

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2017-107587
(P2017-107587A)

(43) 公開日 平成29年6月15日(2017.6.15)

(51) Int.Cl.		F I		テーマコード (参考)
G06F 9/315 (2006.01)		G06F 9/30	340D	5B033
G06F 17/16 (2006.01)		G06F 17/16	E	5B056
		G06F 17/16	F	
		G06F 17/16	J	

審査請求 有 請求項の数 1 O L (全 27 頁)

(21) 出願番号 特願2017-21703 (P2017-21703)
 (22) 出願日 平成29年2月8日(2017.2.8)
 (62) 分割の表示 特願2015-534475 (P2015-534475) の分割
 原出願日 平成25年6月25日(2013.6.25)
 (31) 優先権主張番号 13/630, 131
 (32) 優先日 平成24年9月28日(2012.9.28)
 (33) 優先権主張国 米国 (US)

(71) 出願人 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 プロトニコフ、ミカイル
 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

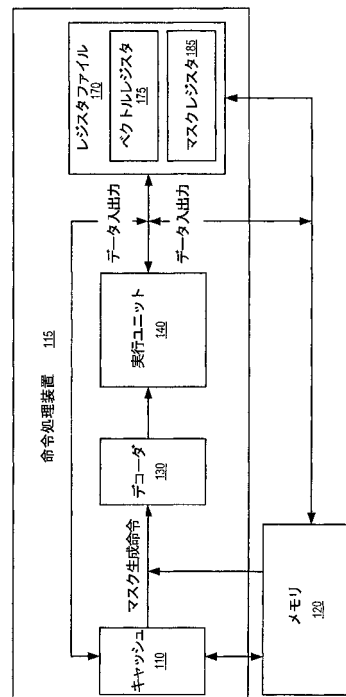
(54) 【発明の名称】 複数のビットを左にシフトし、複数の1を複数の下位ビットにプルインするための命令

(57) 【要約】 (修正有)

【課題】 複数のデータエレメントの配列に対する複数のベクトルオペレーションの効率性を改善するマスク生成命令を提供する。

【解決手段】 プロセッサは、複数のベクトルレジスタを含み、その1つは、配列のデータエレメントを格納する。プロセッサは、少なくとも第1のオペランド及び第2のオペランドを指定するマスク生成命令を受信する実行回路をさらに含む。マスク生成命令にตอบสนองして、実行回路は、第2のオペランドにおいて規定された回数だけ、第1のオペランドのビットを左にシフトし、第1のオペランドの最上位ビットが左からシフトアウトするたびに、1であるビットを右からプルインすることにより、結果を生成する。結果の各ビットは、配列の複数のデータエレメントの1つに対応する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

複数のベクトルレジスタと、前記複数のベクトルレジスタに連結される実行回路とを備え、前記複数のベクトルレジスタの1つは、アレイの複数のデータエレメントを格納し、前記実行回路は、

少なくとも第1のオペランド及び第2のオペランドを指定するマスク生成命令を受信し

、
前記マスク生成命令に応答して、前記第2のオペランドにおいて規定された回数だけ、前記第1のオペランドの複数のビットを左シフトし、前記第1のオペランドの最上位ビットがシフトアウトするたびに、1である最下位ビットをプルインすることにより、複数のビットを含む結果を生成し、前記結果の各ビットは、前記アレイの前記複数のデータエレメントの1つに対応し、

前記第2のオペランドは、ベクトルオペレーションの残りのループにおける残りのイタレーションの数を指定する、装置。

【請求項 2】

前記第2のオペランドは、前記ベクトルオペレーションに対して、ループ制限から現在のイタレーションカウントを減じた減算結果を指定する、請求項1に記載の装置。

【請求項 3】

前記第1のオペランド及び前記第2のオペランドの両方は、汎用レジスタである、請求項1または2に記載の装置。

【請求項 4】

前記第1のオペランドは、マスクレジスタであり、前記第2のオペランドは、汎用レジスタである、請求項1または2に記載の装置。

【請求項 5】

1つまたは複数の状態レジスタは、前記結果に基づいて設定される、請求項1から3のいずれか1項に記載の装置。

【請求項 6】

複数のベクトルレジスタと、前記複数のベクトルレジスタに連結される実行回路とを備え、前記複数のベクトルレジスタの1つは、アレイの複数のデータエレメントを格納し、前記実行回路は、

少なくとも第1のオペランド及び第2のオペランドを指定するマスク生成命令を受信し

、
前記マスク生成命令に応答して、前記第2のオペランドにおいて規定された回数だけ、前記第1のオペランドの複数のビットを左シフトし、前記第1のオペランドの最上位ビットがシフトアウトするたびに、1である最下位ビットをプルインすることにより、複数のビットを含む結果を生成し、前記結果の各ビットは、前記アレイの前記複数のデータエレメントの1つに対応し、

前記複数のベクトルレジスタは、第1のベクトルレジスタ及び第2のベクトルレジスタを含み、前記第2のオペランドは、ベクトル計算に対して、前記第1のベクトルレジスタ内の既存の複数のデータエレメントにマージされるべき前記第2のベクトルレジスタ内のデータエレメントの数を指定する、装置。

【請求項 7】

プロセッサによって、少なくとも第1のオペランド及び第2のオペランドを指定するマスク生成命令を受信する段階と、

前記マスク生成命令に応答して、前記第2のオペランドにおいて規定された回数だけ、前記第1のオペランドの複数のビットを左シフトし、前記第1のオペランドの最上位ビットがシフトアウトするたびに、1である最下位ビットをプルインすることにより、複数のビットを含む結果を生成するオペレーションを実行する段階とを備え、前記結果の各ビットは、アレイのデータエレメントに対応し、

前記第2のオペランドは、ベクトルオペレーションの残りのループにおける残りのイタ

10

20

30

40

50

レーションの数を指定する、方法。

【請求項 8】

前記第 2 のオペランドは、前記ベクトルオペレーションに対して、ループ制限から現在のイタレーションカウントを減じた減算結果を指定する、請求項 7 に記載の方法。

【請求項 9】

前記第 1 のオペランド及び前記第 2 のオペランドの両方は、汎用レジスタである、請求項 7 または 8 に記載の方法。

【請求項 10】

前記第 1 のオペランドは、マスクレジスタであり、前記第 2 のオペランドは、汎用レジスタである、請求項 7 または 8 に記載の方法。

10

【請求項 11】

前記結果に基づいて、1 つまたは複数の状態レジスタを修正する段階をさらに備える、請求項 7 から 10 のいずれか 1 項に記載の方法。

【請求項 12】

プロセッサによって、少なくとも第 1 のオペランド及び第 2 のオペランドを指定するマスク生成命令を受信する段階と、

前記マスク生成命令に応答して、前記第 2 のオペランドにおいて規定された回数だけ、前記第 1 のオペランドの複数のビットを左シフトし、前記第 1 のオペランドの最上位ビットがシフトアウトするたびに、1 である最下位ビットをプルインすることにより、複数のビットを含む結果を生成するオペレーションを実行する段階とを備え、前記結果の各ビットは、アレイのデータエレメントに対応し、

20

前記第 2 のオペランドは、ベクトル計算に対して、第 1 のベクトルレジスタ内の既存の複数のデータエレメントにマージされるべき、第 2 のベクトルレジスタ内のデータエレメントの数を指定する、方法。

【請求項 13】

ランダムアクセスメモリと、

前記ランダムアクセスメモリに連結されるプロセッサとを備え、前記プロセッサは、複数のベクトルレジスタと、前記複数のベクトルレジスタに連結される実行回路とを備え、前記複数のベクトルレジスタの 1 つは、アレイの複数のデータエレメントを格納し、前記実行回路は、

30

少なくとも第 1 のオペランド及び第 2 のオペランドを指定するマスク生成命令を受信し、

前記マスク生成命令に応答して、前記第 2 のオペランドにおいて規定された回数だけ、前記第 1 のオペランドの複数のビットを左シフトし、前記第 1 のオペランドの最上位ビットがシフトアウトするたびに、1 である最下位ビットをプルインすることにより、複数のビットを含む結果を生成し、前記結果の各ビットは、前記アレイの前記複数のデータエレメントの 1 つに対応し、

前記第 2 のオペランドは、ベクトルオペレーションの残りのループにおける残りのイタレーションの数を指定する、システム。

【請求項 14】

40

前記第 1 のオペランド及び前記第 2 のオペランドの両方は、汎用レジスタである、請求項 13 に記載のシステム。

【請求項 15】

前記第 1 のオペランドは、マスクレジスタであり、前記第 2 のオペランドは、汎用レジスタである、請求項 13 に記載のシステム。

【請求項 16】

1 つまたは複数の状態レジスタは、前記結果に基づいて設定される、請求項 13 から 15 のいずれか 1 項に記載のシステム。

【請求項 17】

ランダムアクセスメモリと、

50

前記ランダムアクセスメモリに連結されるプロセッサとを備え、前記プロセッサは、複数のベクトルレジスタと、前記複数のベクトルレジスタに連結される実行回路とを備え、前記複数のベクトルレジスタの1つは、アレイの複数のデータエレメントを格納し、前記実行回路は、

少なくとも第1のオペランド及び第2のオペランドを指定するマスク生成命令を受信し

、前記マスク生成命令に応答して、前記第2のオペランドにおいて規定された回数だけ、前記第1のオペランドの複数のビットを左シフトし、前記第1のオペランドの最上位ビットがシフトアウトするたびに、1である最下位ビットをプルインすることにより、複数のビットを含む結果を生成し、前記結果の各ビットは、前記アレイの前記複数のデータエレメントの1つに対応し、

前記複数のベクトルレジスタは、第1のベクトルレジスタ及び第2のベクトルレジスタを含み、前記第2のオペランドは、ベクトル計算に対して、前記第1のベクトルレジスタ内の既存の複数のデータエレメントにマージされるべき、前記第2のベクトルレジスタ内のデータエレメントの数を指定する、システム。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、プロセッサまたは他の処理ロジックによって実行された場合に、論理的、数学的または他の複数の関数オペレーションを実行する処理ロジック、複数のマイクロプロセッサ及び関連づけられた命令セットアーキテクチャの分野に関する。

【背景技術】

【0002】

命令セット、すなわち命令セットアーキテクチャ (ISA) 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、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、複数のアドレッシングモード、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出力 (I/O) を含んでもよい。命令という用語は、概して、本明細書では、プロセッサのデコーダが複数のマクロ命令を復号した結果である複数のマイクロ命令または複数のマイクロオペレーション (micro-op) と対照的に、実行のためにプロセッサ (または (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る) 変換、モーフィング、エミュレートまたは他の方法で命令をプロセッサによって処理されるべき1つまたは複数の他の命令に変換する命令変換部) に付与された複数の命令である複数のマクロ命令をいう。

【0003】

ISA は、命令セットを実装するプロセッサの内部設計であるマイクロアーキテクチャと区別される。複数の異なるマイクロアーキテクチャを有する複数のプロセッサは、共通の命令セットを共有することができる。例えば、複数のインテル (登録商標) コア (商標) プロセッサ及びカリフォルニア州サンバレーのアドバンスト・マイクロ・デバイゼズ社の複数のプロセッサは、x86命令セットの複数のほぼ同一バージョン (複数のより新しいバージョンに追加されたいくつかの拡張を有する) を実装するが、複数の内部設計は異なる。例えば、ISA の同じレジスタアーキテクチャは、複数の専用物理レジスタ、レジスタリネーミングメカニズム等を用いて動的に割り当てられた1つまたは複数の物理レジスタを含む複数の周知技術を用いて、複数の異なるマイクロアーキテクチャにおける複数の異なる態様で実装されてもよい。

【0004】

近年の多くのISA は、パックデータオペレーションまたは単一命令複数データ (SIMD) オペレーションとも称される複数のベクトルオペレーションをサポートする。1つだけのデータエレメントまたは複数のデータエレメントのペアに対して動作するスカラー命令の代わりに、ベクトル命令 (パックデータ命令またはSIMD命令とも称される) は、複数のデータエレメントまたは複数のデータエレメントの複数のペアに対して、同時にま

10

20

30

40

50

たは並列に動作してもよい。プロセッサは、ベクトル命令に応答して複数のオペレーションを同時にまたは並列に実行する並列の実行ハードウェアを有してもよい。

【0005】

ベクトルオペレーションは、1つのオペレーションにおいて、1つのレジスタまたはメモリ位置内でパックされた複数のデータエレメントに対して動作する。これらの複数のデータエレメントは、ベクトルデータエレメントまたは複数のパックデータエレメントと称される。複数のベクトルデータエレメントのそれぞれは、他とは別個独立にオペレーションされ得る個別のデータ（例えば、ピクセルの色等）を表してもよい。

【図面の簡単な説明】

【0006】

複数の実施形態が、添付図面の複数の図において、限定的ではなく、例として示される。

【図1】一実施形態に係る複数のベクトルレジスタ及び複数のマスクレジスタを含む命令処理装置のブロック図である。

【図2A】一実施形態に係る複数のマスク生成命令の例を示す。

【図2B】一実施形態に係る複数のマスク生成命令の例を示す。

【図2C】一実施形態に係る複数のマスク生成命令の例を示す。

【図3A】一実施形態に係るアレイデータアラインメントの例を示す。

【図3B】一実施形態に係るアレイデータアラインメントの例を示す。

【図3C】一実施形態に係るマスクを用いるマスクされたベクトル命令の例を示す。

【図4】一実施形態に係る所定のベクトルレジスタ幅及びデータエレメント幅に対するマスクビットの数を示す。

【図5】一実施形態に係るマスク生成命令に応答して実行されるべき複数のオペレーションを示すフロー図である。

【図6】一実施形態に係るソース命令セットにおける複数のバイナリ命令をターゲット命令セットにおける複数のバイナリ命令に変換するソフトウェア命令変換部の使用を示すブロック図である。

【図7A】一実施形態に係るインオーダ及びアウトオブオーダパイプラインのブロック図である。

【図7B】一実施形態に係るインオーダ及びアウトオブオーダコアのブロック図である。

【図8A】一実施形態に係るより詳細な例示的インオーダコアアーキテクチャのブロック図である。

【図8B】一実施形態に係るより詳細な例示的インオーダコアアーキテクチャのブロック図である。

【図9】一実施形態に係るプロセッサのブロック図である。

【図10】一実施形態に係るシステムのブロック図である。

【図11】一実施形態に係る第2のシステムのブロック図である。

【図12】本発明の実施形態に係る第3のシステムのブロック図である。

【図13】一実施形態に係るシステムオンチップ（SoC）のブロック図である。

【発明を実施するための形態】

【0007】

以下の説明では、多数の具体的な詳細が記載される。しかしながら、本発明の複数の実施形態は、これらの具体的な複数の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。

【0008】

本明細書に記載される複数の実施形態は、複数のマスクされたベクトル命令によって用いられるマスクを生成するプロセッサを生じさせまたはもたらしように動作可能な複数のマスク生成命令を提供する。複数のマスクされたベクトル命令は、計算ループのトリップカウント（すなわち、イタレーション回数）がベクトルレジスタに収容可能な複数のエレ

10

20

30

40

50

メントの数によって割り切れないシナリオに適用可能である。したがって、複数の残りのイタレーションは、別個に処理される必要がある。複数の残りのイタレーションにおける複数のエレメントを処理するために、マスク生成命令は、例外（例えば、割り当てられたメモリの陰でのアクセスまたは / 及び定義されていない複数の結果によって生じる複数の例外）が生成されないように、ベクトルレジスタの一部（例えば、最も重要な複数のエレメント）を複数の計算から省略またはマスクする適切な叙述マスクを生成する。

【 0 0 0 9 】

マスク生成命令は、他の複数のシナリオで用いられることもできる。例えば、命令は、複数の疎ベクトル計算のためのデータ蓄積において、制御マスクを更新するために用いられることができる。データ蓄積は、複数のイタレーションにわたって実行されてもよい。複数のイタレーションのいくつかでは、いくつかのデータエレメントは、計算を終了してもよく、いくつかの新たなデータエレメントが計算に加わってもよい。制御マスクは、さらなる計算を必要とする複数のエレメントを記録するために、更新される。制御マスクは、ベクトル計算の効率性を改善するために、複数のマスクベクトル命令で用いられることができる。

10

【 0 0 1 0 】

複数のベクトル命令と同様に、マスクされたベクトル命令は、1つまたは複数のベクトルオペランドの複数のデータエレメントに対してベクトルオペレーションを実行するプロセッサを生じさせまたはもたらすように、動作可能である。さらに、マスクされたベクトル命令のそれぞれは、ベクトルオペレーションをマスクする、叙述する、または条件に応じて制御するために、マスクを用いる。複数のマスクは、ベクトル処理をデータエレメント粒度あたりでマスクする、または条件に応じて制御するように、動作可能である。例えば、複数のマスクは、単一のソースベクトルオペランドからの個々のデータエレメントまたは2つのソースベクトルオペランドからの対応する複数のデータエレメントの個々のペアに対して実行されたベクトルオペレーションの結果がデスティネーションに格納されようとされなかりと、マスクするように動作可能であってもよい。複数のマスクされたベクトル命令によれば、複数のデータエレメントとは別個独立に、叙述されまたは条件に応じて制御されるべき各データエレメントまたは対応する複数のデータエレメントのペアのベクトル処理が可能となる。複数のマスクされたベクトル命令、複数のオペレーション及び複数のマスクは、例えば、高められたコード密度及び / またはより高い命令スループットのよう特定の複数の利点を提供してもよい。

20

30

【 0 0 1 1 】

図1は、本明細書に記載される複数のマスク生成命令を含む複数の命令を実行するように動作可能な回路を含む実行ユニット140を有する命令処理装置115の実施形態のブロック図である。いくつかの実施形態では、命令処理装置115は、プロセッサ、マルチコアプロセッサのプロセッサコアまたは電子システムの処理エレメントであってもよい。

【 0 0 1 2 】

デコーダ130は、複数の高水準機械命令または複数のマクロ命令の形で入力された複数の命令を受信し、低水準複数のマイクロオペレーション、複数のマイクロコードエントリポイント、複数のマイクロ命令、または元の高水準命令を反映し、及び / またはこれから得られる他の複数の低水準命令もしくは複数の制御信号を生成するために、これらを復号する。複数の低水準命令または複数の制御信号は、複数の低水準（例えば、回路レベルまたはハードウェアレベルの）オペレーションを介して、高水準命令のオペレーションを実装してもよい。デコーダ130は、様々な複数の異なるメカニズムを用いて実装されてもよい。適切な複数のメカニズムの複数の例は、限定されるものではないが、マイクロコード、複数のルックアップテーブル、複数のハードウェア実装、複数のプログラマブルロジックアレイ（PLA）、当技術分野で公知の複数のデコーダを実装するために用いられる他の複数のメカニズム等を含む。

40

【 0 0 1 3 】

デコーダ130は、キャッシュ110、メモリ120または他の複数のソースに対して

50

入力された複数の命令を受信してもよい。復号された複数の命令は、実行ユニット140に送信される。実行ユニット140は、1つまたは複数のマイクロオペレーション、複数のマイクロコードエントリポイント、複数のマイクロ命令、他の複数の命令、または受信された複数の命令を反映し、もしくはこれらから得られた他の複数の制御信号を、デコーダ130から受信してもよい。実行ユニット140は、レジスタファイル170、キャッシュ110及び/またはメモリ120からのデータ入力を受信し、これらへのデータ出力を生成する。

【0014】

一実施形態では、レジスタファイル170は、レジスタとも称される複数のアーキテクチャレジスタを含む。他に指定され、または明らかではない限り、アーキテクチャレジスタ、レジスタファイル及びレジスタという用語は、本明細書では、ソフトウェア及び/またはプログラマに可視な(例えば、ソフトウェアビジブルな)レジスタ、及び/またはオペランドを特定するためにマイクロ命令によって指定されるレジスタを指すために用いられる。これらの複数のレジスタは、所定のマイクロアーキテクチャ(例えば、複数のテンポラリレジスタ、複数のリオーダバッファ、複数のリタイアメントレジスタ等)における他の複数の非アーキテクチャレジスタと対比される。

10

【0015】

代わりに、デコーダ130を有するのではなく、1つまたは複数の他の実施形態では、命令処理装置115は、命令エミュレータ、トランスレータ、モーフア、インタプリタまたは他の命令変換ロジックを代わりに有してもよい。様々な複数の異なるタイプの命令変換ロジックが、当技術分野で公知であり、ソフトウェア、ハードウェア、ファームウェアまたはこれらの組み合わせで実装されてもよい。命令変換ロジックは、複数のマスク生成命令のうちの一つまたは複数を受信し、これを一つまたは複数の対応する導出された複数の命令または複数の制御信号にエミュレート、変換、モーフィング、インタプリタまたは他の方法で変換してもよい。さらに複数の他の実施形態では、命令処理装置115は、デコーダ及び追加の命令変換ロジックの両方を有してもよい。例えば、命令処理装置115は、複数のマスク生成命令のうちの一つまたは複数を変換する命令変換ロジック、及び一つまたは複数の中間命令を、命令処理装置のネイティブのハードウェアによって実行可能な一つまたは複数の低水準命令または複数の制御信号に復号するデコーダを有してもよい。命令変換ロジックのいくつかまたは全ては、個別のダイ上またはオフダイメモリ内のような、命令処理装置の残りからオフダイに位置してもよい。

20

30

【0016】

一実施形態によれば、レジスタファイル170は、複数のベクトルレジスタ175のセット及び複数のマスクレジスタ185のセットを含み、これらの両方は、複数のマスク生成命令の複数のオペランドを格納するために用いられることができる。各ベクトルレジスタ175は、512ビット、256ビットまたは128ビット幅であってもよく、異なるベクトル幅が用いられてもよい。各マスクレジスタ185は、各マスクビットが複数のベクトルレジスタ175の1のデータエレメントの1つに対応する多数のマスクビットを含む。各マスクビットは、ベクトルレジスタのデータエレメントをマスクするために用いられるため、64ビットのマスクレジスタは、512ビットレジスタの64個の8ビットデータエレメントをマスクするために用いられることができる。異なる幅(例えば、256ビットまたは128ビット)を有するベクトルレジスタ及び異なるサイズの複数のデータエレメント(例えば、16ビット、32ビットまたは64ビット)に対して、異なる数の複数のマスクビットがベクトルオペレーションに関連して用いられてもよい。

40

【0017】

説明を分かりにくくしないように、比較的簡単な命令処理装置115が示され、記載されている。複数の他の実施形態は、1つより多くの実行ユニットを有してもよいことを理解されたい。例えば、装置115は、例えば、複数の算術ユニット、複数の算術ロジックユニット(ALU)、複数の整数ユニット、複数の浮動小数点ユニット等のような複数の異なるタイプの実行ユニットを含んでもよい。命令処理装置または複数のプロセッサのさ

50

らに複数の他の実施形態は、複数のコア、複数の論理プロセッサまたは複数の実行エンジンを有してもよい。命令処理装置 115 の多数の実施形態が、図 7A - 13 に関して提供される。

【0018】

本発明の複数の実施形態によれば、本明細書に記載されるマスク生成命令は、命令のレジスタオペランド内の複数のビットをシフトすることによって、マスクを生成する。レジスタオペランドは、マスクレジスタまたは汎用レジスタであってもよい。図 2A - 2C は、複数のマスク生成命令のための疑似コードの複数の例を示す。これらの複数の図では、 r_1 、 r_2 は、独立したサイズの複数の汎用レジスタ（例えば、 r_1 は 32 ビット、 r_2 は 64 ビットであってもよい）を表し、 k_1 は、マスクレジスタを表す。値 KL は、命令の末尾に付されたニーモニック $B/W/D/Q$ から判断可能なマスクビットの数を表す。

10

【0019】

図 2A は、マスク生成命令 $KSHLONES [B/W/D/Q] k_1, r_2$ の例を示す。ニーモニック $B/W/D/Q$ は、命令 $KSHLONES$ が、8、16、32、64 ビットのマスクにそれぞれ対応する $KSHLONESB$ 、 $KSHLONESW$ 、 $KSHLONESD$ 及び $KSHLONESQ$ という 4 つの形式を有することを意味する。

【0020】

この例では、 k_1 マスクは、ソースオペランド及びデスティネーションの両方として機能する。他のソースオペランドは、汎用レジスタまたはメモリからの値である。

【0021】

$KSHLONES$ 命令は、 k_1 マスクの複数のビットをソースオペランド（ r_2 またはメモリ）において規定された回数だけ左にシフトし、複数の下位ビット位置を埋めるために複数の 1 をプルインする。「左にシフト」または「左シフト」という用語は、本明細書では、ビットが最下位ビット（LSB）から最上位ビット（MSB）の方向にシフトされることを意味する。すなわち、 k_1 マスクが 1 つのビット位置だけ左にシフトされるたびに、最下位ビット位置を埋めるために、ビット値 1 がプルインされる。例えば、 $k_1 = 1 : 0 : 0 : 0 : 1 : 1 : 0 : 0$ かつ $r_2 = 4$ の場合、「 $KSHLONESB k_1, r_2$ 」は、 $k_1 = 1 : 1 : 0 : 0 : 1 : 1 : 1 : 1$ という結果を生じさせ、ここで、各「0」及び「1」は、ビット値を表す。結果的な（デスティネーション） k_1 に残ったこれらの k_1 ビットは、単に位置をシフトされただけであり、これらの値はシフトによって変更されていないことに留意されたい。複数の LSB 位置に追加する新たなビットは、全て 1 である。

20

30

【0022】

図 2B は、汎用レジスタ r_1 をソースオペランド及びデスティネーションの両方として用いるマスク生成命令 $KSHLONES [B/W/D/Q] r_1, r_2$ の代替的な実施形態を示す。

【0023】

命令のこの形式により、補完的なビット操作命令としての用途が可能になる。図 2C は、命令が制御フローに直接用いられることができるように複数の状態フラグ（ZF, CF）を修正するマスク生成命令の別の代替的な実施形態を示す。マスク生成命令の他の実施形態は、シフトされた結果（すなわち、結果的なマスク）を、複数のソースオペランド、例えば、 $KSHLONES k_1, k_2, r_2$ 及び $KSHLONES r_1, r_2, r_3$ とは異なるデスティネーションレジスタに格納する。複数の命令の追加的な複数の代替的な実施形態は、上述された複数のマスク生成命令と同じ複数の命令フォーマットを必ずしも有さない。以下の説明では、複数のマスク生成命令の様々な複数の形式は、 $KSHLONES$ 及びその複数の変形と称される。

40

【0024】

図 3A 及び 3B は、 $KSHLONES$ 及びその複数の変形が、ベクトル計算の効率性を改善するために用いられ得る複数のシナリオの例を示す。これらの複数の例では、ベクトルオペレーションの残りのループにおける複数の残りのアレイメントは、ベクトルレ

50

レジスタ全体を埋めない。これらの複数のエレメントでは、ベクトルレジスタは、最大で16のアレイエレメントを格納することができ、例えば、ベクトルレジスタは512ビットを有し、各アレイエレメントは32ビットのダブルワードであると仮定される。複数のアレイエレメントの総数が35で、ループの最初がベクトルレジスタと整合される場合（図3Aに示されるように）、ベクトル化されたループにおいて処理されておらず、別個に処理されることが必要な残りのアレイエレメントが最後に3つ存在する。複数のアレイエレメントの総数が35で、ループの最初がベクトルレジスタと整合されない場合（図3Bに示されるように、第1にベクトル化されたループにおける2のアレイエレメント）、ベクトル化されたループにおいて処理されておらず、別個に処理されることが必要な残りのアレイエレメントが最後に1つ存在する。本明細書に記載されるマスク生成命令は、ループのベクトル化を改善するために、複数のマスクベクトルオペレーションにおいて、複数の残りのアレイエレメントとともに用いられることができるマスクを生成する。

10

20

30

40

50

【0025】

データアクセスの効率性を改善するために、コンパイラは、最後にベクトル化されたループにおいて個別に複数の残りのアレイエレメントを処理するコードを生成することができる。しかしながら、最後にベクトル化されたループにおけるアレイエレメントの数は、概して、複数のアレイエレメントのアドレス及び/またはループトリップカウントがその時点で不明なため、コンパイル時には解消されることができない。本明細書に記載される複数の実施形態により、コンパイル時に、コンパイラは、同じ複数のタスクを実行する他の複数のコードシーケンスの代わりに、複数のマスク生成命令のうちの1つまたは複数を作成することができる。したがって、コンパイラは、ループ最適化の当該タスクを単純化するために、これらの複数のマスク生成命令を用いることができる。複数の代替的な実施形態では、複数のマスク生成命令は、プログラマまたは他のコード生成エンティティによって用いられることができる。

【0026】

KSHLONES命令及びその複数の変形は、ループ末尾における複数の残りのデータエレメントの合計サイズが、ベクトルレジスタの幅より小さいというシナリオを処理するために用いられることができる。これは、全幅のベクトルオペレーションをなすためにループ内の複数のイタレーションが十分でない（すなわち、アレイ内の複数のデータエレメントが十分でない）場合に、KSHLONES命令及びその複数の変形が、用いられることができることを意味する。

【0027】

図3Cの例では、アレイの最後の3のデータエレメント（すなわち、A(32)、A(33)、A(34)）は、ソースベクトル307の全幅を占めない。すなわち、Aに残された複数のエレメントは、ベクトルレジスタ全体を埋めるために十分でない。ソースベクトル307は、その複数の最下位データエレメントとしてA(32)、A(33)、A(34)を含むため、A(32)、A(33)、A(34)に対して加算が実行されるべきであり、加算の複数の結果が格納されるべきであることを示すために、マスク308の最下位3ビットのみが（例えば、1に）設定される。マスク308の上位13ビットは、クリアされる（例えば、0）。マスク308は、KSHLONES命令または当該複数の変形の1つを実行するプロセッサによって生成された結果となり得る。

【0028】

一実施形態では、アレイの最後における複数のデータエレメントの欠如（ベクトルレジスタ全体を埋めるために）は、アレイのベースアドレスにおける初期の不整合という結果になり得る。例えば、複数の画像処理アプリケーションでは、多くの場合、画像アレイのサイズは、ベクトルレジスタ幅の整数倍である。しかしながら、画像アレイの最初が整合していない場合、ベクトルレジスタ全体を埋められない多数のデータエレメントがループの最後において残され得る。

【0029】

マスク308を用いるは、アレイの複数のデータエレメントが複数のオペランドである

ループの実行をベクトル化する助けとなる。図 3 C の複数の例では、イタレーションインデックス $i = 32, 33$ 及び 34 は、ソースベクトル 307 がマスク 308 とともに用いられるマスクされたベクトルオペレーションによりベクトル化されることができ、一実施形態では、ループ検出時に、コンパイラは、本明細書に記載される複数のマスク生成命令のうちの一つまたは複数を含むループ最適化コードを生成することができる。

【0030】

例示されたマスクされたベクトルオペレーション 303 に対する命令は、スカラー値に加算されるべきソースベクトルを示す。他の複数のマスクされたベクトル命令は、2つまたはそれより多くのソースベクトルを示してもよい。マスクされたベクトルオペレーション 303 の命令は、マスク 308 をさらに指定する。複数のマスクのそれぞれは、複数のマスクエレメント、叙述エレメント、条件付き制御エレメント、またはフラグを含む。図に示されるように、オペレーションが1つのソースベクトルオペランドを伴う場合、対応するソースデータエレメントの各々に対し、1つのそのようなマスクエレメントまたはフラグがあってもよい。一般に、各エレメントまたはフラグは、単一のビットであってもよい。単一のビットによれば、2つの異なる可能性（例えば、オペレーションを実行する対オペレーションを実行しない、オペレーションの結果を格納する対オペレーションの結果を格納しない等）のいずれかを指定してもよい。

10

【0031】

代わりに、2つより多くの異なるオプションから選択することが望まれる場合、2つまたはそれより多くのビットが、各フラグまたはエレメントのために用いられてもよい。

20

【0032】

例示された従来技術によれば、所定のマスクビットが1に設定される場合、ベクトルオペレーションの結果は、ソースベクトルの対応するデータエレメントに対して実行され、結果の対応するデータエレメントに格納される。逆に、所定のマスクビットがゼロにクリアされる場合、ベクトルオペレーションは、ソースベクトルの対応するデータエレメントについて省略される（すなわち、実行されない）、または結果が、結果の対応するデータエレメントに格納されることが許容されないのいずれかである。むしろ、別の値は、結果データエレメントに格納されてもよい。例えば、ソースベクトルからの対応するデータエレメントの数値が、格納される。代替的な実施形態では、ゼロまたは別の予め定められた値は、結果の対応するデータエレメントに格納されてもよい。図示されたものと逆の、複数の結果が格納されるようにビットがクリアされる（すなわち、0）、または複数の結果が格納されないように設定される（すなわち、1）従来技術も、可能である。

30

【0033】

以下のコードシーケンスの例は、現在のイタレーションカウントは rbx に、ループ制限は rcx に格納される、残りのループのためのマスクを生成する。図 3 C の例示された実施形態を用いると、現在のイタレーションカウントは 31 、ループ制限は 34 である。

```

SUB    rbx, rcx // calculate number of remaining iterations
KXOR  k1, k1, k1 // zeroing mask
KSHLONES k1, rbx // generate mask for remainder loop

```

40

【0034】

KSHLONES 命令（その複数の変形を含む）を用いることには、残りのループのためにマスクを生成するために多数の利点がある。複数の KSHLONES 命令は、減算結果により動作する。当該オペレーションの一部として減算を含む別の命令に対して、減算前にオペランド型の比較を実行するために、追加の計算前オーバーヘッドが生じる。さらに、複数の KSHLONES 命令は、イタレーションカウンタ及び/またはループ制限が負となり得る複数のシナリオをカバーすることにより、コンパイラにさらなる変動性を与え、コードを最適化することが可能となる。さらに、残りのループのためにマスクを生成するためのコードは、3つのフェーズ（すなわち、上述したコードシーケンスにおける3の

50

命令)に分割されることにより、KSHLONES命令を用いる際に、実行スケジューリングを改善し、さらなる変動性及び柔軟性を提供する。複数のKSHLONES命令は、それ自体または複数のオペランドの減算が必要ない他の複数の命令との組み合わせによって用いられることができる。例えば、1の個数(N)がわかる場合、KSHLONESは、以下のように最下位Nビットに複数の1を有するマスクを生成するために用いられることができる。N = 5、k1 = 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0の場合、KSHLONES k1、Nはk1 = 0 : 0 : 0 : 1 : 1 : 1 : 1 : 1となる。

【0035】

複数のKSHLONES命令は、図4の例に示されるように、疎ベクトル計算のためのデータ蓄積でさらに用いられることができる。この例では、ベクトルレジスタのペア(V1及びV2)及びマスクレジスタのペア(K1及びK2)は、データ蓄積を実行するために用いられる。V1及びV2の両方は、全てのデータエレメント位置が埋められてはいない疎ベクトルである。V1は、計算のために複数のベクトルエレメントを蓄積するアキュムレータとして機能し、V2は、V1の複数の未使用スロットを埋めるために、複数の新たなデータエレメントを提供する。マスクレジスタK1及びK2は、対応する複数のベクトルレジスタ内で計算のために有効な複数のデータエレメントを含む複数の位置を示すために用いられる。この例では、有効な複数のデータエレメントに対応する複数のマスクビットが、K1及びK2の両方に対して1に設定される。K2の複数のビット値は、V2の同じ複数のデータエレメントに対して、反転されることができることが理解されよう。

10

【0036】

図4の例では、V2は、最初、B0として示される4つのエレメントを含む。K2の対応する複数のマスクビットは、これら4つのエレメントの複数の位置を示す。N = POPCNT(K2)を用いることにより、Nの値は、1の値を有するK2ビットの数に設定される。したがって、この例では、N = 4である。K1の複数のマスクビットは、初期のV1のエレメント位置0 - 2に対応する3つの1を含む。K1に含まれる情報は、蓄積されたエレメントA0の数のみならず、V1内における空の複数のスロットの右側境界(この例では、右側境界は、第3のエレメント位置にある)を示す。K1は、そのまま用いられることができ、または、COMPRESS及び/またはEXPAND命令を含むさらなるデータ蓄積のために反転されることができ。

20

【0037】

4つのB0のエレメントは、既存の複数のベクトル命令を用いて、V1のエレメント位置3 - 6に圧縮及びマージされることができ。更新されたV1は、初期のV1よりも高密度となり、故に、効率的なベクトル計算により適したものとなる。マージ後の対応するK1は、K1のソース値における初期の3ビットの1を保存し、さらなる4つのビットの1を加算するK1 = KSHLONES(K1, N)によって計算されることができ。K1のソース値を保存することにより、マージ前後におけるアキュムレータ内のエレメントの数を記録するために個別の複数のカウンタを維持する必要がなくなる。更新されたV1がベクトル計算で用いられた後、図4の複数のオペレーションは、アキュムレータがベクトル計算のために複数のデータエレメントを築盛し続けることができるように、繰り返されることができ。

30

40

【0038】

本明細書に開示される複数のマスク生成命令は、一般的な用途を有する複数の汎用的な命令である。例えば、これらの複数の命令は、複数のベクトルオペレーションの残りのループのため、または疎ベクトル計算におけるデータ蓄積のためのマスクを計算するために、単独で、または他の複数の命令との組み合わせのいずれかで、用いられてもよい。他の用途も、本開示に基づいて企図されてもよい。

【0039】

図5は、一実施形態に係るマスク生成命令を実行するための方法500のブロックフロー図である。方法500は、プロセッサ(より詳細には、例えば、図1の実行ユニット140)が、少なくとも第1のオペランド及び第2のオペランドを指定するマスク生成命令

50

を受信することにより開始する（ブロック510）。複数のマスク生成命令の複数の例は、上述のように、KSHLONES命令及びその複数の変形を含む。一実施形態では、第1のオペランドは、マスクレジスタであり、第2のオペランドは、汎用レジスタである。代替的な実施形態では、第1のオペランド及び第2のオペランドは、両方とも汎用レジスタである。マスク生成命令に回答して、プロセッサは、以下の複数のオペレーションを実行する（ブロック520）。第2のオペランドにおいて規定された回数だけ、第1のオペランドの複数のビットを左シフトし（ブロック530）、第1のオペランドの最上位ビットがシフトアウト（左へ）するたびに1である最下位ビットをプルインすることにより、結果を生成する（ブロック540）。結果の各ビットは、データエレメントに対応する。結果は、マスクされたベクトルオペレーションで用いられるべきマスクである。

10

【0040】

複数の様々な実施形態において、方法500は、汎用プロセッサ、特別用途プロセッサ（例えば、グラフィクスプロセッサもしくはデジタルシグナルプロセッサ）、または別の型のデジタルロジックデバイスまたは命令処理装置によって実行されてもよい。いくつかの実施形態では、方法500は、図1の命令処理装置115、または図7A-13に示される複数の実施形態のような同様のプロセッサ、装置もしくはシステムによって実行されてもよい。さらに、図1の命令処理装置115は、図7A-13に示されるプロセッサ、装置またはシステムと同様に、方法500と同じ、同様の、または異なるもののいずれかである、複数のオペレーション及び複数の方法の複数の実施形態を実行してもよい。

20

【0041】

いくつかの実施形態では、図1の命令処理装置115は、命令をソース命令セットからターゲット命令セットに変換する命令変換部と連携して動作してもよい。例えば、命令変換部は、コアによって処理されるべき1つまたは複数の他の複数の命令に、命令を（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る）変換、モーフィング、エミュレート、または他の方法で変換してもよい。命令変換部は、ソフトウェア、ハードウェア、ファームウェアまたはこれらの組み合わせで実装されてもよい。命令変換部は、プロセッサ上にあってもよく、プロセッサ外にあってもよく、または一部がプロセッサ上かつ一部がプロセッサ外にあってもよい。

【0042】

図6は、本発明の複数の実施形態に係るソフトウェア命令変換部の使用を対比したブロック図である。例示された実施形態では、命令変換部は、ソフトウェア命令変換部であるが、代わりに、命令変換部は、ソフトウェア、ファームウェア、ハードウェアまたは様々なこれらの組み合わせで実装されてもよい。図6は、少なくとも1つのx86命令セットコア616を有するプロセッサによってネイティブで実行され得るx86バイナリコード606を生成するために、x86コンパイラ604を用いてコンパイルされ得る高水準言語602のプログラムを示す。少なくとも1つのx86命令セットコア616を有するプロセッサは、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出すべく、（1）インテル社製x86命令セットコアの命令セットの大部分、または（2）複数のアプリケーションのオブジェクトコードバージョン、もしくは少なくとも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上で動作することが想定された他のソフトウェアを互換可能に実行または他の方法で処理することにより、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複数の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604は、さらなるリンク処理の有無に関わらず、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616上で実行可能なx86バイナリコード606（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。

30

40

【0043】

同様に、図6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614（例えば、カリフォルニア州サンバールのMIPSテクノロジーズ（MIPS Technologies）のMIPS命令セットを実行する、及び/またはカリフォルニア州

50

サニーベールのARMホールディングス（ARM Holdings）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され得る、代替的な命令セットバイナリコード610を生成するべく、代替的な命令セットコンパイラ608を用いてコンパイルされ得る高水準言語602のプログラムを示す。命令変換部612は、x86バイナリコード606を、x86命令セットコアを有さないプロセッサ614によってネイティブで実行され得る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、このような変換が可能な命令変換部の製造は難しいため、代替的な命令セットバイナリコード610と同じとなる可能性は低い、しかしながら変換されたコードは、全般的なオペレーションを達成し、代替的な命令セットからの複数の命令により補完される。したがって、命令変換部612は、エミュレーション、シミュレーションまたは任意の他の処理を介して、プロセッサまたはx86命令セットプロセッサもしくはコアを有さない他の電子デバイスにx86バイナリコード606を実行させるソフトウェア、ファームウェア、ハードウェアまたはこれらの組み合わせを表す。

【0044】

[例示的な複数のコアアーキテクチャ]

[インオーダ及びアウトオブオーダコアのブロック図]

図7Aは、本発明の複数の実施形態に係る例示的なインオーダ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7Bは、本発明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7A及び7Bの複数の実線のボックスは、インオーダパイプライン及びインオーダコアを示し、選択的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、アウトオブオーダ発行/実行パイプライン及びコアを示す。インオーダ態様がアウトオブオーダ態様のサブセットであるとして、アウトオブオーダ態様について説明する。

【0045】

図7Aでは、プロセッサパイプライン700は、フェッチステージ702、長さ復号ステージ704、復号ステージ706、配分ステージ708、リネームステージ710、スケジューリング（ディスパッチまたは発行としても知られる）ステージ712、レジスタ読み出し/メモリ読み出しステージ714、実行ステージ716、ライトバック/メモリ書き込みステージ718、例外処理ステージ722及びコミットステージ724を含む。

【0046】

図7Bは、実行エンジンユニット750に連結されるフロントエンドユニット730を含むプロセッサコア790を示し、両方ともメモリユニット770に連結される。コア790は、縮小命令セットコンピュータ（RISC）コア、複合命令セットコンピュータ（CISC）コア、超長命令語（VLIW）コアまたはハイブリッドもしくは代替的なコアタイプであってもよい。さらに他のオプションとして、コア790は、例えば、ネットワークまたは通信コア、圧縮エンジン、コプロセッサコア、汎用演算画像処理ユニット（GPGPU）コア、グラフィクスコアなどのような特別用途コアであってもよい。

【0047】

フロントエンドユニット730は、命令キャッシュユニット734に連結された分岐予測ユニット732を含み、命令キャッシュユニット734は、命令トランスレーションルックアサイドバッファ（TLB）736に連結され、TLB736は、命令フェッチユニット738に連結され、命令フェッチユニット738は、復号ユニット740に連結される。復号ユニット740（またはデコーダ）は、複数の命令を復号し、出力として、1つまた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、または元の複数の命令から復号された、もしくはこれらを他の方法で反映する、もしくはこれらから得られた他の制御信号を生成してもよい。復号ユニット740は、複数の様々な異なるメカニズムを用いて実装されてもよい。適切な複数のメカニズムの例は、限定されるものではないが、複数のルックアップテーブル、複数のハードウェア実

10

20

30

40

50

装、複数のプログラマブルロジックアレイ（PLA）、複数のマイクロコードリードオンリメモリ（ROM）等を含む。一実施形態では、コア790は、マイクロコードROMまたは（例えば、復号ユニット740、さもなければフロントエンドユニット730内で）特定の複数のマクロ命令に対するマイクロコードを格納する他の媒体を含む。復号ユニット740は、実行エンジンユニット750内のリネーム/アロケータユニット752に連結される。

【0048】

実行エンジンユニット750は、リタイアメントユニット754及び1つまたは複数のスケジューラユニット756のセットに連結されたリネーム/アロケータユニット752を含む。スケジューラユニット756は、予約ステーション、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756は、物理レジスタファイルユニット758に連結される。複数の物理レジスタファイルユニット758のそれぞれは、1つまたは複数の物理レジスタファイルを表し、異なる物理レジスタファイルの異なるいくつかは、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、次に実行されるべき命令のアドレスである命令ポインタ）のような、1つまたは複数の異なるデータ型を格納する。一実施形態では、物理レジスタファイルユニット758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。これらの複数のレジスタユニットは、複数のアーキテクチャ上のベクトルレジスタ、ベクトルマスクレジスタ及び汎用レジスタを提供してもよい。物理レジスタファイルユニット758は、リタイアメントユニット754にオーバーラップされることにより、レジスタリネーミング及びアウトオブオーダ実行が実装され得る様々な複数の態様（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、ヒストリ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て、等）を示す。リタイアメントユニット754及び物理レジスタファイルユニット758は、実行クラスタ760に連結される。実行クラスタ760は、1つまたは複数の実行ユニット762のセット及び1つまたは複数のメモリアクセスユニット764のセットを含む。複数の実行ユニット762は、複数の様々なタイプのデータ（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、様々な複数のオペレーション（例えば、複数のシフト、加算、減算、乗算）を実行してもよい。いくつかの実施形態は、具体的な複数の機能また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もよく、複数の他の実施形態は、1つだけの実行ユニットまたは全ての機能を全てが実行する複数の実行ユニットを含んでもよい。スケジューラユニット756、物理レジスタファイルユニット758及び実行クラスタ760は、複数として示される可能性があるが、その理由は、特定の複数の実施形態は、特定の複数のデータ型/複数のオペレーションに対して個別の複数のパイプラインを生成するからである（例えば、各々が自己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/また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/またはメモリアクセスパイプライン、及び、個別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764を有する特定の複数の実施形態が実装される）。個別の複数のパイプラインが用いられる場合、これらのパイプラインのうちの1つまたは複数がアウトオブオーダ発行/実行であり、残りがインオーダであってもよいことも理解されたい。

【0049】

メモリアクセスユニット764のセットは、データTLBユニット772を含むメモリユニット770に連結され、データTLBユニット772は、データキャッシュユニット774に連結され、データキャッシュユニット774は、二次（L2）キャッシュユニット776に連結される。例示的な一実施形態では、メモリアクセスユニット764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もよく、これ

10

20

30

40

50

らのそれぞれは、メモリユニット 770 内のデータ L B ユニット 772 に連結される。命令キャッシュユニット 734 は、メモリユニット 770 内の二次 (L2) キャッシュユニット 776 に、さらに連結される。L2 キャッシュユニット 776 は、1 つまたは複数の他のレベルのキャッシュ、及び最終的にはメインメモリに連結される。

【0050】

例として、例示的なレジスタリネーミング、アウトオブオーダ発行 / 実行コアアーキテクチャは、パイプライン 700 を以下のとおり実装してもよい。1) 命令フェッチ 738 が、フェッチステージ 702 及び長さ復号ステージ 704 を実行し、2) 復号ユニット 740 が、復号ステージ 706 を実行し、3) リネーム / アロケータユニット 752 が、配分ステージ 708 及びリネームステージ 710 を実行し、4) スケジューラユニット 756 が、スケジューリングステージ 712 を実行し、5) 物理レジスタファイルユニット 758 及びメモリユニット 770 が、レジスタ読み出し / メモリ読み出しステージ 714 を実行し、実行クラスタ 760 が、実行ステージ 716 を実行し、6) メモリユニット 770 及び物理レジスタファイルユニット 758 が、ライトバック / メモリ書き込みステージ 718 を実行し、7) 様々な複数のユニットが、例外処理ステージ 722 に関与してもよく、かつ 8) リタイアメントユニット 754 及び物理レジスタファイルユニット 758 が、コミットステージ 724 を実行する。

【0051】

コア 790 は、本明細書に記載される命令を含む 1 つまたは複数の命令セット (例えば、x86 命令セット (複数のより新しいバージョンに追加されたいくつかの拡張を有する)、カリフォルニア州サニーベールの MIPS テクノロジーズの MIPS 命令セット、カリフォルニア州サニーベールの ARM ホールディングスの ARM 命令セット (NEON などの複数の選択的なさらなる拡張を有する)) をサポートしてもよい。一実施形態では、コア 790 は、バックデータ命令セット拡張 (例えば、SSE、AVX1、AVX2 等) をサポートするロジックを含むことによって、多くのマルチメディアアプリケーションによって用いられる複数のオペレーションが、バックデータを用いて実行される。

【0052】

コアは、マルチスレッディング (複数のオペレーションまたは複数のスレッドの 2 つまたはそれより多くの並列セットを実行すること) をサポートしてもよく、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 (単一の物理的コアが複数のスレッドのそれぞれに対して論理的コアを提供することにより、物理的コアが同時マルチスレッディングを実行すること) また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もよいことを理解されたい。

【0053】

レジスタリネーミングを、アウトオブオーダ実行との関連で説明しているが、インオーダーアーキテクチャにおいて、レジスタリネーミングが用いられてもよいことを理解されたい。プロセッサの例示された実施形態は、個別の命令およびデータキャッシュユニット 734 / 774、及び共有の L2 キャッシュユニット 776 をさらに含むが、複数の代替的な実施形態は、複数の命令およびデータの両方のために、例えば、一次 (L1) 内部キャッシュ、または複数のレベルの内部キャッシュのような単一の内部キャッシュを有してもよい。いくつかの実施形態では、システムは、内部キャッシュとコア及び / また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もよい。代わりに、全てのキャッシュは、コア及び / またはプロセッサの外部にあってもよい。

【0054】

[例示的なインオーダーコアアーキテクチャの詳細]

【0055】

図 8A - B は、より詳細な、例示的な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 (同じタイプ及び / または異なる

10

20

30

40

50

るタイプの他の複数のコアを含む)の中の1つであってもよい。複数の論理ブロックは、用途に応じて、高帯域幅相互接続ネットワーク(例えば、リングネットワーク)を介して、いくつかの固定機能ロジック、メモリI/Oインターフェース及び他の必要なI/Oロジックと通信を行う。

【0056】

図8Aは、本発明の複数の実施形態に係る単一のプロセッサコアのブロック図であり、オンダイの相互接続ネットワーク802との接続及び二次(L2)キャッシュローカルサブセット804と共に示される。一実施形態では、命令デコーダ800は、パック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806によれば、キャッシュメモリによる、スカラ及びベクトルユニットへの低レイテンシアクセスが可能である。一実施形態では、(設計の単純化のために)スカラユニット808及びベクトルユニット810は、個別のレジスタセット(それぞれ、複数のスカラレジスタ812及び複数のベクトルレジスタ814)を用い、これらの中で転送されるデータは、一次(L1)キャッシュ806のメモリに書き込まれてから再読み出しされるが、本発明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、また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でデータを転送させる通信パスを含む)を用いてもよい。

10

【0057】

L2キャッシュローカルサブセット804は、1つのプロセッサコアあたり1つの個別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、自己のL2キャッシュローカルサブセット804に対するダイレクトアクセスパスを有する。プロセッサコアに読み出されたデータは、自己のL2キャッシュサブセット804に格納され、迅速かつ、自己の複数のローカルL2キャッシュサブセットにアクセスする他の複数のプロセッサコアと並列に、アクセスされることができる。プロセッサコアに書き込まれたデータは、自己のL2キャッシュサブセット804に格納され、必要な場合には、他の複数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、共有のデー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が双方向であることにより、複数のプロセッサコア、複数のL2キャッシュ及び他の複数の論理ブロックなどのエージェントは、チップ内で互いに通信を行うことができる。各リングデータパスは、1方向あたり1012ビット幅である。

20

30

【0058】

図8Bは、本発明の複数の実施形態に係る図8Aのプロセッサコアの一部の拡大図である。図8Bは、L1キャッシュ806の一部であるL1データキャッシュ806Aと、ベクトルユニット810及び複数のベクトルレジスタ814に関するさらなる詳細とを含む。

【0059】

具体的には、ベクトルユニット810は、整数、単精度浮動及び倍精度浮動命令のうちの1つまたは複数を実行する16幅ベクトル処理ユニット(VPU)(16幅ALU828を参照)である。VPUは、再構成ユニット820による複数のレジスタ入力の再構成、数字変換ユニット822A-Bによる数字変換、及び複製ユニット824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826によれば、結果的な複数のベクトルの書き込みを叙述することが可能となる。

40

【0060】

[集積メモリコントローラ及びグラフィクスを有するプロセッサ]

図9は、本発明の複数の実施形態に係るプロセッサ900のブロック図であり、プロセッサ900は、1つより多くのコアを有してもよく、集積メモリコントローラを有してもよく、集中画像表示を有してもよい。図9の複数の実線のボックスは、単一のコア902A、システムエージェント910、1つまたは複数のバスコントローラユニット916のセットを有するプロセッサ900を示し、選択的に追加された複数の破線のボックスは、複数のコア902A-Nを有する代替的なプロセッサ900、システムエージェントユニ

50

ット 9 1 0 内の 1 つまたは複数の集積メモリコントローラユニット 9 1 4 のセット及び特別用途ロジック 9 0 8 を示す。

【 0 0 6 1 】

したがって、プロセッサ 9 0 0 の異なる複数の実装は、1) 集中画像表示及び/または科学的(スループット)ロジック(1 つまたは複数のコアを含んでもよい)である特別用途ロジック 9 0 8 を有する CPU、及び 1 つまたは複数の汎用コア(例えば、汎用インオーダーコア、汎用アウトオーダーコア、これらの 2 つの組み合わせ)であるコア 9 0 2 A - N、2) 主にグラフィクス及び/または科学的(スループット)用として意図された多数の特別用途コアであるコア 9 0 2 A - 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コア 9 0 2 A - N を有するコプロセッサを含んでもよい。したがって、プロセッサ 9 0 0 は、汎用プロセッサ、コプロセッサまたは、例えば、ネットワークまたは通信プロセッサ、圧縮エンジン、グラフィクスプロセッサ、G P G P U (汎用画像処理ユニット)、ハイスループット多集積コア(M I C)コプロセッサ(3 0 またはそれより多くのコアを含む)、組み込みプロセッサなどのような特別用途プロセッサであってもよい。

10

【 0 0 6 2 】

プロセッサは、1 つまたは複数のチップ上に実装されてもよい。プロセッサ 9 0 0 は、例えば、B i C M O S、C M O S または N M O S などの多数の処理技術のいずれかを用いて、1 つまたは複数の基板の一部であってもよく、及び/またはその上に実装されてもよい。

20

【 0 0 6 3 】

メモリ階層は、複数のコア、1 つまたは複数の共有キャッシュユニット 9 0 6 またはそのセット、及び複数の集積メモリコントローラユニット 9 1 4 のセットに連結される外部メモリ(不図示)内に、1 つまたは複数のレベルのキャッシュを含む。複数の共有キャッシュユニット 9 0 6 のセットは、二次(L 2)、三次(L 3)、四次(L 4)または他の複数のレベルのキャッシュなどの 1 つまたは複数の中レベルキャッシュ、ラストレベルキャッシュ(L L C)及び/またはこれらの組み合わせを含んでもよい。一実施形態では、リングベースの相互接続ユニット 9 1 2 が、集中画像表示ロジック 9 0 8、複数の共有キャッシュユニット 9 0 6 のセット及びシステムエージェントユニット 9 1 0 / 集積メモリコントローラユニット 9 1 4 を相互接続するが、複数の代替的な実施形態は、そのような複数のユニットを相互接続するための任意の数の周知技術を用いてもよい。一実施形態では、1 つまたは複数のキャッシュユニット 9 0 6 と複数のコア 9 0 2 A - N との間で、コヒーレンシが維持される。

30

【 0 0 6 4 】

いくつかの実施形態では、複数のコア 9 0 2 A - N のうちの 1 つまたは複数は、マルチスレディングが可能である。システムエージェント 9 1 0 は、複数のコア 9 0 2 A - N の調整及び操作を行う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 9 1 0 は、例えば、電力制御ユニット(P C U)及びディスプレイユニットを含んでもよい。P C U は、複数のコア 9 0 2 A - N 及び集中画像表示ロジック 9 0 8 の電力状態を調整するために必要なロジック及び複数のコンポーネントであってもよく、またはこれらを含んでもよい。ディスプレイユニットは、1 つまたは複数の外部接続ディスプレイを駆動するためのものである。

40

【 0 0 6 5 】

複数のコア 9 0 2 A - N は、アーキテクチャ命令セットに関して同種または異種であってもよく、すなわち、複数のコア 9 0 2 A - N のうちの 2 つまたはそれより多くは、同じ命令セットを実行可能であってもよいが、他は、その命令セットまた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もよい。

【 0 0 6 6 】

[例示的なコンピュータアーキテクチャ]

図 1 0 - 1 3 は、例示的な複数のコンピュータアーキテクチャのブロック図である。ラ

50

ップトップ、デスクトップ、ハンドヘルド型PC、携帯情報端末、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、デジタルシグナルプロセッサ(DSP)、グラフィクスデバイス、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用の当技術分野で公知の他の複数のシステム設計及び複数の構成も、適切である。概して、本明細書で開示されるように、プロセッサ及び/または他の実行ロジックを組み込み可能な多様なシステムまたは電子デバイスが、概して適切である。

【0067】

ここで、図10を参照すると、本発明の一実施形態に係るシステム1000のブロック図が示される。システム1000は、コントローラハブ1020に連結される1つまたは複数のプロセッサ1010、1015を含んでもよい。一実施形態では、コントローラハブ1020は、(個別のチップ上にあってもよい)グラフィクスメモリコントローラハブ(GMCH)1090及び入出力ハブ(IOH)1050を含み、GMCH1090は、メモリ1040及びコプロセッサ1045が連結されるメモリ及び複数のグラフィクスコントローラを含み、IOH1050は、複数の入出力(I/O)デバイス1060をGMCH1090に連結する。代わりに、メモリ及び複数のグラフィクスコントローラ的一方または両方は、プロセッサ内に集積され(本明細書に記載されるように)、メモリ1040及びコプロセッサ1045は、プロセッサ1010及び単一のチップ内でIOH1050を有するコントローラハブ1020に直接連結される。

10

20

【0068】

複数の追加のプロセッサ1015の選択的な特性が、図10に複数の破線で示される。各プロセッサ1010、1015は、本明細書に記載される複数のプロセッサコアのうちの1つまたは複数を含んでもよく、いくつかのバージョンのプロセッサ900であってもよい。

【0069】

メモリ1040は、例えば、ダイナミックランダムアクセスメモリ(DRAM)、位相変化メモリ(PCM)またはこれら2つの組み合わせであってもよい。少なくとも1つの実施形態について、コントローラハブ1020は、フロントサイドバス(FSB)のようなマルチドロップバス、クイックパスインターコネクト(QPI)のようなポイントツーポイントインターフェースまたは同様の接続1095を介して、プロセッサ1010、1015と通信を行う。

30

【0070】

一実施形態では、コプロセッサ1045は、例えば、ハイスループットMICプロセッサ、ネットワークまたは通信プロセッサ、圧縮エンジン、グラフィクスプロセッサ、GPGPU、組み込みプロセッサなどのような特別用途プロセッサである。一実施形態では、コントローラハブ1020は、集中画像表示アクセラレータを含んでもよい。

【0071】

物理的リソース1010、1015の間には、アーキテクチャ、マイクロアーキテクチャ、温度、電力消費特性等を含む様々な利益の基準に関して、様々な複数の違いが存在しうる。

40

【0072】

一実施形態では、プロセッサ1010は、一般的なタイプ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。複数の命令内に、複数のコプロセッサ命令が組み込まれていてもよい。プロセッサ1010は、これらの複数のコプロセッサ命令を、取り付けられたコプロセッサ1045によって実行されるべきタイプのもものと認識する。従って、プロセッサ1010は、これらの複数のコプロセッサ命令(または複数のコプロセッサ命令を表す複数の制御信号)を、コプロセッサバスまたは他のインターコネクト上で、コプロセッサ1045に対して発行する。コプロセッサ1045は、受信された複数のコプロセッサ命令を受け付けて実行する。

50

【0073】

ここで、図11を参照すると、本発明の実施形態に係る第1のより詳細な例示的なシステム1100のブロック図が示される。図11に示されるように、マルチプロセッサシステム1100は、ポイントツーポイントインターコネクトシステムであり、ポイントツーポイントインターコネクト1150を介して連結される第1のプロセッサ1170及び第2のプロセッサ1180を含む。プロセッサ1170および1180のそれぞれは、いくつかのバージョンのプロセッサ900であってもよい。本発明の一実施形態では、プロセッサ1170および1180は、それぞれプロセッサ1010および1015であり、コプロセッサ1138は、コプロセッサ1045である。他の実施形態では、プロセッサ1170および1180は、それぞれプロセッサ1010、コプロセッサ1045である。

10

【0074】

プロセッサ1170および1180は、集積メモリコントローラ(IMC)ユニット1172および1182をそれぞれ含むものとして示される。プロセッサ1170は、自己のバスコントローラユニットの一部として、複数のポイントツーポイント(P-P)インターフェース1176および1178をさらに含み、同様に、第2のプロセッサ1180は、複数のP-Pインターフェース1186および1188を含む。複数のプロセッサ1170、1180は、複数のP-Pインターフェース回路1178、1188を用いたポイントツーポイント(P-P)インターフェース1150を介して、情報を交換してもよい。図11に示されるように、IMC1172および1182は、複数のプロセッサを個別のメモリ、すなわち、個別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となり得るメモリ1132及びメモリ1134に連結する。

20

【0075】

複数のプロセッサ1170、1180の各々は、複数のポイントツーポイントインターフェース回路1176、1194、1186、1198を用いる個々のP-Pインターフェース1152、1154を介して、チップセット1190と情報を交換してもよい。チップセット1190は、高性能インターフェース1139を介して、コプロセッサ1138と任意選択的に情報を交換してもよい。一実施形態では、コプロセッサ1138は、例えば、ハイスループットMICプロセッサ、ネットワークまたは通信プロセッサ、圧縮エンジン、グラフィクスプロセッサ、GPGPU、組み込みプロセッサなどのような特別用途プロセッサである。

30

【0076】

共有キャッシュ(不図示)は、いずれかのプロセッサの内部に含まれ、または両方のプロセッサの外部にあってもよいが、プロセッサが低電力モードであっても、いずれかまた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に格納され得るように、P-Pインターコネクトを介して複数のプロセッサに接続される。

【0077】

チップセット1190は、インターフェース1196を介して、第1のバス1116に連結されてもよい。一実施形態では、第1のバス1116は、ペリフェラルコンポーネントインターコネクト(PCI)バスまたはPCI Expressバスもしくは他の第3世代I/O相互接続バスなどのバスであってもよいが、本発明の範囲はこれに限定されない。

40

【0078】

図11に示されるように、様々なI/Oデバイス1114が、第1のバス1116を第2のバス1120に連結するバスブリッジ1118と共に、第1のバス1116に連結されてもよい。一実施形態では、コプロセッサ、ハイスループットMICプロセッサ、GPGPUのアクセラレータ(例えば、グラフィクスアクセラレータもしくはデジタル信号処理(DSP)ユニットなど)、フィールドプログラマブルゲートアレイまたは任意の他のプロセッサなどの1つまたは複数の追加のプロセッサ1115が、第1のバス1116に連結される。一実施形態では、第2のバス1120は、ローピンカウント(LPC)バスであってもよい。

50

【 0 0 7 9 】

例えば、キーボード及び/またはマウス 1 1 2 2、通信デバイス 1 1 2 7、及びディスクドライブ、または複数の命令/コード及びデータ 1 1 3 0 を含む他の大容量ストレージデバイスなどのストレージユニット 1 1 2 8 を含む様々な複数のデバイスが、一実施形態では、第 2 のバス 1 1 2 0 に連結されてもよい。さらに、オーディオ I/O 1 1 2 4 が、第 2 のバス 1 1 2 0 に連結されてもよい。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 1 1 のポイントツーポイントアーキテクチャの代わりに、システムは、マルチドロップバスまたは他のそのようなアーキテクチャを実装してもよい。

【 0 0 8 0 】

ここで、図 1 2 を参照すると、本発明の実施形態に係る第 2 のより詳細な、例示的なシステム 1 2 0 0 のブロック図が示される。図 1 1 および 1 2 における同様の複数のエレメントには、同様の参照番号が付され、図 1 1 の複数の特定の態様は、図 1 2 の他の複数の態様の妨げとならないよう、図 1 2 では省略されている。

10

【 0 0 8 1 】

図 1 2 は、複数のプロセッサ 1 1 7 0、1 1 8 0 は、それぞれ集積メモリ及び I/O 制御ロジック(「CL」) 1 1 7 2 及び 1 1 8 2 を含んでもよいことを示す。したがって、CL 1 1 7 2、1 1 8 2 は、複数の集積メモリコントローラユニットを含み、かつ、I/O 制御ロジックを含む。図 1 2 は、メモリ 1 1 3 2、1 1 3 4 のみが CL 1 1 7 2、1 1 8 2 に連結されるのではなく、複数の I/O デバイス 1 2 1 4 も、複数の制御ロジック 1 1 7 2、1 1 8 2 に連結されることを示す。複数のレガシー I/O デバイス 1 2 1 5 は、チップセット 1 1 9 0 に連結される。

20

【 0 0 8 2 】

ここで、図 1 3 を参照すると、本発明の実施形態に係る SoC 1 3 0 0 のブロック図が示される。図 9 における同様の複数のエレメントには、同様の参照番号が付される。また、複数の破線のボックスは、より高度な SoC 上のオプションの機能である。図 1 3 では、相互接続ユニット 1 3 0 2 は、1 つまたは複数のコア 2 0 2 A N 及び共有キャッシュユニット 9 0 6 のセットを含むアプリケーションプロセッサ 1 3 1 0、システムエージェントユニット 9 1 0、バスコントローラユニット 9 1 6、集積メモリコントローラユニット 9 1 4、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 つまたは複数のコプロセッサ 1 3 2 0 またはそのセット、スタティックランダムアクセスメモリ(SRAM)ユニット 1 3 3 0、ダイレクトメモリアccess(DMA)ユニット 1 3 3 2 及び 1 つまたは複数の外部ディスプレイに連結するためのディスプレイユニット 1 3 4 0 に連結される。一実施形態では、コプロセッサ 1 3 2 0 は、例えば、ネットワークまたは通信プロセッサ、圧縮エンジン、GP GPU、ハイスループット MIC プロセッサ、組み込みプロセッサなどのような特別用途プロセッサを含む。

30

【 0 0 8 3 】

本明細書に開示される複数のメカニズムの複数の実施形態は、ハードウェア、ソフトウェア、ファームウェアまたはそ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もよい。本発明の複数の実施形態は、少なくとも 1 つのプロセッサ、ストレージシステム(揮発性及び不揮発性メモリ及び/または複数のストレージ要素を含む)、少なくとも 1 つの入力デバイスおよび少なくとも 1 つの出力デバイスを備えるプログラム可能な複数のシステム上で実行する複数のコンピュータプログラムまたはプログラムコードとして実装されてもよい。

40

【 0 0 8 4 】

図 1 1 に示すコード 1 1 3 0 などのプログラムコードは、本明細書に記載される複数の機能を実行し、出力情報を生成するために、複数の入力命令に適用されてもよい。出力情報は、1 つまたは複数の出力デバイスに、公知の態様で適用されてもよい。この適用のために、処理システムは、例えば、デジタルシグナルプロセッサ(DSP)、マイクロコントローラ、特定用途向け集積回路(AASIC)またはマイクロプロセッサなどのプロセッサを有する任意のシステムを含む。

50

【 0 0 8 5 】

プログラムコードは、処理システムと通信を行うために、高水準の手順型またはオブジェクト指向プログラミング言語で実装されてもよい。プログラムコードは、必要に応じて、アセンブリ言語または機械言語でさらに実装されてもよい。実際には、本明細書に記載される複数のメカニズムは、その範囲において、任意の特定のプログラム言語に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、コンパイラ型またはインタプリタ型言語であってもよい。

【 0 0 8 6 】

少なくとも1つの実施形態のうち1つまたは複数の態様は、機械可読媒体に格納された、プロセッサ内の様々なロジックを表す複数の代表的な命令によって実装されてもよく、このような命令は、機械に読み出された場合に、本明細書に記載される複数の技術を実行するべく、機械にロジックを組み立てさせる。「IPコア」として知られるそのような複数の表現は、有形の機械可読媒体上に格納され、様々な顧客または製造設備に供給されて、実際にロジックまたはプロセッサを作り出す製造機械にロードされてもよい。

10

【 0 0 8 7 】

そのような機械可読記憶媒体は、限定的ではないが、ハードディスク、フロッピー（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）及び磁気光ディスクを含む任意の他のタイプのディスク、リードオンリメモリ（ROM）などの半導体デバイス、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）、消去可能プログラマブルROM（EPROM）、フラッシュメモリ、電気的消去可能プログラマブルROM（EEPROM）、位相変化メモリ（PCM）、磁気または光カードなどのランダムアクセスメモリ（RAM）、または複数の電子的命令を格納するために適切な任意の他のタイプのメディアなどのストレージ媒体を含む、機械またはデバイスによって製造もしくは形成される複数の物品の非一時的かつ有形の構成を含んでもよい。

20

【 0 0 8 8 】

従って、本発明の複数の実施形態は、複数の命令を含む、または本明細書に記載される複数の構造、複数の回路、複数の装置、複数のプロセッサ及び/または複数のシステム機能を定義するハードウェア記述言語（HDL）などの設計データを含む非一時的かつ有形の機械可読媒体をさらに含む。そのような複数の実施形態は、プログラム製品と称されてもよい。

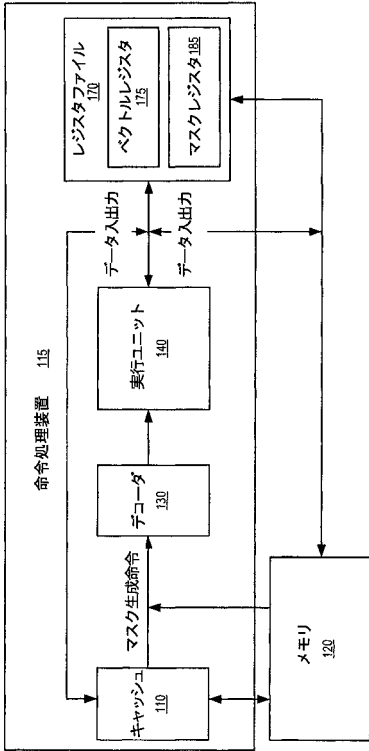
30

【 0 0 8 9 】

特定の例示的な複数の実施形態が、複数の添付図面で説明及び図示されたが、そのような複数の実施形態は、単に例示であり、広範な発明を限定するものではなく、当業者であれば、本開示を参照することにより様々な他の複数の変更にも想到し得ることから、本発明は、図示及び説明された詳細な複数の構造及び構成に限定されるものではないことが理解されよう。成長が速く、さらなる進歩が容易に予見し得ないこのような技術領域においては、本開示の原理または添付された特許請求の範囲から逸脱することなく、技術的進歩を可能とすることによって促進されるように、開示された複数の実施形態は、構成及び詳細において容易に変更され得る。

40

【 図 1 】



【 図 2 A 】

```

KSHLONES[BW/D/Q] k1, r2           ; destination, SRC1
KL ← 8, 16, 32, 64                 ; number of mask bits (based on mnemonic)
N = min(KL, SRC1)
for (i=0; i<N; i++){
    k1 <<= 1+i
}
    
```

【 図 2 B 】

```

SHLONES[BW/D/Q] r1, r2             ; destination, SRC1
KL ← 8, 16, 32, 64                 ; number of mask bits (based on mnemonic)
N = min(KL, SRC1)
for (i=0; i<N; i++){
    r1 <<= 1+i
}
    
```

【 図 2 C 】

```

KSHLONES[BW/D/Q] k1, r2           ; destination, SRC1
KL ← 8, 16, 32, 64                 ; number of mask bits (based on mnemonic)
N = min(KL, SRC1)
for (i=0; i<N; i++){
    k1 <<= 1+i
}
if (N==0) ZF = 1
if (N==KL) CF = 1
    
```

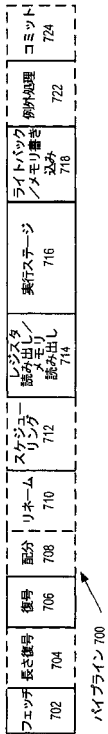
【 図 3 A 】

第1ループ	A(15)	A(14)	A(13)	...	A(3)	A(2)	A(1)	A(0)
第2ループ	A(31)	A(30)	A(29)	...	A(19)	A(18)	A(17)	A(16)
残りのループ	X	X	X	...	X	A(34)	A(33)	A(32)

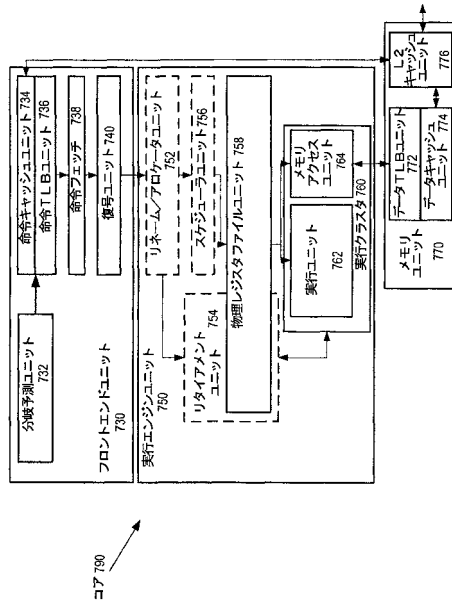
【 図 3 B 】

第1ループ	X	X	X	...	X	X	A(1)	A(0)
第2ループ	A(17)	A(16)	A(15)	...	A(5)	A(4)	A(3)	A(2)
第3ループ	A(33)	A(32)	A(31)	...	A(21)	A(20)	A(19)	A(18)
残りのループ	X	X	X	...	X	X	X	A(34)

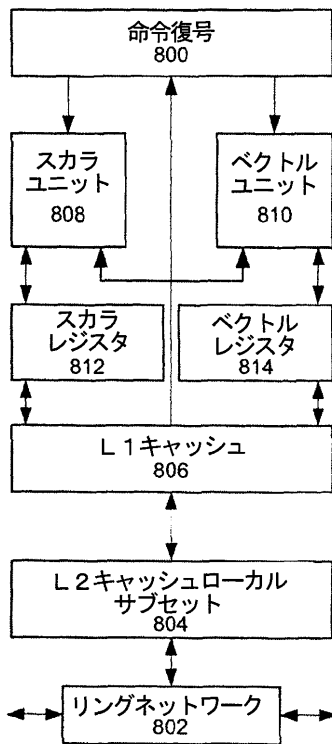
【 図 7 A 】



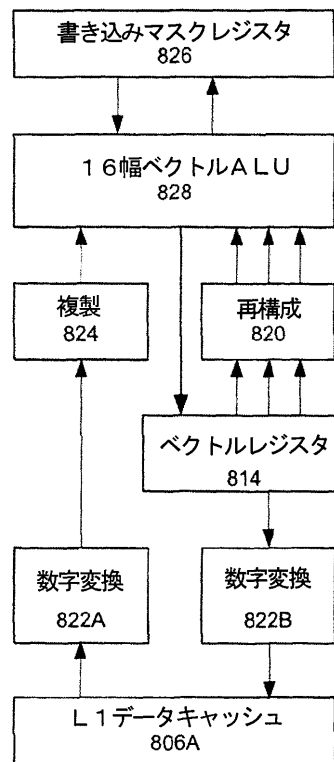
【 図 7 B 】



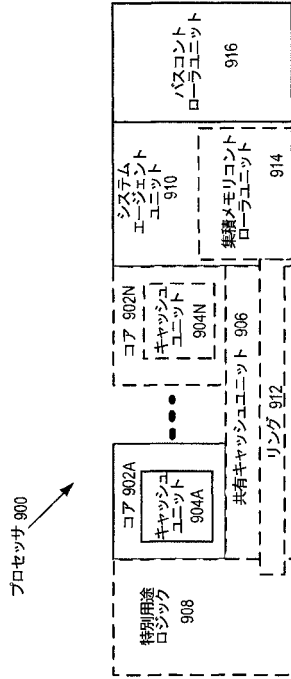
【 図 8 A 】



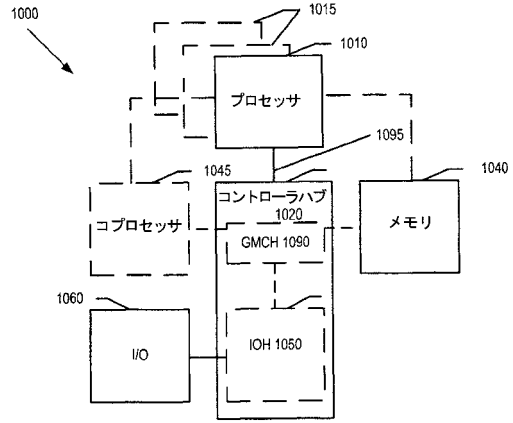
【 図 8 B 】



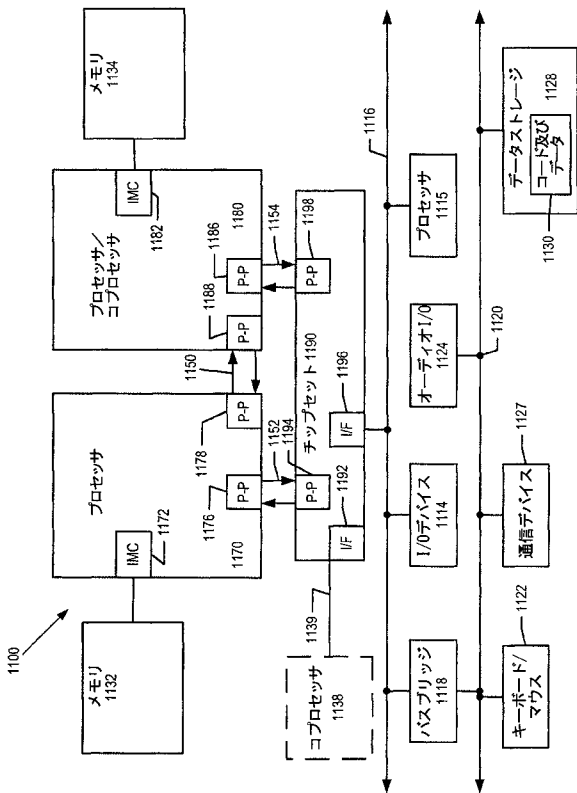
【図9】



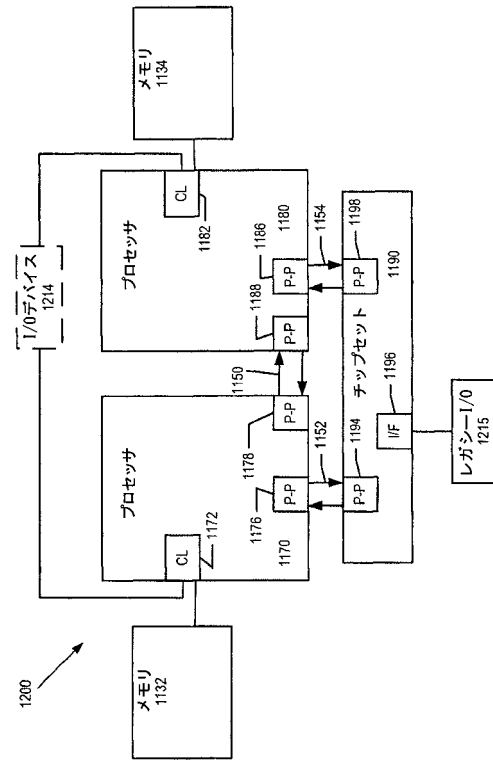
【図10】



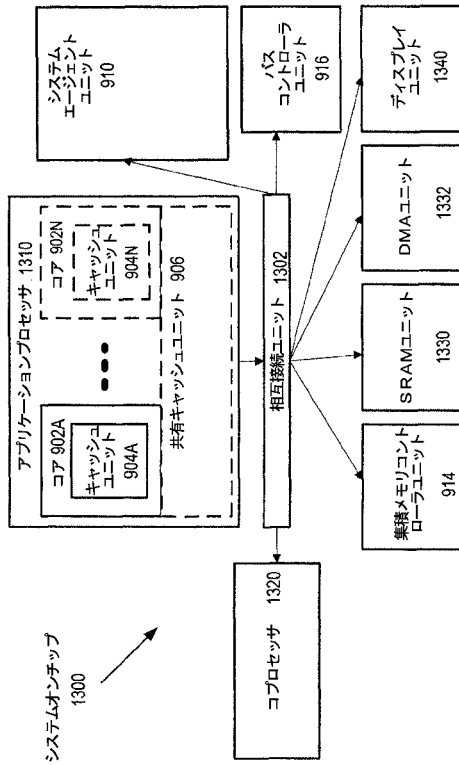
【図11】



【図12】



【図 13】



【手続補正書】

【提出日】平成29年3月10日(2017.3.10)

【手続補正 1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

複数のベクトルレジスタと、前記複数のベクトルレジスタに連結される実行回路とを備え、前記複数のベクトルレジスタの1つは、アレイの複数のデータエレメントを格納し、前記実行回路は、

少なくとも第1のオペランド及び第2のオペランドを指定するマスク生成命令を受信し、

前記マスク生成命令に応答して、前記第2のオペランドにおいて規定された回数だけ、前記第1のオペランドの複数のビットを左シフトし、前記第1のオペランドの最上位ビットがシフトアウトするたびに、1である最下位ビットをプルインすることにより、複数のビットを含む結果を生成し、前記結果の各ビットは、前記アレイの前記複数のデータエレメントの1つに対応し、

前記第2のオペランドは、ベクトルオペレーションの残りのループにおける残りのイテレーションの数を指定する、装置。

フロントページの続き

- (72)発明者 エルモラエフ、イゴール
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 ブレ
バード・2200 インテル・コーポレーション内
- (72)発明者 ナライキン、アンドレー
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 ブレ
バード・2200 インテル・コーポレーション内
- (72)発明者 パレンタイン、ロバート
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 ブレ
バード・2200 インテル・コーポレーション内

Fターム(参考) 5B033 BD03

5B056 BB35 FF08 FF10 FF11