

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4990977号  
(P4990977)

(45) 発行日 平成24年8月1日(2012.8.1)

(24) 登録日 平成24年5月11日(2012.5.11)

(51) Int.Cl.

G06F 7/38 (2006.01)

F 1

G06F 7/38

B

請求項の数 4 (全 15 頁)

(21) 出願番号	特願2009-528536 (P2009-528536)
(86) (22) 出願日	平成19年9月21日 (2007.9.21)
(65) 公表番号	特表2010-503935 (P2010-503935A)
(43) 公表日	平成22年2月4日 (2010.2.4)
(86) 國際出願番号	PCT/US2007/079233
(87) 國際公開番号	W02008/036944
(87) 國際公開日	平成20年3月27日 (2008.3.27)
審査請求日	平成21年3月17日 (2009.3.17)
(31) 優先権主張番号	11/525,420
(32) 優先日	平成18年9月22日 (2006.9.22)
(33) 優先権主張国	米国(US)

前置審査

(73) 特許権者	593096712 インテル コーポレイション アメリカ合衆国 95052 カリフォルニア州 サンタクララ ミッション カレッジ ブールバード 2200
(74) 代理人	100070150 弁理士 伊東 忠彦
(74) 代理人	100091214 弁理士 大貫 進介
(74) 代理人	100107766 弁理士 伊東 忠重
(72) 発明者	ゾハール、ローネン アメリカ合衆国 94087 カリフォルニア州 サニーヴェイル ホーレンベック アヴェニュー 857

最終頁に続く

(54) 【発明の名称】命令に応じた丸め処理の実行

## (57) 【特許請求の範囲】

## 【請求項 1】

丸められた結果を得るための第一のオペランドに対する丸め命令の実行を、前記丸め命令に関連付けられた制御フィールド内に、丸めモード部分とは別個のオーバーライド指標が存在していれば、前記丸め命令に関連付けられた制御フィールド内に記述されている丸めモードに従って行う手段であって、前記オーバーライド指標はデフォルト丸めモードがオーバーライドされるべきかどうかを示す、手段と；

前記丸め命令および前記制御フィールドを受領し、前記制御フィールドの丸めモード部分を復号し、前記オーバーライド指標が存在していれば前記丸めモード部分から制御信号を生成する、前記実行を行う手段に結合された選択器手段であって、前記丸め命令の実行をさせるよう前記制御信号を前記実行を行う手段に与え、前記丸めモード部分は最も近い偶数への丸めモード、中間点で零から遠いほうへの丸めモード、零から遠いほうへの丸めモードおよび零への丸めモードを含む複数の丸めモードの一つをエンコードする、選択器手段と；

前記実行を行う手段のためのデフォルト丸めモードを記憶する制御レジスタと；

前記実行を行う手段に結合された記憶手段とを有しており、

前記実行を行う手段が前記丸め命令の実行を、前記制御フィールド内に前記オーバーライド指標が存在していれば、前記デフォルト丸めモードとは異なる丸めモードに従って行い、

前記実行を行う手段が前記異なる丸めモードを実行するのを、前記第一のオペランドが

閾値以下である場合には前記第一のオペランドへの第一の値の加算を介して、そうでない場合には前記第一のオペランドへの第二の値の加算を介して行い、前記第一の値は浮動小数点形式で表された前記第二の値と絶対値が等しく符号が異なるものである、システム。

#### 【請求項 2】

前記実行を行う手段が、前記選択器手段からの制御信号に応答して前記第一のオペランドに対して丸め処理を実行する浮動小数点ユニットを有し、前記第一のオペランドが限られた精度の値をもつ、請求項 1 記載のシステム

#### 【請求項 3】

前記実行を行う手段が前記丸め命令を前記異なる丸めモードに従って実行することを、10  
前記デフォルト丸めモードが前記制御レジスタに記憶されている間に行う、請求項 1 記載のシステム。

#### 【請求項 4】

命令が記憶されている機械可読媒体であって、前記命令は機械によって実行されたときに、該機械に：

前記命令によって規定されるモードに従ってオペランドに対する丸め処理を実行する段階であって、前記命令は該命令に関連付けられた、デフォルトの丸めモードをオーバーライドするためのオーバーライド指標を有し、当該段階は前記命令に関連付けられた直接値の一部分を復号してモードを決定することを含み、前記直接値の前記一部分は複数の丸めモードの一つに対応する符号 (code) を含み、前記直接値は前記直接値の一部分とは別個の前記オーバーライド指標を含み、前記モードに基づいて、前記オペランドが閾値以下である場合には前記オペランドは第一の値を加算され、そうでない場合には前記オペランドは第二の値を加算され、前記第一の値は浮動小数点形式で表された前記第二の値と絶対値が等しく符号が異なるものである、段階と；20

前記丸め命令の結果を第一の記憶領域に記憶する段階とを有する方法を実行させ、

前記モードが、制御レジスタに記憶されているデフォルト丸めモードとは異なる丸めモードに従って前記丸め処理の実行を引き起こすものである、

機械可読媒体。

#### 【発明の詳細な説明】

##### 【技術分野】

30

##### 【0 0 0 1】

本願は丸め処理に関する。

##### 【背景技術】

##### 【0 0 0 2】

プロセッサはデータに対してさまざまな数学的操作を実行する。データは、たとえば種々の本来的精度をもつ整数値および浮動小数点 (FP: floating point) 値を含む種々の型のものであってよい。FP値を扱うとき、数学的操作、こうした操作のうちでも乗算または加算といった演算の結果は、より低精度フォーマットに変換される必要のある結果を生じる。したがって、FP結果を丸めるよう丸め処理 (rounding operation) が実行されることができる。40

##### 【発明の概要】

##### 【発明が解決しようとする課題】

##### 【0 0 0 3】

そのような丸め処理は種々の数学的演算の一部として実行されることがあるが、一部のプロセッサ・アーキテクチャでは、データ要素に対して丸め処理を、単体の操作として実行する、あるいは複数の複雑なステップの必要なしに実行するための機能が限られていたり、あるいは全くなかつたりする。たとえば、プロセッサはFP値の整数値への丸めをデフォルトの丸めモードに従って実行するよう構成されていることがある。しかしながら、所与のソース・オペランドは、さまざまな理由のため、別のモードに従って丸められる必要があることがある。そのような処理を実施するため、プロセッサの現在の構成設定状態50

( configuration state) を保存し、所望される丸めモードに関する情報を含む新しい構成設定状態をロードし、丸め処理を実行し、もとのプロセッサ状態を復元するという入り組んだステップが生起することがある。これらの処理は時間がかかり、計算量を高め、過剰なプロセッサ・サイクルを消費することがある。さらに、プロセッサにおいて実行される丸め処理は典型的には限られた量の丸めモード、すなわち米国電気電子技術者協会 (IEEE: Institute of Electrical and Electronics Engineers) 規格754-1985 (1985年発表) に述べられている丸めモードに従って生起するが、新しいプログラミング言語が進化するにつれ、他の丸めモードのサポートが望ましいことがありうる。

**【課題を解決するための手段】**

**【0 0 0 4】**

解決しようとする課題は請求項記載の発明によって解決される。

10

**【図面の簡単な説明】**

**【0 0 0 5】**

【図1】本発明のある実施形態に基づく方法の流れ図である。

【図2】本発明のある実施形態に基づくプロセッサの一部のブロック図である。

【図3】本発明のある実施形態に基づく命令との関連で使用されるべき直接データ要素 (immediate data element) のブロック図である。

【図4】本発明のある実施形態に基づく丸め処理を実行する方法の流れ図である。

【図5】本発明のある実施形態に基づくシステムのブロック図である。

**【発明を実施するための形態】**

**【0 0 0 6】**

さまざまな実施形態において、プロセッサにおいて、たとえば該プロセッサの浮動小数点ユニット (FPU: floating point unit) において丸め処理を効率的に実行するためには、命令セット・アーキテクチャ (ISA: instruction set architecture) の複数の丸め命令が使用されうる。米国電気電子技術者協会 (IEEE: Institute of Electrical and Electronics Engineers) 規格754-1985 (1985年発表) (本稿では二進浮動小数点演算のためのIEEE規格またはIEEE規格754) に述べられている丸めモードに加えて、追加的な丸めモードに従って丸め処理を実行するために諸実施形態が使用されうる。たとえばいくつかの実施形態では、命令が、下記に述べるような中間点で零から遠いほうへの (halfway away from zero) 丸め処理および零から遠いほうへの (away from zero) 丸め処理のサポートを提供してもよい。さらに、これらの丸め処理は多くのデータ型とともに使用されうる。いくつかの実装では、丸め処理は単一命令複数データ (SIMD: single instruction multiple data) データ型に対して実行できることにより、それにより命令は、パックされたデータ要素といった拡張されたデータ型に対して実行ができる。パックされたデータ要素 (packed data element) では、複数のデータ要素が、プロセッサの拡張されたレジスタのような单一の位置にパックされる。

20

30

**【0 0 0 7】**

柔軟性を受け入れ、効率的な命令実行を提供するために、諸実施形態は、ソース・オペランドに対して実行されることのできるISAベースの命令を提供しうる。これらのISAベースの命令は、ソース・オペランドの最も近い整数値への丸めを実行する丸め処理のさまざまな実装であってよい。そのようなソース・オペランドはすでに限られた精度のフォーマットにあってもよい (すなわち、算術演算の結果ではなくレジスタ / メモリから読まれたデータ)。そのような命令は、マルチメディア・アプリケーション、ゲーム・アプリケーションなどを含むさまざまなアプリケーションに使用されうる。さらに、諸実施形態は、さまざまなプログラミング言語に適用可能でありうる丸め処理を可能にするコンパイラ・ベースのプリミティブにおいて実装されてもよい。さまざまな実施形態において、丸め命令は、ソース・オペランドとして浮動小数点数を取り、それを最も近い整数値に丸め、結果をやはり整数値をもつ浮動小数点値として記憶しうることを注意しておく。

40

**【0 0 0 8】**

さまざまな実施形態において、実行の制御は、少なくとも部分的には、命令とともに受

50

領された情報、たとえば命令とともに受信された直接データ (immediate data) に基づいて扱われる。種々の実装において、そのような直接データは、プロセッサによって現在使用中のデフォルト丸めモードをオーバーライドしうる。そのようなオーバーライドの場合には、直接データはさらに丸めモードの制御を提供しうる。さらに、直接データは精度例外 (precision exceptions) のオーバーライド (すなわち精度抑制 (precision suppression)) を提供しうる。こうして、直接データは、特定の丸め処理の非固定制御 (non-sticky control) を提供し、それにより該処理が最小限のサイクル数で実行されうるようにするために使用されうる。これがそうでありうるのは、命令に付随して受信された直接データが丸め制御情報を含むとき、構成設定レジスタ内に存在するそのような情報を更新する必要がないことがありうるからである。ここで、構成設定レジスタは、拡張された制御・状態レジスタ (CSR: control and status register)、たとえばインテル (登録商標) アーキテクチャ (たとえばIA-32アーキテクチャ) に従ってプロセッサ内に存在するマルチメディア拡張CSR (MXCSR: multimedia extension CSR) のようなものである。しかしながら、諸実施形態はさまざまなプロセッサ型において使用されうこと、本発明の範囲がこの点で限定されるものではないことを理解されたい。

#### 【0009】

ここで図1を参照すると、本発明のある実施形態に基づく方法の流れ図が示されている。図1に示されるように、方法100は、プロセッサ内で丸め命令および付随する直接データを受信することによって始まりうる(ブロック110)。たとえば、多くの実装において、ユーザー・レベル命令、たとえばISAベースの命令がプロセッサにおいて受信されうる。該命令に加え、それに付随する直接データが提供されうる。のちにさらに述べるように、そのような直接データは、処理のさまざまな側面を制御するための複数のフィールドを含みうる。

#### 【0010】

引き続き図1を参照すると、制御はブロック110から菱形115に進む。菱形115では、直接データが構成設定レジスタの丸めモードをオーバーライドするかどうかが決定される。すなわち、直接データのフィールドが、デフォルト丸めモードがオーバーライドされるべきかどうかを示すオーバーライド指標を含んでいてもよい。さまざまな実施形態において、そのようなデフォルト丸めモードは、CSR、たとえばMXCSRのような構成設定レジスタのフィールド内に存在していてもよい。ただし、本発明の範囲はこの点に関して限定されるものではない。直接データがオーバーライド指標を含む場合、制御はブロック120に進む。ブロック120では、前記命令によって指定されたソース・オペランドが、プロセッサのたとえば浮動小数点ユニット(FPU)に送付されうる。さらに、ソース・オペランドは、丸め処理の丸めモードを制御するための情報とともに送付されうる。制御情報は直接データから、すなわち直接データの丸めモード・フィールドに記述されているとおりに得られてもよい。のちにさらに述べるように、いくつかの実装では、プロセッサの制御選択器ユニットのような制御ユニットが前記命令および前記直接データを受信し、前記直接データを復号してデフォルト丸めモードがオーバーライドされるべきかどうかを決定し、もしそうであれば、直接データに記述されている丸めモードを得る。

#### 【0011】

引き続き図1を参照すると、菱形115においてもし代わりに、直接データがオーバーライド指標を含まないと判定される場合、制御はブロック125に進む。ブロック125では、ソース・オペランドは、FPUにおける実行のために送付されうる。さらに、丸め処理が、たとえば構成設定レジスタに記述されているデフォルト丸めモードに基づいて実行されうる。

#### 【0012】

いずれにせよ、ブロック120および125のどちらからも、制御は、丸め処理が実行されうるブロック130に進む。丸め処理は、入力(すなわちソース・オペランド)の端数精度を丸めモードに従って除去する。さまざまな実施形態において、丸め処理を実行する種々の仕方が実現されうる。たとえば、多くの実装では、丸め処理を実行するために、

10

20

30

40

50

FPUが加算器および丸めユニットを含みうる。IEEE規格754に従った丸めモードを実行するために、加算器は第一のオペラントとしてのソース・オペラントと、第二のオペラントについての定数値、たとえば0とを提供されうる。加算器の出力は次いで丸めユニットに供給され、該丸めユニットが前記の結果を選択された動作モードに従って丸めうる。こうして、丸めユニットは、その入力値を丸めて整数値をとる浮動小数点結果 (integral valued floating point result) にしうる。

#### 【0013】

他の諸実施形態では、IEEE規格754の丸めモードに加えて、追加的な丸めモードが実行されうる。そのような実装では、FPU加算器はソース・オペラントと、第二のオペラントとしての、ソース・オペラントの値および丸めモードに基づく特定のデータ値を供給されうる。これについてはのちにさらに述べる。次いで、丸め処理が前記結果に対して実行されうる。ここで、丸め処理はIEEE規格754の処理でもよい。拡張丸めモードのさらに別の実装では、ソース・オペラントおよび0の値がFPU加算器への入力に与えられてもよく、結果として得られる値が次いで、丸めユニットに送られた制御情報に従って丸められてもよい。

10

#### 【0014】

実行後、丸め処理の結果は行き先 [デスティネーション] オペラントに記憶されてもよい (ブロック140)。さまざまな実施形態において、行き先オペラントはプロセッサの拡張されたメモリ・レジスタであってよい。ただし、本発明の範囲はそのように限定されるものではない。さらに、丸め処理の間に精度例外が発生したかどうかが判定されてもよい (菱形145)。すなわち、丸め処理が例外を起こす不正確な結果を発展させたかどうかが判定されうる。もしそうでなければ、方法100は終結しうる。

20

#### 【0015】

もしその代わり、精度例外が起こされている場合、制御は菱形150に進みうる。菱形150では、直接データが精度例外を抑制するフィールドを含むかどうかが判定されうる。すなわち、いくつかの実装では、直接データが抑制フィールドを含んでいてもよい。このフィールドの値が、精度例外が持ち上がった場合に関連する丸め命令が精度例外を抑制すべきかどうかを示しうる。精度抑制指標が存在する場合、たとえ精度例外が発生しても、さらなる行動は取られず、方法100は終結しうる。もしその代わり、直接データが精度例外を抑制する指標を含まない場合、制御はブロック160に進む。ブロック160において、精度例外フラグが状態レジスタ中にセットされうる。たとえば、いくつかの実装では、状態レジスタはMXCSRに対応しうる。ただし、本発明の範囲はこの点に関して限定されるものではない。状態レジスタ内のこのフラグの状態に基づいて、精度例外が持ち上げられ (raised) うる (たとえば、フラグがマスクされていない場合)。もしそうであれば、その例外に対処するために、たとえばソフトウェア・ハンドラを介した適切な処置が実行されうる。もしその代わりにフラグがマスクされている場合、たとえ精度例外が起こって、状態レジスタにフラグが立っていても、そのセットされたフラグに関して何の行動も取られなくてもよい。図1の実施形態におけるこの特定の実装を用いて記載されているが、本発明の範囲はこの点に関して限定されるものではないことは理解しておくべきである。

30

#### 【0016】

ここで図2を参照すると、本発明のある実施形態に基づくプロセッサの一部分のブロック図が示されている。図2に示されるように、プロセッサ200は、レジスタ205 (これは汎用プロセッサ・レジスタであってもよい) からの、たとえばマイクロ処理 ( $\mu$ op : micro-operation) によって生成された、はいってくる命令情報と、それに付随する直接データ (immediate data) とを受領するよう結合されている制御選択器ユニット210を含みうる。マイクロ処理は、所与の丸め処理を実行するためにISAの単一命令に応じて生成されうる。さまざまな実施形態において、ハードウェア、ファームウェアまたはそれらの組み合わせで実装されうる制御選択器ユニット210が直接データを復号する。直接データに基づいて、たとえば現在の丸め制御状態220を記憶する制御もしくは構成設定

40

50

レジスタにおいて表されているようなプロセッサの現在の丸めモードがオーバーライドされるべきかどうかが決定されうる。もしそうであれば、制御選択器ユニット210は直接データのモード・フィールド、つまり丸めモード・フィールドを復号して適正な丸めモードを決定しうる。

#### 【0017】

制御選択器ユニット210は、浮動小数点ユニット(FPU)240に結合されていて、はいってくる情報に基づいて該ユニットに制御命令を与えてよい。図2にさらに示されるように、いわゆる拡張された(XMM)レジスタ230のような拡張されたレジスタ・ファイルがプロセッサ200内に存在していてよい。それは、命令において丸め処理のためのソース・オペランドおよび行き先オペランドとしてはたらくよう指定されているレジスタを含みうる。このように、XMMレジスタ230は、FPU240にソース・オペランドを与える、FPU240から行き先オペランドを受領するためにFPU240に結合されうる。10

#### 【0018】

さまざまな実施形態において、FPU240はデータに対して処理を実行するさまざまな回路を含みうる。図2の実施形態では、FPU240はFPU加算器242を含む。具体的には、図2に示されるように、FPU加算器242は、はいてくるオペランド、たとえば第一のソース・オペランドおよび第二のソース・オペランド(すなわちオペランドS1およびS2)を受領するよう結合されうる。FPU240はまた、FPU加算器242の出力に結合されたFPU丸め器244を含んでいてよい。さまざまな実施形態において、FPU加算器242は無限精度をもつ演算結果を生成してもよい。しかしながら、記憶および他の制約から、結果は丸められて、所望のフォーマットの最終結果を、たとえば単精度または倍精度の浮動小数点要素を与えてよい。したがって、FPU丸め器244は、無限に精密な結果をFPU加算器242から受領し、プロセッサ200の現在の丸めモードによって指定される、あるいは命令と一緒にすなわち制御選択器ユニット210を介して得られた直接データからの制御に基づく、丸め処理を実行しうる。FPU丸め器244は、一般に、FPU加算器242における数学的演算の結果として発生する限りなく精密な結果を受領してもよいが、さまざまな実装では、丸め命令と一緒に与えられるソース・オペランドはすでに限られた精度のフォーマットにあってもよい。これらの例において、FPU丸め器244はその入力値(たとえば、与えられた丸め命令のソース・オペランドに対応する)を受領し、たとえば最も近い整数値に対応する丸められた結果を生成しうる。20

#### 【0019】

こうして、与えられた丸め命令に基づいて、FPU240は、たとえばXMMレジスタ230の一つからの与えられたソース・オペランドに対して、制御選択器ユニット210からの情報によって制御されるように丸め処理を実行しうる。さらに、丸め処理の完了に際し、結果はXMMレジスタ230内のたとえば別のレジスタに記憶されてもよい。処理の間に精度例外が発生する場合、通常、FP状態レジスタ225にそのことを示すフラグがセットされうる。しかしながら、さまざまな実施形態において、丸め命令に関連付けられた直接データが精度抑制を示す場合、そのようなフラグはセットされなくてもよい。図2の実施形態ではこの特定の実装について記載されているが、本発明の範囲がこの点に関して限定されるものではないことは理解しておくべきである。たとえば、いくつかの実施形態では、たとえば丸め制御状態(rounding control state)220およびFP状態レジスタ(FP status register)225によって表されるような、制御および状態の状態(control and status state)が、MXCSRのような单一のCSR内に記憶されてもよい。40

#### 【0020】

直接データは、さまざまな形で制御選択器ユニット210に与えられうることを注意しておく。たとえば、いくつかの実装では、直接データは單一バイトのデータ要素の形であってもよい。ただし、本発明の範囲はそれに限定されるものではない。さらに、直接データ要素内に制御情報をエンコードするさまざまな仕方が実現されうる。ここで図3を参照すると、本発明のある実施形態に基づく直接データ要素のプロック図が示されている。図3に示されるように、直接データ要素300は、オーバーライド指標310、モード制御50

フィールド320、精度オーバーライド指標330およびリザーブ・フィールド340を含む8ビット語であってもよい。図3の実施形態ではこの特定の実装について示しているが、本発明の範囲はこのように限定されるものではない。

## 【0021】

図3の実施形態において、オーバーライド指標310は、直接データ要素300に関連付けられた丸め命令のオーバーライド状態を決定するために使用されうる。下記の表1に示されるように、オーバーライド指標310は、デフォルトの丸めモード(MXCSRのような構成設定レジスタによって表現されるような)のオーバーライドを示すために論理的な低レベルにセットされうる。論理的な高い値はデフォルト・モードの使用を示す。

## 【0022】

## 【表1】

10

表1

丸めモード・オーバーライド指標
0: 直接のビット1:3を使用
1: デフォルト・丸めモードを使用

オーバーライド指標310がデフォルト丸めモードがオーバーライドされるべきであることを示す場合、丸め命令に関連付けられる丸めモードを決定するために丸めモード・フィールド320が復号されうる。下記の表2に示されるように、いくつかの実装では、IEEE規格754によって規定される4つの丸めモードを2つの拡張丸めモードとともに含む6つの丸めモードがサポートされうる。拡張丸めモードについてはのちにさらに論じる。

20

## 【0023】

## 【表2】

30

表2

丸めモード・フィールド
000: 最も近い偶数
001: $-\infty$ のほうに
010: $+\infty$ のほうに
011: 打ち切り(零への丸め)
100: 中間点で零から遠いほうへの
101: 零から遠いほうへの丸め

直接データ要素300はさらに、精度抑制指標330を含む。この指標は、たとえ関連する命令の処理の間に精度例外が起こったとしても精度例外が状態レジスタ内の例外フラグのセットを引き起こさないようにする、不正確な結果の許容を示すためにセットされうる。

40

## 【0024】

具体的には、下記の表3に示されるように、精度抑制指標330は次の形を取りうる。

## 【0025】

## 【表3】

表3

精度抑制指標
1: 不正確（精度）フィールドが更新されない
0: 通常の振る舞い

精度抑制指標 330 はさまざまな言語、たとえばC99、フォートランおよびジャバのユーザー・レベル命令との関連で使用されうることを注意しておく。最後に、いくつかの実施形態では、リザーブ・フィールド 340 は追加的な情報のためにリザーブされうる。さらに、表1～表3に記載される個別的な値は、指標およびフィールドの特定の位置およびサイズとともに、限定されるものではなく、さまざまな変更、修正および拡張が本発明の範囲内であることを注意しておく。

## 【0026】

上記のように、多くの実装において、丸め処理は、ISAの単一命令に応じて実行されうる。このようにして、ユーザー・レベルのサポートが提供され、丸め処理は効率的に実行できる。所与のISAにおいて、複数のそのような丸め命令が存在し、個別的な丸め処理を扱うために利用可能であることがある。たとえば、倍精度および单精度の浮動小数点値やパックされた値およびスカラー値の丸めといったものである。これらの丸め命令は、浮動小数点データ要素の端数部分を切り捨てるために使用されてもよい。ISAレベルの命令の存在に加えて、直接データまたは他の制御フィールド情報のおかげで、プロセッサの現在のデフォルト状態を修正する必要なしに、（他の属性のなかでも）丸めモードの効率的なローカル制御が許容されうる。

## 【0027】

下記の表4に示されるように、さまざまな型のデータ要素に対する効率的な丸め処理を可能にするために、ISAには丸め命令のさまざまな種類が存在しうる。

## 【0028】

10

20

【表4】

表4

命令	説明
ROUNDPD xmm1, xmm2/m128, imm8	xmm2/m128 内のパックされた倍精度浮動小数点値を丸め、結果を xmm1 に入れる。丸めモードは imm8 によって決定される。
ROUNDPS xmm1, xmm2/m128, imm8	xmm2/m128 内のパックされた単精度浮動小数点値を丸め、結果を xmm1 に入れる。丸めモードは imm8 によって決定される。
ROUNDSD xmm1, xmm2/m64, imm8	xmm2/m64 内の低いパックされた倍精度浮動小数点値を丸め、結果を xmm1 に入れる。丸めモードは imm8 によって決定される。
ROUNDSS xmm1, xmm2/m32, imm8	xmm2/m32 内の低いパックされた単精度浮動小数点値を丸め、結果を xmm1 に入れる。丸めモードは imm8 によって決定される。

10

20

これらのISAベース命令がどのように機能するかの例として、ROUNDPD命令は、直接要素(immediate element)（すなわちIMM8）において指定されている丸めモードによってソース・オペランド（すなわち、XMMレジスタまたはメモリから得られる第二のオペランド）内の2つの倍精度浮動小数点値を丸め、結果を行き先オペランド（すなわち、XMMレジスタであってもよい第一のオペランド）に入れるために使用される。直接要素は、丸め処理のための制御フィールドを指定しうる。先の表1～表3を参照すると、直接データのビット4（すなわち図3の指標330）が精度例外についてのプロセッサの振る舞いを制御しうる。一方、ビット0（すなわち図3の指標310）は丸めモード制御のソースを選択しうる。最後に、ビット3:1（すなわち、図3のフィールド320）は非固定丸めモード値を指定しうる。いくつかの実施形態では、いずれかのソース・オペランドがシグナル発生非数(SNaN: signaling not a number)であれば、それは静穏NaN(QNaN)に変換されることを注意しておく。構成設定レジスタが非正規化数を0とみなす(DAZ: denormals as zeros)ようセットされている場合、非正規化数は丸めの前に0に変換されうる。構成設定レジスタが非正規化数を0にフラッシュする(FTZ: flush denormals to zeros)ようセットされている場合、非正規化数は丸めの後で0に変換されうる。

30

## 【0029】

これらのISA命令がどのように機能しうるかのさらなる例として、ROUNDPS命令は、ソース・オペランド内の4つのパックされた単精度浮動小数点値を丸め、結果を行き先オペランドに入れるために使用される。例示の目的のため、具体的な丸め命令は次の形を取りうる。

40

## 【0030】

ROUNDPS xmm0, xmm1, imm8 (最も近い整数への丸め)

この命令は、第一のレジスタ、すなわちxmm1内のパックされた単精度値を取り、各値を直接データ（すなわちimm8）の丸めモードによって記述されるように最も近い整数値に丸め、結果を第二のレジスタ、すなわちxmm0に記憶しうる。下記の表5は、ソース・オペランド（すなわちxmm1）内に存在する、それぞれ限られた精度の浮動小数点値に対応する代表的な値と、行き先オペランド（すなわちxmm0）に記憶される、整数値をとる浮動小数点数

50

、つまり元のソース値に最も近い整数値に対応する、結果として得られる丸められた値とを示している。

【 0 0 3 1 】

【表5】

表5

1.01f	2.9f	3.6f	4.2f	xmm1
-------	------	------	------	------

1.0f	3.0f	4.0f	4.0f	xmm0
------	------	------	------	------

10

さらなる実装では、丸め処理は、ソースFP値から整数値（すなわち、整数値をとるFP値ではなく）を生成する命令に応じてであってもよいことを注意しておく。他の実装は、より低い精度の浮動小数点表現への丸めを可能にしうる。よって、諸実施形態は、構成設定レジスタ内のデフォルト丸めモードまたは命令に関連付けられた直接データに記述されているローカル丸めモードのいずれかによって制御される、標準的な丸めモードまたは特殊化された丸めモードに従ってソース値を丸める効率的な手段を提供しうる。

20

【 0 0 3 2 】

さまざまな実施形態において、直接データは、IEEE規格754の丸め処理とは異なる丸めモードを実行するような制御情報を提供しうる。これらの丸めモードは、中間点で零から遠いほうへの丸め (round halfway away from zero) および零から遠いほうへの丸め (round away from zero) の丸めモードを含みうる。ここで図4を参照すると、本発明のある実施形態に基づく丸め処理を実行する方法の流れ図が示されている。図4に示されるように、方法400は、これらの拡張された丸めモードを実行するために使用される。方法400は、ソース・オペランドが0以下であるかどうかを判定することによって始まりうる(菱形410)。もしそうであれば、制御はブロック420に進み、所定の値がソース・オペランドから減算されうる(ブロック420)。たとえば、FP加算器が、選択されている特定の丸めモードに基づいて、ソース・オペランドから所与の値を減算しうる。もちろん、この減算は、前記所定の値についての負の値との加算として実行されうる。次いで、選択された丸め処理が、このFP加算の結果に対して実行されうる(ブロック430)。いくつかの実装では、打ち切り(truncate)(零への丸め(round to zero)とも呼ばれる)のようなIEEE規格754丸め処理が前記結果に対して実行されて拡張された丸めモードの結果が得られるのでもよい。もしその代わりに菱形410においてソース・オペランドが0より大きいと判定される場合、制御はブロック440に進む。ブロック440では、所定の値(これは上記と同じ値であってもよい)がFP加算器においてソース・オペランドに加算されてもよい。次いで、ブロック450において、選択された丸め処理が前記結果に対して実行され、結果としての丸められた値が得られる。

30

【 0 0 3 3 】

本発明の範囲はこの点に関して限定されるものではないが、中間点で零から遠いほうへの丸めの処理は、その所定の値として0.5の値を使ってもよい。一方、零から遠いほうへの丸めの処理は、 $1^-$ を使ってもよい。ここで、 $1^-$ とは、1より小さく1に等しくない、最も近い表現可能なFP値に対応する。単精度および倍精度のFP値について、0.5はそれぞれ0x3f000000および0x3fe00000000000000に対応しうる。単精度および倍精度のFP値について、-0.5はそれぞれ0xbff000000および0xbfe000000000000に対応しうる。単精度および倍精度のFP値について、 $1^-$ はそれぞれ0x3f7fffffおよび0x3fefffffffffffffに対応しうる。単精度および倍精度のFP値について、 $-1^-$ はそれぞれ0xbff7fffffおよび0xbfefffffffffffffに対応しうる。下記の表6には、これらの処理を実行するためのソース・コードの例が示さ

40

れている。

### 【0034】

#### 【表6】

表6

<pre> ROUND_HALF_AWAY_ZERO(A): IF (A &lt;= 0)     A←ROUND_TOWARD_ZERO(A - 0.5) ELSE IF (A &gt; 0)     A←ROUND_TOWARD_ZERO(A + 0.5) </pre> <p>2) Round Away From Zero (A). 「結果は、限りなく精密な結果より絶対値が小さくなく、かつ最も近い値となる」</p> <pre> IF (A &lt;= 0)     A←ROUND_TOWARD_ZERO(A - 1) ELSE IF (A &gt; 0)     A←ROUND_TOWARD_ZERO(A + 1) </pre>	10
--	----

<pre> ROUND_HALF_AWAY_ZERO(A): IF (A &lt;= 0)     A←ROUND_TOWARD_ZERO(A - 0.5) ELSE IF (A &gt; 0)     A←ROUND_TOWARD_ZERO(A + 0.5) </pre> <p>2) Round Away From Zero (A). 「結果は、限りなく精密な結果より絶対値が小さくなく、かつ最も近い値となる」</p> <pre> IF (A &lt;= 0)     A←ROUND_TOWARD_ZERO(A - 1) ELSE IF (A &gt; 0)     A←ROUND_TOWARD_ZERO(A + 1) </pre>	20
--	----

これらの例において、処理ROUND\_TOWARD\_ZEROはIEEE規格754の打ち切り処理であり、加算／減算の結果に対して実行される。これらの拡張された丸めモードの処理を実行する際、前記所定の値は第二のソース・オペランドとして（たとえば図2の実施形態におけるS2のように）FP加算器に与えられてもよいことを注意しておく。あるいはまた、実施形態によっては、他の丸め処理と同様に、第二のソース・オペランドは0であってもよく、選択された拡張丸めモード処理を実施するために制御信号が丸めユニットに送られてもよい。

### 【0035】

このように、さまざまな実施形態において、丸めを実行するための向上が実現されうる。これらの向上は、制御レジスタの状態を保存し、ダミーFP演算を実行し、前記状態を再セットする、またさらには数を整数に変換して浮動小数点に戻す近似的な単純化といったさまざまな処理を実行する必要を回避しうる。不正確な精度例外を抑制することにより、準拠性が異なる言語の丸めのサポートが簡略化されうる。一方、実装は、たとえばC99言語における、ある種の丸め機能についての標準的な丸めモードに従ってもよい。

### 【0036】

諸実施形態は、多くの異なるシステム型において実装されうる。ここで図5を参照すると、本発明のある実施形態に基づくシステムのブロック図が示されている。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・ポイントの相互接続550を介して結合された第一のプロセッサ570および第二のプロセッサ580を含む。図5に示されるように、プロセッサ570および580のそれぞれは、第一および第二のプロセッサ・コア（すなわち、プロセッサ・コア574aおよび574bならびにプロセッサ・コア584aおよび584b）を含むマルチコア・プロセッサであってもよい。コアのそれぞれが、本発明の実施形態に基づいて、ISAレベルの命令に応じて丸め処理を実行しうることを注意しておく。

### 【0037】

第一のプロセッサ570はさらに、ポイント・ツー・ポイントの（P-P）インターフェ

30

40

50

ース 576 および 578 を含む。同様に、第二のプロセッサ 580 は P-P インターフェース 586 および 588 を含む。図 5 に示されるように、メモリ・コントローラ・ハブ (MCH)

572 および 582 はプロセッサをそれぞれのメモリ、すなわちメモリ 532 およびメモリ 534 に結合させる。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・メモリの部分であってもよい。

#### 【0038】

第一のプロセッサ 570 および第二のプロセッサ 580 は、それぞれ P-P 相互接続 552 および 554 を介してチップセット 590 に結合されうる。図 5 に示されるように、チップセット 590 は P-P インターフェース 594 および 598 を含む。さらに、チップセット 590 は、チップセット 590 を高性能グラフィック・エンジン 538 と結合するインターフェース 592 を含む。ある実施形態では、グラフィック・エンジン 538 をチップセット 590 に結合させるために、先進グラフィック・ポート (AGP: Advanced Graphics Port) バス 539 が使用されうる。AGP バス 539 は、米国カリフォルニア州サンタクララのインテル・コーポレーションによって 1998 年 5 月 4 日に発表された、加速されたグラフィック・ポート・インターフェース (Accelerated Graphics Port Interface) 規格、第 2.0 版に準拠してもよい。あるいはまた、ポイント・ツー・ポイント 相互接続 539 がこれらのコンポーネントを結合してもよい。

#### 【0039】

次いで、チップセット 590 はインターフェース 596 を介して第一のバス 516 に結合されうる。ある実施形態では、第一のバス 516 は、PCI ローカル・バス規格、プロダクション・バージョン、第 2.1 版、1995 年 6 月、によって定義されるような周辺コンポーネント相互接続 (PCI: Peripheral Component Interconnect) バスまたは PCI エクスプレス (PCI Express (商標)) バスまたは他の第三世代の入出力 (I/O) 相互接続バスであってもよい。ただし、本発明の範囲はそのように限定されるものではない。

#### 【0040】

図 5 に示されるように、さまざまな I/O 装置 514 が、第一のバス 516 を第二のバス 520 に結合するバス・ブリッジ 518 とともに、第一のバス 516 に結合されうる。ある実施形態では、第二のバス 520 は低ピン数 (LPC: low pin count) バスであってもよい。第二のバス 520 にはさまざまな装置が結合されうる。それには、ある実施形態ではたとえば、キーボード / マウス 522、通信装置 526 およびコード 530 を含みうるディスク・ドライブまたは他の大容量記憶装置といったデータ記憶ユニット 528 が含まれる。さらに、オーディオ I/O 524 が第二のバス 520 に結合されていてもよい。他のアーキテクチャが可能であることを注意しておく。たとえば、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。

#### 【0041】

実施形態は、コードにおいて実装されてもよく、記憶媒体上に記憶されてもよい。記憶媒体がその上に記憶された命令を有し、該命令が、システムが該命令を実行するようプログラムするために使用されるのである。記憶媒体は、これに限られないが、フロッピー (登録商標) ディスク、光ディスク、コンパクト・ディスク読み出し専用メモリ (CD-ROM)、書き換え可能型コンパクト・ディスク (CD-RW) および光磁気ディスクを含む任意の型のディスク、読み出し専用メモリ (ROM)、動的ランダム・アクセス・メモリ (DRAM)、静的ランダム・アクセス・メモリ (SRAM) といったランダム・アクセス・メモリ (RAM)、消去可能なプログラム可能型読み出し専用メモリ (EPROM)、フラッシュメモリ、電気的に消去可能なプログラム可能型読み出し専用メモリ (EEPROM) といった半導体デバイス、磁気カードもしくは光カードまたは電子的な命令を記憶するために好適な他の任意の型の媒体を含みうる。

#### 【0042】

本発明について、限られた数の実施形態に関して記載してきたが、当業者はそうした実

10

20

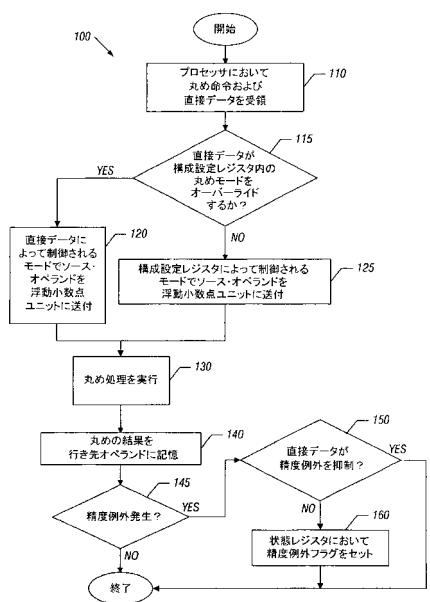
30

40

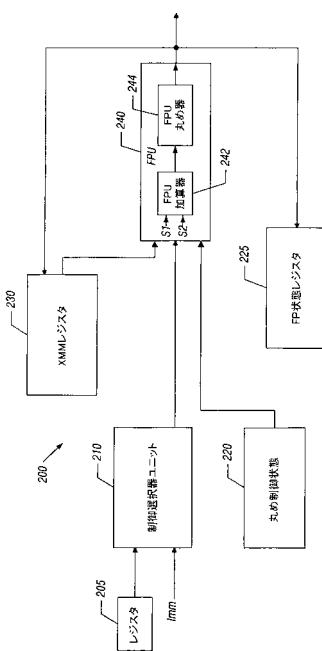
50

施形態からの数多くの修正および変形を認識するであろう。付属の請求項は本発明の真の精神および範囲にはいるそのようなすべての修正および変形をカバーすることが意図されている。

【図1】



【図2】



【 四 3 】

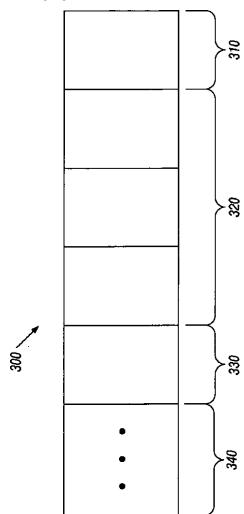
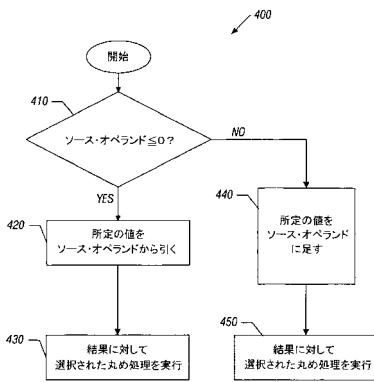
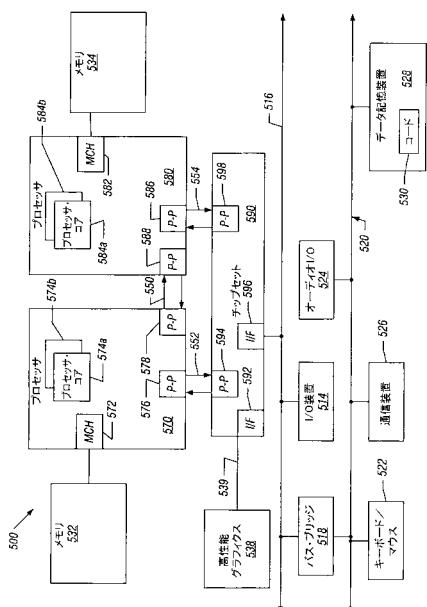


FIG. 3

【 図 4 】



【 四 5 】



---

フロントページの続き

(72)発明者 ストーリー, シエイン

アメリカ合衆国 97229 オレゴン州 ポートランド ノースウェスト 102ンド アヴェ  
ニュー 1595

審査官 緑川 隆

(56)参考文献 米国特許第05696709(US, A)

米国特許第06058410(US, A)

特開平09-128217(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 7/38