

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2008-546121

(P2008-546121A)

(43) 公表日 平成20年12月18日(2008.12.18)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 9/455 (2006.01)	G06F 9/44 310A	5B022
G06F 7/00 (2006.01)	G06F 7/00 201	5B033
G06F 9/30 (2006.01)	G06F 9/30 310A	

審査請求 有 予備審査請求 未請求 (全 18 頁)

(21) 出願番号 特願2008-516047 (P2008-516047)  
 (86) (22) 出願日 平成18年6月27日 (2006.6.27)  
 (85) 翻訳文提出日 平成19年12月5日 (2007.12.5)  
 (86) 国際出願番号 PCT/US2006/025321  
 (87) 国際公開番号 W02007/002809  
 (87) 国際公開日 平成19年1月4日 (2007.1.4)  
 (31) 優先権主張番号 11/166,628  
 (32) 優先日 平成17年6月27日 (2005.6.27)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 591003943  
 インテル・コーポレーション  
 アメリカ合衆国 95052 カリフォル  
 ニア州・サンタクララ・ミッション カレ  
 ッジ ブレーバード・2200  
 (74) 代理人 100104156  
 弁理士 龍華 明裕  
 (72) 発明者 バーコウィツ、シオン  
 イスラエル国、32546 ハイファ、ア  
 パートメント ナンバー 5、ブレンナー  
 ストリート ナンバー 1  
 (72) 発明者 エツィオン、オルナ  
 イスラエル国、34761 ハイファ、カ  
 ーリイ ストリート 5

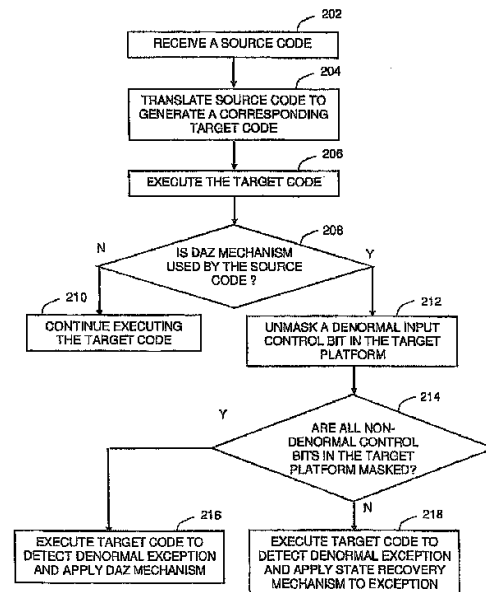
最終頁に続く

(54) 【発明の名称】 デノーマル入力ハンドリングメカニズムをサポートする動的バイナリ変換の装置、システム、及び方法

(57) 【要約】

本発明の一実施の形態は、一組のソース命令を一組のターゲット命令に変換し、その一組のターゲット命令を実行し、且つ、一組のソース命令がデノーマル入力ハンドリングメカニズムを使用する場合には、デノーマル入力制御ビットをマスク解除する方法を提供する。本発明の別の実施の形態は、一組のターゲット命令を実行することによって、障害のあるターゲット命令の少なくとも1つのデノーマル例外を検出し、障害のあるターゲット命令の1つ又は2つ以上のデノーマルオペランドに所定の値を代入し、且つ、1つ又は2つ以上のデノーマルオペランドについて所定の値を有する障害のあるターゲット命令を実行する方法を提供する。また、本発明の実施の形態は、その装置、システム、及びマシン可読媒体も提供する。

【選択図】 図2



## 【特許請求の範囲】

## 【請求項 1】

一組のソース命令を一組のターゲット命令に変換すること、  
前記一組のターゲット命令を実行すること、及び  
前記一組のソース命令がデノーマル入力ハンドリングメカニズムを使用する場合には、  
デノーマル入力制御ビットをマスク解除すること  
を含む、方法。

## 【請求項 2】

前記一組のターゲット命令を実行することによって、障害のあるターゲット命令の少なくとも 1 つのデノーマル例外を検出すること、

10

1 つ又は 2 つ以上の非デノーマル例外に対応する 1 つ又は 2 つ以上の制御ビットがマスクされている場合には、前記障害のあるターゲット命令の 1 つ又は 2 つ以上のデノーマルオペランドに所定の値を代入すること、及び

前記 1 つ又は 2 つ以上のデノーマルオペランドについて前記所定の値を有する前記障害のあるターゲット命令を実行すること  
を含む、請求項 1 に記載の方法。

## 【請求項 3】

前記障害のあるターゲット命令の前記実行から得られた結果を適用することによって、前記障害のあるターゲット命令に続くターゲット命令の実行を再開すること  
を含む、請求項 2 に記載の方法。

20

## 【請求項 4】

前記障害のあるターゲット命令に続くターゲット命令の実行を再開することは、  
前記障害のあるターゲット命令を指し示すことから、実行される後続のターゲット命令を指し示すことへ、命令ポインタを移動すること、及び  
前記後続のターゲット命令を実行すること  
を含む、請求項 3 に記載の方法。

## 【請求項 5】

前記所定の値は 0 である、請求項 2 に記載の方法。

## 【請求項 6】

前記一組のターゲット命令を実行することによって、障害のあるターゲット命令の少なくとも 1 つのデノーマル例外を検出すること、

30

1 つ又は 2 つ以上の非デノーマル例外に対応する 1 つ又は 2 つ以上の制御ビットがマスク解除されている場合には、前記障害のあるターゲット命令の変換元のソース命令を回復すること、及び

前記ソース命令を再生成されるターゲット命令に再変換すること  
を含む、請求項 1 に記載の方法。

## 【請求項 7】

前記一組のソース命令が前記デノーマル入力ハンドリングメカニズムを使用するか否かを判断すること、

前記デノーマル入力ハンドリングメカニズムが使用される場合には、前記デノーマル例外に対する入力である 1 つ又は 2 つ以上のデノーマルオペランドを識別すること、及び

40

前記 1 つ又は 2 つ以上のデノーマルオペランドに所定の値を代入すること  
を含む、請求項 6 に記載の方法。

## 【請求項 8】

前記所定の値は 0 である、請求項 7 に記載の方法。

## 【請求項 9】

ターゲット命令ブロックの実行中にデノーマル例外の個数をカウントすること、及び  
前記デノーマル例外の前記個数が所定のしきい値に達する場合には、前記ターゲット命令ブロックを書き換えることであって、それによって、1 つ又は 2 つ以上の組み込み命令を含める、書き換えること

50

を含む、請求項 1 に記載の方法。

【請求項 1 0】

前記 1 つ又は 2 つ以上の組み込み命令のうちの少なくとも 1 つを実行することによって、それによって、ターゲット命令が前記デノーマル例外を起こすか否かを判断する、実行すること、

前記ターゲット命令が前記デノーマル例外を起こす場合には、前記ターゲット命令の 1 つ又は 2 つ以上のデノーマルオペランドを識別すること、

前記 1 つ又は 2 つ以上のデノーマルオペランドに 0 の値を代入すること、及び

前記代入された 0 の値のデノーマルオペランドを有する前記ターゲット命令を実行すること

10

を含む、請求項 9 に記載の方法。

【請求項 1 1】

前記デノーマル入力ハンドリングメカニズムは、D A Zメカニズムである、請求項 1 に記載の方法。

【請求項 1 2】

一組のソース命令を一組のターゲット命令に変換し、前記一組のソース命令がデノーマル入力ハンドリングメカニズムを使用する場合には、デノーマル入力制御ビットをマスク解除する動的バイナリ変換器と、

前記一組のターゲット命令を実行して、障害のあるターゲット命令の少なくとも 1 つのデノーマル例外を検出するプロセッサと

20

を備える、装置。

【請求項 1 3】

前記動的バイナリ変換器は、前記障害のあるターゲット命令の 1 つ又は 2 つ以上のデノーマルオペランドに所定の値を代入する例外ハンドラを備え、前記プロセッサは、前記 1 つ又は 2 つ以上のデノーマルオペランドについて前記所定の値を有する前記障害のあるターゲット命令を実行する、請求項 1 2 に記載の装置。

【請求項 1 4】

前記プロセッサは、前記障害のあるターゲット命令の前記実行から得られた結果を適用することによって、前記障害のあるターゲット命令に続くターゲット命令の実行を再開する、請求項 1 3 に記載の装置。

30

【請求項 1 5】

前記プロセッサは、前記障害のあるターゲット命令を指し示すことから、実行される後続のターゲット命令を指し示すことへ、命令ポインタを移動することによって、前記障害のあるターゲット命令に続くターゲット命令の実行を再開する、請求項 1 4 に記載の装置。

【請求項 1 6】

前記動的バイナリ変換器は、前記障害のあるターゲット命令の変換元のソース命令を回復すると共に前記ソース命令を再生成されるターゲット命令に再変換する例外ハンドラを備える、請求項 1 2 に記載の装置。

【請求項 1 7】

40

前記プロセッサは、前記一組のソース命令が前記デノーマル入力ハンドリングメカニズムを使用するか否かを判断し、前記デノーマル入力ハンドリングメカニズムが使用される場合には、前記デノーマル例外に対する入力である 1 つ又は 2 つ以上のデノーマルオペランドを識別し、且つ、前記 1 つ又は 2 つ以上のデノーマルオペランドに所定の値を代入する、請求項 1 6 に記載の装置。

【請求項 1 8】

前記プロセッサは、ターゲット命令ブロックの実行中にデノーマル例外の個数をカウントし、且つ、前記デノーマル例外の前記個数が所定のしきい値に達する場合には、前記ターゲット命令ブロックを書き換えて 1 つ又は 2 つ以上の組み込み命令を含める、請求項 1 2 に記載の装置。

50

**【請求項 19】**

前記プロセッサは、前記組み込み命令の少なくとも1つを実行して、ターゲット命令が前記デノーマル例外を起こすか否かを判断し、前記ターゲット命令が前記デノーマル例外を起こす場合には、前記障害のあるターゲット命令の1つ又は2つ以上のデノーマルオペランドを識別し、且つ、前記1つ又は2つ以上のデノーマルオペランドに所定の値を代入する、請求項18に記載の装置。

**【請求項 20】**

一組のソース命令と、一組のターゲット命令と、動的バイナリ変換器とを記憶するメモリであって、前記動的バイナリ変換器は、前記一組のソース命令を前記一組のターゲット命令に変換し、且つ、前記一組のソース命令がデノーマル入力ハンドリングメカニズムを使用する場合には、デノーマル入力制御ビットをマスク解除するようになっている、メモリと、

10

前記一組のターゲット命令を実行して、障害のあるターゲット命令の少なくとも1つのデノーマル例外を検出するプロセッサとを備える、システム。

**【請求項 21】**

前記動的バイナリ変換器は、前記障害のあるターゲット命令の1つ又は2つ以上のデノーマルオペランドに所定の値を代入する例外ハンドラを備え、前記プロセッサは、前記1つ又は2つ以上のデノーマルオペランドについて前記所定の値を有する前記障害のあるターゲット命令を実行する、請求項20に記載のシステム。

20

**【請求項 22】**

前記プロセッサは、前記障害のあるターゲット命令の前記実行から得られた結果を適用することによって、前記障害のあるターゲット命令に続くターゲット命令の実行を再開する、請求項21に記載のシステム。

**【請求項 23】**

前記動的バイナリ変換器は、前記障害のあるターゲット命令の変換元のソース命令を回復すると共に前記ソース命令を再生成されるターゲット命令に再変換する例外ハンドラを備える、請求項20に記載のシステム。

**【請求項 24】**

前記プロセッサは、前記一組のソース命令が前記デノーマル入力ハンドリングメカニズムを使用するか否かを判断し、前記デノーマル入力ハンドリングメカニズムが使用される場合には、前記デノーマル例外に対する入力である1つ又は2つ以上のデノーマルオペランドを識別し、且つ、前記1つ又は2つ以上のデノーマルオペランドに所定の値を代入する、請求項23に記載のシステム。

30

**【請求項 25】**

前記プロセッサは、ターゲット命令ブロックの実行中にデノーマル例外の個数をカウントし、且つ、デノーマル例外の前記個数が所定のしきい値に達する場合には、前記ターゲット命令ブロックを書き換えて1つ又は2つ以上の組み込み命令を含める、請求項20に記載のシステム。

**【請求項 26】**

前記プロセッサは、前記組み込み命令のうちの少なくとも1つを実行して、ターゲット命令がデノーマル例外を起こすか否かを判断し、前記ターゲット命令がデノーマル例外を起こす場合には、前記障害のあるターゲット命令の1つ又は2つ以上のデノーマルオペランドを識別し、且つ、前記1つ又は2つ以上のデノーマルオペランドに所定の値を代入する、請求項25に記載のシステム。

40

**【請求項 27】**

マシンによって実行されると、一組のソース命令を一組のターゲット命令に変換し、前記一組のターゲット命令を実行し、且つ、前記一組のソース命令がデノーマル入力ハンドリングメカニズムを使用する場合には、デノーマル入力制御ビットをマスク解除するという結果になる一組の命令が記憶されている、マシン可読媒体。

50

**【請求項 28】**

前記命令は、前記一組のターゲット命令を実行することによって、障害のあるターゲット命令の少なくとも1つのデノーマル例外を検出し、1つ又は2つ以上の非デノーマル例外に対応する1つ又は2つ以上の制御ビットがマスクされている場合には、前記障害のあるターゲット命令の1つ又は2つ以上のデノーマルオペランドに所定の値を代入し、且つ、前記1つ又は2つ以上のデノーマルオペランドについて前記所定の値を有する前記障害のあるターゲット命令を実行するという結果になる、請求項27に記載のマシン可読媒体。

**【請求項 29】**

前記命令は、前記一組のターゲット命令を実行することによって、障害のあるターゲット命令の少なくとも1つのデノーマル例外を検出し、1つ又は2つ以上の非デノーマル例外に対応する1つ又は2つ以上の制御ビットがマスク解除されている場合には、前記障害のあるターゲット命令の変換元のソース命令を回復し、且つ、前記ソース命令を再生成されるターゲット命令に再変換するという結果になる、請求項27に記載のマシン可読媒体。

10

**【発明の詳細な説明】****【背景技術】****【0001】**

たとえばIntel（登録商標）Architecture-32（IA-32）に準拠したいくつかのコンピュータプラットフォームは、プログラムコードの実行中にデノーマル入力又はデノーマルオペランドをハンドリングするための、デノーマルがゼロである（Denormals-Are-Zeroes）（DAZ）、として知られている機能をサポートすることができる。プラットフォームは、本明細書ではDAZビットと呼ぶ制御ビットを含むことができる。このDAZビットは、浮動小数点（FP）デノーマル例外状況下にある単一命令複数データ（SIMD）命令に対する応答を制御する。プログラムコードは、プラットフォームにおいてDAZビットを制御する1つ又は2つ以上の制御命令を有することができる。これらの制御命令は、DAZビットを、たとえば「1」及び「0」の2つの値の一方に設定することができる。DAZビットが「1」に設定されると、たとえば、DAZ機能をサポートするプラットフォーム上で実行されているプロセッサは、コード実行中にデノーマル例外状況を起こす命令を検出することができ、所定の値を有する、その命令のデノーマルオペランドを代入することができる。この命令は例外命令である。所定の値は、たとえば、元のオペランドの符号を有する0である。プロセッサは、次に、この新しい値のオペランドを有する例外命令を実行することができる。上述したデノーマルオペランドのハンドリングを、本明細書では、デノーマル入力ハンドリングメカニズム、すなわちDAZメカニズムと呼ぶ場合がある。

20

30

**【0002】**

動的バイナリ変換器は、ソースコードを、ターゲットプラットフォーム上で実行されるターゲットコードに変換することができる。このソースコードは、たとえば、DAZ機能をサポートするソースアーキテクチャに準拠したIA-32コードである。ターゲットプラットフォームは、浮動小数点モデルを有することができ、したがって、デノーマル例外等のストリーミングSIMD拡張命令（SSE）例外を起こすことができる場合がある。しかしながら、ターゲットプラットフォームは、デノーマル例外のデノーマル入力をハンドリングする際にDAZ機能をサポートしていない場合がある。このような1つのターゲットプラットフォームは、たとえば、Intel Itanium（商標）プロセッサファミリーのアーキテクチャに準拠したプラットフォームとすることができる。

40

**【発明の開示】****【0003】**

本発明とみなされる主題は、特に、明細書の結びの部分に指し示され、明確に請求されている。しかしながら、本発明の目的、機能、及び利点と共に、オペレーションの編成及び方法の双方に関する本発明は、以下の詳細な説明を参照して、この詳細な説明を添付図

50

面と共に読むと最も良く理解することができる。

【0004】

説明図を簡単且つ明瞭にするために、図に示した要素は、必ずしも一律の縮尺で描かれていないことが十分理解されよう。たとえば、要素のいくつかの寸法は、明瞭にするために、他の要素に比べて誇張されている場合がある。

【発明を実施するための最良の形態】

【0005】

以下の詳細な説明では、本発明の実施形態の徹底した理解を提供するために、多数の特定の詳細が述べられている。しかしながら、本発明の実施形態は、これらの特定の詳細がなくても実施できることが当業者には理解されよう。それ以外の場合には、本発明の実施形態を不明瞭にしないように、既知の方法及び手順は詳細に説明されていない。

10

【0006】

以下の詳細な説明のいくつかの部分は、コンピュータメモリ内のデータビット又はバイナリデジタル信号に対するオペレーションのアルゴリズム及びシンボル表現の観点で提示されている。これらのアルゴリズム的な説明及び表現は、データ処理技術の当業者が自身の作業の内容を他の当業者に伝えるのに使用する手段とすることができる。

【0007】

アルゴリズムは、ここでは、一般的に、所望の結果をもたらす行為又はオペレーションの自己矛盾のないシーケンスであると考えられる。これらの行為又はオペレーションは、物理量の物理的な操作を含む。必ずしもそうではないが、通例、これらの物理量は、記憶、転送、結合、比較、及びそれ以外の操作を行うことができる電気信号又は磁気信号の形態を取る。主として一般に使用されるこの理由から、これらの信号を、ビット、値、要素、シンボル、文字、用語、数字等と呼ぶことが時に便利であることが分かっている。しかしながら、これらの用語及び類似の用語のすべては、適切な物理量に関連付けられることになり、これらの量に適用される単なる便利なラベルにすぎないことが理解されるべきである。

20

【0008】

特に指定のない限り、以下の解説から明らかなように、この明細書の解説全体を通じて、「処理」、「計算」、「算出」、「決定(求める)」等の用語を利用することは、コンピューティングシステムのレジスタ及び/又はメモリ内の電子量等の物理量として表されるデータを操作し且つ/又はそのようなデータを、コンピューティングシステムのメモリ、レジスタ、若しくは他のこのような情報ストレージ、伝送デバイス、又は表示デバイス内の物理量として同様に表される他のデータに変換するコンピュータ若しくはコンピューティングシステム又は同様の電子コンピューティングデバイスの動作及び/又はプロセスを指すことが十分理解される。

30

【0009】

本発明のいくつかの実施形態は、たとえば、マシンによって実行されると、本発明の実施形態による方法及び/若しくはオペレーションをマシンに実行させる1つの命令若しくは一組の命令を記憶できるマシン可読媒体又はマシン可読製品を使用して実施することができる。このようなマシンには、たとえば、あらゆる適した処理プラットフォーム、コンピューティングプラットフォーム、コンピューティングデバイス、処理デバイス、コンピューティングシステム、処理システム、コンピュータ、プロセッサ等が含まれ得る。このようなマシンは、ハードウェア及び/又はソフトウェアのあらゆる適した組み合わせを使用して実施することができる。マシン可読媒体又はマシン可読製品には、たとえば、あらゆる適したタイプのメモリユニット、メモリデバイス、メモリ製品、メモリ媒体、ストレージデバイス、ストレージ製品、ストレージ媒体及び/又はストレージユニットが含まれることができ、たとえば、メモリ、着脱可能媒体又は着脱不能媒体、消去可能媒体又は消去不能媒体、書き込み可能媒体又は再書き込み可能媒体、デジタル媒体又はアナログ媒体、ハードディスク、フロッピーディスク、コンパクトディスク読み出し専用メモリ(CD-ROM)、記録可能コンパクトディスク(CD-R)、再書き込み可能コンパクトディ

40

50

スク(CD-RW)、光ディスク、磁気媒体、さまざまなタイプのデジタル多用途ディスク(DVD)、テープ、カセット等が含まれ得る。命令には、たとえば、ソースコード、コンパイラ型コード、インタプリタ型コード、実行可能コード、静的コード、動的コード等といったあらゆる適したタイプのコードが含まれ得る。命令は、たとえば、C、C++、Java、BASIC、Pascal、Fortran、Cobol、アセンブリ言語、マシンコード等のあらゆる適した高水準プログラミング言語、低水準プログラミング言語、オブジェクト指向型プログラミング言語、ビジュアルプログラミング言語、コンパイラ型プログラミング言語、及び/又はインタプリタ型プログラミング言語を使用して実施することができる。

#### 【0010】

本発明の実施形態は、本明細書では、オペレーションを実行するための装置を含むことができる。これらの装置は、所望の目的で特別に構築することもできるし、コンピュータに記憶されたコンピュータプログラムによって選択的にアクティブにされるか又は再構成される汎用コンピュータを含むこともできる。このようなコンピュータプログラムは、コンピュータ可読ストレージ媒体に記憶することができる。このコンピュータ可読ストレージ媒体は、フロッピーディスク、光ディスク、CD-ROM、光磁気ディスク等のあらゆるタイプのディスク、読み出し専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、電氣的プログラマブル読み出し専用メモリ(EPROM)、電氣的消去可能プログラマブル読み出し専用メモリ(EEPROM)、磁気カード若しくは光カード、又は電子的命令を記憶するのに適し且つコンピュータシステムバスに結合可能な他のあらゆるタイプの媒体であるが、これらに限定されるものではない。

#### 【0011】

本明細書で提示されるプロセス及び表示は、本来的には、どの特定のタイプのコンピュータにも他の装置にも関係付けられない。さまざまな汎用システムが、本明細書の教示に従ったプログラムと共に使用される場合もあるし、所望の方法を実行するより特殊化された装置を構築することが便利であると分かる場合もある。さまざまなこれらのシステムの所望の構造は、以下の説明から現れる。加えて、本発明の実施形態は、どの特定のプログラミング言語についても説明されていない。さまざまなプログラミング言語を使用して、本明細書に説明するような本発明の教示を実施できることが十分理解されよう。

#### 【0012】

以下の説明では、さまざまな図、ダイアグラム、フローチャート、モデル、及び説明が、この出願で提案されている本発明の内容を有効に伝え、且つ、本発明の異なる実施形態を示す異なる手段として提示されている。これらは、単なる例示的なサンプルとして提供され、本発明の限定として構成されるものでないことが当業者には理解されよう。

#### 【0013】

図1は、本発明のいくつかの例示的な実施形態による、DAXメカニズムのサポートを有する動的バイナリ変換を行うことができる装置100のブロックダイアグラム説明図である。装置100は、たとえばコンピューティングプラットフォームとすることができ、メモリ106に動作可能に接続されたプロセッサ102を含むことができる。このコンピューティングプラットフォームは、本明細書では、ターゲットプラットフォームと呼ばれる。

#### 【0014】

本発明の実施形態によれば、メモリ106は、一組のソース命令110及び一組のターゲット命令120を記憶するようになっていてもよい。一組のソースメモリ例110は、本明細書では「ソースコード」とも呼ばれ、一組のターゲット命令120は、本明細書では「ターゲットコード」とも呼ばれる。また、メモリ106は、動的バイナリ変換器108の命令も記憶することができる。プロセッサ102は、動的バイナリ変換器108を実行して、ソース命令セット110をターゲット命令セット120に変換することができる。図2について以下で詳細に説明するように、プロセッサ102は、次に、入力103を印加することによってターゲット命令セット120を実行することができ、ターゲットコ

10

20

30

40

50

ード実行中にターゲットプラットフォーム上でアルゴリズム104を適用して、デノーマル入力制御ビットをマスク解除することができる。動的バイナリ変換器108は、図3～図7について以下で詳細に説明するように、デノーマル例外等の例外をハンドリングする例外ハンドラ109を含むことができる。

【0015】

ソース命令セット110は、デノーマル入力ハンドリングメカニズムをサポートできるたとえばIA-32といったソースアーキテクチャに準拠することができる。デノーマル入力ハンドリングメカニズムは、本明細書ではDAZメカニズムとも呼ばれる。ソース命令セット110は、1つ又は2つ以上の制御命令を含むことができる。これらの制御命令は、ソースプラットフォーム上で実行されている時、そのソースプラットフォームによってサポートされるDAZメカニズムがソースコードによって使用されるか否かを決定することができる。DAZメカニズムがソースコードによって使用される場合、ソースアーキテクチャは、デノーマルオペランドを元のオペランドの符号を有する0に変換することによって、例外命令のSIMD浮動小数点デノーマルオペランド状況に対応し、次に、デノーマルオペランドのこの新しい値を使用して例外命令を実行することができる。

10

【0016】

プロセッサ102は、たとえばIntel Itanium(商標)プロセッサファミリーのアーキテクチャといったターゲットアーキテクチャに準拠することができる。本発明の実施形態によれば、プロセッサ102は、DAZメカニズムをサポートするために、ソースアーキテクチャで必要とされるようなハードウェアを必要としない場合がある。その代わりに、本発明の例示的な実施形態によれば、以下で詳細で説明するように、DAZメカニズムは、ターゲット命令セット120の実行中に、動的バイナリ変換を通じてエミュレーションすることができる。

20

【0017】

装置100の例を非網羅的に列挙したものには、デスクトップパーソナルコンピュータ、ワークステーション、サーバコンピュータ、ラップトップコンピュータ、ノートブックコンピュータ、ハンドヘルドコンピュータ、携帯情報端末(PDA)、携帯電話、ゲームコンソール等が含まれ得る。

【0018】

プロセッサ102の例を非網羅的に列挙したものには、中央処理装置(CPU)、デジタル信号プロセッサ(DSP)、縮小命令セットコンピュータ(RISC)、複合命令セットコンピュータ(CISC)等が含まれ得る。その上、プロセッサ102は、特定用途向け(ASIC)の一部とすることもできるし、特定用途向け標準品(ASSP)の一部とすることもできる。

30

【0019】

メモリ106は、装置100内に固定することもできるし、装置100から着脱可能とすることもできる。メモリ106の例を非網羅的に列挙したものには、次の半導体デバイス、すなわち、シンクロナスダイナミックランダムアクセスメモリ(SDRAM)デバイス、RAMBUSダイナミックランダムアクセスメモリ(RDRAM)デバイス、ダブルデータレート(DDR)メモリデバイス、スタティックランダムアクセスメモリ(SRAM)デバイス、フラッシュメモリ(FM)デバイス、電氣的消去可能プログラマブル読み出し専用メモリ(EEPROM)デバイス、不揮発性ランダムアクセスメモリ(NVRAM)デバイス、ユニバーサルシリアルバス(USB)着脱可能メモリ等；コンパクトディスク読み出し専用メモリ(CD-ROM)等の光デバイス；及び、ハードディスク、フロッピーディスク、磁気テープ等の磁気デバイスの1つ又はいずれかの組み合わせが含まれ得る。

40

【0020】

ソースコードの実行中、ソースコードの命令は、さまざまな例外を起こす場合がある。たとえば、Intel(登録商標)IA-32アーキテクチャに準拠したソース命令は、次の特定の例外をサポートすることができる。すなわち、バイナリ浮動小数点算術につい

50



てANSI/IEEE標準規格754~1985で定義されて必要とされる「無効な演算」、「0による除算」、「オーバーフロー」、「アンダーフロー」、及び「不正確な計算」の浮動小数点例外をサポートすることができる。上記例外は、一般に、SSE例外と呼ばれる場合がある。ターゲットプラットフォームは、SSE例外のハンドリングを制御するためにマスク又はマスク解除することができる1つ又は2つ以上の制御ビットを含むことができる。この制御ビットには、デノーマル入力制御ビットが含まれる。制御ビットは、図2について以下で詳細に説明するように、アルゴリズム104を通じて、プロセッサ102が制御することができる。加えて、ソースコードは、DAXメカニズムがソースコードによって使用されるか否かを判断する、ソースプラットフォームのDAXビットの設定を制御する1つ又は2つ以上の制御命令を含むことができる。

10

**【0021】**

本発明の例示的な実施形態によれば、ソースプラットフォームによってサポートされ且つソースコード110によって使用されるDAXメカニズムは、ターゲットプラットフォーム上でのターゲットコード120の実行を通じて、動的バイナリ変換器108によりエミュレーションすることができる。プロセッサ102は、動的バイナリ変換器108を実行して、たとえばソース命令112及び/又は113を含むソース命令セット110を、たとえばターゲット命令122及び/又は123を含むターゲット命令セット120に変換するようになっていてもよい。ターゲットコードの実行中、たとえばソース命令112といった1つ又は2つ以上の例外ソース命令を、たとえばターゲット命令122といった1つ又は2つ以上の対応する障害のあるターゲット命令の実行を通じて検出できるように、プロセッサ102は、アルゴリズム104を実行して、ターゲットプラットフォーム上でデノーマル入力制御ビットをマスク解除することができる。

20

**【0022】**

本発明の例示的な実施形態によれば、デノーマル例外が検出されると、DAXメカニズムが、動的バイナリ変換器108によりエミュレーション可能である。このエミュレーションは、動的バイナリ変換器108が、所定の値を有する、ターゲット命令のデノーマルオペランドを代入することを通じて行うことができる。このターゲット命令は、デノーマル例外に対応する障害のあるターゲット命令であり、所定の値は、たとえば、元のオペランドの符号を有する0である。プロセッサ102は、次に、デノーマルオペランドについて新しく代入された値を有する障害のあるターゲット命令を実行することができる。

30

**【0023】**

本発明のいくつかの例示的な実施形態によれば、デノーマル例外が、障害のあるターゲット命令122で検出されると、対応する例外ソース命令112を再変換して、新しいターゲット命令122を生成することができる。加えて、プリフィックスコード121を生成して、新しく生成されたターゲット命令122にアタッチすることができる。プリフィックスコード121は、ソース命令112に対するすべてのデノーマルオペランド、及び、ソースコード110の制御命令によって行われる制御を識別して、DAXメカニズムがソースコードによって使用されるか否かを判断することができる。DAXメカニズムがソースコードによって使用される場合、プリフィックスコード121は、上記で詳細に説明したように、DAXメカニズムをエミュレーションすることができる。

40

**【0024】**

本発明の例示的な実施形態によれば、プロセッサ102は、ターゲット命令ブロックにおけるデノーマル例外の個数をカウントする、たとえばキャッシュベースカウンタ105といったカウンタを含むことができる。デノーマル例外の個数が所定のしきい値を超える場合、動的バイナリ変換器108は、図6について以下で詳細に説明するように、DAXメカニズムをエミュレーションすることができる。

**【0025】**

図2は、本発明の例示的な実施形態による、DAXメカニズムのサポートを有する動的バイナリ変換の方法の概略フローチャートである。

**【0026】**

50

本発明の例示的な実施形態によれば、動的バイナリ変換器 108 は、ブロック 202 に示すように、たとえばソースコード 110 といったソースコードを受け取ることができる。次に、動的バイナリ変換器 108 は、ブロック 204 に示すように、ソースコード 110 を変換して、対応するターゲットコード 120 を生成し、ブロック 206 に示すように、ターゲットコード 120 を実行することができる。ソースコード 110 は、1 つ又は 2 つ以上の制御命令を含むことができる。これらの制御命令は、D A Z メカニズムがソースコードにより使用されるか否かを決定する、ソースプラットフォームの D A Z ビットの設定を変更することができる。実行中、ブロック 208 に示すように、ソースコードの制御命令によって行われる制御が、D A Z メカニズムがソースコードで使用されることを示していると判断された場合、D A Z ハンドリングのコード実行の段階中にデノーマル例外を検出できるように、プロセッサ 102 は、ブロック 212 に示すように、ターゲットプラットフォームのデノーマル入力制御ビットをマスク解除することができる。デノーマル入力制御ビットのマスク解除は、たとえば、プロセッサ 102 が、たとえばアルゴリズム 104 (図 1) といったマスク解除アルゴリズムを実行することによって行うことができる。

10

20

30

40

50

#### 【0027】

本発明の例示的な実施形態によれば、ターゲットコード実行中に検出されたデノーマル例外の個別のハンドリングを必要とする 2 つの状況が存在し得る。ブロック 214 に示すように、第 1 の状況は、デノーマル例外を含まないすべての S S E 例外がソースコードにおいてマスクされ、その結果、同じ非デノーマル例外が、対応するターゲットコードにおいてマスクされているときである。この状況では、ブロック 216 に示し、且つ、図 3 について以下で詳細に説明するように、プロセッサ 102 は、ターゲットコードを実行し、デノーマル例外を検出し、そして、D A Z メカニズムを適用することができる。

#### 【0028】

第 2 の状況は、非デノーマル例外の 1 つ又は 2 つ以上がソースコードにおいてマスク解除され、その結果、対応するターゲットコードにおいてマスク解除されているときである。この状況では、ブロック 218 に示し、且つ、図 4 について以下で詳細に説明するように、プロセッサ 102 は、ターゲットコードを実行し、デノーマル例外を検出し、そして、状態回復メカニズムを適用して例外を処理することができる。

#### 【0029】

ブロック 208 に示すように、ソースコードの制御命令によって行われる制御が、D A Z メカニズムがソースコードによって使用されることを示していないと判断された場合、プロセッサ 102 は、ブロック 210 に示すように、ターゲットコードの実行を継続することができる。

#### 【0030】

図 3 は、本発明の例示的な実施形態による、すべての非デノーマル例外がマスクされているときに、プログラムコード実行中にデノーマル例外をハンドリングする方法の概略フローチャートである。ターゲットプラットフォームのデノーマル入力制御ビットは、プロセッサ 102 が、ターゲットコード実行中にデノーマル例外を検出できるようにマスク解除されている。

#### 【0031】

本発明の例示的な実施形態によれば、プロセッサ 102 は、ブロック 302 に示すように、たとえばターゲット命令セット 120 といった一組のターゲット命令の実行を開始することができる。この一組のターゲット命令は、D A Z メカニズムを使用する、たとえばソース命令セット 110 といった一組のソース命令に対応することができる。実行中、プロセッサ 102 は、ブロック 304 に示すように、デノーマル例外を起こす、たとえばターゲット命令 122 といった障害のあるターゲット命令を検出することができる。この例外はデノーマル例外である。その理由は、すべての非デノーマル例外、すなわちデノーマル例外以外の S S E 例外がマスクされており、したがって、検出できないからである。ブロック 306 に示すように、デノーマル例外が検出されると、プロセッサ 102 は、その

デノーマル例外に対するデノーマルオペランドを識別することができ、元のオペランドの符号を有する、たとえば0といった所定の値をデノーマルオペランドに代入することができる。次に、プロセッサ102は、ブロック308に示すように、新しい値のデノーマルオペランドを有するターゲット命令122を実行することができる。ブロック310に示すように、ターゲット命令122の実行から結果を得ることができ、その後、次のターゲット命令の実行で適用されるように、たとえばプロセッサに関連した命令キャッシュに保存することができる。ブロック312において、ターゲットコード実行で使用される命令ポインタを、たとえば障害のあるターゲット命令122に続く命令といった新しいターゲット命令を指し示すように移動又はインクリメントすることができる。ブロック314に示すように、残りのターゲット命令の実行は、命令ポインタによって指し示される新しいターゲット命令において再開することができ、ブロック310に示すように、障害のあるターゲット命令122の実行から得られた結果をその実行において適用することができる。

10

#### 【0032】

図4は、本発明の例示的な実施形態による、1つ又は2つ以上の非デノーマル例外がマスク解除されているときに、ターゲットコード実行中にデノーマル例外をハンドリングする方法の概略説明図である。プロセッサ102がターゲットコード実行中にデノーマル例外を検出できるように、ターゲットプラットフォームのデノーマル入力制御ビットはマスク解除されている。

20

#### 【0033】

本発明の例示的な実施形態によれば、プロセッサ102は、ブロック402に示すように、ターゲット命令を実行することができ、ブロック404に示すように、デノーマル例外を起こす、障害のあるターゲット命令を検出することができる。しかしながら、この場合、この障害のあるターゲット命令は、他の非デノーマル例外を起こす場合もある。その理由は、非デノーマル制御ビットの1つ又は2つ以上がマスクされていないからである。たとえば、「オーバーフロー」例外や「アンダーフロー」例外等の別のタイプのSSE例外が、障害のあるターゲット命令によって起こされる場合がある。障害のあるターゲット命令が起こすことができる他のすべての例外を知ることなく、プロセッサ102は、以下で詳細に説明するように、状態回復メカニズムを使用してデノーマル例外をハンドリングすることが必要な場合がある。

30

#### 【0034】

本発明の例示的な実施形態によれば、バイナリ変換器108の例外ハンドラ109は、ブロック406に示すように、たとえば、ソース命令を回復することができる。このソース命令は、障害のあるターゲット命令の変換元の例外ソース命令である。しかしながら、本発明は、この点に限定されるものではなく、ソース命令は、他の手段又は例外ハンドラによって回復させることもできる。ブロック408に示し、且つ、図5について以下で詳細に説明するように、回復されたソース命令は、再変換され、再実行することができる。回復されたソース命令の実行後、プロセッサ102は、ブロック410に示すように、残りのターゲット命令の実行を再開することができる。

40

#### 【0035】

図5は、本発明の例示的な実施形態による、障害のあるターゲット命令から回復された例外ソース命令を処理する方法の概略説明図である。

#### 【0036】

本発明の例示的な実施形態によれば、回復されたソース命令は、ブロック502に示すように、再生成されるターゲット命令に再変換することができる。この変換は、他の例外による潜在的な影響を回避するために、他の命令の変換とは別個に行うことができる。ブロック504において、たとえばプリフィックスコード121(図1)といったプリフィックスコードを、たとえばターゲット命令122といった再生成されたターゲット命令にアタッチすることができる。プロセッサ102は、その後、ブロック506に示すように、プリフィックスコードを実行して、DAZメカニズムがソースコードによって使用され

50

るか否かを判断することができる。ソースコードの制御命令が、D A Zビットを、D A Zメカニズムがソースコードによって使用されることを示す「1」に設定している場合、プリフィックスコードは、ブロック508に示すように、続いてデノーマル例外に対する1つ又は2つ以上の入力を識別し、デノーマルオペランドであるそれらの入力を0に変更することを始める。次に、プロセッサ102は、ブロック510に示すように、たとえばターゲット命令122といった再生成されたターゲット命令の実行を始めることができる。

【0037】

図6は、本発明のいくつかの例示的な実施形態による、多数のデノーマル例外を有する変換されたコードブロックを構築する方法の概略説明図である。

【0038】

本発明の例示的な実施形態によれば、たとえばターゲット命令セット120といったターゲット命令セットを、複数の命令ブロックに分割することができる。これらの命令ブロックは、たとえば、ブロックの最初の命令のアドレスによって区別することができる。実行中に多数のデノーマル例外を起こすターゲット命令ブロックの実行は、デノーマルオペランド状況が、潜在的に障害のあるターゲット命令の実行の各繰り返し中に存在するか否かを判断することによってさらに改善することができる。

【0039】

本発明の例示的な実施形態によれば、ブロック602に示すように、ターゲット命令ブロックのデノーマル例外をカウントするための、たとえばカウンタ105(図1)等のカウンタを作成することができる。カウンタ105は、キャッシュベースカウンタとすることができ、且つ/又は、他の任意のカウントメカニズムとすることができる。カウンタ105は、ブロック604に示すように、デノーマル例外が起こるごとに、たとえばターゲットコード120のターゲット命令122及び123といったターゲット命令ブロックのデノーマル例外の個数をカウントすることができる。ターゲット命令ブロックによって起こされたデノーマル例外の個数は、ブロック606に示すように、所定のしきい値と比較することができる。この個数がしきい値に達した場合、ブロック608に示すように、ターゲット命令ブロックを書き換えて、たとえば組み込み命令124(図1)といった1つ又は2つ以上の組み込み命令を含めることができる。これらの組み込み命令は、図7について以下で詳細に説明するように、実行されると、ブロックの各SIMD命令のインラインデノーマル状況を識別し、デノーマル例外が検出された時にデノーマル例外のデノーマルオペランドを0に設定することができる。次に、書き換えられたターゲット命令ブロックは、ブロック610に示すように実行することができる。

【0040】

図7は、本発明のいくつかの例示的な実施形態によるターゲット命令ブロックを実行する方法の概略説明図である。

【0041】

本発明の例示的な実施形態によれば、たとえば命令123といったターゲット命令が実行される前に、プロセッサ102は、ブロック702に示すように、たとえば命令124といった組み込み命令を実行して、命令123がデノーマル例外を起こすか否かを判断することができる。ブロック704に示すように、デノーマル例外が検出されず、起こらない場合、ブロック706に示すように、ターゲット命令123を、他のあらゆる命令と同様に実行することができる。一方、デノーマル例外状況が検出された場合、プロセッサ102は、ブロック708に示すように、ターゲット命令123に対するデノーマルオペランドである入力を識別し、そのデノーマルオペランドに0を代入することができる。その後、ブロック710に示すように、デノーマルオペランドについて新しく代入された値を有するターゲット命令123を実行することができる。

【0042】

本明細書では、本発明の一定の機能を図示して説明してきたが、多くの修正、置換、変更、及び均等物が、当業者には思いつくであろう。したがって、添付の特許請求の範囲は、本発明の趣旨内に入るこのようなすべての修正及び変更をカバーすることを目的として

10

20

30

40

50

いることを理解されたい。

【図面の簡単な説明】

【0043】

【図1】本発明のいくつかの例示的な実施形態による、DAZメカニズムのサポートを有する動的バイナリ変換を行うことができる装置のブロックダイアグラム説明図である。

【図2】本発明の例示的な実施形態による、DAZメカニズムのサポートを有する動的バイナリ変換の方法の概略フローチャートである。

【図3】本発明の例示的な実施形態による、すべての非デノーマル例外がマスクされているときに、プログラムコード実行中にデノーマル例外をハンドリングする方法の概略フローチャートである。

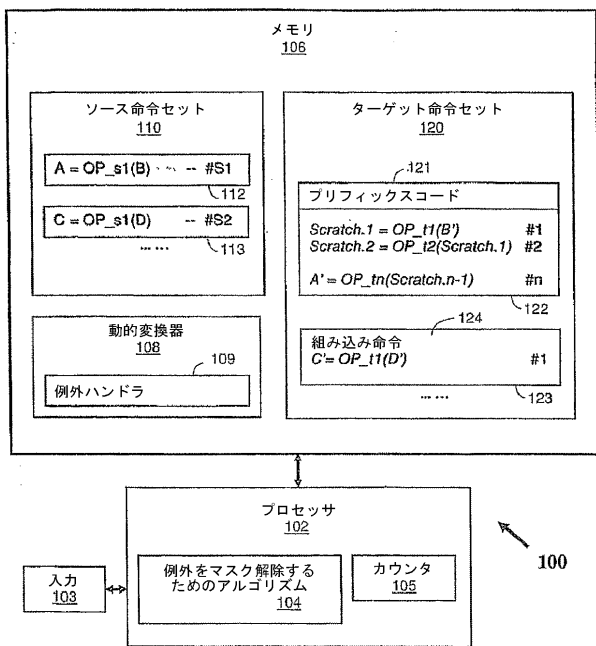
【図4】本発明の例示的な実施形態による、1つ又は2つ以上の非デノーマル例外がマスク解除されているときに、プログラムコード実行中にデノーマル例外をハンドリングする方法の概略説明図である。

【図5】本発明の例示的な実施形態による、対応する障害のあるターゲット命令から回復された例外ソース命令を処理する方法の概略説明図である。

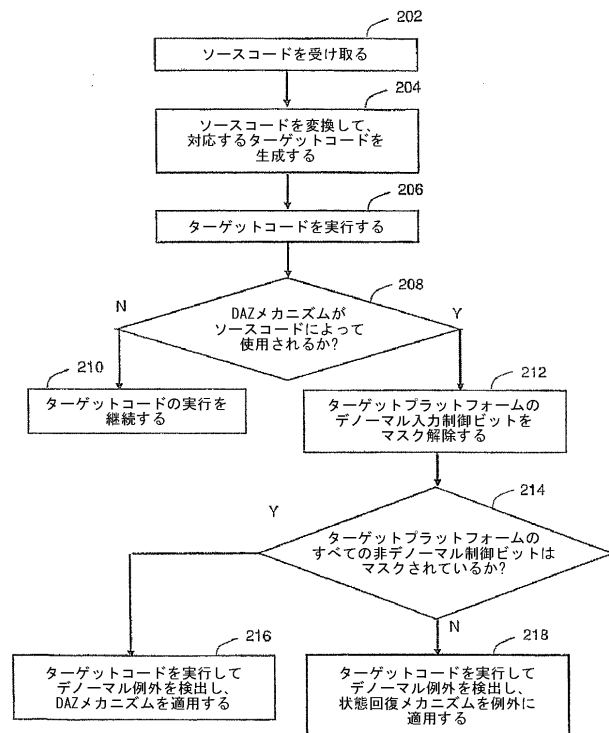
【図6】本発明のいくつかの例示的な実施形態による、多数のデノーマル例外を有する変換されたコードブロックを構築する方法の概略説明図である。

【図7】本発明のいくつかの例示的な実施形態によるターゲット命令ブロックを実行する方法の概略説明図である。

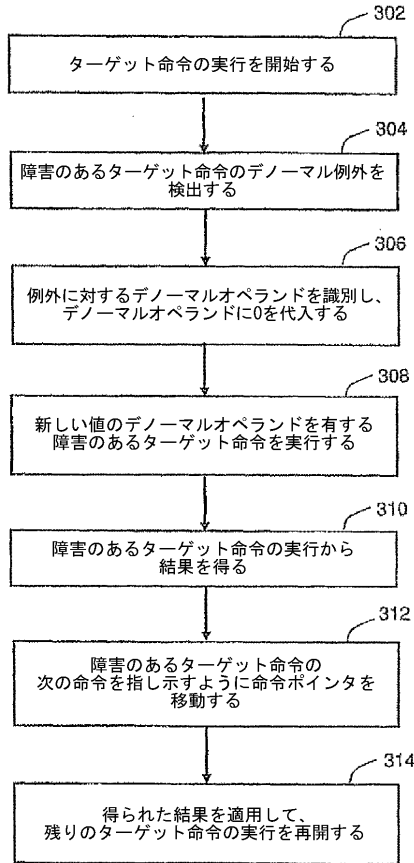
【図1】



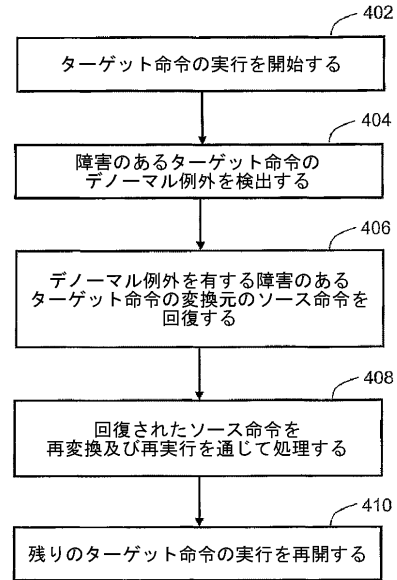
【図2】



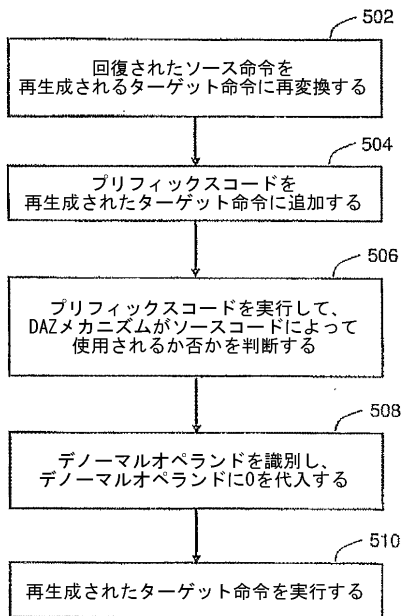
【 図 3 】



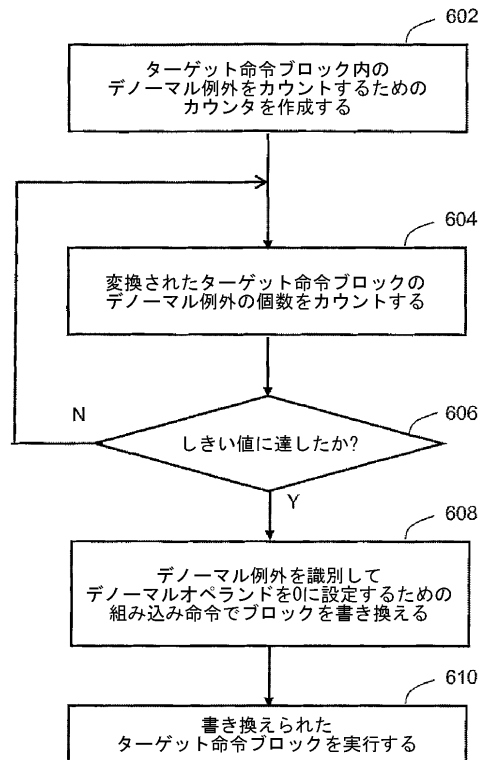
【 図 4 】



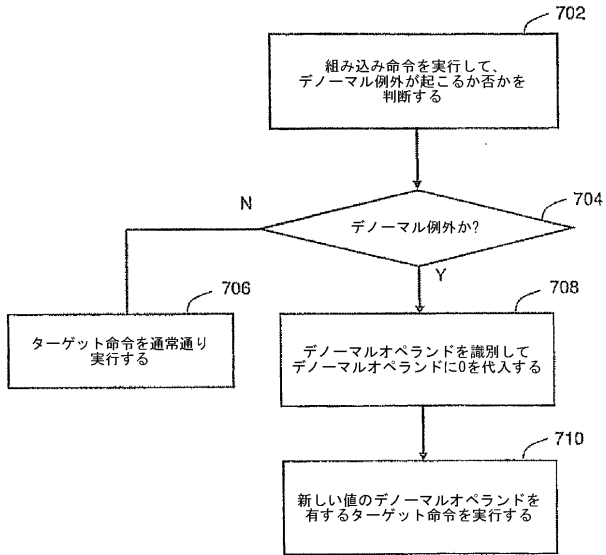
【 図 5 】



【 図 6 】



【 図 7 】



## 【 国際調査報告 】

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2006/025321

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F9/455 G06F11/34		
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		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the International search (name of data base and, where practical, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 681 322 B1 (SAFFORD KEVIN DAVID [US] ET AL) 20 January 2004 (2004-01-20) abstract column 1, line 34 - column 2, line 64	1-29
A	WO 01/27873 A (TRANSMETA CORP [US]) 19 April 2001 (2001-04-19) abstract	9,18,25
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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 document 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		*I* 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 *K* 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. *F* document member of the same patent family
Date of the actual completion of the international search 18 October 2006		Date of mailing of the international search report 25/10/2006
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 051 epo nl, Fax: (+31-70) 340-3016		Authorized officer Kingma, Ype

4



**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No  
PCT/US2006/025321

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6681322	B1	20-01-2004 FR 2801693 A1	01-06-2001
WO 0127873	A	19-04-2001 CA 2380077 A1	19-04-2001
		CN 1378679 A	06-11-2002
		EP 1234277 A1	28-08-2002
		JP 2003511800 T	25-03-2003
		US 6356615 B1	12-03-2002

## フロントページの続き

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

(特許庁注：以下のものは登録商標)

1. フロッピー
2. J A V A

(72)発明者 リ、チアンホイ

中華人民共和国、200127 シャンハイ、ランクン ロード、レーン 471、ナンバー 3  
、ルーム 504

Fターム(参考) 5B022 BA01  
5B033 FA06