

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3842218号

(P3842218)

(45) 発行日 平成18年11月8日(2006.11.8)

(24) 登録日 平成18年8月18日(2006.8.18)

(51) Int. Cl. F I
G06F 9/38 (2006.01) G06F 9/38 310A
 G06F 9/38 370X

請求項の数 7 (全 16 頁)

| | | | |
|---------------|-------------------------------|-----------|----------------------|
| (21) 出願番号 | 特願2002-561678 (P2002-561678) | (73) 特許権者 | 590000248 |
| (86) (22) 出願日 | 平成14年1月4日(2002.1.4) | | コーニンクレッカ フィリップス エレク |
| (65) 公表番号 | 特表2004-519028 (P2004-519028A) | | トロニクス エヌ ヴィ |
| (43) 公表日 | 平成16年6月24日(2004.6.24) | | Koninklijke Philips |
| (86) 国際出願番号 | PCT/IB2002/000025 | | Electronics N. V. |
| (87) 国際公開番号 | W02002/061574 | | オランダ国 5621 ペーアー アイ |
| (87) 国際公開日 | 平成14年8月8日(2002.8.8) | | ドーフエン フルーネヴァウツウェッハ |
| 審査請求日 | 平成17年1月4日(2005.1.4) | | 1 |
| (31) 優先権主張番号 | 01200333.1 | | Groenewoudseweg 1, 5 |
| (32) 優先日 | 平成13年1月30日(2001.1.30) | (74) 代理人 | 100075812 |
| (33) 優先権主張国 | 欧州特許庁 (EP) | | 弁理士 吉武 賢次 |
| | | (74) 代理人 | 100088889 |
| | | | 弁理士 橘谷 英俊 |

最終頁に続く

(54) 【発明の名称】 命令フェッチ制御ビットを有するコンピュータ命令

(57) 【特許請求の範囲】

【請求項 1】

処理ユニットおよびメモリを有し、前記処理ユニットは前記メモリからメモリ・ラインをフェッチして前記メモリ・ラインからの命令を実行するように構成され、各々のメモリ・ラインは全体をフェッチされて複数の命令を保持することができ、少なくとも1つの命令は、前記処理ユニットが現在のメモリ・ラインからの命令を処理している時に、後続のメモリ・ラインへの境界を越えることによって処理の一部分が受ける影響の程度を前記処理ユニットがどのように制御すべきかを明示的に知らせる情報を含み、前記処理ユニットは、前記情報によって知らされたように前記一部分を制御することによって前記情報に応答するように構成され、

前記情報は、更に命令ポインタが前記現在のメモリ・ラインにおける前記命令の後の位置から前記後続のメモリ・ラインのスタートへ更新されるべきか否かを明示的に知らせ、それによって前記現在のメモリ・ラインの命令に続く情報がスキップされ、前記処理ユニットが、前記情報に応答して前記後続のメモリ・ラインのスタートへ前記命令ポインタを更新するように構成されているコンピュータ・システム。

【請求項 2】

前記情報は前記命令の処理中に前記後続のメモリ・ラインがプリフェッチされるべきか否かを明示的に知らせ、前記処理ユニットは前記情報に応答して前記後続のメモリ・ラインのプリフェッチをスタートするように構成されている請求項 1 記載のコンピュータ・システム。

10

20

【請求項 3】

前記情報はプリフェッチ・ビットを含み、前記プリフェッチ・ビットの値は、前記後続のメモリ・ラインがプリフェッチされるべきか否かを明示的に知らせる請求項 2 記載のコンピュータ・システム。

【請求項 4】

前記情報は、前記命令が分岐命令から到達された時に、前記命令の処理が停止されるべきか否かを明示的に知らせ、前記命令の一部を含む前記後続のメモリ・ラインをフェッチするため処理が停止され、前記処理ユニットは、前記命令が前記分岐命令から到達された時に前記情報に回答して停止する請求項 1 乃至 3 のいずれかに記載のコンピュータ・システム。

10

【請求項 5】

前記処理ユニットは、前記命令からの操作を並列に機能ユニットへ出す 2 つ以上の発行スロットを含む V L I W 処理ユニットであり、前記命令は、2 つ以上の操作を含むことができる V L I W 命令であり、前記命令は前記情報を指定するため操作から区別されるフィールドを含む請求項 1 乃至 4 のいずれかに記載のコンピュータ・システム。

【請求項 6】

前記フィールドは、前記情報に加えて、どの発行スロットのために前記命令が操作を含むかを指定する解凍コードを含む請求項 5 記載のコンピュータ・システム。

【請求項 7】

処理ユニットおよびメモリを有し、前記処理ユニットは前記メモリからメモリ・ラインをフェッチして前記メモリ・ラインからの命令を実行するように構成され、各々のメモリ・ラインは全体をフェッチされて複数の命令を保持することができ、少なくとも 1 つの命令は、前記処理ユニットが現在のメモリ・ラインからの命令を処理している時に、後続のメモリ・ラインへの境界を越えることによって処理の一部が受ける影響の程度を前記処理ユニットがどのように制御すべきかを明示的に知らせる情報を含むコンピュータ・システムにおいて命令を処理する方法であって、

20

各々のメモリ・ラインを全体としてフェッチするステップと、

現在のメモリ・ラインからの命令を処理するステップと、

処理の間に前記命令から情報を読み取るステップと、

前記情報によって知らされたように前記一部分を制御するステップと、

30

を備え、

前記制御するステップは、前記後続のメモリ・ラインのスタートへスキップするプログラム・カウンタを含むか、又は前記命令が分岐目標として到達された時に停止される処理を含むことを特徴とする方法。

【発明の詳細な説明】

【0001】

(技術分野)

本発明は、コンピュータ・システムにおける命令のフェッチに関する。

【0002】

(発明の背景)

40

V L I W (V e r y L a r g e I n s t r u c t i o n W o r d) 命令を圧縮することは知られている。コード・サイズを縮小するため、通常、可変長命令フォーマットが使用される。しかし、そのような可変長命令フォーマットを処理するために必要な命令フェッチ・ハードウェアは、複雑になる傾向がある。命令のフェッチを制御するため、様々な解決法が応用されてきた。

【0003】

W O 9 8 / 0 2 7 9 8 は、命令整列ユニットを有するスーパースカラ・マイクロプロセッサを開示する。命令整列ユニットは、決まった数の命令を、命令キャッシュから複数のデコード・ユニットの各々へ転送する。命令はプレデコード・タグに従って選択される。プレデコード・タグは、どのバイトが命令の最初のバイトであることを示すスタートバイト・

50

ビットを含む。従って、命令の始まりは、圧縮された命令を解凍する目的のために識別される。

【0004】

トリメディア・プロセッサに関連する米国特許第6,134,633号およびWO9743710には、可変長VLIWを高クロック速度で伸長する方法が開示される。命令デコーディング・ロジックはパイプラインにされる。これは、操作セグメントの1サイクル前にヘッダが利用可能でなければならないことを意味する。従って、所与の可変長命令のヘッダ・ビットが、その特定の命令に先行して可変長命令と一緒にプログラム・メモリの中に記憶される。しかし、これは分岐目標の場合に特別の結果を生じる。分岐目標へは、異なった起点から入ることができるので、先行する命令を独自に決定することはできない。これは、分岐目標の解凍が、分岐が取られた後でのみスタートできることを意味し、実行サイクルの損失を生じることを意味する。そのような損失を防止するため、分岐目標は圧縮されない。これはコード・サイズの高度の不利点となる。

10

【0005】

VLIWプロセッサのデンスコードを得るため、VLIW命令は、好ましくは、圧縮されるべきである。それによって、1クロック・サイクル当たり可変数の命令ビットが必要となる。これらの命令ビットを時間どおりにフェッチおよびデコードするためには、命令をデコードする時に幾つのビットが必要であるか、従って次の命令がどこからスタートするかを、命令ビット自身から推定できるデコーダが必要となる。このプロセスは、デコーダ内にフィードバック・ループを含む。その場合、現在の命令のデコーディングは、次の命令のデコーディング、即ち次の命令が置かれているプログラム・メモリ内の位置の決定に強く影響する。

20

【0006】

フィードバック・ループは、可能性として処理を遅らせる。特に、命令が、複数の命令(の一部)を含むことのできるメモリ・ラインからバッチでフェッチされる時にそうである。新しいメモリ・ラインをいつフェッチ(プリフェッチ)すべきかを決定するためには、次の命令のメモリ内の位置が、いつメモリ・ラインの間の境界を横切るかを決定することが必要である。

【0007】

パイプライン、即ち現在の命令を実行しながら後続の命令を解凍することによって、フィードバック・ループをスピードアップすることは不可能である。なぜなら、これは、命令をフェッチできるレートに直接影響を与えるからである。例えば、1つのパイプライン・ステージを追加することは、1クロック・サイクルごとに1つの命令ではなく、2クロック・サイクル当たり1つの命令しかフェッチできないことを意味するであろう。

30

【0008】

他の欠点は、分岐目標のフェッチを時間どおりに行なわせるため、目標命令の全体が単一のプログラム・メモリ・ラインの上にフィットしなければならないことである。(新しい命令がサイクルごとに実行へ移される時に、即ち、先行する命令の処理の第1のステージが完了すると、直ちに命令が処理の第1のステージ(解凍を含む)のために利用可能であれば、命令は「時間どおりに」フェッチされたものと考えられる。)もし分岐目標が圧縮されなければ、命令がメモリ・ラインの中で他の命令に続く時に、命令がメモリ・ラインの中にフィットする確率は小さい。従って、そのような目標命令は、他のプログラム・メモリ・ラインのスタートへ再整列させられ、パディング・ビットが元のメモリ・ラインへ付加される。それによって、コード・サイズは更に増加する。

40

【0009】

(発明の概要)

本発明の目的は、デンスコードを可能にし、同時に処理パフォーマンスを比較的高レベルに維持するコンピュータ・システムを提供することである。

【0010】

本発明の更なる目的は、フェッチを制御するため通常必要となる幾つかのコストの高い計

50

算を除去するコンピュータ・システムを提供することである。

【0011】

本発明の他の目的は、そのようなコンピュータ・システムで使用されるプログラムを実行する方法、そのようなコンピュータ・システムのためのプログラム、およびそのようなコンピュータ・システムのためのプログラムを生成する方法を提供することである。

【0012】

本発明は、処理ユニットおよびメモリを含み、前記処理ユニットは前記メモリからメモリ・ラインをフェッチして前記メモリ・ラインからの命令を実行するように構成され、各々のメモリ・ラインは全体をフェッチされて複数の命令を保持することができ、少なくとも1つの命令は、前記処理ユニットが現在のメモリ・ラインからの命令を処理している時に、後続のメモリ・ラインへの境界を越えることによって処理の一部分が受ける影響の程度を前記処理ユニットがどのように制御すべきかを明示的に知らせる情報を含み、前記処理ユニットは、前記情報によって知らされたように前記一部分を制御することによって前記情報に応答するように構成されているコンピュータ・システムを提供する。

10

【0013】

命令は、境界の横断がどのように制御されなければならないかを、命令の中で明示的に知らせるので、コンピュータ・システムは、前もって、遅れることなく、境界横断の影響を処理するように進行することができる。例えば、或る適切な明示的信号を使用することによって、新しいメモリ・ラインは、必要な時に、遅れることなくフェッチされることができ、他の明示的信号を使用することによって、プログラム・カウンタは、パディング・ビットが処理される前に、新しいプログラムラインへ再整列することができ、更に他の明示的信号を使用することによって、もし分岐目標を処理するため他のメモリ・ラインが必要とされるならば、処理は即時に停止されることができ。

20

【0014】

「明示的信号」は、例えば、メモリ・ラインをプリフェッチする必要性が、現在のプログラム・カウンタおよび計算された命令長から続く時のような、暗黙的信号から区別される。従って、プリフェッチに関連する操作の制御は、明示的信号によって単純化およびスピードアップがなされる。暗黙的信号を有する情報と対比して、分離された明示的信号を有する情報は、明示的信号によって制御されるアクションを除いて、アクションを制御する十分な情報を提供することはない。

30

【0015】

好ましくは、プログラム内の全ての命令は、ロード命令、記憶命令、加算命令など、命令のタイプに関係なく、そのような情報を含む。実施形態において、情報の一部分又は全体は、或る命令から省略されてよい。例えば、実施形態において、プリフェッチおよび停止に関する信号は、メモリ・ラインのスタートに置かれる命令から省略される。

【0016】

本発明に従ったコンピュータ・システムの実施形態において、現在のメモリ・ラインの中の命令は、後続の命令の少なくとも一部分が後続のメモリ・ラインに記憶されているため後続のメモリ・ラインをフェッチしなければならないか否かを明示的に知らせる。従って、後続のメモリ・ラインは、必要な時に、遅れることなく、フェッチされることができ。これは、次のメモリ・ラインを待つことに起因する命令サイクルの損失を防止し、また、メモリ・ラインの不必要なフェッチを防止する。好ましくは、次のメモリ・ラインをフェッチする信号は、この目的に専用の、命令の中のビットである。

40

【0017】

本発明に従ったコンピュータ・システムの実施形態において、現在のメモリ・ラインの中の現在の命令は、プログラム・カウンタが、現在のメモリ・ラインの残りの部分にあるパディングをスキップして、後続のメモリ・ラインのスタートへ進むべきか否かを明示的に知らせる。これは、特に、次の命令（後続のメモリ・ラインのスタートに存在する）が、現在のメモリ・ラインの現在の命令の後にフィットしない非圧縮分岐目標命令である時に有用である。再整列する信号を使用することによって、この命令の実行は、再整列するか

50

否かをパディングから決定する遅延を伴うことなくスタートされることができる。好ましくは、再整列する信号は、この目的に専用の、命令の中のビットである。

【0018】

本発明に従ったコンピュータ・システムの実施形態において、現在のメモリ・ラインの中の現在の命令は、もしその命令が分岐目標として到達されるのであれば、後続のメモリ・ラインから命令の残りをフェッチするため命令の処理を停止すべきか否かを明示的に知らせる。これは、異なったメモリ・ラインにまたがる分岐目標命令の処理を簡単にする。停止ビットによって、先行命令の後にすぐ続いて異なったメモリ・ラインへ伸びるように、分岐目標を置くか(メモリ空間を節約するが、停止に起因するスピードの費用がかかる)、分岐目標命令をメモリ・ラインのスタートに再整列して、ただ1つのメモリ・ラインをフェッチした後で、分岐目標命令がスタートから実行されるようにし、また非圧縮形式で記憶できるようにするか(メモリ空間の費用で1クロック・サイクルを得る)を選択することができる。好ましくは、この選択は、命令が分岐目標として使用される頻度に依存し、頻度の多い分岐目標は再整列される。好ましくは、停止する信号は、この目的に専用の、命令の中のビットである。

10

【0019】

実施形態において、コンピュータ・システムはVLIWシステムである。その命令は、異なった機能ユニットのための操作およびヘッダを含み、ヘッダは明示的信号を有する情報を含む。従って、ヘッダのオーバーヘッドは、異なった操作によって共用される。本発明は、特に、VLIWプロセッサに有用である。なぜなら、VLIWプロセッサは、メモリ・ライン境界を横断する可能性が高い比較的長い命令を有するからである。

20

【0020】

本発明は、更に、そのようなコンピュータ・システムのためのコンピュータ・プログラム(コンピュータ読み取り可能媒体によって移送される)に関する。そのようなプログラムにおいて、コンピュータ・システムがメモリ・ラインを効率的に処理するように強制するため、明示的信号は、命令がメモリ・ラインに伸びる様式に従って設定される。例えば、正しい時間にメモリ・ラインをプリフェッチし、もしメモリ・ラインがパディングを含むならば再整列し、又は、もし分岐目標として到達される命令が複数のラインへ伸びるならば停止するように、システムへ指令を与える。本発明は、更に、そのようなコンピュータ・プログラムを生成する方法に関する。

30

【0021】

コンピュータ・システムは、例えば、組み込みシステム分野におけるように、単一のチップの上に多数のプロセッサ(例えばDSP)を含んでよい。しかし、コンピュータ・システムは、1つのコンピュータ・プロセッサ、例えば、ローカル・コンピュータ装置、例えばパーソナル・コンピュータに置かれた中央プロセッサ・ユニット(CPU)を含むか、代替的に、ローカル又はグローバルのコンピュータ・ネットワークの中で相互接続された多数のコンピュータ装置に置かれた多数のプロセッサ・ユニットを含んでよい。代替的に、それは、ローカル・コンピュータ装置から離れた中央コンピュータに置かれた処理ユニットを含んでよい。中央コンピュータ装置は、例えばサーバ装置であってよい。

40

【0022】

好ましくは、各々の命令はヘッダを設けられる。即ち、各々の命令へ、命令ワードに関連した情報を含む多数のビットが付加される。ヘッダは少なくとも1つのフェッチ制御ビットを含む。更に、ヘッダは、命令ワードをどのように解凍するかに関連した情報、例えば、どこで非操作(NOP)ビットをどのように再挿入するかに関する情報を含んでよい。少なくとも1つのフェッチ制御ビットの値は、続いて実行される命令ワードの解凍および実行を可能にする十分なビットがフェッチされたか否かに関する情報を提供する。即ち、ヘッダは、続いて実行される命令ワードに関する情報、およびヘッダが属する命令ワードに関する情報を含む。従って、フェッチ制御ビットは、次のメモリ・ラインがフェッチされるべきか否か、分岐目標がフェッチされる可能性があるか否か、そしてこの場合、分岐目標の解凍および実行を可能にするため幾つのメモリ・ラインをフェッチするのが必要で

50

あるかに関する情報を提供し、および/又は、フェッチ制御ビットは、圧縮されたVLIWのコード・サイズおよびプロセッサ・パフォーマンスを最適化するため、即ち、コードをできるだけ稠密(デンス)にしてコンピュータ・ロジック回路をできるだけ効率的に使用するため、どのメモリ・ラインが、どのようにして、また、いつフェッチされるべきかに関する適切な他の情報を提供してよい。

【0023】

命令ワードを形成する全てのビットがフェッチされた時に、十分なビットがフェッチされている。即ち、解凍および実行がスタートされる前に、全ての圧縮された命令ワードがメモリからフェッチされていなければならない。そうでない場合、コンピュータ・ロジック回路は、解凍および実行の間、フェッチされる命令ワードの残りの部分を待たなくてはならず、それによってコンピュータ・ロジック回路の非効率的な使用を生じる。従って、もし現在の命令ワードの実行が完了した時に、直ちに、続いて実行される命令ワードが完全に利用可能であることを確認することができれば、コンピュータ・ロジック回路の非常に効率的な利用が提供されるであろう。

10

【0024】

更に、本発明は、前述したようなコンピュータ・プログラムを含むコンピュータ読み取り可能媒体を提供する。コンピュータ読み取り可能媒体は、例えば、静止コンピュータ読み取り可能媒体、例えば、ハード・ディスク、読み出し専用メモリ(ROM)、例えばEPROM又はEEPROM、および/又は他の任意適切な静止コンピュータ読み取り可能媒体であってよい。代替的又は追加的に、それは携帯用コンピュータ読み取り可能媒体、例えば、コンパクト・ディスク(CD)、例えばCD-ROM、フロッピー(登録商標)・ディスク、DVDディスク、磁気テープ、ZIP媒体、携帯用メモリカード、および/又は他の任意適切な携帯用コンピュータ読み取り可能媒体であってよい。

20

【0025】

本発明に従ったシステム、方法、およびプログラムのこれらおよび他の有利な様相は、図面を使用して説明されるであろう。

【0026】

(好ましい実施形態の説明)

図3はコンピュータ・システムを示す。本発明を説明するために使用されるシステム部分のみが示される。システムは、メモリ30、命令発行ユニット32、および命令処理ユニット34を含む。命令発行ユニット32は、ヘッダ選択ユニット320、命令選択ユニット322a~322c、選択制御ユニット324、アドレッシング・ユニット326、先行ライン・レジスタ328、およびNOPソース329を含む。例として、3つの命令選択ユニット322a~322cが示されるが、それより多いか少ない命令選択ユニットが使用されてよい。アドレッシング・ユニット326は、メモリ30へ結合されたアドレス出力を有する。メモリ30は、データ出力として命令選択ユニット322a~322cへ結合される。NOPソース329も命令選択ユニット322a~322cへ結合される。命令選択ユニット322a~322cの各々は、命令処理ユニット34の多数の発行スロット入力340a~340cのそれぞれに結合された出力を有する。

30

【0027】

メモリ30のデータ出力は、更に、ヘッダ選択ユニット320および先行ライン・レジスタ328の入力へ結合される。先行ライン・レジスタ328は、ヘッダ選択ユニット320および命令選択ユニット322a~322cへ結合された出力を有する。ヘッダ選択ユニット320は、命令処理ユニット34、選択制御ユニット324、命令選択ユニット322a~322c、およびアドレッシング・ユニット326へ結合された出力を有する。選択制御ユニットは、ヘッダ選択ユニット320および命令選択ユニット322a~322cへ結合された出力を有する。

40

【0028】

本発明を説明するために使用されるシステム部分のみが示される。例えば、様々なコンポーネントへのクロック信号接続は示されず、分岐、例外処理などの接続も図示されない

50

。更に、システムは、メモリ30からの情報を処理ユニット34でバッファするための様々なレジスタ(図示されない)を含むことになる。これらのレジスタによって、命令発行ユニット32との間で送信又は受信された情報が命令発行ユニット32によって処理又は組み立てられている間に、メモリ30および処理ユニット34は他のアクションを実行することができる。

【0029】

動作において、命令発行ユニット32は、メモリ30からメモリ・ラインをフェッチし、メモリ・ラインからの命令を解凍し、命令を実行のために命令処理ユニット34へ送る。命令処理ユニット34は、命令をデコードし、命令のオペランドをフェッチし、命令を実行し、命令の結果を書き戻す。例として、本発明はVLIWプロセッサについて説明されることになる(VLIW=Very Large Instruction Word、極長命令ワード)。VLIWプロセッサは、並列に実行するそれぞれの命令(通常、「操作」と呼ばれる)を受け取るための多数の発行スロット340a~340cを有する。

10

【0030】

命令発行ユニット32の中で、アドレッシング・ユニット326はメモリ・ラインのアドレスをメモリ30へ出力する。メモリからの各々のメモリ・ライン(即ち、アドレッシング・ユニット326からのアドレスによってアドレスされた一連の、例えば、8つのロケーションからのデータ)出力は、命令選択ユニット322aから322cへ印加される。各々の命令選択ユニット322a~322cは、命令処理ユニット34の発行スロット340a~340cのそれぞれの1つのための操作又はNOPを選択する。操作は、メモリ30から受け取られたメモリ・ラインから選択されるか、先行ライン・レジスタ328に記憶されたメモリ30からの先行メモリ・ライン、又はNOPソース329から選択される。NOPソース329はNOP(無操作)命令を生成する(実際には、NOP命令は非常に簡単なので、それらは命令選択ユニット322a~322cの内部で生成されてよく、NOPソース329は省略されてよい)。

20

【0031】

命令選択ユニット322a~322cによる選択は、選択制御ユニット324およびヘッダの内容によって制御される。ヘッダはヘッダ選択ユニット320によって命令から選択される。ヘッダ選択ユニット320は、メモリ30又は先行ライン・レジスタ328から受け取られたメモリ・ラインの中の命令のヘッダからの情報を供給する。このヘッダは、発行スロット340a~340cのどれが無操作命令を受け取るべきか、また、発行スロット340a~340cのどれがメモリ・ラインの命令からの操作を受け取るべきかを指定する。原則として、発行スロット340a~340cの連続したスロットに対する操作は、メモリ・ライン内の連続した位置に置かれ、前記連続した位置は、もし命令が複数のメモリ・ラインに伸びていれば、次のメモリ・ラインの連続位置へ続いている。特定の発行スロット340a~340cに対するメモリ・ラインの中の操作の位置は、ヘッダによって指示されたように、メモリ・ラインから操作を受け取る先行発行スロット340a~340cの数に依存する。選択制御ユニット324は、メモリ・ラインの中の命令のスタートの位置を、命令選択ユニット322a~322cおよびヘッダ選択ユニット320へ知らせる。選択制御ユニット324は、知らされた位置を、各々の時間に、無操作ではなくメモリ・ラインからの操作を受け取る発行スロット340a~340cの数に対応する量だけ増加する。

30

40

【0032】

ヘッダ選択ユニット320内のヘッダは、後続の命令を含むメモリ・ラインのフェッチに必要なハードウェアを単純化するヘッダからの追加情報を供給する。ヘッダ選択ユニット320からの最初のビットは、「プリフェッチ(P)ビット」である。このビットは、実行される次の命令がメモリ30から現在受け取られているメモリ・ラインに含まれているかどうか、又は、命令の一部又は全体が次のメモリ・ラインに含まれているかどうかを示す。後者の場合、アドレッシング・ユニット326が、次のメモリ・ラインのためのプリフェッチ・コマンドをメモリ30へ発行することによって、プリフェッチ・ビットに応答

50

する。次のメモリ・ラインがプリフェッチされる時に、古い記憶は先行ライン・レジスタ 3 2 8 へコピーされ、従って先行するラインの中にある命令部分は、命令選択ユニット 3 2 2 a ~ 3 2 2 c によって、先行ライン・レジスタ 3 2 8 からアクセスされることができ

【 0 0 3 3 】

ヘッダ選択ユニット 3 2 0 からの 2 番目のビットは「再整列 (R) ビット」である。再整列ビットは、実行されるべきでないパディングが命令に続いているかどうかを示す。もし続いていれば、選択制御ユニット 3 2 4 は、指示された位置を命令選択ユニット 3 2 2 a ~ 3 2 2 c へ更新することによって再整列ビットに応答し、従って、その位置は次の命令のために次のメモリ・ラインのスタートを指すことになる。

10

【 0 0 3 4 】

ヘッダ選択ユニット 3 2 0 からの 3 番目のビットは、「停止 (S) ビット」である。停止ビットは、複数のラインの最後がまだフェッチされていない場合に、現在の命令が複数のラインへ伸びているかどうかを示す。これは、典型的には、再整列に起因するメモリ空間の損失を防止するように、命令が複数のラインへ伸びる分岐目標である場合に起こる。この場合、命令処理ユニット 3 2 は、命令の残りを含む次のメモリ・ラインをフェッチするために必要な期間の間実行を停止することによって、停止ビットに応答することになる。

【 0 0 3 5 】

もちろん、図 1 は、コンピュータ・システムを実現する方式の 1 つの例を示すにすぎない。多くのバリエーションが可能である。例えば、メモリ (又は、図示されていないメモリ・バッファ・レジスタ) からのメモリ・ラインを直接使用する代わりに、バッファの中にメモリ・ラインを記憶し、新しい命令が処理される度に、バッファの内容が選択制御ユニット 3 2 4 の制御のもとでシフトされ、従って、命令のスタート部 (ヘッダを含む) が常にバッファ内の同じ位置にあるようにすることができる。この場合、特別のヘッダ選択ユニットは必要でない。最初の発行スロットに対する命令選択ユニット 3 2 2 a は、NOP を有する発行スロットが、幾つ最初の発行スロットに先行するかを計算する必要がなく、従って他の命令選択ユニット 3 2 2 b ~ 3 2 2 c よりも、はるかに簡単であることができる。この命令ユニットは、ヘッダの制御のもとで、命令からの最初の操作又は NOP を選択する必要があるだけである。

20

【 0 0 3 6 】

図 1 に示される $V L^2 I W$ 命令フォーマットは、ヘッダ 1、およびそれに続いた多数の操作セグメント 2 を含む。操作セグメント 2 の数は、プロセッサが並列に発行することのできる操作の数、この場合は C、を超過してはならない。ヘッダ 1 は C 個の伸長制御ビット $E_1 \dots E_C$ を含む。i 番目のスロットが無操作 (NOP) を実行する場合、伸長制御ビット E_i は「 0 」である。そうでない場合、それは「 1 」であり、 $V L^2 I W$ は発行スロット i のために操作セグメントを含む。

30

【 0 0 3 7 】

スロットが NOP 操作を実行する時に、それは問題のスロットが問題のクロック・サイクルの間アイドルであることを意味する。 $V L I W$ アーキテクチャにおいて、命令のグループが同時に実行される。これを行なうため、個々の命令は相互に依存しないことを確認することが必要である。この目的のため、命令の中に NOP を含めることが必要であるかも知れない。命令が圧縮される時に、これは通常 NOP を除去することによって行なわれる。命令が後の時点で実行されなければならない時に、同時に実行される命令が相互に依存しないという要件を満足させるため、NOP が再挿入されなければならない。

40

【 0 0 3 8 】

ヘッダ 1 は、更に、フェッチ制御ビット F を含む。フェッチ制御ビット F は、図 2 と関連づけて後で説明される。最後に、ヘッダ 1 は、多数の補助ビット $A_1 \dots A_x$ を含む。これらの補助ビットは、操作セグメント自身の中にフィットしない追加の操作セグメント情報をエンコードするために使用されることができる。

【 0 0 3 9 】

50

図2は、 VL^2IW 命令が、プログラム・メモリの中でどのように記憶されるかを示す。図2において、全部で9つの命令を記憶された6つのメモリ・ラインが示される。各々の命令はヘッダ1および多数の操作セグメント2を含む。1つのプログラム・メモリ・ラインは、連続的に記憶された多数の VL^2IW 命令(の一部)を含むことができる。1つのクロック・サイクルで、最悪のサイズ(即ち、 VL^2IW ヘッダおよび後続の非圧縮 VL^2IW 命令(例えば、アドレス0でスタートする命令)のビット・サイズ)の VL^2IW 命令をフェッチすることができる。

【0040】

順次のプリフェッチは、命令が、伸長、デコーディング、および実行のために時間どおりにフェッチされることを確実にする。即ち、1つの命令を実行している間、次の命令の解凍および実行を可能にする十分なビットがフェッチされることを確実にする。従って、最初の命令の実行が完了した時に、次の命令が解凍および実行の準備を完了する。これは処理時間を節約する。

10

【0041】

「分岐予測」とは、分岐命令の過去の記録に基づいて、分岐命令がジャンプするか否かをマイクロプロセッサが予測しようとする試みである。もし分岐命令が、例えば過去4回ジャンプしたのであれば、次回もジャンプする可能性が高い。パイプラインが使用されるプロセッサで、プロセッサが分岐命令に出会った時に、プロセッサは、前述した予測に従ってパイプラインの中へ次にロードする命令を決定することになる。即ち、プロセッサは、正しい命令がロードされたかどうかを確実に知らない。もし予測が誤りであったことが分かると、パイプラインは消去される必要があり、予測に基づいてなされた全ての計算は廃棄される必要がある。従って、多大の処理時間が失われてしまう。しかし、もし予測が正しかったことが判明すると、多大の時間が節約される。分岐予測は、前述したように統計的情報に基づいて実行されるので、予測は、誤りの可能性よりも正しい可能性が高い。即ち、一日の終わりに、処理時間は節約されている可能性が高い。たとえ、パイプラインが、時折、予測の誤りが判明したことに起因して、消去されなければならないとしてもそうである。

20

【0042】

分岐予測に代えて、他のアプローチが使用されてよい。深くないパイプラインを有するプロセッサは、分岐アドレスを時間どおり得るため、バイパス・ロジックを使用してよい。遅延分岐をサポートするプロセッサは、分岐シャドウ(即ち、分岐命令を含む命令がフェッチされるサイクルと、分岐目標がフェッチされる時間との間に経過するサイクル)を使用して、プログラムで分岐命令に続くが論理的には分岐が取られる前に実行されなければならないプログラム部分に属する命令を実行してよい。このアプローチが、好ましいアプローチである。

30

【0043】

分岐予測、又は前述した類似のアプローチなしに、分岐目標は、分岐が取られる(予測される)場合に、時間どおりにプリフェッチされることはできない。更に、もし分岐目標の VL^2IW 命令が2つのメモリ・ラインに伸びているならば、完全な分岐目標のフェッチは2サイクルを取り、1停止サイクルのパフォーマンス・ペナルティを生じる(プロセッサが完全な分岐目標のフェッチを待つ間に)。分岐目標を含む命令ワードが、コード・サイズを最適化するために圧縮される時に、分岐目標が2つのメモリ・ラインに伸びていないことを確実に知ることはできない。従って、もし分岐予測が使用されなかったならば、かなりのペナルティが生じることになる。停止を防止するため、もし分岐目標が現在のプログラム・メモリ・ラインに完全にフィットしなければ、次のプログラム・メモリ・ラインのスタートに分岐目標を置くことができる。従って、そのような分岐目標の再整列は、現在のラインでパディング・ビットを必要とし、パディング・ビットはコード・サイズを増加させる。

40

【0044】

本発明に従えば、フィットしない分岐目標における停止と、フィットしない分岐目標の再

50

整列との組み合わせが可能である。これは、コード・サイズに対してパフォーマンスをトレードオフすることを可能にする。もちろん、もしヘッダ・ビットを最小に保つ必要があれば、停止又は再整列のいずれかによる解決法も可能である。

【 0 0 4 5 】

本発明に従えば、3つのフェッチ制御ビットが定義される。これらのビットは、停止(S)ビット、再整列(R)ビット、およびプリフェッチ(P)ビットである。プリフェッチ・ビットは常に存在する。それは、フェッチされる次のVL² IW命令が現在のメモリ・ラインにフィットしないことを、プロセッサの命令フェッチ・ロジックへ知らせる。従って、次の命令の解凍と実行を可能にするため、次のメモリ・ラインがフェッチされなければならない。このビットの使用は、ハードウェアがプリフェッチの正しい瞬間を決定する必要性を除く。正しい瞬間は、例えば電力の消費を節減するため、不必要なフェッチが最小にされる瞬間である。もしプリフェッチ・ビットが使用されなければ、命令フェッチ・ハードウェアの臨界経路でコストのかかる追加の計算が必要になる。

10

【 0 0 4 6 】

プロセッサの要件に依存して、停止ビット、再整列ビット、又はこれらの双方が命令フォーマットで使用される。再整列ビットは、解凍されるべき次の命令が再整列されること、即ち、現在の命令に続くパディング・ビットがスキップされるべきことを、命令伸長ロジックへ知らせる。分岐が取られる時に、停止ビットは、分岐先の命令が、1サイクルの代わりに2サイクルでフェッチされなければならないこと、即ち停止サイクルが必要であることを、命令フェッチ・ロジックへ知らせる。

20

【 0 0 4 7 】

図2は、プログラム・メモリの中にVL² IWコードを記憶した例を示す。各々の命令ワードにおけるヘッダ1の最初の3ビットは、停止ビット、再整列ビット、およびプリフェッチ・ビットである。最初の命令(即ち、アドレス0でスタートする命令)の停止ビット、再整列ビット、およびプリフェッチ・ビットは、0、0、および1の値を有する。これは次のことを示す。

【 0 0 4 8 】

停止サイクルの必要性はないこと。

【 0 0 4 9 】

次の命令は再整列されないこと。即ち、現在の命令に続くパディング・ビットは存在しないこと。

30

【 0 0 5 0 】

次の命令の解凍および実行を可能にするため、次のメモリ・ラインをフェッチしなければならないこと。

【 0 0 5 1 】

パディング・ビットは最初の命令に続いていないこと、および、次の命令(即ち、アドレス8でスタートする命令)の全体が次のメモリ・ラインに置かれていること、従ってこのメモリ・ラインは、この命令の解凍および実行の前にフェッチされなければならないことが、図面から明らかである。アドレス8における前のメモリ・ワードに完全にフィットしない分岐目標が、アドレス16に置かれている。分岐目標は、アドレス16のワードのスタートへ再整列される。従って、アドレス8のワードの終わりに、パディング・ビットが必要である。先行する命令(即ち、アドレス13からスタートする命令)の再整列ビットおよびプリフェッチ・ビットは、1および1の値を有する。これは、パディング・ビットが現在の命令に続くこと、および次のメモリ・ライン(次の命令がこのメモリ・ラインへ再整列されたので、次の命令の全体を含む)が、次の命令の解凍および実行の前にフェッチされるべきことを示す。

40

【 0 0 5 2 】

アドレス39に、他の分岐目標が置かれる。この目標は、アドレス32からスタートするメモリ・ラインに完全にフィットしない。しかし、今度は、再整列は使用されない。その代わりに、分岐目標命令(即ち、アドレス39からスタートする命令)のヘッダの停止

50

ビットを高にする（即ち、値 1 を与える）ことによって、停止が強制される。更に、先行する命令、即ちアドレス 32 からスタートする命令のプリフェッチ・ビットは値 1 を有する。これは、次の命令の解凍および実行の前に、次のメモリ・ラインをフェッチしなければならないことを示す。これが正しいことは図面から明らかである。なぜなら、次の命令（アドレス 39 からスタートする）はメモリ・ラインにまたがるからである。従って、既に利用可能な命令部分は別にして（事実として、アドレス 32 からスタートするメモリ・ラインは、先行する命令を解凍および実行するため既にフェッチされているため）、命令の残りの部分を利用可能にすること、即ち次のメモリ・ラインをフェッチすることが必要である。

【 0 0 5 3 】

本発明は、デジタル信号プロセッサ（DSP）のコアのシーケンサの中で命令フェッチおよび解凍ユニットを実現する場合に使用される。ハードウェアは、フェッチ制御ビットを次のように解釈する。

【 0 0 5 4 】

次のような定義を仮定する。

【 0 0 5 5 】

S = 分岐から分岐目標へ到着した時に停止（S = 停止、s = 停止なし）

R = 次の命令の再整列（R = 再整列、r = 再整列なし）

P = 次のメモリ・ラインのプリフェッチ（P = プリフェッチ、p = プリフェッチなし）

命令に到達するためには、2つの方法がある。命令は、先行する命令から直接到達されるか、取られた分岐から到達されてよい。

【 0 0 5 6 】

命令がすぐ前の命令（取られた分岐ではない）から到達される時に、停止ビットは無視され（即ち、停止ビットの値は、フェッチ制御ロジックによるフェッチ制御ビットの解釈からは重要ではない）、他のフェッチ制御ビットは次のように解釈される。

【 0 0 5 7 】

s r p / S r p : 次の命令はフィットする。従って何もしない。

【 0 0 5 8 】

s r P / S r P : 次の命令はフィットしないが、再整列されない。従ってプリフェッチする。

【 0 0 5 9 】

s R P / S R P : 次の命令はフィットせず再整列される。従ってプリフェッチされ、次のメモリ・ラインへ再整列される。

【 0 0 6 0 】

最初の場合、次の命令は、既にフェッチされたメモリ・ラインにフィットする。従って、前記命令を解凍および実行するため、追加のメモリ・ラインをフェッチすることは必要でない。

【 0 0 6 1 】

2番目および3番目の場合、次の命令はフィットしない。従って、双方の場合に、前記命令の解凍および実行の前に次のメモリ・ラインをフェッチすることが必要である。2番目の場合、次の命令は再整列されないが、命令はフィットしないのであるから、メモリ・ラインをまたぐことが必要であり、従ってプリフェッチが必要になる。3番目の場合、次の命令は再整列される。従って現在のメモリ・ラインは、スキップされなければならないパディング・ビットを与えられる。

【 0 0 6 2 】

命令が、取られた分岐から到達される時に、フェッチ・ビットは次のように解釈される。

【 0 0 6 3 】

s r p : 分岐目標はフィットし、次の命令はフィットする。従って何もしない。s r P : 分岐目標はフィットし、次の命令はフィットしないが、再整列されない。従ってプリフェッチする。

10

20

30

40

50

【0064】

sRP： 分岐目標はフィットし、次の命令はフィットせず、再整列される。従ってプリフェッチし、次のメモリ・ラインへ再整列される。

【0065】

Srp： 分岐目標はフィットせず、次の命令はフィットする。従って分岐目標の2番目の部分を含む次のメモリ・ラインをフェッチするために停止する。

【0066】

SrP： 分岐目標はフィットせず、次の命令はフィットしないが再整列されない。従って停止して、分岐目標の2番目の部分を含む次のメモリ・ラインをフェッチし、次のメモリ・ラインの後に置かれたメモリ・ラインをプリフェッチする。

10

【0067】

SRP： 分岐目標はフィットせず、次の命令はフィットせず再整列される。従って停止して、分岐目標の2番目の部分を含む次のメモリ・ラインをフェッチし、次のメモリ・ラインの後に置かれたメモリ・ラインをプリフェッチし、次のメモリ・ラインの後に置かれたメモリ・ラインへ再整列する。

【0068】

命令がすぐ前の命令から到達される場合に関する注意は、ここでも等しく適用される。従って、停止ビットは、分岐目標がまたがるラインの数を示すこと、即ち、解凍が始まる前に分岐目標の全体をフェッチさせるため、停止サイクルが必要であるかどうかを示すことが明らかであろう。

20

【0069】

従って、本発明は、デンスコードが得られ、同時に処理パフォーマンスを比較的高レベルに維持するような方式で、極長命令ワード(VLIW)を作成、圧縮、および解釈する方法を提供する。なぜなら、本発明は、コード密度と処理パフォーマンスとの間のトレードオフを可能にするからである。

【0070】

更に、本発明は、デコーダのフィードバック・ループで通常必要となるコストの高い幾つかの計算を除去するVLIWを作成、圧縮、および解釈する方法を提供する。これは、続いて実行される命令ワードをどのようにフェッチするかの情報を、現在解凍および実行されている命令ワードのヘッダで利用可能にすることによって得られる。

30

【0071】

更に、本発明は、VLIWをデンスコードへ圧縮できるような方式で、また処理パフォーマンスが比較的高レベルに保たれるような方式で、VLIWを形成する命令ワードをフェッチ、解凍、および実行するコンピュータ・ロジック回路を有するコンピュータ・システムを提供する。これは前述したようにして得られる。

【0072】

図4は、命令のプログラムのために、例えば、VLIWプロセッサの命令をコンパイルするコンパイラ・プログラムによって実行される命令生成プロセスのフローチャートである。フローチャートは、命令内の操作が生成される最初のステップ41を含む。命令のヘッダの中で、P、S、およびRビットがクリアされる。2番目のステップ42において、命令がNOP操作を含むか否かが決定される。もし含まなければ、3番目のステップ43aが実行される。ステップ43aは、命令の長さをフル命令長へ設定し、発行スロットがNOPを受け取らないことを指示するように、伸長制御ビットを充填する。もし命令の中にNOPがあれば、代替のステップ43bが実行される。ステップ43bにおいて、非NOP操作がNOP操作の位置へ移動され、命令からNOPを効果的にカットする。伸長制御ビットは、NOPの位置を指示するように設定され、長さは、命令の中に残された非NOP操作の数に従って設定される。

40

【0073】

4番目のステップ44において、命令の長さが、前の命令の終わりの位置に加えられた時に、命令がメモリ・ラインの終わりを越えるような長さであるかどうか決定される。も

50

しそのような長さでなければ、5番目のステップが実行され、命令およびそのヘッダが、後の実行のためにコンピュータ読み取り可能メモリへ書き込まれる。この後に、他のハウスキッピング、例えば、命令のロケーションに関する情報をラベル相互参照テーブルへ書き込むことが続く。ラベル相互参照テーブルは、或る種の命令、例えば分岐命令に割り当てられるラベルに関連付けられた情報のためのエントリを有する。5番目のステップ45の後で、プロセスは最初のステップ41から反復される。

【0074】

もし4番目のステップ44が、前の命令に連続して記憶されたとした場合の命令がメモリ・ラインを越えると決定すれば、6番目のステップ46が実行される。6番目のステップ46は、命令が分岐目標であるかどうかを決定する（例えば、前のコンパイル段階で生成されるラベル相互参照テーブルの中で命令のエントリを検出することによって、又はコンパイルの間に命令を目標として識別するタグから）。もし命令が分岐目標でなければ、7番目のステップ47が実行される。ステップ47において、命令が2ライン以上にまたがることを示すため、前の命令のプリフェッチ・ビットが設定される。その後で、5番目のステップ45が実行される。

10

【0075】

もし命令が分岐目標であれば、命令を再整列すべきか否かを決定するため、8番目のステップ48が実行される。基本的には、比較的頻繁に実行される命令は再整列されることになる。これは、例えば、プロフィール情報に相談し、命令が何回実行されるかをカウントするプログラムのシミュレーション実行から決定することができる。もしプロフィールからのカウントが閾値を超過すれば、8番目のステップ48は命令の再整列を決定する。その場合、9番目のステップ49bが実行される。ステップ49bにおいて、前の命令の再整列ビット(R)が設定され、命令を記憶するための位置インディケータが、次のメモリ・ラインのスタートへ増分される。その後で、5番目のステップ45が実行され、位置インディケータによって指示された位置へ命令が記憶される。

20

【0076】

もし8番目のステップ48が命令の再整列を決定しなければ、代替の9番目のステップ49aが実行される。ステップ49aにおいて、現在の命令の停止ビットSおよび前の命令のプリフェッチ・ビットが設定される。位置インディケータは、前の命令の終わりへ不変のままに維持される。その後で、5番目のステップ45が実行される。

30

【0077】

このように、NOPを除去し、および/又はパディングなしで命令が2つのメモリ・ラインをまたぐようにすることによって、コード・サイズが縮小される。命令がまたがない時の実行スピードの利得を、増加するメモリ・サイズに対してバランスさせるように、分岐目標をまたがらせるかどうかの選択がなされる。停止ビットおよび再整列ビットは、比較的簡単なハードウェアによってこのサポートを確実にする。プリフェッチ・ビットを追加することによって、新しいメモリ・ラインが本当に必要なメモリ・サイクルへプリフェッチを限定することが可能になる。これはスラッシング（不必要なプリフェッチに起因するメモリ・サイクルの損失）を減少させる。

【図面の簡単な説明】

40

【図1】 可変長極長命令ワード(VL² IW)命令フォーマットを示す図である。

【図2】 コード・メモリ内に記憶されたVL² IWコードの例を示す図である。

【図3】 コンピュータ・システムを示す図である。

【図4】 コンパイル・プロセスのフローチャートである。

【符号の説明】

30 メモリ

32 命令発行ユニット

34 命令処理ユニット

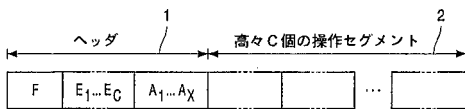
320 ヘッダ選択ユニット

322a ~ 322c 命令選択ユニット

50

- 3 2 4 選択制御ユニット
- 3 2 6 アドレッシング・ユニット
- 3 2 8 先行ライン・レジスタ
- 3 2 9 NOPソース

【図1】



【図4】

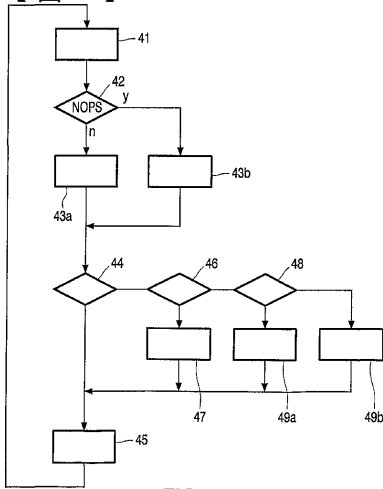
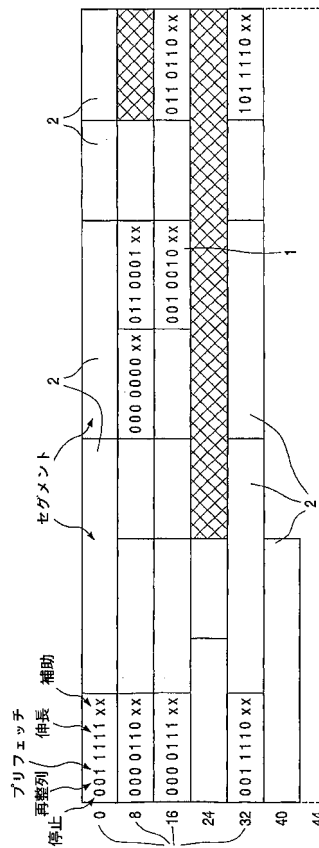


FIG. 4

【図2】



【 図 3 】

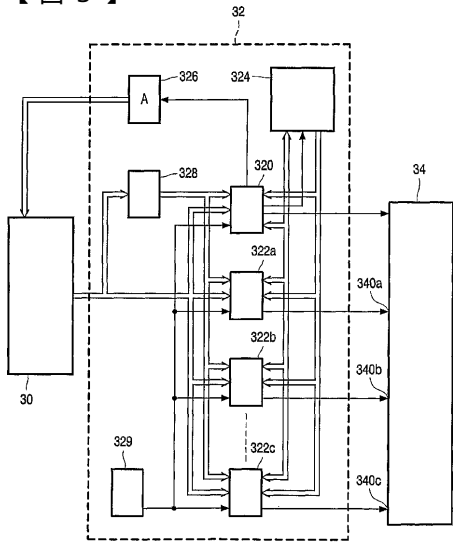


FIG. 3

フロントページの続き

(74)代理人 100082991

弁理士 佐藤 泰和

(74)代理人 100096921

弁理士 吉元 弘

(74)代理人 100103263

弁理士 川崎 康

(72)発明者 ジェルーン、エイ・ジェイ・レイイテン

オランダ国5656、ア-ア-、アインドーフエン、プロフ・ホルストラ-ン、6

審査官 後藤 彰

(56)参考文献 特開平9-288578(JP,A)

特表平9-505920(JP,A)

特表平11-509664(JP,A)

特表2001-515619(JP,A)

特開昭63-146143(JP,A)

(58)調査した分野(Int.Cl.,DB名)

G06F 9/30 - 9/38