

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5404204号
(P5404204)

(45) 発行日 平成26年1月29日 (2014. 1. 29)

(24) 登録日 平成25年11月8日 (2013. 11. 8)

(51) Int. Cl.	F I
G 0 6 F 9/34 (2006. 01)	G O 6 F 9/34 3 5 O Z
G 0 6 F 12/04 (2006. 01)	G O 6 F 12/04 5 4 O C
G 0 6 F 9/455 (2006. 01)	G O 6 F 9/44 3 1 O A

請求項の数 9 外国語出願 (全 16 頁)

(21) 出願番号	特願2009-149905 (P2009-149905)	(73) 特許権者	591003943
(22) 出願日	平成21年6月24日 (2009. 6. 24)		インテル・コーポレーション
(62) 分割の表示	特願2006-541181 (P2006-541181) の分割		アメリカ合衆国 9 5 0 5 4 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・2 2 0 0
原出願日	平成16年10月25日 (2004. 10. 25)	(74) 代理人	110000877
(65) 公開番号	特開2009-282990 (P2009-282990A)		龍華国際特許業務法人
(43) 公開日	平成21年12月3日 (2009. 12. 3)	(72) 発明者	デボアー、ハロルド、シオドール
審査請求日	平成21年7月23日 (2009. 7. 23)		イスラエル、3 0 9 0 0 ヤコブ ジック ロン、ハショフティム ストリート 9 エ ー
(31) 優先権主張番号	10/721, 879	(72) 発明者	エトジョン、オルナ
(32) 優先日	平成15年11月26日 (2003. 11. 26)		イスラエル、3 4 7 6 1 ハイファ、カリ ブ ストリート 5
(33) 優先権主張国	米国 (US)		
前置審査			

最終頁に続く

(54) 【発明の名称】 ミスアラインデータアクセスの検出及び処理のためのデバイス、システム、及び方法

(57) 【特許請求の範囲】

【請求項 1】

ミスアラインデータアクセスを検出して処理するコンピューティングプラットフォームであって、

複数の命令を含むコードブロックを、第1コンピューティングプラットフォームに適合する I A - 3 2 フォーマットから、第2コンピューティングプラットフォームに適合する 6 4 ビットフォーマットに変換する手段と、

変換後の前記コードブロックに対して、データアクセス命令に対するインストゥルメンテーションを実行して、ミスアラインデータアクセスをもたらす命令を含むか否かを示す情報を生成する手段と、

前記情報が前記ミスアラインデータアクセスをもたらす命令を含むことを示す場合、変換後の前記コードブロックについて、命令レベルでのデータアクセスミスアラインメントを検出して、前記データアクセスミスアラインメントが検出されたメモリアドレスの位置をトラッキングする手段と、

変換後の前記コードブロックに含まれ、前記データアクセスミスアラインメントが検出された命令を含むコードシーケンス又は命令のセットを、前記コードブロックの外部の位置に、処理関数として移す手段と、

変換された前記 6 4 ビットフォーマットのコードブロックを、前記第2コンピューティングプラットフォームで実行する手段と、

ミスアラインデータアクセスを処理するために、変換された前記 6 4 ビットフォーマット

トのコードブロックを前記処理関数に分岐させる手段と
を備えるコンピューティングプラットフォーム。

【請求項 2】

ソフトウェアアプリケーションの実行の間に複数回実行される前記コードブロックを決定する手段

をさらに備える請求項 1 に記載のコンピューティングプラットフォーム。

【請求項 3】

ホットコードブロックである前記コードブロックを決定する手段

をさらに備える請求項 1 に記載のコンピューティングプラットフォーム。

【請求項 4】

前記変換する手段は、インタプリタソフトウェアを有する請求項 1 から 3 のいずれか 1 項に記載のコンピューティングプラットフォーム。

【請求項 5】

前記変換する手段は、トランスレータソフトウェアを有する請求項 1 から 3 のいずれか 1 項に記載のコンピューティングプラットフォーム。

【請求項 6】

前記変換する手段は、コンパイラソフトウェアを有する請求項 1 から 3 のいずれか 1 項に記載のコンピューティングプラットフォーム。

【請求項 7】

ミスアラインデータアクセスを検出して処理する方法であって、

複数の命令を含むコードブロックを、第 1 コンピューティングプラットフォームに適合する I A - 3 2 フォーマットから、第 2 コンピューティングプラットフォームに適合する 6 4 ビットフォーマットに変換する段階と、

変換後の前記コードブロックに対して、データアクセス命令に対するインストゥルメンテーションを実行して、ミスアラインデータアクセスをもたらす命令を含むか否かを示す情報を生成する段階と、

前記情報が前記ミスアラインデータアクセスをもたらす命令を含むことを示す場合、変換後の前記コードブロックについて、命令レベルでのデータアクセスミスアラインメントを検出して、前記データアクセスミスアラインメントが検出されたメモリアドレスの位置をトラッキングする段階と、

変換後の前記コードブロックに含まれ、前記データアクセスミスアラインメントが検出された命令を含むコードシーケンス又は命令のセットを、前記コードブロックの外部の位置に、処理関数として移す段階と、

変換された前記 6 4 ビットフォーマットのコードブロックを、前記第 2 コンピューティングプラットフォームで実行する段階と、

ミスアラインデータアクセスを処理するために、変換された前記 6 4 ビットフォーマットのコードブロックを前記処理関数に分岐させる段階と
を備える方法。

【請求項 8】

前記処理関数に分岐させる段階の後に、対応するミスアラインデータアクセス命令シーケンスを実行し、前記ミスアラインデータアクセスを処理する段階
をさらに備える請求項 7 に記載の方法。

【請求項 9】

前記処理関数から分岐して、変換された前記 6 4 ビットフォーマットのコードブロックの実行を再開する段階

をさらに備える請求項 7 または 8 に記載の方法。

【発明の詳細な説明】

【背景技術】

【0001】

コンピューティングプラットフォームの分野において、ソフトウェアアプリケーション

10

20

30

40

50

は元々、第1のコンピューティングプラットフォーム、例えば32ビットベースのコンピューティングプラットフォーム、例えばIntel（登録商標）アーキテクチャ32（IA-32）によって実行されるべく記述される。いくつかのケースにおいて、ソフトウェアアプリケーションを変換して実行する適切なハードウェア及び／又はソフトウェアを用いて、第2のコンピューティングプラットフォーム、例えば64ビットベースのコンピューティングプラットフォーム、例えばIntel（登録商標）Itanium（登録商標）プロセッサ上で、そのソフトウェアアプリケーションを実行することが可能である。

【0002】

第2のコンピューティングプラットフォーム上でのソフトウェアアプリケーションの変換又は実行の間、ミスアラインデータアクセス及びデータミスアラインメントに関連する問題が生じる場合がある。データミスアラインメントは、例えば、プロセッサによって効率的にアクセスされ得ない1つのメモリアドレスに存在するデータアイテムを含む。第2コンピューティングプラットフォームがミスアラインデータアイテムにアクセスしようとした場合に、望ましくないオーバーヘッド、例えば追加のプロセッシングサイクル又はプロセッシング時間が必要となり得る。いくつかのケースにおいて、第2コンピューティングプラットフォーム上でのソフトウェアアプリケーションの実行は、例えば、ソフトウェアアプリケーションが第1（すなわち、元の）コンピューティングプラットフォームによって実行された場合にも生じ得るデータミスアラインメントにより、既存のプラットフォーム問題を悪化及び増大させる。これはパフォーマンス速度を著しく低下させ、プロセッシング時間及び／又はプロセッシングサイクルの数を著しく増加させ得る。その上、いくつかのケースにおいて、ミスアラインデータアクセスイベントは、例えば第2コンピューティングプラットフォーム上で動作するアプリケーションによって又はオペレーティングシステムによってエラーとして扱われ、その結果、アプリケーションの早期の終了又は他の望ましくない結果をもたらす。

【0003】

データミスアラインメントアクセス問題は、ミスアラインデータアクセスイベントをもたらすそれぞれの命令を置き換える比較的長いコードシーケンスを用いて、部分的に緩和され得る。しかしながら、全てのデータミスアラインメントアクセス問題を防ぐためのそのような長いコードシーケンスのアプリケーション体は非効率的であり、著しいオーバーヘッド、例えば追加のプロセッシングサイクル及び／又はプロセッシング時間を招く。

【図面の簡単な説明】

【0004】

本発明に関する主題は、明細書の結びの部分に特に示され、明確に請求される。本発明は、一方で、添付の図面とともに読まれた場合に、以下の詳細な説明を参照することにより、それらの特徴及び利点とともに構成及びオペレーション方法の両方について最も良く理解されるだろう。

【0005】

【図1】本発明のいくつかの実施形態に係る、データミスアラインメントを検出及び処理することができるコンピューティングプラットフォームの概略図である。

【0006】

【図2】本発明のいくつかの実施形態に係る、データミスアラインメントを検出及び処理する方法のフローチャートの概略である。

【0007】

説明の簡単さ及び明確さを目的として、図面に示される構成要素は必ずしも縮尺どおりに描かれる必要がないことが理解されるだろう。例えば、明確さを目的として、いくつかの構成要素の大きさが、他の構成要素に比較して誇張される。さらに、適切である場合には、対応する又は同様の構成要素を示すべく、参照番号が複数の図面にわたって繰り返される。

【発明を実施するための最良の形態】

【0008】

以下の詳細な説明において、本発明の包括的な理解を提供することを目的として、多くの具体的な細部が説明される。しかしながら、本発明は、これらの具体的な細部なしで実施され得ることが当業者に理解されるだろう。他のケースでは、本発明を不明瞭にしないよう、よく知られた方法、手順、構成要素、ユニット、及び／又は回路は、詳細には説明されない。

【0009】

図1は、本発明の典型的な実施形態に係る、データミスアラインメントを検出及び処理することができるコンピューティングプラットフォーム110を概略的に示す。コンピューティングプラットフォーム110は、例えば、データプロセッシング又は種々のソフトウェアアプリケーションの実行のために使用され、本発明のいくつかの実施形態に係るミスアラインデータアクセスの検出及び／又は処理を実装する。コンピューティングプラットフォーム110は、1つのコンピューティングデバイスを備える。例えば、コンピューティングプラットフォーム110は、パーソナルコンピュータ、デスクトップコンピュータ、モバイルコンピュータ、ラップトップコンピュータ、ノートブックコンピュータ、端末、ワークステーション、サーバコンピュータ、携帯情報端末(PDA)、タブレットコンピュータ、特化された又は専用のコンピューティングデバイス、ネットワークデバイス、又は類似のものの一部を少なくとも含んでよい。コンピューティングプラットフォーム110は、ハードウェアコンポーネント及び／又はソフトウェアコンポーネントの任意の適切な組合せを用いて実装される。

【0010】

図1に示された例において、コンピューティングプラットフォーム110は、例えば、1以上のプロセッサ141、1以上のメモリユニット142、1以上のストレージユニット143、1つの出力ユニット144、及び1つの入力ユニット145を備える。コンピューティングプラットフォーム110は、コンピューティングプラットフォーム110の1以上のコンポーネントに動作可能に連携する、技術的に知られた他の適切なコンポーネント又はコンポーネントセットを備えてよい。

【0011】

プロセッサ141は、例えば、1つの中央処理装置(CPU)、1つのデジタルシグナルプロセッサ(DSP)、1以上のコントローラ、或いは任意の適切な特定用途及び／又は汎用及び／又は多目的のプロセッサ又はマイクロプロセッサ或いはコントローラを含む。メモリユニット142は、例えば、ランダムアクセスメモリ(RAM)、読み出し専用メモリ、ダイナミックRAM(DRAM)、シンクロナスDRAM(SDRAM)、又は他の適切なメモリユニットを含む。ストレージユニット143は、例えば、ハードディスクドライブ、フレキシブルディスクドライブ、コンパクトディスク(CD)ドライブ、CD-ROMドライブ、或いは他の適切なリムーバブル又は非リムーバブルなストレージユニット又はメモリモジュールを含む。出力ユニット144は、例えば、1以上のカード、アダプタ、コネクタ、及び／又はモニタに接続可能又はモニタと通信可能なコンポーネントを含む。入力ユニット145は、例えば、1以上のカード、アダプタ、コネクタ、及び／又はキーボード、マウス、又はタッチパッドに接続可能又はキーボード、マウス、又はタッチパッドと通信可能なコンポーネントを含む。メモリユニット142、ストレージユニット143、出力ユニット144、及び／又は入力ユニット145は、プロセッサ141と動作的に関連する。技術的に知られたように、プロセッサ141、メモリユニット142、ストレージユニット143、出力ユニット144、及び／又は入力ユニット145は、他の適切なコンポーネント及び／又は実装を有してよい。

【0012】

いくつかの実施形態において、ソフトウェアは、例えばストレージユニット143又はメモリユニット142内に記憶されてよく、プロセッサ141を用いて実行されてよい。そのようなソフトウェアは、例えば、1以上のオペレーティングシステム、例えば、Microsoft Windows(登録商標)、Linux、Unix(登録商標)、Apple OS、Solaris、Sun-OS、HP-UX、又は他の適切なオペレー

ティングシステムを含んでよい。ソフトウェアはさらに、1以上のソフトウェアアプリケーション、1以上のドライバ、コンパイラ、インタープリタ、エミュレータ、実行レイヤ、ソフトウェア環境、管理されたソフトウェア環境、トレーションレイヤ、及び／又は種々の他の適切なソフトウェアコンポーネントを含んでよい。ソフトウェアは、例えば、ソフトウェアコンポーネント、ソフトウェアアプリケーション、及び／又は本発明の実施形態に従う1以上の方法を実装または使用するソフトウェアレイヤ、及び／又は他の適切なソフトウェアコンポーネントを含んでよい。いくつかの実施形態において、ソフトウェア及び／又はメモリユニット142は、1以上のミスアラインデータアイテム、例えば、メモリユニット142内の、プロセッサ141によって効率的にアクセスされ得ないメモリアドレスにあるデータアイテムを含んでよい。

10

【0013】

いくつかの実施形態において、コンピューティングプラットフォーム110は、場合によっては、ソフトウェアコンポーネント及び／又はハードウェアコンポーネントの任意の適切な組合せを用いて実装されるトランスレータ150を含んでよい。例えば、一実施形態において、トランスレータ150は、メモリユニット142及び／又はストレージユニット143内に記憶された、プロセッサ141を用いて起動又は実行されるソフトウェア及び／又は命令を含んでよい。いくつかの実施形態において、トランスレータ150は、ソフトウェアアプリケーション又は1以上の命令を、第1コンピューティングプラットフォームに適合する第1フォーマットから第2コンピューティングプラットフォームに適合する、例えばコンピューティングプラットフォーム110に適合する第2フォーマットに変換及び／又は変換してよい。一実施形態において、トランスレータ150は、例えば、専用ソフトウェアレイヤ、例えば1つの実行レイヤを用いて実装され得る。いくつかの実施形態において、トランスレータ150は、本発明の実施形態に従う、ミスアラインデータアクセスを検出及び処理する1以上のオペレーションを実行する1以上のモジュール又はコンポーネント（図示せず）、例えば、1つの検出モジュール、1つのインストゥルメンテーションモジュール、1つのライトインストゥルメンテーションモジュール、ヘビーインストゥルメンテーションモジュール、1つの変換モジュール、1つの再変換モジュール、1つの最適化モジュール、1つのコード修正モジュール、1つのコード分類モジュール、又は類似のものを含んでよい。

20

【0014】

いくつかの実施形態において、トランスレータ150は、以下に説明するように、データミスアラインメントを検出及び／又は処理してよい。例えば、トランスレータ150によって又はトランスレータ150に従って実行される命令又はオペレーションは、本発明のいくつかの実施形態に従う、以下に説明されるようなデータミスアラインメントの検出及び処理の典型的な方法からもたらされる修正を含む修正された変換プロセスを反映する。例えば、本発明のいくつかの実施形態において、トランスレータ150は、以下に説明するように、データミスアラインメントに関連するパラメータを考慮する変換方法又は手順を実装してよい。

30

【0015】

図2は、本発明のいくつかの実施形態に係る、データミスアラインメントを検出及び処理する一フローチャートを概略的に示す。図2の方法は、本発明の他の実施形態に係る方法とともに、例えば図1のコンピューティングプラットフォーム110とともに、トランスレータ150、プロセッサ141、コンピューティングプラットフォーム110によって実行されるソフトウェア又は命令によって、及び／又は他の適切なコンピューティングプラットフォーム、デバイス、装置、及び／又はシステムとともに使用される。いくつかの実施形態において、図2の方法は、適切なソフトウェアコンポーネント及び／又はハードウェアコンポーネントを用いて実装される、1つのコンパイラ、1つのインタープリタ、及び／又は1つのエミュレータによって使用される。

40

【0016】

本明細書において使用される"コードのブロック"及び／又は"コードブロック"という語

50

句は、例えば、1以上の命令又は命令セットを含んでよいことが注意される。例えば、1つのコードブロックは、5個の命令、又は20個の命令等を含んでよい。

【0017】

本発明のいくつかの実施形態に従って、"ホットブロック"は、例えば、最適化のため又はさらなる最適化オペレーションのための1つの候補として特定及び／又は分類された1つのコードブロックを含んでよい。例えば、ソフトウェアアプリケーションの実行の間に複数回、例えば多数回実行されるコードブロックが、1つのホットブロックとして特定及び／又は分類され得る。他の適切な規則又は条件が、1つのホットブロックを特定、分類、及び／又は定義すべく使用されてよい。

【0018】

本発明のいくつかの実施形態に従って、"コールドブロック"は、例えば、ホットブロックとして特定及び／又は分類されない1つのコードブロックを含んでよい。例えば、コールドブロックは、ソフトウェアアプリケーションの実行の間に1回だけ又は少数回実行されるコードブロックを含んでよい。いくつかの実施形態において、例えば、コードブロックは、初期の変換オペレーションの間にコールドブロックとして特定及び／又は分類され、さらなる変換及び／又は最適化オペレーションの間にホットブロックとして再び分類されてよいことが注意される。

【0019】

ブロック210で示されるように、この方法は、1つのアプリケーションソフトウェアの1つのコードブロックを、第1コンピューティングプラットフォームに適合した第1フォーマットから、第2コンピューティングプラットフォームに変換又はコンバートしてよい。第1フォーマットにおけるコードブロックは、"元のコードブロック"と呼ばれ、第2フォーマットにおけるコードブロックは"変換されたコードブロック"として呼ばれる。いくつかの実施形態において、例えば、1つのコードブロックは、Intel（登録商標）アーキテクチャ32（IA-32）からIntel（登録商標）アーキテクチャ64（IA-64）に変換され得る。変換は、例えば、トランスレータ150を用いて行われてよい。一実施形態において、当該変換は、1つのコンパイラソフトウェア又は1つのインタープリタソフトウェアを用いて行われてよい。変換は、例えば、動的及び／又はリアルタイムに、例えば実質的に実行時に又は実行時の近くで行われてよい。いくつかの実施形態において、変換は、予め、例えば実行時に先立って、行われてよい。

【0020】

本明細書で説明されるように、本発明の方法の実施形態は、複数のインストゥルメンテーションオペレーションを含んでよい。本発明のいくつかの実施形態に従って、インストゥルメンテーションは、例えば、コードブロック又は命令の振る舞い又はオペレーションをトラッキング、診断、デバッグ、及び／又は解析することを目的として、1以上の命令のあるコードブロック又はある命令に追加することを含む。

【0021】

本明細書で使用される"ライトインストゥルメンテーション"という語句は、例えば、一コードブロックのレベルで行われるインストゥルメンテーションを含む。一実施形態において、例えば、"ライトインストゥルメンテーション"は、1つのコードブロックの実行がミスラインデータアクセスをひき起こすか否か、又は1つのコードブロックの実行が、ミスラインデータアクセスイベントをもたらすか否かを検出するインストゥルメンテーションを含んでよい。いくつかの実施形態において、"ライトインストゥルメンテーション"を実行することは、例えば、与えられたコードブロックの実行がミスラインデータアクセスイベントを含むか否かを決定することをもたらす。

【0022】

本明細書で使用される"ヘビーインストゥルメンテーション"という語句は、例えば、一命令のレベルで行われるインストゥルメンテーションを含む。一実施形態において、例えば、"ヘビーインストゥルメンテーション"は、1つの命令の実行がミスラインデータアクセスをひき起こすか否か、又は1つの命令の実行がミスラインデータアクセスイベン

10

20

30

40

50

トをもたらすか否かを検出するインストールメンテーションを含んでよい。他の実施形態において、"ヘビーインストールメンテーション"は、例えば、実行することがミスラインデータアクセスをもたらす1以上の命令を特定するインストールメンテーションを含む。いくつかの実施形態において、"ヘビーインストールメンテーション"を実行することは、例えば、実行することがミスラインデータアクセスイベントをもたらす命令の位置の決定をもたらす。

【0023】

いくつかの実施形態において、本実施形態で使用される"ミスラインデータアクセスを検出すること"という語句は、例えば、変換されたコードブロックの実行または続く実行がミスラインデータアクセス又はミスラインデータアイテムへのアクセスを含むことを検出することを含む。いくつかの実施形態において、"ミスラインデータアクセスを検出すること"という語句は、例えば、実行された場合に、ミスラインデータアクセス又はミスラインデータアイテムへのアクセスをもたらす命令の位置を検出することを、付加的に又は代替的に含む。

【0024】

いくつかの実施形態において、本明細書で使用される"ミスラインデータアクセスを処理"、"ミスラインデータアクセスを回避"、及び／又は"ミスラインデータアクセスを防止"という語句は、例えば、修正された変換されたコードブロックを生成すべく、変換されたコードブロックを修正すること、又は修正された変換されたコードブロックを生成すべく元のコードブロックを再び変換することを含む。いくつかの実施形態において、修正された変換されたコードブロックは、実行された場合に、ミスラインデータアイテムに、修正されない変換されたコードブロックの実行の間における当該ミスラインデータへの対応するアクセスに比べて、より効率的に、より高速に、より短い時間で、及び／又はより少量のプロセッシングサイクルを用いて、アクセスし得る。いくつかの実施形態において、"ミスラインデータアクセスを処理"、"ミスラインデータアクセスを回避"、及び／又は"ミスラインデータアクセスを防止"は、例えば、ミスラインデータアイテムに、複数に分けて、例えば、2、4、8、16に分けて、又は他の適切な数に分けて、複数のアクセスステージで、複数のアクセス命令を用いて、複数のロード命令を用いて、1つの専用のコードシーケンス、例えば本明細書で詳説されるようなCode1を用いて、又は本発明の実施形態に従う他の適切な方法を用いて、アクセスすることを含む。

【0025】

いくつかの実施形態において、"ミスラインデータアクセスを処理"、"ミスラインデータアクセスを回避"、及び／又は"ミスラインデータアクセスを防止"は、例えば、ミスラインデータアクセスイベントを処理、回避、防止、及び／又は矯正する他の適切な方法を含む。例えば、"ミスラインデータアクセスを処理"、"ミスラインデータアクセスを回避"、及び／又は"ミスラインデータアクセスを防止"は、例えば、データアイテムをアラインすること、データアイテムを第1メモリ位置から第2メモリ位置に移動すること、データアイテムを第1メモリ位置から第2メモリ位置にコピーすること、又は他の適切な方法によって、データアイテムのミスアラインメントを矯正することを含む。

【0026】

ブロック220に示されるように、変換されたコードブロックに、ライトインストールメンテーション又は解析を行う。いくつかの実施形態において、実行がミスラインデータアクセスを必要とする、実質的に全ての命令又は少なくともいくつかの命令が、ライトインストールメントされる。ライトインストールメンテーションは、例えば、修正された命令が、予め定められた条件の発生、又は予め定められた規則が合致した場合を示すよう、1以上の命令の修正を含む。例えば、いくつかの実施形態において、変換されたコードブロックにおいてミスラインデータアクセスイベントが検出された場合に、1つのシグナルが生成される。一実施形態において、例えば、当該シグナル又はインジケーションは、トランスレータ150内の1つの検出モジュールによって、トランスレータ150内の1つの変換モジュールに提供される。

【0027】

一実施形態において、ブロック220のライトインストゥルメンテーションの目的又は結果は、例えば、もしミスアラインデータアクセスイベントを処理、矯正、回避、又は防止することが必要とされる場合に、さらなる解析及び最適化オペレーションが適用できるように、変換されたコードブロックの実行がミスアラインデータアクセスイベントをもたらすか否かを検出することを含む。ブロック230で示されるように、方法は、ミスアラインデータアクセスイベントが、変換されたコードブロック内に検出されたか否かをチェックしてよい。そのチェックが否である場合、ブロック270に至る矢印235によって示されるように、変換されたコードブロックは維持及び／又は使用される。そのチェックが否である場合、ブロック240及び先に示されるように、変換されたコードブロックはさらに調査、解析、及び／又は最適化される。

10

【0028】

さらなる調査、解析、及び最適化は、ブロック240に示されるように、変換されたコードブロックのヘビーインストゥルメンテーション又は解析を含む。いくつかの実施形態において、その実行がミスアラインデータアクセスを含む実質的に全ての命令が、ヘビーインストゥルメントされる。ヘビーインストゥルメンテーションは、例えば、ミスアラインデータアクセスイベントにおいて、ミスアラインデータアクセスについて情報が提供及び／又は登録されるよう1以上の命令の修正を含んでよい。いくつかの実施形態において、その情報は、例えば、ミスアラインデータアクセスをもたらす変換されたコードブロック内の命令又は複数の命令のインジケーションを含む。その情報は、例えば、ミスアラインデータアクセスのタイプまたはプロパティ、例えばミスアラインメントの粒度のインジケーションを含む。例えば一実施形態において、ミスアラインデータアクセスイベントをもたらす8ビットデータアクセスがヘビーインストゥルメントされて、ミスアラインデータアクセスが1バイト又は4バイトの粒度であることを示す。ミスアラインデータアクセスの種々の他のプロパティ、特性、属性、及び／又は特性が特定、検出、登録、及び／又は解析されてよい。

20

【0029】

250に示されるように、ブロック240のヘビーインストゥルメンテーションは、命令レベルでのデータアクセスミスアラインメントの検出及び処理のために使用される。いくつかの実施形態において、ミスアラインデータアクセスをもたらすと検出された命令のいくつか或いは全て又は実質的に全ては、再生成、再変換、修正、最適化、及び／又は置換されて、データアクセスミスアラインメントを処理、矯正、回避、及び／又は防止し、ミスアラインデータアイテムに比較的より効率的にアクセスすることを可能にし、ミスアラインデータアイテムの比較的高速なアクセスを可能にし、又は、ミスアラインデータアイテムにアクセスする代替の又は適格な方法を用いることを可能にする。

30

【0030】

一実施形態において、ブロック250で示された検出及び処理オペレーションは、例えば、ホットブロックの変換の間に行われてよい。例えば、ホットブロックが特定され、ブロック250で示される検出及び処理オペレーションは、ホットブロック内に含まれる1以上のコールドブロックに行われてよい。

40

【0031】

いくつかの実施形態において、ブロック250で示される検出及び処理オペレーションは、例えば、以下の仮想コードを用いて行われてよい。

```
// test bit0 to see if address is 2-byte aligned.
// Predicates p.mis and p.al set appropriately.
// Will use p.mis and p.al to predicate the following instructions
tbit p.mis,p.al      = r.addr, 0
// 2-byte load if aligned
(p.al) ld2 r.val     = [r.addr]
// if misaligned, load each byte separately
```

50

```
(p.mis) ldl r.val    = [r.addr]
(p.mis) add r.addrH = 1, r.addr
(p.mis) ldl r.valH   = [r.addrH]
// combine the separately loaded bytes
(p.mis) dep r.val    = r.valH, r.val, 8, 8
Code 1
```

【0032】

Code 1 は具体的な目的のみのために提示され、本発明の実施形態はこの件に限定されず、Code 1 に加えて又はCode 1 の代わりに、他の適切な命令、命令セット、オペレーション、仮想コード、又はアルゴリズムが本発明の実施形態に従って使用され得ることが指摘される。

10

【0033】

本発明のいくつかの実施形態において、例えば、ブロック250で示される検出及び処理オペレーションは、Code 1 に対する多様な変更を含み得る代替のコードを使用してよい。いくつかの実施形態において、例えば、ミスアラインデータアクセスイベントの初期の検出の1以上の結果は、ミスアラインデータアクセスイベントの、結果として生ずる1又はいくつかの検出を強化及び／又は予防すべく、登録及び／又はトラッキングされてよい。例えば、一実施形態において、例えばホットブロックの解析を通じてミスアラインデータアクセスが検出されたメモリアドレスの位置が、登録及び／又はトラッキングされてよい。登録又はトラッキングは、例えば、適切なリスト、トラッキングリスト、スタック、ルックアップテーブル、配列、データベース、変数、レジスタ、共用体、又は他の適切な任意の方法を用いて行われてよい。以前の検出の登録、トラッキング、及び／又は記憶は、例えば、プロセッサ141、メモリユニット142、ストレージユニット143、トランスレータ150、プロセッサ141を用いて実行される適切なソフトウェア、及び／又はコンピューティングプラットフォーム110の他の適切なコンポーネントを用いて、行われてよい。

20

【0034】

ブロック250で示される検出及び処理オペレーションにおいて、いくつかの実施形態は、例えば、第1の命令に対するインストゥルメンテーションを解析及び／又は実行してよい。結果として、例えば、第1命令によって第1メモリアドレスをアクセスすることがミスアラインデータアクセスをもたらすことが検出される。そのような場合、第1メモリアドレスの位置は、例えばメモリユニット142及び／又はストレージユニット143の中に記憶されるレジストリ又はトラッキングリストを例えば用いて、登録及び／又はトラッキングされてよい。第2命令が第2メモリアドレスをアクセスする必要がある場合、データミスアラインメント検出を目的として、本発明の方法は、第2メモリアドレスの位置が第1メモリアドレスの位置と同じであるか否かを、例えばメモリユニット142及び／又はストレージユニット143の中に記憶されるトラッキングリスト又はレジストリを用いてチェックしてよい。例えば、第1命令の解析が第1メモリアドレスをミスアラインデータアクセスの原因として既に特定している場合、肯定のチェック結果は、第2命令を解析及び／又はインストゥルメントする必要を取り除き得る。いくつかの実施形態において、データミスアラインメント検出について、異なるメモリアドレスも全く同様であることが指摘される。例えば第1及び第2メモリアドレスの間の差又は間隔がNバイトである場合、データミスアラインメント検出を目的としては、第1及び第2メモリアドレスは、Nバイトのミスアラインメント及び／又はNの因数であるKバイトのミスアラインメントについて全く同様であり得る。例えば、一実施形態において、メモリアドレスL及びメモリアドレスL+8は、8バイトのデータミスアラインメントを検出する目的として、或いは4バイト又は2バイトのデータミスアラインメントを検出する目的として、全く同様である。

30

40

【0035】

同様に、いくつかの実施形態において、ブロック250で示される検出及び処理オペレ

50

ーションを通じて、与えられたミスアラインデータアクセスの検出に使用されるパラメータ、属性、プロパティ、変数、特性、結果、チェック結果、及び／又は計算結果のいくつか又は全ては、後続の検出において使用又は再使用されてよい。例えば、Code 1の2バイトロードの例は、ミスアラインデータアクセスの1以上の以前の検出又は解析においてセット、決定、及び／又は計算された1以上の属性を使用し得る。

【0036】

いくつかの実施形態において、ブロック250で示される検出及び処理オペレーションを行うことは、例えばミスアラインデータアクセスが検出された場合、例えばデータアクセスを実行するための1つのコードシーケンス又は命令のセットを使用することを含む。当該コードシーケンス又は命令のセットは比較的長い比較的多数の命令又はオペレーションを含んでよい。いくつかの実施形態において、そのようなコードシーケンス又は命令のセットは、ソフトウェアアプリケーションの中の適切な位置に移されてよい。例えば、一実施形態において、スケジューラプロセスは、これらの命令のいくつか又は全てを、変換されたコードブロックの外部又はリモートの位置に、例えばサブルーチン、プロシージャ、関数、又は同種のものとして、移してよい。実行中、変換されたコードブロックの中の命令は、そのポイントで実行が外部又はリモートの命令に進み得る"分岐"命令に到達するまで実行されてよい。外部又はリモートの命令は、その実行が、変換されたコードブロックの実行が外部又はリモートの命令に"分岐された"ポイントから再開される実行をひき起こす最後の命令を含む。

【0037】

ブロック260で示されるように、必要に応じて、いくつかの実施形態は、例えば、上記のオペレーションを終了した後又はホットブロックが変換された後に発生し得るミスアラインデータアクセスを検出及び処理すべく、変換されたコードブロックをさらに最適化又は解析してよい。例えば、一実施形態において、ミスアラインデータアクセスは、相対的に最適化された変換されたコードが生成された後に表れる場合がある。いくつかの実施形態において、その結果は、例えばブロック250で示される検出及び処理オペレーションとともに又は他の適切な時に使用されるさらなる最適化オペレーションを使用して、排除、処理、回避、又は緩和される。例えば、ブロック250で示されるオペレーションの間、1つの命令はミスアラインデータアクセスイベントを生成又はひき起こさないが、当該命令が、実行中にミスアラインデータアクセスをもたらす有力候補であることが検出される場合がある。そのような命令は、"候補命令"と呼ばれる。候補命令の特定、検出、分類、登録、及び／又は評価が、例えば1以上の適切な規則又は条件、例えば命令のタイプ、ミスアラインデータアクセスを命令がもたらす見込みの可能性、又はミスアラインデータアクセスを命令がもたらす著しい可能性の検出に関して実行されてよい。

【0038】

本発明のいくつかの実施形態に従って、ブロック260で示されるオペレーションは、例えば候補命令を検出すると、候補命令に対するインストゥルメンテーションを実行することを、必要に応じて含んでよい。ミスアラインデータアクセスが、使用されたインストゥルメンテーションを用いて検出された場合、変換されたブロックは破棄されてよい。さらに、方法は、元のコードブロックが変換された場合に、元のコードブロックの中の実質的に全ての候補命令は、例えばCode 1又は任意の他の適切なコードシーケンスを用いて、ミスアラインデータアクセスを検出して処理するように変換されるインジケーションを登録してよい。登録されたインジケーションは、例えば、トランスレータ150、メモリユニット142、及び／又はストレージユニット143の中に記憶されてよい。いくつかの実施形態において、インジケーションは、候補命令だけでなく、候補命令のタイプの実質的に全ての命令にも関連してよい。元のコードブロックの変換において、登録されたインジケーションが検出又はトラッキングされ、そしてそれに応じて、元のコードブロックの中の実質的に全ての候補命令、又は元のコードブロックの中の候補命令のタイプの全ての命令が、例えばCode 1又は任意の他の適切なコードシーケンスを用いて、ミスアラインデータアクセスを検出して処理するように変換されてよい。

【0039】

いくつかの実施形態において、上記の1以上のオペレーション、例えばブロック240のデータミスアラインメント検出及び処理オペレーション、ブロック240に関連して説明されたコード修正、又はブロック260の最適化オペレーションは、1以上の適切なソフトウェア及び／又はハードウェアコンポーネントによって実行されてよい。例えば、いくつかの実施形態において、コード修正は、トランスレータ150、或いはトランスレータ150の外部のモジュール又はコンポーネントによって実行されてよい。一実施形態において、コードの修正は、例えば、その修正がコード修正をひき起こし得るトランスレータ150の修正、或いは他の適切なコンポーネント又はモジュールの修正によって行われてよい。同様に、データミスアラインメントの検出及び／又は処理は、トランスレータ150によって、或いはトランスレータ150の内部又は外部にある1以上のモジュールによって実行されてよい。

10

【0040】

本発明のいくつかの実施形態において、上記の1以上のオペレーション、例えば、ブロック210の変換オペレーション、ブロック220のライトインストゥルメンテーションオペレーション、ブロック230のデータミスアラインメントチェック、ブロック240のヘビーインストゥルメンテーションオペレーション、ブロック250のデータミスアラインメント検出及び処理オペレーション、ブロック260の最適化オペレーション、又は他の適切なオペレーションが繰り返される、或いは1度以上、又は多様なオペレーションの順番で実行されてよい。さらに、いくつかの実施形態において、1以上のこれらのオペレーションの結果として、変換されたコードブロックは破棄、削除、置換、調査、解析、修正、最適化、再生成、再生産、又は再変換されてよい。ブロック270で示されるように、上記の1以上又は全てのオペレーション、或いは上記の1以上のオペレーションの何度かのイタレーション又は繰り返しの後、結果として得られる変換されたコードブロックは保持及び使用されてよく、或いは保持されて使用され得る変換されたコードブロックとして登録されてよい。

20

【0041】

本発明の実施形態は、多様な利益を与える。例えば、いくつかの実施形態において、データミスアラインメント検出及び処理は、プロセッシング時間及び／又はプロセッシング時間を著しく減少させ、パフォーマンスを著しく向上させ得る。一実施形態において、例えば、完了するのにデータミスアラインメント検出及び処理なしで約1236秒を要する作業負荷が、本発明の実施形態に従うデータミスアラインメント検出及び処理を用いて完了するのに約133秒しか要しない。本発明のいくつかの実施形態は、説明された利益に加えて又は代わりに、多様な他の利益を与える。

30

【0042】

本発明のいくつかの実施形態は、例えば、装置、例えばコンピューティングプラットフォーム110、プロセッサ141、又は他の適切な機械によって実行された場合に、機械に、本発明の実施形態に従う方法及び／又はオペレーションを実行させる命令又は命令のセットを記憶する機械可読メディア又は製品を用いて実装されてよい。そのような機械は、例えば、任意の適切なプロセッシングプラットフォーム、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、プロセッシングデバイス、コンピューティングシステム、プロセッシングシステム、コンピュータ、プロセッサ、又は同種のものを含んでよく、ハードウェア及び／又はソフトウェアの任意の適切な組合せを用いて実装されてよい。機械可読メディア又は製品は、例えば、任意のタイプのメモリユニット（例えば、メモリユニット142）、メモリデバイス、メモリ製品、メモリメディア、ストレージデバイス、ストレージ製品、ストレージメディア及び／又はストレージユニット（例えば、ストレージユニット143）、例えば、メモリ、リムーバブル又はノンリムーバブルメディア、消去可能又は消去不可能なメディア、書き込み可能又は再書き込み不可能なメディア、デジタル又はアナログメディア、ハードディスク、フレキシブルディスク、コンパクトディスクリードオンリーメモリ（CD-ROM）、コンパクトディスクレコーダブル（

40

50

CD-R)、コンパクトディスクリライタブル(CD-RW)、光ディスク、磁気メディア、種々のタイプのデジタル多目的ディスク(DVD)、テープ、カセット、或いは同種のものを含んでよい。命令は、任意の適切なタイプのコード、例えば、ソースコード、コンパイル済みコード、インタープリタコード、実行コード、静的コード、動的コード、又は同種のものを含んでよく、任意の適切な高レベル、低レベル、オブジェクト指向、視覚的、コンパイル型、及び／又はインタープリタ型プログラミング言語、例えば、C、C++、Java(登録商標)、BASIC、Pascal、Fortran、Cobol、アセンブリ言語、マシンコード、又は同種のものを用いて実装されてよい。

【0043】

本発明のいくつかの実施形態は、特定の用途又は特定の設計要件に適するように、ソフトウェア、ハードウェア、又はソフトウェア及び／又はハードウェアの任意の組合せによって実装されてよい。本発明の実施形態は、全体的に又は部分的に、互いに切り離された、又は互いに一体化された複数のユニット及び／又は複数のサブユニットを含んでよく、技術的に知られたように、特定の、多目的の、又は汎用のプロセッサ又はデバイスを用いて実装されてよい。本発明のいくつかの実施形態は、データの一時的又は長期記憶のため、又は具体的な実施形態のオペレーションを促進すべく、複数のバッファ、複数のレジスタ、複数のストレージユニット、及び／又は複数のメモリユニットを含んでよい。

【0044】

本発明のいくつかの機能が本明細書に示され及び説明されたが、多くの変更、置き換え、改変、及び／又は均等物が、当業者に考えられる。したがって、添付の請求項は、全てのそのような変更及び／又は改変を含むことが意図されていることが理解されるべきである。

(項目1)

第1コンピューティングプラットフォームに適合する第1フォーマットから第2コンピューティングプラットフォームに適合する第2フォーマットに変換された1つのコードブロックの実行がもたらすミスアラインデータアクセスを検出する段階と、

前記ミスアラインデータアクセスに従って、前記コードブロックを修正する段階とを備える方法。

(項目2)

検出する段階は、前記コードブロックの実行が前記ミスアラインデータアクセスをもたらすか否かを検出すべく、前記コードブロックのインストゥルメンテーションを実行する段階を有する

項目1に記載の方法。

(項目3)

検出する段階は、実行が前記ミスアラインデータアクセスをもたらす1つの命令の位置を検出すべく、前記コードブロックの中の少なくとも1つの命令のインストゥルメンテーションを実行する段階を有する

項目2に記載の方法。

(項目4)

検出する段階は、実行が前記ミスアラインデータアクセスをもたらす1つの命令の位置を検出すべく、前記コードブロックの中の少なくとも1つの命令のインストゥルメンテーションを実行する段階を有する

項目1に記載の方法。

(項目5)

修正する段階は、実行が前記ミスアラインデータアクセスを処理するコードシーケンスに前記コードブロックの実行を分岐させる1つの命令を、前記コードブロックに追加する段階を有する

項目1に記載の方法。

(項目6)

修正する段階は、前記コードブロックの後続の実行におけるミスアラインデータアクセ

10

20

30

40

50

スを処理すべく前記コードブロックを修正する段階を有する
項目 1 に記載の方法。

(項目 7)

前記コードブロックを前記第 1 フォーマットから前記第 2 フォーマットに変換する段階
をさらに備える項目 1 に記載の方法。

(項目 8)

検出する段階は、3 2 ビットベースのコンピューティングプラットフォームに適合する
フォーマットから 6 4 ビットベースのコンピューティングプラットフォームに適合するフ
ォーマットに変換されたコードブロックの実行がもたらす、ミスアラインデータアクセス
を検出する段階を有する
項目 1 に記載の方法。

10

(項目 9)

第 1 コンピューティングプラットフォームに適合する第 1 フォーマットから第 2 コンピ
ューティングプラットフォームに適合する第 2 フォーマットに変換された 1 つのコードブ
ロックの実行がもたらすミスアラインデータアクセスを検出し、前記ミスアラインデー
タアクセスに従って、前記コードブロックを修正する 1 つのプロセッサ
を備える装置。

(項目 10)

前記プロセッサは、前記コードブロックの実行が前記ミスアラインデータアクセスをも
たらすか否かを検出すべく、前記コードブロックのインストゥルメンテーションを実行す
ることが可能である
項目 9 に記載の装置。

20

(項目 11)

前記プロセッサは、実行が前記ミスアラインデータアクセスをもたらす 1 つの命令の位
置を検出すべく、前記コードブロックの中の少なくとも 1 つの命令のインストゥルメンテ
ーションを実行することが可能である
項目 10 に記載の装置。

(項目 12)

前記プロセッサは、実行が前記ミスアラインデータアクセスをもたらす 1 つの命令の位
置を検出すべく、前記コードブロックの中の少なくとも 1 つの命令のインストゥルメンテ
ーションを実行することが可能である
項目 9 に記載の装置。

30

(項目 13)

前記プロセッサは、実行が前記ミスアラインデータアクセスを処理するコードシーケ
ンスに前記コードブロックの実行を分岐させる 1 つの命令を、前記コードブロックに追加
することが可能である
項目 9 に記載の装置。

(項目 14)

前記プロセッサは、前記コードブロックの後続の実行におけるミスアラインデータアク
セスを処理すべく前記コードブロックを修正することが可能である
項目 9 に記載の装置。

40

(項目 15)

前記プロセッサは、前記ミスアラインデータアクセスを検出する前に、前記コードブロ
ックを前記第 1 フォーマットから前記第 2 フォーマットに変換することが可能である
項目 9 に記載の装置。

(項目 16)

前記第 1 コンピューティングプラットフォームは 3 2 ビットベースのコンピューティン
グプラットフォームであり、前記第 2 コンピューティングアーキテクチャは 6 4 ビットベ
ースのコンピューティングプラットフォームである
項目 9 に記載の装置。

50

(項目 17)

第1コンピューティングプラットフォームに適合する第1フォーマットから第2コンピューティングプラットフォームに適合する第2フォーマットに変換された1つのコードブロックの実行がもたらすミスアラインデータアクセスを検出し、前記ミスアラインデータアクセスに従って、前記コードブロックを修正する1つのプロセッサと、

前記プロセッサに動作可能に連携され、前記コードブロックの少なくとも一部を記憶する1つのダイナミックランダムアクセスメモリと
を備えるコンピューティングプラットフォーム。

(項目 18)

前記プロセッサは、前記コードブロックの実行が前記ミスアラインデータアクセスをもたらすか否かを検出するべく、前記コードブロックのインストールメンテーションを実行することが可能である

項目17に記載の装置。

(項目 19)

前記プロセッサは、実行が前記ミスアラインデータアクセスをもたらす1つの命令の位置を検出するべく、前記コードブロックの中の少なくとも1つの命令のインストールメンテーションを実行することが可能である

項目18に記載の装置。

(項目 20)

前記プロセッサは、実行が前記ミスアラインデータアクセスをもたらす1つの命令の位置を検出するべく、前記コードブロックの中の少なくとも1つの命令のインストールメンテーションを実行することが可能である

項目17に記載の装置。

(項目 21)

記憶された複数の命令のセットを備える機械可読メディアであって、前記複数の命令は、機械によって実行された場合に、前記機械に、

第1コンピューティングプラットフォームに適合する第1フォーマットから第2コンピューティングプラットフォームに適合する第2フォーマットに変換された1つのコードブロックの実行がもたらすミスアラインデータアクセスを検出する段階と、

前記ミスアラインデータアクセスに従って、前記コードブロックを修正する段階と
を備える方法を実行させる機械可読メディア。

(項目 22)

検出する段階をもたらす前記複数の命令は、前記コードブロックの実行が前記ミスアラインデータアクセスをもたらすか否かを検出するべく、前記コードブロックのインストールメンテーションを実行する段階をもたらす

項目21に記載の機械可読メディア。

(項目 23)

検出する段階をもたらす前記複数の命令は、実行が前記ミスアラインデータアクセスをもたらす1つの命令の位置を検出するべく、前記コードブロックの中の少なくとも1つの命令のインストールメンテーションを実行する段階をもたらす

項目22に記載の機械可読メディア。

(項目 24)

検出する段階をもたらす前記複数の命令は、実行が前記ミスアラインデータアクセスをもたらす1つの命令の位置を検出するべく、前記コードブロックの中の少なくとも1つの命令のインストールメンテーションを実行する段階をもたらす

項目21に記載の機械可読メディア。

(項目 25)

前記複数の命令は、1つのトランスレータの少なくとも一部を構成する
項目21に記載の機械可読メディア。

(項目 26)

10

20

30

40

50

前記複数の命令は、1つの実行レイヤの少なくとも一部を構成する
項目21に記載の機械可読メディア。

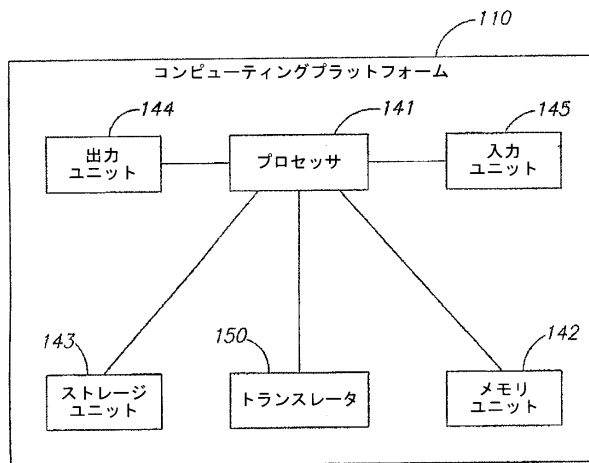
(項目27)

前記複数の命令は、1つのオペレーティングシステムの少なくとも一部を構成する
項目21に記載の機械可読メディア。

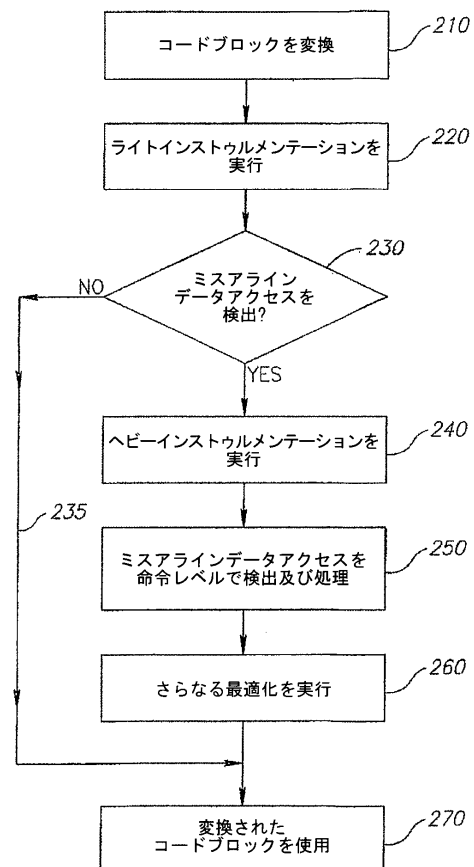
(項目28)

前記複数の命令は、1つのコンパイラの少なくとも一部を構成する
項目21に記載の機械可読メディア。

【図1】



【図2】



フロントページの続き

(72)発明者 チェン、ジャンーピン

中華人民共和国、２００３３６、シャanghai、バイファ ストリート レーン ３４５、ディン
ジャン ユアン ビルディング ２９、ルーム ２０２

審査官 三坂 敏夫

(56)参考文献 特開平１０－１１６２００（ＪＰ，Ａ）

特開２０００－０６６９０２（ＪＰ，Ａ）

国際公開第０３／０３６４７５（ＷＯ，Ａ１）

特表２００５－５０７１１４（ＪＰ，Ａ）

米国特許第０６０６４８１５（ＵＳ，Ａ）

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 9 / 3 4

G 0 6 F 9 / 4 5 5

G 0 6 F 1 2 / 0 4