(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4712974号
(P4712974)
(45)発行日　平成23年6月29日(2011.6.29)　(24)登録日　平成23年4月1日(2011.4.1)

(51)Int.Cl.　F I
G06F 12/08 (2006.01)　G06F 12/08 531B
G06F 12/08 501Z
G06F 12/08 551C
G06F 12/08 575

請求項の数 14　(全 23 頁)

(21)出願番号　特願2000-590061 (P2000-590061)
(86)(22)出願日　平成11年8月26日 (1999.8.26)
(65)公表番号　特表2002-533812 (P2002-533812A)
(43)公表日　平成14年10月8日 (2002.10.8)
(86)国際出願番号　PCT/US1999/019471
(87)国際公開番号　WO2000/038069
(87)国際公開日　平成12年6月29日 (2000.6.29)
審査請求日　平成18年7月12日 (2006.7.12)
(31)優先権主張番号　09/217,367
(32)優先日　平成10年12月21日 (1998.12.21)
(33)優先権主張国　米国(US)

(73)特許権者　591016172
アドバンスト・マイクロ・ディバイシズ・インコーポレイテッド
ＡＤＶＡＮＣＥＤ　ＭＩＣＲＯ　ＤＥＶＩＣＥＳ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国、９４０８８－３４５３　カリフォルニア州、サニィベイル、ピィ・オゥ・ボックス・３４５３、ワン・エイ・エム・ディ・プレイス、メイル・ストップ・６８（番地なし）
(74)代理人　100064746
弁理士　深見　久郎
(74)代理人　100085132
弁理士　森田　俊雄

最終頁に続く

(54)【発明の名称】コヒーレンシ維持のための柔軟なプローブ／プローブ応答経路制御

(57)【特許請求の範囲】
【請求項１】
装置であって、
プローブを受取るよう結合される第１のノードと、前記第１のノードに結合される第２のノードおよび第３のノードと、各前記ノードに結合されたメモリとを含み、前記第１のノードはキャッシュを含み、前記プローブは、前記第１のノードが、前記メモリから読み出されたデータのキャッシュブロックのコピーを前記キャッシュにストアしているかどうかを判断する要求であり、前記キャッシュブロックが前記第１のノードにストアされている場合に前記第１のノードによって行なわれる動作の第１の表示を含んでおり、前記プローブは、ソースノードとして機能する前記第３のノードによってターゲットノードとして機能する前記第２のノードに送信される要求に応答して、前記ターゲットノードによって生成され、前記第１のノードはプローブに応答してプローブ応答を生成するよう構成され、前記第１のノードは、プローブ内の第２の表示に応答してターゲットノードおよびソースノードのいずれの１つがプローブ応答を受取るべきかを選択するよう構成され、前記第１のノードは、前記第２の表示に応答して、ターゲットノードまたはソースノードのいずれか１つにプローブ応答を送信するよう構成されており、前記要求が読出ならば、前記第２の表示は前記ソースノードを示し、前記要求が書込ならば、前記第２の表示は前記ターゲットノードを示す、装置。
【請求項２】
前記プローブは、ターゲットノードを示す第１のノード番号と、ソースノードを示す第

２のノード番号とを含み、前記第１のノードは、プローブ内の前記第２の表示に応答するプローブ応答を経路制御するために前記第１のノード番号および前記第２のノード番号のいずれか１つを選択するよう構成される、請求項１に記載の装置。

【請求項３】

前記プローブ応答は、前記第１のノードがプローブによって識別されたデータの修正されたコピーをストアしている場合に、データを伴いうる、請求項１に記載の装置。

【請求項４】

前記プローブは、前記修正されたコピーがプローブに応答して送信されるべきか否かを示すデータ移動表示を含み、前記プローブ応答は、データ移動表示が修正されたデータを送信すべきことを示す場合に、データを伴いうる、請求項３に記載の装置。

【請求項５】

前記プローブは、前記第１のノードがデータをキャッシュしている場合における、前記第１のノードにおいて前記プローブによって識別されるデータの次の状態を識別するネクストステートフィールドを含み、前記第１のノードは、ネクストステートフィールドに応答して前記第１のノードにおけるデータの状態を変更するよう構成される、請求項１に記載の装置。

【請求項６】

方法であって、

第１のノードにおいてプローブを受取るステップを含み、前記第１のノードは、第２のノードおよび第３のノードに結合されており、各前記ノードはメモリに結合されており、前記第１のノードはキャッシュを含み、前記プローブは、ソースノードとして機能する前記第３のノードによってターゲットノードとして機能する前記第２のノードに送信される要求に応答して、前記ターゲットノードによって生成され、前記プローブは、前記第１のノードが、前記メモリから読み出されたデータのキャッシュブロックのコピーを前記キャッシュにストアしているかどうかを判断する要求であり、前記キャッシュブロックが前記第１のノードにストアされている場合に前記第１のノードによって行なわれる動作の第１の表示を含んでおり、

方法はさらに、

前記プローブに応答して前記第１のノードにおいてプローブ応答を生成するステップと、

プローブ内の第２の表示に応答して、ターゲットノードおよびソースノードのいずれの１つがプローブ応答を受取るべきかを選択するステップと、

前記第２の表示に応答して、ターゲットノードまたはソースノードのいずれか１つにプローブ応答を送信するステップとを含み、前記要求が読出ならば、前記第２の表示は前記ソースノードを示し、前記要求が書込ならば、前記第２の表示は前記ターゲットノードを示す、方法。

【請求項７】

前記プローブは、ターゲットノードを示す第１のノード番号と、ソースノードを示す第２のノード番号とを含み、方法はさらに、

プローブ内の前記第２の表示に応答するプローブ応答を経路制御するために前記第１のノード番号および前記第２のノード番号のいずれか１つを選択するステップを含む、請求項６に記載の方法。

【請求項８】

前記第１のノードがプローブによって識別されたデータの修正されたコピーをストアしている場合に、プローブ応答とともにデータを送信しうるステップをさらに含む、請求項６に記載の方法。

【請求項９】

前記プローブは、前記修正されたコピーがプローブに応答して送信されるべきか否かを表示するデータ移動表示を含み、前記データを送信しうるステップは、データ移動表示が修正されたコピーを送信すべきことを示す場合に、プローブ応答とともにデータを送信し

うるものである、請求項８に記載の方法。
【請求項１０】
前記プローブは、前記第１のノードがデータをキャッシュしている場合における、前記第１のノードにおいて前記プローブによって識別されるデータの次の状態を識別するネクストステートフィールドを含み、方法はさらに、
ネクストステートフィールドに応答して前記第１のノードにおけるデータの状態を変更するステップを含む、請求項６に記載の方法。
【請求項１１】
第１のノードに結合されて、情報を転送するためのトランザクションが向けられるターゲットノードとして機能するノードであって、前記第１のノードはキャッシュを含み、各前記ノードにはメモリが結合されており、
前記ターゲットノードに結合されて、トランザクションを開始するソースノードとして機能するノードによって生成される要求を受信するための手段を含み、
前記受信するための手段は、前記要求に応答してプローブを生成するように構成されており、
前記プローブは、当該プローブを受信する第１のノードが、前記メモリから読み出されたデータのキャッシュブロックのコピーを前記キャッシュにストアしているかどうかを判断するための要求であり、前記キャッシュブロックが前記第１のノードにストアされている場合に前記第１のノードによって行なわれる動作の第１の表示を含んでおり、前記プローブは、当該プローブへの応答を受信するための受信ノードを指定するための手段によって生成される第２の表示を含み、
前記受信ノードは、前記トランザクションが書込であることに応答してターゲットノードとなり、
前記受信ノードは、前記トランザクションが読出であることに応答してソースノードとなる、ノード。
【請求項１２】
前記プローブは、当該プローブとしてパケットを識別するコマンドフィールドを有するパケットを含み、
前記第２の表示は、前記コマンドフィールドに含まれる、請求項１１に記載のノード。
【請求項１３】
前記プローブは、前記ターゲットノードを識別するターゲットノードフィールドと、前記ソースノードを識別するソースノードフィールドとを有する第１のパケットを含む、請求項１１に記載のノード。
【請求項１４】
前記要求を受信するための手段は、前記キャッシュブロックがストアされるメモリと通信するように構成されたメモリコントローラを含み、
前記メモリコントローラは、前記メモリにアクセスする要求を選択することに応答して、前記プローブを生成するように構成されている、請求項１１に記載のノード。
【発明の詳細な説明】
【０００１】
【発明の背景】
１．技術分野
この発明は広くはコンピュータシステムに関し、より特定的には、マルチプロセッシング演算環境を達成するためのメッセージ通信方式に関する。
【０００２】
２．関連技術分野の背景
クメール・Ａ（Kumar A）他による「共有メモリハイパーキューブマイクロプロセッサのための、効率的でスケーラブルなキャッシュコヒーレンス方式（Efficient and Scalable Cache Coherence Schemes for Shared Memory Hypercube Microprocessor）」Proceedings of the Supercomputing Conference、US Los Alamitos、IEEE、Vol.Conf.7. 1994、pp

.498-507は、前掲の請求項１のプリアンブルに記載の特徴を有しキャッシュコヒーレンシが維持されるコンピュータシステムと、前掲の請求項１０のプリアンブルの特徴を有しコンピュータシステム内でキャッシュコヒーレンシを維持するための方法とを開示する。
ＥＰ－Ａ－０　８１７　０７６は、ローカルおよびグローバルアドレススペースと、多数のアクセスモードとを用いる、マルチプロセッシングコンピュータシステムを開示する。ノード内のプロセッサは、ノード間通信を要求するトランザクションを開始し得る。要求するノードからホームノードへ要求が送られると、ホームノードは、要求されたデータのキャッシュコピーを有する従属ノードのいずれかに、読出および／または無効化要求を送る。システムは、キャッシュコヒーレンシを維持するためのグローバルコヒーレンシプロトコルを実現化する。
一般的には、パーソナルコンピュータ（ＰＣ）およびその他の種類のコンピュータシステムは、メモリにアクセスするために共用バスシステムを中心に設計されてきた。１つ以上のプロセッサおよび１つ以上の入力／出力（Ｉ／Ｏ）装置が、共用バスを介してメモリに結合される。Ｉ／Ｏ装置は、共用バスとＩ／Ｏ装置との間の情報の転送を管理するＩ／Ｏブリッジを介して共用バスに結合される場合もある一方、プロセッサは典型的には、直接共用バスに結合されるか、またはキャッシュ階層構造を介して共用バスに結合される。

【０００３】
残念ながら、共用バスシステムはいくつかの欠点を有する。たとえば、共用バスには多数の装置が装着されることから、バスは典型的には比較的低い周波数で動作される。多数の装着物は、バス上で信号を駆動する装置に容量的な高負荷をもたらし、多数の装着点は、比較的複雑な高周波数に対する伝送ラインモデルをもたらす。したがって、周波数は低く留められ、共用バスで使用できる帯域幅も同様に比較的低い。低帯域幅は共用バスに付加的な装置を装着するのに障壁になり得るが、これは使用できる帯域幅によって性能が制限されるおそれがあるためである。

【０００４】
共用バスシステムの他の欠点は、より多くの装置に対するスケーラビリティの欠如である。上述のように、帯域幅が固定される（そして、もし付加的な装置の追加によってバスの動作可能周波数が減じられると、減少し得る）。バスに（直接的にまたは間接的に）装着された装置の帯域幅要件が、一旦バスの利用可能な帯域幅を超えると、装置はバスへのアクセスを試みたときにしばしばストールし得る。全体的な性能が減じられるであろう。

【０００５】
上述の問題のうち１つまたはいくつかには、分散メモリシステムを用いて対処し得る。分散メモリシステムを用いるコンピュータシステムは、複数のノードを含む。２つ以上のノードがメモリに接続され、それらのノードは何らかの好適な相互接続を用いて相互接続される。たとえば、ノードの各々は専用ラインを用いて他のノードに互いに接続されることができる。これに代えて、ノードの各々は固定された数の他のノードに接続され、トランザクションは第１のノードから１つ以上の中間ノードを介して、第１のノードに直接接続されていない第２のノードに経路制御されてもよい。メモリアドレス空間は、各々のノードのメモリにわたって割当てられる。

【０００６】
ノードはさらに、１つ以上のプロセッサを含み得る。プロセッサは典型的には、メモリから読出したデータのキャッシュブロックをストアするキャッシュを含む。さらに、ノードはプロセッサの外部の１つ以上のキャッシュを含み得る。プロセッサおよび／またはノードは、他のノードからアクセスされるキャッシュブロックをストアし得るために、ノード内のコヒーレンシを維持するための機構が望まれる。

【０００７】
【発明の開示】
上に概略を述べた問題は、ここに説明するコンピュータシステムによってほとんどが解決される。このコンピュータシステムは多数の処理ノードを含むことができ、そのうち１つ以上は分散メモリシステムを形成し得る別々のメモリに結合し得る。処理ノードはキャッ

シュを含むことができ、コンピュータシステムは、キャッシュと分散メモリシステムとの間のコヒーレンシを維持し得る。特に、このコンピュータシステムは柔軟なプローブコマンド／応答経路制御方式を実現化し得る。

【０００８】

一実施例においては、この方式はプローブコマンド内の表示を用い、これはプローブ応答を受信する受信ノードを識別する。一般的にはプローブコマンドとは、キャッシュブロックがノード内にストアされているか判断するためのそのノードへの要求であり、かつ、キャッシュブロックがそのノードにストアされていた場合にそのノードが行なうべき動作の表示である。プローブ応答は、動作が行なわれたことを表示し、かつ、キャッシュブロックがノードによって変更されていた場合にはデータの送信を含み得る。送られたコマンドに依存してプローブ応答を異なった受信ノードに経路制御することへ柔軟性を与えることにより、コヒーレンシの維持を比較的効率的な態様で（たとえば最少の数のパケット送信を処理ノード間で用いて）行ない得る一方で、それでもコヒーレンシが維持されることを確実にする。

【０００９】

たとえば、ターゲットまたはトランザクションのソースがトランザクションに対応するプローブ応答を受取るべきことを表示するプローブコマンドを含み得る。プローブコマンドは、トランザクションのソースを読出トランザクションに対する受信ノードとして特定し得る（それによりダーティデータをストアするノードからダーティデータがソースノードに引渡される）。一方で、（データがトランザクションのターゲットノードでメモリ内に更新される）書込トランザクションに対しては、プローブコマンドはトランザクションのターゲットを受信ノードとして特定し得る。このようにして、ターゲットはいつ書込データをメモリにコミットするか判断し、書込データとマージされるべきいかなるダーティデータをも受取ることができる。

【００１０】

概略的には、コンピュータシステムが企図される。コンピュータシステムは、第１の処理ノードと第２の処理ノードとを含み得る。第１の処理ノードは、要求を送信することによりトランザクションを開始するよう構成し得る。第２の処理ノードは、第１の処理ノードからの要求を受取るよう結合されて、要求に応答してプローブを生成するよう構成し得る。プローブは、プローブに対する応答を受けるための受信ノードを指定する表示を含み得る。さらに、第２の処理ノードはトランザクションのタイプに応答して、表示を生成するよう構成し得る。

【００１１】

コンピュータシステム内でのコヒーレンシを維持するための方法もまた企図される。ソースノードからの要求はターゲットノードに送信される。プローブは要求に応答してターゲットノード内で生成される。プローブ内の表示を介して、プローブに対する応答のための受信ノードが指定される。プローブに対するプローブ応答は、受信ノードに経路制御される。

【００１２】

この発明の他の目的と利点とは、以下の詳細な説明を読み、添付の図面を参照することにより、より明らかとなるであろう。

【００１３】

この発明はさまざまな変形と代替形に対処するものであるが、その特定の実施例を図面において例示の目的で示し、以下に詳述する。しかしながら、図面とその詳細な説明とは開示される発明を特定の形に限定することを意図せず、反対に、すべての変形、等価物、および代替例は前掲の特許請求の範囲に規定されるこの発明の範囲に入ることを意図する。

【００１４】

【発明の実施の形態】

例示的なコンピュータシステムの実施例

図１は、マルチプロセッシングコンピュータシステム１０の一実施例を示す。他の実施例

が可能であり企図される。図１の実施例においては、コンピュータシステム１０はいくつかの処理ノード１２Ａ、１２Ｂ、１２Ｃ、および１２Ｄを含む。処理ノードの各々は、処理ノード１２Ａ－１２Ｄにそれぞれ含まれるメモリコントローラ１６Ａ－１６Ｄを介して、それぞれのメモリ１４Ａ－１４Ｄに結合される。さらに、処理ノード１２Ａ－１２Ｄは、処理ノード１２Ａ－１２Ｄの間の通信のために用いるインターフェイスロジックを含む。たとえば、処理ノード１２Ａは、処理ノード１２Ｂと通信するためのインターフェイスロジック１８Ａと、処理ノード１２Ｃと通信するためのインターフェイスロジック１８Ｂと、さらに別の処理ノード（図示せず）と通信するための第３のインターフェイスロジック１８Ｃを含む。同様に、処理ノード１２Ｂはインターフェイスロジック１８Ｄ、１８Ｅ、および１８Ｆを含み、処理ノード１２Ｃはインターフェイスロジック１８Ｇ、１８Ｈ、および１８Ｉを含み、処理ノード１２Ｄはインターフェイスロジック１８Ｊ、１８Ｋ、および１８Ｌを含む。処理ノード１２Ｄは、インターフェイスロジック１８Ｌを介して結合されてＩ／Ｏブリッジ２０と通信する。他の処理ノードは同様の様式で他のＩ／Ｏブリッジと通信し得る。Ｉ／Ｏブリッジ２０はＩ／Ｏバス２２に結合される。

【００１５】

処理ノード１２Ａ－１２Ｄは、ノード通信相互処理のためのパケットベースのリンクを実現化する。この実施例においては、リンクは単方向ラインの組として実現化される（たとえば、ライン２４Ａはパケットを処理ノード１２Ａから処理ノード１２Ｂへ送信するために用いられ、ライン２４Ｂはパケットを処理ノード１２Ｂから処理ノード１２Ａへ送信するために用いられる）。ライン２４Ｃ－２４Ｈの他の組は、図１に示すようにパケットを他の処理ノード間で通信するために用いられる。リンクは、処理ノード間の通信のためにキャッシュコヒーレント様式で動作するか、または処理ノードとＩ／Ｏブリッジとの間の通信のために非コヒーレント様式で動作し得る。一方の処理ノードから他方へ送信されるべきパケットは、１つ以上の中間ノードを通過し得ることに留意されたい。たとえば、処理ノード１２Ａから処理ノード１２Ｄに送信されるパケットは、図１に示すように、処理ノード１２Ｂまたは処理ノード１２Ｃのいずれかを通過し得る。いかなる好適な経路制御アルゴリズムをも用い得る。コンピュータシステム１０の他の実施例は、図１に示す実施例よりもより多いかまたはより少ない処理ノードを含み得る。

【００１６】

処理ノード１２Ａ－１２Ｄは、メモリコントローラとインターフェイスロジックに加えて、１つ以上のプロセッサを含み得る。概略的には、処理ノードは少なくとも１つのプロセッサを含み、選択により、メモリおよび所望の他のロジックとの間で通信するためのメモリコントローラを含む。この開示では「ノード」という用語も使用し得る。ノードという用語は、「処理ノード」を意味することを意図する。

【００１７】

メモリ１４Ａ－１４Ｄは、何らかの好適なメモリ装置を含み得る。たとえば、メモリ１４Ａ－１４Ｄは、１つ以上のＲＡＭＢＵＳ　ＤＲＡＭ（ＲＤＲＡＭ）、シンクロナスＤＲＡＭ（ＳＤＲＡＭ）、スタティックＲＡＭなどを含み得る。コンピュータシステム１０のアドレス空間は、メモリ１４Ａ－１４Ｄの間で分割される。処理ノード１２Ａ－１２Ｄの各々はメモリマップを含むことができ、該メモリマップを用いて、どのアドレスがどのメモリにマッピングされているかを判断し、よって、ある特定のアドレスに対するメモリ要求がどの処理ノード１２Ａ－１２Ｄに経路制御されるべきかを判断する。一実施例においては、コンピュータシステム１０内のアドレスに対するコヒーレンシ点は、アドレスに対応するバイトをストアしているメモリに結合された、メモリコントローラ１６Ａ－１６Ｄである。言い換えると、メモリコントローラ１６Ａ－１６Ｄは、対応するメモリ１４Ａ－１４Ｄへのメモリアクセスの各々を、キャッシュコヒーレントな様式で起こることを確実にすることを担当している。メモリコントローラ１６Ａ－１６Ｄは、メモリ１４Ａ－１４Ｄにインターフェイスするための制御回路を含み得る。さらに、メモリコントローラ１６Ａ－１６Ｄは、メモリ要求を待ち行列として管理するための、要求キューを含み得る。

【００１８】

一般的には、インターフェイスロジック１８Ａ－１８Ｌは、リンクからのパケットを受取り、かつリンクに送信されるべきパケットをバッファするための、さまざまなバッファを含み得る。コンピュータシステム１０は、パケットを転送するための好適なフロー制御であればいずれでも用い得る。たとえば一実施例においては、インターフェイスロジック１８の各々は、そのインターフェイスロジックが接続されたリンクの他端のレシーバ内に、いくつかの各種のバッファのカウントをストアする。インターフェイスロジックは、受信インターフェイスロジックがパケットをストアするフリーのバッファを有さない限り、パケットを送信しない。パケットを次に経路制御することにより受信バッファが解放されると、受信インターフェイスロジックは送信インターフェイスロジックにメッセージを送り、バッファが解放されたことを示す。そのような機構は、「クーポンに基づく」システムと呼べる。

【００１９】

次に図２は、処理ノード１２Ａおよび１２Ｂのブロック図を示し、それらの間のリンクの１実施例を詳細に例示する。。図２の実施例においては、ライン２４Ａは、クロックライン２４ＡＡと、制御ライン２４ＡＢと、制御／アドレス／データバス２４ＡＣとを含む。同様に、ライン２４Ｂは、クロックライン２４ＢＡと、制御ライン２４ＢＢと、制御／アドレス／データバス２４ＢＣとを含む。

【００２０】

クロックラインは、制御ラインおよび制御／アドレス／データバスに対するサンプルポイントを示すクロック信号を送信する。特定の一実施例においては、データ／制御ビットはクロック信号のエッジの各々（すなわち立上がりエッジおよび立下がりエッジ）で送信される。したがって、クロックサイクルごとに、ラインごとに２つのデータビットを送信し得る。ラインごとに１ビットを送信するために使用される時間は、ここでは「ビット時間」と呼ぶ。上述の実施例は、クロックサイクルごとに２つのビット時間を含む。パケットは２つ以上のビット時間で送信し得る。制御／アドレス／データバスの幅に依存して、多数のクロックラインを用い得る。たとえば３２ビット制御／アドレス／データバスに対しては２つのクロックラインを用い得る（制御／アドレス／データバスの半分では一方のクロックラインが参照され、残りの半分の制御／アドレス／データバスと制御ラインとでは他方のクロックラインが参照される）。一般的には、「パケット」とは２つの処理ノード１２Ａ－１２Ｄの間の通信である。１つ以上のパケットが「トランザクション」を形成し得るが、これは一方の処理ノードから他方への情報の転送である。トランザクションを形成するパケットは、ソースノード（転送を要求する開始するノード）からのターゲットノード（トランザクションが向けられるノード）へのトランザクションを開始する要求パケットと、コヒーレンシを維持するために他の処理ノード間で送信されるパケットと、データパケットと、トランザクションを終了させる肯定応答パケットとを含み得る。

【００２１】

制御ラインは、制御／アドレス／データバスに送信されたデータが、ビット時間の制御パケットか、またはビット時間のデータパケットであるかを示す。制御ラインはアサートされて制御パケットを示し、デアサートされてデータパケットを示す。ある制御パケットは、後にデータパケットが続くことを示す。データパケットは、対応する制御パケットのすぐ後に続き得る。一実施例においては、他の制御パケットがデータパケットの送信に割込むおそれがある。そのような割込は、データパケットの送信の間に制御ラインをいくつかのビット時間アサートし、かつ制御ラインがアサートされている間にビット時間の制御パケットを送信することにより行なわれる可能性がある。データパケットに割込む制御パケットは、データパケットが後に続くことを示さないおそれがある。

【００２２】

制御／アドレス／データバスは、データ／制御ビットを送信するための１組のラインを含む。一実施例においては、制御／アドレス／データバスは、８、１６、または３２のラインを含み得る。処理ノードまたはＩ／Ｏブリッジの各々は、設計選択にしたがってサポートされる数のラインのうちのいずれかを用い得る。他の実施例は、所望により他のサイズ

の制御／アドレス／データバスをサポートし得る。

【００２３】

一実施例によると、コマンド／アドレス／データバスラインおよびクロックラインは、反転データを担持し得る（すなわち、論理１はライン上の低電圧として表わされ、論理０が高電圧として表わされる）。これに代えて、これらのラインは非反転データを担持してもよい（論理１はライン上の高電圧として表わされ、論理０は低電圧として表わされる）。

【００２４】

図３から図６は、コンピュータシステム１０の一実施例に従って用いられる、例示的なパケットを示す。図３から図５は制御パケットを示し、図６はデータパケットを示す。他の実施例は、所望により異なったパケット定義を用い得る。パケットの各々は、「ビット時間」の見出しの下に列挙される一連のビット時間で示される。パケットのビット時間は、リストされたビット時間順序に従って送信される。図３から図６は、８ビット制御／アドレス／データバス実現化のためのパケットを示す。したがって、ビット時間の各々は、７から０まで番号が付与された８つのビットを含む。図中、いずれの値も付与されていないビットは、所与のパケットのために予約されているか、またはパケット特定情報を送信するために用いられるかのいずれかであり得る。

【００２５】

図３は情報パケット（infoパケット）３０を示す。情報パケット３０は、８ビットリンク上の２つのビット時間を含む。この実施例においては、コマンド符号化はビット時間１の間に送信され、かつ６ビットを含む。図４、および図５に示す他方の制御パケットの各々は、ビット時間１の間に同じビット位置においてコマンド符号化を含む。メッセージがメモリアドレスを含まないときに、情報パケット３０を用いてこのメッセージを処理ノード間で送信し得る。

【００２６】

図４はアドレスパケット（addressパケット）３２を示す。アドレスパケット３２は、８ビットリンク上の８つのビット時間を含む。コマンド符号化は、宛先ノード番号の一部と併せて、ビット時間１の間に送信される。宛先ノード番号の残りとソースノード番号とは、ビット時間２の間に送信される。ノード番号はコンピュータシステム１０内の処理ノード１２Ａ－１２Ｄのうちの１つを明確に識別し、かつ用いられてパケットをコンピュータシステム１０を介して経路制御する。さらに、パケットのソースは、ビット時間２および３の間に送信されるソースタグを割当て得る。ソースタグは、ソースノードによって開始される特定のトランザクションに対応するパケットを識別する（すなわち、特定のトランザクションに対応するパケットの各々は、同一のソースタグを含む）。ビット時間４から８までを用いて、トランザクションによって影響されたメモリアドレスを送信する。アドレスパケット３２は、トランザクション（たとえば、読出または書込トランザクション）を開始するのに用いられるだけでなく、トランザクションを実行する過程において、トランザクションによって影響を受けるメモリアドレスを担持するコマンドについて、コマンドを送信し得る。

【００２７】

図５は、応答パケット（responseパケット）３４を示す。応答パケット３４は、コマンド符号化、宛先ノード番号、ソースノード番号、およびアドレスパケット３２と同様のソースタグを含む。さまざまな種類の応答パケットは付加的な情報を含み得る。たとえば、読出応答パケットは、後に続くデータパケットで提供される読出データの量を示し得る。プローブ応答は、要求されたキャッシュブロックに対してヒットが検出されたかどうかを示し得る。一般的に、応答パケット３４は、トランザクションを行なう間にトランザクションによって影響されるメモリアドレスの送信を必要としないコマンドに対して用いられる。さらに、応答パケット３４を用いて肯定応答パケットを送信してトランザクションを終了させることができる。

【００２８】

図６は、データパケット（dataパケット）３６を示す。データパケット３６は、図６の実

施例において、８ビットリンク上の８つのビット時間を含む。データパケット３６は、送信されるデータの量に依存して、異なった数のビット時間を含み得る。たとえば、一実施例においてはキャッシュブロックは６４バイトを含み、したがって８ビットリンク上の６４のビット時間を含む。他の実施例では、キャッシュブロックのサイズを所望により別に定義し得る。さらに、キャッシュ不可能な読出および書込に対しては、キャッシュブロックサイズよりも小さなサイズでデータを送信し得る。キャッシュブロックサイズより小さなデータを送信するためのデータパケットは、より少ないビット時間を用いる。

【００２９】

図３から図６は、８ビットリンクのためのパケットを示す。１６および３２ビットリンクのためのパケットは、図３から図６に示す連続的なビット時間を連結することにより形成し得る。たとえば、１６ビットリンク上のパケットのビット時間１は、８ビットリンク上のビット時間１および２の間に送信される情報を含み得る。同様に、３２ビットリンク上のパケットのビット時間１は、８ビットリンク上のビット時間１から４までの間に送信される情報を含み得る。以下の式（１）および式（２）は、８ビットリンクによるビット時間における、１６ビットリンクのビット時間１および３２ビットリンクのビット時間１の構成を示す。

【００３０】

【数１】

$$BT1_{16}[15:0] = BT2_{8}[7:0] \,||\, BT1_{8}[7:0] \quad \cdots\cdots (1)$$

$$BT1_{32}[31:0] = BT4_{8}[7:0] \,||\, BT3_{8}[7:0] \,||\, BT2_{8}[7:0] \,||\, BT1_{8}[7:0] \quad \cdots\cdots (2)$$

【００３１】

図７は、コンピュータシステム１０内のリンクの１つの例示的な実施例によって用いられるコマンドを示すテーブル３８を示す。他の実施例も可能であり企図される。テーブル３８は、コマンドの各々に割当てられたコマンド符号化を示すコマンド符号化列、コマンドの名前を示すコマンド列、およびどのコマンドパケット３０－３４がそのコマンドに対して用いられるかを示すパケットタイプ列を含む。

【００３２】

読出トランザクションは、ReadSized, RdBlk, RdBlkSまたはRdBlkModのコマンドのうち、１つを用いて開始される。サイズ指定された読出コマンドであるReadSizedは、キャッシュ不可能な読出のために、またはサイズの合ったキャッシュブロック以外のデータの読出のために用いられる。読出されるべきデータ量は、ReadSizedコマンドパケット内に符号化される。キャッシュブロックの読出には、以下の場合以外にRdBlkコマンドを用いることができる。すなわち、（ｉ）キャッシュブロックの書込可能なコピーを所望である場合。この場合はRdBlkModコマンドを用い得る。または（ii）キャッシュブロックのコピーを所望するが、ブロックを変更する意図があるとは分らない場合。この場合はRdBlkS コマンドを用いることができる。RdBlkSコマンドを用いて、ある種のコヒーレントな方式（たとえばディレクトリに基づくコヒーレントな方式）をより効率化できる。一般的に、適切な読出コマンドはソースから送信されて、キャッシュブロックに対応するメモリを有するターゲットノードへの読出トランザクションを開始する。ターゲットノード内のメモリコントローラは、システム内の他のノードにProbe/Srcコマンドを送信して、これらのノード内のキャッシュブロックの状態を変化させること、およびキャッシュブロックの更新されたコピーを含むノードにキャッシュブロックをソースノードに送らせることにより、コヒーレンシを維持する。Probe/Srcコマンドを受取ったノードの各々は、ProbeRespパケットをソースノードに送信する。プローブされたノードが読出データの更新されたコピー（

すなわちダーティデータ）を有していれば、そのノードはRdResponseパケットおよびダーティデータを送信する。ダーティデータを送信するノードはまた、ターゲットノードによる要求された読出データの送信をキャンセルしようと試みて、ターゲットノードにMemCancelパケットをも送信し得る。さらに、ターゲットノード内のメモリコントローラは、RdResponseパケットとその後に続くデータパケット内のデータとを用いて要求された読出データを送信する。ソースノードがプローブされたノードからRdResponseパケットを受取れば、その読出されたデータが用いられる。そうでなければ、ターゲットノードからのデータが用いられる。一旦ソースノードにおいてプローブ応答および読出されたデータの各々が受取られると、ソースノードはトランザクションの終了の肯定応答として、ターゲットノードにSrcDone応答パケットを送信する。

【００３３】

書込トランザクションはWrSizedまたはVicBlkコマンドを用いて開始され、その後に関連のデータパケットが続く。WrSizedコマンドは、キャッシュ不可書込またはサイズの合わないキャッシュブロックのデータの書込に用いられる。WrSizedコマンドに対するコヒーレンシの維持のために、ターゲットノード内のメモリコントローラはシステム内の他のノードの各々にProbe/Tgtコマンドを送信する。Probe/Tgtコマンドに応答して、プローブされたノードの各々はターゲットノードにProbeRespパケットを送信する。もしプローブされたノードがダーティデータをストアしていれば、プローブされたノードはRdResponseパケットおよびダーティデータで応答する。このようにして、WrSizedコマンドによって更新されたキャッシュブロックは、メモリコントローラに返されて、WrSizedコマンドによって提供されるデータとマージされる。メモリコントローラは、プローブされたノードの各々からプローブ応答を受取ると、ソースノードにTgtDoneパケットを送信してトランザクションの終了の肯定応答を提供する。ソースノードはSrcDone応答パケットで応答する。

【００３４】

ノードによって変更され、かつノード内のキャッシュで置き換えられたヴィクティムキャッシュブロックは、VicBlkコマンドを用いてメモリに返される。VicBlkコマンドに対してはプローブは必要ではない。したがって、ターゲットメモリコントローラがビクティムブロックデータをメモリにコミットする準備ができたとき、ターゲットメモリコントローラはビクティムブロックのソースノードにTgtDoneパケットを送信する。ソースノードは、データがコミットされるべきことを示すSrcDoneパケットか、またはデータが（たとえば介入するプローブに応答して）VicBlkコマンドの送信とTgtDoneパケットの受信との間で無効化されたことを示すMemCancelパケットのいずれかで応答する。

【００３５】

ソースノードにストアされた書込不可能状態のキャッシュブロックへの書込許可を得るために、ソースノードはChangetoDirtyパケットを送信し得る。ChangetoDirtyコマンドによって開始されるトランザクションは、ターゲットノードがデータを返さないという点を除いて、読出と同様に動作し得る。もしソースノードがキャッシュブロック全体を更新する意図があるのであれば、ValidateBlkコマンドを用いて、ソースノードによってストアされていないキャッシュブロックへの書込許可を得ることができる。そのようなトランザクションに対してはソースノードへデータは転送されないが、それ以外では読出トランザクションと同様に動作する。

【００３６】

InterruptBroadcast、InterruptTarget、およびIntrResponseパケットを用いて、それぞれ割込をブロードキャストし、特定のターゲットノードに割込を送り、割込に応答し得る。CleanVicBlkコマンドを用いて、（たとえば、ディレクトリに基づくコヒーレントな方式のために）クリーンなヴィクティムブロックがノードから捨てられたことを伝えることができる。TgtStartコマンドはターゲットによって用いられて、（たとえば、後のトランザクションの順序付けのために）トランザクションが開始したことを示す。エラーコマンドを用いて、エラー表示を送信する。

【００３７】
プローブ／プローブ応答経路制御
図８は、プローブパケット４０の一実施例のブロック図を示す。異なった態様でプローブパケットを構成し、代替的な、同様の、または代用の情報を有する他の実施例も可能であり企図される。プローブパケット４０は、図４に示すアドレスパケット３２の１種である。図８に示すように、プローブパケット４０はコマンドフィールド（図８におけるＣＭＤ［５：０］）、ターゲットノードフィールド（図８におけるTgtNode［１：０］およびTgtNode［３：２］）、ソースノードフィールド（図８におけるSrcNode［３：０］）、ソースタグフィールド（図８におけるSrcTag［１：０］およびSrcTag［６：２］）、データ移動フィールド（図８におけるＤＭ）、ネクストステートフィールド（NextState［１：０］）、およびアドレスフィールド（図８におけるビット時間４－８にわたるAddr［３９：０］）を含む。
【００３８】
コマンドフィールドは符号化されてパケット４０をプローブパケットとして識別する。たとえば、図７に示すProbe/SrcおよびProbe/Tgtに対する符号化を用い得る。一般的には、トランザクションのターゲットノードはプローブコマンドを生成して、トランザクションによって影響されるキャッシュブロックのコヒーレンシを維持する。開始されるトランザクションの種類に基づいて、プローブコマンドの組の中の１つがターゲットノードによって選択される。選択されたプローブコマンドは、プローブコマンドに対する応答のための受信ノードを識別する。多数の使用できる受信ノードのうちの１つに、柔軟にプローブ応答を経路制御することにより、（予め定められた受信ノードにプローブ応答を経路制御することとは対照的に）、コヒーレンシの維持は（たとえば、ノード間で送信されるパケットの数の観点から）効率的であって、正確な結果を導く態様で行なうことができる。
【００３９】
たとえば、キャッシュブロックをトランザクションのソースノードに転送させるトランザクションは、プローブ応答（データを転送する応答を含む）をソースノードに向けることにより、コヒーレンシを維持し得る。ソースノードは、他のノード（プローブされたノード）の各々からの応答とターゲットノードからの応答とを待ち、次いでノード内に送信されたキャッシュブロックを確立し、かつターゲットノードに肯定応答パケットを送信してトランザクションを終了させる。トランザクションを終了させる前にプローブされたノードからのプローブ応答を待つことにより、ノードの各々は終了の前にそのトランザクションに対する正しいコヒーレンシ状態を確立し得る。一方、WrSizedトランザクション（キャッシュブロックに満たない更新をする）は、WrSizedコマンドと関連のデータとをターゲットノードに送信することにより開始される。ターゲットノードは、ターゲットノードのメモリにデータをコミットするために、ターゲットノードはソースノードの代わりにプローブ応答を受信し得る。特に、WrSizedトランザクションによって更新されたバイトを含むキャッシュブロックがプローブノード内でダーティであれば、ターゲットノードはプローブに応答してダーティなデータを受信し、そのデータをWrSizedコマンドに関連のデータとマージし得る。一旦プローブ応答が受取られると、書込動作によって命令されるコヒーレンシ状態はプローブされたノード内で確立され、書込データはメモリにコミットし得る。したがってこの実施例においては、２つの種類のプローブコマンドがサポートされ、どのノードがプローブ応答を受取るべきであるかをプローブされたノードに対して示す（たとえばソースまたはターゲットノード）。他の実施例は、所望により、付加的な受信ノードを示す付加的なプローブコマンドをサポートし得る。
【００４０】
１つの例示的な実施例においては、ターゲットノード内のメモリコントローラは処理されるべき要求のキューからトランザクションを選択することができ、かつその選択に応答してプローブコマンドを生成し得る。プローブコマンドはコンピュータシステム内のプローブされたノードにブロードキャストされることができる。コマンドフィールドは、プローブ応答がターゲットノードまたはソースノードに経路制御されるべきかどうかを示す。た

とえば、図７に示すテーブル３８において例示するように、コマンドフィールドのビット０は受信ノードを示し得る（バイナリ０はソースノードを、バイナリ１はターゲットノードを示す）。ターゲットノードフィールドはトランザクションのターゲットノードを識別し、ソースノードフィールドはトランザクションのソースノードを識別する。プローブコマンドもまたターゲットノード内のキャッシュにブロードキャストされ、ターゲットノードはプローブ応答を提供し得ることに留意されたい。言い換えると、ターゲットノードもまたプローブされたノードであり得る。

【００４１】

さらに、データ移動フィールド（一実施例においては、たとえば１ビット）を用いて、プローブされたノード内でキャッシュブロックがダーティであった場合にプローブに応答してデータが返されるべきかどうかを表示し得る。データ移動フィールドがいずれの移動も示さなければ（たとえばクリアな状態）、ダーティデータを持つノードはプローブ応答を宛先指定された受信ノードに戻すが、ダーティデータを受信ノードに送信しない。データ移動フィールドが移動を示せば（たとえばセットされた状態）、ダーティデータを持つノードは、ダーティデータを含む読出応答をソースノードに返す。ネクストステートフィールドは、（プローブされたノードがキャッシュブロックのコピーをストアしている場合）プローブされたノードにおいてキャッシュブロックの次の状態がどうなるかを示す。一実施例においては、ネクストフィールドの符号化は以下のテーブル１に示すとおりである。

【００４２】

一実施例によると、以下のテーブル２は例示的なトランザクションタイプと対応のプローブコマンドを示す。

【００４３】

【表１】

テーブル１：ネクストステートフィールド符号化

符号化	意味
00	変化なし
01	共用： クリーン→共用 ダーティ→共用／ダーティ
10	無効
11	予約

テーブル２：トランザクションタイプおよびプローブコマンド

トランザクションタイプ	プローブコマンド	ネクストステート	データ移動
ReadSized	Probe/Src	変化なし(00)	Y(1)
Block Reads	Probe/Src	共用(01)	Y(1)
ChangetoDirty	Probe/Src	無効(10)	Y(1)
ValidateBlk	Probe/Src	無効(10)	N(0)
WrSized	Probe/Tgt	無効(10)	Y(1)
VicBlk	None	-	-

【００４４】

一般的には、「プローブ」または「プローブコマンド」という用語は、プローブノードに

対する要求を指し、これによりプローブによって（たとえばアドレスフィールドを介して）定義されるキャッシュブロックがプローブされたノードによってストアされているかどうか判断し、かつプローブされたノードからの所望の応答を示す。応答は、キャッシュブロックに対する異なったコヒーレンシ状態を確立すること、および／またはキャッシュブロックを受信ノードに転送することを含む。プローブされたノードは「プローブ応答」を用いて応答するが、これは所望のコヒーレンシ状態変化が行なわれたことを受信ノードに肯定応答する。プローブ応答は、プローブされたノードがダーティデータをストアしていた場合、データをも含み得る。

【００４５】

図９は、プローブ応答パケット４２の一実施例のブロック図を示す。応答パケットを異なった態様で構成し、代替的な、同様の、または代用の情報を有する他の実施例も可能であり企図される。プローブ応答パケット４２は、図５に示す応答パケット３４の１種である。図８に示すプローブパケットと同様に、プローブ応答パケット４２はコマンドフィールド（ＣＭＤ［５：０］）、ソースノードフィールド（SrcNode［３：０］）、およびソースタグフィールド（SrcTag［１：０］およびSrcTag［６：２］）を含み得る。さらに、プローブ応答パケット４２は、応答ノードフィールド（図９におけるRespNode［３：２］およびRespNode［１：０］）およびヒット表示（図９におけるHit）を含み得る。

【００４６】

プローブされたノードは、プローブパケット４０で受信されたアドレスによって識別されたキャッシュブロックに対して、そのキャッシュ内を探索する。キャッシュブロックのコピーを見出すと、プローブされたノードはネクストステートフィールドにより規定されていることにしたがってキャッシュブロックの状態を変化させる。さらに、キャッシュがキャッシュブロックのダーティコピーをストアしており、かつプローブパケット４０のデータ移動フィールドがキャッシュブロックが転送されるべきであることを示すと、プローブされたノードはキャッシュからダーティデータを読出す。

【００４７】

プローブされたノードは、プローブ応答パケット４２を以下の場合において指定された受信ノードに経路制御する。すなわち、（ｉ）ダーティデータがない場合、または（ii）ダーティデータおよびデータ移動フィールドがいずれの移動も示さない場合。この実施例においてより特定的には、プローブされたノードはプローブパケット４０のターゲットノードフィールド（指定された受信ノードがターゲットノードである場合）か、またはプローブパケット４０のソースノードフィールド（指定された受信ノードがソースノードである場合）のいずれかを読出し、結果として生じるノードＩＤをプローブ応答パケット４２の応答ノードフィールドにストアする。さらに、ヒット表示を用いて、プローブノードがキャッシュブロックのコピーを保持しているかどうかを示す。たとえば、ヒット表示は、セットされている場合に、プローブされたノードがキャッシュブロックのコピーを保持していることを示し、そしてクリアされている場合に、プローブされたノードがキャッシュブロックのコピーを保持していないことを示すビットを含み得る。ヒットビットは、以下の場合にクリアされている。すなわち、（ｉ）プローブされたノード内のキャッシュにおいてキャッシュブロックのコピーが見出されなかった場合、または（ii）プローブされたノードにおけるキャッシュ内でキャッシュブロックのコピーが見出されたが、プローブコマンドに応答して無効化されていた場合、である。

【００４８】

図７のテーブル３８に示す一実施例においては、プローブ応答パケット４２のコマンドフィールドはProbRespを示すことができ、さらにプローブ応答がProbe/Srcコマンドへの応答であるか、またはProbe/Tgtコマンドへの応答であるかをさらに示すことができる。より特定的には、コマンドフィールドのビット０はクリアされてプローブ応答がProbe/Srcコマンドへの応答であることを示し、セットされてプローブ応がはProbe/Tgtコマンドへの応答であることを示す。この表示は受信ノードによって用いられて、プローブ応答が受信ノード内のメモリコントローラに経路制御されるべきか（ビット０がセットされている

場合）、または受信ノード内のキャッシュに対するフィル受信ロジックに対して経路制御されるべきか（ビット０がクリアされている場合）を判断し得る。

【００４９】

図１０は、読出応答パケット４４の一実施例のブロック図を示す。読出応答パケットを異なった態様で構成し、代替的な、同様の、または代用の情報を有する他の実施例も可能であり企図される。応答パケット４４は図５に示す応答パケット３４の１種である。図９に示すプローブ応答パケットと同様に、読出応答パケット４４は、コマンドフィールド（ＣＭＤ［５：０］）、ソースノードフィールド（SrcNode［３：０］）、ソースタグフィールド（SrcTag［１：０］およびSrcTag［６：２］）、および応答ノードフィールド（RespNode［３：２］およびRespNode［１：０］）を含む。さらに、読出応答パケット４４は、カウントフィールド（図１０におけるCount）、タイプフィールド（図１０におけるType）およびプローブフィールド（図１０におけるPrb）を含む。

【００５０】

データ移動が要求されることを示すプローブコマンドに対してダーティデータのヒットを検出した、プローブされたノードは、読出パケット４４を用いてプローブ応答およびダーティデータを送信し得る。カウントフィールドを用いて送信されるデータの量を示し、タイプフィールドはカウントがバイトまたはクワッドワード（８バイト）で測定されるかを示す。プローブに対して応答するキャッシュブロック転送に対しては、カウントフィールドは８（バイナリ「１１１」として符号化される）を示し、タイプフィールドはクワッドワード（たとえば、一実施例においてはタイプフィールドにおいてセットされたビット）を示す。プローブフィールドを用いて、読出応答パケット４４がターゲットノードまたはプローブされたノードのいずれから送信されているかを示す。たとえば、プローブフィールドは、セットされている場合に読出応答パケット４４がプローブされたノードから転送されたことを示し、クリアされている場合に読出応答パケット４４がターゲットノードから送信されたことを示す、ビットを含み得る。したがって、プローブされたノードは読出応答パケット４４を用いる場合にプローブビットをセットする。

【００５１】

読出応答パケット４４は、データパケットが後に続くことを示すパケットタイプである。したがって、プローブされたノードは読出応答パケット４４の後にデータパケットでキャッシュブロックを送信する。プローブされたノードは、読出応答パケット４４を、上述のようにプローブ読出パケット４２を経路制御するのと同様の態様で経路制御する（たとえば、プローブされたノードは（指定された受信ノードがターゲットノードであれば）プローブパケット４０のターゲットノードフィールドを読出すか、または（指定された受信ノードがソースノードであれば）プローブパケット４０のソースノードフィールドを読出し、結果として生じるノードＩＤを読出応答パケット４４の応答ノードフィールドにストアする）。

【００５２】

図７のテーブル３８において示される一実施例においては、読出応答パケット４４のコマンドフィールドはRdResponseを示すことができ、さらに読出応答がProbe/Srcコマンドへの応答であるか、またはProbe/Tgtコマンドへの応答であるかを示し得る。より特定的には、コマンドフィールドのビット０はクリアされて読出応答がProbe/Srcコマンドへの応答であることを示し、セットされて読出応答がProbe/Tgtコマンドへの応答であることを示す。この表示を受信ノードによって用いて、読出応答が受信ノード内のメモリコントローラに経路制御されるべきか（ビット０がセットされている場合）、または受信ノード内のキャッシュに対するフィル受信ロジックに経路制御されるべきか（ビット０がクリアされている場合）を判断し得る。

【００５３】

図１１は、例示的な読出ブロックトランザクションに対応する１組のノードの間のパケットフローを示す図である。ソースノード５０、ターゲットノードメモリコントローラ５２、および１組のプローブされたノード５４Ａ－５４Ｎを示す。図１１にはパケットの（時

系列での）順序を左から右に示す。言い換えると、RdBlkパケットはソースノード５０からターゲットノードメモリコントローラ５２に送信され、その後でターゲットノードメモリコントローラ５２はProbe/Srcパケットをプローブノード５４Ａ－５４Ｎに送り、以下も同様である。パケットの時間順序を示すために、図１１ではソースノード５０およびターゲットメモリコントローラ５２を２箇所に示す。同様に、図１２および図１３では特定のブロックを１箇所以上に示す。ソースノード５０、ターゲットノードメモリコントローラ５２を含むターゲットノード、プローブされたノード５４Ａ－５４Ｎの各々は、図１に示す処理ノード１２Ａ－１２Ｄと同様の処理ノードを含み得る。

【００５４】

ソースノード５０はRdBlkパケットをターゲットノードメモリコントローラ５２に送信して、読出ブロックトランザクションを開始する。次いでターゲットノードメモリコントローラ５２は処理されるべきRdBlkパケットを選択する。ターゲットノードメモリコントローラ５２はProbe/Srcパケットを生成し、パケットをプローブされたノード５４Ａ－５４Ｎにブロードキャストする。さらに、ターゲットノードメモリコントローラ５２は、ターゲットノードメモリコントローラ５２が結合されているメモリ１４Ａ－１４Ｄからの読出を開始する。メモリ１４Ａ－１４Ｄからの読出が完了すると、ターゲットノードメモリコントローラ５２はデータを含むRdResponseパケットを生成し、パケットをソースノード５０に送信する。

【００５５】

プローブされたノード５４Ａ－５４Ｎの各々はそのキャッシュ内を探索して、RdBlkパケットによって読出されたキャッシュブロックがその中にストアされているかどうか判断する。ヒットが検出されると、対応するプローブされたノード５４Ａ－５４Ｎは、ターゲットノードメモリコントローラ５２から受取ったプローブパケット内のネクストステートフィールドに従ってキャッシュブロックの状態を更新する。さらに、プローブされたノード５４Ａ－５４Ｎの各々は（Probe/Srcパケットが既に受信されているために）、ProbeRespパケットをソースノード５０に経路制御する。この例においては、いずれのプローブされたノード５４Ａ－５４Ｎもキャッシュブロックのダーティコピーをストアしていない。

【００５６】

ソースノード５０は、プローブされたノード５４Ａ－５４NからのProbeRespパケットおよびターゲットメモリコントローラ５２からのRdResponseパケットを待機する。一旦これらのパケットが受取られると、ソースノード５０はSrcDoneパケットをターゲットメモリコントローラ５２に送信し、トランザクションを終了する。

【００５７】

次いで図１２は、第２の例示的な読出ブロックトランザクションを例示する図を示す。ソースノード５０、ターゲットメモリコントローラ５２、プローブされたノード５４Ａ－５４Ｎが示される。ソースノード５０、ターゲットメモリコントローラ５２を含むターゲットノード、およびプローブされたノード５４Ａ－５４Ｎの各々は、図１に示す処理ノード１２Ａ－１２Ｄと同様の処理ノードを含み得る。

【００５８】

図１１に示す例と同様に、ソースノード５０はRdBlkパケットをターゲットメモリコントローラ５２に送信する。ターゲットノードメモリコントローラ５２はProbe/Srcパケットをプローブされたノード５４Ａ－５４Ｎに送信し、かつRdResponseパケットをソースノード５０に送信し得る。

【００５９】

図１２の例においては、プローブされたノード５４Ａは読出ブロックトランザクションによってアクセスされたキャッシュブロックに対してダーティデータを検出する。したがって、プローブされたノード５４Ａは（Probe/Srcコマンドにより命令されて）ソースノード５０に、RdResponseパケットとプローブされたノード５４Ａの内部キャッシュから読出されたダーティキャッシュブロックとを送信する。したがって、一実施例においては、プローブされたノード５４ＡはMemCancelパケットをターゲットノードメモリコントローラ

５２に送信し得る。ターゲットノードメモリコントローラ５２がRdResponseパケットをソースノード５０に送信する前に、MemCancelパケットがターゲットノードメモリコントローラ５２に到達すると、ターゲットノードメモリコントローラ５２はRdResponseパケットを送信しない。よって、ターゲットノードメモリコントローラ５２からソースノード５０への「RdResponse」と表示された線は、その選択的な性質を示すために点線で表わされる。MemCancelメッセージに応答して、ターゲットノードメモリコントローラ５２はTgtDoneパケットをソースノード５０に送信する。

【００６０】

プローブされたノード５４Ｂ－５４Ｎは、この実施例においてはダーティデータを検出せず、よってProbeRespパケットをソースノード５０に経路制御する。一旦ソースノード５０がTgtDone、RdResponse、およびProbeRespパケットを受取ると、ソースノード５０はSrcDoneパケットをターゲットメモリコントローラ５２に送信して読出ブロックトランザクションを終了させる。

【００６１】

図１３は、例示的なサイズ指定された書込トランザクションを示す図である。ソースノード５０、ターゲットノードメモリコントローラ５２、およびプローブされたノード５４Ａ－５４Ｎを示す。ソースノード５０、ターゲットノードメモリコントローラ５２を含むターゲットノード、およびプローブされたノード５４Ａ－５４Ｎの各々は、図１に示す処理ノード１２Ａ－１２Ｄと同様の処理ノードを含み得る。

【００６２】

ソースノード５０は、WrSizedパケットおよび書込まれるべきデータをターゲットノードメモリコントローラ５２に送信することにより、サイズ指定された書込トランザクションを開始する。サイズ指定された書込トランザクションは、キャッシュブロックの一部を更新するが、キャッシュブロックの残りの部分は更新しないために、ターゲットノードメモリコントローラ５２はプローブされたノード５４Ａ－５４Ｎから（もし存在すれば）ダーティキャッシュブロックを収集する。さらに、キャッシュブロックのクリーンなコピーはプローブされたノード５４Ａ－５４Ｎにおいて無効化され、コヒーレンシを維持する。ターゲットメモリコントローラ５２は、処理されるべきサイズ指定された書込トランザクションを選択する際に、Probe/Tgtパケットをプローブされたノード５４Ａ－５４Ｎに送信する。プローブされたノード５４Ａ－５４Ｎは、（Probe/Tgtパケットが受取られているために）ターゲットノードメモリコントローラ５４に、（いずれのダーティデータも検出されなければ）ProbeRespパケットか、または（ダーティデータが検出されると）RdResponseパケットのいずれかを返す。一旦ターゲットノードメモリコントローラ５２が、プローブされたノード５４Ａ－５４Ｎからの応答を受取ると、ターゲットノードメモリコントローラ５２はTgtDoneパケットをソースノード５０に送信し、これはサイズ指定された書込トランザクションを終了させるSrcDoneパケットで応答する。

【００６３】

ターゲットノードメモリコントローラ５２が、プローブされたノード５４Ａ－５４Ｎのうちの１つからダーティなキャッシュブロックを受取ると、ターゲットノードメモリコントローラ５２は、ダーティキャッシュブロックとWrSizedデータパケット内のソースノード５０によって提供されたバイトとのマージを実行する。マージを実行するためのいかなる好適な機構をも用い得る。たとえば、ターゲットノードメモリコントローラ５２はデータをマージし、単一ブロック書込を行なってメモリを更新してもよい。これに代えて、ダーティブロックが第１にメモリに書込まれ、次いでソースノード５０によって提供されたバイトの書込が続いてもよい。

【００６４】

この説明は、ノード間で通信されるパケットについて記載する一方で、コマンド、応答および他のメッセージを送信するためのいかなる好適な機構をも用い得ることに留意されたい。

【００６５】

図１４は、処理のためのトランザクションの選択に応答する、メモリコントローラ１６Ａ－１６Ｄの一実施例の一部の動作を例示するフローチャートを示す。特に、プローブを生成するメモリコントローラ１６Ａ－１６Ｄの部分が示される。他の実施例も可能であり企図される。図１４において、理解を容易にするためにステップを特定の順序に従って示すが、いかなる好適な順序をも用い得る。さらに、ステップはデザイン選択によって所望のように、メモリコントローラ１６Ａ－１６Ｄ内で並列ハードウェアを用いてステップを並行して行なってもよい。

【００６６】

メモリコントローラは、選択されたトランザクションがWrSizedトランザクションであるかどうか判断する（判断ブロック６０）。もし選択されたトランザクションがWrSizedトランザクションであれば、メモリコントローラはProbe/Tgtパケットをプローブノードの各々に送信する（ステップ６２）。そうでなければ、メモリコントローラは選択されたトランザクションがVicBlkまたはCleanVicBlkトランザクションであるかどうか判断する（判断ブロック６４）。選択されたトランザクションがVicBlkまたはCleanVicBlkトランザクションであれば、プローブパケットは生成されない。しかしながら、選択されたトランザクションがWrSized、VicBlk、またはCleanVicBlkでなければ、Probe/Srcパケットはプローブされたノードに送信される（ステップ６６）。

【００６７】

図１５は、プローブパケットに応答するプローブされたノードの一実施例の動作を示すフローチャートである。他の実施例も可能であり企図される。図１５において、理解を容易にするためにステップを特定の順序に従って示すが、いかなる好適な順序をも用い得る。さらに、設計選択によって所望のように、プローブされたノード内で並列ハードウェアを用いてステップを並行して行なってもよい。

【００６８】

プローブされたノードはそのキャッシュを探索して、プローブによって表示されたキャッシュブロックがその中にストアされているかどうか判断し、もしキャッシュブロックが見出されればその状態を判断する。キャッシュブロックがダーティな状態で見出されると（判断ブロック７０）、プローブされたノードはRdResponseパケットを生成する。プローブされたノードはRdResponseパケットの後にキャッシュからダーティデータを読出して、データパケットとして送信する（ステップ７２）。一方で、キャッシュブロックが見出されないか、またはダーティな状態になければ、プローブされたノードはProbeRespパケットを生成する（ステップ７４）。さらに、キャッシュブロックの状態はプローブパケットのネクストステートフィールドに特定されるように更新される。

【００６９】

プローブされたノードは受取ったプローブパケットを調べる（判断ブロック７６）。プローブパケットがProbe/Srcパケットであれば、プローブされたノードは上で生成された応答を、Probe/Srcパケットに表示されるソースノードに経路制御する（ステップ７８）。言い換えると、プローブされたノードは応答パケット内のRespNodeフィールドを、Probe/SrcパケットのSrcNodeフィールド内の値にセットする。一方、プローブパケットがProbe/Tgtパケットであれば、プローブされたノードは上で生成された応答を、Probe/Tgtパケットに表示されるターゲットノードに経路制御する（ステップ８０）。言い換えると、プローブされたノードは応答パケット内のRespNodeフィールドを、Probe/TgtパケットのTgtNodeフィールド内の値にセットする。

【００７０】

図１６は、例示的な処理ノード１２Ａの一実施例のブロック図を示す。他の実施例も可能であり企図される。図１６の実施例においては、処理ノード１２Ａはインターフェイスロジック１８Ａ、１８Ｂ、１８Ｃおよびメモリコントローラ１６Ａを含む。さらに、処理ノード１２Ａは、プロセッサコア９２、キャッシュ９０、コヒーレンシ管理ロジック９８、および選択により第２のプロセッサコア９６および第２のキャッシュ９４を含み得る。インターフェイスロジック１８Ａ－１８Ｃは互いに結合され、さらにコヒーレンシ管理ロジ

ック９８に結合される。プロセッサコア９２および９６は、それぞれキャッシュ９０および９４に結合される。キャッシュ９０および９４は、コヒーレンシ管理ロジック９８に結合される。コヒーレンシ管理ロジック９８はメモリコントローラ１６Ａに結合される。

【００７１】

一般的に、コヒーレンシ管理ロジック９８は、メモリコントローラ１６Ａによって処理のために選択されたトランザクションに応答してプローブコマンドを生成し、かつ処理ノード１２Ａによって受取られるプローブコマンドに応答するよう構成される。コヒーレンシ管理ロジック９８は、処理のために選択されたトランザクションのタイプに依存して、Probe/SrcコマンドまたはProbe/Tgtコマンドのいずれかをブロードキャストする。さらに、コヒーレンシ管理ロジック９８は、受取ったプローブコマンドによって特定されるキャッシュブロックに対してキャッシュ９０および９４を探索し、適切なプローブ応答を生成する。さらに、Probe/Tgtコマンドを生成する場合には、コヒーレンシ管理ロジック９８は、Probe/Tgtコマンドに応答して返されたプローブ応答を収集し得る。キャッシュ９０および９４は、処理ノード１２Ａ内で開始された読出要求からのデータの受取を管理するフィルロジックを含むか、またはコヒーレンシ管理ロジック９８内にフィルロジックが含まれてもよい。コヒーレンシ管理ロジック９８はさらに、メモリコントローラ１６Ａに対する非コヒーレント要求を経路制御するよう構成されてもよい。一実施例においては、プロセッサ９２とプロセッサ９６とは、キャッシュ９０、キャッシュ９４、およびコヒーレンシ管理ロジック９８をバイパスして、特定のキャッシュ不可および／または非コヒーレントメモリ要求に対して直接メモリコントローラ１６Ａにアクセスし得る。

【００７２】

キャッシュ９０および９４は、データのキャッシュブロックをストアするよう構成された高速キャッシュメモリを含む。キャッシュ９０および９４は、それぞれのプロセッサコア９２および９６内に統合されてもよい。これに代えて、キャッシュ９０および９４は、所望のようにプロセッサコア９２および９６に、バックサイドキャッシュ構成またはインライン構成で結合されてもよい。さらに、キャッシュ９０および９４はキャッシュ階層構造として実現化されてもよい。プロセッサコア９２および９６に（階層内で）より近いキャッシュは、所望であればプロセッサコア９２および９６に統合されてもよい。

【００７３】

プロセッサコア９２および９６は、予め定められた命令セットに従って命令を実行するための回路を含む。たとえば、ｘ８６命令セットアーキテクチャを選択し得る。これに代えて、Alpha、PowerPC、または他のいかなる命令セットアーキテクチャを選択してもよい。一般的に、プロセッサコアはデータおよび命令のためにキャッシュにアクセスする。キャッシュミスが検出されると、読出要求が生成され、ミスの発生したキャッシュブロックがマッピングされているノード内のメモリコントローラに送信される。

【００７４】

分散メモリシステムを用いて、コンピュータシステム１０の特定の実施例を説明してきたが、分散メモリシステムを用いない実施例も、ここで説明した柔軟なプローブ／プローブ応答経路制御を用い得ることに留意されたい。そのような実施例が企図される。

【００７５】

上の開示を完全に理解すると、当業者においてはいくつもの変更および変形が明らかとなるであろう。前掲の特許請求の範囲がすべてのそのような変更および変形を包含することが理解されることを意図する。

【００７６】

【産業上の用途】

この発明は一般的にコンピュータシステムに適用可能である。

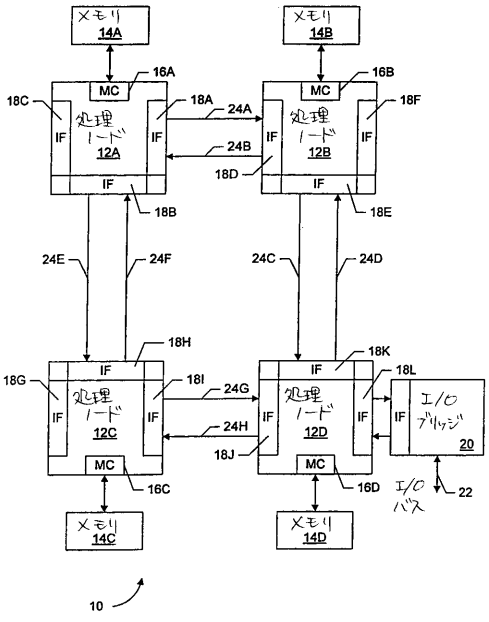
【図面の簡単な説明】

【図１】　コンピュータシステムの一実施例のブロック図である。

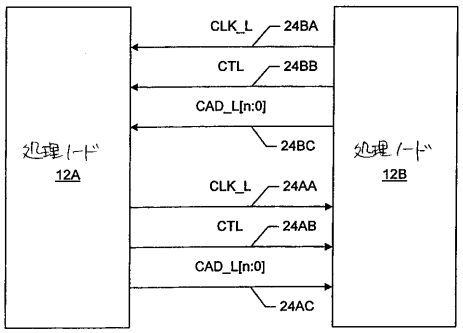
【図２】　図１に示す１対の処理ノードの間の相互接続の一実施例を強調した、ブロック図である。

【図３】　情報パケットの一実施例のブロック図である。
【図４】　アドレスパケットの一実施例のブロック図である。
【図５】　応答パケットの一実施例のブロック図である。
【図６】　データパケットの一実施例のブロック図である。
【図７】　コンピュータシステムの一実施例によって用い得る、例示的なパケットタイプを示すテーブルである。
【図８】　プローブパケットの一実施例のブロック図である。
【図９】　プローブ応答パケットの一実施例のブロック図である。
【図１０】　読出応答パケットの一実施例のブロック図である。
【図１１】　読出ブロックトランザクションに対応するパケットの例示的なフローを示す図である。
【図１２】　読出ブロックトランザクションに対応するパケットの第２の例示的なフローを示す図である。
【図１３】　サイズ指定された書込トランザクションに対応するパケットの例示的なフローを示す図である。
【図１４】　メモリコントローラの一実施例の動作を示すフローチャートの図である。
【図１５】　プローブパケットを受取る処理ノードの一実施例の動作を示すフローチャートの図である。
【図１６】　処理ノードの一実施例のブロック図である。

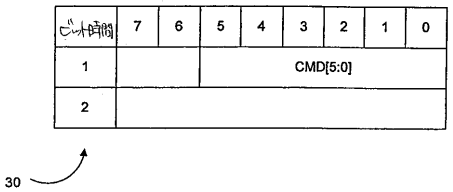
【図１】



【図２】



【図３】



【図４】

ビット時間	7	6	5	4	3	2	1	0
1	DestNode [1:0]		CMD[5:0]					
2	SrcTag [1:0]		SrcNode[3:0]				DestNode [3:2]	
3			SrcTag[6:2]					
4	Addr[7:0]							
5	Addr[15:8]							
6	Addr[23:16]							
7	Addr[31:24]							
8	Addr[39:32]							

32

【図５】

ビット時間	7	6	5	4	3	2	1	0
1	DestNode [1:0]		CMD[5:0]					
2	SrcTag [1:0]		SrcNode[3:0]				DestNode [3:2]	
3			SrcTag[6:2]					
4								

34

【図６】

ビット時間	7	6	5	4	3	2	1	0
1	Data[7:0]							
2	Data[15:8]							
3	Data[23:16]							
4	Data[31:24]							
5	Data[39:32]							
6	Data[47:40]							
7	Data[55:48]							
8	Data[63:56]							

36

【図７】

CMD符号	コマンド	パケットタイプ
000000	Nop	Info
000001	Interrupt Broadcast	
00001x	Reserved	-
000100	Probe/Src	Address
000101	Probe/Tgt	Address
00011x	Reserved	-
001000	RdBlkS	Address
001001	RdBlk	Address
001010	RdBlkMod	Address
001011	ChangetoDirty	Address
001100	ValidateBlk	Address
001101	CleanVicBlk	Address
001110	Interrupt Target	
001111	VicBlk	Address
01xxxx	ReadSized	Address
10xxxx	WrSized	Address
11000x	RdResponse	Response
11001x	Reserved	-
11010x	ProbeResp	Response
11011x	Reserved	-
111000	SrcDone	Response
111001	MemCancel	Response
111010	TgtStart	Response
111011	Reserved	-
11110x	TgtDone	Response
111110	IntrResponse	Response
111111	Error	Info

38

【図８】

ビット時間	7	6	5	4	3	2	1	0
1	TgtNode [1:0]		CMD[5:0]					
2	SrcTag [1:0]		SrcNode[3:0]				TgtNode [3:2]	
3	NextState [1:0]		DM	SrcTag[6:2]				
4	Addr[7:0]							
5	Addr[15:8]							
6	Addr[23:16]							
7	Addr[31:24]							
8	Addr[39:32]							

40

【図９】

ビット時間	7	6	5	4	3	2	1	0
1	RespNode [1:0]		CMD[5:0]					
2	SrcTag [1:0]		SrcNode[3:0]				RespNode [3:2]	
3	Rsv		Hit	SrcTag[6:2]				
4	Rsv							

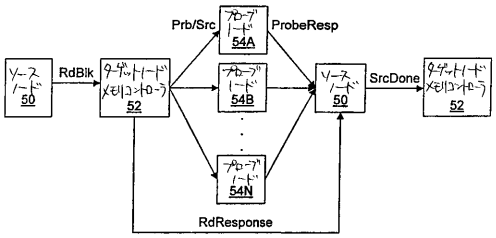
42

【図１０】

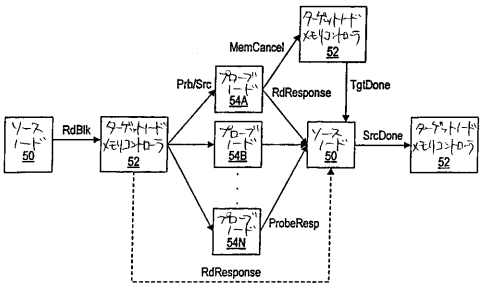
ビット時間	7	6	5	4	3	2	1	0
1	RespNode [1:0]		CMD[5:0]					
2	SrcTag [1:0]		SrcNode[3:0]				RespNode [3:2]	
3	Count				SrcTag[6:2]			
4	Rsv						Prb	Type

44

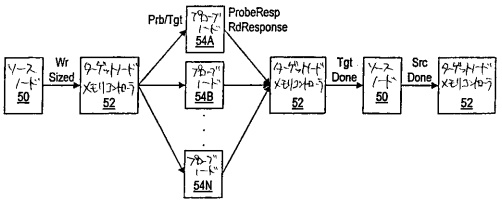
【図１１】



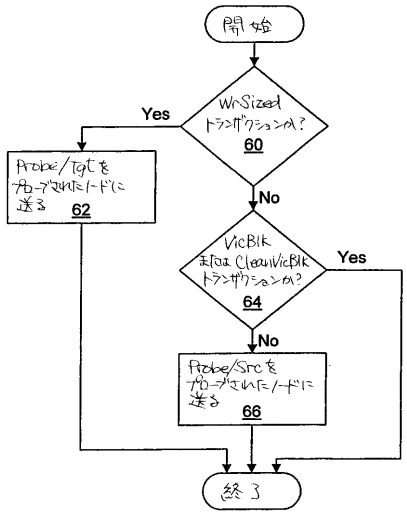
【図１２】



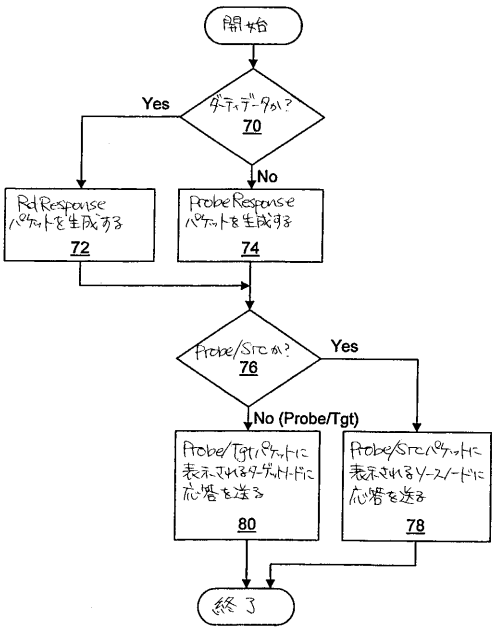
【図１３】



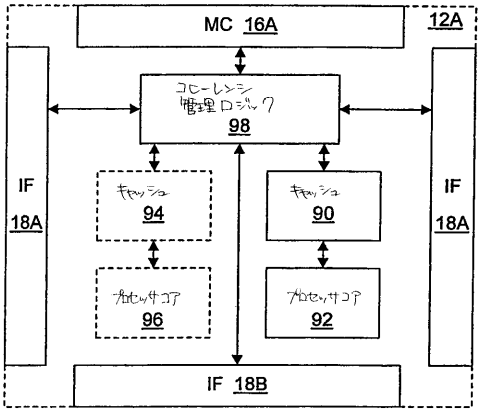
【図１４】



【図１５】



【図１６】


MC 16A
12A
コヒーレンシ
管理ロジック
98
IF
18A
キャッシュ
94
キャッシュ
90
IF
18A
プロセッサコア
96
プロセッサコア
92
IF 18B

フロントページの続き

(74)代理人 100083703
弁理士 仲村 義平
(74)代理人 100091409
弁理士 伊藤 英彦
(74)代理人 100096781
弁理士 堀井 豊
(74)代理人 100096792
弁理士 森下 八郎
(72)発明者 ケラー, ジェイムズ・ビィ
アメリカ合衆国、９４３０３ カリフォルニア州、パロ・アルト、アイリス・ウェイ、２１０
(72)発明者 ギューリック, デイル・イー
アメリカ合衆国、７８７４８ テキサス州、オースティン、フェスタス・ドライブ、３１２２

審査官 高瀬 勤

(56)参考文献 特開平１０－１０５４６４（ＪＰ，Ａ）
特開平０６－２７４４６１（ＪＰ，Ａ）
特開平１０－３４０２２７（ＪＰ，Ａ）
特開平０８－３２０８２７（ＪＰ，Ａ）
特開平０８－０１６４７４（ＪＰ，Ａ）
特表２００２－５３３８１３（ＪＰ，Ａ）
Akhilesh Kumar, Phanindra K. Mannava, Laxmi N. Bhuyan, Efficient and Scalable Cache Coherence Schemes for Shared Memory Hypercube Multiprocessors, Proceedings of Supercomputing'94, IEEE, １９９４年１１月１４日, Pages:498-507
James Laudon, Daniel Lenoski, The SGI Origin:A ccNUMA Highly Scalable Server, Proceedings of the 24th Annual International Symposium on Computer Architecture(ISCA'97), ACM, １９９７年 ６月, Pages:241-251

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 12/08