11に示す2つの B L T 変換動作について、G P U ドライバ28は、以下の擬似コードに基いて B L T 領域座標および次元を生成し得る。

【 0 1 2 5 】

## 【数2】

```

BLT1:  SRC_align = (SRC_Y & 0x1)?((SRC_base & 0x3f) - 64) : SRC_base &
0x3f;
          DST_align = (DST_Y & 0x1)?((DST_base & 0x3f) - 64) : DST_base &
0x3f;
          NEW_SRC_base = SRC_base - SRC_align;
          NEW_DST_base = DST_base - DST_align;
          New_SRC_X = SRC_X + SRC_align + (SRC_Y & 0x1)* SRC_width;
          New_SRC_Y = SRC_Y/2;
          New_DST_X = DST_X + DST_align + (DST_Y & 0x1) * DST_width;
          New_DST_Y = DST_Y/2;
          New_BLT_width = BLT_width
          New_BLT_height = (BLT_Height +1)/2;
          New_SRC_SURFACE_WIDTH = SRC_SURFACE_WIDTH*2;
          New_SRC_SURFACE_HEIGHT = SRC_SURFACE_HEIGHT +
          (~SRC_Y)&1)/2;
          New_DST_SURFACE_WIDTH = DST_SURFACE_WIDTH*2;
          New_DST_SURFACE_HEIGHT = DST_SURFACE_HEIGHT +
          (~SRC_Y)&1)/22;
BLT2:  SRC_align = ((SRC_Y+1)& 0x1)?((SRC_base & 0x3f) - 64) : SRC_base
& 0x3f;
          DST_align = ((DST_Y+1)& 0x1)?((DST_base & 0x3f) - 64) :
DST_base & 0x3f;
          NEW_SRC_base = SRC_base - SRC_align;
          NEW_DST_base = DST_base - DST_align;
          New_SRC_X = SRC_X + SRC_align + ((SRC_Y+1) & 0x1)*
SRC_width;
          New_SRC_Y = (SRC_Y+1)/2;
          New_DST_X = DST_X + DST_align + ((DST_Y+1) & 0x1) *
DST_width;
          New_DST_Y = (DST_Y+1)/2;
          New_BLT_width = BLT_width
          New_BLT_height = (BLT_Height)/2;
          New_SRC_SURFACE_WIDTH = SRC_SURFACE_WIDTH*2;
          New_SRC_SURFACE_HEIGHT = (SRC_SURFACE_HEIGHT+1) +
          ((~SRC_Y+1)&1)/2;
          New_DST_SURFACE_WIDTH = DST_SURFACE_WIDTH*2;
          New_DST_SURFACE_HEIGHT = (DST_SURFACE_HEIGHT+1) +
          ((~SRC_Y+1)&1)/2;

```

## 【0 1 2 6】

例示的な擬似コードでは、BLT1は、図10に示す偶数BLT動作に対応し、BLT2は、図11に示す奇数BLT動作に対応し、SRC\_Xは、ソースBLT領域88の左上隅のx座標に対応し、SRC\_Yは、ソースBLT領域88の左上隅のy座標に対応し、DST\_Xは、宛先BLT領域90の左上隅のx座標に対応し、DST\_Yは、宛先BLT領域90の左上隅のy座標に対応し、BLT\_widthは、(たとえば、バイト単位

10

20

30

40

50

での ) ソース B L T 領域 8 8 の幅に対応し、 B L T \_ H e i g h t は、 ( たとえば、線単位での ) ソース B L T 領域 8 8 の高さに対応し、 S R C \_ w i d t h は、 ( たとえば、バイト単位での ) ソース面 8 4 の面幅に対応し、 D S T \_ w i d t h は、 ( たとえば、バイト単位での ) 宛先面 8 6 の面幅に対応し、 S R C \_ b a s e は、 ソース面 8 4 の面ポインタに対応し、 D S T \_ b a s e は、 宛先面 8 6 の面ポインタに対応し、 S R C \_ S U R F A C E \_ W I D T H は、 ソース面 8 4 の面幅に対応し、 S R C \_ S U R F A C E \_ H E I G H T は、 ソース面 8 4 の高さに対応し、 D S T \_ S U R F A C E \_ W I D T H は、 宛先面 8 6 の面幅に対応し、 D S T \_ S U R F A C E \_ H E I G H T は、 宛先面 8 6 の高さに対応し、 N e w \_ S R C \_ X は、 ( B L T 1 の場合は ) ソース B L T 領域 1 2 0 または ( B L T 2 の場合は ) ソース B L T 領域 1 2 2 の左上隅の x 座標に対応し、 N e w \_ S R C \_ Y は、 ( B L T 1 の場合は ) ソース B L T 領域 1 2 0 または ( B L T 2 の場合は ) ソース B L T 領域 1 2 2 の左上隅の y 座標に対応し、 N e w \_ D S T \_ X は、 ( B L T 1 の場合は ) 宛先 B L T 領域 1 2 4 または ( B L T 2 の場合は ) 宛先 B L T 領域 1 2 6 の左上隅の x 座標に対応し、 N e w \_ D S T \_ Y は、 ( B L T 1 の場合は ) 宛先 B L T 領域 1 2 4 または ( B L T 2 の場合は ) 宛先 B L T 領域 1 2 6 の左上隅の y 座標に対応し、 N e w \_ B L T \_ w i d t h は、 ( B L T 1 の場合は ) ソース B L T 領域 1 2 0 または ( B L T 2 の場合は ) ソース B L T 領域 1 2 2 の ( たとえば、バイト単位での ) 幅に対応し、 N e w \_ B L T \_ H e i g h t は、 ( B L T 1 の場合は ) ソース B L T 領域 1 2 0 または ( B L T 2 の場合は ) ソース B L T 領域 1 2 2 の ( たとえば、線単位での ) 高さに対応し、 N e w \_ S R C \_ b a s e は、 ( B L T 1 の場合は ) 仮想ソース面 1 1 6 または ( B L T 2 の場合は ) 仮想ソース面 1 2 8 の面ポインタに対応し、 N e w \_ D S T \_ b a s e は、 ( B L T 1 の場合は ) 仮想宛先面 1 1 8 または ( B L T 2 の場合は ) 仮想宛先面 1 3 0 の面ポインタに対応し、 N e w \_ S R C \_ S U R F A C E \_ W I D T H は、 ( B L T 1 の場合は ) 仮想ソース面 1 1 6 または ( B L T 2 の場合は ) 仮想ソース面 1 2 8 の面幅に対応し、 N e w \_ S R C \_ S U R F A C E \_ H E I G H T は、 ( B L T 1 の場合は ) 仮想ソース面 1 1 6 または ( B L T 2 の場合は ) 仮想ソース面 1 2 8 の高さに対応し、 N e w \_ D S T \_ S U R F A C E \_ W I D T H は、 ( B L T 1 の場合は ) 仮想宛先面 1 1 8 または ( B L T 2 の場合は ) 仮想宛先面 1 3 0 の面幅に対応し、 N e w \_ D S T \_ S U R F A C E \_ H E I G H T は、 ( B L T 1 の場合は ) 仮想宛先面 1 1 8 または ( B L T 2 の場合は ) 仮想宛先面 1 3 0 の高さに対応する。

#### 【 0 1 2 7 】

[0133] 例示的な擬似コードでは、演算子「 / 」は、整数除算に対応し、演算子「 & 」は、ビット単位の A N D 演算に対応し、 ( A ) ? ( B ) : ( C ) 演算子は、 3 値演算子 ( たとえば、 i f - t h e n - e l s e フレームワーク ( 詳細には、 A の場合、 B であり、それ以外の場合 C である ) ) に対応し、プレフィックス「 0 x 」は、以下の数が 16 進定数であることを示す。いくつかの例では、仮想ソース面 9 2 の N e w \_ S R C \_ S U R F A C E \_ H E I G H T は、 B L T 1 と B L T 2 とで異なり、その結果、 B L T 1 と B L T 2 とのためにわずかに異なる仮想ソース面が使用されることに結果し得る。さらなる例では、仮想宛先面 9 4 の N e w \_ D S T \_ S U R F A C E \_ H E I G H T は、 B L T 1 と B L T 2 とで異なり、その結果、 B L T 1 と B L T 2 とのためにわずかに異なる仮想宛先面が使用されることに結果し得る。

#### 【 0 1 2 8 】

[0134] 図 1 2 は、本開示による、非整列 B L T 動作を処理するための例示的な技法を示す流れ図である。 C P U 6 は、非整列 B L T コマンドを受信する ( 1 3 1 ) 。 C P U 6 ( たとえば、 G P U ドライバ 2 8 ) は、非整列 B L T コマンドを複数の整列 B L T コマンドに変換する ( 1 3 2 ) 。

#### 【 0 1 2 9 】

[0135] いくつかの例では、非整列 B L T コマンドは、非整列 B L T コマンドに関連する面のうちの少なくとも 1 つのための第 1 の面幅を指定する。そのような例では、非整列 B L T コマンドを変換するために、 C P U 6 は、第 1 の面幅に基づいて第 2 の面幅を決定す

10

20

30

40

50

ることと、ここにおいて、第2の面幅が第1の面幅のN倍である、複数の整列B L Tコマンドの各々が、複数の整列B L Tコマンドの各々に関連する面のうちの少なくとも1つのための第2の面幅を指定するように複数の整列B L Tコマンドを生成することを行い得る。そのような例では、Nは、2以上の整数であり得る。いくつかの例では、Nは2に等しくなり得る。

【0130】

[0136]さらなる例では、第2の面幅を決定するために、CPU6は、第1の面幅と、複数の整列B L Tコマンドを実行するプロセッサに関連する所定の整列幅制約とに基づいて第2の面幅を決定し得る。

【0131】

[0137]追加の例では、第1の面幅は、所定の整列幅制約の整数倍でないことがある。そのような例では、第1の面幅と所定の整列幅制約とに基づいて第2の面幅を決定するために、CPU6は、第2の面幅が所定の整列幅制約の整数倍であるように第2の面幅を決定し得る。

【0132】

[0138]いくつかの例では、複数の整列B L Tコマンドは、第1の整列B L Tコマンドと第2の整列B L Tコマンドとを含み得、非整列B L Tコマンドは、第1のB L T領域を指定する。そのような例では、複数の整列B L Tコマンドを生成するために、CPU6は、第1のB L T領域に基づいて第2のB L T領域と第3のB L T領域とを決定することと、第2のB L T領域が、第3のB L T領域とは異なる、第1の整列B L Tコマンドが第2のB L T領域を指定するように第1の整列B L Tコマンドを生成することと、第2の整列B L Tコマンドが第3のB L T領域を指定するように第2の整列B L Tコマンドを生成することを行い得る。場合によっては、第2のB L T領域は、第1のB L T領域の偶数線を含み得、第3のB L T領域は、第1のB L T領域の奇数線を含み得る。

【0133】

[0139]さらなる例では、非整列B L Tコマンドは、非整列B L Tコマンドに関連する第1の面の第1の面ポインタ値を指定する。そのような例では、複数の整列B L Tコマンドを生成するために、CPU6は、複数の整列B L Tコマンドの各々が、複数の整列B L Tコマンドの各々に関連する面のうちの1つの第1の面ポインタ値を指定するように複数の整列B L Tコマンドを生成し得る。

【0134】

[0140]追加の例では、複数の整列B L Tコマンドは、第1の整列B L Tコマンドと第2の整列B L Tコマンドとを含み、非整列B L Tコマンドは、第1のB L T領域を指定する。そのような例では、複数の整列B L Tコマンドを生成するために、CPU6は、第1のB L T領域に基づいて第2のB L T領域と第3のB L T領域とを決定することと、第2のB L T領域が、第3のB L T領域とは異なる、第1の整列B L Tコマンドが第2の面幅と、第1の面ポインタ値と、第2のB L T領域とを指定するように第1の整列B L Tコマンドを生成することを行い得る。そのような例では、CPU6は、さらに、第2の整列B L Tコマンドが、第2の面幅と、第1の面ポインタ値と、第3のB L T領域とを指定するように第2の整列B L Tコマンドを生成し得る。場合によっては、第2のB L T領域は、第1のB L T領域の偶数線を含み得、第3のB L T領域は、第1のB L T領域の奇数線を含み得る。

【0135】

[0141]いくつかの例では、非整列B L Tコマンドは、非整列B L Tコマンドに関連する第1の面の第1の面ポインタ値を指定し、第1の面ポインタ値は、複数の整列B L Tコマンドを実行するプロセッサに関連する所定の整列幅制約の整数倍ではなく、複数の整列B L Tコマンドは、第1の整列B L Tコマンドと第2の整列B L Tコマンドとを備える。そのような例では、複数の整列B L Tコマンドを生成するために、CPU6は、第2の面ポインタ値と第3の面ポインタ値とが所定の整列幅制約の異なる整数倍であるように第1の面ポインタ値と所定の整列幅制約とに基づいて第2の面ポインタ値と第3の面ポインタ値

10

20

30

40

50

とを決定することと、第1の整列B L Tコマンドが、第1の整列B L Tコマンドに関連する面のうちの1つの第2の面ポインタ値を指定するように第1の整列B L Tコマンドを生成することと、第2の整列B L Tコマンドが、第2の整列B L Tコマンドに関連する面のうちの1つの第3の面ポインタ値を指定するように第2の整列B L Tコマンドを生成することを行い得る。場合によっては、第2のB L T領域は、第1のB L T領域の偶数線を含み得、第3のB L T領域は、第1のB L T領域の奇数線を含み得る。

#### 【0136】

[0142]さらなる例では、非整列B L Tコマンドは、第1のB L T領域を指定する。そのような例では、C P U 6は、第1のB L T領域に基づいて第2のB L T領域と第3のB L T領域とを決定することと、第2のB L T領域が、第3のB L T領域とは異なる、第1の整列B L Tコマンドが、第2の面幅と、第2の面ポインタ値と、第2のB L T領域とを指定するように第1の整列B L Tコマンドを生成することと、第2の整列B L Tコマンドが、第2の面幅と、第3の面ポインタ値と、第3のB L T領域とを指定するように第2の整列B L Tコマンドを生成することを行い得る。場合によっては、第2のB L T領域は、第1のB L T領域の偶数線を含み得、第3のB L T領域は、第1のB L T領域の奇数線を含み得る。

#### 【0137】

[0143]本明細書で説明する技法は、仮想面の単一の線を形成するために3本以上の面線が一緒にグループ化される面に拡張され得る。たとえば、N本の面線が一緒にグループ化されるとき、単一の非整列B L Tコマンドは、N個の整列B L Tコマンドに変換され得、ここで、Nは、2以上の整数である。

#### 【0138】

[0144]3本以上の面線が一緒にグループ化される場合、非整列B L Tコマンドのための非整列面のうちの1つの面ポインタが非整列である場合、C P U 6は、(図9と同様の)2つの異なる仮想面を生成し得、ここで、第1の仮想面は、非整列面ポインタよりも小さい(less than)整列面ポインタを有し、第2の仮想面は、非整列面よりも大きい整列面ポインタを有する。C P U 6は、特定のB L Tコマンド中のアクセスである面線が割り当てられた仮想面の端をラップアラウンドしないことを保証する方式で2つの仮想面のうちの1つをN個のB L Tコマンドの各々に割り当て得る。

#### 【0139】

[0145]いくつかの例では、本開示の技法は、スケーリングおよび/またはフィルタ処理を実行するB L T動作をサービスするために使用され得る。そのような例では、読み取りエンジン38は、整列制約されないことがあるが、書き込みエンジン40は、整列制約され得る。

#### 【0140】

[0146]本開示に記載された技法は、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、またはそれらの任意の組合せで実装され得る。たとえば、説明する技法の様々な態様は、1つまたは複数のマイクロプロセッサ、デジタル信号プロセッサ(D S P)、特定用途向け集積回路(A S I C)、フィールドプログラマブルゲートアレイ(F P G A)、あるいは他の等価な集積回路またはディスクリート論理回路を含む、1つまたは複数のプロセッサ、ならびにそのような構成要素の任意の組合せ内で実装され得る。「プロセッサ」または「処理回路」という用語は、概して、単独で、あるいは他の論理回路、または処理を実施する個別ハードウェアなどの他の等価回路との組合せで上記の論理回路のいずれかを指すことがある。

#### 【0141】

[0147]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、本開示で説明した様々な動作および機能をサポートするために、同じデバイス内で、または別々のデバイス内で実装され得る。さらに、説明したユニット、モジュールまたは構成要素のいずれも、個別であるが相互運用可能な論理デバイスとして、一緒にまたは別々に実装され得る。モジュールまたはユニットとしての様々な機能の図は、様々な機能的態様を強調する

10

20

30

40

50

ものであり、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またはソフトウェア構成要素によって実現されなければならないことを必ずしも暗示するとは限らない。そうではなく、1つまたは複数のモジュールあるいはユニットに関連する機能は、別々のハードウェア、ファームウェア、および/またはソフトウェア構成要素によって実行されるか、あるいは共通または別々のハードウェアもしくはソフトウェア構成要素内に組み込まれることがある。

【0142】

[0148]また、本開示で説明した技法は、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中に記憶、実施または符号化され得る。コンピュータ可読媒体中に埋め込まれたまたは符号化された命令は、たとえば、その命令が1つまたは複数のプロセッサによって実行されたとき、1つまたは複数のプロセッサに本明細書で説明した技法を実行させ得る。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)、読み取り専用メモリ(ROM)、プログラマブル読み取り専用メモリ(PROM)、消去可能なプログラマブル読み取り専用メモリ(EEPROM)、電子的に消去可能なプログラマブル読み取り専用メモリ(EEPROM)、フラッシュメモリ、ハードディスク、CD-ROM、フロッピー(登録商標)ディスク、カセット、磁気媒体、光媒体、または有形である他のコンピュータ可読記憶媒体を含み得る。

【0143】

[0149]コンピュータ可読媒体は、上記に記載した有形記憶媒体などの有形記憶媒体に対応するコンピュータ可読記憶媒体を含み得る。コンピュータ可読媒体はまた、たとえば、通信プロトコルに従って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通信媒体を備え得る。この様式で、「コンピュータ可読媒体」という語句は、一般的に、(1)非一時的である有形のコンピュータ可読記憶媒体、および(2)一時的な信号または搬送波などの無形のコンピュータ可読通信媒体に対応し得る。

【0144】

[0150]様々な態様および例について説明した。しかしながら、以下の特許請求の範囲から逸脱することなく、本開示の構造または技法に変更が行われ得る。

以下に本願の出願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、非整列ブロック転送(BLT)コマンドを複数の整列BLTコマンドに変換すること  
を備える方法。

[C2]

前記非整列BLTコマンドが、前記非整列BLTコマンドに関連する面のうちの少なくとも1つのための第1の面幅を指定する、ここにおいて、前記非整列BLTコマンドを変換することが、

前記第1の面幅に基づいて第2の面幅を決定することと、ここにおいて、前記第2の面幅が前記第1の面幅のN倍である、ここで、Nは、2以上の整数である、

前記複数の整列BLTコマンドの各々が、前記複数の整列BLTコマンドの各々に関連する面のうちの少なくとも1つのための前記第2の面幅を指定するように前記複数の整列BLTコマンドを生成すること

を備える、C1に記載の方法。

[C3]

前記第2の面幅を決定することが、前記第1の面幅と、前記複数の整列BLTコマンドを実行するプロセッサに関連する所定の整列幅制約とに基づいて前記第2の面幅を決定することを備える、C2に記載の方法。

[C4]

前記第1の面幅が、前記所定の整列幅制約の整数倍でない、ここにおいて、前記第1の面幅と前記所定の整列幅制約とに基づいて前記第2の面幅を決定することは、前記第2の面幅が前記所定の整列幅制約の整数倍であるように前記第2の面幅を決定することを備え

10

20

40

50

る、C 3 に記載の方法。

[ C 5 ]

N が 2 に等しい、C 2 に記載の方法。

[ C 6 ]

前記複数の整列 B L T コマンドが、第 1 の整列 B L T コマンドと第 2 の整列 B L T コマンドとを備える、ここにおいて、前記非整列 B L T コマンドが、第 1 の B L T 領域を指定する、ここにおいて、前記複数の整列 B L T コマンドを生成することが、

前記第 1 の B L T 領域に基づいて第 2 の B L T 領域と第 3 の B L T 領域とを決定することと、前記第 2 の B L T 領域が、前記第 3 の B L T 領域とは異なる、

前記第 1 の整列 B L T コマンドが前記第 2 の B L T 領域を指定するように前記第 1 の整列 B L T コマンドを生成することと、

前記第 2 の整列 B L T コマンドが前記第 3 の B L T 領域を指定するように前記第 2 の整列 B L T コマンドを生成することと

を備える、C 2 に記載の方法。

[ C 7 ]

前記第 2 の B L T 領域が、前記第 1 の B L T 領域の偶数線を含み、前記第 3 の B L T 領域が、前記第 1 の B L T 領域の奇数線を含む、C 6 に記載の方法。

[ C 8 ]

前記非整列 B L T コマンドが、前記非整列 B L T コマンドに関連する第 1 の面の第 1 の面ポインタ値を指定する、ここにおいて、前記複数の整列 B L T コマンドを生成することが、

前記複数の整列 B L T コマンドの各々が、前記複数の整列 B L T コマンドの各々に関連する前記面のうちの 1 つの前記第 1 の面ポインタ値を指定するように前記複数の整列 B L T コマンドを生成すること

を備える、C 2 に記載の方法。

[ C 9 ]

前記複数の整列 B L T コマンドが、第 1 の整列 B L T コマンドと第 2 の整列 B L T コマンドとを備える、ここにおいて、前記非整列 B L T コマンドが、第 1 の B L T 領域を指定する、ここにおいて、前記複数の整列 B L T コマンドを生成することが、

前記第 1 の B L T 領域に基づいて第 2 の B L T 領域と第 3 の B L T 領域とを決定することと、前記第 2 の B L T 領域が、前記第 3 の B L T 領域とは異なる、

前記第 1 の整列 B L T コマンドが、前記第 2 の面幅と、前記第 1 の面ポインタ値と、前記第 2 の B L T 領域とを指定するように前記第 1 の整列 B L T コマンドを生成することと

、前記第 2 の整列 B L T コマンドが、前記第 2 の面幅と、前記第 1 の面ポインタ値と、前記第 3 の B L T 領域とを指定するように前記第 2 の整列 B L T コマンドを生成することとを備える、C 8 に記載の方法。

[ C 10 ]

前記第 2 の B L T 領域が、前記第 1 の B L T 領域の偶数線を含み、前記第 3 の B L T 領域が、前記第 1 の B L T 領域の奇数線を含む、C 9 に記載の方法。

[ C 11 ]

前記非整列 B L T コマンドが、前記非整列 B L T コマンドに関連する第 1 の面の第 1 の面ポインタ値を指定する、ここにおいて、前記第 1 の面ポインタ値が、前記複数の整列 B L T コマンドを実行するプロセッサに関連する所定の整列幅制約の整数倍ではない、ここにおいて、前記複数の整列 B L T コマンドが、第 1 の整列 B L T コマンドと第 2 の整列 B L T コマンドとを備える、ここにおいて、前記複数の整列 B L T コマンドを生成することが、

第 2 の面ポインタ値と第 3 の面ポインタ値とが前記所定の整列幅制約の異なる整数倍であるように前記第 1 の面ポインタ値と前記所定の整列幅制約とに基づいて前記第 2 の面ポインタ値と前記第 3 の面ポインタ値とを決定することと、

10

20

30

40

50

前記第1の整列B L Tコマンドが、前記第1の整列B L Tコマンドに関連する前記面のうちの1つの前記第2の面ポインタ値を指定するように前記第1の整列B L Tコマンドを生成することと、

前記第2の整列B L Tコマンドが、前記第2の整列B L Tコマンドに関連する前記面のうちの1つの前記第3の面ポインタ値を指定するように前記第2の整列B L Tコマンドを生成することと

を備える、C 2に記載の方法。

[ C 1 2 ]

前記第2の面ポインタ値が、前記第1の面ポインタ値よりも小さく、前記第3の面ポインタ値が、前記第1の面ポインタ値よりも大きい、C 1 1に記載の方法。

10

[ C 1 3 ]

前記非整列B L Tコマンドが、第1のB L T領域を指定する、

ここにおいて、前記複数の整列B L Tコマンドを生成することが、前記第1のB L T領域に基づいて第2のB L T領域と第3のB L T領域とを決定することを備える、前記第2のB L T領域が、前記第3のB L T領域とは異なる、

ここにおいて、前記第1の整列B L Tコマンドを生成することは、前記第1の整列B L Tコマンドが、前記第2の面幅と、前記第2の面ポインタ値と、前記第2のB L T領域とを指定するように前記第1の整列B L Tコマンドを生成することを備える、

ここにおいて、前記第2の整列B L Tコマンドを生成することは、前記第2の整列B L Tコマンドが、前記第2の面幅と、前記第3の面ポインタ値と、前記第3のB L T領域とを指定するように前記第2の整列B L Tコマンドを生成することを備える、C 1 1に記載の方法。

20

[ C 1 4 ]

前記第2のB L T領域が、前記第1のB L T領域の偶数線を含み、前記第3のB L T領域が、前記第1のB L T領域の奇数線を含む、C 1 3に記載の方法。

[ C 1 5 ]

非整列ブロック転送(B L T)コマンドを複数の整列B L Tコマンドに変換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るデバイス。

[ C 1 6 ]

前記非整列B L Tコマンドが、前記非整列B L Tコマンドに関連する面のうちの少なくとも1つのための第1の面幅を指定する、ここにおいて、前記1つまたは複数のプロセッサが、

30

前記第1の面幅に基づいて第2の面幅を決定することと、ここにおいて、前記第2の面幅が前記第1の面幅のN倍である、ここで、Nは、2以上の整数である、

前記複数の整列B L Tコマンドの各々が、前記複数の整列B L Tコマンドの各々に関連する面のうちの少なくとも1つのための前記第2の面幅を指定するように前記複数の整列B L Tコマンドを生成することと

を行うようにさらに構成された、C 1 5に記載のデバイス。

[ C 1 7 ]

前記1つまたは複数のプロセッサが、前記第1の面幅と、前記複数の整列B L Tコマンドを実行するプロセッサに関連する所定の整列幅制約とに基づいて前記第2の面幅を決定することを行うようにさらに構成された、C 1 6に記載のデバイス。

40

[ C 1 8 ]

前記第1の面幅が、前記所定の整列幅制約の整数倍でない、ここにおいて、前記1つまたは複数のプロセッサは、前記第2の面幅が前記所定の整列幅制約の整数倍であるように前記第2の面幅を決定することを行うようにさらに構成された、C 1 7に記載のデバイス。

。

[ C 1 9 ]

Nが2に等しい、C 1 6に記載のデバイス。

[ C 2 0 ]

50

前記複数の整列 B L T コマンドが、第 1 の整列 B L T コマンドと第 2 の整列 B L T コマンドとを備える、ここにおいて、前記非整列 B L T コマンドが、第 1 の B L T 領域を指定する、ここにおいて、前記 1 つまたは複数のプロセッサが、

前記第 1 の B L T 領域に基づいて第 2 の B L T 領域と第 3 の B L T 領域とを決定することと、前記第 2 の B L T 領域が、前記第 3 の B L T 領域とは異なる、

前記第 1 の整列 B L T コマンドが前記第 2 の B L T 領域を指定するように前記第 1 の整列 B L T コマンドを生成することと、

前記第 2 の整列 B L T コマンドが前記第 3 の B L T 領域を指定するように前記第 2 の整列 B L T コマンドを生成することと

を行うようにさらに構成された、C 1 6 に記載のデバイス。

10

[ C 2 1 ]

前記第 2 の B L T 領域が、前記第 1 の B L T 領域の偶数線を含み、前記第 3 の B L T 領域が、前記第 1 の B L T 領域の奇数線を含む、C 2 0 に記載のデバイス。

[ C 2 2 ]

前記非整列 B L T コマンドが、前記非整列 B L T コマンドに関連する第 1 の面の第 1 の面ポインタ値を指定する、ここにおいて、前記 1 つまたは複数のプロセッサが、

前記複数の整列 B L T コマンドの各々が、前記複数の整列 B L T コマンドの各々に関連する前記面のうちの 1 つの前記第 1 の面ポインタ値を指定するように前記複数の整列 B L T コマンドを生成すること

を行うようにさらに構成された、C 1 6 に記載のデバイス。

20

[ C 2 3 ]

前記複数の整列 B L T コマンドが、第 1 の整列 B L T コマンドと第 2 の整列 B L T コマンドとを備える、ここにおいて、前記非整列 B L T コマンドが、第 1 の B L T 領域を指定する、ここにおいて、前記 1 つまたは複数のプロセッサが、

前記第 1 の B L T 領域に基づいて第 2 の B L T 領域と第 3 の B L T 領域とを決定することと、前記第 2 の B L T 領域が、前記第 3 の B L T 領域とは異なる、

前記第 1 の整列 B L T コマンドが、前記第 2 の面幅と、前記第 1 の面ポインタ値と、前記第 2 の B L T 領域とを指定するように前記第 1 の整列 B L T コマンドを生成することと、

前記第 2 の整列 B L T コマンドが、前記第 2 の面幅と、前記第 1 の面ポインタ値と、前記第 3 の B L T 領域とを指定するように前記第 2 の整列 B L T コマンドを生成することと

を行うようにさらに構成された、C 2 2 に記載のデバイス。

30

[ C 2 4 ]

前記第 2 の B L T 領域が、前記第 1 の B L T 領域の偶数線を含み、前記第 3 の B L T 領域が、前記第 1 の B L T 領域の奇数線を含む、C 2 3 に記載のデバイス。

[ C 2 5 ]

前記非整列 B L T コマンドが、前記非整列 B L T コマンドに関連する第 1 の面の第 1 の面ポインタ値を指定する、ここにおいて、前記第 1 の面ポインタ値が、前記複数の整列 B L T コマンドを実行するプロセッサに関連する所定の整列幅制約の整数倍ではない、ここにおいて、前記複数の整列 B L T コマンドが、第 1 の整列 B L T コマンドと第 2 の整列 B L T コマンドとを備える、ここにおいて、前記 1 つまたは複数のプロセッサが、

第 2 の面ポインタ値と第 3 の面ポインタ値とが前記所定の整列幅制約の異なる整数倍であるように前記第 1 の面ポインタ値と前記所定の整列幅制約とに基づいて前記第 2 の面ポインタ値と前記第 3 の面ポインタ値とを決定することと、

前記第 1 の整列 B L T コマンドが、前記第 1 の整列 B L T コマンドに関連する前記面のうちの 1 つの前記第 2 の面ポインタ値を指定するように前記第 1 の整列 B L T コマンドを生成することと、

前記第 2 の整列 B L T コマンドが、前記第 2 の整列 B L T コマンドに関連する前記面のうちの 1 つの前記第 3 の面ポインタ値を指定するように前記第 2 の整列 B L T コマンドを生成することと

40

50

を行うようにさらに構成された、C 1 6 に記載のデバイス。

[ C 2 6 ]

前記第2の面ポインタ値が、前記第1の面ポインタ値よりも小さく、前記第3の面ポインタ値が、前記第1の面ポインタ値よりも大きい、C 2 5 に記載のデバイス。

[ C 2 7 ]

前記非整列 B L T コマンドが、第1の B L T 領域を指定する、ここにおいて、前記1つまたは複数のプロセッサが、

前記第1の B L T 領域に基づいて第2の B L T 領域と第3の B L T 領域とを決定することと、前記第2の B L T 領域が、前記第3の B L T 領域とは異なる、

前記第1の整列 B L T コマンドが、前記第2の面幅と、前記第2の面ポインタ値と、前記第2の B L T 領域とを指定するように前記第1の整列 B L T コマンドを生成することと、

前記第2の整列 B L T コマンドが、前記第2の面幅と、前記第3の面ポインタ値と、前記第3の B L T 領域とを指定するように前記第2の整列 B L T コマンドを生成することと、を行うようにさらに構成された、C 2 5 に記載のデバイス。

[ C 2 8 ]

前記デバイスが、ワイヤレス通信デバイスとモバイルフォンハンドセットとのうちの少なくとも1つを備える、C 1 5 に記載のデバイス。

[ C 2 9 ]

非整列ブロック転送( B L T )コマンドを受信するための手段と、

前記非整列 B L T コマンドを複数の整列 B L T コマンドに変換するための手段とを備える装置。

[ C 3 0 ]

実行されたとき、1つまたは複数のプロセッサに、

非整列ブロック転送( B L T )コマンドを複数の整列 B L T コマンドに変換するこ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。

【図1】

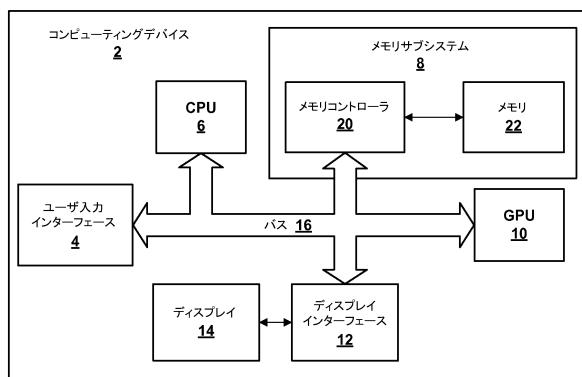


FIG. 1

【図2】

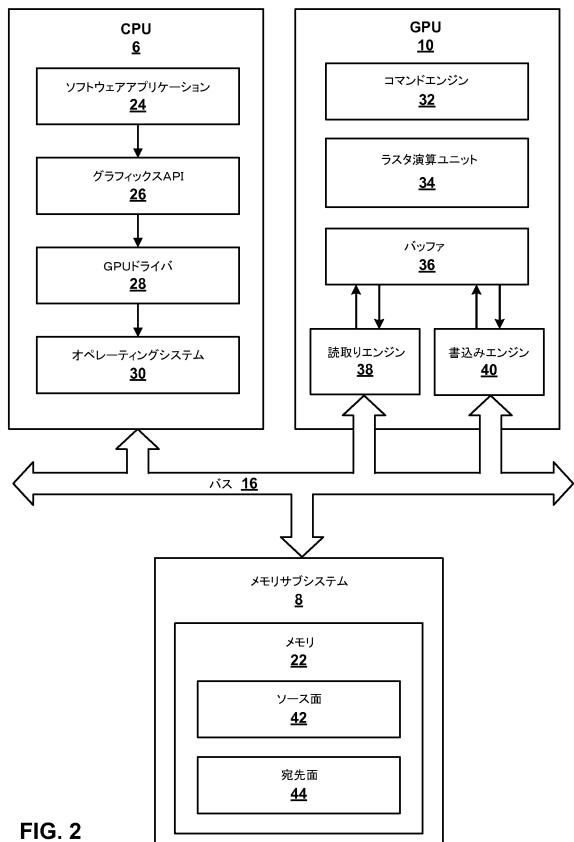


FIG. 2

【図3】

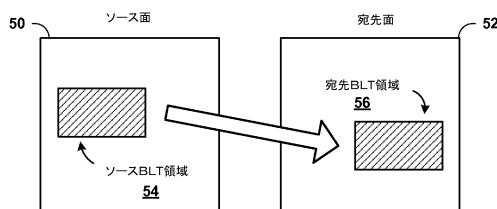


FIG. 3

【図4】

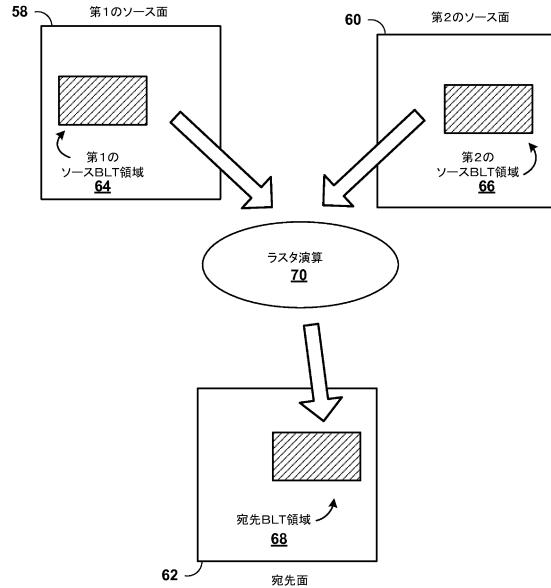


FIG. 4

【図5】

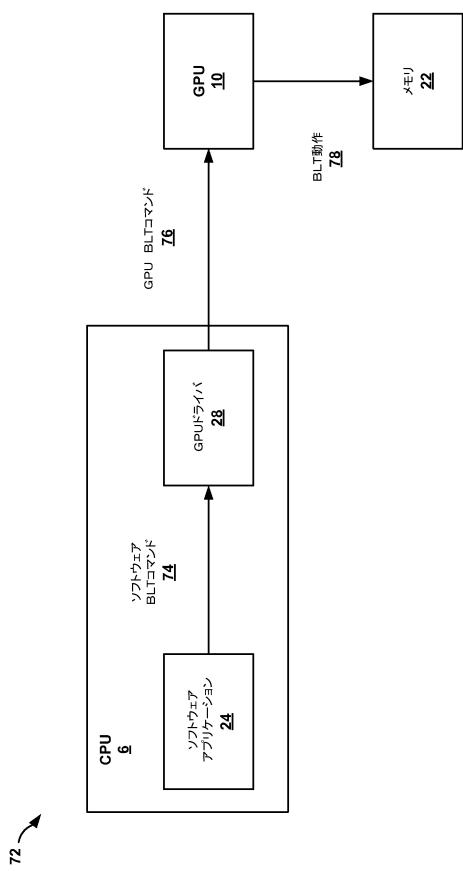


FIG. 5

【図6】

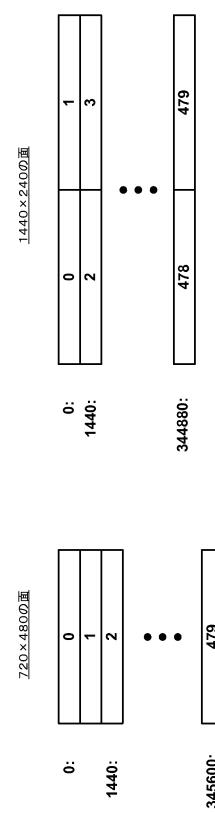


FIG. 6

【図7】

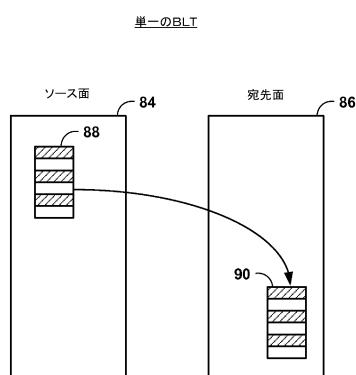


FIG. 7

【図8】

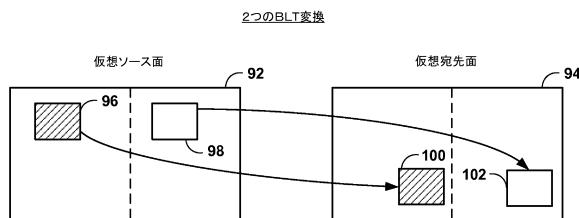


FIG. 8

【図9】

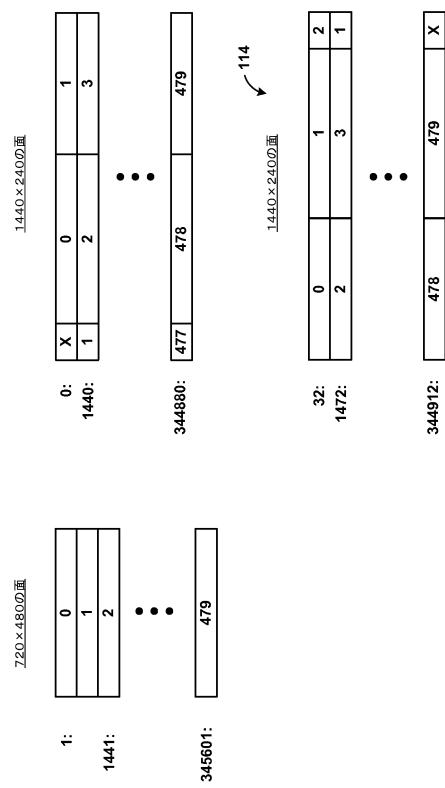


FIG. 9

【図10】

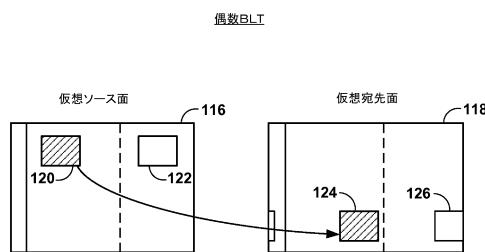


FIG. 10

【図11】

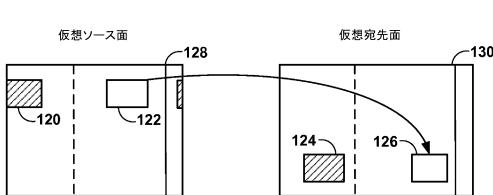


FIG. 11

【図12】

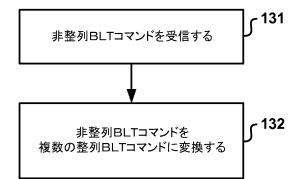


FIG. 12

---

フロントページの続き

(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

審査官 田上 隆一

(56)参考文献 米国特許出願公開第2014/0176586(US, A1)

特開2009-205366(JP, A)

国際公開第99/034273(WO, A2)

特表2007-501473(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 13/28