(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2004-199677
(P2004-199677A)
(43)公開日　平成16年7月15日(2004.7.15)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 12/08	G06F 12/08	519E	5B005
	G06F 12/08	503C	
	G06F 12/08	503F	
	G06F 12/08	511E	
	G06F 12/08	573	

審査請求 未請求　請求項の数 10　OL　（全 18 頁）

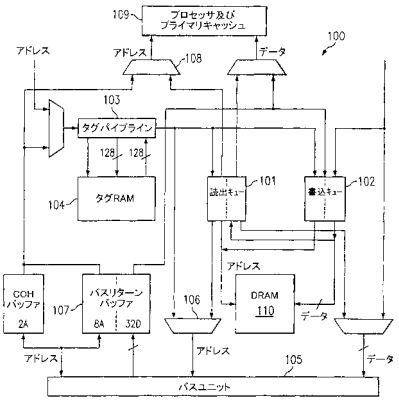
(21)出願番号　特願2003-415025(P2003-415025)
(22)出願日　平成15年12月12日(2003.12.12)
(31)優先権主張番号　10/319205
(32)優先日　平成14年12月13日(2002.12.13)
(33)優先権主張国　米国(US)

(71)出願人　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　100087642
弁理士　古谷　聡
(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】キャッシュを動作させるためのシステム及び方法

(57)【要約】
【課題】高速で旧いデータが使用されることのないキャッシュメモリを提供すること。
【解決手段】キャッシュの動作方法であって、少なくとも２つのキャッシュラインについてタグメモリ(104)から第１の情報を読み出し、少なくとも２つのキャッシュラインについてタグメモリ(104)から第２の情報を読み出し、タグメモリに第３の情報を書き込んで前記第１の情報を更新し、(i)前記第２の情報を読み出す前記ステップに関連するタグメモリのアドレスを(ii)前記第３の情報を書き込む前記ステップに関連する前記タグメモリのアドレスと比較し、該比較に応じて前記第２の情報を前記第３の情報に選択的に置換し、前記比較ステップの後に第４の情報を前記タグメモリに書き込んで前記第２の情報を更新する、という各ステップを含む方法。
【選択図】図１



【特許請求の範囲】
【請求項１】
キャッシュを動作させるための方法であって、
少なくとも２つのキャッシュラインについてタグメモリから第１の情報を読み出し、
少なくとも２つのキャッシュラインについて前記タグメモリから第２の情報を読み出し、
前記タグメモリに第３の情報を書き込んで前記第１の情報を更新し、
(i) 前記第２の情報を読み出す前記ステップに関連する前記タグメモリのアドレスを、(ii) 前記第３の情報を書き込む前記ステップに関連する前記タグメモリのアドレスと比較し、該比較に応じて前記第２の情報を前記第３の情報に選択的に置換し、
前記比較ステップの後に、第４の情報を前記タグメモリに書き込んで前記第２の情報を更新する、
という各ステップを含む、キャッシュを動作させるための方法。
【請求項２】
前記比較ステップが、
(i) 前記第２の情報を読み出す前記ステップに関連する前記タグメモリのウェイを、(ii) 前記第３の情報を書き込む前記ステップに関連する前記タグメモリのウェイと比較し、該比較に応じて前記第２の情報を前記第３の情報に選択的に置換する前記ステップを実行する、というステップを含む、請求項１に記載の方法。
【請求項３】
選択的な置換を行う前記ステップが、所定の時間にわたり前記第３の情報を遅延させて遅延された情報を提供するステップを含む、請求項１に記載の方法。
【請求項４】
前記遅延された情報を前記第２の情報の代わりにタグパイプラインに提供するステップを更に含む、請求項６に記載の方法。
【請求項５】
前記第３の情報(W_A)を書き込む前記ステップの前に、少なくとも２つのキャッシュラインについて前記タグメモリから第５の情報(R_B)を読み出し、
(i)前記第２の情報を読み出す前記ステップに関連する前記タグメモリのアドレスを、(ii)前記第６の情報を書き込む前記ステップに関連する前記タグメモリのアドレスと比較し、該比較に応じて、前記第２の情報を前記第６の情報に選択的に置換する、
という各ステップを更に含む、請求項１に記載の方法。
【請求項６】
キャッシュを動作させるための回路であって、
少なくとも２つのキャッシュラインの各々毎に第１及び第２の情報を格納するタグメモリ(104)と、
前記タグメモリ(104)にアドレス指定情報を供給して前記第１及び第２の情報を読み出すために前記タグメモリに接続されたタグパイプライン(103)と、
前記第１及び前記第２の情報を前記タグパイプライン(103)に送信するために前記タグメモリを前記タグパイプライン(103)に接続するデータバスとを含み、
前記タグパイプライン(103)が更に、前記タグメモリ(104)に第３の情報を書き込ませて前記第１の情報を更新させるよう構成されており、該回路が更に、
(i) 前記第２の情報に関連する前記タグメモリ(104)のアドレスを、(ii) 前記第３の情報に関連する前記タグメモリ(104)のアドレスと比較するよう構成されたコンパレータと、
該コンパレータにより出力された比較信号に応じて前記第２の情報を前記第３の情報により選択的に更新させて更新された第２の情報を形成する、第１のマルチプレクサと、
前記タグパイプラインに前記更新された第２の情報を前記第４の情報で選択的に変更させるよう動作する第２のマルチプレクサと
を含む、キャッシュを動作させるための回路。

【請求項７】
(i) 前記第２の情報を読み出す前記ステップに関連する前記タグメモリ(104)のウェイと、(ii) 前記第３の情報を書き込む前記ステップに関連する前記タグメモリ(104)のウェイとを受信して比較し、該比較に応じて前記第２の情報の前記第３の情報への置換を選択的に開始するよう動作する、バイパス制御ロジックを更に含む、請求項６に記載の回路。
【請求項８】
前記第３の情報を所定の時間にわたり選択的に遅延させて、遅延された情報を提供するよう動作する、遅延回路を更に含む、請求項６に記載の回路。
【請求項９】
前記第２の情報の代わりに前記遅延された情報を前記タグパイプライン(103)に選択的に提供するために前記タグパイプライン(103)に接続されたマルチプレクサを更に含む、請求項８に記載の回路。
【請求項１０】
前記第３の情報を同時に遅延させるよう互いに並列に動作する複数の前記タグパイプライン(103)を更に含む、請求項８に記載の回路。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般にコンピュータメモリシステムに関し、特にメモリを使用するデータへのアクセス時間を改善するためのシステム内のメモリ制御に関するものである。
［関連特許出願の相互参照］
本特許出願は、本出願人の先の出願である、2001年5月10日出願の「FAST PRIORITY DETERMINATION CIRCUIT WITH ROTATING PRIORITY」と題する米国特許出願第09/853,738号、2001年5月10日出願の「SYSTEM OF AND METHOD FOR MEMORY ARBITRATION USING MULTIPLE QUEUES」と題する米国特許出願09/853,951号、及び2002年4月9日出願の「SYSTEM OF AND METHOD FOR FLOW CONTROL WITHIN A TAG PIPELINE」と題する米国特許出願第10/118,801号に関連するものである。
【背景技術】
【０００２】
コンピュータが情報を処理する速度を上げる必要性が次第に高まっている。全体的な処理速度を上げるための１つの要素は、メモリアクセス時間を改善することを含む。当業者であれば認識しているように、処理速度に関する主な制約は、メモリレイテンシであり、これは、多数の技術及び方策を用いて取り組まれてきた問題である。
【０００３】
メモリアクセス時間を改善するための一般的な方法は、メインメモリに加えてキャッシュメモリを設けることである。キャッシュメモリは、典型的にはプロセッサに付随し、メインメモリよりも短いアクセス時間を必要とするものである。プロセッサにより読み出され及び書き込まれたデータのコピーがキャッシュに保持される。キャッシュシステムによっては、最新の読み出し及び書き込みを保持するが、別のシステムでは、キャッシュメモリ中にどのデータが保持されているかを決定するための一層複雑なアルゴリズムを有することが可能である。プロセッサが、キャッシュ内に現時点で存在するデータを要求すると、キャッシュメモリのみがアクセスされる。キャッシュメモリへのアクセス時間がメインメモリよりも短いため、処理速度が改善される。今日、メインメモリからのメモリアクセスに要する時間は250ナノ秒又はそれ以上であるが、キャッシュアクセスに要する時間は２～３ナノ秒程度に短くすることができる。
【０００４】
キャッシュシステムはまた、データ書き込みの実効速度を増大させるために使用することが可能である。例えば、プロセッサは、特定の記憶場所に書き込みを行うことになる場合に、キャッシュメモリへのデータ書き込みを実行することが可能である。次いで、該キャッシュメモリ及びそれに関連する制御ロジックは、プロセッサが他のタスクを進める間

に、該キャッシュメモリ内のデータをメインメモリに書き込むことができる。

【０００５】

コンピュータシステムはまた、キャッシュの使用を拡張させて、多レベルの階層のキャッシュメモリを採用することが可能であり、該キャッシュメモリは、最高レベルの階層に位置する少量の相対的に速い一次又は第１レベルのキャッシュメモリから、最低レベルの階層に位置する相対的に遅くて低コストの大容量メモリへと続くものである。該一次キャッシュは、プロセッサ集積回路内に物理的に集積化させることが可能であり、又はプロセッサの近くに物理的に取り付けることが可能である。中央処理装置（CPU）と同じチップ上に組み込まれる一次キャッシュは、CPUのサイクル周波数と等しいクロック周波数（ひいてはアクセス時間）を有することができる。命令用の一次キャッシュとデータ用の一次キャッシュとが別個に存在することが可能である。一次キャッシュは一般に、データ及び／又は命令レイテンシを最小限にするよう相対的に少量のメモリに関する性能を最大限にするものである。加えて、一次キャッシュは一般に広帯域幅の転送をサポートする。二次キャッシュ又は三次キャッシュもまた使用することが可能であり、一般にプロセッサから更に離れた場所に配置される。かかる二次及び三次キャッシュは、一次キャッシュに対する「補佐」を提供するものであり、また一般に一次キャッシュよりも大きい容量、大きいレイテンシ、及び狭い帯域幅を提供するものである。

【０００６】

プロセッサが一次キャッシュからのデータ又は命令を要求し、及びそのアイテムが該一次キャッシュ内に存在する場合には、キャッシュ「ヒット」が生じることになる。逆に、そのアイテムが存在しない場合には、一次キャッシュ「ミス」が生じる。一次キャッシュミスが生じた場合には、要求されたアイテムは、次のレベルのキャッシュメモリから取り出され、又は要求されたアイテムがキャッシュメモリ内に含まれていない場合には、メインメモリから取り出される。

【０００７】

一般に、メモリは、ワードと呼ばれる複数のビットグループ（例えば、32ビット又は64ビット／ワード）へと編成される。１つのキャッシュとその次に低いレベルのメモリ階層との間で転送することができる最小のメモリ量は、１キャッシュライン、又は場合によっては１ブロックと呼ばれる。１キャッシュラインは一般に複数のワードである（例えば、16ワード／キャッシュライン又はブロック）。

【０００８】

キャッシュは、３つの主なアーキテクチャ、すなわち、ダイレクトマップ、セットアソシエイティブ、及びフルアソシエイティブを用いて構成されてきた。該３つのキャッシュタイプの詳細については、以下の３つの従来技術に関する非特許文献１～３に記載されている。

【非特許文献１】De Blasi著「Computer Architecture」 (ISBN0-201-41603-4、Addison-Wesley、1990、pp.273-291)

【非特許文献２】Stone著「High Performance Computer Architecture」 (ISBN0-201-51377-3、Addison-Wesley、第２版、1990、pp.29-39)

【非特許文献３】Tabak著「Advanced Microprocessors」 (ISBN0-07-062807-6、McGraw Hill、1991、pp.244-248) ダイレクトマッピングの場合、１ラインのメモリが要求された際には、キャッシュ内の１ラインのみが、一致するインデックスビットを有する。それゆえ、そのデータを直ちに取り出してデータバス上に出力した後、システムがそのアドレスの残りが一致するか否かを判定することができる。該データは、有効である場合も有効でない場合もあるが、該データが有効である通常の場合には、該データビットがデータバス上で利用可能となった後、システムが該データの妥当性を確認することができる。

【０００９】

また、セットアソシエイティブキャッシュの場合には、インデックスアドレスが計算され、及びタグアドレスが読み出されて比較されるまで、そのアドレスにどのラインが対応

するかは分からない。すなわち、セットアソシエイティブキャッシュの場合には、タグの比較結果を使用して、１セットのラインのうちのどのデータビットラインをプロセッサに提示するかを選択する。

【００１０】

１つのキャッシュがラインアドレス全体をデータと共に格納し、及び該キャッシュ内のあらゆる場所に任意のラインを配置することができる場合、該キャッシュはフルアソシエイティブであると呼ばれる。しかし、任意のラインをどの場所にでも配置することができる大きなキャッシュの場合には、該キャッシュ内に所与のエントリが格納されるているか否か又はその格納場所を迅速に判定するために大量のハードウエアが必要となる。大きなキャッシュの場合、より高速で省スペースの代替的な形態として、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、次いで各物理アドレスの上位ビットからなる残りのセット（タグと呼ばれる）をデータと共に格納することが挙げられる。インデックスを用いるキャッシュでは、特定のアドレスを有するアイテムは、インデックスにより指定される１セットのキャッシュライン内にのみ配置することができる。所与のアドレスに関するインデックスがそのサブセット内の厳密に１ラインへとマップするようキャッシュが構成される場合には、該キャッシュはダイレクトマップであると呼ばれる。インデックスがサブセット内の２つ以上のラインにマップする場合には、そのキャッシュはセットアソシエイティブであると呼ばれる。アドレス空間を複数のセットへと区分化するセットインデックスを提供するために、アドレスの全て又は一部がハッシュされる。

【００１１】

３タイプのキャッシュの全てにおいて、入力アドレスが比較ロジックに加えられて、所望の記憶場所が該キャッシュ内に存在するか否かが判定される。一般に、アドレスのサブセット（タグビットと呼ばれる）が入力アドレスから抽出されて、各キャッシュエントリのタグビットと比較される。該タグビットが一致する場合には、それに対応するデータがキャッシュから抽出される。

【００１２】

一般にダイレクトマッピングキャッシュは、最も速いアクセスを提供するが、タグビットの比較に最も時間を要するものである。フルアソシエイティブキャッシュは、より長いアクセス時間を有し、より大きな電力を消費し、及びより複雑な回路を必要とするものである。

【００１３】

自らのキャッシュをそれぞれ備える複数のプロセッサが１つのシステム内に含まれる場合には、それらキャッシュ間のコヒーレンシを維持するために、キャッシュコヒーレンシプロトコルが使用される。これは、同じデータが、２つ以上のキャッシュに格納され又は２つ以上のキャッシュにより要求される可能性があるからである。キャッシュコヒーレンシプロトコルには次の２種類が存在する。

１．ディレクトリベース：

１ブロックの物理的なメモリに関する情報が単一の共通の場所に維持される。該情報は通常は、どの（１つ又は２つ以上の）キャッシュが該ブロックのコピーを有しているか、及び該コピーが将来的な変更について排他的であるか否かを含む。特定のブロックへのアクセスは、最初に、ディレクトリの照会を行って、メモリデータが旧いものであり現在のデータが他の何れかのキャッシュ（もしあるなら）内に存在するかを確認する。そうである場合には、変更されたブロックを含むキャッシュが強制的にそのデータをメモリに戻す。次いで該メモリが該データを新たなリクエスタに転送し、該ブロックの新たな場所でディレクトリが更新される。このプロトコルは、バスモジュール間（又はキャッシュ間）の混乱を最小限に抑えるが、大きなディレクトリサイズを必要とすることに起因して、一般にレイテンシが長く、及び構築に要するコストが高いものとなる。

２．スヌーピング：

物理的なメモリブロックからのデータのコピーを有するあらゆるキャッシュは、そのデ

ータブロックに関する情報のコピーも有する。各キャッシュは、通常は共有メモリバス上に配置され、全てのキャッシュコントローラが、該バスを監視し又は「スヌープ」して、それらキャッシュが共有ブロックのコピーを有するか否かを判定する。

【００１４】

スヌーピングプロトコルは、キャッシュ及び共有メモリを利用するマルチプロセッサシステムアーキテクチャに良く適したものである。これは、該プロトコルが、バスとメモリとの間で一般に提供される既存の物理的な接続に関連して動作するものであるからである。コヒーレンシ情報の量が、メインメモリ内のブロックの数ではなくキャッシュ内のブロックの数に比例するため、ディレクトリプロトコルよりもスヌーピングの方が好ましい場合が多い。

【発明の開示】

【発明が解決しようとする課題】

【００１５】

コヒーレンシ問題は、マルチプロセッサアーキテクチャにおいて、プロセッサが、メモリブロック又はオブジェクトをメモリに書き込むために排他的にアクセスしなければならない場合、及び／又はオブジェクトを読み出す際に最新のコピーを有していなければならない場合に生じる。スヌーピングプロトコルは、書き込まれることになるオブジェクトを共有する全てのキャッシュの位置を特定しなければならない。共有データに対する書き込みの結果は、そのデータの全ての他のコピーを無効にし、又は共有されるコピーの全てに対して書き込みをブロードキャストするものとなる。コヒーレンシプロトコルはまた、メモリ読み出し中に全てのキャッシュを検査して、どのプロセッサが情報の最新のコピーを有しているかを判定しなければならない。

【００１６】

プロセッサ間で共有される情報に関するデータが、キャッシュブロック内に提供されるステータスビットに加えられて、スヌーピングプロトコルが実施される。この情報は、バスアクティビティを監視する際に使用される。読み出しミスの発生時には、全てのキャッシュが、要求された情報ブロックのコピーを有しているか否かを確認して、ミスしたキャッシュに情報を供給するといった適切な動作を行う。同様に、書き込み時には、全てのキャッシュが、データのコピーを有しているか否かを確認し、次いで、例えば、それらのデータのコピーを無効にすることにより又はそれらのデータのコピーを変更して最新の値を反映させることにより動作する。

【００１７】

スヌーピングプロトコルは次の２種類からなる。

【００１８】

書き込み無効化：

書き込み時のプロセッサは、他のキャッシュ内の全てのコピーを無効化した後、ローカルコピーを変更する。該プロセッサは次いで、別のプロセッサがそのデータを要求する時まで、該データを自由に更新することができる。書き込み時のプロセッサは、バスを介して無効化信号を送出し、全てのキャッシュがデータのコピーを有するか否かを確認する。データのコピーを有する場合には、それらのキャッシュは該データを含むブロックを無効化しなければならない。この方式は、多数のリーダ（すなわち読み出し）を許容するが、単一のライタ（すなわち書き込み）しか許容しないものである。

【００１９】

書き込みブロードキャスト：

共有されている全てのブロックを無効化するのではなく、書き込み時のプロセッサは、バスを介して新たなデータをブロードキャストする。次いで全てのコピーがその新たな値で更新される。この方式は、共有データに対する書き込みを連続してブロードキャストし、一方、上記の書き込み無効化方式は、後続の書き込みのために１つローカルコピーしか存在しないように全ての他のコピーを削除するものである。書き込みブロードキャストプロトコルは通常は、データを共有されたもの（ブロードキャスト）としてタグ付けするこ

とを可能にし、又はデータを個人的なもの（ローカル）としてタグ付けすることを可能にする。コヒーレンシに関する更なる情報については、J.Hennessy,D.Patterson、Computer Architecture:A Quantitative Approach、Morgan Kaufmann Publishers,Inc(1990)を参照されたい。

【００２０】

スヌーピングコヒーレンシマルチプロセッサシステムアーキテクチャでは、システムバス上の各コヒーレントトランザクションが各プロセッサのキャッシュサブシステムに転送されて、コヒーレンシ検査が実行される。この検査は通常は、プロセッサのパイプラインを妨害し及び／又は中断させるものとなる。これは、該コヒーレンシ検査が行われている間はプロセッサがキャッシュにアクセスすることができないからである。

【００２１】

二重のキャッシュタグを有さない従来のシングルポートキャッシュでは、プロセッサのパイプラインは、キャッシュコントローラが他のプロセッサのためのキャッシュコヒーレンシ検査を処理してビジーである際には、キャッシュアクセス命令に応じて停止される。各スヌープ毎に、キャッシュコントローラは、最初にそのスヌープアドレスについてキャッシュタグを検査し、次いでヒットが存在する場合にはそのキャッシュ状態を変更しなければならない。アトミック（分離できない）タグ読み出し及び書き込みのためのキャッシュ帯域幅を割り当てることにより（変更が見込まれる場合）、スヌープがタグ書き込みを必要としない場合に要する時間よりも長く、プロセッサからキャッシュがロックされる。「アトミック」とは、読み出し及び書き込みが緊密に結束され、書き込みと読み出しとの間に何も行われないよう結合されることを意味している。例えば、キャッシュ問い合わせの80～90％はミスであり、すなわちタグ書き込みは必要とされない。マルチレベルキャッシュ階層の場合には、これらのミスの多くは、包含関係が遵守される場合には、フィルタリングすることができる。包含関係は、より低いキャッシュレベルの内容に関する情報を最も高いレベルのキャッシュに格納することを可能にする。

【００２２】

数多くの用途でコンピュータが情報を処理する速度はまた、キャッシュ、特に一次キャッシュのサイズを大きくすることにより増大させることができる。一次キャッシュのサイズを大きくすると、メインメモリアクセスが削減され、全体的な処理速度が増大する。同様に、二次キャッシュのサイズを大きくすると、メインメモリアクセスが削減され、全体的な処理速度が増大するが、これは一次キャッシュのサイズを大きくする場合ほど効果的なものではない。

【００２３】

一般に、コンピュータシステムでは、一次キャッシュ、二次キャッシュ、及び三次キャッシュは、スタティックランダムアクセスメモリ（SRAM）を用いて実施される。SRAMを使用することにより、アクセス時間を短縮することが可能となり、これにより情報を処理する速度が増大する。ダイナミックランダムアクセスメモリ（DRAM）は一般にメインメモリに使用される。これは、DRAMが、安価であり、必要とされる電力が少なく、かつより大きな記憶密度を提供するからである。

【００２４】

一般に、従来技術によるコンピュータシステムは、所与の時点におけるキャッシュへの未処理のトランザクションの数を制限している。１つのキャッシュにより２つ以上のトランザクションが受容された場合には、該キャッシュはその要求をシリアルに処理することになる。例えば、１つのキャッシュにより２つのトランザクションが受信された場合には、受信された第１のトランザクション要求が最初に処理され、該第１のトランザクションが完了するまで第２のトランザクションが保持されることになる。第１のトランザクションが完了すると、キャッシュは第２のトランザクション要求を処理することになる。

【００２５】

複数のキャッシュ及びメインメモリにわたりキャッシュコヒーレンシを維持するプロトコルは非常に多く存在する。かかるプロトコルの１つが、MESIと呼ばれるものであり、こ

れについては、M. Papamarcos及びJ. Patel著「A Low Overhead Coherent Solution for Multiprocessors with Private Cache Memories」（Proceedings of the 11th International Symposium on Computer Architecture, IEEE, New York (1984), pp.348-354）に詳述されている。MESIは、変更、排他、共有、及び無効という、データに関する４つのステータス条件を意味する。MESIプロトコルでは、キャッシュラインはその用途に従って分類される。変更キャッシュラインは、特定のラインが該ラインの現在の「所有者」であるキャッシュにより書き込みが行われたことを示す（ここで用いる用語「所有者」は、データに関する制御を遂行する権限を表す呼称のことである）。排他キャッシュラインは、所与のキャッシュがそのキャッシュラインの排他的な所有権（キャッシュコントローラがキャッシュラインを変更することを可能にするもの）を有していることを示す。共有キャッシュラインは、１つ又は複数のキャッシュが該ラインの所有権を有することを示す。共有キャッシュラインは、読み出し専用であると見なされ、そのキャッシュ下にある任意の装置はそのラインを読み出すことはできるが該キャッシュに書き込みを行うことは許可されない。無効キャッシュラインは、キャッシュがもはやキャッシュラインを所有していないためにデータが有効でない可能性があるキャッシュラインを特定する。

【課題を解決するための手段】

【００２６】

本発明の一態様によるキャッシュ動作方法は、タグメモリから少なくとも２つのキャッシュラインについて第１の情報を読み出し、タグメモリから少なくとも２つのキャッシュラインについて第２の情報を読み出し、タグメモリに第３の情報を書き込んで前記第１の情報を更新し、（ｉ）前記第２の情報を読み出す前記ステップに関連するタグメモリのアドレスを、（ｉｉ）前記第３の情報を書き込む前記ステップに関連するタグメモリのアドレスと比較し、該比較に応じて、前記第２の情報を前記第３の情報に選択的に置換し、前記比較ステップの後に第４の情報をタグメモリに書き込んで前記第２の情報を更新する、という各ステップを含むものである。

【００２７】

本発明の別態様によるキャッシュ動作回路は、それぞれ少なくとも２つのキャッシュラインについて第１の情報及び第２の情報を格納するタグメモリと、該第１及び第２の情報を読み出すために該タグメモリにアドレス指定情報を供給するよう該タグメモリに接続されたタグパイプラインと、該タグパイプラインに前記第１及び第２の情報を送信するために前記タグメモリを前記タグパイプラインに接続するデータバスとを含み、前記タグパイプラインが更に、前記タグメモリに第３の情報を書き込ませて前記第１の情報を更新させるよう構成されており、該動作回路が更に、（ｉ）前記第２の情報に関連するタグメモリのアドレスを（ｉｉ）前記第３の情報に関連するタグメモリのアドレスと比較するよう構成されたコンパレータと、該コンパレータにより出力される比較信号に応じて前記第２の情報を前記第３の情報により選択的に更新させて更新された第２の情報を生成させる、第１のマルチプレクサと、前記タグパイプラインに前記更新された第２の情報を第４の情報に選択的に変更させるよう動作する第２のマルチプレクサとを含む。

【００２８】

本発明の別の態様によるキャッシュ動作回路は、タグメモリから少なくとも２つのキャッシュラインについて第１の情報を読み出す手段と、前記タグメモリから少なくとも２つのキャッシュラインについて第２の情報を読み出す手段と、前記タグメモリに第３の情報を書き込んで前記第１の情報を更新させる手段と、（ｉ）前記第２の情報の読み出しに関連するタグメモリのアドレスと（ｉｉ）前記第３の情報の書き込みに関連するタグメモリのアドレスとを比較し、及び該比較に応じて前記第２の情報を前記第３の情報に選択的に置換する手段と、該選択的な置換の後に前記タグメモリに第４の情報を書き込んで前記第２の情報を更新させる手段とを含む。

【発明を実施するための最良の形態】

【００２９】

一般に、メモリ階層は、種々の速度で動作する極めて多数の構成要素及びサブシステム

を含む。これらの速度は、関連する中央処理装置（CPU）の速度とは異なる可能性がある。一般に、CPUからの「距離」が増すと構成要素の速度は遅くなる。かかる速度の不一致は、遅延される動作をキューイングし又は格納することにより対処することができる。例えば、スタティックランダムアクセスメモリ（SRAM）は、その固有の動作速度が故にキャッシュ動作で使用される。これとは対照的に、ダイナミックランダムアクセスメモリ（DRAM）技術は一般にキャッシュには使用されてこなかった。これは、アクセス時間に関してメインメモリと比べた場合にほとんど利点を提供しないからである。しかし、DRAM技術は、SRAMと比べて１記憶ビット当たりのコストが約4分の1であり、その高い密度により、所与の面積当たり遙かに大きなキャッシュを実施することが可能となる。「パッケージ上の」面積が重要である場合には、SRAMに対するDRAMの密度の利点もまた重要となる。

【００３０】

SRAMで実施された一次キャッシュのサイズが大きくなると、二次又は三次キャッシュに必要となるメモリのサイズも大きくなる。一般に、多段キャッシュ階層が実施される場合には、各後続レベルにおけるメモリのサイズは４倍又は８倍に大きくなる。それゆえ、１メガバイトの一次キャッシュの場合には、４～８メガバイトの二次キャッシュが望ましい。二次キャッシュのサイズが大きくなると、密度が限られていることに起因して、SRAMを利用することができなくなる。DRAM技術を用いることにより、32メガバイト又はそれ以上の二次キャッシュが実用化される。DRAM製二次キャッシュに格納されている情報にアクセスするための時間は増大するが、その全体的な影響は、より大きな一次キャッシュによる一次キャッシュのミス率の低下により相殺される。換言すれば、一次キャッシュのサイズが大きくなると、システム性能を劣化させることなく二次キャッシュが一層長いレイテンシを呈することが可能となる。

【００３１】

二次キャッシュに関するレイテンシを更に短縮させるために、より速いアクセス時間を有するようDRAMメモリを設計することができる。このより速いアクセス時間は、メインメモリよりも小さなDRAMチップを使用し、DRAMとの間でのデータ転送に使用されるピンの数を増やし、及びDRAMチップが動作する周波数を高くすることにより、達成される。かかる技術を用いる（すなわち転送帯域幅を増大させる）ことにより、約15ナノ秒でキャッシュラインにアクセスし、及び次の15ナノ秒内に転送を完了するように、DRAMチップを設計することができる。

【００３２】

（一次キャッシュと比較して）二次キャッシュのサイズが大きくなること及びそのレイテンシが長くなることの両方により、二次キャッシュからの複数の未達成のデータ要求を扱うための方法が必要となる。例えば、２ナノ秒毎に要求が受信される可能性があり、１つの要求を完遂するのに平均15ナノ秒を要する場合に、以前の要求を処理する間に多数の追加の要求を受信することが可能であるものと仮定する。従来技術のシステムは、SRAM二次キャッシュへの多数の要求をシーケンシャルに扱ってきたが、より大きなDRAM二次キャッシュ構造を使用するためには一層強力な手法が必要となる。なお、本書では、１つのキャッシュラインを要求する単一のプロセッサについて説明するが、その構造は、二次キャッシュを共有する多数のプロセッサにも等しく当てはまるものである。

【００３３】

図１は、２つのキュー、すなわち、読み出しキュー101及び書き込みキュー102を含む、二次キャッシュ構造100を示している。タグパイプライン103及びタグRAM104は、要求されたキャッシュラインが二次キャッシュ内に存在するか否かを判定するために使用される。タグパイプライン103はまた、二次キャッシュに書き込まれることになる新たなキャッシュラインのための余地を作成するよう動作する。キャッシュラインが二次キャッシュ内に存在する場合には、その要求はタグパイプライン103により読み出しキュー101に送信され、次いで該読み出しキュー101が該要求に基づいて動作する。次いで読み出しキュー101はキャッシュラインをCPUに供給する。また、キャッシュラインが存在しない場合には、その要求はタグパイプライン103によりマルチプレクサ106を介してバスユニット105に送信

される。バスユニット105から戻るキャッシュラインはバスリターンバッファ107を通過し、マルチプレクサ108を介してプロセッサ及び／又は一次キャッシュ109に送信される。バスユニット105から戻るキャッシュラインはまた、同じキャッシュラインを後に検索するためのアクセス時間を短縮するために、二次キャッシュに格納されることができる。キャッシュアーキテクチャによっては、タグパイプライン103及びタグRAM104がCPUからの操作を自動的かつシーケンシャルに処理する。これは、データを供給するために必要とされるキュー動作を見えなくする。読み出しキュー101及び書き込みキュー102の使用は、単なる例示にすぎず、本発明を限定するものではないことに留意されたい。

【００３４】

好ましい実施形態では、タグRAM104は、65,536行（インデックス）及び4列（ウェイ）を含み、DRAM110内のキャッシュの位置を判定するために使用される。一次キャッシュから要求が受信されるとき、タグパイプライン103は、タグRAM104にアクセスするために使用されるインデックスを計算する。好ましい一実施形態では、メインメモリをアドレス指定するために44ビット(0～43)が使用される（0は最上位ビット、43は最下位ビット）。各キャッシュラインが128バイトを含むので、下位7ビット(37～43)は使用されず、省略することができる。残りビットのうちの16ビット(21～36)は、タグRAM104並びにDRAM110の両方のためのインデックス（図２の符号205）を計算するためにタグパイプライン103により使用される。残りのビット、すなわち、ビット0～20は、「タグ」と呼ばれ、タグRAM104の適当な部分に格納される。タグRAM104に格納されるビット、並びに該ビットが格納される場所は、二次キャッシュ内に所望のキャッシュラインが存在するか否かを判定するためにタグパイプライン103により使用される。図１はDRAMを使用するが、SRAMを含む他のあらゆるタイプのメモリを使用して本発明を実施することが可能であることに留意されたい。

【００３５】

キャッシュシステムが要求を処理するために必要とされるステップ又はステージは通常は、キャッシュの内容を判定するために使用される読み出しステージ、１つ又は２つ以上の動作ステージ、及びキャッシュの状態を更新するために使用される書き込みステージからなる。（１つ又は２つ以上の）動作ステージは、キャッシュビットステータスの特定、タグRAMにライトバックするためのデータの生成、誤り訂正ロジック、タグRAM104と制御ロジックとの間の物理的な距離のカバー、又はそれらと類似した事柄といった様々な理由で使用することが可能である。キャッシュアーキテクチャによっては、自動的かつシーケンシャルに要求を処理し、この場合には、キャッシュ要求の全てのステージは、その後続の要求の如何なる部分も開始される前に完了される。動作を自動処理することは、ステージをパイプライン化するよりも効率が悪い。しかし、パイプラインの場合には、所与のトランザクションがタグRAM104への対応する書き込みステージを完了していないときに、「旧い」データが、後続のトランザクションに関する読み出しステージ中に読み出される可能性がある。

【００３６】

図２は、タグRAM104の好ましい一実施形態を示したものであり、キャッシュラインが二次キャッシュ内に存在するか否かを判定するために検査される４つのウェイ201～204が含まれている。

【００３７】

パイプラインは、プロセス又は作業を複数のクロックサイクルにわたって分散させるために使用される一連のラッチから構成することが可能であることに留意されたい。図１のタグパイプライン103及びタグRAM104は、図１のシステムの残りの部分に対して高速キャッシュの特徴を示している。タグパイプライン103により受信されるトランザクションは、それらがCPUから受信された順番に処理される。

【００３８】

タグRAMのための読み出し動作及びそれに関連する書き込み動作が幾つかのステージ（及びそれに対応するサイクル）により分離される構成では、これらの読み出し及び書き込

みをアトミック動作として結合する結果として無駄なサイクルが生成されることになる。読み出し及び書き込み動作は複数のステージにより分離することが可能である。これは、タグRAM104（図１）及びキャッシュロジックが物理的に分離され、その結果として、アクセス時間が比較的長くなり、又はデータ精度を確保するために誤り検出訂正（ECC）を含む構成になるからである。好ましい実施形態では、タグRAM104の読み出し及び書き込みは、４つの中間サイクルにより分離される。この構成では、読み出し及び書き込みトランザクションがアトミックトランザクションである必要があるため、後続の動作を４つの中間サイクルにわたって保留状態すなわち待機状態にすることが必要となる。これら４サイクル中に動作が生じることを可能にすることにより２つの問題が生じる可能性がある。第１に、プロセッサ又はメモリ位置が、読み出しステージ間であって第１のトランザクションのための後続の書き込みステージの前に第２の読み出しを実行することにより、旧いデータを読み出す可能性がある。第２に、読み出しが行われた後であってそれに対応する書き込みが行われる前に、以前の関係のない書き込みが生じる可能性がある。かかる２つの場合の結果として、旧いデータがタグRAM104から読み出されることになる。本発明はバイパスロジックを使用することにより旧いデータを使用しないようにする。該バイパスロジックは、旧いデータを読み出すトランザクションに対して、更新された情報を供給する。

【００３９】

図３は、３回の読み出し及びそれに関連する３回の書き込みを含む時間線を示している。キャッシュタグにアクセスする度、読み出し及びそれに対応する書き込みが存在する。時刻0で、インデックス「A」の読み出し（R_A、キャッシュアレイ）又はタグRAM104が生じ、時刻5で、インデックスAの書き込み（W_A）が生じる。時刻1,2,3,4に対応する４つの中間サイクルは、一層長いアクセス時間又はECCを可能とするために含まれる。このプロセス中に、情報がキャッシュ内に含まれているか否かについての判定が行われることに留意されたい。このステップは、時刻0で読み出されたタグと、対象となるキャッシュラインについての要求との比較を含むことが可能である。キャッシュラインが存在すると判定された場合、及びタグが要求と一致する場合には、キャッシュラインが存在し、キャッシュから取り出されることになる。時刻5に関連するサイクルは、タグRAM書き込みが生じる時点に対応し、タグRAM104内のキャッシュラインのあらゆる変更を反映させるようアレイが実際に更新されるされるときである。同様に、R_Bは、時刻2で生じ、５サイクル後（読み出し及び書き込み間に４サイクルが存在することが可能）の時刻7で、対応するW_Bが生じる。時刻4ではR_Cが生じ、時刻9ではW_Cが生じる。このシーケンスでは、時刻1,3,6,8に対応するサイクルはデッドサイクルである。A,B,Cが同じインデックスに対応しない限り、図３では旧いデータは使用されないことに留意されたい。また、A,B,Cが同じインデックスに対応する場合には、読み出しB及び読み出しC中に旧いデータが読み出されることになる点に留意されたい。これは、該読み出しB及び読み出しCの両者が、時刻0におけるR_Aの後であって時刻5におけるW_Aの前に生じるからである。

【００４０】

図４は、旧いデータがプロセッサにより使用されないようにするための従来のすなわち「ブルートフォース（強引な）」手法を示している。N－ウェイアソシエイティブキャッシュにおけるブルートフォース手法は、A,B,Cがそれぞれ同じインデックスに対応する際にW_A（図３参照）からR_B,R_Cの両者へ情報をバイパスするものである。本発明の好ましい一実施形態では、タグRAM104は120ビット長を有する。ブルートフォース手法では、タグRAM104のビットがそれぞれバイパスされる必要があり、その結果として、該バイパスのために120本の物理的なワイヤが必要となる。より多くのウェイを含むタグRAMの場合には更に多くのワイヤが必要となることに留意されたい。ブルートフォース手法における実際のワイヤの数は、タグRAMに格納されるタグビットの数と、タグRAMに格納されるキャッシュラインに関する状態の数との関数である。

【００４１】

図５は、本発明のバイパスロジックが使用されるべき時点を判定するための判定プロセスを示すフローチャートである。ステップ501で、所与のインデックスについてのキャッ

シュタグが読み出される。ステップ502で、それと同じインデックスに対する先行する古い書き込み（図３のW_A）が存在するか否かに関する判定が行われる。この場合、古い書き込みは、関連する読み出しの後であって（４つのデッドサイクルが許容される）５サイクル後の対応する書き込みの前に生じる書き込みであると定義する。同じインデックスに対する先行する古い書き込みが存在する場合には、ステップ503で、特定のウェイについての書き込みデータが確実に使用されるようにする。同様に、ステップ504で、同じインデックスに対する古い書き込みであって一層新しいものが存在するか否かについて第２の判定が行われる。一層新しい古い書き込み（図３のW_B）が存在する場合には、ステップ505で、特定のウェイについての書き込みデータが使用される。ステップ506で、所与のインデックスについてのキャッシュタグがタグRAM104に書き込まれる。ステップ502,504において、中間に生じる書き込みが存在しなかった場合には、キャッシュタグがタグRAM104の適当な部分に書き込まれる。他の幾つかのトランザクションに関し、中間で生じる書き込みが同じインデックスについてステップ502,504の何れかで生じた場合には、該中間で生じた書き込みからの情報を使用しなければならない。これは、そのデータが、キャッシュタグから以前に読み出されたデータに取って代わるデータであるからである。好ましい実施形態では、情報は４ウェイアレイのうちの１ウェイに書き込まれ、換言すれば、１回の書き込みは１ウェイの内容しか変更させないことに留意されたい。フローチャート500は、同じインデックスに対する古い書き込みに関する２回の検査を含む（ステップ502,504）。これは、この好ましい実施形態では、最初の読み出しと５サイクル後の次の書き込みとの間に２回の書き込みが生じ得るからである。他の実施形態では、古い書き込みが存在するか否かを判定するために必要とされる検査は、より多い場合も、より少ない場合もあることに留意されたい。読み出しとそれに関連する書き込みとの間の各サイクル毎に古い書き込みに関する検査が別個に必要となる可能性がある。また、複数の古い書き込みに関する検査を１回しか行わないことも可能であり、複数の書き込みが生じる場合には、以前の書き込みの優先順位付けを行って、どの書き込みが正しい情報を含むかを判定することが可能であることに留意されたい。再び図５を参照する。ステップ502,504の判定が何れも肯定された場合には、ステップ505で受信される書き込みデータが、ステップ503で受信される書き込みデータを上書きすることになる。

【００４２】

図６Ａ及び図６Ｂは、本発明のバイパス回路の一実施形態を含むパイプラインを示している。符号A～Fは、図６Ａと図６Ｂとの間の電気的な接続を示すために用いたものであることに留意されたい。参照点601で、回路により要求が受信される。該要求は、バスユニット自体からのもの、又はレベル1（一次）キャッシュから受信されたものである可能性がある。この好ましい実施形態では、該要求は、レベル２キャッシュが所与の動作を実行するための要求を表すものである。図６Ａ及び図６Ｂは、複数のラッチ603と、それに関連するクロック信号604とを含む。ラッチ603はトランスペアレントラッチであり、関連するクロック信号604がオン（HIGH）であるときにラッチ603への入力が該ラッチ603の出力に現れる。一連のラッチのうちの後続のラッチ605は、逆相のクロック信号606を使用する。ラッチ603,604のような２つの隣接するラッチは、協動してフリップフロップとして動作する。図６は実質的にパイプラインであり、時間と共に要求が該パイプラインに流れる。クロックの各位相で、要求は後続のラッチへと進む。参照点607で、パイプラインは、タグパイプラインセクション608とインデックスパイプラインセクション609とに分かれている。

【００４３】

インデックスパイプラインセクション609の参照点610において、インデックスを用いてタグRAM104（図１）がアドレス指定される。タグRAM読み出し611は、タグRAM104にアドレスを提供して参照点612においてデータを提供するために、フルサイクル（２つのラッチに相当）を必要とする。参照点612におけるデータは、ラッチ615,616を通過して、ECC検出ロジック613の入力に供給される。該ECC検出ロジック613の出力はラッチ617の入力に供給され、該ラッチ617の出力はECC訂正ロジック614の入力に与えられる。参照点618におい

て、該ECC訂正ロジック614の出力で４つのウェイについてのタグが利用可能となる。好ましい一実施形態では、全てのECCが、１グループとしての４ウェイの全てにわたり実行される。この手法は、ECCに必要となるビットの数を削減するものとなる。

【００４４】

タグRAM読み出し611の入力にアドレスが加えられるのと同時に、時刻の異なる２つのインデックスがコンパレータ619の入力に加えられるまで、インデックスが複数のラッチを通過する。コンパレータ619は、参照点620のインデックスと参照点621のインデックスとを比較する。参照点621に存在するインデックスは、参照点620に存在するインデックスとは完全に２クロック状態だけ異なる。参照点621において利用可能なインデックスを４つの更に別のラッチに通過させることにより、第１の受信された要求が参照点621において利用可能となり、第２の受信された要求（第１の受信された要求の完全に２クロック後に受信された要求）が参照点620に存在する。参照点620におけるインデックスが参照点621におけるインデックスと等しい場合には、後行するバイパス条件が生存する。

【００４５】

コンパレータ624の参照点622において利用可能なインデックスは、コンパレータ619の参照点620において利用可能なインデックスと同じである。しかし、コンパレータ624の参照点623において利用可能なインデックスは、参照点622において利用可能なインデックスと比べて８つ多くのラッチを通過している。コンパレータ624内の比較は、完全に４クロック状態だけ離れて受信された２つの要求の２つのインデックスからなる。参照点622におけるインデックスが参照点623におけるインデックスに等しい場合には、先行するバイパス条件が生存する。参照点625では、その点において利用可能なインデックスがタグRAM書き込み626の入力に加えられ、これもまた完全な１サイクルを必要とする（２つのラッチのために必要となる時間に等しい）。

【００４６】

図６Ｂは図６Ａの続きである。図６Ｂはウェイバイパス回路629を含む。該ウェイバイパス回路629は、読み出しバイパスマルチプレクサ630と、書き込みマルチプレクサ631と、コンパレータ632とを含む。キャッシュ書き込みマルチプレクサ631は、更新されたデータがキャッシュに書き込まれることを確実にする。ウェイバイパス回路629は、Nウェイ毎に繰り返される。読み出しバイパスマルチプレクサ630は３つの入力、すなわち、17個のラッチにより遅延された（9.5サイクル前に受信された）要求入力を表す第１の入力と、13個のラッチにより遅延された（6.5サイクル前に受信された）第２の入力と、ECC訂正ロジック614からのウェイn読み出しデータ（WAY n RDDATA）とを含む。バイパス条件が存在しない場合には、読み出しバイパスマルチプレクサは、WAY n RDDATA値を無視することになる。読み出しバイパスマルチプレクサ630のための選択信号入力はバイパス制御ロジック633に電気的に接続される。各ウェイ毎に一意の信号が使用される。バイパス制御ロジック633のための入力は、後行バイパス信号（ラッチを通過したコンパレータ619の出力）及び先行バイパス信号（ラッチを通過したコンパレータ624の出力）である。更に、バイパス制御ロジック633は、犠牲制御ロジック634からの入力も受信する。読み出しバイパスマルチプレクサ630のための選択信号入力は、キャッシュに対する１回の書き込みが生じた際にキャッシュラインのNウェイのうちの１つだけが選択されることを確実にするよう構成される。バイパス処理が必要とされない場合には、ECC訂正ロジック618からのデータを読み出しバイパスマルチプレクサ630のWAY n RDDATAに加える。参照点618において示されるように、４つのウェイの各々毎に出力が利用可能となる。

【００４７】

犠牲制御ロジック634は、入力された要求を、キャッシュ内に既に存在する情報と比較するために使用される。好ましい一実施形態では、所与の時点でキャッシュラインの１つのコピーのみがキャッシュ内に存在することが可能となる。犠牲制御ロジック634は、３つの入力、すなわち、トランザクション情報入力、犠牲一致入力、及び置換状態入力を含む。トランザクション情報入力は、実行されるトランザクションの種類を特定する。トランザクションの中にはウェイの識別を含むものもある。犠牲一致入力は、キャッシュ内に

ある情報に対して要求タグを比較し、それらが一致する場合には、そのウェイが犠牲にならなければならない。犠牲一致入力は、キャッシュ内の特定のラインをフラッシュ命令でフラッシュするために使用することが可能である。好ましい一実施形態では、命令についてはコヒーレンシが維持されないこと、及びデータラインをキャッシュに書き込むことが望まれる時点でキャッシュ内に命令ラインが存在し得ることに留意されたい。かかる状況では、命令ラインにデータラインが上書きされることになり、そのウェイは該命令ラインに対応するウェイとして識別される。置換状態入力は、擬似乱数生成アルゴリズムといった犠牲選択アルゴリズムからの出力である。置換状態入力は、どのウェイを置換すべきかを判定するために犠牲制御ロジック634によって使用される状態を含む。犠牲制御ロジック634は、Nウェイの全てについて出力が１つのみ存在するよう構成される。該犠牲制御ロジック634の出力は、バイパスキャッシュ書き込みマルチプレクサ631のための選択信号に電気的に接続される。キャッシュ書き込みマルチプレクサ631は、キャッシュを更新するために使用される。

【００４８】

図６Ｃは、図６Ｂに図示し同図に関して解説したバイパス回路の代替的な実施形態を示したものである。図６Ｃに示すように、要求タグパイプラインのラッチは、ウェイバイパス回路636の各実例で、すなわち本開示の４ウェイの各々毎に、繰り返される。このラッチの繰り返しは、図６Ｂに示す要求タグパイプライン構成の数多くの共有されるラッチと、図６Ｃの要求パイプラインのより少数の（例えば９個の共有されるラッチに対して２個の）共有されるラッチとの間で必要とされる接続の数を削減する。共有されるラッチの数が削減される結果、各ウェイ専用の別個の組をなすラッチのために必要となるラッチの繰り返し数の増加の代償として、配線数が削減される。全てのウェイにより共有され及び共通である第１のグループのラッチ637と、個々のウェイのための別個の組をなすラッチ638を有する第２のグループとからなるハイブリッド構成は、特定の環境に特に適したものとなる。例えば、該ハイブリッド構成は、多数のウェイ（例えば少なくとも８ウェイ）を有するシステムに、又は読み出しと書き込みとの間に多数のステージが存在する場合、すなわち多数のバイパス経路又はバスが存在し得る場合に、有利に適用することが可能である。例えば、この例は、２つのバイパスバス（例えば後行及び先行バイパスバス）しか示していないが、より多くのバイパスバスを有する構成もまた、要求タグパイプラインを形成する共有され繰り返されたラッチの分割に特になじみ易いものとなる。

【００４９】

図７は、本発明の一実施形態によるキャッシュの動作方法を示すフローチャートである。ステップ701で開始し、ステップ702で第１の読み出し動作を実行してタグメモリからタグ及び状態情報を読み出す。この読み出しは、図３に示すR_Aの読み出しといった第１の情報の取り出しに相当する。ステップ703で、第２の読み出し動作を実行し、タグメモリからタグ及び状態情報を読み出す。この読み出しは、図３のR_Bの取り出しに相当する。R_Bは本書では第５の情報と称する場合もある。ステップ704で、第３の読み出し動作を実行して、再びタグメモリからタグ及び状態情報を読み出す。この読み出しはR_Cに相当し、該情報を第２の情報と称する場合もある。ステップ705は、ステップ702で実行した第１の読み出し動作の結果を、タグメモリに書き込まれるタグ及び状態情報で変更することを含み、この書き換えられた情報を本書では第３の情報と称する。同様に、ステップ706で、第２の読み出し動作（すなわちステップ703）について変更を実行し、タグ及び状態情報を第６の情報としてタグメモリに書き込む。ステップ707で、第３の読み出し（すなわちステップ704）に関するアドレスと、第１の書き込み（すなわちステップ705）を実行するために使用されたアドレスとの比較を実行する。第３の読み出しのアドレスが第１の書き込みのアドレスと等しい場合には、ステップ708を実行し、第１の書き込みにより変更されたウェイを更新することによりデータを選択的に置換する。同様に、ステップ709で、第３の読み出しのアドレスと第２の書き込みに関するアドレスとの比較を実行する。この場合も、それらのアドレスが一致する場合には、ステップ710で、選択的な置換ステップを実行して、第２の書き込み（すなわちステップ706）により変更されたウェイを更新する。

最後に、ステップ711で、第３の読み出しデータ（すなわちステップ704）を必要に応じて変更して、タグ及び状態情報をタグメモリに書き込む。この情報は本書では第４の情報と称する場合もある。該プロセスは、ステップ712で終了するものとして示したが、これと同じプロセスが、後続のトランザクションがタグパイプライン内を流れる際にその各トランザクション毎に実行されることになる。後続の各トランザクションは、図７に示すタグ及び情報の時間的にインクリメントされた繰り返しで動作することが理解されよう。例えば、ステップ702,703,704で第１～第３の読み出しを処理した後、次の後続する繰り返しで、第２の読み出しが第１の読み出しになり、及び第３の読み出しが第２の読み出しになる一方、第３の読み出しが新たなトランザクションを処理する。

【図面の簡単な説明】

【００５０】

【図１】タグパイプライン及びタグRAMを含む二次キャッシュ構造を示すブロック図である。

【図２】図１の２次元タグRAMを示す説明図である。

【図３】３回の読み出し及び３回の関連する書き込みを含む時間線である。

【図４】旧いデータが使用されないことを確実化するための従来の「ブルートフォース（力ずくの）」手法を示す説明図である。

【図５】本発明のバイパスロジックが使用されるべきときを判定するための決定プロセスを示すフローチャートである。

【図６Ａ】本発明のバイパス回路の一実施形態を含むパイプラインを示す説明図である。

【図６Ｂ】本発明のバイパス回路の一実施形態を含むパイプラインを示す説明図である。

【図６Ｃ】本発明のバイパス回路の代替的な実施形態を含むパイプラインを示す説明図である。

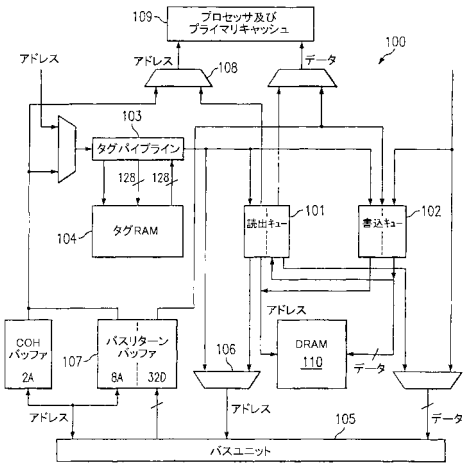
【図７】本発明の一実施形態によるキャッシュ動作方法を示すフローチャートである。

【符号の説明】

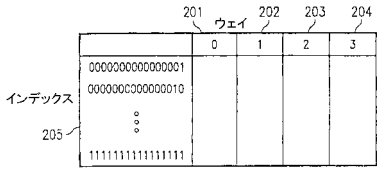
【００５１】

100　二次キャッシュ構造
101　読み出しキュー
102　書き込みキュー
103　タグパイプライン
104　タグRAM
105　バスユニット
106　マルチプレクサ
107　バスリターンバッファ
108　マルチプレクサ
109　プロセッサ及び／又は一次キャッシュ
110　DRAM

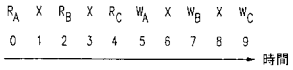
【図１】



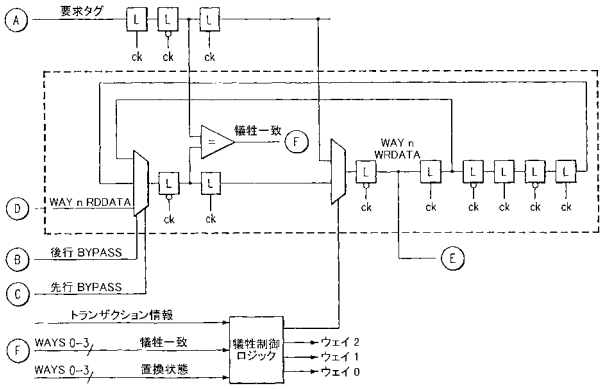
【図２】



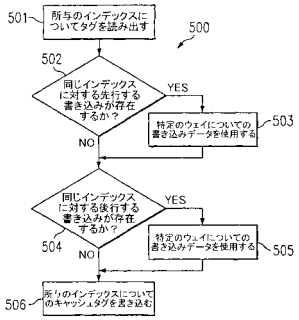
【図３】



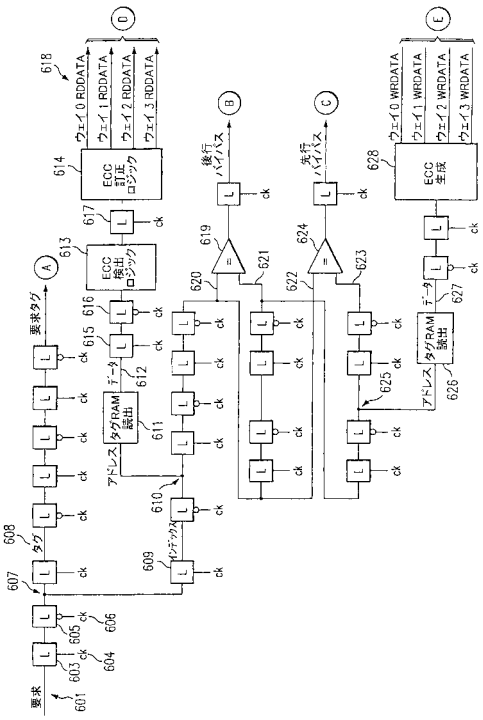
【図４】



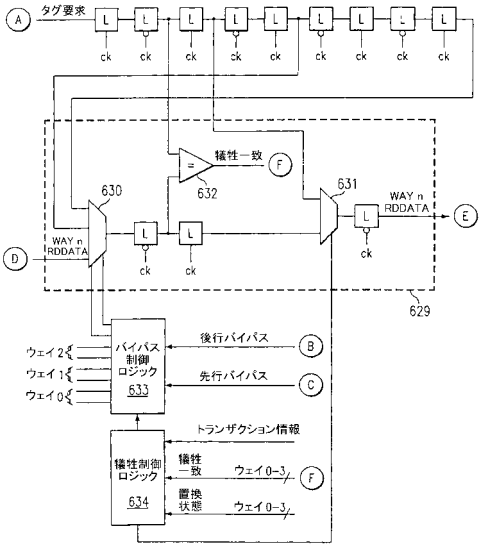
【図５】



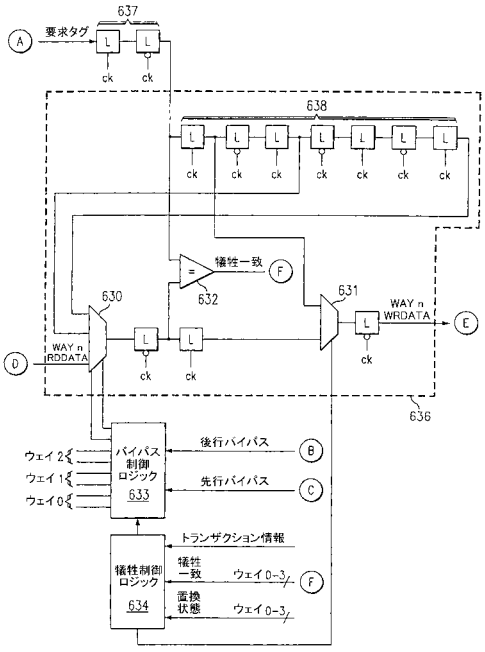
【図６Ａ】



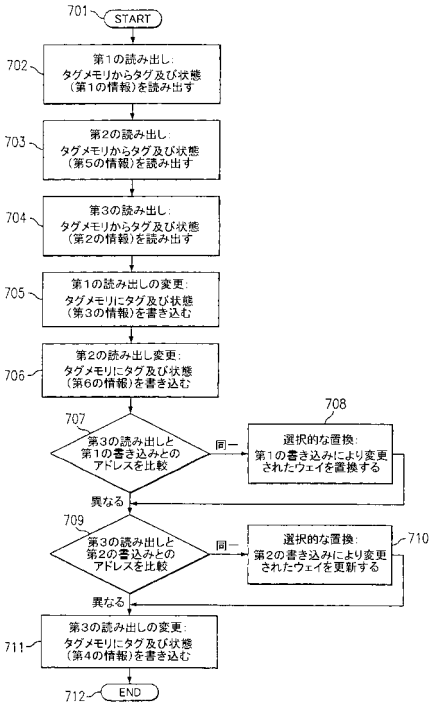
【図６Ｂ】



【図６Ｃ】


637
要求タグ
A
L
ck
638
=
犠牲一致
F
632
630
631
WAY n WRDATA
E
WAY n RDDATA
D
636
ウェイ2
ウェイ1
ウェイ0
バイパス制御ロジック
633
後行バイパス
B
先行バイパス
C
トランザクション情報
犠牲制御ロジック
634
犠牲一致
ウェイ0-3
置換状態
ウェイ0-3

【図７】


701
START
702
第1の読み出し:
タグメモリからタグ及び状態
(第1の情報)を読み出す
703
第2の読み出し:
タグメモリからタグ及び状態
(第5の情報)を読み出す
704
第3の読み出し:
タグメモリからタグ及び状態
(第2の情報)を読み出す
705
第1の読み出しの変更:
タグメモリにタグ及び状態
(第3の情報)を書き込む
706
第2の読み出し変更:
タグメモリにタグ及び状態
(第6の情報)を書き込む
707
第3の読み出しと
第1の書き込みとの
アドレスを比較
同一
708
選択的な置換:
第1の書き込みにより変更
されたウェイを置換する
異なる
709
第3の読み出しと
第2の書込みとの
アドレスを比較
同一
710
選択的な置換:
第2の書き込みにより変更
されたウェイを更新する
異なる
711
第3の読み出しの変更:
タグメモリにタグ及び状態
(第4の情報)を書き込む
712
END

フロントページの続き

(72)発明者　ロバート・エフ・クリック
アメリカ合衆国コロラド州８０５２５，フォートコリンズ，ウォーターストーン・コート・３００３
(72)発明者　デュアン・エイ・ウィーンズ
アメリカ合衆国コロラド州８０５２６，フォートコリンズ，ストダード・７４２

Ｆターム(参考) 5B005 JJ12 MM01 NN12 NN31 NN74 NN75 TT02