

(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.	F I
G06T 15/00 (2011.01)	G06T 15/00 501
G06T 1/20 (2006.01)	G06T 1/20 C

請求項の数 18 (全 30 頁)

(21) 出願番号	特願2016-569551 (P2016-569551)	(73) 特許権者	595020643
(86) (22) 出願日	平成27年1月26日 (2015. 1. 26)		クアルコム・インコーポレイテッド
(65) 公表番号	特表2017-509092 (P2017-509092A)		QUALCOMM INCORPORATED
(43) 公表日	平成29年3月30日 (2017. 3. 30)		アメリカ合衆国、カリフォルニア州 92
(86) 国際出願番号	PCT/US2015/012917		121-1714、サン・ディエゴ、モア
(87) 国際公開番号	W02015/126574		ハウス・ドライブ 5775
(87) 国際公開日	平成27年8月27日 (2015. 8. 27)	(74) 代理人	100108855
審査請求日	平成29年7月20日 (2017. 7. 20)		弁理士 蔵田 昌俊
(31) 優先権主張番号	14/182, 976	(74) 代理人	100109830
(32) 優先日	平成26年2月18日 (2014. 2. 18)		弁理士 福原 淑弘
(33) 優先権主張国	米国 (US)	(74) 代理人	100158805
早期審査対象出願			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って、

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 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GPU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

10

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GPU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、方法。

20

【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GPU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。

30

【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(FIFO)モードで動作する、請求項1に記載の方法。

【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(GPU)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、

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、

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。

【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

10

20

30

40

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のデータのキュ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11に記載の装置。

【請求項13】

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11に記載の装置。

【請求項14】

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11に記載の装置。

【請求項15】

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 1 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 1 5 に記載の非一時的コンピュータ可読記憶媒体。

【請求項 1 7】

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 5 に記載の非一時的コンピュータ可読記憶媒体。

【請求項 1 8】

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 1 5 に記載の非一時的コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【 0 0 0 1 】

[0001] 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。

【背景技術】

【 0 0 0 2 】

[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 (G P U) 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。

【 0 0 0 3 】

[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、G P U 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】

【 0 0 0 4 】

[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こと（allocating）を含み得る。方法は、GPUの中のシェードユニット上で、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、シェード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

【 0 0 0 5 】

[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ことと、GPUの中のシェードユニット上で、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、シェード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

【 0 0 0 6 】

[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。

【 0 0 0 7 】

[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ことと、シェードユニット上で、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、シェード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。

30

【 0 0 0 8 】

[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】

【 0 0 0 9 】

【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】

【 0 0 1 0 】

[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され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。

【 0 0 1 1 】

[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。

【 0 0 1 2 】

[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単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な、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得る。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おり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。

【 0 0 1 3 】

【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 s）、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（IC）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ICまたは複数のICs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ICの外部に存在するICの中に形成され得る。

10

【 0 0 1 4 】

【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ホン、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

【 0 0 1 5 】

【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

【 0 0 1 6 】

【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させ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、マイクロホン、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

【 0 0 1 7 】

[0022] CPU 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 CPU 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 (OpenGL (登録商標)) アプリケーションプログラミングインタフェース (API)、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) (OpenGL ES) API、Direct3D API、X3D API、RenderMan API、WebGL API、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 API (public or proprietary standard graphics API) のようなグラフィックス API にしたがいで得る。グラフィックスレンダリング命令を処理するために、GPU 12 は、GPU 12 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、GPU ドライバ 22 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 GPU 12 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 (triangle strips) 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。

10

【 0 0 1 8 】

[0023] GPU 12 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、CPU 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、CPU 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 GPU 12 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。GPU 12 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 CPU 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、GPU 12 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。GPU 12 の高度に並列な性質は、いくつかの例では、GPU 12 が、CPU 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、GUIs および 2 次元 (2D) および / または 3 次元 (3D) のグラフィックスシーン) を描写することを可能し得る。

20

30

【 0 0 1 9 】

[0024] GPU 12 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、GPU 12 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。GPU 12 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 (ASICs)、フィールドプログラマブルゲートアレイ (FPGAs)、デジタルシグナルプロセッサ (DSPs)、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。GPU 12 はまた、GPU 12 がマルチコアプロセッサと称され得るように、1 つ以上のプロセッサコアを含み得る。

40

【 0 0 2 0 】

[0025] GPU 12 は、グラフィックスメモリ 40 に直接結合され得る。したがって、GPU 12 は、バスを使用することなしにグラフィックスメモリ 40 からデータを読み取り、グラフィックスメモリ 40 にデータを書き込み得る。言い換えれば、GPU 12 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 40 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 GPU 12 の必要性を取り

50

除くことによって、GPU 12 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、GPU 12 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 10 を代わりに利用し得る。グラフィックスメモリ 40 は、例えば、ランダムアクセスメモリ (RAM)、スタティック RAM (SRAM)、ダイナミック RAM (DRAM)、消去可能なプログラマブル ROM (EPROM)、電氣的に消去可能なプログラマブル ROM (EEPROM (登録商標))、フラッシュメモリ、磁気データ媒体または光記憶媒体のような、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。

【0021】

[0026] いくつかの例では、GPU 12 は、システムメモリ 10 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 14 は、システムメモリ 10 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、GPU 12 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 (LCD)、有機発光ダイオードディスプレイ (OLED)、陰極線管 (CRT) ディスプレイ、プラズマディスプレイ、または別のタイプのディスプレイデバイスであり得る。

【0022】

[0027] 本開示の態様にしたがって、GPU 12 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。GPU 12 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 40 の中で共有されるデータチャネルを作り得る。GPU 12 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。

【0023】

[0028] 図 2 は、3 次元シーンの 2 次元表現を作るために GPU 12 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 24 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 24 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 24 は、入力アセンブラ 26、頂点シェーダステージ (vertex shader stage) 28、ハルシェーダステージ (hull shader stage) 30、テッセレータステージ (tessellator stage) 32、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 を含み得る。グラフィックス処理パイプライン 24 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。

【0024】

[0029] GPU 12 は、GPU ドライバ 22 を介して、CPU 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) を生成するためにグラフィックス処理パイプライン 24 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 24 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。

【0025】

[0030] 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 26 は、グラフィックス処理パイプライン 24 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、入力アセンブラステージ 26 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

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】

20

[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) に変換し (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) を生成することも担う。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。

30

【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。

40

【0029】

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パ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ンド (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。

【 0 0 3 4 】

[0039] 図 3 は、さらに詳細に図 1 の C P U 6、G P U 1 2、およびシステムメモリ 1 0 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、C P U 6 は、少なくとも 1 つのソフトウェアアプリケーション 1 8、グラフィックス A P I 2 0、および G P U ドライバ 2 2 を含むことができ、それらの各々は、C P U 6 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。

【 0 0 3 5 】

[0040] C P U 6 および G P U 1 2 に利用可能であるメモリは、システムメモリ 1 0 およびフレームバッファ 1 6 を含む得る。フレームバッファ 1 6 は、システムメモリ 1 0 の一部であることができ、またはシステムメモリ 1 0 から分離され得る。フレームバッファ 1 6 は、レンダリングされた画像データを記憶し得る。

10

【 0 0 3 6 】

[0041] ソフトウェアアプリケーション 1 8 は、G P U 1 2 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 1 8 は、G U I 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 2 D または 3 D 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。

20

【 0 0 3 7 】

[0042] ソフトウェアアプリケーション 1 8 は、グラフィカルユーザインタフェース (G U I) 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 G P U 1 2 に命令する 1 つ以上の描画命令を含む得る。例えば、描画命令は、G P U 1 2 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む得る。いくつかの例では、描画命令は、G U I 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散的に定義し得る。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散的に定義し得る。

30

【 0 0 3 8 】

[0043] ソフトウェアアプリケーション 1 8 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 G P U 1 2 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 A P I 2 0 を介して、G P U ドライバ 2 2 を呼び出し得る。例えば、ソフトウェアアプリケーション 1 8 は、G P U 1 2 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 A P I 2 0 を介して、G P U ドライバ 2 2 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 G P U 1 2 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) を含む得る。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む得る。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)、スケールリング情報、回転情報等も含み得る。ソフトウェアアプリケーション 1 8 によって G P U ドライバ 2 2 に発行される命令に基づいて、G P U ドライバ 2 2 は、プリミティブをレンダリングするために行うための G P U 1 2 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。G P U 1 2 が C P U 6 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 2 4 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

40

50

イン 2 4 を構成する。例えば、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 2 6 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 2 4 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 2 4 は、ディスプレイデバイスと関連するフレームバッファ 1 6 にレンダリングされたデータを出力する。

【 0 0 3 9 】

[0044] フレームバッファ 1 6 は、GPU 1 2 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 1 6 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 1 6 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 1 6 およびシステムメモリ 1 0 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 1 6 は、システムメモリ 1 0 の一部であり得る。

【 0 0 4 0 】

[0045] 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 の頂点シェーダステージ 2 8、ハルシェーダステージ 3 0、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、およびピクセルシェーダステージ 3 8 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、GPU 1 2 の中のシェーダユニット 4 6 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 4 6 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) 4 6 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 4 6 は、プロセッサコア 4 8 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。

【 0 0 4 1 】

[0046] GPU 1 2 は、グラフィックス処理パイプライン 2 4 の中の頂点シェーダステージ 2 8、ハルシェーダステージ 3 0、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 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 4 6 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 4 6 を指定し得る。いくつかの例では、GPU ドライバ 2 2 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 GPU 1 2 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、GPU 1 2 内のシェーダユニット 4 6 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 2 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頂点シェーダプログラム、ハルシェーダステージ 3 0 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ハルシェーダプログラム、ドメインシェーダステージ 3 4 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36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38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40は、GPU12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40がオンチップであるため、GPU12は、システムバスを介してシステムメモリ10から値を読み取ること、またはシステムメモリ10に値を書き込むことより速くグラフィックスメモリ40から値を読み取り、またはグラフィックスメモリ40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46のパフォーマンスは、グラフィックスメモリ40からグラフィックス処理パイプライン24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46は、プロセッサコア48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。GPU12は、異なるプロセッサコア48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24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46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24のステージをインタリーブする。例えば、GPU12は、シェーディングユニット46に、シェーダユニット46の異なるプロセッサコア48上で同時に頂点シェーダステージ28およびジオメトリシェーダステージ36を実行させるコマンドをシェーディングユニット46に送り得る。別の例では、GPU12は、シェーディングユニット46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36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46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40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24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50A-50N(「共有されるデータチャネル50」)のうちの1つ以上を含むことができ、それによってGPU12がグラフィックスメモリ40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46がグラフィックス処理パイプライン24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48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50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24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24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ことは対照的に、共有されるデータチャネル50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24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)42は、共有されるデータチャネル50のステータスに基づいてシェーダプロセッサクラスタ46の実行をスケジューリングし得る。GPC42は、シェーダプロセッサクラスタ46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24のステージによって消費されるた

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。GPC42はまた、シェーダプロセッサクラスタ46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24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。GPC42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、GPC42は、グラフィックス処理パイプライン24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46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46は、プロセッサクラスタ46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GPC42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）44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トをアップデートし得る。GPC42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44を含み得る。データチャネルマネージャ44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し示す共有されるデータチャネル50のための読み取りおよび書き込みポイントを管理し得る。

10

【0047】

[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50Aは、共有されるデータチャネル50Aがグラフィックス処理パイプライン24の第1のステージによって出力されたデータ55Aと、グラフィックス処理パイプライン24の第2のステージによって出力されたデータ55Bの両方を記憶し得るように、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50Aは、データ55Aおよび55B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50Aに割り当てられたメモリブロックのより効率的な使用を可能にする。GPC42は、書き込みポイント51Aおよび51B、ならびに読み取りポイント53Aおよび53Bを管理し得る。書き込みポイント51Aは、データ55Aを書き込むために、共有されるデータチャネル50Aのメモリロケーションを指し示すことができ、読み取りポイント53Aは、データ55Aを読み取るために、共有されるデータチャネル50Aのメモリロケーションを指し示し得る。

20

30

【0048】

[0053] 典型的に、読み取りポイント53Aおよび53B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55Aおよび55B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50Aのメモリロケーションを指し示すように、ならびに書き込みポイント51Aおよび51Bが、それぞれ、キューの末端（tail of the queue）と称されることもある、データ55Aおよび55B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50Aのメモリロケーションを指し示すように、GPU12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50Aの中にデータ55Aおよび55Bを記憶する。

40

【0049】

[0054] 共有されるデータチャネル50Aはまた、データ55Aおよび55Bから読み取られたデータが共有されるデータチャネル50A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）得るようにFIFOモードで動作し得る。以上のように、GPU12が共有されるデータチャネル50Aからデータ55A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57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

50

5 5 Bにデータを書き込むことを可能にする。同様に、GPU 12が共有されるデータチャンネル5 0 Aからデータ5 5 Bを読み取るとき、共有されるデータチャンネル5 0 Aの中のフリースペース5 9は増加し、それによってGPU 12のための共有されるデータチャンネル5 0 Aの中の追加のスペースがデータ5 5 A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ンネル5 0 A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ンネル5 0の中の各共有されるデータチャンネルが、共有されるデータチャンネル5 0 Aに関し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ンネル50の例を例示するブロック図である。図4に示されるように、共有されるデータチャンネル50 A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ンネル50 A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ンネル50 A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。データチャンネル50 A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ンネル50 Aから削除されるFIFOキューとして各々動作し得る。

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ンネル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、GPU 12は、キャッシュモード共有チャンネル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

50

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、キャッシュモード共有チャネル 5 6 の中でキャッシュされ得る。

【 0 0 5 4 】

[0059] 図 5 は、キャッシュ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、キャッシュ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 5 0 B および共有される頂点キャッシュ 5 0 C、ならびにキャッシュウィンドウ 7 0 を含み得る。共有される頂点キャッシュ 5 0 C は、共有される頂点キャッシュ 5 0 C に記憶されるデータが共有される頂点キャッシュ 5 0 C 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ュモードで動作し得る。共有されるプリミティブキュー 5 0 B に記憶されるデータ 6 2 およびデータ 6 4 は、頂点シェーダステージ 2 8 およびドメインシェーダステージ 3 4 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 6 2 は、各プリミティブのために頂点シェーダステージ 2 8 によって作り出された、共有される頂点キャッシュ 5 0 C 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 6 4 は、各プリミティブのためにドメインシェーダステージ 3 4 によって作り出された、共有される頂点キャッシュ 5 0 C 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 6 2 および 6 4 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ュ 5 0 C に記憶されたデータ 6 6 は、頂点シェーダステージ 2 8 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ュ 5 0 C に記憶されたデータ 6 8 は、ドメインシェーダステージ 3 4 によって出力された頂点位置を含み得る。GPC 4 2 は、キャッシュ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 5 0 B と共有される頂点キャッシュ 5 0 C の両方のフリースペースをチェックし得る。

【 0 0 5 5 】

[0060] キャッシュウィンドウ 7 0 は、特定の頂点が、共有される頂点キャッシュ 5 0 C の限られたウィ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、キャッシュウィンドウ 7 0 は、フリーアソシアティブキャッシュ (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ュ 5 0 C 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。

【 0 0 5 6 】

[0061] プリミティブによる GPC 4 2 処理ジオメトリプリミティブ。頂点シェーダ 2 8 およびドメインシェーダ 3 4 のために、GPC 4 2 が、頂点インデックス上のキャッシュウィンドウ 7 0、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ュ 5 0 C の中に存在しないことを決定する場合、キャッシュミスが発生することがあり得、そして GPC 4 2 は、所望の頂点を作り出すために、およびキャッシュ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 2 8 またはドメインシェーダ 3 4) を実行するためのコマンドをシェーダユニット 4 6 に送り得る。GPC 4 2 は、共有される頂点キャッシュ 5 0 C 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 5 0 B に追加し得る。GPC 4 2 は、キャッシュモード共有チャネル 5 6 の中の今キャッシュされた頂点に関する適切なデータをキャッシュウィンドウ 7 0 に追加し得る。キャッシュウィンドウ 7 0 は、キャッシュミスの後にキャッシュウィンドウ 7 0 の中に余地がない場合、キャッシュウィンドウ 7 0 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 5 0 B の中にセットされたその割り当て解除フラグを有する頂点が、キャッシュ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 (FIFO) の方法で動作し得る。しかしながら、GPC 4 2 が特定の頂点がキャッシュモード共有チャネル 5 6 の中でキャッシュされることを決定する場合、GPC 4 2 は、所望の

頂点の共有される頂点キャッシュ 50C 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、GPU 12 は、グラフィックス処理パイプライン 24 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。

【0057】

[0062] ハルシェーダ 30 およびジオメトリシェーダ 36 を実行するために、GPC 42 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。GPC 42 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 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。GPC 42 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。GPC 42 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、GPC 42 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 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、GPC 42 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ンネル 56 から頂点を割り当て解除することを可能にされ得る。GPC 42 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 30 およびジオメトリシェーダ 36) 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ンネル 50A 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 46 に送り得る。

【0058】

[0063] GPC 42 は、デッドロックのためのキャッシュモード共有チャンネル 56 および共有されるデータチャンネル 50A をモニタし得る。一例では、キャッシュモード共有チャンネル 56 が頂点シェーダステージ 28 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ンネル 50A がハルシェーダステージ 30 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。この場合では、ハルシェーダステージ 30 が頂点ステージ 28 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 30 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ンネル 50A 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ンネル 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 28 によって作り出され、キャッシュモード共有チャンネル 56 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ンネル 56 が、頂点シェーダステージ 28 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 30 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 34 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ンネル 56 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 34 がハルシェーダステージ 30 によって作り出され、共有されるデータチャンネル 50A 中に記憶されるデータを消費するため、ハルシェーダ 30 A によって作り出され、共有されるデータチャンネル 50A 中に記憶されるデータのいずれも、ジオメトリシェーダ 36 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ンネル 50A に関する、共有されるデータチャンネル 50A 中のスペースを解放するためにドメインシェーダ 34 によって消費されることができない。

【0059】

[0064] キャッシュモード共有チャンネル 56 と共有されるデータチャンネル 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ンネル 56 および共有されるデータチャンネル 50A がそれぞれ、頂点シェーダ 28 およびハルシェーダ 30 によって作り出

されるデータを記憶するだけでないように、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ンネル 5 6 および共有されるデータチャンネル 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ンネル 5 6 のスペースの量、および例えば、シェーダクラス 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、保持すべき共有されるデータチャンネル 5 0 A のスペースの量を決定し得る。

【 0 0 6 0 】

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ン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ンネル 5 0 A を割り当てることを含み得る (5 0 2)。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 (5 0 4)。プロセスは、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ンネル 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 (5 0 6)。

20

【 0 0 6 1 】

[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ンネル 5 6 を割り当てること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ンネル 5 0 A は、第 1 の共有されるデータチャンネルである。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ンネル 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。

30

【 0 0 6 2 】

[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ンネル 5 0 A、または第 2 のキャッシュモード共有チャンネル 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ンネル 5 0 A、または第 2 のキャッシュモード共有チャンネル 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ンネル 5 0 A または第 2 のキャッシュモード共有チャンネル 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。

40

【 0 0 6 3 】

[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。

【 0 0 6 4 】

[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50A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50A、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せられ得るか、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。

【 0 0 6 8 】

[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 (G P U) 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 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、

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 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 G P U 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 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、

前記第2の共有されるデータチャネルの中に前記 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 G P U 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 (F I F O) 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

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 GPU 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 (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、

20

を行うように構成される前記 GPU を備える、装置。

[C 11]

前記 GPU 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 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、

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 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、ファーストインファーストアウト（FIFO）モードで動作する、C 1 1 に記載の装置。

[C 1 4]

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 1 1 に記載の装置。

[C 1 5]

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 1 1 に記載の装置。

10

[C 1 6]

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ようにさらに構成される、C 1 1 に記載の装置。

[C 1 7]

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（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、

20

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。

[C 1 8]

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、

30

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、C 1 7 に記載の装置。

40

[C 1 9]

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、C 1 8 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 (F I F O) モードで動作する、C 1 8 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、C 1 8 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、C 1 8 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、C 1 8 に記載の装置。

[C 2 4]

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、

20

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、C 2 4 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 2 5 に記載のコンピュータ可読記憶媒体。

[C 2 7]

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 (F I F O) モードで動作する、C 2 5 に記載のコンピュータ可読記憶媒体。

10

[C 2 8]

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 5 に記載のコンピュータ可読記憶媒体。

[C 2 9]

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 2 5 に記載のコンピュータ可読記憶媒体。

20

[C 3 0]

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 2 5 に記載のコンピュータ可読記憶媒体。

【図 1】

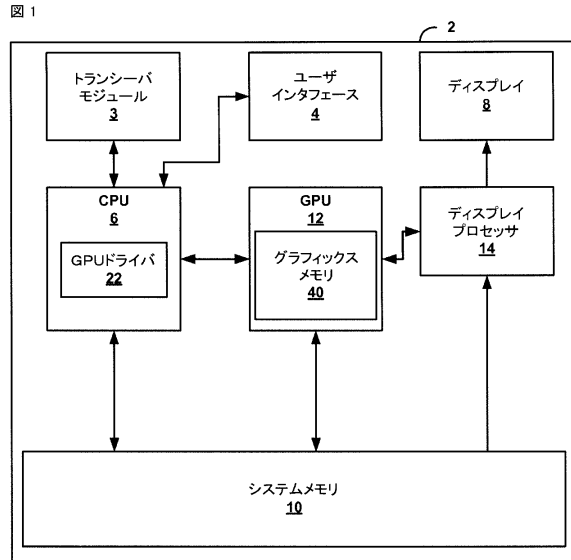


FIG. 1

【図 2】

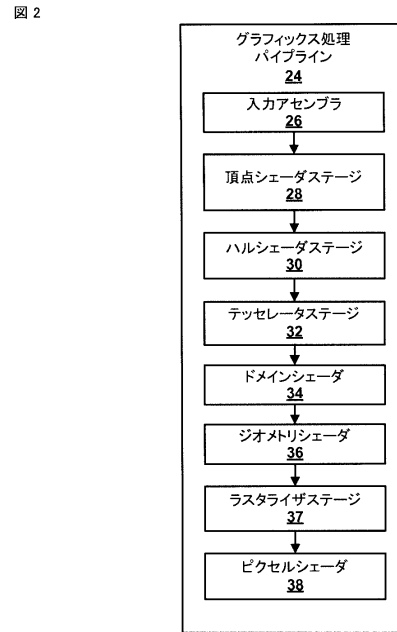


FIG. 2

【図 3】

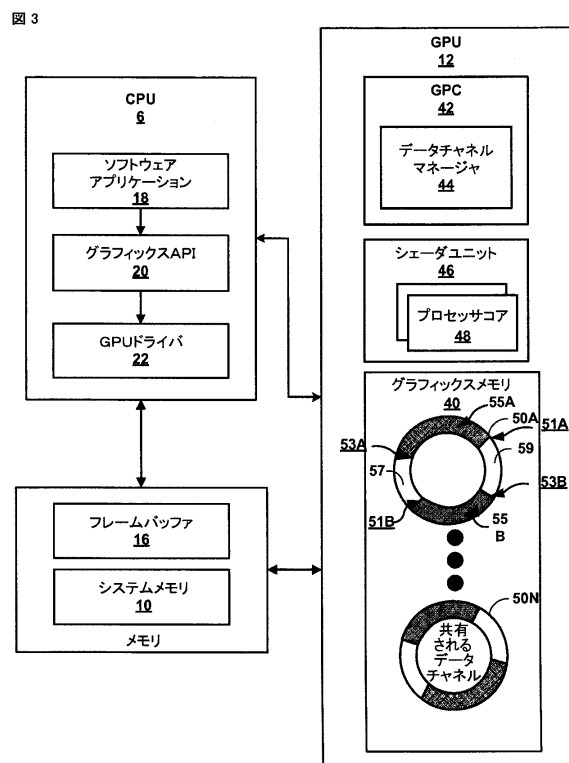


FIG. 3

【図 4】

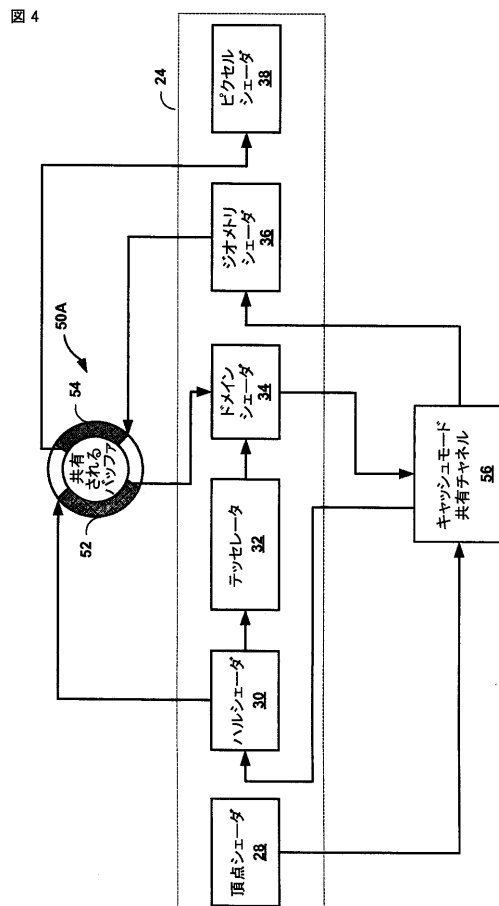


FIG. 4

【図 5】

図 5

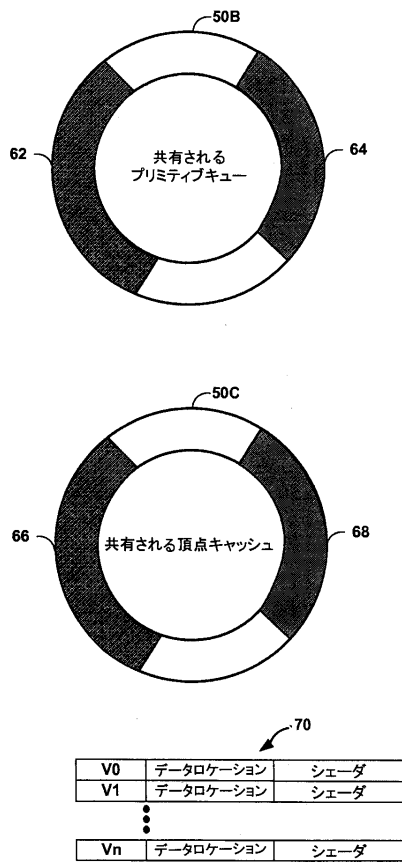


FIG. 5

【図 6】

図 6

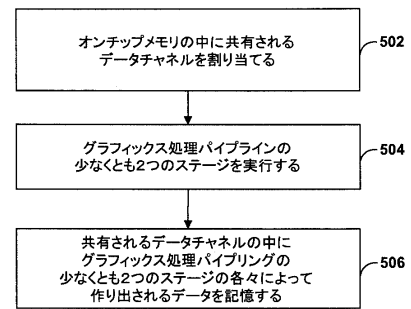


FIG. 6

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 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドラ
イブ 5 7 7 5
- (72)発明者 ゴエル、ビネート
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドラ
イブ 5 7 7 5
- (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 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドラ
イブ 5 7 7 5

審査官 岡本 俊威

- (56)参考文献 国際公開第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)
G 0 6 T 1 1 / 0 0 - 1 5 / 8 7
G 0 6 T 1 / 2 0