(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4700773号
(P4700773)

(45)発行日　平成23年6月15日(2011.6.15)　(24)登録日　平成23年3月11日(2011.3.11)

(51)Int.Cl.　　F I
G06F 15/177 (2006.01)　　G06F 15/177 682F
G06F 15/167 (2006.01)　　G06F 15/177 682J
　　G06F 15/177 672F
　　G06F 15/167 B

請求項の数 25　外国語出願　(全 73 頁)

(21)出願番号　特願平10-340924
(22)出願日　平成10年10月26日(1998.10.26)
(65)公開番号　特開平11-282820
(43)公開日　平成11年10月15日(1999.10.15)
審査請求日　平成17年10月5日(2005.10.5)
審判番号　不服2008-19108(P2008-19108/J1)
審判請求日　平成20年7月28日(2008.7.28)
(31)優先権主張番号　08/957298
(32)優先日　平成9年10月24日(1997.10.24)
(33)優先権主張国　米国(US)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT-PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000
(74)代理人　100059959
弁理士　中村　稔
(74)代理人　100067013
弁理士　大塚　文昭
(74)代理人　100084009
弁理士　小川　信夫

最終頁に続く

(54)【発明の名称】スイッチをベースとするマルチプロセッサシステムに使用するための順序サポート機構

(57)【特許請求の範囲】
【請求項１】
複数の接続されたプロセッサノード及び共用メモリを有するコンピュータシステムであって、当該ノードのそれぞれが少なくとも１つのプロセッサと共用メモリの一部分とを含んでおり、少なくとも一つのプロセッサは複数の、共用メモリにアクセスするためのメモリ要求コマンドを発生し、コンピュータシステムは、複数の通信チャンネルを有し、当該通信チャンネルは、ノードを相互接続し、コンピュータシステムを通って前記少なくとも１つのプロセッサによって発生されたメモリ要求コマンド及び１つあるいはそれより多いメモリ応答コマンドへの論理的に独立しているパスを与え、それに対する応答として発生されたデータ・メッセージを戻すものである、コンピュータシステムにおいて、
前記共用メモリへのメモリ要求コマンドに応じて発生された１つあるいはそれより多いメモリ応答コマンドを順序付ける、各ノードに配置されたシリアル化ポイントと、
第1の順序付けされた通信チャンネルであって、当該第1の順序付けされた通信チャンネルを通じて、前記順序付けされたメモリ応答コマンドが発送される、第1の順序付けされた通信チャンネルと、
第２の順序付けされていない通信チャンネルであって、当該第２の順序付けされていない通信チャンネルを通じて、戻りデータ・メッセージが発送される、第２の順序付けされていない通信チャンネルと、
各ノードに配置されたトランザクション追跡テーブルであって、当該トランザクション追跡テーブルは、保留中の要求に関する共用メモリのアドレス、コマンド、及びコマンダ

ＩＤを格納して、共用メモリからのデータに対して保留中の要求を識別し、順序付けされたメモリ応答コマンドと順序付けされていない戻りデータ・メッセージの間の相対的な順序を示すものであるトランザクション追跡テーブルと、を備え、
トランザクション追跡テーブルに格納されたコマンド及びコマンダＩＤから、トランザクション追跡テーブルに格納された保留中の要求と同じアドレスに対する後続の要求が、遅延されるべきか、無視されるべきかが決定される
ことを特徴とするコンピュータシステム。
【請求項２】
データに対する各保留中の要求に対して、前記トランザクション追跡テーブルに、順序付けされていない戻りデータ・メッセージが、関連する、順序付けされたメモリ応答コマンドを有することを示すための指示手段を更に備えている請求項１に記載のコンピュータシステム。
【請求項３】
指示手段は、トランザクション追跡テーブルへの少なくとも一つの順序付けされたチャンネル上で発生されたマーカー・メモリ応答コマンドを更に含む請求項２に記載のコンピュータシステム。
【請求項４】
トランザクション追跡テーブルは、複数のエントリを更に備え、
各エントリは、複数のノードの他の１つのメモリ位置のアドレスを記憶するためのものであり、
そして各エントリは、複数の状態ビットであって、その夫々が、関連する要求の対応する状態を指示するために選択的に設定され得る、複数の状態ビットを更に含む請求項３に記載のコンピュータシステム。
【請求項５】
トランザクション追跡テーブルは、順序付けされたチャンネル上のマーカー・コマンドがノードへ戻されたかどうかを指示する第1状態のビットセットを含む請求項４に記載のコンピュータシステム。
【請求項６】
トランザクション追跡テーブルは、
第２状態のビットセットであって、順序付けされていない戻りデータ・メッセージがノードへ戻されたかどうかを指示するために選択的に設定され得る、第２状態のビットセットと、
状態ビットセットの中の第1と第2ビットの両方を有するトランザクション追跡テーブルからエントリを取除く手段、
とを更に含む請求項５に記載のコンピュータシステム。
【請求項７】
トランザクション追跡テーブルに記憶されたアドレスへ発生されるメモリ要求コマンドであって、順序付けされたチャンネル上のマーカー・コマンドの受信を指示するためにそのアドレスに対応するトランザクション追跡テーブルの第１ビットがセットされる前に受け取られたメモリ要求コマンドを無視するための手段を更に備えた請求項６に記載のコンピュータシステム。
【請求項８】
無視されたメモリ要求コマンドは無効要求である請求項７に記載のコンピュータシステム。
【請求項９】
アドレスへ発生されたメモリ要求コマンドを無視する手段は、その要求を発生したプロセッサがそのアドレスをトランザクション追跡テーブルに入力させたプロセッサに対応する場合だけその要求を無視する請求項７に記載のコンピュータシステム。
【請求項１０】
トランザクション追跡テーブルに記憶されたアドレスへ発生されるメモリ要求コマンド

を、順序付けされたチャンネル上のメモリ応答コマンドが受け取られるまで遅延する手段を更に備え、そのメモリ要求コマンドは、そのアドレスに対応するトランザクション追跡テーブルの第１ビットがセットされる前に受け取られている請求項６に記載のコンピュータシステム。

【請求項１１】

アドレスに関連した所望のバージョンのデータがノードへ戻されるまでメモリ要求コマンドを更に遅延させる請求項１０に記載のコンピュータシステム。

【請求項１２】

アドレスをトランザクション追跡テーブルへ入れるようにした複数のプロセッサの一つへそのアドレスに関連したデータの所望のバージョンが戻されるまでメモリ要求コマンドを更に遅延させる請求項１０に記載のコンピュータシステム。

【請求項１３】

マルチプロセッサコンピュータシステムの共通のアドレスへ発生される複数のメモリ要求コマンド間の順序を維持する方法であって、前記のマルチプロセッサコンピュータシステムは、スイッチを経て接続された複数のマルチプロセッサノードを備え、マルチプロセッサノードの各々は、少なくとも２つのプロセッサと共用メモリの一部分とを含んでおり、

前記マルチプロセッサコンピュータシステムは、複数の通信チャネルを有し、該通信チャネルは、前記マルチプロセッサノードを前記スイッチを経て相互接続し、前記マルチプロセッサコンピュータシステムを通って前記プロセサの少なくとも１つによって発生されたメモリ要求コマンド、１つあるいはそれより多いメモリ応答コマンド、及び前記メモリ応答コマンドに対する応答として発生された戻りデータ・メッセージへの論理的に独立しているパスを与えるものであり、

前記複数の通信チャネルが、

第1の順序付けされた通信チャンネルであって、該第1の順序付けされた通信チャンネルを通じて、前記順序付けされたメモリ応答コマンドが発送される、第1の順序付けされた通信チャンネルと、

第２の順序付けされていない通信チャンネルであって、該第２の順序付けされていない通信チャンネルを通じて、戻りデータ・メッセージが発送される、第２の順序付けされていない通信チャンネルと、から成り、

前記マルチプロセッサコンピュータシステムは、前記共用メモリへのメモリ要求コマンドに応じて発生された１つあるいはそれより多いメモリ応答コマンドを順序付ける、各マルチプロセッサノードに配置されたシリアル化ポイントを有する、方法において、

リモートマルチプロセッサノードの共用メモリの一部分におけるそれぞれのアドレスに対してメモリ応答コマンドと順序付けられていない戻りデータ・メッセージの間の相対的な順序を識別するために、保留中の要求に関する共用メモリのアドレス、コマンド、及びコマンダＩＤを格納するための、各マルチプロセッサノードに配置されたトランザクション追跡テーブルにおいて、前記のマルチプロセッサノードの各々から前記のスイッチへ送られるメモリ要求コマンドのアドレスリストを維持し、

トランザクション追跡テーブルを通じて、要求の相対的な順序と関連するリモートなメモリ要求コマンドを追跡し、

トランザクション追跡テーブルの中で、メモリ要求コマンドの相対的な順序に対するリモートなメモリ要求コマンドのシリアル位置を識別する、

ことを特徴とする方法。

【請求項１４】

請求項１３に記載の方法であって、

メモリ要求コマンドのアドレスのリストは、少なくとも一つの順序付けされたチャンネルにおける参照情報の順序を識別するものであり、

他のチャンネル上のトランザクションに対して、少なくとも１つの順序付けされたチャンネルにおける参照情報の識別された順序を再編成するステップを更に含む方法。

【請求項１５】
共用メモリの部分へのリモートな参照情報毎に、順序付けされた参照情報に対応するメモリの共用部分に関連したマルチプロセッサノードのアドレスリストへ、その順序付けされた参照情報が対応するリモートな参照情報を有することを指示する段階を更に含む請求項１４に記載の方法。
【請求項１６】
指示段階が更に、少なくとも１つの順序付けされたチャンネル上でアドレスリストへコマンドを発生する段階を含む請求項１５に記載の方法。
【請求項１７】
アドレスリストは複数のエントリを備え、各エントリは、別のマルチプロセッサノードのメモリ位置をアドレスする参照情報のアドレスを記憶するためのものであり、そして関連する要求の状態を指示するための複数の状態ビットを含んでいる請求項１６に記載の方法。
【請求項１８】
状態ビットは、順序付けされたチャンネルのコマンドがマルチプロセッサノードに返送されたかどうかを示す第１ビットセットを更に含んでいる請求項１７に記載の方法。
【請求項１９】
状態ビットは、リモートな参照情報がマルチプロセッサノードに返送されたかどうかを指示するための第２ビットセットを更に含み、そして前記の方法は、状態ビットのセットの第１ビットと第２ビットの両方を有するトランザクション追跡テーブルからエントリを除去する段階を更に含んでいる請求項１８に記載の方法。
【請求項２０】
アドレスリストに記憶されたアドレスへ発生される要求であって、順序付けされたチャンネルにおけるコマンドの受信を指示するためにそのアドレスに対応するアドレスリストの第１ビットがセットされる前に受け取られた当該要求を無視する段階を更に含む請求項１９に記載の方法。
【請求項２１】
無視される要求は無効の要求である請求項２０に記載の方法。
【請求項２２】
アドレスへ発生された要求を無視する段階は、その要求を発生したプロセッサが、そのアドレスをアドレスリストに入力させたプロセッサに対応する場合だけ要求を無視する請求項２１に記載の方法。
【請求項２３】
アドレスリストに記憶されたアドレスへ発生される参照情報を、順序付けされたチャンネルのコマンドが受け取られるまで遅延する段階を更に含み、当該参照情報は、そのアドレスに対応するアドレスリストの第1ビットがセットされる前に受け取られている請求項２２に記載の方法。
【請求項２４】
参照情報は、アドレスに関連したデータの所望のバージョンがマルチプロセッサノードに返送されるまで更に遅延される請求項２３に記載の方法。
【請求項２５】
参照情報は、アドレスに関連したデータの所望のバージョンが、そのアドレスをアドレスリストに入力させた複数のプロセッサの１つに返送されるまで更に遅延される請求項２４に記載の方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、一般に、コンピュータアーキテクチャーの分野に係り、より詳細には、分散型共用メモリマルチプロセッサシステムに係る。
【０００２】

【従来の技術】
この分野で良く知られているように、対称型のマルチプロセッサコンピュータは、高性能のアプリケーション処理を行うことができる。通常の対称型マルチプロセッサコンピュータシステムは、バスによって互いに接続された多数のプロセッサを備えている。対称型マルチプロセッサシステムの１つの特徴は、メモリ空間が全てのプロセッサ間で共用されることである。１つ以上のオペレーティングシステムがメモリに記憶され、種々のプロセッサ間でのプロセッサ又はスレッドの分散を制御する。
異なるプロセッサ又はスレッドが多数の異なるプロセスを同時に実行できるようにすることにより、所与のアプリケーションの実行速度を著しく高めることができる。理論的に、システムの性能は、マルチプロセッサシステムにおけるプロセッサの台数を増加するだけで改善することができる。実際には、ある飽和点を越えてプロセッサを追加し続けると、単に通信ボトルネックが増えるだけとなり、従って、全システム性能を制限することになる。
【０００３】
例えば、図１Ａには、共通の相互接続バスを経て互いに接続された８個のプロセッサを含む典型的な公知のマルチプロセッサシステム２が示されている。動作中に、各プロセッサ３ａ－３ｈは、共用相互接続バス５を経て互いに他のプロセッサ及び共用メモリ４と通信する。図1Aの対称型マルチプロセッサ構成は、今日までに構築されたマルチプロセッサについて充分である。しかしながら、より高速のマイクロプロセッサの出現に伴い、通常の共用相互接続バスは、接続されたマイクロプロセッサの潜在的な全性能を充分に働かせることができない。プロセッサとメモリとの間の唯一の通信リンクは、共用バスであるから、バスはプロセッサからの要求で急速に飽和状態となり、各プロセッサがシステムバスへのアクセスを得るよう試みるときに遅延が増大する。それ故、プロセッサは、高い速度で動作することができるが、性能に関する制限ファクタは、システムバスの使用可能な帯域である。
【０００４】
通信帯域巾は、ＳＭＰシステムの性能において重要なファクタである。帯域巾は、ＳＭＰシステムにおけるノードの対又はサブセットの間で均一ではないから、業界では、ＳＭＰシステムの通信帯域巾を決定するために「二等分帯域巾」測定を使用している。二等分帯域巾は、次のように決定される。システムを等しい計算能力（等しいプロセッサ数）の２つの部分に区分化する全ての考えられる方法が確かめられている。各区分に対し、２つの区分間に維持し得る帯域巾が決定される。全ての維持し得る帯域巾の最小値は、相互接続の二等分帯域巾である。２つの区分間の最小帯域巾は、最悪の通信パターンが存在するときにマルチプロセッサシステムにより維持できる通信帯域巾を指示する。従って、大きな二等分帯域巾が望まれる。
【０００５】
公知技術では、バス飽和の問題を克服するために、多数の相互接続アーキテクチャー即ち「トポロジー」が使用されている。これらのトポロジーは、メッシュ、トーラス（円環体）、ハイパーキューブ（超立体）及び拡張ハイパーキューブを含む。
【０００６】
【発明が解決しようとする課題】
例えば、メッシュ相互接続は、図１Ｂにシステム７として示されている。メッシュネットワークの主な利点は、簡単で且つ配線が容易なことである。各ノードは、少数の他の隣接ノードに接続される。しかしながら、メッシュ相互接続は、３つの重大な欠点を有する。第１に、メッセージは、それらの行先に到達するために平均的に多数のノードを横断しなければならず、その結果、通信の待ち時間が長くなる。第２に、二等分帯域巾は、他のトポロジーに対するものであるから、メッシュトポロジーの場合に充分に計測しない。最後に、各メッセージはメッシュ内の異なる経路を進行するので、ＳＭＰシステム内には自然の順序付けポイントが存在せず、それ故、メッシュトポロジーの実施を必要とするキャッシュコヒレンスプロトコルがしばしば非常に複雑なものとなる。

【０００７】
トーラス、ハイパーキューブ及び拡張ハイパーキューブトポロジーは、全て、ノードが種々の複雑な構成、例えば円環体構成又は立体構成で相互接続されたトポロジーである。トーラス、ハイパーキューブ及び拡張ハイパーキューブの相互接続は、メッシュ相互接続よりも複雑であるが、その待ち時間及び帯域巾は、メッシュ相互接続よりも優れている。しかしながら、メッシュ相互接続と同様に、トーラス、ハイパーキューブ及び拡張ハイパーキューブトポロジーは、自然の順序付けポイントを与えず、従って、これらのシステムの各々に対して複雑なキャッシュコヒレンスプロトコルを実施しなければならない。
共用メモリのマルチプロセッサシステムでは、プロセッサは、通常、将来アクセスされる見込みが高いと決定されたデータを記憶するために専用キャッシュを使用している。プロセッサは、それらの専用キャッシュからデータを読み取りそしてメモリへ書き戻すことなく専用キャッシュにおいてデータを更新するので、各プロセッサの専用キャッシュが一貫して即ちコヒレントに保持されるよう確保するための機構が必要となる。ＳＭＰシステムのデータのコヒレンス性を確保するのに使用される機構は、キャッシュコヒレンスプロトコルと称される。
【０００８】
物理的な相互接続部のトポロジー、帯域巾及び待ち時間に加えて、キャッシュコヒレンスプロトコルの効率も、システム性能の重要なファクタである。キャッシュコヒレンスプロトコルは、待ち時間、ボトルネック、非効率性又は複雑さを多数の仕方で導入する。
ロード及び記憶動作の待ち時間は、設計のプロトコルによって直接影響されることがしばしばある。例えば、あるプロトコルでは、全ての無効化メッセージがそれらのターゲットプロセッサへ送られそして確認メッセージがその元のプロセッサへ完全に返送されるまで記憶動作が完了したとみなされない。従って、記憶の待ち時間は、無効化がその行先へ送られるのを元のプロセッサが待機しなくてよいプロトコルよりも相当に長いものとなる。更に、確認は、システム帯域巾の相当の部分を消費する。
【０００９】
ボトルネックは、コントローラの高い占有度によりしばしば生じる。「占有度」とは、コントローラが要求を受け取った後に使用できなくなる時間の長さを示す用語である。あるプロトコルでは、直接的なコントローラは、メモリ位置に対応する要求を受け取ると、その前のコマンドに対応するある確認がディレクトリに到着するまで同じメモリ位置への他の要求に対して使用できなくなる。コントローラは、平均より高いレートで競合する要求を受け取る場合に、ボトルネックとなる。
又、キャッシュコヒレンスプロトコルの設計は、ハードウェアの複雑さにも影響する。例えば、あるプロトコルは、停滞及び公正さの問題を招き、これらは、付加的な機構で対処される。その結果、ハードウェアの複雑さが増大する。
【００１０】
そこで、オペレーションの待ち時間を最小にし、広い通信帯域巾を与え、コントローラの占有度を低くし、そして多数のプロセッサへと拡張することのできる対称的なマルチプロセッサシステムを提供することが要望される。
【００１１】
【課題を解決するための手段】
本発明は、少なくとも１つのプロセッサ及び共用メモリの一部分を含む多数のマルチプロセッサノードがスイッチを経て互いに接続された対称的なマルチプロセッサシステムに効果的に使用される。マルチプロセッサノードの各々にはトランザクション追跡（トラッキング）テーブル（ＴＴＴ）が維持される。ＴＴＴは、ノードをスイッチに接続するノードのグローバルポートに存在してもよいし、或いはマルチプロセッサノードの少なくとも１つのプロセッサの各々に存在してもよい。
ＴＴＴは、マルチプロセッサノードから発生され及びそれにより受け取られる要求の順序を決定しそしてそれを強制するのに使用される。本発明の１つの特徴によれば、ＴＴＴは、マルチプロセッサノードへ返送される要求の順序を次のように決定するのに使用される

。各要求は、多数のトランザクションに細分化され、各トランザクションは、異なる仮想チャンネルを経て搬送される。少なくとも１つのチャンネルが順序付けされるが、他のチャンネルの返送データは、ばらばらの順序で受け取ることができる。コヒレンス性を維持するために、共通のアドレスへ発生されるトランザクションが順序正しく取り扱われるのが望ましい。本発明の１つの特徴によれば、マーカーパケットが、順序付けされたチャンネルを経てＴＴＴへ発生され、アドレスに関連したデータが依然処理されていることを指示する。このような構成では、ＴＴＴは、マーカーパケットに続いて受け取られた上記順序付けされたチャンネルの他の要求を無視するか、又はデータが受け取られるまで遅延するように確保する。

【００１２】

従って、本発明の１つの特徴によれば、各々少なくとも１つのプロセッサ及び共用メモリの一部分を含む複数の接続されたマルチプロセッサノードを有するコンピュータシステムは、上記複数のマルチプロセッサノードの各々における複数のプロセッサに関連した追跡機構であって、上記複数のマルチプロセッサノードの１つにおける少なくとも１つのプロセッサの１つにより発生された共用メモリのリモート部分のアドレスへの要求の位置を、上記複数の接続されたマルチプロセッサノードにおける少なくとも１つのプロセッサにより上記アドレスへ発生された複数の他の要求に対して識別するための追跡機構を備えている。

本発明の更に別の特徴によれば、マルチプロセッサコンピュータシステムの共通のアドレスへ発生される複数の要求間の順序を維持するための方法が提供される。マルチプロセッサコンピュータシステムは、スイッチを経て接続された複数のマルチプロセッサノードを備え、各マルチプロセッサノードは、少なくとも１つのプロセッサ及び共用メモリの一部分を含む。上記方法は、マルチプロセッサノードの各々からスイッチへ送られる要求のアドレスリストを維持して、リモートマルチプロセッサノードの共用メモリの一部分における各アドレスに対して要求の相対的な順序を識別し、アドレスに関連した要求が満足されるまでアドレスをリストに維持するという段階を含む。

【００１３】

【発明の実施の形態】

本発明の上記及び他の特徴は、添付図面を参照した以下の詳細な説明から明らかとなろう。

本発明の１つの実施形態によれば、ハイアラーキー式の対称的マルチプロセッサ（ＳＭＰ）システムは、高性能スイッチを経て互いに接続された多数のＳＭＰノードを備えている。従って、ＳＭＰノードの各々は、ＳＭＰシステムにおいてビルディングブロックとして働く。以下、１つのＳＭＰノードビルディングブロックの要素及び動作を最初に説明し、その後に、ＳＭＰシステムの動作を説明し、それに続いて、大規模のＳＭＰシステムにおいてメモリのコヒレンス性を維持するために使用されるキャッシュコヒレンスプロトコルを説明する。

【００１４】

ＳＭＰノードビルディングブロック

図２を参照すれば、マルチプロセッサノード１０は、４つのプロセッサモジュール１２ａ、１２ｂ、１２ｃ及び１２ｄを備えている。各プロセッサモジュールは、中央処理ユニット（ＣＰＵ）を備えている。好ましい実施形態では、デジタル・イクイップメント社で製造されたＡｌｐｈａ（登録商標）２１２６４プロセッサチップが使用されるが、以下に述べるコヒレンスプロトコルをサポートすることのできるものであれば、他の形式のプロセッサチップも使用できる。

マルチプロセッサノード１０は、多数のメモリモジュール１３ａ－１３ｄを含むメモリ１３を備えている。このメモリは、３２ギガバイトの記憶容量を備え、４つのメモリモジュールの各々が８ギガバイトを記憶する。各メモリモジュールは、多数のメモリブロックに分割され、各ブロックは、例えば、６４バイトのデータを含む。データは、一般に、メモリからブロックで検索される。

【００１５】
更に、マルチプロセッサノード１０は、接続されたＩ／Ｏバス１４ａを経て外部装置（図示せず）とマルチプロセッサノード１０との間で行われるデータ転送を制御するためのＩ／Ｏプロセッサ（ＩＯＰ）モジュール１４を備えている。本発明の１つの実施形態では、Ｉ／Ｏバスは、周辺コンピュータ相互接続（ＰＣＩ）プロトコルに基づいて動作する。ＩＯＰ１４は、ＩＯＰキャッシュ１４ｃ及びＩＯＰタグ記憶装置１４ｂを含む。ＩＯＰキャッシュ１４ｃは、ＰＣＩバス１４ａを経て外部装置へ転送されるメモリ１３からのデータのための一時的な記憶装置である。ＩＯＰタグ記憶装置１４ｂは、外部装置とプロセッサとメモリとの間に移動されるデータに対するコヒレンス情報を記憶するための６４エントリのタグ記憶装置である。

【００１６】
マルチプロセッサノードのメモリ１３に記憶されたデータのコヒレンス性は、デュープリケートタグ記憶装置（ＤＲＡＧ）２０によって維持される。ＤＴＡＧ２０は、全てのプロセッサ１２ａ－１２ｄにより共用され、そして４つのバンクに分割される。各バンクは、関連するプロセッサにより使用されるデータに対応する状態情報を専用に記憶する。
ＤＴＡＧ、メモリ及びＩＯＰは、ＡＲＢバス１７と称する論理バスに接続される。プロセッサにより発生されるメモリブロック要求は、ローカルスイッチ１５を経てＡＲＢバス１７にルート指定される。ＤＴＡＧ２０及びＩＯＰ１４は、プロセッサ及びＩＯＰのキャッシュにおけるブロックの状態をルックアップし、そしてメモリブロックに対しそれらの状態を原子的に更新する。ＡＲＢバス１７は、全てのメモリ参照に対してシリアル化ポイントとして働く。メモリ要求がＡＲＢバスに現れる順序は、プロセッサが要求の結果を認知する順序である。

【００１７】
プロセッサモジュール１２ａ－１２ｄ、メモリモジュール１３ａ－１３ｄ及びＩＯＰモジュール１４は、ローカルの９ポートスイッチ１５を経て互いに接続される。インターフェイスモジュール１２ａ－１２ｄ、１３ａ－１３ｄ及び１４の各々は、同数の両方向性クロック送信データリンク１６ａ－１６ｉによりローカルスイッチに接続される。１つの実施形態では、データリンクの各々は、１５０ＭＨｚのレートで動作するシステムクロックの各縁で６４ビットのデータ及び８ビットのエラー修正コード（ＥＣＣ）を送信する。従って、データリンク１６ａ－１６ｉの各々のデータ帯域巾は、２．４ギガバイト／ｓである。
ローカルスイッチ１５は、クオドスイッチアドレス制御チップ（ＱＳＡチップ）１８及びクオドスイッチデータスライスチップ（ＱＳＤチップ）１９を備えている。ＱＳＡチップ１８は、プロセッサモジュールＩＯＰとメモリとの間のアドレス経路を制御するためのアービター（ＱＳ ＡＲＢ）１１を備えている。更に、ＱＳＡチップ１８は、以下に述べるようにローカルスイッチ１５を通るデータの流れを制御するためにＱＳＤチップ１９を制御する。ＱＳＤチップ１９は、プロセッサモジュールと、メモリモジュールと、ＩＯＰとの間の全てのデータ経路に対するスイッチ相互接続を与える。図２には示されていないが、以下に述べるように、マルチプロセッサノード１０がグローバルポートを経て他のマルチプロセッサノードに接続された場合には、ＱＳＤ及びＯＳＡがグローバルポートに対するスイッチ相互接続部を付加的に形成する。各プロセッサは、メモリデバイス１３ａ－１３ｄ、他のプロセッサ１２ａ－１２ｄ、ＩＯＰ１４のような使用可能なリソースの１つからデータを要求することもできるし、或いは他のマルチプロセッサノードのリソースからグローバルポートを経てデータを要求することもできる。従って、ローカルスイッチ１５は、２．４ギガバイトの広いバス帯域巾を維持しながら、種々のリソースから同時入力を受け入れることができねばならない。

【００１８】
ローカルスイッチは、多数の同時トランザクションを取り扱うことができる。各トランザクションは、通常、多数のリソース（メモリバンクや、データ経路や、待ち行列のような）を使用するので、ローカルスイッチの制御機能は非常に複雑になる。例えば、あるトラ

ンザクションは、そのトランザクションの段階０でメモリバンクを使用でき、段階１でメモリバンクからプロセッサポートへのデータ経路を使用でき、そして段階２でプロセッサポートからプロセッサへのデータ経路を使用できることを必要とする。ローカルスイッチアービター（ＱＳＡ１８のＱＳＡ　ＡＲＢ１１）は、あるトランザクションが開始されると、各段階でトランザクションにより必要とされるリソースが必要に応じて使用できるように要求を裁定する。

【００１９】

より重要なことに、アービターは、特定の要求が、他の要求の進行中に長時間にわたり（潜在的に不定に）裁定に負けることのないよう確保することにより、全ての要求及びプロセッサがリソースに対して公平なアクセスを得るように保証する。例えば、３つのリソースＡ、Ｂ及びＣを要求するトランザクションＴについて考える。このトランザクションＴは、トランザクションの適当な段階に３つのリソース全部が使用できるよう保証されるまで裁定に勝てない。リソースが使用可能であることのみに基づいてアービターがその判断を行う場合には、トランザクションＴは、Ａ、Ｂ又はＣの１つを使用する（他のリソースＤ、Ｅ等と共に）他のトランザクションが裁定に勝ち続ける間は、長時間にわたって成功しないことが考えられる。

【００２０】

各々が多数のリソースを使用して完了するような非常に多数の同時要求を伴うスイッチにおいて公平な裁定を保証するのは、計算上複雑である上に、高速データ経路において遅延を増加し勝ちである。ここに示す装置においては、ＱＳＡＡＲＢ１１が、特定のトランザクションをスケジュールする前に、１つのリソース（メモリバンク）のみについて裁定を行う。プロセッサに通じる待ち行列である第２のリソースは、ＱＳＡ　ＡＲＢ１１により第１のリソースについて裁定を行うときに、それが使用可能であるかどうかについてチェックする必要がない。というのは、ＱＳＤのアーキテクチャーがそのデータ経路を保証しそして待ち行列に通じる待ち行列スロットが常に使用できるからである。リソースに対する公平な裁定は、ＱＳＡＡＲＢ１１に著しい複雑さを伴うことなく与えられる。

【００２１】

本発明の１つの実施形態によれば、ＱＳＤは、対応する行先に通じるバッファに対してアップフロント裁定を必要とせずに全てのリソース（プロセッサ、メモリ、ＩＯＰ及びグローバルポート）からの入力を同時に受け取ることができる。次いで、全てのデータリソースは、データ経路又はスイッチにおける待ち行列スロットへのアクセスを裁定する必要なく、スイッチへデータを独立して送信することができる。というのは、ＱＳＤは、全てのリソースからのデータを実質的に同時に受信することのできる多数の同時挿入バッファを備えているからである。同時挿入バッファの２つの実施形態を以下に説明する。

【００２２】

同時挿入バッファスイッチ

上記のように、マルチプロセッサノードにおけるプロセッサ１２ａ－１２ｄ、ＩＯＰ１４及びメモリデバイス１３ａ－１３ｄの各々は、マルチプロセッサノードのプロセッサ及びＩＯＰからの要求を取り扱うためのリソースとして働く。データは、各リソース要素と、要求を発する要素との間でパケットの形態で転送される。各パケットは、５１２ビットのデータと、６４ビットのＥＣＣとを含む。上記したように、各データリンクは、６４ビットのデータ及び８ビットのＥＣＣを１５０ＭＨｚクロックの各縁において搬送する。従って、ＱＳＤの外部には、パケット当たり８個のデータ転送サイクルがある。しかしながら、ＱＳＤの内部では、クロックの１つの縁においてのみデータが収集される。従って、ＱＳＤの内部のロジックの各クロックサイクルに対し、潜在的に１２８ビットのデータがデータリンクから受け取られる。各パケットは、５１２ビットのデータ及び６４ビットのＥＣＣを含むので、ＱＳＤの内部では、各パケットごとに４つのデータ転送サイクルがあり、各ＱＳＤクロックサイクルに、１２８ビットのデータ及び１６ビットのＥＣＣがプロセッサ、ＩＯＰ又はメモリデバイスからＱＳＤへ転送される。

【００２３】

図３を参照すれば、ＱＳＤ１９は、５つの同時挿入バッファ（ＳＩＢ）２５ａ－２５ｅを含むように詳細に示されている。各ＳＩＢは、要求側要素、即ちプロセッサ１２ａ－１２ｄ又はＩＯＰの１つに専用である。各ＳＩＢは、それに関連した要求側要素と、ノード内の他のリソース要素、即ちプロセッサ１２ａ－１２ｄ、メモリ１３ａ－１３ｄ、ＩＯＰ１４及び好ましくはグローバルポートとの間でパケットを転送するためのデータ経路を制御する。グローバルポートは、他のマルチプロセッサノードへの相互接続部として働き、以下に詳細に説明する。ＳＩＢは、スイッチへのアクセスに対し要求側要素間の裁定を必要とせずに、スイッチに接続されたいずれのリソースからでも要求側要素によりパケットを同時に受信できるようにする。

【００２４】

既に述べたように、ＱＳＡ　ＡＲＢ１１は、スイッチ１９への制御を与えるように接続される。ＱＳＡ　ＡＲＢ１１には、メインアービター２７が含まれる。このメインアービター２７は、リソース（ＩＯＰ、プロセッサ１２ａ－１２ｄ及びメモリ１３ａ－１３ｄ）とスイッチ１９との間のデータの移動を管理する。プロセッサ１２ａ－１２ｄ及びＩＯＰ１４の各々は、ライン２８ａ－２８ｅ上のリソースの１つへアクセスするための要求を発生し、これらは、メインアービター２７に送られる。次いで、メインアービターは、各リソースが要求を受信できるときにこれらの要求をその関連リソースへ送る。リソースが要求を受け取るときに、スイッチ１９の裁定は必要とされない。というのは、ＳＩＢの各々は、全ての入力からの入力を実質的に同時に即ち同じデータサイクル内に受け取ることができるからである。　又、ＱＳＡ　ＡＥＢ１１には、多数の個々のアービター２３ａ－２３ｄも含まれる。これらアービター２３ａ－２３ｄの各々は、プロセッサ１２ａ－１２ｄの関連する１つと、それに対応するＳＩＢ２５ｂ－２５ｅとの間のデータ路を管理するのに使用される。ＩＯＰ１４とＳＩＢ２５ａとの間のデータ路を管理するために、ＩＯＰ１４には同様のアービター（図示せず）が含まれる。各プロセッサは、その関連ＳＩＢからデータを受け取ることができるので、その関連アービターは、接続されたデータ路にデータを送信する。

【００２５】

従って、スイッチ１９内の同時挿入バッファの使用により、要求側要素とリソースとの間の裁定経路は、２つの別々の区分に分割される。即ち、接続されたりソースからデータを受け取るために要求を発しているプロセッサが使用できるかどうかに拘わりなくプロセッサからの要求に応答してメインアービター２７がリソースを裁定するところの第１裁定区分と、プロセッサに関連したアービターがスイッチからのデータを送信するためにプロセッサへのアクセスを裁定するところの第２裁定区分である。このような構成では、裁定が分離されるために、接続されたりソース各々への公平なアクセスが与えられるように保証することができる。

図４Ａを参照すれば、ＳＩＢ２５ａの１つの実施形態が詳細に示されており、これは、ライン３６ａを経て８個の接続されたマルチプレクサ３４ａ－３４ｈにＭＵＸ選択信号＜３１：０＞を与えるように接続された入力アービター３６を備え、ＭＵＸ選択信号の４つが８個のマルチプレクサの各々に送られて、各マルチプレクサの９個の入力の１つが選択される。ＳＩＢ２５ａ－２５ｄは全て同様の構造にされ、従って、その１つについてのみ詳細に説明する。上記したように、潜在的に１０個のリソースがＳＩＢに接続される。１０個のリソースの１つは、ＳＩＢから出力を受信する要求側デバイスであり、一方、他の９個のリソースは、ＳＩＢに入力を与える。それ故、マルチプレクサ３４ａ－３４ｈの各々は、ＳＩＢに接続された９個のリソースから入力を受け取る。接続されたプロセッサの３つからの入力は、ラインＰｘ、Ｐｙ及びＰｚを経て受け取られる。第４のプロセッサ（ＳＩＢがＩＯＰデバイスに関連するとき）又はＩＯＰデバイス（ＳＩＢが１つのプロセッサに関連するとき）からの別の入力はラインＰＷ／ＩＯＰを経て受け取られる。メモリバンク１３ａ－１３ｄからの入力は、各々、ラインｍｅｍ０、ｍｅｍ１、ｍｅｍ２及びｍｅｍ３を経て受け取られ、そしてグローバルポートからの入力は、グローバルポートラインを経て受け取られる。

【００２６】
マルチプレクサ３４ａ－３４ｈの各々からの各出力は、バッファ３２の８個のバンクの１つに接続される。各バンクは８個のエントリを有し、各エントリは、１２８ビットのデータ及び１６ビットのＥＣＣを記憶する。従って、ＳＩＢにより受信されるデータの各パケットは、バッファ３２の同じ行において４つの異なるバンクに書き込まれる。以下に述べるように、入力アービター３６は、データを記憶するのに使用できるバッファのバンクを指示するための状態ビットを維持する。従って、１つ以上のリソースから１２８ビットのパケットデータが受け取られる各サイクルごとに、入力アービター３６は、バンクの使用状態に基づいて関連バンク３２ａ－３２ｈへパケットデータのサイクルを送信するために、各マルチプレクサ３４ａ－３４ｈにおける考えられる９個のリソース入力の１つを選択する。又、入力アービターは、ライン３６ｂを経てマルチプレクサ３０へバイパスデータも与える。入力アービターの状態ビットが、全てのバンク３２ａ－３２ｈが空であることを指示するときには、９個のリソース入力の１つが入力アービター３６を経て関連する要求側要素へ直接バイパスされる。
【００２７】
バンク３２ａ－３２ｈの各々は、マルチプレクサ３０に接続される。マルチプレクサ３０は、出力アービター３８により制御される。ＳＩＢ２５ａに関連する要求側要素がＳＩＢからデータを受け取る準備ができ、そしてパケットの一部分がＳＩＢのエントリに書き込まれると、出力アービターは、バンク３２ａ－３２ｈから要求側要素に８個のエントリーの１つを供給する。或いは又、出力アービターは、いずれのバンクも転送保留データをもたずそして入力アービターからライン３６ｂを経てデータが得られる場合には、ライン３６ｂを経て要求側要素にバイパスデータを供給する。
動作中に、パケットデータの第１の１２８ビットがＳＩＢに受け取られたときに、８個のバンクの１つが、パケットデータの第１の１２８ビットを記憶するために選択される。本発明の１つの実施形態によれば、パケットデータの１２８ビットが受け取られる次の３サイクルの各々の間に、手前の書き込みを実行するのに使用したバンクに隣接するバンクが、パケットデータの次の１２８ビットを書き込むのに選択される。例えば、バンク３２ａが、ソースｍｅｍ０からパケットデータの第１サイクルを書き込むのに使用できるバンクとして選択された場合には、パケットデータの第２サイクルはバンク３２ｂに書き込まれ、第３サイクルはバンク３２ｃに書き込まれ、そして第４サイクルはバンク３２ｄに書き込まれる。従って、パケットデータのその後のサイクルを書き込むためにどのバンクを使用すべきかの選択は、入力アービターにより選択されたバンクでスタートしそして各次々のパケット書き込みに対して隣接バンクに続くようにして回転ベースで実行される。その結果、受け取られたパケットがバッファ３２の共通の行における４つのバンクにわたって分散される。
【００２８】
８個のバンクが設けられ、そして本発明の１つの実施形態では、いずれの要求側要素においても保留となり得るリソース読み取りの最大数は８であるから、各書き込みサイクルの間に各リソースに対して少なくとも１つのバンクを使用できることが確保される。それ故、所与の瞬間に、全部で８個の保留の読み取り応答がスイッチによって受け取られた場合に、バンク３２ａ－３２ｈの各々を使用して、第１のパケットデータ書き込みサイクルを受け入れることができ、バンクの選択は、次の３つの書き込みサイクルについて回転される。
本発明の１つの実施形態では、ＳＩＢの各バッファは、先入れ先出し（ＦＩＦＯ）プロトコルのもとで動作する。パケットの２つの部分が同時に受け取られるので、それらに対しスイッチへ「読み込まれる」順序が選択される。リソースに対して裁定を行う要求側要素のロジックは、ＳＩＢと通信せず、そしてリソースに対して裁定するための他の要求側要素とも通信しないので、標準的なルールに従ってデータの完全性を確保する。例えば、リソースに固定の優先順位番号が指定される場合には、「低い番号の入力リソースからのデータが、常に、高い番号の入力リソースからのデータの前にスイッチに書き込まれる」と

いうようなルールに従う。

【００２９】

上記のように、図４Ａに示すＳＩＢの実施形態では、８個のバンクの使用について説明した。というのは、要求側要素が所与の瞬間にもつことのできる保留メモリ要求の数が８に対応するからである。しかしながら、設計上の制約により、それより少数のバンクを設けることが必要な場合には、インターリーブ又は同様の技術を使用して多数のデータチャンクを共通のバンクの異なる位置に同時に書き込みできるように当業者によって容易に設計を変更することができよう。それ故、本発明は、図４Ａに示す特定の実施形態に限定されるものではない。

上記のように、動作中に、入力アービターは、リソースからデータを書き込むための適当なバンクを選択するためにバンクにおける入力の利用性に関する状態情報を維持する。ＳＩＢへの入力を制御するための入力アービター３６の実施形態が図４Ｂに示されている。上記では９個の入力リソースについて述べたが、図４Ｂには、明瞭化のために、２つのリソース入力のみの書き込みを制御するロジックが示されている。入力パケットデータがライン３５を経て受け取られるときに、「入力１」のような指示信号がラッチチェーン４０に送られ、このラッチチェーンは、４個のラッチ、フリップ－フロップ又は同様の状態装置を含む。ラッチチェーン４０は、カウンタ機構として使用される。この例の目的として、４つの次々のデータ転送サイクルにパケットデータが受け取られると仮定する。４つのデータ転送サイクルの間に、入力信号はラッチチェーンを経て伝播する。ラッチチェーンには、オアゲート４６が接続される。入力値がラッチチェーン４０を経て伝播するときに、オアゲート４６の出力がアサートされる。

【００３０】

オアゲート４６の出力は、シフトレジスタ４８へのシフト信号を与える。シフトレジスタは、ＳＩＢの各バンクについて１つづつ、８個のビット位置を含む。シフトレジスタ４８は、入力信号サンプルを最初に受信する際に、バンク選択ロジック４４からのビットベクトルがロードされる。バンク選択ロジック４４から受け取られたビットベクトルは、１ビットがセットされるだけであり、ベクトル内のビットの相対的な位置が、パケットデータの書き込みを開始すべきバンクを指示する。

従って、バンク選択ロジック４４は、パケットデータの第１サイクルの書き込み行先を制御する。バンク選択ロジック４４は、利用性ベクトル４２を入力として受け取り、利用性ベクトルにおけるビットの相対的な位置が、書き込みデータを受け取ることのできない関連バッファを指示する。

【００３１】

バンク選択ロジックがシフトレジスタ４８へビットを与えるときに、シフトレジスタ４８の値がデマルチプレクサ４９に送られる。又、デマルチプレクサ４９は、入力１ソースが接続されるところのマルチプレクサ３４ａ－３４ｈの入力の数値表示も入力として受け取る。例えば、デマルチプレクサ４９は、「１」のマルチプレクサ選択値を用いてマルチプレクサ３４ａを経て入力１リソースデータが送られることを指示する「１」入力値を受け取る。選択されたバンクを指示するシフトレジスタ内のビットの位置に基づいて、値「１」がＭＵＸ選択＜３１：０＞信号３６ａの適当な位置へ伝播される。各入力ソースに対する各デマルチプレクサは、全てのＭＵＸ選択信号を駆動し、それらの出力は、これらの信号がマルチプレクサ３４ａ－３４ｈを駆動する前にオアされる。

【００３２】

バンクエントリの書き込みの後に、シフトレジスタの内容がオアゲート５０によりオアされ、利用性バンクベクトル４２として記憶される。これは、次のサイクルの間に、どのバンクが到来する書き込みに対して使用できるかをバンク選択ロジック４４により決定するために使用される。

ライン４６ａのシフト信号がアサートされる各サイクルに、シフトレジスタ４８のビットが右へシフトされる。ビットが右へシフトするときには、ＭＵＸ選択信号＜３１：０＞の選択値も右へシフトされ、次の書き込み動作中に入力ソースを次の隣接バンクへ供給する

ようにさせる。
従って、ローカルＱＳＤスイッチ内のＳＩＢを使用することにより、多数の同時に受け取られた入力がそれらの行先である要求側要素へ到達するように確保できる簡単且つ効率的なスイッチング機構が設けられる。このような構成では、リソースへのアクセスに対してソースがいったん裁定されると、ソースにより実行されねばならない全ての裁定が完了する。ソースは、リソースが常にスイッチバッファ３２へのアクセスを得ることができるという事実に依存する。ソースアービターが互いに独立して動作してリソースを管理できることにより、最小限の複雑さで公平な裁定を確保する機構が設けられる。更に、ＳＩＢは、要求側要素の最大数の保留中読み取りに対してデータを記憶できるので、たとえ全てのリソースからデータが同時に受け取られても、バッファ３２に対するリソースを裁定する必要はなく、リソースロジックの全体的な複雑さが低減される。

【００３３】
図５には、図３に示すようにプロセッサ又はＩＯＰデバイス（キャッシュを含む任意の要求側デバイス）へ接続することのできる同時挿入バッファ（ＳＩＢ）６１の第２の実施形態が示されている。ＳＩＢ６１は、９個のマルチプレクサ６０ａ－６０ｉを含み、そのうちの８個は、８個のバッファ６２ａ－６２ｈの各々に接続される。第９マルチプレクサ６０ｉは、以下に述べるようにバイパス経路を与えるのに使用される。マルチプレクサ６０ａ－６０ｉの各々は、接続されたメモリデバイスｍｅｍ０－ｍｅｍ３からの４つの入力、グローバルポートからの１つの入力、接続されたプロセッサからラインＰｘ、Ｐｙ及びＰｚを経て送られる３つの入力、そしてＩＯＰ（ＳＩＢに関連したデバイスがプロセッサの場合）又は別のプロセッサ（ＳＩＢに関連したデバイスがＩＯＰの場合）からラインＰＷ／ＩＯＰを経て送られる１つの入力を含む９つの入力を受け取る。

【００３４】
バッファ６２ａ－６２ｈの各々は、４つの１２８ビットエントリを含む。従って、各入力バッファは、ＳＩＢにおいて次々のサイクル中に４つの１２８ビット部分で受け取られた１つの５１２ビット情報パケットを記憶する。各バッファには、４対１のマルチプレクサ６４ａ－６４ｈが各々接続される。これらのマルチプレクサ６４ａ－６４ｈは、関連バッファの４つの入力のうちの１つを選択して、マルチプレクサ６６を経てＳＩＢの出力へ供給するのに使用される。
図４Ａについて上述したように、本発明の１つの実施形態では、各要求側要素がいかなる所与の瞬間にも異なるリソースに対してせいぜい８個の保留中読み取り参照を有するだけであるから、８個のバッファが含まれる。従って、図５には８個のバッファが示されているが、本発明はこれに限定されるものではない。むしろ、選択されるバッファの数は、関連するプロセッサ又はＩＯＰデバイスのバッファ特性に依存する。

【００３５】
動作中に、接続されたりソースの各々から入力が受け取られるときに、入力アービター６７は、各マルチプレクサにおける入力ラインの１つを選択し、データのパケットを空きバッファへ供給する。所与のリソースからのパケット書き込みの時間中に同じバッファが選択され、パケットの全ての部分が単一のバッファに維持される。パケットの少なくとも１つの部分がバッファに書き込まれると、それがマルチプレクサ６６に送られ、関連する要求側要素の準備ができたときにその要求側要素へ供給される。或いは又、いずれのバッファにもパケットデータが存在しない場合には、マルチプレクサ６０ｉを経、マルチプレクサ６６を経てパケットデータを出力へ直接的に供給することによりバイパス経路を選択することができる。

【００３６】
８個のバッファが設けられるので、ＳＩＢデバイス６１は、接続されたりソースの各々から実質的に同時に（即ち、同じデータサイクルに）データを受け取ることができる。ＱＳＤにＳＩＢを使用することにより、前記の実施形態の場合のように、ＳＩＢへのアクセスに対し要求側要素の間に裁定は必要とされない。その結果、リソースがローカルスイッチを使用する準備ができたときにローカルスイッチの利用性が保証される。更に、本来的に

公平な裁定機構が設けられる。というのは、スイッチに対する裁定の結果としてリソースへの要求が他のリソースへの他の要求により阻止されないからである。従って、裁定の複雑さを最小限に抑えながら最大のバス帯域巾を維持することのできる公平で且つ比較的簡単な構造体が与えられる。

【００３７】

従って、同時挿入バッファを使用して広いバス帯域巾をサポートするローカルスイッチを実施することにより処理リソースを最適に使用するマルチプロセッサノード１０が提供される。更に、ＡＲＢバス１３において参照の順序がシリアル化されるので、マルチプロセッサ１０のメモリのコヒレンス性を容易に維持する中央順序付けポイントが設けられる。ローカルスイッチに接続されるプロセッサモジュールの数を増加することにより処理能力を高める可能性が存在するので、図２の４プロセッサ／ローカルスイッチ構成体は、待ち時間の短いそしてコストの安い高性能のシステムを提供する。

【００３８】

大型の対称的マルチプロセッサシステム

モノリシックマルチプロセッサノードに含むことのできるプロセッサの数は、２つのファクタにより制限される。第１に、ローカルスイッチを経て互いに接続できるプロセッサの数は、ローカルスイッチを構成するチップにおいて使用できるピンの数により制限される。第２に、単一のモノリシックスイッチによりサポートされるデータ帯域巾が制限される。従って、接続されるプロセッサの数をある点を越えて増加すると、何ら性能利得が得られないことになる。

本発明の１つの実施形態によれば、ハイアラーキースイッチを経て複数のマルチプロセッサノードを相互接続することにより大型の対称的なマルチプロセッサシステムを形成することができる。例えば、ハイアラーキースイッチを経て８個のマルチプロセッサノードを接続して、３２個のプロセッサモジュール、８個のＩＯＰデバイス及び２５６ギガバイトのメモリを含む対称的なマルチプロセッサ（ＳＭＰ）システムが形成される。説明上、ここでは、少なくとも２つのマルチプロセッサノードを含むＳＭＰを大型ＳＭＰと称する。以下に詳細に述べるように、ＳＭＰノードにローカルスイッチを用いて少数のプロセッサを接続し、そしてハイアラーキースイッチを用いて多数のノードを大型のＳＭＰへと接続することにより、拡張可能な高性能システムを実現することができる。

【００３９】

マルチプロセッサノードをハイアラーキースイッチ式ノードへと接続するために、マルチプロセッサは、グローバルなポートインターフェイスを含むように拡張される。例えば、図６には、変更されたマルチプロセッサノード１００が示されている。図２のマルチプロセッサノードと同様に、ローカルスイッチ１１０は、４つのプロセッサモジュール、４つのメモリモジュール及びＩＯＰモジュールを接続する。図２及び６の同様の要素は、同じ参照番号を有する。マルチプロセッサノード１００のローカルスイッチ１１０は、図２のポート１６ａ－１６ｉと同様に構成された９個のポート１１６ａ－１１６ｉを含む１０ポートスイッチである。付加的なポート１１６ｊは、グローバルリンク１３２を経てグローバルポート１２０へ至る全二重のクロック供給データリンクを形成する。

【００４０】

グローバルポートは、マルチプロセッサノードをハイアラーキースイッチに接続し、大型のＳＭＰを実現する。例えば、図７Ａを参照すれば、本発明の１つの実施形態において、８ｘ８のハイアラーキースイッチ１５５を経て互いに接続された８個のノード１００ａ－１００ｈを含む大型のＳＭＰシステム１５０が示されている。これらノード１００ａ－１００ｈの各々は、図６に示すノード１００と実質的に同一である。

ノード１００ａ－１００ｈの各々は、全二重クロック供給データリンク１７０ａ－１７０ｈの各々によりハイアラーキースイッチ１５５に接続される。１つの実施形態において、データリンク１７０ａ－１７０ｈは、１５０ＭＨｚのクロック速度で動作され、従って、スイッチ１５５との間でデータをやり取りするための２．４ギガバイト／秒のデータ帯域巾をサポートする。これは、最大３８．４ギガバイト／秒の生の相互接続データ帯域巾、

及び１９．２ギガバイト／秒の二等分データ帯域巾をスイッチに与える。

【００４１】

大型のＳＭＰシステムは、マルチプロセッサノード１００ａ－１００ｈの各々が全システムメモリのアドレス可能な部分を含むか又は物理的メモリの分割部分を含むような分散型共用メモリシステムである。本発明の１つの実施形態では、全システムメモリに2^{43}個の物理的アドレス位置が存在する。ＳＭＰマルチプロセッサシステム１００の１つの実施形態は、「大フォーマット」及び「小フォーマット」と称する２つのアドレスフォーマットをサポートする。大フォーマットは、各ノードのプロセッサが動作するところの４３ビットの物理的アドレスを、マルチプロセッサシステムに使用するための４３ビットの物理的アドレスに直接マップする。大フォーマットアドレスを使用すると、物理的メモリアドレスのビット＜３８：３６＞をノード識別番号として使用することができる。アドレスビット３８：３６は、メモリスペースアドレスのホームノードを直接デコードし、一方、アドレスビット３８：３６の逆数は、Ｉ／Ｏスペースアドレスのホームノードをデコードし、ここで「ホーム」とは、メモリスペース又はＩ／Ｏスペースに関連したメモリ及びＩ／Ｏデバイスが存在するところの物理的マルチプロセッサノードを指す。

【００４２】

小フォーマットのアドレスモードは、マルチプロセッサシステムに４つ以下のノードが存在することを仮定するものである。小フォーマットは、各ノードのプロセッサが３６ビットの物理的にアドレスされたシステムで動作できるようにする。小フォーマットにおいて、物理的アドレスのビット３４：３３は、データ又はＩ／Ｏデバイスのホームノード番号を識別する。

しかしながら、たとえＣＰＵが３６ビットの物理的アドレスを用いて動作しても、マルチプロセッサシステムは、データ位置を特定するのに４３ビットの物理的アドレスを一貫して使用し、物理的アドレスのビット３７：３６がデータ又はＩ／Ｏデバイスのホームノード番号を識別する。従って、ＣＰＵにより発生された小フォーマットアドレスと、データライン１３ａ－１３ｈを経てハイアラーキースイッチ１５５へ送信されるものとの間で何らの変換が実行される。

【００４３】

マルチプロセッサシステム１５０のここに示す構成は、３２個のプロセッサ間に広帯域巾のキャッシュコヒレントな共用メモリを与えることができる。本発明の１つの実施形態による大型ＳＭＰの別の実施形態が図７Ｂに示されており、ここでは、２つのマルチプロセッサノード１００ａ及び１００ｂがハイアラーキースイッチを使用せずに互いに接続される。むしろ、２つのマルチプロセッサノードは、それらのグローバルポート出力を互いに接続することにより直接接続される。

図７Ｂの２ノード実施形態が使用されるか、図７Ａのマルチノード実施形態が使用されるかに拘わりなく、大きなアドレススペース及び処理能力をもつマルチプロセッサシステムが得られる。

【００４４】

両実施形態において、システムメモリアドレススペース及びＩ／Ｏアドレススペースは、全てのノード１００ａ－１００ｈ間にセグメントで物理的に分配される。システムの各ノードは、メモリスペースの物理的アドレスの上位３ビットを使用してアクセスされるメインメモリの一部分を含む。従って、各メモリ又はＩ／Ｏアドレスは、１つのノードのみにおける１つの唯一のメモリ位置又はＩ／Ｏデバイスへとマップされる。従って、上位３つのアドレスビットは、メモリ又はＩ／Ｏアドレスがマップされるノードである「ホーム」ノードを識別するためのノード番号を与える。各マルチプロセッサノードは、それらのホームノード又は他のマルチプロセッサノードに記憶された共用メモリの部分をアクセスすることができる。ホームノードがプロセッサ自身のノードであるところの共用メモリブロックにプロセッサがアクセス（ロード又は記憶）するときには、参照は、「ローカル」メモリ参照と称される。ホームノードがプロセッサ自身のノード以外のノードであるようなブロックを参照する場合には、参照は、「リモート」又は「グローバル」メモリ参照と

称する。ローカルメモリアクセスの待ち時間は、リモートメモリアクセスの待ち時間と異なるので、ＳＭＰシステムは、非均一メモリアクセス（ＮＵＭＡ）アーキテクチャを有すると言える。更に、システムはコヒレントなキャッシュを備えているので、システムは、キャッシュコヒレントなＮＵＭＡアーキテクチャと呼ばれる。

【００４５】

ここに示すキャッシュコヒレントなＮＵＭＡアーキテクチャは、高い性能と低い複雑さに寄与する多数の特徴を含む。設計上の１つの特徴は、メッセージ間の順序の固執及び利用である。メッセージがある順序特性に基づいてシステムに流れるよう保証することにより、オペレーションの待ち時間を著しく短縮することができる。例えば、記憶オペレーションは、記憶が完了したとみなされる前に無効メッセージがそれらの最終的な行先プロセッサに供給されることを必要とせず、むしろ、無効メッセージが行先プロセッサへと通じるある順序付けされた待ち行列に入れられるや否や記憶が完了したとみなされる。

更に、ある順序が維持されるよう保証することにより、設計上、確認又は完了メッセージの必要性が排除される。メッセージは、それらがある待ち行列に入れられた順序でそれらの行先に到達するように保証される。従って、メッセージがその行先に到達したときに確認を返送する必要性が排除される。これは、システムの帯域巾を改善する。

【００４６】

更に、事象順序及びメッセージ順序は、「ホットポテト」オペレーションを行うのに使用される。ある待ち行列に順序を利用することにより、ディレクトリ又はＤＴＡＧコントローラのようなコントローラは、単一ビジットにおいて要求をリタイアすることができる。他の要求との競合により要求を否定的に確認しそして再トライする必要はない。「ホットポテト」オペレーションの結果として、公平さ及び欠乏の問題が解消される。

設計に使用される第２の特徴は、仮想チャンネルである。仮想チャンネルとは、メッセージを「チャンネル」へと分類する構成であって、チャンネルは物理的なリソースを共用する（従って、「仮想」である）が、各チャンネルは、他のものとは独立して流れ制御される。仮想チャンネルは、システムのメッセージ間で流れに依存しそしてリソースに依存するサイクルを排除することにより、キャッシュコヒレンスプロトコルにおける停滞を排除するのに使用される。これは、選択されたメッセージを否定的に確認しそしてそれに対応するコマンドを再トライすることにより停滞を検出しそして停滞状態を解消する機構を用いた公知のＮＵＭＡマルチプロセッサにおけるキャッシュコヒレンスプロトコルとは対照的である。

【００４７】

チャンネルの使用について以下に簡単に説明するが、詳細な説明は後で行う。上述したように、メッセージは、「チャンネル」と称する論理的なデータ路を用いて大型ＳＭＰ内をルート指定される。本発明の１つの実施形態には、以下のチャンネルが含まれる。即ち、要求側プロセッサから、トランザクションのアドレスに対応するホームノードのＡＲＢバスへトランザクションを搬送するためのＱ０チャンネルと、ホームＡＲＢバスから１つ以上のプロセッサ及びＩＯＰへトランザクションを搬送するためのＱ１チャンネルと、所有者プロセッサから要求側プロセッサへデータ記入トランザクションを搬送するためのＱ２チャンネルとである。変更されたデータを書き込むためにプロセッサからメモリへビクティム（Victim）トランザクションを搬送するためにＱ０Ｖｉｃチャンネルを設けることもできる。更に、Ｑ０Ｖｉｃチャンネルは、ビクティムトランザクションの背後に保持しなければならないＱ０トランザクションを搬送するのに使用できる。最後に、プロセッサからＩＯＰへＩＯスペーストランザクションを搬送するためにＱＩＯチャンネルが設けられる。

【００４８】

チャンネルは、以下に示すようなハイアラーキーを構成する。

（最低）ＱＩＯ＞Ｑ０Ｖｉｃ＞Ｑ０＞Ｑ１＞Ｑ２（最高）

以下に述べるように、停滞を回避するために、いずれのチャンネルのメッセージも、下位チャンネルのメッセージによって決して阻止されてはならない。順序付け特性及び仮想チ

ャンネルを形成しそして使用する機構の設計及び実施に関する詳細は、後で述べる。
従って、図７Ａ及び７Ｂに示すように、大型ＳＭＰは、図２のＳＭＰノードを任意の数だけ互いに接続することにより形成することができる。図７Ａ及び７Ｂに示すような大型ＳＭＰシステムのオペレーションは、以下に３つの部分について説明する。第１に、大型ＳＭＰに含まれるハードウェア要素について説明する。次いで、ＳＭＰのプロセッサ間にコヒレントなデータ共用を与えるキャッシュコヒレンスプロトコルについて説明する。更に、ハイアラーキースイッチの仮想チャンネルのために設けられたサポート機構を含む仮想チャンネルの実施及び使用について説明する。

【００４９】

大型ＳＭＰのハードウェア要素

マルチプロセッサノードの各々には、チャンネルを用いてコヒレントなデータ共用を実施するための多数の要素が設けられる。図６に戻ると、これらの要素は、ディレクトリ１４０と、ＤＴＡＧ２０と、ＩＯＰタグ１４ｂと、グローバルポート１２０と、ディレクトリ１４０とを備えている。更に、シリアル化ポイントのハイアラーキーは、キャッシュコヒレンスプロトコルを容易にするために参照の順序を維持できるようにする。これら要素の各々について、以下に詳細に述べる。

グローバルポート

グローバルポート１２０は、マルチプロセッサノード１００を、ハイアラーキースイッチリンク１７０を経て１つ以上の同様に構成されたマルチプロセッサノードに直接接続できるようにする。各ノード１００は対称的なマルチプロセッサシステムとして動作するので、システムにより多くのノードが追加されるにつれて、使用可能なアドレススペース及び処理能力が増加される。

【００５０】

図８は、グローバルポート１２０の拡張ブロック図である。グローバルポートは、トランザクション追跡テーブル（ＴＴＴ）１２２と、ビクティムキャッシュ１２４と、マルチプロセッサノードからハイアラーキースイッチへ送られるパケットを記憶するためのパケット待ち行列１２７、１２２、１２３及び１２５と、ハイアラーキースイッチから受け取られるパケットを記憶するためのパケット待ち行列１２１とを備えている。グローバルポート１２０は、ＡＲＢバス１３０と、ローカルスイッチの２つの専用ポート即ちＧＰリンク入力１３２ｂ及びＧＰリンク出力１３２ａとを経てノードの他のロジック（特にＱＳＡチップ）と通信する。
ＴＴＴは、マルチプロセッサノードにおいて保留中のトランザクション、即ちノードからグローバルポートを経て発生されて、他のマルチプロセッサノード又はハイアラーキースイッチからの応答を待機しているトランザクションを追跡する。グローバルポートにコマンドが送られるたびに、ＴＴＴにエントリが形成される。対応する応答がノードに受け取られたときに、ＴＴＴエントリがクリアされる。ＴＴＴは、２つの部分、即ちＱ０ＴＴＴ及びＱ１ＴＴＴで構成され、Ｑ０及びＱ１は、上記のようにＱ０及びＱ１チャンネルを進むパケットを指す。エントリーがＴＴＴにいかに割り当てられるか及びそれがいつリタイアされるかについては、以下に詳細に述べる。

【００５１】

又、グローバルポート１２０は、ビクティムキャッシュ１２４を含む。ビクティムキャッシュ１２４は、マルチプロセッサノードの各プロセッサから受け取られて別のマルチプロセッサノードのメモリに向けられるビクティム化データを記憶する。ビクティム化データとは、プロセッサのキャッシュ位置に記憶されてそのプロセッサにより変更されたデータである。変更データを記憶するキャッシュ位置に記憶する必要のある新たなデータがプロセッサに受け取られると、変更データは、ビクティム化されると言われ、ビクティムデータと称される。
ビクティムキャッシュ１２４は、プロセッサからリモートマルチプロセッサノードのメモリへ向けられたビクティムデータからのビクティムデータの一時的な記憶装置である。グローバルポートを経て別のノードへビクティムデータを送信するための機会があるときに

は、マルチプレクサ１６７は、ビクティムキャッシュ１２４からバス１７０の出力部分にデータを供給するように切り換えられる。グローバルポートにビクティムキャッシュを設けることにより、個々のプロセッサがグローバルシステムのメモリ書き込み待ち時間を待機せずに、プロセッサが各々のビクティムデータバッファを空にすることができる。むしろ、ビクティム書き込みは、使用できるデータサイクルがあるときに書き込みが実行されるようにグローバルポートにより制御される。ビクティムキャッシュからデータを解放する適切さに関連した幾つかの制御の問題があるが、これらは以下に説明する。

【００５２】

ＤＴＡＧ及びＩＯＰタグ

ＤＴＡＧ及びＩＯＰタグは、小型のＳＭＰシステムにも含まれるが、これについては以下に詳細に述べる。ＤＴＡＧ２０は、マルチプロセッサノードのプロセッサのキャッシュに記憶されたデータブロック各々に対する状態情報を記憶する。同様に、ＩＯタグ１４ａは、ＩＯＰに記憶された各データブロックに対する状態情報を記憶する。ディレクトリは、どのマルチプロセッサノードがデータのコピーを記憶するかを識別するおおよその情報を与えるが、ＤＴＡＧ及びＩＯタグは、マルチプロセッサノード内のどのプロセッサがデータのコピーを記憶するかに関する正確な指示を与えるのに使用される。それ故、ＤＴＡＧ及びＩＯタグは、参照情報がマルチプロセッサノードに到達したときに、そのノードのどのプロセッサがターゲットとなるべきかを決定するのに使用される。

【００５３】

図６に示すように、ＤＴＡＧ２０及びＩＯＰタグ１４ｂは、ＱＳＡチップ１８に接続されたメモリ領域を参照するアドレスを監視するためにＡＲＢバス１３０に接続される。ＤＴＡＧは、４つのプロセッサ１２ａ－１２ｄに対応する４つのセグメントに分割される。各プロセッサは、メモリ１３からのデータのサブセットを一時的に記憶するためのキャッシュ（図示せず）を備えている。各プロセッサのキャッシュに記憶されたメモリのブロックの上位アドレスビット（タグ）を記憶するためのタグ記憶装置が各キャッシュに関連される。ＤＴＡＧ２０の各セグメントは、関連プロセッサのキャッシュタグの状態を指示するデータを維持する。処理ユニットの外部のＤＴＡＧ２０にタグのコピーを記憶することにより、システムは、ＡＲＢバスを経て受け取ったコマンドをフィルタし、そしてプロセッサのキャッシュのデータに関連した調査（読み取り）及び無効化コマンドのみを各プロセッサに供給することができる。ＩＯＰタグ１４ａは、ＩＯＰキャッシュ１４ｃに記憶されたデータブロック各々の上位アドレスビットを記憶する。ＩＯＰタグ記憶装置は、プロセッサ１２ａ－１２ｄの各々に維持されたタグ記憶装置と同様である。

【００５４】

ＤＴＡＧ２０及びＩＯＰタグ１４ａの各エントリは、多数の状態ビットを含む。ＤＴＡＧ状態ビットは、次の４つの状態、即ちＩｎｖａｌｉｄ（無効）、Ｃｌｅａｎ（クリーン）、Ｄｉｒｔｙ＿Ｎｏｔ＿Ｐｒｏｂｅｄ、及びＤｉｒｔｙ＿Ｐｒｏｂｅｄのうちの１つを指示する。ＩＯＰタグのエントリの状態ビットは、次の２つの状態、即ちＶａｌｉｄ（有効）及びＤｉｒｔｙ（ダーティ）のうちの１つを指示する。「有効」ビットは、関連キャッシュの対応エントリに記憶されたデータが、メモリに記憶されたデータと一致することを指示する。「ダーティ」ビットは、関連キャッシュの対応エントリに記憶されたデータが関連プロセッサによって変更されそしてメモリに記憶されたデータに一致しないことを指示する。

【００５５】

ＤＴＡＧ２０及びＩＯＰタグ１４ｂは、マイクロプロセッサノード１００のＡＲＢバスにコマンドが現れるたびにアクセスされる。「無効」の状態がプロセッサ１のＤＴＡＧアクセスに応答して返送される場合には、ノードのプロセッサ１は、メモリアドレスに関連したデータの有効コピーを記憶しない。「有効」の状態がＩＯＰタグ１４ａへのアクセスから返送される場合には、ＩＯＰキャッシュ１４ｃがデータの有効コピーを記憶する。「クリーン」状態がプロセッサ１に対するＤＴＡＧアクセスに応答して返送される場合には、これは、プロセッサ１がメモリアドレスに対応するデータの無変更コピーを有するが、そ

のデータを読み取るための他のプロセッサによる試みがなされていないことを指示する。Ｄｉｒｔｙ＿Ｎｏｔ＿Ｐｒｏｂｅｄの状態がＤＴＡＧに応答して返送される場合には、これは、プロセッサ１がメモリアドレスに対応するデータの変更コピーを有し、そしてプロセッサが最後にデータを変更して以来、少なくとも１つのプロセッサがデータを読み取る試みをしていることを指示する。

【００５６】

ディレクトリオペレーション

一般に、ディレクトリは、関連マルチプロセッサノード（ホームノード）におけるメモリの各ブロックの所有権情報を与えるのに使用され、メモリのブロックは、一般に、メモリとＳＭＰシステムのプロセッサとの間に転送される最小量のデータである。例えば、本発明の１つの実施形態において、ブロックは、パケットのサイズと同様であり、即ち５１２ビット（６４バイト）のデータである。更に、ディレクトリは、どのマルチプロセッサノードがメモリデータのブロックのコピーを記憶するかを指示する。従って、読み取り型のコマンドの場合に、ディレクトリは、データの最新バージョンの位置を識別する。ビクティム型のコマンドの場合には、データの変更ブロックがメモリに書き戻される場合に、ディレクトリは、データの変更ブロックが現在のものであってメモリに書き込まねばならないかどうか決定するために検討される。それ故、ディレクトリは、参照情報がリモートマルチプロセッサノードのプロセッサにより発生されたものであるかローカルマルチプロセッサノードのプロセッサにより発生されたものであるかに拘わりなく、関連するマルチプロセッサノードのメモリブロックへの参照に対する第１アクセスポイントである。

【００５７】

ディレクトリは、対応するノード１００においてメモリ１３の各６４バイトのデータブロック（以下、キャッシュラインとも称する）に対して１つの１４ビットエントリを記憶する。メモリ１３と同様に、ディレクトリは、メモリアドレスがノードＮに存在する場合に、対応するディレクトリエントリもノードＮに存在するように、システムのノードにわたって物理的に分配される。

図９を参照すれば、ディレクトリエントリ１４０ａの１つの実施形態は、所有者ＩＤフィールド１４２及びノード存在フィールド１４４を含むように示されている。所有者ＩＤフィールドは、各６４バイトブロックに対する６ビットの所有者情報を含む。所有者ＩＤは、ブロックの現在所有者を特定し、現在所有者は、システムにおける３２個のプロセッサの１つ、又はシステムにおける８個のＩ／Ｏプロセッサの１つ、又はメモリのいずれかである。８ビットのノード存在情報は、システムの８個のノードのどれがキャッシュラインの現在バージョンを獲得したか指示する。ノード存在ビットは、同じノードにおける４つのプロセッサの累積状態を１ビットで表わすおおよそのベクトルである。共用データの場合には、２つ以上のノードが、情報を記憶する少なくとも１つのプロセッサを有する場合に、２つ以上のノード存在ビットがセットされる。

【００５８】

時々、状態情報のある断片がＤＴＡＧ又はディレクトリから得られる。このような場合、ＤＴＡＧからの状態情報を使用するのが好ましい。というのは、これは非常に高速で検索されるからである。例えば、メモリアドレスの所有者プロセッサがそのアドレスに対しホームノードに配置される場合には、所有者ＩＤを供給するのにＤＴＡＧが使用される。性能上の理由でＤＴＡＧによりサービスされない情報又は参照については、ディレクトリ１４０は、全てのコヒレンス性判断の焦点であり、従って、多数の機能を実行する。ディレクトリは、メモリデータブロックの所有者を識別する。所有者は、プロセッサ又はメモリのいずれかである。ディレクトリからの所有者情報は、データブロックの最新バージョンのソースを決定するために読み取り型コマンド（例えば、読み取り、読み取り－変更）により使用される。又、所有者情報は、以下に詳細に述べるようにビクティム化データをメモリに書き戻さねばならないかどうか決定するのにも使用される。

【００５９】

全ての読み取り型コマンドに対して、データの所有者を識別するのに加えて、ディレクト

リは、プロセッサからの「クリーン－ダーティ(Clean-to-Dirty)」及び「シェアド－ダーティ(Shared-to-Dirty）」コマンドを分析するのにも使用される。「クリーン－ダーティ」コマンドは、プロセッサがそのキャッシュにおいて現在「クリーン」状態にあるキャッシュラインを変更するよう希望するときにプロセッサにより発生される。「シェアド－ダーティ」コマンドは、「ダーティ－シェアド」状態にあるキャッシュラインを変更するよう希望するときに発生される。これらのコマンドは、ホームＡＲＢバスに送られ、そこで、ディレクトリは、プロセッサがキャッシュラインの最新バージョンを有するかどうか決定する。もしそうであれば、コマンドは成功となり、プロセッサは、キャッシュラインを変更することが許される。さもなくば、コマンドは失敗となり、プロセッサは、最初に、キャッシュラインの最新バージョンを獲得しなければならない。これらの記憶型オペレーションは、ディレクトリのノード存在情報を使用して、成功又は失敗を決定する。

【００６０】

上記のように、ディレクトリの存在ビットは、記憶型コマンドが発生されたときに各データブロックのコピーでマルチプロセッサノードを識別する。記憶コマンドは、キャッシュラインの内容が更新されようとしていることを指示する。関連するディレクトリエントリの存在ビット１４４を検討することにより、記憶コマンドがディレクトリ１４０に受け取られたときに、存在ビットを有するノードを用いて、これらのマルチプロセッサノードをそのノードにおけるキャッシュラインのコピーで識別し、従って、各ノードにおけるキャッシュラインを無効化できるようにする。

従って、ディレクトリ及びＤＴＡＧは、ローカルマルチプロセッサのメモリにおける各データブロック及びローカルプロセッサのキャッシュに記憶された各データブロックに対する状態情報を与えるように協働する。ホームノードのディレクトリは、キャッシュブロックのコピーの状態に関するおおよその情報を供給する。次いで、無効化コマンドがディレクトリにより識別されたノードへと進み、そこで、ＤＴＡＧがアクセスされて、コピー情報を更に改善する。従って、これらノードにおけるＤＴＡＧは、各ノードのどのプロセッサがそれらのキャッシュにラインのコピーを記憶するか指示する。

【００６１】

ＴＴＴ：

ＴＴＴは、マルチプロセッサノードからの保留中のトランザクション、即ち別のマルチプロセッサノード又はハイアラーキースイッチからの応答を待機している参照を追跡するのに使用される。保留中トランザクションに関する情報は、関連メモリアドレスへのその後のコマンドを処理する際にキャッシュコヒレンスプロトコルにより使用される。

図１０を参照すれば、ＴＴＴ１２２の１つの実施形態は、アドレスフィールド１５２と、コマンドフィールド１５４と、コマンダＩＤフィールド１５６と、ビット１５８ａ－１５８ｃを含む多数の状態ビット１５８とを含むように示されている。アドレスフィールド１５２は、現在進行中であるトランザクションに対するキャッシュラインのアドレスを記憶し、一方、コマンドフィールドは、現在進行中であるトランザクションに対するキャッシュラインに関連したコマンドを記憶する。コマンダＩＤフィールド１５６は、コマンドフィールドに記憶されたコマンドを開始したプロセッサのプロセッサ番号を記憶する。状態ビット１５８は、コマンドが進行中であるときにコマンドの状態を表わす。或いは又、状態ビット１５８は、進行中であるコマンドの種々の特性をあらわすように使用されてもよい。

【００６２】

例えば、「記入」状態ビット１５８ａは、読み取り型コマンドに応答して「記入」データ応答が受け取られたときに更新される。「シャドー」状態ビット１５８ｂは、グローバルポートを経て発生されたコマンドが「シャドー」型コマンド（以下に詳細に述べる）である場合にセットされる。ＡＣＫ状態ビット１５８ｃは、確認型応答を期待しているメッセージが応答を受信した場合にセットされる。応答が到着した場合に、このビットはクリアされる。ＴＴＴに含むことのできる全ての状態ビットが示されているのではないことに注意されたい。むしろ、以下の説明に関連のある状態ビットが含まれている。更に、メモリ

のコヒレンス性を維持するために必要と考えられれば、他の状態ビットを設けてもよく、従って、本発明は、ＴＴＴにおける特定のビット指定に限定されるものではないことが明らかであろう。

【００６３】

従って、ディレクトリ、ＤＴＡＧ、ＩＯＰタグ及びＴＴＴの各々は、ＳＭＰシステムにおけるキャッシュラインのコヒレンス性（以下、キャッシュコヒレンス性と称する）を維持するのに使用される。これら要素の各々は、ハイアラーキースイッチ１５５に接続されたマルチプロセッサノード間にコヒレント通信を与えるためにグローバルポートとインターフェイスする。

【００６４】

シリアル化ポイント：

上記要素に加えて、各マルチプロセッサノードにシリアル化ポイントを設けることによりデータ共用コヒレンス性が維持される。本発明の１つの実施形態において、各マルチプロセッサノードにおけるシリアル化ポイントは、ＡＲＢバス１３０である。全てのＱ０参照は、ローカルプロセッサにより発生されたものであるかリモートプロセッサにより発生されたものであるかに拘わりなく、ＱＳＡによりＡＲＢバス１３０を経てディレクトリ１４０及びＤＴＡＧ２０へ供給される。参照がディレクトリ及び／又はＤＴＡＧをアクセスすると、それにより得られるＱ１チャンネルコマンドが厳密な順序でＡＲＢバスに出力され、ここで、順序は参照のシリアル化順序である。マルチプロセッサノードの各々にシリアル化ポイントを設けることにより、ＳＭＰにおいて実施されるデータ共用コヒレンスプロトコルが相当に簡単化される。

【００６５】

マルチプロセッサノードの各々にシリアル化ポイントを設けるのに加えて、ハイアラーキースイッチ１５５は、ＳＭＰシステムに第２のシリアル化ポイントを与える。以下に詳細に述べるように、ハイアラーキースイッチは、第１のシリアル化ポイントに導入されたコヒレンス性が大型のＳＭＰシステムに維持されるよう確保するある順序付けルールに適合する。　グローバルポート／ハイアラーキースイッチインターフェイス：

図１１は、８個の入力ポート１５５ｉ０－１５５ｉ７及び８個の出力ポート１５５ｏ０－１５５ｏ７を含むハイアラーキースイッチ１５５のブロック図である。ハイアラーキースイッチ１５５の入力ポート１５５ｉ０－１５５ｉ７は、接続されたマルチプロセッサノード各々のグローバルポートからパケットを受け取る。ハイアラーキースイッチの出力ポート１５５ｏ０－１５５ｏ７は、接続されたマルチプロセッサノード各々のグローバルポートへパケットを供給する。

【００６６】

本発明の１つの実施形態において、受信したパケットをバッファするためのバッファ１６０ａ－１６０ｈが各入力ポートに関連される。図１１の実施形態は、各入力に１つのバッファを示しているが、いかなる数の入力ポート間にバッファが共用されてもよい。各パケットは、５つのチャンネルのいずれか１つと関連される。本発明の１つの実施形態では、以下に述べるように、各入力バッファ１６０ａ－１６０ｈの部分が、あるチャンネルのパケットを専用に記憶するようにされる。従って、グローバルポートからハイアラーキースイッチ１５５への流れ制御は、チャンネルベースで実行される。チャンネルベースでスイッチへのデータの流れを制御しそして入力バッファの部分を選択されたチャンネルに専用とすることにより、スイッチは、ＳＭＰシステムにおけるマルチプロセッサノード間で停滞のない通信を行う。

【００６７】

停滞のない通信を与えるのに加えて、ハイアラーキースイッチ１５５は、更に、メモリのコヒレンス性を確保するためにＳＭＰシステムの順序付け制約をサポートするように設計される。順序付け制約は、スイッチ１５５から関連マルチプロセッサノードのグローバルポートへ送出されるパケットの順序を制御することにより課せられる。いずれかの入力バッファ１６０ａ－１６０ｈからのパケットは、マルチプレクサ１８２ａ－１８２ｈを経て

いずれかの出力ポートへ送られる。更に、以下に述べるように、スイッチ１５５は、パケットをマルチキャスティングすることができる。従って、１つの入力バッファからのパケットは、いかなる数の出力ポートに送ることもできる。グローバル出力ポートに順序を強制することにより、マルチプロセッサノード各々に得られるシリアル化順序を維持して、完全にコヒレントなデータ共用機構をＳＭＰシステムに形成することができる。

【００６８】

ハイアラーキースイッチにおける停滞の回避

上述したように、図７Ａの８個のノードの各々は、ハイアラーキースイッチにデータを供給し、全てのノードがデータを同時に供給することもある。パケットは、異なる仮想チャンネルに供給される多数の異なるチャンネル形式（Ｑ０、Ｑ０Ｖｉｃ、Ｑ１、Ｑ２及びＱＩＯ）に分割され、ここで、仮想チャンネルとは、本質的に、他のチャンネルとの共通の相互接続部を共用するがその相互接続部のいずれかの端において独立してバッファされる特定形式のパケットに専用のデータ経路である。各ノードのグローバルポートとハイアラーキースイッチとの間には１つのデータ経路しかないので、異なる仮想チャンネルからの全てのパケットは、１つのデータ経路を使用してハイアラーキースイッチに書き込まれる。

【００６９】

８個のノード１００ａ－１００ｈの各々は、ハイアラーキースイッチへデータを送信することができるので、全てのメッセージがスイッチにより受信されて、スイッチから適当な順序で供給されるよう適切に確保するために、ある形式の制御が必要となる。更に、本発明の１つの目的は、対称的なマルチプロセッサシステムに停滞（デッドロック）が生じないよう保証するために上位順序のパケット形式が下位順序のパケット形式により阻止されないよう確保することである。本発明の１つの実施形態では、最高順序から最低順序までのパケットの順序は、Ｑ２、Ｑ１、Ｑ０、Ｑ０Ｖｉｃ及びＱＩＯである。

本発明の１つの特徴によれば、スイッチの入力ポートに到着するパケットの流れ制御を行うための機構であって、上記の停滞回避ルールが常に満足されるよう確保する機構が提供される。更に、スイッチにおいて使用できるバッファは最適に利用されねばならず、そして最大の帯域巾が維持されねばならない。

【００７０】

本発明の１つの実施形態によれば、ハイアラーキースイッチへのデータの書き込みを制御するための制御装置は、パケットの各形式に対し、ハイアラーキースイッチのバッファに専用スロットを設けることにより実施される。又、バッファは、任意の形式のパケットを記憶するのに使用できる多数の一般的なスロットも含んでいる。ハイアラーキースイッチに専用のバッファスロットを設けることにより、上位順序のパケット形式が常にスイッチを通る経路を使用できるよう保証することによって停滞を回避することができる。更に、使用できる一般的スロット及び専用スロットの数を監視し、そしてバッファに記憶されるパケットの異なる形式の数を監視することにより、ハイアラーキースイッチのバッファが容量に達したときにノードがバッファに書き込みするのを防止するような簡単な流れ制御機構を実施することができる。

【００７１】

図１２Ａには、多数のソースノードによる共通の行先バッファへの書き込みを制御するのに使用するための制御ロジックの一例が示されている。図１２Ａのブロック図には、２つの異なるノードのグローバルポート１２０ａ及び１２０ｂが一例として示されている。

図１２Ａにおいて、ノード１００ａ及び１００ｂのグローバルポート各々１２０ａ及び１２０ｂの部分は、ハイアラーキースイッチ１５５へ転送するためにＱ０／Ｑ０Ｖｉｃ、Ｑ１、Ｑ２及び一般形式のパケット（Ｑ０、Ｑ０Ｖｉｃ、Ｑ１、Ｑ２又はＱＩＯパケットのいずれか）を各々記憶するためのエントリ１３５ａ－１３５ｂを含むバッファ１３５を備えて詳細に示されている。バッファ１３５にはマルチプレクサ１６７ａが接続され、ＧＰアービター１３４からの選択信号を使用してリンクを経てハイアラーキースイッチへ送るためにパケット形式の１つを選択する。

【００７２】
更に、各グローバルポートは、専用のカウントレジスタ１３６を備えている。この専用のカウントレジスタは、パケットの各Ｑ０／Ｑ０Ｖｉｃ、Ｑ１及びＱ２チャンネル形式に対して、ハイアラーキースイッチ１５５において現在保留となっているチャンネル形式のパケットの数のカウントを記憶する。このカウントは、各チャンネル形式のパケットがハイアラーキースイッチへ転送されるときに増加され、そしてパケットがハイアラーキースイッチから転送されるときに減少される。
本発明の１つの実施形態において、ハイアラーキースイッチ１５５は、８個の入力ソースの各々に１つのバッファを備えている。図１２Ａには、２つのグローバルポート１２０ａ及び１２０ｂに対応する２つのバッファ１６０ａ及び１６０ｂのみが示されている。本発明の１つの実施形態では、バッファ１６０ａ及び１６０ｂの各々に少なくとも（ｍ－１）ｘｎ個の専用スロットがあり、但し、ｍは、バッファに専用エントリを有する仮想チャンネル形式の数に対応し、そしてｎは、バッファを共用するノードの数に対応する。図１２Ａの実施形態において、各バッファは、８個のエントリを有する。エントリのうちの５つは、一般的エントリであり、グローバルポート１３５から送られたパケットの形式を記憶することができる。残りの３つのエントリの各々は、特定形式のパケットを専用に記憶し、即ち１つのエントリは、Ｑ０／Ｑ０Ｖｉｃパケットを専用に記憶し、１つのエントリは、Ｑ１形式パケットを専用に記憶し、そして１つのエントリは、Ｑ２形式パケットを専用に記憶する。
【００７３】
専用エントリがバッファ１６０ａ及び１６０ｂの固定位置に存在するものとして示されているが、実際には、バッファのいずれの位置も専用のバッファ位置であり、即ちエントリの位置に拘わりなく、パケットの各特定形式ごとにバッファには常に１つの専用エントリがある。
ハイアラーキースイッチは、更に、各バッファ１６０ａ及び１６０ｂに対し、専用カウンタ１６２ａ及び１６２ｂと、フラグレジスタ１６３ａ及び１６３ｂとを含む。図１２Ａの実施形態において、専用カウンタ１６２ａは、４つのエントリを有し、その３つは、バッファ１６０ａに現在記憶されているＱ０／Ｑ０Ｖｉｃ、Ｑ１及びＱ２パケットの数を記憶するためのもので、そして１つは、バッファに使用される一般的エントリの数のカウントを記憶するためのものである。フラグレジスタは、３つのビットを含み、各ビットは、パケットのＱ０／Ｑ０Ｖｉｃ、Ｑ１及びＱ２形式の１つに対応し、そして関連する専用カウントがゼロであるかどうか（即ち、その形式のパケットの専用エントリが使用されたかどうか）を指示する。従って、フラグレジスタの値は、その形式の少なくとも１つのパケットがバッファに記憶されたことを指示する１であるか、又はその形式のパケットがバッファに記憶されないことを指示する０である。
【００７４】
更に、ハイアラーキースイッチ１５５は、各バッファ１６０ａ及び１６０ｂに対し、トランシットカウント１６４ａ及び１６４ｂを各々含む。トランシットカウントは、各ソースに対して、所与のデータサイクル中にトランシット状態であるいずれかの形式の保留中パケットの数を維持する。
所与のデータサイクル中にトランシット状態にあるパケットの数は、ハイアラーキースイッチとグローバルポートとの間の流れ制御待ち時間に直接関係している。流れ制御信号は、ハイアラーキースイッチからグローバルポートへ送られて、ハイアラーキースイッチへのデータの送信を停止するようにグローバルポートに通知する。流れ制御待ち時間（Ｌ）は、ハイアラーキースイッチによる流れ制御信号のアサートと、グローバルポートによるデータ送信の停止との間に生じるデータ転送サイクルの数として測定される。
【００７５】
又、ハイアラーキースイッチは、各バッファ１６８ａ及び１６８ｂの書き込みを制御するための書き込み制御ロジック１６６ａ及び１６６ｂも備えている。この書き込み制御ロジックは、ライン１６８ａに「流れ制御」信号をそしてライン１６８ｂに「確認（ＡＣＫ）

」信号＜３：０＞をアサートすることにより関連バッファへのデータの流れを制御する。「流れ制御」及びＡＣＫ信号は、各データ転送サイクルに送信される。上記のように、「流れ制御」信号は、接続されたグローバルポートによるパケットデータの送信を停止するのに使用される。ライン１６８ｂのＡＣＫ信号＜３：０＞は、パケットの専用形式の各々に対して１ビットを含み、そして接続されたグローバルポートに、その形式のパケットが関連バッファから解放されたことを通知するのに使用される。従って、ＡＣＫ信号は、グローバルカウントにより、専用カウンタ１３６の値を増加するのに使用される。

【００７６】

書き込み制御ロジックは、バッファの使用可能な全ての一般的エントリが、ハイアラーキースイッチへのトランシット状態にある考えられる全てのパケットを受け入れるのに充分でないと決定されたときに、流れ制御をアサートする。使用可能な一般的スロットの数は、次の式Ｉにより決定することができる。

式Ｉ：

Generic count＝（バッファサイズ）－（バッファに使用される一般的エントリの数）－（非アサートフラグの数）

使用可能な一般的エントリの数が決定されると、式ＩＩが真である場合に、流れ制御信号がアサートされる。

式ＩＩ：

Generic Count＝（トランシットカウント）＊（バッファを使用するノードの数）

従って、書き込み制御ロジック１６６は、使用中の一般的及び専用のスロットの数、トランシットカウント及び全バッファサイズを監視し、「流れ制御」信号をいつアサートすべきかを決定する。

【００７７】

「流れ制御」信号をアサートしても、ソースノードのグローバルポートによる全ての送信は停止されない。グローバルポートは、専用パケット形式に対応する専用スロットがハイアラーキースイッチのバッファに使用できる場合に、専用パケットデータをハイアラーキースイッチに常に転送する。従って、専用カウンタにおけるいずれかの専用カウントの値がゼロに等しい場合には、グローバルポートは、常に、対応する専用パケット形式のパケットデータを転送することができる。従って、バッファに専用エントリを設けることにより、ハイアラーキースイッチを通る１つの形式のパケットの進行が、そのスイッチを通る他のパケットの進行によって左右されないように効果的に保証される。

バッファ１６０ａ及び１６０ｂに専用及び一般的なスロットを使用することにより、各パケット形式ごとに最小数のスロットを指定するだけでよい。トランシット状態のパケットの数を追跡することにより、流れ制御を微細な粒度で行うことができる。バッファの利用性及びバスの帯域巾の両方が最大にされる。例えば、Ｘの一般的スロットしか使用できないときには、流れ制御が１サイクル放棄され、そして次のサイクルに再アサートされる。その結果、Ｘまでのメッセージを時間周期内に受け取ることができる。

【００７８】

図１２Ｂは、ハイアラーキースイッチへデータを供給するためにグローバルポートにより使用される方法を示すフローチャートである。このプロセスは、１つの形式のパケットについて説明するが、他の形式のパケットにも容易に拡張できる。ステップ１６９では、ハイアラーキースイッチ１５５へ供給すべきパケットがバッファ１３５ａ－１３５ｄの１つに存在するかどうかＧＳアービター１３４において決定される。パケットがある場合には、ステップ１７１において、「流れ制御」信号の状態がアービター１３４により評価される。「流れ制御」信号がアサートされる場合には、ステップ１７２において、ハイアラーキースイッチにより送られるべきパケットの特定形式に対する専用カウントを検査して、それがゼロに等しいかどうか決定される。専用カウントがゼロに等しくない場合には、その形式のパケットに対するバッファ内の専用エントリが既に使用中であり、プロセスはステップ１７０へ戻り、そのパケット形式の専用カウントがゼロに等しくなるまで又は流れ制御信号がデアサートされるまで、ステップ１６９、１７１及び１７２間をループする。

ステップ１７２において専用カウントがゼロに等しいと決定された場合には、ステップ１７３において、ＧＰアービター１３４は、適当な選択信号をマルチプレクサ１６７へアサートし、所望のパケットをハイアラーキースイッチ１５５へ送信する。ステップ１７４において、パケットの選択された形式に対応する専用カウントがグローバルポートの専用カウントレジスタ１３４及びハイアラーキースイッチ１５５の専用カウントレジスタ１６２ａにおいて増加され、そしてフラグレジスタ１６３ａの関連フラグがアサートされる。

【００７９】

上記のように、フラグレジスタ１６３ａは、一般的カウント及びトランシットカウントと共に使用されて、次のデータサイクルに対する「流れ制御」信号の状態を決定する。図１３には、ハイアラーキースイッチによる「流れ制御」信号のアサートを制御するためのプロセスの一実施形態が示されている。ステップ１７５において、フラグレジスタ１６３ａが検査されて、ゼロに等しい専用カウントエントリの数が計数される。上記のように、ゼロの数は、「流れ制御」がアサートされた後であってもバッファに接続された各ノードにより送られる潜在的な専用パケットの数を指示する。従って、図１１の例においていずれのノードについても専用スロットが全く使用されない場合には、フラグレジスタの全てのエントリがゼロに等しくなり、従って、専用パケットのために指定されねばならないバッファ位置が３つあることを指示する。

【００８０】

フラグレジスタ１６３ａの値が検査された後、ステップ１７６において、使用可能な全一般的スロットが上記式Ｉを用いて決定される。次いで、ステップ１７７において、各ノードのトランシットカウントが決定される。上述したように、トランシットカウントは、所与のデータサイクル中にグローバルポートとハイアラーキースイッチとの間でトランシット状態にあるメッセージの数を示す。最悪の場合のトランシットカウントは、流れ制御の待ち時間ＬにバッファＮを使用するノードの数を乗じたものに等しい。しかしながら、本発明の１つの実施形態によれば、トランシットカウントの決定には、「流れ制御」信号が手前のサイクル中にアサートされたかどうかが考慮される。上記のように、「流れ制御」信号が手前のサイクルにアサートされた場合には、グローバルポートとハイアラーキースイッチとの間でトランシット状態となるパケットはない。例えば、手前のＪ個の周期中に「流れ制御」がゼロであった場合には、ＪｘＮ個までのメッセージがトランシット状態となる。しかしながら、Ｊ－１個の手前のデータサイクル中に「流れ制御」信号がゼロであった場合には、（Ｊ－１）ｘＮ個のメッセージのみがトランシット状態となる。

【００８１】

従って、本発明の１つの実施形態では、ソース（グローバルポート）と行先（ハイアラーキースイッチ）との間の全待ち時間を検査すると共に、手前のデータサイクルにおけるソースと行先との間の相互作用を検査することにより、トランシット状態のパケットの数がインテリジェントに決定される。各ノードに対するトランシットカウントが決定された後に、ステップ１７８において、上記の式ＩＩを用いて保留中の専用パケット及びトランシット状態のパケットを受け入れるに充分な使用可能な一般的エントリがバッファにあるかどうかの判断がなされる。使用可能な一般的パケットの全数が、トランシット状態にあるパケットの数にバッファを共用するノードの数を乗じた値より少ない場合には、ステップ１７８において、「流れ制御」信号がグローバルポート１２０ａにアサートされ、ハイアラーキースイッチ１５５へのデータの供給が阻止される。しかしながら、全カウントが、潜在的に受け取られるパケットの数をバッファ１６０ａで受け入れできることを指示する場合には、「流れ制御」信号がアサートされず、プロセスは、次のデータサイクルのためにステップ１７５へ復帰する。

【００８２】

従って、トランシット状態にあるメッセージの数と、流れ制御信号がアサートされた手前のサイクルの数とを追跡することにより、流れ制御は、グローバルポートをハイアラーキースイッチに接続するデータリンクの利用性が最大となるよう確保するように微同調される。　図１１ないし１３に示すバッファ書き込み制御ロジック及び方法は、ノードからハ

イアラーキースイッチへのデータの送信に関して説明したが、本発明は、このような構成に限定されるものではないことに注意されたい。むしろ、本発明の１つの実施形態は、共通の受信器に信号供給する多数のソースがありそして停滞を回避する必要があるいかなる環境にも使用できる。

【００８３】

チャンネル順序付け制約をサポートするハイアラーキースイッチの機構：

ハイアラーキースイッチからのデータの読み取りは、本質的に、パケットの順序と、パケット間のデータ依存性との両方が維持されるように入力バッファから多数の出力ソースへデータを供給することを含む。上述したように、パケットは種々のチャンネルに供給される。異なるチャンネルにおいてパケットに関連するのは、ある順序付け制約即ち依存性である。本発明の１つの実施形態では、１つの順序付け制約は、Ｑ１チャンネルの全てのパケットが順序正しく維持されることである。別のパケット順序付け依存性は、優先順位の高いチャンネルを進行するパケットが、優先順位の低いチャンネルを進行するパケットによって阻止されてはならないことであり、チャンネルの優先順位は、最も高いものから最も低いものへ、Ｑ２、Ｑ１、Ｑ０、ＱＯＶｉｃ及びＱＩＯである。順序の維持は、以下に述べる種々の技術を用いてＳＭＰ全体にわたり達成される。ハイアラーキースイッチにおいては、データ依存性及びＱ１チャンネル順序付けを満足するよう確保するために３つの基本的なガイドラインに従う。これらのガイドラインは、次の通りである。

【００８４】

ガイドライン１：所与のハイアラーキースイッチ入力ポートに受け取られた多数のＱ１パケットが共通の出力ポートをターゲットとする場合には、Ｑ１パケットは、それらが入力ポートに現れたのと同じ順序で出力ポートに現れる。

ガイドライン２：ハイアラーキースイッチにおいて多数の入力ポートからのＱ１パケットが共通の出力ポートへマルチキャスティングされるときには、Ｑ１パケットは、それらがターゲットとする全ての出力ポートに同じ順序で現れる。

ガイドライン３：ハイアラーキースイッチの多数の入力ポートからのＱ１パケットの順序付けリストが多数の出力ポートをターゲットとするときには、Ｑ１パケットは、全ての到来するＱ１パケットの単一の共通の順序付けに合致するように出力ポートに現れる。各出力ポートは、共通の順序付けリストにおける幾つかの又は全てのパケットを送信することができる。

【００８５】

コヒレンス性の目的で全体的なシステム順序を維持するのに加えて、スイッチから出力されるパケットを、アドレス及びデータバスの性能が完全に実現されるように順序付けすることも望まれる。例えば、図１４は、ＨＳリンク１７０のアドレス及びデータバス構造の利用を示すタイミング図である。

ＨＳリンク１７０は、２対の単一方向性アドレス及びデータバスによりマルチプロセッサノード１００の各々に接続される。データバスは、５１２ビットのデータパケットを搬送し、そしてアドレスバスは、８０ビットのアドレスパケットを搬送する。データパケットの送信は、アドレスパケットの送信の２倍のサイクル数を必要とする。書き込みコマンドのようなあるコマンドは、アドレス及びデータパケットの両方を含む。例えば、図１４において、アドレスパケット１７９ａは、データパケット１７９ｄに対応する。各コマンドがアドレス及びデータパケットの両方を含む場合には、アドレスバスの１つおきのアドレススロットがアイドル状態となる。しかしながら、読み取りコマンドのような多数のコマンドは、アドレスパケットしか含まず、データパケットを転送するためのデータバスのスロットを必要としない。従って、全体的なシステム性能を向上するためには、データ部分及びアドレス部分の両方が「パック」され、即ちＨＳリンクのアドレス及びデータ部分の各考えられるタイムスロットにアドレス及びデータが存在するような順序でバスから送出すべきパケットを選択するスイッチを有するのが好ましい。アドレス及びデータがＨＳリンクにおいて「パック」されるときには、ＨＳリンクが最適に利用される。

【００８６】

多数の入力ポートを経て多数のソースからデータを同時に受け取りそして多数の出力ポートを経て多数の行先へデータを供給できる一方、データ依存性を満足し、システム順序を維持し、そしてデータ転送レートを最大にすることのできるハイアラーキースイッチを実施するための種々の実施形態が提供される。これらの種々の実施形態を、図１５ないし１８を参照して説明する。

図１５には、上記順序付け制約を実施することのできるスイッチ１８１の１つの実施形態が示されている。図１１について述べたように、スイッチ１５５は、複数のバッファ１６０ａ－１６０ｈを含む。入力バッファの各々は、１書き込みポート／８読み取りポートバッファであり、８個の各入力の１つからパケットを受け取るように接続される。又、スイッチは、８個の出力ポートも含むが、１つの出力ポート、即ち出力ポート＜０＞のみに対するロジックが示されている。残りの出力ポートに対するロジックも同様であり、明瞭化のために、ここでは詳細に述べない。

【００８７】

本発明の１つの実施形態では、各バッファの各エントリは、バッファのエントリに記憶されるパケットのチャンネルを識別するチャンネルフィールド１８５を含む。更に、各エントリは、一連のリンクインデックス１８６を含む。各リンクインデックスは、入力バッファ１６０ａ－１６０ｈのエントリの１つに対するインデックスである。これらのリンクインデックスは、パケット順序付け制約に基づきバッファ１６０ａから同じチャンネルを経て次々のパケットをアクセスするためのリンクリストアドレス構造体を形成するのに使用される。３つのリンクインデックスＬ１、Ｌ２及びＬ３が各エントリと共に含まれ、各リンクインデックスは、３つまでの順序付けリストの１つにおけるエントリの位置を識別する。

又、各エントリは、依存性フラグ１８９も含む。依存性フラグは、チャンネル間の依存性をマークするのに使用される。依存性フラグＦ１は、対応するエントリのパケットがＱ１、ＱＩＯ又はＱ０Ｖｉｃチャンネルを進行するパケットである場合にセットされる。依存性フラグＦ２は、対応するエントリのパケットがＱ０又はＱ０Ｖｉｃチャンネルを進行するパケットである場合にセットされる。依存性フラグは、パケットの処理順序を次のように維持する上で助けとなる。

【００８８】

概念的に、受け取ったパケットは、Ｑ２チャンネル待ち行列、合成Ｑ１／ＱＩＯ／Ｑ０Ｖｉｃチャンネル待ち行列、合成Ｑ０／Ｑ０Ｖｉｃチャンネル待ち行列、Ｑ０Ｖｉｃチャンネル待ち行列及びＱＩＯ待ち行列を含む５つの順序付けされた待ち行列に分割される。従って、パケットは、２つ以上の待ち行列に含まれる。ヘッドポインタは、各待ち行列ごとに１つのポインタ１８７ａ－１８７ｅを含む。ヘッドポインタは、その待ち行列に対応するバッファにおける次のパケットを識別するバッファ１６０ａ－１６０ｈのインデックスを与えるのに使用される。従って、ヘッドポインタ１８７は、Ｑ２ヘッドポインタ１８７ａ、Ｑ１／ＱＩＯ／Ｑ０Ｖｉｃヘッドポインタ１８７ｂ、Ｑ０／Ｑ０Ｖｉｃヘッドポインタ１８７ｃ、Ｑ０Ｖｉｃヘッドポインタ１８７ｄ及びＱＩＯヘッドポインタ１８７ｅを含む。パケットが入力バッファに最初に書き込まれるときには、それが１つ以上の順序付けされた待ち行列に入れられる。１つ以上の順序付けされた待ち行列に入れられるときには、１つ以上の依存性フラグ１８９がアサートされる。チャンネルの形式及び依存性フラグが検査されて、チャンネル依存性を満足するように出力すべきバッファの適当なエントリが選択される。

【００８９】

８個の入力バッファ１６０ａ－１６０ｈ各々の各エントリは、マルチプレクサ１８２へ送られる。マルチプレクサ１８２は、マネージャー１８０からの選択信号に応答して入力バッファの１つからパケットの１つを選択する。マネージャー１８０は、入力バッファ１６０ａ－１６０ｈの６４個の考えられる読み取りポートからのエントリを関連出力ポートの出力として選択する。マネージャー１８０は、全体的なシステム順序及びチャンネル依存性が満足されるようにパケットを選択する。

入力バッファ１６０ａ－１６０ｈの１つにパケットが受け取られるときには、エントリのチャンネルフィールドにチャンネル形式が書き込まれ、そしてそのエントリの関連フラグがフラグフィールド１８９においてアサートされる。上述したように、入力バッファの各エントリごとに、３つのリンクインデックスがあり、その各々は、３つの順序付けされた待ち行列の１つに対応する。本発明の１つの実施形態では、パケットを３つの異なる出力ポートにマルチキャスティングするために多数のリンクインデックスが使用される。マルチキャスティングされるべきパケットが入力バッファに記憶されるときには、それが２つ以上のリンクされたリストに入れられ、リンクされたリストの各々は、異なる出力ポートに対応する。その結果、異なる出力ポートに関連する出力マネージャーは、各々、異なるリンクリストインデックスを用いて同じ入力バッファエントリにアクセスすることができる。

【００９０】

上述したように、リンクインデックス値は、バッファ１６０ａ－１６０ｈにおいて対応する形式の次のパケットをアドレスするためのバッファインデックス値である。従って、リンクインデックス値は、対応する形式のその後のパケットがバッファに書き込まれるまで書き込まれない。その後のパケットがバッファに書き込まれるときには、その後のパケットのアドレスが手前のパケットのリンクインデックスに書き込まれ、これにより、そのチャンネル形式の次のパケットのインデックスを与える。各エントリは、３つの考えられるリンクインデックスフィールドを含むので、手前のエントリにアドレスを書き込むのに加えて、２ビットフィールド（図示せず）がアドレスと共に記憶され、順序付けリストを構成するために３つのリンクインデックスの適当な１つをエントリで識別できるようにする。

【００９１】

マネージャー１８０は、出力ポートへ供給するためにバッファ１６０ａ－１６０ｈのパケットの１つを次のように選択する。上述したように、ヘッドポインタ１８７ａ－１８７ｅは、各待ち行列の最上部に対応するバッファインデックスを記憶する。所与のチャンネルに対するパケットを処理するときに、マネージャーは、対応するヘッドポインタにより指示されたエントリを選択する。１つ以上のフラグ１８９がセットされ、そして高い優先順位のチャンネルに関連した待ち行列のパケットが処理されていない場合には、パケットは、その待ち行列内のより優先順位の高い全ての手前のパケットが処理されるまで処理されない。

例えば、出力マネージャーがＱ０形式のパケットを処理する場合に、Ｑ１／ＱＩＯ／Ｑ０Ｖｉｃ及びＱ０／Ｑ０Ｖｉｃヘッドポインタで指示されたエントリを検査する。パケットがＱ０チャンネルパケットであるが、Ｑ１パケットの処理がまだ完了していない場合には、エントリは処理されない。パケットの処理は、チャンネルＱ１又はＱ０パケットが既に処理されたことを指示する処理フラグ（図示せず）を各フラグＦ１及びＦ２と共に与えることにより指示される。高い優先順位のチャンネルを有する待ち行列における全てのパケットの処理が行われると（処理フラグにより指示される）、そのエントリに関連したパケットは自由に処理される。

【００９２】

あるエントリが処理のために選択されると、マネージャーは、そのエントリが存在する待ち行列に関連したヘッドポインタをバッファインデックスとして選択する。バッファインデックスはマルチプレクサ１８２へ送られ、そしてバッファエントリが出力ポートへ送られる。リンクインデックスはヘッドポインタへ返送され、そしてヘッドリストポインタがその待ち行列の次のパケットのバッファインデックスで更新される。従って、図１５のスイッチ実施形態は、リンクリストデータ構造体、順序付けされた待ち行列及びフラグを用いて、出力ポートへパケットを与え、全体的なシステム順序が維持されるようにする。更に、多数のリンクインデックスを含むリンクリストデータ構造体は、マルチキャストパケット順序付けルールに固執しながらパケットをマルチキャスティングするための簡単な機構を形成する。

【００９３】
従って、図１５の実施形態は、フラグ及び順序付けされた待ち行列を使用して、チャンネルの順序が維持されるようにする。図１６には、所定の順序依存性に基づいて出力データを与えることのできるスイッチの第２の実施形態が示されている。図１６の実施形態では、スイッチの各出力ポートに対してバッファ２００が設けられる。バッファ２００は、入力パケット受信経路２０１を経てバッファ１６０ａ－１６０ｈ（図１１）の各々から入力を受け取るように接続され、入力バッファからのパケットは、パケットの行先に基づいて出力ポートの適当なバッファへ送られる。本発明の１つの実施形態では、バッファは、コラップス(collapsing)ＦＩＦＯとして実施されるが、当業者に知られた他のバッファアーキテクチャｘを使用することもできる。

【００９４】
バッファ２００は、スイッチから送出されるべき種々のパケットを記憶するように示されている。バッファ２００は、ここでは、５つの異なるチャンネルＱ０、Ｑ１、Ｑ２、Ｑ３及びＱ４を経て送信されるパケットを記憶する。チャンネルＱ０－Ｑ４は、上記のチャンネルＱ０、Ｑ１、Ｑ２、Ｑ０Ｖｉｃ及びＱＩＯと同様ではない。むしろ、これらは、単にスイッチの出力動作を示すためにのみ使用される。従って、パケットＱ０－Ｑ４は、異なるチャンネルにおける一般的パケットを表わし、チャンネルの依存性は、図１６Ａの流れ図において矢印に基づいて定められる。図１６Ａにおいて、あるチャンネルから別のチャンネルへ向けられた矢印は、第１チャンネルのパケットが出力ポートへ送られず、一方、第１チャンネルのパケットの前に受け取られた第２チャンネルのパケットは、スイッチによる処理が保留中であることを指示する。例えば、図１６Ａにおいて、チャンネルＱ０のパケットは、チャンネルＱ３のパケットの処理に依存するように示されており、従って、チャンネルＱ０のパケットは、チャンネルＱ３のパケットを「プッシュ」したと言える。図１６Ａの流れ図に示された付加的な依存性は、チャンネルＱ１のパケットがチャンネルＱ２及びＱ３のパケットをプッシュしたことを指示する。この場合も、図１６Ａの流れ図で表わされた依存性は、既に述べたＱ０、Ｑ１、Ｑ２、Ｑ０Ｖｉｃ及びＱＩＯチャンネルの依存性を表わすものではないことに注意されたい。以下に述べるように、Ｑ０、Ｑ１、Ｑ２、Ｑ０Ｖｉｃ及びＱＩＯチャンネルにおけるパケットの依存性は複雑であり、従って、バッファ２００の動作を容易に説明するために、一般的パケット及び依存性が与えられる。

【００９５】
上述したように、入力パケットは、スイッチの入力バッファ１６０ａ－１６０ｈの各々に正しい順序で受け取られ、そしてそのパケットにより指示された行先に基づいて、バッファ２００のような出力バッファに正しい順序で供給される。各出力バッファの各パケットエントリ、例えば、エントリ２００ａは、パケットの送信及び受信ノードを指示するソース及び行先フィールドと、パケットが送信されるチャンネルを指示するチャンネルフィールドと、一連のビット２０６ａ－２０６ｅとを備えている。一連のビット２０６ａ－２０６ｅは、ハイアラーキースイッチを経てパケットを供給する各チャンネルごとに１ビットを含む。例えば、図１６の実施形態では、一連のビットは、チャンネルＱ０、Ｑ１、Ｑ２、Ｑ３及びＱ４の各々について１ビットを含む。

【００９６】
出力ポートに対して入力パケット受信経路に接続された書き込み制御ロジック２０５は、受信パケットのチャンネルに基づくと共に、図１６Ａの流れ依存性図に示されたチャンネル間の依存性に基づいて一連のビットの各々の設定を制御する。又、以下に詳細に述べるように、書き込み制御ロジックは、静的又は動的に依存性を確認することによりビットを更新することができる。依存性を静的に確認するときには、チャンネルに対して定められた依存性が、バッファ内の他のパケットに拘わりなく適用される。依存性を動的に確認するときには、チャンネルの依存性が、バッファ２００内の他のパケットのチャンネル及びアドレス行先を考慮して適用される。
一連のビットの各々には、対応するサーチエンジン２０８ａ－２０８ｅが接続される。各

サーチエンジンは、ビットの関連列をサーチして、列セットの対応ビットを有するバッファ２００のエントリを選択する。選択されたエントリは、各列（又はチャンネル）ごとに、一連の信号Ｓ４－Ｓ０により出力バッファマネージャー２０２へ指示される。チャンネル間の既知のデータ依存性に関連してサーチエンジンの各々により受信された選択信号を用いて、出力バッファマネージャーは、グローバルポート出力に供給するために出力バッファ２００からのパケットの１つを選択する。

【００９７】

動作中に、入力パケット受信経路２０１を経てパケットが受信されるときに、パケットのチャンネルは、書き込み制御ロジック２０５により評価され、そして一連のビット２０６ａ－２０６ｅのうちの、そのチャンネルに対応するビットがアサートされる。図１６において、パケットの形式を指示するためにセットされたビットは、「丸内のＸ印」で示され、そしてこれはチャンネル識別子フラグと称する。従って、図１６では、パケット１がＱ３形式のパケットである。図１５の実施形態によれば、エントリのチャンネルを指示するビットをアサートするのに加えて、そのチャンネルのパケットがプッシュするところの各チャンネルに対してビットが付加的にアサートされる。これらビットの各々は、依存性フラグと称され、図１６に「Ｘ」で示されている。それ故、Ｑ０チャンネルパケットであるパケット２の場合に、Ｑ３チャンネルパケットに関連したビットが付加的にアサートされる。というのは、図１６Ａの流れ図に示されるように、Ｑ０パケットがパケットをプッシュするからである。

【００９８】

パケットがバッファ２００に記憶され、そしてそれらの関連する一連のビット２０６ａ－２０６ｅがアサートされるときには、ビットの各列に関連したサーチエンジン２０８ａ－２０８ｅの各々が、ビットセットを有するバッファ内の第１エントリを選択する。それ故、サーチエンジン２０８ａの選択値は、パケット２を指し、サーチエンジン２０８ｂの選択値は、パケット３を指し、等々となる。

Ｓ０－Ｓ４信号は、マネージャー２０２に送られる。マネージャー２０２は、サーチエンジンによる選択信号のアサートに応答するのに加えて、システムに存在する依存性に応答して、パケットの１つを選択する。例えば、本発明の１つの実施形態によれば、チャンネルＱ０にあるパケット２のようなパケットは、チャンネルＱ０のサーチエンジン（２０８ａ）及びチャンネルＱ３のサーチエンジン（２０８ｄ）の両方が同じパケットを選択しない限り、スイッチから送出されない。従って、多数のフラグが所与のパケットに対してセットされたときに、マネージャー２０２は、セットされたフラグに対応するサーチエンジンの両方がその所与のパケットを選択しない限り、出力に対してそのパケットを選択しない。

【００９９】

本発明の別の実施形態によれば、サーチエンジンが、その依存性フラグがセットされたためにエントリを選択した場合に、サーチエンジンは、依存性フラグをクリアしそしてバッファを下方に進んで、依存性フラグ又は認識フラグがセットされた次のエントリを選択することができる。このような構成では、サーチエンジンが他のチャンネルによりストールされて処理を保留にすることがないので、パケットの処理が改善される。

依存性を識別するために多数のフラグをアサートする作用は、パケットがスイッチを経て伝播するときにパケットの全体的なシステム順序を維持する上で助けとなる。例えば、図１６において、Ｑ０パケットとＱ３パケットとの間の関係は、Ｑ０チャンネルパケットが実行の前に各手前のＱ３チャンネルパケットをプッシュすることである。従って、Ｑ３チャンネルパケットの後に受け取られたＱ０チャンネルパケットは、Ｑ３パケットの前に実行されてはならない。パケット１は、パケット２のＱ０チャンネルパケットの前に受け取られるＱ３チャンネルパケットである。パケット２に対してビット２０６ｄをセットすることにより、パケット２のＱ０パケットがパケット１のＱ３パケットの前に出力ポートに発生されないよう確保することができる。というのは、マネージャー２０８は、Ｓ３及びＳ０の両方がパケット２への信号を選択するまでＱ０パケットを選択しないからである。

Ｓ３値は、パケット１が処理されるまでパケット２を指さない。その結果、所与のチャンネルのパケットによりプッシュされた各パケットごとにビットをアサートすることにより、所与のチャンネルによりプッシュされたパケットが処理されるまでチャンネルが効果的に阻止される。その結果、全体的なシステム順序が維持される。

【０１００】

上記のように、図１６のバッファ制御ロジックは、静的又は動的な依存性を確認するように動作される。静的な依存性とは、図１６Ａの流れ図で示されたような依存性である。動的な依存性は、バッファ内の２つのパケット間に静的な依存性が実際に存在するかどうかを決定するためにバッファの内容を評価することにより確認される。静的な依存性は、メモリデータがＳＭＰにおいてコヒレンス性を失わないよう確保する順序付けルールを形成するのに使用される。しかしながら、データのコヒレンス性は、パケットがメモリデータの同じブロックをアクセスする場合にしか影響されない。それ故、動的な依存性は、バッファに既にあるパケットの行先アドレスを検査することによってバッファの内容を微細な粒度で検査して、異なるチャンネルの２つのパケット間に依存性が実際に存在するかどうか決定する。

【０１０１】

バッファ２００内のパケット間の依存性を動的に確認する１つの効果は、バッファ内のパケットを処理するのに必要な時間を短縮することである。例えば、上記のパケット１及びパケット２の動作を使用すると、Ｑ０パケット２及びＱ３パケット１が同じアドレスにマップしない場合には、何ら問題なく、Ｑ０パケットをＱ３パケットの前に処理することができる。手前のＱ３パケットの処理を待機する際に受ける遅延時間が排除され、これにより、ＳＭＰシステムの全体的な性能が改善される。

例えば、図１７は、依存性を動的に確認することによるプロセスへのパケットの選択動作を示すフローチャートである。ステップ２２０において、パケットがバッファ２００に受け取られる。ステップ２２２において、パケットのチャンネルに対するビットが書き込み制御ロジック２０５により一連のビット２０６においてセットされる。ステップ２２４において、バッファ２００に記憶された手前のパケットが検査されて、パケットがプッシュするところのチャンネルのパケットがメモリの同じブロックにあるかどうか決定される。それらがメモリの同じブロックにある場合には、ステップ２２６において、パケットがプッシュするところのチャンネルにあり且つ同じメモリブロックに存在するパケットに対応するビットがアサートされる。従って、パケット２に対して図１６の例を使用すると、パケット形式Ｑ３に対するビットは、パケット１がパケット２と同じメモリブロックをアクセスする場合だけアサートされる。従って、依存性を動的に確認することにより、全体的なシステム性能を向上しながらメモリコヒレンス性を維持することができる。

【０１０２】

図１８には、全体的なシステム順序を維持しながら、多数の入力ソースから受け取ったデータを多数の出力ソースへ出力する方法の別の実施形態が示されている。図１８の実施形態は、図１６の場合と同様の要素を含むように示されている。しかしながら、図１８の書き込み制御ロジック２０９は、パケットの依存性を異なるやり方で分析することにより一連のビット２０６ａ－２０６ｅの各々を更新する。図１６の場合のように、パケットが関連チャンネルのものであることを指示するために、一連のビットの１つが各パケットごとにセットされる。しかしながら、チャンネルがプッシュするところのチャンネルの全てのパケットに対して付加的なビットをセットするのではなく、そのチャンネルのパケットをプッシュするところのチャンネルのパケットに対してビットがセットされる。

【０１０３】

従って、図１８の実施形態は、チャンネル識別フラグをセットするのに加えて、そのパケットによりマスク又は阻止された全てのチャンネルに対して付加的なビットがセットされる。例えば、図１８の例において、パケット１は、Ｑ３チャンネルパケットである。Ｑ３チャンネルのパケットは、図１８Ａの依存性流れ図に示すようにＱ３パケットが実行されるまで、Ｑ１及びＱ０パケットの実行を阻止する。従って、ビット２０６ｄ、２０６ｂ及

び２０６ａがパケット１に対してセットされる。しかしながら、パケット２は、他のパケットの実行を阻止しないＱ０パケットである。その結果、ビット２０６ｂのみがパケット２に対してセットされる。
従って、図１８のスイッチ実施形態は、依存性を静的に確認することによりシステム順序を維持しながら出力ポートへデータを供給する別の方法を提供する。図１８のバッファ実施形態は、依存性を動的に確認するようには使用できないことに注意されたい。というのは、そのようにするには、データがバッファ２００に書き込まれる前にデータのアドレスを知る必要があるからである。しかしながら、ここに述べる静的及び動的な方法は、全て、パケット間の依存性を満足するよう確保するために使用できる。

【０１０４】

従って、多数の入力ポートを経て多数のソースからデータを同時に受け取りそして多数の出力ポートを経て多数の行先へデータを供給できる一方、データ依存性を満足し、システム順序を維持し、そしてデータ転送レートを最大にすることのできるスイッチの３つの実施形態が説明された。１つの実施形態では、フラグを記憶する多数の待ち行列の使用により順序付け依存性が達成されそして依存性を識別するように待ち行列が選択されるリンクリストバッファ機構が説明された。第２および第３の実施形態では、スイッチの入力バッファからデータを正しい順序で受け取る出力バッファが、ある形式のパケットを阻止するのに使用される一連のビットを備えていて、データ依存性及びコヒレンス性制約を満足するように確保する。全ての実施形態において、潜在的な依存性の競合をマークするためにセットされるフラグを含む順序付け待ち行列の使用により、順序付け依存性が追跡される。フラグの順序付けリストを用いて依存性を識別することにより、バスの利用性を最大にしながら順序を維持し且つコヒレンス性を確保するためにマネージャーにより実行されるオペレーションの複雑さが簡単化される。

【０１０５】

キャッシュコヒレンスプロトコル

本発明の１つの実施形態におけるキャッシュコヒレンスプロトコルは、書き込み無効化所有権をベースとするプロトコルである。「書き込み無効化」とは、プロセッサがキャッシュラインを変更するときに、他のプロセッサキャッシュにおける効力のないコピーを無効化することを意味し、新たな値でそれらを更新するのではない。このプロトコルは、システム内のメモリであるかプロセッサ又はＩＯＰの１つであるかに拘わりなくキャッシュラインに対する識別可能な所有者が常に存在するので、「所有権プロトコル」と称される。キャッシュラインの所有者は、必要なときにキャッシュラインの最新の値を供給する責任がある。プロセッサ／ＩＯＰは、キャッシュラインを「独占的に」又は「共用して」所有することができる。プロセッサがキャッシュラインの独占的所有権を有する場合には、システムに通知せずにそれを更新することができる。さもなくば、システムに通知し、そして他のプロセッサ／ＩＯＰキャッシュのコピーを潜在的に無効化しなければならない。

【０１０６】

キャッシュコヒレンスプロトコルの詳細な説明に入る前に、ハイアラーキーネットワークに使用される全通信手順について最初に説明する。
図７Ａについて述べたように、大型のＳＭＰシステム１５０は、スイッチ１５５を経て互いに接続された多数のノードを含む。各ノードにおける各プロセッサは、メモリのデータにアクセスするコマンドを発生する。これらのコマンドは、ソースノード内で完全に処理することもできるし、又はアドレス及び要求の形式に基づいてシステムの他のノードへ送信することもできる。
アドレススペースは、メモリスペース及びＩＯスペースに分けられる。プロセッサ及びＩＯＰは、専用キャッシュを使用して、メモリスペースアドレスのみに対するデータを記憶し、そしてＩＯスペースデータは、専用キャッシュには記憶されない。従って、キャッシュコヒレンスプロトコルは、メモリスペースコマンドのみに関連している。

【０１０７】

キャッシュコヒレンスプロトコルの重要な要素は、ロード及び記憶動作をシリアル化する

解決策である。キャッシュコヒレンスプロトコルは、各メモリアドレスＸへの全てのロード及び記憶に順序を課さねばならない。この順序は、Ｘへの全ての「記憶」が順序付けされるものであり、即ち、第１記憶、第２記憶、第３記憶、等々とならねばならない。第ｉ番目の記憶は、（Ｉ－１）番目の記憶により決定されたようにキャッシュラインを更新する。更に、各ロードには最新の記憶が関連され、そこからロードはキャッシュラインの値を得る。この順序をここでは「ロード－記憶シリアル化順序」と称する。
ここに述べるプロトコルの特性は、アドレスＸに対するホームＡＲＢバスが、Ｘへの全てのロード及び記憶に対する「シリアル化ポイント」であることである。即ち、Ｘへの要求がＸのホームＡＲＢバスに到着する順序は、対応するロード及び記憶がシリアル化される順序である。大型のＳＭＰシステムに対するほとんどの公知のプロトコルは、この特性を有しておらず、従って、効率が悪く、複雑である。

【０１０８】
図２に示す小型のＳＭＰノードシステムには、１つのＡＲＢバスがある。このバスは、小型ＳＭＰにおける全てのメモリロード及び記憶に対するシリアル化ポイントである。ＡＲＢバスに接続されたＤＴＡＧは、小型ＳＭＰのプロトコルに必要とされる全ての状態を捕獲する。大型のＳＭＰシステムでは、ホームＡＲＢバスのＤＩＲがプロトコルに対するおおよその状態を捕獲し、ＴＴＴ及びＤＴＡＧは、より微細なレベルにおける状態情報を捕獲する。　要求Ｒがホーム ＡＲＢバスに到着すると、ＤＩＲ、ＤＴＡＧ及びＴＴＴ状態が検査され、他のプロセッサへの調査コマンド及び／又はソースプロセッサへの応答コマンドを発生することができる。更に、ＤＩＲ、ＤＴＡＧ及びＴＴＴの状態が要求Ｒの「シリアル化」を反映するように自動的に更新される。従って、要求アドレスがＲのアドレスに等しく且つ要求Ｒの後にホームＡＲＢに到着する要求Ｑは、ロード－記憶シリアル化順序においてＲの後に現れる。

【０１０９】
その結果、ホームＡＲＢバスは、メモリアドレスへの全ての要求に対し「シリアル化ポイント」と定義される。各メモリアドレスＸに対し、対応する要求（ＲｄＭｏｄ又はＣＴＤ）がホームＡＲＢバスに到着する順序で記憶が見掛け上実行される。アドレスＸへのロードは、ホームＡＲＢにおいて最後にシリアル化された記憶Ｘに対応するＸのバージョンを得る。　以下に述べるキャッシュコヒレンスプロトコルの前書きにおいて、「システム」という用語は、プロセッサ及びＩＯＰを除く大型ＳＭＰの全ての要素を指す。プロセッサ及びシステムは、「コマンドパケット」又は単に「コマンド」を送信することにより互いに対話する。コマンドは、要求、調査及び応答の３つの形式に分類される。
プロセッサによりシステムに発生されるコマンド及びシステムによりプロセッサに発生されるコマンドは、所与のプロセッサのメモリシステムインターフェイスに基づく。ＳＭＰの動作を説明する目的上、デジタル・イクイップメント・コーポレーションからのＡｌｐｈａ（登録商標）システムインターフェイスの定義に基づいて発生される要求及びコマンドについて説明するが、他の形式のプロセッサも使用できることを理解されたい。

【０１１０】
要求は、ロード又は記憶動作を実行する結果として、データのコピーを得なければならないときにプロセッサにより発生されるコマンドである。又、要求は、システムからのデータの断片に対して独占的な所有権を得るのにも使用される。要求は、読み取りコマンド、読み取り／変更（ＲｄＭｏｄ）コマンド、ダーティへの変更コマンド、ビクティムコマンド、及びエビクト(Evict）コマンド（データのキャッシュラインが各キャッシュから除去される場合）を含む。
調査(Probe）コマンドは、データ及び／又はキャッシュタグ状態更新を要求する１つ以上のプロセッサへシステムにより発生されるコマンドである。調査コマンドは、送信読み取り(Forwarded Read)（ＦＲｄ）コマンド、送信読み取り変更(Forwarded Read Modify）（ＦＲｄＭｏｄ）コマンド、及び無効化コマンドを含む。プロセッサＰがシステムへの要求を発生するときには、システムは、１つ以上の調査コマンドを他のプロセッサへ発生しなければならない。Ｐがキャッシュラインのコピーを要求する（読み取り要求で）場合には

、システムは、所有者プロセッサ（もしあれば）へ調査コマンドを送信する。Ｐがキャッシュラインの独占的所有権を要求する（ＣＴＤ要求で）場合には、システムは、キャッシュラインのコピーをもつ１つ以上のプロセッサへ無効化調査コマンドを送信する。Ｐがキャッシュラインのコピー及びキャッシュラインの独占的所有権の両方を要求する（ＲｄＭｏｄ要求で）場合には、システムは、データのキャッシュラインのダーティコピーを現在記憶しているプロセッサにＦＲｄコマンドを送信する。ＦＲｄコマンドに応答して、キャッシュラインのダーティコピーがシステムに返送される。又、送信読み取り変更（ＦＲｄＭｏｄ）コマンドも、キャッシュラインのダーティコピーを記憶しているプロセッサにシステムにより発生される。ＦＲｄＭｏｄに応答して、ダーティキャッシュラインがシステムに返送され、そしてキャッシュに記憶されたダーティコピーが無効化される。キャッシュラインを別のプロセッサにより更新すべきときには、キャッシュラインのコピーをキャッシュに記憶しているプロセッサに、システムにより無効化コマンドが発生される。

【０１１１】

応答は、プロセッサにより要求されたデータ又は要求に対応する確認を搬送するシステムからプロセッサ／ＩＯＰへのコマンドである。読み取り及びＲｄＭｏｄコマンドの場合に、応答は、各々要求されたデータを搬送するＦｉｌｌ又はＦｉｌｌＭｏｄコマンドである。ＣＴＤコマンドの場合に、応答は、ＣＴＤの成功又は失敗を指示するＣＴＤ成功又はＣＴＤ失敗コマンドである。ビクティムコマンドの場合には、応答がビクティム－リリースコマンドである。

図１９には、要求と要求との間の関係、及び個々のプロセッサにおける対応キャッシュラインの状態を説明するためのテーブルが示されている。又、図１９は、キャッシュラインの要求及び状態の各々に対して得られる調査形式のコマンドも示している。カラム３００及び３００ａは、プロセッサにより発生される要求を示し、カラム３０５及び３０５ａは、システムの他のプロセッサにおけるキャッシュの状態を示し、そしてカラム３２０及び３２０ａは、システムにより発生される調査コマンドを示す。

【０１１２】

図１９のテーブルは、プロセッサＡと称するプロセッサがシステムに要求を発生することを仮定している。プロセッサＡのコマンドは、次いで、プロセッサＢと称する１つ以上の他のプロセッサと相互作用する。プロセッサＡによりアドレスされるキャッシュラインが、ＤＴＡＧ及び／又はディレクトリ情報を用いて決定されたプロセッサＢのキャッシュに記憶される場合には、プロセッサＢのキャッシュ状態が、プロセッサＢへ調査コマンドを発生する必要があるかどうか及びどんな形式の調査コマンドを発生すべきかを決定する。以下、コヒレンスプロトコル及び機構について詳細に述べる。コマンドパケットがとる経路、各コマンド形式に対する状態情報のソース、及びそれにより生じるアクションが含まれる。全てのコマンドは、プロセッサ又はＩＯＰから発生され、ＩＯＰの発生プロセッサは「ソースプロセッサ」と称する。要求に含まれるアドレスは、「要求アドレス」と称する。アドレスの「ホームノード」は、そのアドレススペースが要求アドレスをマップするところのノードである。要求は、ソースプロセッサが要求アドレスのホームノードである場合は「ローカル」と称し、さもなくば、「グローバル」要求と称する。ホームノードのＡＲＢバスは、「ホームＡＲＢバス」と称する。「ホームディレクトリ」は、要求アドレスに対応するディレクトリである。従って、ホームディレクトリ及びメモリは、要求アドレスに対するホームＡＲＢバスに接続される。

【０１１３】

プロセッサ又はＩＯＰから発せられるメモリ要求は、先ず、ホームＡＲＢバスにルート指定される。この要求は、それがローカルである場合にはローカルスイッチを経てルート指定され、それがグローバルである場合にはハイアラーキースイッチを経て送られる。後者の場合には、ローカルスイッチ及びＧＰリンクを横断してＧＰに達し、次いで、ＨＳリンクを経てハイアラーキースイッチへ至り、次いで、ＧＰ及びホームノードのローカルスイッチを経てホームＡＲＢバスへ至る。

グローバル要求は、ソースノードのＡＲＢバスに最初に現れず、むしろ、ＧＰリンクを経

てＨＳに直接ルート指定されることに注意されたい。公知のプロトコルでは、グローバル要求は、それが別のノードへ送出される前にソースノードの状態をアクセスする。本発明は、グローバル要求をＨＳへ直接発生することによりグローバル要求の平均待ち時間を短縮する。

【０１１４】

図２０Ａ－２０Ｊは、多数の基本的なメモリトランザクションを例示するフローチャートである。

ローカル読み取り：

図２０Ａにおいて、ソースプロセッサ３２０からホームＡＲＢバスへ要求が送られる。ディレクトリ３２２は、どのプロセッサがメモリブロックを所有するか決定する。ローカルメモリ３２３が所有者である場合には、「短い記入」コマンドがホームＡＲＢバスからソースプロセッサ３２０へ発生される。

【０１１５】

グローバル読み取り：

図２０Ｂにおいて、ノード３２５のプロセッサ３２０が、「ホーム」がノード３２６にあるメモリのキャッシュラインへ読み取りを発生すると仮定する。（グローバル）読み取りコマンドは、ライン３２７で示された経路を経てスイッチ３２４を通り「ホーム」ＡＲＢバス及びディレクトリ３２１へルート指定される。ノード３２６のメモリ３３０がキャッシュラインの所有者である場合には、「短い記入応答」を発生するノード３２６によりノード３２６からノード３２５へデータが返送される。

キャッシュラインが別のプロセッサ／ＩＯＰにより現在所有されている場合には、要求されたキャッシュラインを得るために異なるステップが取られる。図２０Ｃを参照すれば、プロセッサ３２０が、「ホーム」がノード３２６にあるメモリのキャッシュラインへ読み取りを発生する場合には、読み取りは、再び、経路３２７を経てホームＡＲＢバス及びディレクトリ３２１へルート指定される。ディレクトリ３２１のエントリは、上述したように、メモリの各キャッシュラインに対し、所有者情報を含む１４ビットの状態情報を備えている。所有者情報は、この場合に、所有者をノード３２８におけるプロセッサ３４２として識別する。

ノード３２８が要求されたキャッシュラインを所有するというディレクトリの指示に応答して、２つの事象が生じる。第１に、「ホーム」ノードであるノード３２６は、ライン３２９で示すように、所有者プロセッサ３４２へ「送信読み取り」調査を発生する。同時に、ホームノード３２６は、ライン３３１で示すように、プロセッサ３２０へ「記入マーカー」応答を送信する。「記入マーカー」応答の役割は、以下で説明する。

【０１１６】

「送信読み取り」に応答して、プロセッサ３４２は、「記入」コマンドをプロセッサ３２０へ発生し、「記入」コマンドは、当該キャッシュラインを含む。「読み取り」要求に対するこの形式の応答は、データ返送に対して一連の３つのコマンドを必要とするので、「長い記入」と称される。従って、「読み取り」トランザクションは、メモリからの応答である「短い記入」と、所有者プロセッサからの応答である「長い記入」の２つの形式に分割することができる。

ローカルＲｄＭｏｄ：

図２０Ｄを参照すれば、ローカル読み取り変更トランザクションは、（１）キャッシュラインの現在バージョンのコピーを得ている全てのプロセッサに無効化調査が送られ、そして（２）ＦＲＭｏｄ及びＦｉｌｌＭｏｄｓが、Ｆｒｄｓ及びＦｉｌｌｓに代わって所有者に送られる点を除くと、ローカル読み取りトランザクションと同様に働くことが明らかである。　図２０Ｄ図において、ホームノードのディレクトリは、ローカルプロセッサ又はメモリがブロックを所有することを示す。ホームＡＲＢバスにおいて、ディレクトリ３２２は、ブロックの現在バージョンを得ている全ての外部ノードを識別する。無効化コマンドは、ＨＳ３２４へ送られ、全ての当該ノードはマルチキャストベクトルで識別される。ＨＳは、ベクトルで識別された全てのノードへ無効化メッセージをマルチキャストする。

無効化メッセージは、各ノードにおいてＡＲＢバスへ進み、そこで、ＤＴＡＧは、それらを更にフィルタし、キャッシュラインの現在バージョンを有すると識別されたプロセッサ又はＩＯＰのみへ無効化調査を送信する。

【０１１７】

グローバルＲｄＭｏｄ：

図２０Ｅを参照すれば、読み取り変更トランザクションは、図２０Ａ及び２０Ｂについて述べた読み取りトランザクションと同様に作用することが明らかである。読み取り変更（ＲｄＭｏｄ）コマンドは、先ず、プロセッサ３２０からキャッシュラインのホームＡＲＢ及びホームディレクトリ３２１へルート指定される。ホームノードであるノード３２６のメモリがキャッシュラインを記憶する場合には、要求されたデータを含む「短い記入変更」コマンドがノード３２６からプロセッサ３２０へ送られる。ディレクトリ３２１は、このトランザクションの結果として更新される。

「読み取り変更」コマンドは、プロセッサ３２０が、キャッシュラインの内容を変更できるようにキャッシュラインの独占的所有権を要求することを指示する。それ故、「短い記入変更」コマンドに加えて、ノード３２６は、キャッシュラインの現在バージョンのコピーを得ている他の全てのプロセッサに「無効化」コマンドを発生する。ＤＩＲは、１つ以上のプロセッサがキャッシュラインの現在バージョンのコピーを得ているノードを識別する。ＤＩＲの存在ビットは、この情報を含む。ＤＴＡＧは、キャッシュラインのコピーを得ている全てのホームノードプロセッサを識別する。各々のＤＩＲ存在ビットがセットされた全てのノードに「無効化」が送信される。「無効化」を受信する各ノードにおいて、ＤＴＡＧをアクセスして、どのプロセッサがキャッシュラインのコピーを現在記憶するかを決定する。「無効化」は、これらのプロセッサのみに送られる。ＩＯＰタグは、ＩＯＰがコピーを有するかどうか決定するのに使用され、もしそうであれば、ＩＯＰは「無効化」調査も受け取る。

【０１１８】

要求を発しているプロセッサ以外のプロセッサが所有者である場合には、ホームノードは、「記入変更マーカー」、「送信読み取り変更」及びゼロ以上の「無効化」を１つのコマンドとして発生する。スイッチにおいて、コマンドは、全ての行先ノードにマルチキャストされる。各行先ノードにおいて、コマンドは、その要素に分離され、各ノードのグローバルポートは、各ノードにおいてどんなアクションをとるべきかを決定する。上記の例では、「送信読み取り変更」がプロセッサ３４２により処理され、そして「記入変更マーカー」がプロセッサ３２０により処理される。更に、ＤＴＡＧエントリに基づき、ホームノード、「記入変更マーカー」を受け取るノード、及び「送信変更」を受け取るノードにおいて「無効化」が実行される。「送信読み取り変更」に応答して、ダーティデータが「長い記入変更」コマンドを経てプロセッサ３４２からプロセッサ３２０へ送られる。

【０１１９】

従って、「読み取り変更」コマンドは、２つ又は３つのノード接続即ち「ホップ」を実行することができる。本発明の１つの実施形態では、読み取り型コマンド（「読み取り」及び「読み取り変更」）のみが３つのホップを生じ、但し、第３のホップは「記入」型コマンド（「記入」又は「記入変更」）である。しかしながら、本発明は、以下に述べる追加コマンドを仮想チャンネル待ち行列に適当に割り当てることにより３つ以上のホップを必要とする他のトランザクションも含むように容易に変更できる。

ＣＴＤ：

図２０Ｇ及び２０Ｈには、クリーン－ダーティ（ＣＴＤ）及び無効化－ダーティ（ＩＴＤ）の基本的な流れが示されている。図２０Ｇでは、クリーン－ダーティは、ホームノードにおいてプロセッサ３２０からディレクトリ３２１へ発生される。プロセッサ３２０が更新を希望するところのクリーンキャッシュラインが現在のものであるか効力を失ったものであるかに基づいて、「確認」コマンド（ＡＣＫ）又は「非確認」（ＮＡＣＫ）コマンドのいずれかがプロセッサ３２０へ返送される。対応的に、ＣＴＤは成功又は失敗と言える。更に、ＣＴＤが成功の場合にデータのキャッシュラインのコピーをもつものとしてディ

レクトリ３２１の存在ビットにより指示された全てのノードに「無効化」が送られる。
【０１２０】
図２０Ｈに示すように、ＩＴＤコマンドは、ＣＴＤと実質的に同様に働く。しかしながら、ＩＴＤは決して失敗とならない。ＡＣＫが常にプロセッサ３２０に送られ、そしてデータのキャッシュラインのコピーを記憶するシステムの他のノードには「無効化」が送られる。
ローカル及びグローバル書き込みビクティム：
上記のように、書き込みビクティムコマンドは、ダーティデータをプロセッサのキャッシュから適当なホームメモリへ返送する。図２０Ｉ及び２０Ｊを参照すれば、書き込みビクティムの流れは、「ホーム」メモリが書き込みビクティムを発生するプロセッサと同じノードであるかどうかに基づいて若干異なることが明らかである。図２０Ｉに示すように、「ホーム」ノードがプロセッサのノードである場合には、プロセッサ３２０が書き込みビクティムを発生し、そしてデータは、同じノードのメモリへ直接送られる。
【０１２１】
しかしながら、図２０Ｊに示すように、ビクティムデータがプロセッサとは異なるホームにある場合には、データが２つの段階で転送される。第１に、ビクティムキャッシュラインがプロセッサ３２０のキャッシュ（又はビクティムバッファ）から送出され、そしてプロセッサノードのグローバルポートにおけるビクティムキャッシュ（図６の要素１２４）に記憶される。ビクティムキャッシュは、「ビクティムリリース」信号でプロセッサに応答し、プロセッサがそのビクティムバッファエントリを再使用できることを指示する。次いで、スイッチに使用可能な帯域巾が存在するときには、ビクティムデータは、「書き込みビクティム」コマンドによりビクティムキャッシュからホームプロセッサのメモリへ送られる。
【０１２２】
ソースプロセッサＰによりホームメモリに送られたビクティムデータは、それがメモリに到達するときまでに効力を失うことがあることに注意されたい。このような場合に、ビクティムは、「失敗」と言われ、ホームメモリは更新されない。このようなケースは、Ｐがキャッシュラインの所有権を獲得するときと、Ｐのビクティムがホームディレクトリに到着するときとの間のインターバルに別のプロセッサがキャッシュラインの所有権を獲得したときに生じる。このような場合には、Ｐのビクティムがホーム ＡＲＢに到達する前に、キャッシュラインに対する「無効化」又は「ＦｒｄＭｏｄ」調査をプロセッサＰに送信しなければならない。
ビクティムデータをメモリに書き込まねばならないかどうか決定するために、「書き込みビクティム」コマンドがホームＡＲＢバスに現れるときに、要求されたアドレスに対するディレクトリエントリがルックアップされる。ソースプロセッサが依然としてキャッシュラインの所有者であることをディレクトリが指示する場合には、ビクティムが成功となり、メモリを更新する。さもなくば、失敗となり、メモリは更新しない。いずれにせよ、ディレクトリ３２１においてビクティムに対して判断がなされると、「ビクティムＡＣＫ」コマンドがノード３２５のグローバルポートに返送され、ビクティムキャッシュは関連エントリをクリアすることができる。
【０１２３】
この設計の１つの実施形態では、ＤＴＡＧを使用して、「書き込みビクティム」コマンドがローカルである場合に「書き込みビクティム」コマンドの成功又は失敗を判断する。この特定の例（ローカル「書き込みビクティム」要求の例）では、ＤＴＡＧ及びＤＩＲの両方が「書き込みビクティム」要求の成功又は失敗を決定するに必要な情報を与えることができる。ＤＴＡＧは、単にＤＴＡＧをベースとする機構が小型のＳＭＰノードハードウェアに対して既に設けられているという理由でＤＩＲに代わって使用される。
キャッシュコヒレンスプロトコルの上記説明では、最も一般的な動作及びコマンド形式について述べた。これら機構は、以下に詳細に説明する。
上記のように、本発明の１つの実施形態では、２つ以上の関連メッセージパケットを効率

化のために１つに結合することができる。結合されたパケットは、次いで、ＨＳ又はノードのＡＲＢバスにおいてその成分に分割することができる。例えば、ＨＳへのＦｒｄＭｏｄメッセージは、所有者プロセッサをもつノードへのＦｒｄＭｏｄメッセージと、キャッシュラインのコピーをもつノードへの「無効化」メッセージと、ソースノードへのＦｉｌｌＭａｒｋｅｒＭｏｄメッセージとに分割される。所有者プロセッサノードへのＦｒｄＭｏｄは、ノードのＡＲＢバスにおいて、所有者プロセッサへのＦｒｄＭｏｄメッセージと、ノードの他のプロセッサへのゼロ以上の「無効化」メッセージとに更に分割される。

【０１２４】

ビクティムコヒレンス性を維持するための遅延書き込みバッファ動作：

図２０Ｉ及び２０Ｊについて上述したように、ホームメモリに送られるビクティムデータは、「書き込みビクティム」がホームＡＲＢに到達する前に受け取られるキャッシュラインに対し「無効化」又はＦｒｄＭｏｄ調査が介在する結果としてそれが到着するときまでに効力を失うことがある。

ビクティムデータをメモリに書き込まねばならないかどうかを決定する１つの方法は、各書き込みビクティムコマンドに対してディレクトリエントリをルックアップすることである。ビクティム書き込みコマンドを発生するプロセッサがダーティ所有者であることをディレクトリが指示する場合には、ビクティムを進めることが許されねばならない。さもなくば、失敗となってしまう。この方法が望ましい理由は、プロセッサとシリアル化ポイントとの間のビクティム書き込みコマンドを、シリアル化ポイントとプロセッサとの間の調査コマンドと一致させるための複雑な比較論理構造体の必要性が回避されるからである。

【０１２５】

この解決策は、データコヒレンス性の維持を簡単化するが、メモリ帯域巾が減少するという形態の性能欠陥を生じさせる。この構成によれば、システムがビクティム書き込みコマンドを実行するたびに、先ず、ディレクトリ状態をアクセスし、次いで、その状態を評価し、そして最終的に、その状態に基づいて、ビクティムデータのＤＲＡＭ書き込みを実行しなければならない。メモリ及びディレクトリは原子的にアクセスされるので、公知の設計方法に基づいてシステムがシステムが設計された場合に、全ビクティム書き込みサイクルは、ディレクトリルックアップ時間と、状態評価時間と、ＤＲＡＭ書き込み時間との和に等しくなる。このようなシステムは、全ビクティムサイクルがＤＲＡＭ書き込みのみで構成されるシステムに対して甚だしい性能上の不利益をこうむる。

【０１２６】

本発明の１つの実施形態は、メモリの各バンクに遅延書き込みバッファを設けることにより、このメモリバンク利用低下問題を克服する。ビクティム書き込みがメモリシステムへ発生されるたびに、メモリシステムは、次の機能を並列に実行することにより応答する。即ち、ビクティム書き込みデータをターゲットメモリバンクの遅延書き込みバッファに記憶しそしてそのブロックを「非書き込み可能」又は「無効」と表示し、ビクティム書き込みに関連したディレクトリ状態をアクセスし、そして現在ビクティム書き込みに代わって、「書きこみ可能」又は「有効」と表示された既にバッファされたビクティム書き込みのＤＲＡＭ書き込みを実行する。ディレクトリアクセスが完了したときに、ビクティム書き込みに関連したディレクトリ状態が、ビクティム書き込みが成功したことを示す場合には、ビクティムが存在する遅延書き込みバッファが「書き込み可能」又は「有効」状態へと移行する。遅延書き込みバッファにおけるデータブロックの「書き込み可能」又は「有効」状態は、バッファのデータが、ＤＲＡＭメモリに記憶されたバージョンよりも最新のキャッシュラインのバージョンであることを指示する。バッファが「書き込み可能」又は「有効」と表示された場合には、そのデータが、メモリシステムへのビクティム書き込みのその後に発生によりＤＲＡＭへ書き込まれる。

【０１２７】

既に発生されたビクティム書き込みのＤＲＡＭ書き込みと並列にディレクトリルックアップを実行することにより、この実施形態は、全ビクティムサイクル時間を単一のＤＲＡＭ書き込み時間に減少する。この実施形態は、「書き込み可能」な又は「有効」なデータブ

ロックを多数のサイクルにわたり遅延書き込みバッファに保持し、そのサイクル中にバッファされたブロックへのその後の参照をメモリへ発生することができるので、遅延書き込みバッファは、連想アドレスレジスタを備えている。ビクティム書き込みブロックのアドレスは、その関連データが遅延書き込みバッファに記憶されるのと同時に連想アドレスレジスタに記憶される。その後の参照がメモリシステムへ発生されるときには、メモリシステムは、アドレスレジスタに対するアドレス一致により遅延書き込みバッファにおけるアドレスブロックを識別する。これは、メモリシステムが、ＤＲＡＭメモリの効力を失ったデータに代わってバッファからの最新のデータで遅延書き込みバッファのブロックへの全ての参照にサービスすることを意味する。

【０１２８】

ビクティムデータの遅延書き込みバッファ動作を与える上記技術は、ＤＴＡＧ状態を直接含まずにＤＴＡＧ状態を使用してデータブロックの有効性を決定するスヌーピーバスをベースとするシステムにも使用できる。

図２１を参照すれば、遅延書き込み動作を与えるメモリ制御システムの１つの実施形態は、ディレクトリ１４０からライン１４０ａを経てＯｗｎｅｒ＿Ｍａｔｃｈ信号を受け取るように接続されたメモリコントローラ３３２を含むように示されている。更に、メモリコントローラ３３２は、ディレクトリに入力されるコマンドを追跡するためにＱＳ　ＡＲＢ１１（ディレクトリ１４０にも信号供給する）からも入力を受け取る。

メモリコントローラ３３２は、遅延書き込みバッファ３３６を含む。遅延書き込みバッファ３３６の各エントリは、データ部分３３６ａと、フラグ部分３３６ｂと、アドレス部分３３６ｃとを含む。本発明の１つの実施形態において、設計上の複雑さを最小限にするために、遅延書き込みバッファは、１つのアドレス、データ及びフラグエントリのみを保持するが、本発明は、このような構成に限定されるものではない。

【０１２９】

遅延書き込みバッファは、次のように動作する。動作中に、コマンド、アドレス及びデータがＡＲＢ＿ＢＵＳ１３０を経て受け取られると、それらはディレクトリ１４０及びメモリコントローラ３３２へ送られる。メモリコントローラ３３２は、コマンド、アドレス及びデータを書き込みバッファ３３６に１トランザクション周期中（ここでは１８クロックサイクル中）記憶する。トランザクション周期中に、ディレクトリ１４０がアクセスされ、そしてアクセスの結果がＯＷＮＥＲ＿ＭＡＴＣＨライン１４０ａにアサートされる。ＯＷＮＥＲ＿ＭＡＴＣＨラインは、メモリの更新を求めるプロセッサのプロセッサＩＤが実際にデータのキャッシュラインの所有者であることをディレクトリエントリが指示する場合にアサートされる。ＯＷＮＥＲ＿ＭＡＴＣＨ信号は、遅延書き込みバッファエントリ３３６のフラグ３３６ｂをセットするのに使用される。次に続くトランザクション周期中に、メモリバスが使用できそしてフラグ３３６ｂがアサートされた場合には、メモリ３３４に記憶データが書き込まれる。本発明の１つの実施形態では、書き込み動作のみがバッファされ、到来する読み取り動作は、遅延なくメモリバスをアクセスすることが許される。遅延書きこみバッファに記憶されたビクティムデータへのその後の読み取り動作は、遅延書きこみバッファからサービスされる。

【０１３０】

図２２は、遅延書き込み動作のタイミング図である。時間Ｔ０に、読み取り０動作がＡＲＢ＿ＢＵＳに受け取られる。この読み取り動作は、ＤＲＡＭ３３４をアクセスするためにメモリへ直ちに伝播される。時間Ｔ１に、書き込み１動作がＡＲＢ＿ＢＵＳに受け取られる。このＴ１サイクル中に、ディレクトリ１４０がアクセスされ、そしてＴ１サイクルの終わりに、書き込み１アドレスの一致を示すＯＷＮＥＲ＿ＭＡＴＣＨ信号がアサートされる。その結果、遅延書き込みバッファエントリのフラグ３３６ｂがセットされる。時間Ｔ２に、読み取り２動作が受け取られ、書き込み１動作の前にメモリへ送られる。時間Ｔ３の間に、書き込み１動作に対応するフラグがアサートされた場合に、次の書き込み３動作が遅延書き込みバッファに受け取られると、書き込み１動作がＤＲＡＭ３により処理するためにメモリに送られる。

【０１３１】
ローカルメモリの読み取りについては、遅延書き込みバッファのフラグビットをセットするのにＤＴＡＧも使用できることに注意されたい。ローカルメモリからのキャッシュラインの１つをローカルノードにおけるプロセッサのキャッシュの１つに記憶することができる。プロセッサの１つがキャッシュラインをビクティム化しそしてキャッシュラインが遅延書き込みバッファに書き込まれたときに、そのキャッシュラインのＤＴＡＧエントリを検査して、キャッシュラインがプロセッサの１つに常駐したかどうか決定することができる。キャッシュラインがプロセッサの１つに常駐した場合には、ＤＴＡＧエントリの有効ビットを検査して、プロセッサがビクティム化するコピーが有効であることを確保する。ＤＴＡＧにヒットがありそしてキャッシュラインが有効であった場合には、ＤＴＡＧが遅延書き込みバッファのフラグをセットし、キャッシュラインをローカルメモリに書き込みさせる。これは、簡単なスヌーピーバスをベースとする（即ちディレクトリのない）システムがこの同じ簡単なアルゴリズムを適用できるようにする。

【０１３２】
従って、図２１のメモリ制御ロジックは、読み取り動作を読み取りサイクルにおいて直ちに実行することができそして書き込み動作を各書き込みサイクルに実行できるようにする（たとえ遅延書き込みであっても）。その結果、ディレクトリのアクセスにより遅延をこうむることなくデータの定常流がＤＲＡＭに送られ、そしてコヒレンス性を維持しながら性能が高められる。遅延書き込みバッファ技術は、ビクティム書き込み動作に関連して説明したが、メモリ性能を改善するためにコヒレンス状態が集中され且つ一定保持されるようないかなるシステムにも使用することができる。

【０１３３】
仮想チャンネル：
従って、キャッシュコヒレンスプロトコルを実施するために、プロセッサと、ディレクトリと、メモリと、ＤＴＡＧとの間に多数のメモリ参照が送信されることが明らかである。更に、各メモリ参照は、多数のトランザクション即ちホップをノード間に備え、メモリ参照のためのメッセージは、参照全体が完了する前に転送される。メッセージ間の依存性が参照を不定に阻止する場合には、マルチプロセッサシステムが停滞（デッドロック）状態となる。　上記で簡単に述べたように、本発明の１つの実施形態は、仮想チャンネル流れ制御を使用することにより、ノード間のトラフィックをマネージしそして停滞を生じることなくデータコヒレンス性を維持する。仮想チャンネルは、相互接続ネットワークに停滞のないルートを形成するために最初に導入された。本発明の１つの実施形態によれば、仮想チャンネルは、更に、共用メモリコンピュータシステムのためのキャッシュコヒレンスプロトコルにおけるリソース停滞を防止するのにも使用できる。

【０１３４】
公知の関連するキャッシュコヒレンスプロトコルでは、２つの形式の解決策が使用されている。少数のプロセッサと少数の同時保留中要求とを有するシステムの場合には、実行中の任意の点に生じ得る考えられる最大数の応答を受け入れるに足る大きさの待ち行列及びバッファが設けられている。充分な待ち行列及びバッファスペースを設けることにより、メッセージが進行のために別のメッセージに決して影響されないよう保証している。
多数の保留中要求を伴う大型のシステムでは、考えられる最大数の応答を受け入れるに足る大きさのバッファ及び待ち行列を設けることは実際的ではない。従って、停滞検出及び分析機構に接続された２チャンネル相互接続を使用して問題が解決される。第１に、相互接続部（プロセッサ及びメモリのようなシステム要素間にメッセージを移動するのに使用される論理的経路）は、２つのチャンネル、即ち要求チャンネル（又は下位チャンネル）と、応答チャンネル（又は上位チャンネル）とを使用する。これらのチャンネルは、一般に、物理的なものであり、即ち個別のバッファ及び待ち行列を使用する。第２に、潜在的な停滞を検出するために発見的手法が一般的に実施される。例えば、コントローラは、待ち行列がいっぱいでありそして待ち行列からある時間中にメッセージが出力されないときに潜在的な停滞を通知する。第３に、選択されたメッセージが否定的に確認されて、リソ

ースを解放し、他のメッセージを進行できるようにする停滞分析機構が実施される。否定的な確認メッセージは、それに対応するコマンドをリタイアさせる。

【０１３５】

上記の大型システムの解決策は、公平さ／欠乏の問題及び性能不利益の問題を含む２つの主たる問題を有している。あるメッセージが否定的に確認されるので、あるコマンドが長時間完了しない（潜在的に不定である）ことが考えられる。コマンドが所与の時間周期内に完了するよう保証されない場合には、そのコマンドを発生するリソースは、システムデータへの公平なアクセスを得ることができない。更に、リソースがシステムデータへの公平なアクセスを得ることができないために、データに対して欠乏状態となり、潜在的にシステムの停滞を生じさせる。更に、あるメッセージが否定的に確認され、従って、それらの行先に到達しないので、無効化メッセージのようなプロトコルメッセージは、それらが行先に首尾良く到達することを指示するための確認を発生しなければならない。更に、コントローラは、対応するコマンドが完了したとみなし得る前に全ての確認が受け取られるまで待機しなければならない。この非決定論的結果は、キャッシュコヒレンスプロトコルの全性能を低減するようなメッセージオーバーヘッド及び余計な待ち時間を生じさせる。

【０１３６】

本発明の１つの実施形態によれば、停滞回避に対する系統的及び決定論的解決策を採用したキャッシュコヒレンスプロトコルが使用される。潜在的な停滞を検出しそして矯正動作を行うのではなく、停滞が設計により排除される。従って、停滞検出及び分析機構の必要性がなくなる。第２に、メッセージは、停滞回避のための否定的確認ではなくなるので、「無効化」のようなプロトコルメッセージに対する確認が不要となり、それ故、帯域巾及び待ち時間が改善される。

仮想チャンネルの使用を説明する目的で、幾つかの有用な用語について最初に説明する。

依存性：メッセージＭ２が進行しない限りメッセージＭ１が進行できない場合に、メッセージＭ１はメッセージＭ２に「依存」すると定義する。更に、依存性は、移行的であるとも定義する。本発明のキャッシュコヒレンスプロトコルを実施する場合に、リソース依存性及び流れ依存性の少なくとも２種類の依存性がある。Ｍ２が待ち行列スロットのようなリソースを解放するまでＭ１が進行できない場合に、Ｍ１はＭ２に「リソース依存」すると定義する。Ｍ２が進行するまでＭ１が進行しないことをキャッシュコヒレンスプロトコルが必要とする場合には、Ｍ１はＭ２に「流れ依存」すると定義する。例えば、キャッシュコヒレンスプロトコルは、ディレクトリがある状態に達するまでＭ１が阻止状態であり、そしてディレクトリの状態を所望の値にセットするのがＭ２であることを要求する。従って、Ｍ１からＭ２へのリソース又は流れ依存性のチェーンが存在する場合に、Ｍ１はＭ２に依存すると定義する。

【０１３７】

依存性サイクル：Ｍ１の進行がＭ２の進行に依存し；Ｍ２の進行がＭ３の進行に依存し；Ｍｋ－１の進行がＭｋの進行に依存し；そして最終的に、Ｍｋの進行がＭ１の進行に依存するときに、１組のメッセージＭ１、ＭＫ（≧２）の間に「依存性サイクル」が存在すると定義する。メッセージのあるサブセットが依存性サイクルを形成するときにメッセージのシステムは停滞状態になる。Ｍ１はＭｋに依存し、Ｍｋは次いでＭ１に依存するので、サイクル内のどのメッセージも進行することができない。

ここに開示する方法及び装置は、仮想チャンネルを使用して、キャッシュコヒレンスプロトコルにおける停滞を決定論的に回避する。キャッシュコヒレンスプロトコルの設計において必要とされるハードウェア機構及び従うべき１組のルールについて説明する。

【０１３８】

１つの実施形態において、キャッシュコヒレンスプロトコルは、全てのメモリ動作がせいぜい３段階で完了すると定める。各段階において、システムの要素間に１つ以上のメッセージが転送される。それ故、各段階は、「ホップ」と称される。ホップは、０、１及び２と番号付けされる。ホップ０では、プロセッサ又はＩＯプロセッサからの要求がホームディレクトリへ送られる。ホップ１では、ホームディレクトリにより発生されたメッセージ

が１つ以上のプロセッサ又はＩＯプロセッサへ送られる。ホップ２では、メッセージが所有者プロセッサからソースプロセッサへ送られる。これらホップは、図２３に示されている。
キャッシュコヒレンスプロトコルの顕著な特性は、全ての動作が所定数のホップ内に完了することである。ここに示す実施形態では、所定数が３であるが、本発明は、選択される数が比較的小さく且つ一貫したものである限り、特定のホップ数に限定されるものではない。この特性は、停滞を検出しそして停滞を解消するためのメッセージを失敗して再トライする機構を伴わずに、全てのメッセージをそれらの行先にルート指定できることを保証するための鍵である。
【０１３９】
上記のように、ここに示す実施形態では、最大ホップ数が３である。従って、システムは、各々Ｑ０、Ｑ１及びＱ２と示された３つのチャンネルを備えている。これらのチャンネルは、システム相互接続部を通る論理的に独立したデータ経路である。これらのチャンネルは、物理的なものでもよいし、仮想のもの（或いは一部分物理的で且つ一部分仮想）でもよい。物理的なものであるときには、各チャンネルは、システム全体にわたり個別の待ち行列及びバッファリソースを有する。仮想のものであるときには、チャンネルは、待ち行列及びバッファリソースを共用し、以下に述べる制約及びルールを受ける。
３つのチャンネルは、ハイアラーキーを構成し、Ｑ０は最下位であり、Ｑ１はその次であり、そしてＱ２は最上位のチャンネルである。システムにおける停滞回避のための重要なルールは、チャンネルＱｉのメッセージが、Ｑｉより下位のチャンネルのメッセージに決して依存しないことである。
【０１４０】
更に、本発明の１つの実施形態において、ＩＯシステムからの応答メッセージと、ＩＯシステムからのメモリスペースコマンドとの間の流れ依存性サイクルを排除するためにＱＩＯチャンネルが追加される。
最後に、本発明の１つの実施形態では、ビクティムメッセージと、ビクティムメッセージが発生されるがビクティムメッセージが保留中である間に発生されるその後の依存性メッセージとに対して、Ｑ０Ｖｉｃチャンネルが使用される。
図２０ａ－２０ｈに関連して上述したように、スイッチへ発生される所与のコマンドメッセージは、一連の多数の個別トランザクションを発生する。本発明の１つの実施形態において、所与のコマンドパケットに対する各個別のトランザクションは、チャンネルに割り当てられる。チャンネルは、本質的に、所与のコマンドパケットの完了段階及び依存性を定義する順序付けされた構造体を形成する。
【０１４１】
例えば、図２３は、図２０Ａ－２０Ｊについて述べた動作の個別トランザクションにチャンネルを割り当てるところを示すフローチャートである。個別トランザクションは、次の用語で識別される。即ち、参照により生じる一連のトランザクションにおける第１トランザクションは、Ｑ０又はＱ０Ｖｉｃトランザクションと称し、一連のトランザクションにおける第２トランザクションは、Ｑ１トランザクションと称し、そして一連のトランザクションにおける第３トランザクションは、Ｑ２トランザクションと称する。
Ｑ０又はＱ０Ｖｉｃチャンネルは、まだディレクトリを訪れていないプロセッサ及びＩＯＰからの初期コマンドを搬送する。従って、Ｑ０／Ｑ０Ｖｉｃパケットの行先は、常に、ディレクトリである。Ｑ０Ｖｉｃチャンネルは、「書き込みビクティム」コマンドに対して特に指定され、一方、Ｑ０チャンネルは、プロセッサ又はＩＯＰにより開始された他の全ての形式のコマンドを搬送する。
【０１４２】
ステップ３８０で発生されるコマンドは、データを得るか又は状態を更新しようと求める。状態は、常に、データのアドレスに対応するホームディレクトリで得ることができる。ステップ３８２において、ホームディレクトリがアクセスされ、そして使用可能なキャッシュラインがホームメモリにより所有される（ディレクトリに対して）か、別のプロセッ

サにより所有されるかが決定される。いずれの場合にも、応答はＱ１チャンネルを経て発生される。ステップ３８２において、状態又はデータが第２ノードに得られると決定された場合には、ステップ３８４において、Ｑ１チャンネルの応答が第１ノードへ返送される。Ｑ１形式のトランザクションは、ＳｈｏｒｔＦｉｌｌ、ＳｈｏｒｔＦｉｌｌＭｏｄ、ＶｉｃＡｃｋ、ＣＴＤ－ＡＣＫ／ＮＡＣＫ当を含む。

【０１４３】

ステップ３８２において、ホームノードがデータを所有せず、データがダーティであって別のプロセッサにより所有されると決定された場合には、ステップ３８６において、「送信読み取り」又は「送信読み取り変更」のＱ１形式のトランザクションがＱ１チャンネルを経てリモートノードへ発生される。

ダーティへと状態変化したデータを他のノードが共用することを指示するホームノードの状態チェックに応答するか、又は「読み取り変更」に応答する場合には、ステップ３８８において、無効化Ｑ１形式トランザクションがシステムの他の当該ノードに送られる。

従って、Ｑ１チャンネルは、第２の「ホップ」におけるパケットを搬送するためのものであり、第１のホップはディレクトリに対するものである。第２の「ホップ」の行先は、常にプロセッサであり、プロセッサは、元のコマンドを開始したノードにあるか、又はシステム内の別のリモートノードにある。

【０１４４】

Ｑ２チャンネルは、「長い記入」又は「長い記入変更」トランザクションのいずれかを搬送する。Ｑ２チャンネルは、第３の「ホップ」による第３ノードからのデータを、元のコマンドを開始したノードへ返送する。

Ｑ０／Ｑ０Ｖｉｃ、Ｑ１及びＱ２形式のコマンドへのコマンドの割り当ては、ＳＭＰシステムにおいて停滞のないメッセージ送信を確保するために次のように使用できる。図２３のフローチャートは、４つの仮想チャンネル間の対話を示すが、本発明の１つの実施形態では、キャッシュコヒレンス性を維持する目的で５つの仮想チャンネルを使用することができる。その追加チャンネルは、ＱＩＯチャンネルである。一般に、ＱＩＯチャンネルは、制御状態レジスタ（ＣＳＲ）アクセスを含むＩＯアドレススペースへ全ての読み取り及び書き込みを搬送する。

【０１４５】

以下のテーブルＩＩは、チャンネル経路へのコマンドマッピングを例示するリストである。

テーブルＩＩ：

QIO	ＣＰＵへの全ＩＯスペース要求	RdByteIO, RdWordIO, WrByteIO, WrWordIO
QO	ＣＰＵ又はＩＯＰからの全メモリスペース要求	Rd, RdMod, Fetch, CTD, ITD, Vic, RdVic, RdModVic
QOVic	データを転送するＣＰＵ又はＩＯＰからの全メモリスペース要求	WrVic, Full Cache Line Write, QV_Rd, QV_RdMod, QV_Fetch
Q1	全送信コマンド	FRd, FrdMod, Ffetch
	全シャドーコマンド	SFRd, SFRdMod, SFEtch, Sinval, Ssnap
	短い記入	SFill, SfillMod
	マーカー記入の全性質	FM, FMMod, Pseudo-FM, Pseudo-DMMod, FrdMod with FM
	その他	CTD-ACK, CTD-NACK, ITD-ACK, Vic-ACK, VicRel
	IOスペース応答	IOFillMarker, IOWriteACK
	関連Consig	Invi-Ack, LoopComSig
Q2	長い記入	Fill, FillMod
	IOスペース記入	IOFill

スイッチをベースとするシステムにおける仮想チャンネルの１つの実施形態は、各チャンネルに対して物理的に個別の待ち行列、バッファ又は経路を使用することを含む。或いは又、待ち行列、バッファ又はデータ経路は、チャンネル間で共用されてもよく、従って、真の「仮想」であってもよい。本発明の１つの実施形態では、これら技術の組み合わせを使用して、ハードウェアの最適な使用がなされる。

【０１４６】

図２４には、２つ以上の仮想チャンネル間で単一バッファをいかに共用するかが示されている。バッファ４００は、多数の「スロット」を含むように示されている。各スロットは、１つのチャンネルのみにより専用に使用される。例えば、スロット４０２は、Ｑ２型コマンドに専用の多数のバッファエントリを含み、スロット４０４は、Ｑ１型コマンドに専

用の多数のバッファエントリを含み、等々となる。
残りのスロット４１０は、いずれのチャンネルについても、メッセージにより使用することができ、それ故、「共用」又は「一般的」スロットと称される。各チャンネルについてビジー信号が与えられる。ビジー信号は、バッファがそれ以上のメッセージを記憶できず、それ故、そのバッファに何も送信してはならないことを指示する。
【０１４７】
所与のチャンネルに対する所与のリソースにおいてビジー信号がアサートされるときと、そのリソースにコマンドを発生するデバイスがビジー信号に応答して発生を停止するときとの間には待ち時間周期がある。この待ち時間の間に、１つ以上のコマンドパケットがリソースへ発生されることが考えられ、それ故、リソースは、コマンドが脱落しないように設計されねばならない。
それ故、受信器がビジー流れ制御信号をアサートした後にも、M個のメッセージを受け入れることができねばならず、但し、Mは、次の式III で定められる。
式III ：
Ｍ＝（フレームクロックでの流れ制御待ち時間）／（フレームクロックでのパケット長さ）
「M」の値は、ここでは、チャンネル当たりに得られる専用スロットの数を定義する。
【０１４８】
図２５には、各チャンネルごとに個別のリソースを使用して仮想チャンネルが実施される例が示されている。２つのノード４２０及び４２４の部分は、ハイアラーキースイッチ（ＨＳ）４２２を経て互いに接続されて示されている。
グローバルポート４２０は、バス４２１ａを経てスイッチ４２２から入力データを受け取り、そしてバス４２１ｂを経てスイッチ４２２にデータを送信するように接続される。同様に、グローバルポート４２４は、バス４２３ａを経てスイッチ４２２にデータを送信し、そしてバス４２３ｂを経てスイッチ４２２からデータを受け取るように接続される。
データバス４２１ａ、４２１ｂ、４２３ａ及び４２３ｂの各々は、全ての形式のチャンネルコマンドを送信又は受信する。待ち行列機構４２５のような待ち行列機構は、各リソースの各入力及び出力端子に設けられる。この待ち行列機構は、多数の個々に制御されるバッファ４２５ａ－４２５ｅを備え、各バッファは、１つの形式のチャンネルコマンドのみを専用に記憶する。バッファ４２５ａは、Ｑ０チャンネルコマンドのみを記憶し、バッファ４２５ｂは、Ｑ０Ｖｉｃチャンネルコマンドのみを記憶し、等々となる。
【０１４９】
コマンドパケットが各リソースインターフェイスに受け取られるときに、コマンドの形式がパーズされ、そしてパケットは、適当なバッファへ送られる。コマンドパケットがノードの適当なプロセッサ又はＩＯＰへ送られる準備ができると、それらが適当なバッファから選択され、そしてＡＲＢバス及びＱＳＡ（図６）を経て送られる。各チャンネルごとに１つづつ、５つのサーチエンジンがあり、各チャンネルに対して次のメッセージを探索する。
上記機構においては、各チャンネルが独立して流れ制御され、そしてシステム全体にわたりハイアラーキーの最下位チャンネル以外の各チャンネルにスロットが指定される。これは、チャンネルがリソース依存性により下位チャンネルによって決して阻止されないことを保証する。上位チャンネルメッセージの移動は、下位チャンネルメッセージによるリソースの占有により阻止されない。
【０１５０】
仮想チャンネル間で物理的バッファを共用する上記機構は、簡単なものである。より精巧な機構については、ハイアラーキースイッチに関して最初に述べた。
仮想チャンネル：裁定及びコヒレンスプロトコル設計のルール
コヒレンスプロトコルにおいて停滞のないメッセージ送信を保証するためにはハードウェア機構のみでは不充分である。というのは、問題のリソース依存性の部分しか対処しないからである。全てのリソース及び流れ依存性サイクルを排除するために、多数の付加的な

裁定及びコヒレンスプロトコル設計ルールが適用される。
第１に、メッセージの進行は、下位チャンネルメッセージの進行に依存してはならず、この場合に、Ｑ２は上位チャンネルであり、そしてＱ０は下位チャンネルである。アービターは、各チャンネルの流れ制御を互いに独立して維持しなければならない。例えば、ビジーの流れ制御信号がＱ１に対してアサートされるが、Ｑ２に対してはアサートされない場合には、アービターは、Ｑ２メッセージを進行させねばならない。保留中のコマンドパケットに対してリソースをサーチするのに使用される全てのサーチエンジンは、同じ特性をサポートしなければならない。
【０１５１】
第２に、２つ以上のチャンネル間に共用されるいかなるリソースも、下位のチャンネルが阻止された場合に上位のチャンネルが進行できるようにするために、上位のチャンネルの各々に対してある専用のスロットを含まねばならない。
第３に、全てのチャンネルコマンドは、一貫して作用しなければならない。Ｑ０コマンドの終了点は、常に、ディレクトリである。Ｑ１コマンド及びＱ２コマンドの終了点は、常に、プロセッサである。終了点において、トランザクションを継続するために、それらを上位チャンネルへ移動しなければならない。例えば、Ｑ０メッセージがディレクトリに到達したときには、Ｑ０メッセージを発生することができず、Ｑ１又はＱ２メッセージを発生しなければならない。それ故、メッセージは、下位チャンネルメッセージへと分岐又は変換することはできない。
【０１５２】
他の点において分岐するトランザクションの場合には、同じか又は上位のチャンネルのメッセージしか形成できない。例えば、「送信読み取り変更」（Ｑ１メッセージ）がハイアラーキースイッチにおいて「送信読み取り変更」、「無効化」及び「記入変更マーカー」を形成するときには、これら全てのメッセージがＱ１メッセージとなる。
従って、バスをベースとするシステム又はスイッチをベースとするシステムのいずれかに仮想チャンネルを設ける装置及び方法が提供される。仮想チャンネル及び上記の順序付け制約を使用することにより、参照は、ディレクトリによっていったんサービスされると完了することが保証される。その結果、ＮＡＣＫ（１つのプロセッサが別のプロセッサにプロセスが完了しないことを指示する）及びリタイアを必要とする公知の複雑なプロトコルは排除される。
【０１５３】
５つまでの独立したチャンネルを伴う実施形態を示したが、本発明の１つの実施形態は、所与の数のチャンネルに限定されず又は対称的なマルチプロセッサシステムに限定されないことを理解されたい。むしろ、選択されるチャンネルの数は、各チャンネルに固有の制御及びハードウェアオーバーヘッドが与えられると、コヒレントな通信をサポートするに必要な数でなければならない。従って、仮想チャンネル制御方法及び装置は、マルチプロセッサシステムにおいて高性能の、停滞のない通信を行えるようにする。
コヒレンス性を維持するためのディレクトリの動作：
以上に、基本的な通信構成を説明し、そしてＳＭＰのノード間に通信が自由に流れるようにするための基本的な制御構造体が提供された。しかしながら、コヒレンス性のための鍵は、自由に流れるコマンドがシステム内の各プロセッサにより正しい順序で「取り扱われる」ように確保することである。ＳＭＰシステム内の全てのコマンドに対しシリアル化ポイントを与える機構は、各ノードにおけるディレクトリである。
【０１５４】
上述したように、全てのＱ０形式コマンドは、先ず、関連メモリアドレスのホームディレクトリをアクセスする。いずれのコマンドに対してもホームディレクトリが最初にアクセスされるよう確保することにより各コマンドを共通のソースから正しい順序で検討することができる。
本発明の１つの実施形態では、シリアル化順序は、アドレスＸに対するディレクトリからの裁定に勝った後にＸに対するＱ０コマンドがＡＲＢバスに現れるという順序である。「

ロード」形式のコマンドは、それに対応する読み取りコマンドがホームディレクトリにアクセスしたときに順序付けされる。「記憶」形式のコマンドは、それに対応する「読み取り変更」コマンドがディレクトリにアクセスするか又はそれに対応する「クリーン－ダーティ」コマンドがディレクトリにアクセスしてＡＲＢバスに現れるときに順序付けされる。

【０１５５】

例えば、１０個のコマンドの以下のシーケンスが種々のプロセッサ（Ｐ＃）により共通のホームディレクトリへ発生されると仮定する。但し、X_i は、キャッシュラインＸの一部分である。

テーブルIV：

１	Ｐ１：記憶	X_1 （１）
２	Ｐ２：ロード	X_1
３	Ｐ３：ロード	X_1
４	Ｐ５：ロード	X_1
５	Ｐ１：記憶	X_2 （２）
６	Ｐ２：記憶	X_1 （３）
７	Ｐ４：ロード	X_1
８	Ｐ５：ロード	X_2
９	Ｐ６：ロード	X_1
１０	Ｐ２：記憶	X_1 （４）

キャッシュラインのバージョンは、各記憶動作の結果として更新される。従って、コマンド１はバージョン１を形成し、コマンド５はバージョン２を形成し、コマンド６はバージョン３を形成し、そしてコマンド１０はバージョン４を形成する。

【０１５６】

シリアル化順序は、ディレクトリに到達する事象の各シーケンスがキャッシュラインＸの正しいバージョンを得るように確保する。例えば、コマンド２ないし４は、バージョン１を得なければならない。プロセッサＰ１のコマンド５が記憶を行うときには、全てのバージョン１キャッシュライン（プロセッサＰ２、Ｐ３及びＰ５における）に「無効化」を送信しなければならない。同様に、プロセッサＰ２のコマンド６がバージョン３データでＸを更新するときには、プロセッサＰ１のバージョン２データを無効化しなければならない。プロセッサＰ４、Ｐ６及びＰ７は、バージョン３データを得るが、これは、プロセッサＰ８のバージョン４データの記憶により後で無効化される。

共通のアドレスキャッシュラインＸに対する多数のロード及び記憶動作は、システムにおいていかなる所与の時間にも進行し得ることを述べれば充分であろう。システムは、ロード及び記憶がディレクトリによりシリアル化順序で処理されるようにこれらのコマンドを処理する。

【０１５７】

システムがシリアル化順序を維持しそして付随的にデータのコヒレンス性を維持するのを助けるために多数の技術が使用される。これらの技術は、Ｑ１チャンネルコマンドの厳密な順序付け、ＣＴＤ明瞭化、「シャドーコマンド」、「マーカー記入」及び「遅延ビクティム書き込みバッファ動作」を含む。各技術について、以下に詳細に説明する。

【０１５８】

Ｑ１チャンネル順序付け：

コヒレンス性を維持するのに使用される第１の方法は、Ｑ１チャンネル上を進行する全てのメッセージ、即ちディレクトリから送られる全てのメッセージが、先入れ先出し順序で進むように確保することである。即ち、ディレクトリから別のプロセッサ又はＩＯＰへ送られるＱ１型メッセージは、コマンドがディレクトリにおいてシリアル化された順序に基づいて送られる。

例えば、図２６のサブシステムの例では、ノード４３０における第１プロセッサＰ１（４３１）がキャッシュラインＸをそのキャッシュ「ダーティ」に記憶すると仮定する。ノー

ド４３２におけるプロセッサＰ１６（４３３）は、Ｑ０チャンネルに「Ｘ読み取り(Read X)」を発生し、これは、ノード４３６におけるＸのホームディレクトリ４３７へ送られる。又、ノード４３２におけるプロセッサＰ１７は、Ｑ０チャンネルに「無効－ダーティ」コマンドを発生し、これも、ノード４３６におけるＸのホームディレクトリ４３７へ送られる。「Ｘ読み取り」の受信に応答して、ディレクトリエントリに基づき、「送信Ｘ読み取り(Forwarded Read X)」がＱ１チャンネルを経てプロセッサＰ１（４３１）へ送られる。ＩＴＤの受信に応答して、ディレクトリエントリの状態に基づき、「無効化」がハイアラーキースイッチ４３５へ送られ、これは、Ｑ１チャンネルを経てプロセッサＰ１及びプロセッサＰ１６へ「無効化」を送る。従って、同じ時点で、「Ｘ無効化」及び「Ｘ読み取り供給」がＱ１チャンネルコマンドとしてＰ１へ送られる。

【０１５９】

Ｑ１チャンネルのコマンドが順序ずれして実行することが許された場合には、「読み取り」の前に「無効化」が生じることがある。その結果、「読み取り」のための記入データがプロセッサＰ１６に送られないことになり、それ移行の動作の結果が予想し得ないものとなる。

しかしながら、チャンネルＱ１のコマンドを正しい順序で保つことにより「読み取り」は「無効化」を受け取る前に処理され、コヒレンス性が維持される。

【０１６０】

本発明の１つの実施形態では、チャンネルＱ１についてのみＦＩＦＯ順序が維持され、ＦＩＦＯ順序とは、同じメモリアドレスに対応する全てのメッセージがＦＩＦＯ順序に留まることを意味する。しかしながら、本発明は、Ｑ１チャンネルに対する順序を維持することのみに限定されるものではなく、チャンネルのいかなる組み合わせに対する順序の維持も含むように拡張することができる。

上記の順序付け手順を実施する１つの方法は、ＱＳＡチップ（図６）のＱＳＡＲＢ１１により実行される。ＱＳ　ＡＲＢは、全てのＱ０トランザクションをノードのホームメモリスペースに対してシリアル化する。その結果、Ｑ１パケットのシリアル流が発生されて、ノードのローカルプロセッサと、グローバルポート及びハイアラーキースイッチを経てノードから離れたプロセッサとの両方に向けられる。

第１の順序付けルールを次に説明する。所与のＱＳ　ＡＲＢにより発生される全てのＱ１パケットは、シリアルな順序で発生される。所与のＱＳ　ＡＲＢからの幾つかの又は全てのＱ１パケットがターゲットとする全てのプロセッサは、これらのＱ１パケットを、それらがＱＳ　ＡＲＢにより発生された順序で見る。

【０１６１】

このルールをサポートするために、ＱＳＡチップは、ノード内の接続されたプロセッサとやり取りされる全てのＱ１パケットに順序を維持する。グローバルポートのロジックは、ハイアラーキースイッチとＱＳＡチップとの間に転送される全てのパケットにＦＩＦＯ順序を維持する。更に、ハイアラーキースイッチは、所与の入力から所与の出力へ送られる全てのＱ１パケットにも順序を維持する。

このルールは、１つのＱＳ　ＡＲＢからのＱ１パケットと、別のノードのＱＳＡＲＢからのＱ１パケットとの間に特定の順序を命令するものではないことに注意されたい。他のノードから受け取られたＱ１パケットは、ハイアラーキースイッチを経てホームノードにより発生されたＱ１パケットと次のようにシリアル化される。リモートノードのプロセッサをターゲットとする全てのＱ１パケットは、リモートノードのＱＳ　ＡＲＢにより処理される。これらのＱ１パケットは、ハイアラーキースイッチによりリモートノードで発生されたＱ１パケットとシリアル化される。所与のＱＳ　ＡＲＢからのＱ１パケットの全ての受信者は、Ｑ１パケットを、それらがＱＳ　ＡＲＢにおいてシリアル化されたのと同じ順序で見なければならない。

【０１６２】

図２７は、多数のＱ０及びＱ１コマンドの順序付けが上記の順序付けガイドラインに基づいてＳＭＰを通して処理されるところを示すブロック図である。ノード４４０のプロセッ

サＰｘはコマンドＱ０ａを発生し、プロセッサＰｙはコマンドＱ０ｂを発生し、そしてプロセッサＰｚはコマンドＱ０ｃを発生すると仮定する。同じ時間中に、ＱＳ　ＡＲＢ４４１は、プロセッサＰｒ及びＰｑからのＱ１メッセージをグローバルポート４４３から受け取る。

これらのメッセージは、次のように順序付けされる。ＱＳ　ＡＲＢ４４１は、Ｑ０ａ、Ｑ０ｂ及びＱ０ｃを処理して、Ｑ１ａ、Ｑ１ｂ及びＱ１ｃ応答を発生する。これらの発生されたＱ１コマンドは、到来するＱ１コマンドと合成されて、コマンドの順序付けされた流れをＦＩＦＯ４４２へ供給し、ローカルプロセッサへと送る。ＦＩＦＯコマンドの順序は、ＱＳ　ＡＲＢにより処理されたコマンドの順序を反映する。

Ｑ１ａ、Ｑ１ｂ及びＱ１ｃコマンドは、グローバルポート４４３へ送られ、リモートノードへ送信される。グローバルポートの出力バッファ４４４は、これらのコマンドを、それらがＱＳ　ＡＲＢにより処理されたのと同じ順序で記憶する。この順序は、図１４－１９について上述した方法を用いてメッセージがリモートＣＰＵ４５４へ送られるときにハイアラーキースイッチ４４６により維持される。

【０１６３】

図２７Ａは、ハイアラーキースイッチにおいて従う別の順序付けガイドラインを示す。上述したように、ハイアラーキースイッチは、ハイアラーキースイッチの所与の入力ポートに現れてハイアラーキースイッチの共通の出力ポートをターゲットとする多数のパケットが、それらが入力ポートに現れたのと同じ順序で出力ポートに現れるよう確保することにより、順序を維持する。

【０１６４】

図２７Ｂを参照すれば、上述したように、ハイアラーキースイッチは、入力メッセージをマルチキャスティングする役目も果たし、即ち受け取った１つのＱ１パケットを２つ以上の行先ノードに送信するという役目も果たす。スイッチによりマルチキャスティングされるパケットの一例は、無効化パケットである。ハイアラーキースイッチの異なるポートから入力された多数のパケットが共通の出力ポートにマルチキャスティングされるときには、Ｑ１パケットは、全ての出力ポートにおいて同じ順序で現れねばならない。例えば、パケット１及びパケット２の両方がハイアラーキースイッチ４６０に受け取られる場合に、２つのメッセージをプロセッサ４６４及び４６６にマルチキャスティングする１つの許された方法は、上記のように、メッセージ２がメッセージ１の前に両プロセッサに到着するようにすることである。別の許された方法は、メッセージ１のパケットがメッセージ２のパケットの前に両プロセッサに到着するようにすることである。しかしながら、２つのプロセッサは、２つのパケットを異なる順序で受け取ってはならない。

【０１６５】

ハイアラーキースイッチが従わねばならない別の順序付けルールは、多数の入力ポートからのＱ１パケットの順序付けされたリストが共通の出力ポートをターゲットとするときに、Ｑ１パケットが、全ての到来するＱ１パケットの１つの共通の順序付けに合致する仕方で出力ポートに現れるように確保することである。

例えば、図２７Ｃにおいて、入力ポート４６１には、パケット２がパケット４の前に受け取られる。同様に、入力ポート４６２には、パケット１がパケット３の前に受け取られる。停滞を防止するには、これら命令の全体的な順序を遵守しなければならない。出力パケットを与える１つの許された方法は、パケット３を最初にノード４６４に送信し、そしてパケット１を最初にノード４６６に送信することである。この送信が図２７Ｃに示されている。別の許された出力は、パケット２及び４を受信者のプロセッサにより最初に受け取ることである。しかしながら、１つのプロセッサがパケット３を最初に受け取りそして別のプロセッサがパケット４を最初に受け取る場合には、プロセッサがそれらの元のシーケンスの他のパケットの受信を待機してストールするので停滞が生じ得る。

それ故、Ｑ１チャンネルにおいて順序が維持されるよう確保するルールが設けられる。本発明の１つの実施形態では、性能の理由で、Ｑ０及びＱ２チャンネルパケットを順序ずれして処理するのが望ましい。データの一貫性を確保するために、多数のコヒレンス性機構

が以下に述べるように設けられる。

【０１６６】

ダーティへの変更の明瞭化

上述したように、Ｑ１形式のコマンドのみが、ディレクトリに定義されたシリアル化順序で維持される。本発明の１つの実施形態では、Ｑ０及びＱ２コマンドは順序付けされない。従って、受け取られるＱ０及びＱ２コマンドの相対的なタイミングの結果としてディレクトリにコヒレンス性の問題が生じないように予防策がとられる。

発生する１つのコヒレンス性の問題は、ディレクトリエントリの構造によるものである。図９に示すように、各ディレクトリエントリは、所有権フィールドと、各ノードに対して１つの存在ビットとを含む。存在ビットは、関連ノードの４つのプロセッサの１つにデータが存在することを示すおおよそのベクトルである。４つのプロセッサのいずれかが動作すると、存在ビットがセットされる。従って、ノードのどのプロセッサが存在ビットをセットしたかに関してある種の曖昧さが生じる。この曖昧さは、ある場合にコヒレンス性の問題を引き起こす。

【０１６７】

例えば、図２８Ａ及び２８Ｂは、２つのノード４７０及び４７２のブロック図である。ノード４７０［グローバルシステムのノードＩＤ３］は、プロセッサＰ１２、Ｐ１３、Ｐ１４及びＰ１５を備え、一方、ノード４７２［グローバルシステムのノードＩＤ７］は、ノードＰ２８、Ｐ２９、Ｐ３０及びＰ３１を含む。

【０１６８】

時間Ｔ０－Ｔ３の種々の一連の周期における所与のキャッシュラインＸのディレクトリエントリの状態は、図２８Ｂにおいてディレクトリ状態テーブル４５５に示されている。この例では、キャッシュラインＸのホームノードは、ノード４７０又は４７２以外のノードである。

時間Ｔ０において、キャッシュラインＸの所有者は、所有者ＩＤ８０で示すようにメモリである。更に、時間Ｔ０において、ノードＩＤ７のプロセッサ３０は、キャッシュラインＸのクリーンなコピーを記憶する。

時間Ｔ１において、プロセッサ１４は、「記憶」コマンドを送信し、これは、「読み取りブロック変更Ｘ」に変換され、そしてキャッシュラインＸのホームディレクトリへ送られる。メモリが所有者であるから、プロセッサＰ１４は、メモリからデータを得ることができ、そしてキャッシュラインの所有者となる。キャッシュラインＸの古いバージョンを無効化するためにノード７に無効化が送信され、そしてノード７の存在ビットがクリアされる。更に、プロセッサＰ１４は、そのノード存在ビット４５６（ビット３）をセットする。キャッシュラインＸは、変更及び記憶のためにホームメモリからプロセッサＰ１４へ送られる。

【０１６９】

時間Ｔ２に、プロセッサ３１のような別のプロセッサが、キャッシュラインＸの「読み取り」を発生する。この「読み取り」は、プロセッサＰ１４から「記入」を経てデータを得る。従って、時間Ｔ２に、ディレクトリは、ノードＩＤ３（プロセッサＰ１４）及びノードＩＤ７（プロセッサＰ３１）の両方が、ノード存在ビット４５８及び４５６で示すように、キャッシュラインＸのコピーを記憶することを指示する。

時間Ｔ３に、プロセッサ３０によりＣＴＤが発生される場合には、システムの異なるプロセッサから見たキャッシュラインＸの状態は、次の理由でインコヒレントとなる。ＣＴＤがディレクトリに到達すると、Ｘのディレクトリエントリを読み取り、そしてそのノード、即ちノードＩＤ７の存在ビット４５８が既にオンであるかどうか決定する。その結果、プロセッサ３０は、次いで、ＣＴＤ要求において成功したと仮定する。プロセッサ３０は、キャッシュラインＸのプロセッサ１４のコピーを無効化し、そしてディレクトリの所有者フィールドを更新する。この動作は、予想し得ない結果を招くことがある。というのは、プロセッサＰ１４がプロセッサＰ３０よりも最新のデータバージョンを記憶するからである。

【０１７０】
１つの問題は、プロセッサ３０がプロセッサ１４により形成されたキャッシュラインの古いバージョンをまだ記憶しており、そしてプロセッサ１４がデータの最新のバージョンを無効化するように通知したことである。このような状態は、ＳＭＰシステムで重大なコヒレンスの問題を生じさせる。
上記問題を解消するのに使用できる幾つかの方法がある。その１つの方法は、システムの各プロセッサごとに１ビットを与えるようにディレクトリエントリの存在ビット拡張することである。従って、分解能がノードレベルからプロセッサレベルへ変更される。しかしながら、この解決策は、不都合なことに、ディレクトリのサイズを増大する。
【０１７１】
本発明の１つの実施形態は、同じアドレスへの保留中参照がそのノードに対してトランシット状態にあるときにＣＴＤコマンドを低速化することにより上記曖昧さの問題を防止するより簡単な方法を提供する。同じアドレスに対して保留中の要求がある場合には、その以前の要求がリタイアするまでＣＴＤが保持される。所与のノードのトランザクション追跡テーブル（ＴＴＴ）（図１０）を使用して、そのノードに対する保留中のグローバル参照を監視する。更に、ＣＴＤがＴＴＴに受け取られた後に受け取った要求は、失敗となる。
図１０を参照して述べたように、ＴＴＴは、完全に連想式の多機能制御構造体である。ＴＴＴは、２つの一般的なタスクを実行する。これは、その関連ノードにより発生された全てのリモート参照のアドレスを記憶する。従って、ＴＴＴは、そのトランザクションが完了したとみなされるまで、ノードにより発生された各リモートアクセスに対して１つの情報エントリを記憶する。更に、ＴＴＴは、ローカルアドレスの要求に応答して、過渡的なコヒレンス状態に関してコヒレンス情報を与える。従って、ＴＴＴは、アクセスがトランシット状態にある間にその状態を追跡するためのテーブルである。
他の処理システムは、いかなる瞬間にも所与のキャッシュラインへの１つの参照をトランシット状態にすることができる。トランシット状態にあるキャッシュラインへのその後の参照は、トランシット状態の参照が完了するまで阻止される。
【０１７２】
これに対し、ディレクトリにおけるコマンドのシリアル化と、チャンネル順序付けルールとにより、本発明のＳＭＰは、同じキャッシュラインへの多数の参照を所与の瞬間に進行させることができる。その結果、ＳＭＰの全性能が改善される。
ＴＴＴ５２２は、ＱＳＡチップ５３５のロジックにより、グローバルポートに発生されたトランザクションの状態を決定するのに使用される。グローバルポートへ応答を発生する前に、ＱＳＡは、先ず、ＴＴＴにアクセスして、同じキャッシュラインへのどんな参照が保留中であるかを決定する。参照は、最後に受け取ったトランザクションに応答してＴＴＴからリタイアしていない場合には保留中である。
【０１７３】
参照がＴＴＴからいかにリタイアするかは、コマンドフィールド５８４に示された参照の形式に依存する。例えば、ＴＴＴに記憶するためにグローバルポートへ送られる「Ｘ読み取り」参照は、「ここに記入」５８８a 及び「マーカーをここに記入」５８８b の両方の状態ビットを受け取ることを必要とする。（「マーカーの記入」は、いかに詳細に述べる。）ＣＴＤ又はＩＴＤのような状態型の参照の場合に、ＴＴＴにおいてＡＣＫ／ＮＡＣＫビット５８８c をセットすれば、そのエントリをリタイアするのに充分である。
【０１７４】
図２９は、ＴＴＴを使用して曖昧なディレクトリエントリを排除するところを示すフローチャートである。ステップ５００において、キャッシュラインＸは、そのホームノードのメモリに記憶され、そしてノード７のプロセッサ３０は、データのコピーを記憶する。ステップ５０２において、「ＲｅａｄＭｏｄＸ」がプロセッサ１４により発生される。その結果、無効化がノード７に送られる。ステップ５０４において、プロセッサＰ３１は、「ＲｅａｄＸ」を発生し、これは、ノード７のＴＴＴのエントリを次の状態で形成する。

【０１７５】

アドレス	コマンドＩＤ	状態			
		Fill	Fmark	Shadow	ACK/NACK
X	Read 31		-	-	-

ステップ５０６において、プロセッサＰ３０は、ＣＴＤＸを発生する。ＱＳＡチップは、ＣＴＤ命令のアドレスを検査し、それがリモートＣＴＤであることを決定し、そしてＴＴＴへのＧＰリンクを経てグローバルポートへ送信する。ＴＴＴの内容は、以下に示す通りである。

アドレス	コマンドＩＤ	状態			
		Fill	Fmark	Shadow	ACK/NACK
X	Read 30		-	-	-
X	Read 31		-	-	-

図６について述べたように、グローバルポートは、ＴＴＴからの情報を使用して、どのコマンドをハイアラーキースイッチから送出することが許されたかを決定する。本発明の１つの実施形態では、保留中の「読み取り」がトランシット状態にあるとＴＴＴが決定した場合に、グローバルポートは、「読み取り」結果が返送されるまでＣＴＤをスイッチへ送ることが防止される。

【０１７６】

図２９のフローチャートに示す例では、アドレスＸへの保留中の読み取り要求は、ＴＴＴにより識別される。その結果、ステップ５０８において、ＣＴＤは、「読み取り」がもはや保留中でなくなるまで、オフに保たれる。

「読み取り」は、「記入」及び「マーカー記入」の両方がノード７に返送されるまで保留となる。この時間中に、ステップ５０２においてＲｅａｄＭｏｄにより発生された無効化がノード７に到達し、各ノードのＤＴＡＧＳを更新する。Ｘの無効化がＴＴＴに到達すると、ＴＴＴは、ＴＴＴに保持されたＣＴＤを失敗と表示し、これは直ちに解除される。ステップ５１０において、ＣＴＤが依然ＴＴＴにある場合には、グローバルポートを経て送信される。

従って、ＴＴＴを使用して、ＣＴＤコマンドを適当にオフに保持し又は失敗状態とすることにより、ディレクトリの存在ビットの曖昧さにより生じるコヒレンスの問題を排除することができる。

【０１７７】

マーカーの記入：

プロセッサに対するほとんどの応答は、Ｑ１チャンネルにおけるものであり、従って、上記のルールによれば、正しい順序が維持される。しかしながら、Ｑ２チャンネルで受け取られたメッセージは、この順序制約を受けない。Ｑ２型のメッセージは、「記入」及び「記入変更」を含む。

Ｑ２型メッセージの到着は、ディレクトリにおいて明らかなように、シリアル化順序を表わさないので、返送データに潜在的な曖昧さが生じる。例えば、「無効化」がＱ１を進行し、そして「記入変更」がＱ２を進行するので、コヒレンス性を維持するためにどの動作が順序において最初に生じるべきかを決定する何らかの方法がなければならない。

【０１７８】

例えば、図３０を参照すれば、２つのノード５２０及び５３２が示されている。説明上必要なノードの部分しか示されていない。プロセッサＰ２（５２４）及びプロセッサＰ４（５３４）がキャッシュラインＸのコピーを記憶すると仮定する。キャッシュラインＸのホームノードは、ノード５３２である。

以下の説明において、次のパケットにより使用されるチャンネルは、異なる線を用いて指

示される。Ｑ０コマンドは、単一線矢印で指示され、Ｑ１コマンドは、二重線矢印で指示され、そしてＱ２コマンドは、破線矢印で指示される。
プロセッサＰ４がキャッシュラインＸの独占的所有権を得るためにＣＴＤ　Ｘを発生すると仮定する。これに応答して、ディレクトリ存在ビット及びＤＴＡＧ（図示せず）により、ディレクトリ５４２は、ノード５２０へ無効化を発生する。この無効化は、Ｑ１チャンネルを経てノード５２０のＤＴＡＧＳを更新し、そしてコピーを有する全てのプロセッサ（ここではプロセッサＰ２）に無効化調査を送信する。
【０１７９】
次いで、プロセッサＰ１は、Ｘのホームディレクトリ５４２へＲｅａｄＭｏｄＸを発生する。上記のように、Ｘは、現在プロセッサＰ４により所有され、それ故、コヒレンスプロコルによれば、ＦｏｒｗａｒｄｅｄＲｅａｄＭｏｄ　Ｘがプロセッサ４へ送られる。プロセッサＰ４は、それに応答して、Ｑ２チャンネルを経てプロセッサＰ１へＦｉｌｌＭｏｄを発生する。
Ｑ２チャンネルの通信は、Ｑ１の通信とシリアル化されないので、ＣＴＤ　Ｘからの「無効化」がノード５２０に到達する前にＱ２のＦｉｌｌＭｏｄがプロセッサＰ１に到達する可能性が存在する。その結果、Ｐ１のキャッシュには有効データが書きこまれるが、そのすぐ後で、ＤＴＡＧＳがノードにおけるＸのコピーを無効化するようにセットされ、そしてＰ２及びＰ１に「無効化」が送られる。しかしながら、「無効化」は、Ｐ２のバージョンのみに対応し、Ｐ１におけるバージョンには対応しない。ここで、システムは、インコヒレントな状態となる。ディレクトリ５４４は、Ｐ１を所有者として記録するが、Ｐ１はまだ無効化されている。
【０１８０】
本発明の１つの実施形態は、各ノードのグローバルポートに「マーカー記入」及びトランザクション追跡テーブル（図１０）を使用することによりこの問題を克服する。
「マーカー記入(Fill Marker)」又は「マーカー記入変更(Fill Marker Mod)」は、ホームノードのメモリに現在記憶されていないデータに対する「読み取り」又は「読み取り変更」要求に応答して発生されるパケットである。即ち、「マーカー記入」又は「マーカー記入変更」は、「送信読み取り(Forwarded Read)」又は「送信読み取り変更(Forwarded Read Mod)」と同時に発生される。従って、「マーカー記入」及び「マーカー記入変更」は、Ｑ１チャンネルコマンドである。「送信読み取り」又は「送信読み取り変更」コマンドは、キャッシュラインを記憶するプロセッサに送られるが、「マーカー記入」又は「マーカー記入変更」の行先は、元の「読み取り」又は「読み取り変更」を供給したプロセッサである。
【０１８１】
「マーカー記入」は、発生元プロセッサが、ディレクトリに生じるシリアル化順序を決定できるようにする。図３１を参照すれば、「マーカー記入」の適用は、上記問題を次のように矯正する。前記したように、プロセッサ５３４がＸのＣＴＤをＸのホームディレクトリに発生し、その結果、「無効化」５５０がＱ１チャンネルを経てノード５２０へ送られると仮定する。プロセッサＰ１（５２２）がＲｅａｄ Ｍｏｄ Ｘをリモートディレクトリに発生すると、その要求に対してＴＴＴエントリが発生される。この要求に対するＴＴＴテーブルエントリの例が図３２に示されている。ＴＴＴテーブルエントリは、「ここに記入」及び「マーカーをここに記入」状態ビットを含むことに注意されたい。これらビットの各々は、ノード５２０のグローバルポートに各パケットが受け取られるのに応答してセットされる。ＴＴＴエントリは、「記入」及び「マーカー記入」の両方が返送されるまでクリアされない。
【０１８２】
図３１に戻ると、上述したように、プロセッサ５２２からのＲｅａｄ ＭｏｄＸは、プロセッサ５３４へのＦＲｄＭｏｄＸを生じる。同時に、チャンネルＱ１を経て、Ｆｉｌｌ Ｍａｒｋｅｒ Ｍｏｄ Ｘ５５２がプロセッサＰ１に返送される。「無効化」及びＦｉｌｌ Ｍｏｄ Ｍａｒｋｅｒの両方が同じＱ１チャンネルに送られる。チャンネルＱ２のＦｉ

ｌｌ Ｍｏｄ５５４は、「無効化」の前にノード５２０に到着すると仮定する。グローバル参照の「タグ複製」状態は、Ｆｉｌｌ Ｍｏｄ又はＦｉｌｌ Ｍｏｄ Ｍａｒｋｅｒの返送に応答して更新される。従って、Ｆｉｌｌ Ｍｏｄは、Ｘの所有権をプロセッサＰ１として表わすようにＸのＤＴＡＧ状態を更新させる。

【０１８３】

「無効化」５５０が、ノード５２０に到達する次の命令であると仮定する。ＴＴＴは、「送信読み取り」命令の状態を決定するためにアクセスされる。この点において、ＴＴＴエントリは、「ここに記入」ビットをセットするが、「マーカーをここに記入」ビットはセットされない。従って、ＴＴＴは、無効化及びリモート読み取り動作の相対的なタイミングに関する指示を与える。Ｑ１コマンドのシリアル化のために、無効化は、プロセッサ５２２からのＲｄＭｏｄＸよりも早い時間にディレクトリ５４２に発生されたと推測でき、従って、Ｆｉｌｌ Ｍｏｄが新しいバージョンであり、プロセッサ５２２のデータコピーには無効化が適用されない。その結果、プロセッサＰ１のＤＴＡＧエントリは、無効化されない。

【０１８４】

上記実施形態は、ＴＴＴをグローバルポートに存在するものとして示したが、別の実施形態によれば、各ノードの各プロセッサは、ディレクトリへの要求を監視することにより共通のアドレスへのリモート要求の状態を追跡することができる。従って、「マーカー記入」は、単にＴＴＴへ送られるのではなく、ディレクトリにより関連プロセッサへ送られる。

従って、ＴＴＴは、２つの目的を果たすことが明らかである。マルチプロセッサノードから送出されたコマンドの形式を監視することにより、ＴＴＴは、同じアドレスへの他のコマンドが完了するまで、あるコマンド（ＣＴＤのような）の送信を禁止することができる。更に、要求がＱ２チャンネル（「マーカー記入」のような）へ移行したときにＴＴＴに指示する表示機構を設けることにより、ＴＴＴを用いて、異なるチャンネルに返送されるコマンド（即ち、Ｑ２記入及びＱ１コマンド）間の相対的なタイミング指示を与えることができ、従って、メモリを崩壊することのあるコマンドがプロセッサへ送られるのを防止することができる。

【０１８５】

シャドーコマンド

上記説明から明らかなように、ローカルアクセスは、通常、リモートアクセスよりも相当に早い。従って、性能に関しては、ローカル及びリモートアクセスの両方がＳＭＰシステムにおいて同時に生じることが許される。

しかしながら、ローカルアクセスの発生によりリモートアクセスに対して停滞の問題を生じさせる幾つかの場合がある。例えば、図３３Ａを参照すれば、１つのプロセッサ５６２がキャッシュラインＸにＲｄＸを発生すると仮定する。キャッシュラインＸのホームノードは、ノード５６０である。ノード５６０のディレクトリは、プロセッサ５８２がキャッシュラインを現在所有することを指示する。従って、Ｆｏｒｗａｒｄｅｄ Ｒｅａｄ Ｘが５８２に送られる。

その後、ノード５６０のプロセッサ５６４がＣＴＤ Ｘを発生すると仮定する。上記のように、キャッシュラインＸは、ノード５６０に対してローカルであり、ＣＴＤが成功すると、「無効化」をプロセッサＰ１に（及び図示のようにプロセッサＰ５にも）送る。

【０１８６】

図３３Ｂを簡単に参照すれば、参考としてここに取り上げる本発明と同日に出願されたバンドレン氏等の「分散型データ依存性ストール機構(Distributed Data Dependency Stall Mechanism)」と題する特許出願に開示されたように、プロセッサＰ１のような各プロセッサは、同じキャッシュ位置に対する保留中の読み取りがある場合にキャッシュへの調査をストールするためのロジックを備えている。上記の例が与えられると、Ｒｅａｄ Ｘの作用は、ミスアドレスファイル（ＭＡＦ）５７４にアドレスＸを記憶することである。ＭＡＦの内容は、到来する調査に対して比較され、そして到来する調査とＭＡＦのアドレス

間に一致があるときに、調査待ち行列がストールされる。

【０１８７】

「記入」データがプロセッサ５８２から返送されるときに調査待ち行列が解除される。しかしながら、同じ形式のトランザクション（即ち、Ｐ５がリモートＲｄ　Ｙを実行し、次いで、Ｐ６がＣＴＤ　Ｙを発生する）がノード５８０に生じる場合に、プロセッサＰ５の調査待ち行列がストールされ、Ｒｅａｄ　Ｙ要求が満足されるのを保留する。

Ｐ２により発生された「無効化」の後にＰ５からＦｏｒｗａｒｄｅｄ　Ｒｅａｄ　Ｙが送られる状態でＰ１調査待ち行列がストールされるのと同時に、Ｐ６により発生された「無効化」の後にプロセッサＰ１からＦｏｒｗａｒｄｅｄ　Ｒｅａｄ　Ｘが送られる状態でＰ５の調査待ち行列がストールされた場合には、停滞が生じる。

【０１８８】

この停滞問題を防止するための多数の解決策が存在する。第１に、全ての参照をリモートとすることができ、即ち全ての参照を（ホームノードからの参照も）、それらがホームノードに送られる前にスイッチに送ることができる。全ての参照がリモートにされた場合には、上述した中央の順序付けルールに基づき、停滞状態は生じない。第２の解決策は、キャッシュラインへのいずれかの参照がリモートから送られたときに所与のキャッシュラインへの全ての参照をストールすることである。しかしながら、この解決策は、これまでのローカル動作の性能に著しく影響し、それ故、好ましいものではない。

【０１８９】

本発明の１つの実施形態は、コマンドシャドー作用の使用によるローカル及びリモート参照の混合により課せられる潜在的な停滞を克服する。キャッシュラインＸへのローカル参照がリモートプロセッサへ送られると、そのキャッシュラインへのその後の全ての参照がハイアラーキースイッチへリモートから送られ、キャッシュラインのローカル参照及びその後の全ての参照が完了するまで、中央で順序付けされる。従って、まだシャドー状態であるキャッシュラインへの以前の参照は、キャッシュラインへの現在の参照もシャドー状態にする。　図３４及び３５を参照して、上記例をシャドーコマンドの使用と共に説明する。図３５は、ＴＴＴの内容を例示している。第１プロセッサＰ１は、ＲｄＸをアービターに発生する。上述したように、プロセッサＰ５へのＦＲｄＸを生じ、これはＴＴＴに記録される。その後、プロセッサＰ２は、ＣＴＤ　ＸをＡＲＢに発生する。ＡＲＢは、ＴＴＴを検査し、リモートプロセッサへ送られる保留中のローカル読み取りがあると決定し、そしてグローバルポートからプロセッサＰ５へＩｎｖａｌＸを送る。又、この動作を表わすエントリもＴＴＴに形成され、そのシャドービットがセットされる。

【０１９０】

同時に、ノード５８０において、同様の一連のトランザクションが生じる。プロセッサＰ５は、Ｒｄ　Ｙを発生し、これはノード５６０に送られると共に、Ｐ５アドレスをエントリに含ませることによりＴＴＴに記録される。プロセッサＰ６は、その後、ＣＴＤ　Ｙを発生する。ノード５８０のアービターは、ＣＴＤアドレスをＴＴＴ内の保留中読み取りに対して一致させ、そしてＣＴＤ　Ｙをグローバルポートにわたり「シャドー」処理する。そのＣＴＤ　Ｙに対してＴＴＴにエントリが形成され、このエントリは、ＴＴＴにおいてそのシャドービットをセットし、ＣＴＤ　Ｙが、Ｙへの要求の適切な順序付けを確保するためにリモート送信されたローカル参照であることを指示する。

上述したように、両ノードにおいて調査シーケンスで「無効化」の後にＦＲｄがあるときに問題が生じる。「無効化」は、ここでは中央で順序付けされるので、両無効化を両方の「送信読み取り」の前にそれらの調査待ち行列へ送信できないことにはならない。というのは、それらは、共通点即ちハイアラーキースイッチにおいてシリアル化されるからである。従って、図３６を参照すれば、コマンドの入力シーケンスは、ハイアラーキースイッチ５６８へ入力されるように示されている。許容し得る出力シリアル化順序は、順序ａ－ｆとして識別される。上記のＱ１チャンネル順序付けルールによれば、ハイアラーキースイッチへのパケット入力のシリアル化順序がスイッチ出力に維持されることに注意されたい。それ故、上記の場合には、ＦＲｄは、行先ノードへ送られるときにその関連する「無

効化」に先行する。
【０１９１】
ノードの１つは、調査待ち行列に「無効化」を受け取り、その後、「送信読み取り」を受け取る。例えば、シリアル化順序を用いて、プロセッサＰ５の調査待ち行列は、Ｉｎｖａｌ　Ｙによりストールされ、そしてＦｒｄ　Ｘがストールされて、記入を保留する。しかしながら、この例では、Ｆｒｄ　Ｙは、ＩｎｖａｌＸの後ではなく、従って、Ｐ５調査待ち行列を阻止しないように「記入」データを与えることができる。
リモート参照のためにデータが返送されるときには、その参照に対応するＴＴＴエントリがドロップされる。元の参照をシャドー処理した他の参照がＴＴＴに存在することがある。これらコマンドがハイアラーキースイッチから受け取られるときには、シャドー処理されたコマンドの各々に対するＴＴＴエントリもドロップされる。最終的に、リモートアクセス及びシャドーアクセスが全て完了し、そしてＴＴＴがもはやキャッシュラインへマップするエントリを含まなくなると、そのキャッシュラインへのその後のローカル参照をシャドー処理する必要がなくなる。
【０１９２】
従って、シャドーコマンドの使用により、ローカル及びリモートコマンドの共存から生じるリソース依存性の停滞を、ハードウェアの複雑さを著しく増加せずに排除することができる。上記の例は、「送信読み取り」及びＣＴＤの使用を含むが、シャドーコマンド方法は、他の形式の命令及びマルチプロセッサにも等しく適用できることに注意されたい。一般に、ローカルアドレスＸへの参照が存在し、そしてローカルアドレスＸへの以前のメッセージがリモートプロセッサ（ＴＴＴにより指示された）へ送られるか、又はＸへの以前の参照がまだシャドー処理されるときには、Ｘへの現在の参照もシャドー処理される。
更に、この方法は、上記の単なるマルチプロセッサ／スイッチハイアラーキーよりも多数のハイアラーキーレベルを含む他の形式のアーキテクチャーにも使用できる。例えば、上記方法は、多数のハイアラーキーレベルを含み、コマンドがキャッシュラインへの以前の保留中参照のハイアラーキーレベルに基づいて適当なハイアラーキーレベルに送られるコンピュータシステムにも使用できる。
【０１９３】
従って、大型のＳＭＰコンピュータシステムに使用するためのアーキテクチャ及びコヒレンスプロトコルについて説明した。ＳＭＰシステムのアーキテクチャは、多数のマルチプロセッサノードをスイッチに接続して最適な性能で動作することのできるハイアラーキースイッチ構造体を備えている。各マルチプロセッサノード内には、マルチプロセッサノードの全てのプロセッサを最高の性能で動作できるようにする同時バッファシステムが設けられる。メモリはノード間で共用され、マルチプロセッサノードの各々にメモリの一部分が常駐する。
マルチプロセッサノードの各々は、メモリコヒレンス性を維持するための多数の要素、即ちビクティムキャッシュ、ディレクトリ及びトランザクション追跡テーブルを含む。ビクティムキャッシュは、リモートのマルチプロセッサノードに記憶されたメモリを行先とするビクティムデータを選択的に更新することができ、これにより、メモリの全性能が改善される。ディレクトリに関連して使用されて、メモリに書きこまれるべきビクティムを識別する遅延書き込みバッファを各メモリに含ませることによりメモリ性能が更に改善される。
【０１９４】
各ノードのディレクトリの出力に接続されたＡＲＢバスは、ＳＭＰを経て転送される全てのメッセージに対して中央の順序付けポイントとなる。本発明の１つの実施形態によれば、メッセージは、多数のトランザクションを含み、各トランザクションは、メッセージの処理段階に基づいて多数の異なる仮想チャンネルに指定される。従って、仮想チャンネルの使用は、システム順序を維持する簡単な方法を与えることによりデータのコヒレンス性を維持する上で助けとなる。仮想チャンネル及びディレクトリ構造体を使用すると、従来停滞を生じるキャッシュコヒレンス性の問題を回避することができる。

以上、本発明の好ましい実施形態を説明したが、その概念を組み込んだ他の実施形態も使用できることが当業者に明らかであろう。それ故、本発明は、上記の実施形態に限定されるものではなく、特許請求の範囲のみによって限定されるものとする。

【図面の簡単な説明】

【図１Ａ】公知の対称的なマルチプロセッサコンピュータシステムのブロック図である。

【図１Ｂ】公知の対称的なマルチプロセッサコンピュータシステムのブロック図である。

【図２】スイッチを備えた本発明によるマルチプロセッサコンピュータノードの１つの一実施形態を示すブロック図である。

【図３】多数の同時挿入バッファを備えた図１のスイッチのデータ経路を示すブロック図である。

【図４Ａ】図３に示す同時挿入バッファの１つの一実施形態を示すブロック図である。

【図４Ｂ】図４に示す同時挿入バッファの１つを制御するためのロジックの一実施形態を示すブロック図である。

【図５】図３に示す同時挿入バッファの１つの別の実施形態を示すブロック図である。

【図６】同様のノードの大きなネットワークへ接続するように拡張された図２のマルチプロセッサコンピュータノードのブロック図である。

【図７Ａ】図６のマルチプロセッサノードと同様の多数のノードを使用して実施されたＳＭＰシステムの一実施形態を示す図である。

【図７Ｂ】図６のマルチプロセッサノードと同様の多数のノードを使用して実施されたＳＭＰシステムの別の実施形態を示す図である。

【図８】図６のグローバルポートのブロック図である。

【図９】図６のマルチプロセッサノードのディレクトリにおけるエントリーを示す図である。

【図１０】図８のグローバルポートに使用するためのトランザクション追跡テーブル（ＴＴＴ）を示す図である。

【図１１】図７Ａにおいて多数のノードを接続するためのハイアラーキー式スイッチを示すブロック図である。

【図１２Ａ】停滞を排除するハイアラーキースイッチ用の相互接続ロジックの一実施形態を示すブロック図である。

【図１２Ｂ】図１２Ａの相互接続ロジックの動作を示すフローチャートである。

【図１３】マルチプロセッサノードの１つから送信されるデータを停止する流れ制御を与えるために図１２Ａの相互接続ロジックに使用される方法を示すフローチャートである。

【図１４】ハイアラーキースイッチに対してバスを経て行われるアドレス及びデータパケットの転送を示すタイミングである。

【図１５】ハイアラーキースイッチにおいて順序を維持するためのバッファロジックの一実施形態を示すブロック図である。

【図１６】ハイアラーキースイッチに対して順序を維持するためのバッファロジックの別の実施形態を示すブロック図である。

【図１６Ａ】チャンネルの依存性を矢印で示す図である。

【図１７】図１６のバッファロジックを動作する１つの方法を示すフローチャートである。

【図１８】ハイアラーキースイッチにおいて順序を維持するためのバッファロジックの別の実施形態を示すブロック図である。

【図１８Ａ】チャンネルの依存性を矢印で示す図である。

【図１９】図７Ａ又は７ＢのＳＭＰに使用するためのプロセッサ命令－ネットワーク命令の変換を示すテーブルである。

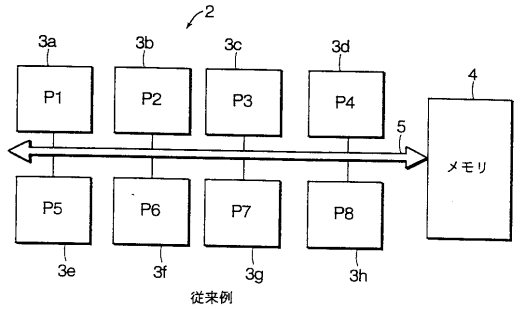
【図２０Ａ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。

【図２０Ｂ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。

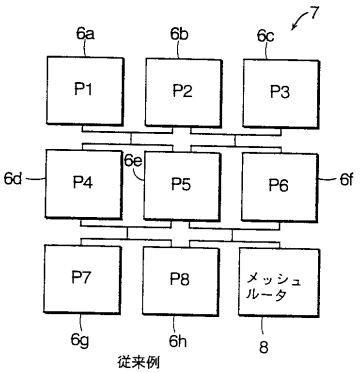
【図２０Ｃ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。
【図２０Ｄ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。
【図２０Ｅ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。
【図２０Ｆ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。
【図２０Ｇ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。
【図２０Ｈ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。
【図２０Ｉ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。
【図２０Ｊ】図７Ａ又は７ＢのＳＰＭにおいてノード間にパケットを転送するための多数の通信流を示す図である。
【図２１】図２又は６のマルチプロセッサシステムに使用するためのメモリモジュールのレイアウトを示すブロック図である。。
【図２２】遅延書き込み動作のために図２１のメモリモジュールにより使用される制御ロジックを示すタイミング図である。
【図２３】本発明の１つの実施形態においてキャッシュコヒレンス性を維持するためにチャンネルに対してマップされる個別のトランザクションの使用を示すフローチャートである。
【図２４】図７Ａ又は７ＢのＳＭＰにおいて仮想チャンネルを取り扱うための共用待ち行列構造体の一実施形態を示すブロック図である。
【図２５】図７Ａ又は７ＢのＳＭＰのノード及びハイアラーキーチャンネルにおける個々のチャンネルバッファの一実施形態を示すブロック図である。
【図２６】仮想チャンネル間にある程度の順序が維持されない場合に生じる問題を説明するためのブロック図である。
【図２７Ａ】図７Ａ又は７ＢのＳＭＰにおいてコヒレントな通信を与えるためのＱ１チャンネルにおける流れ及び順序付けの制約を示すブロック図である。
【図２７Ｂ】図７Ａ又は７ＢのＳＭＰにおいてコヒレントな通信を与えるためのＱ１チャンネルにおける流れ及び順序付けの制約を示すブロック図である。
【図２７Ｃ】図７Ａ又は７ＢのＳＭＰにおいてコヒレントな通信を与えるためのＱ１チャンネルにおける流れ及び順序付けの制約を示すブロック図である。
【図２８Ａ】図７Ａ及び７ＢのＳＭＰのディレクトリエントリーにおおよそのベクトル存在ビットがあるために生じる曖昧さの問題を説明するブロック図である。
【図２８Ｂ】図７Ａ及び７ＢのＳＭＰのディレクトリエントリーにおおよそのベクトル存在ビットがあるために生じる曖昧さの問題を説明するブロック図である。
【図２９】図２８に示す問題の結果として生じるデータの曖昧さを防止するために使用される方法を示すブロック図である。
【図３０】異なるチャンネルのパケットが順序づれして受け取られるために生じるコヒレンス性の問題を示すブロック図である。
【図３１】図２９に示すコヒレンス性の問題を防止するための記入マーカーの使用を示すブロック図である。
【図３２】図３１について述べた流れ間の命令の状態を表わすＴＴＴのエントリを示す図である。
【図３３Ａ】ＳＭＰシステムにおけるダーティへの変更コマンドの作用を示すブロック図である。
【図３３Ｂ】ＳＭＰシステムにおけるダーティへの変更コマンドの作用を示すブロック図

である。
【図３４】図３３について述べた問題を矯正するためのシャドーコマンドの使用を示すブロック図である。
【図３５】図３４について述べたフロー間の命令の状態を反映するＴＴＴのエントリを示す図である。
【図３６】図３５に示す例における許容し得る逐次順序付けを示すフローチャートである。
【符号の説明】
１０　マルチプロセッサノード
１１　アービター（ＱＳＡ　ＡＲＢ）
１２ａ、１２ｂ、１２ｃ、１２ｄ　プロセッサモジュール
１３　メモリ
１３ａ－１３ｄ　メモリモジュール
１４　Ｉ／Ｏプロセッサ（ＩＯＰ）モジュール
１４ａ　Ｉ／Ｏバス
１４ｂ　ＩＯＰタグ記憶装置
１４ｃ　ＩＯＰキャッシュ
１５　ローカルスイッチ
１６ａ－１６ｉ　データリンク
１７　ＡＲＢバス
１８　ＱＳＡチップ
１９　ＱＳＤチップ
２０　デュープリケートタグ記憶装置（ＤＴＡＧ）
２５ａ－２５ｅ　同時挿入バッファ（ＳＩＢ）
２７　メインアービター
３２　バッファ
３４ａ－３４ｈ　マルチプレクサ
３６　入力アービター
３８　出力アービター
１２２　トランザクション追跡テーブル
１２４　ビクティムキャッシュ
１４０　ディレクトリ

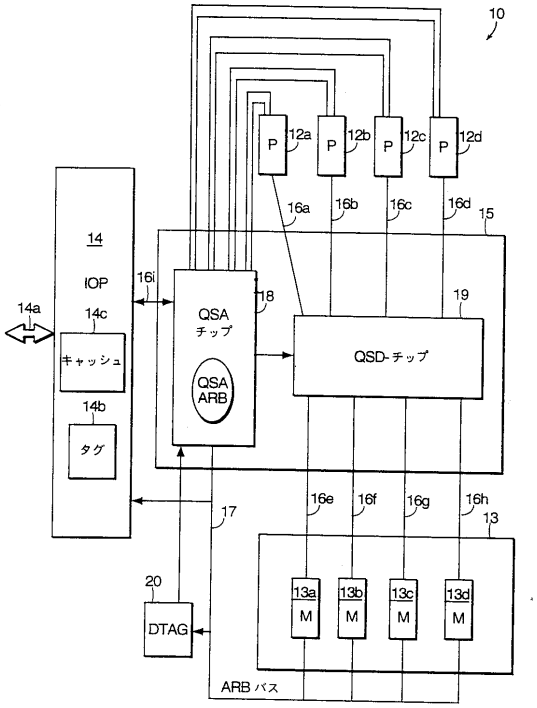
【図１Ａ】



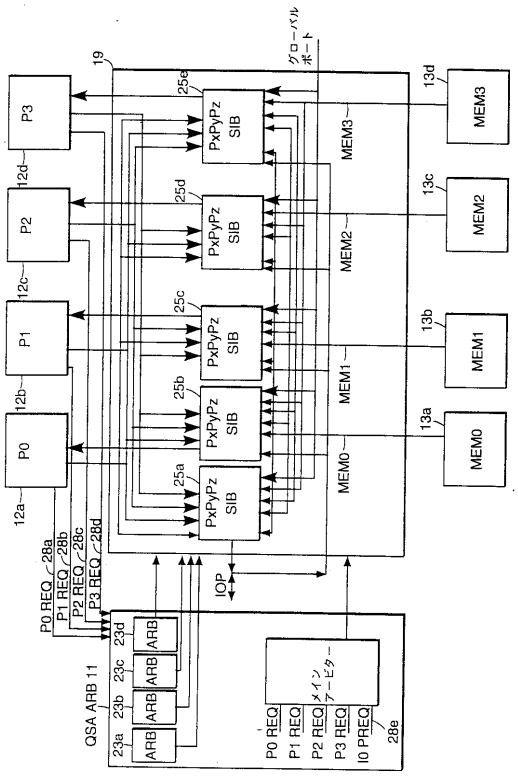
【図１Ｂ】



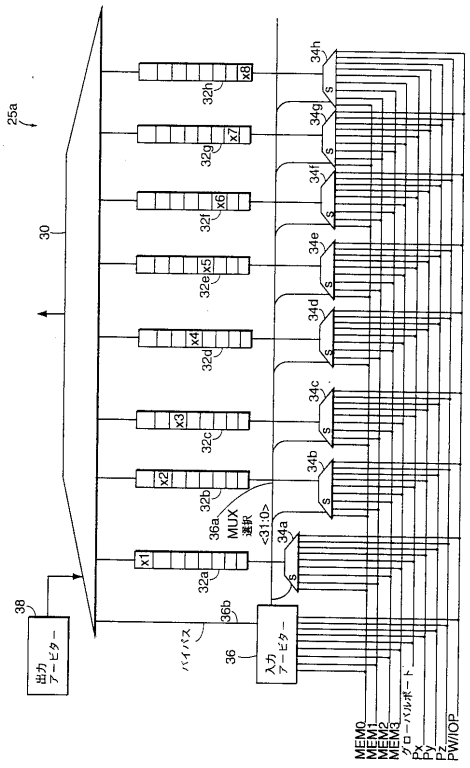
【図２】



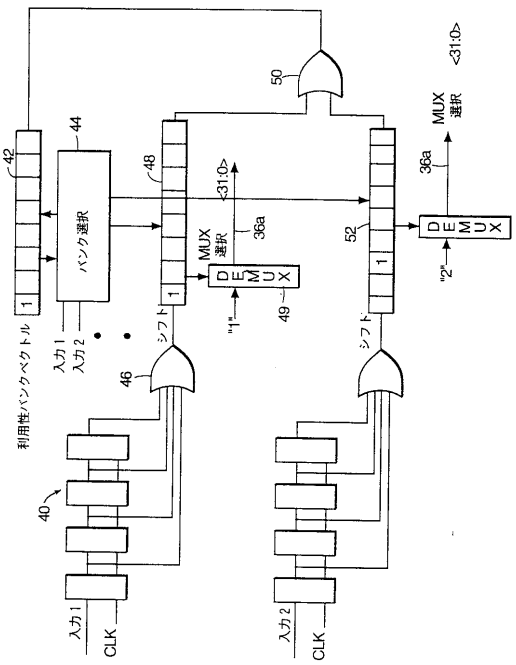
【図３】



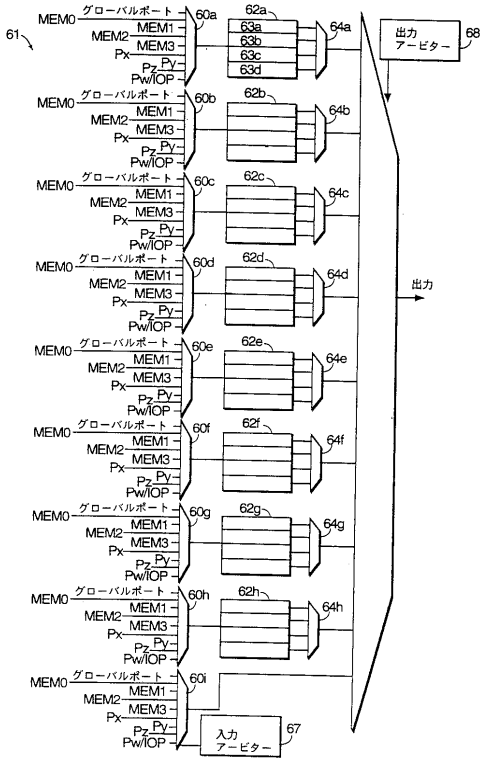
【図４Ａ】



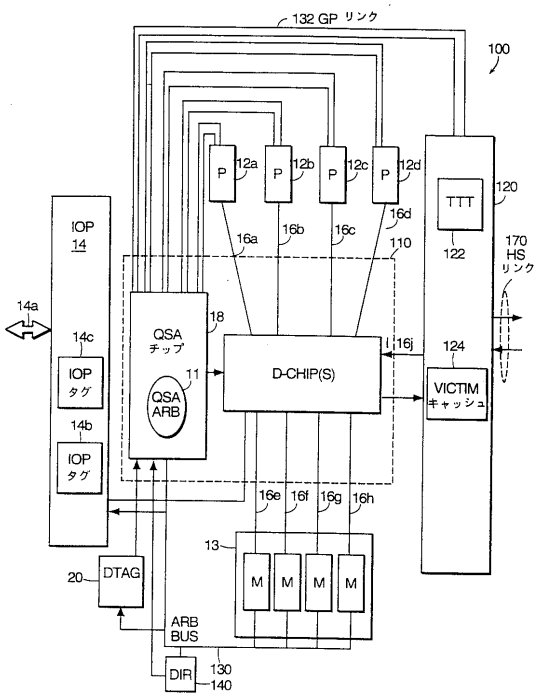
【図４Ｂ】


利用性バンクベクトル
バンク選択
入力1
入力2
シフト
MUX 選択
DEMUX
"1"
"2"
CLK
入力1
入力2

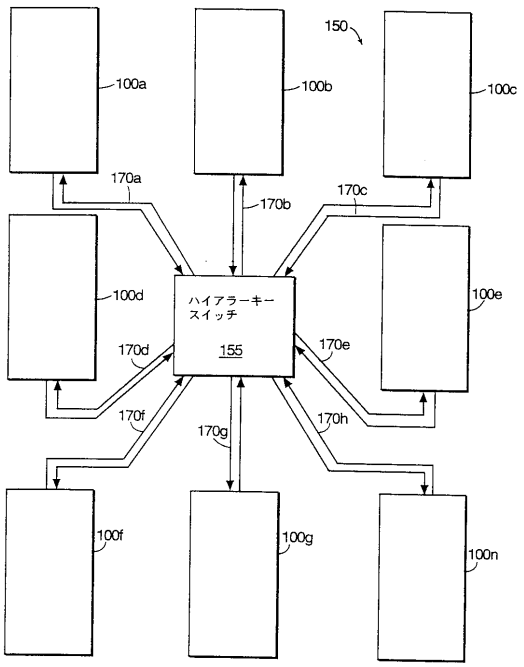
【図５】


MEM0
MEM1
MEM2
MEM3
Px
Py
Pz
Pw/IOP
グローバルポート
出力
アービター
入力
アービター

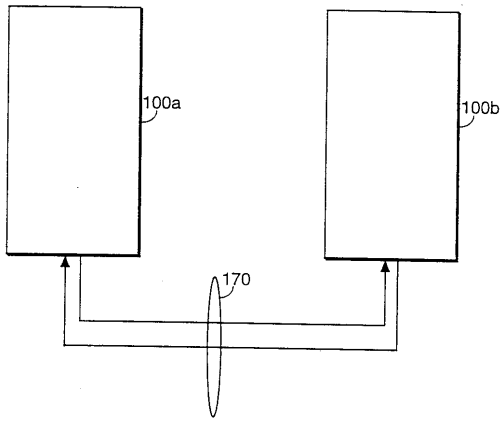
【図６】


132 GP リンク
IOP
14
14a
14c
IOP
タグ
14b
IOP
タグ
QSA
チップ
QSA
ARB
D-CHIP(S)
TTT
122
VICTIM
キャッシュ
170
HS
リンク
DTAG
ARB
BUS
DIR
M

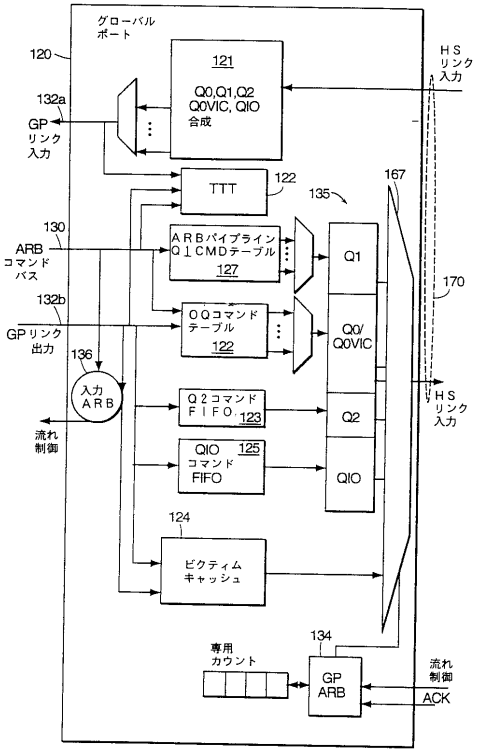
【図７Ａ】


ハイアラーキー
スイッチ
155

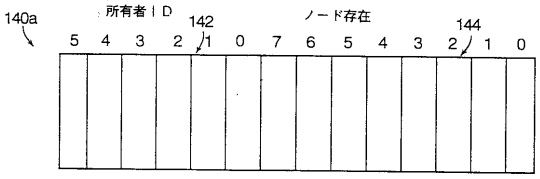
【図７Ｂ】


100a
100b
170

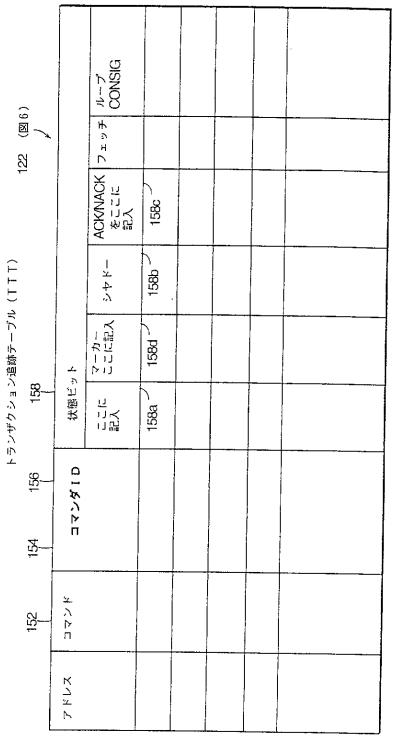
【図８】


グローバル
ポート
120
121
Q0,Q1,Q2
Q0VIC, QIO
合成
HS
リンク
入力
132a
GP
リンク
入力
TTT
122
135
167
130
ARB
コマンド
バス
ARBパイプライン
Q1CMDテーブル
127
Q1
170
132b
GPリンク
出力
QQコマンド
テーブル
122
Q0/
Q0VIC
136
入力
ARB
Q2コマンド
FIFO
123
Q2
HS
リンク
入力
流れ
制御
QIO 125
コマンド
FIFO
QIO
124
ビクティム
キャッシュ
134
専用
カウント
GP
ARB
流れ
制御
ACK

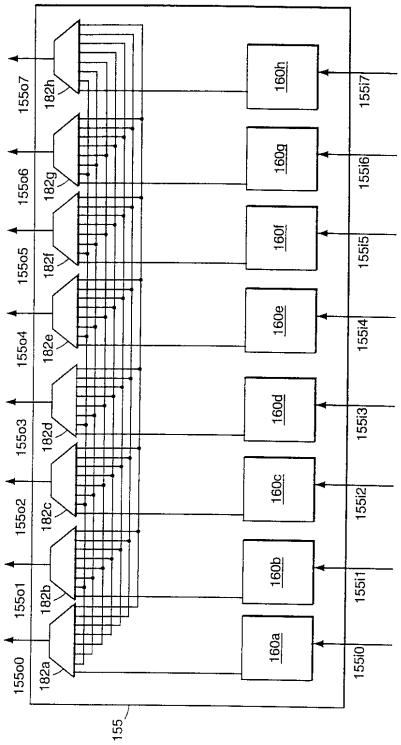
【図９】


140a
所有者ＩＤ
142
ノード存在
144
5 4 3 2 1 0 7 6 5 4 3 2 1 0

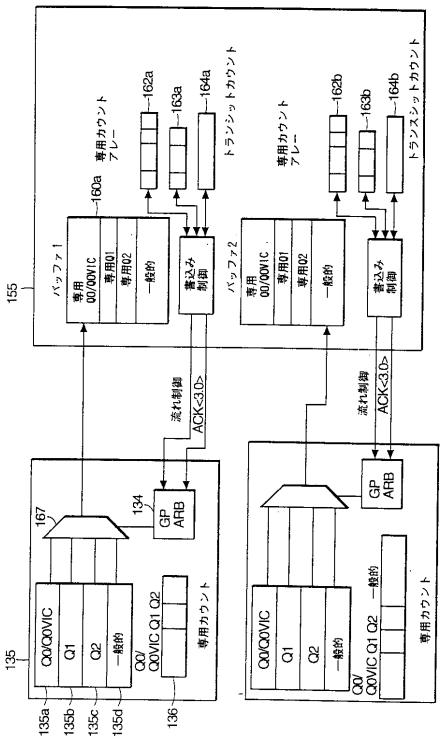
【図１０】


トランザクション追跡テーブル（ＴＴＴ）
122 (図6)
158
状態ビット
156
コマンドＩＤ
154
152
コマンド
アドレス
ここに記入
158a
マーカー
ここに記入
158d
シャドー
158b
ACK/NACK
をここに
記入
158c
フェッチ
ループ
CONSIG

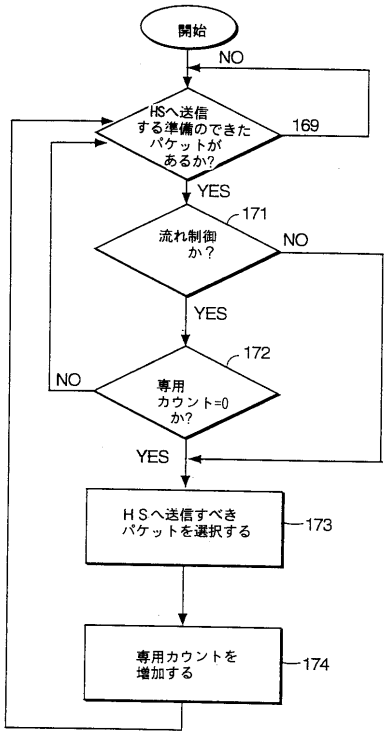
【図１１】



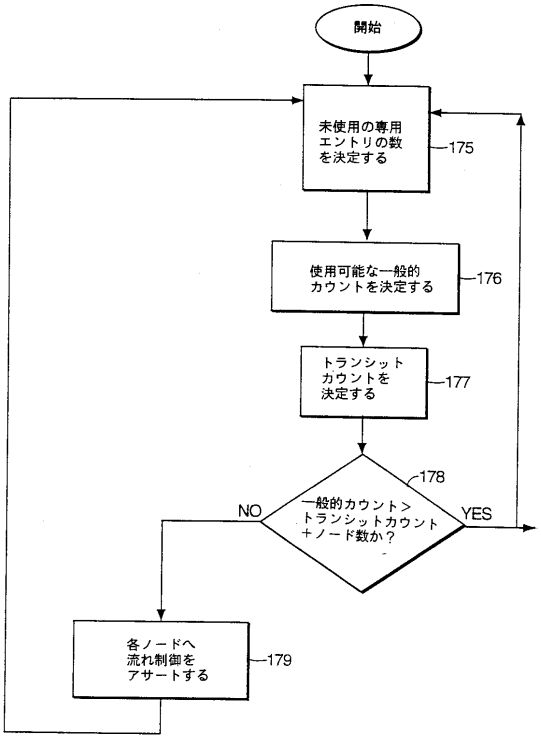
【図１２Ａ】



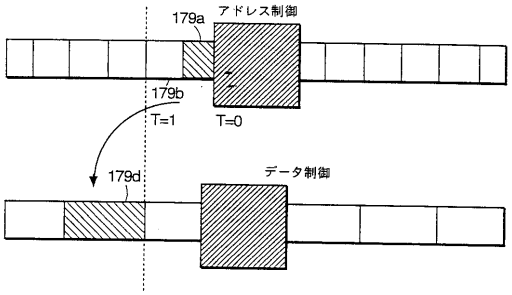
【図１２Ｂ】



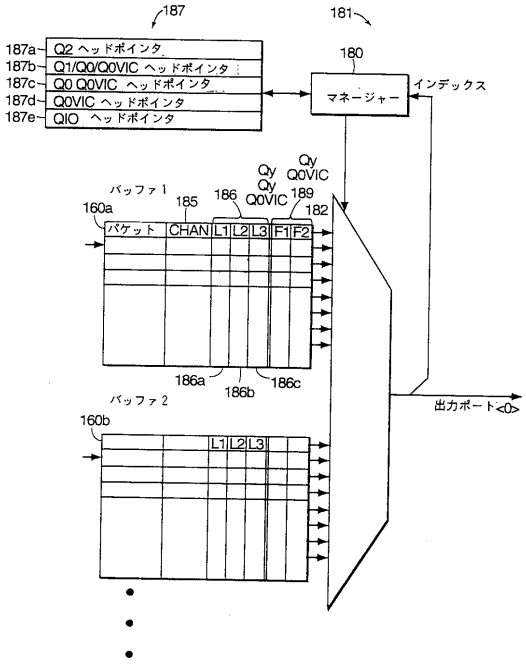
【図１３】



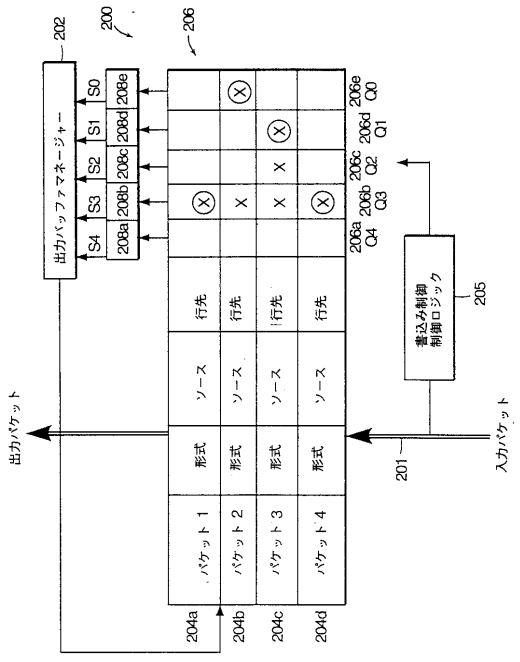
【図１４】



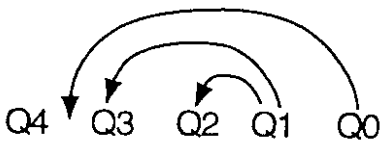
【図１５】



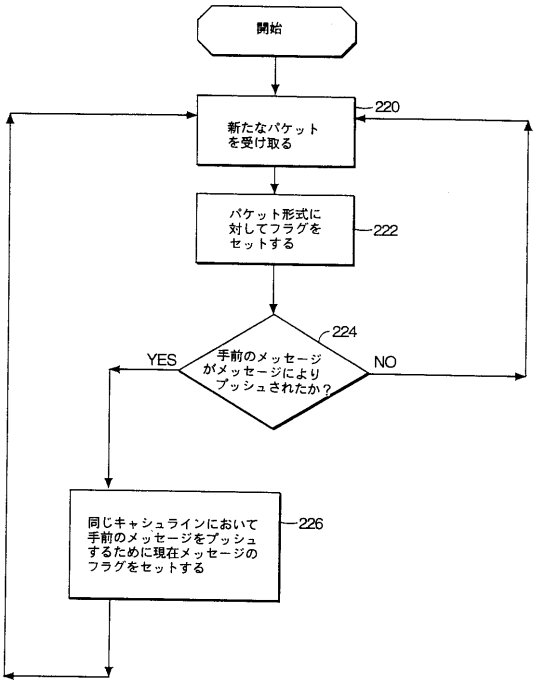
【図１６】



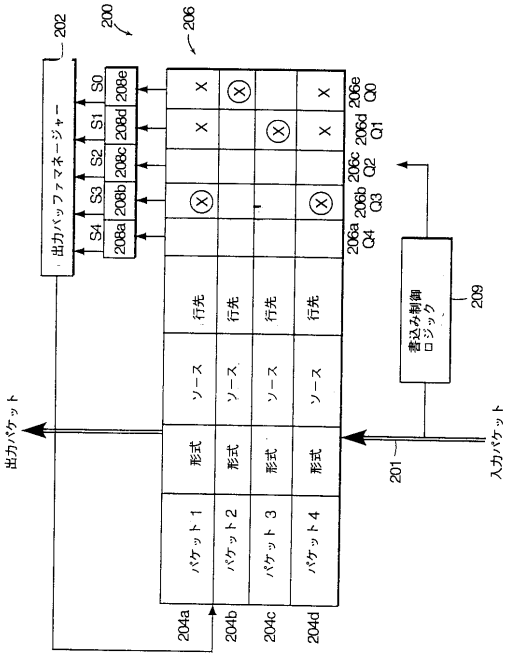
【図１６Ａ】



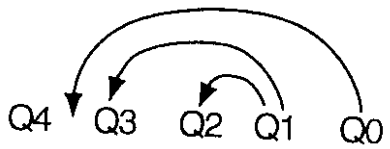
【図１７】



【図１８】



【図１８Ａ】



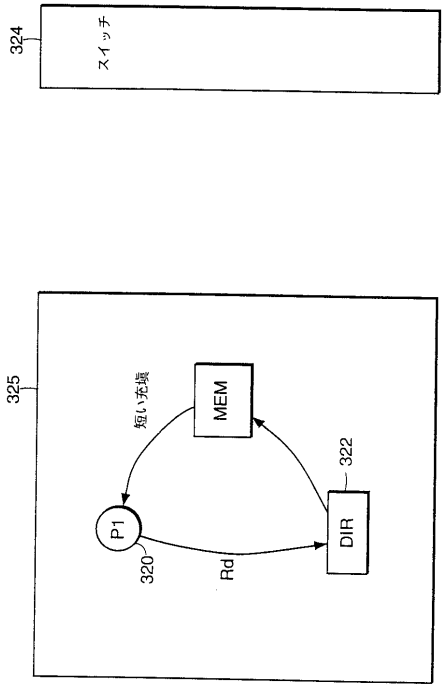
【図１９】

22/43

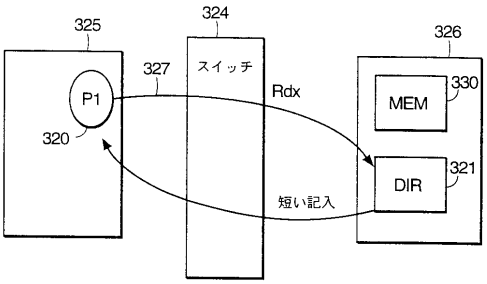
300	305	320
ブロック読み取り	無効	
	クリーン	
	ダーティ	送信読み取り
	ダーティ共用	送信読み取り
ブロック読み取り変更	無効	
	クリーン	無効化
	ダーティ	送信読み取り変更
	ダーティ共用	送信読み取り変更
フェッチ	無効	
	クリーン	
	ダーティ	送信読み取り
	ダーティ共用	送信読み取り
ブロック読み取りビクティム	無効	
	クリーン	
	ダーティ	送信読み取り
	ダーティ共用	送信読み取り
ブロック読み取り変更ビクティム	無効	
	クリーン	無効化
	ダーティ	送信読み取り変更
	ダーティ共用	送信読み取り変更
ブロックフェッチビクティム	無効	
	クリーン	
	ダーティ	送信読み取り
	ダーティ共用	送信読み取り
ビクティム	任意の状態	
クリーンビクティム	任意の状態	

300a	305a	320a
クリーン－ダーティ	無効	
	クリーン	無効化
	ダーティ	N/A
	ダーティ共用	無効化
共用－ダーティ	無効	
	クリーン	無効化
	ダーティ	N/A
	ダーティ共用	N/A
ダーティへのSTC変更	無効	
	クリーン	無効化
	ダーティ	N/A
	ダーティ共用	無効化
無効－ダーティ	無効	
	クリーン	無効化
	ダーティ	無効化
	ダーティ共用	無効化
全ブロック書き込み	無効	
	クリーン	無効化
	ダーティ	無効化
	ダーティ共用	無効化

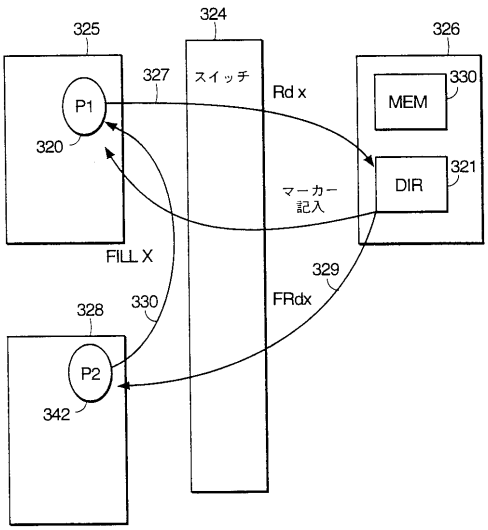
【図２０Ａ】



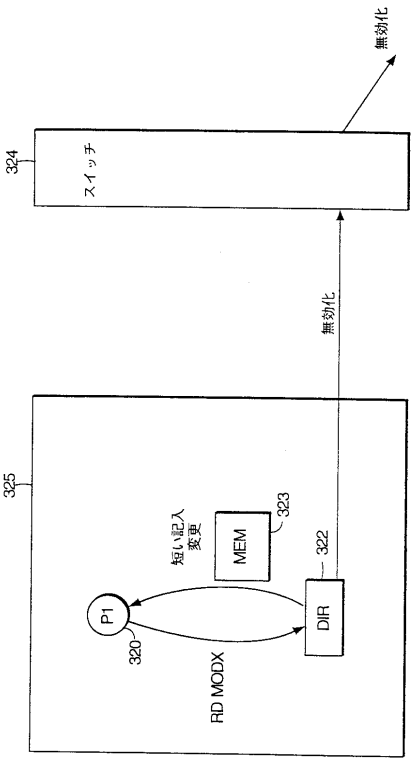
【図２０Ｂ】



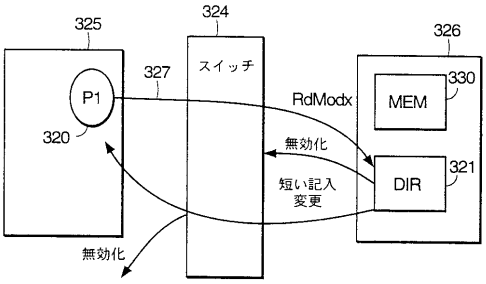
【図２０Ｃ】



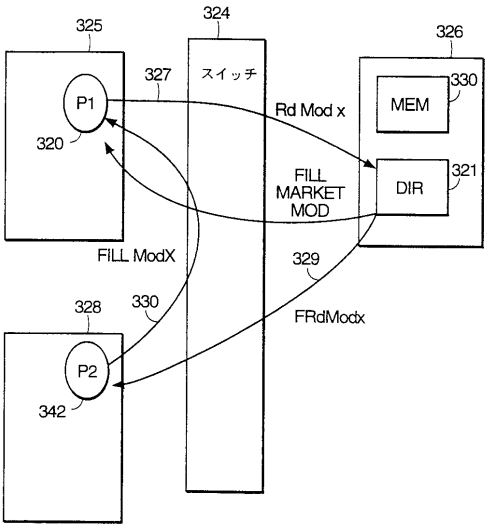
【図２０Ｄ】



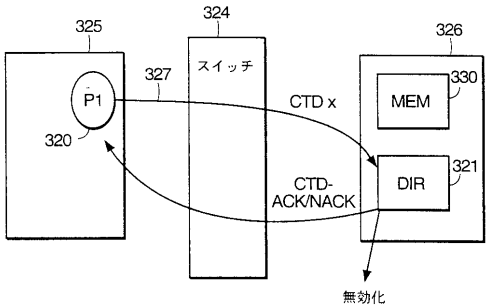
【図２０Ｅ】



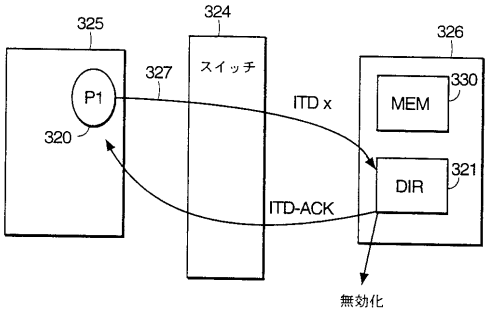
【図２０Ｆ】



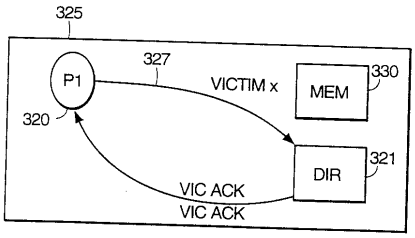
【図２０Ｇ】



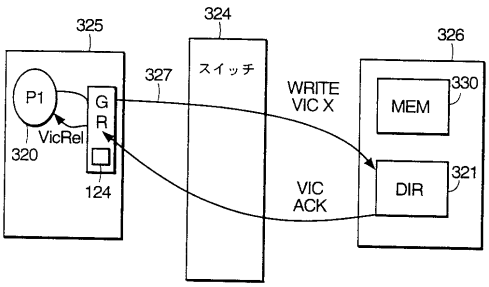
【図２０Ｈ】



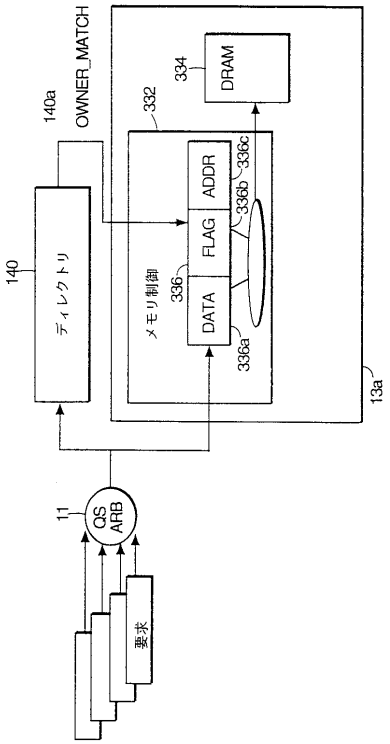
【図２０Ｉ】



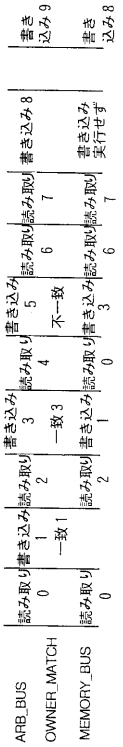
【図２０Ｊ】



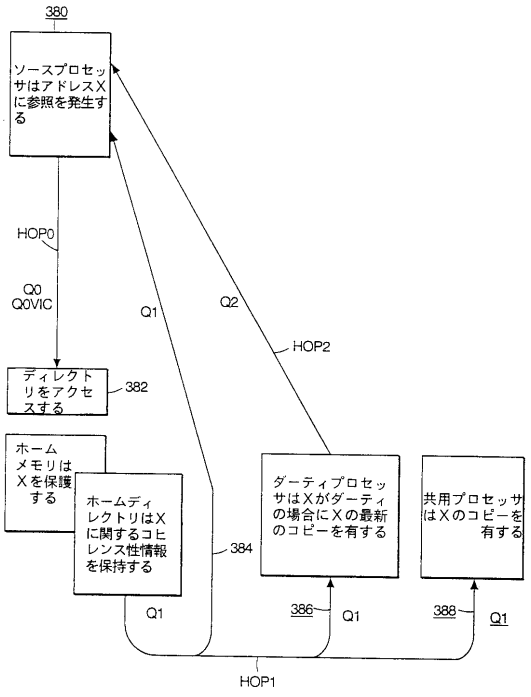
【図２１】



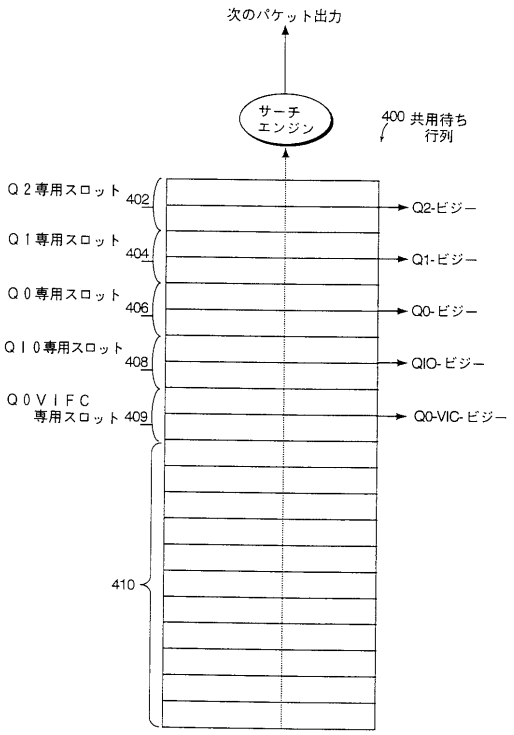
【図２２】



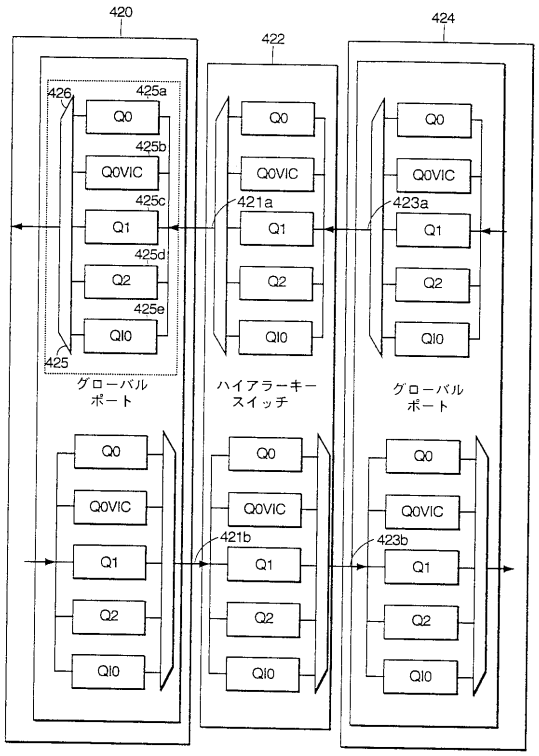
【図２３】



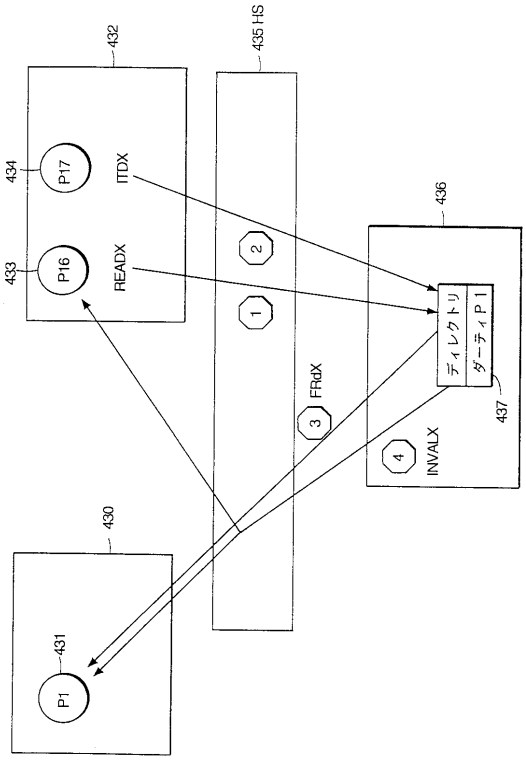
【図２４】



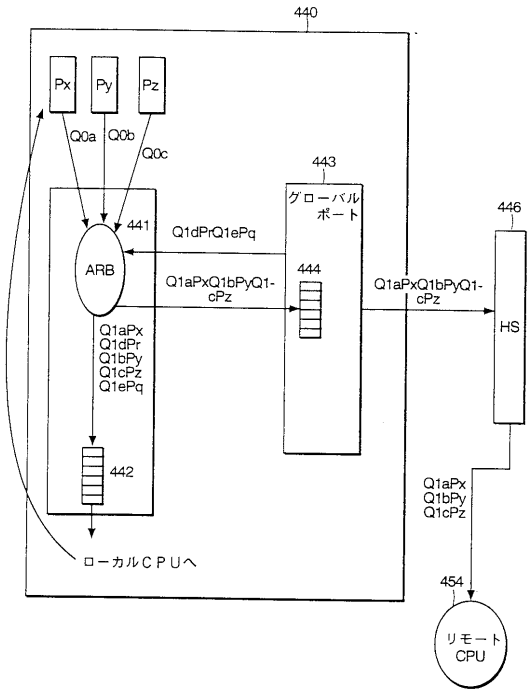
【図２５】



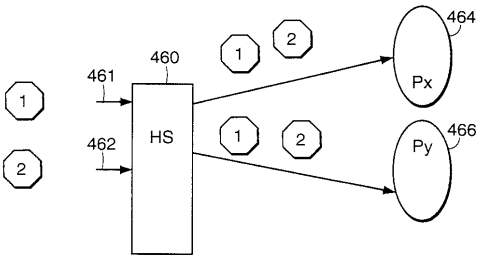
【図２６】



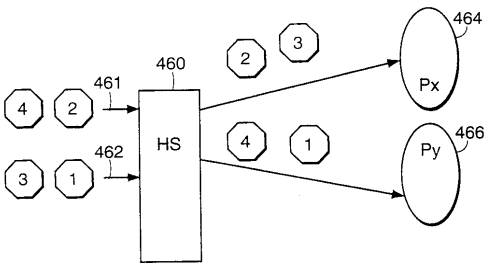
【図２７Ａ】



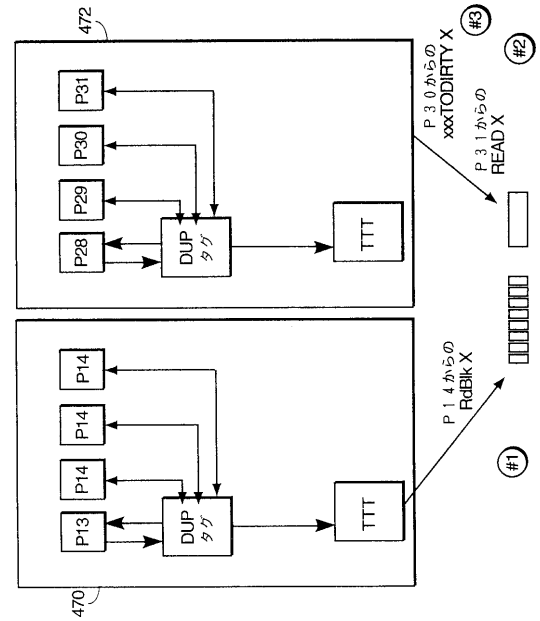
【図２７Ｂ】



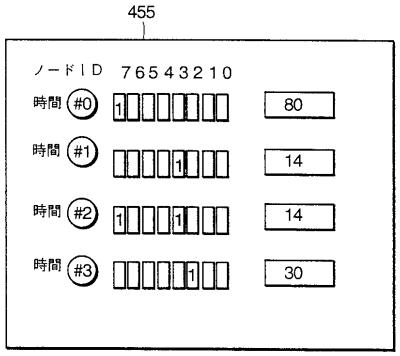
【図２７Ｃ】



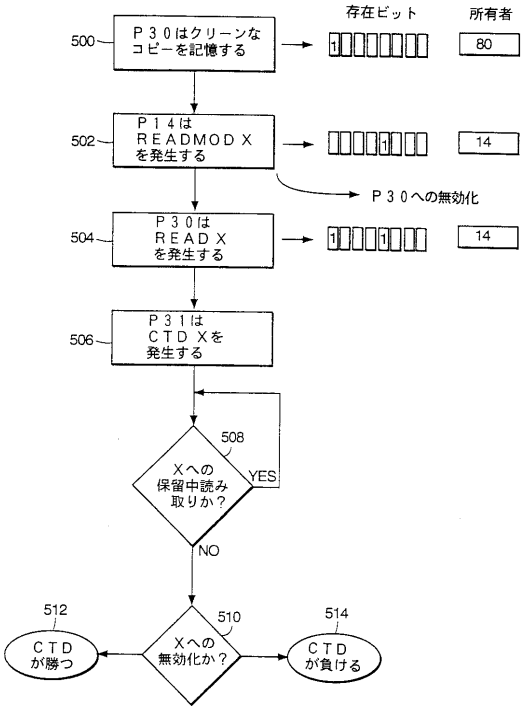
【図２８Ａ】



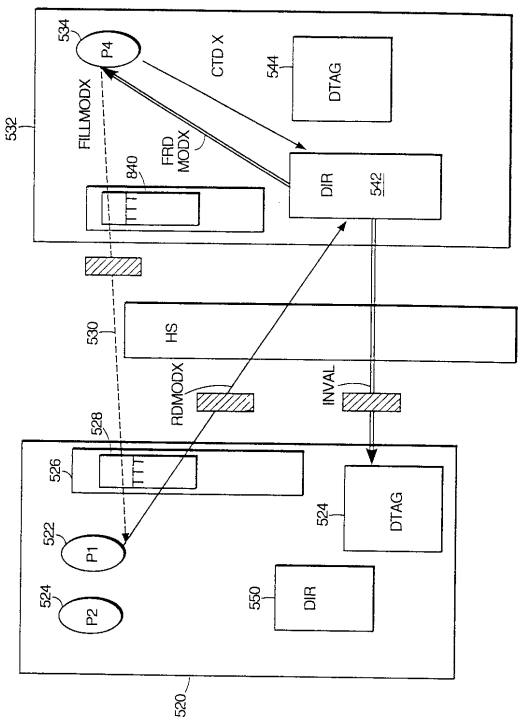
【図２８Ｂ】



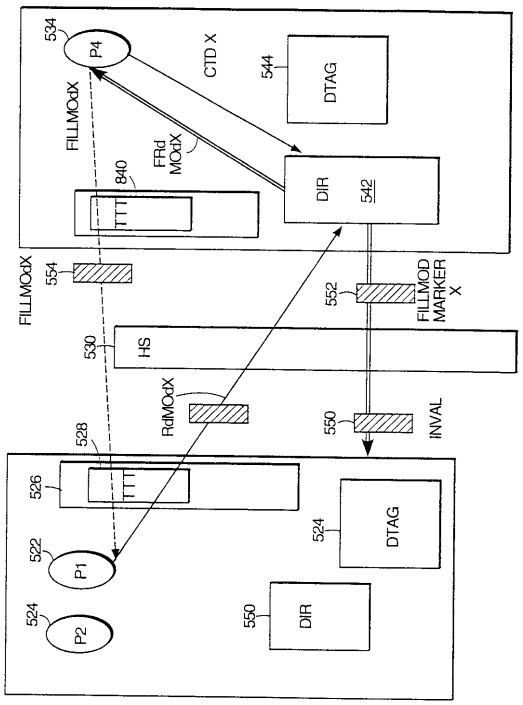
【図２９】



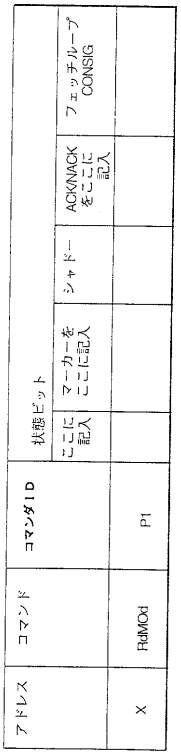
【図３０】



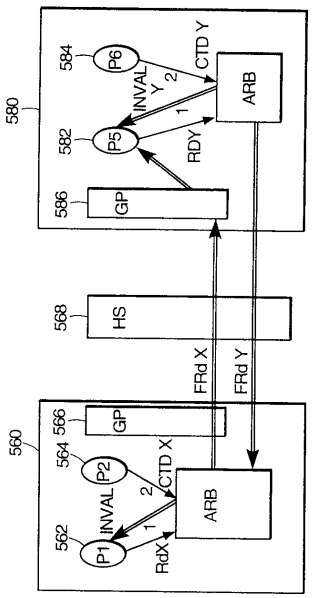
【図３１】



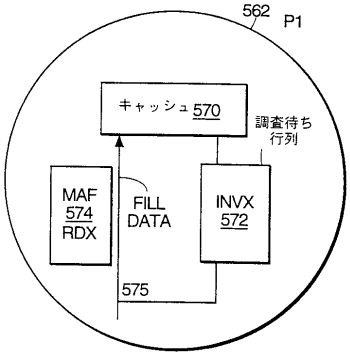
【図３２】



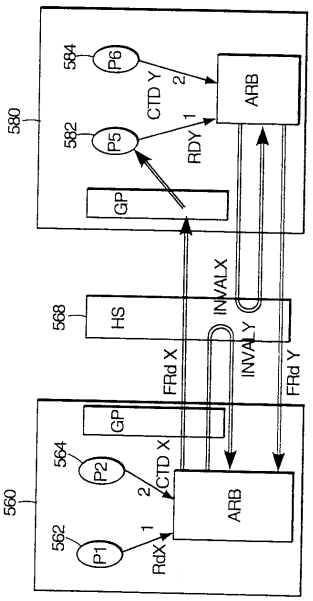
【図３３Ａ】



【図３３Ｂ】



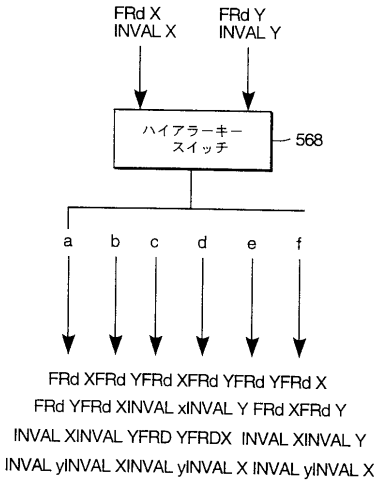
【図３４】



【図３５】

アドレス	コマンド	コマンドID	状態バス				
			ここに記入	マーカーをここに記入	シャドー	ACK/NACKをここに記入	フェッチ	ループCONSIG
X	FRD	P1			X			
X	INVAL	P2						

【図３６】



フロントページの続き

(72)発明者 スティーヴン アール ヴァンドーレン
アメリカ合衆国 マサチューセッツ州 ０１５３２ ノースボロー デイヴィス ストリート ２２４
(72)発明者 シモン シー ステイーリイ
アメリカ合衆国 ニューハンプシャー州 ０３０５１ ハドソン アンナ ルイス ドライヴ ８
(72)発明者 マドハミトラ シャルマ
アメリカ合衆国 マサチューセッツ州 ０１５４５ シュローズバリー コモンズ ドライヴ ５５－４６
(72)発明者 ディヴィッド エム フェンウィック
アメリカ合衆国 マサチューセッツ州 ０１５４５ アクトン ブラウン ベア クロッシング ２９７

合議体
審判長 吉岡 浩
審判官 清木 泰
審判官 田中 秀人

(56)参考文献 特開平７－３１１７５１（ＪＰ，Ａ）
特開平４－３２８６５３（ＪＰ，Ａ）
特開平８－３２０８２７（ＪＰ，Ａ）