

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4065660号
(P4065660)

(45) 発行日 平成20年3月26日 (2008.3.26)

(24) 登録日 平成20年1月11日 (2008.1.11)

(51) Int.Cl.

F I

G O 6 F 12/08 (2006.01)

G O 6 F 12/10 (2006.01)

G O 6 F 12/08 5 2 5 C

G O 6 F 12/08 5 0 7 Z

G O 6 F 12/08 5 1 9 Z

G O 6 F 12/08 5 0 9 Z

G O 6 F 12/10 5 0 1 E

請求項の数 3 (全 13 頁)

(21) 出願番号 特願2000-379986 (P2000-379986)
 (22) 出願日 平成12年12月14日 (2000.12.14)
 (65) 公開番号 特開2001-195303 (P2001-195303A)
 (43) 公開日 平成13年7月19日 (2001.7.19)
 審査請求日 平成16年10月26日 (2004.10.26)
 (31) 優先権主張番号 09/466494
 (32) 優先日 平成11年12月17日 (1999.12.17)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 398038580
 ヒューレット・パカード・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 100087642
 弁理士 古谷 聡
 (74) 代理人 100063897
 弁理士 古谷 馨
 (74) 代理人 100076680
 弁理士 溝部 孝彦

最終頁に続く

(54) 【発明の名称】 機能が並列に分散された変換索引バッファ

(57) 【特許請求の範囲】

【請求項 1】

コンピュータマイクロアーキテクチャであって、
整数ロードデータを受信するように適合された整数ロード変換索引バッファ (TLB)
 と、及び

前記整数ロード TLBと並列に接続され、前記整数ロードデータを受信し、かつ前記整数ロード TLBにおける前記整数ロードデータの受信と並列に、整数ストアデータ、浮動小数点ロードまたはストアデータ、及び TLB 参照命令および更新命令のうちの 1 つ又は複数を受信するように適合された マスタ TLBとを含み、

前記整数ロード TLBが、第 1 および第 2 のメモリポートによってアクセスされ、前記 マスタ TLBが、前記第 1 および第 2 のメモリポート、ならびに第 3 および第 4 のメモリポートによってアクセスされる、コンピュータマイクロアーキテクチャ。

【請求項 2】

整数ロードのための仮想アドレスが供給される整数ロード TLBであって、当該仮想アドレスに対応する仮想タグの当該整数ロード TLB における位置を示す仮想 TLB ヒットを提供する 整数ロード TLBと、

データセクションとタグセクションを有する整数ロードデータキャッシュであって、当該データセクションに格納されているキャッシュデータの、物理タグの、前記整数ロード TLB における位置を示す物理 TLB ヒットを当該タグセクションに保持し、前記整数ロード TLB から提供される前記仮想 TLB ヒットと前記タグセクションに保持されている

10

20

前記物理 T L B ヒットとを用いてヒット判定を行う整数ロードデータキャッシュと、

前記整数ロード T L B と並列に動作し、ロード及びストアのための仮想アドレスが供給されるマスタ T L B であって、当該仮想アドレスに対応する仮想タグのための物理タグを提供するマスタ T L B と、

データセクションとタグセクションを有するデータキャッシュであって、当該データセクションに格納されているキャッシュデータの物理タグを当該タグセクションに保持し、前記マスタ T L B から提供される物理タグと当該タグセクションに保持されている物理タグとを用いてヒット判定を行うデータキャッシュを含む、装置。

【請求項 3】

前記整数ロード T L B が第 1 のメモリポートによってアクセスされ、前記マスタ T L B が前記第 1 のメモリポート、ならびに第 2 のメモリポートによってアクセスされる、請求項 2 に記載の装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明の属する技術分野には、変換索引バッファを事前検査済み(prevalidated)キャッシュタグ設計に使用するコンピュータシステムが含まれる。

【0002】

【従来の技術】

コンピュータシステムは、マルチレベル階層のメモリシステムを使用する。そのマルチレベル階層のメモリシステムは、最上位レベルの階層には、比較的高速かつ高価であるが容量が制限されるメモリから、最下位レベルの階層には、比較的低速かつ低コストであるが容量の大きいメモリに至る階層構造を有する。階層は、物理的にプロセッサ内に一体化されるか、あるいは物理的にプロセッサに近接して取り付けられる、キャッシュと呼ばれる小型かつ高速のメモリを速度を得るために含むことができる。コンピュータシステムは、命令キャッシュとデータキャッシュを別個に採用することも可能である。さらに、コンピュータシステムは、マルチレベルのキャッシュを使用することも可能である。キャッシュの使用は、命令レベルにおいてコンピュータプログラムにとって透過的であるため、命令セットを変更せずに、または既存のプログラムに対する修正を必要とせずにコンピュータアーキテクチャに追加可能である。

【0003】

プロセッサがキャッシュからある項目を要求し、その項目がキャッシュに存在する場合、キャッシュヒットが発生する。一方、プロセッサがキャッシュからある項目を要求し、その項目がキャッシュに存在しない場合、キャッシュミスが発生する。キャッシュミスの場合、プロセッサは、要求された項目を、より低いレベルのメモリ階層から検索する。多くのプロセッサ設計において、キャッシュヒットのために、項目にアクセスするのに要する時間は、設計者が単一サイクルのキャッシュアクセス時間を求めている(seek)場合には、プロセッサのクロックレートの主要なリミッタの 1 つである。他の設計において、キャッシュアクセス時間は、複数サイクルとしてもよいが、プロセッサの性能は、サイクル単位でのキャッシュアクセス時間が短縮されるほとんどの場合には、改善可能である。したがって、キャッシュヒットのためにアクセス時間を最適化することは、コンピュータシステムの性能にとって重要である。

【0004】

キャッシュ設計には、仮想記憶装置の概念が結びつけられる。仮想記憶システムにより、コンピュータプログラムは、メモリを単一レベルの一定の 1 つの記憶ユニットとして考えることができるが、実際には、補助記憶装置と高速記憶装置(キャッシュ)との間で、ページ上のプログラムブロックをオンデマンドで自動的に移動させる動的アドレス変換ユニットが提供されている。

【0005】

また、キャッシュ設計には、フルアソシエイティブまたは内容指定メモリ(CAM: Cont

10

20

30

40

50

ent Addressable Memory) の概念も結びつけられる。内容指定メモリは、従来のハード的に組み込まれたアドレス指定機構を有することに加え、所望のビット位置との比較を可能にして、1つのメモリサイクル時間中にすべてのエントリが同時に指定のマッチングをされるようにハード的に組み込まれた論理も有するランダムアクセスメモリである。したがって、その内容の一部を使用することでエントリにアクセスできるため、所望のエントリの特定アドレスを知る必要はない。指定されたビット位置と一致するすべてのエントリにはフラグが立てられ、現在または後続のメモリサイクルにおいてアドレス指定が可能である。

【0006】

メモリは、ワードに編成されても良い(たとえば、1ワードあたり32ビットまたは64ビット)。キャッシュと次に下位レベルのメモリ階層との間で転送可能な最小メモリ量をラインまたはブロックという。1ラインは、複数のワードとしてもよい(たとえば、16ワード/ライン)。また、メモリをページまたはセグメントに分割して、各ページが多数のラインを有するようにしてもよい。コンピュータシステムによっては、ページサイズを可変にすることも可能である。

【0007】

今日のコンピュータメモリアーキテクチャにおいて、中央処理装置(CPU)は、ハードウェアとソフトウェアとの組合せにより物理アドレスに変換される仮想アドレスを生成する。物理アドレスは、物理的なメインメモリにアクセスするために使用される。一群の仮想アドレスを、各ページに動的に割り当てることが可能である。仮想メモリでは、仮想アドレスを物理アドレスに変換する、ページテーブルと呼ばれることもあるデータ構造が必要とされる。アドレス変換時間を短縮するために、コンピュータは、変換索引バッファ(TLB: Translation Lookaside Buffer)と呼ばれるアドレス位置専用の特殊な連想キャッシュを使用してもよい。

【0008】

キャッシュは、多数のセグメントまたはウェイ(way)を含むことができる。キャッシュがデータとともにラインアドレス全体を記憶し、キャッシュ内の任意の場所に任意のラインを配置することが可能な場合、そのキャッシュはフルアソシエイティブであるという。任意のラインが任意の場所に配置できる大容量キャッシュの場合、キャッシュに項目があるか否か、そしてどこにあるかを迅速に決定するために要するハードウェアは、極めて大型かつ高価になる可能性がある。キャッシュが大容量であるほど、高速かつ省スペースの代替物が、アドレスのサブセット(インデックスと呼ばれる)を用いてキャッシュ内のライン位置を指定し、次いでタグと呼ばれる各物理アドレスのより上位ビットの残りのセットをデータとともに記憶する。索引づけを行うキャッシュにおいて、特定アドレスを有する項目は、インデックス(索引)により指定されるラインのセット内にしか配置することができない。所定アドレスのインデックスがサブセットまたはウェイにおける1ラインに正確にマッピングするようにキャッシュが構成される場合、そのキャッシュは直接マッピング式であるといわれる。インデックスが、サブセットまたはウェイにおいて2ライン以上にマッピングする場合、そのキャッシュはセットアソシエイティブであるといわれる。アドレス空間をセットに分割するセットインデックスを提供するためにアドレスの一部または全部をハッシュしてもよい。

【0009】

直接マッピングでは、あるラインが要求されると、キャッシュの1ラインのみが、一致するインデックスビットを有する。したがって、アドレスの残りが一致するか否かをコンピュータシステムが決定する前に、データは直ちに検索され、データバスに送り込むことが可能である。データは、有効であっても無効であってもよいが、データが有効である通常の場合には、データビットは、コンピュータシステムが有効性を決定する前に、データバスにおいて使用可能である。セットアソシエイティブキャッシュでは、コンピュータシステムは、完全なアドレスを比較するまで、どのラインがアドレスに対応しているか分からない。すなわち、セットアソシエイティブキャッシュでは、タグの比較結果を使用して、

10

20

30

40

50

1 セットのラインにおけるどのラインのデータビットがプロセッサに供給されるかを選択する。

【 0 0 1 0 】

【発明が解決しようとする課題】

T L Bを有するキャッシュにおいて、ヒットのためのクリティカルタイミングパスは、以下の4つの動作のシーケンスを必要とする。すなわち、1) 仮想タグをT L BにおけるC A Mに提示して、T L Bのランダムアクセスメモリ(R A M)において対応する物理タグの位置を決定する必要がある、2) 次に、物理タグをT L Bランダムアクセスメモリから検索する必要がある、3) 次に、T L B R A Mからの物理タグを、キャッシュのタグセクションからアクセスした物理タグと比較する必要がある、そして、4) 適切なデータラ

10

【 0 0 1 1 】

【課題を解決するための手段】

事前検査済みタグキャッシュは、従来の慣例のようにアドレス情報ではなく、変換索引バッファ(T L B)からのヒット情報を保有する。すなわち、T L Bからの出力として物理アドレスが提供されるのではなく、仮想アドレスがT L Bに提供される場合、T L Bヒットビットが提供される。次に、T L Bヒットビットは、(T L Bヒット情報を保有している)事前検査済みキャッシュタグと比較され、キャッシュヒットが発生したか、またはキャッシュミスが発生したかを決定する。処理速度を高めるために、T L Bは、たとえば、

20

【 0 0 1 2 】

小型T L Bは、たとえば、キャッシュの性能を低下させるというスラッシング、すなわち、ページまたはT L Bエントリをスワップアウトするという、潜在的な問題を呈する。T L Bが小さいので、T L Bエントリが非常に素早く排除できるため、事実上、キャッシュからデータが損失する。すなわち、事前検査済みキャッシュタグは、T L Bのヒット情報を含み、物理アドレスを含まないため、T L Bエントリが変化すると、ヒット情報は有効ではなくなり、古い方のT L Bエントリと関連づけられたデータがいずれも失われる。

【 0 0 1 3 】

このマイクロアーキテクチャにおいて、T L Bエントリを小さいまま、T L Bのスラッシングを防ぐために、別個の大型T L Bを、ストア動作等、より長いレイテンシーで対応できるプロセス動作に追加してもよい。この大型T L Bは、たとえば浮動小数点演算に対して、より大きい帯域幅を提供することもできる。小型T L Bにおいて必要とされるT L Bエントリを、たとえば、整数ロードに必要とされるT L Bエントリに限定し、他のすべてのT L Bエントリを大型T L Bに転送してもよい。T L Bは、並列に動作する。さらに、すべての例外および特権の検査を大型T L Bにおいて実行してもよく、大型T L Bには、整数ロードアクセス時間を短縮する作用もある。この並列T L Bアーキテクチャを使用することで、小型および大型T L Bが直列にアクセスされると発生する可能性のあるパイプラインの機能停止を低減する利点がさらにある。

30

40

【 0 0 1 4 】

物理アドレスの代わりにT L Bヒット情報のみを有するキャッシュタグのさらなる結果として、キャッシュラインを更新および無効にする機構を設けてもよい。この機構において、小型T L Bの機構および整数ロードデータパスとは別にキャッシュラインを探索、更新および/または無効にする並列構造を設けることができる。さらに、状況によっては、データキャッシュに単なる物理アドレスが設けられる。適切なキャッシュラインを見つけるために、物理アドレスのみに基づいて、物理アドレスを有するタグと大型T L Bの周りをバイパスする物理アドレスとを含むコントロールが提供される。

【 0 0 1 5 】

【発明の実施の形態】

50

以下、図面を参照して並列分散機能変換索引バッファを説明する。図中、同様の符号は同様の要素を表す。

【0016】

物理タグをTLBから検索する必要のない、TLBを有するキャッシュは、TLBを有するキャッシュにアクセスするためにクリティカルパスの全体時間を改良できる。かかる設計において、物理タグをキャッシュに記憶する代わりに、キャッシュは、物理タグが記憶されるTLB内の位置を記憶する。TLBは、2つ以上のCAMを含むことができる。キャッシュヒットにおいて、TLBにおけるCAMの1つは、物理アドレスが記憶されるTLB内の位置を指定するベクトルを生成することができる。このベクトルを、キャッシュに記憶されている位置ベクトルと比較してもよい。位置ベクトルの比較により、物理アドレスを実際に検索することを必要とせずに、1セット内の1つのデータラインの選択を可能にするのに十分な情報が提供される。その結果、相当な時間を消費する動作（物理アドレス検索）が、キャッシュヒットのクリティカルタイムパスから排除される。さらに、物理タグではなく位置ベクトルを比較することで、従来のデジタル比較器に比べて高速かつ単純な比較論理の使用が可能になる。

【0017】

図1は、従来技術のキャッシュの一例を示す。システム10は、仮想アドレス12と、ランダムアクセスメモリアレイ14と、比較器16と、物理アドレスレジスタ18と、を含む。システム10は、セットアソシエイティブ論理を採用している。ランダムアクセスメモリアレイ14は、2個の仮想ページアドレスビットを必要とする合計128（4個）のエントリを含む。4個の各セットは、ランダムアクセスアレイの1物理ワード（水平方向）の部分であるため、かかるワードが128個あり、7個のアドレスインデックスビットが必要とされる。キャッシュページがあるか否か、そして、どこにあるかを決定するために、アドレス変換において、合計仮想ページ数アドレス $n = 9$ が使用される必要がある。下位ビット n は、ページ内のバイトを表すが、変換される必要はない。7個の仮想ビットは、128セットの1つを直接選択するために使用される。セットから読み出されたワードは、比較器16を使用して、同時に仮想アドレスと比較される。比較の1つが「YES」である場合、ランダムアクセスメモリアレイにある、キャッシュにおけるページの正しい実アドレスまたは物理アドレスは、物理キャッシュアドレスレジスタ18にゲートされる。この物理アドレスが次のサイクルで使用され、キャッシュアレイ（図示せず）から正しい情報を入手する。

【0018】

図2は、別の従来技術のキャッシュを示す。4ウェイのセットアソシエイティブキャッシュが説明のために用いられる。仮想アドレス100は、下位インデックスビット102と上位（仮想）タグビット104からなる。インデックスビット102は、一般に、仮想アドレスおよび物理アドレスについて同一である。インデックスビット102は、キャッシュのデータセクション106において、1セットのデータのラインを選択するために使用される。データセクション106の出力は4つのデータライン108である。また、インデックスビット102は、キャッシュのタグセクション110において1セットの物理タグを選択するためにも使用される。タグセクション110の出力は4つの物理タグ112であり、それぞれが1つのデータライン108に対応している。仮想タグビット104は、TLB114内のCAM116における1つのエントリを選択するために使用される。TLB114は、仮想タグと物理タグの両方を記憶する。仮想タグビット104の一致がCAM116にない場合、TLBミスが発生する。図2に示すシステムにおいて、複数の仮想タグを1つの物理タグにマッピングしてもよい。TLBヒットに関して、選択されたCAMエントリは、仮想タグ104に対応する物理タグについて、TLBRAM118におけるアドレスを指定する。そして、TLBRAM118から物理タグが検索される。4個のデジタル比較器120のそれぞれは、TLBRAM118からの物理タグを、タグセクション110からの物理タグ112と比較する。一致する一対の物理タグは、論理122を介して、データの4ラインのうちどのラインがマルチプレクサ124により

選択されるかを示す。特定のインデックスビットに関して、一致する一対の物理タグが存在しないこともあり、この場合、キャッシュミスが発生する。

【 0 0 1 9 】

図 3 は、4 ウエイのセットアソシエイティブキャッシュ 2 0 0 を有するコンピュータマイクロアーキテクチャを示す。キャッシュ 2 0 0 は、インデックスビット 2 0 2 と、データセクション 2 0 3 と、マルチプレクサ 2 0 5 と、を含む。キャッシュタグセクション 2 0 4 は、データラインに対応する物理 T L B ヒットタグを含む。物理アドレスタグがキャッシュタグセクション 2 0 4 に記憶されるのではなく、新たなデータのラインがキャッシュ 2 0 0 に配置されると、ベクトル 2 1 2 (物理 T L B ヒットベクトルと呼ばれる) がキャッシュタグセクション 2 0 4 に記憶される。

10

【 0 0 2 0 】

事前検査済みキャッシュアーキテクチャに関するさらなる詳細については、1997 年 1 0 月 2 2 日付けで出願された「CACHE MEMORY WITH REDUCED ACCESS TIME」と題する同時係属中の米国特許出願第 0 8 / 9 5 5 , 8 2 1 号に提示されている。その開示は、参照することにより、本明細書に組み込まれる。

【 0 0 2 1 】

キャッシュ 2 0 0 において、T L B 2 1 0 は、物理タグを含む物理 C A M 2 0 6、および仮想タグを含む仮想 C A M 2 0 8 の 2 つの C A M を有する。新たな仮想タグ 2 0 7 が仮想 C A M 2 0 8 に記憶されると、対応する物理タグ 2 0 9 もコンピュータオペレーティングシステムを用いて入手可能となり、この対応する物理タグ 2 0 9 が物理 C A M 2 0 6 に記憶される。物理 T L B ヒットベクトル 2 1 2 は、物理タグ 2 0 9 を有する物理 C A M 2 0 6 における各位置に対応するバイナリ「1」を有する。新たなラインをキャッシュ 2 0 0 に入力すると、この新たなラインの物理タグ 2 0 9 の物理 C A M 2 0 6 におけるすべてのインスタンスの位置を示す物理 T L B ヒットベクトル 2 1 2 が、物理 C A M 2 0 6 により生成され、インデックスビット 2 0 2 により決定される行位置およびセット配置アルゴリズムにより決定される列位置において、キャッシュタグセクション 2 0 4 へ記憶される。

20

【 0 0 2 2 】

キャッシュアクセスに関して、仮想タグ 2 0 7 は、仮想 C A M 2 0 8 により使用され、仮想 T L B ヒットベクトル 2 1 4 を生成する。T L B ミスがある場合、仮想 T L B ヒットベクトル 2 1 4 は、すべてバイナリ「0」になる。T L B ヒットがある場合、仮想 T L B ヒットベクトル 2 1 4 は、仮想 C A M 2 0 8 における仮想タグ 2 0 7 の位置を示す単一のバイナリ「1」を有する。T L B 2 1 0 における各仮想タグ 2 0 7 は、一意でなければならない。

30

【 0 0 2 3 】

キャッシュアクセスに関して、インデックスビット 2 0 2 は、キャッシュタグセクション 2 0 4 において、4 個の物理 T L B ヒットベクトル 2 1 2 の 1 セットを選択する。キャッシュタグセクション 2 0 4 における 4 個の物理 T L B ヒットベクトル 2 1 2 のそれぞれは、4 個の比較器 2 1 6 の 1 個を使用して、仮想 C A M 2 0 8 からの仮想 T L B ヒットベクトル 2 1 4 と比較される。任意の所定セットのインデックスビット 2 0 2 に関して、キャッシュタグセクション 2 0 4 における 4 個の選択された物理タグの 1 個だけが、T L B 2 1 0 からの仮想 T L B ヒットベクトル 2 1 4 と固定ページサイズに関して一致する。固定ページサイズに関して、4 個の物理 T L B ヒットベクトル 2 1 2 における単一对をなす一致する「1」は、次に、どのデータラインがマルチプレクサ 2 0 5 により選択されるかを決定する。所定セットのインデックスビット 2 0 2 に関して、比較された 4 個の物理 T L B ヒットベクトル 2 1 2 において一致する「1」がない場合、キャッシュミスが発生する。

40

【 0 0 2 4 】

キャッシュ 2 0 0 において、T L B 2 1 0 からの物理タグは、キャッシュアクセスのために検索されない。T L B 2 1 0 からの物理タグの検索動作を無くすことにより、キャッシュアクセスのクリティカルタイムパスの時間が実質的に短縮される。キャッシュ 2 0 0 は

50

一対の一致する論理「1」を検索して一致を決定するため、比較器216は、単純なANDゲートに続いて大きいファンインのORゲートとしてもよい。

【0025】

図3に示すマイクロアーキテクチャは、事前検査済みタグキャッシュを含む。事前検査は、コンピュータマイクロアーキテクチャ設計者がキャッシュロードのレイテンシーを最小にしつつ全帯域幅を最大にしようとする場合にマイクロアーキテクチャにおけるTLBがどのように動作するかについて制限を加える。たとえば、事前検査済みタグキャッシュは、特定のロードについて極めて高速なアクセス時間を提供するが、マイクロアーキテクチャ設計者は、仮想アドレスと物理アドレスとの間の変換を制限し、かつ異なるキャッシュレベルの中での処理の分散を制限するように選択してもよい。マイクロアーキテクチャは、たとえば、高速整数ロードおよび大きい帯域幅の浮動小数点ロードを提供するようにしてもよい。すなわち、整数ロードデータは、高速アクセス時間を有する必要があるが、その動作設定サイズは一般的に小さい。整数ロードのレイテンシーを最適化するために、プロセッサの中には、小型だが高速の第1のレベルキャッシュを提供するものもある。仮想アドレス変換を提供し、かつアドレスエイリアシング問題を回避するために、プロセッサの中にはTLBにアクセスして、データがキャッシュに存在しているか否かを決定するためにキャッシュタグによる確認を行うための物理アドレスを提供する必要があるものもある。高速整数データアクセスの目的で、メモリのレイテンシーを減らすために、TLBサイズを最小数のエントリ(16~32等)に制限してもよい。これは、256以上のTLBエントリを必要とする可能性がある大型キャッシュ構造を有するプロセッサ上で必要とされる多数のエントリと競合する。

【0026】

図3に示すような事前検査済みキャッシュタグシステムにおいて、TLBエントリは、キャッシュラインを識別するためにキャッシュタグにおいて論理的に使用される。TLBエントリが除去されると、通常、コントロールを使用して、除去されたTLBエントリと関連づけられた事前検査済みキャッシュタグにおけるデータをすべて無効にする。しかしながら、1つのTLBエントリをデータキャッシュの大部分またはすべてにマッピングできるため、このアクションは、処理を遅らせる可能性がある。従ってTLBは、プログラム実行を支援するのではなく、メモリへ及びメモリからページを連続的にスワッピング(すなわち、スラッシング)することができる。

【0027】

浮動小数点データ処理の性能は、通常、浮動小数点実行ユニットへの及びそれからのメモリ帯域幅により制限される。短いレイテンシーを必要とする整数ロードデータアクセスとは対照的に、通常、浮動小数点アクセスはスケジュールされることができ、潜在期間(latency period)を長くすることができる。同様に、整数データサイズは通常小さいが、浮動小数点データセットは、通常、非常に大きい。理想的には、浮動小数点ロード/ストア動作のTLB動作は、大きい帯域幅かつ大型データスペースの変換(アクセスされた大量のTLBエントリ)の両方を提供する。1つの設計は、すべてのメモリポートに全帯域幅を提供するとともに、浮動小数点要求の変換については大型であるが遅いTLBを提供する。さらに、メモリポートの使用は、制限されなくてもよい。より多くのロードとストアの組合せを可能にする。

【0028】

事前検査済み整数ロードデータキャッシュへの記憶、またはこのデータキャッシュの無効化は、キャッシュラインの物理アドレスがキャッシュタグにおいて使用可能でないため困難であるが、これは、キャッシュタグが事前検査済みTLBヒット情報を保有しているにすぎないことによる。さらに、TLBが、高速のレイテンシーの必要性により小さい(たとえば、16または32エントリ)ため、小型の整数ロードデータキャッシュTLBはストア変換に使用されない。1つのオプションは、より大型かつ低速のTLBへのストアTLBアクセスを分離して、(浮動小数点ストア等)第1レベルのキャッシュにおいて実行されないおそれがあるストア動作、およびフラッシュキャッシュ、バススヌープ等の他の

10

20

30

40

50

無効化動作に対して、事前検査済み整数ロードキャッシュを無効化するための機構を提供することである。

【0029】

浮動小数点ロードおよびストア、パージ、挿入およびプローブを含むTLB支援命令、整数ストア等の多数のコンピュータ命令は、高速整数データキャッシュに即座にアクセスする必要がない。コンピュータ命令の一部は、より小型キャッシュのスラッシングを回避するために、高速整数ロードキャッシュにあるデータを有するものでなくてもよい。高速整数データキャッシュのより良好な使用を支援するため、このタイプのデータが整数データキャッシュをロードしないようにする機構を設けてもよい。これらの命令が整数キャッシュをバイパスできるために、すべての例外情報が大型TLBにのみ記憶され、大型TLBが、たとえば、すべてのストア動作、浮動小数点ロード、またはTLB参照（読み出し）と更新命令を含むTLB支援命令でのみアクセスされる必要があるようにしてもよい。

10

【0030】

最後に、すべての命令を第1レベルのTLB（高速にするためには小型であることが必要とされる）に通過させることで、TLBミス率が高くなるため、プロセッサにおいてパイプライン機能停止が生じることがある。したがって、図3に示すマイクロアーキテクチャは、TLBミス率およびパイプライン機能停止を低減する並列TLB構造を使用するように適合されても良い。

【0031】

図4は、並列分散機能TLB構造を採用するコンピュータマイクロアーキテクチャ250を示す。コンピュータマイクロアーキテクチャ250は、整数ロード命令ではない命令によって必要とされるエントリにより小型TLBをスラッシュしないことによって高速整数ロードデータアクセスに対し、小型TLBを効率よく使用できるようにする。マイクロアーキテクチャ250は、小型TLBにおけるスラッシングを減少させることによって事前検査済みキャッシュタグ設計のキャッシュデータ無効化を減少させ、減少したメモリポートの帯域を提供して、小型（整数ロードデータ）TLBの設計をより小型かつ高速にする。マイクロアーキテクチャ250は、より大型キャッシュにある浮動小数点アプリケーション等大型データ構造に対処するためにより大型のTLBを提供し、かつ、浮動小数点アプリケーションの大きい帯域幅の必要性に対処するために大きい帯域幅のTLBを提供する。また、マイクロアーキテクチャ250は、高速整数データキャッシュを中断（disrupt）することなくストアおよび無効化の要求変換を処理する方法も提供する。最後に、マイクロアーキテクチャ250は、アドレス指定するための例外および保護コントロールに単一の位置を提供するが、これは、整数ロードデータキャッシュにおいて、小型TLBとは別のものである。

20

30

【0032】

コンピュータマイクロアーキテクチャ250は、整数ロードデータTLB（L0TLB）252と、マスタTLB（DTLB）254とからなる並列TLBを含む。また、マイクロアーキテクチャ250は、整数ロードデータキャッシュ（L0D）264、L1データキャッシュ266、および並列物理タグ検索を有するストア/無効化コントロール262も含む。図4に示すように、TLBは、4個のメモリポート0～3を介してアクセスされる。しかしながら、マイクロアーキテクチャ250は、この構成に制限されるものではなく、任意の数のメモリポートがTLBにアクセスできる。

40

【0033】

L0TLB252は、整数ロードデータに短いレイテンシーを提供するように設計される。これを達成するために、L0TLB252にはいくつかの制限が加えられる。図4に示す例では、4個のメモリポートの2個のみがL0TLB252にアクセスすることを許されて、L0TLB252のサイズおよび複雑性を低減させ、従ってその速度が増加する。この低減は、整数ロードを4個のメモリポートの2個にのみ割り当てることによって可能である。さらに、整数ストア命令は、L0TLB252にアクセスしないが、DTLB254において処理される。この構成は、マイクロアーキテクチャ250がL0TLB25

50

2においてロード専用TLBエントリを割り付けることを可能とし、TLBエントリのスラッシングを低減させ、事前検査済みタグキャッシュ設計においてはキャッシュラインを無効化するという副作用を有する。

【0034】

次に、浮動小数点ロードおよびストアならびにその他の様々なTLBアクセス命令は、L0TLB252にアクセスしなくてもよく、L0TLB252におけるTLBエントリのスラッシングを低減させる。浮動小数点アクセスは、L1データキャッシュ266により処理され、DTLB254を介して変換される。

【0035】

例外および保護の情報は、L0TLB252に含まれないため、より小型でより高速の設計が可能になる。整数ロード命令は、例外および保護情報がDTLB254にのみ存在することから、L0TLB252とDTLB254の両方において、処理されるべき有効キャッシュヒットのエントリを有する。最後に、すべての整数ロード命令を検索してキャッシュヒットを決定するためのパスが、L0TLB252からL0D264まで提供される。

10

【0036】

DTLB254は、大量のTLBエントリを提供し、かつL0TLB252の場合より長いレイテンシーを有する大きい帯域幅を提供するように設計される。DTLB254は、すべてのTLBアクセスにパラレルポートを提供して、1サイクルにつき4つのメモリ動作の実行を支援するための大きい帯域幅を提供する。しかしながら、上述したように、マイクロアーキテクチャ250は、4個のメモリポートに制限されるものではない。また、DTLB254は、各メモリポートがロードまたはストアをサービスできるように汎用メモリポートをさらに提供する。L0D264は、1サイクルにつき2個のロードまたは2個のストアに制限され、高速のレイテンシーを提供するので、DTLB254は、1サイクルあたり最大4個の浮動小数点ロードまたは最大4個の浮動小数点ストアの実行を可能にする。

20

【0037】

DTLB254は、すべての4個のメモリポートに例外および保護の情報を提供する。これにより、この情報を必要とすることなく、L0TLB252がより小型かつ高速になるようにL0TLB252を構成することができる。DTLB254は、整数データパスに対する変換パスを、ストアアドレス更新およびキャッシュライン無効化機能に提供する。最後に、DTLB254は、L1データキャッシュ266に対するパスを提供し、すべての浮動小数点ロードおよびストア動作についてキャッシュ検索を実行する。

30

【0038】

マイクロアーキテクチャ250において、整数ストア更新アドレスおよび無効化アドレスがL0D264に提供されてもよい。コントロール262により、マイクロアーキテクチャ250は、キャッシュラインの物理アドレスを、事前検査済みキャッシュタグエントリと並列で記憶することができる。さらに、パス261がDTLB254からストア更新およびストア無効化の整数データパスまで、提供される。DTLB254アドレスは、物理アドレスキャッシュタグ構造におけるあらゆる方式の連想性(associativity)と比較され、どの方式の連想性が現在のストアまたは無効化要求に一致するかを検出する。コントロール262により、事前検査済みタグキャッシュL0D264の有効状態が無効化要求に応じて更新され、整数ストア動作に応じて整数ロードデータキャッシュを更新する。最後に、例外情報モジュール268は、4個のメモリポート0~3すべてについてDTLB254からの例外情報を受信する。

40

【0039】

以上、例示および説明の目的で、機能が並列に分散された変換索引バッファについて説明してきた。本記載は、これをもってすべてとするものでもなければ、本発明を開示される厳密な形式に制限するものでもなく、上記の教示に照らして他の修正および変形も可能である。本記載は、機能が並列に分散された変換索引バッファおよびその実用的な適用の原

50

理を説明することで、当業者が、企図された特定用途に適合される各種実施態様および修正案において最もよく使用できるようにしたものである。

【 0 0 4 0 】

本発明を要約すると次のようになる。コンピュータシステムにおいて、機能が並列に分散された変換索引バッファ（ＴＬＢ）は、小型で高速のＴＬＢ（252）及び大型であるがより低速の別のＴＬＢ（254）を含む。2つのＴＬＢは並列に動作する。小型のＴＬＢ（252）は整数ロードデータを受信し、大型のＴＬＢ（254）は他の仮想アドレス情報を受信する。2つのＴＬＢ間において、ロードとストア命令、整数と浮動小数点命令等の機能を分散することにより、小型のＴＬＢは短いレイテンシーで動作でき、スラッシング及び類似の問題を回避する一方、大型のＴＬＢはメモリの集約的な動作に対して、大きい帯域幅を提供する。この機構は、事前検査済みキャッシュタグ設計に特に有用である並列ストア更新および無効化機構（262）も提供する。

10

【 0 0 4 1 】

以下においては、本発明の種々の構成要件の組み合わせからなる例示的な実施態様を示す。

1．第1のデータを受信するように適合される第1の変換索引バッファ（ＴＬＢ）（252）と、
前記第1のＴＬＢ（252）と並列な第2のＴＬＢ（254）であって、前記第1のデータを受信し、かつ前記第1のＴＬＢにおいて前記第1のデータの受信と並列に第2のデータを受信するように適合される第2のＴＬＢ（254）と、
を含むコンピュータマイクロアーキテクチャ。

20

2．前記第1のデータが整数ロードデータからなり、前記第2のデータは、整数ストアデータ、浮動小数点ロードまたはストアデータ、ＴＬＢ参照および更新命令、ならびに例外およびキャッシュミスのための整数ロードデータの1つ以上を含む、上記1記載のコンピュータマイクロアーキテクチャ。

3．第1のキャッシュ（264）と、
第2のキャッシュ（266）とをさらに含み、
前記第1のＴＬＢ（252）が、受信した第1のデータに基づいてＴＬＢヒット情報を提供し、前記第1のキャッシュ（264）が、前記ＴＬＢヒット情報を保有し、
前記第2のＴＬＢ（254）が物理アドレス情報を提供し、前記第2のキャッシュ（266）が、前記物理アドレス情報を保有する、上記1記載のコンピュータマイクロアーキテクチャ。

30

4．事前検査済みキャッシュタグエントリの物理アドレスを保持するコントロール（262）をさらに含む、上記1記載のコンピュータマイクロアーキテクチャ。5．前記第1のＴＬＢ（252）は、第1および第2のメモリポートによってアクセスされ、前記第2のＴＬＢ（254）は、前記第1および第2のメモリポート、ならびに第3および第4のメモリポートによってアクセスされ、かつ前記第2のＴＬＢ（254）は、例外および特権の情報を記憶する、上記1記載のコンピュータマイクロアーキテクチャ。

6．事前検査済みタグキャッシュを有するコンピュータマイクロアーキテクチャにおいて、
整数ロードＴＬＢ（252）と、
前記整数ロードＴＬＢ（252）と並列に動作するマスタＴＬＢ（254）であって、整数ロードが前記整数ロードＴＬＢに供給され、かつロードおよびストアが前記マスタＴＬＢ（254）に提供される、マスタＴＬＢ（254）と、
前記整数ロードＴＬＢ（252）からの出力を受信する整数ロードデータキャッシュ（264）と、及び

40

前記マスタＴＬＢ（254）からの出力を受信するデータキャッシュ（266）であって、前記整数ロードＴＬＢが、仮想アドレスに基づいてＴＬＢヒット情報を提供するとともに、前記整数ロードデータキャッシュが前記ＴＬＢヒット情報を記憶し、かつ前記マスタＴＬＢが、物理アドレスを前記データキャッシュに提供するとともに、前記データキャッシュが

50

前記物理アドレスを保持する、データキャッシュ（266）と、
を含む変換索引バッファ（TLB）構造。

7．前記マスタTLB（254）および前記整数ロードデータキャッシュ（264）に結合され、前記整数ロードデータキャッシュのキャッシュラインに更新信号または無効化信号を提供する、ストア更新および無効化コントロール（262）と、及び

前記マスタTLBが例外および特権の情報を受信する、例外および特権の情報モジュール（268）と、

をさらに含む、前記6記載の変換索引バッファ（TLB）構造。

8．コンピュータシステムにおける仮想アドレス情報変換の方法であって、

第1の変換索引バッファ（TLB）（252）において第1の仮想アドレス情報を受信する
ステップと、及び

10

同時に、第2のTLB（254）において第2の仮想アドレス情報を受信するステップとを含む方法。

9．前記第1の仮想アドレス情報は整数ロード情報からなり、前記第2の仮想アドレス情報は、整数ストアデータ、浮動小数点ロードおよびストアデータ、ならびにTLB参照および更新命令の1つ以上を含む、上記8記載の方法。

10．前記第1の仮想アドレス情報をTLBヒット情報に変換するステップと、

前記TLBヒット情報を第1のキャッシュに提供するステップと、

前記第2の仮想アドレス情報を物理アドレス情報に変換するステップと、

前記物理アドレス情報を前記第1のキャッシュおよび第2のキャッシュに提供するステップと、

20

前記第2のTLBに例外および特権の情報を記憶するステップと、

をさらに含む、上記8記載の方法。

【0042】

【発明の効果】

本発明により、変換索引バッファを有するキャッシュにおいて、キャッシュヒットのためのアクセス時間が最適化され、コンピュータシステムの性能が改善される。

【図面の簡単な説明】

【図1】従来技術のキャッシュ構造を示す。

【図2】従来技術のキャッシュ構造を示す。

30

【図3】事前検査済みタグキャッシュを有するコンピュータシステムを示す。

【図4】図3の事前検査済みタグキャッシュとともに使用される並列分散型変換索引バッファ構造を示す。

【符号の説明】

250 コンピュータマイクロアーキテクチャ

252 整数ロードTLB（L0TLB）

254 マスタTLB（DTLB）

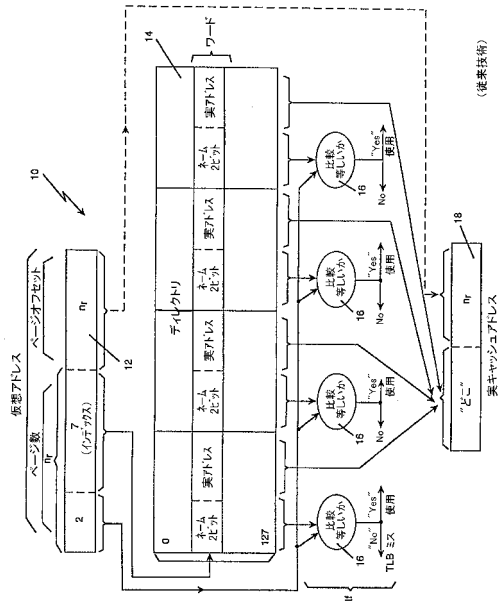
262 ストア／無効化コントロール

264 整数ロードデータキャッシュ（L0D）

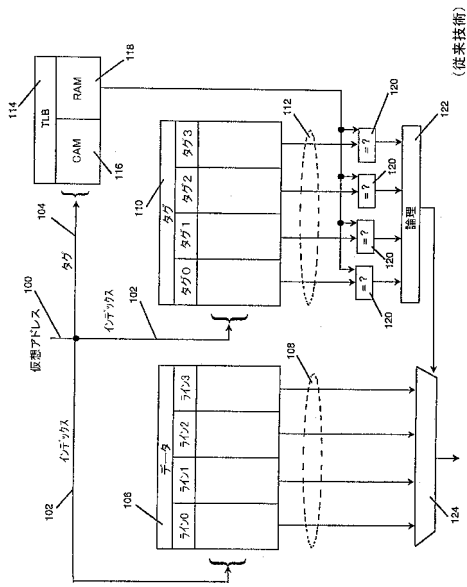
266 L1データキャッシュ

40

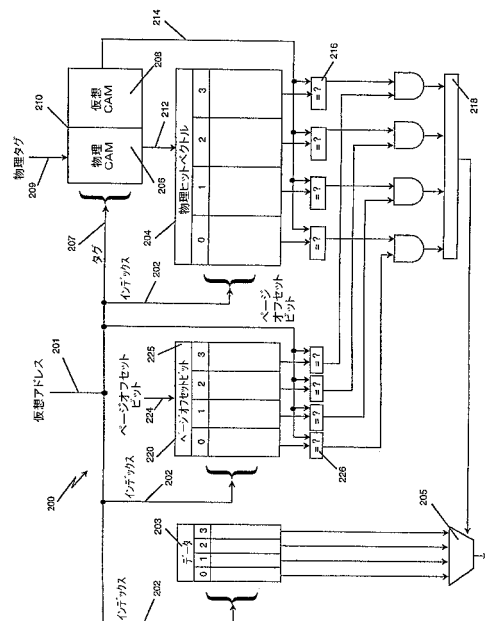
【 図 1 】



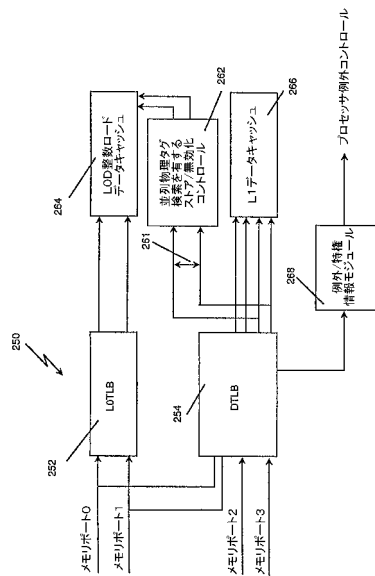
【 図 2 】



【 図 3 】



【 図 4 】



フロントページの続き

(72)発明者 テリー・エル・リヨン

アメリカ合衆国コロラド州 8 0 5 2 5 , フォートコリンズ , シルク・オーク・ドライブ・ 1 4 3 0

審査官 清木 泰

(56)参考文献 特開平 0 4 - 0 4 8 3 5 7 (J P , A)

特開 2 0 0 1 - 1 8 4 2 6 3 (J P , A)

特開平 1 1 - 2 0 3 1 9 9 (J P , A)

特開平 0 8 - 1 3 7 7 5 6 (J P , A)

特表平 1 1 - 5 0 1 7 4 4 (J P , A)

特開平 0 4 - 1 1 7 5 4 0 (J P , A)

星野友彦, 明らかになり始めた P 5 の姿, 日経バイト , 日本, 日経 B P 社, 1 9 9 2 年 1 1 月

, No:105, Pages:226-233

(58)調査した分野(Int.Cl. , D B 名)

G06F12/08-12/12