

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3958224号
(P3958224)

(45) 発行日 平成19年8月15日(2007.8.15)

(24) 登録日 平成19年5月18日(2007.5.18)

(51) Int. Cl.

F I

G 0 6 F 9/308 (2006.01)

G 0 6 F 9/30 3 4 O B

G 0 6 F 9/315 (2006.01)

G 0 6 F 9/30 3 4 O D

請求項の数 5 (全 13 頁)

(21) 出願番号 特願2003-21481 (P2003-21481)
 (22) 出願日 平成15年1月30日(2003.1.30)
 (65) 公開番号 特開2003-296103 (P2003-296103A)
 (43) 公開日 平成15年10月17日(2003.10.17)
 審査請求日 平成17年11月21日(2005.11.21)
 (31) 優先権主張番号 10/112783
 (32) 優先日 平成14年3月30日(2002.3.30)
 (33) 優先権主張国 米国(US)

(73) 特許権者 398038580
 ヒューレット・パッカード・カンパニー
 HEWLETT-PACKARD COMPANY
 アメリカ合衆国カリフォルニア州パロアル
 ト ハノーバー・ストリート 3000
 (74) 代理人 110000039
 特許業務法人アイ・ピー・エス
 (72) 発明者 デール・モリス
 アメリカ合衆国コロラド州スティームボ
 ートスプリングス アガテクリークロード3
 5920

審査官 石川 正二

最終頁に続く

(54) 【発明の名称】 結果が分散される並列サブワード命令

(57) 【特許請求の範囲】

【請求項1】

マイクロプロセッサ(110)であって、
 デコードするための命令デコーダ(DEC)と、
 少なくとも第1のサブワード並列命令および第2のサブワード並列命令を実行するための
 実行ユニットであって、前記第1のサブワード並列命令は、結果レジスタが有するサブ
 ワードロケーション内に、前記第1のサブワード並列命令の結果を格納するための第1の
 サブワードビット位置を指定し、前記第2のサブワード並列命令は、前記サブワードロケ
 ーション内に、前記第2のサブワード並列命令の結果を格納するための第2のサブワード
 ビット位置を指定し、前記第1のサブワードビット位置および前記第2のサブワードビッ
 ト位置は異なる実行ユニット(EXU)と
 を備える
 マイクロプロセッサ。

【請求項2】

前記サブワード並列命令それぞれは、前記サブワード並列命令の結果それぞれを格納す
 るために使用されない結果レジスタそれぞれのサブワードビット位置を上書きする
 請求項1記載のマイクロプロセッサ。

【請求項3】

前記サブワード並列命令それぞれは、前記サブワード並列命令の結果それぞれを格納す
 るために使用されない結果レジスタそれぞれのサブワードビット位置の内容を保存する

10

20

請求項 1 記載のマイクロプロセッサ。

【請求項 4】

コンピュータによって実施される方法であって、

結果レジスタが有する各サブワードロケーション内の対応するサブワードビット位置に、第 1 のサブワード並列命令の結果を格納する、前記第 1 のサブワード並列命令を実行するステップと、

前記各サブワードロケーション内の対応するサブワードビット位置に、第 2 のサブワード並列命令の結果を格納する、前記第 2 のサブワード並列命令を実行するステップとを有し、

前記第 2 のサブワード並列命令は、前記第 2 のサブワード並列命令の結果を、前記第 1 の並列サブワード命令が前記第 1 の並列サブワード命令の結果を格納するサブワードビット位置と同じサブワードビット位置に格納し、前記第 1 のサブワード並列命令の結果を、前記サブワードロケーション内の新しいサブワードビット位置にシフトさせる方法。

10

【請求項 5】

コンピュータによって実施される方法であって、

結果レジスタが有する各サブワードロケーション内の対応するサブワードビット位置に、第 1 のサブワード並列命令の結果を格納する、前記第 1 のサブワード並列命令を実行するステップと、

前記各サブワードロケーション内の対応するサブワードビット位置に、第 2 のサブワード並列命令の結果を格納する、前記第 2 のサブワード並列命令を実行するステップとを有し、

20

前記第 2 のサブワード並列命令は、前記第 2 のサブワード並列命令の結果を、第 1 のサブワード並列命令が前記第 1 のサブワード並列命令の結果を格納するサブワードビット位置とは異なるサブワードビット位置に格納する

方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はデータ処理に関し、特に、データプロセッサが実行可能な命令に関する。本発明の主な目的は、マルチメディアアプリケーションにおいて一般的な特定の状況で改良されたプロセッサパフォーマンスを達成する並列サブワード比較命令を提供することにある。

30

【0002】

【従来の技術】

近代の進歩の大部分は、コンピュータ技術の進展に関連する。通常のコンピュータは、プログラム命令に従ってデータに対して演算を行う 1 つまたは複数のプロセッサを備える。各マイクロプロセッサには、命令セット、すなわちマイクロプロセッサが実行可能な命令のセットが関連付けられる。

特定のアプリケーションのコンテキストにおけるプロセッサのパフォーマンスは、プロセッサの命令セットを使用して、アプリケーションが要求するタスクをどの程度効率的に特徴付けることができるかに大きく依存する。

40

【0003】

コンピュータは、高性能になればなるほど、リアルタイムでのビデオ操作等さらに多くを要求するアプリケーションに直面してきた。

通常、ビデオは画像の同期ストリームとして提示される。

各画像は、二次元アレイになった画素（ピクセル）として説明することができる。

通常、各ピクセルは色次元ごとに 1 つのマルチビット（たとえば 8 ビット）色値で記述される。多くのアプリケーション、たとえば映像圧縮は、ピクセルをブロック（たとえば、8 × 8 ピクセルブロック）にグループ化する。

【0004】

50

一般的な二値画像演算（同じ画像または異なる画像のブロック同士の比較等）では、オペランドデータの $2 \times 64 \times 8$ （2 ブロック、64 ピクセル/ブロック、8 ビット/ピクセル）ビットを同時に処理する必要がある。

ますます普及しつつある 64 ビットプロセッサを使用する場合、二値画像ブロックの比較は、オペランドに 8 対の 64 ビットレジスタを使用して実施することができ、追加のレジスタが中間結果および最終結果の格納に使用される。

【0005】

全部で 64 の比較が必要である。

各比較の結果は 1 ビットの情報である。

総計で、64 の比較の結果は、単一の結果レジスタに格納することのできる単一の 64 ビットワードになる。 10

これら 64 の比較をどの程度効率的に行い、かつ組み合わせることができるかは、マイクロプロセッサに関連する命令セットに依存する。

ワード比較命令が使用される場合、8 つのデータビットを使用し、固定値が書き込まれたオペランドを生成しなければならない。

次いで、ワードが比較される。次に、1 ビット値が結果レジスタの適当なビット位置に格納される。

連続した結果は、前の結果を消去することなく結果レジスタの適当な位置に格納しなければならない。

したがって、ワード比較命令を使用して画像ブロックの比較を完了するには、64 の比較命令および他の多くの命令が必要である。 20

【0006】

プロセッサによっては、並列サブワード比較命令に対応するものもある。一对の 64 ビットレジスタに適用されるように、レジスタに格納された 8 対の 8 ビットサブワードを並列に比較して、たとえば、結果レジスタの 8 つの最下位ビット位置に格納することのできる 8 ビット結果を生成することができる。この結果をより上位のビット位置にシフトして、他のレジスタ対での並列サブワード比較演算の結果のために場所を空けることができる。画像ブロックの比較を完了するために、8 つの並列サブワード命令、7 つのシフト命令、および 7 つの OR 命令（結果を組み合わせるため）が必要であり、総計で 22 の命令が必要である。 30

【0007】

命令の総数は、並列サブワード比較命令の特定の実施によって決まる。たとえば、Intel Itanium プロセッサで使用される代替の並列サブワード比較命令は、結果レジスタの対応するサブワードロケーションに各サブワード比較結果の 8 つのレプリカを格納することができる。

この場合、8 つの並列サブワード比較命令の結果を単一の結果にするために、より多くの補助命令が必要である。したがって、画像ブロックの比較に必要な命令の総数は、最初の例に提示された総数を 2 倍したものよりも多い場合がある。

【0008】

Ruby Lee による関連特許出願（代理人整理番号 10971660-1）は、結果を結果レジスタの異なるサブワードロケーションに格納させて、結果を組み合わせる前にシフトする必要がないようにする並列サブワード比較命令を開示する。 40

ターゲティングアプローチでは、命令が結果のサブワードロケーションを指定することができ、シフトアプローチでは、結果レジスタの前の内容を新しいサブワードロケーションにシフトして、前の結果を失うことなく元のロケーションに現在の演算結果を格納することができる。レプリカアプローチでは、結果の完全なレプリカが複数のサブワードロケーションに格納される。

OR 命令または MIX 命令を使用して、別個の「シフト命令」なしで複数の並列サブワード命令からの結果を組み合わせることができる。

これにより、一連の並列比較命令を組み合わせるために必要な命令の数が $1/3 \sim 2/3$ 50

節約される。

【 0 0 0 9 】

上記アプローチはコンピュータパフォーマンスを実質的に向上させるが、さらなる向上が望まれる。

【 0 0 1 0 】

【 発明が解決しようとする課題 】

本発明は、結果を結果レジスタに不連続に分散させる並列サブワード命令を提供する。好ましくは、結果は、オペランドレジスタと結果レジスタとの間の情報の横方向移動を最小化するように結果レジスタのビット位置の間で分散される。

いずれにしても、分散は、「最大横方向移動」が、結果が連続して格納される場合よりも小さくなるようなものであるべきである。

本明細書では、結果の格納に使用されたビット位置における少なくとも1つのギャップがある場合に結果が格納され、好ましい実施形態では、2つの結果ビット位置が連続することはない。

【 0 0 1 1 】

【 課題を解決するための手段 】

最大横方向移動を最小化するために、サブワード演算の結果を、結果レジスタの対応するサブワードロケーション内に格納することができる。

たとえば、複数の並列サブワード演算それぞれの結果を、結果レジスタの対応するサブワードロケーションの最下位ビット位置に格納することができる。

これにより、最大横方向移動が1サブワード未満になることが保証されるのに対し、上記関連技術では、少なくとも1サブワード分、最大でデータパス全幅と同じだけの横方向移動が必要である。

【 0 0 1 2 】

本発明は、「シフト」アプローチおよび「ターゲッティング」アプローチを提供する。

シフトアプローチでは、レジスタの内容が、現在の演算の結果を格納するとき、前の演算結果が保存されるようにシフトされる。

たとえば、並列サブワード比較演算の結果として、結果が結果レジスタのサブワードロケーションの最下位ビット位置に格納され、その命令の次の繰り返しにより、前の結果が結果レジスタサブワードロケーションの次の最下位ビット位置に前の結果がシフトされ、現在の並列サブワード比較演算の結果を結果レジスタサブワードロケーションの最下位ビット位置に格納するものと仮定する。

このシフトアプローチの利点は、命令が繰り返されるとき結果が自動的に組み合わせられ、結果を組み合わせるためのさらなる（たとえばOR）命令の必要性がなくなることである。

【 0 0 1 3 】

「ターゲッティング」アプローチでは、複数の命令それぞれが、結果を格納すべき不連続のビット位置のセットを指定（暗黙的もしくは明示的に）する。

たとえば、ある命令が結果レジスタサブワードロケーションの最下位ビットロケーションを指定し、別の命令がサブワードロケーションの次の最下位ビットロケーションを指定し、以下同様である。

この例では、各結果レジスタサブワードビット位置専用の1つの命令があり得る。

結果を異なるビットロケーションセットに格納することによって、複数の並列サブワード命令の結果をシフトなしで組み合わせることができる。

【 0 0 1 4 】

本発明は、ターゲッティングアプローチの「上書き」および「内容保存」変形を提供する。

内容保存変形では、結果の格納に使用されない結果レジスタビット位置には手を付けずに残しておく。

これにより、集合的な結果がさらなる命令なしで組み合わせられるように、連続した並列

サブワード比較演算（異なるビット位置セットを指定する）を同じ結果レジスタに書き込むことができる。

このアプローチは、シフトアプローチと同等に、結果の組み合わせの達成に必要な命令の数を最小化する。

この場合、命令は結果のシフトを要求せず、他方でシフトアプローチは、ロケーションの指定を要求しない。

【 0 0 1 5 】

上書き変形では、結果の格納に使用されないビット位置が、たとえばゼロで上書きされる。

この変形では、連続した並列サブワード演算の結果を、異なるレジスタの異なる位置に書き込むことができ、シフトすることなく結果を組み合わせる（たとえば、論理和をとることにより）ことができる。

結果を組み合わせるために別個の命令が必要であるため、ターゲッティングアプローチの上書き変形では、ターゲッティングアプローチの内容保存変形ならびにシフトアプローチよりも多くの命令が必要である。

【 0 0 1 6 】

タスクを完了するために必要な命令の数は、常にパフォーマンスの最もよい指標であるわけではない。いくつかの命令を並列に実行することができる場合、

「依存性深さ（dependency-depth）」のほうがよい尺度であり得る。

シフトアプローチでは、依存性深さは命令の数に等しく、ターゲッティングアプローチでは、並列サブワード結果が、異なるレジスタに書き込まれる限り、依存性深さははるかに少ないはずである。

【 0 0 1 7 】

本発明によってもたらされる最大横方向移動の劇的な低減は、命令実行に必要な時間の大幅な低減に対応する。

この低減を用いて、命令サイクルに割り当てる時間を短縮し、また命令セットの結果に応じて、内蔵したマイクロプロセッサのルーティング要件を低減することができる。

また、通常、異なる命令からの情報は、重ならないビット位置セットに格納されるが、本発明は、たとえば暗号化目的のために、重なったビット位置への情報の格納を提供する。

上記に加えて、または上記に代わる他の利点および特徴は、添付図面を参照して以下の説明から明白になるう。

【 0 0 1 8 】

【発明の実施の形態】

本発明によれば、コンピュータプログラム 100 は、分散結果、上書き、ターゲッティング並列サブワード比較命令 P C o m p D O T 0 ~ P C o m p D O T 7 の命令サブセットから選択される命令を含む。

これらの命令は、以下のようにオペランド解析する (parse) ことができる。

すなわち、「並列サブワード比較を行い（8ビットサブワードを使用して）、各並列サブワード比較の結果を結果レジスタの対応するサブワードの（ $n + 1$ ）番目の（ $n = 0 \sim 7$ ）最下位ビット位置に格納し、結果データの格納に使用されていない結果レジスタビット位置にゼロを上書きする。」現在の比較演算の場合、対応するサブワード内容が等しい場合には「1」が格納され、他の場合には「0」が格納される。

しかし、他のサブワード比較命令は、他の関係、たとえば「以上」等に基づいて「1」および「0」結果を割り当てることができる。

【 0 0 1 9 】

連続した P C o m p D O T 0 および P C o m p D O T 1 を含むプログラム 100 のセグメントの演算を図 1 に示す。（完全なレジスタでの値は 16 進数であるが、サブワード細部における値は二進数である）。

P C o m p D O T 0 の結果は結果レジスタ R C 0 のサブワード（バイト）の最下位ビットの間で分散されるが、P C o m p D O T 1 の結果は結果レジスタ R C 1 のサブワードの 2

10

20

30

40

50

番目の最下位ビットの間で分散されることに留意する。

結果の格納に使用されないビットは、図 1 には“ Z ”で示すように、デフォルトのゼロが書き込まれる。

8 つの命令すべてならびに結果を組み合わせるための O R 命令を含むより全体的なプログラムセグメントを図 2 に示し、図 3 に示すデータを使用する。

【 0 0 2 0 】

これら命令は、図 4 に示すように、マイクロプロセッサ 1 1 0 およびメモリ 1 1 2 を備えたコンピュータシステム A P 1 のコンテンツで実行される。

メモリ 1 1 2 の内容は、プログラムデータ 1 1 4 およびプログラム 1 0 0 を構成する命令を含む。

マイクロプロセッサ 1 1 0 は、実行ユニット E X U、命令デコーダ D E C、レジスタ R G S、アドレスジェネレータ A D G、およびルータ R T E を備える。

【 0 0 2 1 】

概して、実行ユニット E X U が、プログラム 1 0 0 に従ってデータ 1 1 4 に対して演算を行う。

このために、実行ユニット E X U は、アドレスバス A D R に沿ってアドレスジェネレータ A D G に要求される次の命令またはデータのアドレスを生成するように命令することができる（内部データバス D T B に付随する制御ラインを使用して）。

メモリ 1 1 2 は、これに回答して、要求されたアドレスに格納されている内容をデータおよび命令バス D I B に沿って供給する。

【 0 0 2 2 】

内部データバス D T B に付随するインジケータラインに沿って実行ユニット E X U から受信されるインジケータによって決定されるように、ルータ R T E が、命令バス I N B を介して命令を、内部データバス D T B に沿ってデータを命令デコーダ D E C にルーティングする。

デコードされた命令は、制御ライン C C D を介して実行ユニット E X U に供給される。

通常、データは命令に従ってレジスタ R G S に / から転送される。

【 0 0 2 3 】

マイクロプロセッサ 1 1 0 には、命令デコーダ D E C がデコードし、実行ユニット E X U が実行することのできる命令 I N S の命令セットが関連付けられる。

プログラム 1 0 0 は、命令セット I N S から選択される順序付き命令セットである。

解説目的のために、マイクロプロセッサ 1 1 0、その命令セット I N S、およびプログラム 1 0 0 は、後述するすべての命令の例を提供する。

本発明はまた、本明細書に述べる新規命令のすべてではなくいくつかを使用する実施形態も提供する。

【 0 0 2 4 】

図 2 に示すプログラムセグメントは、8 つすべての命令 P C o m p D O T 0 ~ P C o m p D O T 7 および 7 つの O R 命令、合わせて 1 5 の命令を含む。

命令は、図 2 では 4 列に配置され、各列が、合計 4 つの依存性深さのうちの 1 つのレベルに対応する。

P C o m p D O T 0 ~ P C o m p D O T 7 命令の結果（図 3 に示すデータに適用される）は、レジスタ R C 0 ~ R C 7 に格納されて示される。

4 つの O R 命令の最初のセットの結果は、レジスタ R D 0 ~ R D 3 に示され、2 つの O R 命令の第 2 のセットの結果はレジスタ R E 0 および R E 1 に示され、最後の O R 命令の結果は結果レジスタ R F 0 に示される。

【 0 0 2 5 】

本願を通して提示し、使用するデータは、本発明の命令の説明を容易にするように選択されたものであり、このデータは図 3 に示される。

オペランドレジスタ R A 0 ~ R A 7 および R B 0 ~ R B 7 中のデータは 1 6 進数フォーマットであるが、結果は 2 進数フォーマットで示される。

10

20

30

40

50

結果のビット値は離間されて、結果レジスタ中で分散されることを示す。

【0026】

本発明によって提供される最大横方向移動の低減を図5に示す。

通常のアンドサブワードについて、本発明の分散アプローチに必要な最大横方向移動を実線の矢印で示し、関連技術の連続アプローチに必要な最大横方向移動を点線の矢印で示す。

本発明では、情報をサブワード内に残すことができるため、最大横方向移動は1サブワード未満である。

結果を連続して格納すると、最大横方向移動は1サブワードを超えることになる。

たとえば、結果が結果レジスタの最下位サブワードに連続して格納される場合、最上位オペランドビットに関連する横方向移動は、レジスタ全体から1サブワードを差し引いたものである。

したがって、本発明は、情報の最大横方向移動を大幅に低減し、レジスタ間のルーティングをより単純にし、データの移動をより高速にし、したがってプロセッサパフォーマンスをより高くすることができる。

【0027】

プログラム100の別のセグメントは、分散結果、内容保存、ターゲッティング並列サブワード比較命令PCompDP0~PCompDP7を使用する。

これら命令は以下のようにオペランド解析することができる。

すなわち、「並列サブワード比較を行い(8ビットサブワードを使用して)、各並列サブワード比較の結果を結果レジスタの対応するサブワードの(n+1)番目の(n=0~7)最下位ビット位置に格納し、結果データの格納に使用されていない結果レジスタビット位置を変更せずにそのままにする。」PCompDP0およびPCompDP1の連続した対の演算を図7に示す。

PCompDP0の結果は、結果レジスタRC0のサブワードの最下位ビットに格納される(PCompDOT0の場合と同様に)。

しかし、結果の格納に使用されないビット位置は、図6にZの代わりにXを使用して示すように、レジスタRC0に予め存在するデータで埋められる。

【0028】

図1の上書き命令と図6の内容保存命令との違いは、PCompDPT1の結果を参照するとより明確になる。

この命令の結果は、結果レジスタサブワードロケーションの2番目の最下位ビット位置に書き込まれる。

この場合、結果は、PCompDPT0の結果と同じ結果レジスタRC0に書き込まれるため、結果はさらなる命令なしで組み合わせられる。

したがって、望ましい組み合わせられた結果は、図7に示すように、8つの命令の後でのみ結果レジスタRC0に見られる(図4のデータを使用して)。

しかし、依存性深さ、したがってマイクロプロセッサ110によってもたらされる並列性を利用する機会はより少ない(However, the dependency depth is, so there is less opportunity to take advantage of parallelism provided by microprocessor 110)。

【0029】

並列サブワード比較命令のターゲッティング変形では、結果ロケーションを暗黙的あるいは明示的に指定しなければならない。

これは、各サブワードロケーションごとに異なる命令を使用することによって暗黙的に指定することができる。

この目的のために命令中に確保されたフィールドに入力された値によって明示的に指定することができる。

代替として、命令が、ロケーションを示す値が格納されるロケーションを明示的または暗黙的にポイントすることができる。

たとえば、命令は、結果を格納すべきサブワードロケーションを示す値を格納するレジス

10

20

30

40

50

タをポイントすることができる。

別の代替では、結果を格納すべきレジスタにサブワードロケーションを結び付ける。

たとえば、レジスタ R C 0 に格納される結果を最下位サブワード位置に格納し、レジスタ R C 1 に格納された結果を 2 番目の最下位サブワード位置に格納することができる。

【 0 0 3 0 】

分散した、内容保存、シフト並列サブワード比較命令 P C o m p D P S を使用するプログラムセグメントを図 8 および図 9 に示す。

この命令は、以下のようにオペランド解析することができる。

すなわち、「並列サブワード比較を行い(8 ビットサブワードを使用して)、各並列サブワード比較の結果を結果レジスタの対応するサブワードの最下位ビット位置に格納し、予め存在するデータを 1 ビット位置分左にシフトし、各サブワードの予め存在する最上位ビットをドロップする。」この場合、結果の宛先は 1 つしかないため、命令も 1 つしかない(2 つのターゲットティング命令サブセットそれぞれに 8 つの命令があることとは対照的に)。

【 0 0 3 1 】

図 8 に示すように、P C o m p D P S の結果は、図 6 の P C o m p D P T 0 の結果と同一に見える。

しかし、予め存在するデータを所定位置にしたままにするのではなく、P C o m p D P S は予め存在するデータを左に 2 ビットシフトする。

各サブワードごとの予め存在する最上位データビットはドロップされ、空いたサブワードの最下位ビット位置が命令結果で埋められる。

【 0 0 3 2 】

このシフトは、図 8 における第 2 の P C o m p D P S と合わせてより明確に例示する。

第 1 の P C o m p D P S の結果は左に 1 ビットシフトされ、第 2 の P C o m p D P S の結果はここでも、サブワードの最下位ビット位置に格納されることに留意する。

5 番目の最下位サブワード(図 8 において強調される) の 2 つの最下位ビット(これもまた図 8 において強調される) の検査は、P C o m p D P T 1 の対応する結果(図 6 において強調される) と異なる。

明らかに、情報は同じであるが、P C o m p D P T n と P C o m p D P S に関連する順序付け規則が異なるため、別様に表現される。

内容はシフトされて保存されるため、連続した P C o m p D P S 命令が同じレジスタに書き込むことができ、よって 8 つの命令の後に、所望の組み合わせられた結果が得られる。内容保存ターゲットティング命令の場合と同様に、プログラムセグメント長および依存性深さは双方とも 8 である。

【 0 0 3 3 】

2 つの連続した比較(または他の特徴付け) 命令は同じタイプのものと意図するが、本発明では同じタイプのものである必要はない。

たとえば、シフトタイプ命令はターゲットティング命令の後であってもよく、またその逆であってもよい。

さらに、2 つの命令が必要とする演算は同じである必要はなく、たとえば、一方が二値比較であり、他方が「ゼロに等しい(equals zero)」命令であってもよい。

【 0 0 3 4 】

したがって、本発明は、R u b y L e e の出願(代理人整理番号 1 0 9 7 1 5 5 0 - 1) に開示されるもののよう、並列サブワード比較命令に必要な命令の数を大幅に低減することができる異なるタイプの命令に対応する(provide for)。

例示した実施形態では、レジスタは 6 4 ビットであり、サブワードは 8 ビットであり、よってレジスタ当たり 8 つの互いに重ならないサブワードロケーションがある。

このレジスタおよびサブワードサイズの選択により、結果は 1 サブワード長である。

より長い、たとえば 1 6 ビットのサブワードが使用される場合、6 4 ビットレジスタ当たり 4 つのみの 1 6 ビットサブワード、ひいては結果当たり 4 ビットしかない。

10

20

30

40

50

したがって、16の互いに重ならない4ビット結果ロケーションがレジスタに存在しうる。

命令セットは、並列サブワード比較演算にこれらのうちいずれかを選択できるように設計することができる。

よって、このような命令の1つのフォーマットは、PComp(第1のオペランドレジスタ、第2のオペランドレジスタ、結果レジスタ、サブワードサイズ、結果レジスタにおける結果のターゲットロケーション)であることができる。

【0035】

上記説明では、異なる演算の結果を、同じあるいは異なる結果レジスタの異なるビット位置に格納した。

10

しかし、暗号化アプリケーションでは、結果格納に使用するビット位置セットがいくらか重なっている必要があり得る。

たとえば、ターゲットビット位置のセットが重なってもよく、別の例では、シフト命令が以前の内容のすべてではなくいくらかをシフトすることができる。

結果が同じレジスタに書き込まれる場合、前の結果のいくらかに上書きされ、結果が異なるレジスタに書き込まれる場合、たとえばXOR命令を使用して結果を組み合わせる場合、いくらかの情報をマスクすることができる。

失われた情報は、意図した受信者しか利用することができない暗号化鍵に従って別個に供給することができる。

【0036】

20

本発明は、比較演算のみならず、単項演算、2進演算、およびより高次の演算を含む他の演算にも適用される。

レジスタ長は変更可能であり、2のべき乗である必要はない。

サブワード長も変更可能であり、2のべき乗である必要はない。

演算結果によって決定されない結果レジスタのビットは、変更しないままにしてもよく、またゼロあるいは1のいずれかに設定してもよい。

例示した実施形態に対するこれらおよび他の変更および変形は本発明によって提供され、本発明の範囲は併記の特許請求の範囲によってのみ限定される。

【0037】

【発明の効果】

30

以上説明したように、本発明に係る並列サブワード命令によれば、結果を結果レジスタに不連続に分散させることができる。

また、本発明に係る並列サブワード命令によれば、結果は、オペランドレジスタと結果レジスタとの間の情報の横方向移動を最小化するように結果レジスタのビット位置の間で分散される。

また、本発明に係る並列サブワード命令によれば、分散を、「最大横方向移動」が、結果が連続して格納される場合よりも小さくすることができる。

また、本発明に係る並列サブワード命令によれば、結果の格納に使用されたビット位置における少なくとも1つのギャップがある場合に結果が格納され、2つの結果ビット位置が連続することはない。

40

【図面の簡単な説明】

【図1】本発明による一対の分散した上書きターゲットティング命令を含む第1のコンピュータプログラムセグメントの一部の概略図である。

【図2】本発明によるサブワード比較結果の組み合わせに使用する8つの並列サブワード比較命令およびOR命令を示す図1のプログラムセグメントのより全体的な図である。

【図3】本発明の演算を説明するために使用する例示的なオペランドデータおよび結果を提示する。

【図4】図1のコンピュータプログラム命令を実行するシステムの概略図である。

【図5】本発明によって提供される最大横方向移動と、ある関連技術の並列サブワード比較命令に必要な最大横方向移動とを比較する。

50

【図 6】分散内容保存ターゲット命令の連続した対の演算を示す第 2 のコンピュータプログラムセグメント対の概略図である。

【図 7】本発明による 8 つの連続した分散内容保存ターゲット命令を包含する図 6 のセグメントのより全体的な図である。

【図 8】本発明による一対の分散シフト命令を含む第 3 のコンピュータプログラムセグメント対の概略図である。

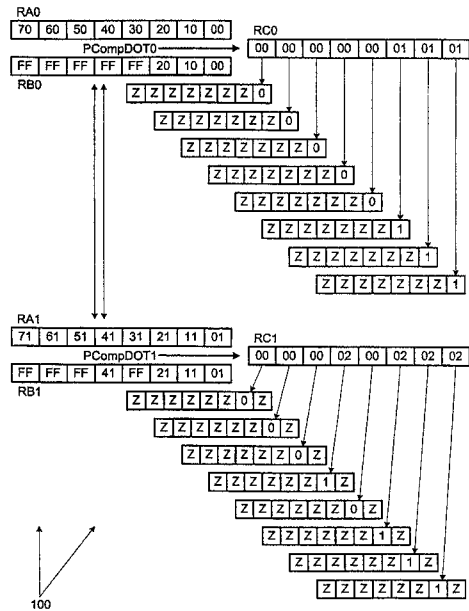
【図 9】本発明による 8 つの連続した分散シフト命令を示す第 3 のコンピュータプログラムセグメントのより全体的な図である。

【符号の説明】

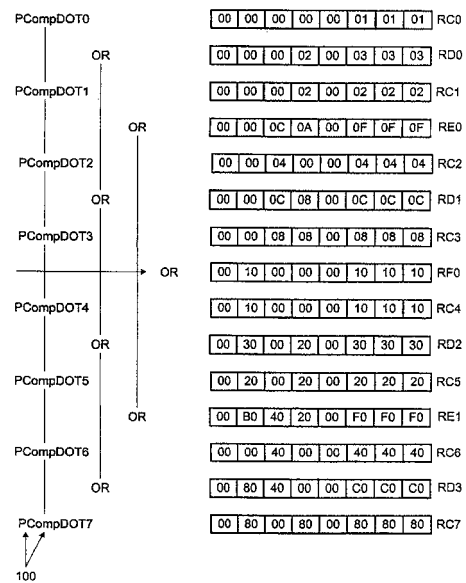
1 0 0 . . . プログラム命令、 1 1 0 . . . マイクロプロセッサ、 1 1 2 . . . メモリ、
1 1 4 . . . プログラムデータ

10

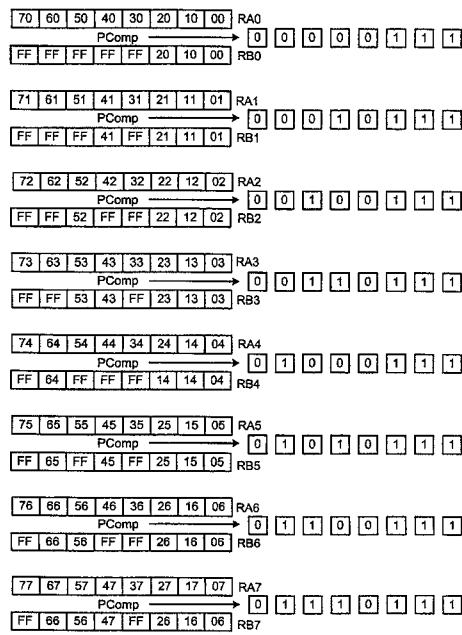
【図 1】



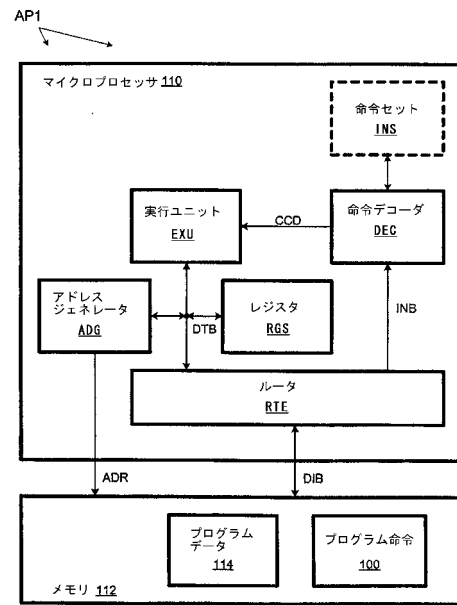
【図 2】



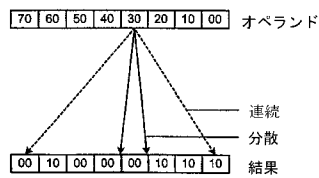
【図 3】



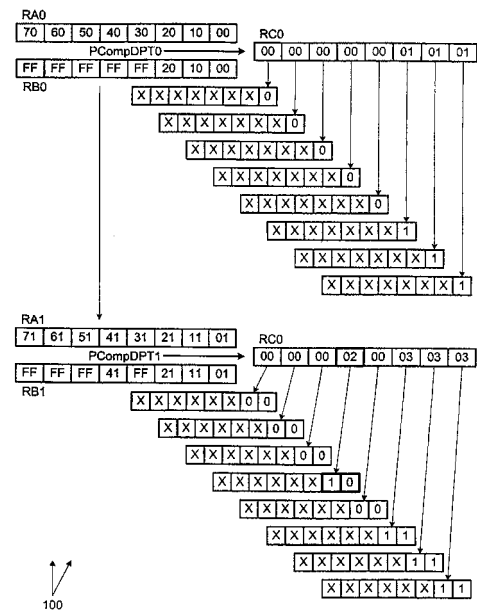
【図 4】



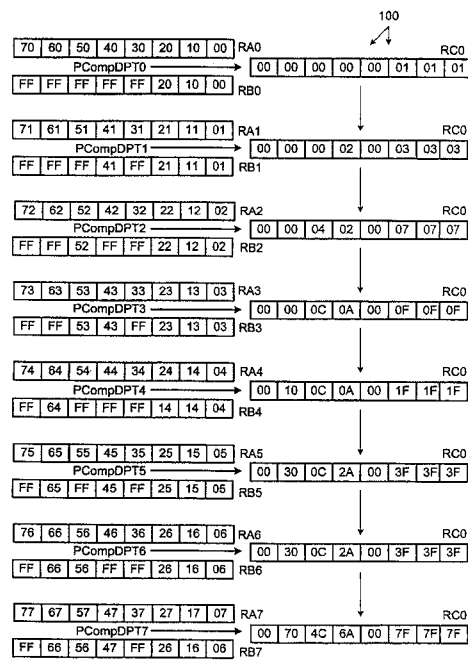
【図 5】



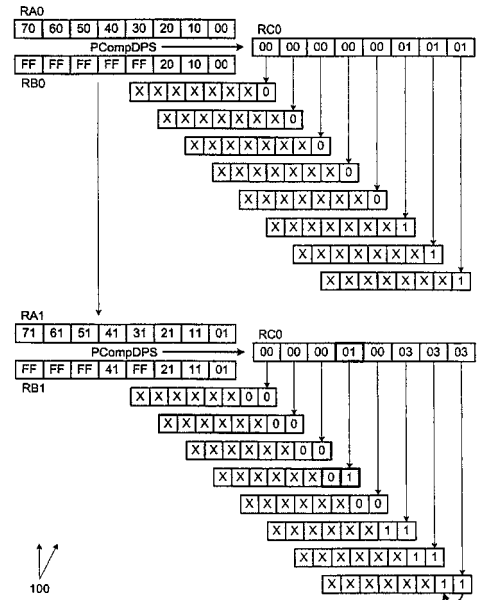
【図 6】



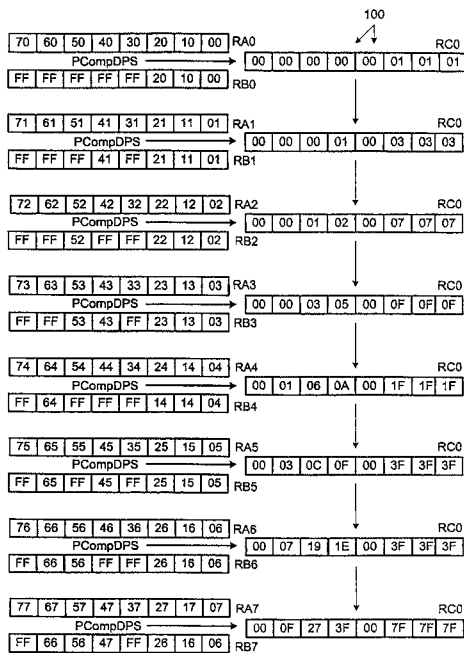
【図 7】



【図 8】



【図 9】



フロントページの続き

(56)参考文献 国際公開第00/022511(WO,A1)
特表平11-511575(JP,A)

(58)調査した分野(Int.Cl.,DB名)

G06F 9/308

G06F 9/315