

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3649470号  
(P3649470)

(45) 発行日 平成17年5月18日(2005.5.18)

(24) 登録日 平成17年2月25日(2005.2.25)

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

G06F 9/30

F I

G06F 9/30 310C

請求項の数 6 (全 11 頁)

|              |                              |           |                       |
|--------------|------------------------------|-----------|-----------------------|
| (21) 出願番号    | 特願平7-143544                  | (73) 特許権者 | 594154428             |
| (22) 出願日     | 平成7年6月9日(1995.6.9)           |           | エイアールエム リミテッド         |
| (65) 公開番号    | 特開平8-44557                   |           | イギリス国 シービー1 9エヌジェイ    |
| (43) 公開日     | 平成8年2月16日(1996.2.16)         |           | ケンブリッジ, チェリー ヒントン, フル |
| 審査請求日        | 平成12年1月20日(2000.1.20)        |           | バーン ロード 110           |
| 審査番号         | 不服2002-15642(P2002-15642/J1) | (74) 代理人  | 100066692             |
| 審査請求日        | 平成14年8月15日(2002.8.15)        |           | 弁理士 浅村 皓              |
| (31) 優先権主張番号 | 9411670 4                    | (74) 代理人  | 100072040             |
| (32) 優先日     | 平成6年6月10日(1994.6.10)         |           | 弁理士 浅村 肇              |
| (33) 優先権主張国  | 英国 (GB)                      | (74) 代理人  | 100094673             |
|              |                              |           | 弁理士 林 拓三              |
|              |                              | (74) 代理人  | 100091339             |
|              |                              |           | 弁理士 清水 邦明             |

最終頁に続く

(54) 【発明の名称】 データ処理装置

(57) 【特許請求の範囲】

【請求項1】

データ処理装置であって、

所定の複数の命令セットの連続するプログラム命令ワードを実行する手段を有するプロセッサコアであって、各命令ワードは少なくとも2以上の偶数バイト長である、前記プロセッサコアと、

第1の命令セットのプログラム命令ワードをデコードするための第1の命令デコーダと、

第2の命令セットのプログラム命令ワードをデコードするための第2の命令デコーダと、

実行すべき命令ワードを記憶するデータメモリ内の次のプログラム命令ワードのアドレスを示すプログラムカウンタレジスタと、

カレントプログラム命令ワードにตอบสนองして前記プログラムカウンタレジスタの内容を変更する手段であって、変更された内容は次のプログラム命令ワードのアドレスと前記第1および第2の命令セットのいずれが次のプログラム命令ワードに対応するかの表示との双方を表している、前記変更する手段と、

前記プログラムカウンタレジスタの1つ以上の所定の表示ビットにตอบสนองして、前記第1の命令デコーダまたは前記第2の命令デコーダを制御してカレントプログラム命令ワードをデコードするための、および、前記プロセッサコアを制御して前記第1および第2の命令セットから選択されたカレント命令セットのデコードされたプログラム命令ワードを実

行するための制御手段であって、該選択された命令セットとその対応命令デコーダとは前記プログラムカウンタレジスタの前記1つ以上の表示ビットの状態によって指定され、前記実行すべきプログラム命令ワードの前記データメモリ内のアドレスは前記プログラムカウンタレジスタによって指定され、前記プログラムカウンタレジスタの前記1つ以上の表示ビットが、前記プログラムカウンタレジスタの1つ以上の最下位ビットである、前記制御手段と

を備えた前記データ処理装置。

【請求項2】

請求項1記載のデータ処理装置において、

前記第1命令セットのプログラム命令ワードが、Xビットのプログラム命令ワードであり、

前記第2の命令セットのプログラム命令ワードが、Yビットのプログラム命令ワードであり、

YがXと異なる、

前記データ処理装置。

【請求項3】

請求項2記載のデータ処理装置において、

Yが16であり、Xが32である、

前記データ処理装置。

【請求項4】

請求項1から請求項3のいずれかに記載のデータ処理装置において、ただ1つの表示ビットが使用されている、前記データ処理装置。

【請求項5】

請求項1から請求項4のいずれかに記載のデータ処理装置において、実行すべきプログラム命令ワードを記憶するデータメモリを含む、前記データ処理装置。

【請求項6】

請求項5記載のデータ処理装置において、前記データメモリに記憶されたプログラム命令ワードにアクセスするための手段を含み、該アクセスする手段が、前記プログラムカウンタレジスタの前記1つ以上の表示ビットには応答しない、前記データ処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明はデータ処理技術に関し、より詳細には、複数個のプログラム命令ワードを用いたデータ処理に関する。

【0002】

【従来技術】

データ処理システムはプログラム命令ワードの制御により作動するプロセッサコアにより作動し、プロセッサ命令ワードはデコードされるとプロセッサ内の個々の素子を制御するコア制御信号を発生し、プログラム命令ワード内に指定された処理を行うよう、必要な演算を実行する。

【0003】

従来、異なる命令セット間でその使用を切り替える手段を設けることによって、2以上の命令セットからなるプログラム命令ワードを実行するシステムが提供されていた。デジタルイクイップメント社のVAX11コンピュータはVAX命令モード並びにPDP11コンピュータの従来機種のための命令をデコードを可能にするコンパチブルモードの双方を有している。

【0004】

異なる命令セットへの切り替えを行うための、命令セット切替は、プロセッサコアへのハードウェア配線であっても良いが、これは命令セットの切替のために、プロセッサの物理的な再配線を必要とする。この方法とは異なり、使用される現在の命令セットを指定する

10

20

30

40

50

ようにプロセッサレジスタを使用することもできる。この場合命令セットを指定する値をそのプロセッサレジスタに書き込むことにより、作動中のソフトウェアによりカレント命令セットを選択することができる。しかしながら以下に述べるように、この技術は別のプログラム命令ワードを必要とするので、このことによりソフトウェア作成に余分な時間がかかり、更にプログラム命令ワードを記憶する余分なメモリスペースが必要になる。

【 0 0 0 5 】

1つのコード片を実行するために、2つ以上の命令セットを使用可能なプロセッサは次の2つの情報を有していなければならない。

- 1) メモリ内のコードのアドレス及び
- 2) 使用する命令セット(すなわちコードが書き込まれる命令セット)

10

【 0 0 0 6 】

【 発明が解決しようとする課題 】

一般にこれまで提案されたプロセッサでは、異なる命令内のルーチンの呼び出しには、次に述べる処理を実行しなければならなかった。

1) サブルーチンの呼び出し先を、オリジナルから、自動的に生成される命令セットシーケンス即ちベニヤ(vener)に切り替える。

2) 次にベニヤは、次を実行しなければならない。

- 呼び出し側のコンテキストをセーブ
- 正しい命令セットを選択
- オリジナルのルーチンを呼び出す
- オリジナルのルーチンから復帰する際に元の命令セットを選択
- 呼び出し側のコンテキストを復帰

20

【 0 0 0 7 】

このプロセスはリンカーと称される従来のソフトウェアツールを使用することにより、プログラマーに対して比較的見通しの良いものとなるようにすることができる。しかしながらこのプロセスは異なる命令セットから呼び出される命令ごとに5命令のオーバーヘッドを有しており、かなりの処理オーバーヘッドがもたらされる。

【 0 0 0 8 】

本発明の目的は複数の命令セットを切り替えるデータ処理装置の能力を向上させることにある。

30

【 0 0 0 9 】

【 課題を解決するための手段 】

本発明は、所定の複数の命令セットの連続するプログラム命令ワードを実行する手段を有するプロセッサコアと、

実行すべきプログラム命令ワードを記憶するためのデータメモリと、

前記データメモリ内の次のプログラム命令ワードのアドレスを表示するためのプログラムカウンタレジスタと、

カレントプログラム命令ワードに回答して前記プログラムカウンタレジスタの内容を変更するための手段と、

前記プログラムカウンタレジスタの1つ以上の所定の表示ビットに回答し、前記所定の複数の命令セットから選択され、前記プログラムカウンタレジスタの前記1つ以上の表示ビットの状態によって指定されるカレント命令セットのうちのプログラム命令ワードを実行するよう、前記プロセッサコアを制御するための制御手段とを備えたデータ処理装置を提供するものである。

40

【 0 0 1 0 】

本発明ではプログラムカウンタレジスタ内にカレント命令セットを選択するための、1つ或いは複数の制御フラグが設けられる。これにより、例えばランチ命令の実行の一部としてプログラムカウンタレジスタ内に新しい値が書き込まれる際、カレント命令セットを変更することが可能となる。

【 0 0 1 1 】

50

本発明は、上記の、これまで提案されているプロセッサの場合と同じように、別個のプロセッサレジスタ（命令セットレジスタとプログラムカウンタレジスタ）で、要求された命令セットとその次の命令アドレスをエンコードする場合、異なる命令セットで書かれたコードセクションの呼び出しを行うのに、2つの別個のレジスタを更新しなければならないので、命令セットを切り替えることが困難になることに注目したものである。

【0012】

例えば、ソーティングすなわち照合（collation）機能を実行するプログラムについて検討する。一般にこのソートを実行するには、一般的なソートルーチン呼び出す。このソートルーチンは一般的であるので、所定のシーケンスでソーティングできるものでなければならない。例えば数字の順序、アルファベットの順序、ケース（case）に依存しないアルファベットの順序またはプログラマーが指定する他の順序で項目をソートするように、呼び出し可能である。プログラマーがソートの順序を指定する手段は、ルーチンのアドレス（比較ルーチンと称される）をソートルーチンに送ることである。この比較ルーチンは次にソートルーチンにより呼び出され、2つの項目のデータが与えられると、ソートされるシーケンス内で第1の項目を第2の項目の前に置くか、或いは後に置くかを示す値を返す。

10

【0013】

比較ルーチンのアドレスがソートルーチンへ送られると、ソートルーチンは、ルーチンが呼び出される際にどの命令セットが選択されるべきかを知る方法はない。比較ルーチンの実行が試みられる際に、誤った命令セットがカレント命令セットとなっている場合、その結果はひどい失敗をもたらす得る。比較ルーチンが呼び出される際に、どの命令セットが有効にされるべきかを示すため、ソートルーチンに余分な情報を送る必要がある。しかしながらハイレベルの言語、例えばCおよびC++言語で書かれた多くの現行のプログラムでは、ターゲットルーチン（本例ではアドレスおよび命令セット情報）を一義的に識別するのに必要なすべての情報は、単一の機械語で表現できるものとしている。

20

【0014】

本発明は、使用される命令セットを示すための、プログラムカウンタレジスタ（PC）の所定のビットを定義することにより、これらの問題を解消している。上記特定の実施例では、ソートルーチンへ送られる比較ルーチンのアドレスは、そのアドレスの所定のビット内にエンコードされた必要な命令セットを有することができる。比較ルーチンが呼び出されると、表示ビットを含むアドレスはプログラムカウンタレジスタに送られるだけである。

30

【0015】

プログラムカウンタレジスタの所定ビットを、表示ビット（indicator bit）として使用するため確保することもできるが、別の方法では、対応するメモリエリア内の種々の命令セットを用いて、実行されるコードの部分を記憶し、これにより、そのメモリエリアがアクセスされる間、プログラムカウンタは、使用される適当な命令セットを特定する、特定の範囲の値を含む。

【0016】

様々な命令セットから命令をデコードするには、装置は第1命令セットのプログラム命令ワードをデコードするための第1命令デコーダと、第2命令セットのプログラム命令ワードをデコードするための第2命令デコーダを含み、制御手段が現在のプログラム命令ワードをデコードするのに、第1命令デコーダまたは第2命令デコーダのいずれかを制御するように作動できることが好ましい。

40

【0017】

第1命令セットのプログラム命令ワードは、Xビットのプログラム命令ワードであり、第2の命令セットのプログラム命令ワードは、Yビットのプログラム命令ワードであり、YはXと異なることが好ましい。このように共通のプロセッサコアが、より長いプログラム命令ワードを有し、潜在的によりパワフルな関連命令を可能とする命令セットか、或いはより短いプログラム命令ワードを有し、よって潜在的により限られた命令セットが許容さ

50

れる、メモリスペースをセーブする命令セット、のいずれかによりプログラムされ得る。

【0018】

本発明の実施例ではないが、プログラムカウンタレジスタの1つ以上のビットはプログラムカウンタレジスタの1つ以上の最上位ビットであり、例えば32ビットのプログラムカウンタレジスタでは、かかる大きなプログラムカウンタレジスタによってアクセスできる最大メモリスペースは、通常使用されるメモリスペースよりもかなり大きいので、最高位ビットが必要になることはほとんどない。

【0019】

上記と異なり、本発明の実施例では、プログラムカウンタレジスタの1つ以上のビットはプログラムカウンタレジスタの1つ以上の最下位ビットである。この場合プログラム命令ワードまたはデータワードの最小長さが少なくとも2バイトであり、これらビットは多くの場合使用されない。

【0020】

アクセスされるデータメモリ内の無効アドレスを避けるため、データメモリに記憶されたプログラム命令ワードにアクセスするための手段を設けることが好ましく、このアクセス手段はプログラムカウンタレジスタの1つ以上のビットには応答しない。

【0021】

添付図面を参照して次の図示した実施例の詳細な説明を読めば、本発明の上記およびそれ以外の目的、特徴および利点が明らかとなる。

【0022】

【実施例】

図1は、プロセッサコア10がメモリスシステム20に結合されたデータ処理装置の略図である。

【0023】

このプロセッサコア10はレジスタバンク30と、ブースマルチプライヤ40と、バレルシフタ50と、32ビットの算術論理ユニット(ALU)60と、書き込みデータレジスタ70とを含む。プロセッサコア10とメモリスシステム20の間には、命令パイプライン80と、マルチプレクサ90と、第1命令デコーダ100と、第2命令デコーダ110と、読み出しデータレジスタ120とがある。

【0024】

プロセッサコア10の一部であるプログラムカウンタ(PC)は、メモリスシステム20のアドレス指定を表示するように示されている。プログラムカウンタコントローラ140は、各々の命令が実行され、命令パイプライン80のための新しい命令をフェッチする必要があるたびに、プログラムカウンタレジスタ130内のプログラムカウンタ値をインクリメントするように働く。更に分岐命令が実行されると分岐命令のターゲットアドレスがプログラムカウンタコントローラ140によりプログラムカウンタ130内にロードされる。

【0025】

プロセッサコア10は、種々の機能ユニットの間の32ビットのデータ通路を含む。作動中、命令パイプライン80内の命令は、(マルチプレクサ90の制御により)第1命令デコーダ100または第2命令デコーダ110のいずれかによりデコードされ、プロセッサコア10の個々の機能素子へ送られる種々のコア制御信号を発生する。これらコア制御信号に応答し、プロセッサコアの異なる部分は32ビットの処理演算、例えば32ビットの乗算、32ビットの加算および32ビットの論理演算を実行する。

【0026】

レジスタバンク30はカレントプログラミングステータスのレジスタ(CPSR)150と、セーブされたプログラミングステータスのレジスタ(SPSR)160を含む。カレントプログラミングステータスレジスタ160はプロセッサコア10のための種々の条件およびステータスフラグをホールドする。これらフラグは処理モードフラグ(例えばシステムモード、ユーザーモード、メモリアポートモード等のフラグ)のみならず、算術演算

10

20

30

40

50

におけるゼロの結果、桁上げ等の発生を表示するフラグを含むことができる。セーブプログラミングステータスレジスタ160（これはバンク状の複数のかかるセーブプログラミングステータスレジスタのうちの一つでよい）は、処理モードスイッチの切り替えをトリガーする例外条件が発生した場合、現在プログラミングステータスレジスタ150の内容を一時的に記憶するのに使用される。

**【0027】**

プログラムカウンタレジスタ130は命令セットフラグTを含む。この命令セットフラグはマルチプレクサ90の作動を制御するのに使用され、よって現在データ処理命令をデコードするのに第1命令デコーダ100または第2命令デコーダ110のいずれを使用するかを制御するのに使用される。本実施例では、2つの命令セット、すなわち第1命令セットと第2命令セットとを示す。第1命令セットは32ビットのプログラム命令ワードから成り、第1命令デコーダ110によってデコードされ、第2命令セットは16ビットのプログラム命令ワードを含み、第2命令デコーダ110によってデコードされる。第1命令デコーダ100と、第2命令デコーダ110によって発生されるコア制御信号は、コア10の種々の機能ユニットとコンパチブルとなっている。

10

**【0028】**

プログラム命令ワード長さが異なる1つの命令セットを使用することにより、より長いワードを有し、潜在的によりパワフルな関連する命令を可能とする第1命令セットまたはより短いプログラム命令ワードを有し、このため、潜在的により限られた命令セットを許容できるメモリスペースをセーブできる第2命令セットのいずれかにより共通処理コア10にプログラムすることが可能となっている。

20

**【0029】**

命令セットフラグTを設けたことにより第1命令セットに対し第2命令セットを非直交状態にできる。これは特に別の直交命令セットを提出し、デコードできるようにするのに用いられるフリービットを用いることなく、第1命令セットが現在の命令セットである場合に有効である。

**【0030】**

命令セットフラグTはプログラムカウンタレジスタの通常は使用されないビット内に隠されている。このことは、Tフラグはプログラムカウンタコントローラ140によりセットまたはリセットできるが、Tフラグのステートはメモリスシステム20および命令パイプライン80の作動に直接影響する必要がないことを意味している。

30

**【0031】**

図2および図3はTビットをプログラムカウンタレジスタ内にエンコードできる2つの可能な方法を示すプログラムカウンタレジスタの略図である。これら2つの方法ではプログラムカウンタレジスタの通常使用されない高い桁（最高位）ビットとして、またはプログラムカウンタレジスタの通常使用されない低い桁（最下位）のビットとしてTビットをエンコードする。

**【0032】**

図2はTビットがプログラムカウンタレジスタの最高の桁のビットとしてエンコードされているプログラムカウンタレジスタ130'の略図である。

40

プログラムカウンタレジスタは32ビットレジスタであり、これにより $2^{32}$ バイトをメモリスシステム20内でアドレス指定できる。しかしながらこのことは、4ギガバイトのアドレス指定可能なメモリスペースに等しいので、21ビットプログラムカウンタレジスタにより可能とされるフルアドレスレンジが必要となることは、ほとんど可能性がない。

**【0033】**

従って図2におけるTビットはプログラムカウンタレジスタ130'の最高の桁のビットとしてエンコードされる。これによっても2ギガバイトのメモリをアドレス指定できるが、実際には通常この数よりも少ない数のメモリがアドレス指定され、プログラムカウンタレジスタのうちの上位ビットは（図2に示すようにゼロにできる）。

**【0034】**

50

解決すべき問題は、Tビットをセットした時、プログラムカウンタレジスタ130'はメモリスistem20のアドレスレンジをはるかに越えるメモリアドレスをポイントする可能性があることである。換言すれば、プログラムカウンタレンジ130の32ビットによりポイントされるメモリアドレスはメモリスistem20に関する限り無効アドレスである。

【0035】

この問題は2つの直接的方法によって解決できる。1つの技術ではプログラムカウンタレジスタ130'の最上位のビット(Tビット)を、メモリスistem20の単なるアドレスビットとしては供給しない。これとは異なり、メモリスistem20内のアドレスデコーディングが所定の数の最小の桁のビット(すなわち16メガバイトのアドレススペースをアドレス指定するのに最小位の24ビット)だけを検出してもよい。この場合、残りの、より高い桁のビットのステートは、デコードされるアドレスに対応する。この方法は、ある数のアドレスビットしか必要でないことが予め判っている時のメモリアドレスデコーディングの標準的技術となっている。

10

【0036】

先に述べたように、Tビットがプログラムカウンタレジスタ130'からマルチプレクサ90へ送られ、このTビットは第1命令デコーダ100または第2命令デコーダ110のいずれかへの命令のルーチングを決定する。

【0037】

図3は、プログラムカウンタレジスタの最下位ビットとして命令セットスイッチング用Tビットをエンコードする第2プログラムカウンタレジスタ130"の略図である。

20

このプログラムカウンタレジスタの最下位ビットは、最小命令すなわちデータワードサイズが少なくとも2バイト(この場合16ビット)であるプロセッサ内では、通常、使用されない。従って本実施例では命令プログラムワードは32ビット長(4バイト)または16ビット長(2バイト)のいずれかとなり得るので、プログラムカウンタ130からメモリスistem20へ供給されるアドレスは常に2の倍数となり、よって、アドレスの最下位ビットはゼロとなる。

【0038】

プログラムカウンタレジスタ130"の最下位ビットは、Tビットを記憶するのに使用され、このTビットは上記のようにマルチプレクサ90へ供給される。上記のように、プログラムカウンタレジスタ130"の最下位ビットは、メモリスistem20により無効アドレスにアクセスされないように、メモリスistemへは供給されない。

30

【0039】

プログラムカウンタ130はプログラムカウンタコントローラ130によって制御されるという事実によりコア10によって実行される分岐命令の一部としてTビットをセットできることを意味している。例えばTビットが現在第1(32ビット)の命令セットの使用を表示するようにセットされており、第2(16ビット)の命令セットを使用するコードの一部への分岐を望んでいる場合、実行すべき16ビットコードへジャンプし、同時にプログラムカウンタレジスタ内のTビットを変更するよう、分岐命令を実行できる。特にプログラムカウンタレジスタ130'の最高位ビットとしてTビットをエンコードする、図2に示される回路では、分岐命令は(ターゲットアドレスプラス10000000000000000000000000000000)への分岐によりTビットを1にセットできる。これとは異なり、図3のプログラムカウンタレジスタ130"内でTビットを1にセットするには、分岐命令は(ターゲットアドレスプラス1)への分岐の形態をとることができる。再びTビットを0に変えるのに、同様な回路を使用できる。

40

【0040】

図4に、このプロセスを略図で示す。図4は、図3のプログラムカウンタレジスタ130"を用いた32ビット命令セットと16ビット命令セットとの切り替えを示すフローチャートである。図4ではTビットが1にセットされている時は、このことは16ビットの命令セットを使用すべきことを意味している。

【0041】

50

図4を参照すると、32ビット命令セットで処理を開始する(200)。種々のデータ処理演算(210)の後にアドレスBadd(1)+1へ分岐するように分岐命令210を実行する。アドレスBadd(1)は16ビット命令セットを使ったコードの一部のスタートアドレスであり、付け加えられる+1は、Tビットのコードを使用すべきことを表示するTビットの切り替えに使用される。ターゲットアドレスBadd(1)では、16ビット命令セットを用いて種々のデータ処理演算230が実行される。次に32ビット命令セットに戻るように、分岐命令240が実行される。特に分岐命令240は32ビットコードの一部を参照するターゲットアドレスBadd(2)であり、このアドレスにはTビットをゼロ状態へ戻すようにゼロが加算される。ターゲットアドレスBadd(2)では、種々のデータ処理演算210が実行され、処理が終了する(260)。

10

【0042】

プログラムカウンタ130内のTビットを変えることにより、2つの命令セット間の切り替えが行われると、マルチプレクサ90による実際の切り替えが遅延され、現在、パイプライン80内に存在している命令が可能となる。

【0043】

要約すれば、異なる処理モード(特に異なる命令セットの使用)の切り替えは、分岐命令の実行の一部としてプログラムカウンタにターゲットアドレスおよびモードフラグ(T)を書き込むことによって行うことができる。

【0044】

現行のプロセッサで第1命令セットを予め定義し、使用する別のケースでは、プログラムカウンタレジスタ130の通常使用されないビットが、命令セットによって変えられることがないように、現行の第1命令セットにおいて論理的制限があり得る。第2の別の命令セットを含むプロセッサの後方コンパチビリティのために、第1の(現行の)命令セットから第2の命令セットへ1方向に切り替えるよう、短い命令セット選択シーケンスのコードを使用しなければならないこともある。切り替え機構が追加されると同時に、一般に第2命令セットも追加されるので、プログラムカウンタレジスタ130の通常使用されないビットへのアクセスを限定することなく、第2命令セットを定義できる。このことは、第2命令セットから第1命令セットへ切り替えるのに、上記分岐機構を使用できることを意味している。

20

【0045】

命令セット選択シーケンス(ベニアとして知られている)の一例は次の通りである。

30

【0046】

【表1】

|         |         |
|---------|---------|
| ラベル_ベニア |         |
| XOR     | (PC, 1) |
| 分岐      | ラベル     |

【0047】

このルーチンでは、図3のプログラムカウンタレジスタ130"の現在の内容を1と排他的OR演算し、Tビットを1にセットする。(これとは異なり、図2のプログラムカウンタ130"を用いて現在の内容を10000000000000000000000000000000と排他的OR演算し、Tビットをセットすることもできる。)

40

【0048】

別のベニアルーチンでは、排他的OR演算の代わりに減算演算を用い、プログラムカウンタレジスタ130"のTビットを変えることができる。この方法は、あるプロセッサでは減算演算により命令パイプライン80をフラッシュすなわちクリアできるという利点もある。

【0049】

次の例は、プログラムカウンタ130"が現在の命令を8ビットだけ越えてポイントし、

50



現在の命令が32ビット(4バイト)の命令であると仮定したものである。従ってプログラムカウンタレジスタ130"の最下位ビットを1に変えるには、現在プログラムカウンタレジスタの内容に対して次の値を加減する必要がある。

【0050】

【表2】

|      |                                     |
|------|-------------------------------------|
| 1を加算 | (Tビットを1へ変更するため)                     |
| 8を減算 | (カレント命令の前をポイントしているプログラムカウンタを補償するため) |
| 4を加算 | (カレント命令の長さを補償するため)                  |
| ...  |                                     |
| 3を減算 | (全体の変更)                             |
| ...  |                                     |

10

【0051】

従って使用される命令シーケンスは次の通りとなる。

【0052】

【表3】

ラベル\_ベニア

SUB (PC, PC, 3) (PCをPC-3と置換)  
分岐 ラベル

20

【0053】

要約すれば、命令セットしてビットを記憶するためにプログラムカウンタを使用することには、少なくとも次のような利点がある。

1. ターゲットアドレスと単一マシンワード内の対応する命令セットの双方を表示することにより、ターゲットルーチンを識別する1つの均一な方法が得られる。
2. より少ない数のシーケンスで足りるので、コードサイズが小さくなる。
3. 各命令内セットルーチンの呼び出しごとにシーケンスを実行することが不要となるので、プロセッサの性能を向上させることができる。

【0054】

30

以上で添付図面を参照して本発明の図示した実施例について説明したが、本発明はこれら実施例のみに限定されるものでなく、当業者であれば特許請求の範囲に記載した発明の要旨から逸脱することなく、種々の変形および変更が可能であると理解すべきである。

【図面の簡単な説明】

【図1】プロセッサコアおよびメモリシステムを有するデータ処理装置の略図である。

【図2】プログラムカウンタレジスタの略図である。

【図3】プログラムカウンタレジスタの略図である。

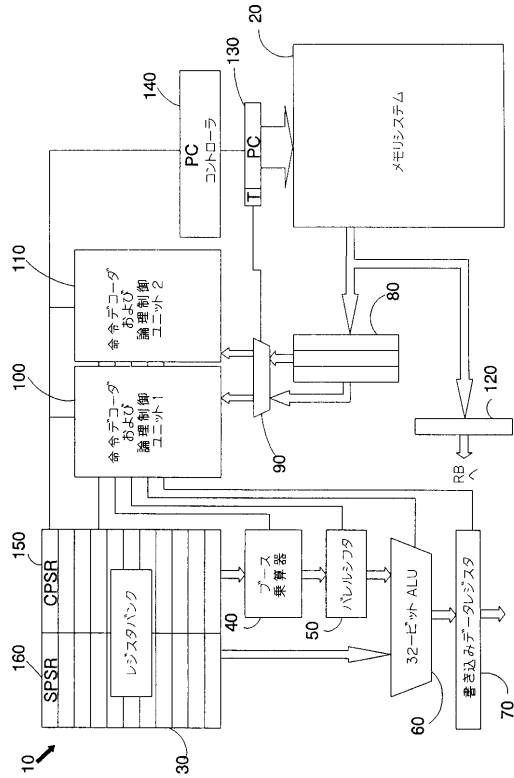
【図4】図3のプログラムカウンタレジスタを用いた2つの命令セットの切り替えを示す略フローチャートである。

【符号の説明】

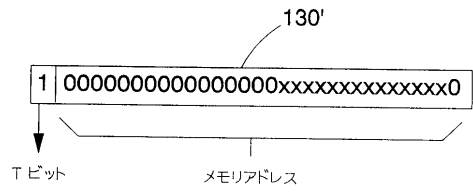
40

- |         |               |
|---------|---------------|
| 10      | レジスタバンク       |
| 20      | メモリシステム       |
| 40      | ブース乗算器        |
| 50      | バレルシフタ        |
| 60      | 32ビットALU      |
| 70      | 書き込みデータレジスタ   |
| 100、110 | 命令デコーダ        |
| 130     | プログラムカウンタレジスタ |

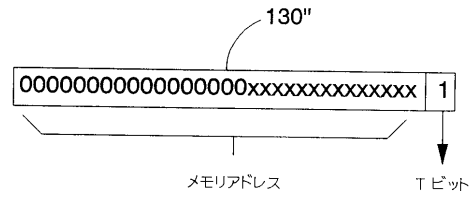
【 図 1 】



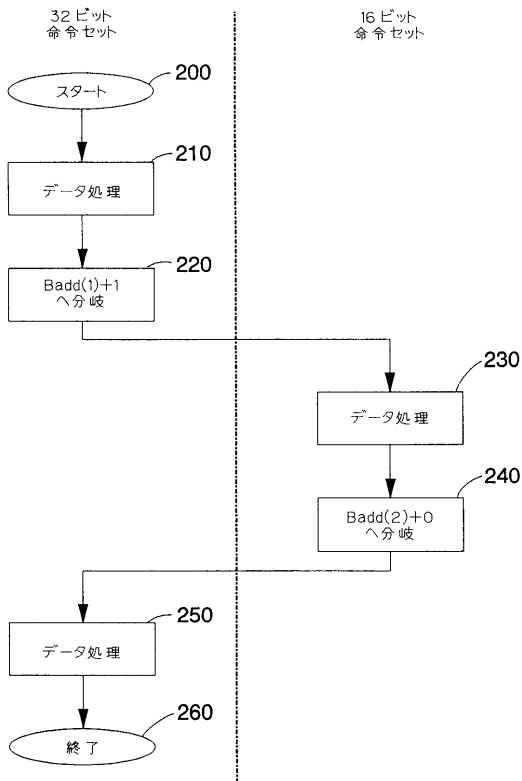
【 図 2 】



【 図 3 】



【 図 4 】



---

フロントページの続き

(72)発明者 エドワード コレス ネビル  
イギリス国ケンブリッジ, ウォータービーチ, ステーション ロード 51

合議体

審判長 井関 守三

審判官 堀江 義隆

審判官 山中 実

(56)参考文献 特開昭63-111533(JP,A)  
特開平4-287135(JP,A)  
前田英明, 「32ビットマイクロプロセッサ80386その特徴と性能」, bit, 共立出版,  
1986年5月1日, 通巻230号, P.14~22

(58)調査した分野(Int.Cl.<sup>7</sup>, DB名)  
G06F 9/30