【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第３区分
【発行日】平成23年2月24日(2011.2.24)

【公表番号】特表2003-514299(P2003-514299A)
【公表日】平成15年4月15日(2003.4.15)
【出願番号】特願2001-536680(P2001-536680)
【国際特許分類】
Ｇ０６Ｆ　12/08　　(2006.01)
Ｇ０６Ｆ　 9/38　　(2006.01)
【ＦＩ】
Ｇ０６Ｆ　12/08　　５０３Ｅ
Ｇ０６Ｆ　12/08　　５０７Ｚ
Ｇ０６Ｆ　12/08　　５２５Ｅ
Ｇ０６Ｆ　 9/38　　３５０Ａ
Ｇ０６Ｆ　 9/38　　３５０Ｂ

【誤訳訂正書】
【提出日】平成22年12月21日(2010.12.21)
【誤訳訂正１】
【訂正対象書類名】明細書
【訂正対象項目名】全文
【訂正方法】変更
【訂正の内容】
【書類名】　　明細書
【発明の名称】　　インデックスおよび任意選択的ウェイ一致に基づいてデータをフォワードするストアバッファ
【特許請求の範囲】
　　【請求項１】　各々が、（ｉ）ストアメモリ動作のストアアドレスの少なくともインデックス部と、（ii）前記ストアメモリ動作がデータキャッシュ内でヒットしているかどうかを示すヒット表示と、（iii）前記ストアメモリ動作に対応するストアデータとをストアするように構成された複数のエントリを含むバッファと、
　前記バッファに結合された回路とを含み、前記回路は、（ｉ）前記データキャッシュを探索するロードメモリ動作のロードアドレスのインデックス部と、（ii）前記ロードメモリ動作が前記データキャッシュ内でヒットしているかどうかを示すロードヒット信号とを受けるように結合されるとともに、前記複数のエントリのうちの第１のエントリにストアされた前記インデックス部が前記ロードアドレスの前記インデックス部と一致しかつ前記第１のエントリ内の前記ヒット表示がヒットを示し、かつさらに前記ロードヒット信号がヒットを示すことに応じて、前記第１のエントリのストアデータが前記第１のエントリからフォワードされるように構成される、装置。
　　【請求項２】　前記回路は、前記ロードアドレスの前記インデックス部が前記ストアアドレスの前記インデックス部と一致することに応じて、フォワード信号をアサートするように構成される、請求項１に記載の装置。
　　【請求項３】　前記回路は、さらに前記ヒット表示がヒットを示すことに応じて、前記フォワード信号をアサートするように構成される、請求項２に記載の装置。
　　【請求項４】　前記回路は、その後前記ロードヒット信号を受け、前記ロードヒット信号が前記ロードが前記データキャッシュ内でミスしていると示すことに応じて、フォワード取消信号をアサートするように構成される、請求項３に記載の装置。
　　【請求項５】　前記複数のエントリの各々は、前記データキャッシュにおいて前記ストアメモリ動作がヒットするウェイを示すストアウェイ表示をストアするように構成され

る、請求項４に記載の装置。

【請求項６】　前記回路はさらに、前記データキャッシュにおいて前記ロードメモリ動作がヒットするウェイを示すロードウェイ表示を受けるように結合されるとともに、前記第１のエントリにストアされた前記ストアウェイ表示が前記ロードウェイ表示と一致しないことに応じて前記フォワード取消信号をアサートするように構成される、請求項５に記載の装置。

【請求項７】　前記複数のエントリの各々は、前記データキャッシュにおいて前記ストアメモリ動作がヒットするウェイを示すストアウェイ表示をストアするように構成される、請求項１に記載の装置。

【請求項８】　前記回路はさらに、前記データキャッシュにおいて前記ロードメモリ動作がヒットするウェイを示すロードウェイ表示を受けるように結合されるとともに、さらに前記ロードウェイ表示および前記第１のエントリにストアされた前記ストアウェイ表示が一致することに応じて、前記ストアデータがフォワードされるように構成される、請求項７に記載の装置。

【請求項９】　データキャッシュと、

前記データキャッシュに結合されたロード／ストアユニットとを含み、前記ロード／ストアユニットは、複数のエントリを含むバッファを含み、前記複数のエントリは各々、（ｉ）ストアメモリ動作のストアアドレスの少なくともインデックス部と、（ii）前記ストアメモリ動作が前記データキャッシュ内でヒットしているかどうかを示すヒット表示と、（iii）前記ストアメモリ動作に対応するストアデータとをストアするように構成され、前記ロード／ストアユニットは、前記データキャッシュをロードアドレスを用いて探索し、それに応じて前記データキャッシュからヒット信号を受けるように構成されるとともに、前記ロードアドレスのインデックス部が前記複数のエントリの第１のエントリにストアされた前記インデックス部と一致していることに応じてかつさらに前記第１のエントリ内の前記ヒット表示がヒットを示し前記ヒット信号がヒットを示すことに応じて、ストアデータを、前記第１のエントリからフォワードすると決定するように構成される、プロセッサ。

【請求項１０】　前記ロード／ストアユニットは、前記データキャッシュに対しフォワードデータ信号をアサートするように構成され、前記データキャッシュは、前記ストアデータを前記データキャッシュから読出したキャッシュデータの代わりにフォワードするように構成される、請求項９に記載のプロセッサ。

【請求項１１】　前記ロード／ストアユニットは、前記ロードアドレスの前記インデックス部が前記第１のエントリにストアされたインデックス部と一致することに応じて、前記フォワードデータ信号をアサートするように構成される、請求項１０に記載のプロセッサ。

【請求項１２】　前記ロード／ストアユニットは、さらに前記第１のエントリ内の前記ヒット表示がヒットを示すことに応じて前記フォワードデータ信号をアサートするように構成される、請求項１１に記載のプロセッサ。

【請求項１３】　前記ロード／ストアユニットは、前記ヒット信号がミスを示すことに応じてフォワード取消信号をアサートするように構成される、請求項１２に記載のプロセッサ。

【請求項１４】　前記フォワード取消信号を受けるように結合された１以上のリザベーションステーションをさらに含み、前記１以上のリザベーションステーションは、前記フォワード取消信号に応じて前記ロードのためにフォワードされた前記ストアデータを無効にするように構成される、請求項１３に記載のプロセッサ。

【請求項１５】　前記複数のエントリの各々は、前記データキャッシュにおいて前記ストアメモリ動作がヒットするウェイを示すストアウェイ表示をストアするように構成される、請求項１３に記載のプロセッサ。

【請求項１６】　前記データキャッシュは、前記データキャッシュにおいて前記ロードメモリ動作がヒットするウェイを示すロードウェイ表示を前記ロード／ストアユニット

に与えるように構成され、前記ロード／ストアユニットは、前記第１のエントリ内の前記ストアウェイ表示が前記ロードウェイ表示と一致しないことに応じて前記フォワード取消信号をアサートするように構成される、請求項１５に記載のプロセッサ。

【請求項１７】　前記複数のエントリの各々は、前記データキャッシュにおいて前記ストアメモリ動作がヒットするウェイを示すストアウェイ表示をストアするように構成される、請求項９に記載のプロセッサ。

【請求項１８】　前記データキャッシュは、前記データキャッシュにおいて前記ロードメモリ動作がヒットするウェイを示すロードウェイ表示を前記ロード／ストアユニットに与えるように構成され、前記ロード／ストアユニットは、さらに前記第１のエントリにストアされた前記ストアウェイ表示が前記ロードウェイ表示と一致することに応じて、ストアデータを前記第１のエントリからフォワードすると決定するように構成される、請求項１７に記載のプロセッサ。

【請求項１９】　方法であって、

データキャッシュをロードアドレスを用いて探索するステップと、

前記ロードアドレスのインデックス部をバッファにストアされたストアアドレスのインデックス部と比較するステップと、

前記ストアアドレスに対応するストアデータを、前記ロードアドレスに対応するロードメモリ動作のためにフォワードするステップとを含み、前記フォワードするステップは、前記比較により前記ロードアドレスの前記インデックス部が前記ストアアドレスの前記インデックス部と一致していると判断されたことに応じてかつさらに前記ロードアドレスおよび前記ストアアドレス双方がデータキャッシュ内でヒットしていることに応じて行なわれる、方法。

【請求項２０】　前記フォワードするステップは、前記比較により前記ロードアドレスの前記インデックス部が前記ストアアドレスの前記インデックス部と一致していると判断されたことに応じて、フォワードデータ信号をアサートするステップを含む、請求項１９に記載の方法。

【請求項２１】　前記アサートするステップは、さらに前記ストアアドレスが前記データキャッシュにおいてヒットしていることに応じて行なわれる、請求項２０に記載の方法。

【請求項２２】　前記フォワードするステップはさらに、前記ロードアドレスが前記データキャッシュ内でミスしていることに応じてフォワード取消信号をアサートするステップを含む、請求項２１に記載の方法。

【請求項２３】　前記方法はさらに、前記バッファにストアされたストアウェイ表示を前記ロードアドレスに対応するロードウェイ表示と比較するステップを含み、前記ストアウェイ表示は、前記データキャッシュにおいて前記ストアアドレスがヒットするウェイを示し、前記ロードウェイ表示は、前記データキャッシュにおいて前記ロードアドレスがヒットするウェイを示し、前記方法はさらに、前記比較によりストアウェイ表示が一致しないと判断されたことに応じて、前記フォワード取消信号をアサートするステップを含む、請求項２２に記載の方法。

【請求項２４】　前記方法はさらに、前記バッファにストアされたストアウェイ表示を前記ロードアドレスに対応するロードウェイ表示と比較するステップを含み、前記ストアウェイ表示は、前記データキャッシュにおいて前記ストアアドレスがヒットするウェイを示し、前記ロードウェイ表示は、前記データキャッシュにおいて前記ロードアドレスがヒットするウェイを示し、前記方法はさらに、さらに前記比較によりストアウェイ表示が一致すると判断されたことに応じて、前記ストアデータをフォワードするステップを含む、請求項１９に記載の方法。

【発明の詳細な説明】

【０００１】

【発明の分野】

本発明は、プロセッサの分野に関し、より特定的にはデータをストアバッファから従属

するロードのためにフォワードすることに関する。

【０００２】

【関連技術の説明】

プロセッサは典型的に、ストアメモリ動作をストアするためのバッファを用いる。このストアメモリ動作は、実行済みである（たとえばストアアドレスが既に生成されている、ストアデータがあるかもしれない）が、まだ投機的であるためメモリ（またはこのプロセッサが用いるデータキャッシュ）に記憶させる準備が整っていないものである。本明細書で用いている「メモリ動作」という用語は、プロセッサおよびメモリ間のデータ転送（この転送はキャッシュ内で行なわれる可能性がある）を指定する動作を意味する。ロードメモリ動作は、メモリからプロセッサへのデータ転送を指定し、ストアメモリ動作はプロセッサからメモリへのデータ転送を指定する。本明細書においては、ロードメモリ動作をより簡潔に「ロード」と呼ぶこともあり、同様に、ストアメモリ動作を「ストア」と呼ぶこともある。メモリ動作は、プロセッサが用いる命令セット次第で、メモリオペランドに直接アクセスして定められた機能（たとえば算術演算、論理など）を果たすための命令内で暗示されていることもあり、データ転送のみを行なう明示命令のこともある。一般的に、メモリ動作は、このメモリ動作の１以上のオペランドから生成されたアドレスを介して関係する記憶場所を指定する。本明細書ではこのアドレスのことを一般的に「データアドレス」と呼ぶ、または、（対応するメモリ動作がロードであれば）ロードアドレスと呼び、（対応するメモリ動作がストアであれば）ストアアドレスと呼ぶ。他方、命令そのもののメモリ内での場所を示すアドレスは「命令アドレス」と呼ぶ。

【０００３】

ストアは、後に続くロードが実行される際にバッファ内で待ち行列に入れられていることがあり、そのため、典型的にプロセッサはバッファを検査して、ロードが読出す１以上のバイトを更新するストアがバッファ内で待ち行列に入れられているかどうか判断する（すなわちこのロードがストアに従属するのかまたはストアに「ヒット」するのか判断）。一般的には、ロードアドレスをストアアドレスと比較してロードがストアにヒットしているかどうか判断する。ヒットが検出されれば、このロードのためにストアデータをキャッシュデータの代わりにフォワードする。したがって、ヒットを、キャッシュからデータにアクセスするのに要する時間と同じまたはそれより短い時間で検出することが望ましい。

【０００４】

ロードの待ち時間（たとえば、あるロードを実行してからこのロードによって読出されたデータを使用できるようになるまでの時間）を最短にすることは、多くのプロセッサの性能にとって重要である。あいにく、アドレスの比較は、アドレスのビット数が比較的多い（たとえば３２ビット、または３２ビットを超えて６４ビットまでが一般的になりつつある）ため、時間のかかる作業である。したがって、ロードがバッファ内のストアにヒットしているかどうか判断するのに要する時間を短縮化することが、プロセッサの性能を高めることになる。なぜなら、この短縮化がロードの待ち時間の短縮になり得るからである。その代わりとしては、アドレスを比較するのにかかる時間が減少すれば、所与のサイクル時間および所与のロード待ち時間に対するタイミングの制約を満たしやすくなる。

【０００５】

仮想アドレス指定およびアドレス変換を用いると、ロードアドレスをバッファ内のストアアドレスと照合する間に経過する時間を短縮するというさらなる問題が生じ得る。仮想アドレス指定を用いる場合、ロードおよびストアの実行により生成されるデータアドレスは、（たとえばページング変換方式によって）物理アドレスに変換される仮想アドレスである。複数の仮想アドレスが所与の物理アドレスに対応しているかもしれないため（「別名（エイリアス)」と呼ばれる）、ロードおよびストアの物理データアドレスを比較して確実にバッファから正確な転送を行なう（または行なわない）ようにする。不都合なことにロードの物理アドレスは、典型的に変換索引バッファ（ＴＬＢ）から生成されるため、キャッシュアクセスがほぼ完了するまで利用できないことが多く、バッファ内のストアにヒットしていることを迅速だが正確なやり方で検出するという問題がさらに悪化する。

【０００６】
【発明の概要】
上記の問題は主として、本明細書で説明するようにストアデータをロードのためにフォワードする装置により解決される。この装置は、ストアメモリ動作に対応する情報をストアするように構成されたバッファと、バッファにおいて示されたストアの１つにヒットするロードを検出する回路とを含む。より具体的には、この回路は、ロードアドレスのインデックス部をバッファにストアされているストアアドレスのインデックス部と比較する。これらのインデックスが一致しかつロードおよびストア双方がデータキャッシュ内でヒットしていれば、ロードおよびストアは同じキャッシュラインにアクセスしている。このキャッシュライン内の１以上のバイトがストアにより更新されロードにより読出されるならば、ストアデータはそのロードのためにフォワードされる。好都合なことに、この比較的小規模のロードおよびストアインデックス比較は速やかに完了する。加えて、（すべてではないにしても）ほとんどのインデックスが典型的には物理（変換されていない）ビットなので、比較は、変換されているロードアドレスよりも前に実施することができ、比較の正確性に大きな影響はない。
【０００７】
ある実施例では、上記回路は、ロードおよびストアインデックスが一致しかつストアがデータキャッシュ内でヒットしていれば、データを投機的にフォワードする。次に、ロードがキャッシュ内でヒット／ミスしていると判断されると、このフォワーディングが、ロードのヒット／ミス表示を用いて検査される。セットアソシアティブの実施例では、ロードのヒットのウェイをストアのヒットのウェイと比較してフォワーディングの正確度をさらに検査する。
【０００８】
概して、ある装置が意図されている。この装置は、バッファとバッファに結合された回路とを含む。バッファは複数のエントリを含み、複数のエントリは各々、（ｉ）ストアメモリ動作のストアアドレスの少なくともインデックス部と、（ii）ストアメモリ動作がデータキャッシュ内でヒットしているかどうか示すヒット表示と、（iii）このストアメモリ動作に対応するストアデータとをストアするように構成される。回路は、（ｉ）データキャッシュを検索するロードメモリ動作のロードアドレスのインデックス部と、（ii）ロードメモリ動作がデータキャッシュ内でヒットしているかどうか示すロードヒット信号とを受けるように結合される。この回路は、ストアデータを、複数のエントリのうち第１のエントリから、第１のエントリにストアされたインデックス部がロードアドレスのインデックス部に一致することに応じて、かつさらに、第１のエントリのヒット表示がヒットを示しロードヒット信号がヒットを示すことに応じて、フォワードするように構成される。
【０００９】
さらに、データキャッシュとデータキャッシュに結合されたロード／ストアユニットとを含むプロセッサが意図されている。ロード／ストアユニットは、複数のエントリを含むバッファを備え、複数のエントリは各々、（ｉ）ストアメモリ動作のストアアドレスの少なくともインデックス部と、（ii）ストアメモリ動作がデータキャッシュ内でヒットしているかどうか示すヒット表示と、（iii）このストアメモリ動作に対応するストアデータとをストアするように構成される。ロード／ストアユニットは、ロードアドレスを用いてデータキャッシュを探索し、それに応じてデータキャッシュからヒット信号を受けるように構成される。加えて、ロード／ストアユニットは、ストアデータを、複数のエントリのうち第１のエントリから、ロードアドレスのインデックス部が第１のエントリにストアされたインデックス部に一致することに応じて、かつさらに、第１のエントリのヒット表示がヒットを示しヒット信号がヒットを示すことに応じて、フォワードすると決定する。
【００１０】
さらに、ある方法が意図されている。データキャッシュはロードアドレスを用いて検索される。ロードアドレスのインデックス部がバッファにストアされたストアアドレスのインデックス部と比較される。ストアアドレスに対応するストアデータが、ロードアドレス

に対応するロードメモリ動作のためにフォワードされる。このフォワーディングは、比較によりロードアドレスのインデックス部がストアアドレスのインデックス部と一致すると判断されたことに応じて、かつさらに、ロードアドレスおよびストアアドレス双方がデータキャッシュ内でヒットしていることに応じて、行なわれる。

【００１１】

本発明の上記以外の目的および利点は、以下の詳細な説明を読み添付の図面を参照することによって明らかになるであろう。

【００１２】

本発明には種々の変形および代替形が可能であり、本発明の具体的な実施例が図面において例示され本明細書において詳細に説明されている。しかしながら、図面およびその詳細な説明は本発明を開示された特定の形態に限定することを意図したものではなく、反対に、前掲の特許請求の範囲において定められた本発明の精神および範囲に含まれるすべての変形、等価物および代替形を包含することを意図している。

【００１３】

【好ましい実施例の詳細な説明】

次に図１を参照して、ストア待ち行列４００、ヒット制御回路４０２ならびに比較器４０４および４０６の一実施例のブロック図が示される。この図１に示した装置は、ストアに関連する情報をストアがデータキャッシュ（および／またはメモリ）に与えられるまで保持するデータキャッシュを有するプロセッサにおいて用いることができ、さらに、ストアにヒットするロードを検出しストアデータをストア待ち行列４００からロードのためにフォワードするのに用いることができる。他の実施例も可能であり意図されている。図１の実施例では、ストア待ち行列４００は、実行されたストアに対応するストア情報を受けるように結合され、さらにヒット制御回路４０２ならびに比較器４０４および４０６に結合される。比較器４０４および４０６はさらに、ヒット制御回路４０２に結合される。ヒット制御回路４０２は、フォワード信号およびフォワード取消信号を与えるように結合される。

【００１４】

概して、図１に示した装置は、ストア待ち行列４００において示されているストアにヒットするロードを検出し、そのロードのためにストアに対応するデータを（データキャッシュからのキャッシュデータの代わりに）ストア待ち行列４００からフォワードするように構成される。この装置は、ロードアドレス全体をストア待ち行列４００にストアされたストアアドレスと比較するのではなく、ロードアドレスのインデックス部（「ロードインデックス」）をストアアドレスのインデックス部（「ストアインデックス」）と比較する。アドレスの一部を比較するため、この比較はより速やかに行なわれ、結果として、ロードがストア待ち行列４００に示されたストアにヒットするかどうか判断するのに要する時間が短縮される。ロードおよびストア双方がデータキャッシュにおいてヒットしかつインデックス部が一致すれば、ロードおよびストアはデータキャッシュ内の同じキャッシュラインにアクセスしている。データキャッシュがダイレクトマッピングの場合、ロードおよびストアは同じキャッシュラインにアクセスしている。データキャッシュがセットアソシアティブの場合、ストアのヒットのウェイおよびロードのヒットのウェイを比較して、ロードおよびストアが同じキャッシュラインにアクセスしているかどうか判断することができる。ロードがヒットでありストアがミスである（またはその逆）の場合、ロードおよびストアは同じキャッシュラインにアクセスしていない（インデックス部はどれも仮想でないと仮定）。したがって、ロードはストアにヒットしておらず、ストアデータをストア待ち行列４００からフォワードする必要はない。ロードおよびストア双方がミスの場合、ロードおよびストアは同じキャッシュラインにアクセスしているかもしれない。しかしながら、データキャッシュは、ロードがミスの場合データをキャッシュからロードのためにフォワードしておらず、したがって、ストアデータをストア待ち行列４００からフォワードする必要はない。ロードを、データキャッシュがロードが読出したキャッシュラインで満たされた後に（または満たすデータをキャッシュに書込んでいる間に）再び試みてもよく

、そのキャッシュラインへのストアがキャッシュを満たす間にヒットとなる可能性がある。こうして、再度ロードを試みる間に、ロードがストアにヒットすることが検出される可能性がある。

【００１５】

典型的に、仮想アドレスから物理アドレスへの変換は、ページの粒度に対して実施される。最下位アドレスビットは、ページに対するオフセットを形成しこの変換では変更されない。最上位アドレスビットは仮想から物理に変換される。たとえば、３２ビットの仮想アドレスおよび４キロバイトのページサイズを用いた実施例では、下位１２ビットがページオフセットであり上位２０ビットが変換される。これ以外のページサイズが意図されている。典型的には、アドレスのインデックス部の（すべてではないにしても）ほとんどの部分は、ページオフセット内にあるため、仮想から物理へのアドレス変換中に変更されることはない。したがって、ロードがストアにヒットするかどうか検出する際の正確度に対するエイリアスの影響を、減じるまたは排除することができる。さらに、仮想ロードアドレスを比較において用い、物理ストアアドレス（メモリなどに与えるために用いられる）をストア待ち行列４００にストアしてもよい。インデックス部の１以上のビットが仮想－物理変換において変更されるならば、仮想ビットもストアすればよい。したがって、仮想ロードアドレスと比較するために仮想ストアアドレスをストアするのに追加する記憶量は最小でよい（たとえば変換されかつインデックスの一部でもあるビット）。

【００１６】

図１に示した実施例を、セットアソシアティブデータキャッシュを用いたプロセッサで使用してもよい。ダイレクトマッピングデータキャッシュを用いる実施例では、ウェイ表示および関連する比較器はない。より具体的には、ストア待ち行列４００が複数のエントリを含む。たとえば、図１にはエントリ４０８Ａおよび４０８Ｂが示されており、ストア待ち行列４００はさらなるエントリ（図示せず）を含み得る。各エントリ４０８は、ストアメモリ動作に対応する情報をストアするように構成されている。ストア待ち行列４００は、ストアに対応する情報をこのストアの実行の際に受取り、この情報を、ストアがリタイア（終了処理）されてデータキャッシュおよび／またはメモリに与えられてしまうまで保持する。ここに示されている実施例では、ひとつのエントリは、有効表示（Ｖ）、ヒット表示（Ｈ）、リタイア表示（Ｒ）、アドレスタグ部（ADDR－Tag）、アドレスインデックス部（ADDR－Index）、オフセットおよびサイズ情報（Offset and Size）、ウェイ表示（Way）、およびデータ（Data）を含む。有効表示は、エントリが有効かどうか（たとえばストアがエントリ内の情報で表わされているかどうか）示す。ヒット表示は、ストアがデータキャッシュ内でヒットしているかどうか示す。リタイア表示は、ストアがリタイアされている（したがって、データキャッシュおよび／またはメモリに与えるのに適切である）かどうか示す。この有効、ヒットおよびリタイア表示に対しては適切な表示を用いればよい。たとえば、各表示は、セットされると一方の状態を示しクリアされると他方の状態を示すビットを含み得る。以下の説明（下記図５および６に示された実施例の説明を含む）では、有効、ヒットおよびリタイア表示のことを、有効、ヒットおよびリタイアビットと呼ぶ。しかしながら、他の実施例でこの符号化を逆にしたり他の符号化を用いることもある。アドレスタグ部はタグとしてデータキャッシュにストアされるアドレスの部分であり、アドレスインデックス部はインデックスとしてデータキャッシュが用いる部分である。オフセットおよびサイズ情報は、ストアにより更新されてキャッシュライン内にあるバイトを示す。ウェイ表示は、ヒットビットがセットされた場合（ストアのヒットを示す）、（セットアソシアティブの実施例において）ストアがデータキャッシュでヒットするウェイを示す。最後に、ここでのデータは、データキャッシュおよび／またはメモリに与えられるストアデータである。

【００１７】

比較器４０４は、ストア待ち行列４００の各エントリからストアインデックスを受けるように結合され、かつ、実行されているロードのロードインデックスを受けるように結合される。比較器４０４は、ロードおよびストアインデックスを比較し、一致が検出される

とヒット制御回路４０２に対し信号をアサートする。比較器４０４はこのように、ストア待ち行列４００の各エントリに対する比較器回路を表わしており、各比較器回路が出力信号をヒット制御回路４０２に与える。同様に、比較器４０６はストア待ち行列４００の各エントリにストアされたウェイ表示を受けるように結合され、かつ、ロードウェイ表示を受けるように結合される。比較器４０６は、ロードおよびストアのウェイ表示を比較し、一致が検出されるとヒット制御回路４０２に対し信号をアサートする。比較器４０６はこのように、ストア待ち行列４００の各エントリに対する比較器回路を表わしており、各比較器回路が出力信号をヒット制御回路４０２に与える。なお、所望されれば比較器４０４および４０６を連想記憶装置（ＣＡＭ）構成としてストア待ち行列４００に組み込んでもよい。

【００１８】

ヒット制御回路４０２は、各エントリからのヒットビットおよび実行されているロードについてのヒット信号を受けるように結合される。ロードインデックスおよびストア待ち行列４００に示されたストアのストアインデックスが一致すれば、ロードおよびこのストアはヒットであり、ロードおよびこのストアのウェイ表示は一致し、ヒット制御回路４０２により、データがストア待ち行列４００からロードのためにフォワードされる。より具体的には、ヒット制御回路４０２は、ストア待ち行列４００に、ヒットしているエントリのエントリ番号の表示で知らせ、ストア待ち行列４００は、このエントリからのデータを、データキャッシュからのキャッシュデータの代わりにフォワードするために与える。

【００１９】

なお、ロードアドレスは、ロードによるデータキャッシュの探索の開始時に、比較のために利用でき、ロードヒット信号は、データキャッシュの探索の終了近くまで（たとえばロードアドレスが変換されてキャッシュタグと比較された後）決定されない。さらに、ロードのウェイ表示も、ヒット信号が決定されるまで決定されない。したがって、この実施例において、ヒット制御回路４０２は、ロードインデックスおよびストアインデックスの一致、ならびに、ストアのヒットビットがストアがヒットであると示すことに応じて、ストア待ち行列４００からのデータフォワードを合図する（かつストア待ち行列４００にデータをフォワードさせる）ように構成される。ヒット制御回路４０２は、図１に示したフォワード信号をアサートしてデータフォワーディングを知らせる。次に、ロードのために、ヒット信号およびウェイ表示を求める。ヒット制御回路４０２は、ロードがストアにヒットしていることを、ロードウェイ表示をストアウェイ表示と比較しヒット信号がアサートされてヒットを示していることを確認することにより、確かめる。ウェイ表示が一致しかつロードのヒット信号がヒットを示しているならば、ヒット制御回路４０２は、フォワーディングが正しいと判断する。他方、フォワーディングが誤りである場合、ヒット制御回路４０２は、図１に示したフォワード取消信号をアサートして、フォワードされたストアデータを受けたプロセッサの部分に、フォワードが誤りであると知らせる。ある具体的な実施例では、データのフォワードを第１のクロックサイクルで行ない、フォワードの取消しを第１のクロックサイクルに続く第２のクロックサイクルで行なう。

【００２０】

上記は、図１に示した装置の、１つのロードが実行される場合の動作について説明したものである。しかしながら、複数のロードが同時に実行される実施例も意図されている。上記のように各ロードを同時に処理することができる。

【００２１】

ロードおよびストアインデックス（ならびにデータアドレスのヒットウェイ）を比較して、ロードおよびストアが同じキャッシュラインにアクセスしていると判断する。さらなる情報を用いて、ロードが読出した少なくとも１バイトがストアにより更新されていると判断する。たとえば、アドレスのオフセット部分ならびにロードおよびストアに影響されるサイズ（すなわちバイト数）を用いることができる。オフセットおよびサイズ情報は、設計上の選択によって、何らかの適切なフォーマットで与えて符号化することができる。たとえば、オフセットおよびサイズ情報は、バイトイネーブルマスクを含み、ここで各バ

イトの１ビットがキャッシュラインにある。このビットがセットされると、対応するバイトがアクセスされる。ロードおよびストアに対するバイトイネーブルマスクの各ビットの論理和をとり、このバイトがロードにより読出されかつストアにより書込まれるかどうか判断する。バイトイネーブルマスクを、キャッシュラインの一部のために生成してもよく（たとえばキャッシュが１キャッシュラインあたり複数のバンクを有する場合）、バンク選択のために用いるオフセットの部分を、バイトイネーブルマスクビットの論理和に加え、ロードおよびストアアドレス間で比較する。ロードおよびストアアドレスのオフセットの部分を、インデックス比較に加えて、比較器４０４を用いて比較する。ヒット制御回路４０２は、（上記のインデックス比較、ヒットビットおよびウェイ表示に加えて）オフセットおよびサイズ情報を用いて、ストア待ち行列４００にストアされたデータをロードのためにフォワードするかどうか判断する。

【００２２】

なお、ロードの実行中にストア待ち行列４００の２以上のエントリがヒットする場合がある。ヒット制御回路４０２は、ヒットしているエントリに対応するストアのうち、プログラム順序で最も新しい（最後に実行された）ストアを求め、データをそのエントリからフォワードする。また、ロードが読出した１以上のバイトを、ロードがヒットしたストアが、ロードが読出した１以上の他のバイトについては更新しない場合がある。その場合、データキャッシュは、ストアデータをキャッシュデータと組合せて、ロードが読出したバイトを与える。複数のストアが、あるロードが読出したバイトのうち異なるバイトを与える場合、このロードをリタイアして再び試みる。複数のストアのうち１以上をリタイアしてデータキャッシュに与え、これらストアにより更新されロードによって読出されたバイトを、データキャッシュから与える。その代わりとして、図１の装置は、異なるストアからのバイトを組合せてロードデータを与えてもよい。所望に応じ、これ以外の実施例において上記のモデルを他のやり方で処理してもよい。

【００２３】

なお、ここで示されている比較器４０６は、ストア待ち行列４００にストアされたウェイ表示をロードのウェイ表示と比較しているが、これに代わる実施例では、ロードのためにデータをフォワードするのに用いるエントリからウェイ表示を読出し（このフォワーディングがインデックス比較およびデータキャッシュにおけるストアヒットに基づく場合）、読出されたウェイ表示をロードウェイ表示と比較してロードおよびストアが同じウェイでヒットしているかどうか検査する。

【００２４】

本明細書で用いているアドレスのインデックス部（または簡潔に「インデックス」）は、このアドレスに対応するデータをストアするのに適切な１以上のキャッシュエントリを選択するために用いる部分である。加えて、データアドレスは、データキャッシュ内で、このデータアドレスが識別するデータがそのデータキャッシュにストアされている場合に「ヒットする」。データアドレスは、データキャッシュ内で、このデータアドレスが識別するデータがそのデータキャッシュにストアされている場合に「ミスする」。さらに、セットアソシアティブデータキャッシュは、所与のインデックスに対応するキャッシュラインをストアするのに適切な複数のキャッシュエントリを含む。各エントリはそのインデックスに対しては異なるウェイである。

【００２５】

図２は、ロード／ストアユニット内でストア待ち行列４００を用いるプロセッサの実施例を示す。その代わりとして、このプロセッサおよびロード／ストアユニットは、図４－６に関連して説明する待ち行列構成を用いてもよい。図１の装置または図４－６の実施例を用いるこれ以外のプロセッサの実施例も意図されている。

【００２６】

プロセッサ概観

次に図２を参照して、プロセッサ１０の一実施例のブロック図が示される。これ以外の実施例が可能であり意図されている。図２に示すように、プロセッサ１０は、プリフェッ

チ／プリデコードユニット１２と、分岐予測ユニット１４と、命令キャッシュ１６と、命令アライメントユニット１８と、複数のデコードユニット２０Ａ－２０Ｃと、複数のリザベーション（保留、reservation）ステーション２２Ａ－２２Ｃと、複数の機能ユニット２４Ａ－２４Ｃと、ロード／ストアユニット２６と、データキャッシュ２８と、レジスタファイル３０と、リオーダ（reorder）バッファ３２と、ＭＲＯＭユニット３４と、バスインターフェイスユニット３７とを含む。本明細書において特定の参照番号およびこれに続く文字で示されている構成要素はまとめて、参照番号のみを用いて表わす。例として、デコードユニット２０Ａ－２０Ｃはまとめてデコードユニット２０と示す。

【００２７】

プリフェッチ／プリデコードユニット１２は、バスインターフェイスユニット３７から命令を受けるように結合され、かつさらに、命令キャッシュ１６および分岐予測ユニット１４に結合される。同様に、分岐予測ユニット１４は、命令キャッシュ１６に結合される。さらに、分岐予測ユニット１４は、デコードユニット２０および機能ユニット２４に結合される。命令キャッシュ１６はさらに、ＭＲＯＭユニット３４および命令アライメントユニット１８に結合される。命令アライメントユニット１８は、デコードユニット２０に結合される。各デコードユニット２０Ａ－２０Ｃは、ロード／ストアユニット２６およびリザベーションステーション２２Ａ－２２Ｃにそれぞれ結合される。リザベーションステーション２２Ａ－２２Ｃはさらに、それぞれの機能ユニット２４Ａ－２４Ｃに結合される。加えて、デコードユニット２０およびリザベーションステーション２２は、レジスタファイル３０およびリオーダバッファ３２に結合される。機能ユニット２４はまた、ロード／ストアユニット２６、レジスタファイル３０およびリオーダバッファ３２に結合される。データキャッシュ２８は、ロード／ストアユニット２６およびバスインターフェイスユニット３７に結合される。バスインターフェイスユニット３７はさらに、Ｌ２キャッシュへのＬ２インターフェイスおよびバスに結合される。最後に、ＭＲＯＭユニット３４は、デコードユニット２０に結合される。

【００２８】

命令キャッシュ１６は、命令をストアするために設けられた高速キャッシュメモリである。命令は、命令キャッシュ１６からフェッチされデコードユニット２０にディスパッチされる。ある実施例において、命令キャッシュ１６は、６４キロバイトまでの命令を、６４のバイトライン（１バイトは８バイナリビットを含む）を有する２ウェイセットアソシアティブ構成内にストアするように構成される。その代わりとして、これ以外の所望の構成およびサイズを用いてもよい。例として、命令キャッシュ１６を、フルアソシアティブ、セットアソシアティブまたはダイレクトマッピング構成として実現してもよい。

【００２９】

命令は、プリフェッチ／プリデコードユニット１２により、命令キャッシュ１６にストアされる。命令を、要求される前に、命令キャッシュ１６からプリフェッチ方法に従いプリフェッチしてもよい。プリフェッチ／プリデコードユニット１２は、種々のプリフェッチ方法を用いることができる。プリフェッチ／プリデコードユニット１２は、命令を命令キャッシュ１６に転送する際、命令の各バイトにつき３つのプリデコードビットすなわちスタートビット、エンドビットおよび機能ビットを生成する。プリデコードビットは、各命令の境界を示すタグを形成する。プリデコードタグは、以下でより具体的に説明するように、所与の命令をデコードユニット２０が直接デコードできるかどうか、または、命令をＭＲＯＭユニット３４が制御するマイクロコード手続を呼出すことにより実行するかどうかといった、さらなる情報も伝える。さらに、プリフェッチ／プリデコードユニット１２を、分岐命令を検出し分岐命令に対応する分岐予測情報を分岐予測ユニット１４にストアするように構成してもよい。これ以外の実施例において何らかの適切なプリデコード方法を用いてもよい。

【００３０】

次に、可変バイト長命令セットを用いるプロセッサ１０のある実施例においてプリデコードタグを符号化することについて説明する。可変バイト長命令セットは、異なる命令が

異なる数のバイトを占める命令セットである。プロセッサ１０の一実施例で用いる可変バイト長命令セットの一例としてｘ８６命令セットを挙げる。

【００３１】

ここで例として挙げる符号化では、所与のバイトがある命令の第１のバイトである場合、このバイトに対しスタートビットがセットされる。このバイトが命令の最終バイトである場合、このバイトに対しエンドビットがセットされる。デコードユニット２０が直接デコードできる命令のことを「高速経路」命令と呼ぶ。残りのｘ８６命令のことを、ある実施例ではＭＲＯＭ命令と呼ぶ。高速経路命令については、機能ビットは、命令に含まれる各プレフィックスバイトに対しセットされ、これ以外のバイトに対してはクリアされる。その代わりとして、ＭＲＯＭ命令については、機能ビットは、各プレフィックスバイトに対しクリアされ、これ以外のバイトに対してはセットされる。命令の種類を、エンドバイトに対応する機能ビットを調べることによって決定できる。この機能ビットがクリアされている場合、命令は高速経路命令である。逆に、この機能ビットがセットされている場合、命令はＭＲＯＭ命令である。したがって、ある命令の演算コードの場所は、デコードユニット２０により直接デコードされる命令内の、命令の第１のクリアの機能ビットに関連するバイトとして、求めることができる。たとえば、２つのプレフィックスバイト、Mod R/Mバイト、および中間バイトを含む高速経路命令は、以下のようなスタート、エンドおよび機能ビットを有する。

【００３２】

スタートビット１００００
エンドビット　００００１
機能ビット　　１１０００

ＭＲＯＭ命令は、デコードユニット２０がデコードするには複雑すぎると判断される命令である。ＭＲＯＭ命令は、ＭＲＯＭユニット３４を呼出すことによって実行される。より具体的には、ＭＲＯＭ命令がある場合、ＭＲＯＭユニット３４は、この命令を解析し規定された高速経路命令のサブセットに発行して所望の動作を実施する。ＭＲＯＭユニット３４は、この高速経路命令のサブセットをデコードユニット２０にディスパッチする。

【００３３】

プロセッサ１０は、条件付分岐命令に続く命令を投機的にフェッチするために分岐予測を用いる。分岐予測ユニット１４は、分岐予測動作を行なうために含まれている。ある実施例では、分岐予測ユニット１４は、命令キャッシュ１６内の１つのキャッシュラインの１６バイト部分につき２つまでの分岐ターゲットアドレスおよび対応する分岐発生／分岐非発生（taken/not taken）予測をキャッシュする分岐ターゲットバッファを用いる。この分岐ターゲットバッファは、たとえば２０４８のエントリまたはこれ以外の適切な数のエントリを含む。プリフェッチ／プリデコードユニット１２は、特定のラインがプリデコードされたときに最初の分岐ターゲットを求める。続いて、キャッシュラインに対応する分岐ターゲットの更新が、キャッシュライン内で命令を実行したことによって発生する。命令キャッシュ１６は、フェッチされている命令アドレスを示し、分岐予測ユニット１４は、どの分岐ターゲットアドレスを選択して分岐予測を形成するか判断する。デコードユニット２０および機能ユニット２４は、分岐予測ユニット１４に更新情報を与える。デコードユニット２０は、分岐予測ユニット１４が予測しなかった分岐命令を検出する。機能ユニット２４は、分岐命令を実行し、予測された分岐方向が誤りかどうか判断する。分岐方向が「テイクン（発生）」となるのは、後続の命令が分岐命令のターゲットアドレスからフェッチされる場合である。逆に、分岐方向が「ノットテイクン（非発生）」であるのは、後続の命令が分岐命令に続く記憶場所からフェッチされる場合である。分岐命令予測誤りが検出されると、誤って予測された分岐に続く命令が、プロセッサ１０の多様なユニットから廃棄される。これに代わる構成では、分岐予測ユニット１４は、デコードユニット２０および機能ユニット２４ではなくリオーダバッファ３２に結合されて、リオーダバッファ３２から分岐予測誤り情報を受ける。分岐予測ユニット１４は種々の適切な分岐予測アルゴリズムを用いることができる。

【００３４】
命令キャッシュ１６からフェッチされた命令は、命令アライメントユニット１８に送られる。命令が命令キャッシュ１６からフェッチされると、対応するプリデコードデータがスキャンされ、命令アライメントユニット１８に（かつＭＲＯＭユニット３４に）、フェッチされた命令に関する情報が与えられる。命令アライメントユニット１８は、スキャンデータを用いて各デコードユニット２０に命令を整列させる。ある実施例において、命令アライメントユニット１８は、３組の８命令バイトからの命令をデコードユニット２０に整列させる。デコードユニット２０Ａは、現在デコードユニット２０Ｂおよび２０Ｃが受けている命令に（プログラム順序で）先行する命令を受ける。同様に、デコードユニット２０Ｂは、現在デコードユニット２０Ｃが受けている命令にプログラム順序で先行する命令を受ける。

【００３５】
デコードユニット２０は、命令アライメントユニット１８から受けた命令をデコードするように構成される。レジスタオペランド情報が検出され、レジスタファイル３０およびリオーダバッファ３２に送られる。さらに、命令が１以上のメモリ動作の実施を要求するものであれば、デコードユニット２０は、このメモリ動作をロード／ストアユニット２６に送る。各命令は、機能ユニット２４に対する１組の制御値にデコードされ、これらの制御値が、オペランドアドレス情報および命令に含まれた変位または即値データとともに、リザベーションステーション２２に送られる。ある特定の実施例では、各命令は、２つまでの動作にデコードされ、機能ユニット２４Ａ－２４Ｃにより別々に実行される。

【００３６】
プロセッサ１０は、命令順変更（アウトオブオーダ、out-of-order）実行をサポートし、そのため、リオーダバッファ３２を用いて、レジスタの読出および書込動作の当初のプログラムシーケンスを追跡し、レジスタ再命名を実施し、投機的な命令が実行され分岐予測誤りが修復されるようにし、的確な例外にし易くすくする。リオーダバッファ３２内の一時記憶場所を、レジスタを更新して投機的レジスタ状態をストアすることを含む命令のデコードの際に確保しておく。分岐予測が誤りであれば、予測誤り経路に沿い投機的に実行された命令の結果を、バッファにおいて、レジスタファイル３０への書込み前に無効化することができる。同様に、特定の命令が結果として例外をもたらした場合、その特定の命令に続く命令を廃棄する。このようにして、例外は「的確」となる（すなわち例外を生じさせる特定の命令に続く命令は、その命令の前に完了しない）。なお、特定の命令は、プログラム順序でその特定の命令に先行する命令よりも先に実行される場合、投機的に実行される。先行する命令は、分岐命令または例外発生命令であり、その場合、リオーダバッファ３２は投機的結果を廃棄する。

【００３７】
デコードユニット２０の出力で与えられた命令制御値および即値または変位データは直接それぞれのリザベーションステーション２２に送られる。ある実施例では、各リザベーションステーション２２が、対応する機能ユニットに発行されるのを待っている６つまでのペンディング命令についての命令情報（すなわち命令制御値、オペランド値、オペランドタグおよび／または即値データ）を保持できる。なお、図２の実施例では、各リザベーションステーション２２は専用の機能ユニット２４と関連付けられている。したがって、リザベーションステーション２２および機能ユニット２４により３つの専用「発行位置」が形成される。言換えれば、発行位置０がリザベーションステーション２２Ａおよび機能ユニット２２Ａにより形成される。整列してリザベーションステーション２２Ａにディスパッチされる命令を、機能ユニット２４Ａが実行する。同様に、発行位置１は、リザベーションステーション２２Ｂおよび機能ユニット２４Ｂにより形成され、発行位置２は、リザベーションステーション２２Ｃおよび機能ユニット２４Ｃにより形成される。

【００３８】
特定の命令をデコードしたときに、必要なオペランドがレジスタ場所であった場合、レジスタアドレス情報が同時にリオーダバッファ３２およびレジスタファイル３０に送られ

る。当業者であれば、ｘ８６レジスタファイルが８つの３２ビット実レジスタを含む（すなわち典型的にはＥＡＸ、ＥＢＸ、ＥＣＸ、ＥＤＸ、ＥＢＰ、ＥＳＩ、ＥＤＩおよびＥＳＰと呼ばれる）ことがわかるであろう。ｘ８６プロセッサアーキテクチャを用いたプロセッサ１０の実施例では、レジスタファイル３０は、各３２ビット実レジスタに対する記憶場所を含む。さらなる記憶場所がＭＲＯＭユニット３４が使用するためにレジスタファイル３０内に含まれている。リオーダバッファ３２に含まれる一時記憶場所は、こうしたレジスタの内容を変更してアウトオブオーダ（命令順変更）実行が行なえるようにするためのものである。リオーダバッファ３２の一時記憶場所が、各命令に対して確保され、これは、デコードの際に実レジスタのうち１つの内容を変更するものと決定される。したがって、特定のプログラムの実行中の種々のポイントで、リオーダバッファ３２は、所与のレジスタの投機的に実行された内容を含む１以上の場所を含み得る。所与の命令のデコードに続いて、リオーダバッファ３２が所与の命令においてオペランドとして用いられたレジスタに割当てられた以前のひとつまたは複数の場所を有すると判断されれば、リオーダバッファ３２は、対応するリザベーションステーションに、１）最後に割当てられた場所の値、または２）その値を、最終的には以前の命令を実行する機能ユニットが生成していない場合は、最後に割当てられた場所のタグを転送する。リオーダバッファ３２が、所与のレジスタのために確保された場所を有していれば、オペランド値（またはリオーダバッファタグ）が、レジスタファイル３０からではなくリオーダバッファ３２から与えられる。リオーダバッファ３２において必要とされるレジスタのために確保された場所がなければ、その値は直接レジスタファイル３０から取込まれる。オペランドが記憶場所に対応していれば、オペランド値がロード／ストアユニット２６を通してリザベーションステーションに与えられる。

【００３９】

ある具体的な実施例において、リオーダバッファ３２は、同時にデコードされた命令を１単位としてストアし操作するように構成される。この構成のことを本明細書では「ライン指向」と呼ぶ。いくつかの命令をまとめて処理することにより、リオーダバッファ３２内で用いるハードウェアを単純化することができる。たとえば、この実施例に含まれるライン指向リオーダバッファは、デコードユニット２０が１以上の命令をディスパッチするときは常に、３つの命令（各デコードユニット２０から１つずつ）に関する命令情報に十分な記憶を割当てる。対照的に、従来のリオーダバッファでは、実際にディスパッチされる命令の数に応じて、可変量の記憶が割当てられる。この可変量の記憶を割当てるには比較的多数の論理ゲートが必要である。同時にデコードされた命令の各々が実行されたとき、命令結果は同時にレジスタファイル３０にストアされる。したがって、記憶は、同時にデコードされる命令の別の組に割当てるために空いている。さらに、１命令当り用いられる制御論理回路の量は減少する。なぜなら、制御論理は同時にデコードされるいくつかの命令に対して償却されるからである。特定の命令を識別するリオーダバッファタグを、２つのフィールド、すなわちラインタグおよびオフセットタグに分割できる。ラインタグは、特定の命令を含む同時にデコードされた命令の組を識別し、オフセットタグは、この組内のどの命令が特定の命令に対応するか識別する。なお、命令結果をレジスタファイル３０にストアし対応する記憶を空けることを、命令を「リタイアする」という。さらに、プロセッサ１０の種々の実施例においていかなるリオーダバッファ構成を用いてもよい。

【００４０】

先に述べたように、リザベーションステーション２２は、命令を、その命令が対応する機能ユニット２４によって実行されるまでストアする。ある命令が実行のために選択されるのは、（ｉ）その命令のオペランドが既に与えられている場合、および（ii）同じリザベーションステーション２２Ａ－２２Ｃにありプログラム順序で当該命令に先行している命令のためのオペランドがまだ与えられていない場合である。なお、ある命令を機能ユニット２４の１つが実行すると、その命令の結果は直接、その結果を待っているリザベーションステーション２２に送られ、同時にその結果が送られてリオーダバッファ３２を更新する（この技術を一般に「結果フォワーディング」と呼ぶ）。ある命令は、実行のために

選択されて機能ユニット２４Ａ－２４Ｃに送られるが、これは、関連する結果がフォワードされるクロックサイクル中に行なわれる。この場合、リザベーションステーション２２は、フォワードされた結果を機能ユニット２４に送る。命令が複数の動作にデコードされて機能ユニット２４により実行される実施例では、この動作は別々にスケジューリングされる。

【００４１】

ある実施例において、各機能ユニット２４は、加算および減算という整数算術演算、シフト、ローテート、論理演算ならびに分岐演算を行なうように構成されている。これらの動作（演算）は、デコードユニット２０が特定の命令のためにデコードした制御値に応じて行なわれる。なお、浮動小数点ユニット（図示せず）を用いて浮動小数点演算に対応してもよい。浮動小数点ユニットは、コプロセッサとして動作し、ＭＲＯＭユニット３４またはリオーダバッファ３２から命令を受けた後にリオーダバッファ３２と連絡してその命令を完了する。加えて、機能ユニット２４は、ロード／ストアユニット２６が実行するロードおよびストアメモリ動作のためにアドレス生成を行なうように構成されていてもよい。ある特定の実施例で、各機能ユニット２４は、アドレスを生成するためのアドレス生成ユニットおよび残余の機能を果たすための実行ユニットを含む。これら２つのユニットは、１クロックサイクルにおいて異なる命令または動作に対し独立して動作する。

【００４２】

各機能ユニット２４はまた、条件付分岐命令の実行に関連する情報を分岐予測ユニット１４に与える。分岐予測が誤っている場合、分岐予測ユニット１４は、既に命令処理パイプラインに入っている誤って予測された分岐に続く命令をフラッシュし、命令キャッシュ１６またはメインメモリから必要な命令をフェッチする。なお、こうした状況下では、当初のプログラムシーケンスにおいて予測誤り分岐命令後に発生した命令の結果は廃棄され、これは、投機的に実行され一時的にロード／ストアユニット２６およびリオーダバッファ３２にストアされているものを含む。なお、分岐実行結果は、機能ユニット２４がリオーダバッファ３２に与えるもので、これは、機能ユニット２４に分岐の予測誤りを示す。

【００４３】

機能ユニット２４が生成した結果は、レジスタ値が更新されていればリオーダバッファ３２に送られ、記憶場所の内容が変更されていればロード／ストアユニット２６に送られる。結果をレジスタにストアするのであれば、リオーダバッファ３２は、命令がデコードされたときにレジスタの値のために確保しておいた場所に結果をストアする。複数の結果バス３８が、機能ユニット２４およびロード／ストアユニット２６から結果をフォワードするために含まれている。結果バス３８は、発生した結果、および、実行されている命令を識別するリオーダバッファタグを送る。

【００４４】

ロード／ストアユニット２６は、機能ユニット２４およびデータキャッシュ２８間のインターフェイスを与える。ある実施例において、ロード／ストアユニット２６は、データキャッシュ２８にまだアクセスしていないペンディング中のロードまたはストアのデータおよびアドレス情報のための記憶場所を有する第１のロード／ストアバッファと、既にデータキャッシュ２８にアクセスしているロードおよびストアのデータおよびアドレス情報のための記憶場所を有する第２のロード／ストアバッファとを含むように構成されている。たとえば、第１のバッファは１２の場所を含み、第２のバッファは３２の場所を含む。デコードユニット２０は、ロード／ストアユニット２６へのアクセスを調停する。第１のバッファが一杯であるとき、デコードユニットは、ペンディング中のロードまたはストア要求情報のための場所がロード／ストアユニット２６にできるまで、待機しなければならない。ロード／ストアユニット２６はまた、ロードメモリ動作の、ペンディング中のストアメモリ動作に対する従属性検査も行ない、データコヒーレンシ（一貫性）が確実に維持されるようにする。メモリ動作は、プロセッサ１０およびメインメモリサブシステム間のデータ転送である。メモリ動作は、メモリにストアされたオペランドを用いる命令の結果である、または、データ転送を行なわせるが他の動作は行なわせないロード／ストア命令

の結果である。加えて、ロード／ストアユニット２６は、セグメントレジスタなどの特殊レジスタおよびｘ８６プロセッサアーキテクチャが定めるアドレス変換メカニズムに関する他のレジスタのための特殊レジスタ記憶を含む。

【００４５】

データキャッシュ２８は、ロード／ストアユニット２６およびメインメモリサブシステム間で転送されているデータを一時的にストアするために設けられた高速キャッシュメモリである。ある実施例では、データキャッシュ２８には、２ウェイセットアソシアティブ構成において６４キロバイトまでのデータをストアする容量がある。データキャッシュ２８を、セットアソシアティブ構成、フルアソシアティブ構成、ダイレクトマッピング構成およびその他の構成の適切な大きさを含む、種々の特殊メモリ構成で実現できることがわかるであろう。

【００４６】

ｘ８６プロセッサアーキテクチャを用いたプロセッサ１０の特定の実施例では、命令キャッシュ１６およびデータキャッシュ２８は、線形的にアドレス指定され物理的にタグ付けされる。線形アドレスは、命令が特定するオフセットおよびｘ８６アドレス変換メカニズムのセグメント部が特定するベースアドレスから形成される。任意選択として、線形アドレスをメインメモリへのアクセスのために物理アドレスに変換してもよい。線形－物理変換は、ｘ８６アドレス変換メカニズムのページング部によって指定されている。物理アドレスは、物理タグと比較されてヒット／ミス状態が判断される。

【００４７】

バスインターフェイスユニット３７は、コンピュータシステム内でバスを介してプロセッサ１０とこれ以外の構成要素との間で連絡が行なえるように構成されている。たとえば、このバスは、Digital Equipment Corporationが開発したＥＶ－６バス互換のものでもよい。その代わりとして、パケットに基づいたもの、単方向リンクまたは双方向リンクなどを含む適切な相互接続構成を用いてもよい。任意選択のＬ２キャッシュインターフェイスを用いてレベル２キャッシュに対するインターフェイスを設けてもよい。

【００４８】

ロード／ストアユニット

次に、ロード／ストアユニット２６のある実施例についてより詳細に説明する。これ以外の実施例が可能であり意図されている。図３は、プロセッサ１０のある実施例に従う、ロード／ストアユニット２６、リオーダバッファ３２、データキャッシュ２８、バスインターフェイスユニット（ＢＩＵ）３７、デコードユニット２０Ａ、リザベーションステーション２２Ａおよび機能ユニット２４Ａを示し、相互接続を強調している。他の実施例において、所望に応じてさらなる、代替のまたはこれに代わる相互接続を用いてもよい。デコードユニット２０Ｂ－２０Ｃ、リザベーションステーション２２Ｂ－２２Ｃ、機能ユニット２４Ｂ－２４Ｃおよび図３に示した他のユニット間の相互接続も図３に示したものと同様である。

【００４９】

デコードユニット２０Ａは、命令アライメントユニット１８から命令を受けてその命令をデコードする。デコードユニット２０Ａは、デコードした命令をリザベーションステーション２２Ａに与え、リザベーションステーション２２Ａは、デコードされた命令を、この命令が実行のために選択されるまでストアする。加えて、この命令がロードまたはストアメモリ動作を指定していれば、デコードユニット２０Ａは、Ｌ／Ｓライン４６Ａを介してロード／ストアユニット２６に信号を送る。ロード／ストアユニット２６はデコードユニット２０Ｂ－２０Ｃからも同様の信号を受ける。Ｌ／Ｓライン４６Ａは、デコードされている命令が指定しているのはロードメモリ動作なのか、ストアメモリ動作なのかまたは双方なのかを示す。たとえば、Ｌ／Ｓライン４６Ａは、ロードラインおよびストアラインを含み得る。何のメモリ動作も指定されていない場合、双方のラインの信号はデアサートされる。ロードライン上の信号は、ロードメモリ動作が指定された場合にアサートされ、同様に、ストアライン上の信号はストアメモリ動作が指定された場合にアサートされる。

ロードメモリ動作およびストアメモリ動作双方が指定されていれば、これらの信号双方がアサートされる。Ｌ／Ｓライン４６Ａ上の信号に応答して、ロード／ストアユニット２６は、含まれているロード／ストアバッファにエントリを割当てて、対応するメモリ動作をストアする。

【００５０】

上記に加え、デコードユニット２０Ａは、リオーダバッファ３２に、デコードされている命令に関する情報を与える。リオーダバッファ３２は、この情報（同様の情報が他のデコードユニット２０Ｂ－２０Ｃから与えられる）を受け、これに応じてリオーダバッファエントリを割当てる。割当てられたリオーダバッファエントリは、命令タグバス４８でロード／ストアユニット２６に送られるリオーダバッファタグにより識別される。命令タグバス４８は、可能な命令各々（たとえばこの実施例では３つありこれらはそれぞれデコードユニット２０Ａ－２０Ｃからのものである）に対するタグを送るように構成される。これに代えて、前述のライン指向構成を用いる実施例では、リオーダバッファ３２を、ラインのラインタグを送るように構成し、ロード／ストアユニット２６が、特定のロードまたはストアを信号で知らせる発行位置のオフセットタグで、ラインタグを増大してもよい。

【００５１】

リオーダバッファ３２はさらに、命令のレジスタオペランドに対する従属性検査を実施するように構成されてもよい。レジスタオペランドは、デコードユニット２０が送る命令情報において識別される。ストアメモリ動作については、ストアデータは、ロード／ストアユニット２６がストアアドレスに加えて受けるソースオペランドである。したがって、リオーダバッファ３２は、各ストアメモリ動作に対しストアデータを生成した命令を求め、ストアデータ／タグバス５０で、ストアデータ（ストアメモリ動作のディスパッチの際にリオーダバッファ３２またはレジスタファイル３０で利用できる場合）、または、そのストアデータに対するストアデータタグを送る。ストアメモリ動作に対応する命令が、レジスタの内容をメモリにストアするという明示ストア命令であれば、ストアデータを生成する命令の命令タグ（利用できるならばストアデータ）が送られる。他方、命令がストアメモリ動作を暗示動作として含む場合は、命令そのものがストアデータを生成する。こうした場合、リオーダバッファ３２は、この命令の命令タグをストアデータタグとして与える。

【００５２】

図３では簡潔にするために示していないが、リザベーションステーション２２Ａは、リオーダバッファ３２から、命令のオペランドタグおよび／またはデータも受ける。リザベーションステーション２２Ａは、オペランドタグおよび／またはデータを入手し、結果バス３８から残りのオペランドデータ（オペランドタグが識別するもの）が送られるのを待つ。ある命令がそのオペランドを受けると、機能ユニット２４Ａはそれを実行することができる。より具体的にいえば、ここに示している実施例では、機能ユニット２４Ａは、実行ユニット（ＥＸＵ）４０およびアドレス生成ユニット（ＡＧＵ）４２を含む。実行ユニット４０は、命令動作を行ない（たとえば算術演算および論理演算）、結果を生成し、その結果が結果バス３８Ａ（結果バス３８の１つ）でロード／ストアユニット２６、リザベーションステーション２２およびリオーダバッファ３２に転送される。ＡＧＵ４２は、データアドレスを生成しこのデータアドレスは命令が指定した１つまたは複数のメモリ動作で使用され、ＡＧＵ４２はこのデータアドレスをアドレスバス４４Ａを介してロード／ストアユニット２６に送る。なお、ＡＧＵ４２および実行ユニット４０が結果バス３８Ａを共有し機能ユニット２４Ａがアドレス生成およびその他命令実行動作を実施する実行ユニットのみを含む実施例を用いてもよい。ロード／ストアユニット２６はさらに、他の機能ユニット２４Ｂ－２４Ｃ内の実行ユニットおよびＡＧＵからの結果バスおよびアドレスバスを受けるように結合される。

【００５３】

ここで示している実施例ではＡＧＵ４２を用いているため、リザベーションステーション２２Ａはある命令のアドレス生成部を選択し、ＡＧＵ４２がこれを実行するが、これは

、アドレスを形成するオペランドは既に受けているが命令が指定するさらなるオペランドをまだ受けていないときに行なわれる。ＡＧＵ４２は、発生したアドレスを、ロード／ストアユニット２６に、アドレスバス４４Ａで、データアドレスを生成した命令の命令タグとともに送る。これに応じて、ロード／ストアユニット２６は、アドレスバス４４Ａで受けたタグを、ロード／ストアバッファにストアされた命令タグと比較して、データアドレスが対応するのはロードなのかストアなのかを判断する。

【００５４】

ロード／ストアユニット２６は、結果バス３８で与えられた結果タグをモニタして、ストアメモリ動作のストアデータを入手する。結果タグがロード／ストアユニット内のストアデータタグと一致していれば、ロード／ストアユニット２６は、対応するデータを入手し、このデータを対応するストア命令と関連付ける。

【００５５】

ロード／ストアユニット２６は、データキャッシュインターフェイスを介してデータキャッシュ２８に結合される。ロード／ストアユニット２６は、メモリ動作を選択して、データキャッシュインターフェイスを介してデータキャッシュ２８を探索し、データキャッシュインターフェイスから探索結果を受ける。一般的に、特定のメモリ動作に対するデータキャッシュの「探索」は、その特定のメモリ動作のデータアドレスをデータキャッシュ２８に送りデータキャッシュ２８によりデータアドレスがヒットしているかどうかを判断することを含む。データキャッシュ２８は、探索結果（たとえばヒット／ミス表示）をロード／ストアユニット２６に返す。加えて、特定のメモリ動作がロードでありヒットしていれば、データキャッシュ２８は対応するロードデータを結果バス３８Ｄでリザベーションステーション２２、リオーダバッファ３２およびロード／ストアユニット２６にフォワードする。ある実施例において、データキャッシュ２８は、２つのポートを含みこれに応じて２つまでの探索を同時に受ける。データキャッシュ２８は、たとえばバンク構成を用いてもよく、この構成では、キャッシュラインが少なくとも２つのバンクにわたりストアされ、２つの探索が、異なるバンクにアクセスしている限り同時に処理される。ある特定の実施例では、データキャッシュ２８は８つのバンクを用いる。データキャッシュインターフェイスの種々の実施例について以下でさらに詳しく説明する。

【００５６】

データキャッシュ２８は、ミスであった探索に応じてキャッシュラインを割当てるように構成され、バスインターフェイスユニット３７と連絡してミスのキャッシュラインをフェッチする。加えて、データキャッシュ２８は、取戻した変更されているキャッシュラインを、メインメモリ更新のためにバスインターフェイスユニット２７に送る。

【００５７】

バスインターフェイスユニット３７は、データキャッシュ２８に結合されまたスヌープインターフェイス５２を介してロード／ストアユニット２６にも結合される。バスインターフェイスユニット３７がスヌープインターフェイス５２を用いて、バスから受けるスヌープ動作に応じてコヒーレンシ（coherency）動作を行なう必要があるかどうかが判断される。一般的に、「スヌープ動作」は、バス上で、このバスに接続されたキャッシュに対するメモリのコヒーレンシを保つために（たとえばプロセッサ内で）行なわれる動作である。コヒーレンシが適切に保たれていれば、特定の記憶場所に対応しキャッシュの１つにストアされているデータのコピーは、他のキャッシュ各々にストアされたコピーと一致している。スヌープ動作は、明示動作である、または、特定の記憶場所のアドレスに対して実施される動作の暗示部分である。一般的に、スヌープ動作は、スヌープされるアドレス（「スヌープアドレス」）を指定し、かつ、アドレスがキャッシュにストアされている場合はキャッシュラインの所望の状態を指定する。バスインターフェイスユニットは、スヌープインターフェイス５２を介してスヌープ要求をデータキャッシュ２８およびロード／ストアユニット２６に送り、スヌープ動作を行なう。

【００５８】

リオーダバッファ３２は、命令のリタイアを管理する。リオーダバッファ３２は、リタ

イアインターフェイス５４を介してロード／ストアユニット２６と連絡をとり、リタイアされているかこれからリタイアされようとしている命令を識別する。たとえば、ある実施例では、ストアは、リタイアされるまでデータキャッシュ２８（またはメインメモリ）を更新しない。加えて、いくつかのロード命令を非投機的実行に制限してもよい。リオーダバッファ３２は、リタイアインターフェイス５４を介してロード／ストアユニット２６に、リタイアされているまたはリタイア可能なメモリ動作を示す。このように、デコードユニット２０が各命令のためにリオーダバッファ３２に与える命令情報は、この命令がロードまたはストア動作を含むかどうかを示す。ロード／ストアユニット２６は、特定のメモリ動作がリタイアされたときにロギングされているという肯定応答をリオーダバッファ３２に返し、リオーダバッファ３２はこれに続いて対応する命令をリタイアさせる。

【００５９】

ロード／ストアバッファは一杯になることがあるため、ロード／ストアユニット２６は、フロー制御メカニズムを用い、デコードユニット２０で、後続のメモリ動作を、その後続のメモリ動作のために十分なエントリがロード／ストアバッファにおいて（先のメモリ動作の完了によって）空になるまで、停止させる。たとえば、ロード／ストアユニット２６は、空いているエントリの数のカウントをデコードユニット２０に一斉に送り、デコードユニットは、このカウントが、デコードされている命令のメモリ動作に利用できるエントリが不十分であることを示す場合、停止する。ある特定の実施例に従うと、同時にデコードユニット２０によりデコードされている命令は、ロックステップ（lockstep）でリザベーションステーション２２に移動する。（図２に関して先に述べたようにラインが命令のためにリオーダバッファ３２に割当てられる）このような実施例では、デコードユニット２０は、同時にデコードされる命令の組内ですべてのメモリ動作に対して十分なエントリが利用できるまで停止する。これに代わるものとして、ロード／ストアユニット２６が、後続のメモリ動作をバッファエントリが利用できるようになるまで停止させるための停止信号を用いてもよい。何らかの適切なフロー制御メカニズムを用いればよい。

【００６０】

次に図４を参照して、ロード／ストアユニット２６のある実施例のブロック図が示される。これ以外の実施例が可能であり意図されている。図４に示した実施例では、ロード／ストアユニット２６は、第１のロード／ストアバッファ（ＬＳ１バッファ）６０、第２のロード／ストアバッファ（ＬＳ２バッファ）６２、ＬＳ１制御回路６４、ＬＳ２制御回路６６、一時バッファ６８、セグメント加算器７０、ポート０マルチプレクサ（ｍｕｘ）７２、ポート１マルチプレクサ７４、およびＬＳ２再探索マルチプレクサ７６を含む。セグメント加算器７０は、機能ユニット２４Ａ－２４Ｃ内のＡＧＵ４２からデータアドレスを受けるように結合される。（たとえば図３に示したアドレスバス４４Ａの一部、アドレスバス４４ＡＡは、機能ユニット２４Ａ内のＡＧＵ４２からのデータアドレスを送る。）マルチプレクサ７０および７２は、ＡＧＵ４２からのデータアドレスおよびセグメント加算器７０の出力を受けるように結合され、ＬＳ１バッファ６０にも結合される。マルチプレクサ７２はまた、ＬＳ２再探索マルチプレクサ７６からの入力を受ける。さらに、ＬＳ１バッファ６０は、セグメント加算器７０、ＬＳ１制御回路６４、一時バッファ６８、命令タグバス４８、ストアデータ／タグバス５０および結果バス３８ａ（結果バス３８の結果データ部分）に結合される。ＬＳ１制御回路６４は、マルチプレクサ７２および７４ならびにＬＳ２制御回路６６に結合される。さらに、ＬＳ１制御回路６４は、ＡＧＵ４２からのアドレスタグ（たとえば図３に示したアドレスタグバス４４Ａの一部、アドレスタグバス４４ＡＢは、機能ユニット２４Ａ内のＡＧＵ４２からのアドレスタグを送る）、結果タグバス３８ｂを介して結果タグ（結果バス３８の結果タグ部分）、およびＬ／Ｓライン４６（デコードユニット２０ＡからのＬ／Ｓライン４６Ａを含む）を受けるように結合される。一時バッファ６８およびＬＳ２バッファ６２は、結果バス３８ａおよび結果タグバス３８ｂに結合される。ＬＳ２バッファ６２はさらに、ＭＡＢタグバス７８上のミスアドレスバッファ（ＭＡＢ）タグを受け、データキャッシュ２８からの、物理アドレスバス８０上の物理アドレスを受けるように結合される。ＬＳ２バッファ６２はさらに、マルチプレ

クサ７６、ＬＳ２制御回路６６および一時バッファ６８に結合される。ＬＳ２制御回路６６はさらに、マルチプレクサ７６、リタイアインターフェイス５４、結果タグバス３８ｂ、スヌープインターフェイス５２、データキャッシュ２８からのヒット／ミス信号８２、およびバスインターフェイスユニット３７からの充填タグバス８４に結合される。

【００６１】

一般的に、ロード／ストアユニット２６は、プリキャッシュバッファ（ＬＳ１バッファ６０）およびポストキャッシュバッファ（ＬＳ２バッファ６２）を含む。メモリ動作は、プロセッサ１０内でディスパッチの際にＬＳ１バッファ６０に割当てられ、選択されてデータキャッシュ２８の探索が行なわれるまでＬＳ１バッファ６０内にある。データキャッシュ２８の探索に続き、メモリ動作は、探索状態（たとえばヒット／ミスなど）にかかわらず、ＬＳ２バッファ６２に移される。

【００６２】

ミスしたメモリ動作は、その後ＬＳ２再探索マルチプレクサ７６およびポート０マルチプレクサ７２を通して選択して、データキャッシュ２８を再び探索すればよい。本明細書で用いる「再探索」という用語は、特定の動作に対する第１の探索の後に第２のまたは後続の試みとしてキャッシュを探索することを指す。加えて、ストアメモリ動作を、ストアがリタイアの状態になるまでＬＳ２バッファ６２内に保持しておいてもよい。

【００６３】

Ｌ／Ｓライン４６上の信号に応答して、ＬＳ１制御回路６４は、ＬＳ１バッファ６０内のエントリを識別されたロードおよびストアメモリ動作に割当てる。ＬＳ１制御回路６４の制御により、ＬＳ１バッファ６０はそれぞれの命令タグおよびストアデータ／タグ（適用できる場合）を割当てられたエントリに受ける。次に、対応するデータアドレスをＡＧＵから受けて（ＬＳ１制御回路６４が受けたアドレスタグにより識別される）、割当てられたエントリにストアする。

【００６４】

メモリ動作はそのアドレスを受けるとデータキャッシュ２８を探索できるようになる。ＬＳ１制御回路６４は、メモリ動作を求めてＬＳ１バッファエントリをスキャンしてデータキャッシュ２８を探索し、ポート０マルチプレクサ７２およびポート１マルチプレクサ７４のために選択制御を生成する。したがって、ここで示している実施例では、１クロックサイクル当り２つまでのメモリ動作がデータキャッシュ２８を探索することができる。ある具体的な実施例に従うと、ＬＳ１制御回路６４は、データキャッシュ２８を探索するためにプログラム順序でメモリ動作を選択する。よって、ＬＳ１制御回路６４を、ＬＳ１バッファ６０内の最も古いメモリ動作に対するスキャンに制限するように構成してもよい。メモリ動作の「プログラム順序」とは、命令のフェッチおよび実行が一度に１つずつ行なわれる場合の命令の実行順序である。さらに、投機的にフェッチされた命令のプログラム順序とは（たとえば分岐予測に従ったもの）、上記のようにこの投機が正しいと仮定した上で命令が実行される順序である。命令がプログラム順序で他の命令に先行する場合、前者の命令は他の命令よりも古いといえる。逆に、命令がプログラム順序で他の命令の後続命令の場合、前者の命令は他の命令よりも新しいといえる。なお、他の実現化例では、所望に応じ順序を崩してメモリ動作を選択しデータキャッシュ２８を探索してもよい。

【００６５】

ＬＳ１制御回路６４は、メモリ動作を選択して、データアドレスを受けたときにデータキャッシュ２８を探索するように構成される。（この実施例ではメモリ動作がＬＳ１制御回路６４によりスキャンされているエントリ内にあると仮定する。）ＡＧＵ４２から受けたアドレスタグがそうでなければ選択可能なメモリ動作の命令タグに一致していれば、ＬＳ１制御回路６４は、マルチプレクサ７２および７４の一方を介してＡＧＵ４２から受けた対応するデータアドレスを選択する。

【００６６】

データアドレスは、ロード／ストアユニット２６に与えられたときに選択され探索が行なわれるが、データアドレスは、セグメント加算器７０の１つにも与えられる。この実施

例では、セグメント加算器７０は、ｘ８６アドレス指定方法のセグメント化部分を処理するために設けられている。ｘ８６命令セットアーキテクチャを用いない実施例では、セグメント加算器７０がなくてもよい。一般的に、ＡＧＵ４２はメモリ動作に対応する論理アドレスを生成する。この論理アドレスは、命令のアドレスオペランドの加算により生成されるアドレスである。ｘ８６アーキテクチャでは、２段の変換方法が定められており、セグメント化方法により論理アドレスから線形アドレスに、次にページング方法により物理アドレスに変換される。ＡＧＵ４２は命令のアドレスオペランドを加算するため、ＡＧＵが与えるデータアドレスは論理アドレスである。しかしながら、現代の命令コードは一般的に「フラットアドレス指定モード」を用いており、このモードでは、セグメントベースアドレス（論理アドレスに加算されて線形アドレスを形成するもの）は、ゼロにプログラミングされる。したがって、ロード／ストアユニット２６は、セグメントベースアドレスがゼロである（したがって論理および線形アドレスが等しい）と推定し、論理アドレスを選択してデータキャッシュ２８を探索する。セグメント加算器７０は、メモリ動作のために選択されたセグメントのセグメントベースアドレスを加算し、線形アドレスをマルチプレクサ７２および７４ならびにＬＳ１バッファ６０に送って記憶させる。特定のメモリ動作に対するセグメントベースアドレスが非ゼロであり、メモリ動作を選択して論理アドレスを受けた際にデータキャッシュ２８を探索するのであれば、ＬＳ１制御回路６４は、先行するアクセスを取消して（ロードデータがフォワードされないように）、対応する線形アドレスを対応するセグメント加算器７０の出力から選択してデータキャッシュ２８を探索する。これ以外の代替実施例では、ＡＧＵ４２がセグメントベースアドレスを受け線形アドレスを生成してもよい。さらに他の実施例では、フラットアドレス指定モードを要求してセグメントベースアドレスを無視してもよい。

【００６７】

マルチプレクサ７２および７４は、ＬＳ１バッファ６０内のエントリからデータアドレスを受けるようにも結合される。メモリ動作に対応するデータアドレスは、ＡＧＵ４２から受けた際にメモリ動作に割当てられるＬＳ１エントリにストアされる。データアドレスが、メモリ動作選択に際してエントリから選択され、データキャッシュ２８が探索される。なお、データアドレスに加え、他の情報をマルチプレクサ７０および７２を介してデータキャッシュ２８に送ってもよい。たとえば、メモリ動作がロードであるかストアであるかを示すものを送ってもよい。メモリ動作の命令タグを送って、ロードメモリ動作のためのロードデータとともに結果バス３８Ｄで転送することができる。この動作（適切なデータをマルチプレクスする）のサイズを送ることもできる。設計上の選択に応じて所望の情報を送ることができる。

【００６８】

ストアデータはストアメモリ動作のために与えられ、ストアメモリ動作はＬＳ１バッファ６０にある。これに応じて、ＬＳ１制御回路６４は結果タグバス３８ｂをモニタする。ＬＳ１バッファ６４内のストアデータタグに一致するタグが与えられると、結果バス３８ａのうち対応する結果バスからの対応するストアデータが、一致するストアデータタグを有するエントリ内に取り込まれる。

【００６９】

ＬＳ１制御回路６４は、ＬＳ１バッファ６０から、メモリ動作を、メモリ動作のデータキャッシュ２８探索に応じて除去する。ある具体的な実施例では、メモリ動作は、データキャッシュ２８の探索のために選択された後のサイクルで除去される。当該後のサイクルを用いて、メモリ動作がＡＧＵ４２の１つによりデータアドレスの生成が行なわれた際に選択された場合は、データアドレスをＬＳ１バッファ６０に伝搬してもよい。他の実施例では、メモリ動作が選択されたサイクル中にメモリ動作を除去することを選択してもよい。メモリ動作は選択された後のサイクルで除去されるため、ＬＳ１制御回路６４は、ＬＳ１バッファ６０内の最も古い４つのエントリをスキャンしてメモリ動作を選択してデータキャッシュ２８を探索するように構成される（先行するクロックサイクルでは２つまでのエントリを選択し現在のクロックサイクルでは２つまでのエントリを選択できる）。

【００７０】
ＬＳ１バッファ６０から除去されたメモリ動作は、一時バッファ６８に移される。一時バッファ６８を設けて、ＬＳ１バッファ６０からエントリを読出しこれらをＬＳ２バッファ６２に書込む際のタイミング上の制約を緩和してもよい。したがって、一時バッファ６８は、設計上好都合であるに過ぎず、全く任意的選択である。メモリ動作が一時バッファ６８に移された後のクロックサイクルで、メモリ動作はＬＳ２バッファ６２に移される。ストアデータは、ストアメモリ動作が一時バッファ６８に保持されているクロックサイクル中に結果バス３８上で与えられるので、一時バッファ６８は、結果タグバス３８ｂ上の結果タグをモニタし、ＬＳ１バッファ６０がデータを収集するのと同じやり方で結果バス３８ａからデータを収集する。
【００７１】
このように、データキャッシュ２８を探索したメモリ動作は、ＬＳ２バッファ６２に与えられる。この実施例では、すべてのメモリ動作が、データキャッシュ２８の最初の探索後にＬＳ２バッファ６２に与えられる。ストアは、データキャッシュ２８に与えられるまでＬＳ２バッファ６２に保持される（すなわちデータキャッシュ２８を更新できるようになるまで）。一般的に、ストアは、投機的でなくなったときに与えられる。ある実施例では、ストアは、リタイアに応じて（リタイアインターフェイス５４を介して示される）またはその後いずれかの時点で与えられる。ロードは、この実施例でもリタイアまでＬＳ２バッファ６２に保持される。ロードヒットは、スヌープのためＬＳ２バッファ６２内に留まる。ロードミスは、少なくともロードがアクセスするキャッシュラインがデータキャッシュ２８に送られるまではＬＳ２に保持される。キャッシュライン（またはロードデータを含むその一部分）がキャッシュの更新にスケジューリングされていることに応じて、ロードミスはデータキャッシュ２８の再探索にスケジューリングされる。再探索の際、ロードミスはロードヒットとなり（かつロードデータはデータキャッシュ２８によりフォワードされる）、ヒットとしてリタイアまで保持される。
【００７２】
ＬＳ２制御回路６６は、データキャッシュ２８を探索したメモリ動作のためにＬＳ２バッファ６２内にエントリを割当てる。加えて、ＬＳ２制御回路６６は、データキャッシュ２８から、ヒット／ミス信号８２の各探索について、探索状態情報を受ける。ヒット／ミス情報は、探索状態が与えられたメモリ動作に対応するＬＳ２バッファエントリにストアされる。ある実施例では、データキャッシュ２８は、データキャッシュへのアクセスと並行して仮想アドレスを物理アドレスに変換しようと試みるアドレス変換回路を含む。アドレス変換回路で変換を行なうことができなければ、変換が行なわれるまで探索はミスであると識別される（たとえばメインメモリ内のソフトウェア管理変換テーブルを探索することによって）。ある具体的な実現化例では、データキャッシュ２８内のアドレス変換回路は、３２エントリレベル１ＴＬＢと、４ウェイセットアソシアティブ、２５６エントリレベル２ＴＬＢとを含む、２レベル変換索引バッファ（ＴＬＢ）構成を有する。
【００７３】
メモリ動作のデータアドレスをデータキャッシュ２８が変換することができれば、対応する物理アドレスが物理アドレスバス２８で与えられる。ＬＳ２制御回路は、対応するエントリが仮想アドレスを物理アドレスで上書きするようにする。しかしながら、再探索におけるインデックス付けのためにいくつかの仮想アドレスビットは別に保持しておき、ストアデータは、データキャッシュ２８に対する仮想インデックス付けおよび物理的タグ付けが行なわれる実施例においては与えられる。
【００７４】
データキャッシュ２８にミスするメモリ動作に対し、データキャッシュ２８はエントリを含まれているミスアドレスバッファ内に割当てる。ミスアドレスバッファは、ミスアドレスを、バスインターフェイスユニット３７に送るために待ち行列に入れ、インターフェイスユニット３７はアドレスをＬ２キャッシュからまたはメインメモリからフェッチする。ミスアドレスバッファ内のエントリを識別するタグ（ＭＡＢタグ）が、ミスした各メモ

リ動作に対しＭＡＢタグバス７８上に与えられる。なお、データキャッシュ２８は、ミスアドレスバッファエントリをキャッシュラインに基づいて割当てる。したがって、後続の、同じキャッシュラインに対するミスは、同じＭＡＢタグを受け、さらなるミスアドレスバッファエントリが割当てられないようにする。

【００７５】

次に、バスインターフェイスユニット３７は、ミスのキャッシュラインをフェッチし、このキャッシュラインを充填データとしてデータキャッシュ２８に戻す。バスインターフェイスユニット３７はまた、キャッシュラインに対応するＭＡＢタグを充填タグとして充填タグバス８４上に置く。ＬＳ２制御回路６６は、充填タグをＬＳ２バッファ６２内のＭＡＢタグと比較する。ロードメモリ動作についてＭＡＢタグの一致が生じれば、そのロードを選択してデータキャッシュ２８を再探索する。２以上の一致が検出されれば、最も古い一致ロードを、後続のクロックサイクル中に選択される他のメモリ動作とともに選択する。ＭＡＢタグに一致するストアはヒットとしてマークされるが、データを与える試みを行なう前に非投機となるのを待つ。

【００７６】

ある実施例では、データのキャッシュラインは、複数のパケットを用いて戻される。各ロードメモリ動作は、どのパケットにアクセスしているかを記録し（またはそのパケットはロードアドレスの適切なアドレスビットから区別される）、バスインターフェイスユニット３７は、充填タグとともに戻されているパケットを識別する。このように、戻されるパケットにアクセスするロードのみを選択して再探索を行なう。

【００７７】

バスインターフェイスユニット３７は、充填データよりも前に充填タグを与えて、ロードが再探索のために選択されポート０を介してデータキャッシュ２８に送られて、データキャッシュ２８にデータのパケットが届くのと同時にデータ転送段に達するようにする。アクセスされたデータをこうして転送することができる。

【００７８】

ストアがデータキャッシュ２８の探索後にＬＳ２バッファ６２に送られ後続のロードがＬＳ１バッファ６０からデータキャッシュ２８を探索しそこからデータをフォワーディングできるようになっているため、古いストアと同じ記憶場所にアクセスしている新しいロードは、古いストアがデータをデータキャッシュ２８に与える前に、データキャッシュ２８を探索することができる。ロードの正しい結果とは、その古いストアに対応するストアデータを受けることである。したがって、ＬＳ２制御回路６６は、探索アドレスをモニタし、このアドレスへの古いストアがＬＳ２バッファ６２内にあるかどうか判断する。一致が検出されストアデータをＬＳ２バッファ６２内で利用できるのであれば、ＬＳ２制御回路６６はデータキャッシュ２８に信号で知らせフォワーディングのためにＬＳ２バッファから与えられるデータを選択し、選択されたデータを与える。他方、一致が検出されストアデータをＬＳ２バッファ６２内で利用することができなければ、データキャッシュ２８からのデータフォワーディングは取消される。ロードはＬＳ２バッファ６２に送られ、ストアデータを利用できるようになるまで探索のために選択される。ストアからロードへのフォワードについてのさらなる詳細事項については以下に示す。

【００７９】

一般的に、ＬＳ２制御回路６６は、ＬＳ２バッファ６２内のエントリをスキャンしメモリ動作を選択してデータキャッシュ２８を再探索するように構成されている。ロードミスを選択し、データがデータキャッシュ２８に戻されることに応じて再探索を行なう。古いストアにヒットしたロードを選択し、現在再探索を行なっていないならば再探索を行なう。ストアを選択し、リタイアされたことに応じて再探索を行なう。複数のメモリ動作を選択できる場合、ＬＳ２制御回路６６は、複数のメモリ動作のうち最も古いものを選択する。ＬＳ２制御回路６６は、ポート０を使用している（ポート０マルチプレクサ７２を介して）ならば、ＬＳ１制御回路６４に信号で知らせ、制御回路６４は、ポート０マルチプレクサ７２を通してＬＳ２入力を選択し、そのクロックサイクルでポート０上のＬＳ１バッ

ファ６０からのメモリ動作の選択を不能にする。

【００８０】

ＬＳ２制御回路６６はさらに、スヌープインターフェイス５２を介してバスインターフェイスユニット３７からスヌープ要求を受けるように結合される。一般的に、ＬＳ２バッファ６２内のメモリ動作はスヌープされる。なぜなら、このメモリ動作は既にデータキャッシュ２８を探索しておりスヌープ動作に応じて訂正作業が必要だからである。たとえば、ロードヒット（従属する命令にデータをフォワード済み）は、廃棄されて再び実行されなければならない。ストアは、探索から変更の必要があるキャッシュ状態をストアしている。対照的に、ＬＳ１バッファ６０内のメモリ動作は、データキャッシュ２８を探索しておらずしたがってスヌープの必要はない。

【００８１】

ＬＳ２制御回路６６は、スヌープ要求を受け、ＬＳ２バッファエントリをスヌープ要求に対して調べて、スヌープインターフェイス５２を介してバスインターフェイスユニット３７に応答する。加えて、ＬＳ２制御回路６６は、スヌープに応じてＬＳ２バッファエントリ内で更新を行なってもよい。

【００８２】

一般的に、バッファは、２以上の情報項目を後で検索するためにストアするのに用いられる記憶素子である。バッファは、複数のレジスタ、ラッチ、フリップフロップまたは他のクロックされた記憶装置を含む。その代わりとして、バッファが、適切に配列された１組のランダムアクセスメモリ（ＲＡＭ）セルを含んでいてもよい。バッファは多数のエントリに分割され、各エントリは、バッファの設計対象である１つの情報項目をストアするように設計されている。エントリは、適切な方法で割当および割当解除可能である。たとえば、バッファは、シフト先入れ先出し（ＦＩＦＯ）バッファとして動作してもよく、この場合、エントリは、古いエントリが削除されるときにシフトダウンされる。それに代えて、ヘッドおよびテールポインタを用いてバッファ内の最も古いおよび最も新しいエントリを示してもよく、エントリは、削除されるまでバッファの特定の記憶場所に保持される。図１に示したストア待ち行列４００は、一種のバッファである。本明細書で用いる「制御回路」という用語は、入力に対し動作を行なってこれに応じて出力を生成し上記の動作を実現する、組合せ論理回路、クロックト記憶回路および／またはステートマシンの組合せのことをいう。

【００８３】

なお、ある実施例では、ロード／ストアユニット２６は、ＬＳ１からのストア探索を同じポート上の古いストアのデータ提供と重畳しようとする。これが実施される理由は、ストア探索ではヒット／ミスについてデータキャッシュタグを検査しているだけでデータ記憶内のデータの検索または更新の試みがされていないからである。さらに、上記の説明では、すべてのメモリ動作がＬＳ２バッファ６２内にあるものとして実施例の説明をしているが、これ以外の実施例ではこのような方法で動作が行なわれないかもしれない。たとえば、ロードヒットは実施例によってはＬＳ２バッファ６２内にストアされないことがある。例として、こうした実施例は厳密なメモリ順序が望まれない場合に用いられる。

【００８４】

ストアーロードフォワーディング

図５は、ロード／ストアユニット２６およびデータキャッシュ２８の一部の一実施例を示す。これ以外の実施例が可能であり意図されている。図５の実施例では、ロード／ストアユニット２６は、ＬＳ２バッファ６２、ＬＳ２制御回路６６、データフォワードマルチプレクサ１００、ならびにアドレスおよびウェイ比較器１０２Ａ－１０２Ｂを含む。加えて、図５に示した実施例では、データキャッシュ２８は、ポート１データマルチプレクサ１１０およびポート０データマルチプレクサ１１２を含む。ＬＳ２バッファ６２は、データフォワードマルチプレクサ１００、比較器１０２Ａ－１０２ＢおよびＬＳ２制御回路６６に結合される。ＬＳ２制御回路６６はさらに、マルチプレクサ１００、１１０および１１２に結合される。ＬＳ２制御回路６６はさらに、比較器１０２Ａ－１０２Ｂに結合され

る。比較器１０２Ａ－１０２Ｂは、データキャッシュ２８のポート０および１上に与えられたデータアドレスおよびウェイを受けるように結合される。マルチプレクサ１１２は、結果を結果バス３８ＤＡに与えるように結合され、同様に、マルチプレクサ１１０は、結果を結果バス３８ＤＢに与えるように結合される。結果バス３８ＤＡ－３８ＤＢは、図３に示す結果バス３８Ｄの一実施例を成す。

【００８５】

一般的に、ロード／ストアユニット２６は、ロードメモリ動作の探索がＬＳ２バッファ６２にストアされた古いストアメモリ動作にヒットする場合を処理するように構成される。ロード／ストアユニット２６は、ＬＳ１バッファ６０からデータキャッシュ２８を探索するメモリ動作のデータアドレスのインデックス部を、ＬＳ２バッファ６２内のメモリ動作のデータアドレスのインデックス部と比較する。これらのインデックスが一致しメモリ動作がデータキャッシュ２８内でデータキャッシュ２８と同じウェイにヒットしていれば、探索しているメモリ動作はＬＳ２バッファ６２内のストアにヒットしている。探索しているロードがＬＳ２バッファ６２内のストアにヒットしておりストアデータがＬＳ２バッファ６２で利用できるのであれば、ストアデータはデータキャッシュ２８に送られてキャッシュにあるロードデータの代わりにフォワードされる。他方、探索しているロードが、ストアデータが利用できないＬＳ２バッファ６２内のストアにヒットするかもしれない。このような場合、データキャッシュ２８からのデータフォワードは取消され、ロード記憶動作を選択して、ストアデータが利用できるようになるまでＬＳ２バッファ６２からの再探索を行なう。最終的に、ストアデータはＬＳ２内で利用できるようになり、バッファから、ロードによる再探索中にフォワードされる、または、ストアがデータキャッシュ２８を更新しデータがデータキャッシュ２８からロードによる再探索中にフォワードされる。

【００８６】

一般的に、ストアデータが記憶場所から「利用可能」であるのは、このストアデータが実際にその記憶場所にストアされている場合である。その後のある時点でデータが記憶場所にストアされるかもしれないがまだそこにストアされていない場合、データは、「利用可能でない」、「まだ利用可能でない」または「利用できない」。たとえば、ストアデータは、そのストアデータがストアデータのソースからＬＳ２バッファエントリにまだ転送されていない場合は、ＬＳ２バッファエントリにおいて利用可能でない。ストアデータのソースとは、実行によりストアデータを発生させる命令であり、ストアが対応する命令と同じ命令である（メモリオペランドを宛先として指定する命令）、またはより古い命令である。ストアデータタグは、ストアデータのソースを識別し、よって、実行ユニット４０からの結果タグと比較されてストアデータを収集する。

【００８７】

上記のように、ロードアドレスおよびウェイ表示が、ＬＳ２バッファ６２内のストアアドレスおよびウェイ表示と比較され、古いストアにヒットするロードが検出される。したがって、比較器１０２のような比較器が設けられる。比較器１０２は、データキャッシュ２８の各ポート上のアドレスおよびウェイ表示をＬＳ２バッファ６２内にストアされたデータアドレスおよびウェイ表示と比較するために設けられる。さらに、所望されれば、比較器１０２をＬＳ２バッファ６２にＣＡＭ構成として組込んでもよい。

【００８８】

ストアエントリ上のロードヒットが検出され対応するストアデータが利用可能の場合、ＬＳ２制御回路６６は、データフォワードマルチプレクサ１００を用いてストアデータを選択し、そのデータをポート０マルチプレクサ１１２またはポート１マルチプレクサ１１０のいずれかに、ヒットが検出されたポートに基づいて与える。したがって、データフォワードマルチプレクサ１００は、各ポートに１つずつ、１組の独立したマルチプレクサを含む。加えて、ＬＳ２制御回路６６は、データキャッシュ２８に対して対応する信号をアサートし、データキャッシュ２８が、ヒットしているロードに対しデータキャッシュ２８から読出されたキャッシュデータの代わりにフォワードされたデータを選択できるようにする。

【００８９】
さらに、この実施例は、ＬＳ２バッファ６２を用いるものとして示されているが、他の実施例も意図されている。この他の実施例においては、上記のストアフォワードメカニズムが、従来のストア待ち行列とともに実施され、このストア待ち行列は、データキャッシュ２８を既に探索しているストアメモリ動作のみをストアする。（たとえばストア待ち行列４００をある特定の実施例で用いることができる）。またさらに、ここで示されているマルチプレクサ１１０および１１２はデータキャッシュ２８内にあるが、この回路を所望に応じてロード／ストアユニット２６内で用いてもよい。加えて、ここで示されているマルチプレクサ１００はＬＳ２バッファ６２からデータを選択してフォワードするものであるが、マルチプレクサ１００を省いて、読出エントリ番号をデータを読出したＬＳ２バッファ６２に与えてもよい。これは、ＬＳ２バッファ６２が個別クロックド記憶装置（たとえばレジスタ）ではなくＲＡＭ構成の場合である。

【００９０】
さらに、ある具体的な実現化例では、ロード／ストアユニット２６が、従属性リンクファイルを用いて、対応するストアデータが利用可能でないストアにヒットするロードが検出されたときにデータのフォワードを加速してもよい。このようなロードの検出に応じて、ロード／ストアユニット２６は、ロードのために従属性リンクファイルにエントリを割当てる。従属性リンクファイルエントリは、ストアにヒットしたロードを識別するロード識別子（たとえばリオーダバッファ３２がロードに対応する命令に割当てた命令タグ）およびロードがヒットしたストアに対応するストアデータのソースを識別するストアデータ識別子（たとえばストアデータタグ）をストアする。次に、ロード／ストアユニット２６は、従属性リンクファイル内にストアされたストアデータタグに対し、結果バス３８をモニタする。ストアデータが結果バス３８の１つに与えられていることが検出されると、ロード／ストアユニット２６は、データキャッシュ２８に、対応する結果バスからのデータをデータキャッシュ２８からの結果バスにフォワードさせる。加えて、対応するエントリからのロード識別子が結果タグとしてフォワードされる。なお、従属性リンクファイルは、全く任意選択の性能の向上のためのものである。従属性リンクファイルを用いない実施例も意図されている。

【００９１】
次に図６を参照して、ＬＳ２制御回路６６およびＬＳ２エントリ９４の一実施例の一部のブロック図が示される。これ以外の実施例および特定の実現化例が意図されている。図６の実施例は、比較器１０２ＡＡ、比較器１０２ＡＢ、ＡＮＤゲート１２０、ヒット制御回路１３２、およびデータ転送マルチプレクサ１００を含む。ヒット制御回路１３２はヒットエントリレジスタ１３４を含む。比較器１０２ＡＡは、ポート０からデータアドレスの少なくともインデックス部を受け（参照番号１３６）、かつ、エントリ９４のアドレス－インデックスフィールド９６Ａにストアされたデータアドレスのインデックス部を受けるように結合される。比較器１０２ＡＡは、出力をＡＮＤゲート１２０に与え、ＡＮＤゲート１２０はさらに、エントリ９４からストア有効ビット（ＳＴＶフィールド９６Ｂ）およびヒットビット（Ｈフィールド９６Ｃ）を受けるように結合される。ＡＮＤゲート１２０の出力は、ヒットストア信号としてヒット制御回路１３２に結合され、ヒット制御回路１３２はさらに、ポート０ロード信号（参照番号１４０）、ポート０ヒット信号（参照番号１２２）およびポート０オフセットおよびサイズ情報（参照番号１２４）を受ける。比較器１０２ＡＢは、ウェイフィールド９６Ｅの内容を受けるように結合され、かつ、ポート０ウェイ表示（参照番号１４２）を受けるように結合される。比較器１０２ＡＢは、出力をヒットウェイ信号としてヒット制御回路１３２に与えるように結合される。ヒット制御回路１３２はさらに、データ有効フィールド９６Ｇからデータ有効ビットを受け、オフセットおよびサイズフィールド９６Ｆからオフセットおよびサイズ情報を受けるように結合される。ヒット制御回路１３２は、他のエントリに対応する、同様のヒットストア、ヒットウェイ、データ有効ならびにオフセットおよびサイズ信号を受ける。ヒット制御回路１３２は、リザベーションステーション２２およびリオーダバッファ３２に取消しデータ

ＦＷＤ信号を与え（参照番号１４６）、データキャッシュ２８にＬＳ２信号を選択する（参照番号１４８）ように結合される。加えて、ヒット制御回路１３２は、選択制御をマルチプレクサ１００に与えるように結合される。マルチプレクサ１００は、ストアデータフィールド９６Ｈからのストアデータ（および他のＬＳ２バッファエントリからのストアデータ）を受けるように結合される。

【００９２】

一般に、図６に示した論理では、エントリ９４内のストアに対するポート０上のロードのヒットを検出する。ポート１およびエントリ９４について、ならびに他のエントリに対する上記ポート双方について、同様の論理を用いることができる。より具体的には、比較器１０２ＡＡは、ポート０上のデータアドレスのインデックス部をアドレス－インデックスフィールド９６Ｃのインデックスと比較する。これらのインデックスが一致していれば、比較器１０２ＡＡはその出力信号をアサートする。ＡＮＤゲート１２０は、比較器１０２ＡＡの出力信号を受け、この出力信号をストア有効ビットおよびヒットビットと組合せる。ストア有効ビットは、エントリ９４がストアメモリ動作に対応する情報をストアしているかどうかを示し（なぜならエントリ９４および他のＬＳ２バッファエントリはロードおよびストアいずれかに対応する情報をストアしている可能性があるため）、ヒットビットは、ストアがデータキャッシュ２８を探索したときにデータキャッシュ２８内でヒットしているかどうかを示す。したがって、ＡＮＤゲート１２０が与えるヒットストア信号は、アサートされたときに、ロードインデックスが、データキャッシュ２８内でヒットであるストアインデックスにヒットしたことを示す。

【００９３】

ヒット制御回路１３２は、エントリ９４に対応するヒットストア信号、ポート０に対応する他のヒットストア信号、およびポート０ロード信号１４０を組合せて、ポート０上にメモリ動作のためのデータフォワード信号を生成する。この実施例では、ヒット制御回路１３２は、ロードについて２つの場合を検出する。すなわち（ｉ）ヒットストア信号がアサートされ対応するデータ有効ビット９６Ｇがセットされる。（ii）ヒットストア信号がアサートされ対応するデータ有効ビット９６Ｇがクリアされる。アサートされるヒットストア信号がない場合、またはポート０上のメモリ動作がロードでない場合、ヒット制御回路１３２は、そのメモリ動作には使用されない。この実施例では、同様のヒット制御回路を用いてポート１上のメモリ動作を行なうことができる。

【００９４】

（ｉ）の場合、ヒット制御回路１３２は、データフォワードマルチプレクサ１００に対しマルチプレクサ選択信号を生成し、これにより、データフォワードマルチプレクサ１００は、アサートされたヒットストア信号に対応するＬＳ２バッファエントリのストアデータフィールド９６Ｈからのストアデータを選択する。たとえば、ＡＮＤゲート１２０が生成したヒットストア信号がアサートされる場合、ヒット制御回路１３２は、マルチプレクサ１００が、エントリ９４からストアデータフィールド９６Ｈからのストアデータを選択するようにし、ポート０マルチプレクサ１１２に対応する選択ＬＳ２信号１４８をアサートする。選択されたデータは、データキャッシュ２８によりフォワードされるが、これは図５について先に説明したとおりである。（ii）の場合、ヒット制御回路１３２は、転送（ＦＷＤ）取消信号１４６をリザベーションステーション２２およびリオーダバッファ３２に対してアサートして、これらの構成要素に、そのクロックサイクル中ポート０上のロードに対してフォワードされたデータを無視するよう伝える。

【００９５】

このように、ヒット制御回路１３２は、エントリ９４からのデータを、エントリ９４内のストアインデックスおよびロードインデックスの一致に基づいて、かつ、ストアがデータキャッシュ２８内でヒットであることに基づいて、フォワードする。特に、ロードがデータキャッシュ２８内でヒットしているかどうかまたはロードおよびストアが同じウェイでヒットしているかどうか判断されていないことがある。この情報は、ロードの探索が終了するまで利用できないかもしれない。このような事態は、この実施例では後続のクロッ

クサイクルで生じる。したがって、ヒット制御回路１３２は、ヒットエントリレジスタ１３４内で、フォワードされたデータがもとあったＬＳ２バッファ６２のエントリ番号を収集する。次のクロックサイクル中に、ヒット制御回路１３２は、ＬＳ２バッファ６２からのデータフォワードが正しいかどうか判断する。データフォワードが正しいのは、ロードがデータキャッシュ２８内でヒットし（ポート０ヒット信号１２２で通知される）、ヒットエントリレジスタ１３４により識別されたエントリ内のロードおよびストアのウェイ表示が一致する場合（たとえば比較器１０２ＡＢがポート０のウェイ表示およびウェイフィールド９６Ｅからのウェイ表示が一致することを検出した場合、エントリ９４がヒットエントリレジスタ１３４によって示されている場合）である。フォワーディングが誤りであれば、ヒット制御回路１３２は、フォワード取消信号１４６をアサートして、リザベーションステーション２２および／またはリオーダバッファ３２に、以前にフォワードされたポート０上のデータを無視するよう知らせる。ヒット制御回路１３２は、別のデータフォワード取消信号１４６を与えてフォワーディングを取消すようにしてもよい。これは、（上記のように）データが利用可能でないため、および、ストア（データはここからフォワードされた）と異なるウェイのミスまたはヒットであるロードに対する誤ったフォワーディングのためである。この信号は、同じロードに対して異なる時間にアサートし得る。

【００９６】

図１に関して先に述べたように、ヒット制御回路１３２はさらに、ＬＳ２バッファ６２からのストアデータをロードのためにフォワードすべきかどうかをロードおよびストアに対する（キャッシュライン内の）オフセットおよびサイズ情報を用いて判断し、ロードが読出す少なくとも１バイトをストアにより更新するかどうか決定する。オフセットおよびサイズ情報は上記のようにどのような適切なフォーマットで与えてもよい（たとえばアドレスビットおよびバイトイネーブルマスクの組合せ）。なお、ヒット制御回路１３２がロードおよびストアアドレスのオフセットの一部を比較する場合、その部分を、所望される場合はインデックス部に加えて比較器１０２ＡＡで比較することができる。

【００９７】

さらに、ヒット制御回路１３２は、所与のロードについて２以上のストアのヒットを検出してもよい。ヒット制御回路１３２は、データフォワードのためにロードよりも古い、最も新しいストアを求める。それに代えて、各ＬＳ２バッファエントリが、所与のアドレスを更新するＬＳ２バッファ６２内の最後のストアを識別するバッファ表示のうちの最終を含んでいてもよい。ＡＮＤゲート１２０内でＬＩＢ表示を用いて、ヒットストア信号が、ＬＳ２バッファ６２内の最も新しいストアを除いてアサートされないようにする。このように、複数のヒットの優先順位をつけないようにする。ストアはＬＳ２バッファ６２内に置かれているため、そのＬＩＢビットをセットし、同じアドレスへの古いストアのＬＩＢビットをクリアしてもよい。

【００９８】

なお、ここで示している比較器１０２ＡＢは、ＬＳ２バッファエントリ９４にストアされたウェイ表示をロードのウェイ表示と比較するが、代替実施例においては、ロードのためにデータをフォワードするのに用いるエントリからのウェイ表示を読出して（フォワーディングがインデックス比較およびストアのデータキャッシュでのヒットに基づいている場合）、読出されたウェイ表示をロードウェイ表示と比較して、ロードおよびストアが同じウェイでヒットしていることを確認する。ウェイ表示を、ヒットエントリレジスタ１３４と同様のレジスタにストアして、将来の比較に備えることができる。

【００９９】

さらに、ロードが読出した１以上のバイトが、ロードが読出した１以上の他のバイトについてロードがヒットしたストアによって、更新されないことがある。このような場合、データキャッシュは、ストアデータをキャッシュデータと併合して、ロードが読出したバイトを与える。複数のストアが、ロードが読出したバイトのうち異なるバイトを与えるとき、このロードをリタイアさせて再探索する。複数のストアのうち１つ以上のストアをリタイアさせてデータキャッシュに与えてもよく、これらのストアが更新しロードが読出し

たバイトをデータキャッシュから与えてもよい。その代わりとして、図６の装置が、異なるストアからのバイトを併合してロードデータを与えてもよい。これ以外の実施例では、所望に応じて上記の方法を他のやり方で処理する。

【０１００】

なお、図６に示した論理は例示にすぎない。適切な組合せ論理（ここで示している論理のブール等価物を含む）を用いることができる。なお、エントリ９４はＬＳ２バッファエントリの一例である。エントリ９４は、設計上の選択に応じて図６に示したものの上にさらなる情報をストアしてもよい。

【０１０１】

次に図７を参照して、ＬＳ１バッファ６０からデータキャッシュ２８を探索するメモリ動作のパイプラインの例を示すタイミング図が示されている。異なるパイプラインを用いたこれ以外の実施例が可能であり意図されている。図７において、クロックサイクルは垂直方向の実線で区切られている。水平方向の点線も示されている。プロセッサ１０の他の部分に関連するパイプライン段を示して、他の構成要素のロード／ストアユニット２６へのインターフェイスを示している。

【０１０２】

クロックサイクルＣＬＫ０は、メモリ動作を指定する命令のデコード／ディスパッチサイクルである。クロックサイクルＣＬＫ０において、この命令をデコードするデコードユニット２０は、メモリ動作に関しロード／ストアユニット２６に信号を送る。ＬＳ１制御回路６４は、対応する命令に対するデコード／ディスパッチ段においてメモリ動作のためにＬＳ１バッファエントリを割当てる。加えて、デコードユニット２０は、デコードされた命令を対応するリザベーションステーション２２に送る。

【０１０３】

クロックサイクルＣＬＫ１において、アドレス生成ユニットは、このメモリ動作のためにデータアドレスを生成し、このデータアドレスをロード／ストアユニット２６に送る。このクロックサイクル中に、メモリ動作が、ＬＳ１制御回路６４による（与えられたデータアドレスにより行なう）スキャンに加わり、データキャッシュ２８を探索するために選択される。このようにして、メモリ動作はＬＳ１パイプラインのスキャンパイプライン段にある。

【０１０４】

クロックサイクルＣＬＫ２において、データアドレスはデータキャッシュ２８に送られる。クロックサイクルＣＬＫ２内の矢印で示しているように、メモリ動作は、ＬＳ１バッファ６０から、クロックサイクルＣＬＫ２の最後に一時バッファ６８に移動する。メモリ動作は、クロックサイクルＣＬＫ２の間は、ＬＳ１パイプラインのデータキャッシュ段へのアドレスにある。

【０１０５】

クロックサイクルＣＬＫ３において、データアドレスはデータキャッシュ２８にアクセスする。メモリ動作に対応するデータは（メモリ動作がロードの場合）、クロックサイクルＣＬＫ３の最後にフォワードされる。より具体的には、ロードアドレスのインデックス部がＬＳ２バッファ６２内のストアアドレスのインデックス部と一致し、かつ、ストアがデータキャッシュ２８内でヒットしていれば、ＬＳ２バッファ６２からのデータは、クロックサイクルＣＬ３においてキャッシュデータの代わりにフォワードされる。加えて、メモリ動作は、一時バッファ６８からＬＳ２バッファ６２に送られる。メモリ動作は、クロックサイクルＣＬＫ３においてはキャッシュアクセス段にある。

【０１０６】

クロックサイクルＣＬＫ４において、メモリ動作に従属する命令（メモリ動作がロードの場合）が実行される。したがって、図７に示したパイプラインでは、３クロックサイクルアドレス生成が、従属動作実行ロードレイテンシに対して与えられている。加えて、メモリ動作は、クロックサイクル４においては応答パイプライン段にある。データキャッシュ２８は、この応答段においてヒット／ミス情報（ヒットのウェイ表示を含む）および物

理アドレスを与える。このように、ＬＳ２制御回路６６は、ヒット／ミス情報および物理アドレスをこの応答段においてメモリ動作と関連付ける。さらに、ロードに対するヒット／ミス表示およびウェイ表示を用いて、クロックサイクルＣＬＫ３においてＬＳ２バッファ６２からフォワードされたデータを確認する（利用できる場合）。フォワードされたデータが、ロードがミスであるまたは異なるウェイでヒットしたために誤ってフォワードされたものであれば、フォワード取消信号がアサートされる。

【０１０７】

クロックサイクルＣＬＫ５において、メモリ動作は応答２パイプライン段にある。この段の間、メモリ動作がアクセスしたキャッシュラインに割当てられたミスアドレスバッファエントリを識別するミスアドレスバッファタグ（メモリ動作がミスの場合）が、データキャッシュ２８により与えられる。このように、ＬＳ２制御回路６６は、データキャッシュ２８から受けたＭＡＢタグを応答２段のメモリ動作に関連付ける。

【０１０８】

次に図８を参照して、ＬＳ２バッファ６２からデータキャッシュ２８を再探索するメモリ動作のパイプラインの例を示すタイミング図が示されている。異なるパイプラインを用いたこれ以外の実施例が可能であり意図されている。図８において、クロックサイクルは垂直方向の実線で区切られている。水平方向の点線も示されている。プロセッサ１０の他の部分に関連するパイプライン段を示して、他の構成要素のロード／ストアユニット２６へのインターフェイスを示している。

【０１０９】

クロックサイクルＣＬＫ０において、メモリ動作が、ＬＳ２バッファエントリのスキャンに加わり、データキャッシュ２８を再探索するために選択される。クロックサイクルＣＬＫ０の下の矢印によって示しているように、このメモリ動作が選択されるのは、メモリ動作についてＭＡＢタグに一致する充填タグを受けた場合、メモリ動作がＬＳ２バッファ６２内のより古いストアにヒットしているロードである場合（データは過去の探索に利用可能でなかった）、または、メモリ動作がリオーダバッファ３２がリタイアしたストアである場合、である。

【０１１０】

クロックサイクルＣＬＫ１において、スキャン１段で選択されたメモリ動作はスキャン２段に進む。スキャン２段において、メモリ動作はマルチプレクサ７６および７２を通して選択されてデータキャッシュ２８に送られる。したがって、ＬＳ２制御回路６６は、マルチプレクサ７６を通してスキャン２段のメモリ動作を選択する。クロックサイクルＣＬＫ２、ＣＬＫ３、ＣＬＫ４およびＣＬＫ５は、ＬＳ２バッファ再探索パイプラインのデータキャッシュへのアドレス、キャッシュアクセス、応答、および応答２段であり、先に述べた対応する段と同様である。したがって、この実施例では、バスインターフェイスユニット３７は、対応するデータを与える４クロック前にＭＡＢタグを与えて、対応する充填データにアクセスするロードの選択が、充填データがデータキャッシュ２８に到達した（従って充填データはフォワードされた）クロックサイクルのキャッシュアクセス段において行なわれるようにする。

【０１１１】

なお、図７および８の点線の上の命令パイプライン段と点線の下のメモリ動作パイプライン段との間のタイミングを、図７および８に示したものから延長してもよい。たとえば、図７では、アドレスは、厳密にデコード／ディスパッチサイクル直後のクロックサイクルで生成されない場合がある。その代わりに、オペランドが利用できないまたはアドレス生成のために古い命令が選択される場合がある。さらに、メモリ動作は、アドレスが与えられたクロックサイクル中にアクセスのためにスキャンされないかもしれず、他の古いメモリ動作がその代わりにスキャンされるかもしれない。

【０１１２】

次に図９を参照して、ヒット制御回路１３２が、ロードの探索中にデータを選択してＬＳ２バッファ６２からフォワードする実施例の動作を示すフローチャートが示される（た

とえばロードの探索のキャッシュアクセスパイプライン段）。他の実施例が可能であり意図されている。図９に示したステップは、理解しやすくするために特定の順序で示されているが、その他の適切な順序を用いてもよい。加えて、ステップを、ヒット制御回路１３２内で組合せ論理により並列に実行してもよい。

【０１１３】

ヒット制御回路１３２は、ロードがストアにヒットしストアがキャッシュヒットであるかどうかを判断する（判断ブロック１５０）。より具体的には、ヒット制御回路１３２は、ロードインデックスがストアインデックスに一致している（かつオフセットおよびサイズ情報が一致している）場合にロードがストアにヒットしていると判断する。この判断は、ロードヒット情報およびウェイ表示を後続のクロックサイクルで利用できるときに、その正誤が検査される（図１０に示す）。判断ブロック１５０の結果が「イエス」であれば、ヒット制御回路１３２は、データキャッシュ２８に信号を送り、キャッシュデータの代わりに、ＬＳ２バッファ６２から与えられるデータを選択し、ヒットであるエントリからのデータをマルチプレクスし（ステップ１５２）、ヒットであるＬＳ２バッファエントリをヒットエントリレジスタ１３４に記録する（ステップ１５４）。判断ブロック１５０の結果が「ノー」であれば、ヒット制御回路１３２はロードに関してそれ以上の作業は行なわない。

【０１１４】

図１０は、ヒット制御回路１３２がロードの探索中にＬＳ２バッファ６２からのフォワーディングを検査する実施例の動作を示すフローチャートである（たとえばロードの探索の応答パイプライン段）。これ以外の実施の形態が可能であり意図されている。図１０に示したステップは理解しやすくするために特定の順序で示されているが、これ以外の適切な順序を用いてもよい。加えて、これらのステップをヒット制御回路１３２内で組合せ論理により並列に実行してもよい。

【０１１５】

ヒット制御回路１３２は、エントリがヒットエントリレジスタ１３４に記録されているかどうか判断する（判断ブロック１６０）。たとえば、ヒットエントリレジスタ１３４は、データがインデックス比較およびストアがヒットであることに基づいてフォワードされたときにセットされ、ロードのヒットの検査およびウェイ表示の一致後にリセットされる、有効ビットを含み得る。エントリがヒットエントリレジスタ１３４に記録されていなければ、ヒット制御回路１３２はロードに関してさらなる作業は行なわない。エントリがヒットエントリレジスタ１３４に記録されていれば、ヒット制御回路１３４は、ロードウェイ表示がヒットエントリレジスタ１３４に記録されたエントリのストアウェイ表示に一致しているかどうか、および、ロードがヒットかどうか判断する（判断ブロック１６２）。ロードがミスであるまたはロードウェイ表示がストアウェイ表示に一致していなければ、ヒット制御回路１３２は、フォワード取消信号をアサートする（ステップ１６４）。ロードがヒットでありロードウェイ表示がヒットエントリレジスタ１３４に記録されたエントリのストアウェイ表示に一致していれば、ヒット制御回路１３２はロードに関しさらなる作業は行なわない。

【０１１６】

なお、ここでは種々の信号のアサートについて述べている。本明細書では、ある信号は、この信号が特定の状態を示す値を伝える場合に「アサートされる」。逆に、ある信号は、その信号が特定の状態がないことを示す値を伝える場合に「デアサートされる」。ある信号は、論理ゼロ値を伝える場合または逆に論理１値を伝える場合にアサートされると定義してもよい。

【０１１７】

コンピュータシステム

次に図１１を参照して、バスブリッジ２０２を通して種々のシステム構成要素に結合されたプロセッサ１０を含むコンピュータシステム２００の一実施例のブロック図が示される。これ以外の実施例が可能であり意図されている。ここで示しているシステムでは、メ

インメモリ２０４はメモリバス２０６を通してバスブリッジ２０２に結合され、グラフィックスコントローラ２０８はＡＧＰバス２１０を通してバスブリッジ２０２に結合される。最後に、複数のＰＣＩ装置２１２Ａ－２１２ＢがＰＣＩバス２１４を通してバスブリッジ２０２に結合される。第２のバスブリッジ２１６をさらに設けて、ＥＩＳＡ／ＩＳＡバス２２０を通して１以上のＥＩＳＡまたはＩＳＡ装置２１８への電気的インターフェイスに対応するようにしてもよい。プロセッサ１０は、ＣＰＵバス２２４を通してバスブリッジ２０２に結合され、任意的選択のＬ２キャッシュ２２８に結合される。ＣＰＵバス２２４およびＬ２キャッシュ２２８へのインターフェイスは、バスインターフェイスユニット３７が結合されたインターフェイスを含むものでもよい。

【０１１８】

バスブリッジ２０２は、プロセッサ１０、メインメモリ２０４、グラフィックスコントローラ２０８およびＰＣＩバス２１４に取付けられた装置間のインターフェイスとなる。バスブリッジ２０２に接続されたこれら装置のうち１つから動作を受けると、バスブリッジ２０２はその動作のターゲットを識別する（たとえば特定の装置、または、ＰＣＩバス２１４の場合はターゲットがＰＣＩバス２１４にあることを識別）。バスブリッジ２０２は、その動作をターゲットの装置に送る。一般的に、バスブリッジ２０２は、ある動作を、ソースである装置またはバスが用いるプロトコルから、ターゲットである装置またはバスが用いるプロトコルに変換する。

【０１１９】

第２のバスブリッジ２１６は、ＰＣＩバス２１４に対してＩＳＡ／ＥＩＳＡバスへのインターフェイスを与えることに加え、所望に応じてさらなる機能を取入れてもよい。入出力コントローラ（図示せず）は、第２のバスブリッジ２１６外のものでもバスブリッジ２１６と一体化されたものであっても、コンピュータシステム２００内に設けられて、所望に応じて、キーボードおよびマウス２２２のためのならびに種々のシリアルおよびパラレルポートのための動作支援を行なってもよい。他の実施例において、外部キャッシュユニット（図示せず）をさらに、プロセッサ１０およびバスブリッジ２０２間でＣＰＵバス２２４に結合してもよい。その代わりに、外部キャッシュをバスブリッジ２０２に結合し、外部キャッシュのためのキャッシュ制御回路をバスブリッジ２０２と一体化させてもよい。Ｌ２キャッシュ２２８が、プロセッサ１０の裏側の構成として示されている。なお、Ｌ２キャッシュ２２８はプロセッサ１０から離れていてもよく、プロセッサ１０とともにカートリッジに組込まれていてもよく（たとえばスロット１またはスロットＡ）、またはプロセッサ１０とともに半導体基板上に集積されてもよい。

【０１２０】

メインメモリ２０４は、アプリケーションプログラムがストアされているメモリであり、このメモリから、プロセッサ１０は主として実行する。適切なメインメモリ２０４は、ＤＲＡＭ（ダイナミックランダムアクセスメモリ）を含む。たとえば、ＳＤＲＡＭ（同期型（synchronous）ＤＲＡＭ）またはランバス(Rambus）ＤＲＡＭ（ＲＤＲＡＭ）の複数のバンクが適切である。

【０１２１】

ＰＣＩ装置２１２Ａ－２１２Ｂは、例えばネットワークインターフェイスカード、ビデオアクセラレータ、オーディオカード、ハードまたはフロッピー（Ｒ）ディスクドライブまたはドライブコントローラ、ＳＣＳＩ（スモールコンピュータシステムインターフェイス）アダプタおよびテレホンカードといった種々の周辺装置の例である。同様に、ＩＳＡ装置２１８は、モデム、サウンドカードおよび種々のデータ取得カード（たとえばＧＰＩＢまたはフィールドバスインターフェイスカード）といった種々の周辺装置の例である。

【０１２２】

グラフィックスコントローラ２０８は、テキストおよび画像のディスプレイ２２６へのレンダリングを制御するために設けられる。グラフィックスコントローラ２０８は、一般的に当該技術ではメインメモリ２０４におよびメインメモリ２０４から効果的にシフト可能な３次元データ構造をレンダリングする典型的なグラフィックスアクセラレータを実現

している。グラフィックスコントローラ２０８はしたがって、バスブリッジ２０２内のターゲットインターフェイスにアクセスを要求しアクセスを受けることができ、従ってメインメモリ２０４へのアクセスを得るという点において、ＡＧＰバス２１０のマスタである。専用グラフィックスバスが、メインメモリ２０４からデータを迅速に取出すことができるようにする。いくつかの動作では、グラフィックスコントローラ２０８がさらに、ＡＧＰバス２１０上でＰＣＩプロトコルトランザクションを生成するように構成されていてもよい。バスブリッジ２０２のＡＧＰインターフェイスはこのように、ＡＧＰプロトコルトランザクションならびにＰＣＩプロトコルターゲットおよびイニシエータトランザクション双方を支援する機能を有する。ディスプレイ２２６は、画像またはテキストを表示できる電子表示装置である。適切なディスプレイ２２６は、陰極線管（「ＣＲＴ」）、液晶ディスプレイ（「ＬＣＤ」）などを含む。

【０１２３】

上記の説明ではＡＧＰ、ＰＣＩ、およびＩＳＡまたはＥＩＳＡバスを例として用いているが、所望に応じていかなるバスアーキテクチャに代えてもよい。さらに、コンピュータシステム２００は、さらなるプロセッサ（たとえばコンピュータシステム２００の任意的選択構成要素としてのプロセッサ１０ａ）を含む多重処理コンピュータシステムでもよい。プロセッサ１０ａはプロセッサ１０と同様のものである。より具体的には、プロセッサ１０ａは、プロセッサ１０をそのままコピーしたものである。プロセッサ１０ａは独立したバス（図１１に示したもの）を介してバスブリッジ２０２に接続されていてもよく、プロセッサ１０とＣＰＵバス２２４を共有していてもよい。さらに、プロセッサ１０ａはＬ２キャッシュ２２８と同様の任意的選択のＬ２キャッシュ２２８ａに結合されていてもよい。

【０１２４】

次に図１２を参照して、コンピュータシステム３００のもう１つの実施例が示されている。これ以外の実施例が可能であり意図されている。図１２に示した実施例では、コンピュータシステム３００は、数個の処理ノード３１２Ａ、３１２Ｂ、３１２Ｃおよび３１２Ｄを含む。各処理ノードは、それぞれ処理ノード３１２Ａ－３１２Ｄ内に含まれたメモリコントローラ３１６Ａ－３１６Ｄを介してメモリ３１４Ａ－３１４Ｄ各々に結合される。加えて、処理ノード３１２Ａ－３１２Ｄは、処理ノード３１２Ａ－３１２Ｄ間の連絡に用いられるインターフェイス論理を含む。たとえば、処理ノード３１２Ａは、処理ノード３１２Ｂとの通信のためのインターフェイス論理３１８Ａ、処理ノード３１２Ｃとの通信のためのインターフェイス論理３１８Ｂ、および、別のさらなる処理ノード（図示せず）との通信のための第３のインターフェイス論理３１８Ｃを含む。同様に、処理ノード３１２Ｂは、インターフェイス論理３１８Ｄ、３１８Ｅおよび３１８Ｆを含み、処理ノード３１２Ｃは、インターフェイス論理３１８Ｇ、３１８Ｈおよび３１８Ｉを含み、処理ノード３１２Ｄは、インターフェイス論理３１８Ｊ、３１８Ｋおよび３１８Ｌを含む。処理ノード３１２Ｄは、複数の入出力装置（たとえばデイジーチェーン構成の装置３２０Ａ－３２０Ｂ）とインターフェイス論理３１８Ｌを介して通信するよう結合されている。同様にこれ以外の処理ノードが他のＩ／Ｏ装置と通信してもよい。

【０１２５】

処理ノード３１２Ａ－３１２Ｄは、処理ノード間通信のためのパケットに基づいたリンクを実現する。この実施例において、このリンクは、単方向ラインの組として実現される。（たとえばライン３２４Ａは処理ノード３１２Ａから処理ノード３１２Ｂにパケットを転送するために用いられ、ライン３２４Ｂは処理ノード３１２Ｂから処理ノード３１２Ａにパケットを転送するために用いられる。）図１２に示されているように、これ以外のライン３２４Ｃ－３２４Ｈの組を用いて他の処理ノード間でのパケットの転送を行なう。一般的に、ライン３２４の組各々は、１以上のデータライン、このデータラインに対応する１以上のクロックラインおよび送られているパケットの種類を示す１以上の制御ラインを含む。リンクは、処理ノード間の通信に対してはキャッシュコヒーレントの態様で動作し、または、処理ノードおよびＩ／Ｏ装置間の（またはバスブリッジからＰＣＩバスまたは

ＩＳＡバスといった従来の構成のＩ／Ｏバスへの）通信については、非コヒーレントな態様で動作する。さらに、リンクは、図示のようなＩ／Ｏ装置間のデイジーチェーン構成を用いて非コヒーレントな態様で動作してもよい。なお、１つの処理ノードから別の処理ノードに転送されるパケットは、１以上の中間ノードを通過し得る。たとえば、処理ノード３１２Ａにより処理ノード３１２Ｄに転送されるパケットは、図１２に示すように、処理ノード３１２Ｂまたは処理ノード３１２Ｃを通過する。任意の適切な経路指定アルゴリズムを用いることができる。コンピュータシステム３００のこれ以外の実施例では、図１２に示した実施例よりも多いまたは少ない処理ノードを含み得る。

【０１２６】

一般的に、パケットは、ノード間のライン３２４上で１以上のビットタイムとして転送される。ビットタイムとは、対応するクロックライン上のクロック信号の立上がりまたは立下がりエッジである。パケットは、トランザクション開始のためのコマンドパケット、キャッシュコヒーレント維持のための探索パケット、および探索およびコマンドへの応答のための応答パケットを含み得る。

【０１２７】

処理ノード３１２Ａ－３１２Ｄは、メモリコントローラおよびインターフェイス論理に加えて、１以上のプロセッサを含む。一般的にいえば、処理ノードは少なくとも１つのプロセッサを含み、任意として、所望に応じてメモリおよび他の論理との通信のためのメモリコントローラを含む。具体的には、処理ノード３１２Ａ－３２１Ｄはプロセッサ１０を含む。外部インターフェイスユニット４６は、ノード内のインターフェイス論理３１８およびメモリコントローラ３１６を含む。

【０１２８】

メモリ３１４Ａ－３１４Ｄは、適切な記憶装置を含む。たとえば、メモリ３１４Ａ－３１４Ｄは、１以上のランバスＤＲＡＭ（ＲＤＲＡＭ）、同期型ＤＲＡＭ（ＳＤＲＡＭ）、スタティックＲＡＭなどを含む。コンピュータシステム３００のアドレス空間は、メモリ３１４Ａ－３１４Ｄ間で分割される。各処理ノード３１２Ａ－３１２Ｄは、どのアドレスがどのメモリ３１４Ａ－３１４Ｄにマッピングされているかを判断して、特定のアドレスに対するメモリ要求をどの処理ノード３１２Ａ－３１２Ｄに送るかを判断するために用いるメモリマップを含む。ある実施例では、コンピュータシステム３００内のアドレスに対するコヒーレントなポイントは、アドレスに対応するバイトをストアするメモリに結合されたメモリコントローラ３１６Ａ－３１６Ｄである。言い換えれば、メモリコントローラ３１６Ａ－３１６Ｄは、対応するメモリ３１４Ａ－３１４Ｄへの各メモリアクセスが確実にキャッシュコヒーレントな態様で生じるようにする。メモリコントローラ３１６Ａ－３１６Ｄは、メモリ３１４Ａ－３１４Ｄへのインターフェイスのための制御回路を含む。加えて、メモリコントローラ３１６Ａ－３１６Ｄは、メモリ要求を待ち行列に入れるための要求待ち行列を含む。

【０１２９】

一般的に、インターフェイス論理３１８Ａ－３１８Ｌは、リンクからパケットを受けるためかつそのリンク上で転送されるパケットをバッファするための種々のバッファを含む。コンピュータシステム３００は、パケット転送のために適切なフロー制御メカニズムを用いる。たとえば、ある実施例では、各インターフェイス論理３１８は、そのインターフェイス論理が接続されたリンクの他端の受信装置内のバッファの各種類の数のカウントをストアする。インターフェイス論理は、受けたインターフェイス論理にパケットをストアする空きのバッファがないのであればパケットを転送しない。受けたバッファをパケットを送出することによって空にすれば、受けたインターフェイス論理は送ったインターフェイス論理にメッセージを送信してバッファが空になっていることを知らせる。こうしたメカニズムのことを「クーポンに基づいた」システムと呼ぶ。

【０１３０】

Ｉ／Ｏ装置３２０Ａ－３２０Ｂは、任意の適切なＩ／Ｏ装置である。たとえば、Ｉ／Ｏ装置３２０Ａ－３２０Ｂは、ネットワークインターフェイスカード、ビデオアクセラレー

タ、オーディオカード、ハードまたはフロッピー（Ｒ）ディスクドライブまたはドライブコントローラ、ＳＣＳＩ（スモールコンピュータシステムインターフェイス）アダプタおよびテレホンカード、モデム、サウンドカード、ならびに種々のデータ取得カード（ＧＰＩＢまたはフィールドバスインターフェイスカードなど）を含む。

【０１３１】

上記の開示を十分に理解すれば当業者には多数の変形例および修正例が明らかになるであろう。前掲の特許請求の範囲はこうした変形例および修正例すべてを含むと解釈すべきものである。

【図面の簡単な説明】

【図１】　ストア待ち行列の一実施例のブロック図である。

【図２】　プロセッサの一実施例のブロック図である。

【図３】　図２に示したデコードユニット、リザベーションステーション、機能ユニット、リオーダバッファ、ロード／ストアユニット、データキャッシュおよびバスインターフェイスユニットの一実施例を示すブロック図であり、相互接続の一実施例を強調している。

【図４】　図２および３に示したロード／ストアユニットの一実施例のブロック図である。

【図５】　ロード／ストアユニットおよびデータキャッシュの一実施例の一部のブロック図である。

【図６】　図５に示した制御回路の一部を示すブロック図である。

【図７】　一実施例に従う図４に示したＬＳ１バッファから選択されるメモリ動作に対応するタイミング図である。

【図８】　一実施例に従う図４に示したＬＳ２バッファから選択されるメモリ動作に対応するタイミング図である。

【図９】　図６に示した制御回路の一実施例の、ストアアドレスにヒットするロードアドレスの検出中の動作を示すフローチャートである。

【図１０】　図６に示した制御回路の一実施例の、ロードアドレスがストアアドレスにヒットすることを検査する際の動作を示すフローチャートである。

【図１１】　コンピュータシステムの第１の実施例のブロック図である。

【図１２】　コンピュータシステムの第２の実施例のブロック図である。

【誤訳訂正２】

【訂正対象書類名】図面

【訂正対象項目名】図１

【訂正方法】変更

【訂正の内容】

【図１】

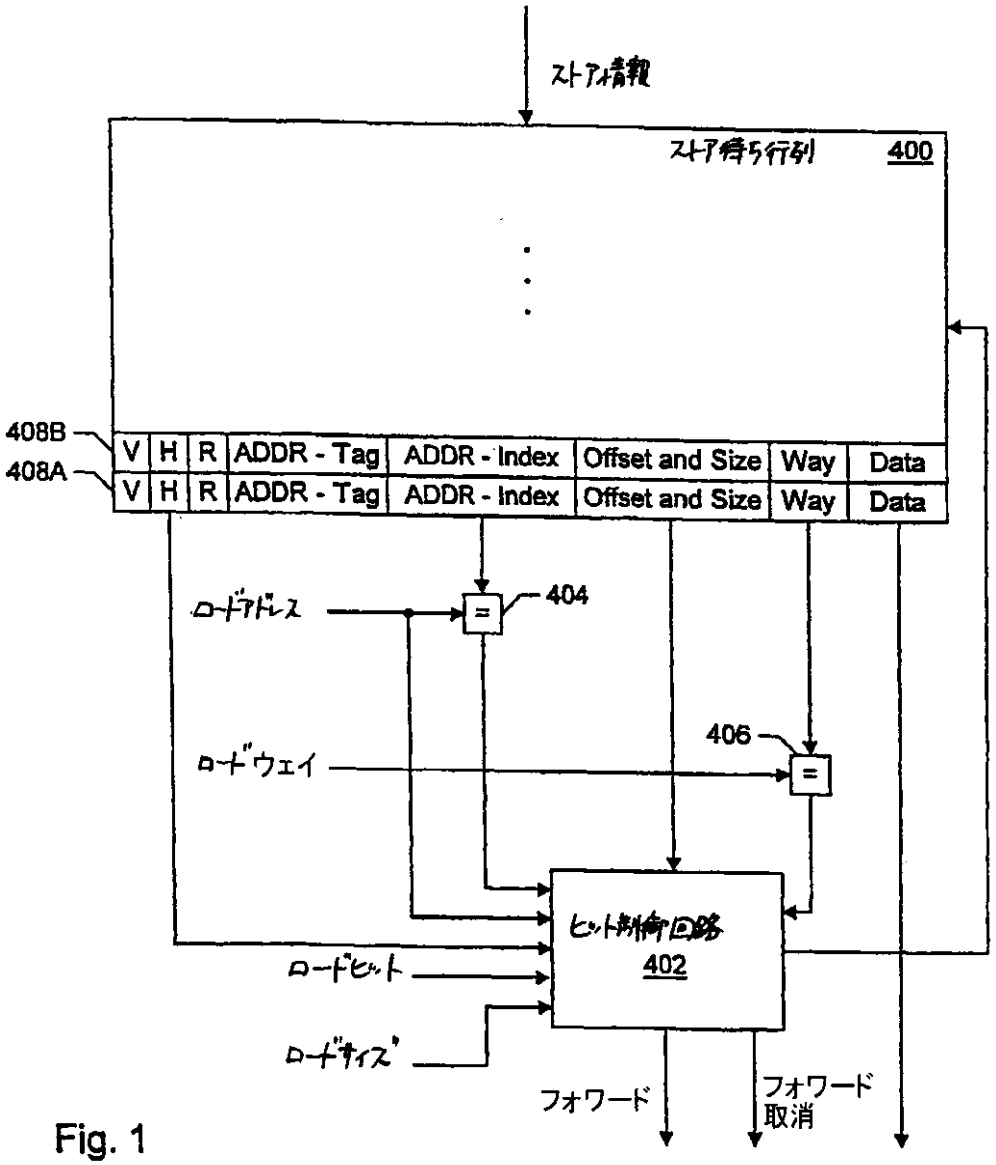

Fig. 1

【誤訳訂正３】
【訂正対象書類名】図面
【訂正対象項目名】図４
【訂正方法】変更
【訂正の内容】

【図４】

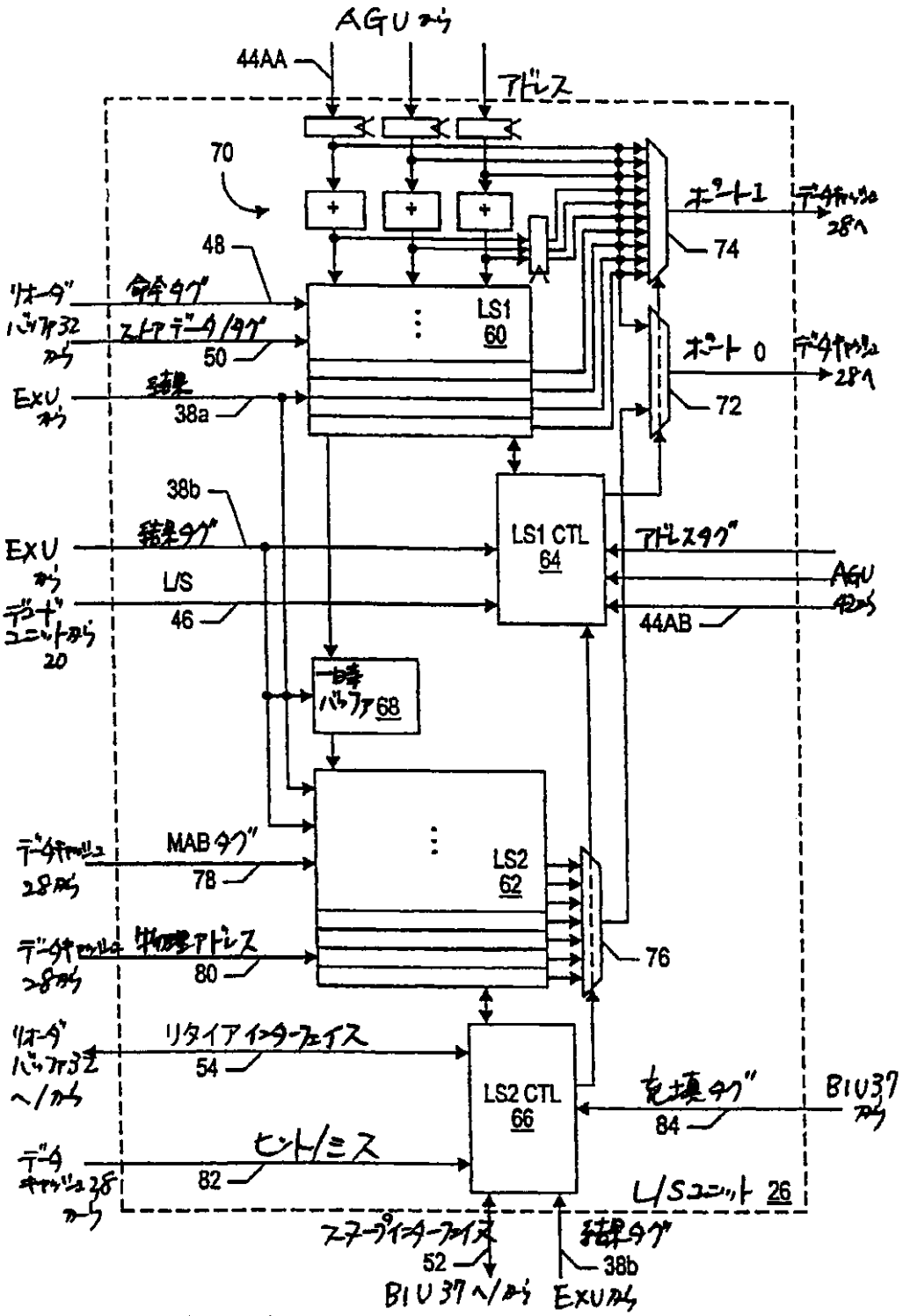

Fig. 4

【誤訳訂正４】
【訂正対象書類名】図面
【訂正対象項目名】図５
【訂正方法】変更
【訂正の内容】

【図５】

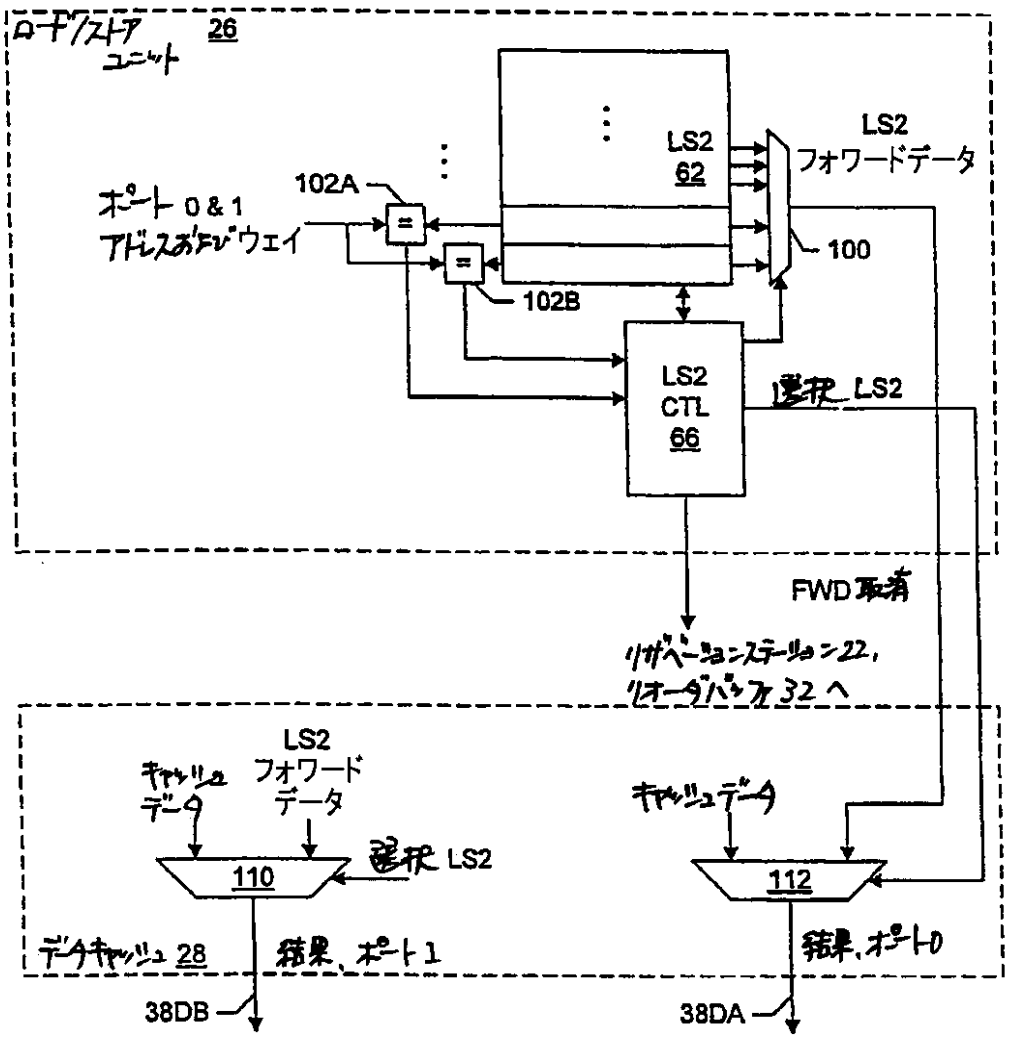

Fig. 5

【誤訳訂正５】
【訂正対象書類名】図面
【訂正対象項目名】図６
【訂正方法】変更
【訂正の内容】

【図６】

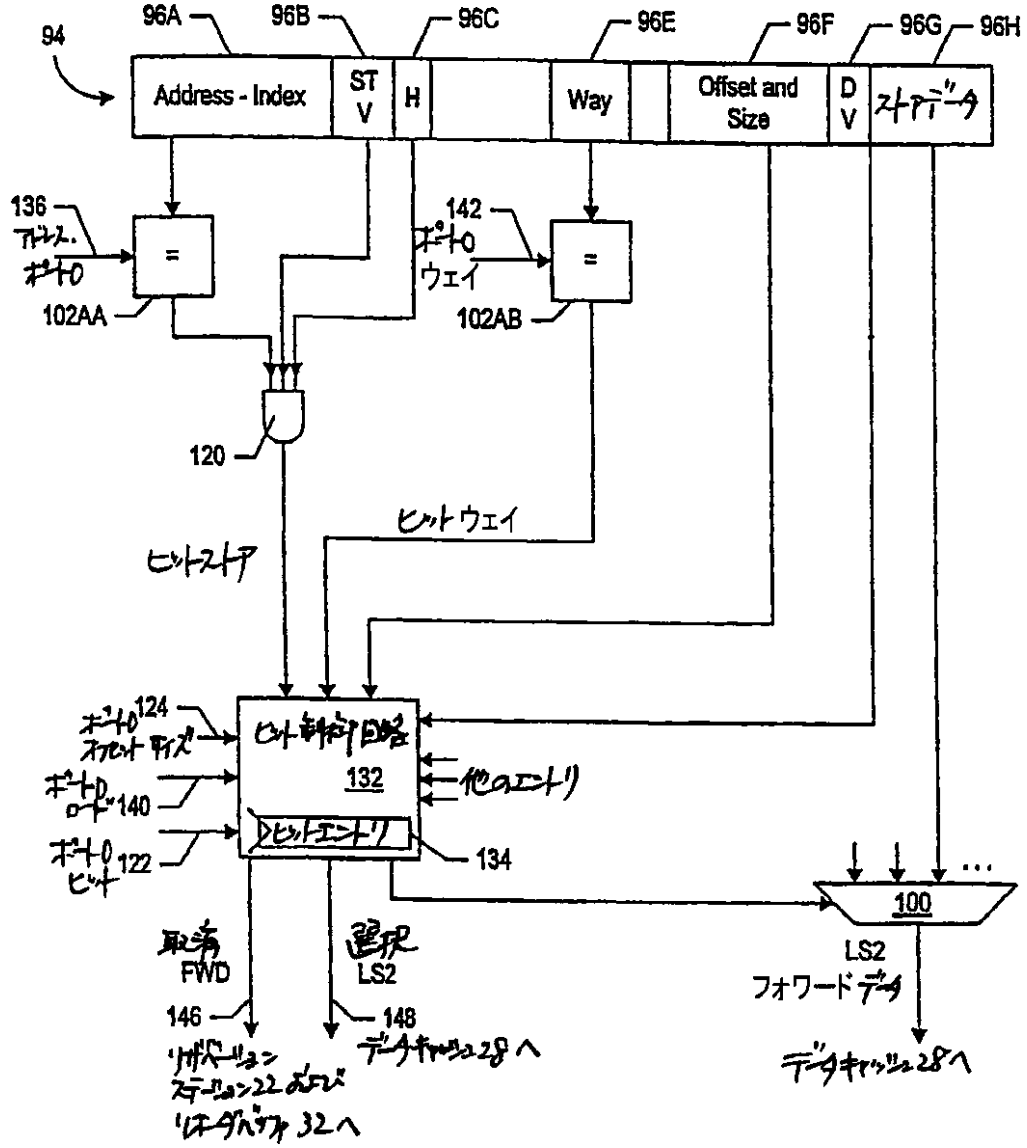

Fig. 6

【誤訳訂正６】
【訂正対象書類名】図面
【訂正対象項目名】図７
【訂正方法】変更
【訂正の内容】

【図７】

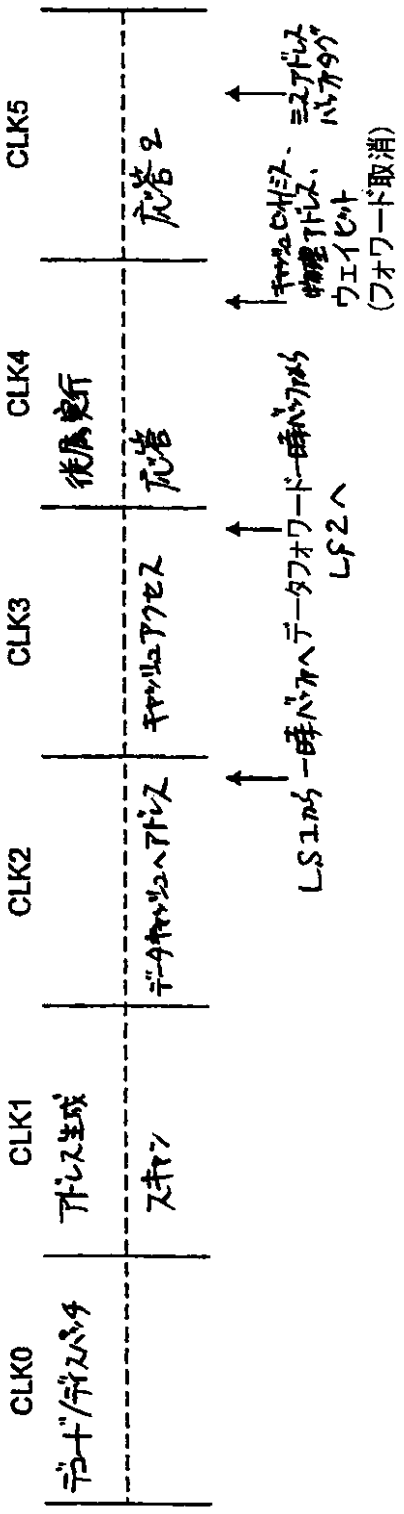

Fig. 7

【誤訳訂正７】
【訂正対象書類名】図面
【訂正対象項目名】図８
【訂正方法】変更
【訂正の内容】

【図８】

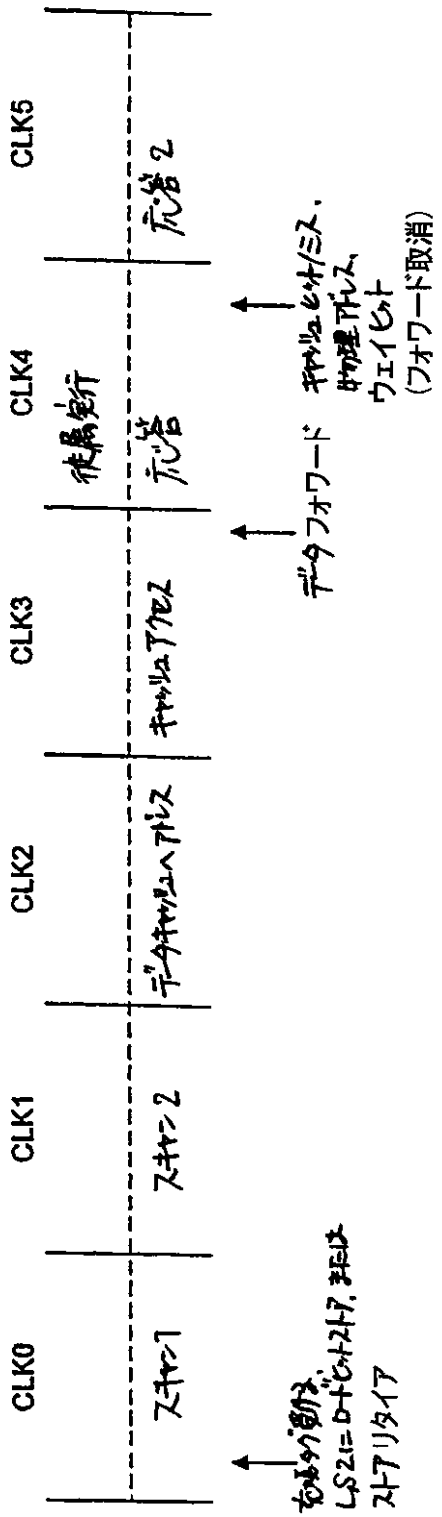

Fig. 8

【誤訳訂正８】
【訂正対象書類名】図面
【訂正対象項目名】図１０
【訂正方法】変更
【訂正の内容】

【図１０】

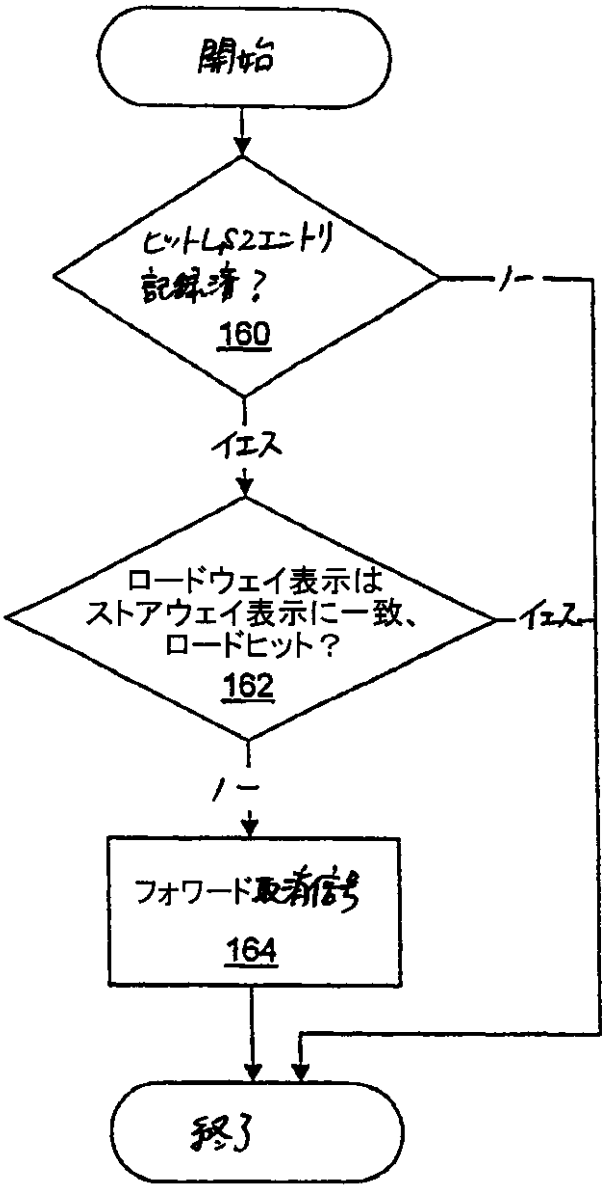
開始
ヒットしS2エントリ
記録済？
160
ノー
イエス
イエス
ロードウェイ表示は
ストアウェイ表示に一致、
ロードヒット？
162
イエス
ノー
フォワード取消信号
164
終了

Fig. 10