

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6456867号  
(P6456867)

(45) 発行日 平成31年1月23日(2019.1.23)

(24) 登録日 平成30年12月28日(2018.12.28)

(51) Int.Cl. F I  
**G06F 9/38 (2006.01)** G06F 9/38 370C  
 G06F 9/38 310G

請求項の数 25 外国語出願 (全 43 頁)

(21) 出願番号 特願2016-93449(P2016-93449)  
 (22) 出願日 平成28年5月6日(2016.5.6)  
 (65) 公開番号 特開2017-16637(P2017-16637A)  
 (43) 公開日 平成29年1月19日(2017.1.19)  
 審査請求日 平成28年6月22日(2016.6.22)  
 (31) 優先権主張番号 14/752,047  
 (32) 優先日 平成27年6月26日(2015.6.26)  
 (33) 優先権主張国 米国(US)

(73) 特許権者 591003943  
 インテル・コーポレーション  
 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 ブレーバード・2200  
 (74) 代理人 110000877  
 龍華国際特許業務法人  
 (72) 発明者 カン、チャン ヨン  
 アメリカ合衆国 95052 カリフォル  
 ニア州・サンタクララ・ミッション カレ  
 ッジ ブレーバード・2200 インテル  
 ・コーポレーション内

最終頁に続く

(54) 【発明の名称】 密結合ヘテロジニアスコンピューティングのためのハードウェアプロセッサ及び方法

(57) 【特許請求の範囲】

【請求項1】

並列な複数の実行ユニットと、  
 前記複数の実行ユニットの複数の入力を、第1のバッファ及び複数のメモリバンクの複  
 数の出力に接続する第1のマトリクススイッチと、

並列な前記複数のメモリバンク及び複数の第2のバッファの複数の入力を、前記第1の  
 バッファ、前記複数のメモリバンク、及び前記複数の実行ユニットの複数の出力に接続す  
 る第2のマトリクススイッチと、

前記複数の第2のバッファの複数の出力に接続される複数の入力を有するオフロードエ  
 ンジンと、

を備える、ハードウェアプロセッサ。

【請求項2】

前記オフロードエンジンの出力は、前記第1のバッファの入力に接続する、請求項1に  
 記載のハードウェアプロセッサ。

【請求項3】

前記第1のバッファの出力からの読み出しと前記複数の第2のバッファの複数の入力へ  
 の書き込みとを同時に実行するデータハザード解消ロジックをさらに備える、請求項1に  
 記載のハードウェアプロセッサ。

【請求項4】

前記データハザード解消ロジックは、ストールを挿入しない、請求項3に記載のハード

ウェアプロセッサ。

【請求項 5】

前記複数の実行ユニットは、第 1 のクロック速度で実行し、前記オフロードエンジンは、第 2 のより低速のクロック速度で実行する、請求項 1 に記載のハードウェアプロセッサ。

【請求項 6】

前記複数の実行ユニットの各々は、シフトレジスタを含む、請求項 1 に記載のハードウェアプロセッサ。

【請求項 7】

前記第 1 のバッファ及び前記複数の第 2 のバッファは、ファーストインファーストアウト ( F I F O ) バッファである、請求項 1 に記載のハードウェアプロセッサ。

10

【請求項 8】

前記複数のメモリバンクは、4 つ又はそれより多くのメモリバンクであり、各メモリバンクは、複数の他のメモリバンクの複数の入力ポート及び複数の出力ポートとは別個の入力ポート及び出力ポートを含む、請求項 1 から 7 のいずれか 1 項に記載のハードウェアプロセッサ。

【請求項 9】

制御信号に基づいて、第 1 のマトリクススイッチにより、ハードウェアプロセッサの並列な複数の実行ユニットの複数の入力を、第 1 のバッファ及び複数のメモリバンクの複数の出力に接続し、第 2 のマトリクススイッチにより、並列な前記複数のメモリバンク及び複数の第 2 のバッファの複数の入力を、前記第 1 のバッファ、前記複数のメモリバンク、及び前記複数の実行ユニットの複数の出力に接続する段階と、

20

前記複数の第 2 のバッファの複数の出力からオフロードエンジンの複数の入力にデータを提供する段階と、

を備える、方法。

【請求項 10】

前記オフロードエンジンの出力から前記第 1 のバッファの入力にデータを提供する段階をさらに備える、請求項 9 に記載の方法。

【請求項 11】

前記第 1 のバッファの出力からの読み出しと前記複数の第 2 のバッファの複数の入力への書き込みとを同時に実行する段階をさらに備える、請求項 9 に記載の方法。

30

【請求項 12】

ストールを挿入しない段階をさらに備える、請求項 11 に記載の方法。

【請求項 13】

前記複数の実行ユニットによって第 1 のクロック速度で実行し、前記オフロードエンジンによって第 2 のより低速のクロック速度で実行する段階をさらに備える、請求項 9 に記載の方法。

【請求項 14】

前記複数の実行ユニットの各々は、シフトレジスタを含む、請求項 9 に記載の方法。

【請求項 15】

40

前記第 1 のバッファ及び前記複数の第 2 のバッファは、ファーストインファーストアウト ( F I F O ) バッファである、請求項 9 に記載の方法。

【請求項 16】

前記複数のメモリバンクは、4 つ又はそれより多くのメモリバンクであり、各メモリバンクは、複数の他のメモリバンクの複数の入力ポート及び複数の出力ポートとは別個の入力ポート及び出力ポートを含む、請求項 9 から 15 のいずれか 1 項に記載の方法。

【請求項 17】

ハードウェアプロセッサであって、  
命令を復号するハードウェアデコーダと、  
前記命令を実行することにより、

50

制御信号に基づいて、第 1 のマトリクススイッチにより、前記ハードウェアプロセッサの並列な複数の実行ユニットの複数の入力を、第 1 のバッファ及び複数のメモリバンクの複数の出力に接続し、第 2 のマトリクススイッチにより、並列な前記複数のメモリバンク及び複数の第 2 のバッファの複数の入力を、前記第 1 のバッファ、前記複数のメモリバンク、及び前記複数の実行ユニットの複数の出力に接続し、

前記複数の第 2 のバッファの複数の出力からオフロードエンジンの複数の入力にデータを提供する

ハードウェア実行ユニットと、  
を備える、ハードウェアプロセッサ。

【請求項 18】

前記オフロードエンジンの出力は、前記第 1 のバッファの入力に接続する、請求項 17 に記載のハードウェアプロセッサ。

【請求項 19】

前記ハードウェア実行ユニットは、前記命令を実行し、これにより、前記第 1 のバッファの出力からの読み出しと、前記複数の第 2 のバッファの複数の入力への書き込みとを同時に実行させる、請求項 17 に記載のハードウェアプロセッサ。

【請求項 20】

前記ハードウェア実行ユニットは、ストールを挿入することなく前記命令を実行する、請求項 19 に記載のハードウェアプロセッサ。

【請求項 21】

前記複数の実行ユニットは、第 1 のクロック速度で実行し、前記オフロードエンジンは、第 2 のより低速のクロック速度で実行する、請求項 17 に記載のハードウェアプロセッサ。

【請求項 22】

前記複数の実行ユニットの各々は、シフトレジスタを含む、請求項 17 に記載のハードウェアプロセッサ。

【請求項 23】

前記第 1 のバッファ及び前記複数の第 2 のバッファは、ファーストインファーストアウト (FIFO) バッファである、請求項 17 に記載のハードウェアプロセッサ。

【請求項 24】

前記複数のメモリバンクは、4 つ又はそれより多くのメモリバンクであり、各メモリバンクは、複数の他のメモリバンクの複数の入力ポート及び複数の出力ポートとは別個の入力ポート及び出力ポートを含む、請求項 17 から 23 のいずれか 1 項に記載のハードウェアプロセッサ。

【請求項 25】

並列な複数の実行ユニットと、

第 1 のマトリクススイッチにより、前記複数の実行ユニットの複数の入力を、第 1 のバッファ及び複数のメモリバンクの複数の出力に接続し、第 2 のマトリクススイッチにより、並列な前記複数のメモリバンク及び複数の第 2 のバッファの複数の入力を、前記第 1 のバッファ、前記複数のメモリバンク、及び前記複数の実行ユニットの複数の出力に接続する手段と、

前記複数の第 2 のバッファの複数の出力に接続される複数の入力を有するオフロードエンジンと、

を備える、ハードウェア装置。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、概して電子分野に関し、より詳細には、本開示の実施形態は、密結合ヘテロジニアスコンピューティングのためのハードウェアプロセッサに関する。

【背景技術】

10

20

30

40

50

## 【 0 0 0 2 】

プロセッサ又は複数のプロセッサのセットは、命令セット、例えば命令セットアーキテクチャ ( I S A ) からの複数の命令を実行する。命令セットは、プログラミングに関するコンピュータアーキテクチャの一部であり、概して、複数のネイティブデータ型、複数の命令、レジスタアーキテクチャ、複数のアドレス指定モード、メモリアーキテクチャ、割り込み及び例外処理ならびに外部入力及び出力 ( I / O ) を含む。

## 【 図面の簡単な説明 】

## 【 0 0 0 3 】

本開示は、添付図面の複数の図において、限定としてではなく例として示されるものであり、ここで、同様の参照符号は、同様の要素を示す。

【 図 1 】本開示の複数の実施形態に係るハードウェアプロセッサを示す。

【 図 2 A 】本開示の複数の実施形態に係るハードウェアプロセッサを通して、複数のデータパスを示す。

【 図 2 B 】本開示の複数の実施形態に係る図 2 A におけるハードウェアプロセッサを通る複数のアクティブなデータパスを示す。

【 図 3 】本開示の複数の実施形態に係るデータパスを通して移行する複数のベクトルを示す。

【 図 4 】本開示の複数の実施形態に係るフロー図を示す。

【 図 5 A 】本開示の複数の実施形態に係る汎用ベクトル向け命令フォーマット及びこれらのクラス A 命令テンプレートを示すブロック図である。

【 図 5 B 】本開示の複数の実施形態に係る汎用ベクトル向け命令フォーマット及びこれらのクラス B 命令テンプレートを示すブロック図である。

【 図 6 A 】本開示の複数の実施形態に係る図 5 A 及び 5 B における汎用ベクトル向け命令フォーマットの複数のフィールドを示すブロック図である。

【 図 6 B 】本開示の一実施形態に係るフルオペコードフィールドを形成する図 6 A における特定ベクトル向け命令フォーマットの複数のフィールドを示すブロック図である。

【 図 6 C 】本開示の一実施形態に係るレジスタインデックスフィールドを形成する図 6 A における特定ベクトル向け命令フォーマットの複数のフィールドを示すブロック図である。

。

【 図 6 D 】本開示の一実施形態に係る増加オペレーションフィールド 5 5 0 を形成する図 6 A における特定ベクトル向け命令フォーマットの複数のフィールドを示すブロック図である。

【 図 7 】本開示の一実施形態に係るレジスタアーキテクチャのブロック図である。

【 図 8 A 】本開示の複数の実施形態に係る例示的なインオーダーパイプライン及び例示的なレジスタリネーミング、アウトオブオーダー発行 / 実行パイプラインの両方を示すブロック図である。

【 図 8 B 】本開示の複数の実施形態に係るプロセッサに含まれるべきインオーダーアーキテクチャコアの例示的な実施形態及び例示的なレジスタリネーミング、アウトオブオーダー発行 / 実行アーキテクチャコアの両方を示すブロック図である。

【 図 9 A 】本開示の複数の実施形態に係る単一のプロセッサコアを、そのオンダイ相互接続ネットワークへの接続及びその二次 ( L 2 ) キャッシュのローカルサブセットと共に示すブロック図である。

【 図 9 B 】本開示の複数の実施形態に係る図 9 A におけるプロセッサコアの一部の拡大図である。

【 図 1 0 】 1 つより多くのコアを有してよく、集積メモリコントローラを有してよく、集中画像表示を有してよい、本開示の複数の実施形態に係るプロセッサのブロック図である。

。

【 図 1 1 】本開示の一実施形態に係るシステムのブロック図である。

【 図 1 2 】本開示の実施形態に係る、より具体的な例示的システムのブロック図である。

【 図 1 3 】本開示の実施形態に係る、第 2 のより具体的な例示的システムのブロック図を

10

20

30

40

50

示す。

【図14】本開示の実施形態に係るシステムオンチップ（SoC）のブロック図を示す。

【図15】本開示の複数の実施形態に係るソース命令セットの複数のバイナリ命令をターゲット命令セットの複数のバイナリ命令に変換するソフトウェア命令コンバータの使用を対比したブロック図である。

【発明を実施するための形態】

【0004】

以下の説明において、多数の具体的な詳細が示される。しかしながら、本開示の複数の実施形態は、これらの具体的な詳細がなくとも実施可能であることを理解されたい。他の複数の例において、周知の複数の回路、構造及び技術は、この説明に対する理解を曖昧にしないよう、詳細には示されていない。

10

【0005】

本明細書における「一実施形態」、「実施形態」、「例示的な実施形態」等の記載は、説明される実施形態が、特定の機能、構造、又は特性を含んでよいが、各実施形態がその特定の機能、構造、又は特性を必ずしも含まなくてよいことを示す。さらに、このような複数の語句は、必ずしも同じ実施形態を参照するものではない。さらに、ある実施形態に関連して特定の機能、構造、又は特性が説明される場合、明示的に説明されているか否かに関わらず、複数の他の実施形態に関連してこのような機能、構造、又は特性に影響を与えることは、当業者の知識の範囲内であると考えられる。

【0006】

20

（例えばハードウェア）プロセッサ又はプロセッサのセットは、命令セット、例えば命令セットアーキテクチャ（ISA）からの複数の命令を実行する。命令セットは、プログラミングに関するコンピュータアーキテクチャの一部であり、概して、複数のネイティブデータ型、複数の命令、レジスタアーキテクチャ、複数のアドレス指定モード、メモリアーキテクチャ、割り込み及び例外処理ならびに外部入力及び出力（I/O）を含む。本明細書において、命令という用語は、マクロ命令、例えば、実行のためにプロセッサに提供される命令、又は、マイクロ命令、例えば、複数のマクロ命令を復号するプロセッサの復号ユニット（デコーダ）から得られた命令を意味してよいことに留意されたい。プロセッサ（例えば、複数の命令を復号及び/又は実行する1つ又は複数のコアを有するもの）は、例えば、演算、ロジック、又は他の複数の機能の実行において、データに対して動作してよい。

30

【0007】

複数の特定の機能は、複数のベクトル（例えば、各エレメントを特定する対応インデックスを有する複数のデータエレメント（エントリ）のアレイ又は他の順序のリスト）、例えば、複数の他のベクトルに対して動作するベクトルに対するオペレーションを含んでよい。ベクトルのエレメントは、概して、単一の値、例えば、それ自身のインデックス値によって特定される値を表すデータの別個のセクションを意味してよい。ベクトルの複数のエレメントは、数値（例えば、整数、浮動小数点等）であってよい。一実施形態において、ベクトルは、単一の数値、例えば単一の浮動小数点数を表す複数のエレメントを含む。

【0008】

40

プロセッサは、（例えば、プロセッサダイとは別個の）メモリ（例えば、データストレージデバイス）のデータにアクセス（例えば、ロード及び/又はストア）してよい。メモリは、システムメモリ、例えば、ランダムアクセスメモリ（RAM）であってよい。データストレージデバイスは、プロセッサキャッシュを含まなくてよく、及び/又は、限定されるものではないが、ハードディスクドライブ（HDD）ストレージのような外部ストレージを含まなくてよい。

【0009】

プロセッサは、特定のオペレーション（又はオペレーションの一部）をオフロードエンジンにオフロードしてよい。ハードウェアオフロードエンジンは、プロセッサとは別個であってよく（例えば、オフダイ）、又はプロセッサの一部、例えば、コアの一部又はコア

50

とは別個であってよい。オフロードエンジンを用いるプロセッサ（例えば、プロセッサコア）は、その命令セットにおける命令（例えば、特定のオペコードを有するもの）を含んでよく、命令セットは、例えば、当該命令が実行される場合に、データをハードウェアオフロードエンジンに送信させる。一実施形態において、オフロードエンジンは、例えば、本明細書において説明されるように、そのオペレーションを制御する有限ステートマシン（FSM）を含む。一実施形態において、プロセッサコア（例えば、複数の（例えばベクトル）実行ユニット）は、第1のクロック速度で実行してよく、オフロードエンジンは、第2のより高速（又はより低速）のクロック速度で（例えば同時に）実行してよい。一実施形態において、オフロードエンジンは、プロセッサのクロック速度の倍より大きい速度で動作する。一実施形態において、オフロードエンジンは、プロセッサのクロック速度の2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、又は25倍等、より高速（又はより低速）の速度で動作する。オフロードエンジンは、オフロードエンジンを用いるプロセッサと異なる型のプロセッサであってよい。一実施形態において、プロセッサ及びオフロードエンジンは、同じ命令セット、例えばISAを用いる。

#### 【0010】

一実施形態において、プロセッサは、領域が特化された問題に対処する特定用途向けオフロードエンジンを含む。ハードウェアのみ及びソフトウェアのみのアプローチと比較して、ハイブリッドプロセッサアーキテクチャの複数の特定の形態は、ソフトウェア（例えば、プロセッサ上で動作するファームウェア）の柔軟性と、最適化されたハードウェア（例えば、特定用途向けオフロードエンジン）の効率性との両方を提供することができる。例えば、公開鍵暗号化（PKE）の一実施形態において、回路（例えば、コンテンツ処理モジュール（CPM））は、（例えば、プログラマブル）ベクトルプロセッサ（例えば、モジュラー計算プロセッサ（MMP））と、（例えば大型）整数乗算器（ $521b \times 521b$ ）オフロードエンジンとを含んでよい。本実施形態は、オフロードエンジンのモデルの非同期利用を含んでよく、例えば、これにより、複数のオペランドのソーシング及び結果のドレインは、2つの（例えばファームウェア）ベクトル命令の実行を含む。複数の命令のこのようなシケンシャルな実行モデルは、オフロードエンジンの利用、従って、システムのスループット全体を低減することができる。（例えば高）並列データパスを有するこのようなベクトルプロセッサ及びオフロードエンジンは、データハザード解消ロジックの利用を含んでもよい。

#### 【0011】

本開示の一実施形態において超長命令語（VLIW）型の命令に対するベクトルプロセッサ（例えば、ベクトルプロセッサアーキテクチャ）は、例えば、プロセッサ（例えばコア）によってオフロードエンジンに送信されたデータ及び/又はオフロードエンジンによってプロセッサに送信されたデータの、プロセッサ自体の内部における、又はオフロードエンジンへ及び/又はオフロードエンジンからの（例えば高）並列データ移動を可能とする。VLIW命令は（例えば、実行された場合に）、複数の（例えば、別個の又は異なる）複数のオペレーションを生じさせてよい。例えば、1つのVLIW命令は（例えば、実行された場合に）、1つの（例えば、異なる）複数のオペレーションをプロセッサの複数の実行ユニットの各々に対して実行させてよい。

#### 【0012】

本開示の複数の特定の形態は、スイッチ（又は複数のスイッチ）と、複数の並列データオペレーションを可能にする複数の実行ユニットとを含む。スイッチは、マトリックス（例えば、クロスバー）スイッチであってよい。マトリックススイッチは、概して、複数の入力と複数の出力との間にある個々のスイッチのアセンブリを意味してよい。スイッチは、非ブロックスイッチであってよく、これにより、他の複数の同時接続が、他の複数の入力を他の複数の出力に接続することの妨げとならない。単一の入力は、複数の出力に接続されてよい。

#### 【0013】

10

20

30

40

50

一実施形態において、(例えば、パイプラインの第1の命令からの)データが(例えば、パイプラインの第2の命令によって)当該データに対するオペレーションを実行することを必要とされる場合、(例えば、第1の命令からの)データの欠損は、(例えば、第2の命令による)オペレーションの実行において遅延を生じさせ得る。このような場合、実行パイプラインの一部は、データが利用可能になる(例えば、第1の命令が実行を完了し、結果のデータが第2の命令によって利用可能となる)まで、ストール(例えば、停止)のような特別な処理を用いてよい。この条件は、概して、データハザードと称されることがある。

#### 【0014】

本開示の複数の特定の実施形態は、データハザード解消ロジックを含む。一実施形態において、ハードウェアプロセッサは、1つ又は複数の(例えば、整数)オフロードエンジン、(例えば、ベクトルプロセッサの)1つ又は複数の実行ユニット、複数のメモリバンク、及び、データハザードを検出及び/又は防止する能力により複数の処理エレメント及びメモリのいずれか又は全ての中におけるデータ移動を容易にする(例えば、データスイッチングロジック及び/又はデータハザード解消ロジックを含む)スイッチを含む。例えば、データハザード解消ロジックは、複数の(例えば右又は左)シフトオペレーション、例えばベクトルシフトオペレーションをサポートするベクトル処理データパスを制御してよい。

#### 【0015】

図1は、本開示の複数の実施形態に係るハードウェアプロセッサ100を示す。図示されるハードウェアプロセッサ100は、プロセッサコア102と、オフロードエンジン104と、メモリ106へのアクセスと、スイッチ108とを含む。これらのコンポーネントの任意の組み合わせが用いられてよい。メモリ106は、別個のコンポーネントとして示されるが、ハードウェアプロセッサ100の一部であってよい。プロセッサコアは、1つ又は複数の実行ユニットを(例えば、直列及び/又は並列に)含んでよい。オフロードエンジンは、例えば、プロセッサコアに対して複数のオペレーションを実行してよく、これにより、プロセッサコアは、他の複数のオペレーションを(例えば同時に)実行してよい。オフロードエンジンは、その複数のオペレーションが完了した(例えば、データが利用可能である)ことを示す信号を(例えば、プロセッサコアに)提供してよい。オフロードエンジンは、プロセッサとは別個(例えば、オフダイ)であってよい。オフロードエンジンは、(例えば、プロセッサとは)別個のメモリへのアクセス、例えば、ダイレクトメモリアccessを有してよい。

#### 【0016】

メモリ106は、任意のデータストレージデバイスであってよい。図示されるメモリ106は、複数のメモリバンク、例えば、メモリバンクA、B、F、及びGを含む。メモリは、メモリバンクA-Gを含んでよい。メモリは、任意の複数のメモリバンクを含んでよい。例えば、メモリは、2つ又はそれより多くのメモリバンク、3つ又はそれより多くのメモリバンク、4つ又はそれより多くのメモリバンク、5つ又はそれより多くのメモリバンク等を含んでよい。各メモリバンクは、それ自身のアクセスポート又は複数のポート(例えば、入力及び/又は出力)を有してよい。各メモリバンクは、他のメモリバンク、例えば複数のメモリバンクから独立してアクセス可能であり、又は、各メモリバンクは、同時にアクセス可能である。プロセッサは、メモリを含んでもよい。プロセッサ及び/又はメモリは、バッファ、例えばシフトレジスタを含んでよい。

#### 【0017】

スイッチは、特定の入力を特定の出力に接続する制御信号を(例えばプロセッサコア102から)受信してよい。スイッチ108は、任意の複数の通信パスを含んでよく、図示されたものに限定されるものではない。スイッチは、コンポーネントを選択的に接続してよい。例えば、スイッチ108は、プロセッサコア、オフロードエンジン、メモリ、及びこれらの任意の組み合わせを接続してよい。なお、本明細書における単一方向の矢印は、一方向の通信であることを必要としないことがあり、例えば、これは、(例えば、当該コ

10

20

30

40

50

ンポーネントへ、及び当該コンポーネントからの)二方向の通信を示してよい。複数の通信パスのいずれか又は全ての組み合わせは、本明細書の複数の実施形態において用いられてよい。スイッチは、どのコンポーネントが共に接続されるか(例えば、2つのコンポーネント間においてアクティブな接続)を制御するデータハザード解消ロジックを含んでよい。スイッチは、プロセッサ(例えば、プロセッサコア)によって、例えば、スイッチに送信される制御信号又は複数の信号によって制御されてよい。一実施形態において、スイッチは、プロセッサのコンポーネント内、例えば、プロセッサコア及び/又はオフロードエンジン内においてよい。スイッチは、複数の接続を制御(例えば、アクティブ化)してよく、これにより、データパスが、データハザードを許さない(例えば、含まない)ようにする。あらゆるデータパスは、スイッチにルートバックされてよく、例えばこれにより、出力は、入力としてルートバックされる。

10

## 【0018】

図2Aは、本開示の複数の実施形態に係るハードウェアプロセッサ200を通る複数のデータパスを示す。図2Aのデータフローは、概して、左から右への態様で進む。図示されたデータパスは、命令をフェッチするプロセッサの命令フェッチユニット210を含む。フェッチされた命令(例えば、マクロ命令)は次に、デコーダ212に進んでよく、これにより、例えば、元の命令から復号され、又は元の命令を他の方法で反映し、又は元の命令から派生した1つ又は複数のマイクロオペレーション、マイクロコードエントリポイント、マイクロ命令、他の命令、又は他の制御信号を、出力として生成するべく復号される。復号された命令は、次に、ベクトル分解ユニット214に進んでよく、これにより、例えば、(例えばVLIW)ベクトル命令を、プロセッサ200の示されたデータパスによって(例えば、露出された命令セットアーキテクチャに対してトランスペアレントな態様で)実行可能な複数の(例えばダブルクワッドワード)サブ命令に変換する。命令フェッチユニット210、デコーダ212、及びベクトル分解ユニット214のいずれか又は全ては、任意であってよい。さらに、複数のパイプラインレジスタ(例えば、EC/MRレジスタ216、MR/EXレジスタ220、EX0/EX1レジスタ222、及びEX/WBレジスタ224)が示されるが、任意の単一又は複数のパイプラインレジスタが用いられてよい。一実施形態において、パイプラインレジスタは用いられない。

20

## 【0019】

示されたデータパスの残りは、複数の連続的なプロセッサパイプラインステージを含み、これらは、メモリ読み出し(MR)、実行0(EX0)、実行1(EX1)、及びメモリライトバック(WB)とラベリングされている。一実施形態において、複数の実行ステージは、例えば、図示された回路を有する(例えば、コアを含む)プロセッサであるプロセッサ200(例えば、プロセッサ200の単一のコア202)の一部である。プロセッサコア202は、複数の実行ステージを(例えば直列に)有する回路、又は単一の実行ステージから出力されたデータを当該単一の実行ステージへの入力として再ルーティングする、例えば、出力を入力として帰還させる回路を含んでよい。一実施形態において、スイッチ208A及びスイッチ208Bは、同じスイッチであってよい。命令フェッチユニット210、デコーダ212、及びベクトル分解ユニット214のいずれか又は全ては、ハードウェアプロセッサ200の一部又はプロセッサコア202であってよい。単一のプロセッサコアが特定の図に示されるが、ハードウェアプロセッサは、複数のプロセッサコアを含んでよい。一実施形態において、誤り訂正(例えば、誤り訂正ユニット)は、パイプラインにおいて、例えばメモリ読み出し(MR)の前に行われてよい。

30

40

## 【0020】

一実施形態において、MRステージにおける複数のメモリ(例えば、RAM)バンクは、ライトバック(WB)ステージにおける複数の同じ物理的インスタンスである。図示された実施形態において、MRステージにおける(例えば、ファーストインファーストアウト(FIFO))バッファA(例えば、入力バッファ)は、オフロードエンジン204の出力を受信するバッファと同じである。一実施形態において、オフロードエンジン204の出力を受信するバッファは、異なるバッファである。オフロードエンジン204は、そ

50

の出力をバッファ 218 の入力に直接提供するものとして示される。一実施形態において、オフロードエンジンは、パイプラインを通して、例えば、パイプラインレジスタ 216 の入力を通して、その出力を提供する。一実施形態において、上記回路のいずれか又はいくつかを迂回するためのバイパス、例えば、マトリックススイッチ 208A とマトリックススイッチ 208B の間に、例えば、図 2B のパス (2) のために用いられる別個のバイパスが存在してよい。一実施形態において、マトリックススイッチ 208A とマトリックススイッチ 208B と間の回路は、例えば、データに対して前方転送以外のいずれのオペレーションも実行することなく、例えば、ALU2、(存在する場合には) EX0/EX1 レジスタ 222、及び(存在する場合には) シフト 2 を通るバイパスとして用いられてよい。

10

#### 【0021】

一実施形態において、プロセッサは、(例えば、図 2A に示される複数のデータパスを通して) 第 2 の(例えば FIFO) バッファ又は複数のバッファ(例えば、バッファ B226 及びバッファ C228) のセットへの書き込みと、第 1 のバッファ(例えば、バッファ A218) からの読み出しとを同時に(例えば、時間的に重複して) 実行する。一実施形態において、ハードウェアプロセッサは、オフロードエンジン 204 へのデータ及び/又はオフロードエンジン 204 からのデータを前処理及び後処理してよい。公開鍵の算出におけるこの利用の一例は、Karatsuba 法による乗算であり、中間乗算結果が合算されてよい。他の例は、楕円曲線暗号(ECC)の点の加算及び点の 2 倍算において用いられる式である。

20

#### 【0022】

複数の特定の実施形態において、プロセッサは、複数(例えば、2 つ)の(例えばベクトル) 実行ユニットを(例えば、並列及び/又は直列に) 含んでよい。一実施形態において、(例えばベクトル) 演算ロジックユニット(ALU) 実行ユニットは、シフト実行ユニットと直列である(例えば、シフト実行ユニットに出力を提供する)。複数の特定の実施形態において、これにより、並列オペレーションにおいて起こり得る(例えば構造的な) データハザードを回避することができる。(例えば構造的な) データハザードは、メモリ(例えば RAM) バンク、例えばメモリバンク 206 に対しても発生し得る。複数のメモリバンク(例えば、メモリバンク A 及びメモリバンク B) は、例えば、オフロードエンジン 204 (例えば、一実施形態において多倍長整数(big integer) 乗算器) が 2 つの入力(例えば、入力オペランド) をとる場合に、(例えば同時に) 用いられてよい。これにより、1 つのメモリバンクにおいて必要とされる同時の読み出し及び/又は書き込みを実行することなく、並列なソーシング及びドレインが可能となり得る。例えば、Karatsuba 法による乗算の分解は、2 つのメモリ書き込み及び 3 つのメモリ読み出しを同時に用いてよく、これにより、アルゴリズムにおいて利用可能な並列性を完全に活用する。図示されたデータパスは、他の複数のメモリバンク(例えば、メモリバンク F 及びメモリバンク G) を含む。一実施形態において、複数のメモリバンク(例えば、各々が、複数の他のメモリバンクの入力ポート及び出力ポートとは別個にそれ自身の入力ポート及び出力ポートを有するもの) を用いることは、より少ない数のメモリバンクに複数のさらなるポートを追加するよりも、(例えば、エリア的に) より効率的なことがある。

複数の特定の実施形態において(例えば、MMP)、複数の(例えば一部の) メモリバンク(例えば、メモリバンク F 及びメモリバンク G) は、連続乗算用の一時ストレージとして、及び/又は内積のアキュムレータとして、用いられてよい。一実施形態において、複数のメモリバンクの第 1 のセット(例えば、メモリバンク F 及びメモリバンク G) は、複数のメモリバンクの第 2 のセット(例えば、メモリバンク A 及びメモリバンク B) と比べてストレージ容量がより小さい。

30

40

#### 【0023】

ソースは、複数の入力ポートにルーティングされてよい。本明細書において開示される複数のデータパスは、前述された複数の例の利用をサポートしてよいが、複数のメモリ(例えば RAM) バンク、及び複数のバッファ、例えば入力及び出力(例えば FIFO) バ

50

ッファの中における直接的な並列データ移動をも可能にしてよい。一実施形態において、プロセッサのデータパスは、例えば他の複数のオペレーションと並列に、一時ストレージ（例えば、メモリバンクF及びG）とメインストレージ（例えば、メモリバンクA及びB）との間におけるデータの前後移動を可能にしてよい。

【0024】

一実施形態において、例えば、オフロードエンジンの同時のソーシング及びドレインに関連する並列性を最大限活用すべく、プロセッサは、以下の複数のソースシンクペア、すなわち、

複数のソース（例えば、メモリバンクA、B、F、及びG及びパッファA）複数の（例えば5つの）出力ポートから複数のALU（例えば、図2A - 2Bにおいて並列なALU1及びALU2）の（例えば4つの）入力ポート（例えば、その各々）のいずれかと、

複数のソース（例えば、メモリバンクA、B、F、及びG、パッファA、及びALU1及びALU2）の複数の（例えば7つの）出力ポートから複数のシンク（例えば、メモリバンクA、B、F、及びG、パッファB、及びパッファC）の（例えば6つの）入力ポートのいずれかと

のペアのうちいずれか又は両方をサポートするデータパスを含んでよい。

【0025】

示された複数のソースシンクペアにおける複数の並列移動は、スイッチ208A及びスイッチ208Bによって接続（例えば、可能に）されてよい。一実施形態において、スイッチは、どの入力又は複数の入力（例えば、ポート）がどの出力（例えば、ポート）又は複数の出力に接続すべきかを選択する制御信号を送信する。例えば、命令は、例えば、どの複数の接続がアクティブであるかを制御する複数の制御信号を、（例えば、これらのフィールドとして）含んでよい。命令は（例えば、プロセッサによって実行された場合に）、どの型のオペレーションが入力データに対してALUによって実行されるべきかを示してよい。ALU1及びALU2は、複数の異なる型のオペレーションを同時に実行してよい。一実施形態、例えば、命令セットアーキテクチャレベルにおけるデータパスの並列性をサポートする実施形態は、この例において「dexec」と称され、以下の表1のフォーマットを有する64ビットVLWスタイルの命令を含んでよい。

【表1】

命令フィールド	説明
dexec ID(4ビット)	dexec命令識別子
メモリバンク読み出し制御 (13ビット)	メモリ読み出しアドレス及び長さを構成
ALU入力ソースルーティング (6ビット)	5つのデータソース間で2つのALUの4つの入力ポートへのマッピングを構成
ALU及びシフトモード制御 (14ビット)	ALU及びシフトのオペレーションを構成
書き込みソースルーティング (27ビット)	7つのデータソース間で6つのデータシンクへのマッピングを構成

【0026】

ここで図 2 B を参照すると、他の命令の例が説明される。図 2 B は、本開示の複数の実施形態に係る図 2 A におけるハードウェアプロセッサ 2 0 0 を通る複数のアクティブなデータパスを示す。この例示的な「d e x e c」命令 2 0 1 は、4 つの異なるフィールド（メモライトバック、複数の乗算器バッファ入力、命令の A L U 1 部分、及び命令の A L U 2 部分）を含む。複数の異なる破線は、命令が（例えば、プロセッサによる命令の実行から）スイッチ 2 0 8 A 及び 2 0 8 B を介して生じさせる複数の異なる接続を示す。例えば、メモライトバックフィールドの第 1 のセクションにおいて、 $a_3[1] = b_0[3]$  は、メモリバンク A における第 3 のアレイの第 2 のエレメントに書き込まれる、メモリバンク B におけるゼロ番目のアレイの第 4 のエレメントを意味してよい。例えば、バンク B の第 4 のエレメントにおいて開始するベクトルは、メモリバンク A の第 2 のエレメントにおいて開始するメモリバンク A に書き込まれ（例えば、コピーされ）てよい。「-」は、フィールドに値がないこと、例えば、特定の例においてオペレーションフィールドが用いられていないことを示してよい。一実施形態において、プロセッサは、（例えば、実行パイプラインにおいて）ストールを含まない。ストールは、ノーオペレーション（NOP）命令、バブル等であってよい。ストールは、ベクトル内ストール、例えば、ベクトルオペレーションが例えば実行ステージ 0 又は実行ステージ 1 において生じている間のストールを意味してよい。

#### 【 0 0 2 7 】

図 2 B に示される命令は、命令内及び / 又は複数の連続的な命令間における潜在的なデータハザードを解決しつつ、複数のソース及びデスティネーションオペランドに対して同時に動作する命令の例であってよい。高いレベルにおいて、示された命令は 2 つの演算オペレーションを含み、これらは、A L U 1 及び A L U 2 によってそれぞれ促進され、かつ、ハードウェアオフロードエンジン（例えば、乗算器）及び 4 つのメモリバンク A、B、F 及び G をターゲットにするデータ移動オペレーションの 3 つのインスタンスに対する複数のキューイングオペランドの 1 つのインスタンスを有する。これらの（例えばシーケンスの）オペレーションの例は、以下のとおりであってよい。

#### 【 0 0 2 8 】

A L U 1 : メモリバンク A からソースされたベクトルオペランド  $a_3[5]$  は、ビット回転を受け、出力にコピーされる。出力は、その最上位ビット（MSB）部分に追加のデータ（例えばクワッドワード）を有する出力オペランドを含む。A L U 1 出力のデスティネーションは、この命令の他の部分に表される。

#### 【 0 0 2 9 】

A L U 2 : A L U 2 は、単一のオペランドをソースし、これは、ハードウェアオフロードエンジン（例えば乗算器）の出力からソースされたバッファ A の出力である。オペレーションは、例えば、さらなるデータ（例えばクワッドワード）を MSB 部分に追加した後、このオペランドを A L U 2 の出力にコピーしてよい。（この最後のオペレーションは、「+ 1」によって指定されてよい。）この例示的命命令は、A L U 1 及び A L U 2 のバイナリオペレータとしての能力を適用しないが、代わりに、単項オペランドの例において適用する。A L U は、バイナリ及び / 又は単項オペレーションを実行してよい。

#### 【 0 0 3 0 】

ハードウェア乗算器オフロードエンジン（例えば乗算器）：乗算器は、（例えば、バッファ B 及びバッファ C からの）その複数のオペランドを（例えば、上述されたように）A L U 1 の出力からソースしてよく、メモリバンク F からソースされたベクトルオペランド  $f[1]$  をソースしてよい。

#### 【 0 0 3 1 】

メモリコピーオペレーション（左から右への読み出し）

( 1 )  $a_2[1] = b_0[3]$  は、（例えばベクトル）オペランド  $b_0[3]$  を  $a_2[1]$  にコピーする。

( 2 )  $b_3[2] = A L U 2$  は、（例えば、上述された）A L U 2 の出力を、メモリバンク B において（例えばベクトル）オペランド  $b_3[2]$  にコピーする。

10

20

30

40

50

(3)  $g[0] = ALU2$ も、 $ALU2$ の出力を(例えば、第2のデスティネーションオペランドとして)、メモリバンクGにおいて(例えばベクトル)オペランド $g[0]$ にコピーする。複数の特定の実施形態は、メモリバンクへの第4のデータコピーオペレーションを可能にし得るが、この選択肢は、この例示的命令においては用いられない。

#### 【0032】

他の例として、ハードウェアプロセッサは、メモリバンクAにストアされた、オフロードエンジン(例えば、乗算器)の幅( $w$ )より大きい数を二乗する複数のデータパスのアクティブ化を含んでよい。例えば、メモリバンクAにストアされた数は、乗算器オフロードエンジンのサイズの2倍のベクトルであってよい。この例において、段階1として、命令が受信されてよく、これにより、バンクAのアドレス $a$ から乗算器バッファB及びCに半ベクトルを送信する(例えば、乗算1)。段階2として、命令は、バンクAのアドレス $a + 幅(w)$ から乗算器バッファB及びCに半ベクトルを送信してよい(例えば、乗算2)。段階3として、命令は、バンクAから乗算器オフロードエンジンに両方の半ベクトルを送信してよく、途中で1をシフトする(例えば、乗算3)。段階4として、ハードウェアプロセッサは、乗算器オフロードエンジンの出力(例えば、乗算1の結果)をメモリバンクBのアドレス $[b, b + 2w]$ にストアしてよい。段階5として、ハードウェアプロセッサは、乗算器オフロードエンジンの出力(例えば、乗算2の結果)をバンクBのアドレス $[b + 2w, b + 4w]$ にストアしてよい。段階6として、ハードウェアプロセッサは、アドレス $[b + w, b + 3w]$ 間でバンクBを読み出し、 $ALU$ を通してデータを送信することにより当該データを乗算器オフロードエンジンの出力(例えば、乗算3の結果)に加算し、これをバンクBのアドレス $[b + w, b + 3w]$ にストアしてよい。

#### 【0033】

段階5及び6の間において、リードアフターライトが行われ、そのロジックは、そこにおけるデータハザードを防止し得る。段階6内において、同じ命令内での読み出しと書き込みとの(例えば完全な)重複が存在してよい。一実施形態において、段階4及び段階1は単一の命令に併合され、同時に動作する。一実施形態において、段階5及び段階2は単一の命令に併合され、同時に動作する。一実施形態において、段階6及び段階3は単一の命令に併合され、同時に動作する。本明細書における複数の特定の実施形態は、パイプライン化されたプロセッサのためのデータハザード解消ロジックを提供する。MMPの例として、リードアフターライト(RAW)ハザードが適用可能であってよく、これは、2つの連続的なベクトル命令の間で、先行する命令の書き込みと後の命令の読み出しとが、同じデータストレージ(例えば、メモリバンク)の重複するアドレス(例えば、アドレス範囲)に対して実行された場合に発生し得る。複数の特定の実施形態において、ストール(例えばNOP)は、ベクトル内(例えば、ベクトル中央における)命令の実行に挿入されるものではなく、例えば、最下位ビット(LSB)をEX0からEX1ステージに転送することを含むデータパスの右シフトオペレーションに起因する。一実施形態において、ベクトルオペレーションは、最下位エレメント(例えば、クワッドワード)から最上位エレメント(例えば、クワッドワード)に向かってよく、例えば、これによりシフトは、右シフトオペレーションを実行すべく、ベクトルの次のエレメント(例えばクワッドワード)からLSBをとる。図2A-2Bのデータパスにおいて、これは、EX0ステージから転送されたLSBをとり、EX0/EX1パイプラインを迂回することに対応する。ストールが複数の特定の実施形態に挿入される、例えば、ベクトルオペレーションの中央において右シフトを実行するMR/EXパイプラインに挿入される場合、EX1ステージは、EX0ステージから転送する有効なLSBの欠如により前進できないことがあり、例えば、ハザード条件が持続するデッドロックを発生させる。ストール挿入における上述された制約のため、(例えば、MMPにおける)RAWデータハザード条件の検出は、命令がベクトル分解ユニットに発行される前に、例えば、データパスにおける保留の書き込みのアドレス範囲に対する命令のアドレス範囲をチェックすることによって、デコーダにおいて実行されてよい。複数の特定の実施形態において、複数のベクトル命令から複数の異なるメモリバンクへのデータパスにおいて、複数の保留の書き込みが存在することがある。本開

10

20

30

40

50

示の複数の特定の形態は、ベクトル内ストール（例えば、NOP遮断）を用いることなく、データパスにおけるハザードを防止する。一実施形態において、ベクトル命令は、（例えばRAW）データハザードをチェックすることなく、デコーダによって発行される。データハザードは、メモリバンクからの読み出し前に、例えば、EC/MRパイプラインステージにおいて、チェックされてよい。チェックは、ベクトルの各エレメント（例えば、ワード）が特定のステージ、例えば、EC/MRステージに到達した場合に、当該エレメントに対して（例えば、エラーチェックユニットによって）実行されてよい。チェックは、データパスの残りに、読み出されるべき即値アドレスである保留の書き込みが1つもないことをチェックすることによってよい。（例えばRAW）データハザードが検出された場合、プロセッサは、ベクトル内ストール（例えばNOP）を挿入せず、ベクトル間ストール（例えばNOP）をそこに挿入してよい。より積極的な命令スケジューリングにおいて、早期段階にスケジューリングされる追従命令は、前の命令のパイプラインが完了しておらず、ハザードを発生させ得ることを検出してよい。例えば、ほぼ重複する複数のアドレス範囲に対するチェックの実行は、当該範囲がメモリの境界及びメモリのラップアラウンドを含む場合に好ましくない影響をもたらすような実装を必要とすることがあるが、これと対照的に、パイプラインが最大でn回サイクルの場合、n個のアドレスコンパレータがハザード検出のために用いられてよい。

#### 【0034】

図3は、本開示の複数の形態に係るデータパス300を通る複数のベクトルの移行を、例えばパイプラインスナップショットとして示す。データハザード解消（例えばロジック）は、以下のいずれか又は全てを含んでよい。一実施形態（a）において、データハザード解消は、例えば複数のベクトルが右に移動した場合に、1つだけインクリメントされるパイプラインのアドレスを有することを含む。このように、（例えばエレメント）xをパイプラインに発行した時点で、 $x = m - d < m$ 、ここでdが正の整数の場合、ストライドdは、2つのベクトルが前進する場合に整合するように維持されてよい。具体的には、ベクトルxからの複数のアドレス読み出しは、dの距離だけ、ベクトルmからの複数のアドレス書き込みを継続的に遅らせてよい。一実施形態（b）において、xの発行時に $x = m + j + d > m + j$ であり、ここでdが正の整数である場合、dのストライドは、2つのベクトルが前進する場合に整合するように維持されてよい。具体的には、ベクトルxからの複数のアドレス読み出しは、パイプラインにおいてアクティブなベクトルmからの書き込みアドレス範囲の外部に（例えば、常に）あってよい。一実施形態において、アドレスチェックにより、（例えば）上記2つの条件（a）及び（b）の1つがd-1を満たす場合に（又はその場合にのみ）、ベクトルxは移動してよい。2つのいずれも適用されない場合、（例えばベクトル間）ストール（例えばNOP）は、例えば、最終的に（a）がd=1により適用可能となるまで又はベクトルMが消失するまで、挿入されてよい。一実施形態（d）において、ベクトルmに追従する他のベクトル、例えばベクトルnが存在する場合、ベクトルxの前進は、さらに $x < n$ 又は $x > n + k$ の対象とされてよい。いずれも適用可能ではない場合、（例えばベクトル間）ストール（例えばNOP）は、例えば、ベクトルnがパイプラインの末尾に配置されるまで、ベクトルm及びnの進行を可能とするように挿入されてよい。次に、（a）、（b）、及び（c）は、ベクトルnに再帰的に適用可能であってよい。一実施形態（e）において、パイプラインにおいてベクトルnに追従するより多くのベクトルが存在する場合、（d）は、再帰的に適用可能であってよい。一実施形態において、アドレスチェックは、（j+k+2）個のコンパレータのいずれかが、読み出されたxのアドレスに等しいmのアドレスを示すか否かをチェックすることを含んでよい。

#### 【0035】

本開示の複数の特定の形態は、いずれの範囲もチェックしないでよく、例えば、入手された単一の読み出アドレスに対して、パイプラインにおける複数の保留の書き込みアドレスの等価性のみチェックしてよい。本開示の複数の特定の形態は、例えば、動的ベクトル追従ロジックを用いることなく、複数の保留の書き込みが1つのベクトル命令に

10

20

30

40

50

属するか否かをチェックしないデータハザード解消ロジックを可能にしてよい。

【 0 0 3 6 】

本明細書における複数の特定の実施形態は、オフロードエンジンの100%又は約100%の利用を実現してよい。本明細書における複数の特定の実施形態は、例えば、（例えば、単一の命令、複数データ（SIMD）実装における場合のように）密結合均一処理、又は、（例えば、画像処理ユニット（GPGPU）コンピューティングにおける汎用コンピューティングの場合のように）1つの処理グループ（例えばアンサンブル）だけがホモジニアスの処理でビジーだが、他の処理グループは待機している非常に緩く結合された実行のいずれかに提供する並列処理アーキテクチャと対照的に、2つの異なる処理エンジン（例えば、プロセッサコア、又はさらなる例として、その実行ユニット）による、（例えばVLIW）命令セットを通してのヘテロジニアス並列処理に、密な結合を提供する。対照的に、本明細書における複数の特定の実施形態において、複数の処理エレメント（例えば、当該システムのプロセッサと異なる速度で動作するオフロードエンジン（又は他のプロセッサ）であるもの）の両方は、命令レベルで協調し、例えば、SIMD構造と対照的にVLIW構造を有する命令セットにより所与のタスクを実行する。

10

【 0 0 3 7 】

図4は、本開示の複数の実施形態に係るフロー図400を示す。フロー図400は、制御信号に基づいて、スイッチにより、ハードウェアプロセッサの並列な複数の実行ユニットの複数の入力を第1のバッファ及び複数のメモリバンクの複数の出力に接続し、並列な複数のメモリバンク及び複数の第2のバッファの複数の入力を第1のバッファ、複数のメモリバンク、及び複数の実行ユニットの複数の出力に接続する段階402と、複数の第2のバッファの複数の出力からオフロードエンジンの複数の入力にデータを提供する段階404と、を含む。プロセッサは、例えば有限ステートマシンのような、本明細書のフロー図に従って動作するロジックを含んでよい。

20

【 0 0 3 8 】

一実施形態において、ハードウェアプロセッサは、並列な複数の（例えばベクトル）実行ユニットと、複数の実行ユニットの複数の入力を第1のバッファ及び複数のメモリバンクの複数の出力に接続し、並列な複数のメモリバンク及び複数の第2のバッファの複数の入力を第1のバッファ、複数のメモリバンク、及び複数の実行ユニットの複数の出力に接続するスイッチと、複数の第2のバッファの複数の出力に接続される複数の入力を有するオフロードエンジンと、を含む。オフロードエンジンの出力は、第1のバッファの入力に接続してよい。ハードウェアプロセッサは、第1のバッファの出力からの読み出しと複数の第2のバッファの複数の入力への書き込みとを同時に実行するデータハザード解消ロジックをさらに含んでよい。データハザード解消ロジックは、（例えばベクトル内）ストールを挿入しなくてよい。複数の（例えばベクトル）実行ユニットは、第1のクロック速度で実行してよく、オフロードエンジンは、第2のより高速（又はより低速）のクロック速度で実行してよい。実行ユニットは、シフトレジスタを含んでよい。複数の実行ユニットの各々は、シフトレジスタを含んでよい。第1のバッファ及び複数の第2のバッファは、ファーストインファーストアウト（FIFO）バッファであってよい。複数のメモリバンクは、4つ又はそれより多くのメモリバンクであってよく、各メモリバンクは、複数の他のメモリバンクの複数の入力ポート及び複数の出力ポートとは別個の入力ポート及び出力ポートを含んでよい。

30

40

【 0 0 3 9 】

他の実施形態において、方法は、制御信号に基づいて、スイッチにより、ハードウェアプロセッサの並列な複数の実行ユニットの複数の入力を第1のバッファ及び複数のメモリバンクの複数の出力に接続し、並列な複数のメモリバンク及び複数の第2のバッファの複数の入力を第1のバッファ、複数のメモリバンク、及び複数の実行ユニットの複数の出力に接続する段階と、複数の第2のバッファの複数の出力からオフロードエンジンの複数の入力にデータを提供する段階と、を含む。方法は、オフロードエンジンの出力から第1のバッファの入力にデータを提供する段階を含んでよい。方法は、第1のバッファの出力か

50

らの読み出しと複数の第2のバッファの複数の入力への書き込みとを同時に実行する段階を含んでよい。方法は、例えば、実行ユニットによって実行されるべき、ストールを挿入しない段階を含んでよい。方法は、複数の実行ユニットによって第1のクロック速度で実行し、オフロードエンジンによって第2のより高速（又はより低速）のクロック速度で実行する段階をさらに含んでよい。複数の実行ユニットの各々は、シフトレジスタを含んでよい。複数のメモリバンクは、4つ又はそれより多くのメモリバンクであり、各メモリバンクは、複数の他のメモリバンクの複数の入力ポート及び複数の出力ポートとは別個の入力ポート及び出力ポートを含む。第1のバッファ及び複数の第2のバッファは、ファーストインファーストアウト（FIFO）バッファであってよい。

#### 【0040】

さらに他の実施形態において、ハードウェアプロセッサは、命令を復号するハードウェアデコーダと、命令を実行することにより、制御信号に基づいて、ハードウェアプロセッサの並列な複数の実行ユニットの複数の入力を第1のバッファ及び複数のメモリバンクの複数の出力に接続し、並列な複数のメモリバンク及び複数の第2のバッファの複数の入力を第1のバッファ、複数のメモリバンク、及び複数の実行ユニットの複数の出力に接続し、複数の第2のバッファの複数の出力からオフロードエンジンの複数の入力にデータを提供するハードウェア実行ユニットと、を含む。オフロードエンジンの出力は、第1のバッファの入力に接続してよい。ハードウェア実行ユニットは、命令を実行してよく、これにより、第1のバッファの出力からの読み出しと、複数の第2のバッファの複数の入力への書き込みとを同時に実行させる。ハードウェア実行ユニットは、ストールを挿入することなく、命令を実行してよい。複数の実行ユニットは、第1のクロック速度で実行してよく、オフロードエンジンは、第2のより高速（又はより低速）のクロック速度で実行してよい。複数の実行ユニットの各々は、シフトレジスタを含んでよい。第1のバッファ及び複数の第2のバッファは、ファーストインファーストアウト（FIFO）バッファであってよい。複数のメモリバンクは、4つ又はそれより多くのメモリバンクであってよく、各メモリバンクは、複数の他のメモリバンクの複数の入力ポート及び複数の出力ポートとは別個の入力ポート及び出力ポートを含んでよい。

#### 【0041】

他の実施形態において、ハードウェア装置は、並列な複数の実行ユニットと、複数の実行ユニットの複数の入力を第1のバッファ及び複数のメモリバンクの複数の出力に接続し、並列な複数のメモリバンク及び複数の第2のバッファの複数の入力を第1のバッファ、複数のメモリバンク、及び複数の実行ユニットの複数の出力に接続する手段と、複数の第2のバッファの複数の出力に接続される複数の入力を有するオフロードエンジンと、を含む。

#### 【0042】

さらに他の実施形態において、装置は、コードをストアするデータストレージデバイスを備え、コードは、ハードウェアプロセッサによって実行された場合に、本明細書に開示される任意の方法をハードウェアプロセッサに実行させる。

#### 【0043】

命令セットは、1つ又は複数の命令フォーマットを含んでよい。所与の命令フォーマットは、実行されるべきオペレーション（例えば、オペコード）及び当該オペレーションが実行されるべき対象であるオペランド、及び/又は他のデータフィールド（例えば、マスク）を、他のものの中から指定する様々なフィールド（例えば、ビットの数、ビットの位置）を定義してよい。いくつかの命令フォーマットは、複数の命令テンプレート（又はサブフォーマット）の定義によってさらに分解される。例えば、所与の命令フォーマットの命令テンプレートは、命令フォーマットの複数のフィールドの異なる複数のサブセットを有するように定義されてよく（含まれる複数のフィールドは、典型的には同じ順序だが、少なくともいくつかは、含まれるフィールドがより少ないため、異なる複数のビット位置を有する）、及び/又は、異なるように解釈される所与のフィールドを有するように定義されてよい。従って、ISAの各命令は、所与の命令フォーマットを用いて（かつ、定義

10

20

30

40

50

される場合には、その命令フォーマットの複数の命令テンプレートの所与のもので表され、オペレーション及び複数のオペランドを指定するための複数のフィールドを含む。例えば、例示的なADD命令は、オペコード及びオペランドフィールドを指定することにより複数のオペランド（ソース1 / デスティネーション及びソース2）を選択するために、具体的なオペコード及びオペコードフィールドを含む命令フォーマットを有し、命令ストリームにおけるこのADD命令の発生は、具体的な複数のオペランドを選択する複数のオペランドフィールド内の具体的な複数の内容を有する。次世代ベクトル拡張（AVX）（AVX1及びAVX2と称され、ベクトル拡張（VEX）符号化スキームを用いるSIMD拡張のセットが、リリース及び/又は公開されている（例えば、2015年4月発行のインテル（登録商標）64及びIA-32アーキテクチャソフトウェアデベロッパーズマニユアル及び2014年10月発行のインテル（登録商標）アーキテクチャ命令セット拡張プログラミングリファレンスを参照）。

10

## 【0044】

## [例示的な命令フォーマット]

本明細書で説明される命令の複数の実施形態は、異なる複数のフォーマットで具現されてよい。さらに、例示的なシステム、アーキテクチャ、及びパイプラインが以下に詳述される。命令の複数の実施形態は、このようなシステム、アーキテクチャ、及びパイプライン上で実行可能であるが、詳述されるものに限定されるものではない。

## 【0045】

## [汎用ベクトル向け命令フォーマット]

ベクトル向け命令フォーマットは、（例えば、特定の複数のフィールド固有の複数のベクトルオペレーションが存在する）複数のベクトル命令に適した命令フォーマットである。ベクトル及びスカラーオペレーションの両方がベクトル向け命令フォーマットを通してサポートされる複数の実施形態が説明されるが、複数の代替的な実施形態は、ベクトル向け命令フォーマットのベクトルオペレーションのみを用いる。

20

## 【0046】

図5A - 5Bは、本開示の複数の実施形態に係る汎用ベクトル向け命令フォーマット及びこれらの命令テンプレートを示すブロック図である。図5Aは、本開示の複数の実施形態に係る汎用ベクトル向け命令フォーマット及びその複数のクラスA命令テンプレートを示すブロック図であり、図5Bは、本開示の複数の実施形態に係る汎用ベクトル向け命令フォーマット及びその複数のクラスB命令テンプレートを示すブロック図である。具体的には、クラスA及びクラスB命令テンプレートが定義される汎用ベクトル向け命令フォーマット500であり、これらは両方とも、複数の非メモリアクセス505命令テンプレート及びメモリアクセス520命令テンプレートを含む。汎用という用語は、ベクトル向け命令フォーマットとの関連では、任意の具体的な命令セットに拘束されない命令フォーマットを指す。

30

## 【0047】

ベクトル向け命令フォーマットが、32ビット（4バイト）又は64ビット（8バイト）のデータエレメント幅（又はサイズ）を有する64バイトベクトルオペランド長（又はサイズ）（従って、64バイトベクトルは、16ダブルワードサイズの複数のエレメント、又は代わりに、8クワッドワードサイズの複数のエレメントのいずれかからなる）、16ビット（2バイト）又は8ビット（1バイト）のデータエレメント幅（又はサイズ）を有する64バイトベクトルオペランド長（又はサイズ）、32ビット（4バイト）、64ビット（8バイト）、16ビット（2バイト）又は8ビット（1バイト）のデータエレメント幅（又はサイズ）を有する32バイトベクトルオペランド長（又はサイズ）、及び32ビット（4バイト）、64ビット（8バイト）、16ビット（2バイト）又は8ビット（1バイト）のデータエレメント幅（又はサイズ）を有する16バイトベクトルオペランド長（又はサイズ）をサポートする本開示の複数の実施形態が説明されるが、複数の代替的な実施形態は、より多くの、より少ない、又は複数の異なるデータエレメント幅（例えば、128ビット（16バイト）のデータエレメント幅）を有する、より多くの、より少

40

50

ない、及び/又は異なる複数のベクトルオペランドサイズ（例えば、256バイトの複数のベクトルオペランド）をサポートしてよい。

【0048】

ハードウェアプロセッサ、例えば、乗算器オフロードエンジンを有するMMPハードウェアプロセッサは、複数の多倍精度演算を実行してよい。ハードウェアプロセッサは、128、256、512、1024等ビットのベクトルをシフトし、例えば、単一の（例えばVLW）命令において（例えば非常に大きい）整数を2で除算すること又は（例えば非常に大きい）整数を2で乗算することを実行してよい。

【0049】

図5AのクラスA命令テンプレートは、1)その内部に、非メモリアクセス、フルラウンド制御型オペレーション510命令テンプレート、及び非メモリアクセス、データ変換型オペレーション515命令テンプレートが示される複数の非メモリアクセス505命令テンプレートと、2)その内部に、メモリアクセス、一時的525命令テンプレート、及びメモリアクセス、非一時的530命令テンプレートが示されるメモリアクセス520命令テンプレートとを含む。図5BのクラスB命令テンプレートは、1)その内部に、非メモリアクセス、書き込みマスク制御、部分的ラウンド制御型オペレーション512命令テンプレート、及び非メモリアクセス、書き込みマスク制御、vsizetypeオペレーション517命令テンプレートが示される非メモリアクセス505命令テンプレートと、2)その内部に、メモリアクセス、書き込みマスク制御527命令テンプレートが示されるメモリアクセス520命令テンプレートとを含む。

【0050】

汎用ベクトル向け命令フォーマット500は、図5A-5Bに示される順序で、以下に列挙される以下のフィールドを含む。

【0051】

フォーマットフィールド540：このフィールドにおける特定値（命令フォーマット識別子の値）は、ベクトル向け命令フォーマットを一意に特定し、従って、命令ストリームにおけるベクトル向け命令フォーマットの複数の命令の発生を特定する。このように、このフィールドは、汎用ベクトル向け命令フォーマットのみを有する命令セットには必要とされないという意味で任意である。

ベースオペレーションフィールド542：その内容は、異なる複数のベースオペレーションを区別する。

【0052】

レジスタインデックスフィールド544：その内容は、直接的に又はアドレス生成を通して、ソース及びデスティネーションオペランドの位置が、レジスタ内にあるか又はメモリ内にあるかを指定する。これらは、 $P \times Q$ （例えば、 $32 \times 512$ 、 $16 \times 128$ 、 $32 \times 1024$ 、 $64 \times 1024$ ）レジスタファイルからN個のレジスタを選択するために十分な数のビットを含む。一実施形態において、Nは、3つのソース及び1つのデスティネーションレジスタまでであってよいが、複数の代替的な実施形態は、より多くの又はより少ないソース及びデスティネーションレジスタをサポートしてよい（例えば、2つのソースまでであっても、これらのソースのうち1つがデスティネーションとしても動作するようなソースをサポートしてよく、3つのソースまでであっても、これらのソースのうち1つがデスティネーションとしても動作するソースをサポートしてよく、2つのソース及び1つのデスティネーションまでをサポートしてよい。）

【0053】

修飾子フィールド546：その内容は、メモリアクセスを指定する汎用ベクトル命令フォーマットの複数の命令の発生を、メモリアクセスを指定しないものから、すなわち、複数の非メモリアクセス505命令テンプレートとメモリアクセス520命令テンプレートとを区別する。複数のメモリアクセスオペレーションは、（いくつかの場合において、複数のレジスタ内の複数の値を用いて、ソース及び/又はデスティネーションアドレスを指定する）メモリ階層の読み出し及び/又は書き込みを実行するが、複数の非メモリアクセ

10

20

30

40

50

スオペレーションは、これを実行しない（例えば、ソース及びデスティネーションがレジスタである）。一実施形態において、このフィールドは、複数のメモリアドレス算出を実行する3つの異なる態様の間でさらに選択するが、複数の代替的な実施形態は、複数のメモリアドレス算出を実行するより多くの、より少ない、又は異なる態様をサポートしてよい。

**【0054】**

増加オペレーションフィールド550：その内容は、ベースオペレーションに加えて、様々な異なるオペレーションの中で実行されるべきいずれかを区別する。このフィールドは、状況に応じて異なる。本開示の一実施形態において、このフィールドは、クラスフィールド568、アルファフィールド552、及びベータフィールド554に分割される。増加オペレーションフィールド550は、2、3、又は4つの命令ではなく単一の命令において実行されるべき複数のオペレーションの共通グループを可能とする。

10

**【0055】**

スケールフィールド560：その内容は、メモリアドレス生成のために（例えば、 $2^{\text{scale}} \times \text{インデックス} + \text{ベース}$ を用いるアドレス生成のために）、インデックスフィールドの内容のスケールリングを可能とする。

**【0056】**

変位フィールド562A：その内容は、（例えば、 $2^{\text{scale}} \times \text{インデックス} + \text{ベース} + \text{変位}$ を用いるアドレス生成のために）メモリアドレス生成の一部として用いられる。

**【0057】**

変位ファクタフィールド562B（なお、変位ファクタフィールド562Bの直上に変位フィールド562Aを並べることは、一方又は他方が用いられることを示す）：その内容は、アドレス生成の一部として用いられる。これは、メモリアクセスのサイズ（N）によりスケールされるべき変位ファクタを指定する。ここで、Nは、（例えば、 $2^{\text{scale}} \times \text{インデックス} + \text{ベース} + \text{スケールされた変位}$ を用いるアドレス生成のための）メモリアクセスにおけるバイト数である。複数の冗長下位ビットは無視され、従って、変位ファクタフィールドの内容は、実効アドレスの算出に用いられる最終的な変位を生成すべく、複数のメモリオペランドの合計サイズ（N）により乗算される。Nの値は、プロセッサハードウェアによって、（本明細書において後述される）フルオペコードフィールド574及びデータ操作フィールド554Cに基づいて、ランタイムで決定される。変位フィールド562A及び変位ファクタフィールド562Bは、これらが非メモリアクセス505命令テンプレートのために用いられないという意味で任意であり、及び/又は、異なる複数の実施形態は、これら2つのうち1つだけを実装してよく、又はこれらのうち1つも実装しなくてよい。

20

30

**【0058】**

データエレメント幅フィールド564：その内容は、多数のデータエレメント幅のうちのいずれが（いくつかの実施形態においては全ての命令に対して、複数の他の実施形態では、複数の命令のうちのいくつかのみに対して）用いられるかを区別する。このフィールドは、複数のオペコードのいくつかの態様を用いて、1つだけのデータエレメント幅がサポートされる場合、及び/又は複数のデータエレメント幅がサポートされる場合には必要とされないという意味で任意である。

40

**【0059】**

書き込みマスクフィールド570：その内容は、データエレメント位置毎に、デスティネーションベクトルオペランドにおける当該データエレメント位置がベースオペレーション及び増加オペレーションの結果を反映するか否かを制御する。クラスA命令テンプレートは、併合書き込みマスクをサポートするが、クラスB命令テンプレートは、併合及びゼロ書き込みマスクの両方をサポートする。併合の場合、複数のベクトルマスクは、（ベースオペレーション及び増加オペレーションによって指定された）いずれのオペレーションの実行中も、デスティネーションの複数のエレメントの任意のセットを更新から保護することを可能にする。他の一実施形態において、対応するマスクビットが0を有する場合、

50

デスティネーションの各エレメントの古い値を保持する。対照的に、複数のゼロ書き込みベクトルマスクが（ベースオペレーション及び増加オペレーションによって指定された）いずれのオペレーションの実行中もデスティネーションの複数のエレメントの任意のセットへのゼロ書き込みを可能にする場合、一実施形態において、対応するマスクビットが0値を有する場合、デスティネーションのエレメントは0にセットされる。この機能性のサブセットは、実行されているオペレーションのベクトル長を制御する（すなわち、複数のエレメントのスパンが最初から最後のものまで変更される）能力である。しかしながら、変更される複数のエレメントが連続する必要はない。従って、書き込みマスクフィールド570は、ロード、ストア、演算、論理等を含む複数の部分的なベクトルオペレーションを可能とする。書き込みマスクフィールド570の内容が多数の書き込みマスクレジスタの中から使用されるべき書き込みマスクを含む1つを選択する（従って、書き込みマスクフィールド570の内容は、実行されるべきマスキングを間接的に特定する）本開示の複数の実施形態が説明されるが、代替的な実施形態によれば、代わりに又はさらに、マスク書き込みフィールド570の内容は、実行されるべきマスキングを直接的に指定することを可能とする。

10

【0060】

即値フィールド572：その内容は、即値の指定を可能とする。このフィールドは、即値をサポートしない汎用ベクトル向けフォーマットの実装では存在せず、即値を用いない複数の命令に存在しないという意味で任意である。

【0061】

20

クラスフィールド568：その内容は、複数の命令の異なるクラス間を区別する。図5A-Bを参照すると、このフィールドの内容は、クラスA及びクラスB命令の間で選択する。図5A-Bにおいて、複数の角が円い四角は、特定値がフィールドに存在する（例えば、図5A-Bにおいて、クラスフィールド568に対してクラスA568A及びクラスB568Bのそれぞれ）ことを示すために用いられる。

【0062】

[クラスA命令テンプレート]

複数のクラスAの非メモリアクセス505命令テンプレートの場合、アルファフィールド552は、RSフィールド552Aと解釈され、その内容は、複数の異なる増加オペレーション型の中でいずれが実行されるべきかを区別し（例えば、非メモリアクセス、ラウンド型オペレーション510及び非メモリアクセス、複数のデータ変換型オペレーション515命令テンプレートに対し、ラウンド552A.1及びデータ変換552A.2がそれぞれ指定される）、ベータフィールド554は、指定される型の複数のオペレーションの中でいずれが実行されるべきかを区別する。非メモリアクセス505命令テンプレートには、スケールフィールド560、変位フィールド562A、及び変位スケールフィールド562Bは存在しない。

30

【0063】

[メモリアクセス命令テンプレート：フルラウンド制御型オペレーション]

非メモリアクセスフルラウンド制御型オペレーション510の命令テンプレートにおいて、ベータフィールド554は、ラウンド制御フィールド554Aと解釈され、その内容は、静的なラウンドを提供する。説明された本開示の複数の実施形態において、ラウンド制御フィールド554Aは、全浮動小数点例外抑制（SAE）フィールド556及びラウンドオペレーション制御フィールド558を含むが、複数の代替的な実施形態は、これらの複数の概念の両方を同じフィールドに符号化することをサポートしてよく、又はこれらの複数の概念/複数のフィールドの1つ又は他を有するだけでよい（例えば、ラウンドオペレーション制御フィールド558のみを有してよい）。

40

【0064】

SAEフィールド556：その内容は、例外イベント報告を無効化するか否かを区別する。SAEフィールド556の内容が、抑制可能であることを示す場合、所与の命令は、いずれの種類の浮動小数点例外フラグも報告せず、いずれの浮動小数点例外処理部も起動

50

しない。

【 0 0 6 5 】

ラウンドオペレーション制御フィールド 5 5 8 : その内容は、複数のラウンドオペレーションのいずれのグループ (例えば、切り上げ、切り捨て、0 への丸め及び近似値への丸め) を実行するかを区別する。従って、ラウンドオペレーション制御フィールド 5 5 8 は、命令に基づいて、ラウンドオペレーションモードの変更を可能とする。プロセッサが複数のラウンドオペレーションモードを指定するための制御レジスタを含む本開示の一実施形態において、ラウンドオペレーション制御フィールド 5 5 0 の内容は、当該レジスタ値にオーバライドする。

【 0 0 6 6 】

[ 非メモリアクセス命令テンプレート - データ変換型オペレーション ]

非メモリアクセスデータ変換型オペレーション 5 1 5 命令テンプレートにおいて、ベータフィールド 5 5 4 はデータ変換フィールド 5 5 4 B と解釈され、その内容は、多数のデータ変換のうちいずれが実行されるか (例えば、データ変換なし、スウィズル、ブロードキャスト) を区別する。

【 0 0 6 7 】

クラス A のメモリアクセス 5 2 0 命令テンプレートの場合、アルファフィールド 5 5 2 は、エビクションヒントフィールド 5 5 2 B と解釈され、その内容は、複数のエビクションヒントの中からいずれが用いられるべきかを区別し (図 5 A において、一時的 5 5 2 B . 1 及び非一時的 5 5 2 B . 2 が、メモリアクセス、一時的 5 2 5 命令テンプレート及びメモリアクセス、非一時的 5 3 0 命令テンプレートに対してそれぞれ指定される)、ベータフィールド 5 5 4 は、データ操作フィールド 5 5 4 C と解釈され、その内容は、(プリミティブとしても知られる) 多数のデータ操作オペレーションのうちいずれが実行されるべきか (例えば、操作なし、ブロードキャスト、ソースのアップコンバート及びデスティネーションのダウンコンバート) を区別する。複数のメモリアクセス 5 2 0 命令テンプレートは、スケールフィールド 5 6 0 と、任意に、変位フィールド 5 6 2 A 又は変位スケールフィールド 5 6 2 B とを含む。

【 0 0 6 8 】

複数のベクトルメモリ命令は、変換サポートにより、メモリからの複数のベクトルロード、メモリへの及び複数のベクトルストアを実行する。通常のベクトル命令と同様に、複数のベクトルメモリ命令は、書き込みマスクとして選択されるベクトルマスクの内容によって命令される、実際に転送される複数のエレメントとともに、データエレメント単位の態様で、メモリから / メモリへデータを転送する。

【 0 0 6 9 】

[ メモリアクセス命令テンプレート - 一時的 ]

一時的データは、キャッシュによる利益を十分に受けられるほどすぐに再使用される可能性が高いデータである。しかしながら、これは示唆であり、複数の異なるプロセッサが、この示唆を完全に無視することを含む異なる態様で、これを実装してよい。

【 0 0 7 0 】

[ メモリアクセス命令テンプレート - 非一時的 ]

非一時的データは、一次レベルキャッシュにおいてキャッシュによる利益を受けられるほど十分即座に再使用される可能性が低いデータであり、エビクションの優先度が与えられなければならない。しかしながら、これは示唆であり、複数の異なるプロセッサが、この示唆を完全に無視することを含む異なる態様で、これを実装してよい。

【 0 0 7 1 】

[ 命令テンプレート - クラス B ]

複数のクラス B 命令テンプレートの場合、アルファフィールド 5 5 2 は、書き込みマスク制御 (Z) フィールド 5 5 2 C と解釈され、その内容は、書き込みマスクフィールド 5 7 0 によって制御される書き込みマスキングが、併合又はゼロ書き込みを実行すべきかを区別する。

10

20

30

40

50

## 【 0 0 7 2 】

複数のクラスB非メモリアクセス505命令テンプレートの場合、ベータフィールド554の一部は、RLフィールド557Aと解釈され、その内容は、複数の異なる増加オペレーション型の中でいずれが実行されるべきかを区別し(例えば、ラウンド557A.1及びベクトル長(VSIZE)557A.2は、非メモリアクセス、書き込みマスク制御、部分的ラウンド制御型オペレーション512命令テンプレート及び非メモリアクセス、書き込みマスク制御、VSIZE型オペレーション517命令テンプレートに対してそれぞれ指定される)、ベータフィールド554の残りは、指定される型の複数のオペレーションのいずれが実行されるべきかを区別する。非メモリアクセス505命令テンプレートには、スケールフィールド560、変位フィールド562A、及び変位スケールフィールド562Bは存在しない。

10

## 【 0 0 7 3 】

非メモリアクセス、書き込みマスク制御、部分的ラウンド制御型オペレーション510の命令テンプレートにおいて、ベータフィールド554の残りは、ラウンドオペレーションフィールド559Aと解釈され、例外イベント報告は、無効化される(所与の命令は、いずれの種類の浮動小数点例外フラグも報告せず、いずれの浮動小数点例外処理部も起動しない)。

## 【 0 0 7 4 】

ラウンドオペレーション制御フィールド559A:単にラウンドオペレーション制御フィールド558として、その内容は、複数のラウンドオペレーションのいずれのグループが(例えば、切り上げ、切り捨て、0への丸め及び近似値への丸めを)実行するかを区別する。従って、ラウンドオペレーション制御フィールド559Aは、命令に基づいて、ラウンドオペレーションモードの変更を可能とする。プロセッサが複数のラウンドオペレーションモードを指定するための制御レジスタを含む本開示の一実施形態において、ラウンドオペレーション制御フィールド550の内容は、当該レジスタ値にオーバライドする。

20

## 【 0 0 7 5 】

非メモリアクセス、書き込みマスク制御、VSIZE型オペレーション517命令テンプレートにおいて、ベータフィールド554の残りは、ベクトル長フィールド559Bと解釈され、その内容は、多数のデータベクトル長の中からいずれが実行されるべきか(例えば、128、256、又は512バイト)を区別する。

30

## 【 0 0 7 6 】

複数のクラスBのメモリアクセス520命令テンプレートの場合、ベータフィールド554の一部は、ブロードキャストフィールド557Bと解釈され、その内容は、ブロードキャスト型のデータ操作オペレーションが実行されるべきか否かを区別し、ベータフィールド554の残りは、ベクトル長フィールド559Bと解釈される。複数のメモリアクセス520命令テンプレートは、スケールフィールド560と、任意に、変位フィールド562A又は変位スケールフィールド562Bとを含む。

## 【 0 0 7 7 】

汎用ベクトル向け命令フォーマット500に関して、フォーマットフィールド540、ベースオペレーションフィールド542、及びデータエレメント幅フィールド564を含むフルオペコードフィールド574が示される。一実施形態において、フルオペコードフィールド574はこれらフィールドの全てを含むものとして示されるが、フルオペコードフィールド574は、これらの全てをサポートしてはいない複数の実施形態においては、これらフィールドの一部を含む。フルオペコードフィールド574は、オペレーションコード(オペコード)を提供する。

40

## 【 0 0 7 8 】

増加オペレーションフィールド550、データエレメント幅フィールド564、及び書き込みマスクフィールド570は、汎用ベクトル向け命令フォーマットにおいて、これら全ての機能が命令に基づいて指定されることを可能とする。

## 【 0 0 7 9 】

50

書き込みマスクフィールド及びデータエレメント幅フィールドの組み合わせは、型別の複数の命令を形成し、これらによれば、マスクは複数の異なるデータエレメント幅に基づいて適用されることが可能となる。

#### 【 0 0 8 0 】

クラス A 及びクラス B 内で見られる様々な複数の命令テンプレートは、複数の異なる状況において有益である。本開示の複数の実施形態のいくつかにおいて、複数の異なるプロセッサ又はプロセッサ内の複数の異なるコアは、クラス A のみ、クラス B のみ、又は両方のクラスをサポートしてよい。例えば、汎用コンピューティング向けの高性能汎用アウトオブオーダーコアは、クラス B のみをサポートしてよく、主にグラフィック及び/又は科学的(スループット)コンピューティング向けのコアは、クラス A のみをサポートしてよく、両方向けのコアは、両方をサポートしてよい(勿論、コアは、両方のクラスからの複数のテンプレート及び命令のいくつかの組み合わせを有するが、両方のクラスからの複数のテンプレート及び命令の全てが本開示の範囲内にある訳ではない)。また、単一のプロセッサは、全てが同じクラスをサポートする、又は異なるコアが異なるクラスをサポートする複数のコアを含んでよい。例えば、別個のグラフィックコア及び汎用コアを有するプロセッサにおいて、主にグラフィック及び/又は科学コンピューティング向けに意図された複数のグラフィックコアの1つは、クラス A のみをサポートしてよく、複数の汎用コアのうち1つ又は複数の、クラス B のみをサポートする汎用コンピューティング向けに意図されたアウトオブオーダー実行及びレジスタリネーミングを有する高性能汎用コアであってよい。別個のグラフィックコアを有さない他のプロセッサは、クラス A 及びクラス B の両方をサポートする汎用インオーダー又はアウトオブオーダーコアをもう1つ含んでよい。勿論、1つのクラスからの複数の機能は、本開示の複数の異なる実施形態における他のクラスで実装されてもよい。高水準言語で記述された複数のプログラムは、1) 実行のためにターゲットのプロセッサにサポートされるクラスの命令のみを有する形式、又は2) 全クラスの命令の複数の異なる組み合わせを用いて記述された代替的な複数のルーチンを有し、コードを現在実行中のプロセッサにサポートされる複数の命令に基づいて実行する複数のルーチンを選択する制御フローコードを有する形式を含む、様々な異なる実行可能形式にされる(例えば、ジャストインタイムにコンパイルされる又は静的にコンパイルされる)ことがある。

#### 【 0 0 8 1 】

[ 例示的な特定ベクトル向け命令フォーマット ]

図 6 A は、本開示の複数の実施形態に係る例示的な特定ベクトル向け命令フォーマットを示すブロック図である。図 6 A は、複数のフィールドの位置、サイズ、解釈及び順序、ならびにこれらのフィールドのうちいくつかに対する複数の値を指定するという意味で具体的な特定ベクトル向け命令フォーマット 6 0 0 を示す。特定ベクトル向け命令フォーマット 6 0 0 は、x 8 6 命令セットを拡張するために用いられてよく、従って、複数のフィールドのうちいくつかは、既存の x 8 6 命令セット及びこれらの拡張(例えば、AVX)において用いられるものと同様又は同じである。このフォーマットは、複数の拡張を有する既存の x 8 6 命令セットのプレフィクス符号化フィールド、リアルオペコードバイトフィールド、MOD R / M フィールド、SIB フィールド、変位フィールド、及び複数の即値フィールドとの整合性を維持する。図 6 A の複数のフィールドがマッピングされる図 5 A - B の複数のフィールドが示される。

#### 【 0 0 8 2 】

本開示の複数の実施形態は、例示目的のため、汎用ベクトル向け命令フォーマット 5 0 0 との関連で特定ベクトル向け命令フォーマット 6 0 0 を参照して説明されるが、本開示は、特許請求の範囲に記載されたものを除いて、特定ベクトル向け命令フォーマット 6 0 0 に限定されるものではないことを理解されたい。例えば、汎用ベクトル向け命令フォーマット 5 0 0 は、様々なフィールドの様々な可能なサイズを検討し、特定ベクトル向け命令フォーマット 6 0 0 は、複数の具体的なサイズの複数のフィールドを有するものとして示される。具体的な例を用いて、データエレメント幅フィールド 5 6 4 は、特定ベクトル

向け命令フォーマット600の1ビットフィールドとして示されるが、本開示は、この例に限定されるものではない(すなわち、汎用ベクトル向け命令フォーマット500は、複数の他のサイズのデータエレメント幅フィールド564を検討する)。

【0083】

汎用ベクトル向け命令フォーマット500は、図6Aに示される順序で、以下に列挙される以下のフィールドを含む。EVEXプレフィクス(バイト0-3)602は、4バイト形式で符号化される。

【0084】

フォーマットフィールド540(EVEXバイト0、ビット[7:0])]:第1のバイト(EVEXバイト0)は、フォーマットフィールド540であり、これは、0x62(本開示の一実施形態において、ベクトル向け命令フォーマットを区別するために用いられる固有値)を含む。

【0085】

第2-第4のバイト(EVEXバイト1-3)は、具体的な能力を提供する多数のビットフィールドを含む。

【0086】

REXフィールド605(EVEXバイト1、ビット[7-5])は、EVEX.Rビットフィールド(EVEXバイト1、ビット7-R)、EVEX.Xビットフィールド(EVEXバイト1、ビット[6]-X)、及び557BEXバイト1、ビット[5]-B)からなる。EVEX.R、EVEX.X、及びEVEX.Bビットフィールドは、対応する複数のVEXビットフィールドと同じ機能性を提供し、1の補数形式を用いて符号化される、すなわち、ZMM0は1111Bとして符号化され、ZMM15は0000Bとして符号化される。当技術分野において公知であるように、複数の命令の複数の他のフィールドは、複数のレジスタインデックスの下位3ビット(rrr、xxx、及びbbb)を符号化し、これにより、Rrrr、xxxx、及びBbbbは、EVEX.R、EVEX.X、及びEVEX.Bを追加することによって形成されてよい。

【0087】

REX'フィールド510:これは、REX'フィールド510の第1の部分であり、拡張された32レジスタセットの上位16又は下位16のいずれかを符号化するために用いられるEVEX.R'ビットフィールド(EVEXバイト1、ビット[4]-R')である。本開示の一実施形態において、このビットは、以下に示される他のものと共に、ビット反転フォーマットでストアされることにより、リアルオペコードバイトは62であるが、(後述の)MOD R/MフィールドにおいてMODフィールドの値11を受け付けない(周知のx86の32ビットモードにおける)BOUND命令から区別するが、本開示の複数の代替的な実施形態は、このビット及び以下に示される反転フォーマットの他のビットをストアしない。1の値は、下位16個のレジスタを符号化するために用いられる。換言すると、R'Rrrrは、EVEX.R'、EVEX.R、及び複数の他のフィールドからの他のRRRを組み合わせることによって形成される。

【0088】

オペコードマップフィールド615(EVEXバイト1、ビット[3:0]-mmmm):その内容は、暗示された先頭オペコードバイト(0F、0F38、又は0F3)を符号化する。

【0089】

データエレメント幅フィールド564(EVEXバイト2、ビット[7]-W)は、表記EVEX.Wにより表される。EVEX.Wは、データ型(32ビットデータエレメント又は64ビットデータエレメントのいずれか)の粒度(サイズ)を定義するために用いられる。

【0090】

EVEX.vvvv620(EVEXバイト2、ビット[6:3]-vvvv):EVEX.vvvvの役割は、1)EVEX.vvvvは、反転(1の補数)形式で指定され

10

20

30

40

50

る第1のソースレジスタオペランドを符号化し、2つ又はそれより多くのソースオペランドを有する複数の命令に対して有効であり、2) `EVEX.vvvv`は、特定の複数のベクトルシフトに対して1の補数形式で指定されるデスティネーションレジスタオペランドを符号化し、又は3) `EVEX.vvvv`は、いずれのオペランドも符号化せず、フィールドは保持されて1111bを含まなくてはならないということを含んでよい。従って、`EVEX.vvvv`フィールド620は、反転(1の補数)形式でストアされた第1のソースレジスタ指定子の下位4ビットを符号化する。命令に応じて、追加の異なる`EVEX`ビットフィールドは、指定子サイズを32のレジスタに拡張するために用いられる。

#### 【0091】

`EVEX.U568`クラスフィールド(`EVEX`バイト2、ビット[2]-U): `EVEX.U=0`の場合、これは、クラスA又は`EVEX.U0`を示す。`EVEX.U=1`の場合、これは、クラスB又は`EVEX.U1`を示す。

#### 【0092】

プレフィクス符号化フィールド625(`EVEX`バイト2、ビット[1:0]-pp)は、ベースオペレーションフィールドに対してさらなる複数のビットを提供する。`EVEX`プレフィクスフォーマットにおける複数のレガシSSE命令に対するサポートを提供することに加えて、これは、SIMDプレフィクスを圧縮する利益をも有する(SIMDプレフィクスを表現するバイトを必要とするのではなく、`EVEX`プレフィクスは2ビットのみを必要とする)。一実施形態において、レガシフォーマット及び`EVEX`プレフィクスフォーマットの両方においてSIMDプレフィクス(66H、F2H、F3H)を用いる複数のレガシSSE命令をサポートすべく、これらのレガシSIMDプレフィクスは、SIMDプレフィクス符号化フィールドに符号化され、デコーダのPLAに提供される前に、ランタイムにおいてレガシSIMDプレフィクスに拡張される(従って、PLAは、これらのレガシ命令のレガシ及び`EVEX`フォーマットの両方を変更することなく実行することができる)。より新たな複数の命令は、`EVEX`プレフィクス符号化フィールドの内容を直接、オペコード拡張として用いることができたが、複数の特定の実施形態は、整合性のために同様に拡張するが、これらのレガシSIMDプレフィクスにより指定された異なる意味を可能とする。代替的な実施形態は、2ビットSIMDプレフィクスの符号化をサポートするPLAを再設計してよく、従って、拡張を必要としない。

#### 【0093】

アルファフィールド552(`EVEX`バイト3、ビット[7]-EH、`EVEX.EH`、`EVEX.rs`、`EVEX.RL`、`EVEX.write mask control`、及び`EVEX.N`としても知られ、また で示される): 前述されたように、このフィールドは状況に応じて異なる。

#### 【0094】

ベータフィールド554(`EVEX`バイト3、ビット[6:4]-SSS、`EVEX.s2-0`、`EVEX.r2-0`、`EVEX.r1`、`EVEX.LL0`、`EVEX.LLB`としても知られ、また で示される): 前述されたように、このフィールドは状況に応じて異なる。

#### 【0095】

`REX'`フィールド510: これは、`REX'`フィールドの残りであり、拡張された32レジスタセットの上位16又は下位16のいずれかを符号化するために用いられてよい`EVEX.V'`ビットフィールド(`EVEX`バイト3、ビット[3]-V')である。このビットは、ビット反転フォーマットでストアされる。1の値は、下位16個のレジスタを符号化するために用いられる。換言すると、`V'VVVV`は、`EVEX.V'`、`EVEX.vvvv`を組み合わせることによって形成される。

#### 【0096】

書き込みマスクフィールド570(`EVEX`バイト3、ビット[2:0]-kkk): その内容は、前述されたように、複数の書き込みマスクレジスタ内のレジスタにおいてインデックスを指定する。本開示の一実施形態において、特定値`EVEX.kkk=000`

10

20

30

40

50

は、特定の命令のために用いられる書き込みマスクがないことを示唆する特別な挙動を有する（これは、ハードウェアに組み込まれた全ての書き込みマスク又はマスキングハードウェアを迂回するハードウェアを用いることを含む、様々な態様で実装されてよい）。

【0097】

リアルオペコードフィールド630（バイト4）は、オペコードバイトとしても知られる。オペコードの一部は、このフィールドにおいて指定される。

【0098】

MOD R / Mフィールド640（バイト5）は、MODフィールド642、Regフィールド644及びR / Mフィールド646を含む。前述されたように、MODフィールド642の内容は、メモリアクセス及び非メモリアクセスオペレーション間を区別する。Regフィールド644の役割は、2つの状況に対して要約されてよい。すなわち、デスティネーションレジスタオペランド又はソースレジスタオペランドのいずれかを符号化する場合、又はオペコード拡張として扱われ、いずれの命令オペランドの符号化にも用いられない場合である。R / Mフィールド646の役割は、以下を含んでよい。すなわち、メモリアドレスを参照する命令オペランドを符号化すること、又はデスティネーションレジスタオペランド又はソースレジスタオペランドのいずれかを符号化することである。

【0099】

スケールリング、インデックス、ベース（SIB）バイト（バイト6）：前述されたように、スケールフィールド550の内容は、メモリアドレス生成のために用いられる。SIB . xxx654及びSIB . bbb656：これらのフィールドの内容は、レジスタインデックスXxx及びBbbbに関して前に説明されている。

【0100】

変位フィールド562A（バイト7 10）：MODフィールド642が10を含む場合、バイト7 - 10は変位フィールド562Aであり、これは、レガシ32ビット変位（disp32）と同じ機能をはたし、かつ、バイト粒度において機能する。

【0101】

変位ファクタフィールド562B（バイト7）：MODフィールド642が01を含む場合、バイト7は、変位ファクタフィールド562Bである。このフィールドの位置は、バイト粒度において機能するレガシx86命令セットの8ビット変位（disp8）の位置と同じである。disp8は符号拡張されるので、これは、-128及び127バイトオフセット間でのみアドレスすることができる。複数の64バイトキャッシュラインに関して、disp8は、-128、-64、0、及び64の4つのみという実には有用な値にセットされ得る8ビットを用いる。多くの場合、より広範囲が必要とされるので、disp32が用いられる。しかしながら、disp32は、4バイトを必要とする。disp8及びdisp32と対照的に、変位ファクタフィールド562Bは、disp8の再解釈である。変位ファクタフィールド562Bを用いる場合、実際の変位は、メモリアクセスのサイズ（N）で乗算された変位ファクタフィールドの内容によって決定される。この型の変位は、disp8 x Nと称される。これは、（変位のために用いられる単一バイトであるが、はるかにより広範囲を有する）平均命令長を低減させる。このような圧縮された変位は、有効な変位がメモリアクセスの粒度の倍数であるという前提に基づいており、従って、アドレスオフセットの冗長下位ビットは、符号化される必要がない。換言すると、変位ファクタフィールド562Bは、レガシx86命令セットの8ビット変位を代替する。従って、変位ファクタフィールド562Bは、disp8がdisp8 x Nにオーバーロードされる唯一の例外はあるものの、x86命令セットの8ビット変位と同じ態様で符号化される（ModRM / SIB符号化ルールに変更はない）。換言すると、（バイト単位のアドレスオフセットを得るために、変位をメモリアクセスのサイズによりスケールリングする必要がある）ハードウェアによる変位値の解釈のみを除いて、複数の符号化ルール又は符号化長に変更はない。即値フィールド572は、前述されたように動作する。

【0102】

10

20

30

40

50

[フルオペコードフィールド]

図6Bは、本開示の一実施形態に係るフルオペコードフィールド574を形成する特定ベクトル向け命令フォーマット600の複数のフィールドを示すブロック図である。具体的には、フルオペコードフィールド574は、フォーマットフィールド540、ベースオペレーションフィールド542、及びデータエレメント幅(W)フィールド564を含む。ベースオペレーションフィールド542は、プレフィクス符号化フィールド625、オペコードマップフィールド615、及びリアルオペコードフィールド630を含む。

【0103】

[レジスタインデックスフィールド]

図6Cは、本開示の一実施形態に係るレジスタインデックスフィールド544を形成する特定ベクトル向け命令フォーマット600の複数のフィールドを示すブロック図である。具体的には、レジスタインデックスフィールド544は、REXフィールド605、REX'フィールド610、MODR/M.r e gフィールド644、MODR/M.r / mフィールド646、VVVVフィールド620、xxxフィールド654、及びbbbフィールド656を含む。

【0104】

[増加オペレーションフィールド]

図6Dは、本開示の一実施形態に係る増加オペレーションフィールド550を形成する特定ベクトル向け命令フォーマット600の複数のフィールドを示すブロック図である。クラス(U)フィールド568が0を含む場合、これはEVEX.U0(クラスA568A)を意味し、1を含む場合、これはEVEX.U1(クラスB568B)を意味する。U=0、かつMODフィールド642が(非メモリアクセスオペレーションを意味する)11を含む場合、アルファフィールド552(EVEXバイト3、ビット[7]-EH)は、rsフィールド552Aと解釈される。rsフィールド552Aが1(ラウンド552A.1)を含む場合、ベータフィールド554(EVEXバイト3、ビット[6:4]-SSS)は、ラウンド制御フィールド554Aと解釈される。ラウンド制御フィールド554Aは、1ビットのSAEフィールド556及び2ビットのラウンドオペレーションフィールド558を含む。rsフィールド552Aが0(データ変換552A.2)を含む場合、ベータフィールド554(EVEXバイト3、ビット[6:4]-SSS)は、3ビットのデータ変換フィールド554Bと解釈される。U=0、かつMODフィールド642が00、01、又は10(メモリアクセスオペレーションを意味する)を含む場合、アルファフィールド552(EVEXバイト3、ビット[7]-EH)は、エビクションヒント(EH)フィールド552Bと解釈され、ベータフィールド554(EVEXバイト3、ビット[6:4]-SSS)は、3ビットデータ操作フィールド554Cと解釈される。

【0105】

U=1の場合、アルファフィールド552(EVEXバイト3、ビット[7]-EH)は、書き込みマスク制御(Z)フィールド552Cと解釈される。U=1、かつMODフィールド642が(非メモリアクセスオペレーションを意味する)11を含む場合、ベータフィールド554の一部(EVEXバイト3、ビット[4]-S<sub>0</sub>)は、RLフィールド557Aと解釈され、1(ラウンド557A.1)を含む場合、ベータフィールド554の残り(EVEXバイト3、ビット[6-5]-S<sub>2-1</sub>)は、ラウンドオペレーションフィールド559Aと解釈され、RLフィールド557Aが0(VSIZE557.A2)を含む場合、ベータフィールド554の残り(EVEXバイト3、ビット[6-5]-S<sub>2-1</sub>)は、ベクトル長フィールド559B(EVEXバイト3、ビット[6-5]-L<sub>1-0</sub>)と解釈される。U=1、かつMODフィールド642が00、01、又は10(メモリアクセスオペレーションを意味する)を含む場合、ベータフィールド554(EVEXバイト3、ビット[6:4]-SSS)は、ベクトル長フィールド559B(EVEXバイト3、ビット[6-5]-L<sub>1-0</sub>)及びブロードキャストフィールド557B(EVEXバイト3、ビット[4]-B)と解釈される。

10

20

30

40

50

## 【 0 1 0 6 】

[ 例示的なレジスタアーキテクチャ ]

図7は、本開示の一実施形態に係るレジスタアーキテクチャ700のブロック図である。示された実施形態には、512ビット幅の32個のベクトルレジスタ710が存在する。これらのレジスタは、zmm0からzmm31と記載される。下位16個のzmmレジスタの下位256ビットは、レジスタymm0 - 16にオーバーレイされる。下位16個のzmmレジスタの下位128ビット(ymmレジスタの下位128ビット)は、レジスタxmm0 - 15の上にオーバーレイされる。特定ベクトル向け命令フォーマット600は、以下の表に示されるように、これらのオーバーレイされたレジスタファイルにおいて動作する。

【表2】

調節可能ベクトル長	クラス	オペレーション	レジスタ
ベクトル長フィールド559Bを含まない命令テンプレート	A(図5A、U=0)	510、515、525、530	zmmレジスタ (ベクトル長は64バイト)
	B(図5B、U=1)	512	zmmレジスタ (ベクトル長は64バイト)
ベクトル長フィールド559Bを含む命令テンプレート	B(図5B、U=1)	517、527	ベクトル長フィールド559Bに応じて、zmm、ymm、又はxmmレジスタ (ベクトル長は64バイト、32バイト、又は16バイト)

## 【 0 1 0 7 】

換言すると、ベクトル長フィールド559Bは、最大長と1つ又は複数の他のより短い長さとの間で選択し、ここで、このようなより短い長さの各々は、先行の長さの半分の長さであり、ベクトル長フィールド559Bを有さない複数の命令のテンプレートは、最大ベクトル長において動作する。さらに、一実施形態において、特定ベクトル向け命令フォーマット600の複数のクラスB命令テンプレートは、パック型又はスカラ単/倍精度浮動小数点データ及びパック型又はスカラ整数データにおいて動作する。複数のスカラオペレーションは、zmm/ymm/xmmレジスタ内の最下位のデータエレメント位置で実行されるオペレーションである。より高位のデータエレメント位置は、命令の前のそれらと同じ状態のままに残される、又は実施形態に応じてゼロ書き込みが実行される。

## 【 0 1 0 8 】

書き込みマスクレジスタ715：示された実施形態において、8個の書き込みマスクレジスタが存在し(k0からk7)、各々のサイズは64ビットである。代替的な実施形態において、書き込みマスクレジスタ715は、16ビットサイズである。前述されたように、本開示の一実施形態において、ベクトルマスクレジスタk0は、書き込みマスクとして用いられることができない。通常k0を示す符号化が、書き込みマスクのために用いられる場合、これは、ハードウェアに組み込まれる書き込みマスク0xFFFFを選択し、当該命令に対し、有効に書き込みマスキングを無効化する。

## 【 0 1 0 9 】

汎用レジスタ 725 : 示された実施形態において、複数のメモリオペランドにアドレスする既存の複数の  $\times 86$  アドレス指定モードと共に用いられる 16 個の 64 ビット汎用レジスタが存在する。これらのレジスタは、RAX、RBX、RCX、RDX、RBP、RSI、RDI、RSP、及び R8 から R15 という名称で記載される。

#### 【0110】

MMX パック型整数フラットレジスタファイル 750 がエイリアスされるスカラ浮動小数点スタックレジスタファイル ( $\times 87$  スタック) 745 : 示された実施形態において、 $\times 87$  スタックは、 $\times 87$  命令セット拡張を用いて 32 / 64 / 80 ビット浮動小数点データに対してスカラ浮動小数点の複数のオペレーションを実行するために用いられる 8 エレメントスタックであり、複数の MMX レジスタは、64 ビットパック型整数データに対して複数のオペレーションを実行し、MMX 及び XMM レジスタの間で実行されるいくつかのオペレーションのための複数のオペランドを保持するために用いられる。

10

#### 【0111】

本開示の複数の代替的な実施形態は、より広い又はより狭い複数のレジスタを用いてよい。さらに、本開示の複数の代替的な実施形態は、より多くの、より少ない、又は異なる複数のレジスタファイル及び複数のレジスタを用いてよい。

#### 【0112】

[ 例示的なコアアーキテクチャ、プロセッサ、及びコンピュータアーキテクチャ ]

複数のプロセッサコアは、異なる複数の目的に対して複数の異なる態様で実装されてよく、複数の異なるプロセッサで実装されてよい。例えば、このような複数のコアの実装は、1) 汎用コンピューティング向けの汎用インオーダーコア、2) 汎用コンピューティング向けの高性能汎用アウトオブオーダーコア、3) 主にグラフィック及び / 又は科学用途 (スループット) コンピューティング向けの特定用途向けコアを含んでよい。複数の異なるプロセッサの実装は、1) 汎用コンピューティング向けの 1 つ又は複数の汎用インオーダーコア及び / 又は汎用コンピューティング向けの 1 つ又は複数の汎用アウトオブオーダーコアを含む CPU、並びに 2) 主にグラフィック及び / 又は科学用途 (スループット) 向けの 1 つ又は複数の特定用途向けコアを含むコプロセッサを含んでよい。このような複数の異なるプロセッサは、複数の異なるコンピュータシステムアーキテクチャをもたらし、複数の異なるコンピュータシステムアーキテクチャは、1) CPU とは別個のチップ上のコプロセッサ、2) CPU と同じパッケージにおける別個のダイ上のコプロセッサ、3) CPU と同じダイ上のコプロセッサ (この場合、このようなコプロセッサは、場合によっては、集中画像表示及び / 又は科学用途 (スループット) ロジックのような特定用途向けロジック、又は特定用途向けコア等と称される)、及び 4) 同じダイ上に、説明された CPU (場合によっては、アプリケーションコア又はアプリケーションプロセッサと称される)、上述されたコプロセッサ、及び追加的な機能性を含み得るシステムオンチップを含んでよい。複数の例示的なコアアーキテクチャが、次に説明され、次いで、例示的な複数のプロセッサ及びコンピュータアーキテクチャが説明される。

20

30

#### 【0113】

[ 例示的なコアアーキテクチャ ]

インオーダー及びアウトオブオーダーコアのブロック図

40

図 8 A は、本開示の複数の実施形態に係る例示的なインオーダーパイプライン及び例示的なレジスタリネーミング、アウトオブオーダー発行 / 実行パイプラインの両方を示すブロック図である。図 8 B は、本開示の複数の実施形態に係るプロセッサに含まれるべきインオーダーアーキテクチャコアの例示的な実施形態及び例示的なレジスタリネーミング、アウトオブオーダー発行 / 実行アーキテクチャコアの両方を示すブロック図である。図 8 A - B における複数の実線のボックスは、インオーダーパイプライン及びインオーダーコアを示し、任意に追加された複数の破線のボックスは、レジスタリネーミング、アウトオブオーダー発行 / 実行パイプライン及びコアを示す。インオーダー態様はアウトオブオーダー態様のサブセットであるとして、アウトオブオーダー態様が説明される。

#### 【0114】

50

図 8 A において、プロセッサパイプライン 8 0 0 が、フェッチステージ 8 0 2、長さ復号ステージ 8 0 4、復号ステージ 8 0 6、配分ステージ 8 0 8、リネーミングステージ 8 1 0、スケジューリング（ディスパッチ又は発行としても知られる）ステージ 8 1 2、レジスタ読み出し/メモリ読み出しステージ 8 1 4、実行ステージ 8 1 6、ライトバック/メモリ書き込みステージ 8 1 8、例外処理ステージ 8 2 2、及びコミットステージ 8 2 4 を含む。

#### 【 0 1 1 5 】

図 8 B は、実行エンジンユニット 8 5 0 に結合されるフロントエンドユニット 8 3 0 を含むプロセッサコア 8 9 0 を示し、両方ともメモリユニット 8 7 0 に結合される。コア 8 9 0 は、縮小命令セットコンピューティング（RISC）コア、複合命令セットコンピューティング（CISC）コア、超長命令語（VLIW）コア、又はハイブリッドもしくは代替的な型のコアであってよい。さらに他の選択肢として、コア 8 9 0 は、例えば、ネットワーク又は通信コア、圧縮エンジン、コプロセッサコア、汎用コンピューティング画像処理ユニット（GPGPU）コア、グラフィックコア等のような特定用途向けコアであってよい。

#### 【 0 1 1 6 】

フロントエンドユニット 8 3 0 は、命令キャッシュユニット 8 3 4 と結合される分岐予測ユニット 8 3 2 を含み、命令キャッシュユニット 8 3 4 は命令トランシェーションルックアサイドバッファ（TLB）8 3 6 と結合され、命令トランシェーションルックアサイドバッファ 8 3 6 は命令フェッチユニット 8 3 8 と結合され、命令フェッチユニット 8 3 8 は復号ユニット 8 4 0 と結合される。復号ユニット 8 4 0（又はデコーダもしくはデコーダユニット）は、複数の命令（例えばマクロ命令）を復号し、1 つ又は複数のマイクロオペレーション、マイクロコードエントリポイント、マイクロ命令、他の命令、又は元の命令から復号され、又は他の方法でこれを反映し、又はこれから派生した他の制御信号を、出力として生成してよい。復号ユニット 8 4 0 は、様々な異なるメカニズムを用いて実装されてよい。複数の適したメカニズムの複数の例は、限定されるものではないが、ルックアップテーブル、ハードウェア実装、プログラマブルロジックアレイ（PLA）、マイクロコードリードオンリメモリ（ROM）等を含む。一実施形態において、コア 8 9 0 は、複数の特定のマクロ命令に対するマイクロコードを（例えば、復号ユニット 8 4 0、又は他の方法でフロントエンドユニット 8 3 0 内に）ストアする、マイクロコード ROM 又は他の媒体を含む。復号ユニット 8 4 0 は、実行エンジンユニット 8 5 0 内のリネーミング/アロケータユニット 8 5 2 と結合される。

#### 【 0 1 1 7 】

実行エンジンユニット 8 5 0 は、リタイアメントユニット 8 5 4 及び 1 つ又は複数のスケジューラユニット 8 5 6 のセットと結合されるリネーミング/アロケータユニット 8 5 2 を含む。スケジューラユニット 8 5 6 は、複数のリザーベーションステーション、中央命令ウィンドウ等を含む、任意の数の異なるスケジューラを表す。スケジューラユニット 8 5 6 は、物理レジスタファイルユニット 8 5 8 に結合される。複数の物理レジスタファイルユニット 8 5 8 の各々は、1 つ又は複数の物理レジスタファイル、スカラ整数、スカラ浮動小数点、パック型整数、パック型浮動小数点、ベクトル整数、ベクトル浮動小数点、ステータス（例えば、実行される次の命令のアドレスである命令ポインタ）等のような 1 つ又は複数の異なるデータ型をストアする異なるものを表す。一実施形態において、物理レジスタファイルユニット 8 5 8 は、ベクトルレジスタユニット、書き込みマスクレジスタユニット、及びスカラレジスタユニットを備える。これらのレジスタユニットは、複数のアーキテクチャベクトルレジスタ、ベクトルマスクレジスタ、及び汎用レジスタを提供してよい。物理レジスタファイルユニット 8 5 8 は、リタイアメントユニット 8 5 4 にオーバーラップされて、（例えば、リオーダバッファ及びリタイアメントレジスタファイルを用いて、フューチャファイル、履歴バッファ、及びリタイアメントレジスタファイルを用いて、レジスタマップ及び複数のレジスタのプールを用いる等）レジスタリネーミング及びアウトオブオーダー実行が実装されてよい様々な態様を示す。リタイアメントユニット 8

10

20

30

40

50

5 4 及び物理レジスタファイルユニット 8 5 8 は、実行クラスタ 8 6 0 と結合される。実行クラスタ 8 6 0 は、1 つ又は複数の実行ユニット 8 6 2 のセット及び 1 つ又は複数のメモリアクセスユニット 8 6 4 のセットを含む。実行ユニット 8 6 2 は、様々なオペレーション（例えば、シフト、加算、減算、乗算）を様々な型のデータ（例えば、スカラ浮動小数点、パック型整数、パック型浮動小数点、ベクトル整数、ベクトル浮動小数点）に対して実行してよい。いくつかの実施形態は、複数の具体的な機能又は複数の機能の複数のセット専用の、多数の実行ユニットを含んでよいが、複数の他の実施形態は、1 つだけの実行ユニット、又は、その全てで全ての機能を実行し得る複数の実行ユニットを含んでよい。スケジューラユニット 8 5 6、物理レジスタファイルユニット 8 5 8 及び実行クラスタ 8 6 0 は、場合によっては複数として示されるが、その理由は、複数の特定の実施形態は、特定の複数の型のデータ / オペレーションに対して、複数の別個のパイプラインを形成するからである（例えば、各々がこれら自身のスケジューラユニット、物理レジスタファイルユニット及び / 又は実行クラスタを有する、スカラ整数パイプライン、スカラ浮動小数点 / パック型整数 / パック型浮動小数点 / ベクトル整数 / ベクトル浮動小数点パイプライン、及び / 又はメモリアクセスパイプラインであり、別個のメモリアクセスパイプラインの場合、このパイプラインの実行クラスタのみがメモリアクセスユニット 8 6 4 を有する複数の特定の実施形態が実装される）。複数の別個のパイプラインが使用される場合、これらのパイプラインのうちの 1 つ又は複数は、アウトオブオーダー発行 / 実行であり、残りはインオーダーであってよいことも理解されたい。

10

#### 【 0 1 1 8 】

20

複数のメモリアクセスユニット 8 6 4 のセットは、メモリユニット 8 7 0 と結合される。メモリユニット 8 7 0 は、二次 ( L 2 ) キャッシュユニット 8 7 6 と結合されるデータキャッシュユニット 8 7 4 と結合される、データ T L B ユニット 8 7 2 を含む。例示的な一実施形態において、メモリアクセスユニット 8 6 4 は、ロードユニット、ストアアドレスユニット、及びストアデータユニットを含んでよく、それらの各々は、メモリユニット 8 7 0 内のデータ T L B ユニット 8 7 2 に結合される。命令キャッシュユニット 8 3 4 は、メモリユニット 8 7 0 内の二次 ( L 2 ) キャッシュユニット 8 7 6 とさらに結合される。L 2 キャッシュユニット 8 7 6 は、1 つ又は複数の他のレベルのキャッシュと結合され、最終的にメインメモリと結合される。

#### 【 0 1 1 9 】

30

例として、例示的なレジスタリネーミング、アウトオブオーダー発行 / 実行コアアーキテクチャは、以下のようにパイプライン 8 0 0 を実装してよい。1 ) 命令フェッチ 8 3 8 が、フェッチ及び長さ復号ステージ 8 0 2 及び 8 0 4 を実行する、2 ) 復号ユニット 8 4 0 が、復号ステージ 8 0 6 を実行する、3 ) リネーミング / アロケータユニット 8 5 2 が、配分ステージ 8 0 8 及びリネーミングステージ 8 1 0 を実行する、4 ) スケジューラユニット 8 5 6 が、スケジューリングステージ 8 1 2 を実行する、5 ) 物理レジスタファイルユニット 8 5 8 及びメモリユニット 8 7 0 が、レジスタ読み出し / メモリ読み出しステージ 8 1 4 を実行し、実行クラスタ 8 6 0 が、実行ステージ 8 1 6 を実行する、6 ) メモリユニット 8 7 0 及び物理レジスタファイルユニット 8 5 8 が、ライトバック / メモリ書き込みステージ 8 1 8 を実行する、7 ) 複数の様々なユニットが、例外処理ステージ 8 2 2 に関わってよく、8 ) リタイアメントユニット 8 5 4 及び物理レジスタファイルユニット 8 5 8 が、コミットステージ 8 2 4 を実行する。

40

#### 【 0 1 2 0 】

コア 8 9 0 は、本明細書において説明される命令を含む 1 つ又は複数の命令セット（例えば、（複数のより新たなバージョンを伴う、いくつかの拡張が追加された）× 8 6 命令セット、カリフォルニア州サンバールの M I P S テクノロジーズの M I P S 命令セット、カリフォルニア州サンバールの A R M ホールディングスの ( N E O N のような任意のさらなる拡張を有する ) A R M 命令セット ) をサポートしてよい。一実施形態において、コア 8 9 0 は、パック型データ命令セット拡張（例えば、A V X 1、A V X 2）をサポートするロジックを含み、これによって、多くのマルチメディアアプリケーションによって

50

用いられる複数のオペレーションが、パック型データを用いて実行されることが可能となる。

【 0 1 2 1 】

コアは、マルチスレッディング（複数のオペレーション又は複数のスレッドの2つ又はそれより多くの並列セットを実行すること）をサポートしてよく、時分割マルチスレッディング、同時マルチスレッディング（単一の物理コアが複数のスレッドの各々に対して論理のコアを提供することにより、物理コアが同時マルチスレッディングを実行すること）又はこれらの組み合わせ（例えば、時分割フェッチ及び復号化、その後、インテル（登録商標）ハイパースレッディングテクノロジーなどでの同時マルチスレッディング）を含む様々な態様で、マルチスレッディングを実行してよいことを理解されたい。

10

【 0 1 2 2 】

アウトオブオーダー実行との関連で、レジスタリネーミングを説明するが、レジスタリネーミングは、インオーダーアーキテクチャで用いられてよいことを理解されたい。プロセッサの示された実施形態は、別個の命令ならびにデータキャッシュユニット834 / 874及び共有L2キャッシュユニット876をさらに含むが、複数の代替的な実施形態は、命令及びデータの両方のための、例えば、一次（L1）内部キャッシュのような単一の内部キャッシュ、又は複数のレベルの内部キャッシュを有してよい。いくつかの実施形態において、システムは内部キャッシュと、コア及び/又はプロセッサの外部にある外部キャッシュとの組み合わせを含んでよい。代わりに、全てのキャッシュはコア及び/又はプロセッサの外部にあってよい。

20

【 0 1 2 3 】

[ 具体的な例示的インオーダーコアアーキテクチャ ]

図9A - Bは、より具体的な例示的インオーダーコアアーキテクチャのブロック図を示し、ここで、コアは、チップ内のいくつかの論理ブロック（同じ型及び/又は異なる型の他の複数のコアを含む）の中の1つであってよい。複数の論理ブロックは、用途に応じて、高帯域幅の相互接続ネットワーク（例えば、リングネットワーク）を通して、いくつかの固定機能ロジック、メモリI/Oインタフェース、及び他の必要なI/Oロジックと通信を行う。

【 0 1 2 4 】

図9Aは、本開示の複数の実施形態に係る単一のプロセッサコアを、そのオンダイ相互接続ネットワーク902への接続及びその二次（L2）キャッシュのローカルサブセット904と共に示すブロック図である。一実施形態において、命令復号ユニット900は、パック型データ命令セット拡張を有するx86命令セットをサポートする。L1キャッシュ906が、スカラ及びベクトルユニットに入るキャッシュメモリへの低レイテンシアクセスを可能にする。一実施形態において、（設計の単純化のために）スカラユニット908及びベクトルユニット910は、別個のレジスタセット（それぞれ、複数のスカラレジスタ912及び複数のベクトルレジスタ914）を用い、これらの間で転送されるデータは、一次（L1）キャッシュ906のメモリに書き込まれてから再読み出しされるが、本開示の複数の代替的な実施形態は、異なるアプローチ（例えば、単一のレジスタセットを用いる、又は書き込み及び再読み出しを行うことなく、2つのレジスタファイル間におけるデータ転送を可能とする通信パスを含む）を用いてよい。

30

40

【 0 1 2 5 】

L2キャッシュのローカルサブセット904は、プロセッサコア毎に1つの別個のローカルサブセットに分割されるグローバルL2キャッシュの一部である。各プロセッサコアは、L2キャッシュのそれ自身のローカルサブセット904へのダイレクトアクセスパスを有する。プロセッサコアによって読み出されたデータは、そのL2キャッシュサブセット904内にストアされ、他のプロセッサコアによるこれら自身のローカルL2キャッシュサブセットへのアクセスと並列に、迅速なアクセスが可能である。プロセッサコアによって書き込まれたデータは、必要に応じて、それ自身のL2キャッシュサブセット904にストアされ、他のサブセットからフラッシュされる。リングネットワークは、共有デー

50

タに対するコヒーレンシを保証する。リングネットワークは、双方向性であり、複数のプロセッサコア、L2キャッシュ、及び他の論理ブロックのようなエージェントが、チップ内で互いに通信を行うことを可能にする。各リングデータパスは、各方向につき1012ビット幅である。

#### 【0126】

図9Bは、本開示の複数の実施形態に係る図9Aにおけるプロセッサコアの一部の拡大図である。図9Bは、L1キャッシュ906の一部であるL1データキャッシュ906A、ならびにベクトルユニット910及び複数のベクトルレジスタ914に関するさらなる詳細を含む。具体的には、ベクトルユニット910は、整数、単精度浮動小数、及び倍精度浮動小数命令のうちの1つ又は複数を実行する、16幅ベクトル処理ユニット(VPU) (16幅ALU928参照)である。VPUは、スイズルユニット920によるレジスタ入力のスイズル、数値変換ユニット922A-Bによる数値変換、及び複製ユニット924によるメモリ入力に対する複製をサポートする。書き込みマスクレジスタ926は、結果的な複数のベクトル書き込みの叙述を可能とする。

#### 【0127】

図10は、本開示の複数の実施形態に係るプロセッサ1000のブロック図であり、当該プロセッサは、1つより多くのコアを有してよく、集積メモリコントローラを有してよく、集中画像表示を有してよい。図10における複数の実線のボックスは、単一のコア1002A、システムエージェント1010、1つ又は複数のバスコントローラユニット1016のセットを備えるプロセッサ1000を示し、任意に追加された複数の破線のボックスは、複数のコア1002A-N、システムエージェントユニット1010内の1つ又は複数の集積メモリコントローラユニット1014のセット、及び特定用途向けロジック1008を備える代替的なプロセッサ1000を示す。

#### 【0128】

従って、プロセッサ1000の複数の異なる実装は、1)(1つ又は複数のコアを含み得る)集中画像表示及び/又は科学用途(スループット)ロジックである特定用途向けロジック1008、及び1つ又は複数の汎用コア(例えば、複数の汎用インオーダーコア、複数の汎用アウトオブオーダーコア、これら2つの組み合わせ)である複数のコア1002A-Nを有するCPU、2)主にグラフィック及び/又は科学用途(スループット)向けの多数の特定用途向けコアである複数のコア1002A-Nを有するコプロセッサ、及び3)多数の汎用インオーダーコアである複数のコア1002A-Nを有するコプロセッサを含んでよい。従って、プロセッサ1000は、汎用プロセッサ、コプロセッサ、又は例えば、ネットワーク又は通信プロセッサ、圧縮エンジン、グラフィックプロセッサ、GPGPU(汎用画像処理ユニット)、ハイスループット多集積コア(MIC)コプロセッサ(30又はそれより多くのコアを含む)、組み込みプロセッサ等のような特定用途向けプロセッサであってよい。プロセッサは、1つ又は複数のチップ上に実装されてよい。プロセッサ1000は、例えば、BiCMOS、CMOS、又はNMOSのような、多数の処理技術のいずれかをを用いた1つ又は複数の基板の一部であってよく、及び/又はその上に実装されてよい。

#### 【0129】

メモリ階層は、複数のコア内における1つ又は複数のレベルのキャッシュ、1つ又は複数の共有キャッシュユニット1006のセット、及び複数の集積メモリコントローラユニット1014のセットに結合される外部メモリ(不図示)を含む。共有キャッシュユニット1006のセットは、二次(L2)、三次(L3)、四次(L4)、又は他のレベルのキャッシュ等のような1つ又は複数の中レベルキャッシュ、ラストレベルキャッシュ(LLC)、及び/又はこれらの組み合わせを含んでよい。一実施形態において、リングベースの相互接続ユニット1012が、集中画像表示ロジック1008、複数の共有キャッシュユニット1006のセット、及びシステムエージェントユニット1010/集積メモリコントローラユニット1014を相互接続するが、複数の代替的な実施形態は、このようなユニットを相互接続する任意の数の周知技術を用いてよい。一実施形態において、1つ

10

20

30

40

50

又は複数のキャッシュユニット1006とコア1002-A-Nとの間において、コヒーレンシが維持される。

【0130】

いくつかの実施形態において、コア1002A-Nのうちの1つ又は複数は、マルチスレッディングが可能である。システムエージェント1010は、コア1002A-Nを調整及び動作させるこれらのコンポーネントを含む。システムエージェントユニット1010は、例えば、電力制御ユニット(PCU)及びディスプレイユニットを含んでよい。PCUは、コア1002A-N及び集中画像表示ロジック1008の電力状態の調整に必要なロジック及び複数のコンポーネントであってよく、又はこれらを含んでよい。ディスプレイユニットは、1つ又は複数の外部接続されたディスプレイを駆動するためのものである。

10

【0131】

コア1002A-Nは、アーキテクチャ命令セットに関してホモジニアス又はヘテロジニアスであってよく、すなわち、コア1002A-Nのうち2つ又はそれより多くは、同じ命令セットを実行可能であってよく、他のものは、当該命令セット又は異なる命令セットのサブセットのみを実行可能であってよい。

【0132】

[例示的なコンピュータアーキテクチャ]

図11-14は、複数の例示的なコンピュータアーキテクチャのブロック図である。ラップトップ、デスクトップ、ハンドヘルド型PC、パーソナルデジタルアシスタント、エンジニアリングワークステーション、サーバ、ネットワークデバイス、ネットワークハブ、スイッチ、組み込みプロセッサ、デジタル信号プロセッサ(DSP)、グラフィックデバイス、ビデオゲームデバイス、セットトップボックス、マイクロコントローラ、携帯電話、ポータブルメディアプレイヤー、ハンドヘルドデバイス、及び様々な他の電子デバイスのための当技術分野において公知の他の複数のシステム設計及び構成も、適している。概して、本明細書において開示されるように、プロセッサ及び/又は他の実行ロジックを組み込むことが可能な多種多様なシステム又は電子デバイスが、概して適している。

20

【0133】

ここで図11を参照すると、本開示の一実施形態に係るシステム1100のブロック図が示される。システム1100は、コントローラハブ1120に結合される1つ又は複数のプロセッサ1110、1115を含んでよい。一実施形態において、コントローラハブ1120は、グラフィックメモリコントローラハブ(GMCH)1190及び入力/出力ハブ(IOH)1150(複数の別個のチップ上にあってよい)を含む。GMCH1190は、メモリ1140及びコプロセッサ1145が結合される複数のメモリ及びグラフィックコントローラを含む。IOH1150は、複数の入力/出力(I/O)デバイス1160をGMCH1190に結合する。代わりに、メモリ及びグラフィックコントローラの一方又は両方は、(本明細書において説明されるように)プロセッサ内で集積され、メモリ1140及びコプロセッサ1145は、プロセッサ1110と、IOH1150を有する単一のチップ内のコントローラハブ1120とに直接結合される。メモリ1140は、スイッチ制御モジュール1140A(及び/又はデータハザード解消モジュール)を含んでよく、これにより、例えば、実行された場合に、プロセッサに本開示の任意の方法を実行させるコードをストアする。

30

40

【0134】

複数の追加のプロセッサ1115が任意の性質であることが、図11において、破線で示される。各プロセッサ1110、1115は、本明細書において説明される処理コアのうちの1つ又は複数を含んでよく、プロセッサ1000の何らかのバージョンであってよい。

【0135】

メモリ1140は、例えば、ダイナミックランダムアクセスメモリ(DRAM)、相変化メモリ(PCM)又はこれら2つの組み合わせであってよい。少なくとも1つの実施形

50

態に対して、コントローラハブ1120は、フロントサイドバス(FSB)のようなマルチドロップバス、クイックバスインタコネクタ(QPI)のようなポイントツーポイントインタフェース、又は同様の接続部1195を介して、プロセッサ1110、1115と通信を行う。

【0136】

一実施形態において、コプロセッサ1145は、例えば、ハイスループットMICプロセッサ、ネットワーク又は通信プロセッサ、圧縮エンジン、グラフィックプロセッサ、GPGPU、組み込みプロセッサ等のような特定用途向けプロセッサである。一実施形態において、コントローラハブ1120は、集中画像表示アクセラレータを含んでよい。

【0137】

物理リソース1110、1115の間には、アーキテクチャ上、マイクロアーキテクチャ上、熱的、電力消費等の特性を含む利点の様々な基準に関して、様々な相違が存在し得る。

【0138】

一実施形態において、プロセッサ1110は、一般的な型の複数のデータ処理オペレーションを制御する複数の命令を実行する。複数のコプロセッサ命令が、複数の命令内に組み込まれてよい。プロセッサ1110は、これらのコプロセッサ命令を、取り付けられたコプロセッサ1145によって実行されるべき型のものであると認識する。従って、プロセッサ1110は、コプロセッサバス又は他のインタコネクタ上において、これらのコプロセッサ命令(又は複数のコプロセッサ命令を表す制御信号)を、コプロセッサ1145に発行する。コプロセッサ1145は、複数のコプロセッサ命令を受け付け、受信されたコプロセッサ命令を実行する。

【0139】

ここで図12を参照すると、本開示の実施形態に係る第1のより具体的な例示的システム1200のブロック図が示される。図12に示されるように、マルチプロセッサシステム1200は、ポイントツーポイントインタコネクタシステムであり、ポイントツーポイントインタコネクタ1250を介して結合される第1のプロセッサ1270及び第2のプロセッサ1280を含む。プロセッサ1270及び1280の各々は、プロセッサ1000の何らかのバージョンであってよい。本開示の一実施形態において、プロセッサ1270及び1280は、それぞれ、プロセッサ1110及び1115であり、コプロセッサ1238は、コプロセッサ1145である。他の実施形態において、プロセッサ1270及び1280は、それぞれ、プロセッサ1110及びコプロセッサ1145である。

【0140】

プロセッサ1270及び1280は、それぞれ、集積メモリコントローラ(IMC)ユニット1272及び1282を含むものとして示される。プロセッサ1270は、そのバスコントローラユニットの一部として、ポイントツーポイント(P-P)インタフェース1276及び1278をさらに含み、同様に、第2のプロセッサ1280は、P-Pインタフェース1286及び1288を含む。プロセッサ1270、1280は、P-Pインタフェース回路1278、1288を用いて、ポイントツーポイント(P-P)インタフェース1250を介して情報を交換してよい。図12に示されるように、IMC1272及び1282は、プロセッサをそれぞれのメモリ、すなわち、メモリ1232及びメモリ1234に結合し、これらは、それぞれのプロセッサにローカルに取り付けられたメインメモリの一部であってよい。

【0141】

プロセッサ1270、1280は、各々、ポイントツーポイントインタフェース回路1276、1294、1286、1298を用いて、個々のP-Pインタフェース1252、1254を介してチップセット1290と情報を交換してよい。チップセット1290は、任意に、高性能インタフェース1239を介して、コプロセッサ1238と情報を交換してよい。一実施形態において、コプロセッサ1238は、例えば、ハイスループットMICプロセッサ、ネットワーク又は通信プロセッサ、圧縮エンジン、グラフィックプロ

10

20

30

40

50

セッサ、G P G P U、組み込みプロセッサ等のような特定用途向けプロセッサである。

【0142】

共有キャッシュ(不図示)は、いずれかのプロセッサに含まれてよく、又は両方のプロセッサの外部にあってよく、さらに、P-P相互接続を介して複数のプロセッサと接続されてよく、これにより、プロセッサが低電力モードに置かれた場合に、いずれか又は両方のプロセッサのローカルキャッシュ情報が、共有キャッシュ内にストアされ得る。

【0143】

チップセット1290は、インタフェース1296を介して第1のバス1216に結合されてよい。一実施形態において、第1のバス1216は、ペリフェラルコンポーネントインタコネクト(PCI)バス、もしくはPCIエクスプレスバス又は他の第3世代I/O相互接続バスのようなバスであってよいが、本開示の範囲はこれらのバスに限定されるものではない。

【0144】

図12に示されるように、様々なI/Oデバイス1214は、第1のバス1216を第2のバス1220に結合するバスブリッジ1218と共に、第1のバス1216に結合されてよい。一実施形態において、コプロセッサ、ハイスループットMICプロセッサ、G P G P U、アクセラレータ(例えば、グラフィックアクセラレータ又はデジタル信号処理(DSP)ユニットのような)、フィールドプログラマブルゲートアレイ、又は任意の他のプロセッサのような1つ又は複数の追加のプロセッサ1215が、第1のバス1216に結合される。一実施形態において、第2のバス1220は、ローピンカウント(LPC)バスであってよい。一実施形態において、例えば、キーボード及び/又はマウス1222、通信デバイス1227、ならびに命令/コード及びデータ1230を含んでよいディスクドライブ又は他の大容量ストレージデバイスのようなストレージユニット1228を含む様々なデバイスが、第2のバス1220に結合されてよい。さらに、オーディオI/O1224は、第2のバス1220に結合されてよい。なお、他の複数のアーキテクチャが、適用可能である。例えば、図12のポイントツーポイントアーキテクチャの代わりに、システムがマルチドロップバス又は他のこのようなアーキテクチャを実装してよい。

【0145】

ここで図13を参照すると、本開示の実施形態に係る第2のより具体的な例示的システム1300のブロック図が示される。図12及び13において、複数の同様の要素は同様の参照符号を有し、図12の複数の特定の態様は、図13の他の複数の態様への妨げとならないよう、図13から省略されている。

【0146】

図13は、プロセッサ1270、1280が集積メモリ及びI/O制御ロジック(「CL」)1272及び1282をそれぞれ含んでよいことを示す。従って、CL1272、1282は、複数の集積メモリコントローラユニットを含み、I/O制御ロジックを含む。図13は、メモリ1232、1234がCL1272、1282に結合されることのみならず、I/Oデバイス1314もが制御ロジック1272、1282に結合されることをも示す。レガシI/Oデバイス1315は、チップセット1290に結合される。

【0147】

ここで図14を参照すると、本開示の実施形態に係るS o C 1400のブロック図が示される。図10において、複数の同様の要素は、同様の参照符号を有する。また、複数の破線のボックスは、より高度な複数のS o Cにおいて、任意の機能である。図14において、相互接続ユニット1402は、1つ又は複数のコア202A-Nのセット、及び共有キャッシュユニット1006を含むアプリケーションプロセッサ1410、システムエージェントユニット1010、バスコントローラユニット1016、集積メモリコントローラユニット1014、集中画像表示ロジック、画像プロセッサ、オーディオプロセッサ、及びビデオプロセッサを含み得る1つ又は複数のコプロセッサ1420のセット、スタティックランダムアクセスメモリ(SRAM)ユニット1430、ダイレクトメモリアクセス(DMA)ユニット1432、及び1つ又は複数の外部ディスプレイと結合するための

10

20

30

40

50

ディスプレイユニット1440と結合されている。一実施形態において、コプロセッサ1420は、例えば、ネットワーク又は通信プロセッサ、圧縮エンジン、GPGPU、ハイスループットMICプロセッサ、組み込みプロセッサ等のような特定用途向けプロセッサを含む。

#### 【0148】

本明細書に開示される（例えば、複数のメカニズムの）実施形態は、ハードウェア、ソフトウェア、ファームウェア、又はこのような複数の実装アプローチの組み合わせで実装されてよい。本開示の複数の実施形態は、少なくとも1つのプロセッサ、（揮発性及び不揮発性メモリ及び/又は複数のストレージエレメントを含む）ストレージシステム、少なくとも1つの入力デバイス及び少なくとも1つの出力デバイスを備える複数のプログラマブルシステム上で実行する複数のコンピュータプログラム又はプログラムコードとして実装されてよい。

10

#### 【0149】

図12に示されるコード1230のようなプログラムコードは、本明細書で説明される複数の機能を実行し、出力情報を生成する複数の入力命令に適用されてよい。出力情報は、公知の態様で、1つ又は複数の出力デバイスに適用されてよい。この用途の複数の目的のために、処理システムは、例えば、デジタル信号プロセッサ(DSP)、マイクロコントローラ、特定用途向け集積回路(ASIC)、又はマイクロプロセッサのようなプロセッサを有する任意のシステムを含む。

#### 【0150】

プログラムコードは、処理システムと通信を行うべく、高水準の手順型又はオブジェクト指向プログラミング言語で実装されてよい。プログラムコードは、所望の場合には、アセンブリ又は機械言語で実装されてもよい。実際には、本明細書で説明される複数のメカニズムの範囲は、いずれの特定のプログラミング言語にも限定されるものではない。いずれの場合であっても、言語は、コンパイラ型又はインタプリタ型言語であってよい。

20

#### 【0151】

少なくとも1つの実施形態の1つ又は複数の態様は、プロセッサ内の様々なロジックを表す機械可読媒体にストアされた代表的な複数の命令によって実装されてよく、これらは、機械によって読み出された場合に、機械にロジックを製造させることにより、本明細書で説明される複数の技術を実行する。「IPコア」として公知であるこのような複数の表現は、有形の機械可読媒体にストアされてよく、様々な顧客、又は実際にロジック又はプロセッサを作成する複数の製造機械にロードする複数の製造設備に供給されてよい。

30

#### 【0152】

このような機械可読記憶媒体は、ハードディスク、フロッピ(登録商標)ディスク、光ディスク、コンパクトディスクリードオンリメモリ(CD-ROM)、書き換え可能コンパクトディスク(CD-RW)、及び光磁気ディスクを含む任意の他のタイプのディスクのような記憶媒体、リードオンリメモリ(ROM)、ダイナミックランダムアクセスメモリ(DRAM)、スタティックランダムアクセスメモリ(SRAM)のようなランダムアクセスメモリ(RAM)、消去可能プログラマブルリードオンリメモリ(EPROM)、フラッシュメモリ、電氣的消去可能プログラマブルリードオンリメモリ(EEPROM)、及び相変化メモリ(PCM)のような半導体デバイス、磁気もしくは光カード、又は複数の電子的命令のストアに適した任意の他のタイプの媒体を含む、機械又はデバイスにより製造又は形成される、非一時的で有形の複数の構成の物品を含んでよいが、これらに限定されるものではない。

40

#### 【0153】

従って、本開示の複数の実施形態は、複数の命令を含み、又は本明細書で説明される複数の構造、複数の回路、複数の装置、複数のプロセッサ及び/又はシステムの複数の機能を定義するハードウェア記述言語(HDL)のような設計データを含む、非一時的な有形の機械可読媒体をさらに含む。このような複数の実施形態は、プログラム製品と称されることもある。

50

## 【 0 1 5 4 】

[ エミュレート ( バイナリ変換、コードモーフィング等を含む ) ]

いくつかの場合において、命令コンバータは、ソース命令セットからターゲット命令セットに命令を変換するために用いられてよい。例えば、命令コンバータは、命令を、コアにより処理されるべき他の1つ又は複数の他の命令にトランスレート ( 例えば、静的バイナリ変換、動的コンパイルを含む動的バイナリ変換を用いて )、モーフィング、エミュレート、又は他の方法で変換してよい。命令コンバータは、ソフトウェア、ハードウェア、ファームウェア、又はこれらの組み合わせで実装されてよい。命令コンバータは、プロセッサ上であってよく、プロセッサ外であってよく、又は部分的にプロセッサ上かつ部分的にプロセッサ外であってよい。

10

## 【 0 1 5 5 】

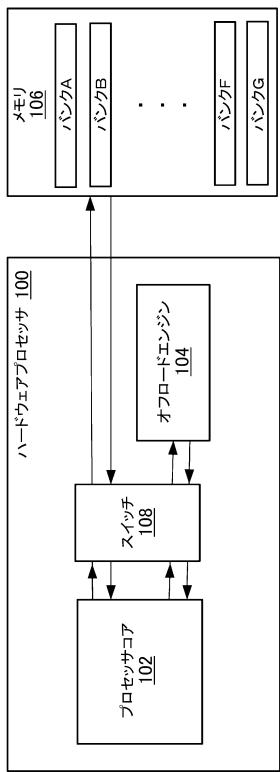
図15は、本開示の複数の実施形態に係るソース命令セットの複数のバイナリ命令をターゲット命令セットの複数のバイナリ命令に変換するソフトウェア命令コンバータの使用を対比したブロック図である。示された実施形態において、命令コンバータはソフトウェア命令コンバータであるが、代わりに、命令コンバータは、ソフトウェア、ファームウェア、ハードウェア、又はこれらの様々な組み合わせで実装されてよい。図15は、少なくとも1つのx86命令セットコアを有するプロセッサ1516によってネイティブで実行可能なx86バイナリコード1506を生成するために、高水準言語1502のプログラムが、x86コンパイラ1504を用いてコンパイルされてよいことを示す。少なくとも1つのx86命令セットコアを有するプロセッサ1516は、少なくとも1つのx86命令セットコアを有するインテル社製プロセッサと実質的に同じ結果を実現すべく、(1)インテルx86命令セットコアの命令セットの大部分、もしくは(2)少なくとも1つのx86命令セットコアを有するインテル社製プロセッサ上で動作することをターゲットとしたアプリケーション又は他のソフトウェアのオブジェクトコードバージョンを互換的に実行するか、又は他の方法で処理することによって、少なくとも1つのx86命令セットコアを有するインテル社製プロセッサと実質的に同じ機能を実行可能な任意のプロセッサを表す。x86コンパイラ1504は、さらなるリンク処理を用いて、又は用いることなく、少なくとも1つのx86命令セットコアを有するプロセッサ1516上で実行可能なx86バイナリコード1506 ( 例えば、オブジェクトコード ) を生成するように動作可能なコンパイラを表す。同様に、図15は、少なくとも1つのx86命令セットコアを有さないプロセッサ1514 ( 例えば、カリフォルニア州のサンニールにあるMIPSテクノロジーのMIPS命令セットを実行する及び/又はカリフォルニア州のサンニールのARMホールディングスのARM命令セットを実行する複数のコアを有するプロセッサ ) によってネイティブで実行可能な代替的な命令セットバイナリコード1510を生成すべく、代替的な命令セットコンパイラ1508を用いてコンパイル可能な高水準言語1502のプログラムを示す。命令コンバータ1512は、x86バイナリコード1506を、x86命令セットコアを有さないプロセッサ1514によってネイティブで実行可能なコードに変換するために用いられる。この変換されたコードは、代替的な命令セットバイナリコード1510と同じである可能性が低い。なぜなら、この変換が可能な命令コンバータは、製造が難しいからである。しかしながら、変換されたコードは、一般的なオペレーションを実現し、代替的な命令セットからの複数の命令で構成される。従って、命令コンバータ1512は、エミュレート、シミュレーション又は任意の他の処理を通じて、x86命令セットプロセッサ又はコアを有さないプロセッサ又は他の電子デバイスがx86バイナリコード1506を実行することを可能にする、ソフトウェア、ファームウェア、ハードウェア、又はこれらの組み合わせを表す。

20

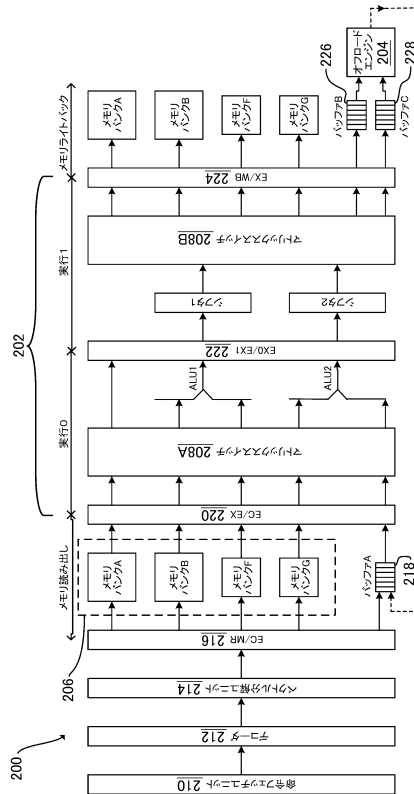
30

40

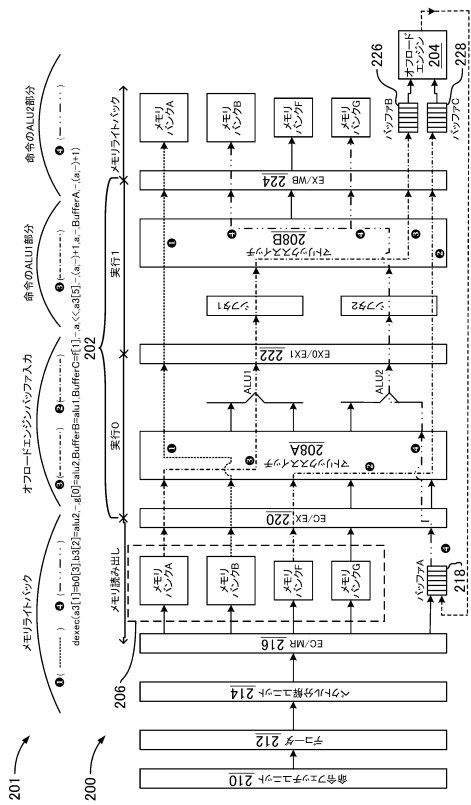
【図1】



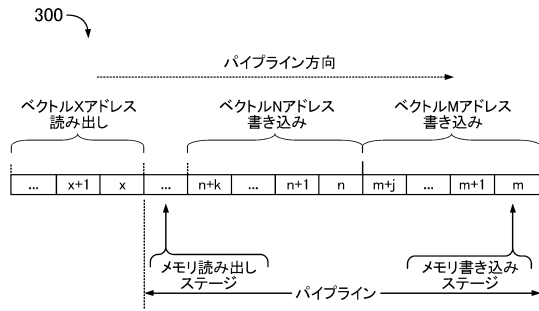
【図2A】



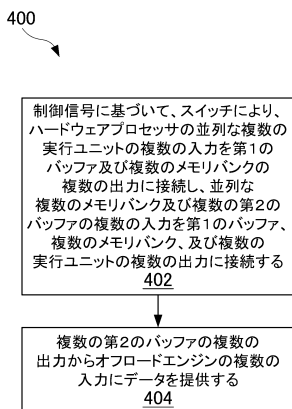
【図2B】



【図3】

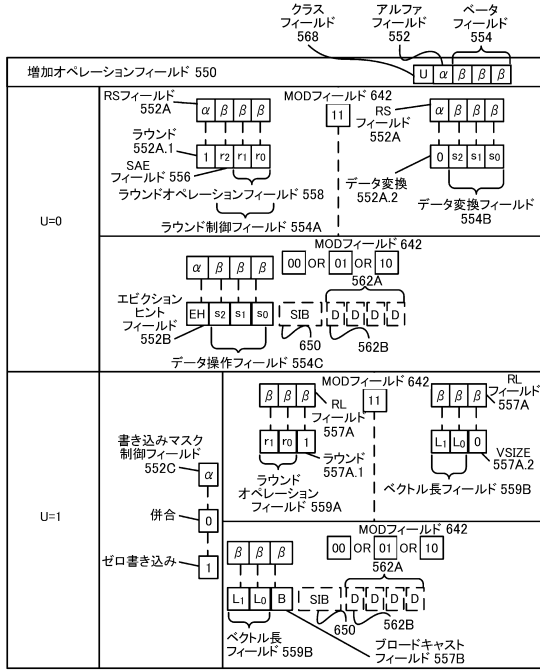


【図4】

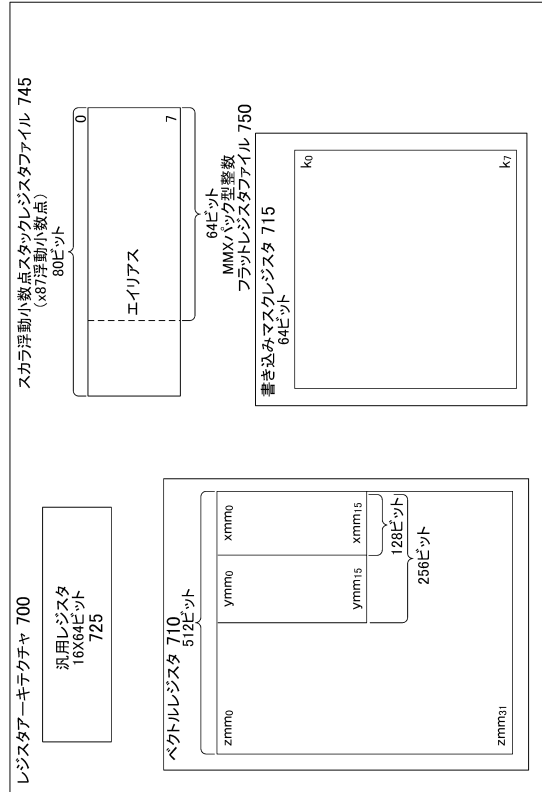




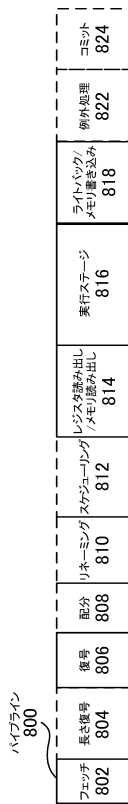
【図6D】



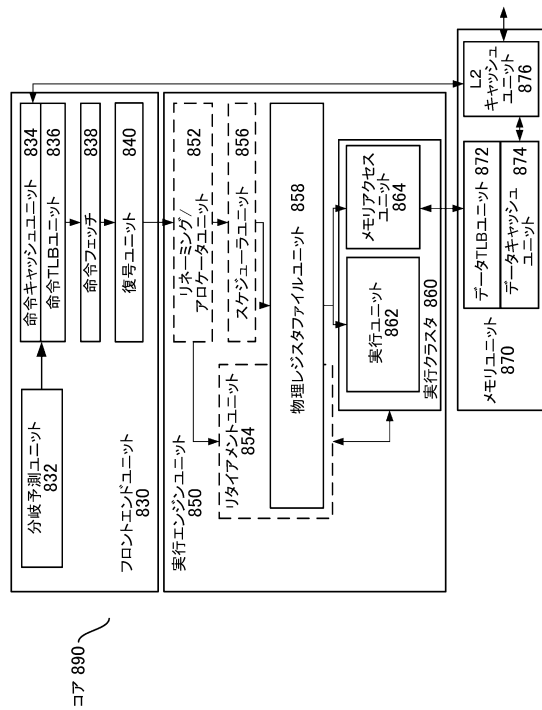
【図7】



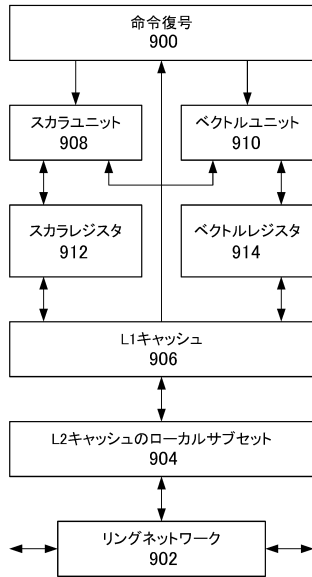
【図8A】



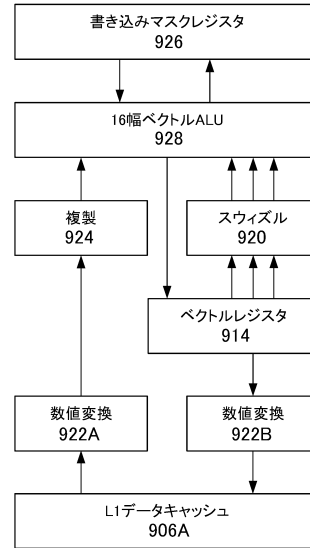
【図8B】



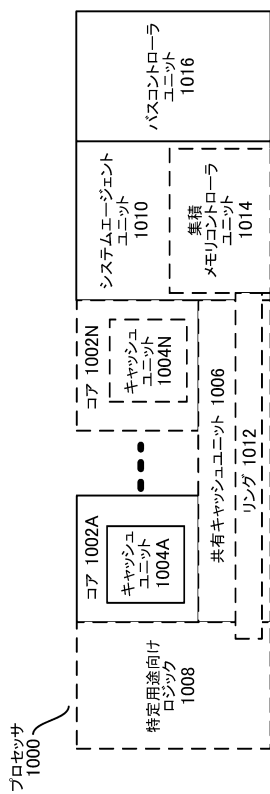
【図9A】



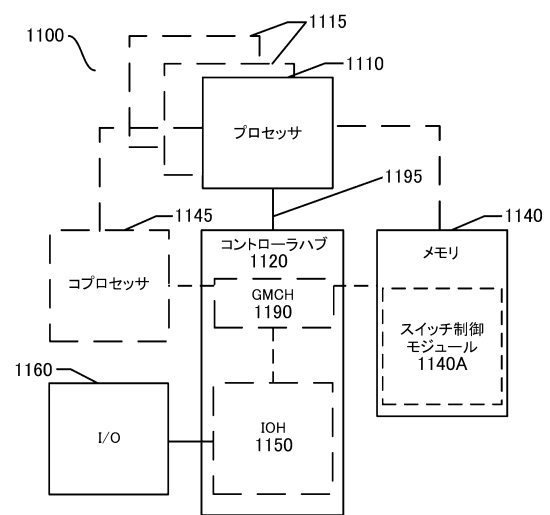
【図9B】



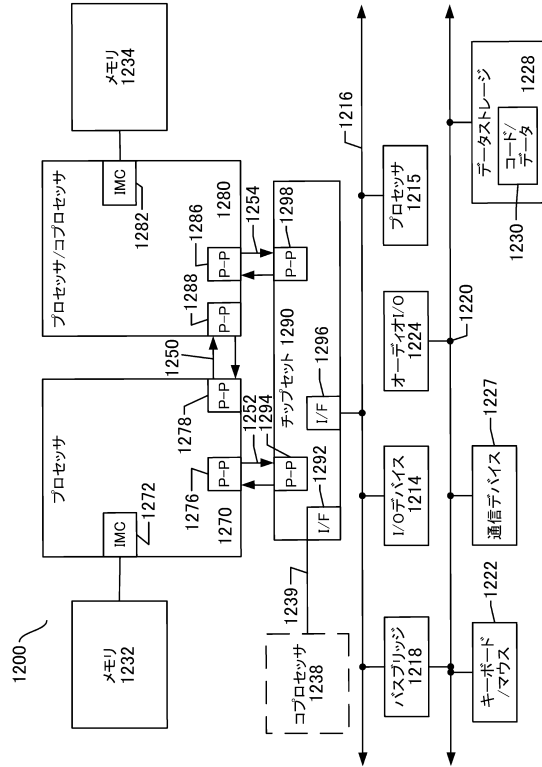
【図10】



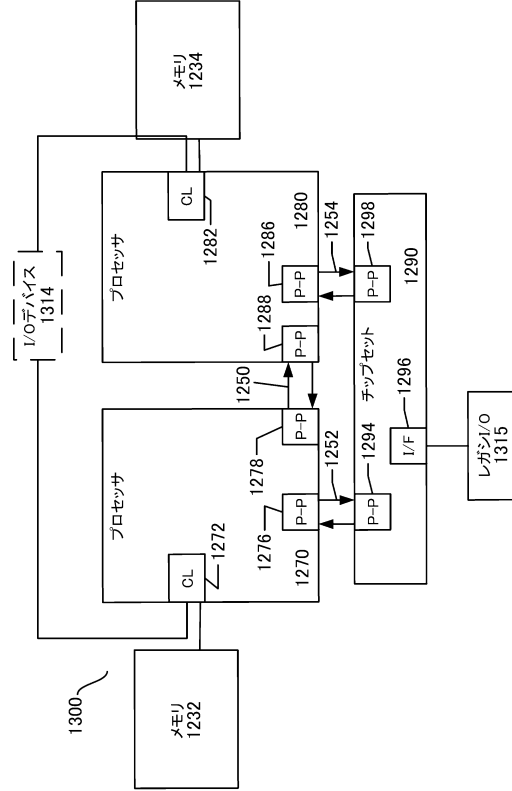
【図11】



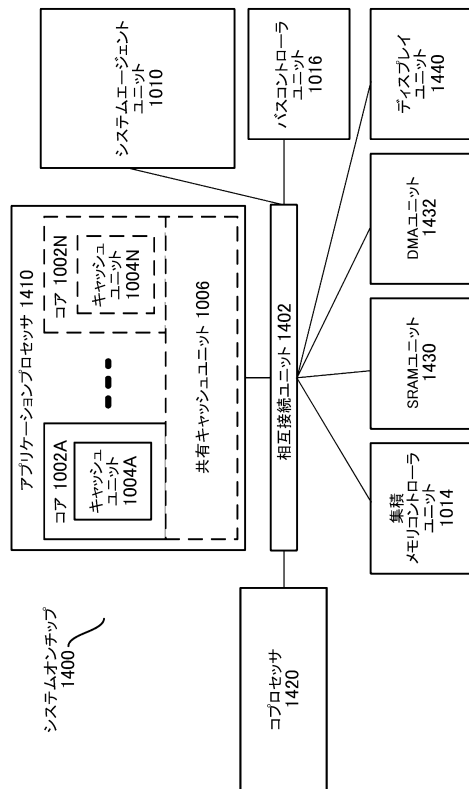
【図 12】



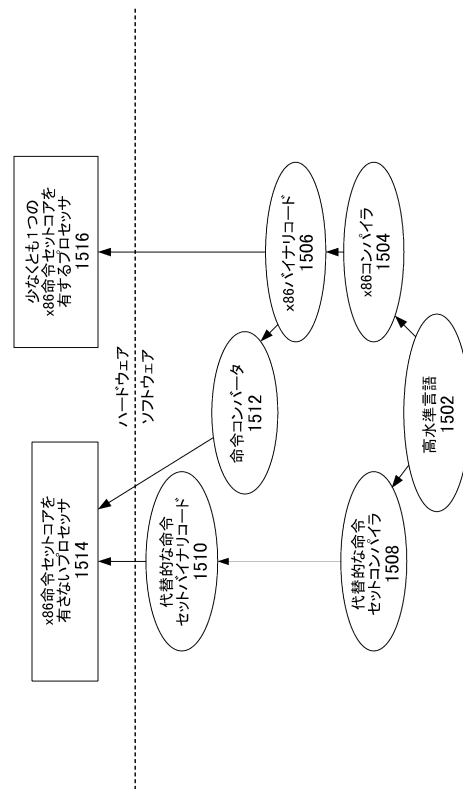
【図 13】



【図 14】



【図 15】



## フロントページの続き

- (72)発明者 ローラン、ピエール  
アメリカ合衆国 95052 カリフォルニア州・サンタクララ・ミッション カレッジ ブレ  
バード・2200 インテル・コーポレーション内
- (72)発明者 タデパリ、ハリ ケイ.  
アメリカ合衆国 95052 カリフォルニア州・サンタクララ・ミッション カレッジ ブレ  
バード・2200 インテル・コーポレーション内
- (72)発明者 ガティガー、プラサッド エム.  
アメリカ合衆国 95052 カリフォルニア州・サンタクララ・ミッション カレッジ ブレ  
バード・2200 インテル・コーポレーション内
- (72)発明者 オドゥヤー、ティー. ジェイ.  
アメリカ合衆国 95052 カリフォルニア州・サンタクララ・ミッション カレッジ ブレ  
バード・2200 インテル・コーポレーション内
- (72)発明者 ジリヤエフ、サージ  
アメリカ合衆国 95052 カリフォルニア州・サンタクララ・ミッション カレッジ ブレ  
バード・2200 インテル・コーポレーション内

審査官 原 忠

- (56)参考文献 特開2008-165780(JP,A)  
特表2010-541088(JP,A)  
特開2009-026135(JP,A)  
米国特許第05497467(US,A)  
特表2005-538439(JP,A)  
米国特許第06012139(US,A)  
特開2002-149399(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/38