(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3735105号
(P3735105)
(45)発行日　平成18年1月18日(2006.1.18)　(24)登録日　平成17年10月28日(2005.10.28)

(51)Int.Cl.　F I
*G06F 9/315 (2006.01)*　G06F 9/30 340D

請求項の数 9 (全 12 頁)

(21)出願番号　特願2004-43849(P2004-43849)
(22)出願日　平成16年2月20日(2004.2.20)
(65)公開番号　特開2004-303203(P2004-303203A)
(43)公開日　平成16年10月28日(2004.10.28)
審査請求日　平成16年4月15日(2004.4.15)
(31)優先権主張番号　10/403785
(32)優先日　平成15年3月31日(2003.3.31)
(33)優先権主張国　米国(US)

(73)特許権者　503003854
ヒューレット－パッカード　デベロップメント　カンパニー　エル．ピー．
アメリカ合衆国　テキサス州　７７０７０　ヒューストン　２０５５５　ステイト　ハイウェイ　２４９
(74)代理人　110000039
特許業務法人アイ・ピー・エス
(72)発明者　ルビー・ビー・リー
アメリカ合衆国ニュージャージー州プリンストン　エッティサークル５５
(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３５９２０

最終頁に続く

(54)【発明の名称】レジスタから並列テーブルルックアップを行う可変再配列（ＭＵＸ）命令

(57)【特許請求の範囲】
【請求項１】
データプロセッサ（１１０）であって、
第１のレジスタ（Ｒ１１）、第２のレジスタ（Ｒ１０）、および第３のレジスタ（Ｒ１２）と、
前記第２のレジスタの内容の少なくとも一部を再配列して前記第３のレジスタに書き込む第１の可変再配列命令を実行する実行ユニット（ＥＸＵ）であって、前記第１の可変再配列命令が引数として指定した前記第１のレジスタ（Ｒ１１）の内容によって指定される位置に書き込まれている前記第２のレジスタ（Ｒ１０）の内容を、前記第３のレジスタ（Ｒ１２）の前記第１のレジスタ（Ｒ１１）と対応する位置に書き込む実行ユニット（ＥＸＵ）と
を備えるデータプロセッサ。
【請求項２】
前記第１の可変再配列命令は、前記第２のレジスタを引数として直接指定する
請求項１に記載のデータプロセッサ。
【請求項３】
前記第２のレジスタを指定し、前記第１の可変再配列命令によって引数として指定される第４のレジスタ（ＲＲＳ）をさらに備える
請求項１に記載のデータプロセッサ。
【請求項４】

第５のレジスタ（ＲＳＳ）
をさらに備え、
前記第４のレジスタは、該第５のレジスタを指定し、それによって、前記実行ユニットが、前記第１の可変再配列命令の実行中に、前記第１のレジスタおよび前記第４のレジスタの内容に対応する順序で、前記第２のレジスタおよび前記第５のレジスタの内容の少なくとも一部を前記第３のレジスタに書き込む
請求項３に記載のデータプロセッサ。
【請求項５】
第４のレジスタ、第５のレジスタ、および第６のレジスタ
をさらに備え、
前記実行ユニットは、前記第４のレジスタを引数として指定する第２の可変再配列命令をさらに実行し、前記実行ユニットは、前記第２の可変再配列命令の実行中に、前記第１のレジスタの内容に対応する順序で、前記第５のレジスタの内容の少なくとも一部を前記第６のレジスタに書き込み、前記実行は、少なくとも１つの他の命令に応じてサブワード単位で、前記第３のレジスタの内容および前記第６のレジスタの内容を連結する
請求項１に記載のデータプロセッサ。
【請求項６】
第１の組のレジスタ（Ｒ１０）にテーブルエントリを書き込むことと、
前記第１の組のレジスタ（Ｒ１０）の内容の少なくとも一部を再配列して第２の組のレジスタ（Ｒ１２）に書き込む可変再配列命令を使用することと
を含み、
前記可変再配列命令は、該可変再配列命令によって引数として指定される第３の組のレジスタ（Ｒ１１）の内容によって指定される位置に書き込まれている前記第１の組のレジスタ（Ｒ１０）の内容を、前記第２の組のレジスタ（Ｒ１２）の前記第３のレジスタ（Ｒ１１）と対応する位置に書き込む
並列テーブルルックアップ方法。
【請求項７】
前記第２の組のレジスタに保持されるデータをサブワード単位で連結すること
をさらに含む請求項６に記載の並列テーブルルックアップ方法。
【請求項８】
前記可変再配列命令は、前記第１の組のレジスタを指定する
請求項６に記載の並列テーブルルックアップ方法。
【請求項９】
前記可変再配列命令は、前記第１の組のレジスタを指定する第４の組の１つまたは複数のレジスタ（ＲＲＳ）を指定する
請求項６に記載の並列テーブルルックアップ方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、コンピュータに関し、詳細には、コンピュータの命令セットに関する。
本発明の主目的は、多数のテーブルルックアップの実行時のコンピュータ性能を改善することである。
【背景技術】
【０００２】
デジタルビデオ、暗号化、および符号変換は、ｙ＝ｆ（ｘ）の形式の関数が、多数の入力値に繰り返し適用される３つの用途である。
一般に、適用される関数は、数式として表すことができる。
しかし、関数によっては、アドレスに対応する独立変数（例えばｘ）およびそのアドレスの内容に対応する従属変数（例えばｙ）を有するルックアップテーブルの形式で実施するほうが便利なことが多いものもある。

【０００３】
ルックアップテーブルは、従来、メインメモリに保持される。
テーブルは、通常、メモリでは、複数の要素からなるアレイとして編成される。
アレイでは、各要素は、１つのテーブルエントリの値を保持する。
したがって、命令は、単に、所望のメモリ位置を指定するだけで、そのメモリ位置の内容が読み出されて、所望の関数の結果が提供される。
このようなルックアップ（lookups：参照）は、非常に高速で実施できるが、それでもなお、例えばビデオ処理、暗号化、および符号変換といった、このようなルックアップが非常に多く必要とされる用途では、性能を制限することがある。
【発明の開示】
【発明が解決しようとする課題】
【０００４】
テーブルルックアップが繰り返し必要とされる時の性能をさらに改善する方法が必要とされている。
【課題を解決するための手段】
【０００５】
本発明は、コンピュータプログラムおよびデータプロセッサ命令セットに可変再配列「Ｍｕｘ」命令を提供する。
具体的には、本発明は、一組の１つまたは複数の「テーブル」レジスタの内容を、一組の１つまたは複数の「インデックス」レジスタの内容の関数として再配列することを提供する。
その結果として再配列されたデータは、一組の１つまたは複数の「結果」レジスタに置くことができる。
本明細書で、「テーブル」、「インデックス」、および「結果」の資格を有するレジスタは、すべて汎用レジスタであり、そのラベルは、Ｍｕｘ命令によって提供される状況（context）に依存することに留意されたい。
【０００６】
この新規なＭｕｘ命令は、所望の順序を表すデータを一組のインデックスレジスタに置くことによって再配列を任意に指定できる点で、「可変」である。
したがって、可変Ｍｕｘ命令は、一組のインデックスレジスタを指定する。
この指定は、「明示的に」行うことができ、この場合、インデックスレジスタの識別情報（identity）が、命令の引数として指定される。
すなわち、この識別情報は、命令フィールドに入力されるデータで記述される。
あるいは、インデックスレジスタの指定は、「暗黙的に」行うことができる。
例えば、専用インデックスレジスタの識別情報を命令内に本来的に備えておくことができる。
同様に、結果レジスタおよびテーブルレジスタも、可変Ｍｕｘ命令によって暗黙的または明示的に指定することができる。
【０００７】
可変Ｍｕｘ命令が、一組のテーブルレジスタを（暗黙的または明示的に）指定する場合、その可変Ｍｕｘ命令は、「直接」とみなされる。
「間接」可変Ｍｕｘ命令では、テーブルレジスタは、直接識別されず、Ｍｕｘ命令によって直接指定される１つまたは複数のインデックスレジスタによって間接的に識別される。
例えば、間接可変Ｍｕｘ命令は、テーブル選択インデックスレジスタおよびサブワード選択インデックスレジスタの２つのインデックスレジスタを指定することができる。
テーブル選択インデックスレジスタは、要求された各エントリのテーブルレジスタを選択し、サブワード選択インデックスレジスタは、要求された各エントリの、選択されたテーブルレジスタ内のサブワード位置を選択する。
あるいは、単一のインデックスレジスタを使用して、インデックスレジスタのサブワー

ドの上位が、テーブルレジスタを選択することができる一方、そのサブワードの下位ビットが、選択されたテーブルレジスタの所望のサブワードを選択する。

【０００８】

テーブル全体を単一のテーブルレジスタで表すことができる場合、各テーブルエントリの位置は、そのレジスタ内のサブワードアドレスとして識別することができる。

「直接」可変Ｍｕｘ命令は、単一のテーブルレジスタに加えて、単一の結果レジスタおよびインデックスレジスタを指定することができる。

Ｍｕｘ命令が実行されると、テーブルレジスタの内容は、インデックスレジスタによって指定されたサブワードの順序で、結果レジスタに選択的に書き込まれる。

【０００９】

ルックアップテーブルがあまりにも大きく、単一のレジスタで表せない場合、そのルックアップテーブルは、２つ以上のテーブルレジスタ間で分割され得る。

この分割は、垂直方向に行うこともできるし、水平方向に行うこともでき、あるいは、その両方向に行うこともできる。

分割が垂直方向のみで行われている場合、エントリは、原型を保っており、エントリのいくつかは、あるレジスタで表される一方、他のエントリは、１つまたは複数の他のレジスタで表される。

分割が、水平方向のみで行われている場合、各エントリは、（通常、上位下位などの位置によって）２つ以上のレジスタ間で分割される。

それらレジスタのそれぞれは、すべてのエントリの一部を表す。

垂直方向の分割および水平方向の分割がともに使用される場合、どのレジスタも、すべてのエントリを表さず、どの１つのエントリ全体も表さない。

例えば、テーブルは、４つのレジスタ間で分割することができ、４つのレジスタのそれぞれは、テーブルエントリを二分し、さらに二分したそれぞれを二分したものを表す。

【００１０】

垂直方向のみでの分割の場合、各テーブルレジスタは、一部のテーブルエントリを保持するが、すべてのテーブルエントリを保持するとは限らない。

例えば、８つのテーブルエントリが２つのテーブルレジスタ間で分割された場合、一方のテーブルレジスタは、４つの最下位アドレスのエントリを保持できる一方、他方のテーブルレジスタは、４つの最上位アドレスのエントリを保持できる。

次に、各テーブルエントリは、特定のレジスタにより特定のサブワード位置に保持される。

垂直方向に分割されたテーブルのエントリの再配列において十分な柔軟性を可能にするために、本発明は、垂直方向に分割されたテーブルと共に間接可変ｍｕｘ命令を使用することを提供する。

【００１１】

水平方向のみで分割されたテーブルの場合、各テーブルエントリは、２つ以上のテーブルレジスタ間で分割されるが、すべてのエントリが、各テーブルレジスタで表される。

このような場合、同じインデックスレジスタを使用して、直接可変Ｍｕｘ命令を各テーブルレジスタに適用することができ、それによって、共通の再配列が、各エントリの各セグメントに適用されることになる。

通常、各テーブルレジスタに対して個々の結果レジスタが存在する。

各結果レジスタは、所望の各エントリのセグメントを保持する。

例えば「ＵｎＰａｃｋ（展開）」命令といった命令が追加され、この命令を使用して、エントリのセグメントを連結することができ、その結果、各エントリは、レジスタ内で完全な形となる。

【００１２】

水平方向および垂直方向の双方で分割されたテーブルは、各セルがテーブルアドレスのサブセットのエントリのセグメントを表す２次元アレイを形成するものと概念化することができる。

このような場合、本発明は、列単位で間接可変Ｍｕｘ命令を適用することを提供し、それにより、それぞれのエントリのセグメントに対して、所望の再配列が行われる。
オプションとして、Ｕｎｐａｃｋ命令または他の連結命令を使用して、レジスタ内で完全なエントリを組み立てることができる。
【００１３】
「Ｍｕｘ」命令は、データの並べ替え（permutation）を可能にするので、時に、「ｐｅｒｍｕｔｅ（並べ替え）」命令と呼ばれることがある。
しかしながら、Ｍｕｘ命令は、普通は並べ替えとみなされない他の形の再配列も可能である。
例えば、テーブルレジスタのすべてのサブワードが結果レジスタに表される必要があるとは限らず、テーブルレジスタの１つまたは複数のサブワードを、結果レジスタに２回以上表すことができる。
【００１４】
本発明は、より幅の広いより多くの並列性を有するデータプロセッサに向かっているコンピュータアーキテクチャの流行（trend：傾向）を活用する。
この流行と調和させた場合、メモリユニットを追加するよりも付加的な計算機能ユニットを追加することが容易（より単純で、かつ、より経済的）である。
テーブルをメモリに置くと、スループットは、メモリポートの個数によって制限される。
ルックアップテーブルをレジスタに置くと、スループットは、並列計算ユニットの個数によって制限される。
計算ユニットの個数が、メモリポートの個数を上回るにつれて、レジスタに保持されたテーブルに可変Ｍｕｘ命令を使用するというこの新規なアプローチは、従来技術をますます上回って有利になる。
【００１５】
本明細書で説明する可変Ｍｕｘ命令は、従来技術で使用される固定Ｍｕｘ命令を可変に変形したものである。
固定Ｍｕｘ命令は、結果レジスタに表すサブワードの順序を、計算した値によって指定できないので、そのテーブルルックアップの有用性において、はるかに多くの制限を受ける。
本発明の可変Ｍｕｘ命令は、サブワードの任意の順序を可能にし、したがって、単一のＭｕｘ命令を（大きなテーブルに必要とされる、結果の付加的な組み立てと共に）使用して、複数のテーブルエントリに並列にアクセスすることが可能である。
一方、固定Ｍｕｘ命令を使用するコンピュータアーキテクチャには、可変Ｍｕｘ命令が使用できるデータパスがすでに配備されている。
本発明は、１つの命令につき複数のテーブルルックアップが実行可能であるので、従来技術に対して劇的な性能向上を提供する。
本発明の他の特徴および利点は、図面を参照した以下の詳細な説明から明らかである。
【発明を実施するための最良の形態】
【００１６】
図１に示すプログラム１００の１命令セグメントは、６４ビットの「インデックス」レジスタＲ１１に共に保持されるｘの１６個の値に対して、関数ｙ＝ｆ（ｘ）の値を求める。
関数ｙ＝ｆ（ｘ）の値は、ルックアップテーブルＩにアクセスすることにより求められる。
ルックアップテーブルＩは、１６個のそれぞれの４ビットアドレス（ｘ）に１６個の４ビットのエントリ（ｙ）を有する。
テーブルＩのエントリは、６４ビットの「テーブル」レジスタＲ１０にリトルエンディアンの順序で配列される。
すなわち、最下位のアドレスのエントリは、レジスタＲ１０の最下位ニブル（４ビット

）に配置される。

【００１７】

この関数の値は、可変多重化命令ＭｕｘＶ　４，Ｒ１０，Ｒ１１，Ｒ１２を使用して求められる。

インデックスレジスタＲ１１からテーブルレジスタＲ１０への矢印で示すように、インデックスレジスタＲ１１の各ニブル位置（４ビットのサブワード位置）は、テーブルのアドレスを保持し、したがって、そのアドレスのエントリを保持するテーブルレジスタＲ１０のニブル位置を指し示す。

インデックスレジスタＲ１１から６４ビットの結果レジスタＲ１２への矢印で示すように、レジスタＲ１１の各ニブル位置は、結果レジスタＲ１２の各ニブル位置に対応する。

したがって、命令ＭｕｘＶ　４，Ｒ１０，Ｒ１１，Ｒ１２を次のように構文解析することができる。

すなわち、インデックスレジスタＲ１１に保持される各「インデックス」ニブルについて、各インデックスニブルにより示されたテーブルエントリ（レジスタＲ１０内）を結果レジスタＲ１２の対応するニブル位置に書き込む、というように構文解析することができる。

【００１８】

図１では、テーブル全体を単一のレジスタで表すことができるので、１命令で並列テーブルルックアップを行うことができる。

テーブルが、単一のレジスタに収まらない場合に、本発明は、２つ以上のレジスタ間でテーブルを分割することを提供する。

【００１９】

図２では、テーブルＩＩが、水平方向に分割されて、並列ルックアップが提供される。

テーブルＩＩは、１６個の８ビットエントリを含み、合計１２８ビットを含む。

２つの６４ビットレジスタＲ２０およびＲ２１が、テーブルＩＩの内容を保持するのに必要とされる。

各エントリは、最上位ニブル（例えば、テーブルアドレス０［１６進］のエントリとしてＥ［１６進］）および最下位ニブル（例えば、テーブルアドレス０のエントリとしてＦ）を含む。

テーブルＩＩエントリのすべての最下位ニブルは、テーブルレジスタＲ２０に書き込まれる一方、テーブルＩＩエントリのすべての最上位ニブルは、テーブルレジスタＲ２１に書き込まれる。

この分割は、テーブルＩＩの破線で示すように、テーブルの列が４ビット幅の２つの列グループに分割される点で水平方向である。

【００２０】

関数ｙ＝ｆ（ｘ）の値を求める場合のｘの値は、インデックスレジスタＲ２２に表される（その内容は、図１のインデックスレジスタＲ１１と同一である）。

インデックスレジスタＲ２２の各ニブルは、図２のテーブルレジスタＲ２０およびＲ２１への概ね上向きの矢印で示すように、テーブルレジスタＲ２０の対応するニブルを指し示すことに加えて、テーブルレジスタＲ２１の対応するニブルも指し示す。

インデックスレジスタＲ２０の各ニブル位置は、図２のレジスタＲ２３およびＲ２４への垂直下向きの矢印で示すように、結果レジスタＲ２３の対応するニブル位置を有し、また、結果レジスタＲ２４の対応するニブル位置も有する。

【００２１】

図２に表すプログラム１００のセグメントは、２つの可変Ｍｕｘ命令を使用する。

ＭｕｘＶ　４，Ｒ２０，Ｒ２２，Ｒ２３は、インデックスレジスタＲ２２の内容によって指示されるように、テーブルレジスタＲ２０のエントリを結果レジスタＲ２３に書き込む。

同様に、ＭｕｘＶ　４，Ｒ２２，Ｒ２２，Ｒ２４は、インデックスレジスタＲ２２の内容によって指示されるように、テーブルレジスタＲ２１のエントリを結果レジスタＲ２４

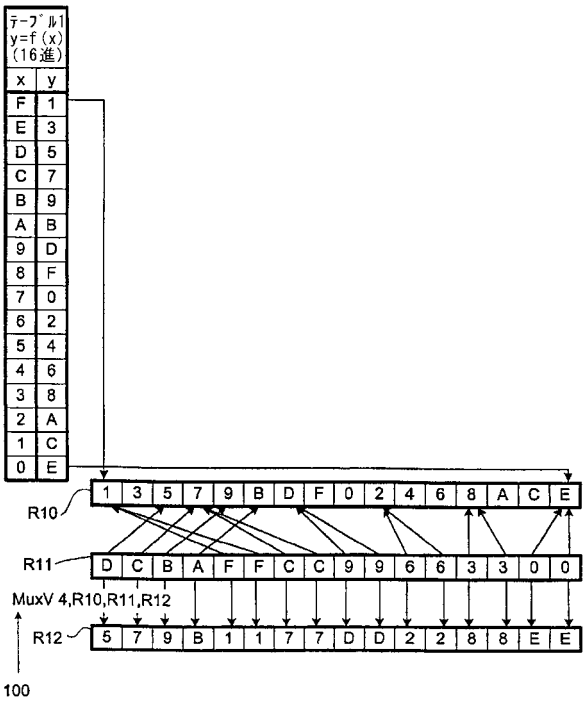
に書き込む。
２つの命令は、異なるテーブルおよび異なる結果レジスタを参照する一方で、同じインデックスレジスタＲ２２を共有することに留意されたい。
その結果、結果レジスタＲ２３の最下位エントリニブルの順序は、結果レジスタＲ２４の最上位エントリニブルの順序に対応する。
【００２２】
次に、ＵｎＰａｃｋ命令は、エントリを「リアセンブル」するのに使用することができる。
ＵｎＰａｃｋ　Ｌ，４　Ｒ２３，Ｒ２４，Ｒ２５は、レジスタＲ２３の最下位ニブルを、レジスタＲ２５の最下位ニブル位置に書き込み、レジスタＲ２４の最下位ニブルをレジスタＲ２５の第２の最下位ニブル位置に書き込む。
その結果、インデックスレジスタＲ２２の最下位ニブルによって要求されたエントリの全８ビットが、レジスタＲ２５の最下位バイト（８ビット）に書き込まれる。
同様に、次の７つの最下位のテーブルルックアップは、レジスタＲ２５の他の７つのバイト位置に表される。
同様に、ＵｎＰａｃｋ　Ｈ，４　Ｒ２３，Ｒ２４，Ｒ２６は、レジスタＲ２３およびＲ２４のそれぞれの４つの最上位ニブルをバイトに連結し、それらをレジスタＲ２６にバイトのように置く。
レジスタＲ２５およびＲ２６は、集合的に、インデックスレジスタＲ２２によって指定された順序のテーブルエントリのバイトからなる単一の１２８ビット値Ｖ２７を表すものとみなすことができる。
【００２３】
図２の所望の結果を得るには、４つの命令が必要とされる。
特定用途向けの実施の形態では、単一の可変Ｍｕｘ命令ＭｕｘＶｈ　４，８　Ｒ２０，Ｒ２１，Ｒ２２，Ｒ２５，Ｒ２６を使用して、所望の結果を直接達成することができる。
しかしながら、このアプローチは、ハードウェアの複雑さが追加され、３回のレジスタ読み出しおよび２回のレジスタ書き込みを伴う。
また、インデックスの長さ（４ビット）は、エントリの長さ（８ビット）と異なるので、２つのビット長パラメータが必要とされることに留意されたい。
【００２４】
もちろん、エントリが分割されていない場合には、エントリはリアセンブルされる必要はなく、ＵｎＰａｃｋ命令は必要とされない。
図３のテーブルＩＩＩ（図２のテーブルＩＩと同じデータを表す）は、水平方向に分割される代わりに垂直方向に分割される。
したがって、一方のテーブルレジスタＲ３０は、テーブルＩＩＩの８つの最下位アドレスの８ビットのエントリを保持する一方、他方のレジスタＲ３１は、テーブルＩＩＩの８つの最上位アドレスのエントリを保持する。
【００２５】
関数ｆ（ｘ）を求める場合のｘの値は、インデックスレジスタＲ３２に表される。
ｘの各値は、インデックスレジスタＲ３２の２つの「９」の一方に示すように、４ビットである。
インデックスの最上位ビット（この場合、「１」）は、所望のエントリがどちらのテーブルレジスタ（この場合、テーブルレジスタＲ３１）に存在するかを決定する一方、インデックスの３つの最下位ビット（この場合、「００１」）は、選択されたレジスタ内のサブワード位置（この場合、２番目の下位サブワード位置）を示すことに留意されたい。
【００２６】
図３に表すプログラム１００のセグメントは、可変Ｍｕｘ命令ＭｕｘＶｖ　８，４，Ｒ３０，Ｒ３１，Ｒ３２，Ｒ３３，Ｒ３４を使用して、結果レジスタＲ３３およびＲ３４に所望の順序でエントリを書き込む。
この命令は、３つの読み出しポートおよび２つの書き込みポートを必要とし、したがっ

て、特定用途向けの状況に最もよく適することに留意されたい。
　各テーブルレジスタに個別にｍｕｘ（多重化）演算を実行することが可能であり、その後、その結果のデータを再び結合することが可能であることに留意されたい。
　しかしながら、プログラムシーケンスは、水平方向に分割されたテーブルのアプローチの場合よりも複雑になる。
【００２７】
　一般に、インデックスのレジスタ選択ビットおよびインデックスのサブワード選択ビットは、異なる機能を実行するものであるので、個別のレジスタにそれらのビットを書き込みことが有用な場合がある。
　ほとんどのプロセッサ設計では、命令が、ある汎用レジスタを、別の汎用レジスタの内容の関数として選択することは許可されていないので、特に、レジスタ選択ビットを特殊用途のレジスタに書き込むことが好都合な場合がある。
【００２８】
　図４は、連続した６４ビットテーブルレジスタＲＴ０、ＲＴ１、ＲＴ２、およびＲＴ３に表された６４×４ビットエントリテーブルに含まれるレジスタを示している。
　レジスタ選択レジスタＲＲＳおよびサブワード選択レジスタＲＳＳの２つのインデックスレジスタが存在する。
　これらのレジスタは、集合的に、関数ｙ＝ｆ（ｘ）の値を求める場合のｘの水平方向に分割された値を保持する。
　その結果は、連続した結果レジスタＲＲ０、ＲＲ１、ＲＲ２、およびＲＲ３に書き込まれる。
　間接的な可変Ｍｕｘ命令ＭｕｘＶＩ　４，６　ＲＴ０，ＲＳＳ，ＲＲ０が、所望の並列テーブルルックアップを実行する。
【００２９】
　Ｍｕｘ命令ＭｕｘＶＩ　４，６　ＲＴ０，ＲＳＳ，ＲＲ０は、図４に示す７つのレジスタを指定するが、これらのレジスタのうち、第１のテーブルレジスタＲＴ０、サブワード選択レジスタＲＳＳ、および結果レジスタの３つのみが、明示的に指定される。
　この命令は、連続した４つのレジスタを必要とし、その第１のもの（ＲＴ０）を明示的に識別することにより、残りの３つの特定を行う。
　したがって、残りの３つは、暗黙的に指定される。
　この命令は、レジスタ選択データを特殊目的のレジスタＲＲＳに記憶することを必要とする。
　したがって、このレジスタは、暗黙的に指定される。
　この命令が４ビットサブワードの１６回の並列ルックアップを要求すると、すべての結果は、明示的に指定された結果レジスタＲＲ０に記憶することができる。
【００３０】
　ＭｕｘＶＩ命令は、１つのテーブルレジスタ（ＲＴ０）を明示的に指定し、他のレジスタ（ＲＴ１、ＲＴ２、ＲＴ３）を暗黙的に指定する一方、結果レジスタのどのサブワードにどのテーブルレジスタを使用するかを指定しない。
　その代わり、テーブルレジスタの結果レジスタのサブワードへのマッピングは、レジスタ選択レジスタＲＲＳの内容によって間接的に指定される。
　テーブルレジスタの結果レジスタのサブワードへのマッピングが、命令の内容によって直接決定されるのではなく、レジスタによって間接的に決定される点で、ＭｕｘＶＩは間接的である。
【００３１】
　レジスタ選択レジスタＲＲＳは、その内容が他のレジスタの選択を制御できるように設計された特殊目的のレジスタである。
　レジスタ選択レジスタＲＲＳは、ｙ＝ｆ（ｘ）の値を求める際に使用される各ｘにつき、４ビットを保持する。
　４つのテーブルレジスタの中からテーブルレジスタの選択を行うのに、２ビットのみが

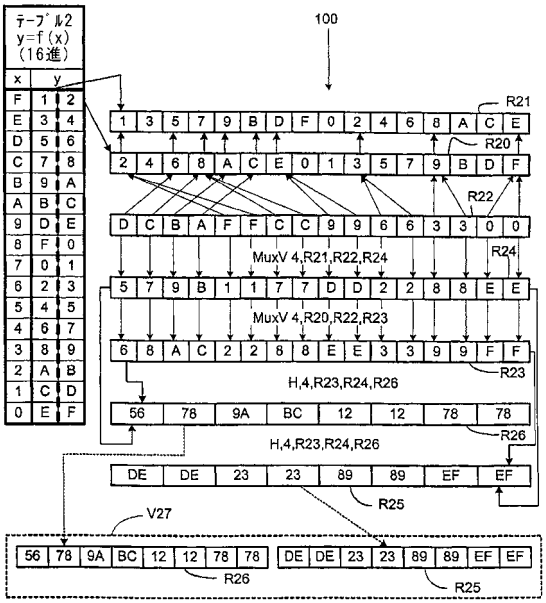
必要とされ、他の２ビットは、デフォルト値０を保持する。
テーブルデータは、レジスタ０～３、レジスタ３２～３５、レジスタ６４～６７、およびレジスタ９６～９９の４つの可能なものの１つに強制的にされる（is forced）。
【００３２】
間接的な可変Ｍｕｘ命令は、１つの命令で所望の再配列を提供する。
また、レジスタ選択ビットおよびサブワード選択ビットは、異なるレジスタに書き込まれるので、レジスタには、より多くのインデックスおよびそれによるさらに多くの並列性を得るための余裕もある。
他方で、より一般的な２つのオペランドレジスタの代わりに６つのオペランドレジスタが存在する。
４つのレジスタ内に収まることができる範囲を越えてテーブルを拡張することは、かなりのハードウェアの複雑さを伴う可能性がある。
【００３３】
本発明のいくつかの実施の形態は、レジスタの一部またはすべてが２０４８ビット以上であるマイクロプロセッサを使用し、それによって、２５６×８ビットのテーブルを単一の命令でルックアップすることを可能にする。
このようなテーブルは、各ピクセルに８ビットのデータを含むことが多いビデオ、暗号化（例えば、ＤＥＳのＳ－ｂｏｘのルックアップ）、および符号変換（例えば、ＡＳＣＩＩの別のテキストフォーマットへの変換）に役立つ。
他の実施の形態では、複数のレジスタが集合的に２０４８ビットレジスタとして機能するように、複数のレジスタを一組に構成する設備が作成される。
それ以外のものとして、上記に開示した水平方向のテーブル分割技法および垂直方向のテーブル分割技法を活用して、このような大きなテーブルを処理するものがある。
【００３４】
コンピュータの２値特性を考慮すると、多数のビットで表現された２の累乗のデータを取り扱うことが便利であることが多い。
この制約を満たすように、適合しないデータには、先頭部分にゼロを追加することができる。
例えば、７ビットのエントリに対して、それらのエントリがレジスタのバイト境界と整合するように、先頭にゼロを追加することができる。
同様に、６ビットのインデックスには、それらのインデックスがレジスタのバイト境界に適合するように、先頭に２つのゼロを追加することができる。
あるいは、適合しないデータ同士を「詰め込んで」、レジスタに収めることができるデータ量を最大にすることができる。
例えば、３ビットのインデックスによる２１回のルックアップは、６４ビットレジスタによって処理できる一方、ちょうど１６個の４ビットインデックスも、６４ビットレジスタによって処理できる。
データを連続的に詰め込んで、２の累乗に最も近くなるまで書き入れる代わりに、未使用のレジスタのビットをインデックス間で均等に分配することができる。
【００３５】
プログラム１００は、図５に示す例えばシステムＡＰ１といったコンピュータシステム上で実施される。
コンピュータシステムＡＰ１は、マイクロプロセッサ１１０およびメモリ１１２を含む。
メモリ１１２の内容には、プログラムデータ１１４、およびプログラムを構成するプログラム命令１００が含まれる。
マイクロプロセッサ１１０は、実行ユニットＥＸＵ、命令デコーダＤＥＣ、レジスタＲＧＳ、アドレスジェネレータＡＤＧ、およびルータＲＴＥを含む。
レジスタバンクＲＧＳには、本明細書で参照かつ例示されたすべてのレジスタが含まれる。

【００３６】
一般に、実行ユニットＥＸＵは、プログラム１００に従ってデータ１１４に演算を実行する。
このため、実行ユニットＥＸＵは、（内部データバスＤＴＢに付属する制御信号線を使用して）アドレスジェネレータＡＤＧに、必要な次の命令またはデータのアドレスをアドレスバスＡＤＲに沿って生成するように命令することができる。
メモリ１１２は、要求されたアドレスに保持される内容を、データ／命令バスＤＩＢに沿って供給することにより応答する。
【００３７】
ルータＲＴＥは、内部データバスＤＴＢに付属する指示子信号線に沿って実行ユニットＥＸＵから受信した指示子により決定されるように、命令を命令デコーダＤＥＣに命令バスＩＮＢを介して送り、内部データバスＤＴＢに沿ってデータを送る。
復号された命令は、制御信号線ＣＣＤを介して実行ユニットＥＸＵに供給される。
データは、通常、命令に従ってレジスタＲＧＳの内外に転送される。
【００３８】
マイクロプロセッサ１１０には、命令デコーダＤＥＣが復号でき、かつ、実行ユニットＥＸＵが実行できる複数の命令からなる命令セットＩＮＳが付随する。
プログラム１００は、命令セットＩＮＳから選択された複数の命令からなる順序付けられた組である。
説明の目的のため、マイクロプロセッサ１１０、その命令セットＩＮＳ、およびプログラム１００が、本明細書で説明されたすべての命令の例を提供する。
【００３９】
本発明は、２の累乗でないサイズを含むあらゆるサイズのレジスタを提供する。
テーブルは、任意の個数のエントリを有することができ、それらのエントリは、任意の長さのビット数を有することができる。
上述したように、並列インデックスが、１つの並列インデックスレジスタ（または複数の並列インデックスレジスタ）を満たさない場合、未使用ビットは、さまざまな方法で分配することができ、おそらく、さまざまな用途に使用することができる。
本発明は、間接的な並列ルックアップの多くの実施を提供する。
添付の特許請求の範囲によって記載される本発明は、本発明のこれらの変形および変更ならびに本発明の他の変形および変更を提供する。
【図面の簡単な説明】
【００４０】
【図１】単一の可変ｍｕｘ命令を使用して並列テーブルルックアップを実行する本発明によるプログラムのセグメントの概略図である。
【図２】複数の可変Ｍｕｘ命令に加えて、結果レジスタ間で結果を連結するＵｎＰａｃｋ命令を使用して、水平方向に分割されたテーブルに並列テーブルルックアップを実行する図１のプログラムのセグメントの概略図である。
【図３】可変Ｍｕｘ命令を使用して、垂直方向に分割されたテーブルに並列テーブルルックアップを実行する図１のプログラムのセグメントの概略図である。
【図４】間接的な可変Ｍｕｘ命令を使用して、垂直方向に分割されたテーブルに並列テーブルルックアップを実行する図１のプログラムのセグメントの概略図である。
【図５】図１のプログラムを実行する本発明のマイクロプロセッサシステムの概略図である。
【符号の説明】
【００４１】
１００・・・プログラム、
１１０・・・マイクロプロセッサ、
１１２・・・メモリ、
１１４・・・プログラムデータ、

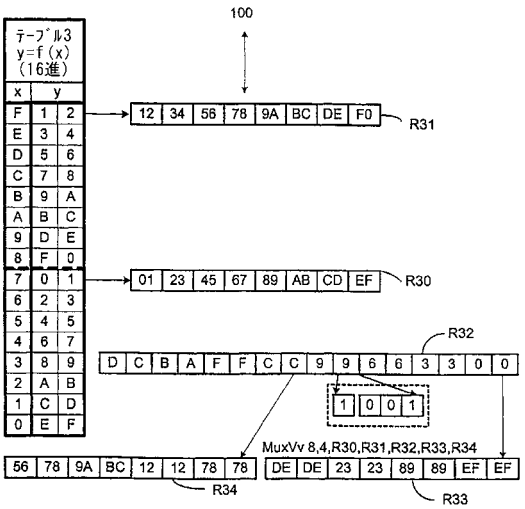
【図１】


テーブル1
y=f(x)
(16進)
x y
F 1
E 3
D 5
C 7
B 9
A B
9 D
8 F
7 0
6 2
5 4
4 6
3 8
2 A
1 C
0 E
R10
R11
MuxV 4,R10,R11,R12
R12
100

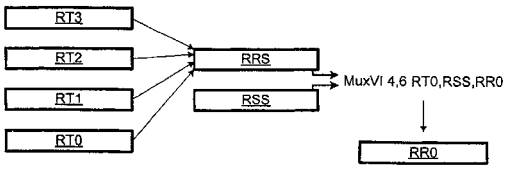
【図２】


テーブル2
y=f(x)
(16進)
100
R21
R20
R22
MuxV 4,R21,R22,R24
R24
MuxV 4,R20,R22,R23
R23
H,4,R23,R24,R26
R26
H,4,R23,R24,R26
R25
V27
R26
R25

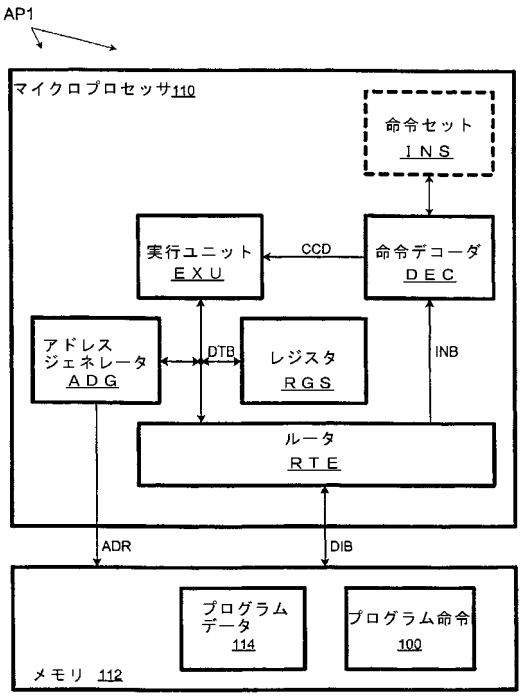
【図３】


テーブル3
y=f(x)
(16進)
100
R31
R30
R32
MuxVv 8,4,R30,R31,R32,R33,R34
R34
R33

【図４】


RT3
RT2
RT1
RT0
RRS
RSS
MuxVI 4,6 RT0,RSS,RR0
RR0

【図５】


AP1
マイクロプロセッサ110
命令セット
INS
実行ユニット
EXU
CCD
命令デコーダ
DEC
アドレス
ジェネレータ
ADG
DTB
レジスタ
RGS
INB
ルータ
RTE
ADR
DIB
プログラム
データ
114
プログラム命令
100
メモリ 112

フロントページの続き

審査官　後藤　彰

(56)参考文献　特表２００３－５３３８２９（ＪＰ，Ａ）
特開平１１－３２２６（ＪＰ，Ａ）
特開平９－２２３４３（ＪＰ，Ａ）
特開平９－３３０２１０（ＪＰ，Ａ）
特開平２－１８１８２１（ＪＰ，Ａ）
特開昭５８－２７２３９（ＪＰ，Ａ）
特開昭５４－１８２４８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　９／３０　－　９／３５５
Ｇ０６Ｆ　１／０２　－　１／０３５