

【公報種別】特許法第 17 条の 2 の規定による補正の掲載
 【部門区分】第 6 部門第 3 区分
 【発行日】平成30年3月8日 (2018.3.8)

【公表番号】特表2018-500657(P2018-500657A)
 【公表日】平成30年1月11日 (2018.1.11)
 【年通号数】公開・登録公報2018-001
 【出願番号】特願2017-527588(P2017-527588)
 【国際特許分類】

G 0 6 F 9/30 (2018.01)

G 0 6 F 9/32 (2006.01)

G 0 6 F 9/305 (2006.01)

【 F I 】

G 0 6 F 9/30 3 5 0 F

G 0 6 F 9/32 3 2 0 F

G 0 6 F 9/30 3 4 0 C

G 0 6 F 9/30 3 4 0 A

【誤訳訂正書】
 【提出日】平成30年1月19日 (2018.1.19)
 【誤訳訂正 1】
 【訂正対象書類名】明細書
 【訂正対象項目名】0 0 0 1
 【訂正方法】変更
 【訂正の内容】
 【 0 0 0 1】

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る場合、単一の機械命令へと複数の命令を融合することを含む論理的、数学的、または他の機能動作を実施する処理ロジック、マイクロプロセッサ、及び関連付けられた命令セットアーキテクチャの分野に関する。

【誤訳訂正 2】
 【訂正対象書類名】明細書
 【訂正対象項目名】0 0 0 2
 【訂正方法】変更
 【訂正の内容】
 【 0 0 0 2】

命令セットまたは命令セットアーキテクチャ (I S A : i n s t r u c t i o n s e t a r c h i t e c t u r e) は、ネイティブデータタイプ、命令、レジスタアーキテクチャ、アドレス指定モード、メモリアーキテクチャ、割り込み及び例外ハンドリング、ならびに外部入出力 (I / O : i n p u t a n d o u t p u t) を含む、プログラミングに関係付けられたコンピュータアーキテクチャの一部である。バイナリトランスレーション (「 B T 」 : B i n a r y T r a n s l a t i o n) は、1 つのソース (「 ゲスト 」) のために構築されたバイナリを、別のターゲット (「 ホスト 」) I S A にトランスレートするための一般的な技法である。B T を使用すると、高レベルのソースコードを再コンパイルすることもなく、低レベルのアセンブリコードを書き直すことなく、異なるアーキテクチャを用いてプロセッサ上で 1 つのプロセッサ I S A のために構築されたアプリケーションバイナリを実行することが可能である。ほとんどのレガシーコンピュータアプリケーションがバイナリフォーマットのみ利用可能なため、B T は、プロセッサ用に作成されておらず利用できないアプリケーションを、そのプロセッサが実行可能とする潜在的なために非常に魅力的である。バイナリトランスレーションは、動的にまたは静的に実

施され得る。動的 B T (D B T : D y n a m i c B T) は、アプリケーションが実行されるとき、ランタイムにおいてバイナリトランスレーションを実施する。静的 B T (S B T : S t a t i c B T) は、バイナリが実行される前に、バイナリに対して実施される。

【誤訳訂正 3】

【訂正対象書類名】明細書

【訂正対象項目名】0 0 1 2

【訂正方法】変更

【訂正の内容】

【0 0 1 2】

【図 8】実施形態に従う、ソース命令セットにおけるバイナリ命令をターゲット命令セットにおけるバイナリ命令に転換するためのソフトウェア命令転換器の使用を対比するブロック図を図解する。

【誤訳訂正 4】

【訂正対象書類名】明細書

【訂正対象項目名】0 0 1 6

【訂正方法】変更

【訂正の内容】

【0 0 1 6】

【図 1 2】或る実施形態に従う、例示的な融合されたインクリメント__比較__ジャンプ命令を処理するためのロジックの流れ図である。

【誤訳訂正 5】

【訂正対象書類名】明細書

【訂正対象項目名】0 0 1 8

【訂正方法】変更

【訂正の内容】

【0 0 1 8】

【図 1 4 A】本発明の実施形態に従う、例示的な特有のベクトルフレンドリー命令フォーマットを図解するブロック図である。

【図 1 4 B】本発明の実施形態に従う、例示的な特有のベクトルフレンドリー命令フォーマットを図解するブロック図である。

【図 1 4 C】本発明の実施形態に従う、例示的な特有のベクトルフレンドリー命令フォーマットを図解するブロック図である。

【図 1 4 D】本発明の実施形態に従う、例示的な特有のベクトルフレンドリー命令フォーマットを図解するブロック図である。

【誤訳訂正 6】

【訂正対象書類名】明細書

【訂正対象項目名】0 0 3 2

【訂正方法】変更

【訂正の内容】

【0 0 3 2】

コアが(動作またはスレッドの 2 以上の並列セットを実行する) マルチスレッディングをサポートし得、時分割マルチスレッディング、同時マルチスレッディング(単一の物理コアが、物理コアが同時マルチスレッディングしているスレッドの各々に、論理のコアを提供する場合)、またはこれらの組み合わせ(例えば、Intel(登録商標)の Hyper-Threading Technology 等、時分割フェッチならびにデコーディング及びその後の同時マルチスレッディング)を含む様々な手段でそれを行い得ることを理解されたい。

【誤訳訂正 7】

【訂正対象書類名】明細書

【訂正対象項目名】 0 0 6 7

【訂正方法】 変更

【訂正の内容】

【 0 0 6 7 】

少なくとも1つのx86命令セットコア816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Intel（登録商標）プロセッサと実質的に同じ結果を成し遂げるために、以下を互換的に実行するか、そうでなければ別の方法で処理することによって、少なくとも1つのx86命令セットコアを有するIntel（登録商標）プロセッサと実質的に同じ機能を実施し得る任意のプロセッサを表す。（1）Intel（登録商標）x86命令セットコアの命令セットの実質的な部分、または（2）少なくとも1つのx86命令セットコアを有するIntel（登録商標）プロセッサ上で動作することを目標とされたアプリケーションまたは他のソフトウェアのオブジェクトコードバージョン。x86コンパイラ804は、追加のリンケージ処理の有る無しに関わらず、少なくとも1つのx86命令セットコア816を有するプロセッサ上で実行され得るx86バイナリコード806（例えば、オブジェクトコード）を生成するように動作可能なコンパイラを表す。同様に、図8は、高レベル言語802におけるプログラムが、代替の命令セットコンパイラ808を使用してコンパイルされて、少なくとも1つのx86命令セットコア814を有しないプロセッサ（例えば、カリフォルニア州サニーベールのMIPS TechnologiesのMIPS命令セットを実行するか、及び/またはイングランドのケンブリッジのARM HoldingsのARM命令セットを実行するコアを有するプロセッサ）によってネイティブに実行され得る代替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。

【誤訳訂正8】

【訂正対象書類名】 明細書

【訂正対象項目名】 0 0 7 8

【訂正方法】 変更

【訂正の内容】

【 0 0 7 8 】

しかしながら、fragment__Aまたはfragment__Bの命令が、追加、比較、またはジャンプ命令のオペランドとのデータ依存性を有しない場合、入来コードストリームにおいて追加の命令を可能にすることが正当であり、トランスレータは、任意のデータ依存性に違反することなく、自由にこれらの命令を順序変更するべきである。したがって、トランスレーションロジックは、ブロック928において、命令の検出されたシーケンス内のコード断片において、任意の命令を順序変更し得る。ブロック930において、トランスレーションロジックは、比較動作のためのレジスタ及び定数値、ならびにジャンプ動作のためのジャンプラベルを含む、命令シーケンスを実施するよう要求されるオペランドを含む、単一のインクリメント__比較__ジャンプ命令と、別個のインクリメント、比較、ジャンプ命令を置き換える。例示的な順序変更されたコードシーケンスは、以下の表2に示される。

【表2】

表2 例示のプログラムコード

(6)	<0 or more instructions_fragment_A>
(7)	<0 or more instructions_fragment_B>
(8)	INC_CMP_JE EAX, constant_value_for_compare, jump_label

【誤訳訂正9】

【訂正対象書類名】 明細書

【訂正対象項目名】 0 0 8 0

【訂正方法】 変更

【訂正の内容】

【 0 0 8 0 】

例示的な融合された命令プロセッサ実装 図 1 0 A ~ 1 0 B は、インクリメント__比較__ジャンプ命令の例示的なプロセッサ実装を図解するブロック図である。いくつかの実施形態では、実装するプロセッサは、命令を実装するためのいくつかのアーキテクチャの特徴を含む。図 1 0 A は、或る実施形態に従う、動作を実施するためのロジックを含むプロセッサコアのブロック図である。図 1 0 B は、或る実施形態に従う、インクリメント__比較__ジャンプ命令を実装するための例示的な具体的なマイクロアーキテクチャのブロック図である。

【誤訳訂正 1 0】

【訂正対象書類名】 明細書

【訂正対象項目名】 0 0 8 5

【訂正方法】 変更

【訂正の内容】

【 0 0 8 5 】

一実施形態では、アロケータは、命令スケジューラ、メモリスケジューラ、高速スケジューラ 1 0 0 2、遅い/一般的な浮動小数点スケジューラ 1 0 0 4、及び簡易浮動小数点スケジューラ 1 0 0 6 の前に、各 u o p のエントリを 2 つの u o p キューのうち 1 つに対して、即ち、1 つをメモリ動作に、1 つを非メモリ動作という方式で割り当てる。u o p スケジューラ 1 0 0 2、1 0 0 4、及び 1 0 0 6 は、それらの従属入力レジスタオペランドソースの準備ができていて、及び u o p がそれらの動作を完了する必要がある実行リソース u o p の利用可能性に基づいて、いつ u o p の準備ができていてのかを判定する。一実施形態の高速スケジューラ 1 0 0 2 は、主クロックサイクルの各半分にスケジューリングをする場合がある一方で、他のスケジューラは、主プロセッサクロックサイクルにつき一度しかスケジューリングをしない場合がある。スケジューラは、実行のための u o p のスケジューリングをするために、ディスパッチポート間を調整する。

【誤訳訂正 1 1】

【訂正対象書類名】 明細書

【訂正対象項目名】 0 0 8 6

【訂正方法】 変更

【訂正の内容】

【 0 0 8 6 】

レジスタファイル 1 0 0 8、1 0 1 0 は、実行ブロック 1 0 1 1 において、スケジューラ 1 0 0 2、1 0 0 4、1 0 0 6、及び実行ユニット 1 0 1 2、1 0 1 4、1 0 1 6、1 0 1 8、1 0 2 0、1 0 2 2、1 0 2 4 の間に位置する。一実施形態では、整数及び浮動小数点動作に対してそれぞれ、別個のレジスタファイル 1 0 0 8、1 0 1 0 が存在する。一実施形態では、各レジスタファイル 1 0 0 8、1 0 1 0 は、まだレジスタファイルに書き込まれていない完了した結果を、新しい従属 u o p にバイパスまたは転送し得るバイパスネットワークを含み得る。整数レジスタファイル 1 0 0 8 及び浮動小数点レジスタファイル 1 0 1 0 は、また、データを他と通信することができる。一実施形態について、整数レジスタファイル 1 0 0 8 は、2 つの別個のレジスタファイルに、つまり、1 つのレジスタファイルをデータの低オーダ 3 2 ビットに、第 2 のレジスタファイルをデータの高オーダ 3 2 ビットにという方式で、分けられる。一実施形態では、浮動小数点レジスタファイル 1 0 1 0 は 1 2 8 ビット幅エントリを有する。

【誤訳訂正 1 2】

【訂正対象書類名】 明細書

【訂正対象項目名】 0 0 8 8

【訂正方法】 変更

【訂正の内容】

【0088】

一実施形態では、浮動小数点値に関わる命令は、浮動小数点ハードウェアを用いて扱われ得る。ALU動作は、高速ALU実行ユニット1016、1018に移行する。一実施形態の高速ALU1016、1018は、クロックサイクルの半分の効果的なレイテンシーで高速動作を実行し得る。一実施形態について、ほとんどの複合整数動作は遅いALU1020に移行する。なぜなら、遅いALU1020は、乗算器、シフト、フラグロジック、及び分岐処理等の長いレイテンシータイプの動作のための整数実行ハードウェアを含むからである。メモリロード/記憶動作は、AGU1012、1014によって実行される。一実施形態について、整数ALU1016、1018、1020は、64ビットデータオペランドに対して整数動作を実施するコンテキストにおいて記載される。代替の実施形態では、ALU1016、1018、1020は、16、32、128、256等を含む様々なデータビットをサポートするように実装され得る。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オペランドの範囲をサポートするように実装され得る。一実施形態について、浮動小数点ユニット1022、1024は、SIMD及びマルチ媒体命令と併せて、128ビット幅パックデータオペランド上で動作し得る。

【誤訳訂正13】

【訂正対象書類名】明細書

【訂正対象項目名】0092

【訂正方法】変更

【訂正の内容】

【0092】

実行ブロック1011の例示的な部分は、図10Bに示されるようなロジックを含み、それは、単一のサイクルインクリメント__比較__ジャンプ命令を実装するためのマイクロアーキテクチャ1050を図解する。一実施形態では、図解されたマイクロアーキテクチャ1050は、プロセッサ実行パイプライン内で実行ステージを実施するように構成される。マイクロアーキテクチャ1050は、算術ロジックユニット(ALU)1054及びジャンプ実行ユニット(JEU: jump execution unit)1056を含み、分岐及び算術命令を実行することができる。パイピングロジック1052A~Bは、マイクロアーキテクチャを、前の及び連続するパイプラインステージのためのロジックと繋げ、ALU演算の結果1063(例えば、 $B+1$)を連続するパイプラインステージに渡すために、ALU1054にオペランド(例えば、オペランド__A1060、オペランド__B1061)を供給する。一実施形態では、インクリメント動作の結果は、入力オペランドによって指示された適切なレジスタにコミットされる。制御ユニットからのALU1054への制御信号1066は、使用されて、ALU動作の間で選択するか、一実施形態では、オペコードをALUに提供する。制御信号1067は、また、制御ユニットから制御JEU動作までJEUに提供される。

【誤訳訂正14】

【訂正対象書類名】明細書

【訂正対象項目名】0094

【訂正方法】変更

【訂正の内容】

【0094】

単一の実行サイクル内でインクリメント__比較__ジャンプ命令を実施するために、各コンポーネントは、サイクル内の適切な点における適切な入力を要求する。例えば、ALUフラグ1064は、サイクルにおける早期にJEU1056に到達するべきで、それらは、マルチサイクルバイパスの結果であり得ない。一実施形態では、フラグの具体的なサブセット(例えば、桁上げ、ゼロ、符号、オーバーフロー等)は、タイミング限定に基づいて、条件付きジャンプのために使用される。一実施形態では、アーキテクチャフラグレジ

スタ内のすべてのフラグは、パリティフラグを含むジャンプ状況のために使用され得る。

【誤訳訂正 15】

【訂正対象書類名】明細書

【訂正対象項目名】0096

【訂正方法】変更

【訂正の内容】

【0096】

図11は、或る実施形態に従う、インクリメント__比較__ジャンプ命令を実施するための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、主メモリ1100に結合されたプロセッサ1155を含む。プロセッサ1155は、インクリメント__比較__ジャンプ命令をデコードするためのデコードロジック1131を有するデコードユニット1130を含む。追加として、プロセッサ実行エンジンユニット1140は、命令を実行するための追加の実行ロジック1141を含む。レジスタ1105は、実行ユニット1140が命令ストリームを実行するとき、オペランド、制御データ、及び他のタイプのデータに、レジスタ記憶を提供する。

【誤訳訂正 16】

【訂正対象書類名】明細書

【訂正対象項目名】0099

【訂正方法】変更

【訂正の内容】

【0099】

図12は、或る実施形態に従う、ロジックが、インクリメント__比較__ジャンプ命令を処理するための流れ図である。ブロック1202において、命令パイプラインは、インクリメント__比較__ジャンプ命令を実施するための命令のフェッチから始まる。命令は、命令のインクリメント及び比較部分のための第1の及び第2の入力オペランド、ならびに命令の条件付きジャンプ部分のためのジャンプラベルオペランドを受け入れる。一実施形態では、第1のオペランドは、レジスタまたは即値であり得る一方で、第2のオペランドは、レジスタ、即値、またはメモリアドレスであり得る。いくつかの実施形態では、ジャンプラベルは、ジャンプターゲットアドレスに転換されるジャンプ命令からオフセットされる即値である。

【誤訳訂正 17】

【訂正対象書類名】明細書

【訂正対象項目名】0100

【訂正方法】変更

【訂正の内容】

【0100】

ブロック1204において、デコードユニットは、インクリメント__比較__ジャンプ命令をデコードされた命令へとデコードする。一実施形態では、デコードされた命令は、単一のプロセッササイクルにおいて実行される単一の動作である。一実施形態では、デコードされた命令は、命令の各サブ要素を実施するための1または複数のマイクロ動作を含む。マイクロ動作は、ハードワイヤードであり得、あるいは、マイクロコード動作は、実行ユニット等のプロセッサのコンポーネントに、命令を実装するための様々な動作を実施させ得る。

【誤訳訂正 18】

【訂正対象書類名】明細書

【訂正対象項目名】0106

【訂正方法】変更

【訂正の内容】

【0106】

上記表 4 に示されるように、融合されたインクリメント__比較__ジャンプ命令は、スケジューリングされ、レジスタファイル読み出しを遂行し、別個の命令よりも早期に 2 つのサイクルを実行する。追加として、別個のアクションを実施するために要求されるハードウェア命令の数を減少させることは、様々な機能的ユニットへの圧力を減少させ、それらのユニットに自由に他の動作を実施させておき得る。一実施形態では、減少された数の命令が、プロセッサハードウェア内で、スケジューリングされ、管理されるので、融合された命令は、スケジューリング及び記帳ハードウェアに対する要求を減少させる。追加として、減少されたリソースが順序変更バッファ及びリザベーションステーションに要求される。

【誤訳訂正 19】

【訂正対象書類名】明細書

【訂正対象項目名】0115

【訂正方法】変更

【訂正の内容】

【0115】

レジスタインデックスフィールド 1344 - その内容は、直接的にまたはアドレス生成を通して、ソース及び宛先オペランドの位置を指定する（それらがレジスタ内またはメモリ内にある場合）。これらは、 $P \times Q$ （例えば 32×512 、 16×128 、 32×1024 、 64×1024 ）レジスタファイルから N レジスタを選択するための十分なビット数を含む。一実施形態では、 N は、3 つのソース及び 1 つの宛先レジスタまでであり得、代替の実施形態は、より多くのまたはより少ないソース及び宛先レジスタをサポートし得る（例えば、2 つのソースまでをサポートし得て、この場合には、これらのソースのうち 1 つがまた、宛先の働きをする。3 つのソースまでをサポートし得て、この場合には、これらのソースのうち 1 つがまた、宛先の働きをする。あるいは、2 つのソース及び 1 つの宛先までをサポートし得る）。

【誤訳訂正 20】

【訂正対象書類名】明細書

【訂正対象項目名】0144

【訂正方法】変更

【訂正の内容】

【0144】

[例示的な特有のベクトルフレンドリー命令フォーマット]

図 14 は、本発明の或る実施形態に従う、例示的な特有のベクトルフレンドリー命令フォーマットを図解するブロック図である。図 14 は、それが、フィールドの位置、サイズ、解釈、及びオーダ、ならびにそれらのフィールドのいくつかのための値を指定するという意味において特有である、特有のベクトルフレンドリー命令フォーマット 1400 を示す。特有のベクトルフレンドリー命令フォーマット 1400 は、 $\times 86$ 命令セットを拡張するために使用され得、したがって、フィールドのいくつかは、既存の $\times 86$ 命令セット及びその拡張（例えば、 AVX ）において使用されるものと類似しているか同じである。このフォーマットは、拡張を有する既存の $\times 86$ 命令セットのプレフィックスエンコーディングフィールド、真の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整合するままである。図 14 のマップからのフィールドがマッピングする図 13 からのフィールドが図解される。

【誤訳訂正 21】

【訂正対象書類名】明細書

【訂正対象項目名】0145

【訂正方法】変更

【訂正の内容】

【0145】

特有のベクトルフレンドリー命令フォーマット 1400 を参照して、図解の目的のため

の汎用ベクトルフレンドリー命令フォーマット 1300 のコンテキストにおいて、実施形態が記載されるが、本発明は、特許請求の範囲に記載される場合を除いて、特有のベクトルフレンドリー命令フォーマット 1400 に限定されないことを理解されたい。例えば、汎用ベクトルフレンドリー命令フォーマット 1300 は、様々なフィールドについての様々な可能なサイズを考慮する一方で、特有のベクトルフレンドリー命令フォーマット 1400 は、具体的なサイズのフィールドを有するものとして示される。具体的な例として、データ要素幅フィールド 1364 は、特有のベクトルフレンドリー命令フォーマット 1400 において、1 ビットフィールドとして図解される一方で、本発明はそのように限定されない（つまり、汎用ベクトルフレンドリー命令フォーマット 1300 は、データ要素幅フィールド 1364 の他のサイズを考慮する）。

【誤訳訂正 22】

【訂正対象書類名】明細書

【訂正対象項目名】0157

【訂正方法】変更

【訂正の内容】

【0157】

アルファフィールド 1352 (EVEX バイト 3、ビット [7] - EH、EVEX . EH、EVEX . rs、EVEX . RL、EVEX . 書き込みマスク制御、及び EVEX . N としても知られ、また、 と共に図解された) - 前述のように、このフィールドはコンテキスト固有である。

【誤訳訂正 23】

【訂正対象書類名】明細書

【訂正対象項目名】0167

【訂正方法】変更

【訂正の内容】

【0167】

[フルオペコードフィールド]

図 14B は、一実施形態に従うフルオペコードフィールド 1374 を作り上げる特有のベクトルフレンドリー命令フォーマット 1400 のフィールドを図解するブロック図である。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、ベース動作フィールド 1342、及びデータ要素幅 (W) フィールド 1364 を含む。ベース動作フィールド 1342 は、プレフィックスエンコーディングフィールド 1425、オペコードマップフィールド 1415、及び真のオペコードフィールド 1430 を含む。

【誤訳訂正 24】

【訂正対象書類名】明細書

【訂正対象項目名】0168

【訂正方法】変更

【訂正の内容】

【0168】

[レジスタインデックスフィールド]

図 14C は、一実施形態に従うレジスタインデックスフィールド 1344 を作り上げる特有のベクトルフレンドリー命令フォーマット 1400 のフィールドを図解するブロック図である。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、R'EX 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。

【誤訳訂正 25】

【訂正対象書類名】明細書

【訂正対象項目名】0169

【訂正方法】変更

【訂正の内容】

【0169】

〔増大動作フィールド〕

図14Dは、一実施形態に従う増大動作フィールド1350を作り上げる特有のベクトルフレンドリー命令フォーマットの1400のフィールドを図解するブロック図である。クラス(U)フィールド1368が0を含む場合、それはEVEX.U0(クラスA1368A)を意味し、それが1を含む場合、それはEVEX.U1(クラスB1368B)を意味する。U=0、かつMODフィールド1442が11(メモリアクセス無し動作を意味する)を含む場合、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として解釈される。rsフィールド1352Aが、1(丸め1352A.1)を含む場合、ベータフィールド1354(EVEXバイト3、ビット[6:4]-SSS)は、丸め制御フィールド1354Aとして解釈される。丸め制御フィールド1354Aは、1ビットSAEフィールド1356及び2ビット丸め動作フィールド1358を含む。rsフィールド1352Aは、0(データ変換1352A.2)を含む場合、ベータフィールド1354(EVEXバイト3、ビット[6:4]-SSS)は、3ビットデータ変換フィールド1354Bとして解釈される。U=0、かつMODフィールド1442が、00、01、または10(メモリアクセス動作を意味する)を含む場合、アルファフィールド1352(EVEXバイト3、ビット[7]-EH)は、放逐ヒント(EH)フィールド1352Bとして解釈され、ベータフィールド1354(EVEXバイト3、ビット[6:4]-SSS)は、3ビットデータ操作フィールド1354Cとして解釈される。

【誤訳訂正26】

【訂正対象書類名】明細書

【訂正対象項目名】0171

【訂正方法】変更

【訂正の内容】

【0171】

〔例示的なレジスタアーキテクチャ〕

図15は、一実施形態に従う、レジスタアーキテクチャ1500のブロック図である。図解された実施形態では、512ビット幅である32ベクトルレジスタ1510が存在し、これらのレジスタは、zmm0~zmm31として参照される。下位16zmmレジスタの下位オーダ256ビットは、レジスタymm0-16上でオーバーレイされる。下位16zmmレジスタの下位オーダ128ビット(ymmレジスタの下位オーダ128ビット)は、レジスタxmm0-15上でオーバーレイされる。特有のベクトルフレンドリー命令フォーマット1400は、下の表5に図解されるように、これらのオーバーレイされたレジスタ上で動作する。

【表 5】

表 5ーレジスタファイル

調節可能なベクトル長	クラス	動作	レジスタ
ベクトル長フィールド 1 3 5 9 Bを含まない 命令テンプレート	A (図 1 3 A、 U=0)	1 3 1 0、1 3 1 5、1 3 2 5、1 3 3 0	z mmレジスタ (ベク トル長は6 4 バイトで ある)
	B (図 1 3 B、 U=1)	1 3 1 2	z mmレジスタ (ベク トル長は6 4 バイトで ある)
ベクトル長フィールド 1 3 5 9 Bを含まない 命令テンプレート	B (図 1 3 B、 U=1)	1 3 1 7、1 3 2 7	z mm、y mm、また は x mmレジスタ (ベ クトル長は、6 4 バイ ト、3 2 バイト、また は1 6 バイトである) ベクトル長フィールド 1 3 5 9 B次第である

【誤訳訂正 2 7】

【訂正対象書類名】明細書

【訂正対象項目名】0 1 7 2

【訂正方法】変更

【訂正の内容】

【0 1 7 2】

言い換えると、ベクトル長フィールド 1 3 5 9 B は、最大長さ及び 1 または複数の他のより短い長さの間で選択し、各々のそのようなより短い長さは、先行する長さの半分の長さであり、ベクトル長フィールド 1 3 5 9 B を有しない命令テンプレートは、最大ベクトル長上で動作する。さらに、一実施形態では、特有のベクトルフレンドリー命令フォーマット 1 4 0 0 のクラス B 命令テンプレートは、パックまたはスカラ単一 / 倍精度浮動小数点データ及びパックまたはスカラ整数データ上で動作する。スカラ動作は、z mm / y mm / x mm レジスタ内の最低のオーダデータ要素位置に対して実施される動作であり、より高いオーダデータ要素位置は、それらが命令の前の状態か、または実施形態に従うゼロ化の状態のままかのどちらかにされる。