

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4008826号

(P4008826)

(45) 発行日 平成19年11月14日(2007.11.14)

(24) 登録日 平成19年9月7日(2007.9.7)

(51) Int. Cl. F I
G06F 12/08 (2006.01)
 G06F 12/08 559Z
 G06F 12/08 507Z
 G06F 12/08 511E

請求項の数 3 (全 14 頁)

(21) 出願番号	特願2003-8220 (P2003-8220)	(73) 特許権者	398038580
(22) 出願日	平成15年1月16日(2003.1.16)		ヒューレット・パカード・カンパニー
(65) 公開番号	特開2003-216494 (P2003-216494A)		HEWLETT-PACKARD COMPANY
(43) 公開日	平成15年7月31日(2003.7.31)		アメリカ合衆国カリフォルニア州パロアルト
審査請求日	平成16年6月23日(2004.6.23)		ハノーバー・ストリート 3000
(31) 優先権主張番号	10/050736	(74) 代理人	100087642
(32) 優先日	平成14年1月16日(2002.1.16)		弁理士 古谷 聡
(33) 優先権主張国	米国 (US)	(74) 代理人	100076680
			弁理士 溝部 孝彦
		(74) 代理人	100121061
			弁理士 西山 清春

最終頁に続く

(54) 【発明の名称】 オンチップキャッシュのデータ圧縮により有効キャッシュサイズを増大させるキャッシュ圧縮エンジンのための装置

(57) 【特許請求の範囲】

【請求項1】

キャッシュメモリであって、
 タグ比較及びヒットロジックと共に動作してキャッシュヒットを判定することが可能なキャッシュタグメモリと、
 キャッシュデータメモリと、
 プロセッサによる前記キャッシュタグメモリ及び前記キャッシュデータメモリへの参照を可能にするよう結合されたプロセッサポートと、
 上位レベルキャッシュ及びメインメモリからなるグループから選択されたメモリへミスしたリファレンスを渡すよう結合されたメモリポートと、
 前記キャッシュデータメモリに結合された圧縮エンジンであって、該キャッシュデータメモリからデータを読み出し、該キャッシュデータメモリに圧縮データを書き込み、及び該キャッシュデータメモリ及び前記キャッシュタグメモリからなるグループから選択されたメモリに記憶されている圧縮済フラグを更新するよう結合された、圧縮エンジンと、
 前記キャッシュデータメモリから圧縮データを受容してその解凍データを前記プロセッサポートに提供するよう結合された解凍エンジンと、
 前記メモリポートを介してフェッチされた新しいデータのための場所を形成するために必要に応じて前記キャッシュメモリからデータを追い出すよう結合されたキャッシュ制御ロジックと
 を含み、

前記プロセッサポートを介してデータが読み出される場合には必ず、前記キャッシュデータメモリから読み出された圧縮データが解凍された後に前記プロセッサポートに提示される、
キャッシュメモリ。

【請求項 2】

前記キャッシュデータメモリが、複数の別個に割当可能なサブラインとして編成され、
該キャッシュデータメモリ内の利用可能な該サブラインのリンクされたリストが維持される、請求項 1 に記載のキャッシュメモリ。

【請求項 3】

前記キャッシュデータメモリが、複数のライングループ内の複数の別個に割当可能なサブラインとして編成され、前記キャッシュタグメモリ及び前記タグ比較及びヒットロジックが、キャッシュデータメモリ内の各ライングループで提供されるストレージのWayよりも多数のWayのアソシエティビティを有する、複数Wayのアソシエティビティを含む、
請求項 1 に記載のキャッシュメモリ。

10

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、高性能プロセッサ集積回路のキャッシュ設計の分野に関し、特に、大きくて高レベルのオンチップキャッシュにおいてデータを圧縮する装置に関する。

【0002】

20

【従来の技術】

キャッシュメモリは、それよりも大きくて低速のメモリシステムの内容の部分的な複製を記憶する高速メモリシステムである。キャッシュに記憶された部分的複製は通常は、プロセッサにより最近アクセスされた一層大きなメモリシステムの内容の一部を含むものである。キャッシュメモリは、多数のプログラムが同一の又は近くのコード及びデータロケーションに繰り返しアクセスするという点において利点を提供する。すなわち、命令の実行は、統計的に、最近アクセスされたロケーション又はその最近アクセスされたロケーションの近くのロケーションにアクセスする可能性の方が、メモリ内の他のロケーションにアクセスする可能性よりも高いためである。

【0003】

30

多くの最新のコンピュータシステムは、メインメモリにメモリデータをキャッシュするために階層をなすキャッシュメモリシステムを実施している。かかるシステムのメインメモリは、一般に、ダイナミックランダムアクセスメモリ (DRAM) からなる。IntelのPentium(R)-II及びPentium(R)-III回路を含む多くの一般的なプロセッサは2レベルのキャッシュを有する。3レベル及び4レベルのキャッシュを有する計算処理システムも存在する。

【0004】

キャッシュメモリシステムは、ストレージ(storage)に加えて、一層大きくて低速のメモリシステムのうちキャッシュ内に保持されている部分を識別する装置もまた有し、これはキャッシュタグメモリという形をとることが多い。

【0005】

40

キャッシュシステムは一般に、キャッシュタグメモリサブシステム及びキャッシュデータメモリサブシステムを有する。各キャッシュデータメモリは一般に、キャッシュラインとして知られる所定サイズのデータ単位で動作する。該キャッシュラインのサイズは、マルチレベルのキャッシュの各レベル毎に異なり得るものである。キャッシュラインは、一般にプロセッサにより使用されるワード又はバイトサイズよりも大きく、したがって最近使用されたロケーションだけでなく最近使用されたロケーションの近くのデータを含むことが可能である。

【0006】

代表的なキャッシュメモリシステムでは、特定のメインメモリアドレスにあるメモリロケーションに対して読み出しが行われる際に、メインメモリアドレスの一部からキャッシュ

50

ラインアドレスが導出される。該キャッシュラインアドレスの一部は一般に、キャッシュタグメモリとキャッシュデータメモリに提示され、すなわちそれら両メモリに対する読み出し操作が行われる。

【0007】

キャッシュタグメモリは一般に1つ又は2つ以上のアドレスタグフィールドを含む。キャッシュにおける複数の「Way」の「アソシエイティビティ」に対応するために複数のアドレスタグフィールドを設けることが可能であり、また同フィールドを設けることが多い。各アドレスタグフィールドが、キャッシュラインアドレスの残りのビットと比較されて、キャッシュデータメモリから読み取ったデータのどこかの部分が所望のメインメモリアドレスにおけるデータに対応しているか否かが判定される。そのタグが、所望のデータがキャッシュデータメモリ内にあることを示している場合には、該データがプロセッサと次の下位レベルのキャッシュとに提示され、そうでない場合には、読み出し操作は、次の上位レベルのキャッシュへと進められる。上位レベルのキャッシュが存在しない場合には、読み出し操作はメインメモリへと進められる。NWayセットアソシエイティブキャッシュは、かかる所望のデータアドレスの一部に対するアドレスタグフィールドの比較をN回同時に実行する。

10

【0008】

一般に、タグメモリは、データ情報だけでなくステータス情報も含む。このステータス情報は、キャッシュ内の情報が一層高レベルのメモリに既書き込まれているがまだ更新されていないことを示す「書込済」フラグと、キャッシュ内の情報が有効であることを示す「有効」フラグとを含むことができる。

20

【0009】

キャッシュ「ヒット」は、キャッシュへのメモリアクセスが行われて、該キャッシュシステムが、そのタグメモリを調べることにより、要求されたデータがキャッシュ内に存在し有効であることを見出した際に発生する。キャッシュ「ミス」は、キャッシュへのメモリアクセスが行われて、キャッシュシステムが、そのタグメモリを調べて、要求されたデータがキャッシュ内に存在せず有効でないことを見出した際に発生する。

【0010】

キャッシュ「ミス」が、典型的なマルチレベルキャッシュシステムの低レベルキャッシュで発生した場合には、メインメモリアドレスが次のレベルのキャッシュに渡されて、上位レベルのキャッシュタグメモリが調べられ、該上位レベルで「ヒット」又は「ミス」が存在するか否かが判定される。キャッシュ「ミス」が最高レベルのキャッシュで発生した場合には、一般にリファレンスがメインメモリに渡される。

30

【0011】

一般に、セットアソシエイティブキャッシュタグサブシステムにおけるアソシエイティビティの「Way」の数は、タグメモリの各ラインにおけるアドレスタグの組とそれに対応するコンパレータの組の数である。ストレージ(storage)の「Way」の数は、キャッシュタグメモリの単一ラインを介して記憶し個々に参照することができるキャッシュライン(又はスーパーライン)の数である。殆どのキャッシュでは、アソシエイティビティのWayの数は、ストレージのWayの数と同じである。キャッシュスーパーラインは、タグメモリの1ライン中の単一のアドレスタグにより参照することができる複数のキャッシュラインを組み合わせたものである。

40

【0012】

ライトスルーキャッシュは、キャッシュに記憶されているデータに対する書き込み操作によって、一層高レベルのキャッシュ内又はメインメモリ内のデータを即座に更新するものである。ライトバックキャッシュは、キャッシュに記憶されているデータに対する書き込み操作によって、キャッシュにデータを書き込むが上位レベルのキャッシュ内又はメインメモリ内のデータの更新を遅延させるものである。ライトバックモード及びライトスルーモードでのキャッシュの動作は、当業界で周知のところである。

【0013】

50

キャッシュ「ミス」が何れかのレベルのキャッシュで生じた場合には、上位レベルのキャッシュ又はメインメモリからフェッチされたデータは、一般にキャッシュのデータメモリに記憶され、データが現在存在することを反映するようタグメモリが更新される。また一般に、この新しくフェッチされるデータのための空きを作成するために他のデータを排除する必要がある可能性がある。

【0014】

キャッシュ「ヒット率」は、システム内の全メモリリファレンスに対するキャッシュ内の「ヒット」するメモリリファレンスの比率である。キャッシュを有するプロセッサの実効性能は、キャッシュ「ヒット率」により劇的に変動し得ることが知られている。またヒット率は、プログラムの特性、キャッシュサイズ、割り込みイベントの発生、及びその他の

10

【0015】

したがって、有効キャッシュサイズが大きいと有利である。

【0016】

多くのコンピュータシステムは、マルチプロセッサを実施しており、その個々のプロセッサは、メインメモリリファレンスをキャッシュするためのそれ自体のキャッシュシステムを有する。一般に、かかるシステムのプロセッサは、共有メモリにアクセスすることが可能である。かかるコンピュータシステムのキャッシュメモリではコヒーレンシが必要となる。キャッシュコヒーレンシとは、システム内の各キャッシュが同じメモリ値を「見ている」という意味である。したがって、キャッシュが、特定のメモリロケーションの内容を変更したい場合には、該キャッシュ内の該メモリロケーションの複製を有するシステム内の他の全てのキャッシュの内容を更新し又は無効にしなければならない。

20

【0017】**【発明が解決しようとする課題】**

データを圧縮することができる多数の方法が当業界で知られている。かかる方法として、ランレングスアルゴリズム、反復ベースのアルゴリズム、及び辞書ベースのアルゴリズムが挙げられる。ランレングスアルゴリズムは、一般にファクシミリ伝送に使用されている。ディスクストレージ及びモデムデータ伝送のためにデータを圧縮し解凍するソフトウェアツールは当業界で一般的なものである。ディスクデータをメインメモリにキャッシュする際に該ディスクデータを圧縮し解凍するソフトウェアツールが知られている。メインメモリページを圧縮し解凍するためのソフトウェアユーティリティもまた知られている。かかるユーティリティとツールの殆どは、その圧縮操作及び解凍操作の両方を実行するためにシステムのプロセッサを利用するものであり、かかるユーティリティ及びツールは、システム性能に悪影響を与えるほどプロセッサ時間を消費し得るものである。

30

【0018】

多くのシステムは、メインメモリ又はメインメモリと類似した速度の他のメモリへのディスクデータのキャッシングに備えるものである。本発明の目的上、メインメモリ又はメインメモリと類似した速度のメモリにディスクデータをキャッシュするキャッシュがディスクキャッシュであり、プロセッサにより情報がフェッチされる際にメインメモリリファレンスをキャッシュするキャッシュがプロセッサキャッシュである。オンチップキャッシュは、プロセッサと同じ集積回路上に配置されたプロセッサキャッシュである。

40

【0019】

キャッシュメモリに記憶されるデータは、一般に圧縮形式では記憶されない。これは、有効キャッシュサイズを大きくしそれによりヒット率を高めるのに有利である。

【0020】

一般に、データ圧縮は、データ解凍よりも多くの時間を要する。これは、ランレングスをカウントし、反復を検出し、及び辞書を構築するのに時間を要するためである。

【0021】

50

従来例には、圧縮した命令のストリームを記憶するプロセッサキャッシュが開示されている（例えば特許文献1参照）。また別の従来例にも、圧縮した命令のストリームを記憶するプロセッサキャッシュが開示されている（例えば特許文献2参照）。特許文献1の図8は、ロードブルモジュールを生成する際に命令ストリームを圧縮することを開示しており、プロセッサ上で実行するソフトウェア圧縮ユーティリティが意図されていることは明らかである。特許文献1及び特許文献2の圧縮された命令ストリームは、その命令がキャッシュからプロセッサへと読み込まれる際に専用のハードウェアにより「オンザフライ」で解凍されるものと思われる。

【0022】

プロセッサキャッシュが少なくとも何らかのデータ及び命令を圧縮形式で記憶することにより、圧縮なしで利用可能なサイズよりも大きい有効キャッシュサイズを提供することが有利である。また、残りのシステム構成要素に対してトランスペアレントに（すなわち透過的に）圧縮及び解凍を実行して、情報の圧縮に関連する遅延を伴うことなくキャッシュへの書き込みを行うことが有利である。

10

【0023】

【特許文献1】

米国特許第5,826,054号明細書

【0024】

【特許文献2】

米国特許第6,216,213号明細書

20

【0025】

【課題を解決するための手段】

命令及びデータを記憶するプロセッサキャッシュサブシステムは、命令及びデータ情報を圧縮形式及び非圧縮形式の両方で記憶することが可能である。最初に該キャッシュサブシステムに非圧縮情報が書き込まれ、圧縮エンジンが、そのキャッシュをスキャンして前記非圧縮情報を圧縮情報に置換すると共に再使用に備えて空きスペースを解放するために提供される。

【0026】

特定の実施形態では、キャッシュサブシステムは、システム内に第2レベルのキャッシュを構成しているが、本発明が第3更には第4レベルのキャッシュサブシステムにも適用可能であることが予想される。

30

【0027】

第1レベルキャッシュ内でのキャッシュミスの発生時には、プロセッサリファレンスがキャッシュサブシステムに渡される。タグメモリをアドレス指定するために各リファレンスアドレスの一部が使用される。タグメモリは、少なくとも1つ（好ましくは2つ以上の）タグアドレスフィールドと、キャッシュ管理用のフラグと、キャッシュデータメモリ中の各スーパーラインの各キャッシュラインのロケーションを示す各Way毎のデータラインポインタとを含む。特定の実施形態では、キャッシュは、4ラインのスーパーラインにおける64バイトのラインとして編成される。キャッシュデータメモリは、サブラインのアレイとして編成される。この場合、2つ又は3つ以上のサブラインが各キャッシュラインを構成し、特定の実施形態ではキャッシュは32バイトのサブラインを有するものとなる。キャッシュデータメモリ内の各サブラインに「圧縮済」フラグが関連付けされる。

40

【0028】

キャッシュヒットは、タグメモリのタグアドレスフィールドをリファレンスアドレスの追加ビットと比較することにより決定される。キャッシュヒットが生じると、タグメモリからのデータラインポインタが使用されて、キャッシュデータメモリ内の情報の位置が特定される。圧縮された情報は、第1レベルキャッシュ及びプロセッサに送られる前に解凍される。

【0029】

速度を高めるために、キャッシュミスが生じた際に上位レベルのキャッシュ又はメインメ

50

メモリから取り出された情報は、適切に更新されたタグ情報及びポインタと共にキャッシュデータメモリに非圧縮形式で書き込まれる。この情報は、空きスペースリストに示されたロケーションに書き込まれ、圧縮エンジンにより探し出されて圧縮されるまで非圧縮状態のままとなる。

【0030】

キャッシュに入れられた情報は、タグが再利用されるとき、又は空きスペースリストが所定のしきい値未満になったときに追い出される。

【0031】

もう1つの実施形態では、キャッシュは、16Wayのアソシエティビティのための十分なアドレスタグを有するキャッシュタグメモリによりアドレス指定される。各アドレスタグはWay指示子及びフラグに関連付けされる。各アドレスタグ毎のフラグには、圧縮済フラグ及び幅指示子が存在する。

10

【0032】

関連付けされたキャッシュデータメモリは、12Wayの非圧縮情報を記憶するのに十分なメモリを有する。キャッシュデータメモリ内の各タグメモリラインに対応するロケーションにキャッシュライングループが記憶される。各キャッシュライングループ内の非圧縮データを指定するのに必要となる数よりも多く提供されるアドレスタグは、本書では過剰アドレスタグと称することとする。

【0033】

各アドレスタグと関連付けされたWay指示子は、アドレスタグと関連付けされたキャッシュラインを記憶している可能性のあるキャッシュライングループ中の場所を示す。

20

【0034】

圧縮エンジンは、定期的にキャッシュライングループを読み出し、該キャッシュライングループ内のデータが圧縮可能な場合に該キャッシュライングループを圧縮形式で書き換え、圧縮済フラグ及びそれぞれ関連付けされたタグラインのWay指示子を更新して、データの各ラインを記憶しているキャッシュライングループ中の場所を示す。

【0035】

この実施形態では、キャッシュライングループの一部又は全てのラインが圧縮形式で記憶されている場合は、ライングループ内の残りの空間は、過剰アドレスタグに関連付けされたキャッシュデータにより使用することが可能である。

30

【0036】

本発明は、他のアソシエティビティ及びストレージのWayに適用可能であると予想される。例えば、キャッシュタグメモリの各ラインに32組のアドレスタグを有するキャッシュと24Wayのストレージとを同様にして実施することが可能である。

【0037】

【発明の実施の形態】

コンピュータシステム100（図1）は、プロセッサ集積回路103上に当業界で知られているような第1レベルキャッシュ104と共にプロセッサ102を有している。第1レベルキャッシュ104は、統合化（又は一体化）された命令／データキャッシュとすることが可能であり、また個別の命令及びデータキャッシュを含むことが可能である。システム100はまた、第2レベルキャッシュ106及びシステムコントローラ108を有する。システムコントローラ108は、第2レベルキャッシュ106を、メインメモリ110並びにディスク記憶システム112やその他の入出力装置114に接続する。システムコントローラ108はまた、ディスク記憶システム112からメインメモリ110へのアクセスを提供する。

40

【0038】

第2レベルキャッシュ106（図2）はプロセッサキャッシュである。第2レベルキャッシュ106は、プロセッサポート202を介して、第1レベルキャッシュ104内で失ったりファレンスを受け取る。かかるリファレンスには、読み出し及び書き込みの両操作のためのリファレンスアドレスと書き込み操作のためのデータとを含む。リファレンスアドレスは、4つの部分、すなわち、タグアドレス部分、高アドレスビット部分、ライン・イン・スーパ

50

ーライン (line-in-superline) 部分、及びロケーション・イン・ライン・ライン (location-in-line line) 部分に分割される。リファレンスアドレスのタグアドレス部分は、タグメモリ204中の対応するタグラインを見出すために使用される。

【0039】

第2レベルキャッシュ106は、NWayセットアソシエイティブである。ここで、Nは2以上であることが好ましく、特定の実施形態では6である。タグメモリ204内の各タグラインは、NWayのアソシエイティビティの各々のタグアドレスフィールドとキャッシュ管理用のフラグとを有する。NWayのアソシエイティビティの各々のタグアドレスフィールドは、タグコンパレータ及びヒットロジック206内のリファレンスアドレスの高アドレスビット部分と比較され、その比較結果とタグラインのフラグとを使用してヒットが存在するかどうか判定される。

10

【0040】

また、タグメモリ204内の各タグラインは、キャッシュデータメモリ210内の各スーパーラインの各キャッシュラインの各サブラインのロケーションを示す各Way毎のデータラインポインタ208を有する。特定の実施形態では、キャッシュは、合計256バイトの4ラインのスーパーライン内の64バイトのラインとして編成される。キャッシュデータメモリ210は、サブラインのアレイとして編成され、この場合、2つ又は3つ以上のサブラインが各キャッシュラインを構成し、特定の実施形態では、キャッシュは32バイトのサブラインを有する。「圧縮済」フラグ212は、キャッシュデータメモリ内の各サブラインと関連付けられる。

20

【0041】

第2レベルキャッシュ106が、ヒットを評価する読み出し要求を受け取ると、タグメモリからのデータラインポインタ208の第1のデータラインポインタを使用して、キャッシュデータメモリ210内の参照情報の第1のサブラインとそれに関連する圧縮済フラグ212との位置が特定される。第1のサブラインに関連する圧縮済フラグ212がセットされた場合には、サブラインが解凍エンジン214によりフルキャッシュラインへと拡張され、解凍された情報が下位レベルのキャッシュ104及びプロセッサ102に渡される。圧縮済フラグ212が、サブラインが非圧縮情報を含むことを示している場合には、タグメモリからのデータラインポインタ208の第2のデータラインポインタを使用して参照情報の第2のサブラインの位置が特定され、該第1及び第2のサブラインが下位レベルのキャッシュ104及びプロセッサ102に渡される。

30

【0042】

第1の代替的な実施形態では、データラインポインタ208の第2のデータラインポインタの代わりに、第1のサブラインと共に記憶された順方向リンク (forward link) 216を使用して、参照情報の第2のサブラインの位置が特定される。

【0043】

第2レベルキャッシュ106が、ヒットを評価する書き込み要求を受け取ると、タグメモリからのデータラインポインタ208の第1のデータラインポインタを使用して、キャッシュデータメモリ210に以前に記憶された情報の第1のサブライン及びそれに関連する圧縮済フラグ212の位置が特定される。第1のサブラインに関連する圧縮済フラグ212が、サブラインが非圧縮情報を含むことを示している場合には、第1のサブラインは、第1レベルキャッシュ104又はプロセッサ102からの情報の第1のサブラインで上書きされる。次いで、サブラインポインタ208の第2のサブラインポインタを使用して、第1レベルキャッシュ104又はプロセッサ102からの情報の第2のサブラインで上書きされることになるサブラインの位置が特定される。

40

【0044】

第1の代替的な実施形態では、データラインポインタ208の第2のデータラインポインタの代わりに、第1のサブラインと共に記憶された順方向リンク216を使用して、置換のための参照情報の第2のサブラインの位置が特定される。

【0045】

50

圧縮済フラグ212が、サブラインが圧縮情報を含むことを示している場合には、第1のサブラインが、既述のように下位レベルキャッシュ104及びプロセッサ102からの情報の第1のサブラインで上書きされ、圧縮済フラグがクリアされて該ラインが非圧縮情報を含むことが示される。第2のサブラインが空きスペースリスト218から削除され、データラインポインタ208の第2のデータラインポインタに第2のサブラインへのポインタが書き込まれ、第2のサブラインには、第1レベルキャッシュ104又はプロセッサ102からの情報の第2のサブラインが書き込まれる。

【0046】

第1の代替的な実施形態では、第1のサブラインの順方向リンク216には、データラインポインタ208の第2のデータラインポインタを書き込む代わりに、書き込まれた情報の第2のサブラインへのポインタが書き込まれる。

10

【0047】

空きスペースリスト218は、リンクされたリストとして維持される。特定の実施形態では、この空きスペースリストは、未使用の各サブラインに記憶された順方向リンクを含む。第1の代替的な実施形態では、空きスペースリストに関連付けられたポインタは、未使用の各サブラインの順方向リンク216のフィールドに記憶される。このリストは、空きスペースリストに全てのサブラインが存在するようにブート時に初期化される。データメモリ210の輻輳を回避するために、空きスペースリストの最初の幾つかのエントリが空きリストバッファ222にキャッシュにされ、これにより、空きスペースリストに従うためにデータメモリ210を循環することなくキャッシュアクティビティの短いバーストを行うことが可能になる。

20

【0048】

読み出し及び書き込みのためにキャッシュデータメモリ210の循環が必要ない場合には、圧縮エンジン230は、非圧縮情報の存在を示す圧縮済フラグ212を有するサブラインを求めてキャッシュデータメモリ210をスキャンする。かかるサブラインが見つかった場合には、圧縮エンジン230は、データを圧縮し、逆方向リンク232をたどってサブラインに関連するキャッシュタグの位置を見出す。キャッシュライン全体の残りのサブラインの位置を特定するために、キャッシュタグからデータラインポインタ208の第2のデータラインポインタが取得される。

【0049】

30

第1の代替的な実施形態では、順方向リンク216をたどってキャッシュライン全体の残りのサブラインの位置を特定する。この実施形態では逆方向リンク232は必要ない。

【0050】

次いで圧縮エンジンはキャッシュラインの圧縮を試みる。該圧縮により非圧縮データよりも少数のサブラインに適合する情報が得られた場合には、該圧縮済データがキャッシュデータメモリ210に書き込まれ、圧縮済フラグがセットされ、解放されたサブラインが空きスペースリスト218にリンクされる。

【0051】

キャッシュミスは、上位レベルキャッシュ又はメインメモリに渡され、当業界で知られているようにミスロジック234を介して新しいデータがフェッチされる。必要に応じて追出し制御部244によりデータが追い出されて、新しいデータを記憶するための空間が形成される。

40

【0052】

キャッシュシステムは、当業界で知られているようにライトバックキュー242及び追出し制御ロジック244を含むキャッシュ制御ロジック240を有する。

【0053】

第2の代替的な実施形態では、プロセッサリファレンスは、下位レベルキャッシュにミスが生じた場合に、プロセッサポート300(図3)を介してキャッシュサブシステムに入る。各リファレンスアドレスは、3つの部分、すなわち、タグアドレス部分、高アドレスビット部分、及びロケーションインラインライン部分に分割される。リファレンスアドレス

50

のタグアドレス部分は、タグメモリ304内の対応するタグラインを見つけるために使用される。

【0054】

この実施形態のタグメモリ304の各ラインは、16Wayのアソシエイティビティにとって十分なアドレスタグ400（図4）を有する。各アドレスタグ400は、Way指示子404及びフラグ402に関連付けされる。各アドレスタグ毎のフラグ402には、圧縮済フラグ406、幅指示子410、及び有効フラグ408が含まれる。

【0055】

関連付けされたキャッシュデータメモリ306（図3）は、12Wayの非圧縮情報を記憶する。タグメモリラインに対応するキャッシュデータメモリ内の各ロケーションにキャッシュライングループが記憶される。ここで、各キャッシュライングループ中の非圧縮データを指定するために必要となる数よりも多く提供されるアドレスタグを本書では過剰アドレスタグと称することとする。

10

【0056】

各アドレスタグと関連付けされたWay指示子404,424（図4）は、キャッシュライン412,432といったアドレスタグに関連付けされたキャッシュラインが記憶されているキャッシュデータメモリ310内のキャッシュライングループ440中の位置を示すものとなる。幅指示子410,430は、キャッシュラインの幅をサブラインの個数（整数値）として示すものである。

【0057】

圧縮エンジン308（図3）は、タグメモリ304及びそのフラグ306に対して定期的に読み出しを行い、キャッシュデータメモリ310からキャッシュライングループ440を読み出す。圧縮エンジン308は、キャッシュライングループ440が圧縮可能である場合にはそれを圧縮してキャッシュデータメモリ310に圧縮形式で再び書き込む。圧縮エンジン308はまた、幅指示子410、Way指示子404、及びタグメモリフラグ306の圧縮済フラグ406を、圧縮したキャッシュライングループ440と一致するよう修正する。

20

【0058】

データをキャッシュから読み出す際、該データは、解凍エンジン314を介して読み出され、データが圧縮されていることを圧縮済フラグ404が示している場合に解凍される。

【0059】

キャッシュヒットは、リファレンスアドレスの上位アドレスフィールドを、タグコンパレータ及びヒットロジック320内のアドレスタグ400と比較することにより決定される。キャッシュミスは、ミスロジック322を介して上位レベルキャッシュ又はメインメモリに渡される。キャッシュはまた、当業界で知られているように、ライトバックキュー332、先読み(read ahead)ロジック334、及び追出し制御ロジック336を含む制御ロジック330を有する。追出し制御ロジックは、図示されていない最新使用(recency of use)フラグを含む追加のフラグを利用する。

30

【0060】

この実施形態では、圧縮エンジン308（図5）は、キャッシュタグメモリ304及びキャッシュデータメモリ310を指すアドレスポインタ500を有する。一実施形態では、このポインタをカウンタとして実施して、キャッシュタグメモリ及びキャッシュデータメモリ310の全てのロケーションがスキャンされるようにすることが可能である。別の実施形態では、このポインタを、キャッシュ内の最も最近書き込みが行われたロケーションのキューとして実施し、これにより非圧縮データをキャッシュ内で迅速に見つけることが可能となる。圧縮エンジン308はまた、キャッシュタグメモリ304及びキャッシュデータメモリ310に対する読み出し及び書き込みを行うためのインタフェイス502を有する。

40

【0061】

圧縮エンジン308は、インタフェイス502を介してアドレスポインタ500により示される通りにキャッシュタグメモリ304及びキャッシュデータメモリ310のロケーションを読み出し、キャッシュデータメモリ310から圧縮ロジック504へデータを渡す。圧縮ロジック504は

50

、1つ又は2つ以上の圧縮アルゴリズムをデータに適用することを試みる。特定の実施形態では、圧縮ロジック504は、ランレングスカウント圧縮アルゴリズム、繰返しビットシーケンスカウント圧縮アルゴリズム、及び共通演算コード(common opcode recognition)認識及び符号化アルゴリズムを使用するが、他の実施形態では他の圧縮アルゴリズムを使用することが可能である。共通演算コード認識及び符号化アルゴリズムは、ノーオペレーション(no operation)命令を含む多数の共通命令コードを認識し、それを効率的に符号化することができるものである。圧縮ロジック504が、多数の圧縮アルゴリズムを適用することができる場合には、アルゴリズム指示子は、圧縮ロジックにより生成される圧縮データの一部を構成するものとなる。

【0062】

10

データの圧縮時に圧縮アルゴリズムが必ずしも成功するとは限らないので、圧縮ロジック504は、圧縮したデータだけでなく圧縮成功指示子を生成する。この圧縮成功指示子が制御ロジック506により使用されて、圧縮したデータをキャッシュデータメモリ310に書き戻すべきか否か、及びキャッシュタグメモリ304をキャッシュタグ更新回路508により更新すべきか否かが判定される。

【0063】

この実施形態では、キャッシュライングループの一部又は全てが圧縮形式で記憶されている場合には、該キャッシュライングループ内の残りの空間を、過剰アドレスタグに関連するキャッシュデータにより使用することが可能である。過剰アドレスタグを使用すると、キャッシュデータが圧縮可能であることが判明した場合に最大16通りのキャッシュデータのアドレス指定が可能になり、これにより、3メガバイトのキャッシュが4メガバイトの情報量を保持することが可能となり得る。

20

【0064】

本発明は、他のアソシエイティビティ及びストレージのWayに適用可能であると予想される。例えば、キャッシュタグメモリの各ラインに32組のアドレスタグを有するキャッシュと20Wayのストレージとを同様に実施することが可能である。最適なアソシエイティビティ、ストレージのWay数、及びキャッシュサイズは、キャッシュ階層における圧縮可能なキャッシュのレベル、集積回路の製造技術の改善、及び集積回路が実行することが意図されたアプリケーションによって、変化することが予想される。

【0065】

30

本発明はまた、スーパーラインを有するキャッシュ編成及びスーパーラインを有さないキャッシュ編成、並びに第2レベル以外のレベルのキャッシュにも、等しく適用可能であると予想される。特に、本発明は第3レベルキャッシュとして動作可能であると予想される。本発明はまた、ライトスルー及びライトバックキャッシュシステムに適用可能であると予想される。

【0066】

本発明を特に特定の実施形態に関して示し説明したが、当業者は、本発明の思想及び範囲から逸脱することなく形態及び細部に他の様々な変更を行うことができることを理解されよう。本発明を様々な実施形態に適用する際に、本明細書に開示され特許請求の範囲により理解される一層広義の本発明の思想から逸脱することなく様々な変更を行うことが可能であることが理解されよう。

40

【0067】

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. キャッシュメモリサブシステムのための圧縮エンジン(308)であって、該キャッシュメモリサブシステムが、キャッシュタグメモリ(304)及びキャッシュデータメモリ(310)を有しており、該キャッシュデータメモリ(310)が複数のサブラインとして編成されており、前記圧縮エンジン(308)が、前記キャッシュタグメモリ(304)及び前記キャッシュデータメモリ(310)内へのポインタ(500)と、

50

該ポインタ(500)に結合されたインタフェイス(502)であって、前記キャッシュタグメモリ(304)及び前記キャッシュデータメモリ(310)と結合することができ、タグ情報及び非圧縮データを読み出し、及び修正したタグ情報及び圧縮データを書き込むことが可能である、インタフェイス(502)と、

該インタフェイス(502)により読み出されたデータに少なくとも1つの圧縮アルゴリズムを適用して圧縮データを生成し、及び圧縮成功情報を生成するように結合された、圧縮ロジック(504)と、

前記圧縮成功情報及び前記タグ情報に従って修正したタグ情報を生成するタグライン更新回路(508)と

を含む、圧縮エンジン(308)。

10

2. 前記圧縮ロジック(504)が、複数の圧縮アルゴリズムを用いた圧縮を試行することが可能なものである、前項1に記載の圧縮エンジン。

3. 前記ポインタ(500)が、最近書き込みが行われたアドレスのキューを含む、前項1に記載の圧縮エンジン。

4. 前記ポインタ(500)がカウンタを含む、前項1に記載の圧縮エンジン。

5. キャッシュメモリであって、

タグ比較及びヒットロジック(320)と共に動作してキャッシュヒットを判定することが可能なキャッシュタグメモリ(304)と、

キャッシュデータメモリ(310)と、

プロセッサによる前記キャッシュタグメモリ(304)及び前記キャッシュデータメモリ(310)への参照を可能にするよう結合されたプロセッサポート(300)と、

20

上位レベルキャッシュ及びメインメモリからなるグループから選択されたメモリへミスしたリファレンスを渡すよう結合されたメモリポート(322)と、

前記キャッシュデータメモリ(310)に結合された圧縮エンジン(308)であって、該キャッシュデータメモリ(310)からデータを読み出し、該キャッシュデータメモリ(310)に圧縮データを書き込み、及び前記キャッシュデータメモリ(310)及び前記キャッシュタグメモリ(304)からなるグループから選択されたメモリに記憶されている圧縮済フラグを更新するよう結合された、圧縮エンジン(308)と、

前記キャッシュデータメモリ(310)から圧縮データを受容してその解凍データを前記プロセッサポート(300)に提供するよう結合された解凍エンジン(314)と、

30

前記メモリポート(322)を介してフェッチされた新しいデータののための場所を形成するために必要に応じて前記キャッシュメモリからデータを追い出すよう結合されたキャッシュ制御ロジック(330)と

を含む、キャッシュメモリ。

6. 前記キャッシュデータメモリ(310)が、複数の別個に割当可能なサブラインとして編成され、キャッシュデータメモリ(310)内の利用可能なサブライン(218)のリンクされたリストが維持される、前項5に記載のキャッシュメモリ。

7. 前記キャッシュデータメモリ(310)が、ライングループ内の複数の別個に割当可能なサブラインとして編成され、前記キャッシュタグメモリ(304)及び前記タグ比較及びヒットロジック(320)が、キャッシュデータメモリ(310)内の各ライングループ(440)で提供されるストレージのWayよりも多数のWayのアソシエティビティを有する、複数Wayのアソシエティビティを含む、前項5に記載のキャッシュメモリ。

40

8. キャッシュタグメモリ(304)内の複数Wayのアソシエティビティが、キャッシュデータメモリ(310)内のライングループ(440)内の関連するキャッシュラインの開始点を示すWay指示子が関連付けされたものである、前項7に記載のキャッシュメモリ。

【図面の簡単な説明】

【図1】2レベルキャッシュサブシステムを有するコンピュータシステムを示すブロック図である。

【図2】有効キャッシュサイズの増大のためのデータ圧縮をサポートする上位レベルキャッシュサブシステムを示すブロック図である。

50

【図 3】 上位レベルキャッシュサブシステムの第 2 の代替的な実施形態を示すブロック図である。

【図 4】 上位レベルキャッシュサブシステムのキャッシュデータメモリ編成を示す説明図である。

【図 5】 上位レベルキャッシュサブシステムのための圧縮エンジンを示すブロック図である。

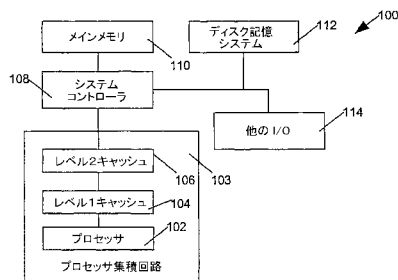
【符号の説明】

- 218 サブライン
- 300 プロセッサポート
- 304 キャッシュタグメモリ
- 308 圧縮エンジン
- 310 キャッシュデータメモリ
- 320 コンパレータ及びヒットロジック
- 320 ヒットロジック
- 322 メモリポート
- 440 ライングループ
- 500 ポインタ
- 502 インタフェイス
- 504 圧縮ロジック
- 508 タグライン更新回路

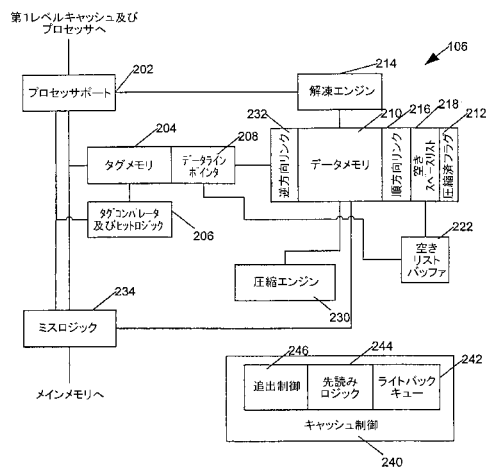
10

20

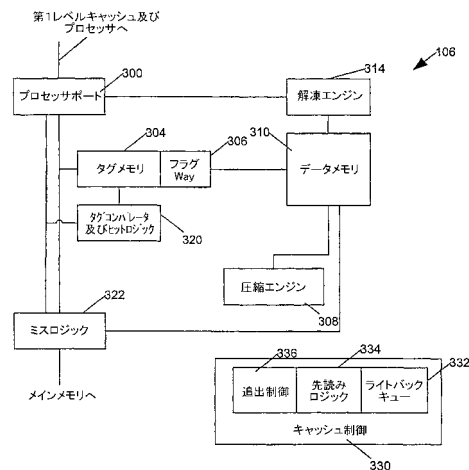
【図 1】



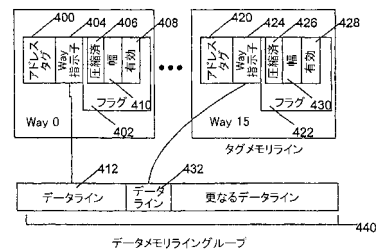
【図 2】



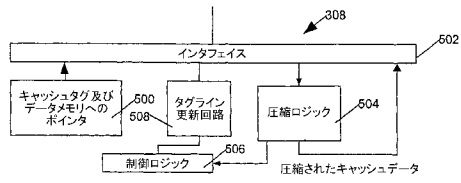
【図 3】



【図 4】



【図 5】



フロントページの続き

- (72)発明者 サミュエル・ナフジガー
アメリカ合衆国コロラド州80525, フォートコリンズ, アシュマウント・ドライブ・3749
- (72)発明者 ウェイン・ケヴァー
アメリカ合衆国コロラド州80525, フォートコリンズ, クレイトン・コート・4101

審査官 清木 泰

- (56)参考文献 特開平10-232838 (JP, A)
特開平10-097421 (JP, A)
特開平08-249159 (JP, A)
特開平08-115239 (JP, A)
特開平05-073413 (JP, A)
特開2000-285019 (JP, A)
国際公開第00/045516 (WO, A1)
特表2002-536863 (JP, A)
Jun Yang, Youtao Zhang, Rajiv Gupta, Frequent Value Compression in Data Caches, Proceedings of 33rd Annual IEEE/ACM International Symposium on Microarchitecture(MICRO-33), IEEE, 2000年12月10日, Pages:258-265
小原清弘、高本良史、小田原宏明、ディスクアレイのキャッシュ上圧縮データ管理方式, 情報処理学会研究報告, 日本, 社団法人情報処理学会, 1997年 6月27日, Vol:97, No:61, (97-ARC-124), Pages:31-36
96 Hot Chips シンポジウムのデッドヒート2, エレクトロニクス, 日本, 株式会社オーム社, 1997年 1月 1日, Vol:42, No:1, Pages:9-16
Jang-Soo Lee, Won-Kee Hong, Shin-Dug Kim, Design and Evaluation of a Selective Compressed Memory System, Proceedings of International Conference on Computer Design 1999(ICCD '99), IEEE, 1999年10月10日, Pages:184-191
Simon Segars, Keith Clarke, Liam Goudge, Embedded Control Problems, Thumb, and the ARM 7TDMI, IEEE Micro, IEEE, 1995年10月, Vol:15, No:5, Pages:22-30

- (58)調査した分野(Int.Cl., DB名)
G06F12/08-12/12