

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3563724号  
(P3563724)

(45) 発行日 平成16年9月8日(2004.9.8)

(24) 登録日 平成16年6月11日(2004.6.11)

(51) Int. Cl.<sup>7</sup>

F I

G 0 6 F 9/34

G O 6 F 9/34 3 3 0

G 0 6 F 9/35

G O 6 F 9/34 3 1 0

請求項の数 7 (全 9 頁)

(21) 出願番号	特願2002-56311 (P2002-56311)	(73) 特許権者	000003078
(22) 出願日	平成14年3月1日(2002.3.1)		株式会社東芝
(65) 公開番号	特開2003-256195 (P2003-256195A)		東京都港区芝浦一丁目1番1号
(43) 公開日	平成15年9月10日(2003.9.10)	(74) 代理人	100058479
審査請求日	平成14年3月1日(2002.3.1)		弁理士 鈴江 武彦
		(74) 代理人	100084618
			弁理士 村松 貞男
		(74) 代理人	100068814
			弁理士 坪井 淳
		(74) 代理人	100092196
			弁理士 橋本 良郎
		(74) 代理人	100091351
			弁理士 河野 哲
		(74) 代理人	100088683
			弁理士 中村 誠

最終頁に続く

(54) 【発明の名称】 マイクロプロセッサ、エミュレータおよび命令セット解釈方法

## (57) 【特許請求の範囲】

## 【請求項1】

レジスタをオペランドとして指定可能な命令セットを実行するマイクロプロセッサにおいて、

前記レジスタオペランド内の上位nビットからなる上位フィールドの値が予め定められた値であったときに、前記上位フィールドを除く下位フィールドの値で指し示されるレジスタにアクセスするレジスタアクセス手段と、

前記上位フィールドの値が予め定められた値でなかったときに、その上位フィールドの値で指し示されるレジスタの格納値をベースアドレス、前記下位フィールドの値をオフセットとしてメモリにアクセスするメモリアクセス手段と

を具備することを特徴とするマイクロプロセッサ。

## 【請求項2】

前記レジスタオペランド内のすべてをレジスタ番号の格納フィールドとする通常のレジスタアクセスとして命令セットを解釈する第1の実行モードと、前記レジスタオペランド内を前記上位フィールドおよび前記下位フィールドの2つのフィールドに分割し、前記レジスタアクセス手段または前記メモリアクセス手段によるレジスタアクセスまたはメモリアクセスとして命令セットを解釈する第2の実行モードとを切り換えるモード切替手段をさらに具備することを特徴とする請求項1記載のマイクロプロセッサ。

## 【請求項3】

前記第2の実行モード時に必要となる最小限数のレジスタのみを備えた請求項2記載のマ

イクロプロセッサ。

【請求項 4】

前記モード切換手段により前記第 1 の実行モードに切り換えるときに、不足分のレジスタをメモリ上にマップするマッピング手段をさらに具備することを特徴とする請求項 3 記載のマイクロプロセッサ。

【請求項 5】

前記第 1 の実行モード時に必要となる数のレジスタを備えたプロセッサであって、前記モード切換手段により前記第 2 の実行モードに切り換えられるときに、前記レジスタアクセス手段が用いる以外のレジスタに、当該レジスタと前記メモリとの対応関係の一貫性を保持しつつ前記メモリの値をキャッシュするキャッシュ手段をさらに具備し、前記メモリアccess手段は、前記レジスタにキャッシュされた値にアクセスすることによってメモリアccessを代替的に処理することを特徴とする請求項 2 記載のマイクロプロセッサ。

10

【請求項 6】

命令セットに含まれるレジスタオペランド内のすべてをレジスタ番号の格納フィールドとしてレジスタアクセスを実行する第 1 の実行モードのみをサポートするマイクロプロセッサ上で、前記レジスタオペランド内の上位 n ビットからなる上位フィールドの値で指し示されるレジスタの格納値をベースアドレス、前記上位フィールドを除く下位フィールドの値をオフセットとしてメモリアccessを実行する第 2 の実行モード用に記述されたプログラムを動作させるためのエミュレータであって、前記メモリアccessに相当するロード/ストア命令を前記プログラムに挿入する手段を具備することを特徴とするエミュレータ。

20

【請求項 7】

レジスタをオペランドとして指定可能な命令セットを実行するマイクロプロセッサのオペランド解釈方法であって、前記レジスタオペランド内の上位 n ビットからなる上位フィールドの値が予め定められた値であったときに、前記上位フィールドを除く下位フィールドの値で指し示されるレジスタにアクセスし、前記上位フィールドの値が予め定められた値でなかったときに、その上位フィールドの値で指し示されるレジスタの格納値をベースアドレス、前記下位フィールドの値をオフセットとしてメモリにアクセスするように命令セットを解釈することを特徴とするマイクロプロセッサの命令セット解釈方法。

30

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、レジスタをオペランドとして指定可能な命令セットを実行するマイクロプロセッサ、第 1 の実行モードをサポートするマイクロプロセッサで第 2 の実行モード用に記述されたプログラムを動作させるためのエミュレータおよび同マイクロプロセッサのオペランド解釈方法に係り、特に、RISC (Reduced Instruction Set Computer) アーキテクチャのもつ高速な処理という利点を維持しつつ、コンテキストスイッチの高速化とコードサイズのコンパクト化とを実現したマイクロプロセッサ、エミュレータおよび同マイクロプロセッサのオペランド解釈方法に関する。

40

【0002】

【従来の技術】

マイクロプロセッサの命令セットアーキテクチャは、命令長 (ビット幅)、オペランドとして指定可能なレジスタの個数、アドレッシングモードなどによって特徴づけられる。RISC アーキテクチャは、多数のレジスタ、ロード/ストア命令によるメモリアccess、固定長命令、シンプルな命令フォーマットを特徴とするアーキテクチャであり、比較的命令デコードが容易であり、回路規模を小さく押えることができるため、少ない消費電力で高い動作周波数のマイクロプロセッサを実現することが可能である。一方で、多数のレジスタを使用するために、コンテキストスイッチに時間が掛かることや、ロード/ストア命

50

令の頻度が高いために、多数のレジスタを必要としないプログラムであってもコードサイズが大きくなってしまふという欠点がある。

【0003】

【発明が解決しようとする課題】

前述したように、RISCアーキテクチャでは、高速な処理という利点をもつ反面、コンテキストスイッチに時間が掛かり、また、プログラムのコードサイズが大きくなるという欠点をもっていた。

【0004】

この発明は、このような事情を考慮してなされたものであり、RISCアーキテクチャのもつ高速な処理という利点を維持しつつ、コンテキストスイッチの高速化とコードサイズのコンパクト化とを実現したマイクロプロセッサ、エミュレータおよび同マイクロプロセッサのオペランド解釈方法を提供することを目的とする。

10

【0005】

【課題を解決するための手段】

前述した目的を達成するために、この発明は、レジスタをオペランドとして指定可能な命令セットを実行するマイクロプロセッサにおいて、前記レジスタオペランド内の上位nビットからなる上位フィールドの値が予め定められた値であったときに、前記上位フィールドを除く下位フィールドの値で指し示されるレジスタにアクセスするレジスタアクセス手段と、前記上位フィールドの値が予め定められた値でなかったときに、その上位フィールドの値で指し示されるレジスタの格納値をベースアドレス、前記下位フィールドの値をオフセットとしてメモリにアクセスするメモリアccess手段とを具備することを特徴とする。

20

【0006】

また、このマイクロプロセッサは、前記レジスタオペランド内のすべてをレジスタ番号の格納フィールドとする通常のレジスタアクセスとして命令セットを解釈する第1の実行モードと、前記レジスタオペランド内を前記上位フィールドおよび前記下位フィールドの2つのフィールドに分割し、前記レジスタアクセス手段または前記メモリアccess手段によるレジスタアクセスまたはメモリアccessとして命令セットを解釈する第2の実行モードとを切り換えるモード切替手段をさらに具備することを特徴とする。

【0007】

この発明のマイクロプロセッサにおいては、たとえば多数のレジスタを使うメディア処理では第1の実行モード、コンパクトなリアルタイム処理では第2の実行モードで動作する等、性格の異なるプログラムそれぞれを1つの命令セットアーキテクチャで効率的に処理することを実現する。

30

【0008】

また、この発明は、命令セットに含まれるレジスタオペランド内のすべてをレジスタ番号の格納フィールドとしてレジスタアクセスを実行する第1の実行モードのみをサポートするマイクロプロセッサ上で、前記レジスタオペランド内の上位nビットからなる上位フィールドの値で指し示されるレジスタの格納値をベースアドレス、前記上位フィールドを除く下位フィールドの値をオフセットとしてメモリアccessを実行する第2の実行モード用に記述されたプログラムを動作させるためのエミュレータであって、前記メモリアccessに相当するロード/ストア命令を前記プログラムに挿入する手段を具備することを特徴とする。

40

【0009】

この発明のエミュレータにおいては、レジスタオペランド内のすべてをレジスタ番号の格納フィールドとする通常のレジスタアクセスとして命令セットを解釈する第1の実行モードしかサポートしないマイクロプロセッサであっても、その実行前に、メモリアccessに相当するロード/ストア命令をプログラムに挿入することによって、レジスタオペランド内を上位フィールドおよび下位フィールドの2つのフィールドに分割し、レジスタアクセスまたはメモリアccessとして命令セットを解釈する第2の実行モード用に記述されたプ

50

プログラムを動作させることを可能とする。

【0010】

【発明の実施の形態】

以下、図面を参照してこの発明の実施形態を説明する。

(第1実施形態)

まず、この発明の第1実施形態について説明する。

図1は、この第1実施形態に係るマイクロプロセッサにおける命令セット解釈の基本原則を説明するための概念図である。

【0011】

この実施形態のマイクロプロセッサは、レジスタをオペランドとして指定可能なRISC命令セットを実行するマイクロプロセッサであり、実行モードによってレジスタオペランドの解釈を変える仕組みを備えている。ここでは、このマイクロプロセッサが次の2つの実行モードをサポートするものと想定する。

【0012】

(1) ベースモード

レジスタオペランドを通常のレジスタアクセスとして解釈するモード。

【0013】

(2) 拡張モード

レジスタオペランドを2つのフィールドに分割し、上位フィールドの値が所定の値(たとえば0)である場合には、通常のレジスタアクセスとして解釈し、それ以外の値の場合には、上位フィールドの値が指し示すレジスタの内容をベースアドレス、下位フィールドの値をオフセットとするメモリアccessとして解釈するモード。

【0014】

そして、このマイクロプロセッサは、(2)の拡張モードを備えた点をその特徴としており、以下では、この拡張モードでの動作原理を詳述する。なお、このマイクロプロセッサは、複数のプログラムを時分割で並列に実行するが、あるプログラムから別のプログラムに切り換える、いわゆるコンテキストスイッチを行った際に、所定のレジスタを参照することにより、新たにディスパッチされたプログラムがベースモードおよび拡張モードのいずれの実行モードで動作するものであるのかを知得し、前述した2つの実行モード間の切り換えを実行する。

【0015】

拡張モード時、マイクロプロセッサは、命令セット10の各レジスタオペランド12について、まず、上位フィールド121の値を参照する。ここでは、上位フィールド121を先頭から2ビットとし、3ビット目以降を下位フィールド122とする例を示す。

【0016】

この上位フィールド121の値が0、つまり“00”であったとき、マイクロプロセッサは、通常のレジスタアクセスと同じ様に、下位フィールド122の値で指し示されるレジスタ1にアクセスする。一方、この上位フィールド121の値が0以外、つまり“01”、“10”、“11”のいずれかであったとき、マイクロプロセッサは、この3つの値にそれぞれ割り当てられた所定のレジスタ1を参照し、そのレジスタ1の値をベースアドレス、レジスタオペランド12の下位フィールド122の値をオフセットとしてメモリ2にアクセスする。

【0017】

つまり、この拡張モードでは、たとえばメディア処理などのためにベースモードをサポートすべく多数(n個)のレジスタが搭載された場合であっても、利用するレジスタ数は1/4n個で済むため、コンテキストスイッチ時のレジスタの入れ替え量を大幅に削減し、その高速化を実現する。なお、図1ではやや分かりづらいが、上位フィールド121の値が所定の値以外であった場合に参照するレジスタの数は、1/4n個のレジスタの中の僅かに過ぎないものである。

【0018】

10

20

30

40

50

また、たとえば図2に示すようなコードを処理する場合を考えると、通常のRISCでは、このようなコードは図3に示すようなマシン語になるのに対し、この拡張モードを利用すれば、図4に示すようなマシン語で良い。つまり、この拡張モードは、コードサイズの2~3割を占めるといわれるロード/ストアを大幅に削減することにより、コードサイズのコンパクト化も実現する。

#### 【0019】

さらに、この拡張モードでは、レジスタアクセスに加えて、最大3箇所までのメモリアクセスを1つの命令で扱えるため、レジスタ-メモリ間、メモリ-レジスタ間、メモリ-メモリ間の演算を効率良く行うことが可能となる。

#### 【0020】

なお、ここでは、上位フィールドを先頭から2ビットとし、3ビット目以降を下位フィールドとする例を示したが、このマイクロプロセッサにおける命令セット解釈の手法は、これに限られるものではない。たとえば、上位フィールドを先頭から1ビットとし、2ビット目以降を下位フィールドとすれば、レジスタオペランドの解釈を $1/2n$ 個のレジスタを対象としたレジスタアクセスと、1箇所のメモリアクセスとの2種類を考慮しながら行えることになる。また、この上位フィールドのビット幅を可変とすることによって、拡張モードにバリエーションを持たせることも可能である。さらに、たとえばレジスタオペランドを3つに分割して、レジスタ+レジスタ+オフセットというアドレッシングをサポートすることも可能である。

#### 【0021】

(第2実施形態)

次に、この発明の第2実施形態について説明する。

図5は、この発明の第2実施形態に係るマイクロプロセッサにおける命令セット解釈の基本原理を説明するための概念図である。

#### 【0022】

この実施形態のマイクロプロセッサと前述した第1実施形態のマイクロプロセッサとの違いは、前述した第1実施形態のマイクロプロセッサが、ベースモード時に必要となる多数( $n$ 個)のレジスタを実装するのに対し、この実施形態のマイクロプロセッサでは、拡張モードで必要となる数(たとえば $1/4n$ 個)のみを実装する点にある。そして、この実施形態のマイクロプロセッサは、この拡張モードで必要となる数のレジスタのみで、ベースモード用に記述されたプログラムを動作させることを可能とした点をその特徴とする。以下、この点について詳述する。

#### 【0023】

前述したレジスタ数でベースモード用に記述されたプログラムを動作させるためには、不足分のレジスタを何らかの形で補わなければならない。そこで、このマイクロプロセッサは、図5に示すように、この不足分のレジスタをメモリ上にマップし、そのマップ先のメモリ上のアドレスを所定のレジスタに格納する。

#### 【0024】

いま、このマイクロプロセッサが、第1実施形態で説明した拡張モードをサポートしているものと想定する。この場合、このマイクロプロセッサは、ベースモード時に、 $3/4n$ 個のレジスタをメモリ上にマップする。その後、マイクロプロセッサは、拡張モードと同じように動作する。

#### 【0025】

具体的には、マイクロプロセッサは、ベースモードの命令セット10の各レジスタオペランド12について、まず、先頭から2ビットの上位フィールド121の値を参照し、この上位フィールド121の値が0、つまり“00”であったとき、通常のレジスタアクセスと同じ様に、上位フィールド121以外の下位フィールド122の値で指し示されるレジスタ1にアクセスする。一方、この上位フィールド121の値が0以外、つまり“01”、“10”、“11”のいずれかであったとき、マイクロプロセッサは、たとえばこの3つの値にそれぞれ割り当てべく別途設けられたレジスタ1bを参照し、そのレジスタ1

10

20

30

40

50

bの値をベースアドレス、レジスタオペランド12の下位フィールド122の値をオフセットとしてメモリ2にアクセスする。

【0026】

このように、このマイクロプロセッサは、少数のレジスタ構成であるにも関わらず、多数のレジスタを使うメディア処理用などのプログラムを動作させることを可能とする。

【0027】

(第3実施形態)

次に、この発明の第3実施形態について説明する。

図6は、この発明の第3実施形態に係るマイクロプロセッサにおける命令セット解釈の基本原理を説明するための概念図である。

10

【0028】

この実施形態のマイクロプロセッサと前述した第1実施形態のマイクロプロセッサとの違いは、この実施形態のマイクロプロセッサでは、拡張モード時に、レジスタアクセスで利用する1/4n個以外のレジスタをメモリアクセス用のキャッシュエリアとして活用する点にある。

【0029】

いま、上位フィールドの値をa、下位フィールドの値をb、この2つのフィールドを含むレジスタオペランドの値を(a, b)と表すものとする。そして、この(a, b)によってアクセスするメモリをワードサイズに限定すると、上位フィールド121の各値ごとに確保されたキャッシュエリアたるレジスタ群内でのレジスタ番号とメモリアドレスとの間

20

に対応関係ができるので、このマイクロプロセッサは、そのコンシステンシを保つことによ

って、高速なメモリアクセスを実現する。

【0030】

より具体的には、レジスタaの内容を(a)、ワードサイズをw、下位フィールドのビット幅をnとすると、

・メモリアドレス：  $(a) + (b \times w)$

・レジスタ番号： $((a) / w + b) \bmod 2^n$

を上位フィールド121の各値ごとに確保されたキャッシュエリアたるレジスタ群内でのレジスタ番号とメモリアドレスとの間の対応関係として定義する。そして、マイクロプロセッサは、この対応関係を保ちながら、メモリ2の値をレジスタ1にキャッシングする(

30

図6参照)。

【0031】

こうすると、たとえばレジスタaの値がワードサイズだけインクリメントまたはデクリメントされた場合、このレジスタaをベースとする $2^n$ 個のレジスタのうち、値がダーティとなるレジスタを1つのみとすることができる。

【0032】

このように、このマイクロプロセッサは、拡張モード時、レジスタアクセスで用いられるレジスタ以外のレジスタ群をメモリのキャッシュエリアとして活用することにより、メモリアクセスを高速に処理することを可能とする。

【0033】

ところで、前述の第1乃至第3実施形態では、レジスタオペランドを2つのフィールドに分割し、上位フィールドの値が所定の値である場合には、通常のレジスタアクセスとして解釈し、それ以外の値の場合には、上位フィールドの値が指し示すレジスタの内容をベースアドレス、下位フィールドの値をオフセットとするメモリアクセスとして解釈する拡張モードをサポートするマイクロプロセッサについて説明してきた。そこで、次に、この拡張モード用に記述されたプログラムを通常のベースモードのみをサポートするマイクロプロセッサで動作させることについて触れておく。

40

【0034】

そのために、ここでは、プログラムの実行前に、メモリアクセスに相当するロード/ストア命令をプログラムに埋め込むためのエミュレータを用意することを考える。この場合、

50

ロード/ストアを行う時に一時的に使用するレジスタが必要になるが、これには、たとえばレジスタ番号が ( a , b )、つまり ( a < < n ) + b のレジスタを使用すれば良い。

【 0 0 3 5 】

なお、本願発明は、前記実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。更に、前記実施形態には種々の段階の発明が含まれており、開示される複数の構成要件における適宜な組み合わせにより種々の発明が抽出され得る。たとえば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題が解決でき、発明の効果の欄で述べられている効果が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

10

【 0 0 3 6 】

【発明の効果】

以上のように、この発明によれば、レジスタオペランドを通常のレジスタアクセスとして解釈する第1の実行モードと、レジスタオペランドを2つのフィールドに分割し、上位フィールドの値が所定の値である場合には、通常のレジスタアクセスとして解釈し、それ以外の値の場合には、上位フィールドの値が指し示すレジスタの内容をベースアドレス、下位フィールドの値をオフセットとするメモリアccessとして解釈する第2の実行モードとをもつことにより、RISCアーキテクチャのもつ高速な処理という利点を維持しつつ、コンテキストスイッチの高速化とコードサイズのコンパクト化とを実現する。

【図面の簡単な説明】

20

【図1】この発明の第1実施形態に係るマイクロプロセッサにおける命令セット解釈の基本原理を説明するための概念図。

【図2】同第1実施形態のマイクロプロセッサで処理されるコードを例示する図。

【図3】図2に示したコードを通常のベースモード用としてマシン語にした場合の例を示す図。

【図4】図2に示したコードを同第1実施形態のマイクロプロセッサがサポートする拡張モード用としてマシン語にした場合の例を示す図。

【図5】同第2実施形態に係るマイクロプロセッサにおける命令セット解釈の基本原理を説明するための概念図。

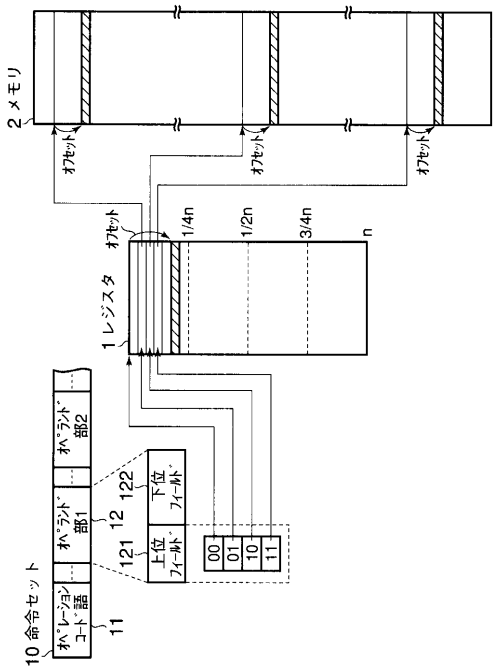
【図6】同第3実施形態に係るマイクロプロセッサにおける命令セット解釈の基本原理を説明するための概念図。

30

【符号の説明】

- 1 ... レジスタ
- 2 ... メモリ
- 1 0 ... 命令セット
- 1 1 ... オペレーションコード部
- 1 2 ... オペランド部
- 1 2 1 ... 上位フィールド
- 1 2 2 ... 下位フィールド

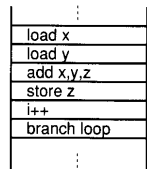
【 図 1 】



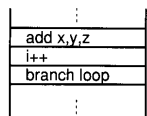
【 図 2 】

```
for(i=0;i<SIZE;i++)s[i].x+s[i].y;
```

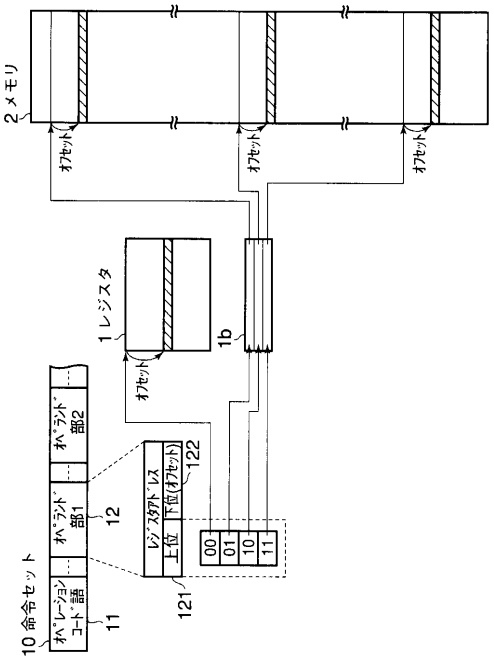
【 図 3 】



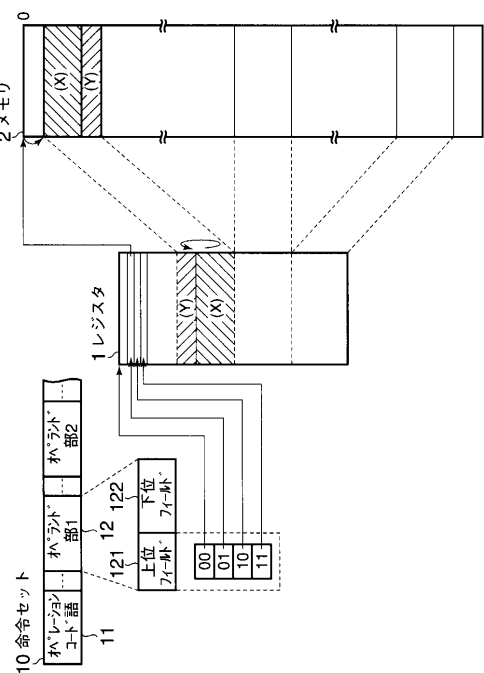
【 図 4 】



【 図 5 】



【 図 6 】



---

フロントページの続き

(74)代理人 100070437

弁理士 河井 将次

(72)発明者 境 隆二

東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内

審査官 後藤 彰

(56)参考文献 特開平10-49369(JP,A)

(58)調査した分野(Int.Cl.<sup>7</sup>, DB名)

G06F 9/30 - 9/355