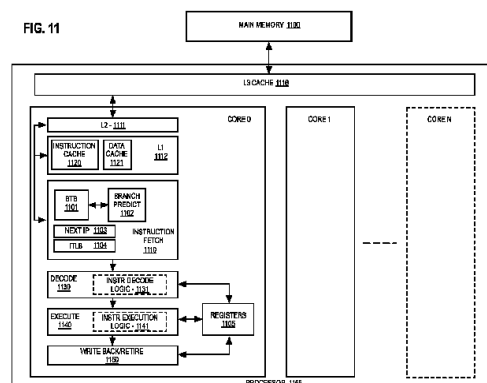




【特許請求の範囲】**【請求項 1】**

第 1 の命令を、第 1 のオペランドと第 2 のオペランドとを含んだ第 1 の復号された命令に復号する復号ロジックと、

前記第 2 のオペランドにより指定されるソースレジスタの両領域のビットを、前記第 1 のオペランドにより示される制御マスクに基づいてインターリーブする逆分離演算を実行するために、前記第 1 の復号された命令を実行する実行ユニットと

を備える
処理装置。

【請求項 2】

10

前記第 1 の命令をフェッチする命令フェッチユニットをさらに備え、前記第 1 の命令は単一の機械レベル命令である、

請求項 1 に記載の処理装置。

【請求項 3】

前記逆分離演算の結果をデスティネーションオペランドにより指定される位置にコミットするレジスタファイルユニットをさらに備える、

請求項 1 又は 2 に記載の処理装置。

【請求項 4】

前記レジスタファイルユニットは、

第 1 のソースオペランド値を格納する第 1 のレジスタと、

20

第 2 のソースオペランド値を格納する第 2 のレジスタと、

前記逆分離演算の前記結果の少なくとも 1 つのデータ要素を格納する第 3 のレジスタとを含むレジスタのセットをさらに格納する、

請求項 3 に記載の処理装置。

【請求項 5】

前記第 1 のレジスタは前記制御マスクを格納し、前記制御マスクの各ビットは、値を読み出すための前記ソースレジスタ内のビット位置を示す、

請求項 4 に記載の処理装置。

【請求項 6】

制御マスクビットの 1 は、前記第 2 のレジスタの第 1 の領域の値が取得されることを示し、制御マスクビットの 0 は、前記第 2 のレジスタの第 2 の領域の値が取得されることを示す、

30

請求項 5 に記載の処理装置。

【請求項 7】

前記第 2 のレジスタの前記第 1 の領域は、前記第 2 のレジスタの下位のバイトオーダービットを含み、前記第 2 のレジスタの前記第 2 の領域は、前記第 2 のレジスタの上位のバイトオーダービットを含む、

請求項 6 に記載の処理装置。

【請求項 8】

前記第 1 のレジスタ又は前記第 2 のレジスタは、32 ビット汎用レジスタ又は 64 ビット汎用レジスタである、

40

請求項 4 から 7 のいずれか一項に記載の処理装置。

【請求項 9】

前記第 1 のレジスタ又は前記第 2 のレジスタはベクトルレジスタである、

請求項 4 から 8 のいずれか一項に記載の処理装置。

【請求項 10】

前記ベクトルレジスタは、パックドデータ要素を格納するための 128 ビットレジスタ、256 ビットレジスタ、又は 512 ビットレジスタである、

請求項 9 に記載の処理装置。

【請求項 11】

50

前記パックドデータ要素は、バイト、ワード、ダブルワード、又はクワッドワードのデータ要素を含み、前記逆分離演算はビットを各データ要素にインターリーブする、請求項 10 に記載の処理装置。

【請求項 12】

プロセッサにより実装される方法であって、
逆分離演算を実行する単一の命令をフェッチする段階であって、前記単一の命令は 2 つのソースオペランドと 1 つのデスティネーションオペランドを有する、段階と、
前記単一の命令を復号された命令に復号する段階と、
少なくとも 1 つのオペランドに関連付けられたソースオペランド値をフェッチする段階と、

10

第 2 のソースオペランドにより指定されるソースレジスタの両領域のビットを、第 1 のソースオペランドにより示される制御マスクに基づいてインターリーブするために、前記復号された命令を実行する段階と
を備える
方法。

【請求項 13】

前記第 1 のソースオペランドは即値オペランドである、
請求項 12 に記載の方法。

【請求項 14】

前記第 1 のソースオペランドは、前記制御マスクを含んだレジスタを指定する、
請求項 12 に記載の方法。

20

【請求項 15】

前記デスティネーションオペランドにより示される位置に結果を書き込む段階をさらに備える、
請求項 12 から 14 のいずれか一項に記載の方法。

【請求項 16】

前記デスティネーションオペランドはベクトルレジスタを示す、
請求項 15 に記載の方法。

【請求項 17】

前記復号された命令を実行する段階は、ソースレジスタの非連続ビットをデスティネーションレジスタに書き込むために、少なくとも 1 つの並列デポジット演算を実行する段階を含む、
請求項 15 に記載の方法。

30

【請求項 18】

前記デスティネーションレジスタは一時レジスタである、
請求項 17 に記載の方法。

【請求項 19】

複数の並列デポジット演算を複数の一時レジスタに対して実行する段階をさらに備える、
請求項 18 に記載の方法。

40

【請求項 20】

前記デスティネーションオペランドにより示される前記位置に前記結果を書き込む段階の前に、前記複数の一時レジスタに対して OR 演算を実行する段階をさらに備える、
請求項 19 に記載の方法。

【請求項 21】

請求項 12 から 20 のいずれか一項に記載の方法を実行する手段を備える
システム。

【請求項 22】

少なくとも 1 つの機械により実行されると、請求項 12 から 20 のいずれか一項に記載の方法を実行する少なくとも 1 つの集積回路を、前記少なくとも 1 つの機械に製造させる

50

データを格納した
機械可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、処理ロジック、マイクロプロセッサ、及び関連する命令セットアーキテクチャの分野に関し、これらはプロセッサ又は他の処理ロジックにより実行される場合、論理的演算、数学的演算、又は他の関数的演算を実行する。

【背景技術】

【0002】

10

特定のタイプのアプリケーションは、同じ演算が多数のデータ項目に対して実行される必要があることが多い（「データ並列処理」と呼ばれる）。単一命令・複数データ処理（SIMD）は、ある演算を複数のデータ項目に対してプロセッサに実行させる命令の一種を指す。SIMD技術は特に、レジスタ中のビットを論理的に複数の固定サイズデータ要素（それぞれが別個の値を表す）に分割し得るプロセッサに適している。例えば、256ビットレジスタ中のビットは、4個の別個の64ビットパックドデータ要素（クワッドワード（Q）サイズのデータ要素）、8個の別個の32ビットパックドデータ要素（ダブルワード（D）サイズのデータ要素）、16個の別個の16ビットパックドデータ要素（ワード（W）サイズのデータ要素）、又は32個の別個の8ビットデータ要素（バイト（B）サイズのデータ要素）として処理されるソースオペランドに指定されてよい。このタイプのデータは「パックド」データタイプ又は「ベクトル」データタイプと呼ばれ、このデータタイプのオペランドは、パックドデータオペランド又はベクトルオペランドと呼ばれる。換言すると、パックドデータ項目又はベクトルは、一連のパックドデータ要素を意味し、パックドデータオペランド又はベクトルオペランドは、SIMD命令のソースオペランド又はデスティネーションオペランドである（パックドデータ命令又はベクトル命令としても知られる）。

20

【図面の簡単な説明】

【0003】

実施形態が、添付図面の図に限定するのではなく、例として示される。

【0004】

30

【図1A】実施形態に従い、例示的なインオーダーフェッチ・復号・リタイアパイプライン、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。

【0005】

【図1B】実施形態に従い、プロセッサに含まれる例示的な実施形態のインオーダーフェッチ・復号・リタイアコア、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。

【0006】

【図2A】より具体的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である。

【図2B】より具体的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である。

40

【0007】

【図3】統合メモリコントローラ及び専用ロジックを搭載した、単一コアプロセッサ及びマルチコアプロセッサのブロック図である。

【0008】

【図4】ある実施形態によるシステムのブロック図を示す。

【0009】

【図5】ある実施形態による第2のシステムのブロック図を示す。

【0010】

【図6】ある実施形態による第3のシステムのブロック図を示す。

【0011】

50

【図 7】ある実施形態によるシステムオンチップ（S o C）のブロック図を示す。

【0 0 1 2】

【図 8】実施形態に従い、ソース命令設定のバイナリ命令をターゲット命令セットのバイナリ命令に変換するソフトウェア命令変換器の使用法を対比させるブロック図を示す。

【0 0 1 3】

【図 9 A】ある実施形態に従い、逆分離演算を実行するビット操作演算を示すブロック図である。

【図 9 B】ある実施形態に従い、逆分離演算を実行するビット操作演算を示すブロック図である。

【図 9 C】ある実施形態に従い、逆分離演算を実行するビット操作演算を示すブロック図である。

【図 9 D】ある実施形態に従い、逆分離演算を実行するビット操作演算を示すブロック図である。

【図 9 E】ある実施形態に従い、逆分離演算を実行するビット操作演算を示すブロック図である。

【0 0 1 4】

【図 1 0】本明細書で説明される実施形態に従ってオペレーションを実行するロジックを含むプロセッサコアのブロック図である。

【0 0 1 5】

【図 1 1】ある実施形態に従い、逆分離演算を実行するロジックを含む処理システムのブロック図である。

【0 0 1 6】

【図 1 2】ある実施形態に従い、例示的な逆分離命令を処理するロジックのフロー図である。

【0 0 1 7】

【図 1 3 A】実施形態に従い、汎用ベクトル対応命令フォーマット、及びその命令テンプレートを示すブロック図である。

【図 1 3 B】実施形態に従い、汎用ベクトル対応命令フォーマット、及びその命令テンプレートを示すブロック図である。

【0 0 1 8】

【図 1 4 A】本発明の実施形態による例示的な特定ベクトル対応命令フォーマットを示すブロック図である。

【図 1 4 B】本発明の実施形態による例示的な特定ベクトル対応命令フォーマットを示すブロック図である。

【図 1 4 C】本発明の実施形態による例示的な特定ベクトル対応命令フォーマットを示すブロック図である。

【図 1 4 D】本発明の実施形態による例示的な特定ベクトル対応命令フォーマットを示すブロック図である。

【0 0 1 9】

【図 1 5】ある実施形態によるスカラレジスタアーキテクチャ及びベクトルレジスタアーキテクチャのブロック図である。

【発明を実施するための形態】

【0 0 2 0】

x 8 6、MMX（登録商標）、ストリーミング SIMD 拡張（SSE）、SSE 2、SSE 3、SSE 4. 1、及び SSE 4. 2 命令を含む命令セットを搭載した Intel（登録商標）Core（登録商標）プロセッサによって利用されるなど、SIMD 技術はアプリケーション性能の著しい改善を可能にした。アドバンスド・ベクトル・エクステンション（AVX）（AVX 1 及び AVX 2）と呼ばれ、ベクトル拡張（VEX）コード体系を用いる SIMD 拡張の追加セットが公開されている（例えば、Intel（登録商標）6 4 及び IA-3 2 アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル（2 0

10

20

30

40

50

14年9月)、及びIntel(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(2014年9月)を参照)。Intel(登録商標)アーキテクチャ(IA)を拡張するアーキテクチャ拡張が説明されている。しかし、基本原理は、いかなる特定のISAにも限定されていない。

【0021】

1つの実施形態では、ベクトルレジスタ又は汎用レジスタを用いて逆分離演算を実行するために、処理デバイスが命令のセットを実行する。逆分離演算は、ソースの両領域のビットをインターリーブし、インターリーブされたビットをデスティネーションに書き込む。命令は制御マスクを用い、1のマスク値を有する各ビットがソースレジスタ又はベクトル要素の一方側から取得され、ゼロのマスクを有するビットが反対側から取得される。逆分離命令は、多くのビット操作ルーチンの構成要素である基本機能を実装するのに用いられてよい。

10

【0022】

本明細書で説明される実施形態に従い、プロセッサコアのアーキテクチャが以下に説明され、その後に例示的なプロセッサ及びコンピュータアーキテクチャの説明が続く。後述される本発明の実施形態について完全な理解を提供するために、多くの具体的な詳細が明記されている。しかし、これらの具体的な詳細の一部がなくても、実施形態は実施され得ることが当業者には明らかであろう。他の例では、様々な実施形態の基本原理をあいまいにしないように、周知の構造及びデバイスがブロック図の形式で示されている。

【0023】

20

プロセッサコアは、異なる方法で、異なる目的のために異なるプロセッサに実装されてよい。例えば、そのようなコアの実装は、1)汎用計算を対象とした汎用インオーダーコア、2)汎用計算を対象とした高性能汎用アウトオブオーダーコア、3)グラフィックス及び／又は科学(スループット)計算を主に対象とした専用コアを含んでよい。プロセッサは、単一のプロセッサコアを用いて実装されてよく、又は複数のプロセッサコアを含むこともできる。プロセッサ内の複数のプロセッサコアは、アーキテクチャ命令セットに関して同種でも異種でもよい。

【0024】

異なるプロセッサの実装は、1)汎用計算用の1つ又は複数の汎用インオーダーコア、及び／又は、汎用計算を対象とした1つ又は複数の汎用アウトオブオーダーコアを含む中央処理装置、並びに2)グラフィックス及び／又は科学的な用途を主に対象とした1つ又は複数の専用コア(例えば、多くの統合コアプロセッサ)を含むコプロセッサを含む。そのような異なるプロセッサによって異なるコンピュータシステムアーキテクチャがもたらされ、そこには次のものが含まれる。つまり、1)中央システムプロセッサとは別個のチップに搭載されたコプロセッサ、2)中央システムプロセッサと同じパッケージ内の別個のダイに搭載されたコプロセッサ、3)他のプロセッサコアと同じダイに搭載されたコプロセッサ(この場合、そのようなコプロセッサは統合グラフィックスロジック及び／又は科学(スループット)ロジックなどの専用ロジック、又は専用コアと呼ばれることがある)、及び4)説明されたプロセッサ(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呼ばれることがある)、上述のコプロセッサ、及び追加機能を同じダイ上に含み得るシステムオンチップである。

30

40

[例示的なコアアーキテクチャ]

[インオーダーコア及びアウトオブオーダーコアのブロック図]

【0025】

図1Aは、ある実施形態に従い、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／実行パイプラインを示すブロック図である。図1Bは、ある実施形態に従い、プロセッサに含まれるインオーダーアーキテクチャコアの例示的な実施形態と、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との両方を示すブロック図である。図1A～図1Bの実線で示されたボックスは、インオーダーパイプライン及びインオーダーコアを示す。一方、破線で示されたボック

50

スの任意の追加は、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仮定して、アウトオブオーダー態様が説明される。

【0026】

図1Aにおいて、プロセッサパイプライン100は、フェッチステージ102、レンダス復号ステージ104、復号ステージ106、割り当てステージ108、リネーミングステージ110、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112、レジスタ読み出し／メモリ読み出しステージ114、実行ステージ116、ライトバック／メモリ書き込みステージ118、例外処理ステージ122、及びコミットステージ124を含む。

10

【0027】

図1Bは、実行エンジンユニット150に結合されたフロントエンドユニット130を含むプロセッサコア190を示し、両方ともメモリユニット170に結合されている。コア190は、縮小命令セット計算（RISC）コア、複合命令セット計算（CISC）コア、超長命令語（VLIW）コア、あるいはハイブリッド又は代替的なコアタイプであってよい。さらに別の選択肢として、コア190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、汎用計算グラフィックス処理ユニット（GPGPU）コア、グラフィックスコアなどの専用コアであってもよい。

【0028】

フロントエンドユニット130は、命令キャッシュユニット134に結合された分岐予測ユニット132を含み、命令キャッシュユニット134は命令変換ルックアサイドバッファ（TLB）136に結合され、命令変換ルックアサイドバッファ（TLB）136は命令フェッチユニット138に結合され、命令フェッチユニット138は復号ユニット140に結合されている。復号ユニット140（又はデコーダ）は、複数の命令を復号し、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、又は他の制御信号を出力として生成し得る。これらは、元の命令から復号され、又は別の方法で元の命令を反映し、又は元の命令から導出される。復号ユニット140は、様々な異なるメカニズムを用いて実装されてよい。適切なメカニズムの例には、限定されないが、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）などが含まれる。1つの実施形態では、コア190は、特定のマクロ命令用のマイクロコードを（例えば、復号ユニット140の中に、そうでなければフロントエンドユニット130内に）格納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140は、実行エンジンユニット150内のリネーム／アロケータユニット152に結合されている。

20

30

【0029】

実行エンジンユニット150は、リタイアメントユニット154と、1つ又は複数のスケジューラユニット156のセットとに結合されたリネーム／アロケータユニット152を含む。スケジューラユニット156は、リザベーションステーション、中央命令ウィンドウなど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156は、物理レジスタファイルユニット158に結合されている。物理レジスタファイルユニット158のそれぞれは、1つ又は複数の物理レジスタファイルを表し、そのそれぞれ異なる物理レジスタファイルは、スカラ整数、スカラ浮動小数点、パックド整数、パックド浮動小数点、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）など、1つ又は複数の異なるデータタイプを格納する。1つの実施形態では、物理レジスタファイルユニット158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含む。これらのレジスタユニットは、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得る。物理レジスタファイルユニット158は、リタイアメントユニット154によってオーバーラップされ、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得る様々な方法を示す（例えば、リオーダーバッファ及びリタイアメント

40

50

レジスタファイルを用いる、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る、並びにレジスタマップ及びレジスタのプールを用いるなど）。リタイアメントユニット154及び物理レジスタファイルユニット158は、実行クラスタ160に結合されている。実行クラスタ160は、1つ又は複数の実行ユニット162のセットと、1つ又は複数のメモリアクセスユニット164のセットとを含む。実行ユニット162は、様々な演算（例えば、シフト、加算、減算、乗算）を様々なタイプのデータ（例えば、スカラ浮動小数点、バックド整数、バックド浮動小数点、ベクトル整数、ベクトル浮動小数点）に実行してよい。いくつかの実施形態は、特定の機能又は機能のセットに専用の複数の実行ユニットを含んでよく、他の実施形態は、1つのみの実行ユニット、又は全ての機能を全て実行する複数の実行ユニットを含んでもよい。特定の実施形態は、特定のタイプのデータ／オペレーションに対して別個のパイプラインを形成するので、スケジューラユニット156、物理レジスタファイルユニット158、及び実行クラスタ160は、可能性として複数であると示されている（例えば、スカラ整数パイプライン、スカラ浮動小数点／バックド整数／バックド浮動小数点／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はそれぞれ、独自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し、別個のメモリアクセスパイプラインの場合には、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164を有する特定の実施形態が実装される）。別個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1つ又は複数がアウトオブオーダー発行／実行であってよく、残りがインオーダーであってもよいことも理解されるべきである。

10

20

【0030】

メモリアクセスユニット164のセットがメモリユニット170に結合され、メモリユニット170は、レベル2（L2）キャッシュユニット176に結合されたデータキャッシュユニット174に結合されたデータTLBユニット172を含む。1つの例示的な実施形態において、メモリアクセスユニット164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、これらのそれぞれはメモリユニット170内のデータTLBユニット172に結合されている。命令キャッシュユニット134は、メモリユニット170内のレベル2（L2）キャッシュユニット176にさらに結合される。L2キャッシュユニット176は、1つ又は複数の他のレベルのキャッシュに結合され、最終的にはメインメモリに結合される。

30

【0031】

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、パイプライン100を以下のように実装してよい。すなわち、1）命令フェッチ138がフェッチステージ102及びレングス復号ステージ104を実行する。2）復号ユニット140が復号ステージ106を実行する。3）リネーム／アロケータユニット152が割り当てステージ108及びリネーミングステージ110を実行する。4）スケジューラユニット156がスケジュールステージ112を実行する。5）物理レジスタファイルユニット158及びメモリユニット170がレジスタ読み出し／メモリ読み出しステージ114を実行する。実行クラスタ160が実行ステージ116を実行する。6）メモリユニット170及び物理レジスタファイルユニット158がライトバック／メモリ書き込みステージ118を実行する。7）様々なユニットが例外処理ステージ122に関与し得る。8）リタイアメントユニット154及び物理レジスタファイルユニット158がコミットステージ124を実行する。

40

【0032】

コア190は、本明細書で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、x86命令セット（より新しいバージョンと共に追加されたいくつかの拡張を有する）、MIPS Technologies（カリフォルニア州／サンニール）のMIPS命令セット、ARM Holdings（英国／ケンブリッジ）のARM（登録商標）命令セット（NEONなどの任意の追加拡張を有する））をサポートしてよい。1つの実施形態では、コア190はバックドデータ命令セット拡張（例えば、AVX1、AVX2など

50

）をサポートするロジックを含み、多くのマルチメディアアプリケーションにより用いられるオペレーションがバックドデータを用いて実行されることを可能にする。

【0033】

コアはマルチスレッディング（オペレーション又はスレッドからなる2つ又はそれより多くの並列セットを実行）をサポートしてよく、タイムスライスマルチスレッディング、同時マルチスレッディング（物理コアが同時にマルチスレッディングしているスレッドのそれぞれに対して、単一の物理コアが論理コアを提供する）、又はこれらの組み合わせ（例えば、タイムスライスフェッチ及び復号、並びにそれ以降のIntel（登録商標）ハイパースレッディング・テクノロジーなどの同時マルチスレッディング）を含む様々な方法でサポートしてよいことが理解されるべきである。

10

【0034】

レジスタリネーミングがアウトオブオーダー実行との関連で説明されるが、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もよいことが理解されるべきである。示されたプロセッサの実施形態はまた、別々の命令キャッシュユニット134とデータキャッシュユニット174、並びに共有L2キャッシュユニット176を含むが、代替的な実施形態は、命令及びデータの両方に対して、例えばレベル1（L1）内部キャッシュ又は複数のレベルの内部キャッシュなど、単一の内部キャッシュを有してもよい。実施形態によっては、システムは、内部キャッシュ及び外部キャッシュの組み合わせを含んでよく、外部キャッシュはコア及び／又はプロセッサの外部に存在する。あるいは、全てのキャッシュが、コア及び／又はプロセッサの外部にあってもよい。

20

〔具体的な例示的インオーダーコアアーキテクチャ〕

【0035】

図2A～図2B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（同じタイプ及び／又は異なるタイプの他のコアを含む）の1つになるであろう。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）を通じて、何ら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ロジックと通信する。

【0036】

図2Aは、ある実施形態による単一のプロセッサコアのブロック図であり、オンダイ相互接続ネットワーク202への接続に加え、レベル2（L2）キャッシュ204のローカルサブセットを有する。1つの実施形態では、命令デコーダ200はバックドデータ命令セット拡張を用いてx86命令セットをサポートする。L1キャッシュ206によって、キャッシュメモリからスカラユニット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となる。1つの実施形態では、（設計を簡略化するために）スカラユニット208及びベクトルユニット210が、別々のレジスタセット（それぞれ、複数のスカラレジスタ212及び複数のベクトルレジスタ214）を用い、これらの間で転送されるデータはメモリに書き込まれ、その後、レベル1（L1）キャッシュ206から読み戻されるが、本発明の代替的な実施形態は、異なる手法を用いてよい（例えば、単一のレジスタセットを用いる、又は書き込み及び読み戻しを行うことなく、2つのレジスタファイル間でのデータ転送を可能にする通信経路を含む）。

30

40

【0037】

L2キャッシュのローカルサブセット204は、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り、プロセッサコアごとに1つである。各プロセッサコアは、独自のL2キャッシュのローカルサブセット204に直接アクセスする経路を有する。プロセッサコアにより読み出されたデータは、L2キャッシュのサブセット204に格納され、他のプロセッサコアが独自のローカルL2キャッシュのサブセットにアクセスすると並行して、高速にアクセスされ得る。プロセッサコアにより書き込まれたデータは、独自のL2キャッシュのサブセット204に格納され、必要に応じて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、共有データのコヒーレンスを保証

50

する。リングネットワークは双方向性であり、プロセッサコア、L 2 キャッシュ、及び他の論理ブロックなどのエージェントが、チップ内で互いに通信することを可能にする。各リングデータ経路は、一方向当たり 1012 ビット幅である。

【0038】

図 2 B は、ある実施形態による図 2 A のプロセッサコアの一部に関する拡大図である。図 2 B は、L 1 キャッシュ 204 の一部である L 1 データキャッシュ 206 A と、ベクトルユニット 210 及びベクトルレジスタ 214 に関するより詳細とを含む。具体的には、ベクトルユニット 210 は 16 幅のベクトル処理ユニット (VPU) (16 幅の ALU 228 を参照) であり、整数命令、単精度浮動小数点命令、及び倍精度浮動小数点命令のうち 1 つ又は複数を実行する。VPU は、スウィズルユニット 220 を用いたレジスタ入力のスウィズル処理、数値変換ユニット 222 A ~ 222 B を用いた数値変換、並びに複製ユニット 224 を用いたメモリ入力の複製をサポートする。書き込みマスクレジスタ 226 は、結果として生じるベクトル書き込みをプレディケートする (predicating) ことを可能にする。

[統合メモリコントローラ及び専用ロジックを有するプロセッサ]

【0039】

図 3 は、ある実施形態によるプロセッサ 300 のブロック図であり、これは 1 つより多くのコアを有してよく、統合メモリコントローラを有してよく、統合グラフィックスを有してよい。図 3 の実線で示されたボックスは、単一のコア 302 A、システムエージェント 310、1 つ又は複数のバスコントローラユニット 316 のセットを有するプロセッサ 300 を示し、破線で示されたボックスの任意の追加は、複数のコア 302 A ~ 302 N、システムエージェントユニット 310 内にある 1 つ又は複数の統合メモリコントローラユニット 314 のセット、及び専用ロジック 308 を有する代替プロセッサ 300 を示す。

【0040】

したがって、プロセッサ 300 の異なる実装は、1) 専用ロジック 308 が統合グラフィックス及び／又は科学 (スループット) ロジック (1 つ又は複数のコアを含んでよい) であり、コア 302 A ~ 302 N が 1 つ又は複数の汎用コア (例えば、汎用インオーダーコア、汎用アウトオブオーダーコア、その 2 つの組み合わせ) である CPU、2) コア 302 A ~ 302 N が、グラフィックス及び／又は科学 (スループット) を主に対象とした多数の専用コアであるコプロセッサ、並びに 3) コア 302 A ~ 302 N が多数の汎用インオーダーコアであるコプロセッサを含んでよい。したがって、プロセッサ 300 は汎用プロセッサ、コプロセッサであってよく、あるいは専用プロセッサ、例えばネットワークプロセッサ又は通信プロセッサ、圧縮エンジン、グラフィックスプロセッサ、GPGPU (汎用グラフィックス処理ユニット)、高スループットの多数統合コア (MIC) コプロセッサ (30 個又はそれより多くのコアを含む)、組み込みプロセッサなどであってもよい。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 300 は、例えば、BiCMOS、CMOS、又は NMOS などの複数のプロセス技術のいずれかを用いる 1 つ又は複数の基板の一部であってよく、及び／又は当該基板上に実装されてもよい。

【0041】

メモリ階層は、コア内にある 1 つ又は複数のレベルのキャッシュと、共有キャッシュユニット 306 のセットあるいは 1 つ又は複数の共有キャッシュユニット 306 と、統合メモリコントローラユニット 314 のセットに結合された外部メモリ (不図示) とを含む。共有キャッシュユニット 306 のセットは、レベル 2 (L2)、レベル 3 (L3)、レベル 4 (L4)、又は他のレベルのキャッシュなど、1 つ又は複数の中間レベルのキャッシュ、又は他のレベルのキャッシュ、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。1 つの実施形態では、リングベースの相互接続ユニット 312 が、統合グラフィックスロジック 308、共有キャッシュユニット 306 のセット、及びシステムエージェントユニット 310 / 統合メモリコントローラユニット 314 を相互接続するが、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のに任意の数の

周知手法を用いてよい。1つの実施形態では、1つ又は複数のキャッシュユニット306と、コア302A～302Nとの間でコヒーレンシが維持される。

【0042】

実施形態によっては、コア302A～302Nのうち1つ又は複数のマルチスレッディング可能である。システムエージェント310は、コア302A～302Nを調整し動作させるこうしたコンポーネントを含む。システムエージェントユニット310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302A～302N及び統合グラフィックスロジック308の電源状態を管理するのに必要なロジック及びコンポーネントであってよく、又は当該ロジック及び当該コンポーネントを含んでもよい。ディスプレイユニットは、外部接続された1つ又は複数のディスプレイを駆動するためのものである。

10

【0043】

コア302A～302Nは、アーキテクチャ命令セットに関して同種でも異種でもよい。すなわち、コア302A～302Nのうち2つ又はそれより多くは同じ命令セットを実行することが可能であってよいが、他のものはその命令セットのサブセット又は別の命令セットだけを実行することが可能であってよい。

[例示的なコンピュータアーキテクチャ]

【0044】

図4～図7は、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。ラップトップ型PC、デスクトップ型PC、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、デジタル信号プロセッサ(DSP)、グラフィックスデバイス、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、携帯型メディアプレーヤ、ハンドヘルド型デバイス、及び様々な他の電子デバイス向けの当技術分野において知られる他のシステム設計及び構成も適している。概して、本明細書に開示されるプロセッサ及び／又は他の実行ロジックを組み込むことが可能である多様なシステム又は電子デバイスが一般的に適している。

20

【0045】

図4は、ある実施形態によるシステム400のブロック図を示す。システム400は、1つ又は複数のプロセッサ410、415を含んでよく、これらはコントローラハブ420に結合されている。1つの実施形態では、コントローラハブ420は、グラフィックスメモリコントローラハブ(GMCH)490と、入力／出力ハブ(IOH)450(これは別個のチップ上にあってよい)とを含む。GMCH490は、メモリ及びグラフィックスコントローラを含み、これらにメモリ440及びコプロセッサ445が結合されている。IOH450は入力／出力(I/O)デバイス460をGMCH490に結合する。あるいは、メモリ及びグラフィックスコントローラの一方又は両方が、(本明細書で説明されるように)プロセッサ内に統合され、メモリ440及びコプロセッサ445は、プロセッサ410と、IOH450と共に単一チップに入ったコントローラハブ420とに直接結合される。

30

【0046】

任意的な性質の追加のプロセッサ415は、図4に破線で示されている。各プロセッサ410、415は、本明細書で説明される処理コアのうち1つ又は複数を含んでよく、何らかのバージョンのプロセッサ300であってよい。

40

【0047】

メモリ440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)、又はこの2つの組み合わせであってよい。少なくとも1つの実施形態では、コントローラハブ420は、フロントサイドバス(FSB)などのマルチドロップバス、QuickPath相互接続(QPI)などのポイントツーポイントインタフェース、又は同種の接続495を介してプロセッサ410、415と通信する。

【0048】

50

1つの実施形態では、コプロセッサ445は、例えば、ハイスループットMICプロセッサ、ネットワークプロセッサ又は通信プロセッサ、圧縮エンジン、グラフィックスプロセッサ、GPGPU、組み込みプロセッサなどの専用プロセッサである。1つの実施形態では、コントローラハブ420は統合グラフィックスアクセラレータを含んでよい。

【0049】

物理リソース410と415との間には、アーキテクチャ特性、マイクロアーキテクチャ特性、熱的特性、電力消費特性などを含む広範な価値基準に関して、様々な差異が存在し得る。

【0050】

1つの実施形態では、プロセッサ410は、一般的タイプのデータ処理オペレーションを制御する命令を実行する。この命令内にコプロセッサ命令が組み込まれてもよい。プロセッサ410は、これらのコプロセッサ命令を、付属のコプロセッサ445が実行すべきタイプの命令であると認識する。したがって、プロセッサ410は、これらのコプロセッサ命令（又はコプロセッサ命令を表す制御信号）をコプロセッサバス又は他の相互接続を使ってコプロセッサ445に発行する。コプロセッサ445は、受信したコプロセッサ命令を受け付けて実行する。

【0051】

図5は、ある実施形態に従って、より具体的な第1の例示的なシステム500のブロック図を示す。図5に示されるように、マルチプロセッサシステム500はポイントツーポイント相互接続システムであり、ポイントツーポイント相互接続550を介して結合される第1のプロセッサ570と、第2のプロセッサ580とを含む。プロセッサ570及び580のそれぞれは、何らかのバージョンのプロセッサ300であってよい。本発明の1つの実施形態では、プロセッサ570及び580はそれぞれ、プロセッサ410及び415であり、コプロセッサ538はコプロセッサ445である。別の実施形態では、プロセッサ570及び580はそれぞれ、プロセッサ410及びコプロセッサ445である。

【0052】

プロセッサ570及び580は、統合メモリコントローラ（IMC）ユニット572及び582をそれぞれ含んで示されている。プロセッサ570はまた、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576及び578を含み、同様に第2のプロセッサ580はP-Pインタフェース586及び588を含む。プロセッサ570、580は、ポイントツーポイント（P-P）インタフェース550を介し、P-Pインタフェース回路578、588を用いて情報を交換してよい。図5に示されるように、IMC572及び582は、プロセッサをそれぞれのメモリ、すなわちメモリ532及びメモリ534に結合する。これらのメモリは、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。

【0053】

プロセッサ570、580はそれぞれ、個々のP-Pインタフェース552、554を介し、ポイントツーポイントインタフェース回路576、594、586、598を用いてチップセット590と情報を交換してよい。チップセット590は任意で、高性能インタフェース539を介してコプロセッサ538と情報を交換してよい。1つの実施形態では、コプロセッサ538は、例えば、ハイスループットMICプロセッサ、ネットワークプロセッサ又は通信プロセッサ、圧縮エンジン、グラフィックスプロセッサ、GPGPU、組み込みプロセッサなどの専用プロセッサである。

【0054】

共有キャッシュ（不図示）がどちらかのプロセッサに含まれても、又は両方のプロセッサの外部に含まれてもよく、さらにP-P相互接続を介してこれらのプロセッサに接続されてもよい。これにより、プロセッサが低電力モードに入っている場合に、どちらかのプロセッサ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に格納され得る。

【0055】

10

20

30

40

50

チップセット 590 は、インタフェース 596 を介して第 1 のバス 516 に結合されてよい。1 つの実施形態では、第 1 のバス 516 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、あるいは PCI エクスプレスバス又は別の第 3 世代 I/O 相互接続バスなどのバスであってよいが、本発明の範囲はそうのように限定されてはいない。

【0056】

図 5 に示されるように、第 1 のバス 516 を第 2 のバス 520 に結合するバスブリッジ 518 と共に、様々な I/O デバイス 514 が第 1 のバス 516 に結合されてよい。1 つの実施形態では、1 つ又は複数の追加のプロセッサ 515 が第 1 のバス 516 に結合される。追加のプロセッサとは、コプロセッサ、ハイスループット MIC プロセッサ、GPU のアクセラレータ (例えば、グラフィックスアクセラレータ、又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、又はその他のプロセッサなどである。1 つの実施形態では、第 2 のバス 520 はローピンカウント (LPC) バスであってよい。様々なデバイスが第 2 のバス 520 に結合されてよく、1 つの実施形態では、そのようなデバイスには例えば、キーボード及び/又はマウス 522、通信デバイス 527、及びストレージユニット 528 が含まれ、ストレージユニットには、命令/コード及びデータ 530 を含む得るディスクドライブ又は他の大容量ストレージデバイスなどがある。さらに、オーディオ I/O 524 が第 2 のバス 520 に結合されてよい。他のアーキテクチャも可能であることに留意されたい。例えば、図 5 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アーキテクチャ又は他のそのようなアーキテクチャを実装してよい。

【0057】

図 6 は、ある実施形態に従って、より具体的な第 2 の例示的なシステム 600 のブロック図を示す。図 5 及び図 6 内の同様の要素は同様の参照番号を有しており、図 5 の特定の態様が、図 6 の他の態様をあいまいにしないために、図 6 から省略されている。

【0058】

図 6 は、プロセッサ 570、580 がそれぞれ、統合メモリと、I/O 制御ロジック (「CL」) 572 及び 582 とを含んでよいことを示す。したがって、CL 572、582 は統合メモリコントローラユニットを含み、且つ I/O 制御ロジックを含む。図 6 は、メモリ 532、534 だけが CL 572、582 に結合されているのではなく、I/O デバイス 614 もまた、制御ロジック 572、582 に結合されていることを示している。レガシ I/O デバイス 615 がチップセット 590 に結合されている。

【0059】

図 7 は、ある実施形態に従って SOC 700 のブロック図を示す。図 3 内の同種の要素は同様の参照番号を有している。また、破線で示されるボックスは、より高度な SOC における任意の機能である。図 7 において、相互接続ユニット 702 が、1 つ又は複数のコア 302A~302N 及び共有キャッシュユニット 306 のセットを含むアプリケーションプロセッサ 710 と、システムエージェントユニット 310 と、バスコントローラユニット 316 と、統合メモリコントローラユニット 314 と、統合グラフィックスロジック、画像プロセッサ、オーディオプロセッサ、及び映像プロセッサを含み得る 1 つ又は複数のコプロセッサ 720 又はそのセットと、スタティックランダムアクセスメモリ (SRAM) ユニット 730 と、ダイレクトメモリアクセス (DMA) ユニット 732 と、1 つ又は複数の外部ディスプレイに結合するためのディスプレイユニット 740 とに結合されている。1 つの実施形態では、コプロセッサ 720 は専用プロセッサを含み、例えば、ネットワークプロセッサ又は通信プロセッサ、圧縮エンジン、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がある。

【0060】

本明細書に開示される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、又はそのような実装手法の組み合わせで実装される。実施形態は、少なくとも 1 つのプロセッサと、ストレージシステム (揮発性メモリ及び不揮発性メモリ、及び/又

は記憶素子を含む)と、少なくとも1つの入力デバイスと、少なくとも1つの出力デバイスとを有するプログラマブルシステム上で実行される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る。

【0061】

図5に示されるコード530などのプログラムコードは、本明細書で説明される機能を実行し、出力情報を生成する命令を入力するのに適用されてよい。出力情報は、1つ又は複数の出力デバイスに既知の方法で適用されてよい。本願の目的のために、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などのプロセッサを有する任意のシステムを含む。

10

【0062】

プログラムコードは、処理システムと通信すべく、高水準の手続き型又はオブジェクト指向型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また、必要に応じて、アセンブリ言語又は機械語で実装されてよい。実際には、本明細書で説明されるメカニズムは、いかなる特定のプログラミング言語にも範囲を限定されない。どのような場合でも、言語はコンパイラ型言語又はインタプリタ型言語であってよい。

【0063】

少なくとも1つの実施形態の1つ又は複数の態様は、機械可読媒体に格納された典型的なデータにより実装されてよい。この命令は、プロセッサ内の様々なロジックを表し、機械により読み出された場合、本明細書で説明される手法を実行すべく機械にロジックを作成させる。「IPコア」として知られるそのような表現は、有形の機械可読媒体(「テープ」)に格納され、ロジック又はプロセッサを実際に作成する製造装置にロードすべく、様々な顧客又は製造施設に供給されてよい。例えば、ARM Holdings, Ltd. 及び、中国科学院の計算技術研究所(ICT)が開発したプロセッサなどのIPコアは、様々な顧客又はライセンス先にライセンス供与又は販売されてよく、これらの顧客又はライセンス先によって製造されたプロセッサに実装されてよい。

20

【0064】

そのような機械可読記憶媒体は、限定されることなく、機械又は装置により製造される又は形成される非一時的な有形の構成の物品を含んでよく、そのような物品には、ハードディスクや、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)、リライタブル・コンパクトディスク(CD-RW)、及び光磁気ディスクを含むその他のタイプのディスク、半導体デバイスとして、例えば、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)やスタティックランダムアクセスメモリ(SRAM)などの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)、相変化メモリ(PCM)など、磁気カード又は光カード、又は電子命令を格納するのに適したその他のタイプの媒体などの記憶媒体を含む。

30

【0065】

したがって、実施形態はまた、命令を含んだ、又はハードウェア記述言語(HDL)などの設計データを含んだ非一時的な有形の機械可読媒体を含む。HDLは、本明細書で説明される構造、回路、装置、プロセッサ、及び/又はシステム機能を定義する。そのような実施形態はまた、プログラム製品と呼ばれ得る。

40

[エミュレーション(バイナリ変換、コードモーフィングなどを含む)]

【0066】

場合によっては、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のに命令変換器が用いられてよい。例えば命令変換器は、ある命令を、コアによって処理される1つ又は複数の他の命令に翻訳(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)、モーフィング、エミュレーション、又は別の方法で変換してよい。命令変換器は、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

50

わせで実装されてよい。命令変換器は、プロセッサ上にあっても、プロセッサ外にあっても、又は一部がプロセッサ上にあり且つ一部がプロセッサ外にあってもよい。

【0067】

図8は、ある実施形態に従って、ソース命令セットのバイナリ命令をターゲット命令セットのバイナリ命令に変換するソフトウェア命令変換器の使用法を対比するブロック図である。図示された実施形態では、命令変換器はソフトウェア命令変換器であるが、代わりに命令変換器は、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もよい。図8は、高水準言語802のプログラムがx86コンパイラ804を用いてコンパイルされ、少なくとも1つのx86命令セットコアを搭載するプロセッサ816によってネイティブに実行され得るx86バイナリコード806を生成し得ることを示す。

10

【0068】

少なくとも1つのx86命令セットコアを搭載するプロセッサ816は、少なくとも1つのx86命令セットコアを搭載するIntel（登録商標）プロセッサと実質的に同じ結果を実現するために、（1）Intel（登録商標）x86命令セットコアの命令セットの大部分、又は（2）少なくとも1つのx86命令セットコアを搭載するIntel（登録商標）プロセッサ上で動作することを目的としたオブジェクトコード形式のアプリケーション又は他のソフトウェアを、互換的に実行する、又は別の方法で処理することで、少なくとも1つのx86命令セットコアを搭載するIntel（登録商標）プロセッサと実質的に同じ機能を実行し得る任意のプロセッサを表す。x86コンパイラ804は、追加のリンケージ処理をしてもしなくても、少なくとも1つのx86命令セットコアを搭載するプロセッサ816上で実行され得るx86バイナリコード806（例えば、オブジェクトコード）を生成するよう動作可能なコンパイラを表す。同様に、図8は、高水準言語802のプログラムが、別の命令セットコンパイラ808を用いてコンパイルされ、少なくとも1つのx86命令セットコアを搭載しないプロセッサ814（例えば、MIPS Technologies（カリフォルニア州／サニーベール）のMIPS命令セットを実行するコア、及び／又は、ARM Holdings（英国／ケンブリッジ）のARM命令セットを実行するコアを搭載したプロセッサ）によりネイティブに実行され得る別の命令セットバイナリコード810を生成し得ることを示す。

20

【0069】

命令変換器812は、x86バイナリコード806を、x86命令セットコアを搭載しないプロセッサ814によりネイティブに実行され得るコードに変換するのに用いられる。この変換されたコードは、別の命令セットバイナリコード810と同じになる可能性は低い。なぜなら、同じにできる命令変換器を作るのは難しいからである。しかし、変換されたコードは一般的なオペレーションを実現し、別の命令セットの命令で構成される。したがって、命令変換器812は、エミュレーション、シミュレーション、又はそ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持た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806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。

30

[逆分離命令]

40

[逆分離演算]

【0070】

本明細書で説明される実施形態は、ビット単位の分離演算の逆演算を実行する。「羊と山羊（sheep and goats）」とも呼ばれる分離演算では、マスクビット1に当たるビットがデスティネーション要素の一方側（例えば、右側）に分離され、0に当たるビットがデスティネーション要素の他方側（例えば、左側）に置かれる。逆分離演算では、ソースレジスタの両側のビットがデスティネーションレジスタにインターリーブされる。汎用レジスタ又はベクトルレジスタが、ソースレジスタ又はデスティネーションレジスタとして用いられてよい。1つの実施形態では、32ビットレジスタ又は64ビットレジスタを含む汎用レジスタがサポートされている。1つの実施形態では、128ビット

50

、256ビット、又は512ビットを含むベクトルレジスタがサポートされ、ベクトルレジスタは、パックドバイト、ワード、ダブルワード、又はクワッドワードのデータ要素へのサポートを有している。

【0071】

既存の命令セットからの命令を用いて逆分離を実行するには、一連の複数の命令を必要とする。既存の命令セットは、逆分離演算を実行するのに必要な命令数を減少させる拡張命令を含んでよいが、本明細書で説明される実施形態は、単一の命令で逆分離機能を実行する。1つの実施形態では、本明細書で説明される逆分離命令は、マスク値を示す第1のソースオペランドを含む。1の値を持つ各マスクビットは、デスティネーションレジスタの対応するビットがソースレジスタの「右」側から取得されることを示す。0の値を持つマスクビットは、ソースレジスタの「左」側から取得される。1つの実施形態では、ソースレジスタは第2のソースオペランドで示されている。

10

【0072】

逆分離命令について、例示的なソースレジスタの値及びデスティネーションレジスタの値が以下の表1に示されている。

【表1】

逆分離命令

ビット	FEDCBA9876543210
SRC1	1010001110101110
SRC2	omlkgeapnjihfdcb
DEST	ponmlkjihgfedcba

20

【0073】

上記の表1では、SRC1オペランドはビットマスク値を格納するマスクレジスタを示す。SRC2オペランドは、逆分離演算用のソース値を格納するレジスタを示す。SRC2値を示すのに用いられる文字は、特定の値を示すのではなく、ビットフィールド内の特定のビット位置を示すように示されている。DESTオペランドは、逆分離命令の出力を格納するデスティネーションレジスタを示す。表1には例示的な16ビットが示されているが、様々な実施形態では、命令は32ビット汎用レジスタオペランド又は64ビット汎用レジスタオペランドを受け入れる。1つの実施形態では、ベクトル命令は、パックドバイト、ワード、ダブルワード、又はクワッドワードのデータ要素を有するベクトルレジスタ上で動作するよう実装される。1つの実施形態では、レジスタは、128ビットレジスタ、256ビットレジスタ、及び512ビットレジスタを含む。

30

【0074】

例示的な命令のオペレーションを示すために、以下の表2は、レジスタのセットに対して逆分離演算を実行するのに用いられ得る例示的な一連の複数のIntel（登録商標）アーキテクチャ（IA）命令を示す。例示的な命令は、ポピュレーションカウント命令、並列デポジット命令、及びシフト命令を含む。1つの実施形態では、ベクトル命令が、複数のベクトルデータ要素にわたって並列に実行されるのに用いられてもよい。

40

【表 2】

逆分離演算

01	popcnt rsi, rbx
02	pdep rax, rcx, rbx
03	shrx rcx, rcx, rsi
04	not rbx
05	pdep rdx, rcx, rbx
06	or rax, rdx

10

【0075】

上記の表 2 に示される例示的な逆分離ロジックにおいて、「popcnt」記号は、ポピュレーションカウント命令を示す。ポピュレーションカウント命令は、入力ビットフィールドのハミング重み（例えば、等しい長さのゼロビットフィールドからの、ビットフィールドのハミング距離）を計算する。この命令は、1 にセットされるビットの数を決定するために、ビットマスク上で用いられる。1 つの実施形態では、ビットフィールドにおいて 1 にセットされるビットの数は、レジスタの「右」側と「左」側とを分けるディバイダを決定する。「pdep」記号は、並列デポジット命令を示す。1 つの実施形態では、並列デポジット命令は、右寄せしたビットのフィールドをソースレジスタから取り出し、ビットマスクにより示される異なる非連続位置にこれらのビットをデポジットする。「shrx」記号は、論理的な右シフト命令を示し、この命令は、指定された数のビット位置だけ、ソースビットフィールドを右にシフトする。

20

【0076】

示される例示的な「否定（not）」命令及び「論理和（or）」命令は、これらの命令が名付けられた論理演算をそれぞれ実行する。「否定（not）」命令は、入力値の論理補数を計算する（例えば、1 のビットはそれぞれ 0 のビットになる）。「論理和（or）」命令は、ソースオペランドにより示されるレジスタの値の論理和を計算する。SRC 1 及び SRC 2 の値から表 1 の DEST 値を計算する論理演算は、表 2 の例示的なロジックを用いて、図 9 A ～ 図 9 E に示されている。

30

【0077】

図 9 A ～ 図 9 E は、ある実施形態に従い、逆分離演算を実行するビット操作演算を示すブロック図である。図 9 A に示されるように、表 2 の行（2）にも示される並列デポジット演算が、SRC 1（904）に提供されるビットに基づいて、SRC 2（902）のビットを一時レジスタ（例えば、TMP 1（906））に割り当てる。

40

【0078】

図 9 B に示されるように、表 2 の行（3）にも示される右シフト演算が、SRC 2（902）内のビットを、シフトして作成されたソース（例えば、SRC 2'（912））にシフトする。SRC 2（902）をシフトする位置の数は、表 2 の行（1）に示されるポピュレーションカウント命令によって決定される。

【0079】

図 9 C に示されるように、表 2 の行（4）にも示される否定演算が、SRC 1（904）のビットを否定して、否定の制御マスク（例えば、SRC 1'（914））を作成する。

【0080】

50

図 9 D に示されるように、表 2 の行 (5) にも示される第 2 の並列デポジット演算が、SRC 1' (914) に提供されるビットに基づいて、SRC 2' (912) のビットを第 2 の一時レジスタ (例えば、TMP 2 (916)) に割り当てる。

【0081】

図 9 E に示されるように、表 2 の行 (6) にも示される「論理和 (or)」演算が、TMP 2 (916) と TMP 1 (906) とからデスティネーションレジスタ (例えば、DEST (926)) へとビットを結合する。実施形態によれば、デスティネーションレジスタは逆分離演算の結果を含む。

[例示的なプロセッサ実装]

【0082】

図 10 は、本明細書で説明される実施形態に従ってオペレーションを実行するロジックを含むプロセッサコア 1000 のブロック図である。1つの実施形態では、インオーダーフロントエンド 1001 は、実行される命令をフェッチして、これらの命令をプロセッサパイプラインにおいて後に用いられるように用意するプロセッサコア 1000 の一部である。1つの実施形態では、フロントエンド 1001 は図 1 B のフロントエンドユニット 130 と類似しており、命令をメモリから事前にフェッチする命令プリフェッチャ 1026 を含んだコンポーネントをさらに含む。フェッチされた命令は、その命令を復号又は解釈するために、命令デコーダ 1028 に提供されてよい。

【0083】

1つの実施形態では、命令デコーダ 1028 は、受信した命令を機械が実行し得る「マイクロ命令」又は「マイクロオペレーション」(マイクロ op 又は uop と呼ばれる) と呼ばれる 1つ又は複数のオペレーションに復号する。他の実施形態では、デコーダはその命令を、1つの実施形態に従ってオペレーションを実行するマイクロアーキテクチャにより用いられるオペコード及び対応するデータ並びに制御フィールドにパースする。1つの実施形態では、トレースキャッシュ 1029 が復号された uop を取り出し、実行のためにそれらを uop キュー 1034 内のプログラム順序付きのシーケンス又はトレースにアセンブルする。

【0084】

1つの実施形態では、プロセッサコア 1000 は複合命令セットを実行する。トレースキャッシュ 1029 で複合命令が発生した場合、マイクロコード ROM 1032 がそのオペレーションを完了させるのに必要な uop を提供する。命令の中には、単一のマイクロ op に変換される命令もあれば、フルオペレーションを完了させるのにいくつかのマイクロ op を必要とする命令もある。1つの実施形態では、命令が、命令デコーダ 1028 で処理するために少数のマイクロ op に復号され得る。別の実施形態では、複数のマイクロ op がオペレーションを実現するのに必要とされる場合、命令がマイクロコード ROM 1032 内に格納され得る。例えば、1つの実施形態では、4 個より多くのマイクロ op が命令の完了に必要な場合、デコーダ 1028 は命令を実行するためにマイクロコード ROM 1032 にアクセスする。

【0085】

トレースキャッシュ 1029 は、1つの実施形態に従い 1つ又は複数の命令を完了させるマイクロコードシーケンスをマイクロコード ROM 1032 から読み出すために、正しいマイクロ命令ポインタを決定するエントリポイントプログラマブルロジックアレイ (PLA) を指す。マイクロコード ROM 1032 が命令用のマイクロ op を順番に並べ終えた後に、機械のフロントエンド 1001 は、トレースキャッシュ 1029 からのマイクロ op のフェッチを再開する。1つの実施形態では、プロセッサコア 1000 は、命令が実行のために用意されるアウトオブオーダー実行エンジン 1003 を含む。アウトオブオーダー実行ロジックは、命令が命令パイプラインを通過するときに、命令フローを並べ替えて性能を最適化するために複数のバッファを有する。マイクロコードのサポートのために構成された実施形態では、アロケータロジックが、各 uop が実行中に用いる機械バッファ及びリソースを割り当てる。さらに、レジスタリネーミングロジックが、レジスタファイル

10

20

30

40

50

の物理レジスタにおいて、論理レジスタを物理レジスタにリネームする。

【0086】

1つの実施形態では、アロケータは、メモリスケジューラ、高速スケジューラ1002、低速／汎用浮動小数点スケジューラ1004、及び簡易浮動小数点スケジューラ1006といった命令スケジューラの前段にある、1つはメモリ演算用、もう1つは非メモリ演算用となる2つのuopキューの一方に各uopのエントリを割り当てる。uopスケジューラ1002、1004、1006は、これらのスケジューラが依存する入力レジスタオペランドソースの準備状態、及びuopがそのオペレーションを完了させるのに必要な実行リソースの利用可能性に基づいて、uopがいつ実行する準備が整うかを判断する。1つの実施形態の高速スケジューラ1002は、メインクロックサイクルの各ハーフサイ

10

【0087】

レジスタファイル1008、1010が、スケジューラ1002、1004、1006と、実行ブロック1011の実行ユニット1012、1014、1016、1018、1020、1022、1024との間に位置している。1つの実施形態では、整数演算及び浮動小数点演算のために、それぞれ別個のレジスタファイル1008、1010が存在する。1つの実施形態では、各レジスタファイル1008、1010は、レジスタファイルにまだ書き込まれていない完了結果を、新たな依存uopにバイパス又は転送し得るバイ

20

【0088】

実行ブロック1011は、命令を実行する実行ユニット1012、1014、1016、1018、1020、1022、1024を含む。レジスタファイル1008、1010は、マイクロ命令が実行するのに必要な整数及び浮動小数点のデータオペランド値を格納する。1つの実施形態のプロセッサコア1000は、複数の実行ユニットで構成される。つまり、アドレス生成ユニット(AGU)1012、AGU1014、高速ALU1016、高速ALU1018、低速ALU1020、浮動小数点ALU1022、浮動小数点移動ユニット1024である。1つの実施形態では、浮動小数点実行ブロック1022、1024は、浮動小数点、MMX、SIMD、及びSSE、又は他の演算を実行する。1つの実施形態の浮動小数点ALU1022は、除算マイクロop、平方根マイクロop、及び剰余マイクロopを実行する64ビット×64ビットの浮動小数点除算器を含む。

30

【0089】

1つの実施形態では、浮動小数点値を伴う命令が、浮動小数点ハードウェアで処理されてよい。ALU演算は、高速ALU実行ユニット1016、1018が担う。1つの実施形態の高速ALU1016、1018は、クロックサイクルの半分の実効レイテンシで高速演算を実行し得る。1つの実施形態では、低速ALU1020は、乗算器、シフト、フラグロジック、分岐処理などの長レイテンシタイプの演算用の整数実行ハードウェアを含むので、最も複雑な整数演算は低速ALU1020が担う。メモリロード／ストアオペレーションは、AGU1012、1014によって実行される。1つの実施形態では、整数ALU1016、1018、1020は、64ビットデータオペランドに対して整数演算を実行するという状況で説明される。代替的な実施形態では、ALU1016、1018、1020は、16、32、128、256などを含む様々なデータビットをサポートするよう実装され得る。同様に、浮動小数点ユニット1022、1024は、様々な幅のビットを有するある範囲のオペランドをサポートするよう実装され得る。1つの実施形態で

40

50

は、浮動小数点ユニット 1022、1024 は、SIMD 及びマルチメディア命令と併せて、128 ビット幅のバックドデータオペランドを処理し得る。

【0090】

1 つの実施形態では、uop スケジューラ 1002、1004、1006 は、親ロードが実行を終了する前に、依存演算をディスパッチする。uop が投機的にスケジューリングされて実行されるので、プロセッサコア 1000 はメモリミスを処理するロジックも含む。データロードがデータキャッシュで失敗した場合、スケジューラに一時的に不正確なデータを残した依存演算がインフライトでパイプライン中に存在する可能性がある。やり直しメカニズムが、不正確なデータを用いる命令を追跡して再実行する。1 つの実施形態では、依存演算だけがやり直される必要があり、独立演算は完了することが可能である。

10

【0091】

1 つの実施形態では、メモリ実行ユニット (MEU) 1041 が含まれている。MEU 1041 は、メモリアーダバッファ (MOB) 1042、SRAM ユニット 1030、データ TLB ユニット 1072、データキャッシュユニット 1074、及び L2 キャッシュユニット 1076 を含む。

【0092】

プロセッサコア 1000 は、様々なコンポーネントを共有又は分割することで同時マルチスレッドオペレーション用に構成されてよい。プロセッサ上で動作する任意のスレッドが、共有のコンポーネントにアクセスしてよい。例えば、共有バッファ又は共有キャッシュの空き領域が、スレッド要求に関係なくオペレーションをスレッドするために割り当てられ得る。1 つの実施形態では、分割されたコンポーネントがスレッドごとに割り当てられる。具体的にどのコンポーネントが共有され、どのコンポーネントが分割されるかは、実施形態によって異なる。1 つの実施形態では、実行ユニット (例えば、実行ブロック 1011) などのプロセッサ実行リソース、及びデータキャッシュ (例えば、データ TLB ユニット 1072、データキャッシュユニット 1074) が共有リソースである。1 つの実施形態では、L2 キャッシュユニット 1076 及び他のより高いレベルのキャッシュユニット (例えば、L3 キャッシュ、L4 キャッシュ) を含むマルチレベルのキャッシュが、全ての実行スレッドの間で共有される。他のプロセッサリソースが、スレッドごとに分割されて、割り当てられ又は割り振られ、分割されたリソースの特定のパーティションが特定のスレッドに特化される。分割された例示的なリソースは、MOB 1042、(例えば、図 1B のリネーム/アロケータユニット 152 及びリタイアメントユニット 154 内の) アウトオブオーダーエンジン 1003 のレジスタエイリアステーブル (RAT) 及びリオーダーバッファ (ROB)、並びにフロントエンド 1001 の命令デコーダ 1028 に関連した 1 つ又は複数の命令復号キューを含む。

20

30

1 つの実施形態では、命令 TLB (例えば、図 1B の命令 TLB ユニット 136) 及び分岐予測ユニット (例えば、図 1B の分岐予測ユニット 132) も分割される。

【0093】

アドバンスド・コンフィグレーション・アンド・パワー・インタフェース (ACPI)仕様は、プロセッサ及び/又はチップセットによってサポートされ得る様々な「C 状態」を含む電源管理ポリシを説明している。このポリシでは、プロセッサが高電圧、高周波数で動作するランタイム状態として、C0 が定義されている。コアクロックが内部で停止する自動停止状態として、C1 が定義されている。コアクロックが外部で停止するクロック停止状態として、C2 が定義されている。全てのプロセッサクロックが停止するディープスリープ状態として C3 が定義され、全てのプロセッサクロックが停止し、且つプロセッサ電圧がより低いデータ保持ポイントに減少するディープスリープ状態として C4 が定義されている。様々な追加のディープスリープ電源状態である C5 及び C6 も、プロセッサによっては実装される。C6 状態の間、全てのスレッドが停止し、C6 状態の間、電源供給されたままの C6 用の SRAM にスレッド状態が格納され、プロセッサコアへの電圧はゼロに減少する。

40

【0094】

50

図 1 1 は、ある実施形態に従い、逆分離演算を実行するロジックを含む処理システムのブロック図である。例示的な処理システムは、メインメモリ 1 1 0 0 に結合されたプロセッサ 1 1 5 5 を含む。プロセッサ 1 1 5 5 は、逆分離命令を復号するための復号ロジック 1 1 3 1 を有する復号ユニット 1 1 3 0 を含む。さらに、プロセッサ実行エンジンユニット 1 1 4 0 は、逆分離命令を実行するための追加の実行ロジック 1 1 4 1 を含む。レジスタ 1 1 0 5 は、実行ユニット 1 1 4 0 が命令ストリームを実行するときに、オペランド、制御データ、及び他のタイプのデータ用のレジスタストレージを提供する。

【0095】

簡略化のために、単一のプロセッサコア（「コア 0」）の詳細が図 1 1 に示されている。しかし、図 1 1 に示される各コアは、コア 0 と同じロジックのセットを有してよいことが理解される。示されるように、各コアはまた、指定されたキャッシュ管理ポリシーに従って命令及びデータをキャッシュするための、専用のレベル 1（L1）キャッシュ 1 1 1 2 及びレベル 2（L2）キャッシュ 1 1 1 1 を含んでよい。L1 キャッシュ 1 1 1 1 は、命令を格納するための別個の命令キャッシュ 1 3 2 0 と、データを格納するための別個のデータキャッシュ 1 1 2 1 とを含む。様々なプロセッサキャッシュ内に格納される命令及びデータは、キャッシュラインの粒度で管理され、その粒度は固定サイズ（例えば、64 バイト、128 バイト、512 バイトの長さ）であってよい。この例示的な実施形態の各コアは、メインメモリ 1 1 0 0 から命令をフェッチするための命令フェッチユニット 1 1 1 0 及び／又は共有レベル 3（L3）キャッシュ 1 1 1 6、命令を復号するための復号ユニット 1 1 3 0、命令を実行するための実行ユニット 1 3 4 0、並びに命令をリタイアして結果をライトバックするためのライトバック／リタイアユニット 1 1 5 0 を有する。

【0096】

命令フェッチユニット 1 1 1 0 は様々な周知のコンポーネントを含み、それらのコンポーネントには、メモリ 1 1 0 0（又は複数のキャッシュのうち 1 つ）からフェッチされるべき次の命令のアドレスを格納するための次の命令ポインタ 1 1 0 3 と、アドレス変換速度を改善するために、最近用いられた仮想対物理の命令アドレスに関するマップを格納するための命令変換ルックアサイドバッファ（ITLB）1 1 0 4 と、命令分岐アドレスを投機的に予測するための分岐予測ユニット 1 1 0 2 と、分岐アドレス及びターゲットアドレスを格納するための分岐ターゲットバッファ（BTB）1 1 0 1 とが含まれる。命令がフェッチされると、その後命令は、復号ユニット 1 1 3 0、実行ユニット 1 1 4 0、及びライトバック／リタイアユニット 1 1 5 0 を含む命令パイプラインの残りのステージにストリームされる。

【0097】

図 1 2 は、ある実施形態に従い、例示的な逆分離命令を処理するロジックのフロー図である。ブロック 1 2 0 2 において、命令パイプラインは、逆分離演算を実行する命令をフェッチすることから始まる。実施形態によっては、命令は第 1 の入力オペランド、第 2 の入力オペランド、及びデスティネーションオペランドを受け入れる。そのような実施形態では、入力オペランドは、制御マスク及びソースレジスタを含む。ソースレジスタは、バックドバイト、ワード、ダブルワード、クワッドワードの値を格納する汎用レジスタ又はベクトルレジスタであってよい。制御マスクは、ソース汎用レジスタからのインターリーブを制御するのに用いられる汎用レジスタに提供されてよく、又はソースベクトルレジスタの各要素に提供されてもよい。1 つの実施形態では、制御マスクは、ソースベクトルレジスタからのインターリーブを制御するために、ベクトルレジスタを介して提供されてよい。1 つの実施形態では、デスティネーションオペランドはデスティネーションレジスタを提供し、そのレジスタは、バックドバイト、ワード、ダブルワード、又はクワッドワードの値を格納するよう構成された汎用レジスタ又はベクトルレジスタでよい。

【0098】

ブロック 1 2 0 4 において、復号ユニットが命令を復号された命令に復号する。1 つの実施形態では、復号された命令は単一のオペレーションである。1 つの実施形態では、復号された命令は、命令の各サブ要素を実行する 1 つ又は複数の論理マイクロオペレーショ

ンを含む。マイクロオペレーションは物理的に組み込まれ得る、又はマイクロコードオペレーションは、実行ユニットなどのプロセッサのコンポーネントに命令を実行する様々なオペレーションを実行させ得る。

【0099】

ブロック1206において、プロセッサの実行ユニットが、制御マスクに基づいてソースレジスタのビットをインターリーブする逆分離（例えば、「羊と山羊」の逆）演算を実行するために復号された命令を実行する。逆分離演算を実行する例示的な論理演算が図9A～図9Eに示されているが、実行される特定の演算は実施形態によって異なってよく、別の又は追加の論理が逆分離演算を実行するのに用いられてもよい。実行中に、プロセッサの1つ又は複数の実行ユニットが、制御マスクに基づいて、ソースレジスタ又はソースレジスタのベクトル要素の一方側又は反対側（例えば、左又は右）からソースデータを読み出す。1つの実施形態では、制御マスクビットの1は、レジスタの「右」側の値が取得されることを示し、制御マスクビットの0は、レジスタの「左」側の値が取得されることを示す。実施形態によれば、レジスタの「右」側及び「左」側はそれぞれ、レジスタの下位ビット及び上位ビットを示してよい。本明細書で説明されるように、上位ビット及び下位ビットは、データワードを構成するバイトがコンピュータメモリに格納される場合、これらのバイトを解釈するのに用いられる規則から独立した最上位ビット及び最下位ビットとして定義される。しかし、バイトオーダが実施形態及び構成によって異なり得るので、レジスタのそれぞれの側及びワードアドレス／オフセットに関連したバイトオーダが、様々な実施形態の範囲に違反することなく異なってよいことが理解されるであろう。

【0100】

ブロック1408において、プロセッサは実行された命令の結果をプロセッサレジスタファイルに書き込む。プロセッサレジスタファイルは、様々なデータタイプを格納する1つ又は複数の物理レジスタファイルを含み、データタイプにはスカラ整数タイプ又はパックド整数データタイプが含まれる。1つの実施形態では、レジスタファイルは、命令デスティネーションオペランドによりデスティネーションレジスタとして示される汎用レジスタ又はベクトルレジスタを含む。

[例示的な命令フォーマット]

【0101】

本明細書で説明される命令の実施形態は、異なるフォーマットに具現化されてもよい。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ている。命令の実施形態は、そのようなシステム、アーキテクチャ、及びパイプライン上で実行されてよいが、詳述されたこれらのものに限定されない。

【0102】

ベクトル対応命令フォーマットは、ベクトル命令に適した命令フォーマットである（例えば、ベクトル演算に固有の特定のフィールドがある）。ベクトル演算及びスカラ演算の両方がベクトル対応命令フォーマットを通じてサポートされる実施形態が説明されるが、代替的な実施形態は、ベクトル対応命令フォーマットを通じてサポートされるベクトル演算のみを用いる。

【0103】

図13A～図13Bは、ある実施形態に従い、汎用ベクトル対応命令フォーマット及びその命令テンプレートを示すブロック図である。図13Aは、ある実施形態に従い、汎用ベクトル対応命令フォーマット及びそのクラスA命令テンプレートを示すブロック図であり、図13Bは、ある実施形態に従い、汎用ベクトル対応命令フォーマット及びそのクラス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、汎用ベクトル対応命令フォーマット1300に対して、クラスA命令テンプレート及びクラスB命令テンプレートが定義され、その両方が非メモリアクセス1305の命令テンプレート及びメモリアクセス1320の命令テンプレートを含む。ベクトル対応命令フォーマットとの関連で汎用という用語は、いかなる特定の命令セットにも関係していない命令フォーマットを意味する。

【0104】

実施形態が説明されるが、その中でベクトル対応命令フォーマットは以下のものをサポートする。つまり、32ビット（4バイト）又は64ビット（8バイト）データ要素幅（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）（したがって、64バイトベクトルは、ダブルワードサイズの16個の要素、又は代わりにクワッドワードサイズの8個の要素から構成される）と、16ビット（2バイト）又は8ビット（1バイト）データ要素幅（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）と、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）データ要素幅（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）と、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）データ要素幅（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）である。しかし、代替的な実施形態は、より大きいデータ要素幅、より小さいデータ要素幅、又は異なるデータ要素幅（例えば、128ビット（16バイト）データ要素幅）を有する、より大きいベクトルオペランドサイズ、より小さいベクトルオペランドサイズ、及び／又は異なるベクトルオペランドサイズ（例えば、256バイトベクトルオペランド）をサポートする。

10

【0105】

図13AのクラスA命令テンプレートは以下のものを含む。つまり、1)非メモリアクセス1305の命令テンプレート内に示されている、非メモリアクセス・フルラウンド制御型オペレーション1310の命令テンプレート、及び非メモリアクセス・データ変換型オペレーション1315の命令テンプレート、並びに2)メモリアクセス1320の命令テンプレート内に示されている、メモリアクセス・一時的1325の命令テンプレート、及びメモリアクセス・非一時的1330の命令テンプレートである。図13BのクラスB命令テンプレートは以下のものを含む。つまり、1)非メモリアクセス1305の命令テンプレート内に示されている、非メモリアクセス・書き込みマスク制御・部分ラウンド制御型オペレーション1312の命令テンプレート、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1317の命令テンプレート、並びに2)メモリアクセス1320命令テンプレート内に示されている、メモリアクセス・書き込みマスク制御1327の命令テンプレートである。

20

【0106】

汎用ベクトル対応命令フォーマット1300は、図13A～図13Bに示される順で以下に列挙する次のフィールドを含む。

30

【0107】

フォーマットフィールド1340：このフィールドの特定値（命令フォーマット識別子の値）は、ベクトル対応命令フォーマットを一意に特定し、したがって、命令ストリーム内のベクトル対応命令フォーマットにおける命令の出現を特定する。そのため、このフィールドは、汎用ベクトル対応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点で、任意なものである。

【0108】

ベースオペレーションフィールド1342：このコンテンツは、異なるベースオペレーションを識別する。

40

【0109】

レジスタインデックスフィールド1344：このコンテンツは、ソース及びデスティネーションオペランドの位置を、それらがレジスタ内にあってもメモリ内にあっても、直接又はアドレス生成を通じて指定する。これらは、P x Q（例えば32 x 512、16 x 128、32 x 1024、64 x 1024）レジスタファイルからN個のレジスタを選択するのに十分な数のビットを含む。1つの実施形態では、Nは3つのソースレジスタ及び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、代替的な実施形態はより多くの又はより少ないソースレジスタ及びデスティネーションレジスタをサポートしてもよい（例えば、2つのソース（このうち1つはデスティネーションの役割も果たす）までをサポートしてよく、3つのソース（このうち1つはデスティネーションの役割も果たす）までを

50

サポートしてもよく、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もよい）。

【0110】

修飾子フィールド1346：このコンテンツは、汎用ベクトル命令フォーマットにおいてメモリアクセスを指定する命令の出現をそうでない命令の出現と識別する。すなわち、非メモリアクセス1305の命令テンプレートとメモリアクセス1320の命令テンプレートとを識別する。メモリアクセスオペレーションは、メモリ階層を読み出す、及び／又はメモリ階層へ書き込む（場合によっては、レジスタ内の値を用いてソースアドレス及び／又はデスティネーションアドレスを指定する）が、非メモリアクセスオペレーションはこうしたことを行わない（例えば、ソース及びデスティネーションはレジスタである）。1つの実施形態では、このフィールドはまた、メモリアドレス計算を実行するための3つの異なる方法から選択するが、代替的な実施形態は、メモリアドレス計算を実行するためのより多くの方法、より少ない方法、又は異なる方法をサポートしてもよい。

10

【0111】

拡大オペレーションフィールド1350：このコンテンツは、様々な異なるオペレーションのどれがベースオペレーションに加えて実行されるかを識別する。このフィールドは、コンテキスト固有のものである。1つの実施形態では、このフィールドは、クラスフィールド1368、アルファフィールド1352、及びベータフィールド1354に分割される。拡大オペレーションフィールド1350は、共通グループのオペレーションが2つ、3つ、又は4つの命令ではなく、単一の命令で実行されることを可能にする。

20

【0112】

スケールフィールド1360：このコンテンツは、メモリアドレス生成のために（例えば、 $2^{[スケール]} \times [インデックス] + [ベース]$ 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コンテンツをスケールリングすることを可能にする。

【0113】

変位フィールド1362A：このコンテンツは、（例えば、 $2^{[スケール]} \times [インデックス] + [ベース] + [変位]$ 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。

【0114】

変位係数フィールド1362B（なお、変位フィールド1362Aを変位係数フィールド1362Bのすぐ上に並置することで、一方又は他方が使用されていることが示される点に注意）：このコンテンツは、アドレス生成の一部として用いられ、これは、メモリアクセスのサイズ（N）でスケールリングされる変位係数を指定する。ここで、Nは、（例えば、 $2^{[スケール]} \times [インデックス] + [ベース] + [スケールリングされた変位]$ を用いるアドレス生成のための）メモリアクセス内のバイト数である。冗長下位ビットは無視され、したがって、有効アドレスの計算に用いられる最終的な変位を生成するために、変位係数フィールドのコンテンツはメモリオペランドの合計サイズ（N）を乗じる。Nの値は、フルオペコードフィールド1374（本明細書に後述）及びデータ操作フィールド1354Cに基づき、プロセッサハードウェアによって実行時に決定される。変位フィールド1362A及び変位係数フィールド1362Bは、これらが非メモリアクセス1305の命令テンプレートには用いられず、及び／又は異なる実施形態では2つのうち一方のみを実装するかどちらも実装しない場合があるという点で任意である。

30

40

【0115】

データ要素幅フィールド1364：このコンテンツは、（実施形態によっては全ての命令に、他の実施形態ではいくつかの命令だけに）複数のデータ要素幅のどれが用いられるべきかを識別する。このフィールドは、1つのデータ要素幅のみがサポートされる場合、及び／又は複数のデータ要素幅がオペコードの何らかの態様を用いてサポートされる場合は、必要とされないという点で任意である。

【0116】

書き込みマスクフィールド1370：このコンテンツは、データ要素位置に基づいて、

50

デスティネーションベクトルオペランドのそのデータ要素位置がベースオペレーション及び拡大オペレーションの結果を反映するかどうかを制御する。クラスA命令テンプレートは、マージ処理・書き込みマスク処理をサポートし、クラスB命令テンプレートは、マージ・書き込みマスク処理、及びゼロ設定・書き込みマスク処理の両方をサポートする。マージする場合、ベクトルマスクは、（ベースオペレーション及び拡大オペレーションによって指定される）任意のオペレーションを実行中に、デスティネーションにおける任意のセットの要素が更新から保護されることを可能とし、他の1つの実施形態では、対応するマスクビットが0である場合、デスティネーションの各要素の古い値を保護する。これに対して、ゼロにセットする場合、ベクトルマスクは、デスティネーションにおける任意のセットの要素が（ベースオペレーション及び拡大オペレーションによって指定される）任意のオペレーションの実行中にゼロにセットされることを可能とし、1つの実施形態では、対応するマスクビットの値が0である場合、デスティネーションの要素は0に設定される。この機能のサブセットは、実行されているオペレーションのベクトル長（すなわち、変更される要素の長さ、つまり最初の要素から最後の要素まで）を制御する能力である。しかし、変更される要素は連続的である必要はない。したがって、書き込みマスクフィールド1370は、ロード演算、ストア演算、算術演算、論理演算などを含む一部のベクトル演算を可能にする。書き込みマスクフィールド1370のコンテンツが用いられる書き込みマスクを含む複数の書き込みマスクレジスタのうち1つを選択する（したがって、書き込みマスクフィールド1370のコンテンツが実行されるマスク処理を間接的に特定する）実施形態が説明されるが、代替的な実施形態では代わりに又は追加的に、書き込みマスクフィールド1370のコンテンツが、実行されるマスク処理を直接指定することを可能にする。

10

20

【0117】

即値フィールド1372：このコンテンツは、即値オペランド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対応フォーマットの実装には存在せず、即値を用いない命令には存在しないという点で任意である。

【0118】

クラスフィールド1368：このコンテンツは、複数の異なるクラスの命令を識別する。図13A～図13Bに関連して、このフィールドのコンテンツは、クラスA命令及びクラスB命令から選択する。図13A～図13Bでは、角が丸い四角が、フィールド内に特定値が存在することを示すのに用いられている（例えば、図13A～図13Bにそれぞれあるクラスフィールド1368用のクラスA 1368A、及びクラスB 1368B）。

30

〔クラスAの命令テンプレート〕

【0119】

クラスAの非メモリアクセス1305命令テンプレートの場合、アルファフィールド1352はRSフィールド1352Aと解釈され、そのコンテンツは、異なる拡大オペレーションタイプのどれが実行されるべきかを識別し（例えば、非メモリアクセス・ラウンド型オペレーション1310及び非メモリアクセス・データ変換型オペレーション1315の命令テンプレートに対し、ラウンド1352A. 1及びデータ変換1352A. 2がそれぞれ指定される）、ベータフィールド1354は、指定されるタイプのオペレーションのどれが実行されるべきかを識別する。非メモリアクセス1305の命令テンプレートには、スケールフィールド1360、変位フィールド1362A、及び変位係数フィールド1362Bが存在しない。

40

〔非メモリアクセス命令テンプレートーフルラウンド制御型オペレーション〕

【0120】

非メモリアクセスフルラウンド制御型オペレーション1310の命令テンプレートにおいて、ベータフィールド1354はラウンド制御フィールド1354Aと解釈され、そのコンテンツは静的なラウンド処理を提供する。説明された実施形態では、ラウンド制御フィールド1354Aは、全浮動小数点例外抑制（SAE）フィールド1356及びラウン

50

ド演算制御フィールド1358を含むが、代替的な実施形態では、これらのコンセプトを両方ともサポートしてよく、それらを同じフィールド内に符号化してよく、あるいはこれらのコンセプト／フィールドの一方又は他方のみを有してもよい（例えば、ラウンド演算制御フィールド1358のみを有してよい）。

【0121】

SAEフィールド1356：このコンテンツは、例外イベント報告を無効化するかどうか識別する。SAEフィールド1356のコンテンツが、抑制が可能であることを示す場合、所与の命令は、いかなる種類の浮動小数点例外フラグも報告せず、いかなる浮動小数点例外ハンドラも呼び出さない。

【0122】

ラウンド演算制御フィールド1358：このコンテンツは、ラウンド演算のグループのどれを実行すべきかを識別する（例えば、切り上げ、切り捨て、0への丸め、及び最近接丸め）。したがって、ラウンド演算制御フィールド1358は、命令に基づいてラウンドモードの変更を可能にする。1つの実施形態では、プロセッサが、ラウンドモードを指定する制御レジスタを含み、ラウンド演算制御フィールド1350のコンテンツは、当該レジスタの値をオーバーライドする。

[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]

【0123】

非メモリアクセスデータ変換型オペレーション1315の命令テンプレートでは、ベータフィールド1354はデータ変換フィールド1354Bとして解釈され、そのコンテンツは、複数のデータ変換のどれが実行されるべきかを識別する（例えば、データ変換なし、スウィズル、ブロードキャスト）。

【0124】

クラスAのメモリアクセス1320の命令テンプレートの場合、アルファフィールド1352はエビクションヒントフィールド1352Bと解釈され、そのコンテンツは、エビクションヒントのどれが用いられるべきかを識別する（図13Aにおいて、一時的1352B.1及び非一時的1352B.2はそれぞれ、メモリアクセス・一時的1325の命令テンプレート及びメモリアクセス・非一時的1330の命令テンプレートに指定される）。ベータフィールド1354はデータ操作フィールド1354Cと解釈され、そのコンテンツは、（プリミティブとしても知られる）複数のデータ操作オペレーションのどれが実行されるべきかを識別する（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ジョン、デスティネーションのダウンコンバージョン）。メモリアクセス1320の命令テンプレートはスケールフィールド1360を含み、任意で変位フィールド1362A又は変位係数フィールド1362Bを含む。

【0125】

ベクトルメモリ命令は、変換サポートを用いて、メモリからのベクトルロード及びメモリへ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、ベクトルメモリ命令はデータ要素単位の形式でデータをメモリから転送し、データをメモリに転送する。実際に転送される要素は、書き込みマスクとして選択されるベクトルマスクのコンテンツによって指示される。

[メモリアクセス命令テンプレート—一時的]

【0126】

一時的データは、すぐに再使用されてキャッシュによる利益を享受するのに十分である可能性の高いデータである。しかし、これはヒントであり、異なるプロセッサが異なる方法でヒントを実行してよく、その方法には、ヒントを完全に無視することも含まれる。

[メモリアクセス命令テンプレート—非一時的]

【0127】

非一時的データは、すぐに再使用されてレベル1キャッシュにキャッシュすることから利益を享受するのに十分である可能性が低いデータであり、エビクションが優先されなければならない。しかし、これはヒントであり、異なるプロセッサが異なる方法でヒントを

10

20

30

40

50

実行してよく、その方法には、ヒントを完全に無視することも含まれる。

〔クラスBの命令テンプレート〕

【0128】

クラスBの命令テンプレートの場合には、アルファフィールド1352は書き込みマスク制御（Z）フィールド1352Cと解釈され、そのコンテンツは、書き込みマスクフィールド1370によって制御される書き込みマスク処理がマージ処理であるべきか、ゼロ設定処理であるべきかを識別する。

【0129】

クラスBの非メモリアクセス1305の命令テンプレートの場合、ベータフィールド1354の一部はRLフィールド1357Aと解釈され、そのコンテンツは、異なる拡大オペレーションタイプのどれが実行されるべきかを識別し（例えば、非メモリアクセス・書き込みマスク制御・部分ラウンド制御型オペレーション1312の命令テンプレート、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1317の命令テンプレートに対し、ラウンド1357A．1及びベクトル長（VSIZE）1357A．2がそれぞれ指定される）、ベータフィールド1354の残りは、指定されるタイプのオペレーションのどれが実行されるべきかを識別する。非メモリアクセス1305の命令テンプレートには、スケールフィールド1360、変位フィールド1362A、及び変位係数フィールド1362Bが存在しない。

【0130】

非メモリアクセス・書き込みマスク制御・部分ラウンド制御型オペレーション1312の命令テンプレートでは、ベータフィールド1354の残りのものはラウンド演算フィールド1359Aと解釈され、例外イベント報告は無効にされる（所与の命令は、いかなる種類の浮動小数点例外フラグも報告せず、いかなる浮動小数点例外ハンドラも呼び出さない）。

【0131】

ラウンド演算制御フィールド1359A：ラウンド演算制御フィールド1358と全く同じように、このコンテンツは、ラウンド演算のグループのどれを実行すべきかを識別する（例えば、切り上げ、切り捨て、0への丸め、及び最近接丸め）。したがって、ラウンド演算制御フィールド1359Aは、命令に基づいてラウンドモードの変更を可能にする。1つの実施形態では、プロセッサが、ラウンドモードを指定する制御レジスタを含み、ラウンド演算制御フィールド1350のコンテンツは、当該レジスタの値をオーバーライドする。

【0132】

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1317の命令テンプレートでは、ベータフィールド1354の残りのものはベクトル長フィールド1359Bと解釈され、そのコンテンツは、複数のデータベクトル長のどれが実行されるべきかを識別する（例えば、128バイト、256バイト、又は512バイト）。

【0133】

クラスBのメモリアクセス1320の命令テンプレートの場合には、ベータフィールド1354の一部はブロードキャストフィールド1357Bと解釈され、そのコンテンツは、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どうかを識別し、ベータフィールド1354の残りはベクトル長フィールド1359Bと解釈される。メモリアクセス1320の命令テンプレートはスケールフィールド1360を含み、任意で変位フィールド1362A又は変位係数フィールド1362Bを含む。

【0134】

汎用ベクトル対応命令フォーマット1300に関して、フォーマットフィールド1340、ベースオペレーションフィールド1342、及びデータ要素幅フィールド1364を含むフルオペコードフィールド1374が示されている。フルオペコードフィールド1374がこれらのフィールド全てを含む1つの実施形態が示されているが、これらを全てサポートしない実施形態では、フルオペコードフィールド1374は、これら全てのフィー

10

20

30

40

50

ルドより少ないフィールドを含む。フルオペコードフィールド 1 3 7 4 は、オペレーションコード（オペコード）を提供する。

【0 1 3 5】

拡大オペレーションフィールド 1 3 5 0、データ要素幅フィールド 1 3 6 4、及び書き込みマスクフィールド 1 3 7 0 は、これらの機能が汎用ベクトル対応命令フォーマットの命令に基づいて指定されることを可能にする。

【0 1 3 6】

書き込みマスクフィールドとデータ要素幅フィールドの組み合わせは、それらが異なるデータ要素幅に基づいてマスクが適用されることを可能にするという点で、型付き命令を形成する。

【0 1 3 7】

クラス A 及びクラス B 内で見られる様々な命令テンプレートは、異なる状況において有益である。実施形態によっては、異なるプロセッサ又はプロセッサ内の異なるコアが、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、汎用計算を対象とした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、グラフィックス及び／又は科学的（スループット）計算を主に対象としたコアは、クラス A のみをサポートしてよく、両方を対象としたコアは、両方をサポートしてよい（もちろん、コアは、両方のクラスのテンプレート及び命令の何らかの組み合わせを有するが、両方のクラスの全てのテンプレート及び命令が本発明の範囲内にあるわけではない）。また、単一のプロセッサは複数のコアを含んでよく、その全てが同じクラスをサポートし、又はその異なるコアが異なるクラスをサポートする。例えば、別個のグラフィックス及び汎用コアを有するプロセッサにおいて、グラフィックス及び／又は科学計算を主に対象とする複数のグラフィックスコアのうち 1 つがクラス A のみをサポートしてよく、複数の汎用コアのうち 1 つ又は複数が、クラス B のみをサポートする汎用計算を対象とし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もよい。別個のグラフィックスコアを持たない別のプロセッサは、クラス A 及びクラス B の両方をサポートするもう 1 つの汎用インオーダー又はアウトオブオーダーコアを含んでよい。もちろん、一方のクラスの特徴はまた、異なる実施形態において他方のクラスに実装されてよい。高水準言語で書かれたプログラムは、以下の形式を含む様々な異なる実行可能形式に変換される（例えば、ジャスト・イン・タイム方式でコンパイルされる、又は静的にコンパイルされる）であろう。例えば、1）実行用ターゲットプロセッサによってサポートされるクラスの命令のみを有する形式、あるいは 2）全クラスの命令の異なる組み合わせを用いて書かれた代替ルーチンを有し、プロセッサによってサポートされる命令に基づいて、実行するルーチンを選択する制御フローコードを有する形式であって、当該プロセッサが当該コードを現時点で実行している、形式である。

〔例示的な特定ベクトル対応命令フォーマット〕

【0 1 3 8】

図 1 4 A～図 1 4 D は、ある実施形態に従って例示的な特定ベクトル対応命令フォーマットを示すブロック図である。図 1 4 A は、特定ベクトル対応命令フォーマット 1 4 0 0 を示し、これは位置、サイズ、解釈、及びフィールドの順序、並びにこれらのフィールドのいくつかに対する値を指定するという点で特定のものである。特定ベクトル対応命令フォーマット 1 4 0 0 は、x 8 6 命令セットを拡張するのに用いられてよく、したがって、フィールドのいくつかは、既存の x 8 6 命令セット及びその拡張版（例えば、AVX）に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、拡張された既存の x 8 6 命令セットのプリフィッ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、及び即値フィールドと一致した状態のままである。図 1 4 A のフィールドがマッピングされる図 1 3 A～図 1 3 B のフィールドが示されている。

【0 1 3 9】

実施形態は、例示を目的として汎用ベクトル対応命令フォーマット 1 3 0 0 との関連で

10

20

30

40

50

特定ベクトル対応命令フォーマット1400に関連して説明されるが、本発明は、特許請求される場合を除いて、特定ベクトル対応命令フォーマット1400に限定されないことが理解されるべきである。例えば、汎用ベクトル対応命令フォーマット1300では、様々なフィールドについて様々な可能なサイズを検討するが、特定ベクトル対応命令フォーマット1400は、特定のサイズのフィールドを有するものとして示されている。具体例として、データ要素幅フィールド1364が、特定ベクトル対応命令フォーマット1400内の1ビットフィールドとして示されているが、本発明はそのように限定されてはいない（すなわち、汎用ベクトル対応命令フォーマット1300では、他のサイズのデータ要素幅フィールド1364を検討する）。

【0140】

汎用ベクトル対応命令フォーマット1300は、図14Aに示される順で以下に列挙される次のフィールドを含む。

【0141】

EVEXPリフィックス（バイト0-3）1402：4バイト形式で符号化される。

【0142】

フォーマットフィールド1340（EVEXPバイト0、ビット[7:0]：1番目のバイト（EVEXPバイト0）はフォーマットフィールド1340であり、ここには0x62（本発明の1つの実施形態において、ベクトル対応命令フォーマットを識別するのに用いられる固有値）が入っている。

【0143】

2～4番目のバイト（EVEXPバイト1-3）は、特定の機能を提供する複数のビットフィールドを含む。

【0144】

REXフィールド1405（EVEXPバイト1、ビット[7-5]）：EVEXP.Rビットフィールド（EVEXPバイト1、ビット[7]-R）、EVEXP.Xビットフィールド（EVEXPバイト1、ビット[6]-X）、及びEVEXP.Bビットフィールド（EVEXPバイト1、ビット[5]-B）から構成される。EVEXP.Rビットフィールド、EVEXP.Xビットフィールド、及びEVEXP.Bビットフィールドは、対応するVEXPビットフィールドと同じ機能を提供し、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知られているように、命令の他のフィールドは、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、EVEXP.R、EVEXP.X、及びEVEXP.Bを加えることで、Rrrr、Xxxx、Bbbbが形成され得る。

【0145】

REX'フィールド1310：これはREX'フィールド1310の1番目の部分であり、拡張された32個のレジスタセットの上位16又は下位16を符号化するのに用いられるEVEXP.R'ビットフィールド（EVEXPバイト1、ビット[4]-R'）である。1つの実施形態では、このビットは、以下に示されるように他のビットと共にビット反転フォーマットで格納され、（周知のx86の32ビットモードにおいて）BOUND命令と識別する。BOUND命令の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。代替的な実施形態は、このビット及び他の以下に示されるビットを反転フォーマットで格納しない。1の値が、下位16個のレジスタを符号化するのに用いられる。換言すると、EVEXP.R'、EVEXP.R、及び他のフィールドの他のRRRを組み合わせることで、R'Rrrrが形成される。

【0146】

オペコードマップフィールド1415（EVEXPバイト1、ビット[3:0]-mmm）：このコンテンツは、暗黙の先頭オペコードバイト（0F、0F38、又は0F3）を符号化する。

【0147】

データ要素幅フィールド 1 3 6 4 (E V E X バイト 2、ビット [7] - W) : E V E X . W という表記によって表される。E V E X . W は、データタイプ (3 2 ビットデータ要素又は 6 4 ビットデータ要素) の粒度 (サイズ) を定義するのに用いられる。

【0 1 4 8】

E V E X . v v v v 1 4 2 0 (E V E X バイト 2、ビット [6 : 3] - v v v v) : E V E X . v v v v の役割は以下のことを含み得る。1) E V E X . v v v v は、第 1 のソースレジスタオペランドを符号化し、反転 (1 の補数) 形式で指定され、2 又はそれより多くのソースオペランドを有する命令に有効である。2) E V E X . v v v v は、デステイネーションレジスタオペランドを符号化し、特定のベクトルシフトについて 1 の補数形式で指定される。又は、3) E V E X . v v v v はいかなるオペランドも符号化せず、フィールドは保留され 1 1 1 1 b を含むことになる。したがって、E V E X . v v v v フィールド 1 4 2 0 は、反転 (1 の補数) 形式で格納される第 1 のソースレジスタ指定子の下位ビット 4 つを符号化する。命令に応じて、追加の異なる E V E X ビットフィールドが、指定子サイズを 3 2 個のレジスタに拡張するのに用いられる。

【0 1 4 9】

E V E X . U クラスフィールド 1 3 6 8 (E V E X バイト 2、ビット [2] - U) : E V E X . U = 0 の場合にクラス A 又は E V E X . U 0 を示し、E V E X . U = 1 の場合にクラス B 又は E V E X . U 1 を示す。

【0 1 5 0】

プリフィックス符号化フィールド 1 4 2 5 (E V E X バイト 2、ビット [1 : 0] - p p) : ベースオペレーションフィールドに追加のビットを提供する。E V E X プリフィックスフォーマットのレガシ S S E 命令にサポートを提供することに加え、S I M D プリフィックスを圧縮するという利点も有する (S I M D プリフィックスを示すのに 1 バイトを必要とするのではなく、E V E X プリフィックスは 2 ビットしか必要としない)。1 つの実施形態では、レガシフォーマット及び E V E X プリフィックスフォーマットの両方で S I M D プリフィックス (6 6 H、F 2 H、F 3 H) を用いるレガシ S S E 命令をサポートすべく、これらのレガシ S I M D プリフィックスは S I M D プリフィックス符号化フィールドに符号化され、実行時には、デコーダの P L A に提供される前にレガシ S I M D プリフィックスに拡張される (そのため、P L A は、これらのレガシ命令のレガシフォーマットと E V E X フォーマットとの両方を変更せずに実行し得る)。より新たな命令が E V E X プリフィックス符号化フィールドのコンテンツをオペコード拡張として直接用いる場合があるが、特定の実施形態は一貫性のために同様の形式で拡張しても、これらのレガシ S I M D プリフィックスによって指定される異なる目的を可能とする。代替的な実施形態は、2 ビット S I M D プリフィックス符号化をサポートするように P L A を再設計してよく、したがって拡張を必要としない。

【0 1 5 1】

アルファフィールド 1 3 5 2 (E V E X バイト 3、ビット [7] - E H、E V E X . E H、E V E X . r s、E V E X . R L、E V E X . w r i t e m a s k c o n t r o l、及び E V E X . N としても知られ、α でも示される) : 前述したように、このフィールドはコンテキスト固有である。

【0 1 5 2】

ベータフィールド 1 3 5 4 (E V E X バイト 3、ビット [6 : 4] - S S S、E V E X . s₂₋₀、E V E X . r₂₋₀、E V E X . r r 1、E V E X . L L 0、E V E X . L L B としても知られ、β β β でも示される) : 前述したように、このフィールドはコンテキスト固有である。

【0 1 5 3】

R E X ' フィールド 1 3 1 0 : これは、R E X ' フィールドの残りであり、拡張された 3 2 個のレジスタセットの上位 1 6 又は下位 1 6 を符号化するのに用いられ得る E V E X . V ' ビットフィールド (E V E X 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格納される。1 の値が、下位 1 6 個のレジスタを符号化す

10

20

30

40

50

るのに用いられる。換言すると、 $V'VVVV$ は、 $EVE X.V'$ 、 $EVE X.vvvv$ を組み合わせることで形成される。

【0154】

書き込みマスクフィールド1370 ($EVE X$ バイト3、ビット[2:0]— kkk) : このコンテンツは、前述したように、書き込みマスクレジスタにおいてレジスタのインデックスを指定する。1つの実施形態では、特定値 $EVE X.kkk=000$ は、いかなる書き込みマスクも特定の命令に用いられないことを示唆する特別な挙動を有する（これは、全て1に物理的に組み込まれた書き込みマスクの使用、又はマスキングハードウェアをバイパスするハードウェアの使用を含む様々な方法で実装され得る）。

【0155】

リアルオペコードフィールド1430 (バイト4) はまた、オペコードバイトとしても知られている。そのオペコードの一部はこのフィールドに指定されている。

【0156】

$MOD R/M$ フィールド1440 (バイト5) は、 MOD フィールド1442、 Reg フィールド1444、及び R/M フィールド1446を含む。前述したように、 MOD フィールド1442のコンテンツは、メモリアクセスオペレーションと非メモリアクセスオペレーションとを識別する。 Reg フィールド1444の役割は、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を符号化すること、あるいはオペコード拡張として扱われ、いかなる命令オペランドを符号化するのにも用いられないこと、という2つの状況に要約され得る。 R/M フィールド1446の役割は、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、あるい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を符号化することを含んでよい。

【0157】

スケール・インデックス・ベース (SIB) バイト (バイト6) : 前述したように、スケールフィールド1350のコンテンツは、メモリアドレス生成に用いられる。 $SIB.xxx1454$ 及び $SIB.bbb1456$: これらのフィールドのコンテンツは、レジスタインデックス xxx 及び $Bbbb$ に関して前述されている。

【0158】

変位フィールド1362A (バイト7—10) : MOD フィールド1442に10が入っている場合、バイト7—10は変位フィールド1362Aであり、これは、レガシ32ビット変位 ($disp32$) と同じように機能し、バイト粒度で機能する。

【0159】

変位係数フィールド1362B (バイト7) : MOD フィールド1442に01が入っている場合、バイト7は変位係数フィールド1362Bである。このフィールドの位置は、バイト粒度で機能するレガシx86命令セットの8ビット変位 ($disp8$) のものと同じである。 $disp8$ は符号拡張されているので、−128と127バイトとの間のオフセットをアドレス指定できるだけであり、64バイトキャッシュラインに関しては、 $disp8$ は本当に有用な4つの値−128、−64、0及び64にだけ設定され得る8ビットを用いる。より広い範囲が必要となることが多いので $disp32$ が用いられるが、 $disp32$ は4バイトを必要とする。 $disp8$ 及び $disp32$ と対照的に、変位係数フィールド1362Bは $disp8$ を再解釈したものであり、変位係数フィールド1362Bを用いる場合、実際の変位は、メモリオペランドアクセスのサイズ (N) を乗じた変位係数フィールドのコンテンツによって決定される。このタイプの変位は、 $disp8 \times N$ と呼ばれる。これにより、平均命令長 (変位のために用いられる単一のバイトであるが、はるかに広い範囲を有する) が減少する。そのような圧縮された変位は、有効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、したがって、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、変位係数フィールド1362Bは、レガシx86命令セットの8ビット変位を代用する。したがって、変位係数フィールド1362Bは、 $disp8$ が $disp8 \times N$ にオーバーロードされることを唯一の例外として、x86命令セットの8ビット変位と同じように符号化される (その

10

20

30

40

50

ため、ModRM/SIB符号化ルールに変更はない)。換言すると、符号化ルール又は符号化長に変更はなく、ハードウェアによる変位値の解釈にだけ変更がある(これにより、バイト単位のアドレスオフセットを取得するために、メモリオペランドのサイズによって変位をスケールリングすることが必要となる)。

【0160】

即値フィールド1372は、前述したように動作する。

[フルオペコードフィールド]

【0161】

図14Bは、1つの実施形態に従って、フルオペコードフィールド1374を構成する特定ベクトル対応命令フォーマット1400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1374は、フォーマットフィールド1340、ベースオペレーションフィールド1342、及びデータ要素幅(W)フィールド1364を含む。ベースオペレーションフィールド1342は、プリフィックス符号化フィールド1425、オペコードマップフィールド1415、及びリアルオペコードフィールド1430を含む。

[レジスタインデックスフィールド]

【0162】

図14Cは、1つの実施形態に従って、レジスタインデックスフィールド1344を構成する特定ベクトル対応命令フォーマット1400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1344は、REXフィールド1405、REX'フィールド1410、MODR/M.regフィールド1444、MODR/M.r/mフィールド1446、VVVVフィールド1420、xxxフィールド1454、及びbbbフィールド1456を含む。

[拡大オペレーションフィールド]

【0163】

図14Dは、1つの実施形態に従って、拡大オペレーションフィールド1350を構成する特定ベクトル対応命令フォーマット1400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1368に0が入っている場合、これはEVEX.U0(クラスA1368A)を意味し、1が入っている場合には、EVEX.U1(クラスB1368B)を意味する。U=0、且つMODフィールド1442に11が入っている場合(非メモリアクセスオペレーションを意味する)、アルファフィールド1352(EVEXバイト3、ビット[7]-EH)はrsフィールド1352Aと解釈される。rsフィールド1352Aに1が入っている場合(ラウンド1352A.1)、ベータフィールド1354(EVEXバイト3、ビット[6:4]-SSS)はラウンド制御フィールド1354Aと解釈される。ラウンド制御フィールド1354Aは、1ビットのSAEフィールド1356及び2ビットのラウンド演算フィールド1358を含む。rsフィールド1352Aに0が入っている場合(データ変換1352A.2)、ベータフィールド1354(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1354Bと解釈される。U=0、且つMODフィールド1442に00、01、又は10が入っている場合(メモリアクセスオペレーションを意味する)、アルファフィールド1352(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1352Bと解釈され、ベータフィールド1354(EVEXバイト3、ビット[6:4]-SSS)は3ビットのデータ操作フィールド1354Cと解釈される。

【0164】

U=1の場合、アルファフィールド1352(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1352Cと解釈される。U=1、且つMODフィールド1442に11が入っている場合(非メモリアクセスオペレーションを意味する)、ベータフィールド1354(EVEXバイト3、ビット[4]-S₀)の一部はRLフィールド1357Aと解釈され、1(ラウンド1357A.1)が入っている場合には、ベータフィールド1354(EVEXバイト3、ビット[6-5]-S₂₋₁)の残

10

20

30

40

50

りはラウンド演算フィールド1359Aと解釈される。RLフィールド1357Aに0（VSIZE1357.A2）が入っている場合、ベータフィールド1354（EVEXバイト3、ビット[6-5]-S₂₋₁）の残りはベクトル長フィールド1359B（EVEXバイト3、ビット[6-5]-L₁₋₀）と解釈される。U=1、且つMODフィールド1442に00、01、又は10が入っている場合（メモリアクセスオペレーションを意味する）、ベータフィールド1354（EVEXバイト3、ビット[6:4]-SSS）は、ベクトル長フィールド1359B（EVEXバイト3、ビット[6-5]-L₁₋₀）及びブロードキャストフィールド1357B（EVEXバイト3、ビット[4]-B）と解釈される。

〔例示的なレジスタアーキテクチャ〕

【0165】

図15は、1つの実施形態によるレジスタアーキテクチャ1500のブロック図である。示される実施形態には、512ビット幅の32個のベクトルレジスタ1510があり、これらのレジスタは、zmm0～zmm31と参照符号が付けられている。下位16個のzmmレジスタの下位256ビットは、レジスタymm0～15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にオーバーレイされる。特定ベクトル対応命令フォーマット1400は、以下の表3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を処理する。

【表3】

レジスタファイル

可調整ベクトル長	クラス	オペレーション	レジスタ
ベクトル長フィールド1359Bを含まない命令テンプレート	A(図13A、U=0)	1310, 1315, 1325, 1330	zmmレジスタ (ベクトル長は64バイト)
	B(図13B、U=1)	1312	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1359Bを含む命令テンプレート	B(図13B、U=1)	1317, 1327	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)、 ベクトル長フィールド 1359Bに依存

【0166】

換言すると、ベクトル長フィールド1359Bは、最大長さとして1つ又は複数の他のより短い長さとの間から選択し、このようなより短い長さはそれぞれ、前述の長さの半分の長さであり、ベクトル長フィールド1359Bを用いない命令テンプレートは、最大ベクトル長を処理する。さらに1つの実施形態では、特定ベクトル対応命令フォーマット1400のクラスB命令テンプレートは、パックド又はスカラ単精度／倍精度浮動小数点データ、及びパックド又はスカラ整数データを処理する。スカラ演算は、zmm／ymm／xmmレジスタ内の最下位データ要素位置において実行される演算であり、上位のデータ要素位置は、実施形態に応じて、命令の前と同じ状態のままにされるか又はゼロにセットされ

る。

【0167】

書き込みマスクレジスタ1515：示される実施形態には、8個の書き込みマスクレジスタ（k0～k7）があり、それぞれのサイズは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1515のサイズは16ビットである。前述したように、1つの実施形態では、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ず、k0を標準的に示すであろう符号化が書き込みマスクに用いられる場合、これは、物理的に組み込まれた0xFFFFという書き込みマスクを選択し、当該命令用の書き込みマスクを効果的に無効にする。

【0168】

汎用レジスタ1525：示される実施形態には、メモリオペランドをアドレス指定する既存のx86アドレッシング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。これらのレジスタには、RAX、RBX、RCX、RDX、RBP、RSI、RDI、RSP、及びR8～R15という名称で参照符号が付けられている。

【0169】

MMXパックド整数フラットレジスタファイル1550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（x87スタック）1545：示される実施形態において、x87スタックは、x87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演算を実行するのに用いられる8要素スタックである。一方、MMXレジスタは、64ビットパックド整数データに対して演算を実行するのに用いられ、さらにMMXとXMMレジスタとの間で実行される一部の演算用にオペランドを保持するのに用いられる。

【0170】

代替的な実施形態は、より広いレジスタを用いても、又はより狭いレジスタを用いてもよい。さらに、代替的な実施形態は、より多くのレジスタファイル、より少ないレジスタファイル、又は異なるレジスタファイル及びレジスタを用いてもよい。

【0171】

本明細書で説明されるのは、システムに動作を実行させるためにソフトウェア、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システム上にインストールすることにより、特定のオペレーション又は動作を実行するよう構成され得る1つ又は複数のコンピュータのシステムである。さらに、1つ又は複数のコンピュータプログラムは、処理装置により実行又は利用された場合に、本明細書で説明された動作を装置に実行させる命令又はハードウェアロジックを含めることにより、特定のオペレーション又は動作を実行するよう構成され得る。1つの実施形態では、処理装置は、第1の命令を第1のオペランド及び第2オペランドを含んだ第1の復号された命令に復号する復号ロジックと、逆分離演算を実行するために第1の復号された命令を実行する実行ユニットとを含む。

【0172】

逆分離命令は、第2のオペランドにより指定されるソースレジスタの両領域のビットを、第1のオペランドにより示される制御マスクに基づいてインターリーブする。1つの実施形態では、第2のオペランドは、それがアーキテクチャレジスタを示す限り、ソースレジスタを指定し、これは、ソースデータ又はソースデータ要素を格納する汎用レジスタ又はベクトルレジスタであってよい。第1のオペランドは、それがアーキテクチャレジスタをリストに加える限り、制御マスクを示し、又は1つの実施形態では即値オペランドとして制御マスク値を直接示してよく、又は制御マスクを含んだメモリアドレスを含んでもよい。他の実施形態は、対応するコンピュータシステム、装置、及び1つ又は複数のコンピュータストレージデバイスに記録されるコンピュータプログラムを含み、それぞれは、本明細書で指定された動作を実行するよう構成される。

【0173】

例えば、1つの実施形態では、処理装置は第1の命令をフェッチする命令フェッチユニットをさらに含み、この命令は単一の機械レベル命令である。1つの実施形態では、処理

10

20

30

40

50

装置は、本明細書で説明される逆分離演算の結果をデスティネーションオペランドにより指定される位置にコミットするレジスタファイルをさらに含み、これは汎用レジスタでもベクトルレジスタでもよい。レジスタファイルユニットは、第1のソースオペランド値を格納する第1のレジスタと、第2のソースオペランド値を格納する第2のレジスタと、前述の分離演算の結果の少なくとも1つのデータ要素を格納する第3のレジスタとを含む物理レジスタのセットを格納するよう構成され得る。

【0174】

1つの実施形態では、第1のレジスタは制御マスクを格納し、制御マスクは複数のビットを含み、制御マスクの各ビットは、値を読み出すためのソースレジスタ内のビット位置を示す。1つの実施形態では、制御マスクビットの1は、第2のレジスタの第1の領域の値が取得されることを示し、制御マスクビットの0は、第2のレジスタの第2の領域の値が取得されることを示す。

10

【0175】

1つの実施形態では、第2のレジスタの第1の領域は、当該レジスタの下位のバイトオーダービットを含み、第2のレジスタの第2の領域は、当該レジスタの上位のバイトオーダービットを含む。1つの実施形態では、第1の領域のより下位のバイトオーダービットは、レジスタの「右」側に分類され、第2の領域の上位のバイトオーダービットはレジスタの「左」側に分類される。しかし、逆分離演算は、レジスタに関連したバイトオーダー又はアドレス規則に関して限定することなく、レジスタの両側、又はベクトルレジスタの場合には複数のベクトル要素を処理するよう構成され得ることが理解されるであろう。

20

【0176】

1つの実施形態では、本明細書で説明される命令は、特定のオペレーションを実行するよう構成された、又は予め定められ機能を有する特定用途向け集積回路（ASIC）など、特定の構成のハードウェアを指す。典型的には、そのような電子デバイスは、1つ又は複数の他のコンポーネントに結合された1つ又は複数のプロセッサのセットを含む。そのようなコンポーネントには、1つ又は複数のストレージデバイス（非一時的機械可読記憶媒体）、ユーザ入力／出力デバイス（例えば、キーボード、タッチスクリーン、及び／又はディスプレイ）、及びネットワーク接続などがある。典型的には、プロセッサのセットと他のコンポーネントとの結合は、1つ又は複数のバス及びブリッジ（バスコントローラとも呼ばれる）を経由する。ストレージデバイス及びネットワークトラフィックを搬送する信号はそれぞれ、1つ又は複数の機械可読記憶媒体及び機械可読通信媒体を表す。したがって典型的には、所与の電子デバイスのストレージデバイスは、その電子デバイスの1つ又は複数のプロセッサのセット上で実行するためのコード及び／又はデータを格納する。

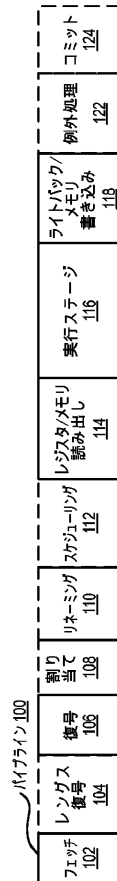
30

【0177】

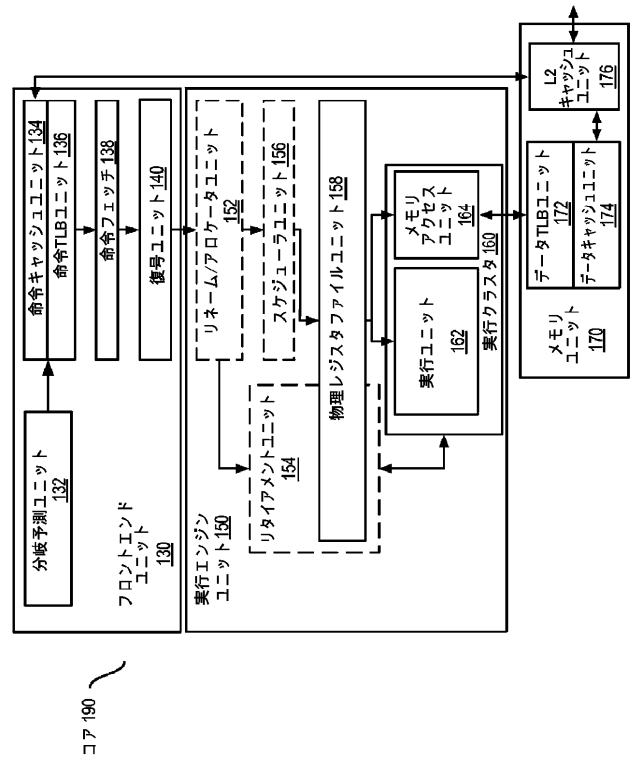
上述の明細書では、本発明がその特定の例示的な実施形態を参照して説明されている。しかし、それに対して、添付の特許請求の範囲に明記されている本発明の大局的な意図及び範囲から逸脱することなく、様々な修正及び変更が行われてよいことが明らかであろう。場合によっては、本発明の主題をあいまいにしないために、周知の構造及び機能はことさら詳細に説明されていない。したがって、本明細書及び図面は限定的な意味ではなく例示的な意味で考えられるべきである。したがって、本発明の範囲及び意図は、続く特許請求の範囲によって判断されるべきである。

40

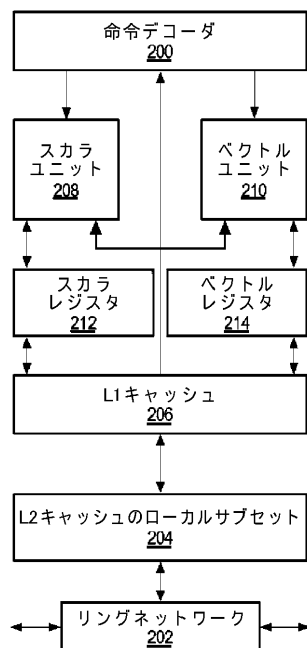
【図 1 A】



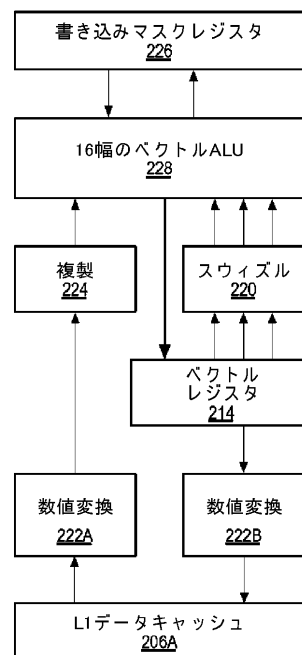
【図 1 B】



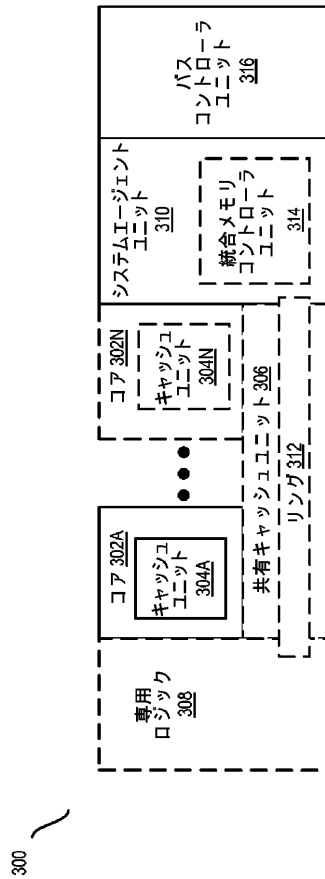
【図 2 A】



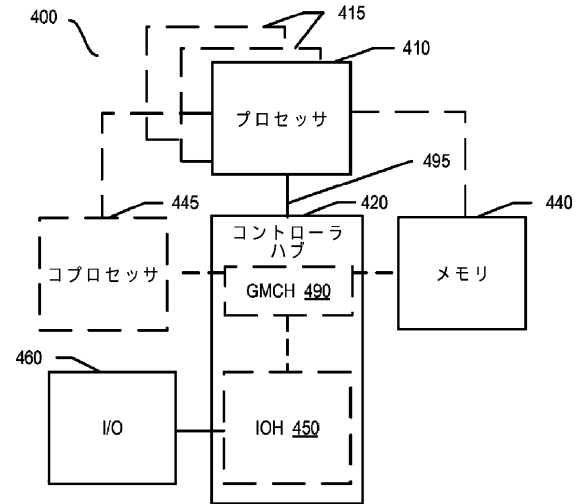
【図 2 B】



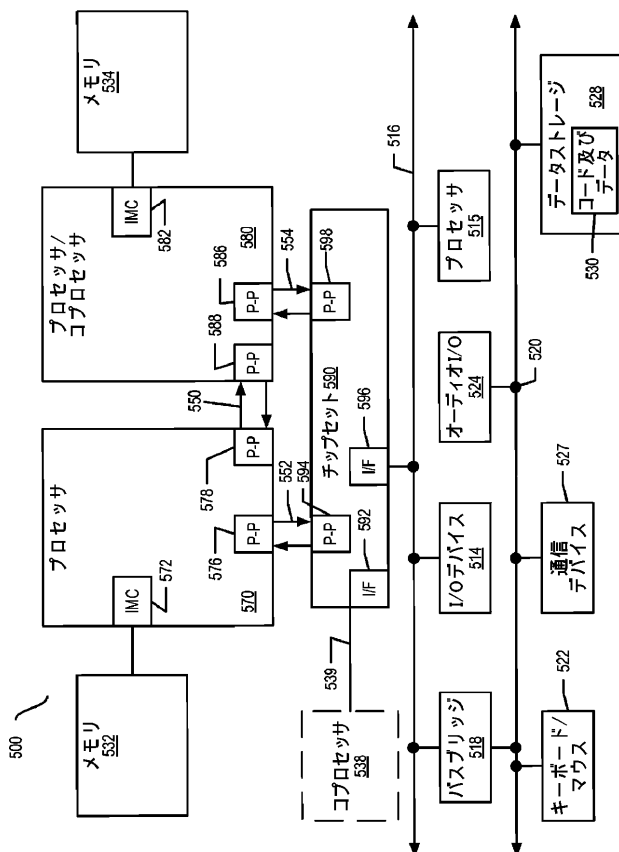
【 図 3 】



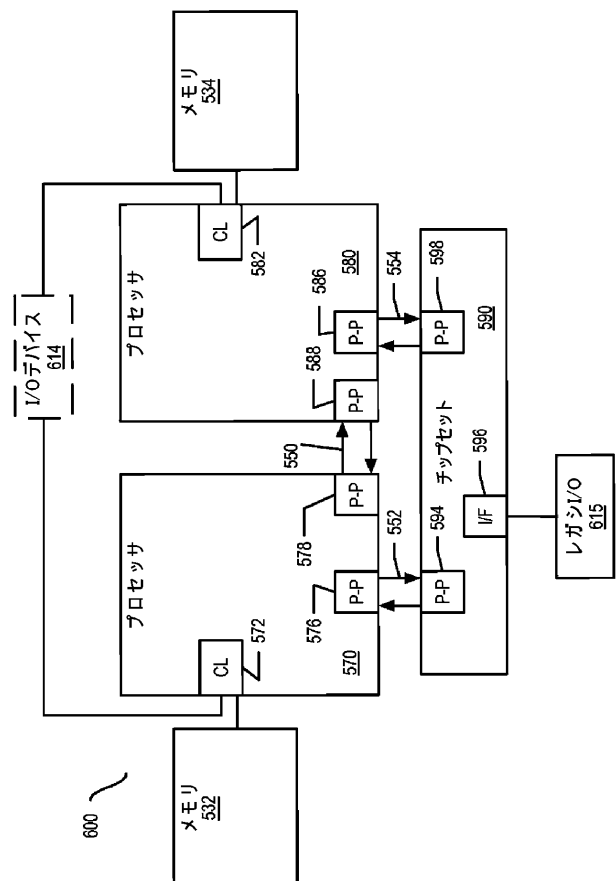
【図 4】



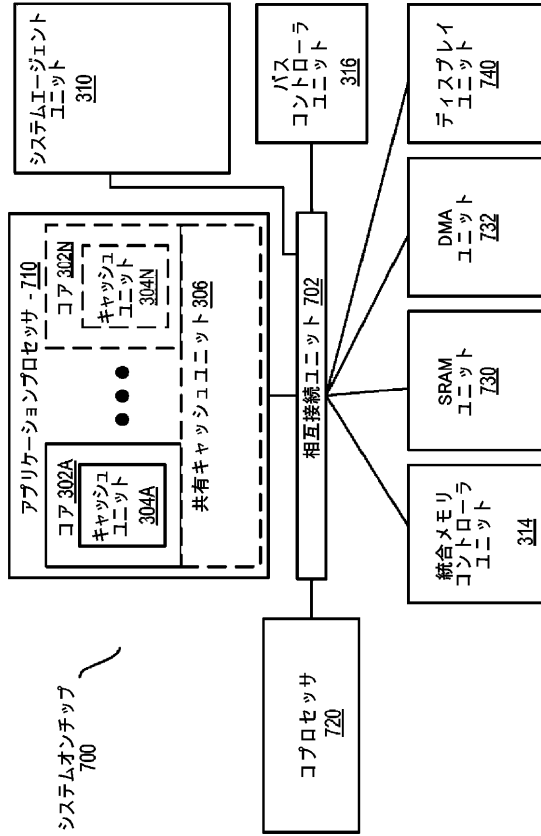
【図 5】



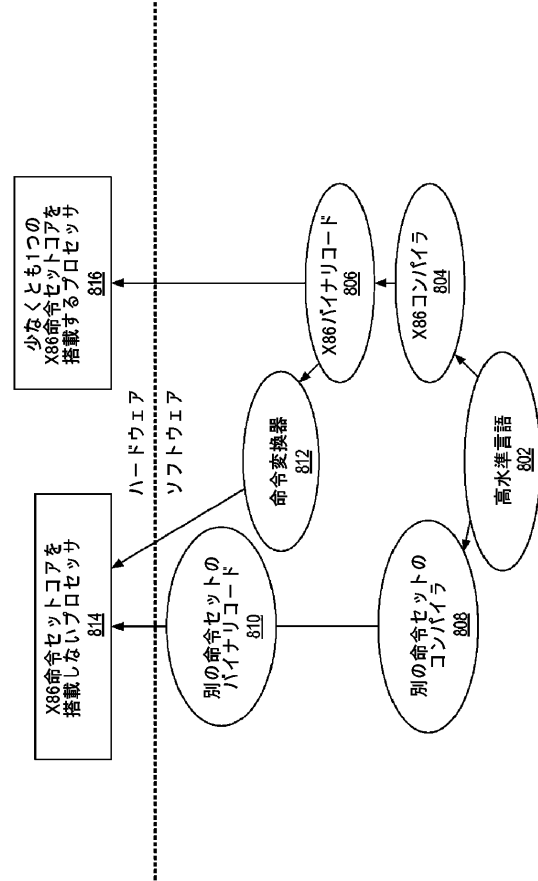
【図 6】



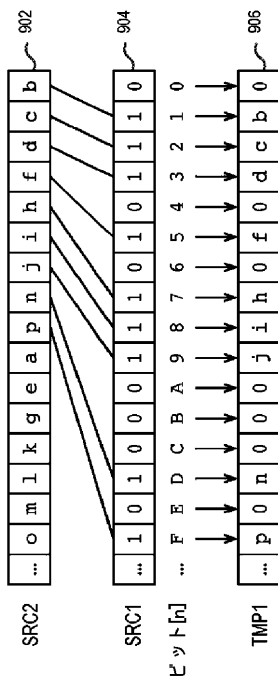
【図 7】



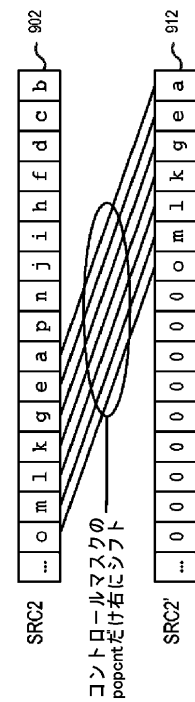
【図 8】



【図 9 A】

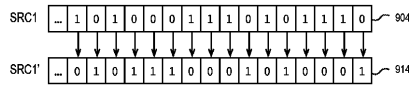


【図 9 B】

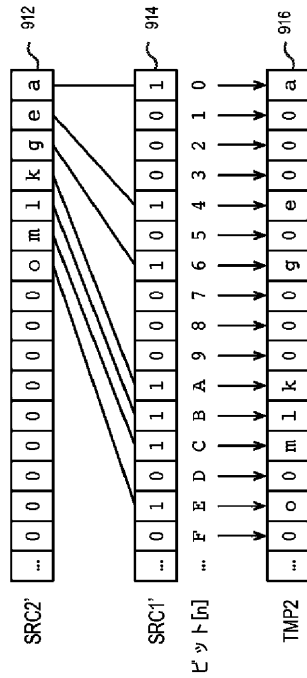


【図 9 C】

FIG. 9C

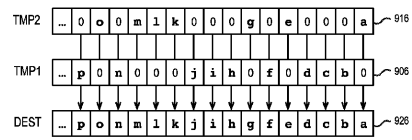


【図 9 D】

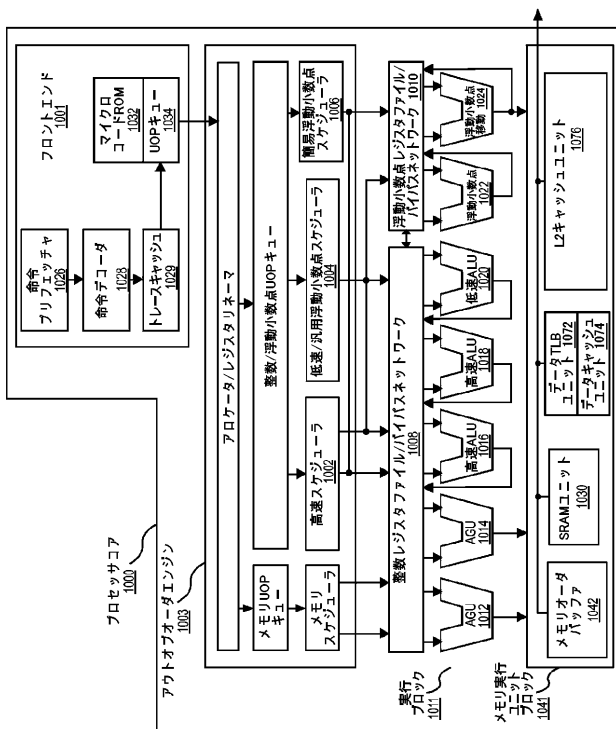


【図 9 E】

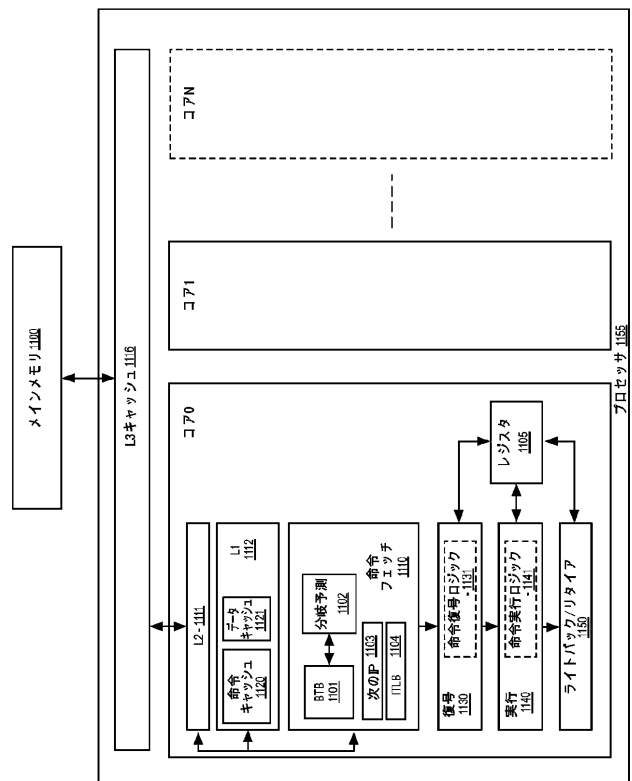
FIG. 9E



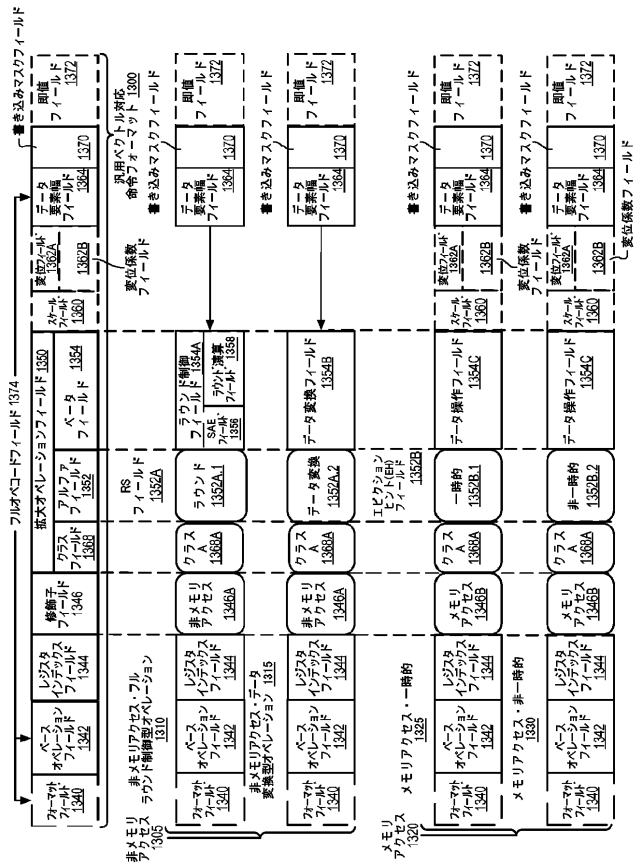
【 図 1 0 】



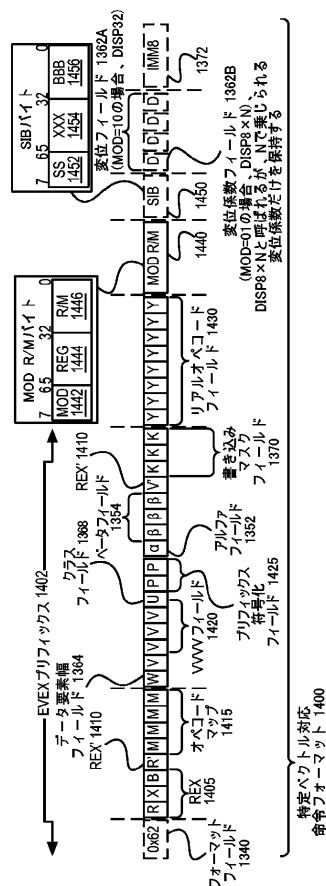
【图 1 1】



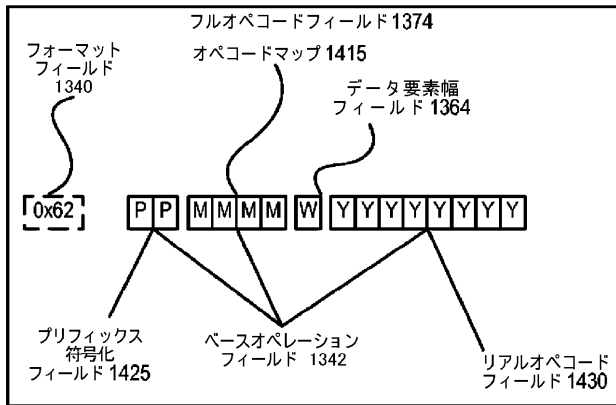
【図 1 3 A】



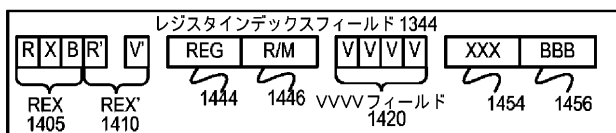
【図 14 A】



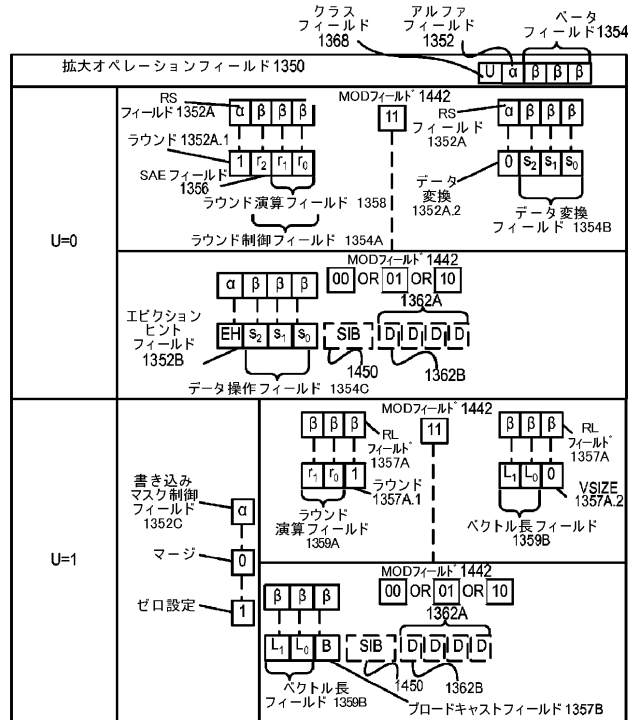
【図 1 4 B】



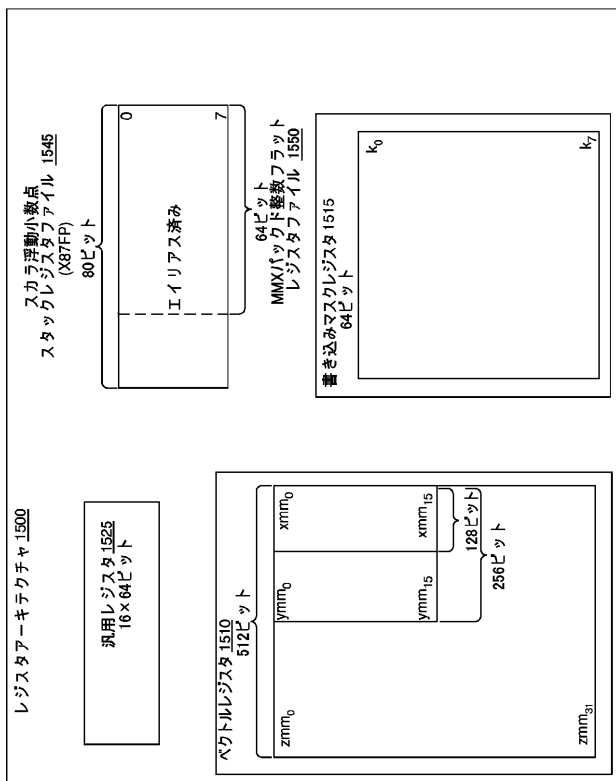
【図 1 4 C】



【図 1 4 D】



【図 1 5】



【手続補正書】

【提出日】平成29年7月20日(2017.7.20)

【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】

第 1 の命令を、第 1 のオペランドと第 2 のオペランドとを含んだ第 1 の復号された命令に復号する復号ロジックと、

前記第 2 のオペランドにより指定されるソースレジスタの両領域のビットを、前記第 1 のオペランドにより示される制御マスクに基づいてインターリーブする逆分離演算を実行するために、前記第 1 の復号された命令を実行する実行ユニットと

を備える

処理装置。

【請求項 2】

前記第 1 の命令をフェッチする命令フェッチユニットをさらに備え、前記第 1 の命令は単一の機械レベル命令である、

請求項 1 に記載の処理装置。

【請求項 3】

前記逆分離演算の結果をデスティネーションオペランドにより指定される位置にコミットするレジスタファイルユニットをさらに備える、

請求項 1 又は 2 に記載の処理装置。

【請求項 4】

前記レジスタファイルユニットは、

第 1 のソースオペランド値を格納する第 1 のレジスタと、

第 2 のソースオペランド値を格納する第 2 のレジスタと、

前記逆分離演算の前記結果の少なくとも 1 つのデータ要素を格納する第 3 のレジスタとを含むレジスタのセットをさらに格納する、

請求項 3 に記載の処理装置。

【請求項 5】

前記第 1 のレジスタは前記制御マスクを格納し、前記制御マスクの各ビットは、値を読み出すための前記ソースレジスタ内のビット位置を示す、

請求項 4 に記載の処理装置。

【請求項 6】

制御マスクビットの 1 は、前記第 2 のレジスタの第 1 の領域の値が取得されることを示し、制御マスクビットの 0 は、前記第 2 のレジスタの第 2 の領域の値が取得されることを示す、

請求項 5 に記載の処理装置。

【請求項 7】

前記第 2 のレジスタの前記第 1 の領域は、前記第 2 のレジスタの下位のバイトオーダービットを含み、前記第 2 のレジスタの前記第 2 の領域は、前記第 2 のレジスタの上位のバイトオーダービットを含む、

請求項 6 に記載の処理装置。

【請求項 8】

前記第 1 のレジスタ又は前記第 2 のレジスタは、32 ビット汎用レジスタ又は 64 ビット汎用レジスタである、

請求項 4 から 7 のいずれか一項に記載の処理装置。

【請求項 9】

前記第 1 のレジスタ又は前記第 2 のレジスタはベクトルレジスタである、
請求項 4 から 8 のいずれか一項に記載の処理装置。

【請求項 10】

前記ベクトルレジスタは、パックドデータ要素を格納するための 128 ビットレジスタ、256 ビットレジスタ、又は 512 ビットレジスタである、
請求項 9 に記載の処理装置。

【請求項 11】

前記パックドデータ要素は、バイト、ワード、ダブルワード、又はクワッドワードのデータ要素を含み、前記逆分離演算はビットを各データ要素にインターリーブする、
請求項 10 に記載の処理装置。

【請求項 12】

プロセッサにより実装される方法であって、
逆分離演算を実行する単一の命令をフェッチする段階であって、前記単一の命令は 2 つのソースオペランドと 1 つのデスティネーションオペランドを有する、段階と、
前記単一の命令を復号された命令に復号する段階と、
少なくとも 1 つのオペランドに関連付けられたソースオペランド値をフェッチする段階と、
第 2 のソースオペランドにより指定されるソースレジスタの両領域のビットを、第 1 のソースオペランドにより示される制御マスクに基づいてインターリーブするために、前記復号された命令を実行する段階と

を備える
方法。

【請求項 13】

前記第 1 のソースオペランドは即値オペランドである、
請求項 12 に記載の方法。

【請求項 14】

前記第 1 のソースオペランドは、前記制御マスクを含んだレジスタを指定する、
請求項 12 に記載の方法。

【請求項 15】

前記デスティネーションオペランドにより示される位置に結果を書き込む段階をさらに備える、
請求項 12 から 14 のいずれか一項に記載の方法。

【請求項 16】

前記デスティネーションオペランドはベクトルレジスタを示す、
請求項 15 に記載の方法。

【請求項 17】

前記復号された命令を実行する段階は、ソースレジスタの非連続ビットをデスティネーションレジスタに書き込むために、少なくとも 1 つの並列デポジット演算を実行する段階を含む、
請求項 15 に記載の方法。

【請求項 18】

前記デスティネーションレジスタは一時レジスタである、
請求項 17 に記載の方法。

【請求項 19】

複数の並列デポジット演算を複数の一時レジスタに対して実行する段階をさらに備える、
請求項 18 に記載の方法。

【請求項 20】

前記デスティネーションオペランドにより示される前記位置に前記結果を書き込む段階の前に、前記複数の一時レジスタに対して OR 演算を実行する段階をさらに備える、

請求項 19 に記載の方法。

【請求項 21】

請求項 12 から 20 のいずれか一項に記載の方法を実行する手段を備えるシステム。

【請求項 22】

請求項 12 から 20 のいずれか一項に記載の方法をコンピュータに実行させるプログラム。

【請求項 23】

請求項 22 に記載のプログラムを格納する、
コンピュータ可読記憶媒体。

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US2015/060812
A. CLASSIFICATION OF SUBJECT MATTER G06F 9/30(2006.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F 9/30; G06F 9/302; G06F 7/00; G06F 5/01; G06F 9/315; G06F 9/305; G06F 7/06		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: decode logic, execution unit, instruction, operand, inverse centrifuge operation, interleave bits, control mask, and similar terms.		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2013-0103730 A1 (TELEPUTER, LLC) 25 April 2013 See paragraphs [0083]-[0098]; and figures 12-14.	1-22
A	US 2011-0314263 A1 (DAN F. GREINER et al.) 22 December 2011 See paragraphs [0295]-[0302]; and figures 6A-6B.	1-22
A	US 2014-0095830 A1 (MIKHAIL PLOTNIKOV et al.) 03 April 2014 See paragraphs [0022]-[0053]; and figures 1-4.	1-22
A	US 6715066 B1 (GUY L. STEELE, JR.) 30 March 2004 See column 2, line 43 - column 3, line 41; claim 1; and figures 1A-1B.	1-22
A	US 6618804 B1 (GUY L. STEELE, JR. et al.) 09 September 2003 See column 2, line 30 - column 3, line 51; claim 1; and figure 1.	1-22
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 30 March 2016 (30.03.2016)		Date of mailing of the international search report 31 March 2016 (31.03.2016)
Name and mailing address of the ISA/KR International Application Division Korean Intellectual Property Office 189 Cheongsu-ro, Seo-gu, Daejeon, 35208, Republic of Korea Facsimile No. +82-42-481-8578		Authorized officer HAN, JOONG SUB Telephone No. +82-42-481-3578

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/060812

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013-0103730 A1	25/04/2013	US 2009-0138534 A1 US 8285766 B2 US 9134953 B2	28/05/2009 09/10/2012 15/09/2015
US 2011-0314263 A1	22/12/2011	AU 2010-355816 A1 CA 2786045 A1 CN 102298515 A EP 2419821 A1 JP 05039905 B2 JP 2012-009021 A KR 10-1464809 B1 KR 10-2011-0139100 A MX 2012014532 A RU 2012149548 A SG 186102 A1 WO 2011-160725 A1	05/07/2012 29/12/2011 28/12/2011 22/02/2012 03/10/2012 12/01/2012 27/11/2014 28/12/2011 03/04/2013 27/05/2014 30/01/2013 29/12/2011
US 2014-0095830 A1	03/04/2014	CN 104919432 A DE 112013004800 T5 GB 201500433 D0 GB 2518104 A JP 2015-534189 A KR 10-2015-0038328 A US 9122475 B2 WO 2014-051782 A1	16/09/2015 03/06/2015 25/02/2015 11/03/2015 26/11/2015 08/04/2015 01/09/2015 03/04/2014
US 6715066 B1	30/03/2004	None	
US 6618804 B1	09/09/2003	None	

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(72)発明者 ヴァレンティン、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 ブレー
バード・2200 インテル・コーポレーション内

(72)発明者 コーバル サン エイドリアン、ジーザス

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

(72)発明者 チャーニー、マーク、ジェイ.

アメリカ合衆国 95054 カリフォルニア州・サンタクララ・ミッション カレッジ ブレー
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