

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6271768号  
(P6271768)

(45) 発行日 平成30年1月31日(2018.1.31)

(24) 登録日 平成30年1月12日(2018.1.12)

(51) Int.Cl.

G06T 15/00 (2011.01)  
G06T 1/20 (2006.01)

F 1

G06T 15/00  
G06T 1/20501  
C

請求項の数 18 (全 30 頁)

(21) 出願番号 特願2016-569551 (P2016-569551)  
 (86) (22) 出願日 平成27年1月26日 (2015.1.26)  
 (65) 公表番号 特表2017-509092 (P2017-509092A)  
 (43) 公表日 平成29年3月30日 (2017.3.30)  
 (86) 國際出願番号 PCT/US2015/012917  
 (87) 國際公開番号 WO2015/126574  
 (87) 國際公開日 平成27年8月27日 (2015.8.27)  
 審査請求日 平成29年7月20日 (2017.7.20)  
 (31) 優先権主張番号 14/182,976  
 (32) 優先日 平成26年2月18日 (2014.2.18)  
 (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】共有されるデータチャネルを用いるシェーダパイプライン

(57) 【特許請求の範囲】

【請求項 1】

グラフィックス処理の方法であつて、  
 グラフィックス処理ユニット(GPU)が、前記GPUのオンチップグラフィックスメモリの中に第1の共有されるデータチャネルとして第1のリングバッファを割り当てることと、ここにおいて前記第1のリングバッファは、グラフィックス処理パイプラインの第1の2つのステージによって作り出される第1および第2のデータを記憶するために、前記グラフィックス処理パイプラインの前記それぞれの第1の2つのステージによって共有され、

前記GPUが、前記グラフィックス処理パイプラインの第2の2つのステージによって作り出される第3および第4のデータを記憶するために、前記グラフィックス処理パイプラインの前記それぞれの第2の2つのステージによって共有される前記GPUの前記オンチップグラフィックスメモリの中に第2の共有されるデータチャネルとして第2のリングバッファを割り当てることと、

前記GPUの中のシェーダユニット上で、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの1つが、前記第1のリングバッファ内に記憶される前記第1のデータを作り出すために、前記第2のリングバッファの中に記憶される前記第3のデータを消費することを含む、前記グラフィックス処理パイプラインの前記第1の2つのステージ、および前記グラフィックス処理パイプラインの前記第2の2つのステージを実行することと、

10

20

前記 G P U が、前記第 1 のリングバッファと前記第 2 のリングバッファ間のデッドロックを防ぐために、前記グラフィックス処理パイプラインの前記第 1 の 2 つのステージのうちの第 2 のものによって作り出される前記第 2 のデータを記憶するための前記第 1 のリングバッファの中のフリースペース、および前記グラフィックス処理パイプラインの前記第 2 の 2 つのステージのうちの 1 つによって作り出される前記第 4 のデータを記憶するための前記第 2 のリングバッファの中のフリースペースを保持することと、

オンチップグラフィックスメモリの中の前記第 1 のリングバッファ内に前記 G P U が、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの第 1 の 2 つのステージによって作り出される前記第 1 および第 2 のデータを、前記グラフィックス処理パイプラインの前記第 1 の 2 つのステージによって作り出される前記第 1 および第 2 のデータのキューとして記憶することと、

オンチップグラフィックスメモリの中の前記第 1 のリングバッファから前記 G P U が、前記第 1 のリングバッファから読み取られる前記グラフィックス処理パイプラインの前記第 1 の 2 つのステージのうちの前記第 2 のものによって作り出される前記第 2 のデータを前記リングバッファから削除することを含む、前記グラフィックス処理パイプラインの前記第 1 の 2 つのステージのうちの前記第 2 のものによって作り出される前記第 2 のデータを読み取ることと、それによって前記 G P U が前記グラフィックス処理パイプラインの前記第 1 の 2 つのステージのうちの前記 1 つによって作り出される追加のデータを記憶するための前記第 1 のリングバッファの中のスペースを増加させ、

を備える、方法。

10

#### 【請求項 2】

データが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの 1 つ以上のステージによって消費されるために前記第 1 のリングバッファ、または前記第 2 のリングバッファにおいて利用可能であり、ならびにフリースペースが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記 1 つ以上のステージによって作り出されるデータを記憶するために前記第 1 のリングバッファ、または前記第 2 のリングバッファにおいて利用可能であるように、前記 G P U が、前記第 1 のリングバッファ、または前記第 2 のリングバッファのステータスに少なくとも部分的に基づいて前記シェーダユニット上で前記グラフィックス処理パイプラインの前記 1 つ以上のステージの前記実行をスケジューリングすることをさらに備える、請求項 1 に記載の方法。

20

#### 【請求項 3】

前記第 2 のリングバッファは、前記第 2 のリングバッファの中に記憶されたデータをキャッシュするためにキャッシュモードで動作し、前記第 1 のリングバッファは、ファーストインファーストアウト ( F I F O ) モードで動作する、請求項 1 に記載の方法。

30

#### 【請求項 4】

前記第 1 のリングバッファの中に記憶された前記第 1 および第 2 のデータは、前記グラフィックス処理パイプラインの前記第 1 の 2 つのステージによって作り出される頂点データを備え、前記第 2 のリングバッファの中に記憶された前記第 3 および第 4 のデータは、前記グラフィックス処理パイプラインの前記第 2 の 2 つのステージによって作り出されるプリミティブを備える、請求項 1 に記載の方法。

40

#### 【請求項 5】

前記グラフィックス処理パイプラインの前記第 1 の 2 つのステージは、頂点シェーダおよびドメインシェーダを備える、請求項 4 に記載の方法。

#### 【請求項 6】

前記グラフィックス処理パイプラインの前記第 2 の 2 つのステージは、ハルシェーダおよびジオメトリシェーダを備える、請求項 4 に記載の方法。

#### 【請求項 7】

グラフィックスデータを処理するように構成された装置であって、  
メモリと、

グラフィックス処理ユニット ( G P U ) のオンチップグラフィックスメモリの中に第

50

1の共有されるデータチャネルとして第1のリングバッファを割り当てることと、ここにおいて前記第1のリングバッファは、グラフィックス処理パイプラインの第1の2つのステージによって作り出される第1および第2のデータを記憶するために、前記グラフィックス処理パイプラインの前記それぞれの第1の2つのステージによって共有され、

前記グラフィックス処理パイプラインの第2の2つのステージによって作り出される第3および第4のデータを記憶するために、前記グラフィックス処理パイプラインの前記それぞれの第2の2つのステージによって共有される前記GPUの前記オンチップグラフィックスメモリの中に第2の共有されるデータチャネルとして第2のリングバッファを割り当てることと、

前記GPUの中のシェーダユニット上で、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの1つが、前記第1のリングバッファ内に記憶される前記第1のデータを作り出すために、前記第2のリングバッファの中に記憶される前記第3のデータを消費することを含む、前記グラフィックス処理パイプラインの前記第1の2つのステージ、および前記グラフィックス処理パイプラインの前記第2の2つのステージを実行することと、 10

前記第1のリングバッファと前記第2のリングバッファ間のデッドロックを防ぐために、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの第2のものによって作り出される前記第2のデータを記憶するための前記第1のリングバッファの中のフリースペース、および前記グラフィックス処理パイプラインの前記第2の2つのステージのうちの1つによって作り出される前記第4のデータを記憶するための前記第2のリングバッファの中のフリースペースを保持することと、 20

オンチップグラフィックスメモリの中の前記第1のリングバッファ内に、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される前記第1および第2のデータを、前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される前記第1および第2のデータのキューとして記憶することと、

前記第1のリングバッファから読み取られる前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの前記第2のものによって作り出される前記第2のデータを前記リングバッファから削除することを含む、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの前記第2のものによって作り出される前記第2のデータを、オンチップグラフィックスメモリの中の前記第1のリングバッファから読み取ることと、それによって前記GPUが前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの前記1つによって作り出される追加のデータを記憶するための前記第1のリングバッファの中のスペースを増加させ、 30

を行うように構成される前記GPUと、

を備える、装置。

#### 【請求項8】

前記GPUは、

データが、前記グラフィックス処理パイプラインの1つ以上のステージによって消費されるために前記第1のリングバッファ、または前記第2のリングバッファにおいて利用可能であり、ならびにフリースペースが、前記シェーダプロセッサ上で実行している前記グラフィックス処理パイプラインの前記1つ以上のステージによって作り出されるデータを記憶するために前記第1のリングバッファ、または前記第2のリングバッファにおいて利用可能であるように、前記第1のリングバッファ、または前記第2のリングバッファのステータスに少なくとも部分的に基づいて前記シェーダプロセッサ上で前記グラフィックス処理パイプラインの前記1つ以上のステージの前記実行をスケジューリングするようにさらに構成される、請求項7に記載の装置。 40

#### 【請求項9】

前記第2のリングバッファは、前記第2のリングバッファの中に記憶されたデータをキャッシュするためにキャッシュモードで動作し、前記第1のリングバッファは、ファース 50

トインファーストアウト( FIFO )モードで動作する、請求項7に記載の装置。

【請求項 10】

前記第1のリングバッファの中に記憶された前記データは、前記シェーダプロセッサ上で実行している前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される頂点データを備え、前記第2のリングバッファの中に記憶された前記第3および第4のデータは、前記シェーダプロセッサ上で実行している前記グラフィックス処理パイプラインの前記第2の2つのステージによって作り出されるプリミティブを備える、請求項7に記載の装置。

【請求項 11】

装置であって、

グラフィックス処理ユニット( GPU )のオンチップグラフィックスメモリの中に第1の共有されるデータチャネルとして第1のリングバッファを割り当てるための手段と、ここにおいて前記第1のリングバッファは、グラフィックス処理パイプラインの第1の2つのステージによって作り出される第1および第2のデータを記憶するために、前記グラフィックス処理パイプラインの前記それぞれの第1の2つのステージによって共有され、

前記グラフィックス処理パイプラインの第2の2つのステージによって作り出される第3および第4のデータを記憶するために、前記グラフィックス処理パイプラインの前記それぞれの第2の2つのステージによって共有される前記 GPU の前記オンチップグラフィックスメモリの中に第2の共有されるデータチャネルとして第2のリングバッファを割り当てるための手段と、

前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの1つが、前記第1のリングバッファ内に記憶される前記第1のデータを作り出すために、前記第2のリングバッファの中に記憶される前記第3のデータを消費することを含む、前記グラフィックス処理パイプラインの前記第1の2つのステージ、および前記グラフィックス処理パイプラインの前記第2の2つのステージを実行するための手段と、

前記第1のリングバッファと前記第2のリングバッファ間のデッドロックを防ぐために、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの第2のものによって作り出される前記第2のデータを記憶するための前記第1のリングバッファの中のフリースペース、および前記グラフィックス処理パイプラインの前記第2の2つのステージのうちの1つによって作り出される前記第4のデータを記憶するための前記第2のリングバッファの中のフリースペースを保持するための手段と、

オンチップグラフィックスメモリの中の前記第1のリングバッファ内に、シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される前記第1および第2のデータを、前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される前記第1および第2のデータのキーとして記憶するための手段と、

前記第1のリングバッファから読み取られる前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの前記第2のものによって作り出される前記第2のデータを前記リングバッファから削除することを含む、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの前記第2のものによって作り出される前記第2のデータを、オンチップグラフィックスメモリの中の前記第1のリングバッファから GPU が読み取るための手段と、それによって前記 GPU が前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの前記1つによって作り出される追加のデータを記憶するための前記第1のリングバッファの中のスペースを増加させ、

を備える、装置。

【請求項 12】

データが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの1つ以上のステージによって消費するために前記第1のリングバッファ、または前記第2のリングバッファにおいて利用可能であり、ならびにフリースペースが、前記グラフィックス処理パイプラインの前記1つ以上のステージによって作り出されるデータを記

10

20

30

40

50

憶するために前記第1のリングバッファ、または前記第2のリングバッファにおいて利用可能であるように、前記第1のリングバッファ、または前記第2のリングバッファのステータスに少なくとも部分的に基づいて前記シェーダユニット上で前記グラフィックス処理パイプラインの前記1つ以上のステージの前記実行をスケジューリングする手段をさらに備える、請求項1\_1に記載の装置。

## 【請求項1\_3】

前記第2のリングバッファは、前記第2のリングバッファの中に記憶されたデータをキャッシュするためにキャッシュモードで動作し、前記第1のリングバッファは、ファーストインファーストアウト(FIFO)モードで動作する、請求項1\_1に記載の装置。

## 【請求項1\_4】

前記第1のリングバッファの中に記憶された前記第1および第2のデータは、前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される頂点データを備え、前記第2のリングバッファの中に記憶された前記第3および第4のデータは、前記グラフィックス処理パイプラインの前記第2の2つのステージによって作り出されるプリミティブを備える、請求項1\_1に記載の装置。

## 【請求項1\_5】

命令を記憶する非一時的コンピュータ可読記憶媒体であって、前記命令は、実行されるとき、1つ以上のプログラマブルプロセッサに、

オンチップグラフィックスメモリの中に第1の共有されるデータチャネルとして第1のリングバッファを割り当てることと、ここにおいて前記第1のリングバッファは、グラフィックス処理パイプラインの第1の2つのステージによって作り出される第1および第2のデータを記憶するために、前記グラフィックス処理パイプラインの前記それぞれの第1の2つのステージによって共有され、

前記グラフィックス処理パイプラインの第2の2つのステージによって作り出される第3および第4のデータを記憶するために、前記グラフィックス処理パイプラインの前記それぞれの第2の2つのステージによって共有される前記オンチップグラフィックスメモリの中に第2の共有されるデータチャネルとして第2のリングバッファを割り当てることと、

シェーダユニット上で、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの1つが、前記第1のリングバッファ内に記憶される前記第1のデータを作り出すために、前記第2のリングバッファの中に記憶される前記第3のデータを消費することを含む、前記グラフィックス処理パイプラインの前記第1の2つのステージ、および前記グラフィックス処理パイプラインの前記第2の2つのステージを実行することと、

前記第1のリングバッファと前記第2のリングバッファ間のデッドロックを防ぐために、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの第2のものによって作り出される前記第2のデータを記憶するための前記第1のリングバッファの中のフリースペース、および前記グラフィックス処理パイプラインの前記第2の2つのステージのうちの1つによって作り出される前記第4のデータを記憶するための前記第2のリングバッファの中のフリースペースを保持することと、

オンチップグラフィックスメモリの中の前記リングバッファ内に、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される前記第1および第2のデータを、前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される前記第1および第2のデータのキーとして記憶することと、

前記第1のリングバッファから読み取られる前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの前記第2のものによって作り出される前記第2のデータを前記リングバッファから削除することを含む、前記グラフィックス処理パイプラインの前記第1の2つのステージのうちの前記第2のものによって作り出される前記第2のデータを、オンチップグラフィックスメモリの中の前記第1のリングバッファから読み取ることと、それによって前記1つ以上のプログラマブルプロセッサが前記グラフィックス処理

10

20

30

40

50

パイプラインの前記第1の2つのステージのうちの前記1つによって作り出される追加のデータを記憶するための前記第1のリングバッファの中のスペースを増加させ、を行わせる、非一時的コンピュータ可読記憶媒体。

【請求項16】

前記命令は、実行されるとき、前記1つ以上のプログラマブルプロセッサに、データが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの1つ以上のステージによって消費されるために前記第1のリングバッファ、または前記第2のリングバッファにおいて利用可能であり、ならびにフリースペースが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記1つ以上のステージによって作り出されるデータを記憶するために前記第1のリングバッファ、または前記第2のリングバッファにおいて利用可能であるように、前記第1のリングバッファ、または前記第2のリングバッファのステータスに少なくとも部分的に基づいて前記シェーダユニット上で前記グラフィックス処理パイプラインの前記1つ以上のステージの前記実行をスケジューリングすることをさらに行わせる、請求項15に記載の非一時的コンピュータ可読記憶媒体。  
10

【請求項17】

前記第2のリングバッファは、前記第2のリングバッファの中に記憶されたデータをキャッシュするためにキャッシュモードで動作し、前記第1のリングバッファは、ファーストインファーストアウト(FIFO)モードで動作する、請求項15に記載の非一時的コンピュータ可読記憶媒体。  
20

【請求項18】

前記第1のリングバッファの中に記憶された前記第1および第2のデータは、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第1の2つのステージによって作り出される頂点データを備え、前記第2のリングバッファの中に記憶された前記第3および第4のデータは、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第2の2つのステージによって作り出されるプリミティブを備える、請求項15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】

【0001】

[0001] この開示は、グラフィックス処理パイプライン(graphics processing pipeline)のステージ(stages)によって作り出され、消費されるデータを記憶するための共有されるデータチャネル(shared data channels)に関する。  
30

【背景技術】

【0002】

[0002] コンピューティングデバイスのグラフィックス処理ユニット(GPU)は、3次元シーンの2次元表現(two-dimensional representation of a three-dimensional scene)をレンダリングするためのグラフィックスコマンドを処理するための複数のステージを含むグラフィックス処理パイプラインを実行することができる。3次元シーンは、複数の頂点で典型的に構成され、グラフィックス処理パイプラインは、3次元シーンの中の各頂点のために、3次元シーンの2次元表現をレンダリングするための固定の順序で実行される一連のステージを含む。  
40

【0003】

[0003] グラフィックス処理パイプラインは、3次元シーンの頂点を変換する(transform)ために実行するシェーダステージのチェーン(chain of shader stages)を含み得る。シェーダステージの各々は、前のステージによって作り出されたデータを消費し、次のステージのためにデータを作り出す。シェーダステージのチェーンを通して流れる膨大なデータ量のため、どのようにしてシェーダステージのチェーンのためのデータが管理されるかは、GPUのパフォーマンスおよびメモリの効率性に影響することができる。

【発明の概要】

50

## 【0004】

[0004] 本開示の一例では、グラフィックス処理のための方法は、グラフィックス処理ユニット (GPU) が、グラフィックス処理パイプラインの少なくとも 2 つのステージによって共有される GPU のオンチップグラフィックスメモリの中に共有されるデータチャネルを割り当てる (allocating) を含み得る。方法は、GPU の中のシェーダユニット上で、グラフィックス処理パイプラインの少なくとも 2 つのステージを実行することをさらに含み得る。方法は、オンチップグラフィックスメモリの中の共有されるデータチャネルの中に GPU が、シェーダユニット上で実行しているグラフィックス処理パイプラインの少なくとも 2 つのステージの各々によって作り出されるデータを記憶することをさらに含み得る。

10

## 【0005】

[0005] 本開示の別の例では、グラフィックス処理のための装置は、グラフィックス処理パイプラインの少なくとも 2 つのステージによって共有されるグラフィックス処理ユニット (GPU) のオンチップグラフィックスメモリの中に共有されるデータチャネルを割り当てる (allocating) ことと、GPU の中のシェーダユニット上で、グラフィックス処理パイプラインの少なくとも 2 つのステージを実行することと、オンチップグラフィックスメモリの中の共有されるデータチャネルの中に、シェーダユニット上で実行しているグラフィックス処理パイプラインの少なくとも 2 つのステージの各々によって作り出されるデータを記憶することとを行うように構成される GPU を含み得る。

20

## 【0006】

[0006] 本開示の別の例では、グラフィックス処理のためのグラフィックスのための装置は、グラフィックス処理パイプラインの少なくとも 2 つのステージによって共有されるグラフィックス処理ユニット (GPU) のオンチップグラフィックスメモリの中に共有されるデータチャネルを割り当てるための手段を含み得る。装置は、グラフィックス処理パイプラインの少なくとも 2 つのステージを実行するための手段をさらに含み得る。装置は、オンチップグラフィックスメモリの中の共有されるデータチャネルの中に、グラフィックス処理パイプラインの少なくとも 2 つのステージの各々の実行によって作り出されるデータを記憶するための手段をさらに含み得る。

30

## 【0007】

[0007] 本開示の別の例では、コンピュータ可読記憶媒体は、命令を記憶することができ、命令は、実行されるとき、1 つ以上のプログラマブルプロセッサに、共有されるデータチャネルを、グラフィックス処理パイプラインの少なくとも 2 つのステージによって共有されるそのオンチップグラフィックスメモリの中に割り当てる (allocating) ことと、シェーダユニット上で、グラフィックス処理パイプラインの少なくとも 2 つのステージを実行することと、オンチップグラフィックスメモリの中の共有されるデータチャネルの中に、シェーダユニット上で実行しているグラフィックス処理パイプラインの少なくとも 2 つのステージの各々によって作り出されるデータを記憶することとを行わせる。

## 【0008】

[0008] 1 つ以上の例の詳細は、添付の図面および以下の説明の中で記述される。他の特徴、目的、および利点は、説明および図面から、ならびに特許請求の範囲から明らかになるであろう。

40

## 【図面の簡単な説明】

## 【0009】

【図 1】[0009] 図 1 は、複数のグラフィックスパイプラインステージの間でデータチャネルを共有するための本開示の 1 つ以上の態様をインプリメントするように構成され得る例示的なコンピューティングデバイスを例示するブロック図である。

【図 2】[0010] 図 2 は、3 次元シーンの 2 次元表現を作るために GPU によって行われ得る例示的なグラフィックス処理パイプラインを例示するブロック図である。

【図 3】[0011] 図 3 は、さらに詳細に図 1 の CPU、GPU、およびシステムメモリの例示的なインプリメンテーションを例示するブロック図である。

50

【図4】[0012] 図4は、グラフィックス処理パイプラインの中で使用されている共有されるデータチャネルの例を例示するブロック図である。

【図5】[0013] 図5は、さらに詳細に図4のキャッシュモード共有チャネル(cache mode shared channel)を例示するブロック図である。

【図6】[0014] 図6は、グラフィックス処理パイプラインのステージによってデータチャネルを共有するための例示的なプロセスを例示する流れ図である。

【発明を実施するための形態】

【0010】

[0015] 一般に、本開示は、共有されるデータチャネルを持つ生産者 - 消費者モデル(producer-consumer model)を使用してシングルパスシェーダパイプライン(single pass shader pipeline)に関する技法を説明する。コンピューティングデバイスの中のグラフィカル処理ユニット(GPU)は、GPU上で同時にシェーダパイプラインの複数のステージを実行し得るシェーダユニット上でシェーダパイプラインを実行することができる。GPUの中のオンチップメモリの中に記憶されるデータは典型的に、コンピューティングデバイスのシステムメモリの中に記憶されるデータより速く、より効率的にアクセスすることができるため、GPUの中のシェーダユニットの効率性は、GPUの中のオンチップメモリの中のデータチャネルからデータを消費することによって、およびGPUの中のオンチップメモリの中のデータチャネルの中に記憶もされるデータを作り出すことによって増加させることができる。

【0011】

[0016] 一例では、GPUは、GPUの中のシェーダユニットによるシェーダパイプラインの実行によって消費され、作り出されるデータを記憶するために等しいサイズのデータチャネルを割り当て得る。しかしながら、GPUの中のオンチップメモリが典型的に、コンピューティングデバイスのシステムメモリより非常に少ない記憶スペースを含むため、GPUの中のオンチップメモリは、GPUの中のシェーダユニットによって消費され、作り出されるデータのすべてのために別個のデータチャネルを割り当てるのに十分な記憶スペースを有しないことがある。さらに、シェーダパイプラインのステージが、シェーダパイプラインのいくつかのステージがシェーダパイプラインの他のステージより多くのデータを作り出す傾向があるような、アンバランスであり得るため、シェーダパイプラインの各ステージによって作り出されるデータのためにオンチップメモリの中に等しいスペースを割り当てるることは、オンチップメモリの中の記憶スペースを浪費し得る。加えて、オンチップメモリは、シェーダパイプラインのステージによって作り出されるデータの少なくともいくつかがより遅いシステムメモリの中に記憶される必要があり得るような、シェーダパイプラインの各ステージによって作り出されるデータのために等しいスペースを割り当てるために十分な記憶スペースを有しないことがあり、それによってGPUのパフォーマンスを低減させる。

【0012】

[0017] 本開示の態様にしたがって、GPUは、シェーダパイプラインの2つ以上のステージが单一の共有されるデータチャネルを共有することができるよう、共有されるデータチャネルを共有しているシェーダパイプラインの第1のステージからのデータを記憶するために使用されていない共有されるデータチャネルの中のスペースがデータチャネルを共有しているシェーダパイプラインの第2のステージからデータを記憶するために使用され得るような、GPUの中のオンチップメモリの中の共有されるデータチャネルを割り当てる。このようにして、GPUの中のオンチップメモリは、より効率的な方法で利用され得る。さらに、他のアプローチに比べて、シェーダパイプラインによって作り出されるより多くのデータを潜在的に(potentially)記憶するために、より効率的な方法でGPUの中のオンチップメモリを利用することによって、GPUの中のオンチップメモリは、シェーダパイプラインのステージを実行しているシェーダユニットによって消費される準備ができているより多くのデータを記憶することができ、それによってシェーダユニットの利用を増大させ、GPUのパフォーマンスを増大させる。

10

20

30

40

50

## 【0013】

[0018] 図1は、複数のグラフィックスパイプラインステージの間でデータチャネルを共有するための本開示の1つ以上の態様をインプリメントするように構成され得る例示的なコンピューティングデバイスを例示するブロック図である。図1に示されるように、デバイス2は、ビデオデバイス、メディアプレーヤ、セットトップボックス、モバイル電話およびいわゆるスマートフォンのようなワイヤレスハンドセット、携帯情報端末(PDA)、デスクトップコンピュータ、ラップトップコンピュータ、ゲーミングコンソール、ビデオ会議ユニット、タブレットコンピューティングデバイス等に限られないが、それらを含むコンピューティングデバイスであり得る。図1の例では、デバイス2は、中央処理ユニット(CPU)6、システムメモリ10、およびGPU12を含み得る。デバイス2はまた、ディスプレイプロセッサ14、トランシーバモジュール3、ユーザインタフェース4、およびディスプレイ8を含み得る。トランシーバモジュール3およびディスプレイプロセッサ14は両方ともに、CPU6および/またはGPU12と同じ集積回路(IGC)の一部であることができ、両方ともにCPU6および/またはGPU12を含む単一のIGCまたは複数のIGCsの外部に存在することができ、またはCPU6および/またはGPU12を含むIGCの外部に存在するIGCの中に形成され得る。

10

## 【0014】

[0019] デバイス2は、明確さの目的で図1に示されていない追加のモジュールまたはユニットを含み得る。例えば、デバイス2は、どちらも図1に示されていない、スピーカおよびマイクロфонを、デバイス2がモバイルワイヤレス電話である例において、またはデバイス2がメディアプレーヤであるスピーカにおいて電話通信を達成するために含み得る。デバイス2はまた、ビデオカメラを含み得る。さらに、デバイス2に示される様々なモジュールおよびユニットは、デバイス2のすべての例において必要とは限らないことがある。例えば、ユーザインタフェース4およびディスプレイ8は、デバイス2がデスクトップコンピュータ、あるいは外部のユーザインタフェースまたはディスプレイとインターフェースをとるために装備される他のデバイスである例におけるデバイス2の外部に存在し得る。

20

## 【0015】

[0020] ユーザインタフェース4の例は、トラックボール、マウス、キーボード、および他のタイプの入力デバイスに限定されないが、それらを含む。ユーザインタフェース4はまた、タッチスクリーンであることもでき、ディスプレイ8の一部として組み込まれ得る。トランシーバモジュール3は、コンピューティングデバイス2と別のデバイスまたはネットワーク間でワイヤレスまたはワイヤード通信を可能にするための回路を含み得る。トランシーバモジュール3は、変調器、復調器、増幅器、およびワイヤードまたはワイヤレス通信のための他のそのような回路を含み得る。

30

## 【0016】

[0021] CPU6は、実行のためのコンピュータプログラムの命令を処理するように構成される中央処理ユニット(CPU)のような、マイクロプロセッサであり得る。CPU6は、コンピューティングデバイス2のオペレーションを制御する汎用、または専用プロセッサを備え得る。ユーザは、CPU6に1つ以上のソフトウェアアプリケーションを実行するために入力をコンピューティングデバイス2に提供し得る。CPU6上で実行するソフトウェアアプリケーションは、例えば、オペレーティングシステム、ワードプロセッサアプリケーション、電子メールアプリケーション、スプレッドシートアプリケーション、メディアプレーヤアプリケーション、ビデオゲームアプリケーション、グラフィカルユーザインタフェースアプリケーションまたは別のプログラムを含み得る。加えて、CPU6は、GPU12のオペレーションを制御するためにGPUドライバ22を実行し得る。ユーザは、キーボード、マウス、マイクロфон、タッチパッドまたはユーザインタフェース4を介してコンピューティングデバイス2に結合される別の入力デバイスのような1つ以上の入力デバイス(示されていない)を介して入力をコンピューティングデバイス2に提供し得る。

40

50

## 【0017】

[0022] 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 8 へのグラフィックスデータのレンダリングをもたらすために C P U 6 に命令する 1 つ以上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、例えば、オープングラフィックスライブラリ ( O p e n G L ( 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 ( A P I ) 、オープングラフィックスライブラリ埋め込みシステム ( Open Graphics Library Embedded Systems ) ( O p e n G L E S ) A P I 、 D i r e c t 3 D A P I 、 X 3 D A P I 、 R e n d e r M a n A P I 、 W e b G L A P I 、または任意の他の公的または独自の基準グラフィックス A P I ( public or proprietary standard graphics API ) のようなグラフィックス A P I にしたがい得る。  
10 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、 G P U 1 2 にグラフィックスデータのレンダリングのいくつか、またはすべてを行わせるために、( 例えば、 G P U ドライバ 2 2 を通して ) 1 つ以上のグラフィックスレンダリングコマンドを G P U 1 2 に発行し ( issue ) 得る。いくつかの例では、レンダリングされるためのグラフィックスデータは、例えば、点、線、三角形、四角形、トライアングルストリップ ( triangles trips ) などのグラフィックスプリミティブ ( graphics primitives ) のリストを含み得る。  
。

## 【0018】

[0023] G P U 1 2 は、ディスプレイ 8 への 1 つ以上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オペレーションを行うように構成され得る。したがって、 C P U 6 上で実行しているソフトウェアアプリケーションのうちの 1 つがグラフィックス処理を要求するとき、 C P U 6 は、ディスプレイ 8 へのレンダリングのためにグラフィックスコマンドおよびグラフィックスデータを G P U 1 2 に提供し得る。グラフィックスデータは、例えば、描画コマンド ( drawing commands ) 、ステート情報、プリミティブ情報、テクスチャ情報等を含み得る。 G P U 1 2 は、いくつかの例では、複雑なグラフィック関連オペレーションの C P U 6 より効率的な処理を提供する高度に並列な構造 ( highly-parallel structure ) で構築され得る。例えば、 G P U 1 2 は、シェーダユニットのような、複数の処理要素を含むことができ、それらは、並行して ( in a parallel manner ) 複数の頂点またはピクセル上で動作するように構成される。 G P U 1 2 の高度に並列な性質は、いくつかの例では、 G P U 1 2 が、 C P U 6 を使用してディスプレイ 8 に直接シーンを描写するより速くディスプレイ 8 上にグラフィックスイメージ ( 例えば、 G U I s および 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 ) を描写することを可能し得る。  
20  
30

## 【0019】

[0024] G P U 1 2 は、いくつかの例では、コンピューティングデバイス 2 のマザーボードに組み込まれ得る。他の例では、 G P U 1 2 は、コンピューティングデバイス 2 のマザーボードの中の一部にインストールされるグラフィックスカード上に存在し得る、または、そうでなければコンピューティングデバイス 2 と相互運用するように構成される周辺デバイス内に組み込まれ得る。 G P U 1 2 は、 1 つ以上のマイクロプロセッサ、特定用途向け集積回路 ( A S I C s ) 、フィールドプログラマブルゲートアレイ ( F P G A s ) 、デジタルシグナルプロセッサ ( D S P s ) 、または他の同等の集積回路またはディスクリート論理回路のような、 1 つ以上のプロセッサを含み得る。 G P U 1 2 はまた、 G P U 1 2 がマルチコアプロセッサと称され得るよう、 1 つ以上のプロセッサコアを含み得る。  
40

## 【0020】

[0025] G P U 1 2 は、グラフィックスメモリ 4 0 に直接結合され得る。したがって、 G P U 1 2 は、バスを使用することなしにグラフィックスメモリ 4 0 からデータを読み取り、グラフィックスメモリ 4 0 にデータを書き込み得る。言い換えれば、 G P U 1 2 は、オフチップメモリの代わりに、ローカル記憶装置をローカルに使用してデータを処理し得る。そのようなグラフィックスメモリ 4 0 は、オンチップメモリと称され得る。このことは、バスを介してデータを読み取り、データを書き込むための G P U 1 2 の必要性を取り  
50

除くことによって、G P U 1 2 がより効率的な方法で動作することを可能にし、それは、重いバストラヒックを経験し得る。しかしながら、いくつかの例では、G P U 1 2 は、別個のメモリを含み得ないが、バスを介してシステムメモリ 1 0 を代わりに利用し得る。グラフィックスメモリ 4 0 は、例えば、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) 、消去可能なプログラマブル R O M (E P R O M) 、電気的に消去可能なプログラマブル 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のような、1つ以上の揮発性、または不揮発性メモリ、または記憶デバイスを含み得る。

#### 【 0 0 2 1 】

[0026] いくつかの例では、G P U 1 2 は、システムメモリ 1 0 の中に十分に形成された画像を記憶し得る。ディスプレイプロセッサ 1 4 は、システムメモリ 1 0 から画像を取り出すことができ、ディスプレイ 8 のピクセルに画像をディスプレイするために明るくさせる値を出力し得る。ディスプレイ 8 は、G P U 1 2 によって生成される画像コンテンツをディスプレイするデバイス 2 のディスプレイであり得る。ディスプレイ 8 は、液晶ディスプレイ (L C D) 、有機発光ダイオードディスプレイ (O L E D) 、陰極線管 (C R T) ディスプレイ、プラズマディスプレイ、または別のタイプのディスプレイデバイスであり得る。

#### 【 0 0 2 2 】

[0027] 本開示の態様にしたがって、G P U 1 2 は、そのシェーダユニット上で複数のグラフィックスパイプラインステージを実行し得る。G P U 1 2 は、シェーダユニット上で実行している複数のグラフィックスパイプラインステージの2つのステージによって共有されるグラフィックスメモリ 4 0 の中で共有されるデータチャネルを作り得る。G P U 1 2 は、共有されるデータチャネルの中のシェーダユニット上で実行している複数のグラフィックスパイプラインステージの2つのステージの各々によって作り出されるデータを記憶し得る。

#### 【 0 0 2 3 】

[0028] 図 2 は、3次元シーンの2次元表現を作るために G P U 1 2 によって行われ得る例示的なグラフィックス処理パイプライン 2 4 を例示するブロック図である。グラフィックス処理パイプライン 2 4 は、グラフィックス処理コマンドを実行するために共に動作する複数のグラフィックス処理ステージを含み得る。図 2 に示されるように、グラフィックス処理パイプライン 2 4 は、入力アセンブラー 2 6 、頂点シェーダステージ (vertex shader stage) 2 8 、ハルシェーダステージ (hull shader stage) 3 0 、テッセレータステージ (tessellator stage) 3 2 、ドメインシェーダステージ 3 4 、ジオメトリシェーダステージ 3 6 、およびピクセルシェーダステージ 3 8 を含み得る。グラフィックス処理パイプライン 2 4 の中のコンポーネントの各々は、固定機能コンポーネント (fixed-function components) 、(例えば、プログラマブルシェーダユニット上で実行しているシェーダプログラムの一部としての) プログラマブルコンポーネントとして、または固定機能とプログラマブルコンポーネントの組み合わせとしてインプリメントされ得る。

#### 【 0 0 2 4 】

[0029] G P U 1 2 は、G P U ドライバ 2 2 を介して、C P U 6 から1つ以上のグラフィックス処理コマンドを受信するように、およびディスプレイ可能なグラフィックス画像 (displayable graphics images) を生成するためにはグラフィックス処理パイプライン 2 4 を介してグラフィックス処理コマンドを実行するように構成され得る。上記に説明されたように、グラフィックス処理パイプライン 2 4 は、グラフィックス処理コマンドを実行するために共に動作する複数のステージを含む。

#### 【 0 0 2 5 】

[0030] グラフィックス処理パイプライン 2 4 の中の入力アセンブラー 2 6 は、グラフィックス処理パイプライン 2 4 にグラフィックスデータ (例えば、三角形、線、および点) を供給することを一般に担う固定機能ステージであり得る。例えば、入力アセンブラーステージ 2 6 は、高次サーフェス (high order surfaces) 、プリミティブ等のための頂点デ

10

20

30

40

50

ータを集めることができ、頂点データおよび属性を頂点シェーダステージ28に出力し得る。よって、入力アセンブラーステージ26は、固定機能オペレーションを使用して、システムメモリ10のような、オフチップメモリから頂点を読み取り得る。入力アセンブラーステージ26は次に、これらの頂点からパイプライン作業項目(pipeline work items)を作ることができ、その一方で頂点識別子('Vertex IDs')、インスタンス識別子(頂点シェーダに利用可能である(made available to) 'Instance IDs')、およびプリミティブ識別子(ジオメトリシェーダおよびピクセルシェーダに利用可能である'Primitive IDs')も生成する。入力アセンブラーステージ26は、頂点を読み取ると、Vertex IDs、Instance IDs、およびPrimitive IDsを自動的に生成し得る。

10

#### 【0026】

[0031] 頂点シェーダステージ28は、受信された頂点データおよび属性を処理し得る。例えば、頂点シェーダステージ28は、変換(transformations)、スキニング(skinning)、頂点変位(vertex displacement)、および頂点ごとの物質属性(per-vertex material attributes)を計算することのような頂点ごとの処理を行い得る。いくつかの例では、頂点シェーダステージ28は、テクスチャ座標、頂点カラー(vertex color)、頂点ライティング(vertex lighting)、フォグファクタ(fog factors)等を生成し得る。頂点シェーダステージ28は一般に、単一の入力頂点を取り、単一の、処理された出力頂点を出力する。

#### 【0027】

[0032] ハルシェーダステージ30、テッセレータ32、およびドメインシェーダステージ34は、テッセレーションステージと集合的に称され得る。テッセレーションステージは、低詳細サブディビジョンサーフェス(low-detail subdivision surfaces)を高詳細プリミティブ(higher-detail primitives)に変換(convert)、レンダリングのために高次サーフェスを適したサーフェス(例えば、三角形)にタイル表示する(tiles)。ハルシェーダステージ30は、頂点シェーダステージ28からプリミティブを受信し、少なくとも2つのアクションを実行することを担う。まず、ハルシェーダステージ30は、テッセレーションファクタのセットを決定することを典型的に担う。ハルシェーダステージ30は、プリミティブごとに1回テッセレーションファクタを生成し得る。テッセレーションファクタは、所与のプリミティブをどのくらい細かく(how finely)テッセレートする(tessellate)(例えば、プリミティブをより小さいパーツに分割する)かを決定するためにテッセレータステージ32によって使用され得る。ハルシェーダステージ30はまた、ドメインシェーダステージ34によって後で使用されることとなる制御点(control points)を生成することを担う。すなわち、例えば、ハルシェーダステージ30は、実際にテッセレートされた頂点(actual tessellated vertices)を作るためにドメインシェーダステージ34によって使用されることとなる制御点を生成することを担い、それは、レンダリングで最終的に使用される。

20

#### 【0028】

[0033] テッセレータステージ32がハルシェーダステージ30からデータを受信するとき、テッセレータステージ32は、現在のプリミティブタイプのための適切なサンプリングパターンを決定するためにいくつかのアルゴリズムのうちの1つを使用する。例えば、一般に、テッセレータステージ32は、(ハルシェーダステージ30によって決定されるような)テッセレーションの要求された量を現在の「ドメイン」内の座標点のグループに変換する。すなわち、ハルシェーダステージ30からのテッセレーションファクタ、ならびにテッセレータステージ32の特定の構成に依存して、テッセレータステージ32は、現在のプリミティブの中のどの点が入力プリミティブをより小さいパーツにテッセレートするためにサンプリングされる必要があるかを決定する。テッセレータステージ32の出力は、ドメイン点のセットであることができ、それは、重心座標(barycentric coordinates)を含み得る。

30

#### 【0029】

40

50

[0034] ドメインシェーダステージ34は、ハルシェーダステージ30によって作り出される制御点に加えて、ドメイン点を取り、新たな頂点を作るためにドメイン点を使用する。ドメインシェーダステージ34は、各テッセレートされた点に関する重心「ロケーション（location）」をパイプラインの中の次のステージにバスされる出力ジオメトリに変換するための現在のプリミティブ、テクスチャ、手続型のアルゴリズム（procedural algorithms）、またはその他のもののために生成される制御点の完全なリスト（complete list）を使用することができる。

#### 【0030】

[0035] ジオメトリシェーダステージ36は、その頂点データ（例えば、三角形のための3つの頂点、線のための2つの頂点、または点のための単一の頂点）によって定義されるプリミティブを受信することができ、プリミティブをさらに処理し得る。例えば、ジオメトリシェーダステージ36は、他の可能性がある処理オペレーションの間で、シルエットエッジ検出（silhouette-edge detection）およびシャドウボリューム押し出し（shadow volume extrusion）のようなプリミティブごとの処理を行い得る。よって、ジオメトリシェーダステージ36は、（1つ以上の頂点を含み得る）入力および出力0、1、または（この場合も先と同様に（again）1つ以上の頂点を含み得る）複数のプリミティブとして1つのプリミティブを受信し得る。出力プリミティブは、ジオメトリシェーダステージ36なしで可能であり得るより多くのデータを包含し得る。出力データの総量は、頂点カウントを乗じた（multiplied by the vertex count）頂点サイズと同等であることができ、起動（invocation）ごとに制限され得る。ジオメトリシェーダステージ36からのストリーム出力は、このステージに達するプリミティブが、システムメモリ10のようなオフチップメモリに記憶されることを可能にし得る。ストリーム出力は、ジオメトリシェーダステージ36に典型的に関係しており（tied）、両方は、（例えば、APIを使用して）共にプログラムされ得る。

#### 【0031】

[0036] ラスタライザステージ（rasterizer stage）37は典型的に、ピクセルシェーダステージ38のために、プリミティブをクリッピングすること、およびプリミティブを準備することを担う固定機能ステージである。例えば、ラスタライザステージ37は、（カスタムクリップ境界（custom clip boundaries）を含む）クリッピング、パースペクティブデバイド（perspective divide）、ビューポート／切り取り選択（viewport/scissor selection）、およびインプリメンテーションを行うことができ、ターゲット選択およびプリミティブセットアップをレンダリングし得る。このようにして、ラスタライザステージ37は、ピクセルシェーダステージ38によってシェーディングするためのいくつかのフラグメント（fragments）を生成し得る。

#### 【0032】

[0037] ピクセルシェーダステージ38は、ラスタライザステージ37からフラグメントを受信し、カラーのような、ピクセルごとのデータを生成する。ピクセルシェーダステージ38はまた、テクスチャブレンディング（texture blending）およびライティングモデル計算（lighting model computation）のようなピクセルごとの処理も行い得る。よって、ピクセルシェーダステージ38は、入力として1つのピクセルを受信することができ、同じ相対位置（relative position）（またはピクセルのためのゼロ値）で1つのピクセルを出力し得る。

#### 【0033】

[0038] 本開示の態様にしたがって、グラフィックス処理パイプライン24の2つ以上のステージは、グラフィックスメモリ40の中で共有されるデータチャネルを共有し得る。例えば、頂点シェーダステージ28およびドメインシェーダステージ34によって作り出される頂点は、共有されるデータチャネルの中に記憶され得る。さらに、ハルシェーダステージ30およびジオメトリシェーダステージ36によって作り出されるプリミティブは、別の共有されるデータチャネルの中に記憶され得る。このようにして、GPU12は、グラフィックスメモリ40をより効率的に利用し得る。

10

20

30

40

50

## 【0034】

[0039] 図3は、さらに詳細に図1のCPU6、GPU12、およびシステムメモリ10の例示的なインプリメンテーションを例示するブロック図である。図3に示されるように、CPU6は、少なくとも1つのソフトウェアアプリケーション18、グラフィックスAPI20、およびGPUドライバ22を含むことができ、それらの各々は、CPU6上で実行する1つ以上のソフトウェアアプリケーションまたはサービスであり得る。

## 【0035】

[0040] CPU6およびGPU12に利用可能であるメモリは、システムメモリ10およびフレームバッファ16を含み得る。フレームバッファ16は、システムメモリ10の一部であることができ、またはシステムメモリ10から分離され得る。フレームバッファ16は、レンダリングされた画像データを記憶し得る。

10

## 【0036】

[0041] ソフトウェアアプリケーション18は、GPU12の機能を利用する任意のアプリケーションであり得る。例えば、ソフトウェアアプリケーション18は、GUIアプリケーション、オペレーティングシステム、ポータブルマッピングアプリケーション(*portable mapping application*)、エンジニアリングまたは芸術的なアプリケーション(*artistic applications*)のためのコンピュータ支援設計プログラム(*computer-aided design program*)、ビデオゲームアプリケーション、あるいは2Dまたは3Dグラフィックスを使用する別のタイプのソフトウェアアプリケーションであり得る。

## 【0037】

20

[0042] ソフトウェアアプリケーション18は、グラフィカルユーザインターフェース(GUI)および/またはグラフィックスシーンをレンダリングするためにGPU12に命令する1つ以上の描画命令を含み得る。例えば、描画命令は、GPU12によってレンダリングされるための1つ以上のグラフィックスプリミティブのセットを定義する命令を含み得る。いくつかの例では、描画命令は、GUIで使用される複数のウィンドウイングサーフェス(*windowing surfaces*)のすべてまたは一部を集合的に定義し得る。追加の例では、描画命令は、アプリケーションによって定義されるモデルスペースまたはワールドスペース内の1つ以上のグラフィックスオブジェクトを含むグラフィックスシーンのすべてまたは一部を集合的に定義し得る。

## 【0038】

30

[0043] ソフトウェアアプリケーション18は、1つ以上のグラフィックスプリミティブをディスプレイ可能なグラフィックス画像にレンダリングするためのGPU12への1つ以上のコマンドを発行するために、グラフィックスAPI20を介して、GPUドライバ22を呼び出し得る。例えば、ソフトウェアアプリケーション18は、GPU12にプリミティブ定義(*primitive definitions*)を提供するために、グラフィックスAPI20を介して、GPUドライバ22を呼び出し得る。いくつかの例では、プリミティブ定義は、例えば、三角形、長方形、トライアングルファン(*triangle fans*)、トライアングルストリップ(*triangle strips*)等の描画プリミティブのリストの形でGPU12に提供され得る。プリミティブ定義は、レンダリングされるためのプリミティブと関連する1つ以上の頂点を特定する頂点仕様(*vertex specifications*)を含み得る。頂点仕様は、各頂点に関する位置座標(*positional coordinates*)および、いくつかの例では、例えば、カラー座標、法線ベクトル(*normal vectors*)、およびテクスチャ座標のような、頂点と関連する他の属性を含み得る。プリミティブ定義はまた、プリミティブタイプの情報(例えば、三角形、長方形、トライアングルファン、トライアングルストリップ等)、スケーリング情報、回転情報等も含み得る。ソフトウェアアプリケーション18によってGPUドライバ22に発行される命令に基づいて、GPUドライバ22は、プリミティブをレンダリングするために行うためのGPU12のための1つ以上のオペレーションを特定する1つ以上のコマンドを公式化し(*formulate*)得る。GPU12がCPU6からコマンドを受信するとき、グラフィックス処理パイプライン24は、そのコマンドを復号し、そのコマンドにおいて特定されたオペレーションを行うようにグラフィックス処理パイプラ

40

50

イン24を構成する。例えば、グラフィックス処理パイプライン24の中の入力アセンブラー26は、プリミティブデータを読み取ることができ、グラフィックス処理パイプライン24の中の他のグラフィックスパイプラインステージによる使用のためにデータをプリミティブにアセンブルし(assemble)得る。特定されたオペレーションを行った後、グラフィックス処理パイプライン24は、ディスプレイデバイスと関連するフレームバッファ16にレンダリングされたデータを出力する。

#### 【0039】

[0044] フレームバッファ16は、GPU12のために宛先ピクセル(destination pixels)を記憶する。各宛先ピクセルは、一意的なスクリーンピクセルロケーションと関連し得る。いくつかの例では、フレームバッファ16は、各宛先ピクセルのための宛先アルファ値(destination alpha value)およびカラーコンポーネントを記憶し得る。例えば、フレームバッファ16は、「RGB」コンポーネントがカラー値に対応し、「A」コンポーネントが宛先アルファ値に対応する各ピクセルに関する赤、緑、青、アルファ(RGBA)コンポーネントを記憶し得る。フレームバッファ16およびシステムメモリ10は、別個のメモリユニットであるように例示されているが、他の例では、フレームバッファ16は、システムメモリ10の一部であり得る。

#### 【0040】

[0045] いくつかの例では、グラフィックス処理パイプライン24の頂点シェーダステージ28、ハルシェーダステージ30、ドメインシェーダステージ34、ジオメトリシェーダステージ、およびピクセルシェーダステージ38は、シェーダステージと見なされ得る。これらのシェーダステージは、GPU12の中のシェーダユニット46上で実行する1つ以上のシェーダプログラムとしてインプリメントされ得る。シェーダユニット46は、処理コンポーネントのプログラマブルパイプラインとして構成され得る。いくつかの例では、シェーディングユニット(shading unit)46は、「シェーダプロセッサ」、または「統合シェーダ(unified shaders)」と称されることができ、グラフィックスをレンダリングするためにジオメトリ、頂点、ピクセル、または他のシェーディングオペレーションを行い得る。シェーダユニット46は、プロセッサコア48を含むことができ、それらの各々は、オペレーションをフェッチする(fetching)および復号するための1つ以上のコンポーネント、算術計算(arithmetic calculations)を実行するための1つ以上の算術論理ユニット(arithmetic logic units)、1つ以上のメモリ、キャッシュ、およびレジスタを含み得る。

#### 【0041】

[0046] GPU12は、グラフィックス処理パイプライン24の中の頂点シェーダステージ28、ハルシェーダステージ30、ドメインシェーダステージ34、ジオメトリシェーダステージ36、およびピクセルシェーダステージ38のうちの1つ以上を実行するためにコマンドをシェーダユニット46に送ることによって、頂点シェーディング、ハルシェーディング、ドメインシェーディング、ジオメトリシェーディング、ピクセルシェーディング等のような様々なシェーディングオペレーションを行うためにシェーダユニット46を指定し得る。いくつかの例では、GPUドライバ22は、1つ以上のシェーダプログラムをコンパイルするように、およびコンパイルされたシェーダプログラムをGPU12内に包含される1つ以上のプログラマブルシェーダユニットにダウンロードするように構成され得る。シェーダプログラムは、例えば、OpenGLシェーディング言語(GLSL)、ハイレベルシェーディング言語(HLSL)、グラフィックスのためのC(Cg)シェーディング言語(C for Graphics(Cg) shading language)等のような、ハイレベルシェーディング言語で書き込まれ得る。コンパイルされたシェーダプログラムは、GPU12内のシェーダユニット46のオペレーションを制御する1つ以上の命令を含み得る。例えば、シェーダプログラムは、頂点シェーダステージ28の機能を行うためにシェーダユニット46によって実行され得る頂点シェーダプログラム、ハルシェーダステージ30の機能を行うためにシェーダユニット46によって実行され得るハルシェーダプログラム、ドメインシェーダステージ34の機能を行うためにシェーダユニット46によって実行

10

20

30

40

50

され得るドメインシェーダプログラム、ジオメトリシェーダステージ3 6の機能を行うためにシェーダユニット4 6によって実行され得るジオメトリシェーダプログラム、および／またはピクセルシェーダ3 8の機能を行うためにシェーダユニット4 6によって実行され得るピクセルシェーダプログラムを含み得る。頂点シェーダプログラムは、プログラマブル頂点シェーダユニット、または統合(unified)シェーダユニットの実行を制御することができ、1つ以上の頂点ごとのオペレーションを特定する命令を含み得る。

#### 【0042】

[0047] グラフィックスメモリ4 0は、GPU1 2の集積回路に物理的に組み込まれるオンチップ記憶装置、またはメモリである。グラフィックスメモリ4 0がオンチップであるため、GPU1 2は、システムバスを介してシステムメモリ1 0から値を読み取ること、またはシステムメモリ1 0に値を書き込むことより速くグラフィックスメモリ4 0から値を読み取り、またはグラフィックスメモリ4 0に値を書き込むことができ得る。そのようなものとして、シェーダユニット4 6のパフォーマンスは、グラフィックスメモリ4 0からグラフィックス処理パイプライン2 4のシェーダステージによって作り出される、および消費されるデータを読み取ること、および記憶することによって増加され得る。

#### 【0043】

[0048] 本開示の態様にしたがって、シェーダユニット4 6は、プロセッサコア4 8上で同時に複数のシェーディングオペレーションを行い得る。GPU1 2は、異なるプロセッサコア4 8上で実行されるためのグラフィックス処理パイプライン2 4の異なるシェーディングステージをイネーブルする(enable)コマンドをシェーディングユニット4 6に送ることができ、それによってグラフィックス処理パイプライン2 4のステージをインターリープする。例えば、GPU1 2は、シェーディングユニット4 6に、シェーダユニット4 6の異なるプロセッサコア4 8上で同時に頂点シェーダステージ2 8およびジオメトリシェーダステージ3 6を実行させるコマンドをシェーディングユニット4 6に送り得る。別の例では、GPU1 2は、シェーディングユニット4 6に、複数のプロセッサ上で同時にジオメトリシェーダステージ3 6の複数の例を実行させるコマンドをシェーディングユニット4 6に送り得る。

#### 【0044】

[0049] 本開示の態様にしたがって、グラフィックスメモリ4 0は、単一のデータチャネルを共有するためにグラフィックス処理パイプライン2 4の異なるステージによって作り出されるデータをイネーブルする(enable)共有されるデータチャネル5 0 A - 5 0 N(「共有されるデータチャネル5 0」)のうちの1つ以上を含むことができ、それによってGPU1 2がグラフィックスメモリ4 0の中の限られたスペースをより効率的に利用することをイネーブルし、またシェーダプロセッサクラスタ4 6がグラフィックス処理パイプライン2 4の複数のステージを同時に実行するためにそのプロセッサコア4 8の利用を増加させることもイネーブルする。

#### 【0045】

[0050] 共有されるデータチャネル5 0の中の各共有されるデータチャネルは、グラフィックス処理パイプライン2 4の2つ以上のステージによって作り出されるデータを記憶し得る。グラフィックス処理パイプライン2 4の個別のステージのためにデータチャネルを割り当てるとは対照的に、共有されるデータチャネル5 0の中の共有されるデータチャネルを共有することによって、グラフィックス処理パイプライン2 4の中のステージがより少ないデータを作り出す場合、同じ共有されるデータチャネルを共有する別のステージは、それが共有されるデータチャネルの中で作り出すより多くのデータを記憶することによってそのファクト(fact)を活用することが可能であり得る。

#### 【0046】

[0051] 本開示の態様にしたがって、ジオメトリ処理ユニット(GPC)4 2は、共有されるデータチャネル5 0のステータスに基づいてシェーダプロセッサクラスタ4 6の実行をスケジューリングし得る。GPC4 2は、シェーダプロセッサクラスタ4 6によって実行されるためにグラフィックス処理パイプライン2 4のステージによって消費されるた

10

20

30

40

50

めの十分なデータが共有されるデータチャネル 50 の中にあるか否かを決定するために、共有されるデータチャネル 50 をモニタし (monitor) 得る。G P C 4 2 はまた、シェーダプロセッサクラスタ 4 6 によって実行されるためにグラフィックス処理パイプライン 2 4 のステージによって作り出されるデータを記憶するための十分なフリースペースが共有されるデータチャネル 50 の中にあるか否かを決定するために、共有されるデータチャネル 50 をモニタし得る。G P C 4 2 が、共有されるデータチャネル 50 の中に十分なデータおよびフリースペースが存在することを決定する場合、G P C 4 2 は、グラフィックス処理パイプライン 2 4 のステージのバッチ (batch of stages) を実行するために、実行コマンドをシェーダプロセッサクラスタ 4 6 に送り得る。ステージのバッチの実行を完了したことに応答して、シェーダプロセッサクラスタ 4 6 は、プロセッサクラスタ 4 6 がステージのバッチの実行を完了したことを示す信号を G P C 4 2 に送り得る。それに応答して、データチャネルマネージャ (data channel manager) 4 4 は、共有されるデータチャネル 50 のための関連のある読み取りおよび書き込みポインタをアップデートし得る。G P C 4 2 は、共有されるデータチャネル 50 を管理するデータチャネルマネージャ 4 4 を含み得る。データチャネルマネージャ 4 4 は、共有されるデータチャネル 50 にデータを書き込む、および共有されるデータチャネル 50 からデータを読み取るために共有されるデータチャネル 50 内でロケーションを指示し得る。データチャネル 50 のための読み取りおよび書き込みポインタを管理し得る。  
10

#### 【 0 0 4 7 】

[0052] 本開示の態様にしたがって、共有されるデータチャネル 50 A は、共有されるデータチャネル 50 A がグラフィックス処理パイプライン 2 4 の第 1 のステージによって出力されたデータ 55 A と、グラフィックス処理パイプライン 2 4 の第 2 のステージによって出力されたデータ 55 B の両方を記憶し得るよう、グラフィックス処理パイプライン 2 4 の 2 つ以上のステージによって共有されるデータチャネルであり得る。共有されるデータチャネル 50 A は、データ 55 A および 55 B が、それらが作り出された、および / または消費されたとき、そのサイズを動的に増加と低減の両方をさせることができるようにリングバッファであることができ、それによって共有されるデータチャネル 50 A に割り当てられたメモリロックのより効率的な使用を可能にする。G P C 4 2 は、書き込みポインタ 51 A および 51 B、ならびに読み取りポインタ 53 A および 53 B を管理し得る。書き込みポインタ 51 A は、データ 55 A を書き込むために、共有されるデータチャネル 50 A のメモリロケーションを指示し得る。読み取りポインタ 53 A は、データ 55 A を読み取るために、共有されるデータチャネル 50 A のメモリロケーションを指示し得る。  
20

#### 【 0 0 4 8 】

[0053] 典型的に、読み取りポインタ 53 A および 53 B が、それぞれ、キューの先頭 (head of the queue) と称されることもある、データ 55 A および 55 B の中のデータの最も古い部分 (oldest piece) を記憶する共有されるデータチャネル 50 A のメモリロケーションを指示し得る。ならびに書き込みポインタ 51 A および 51 B が、それ、キューの末端 (tail of the queue) と称されることもある、データ 55 A および 55 B の中のデータの最も新しい部分を記憶する共有されるデータチャネル 50 A のメモリロケーションを指示し得る。G P U 1 2 は、ファーストインファーストアウト (first-in-first-out) (FIFO) の順序で共有されるデータチャネル 50 A の中にデータ 55 A および 55 B を記憶する。  
30

#### 【 0 0 4 9 】

[0054] 共有されるデータチャネル 50 A はまた、データ 55 A および 55 B から読み取られたデータが共有されるデータチャネル 50 A から削除され、それらのメモリロケーションが割り当てを解除され (deallocated) 得るよう F I F O モードで動作し得る。以上のように、G P U 1 2 が共有されるデータチャネル 50 A からデータ 55 A を読み取るとき、共有されるデータチャネル 50 A の中のフリースペース 57 は増加し、それによって G P U 1 2 のための共有されるデータチャネル 50 A の中の追加のスペースがデータ 40

55Bにデータを書き込むことを可能にする。同様に、GPU12が共有されるデータチャネル50Aからデータ55Bを読み取るとき、共有されるデータチャネル50Aの中のフリースペース59は増加し、それによってGPU12のための共有されるデータチャネル50Aの中の追加のスペースがデータ55Aにデータを書き込むことを可能にする。共有されるデータチャネル50Aのみが上記に詳細に説明されたが、共有されるデータチャネル50の中の各共有されるデータチャネルが、共有されるデータチャネル50Aについて上記に説明された特徴を共有し得ることは理解されるべきである。

#### 【0050】

[0055] 図4は、グラフィックス処理パイプライン24の中で使用されている共有されるデータチャネル50の例を例示するプロック図である。図4に示されるように、共有されるデータチャネル50Aは、ステージによって作り出されるデータを記憶するためにグラフィックス処理パイプライン24のステージによって共有され得る。10

具体的には、共有されるデータチャネル50Aは、グラフィックス処理パイプライン24のハルシェーダステージ30によって作り出されるデータ52を記憶することができ、グラフィックス処理パイプライン24のジオメトリシェーダステージ36によって作り出されるデータ54をさらに記憶し得る。データ52は、グラフィックス処理パイプライン24のドメインシェーダステージ34によって消費されることができ、データ54は、グラフィックス処理パイプライン24のピクセルシェーダステージによって消費され得る。

#### 【0051】

[0056] ハルシェーダステージ30およびジオメトリシェーダステージ36によって共有されるデータチャネル50Aに記憶されたデータ52およびデータ54は、それぞれ、ハルシェーダステージ30によって出力されるパッチ制御点およびジオメトリシェーダステージ36によって出力される頂点を含み得る。データチャネル50Aがデータ52および54をキャッシュしないため、データ52および54は、データ52および54から読み取られるデータが共有されるデータチャネル50Aから削除される FIFOキューとして各自動作し得る。20

#### 【0052】

[0057] いくつかの例では、グラフィックス処理パイプライン24のいくつかのステージによって作り出される同じデータは、グラフィックス処理パイプライン24の他のステージによって複数回 (multiple times) 消費され得る。データが FIFOキューとして動作する共有されるデータチャネル50の1つに記憶される場合、FIFOキューに記憶されるデータがそれが FIFOキューから読み取られるとき削除され得るため、データを作り出すグラフィックス処理パイプライン24のステージは、同じデータを作り出すために複数回実行する必要があり得る。複数回同じ頂点を作り出すために複数回頂点シェーダ28またはドメインシェーダ34を実行することに代わって、GPU12は、キャッシュモード共有チャネル56の中で頂点シェーダ28およびドメインシェーダ34によって作り出されるデータを代わりにキャッシュし得る。30

#### 【0053】

[0058] 例えば、頂点シェーダステージ28によって変換される頂点を含む、グラフィックス処理パイプライン24の頂点シェーダステージ28によって作り出されるデータは、グラフィックス処理パイプライン24のハルシェーダステージ30によって消費され得る。同様に、ドメインシェーダステージ34によって出力される頂点位置のような、グラフィックス処理パイプライン24のドメインシェーダステージ34によって作り出されるデータは、グラフィックス処理パイプライン24のジオメトリシェーダステージ36によって消費され得る。例えば、近接したプリミティブ（例えば、三角形）が頂点を共有し得るため、同じ頂点は、2つの近接した三角形を形成するために使用され得る。したがって、頂点シェーダステージ28およびドメインシェーダステージ34によって作り出される頂点データは、複数回消費され得る。頂点シェーダステージ28およびドメインシェーダステージ34によって作り出されるデータは、複数回消費され得るため、これらのステージによって作り出されるデータは、キャッシュされたデータが、それがキャッシュモード40

共有チャネル 5 6 から読み取られることに応答して削除され得ないように、キャッシングモード共有チャネル 5 6 の中でキャッシングされ得る。

#### 【0054】

[0059] 図 5 は、キャッシングモード共有チャネル 5 6 を例示するブロック図である。図 5 に示されるように、キャッシングモード共有チャネル 5 6 は、2 つの共有されるデータチャネル：共有されるプリミティブキューリミット 50B および共有される頂点キャッシングリミット 50C、ならびにキャッシングウインドウ 70 を含み得る。共有される頂点キャッシングリミット 50C は、共有される頂点キャッシングリミット 50C に記憶されるデータが共有される頂点キャッシングリミット 50C から読み取られる際に削除されない場合があるようにキャッシングモードで動作し得る。共有されるプリミティブキューリミット 50B に記憶されるデータ 62 およびデータ 64 は、頂点シェーダステージ 28 およびドメインシェーダステージ 34 によって作り出されるプリミティブデータを含み得る。例えば、データ 62 は、各プリミティブのために頂点シェーダステージ 28 によって作り出された、共有される頂点キャッシングリミット 50C で記憶された頂点データの頂点インデックスおよびロケーションを含むことができ、データ 64 は、各プリミティブのためにドメインシェーダステージ 34 によって作り出された、共有される頂点キャッシングリミット 50C で記憶された頂点データの頂点インデックスおよびロケーションを含み得る。データ 62 および 64 はまた、関連するプリミティブの各々のための割り当て解除フラグ (deallocation flags) を含み得る。共有される頂点キャッシングリミット 50C に記憶されたデータ 66 は、頂点シェーダステージ 28 によって変換された頂点を含むことができ、一方、共有される頂点キャッシングリミット 50C に記憶されたデータ 68 は、ドメインシェーダステージ 34 によって出力された頂点位置を含み得る。G P C 4 2 は、キャッシングモード共有チャネル 5 6 がデータを受け取るために十分なフリースペースを有するか否かを決定するために、共有されるプリミティブキューリミット 50B と共有される頂点キャッシングリミット 50C の両方のフリースペースをチェックし得る。

10

#### 【0055】

[0060] キャッシングウインドウ 70 は、特定の頂点が、共有される頂点キャッシングリミット 50C の限られたウインドウの中に既に記憶されている場合、インジケーションを記憶し得る。例えば、キャッシングウインドウ 70 は、フリーアソシエイティブキャッシング (fully associative cache) として動作することができ、頂点を消費し得るシェーダの、共有される頂点キャッシングリミット 50C 内の頂点のデータロケーション、頂点インデックス、およびフラグのような、インジケーションを記憶し得る。

20

#### 【0056】

[0061] プリミティブによる G P C 4 2 処理ジオメトリプリミティブ。頂点シェーダ 28 およびドメインシェーダ 34 のために、G P C 4 2 が、頂点インデックス上のキャッシングウインドウ 70、および / または頂点が属するシェーダをチェックすることに基づいて、プリミティブの特定の頂点が共有される頂点キャッシングリミット 50C の中に存在しないことを決定する場合、キャッシングミスが発生することができ、そして G P C 4 2 は、所望の頂点を作り出すために、およびキャッシングモード共有チャネル 5 6 の中に作り出された頂点データを記憶するために適切なシェーダステージ ( 例えば、頂点シェーダ 28 またはドメインシェーダ 34 ) を実行するためのコマンドをシェーダユニット 46 に送り得る。G P C 4 2 は、共有される頂点キャッシングリミット 50C で頂点データの頂点インデックスおよびロケーションを共有されるプリミティブキューリミット 50B に追加し得る。G P C 4 2 は、キャッシングモード共有チャネル 5 6 の中の今キャッシングされた頂点に関する適切なデータをキャッシングウインドウ 70 に追加し得る。キャッシングウインドウ 70 は、キャッシングミスの後にキャッシングウインドウ 70 の中に余地がない場合、キャッシングウインドウ 70 の中の最も古いスロットの中に関連し、共有されるプリミティブキューリミット 50B の中にセットされたそれの割り当て解除フラグを有する頂点が、キャッシングモード共有チャネル 5 6 に追加される最新の頂点に関する情報でセットされ得るようにファーストインファーストアウト ( F I F O ) の方法で動作し得る。しかしながら、G P C 4 2 が特定の頂点がキャッシングモード共有チャネル 5 6 の中でキャッシングされることを決定する場合、G P C 4 2 は、所望の

30

40

50

頂点の共有される頂点キャッシュ 50C の中のメモリロケーションを使用することができ、共有される頂点キャッシュ 50C で頂点データの頂点インデックスおよびロケーションを共有されるプリミティブキュー 50B に追加することができる。このようにして、G P U 1 2 は、グラフィックス処理パイプライン 2 4 の中のステージの無関係な処理 (extraneous processing) を低減することができる。

#### 【 0 0 5 7 】

[0062] ハルシェーダ 3 0 およびジオメトリシェーダ 3 6 を実行するために、G P C 4 2 は、共有されるプリミティブキュー 50B と共有される頂点キャッシュ 50C の両方からデータを消費し得る。G P C 4 2 は、共有されるプリミティブキュー 50B から共有される頂点キャッシュ 50C の中の頂点データの頂点インデックスおよびロケーションを読み取り得る。G P C 4 2 は次に、共有されるプリミティブキュー 50B から読み取ったロケーションを使用して、共有される頂点キャッシュ 50C から頂点データを読み取り得る。G P C 4 2 は、データを読み取った後、共有されるプリミティブキュー 50B の読み取りポインタを動かし得る。しかしながら、続くプリミティブが、共有される頂点キャッシュ 50C から丁度読み取られた同じ頂点も使用し得るため、キャッシュされた頂点が、共有される頂点キャッシュ 50C から読み取られた直後、G P C 4 2 は、共有される頂点キャッシュ 50C の読み取りポインタを即時に動かさないことがある。頂点を消費するプリミティブのための共有されるプリミティブキュー 50B の中の関連する割り当てを解除されたフラグがセットされる場合、G P C 4 2 は、読み取りポインタを動かすこと、およびキャッシュモード共有チャネル 5 6 から頂点を割り当て解除することを可能にされ得る。G P C 4 2 は、頂点データを消費するためのシェーダステージ (例えば、ハルシェーダ 3 0 およびジオメトリシェーダ 3 6 ) を実行するために、および次のシェーダステージのための頂点を作り出し、共有されるデータチャネル 50A の中の作り出された頂点データを記憶するためにコマンドをシェーダユニット 4 6 に送り得る。

#### 【 0 0 5 8 】

[0063] G P C 4 2 は、デッドロックのためのキャッシュモード共有チャネル 5 6 および共有されるデータチャネル 50A をモニタし得る。一例では、キャッシュモード共有チャネル 5 6 が頂点シェーダステージ 2 8 によって作り出されるデータでいっぱいである場合、および共有されるデータチャネル 50A がハルシェーダステージ 3 0 によって作り出されるデータでいっぱいである場合、デッドロックは、発生し得る。この場合には、ハルシェーダステージ 3 0 が頂点ステージ 2 8 によって作り出されたデータを消費するため、ハルシェーダステージ 3 0 は、新たに作り出されたデータを記憶するための共有されるデータチャネル 50A の中のフリースペースが存在しないため、共有されるデータチャネル 50A に記憶されたデータを作り出すために頂点シェーダステージ 2 8 によって作り出され、キャッシュモード共有チャネル 5 6 の中に記憶されるデータを消費することができない。さらに、キャッシュモード共有チャネル 5 6 が、頂点シェーダステージ 2 8 によって作り出されるデータでいっぱいであり、そのデータのいずれもハルシェーダ 3 0 によって消費されることができないため、そのデータのいずれも、ドメインシェーダ 3 4 によって作り出されたデータを記憶するためのキャッシュモード共有チャネル 5 6 に関するスペースを解放する (free up) ために割り当てを解除されることができない。さらに、ドメインシェーダ 3 4 がハルシェーダステージ 3 0 によって作り出され、共有されるデータチャネル 50A の中に記憶されるデータを消費するため、ハルシェーダ 3 0 A によって作り出され、共有されるデータチャネル 50A の中に記憶されるデータのいずれも、ジオメトリシェーダ 3 6 によって作り出されたデータを記憶するための共有されるデータチャネル 50A に関する、共有されるデータチャネル 50A の中のスペースを解放するためにドメインシェーダ 3 4 によって消費されることができない。

#### 【 0 0 5 9 】

[0064] キャッシュモード共有チャネル 5 6 と共有されるデータチャネル 50A 間のデッドロック状況を防ぐために、キャッシュモード共有チャネル 5 6 および共有されるデータチャネル 50A がそれぞれ、頂点シェーダ 2 8 およびハルシェーダ 3 0 によって作り出

10

20

30

40

50

されるデータを記憶するだけでなく、G P C 4 2 は、それぞれドメインシェーダ 3 4 およびジオメトリシェーダ 3 6 によって作り出されるデータを記憶するためにキャッシュモード共有チャネル 5 6 および共有されるデータチャネル 5 0 A の中のスペースを保持し (reserve) 得る。G P C 4 2 は、共有されるプリミティブキュー 5 0 B と共有される頂点キャッシュ 5 0 C の両方のコンポーネントの中でキャッシュモード共有チャネル 5 6 のスペースの量、および例えば、シェーダクラスタ 4 6 の中の所与の数の波 (waves) のためにドメインシェーダ 3 4 およびジオメトリシェーダ 3 6 からの出力を記憶するために必要であるスペースの量を決定することによって、保持するべき共有されるデータチャネル 5 0 A のスペースの量を決定し得る。

## 【0060】

10

[0065] 図 6 は、グラフィックス処理パイプラインのステージによってデータチャネルを共有するための例示的なプロセスを例示する流れ図である。図 6 に示されるように、処理は、G P U 1 2 が、グラフィックス処理パイプライン 2 4 の少なくとも 2 つのステージによって共有される G P U 1 2 のオンチップグラフィックスメモリ 4 0 の中に共有されるデータチャネル 5 0 A を割り当てるなどを含み得る (502)。プロセスは、G P U 1 2 の中のシェーダユニット 4 6 上で、グラフィックス処理パイプライン 2 4 の少なくとも 2 つのステージを実行することをさらに含み得る (504)。プロセスは、オンチップグラフィックスメモリ 4 0 の中の共有されるデータチャネル 5 0 A の中に G P U 1 2 が、シェーダユニット 4 6 上で実行しているグラフィックス処理パイプライン 2 4 の少なくとも 2 つのステージによって作り出されるデータを記憶することをさらに含み得る (506)。

## 【0061】

20

[0066] いくつかの例では、プロセスは、G P U 1 2 が、グラフィックス処理パイプライン 2 4 の第 2 の少なくとも 2 つのステージによって共有される G P U 1 2 のオンチップグラフィックスメモリ 4 0 の中に第 2 のキャッシュモード共有チャネル 5 6 を割り当てるなどをさらに含むことができ、ここにおいて、共有されるデータチャネル 5 0 A は、第 1 の共有されるデータチャネルである。いくつかの例では、プロセスは、G P U 1 2 の中のシェーダユニット 4 6 上で、グラフィックス処理パイプライン 2 4 の第 2 の少なくとも 2 つのステージを実行することをさらに含み得る。いくつかの例では、プロセスは、第 2 のキャッシュモード共有チャネル 5 6 の中に G P U 1 2 が、シェーダユニット 4 6 上で実行しているグラフィックス処理パイプライン 2 4 の第 2 の少なくとも 2 つのステージの各々によって作り出される第 2 のデータを記憶することをさらに含み得る。

## 【0062】

30

[0067] いくつかの例では、データが、シェーダユニット 4 6 上で実行しているとき、グラフィックス処理パイプライン 2 4 の 1 つ以上のステージによって消費されるために第 1 の共有されるデータチャネル 5 0 A、または第 2 のキャッシュモード共有チャネル 5 6 で利用可能であり、ならびにフリースペースが、シェーダユニット 4 6 上で実行しているときグラフィックス処理パイプライン 2 4 の 1 つ以上のステージによって作り出されるデータを記憶するために第 1 の共有されるデータチャネル 5 0 A、または第 2 のキャッシュモード共有チャネル 5 6 で利用可能であるように、プロセスは、G P U 1 2 が、第 1 の共有されるデータチャネル 5 0 A または第 2 のキャッシュモード共有チャネル 5 6 のステータスに少なくとも部分的に基づいてシェーダユニット 4 6 によってグラフィックス処理パイプライン 2 4 の 1 つ以上のステージの実行をスケジューリングすることをさらに含み得る。

## 【0063】

40

[0068] いくつかの例では、グラフィックス処理パイプライン 2 4 の少なくとも 2 つのステージは、頂点シェーダ 2 8 およびドメインシェーダ 3 4 を含む。いくつかの例では、グラフィックス処理パイプライン 2 4 の第 2 の少なくとも 2 つのステージは、ハルシェーダ 3 0 およびジオメトリシェーダ 3 6 を含む。

## 【0064】

[0069] いくつかの例では、プロセスは、G P U 1 2 が、第 1 の共有されるデータチャ

50

ネル 50 A と第 2 のキャッシュモード共有チャネル 56 間のデッドロックを防ぐために、第 1 の共有されるデータチャネル 50 A、および第 2 のキャッシュモード共有チャネル 56 のうちの少なくとも 1 つの中のフリースペースを保持することをさらに含み得る。

【 0065 】

[0070] 1 つ以上の例では、説明された機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せでインプリメントされ得る。ソフトウェアでインプリメントされる場合、これらの機能は、コンピュータ可読媒体上で、1 つ以上の命令またはコードとして記憶または送信され得る。コンピュータ可読媒体は、1 つの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あらゆる媒体を含むコンピュータデータ記憶媒体または通信媒体を含み得る。データ記憶媒体は、本開示で説明された技法のインプリメンテーションのための命令、コード、および / またはデータ構造を取り出すために、1 つ以上のコンピュータまたは 1 つ以上のプロセッサによってアクセスされることができるあらゆる利用可能な媒体であり得る。限定ではなく例として、そのようなコンピュータ可読媒体は、RAM、ROM、EEPROM、CD-ROM または他の光学ディスク記憶装置、磁気ディスク記憶装置または他の磁気記憶デバイス、あるいは命令またはデータ構造の形で所望のプログラムコードを搬送または記憶するために使用されることができ、コンピュータによってアクセスされることができる任意の他の媒体を備えることができる。また、任意の接続は、コンピュータ可読媒体と厳密には称される。例えば、ソフトウェアが、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線 (DSL)、または赤外線、無線、およびマイクロ波のようなワイヤレス技術を使用して、ウェブサイト、サーバ、または他の遠隔ソースから送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、または赤外線、無線、およびマイクロ波のようなワイヤレス技術は、媒体の定義に含まれる。本明細書で使用される場合、ディスク (disk) およびディスク (disc) は、コンパクトディスク (CD)、レーザーディスク (登録商標)、光ディスク、デジタル多用途ディスク (DVD)、フロッピー (登録商標) ディスク、および Blu-ray (登録商標) ディスクを含み、ここでディスク (disk) は通常、磁気的にデータを再生し、一方ディスク (disc) は、レーザーを用いて光学的にデータを再生する。上記の組み合わせはまた、コンピュータ可読媒体の範囲内に含まれるべきである。

【 0066 】

[0071] コードは、1 つ以上のデジタルシグナルプロセッサ (DSPs)、汎用マイクロプロセッサ、特定用途向け集積回路 (ASICs)、フィールドプログラマブル論理アレイ (FPGAs)、または他の同等な集積またはディスクリート論理回路のような、1 つ以上のプロセッサによって実行され得る。よって、本明細書で使用される場合、「プロセッサ」および「処理ユニット」という用語は、前述の構造、または本明細書に説明された技法のインプリメンテーションに適した任意の他の構造のいずれかを指し得る。加えて、いくつかの態様では、本明細書に説明された機能は、符号化および復号のために構成された専用ハードウェアおよび / またはソフトウェアモジュール内で提供され得る、あるいは組み合わせられたコーデックの中に組み込まれ得る。また、技法は、1 つ以上の回路または論理要素で十分にインプリメントされることができる。

【 0067 】

[0072] 本開示の技法は、ワイヤレスハンドセット、集積回路 (IC) または ICs のセット (すなわち、チップセット) を含む、幅広い様々なデバイスまたは装置においてインプリメントされ得る。様々なコンポーネント、モジュール、またはユニットは、開示された技法を行うように構成されるデバイスの機能的な態様を強調するために本開示の中で説明されるが、必ずしも異なるハードウェアユニットによる実現を要求しない。むしろ、上記に説明されたように、様々なユニットは、コーデックハードウェアユニットにおいて組み合わされ得るか、または適切なソフトウェアおよび / またはファームウェアと併せて、上記に説明されたような 1 つ以上のプロセッサを含む、相互運用のハードウェアユニット (interoperative hardware units) の集合によって提供され得る。

## 【0068】

[0073] 様々な例が、説明された。これらおよび他の例は、以下の特許請求の範囲の範囲内にある。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

## [C 1]

グラフィックス処理の方法であって、

グラフィックス処理ユニット(GPU)が、グラフィックス処理パイプラインの少なくとも2つのステージによって共有される前記GPUのオンチップグラフィックスメモリの中に共有されるデータチャネルを割り当てることと、

前記GPUの中のシェーダユニット上で、前記グラフィックス処理パイプラインの前記少なくとも2つのステージを実行することと、

オンチップグラフィックスメモリの中の前記共有されるデータチャネルの中に前記GPUが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記少なくとも2つのステージの各々によって作り出されるデータを記憶することと、

を備える、方法。

## [C 2]

前記GPUが、前記グラフィックス処理パイプラインの第2の少なくとも2つのステージによって共有される前記GPUのオンチップグラフィックスメモリの中に第2の共有されるデータチャネルを割り当てることと、ここにおいて、前記共有されるデータチャネルは、第1の共有されるデータチャネルである、

前記GPUの中の前記シェーダユニット上で、前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージを実行することと、

前記第2の共有されるデータチャネルの中に前記GPUが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージの各々によって作り出される第2のデータを記憶することと、

をさらに備える、C 1に記載の方法。

## [C 3]

データが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの1つ以上のステージによって消費されるために前記第1の共有されるデータチャネル、または前記第2の共有されるデータチャネルにおいて利用可能であり、ならびにフリースペースが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記1つ以上のステージによって作り出されるデータを記憶するために前記第1の共有されるデータチャネル、または前記第2の共有されるデータチャネルにおいて利用可能であるように、前記GPUが、前記第1の共有されるデータチャネル、または前記第2の共有されるデータチャネルのステータスに少なくとも部分的に基づいて前記シェーダユニット上で前記グラフィックス処理パイプラインの前記1つ以上のステージの前記実行をスケジューリングすることをさらに備える、C 2に記載の方法。

## [C 4]

前記第1の共有されるデータチャネルは、前記第1の共有されるデータチャネルの中に記憶されたデータをキャッシュするためにキャッシュモードで動作し、前記第2の共有されるデータチャネルは、ファーストインファーストアウト(FIFO)モードで動作する、C 2に記載の方法。

## [C 5]

前記第1の共有されるデータチャネルは、第1のリングバッファであり、前記第2の共有されるデータチャネルは、第2のリングバッファである、C 2に記載の方法。

## [C 6]

前記第1の共有されるデータチャネルの中に記憶された前記データは、前記グラフィックス処理パイプラインの前記少なくとも2つのステージによって作り出される頂点データを備え、前記第2の共有されるデータチャネルの中に記憶された前記第2のデータは、前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージによって作り出される頂点データを備え、前記第1の共有されるデータチャネルと前記第2の共有されるデータチャネルとの間でデータを転送する、C 2に記載の方法。

10

20

30

40

50

出されるプリミティブを備える、C 2 に記載の方法。

[ C 7 ]

前記グラフィックス処理パイプラインの前記少なくとも 2 つのステージは、頂点シェーダおよびドメインシェーダを備える、C 6 に記載の方法。

[ C 8 ]

前記グラフィックス処理パイプラインの前記第 2 の少なくとも 2 つのステージは、ハルシェーダおよびジオメトリシェーダを備える、C 6 に記載の方法。

[ C 9 ]

前記 G P U が、前記第 1 の共有されるデータチャネルと前記第 2 の共有されるデータチャネル間のデッドロックを防ぐために、前記第 1 の共有されるデータチャネル、および前記第 2 の共有されるデータチャネルのうちの少なくとも 1 つの中のフリースペースを保持することをさらに備える、C 2 に記載の方法。

10

[ C 10 ]

グラフィックスデータを処理するように構成された装置であって、

グラフィックス処理パイプラインの少なくとも 2 つのステージによって共有されるグラフィックス処理ユニット ( G P U ) のオンチップグラフィックスメモリの中に共有されるデータチャネルを割り当てることと、

前記 G P U の中のシェーダユニット上で、前記グラフィックス処理パイプラインの前記少なくとも 2 つのステージを実行することと、

20

オンチップグラフィックスメモリの中の前記共有されるデータチャネルの中に、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記少なくとも 2 つのステージの各々によって作り出されるデータを記憶することと、

を行うように構成される前記 G P U を備える、装置。

[ C 11 ]

前記 G P U は、

前記グラフィックス処理パイプラインの第 2 の少なくとも 2 つのステージによって共有される前記 G P U のオンチップグラフィックスメモリの中に第 2 の共有されるデータチャネルを割り当てることと、ここにおいて、前記共有されるデータチャネルは、第 1 の共有されるデータチャネルである、

前記 G P U の中の前記シェーダユニット上で、前記グラフィックス処理パイプラインの前記第 2 の少なくとも 2 つのステージを実行することと、

30

前記第 2 の共有されるデータチャネルの中に、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第 2 の少なくとも 2 つのステージの各々によって作り出される第 2 のデータを記憶することと、

を行うようにさらに設定される、C 10 に記載の装置。

[ C 12 ]

前記 G P U は、

データが、前記グラフィックス処理パイプラインの 1 つ以上のステージによって消費されるために前記第 1 の共有されるデータチャネル、または前記第 2 の共有されるデータチャネルにおいて利用可能であり、ならびにフリースペースが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記 1 つ以上のステージによって作り出されるデータを記憶するために前記第 1 の共有されるデータチャネル、または前記第 2 の共有されるデータチャネルにおいて利用可能であるように、前記第 1 の共有されるデータチャネル、または前記第 2 の共有されるデータチャネルのステータスに少なくとも部分的に基づいて前記シェーダユニット上で前記グラフィックス処理パイプラインの前記 1 つ以上のステージの前記実行をスケジューリングするようにさらに構成される、C 11 に記載の装置。

40

[ C 13 ]

前記第 1 の共有されるデータチャネルは、前記第 1 の共有されるデータチャネルの中に記憶されたデータをキャッシュするためにキャッシュモードで動作し、前記第 2 の共有さ

50

れるデータチャネルは、ファーストイントファーストアウト（FIFO）モードで動作する、C11に記載の装置。

[C14]

前記第1の共有されるデータチャネルは、第1のリングバッファであり、前記第2の共有されるデータチャネルは、第2のリングバッファである、C11に記載の装置。

[C15]

前記第1の共有されるデータチャネルの中に記憶された前記データは、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記少なくとも2つのステージによって作り出される頂点データを備え、前記第2の共有されるデータチャネルの中に記憶された前記第2のデータは、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージによって作り出されるプリミティブを備える、C11に記載の装置。

10

[C16]

前記GPUは、

前記第1の共有されるデータチャネルと前記第2の共有されるデータチャネル間のデータロックを防ぐために、前記第1の共有されるデータチャネル、および前記第2の共有されるデータチャネルのうちの少なくとも1つの中のフリースペースを保持するようにさらに構成される、C11に記載の装置。

[C17]

装置であって、

20

グラフィックス処理パイプラインの少なくとも2つのステージによって共有されるグラフィックス処理ユニット（GPU）のオンチップグラフィックスメモリの中に共有されるデータチャネルを割り当てるための手段と、

前記グラフィックス処理パイプラインの前記少なくとも2つのステージを実行するための手段と、

オンチップグラフィックスメモリの中の前記共有されるデータチャネルの中に、前記グラフィックス処理パイプラインの前記少なくとも2つのステージの各々の実行によって作り出されるデータを記憶するための手段と、

を備える、装置。

[C18]

30

前記グラフィックス処理パイプラインの第2の少なくとも2つのステージによって共有される前記GPUのオンチップグラフィックスメモリの中に第2の共有されるデータチャネルを割り当てるための手段と、ここにおいて、前記共有されるデータチャネルは、第1の共有されるデータチャネルである、

前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージを実行するための手段と、

前記第2の共有されるデータチャネルの中に、前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージの各々の実行によって作り出される第2のデータを記憶するための手段と、

をさらに備える、C17に記載の装置。

40

[C19]

データが、前記グラフィックス処理パイプラインの1つ以上のステージによって消費されるために前記第1の共有されるデータチャネル、または前記第2の共有されるデータチャネルにおいて利用可能であり、ならびにフリースペースが、前記グラフィックス処理パイプラインの前記1つ以上のステージによって作り出されるデータを記憶するために前記第1の共有されるデータチャネル、または前記第2の共有されるデータチャネルにおいて利用可能であるように、前記第1の共有されるデータチャネル、または前記第2の共有されるデータチャネルのステータスに少なくとも部分的に基づいて前記グラフィックス処理パイプラインの前記1つ以上のステージの前記実行をスケジューリングする手段をさらに備える、C18に記載の装置。

50

## [ C 2 0 ]

前記第1の共有されるデータチャネルは、前記第1の共有されるデータチャネルの中に記憶されたデータをキャッシュするためにキャッシュモードで動作し、前記第2の共有されるデータチャネルは、ファーストインファーストアウト(FIFO)モードで動作する、C18に記載の装置。

## [ C 2 1 ]

前記第1の共有されるデータチャネルは、第1のリングバッファであり、前記第2の共有されるデータチャネルは、第2のリングバッファである、C18に記載の装置。

## [ C 2 2 ]

前記第1の共有されるデータチャネルの中に記憶された前記データは、前記グラフィックス処理パイプラインの前記少なくとも2つのステージによって作り出される頂点データを備え、前記第2の共有されるデータチャネルの中に記憶された前記第2のデータは、前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージによって作り出されるプリミティブを備える、C18に記載の装置。

10

## [ C 2 3 ]

前記第1の共有されるデータチャネルと前記第2の共有されるデータチャネル間のデッドロックを防ぐために、前記第1の共有されるデータチャネル、および前記第2の共有されるデータチャネルのうちの少なくとも1つの中のフリースペースを保持するための手段をさらに備える、C18に記載の装置。

## [ C 2 4 ]

20

命令を記憶するコンピュータ可読記憶媒体であって、前記命令は、実行されるとき、1つ以上のプログラマブルプロセッサに、

共有されるデータチャネルを、グラフィックス処理パイプラインの少なくとも2つのステージによって共有されるそのオンチップグラフィックスメモリの中に割り当てることと、

シェーダユニット上で、前記グラフィックス処理パイプラインの前記少なくとも2つのステージを実行することと、

オンチップグラフィックスメモリの中の前記共有されるデータチャネルの中に、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記少なくとも2つのステージの各々によって作り出されるデータを記憶することと、

30

を行わせる、コンピュータ可読記憶媒体。

## [ C 2 5 ]

前記命令は、実行されるとき、前記1つ以上のプログラマブルプロセッサに、

前記グラフィックス処理パイプラインの第2の少なくとも2つのステージによって共有されるオンチップグラフィックスメモリの中に第2の共有されるデータチャネルを割り当てることと、ここにおいて、前記共有されるデータチャネルは、第1の共有されるデータチャネルである、

前記シェーダユニット上で、前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージを実行することと、

前記第2の共有されるデータチャネルの中に、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージの各々によって作り出される第2のデータを記憶することと、

40

をさらに行わせる、C24に記載のコンピュータ可読記憶媒体。

## [ C 2 6 ]

前記命令は、実行されるとき、前記1つ以上のプログラマブルプロセッサに、

データが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの1つ以上のステージによって消費されるために前記第1の共有されるデータチャネル、または前記第2の共有されるデータチャネルにおいて利用可能であり、ならびにフリースペースが、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記1つ以上のステージによって作り出されるデータを記憶するために前記第1の共

50

有されるデータチャネル、または前記第2の共有されるデータチャネルにおいて利用可能であるように、前記第1の共有されるデータチャネル、または前記第2の共有されるデータチャネルのステータスに少なくとも部分的に基づいて前記シェーダユニット上で前記グラフィックス処理パイプラインの前記1つ以上のステージの前記実行をスケジューリングすることをさらに行わせる、C 25に記載のコンピュータ可読記憶媒体。

[C 27]

前記第1の共有されるデータチャネルは、前記第1の共有されるデータチャネルの中に記憶されたデータをキャッシュするためにキャッシュモードで動作し、前記第2の共有されるデータチャネルは、ファーストインファーストアウト(FIFO)モードで動作する、C 25に記載のコンピュータ可読記憶媒体。

10

[C 28]

前記第1の共有されるデータチャネルは、第1のリングバッファであり、前記第2の共有されるデータチャネルは、第2のリングバッファである、C 25に記載のコンピュータ可読記憶媒体。

[C 29]

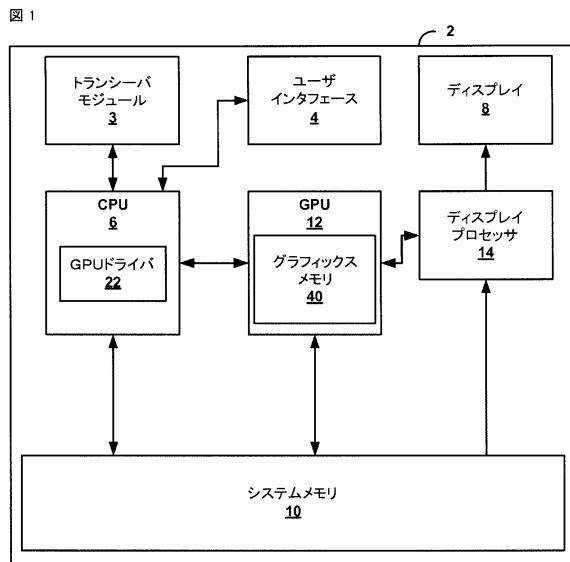
前記第1の共有されるデータチャネルの中に記憶された前記データは、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記少なくとも2つのステージによって作り出される頂点データを備え、前記第2の共有されるデータチャネルの中に記憶された前記第2のデータは、前記シェーダユニット上で実行している前記グラフィックス処理パイプラインの前記第2の少なくとも2つのステージによって作り出されるプリミティブを備える、C 25に記載のコンピュータ可読記憶媒体。

20

[C 30]

前記命令は、実行されるとき、前記1つ以上のプログラマブルプロセッサに、前記第1の共有されるデータチャネルと前記第2の共有されるデータチャネル間のデッドロックを防ぐために、前記第1の共有されるデータチャネル、および前記第2の共有されるデータチャネルのうちの少なくとも1つの中のフリースペースを保持することをさらに行わせる、C 25に記載のコンピュータ可読記憶媒体。

【 図 1 】



**FIG. 1**

【 図 2 】

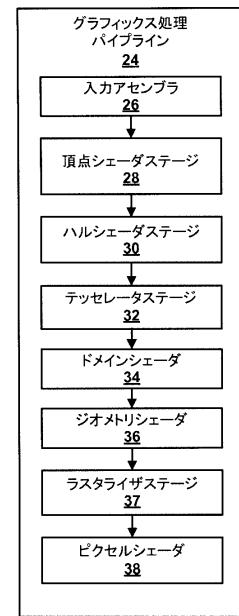


FIG. 2

【図3】

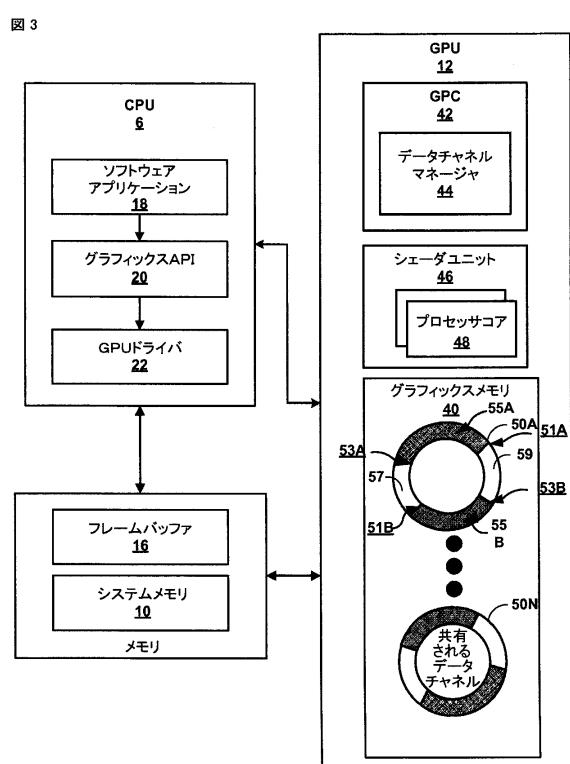


FIG. 3

【図4】

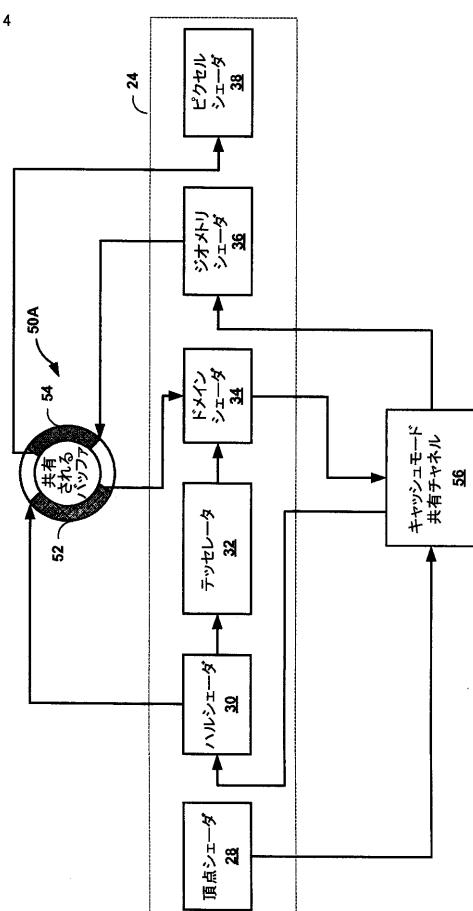
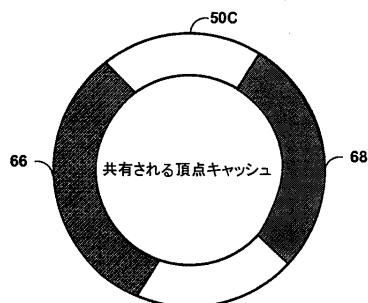
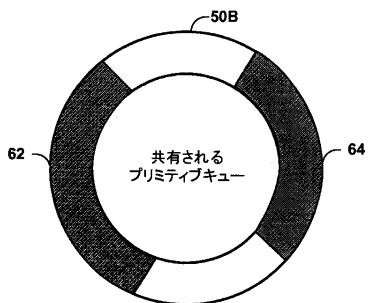


FIG. 4

【図5】

図5



V0	データロケーション	シェーダ
V1	データロケーション	シェーダ
⋮		
Vn	データロケーション	シェーダ

FIG. 5

【図6】

図6

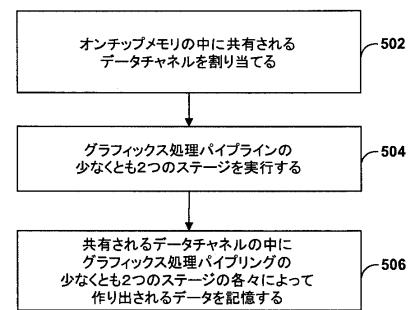


FIG. 6

---

フロントページの続き

(72)発明者 メイ、チュンヒ

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 ゴエル、ビネット

アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775

(72)発明者 キム、ドンヒュン

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 岡本 俊威

(56)参考文献 國際公開第2009/145889 (WO, A1)

米国特許出願公開第2013/0194286 (US, A1)

特表2012-514273 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06T 11/00 - 15/87

G06T 1/20