

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7423644号
(P7423644)

(45)発行日 令和6年1月29日(2024.1.29)

(24)登録日 令和6年1月19日(2024.1.19)

(51)国際特許分類		F I		
G 0 6 T	15/00 (2011.01)	G 0 6 T	15/00	5 0 1
G 0 6 F	9/38 (2018.01)	G 0 6 F	9/38	3 7 0 C
G 0 6 F	17/16 (2006.01)	G 0 6 F	17/16	P
G 0 6 F	17/10 (2006.01)	G 0 6 F	17/10	A
請求項の数 20 (全111頁)				
(21)出願番号	特願2021-547452(P2021-547452)	(73)特許権者	593096712	
(86)(22)出願日	令和2年3月14日(2020.3.14)		インテル コーポレーション	
(65)公表番号	特表2022-523762(P2022-523762 A)		アメリカ合衆国 9 5 0 5 4 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2 2 0 0	
(43)公表日	令和4年4月26日(2022.4.26)	(74)代理人	100107766	
(86)国際出願番号	PCT/US2020/022846		弁理士 伊東 忠重	
(87)国際公開番号	WO2020/190808	(74)代理人	100070150	
(87)国際公開日	令和2年9月24日(2020.9.24)		弁理士 伊東 忠彦	
審査請求日	令和4年7月5日(2022.7.5)	(74)代理人	100135079	
(31)優先権主張番号	62/819,435		弁理士 宮崎 修	
(32)優先日	平成31年3月15日(2019.3.15)	(72)発明者	レイ, ジョイディーブ	
(33)優先権主張国・地域又は機関	米国(US)		アメリカ合衆国 9 5 0 5 4 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2 2 0 0	
(31)優先権主張番号	62/819,337		インテル	
(32)優先日	平成31年3月15日(2019.3.15)		最終頁に続く	最終頁に続く
最終頁に続く				

(54)【発明の名称】 行列アクセラレータアーキテクチャのためのスパース最適化

(57)【特許請求の範囲】
【請求項1】

キャッシュメモリと結合される複数の処理リソースを含み、少なくとも1つの処理リソースが行列アクセラレータを含み、該行列アクセラレータは、第1行列及び第2行列の複数の要素に対して行列乗算演算を実行するよう構成され、第1行列はスパース行列であり、前記第1行列は、前記スパース行列においてゼロ値要素のみを含むか又は限られた数の非ゼロ値要素しか含まない少なくとも1つのスパースサブ行列のロードがバイパスされることを可能にすることによって、スパース符号化に符号化される、計算クラスタを有し、前記スパース符号化は、前記キャッシュメモリに格納され、前記キャッシュメモリは、該キャッシュメモリへの書き込みに関連して前記スパース符号化を圧縮表現に圧縮する圧縮器と結合し、前記圧縮表現は、非ゼロ値要素の組と、該非ゼロ値要素の位置を特定するメタデータとを含み、前記少なくとも1つの処理リソースは、前記圧縮表現を前記キャッシュメモリからロードし、前記圧縮表現を前記少なくとも1つの処理リソース内のメモリに格納し、前記第2行列の選択された要素を前記キャッシュメモリからロードし、前記選択された要素を前記少なくとも1つの処理リソース内の前記メモリに格納し、前記第2行列の前記選択された要素が、前記圧縮表現内に格納された前記第1行列の非ゼロ値と対応し、該非ゼロ値の位置を示す前記メタデータに基づき選択され、前記圧縮表現からの要素及び前記第2行列の前記選択された要素に対して前記行列乗算

演算を実行し、

前記行列乗算演算の出力を前記少なくとも1つの処理リソース内の前記メモリに書き込むよう構成される、
汎用グラフィクスプロセッサ。

【請求項2】

前記キャッシュメモリは、レベル2（L2）キャッシュメモリである、
請求項1に記載の汎用グラフィクスプロセッサ。

【請求項3】

前記圧縮器は、前記L2キャッシュメモリに格納されているデータを圧縮する、
請求項2に記載の汎用グラフィクスプロセッサ。

10

【請求項4】

前記処理リソース内の前記メモリは、レベル1（L1）キャッシュメモリを含む、
請求項1に記載の汎用グラフィクスプロセッサ。

【請求項5】

前記処理リソース内の前記メモリは、レジスタファイル（RF）を含む、
請求項1に記載の汎用グラフィクスプロセッサ。

【請求項6】

前記処理リソース内の前記メモリは、前記行列アクセラレータ内のメモリを含む、
請求項1に記載の汎用グラフィクスプロセッサ。

【請求項7】

前記第1行列は、ニューラルネットワークに関連した重みデータを含む、
請求項1に記載の汎用グラフィクスプロセッサ。

20

【請求項8】

前記第2行列は、前記ニューラルネットワークに関連した入力活性化データを含む、
請求項7に記載の汎用グラフィクスプロセッサ。

【請求項9】

前記行列乗算演算の前記出力は、前記ニューラルネットワークに関連した出力活性化データを含む、
請求項8に記載の汎用グラフィクスプロセッサ。

【請求項10】

前記行列乗算演算の前記出力は、密行列である、
請求項9に記載の汎用グラフィクスプロセッサ。

30

【請求項11】

前記行列アクセラレータは、処理要素のシストリックアレイを含む、
請求項1に記載の汎用グラフィクスプロセッサ。

【請求項12】

メモリデバイスと、
請求項1乃至11のうち何れか一項に記載の汎用グラフィクスプロセッサとを有するデータ処理システム。

【請求項13】

キャッシュメモリと結合される複数の処理リソースを含む計算クラスタにより、第1行列及び第2行列の複数の要素に対して行列乗算演算を実行することであり、少なくとも1つの処理リソースが行列アクセラレータを含み、第1行列はスパース行列であり、前記第1行列は、前記スパース行列においてゼロ値要素のみを含むか又は限られた数の非ゼロ値要素しか含まない少なくとも1つのスパースサブ行列のロードがバイパスされることを可能にすることによって、スパース符号化に符号化される、前記実行することと、

40

前記スパース符号化を前記キャッシュメモリに格納することであり、前記キャッシュメモリは、該キャッシュメモリ内の前記スパース符号化を圧縮表現に圧縮する圧縮器と結合され、前記圧縮表現は、非ゼロ値要素の組と、該非ゼロ値要素の位置を特定するメタデータとを含む、前記格納することと、

50

前記少なくとも1つの処理リソースにより、
前記圧縮表現を前記キャッシュメモリからロードし、
前記圧縮表現を前記少なくとも1つの処理リソース内のメモリに格納し、
前記第2行列の選択された要素を前記キャッシュメモリからロードし、
前記選択された要素を前記少なくとも1つの処理リソース内の前記メモリに格納し、
前記第2行列の前記選択された要素が、前記圧縮表現内に格納された前記第1行列の非ゼロ値と対応し、該非ゼロ値の位置を示す前記メタデータに基づき選択され、
前記圧縮表現からの要素及び前記第2行列の前記選択された要素に対して前記行列乗算演算を実行し、
前記行列乗算演算の出力を前記少なくとも1つの処理リソース内の前記メモリに書き込むことと
を有する方法。

【請求項14】

前記キャッシュメモリは、レベル2（L2）キャッシュメモリであり、前記圧縮器は、前記L2キャッシュメモリに格納されているデータを圧縮する、
請求項13に記載の方法。

【請求項15】

前記処理リソース内の前記メモリは、レベル1（L1）キャッシュメモリを含む、
請求項13に記載の方法。

【請求項16】

前記処理リソース内の前記メモリは、レジスタファイル（RF）を含む、
請求項13に記載の方法。

【請求項17】

前記処理リソース内の前記メモリは、前記行列アクセラレータ内のメモリを含む、
請求項13に記載の方法。

【請求項18】

実行される場合にマシンに請求項13乃至17のうち何れか一項に記載の方法を実行させるコンピュータプログラム。

【請求項19】

請求項18に記載のコンピュータプログラムを記憶しているマシン可読記憶媒体。

【請求項20】

請求項13乃至17のうち何れか一項に記載の方法を実行する手段を有する装置。

【発明の詳細な説明】

【技術分野】

【0001】

【関連出願の相互参照】

本願は、35 U. S. C. 119（e）の下で、2019年3月15日付けでAbhishek Appu等によって「GRAPHICS PROCESSING」と題されて出願された米国特許仮出願第62／819337号（代理人整理番号AC0271-Z）と、2019年3月15日付けでLakshminarayanan Striramassarma等によって「GRAPHICS PROCESSING」と題されて出願された米国特許仮出願第62／819435号（代理人整理番号AC0285-Z）と、2019年3月15日付けでSubramaniam Maiyuran等によって「SYSTEMS AND METHODS FOR PARTITIONING CACHE TO REDUCE CACHE ACCESS LATENCY」と題されて出願された米国特許仮出願第62／819361号（代理人整理番号AC0286-Z）と、2019年11月15日付けでAbhishek Appu等によって「SYSTOLIC ARITHMETIC ON SPARSE DATA」と題されて出願された米国特許仮出願第62／935670号（代理人整理番号AC5197-Z）とに基づく優先権の利益を主張する。これらの先の米国特許出願は、その全文を参照により本願に援用される。

【0002】

【分野】

本願は、概して、データ処理に関係があり、より具体的には、汎用グラフィクス処理ユニットの行列アクセラレータによるスパースデータ処理に関係がある。

【背景技術】

【0003】

現在の並列グラフィクスデータ処理は、例えば、線形補間、テッセレーション、ラスタライズ、テクスチャマッピング、デプステスト、などのような特定の動作をグラフィクスデータに対して実行するために開発されたシステム及び方法を含む。従来、グラフィクスプロセッサは、グラフィクスデータを処理するために固定機能計算ユニットを使用した。より最近では、グラフィクスプロセッサの部分がプログラム可能になっており、頂点及びフラグメントデータを処理するためのより様々な動作をサポートすることをそのようなプロセッサに可能にする。

10

【0004】

性能を更に向上させるために、グラフィクスプロセッサは、通常は、グラフィクスパイプラインの異なる部分にわたって可能な限り多くのグラフィクスデータを並列に処理しようと試みるパイプライン化などの処理技術を実装する。SIMT (single instruction, multiple thread) アーキテクチャを備えた並列グラフィクスプロセッサは、グラフィクスパイプラインにおける並列処理の量を最大限にするよう設計される。SIMT アーキテクチャにおいて、並列スレッドのグループは、処理効率を高めるよう可能な限り頻繁にプログラム命令を同時に一緒に実行しようと試みる。SIMT アーキテクチャのためのソフトウェア及びハードウェアの概要は、Shane Cook, CUDA Programming Chapter 3, pages 37-51 (2013)で見つけられ得る。

20

【図面の簡単な説明】

【0005】

本実施形態の上記の特徴が詳細に理解され得るように、先に簡潔に要約された実施形態は、実施形態を参照して、より具体的に説明され、実施形態のいくつかは、添付の図面に表される。なお、添付の図面は、典型的な実施形態のみを表しているもので、その範囲の限定と見なされるべきではない、ことが留意されるべきである。

【0006】

【図1】本願で記載される実施形態の1つ以上の態様を実装するよう構成されたコンピュータシステムを表すブロック図である。

30

【図2A】並列プロセッサコンポーネントを表す。

【図2B】並列プロセッサコンポーネントを表す。

【図2C】並列プロセッサコンポーネントを表す。

【図2D】並列プロセッサコンポーネントを表す。

【図3A】グラフィクスマルチプロセッサ及びマルチブロックベースGPUのブロック図である。

【図3B】グラフィクスマルチプロセッサ及びマルチブロックベースGPUのブロック図である。

【図3C】グラフィクスマルチプロセッサ及びマルチブロックベースGPUのブロック図である。

40

【図4A】複数のGPUが複数のマルチコアプロセッサへ通信可能に結合されている例示的なアーキテクチャを表す。

【図4B】複数のGPUが複数のマルチコアプロセッサへ通信可能に結合されている例示的なアーキテクチャを表す。

【図4C】複数のGPUが複数のマルチコアプロセッサへ通信可能に結合されている例示的なアーキテクチャを表す。

【図4D】複数のGPUが複数のマルチコアプロセッサへ通信可能に結合されている例示的なアーキテクチャを表す。

【図4E】複数のGPUが複数のマルチコアプロセッサへ通信可能に結合されている例示的なアーキテクチャを表す。

50

【図 4 F】複数の GPU が複数のマルチコアプロセッサへ通信可能に結合されている例示的なアーキテクチャを表す。

【図 5】グラフィクス処理パイプラインを表す。

【図 6】機械学習ソフトウェアスタックを表す。

【図 7】汎用グラフィクス処理ユニットを表す。

【図 8】マルチ GPU コンピューティングシステムを表す。

【図 9 A】例示的なディープニューラルネットワークのレイヤを表す。

【図 9 B】例示的なディープニューラルネットワークのレイヤを表す。

【図 10】例示的な回帰型ニューラルネットワークを表す。

【図 11】ディープニューラルネットワークの訓練及びデプロイを表す。

10

【図 12】分散学習を表すブロック図である。

【図 13】訓練されたモデルを用いて推定を行うのに適した例示的な推定 SOC (system on chip) を表す。

【図 14】実施形態に従う処理システムのブロック図である。

【図 15 A】コンピューティングシステム及びグラフィクスプロセッサを表す。

【図 15 B】コンピューティングシステム及びグラフィクスプロセッサを表す。

【図 15 C】コンピューティングシステム及びグラフィクスプロセッサを表す。

【図 16 A】更なるグラフィクスプロセッサ及び計算アクセラレータアーキテクチャのブロック図を表す。

【図 16 B】更なるグラフィクスプロセッサ及び計算アクセラレータアーキテクチャのブロック図を表す。

20

【図 16 C】更なるグラフィクスプロセッサ及び計算アクセラレータアーキテクチャのブロック図を表す。

【図 17】グラフィクスプロセッサのグラフィクス処理エンジンのブロック図である。

【図 18 A】グラフィクスプロセッサで用いられる処理要素のアレイを含むスレッド実行ロジックを表す。

【図 18 B】グラフィクスプロセッサで用いられる処理要素のアレイを含むスレッド実行ロジックを表す。

【図 19】更なる実行ユニットを表す。

【図 20】グラフィクスプロセッサ命令フォーマットを表すブロック図である。

30

【図 21】更なるグラフィクスプロセッサアーキテクチャのブロック図である。

【図 22 A】グラフィクスプロセッサコマンドフォーマット及びフォーマットシーケンスを表す。

【図 22 B】グラフィクスプロセッサコマンドフォーマット及びフォーマットシーケンスを表す。

【図 23】データ処理システムのための例示的なグラフィクスソフトウェアアーキテクチャを表す。

【図 24 A】IP コア開発システムを表すブロック図である。

【図 24 B】集積回路パッケージアセンブリの断面側面図を表す。

【図 24 C】基板（例えば、ベースダイ）へ接続されたハードウェアロジックチップレットの複数のユニットを含むパッケージアセンブリを表す。

40

【図 24 D】交換可能なチップレットを含むパッケージアセンブリを表す。

【図 25】チップ集積回路上の例示的なシステムを表すブロック図である。

【図 26 A】SOC 内で使用される例示的なグラフィクスプロセッサを表すブロック図である。

【図 26 B】SOC 内で使用される例示的なグラフィクスプロセッサを表すブロック図である。

【図 27】実施形態に従う更なる実行ユニットを表す。

【図 28】実施形態に従って、命令パイプラインによって実行される行列演算を表す。

【図 29 A】いくつかの実施形態に従うハードウェアベースのシストリックアレイの詳細

50

を表す。

【図 29 B】いくつかの実施形態に従うハードウェアベースのシストリックアレイの詳細を表す。

【図 30 A】実施形態に従って、ゼロ値サブ行列をバイパスするシステムを表す。

【図 30 B】実施形態に従って、ゼロ値サブ行列をバイパスするシステムを表す。

【図 31 A】スパースデータに対して行列乗算演算を実行する方法及びシステムを表す。

【図 31 B】スパースデータに対して行列乗算演算を実行する方法及びシステムを表す。

【図 32】G P G P U データ圧縮パイプラインを含むグラフィクス処理システムのブロック図である。

【図 33 A】実施形態に従って、並列計算プロセッサ又は汎用グラフィクス処理ユニットで処理リソースへの圧縮されたデータ伝送を可能にするよう構成された計算アーキテクチャを表す。

10

【図 33 B】実施形態に従って、並列計算プロセッサ又は汎用グラフィクス処理ユニットで処理リソースへの圧縮されたデータ伝送を可能にするよう構成された計算アーキテクチャを表す。

【図 34】出力データに対するゼロ検出ロジックを含む処理リソースを表す。

【図 35】処理リソース内の行列演算のためにデータを圧縮及び圧縮解除する方法を表す。

【図 36】処理リソースによって生成された出力に対してゼロ検出動作を実行する方法を表す。

【図 37】実施形態に従うグラフィクスプロセッサを含むコンピューティングデバイスのブロック図である。

20

【発明を実施するための形態】

【0007】

グラフィクス処理ユニット（GPU）は、例えば、グラフィクス操作、機械学習動作、パターン解析動作、及び／又は様々な汎用 GPU（GP GPU）機能を加速させるようホスト／プロセッサコアへ通信可能に結合されている。GPU は、バス又は他のインターコネクト（例えば、PCI e 又は NV L i n k などの高速インターコネクト）を介してホストプロセッサ／コアへ通信可能に結合されてよい。代替的に、GPU は、コアと同じパッケージ又はチップに集積され、内部プロセッサバス／インターコネクト（すなわち、パッケージ又はチップ内にある）を介してコアへ通信可能に結合されてもよい。GPU が接続されている状態にかかわらず、プロセッサコアは、作業記述子に含まれているコマンド／命令のシーケンスの形で作業を GPU に割り当ててよい。次いで、GPU は、これらのコマンド／命令を効率的に処理するための専用の回路構成／ロジックを使用する。

30

【0008】

本願で記載されている実施形態は、シストリック（systolic）処理ユニットによりスパースデータに対して算術を実行する技術を提供するソフトウェア、ファームウェア、及びハードウェアロジックを含む。本願で記載されている実施形態は、ゼロ充填行列及びサブ行列に対する計算動作をスキップする技術を提供する。実施形態は更に、処理ユニットまでデータ圧縮を維持する技術を提供する。実施形態は更に、スパースウェアロジックユニットのためのアーキテクチャを提供する。

40

【0009】

以下の記載では、多数の具体的な詳細が、より完全な理解をもたらすよう示されている。しかし、当業者に明らかなように、ここで記載される実施形態は、これらの具体的な詳細の 1 つ以上によらずとも実施されてよい。他の事例では、よく知られている特徴は、本実施形態の詳細を不明りようにしないように記載されていない。

【0010】

〔システム概要〕

図 1 は、本願で記載される実施形態の 1 つ以上の態様を実装するよう構成されたコンピューティングシステム 100 を表すブロック図である。コンピューティングシステム 100 は、インターコネクションバスを介して通信する 1 つ以上のプロセッサ 102 及びシス

50

テムメモリ104を備えている処理サブシステム101を含む。インターコネクションパスは、メモリハブ105を含んでよい。メモリハブ105は、チップセットコンポーネント内の別個のコンポーネントであってよく、あるいは、1つ以上のプロセッサ102内に組み込まれてもよい。メモリハブ105は、通信リンク106を介してI/Oサブシステム111と結合する。I/Oサブシステム111は、1つ以上の入力デバイス108から入力を受けることをコンピューティングシステム100に可能にすることができるI/Oハブ107を含む。更には、I/Oハブ107は、1つ以上のプロセッサ102に含まれるディスプレイコントローラが1つ以上の表示デバイス110Aへ出力を供給することを可能にすることができる。一実施形態で、I/Oハブ107と結合されている1つ以上の表示デバイス110Aは、ローカル、内蔵、又は埋め込み表示デバイスを含むことができる。

10

【0011】

処理サブシステム101は、例えば、バス又は他の通信リンク113を介してメモリハブ105へ結合されている1つ以上の並列プロセッサ112を含む。通信リンク113は、例えば、PCI Expressなどの、しかしこれに限られない任意の数の規格に基づいた通信リンク技術又はプロトコルの1つであってよく、あるいは、ベンダー固有の通信インターフェース又は通信ファブリックであってもよい。1つ以上の並列プロセッサ112は、MIC (many integrated core) プロセッサなどの多数の処理コア及び／又は処理クラスタを含むことができる計算的に集束した並列又はベクトル処理システムを形成してよい。例えば、1つ以上の並列プロセッサ112は、I/Oハブ107を介して結合されている1つ以上の表示デバイス110Aの中の1つへピクセルを出力することができるグラフィクス処理システムを形成する。1つ以上の並列プロセッサ112はまた、1つ以上の表示デバイス110Bへの直接接続を可能にするディスプレイコントローラ及びディスプレイインターフェース（図示せず。）も含むことができる。

20

【0012】

I/Oサブシステム111内で、システム記憶ユニット114は、コンピューティングシステム100のための記憶メカニズムを提供するようI/Oサブシステム111へ接続可能である。I/Oスイッチ116は、I/Oハブ107と、プラットフォームに集積され得るネットワークアダプタ118及び／又は無線ネットワークアダプタ119などの他のコンポーネント、及び1つ以上のアドインデバイス120を介して追加可能な様々な他のデバイスとの間の接続を可能にするインターフェースメカニズムを提供するために使用され得る。アドインデバイス120はまた、例えば、1つ以上の外部グラフィクスプロセッサデバイス及び／又は計算アクセラレータを含んでもよい。ネットワークアダプタ118は、Ethernet（登録商標）アダプタ又は他の有線ネットワークアダプタであることができる。無線ネットワークアダプタ119は、Wi-Fi、Bluetooth（登録商標）、近距離通信（NFC）、又は1つ以上の無線ラジオを含む他のネットワークデバイスのうちの1つ以上を含むことができる。

30

【0013】

コンピューティングシステム100は、USB又は他のポート接続を含む、明示されていない他のコンポーネントを含むことができ、光学記憶ドライバ、ビデオ捕捉デバイス、なども、I/Oハブ107へ接続されてよい。図1で様々なコンポーネントを相互接続する通信パスは、PCI (Peripheral Component Interconnect) に基づくプロトコル（例えば、PCI-Express）などの任意の適切なプロトコル、あるいは、NVLink高速インターコネクト又は当該技術で知られているインターコネクトプロトコルなどの任意の他のバス又はポイント・ツー・ポイント通信インターフェース及び／又はプロトコルを用いて実装されてもよい。

40

【0014】

1つ以上の並列プロセッサ112は、例えば、ビデオ出力回路構成を含む、グラフィクス及びビデオ処理のために最適化された回路構成を組み込んでよく、グラフィクス処理ユニット（GPU）を構成する。代替的に、又は追加的に、1つ以上の並列プロセッサ11

50

2は、本願で更に詳細に記載されるように、基礎を成す計算アーキテクチャを保持しながら、汎用処理のために最適化された回路構成を組み込むことができる。コンピューティングシステム100のコンポーネントは、単一の集積回路上で1つ以上の他のシステム要素と集積されてよい。例えば、1つ以上の並列プロセッサ112、メモリハブ105、プロセッサ102、及びI/Oハブ107は、システム・オン・チップ（SoC）集積回路に組み込まれ得る。代替的に、コンピューティングシステム100のコンポーネントは、システム・イン・パッケージ（SIP）構成を形成するよう単一のパッケージに組み込まれ得る。一実施形態で、コンピューティングシステム100のコンポーネントの少なくとも一部は、モジュールコンピューティングシステム内に他のマルチチップモジュールと相互接続され得るマルチ・チップ・モジュール（MCM）に組み込まれ得る。

10

【0015】

本願で示されているコンピューティングシステム100は実例であり、変形及び変更が可能であることが理解されるだろう。ブリッジの数及び配置、プロセッサ102の数、並びに並列プロセッサ112の数を含む接続トポロジは、望まれるように変更されてよい。例えば、システムメモリ104は、ブリッジを通じてよりもむしろ直接にプロセッサ102へ接続可能であり、一方、他のデバイスは、メモリハブ105及びプロセッサ102を介してシステムメモリ104と通信する。他の代替のトポロジでは、並列プロセッサ112は、メモリハブ105へよりもむしろ、I/Oハブ107へ、又は1つ以上のプロセッサ102の中の1つへ直接に接続される。他の実施形態では、I/Oハブ107及びメモリハブ105は、単一のチップに組み込まれてもよい。また、2組以上のプロセッサ102が、並列プロセッサ112の2つ以上のインスタンスと結合することができる複数のソケットを介して取り付けられることも可能である。

20

【0016】

ここで示されている特定のコンポーネントのいくつかは任意であり、コンピューティングシステム100の全ての実施に含まれなくてもよい。例えば、任意の数のアドインカード又はペリフェラルがサポートされてよく、あるいは、いくつかのコンポーネントは削除されてもよい。更には、いくつかのアーキテクチャは、図1に表されているものと類似したコンポーネントに対して異なった用語を使用してもよい。例えば、メモリハブ105は、いくつかのアーキテクチャではノースブリッジと呼ばれることがあり、一方、I/Oハブ107は、サウスブリッジと呼ばれることがある。

30

【0017】

図2Aは、並列プロセッサ200を表す。並列プロセッサ200は、本願で記載されているGPU、GPGPU、などであってよい。並列プロセッサ200の様々なコンポーネントは、プログラム可能なプロセッサ、特定用途向け集積回路（ASIC）、又はフィールド・プログラマブル・ゲート・アレイ（FPGA）などの1つ以上の集積回路デバイスを用いて実装されてよい。表されている並列プロセッサ200は、図1に示されている並列プロセッサ112又はそのうちの1つであってよい。

【0018】

並列プロセッサ200は、並列処理ユニット202を含む。並列処理ユニット202は、並列処理ユニット202の他のインスタンスを含む他のデバイスとの通信を可能にするI/Oユニット204を含む。I/Oユニット204は、他のデバイスへ直接に接続されてよい。例えば、I/Oユニット204は、メモリハブ105などのハブ又はスイッチインターフェースの使用により他のデバイスと接続する。メモリハブ105とI/Oユニット204との間の接続は、通信リンク113を形成する。並列処理ユニット202内で、I/Oユニット204は、ホストインターフェース206及びメモリクロスバー216と接続し、ホストインターフェース206は、処理動作を実行することに向けられたコマンドを受け取り、メモリクロスバー216は、メモリ動作を実行することに向けられたコマンドを受け取る。

40

【0019】

ホストインターフェース206がI/Oユニット204を介してコマンドバッファを受

50

け取るとき、ホストインターフェース 206 は、それらのコマンドを実行するための作業操作をフロントエンド 208 へ向けることができる。一実施形態で、フロントエンド 208 はスケジューラ 210 と結合し、スケジューラ 210 は、コマンド又は他の作業項目を処理クラスタアレイ 212 へ分配するよう構成される。スケジューラ 210 は、処理クラスタアレイ 212 が適切に構成され、処理クラスタアレイ 212 の処理クラスタへタスクが分配される前に有効な状態にあることを確かにする。スケジューラ 210 は、マイクロコントローラで実行されるファームウェアロジックにより実装されてよい。マイクロコントローラにより実装されたスケジューラ 210 は、処理アレイ 212 で実行されるスレッドの迅速なプリエンプション及びコンテキストスイッチングを可能にしながら、複雑なスケジューリング及び作業分配操作を粗い及び細かい粒度で実行するよう構成される。望ましくは、ホストソフトウェアは、複数のグラフィクス処理コアの中の一つにより、処理アレイ 212 に対するスケジューリングのためのワークロードを証明することができる。次いで、ワークロードは、スケジューラマイクロコントローラ内のスケジューラ 210 のロジックによって、処理アレイ 212 にわたって自動的に分配され得る。

【0020】

処理クラスタアレイ 212 は、最大“N”個までの処理クラスタ（例えば、クラスタ 214 A、クラスタ 214 B、乃至クラスタ 214 N）を含むことができる。処理クラスタアレイ 212 の各クラスタ 214 A～214 N は、多数の同時のスレッドを実行することができる。スケジューラ 210 は、各タイプのプログラム又は計算について生じるワークロードに応じて様々であり得る様々なスケジューリング及び／又は作業分配アルゴリズムを用いて、処理クラスタアレイ 212 のクラスタ 214 A～214 N へ作業を割り当てることができる。スケジューリングは、スケジューラ 210 によって動的に処理され得るか、あるいは、処理クラスタアレイ 212 による実行のために構成されたプログラムロジックのコンパイル中にコンパイラロジックによって部分的に支援され得る。任意に、処理クラスタアレイ 212 の異なるクラスタ 214 A～214 N は、異なるタイプのプログラムを処理するために、及び異なるタイプの計算を実行するために、割り当てられ得る。

【0021】

処理クラスタアレイ 212 は、様々なタイプの並列処理動作を実行するよう構成され得る。例えば、クラスタアレイ 212 は、汎用の並列計算動作を実行するよう構成される。例えば、処理クラスタアレイ 212 は、ビデオ及び／又はオーディオデータのフィルタリング、物理演算を含むモデリング動作の実行、及びデータ変形の実行を含む処理タスクを実行するためのロジックを含むことができる。

【0022】

処理クラスタアレイ 212 は、並列なグラフィクス処理動作を実行するよう構成される。並列プロセッサ 200 がグラフィクス処理動作を実行するよう構成されているそのような実施形態で、処理クラスタアレイ 212 は、テッセレーションロジック及び他の頂点処理ロジックとともにテクスチャ動作を実行するテクスチャサンプリングロジックを含むがこれに限られないそのようなグラフィクス処理動作の実行をサポートする追加のロジックを含むことができる。更には、処理クラスタアレイ 212 は、頂点シェーダ、テッセレーションシェーダ、ジオメトリシェーダ、及びピクセルシェーダなどの、しかしこれらに限られないグラフィクス処理関連シェーダプログラムを実行するよう構成され得る。並列処理ユニット 202 は、データを処理のためにシステムメモリから I/O ユニット 204 を介して転送することができる。処理中、転送されたデータは、処理中にオンチップメモリ（例えば、並列プロセッサメモリ 222）に記憶され、次いで、システムメモリへ書き込まれ得る。

【0023】

並列処理ユニット 202 がグラフィクス処理を実行するために使用される実施形態で、スケジューラ 210 は、処理クラスタアレイ 212 の複数のクラスタ 214 A～214 N へのグラフィクス処理動作の分配をより良く可能にするために、処理ワークロードをおおよそ等しいサイズのタスクに分けるよう構成されてよい。これらの実施形態のいくつかで

10

20

30

40

50

、処理クラスタアレイ 2 1 2 の部分は、異なるタイプの処理を実行するよう構成され得る。例えば、表示のためのレンダリングされた画像を生成するために、第 1 部分は、頂点シェーディング及びトポロジ生成を実行するよう構成されてよく、第 2 部分は、テッセレーション及びジオメトリシェーディングを実行するよう構成されてよく、第 3 部分は、ピクセルシェーディング又は他のスクリーン空間動作を実行するよう構成されてよい。クラスタ 2 1 4 A ~ 2 1 4 N のうちの 1 つ以上によって生成された中間データは、中間データが更なる処理のためにクラスタ 2 1 4 A ~ 2 1 4 N の間で伝送されることを可能にするようバッファに格納されてよい。

【0024】

動作中、処理クラスタアレイ 2 1 2 は、スケジューラ 2 1 0 を介して、実行されるべき処理タスクを受け取ることができ、スケジューラ 2 1 0 は、フロントエンド 2 0 8 から、処理タスクを定義するコマンドを受け取る。グラフィクス処理動作のために、処理タスクは、処理されるべきデータのインデックス、例えば、サーフェス（パッチ）データ、プリミティブデータ、頂点データ、及びピクセルデータを、どのようにしてデータが処理されるべきか（例えば、どのようなプログラムが実行されるべきか）を定義する状態パラメータ及びコマンドとともに含むことができる。スケジューラ 2 1 0 は、タスクに対応するインデックスをフェッチするよう構成されてよく、あるいは、フロントエンド 2 0 8 からインデックスを受け取ってもよい。フロントエンド 2 0 8 は、処理クラスタアレイ 2 1 2 が、入来するコマンドバッファ（例えば、パッチバッファ、プッシュバッファ、など）によって指定されたワークロードが開始される前に有効な状態に設定されることを確かにするよう構成され得る。

【0025】

並列処理ユニット 2 0 2 の 1 つ以上のインスタンスの夫々は、並列プロセッサメモリ 2 2 2 と結合することができる。並列プロセッサメモリ 2 2 2 は、メモリクロスバー 2 1 6 を介してアクセスされ得る。メモリクロスバー 2 1 6 は、処理クラスタアレイ 2 1 2 及び I/O ユニット 2 0 4 からメモリリクエストを受け取ることができる。メモリクロスバー 2 1 6 は、メモリインターフェース 2 1 8 を介して並列プロセッサメモリ 2 2 2 にアクセスすることができる。メモリインターフェース 2 1 8 は、並列プロセッサメモリ 2 2 2 の部分（例えば、メモリユニット）へ夫々が結合することができる複数のパーティションユニット（例えば、パーティションユニット 2 2 0 A、パーティションユニット 2 2 0 B、乃至パーティションユニット 2 2 0 N）を含むことができる。パーティションユニット 2 2 0 A ~ 2 2 0 N の数は、メモリユニットの数に等しいよう構成されてよく、それにより、第 1 パーティションユニット 2 2 0 A は、対応する第 1 メモリユニット 2 2 4 A を有し、第 2 パーティションユニット 2 2 0 B は、対応するメモリユニット 2 2 4 B を有し、N 番目のパーティションユニット 2 2 0 N は、対応する N 番目のメモリユニット 2 2 4 N を有する。他の実施形態では、パーティションユニット 2 2 0 A ~ 2 2 0 N の数は、メモリデバイスの数に等しくなくてもよい。

【0026】

メモリユニット 2 2 4 A ~ 2 2 4 N は、グラフィクス・ダブル・データ・レート（GDDR）メモリを含む動的ランダム・アクセス・メモリ（DRAM）又はグラフィクス・ランダム・アクセス・メモリ、例えば、同期グラフィクス・ランダム・アクセス・メモリ（SGRAM）を含む様々なタイプのメモリデバイスを含むことができる。任意に、メモリユニット 2 2 4 A ~ 2 2 4 N は、高帯域幅メモリ（HBM）を含むがこれに限られない 3D 積層メモリも含んでよい。当業者に明らかなように、メモリユニット 2 2 4 A ~ 2 2 4 N は様々なことができ、様々な従来設計のうちの 1 つから選択可能である。フレームバッファ又はテクスチャマップなどのレンダーターゲットは、メモリユニット 2 2 4 A ~ 2 2 4 N にわたって記憶されてよく、パーティションユニット 2 2 0 A ~ 2 2 0 N が、並列プロセッサメモリ 2 2 2 の利用可能な帯域幅を効率よく使用するように各レンダーターゲットの部分を書き込むことを可能にする。いくつかの実施形態において、並列プロセッサメモリ 2 2 2 のローカルインスタンスは、ローカルキャッシュメモリとともにシステムメ

メモリを利用する統合されたメモリ設計を支持して、除かれてもよい。

【0027】

任意に、処理クラスタアレイ212のクラスタ214A~214Nのうちのいずれか1つは、並列プロセッサメモリ222内のメモリユニット224A~224Nのうちのいずれかに書き込まれることになるデータを処理する能力を備えている。メモリクロスバー216は、各クラスタ214A~214Nの出力を、いずれかのパーティションユニット220A~220Nへ、又は出力に対して追加の処理動作を実行することができる他のクラスタ214A~214Nへ転送するよう構成され得る。各クラスタ214A~214Nは、様々な外部メモリデバイスから読み出すために又はそれらに書き込むために、メモリクロスバー216を通じてメモリインターフェース218と通信することができる。メモリクロスバー216を備えた実施形態のうちの1つで、メモリクロスバー216は、I/Oユニット204と通信するためのメモリインターフェース218への接続と、並列プロセッサメモリ222のローカルインスタンスへの接続とを有し、異なる処理クラスタ214A~214N内の処理ユニットが、並列処理ユニット202にとってローカルでないシステムメモリ又は他のメモリと通信することを可能にする。一般に、メモリクロスバー216は、例えば、クラスタ214A~214Nとパーティションユニット220A~220Nとの間でトラフィックストリームを分離するために仮想チャネルを使用することが可能であり得る。

10

【0028】

並列処理ユニット202の単一のインスタンスが並列プロセッサ200内に表されているが、並列処理ユニット202の任意の数のインスタンスが含まれ得る。例えば、並列処理ユニット202の複数のインスタンスが、単一のアドインカードで設けられ得るか、あるいは、複数のアドインカードが相互接続され得る。並列処理ユニット202の異なるインスタンスは、たとえ異なるインスタンスが異なる数の処理コア、異なる量のローカル並列プロセッサメモリ、及び／又は他の構成相違点を有しているとしても、相互動作するよう構成され得る。任意に、並列処理ユニット202のいくつかのインスタンスは、他のインスタンスよりも高い精度の浮動小数点ユニットを含むことができる。並列処理ユニット202又は並列プロセッサ200の1つ以上のインスタンスを組み込むシステムは、デスクトップ、ラップトップ、若しくは携帯型パーソナルコンピュータ、サーバ、ワークステーション、ゲーム機、及び／又は埋め込み型システムを含むがこれらに限られない様々な構成及び形状因子で実装され得る。

20

30

【0029】

図2Bは、パーティションユニット220のブロック図である。パーティションユニット220は、図2Aのパーティションユニット220A~220Nのうちの1つのインスタンスであってよい。表されているように、パーティションユニット220は、L2キャッシュ221、フレームバッファインターフェース225、及びROP226 (raster operations unit) を含む。L2キャッシュ221は、メモリクロスバー216及びROP226から受け取られたロード及びストア動作を実行するよう構成される読み出し／書き込みキャッシュである。読み出しミス及び緊急の書き戻しサーフェスは、L2キャッシュ221によってフレームバッファインターフェース225へ処理のために出力される。アップデートも、処理のためにフレームバッファインターフェース225を介してフレームバッファへ送信され得る。一実施形態で、フレームバッファインターフェース225は、図2Aのメモリユニット224A~224N (例えば、並列プロセッサメモリ222内) などの並列プロセッサメモリ内のメモリユニットの中の1つとインターフェース接続する。パーティションユニット220は、追加的に、又は代替的に、メモリコントローラ (図示せず) を介して並列プロセッサメモリ内のメモリユニットの1つとインターフェース接続してもよい。

40

【0030】

グラフィクスアプリケーションでは、ROP226は、ステンシル、zテスト、ブレンディングなどのようなラスタ動作を実行する処理ユニットである。ROP226は、次い

50

で、処理されたグラフィクスデータを出力し、処理されたグラフィクスデータは、グラフィクスメモリに記憶される。いくつかの実施形態において、ROP226は、メモリに書き込まれるデプス又はカラーデータを圧縮し、メモリから読み出されるデプス又はカラーデータを圧縮解除する圧縮ロジックを含む。圧縮ロジックは、複数の圧縮アルゴリズムのうちの1つ以上を使用するロスレス圧縮ロジックであることができる。ROP226によって実行される圧縮のタイプは、圧縮されるべきデータの統計特性に基づいて様々であることができる。例えば、一実施形態で、デルタカラー圧縮は、タイルごとにデプス及びカラーデータに対して実行される。

【0031】

ROP226は、パーティションユニット220内ではなく、各処理クラスタ（例えば、図2Aのクラスタ214A～214N）内に含まれてもよい。そのような実施形態では、ピクセルデータの読み出し及び書き込みリクエストは、ピクセルフラグメントデータの代わりにメモリクロスバー216を介して伝送される。処理されたグラフィクスデータは、図1の1つ以上の表示デバイス110のうちの1つなどの表示デバイスで表示されるか、プロセッサ102による更なる処理のためにルーティングされるか、あるいは、図2Aの並列プロセッサメモリ222内の処理エンティティのうちの1つによる更なる処理のためにルーティングされてよい。

【0032】

図2Cは、並列処理ユニット内の処理クラスタ214のブロック図である。例えば、処理クラスタは、図2Aの処理クラスタ214A～214Nのうちの1つのインスタンスである。処理クラスタ214は、多くのスレッドを同時に実行するよう構成され得る。ここで、「スレッド」という用語は、特定の組の入力データに対して実行する特定のプログラムのインスタンスを指す。任意に、SIMD（single-instruction, multiple-data）命令発行技術が、多数の独立した命令ユニットを設けずに、多数のスレッドの並列実行をサポートするために使用されてよい。代替的に、SMT（single-instruction, multiple-thread）技術が、処理クラスタの各1つの中の処理エンジンの組へ命令を発行するよう構成された共通命令ユニットを使用して、多数の概ね同期されたスレッドの並列実行をサポートするために使用されてもよい。全ての処理エンジンが通常は同じ命令を実行するSIMD実行レジームとは異なり、SMT実行は、異なるスレッドが、所与のスレッドプログラムを通じて、異なる実行パスをより簡単に辿ることを可能にする。当業者であれば、SIMD処理レジームはSMT処理レジームの機能サブセットを表している、と理解するだろう。

【0033】

処理クラスタ214の動作は、処理タスクをSMT並列プロセッサに分配するパイプラインマネージャ232を介して制御され得る。パイプラインマネージャ232は、図2のスケジューラ210から命令を受け取り、グラフィクスマルチプロセッサ234及び／又はテクスチャユニット236によるこれらの命令の実行を管理する。表されているグラフィクスマルチプロセッサ234は、SMT並列プロセッサの例示的なインスタンスである。しかし、異なるアーキテクチャの様々なタイプのSMT並列プロセッサが処理クラスタ214内に含まれてよい。グラフィクスマルチプロセッサ234の1つ以上のインスタンスが処理クラスタ214内に含まれ得る。グラフィクスマルチプロセッサ234は、データを処理することができ、データクロスバー240は、処理されたデータを、他のシェーダユニットを含む複数の可能なあて先のうちの1つへ分配するために使用され得る。パイプラインマネージャ232は、処理されたデータがデータクロスバー240を介して分配されるためのあて先を指定することによって、処理されたデータの分配を促すことができる。

【0034】

処理クラスタ214内の各グラフィクスマルチプロセッサ234は、同じ組の機能実行ロジック（例えば、算術ロジックユニット、ロードストアユニット、など）を含むことができる。機能実行ロジックは、前の命令が完了する前に新しい命令が発行され得るパイ

10

20

30

40

50

ブライン化された状態で構成され得る。機能実行ロジックは、整数及び浮動小数点演算、比較演算、ブール演算、ビットシフト、及び様々な代数関数の計算を含む様々な演算をサポートする。同じ機能ユニットハードウェアは、種々の演算を実行するために利用可能であり、機能ユニットの如何なる組み合わせも存在してよい。

【0035】

処理クラスタ214へ伝送された命令は、スレッドを構成する。並列処理エンジンにわたって実行されるスレッドの組は、スレッドグループである。スレッドグループは、異なる入力データに対して同じプログラムを実行する。スレッドグループ内の各スレッドは、グラフィクスマルチプロセッサ234内の異なる処理エンジンへ割り当てられ得る。スレッドグループに含まれているスレッドは、グラフィクスマルチプロセッサ234内の処理エンジンの数よりも少なくてもよい。スレッドグループに含まれているスレッドが処理エンジンの数よりも少ない場合に、処理エンジンのうちの1つ以上は、スレッドグループが処理中であるサイクル中にアイドル状態であってよい。スレッドグループに含まれているスレッドはまた、グラフィクスマルチプロセッサ234内の処理エンジンの数よりも多くてもよい。スレッドグループに含まれているスレッドがグラフィクスマルチプロセッサ234内の処理エンジンの数よりも多い場合に、処理は、連続したクロックサイクルにわたって実行され得る。任意に、複数のスレッドグループがグラフィクスマルチプロセッサ234で同時に実行され得る。

【0036】

グラフィクスマルチプロセッサ234は、ロード及びストア動作を実行するよう内部キャッシュメモリを含んでもよい。任意に、グラフィクスマルチプロセッサ234は、内部キャッシュを先送りし、処理クラスタ214内のキャッシュメモリ（例えば、L1キャッシュ248）を使用することができる。各グラフィクスマルチプロセッサ234はまた、全ての処理クラスタ214にわたって共有されてスレッド間でデータを転送するために使用され得るパーティションユニット（例えば、図2Aのパーティションユニット220A～220N）内のL2キャッシュへのアクセスも有する。グラフィクスマルチプロセッサ234はまた、オフチップグローバルメモリにアクセスしてもよい。オフチップグローバルメモリは、ローカル並列プロセッサメモリ及び／又はシステムメモリのうちの1つ以上を含むことができる。並列処理ユニット202の外にある如何なるメモリも、グローバルメモリとして使用されてよい、処理クラスタ214がグラフィクスマルチプロセッサ234の複数のインスタンスを含む実施形態は、L1キャッシュ248に記憶され得る共通命令及びデータを共有することができる。

【0037】

各処理クラスタ214は、仮想アドレスを物理アドレスにマッピングするよう構成されるMMU245（memory management unit）を含んでもよい。他の実施形態では、MMU245の1つ以上のインスタンスは、図2Aのメモリインターフェース218内に存在してよい。MMU245は、仮想アドレスをタイルの物理アドレスと、任意にキャッシュラインインデックスとにマッピングするために使用されるページテーブルエントリ（PTE）の組を含む。MMU245は、グラフィクスマルチプロセッサ234若しくはL1キャッシュ248又は処理クラスタ214内に存在し得るアドレ스트ランスレーション・ルックアサイド・バッファ（TLB）又はキャッシュを含んでもよい。物理アドレスは、パーティションユニットの間で効率的なリクエストインターリーピングを可能にするためにサーフェスデータアクセスの局所性を分散するよう処理される。キャッシュラインインデックスは、キャッシュラインに対するリクエストがヒットかミスかを決定するために使用されてよい。

【0038】

グラフィクス及びコンピューティングアプリケーションでは、処理クラスタ214は、各グラフィクスマルチプロセッサ234が、テクスチャマッピング動作、例えば、テクスチャサンプル位置を決定すること、テクスチャデータを読み出すこと、及びテクスチャデータにフィルタをかけること、を実行するために、テクスチャユニット236へ結合され

10

20

30

40

50

るように構成されてよい。テクスチャデータは、内部テクスチャL1キャッシュ（図示せず）から、又はいくつかの実施形態では、グラフィクスマルチプロセッサ234内のL1キャッシュから読み出され、必要に応じて、L2キャッシュ、ローカル並列プロセッサメモリ、又はシステムメモリからフェッチされる。各グラフィクスマルチプロセッサ234は、処理されたタスクを、更なる処理のために、他の処理クラスタ214へ供給するよう、あるいは、処理されたタスクをL2キャッシュ、ローカル並列プロセッサメモリ、又はシステムメモリにメモリクロスバー216を介して格納するよう、処理されたタスクをデータクロスバー240へ出力する。preROP242（pre-raster operations unit）は、グラフィクスマルチプロセッサ234からデータを受け取り、データを、本願で記載されるパーティションユニット（例えば、図2Aのパーティションユニット220A～220N）により位置を定められ得るROPユニットへ向けるよう構成される。preROP242ユニットは、色混合のための最適化を実行し、ピクセル色データを編成し、アドレス変換を実行することができる。

10

【0039】

本願で記載されるコアアーキテクチャは実例であり、変更及び変形が可能であることが理解されるだろう。任意の数の処理ユニット、例えば、グラフィクスマルチプロセッサ234、テクスチャユニット236、preROP242などが処理クラスタ214に含まれてよい。更に、ただ1つの処理ユニット214しか示されていないが、本願で記載される並列処理ユニットは、処理クラスタ214の任意の数のインスタンスを含んでよい。任意に、各処理クラスタ214は、別々の相異なる処理ユニット、L1キャッシュ、などを用いて他の処理クラスタ214から独立して動作するよう構成され得る。

20

【0040】

図2Dは、グラフィクスマルチプロセッサ234が処理クラスタ214のパイプラインマネージャ232と結合するグラフィクスマルチプロセッサ234の例を示す。グラフィクスマルチプロセッサ234は、命令キャッシュ252、命令ユニット254、アドレスマッピングユニット256、レジスタファイル258、1つ以上の汎用グラフィクス処理ユニット（GPGPU）コア262、及び1つ以上のロード／ストアユニット266を含むがこれらに限られない実行パイプラインを備える。GPGPUコア262及びロード／ストアユニット266は、メモリ及びキャッシュインターコネクト268を介してキャッシュメモリ272及び共有メモリ270と結合される。グラフィクスマルチプロセッサ234は、行列及び／又はレイトレーシング演算を加速させるハードウェアロジックを含むテンソル及び／又はレイトレーシングコア263を更に含んでもよい。

30

【0041】

命令キャッシュ252は、パイプラインマネージャ232から、実行すべき命令のストリームを受け取ってよい。命令は、命令キャッシュ252にキャッシュされ、命令ユニット254による実行のためにディスパッチされる。命令ユニット254は、スレッドグループ（例えば、ワープ）として命令をディスパッチすることができ、スレッドグループの各スレッドは、GPGPUコア262内の異なる実行ユニットに割り当てられる。命令は、統合アドレス空間内でアドレスを指定することによって、ローカル、共有、又はグローバルアドレス空間のいずれかにアクセスすることができる。アドレスマッピングユニット256は、統合アドレス空間内のアドレスを、ロード／ストアユニット266によってアクセスされ得る相異なるメモリアドレスに変換するために使用され得る。

40

【0042】

レジスタファイル258は、グラフィクスマルチプロセッサ234の機能ユニットのためのレジスタの組を提供する。レジスタファイル258は、グラフィクスマルチプロセッサ234の機能ユニット（例えば、GPGPUコア262、ロード／ストアユニット266）のデータパスへ接続されたオペランドの一時記憶を提供する。レジスタファイル258は、各機能ユニットがレジスタファイル258の専用の部分を割り当てられるように、機能ユニットの夫々の間で分割されてよい。例えば、レジスタファイル258は、グラフィクスマルチプロセッサ234によって実行される異なるワープ間で分割される。

50

【0043】

GPGPUコア262は、グラフィクスマルチプロセッサ234の命令を実行するために使用される浮動小数点ユニット(FPU)及び／又は整数算術ロジックユニット(ALU)を夫々含むことができる。いくつかの実施で、GPGPUコア262は、テンソル及び／又はレイトレーシングコア263内にさもなければ存在してもよいハードウェアロジックを含むことができる。GPGPUコア262は、アーキテクチャが類似していることができ、あるいは、アーキテクチャが異なることができる。例えば、一実施形態において、GPGPUコア262の第1部分は、単精度FPU及び整数ALUを含み、一方、GPGPUコア262の第2部分は、倍精度FPUを含む。任意に、FPUは、浮動小数点算術のためのIEEE754-2008を実装するか、あるいは、変数精度浮動小数点算術を有効にすることができる。グラフィクスマルチプロセッサ234は、コピー長方形又はピクセル混合動作などの特定の機能を実行するよう1つ以上の固定機能又は特別の機能を更に含むことができる。GPGPUコアの1つ以上はまた、固定又は特別の機能ロジックも含むことができる。

10

【0044】

GPGPUコア262は、複数組のデータに対して単一の命令を実行することができるSIMDロジックを含んでよい。任意に、GPGPUコア262は、SIMD4、SIMD8、及びSIMD16命令を物理的に実行し、SIMD1、SIMD2、及びSIMD32命令を論理的に実行することができる。GPGPUコアのためのSIMD命令は、シェーダコンパイラによってコンパイル時に生成されるか、あるいは、SPMD(single program multiple data, SPMD)又はSIMTアーキテクチャのために記述及びコンパイルされたプログラムを実行するときに自動的に生成され得る。SIMT実行モデルのために構成されたプログラムの複数のスレッドは、単一のSIMD命令により実行可能である。例えば、一実施形態において、同じ又は類似した動作を実行する8つのSIMTスレッドが、単一のSIMD8ロジックユニットにより並列に実行され得る。

20

【0045】

メモリ及びキャッシュインターコネクト268は、グラフィクスマルチプロセッサ234の機能ユニットの夫々をレジスタファイル258へ及び共有メモリ270へ接続するインターコネクトネットワークである。例えば、メモリ及びキャッシュインターコネクト268は、ロード／ストアユニット266が共有メモリ270とレジスタファイル258との間でロード及びストア動作を実装することを可能にするクロスバーインターコネクトである。レジスタファイル258は、GPGPUコア262と同じ周波数で動作することができるので、GPGPUコア262とレジスタファイル258との間のデータ転送は、レイテンシが非常に小さい。共有メモリ270は、グラフィクスマルチプロセッサ234内の機能ユニットで実行されるスレッド間の通信を可能にするために使用され得る。キャッシュメモリ272は、例えば、機能ユニットとテクスチャユニット236との間で通信されたテクスチャデータをキャッシュするために、データキャッシュとして使用され得る。共有メモリ270はまた、キャッシュされて管理されるプログラムとしても使用され得る。GPGPUコア262で実行されるスレッドは、キャッシュメモリ272内に格納されている自動的にキャッシュされたデータに加えて、共有メモリ内にデータをプログラム可能に格納することができる。

30

40

【0046】

図3A～3Cは、実施形態に従って、更なるグラフィクスマルチプロセッサを表す。図3A～3Bは、図2Cのグラフィクスマルチプロセッサ234に関係があり、それらのうちの1つの代わりに使用されてよいグラフィクスマルチプロセッサ325、350を表す。従って、本願のグラフィクスマルチプロセッサ234と組み合わせられる如何なる特徴の開示も、グラフィクスマイクロプロセッサ325との対応する組み合わせを開示するが、そのように限定されない。図3Cは、マルチコアグループ365A～365Nに配置されたグラフィクス処理リソースの専用の組を含むグラフィクス処理ユニット(GPU)380を表し、GPU380は、グラフィクスマイクロプロセッサ325、350に対応する

50

。表されているグラフィクスマルチプロセッサ 3 2 5、3 5 0 及びマルチコアグループ 3 6 5 A～3 6 5 N は、多数の実行スレッドの同時の実行が可能なストリーミング・マルチプロセッサ (streaming multiprocessors, S M) であることができる。

【0 0 4 7】

図 3 A のグラフィクスマルチプロセッサ 3 2 5 は、図 2 D のグラフィクスマルチプロセッサ 2 3 4 に対して、実行リソースユニットの複数の追加インスタンスを含む。例えば、グラフィクスマルチプロセッサ 3 2 5 は、命令ユニット 3 3 2 A～3 3 2 B、レジスタファイル 3 3 4 A～3 3 4 B、及びテクスチャユニット 3 4 4 A～3 4 4 B の複数のインスタンスを含むことができる。グラフィクスマルチプロセッサ 3 2 5 はまた、複数組のグラフィクス又は計算実行ユニット (例えば、G P G P U コア 3 3 6 A～3 3 6 B、テンソルコア 3 3 7 A～3 3 7 B、レイトラッシングコア 3 3 8 A～3 3 8 B) と、複数組のロード／ストアユニット 3 4 0 A～3 4 0 B とを含む。実行リソースユニットは、共通命令キャッシュ 3 3 0、テクスチャ及び／又はデータキャッシュメモリ 3 4 2、及び共有メモリ 3 4 6 を備える。

【0 0 4 8】

様々なコンポーネントは、インターコネクトファブリック 3 2 7 を介して通信することができる。インターコネクトファブリック 3 2 7 は、グラフィクスマルチプロセッサ 3 2 5 の様々なコンポーネント間の通信を可能にするよう 1 つ以上のクロスバースイッチを含んでよい。インターコネクトファブリック 3 2 7 は、グラフィクスマルチプロセッサ 3 2 5 の各コンポーネントがスタックされる別個の高速ネットワークファブリックレイヤであってよい。グラフィクスマルチプロセッサ 3 2 5 のコンポーネントは、インターコネクトファブリック 3 2 7 を介して遠隔のコンポーネントと通信する。例えば、G P G P U コア 3 3 6 A～3 3 6 B、3 3 7 A～3 3 7 B、及び 3 3 8 A～3 3 8 B は夫々、インターコネクトファブリック 3 2 7 を介して共有メモリ 3 4 6 と通信することができる。インターコネクトファブリック 3 2 7 は、コンポーネント間の公平な帯域幅割り当てを確かなにするために、グラフィクスマルチプロセッサ 3 2 5 内の通信を調停することができる。

【0 0 4 9】

図 3 B のグラフィクスマルチプロセッサ 3 5 0 は、複数組の実行リソース 3 5 6 A～3 5 6 D を含み、実行リソースの各組は、図 2 D 及び図 3 A に表されているような複数の命令ユニット、レジスタファイル、G P G P U コア、及びロード／ストアユニットを含む。実行リソース 3 5 6 A～3 5 6 D は、命令キャッシュ 3 5 4 及び共有メモリ 3 5 3 を共有しながら、テクスチャ動作のためのテクスチャユニット 3 6 0～3 6 0 D に呼応して動作することができる。例えば、実行リソース 3 5 6 A～3 5 6 D は、テクスチャ及び／又はデータキャッシュメモリ 3 5 8 A～3 5 8 B の複数のインスタンスとともに、命令キャッシュ 3 5 4 及び共有メモリ 3 5 3 を共有することができる。様々なコンポーネントは、図 3 A のインターコネクトファブリック 3 2 7 に類似したインターコネクトファブリック 3 5 2 を介して通信することができる。

【0 0 5 0】

当業者であれば、図 1、2 A～2 D、及び 3 A～3 B に示されているアーキテクチャは説明であり、本実施形態の範囲に関する限定ではない、と理解するだろう。よって、本願で説明される技術は、本願で記載される実施形態の範囲から逸脱せずに、制限なしに、1 つ以上のモバイルアプリケーションプロセッサ、マルチコア C P U を含む 1 つ以上のデスクトップ又はサーバ中央演算処理ユニット、図 2 A の並列処理ユニット 2 0 2 のような 1 つ以上の並列処理ユニット、及び 1 つ以上のグラフィクスプロセッサ又は特別目的の処理ユニットを含む如何なる適切に構成された処理ユニットでも実装されてよい。

【0 0 5 1】

本願で記載される並列プロセッサ又は G P G P U は、グラフィクス操作、機械学習動作、パターン解析動作、及び様々な汎用 G P U (G P G P U) 機能を加速させるようホスト／プロセッサコアへ通信可能に結合されてよい。G P U は、バス又は他のインターコネクト (例えば、P C I e 又は N V L i n k などの高速インターコネクト) を介してホストブ

10

20

30

40

50

ロセッサ／コアへ通信可能に結合されてよい。他の実施形態では、GPUは、コアと同じパッケージ又はチップに集積され、内部プロセッサバス／インターコネクト（すなわち、パケット又はチップ内にある）を介してコアへ通信可能に結合されてよい。GPUが接続されている状態にかかわらず、プロセッサコアは、作業記述に含まれているコマンド／命令のシーケンスの形でGPUに作業を割り当ててよい。GPUは、次いで、それらのコマンド／命令を有効に処理するための専用の回路構成／ロジックを使用する。

【0052】

図3Cは、マルチコアグループ365A～365Nに割り当てられたグラフィクス処理リソースの専用の組を含むグラフィクス処理ユニット（GPU）380を表す。ただ1つのマルチコアグループ365Aの詳細しか与えられていないが、他のマルチコアグループ365A～365Nは、グラフィクス処理リソースの同じ又は類似した組を備えられてよいことが理解されるだろう。マルチコアグループ365A～365Nに関して記載される詳細は、本願で記載されているいずれかのグラフィクスマルチプロセッサ234、325、350にも当てはまり得る。

【0053】

表されているように、マルチコアグループ365Aは、グラフィクスコアの組370、テンソルコアの組371、及びレイトレーシングコアの組372を含んでよい。スケジューラ／ディスパッチャ368は、様々なコア370、371及び372で実行されるグラフィクススレッドをスケジューリング及びディスパッチする。レジスタファイルの組369は、グラフィクススレッドを実行するときにコア370、371及び372によって使用されるオペランド値を格納する。これらは、例えば、整数値を格納する整数レジスタ、浮動小数点値を格納する浮動小数点レジスタ、パックされたデータ要素（整数及び／又は浮動小数点データ要素）を格納するベクトルレジスタ、及びテンソル／行列値を格納するタイルレジスタを含んでよい。タイルレジスタは、ベクトルレジスタの組み合わせられた組として実装されてよい。

【0054】

1つ以上の組み合わせられたレベル1（L1）キャッシュ及び共有メモリユニット373は、各マルチコアグループ365A内に局所的にテクスチャデータ、頂点データ、ピクセルデータ、レイデータ、境界ボリュームデータ、などのグラフィクスデータを格納する。1つ以上のテクスチャユニット374はまた、テクスチャマッピング及びサンプリングなどのテクスチャリング動作を実行するためにも使用され得る。マルチコアグループ365A～365Nの全て又はサブセットによって共有されているレベル2（L2）キャッシュ375は、複数の同時のグラフィクススレッドのためのグラフィクスデータ及び／又は命令を格納する。表されているように、L2キャッシュ375は、複数のマルチコアグループ365A～365Nにわたって共有されてよい。1つ以上のメモリコントローラ367は、GPU380をメモリ366へ結合する。メモリ366は、システムメモリ（例えば、DRAM）及び／又は専用のグラフィクスメモリ（例えば、GDDR6メモリ）であってよい。

【0055】

入出力（I/O）回路構成363は、GPU380を、デジタル信号プロセッサ（DSP）、ネットワークコントローラ、又はユーザ入力デバイスなどの1つ以上のI/Oデバイス362へ結合する。オンチップインターコネクトが、I/Oデバイス362をGPU380及びメモリ366へ結合するために使用されてよい。I/O回路構成363の1つ以上のI/Oメモリ管理ユニット（IOMMU）364は、システムメモリ366へ直接にI/Oデバイス362を結合する。任意に、IOMMU364は、システムメモリ366内の物理アドレスへ仮想アドレスをマッピングするよう複数組のページテーブルを管理する。I/Oデバイス362、CPU361、及びGPU380は、同じ仮想アドレス空間を共有してもよい。

【0056】

IOMMU364の1つの実施において、IOMMU364は仮想化をサポートする。

10

20

30

40

50

この場合に、それは、ゲスト／グラフィクス仮想アドレスをゲスト／グラフィクス物理アドレスにマッピングするページテーブルの第1の組と、ゲスト／グラフィクス物理アドレスをシステム／ホスト物理アドレス（例えば、システムメモリ366内）にマッピングするページテーブルの第2の組とを管理してよい。ページテーブルの第1及び第2の組の夫々のベースアドレスは、制御レジスタに格納され、コンテキストスイッチでスワップアップされてよい（例えば、それにより、新しいコンテキストは、ページテーブルの関連する組へのアクセスを与えられる）。図3Cに表されていないが、コア370、371、372及び／又はマルチコアグループ365A～365Nの夫々は、ゲスト仮想からのゲスト物理への変換、ゲスト物理からホスト物理への変換、及びゲスト仮想からホスト物理への変換をキャッシュするようトランスレーション・ルックアサイド・バッファ（TLB）を含んでもよい。

10

【0057】

CPU361、GPU380、及びI/Oデバイス362は、単一の半導体チップ及び／又はチップパッケージで集積されてよい。表されているメモリ366は、同じチップに集積されてよく、あるいは、オフチップインターフェースを介してメモリコントローラ367へ結合されてよい。1つの実施において、メモリ366は、他の物理システムレベルメモリと同じ仮想アドレス空間を共有するGDDR6メモリを有するが、本願で記載されている基礎原理は、この具体的な実施に限定されない。

【0058】

テンソルコア371は、ディープラーニング動作を実行するために使用される基本計算動作である行列演算を実行するよう特に設計された複数の実行ユニットを含んでよい。例えば、同時の行列乗算演算は、ニューラルネットワークの訓練及び推論のために使用されてよい。テンソルコア371は、単精度浮動小数点（例えば、32ビット）、半精度浮動小数点（例えば、16ビット）、整数ワード（16ビット）、バイト（8ビット）及び半バイト（4ビット）を含む様々なオペランド精度を用いて行列処理を実行してよい。例えば、ニューラルネットワーク実施は、高品質の最終画像を構成するために、複数のフレームからの詳細を潜在的に組み合わせる各レンダリングされたシーンの特徴を抽出する。

20

【0059】

ディープラーニング実施では、並列な行列乗算作業が、テンソルコア371での実行のためにスケジューリングされてよい。ニューラルネットワークの訓練は、特に、有意な数の行列内積演算を必要とする。 $N \times N \times N$ 行列かけ算の内積定式化を処理するために、テンソルコア371は、少なくともN個の内積処理要素を含んでよい。行列かけ算が始まる前に、1つの行列全体がタイルレジスタにロードされ、第2行列の少なくとも1つの列がN回のサイクルの各サイクルでロードされる。サイクルごとに、処理されるN個の内積がある。

30

【0060】

行列要素は、16ビットワード、8ビットバイト（例えば、INT8）及び4ビット半バイト（例えば、INT4）を含む、特定の実施に応じた異なる精度で記憶されてよい。異なる精度モードは、最も有効な精度が異なるワークロード（例えば、バイト及び半バイトへの量子化を許すことができるワークロードを推論すること、など）のために使用されることを確かにするよう、テンソルコアの組371に対して指定されてよい。

40

【0061】

レイトレーシングコア372は、実時間のレイトレーシング又は非実時間のレイトレーシングの両方の実施のためにレイトレーシング演算を加速させてよい。特に、レイトレーシングコア372は、境界ボリュームヒエラルキ（BVH）を用いてレイトラバースを実行し、レイとBVHボリューム内に含まれるプリミティブとの間の交差を識別するレイトラバース／交差回路を含んでよい。レイトレーシングコア372はまた、デプス試験及び選択を実行するための回路構成（例えば、Zバッファ又は同様の配置を使用する）も含んでよい。1つの実施において、レイトレーシングコア372は、本願で記載される画像ノイズ除去技術に呼応してトラバース及び交差動作を実行する。それらの技術のうちの少な

50

くとも一部は、テンソルコア 371 で実行されてよい。例えば、テンソルコア 371 は、レイトレーシングコア 372 によって生成されたフレームのノイズ除去を実行するようディープラーニングニューラルネットワークを実装してよい。しかし、CPU 361、グラフィクスコア 370、及び／又はレイトレーシングコア 372 も、ノイズ除去及び／又はディープラーニングアルゴリズムの全て又は一部を実装してよい。

【0062】

更には、上述されたように、ノイズ除去に対する分散型アプローチが用いられてもよく、GPU 380 は、ネットワーク又は高速インターコネクトを介して他のコンピューティングデバイスへ結合されたコンピューティングデバイス内にある。この分散アプローチでは、相互接続されたコンピューティングデバイスは、異なるタイプの画像フレーム及び／又は異なるグラフィクスアプリケーションについてノイズ除去を実行するために、システム全体が学習する速度を改善するようニューラルネットワーク学習／訓練データを共有してよい。

【0063】

レイトレーシングコア 372 は、グラフィクスコア 370 がレイごとの数千の命令で過負荷になることを防ぐように、全ての BVH トラバース及びレイプリミティブ交差を処理してよい。例えば、各レイトレーシングコア 372 は、境界ボックステスト（例えば、トラバース動作のため）を実行する専用回路の第 1 の組と、レイ三角形交差テスト（例えば、トラバースされている交差レイ）を実行する専用回路の第 2 の組とを含む。よって、例えば、マルチコアグループ 365 A は簡単にレイプロブを起動することができ、レイトレーシングコア 372 は、レイトラバース及び交差を独立して実行し、ヒットデータ（例えば、ヒット、ノーヒット、多ヒットなど）をスレッドコンテキストへ返す。レイトレーシングコア 372 がトラバース及び交差演算を実行する間、他のコア 370、371 は、自由に他のグラフィクス及び計算作業を実行することができる。

【0064】

任意に、各レイトレーシングコア 372 は、BVH 試験動作を実行するトラバースユニット、及び／又はレイプリミティブ交差テストを実行する交差ユニットを含んでよい。交差ユニットは、「ヒット」、「ノーヒット」又は「多ヒット」応答を生成し、それを適切なスレッドへ供給する。トラバース及び交差演算中に、他のコア（例えば、グラフィクスコア 370 及びテンソルコア 371）の実行リソースは、自由に他の形式のグラフィクス作業を実行することができる。

【0065】

後述される 1 つの任意の実施形態では、ハイブリッドラスタライゼーション／レイトレーシングアプローチが使用され、作業が、グラフィクスコア 370 とレイトレーシングコア 372 との間で分配される。

【0066】

レイトレーシングコア 372（及び／又は他のコア 370、371）は、Dispatch Rays コマンドを含む Microsoft の DirectX Ray Tracing (DXR) などのレイトレーシング命令と、オブジェクトごとにシェーダ及びテクスチャの一意の組の割り当てを可能にするレイ生成、最接近ヒット、任意のヒット、及びミスシェーダのためのハードウェアサポートを含んでよい。レイトレーシングコア 372、グラフィクスコア 370、及びテンソルコア 371 によってサポートされ得る他のレイトレーシングプラットフォームは、Vulkan 1.85 である。しかし、本願で記載されている基礎原理は、如何なる特定のレイトレーシング ISA にも限定されないことに留意されたい。

【0067】

一般に、様々なコア 372、371、370 は、レイ生成、最接近ヒット、レイプリミティブ交差、プリミティブごとの階層的な境界ボックス構成、ミス (miss)、ビジット (visit)、及び例外 (exceptions) うちの 1 つ以上のための命令／機能を含むレイトレーシング命令セットをサポートしてよい。より具体的には、好適な実施形態は、次の機能

のうちの1つ以上を実行するためのレイトレーシング命令を含む。

【0068】

レイ生成 (Ray Generation) – レイ生成命令は、夫々のピクセル、サンプル、又は他のユーザ定義された作業割り当てについて実行されてよい。

【0069】

最接近ヒット (Closest Hit) – 最接近ヒット命令は、シーン内でレイとプリミティブとの最も近い交差を見つけるために実行されてよい。

【0070】

任意のヒット (Any Hit) – 任意ヒット命令は、潜在的に、新しい最接近交点を識別するために、シーン内でレイとプリミティブとの間の複数の交差を識別する。

【0071】

交差 (Intersection) – 交差命令は、レイ-プリミティブ交差試験を実行し、結果を出力する。

【0072】

プリミティブごとの境界ボックス構成 (Per-primitive Bounding box Construction) – この命令は、所与のプリミティブ又はプリミティブのグループの周りに境界ボックスを形成する (例えば、新しいBVH又は他のアクセラレーションデータ構造を形成する場合)。

【0073】

ミス (Miss) – レイがシーン内の全てのジオメトリ、又はシーンの指定された領域を外すことを示す。

【0074】

ビジット (Visit) – レイがトラバースすることにある子ボリューム (children volumes) を示す。

【0075】

例外 (Exceptions) – 様々なタイプの例外ハンドラ (例えば、様々なエラー条件について呼び出される) を含む。

【0076】

[GPUがプロセッサインターコネクトをホストする技術]

図4Aは、例えば、図2Aに示される並列プロセッサ200などの、複数のGPU410～413が、高速リンク440A～440D (例えば、バス、ポイント・ツー・ポイントインターコネクト、など) を介して複数のマルチコアプロセッサ405～406へ通信可能に結合される例示的なアーキテクチャを表す。高速リンク440A～440Dは、実施に応じて、4GB/s、30GB/s、80GB/s又はそれ以上の通信スループットをサポートする。PCIe4.0又は5.0及びNVLink2.0を含むがこれらに限られない様々なインターコネクトプロトコルが使用されてよい。しかし、本願で記載されている基礎原理は、如何なる特定の通信プロトコル又はスループットにも限定されない。

【0077】

GPU410～413のうちの2つ以上は、高速リンク440A～440Dのために使用されるものと同じ又は異なるプロトコル/リンクを用いて実装され得る高速リンク442A～442Bを介して相互接続されてよい。同様に、マルチコアプロセッサ405～406のうちの2つ以上は、高速リンク443を介して接続されてよい。高速リンク443は、20GB/s、30GB/s、120GB/s又はそれ以上で動作する対称マルチプロセッサ (symmetric multi-processor, SMP) バスであってよい。代替的に、図4Aに示されている様々なシステムコンポーネント間の全ての通信は、同じプロトコル/リンクを用いて (例えば、通信インターコネクションファブリックを介して) 実現されてよい。述べられているように、しかしながら、本願で記載されている基礎原理は、如何なる特定のタイプのインターコネクト技術にも限定されない。

【0078】

各マルチコアプロセッサ405～406は、夫々メモリインターコネクト430A～4

10

20

30

40

50

30 Bを介してプロセッサメモリ401～402へ通信可能に結合されてよく、各GPU410～413は、夫々GPUメモリインターコネクタ450A～450Dを介してGPUメモリ420～423へ通信可能に結合される。メモリインターコネクタ430A～430B及び450A～450Dは、同じ又は異なったメモリアクセス技術を利用してよい。例として、限定としてではなく、プロセッサメモリ401～402及びGPUメモリ420～423は、動的ランダム・アクセス・メモリ（DRAM）（スタックドDRAMを含む）、グラフィクスDDR SDRAM（GDDR）（例えば、GDDR5、GDDR6）、高バンド幅メモリ（HBM）などの揮発性メモリであってよく、かつ／あるは、3D XPoint/Optane又はNano-Ramなどの不揮発性メモリであって
10 もよい。例えば、メモリのいくつかの部分は、揮発性メモリであってよく、他の部分は、不揮発性メモリであってよい（例えば、2レベルメモリ（2LM）ヒエラルキーを使用する）。

【0079】

後述されるように、様々なプロセッサ405～406及びGPU410～413は夫々特定のメモリ401～402、420～423へ物理的に結合されてよいが、統合されたメモリアーキテクチャが実装されてもよく、そのようなアーキテクチャでは、同じ仮想システムアドレス空間（「実効アドレス」空間とも呼ばれる）が様々な物理メモリの全ての間で分配される。例えば、プロセッサメモリ401～402は、夫々が64GBのシステムメモリアドレス空間を有してよく、GPUメモリ420～423は、夫々32GBのシステムメモリアドレス空間を有してよい（結果として、この例では、アドレス可能なメモリは全部で254GBになる）。
20

【0080】

図4Bは、マルチコアプロセッサ407とグラフィクスアクセラレーションモジュール446との間の相互接続についての更なる任意の詳細を表す。グラフィクスアクセラレーションモジュール446は、高速リンク440を介してプロセッサ407へ結合されているラインカード上に集積された1つ以上のGPUチップを含んでよい。代替的に、グラフィクスアクセラレーションモジュール446は、プロセッサ407と同じパッケージ又はチップ上に集積されてもよい。

【0081】

表されているプロセッサ407は、複数のコア460A～460Dを含み、各コアは、トランザクション・ルックアサイド・バッファ461A～461Dと、1つ以上のキャッシュ462A～462Dとを有する。コアは、命令を実行し、データを処理するための様々な他のコンポーネントを含んでよいが、これらのコンポーネントは、本願で記載されている基礎原理を不明りょうしないよう表されていない（例えば、命令フェッチユニット、分岐予測ユニット、デコーダ、実行ユニット、再順序付けバッファ、など）。キャッシュ462A～462Dは、レベル1（L1）及びレベル2（L2）を有してよい。更には、1つ以上の共有キャッシュ456がキャッシュヒエラルキに含まれ、コア460A～460Dの組によって共有されてもよい。例えば、プロセッサ407の一実施形態は24個のコアを含み、各コアは、それ自体のL1キャッシュ、12個の共有L2キャッシュ、及び12個の共有L3キャッシュを有する。この実施形態では、L2及びL3キャッシュのうちの1つが、2つの隣接したコアによって共有される。プロセッサ407及びグラフィクスアクセラレーションモジュール446は、システムメモリ441と接続し、システムメモリ441は、プロセッサメモリ401～402を含んでよい。
30
40

【0082】

様々なキャッシュ462A～462D、456、及びシステムメモリ441に記憶されているデータ及び命令については、コヒーレンスバス464上のインターコア通信を介して、コヒーレンスが保たれている。例えば、各キャッシュは、それらに関連したキャッシュコヒーレンシロジック／回路を有してよく、特定のキャッシュラインへの読み出し又は書き込みの検出に応答してコヒーレンスバス464上で通信する。1つの実施では、キャッシュスヌーピングプロトコルが、キャッシュアクセスをスヌープするようコヒーレンス
50

バス 4 6 4 上で実装される。キャッシュスヌーピング／コヒーレンシ技術は、当業者によく理解されており、本願で記載されている基礎原理を不明りょうしないようここでは詳細に記載されない。

【0083】

プロキシ回路 4 2 5 はが提供されてよく、それは、グラフィクスアクセラレーションモジュール 4 4 6 がコアのピアとしてキャッシュコヒーレンスプロトコルに関与することを可能にするよう、グラフィクスアクセラレーションモジュール 4 4 6 をコヒーレンスバス 4 6 4 へ通信可能に結合する。特に、インターフェース 4 3 5 は、高速リンク 4 4 0（例えば、P C I e バス、N V L i n k など）を介したプロキシ回路 4 2 5 への接続をもたらし、インターフェース 4 3 7 は、グラフィクスアクセラレーションモジュール 4 4 6 を高速リンク 4 4 0 へ接続する。

【0084】

1つの実施では、アクセラレータ集積化回路 4 3 6 は、グラフィクスアクセラレーションモジュール 4 4 6 の複数のグラフィクス処理エンジン 4 3 1、4 3 2、N に成り代わって、キャッシュ管理、メモリアクセス、コンテキスト管理、及び割り込み管理サービスを提供する。グラフィクス処理エンジン 4 3 1、4 3 2、N は、夫々が別個のグラフィクス処理ユニット（G P U）を有してよい。代替的に、グラフィクス処理エンジン 4 3 1、4 3 2、N は、グラフィクス実行ユニット、メディア処理エンジン（例えば、ビデオエンコーダ／デコーダ）、サンブラ、及びビットエンジンなどのような G P U 内の異なったタイプのグラフィクス処理エンジンを有してもよい。言い換えれば、グラフィクスアクセラレーションモジュールは、複数のグラフィクス処理エンジン 4 3 1～4 3 2、N を備えた G P U であってよく、あるいは、グラフィクス処理エンジン 4 3 1～4 3 2、N は、共通のパッケージ、ラインカード、又はチップ上に集積された個別的な G P U であってよい。

【0085】

アクセラレータ集積化回路 4 3 6 は、仮想－物理メモリ変換（実効－実質メモリ変換とも呼ばれる）などの様々なメモリ管理機能と、システムメモリ 4 4 1 にアクセスするためのメモリアクセスポロトコルとを実行するメモリ管理ユニット（M M U）4 3 9 を含んでよい。M M U 4 3 9 は、仮想／実効－物理／実質アドレス変換をキャッシュするためのトランスレーション・ルックアサイド・バッファ（T L B）（図示せず）も含んでよい。1つの実施において、キャッシュ 4 3 8 は、グラフィクス処理エンジン 4 3 1、4 3 2、N による有効なアクセスのためにコマンド及びデータを記憶する。キャッシュ 4 3 8 及びグラフィクスメモリ 4 3 3～4 3 4、M に記憶されているデータは、コアキャッシュ 4 6 2 A～4 6 2 D、4 5 6 及びシステムメモリ 4 4 1 とコヒーレントを保たれ得る。述べられているように、これは、キャッシュ 4 3 8 及びメモリ 4 3 3～4 3 4、M に成り代わってキャッシュコヒーレンシメカニズムに関与するプロキシ回路 4 2 5 を介して達成され得る（例えば、プロセッサキャッシュ 4 6 2 A～4 6 2 D、4 5 6 上のキャッシュラインの変更／アクセスに関連したキャッシュ 4 3 8 へアップデートを送信し、キャッシュ 4 3 8 からアップデートを受信する）。

【0086】

レジスタ 4 4 5 の組は、グラフィクス処理エンジン 4 3 1～4 3 2、N によって実行されるスレッドのコンテキストデータを記憶し、コンテキスト管理回路 4 4 8 は、スレッドコンテキストを管理する。例えば、コンテキスト管理回路 4 4 8 は、コンテキストスイッチ中に様々なスレッドのコンテキストをセーブ及びリストアするようセーブ及びリストア動作を実行してよい（例えば、第 1 スレッドはセーブされ、第 2 スレッドはストアされ、それにより、第 2 スレッドはグラフィクス処理エンジンによって実行可能である）。例えば、コンテキストスイッチで、コンテキスト管理回路 4 4 8 は、メモリ内の指定領域（例えば、コンテキストポインタによって識別される）に現在のレジスタ値を格納してよい。それは次いで、コンテキストに戻る時にレジスタ値をリストアしてよい。割り込み管理回路 4 4 7 は、例えば、システムデバイスから受け取られた割り込みを受け取って処理してよい。

【0087】

1つの実施では、グラフィクス処理エンジン431からの仮想／実効アドレスは、MMU439によってシステムメモリ441内の実質／物理アドレスに変換される。任意に、アクセラレータ集積化回路436は、複数（例えば、4個、8個、16個）のグラフィクスアクセラレータモジュール446及び／又は他のアクセラレータデバイスをサポートする。グラフィクスアクセラレータモジュール446は、プロセッサ407で実行される単一アプリケーションに専用であってよく、あるいは、複数のアプリケーション間で共有されてもよい。任意に、仮想化されたグラフィクス実行環境が提供され、そのような環境では、グラフィクス処理エンジン431～432、Nのリソースは、複数のアプリケーション又は仮想マシン（VM）と共有される。リソースは、「スライス」に細分されてもよく、

10

【0088】

よって、アクセラレータ集積化回路436は、グラフィクスアクセラレーションモジュール446のためのシステムへのブリッジとして動作し、アドレス変換及びシステムメモリキャッシュサービスを提供する。一実施形態において、ブリッジ機能性を促進するために、アクセラレータ集積化回路436はまた、共有I/O497（例えば、PCIe、USB）と、電圧、クロッキング、性能、熱、及びセキュリティのシステム制御を可能にするハードウェアとを含んでもよい。更には、アクセラレータ集積化回路436は、ホストプロセッサがグラフィクス処理エンジンの仮想化、割り込み、及びメモリ管理を管理するための仮想化ファシリティを提供し得る。

20

【0089】

グラフィクス処理エンジン431～432、Nのハードウェアリソースは、ホストプロセッサ407によってみられる実質アドレス空間へ明示的にマッピングされるので、如何なるホストプロセッサも、それらのリソースを直接に、実効アドレス値を用いてアドレッシング可能である。アクセラレータ集積化回路436の1つの任意の機能は、グラフィクスエンジン431～432、Nの物理的分離であり、それにより、それらは独立したユニットとしてシステムには見える。

【0090】

1つ以上のグラフィクスメモリ433～434、Mは、夫々、グラフィクス処理エンジン431～432、Nの夫々へ結合されてよい。グラフィクスメモリ433～434、Mは、グラフィクス処理エンジン431～432、Nの夫々によって処理される命令及びデータを記憶する。グラフィクスメモリ433～434、Mは、DRAM（スタックドDRAMを含む）、GDDRメモリ（例えば、GDDR5、GDDR6）、又はHBMなどの揮発性メモリであってよく、かつ／あるは、3D XPoint/Optane又はNanoo-Ramなどの不揮発性メモリであってもよい。

30

【0091】

高速リンク440上のデータトラフィックを低減するために、バイアシング技術が、グラフィクスメモリ433～434、Mに記憶されているデータが、グラフィクス処理エンジン431～432、Nによって最も頻繁に使用されてよく、望ましくは、コア460A～460Dによって使用されない（少なくとも頻繁にではない）データであることを確かにするよう使用される。同様に、バイアシングメカニズムは、コアのキャッシュ462A～462D、456及びシステムメモリ441内にコアによって必要とされる（望ましくは、グラフィクス処理エンジン431～432、Nには必要とされない）データを保持使用と試みる。

40

【0092】

図4Cに示される変形例に従って、アクセラレータ集積化回路436はプロセッサ407内に集積される。グラフィクス処理エンジン431～432、Nは、インターフェース437及びインターフェース435（先と同じく、任意の形式のバス又はインターフェースプロトコルを利用してよい）を介してアクセラレータ集積化回路436と高速リンク4

50

40上で直接に通信する。アクセラレータ集積化回路436は、図4Bに関して説明されたのと同じ動作を実行し得るが、コヒーレンシバス464及びキャッシュ462A～462D、456とのその近接近を考えると、潜在的に、より高いスループットで実行し得る。
【0093】

記載されている実施形態は、専用プロセスプログラミングモデル（非グラフィクスアクセラレーションモジュール仮想化）及び共有プログラミングモデル（仮想化による）を含む異なったプログラミングモデルをサポートし得る。後者は、アクセラレータ集積化回路436によって制御されるプログラミングモデルと、グラフィクスアクセラレーションモジュール446によって制御されるプログラミングモデルとを含んでよい。

【0094】

専用プロセスモデルの実施形態において、グラフィクス処理エンジン431～432、Nは、単一のオペレーティングシステムの下で単一のアプリケーション又はプロセスに専用であってよい。単一のアプリケーションは、VM／パーティション内で仮想化を提供するように、他のアプリケーションリクエストをグラフィクスエンジン431～432、Nへ向かわせることができる。

【0095】

専用プロセスプログラミングモデルでは、グラフィクス処理エンジン431～432、Nは、複数のVM／アプリケーションパーティションによって共有されてよい。共有されたモデルは、各オペレーティングシステムによるアクセスを可能にするようにグラフィクス処理エンジン431～432、Nを仮想化することをシステムハイパーバイザに要求する。ハイパーバイザなしの単一パーティションシステムについては、グラフィクス処理エンジン431～432、Nは、オペレーティングシステムによって所有される。いずれの場合にも、オペレーティングシステムは、各プロセス又はアプリケーションへのアクセスを提供するようにグラフィクス処理エンジン431～432、Nを仮想化することができる。

【0096】

共有プログラミングモデルについては、グラフィクスアクセラレーションモジュール446又は個別的なグラフィクス処理エンジン431～432、Nが、プロセスハンドルを用いてプロセス要素を選択する。プロセス要素は、システムメモリ441に記憶され、本願で記載されている実効アドレスから実質アドレスへの変換技術を用いてアドレス可能であってよい。プロセスハンドルは、そのコンテキストをグラフィクス処理エンジン431～432、Nに登録する（すなわち、プロセス要素をそのプロセス要素にリンクされたリストに加えるシステムソフトウェアを呼び出す）ときにホストプロセスへ供給される実施に固有の値であってよい。プロセスハンドルの下位16ビットは、プロセス要素にリンクされたリスト内のプロセス要素のオフセットであってよい。

【0097】

図4Dは、例となるアクセラレータ集積化スライス490を表す。本願で使用されているように、「スライス」は、アクセラレータ集積化回路436の処理リソースの指定された部分を有する。システムメモリ441内のアプリケーション実効アドレス空間482は、プロセス要素483を記憶する。プロセス要素483は、プロセッサ407で実行されているアプリケーション480からのGPU起動481に応答して記憶されてよい。プロセス要素483は、対応するアプリケーション480のプロセス状態を含む。プロセス要素483に含まれる作業記述子（work descriptor, WD）484は、アプリケーションによって要求される単一のジョブであることができ、あるいは、ジョブのキューへのポインタを含んでもよい。後者の場合に、WD484は、アプリケーションのアドレス空間482内のジョブリクエストキューへのポインタである。

【0098】

グラフィクスアクセラレーションモジュール446及び／又は個別的なグラフィクス処理エンジン431～432、Nは、システム内のプロセスの全て又はサブセットによって共有され得る。例えば、本願で記載されている技術は、プロセス状態をセットアップし、WD484をグラフィクスアクセラレーションモジュール446へ送って、仮想化された

10

20

30

40

50

環境でジョブを開始するインフラストラクチャを含んでよい。

【0099】

1つの実施において、専用プロセスプログラミングモデルは、実施に固有である。このモデルでは、単一のプロセスがグラフィクスアクセラレーションモジュール446又は個別的なグラフィクス処理エンジン431を所有する。グラフィクスアクセラレーションモジュール446が単一のプロセスによって所有されるので、ハイパーバイザは、所有パーティションのためにアクセラレータ集積化回路436を初期化し、オペレーティングシステムは、グラフィクスアクセラレーションモジュール446が割り当てられる時点で、所有パーティションのためにアクセラレータ集積化回路436を初期化する。

【0100】

動作中に、アクセラレータ集積化回路436内のWDフェッチユニット491は、グラフィクスアクセラレーションモジュール446のグラフィクス処理エンジンの1つによって行われるべき作業の指示を含む次のWD484をフェッチする。WD484からのデータは、レジスタ445に格納され、表されているようにMMU439、割り込み管理回路447及び／又はコンテキスト管理回路448によって使用されてよい。例えば、MMU439は、OS仮想アドレス空間485内のセグメント／ページテーブル486にアクセスするためのセグメント／ページウォーク回路を含んでよい。割り込み管理回路447は、グラフィクスアクセラレーションモジュール446から受け取られた割り込みイベント492を処理してよい。グラフィクス操作を実行するときに、グラフィクス処理エンジン431～432、Nによって生成された実効アドレス493は、MMU439によって実質アドレスに変換される。

【0101】

レジスタ445の同じ組が、各グラフィクス処理エンジン431～432、N及び／又はグラフィクスアクセラレーションモジュール446について複製されてよく、ハイパーバイザ又はオペレーティングシステムによって初期化されてよい。それらの複製されたレジスタの夫々は、アクセラレータ集積化スライス490に含まれてよい。ハイパーバイザによって初期化され得る例示的なレジスタは、表1に示される。

【表1】

表1 - ハイパーバイザ初期化レジスタ

1	スライス制御レジスタ
2	実質アドレス(RA)によりスケジューリングされたプロセスエリアのポインタ
3	権限マスクオーバーライドレジスタ
4	割り込みベクトルテーブルエントリオフセット
5	割り込みベクトルテーブルエントリ制限
6	状態レジスタ
7	論理パーティション ID
8	実質アドレス(RA)ハイパーバイザアクセラレータ利用記録ポインタ
9	ストレージ記述レジスタ

【0102】

オペレーティングシステムによって初期化され得る例示的なレジスタは、表2に示され

る。

【表 2】

表2 - オペレーティングシステム初期化レジスタ

1	プロセス及びスレッド識別
2	実効アドレス(EA)コンテキストセーブ／リストアポインタ
3	仮想アドレス(VA)アクセラレータ利用記録ポインタ
4	仮想アドレス(VA)ストレージセグメントテーブルポインタ
5	権限マスク
6	作業記述子

10

【0103】

各WD 484は、特定のグラフィクスアクセラレーションモジュール446及び／又はグラフィクス処理エンジン431～432、Nに特有であってよい。それは、グラフィクス処理エンジン431～432、Nがその作業を行うために必要とする全ての情報を含むか、あるいは、それは、アプリケーションが完了されるべき作業のコマンドキューをセットアップしているメモリ位置へのポインタであることができる。

20

【0104】

図4Eは、共有モデルの更なる任意の詳細を表す。それは、プロセス要素リスト499が記憶されているハイパーバイザ実質アドレス空間498を含む。ハイパーバイザ実質アドレス空間498は、オペレーティングシステム495のためのグラフィクスアクセラレーションモジュールエンジンを仮想化するハイパーバイザ496を介してアクセス可能である。

【0105】

共有プログラミングモデルは、システム内のパーティションの全て又はサブセットからのプロセスの全て又はサブセットがグラフィクスアクセラレーションモジュール446を使用することを可能にする。グラフィクスアクセラレーションモジュール446が複数のプロセス及びパーティションによって共有される2つのプログラミングモデル、つまり、時間スライス共有及びグラフィクス指向共有がある。

30

【0106】

このモデルでは、システムハイパーバイザ496がグラフィクスアクセラレーションモジュール446を所有し、その機能を全てのオペレーティングシステム495に利用可能にする。グラフィクスアクセラレーションモジュール446がシステムハイパーバイザ496による仮想化をサポートするために、グラフィクスアクセラレーションモジュール446は、次の要件に従ってよい：1) アプリケーションのジョブリクエストは、自律的でなければならない（すなわち、状態がジョブ間で維持される必要はない）、あるいは、グラフィクスアクセラレーションモジュール446は、コンテキストセーブ及びリストアメカニズムを提供しなければならない。2) アプリケーションのジョブリクエストは、グラフィクスアクセラレーションモジュール446によって、如何なる変換不良も含めて指定された時間量で完了することが保証され、あるいは、グラフィクスアクセラレーションモジュール446は、ジョブの処理をプリエンプトする能力を提供する。3) グラフィクスアクセラレーションモジュール446は、指向的な共有プログラミングモデルで動作するときにプロセス間の公平性を保証されなければならない。

40

50

【0107】

共有モデルについては、アプリケーション480が、グラフィクスアクセラレーションモジュール446のタイプ、作業記述子(WD)、権限マスクレジスタ(authority mask register, AMR)値、及びコンテキストセーブ／リストアエリアポインタ(context save/restore area pointer, CSRP)を用いてオペレーティングシステム495のシステム呼び出しを行うために必要とされ得る。グラフィクスアクセラレーションモジュール446のタイプは、システム呼び出しの対象となるアクセラレーション機能を記述する。グラフィクスアクセラレーションモジュール446のタイプは、システム固有の値であってよい。WDは、グラフィクスアクセラレーションモジュール446のために特にフォーマット化され、グラフィクスアクセラレーションモジュール446のコマンド、ユーザにより定義された構造への実効アドレスポインタ、コマンドのキューへの実効アドレスポインタ、又はグラフィクスアクセラレーションモジュール446によって行われるべき作業を記述する何らかの他のデータ構造の形を取ることができる。一実施形態において、AMR値は、現在のプロセスのために使用すべきAMR状態である。オペレーティングシステムへ渡される値は、AMRを設定するアプリケーションと同様である。アクセラレータ集積化回路436及びグラフィクスアクセラレーションモジュール446の実装がユーザ権限マスクオーバーライドレジスタ(User Authority Mask Override Register, UAMOR)をサポートしない場合には、オペレーティングシステムは、ハイパーバイザ呼び出しにおいてAMRを渡す前に現在のUAMOR値をAMR値に適用してもよい。ハイパーバイザ496は、任意に、AMRをプロセス要素483に置く前に現在のAMOR (Authority Mask Override Register) 値を適用してもよい。CSRPは、グラフィクスアクセラレーションモジュール446がコンテキスト状態をセーブ及びリストアするためにアプリケーションのアドレス空間482内のエリアの実効アドレスを含むレジスタ445の1つであってよい。このポインタは、ジョブ間で状態がセーブされる必要がない場合に、あるいは、ジョブがブリエンプトされる場合に、任意である。コンテキストセーブ／リストアエリアは、ピン留めされたシステムメモリであってよい。

【0108】

システム呼び出しを受け取ると、オペレーティングシステム495は、アプリケーション480が登録されており、グラフィクスアクセラレーションモジュール446を使用する権限を与えられていることを確かめてよい。次いで、オペレーティングシステム495は、表3に示される情報を用いてハイパーバイザ496を呼び出す。

【表3】

表3 - OS からハイパーバイザを呼び出すパラメータ

1	作業記述子(WD)
2	権限マスクレジスタ(AMR) 値(マスキングされている場合がある)
3	実効アドレス(EA)コンテキストセーブ／リストアエリアポインタ(CSRP)
4	プロセスID(PID)及び任意のスレッドID(TID)
5	仮想アドレス(VA)アクセラレータ利用記録ポインタ(AURP)
6	ストレージセグメントテーブルポインタ(SSTP)の仮想アドレス
7	論理割り込みサービス番号(LISN)

【0109】

ハイパーバイザ呼び出しを受け取ると、ハイパーバイザ496は、オペレーティングシ

ステム 4 9 5 が登録されており、グラフィクスアクセラレーションモジュール 4 4 6 を使用する権限を与えられていることを確かめる。次いで、ハイパーバイザ 4 9 6 は、対応するグラフィクスアクセラレーションモジュール 4 4 6 のタイプについてプロセス要素 4 8 3 をプロセス要素にリンクされたリスト内に置く。プロセス要素は、表 4 に示される情報を含んでよい。

【表 4】

表4 - プロセス要素情報

1	作業記述子 (WD)	10
2	権限マスクレジスタ (AMR) 値 (マスキングされている場合がある)	
3	実効アドレス (EA) コンテキストセーブ／リストアエリアポインタ (CSRP)	
4	プロセス ID (PID) 及び任意のスレッド ID (TID)	
5	仮想アドレス (VA) アクセラレータ利用記録ポインタ (AURP)	
6	ストレージセグメントテーブルポインタ (SSTP) の仮想アドレス	20
7	論理割り込みサービス番号 (LISN)	
8	ハイパーバイザ呼び出しパラメータから導出された割り込みベクトルテーブル	
9	状態レジスタ (SR) 値	
10	論理パーティション ID (LPID)	
11	実質アドレス (RA) ハイパーバイザアクセラレータ利用記録ポインタ	30
12	ストレージセグメントテーブルポインタ (SDR)	

【0 1 1 0】

ハイパーバイザは、複数のアクセラレータ集積化スライス 4 9 0 のスライス 4 4 5 を初期化してよい。

【0 1 1 1】

図 4 F に表されているように、1 つの任意の実施において、物理プロセッサメモリ 4 0 1 ~ 4 0 2 及び GPU メモリ 4 2 0 ~ 4 2 3 にアクセスするために使用される共通仮想メモリアドレス空間を介してアドレス可能な統合メモリが用いられる。この実施では、GPU 4 1 0 ~ 4 1 3 で実行される動作は、プロセッサメモリ 4 0 1 ~ 4 0 2 にアクセスするために同じ仮想／実効メモリアドレス空間を利用し、逆もまた同様であり、それによって、プログラム可能性を簡素化する。仮想／実効アドレス空間の第 1 部分は、プロセッサメモリ 4 0 1 に割り当てられてよく、第 2 部分は、第 2 プロセッサメモリ 4 0 2 に割り当てられてよく、第 3 部分は、GPU メモリ 4 2 0 に割り当てられよく、その他同様である。全体の仮想／実効メモリ空間（時々、実効アドレス空間と呼ばれる）は、それによって、プロセッサメモリ 4 0 1 ~ 4 0 2 及び GPU メモリ 4 2 0 ~ 4 2 3 の夫々にわたって分配され、いずれかのプロセッサ又は GPU が、いずれかの物理メモリに、そのメモリにマッピングされている仮想アドレスを用いてアクセスすることを可能にし得る。

【0 1 1 2】

MMU 4 3 9 A ~ 4 3 9 E のうちの 1 つ以上の中にあるバイアス／コヒーレンス管理回

路 4 9 4 A ~ 4 9 4 E が提供され、それは、ホストプロセッサ（例えば、4 0 5）のキャッシュと GPU 4 1 0 ~ 4 1 3 との間のキャッシュコヒーレンスを確認にし、特定のタイプのデータが記憶されるべきである物理メモリを示すバイアシング技術を実装する。バイアス／コヒーレンス管理回路 4 9 4 A ~ 4 9 4 E の複数のインスタンスが図 4 F に表されている一方で、バイアス／コヒーレンス回路は、1 つ以上のホストプロセッサ 4 0 5 の MMU 内に、及び／又はアクセラレータ集積化回路 4 3 6 内に実装されてもよい。

【0 1 1 3】

GPU 付属メモリ 4 2 0 ~ 4 2 3 は、システムメモリの部分としてマッピングされ、共有仮想メモリ（SVM）技術を用いて、しかし、完全システムキャッシュコヒーレンスに関連した典型的な性能欠点に苦しまずにアクセスされ得る。GPU 付属メモリ 4 2 0 ~ 4 2 3 が面倒なキャッシュコヒーレンスオーバーヘッドなしでシステムメモリとしてアクセスされる能力は、GPU オフロードのための有利な動作環境をもたらす。この配置は、ホストプロセッサ 4 0 5 のソフトウェアが、従来の I/O DMA データコピーのオーバーヘッドなしで、オペランドをセットアップし、計算結果にアクセスすることを可能にする。そのような従来のコピーは、ドライバ呼び出し、割り込み及びメモリマッピング I/O（memory mapped I/O, MMIO）アクセスを必要とし、これらは全て、簡単なメモリアクセスに対して非効率的である。同時に、キャッシュコヒーレンスオーバーヘッドなしで GPU 付属メモリ 4 2 0 ~ 4 2 3 にアクセスする能力は、オフロードされた計算の実行時間にとって重要であり得る。ストリーミング書き込みメモリトラフィックが相当量である場合に、例えば、キャッシュコヒーレンスオーバーヘッドは、GPU 4 1 0 ~ 4 1 3 によって見られる実効書き込みバンド幅を大幅に減らすことができる。オペランドセットアップの効率、結果アクセスの効率、及び GPU 計算の効率は全て、GPU オフロードの有効性を決定することにおいて役割を果たす。

【0 1 1 4】

GPU バイアスとホストプロセッサバイアスとの間の選択は、バイアストラッカデータ構造によって駆動されてよい。例えば、バイアステーブルが使用されてよい。バイアステーブルは、GPU 付属メモリページごとに 1 又は 2 ビットを含むページ粒度構造であってよい（すなわち、メモリページの粒度で制御される）。バイアステーブルは、（例えば、バイアステーブルの頻繁に／最近使用されたエントリをキャッシュするための）GPU 4 1 0 ~ 4 1 3 内のバイアスキャッシュの有無によらず、1 つ以上の GPU 付属メモリ 4 2 0 ~ 4 2 3 のスチールされたメモリ範囲で実装されてよい。代替的に、エントリバイアステーブルは GPU 内で保持されてもよい。

【0 1 1 5】

1 つの実施において、GPU 付属メモリ 4 2 0 ~ 4 2 3 への各アクセスに関連したバイアステーブルエントリは、GPU メモリへの実際のアクセスの前にアクセスされ、次の動作を引き起こす。最初に、GPU 4 1 0 ~ 4 1 3 からの、GPU バイアスにおいてそれらのページを見つけるローカルリクエストが、対応する GPU メモリ 4 2 0 ~ 4 2 3 へ直接に転送される。GPU からの、ホストバイアスにおいてそれらのページを見つけるローカルリクエストは、プロセッサ 4 0 5 へ（例えば、上述された高速リンク上で）転送される。任意に、プロセッサ 4 0 5 からの、要求されたページをホストプロセッサバイアスにおいて見つけるリクエストは、通常のメモリ読み出しのようなリクエストを完了する。代替的に、GPU バイアスページに向けられたリクエストは、GPU 4 1 0 ~ 4 1 3 へ転送されてもよい。GPU は、次いで、ホストプロセッサがページを現在使用中でない場合に、そのページをホストプロセッサへ移してよい。

【0 1 1 6】

ページのバイアス状態は、ソフトウェアベースメカニズム若しくはハードウェア支援型ソフトウェアベースメカニズムによって、又は限られた場合の組については、単にハードウェアベースのメカニズムによって、変更され得る。

【0 1 1 7】

バイアス状態を変更するための 1 つのメカニズムは、API コール（例えば、Open

10

20

30

40

50

CL)を用いる。これは、次に、GPUのデバイスドライバを呼び出し、これは、次に、バイアス状態を変更するようそれに指示するGPUヘッメッセージを送信し（又はコマンド記述子をエンキューし）、いくつかの遷移については、ホストにおいてキャッシュフラッシュ動作を実行する。キャッシュフラッシュ動作は、ホストプロセッサ405のバイアスからGPUのバイアスへの遷移のために必要とされるが、逆の遷移のためには必要とされない。

【0118】

キャッシュコヒーレンシは、GPUバイアスページをホストプロセッサ405によって一時的にキャッシュ不能にすることによって維持され得る。それらのページにアクセスするために、プロセッサ405は、実装に応じて、直ぐにアクセスを許可する場合も許可しない場合もあるGPU410からのアクセスを要求してよい。よって、ホストプロセッサ405とGPU410との間の通信を減らすために、GPUバイアスページが、GPUによって必要とされるがホストプロセッサ405によっては必要とされないページであり、その逆も同様であることを確かにすることは有益である。

【0119】

[グラフィクス処理パイプライン]

図5は、グラフィクス処理パイプライン500を表す。図2Dで見られるようなグラフィクスマルチプロセッサ234、図3Aで見られるようなグラフィクスマルチプロセッサ325、図3Bで見られるようなグラフィクスマルチプロセッサ350などのグラフィクスマルチプロセッサが、表されているグラフィクス処理パイプライン500を実装することができる。グラフィクスマルチプロセッサは、図1の並列プロセッサ112に関係があってもよく、それらのうちの1つの代わりに使用されてよい図2Aの並列プロセッサ200などの、本願で記載されている並列処理サブシステム内に含まれ得る。様々な並列処理システムは、ここで記載されるような並列処理ユニット（例えば、図2Aの並列処理ユニット202）の1つ以上のインスタンスを介してグラフィクス処理パイプライン500を実装することができる。例えば、シェーダユニット（例えば、図2Cのグラフィクスマルチプロセッサ234）は、頂点処理ユニット504、テッセレーション制御処理ユニット508、テッセレーション評価処理ユニット512、ジオメトリ処理ユニット516、及びフラグメント／ピクセル処理ユニット524のうちの1つ以上の機能を実行するよう構成されてよい。データアセンブラ502、プリミティブアセンブラ506、514、518、テッセレーションユニット510、ラスタライザ522、及びラスタ動作ユニット526の機能はまた、処理クラスタ（例えば、図2Aの処理クラスタ214）及び対応するパーティションユニット（例えば、図2Aのパーティションユニット220A～220N）内の他の処理エンジンによって実行されてもよい。グラフィクス処理パイプライン500はまた、1つ以上の機能のための専用の処理ユニットを用いて実装されてもよい。グラフィクス処理パイプライン500の1つ以上の部分が、汎用プロセッサ（例えば、CPU）内の並列処理ロジックによって実行されることも可能である。任意に、グラフィクス処理パイプライン500の1つ以上の部分は、図2Aのメモリインターフェース218のインスタンスであってよいメモリインターフェース528を介してオンチップメモリ（例えば、図2Aで見られるような並列プロセッサメモリ）にアクセスすることができる。グラフィクスプロセッサパイプライン500はまた、図3Cで見られるようなマルチコアグループ365Aを介して実装されてもよい。

【0120】

データアセンブラ502は、表面の頂点データ及びプリミティブを収集し得る処理ユニットである。データアセンブラ502は、次いで、頂点属性を含む頂点データを頂点処理ユニット504へ出力する。頂点処理ユニット504は、頂点シェーダプログラムを実行するプログラム可能な実行ユニットであり、頂点シェーダプログラムによって指定された頂点データをライティング及び変換する。頂点処理ユニット504は、頂点データを処理する際の使用のために、キャッシュ、ローカル又はシステムメモリに記憶されているデータを読み出し、オブジェクトベースの座標表現から世界空間座標空間又は正規化されたデ

10

20

30

40

50

バース座標空間へ頂点データを変換するようプログラムされてよい。

【0121】

プリミティブアセンブラ506の第1インスタンスは、頂点処理ユニット504から頂点属性を受け取る。プリミティブアセンブラ506は、必要に応じて、記憶されている頂点属性を読み出し、テッセレーション制御処理ユニット508による処理のためにグラフィクスプリミティブを構成する。グラフィクスプリミティブは、様々なグラフィクス処理アプリケーションプログラミングインターフェース（API）によってサポートされているトライアングル、ラインセグメント、ポイント、パッチ、などを含む。

【0122】

テッセレーション制御処理ユニット508は、入力された頂点を幾何学的パッチのための制御ポイントとして扱う。制御ポイントは、パッチ（例えば、パッチベース）からの入力表現から、テッセレーション評価処理ユニット512による適切な評価での使用に適している表現へ変換される。テッセレーション制御処理ユニット508はまた、幾何学的パッチのエッジのテッセレーション係数を計算することもできる。テッセレーション係数は、単一のエッジに適用され、エッジに関連したディテールのビュー依存レベルを定量化する。テッセレーションユニット510は、パッチのエッジのテッセレーション係数を受け取り、直線、三角形、又は四辺形プリミティブなどの複数の幾何学的プリミティブにパッチをモザイク化するように構成される。複数の幾何学的プリミティブは、テッセレーション評価処理ユニット512へ送られる。テッセレーション評価処理ユニット512は、幾何学的プリミティブに関連した各頂点の曲面表現及び頂点属性を生成するよう、細分されたパッチのパラメータ化された座標に作用する。

【0123】

プリミティブアセンブラ514の第2インスタンスは、記憶されている頂点属性を必要に応じて読み出すテッセレーション評価処理ユニット512から頂点属性を受け取り、ジオメトリ処理ユニット516による処理のためのグラフィクスプリミティブを構成する。ジオメトリ処理ユニット516は、ジオメトリシェーダプログラムによって指定されるようにプリミティブアセンブラ514から受け取られたグラフィクスプリミティブを変換するようジオメトリシェーダプログラムを実行するプログラム可能な実行ユニットである。ジオメトリ処理ユニット516は、グラフィクスプリミティブを1つ以上の新しいグラフィクスプリミティブに細分し、新しいグラフィクスプリミティブをラスタライズするために使用されるパラメータを計算するようプログラムされてよい。

【0124】

ジオメトリ処理ユニット516は、ジオメトリストリーム内の要素を追加又は削除することが可能であり得る。ジオメトリ処理ユニット516は、新しいグラフィクスプリミティブを特定するパラメータ及び頂点をプリミティブアセンブラ518へ出力する。プリミティブアセンブラ518は、ジオメトリ処理ユニット516からパラメータ及び頂点を受け取り、ビューポートスケール、カル（cull）、及びクリップ（clip）ユニット520による処理のためにグラフィクスプリミティブを構成する。ジオメトリ処理ユニット516は、ジオメトリデータを処理する際の使用のために並列プロセッサメモリ又はシステムメモリに記憶されているデータを読み出す。ビューポートスケール、カル、及びクリップユニット520は、クリッピング、カリング、及びビューポートスケーリングを実行し、処理されたグラフィクスプリミティブをラスタライザ522へ出力する。

【0125】

ラスタライザ522は、デプスカリング及び他のデプススペースの最適化を実行することができる。ラスタライザ522はまた、フラグメントを生成し、これらのフラグメント及び関連するカバレッジデータをフラグメント／ピクセル処理ユニット524へ出力するよう、新しいグラフィクスプリミティブに対してスキャン変換を実行する。フラグメント／ピクセル処理ユニット524は、フラグメントシェーダプログラム又はピクセルシェーダプログラムを実行するよう構成されるプログラム可能な実行ユニットである。フラグメント／ピクセル処理ユニット524は、フラグメント又はピクセルシェーダプログラムによ

って指定されているように、ラスタライザ 5 2 2 から受け取られたフラグメント又はピクセルを変換する。例えば、フラグメント／ピクセル処理ユニット 5 2 4 は、ラスタ動作ユニット 5 2 6 へ出力される陰影付けされたフラグメント又はピクセルを生成するよう、テクスチャマッピング、シェーディング、ブレンディング、テクスチャ補正及びパースペクティブ補正 (perspective correction) を含むがこれらに限られない動作を実行するようプログラムされてよい。フラグメント／ピクセル処理ユニット 5 2 4 は、フラグメントデータを処理するときの使用のために、並列プロセッサメモリ又はシステムメモリのどちらかに記憶されているデータを読み出すことができる。フラグメント又はピクセルシェーダプログラムは、処理ユニットのために設定されたサンプリングレートに応じたサンプル、ピクセル、タイル、又は他の粒度でシェーディングを行うよう構成されてよい。

10

【0126】

ラスタ動作ユニット 5 2 6 は、ステンシル、z テスト、ブレンディング、などを含むがこれらに限られないラスタ動作を実行し、ピクセルデータを、グラフィクスメモリ (例えば、図 2 A の並列プロセッサメモリ 2 2 2、及び／又は図 1 のシステムメモリ 1 0 4) に記憶されるように、1 つ以上の表示デバイス 1 1 0 で表示されるように、又は 1 つ以上のプロセッサ 1 0 2 若しくは並列プロセッサ 1 1 2 のうちの 1 つによる更なる処理のために、処理されたグラフィクスデータとして出力する処理ユニットである。ラスタ動作ユニット 5 2 6 は、メモリに書き込まれる z 又はカラーデータを圧縮し、メモリから読み出される z 又はカラーデータを圧縮解除するよう構成されてよい。

20

【0127】

〔機械学習概要〕

上記のアーキテクチャは、機械学習モデルを使用して訓練及び推論動作を実行するよう適用され得る。機械学習は、多くの種類のタスクの解決に成功している。機械学習アルゴリズム (例えば、ニューラルネットワーク) を訓練及び使用するときには生じる計算は、効率的な並列実装に必然的に役に立つ。従って、汎用グラフィック処理ユニット (G P G P U) などの並列プロセッサは、ディープニューラルネットワークの実際の実装において重要な役割を果たしている。S I M T (single instruction, multiple thread) アーキテクチャを備えた並列グラフィクスプロセッサは、グラフィクスパイプラインでの並列処理の量を最大にするよう設計される。S I M T アーキテクチャでは、並列スレッドのグループは、処理効率を高めるよう可能な限り頻繁に一緒に同期してプログラム命令を実行しようと試みる。並列な機械学習アルゴリズム実装によってもたらされる効率は、高容量ネットワークの使用を可能にし、それらのネットワークがより大きいデータセットに対して訓練されることを可能にする。

30

【0128】

機械学習アルゴリズムは、データの組に基づいて学習することができる。例えば、機械学習アルゴリズムは、データセット内のハイレベル抽象化をモデル化するよう設計され得る。例えば、画像認識アルゴリズムは、所与の入力がいくつかのカテゴリのうちのどれに属しているかを決定するために使用可能であり、回帰アルゴリズムは、入力が与えられると数値を出力することができ、パターン認識アルゴリズムは、変換されたテキストを生成したり、あるいは、テキストからの発話認識及び／又は発話認識を実行したりするために使用可能である。

40

【0129】

機械学習アルゴリズムの例となるタイプは、ニューラルネットワークである。多くの種類のニューラルネットワークがあり、ニューラルネットワークの簡単なタイプはフィードフォワードネットワークである。フィードフォワードネットワークは、ノードがレイヤに配置されている非周期グラフとして実装されてよい。典型的に、フィードフォワードネットワークトポロジは、少なくとも 1 つの隠れレイヤによって分離されている入力レイヤ及び出力レイヤを含む。隠れレイヤは、入力レイヤによって受け取られた入力を、出力レイヤで出力を生成するために有用である表現に変換する。ネットワークノードは、隣接レイヤ内のノードへエッジを介して完全に接続されるが、各レイヤ内のノード間にエッジは存

50

在しない。フィードフォワードネットワークの入力レイヤのノードで受け取られたデータは、レイヤを接続するエッジの夫々に各々関連付けられた係数（「重み」）に基づいてネットワーク内の各連続レイヤのノードの状態を計算する活性化関数により出力レイヤのノードへ伝搬される（すなわち、前方に送られる）。実行されているアルゴリズムによって表現される具体的なモデルに応じて、ニューラルネットワークアルゴリズムからの出力は、様々な形を取ることができる。

【0130】

機械学習アルゴリズムが特定の問題をモデル化するために使用され得る前に、アルゴリズムは、訓練データセットを用いて訓練される。ニューラルネットワークの訓練は、ネットワークトポジを選択すること、ネットワークによってモデル化される問題を表す訓練データの組を使用すること、及びネットワークモデルが訓練データセットの全てのインスタンスについて最小限のエラーで実行するまで重みを調整することを必要とする。例えば、ニューラルネットワークのための教師あり学習訓練プロセス中に、訓練データセット内のインスタンスを表す入力に回答してネットワークによって生成された出力は、そのインスタンスについての「正しい」とラベル付けされた出力と比較され、出力とラベル付き出力との間の差を表すエラー信号が計算され、接続に関連した重みは、エラー信号がネットワークレイヤを通過して後方伝搬されるということで、そのエラーを最小限にすべく調整される。ネットワークは、訓練データセットのインスタンスから生成された出力の夫々についてのエラーが最小限にされる場合に、「訓練済み」と見なされる。

【0131】

機械学習アルゴリズムの精度は、アルゴリズムを訓練するために使用されるデータセットの品質によって大いに影響を及ぼされ得る。訓練プロセスは、計算量が多く、従来の汎用プロセッサでは相当量の時間を必要とする可能性がある。従って、並列処理ハードウェアが、多くのタイプの機械学習アルゴリズムを訓練するために使用される。これは、ニューラルネットワークで係数を調整する際に実行される計算が並列実装に必然的に役に立つということで、ニューラルネットワークの訓練を最適化することによって特に有用である。具体的に、多くの機械学習アルゴリズム及びソフトウェアアプリケーションは、汎用のグラフィクス処理デバイス内の並列処理ハードウェアを使用するよう適応されている。

【0132】

図6は、機械学習ソフトウェアスタック600の一般化された図である。機械学習アプリケーション602は、訓練データセットを使用してニューラルネットワークを訓練するよう、あるいは、訓練されたディープニューラルネットワークを使用して機械学習インテリジェンスを実装するよう構成され得る。機械学習アプリケーション602は、ニューラルネットワークのための訓練及び推論機能、及び／又はデプロイ前にニューラルネットワークを訓練するために使用され得る特殊化したソフトウェアを含むことができる。機械学習アプリケーション602は、画像認識、マッピング及びローカライゼーション、自律ナビゲーション、発話合成、医療撮像、又は言語変換を含むがこれらに限られない如何なるタイプのマシンインテリジェンスも実装することができる。

【0133】

機械学習アプリケーション602のためのハードウェアアクセラレーションは、機械学習フレームワーク604を介して有効にされ得る。機械学習フレームワーク604は、機械学習プリミティブのライブラリを提供することができる。機械学習プリミティブは、機械学習アルゴリズムによって一般に実行される基本操作である。機械学習フレームワーク604によらないと、機械学習アルゴリズムの開発者は、機械学習アルゴリズムに関連した主な計算ロジックを作成し最適化し、次いで、新しい並列プロセッサ開発される場合には計算ロジックを最適化し直すよう求められることになる。代わりに、機械学習アプリケーションは、機械学習フレームワーク604によって提供されたプリミティブを用いて、必要な計算を実行するよう構成され得る。例となるプリミティブは、テンソル畳み込み、活性化関数、及びプーリングを含み、これらは、畳み込みニューラルネットワーク（convolutional neural network, CNN）を訓練している間に実行される計算操作である

。機械学習フレームワーク604はまた、行列及びベクトル演算などの、多くの機械学習アルゴリズムによって実行される基本線形代数サブプログラムを実装するために、プリミティブを供給することができる。

【0134】

機械学習フレームワーク604は、機械学習アプリケーション602から受け取られた入力データを処理し、計算フレームワーク606への適切な入力を生成することができる。計算フレームワーク606は、GPGPUハードウェア610のアーキテクチャに関する徹底的な知識を有することを機械学習フレームワーク604に求めずに、GPGPUハードウェア610を介してハードウェアアクセラレーションを利用することを機械学習フレームワーク604に可能にするよう、GPGPUドライバ608へ供給される基礎となる命令を抽象化することができる。更には、計算フレームワーク606は、GPGPUハードウェア610の様々なタイプ及び生成にわたって機械学習フレームワーク604のためにハードウェアアクセラレーションを可能にすることができる。

【0135】

[GPGPU機械学習アクセラレーション]

図7は、図2Aの並列プロセッサ200又は図1の並列プロセッサ112であってよい汎用グラフィクス処理ユニット700を表す。汎用処理ユニット(GPGPU)700は、ディープニューラルネットワークの訓練に関連したタイプの計算ワークロードを処理することにおいて特に効率的であるよう構成されてよい。更には、GPGPU700は、特にディープニューラルネットワークのために訓練速度を改善するためにマルチGPUクラスタを生成するようGPGPUの他のインスタンスへ直接にリンクされ得る。

【0136】

GPGPU700は、ホストプロセッサとの接続を可能にするホストインターフェース702を含む。ホストインターフェース702は、PCI Expressインターフェースであってよい。しかし、ホストインターフェースはまた、ベンダー特有の通信インターフェース又は通信ファブリックであることもできる。GPGPU700は、ホストプロセッサからコマンドを受け取り、グローバルスケジューラ704を使用して、それらのコマンドに関連した実行スレッドを処理クラスタ706A~706Hの組へ分配する。処理クラスタ706A~706Hは、キャッシュメモリ708を共有する。キャッシュメモリ708は、計算クラスタ706A~706H内のキャッシュメモリのためのより上位のキャッシュとして働くことができる。表されている処理クラスタ706A~706Hは、図2Aで見られるような処理クラスタ214A~214Nと対応してよい。

【0137】

GPGPU700は、メモリコントローラ712A~712Bの組を介して処理クラスタ706A~706Hと結合されたメモリ714A~714Bを含む。メモリ714A~Bは、グラフィクスダブルデータレート(GDDR)メモリを含む同期グラフィクス・ランダム・アクセス・メモリ(SGRAM)などの動的ランダム・アクセス・メモリ(DRAM)又はグラフィクス・ランダム・アクセス・メモリを含む様々なタイプのメモリデバイスを含むことができる。メモリ714A~714Bはまた、高バンド幅メモリ(HBM)を含むがこれに限られない3Dスタックドメモリも含んでよい。

【0138】

処理クラスタ706A~706Hの夫々は、図2Dのグラフィクスマルチプロセッサ234、図3Aのグラフィクスマルチプロセッサ325、図3Bのグラフィクスマルチプロセッサ350などのグラフィクスマルチプロセッサの組を含んでよく、あるいは、図3Cで見られたようなマルチコアグループ365A~365Nを含んでよい。計算クラスタのグラフィクスマルチプロセッサは、機械学習計算に適したものを含む精度の範囲で計算動作を実行することができる複数のタイプの整数及び浮動小数点ロジックユニットを含む。例えば、処理クラスタ706A~706Hの夫々における浮動小数点ユニットの少なくともサブセットは、16ビット又は32ビットの浮動小数点演算を実行するよう構成され得る一方で、浮動小数点ユニットの別のサブセットは、64ビット浮動小数点演算を実行す

10

20

30

40

50

るよう構成され得る。

【0139】

GPGPU700の複数のインスタンスは、計算クラスタとして動作するよう構成され得る。同期化及びデータ交換のために計算クラスタによって使用される通信メカニズムは、実施形態ごとに様々である。例えば、GPGPU700の複数のインスタンスは、ホストインターフェース702を介して通信する。一実施形態において、GPGPU700は、GPGPUの他のインスタンスへの直接接続を可能にするGPUリンク710とGPGPU700を結合するI/Oハブ709を含む。GPUリンク710は、GPGPU700の複数のインスタンスの間の通信及び同期化を可能にする専用のGPU間ブリッジへ結合されてよい。任意に、GPUリンク710は、データを他のGPGPU又は並列プロセッサへ送信及び受信するよう高速インターコネクトと結合する。GPGPU700の複数のインスタンスは、別のデータ処理システムに位置し、ホストインターフェース702によりアクセス可能なネットワークデバイスを介して通信してよい。GPUリンク710は、ホストインターフェース702に加えて又はその代わりとしてホストプロセッサへの接続を可能にするよう構成されてよい。

10

【0140】

GPGPU700の表されている構成は、ニューラルネットワークを訓練するよう構成され得る一方で、GPGPU700の代替の構成は、高性能又は低電力の推論プラットフォーム内のデプロイのために構成され得る。推論構成では、GPGPU700は、訓練構成と比べて、それに含まれている処理クラスタ706A～706Hが少ない。更には、メモリ714A～714Bに関連したメモリ技術は、推論構成と訓練構成との間とは異なってもよい。一実施形態において、GPGPU700の推論構成は、特定の命令を推論することをサポートすることができる。例えば、推論構成は、デプロイされたニューラルネットワークのために推論動作中に一般的に使用される1つ以上の8ビット整数内積命令をサポートすることができる。

20

【0141】

図8は、マルチGPUコンピューティングシステム800を表す。マルチGPUコンピューティングシステム800は、ホストインターフェーススイッチ804を介して複数のGPGPU806A～806Bへ結合されているプロセッサ802を含むことができる。ホストインターフェーススイッチ804は、プロセッサ802をPCIEクスプレスバスへ結合するPCIEクスプレススイッチデバイスであってよい。PCIEクスプレスバスを介して、プロセッサ802は、GPGPU806A～806Dの組と通信することができる。複数のGPGPU806A～806Dの夫々は、図7のGPGPU700のインスタンスであることができる。GPGPU806A～806Dは、高速ポイント・ツー・ポイントGPU間リンク816の組を介して相互接続することができる。高速GPU間リンクは、図7のGPUリンク710などの専用のGPUリンクを介してGPGPU806A～806Dの夫々へ接続することができる。P2P GPUリンク816は、プロセッサ802が接続されるホストインターフェースバス上での通信を必要とせずに、GPGPU806A～806Dの夫々の間の直接通信を可能にする。P2P GPUリンクへ向けられたGPU間トラフィックによれば、ホストインターフェースバスは、例えば、1つ以上のネットワークデバイスを介して、システムメモリアクセスのために、又はマルチGPUコンピューティングシステム800の他のインスタンスと通信するために、利用可能なままである。図8で、GPGPU806A～806Dは、ホストインターフェーススイッチ804を介してプロセッサ802へ接続する一方で、プロセッサ802は、代替的に、P2P GPUリンク816の直接サポートを含み、GPGPU806～806Dへ直接接続してもよい。

30

40

【0142】

[機械学習ニューラルネットワーク実装]

本願で記載されているコンピューティングアーキテクチャは、機械学習のためにニューラルネットワークを訓練及びデプロイするのに特に適しているタイプの並列処理を実行す

50

るよう構成され得る。ニューラルネットワークは、グラフ関係を有する機能のネットワークとして一般化され得る。当該技術でよく知られているように、機械学習で使用される様々なタイプのニューラルネットワーク実装が存在する。ニューラルネットワークの1つの例示的なタイプは、上述されたように、フィードフォワードネットワークである。

【0143】

ニューラルネットワークの第2の例示的なタイプは、畳み込みニューラルネットワーク（CNN）である。CNNは、画像データなどの既知のグリッド様トポロジを有するデータを処理するための特殊化したフィードフォワードニューラルネットワークである。従って、CNNは、コンピュータビジョン及び画像認識アプリケーションのために広く使用されているが、それらは、発話及び言語処理などの他のタイプのパターン認識のためにも使用されてよい。CNN入力レイヤ内のノードは、「フィルタ」（網膜にある受容野によって呼び起こされた特徴検出部）の組に編成され、フィルタの各組の出力は、ネットワークの連続したレイヤ内のノードへ伝搬される。CNNの計算は、各フィルタに畳み込み数学演算を適用してそのフィルタの出力を生成することを含む。畳み込みは、2つの関数によって実行される特殊化した種類の数学演算であって、2つの元の関数のうち的一方の変更されたバージョンである第3の関数を生成する。畳み込みネットワークの用語では、畳み込みの最初の関数は入力と呼ばれ、一方、第2の関数は、畳み込みカーネルと呼ばれ得る。出力は、特徴マップと呼ばれ得る。例えば、畳み込みレイヤへの入力は、入力画像の様々な色成分を定義するデータの多次元配列であることができる。畳み込みカーネルは、パラメータの多次元配列であることができ、パラメータは、ニューラルネットワークのための訓練プロセスによって適応される。

【0144】

回帰型ニューラルネットワーク（RNN）は、レイヤ間のフィードバック接続を含むフィードフォワードニューラルネットワークの仲間である。RNNは、ニューラルネットワークの異なった部分にわたってパラメータデータを共有することによって、シーケンシャルデータのモデリングを可能にする。RNNのアーキテクチャはサイクルを含む。サイクルは、RNNからの出力データの少なくとも一部がその後の入力を順序通りに処理するためにフィードバックとして使用されるということ、変数の現在の値が将来の時点でのそれ自体の値に与える影響を表す。この特徴は、RNNを、言語データが構成され得る可変な性質により、言語処理に特に有用なものとする。

【0145】

後述される図は、例示的なフィードフォワード、CNN、及びRNNネットワークを提示するとともに、これらのタイプのネットワークの夫々を各々訓練及びデプロイするための一般的なプロセスについて記載する。これらの記載は例であって、本願で記載されている如何なる具体的に実施形態に関しても非限定的であり、表されている概念は、ディープニューラルネットワーク及び機械学習技術全般に概して適用され得る。

【0146】

上記の例示的なニューラルネットワークは、ディープラーニングを実行するために使用され得る。ディープラーニングは、ディープニューラルネットワークを使用する機械学習である。ディープラーニングで 사용되는ディープニューラルネットワークは、単一の隠れレイヤしか含まない浅いニューラルネットワークとは対照的に、複数の隠れレイヤから成る人工ニューラルネットワークである。より深いニューラルネットワークは、一般的に、訓練するのにより多くの計算負荷を要する。しかし、ネットワークの追加の隠れレイヤは、浅い機械学習技術に対して出力エラーが低減されることになるマルチステップパターン認識を可能にする。

【0147】

ディープラーニングで 사용되는ディープニューラルネットワークは、通常は、モデルに与えられた特徴表現に基づいて演算（例えば、オブジェクト分類、発話認識、など）を実行することができる数学モデルを表すバックエンドネットワークへ結合された特徴認識を実行するためのフロントエンドネットワークを含む。ディープラーニングは、Hand

crafted特徴エンジニアリングがモデルに対して実行されることを必要とせずに機械学習が実行されることを可能にする。代わりに、ディープニューラルネットワークは、入力データ内の統計的構造又は相関に基づいて特徴を学習することができる。学習された特徴は、検出された特徴を出力にマッピングすることができる数学モデルへ供給され得る。ネットワークによって使用される数学モデルは、一般的に、特定のタスクが実行されるために特殊化され、異なるモデルが、異なるタスクを実行するために使用されることになる。

【0148】

ニューラルネットワークが構造化されると、学習モデルが、特定のタスクを実行するようネットワークを訓練するためにネットワークに適用され得る。学習モデルは、ネットワークの出力エラーを減らすようモデル内の重みをどのように調整すべきかについて記載する。エラーの後方伝搬は、ニューラルネットワークを訓練するために使用される一般的なモデルである。入力ペルは、処理のためにネットワークに与えられる。ネットワークの出力は、損失関数を用いて所望の出力と比較され、エラー値が、出力レイヤ内のニューロンの夫々について計算される。エラー値は、次いで、各ニューロンが、元の出力へのその寄与を大まかにあらわる関連したエラー値を有するまで、後方に伝搬される。次いで、ネットワークは、ニューラルネットワークの重みを更新するために、確率的勾配降下アルゴリズム (stochastic gradient descent algorithm) などのアルゴリズムを用いてそれらのエラーから学習することができる。

【0149】

図9A～9Bは、例示的な畳み込みニューラルネットワークを表す。図9Aは、CNN内の様々なレイヤを表す。図9Aに示されているように、画像処理をモデル化するために使用される例示的なCNNは、入力画像の赤、緑、及び青 (RGB) 成分を記述する入力902を受け取ることができる。入力902は、複数の畳み込みレイヤ (例えば、畳み込みレイヤ904、畳み込みレイヤ906) によって処理され得る。複数の畳み込みレイヤからの出力は、全結合レイヤ908の組によって任意に処理されてもよい。全結合レイヤ内のニューロンは、フィードフォワードネットワークについて上述されたように、前のレイヤ内の全ての活性化 (activations) への完全接続を有する。全結合レイヤ908からの出力は、ネットワークからの出力結果を生成するために使用され得る。全結合レイヤ908内の活性化は、畳み込みの代わりに行列乗算を用いて計算され得る。全てのCNN実装が、全結合レイヤ908を利用するわけではない。例えば、いくつかの実施では、畳み込みレイヤ906がCNNの出力を生成することができる。

【0150】

畳み込みレイヤは疎結合され、これは、全結合レイヤ908で見られる従来のニューラルネットワーク構成とは異なる。従来のニューラルネットワークレイヤは完全接続され、それにより、あらゆる出力ユニットはあらゆる入力ユニットと相互作用する。しかし、畳み込みレイヤは、表されているように、フィールドの畳み込みの出力が後続のレイヤのノードへ (フィールド内のノードの夫々の各々の状態値の代わりに) 入力されるので、疎結合される。畳み込みレイヤに関連したカーネルが畳み込み演算を実行し、その出力が次のレイヤへ送られる。畳み込みレイヤで実行される次元削減 (dimensionality reduction) は、CNNが大きい画像を処理するようスケーリングすることを可能にする1つの態様である。

【0151】

図9Bは、CNNの畳み込みレイヤ内の例示的な計算段を表す。CNNの畳み込みレイヤへの入力912は、畳み込みレイヤ914の3つ段で処理され得る。3つの段は、畳み込み段916、検出器段918、及びプーリング段920を含むことができる。畳み込みレイヤ914は、連続する畳み込みレイヤへデータを出力することができる。ネットワークの最終の畳み込みレイヤは、出力される特徴マップデータを生成するか、あるいは、全結合レイヤへ入力を供給して、例えば、CNNへの入力に対して分類値を生成することができる。

【0152】

畳み込み段916は、線形活性化の組を生成するよういくつかの畳み込みを並行して実行する。畳み込み段916はアフィン変換を含むことができる。アフィン変換は、線形変換プラス平行移動 (a linear transformation plus a translation) として特定され得る任意の変換である。アフィン変換は、回転、平行移動、スケーリング、及びこれらの変換の組み合わせを含む。畳み込み段は、ニューロンに関連した局所領域と決定され得る入力内の特定の領域に接続されている関数 (例えば、ニューロン) の出力を計算する。ニューロンは、ニューロンの重みと、ニューロンが接続される局所入力内の領域との間の内積を計算する。畳み込み段916からの出力は、畳み込みレイヤ914の続く段によって処理される線形活性化の組を定義する。

10

【0153】

線形活性化は、検出器段918によって処理され得る。検出器段918では、非線形活性化は、非線形活性化関数によって処理される。非線形活性化関数は、畳み込みレイヤの各々のフィールドに影響を及ぼさずにネットワーク全体の非線形特性を高める。いくつかのタイプの非線形活性化関数が使用されてよい。1つの特定のタイプは、正規化線形関数 (rectified linear unit, ReLU) であり、これは、活性がゼロで閾値化されるように、 $f(x) = \max(0, x)$ と定義された活性化関数を使用する。

【0154】

プーリング段920は、畳み込みレイヤ906の出力を近くの出力の要約統計量で置き換えるプーリング関数を使用する。プーリング関数は、入力への小さい並進がプーリングされた出力を変更しないように、ニューラルネットワークに並進不変性を導入するために使用され得る。局所並進に対する不変性は、入力データでの特徴の存在が特徴の正確な位置よりも重要であるシナリオで有用であることができる。最大プーリング (max pooling)、平均プーリング (average pooling)、及び12ノルムプーリング (12-norm pooling) を含む様々なタイプのプーリング関数が、プーリング段920の間に使用され得る。更には、いくつかのCNN実装はプーリング段を含まない。代わりに、そのような実装は、前の畳み込みレイヤに対してストライドが増大している追加の畳み込み段を代用する。

20

【0155】

畳み込みレイヤ914からの出力は、次いで、次のレイヤ922によって処理され得る。次のレイヤ922は、追加の畳み込みレイヤ、又は全結合レイヤ908の1つであることができる。例えば、図9Aの第1畳み込みレイヤ904は、第2畳み込みレイヤ906へ出力することができ、一方、第2畳み込みレイヤは、全結合レイヤ908の第1レイヤへ出力することができる。

30

【0156】

図10は、例となる回帰型ニューラルネットワーク100を表す。回帰型ニューラルネットワーク (RNN) では、ネットワークの前の状態がネットワークの現在の状態の出力に影響を及ぼす。RNNは、様々な機能を用いて様々な方法で構築可能である。RNNの使用は、一般に、前の入力シーケンスに基づいて将来を予測するよう数学モデルを使用することを中心に展開する。例えば、RNNは、前のワードシーケンスを前提として、来たるべきワードを予測するよう統計的言語モデリングを実行するために使用されてよい。表されているRNN1000は、入力ベクトルを受け取る入力レイヤ1002と、回帰関数を実装する隠れレイヤ1004と、前の状態の‘メモリ’を可能にするフィードバックメカニズム1005と、結果を出力する出力レイヤ1006とを有しているものとして記載され得る。RNN1000は、時間ステップに基づいて動作する。所与の時間ステップでのRNNの状態は、フィードバックメカニズム1005を介して前の時間ステップに基づいて影響を及ぼされる。所与の時間ステップについては、隠れレイヤ1004の状態は、前の状態と、現在の時間ステップでの入力とによって定義される。第1時間ステップでの最初の入力 (x_1) が隠れレイヤ1004によって処理され得る。第2の入力 (x_2) は、最初の入力 (x_1) の処理中に決定される状態情報を用いて隠れレイヤ1004によって処理され得る。所与の状態は $s_t = f(Ux_t + Ws_{t-1})$ と計算され得る。このとき、

40

50

U及びWはパラメータ行列である。関数fは、一般的に、双曲線正接関数(hyperbolic tangent function)(Tanh)又は正規化線形関数 $f(x) = \max(0, x)$ などの非線形性である。しかし、隠れレイヤ1004で使用される具体的な数学関数は、RNN1000の具体的な実装詳細に応じて様々であることができる。

【0157】

記載されている基本的なCNN及びRNNネットワークに加えて、それらのネットワークに対する変形が可能であり得る。RNNの変形の一例は、長・短期記憶(long short term memory, LSTM)RNNである。LSTM RNNは、より長い言語シーケンスを処理するために必要であり得る長期依存を学習することが可能である。CNNの変形例は、CNNと類似した構造を有し、ディープ・ビリーフ・ネットワーク(deep belief network)と類似した方法で訓練される畳み込みディープ・ビリーフ・ネットワークである。ディープ・ビリーフ・ネットワーク(DBN)は、確率論的な(ランダムな)変数の複数のレイヤからなる生成ニューラルネットワークである。DBNは、教師なしグリーディ学習を用いてレイヤごとに訓練され得る。DBNの学習された重みが、次いで、ニューラルネットワークのための重みの最適な初期セットを決定することによって、予め訓練されたニューラルネットワークを提供するために使用され得る。

【0158】

図11は、ディープニューラルネットワークの訓練及びデプロイを表す。所与のネットワークがタスクに対して構造化されると、ニューラルネットワークは、訓練データセット1102を用いて訓練される。様々な訓練フレームワーク1104が、訓練プロセスのハードウェアアクセラレーションを可能にするよう開発されている。例えば、図6の機械学習フレームワーク604が訓練フレームワーク1104として構成されてよい。訓練フレームワーク1104は、訓練されていないニューラルネットワーク1106につながり、訓練されていないニューラルネットワーク1106が、訓練されたニューラルネットワーク1108を生成するよう、本願で記載されている並列処理リソースを用いて訓練されることを可能にすることができる。

【0159】

訓練プロセスを開始するために、初期重みが、ランダムに、又はディープ・ビリーフ・ネットワークを用いて予め訓練することによって、選択されてよい。次いで、訓練サイクルは、教師あり又は教師なしのいずれかの方法で実行される。

【0160】

教師あり学習は、例えば、訓練データセット1112が、入力に対する所望の出力と対にされた入力を含む場合、又は訓練データセットが、既知の出力を有する入力を含み、ニューラルネットワークの出力が手動で格付けされる場合など、仲介された動作(mediated operation)として訓練が実行される学習方法である。ネットワークは、入力を処理し、得られた出力を、期待されている又は望まれている出力の組と比較する。次いで、エラーがシステムを通して後方伝搬される。訓練フレームワーク1104は、訓練されていないニューラルネットワーク1106を制御する重みを調整することができる。訓練フレームワーク1104は、訓練されていないニューラルネットワーク1106が、既知の入力データに基づいて正しい答えを生成することに適したモデルに向かってどれくらい上手く収束しているかをモニタするツールを提供することができる。訓練プロセスは、ネットワークの重みが、ニューラルネットワークによって生成される出力を精緻化するように調整されるように、繰り返し行われる。訓練プロセスは、ニューラルネットワークが、訓練されたニューラルネットワーク1108に関連した統計的に望ましい制度に達するまで、続くことができる。訓練されたニューラルネットワーク1108は、次いで、新しいデータ1102の入力に基づいて推論結果1114を生成するために任意の数の機械学習動作を実装するようデプロイされ得る。

【0161】

教師なし学習は、ネットワークがラベルなしデータを用いて自身を訓練しようと試みる学習モデルである。よって、教師なし学習については、訓練データセット1112は、如

何なる関連付けられた出力データも伴わない入力を含むことになる。訓練されていないニューラルネットワーク 1106 は、ラベルなし入力内のグルーピングを学習することができ、個々の入力が全体のデータセットにどのように関係しているかを決定することができる。教師なし訓練は、データの次元を削減するのに有用な動作を実行することが可能な訓練されたニューラルネットワーク 1108 の一種である自己組織化マップ (self-organizing map) を生成するために使用され得る。教師なし訓練はまた、異常検出を実行するために使用され得る。これは、データの正常パターンから外れている入力データセット内のデータポイントの識別を可能にする。

【0162】

教師あり及び教師なし訓練の変形も用いられてよい。半教師あり学習は、訓練データセット 1112 が同じ分布のラベルあり及びラベルなしデータの混合を含む技術である。増分学習は、入力データがモデルを更に訓練するために連続して使用され得る教師あり学習の変形である。増分学習は、訓練されたニューラルネットワーク 1108 が、最初の訓練中にネットワーク内に植え付けられた知識を忘れずに新しいデータ 1112 に適応することを可能にする。

【0163】

教師ありか教師なしかに関わらず、特にディープニューラルネットワークのための訓練プロセスは、単一の計算ノードにとっては計算負荷が高すぎることもある。単一の計算ノードを使用する代わりに、計算ノードの分散ネットワークが、訓練プロセスを加速させるために使用され得る。

【0164】

図 12 は、分散学習 (distributed learning) を表すブロック図である。分散学習は、ニューラルネットワークの教師あり又は教師なし訓練を実行するために複数の分散した計算ノードを使用する訓練モデルである。分散した計算ノードは夫々、1 つ以上のホストプロセッサと、図 7 で見られたような高並列汎用グラフィックス処理ユニット 700 などの汎用処理ノードの 1 つ以上とを含むことができる。表されているように、分散学習は、モデル並列処理 (parallelism) 1202、データ並列処理 1204、又はモデル及びデータ並列処理の組み合わせ 1206 を実行することができる。

【0165】

モデル並列処理 1202 では、分散システム内の異なる計算ノードは、単一のネットワークの異なる部分の訓練計算を実行することができる。例えば、ニューラルネットワークの各レイヤは、分散システムの異なる処理ノードによって訓練され得る。モデル並列処理の利点には、特に大きいモデルにスケーリングする能力がある。ニューラルネットワークの異なるレイヤに関連した計算を分けることは、全てのレイヤの重みが単一の計算ノードのメモリに収まるわけでない非常に大きいニューラルネットワークの訓練を可能にする。いくつかの場合に、モデル並列処理は、大きいニューラルネットワークの教師なし訓練を実行するのに特に有用であることができる。

【0166】

データ並列処理 1204 では、分散ネットワークの異なるノードは、モデルの完全なインスタンスを有し、各ノードは、データの異なる部分を受け取る。異なるノードからの結果は、次いで、結合される。データ並列処理に対する異なるアプローチが可能である一方で、データ並列訓練アプローチは全て、結果を結合し、各ノード間でモデルパラメータを同期させる技術を必要とする。データの結合に対するアプローチの例には、パラメータ平均化及び更新に基づくデータ並列処理がある。パラメータ平均化は、訓練データのサブセットに関して各ノードを訓練し、大域パラメータ (例えば、重み、バイアス) を各ノードからのパラメータの平均にセットする。パラメータ平均化は、パラメータデータを保持する中央パラメータサーバを使用する。更新に基づくデータ並列処理は、ノードからパラメータサーバへパラメータを転送する代わりに、モデルに対する更新が転送されることを除いて、パラメータ平均化と類似している。更には、更新に基づくデータ並列処理は、非中央集権的な方法で実行可能であり、更新は圧縮され、ノード間で転送される。

10

20

30

40

50

【0167】

複合的なモデル及びデータ並列処理1206は、例えば、各計算ノードが複数のGPUを含む分散システムで、実装され得る。各ノードは、モデルの完全なインスタンスを有し、各ノード内の別々のGPUは、モデルの異なる部分を訓練するために使用される。

【0168】

分散学習は、単一のマシンでの訓練と比較して、オーバーヘッドが増大する。しかし、本願で記載されている並列プロセッサ及びGPUPUは夫々、高バンド幅GPU間データ転送及び加速されたりモードデータ同期化を可能にする技術を含め、分散訓練のオーバーヘッドを低減するよう様々な技術を実装することができる。

【0169】

〔例となる機械学習応用〕

機械学習は、コンピュータビジョン、自動運転及び航法、発話認識、並びに言語処理を含むがこれらに限られない様々な技術的課題を解決するために適用され得る。コンピュータビジョンは、従来から、機械学習応用の最も活発な研究分野の1つである。コンピュータビジョンの応用は、顔認識などのヒトの視覚能力を再現することから、視覚能力の新たなカテゴリを作り出すことまで多岐にわたる。例えば、コンピュータビジョンアプリケーションは、映像内で見ることができる対象に引き起こされた振動から音波を認識するよう構成され得る。並列プロセッサにより加速された機械学習は、コンピュータビジョンアプリケーションが、以前に実現可能であったよりも相当に大きい訓練データセットを用いて訓練されることを可能にし、かつ、推論システムが、より低電力の並列プロセッサを用いてデプロイされることを可能にする。

【0170】

並列プロセッサにより加速された機械学習は、レーン及び道路標識認識、障害物回避、ナビゲーション、並びに運転制御を含む自動運転用途がある。加速された機械学習技術は、特定の訓練入力に対する適切な応答を定義するデータセットに基づいて運転モデルを訓練するために使用され得る。本願で記載されている並列プロセッサは、自動運転ソリューションのために使用されるますます複雑なニューラルネットワークの急速訓練を可能にすることができ、自律走行車への組み込みに適したモバイルプラットフォームにおける低電力推論プロセッサのデプロイを可能にする。

【0171】

並列プロセッサにより加速されたディープニューラルネットワークは、自動発話認識 (automatic speech recognition, ASR) に対する機械学習アプローチを可能にしていた。ASRは、入力された音響シーケンスを前提として最も確からしい言語シーケンスを計算する関数の生成を含む。ディープニューラルネットワークを使用する加速された機械学習は、ASRのために従前使用されていた隠れマルコフモデル (hidden Markov model, HMM) 及びガウス混合モデル (Gaussian mixture models, GMM) の置換を可能にしている。

【0172】

並列プロセッサにより加速された機械学習はまた、自然言語処理を加速させるためにも使用され得る。自動学習プロシージャは、誤りのある又は聞き慣れない入力に対してロバストであるモデルを生成するために統計的推論アルゴリズムを利用することができる。例となる自然言語プロセッサアプリケーションには、ヒトの言語間の自動機械翻訳がある。

【0173】

機械学習のために使用される並列処理プラットフォームは、訓練プラットフォーム及びデプロイプラットフォームに分けられ得る。訓練プラットフォームは、一般的に高並列であり、マルチGPU単一ノード訓練及びマルチノードマルチGPU訓練を加速させる最適化を含む。訓練に適した並列プロセッサの例には、図7の汎用グラフィクス処理ユニット700及び図8のマルチGPUコンピューティングシステム800がある。対照的に、デプロイされた機械学習プラットフォームは、一般的に、カメラ、自律ロボット、及び自律走行車などの製品での使用に適した、より低電力の並列プロセッサを含む。

【0174】

図13は、訓練されたモデルを用いて推論を行うのに適した例示的な推論システム・オン・ア・チップ（SOC）1300を表す。SOC1300は、メディアプロセッサ1302、ビジョンプロセッサ1304、GPGPU1306、及びマルチコアプロセッサ1308を含む処理コンポーネントを組み込むことができる。GPGPU1306は、GPGPU700などの、本願で記載されているGPGPUであってよく、マルチコアプロセッサ1308は、マルチコアプロセッサ405～406などの、本願で記載されているマルチコアプロセッサであってよい。SOC1300は、処理コンポーネントの夫々によってアクセス可能である共有されたオンチップデータプールを可能にすることができるオンチップメモリ1305を更に含むことができる。処理コンポーネントは、自律走行車及び自律ロボットを含む様々な機械学習プラットフォームへのデプロイを可能にするよう低電力動作のために最適化され得る。例えば、SOC1300の1つの実施は、自律走行車のための主制御システムの部分として使用され得る。SOC1300が自律走行車で使用のために構成される場合に、SOCは、デプロイ権限の関連した機能安全標準に従うよう設計及び構成される。

10

【0175】

動作中、メディアプロセッサ1302及びビジョンプロセッサ1304は、コンピュータビジョン動作を加速させるよう協働することができる。メディアプロセッサ1302は、複数の高解像度（例えば、4K，8K）ビデオストリームの低遅延復号化を可能にすることができる。復号されたビデオストリームは、オンチップメモリ1305内のバッファに書き込まれ得る。ビジョンプロセッサ1304は、次いで、訓練された画像認識モデルを用いてフレームを処理することに備えて、復号されたビデオをパースし、復号されたビデオのフレームに対して予備処理動作を実行することができる。例えば、ビジョンプロセッサ1304は、高解像度ビデオデータに対して画像認識を実行するために使用されるCNNの畳み込み演算を加速させることができ、一方、バックエンドモデル計算は、GPGPU1306によって実行される。

20

【0176】

マルチコアプロセッサ1308は、メディアプロセッサ1302及びビジョンプロセッサ1304によって実行されるデータ転送及び共有メモリ動作の順序付け及び同期化を支援する制御ロジックを含むことができる。マルチコアプロセッサ1308はまた、GPGPU1306の推論計算能力を利用することができるソフトウェアアプリケーションを実行するようアプリケーションプロセッサとして動作することもできる。例えば、ナビゲーション及び駆動ロジックの少なくとも一部は、マルチコアプロセッサ1308で実行されるソフトウェアで実装され得る。そのようなソフトウェアは、GPGPU1306に対して直接に計算ワークロードを発行することができ、あるいは、計算ワークロードは、これらの動作の少なくとも一部をGPGPU1306にオフロードすることができるマルチコアプロセッサ1308に対して発行され得る。

30

【0177】

GPGPU1306は、汎用グラフィクス処理ユニット700内の処理クラスタ706A～706Hの低電力構成などの計算クラスタを含むことができる。GPGPU1306内の計算クラスタは、訓練されたニューラルネットワークに対して推論計算を実行するよう特に最適化される命令をサポートすることができる。例えば、GPGPU1306は、8ビット及び4ビット整数ベクトル演算などの低精度計算を実行するための命令をサポートすることができる。

40

【0178】

[更なるシステム概要]

図14は、処理システム1400のブロック図である。本願のいずれかの他の図の要素と同じ又は類似した名称を有している図14の要素は、他の図で見られるのと同じ要素について記載しており、本願のどこかで記載されているものと同じように動作又は機能することができ、同じコンポーネントを有することができ、そして、他のエンティティへリン

50

クされ得るが、そのように限定されない。システム1400は、シングルプロセッサデスクトップシステム、マルチプロセッサワークステーションシステム、又は多数のプロセッサ1402又はプロセッサコア1407を備えたサーバシステムで使用されてよい。システム1400は、ローカル又はワイドエリアネットワークへの有線又は無線接続を備えたインターネット・オブ・シングス（IoT）デバイス内など携帯型、手持ち式、埋め込み型デバイスで使用されるシステム・オン・ア・チップ（SoC）集積回路内に組み込まれた処理プラットフォームであってよい。

【0179】

システム1400は、図1のコンポーネントと対応するコンポーネントを備えた処理システムであってよい。例えば、異なる構成では、プロセッサ1402又はプロセッサコア1407は、図1のプロセッサ102と対応してよい。グラフィクスプロセッサ1408は、図1の並列プロセッサ112と対応してよい。外部グラフィクスプロセッサ1418は、図1のアドインデバイス120のうちの1つであってよい。システム1400は、サーバベースのゲームプラットフォーム；ゲーム及びメディアコンソールを含むゲームコンソール；携帯型ゲームコンソール、手持ち式ゲーム機、又はオンラインゲーム機を含むか、それと結合するか、又はその中に組み込まれることが可能である。システム1400は、携帯電話機、スマートフォン、タブレットコンピュータデバイス、又はモバイルインターネット接続デバイス、例えば、低内部記憶容量のラップトップ、の部分であってよい。処理システム1400はまた、スマートウォッチウェアブルデバイスなどのウェアラブルデバイス；現実世界の視覚、音声又は触覚経験を補完するよう視覚、音声又は触覚出力を供給するか、あるいは、テキスト、音声、映像、ホログラフィック画像若しくは映像、又は触覚フィードバックを供給する拡張現実（AR）又は仮想現実（VR）機能により強化されたスマートアイウェア又は衣服；あるいは、他の仮想現実（VR）デバイスを含むか、それと結合するか、又はその中に組み込まれることも可能である。処理システム1400は、テレビ受像機又はセットトップボックスデバイスを含むか、あるいは、その部分であってよい。

【0180】

システム1400は、バス、トレーラトラック、自動車、原動機付き又は電動自転車、飛行機又はグライダー（あるいは、それらの任意の組み合わせ）などの自動運転式輸送手段を含むか、それと結合するか、又はその中に組み込まれることが可能である。自動運転式輸送手段は、輸送手段の周りで検知された環境を処理するためにシステム1400を用いてよい。

【0181】

1つ以上のプロセッサ1402は夫々、実行時に、システム又はユーザソフトウェアのための動作を実行する命令を処理する1つ以上のプロセッサコア1407を含んでよい。1つ以上のプロセッサコア1407のうちの少なくとも1つは、特定の命令セット1409を処理するよう構成されてよい。命令セット1409は、複数命令セットコンピューティング（Complex Instruction Set Computing, CISC）、縮小命令セットコンピューティング（Reduced Instruction Set Computing, RISC）、又は超長命令語（Very Long Instruction Word, VLW）によるコンピューティングを促進してよい。1つ以上のプロセッサコア1407は、異なる命令セット1409を処理してもよい。異なる命令セット1409は、他の命令セットのエミュレーションを促進する命令を含んでよい。プロセッサコア1407はまた、デジタル信号プロセッサ（Digital Signal Processor, DSP）などの他の処理デバイスを含んでもよい。

【0182】

プロセッサ1402は、キャッシュメモリ1404を含んでよい。アーキテクチャに応じて、プロセッサ1402は、単一内部キャッシュ、又は複数レベルの内部キャッシュを有することができる。いくつかの実施形態において、キャッシュメモリは、プロセッサ1402の様々なコンポーネントの間で共有される。いくつかの実施形態において、プロセッサ1402はまた、外部キャッシュ（例えば、レベル-3（L3）キャッシュ又はラス

10

20

30

40

50

トレベルキャッシュ（LLC）（図示せず）も使用する。外部キャッシュは、既知のキャッシュコヒーレンス技術を用いてプロセッサコア1407の間で共有されてよい。プロセッサ1402には、レジスタファイル1406が更に含まれ得る。レジスタファイル1406は、異なるタイプのデータを記憶する異なるタイプのレジスタ（例えば、整数レジスタ、浮動小数点レジスタ、ステータスレジスタ、及び命令ポインタレジスタ）を含んでよい。いくつかのレジスタは、汎用のレジスタであってよく、一方、他のレジスタは、プロセッサ1402の設計に特有であってよい。

【0183】

1つ以上のプロセッサ1402は、アドレス、データ、又は制御信号などの通信信号をプロセッサ1402とシステム1400内の他のコンポーネントとの間で伝送するよう1つ以上のインターフェースバス1410と結合されてよい。インターフェースバス1410は、これらの実施形態のうちの1つで、ダイレクト・メディア・インターフェース（Direct Media Interface, DMI）バスの変形などのプロセッサバスであることができる。しかし、プロセッサバスは、DMIバスに限定されず、1つ以上のペリフェラル・コンポーネント・インターコネクト（例えば、PCI、PCI Express）バス、メモリバス、又は他のタイプのインターフェースバスを含んでもよい。例えば、プロセッサ1402は、集積メモリコントローラ1416及びプラットフォームコントローラハブ1430を含んでよい。メモリコントローラ1416は、メモリデバイスとシステム1400の他のコンポーネントとの間の通信を助け、一方、プラットフォームコントローラハブ（PCH）1430は、ローカルI/Oバスを介したI/Oデバイスへの接続を提供する。

【0184】

メモリデバイス1420は、動的ランダム・アクセス・メモリ（DRAM）デバイス、静的ランダム・アクセス・メモリ（SRAM）デバイス、フラッシュメモリデバイス、相変化メモリデバイス、又はプロセスメモリとして機能するよう適切な性能を備えたその他メモリデバイスであることができる。メモリデバイス1420は、例えば、1つ以上のプロセッサ1402がアプリケーション又はプロセスを実行するときに使用されるデータ1422及び命令1421を記憶するために、システム1400のためのシステムメモリとして動作することができる。メモリコントローラ1416はまた、任意の外部グラフィクスプロセッサ1418とも結合する。外部グラフィクスプロセッサ1418は、グラフィクス及びメディア操作を実行するようプロセッサ1402内の1つ以上のグラフィクスプロセッサ1408と通信してよい。いくつかの実施形態において、グラフィクス、メディア、及び／又は計算操作は、グラフィクス、メディア、又は計算操作の特殊化した組を実行するよう構成され得るコプロセッサであるアクセラレータ1412によって支援されてよい。例えば、アクセラレータ1412は、機械学習又は計算操作を最適化するために使用される行列乗算アクセラレータであってよい。アクセラレータ1412は、グラフィクスプロセッサ1408と協力してレイトレーシング動作を実行するために使用され得るレイトレーシングアクセラレータであることができる。一実施形態において、外部アクセラレータ1419は、アクセラレータ1412の代わりに又はそれと協調して使用されてよい。

【0185】

表示デバイス1411が設けられてよく、これはプロセッサ1402へ接続することができる。表示デバイス1411は、モバイル電子機器若しくはラップトップデバイスで見られるような内蔵表示デバイス、又は表示インターフェース（例えば、Display Portなど）を介して取り付けられた外部表示デバイス、の1つ以上であることができる。表示デバイス1411は、仮想現実（VR）アプリケーション又は拡張現実（AR）アプリケーションで使用される立体視表示デバイスなどのヘッドマウント型ディスプレイ（HMD）であることができる。

【0186】

プラットフォームコントローラハブ1430は、周辺機器が高速I/Oバスを介してメモリデバイス1420及びプロセッサ1402へ接続することを可能にし得る。I/Oペ

10

20

30

40

50

リフェラルは、オーディオコントローラ 1 4 4 6、ネットワークデバイス 1 4 3 4、ファームウェアインターフェース 1 4 2 8、無線トランシーバ 1 4 2 6、タッチセンサ 1 4 2 5、データ記憶デバイス 1 4 2 4（例えば、不揮発性メモリ、揮発性メモリ、ハードディスクドライブ、フラッシュメモリ、NAND、3D NAND、3D XPoint／メモリユニット、など）を含むが、これらに限られない。データ記憶デバイス 1 4 2 4 は、ペリフェラル・コンポーネント・インターコネクト（例えば、PCI、PCIエクスプレス）バスなどのペリフェラルバスを介して又はストレージインターフェース（例えば、SATA）を介して接続することができる。タッチセンサ 1 4 2 5 は、タッチスクリーンセンサ、圧力センサ、又は指紋センサを含むことができる。無線トランシーバ 1 4 2 6 は、Wi-Fi トランシーバ、Bluetooth（登録商標）トランシーバ、又はモバイルネットワークトランシーバ、例えば、3G、4G、5G、又はロング・ターム・エボリューション（LTE）トランシーバであることができる。ファームウェアインターフェース 1 4 2 8 は、システムファームウェアとの通信を可能にし、例えば、ユニファイド・エクステンシブル・ファームウェア・インターフェース（unified extensible firmware interface, UEFI）であることができる。ネットワークコントローラ 1 4 3 4 は、有線ネットワークへのネットワーク接続を可能にすることができる。いくつかの実施形態において、高性能ネットワークコントローラ（図示せず）がインターフェースバス 1 4 1 0 と結合する。オーディオコントローラ 1 4 4 6 は、マルチチャネル・ハイディフィニション・オーディオコントローラであってよい。これらの実施形態のうちのいくつかにおいて、システム 1 4 0 0 は、レガシー（例えば、Personal System 2（PS／2））デバイスをシステムへ結合する任意のレガシー I／O コントローラ 1 4 4 0 を含む。プラットフォームコントローラハブ 1 4 3 0 はまた、キーボード及びマウス 1 4 4 3 の組み合わせ、カメラ 1 4 4 4、又は他の USB 入力デバイスなどの入力デバイスを接続する 1 つ以上のユニバーサル・シリアル・バス（USB）コントローラ 1 4 4 2 へ接続することもできる。

【0187】

明らかなように、示されているシステム 1 4 0 0 は、例であって限定ではなく、別なふうに構成される他のタイプのデータ処理システムも使用されてよい。例えば、メモリコントローラ 1 4 1 6 及びプラットフォームコントローラハブ 1 4 3 0 のインスタンスは、外部グラフィクスプロセッサ 1 4 1 8 などの分離した外部グラフィクスプロセッサに組み込まれてもよい。プラットフォームコントローラハブ 1 4 3 0 及び／又はメモリコントローラ 1 4 1 6 は、1 つ以上のプロセッサ 1 4 0 2 の外であってよい。例えば、システム 1 4 0 0 は、外付けのメモリコントローラ 1 4 1 6 及びプラットフォームコントローラハブ 1 4 3 0 を含むことができ、これらは、プロセッサ 1 4 0 2 と通信するシステムチップセット内のメモリコントローラハブ及びペリフェラルコントローラハブとして構成されてよい。

【0188】

例えば、回路基板（「スレッド（sleds）」）が使用可能であり、その上には、CPU、メモリ、及び他のコンポーネントなどのコンポーネントが載置され、熱性能を向上させるよう設計される。プロセッサなどの処理コンポーネントは、スレッドの上面に置かれてよく、一方、DIMMなどの近くのメモリは、スレッドの底面に置かれる。この設計によってもたらされる強化された気流の結果として、コンポーネントは、通常のシステムよりも高い周波数及び電力レベルで動作して、性能を向上させ得る。更に、スレッドは、ラック内の電源及びデータ通信ケーブルと盲目的に結合するよう構成され、それによって、迅速な取り外し、アップグレード、再インストール、及び／又は交換が可能である。同様に、プロセッサ、アクセラレータ、メモリ、及びデータ記憶ドライブなどの、スレッドに載置されている個々のコンポーネントは、相互の間隔を広げられることで容易にアップグレードされるよう構成される。実例となる実施形態では、コンポーネントは、それらの真正性を証明するためのハードウェア認証機能を更に含む。

【0189】

データセンターは、Ethernet（登録商標）及び Omni-Path を含む複数の他のネットワークアーキテクチャをサポートする単一のネットワークアーキテクチャ（

10

20

30

40

50

「ファブリック」）を利用することができる。スレッドは、光ファイバを介してスイッチへ結合可能であり、これは、典型的なツイステッド・ペア・ケーブル（カテゴリ5、カテゴリ5e、カテゴリ6、など）よりも高いバンド幅及び低いレイテンシをもたらす。高バンド幅、低レイテンシの相互接続及びネットワークアーキテクチャにより、データセンターは、使用中に、物理的に構成要素に分けられているメモリ、アクセラレータ（例えば、GPU、グラフィクスアクセラレータ、FPGA、ASIC、ニューラルネットワーク及び／又は人工知能アクセラレータ、など）及びデータ記憶ドライブなどのリソースをプールし、それらを、必要に応じて計算リソース（例えば、プロセッサ）に提供して、計算リソースが、プールされたリソースに、それらがあたかもローカルであるかのようにアクセスすることを可能にする。

10

【0190】

電力供給又は電源は、電圧及び／又は電流をシステム1400又は本願で記載されているいずれかのコンポーネント若しくはシステムに供給することができる。一例では、電力供給は、壁コンセントにプラグを差し込むようAC-DC（交流から直流）アダプタを含む。そのようなAC電力は、再生可能エネルギー（例えば、太陽エネルギー）電源であることができる。一例では、電源は、外部AC-DCコンバータなどのDC電源を含む。電源又は電力供給はまた、充電野への近接により充電する無線充電ハードウェアを含む。電源は、内蔵バッテリー、交流電源、運動に基づいた電源、太陽電源、又は燃料電池を含むことができる。

【0191】

図15A～Cは、コンピューティングシステム及びグラフィクスプロセッサを表す。本願のいずれかの他の図の要素と同じ又は類似した名称を有している図15A～15Cの要素は、他の図で見られるのと同じ要素について記載しており、本願のどこかで記載されているものと同じように動作又は機能することができ、同じコンポーネントを有することができ、そして、他のエンティティへリンクされ得るが、そのように限定されない。図15Aは、プロセッサ1402のうちの1つの変形であってよく、それらの1つの代わりに使用されてよいプロセッサ1500のブロック図である。従って、本願でのプロセッサ1500と組み合わせられる如何なる特徴も、プロセッサ1402との対応する組み合わせを開示するが、そのように限定されない。プロセッサ1500は、1つ以上のプロセッサコア1502A～1502N、集積メモリコントローラ1514、及び集積グラフィクスプロセッサ1508を備えてよい。集積グラフィクスプロセッサ1508が望まれる場合に、プロセッサを含むシステムは、システムチップセット内にあるか又はシステムバスを介して結合されるグラフィクスプロセッサデバイスを含むことになる。プロセッサ1500は、破線ボックスで表されている追加コア1502Nまでの追加コアを含むことができる。プロセッサコア1502A～1502Nの夫々は、1つ以上の内部キャッシュユニット1504A～1504Nを含む。いくつかの実施形態において、各プロセッサコア1502A～1502Nはまた、1つ以上の共有キャッシュユニット1506へのアクセスも有する。

20

30

【0192】

内部キャッシュユニット1504A～1504N及び共有キャッシュユニット1506は、プロセッサ1500内のキャッシュメモリヒエラルキを表す。キャッシュメモリヒエラルキは、各プロセッサコア内の少なくとも1つのレベルの命令及びデータキャッシュと、レベル2（L2）、レベル3（L3）、レベル4（L4）、又は他のレベルのキャッシュなどの1つ以上のレベルの共有ミッドレベルキャッシュとを含んでよく、外部メモリの前の最も高いレベルのキャッシュは、LLCと分類される。いくつかの実施形態において、キャッシュコヒーレンシロジックは、様々なキャッシュユニット1506及び1504A～1504Nの間でコヒーレンシを保つ。

40

【0193】

プロセッサ1500はまた、1つ以上のバスコントローラユニット1516の組と、システムエージェントコア1510とを含んでもよい。1つ以上のバスコントローラユニッ

50

ト 1 5 1 6 は、1 つ以上の P C I 又は P C I エクスプレスバスなどのペリフェラルバスの組を管理する。システムエージェントコア 1 5 1 0 は、様々なプロセッサコンポーネントのための管理機能性を提供する。システムエージェントコア 1 5 1 0 は、様々な外部メモリデバイス（図示せず）へのアクセスを管理する 1 つ以上の集積メモリコントローラ 1 5 1 4 を含んでよい。

【 0 1 9 4 】

プロセッサコア 1 5 0 2 A ～ 1 5 0 2 N のうちの 1 つ以上は、同時のマルチスレッディングのサポートを含んでよい。システムエージェントコア 1 5 1 0 は、マルチスレッド処理中にコア 1 5 0 2 A ～ 1 5 0 2 N を協調及び作動させるためのコンポーネントを含む。システムエージェントコア 1 5 1 0 は、プロセッサコア 1 5 0 2 A ～ 1 5 0 2 N 及びグラフィクスプロセッサ 1 5 0 8 の電力状態を調整するロジック及びコンポーネントを含む電力制御ユニット（P C U）を更に含んでもよい。

【 0 1 9 5 】

プロセッサ 1 5 0 0 は、グラフィクス処理動作を実行するグラフィクスプロセッサ 1 5 0 8 を更に含んでもよい。これらの実施形態のうちのいくつかで、グラフィクスプロセッサ 1 5 0 8 は、共有キャッシュユニット 1 5 0 6 の組、及び 1 つ以上の集積メモリコントローラ 1 5 1 4 を含むシステムエージェントコア 1 5 1 0 と結合する。システムエージェントコア 1 5 1 0 はまた、結合された 1 つ以上のディスプレイへのグラフィクスプロセッサ出力を駆動するディスプレイコントローラ 1 5 1 1 も含んでもよい。ディスプレイコントローラ 1 5 1 1 はまた、少なくとも 1 つのインターコネクトを介してグラフィクスプロセッサと結合された別個のモジュールであってもよく、あるいは、グラフィクスプロセッサ 1 5 0 8 内に組み込まれてもよい。

【 0 1 9 6 】

リングベースのインターコネクトユニット 1 5 1 2 が、プロセッサ 1 5 0 0 の内部コンポーネントを結合するために使用されてもよい。しかし、代替のインターコネクトユニットが使用されてもよく、例えば、ポイント・ツー・ポイントインターコネクト、スイッチドインターコネクト、又は当該技術でよく知られている技術を含む他の技術がある。これらの実施形態のうちのいくつかで、リングベースのインターコネクトユニット 1 5 1 2 によれば、グラフィクスプロセッサ 1 5 0 8 は、I / O リンク 1 5 1 3 を介してリングベースのインターコネクト 1 5 1 2 と結合する。

【 0 1 9 7 】

例となる I / O リンク 1 5 1 3 は、e D R A M モジュールなどの高性能埋め込みメモリモジュール 1 5 1 8 と様々なプロセッサコンポーネントとの間の通信を助けるパッケージ I / O インターコネクトを含む多種多様な I / O インターコネクトのうちの少なくとも 1 つを表す。任意に、プロセッサコア 1 5 0 2 A ～ 1 5 0 2 N 及びグラフィクスプロセッサ 1 5 0 8 の夫々は、埋め込みメモリモジュール 1 5 1 8 を共有ラストレベルキャッシュとして使用することができる。

【 0 1 9 8 】

プロセッサコア 1 5 0 2 A ～ 1 5 0 2 N は、例えば、同じ命令セットアーキテクチャを実行する同種のコアであってよい。代替的に、プロセッサコア 1 5 0 2 A ～ 1 5 0 2 N は、命令セットアーキテクチャ（I S A）に関して異種であり、プロセッサコア 1 5 0 2 A ～ 1 5 0 2 N のうちの 1 つ以上は、第 1 命令セットを実行し、一方、残りのコアのうちの少なくとも 1 つは、第 1 命令セットのサブセット又は異なる命令セットを実行する。プロセッサコア 1 5 0 2 A ～ 1 5 0 2 N は、マイクロアーキテクチャに関して異種であってよく、相対的により高い電力消費を有する 1 つ以上のコアは、より低い電力消費を有する 1 つ以上の電力コアと結合する。他の例として、プロセッサコア 1 5 0 2 A ～ 1 5 0 2 N は、計算能力に関して異種である。更には、プロセッサ 1 5 0 0 は、1 つ以上のチップ上で、あるいは、他のコンポーネントに加えて、表されているコンポーネントを備えた S o C 集積回路として、実装され得る。

【 0 1 9 9 】

図15Bは、本願で記載されているいくつかの実施形態に従うグラフィクスプロセッサコア1519のハードウェアロジックのブロック図である。グラフィクスプロセッサコア1519は、コアスライスと時々呼ばれ、モジュールグラフィクスプロセッサ内の1つ又は複数のグラフィクスコアであることができる。グラフィクスプロセッサコア1519は、1つのグラフィクスコアスライスの例であり、本願で記載されているグラフィクスプロセッサは、目標電力及び性能エンベロップに基づいて複数のグラフィクスコアスライスを含んでよい。各グラフィクスプロセッサコア1519は、汎用及び固定機能ロジックのモジュールブロックを含む、サブスライスとも呼ばれる複数のサブコア1521A～1521Fと結合されている固定機能ブロック1530を含むことができる。

【0200】

固定機能ブロック1530は、例えば、より低い性能及び／又はより低い電力のグラフィクスプロセッサ実装で、グラフィクスプロセッサコア1519内の全てのサブコアによって共有され得るジオメトリ／固定機能パイプライン1531を含んでよい。ジオメトリ／固定機能パイプライン1531は、3D固定機能パイプライン（例えば、後述される図16Aで見られる3Dパイプライン1612）、ビデオフロントエンドユニット、スレッドスポナー（spawner）及びスレッドディスパッチャ、並びに統合リターンバッファ（例えば、後述される図17の統合リターンバッファ1718）を管理する統合リターンバッファマネージャを含んでよい。

【0201】

固定機能ブロック1530はまた、グラフィクスSoCインターフェース1532、グラフィクスマイクロコントローラ1533、及びメディアパイプライン1534を含んでよい。グラフィクスSoCインターフェース1532は、グラフィクスプロセッサコア1519とSoC集積回路内の他のプロセッサとの間のインターフェースを提供する。グラフィクスマイクロコントローラ1533は、スレッドディスパッチ、スケジューリング、及びプリエンブションを含むグラフィクスプロセッサコア1519の様々な機能を管理するよう構成可能であるプログラム可能なサブプロセッサである。メディアパイプライン1534（例えば、図16A及び図17のメディアパイプライン1616）は、画像及びビデオデータを含むマルチメディアデータの復号化、符号化、前処理、及び／又は後処理を助けるロジックを含む。メディアパイプライン1534は、サブコア1521A～1521F内の計算又はサンプリングロジックへのリクエストによりメディア操作を実装する。

【0202】

SoCインターフェース1532は、汎用アプリケーションプロセッサコア（例えば、CPU）、かつ／あるいは、共有ラストレベルキャッシュメモリなどのメモリヒエラルキ要素、システムRAM、及び／又は埋め込みオンチップ若しくはオンパッケージDRAMを含むSoC内の他のコンポーネントと通信することをグラフィクスプロセッサコア1519に可能にする。SoCインターフェース1532はまた、カメライメージングパイプラインなどのSoC内の固定機能デバイスとの通信も可能にすることができ、グラフィクスプロセッサコア1519とSoC内のCPUとの間で共有され得るグローバルメモリアトミックの使用を可能にしかつ／あるいはそれを実装する。SoCインターフェース1532はまた、グラフィクスプロセッサコア1519のための電力管理制御を実装し、グラフィクスプロセッサコア1519のクロックドメインとSoC内の他のクロックドメインとの間のインターフェースを可能にすることもできる。任意に、SoCインターフェース1532は、グラフィクスプロセッサ内の1つ以上のグラフィクスコアの夫々へコマンド及び命令を供給するよう構成されるコマンドストリーマ及びグローバルスレッドディスパッチャからのコマンドバッファの受信を可能にする。コマンド及び命令は、メディア操作が実行されるべき場合にメディアパイプライン1534に、あるいは、グラフィクス処理動作が実行されるべきである場合にジオメトリ及び固定機能パイプライン（例えば、ジオメトリ及び固定機能パイプライン1531、ジオメトリ及び固定機能パイプライン1537）にディスパッチされ得る。

【0203】

グラフィクスマイクロコントローラ 1533 は、グラフィクスプロセッサコア 1519 のための様々なスケジューリング及び管理タスクを実行するよう構成され得る。1つの構成では、グラフィクスマイクロコントローラ 1533 は、例えば、サブコア 1521A ~ 1521F 内の実行ユニット (execution unit, EU) アレイ 1522A ~ 1522F、1524A ~ 1524F 内の様々なグラフィクス並列エンジンに対してグラフィクス及び／又は計算ワークロードスケーリングを実行することができる。このワークロードスケジューリングでは、グラフィクスプロセッサコア 1519 を含む SoC の CPU コアで実行されるホストソフトウェアは、適切なグラフィクスエンジンでスケーリング動作を呼び出す複数のグラフィックプロセッサコアのうちの 1 つにワークロードを発行することができる。スケジューリング動作は、次にどのワークロードを実行すべきかを決定すること、コマンドストリーマにワークロードを発行すること、エンジンで実行されている既存のワークロードをプリエンプトとすること、ワークロードの進捗を管理すること、及びワークロードが完了する場合にホストソフトウェアに通知することを含む。任意に、グラフィクスマイクロコントローラ 1533 はまた、グラフィクスプロセッサコア 1519 の低電力又はアイドル状態を促進することができ、グラフィクスプロセッサコア 1519 に、システム上のオペレーティングシステム及び／又はグラフィクスドライバソフトウェアから独立して低電力状態遷移にわたってグラフィクスプロセッサコア 1519 内のレジスタにセーブ及びリストアする能力をもたらす。

【0204】

グラフィクスプロセッサコア 1519 は、表されているサブコア 1521A ~ 1521F よりも多い又は少ない、最大 N 個までモジュールサブコアを有してよい。N 個のサブコアの各組について、グラフィクスプロセッサコア 1519 はまた、様々なグラフィクス及び計算処理動作を加速させるよう、共有機能ロジック 1535、共有及び／又はキャッシュメモリ 1536、ジオメトリ／固定機能パイプライン 1537、並びに追加の固定機能ロジック 1538 を含むこともできる。共有機能ロジック 1535 は、グラフィクスプロセッサコア 1519 内の各 N 個のサブコアによって共有され得る図 17 の共有機能ロジック 1720 に関連したロジックユニット (例えば、サンブラ、マス、及び／又はインターレッド通信ロジック) を含むことができる。共有及び／又はキャッシュメモリ 1536 は、グラフィクスプロセッサコア 1519 内の N 個のサブコア 1521A ~ 1521F の組のためのラストレベルキャッシュであることができ、複数のサブコアによってアクセス可能である共有メモリとしても働くことができる。ジオメトリ／固定機能パイプライン 1537 は、固定機能ブロック 1530 内のジオメトリ／固定機能パイプライン 1531 の代わりに含まれ得、同じ又は類似したロジックユニットを含むことができる。

【0205】

グラフィクスプロセッサコア 1519 は、グラフィクスプロセッサコア 1519 によって使用される様々な固定機能アクセラレーションロジックを含むことができる追加の固定機能ロジック 1538 を含んでよい。任意に、追加の固定機能ロジック 1538 は、ポジション・オンリー・シェーディング (position only shading) で使用される追加のジオメトリパイプラインを含む。ポジション・オンリー・シェーディングでは、2つのジオメトリパイプラインが存在し、ジオメトリ／固定機能パイプライン 1538、1531 内のフルジオメトリパイプラインと、追加の固定機能ロジック 1538 内に含まれ得る追加のジオメトリパイプラインであるカルパイプラインとである。例えば、カルパイプラインは、フルジオメトリパイプラインの縮小版であってよい。フルパイプライン及びカルパイプラインは、同じアプリケーションの異なるインスタンスを実行することができ、各インスタンスは、別個のコンテキストを有する。ポジション・オンリー・シェーディングは、破棄された三角形 (discarded triangles) の長いカル実行 (long cull run) を隠すことができ、いくつかのインスタンスにおいては、シェーディングがより早く完了されるのを可能にしている。例えば、追加的な固定機能ロジック 1538 内のカルパイプラインロジックは、主アプリケーションと並列にポジションシェーダを実行することができ、そして、一般的に、フルパイプラインよりも速く、決定的な結果 (critical result) を生成

10

20

30

40

50

する。ピクセルのフレームバッファへのラスタライゼーション及びレンダリングを実行することなく、カルパイプラインは、頂点の位置属性だけをフェッチし、そして、シェーディングするからである。カルパイプラインは、生成された決定的な結果を使用して、それらの三角形がカリングされるかどうかに関係なく、全ての三角形についての可視性 (visibility) 情報を計算することができる。フルパイプライン (このインスタンスにおいては、リプレイ (replay) パイプラインと呼ばれ得る) は、最終的にラスタライゼーションフェーズへ渡される可視的な三角形だけをシェーディングするために、カリングされた三角形をスキップように可視性情報を消費することができる。

【0206】

任意に、追加の固定機能ロジック1538はまた、機械学習訓練又は推論のための最適化を含む実装のために、固定機能行列乗算などの機械学習アクセラレーションロジックも含むことができる。

【0207】

各グラフィクスサブコア1521A~1521F内には、グラフィクスパイプライン、メディアパイプライン、又はシェーダプログラムによるリクエストに応答してグラフィクス、メディア、及び計算操作を実行するために使用され得る実行リソースの組が含まれる。グラフィクスサブコア1521A~1521Fは、複数のEUアレイ1522A~1522F、1524A~1524F、スレッドディスパッチ及びインタースレッド通信 (TD/IC) ロジック1523A~1523F、3D (例えば、テクスチャ) サンプラ1525A~1525F、メディアサンプラ1506A~1506F、シェーダプロセッサ1527A~1527F、及び共有ローカルメモリ (SLM) 1528A~1528Fを含む。EUアレイ1522A~1522F、1524A~1524Fは夫々、グラフィクス、メディア、計算シェーダプログラムを含むグラフィクス、メディア、又は計算操作のサービスにおいて、浮動小数点及び整数/固定小数点ロジック演算を実行可能な汎用グラフィクス処理ユニットである複数の実行ユニットを含む。TD/ICロジック1523A~1523Fは、サブコア内の実行ユニットについてローカルスレッドディスパッチ及びスレッド制御動作を実行し、サブコアの実行ユニットで実行されるスレッド間の通信を促進する。3Dサンプラ1525A~1525Fは、テクスチャ又は他の3Dグラフィクス関連データをメモリ内に読み込むことができる。3Dサンプラは、所与のテクスチャに関連したテクスチャフォーマット及び構成されたサンプル状態に基づいて異なるようにテクスチャデータを読み込むことができる。メディアサンプラ1506A~1506Fは、メディアデータに関連したタイプ及びフォーマットに基づいて同様の読み込み動作を実行することができる。例えば、各グラフィクスサブコア1521A~1521Fは、代替的に、統合された3D及びメディアサンプラを含むことができる。サブコア1521A~1521Fの夫々の中の実行ユニットで実行されるスレッドは、スレッドグループ内で実行されているスレッドがオンチップメモリの共通プールを用いて実行されることを可能にするよう、各サブコア内の共有ローカルメモリ1528A~1528Fを利用することができる。

【0208】

図15Cは、本願で記載されている実施形態に従うグラフィクスプロセッサ、例えば、グラフィクスプロセッサ1508、及び/又は計算アクセラレータとして構成され得る汎用グラフィクス処理ユニット (GPGPU) 1570のブロック図である。GPGPU1570は、1つ以上のシステム及び/又はメモリバスを介してホストプロセッサ (例えば、1つ以上のCPU1546) 及びメモリ1571、1572と相互接続することができる。メモリ1571は、1つ以上のCPU1546と共有され得るシステムメモリであってよく、一方、メモリ1572は、GPGPU1570に専用であるデバイスメモリである。例えば、GPGPU1570内のコンポーネントとデバイスメモリ1572とは、1つ以上のCPU1546にアクセス可能であるメモリアドレスにマッピングされてよい。メモリ1571及び1572へのアクセスは、メモリコントローラ1568を介して促進されてよい。メモリコントローラ1568は、内部ダイレクトメモリアクセス (DMA) コントローラ1569を含んでよく、あるいは、さもないればDMAコントローラによっ

10

20

30

40

50

て実行されることになる動作を実行するロジックを含むことができる。

【0209】

GPGPU1570は、L2キャッシュ1553、L1キャッシュ1554、命令キャッシュ1555、及び共有メモリ1556を含む複数のキャッシュメモリを含み、それらの少なくとも一部は、キャッシュメモリとしてパーティション化されてもよい。GPGPU1570はまた、複数の計算ユニット1560A～1560Nも含む。各計算ユニット1560A～1560Nは、ベクトルレジスタ1561、スカラーレジスタ1562、ベクトルロジックユニット1563、及びスカラーロジックユニット1564の組を含む。計算ユニット1560A～1560Nはまた、ローカル共有メモリ1565及びプログラムカウンタ1566も含むことができる。計算ユニット1560A～1560Nは、GPGPU1570で実行されるカーネル又はシェダプログラムの実行中に変化することがないデータである定数データを記憶するために使用され得る定数キャッシュ1567と結合することができる。定数キャッシュ1567は、スカラーデータキャッシュであってよく、キャッシュされたデータは、スカラーレジスタ1562内に直接にフェッチされ得る。

10

【0210】

動作中、1つ以上のCPU1546は、アクセス可能なアドレス空間にマッピングされているGPGPU1570のレジスタ又はメモリにコマンドを書き込むことができる。コマンドプロセッサ1557は、レジスタ又はメモリからコマンドを読み出し、それらのコマンドがGPGPU1570内でどのように処理されることになるかを決定することができる。次いで、スレッドディスパッチャ1558が、それらのコマンドを実行する計算ユニット1560A～1560Nにスレッドをディスパッチするために使用され得る。各計算ユニット1560A～1560Nは、条件付きの計算のために独立して構成可能であり、計算の結果をメモリへ条件付きで出力することができる。コマンドプロセッサ1557は、発行されたコマンドが完了するときに、1つ以上のCPU1546に割り込むことができる。

20

【0211】

図16A～16Cは、本願で記載されている実施形態によって、例えば、図15A～15Cに従って提供される更なるグラフィクスプロセッサ及び計算アクセラレータアーキテクチャのブロック図を表す。本願のいずれかの他の図の要素と同じ又は類似した名称を有している図16A～16Cの要素は、他の図で見られるのと同じ要素について記載しており、本願のどこかで記載されているものと同じように動作又は機能することができ、同じコンポーネントを有することができ、そして、他のエンティティへリンクされ得るが、そのように限定されない。

30

【0212】

図16Aは、グラフィクスプロセッサ1600のブロック図であり、グラフィクスプロセッサ1600は、別個のグラフィクス処理ユニットであってよく、あるいは、複数の処理コア、又はメモリデバイス若しくはネットワークインターフェースを含むがこれに限られない他の半導体デバイスと一体化されたグラフィクスプロセッサであってもよい。グラフィクスプロセッサ1600は、グラフィクスプロセッサ1508の変形であってよく、グラフィクスプロセッサ1508の代わりに使用されてよい。従って、本願でのプロセッサ1508と組み合わせられる如何なる特徴も、プロセッサ1600との対応する組み合わせを開示するが、そのように限定されない。グラフィクスプロセッサは、メモリマップドI/Oインターフェースを介してグラフィクスプロセッサ上のレジスタと、及びプロセッサメモリ内に置かれたコマンドと通信してよい。グラフィクスプロセッサ1600は、メモリにアクセスするためのメモリインターフェース1614を含んでよい。メモリインターフェース1614は、ローカルメモリ、ひとついじょうの内部キャッシュ、1つ以上の共有外部キャッシュ、及び／又はシステムメモリへのインターフェースであることができる。

40

【0213】

任意に、グラフィクスプロセッサ1600はまた、表示デバイス1618への表示出力

50

データを駆動するディスプレイコントローラ 1602 も含む。ディスプレイコントローラ 1602 は、ビデオの複数のレイヤ又はユーザインターフェース要素の表示及び合成のための 1 つ以上のオーバーレイプレーン用のハードウェアを含む。表示デバイス 1618 は、内蔵又は外付け表示デバイスであることができる。一実施形態において、表示デバイス 1618 は、仮想現実 (VR) 表示デバイス又は拡張現実 (AR) 表示デバイスなどのヘッドマウント型表示デバイスである。グラフィクスプロセッサ 1600 は、MPEG-2 などの MPEG (Moving Picture Experts Group) フォーマット、H.264/MPEG-4 AVC などの AVC (Advanced Video Coding) フォーマット、H.265/HEVC、AOMedia (Alliance for Open Media) VP8、VP9、及び SMPTE (Society of Motion Picture & Television Engineers) 421M/VC-1、並びに JPEG などの JPEG (Joint Photographic Experts Group) フォーマット及び MJPEG (Motion JPEG) フォーマットを含むがこれらに限られない 1 つ以上のメディア符号化フォーマットへ、又はそのようなフォーマットから、又はそれらのフォーマットの間でメディアを符号化、復号化、又はトランスコードするビデオコーデックエンジン 1606 を含んでよい。

【0214】

グラフィクスプロセッサ 1600 は、例えば、ビット境界ブロック転送 (bit-boundary block transfer) を含む 2 次元 (2D) のラスタライザ動作を実行するブロック画像転送 (BLIT) エンジン 1604 を含んでよい。しかし、代替的に、2D グラフィクス操作は、グラフィクス処理エンジン (GPE) 1610 の 1 つ以上のコンポーネントを用いて実行されてよい。いくつかの実施形態において、GPE 1610 は、3 次元 (3D) グラフィクス操作及びメディア操作を含むグラフィクス操作を実行する計算エンジンである。

【0215】

GPE 1610 は、3D プリミティブ形状 (例えば、長方形、三角形、など) に従って動作する処理機能を使用した 3 次元画像及びシーンのレンダリングなどの 3D 操作を実行する 3D パイプライン 1612 を含んでよい。3D パイプライン 1612 は、要素内で様々なタスクを実行しかつ／あるいは実行スレッドを 3D/メディアサブシステム 1615 に対して生成するプログラム可能な固定された機能を含む。3D パイプライン 1612 が、メディア操作を実行するために使用され得る一方で、GPE 1610 の実施形態は、ビデオ後処理及び画像エンハンスメントなどのメディア操作を実行するために特に使用されるメディアパイプライン 1616 も含む。

【0216】

メディアパイプライン 1616 は、ビデオコーデックエンジン 1606 の代わりに、又はその代理として、ビデオ復号化アクセラレーション、ビデオインターレース解除、及びビデオ符号化アクセラレーションなどの 1 つ以上の特殊化したメディア操作を実行する固定された機能又はプログラム可能なロジックユニットを含んでよい。メディアパイプライン 1616 は、3D/メディアサブシステム 1615 で実行されるスレッドを生成するスレッド生成ユニットを更に含んでよい。生成されたスレッドは、3D/メディアサブシステム 1615 に含まれる 1 つ以上のグラフィクス実行ユニットでのメディア操作のための計算を実行する。

【0217】

3D/メディアサブシステム 1615 は、3D パイプライン 1612 及びメディアパイプライン 1616 によって生成されたスレッドを実行するためのロジックを含んでよい。これらのパイプラインは、スレッド実行要求を 3D/メディアサブシステム 1615 へ送ってよい。3D/メディアサブシステム 1615 は、様々な要求を調停して、利用可能なスレッド実行リソースにディスパッチするスレッドディスパッチロジックを含む。実行リソースは、3D 及びメディアスレッドを処理するグラフィクス実行ユニットのアレイを含む。3D/メディアサブシステム 1615 は、スレッド命令及びデータのための 1 つ以上の内部キャッシュを含んでよい。更には、3D/メディアサブシステム 1615 はまた、スレッド間でデータを共有しかつ出力データを記憶するために、レジスタ及びアドレス可

能メモリを含む共有メモリを含んでもよい。

【0218】

図16Bは、グラフィクスプロセッサ1600の変形であるグラフィクスプロセッサ1620を表し、グラフィクスプロセッサ1600の代わりに使用されてよく、その逆もありである。従って、本願でのプロセッサ1600と組み合わせられる如何なる特徴も、プロセッサ1602との対応する組み合わせを開示するが、そのように限定されない。グラフィクスプロセッサ1620は、本願で記載されている実施形態に従って、タイル上のアーキテクチャを有している。グラフィクスプロセッサ1620は、グラフィクスエンジンタイル1610A～1610D内で図16Aのグラフィクス処理エンジン1610の複数のインスタンスを有しているグラフィクス処理エンジンクラスタ1622を含んでよい。各グラフィクスエンジンタイル1610A～1610Dは、インターコネクト1623A～1623Fの組を介して相互接続され得る。各グラフィクスエンジンタイル1610A～1610Dはまた、メモリインターコネクト1625A～1625Dを介してメモリモジュール又はメモリデバイス1626A～1626Dへも接続され得る。メモリデバイス1626A～1626Dは、如何なるグラフィクスメモリ技術も使用することができる。例えば、メモリデバイス1626A～1626Dは、グラフィクスダブルデータレート（GDDR）メモリであってよい。メモリデバイス1626A～1626Dは、それらの各々のグラフィクスエンジンタイル1610A～1610Dとともにオンダイであることができる高バンド幅メモリ（HBM）モジュールであってよい。メモリデバイス1626A～1626Dは、それらの各々のグラフィクスエンジンタイル1610A～1610Dの上にスタックされ得る積層型メモリデバイスであってもよい。各グラフィクスエンジンタイル1610A～1610D及び関連するメモリ1626A～1626Dは、図24B～24Dで更に詳細に説明されるように、ベースダイ又はベース基板に固着されている別個のチップレット上に存在してよい。

【0219】

グラフィクスプロセッサ1620は、メモリデバイス1626A～1626Dが関連するグラフィクスエンジンタイル1610A～1610Dと結合されている非一様メモリアクセス（NUMA）で構成されてよい。所与のメモリデバイスは、それが直接に接続されているタイル以外のグラフィクスエンジンタイルによってアクセスされてよい。しかし、メモリデバイス1626A～1626Dへのアクセスレイテンシは、ローカルタイルにアクセスするときに最も低くなる。一実施形態において、キャッシュコヒーレントNUMA（ccNUMA）システムが有効にされてよく、これは、タイルインターコネクト1623A～1623Fを使用して、グラフィクスエンジンタイル1610A～1610D内のキャッシュコントローラ間の通信が、1つよりも多いキャッシュが同じメモリ位置を記憶するときに、一貫したメモリ画像を保持することを可能にする。

【0220】

グラフィクス処理エンジンクラスタ1622は、オンチップ又はオンパッケージのファブリックインターコネクト1624と接続することができる。ファブリックインターコネクト1624は、グラフィクスエンジンタイル1610A～1610Dと、ビデオコーデック1606及び1つ以上のコピーエンジン1604などのコンポーネントとの間の通信を可能にすることができる。コピーエンジン1604は、メモリデバイス1626A～1626D及びグラフィクスプロセッサ1620の外にあるメモリ（例えば、システムメモリ）からの、それらへの、及びそれらの間のデータの移動のために使用され得る。ファブリックインターコネクト1624はまた、グラフィクスエンジンタイル1610A～1610Dを相互接続するために使用され得る。グラフィクスプロセッサ1620は、外部表示デバイス1618との接続を可能にするディスプレイコントローラ1602を任意に含んでもよい。グラフィクスプロセッサ1620はまた、グラフィクス又は計算アクセラレータとしても構成されてよい。アクセラレータ構成では、ディスプレイコントローラ1602及び表示デバイス1618は、省略されてもよい。

【0221】

グラフィクスプロセッサ1620は、ホストインターフェース1628を介してホストシステムへ接続することができる。ホストインターフェース1628は、グラフィクスプロセッサ1620、システムメモリ、及び／又は他のシステムコンポーネントの間の通信を可能にすることができる。ホストインターフェース1628は、例えば、PCIエクスプレスバス又は他のタイプのホストシステムインターフェースであることができる。

【0222】

図16Cは、本願で記載されている実施形態に従う計算アクセラレータ1630を表す。計算アクセラレータ1630は、図16Bのグラフィクス処理エンジン1620とのアーキテクチャ類似性を含むことができ、計算アクセラレーションのために最適化されている。計算エンジンクラス1632は、並列又はベクトルベースの汎用計算動作のために最適化されている実行ユニットを含む計算エンジンタイル1640A～1640Dの組を含むことができる。計算エンジンタイル1640A～1640Dは、固定機能グラフィクス処理ロジックを含まなくてもよいが、いくつかの実施形態において、計算エンジンタイル1640A～1640Dのうちの1つ以上は、メディアアクセラレーションを実行するためのロジックを含むことができる。計算エンジンタイル1640A～1640Dは、メモリインターコネクト1625A～1625Dを介してメモリ1626A～1626Dへ接続することができる。メモリ1626A～1626D及びメモリインターコネクト1625A～1625Dは、グラフィクスプロセッサ1620で見られるような同様の技術であってよく、あるいは、異なることができる。グラフィクス計算エンジンタイル1640A～1640Dはまた、タイルインターコネクト1623A～1623Fの組を介して相互接続され得、ファブリックインターコネクト1624と接続されかつ／あるいはそれによって相互接続されてもよい。一実施形態において、計算アクセラレータ1630は、デバイスワイドキャッシュとして構成され得るラージL3キャッシュ1636を含む。計算アクセラレータ1630はまた、図16Bのグラフィクスプロセッサ1620と同様にしてホストインターフェース1628を介してホストプロセッサ及びメモリへ接続することもできる。

【0223】

〔グラフィクス処理エンジン〕

図17は、いくつかの実施形態に従うグラフィクスプロセッサのグラフィクス処理エンジン1710のブロック図である。グラフィクス処理エンジン(GPE)1710は、図16Aに示されたGPE1610の変形であってよく、図16Bのグラフィクスエンジンタイル1610A～1610Dを表してもよい。本願のどこかの図の要素と同じ又は類似した名称を持っている図17の要素は、他の図で見られるのと同じ要素について記載しており、本願のどこかで記載されているのと同じように動作又は機能することができ、同じコンポーネントを有することができ、他のエンティティへリンクされ得るが、そのように限定されない。例えば、図16Aの3Dパイプライン1612及びメディアパイプライン1616は、図17でも表されている。メディアパイプライン1616は、GPE1710のいくつかの実施形態では任意であり、GPE1710内に明示的に含まれてなくてもよい。例えば、少なくとも1つの実施形態では、別個のメディア及び／又は画像プロセッサがGPE1710へ結合される。

【0224】

GPE1710は、コマンドストリームを3Dパイプライン1612及び／又はメディアパイプライン1616へ供給するコマンドストリーマ1703と結合するか、又はそれを含んでよい。代替的に、又は追加的に、コマンドストリーム1703は、統合リターンバッファ1718へ直接に結合されてよく、統合リターンバッファ1718は、グラフィクスコアアレイ1714へ通信可能に結合されてよい。任意に、コマンドストリーマ1703は、システムメモリ、又は内部キャッシュメモリ及び共有キャッシュメモリのうちの1つ以上であることができるメモリと結合される。コマンドストリーマ1703は、メモリからコマンドを受け取ってよく、コマンドを3Dパイプライン1612及び／又はメディアパイプライン1616へ送る。コマンドは、3Dパイプライン1612及びメディア

10

20

30

40

50

パイプライン 1 6 1 6 に対するコマンドを記憶するリングバッファからフェッチされた指令である。リングバッファは、複数のコマンドのバッチを記憶しているバッチコマンドバッファを更に含むことができる。3 Dパイプライン 1 6 1 2 に対するコマンドはまた、例えば、3 Dパイプライン 1 6 1 2 に対する頂点及びジオメトリデータ、及び／又はメディアパイプライン 1 6 1 6 のための画像データ及びメモリオブジェクトなどの、しかしこれらに限られない、メモリに記憶されているデータへの参照を含むこともできる。3 Dパイプライン 1 6 1 2 及びメディアパイプライン 1 6 1 6 は、各々のパイプライン内のロジックにより動作を実行することによって、あるいは、1 つ以上の実行スレッドをグラフィクスコアアレイ 1 7 1 4 にディスパッチすることによって、コマンド及びデータを処理する。グラフィクスコアアレイ 1 7 1 4 は、グラフィクスコア（例えば、グラフィクスコア 1 7 1 5 A、グラフィクスコア 1 7 1 5 B）の 1 つ以上のブロックを含んでよく、各ブロックは、1 つ以上のグラフィクスコアを含む。各グラフィクスコアは、グラフィクス及び計算操作を実行する汎用及びグラフィクス特有実行ロジックと、固定機能テクスチャ処理及び／又は機械学習及び人工知能アクセラレーションロジックとを含むグラフィクス実行リソースの組を含む。

【0 2 2 5】

様々な実施形態において、3 Dパイプライン 1 6 1 2 は、命令を処理しかつ実行スレッドをグラフィクスコアアレイ 1 7 1 4 にディスパッチすることによって、頂点シェーダ、ジオメトリシェーダ、ピクセルシェーダ、フラグメントシェーダ、計算シェーダ、又は他のシェーダプログラムなどの 1 つ以上のシェーダプログラムを処理する固定機能及びプログラム可能ロジックを含むことができる。グラフィクスコアアレイ 1 7 1 4 は、これらのシェーダプログラムを処理する際に使用される実行リソースの統合されたブロックを提供する。グラフィクスコアアレイ 1 7 1 4 のグラフィクスコア 1 7 1 5 A～1 7 1 5 B 内の多目的実行ロジック（例えば、実行ユニット）は、様々な 3 D A P I シェーダ言語のサポートを含み、複数のシェーダに関連した複数の同時実行スレッドを実行することができる。

【0 2 2 6】

グラフィクスコアアレイ 1 7 1 4 は、ビデオ及び／又は画像処理などのメディア機能を実行する実行ロジックを含んでよい。実行ユニットは、グラフィクス処理動作に加えて、並列な汎用計算動作を実行するようプログラム可能である汎用ロジックを含んでよい。汎用ロジックは、図 1 4 のプロセッサコア 1 4 0 7 又は図 1 5 A のコア 1 5 0 2 A～1 5 0 2 N 内の汎用ロジックと並行して又はそれと連動して処理動作を実行することができる。

【0 2 2 7】

グラフィクスコアアレイ 1 7 1 4 でのスレッド実行によって生成された出力データは、統合リターンバッファ（unified return buffer, URB）1 7 1 8 内のメモリヘッダを出力することができる。URB 1 7 1 8 は、複数のスレッドのデータを記憶することができる。URB 1 7 1 8 は、グラフィクスコアアレイ 1 7 1 4 で実行される異なるスレッド間データを送るために使用されてよい。URB 1 7 1 8 は、グラフィクスコアアレイ 1 7 1 4 及び共有機能ロジック 1 7 2 0 内の固定機能ロジックでのスレッド間の同期化のために更に使用されてよい。

【0 2 2 8】

任意に、グラフィクスコアアレイ 1 7 1 4 は、アレイが、G P E 1 7 1 0 の目標電力及び性能レベルに基づいて可変な数の実行ユニットを夫々が有する可変な数のグラフィクスコアを含むように、スケーリング可能であってよい。実行リソースは、実行リソースが必要に応じて有効又は無効にされ得るように、動的にスケーリング可能であってよい。

【0 2 2 9】

グラフィクスコアアレイ 1 7 1 4 は、グラフィクスコアアレイ内のグラフィクスコア間で共有される複数のリソースを含む共有機能ロジック 1 7 2 0 と結合する。共有機能ロジック 1 7 2 0 内の共有機能は、特殊化した補助機能をグラフィクスコアアレイ 1 7 1 4 にもたらすハードウェアロジックユニットである。様々な実施形態において、共有機能ロジ

10

20

30

40

50

ック1720は、サンブラ1721、マス（math）1722、及びインタースレッド通信（ITC）1723ロジックを含むが、これらに限られない。更に、共有機能ロジック1720内の1つ以上のキャッシュ1725が実装されてもよい。

【0230】

共有機能は、少なくとも、所与の特殊化した機能に対する需要がグラフィクスコアアレイ1714内に含めるには不十分である場合に、実装される。代わりに、この特殊化した機能の単一のインスタンス化が、共有機能ロジック1720でスタンドアロンのエンティティとして実装され、グラフィクスコアアレイ1714内の実行リソース間で共有される。グラフィクスコアアレイ1714の間で共有され、グラフィクスコアアレイ1714内に含まれる機能の正確な組は、実施形態ごとに様々である。グラフィクスコアアレイ1714によって広範に使用される共有機能ロジック1720内の特定の共有機能は、グラフィクスコアアレイ1714内の共有機能ロジック1716内に含まれてもよい。任意に、グラフィクスコアアレイ1714内の共有機能ロジック1716は、共有機能ロジック1720内のいくつか又は全てのロジックを含むことができる。共有機能ロジック1720内の全てのロジック要素は、グラフィクスコアアレイ1714の共有機能ロジック1716内に複製されてよい。代替的に、共有機能ロジック1720は、グラフィクスコアアレイ1714内の共有機能ロジック1716を支持して除外される。

【0231】

〔実行ユニット〕

図18A～18Bは、本願で記載されている実施形態に従って、グラフィクスプロセッサコアで用いられる処理要素のアレイを含むスレッド実行ロジック1800を表す。本願のいずれかの他の図の要素と同じ又は類似した名称を有している図18A～18Bの要素は、他の図で見られるのと同じ要素について記載しており、本願のどこかで記載されているのと同じように動作又は機能することができ、同じコンポーネントを有することができ、他のエンティティへリンクされ得るが、そのように限定されない。図18A～18Bは、図15Bの各サブコア1521A～1521Fにより表されたハードウェアロジックを表し得るスレッド実行ロジック1800の概要を表す。図18Aは、汎用グラフィクスプロセッサ内の実行ユニットを表し、一方、図18Bは、計算アクセラレータ内で使用され得る実行ユニットを表す。

【0232】

図18Aに表されているように、スレッド実行ロジック1800は、シェーダプロセッサ1802、スレッドディスパッチャ1804、命令キャッシュ1806、複数の実行ユニット1808A～1808Nを含むスケーラブル実行ユニットアレイ、サンブラ1810、共有ローカルメモリ1811、データキャッシュ1812、及びデータポート1814を含んでよい。任意に、スケーラブル実行ユニットアレイは、ワークロードの計算要件に基づいて1つ以上の実行ユニット（例えば、実行ユニット1808A、1808B、1808C、1808D、乃至1808N-1及び1808Nのいずれか）を有効又は無効にすることによって、動的にスケーリング可能である。含まれているコンポーネントは、コンポーネントの夫々へリンクするインターコネクトファブリックを介して相互接続されてよい。スレッド実行ロジック1800は、命令キャッシュ1806、データポート1814、サンブラ1810、及び実行ユニット1808A～1808Nのうちの1つ以上を通じて、システムメモリ又はキャッシュメモリなどのメモリへの1つ以上の接続を含んでよい。各実行ユニット（例えば、1808A）は、各スレッドについて並行して複数のデータ要素を処理しながら、複数の同時ハードウェアスレッドを実行することができるスタンドアロンのプログラム可能な汎用計算ユニットであってよい。様々な実施形態において、実行ユニット1808A～1808Nのアレイは、任意の数の個別的な実行ユニットを含むようスケーリング可能である。

【0233】

実行ユニット1808A～1808Nは、シェーダプログラムを実行するために主として使用されてよい。シェーダプロセッサ1802は、様々なシェーダプログラムを処理し

10

20

30

40

50

、シェーダプログラムに関連した実行スレッドをスレッドディスパッチャ1804によりディスパッチすることができる。スレッドディスパッチャは、グラフィクス及びメディアパイプラインからのスレッド開始リクエストを調停し、要求されたスレッドを実行ユニット1808A~1808Nの中の1つ以上の実行ユニットでインスタンス化するロジックを含んでよい。例えば、ジオメトリパイプラインは、頂点、テッセレーション、又はジオメトリシェーダを処理のためにスレッド実行ロジックにディスパッチすることができる。任意に、スレッドディスパッチャ1804はまた、実行中のシェーダプログラムからのランタイムスレッド生成リクエストを処理することもできる。

【0234】

実行ユニット1808A~1808Nは、グラフィクスライブラリ（例えば、Direct 3D及びOpenGL）からのシェーダプログラムが最小限の変換で実行されるように、多くの標準3Dグラフィクスシェーダ命令のネイティブサポートを含む命令セットをサポートしてよい。実行ユニットは、頂点及びジオメトリ処理（例えば、頂点プログラム、ジオメトリプログラム、頂点シェーダ）、ピクセル処理（例えば、ピクセルシェーダ、フラグメントシェーダ）、並びに汎用処理（例えば、計算及びメディアシェーダ）をサポートする。実行ユニット1808A~1808Nの夫々は、多重発行（multi-issue）のSIMD（single instruction multiple data）実行が可能であり、マルチスレッド操作は、より高いレイテンシのメモリアクセスに直面して効率的な実行環境を可能にする。各実行ユニット内の各ハードウェアスレッドは、専用の高バンド幅レジスタファイル及び関連する独立したスレッド状態を有する。実行は、整数、単精度及び倍精度浮動小数点演算、SIMD分岐機能、論理演算、超越演算（transcendental operations）、及び他の種々雑多な演算が可能なパイプラインにクロックごとに多重発行する。メモリ又は共有機能の1つからのデータを待つ間、実行ユニット1808A~1808N内の依存関係ロジック（dependency logic）は、要求されているデータが返されるまで、待機中のスレッドをスリープさせる。待機中のスレッドがスリープしている間、ハードウェアリソースは、他のスレッドを処理することにあてられてよい。例えば、頂点シェーダ演算に関連した遅延中に、実行ユニットは、ピクセルシェーダ、フラグメントシェーダ、又は図21に表されている頂点シェーダ2107などの別の頂点シェーダを含む他のタイプのシェーダプログラムの演算を実行することができる。様々な実施形態は、SIMDの使用の代わりとして、又はSIMDの使用に加えて、SIMT（single instruction multiple thread）の使用による実行を使用するよう適用可能である。SIMDコア又は動作への言及は、SIMTにも適用することができ、あるいは、SIMTと組み合わせられたSIMDにも適用することができる。

【0235】

実行ユニット1808A~1808Nの中の各実行ユニットは、データ要素のアレイに作用する。データ要素の数は、命令のための「実行サイズ」又はチャネル数である実行チャネルは、命令内のデータ要素アクセス、マスキング、及びフロー制御のための実行のロジックユニットである。チャネルの数は、特定のグラフィクスプロセッサのための物理的な算術論理ユニット（ALU）、浮動小数点ユニット（FPU）、又は他のロジックユニット（例えば、テンソルコア、レイトレーシングコア、など）の数とは無関係であり得る。更には、実行ユニット1808A~1808Nは、整数及び浮動小数点データタイプをサポートしてもよい。

【0236】

実行ユニット命令セットは、SIMD命令を含む。様々なデータ要素が、パックされたデータタイプとしてレジスタに記憶可能であり、実行ユニットは、要素のデータサイズに基づいて様々な要素を処理することになる。例えば、256ビットワイドのベクトルに作用する場合に、ベクトルの256ビットがレジスタに記憶され、実行ユニットは、4つの別個の184ビットのパックされたデータ要素（クワッドワード（Quad-Word, QW）サイズデータ要素）、8つの別個の32ビットのパックされたデータ要素（ダブルワード（Double Word, DW）サイズデータ要素）、16個の別個の16ビットのパックされ

たデータ要素（ワード（W）サイズデータ要素）、又は32個の別個の8ビットデータ要素（バイト（B）サイズデータ要素）としてベクトルに作用する。しかし、異なるベクトル幅及びレジスタサイズが可能である。

【0237】

任意に、1つ以上の実行ユニットは、融合したEUに共通するスレッド制御ロジック（1807A～1807N）を有する融合した実行ユニット1809A～1809Nにまとめられ得る。複数のEUは、EUグループに融合され得る。融合されたEUグループ内の各EUは、別々のSIMDハードウェアスレッドを実行するよう構成され得る。融合されたEUグループ内のEUの数は、実施形態により様々であることができる。更には、様々なSIMD幅がEUごとに実行可能であり、SIMD8、SIMD16、及びSIMD32があるが、これらに限られない。融合されたグラフィクス実行ユニット1809A～1809Nの夫々は、少なくとも2つの実行ユニットを含む。例えば、融合された実行ユニット1809Aは、第1EU1808Aと、第2EU1808Bと、第1EU1808A及び第2EU1808Bに共通する第3制御ロジック1807Aとを含む。スレッド制御ロジック1807Aは、融合されたグラフィクス実行ユニット1809Aで実行されるスレッドを制御して、融合された実行ユニット1809A～1809N内の各EUが共通命令ポインタレジスタを用いて実行することを可能にする。

【0238】

1つ以上の内部命令キャッシュ（例えば、1806）は、実行ユニットのためのスレッド命令をキャッシュするようスレッド実行ロジック1800に含まれる。1つ以上のデータキャッシュ（例えば、1812）が、スレッド実行中にスレッドデータをキャッシュするようスレッド実行ロジック1800に含まれてもよい。実行ロジック1800で実行されるスレッドはまた、共有ローカルメモリ1811に、明示的に管理されたデータを格納することもできる。サンブラ1810が、3D操作のためのテクスチャサンプリングと、メディア操作のためのメディアサンブラとを提供するよう含まれてもよい。サンブラ1810は、サンプリングされたデータを実行ユニットへ供給する前にサンプリングプロセス中にテクスチャ又はメディアデータを処理する特殊化したテクスチャ又はメディアサンプリング機能を含む。

【0239】

実行中、グラフィクス及びメディアパイプラインは、スレッド生成及びディスパッチロジックを介してスレッド実行ロジック1800へスレッド開始リクエストを送る。幾何学オブジェクトのグループが処理されピクセルデータにラスタライズされると、シェーダプロセッサ1802内のピクセルプロセッサロジック（例えば、ピクセルシェーダロジック、フラグメントシェーダロジック、など）が、出力情報を更に計算して、結果を出力表面（例えば、カラーバッファ、デプスバッファ、ステンシルバッファ、など）に書き込めるために呼び出される。ピクセルシェーダ又はフラグメントシェーダは、ラスタライズされたオブジェクトにわたって補間されるべきである様々な頂点属性の値を計算してよい。シェーダプロセッサ1802内のピクセルプロセッサロジックは、次いで、アプリケーションプログラミングインターフェース（API）により供給されたピクセル又はフラグメントシェーダプログラムを実行してよい。シェーダプログラムを実行するために、シェーダプロセッサ1802は、スレッドディスパッチャ1804を介して実行ユニット（例えば、1808A）にスレッドをディスパッチする。シェーダプロセッサ1802は、サンブラ1810内のテクスチャサンプリングロジックを使用して、メモリに記憶されているテクスチャマップ内のテクスチャデータにアクセスしてよい。テクスチャデータ及び入力ジオメトリデータに対する算術演算は、各幾何学フラグメントについてピクセル色データを計算するか、あるいは、更なる処理から1つ以上のピクセルを捨てる。

【0240】

更には、データポート1814は、スレッド実行ロジック1800が処理されたデータをグラフィクスプロセッサ出力パイプラインで更なる処理のためにメモリへ出力するためのメモリアクセスメカニズムを提供してよい。データポート1814は、データポートに

10

20

30

40

50

よるメモリアクセスのためにデータをキャッシュする1つ以上のキャッシュメモリ（例えば、データキャッシュ1812）を含むか又はそれへ結合してよい。

【0241】

任意に、実行ロジック1800はまた、レイトレーシングアクセラレーション機能を提供することができるレイトレーサ1805も含むことができる。レイトレーサ1805は、レイ生成のための命令／関数を含むレイトレーシング命令セットをサポートすることができる。レイトレーシング命令セットは、図3Cのレイトレーシングコア372によってサポートされるレイトレーシング命令セットと同様であっても異なってもよい。

【0242】

図18Bは、実行ユニット1808の例示的な内部詳細を表す。グラフィクス実行ユニット1808は、命令フェッチユニット1837、汎用レジスタファイルアレイ（GFR）1824、アーキテクチャレジスタファイルアレイ（ARF）1826、スレッドアービタ1822、送信ユニット1830、分岐ユニット1832、SIMD浮動小数点ユニット（FPU）1834、及び任意に、専用の整数SIMD ALU1835の組を含むことができる。GFR1824及びARF1826は、グラフィクス実行ユニット1808でアクティブであり得る各同時ハードウェアスレッドに関連した汎用レジスタファイル及びアーキテクチャレジスタファイルの組を含む。スレッドごとのアーキテクチャ状態は、ARF1826で保持されてよく、一方、スレッド実行中に使用されるデータは、GFR1824に格納される。各スレッドの命令ポインタを含む各スレッドの実行状態は、ARF1826内のスレッド固有レジスタに保持され得る。

【0243】

グラフィクス実行ユニット1808は、同時マルチスレッディング（simultaneous multi-threading, SMT）及び微細（fine-grained）インターリーブマルチスレッディング（interleaved multi-threading, IMT）の組み合わせであるアーキテクチャを有してよい。アーキテクチャは、同時スレッドの目標数及び実行ユニットごとのレジスタの数に基づいて設計時に微調整され得るモジュール構成を有してよく、実行ユニットリソースは、複数の同時スレッドを実行するために使用されるロジックにわたって分割される。グラフィクス実行ユニット1808によって実行される可能性がある論理スレッドの数は、ハードウェアスレッドの数に限定されず、複数の論理スレッドが、各ハードウェアスレッドに割り当てられ得る。

【0244】

任意に、グラフィクス実行ユニット1808は、夫々が異なった命令であってよい複数の命令を同時発行（co-issue）することができる。グラフィクス実行ユニット1808のスレッドアービタ1822は、命令を送信ユニット1830、分岐ユニット1832、又はSIMD FPU1834の1つへ実行のためにディスパッチすることができる。各実行スレッドは、GFR1824内の128個の汎用レジスタにアクセスすることができ、各レジスタは、32ビットデータ要素のSIMD8要素ベクトルとしてアクセス可能な32バイトを記憶することができる。各実行ユニットスレッドは、GFR1824内の4Kバイトへのアクセスを有してよいが、実施形態はそうのように限定されず、より多い又はより少ないレジスタリソースが他の実施形態では設けられてもよい。グラフィクス実行ユニット1808は、計算動作を独立して実行することができる7つのハードウェアスレッドにパーティション化されてよいが、実行ユニットごとのスレッドの数も、実施形態により様々であることができ、例えば、最大16個のハードウェアスレッドがサポートされてよい。7つのスレッドが4Kバイトにアクセスし得る例示的な実施形態では、GFR1824は、全部28Kバイトを記憶することができる。16個のスレッドが4Kバイトにアクセスし得る他の例示的な実施形態では、GFR1824は、全部で64Kバイトを記憶することができる。実行ユニットごとのスレッドの数は、しかしながら、これらの例に限られず、与えられている数よりも多くても少なくともよい。フレキシブルアドレッシングモードは、有効により幅広のレジスタを構成するために、又はストライド型長方形ブロックデータ構造を表すために、レジスタが一緒にアドレッシングされることを可能にすること

ができる。

【0245】

追加的に、又は代替的に、メモリ動作、サンブラ動作、及び他のより長いレイテンシのシステム通信は、メッセージパッシング送信ユニット1830によって実行される「送信」命令によりディスパッチされてよい。分岐命令は、SIMDダイバージェンス及び結果として起こるコンバージェンスを促進するよう専用の分岐ユニット1832にディスパッチされてよい。

【0246】

グラフィクス実行ユニット1808は、浮動小数点演算を実行するよう1つ以上のSIMD浮動小数点ユニット(FPU)1834を含んでよい。FPU1834はまた、整数計算をサポートしてもよい。いくつかのインスタンスでは、FPU1834は、最大M個の32ビット浮動小数点(若しくは整数)演算をSIMD実行するか、あるいは、最大2M個の16ビット整数又は16ビット浮動小数点演算をSIMD実行することができる。任意に、FPUの少なくとも1つは、高スループット超越数学関数及び倍精度184ビット浮動小数点をサポートするよう、拡張された数学能力を提供する。8ビット整数SIMD ALU1835の組も存在してよく、機械学習計算に関連した動作を実行するよう特に最適化されてよい。

【0247】

任意に、グラフィクス実行ユニット1808の複数のインスタンスのアレイは、グラフィクスサブコアグルーピング(例えば、サブスライス)でインスタンス化され得る。スケラビリティについては、製品創案者が、サブコアグルーピングごとに実行ユニットの正確な数を選択することができる。実行ユニット1808は、複数の実行チャンネルにわたって命令を実行してもよい。更には、グラフィクス実行ユニット1808で実行される各スレッドは、異なるチャンネルで実行されてもよい。

【0248】

図19は、更なる例示的な実行ユニット1900を表す。本願のいずれかの他の図の要素と同じ又は類似した名称を有している図19の要素は、他の図で見られるのと同じ要素について記載しており、本願のどこかで記載されているのと同じように動作又は機能することができ、同じコンポーネントを有することができ、他のエンティティへリンクされ得るが、そのように限定されない。実行ユニット1900は、例えば、図16Cで見られた計算エンジンタイル1640A~1640Dで使用される、コンピュータで最適化された実行ユニットであってよいが、そのように限定されない。実行ユニット1900はまた、図16Bで見られたグラフィクスエンジンタイル1610A~1610Dで使用されてもよい。実行ユニット1900は、スレッド制御ユニット1901、スレッド状態ユニット1902、命令フェッチ/プリフェッチユニット1903、及び命令復号化ユニット1904を含んでよい。実行ユニット1900は、実行ユニット内のハードウェアスレッドに割り当てられ得るレジスタを記憶するレジスタファイル1906を更に含んでもよい。実行ユニット1900は、送信ユニット1907及び分岐ユニット1908を更に含んでもよい。送信ユニット1907及び分岐ユニット1908は、図18Bのグラフィクス実行ユニット1808の送信ユニット1830及び分岐ユニット1832と同様に動作してよい。

【0249】

実行ユニット1900はまた、多種多様なタイプの機能ユニットを含む計算ユニット1910も含むことができる。計算ユニット1910はまた、算術論理ユニットのアレイを含むALUユニット1911を含んでもよい。ALUユニット1911は、64ビット、32ビット、及び16ビット整数及び浮動小数点演算を実行するよう構成され得る。整数及び浮動小数点演算は、同時に実行されてよい。計算ユニット1910はまた、シストリックアレイ1912及びマスユニット1913も含むことができる。シストリックアレイ1912は、シストリック方式でベクトル又は他のデータ並列演算を実行するために使用され得るデータ処理ユニットのW幅及びD深さのネットワークを含む。シストリックアレ

10

20

30

40

50

イ 1 9 1 2 は、行列内積演算などの行列演算を実行するよう構成され得る。シストリックアレイ 1 9 1 2 は、8 ビット及び 4 ビット整数演算に加えて、1 6 ビット浮動小数点演算もサポートしてよい。シストリックアレイ 1 9 1 2 は、機械学習演算を加速させるよう構成されてよい。シストリックアレイ 1 9 1 2 は、1 6 ビット浮動小数点フォーマットである `bfloat16` の支援を受けて構成されてよい。マスユニット 1 9 1 3 は、ALU ユニット 1 9 1 1 よりも効率的かつ低電力の方法で数学演算の特定のサブセットを実行するために含まれ得る。マスユニット 1 9 1 3 は、記載されている他の実施形態によって提供されているグラフィクス処理エンジンの共有関数ロジックで見受けられるマスロジック、例えば、図 1 7 の共有関数ロジック 1 7 2 0 のマスロジック 1 7 2 2 を含むことができる。マスユニット 1 9 1 3 は、3 2 ビット及び 6 4 ビット浮動小数点演算を実行するよう構成され得る。

10

【0250】

スレッド制御ユニット 1 9 0 1 は、実行ユニット内のスレッドの実行を制御するためのロジックを含む。スレッド制御ユニット 1 9 0 1 は、実行ユニット 1 9 0 0 内のスレッドの実行を開始、停止、及びブリエンプトするスレッド調停ロジックを含むことができる。スレッド状態ユニット 1 9 0 2 は、実行ユニット 1 9 0 0 で実行されるよう割り当てられたスレッドのスレッド状態を記憶するために使用され得る。実行ユニット 1 9 0 0 内のスレッド状態を記憶することは、それらのスレッドがブロック状態又はアイドル状態にあるときにスレッドの迅速なブリエンプションを可能にする。命令フェッチ／プリフェッチユニット 1 9 0 3 は、より上位の実行ロジックの命令キャッシュ（例えば、図 1 8 A で見られるような命令キャッシュ 1 8 0 6）から命令をフェッチすることができる。命令フェッチ／プリフェッチユニット 1 9 0 3 はまた、現在実行中のスレッドの解析に基づいて、命令が命令キャッシュにロードされるためのプリフェッチリクエストを発行することもできる。命令復号化ユニット 1 9 0 4 は、計算ユニットによって実行されるべき命令を復号するために使用され得る。命令復号化ユニット 1 9 0 4 は、複合命令（`complex instructions`）を、構成するマイクロオペレーションに復号するために、二次デコーダとして使用され得る。

20

【0251】

実行ユニット 1 9 0 0 は、実行ユニット 1 9 0 0 で実行されるハードウェアスレッドによって使用され得るレジスタファイル 1 9 0 6 を更に含む。レジスタファイル 1 9 0 6 内のレジスタは、実行ユニット 1 9 0 0 の計算ユニット 1 9 1 0 内で複数の同時スレッドを実行するために使用されるロジックにわたって分割され得る。グラフィクス実行ユニット 1 9 0 0 によって実行され得る論理スレッドの数は、ハードウェアスレッドの数に限定されず、複数の論理スレッドが各ハードウェアスレッドに割り当てられ得る。レジスタファイル 1 9 0 6 のサイズは、サポートされるハードウェアスレッドの数に基づいて実施形態ごとに様々であることができる。レジスタ・リネーミングは、レジスタをハードウェアスレッドに動的に割り当てるために使用されてよい。

30

【0252】

図 2 0 は、グラフィクスプロセッサ命令フォーマット 2 0 0 0 を表すブロック図である。グラフィクスプロセッサ実行ユニットは、複数のフォーマットで命令を有する命令セットをサポートする。実線ボックスは、一般的に実行ユニット命令に含まれているコンポーネントを表し、一方、破線は、任意であるか、あるいは、命令のサブセットにしか含まれないコンポーネントを含む。記載及び例示されている命令フォーマット 2 0 0 0 は、それらが、命令が処理されると命令復号化から生じるマイクロオペレーションとは対照的に、実行ユニットへ供給される命令であるという点で、マクロ命令である。

40

【0253】

本願で記載されているグラフィクスプロセッサ実行ユニットは、元々、1 2 8 ビット命令フォーマット 2 0 1 0 で命令をサポートする。6 4 ビットコンパクト命令フォーマット 2 0 3 0 は、選択された命令、命令オプション、及びオペランドの数に基づいて、いくつかの命令について利用可能である。元々の 1 2 8 ビット命令フォーマット 2 0 1 0 は、全

50

での命令オプションへのアクセスを提供し、一方、64ビットフォーマット2030では、いくつかのオプション及び動作が制限される。64ビットフォーマット2030で利用可能な元々の命令は、実施形態により異なる。命令は、インデックスフィールド2013内のインデックス値の組を用いて部分的に圧縮される。実行ユニットハードウェアは、インデックス値に基づいて圧縮テーブルの組を参照し、そして、圧縮テーブル出力を使用して、128ビット命令フォーマット2010内の元々の命令を再構成する。他のサイズ及びフォーマットの命令が使用可能である。

【0254】

各フォーマットについて、命令オペコード2012は、実行ユニットが実行すべきである動作を定義する。実行ユニットは、各オペランドの複数のデータ要素にわたって並行して各命令を実行する。例えば、加算命令に回答して、実行ユニットは、テクスチャ要素又はピクチャ要素を表す各色チャンネルにわたって同時の加算演算を実行する。デフォルトで、実行ユニットは、オペランドの全データチャンネルにわたって各命令を実行する。命令制御フィールド2014は、チャンネル選択（例えば、プレディケーション（predication））及びデータチャンネル順序（スウィズル（swizzle））などの特定の実行オプションに対する制御を可能にし得る。128ビット命令フォーマット2010での命令について、実行サイズフィールド2016は、並行して実行されるデータチャンネルの数を制限する。実行サイズフィールド2016は、64ビットコンパクト命令フォーマット2030では使用されなくてもよい。

【0255】

いくつかの実行ユニット命令は、2つのソースオペランドsrc0 2020及びsrc1 2022と、1つのあて先2018とを含む最大3つのオペランドを有する。実行ユニットは、あて先の1つが暗示される二重あて先命令をサポートしてよい。データ操作命令は、第3ソースオペランド（例えば、SRC2 2024）を有することができ、命令オペコード2012は、ソースオペランドの数を決定する。命令の最後のソースオペランドは、命令とともに渡される即時（例えば、ハードコードされた）値であることができる。

【0256】

128ビット命令フォーマット2010は、例えば、直接レジスタアドレッシングモード又は間接レジスタアドレッシングモードが使用されるかどうかを指定するアクセス／アドレスモードフィールド2026を含んでよい。直接レジスタアドレッシングモードが使用される場合に、1つ以上のオペランドのレジスタアドレスは、命令内のビットによって直接に供給される。

【0257】

128ビット命令フォーマット2010はまた、命令のアドレスモード及び／又はアクセスモードを指定するアクセス／アドレスモードフィールド2026を含んでもよい。アクセスモードは、命令についてデータアクセスアライメントを定義するために使用されてよい。16バイトアライメントアクセスモード及び1バイトアライメントアクセスモードを含むアクセスモードがサポートされてよく、アクセスモードのバイトアライメントは、命令オペランドのアクセスアライメントを決定する。例えば、第1モードにある場合に、命令は、ソース及びあて先オペランドのために、バイトアライメントされたアドレッシングを使用してよく、第2モードにある場合に、命令は、全てのソース及びあて先オペランドのために、16バイトアライメントされたアドレッシングを使用してよい。

【0258】

アクセス／アドレスモードフィールド2026のアドレスモード部分は、命令が直接又は間接アドレッシングを使用すべきかどうかを決定してよい。直接レジスタアドレッシングモードが使用される場合に、命令内のビットは、1つ以上のオペランドのレジスタアドレスを直接に供給する。間接レジスタアドレッシングモードが使用される場合に、1つ以上のオペランドのレジスタアドレスは、命令内のアドレス中間フィールド及びアドレスレジスタ値に基づいて計算されてよい。

10

20

30

40

50

【0259】

命令は、オペコードでコード2040を簡単にするようオペコード2012のビットフィールドに基づいてグループ化されてよい。8ビットオペコードについては、ビット4、5及び6は、実行ユニットがオペコードのタイプを決定することを可能にする。示されている正確なオペコードグルーピングは、例にすぎない。移動及び論理オペコードグループ2042は、データ移動及び論理命令（例えば、移動（mov）、比較（cmp））を含んでよい。移動及び論理グループ2042は、5つの最上位ビット（MSB）を共有してよく、移動（mov）命令は、0000xxxxbの形を取り、論理命令は、0001xxxxbの形を取る。フロー制御命令2044（例えば、コール、ジャンプ（jmp））は、0010xxxxb（例えば、0x20）の形を取る命令を含む。その他の命令グループ2046は、0011xxxxb（例えば、0x30）の形を取る同期化命令（例えば、待機、送信）を含む命令の混合を含む。並列演算命令グループ2048は、0100xxxxb（例えば、0x40）の形を取るコンポーネントごとの算術命令（例えば、加算、乗算（mul））を含む。並列演算グループ2048は、データチャネルにわたって並行して算術演算を実行する。ベクトル演算グループ2050は、0101xxxxb（例えば、0x50）の形を取る算術命令（例えば、dp4）を含む。ベクトル演算グループは、ベクトルオペランドに対する内積計算などの算術演算を実行する。表されているオペコード復号化2040は、一実施形態において、実行ユニットのどの部分が、復号された命令を実行するために使用されるか、を決定するために使用され得る。例えば、いくつかの命令は、シストリックアレイによって実行されることになるシストリック命令として指定されてよい。レイトレーシング命令（図示せず）などの他の命令は、実行ロジックのスライス又はパーティション内のレイトレーシングコア又はレイトレーシングロジックヘルパーティングされ得る。

【0260】

[グラフィクスパイプライン]

図21は、他の実施形態に従うグラフィクスプロセッサ2100のブロック図である。本願のいずれかの他の図の要素と同じ又は類似した名称を有する図21の要素は、他の図で見られる同じ要素について記載しており、本願のどこかで記載されているのと同じように動作又は機能することができ、同じコンポーネントを有することができ、他のエンティティへリンクされ得るが、そのように限定されない。

【0261】

グラフィクスプロセッサ2100は、ジオメトリパイプライン2120、メディアパイプライン2130、ディスプレイエンジン2140、スレッド実行ロジック2150、及びレンダー出力パイプライン2170などの種々のタイプのグラフィクスパイプラインパイプラインを含んでよい。グラフィクスプロセッサ2100は、1つ以上の汎用処理コアを含むマルチコア処理システム内のグラフィクスプロセッサであってよい。グラフィクスプロセッサは、リングインターコネクト2102を介してグラフィクスプロセッサ2100へ発行されたコマンドにより、あるいは、1つ以上の制御レジスタ（図示せず）へのレジスタ書き込みによって、制御されてよい。リングインターコネクト2102は、グラフィクスプロセッサ2100を、他のグラフィクスプロセッサ又は汎用プロセッサなどの他の処理コンポーネントへ結合してよい。リングインターコネクト2102からのコマンドは、ジオメトリパイプライン2120又はメディアパイプライン2130の個々のコンポーネントへ命令を供給するコマンドストリーマ2103によって解釈される。

【0262】

コマンドストリーマ2103は、メモリから頂点データを読み出し、コマンドストリーマ2103によって供給された頂点処理コマンドを実行する頂点フェッチャ2105の動作を指示してよい。頂点フェッチャ2105は、頂点データを頂点シェーダ2107へ供給してよく、頂点シェーダ2107は、各頂点に対して座標空間変換及び明暗操作を実行する。頂点フェッチャ2105及び頂点シェーダ2107は、スレッドディスパッチャ2131を介して実行ユニット2152A～2152Bへ実行スレッドをディスパッチする

ことによって、頂点処理命令を実行してよい。

【0263】

実行ユニット2152A～2152Bは、グラフィクス及びメディア操作を実行するための命令セットを有するベクトルプロセッサのアレイであってよい。実行ユニット2152A～2152Bは、各アレイについて特有であるか又はアレイ間で共有される付属のL1キャッシュ2151を有してよい。キャッシュは、異なるパーティションでデータ及び命令を含むようパーティション化されているデータキャッシュ、命令キャッシュ、単一キャッシュとして構成され得る。

【0264】

ジオメトリパイプライン2120は、3Dオブジェクトのハードウェアで加速されたテッセレーションを実行するテッセレーションコンポーネントを含んでよい。プログラム可能なハルシェーダ2111は、テッセレーション演算を構成してよい。プログラム可能なドメインシェーダ2117は、テッセレーション出力のバックエンド評価を提供してよい。テッセレータ2113は、ハルシェーダ2111の指示で動作し、そして、ジオメトリパイプライン2120への入力として供給される粗幾何学モデルに基づいて詳細な幾何学オブジェクトの組を生成する専用ロジックを含んでよい。更には、テッセレーションが使用されない場合には、テッセレーションコンポーネント（例えば、ハルシェーダ2111、テッセレータ2113、及びドメインシェーダ2117）はバイパスされ得る。

【0265】

完全な幾何学オブジェクトは、実行ユニット2152A～2152Bにディスパッチされた1つ以上のスレッドを介してジオメトリシェーダ2119によって処理されてよく、あるいは、クリッパ2129へ直接に進むことができる。ジオメトリシェーダ2119は、グラフィクスパイプラインの前の段で見られたように頂点又は頂点のパッチよりもむしろ、幾何学オブジェクト全体に作用してよい。テッセレーションが無効にされる場合には、ジオメトリシェーダ2119は頂点シェーダ2107から入力を受け取る。ジオメトリシェーダ2119は、テッセレーションユニットが無効にされる場合にジオメトリテッセレーションを実行するようジオメトリシェーダプログラムによってプログラム可能であってよい。

【0266】

ラスタライゼーションの前に、クリッパ2129は頂点データを処理する。クリッパ2129は、クリッピング及びジオメトリシェーダ機能を備えた固定機能クリッパ又はプログラム可能なクリッパであってよい。レンダー出力パイプライン2170内のラスタライザ及びデプステストコンポーネント2173は、幾何学オブジェクトをピクセルごとの表現に変換するようピクセルシェーダをディスパッチしてよい。ピクセルシェーダロジックはスレッド実行ロジック2150に含まれてよい。任意に、アプリケーションは、ラスタライザ及びデプステストコンポーネント2173をバイパスし、ラスタライズされていない頂点データにストリーム出力ユニット2123を介してアクセスすることができる。

【0267】

グラフィクスプロセッサ2100は、インターコネクトバス、インターコネクトファブリック、又はプロセッサの主なコンポーネントの間のデータ及びメッセージ受け渡しを可能にするその他のインターコネクトメカニズムを有する。いくつかの実施形態において、実行ユニット2152A～2152B及び関連するロジックユニット（例えば、L1キャッシュ2151、サンブラ2154、テクスチャキャッシュ2158、など）は、メモリアクセスを実行しかつプロセッサのレンダー出力パイプラインコンポーネントと通信するようデータポート2156を介して相互接続する。サンブラ2154、キャッシュ2151、2158、及び実行ユニット2152A～2152Bは夫々、別々のメモリアクセスパスを有してよい。任意に、テクスチャキャッシュ2158はまた、サンブラキャッシュとして構成され得る。

【0268】

レンダー出力パイプライン2170は、頂点に基づくオブジェクトに関連するピクセル

10

20

30

40

50

に基づく表現に変換するラスタイラザ及びデプステストコンポーネント 2 1 7 3 を含んでよい。ラスタイラザロジックは、固定機能三角形及び直線ラスタイラゼーションを実行するウィンドウア (windower) / マスカ (masker) ユニットを含んでよい。関連するレンダーキャッシュ 2 1 7 8 及びデプスキャッシュ 2 1 7 9 も、いくつかの実施形態で利用可能である。ピクセル操作コンポーネント 2 1 7 7 は、データに対してピクセルに基づいた操作を実行するが、いくつかのインスタンスでは、2 D 操作 (例えば、混合 (blending) を伴ったビットブロック画像転送) に関連したピクセル操作は、2 D エンジン 2 1 4 1 によって実行されるか、あるいは、オーバーレイ表示面を用いてディスプレイコントローラ 2 1 4 3 によって表示時点で置換される。共有 L 3 キャッシュ 2 1 7 5 は、全てのグラフィクスコンポーネントに利用可能であってよく、主システムメモリの使用なしでデータの共有を可能にする。

10

【0 2 6 9】

グラフィクスプロセッサメディアパイプライン 2 1 3 0 は、メディアエンジン 2 1 3 7 及びビデオフロントエンド 2 1 3 4 を含んでよい。ビデオフロントエンド 2 1 3 4 は、コマンドストリーマ 2 1 0 3 からパイプラインコマンドを受け取ってよい。メディアパイプライン 2 1 3 0 は、別個のコマンドストリーマを含んでよい。ビデオフロントエンド 2 1 3 4 は、コマンドをメディアエンジン 2 1 3 7 へ送る前に、メディアコマンドを処理してよい。メディアエンジン 2 1 3 7 は、スレッドディスパッチャ 2 1 3 1 によるスレッド実行ロジック 2 1 5 0 へのディスパッチのためにスレッドを生成するようスレッド生成機能を含んでよい。

20

【0 2 7 0】

グラフィクスプロセッサ 2 1 0 0 は、ディスプレイエンジン 2 1 4 0 を含んでよい。ディスプレイエンジン 2 1 4 0 は、プロセッサ 2 1 0 0 の外にあってよく、リングインターコネクト 2 1 0 2 又はその他のインターコネクトバス若しくはファブリックを介してグラフィクスプロセッサと結合してよい。ディスプレイエンジン 2 1 4 0 は、2 D エンジン 2 1 4 1 及びディスプレイコントローラ 2 1 4 3 を含んでよい。ディスプレイエンジン 2 1 4 0 は、3 D パイプラインから独立して動作することが可能な専用ロジックを含んでよい。ディスプレイコントローラ 2 1 4 3 は、表示デバイス (図示せず) と結合してよく、表示デバイスは、ラップトップコンピュータで見られるように、システム内蔵型表示デバイス、又は表示デバイスコネクタを介して取り付けられた外付け表示デバイスであってよい。

30

【0 2 7 1】

ジオメトリパイプライン 2 1 2 0 及びメディアパイプライン 2 1 3 0 は、複数のグラフィクス及びメディアプログラミングインターフェースに基づいて動作を実行するよう構成可能であってよく、いずれか 1 つのアプリケーションプログラミングインターフェース (API) に特有ではない。グラフィクスプロセッサ用のドライバソフトウェアは、特定のグラフィクス又はメディアライブラリに特有である API コールを、グラフィクスプロセッサによって処理され得るコマンドに変換してよい。全てクロノスグループからであるオープン・グラフィクス・ライブラリ (Open Graphics Library, Open GL)、オープン・コンピューティング・ランゲージ (Open Computing Language, Open CL)、及び / 又は Vulkan グラフィクス及び計算 API のために、サポートが提供されてもよい。マイクロソフト・コーポレーションからの Direct 3 D ライブラリのためにも、サポートが提供されてよい。これらのライブラリの組み合わせがサポートされてもよい。オープン・ソース・コンピュータ・ビジョン・ライブラリ (Open Source Computer Vision Library, Open CV) のためにも、サポートが提供されてよい。互換性のある 3 D パイプラインを備えた将来の API も、マッピングが将来の API のパイプラインからグラフィクスプロセッサのパイプラインへ行われ得る場合は、サポートされることになる。

40

【0 2 7 2】

[グラフィクスパイプラインプログラミング]

図 2 2 A は、例えば、図 1 6 A、1 7、2 1 とともに本願で記載されているパイプライン

50

ンなどのグラフィクス処理パイプラインをプログラムするために使用されるグラフィクスプロセッサコマンドフォーマット2200を表すブロック図である。図22Bは、実施形態に従うグラフィクスプロセッサコマンドシーケンス2210を表すブロック図である。図22Aの実線ボックスは、一般的にグラフィクスコマンドに含まれているコンポーネントを表し、破線は、任意であるか、あるいは、グラフィクスコマンドのサブセットにしか含まれないコンポーネントを含む。図22Aの例となるグラフィクスプロセッサコマンドフォーマット2200は、クライアント2202、コマンド操作コード（オペコード）2204、及びコマンドのためのデータ2206を識別するデータフィールドを含む。サブオペコード2205及びコマンドサイズ2208も、いくつかのコマンドには含まれる。

【0273】

クライアント2202は、コマンドデータを処理するグラフィクスデバイスのクライアントユニットを指定してよい。グラフィクスプロセッサコマンドパーサは、コマンドの更なる処理を条件付け、コマンドデータを適切なクライアントユニットへ送るよう、各コマンドのクライアントフィールドを調べてよい。グラフィクスプロセッサクライアントユニットは、メモリインターフェースユニット、レンダーユニット、2Dユニット、3Dユニット、及びメディアユニットを含んでよい。各クライアントユニットは、コマンドを処理する対応する処理パイプラインを有してよい。コマンドがクライアントユニットによって受け取られると、クライアントユニットは、オペコード2204、及び存在する場合には、サブオペコード2205を読み出して、実行すべき動作を決定する。クライアントユニットは、データフィールド2206内の情報を用いてコマンドを実行する。いくつかのコマンドについては、明示的なコマンドサイズ2208が、コマンドのサイズを特定すると期待される。コマンドパーサは、コマンドオペコードに基づいてコマンドの少なくとも一部のサイズを自動的に決定してよい。コマンドは、ダブルワードの倍数によりアライメントされてよい。他のコマンドフォーマットも使用可能である。

【0274】

図22Bのフロー図は、例となるグラフィクスプロセッサコマンドシーケンス2210を表す。例示的なグラフィクスプロセッサを特徴付けるデータ処理システムのソフトウェア又はファームウェアは、グラフィクス操作の組をセットアップ、実行、及び終了することが示されたコマンドシーケンスの変形を使用してよい。サンプルコマンドシーケンスは、例示のみを目的として図示及び記載されており、これらの具体的なコマンドに又はこのコマンドシーケンスに限定されない。更に、コマンドは、グラフィクスプロセッサが少なくとも部分的に同時にコマンドのシーケンスを処理するように、コマンドシーケンスの中のコマンドのバッチとして発行されてよい。

【0275】

グラフィクスプロセッサコマンドシーケンス2210は、いずれかのアクティブなグラフィクスパイプラインにそのパイプラインの目下ペンディング中のコマンドを完了させるよう、パイプラインフラッシュコマンド2212から始まってよい。任意に、3Dパイプライン2222及びメディアパイプライン2224は、同時に動作しなくてもよい。パイプラインフラッシュは、アクティブなグラフィクスパイプラインに如何なるペンディング中のコマンドも完了させるよう実行される。パイプラインフラッシュに応答して、グラフィクスプロセッサのコマンドパーサは、アクティブな描画エンジンがペンディング中の動作を完了し、関連する読み取りキャッシュが無効にされるまで、コマンド処理を一時停止する。任意に、「ダーティ」（dirty）とマークされているレンダーキャッシュ内の如何なるデータも、メモリにフラッシュされ得る。パイプラインフラッシュコマンド2212は、グラフィクスプロセッサを低電力状態に置く前に、又はパイプライン同期化のために、使用され得る。

【0276】

パイプライン選択コマンド2213は、コマンドシーケンスがグラフィクスプロセッサにパイプライン間の明示的な切り替えを求める場合に、使用されてよい。パイプライン選択コマンド2213は、コンテキストが両方のパイプラインについてコマンドを発行すべ

10

20

30

40

50

きでない限り、パイプラインコマンドを発行する前に実行コンテキスト内で一度しか必要とされなくてよい。パイプラインフラッシュコマンド 2 2 1 2 は、パイプライン選択コマンド 2 2 1 3 によるパイプライン切り替えの直前に必要とされてよい。

【0 2 7 7】

パイプライン制御コマンド 2 2 1 4 は、動作のためにグラフィクスパイプラインを設定してよく、そして、3 D パイプライン 2 2 2 2 及びメディアパイプライン 2 2 2 4 をプログラムするために使用されてよい。パイプライン制御コマンド 2 2 1 4 は、アクティブなパイプラインについてパイプライン状態を設定してよい。パイプライン制御コマンド 2 2 1 4 は、パイプライン同期化のために、かつ、コマンドのバッチを処理する前にアクティブなパイプライン内の 1 つ以上のキャッシュメモリからデータをクリアするために、使用

10

【0 2 7 8】

リターンバッファ状態コマンド 2 2 1 6 は、各々のパイプラインがデータを書き込むためにリターンバッファの組を設定するために使用されてよい。いくつかのパイプライン動作は、動作が処理中に中間データを書き込む 1 つ以上のリターンバッファの割り当て、選択、又は設定を必要とするグラフィクスプロセッサはまた、出力データを記憶するために及びクロススレッド通信を実行するために 1 つ以上のリターンバッファを使用してよい。リターンバッファ状態 2 2 1 6 は、パイプライン動作の組に使用すべきリターンバッファのサイズ及び数を選択することを含んでよい。

20

【0 2 7 9】

コマンドシーケンス内の残りのコマンドは、動作のためのアクティブなパイプラインに基づいて異なる。パイプライン決定 2 2 2 0 に基づいて、コマンドシーケンスは、3 D パイプライン状態 2 2 3 0 から開始する 3 D パイプライン 2 2 2 2、又はメディアパイプライン状態 2 2 4 0 で始まるメディアパイプライン 2 2 2 4 に合わせられる。

【0 2 8 0】

3 D パイプライン状態 2 2 3 0 を設定するコマンドは、3 D プリミティブコマンドが処理される前に設定されるべきである頂点バッファ状態、頂点要素状態、一定色状態、デプスバッファ状態、及び他の状態変数に対する 3 D 状態設定コマンドを含む。これらのコマンドの値は、使用中の特定の 3 D A P I に少なくとも部分的に基づいて決定される。3 D パイプライン状態 2 2 3 0 コマンドはまた、特定のパイプライン要素を、それらの要素が使用されない場合に選択的に無効化又はバイパスすることも可能であり得る。

30

【0 2 8 1】

3 D プリミティブ 2 2 3 2 コマンドは、3 D パイプラインによって処理されるべき 3 D プリミティブをサブミットするために使用されてよい。3 D プリミティブ 2 2 3 2 コマンドによりグラフィクスプロセッサへ渡されるコマンド及び関連するパラメータは、グラフィクスパイプライン内の頂点フェッチ機能へ転送される。頂点フェッチ機能は、3 D プリミティブ 2 2 3 2 コマンドデータを使用して、頂点データ構造を生成する。頂点データ構造は、1 つ以上のリターンバッファに格納される。3 D プリミティブ 2 2 3 2 コマンドは、頂点シェーダにより 3 D プリミティブに対して頂点操作を実行するために使用されてよい。頂点シェーダを処理するために、3 D パイプライン 2 2 2 2 は、グラフィクスプロセッサ実行ユニットへシェーダ実行スレッドをディスパッチする。

40

【0 2 8 2】

3 D パイプライン 2 2 2 2 は、実行 2 2 3 4 コマンド又はイベントによりトリガされてよい。レジスタは、トリガコマンド実行を書き込んでよい。実行は、コマンドシーケンス内の「ゴー」(go)又は「キック」(kick)コマンドによりトリガされてよい。コマンド実行は、グラフィクスパイプラインを通じてコマンドシーケンスをフラッシュするようパイプライン同期化コマンドを用いてトリガされてよい。3 D パイプラインは、3 D プリミティブに対するジオメトリ処理を実行する。動作が完了すると、結果として得られた幾何学オブジェクトはラスタライズされ、ピクセルエンジンは、結果として得られたピクセルを彩色する。ピクセルシェーディング及びピクセルバックエンド動作を制御するための追

50

加コマンドも、これらの動作のために含まれてよい。

【0283】

グラフィクスプロセッサコマンドシーケンス2210は、メディア操作を実行する場合にメディアパイプライン2224パスを辿ってよい。一般に、メディアパイプライン2224のためのプログラミングの具体的な使用及び方法は、実行されるべきメディア又は計算操作に依存する。特定のメディア復号化動作が、メディア復号化中にメディアパイプラインにオフロードされてよい。メディアパイプラインはまた、バイパスされてもよく、メディア復号化は、1つ以上の汎用処理コアによって供給されたリソースを用いて全体的に又は部分的に実行され得る。メディアパイプラインはまた、汎用グラフィクスプロセッサユニット（GPGPU）動作のための要素を含んでよく、グラフィクスプロセッサは、グラフィクスプリミティブのレンダリングに明示的には関係がない計算シェーダプログラムを用いてSIMDベクトル演算を実行するために使用される。

10

【0284】

メディアパイプライン2224は、3Dパイプライン2222と同じように構成されてよい。メディアパイプライン状態2240を設定するためのコマンドの組は、メディアオブジェクトコマンド2242の前にコマンドキューにディスパッチ又は挿入される。メディアパイプライン状態2240のためのコマンドは、メディアオブジェクトを処理するために使用されるメディアパイプライン要素を構成するデータを含んでよい。これは、符号化又は復号化フォーマットなどの、メディアパイプライン内のビデオ復号化及びビデオ符号化ロジックを構成するデータを含む。メディアパイプライン状態2240のためのコマンドはまた、状態設定のバッチを含む「間接的な」状態要素への1つ以上のポインタの使用もサポートしてもよい。

20

【0285】

メディアオブジェクトコマンド2242は、メディアパイプラインによる処理のためにメディアオブジェクトへのポインタを供給してよい。メディアオブジェクトは、処理されるべきビデオデータを含むメモリバッファを含む。任意に、全てのメディアパイプライン状態は、メディアオブジェクトコマンド2242を発行する前に有効でなければならない。パイプライン状態が設定され、メディアオブジェクトコマンド2242がキューされると、メディアパイプライン2224は、実行コマンド2244又は同等の実行イベント（例えば、レジスタ書き込み）によりトリガされる。メディアパイプライン2224からの出力は、次いで、3Dパイプライン2222又はメディアパイプライン2224によって供給された演算によって後処理されてよい。GPGPU動作は、メディア操作と同様に設定及び実行されてよい。

30

【0286】

〔グラフィクスソフトウェアアーキテクチャ〕

図23は、データ処理システム2300についての例示的なグラフィクスソフトウェアアーキテクチャを表す。そのようなソフトウェアアーキテクチャは、3Dグラフィクスアプリケーション2310、オペレーティングシステム2320、及び少なくとも1つのプロセッサ2330を含んでよい。プロセッサ2330は、グラフィクスプロセッサ2332及び1つ以上の汎用プロセッサコア2334を含む。プロセッサ2330は、プロセッサ1402又は本願で記載されているプロセッサのいずれか他の変形であってよい。プロセッサ2330は、プロセッサ1402又は本願で記載されているプロセッサのいずれか他の代わりに使用されてよい。従って、プロセッサ1420又は本願で記載されているプロセッサのいずれか他と組み合わせられた如何なる特徴の開示も、グラフィクスプロセッサ2330との対応する組み合わせを開示するが、そのように限定されない。更に、本願のいずれか他の図の要素と同じ又は類似した名称を有する図23の要素は、他の図で見られる同じ要素について記載しており、本願のどこかで記載されているのと同じように動作又は機能することができ、同じコンポーネントを有することができ、他のエンティティへリンクされ得るが、そのように限定されない。グラフィクスアプリケーション2310及びオペレーティングシステム2320は夫々、データ処理システムのシステムメモリ235

40

50

0で実行される。

【0287】

3Dグラフィクスアプリケーション2310は、シェーダ命令2312を含む1つ以上のシェーダプログラムを含んでよい。シェーダ言語命令は、Direct3Dの上位シェーダ言語(High-Level Shader Language, HLSL)、OpenGLシェーダ言語(GLSL)、などのような上位シェーダ言語にあってよい。アプリケーションはまた、汎用プロセッサコア2334による実行に適した機械言語で実行可能命令2314を含んでもよい。アプリケーションはまた、頂点データによって定義されたグラフィクスオブジェクト2316を含んでもよい。

【0288】

オペレーティングシステム2320は、マイクロソフト・コーポレーションからのMicrosoft(登録商標) Windows(登録商標)オペレーティングシステム、独自仕様のUnix様のオペレーティングシステム、又はLinux(登録商標)カーネルの変形を使用するオープンソースのUNIX(登録商標)様のオペレーティングシステムであってよい。オペレーティングシステム2320は、Direct3D API、OpenGL API、又はVulkan APIなどのグラフィクスAPI2322をサポートすることができる。Direct3D APIが使用中であるとき、オペレーティングシステム2320は、HLSLでの如何なるシェーダ命令2312も下位シェーダ言語にコンパイルするためにフロントエンドシェーダコンパイラ2324を使用する。コンパイルは、ジャスト・イン・タイム(JIT)コンパイルであってよく、あるいは、アプリケーションは、シェーダ・プリコンパイルを実行することができる。上位シェーダは、3Dグラフィクスアプリケーション2310のコンパイル中に下位シェーダにコンパイルされてよい。シェーダ命令2312は、Vulkan APIによって使用されるSPIR(Standard Portable Intermediate Representation)の変形などの中間形で供給されてよい。

【0289】

ユーザモードグラフィクスドライバ2326は、シェーダ命令2312をハードウェア特有の表現に変換するようバックエンドシェーダコンパイラ2327を含んでよい。OpenGL APIが使用中であるとき、GLSL上位言語でのシェーダ命令2312は、コンパイルのためにユーザモードグラフィクスドライバ2326へ渡される。ユーザモードグラフィクスドライバ2326は、カーネルモードグラフィクスドライバ2329と通信するためにオペレーティングシステムカーネルモード機能2328を使用してよい。カーネルモードグラフィクスドライバ2329は、コマンド及び命令をディスパッチするためにグラフィクスプロセッサ2332と通信してよい。

【0290】

[IPコア実装]

1つ以上の態様は、プロセッサなどの集積回路内のロジックを表現及び／又は定義する機械可読媒体に記憶された代表コードによって実装されてもよい。例えば、機械可読媒体は、プロセッサ内の様々なロジックを表す命令を含んでよい。機械によって読み出される場合に、命令は、機械に、本願で記載されている技術を実行するようロジックを組み立てさせ得る。「IPコア」として知られているそのような表現は、集積回路の構造を記述するハードウェアモデルとして有形な機械可読媒体に記憶され得る集積回路用ロジックの再利用可能なユニットである。ハードウェアモデルは、集積回路を製造する組立機械にハードウェアモデルをロードする様々なカスタマ又は製造設備に供給されてよい。集積回路は、回路が、本願で記載されている実施形態のいずれかと関連して記載されている動作を実行するように、組み立てられてよい。

【0291】

図24Aは、実施形態に従う動作を実行するよう集積回路を製造するために使用され得るIPコア開発システム2400を表すブロック図である。IPコア開発システム2400は、より大きい設計に組み込まれ得るモジュール式の再利用可能な設計を生成するため

10

20

30

40

50

に使用されるか、あるいは、集積回路（例えば、SOC集積回路）全体を構成するために使用されてよい。設計機能2430は、上位プログラミング言語（例えば、C/C++）でIPコア設計のソフトウェアシミュレーション2410を生成することができる。ソフトウェアシミュレーション2410は、シミュレーションモデル2412を用いてIPコアの挙動を設計、試験、及び検証するために使用され得る。シミュレーションモデル2412は、機能、挙動、及び／又はタイミングシミュレーションを含んでよい。レジスタ転送レベル（register transfer level, RTL）設計2415は、モデル化されたデジタル信号を用いて実行される関連ロジックを含め、ハードウェアレジスタ間のデジタル信号のフローをモデル化する集積回路の挙動の抽象化である。RTL設計2415に加えて、ロジックレベル又はトランジスタレベルでの下位設計も、生成、設計又は合成されてよい。よって、初期設計及びシミュレーションの具体的な詳細は様々であってよい。

10

【0292】

RTL設計2415又は同等物は、設計機能によって、ハードウェア記述言語（HDL）又は物理設計データのその他表現であってよいハードウェアモデルに更に合成されてもよい。HDLは、IPコア設計を検証するよう更にシミュレーション又はテストされてもよい。IPコア設計は、不揮発性メモリ2440（例えば、ハードディスク、フラッシュメモリ、又は任意の不揮発性記憶媒体）を用いて第三者の組立設備2465への配送のために記憶され得る。代替的に、IPコア設計は、有線接続2450又は無線接続2460を介して（例えば、インターネット経由で）伝送されてもよい。組立設備2465は、次いで、少なくとも部分的にIPコア設計に基づく集積回路を組み立ててよい。組み立てられた集積回路は、本願で記載されている少なくとも1つの実施形態に従う動作を実行するよう構成され得る。

20

【0293】

図24Bは、集積回路パッケージアセンブリ2470の側面断面図を表す。集積回路パッケージアセンブリ2470は、本願で記載されている1つ以上のプロセッサ又はアクセラレータデバイスの実装を表す。パッケージアセンブリ2470は、基板2480へ接続されたハードウェアロジックの複数のユニット2672、2674を含む。ロジック2672、2674は、設定可能なロジック又は固定機能ロジックハードウェアにおいて少なくとも部分的に実装されてよく、プロセッサコア、グラフィクスプロセッサ、又は本願で記載されている他のアクセラレータデバイスのいずれかの1つ以上の部分を含むことができる。ロジック2672、2674の各ユニットは、半導体ダイの中に実装され、インターコネクト構造2473を介して基板2480と結合され得る。インターコネクト構造2473は、ロジック2672、2674と基板2480との間で電気信号を伝えるよう構成されてよく、バンプ又はピラーなどの、しかしこれらに限られないインターコネクトを含むことができる。インターコネクト構造2473は、例えば、ロジック2672、2674の動作に関連した入出力（I/O）信号及び／又は電力若しくは接地信号などの電気信号を伝えるよう構成されてよい。任意に、基板2480は、エポキシベースの積層基板であってよい。基板2480は、他の適切なタイプの基板を含んでもよい。パッケージアセンブリ2470は、パッケージインターコネクト2483を介して他の電気デバイスへ接続され得る。パッケージインターコネクト2483は、マザーボード、他のチップセット、又はマルチチップモジュールなどの他の電気デバイスへ電気信号を伝えるよう基板2480の表面へ結合されてよい。

30

40

【0294】

ロジック2672、2674のユニットは、ロジック2672、2674の間で電気信号を伝えるよう構成されるブリッジ2482と電氣的に結合されてよい。ブリッジ2482は、電気信号のルートを提供する密なインターコネクト構造であってよい。ブリッジ2482は、ガラス又は適切な半導体材料から成るブリッジ基板を含んでよい。電気ルーティング構造が、ロジック2672、2674の間のチップ間接続を提供するようブリッジ基板上に形成され得る。

【0295】

50

ロジックの2つのユニット2672、2674及びブリッジ2482が表されているが、本願で記載されている実施形態は、1つ以上のダイ上でより多い又はより少ないロジックユニットを含んでよい。1つ以上のダイは、ロジックが単一のダイ上に含まれる場合にブリッジ2482は除かれてもよいということで、ゼロか又はより多いブリッジによって接続されてもよい。代替的に、複数のダイ又はロジックのユニットが1つ以上のブリッジによって接続され得る。更には、複数のロジックユニット、ダイ、及びブリッジは、3次元構成を含む他の可能な構成で連結され得る。

【0296】

図24Cは、基板2480（例えば、ベースダイ）へ接続されたハードウェアロジックチップレットの複数のユニットを含むパッケージアセンブリ2490を表す。本願で記載されているグラフィクス処理ユニット、並列プロセッサ、及び計算アクセラレータは、別々に製造される多種多様なシリコンチップレットから構成され得る。これに関連して、チップレットは、他のチップレットとともにより大きいデバイス内に組み立てられ得る相異なるロジックユニットを含む少なくとも部分的にパッケージ化された集積回路である。異なるIPコアロジックを備えたチップレットの多種多様な組が単一のデバイス内に組み立てられ得る。更には、チップレットは、アクティブインターポーザ（active interposer）技術を用いてベースダイ又はベースチップレットに集積され得る。本願で記載されている概念は、GPU内の異なる形式のIP感の相互接続及び通信を可能にする。IPコアは、異なるプロセス技術を用いて製造され、製造中に組み立てられ得る。これは、特に、いくつかのプレーヤーIPを備えた大規模SoCで、複数のIPを同じプロセスに収束させる複雑さを回避する。複数のプロセス技術の使用を可能にすることは、市場投入までの時間を改善し、かつ、複数の製品SKUを作成するためのコスト効果の高い方法をもたらす。更には、分解されたIPは、より、独立してパワーゲーティングされやすくなり、所与のワークロードで使用されていないコンポーネントの電源をオフにして、全体の電力消費を削減することができる。

【0297】

ハードウェアロジックチップレットは、専用ハードウェアロジックチップレット2472、ロジック又はI/Oチップレット2474、及び／又はメモリチップレット2475を含むことができる。ハードウェアロジックチップレット2472及びロジック又はI/Oチップレット2474は、設定可能なロジック又は固定機能ロジックハードウェアにおいて少なくとも部分的に実装されてよく、本願で記載されているプロセッサコア、グラフィクスプロセッサ、並列プロセッサ、又は他のアクセラレータデバイスのいずれかの1つ以上の部分を含むことができる。メモリチップレット2475は、DRAM（例えば、GDDR、HBM）メモリ又はキャッシュ（SRAM）メモリであることができる。

【0298】

各チップレットは、別個の半導体ダイとして製造され、インターコネクト構造2473を介して基板2480と結合され得る。インターコネクト構造2473は、基板2480内の様々なチップレット及びロジック間で電気信号を伝えるよう構成されてよい。インターコネクト構造2473は、例えば、バンプ又はピラーなどの、しかしこれらに限られないインターコネクトを含むことができる。いくつかの実施形態において、インターコネクト構造2473は、例えば、ロジック、I/O及びメモリチップレットの動作に関連した入出力（I/O）信号及び／又は電力若しくは接地信号などの電気信号を伝送するよう構成されてよい。

【0299】

基板2480は、エポキシベースの積層基板であってよいが、それに限定されず、基板2480はまた、他の適切なタイプの基板を含んでもよい。パッケージアセンブリ2490は、パッケージインターコネクト2483を介して他の電気デバイスへ接続され得る。パッケージインターコネクト2483は、マザーボード、他のチップセット、又はマルチチップモジュールなどの他の電気デバイスへ電気信号を伝えるよう基板2480の表面へ結合されてよい。

【0300】

ロジック又はI/Oチップレット2474及びメモリチップレット2475は、ロジック又はI/Oチップレット2474とメモリチップレット2475との間で電気信号を伝えるよう構成されるブリッジ2487を介して電氣的に結合されてよい。ブリッジ2487は、電気信号のためのルートを提供する密なインターコネクト構造であってよい。ブリッジ2487は、ガラス又は適切な半導体材料から成るブリッジ構造を含んでよい。電気ルーティング機能は、ロジック又はI/Oチップレット2474とメモリチップレット2475との間のチップ間接続をもたらすようブリッジ構造で形成され得る。ブリッジ2487はまた、シリコンブリッジ又はインターコネクトブリッジとも呼ばれ得る。例えば、ブリッジ2487は、埋め込みマルチダイインターコネクトブリッジ (Embedded Multi-die Interconnect Bridge, EMIB) である。代替的に、ブリッジ2487は、単に、1つのチップレットから他のチップレットへの直接接続であってもよい。

10

【0301】

基板2480は、I/O2491、キャッシュメモリ2492、及び他のハードウェアロジック2493のためのハードウェアコンポーネントを含むことができる。ファブリック2485は、様々なロジックチップレットと基板2480内のロジック2491、2493との間の通信を可能にするよう基板2480に埋め込まれ得る。任意に、I/O2491、ファブリック2485、キャッシュ、ブリッジ、及び他のハードウェアロジック2493は、基板2480の上に積層されているベースダイに集積され得る。

【0302】

更に、パッケージアセンブリ2490はまた、ファブリック2485又は1つ以上のブリッジ2487によって相互接続されるより少ない又はより多いコンポーネント及びチップレットを含むことができる。パッケージアセンブリ2490内のチップレットは、3D又は2.5D配置で配置されてよい。一般に、ブリッジ構造2487は、例えば、ロジック又はI/Oチップレットとメモリチップレットとの間のポイント・ツー・ポイントのインターコネクトを促進するために使用されてよい。ファブリック2485は、様々なロジック及び／又はI/Oチップレット (例えば、チップレット2472、2474、2491、2493) を他のロジック及び／又はI/Oチップレットと相互接続するために使用され得る。基板内のキャッシュメモリ2492は、パッケージアセンブリ2490のためのグローバルキャッシュとして、又は分散グローバルキャッシュの部分として、又はファブリック2485の専用のキャッシュとして動作することができる。

20

30

【0303】

図24Dは、実施形態に従って、交換可能なチップレット2495を含むパッケージアセンブリ2494を表す。交換可能なチップレット2495は、1つ以上のベースチップレット2496、2498上の標準化されたスロット内に組み立てられ得る。ベースチップレット2496、2498は、本願で記載されている他のブリッジインターコネクトと同様であることができ、例えば、EMIBであってよいブリッジインターコネクト2497を介して結合され得る。メモリチップレットはまた、ブリッジインターコネクトを介してロジック又はI/Oチップレットへ接続され得る。I/O及びロジックチップレットは、インターコネクトファブリックを介して通信することができる。ベースチップレットは夫々、ロジック又はI/O又はメモリ／キャッシュの1つのための標準化されたフォーマットで1つ以上のスロットをサポートすることができる。

40

【0304】

SRAM及び配電回路は、ベースチップレットの上にスタックされる交換可能なチップレット2495に対して異なったプロセス技術を用いて製造され得るベースチップレット2496、2498の1つ以上の中に製造されてよい。例えば、ベースチップレット2496、2498は、より大きいプロセス技術を用いて製造され得る一方で、交換可能なチップレットは、より小さいプロセス技術を用いて製造され得る。交換可能なチップレット2495の中の1つ以上は、メモリ (例えば、DRAM) チップレットであってよい。異なるメモリ密度が、パッケージアセンブリ2494を使用する製品の対象とされる電力及

50

び／又は性能に基づいて、パッケージアセンブリ 2 4 9 4 について選択され得る。更には、異なる数のタイプの機能ユニットを備えたロジックチップレットが、製品の対象とする電力及び／又は性能に基づいて組み立て時に選択され得る。更には、異なるタイプの IP ロジックコアを含むチップレットが、交換可能なチップレットスロット内に挿入可能であり、異なる技術の IP ブロックを混合し適合させることができるハイブリッドプロセッサ設計を可能にする。

【0305】

〔例となるシステム・オン・ア・チップ集積回路〕

図 2 5 ～ 2 6 は、1 つ以上の IP コアを使用して組み立てられ得る例示的な集積回路及び関連するグラフィクスプロセッサを表す。表されているものに加えて、追加のグラフィクスプロセッサ／コア、プロセッサ／コア、ペリフェラル・インターフェース・コントローラ、又は汎用プロセッサコアを含む他のロジック及び回路が含まれてもよい。本願のいずれか他の図の要素と同じ又は類似した名称を有する図 2 5 ～ 2 6 の要素は、他の図で見られる同じ要素について記載しており、本願のどこかで記載されているのと同じように動作又は機能することができ、同じコンポーネントを有することができ、他のエンティティへリンクされ得るが、そのように限定されない。

【0306】

図 2 5 は、1 つ以上の IP コアを使用して組み立てられ得る例示的なシステム・オン・ア・チップ集積回路 2 5 0 0 を表すブロック図である。例となる集積回路 2 5 0 0 は、グラフィクスプロセッサ 1 4 0 8、1 5 0 8、2 5 1 0 の、又は本願で記載されているいずれかのグラフィクスプロセッサの変形であってよく、記載されているいずれかのグラフィクスプロセッサの代わりに使用されてよい、1 つ以上のアプリケーションプロセッサ 2 5 0 5（例えば、CPU）や、少なくとも 1 つのグラフィクスプロセッサ 2 5 1 0 を含む。従って、本願でのグラフィクスプロセッサと組み合わせられる如何なる特徴の開示も、グラフィクスプロセッサ 2 5 1 0 との対応する組み合わせを開示するが、そのように限定されない。集積回路 2 5 0 0 は、画像プロセッサ 2 5 1 5 及び／又はビデオプロセッサ 2 5 2 0 を更に含んでもよく、これらのいずれは、同じ又は複数の異なった設計設備からのモジュール IP コアであってよい。集積回路 2 5 0 0 は、USB コントローラ 2 5 2 5、UART コントローラ 2 5 3 0、SPI／SDIO コントローラ 2 5 3 5、及び I²S／I²C コントローラ 2 5 4 0 を含むペリフェラル又はバスロジックを含んでよい。更には、集積回路 2 5 0 0 は、ハイ・デフィニション・マルチメディア・インターフェース（HDMI（登録商標））コントローラ 2 5 5 0 及びモバイル・インダストリ・プロセッサ・インターフェース（MIPI）表示インターフェース 2 5 5 5 の 1 つ以上へ結合される表示デバイス 2 5 4 5 を含むことができる。ストレージは、フラッシュメモリ及びフラッシュメモリコントローラを含むフラッシュメモリサブシステム 2 5 6 0 によって提供されてよい。メモリインターフェースは、SCRAM 又は SRAM メモリデバイスへのアクセスのためにメモリコントローラ 2 5 6 5 を介して提供されてよい。いくつかの集積回路は、埋め込みセキュリティエンジン 2 5 7 0 を更に含む。

【0307】

図 2 6 A ～ 2 6 B は、本願で記載されている実施形態に従って、SoC 内で使用される例示的なグラフィクスプロセッサを表すブロック図である。グラフィクスプロセッサは、グラフィクスプロセッサ 1 4 0 8、1 5 0 8、2 5 1 0 の、又は本願で記載されているいずれかのグラフィクスプロセッサの変形であってよい。グラフィクスプロセッサは、グラフィクスプロセッサ 1 4 0 8、1 5 0 8、2 5 1 0、又は本願で記載されているいずれかのグラフィクスプロセッサの代わりに使用されてよい。従って、グラフィクスプロセッサ 1 4 0 8、1 5 0 8、2 5 1 0、又は本願で記載されているいずれかのグラフィクスプロセッサと組み合わせられる如何なる特徴の開示も、図 2 6 A ～ 2 6 B のグラフィクスプロセッサとの対応する組み合わせを開示するが、そのように限定されない。図 2 6 A は、実施形態に従って、1 つ以上の IP コアを使用して組み立てられ得る SoC 集積回路の例示的なグラフィクスプロセッサ 2 6 1 0 を表す。図 2 6 B は、実施形態に従って、1 つ以上の

10

20

30

40

50

IPコアを使用して組み立てられ得るSoC集積回路の更なる例示的なグラフィクスプロセッサ2640を表す。図26Aのグラフィクスプロセッサ2610は、低電力グラフィクスプロセッサコアの例である。図26Bのグラフィクスプロセッサ2640は、より高性能のグラフィクスプロセッサコアの例である。例えば、グラフィクスプロセッサ2610、2640の夫々は、本段落の最初に述べられているように、図25のグラフィクスプロセッサ2510の変形であることができる。

【0308】

図26Aに示されるように、グラフィクスプロセッサ2610は、頂点プロセッサ2605及び1つ以上のフラグメントプロセッサ2615A~2615N（例えば、2615A、2615B、2615C、2615d、乃至2615N-1、及び2615N）を含む。グラフィクスプロセッサ2610は、頂点プロセッサ2605が頂点シェーダプログラムの動作を実行するよう最適化されるように、別個のロジックを介して異なったシェーダプログラムを実行することができ、一方、1つ以上のフラグメントプロセッサ2615A~2615Nは、フラグメント又はピクセルシェーダプログラムのフラグメント（例えば、ピクセル）シェーディング動作を実行する。頂点プロセッサ2605は、3Dグラフィクスパイプラインの頂点処理段を実行し、プリミティブ及び頂点データを生成する。フラグメントプロセッサ2615A~2615Nは、表示デバイスで表示されるフレームバッファを生成するために、頂点プロセッサ2605によって生成されたプリミティブ及び頂点データを使用する。フラグメントプロセッサ2615A~2615Nは、Direct3D APIで提供されるようなピクセルシェーダプログラムと同じような動作を実行するために使用され得るOpenGL APIで提供されるようなフラグメントシェーダプログラムを実行するよう最適化されてよい。

【0309】

グラフィクスプロセッサ2610は、1つ以上のメモリ管理ユニット(MMU)2620A~2620B、キャッシュ2625A~2625B、及び回路インターコネクト2630A~2630Bを更に含む。1つ以上のMMU2620A~2620Bは、1つ以上のキャッシュ2625A~2625Bに記憶されている頂点又は画像／テクスチャデータに加えて、メモリに記憶されている頂点又は画像／テクスチャデータを参照し得る頂点プロセッサ2605及び／又はフラグメントプロセッサ2615A~2615Nを含むグラフィクスプロセッサ2610のための仮想-物理アドレスマッピングを提供する。1つ以上のMMU2620A~2620Bは、図25の1つ以上のアプリケーションプロセッサ2505、画像プロセッサ2515、及び／又はビデオプロセッサ2520に関連した1つ以上のMMUを含む、システム内の他のMMUと同期してよく、それにより、各プロセッサ2505~2520は、共有又は統合仮想メモリシステムに関与することができる。グラフィクスプロセッサ2610のコンポーネントは、本願で記載されている他のグラフィクスプロセッサのコンポーネントと対応してよい。1つ以上のMMU2620A~2620Bは、図2CのMMU245と対応してよい。頂点プロセッサ2605及びフラグメントプロセッサ2615A~2615Nは、グラフィクスマルチプロセッサ234と対応してよい。1つ以上の回路インターコネクト2630A~2630Bは、実施形態に従って、グラフィクスプロセッサ2610が、SoCの内部バスを介して、又は直接接続を介して、SoC内の他のIPコアとインターフェース接続することを可能にする。1つ以上の回路インターコネクト2630A~2630Bは、図2Cのデータクロスバー240と対応してよい。更なる対応は、グラフィクスプロセッサ2610の類似のコンポーネントと、本願で記載されている様々なグラフィクスプロセッサアーキテクチャとの間で見受けられ得る。

【0310】

図26Bに示されるように、グラフィクスプロセッサ2640は、図26Aのグラフィクスプロセッサ2610の1つ以上のMMU2620A~2620B、キャッシュ2625A~2625B、及び回路インターコネクト2630A~2630Bを含む。グラフィクスプロセッサ2640は、単一のコア又はタイプ若しくはコアが頂点シェーダ、フラグ

10

20

30

40

50

メントシェーダ、及び／又は計算シェーダを実装するようシェーダプログラムコードを含む全てのタイプのプログラム可能なシェーダコードを実行することができる統合シェーダコアアーキテクチャを提供する1つ以上のシェーダコア2655A～2655N（例えば、2655A、2655B、2655C、2655D、2655E、2655F、乃至2655N-1、及び2655N）を含む。シェーダコアの正確な数は、実施形態及び実施により様々であることができる。更には、グラフィクスプロセッサ2640は、シーンに対するレンダリング動作が、例えば、シーン内の局所空間コヒーレンスを利用するために、又は内部キャッシュの使用を最適化するために、画像空間において細分されるタイルベースのレンダリングのためのタイリング動作を加速させるよう実行スレッドを1つ以上のシェーダコア2655A～2655N及びタイリングユニット2658にディスパッチするスレッドディスパッチャとして動作するインターコアタスクマネージャ2645を含む。シェーダコア2655A～2655Nは、例えば、図2Dで見られたグラフィクスマルチプロセッサ234、又は図3A及び図3B夫々のグラフィクスマルチプロセッサ325、350、又は図3Cのマルチコアグループ365Aと対応してよい。

【0311】

本願で記載されている実施形態は、シストリック処理ユニットによりスパースデータに関して算術を実行する技術を提供するソフトウェア、ファームウェア、及びハードウェアロジックを含む。本願で記載されている実施形態は、ゼロ充てん行列及びサブ行列に対する計算動作をスキップする技術を提供する。実施形態は更に、処理ユニットまでデータ圧縮を維持する技術を提供する。実施形態は更に、スパースアウェアロジックユニットのためのアーキテクチャを提供する。

【0312】

〔テンソルアクセラレーションロジック及び統合メモリを備えたGPGPU〕

図27は、実施形態に従うデータ処理システム2700のブロック図である。データ処理システム2700は、プロセッサ2702と、統合メモリ2710と、機械学習アクセラレーションロジックを含むGPGPU2720とを備える異種（heterogeneous）処理システムである。プロセッサ2702及びGPGPU2720は、本願で記載されているプロセッサ及びGPGPU／並列プロセッサのいずれかであることができる。プロセッサ2702は、システムメモリ2712に記憶されているコンパイラ2715に対する命令を実行することができる。コンパイラ2715は、ソースコード2714Aをコンパイル済みコード2714Bにコンパイルするようプロセッサ2702で実行される。コンパイル済みコード2714Bは、プロセッサ2702によって実行され得る命令及び／又はGPGPU2720によって実行され得る命令を含むことができる。コンパイル中、コンパイラ2715は、コンパイル済みコード2714Bに存在するデータ並列性（data parallelism）のレベルに関するヒント及び／又はコンパイル済みコード2714Bに基づいてディスパッチされるべきスレッドに関連したデータ局所性（data locality）に関するヒントを含むメタデータを挿入する動作を実行することができる。コンパイラ2715は、そのような動作を実行するのに必要な情報を含むことができ、あるいは、動作は、ランタイムライブラリ2716の支援を受けて実行され得る。ランタイムライブラリ2716はまた、ソースコード2714Aのコンパイルにおいてコンパイラ2715を支援することもでき、GPGPU2720でのコンパイル済みコード2714Bの実行を助けるよう、ランタイム時にコンパイル済みコード2714Bとリンクされる命令を含むこともできる。

【0313】

統合メモリ2710は、プロセッサ2702及びGPGPU2720によってアクセスされ得る統合アドレス空間を表す。統合メモリは、GPGPUメモリ2718とともにシステムメモリ2712を含むことができる。GPGPUメモリ2718は、GPGPU2720のアドレス空間内のメモリであり、システムメモリ2712の一部又は全部を含むことができる。一実施形態において、GPGPUメモリ2718はまた、もっぱらGPGPU2720による使用に専用の如何なるメモリの少なくとも一部も含むことができる。一実施形態において、システムメモリ2712に記憶されているコンパイル済みコード2

714Bは、GPGPU2720によるアクセスのためにGPGPUメモリ2718にマッピングされ得る。

【0314】

GPGPU2720は、本願で記載されている様々な処理リソースの1つ以上を含むことができる複数の計算ブロック2724A～2724Nを含む。処理リソースは、例えば、実行ユニット、計算ユニット、ストリーミング・マルチプロセッサ、グラフィクスマルチプロセッサ、又はマルチコアグループなどの多種多様な計算リソースであるか、又はそれを含むことができる。一実施形態において、GPGPU2720は、行列演算（例えば、内積など）のサブセットを加速させるよう設計され得る1つ以上の特殊関数計算ユニットを含むことができるテンソル（例えば、行列）アクセラレータ2723を更に含む。テンソルアクセラレータ2723は、テンソルアクセラレータ又はテンソルコアとも呼ばれ得る。一実施形態において、テンソルアクセラレータ2723内のロジックコンポーネントは、複数の計算ブロック2724A～2724Nの処理リソースにわたって分布してよい。

10

【0315】

GPGPU2720はまた、レジスタ2725の組、電力及び性能モジュール2726、及びキャッシュ2727を含むがこれらに限らない、計算ブロック2724A～2724N及びテンソルアクセラレータ2723によって共有され得るリソースの組を含むこともできる。一実施形態において、レジスタ2725は、直接及び間接にアクセス可能なレジスタを含み、間接アクセス可能なレジスタは、テンソルアクセラレータ2723による使用のために最適化される。電力及び性能モジュール2726は、計算ブロック2724A～2724Nが計算ブロック2724A～2724N内のアイドル状態のコンポーネントをパワーゲーティングするための配電及びクロック周波数を調整するよう構成され得る。様々な実施形態において、キャッシュ2727は、命令キャッシュ及び／又は低電力データキャッシュを含むことができる。

20

【0316】

GPGPU2720は、テンソルアクセラレータ2723及び／又は計算ブロック2724A～2724N内の計算要素によって統合メモリ2710からアクセスされたデータをキャッシュするために使用され得るL3データキャッシュ2730を更に含むことができる。一実施形態において、L3データキャッシュ2730は、計算ブロック2724A～2724N内の計算要素及びテンソルアクセラレータ2723によって共有され得る共有ローカルメモリ2732を含む。

30

【0317】

一実施形態において、GPGPU2720は、フェッチ及び復号化ユニット2721並びにスケジューラコントローラ2722などの命令ハンドリングロジックを含む。フェッチ及び復号化ユニット2721は、計算ブロック2724A～2724N又はテンソルアクセラレータ2723のうちの1つ以上による実行のために命令をフェッチ及び復号するフェッチユニット及び復号化ユニットを含む。命令は、スケジューラコントローラ2722により計算ブロック2724A～2724N又はテンソルアクセラレータ2723内の適切な機能ユニットにスケジューリングされ得る。一実施形態において、スケジューラコントローラ2722は、高度なスケジューリング動作を実行するよう構成されるASICである。一実施形態において、スケジューラコントローラ2722は、ファームウェアモジュールからロードされたスケジューラ命令を実行することができるマイクロコントローラ又は低EPI（energy-per-instruction）処理コアである。

40

【0318】

一実施形態において、計算ブロック2724A～2724Nによって実行されるいくつかの関数は、テンソルアクセラレータ2723に対して直接にスケジューリング又はオフロードされ得る。様々な実施形態において、テンソルアクセラレータ2723は、3Dグラフィクス又は計算シェーダプログラムによって使用される乗算及び加算演算並びに内積演算などの行列計算動作を効率的に実行するよう構成された処理要素ロジックを含む。一

50

実施形態において、テンソルアクセラレータ 2723 は、機械学習フレームワークによって使用される動作を加速させるよう構成され得る。一実施形態において、テンソルアクセラレータ 2723 は、並列行列乗算及び／又は加算演算の特定の組を実行するよう明示的に構成された特定用途向け集積回路である。一実施形態において、テンソルアクセラレータ 2723 は、ワークロード間でアップデートされ得る固定関数ロジックを提供するフィールド・プログラマブル・ゲート・アレイ（FPGA）である。テンソルアクセラレータ 2723 によって実行され得る行列演算の組は、計算ブロック 2724A～2724N によって実行され得る動作に対して制限されてよい。しかし、テンソルアクセラレータ 2723 は、計算ブロック 2724A～2724N と比べてより高いスループットでこれらの演算を実行することができる。

10

【0319】

図 28 は、実施形態に従って、命令パイプライン 2800 によって実行される行列演算 2805 を表す。命令パイプライン 2800 は、例えば、内積演算などの、しかしこれに限られない行列演算の組を実行するよう構成され得る。2つのベクトルの内積は、ベクトルの対応する成分の積の和に等しいスカラー値である。内積は、以下の式（1）に示されるように計算され得る。

【数 1】

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + \dots + a_n b_n \quad (1)$$

20

【0320】

内積は、畳み込みニューラルネットワーク（CNN）の畳み込み演算で使用され得る。図 28 は、内積演算を含む行列演算 2805 を使用する 2次元（2D）畳み込みを表す。2D畳み込みが表されているが、N次元畳み込みが、N次元フィルタを用いてN次元ボリュームに対して実行され得る。受容野タイル 2802 は、入力ボリュームバッファ 2804 における入力ボリュームの一部を強調する。入力ボリュームバッファ 2804 は、メモリ 2830 に記憶され得る。内積行列演算 2805 は、出力バッファ 2806 内のデータ点を生成するよう受容野タイル 2802 内のデータと畳み込みフィルタとの間で実行され得る。出力バッファ 2806 もメモリ 2830 に記憶され得る。メモリ 2830 は、図 27 で見られたようなシステムメモリ 2712、GP-GPUメモリ 2718、又は1つ以上のキャッシュ 2727、2730を含む、本願で記載されているメモリのいずれかであることができる。

30

【0321】

出力バッファ 2806 内のデータ点の組み合わせは、畳み込み演算によって生成される活性化マップを表す。活性化マップ内の各点は、入力ボリュームバッファ 2804 にわたって受容野タイル 2802 をスライドさせることによって生成される。活性化マップは、出力活性化値を決定するよう活性化関数に入力され得る。一実施形態において、入力ボリュームバッファ 2804 の畳み込みは、ハイレベル行列演算 2805 としてフレームワーク内で定義され得る。ハイレベル行列演算 2805 は、BLAS（basic linear algebra subprogram）演算などのプリミティブ演算により実行され得る。プリミティブ演算は、命令パイプライン 2800 によって実行されるハードウェア命令により加速され得る。

40

【0322】

ハードウェア命令を加速させるために使用される命令パイプライン 2800 は、ハードウェア命令をフェッチ及び復号することができる命令フェッチ及び復号化ユニット 2721 と、計算ブロック 2724A～2724N 及び／又はテンソルアクセラレータ 2723 内の1つ以上の処理リソースに復号された命令をスケジューリングすることができるスケジューラコントローラ 2722 とを含むことができる。一実施形態において、ハードウェア

50

Ａ命令は、計算ブロック２７２４Ａ～２７２４Ｎにスケジューリングされ、テンソルアクセラレータ２７２３にオフロードされ得る。行列演算２８０５を実行するための１つ以上のハードウェア命令及び関連するデータは、メモリ２８３０に記憶され得る。ハードウェア命令の出力もメモリ２８３０に記憶され得る。

【０３２３】

一実施形態において、テンソルアクセラレータ２７２３は、１つ以上のハードウェア命令を実行して、組み込まれたシストリックアレイ２８０８（ＤＰロジック）を用いて行列演算２８０５を実行することができる。シストリックアレイ２８０８は、内積演算を実行するよう構成されるプログラム可能なハードウェア及び固定機能ハードウェアの組み合わせを含む。計算ブロック２７２４Ａ～２７２４Ｎ内の機能ユニットも、内積演算を実行するよう構成され得るが、シストリックアレイ２８０８は、計算ブロック２７２４Ａ～２７２４Ｎと比べてより一層高いスループットで内積演算の限られたサブセットを実行するよう構成され得る。

【０３２４】

図２９Ａ～２９Ｂは、いくつかの実施形態に従うハードウェアベースのシストリックアレイ２８０８の詳細を表す。図２９Ａは、単一のクロックサイクル内で複数の内積演算を実行するよう構成される複数の機能ユニットのグリッドを表す。図２９Ｂは、単一の、例示的な機能ユニットを表す。シストリック行列計算中、図２９Ａに示されるように、一実施形態において、シストリックアレイ２８０８は、様々な機能ユニットを用いて並列内積演算の組を実行するよう構成される。内積は、ＳＩＭＤデータが機能ユニットの複数のレイヤにわたってポンプされる‘シストリック’方式で実行され得る。シストリックアレイ２８０８は、グリッドで配置されている機能ユニットの集合である。機能ユニットのグリッドは、密集行進法（lockstep）で動作し、積和演算（multiply-accumulate）を実行するよう最適化される。シストリックアレイ２８０８によって作用される行列は、機能ユニットのグリッドにわたってポンプされるサブ行列に分けられる。

【０３２５】

一実施形態において、シストリックアレイ２８０８は、設定可能なシストリックデプスを用いてデータの設定可能な数のＳＩＭＤチャネルを処理することができる。所与の命令については、ＳＩＭＤ幅及びシストリックデプスは、ソースデータの組を処理するよう選択され得る。シストリックデプスは、命令を処理するために使用されることになるハードウェアロジックのシストリックレイヤの数を定義する。シストリックレイヤは、可変なＳＩＭＤ幅を有する乗算器及び加算器ロジックユニットのグループであり、シストリックレイヤは、入力として初期アキュムレータ値を受け取ることができ、その後のシストリックレイヤへ又は出力レジスタへ出力される内積値を生成する。

【０３２６】

いくつかの実施形態において、３つのソースが処理され得る。このとき、各ソースは、ベクトルレジスタ又はイミディエート（immediate）であることができる。一実施形態において、ソース２９００（ＳＲＣ０）は、１つ以上の初期アキュムレータ値であることができ、単一値又はアキュムレータ値のベクトルであることができる。初期アキュムレータ値は、最初のシストリックレイヤ内の各機能ユニットによって計算された内積の第１の組に加えられることになる。機能ユニットによって計算された内積は、所与のＳＩＭＤチャネルについての次のシストリックレイヤへ供給され得る。内積は、パケットデータの１つ以上のチャネルを含むことができるベクトルレジスタであるソース２９０１（ＳＲＣ１）及びソース２９０２（ＳＲＣ２）に基づいて計算され得る。各チャネルは４要素ベクトルを含む。一実施形態において、各チャネルは３２ビット幅であり、４つの８ビットベクトル要素を供給する。いくつかの実施形態は、８ビット要素、４ビット要素、及び／又は２ビット要素を有する入力ベクトルから内積を計算するよう構成される。一実施形態において、混合精度演算は、サポートされている要素サイズの任意の組み合わせ（例えば、８ビット×２ビット、８ビット×４ビット、４ビット×４ビット、など）を用いて実行され得る。一実施形態において、シストリックアレイ２８０８は整数計算のために構成されるが

、自動固定小数点演算がいくつかの実施形態では設定可能である。本願で記載されている命令は4要素内積であるが、いくつかの実施形態では、シストリックアレイ2808はまた、ベクトルごとに異なった要素数で浮動小数点内積計算をサポートするよう構成されてもよい。

【0327】

一実施形態において、4要素ベクトルの複数のチャンネルは、様々な幅（例えば、64ビット、128ビット、256ビット、512ビット、など）の単一ベクトルレジスタにパックされ得る。同時の内積が、ソース2901及びソース2902により供給されるベクトル要素の複数のチャンネルについてシストリックアレイ2808により計算され得る。処理されるべきベクトル要素のチャンネルの数は、内積計算のための選択された実行サイズ及びシストリックデプスに基づいて設定され得る。一実施形態において、指定された実行サイズ及び／又はシストリックデプスよりも広いソースベクトルは、シストリックアレイ2808の複数のサイクルを用いて計算されてよい。

【0328】

所与のクロックサイクル内で実行され得る計算の数は、SIMDレーン及びシストリックレイヤの数に基づいて様々であることができる。シストリックアレイ2808は、表されているように、4のシストリックデプスを用いてスループットのSIMDレーンごとに16個の内積を実行することができる。8つのSIMDレーンについて構成される場合に、ロジックは、所与のサイクル内で128個の8ビット整数（INT8）内積を実行することができる。8つのSIMDレーン及び8のシストリックデプスについて構成される場合に、各レーンは、全部で32個の8ビット整数（INT8）内積及び256個の内積を実行することができる。これらの具体的な演算の数は、一実施形態の例であり、他の実施形態は、スループットにおいて異なる。更には、データタイプが異なる場合には、演算の数は、異なるデータタイプに基づいて増減されることになる。

【0329】

各機能ユニットで、内積は、乗算器及び加算器ロジックにより計算され、内積は、アキュムレータ値に加えられる。結果として得られるデータは、あて先レジスタへ出力されるか、あるいは、次のシストリックレイヤのアキュムレータへ供給することができる。機能ユニット2912の詳細は、図29Bに示される。

【0330】

図29Bに示されるように、機能ユニット2912は、入力データバッファ2904、2906の組及びアキュムレータ2922を含むことができ、これらは夫々、入力データを受け入れることができる。一実施形態において、データバッファ2906は、入力データのパックされたベクトルであることができるソース2902（SRC2）を受け入れることができる。入力データバッファ2904は、やはり入力データのパックされたベクトルであることができるソース2901（SRC1）を受け入れることができる。アキュムレータ2922は、関数ユニット2912のための初期アキュムレータ値を供給するソース2900（SRC0）を受け入れることができる。初期アキュムレータ値は、ソース2901及びソース2902の要素から計算された内積に加えられる。内積は、乗算器2923A～2923Dの組及び加算器2924を用いてソースベクトルの要素ごとの積（element-wise multiplication）により計算される。乗算器2923A～2923Dは、積の組を計算するために使用される。積の組の和は、加算器2924によって計算される。和は、ソース2900により供給された任意の初期値と積算（例えば、それに加算）され得る。一実施形態において、この積算された値は、入力値2626として、その後のシストリックレイヤに存在することができる次のアキュムレータへ供給され得る。一実施形態において、ソース2901は、入力データの複数のチャンネルを含んでよい。ソース2901の更なるチャンネルは、更なるSIMDレーン2928へ入力されるSRC1として中継され得る。一実施形態において、ソース2902は、入力データの複数のチャンネルを含んでよい、ソース2902の更なるチャンネルは、更なるシストリックデプス内のロジックユニットへのSIC2入力データとして使用され得る。一実施形態において、ソース290

0 は、任意に、更なる機能ユニット内のアキュムレータへの入力として供給される更なるチャンネルとともに、複数のチャンネルを含むことができる。一実施形態において、ソース 2900 は、最初のシストリックレイヤの各機能ユニット内の各アキュムレータへ加えられる単一の値であることができる。

【0331】

〔ゼロ充てん行列及びサブ行列に対する計算動作のスキップ〕

一実施形態は、スパースデータを使用するときにシストリックアレイに対する訓練及び推論を最適化する技術を提供する。シストリックテンソルアレイ 2808 又はシストリックアレイ 1912 によって処理されるべき行列又はサブ行列が完全にゼロである場合に、行列又はサブ行列の次元値はゼロにセット可能であり、シストリックテンソルアレイ 2808 は、実行されるべき演算に応じて、サブ行列に関連した 1 つ以上の計算フェーズをバイパスしてよい。行列データの预处理中、ゼロサブ行列が識別可能であり、行列のサブ行列マップは、どのサブ行列がゼロ値しか含まないかを示すよう生成され得る。一実施形態において、少なくともいくつかの演算は、全体としてゼロ値から成る行又は列を含む行列又はサブ行列についてバイパスされてよい。一実施形態において、ただ 1 つの非ゼロ値も含むサブ行列も、バイパスされ得る。

【0332】

図 30A ~ 30B は、実施形態に従って、ゼロ値サブ行列をバイパスするシステムを表す。図 30B に示されるように、行列 3002 及び行列 3004 は、1 つ以上のサブ行列がゼロ値しか含まない行列である。処理ロジックは、サブ行列がゼロ値しか含まないかどうかを示すよう行列 3002 のサブ行列マップ 3012 及び行列 3004 のサブ行列 3014 を生成することができる。サブマップは、各サブ行列についてゼロとのビットワイズの比較を実行することを含む様々な技術を用いて生成され得る。サブ行列マップは、汎用の処理ロジック（例えば、CPU）で実行されるフレームワーク又はドライバによって生成され得るか、あるいは、処理リソース内の専用のハードウェアロジックによって生成され得る。一実施形態において、サブ行列 3005 に含まれている非ゼロ値の数が限られている場合に、そのサブ行列のシストリック演算もバイパスされてよい。代わりに、サブ行列 3005 に対する行列演算の結果は、シストリックテンソルアレイ 2808 の代わりに ALU を用いて計算され得る。

【0333】

図 30B に示されるように、メモリ 3020 は、行列 3002 及び行列 3004 を記憶することができる。シストリックテンソルアレイ 2808 は、行列 A ロードユニット 3026、行列 B ロードユニット 3022、行列 A フィードユニット 3028、及び行列 B フィードユニット 3024 を含むことができる。行列 3002 は、行列 B としてロード及び供給され得、一方、行列 3004 は、行列 A としてロード及び供給され得る。行列 A 及び行列 B のサブ行列は、シストリックテンソルアレイ 2808 の処理要素として動作する機能ユニット 3030 を通じてロード及び供給され得る。

【0334】

一実施形態において、ロード B フィルタ 3021 及びロード A フィルタ 3027 は、行列 3002 に対するサブ行列マップ 3012 及び行列 3004 に対するサブ行列マップ 3014 を記憶するバッファを含むことができる。ロード B フィルタ 3021 は、行列 B ロードユニット 3022 によるゼロ値サブ行列のロードをバイパスすることができる。ロード A フィルタ 3027 は、行列 A ロードユニット 3026 によるゼロ値サブ行列のロードをバイパスすることができる。バイパスされないサブ行列は、機能ユニット 3030 によって処理され得る。シストリックテンソルアレイ 2808 によって実行される演算に応じて、サブ行列の 1 つがゼロである場合に、演算全体がバイパスされ得る。サブ行列が単一の非ゼロ値行列を含むとき、実行されるべき演算に関連したサブ行列は、シストリックテンソルアレイ 2808 をバイパスすることができ、演算は、以下で図 31B に示されるようにシステム 3110 を介して ALU によって実行され得る。

【0335】

図 3 1 A ~ 3 1 B は、スパースデータに対して行列乗算演算を実行する方法 3 1 0 0 及びシステム 3 1 0 0 を表す。図 3 1 A は、ゼロを充てんされた行列及びサブ行列に対する行列乗算演算をスキップする方法 3 1 0 0 を表す。図 3 1 B は、近スパース (near-spars e) 行列及びサブ行列に対する演算がバイパスされ得るシステム 3 1 1 0 を表す。方法 3 1 0 0 は、本願で記載されているシストリックアレイ又はシストリックテンソルアレイを含む処理リソースによって実行されてよく、処理リソースは、ゼロ検出及び／又はゼロスキップ回路を含む圧縮又は符号化ロジックを更に含む。例示的なゼロ検出及び／又はゼロスキップ回路は、例えば、図 3 0 で見られるロード B フィルタ 3 0 2 1 及びロード A フィルタ 3 0 2 7 を含む。システム 3 1 1 0 は、例えば、計算ユニット 1 9 1 0 などの、しかしこれに限られない、本願で記載されている計算ユニットのコンポーネントを含む。

10

【 0 3 3 6 】

図 3 1 A に示されるように、方法 3 1 0 0 は、ゼロ検出及び／又はゼロスキップ回路を有するグラフィクス処理ユニット、グラフィクスマルチプロセッサ、又はグラフィクスプロセッサが、シストリックアレイの処理要素に入力されるべき行列のオペランドについてゼロ値を追跡することを含む (3 1 0 2)。回路が、入力として供給されるべきゼロ充てん行列又はゼロ充てんサブ行列を検出する場合に (3 1 0 4, Y E S)、回路は、行列又はサブ行列の全体に対する演算をバイパスすることができる (3 1 0 9)。ゼロ充てんサブ行列は、入力行列のゼロで満たされたブロック、又は行列のゼロで満たされた行若しくは列を含むことができる。入力として供給されるべき行列又はサブ行列がゼロで満たされていない場合には (3 1 0 4, N O)、回路は、ゼロ値入力に基づいてゼロ結果を有することになる行列乗算演算又は 1 つ以上のベクトル化された計算などの行列又はサブ行列に対する演算を実行することを進めることができる (3 1 0 6)。処理回路は、次いで、必要に応じてオペランドごとに計算動作をバイパスしてよい (3 1 0 7)。例えば、たとえ行列又はサブ行列の全体がバイパスされ得ないとしても、演算の総数は、個々のゼロ値オペランドに基づいて演算をスキップすることによって低減され得る。更には、近スパース行列又はサブ行列は、シストリックアレイをバイパスしてもよく、必要とされる限られた数の演算は、図 3 1 B に示されるように、処理リソースなしの従来のロジックユニットによって実行されてよい。

20

【 0 3 3 7 】

図 3 1 B は、限られた数の非ゼロ値を含むサブ行列に対する演算についてシストリックテンソルアレイをバイパスするシステム 3 1 1 0 を表す。システム 3 1 1 0 は、メモリ 3 0 2 0 と、図 1 9 で見られる計算ユニット 1 9 1 0 とを含む。計算ユニット 1 9 1 0 が表されているが、システム 3 1 1 0 は、代替的に、又は追加的に、本願で記載されている他のタイプの計算ユニット又は処理リソースを含んでよい。計算ユニット 1 9 1 0 は、レジスタファイル 1 9 0 6、A L U 1 9 1 1、シストリックアレイ 1 9 1 2、及び数学演算の特定のサブセットを実行するために最適化されているマスキングユニット 1 9 1 3 を含む。シストリックアレイ 1 9 1 2 に関して記載されている技術はまた、図 2 8 のシストリックテンソルアレイ 2 8 0 8 にも適用可能であってよい。

30

【 0 3 3 8 】

メモリ 3 1 2 0 に記憶されている行列 3 0 0 2 及び行列 3 0 0 4 の要素は、計算ユニット 1 9 1 0 のレジスタファイル 1 9 0 6 にロードされ得る。例えば、行列 3 0 0 4 のサブ行列が限られた数の非ゼロ値を含む場合に (例えば、図 3 0 A のサブ行列 3 0 0 5)、シストリックアレイ 1 9 1 2 内のロジックは、バイパスされた演算のオペランドと、実行されるべきバイパスされた演算とを記憶するレジスタを識別するサブ行列バイパスメッセージ 3 1 1 2 を A L U 1 9 1 1 へ伝送することができる。A L U 1 9 1 1 は、次いで、バイパスされたサブ行列データ 3 1 1 4 をレジスタファイル 1 9 0 6 から読み出し、ベクトル処理ロジックを用いて、バイパスされた演算を実行することができる。バイパスされた演算の処理は、シストリックアレイ 1 9 1 2 によって実行されるバイパスされていない演算と並行して実行され得る。

40

【 0 3 3 9 】

50

上記の技術を用いて、当業者は、例えば、テンソルアクセラレータを含む処理リソースを有し、前記テンソルアクセラレータが、入力テンソルのスパースサブ行列のロードをバイパスするロードフィルタを含む、グラフィクスプロセッサを実装してよい。テンソルアクセラレータは、入力行列データの組の1つ以上のサブ行列を処理するシストリックテンソルアレイを含む。処理リソースは、入力テンソルのバイパスされたサブ行列に対する演算を実行するロジックユニットを更に含むことができ、入力テンソルの前記バイパスされたサブ行列は、ゼロ値のみを含む。一実施形態において、バイパスされたサブ行列は、限られた数の非ゼロ値を含む。

【0340】

上記の技術を用いて、当業者はまた、テンソルアクセラレータ及びデコーダを含む処理リソースを有する装置を実装してもよい。テンソルアクセラレータは、入力テンソルのスパースサブ行列のロードをバイパスするロードフィルタを含む。デコーダは、入力テンソルに関連したデータの符号化された組を復号して、デコーダの復号された組を生成するよう構成される。デコーダは、データの符号化された組に関連したメタデータに基づいて、前記データの符号化された組を復号することができ、ロードフィルタは、前記データの符号化された組に関連したメタデータに基づいて、スパースサブ行列のロードをバイパスする。一実施形態において、デコーダは、前記データの符号化された組に関連したメタデータをロードフィルタへ供給する。前記データの符号化された組に関連したメタデータは、有効性写像を含み、有効性写像は、前記データの復号された組のビットストリームについてのゼロ又は非ゼロ値を示す。一実施形態において、デコーダは、テンソルアクセラレータに含まれ、テンソルアクセラレータは、1つ以上の符号化されたサブ行列を入力として読み出す。

【0341】

【E Uへの圧縮されたデータの送信】

G P Uは、3 D及びメディアアプリケーションによって使用されるピクセルデータの圧縮を可能にするデータ圧縮パイプラインを含むことができる。3 D及びメディアアプリケーションによって使用されるデータは、G P Uデータ圧縮の観点から、型付けされた (typed) データと見なされる。しかし、機械学習及びディープラーニング動作のためのデータなどのG P G P U計算データは、型指定されていない (untyped) データと見なされ、従来のG P G P U圧縮パイプラインは、型付けされたデータ圧縮技術を使用して、そのようなデータを圧縮することができない。例えば、型付けされたデータは、一般的に、メモリに順次に記憶されず、代わりに、圧縮されるべきデータのタイプに特有であるタイル、平面、又は他のデータフォーマットを用いて記憶されてよい。対照的に、型指定されていない計算データは、順次にメモリに記憶される。

【0342】

本願で記載されている実施形態は、G P G P U内の3 D圧縮ブロックが、型指定されていない計算データの圧縮を可能にするよう適応されることを可能にする。いくつかの実施形態において、現在のグラフィクスアセットは、3 Dデータの現在のパイプラインを用いて、機械学習 (M L) データの圧縮のために利用される。

【0343】

3 Dデータについては、どのようにデータがメモリにおいてレイアウトされるかを指定するデータに関連した状態メタデータがある。いくつかの実施形態において、型指定されていないデータの圧縮のために3 D圧縮ハードウェアを再利用するために、装置、システム、又はプロセスは、ソフトウェアによって型指定されていないデータのメモリ割り当てを検出し、割り当てについての状態メタデータを生成するよう構成される。G P G P Uに関連したグラフィクスドライバは、データが圧縮されるべきであるかどうかを決定するために使用可能であり、圧縮されたステータスは、割り当てに関連した状態メタデータ内で構成されてよい。状態メタデータは、型指定されていない計算データに関連したメモリ割り当てに対して圧縮が実行されるべきであることをシステムに示すために使用される。状態メタデータは、データ圧縮のために使用するパラメータを指定するために使用され得る

1つ以上のデータ構造を含むことができる。状態メタデータはまた、データへのポインタを含んでよく、バッファ、圧縮タイプ、及び型指定されていないデータを圧縮する際に使用される圧縮メタデータを記憶するために含まれる補助バッファへのポインタなどの要素を含んでもよい。

【0344】

いくつかの実施形態において、データポートは、フォーマット変換なしのDL/MLデータの伝送を可能にする。3Dデータフォーマットについては、ピクセルデータのためのシェーダフォーマットへの変換がある。しかし、異なるプロセスが、機械学習及びディープラーニングデータのために使用され、データのフォーマット変換は存在しない。いくつかの実施形態において、データポートは、フォーマット変換なしの伝送を可能にするよう変更される。いくつかの実施形態において、ソフトウェアが代わりにDL/MLをアドレスすることになる。

【0345】

図32は、GPGPUデータ圧縮パイプラインを含むグラフィクス処理システム3200のブロック図である。グラフィクス処理システム3200のデータ圧縮パイプラインは、型指定されていない計算データに加えて、3Dグラフィクス及びメディアデータの圧縮を可能にするよう構成される。グラフィクス処理システム3200は、シェーダコア3210、3212、3214の組、データポート3220、及びサーフェス状態キャッシュ3222を含む処理リソース3205を含む。シェーダコア3210、3212、3214は、グラフィクス処理システム3200のメモリサブシステム3234にアクセスするようデータポート3220へメモリ読み出し/書き込みメッセージを送る。メモリサブシステム3234へのアクセスは、キャッシュヒエラルキ3230によってキャッシュされる。キャッシュヒエラルキ内のキャッシュされたデータは、データがメモリサブシステム3234に書き込まれる前に、GPGPUコーデックユニット3232により圧縮されてよい。GPGPUコーデックユニット3232はまた、メモリサブシステム3234から読み出されるデータを、データがキャッシュヒエラルキ3230に書き込まれる前に圧縮解除することもある。

【0346】

シェーダメモリサーフェスは、サーフェス状態メタデータ3222に記憶され得るサーフェス状態に関連する。サーフェス状態は、ピクセルごとのビット、タイリングモード、クリアピクセルステータス、圧縮ステータス、などのサーフェス(surface)に関する情報を有する。この情報は、メモリサブシステムへ送信する前にデータを圧縮するためにGPGPUコーデックユニット3232によって使用される。ディープラーニング及び/又は機械学習ワークロードなどの、処理リソース3205で実行されるGPGPUプログラムについては、データは、通常は、「型指定されていない」(untyped)(すなわち、ハードウェアデータフォーマットは使用されない)。そして、データはタイリングされていない(すなわち、メモリにおいて線形にレイアウトされている)。ソフトウェアによるメモリ割り当て中に、GPUドライバは、データ圧縮がバッファに対して有効にされるか否かを決定するようヒューリスティクスを適用する。データ圧縮が有効にされる場合には、ドライバは、補助バッファを、圧縮メタデータを記憶するために割り当て、サーフェス状態も割り当てる。いくつかの実施形態において、コンパイラは、カーネルからバッファへの全てのアクセスが適切なサーフェス状態ポインタにより行われることを確かにすべきである。いくつかの実施形態において、GPGPUアプリケーションについては、サーフェス状態は、バッファメモリレイアウト(タイリングされていない、又は2D/3Dに構造化されている)を示す。これらのサーフェスのためのデータフォーマットは、サーフェスのデータタイプに依存する。例えば、ディープラーニング推論については、データタイプは、8ビット整数データタイプ(例えば、INT8)であってよい。ディープラーニング訓練については、フォーマットは、16ビット浮動小数点フォーマット(例えば、FP16、bfloat16)であってよい。

【0347】

いくつかの実施形態において、たとえ計算データサーフェスが指定されたフォーマットを有するとしても、データポート 3220 は、アクセス中に如何なるフォーマット変換も行わない。これがそれらのアプリケーションによって必要とされないからである。代わりに、フォーマット情報は、圧縮／圧縮解除アルゴリズムヒューリスティクスを駆動するためにしか使用されない。圧縮データ“ブロック”は、圧縮ユニットによって一緒に圧縮されるキャッシュライン（通常、2又は4つのキャッシュライン）のブロックである。一実施形態において、GPGPU圧縮のために、ブロック内のキャッシュラインアドレスは、常に順次的である。ディープラーニングプログラムは、通常は、アクセスにおいて優れた逐次的局所性（sequential locality）を有しており、このことは、ハードウェアを簡単なままとしながら優れた性能をもたらす。

10

【0348】

図33A～33Bは、実施形態に従って、並列計算プロセッサ又は汎用グラフィクス処理ユニットで処理リソースへのニューラルネットワークデータの圧縮された伝送を可能にするよう構成された計算アーキテクチャ3300、3320を表す。図33Aは、データ圧縮解除が計算ブロック3302内で実行される計算アーキテクチャ3300を表す。表されている計算ブロック3302は、図27で見られる計算ブロック2724A～2724Nのうちの1つを表してよい。図33Bは、データ圧縮及び圧縮解除が処理リソース内で実行される計算アーキテクチャ3320を表す。

【0349】

図33Aに示されるように、計算アーキテクチャ3300は、計算ブロック3302と、DMAコントローラ3306を介してメモリ3308へ結合されるハードウェアスクラッチバッファ3304とを含む。メモリ3308は、本願で記載されているデータ処理システムのメインメモリ又はシステムメモリであることができる。計算ブロック3302は、本願で記載されている処理リソースの組を含み、図27で見られる計算ブロック2724Aから2724Nのうちのいずれかと類似し得る。スクラッチバッファ3304は、オンチップの静的ランダム・アクセス・メモリ（SRAM）などの高速音チップメモリであることができる。一実施形態において、スクラッチバッファ3304は、計算ブロック3302によって実行されるニューラルネットワーク動作のための特徴ブロックユニット（feature block units）又はカーネルブロックユニットを記憶するよう最適化される。

20

【0350】

一実施形態において、デコーダ3312は、計算アーキテクチャにわたってニューラルネットワークデータの圧縮された伝送を可能にするよう計算ブロック3302に組み込まれるハードウェアデコーダロジックであることができる。例えば、CNNを処理する場合に、計算ブロック3302は、圧縮されていないフォーマットでスクラッチバッファ3304内の出力特徴マップ（output feature map, OFM）を生成することができる。エンコーダ3316は、圧縮されたフォーマットでメモリ3308へ出力特徴マップデータを書き込むことを可能にするようDMAコントローラ3306に組み込まれる。1つのレイヤのOFMが次のレイヤの入力特徴マップ（input feature map, IFM）になるとき、それらのIFMは、圧縮されたデータ3314としてメモリ3306から読み出され、スクラッチバッファ3304に記憶される。デコーダ3312は、データが復号される必要なしに、圧縮されたデータ3314で読み出すことを計算ブロック3302に可能にすることができる。代替的に、符号化及び復号化ロジックの両方を備えたコーデックユニットがコントローラ3306に組み込み可能であり、圧縮されたデータがDMAコントローラ3306によって伝送及び読み出しされることを可能にする。次いで、特徴マップデータは、DMAコントローラ3306によって圧縮解除され、計算ブロック3302によって読み出される圧縮されていないフォーマットでスクラッチバッファ3304に書き込まれ得る。

30

40

【0351】

本願で記載されている実施形態において、カーネル及び特徴データのための具体的な符号化フォーマットは、符号化されるべきデータの統計値に基づいて変更され得る。ニュー

50

ラルネットワーク特徴マップの解析は、多くの特徴マップが大いに疎（スパース（sparse））であり得ることを示す。ニューラルネットワークカーネルデータの解析は、カーネルデータが特徴マップほど疎でない一方で、カーネルデータ内の多くのデータが繰り返されることを示す。カーネルデータのダイナミックレンジは比較的に低く、このことは、ロー（raw）データが、係数を記憶するために必要とされるよりも多くのビットを割り当てることを示す。様々な符号化技術を用いて、特徴マップ及びカーネルデータは、様々な符号化技術の選択を用いて可逆的な方法で最大80%程度圧縮され得る。

【0352】

ニューラルネットワーク関連データは、例えば、一意絶対値（unique absolute value, UAV）テーブル符号化、有効性写像（SM）符号化、テーブル符号化（TE）、一意値座標（unique value coordinate, UVC）符号化、及び平均符号化（mean encoding, ME）などの、しかしこれらに限られない様々な符号化技術を用いて符号化（又は圧縮）され得る。符号化されたデータのメタデータは、データに使用される符号化フォーマットのタイプを示す。一実施形態において、特定の符号化フォーマットが、カーネルデータ又は特徴データなどの特定のタイプのデータのために選択され得る。一実施形態において、適切なエンコーダがデータの各ブロックについて選択されることを可能にするために、統計解析が符号化の前にデータに対して実行される。

【0353】

一実施形態において、SM符号化中に生成されたデータは、シストリックテンソルアレイ内のサブ行列バイパスを助けるために使用され得る。SM符号化モードでは、ブロック内の非ゼロ値のみが符号化される。サンプルブロック内の非ゼロ値の数は、ヘッダで示され、その後に、ブロック内の非ゼロ値のマップを示す有効性写像が続く。次いで、サンプルに非ゼロ値は、ストリーム内の出現の順に符号化される。

【0354】

図33Bに示されるように、計算アーキテクチャ3320は、処理リソース内でデータ圧縮及び圧縮解除を実行するロジックを含むことができる。計算アーキテクチャ3320は、図33Aの計算アーキテクチャ3300で見られるメモリ3308及びDMAコントローラ3306を含む。しかし、DMAコントローラ3306は、エンコーダ3316を除いてもよい。代わりに、DMAコントローラ3306を介してメモリ3308から読み出される圧縮されたデータは、圧縮されたフォーマットでキャッシュヒエラルキ（例えば、L3キャッシュ3326、L1キャッシュ3327）に記憶され得る。計算アーキテクチャ3320は、複数の処理リソース3328A～3328Nを有する計算ブロック3322を含み、各処理リソースは、ロード動作を介してキャッシュヒエラルキから圧縮されたデータを復号し、かつ、ストア動作を介してキャッシュヒエラルキに書き込まれた圧縮されたデータを符号化することができる。データは、処理リソース3328A～3328N内のシストリックアレイ3332A～3332Nへデータを供給する前に、コーデック3334A～3334Nによって圧縮解除されてよい。シストリックアレイ3332A～3332Nによって生成された結果は、データがキャッシュヒエラルキ及び／又はメモリ3308に書き込まれる前に、コーデック3334A～3334Nによって圧縮されてよい。

【0355】

図34は、出力データのためのゼロ検出ロジックを含む処理リソース3328を表す。表されている処理リソース3328は、図33Bの処理リソース3328A～3328Nのうちの1つであってよい。処理リソース3328は、処理リソース3328のレジスタファイルに記憶されているソースオペランド（例えば、SRC1 3432、SRC2 3433）に基づいて処理動作を実行するALU3434を含むことができる。ALU3434によって実行された動作の出力は、処理リソース3328のレジスタファイル内の一時あて先レジスタ3436に書き込まれ得る。処理リソース3328は、データが、一時あて先レジスタ3436からレジスタファイル内のあて先レジスタ3440へ流れる最中に、一時あて先レジスタ3436内のデータに対してゼロ検出動作を実行することができ

るゼロ検出口ジック 3 4 3 8 を更に含む。ゼロ検出口ジック 3 4 3 8 は、あて先レジスタ 3 4 4 0 に書き込まれる値がゼロ値であるかどうかを示すゼロ検出メタデータ 3 4 3 9 を書き込むことができる。ゼロ検出口ジック 3 4 3 8 は、処理リソース内のハードウェア回路であってよく、あるいは、処理リソース 3 3 2 8 のシェーダプロセッサによって実行されるシェーダプログラムなどの、より高位のロジックであってもよい。レジスタが、パックされたデータの複数の要素を含むベクトルレジスタである場合に、ゼロ検出メタデータ 3 4 3 9 は、ベクトル要素ごとに 1 ビットを含むビットフィールドであってよい。

【0356】

一実施形態において、ゼロ検出メタデータ 3 4 3 9 は、データ圧縮が有効にされる場合にあて先データを圧縮するために、処理リソース 3 3 2 8 内のコーデック 3 3 3 4 によって使用されてよい。ゼロ検出メタデータ 3 4 3 9 は、圧縮されたデータがキャッシュヒエラルキ及びメモリに記憶される場合に、圧縮されたデータを管理するために使用されるメタデータ（例えば、図 3 2 で見られるサーフェス状態キャッシュ 3 2 2 2 に記憶されているサーフェス状態データなど）を生成するために、コーデック 3 3 3 4 によって使用されてよい。

【0357】

様々な A L U 3 4 3 4 は、本願で記載されている様々な処理要素のうちのいずれか 1 つであってよく、並列整数及び浮動小数点演算を含む整数及び／又は浮動小数点演算を実行するよう構成されてよい。A L U 3 4 3 4 はまた、図 3 3 B で見られるシストリックアレイ 3 3 3 2 A ~ 3 3 3 2 N を表してもよい。

【0358】

図 3 5 は、処理リソース内の行列演算のためにデータを圧縮及び圧縮解除する方法 3 5 0 0 を表す。方法 3 5 0 0 は、図 3 4 で見られる処理リソース 3 3 2 8 又は図 3 3 B で見られる処理リソース 3 3 2 8 A ~ 3 3 2 8 N などの、圧縮及び圧縮解除ロジックを含む処理リソースによって実行されてよい。

【0359】

方法 3 5 0 0 は、処理リソースが、G P G P U のキャッシュヒエラルキから複数の行列の要素を含む圧縮されたデータを読み出すことを含む（3 5 0 2）。処理リソースは、圧縮されたデータを圧縮解除し、複数の行列の要素を処理リソースのレジスタファイルに格納することができる（3 5 0 4）。処理リソースは、次いで、複数の行列の要素を処理リソース内のシストリックアレイにロードすることができる（3 5 0 6）。例えば、処理リソースは、実行すべき演算（例えば、オペコード）並びにそれらの演算に関連したソース及びあて先レジスタとともにシストリックアレイヘメッセージを送ってよい。処理リソースは、次いで、指定された行列演算（例えば、オペコードにより指定された演算）をシストリックアレイにより実行してよく、シストリックアレイは、行列演算の出力をレジスタファイルに格納してよい（3 5 0 8）。

【0360】

いくつかの演算について、処理リソースは、処理リソースの追加の機能ユニット（例えば、A L U、F P U）により行列演算の出力に対して 1 つ以上の追加の演算を実行してもよい（3 5 1 0）。例えば、いくつかの後処理動作が、近スパス行列又はサブ行列に対してシストリックアレイによってバイパスされた演算の組を実行することを含め、シストリックアレイによって生成された出力に対して実行されてもよい。

【0361】

処理リソースは、次いで、処理リソースからの出力のために行列演算又は 1 つ以上の追加の演算の出力を圧縮することができる（3 5 1 2）。例えば、処理リソースは、圧縮されたデータ及び圧縮メタデータを G P G P U のキャッシュヒエラルキへ出力することができる。

【0362】

図 3 6 は、処理リソースによって生成された出力に対してゼロ検出動作を実行する方法 3 6 0 0 を表す。方法 3 6 0 0 は、図 3 4 で見られる処理リソース 3 3 2 8 などの、ゼロ

10

20

30

40

50

検出口ジックを含む処理リソースによって実行されてよい。

【0363】

方法3600は、処理リソースがGPGPUのキャッシュヒエラルキからデータ要素を読み出すことを含む(3602)。データ要素は、処理リソースによって実行されるべきSIMD又はSIMT演算のためのベクトル要素、パックされたデータ要素、又は行列要素であってよい。処理リソースは、次いで、処理リソースの1つ以上の機能ユニットによりデータ要素に対して1つ以上の演算を実行することができる(3604)。1つ以上の機能ユニットは、整数若しくは浮動小数点機能ユニット、混合制精度能ユニット、又は行列アクセラレーションユニットであってよい。行列アクセラレーションユニットは、例えば、本願で記載されているストリッックアレイ1912、ストリッックテンソルアレイ2808、又はストリッックアレイ3332A~3332Nのうちの1つであってよい。

10

【0364】

処理リソースは、次いで、1つ以上の演算の出力に対してゼロ検出動作を実行してよい(3606)。ゼロ検出動作は、処理リソース内のゼロ検出口ジックによって実行される。ゼロ検出口ジックは、処理リソース内の専用のハードウェア回路であってよい。ゼロ検出口ジックはまた、処理リソースにより実行されるシェードロジックであってもよい。ゼロ検出動作は、データが一時あて先レジスタから読み出され、演算のためにオペランドで指定されたあて先レジスタに書き込まれる最中に、実行されてよい。

【0365】

処理リソースは、次いで、1つ以上の演算の出力及びゼロ検出メタデータをGPGPUのキャッシュヒエラルキに書き込んでよい(3608)。一実施形態において、ゼロ検出メタデータは、データをキャッシュヒエラルキに書き込む前に、データを圧縮するためにコーデックによって使用されてよい。データは、GPGPUのキャッシュヒエラルキ及びメモリ内で圧縮されたままであってよい。圧縮されたデータがCPUによって読み出される必要があるならば、圧縮解除ロジックが、CPU可視メモリ(CPU visible memory)への書き込み中にデータを圧縮解除してよい。一実施形態において、圧縮解除ロジックは、データをCPU可視メモリに書き込むために使用され、DMAコントローラ内に存在してよい。

20

【0366】

上記の技術を用いて、当業者は、例えば、1つ以上の行列乗算演算を実行するテンソルアクセラレータを含む処理リソースと、該処理リソースによって実行されるべきロード動作により受け取られた圧縮されたデータを圧縮解除するコーデックハードウェアとを有し、圧縮されたデータが、1つ以上の行列乗算演算のオペランドによって識別される入力行列の要素を含む、汎用グラフィクスプロセッサを実装してよい。汎用グラフィクスプロセッサは、レジスタファイルを含み、コーデックハードウェアは、入力行列の要素をレジスタファイル内のレジスタに書き込むことができる。テンソルアクセラレータは、レジスタファイルから入力行列の要素を読み出すことができる。処理リソースは、コーデックハードウェア及びレジスタファイルを含むことができる。

30

【0367】

上記の技術を用いて、当業者は、例えば、汎用グラフィクスプロセッサユニット(GPGPU)において、GPGPUの処理リソースでGPGPUのキャッシュヒエラルキからデータ要素を読み出すことと、処理リソースの1つ以上の機能ユニットによりデータ要素に対して1つ以上の演算を実行することと、処理リソース内のゼロ検出口ジックにより、1つ以上の演算の出力に対してゼロ検出動作を実行することと、1つ以上の演算の出力及びゼロ検出メタデータをGPGPUのキャッシュヒエラルキに書き込むこととを有する方法を実装してよい。方法は、コーデックユニットの圧縮器によりゼロ検出メタデータを用いて1つ以上の演算の出力を圧縮することを更に有してよく、コーデックユニットは、GPGPU内の処理リソース内に含まれる。

40

【0368】

[更なる例となるコンピューティングデバイス]

50

図37は、実施形態に従うグラフィクスプロセッサ3704を含むコンピューティングデバイス3700のブロック図である。コンピューティングデバイス3700は、上述された実施形態の夫々の機能性を含むコンピューティングデバイスであることができる。コンピューティングデバイス3700は、セットトップボックス（例えば、インターネットに基づくケーブルテレビジョンセットトップボックス、など）、グローバル・ポジショニング・システム（GPS）ベースのデバイス、などのような通信デバイスであっても、あるいは、そのようなものに含まれてもよい。コンピューティングデバイス3700はまた、携帯電話機、スマートフォン、パーソナル・デジタル・アシスタント（PDA）、タブレットコンピュータ、ラップトップコンピュータ、電子リーダー、スマートテレビジョン、テレビジョンプラットフォーム、ウェアラブルデバイス（例えば、メガネ、時計、ブレスレット、スマートカード、宝石、衣服アイテム、など）、メディアプレイヤー、などのモバイルコンピューティングデバイスであっても、あるいは、そのようなものに含まれてもよい。例えば、一実施形態において、コンピューティングデバイス3700は、単一のチップ上にコンピューティングデバイス3700の様々なハードウェア及び／又はソフトウェアコンポーネントを組み込むシステム・オン・ア・チップ（「SoC」又は「SOC」）などの集積回路（「IC」）を用いるモバイルコンピューティングデバイスを含む。

【0369】

コンピューティングデバイス3700は、グラフィクスプロセッサ3704を含む。グラフィクスプロセッサ3704は、本願で記載されている如何なるグラフィクスプロセッサも表す。グラフィクスプロセッサは、本願で記載されている1つ以上のグラフィクスエンジン、グラフィクスプロセッサコア、及び他のグラフィクス実行リソースを含む。そのようなグラフィクス実行リソースは、実行ユニット、シェーダエンジン、フラグメントプロセッサ、頂点プロセッサ、ストリーミング・マルチプロセッサ、グラフィクスプロセッサクラスタ、あるいは、グラフィクスリソース若しくは画像リソースの処理、又は異種プロセッサで汎用計算動作を実行することに適したコンピューティングリソースの任意の集合を含むがこれらに限られない形で、提示されてよい。

【0370】

一実施形態において、グラフィクスプロセッサ3704は、任意の数のL1、L2、L3、又はL4キャッシュ、レンダーキャッシュ、デプスキャッシュ、サンブラキャッシュ、及び／又はシェーダユニットキャッシュを含むがこれらに限られない、単一のキャッシュであるか、あるいは、キャッシュメモリの複数のセグメントに分割され得るキャッシュ3714を含む。キャッシュ3714は、本願で記載されている至る所の領域を有してよい。キャッシュ3714はまた、メモリバンクハッシュアルゴリズムの動的な再構成をサポートする動的ハッシュロジックを含んでもよい。いくつかの実施形態において、グラフィクスプロセッサ3704は、GPGPUエンジン3744によって使用されるレジスタを含むレジスタファイル3724に加えて、共有ローカルメモリ（SLM3734）を含むGPGPUエンジン3744を含む。レジスタファイル3724は、汎用レジスタ、アーキテクチャレジスタ、構成レジスタ、及び他のタイプのレジスタを含むことができる。汎用レジスタファイル（GRF）及びアーキテクチャレジスタファイル（ARF）はまた、GPGPUエンジン3744内の計算ユニット（例えば、計算3750、計算3755）の1つ以上のブロック内の処理リソースの中にも存在することができる。共有ファブリック3742も存在してよく、GPGPUエンジン3744の様々なコンポーネント間の迅速な通信を可能にする。

【0371】

表されているように、一実施形態において、グラフィクスプロセッサ3704に加えて、コンピューティングデバイス3700は、アプリケーションプロセッサ3706、メモリ3708、及び入出力（I/O）ソース3710を含むがこれらに限られない如何なる数及びタイプのハードウェアコンポーネント及び／又はソフトウェアコンポーネントも更に含んでよい。アプリケーションプロセッサ3706は、グラフィクスパイプライン機能性を共有するようハードウェアグラフィクスパイプラインと相互作用することができる。

処理されたデータは、ハードウェアグラフィクスパイプライン内のバッファに格納され、状態情報は、メモリ3708に格納される。結果として得られるデータは、本願で記載されるように、表示デバイスによる出力のためにディスプレイコントローラへ転送され得る。表示デバイス、陰極線管（CRT）、薄膜トランジスタ（TFT）、液晶ディスプレイ（LCD）、有機発光ダイオード（OLED）アレイ、などのような様々なタイプであってよく、グラフィカルユーザインターフェースによりユーザに情報を表示するよう構成されてよい。

【0372】

アプリケーションプロセッサ3706は、1つ以上のプロセッサを含むことができ、コンピューティングデバイス3700のためのオペレーティングシステム（OS）3702を実行するよう少なくとも部分的に使用される中央演算処理装置（CPU）であってよい。OS3702は、コンピューティングデバイス3700のハードウェア及び／又は物理リソースと1人以上のユーザとの間のインターフェースとなることができる。OS3702は、図23のユーザモードグラフィクスドライバ2326及び／又はカーネルグラフィクスドライバ2329などのグラフィクスドライバロジック3722を含む、コンピューティングデバイス3700内の様々なハードウェアデバイスのためのドライバロジックを含むことができる。

【0373】

いくつかの実施形態において、グラフィクスプロセッサ3904は、アプリケーションプロセッサ3906の部分（例えば、物理CPUパッケージの部分）として存在してよいことが企図され、その場合に、メモリ3708の少なくとも一部は、アプリケーションプロセッサ3706及びグラフィクスプロセッサ3704によって共有されてよい。なお、メモリ3708の少なくとも一部は、グラフィクスプロセッサ3704に専用であってよく、あるいは、グラフィクスプロセッサ3704は、メモリの別個のストアを有してもよい。メモリ3708は、バッファ（例えば、フレームバッファ）の前もって割り当てられた領域を有してよいが、実施形態はそのように限定されず、より低いグラフィクスパイプラインにアクセス可能な如何なるメモリも使用されてよいことが、当業者によって理解されるべきである。メモリ3708は、デスクトップ又は3Dグラフィクスシーンをレンダリングするためにグラフィクスプロセッサ3704を利用するアプリケーションを有する様々な形のランダム・アクセス・メモリ（RAM）（例えば、SDRAM、SRAM、など）を含んでよい。メモリコントローラハブは、メモリ3708内のデータにアクセスし、それをグラフィクスパイプライン処理のためにグラフィクスプロセッサ3704へ転送してよい。メモリ3708は、コンピューティングデバイス3700内の他のコンポーネントに利用可能にされてよい。例えば、コンピューティングデバイス3700の様々なI/Oソース3710から受け取られる如何なるデータ（例えば、入力グラフィクスデータ）も、それらが、ソフトウェアプログラム又はアプリケーションの実装において、1つ以上のプロセッサ（例えば、アプリケーションプロセッサ3706）によって作用される前に、メモリ3708に一時的にキューされ得る。同様に、ソフトウェアプログラムが決定するデータは、コンピューティングデバイス3700から外部エンティティへコンピューティングシステムインターフェースを通じて送信されるか、あるいは、内部記憶要素に格納されるべきであり、しばしば、それが伝送又は記憶される前に、メモリ3708に一時的にキューされる。

【0374】

I/Oソース3710は、タッチスクリーン、タッチパネル、タッチパッド、仮想又は通常のキーボード、仮想又は通常のマウス、ポート、コネクタ、ネットワークデバイス、などのようなデバイスを含むことができ、図1で見られたようなI/Oハブ107、図3で見られたような入出力（I/O）回路363、図14で見られたようなプラットフォームコントローラハブ1430、などにより取り付けることができる。更には、I/Oソース3710は、コンピューティングデバイス3700へ及び／又はそれからデータを転送するために（例えば、ネットワーキングアダプタ）、あるいは、コンピューティングデバ

10

20

30

40

50

イス 3700 内の大規模不揮発性ストレージのために（例えば、ハードディスクドライブ）実装される 1 つ以上の I/O デバイスを含んでよい。英数字及び他のキーを含むユーザ入力デバイスは、情報及びコマンド選択をグラフィクスプロセッサ 3704 へ送るために使用されてよい。他のタイプのユーザ入力デバイスは、方向情報及びコマンド選択を GPU へ送りかつ表示デバイス上でカーソル移動を制御するための、マウス、トラックボール、タッチスクリーン、タッチパッド、又はカーソル方向キーなどのカーソルコントロールである。コンピューティングデバイス 3700 のカメラ及びマイクロホンアレイは、ジェスチャを観測し、音声及び映像を記録し、かつ視覚及び音声コマンドを送受信するために用いられてよい。

【0375】

ネットワークインターフェースとして構成された I/O ソース 3710 は、LAN、ワイドエリアネットワーク（WAN）、メトロポリタンエリアネットワーク（MAN）、パーソナルエリアネットワーク（PAN）、Bluetooth（登録商標）、クラウドネットワーク、セルラー又はモバイルネットワーク（第 3 世代（3G）、第 4 世代（4G）、第 5 世代（5G）など）、衛星ネットワーク、イントラネット、インターネット、などのネットワークへのアクセスを提供することができる。ネットワークインターフェースは、例えば、1 つ以上のアンテナを備える無線ネットワークインターフェースを含んでよい。ネットワークインターフェースはまた、例えば、Ethernet（登録商標）ケーブル、同軸ケーブル、光ファイバケーブル、シリアルケーブル、又はパラレルケーブルであってよいネットワークケーブルを介して遠隔のデバイスと通信するための、例えば、有線ネットワークインターフェースを含んでもよい。

【0376】

ネットワークインターフェースは、例えば、IEEE 802.11 標準に従うことによって、LAN へのアクセスを提供してよく、かつ／あるいは、無線ネットワークインターフェースは、例えば、Bluetooth 標準に従うことによって、パーソナルエリアネットワークへのアクセスを提供してよい。従前及びその後のバージョンの標準を含む他の無線ネットワークインターフェース及び／又はプロトコルもサポートされてよい。無線 LAN 標準による通信に加えて、又はその代わりに、ネットワークインターフェースは、例えば、時分割多重アクセス（TDMA）プロトコル、グローバル・システム・フォー・モバイル・コミュニケーションズ（GSM）プロトコル、符号分割多重アクセス（CDMA）プロトコル、及び／又はあらゆる他のタイプの無線通信プロトコルを用いて、無線通信を提供してよい。

【0377】

当然ながら、上記の例よりも少なく又は多く装備されたシステムが、特定の実施のためには望ましいことがある。従って、コンピューティングデバイス 3700 の構成は、価格制約、性能要件、技術的改善、又は他の環境などの多数の因子に応じて実施ごとに異なっている。例は、（限定なしに）モバイルデバイス、パーソナル・デジタル・アシスタント、モバイルコンピューティングデバイス、スマートフォン、携帯電話機、ハンドセット、一方向ページャー、双方向ページャー、メッセージングデバイス、コンピュータ、パーソナルコンピュータ（PC）、デスクトップコンピュータ、ラップトップコンピュータ、ノートブックコンピュータ、手持ち式コンピュータ、タブレットコンピュータ、サーバ、サーバアレイ又はサーバファーム、ウェブサーバ、ネットワークサーバ、インターネットサーバ、ワークステーション、ミニコンピュータ、メインフレームコンピュータ、スーパーコンピュータ、ネットワークアプライアンス、ウェブアプライアンス、分散コンピュータシステム、マルチプロセッサシステム、プロセッサベースのシステム、コンシューマエレクトロニクス、プログラム可能なコンシューマエレクトロニクス、テレビ受像機、デジタルテレビ受像機、セットトップボックス、ワイヤレス・アクセス・ポイント、基地局、加入者局、モバイル加入者センター、無線ネットワークコントローラ、ルータ、ハブ、ゲートウェイ、ブリッジ、スイッチ、マシン、又はそれらの組み合わせを含む。

【0378】

実施形態は、ペアレントボードを用いて相互接続される1つ以上のマイクロチップ若しくは集積回路、ハードワードロジック、メモリデバイスによって記憶されてマイクロプロセッサによって実行されるソフトウェア、ファームウェア、特定用途向け集積回路（ASIC）、及び／又はフィールド・プログラマブル・ゲート・アレイ（FPGA）のうちのいずれか1つ又は組み合わせとして実装されてもよい。「ロジック」との用語は、例として、ソフトウェア若しくはハードウェア、及び／又はソフトウェアとハードウェアとの組み合わせを含んでよい。

【0379】

実施形態は、コンピュータプログラム製品として提供されてよい。コンピュータプログラム製品は、例えば、コンピュータ、コンピュータのネットワーク、又は他の電子デバイスなどの1つ以上のマシンによって実行される場合に、1つ以上のマシンが本願で記載されている実施形態に従って動作することをもたらし得るマシン実行可能な命令を記憶している1つ以上のマシン読み出し可能な媒体を含んでよい。マシン読み出し可能な媒体は、フロッピー（登録商標）ディスク、光ディスク、CD-ROM（Compact Disc-Read Only Memories）、及び磁気光ディスク、ROM、RAM、EPROM（Erasable Programmable Read Only Memories）、EEPROM（Electrically Erasable Programmable Read Only Memories）、磁気若しくは光カード、フラッシュメモリ、又はマシン実行可能な命令を記憶するのに適した他のタイプの非一時的なマシン読み出し可能な媒体を含んでよいが、これらに限定されない。

【0380】

更に、実施形態は、コンピュータプログラム製品としてダウンロードされてよく、プログラムは、通信リンク（例えば、モデム及び／又はネットワーク接続）を介して搬送波又は他の伝搬媒体で具現化及び／又は変調された1つ以上のデータ信号を用いて、遠隔のコンピュータ（例えば、サーバ）から要求元のコンピュータ（例えば、クライアント）へ転送されてよい。

【0381】

本願での「一実施形態」又は「実施形態」との言及は、実施形態に関連して記載されている特定の特徴、構造又は特性が本発明の少なくとも1つの実施形態に含まれ得ることを意味する。本明細書の様々な箇所での「一実施形態において」との表現の出現は、必ずしも全てが同じ実施形態に言及しているわけではない。続く図に表されているプロセスは、ハードウェア（例えば、回路、専用ロジック、など）、ソフトウェア（非一時的なマシン読み出し可能な記憶媒体上の命令として）、又はハードウェア及びソフトウェアの両方の組み合わせを有する処理ロジックによって実行され得る。添付の図面において例として説明されている様々な実施形態が詳細に参照される。続く詳細な説明では、多数の具体的な詳細が、本発明の完全な理解をもたらすために説明されている。しかし、当業者に明らかのように、本発明は、それらの具体的な詳細によらずとも実施されてよい。他の事例では、よく知られている方法、プロシージャ、コンポーネント、及び回路は、実施形態の態様を不必要に不明りようにしないように、詳細に説明されていない。

【0382】

また、第1、第2、などの用語が、様々な要素について記載するために本願で使用されることがあるが、これらの要素は、これらの用語によって限定されるべきでないことも理解されるだろう。これらの用語は、1つの要素を他と区別するためにのみ使用される。例えば、本発明の範囲から外れずに、第1接点は、第2接点と呼ばれてもよく、第2接点は、第1接点と呼ばれてもよい。第1接点及び第2接点は両方とも接点であるが、それらは同じ接点ではない。

【0383】

本願で使用されている専門用語は、特定の実施形態についてのみ記載することを目的とし、全ての実施形態に関して限定であるよう意図されない。本発明の説明及び添付の特許請求の範囲で使用されるように、単数形（「a」、「an」及び「the」）は、文脈上で明示的に別なふうに特定されない限りは、複数形も含むと意図される。また、本願で使用され

10

20

30

40

50

ている「及び／又は」(an/or)との用語は、関連付けて挙げられている項目のうちの1つ以上のありとあらゆる可能な組み合わせを指し包含していることが理解されるだろう。更には、「有する」及び／又は「有している」という用語が、本明細書で使用される場合に、述べられている特徴、整数、ステップ、動作、要素、及び／又はコンポーネントの存在を特定するが、1つ以上の他の特徴、整数、ステップ、動作、要素、コンポーネント、及び／又はそれらのグループの存在又は追加を除外しないことが理解されるだろう。

【0384】

本願で使用されるように、「～である場合」(if)との用語は、文脈に依存して、「～であるとき」(when)又は「～すると」(upon)又は「～を決定することに応答して」(in response to determining)又は「～を検出することに応答して」(in response to detecting)を意味すると解釈され得る。同様に、「～と決定される場合」(if it is determined)又は「(述べられている条件又はイベント)が検出される場合」(if [a stated condition or event] is detected)との表現は、文脈に応じて、「～と決定すると」又は「～と決定することに応答して」又は「(述べられている条件又はイベント)を検出すると」又は「(述べられている条件又はイベント)を検出することに応答して」を意味すると解釈され得る。

【0385】

本願で記載されている実施形態は、シストリック処理ユニットによりスパースデータに関して算術を実行する技術を提供するソフトウェア、ファームウェア、及びハードウェアロジックを含む。本願で記載されている実施形態は、ゼロ充てん行列及びサブ行列に対する計算動作をスキップする技術を提供する。実施形態は更に、処理ユニットまでデータ圧縮を維持する技術を提供する。実施形態は更に、スパースアウェアロジックユニットのためのアーキテクチャを提供する。

【0386】

一実施形態は、行列アクセラレータ及びデコーダを含む処理リソースを有し、前記行列アクセラレータは、入力行列のスパースサブ行列のロードをバイパスするロードフィルタを含み、前記デコーダは、前記入力行列に関連したデータの符号化された組を復号して、データの復号された組を生成する、汎用グラフィクスプロセッサを提供する。前記デコーダは、前記データの符号化された組に関連したメタデータに基づいて、前記データの符号化された組を復号することができる。前記ロードフィルタは、前記データの符号化された組に関連した前記メタデータに基づいて、前記スパースサブ行列のロードをバイパスすることができる。前記デコーダは、前記データの符号化された組に関連した前記メタデータを前記ロードフィルタへ供給することができる。前記データの符号化された組に関連した前記メタデータは、有効性写像を含み、該有効性写像は、前記データの復号された組のビットストリームについてのゼロ又は非ゼロ値を示す。前記デコーダは、前記行列アクセラレータに含まれてもよく、前記行列アクセラレータは、1つ以上の符号化されたサブ行列を入力として読み込むことができる。行列アクセラレータは、複数の処理要素を含むことができ、該複数の処理要素は、シストリックアレイとして配置され、入力行列データの組の1つ以上のサブ行列を処理するよう構成される。前記処理リソースは、前記入力テンソルのバイパスされたサブ行列に対する演算を実行するハードウェアロジックユニットを更に含むことができる。

【0387】

一実施形態は、メモリデバイスと、上記の汎用グラフィクスプロセッサとを有し、前記メモリデバイスが前記汎用グラフィクスプロセッサと結合される、データ処理システムを提供する。

【0388】

一実施形態は、行列アクセラレータを含む汎用グラフィクス処理ユニットで、ゼロ検出回路により、行列アクセラレータへ入力されるべき行列のゼロ値オペランドを監視し、ゼロ値オペランドの検出に応答して、前記行列アクセラレータでの行列乗算演算をバイパスすることを有する方法を提供する。ゼロ値オペランドは、入力行列の第1サブ行列に関連

10

20

30

40

50

してよく、該第1サブ行列は、ゼロ値サブ行列である。前記入力行列がゼロ値行列であるとき、前記行列アクセラレータでの全ての行列演算がバイパスされてよい。一実施形態において、方法は、前記入力行列の第2サブ行列の検出に応答して、行列アクセラレータをバイパスすることを更に有し、前記第2サブ行列は、限られた数の非ゼロ値を有している。該限られた数の非ゼロ値は、単一の非ゼロ値などの所定の値であってよい。方法は、前記行列アクセラレータの外の処理リソースへメッセージを送信することを更に含み、該メッセージは、前記第2サブ行列のバイパスを示す。前記第2サブ行列に対する前記バイパスされた演算は、前記メッセージに基づいて前記処理リソースによって実行されてよく、前記メッセージは、バイパスされた演算及び該バイパスされた演算のためのオペランドを識別することができる。一実施形態において、ゼロ検出回路により、行列アクセラレータへ入力されるべき行列の非ゼロ値オペランドを監視することは、第1入力行列のサブ行列マップを生成することを含み、該サブ行列マップは、前記第1入力行列のゼロ値サブ行列を識別してよい。

【0389】

上記の説明及び図面は、限定の意味ではなく例示の意味でとられるべきである。当業者であれば、添付の特許請求の範囲に示されている特徴のより広い精神及び範囲から逸脱せずに、本願で記載されている実施形態に対して、様々な変更及び改良が行われてよいと理解するだろう。

10

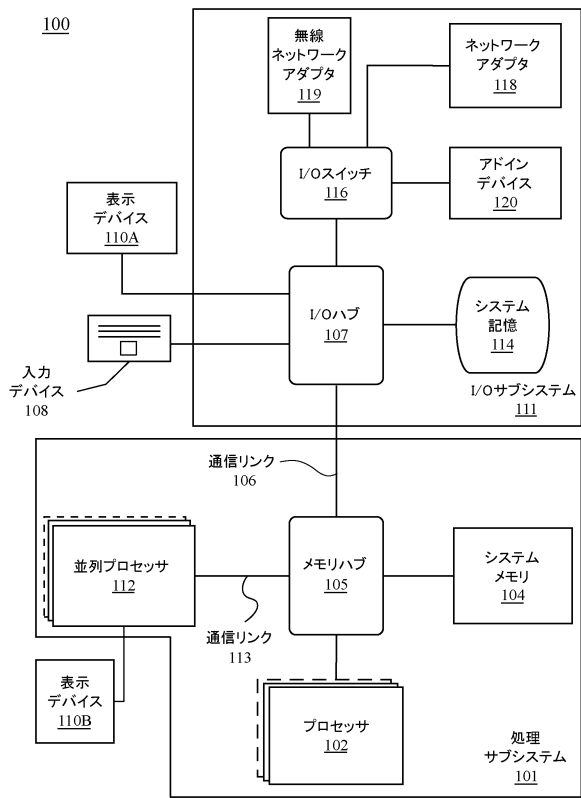
20

30

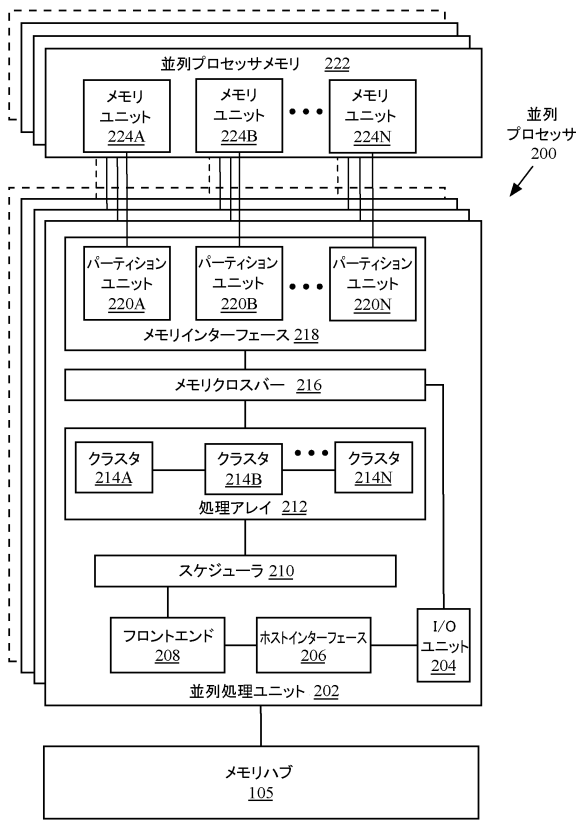
40

50

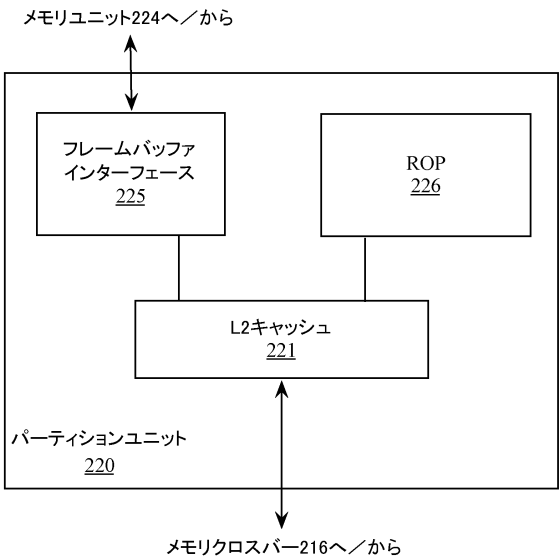
【図面】
【図 1】



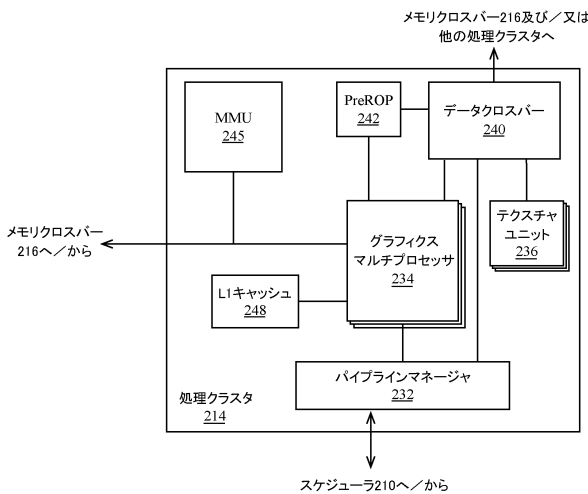
【図 2 A】



【図 2 B】



【図 2 C】



10

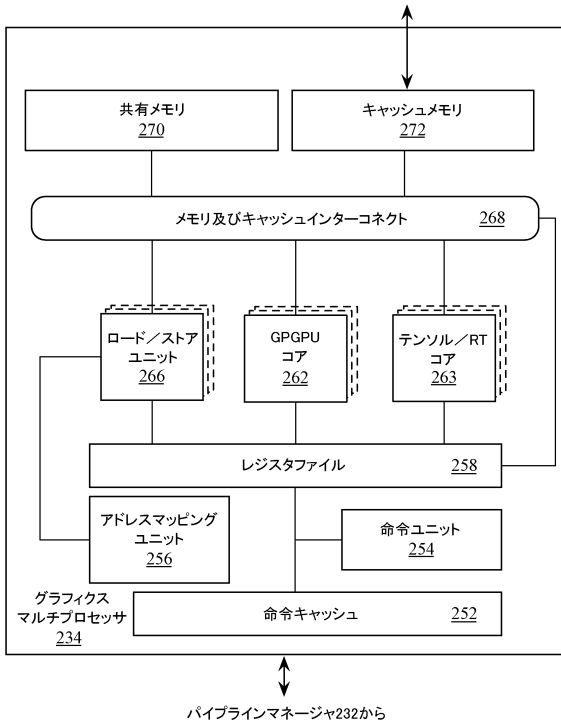
20

30

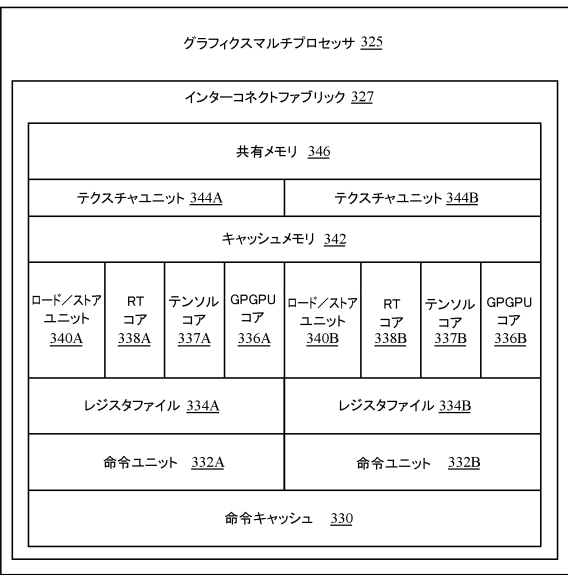
40

50

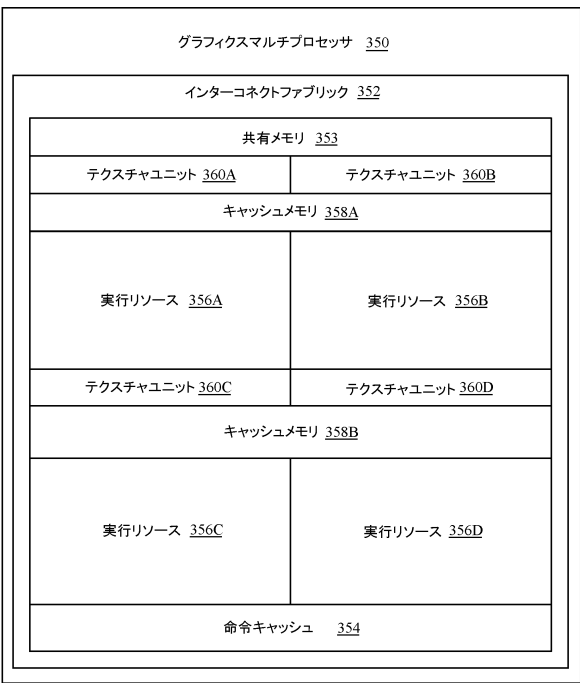
【図 2 D】



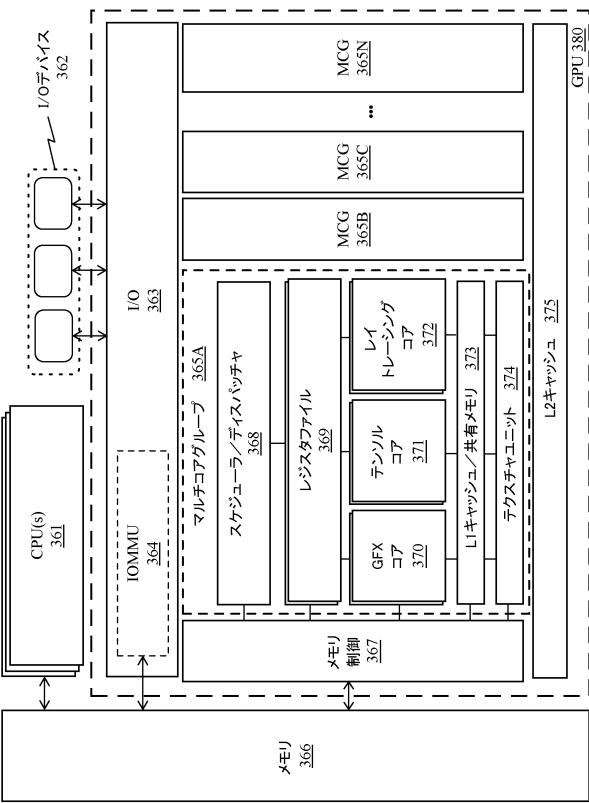
【図 3 A】



【図 3 B】



【図 3 C】



10

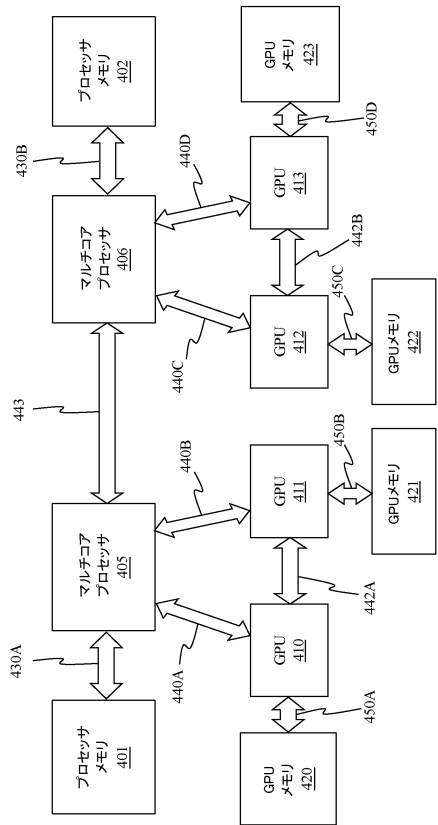
20

30

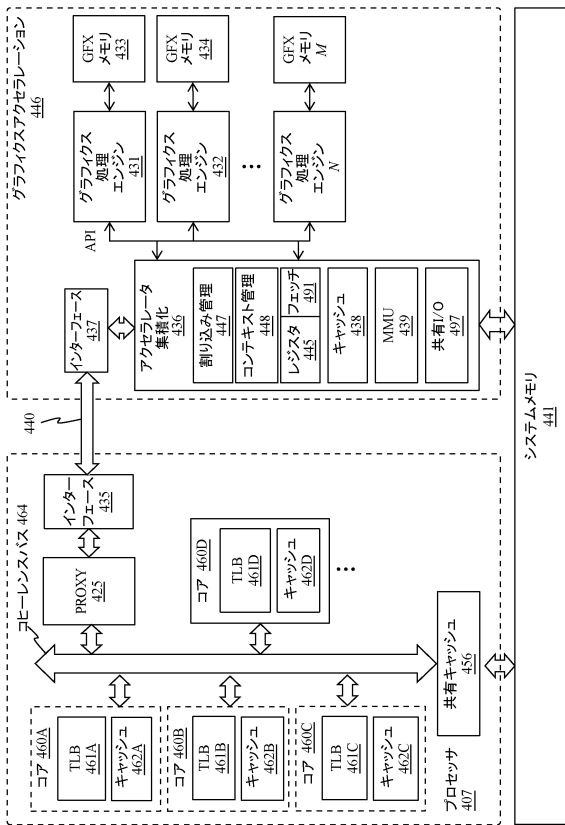
40

50

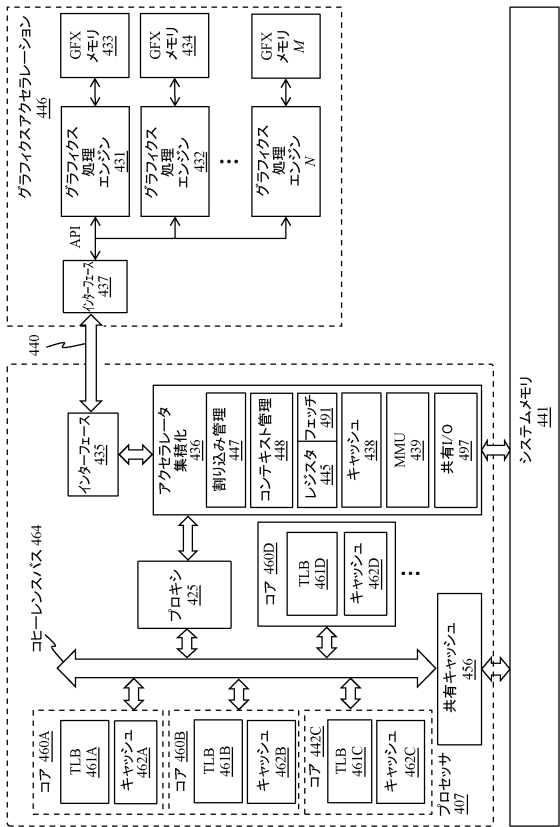
【図 4 A】



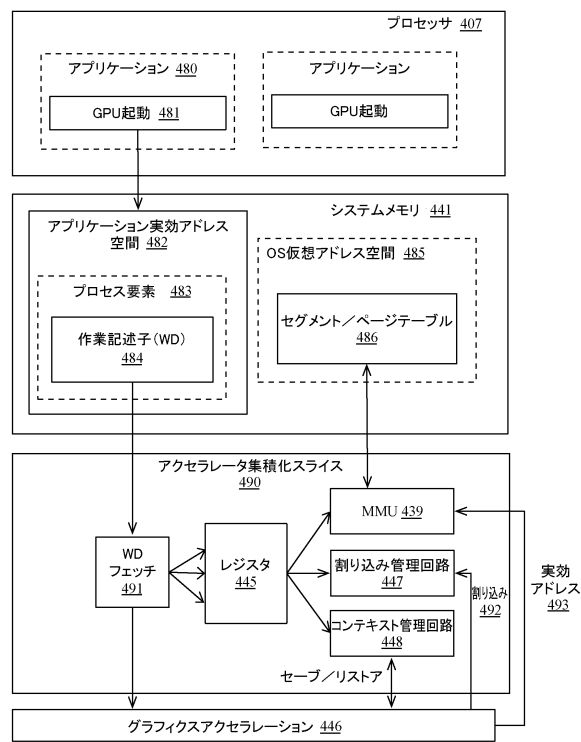
【図 4 B】



【図 4 C】



【図 4 D】



10

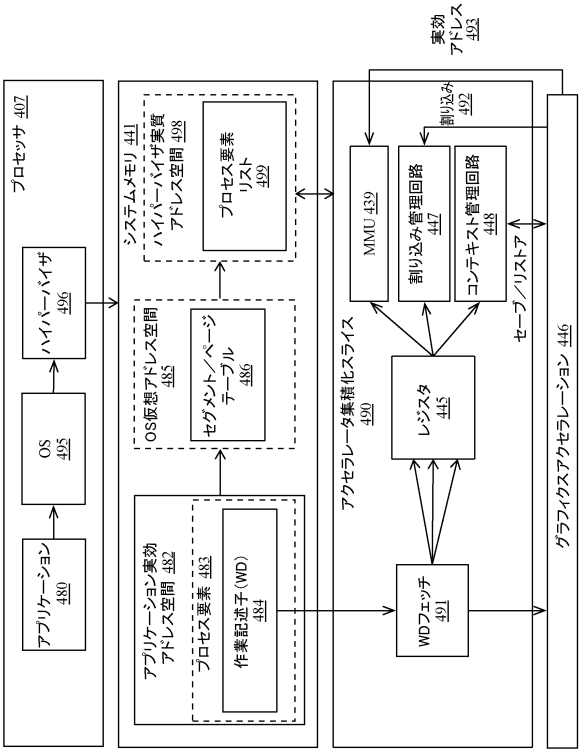
20

30

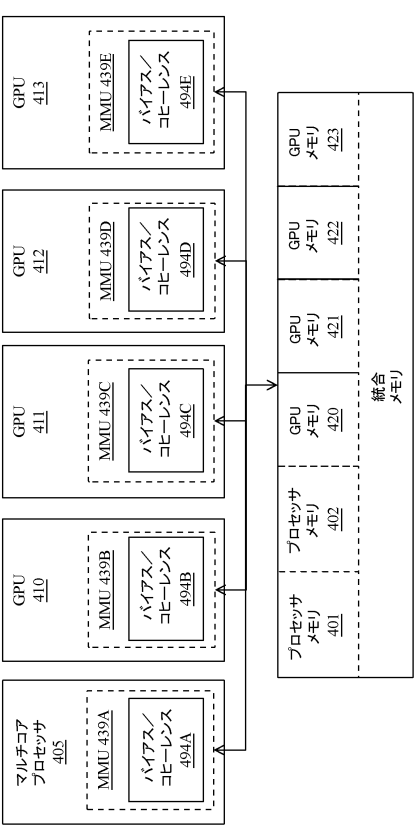
40

50

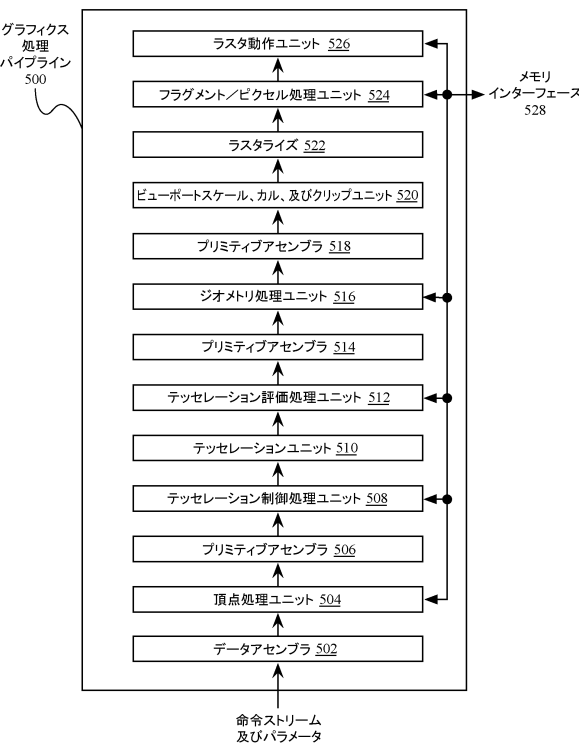
【図 4 E】



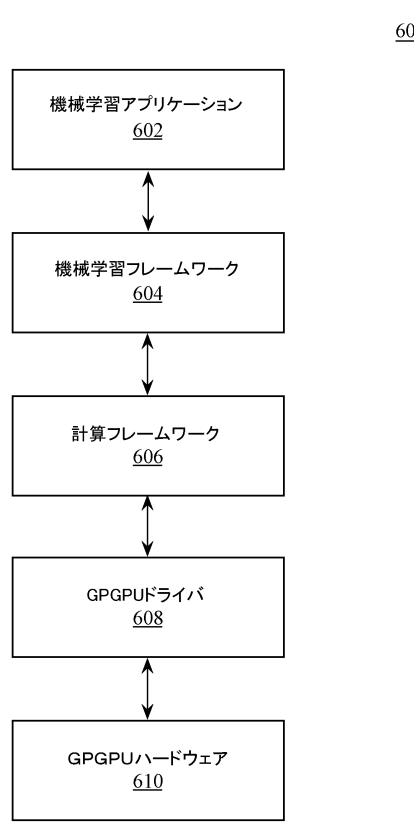
【図 4 F】



【図 5】



【図 6】



10

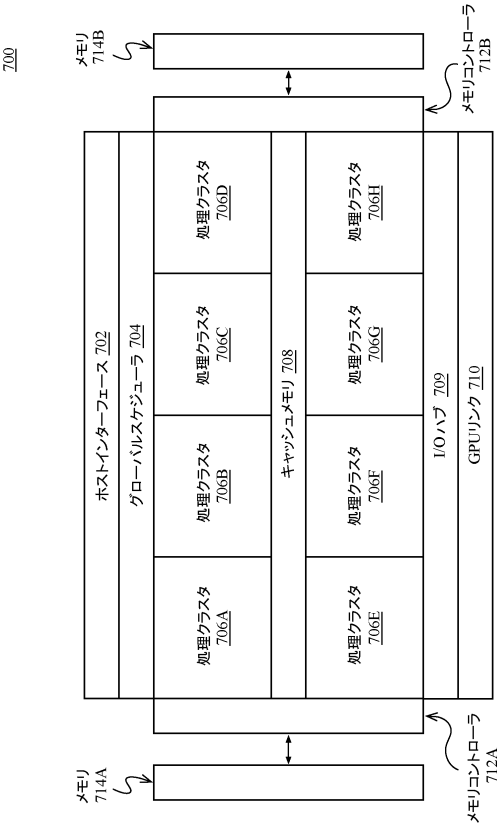
20

30

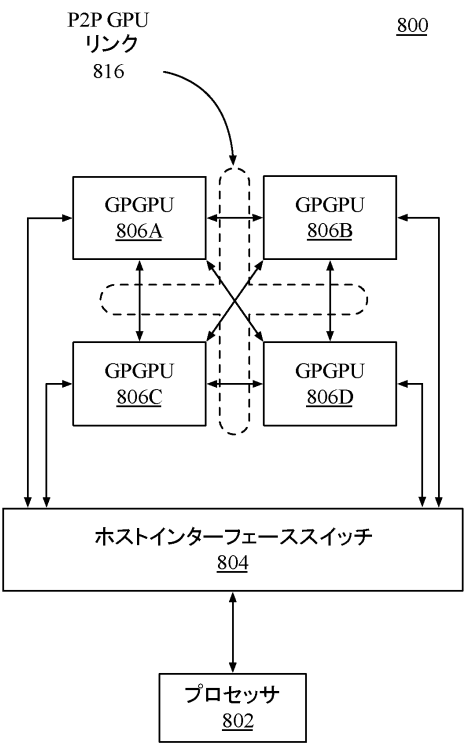
40

50

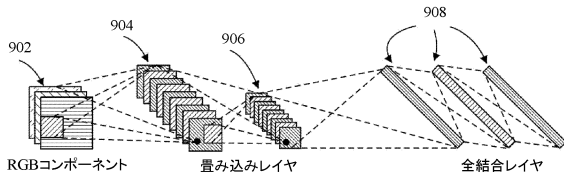
【図 7】



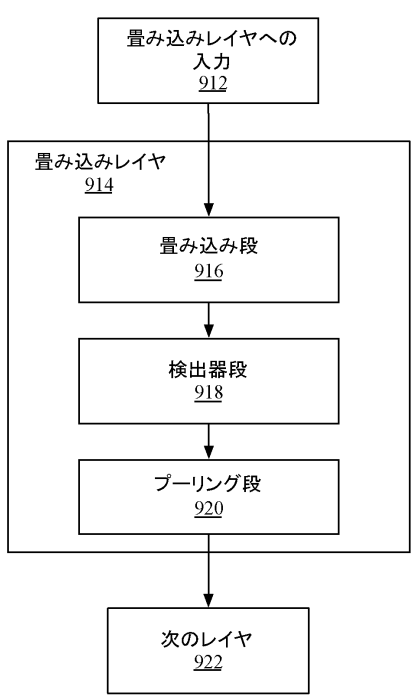
【図 8】



【図 9 A】



【図 9 B】



10

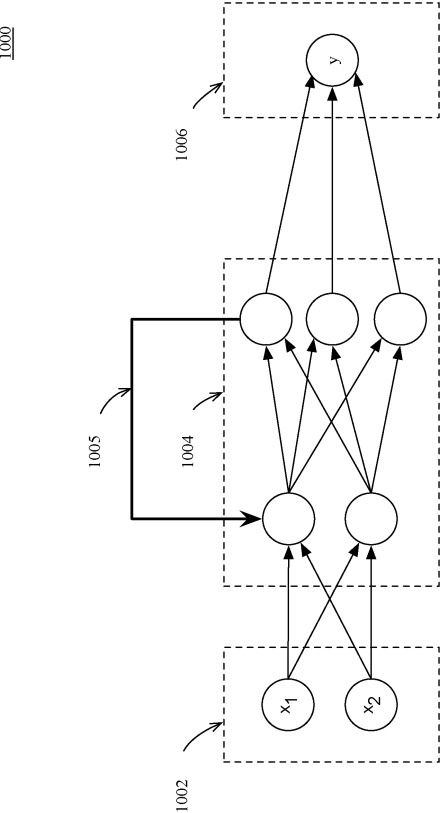
20

30

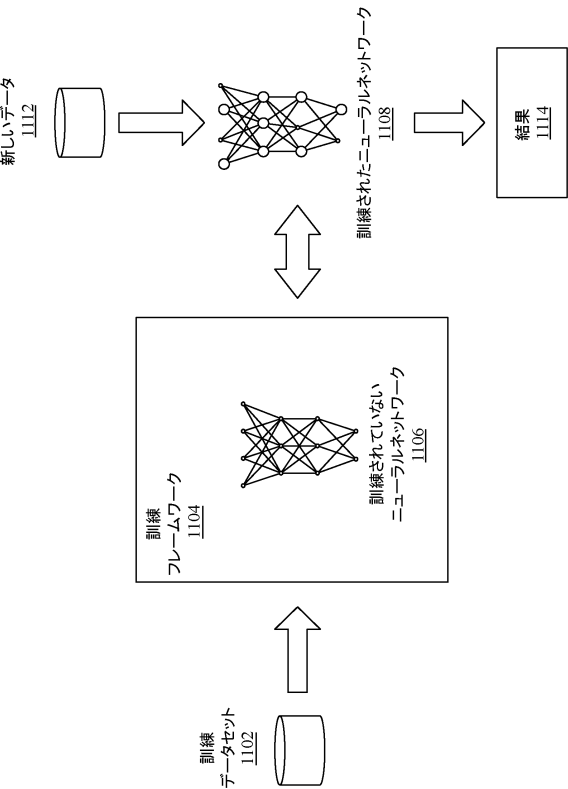
40

50

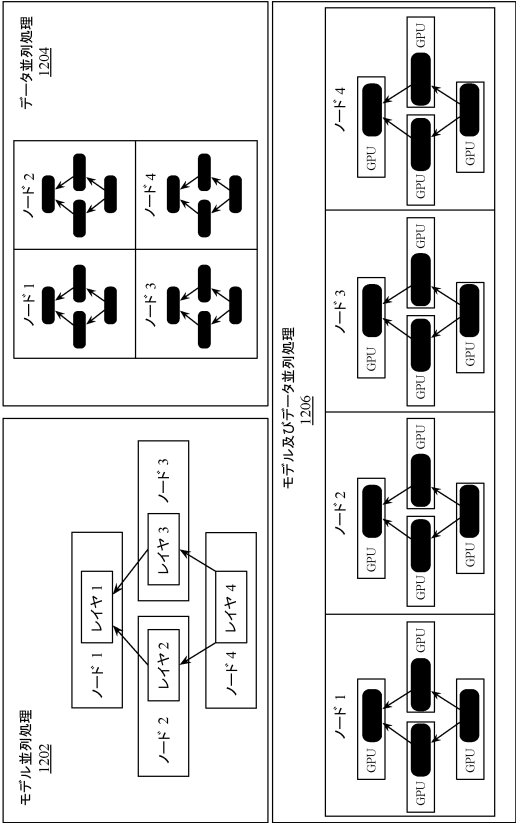
【図 1 0】



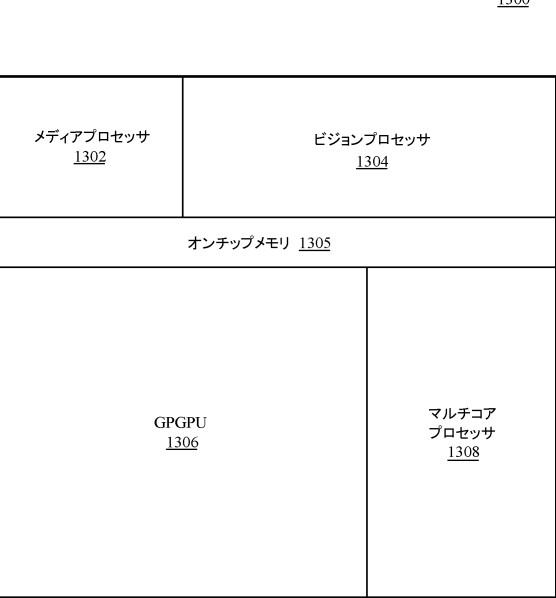
【図 1 1】



【図 1 2】



【図 1 3】



10

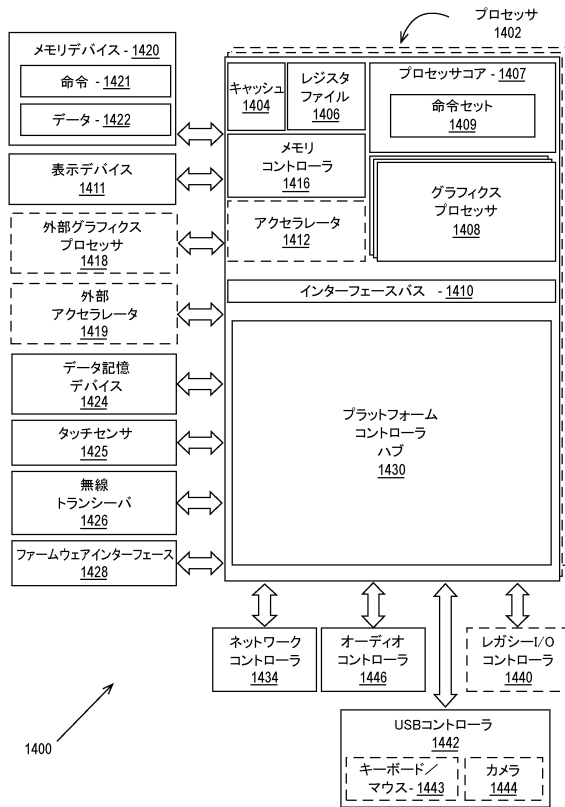
20

30

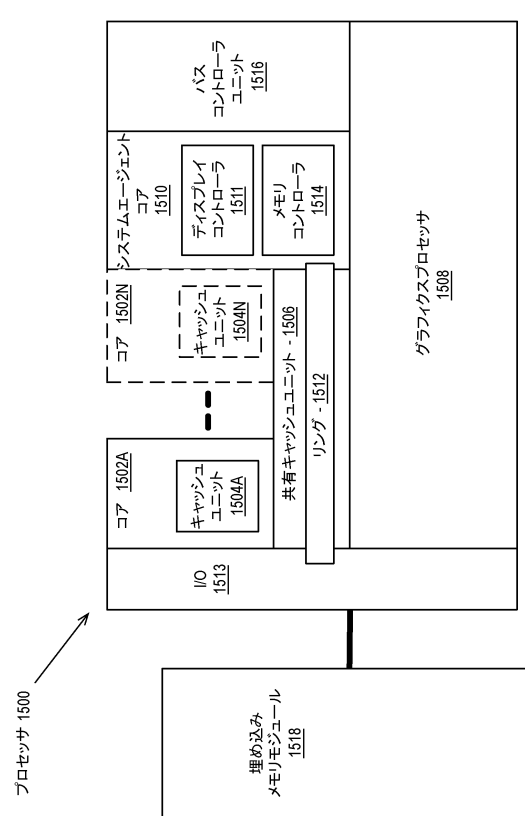
40

50

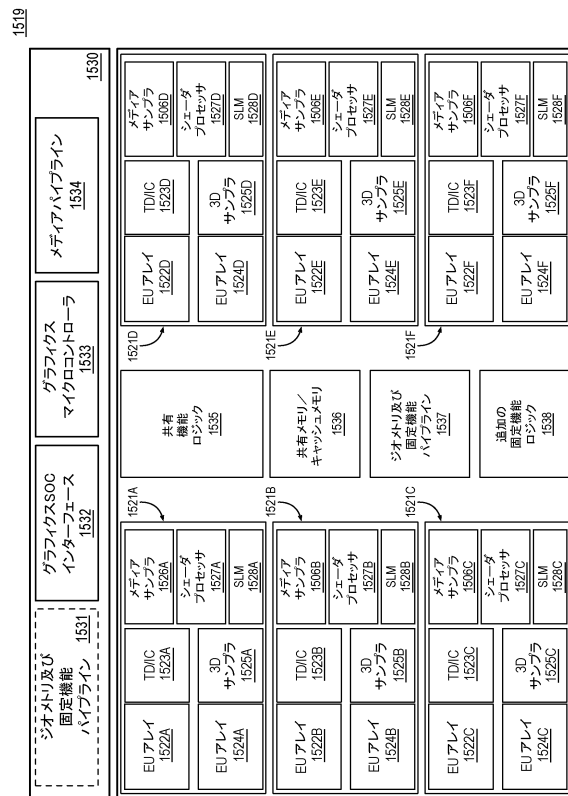
【図 14】



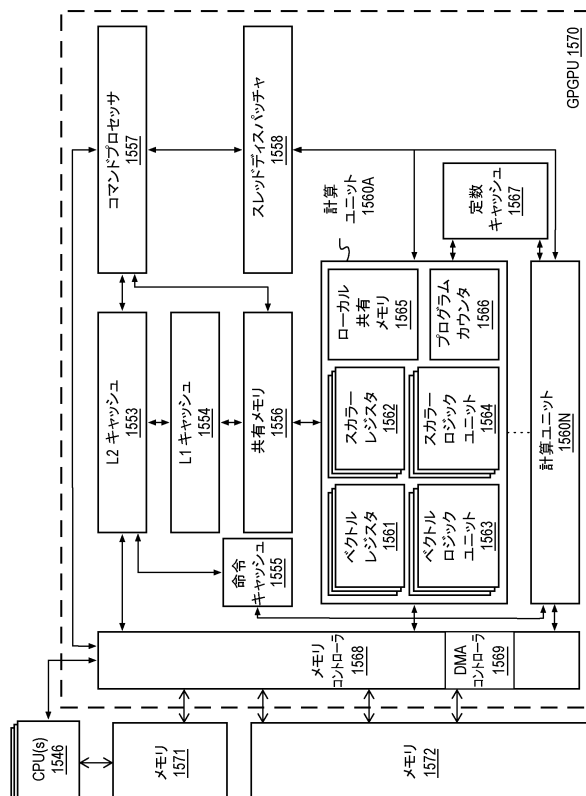
【図 15 A】



【図 15 B】



【図 15 C】



10

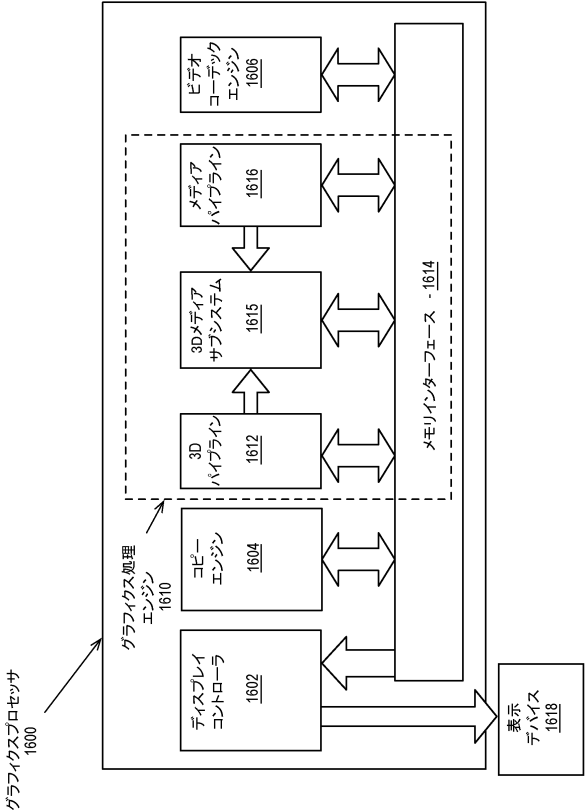
20

30

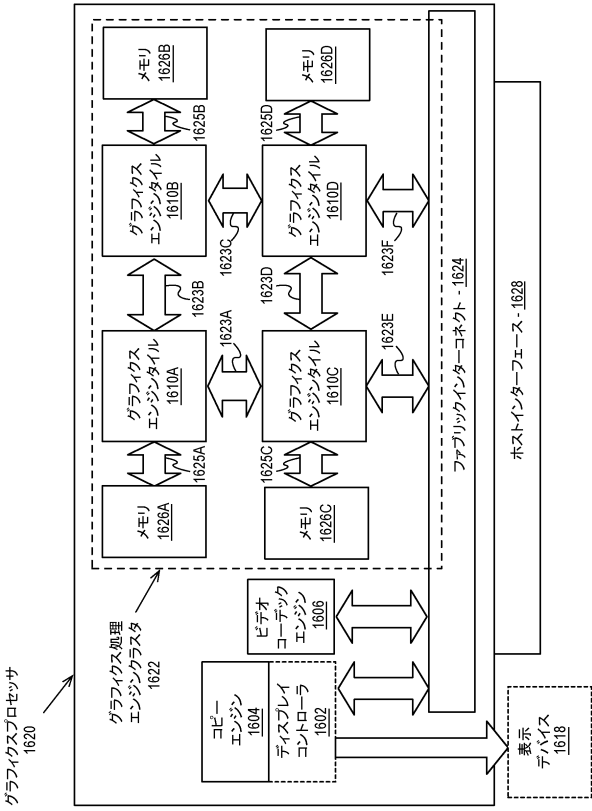
40

50

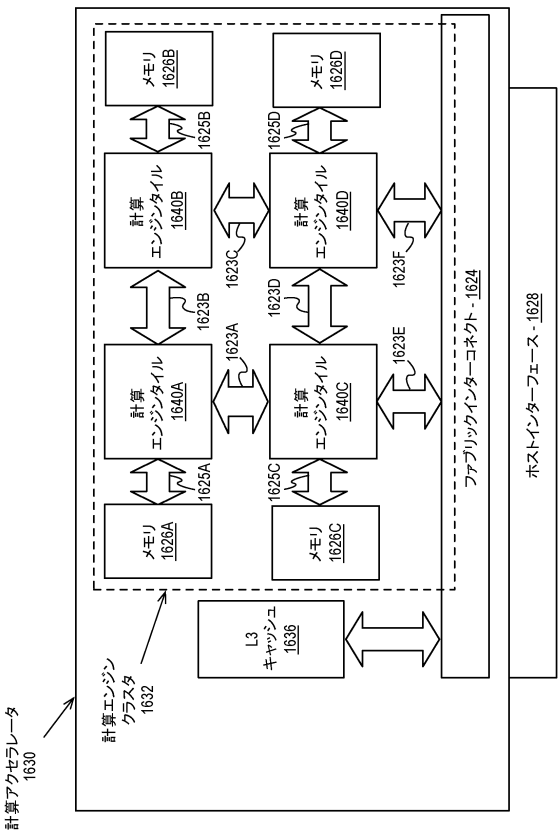
【図 1 6 A】



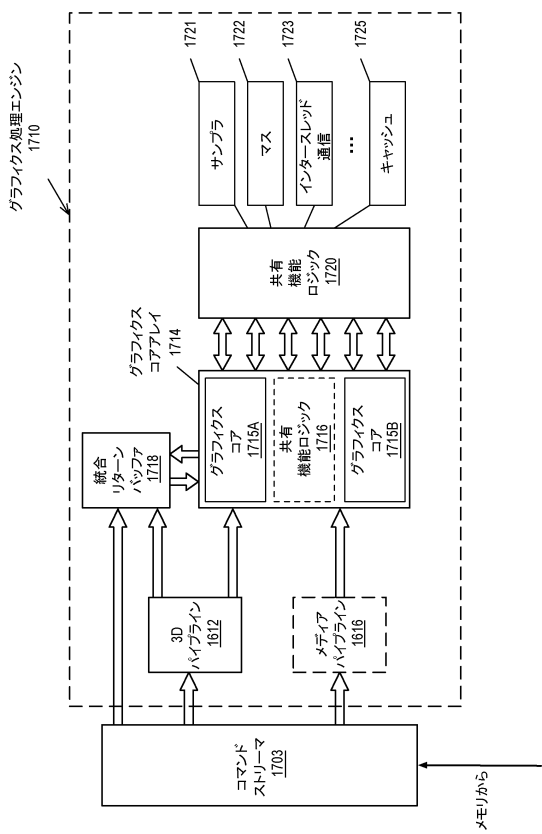
【図 1 6 B】



【図 1 6 C】



【図 1 7】



10

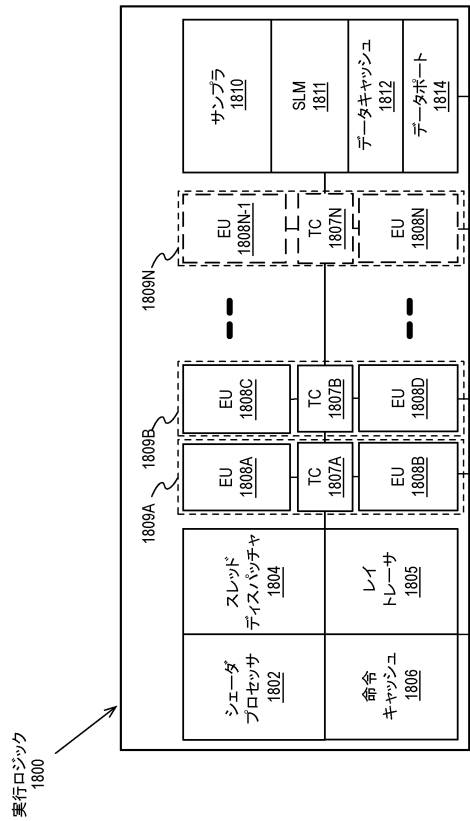
20

30

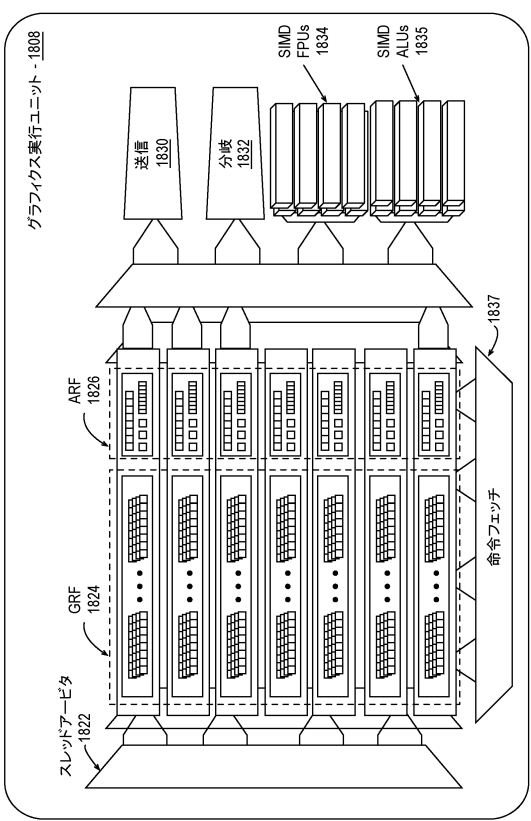
40

50

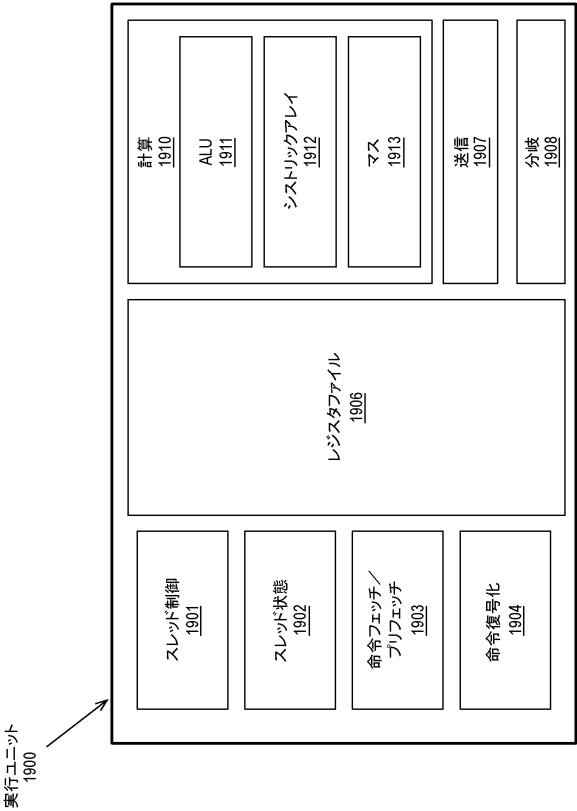
【図 1 8 A】



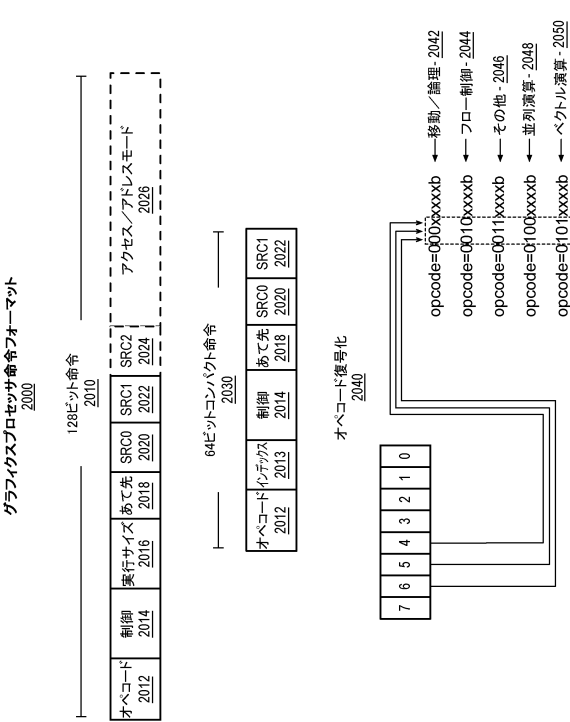
【図 1 8 B】



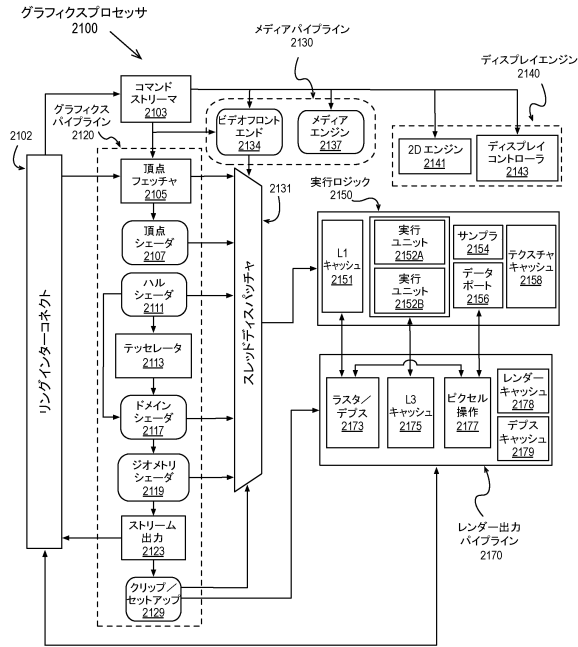
【図 1 9】



【図 2 0】



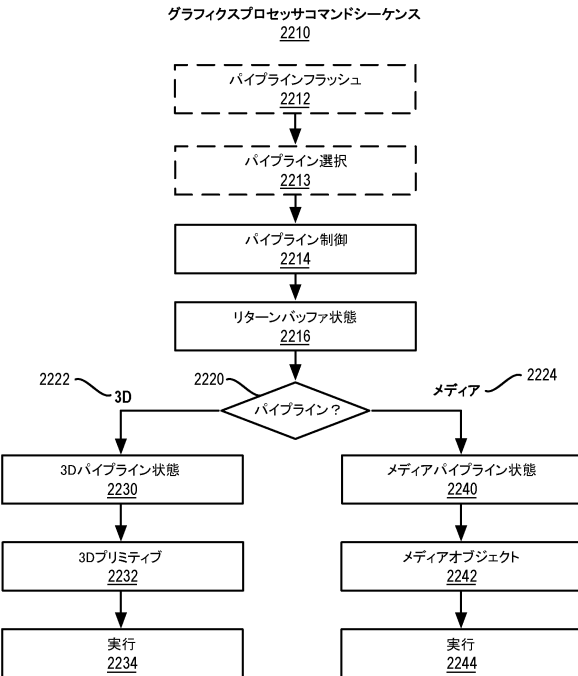
【図 2 1】



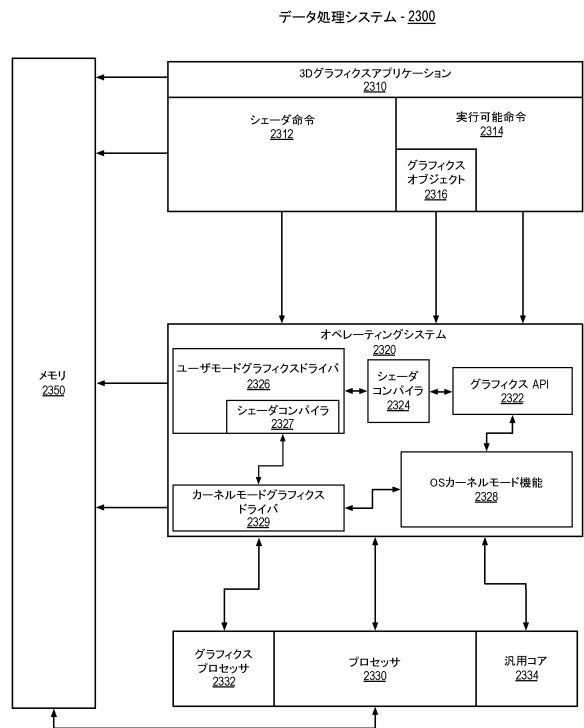
【図 2 2 A】



【図 2 2 B】



【図 2 3】



10

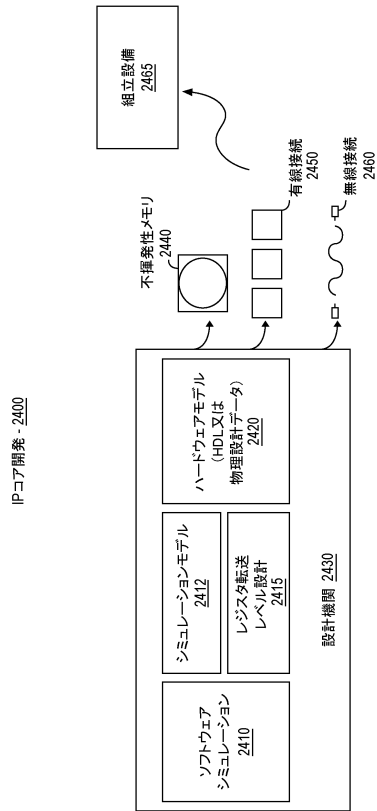
20

30

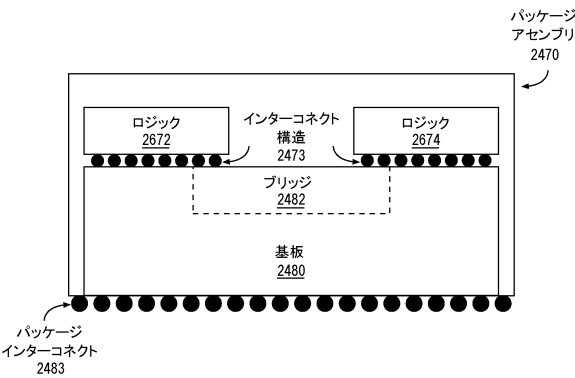
40

50

【図 2 4 A】



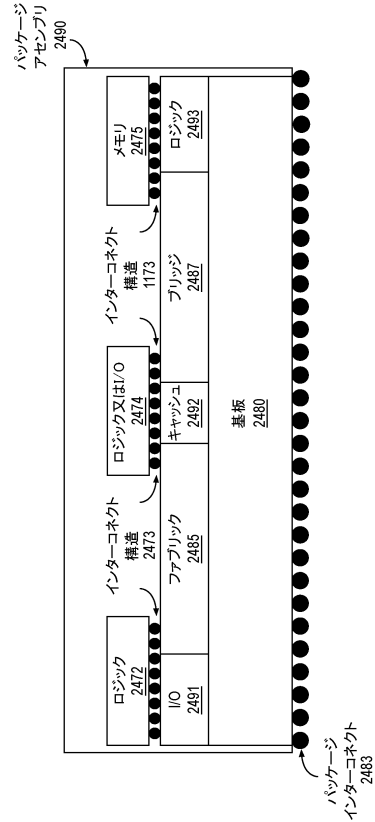
【図 2 4 B】



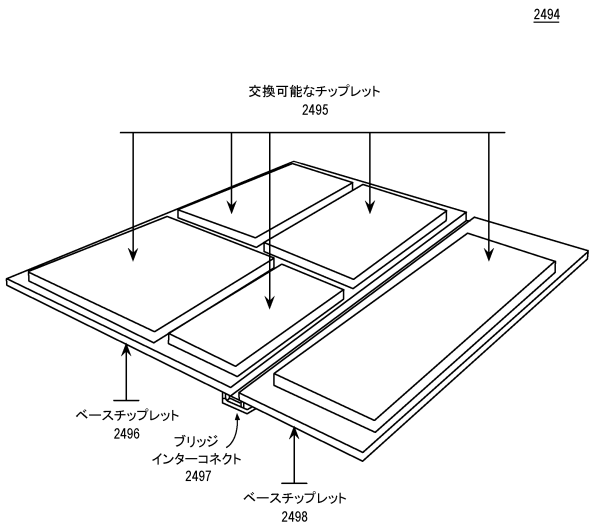
10

20

【図 2 4 C】



【図 2 4 D】

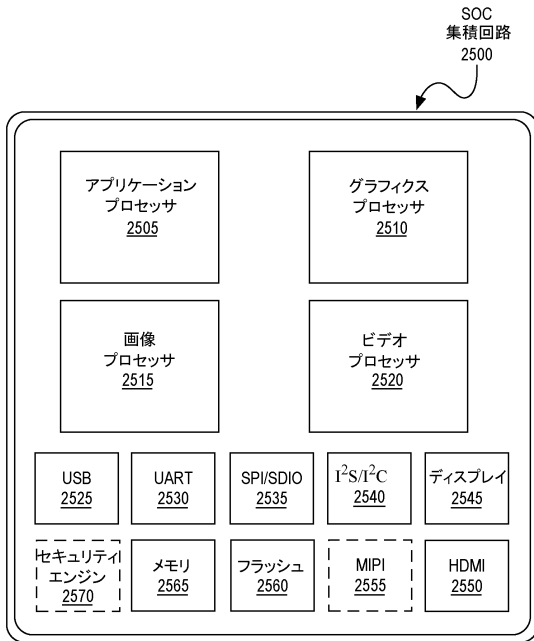


30

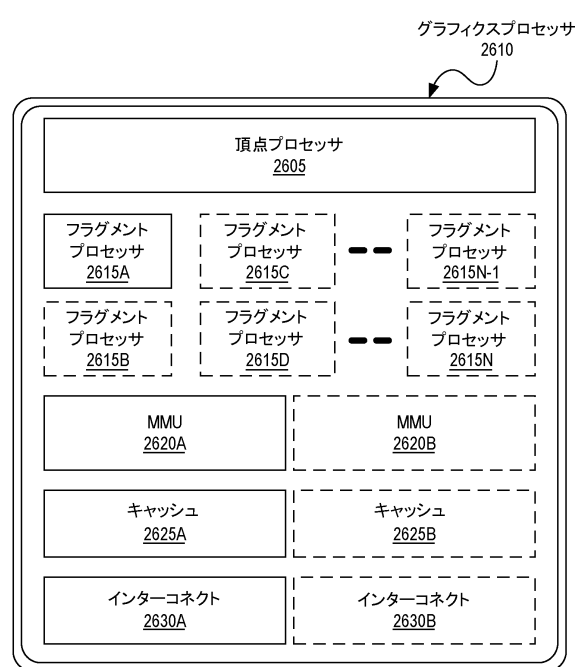
40

50

【図 2 5】



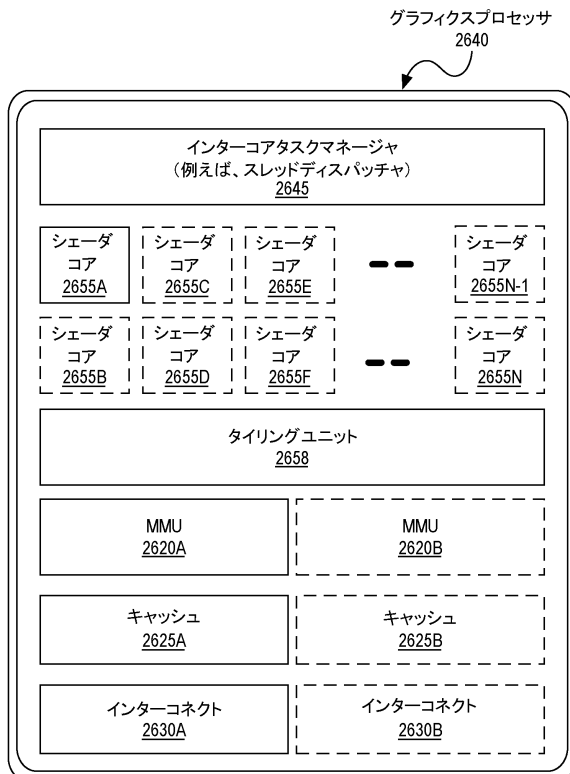
【図 2 6 A】



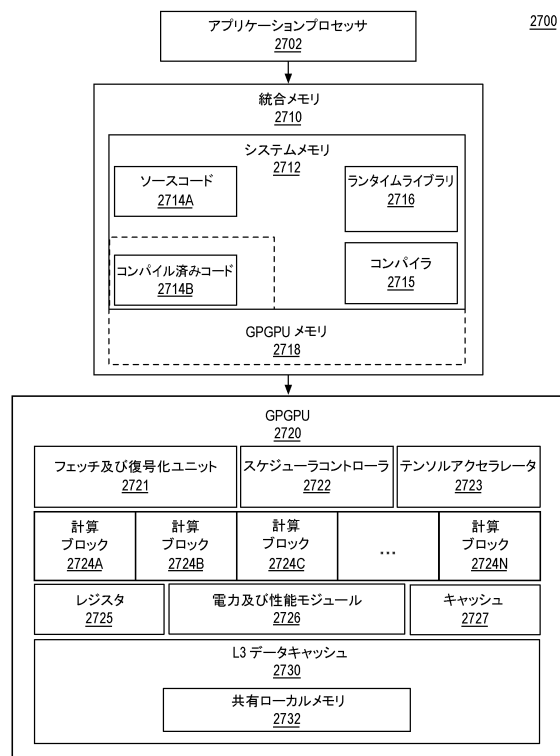
10

20

【図 2 6 B】



【図 2 7】

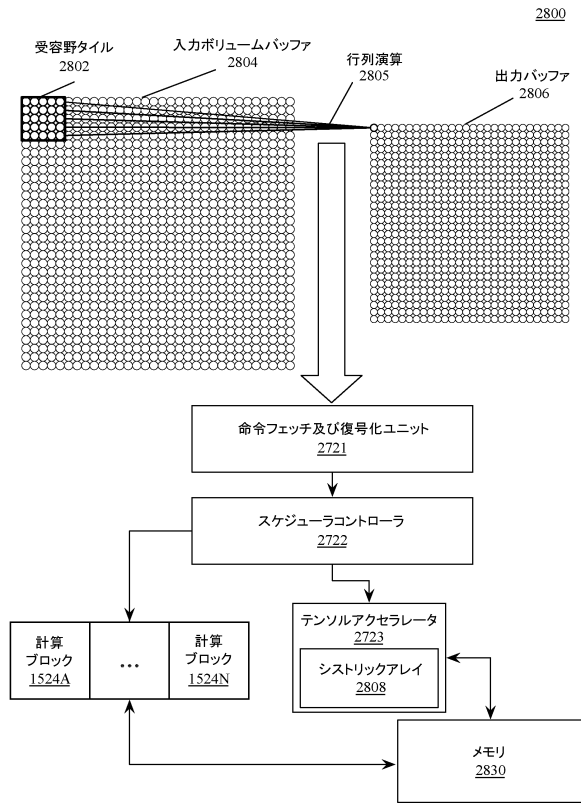


30

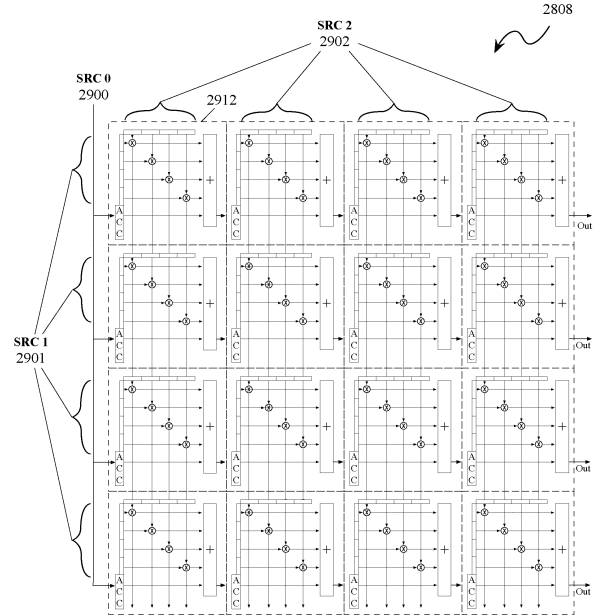
40

50

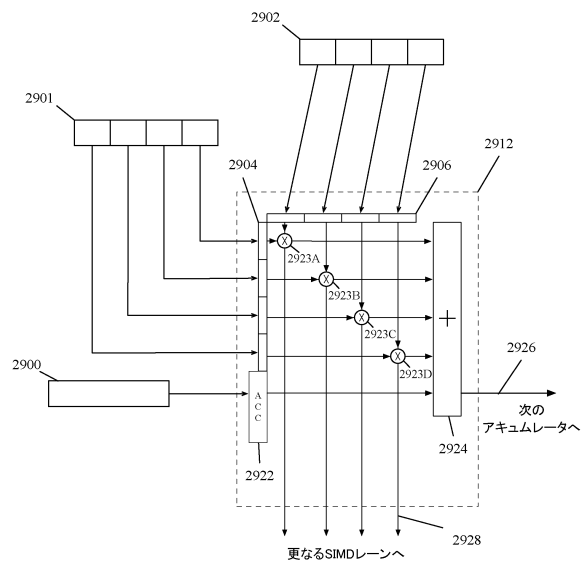
【図 28】



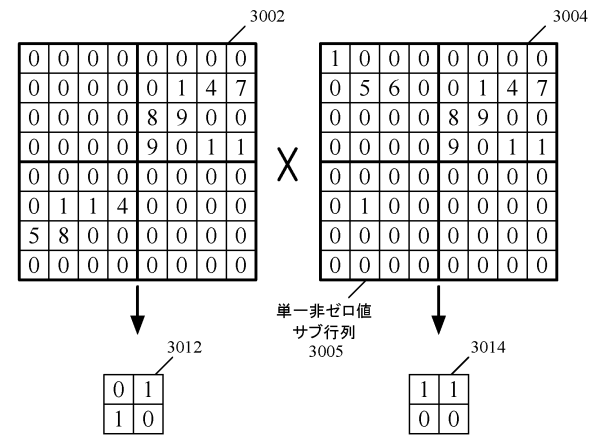
【図 29 A】



【図 29 B】



【図 30 A】



10

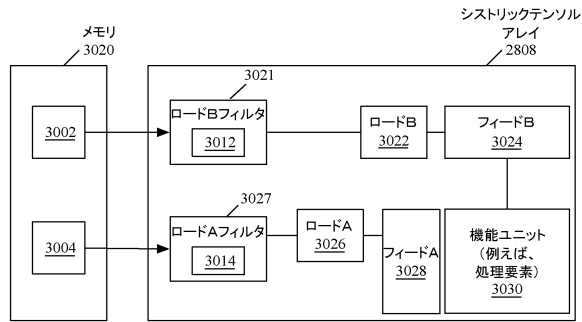
20

30

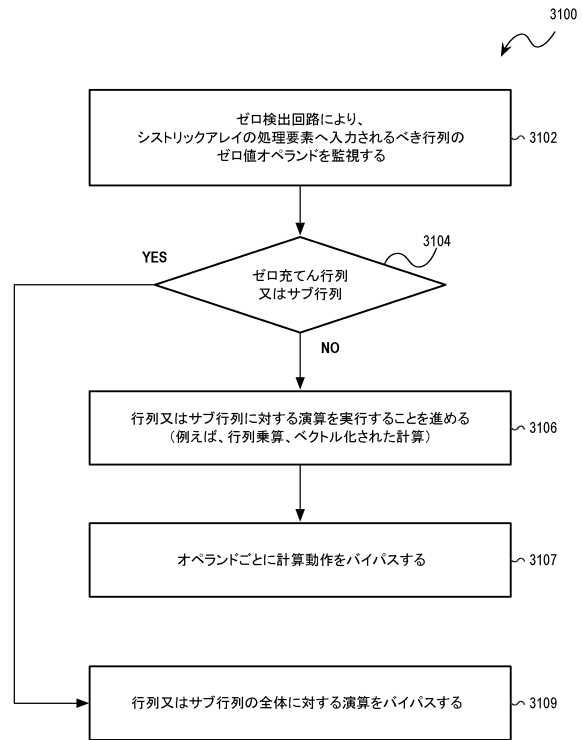
40

50

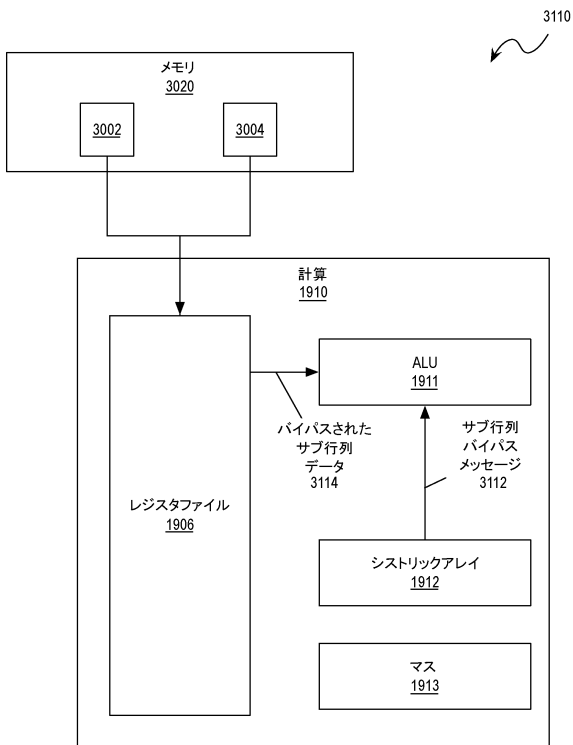
【図 3 0 B】



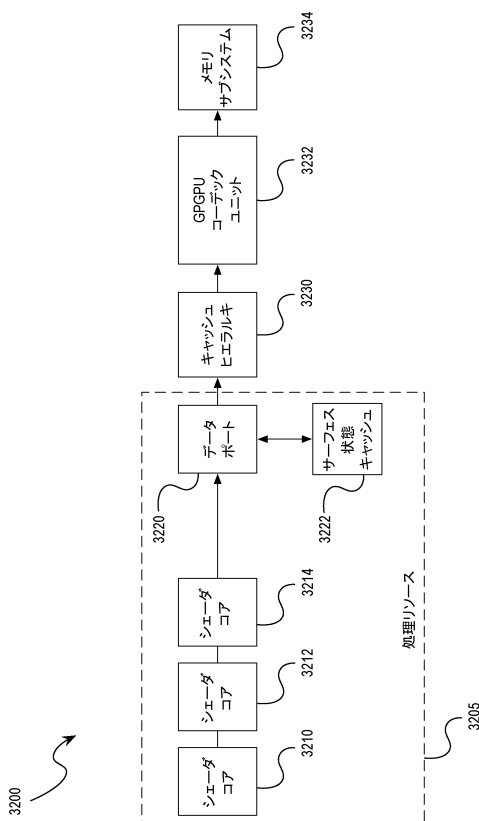
【図 3 1 A】



【図 3 1 B】



【図 3 2】



10

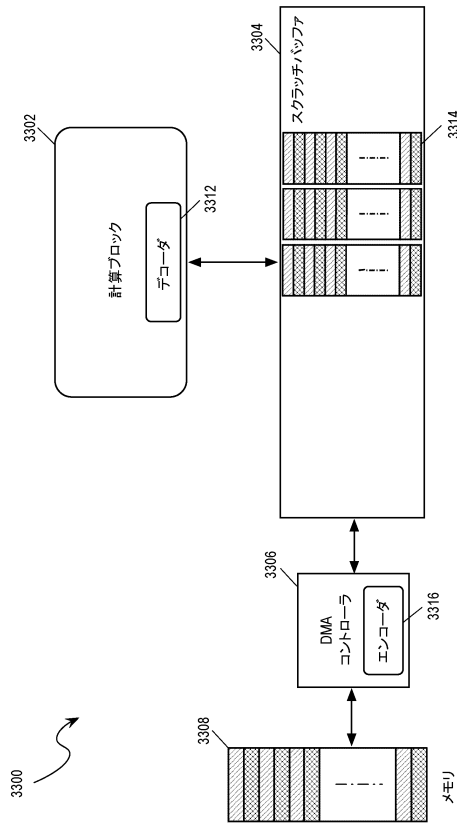
20

30

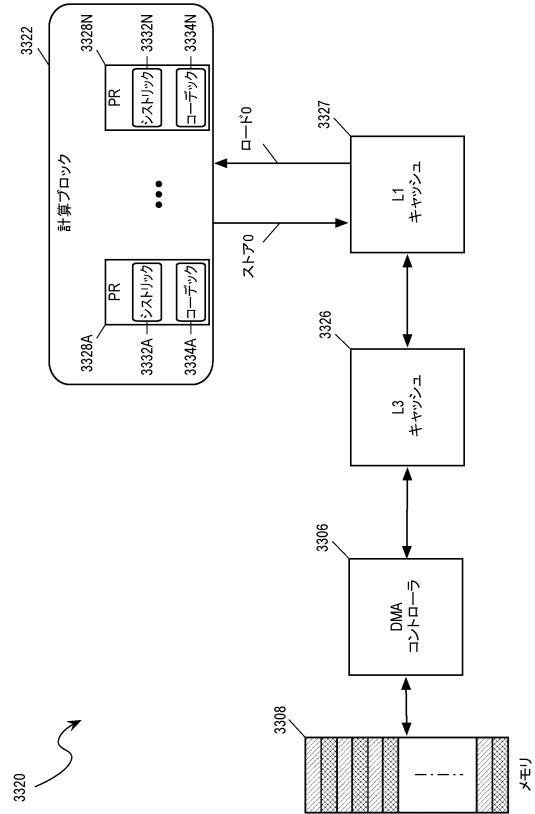
40

50

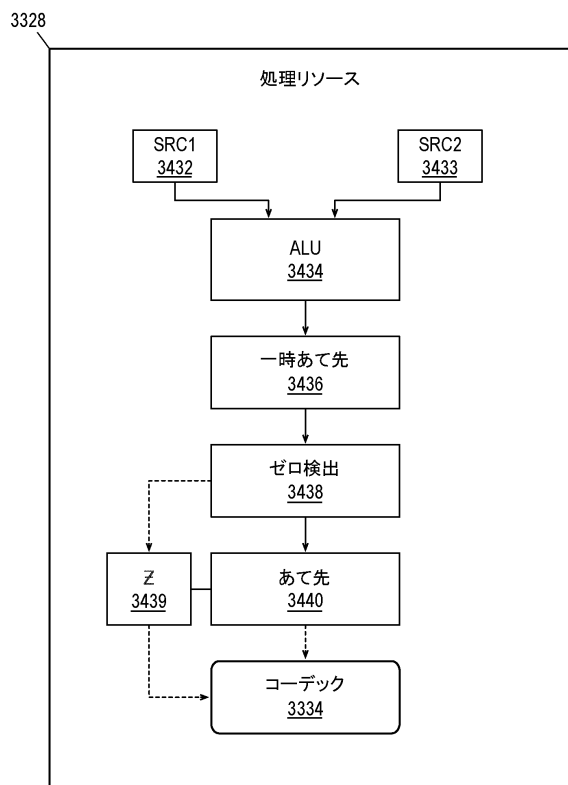
【図 3 3 A】



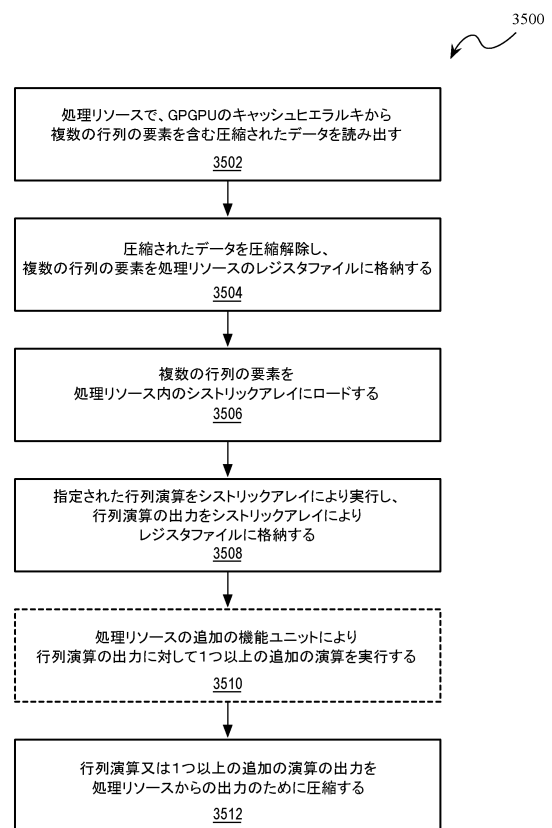
【図 3 3 B】



【図 3 4】



【図 3 5】



10

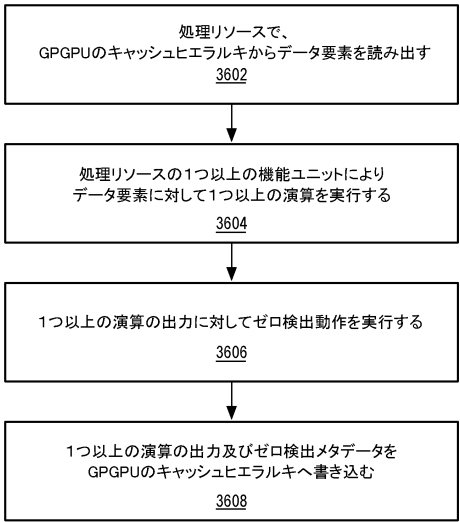
20

30

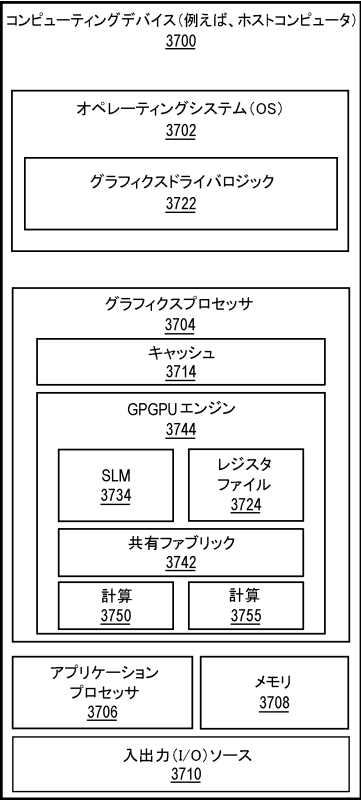
40

50

【図 3 6】



【図 3 7】



10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 62/819,361

(32)優先日 平成31年3月15日(2019.3.15)

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 62/935,670

(32)優先日 令和1年11月15日(2019.11.15)

(33)優先権主張国・地域又は機関

米国(US)

コーポレーション 内

(72)発明者 ジェイナス, スコット

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 ジョージ, ヴァーギーズ

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 マイユラン, スブラマニウム

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 コケル, アルトゥーグ

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 アップー, アビシェク

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 スルティ, プラスーンクマール

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 ランガナタン, ヴァサント

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 アンドレイ, ヴァレンティン

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 ガルグ, アシュトシュ

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 ハレル, ヨアヴ

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 ハンター, ジュニア, アーサー

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 キム, ソンイ

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード 2200 インテル コーポレーション 内

(72)発明者 マクファーソン, マイク

アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバード

-
- ド 2200 インテル コーポレーション 内
(72)発明者 オウルドーアーメドーフアル, エルモウスタファ
アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバー
ド 2200 インテル コーポレーション 内
(72)発明者 サドラー, ウィリアム
アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバー
ド 2200 インテル コーポレーション 内
(72)発明者 ストリラマッサルマ, ラクシュミナラヤナン
アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバー
ド 2200 インテル コーポレーション 内
(72)発明者 ヴェムラパッリ, ヴィクラント
アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブールバー
ド 2200 インテル コーポレーション 内
審査官 中田 剛史
(56)参考文献 国際公開第2018/125250 (WO, A1)
国際公開第2018/213636 (WO, A1)
(58)調査した分野 (Int.Cl., DB名)
G06T 15/00
G06F 9/38
G06F 17/16
G06F 17/10