(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4718012号
(P4718012)
(45)発行日　平成23年7月6日(2011.7.6)　(24)登録日　平成23年4月8日(2011.4.8)

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

請求項の数 10　(全 27 頁)

(21)出願番号　特願2000-590062(P2000-590062)
(86)(22)出願日　平成11年8月26日(1999.8.26)
(65)公表番号　特表2002-533813(P2002-533813A)
(43)公表日　平成14年10月8日(2002.10.8)
(86)国際出願番号　PCT/US1999/019856
(87)国際公開番号　WO2000/038070
(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　09/217,699
(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)
(31)優先権主張番号　09/217,212
(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】メモリキャンセルメッセージを用いたシステムメモリ帯域幅の節約およびキャッシュコヒーレンシ維持

(57)【特許請求の範囲】
【請求項１】
マルチプロセッシングコンピュータシステムであって、
相互接続構造を介して相互接続される複数の処理ノードを含み、前記複数の処理ノードは、
指定されたメモリ位置からデータを読出す第１の読出動作を開始するよう構成される第１の処理ノードと、
第２の処理ノードとを含み、前記第２の処理ノードは、前記第１の読出動作に応答して、前記第２の処理ノードに連結されたメモリ内の前記指定されたメモリ位置からデータを読出す第２の読出動作を開始するよう構成され、前記第２の処理ノードは、さらに、前記第１の読出動作に応答してプローブを発行するように構成され、前記マルチプロセッシングコンピュータシステムはさらに、
第３の処理ノードを含み、前記第３の処理ノードは、前記第２の処理ノードからのプローブを受取るように、かつ、前記プローブに応答して、前記指定されたメモリ位置に対応し、前記第３の処理ノード内に記憶された、変更されたデータを検出するように連結され、前記第３の処理ノードは、前記指定されたメモリ位置の変更されたコピーを前記第３の処理ノード内に検出すると、前記第２の処理ノードにメモリキャンセル応答を送信するよう構成され、前記メモリキャンセル応答は、前記第２の処理ノードに前記第２の読出動作のさらなる処理を打切らせ、
前記第２の処理ノードは、前記第２の読出動作の間に読出された前記データを、前記第

１の処理ノードに第１の読出応答を送信することにより転送するよう構成され、前記メモリキャンセル応答は、前記第２の処理ノードが前記メモリキャンセル応答を前記第１の読出応答の送信前に受信した場合に、前記第２の処理ノードに前記第１の読出応答の送信をキャンセルさせ、

前記第３の処理ノードは前記メモリキャンセル応答と並行に第２の読出応答を送信するよう構成され、前記第２の読出応答は前記第１の処理ノードに送信される、マルチプロセッシングコンピュータシステム。

【請求項２】

前記第２の処理ノードは、前記指定されたメモリ位置が前記第３の処理ノード内にキャッシュされているか否かに拘らず、プローブコマンドを送信するよう構成される、請求項１に記載のマルチプロセッシングコンピュータシステム。

【請求項３】

前記第２の読出動作の間に読出された前記データのサイズは、前記第１の読出動作のタイプに依存する、請求項１に記載のマルチプロセッシングコンピュータシステム。

【請求項４】

前記第２の読出応答は、前記第３の処理ノード内にキャッシュされた前記指定されたメモリ位置の前記変更されたコピーを含むデータパケットを含み、

前記第２の処理ノードは、前記第３の処理ノードから前記メモリキャンセル応答を受信すると、前記第１の処理ノードにtarget done応答を送信するよう構成され、前記target done応答は、前記第１の読出応答が送信されるか否かに拘らず送信される、請求項１に記載のマルチプロセッシングコンピュータシステム。

【請求項５】

前記第１の処理ノードは、前記target done応答および前記第２の読出応答を受信すると、前記第２の処理ノードにsource doneメッセージを送信するよう構成される、請求項４に記載のマルチプロセッシングコンピュータシステム。

【請求項６】

相互接続構造を介して相互接続される複数の処理ノードを含むマルチプロセッシングコンピュータシステムにおいて、前記複数の処理ノードは第１の処理ノードと、第２の処理ノードと、第３の処理ノードとを含み、前記第２の処理ノードに関連のメモリ内のメモリ位置の内容を選択的に読出すための方法であって、

前記第１の処理ノードによる前記メモリ位置の前記内容を読出す第１の読出動作を開始するステップと、

前記第１の読出動作に応答して、前記第２の処理ノードによる第２の読出動作をさらに開始するステップとを含み、前記第２の処理ノードは、前記第２の読出動作の間に前記第２の処理ノードに連結されたメモリ内の前記メモリ位置の前記内容を読出し、前記第２の処理ノードは、さらに、前記第１の読出動作に応答してプローブを発行し、前記第２の読出動作は前記第２の処理ノードから前記第１の処理ノードへの第１の読出応答を含み、前記第１の読出応答は前記メモリ位置の前記内容に対する第１のデータパケットを含み、方法はさらに、

前記第３の処理ノードが、前記第２の処理ノードからのプローブを受取り、前記プローブに応答して、前記メモリ位置に対応し、前記第３の処理ノード内に記憶された、変更されたデータを検出し、前記第３の処理ノードが、前記第３の処理ノード内に前記メモリ位置の変更されたコピーを検出すると、第２の処理ノードにメモリキャンセル応答を送信するステップと、　前記メモリキャンセル応答が、前記第２の処理ノードに前記第２の読出動作のさらなる処理を打切らせるステップと、

前記メモリキャンセル応答が、前記第２の処理ノードが前記メモリキャンセル応答を前記第１の読出応答の送信前に受信した場合に、前記第２の処理ノードに前記第１の読出応答の送信をキャンセルさせるステップと、

前記第３の処理ノードが前記メモリキャンセル応答と並行に第２の読出応答を送信するステップとを含み、前記第２の読出応答は前記第１の処理ノードに送信される、方法。

【請求項７】
前記第１のデータパケットのサイズは、前記第１の読出動作のタイプに依存する、請求項６に記載の方法。
【請求項８】
前記第２の読出応答は、前記第３の処理ノード内にキャッシュされた前記メモリ位置の前記変更されたコピーを含む第２のデータパケットを含む、請求項６に記載の方法。
【請求項９】
前記第２の処理ノードが、前記第３の処理ノードから前記メモリキャンセル応答を受信すると、前記第１の処理ノードにtarget done応答を送信するステップをさらに含み、前記target done応答は、前記第１の読出応答が送信されるか否かに拘らず送信される、請求項８に記載の方法。
【請求項１０】
前記第１の処理ノードが、前記target done応答および前記第２の読出応答を受信すると、前記第２の処理ノードにsource doneメッセージを送信するステップをさらに含む、請求項９に記載の方法。
【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
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Ｉ／Ｏブリッジを介して共用バスに結合される場合もあり、該Ｉ／Ｏブリッジは共用バスとＩ／Ｏ装置との間の情報の転送を管理する。プロセッサは典型的には、直接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。さらに、共用システムを介したシステムメモリ読出および書込サイクルは、プロセッサ内のキャッシュが関連するか、または２つ以上のプロセッサが関連する情報転送よりも、かなり長い時間を必要とする。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。限られたシステムメモリ帯域幅を節約する機構を提供しない限り、全体的な性能は減じられるであろう。
【０００４】
ノンキャッシュシステムメモリに対しアドレス指定された書込または読出動作は、２つのプロセッサの間の、またはプロセッサとその内部キャッシュとの間の同様の動作よりも、より多くのプロセッサクロックサイクルをとる。バス帯域幅への制限は、システムメモリへの読出または書込のための長いアクセス時間とあいまって、コンピュータシステム性能に悪影響を及ぼす。
【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】

ＥＰ－Ａ－０　３７９　７７１は、デジタルコンピュータのためのメモリ制御システムを開示するが、ここでは、要求されたデータの変更された状態のものが別の関連のＣＰＵキャッシュ内に利用可能である場合に、関連するメインメモリを備えたシステム制御ユニット（ＳＣＵ）の読出が、関連の中央演算ユニット（ＣＰＵ）に応答して自動的に打切られる。

Ｃ．Ａ．プリート（C. A. Prete）による「密結合マルチプロセッサシステムのためのＲＴＳキャッシュメモリ設計（RTS Cache Memory Design for a Tightly Coupled Multiprocessor System）」IEEE Micro. US, IEEE Inc. New York Vol.11 No.2、１９９１年４月１１日、pp.１６－１９、４０－５２、は、縮小状態遷移（Reduced State Transitions）として知られる、マルチプロセッサシステムにおけるキャッシュメモリのためのコヒーレンスプロトコルを開示する。読出または書込動作の際にキャッシュミスが起こると、キャッシュはまずキャッシュコピーを置換え、次いで要求された動作を実行する。置換えの段階は、ヴィクティムキャッシュブロックを選択するステップと、ヴィクティムコピーに関連するメモリブロックを更新するステップと、読出ブロックトランザクションにより、要求されたメモリブロックを読出すステップと、からなる。

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。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２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。

【０００８】

この発明の第１の局面によると、コンピュータシステムが提供され、該コンピュータシステムは読出トランザクションによりアドレス指定されたデータの変更されたコピーを保持していることに応答して、該読出トランザクションに対応するメモリキャンセル応答を送信するよう構成される第１の処理ノードを含み、該第１の処理ノードは、（ｉ）第２の処理ノードからプローブを受取り、（ii）該読出トランザクションによってアドレス指定されたデータの変更されたコピーを検出し、（iii）該変更されたコピーの検出に応答して該メモリキャンセル応答を送信するよう構成される、処理ノードであり、該コンピュータシステムはさらに該第２の処理ノードを含み、該第２の処理ノードはトランザクションのターゲットノードを含み、かつシステムメモリの少なくとも一部に結合され、該システムメモリの少なくとも一部は、読出トランザクションによってアドレス指定されたデータに対応する記憶位置を含み、該第２の処理ノードは該第１の処理ノードから該メモリキャンセル応答を受け取るよう結合され、該第２の処理ノードは、該メモリキャンセル応答に応答して、該記憶位置への読出サイクルのさらなる処理を打ち切るよう構成される。

一実施例においては、処理ノードは複数のデュアル単方向リンクを介して相互接続される。単方向リンク対の各々は、処理ノードのうちの２つを接続するコヒーレントなリンク構造を形成する。単方向リンク対の一方のリンクは、第１の処理ノードから信号を、その単方向リンク対を介して接続された第２の処理ノードに送る。単方向リンク対の他方のリン

クは、信号の逆のフローを運ぶ。すなわち、信号を第２の処理ノードから第１の処理ノードへ送る。こうして、単方向リンクの各々は、パケット化情報転送のために設計されたポイントツーポイント相互接続を形成する。２つの処理ノード間の通信は、システム内の１つ以上の残りのノードを介して経路制御されることもある。

【０００９】

処理ノードの各々は、メモリバスを介してそれぞれのシステムメモリに結合されることができる。メモリバスは双方向であってもよい。処理ノードの各々は、少なくとも１つのプロセッサコアを含み、かつ選択によりそれぞれのシステムメモリと通信するためのメモリコントローラを含み得る。１つ以上のＩ／Ｏブリッジを介したさまざまなＩ／Ｏ装置との接続性を可能にするため、１つ以上の処理ノードに他のインターフェイスロジックを含んでもよい。

【００１０】

一実施例においては、１つ以上のＩ／Ｏブリッジを、１組の非コヒーレントなデュアル単方向リンクを介してそれぞれの処理ノードに結合し得る。これらのＩ／Ｏブリッジは、この非コヒーレントなデュアル単方向リンクの組を介してそれらのホストプロセッサと通信するが、これは２つの直接リンクされたプロセッサがコヒーレントなデュアル単方向リンクを介して互いと通信するのとほぼ同じ方法である。

【００１１】

プログラム実行の間のある時点で、キャッシュ内にメモリデータのダーティコピーを持つ処理ノードは、その変更されたデータを含むキャッシュブロックを捨てることができる。一実施例においては、その処理ノード（ソースノードとも呼ばれる）はヴィクティムブロックコマンドをキャッシュされたダーティなデータと併せて第２の処理ノードに送信するが、該第２の処理ノードはすなわち、キャッシュされたデータのための対応するメモリ位置を有するシステムメモリの一部に結合されたものである。この第２の処理ノード（ターゲットノードとも呼ばれる）は、応答してターゲット終了メッセージを送信処理ノードに送り、メモリ書込サイクルを開始して受取ったデータを関連のノンキャッシュメモリに転送し、対応するメモリ位置の内容を更新する。もし送信処理ノードが、ヴィクティムブロックコマンドを送った時間と、ターゲット終了メッセージを受取った時間との間で無効化プローブに出会えば、送信ノードはターゲットノード、すなわち第２の処理ノードにメモリキャンセル応答を送り、メモリ書込サイクルのさらなる処理を打切る。これはシステムメモリ帯域幅を節約するという効果をもたらし、ノンキャッシュメモリに書込まれるべきデータが失効している場合、時間がかかるメモリ書込動作を回避し得る。

【００１２】

メモリキャンセル応答は、ヴィクティムブロック書込動作の間のキャッシュコヒーレンシを維持し得るが、特に、ヴィクティムブロックの宛先であるメモリ位置の内容を読出すための第３の処理ノード（ヴィクティムブロックを送ったソースノード以外のもの）からの読出コマンドの後に、ヴィクティムブロックがターゲットノード（すなわち、第２の処理ノード）に到着する状況において、コヒーレンシを維持し得る。読出コマンドは、そのメモリ位置から読出したデータを変更するという第３の処理ノードの意図を明らかにし得る。したがって、ターゲットノードは応答して、ソースノードを含むシステム内の処理ノードの各々に無効化プローブを伝送し得る。後から到着したヴィクティムブロックは、最新のデータを含み得ず、かつターゲットノードメモリ内の対応するメモリ位置にコミットする必要がないために、ソースノードがターゲット終了応答を受取ったときに、ソースノードはターゲットノードにメモリキャンセル応答を送る。さらに、ターゲット終了応答は無効化プローブの介入の後で受取られるために、ソースノードからのメモリキャンセル応答はこうして処理ノードの間のキャッシュコヒーレンシを維持する助けをする。

【００１３】

一実施例においては、第１の処理ノードが第２の処理ノードに読出コマンドを送って、第２の処理ノードに関連する指定されたメモリ位置からデータを読出すと、第２の処理ノードは、応答してシステム内のすべての残りの処理ノードにプローブコマンドを送信する。

指定されたメモリ位置のキャッシュされたコピーを有する処理ノードの各々は、そのキャッシュされたデータに関連するキャッシュタグを更新してデータの現在のステータスを反映させる。プローブコマンドを受取った処理ノードの各々は次いで、処理ノードがデータのキャッシュされたコピーを有するかどうかを示すプローブ応答を送る。処理ノードが指定されたメモリ位置のキャッシュされたコピーを有する場合には、その処理ノードからのプローブ応答はキャッシュされたデータの状態、すなわち変更、共用などをさらに含む。

【００１４】

プローブコマンドを受取ると、すべての残りのノードは、指定されたメモリ位置のキャッシュされたコピーがもしあれば、上述のようにそのステータスをチェックする。ソースノードとターゲットノード以外の処理ノードが、指定されたメモリ位置のキャッシュされたコピーで、かつ変更された状態のものを見出した場合には、その処理ノードは応答してターゲットノード、すなわち第２の処理ノードにメモリキャンセル応答を送る。このメモリキャンセル応答は、第２の処理ノードにさらなる読出コマンドの処理を打切らせ、かつまだ読出応答を送っていなければ、読出応答の送信を中止させる。それでも他のすべての残りの処理ノードは、それらのプローブ応答を第１の処理ノードに送る。変更されたキャッシュされたデータを有する処理ノードは、その変更されたデータをそれ自体の読出応答を介して第１の処理ノードに送る。プローブ応答と読出応答とを含むメッセージ通信方式はこうして、システムメモリ読出動作の間にキャッシュコヒーレンシを維持する。

【００１５】

メモリキャンセル応答はさらに、第２の処理ノードがそれ以前に読出応答を第１の処理ノードに送ったかどうかにかかわらず、第２の処理ノードがターゲット終了応答を第１の処理ノードに送信するようにさせる。第１の処理ノードは、すべての応答、すなわちプローブ応答、ターゲット終了応答、および変更されたキャッシュされたデータを有する処理ノードからの読出応答を待ち、その後で第２の処理ノードにソース終了応答を送ることにより、データ読出サイクルを完了させる。この実施例においては、メモリキャンセル応答は、要求されたデータの変更されたコピーが異なった処理ノードにおいてキャッシュされたときに時間のかかるメモリ読出動作を打切らせることにより、システムメモリ帯域幅を節約し得る。処理ノードとシステムメモリとの間の比較的低速のシステムメモリバスが関与する同様のデータ伝送よりも、高速デュアル単方向リンクを介した２つの処理ノード間のデータ伝送が実質的に速いことが観察されたとき、こうしてデータ転送レイテンシの減少が達成される。

【００１６】

以下の図面と併せて、以下の好ましい実施例の詳細な説明を考察することにより、この発明はよりよく理解されるであろう。

【００１７】

【発明の実施の形態】

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。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、インターフェイスロジックとしても知られる、１つ以上のインターフェイスポート１８を含んで処理ノード１２Ａ－１２Ｄの間で通信し、かつ処理ノードと対応するＩ／Ｏブリッジとの間でも通信する。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。

【００１８】

処理ノード１２Ａ－１２Ｄを相互接続するインターフェイス構造は、１組のデュアル単方向リンクを含む。デュアル単方向リンクの各々は、パケットベースの１対の単方向リンクとして実現化されて、コンピュータシステム１０内のどの２つの処理ノード間でも高速パケット化情報転送を達成する。単方向リンクの各々は、パイプライン化され分割されたトランザクションによる相互接続として見ることができる。単方向リンク２４の各々は、１組のコヒーレントな単方向ラインを含む。こうして、単方向リンク対の各々は、第１の複数のバイナリパケットを担持する１つの送信バスと、第２の複数のバイナリパケットを担持する１つの受信バスとを含む、と見ることができる。バイナリパケットの内容は第１に、要求される動作の種類と、動作を開始する処理ノードとに依存する。デュアル単方向リンク構造の一例は、リンク２４Ａおよびリンク２４Ｂである。単方向ライン２４Ａを用いてパケットを処理ノード１２Ａから処理ノード１２Ｂに送信し、ライン２４Ｂを用いてパケットを処理ノード１２Ｂから処理ノード１２Ａに送信する。ライン２４Ｃ－２４Ｈの他の組を用いて、図１に示すようにそれらの対応する処理ノードの間のパケットを送信する。

【００１９】

同様のデュアル単方向リンク構造を用いて、処理ノードとその対応のＩ／Ｏ装置、またはグラフィック装置、もしくは処理ノード１２Ｄに関して示すＩ／Ｏブリッジとの間の相互接続を行ない得る。デュアル単方向リンクは、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外部Ｉ／Ｏ、またはグラフィック装置、もしくは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つ以上の残りの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によって処理ノード１２Ｄに送信されるパケットは、図１の構成内の処理ノード１２Ｂまたは処理ノード１２Ｃのいずれをも通過し得る。好適な経路制御アルゴリズムのいずれを用いることもできる。コンピュータシステム１０の他の実施例は、図１に示すものよりもより多くの、またはより少ない処理ノードを含み得る。

【００２０】

処理ノード１２Ａ－１２Ｄは、メモリコントローラおよび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コア、内部キャッシュメモリ、バスブリッジ、グラフィックスロジック、バスコントローラ、周辺装置コントローラなどの他の回路素子を含み得る。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通信するためのメモリコントローラを含む。さらに、処理ノード内の回路素子の各々は、処理ノードによって行なわれる機能に依拠して１つ以上のインターフェイスポートに結合されることができる。たとえばある回路素子は、Ｉ／Ｏブリッジを処理ノードに接続するインターフェイスロジックのみを結合し、他の回路素子は２つの処理ノードを接続するインターフェイスロジックのみを結合し得る。他の組合せは、所望のように容易に実現し得る。

【００２１】

メモリ１４Ａ－１４Ｄは、いずれ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のメモリアドレス空間は、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。

【００２２】

一般的には、インターフェイスロジック１８Ａ－１８Ｌは、１つの単方向リンクからのパケットを受取り、かつ別の単方向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、送信インターフェイスロジック１８の各々は、送信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受信インターフェイスロジック内に、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。

【００２３】

次に図２は、処理ノード１２Ａおよび１２Ｂのブロック図を示し、処理ノード１２Ａおよび１２Ｂを接続するデュアル単方向リンク構造のより詳細な一実施例を例示する。図２の実施例においては、ライン２４Ａ（単方向リンク２４Ａ）は、クロックライン２４ＡＡと、制御ライン２４ＡＢと、コマンド／アドレス／データバス２４ＡＣとを含む。同様に、ライン２４Ｂ（単方向リンク２４Ｂ）は、クロックライン２４ＢＡと、制御ライン２４ＢＢと、コマンド／アドレス／データバス２４ＢＣとを含む。

【００２４】

クロックラインは、対応する制御ラインおよびコマンド／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伝送し得る。コマンド／アドレス／データバスの幅に依拠して、多数のクロックラインを用い得る。たとえば３２ビットコマンド／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（コマンド／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、残りの半分のコマンド／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）。

【００２５】

制御ラインは、コマンド／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。

【００２６】

コマンド／アドレス／データバスは、データ、コマンド、応答、およびアドレス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、コマンド／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、所望の他のサイズのコマンド／アドレス／データバスをサポートし得る。

【００２７】

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）。好適な正および負論理の組合せもまた実現化し得る。

【００２８】

図３から図７は、コンピュータシステム１０の一実施例に従った、キャッシュコヒーレントな通信（すなわち処理ノード間の通信）に用いられる例示的なパケットを示す。図３から図６は制御パケットを示し、図７はデータパケットを示す。他の実施例は異なったパケット定義を用い得る。制御パケットおよびデータパケットは集合的にバイナリパケットとも呼ぶ。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７は、８ビットコマンド／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、（７から０まで番号が付与された）８ビットの制御情報またはデータ情報は、ビット時間の各々の間に８ビットコマンド／アドレス／データバス上を送信される。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伝送するために用いられるかのいずれかであり得る。

【００２９】

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、かつコマンドフィールドＣＭＤ［５：０］で示す、６ビットを含む。例示的なコマンドフィールド符号化を図８に示す。図４、図５、図６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。

【００３０】

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、DestNodeフィールドで示す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（SrcNode）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（SrcTag）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）。こうして、たとえばSrcTagフィールドが７ビット長さであれば、対応するソースノードはシステム内で進行する間に最大１２８（2^7）の異なったトランザクションを有し得る。システム内の他のノードからの応答は、応答内のSrcTagフィールドを介して対応のトランザクションと関連付けられる。ビット時間４から８までを用いて、アドレスフィールドAddr［３９：０］で示す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を用いて、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し得る。

【００３１】

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。SrcNode（ソースノード）フィールドは好ましくは、応答パケットの生成

を促すトランザクションを発信したノードを識別する。一方、DestNode（宛先ノード）フィールドは、応答パケットの最終的なレシーバである処理ノードを、すなわちソースノードまたはターゲットノード（後に説明）を識別する。さまざまな種類の応答パケットが付加的な情報を含み得る。たとえば、図１１Ａを参照して後に説明する読出応答パケットは、以下の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。後に図１２を参照して説明するプローブ応答は、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。

【００３２】

図６は、コマンドパケット（commandパケット）３６の例を示す。上述のように、単方向リンクの各々はパイプライン化され、分割されたトランザクション相互接続であって、トランザクションはソースノードによってタグ付けされ、応答は任意の所与の時間にも、パケットの経路制御に依存して順不同でソースノードに戻ることができる。ソースノードは、コマンドパケットを送信してトランザクションを開始する。ソースノードはアドレスマッピングテーブルを含み、ターゲットノード番号（TgtNodeフィールド）をコマンドパケットに入れて、コマンドパケット３６の宛先である処理ノードを識別する。コマンドパケット３６は、ＣＭＤフィールド、SrcNodeフィールド、SrcTagフィールド、およびAddrフィールドを有するが、これらはアドレスパケット３２（図４）を参照に説明され示されたものと同様である。

【００３３】

コマンドパケット３６の１つの際立った特徴は、Countフィールドの存在である。キャッシュ不可能な読出または書込動作においては、データのサイズはキャッシュブロックのサイズよりも小さくあり得る。こうして、たとえば、キャッシュ不可能な読出動作は、システムメモリまたはＩ／Ｏ装置からのちょうど１バイトまたは１クワッドワード（６４ビット長さ）だけのデータを必要とし得る。この種類のサイズ指定された読出または書込動作は、Countフィールドの助けによって容易となる。この例においては、Countフィールドは３ビット長さで示す。したがって、所与のサイズ指定されたデータ（バイト、クワッドワードなど）は、最高８回まで送信されることができる。たとえば、８ビットリンクにおいては、Countフィールドの値が０（バイナリ０００）である場合、コマンドパケット３６は１つのビット時間でのちょうど１バイトだけのデータの転送を示す。一方、Countフィールドの値が７（バイナリ１１１）である場合、クワッドワード、すなわち８バイトが、合計で８ビット時間の間に伝送されることができる。ＣＭＤフィールドは、いつキャッシュブロックが伝送されたのかを識別し得る。この場合、Countフィールドは固定値を有し、キャッシュブロックが６４バイトサイズである場合７であるが、これは８クワッドワードがキャッシュブロックを読出または書込するために伝送されなければならないためである。８ビットワイドの単方向リンクの場合においては、６４のビット時間にわたる８つの完全なデータパケット（図７）の伝送を必要とし得る。好ましくは、データパケット（図７を参照して後に説明）は、書込コマンドパケットまたは読出応答パケット（後に説明）の直後に続き、データバイトはアドレスの昇順で転送されることができる。単一のバイトまたはクワッドワードのデータ転送は、自然に整地されたそれぞれ８または６４バイト境界をまたがらない。

【００３４】

図７は、データパケット（dataパケット）３８を示す。データパケット３８は、図７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３８は、６４バイトのキャッシュブロックを含み得るが、この場合キャッシュブロック転送を完了させるために（８ビットリンク上の）６４のビット時間がかかるであろう。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、コマンドパケット３６（図６）を参照して上に説明したように、キャッシュ不可能な読出および書込に対する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しか必要としない。

【００３５】

図３から図７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７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ビットリンクに対す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。

【００３６】

【数１】

$$BT1_{16}[15:0] = BT2_8[7:0] \,||\, BT1_8[7:0] \qquad (1)$$

$$BT1_{32}[31:0] = BT4_8[7:0] \,||\, BT3_8[7:0] \,||\, BT2_8[7:0] \,||\, BT1_8[7:0] \qquad (2)$$

【００３７】

図８は、コンピュータシステム１０内のデュアル単方向リンク構造の１つの例示的な実施例に対して用いられるコマンドを示すテーブル４０を示す。テーブル４０は、コマンドの各々に割当てられたコマンド符号化（ＣＭＤフィールド）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３０－３８（図３から図７）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。図８におけるコマンドのいくつかに対する簡単な機能の説明を以下に示す。

【００３８】

読出トランザクションは、Rd（Sized），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Rd（Sized）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、Rd（Sized）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する場合。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。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。RdBlkSコマンドに応答して、ターゲットノードはキャッシュブロックをソースノードに共用状態で返し得る。一般的に、適切な読出コマンドはソースノードから送信されて、ソースノードから要求された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。

【００３９】

ソースノードにストアされた書込不可能または読出専用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。好ましくは、ValidateBlkおよびChangeToDirtyコマンドは、メモリのみに向けられ、かつコヒーレントなノードによってのみ生成されることができる。

【００４０】
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、かつ割込に応答し得る。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）クリーンな状態のキャッシュブロック（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。
【００４１】
図９、図１３および図１４に、コンピュータシステム１０内の処理ノードが指定されたメモリ位置の読出を試みるときのパケットのフローのいくつかの例を示す。指定されたシステムメモリ位置または対応のシステムメモリ位置は、例示のためにのみ、ターゲット処理ノード７２に関連のシステムメモリ４２１内にあると想定する。システムメモリ４２１は、ターゲット処理ノード７２の一部であるか、またはここに示すようにターゲットノード７２の外部にあってもよい。さらに、メモリ読出トランザクションの間に、指定されたメモリ位置のコピーが既にターゲットノード７２の内部または外部キャッシュメモリに存在する可能性がある。いずれにしても、ソースノード７０が関連の指定されたメモリ位置を読出すために読出コマンドをターゲットノード７２に送信するときはいつでも、パケットのフローは同じままである。いずれの処理ノード１２Ａ－１２Ｄ（図１）もソースノードまたはターゲットノードとして機能し得ることに留意されたい。ソースノードでもターゲットノードでもないノードは残りのノードと呼ばれるが、ここではノード７４および７６である。図９、図１３、図１４において、理解を助けるためにのみ、同じ番号を用いてソースノード、ターゲットノードおよび残りのノードを識別する。これは図９におけるソースノード７０が図１３における同じソースノードであることを表示しない。
【００４２】
上述のように、図１におけるいずれの処理ノードも、特定のトランザクションに依拠してソースノード、ターゲットノードまたは残りのノードとして機能し得る。図９、図１３および図１４の構成は例示のためにのみ示し、これらは処理ノード１２Ａ－１２Ｄの間の同様の実際の接続を示すものではない。すなわち、残りのノード、たとえばノード７６、またはターゲットノード７２は、ソースノード７０に直接接続されないかもしれない。したがって、付加的なパケット経路制御が生じ得る。さらに、図９、図１３および図１４の構成は、図１における回路トポロジーを参照して説明される。２つ以上の処理ノードの間の他の相互接続が企図可能であり、これらのさまざまな相互接続で図９、図１３、および図１４のパケット転送方式を容易に実現化し得ることが理解される。矢印は、従属性と、矢印によって結合されるそれぞれのノードの間で送信されるべきパケットとを示す。一般的に外部に向かう矢印は、対応の入来する従属関係のすべてが発生するまで、生じない。これを図９、図１３、および図１４に示す動作を参照して以下に詳述する。
【００４３】
図９を参照して、読出トランザクションの間のパケット４２のフローを示すが、該読出トランザクションは上述のようにRd(Sized)またはブロック読出（RdBlk、RdBlkS、RdBlkMod）である。ソースノード７０内のプロセッサ（図示せず）は、適切な読出コマンドをターゲットノード７２内のメモリコントローラ（図示せず）に送る。典型的なコマンドパケットは、図６を参照して既に説明した。ソースプロセッサから読出コマンドを受取ると、応答してターゲットメモリコントローラは、以下の２つの動作を行なう。（１）RdResponse（読出応答）パケットをメモリ４２１から要求されたデータと併せてソースノード７０に送る。（２）コンピュータシステム１０内のすべての処理ノードにProbe／Srcコマンドをブロードキャストする。一般的には、Probe／Srcコマンド（より簡単には、プローブコマンド）は、キャッシュブロックがノード内に含まれているかどうか判断するためのそのノードへの要求であり、かつもしキャッシュブロックがそのノードにストアされていれば、そのノードが取るべき動作を表示する。一実施例においては、パケットが１つ以上の宛先

に対してブロードキャストされると、パケットを最初に受信する受信ノードのルータは、そのノードでパケットを終了させ、隣接する処理ノードにそのパケットのコピーを再生成し送信し得る。

【００４４】

上述のこれら２つの動作の正確な実行の順序は、ターゲットノード７２内のさまざまな内部バッファにおける未完了の動作のステータスに依拠し得る。好ましくは、コンピュータシステム１０内の処理ノードの各々は、コマンドパケット、さまざまなコマンドパケットに関連のデータパケット（たとえば、メモリ書込コマンド）、プローブ、応答パケット（たとえば、ProbeResp、SrcDone、TgtDone、MemCancel）、および読出応答（RdResponseパケットおよびその関連のデータパケットの両方を含む）をストアするためのいくつものバッファを含む。データバッファの各々は、たとえば６４バイトサイズのキャッシュブロックのための記憶装置を含み得る。これに代えて、設計要件に基づいて他の便利な記憶容量のいずれをも実現化し得る。

【００４５】

上述のバッファを用いた２つの処理ノード間のバイナリパケットのフローは、上述の「クーポンに基づく」システムを実現化することにより、制御し得る。この実現化においては、送信ノードは、受信ノードの各種のバッファに対するカウンタを含み得る。システムリセットの際に送信ノードはそのカウンタをクリアすることができ、リセット信号がデアサートされたときには、受信ノードは情報パケットを（ＣＭＤフィールドがNopコマンドを識別する、図３に示すものと同様のフォーマットによって）送信ノードに送り、それが各種の利用可能なバッファをいくつ有するかを示す。送信ノードがパケットを受信ノードに送ると、これは関連のカウンタをデクリメントし、特定のカウンタが値０に到達すると、送信ノードプロセッサは関連のバッファへのパケットの送信を停止する。レシーバがバッファを解放すると、これは別の情報パケットをトランスミッタに送り、トランスミッタは関連のカウンタをインクリメントする。トランスミッタは、レシーバがコマンドバッファおよびデータバッファの両方を利用可能にしない限り、メモリ書込動作を開始し得ない。

【００４６】

再び図９に戻ると、ターゲットノード７２内のメモリコントローラは、Probe／Srcコマンドをシステム内の他のノードに送信して、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と、キャッシュブロックの更新されたコピーを有するノードに、キャッシュブロックをソースノードに送らせることとにより、コヒーレンシを維持する。方式は、プローブコマンド内で受信ノードを識別する表示を用いてプローブ応答を受信する。ここで、Probe／Srcコマンド（プローブコマンド）は、残りのノード７４、７６のそれぞれに、ProbeResp（プローブ応答）をソースノードに送信させる。プローブ応答は、動作が起こったことを示し、かつもしキャッシュブロックがノードによって変更されていればデータの送信を含み得る。もし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、図１３に関して後に説明するように、ノードはRdResponse（読出応答）パケットとダーティデータとを送信する。Probe／Srcコマンドは、（ターゲットノード７２を含む）所与の処理ノード内のキャッシュコントローラによって受取られ、ProbeRespおよびRdResponseは、そのキャッシュコントローラによって生成されることができる。一般的には、関連のキャッシュを有する処理ノード内のキャッシュコントローラは、Probe／Srcコマンドに応答してプローブ応答パケットを生成し得る。一実施例においては、処理ノードがキャッシュを有さないとき、その処理ノードはプローブ応答パケットを生成し得ない。

【００４７】

一旦（残りのノード７４および７６からの）プローブ応答と、（ターゲットノード７２からの）要求されたデータを備えたRdResponseとがソースノードにおいて受取られると、ソースノードプロセッサはSrcDone（ソース終了）応答パケットをトランザクション終了の肯定応答としてターゲットノードメモリコントローラ（図示せず）に送信する。読出動作の各々の間の処理ノード間のコヒーレンシを維持するために、ソースノードは（残りのノ

ードからの）すべてのプローブ応答をも受取るまで、ターゲットノード７２からRdResponseを介して受取ったデータを使用し得ない。ターゲットノードがSrcDone応答を受取ったとき、これは（ソースノード７０から受取った）読出コマンドをそのコマンドバッファキューから取除き、次いでこれは同様に指定されたメモリアドレスに対してコマンドへの応答を開始し得る。

【００４８】

送られたコマンドに依拠してプローブ応答を異なった受信ノードへ経路制御する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行ない得る（たとえば、処理ノード間で最も少ない数のパケット送信を用いる）一方で、さらにコヒーレンシが維持されることを確実にする。たとえば、トランザクションのターゲットまたは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であることを示すプローブコマンドをも含み得る。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していたノードからソースノードへダーティデータが引き渡される）。一方で、（トランザクションのターゲットノードのメモリ内でデータが更新される）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。こうして、ターゲットは書込データをいつメモリにコミットするか判断し、かつ書込データとマージされるべきダーティデータのいずれかを受取り得る。

【００４９】

図１０から図１２は、プローブコマンド、読出応答およびプローブ応答パケットのそれぞれの一実施例を示す。図１０Ａのプローブコマンドパケット４４は、図６に示す一般的なコマンドパケットとはやや異なる。ＣＭＤフィールドは、受信ノードにその応答をソースノード７０へ送信することを要求するProbe／Srcコマンドとして、プローブを識別する。上述のように特定の他のトランザクションにおいてはターゲットノード７２はプローブコマンドに対する応答の受信側であり得るが、ＣＭＤフィールドはまたこれらの場合でもそのように示すであろう。さらに、関連する経路制御に依拠して、ソースノード７０またはターゲットノード７２のいずれか、またはこれらの両方が、システム内の他の残りのノードよりも前かまたは同時にProbe／Srcコマンドを受取る可能性もある。プローブコマンドのSrcNodeおよびTgtNodeフィールドは、ソースノードとターゲットノードとをそれぞれ識別し、ソースノードキャッシュコントローラがプローブコマンドに応答することを防ぐであろう。SrcTagフィールドは、図４を参照に先に説明したものと同様に機能する。ＤＭ（データ移動）ビットは、このプローブコマンドに応答してデータ移動が要求されているかどうかを示す。たとえば、ＤＭビットがクリアされていれば、それはいずれのデータ移動もないことを示す。一方で、もしＤＭビットがセットされていれば、プローブコマンドが残りのノード７４または７６のうちの１つの中の、内部（外部）キャッシュ内のダーティブロックまたは共用／ダーティブロックをヒットした場合に、データ移動が要求される。

【００５０】

上述のように、ソースノードからの読出コマンドは、サイズ指定された読出コマンド［Rd（sized）］またはブロック読出コマンド［RdBlk、RdBlkS、またはRdPlkMmd］であり得る。どちらの種類の読出コマンドも好ましくはデータ移動を要求し、よってＤＭビットはターゲットノードのメモリコントローラによってセットされてデータ移動要求を示し得る。異なった実施例においては、ＤＭビットはクリアされている場合に、データ移動を示し、ＤＭビットはセットされている場合に、いずれのデータ移動もないことを示す。

【００５１】

NextStateフィールド４６（図１０Ｂ）は、Probeビットがあった場合、すなわち、１つ以上の残りのノードが、プローブコマンドAddrフィールドによって識別される指定されたメモリ位置のキャッシュコピーを有する場合に、起こるべきステートトランザクションを示す２ビットフィールドである。図１０ＢにNextStateフィールド４６に対する１つの例示的な符号化を示す。ブロック読出コマンドの間では、NextStateフィールドは１であり、よってメモリデータのキャッシュコピーを有する残りのノードは、Probe／Srcコマンドを

受信するとそのコピーを共用としてマークする。一方で、サイズ指定された読出コマンドの間では、NextStateフィールドは０であり、よって、いずれの残りのノードも、メモリ４２１からのデータのキャッシュコピーを有する場合であっても、対応のキャッシュタグを変える必要はない。特定の他のターゲットメモリトランザクション（たとえば特定の書込動作）においても、対応する残りのノード内のキャッシュされたデータを、値２でNextStateフィールド４６によって示されるように、無効としてマークすることが望ましいであろう。

【００５２】

こうしてプローブコマンドは、このNextStateフィールドを介してメモリ読出動作の間のシステム処理ノード間のキャッシュコヒーレンシを維持し得る。ターゲットノードキャッシュコントローラは、ターゲットノードメモリコントローラによるProbe／Srcコマンドブロードキャストの受信の際、およびターゲットノード（内部または外部）キャッシュメモリ内の要求されたデータの発見の際に、プローブ応答パケットを読出応答パケットと併せて送信し得る。後に説明するように、ソースノードは、RdResponseおよびProbeRespパケットによって供給される情報によって要求されたデータに関連のキャッシュタグを更新する。このようにしてソースノードは、（対応するキャッシュタグを介して）これが要求されたデータの排他的または共用コピーを有するかどうか表示し得る。ターゲットノードキャッシュコントローラからのプローブ応答パケットは、たとえばターゲットノードだけが要求されたデータのコピーをそのキャッシュ内に有し、他のいずれの残りのノードも要求されたデータのキャッシュコピーを有さない状況において、助けになり得る。しかしながら、ターゲットノードは、ターゲットノードがソースによって要求されたデータをそのキャッシュ内に有するときに、そのキャッシュ状態を自動的に更新するよう構成されてもよく、したがってこれはターゲットノードキャッシュからデータをソースへ送る。

【００５３】

図１１Ａを参照すると、RdResponseパケット４８に対する例示的な符号化を示す。ターゲットノード７２内のメモリコントローラ（図示せず）は、サイズ指定された読出コマンドまたはブロック読出コマンドのいずれであっても、読出コマンドの各々に応答してRdResponseをソースノード７０に送るよう構成されることができる。これに代えて上述のように、ターゲットノードキャッシュコントローラ（図示せず）は、要求されたデータがターゲットノード内にキャッシュされている場合に、適切な読出応答パケットを送るよう構成されてもよい。典型的には、RdResponseパケット４８の後には要求されたデータを含むデータパケット３８（図７）が続く。サイズ指定された読出動作に対するデータパケットは、最も低いアドレスのデータが最初に返され、残りのアドレスのデータが昇順に返されるよう構成されてもよい。しかしながら、キャッシュブロック読出に対するデータパケットは、要求されたクワッドワード（６４ビット）が最初に返され、残りのキャッシュブロックはインタリーブラッピングを用いて返されるよう構成されてもよい。

【００５４】

RdResponseパケット４８内のCountフィールドは、読出トランザクションを開始する読出コマンド内のCountフィールド（たとえば図６を参照）と同一である。Typeフィールドは元の読出要求のサイズを符号化し、かつCountフィールドと併せて、データパケットのサイズの合計を表示する。Typeフィールドはバイナリ値０または１のいずれかをとり得る。一実施例においては、Typeフィールドは０であるとき、バイトサイズのデータが転送されるべきであることを示し得る。Typeフィールドが１であるとき、クワッドワード（６４ビット）のデータが転送されるべきであることを示し得る。一方、Countフィールドは、Typeフィールドによって示されるそのサイズのデータが、リンクをわたって何回転送されるべきであるかを示し得る。こうして、CountフィールドとTypeフィールドとは組合わされて、転送されるべきデータの合計のサイズを判断し得る。たとえば、８ビット単方向リンクをわたるサイズ指定された読出動作の間、ダブルワードの転送のためにはTypeフィールドは０であって、Countフィールドは３でなければならない［バイナリでは０１１］。

【００５５】

RdResponseパケット４８内のRespNodeフィールドは、読出応答パケットが向けられるべきノードを識別する。SrcNodeフィールドは、トランザクションを開始したノード、すなわちソースノード７０を識別する。読出動作の間、RespNodeおよびSrcNodeフィールドは、同一のノード、すなわちソースノード７０を識別するであろう。図１３を参照して後に説明するように、キャッシュ内に（ターゲットメモリ４２１内の）アドレス指定されたメモリ位置のダーティコピーを有する残りのノードのうちの１つによって、RdResponseが生成されるであろう。データ移動を要求するプローブに応答してノードによって読出応答４８が生成されたことを示すために、プローブビットがセットされることができる。クリアされたProbeビットは、メモリコントローラ（図示せず）、またはターゲットノード７２のキャッシュコントローラ（図示せず）のいずれからRdResponse４８が来たのかを示し得る。

【００５６】

Tgtビットは、ＣＭＤ［５：０］フィールド内のビット位置［０］のビットである。一実施例においては、Tgtビットはセットされている場合に、RdResponse４８がターゲットノード７２内のメモリコントローラ（図示せず）に（たとえば、ある書込トランザクションの間に）宛先決めされていることを示し得る。一方で、Tgtビットはクリアされている場合に、RdResponse４８がソースノード７０に宛先決めされていることを示し得る。こうしてTgtビットは、どのようにデータフローがノード内で内部的に管理されているかを識別する。ある実施例においては、Tgtビットを省いてもよい。

【００５７】

図１１Ｂ内のテーブル５０は、Probeビット、Tgtビット、Typeフィールド、およびCountフィールド間の関係の１つの例を示す。ここで示すように、RdResponse４８がターゲットノード７２のキャッシュコントローラ（図示せず）またはメモリコントローラ（図示せず）から来ている場合はいつでも、Probeビットはクリアされている。一実施例においては、ターゲットノード７２は、（たとえばサイズ指定された読出動作の間）キャッシュブロックサイズよりも小さなデータを供給し得る。TypeフィールドとCountフィールドとは共にソースノード７０に転送されるべきデータのサイズを特定し得る。後に説明するように、残りのノード（ノード７４またはノード７６）のうちの１つがソースノード７０にRdResponseパケットを送るとき、転送されることのできるサイズのデータはキャッシュブロックのみである。この状況において（キャッシュブロックサイズが６４バイトであると想定すると）６４バイトのデータ転送を達成するためには、Countフィールドは７（バイナリ１１１）であり、Typeフィールドは１でなくてはならない。

【００５８】

図１２を参照して、ProbeRespパケット５２の例を示す。一般的に、関連のキャッシュメモリを有する処理ノード（１つ以上の残りのノードまたはターゲットノード７２）は、MissまたはHitNotDirtyを示してProbeRespパケットをソースノード７０に向けることにより、Probe／Srcコマンドに応答する。しかしながら、もし応答するノードが要求されたデータの変更されたキャッシュされたコピーを有すれば、これは代わりに、後に説明するようにRdResponseを送信する。ＣＭＤフィールド、RespNodeフィールド、SrcNodeフィールド、およびSrcTagフィールドについては、既に１つ以上の制御パケットを参照して先に説明した。一実施例においては、ヒットビットがセットされていると、応答するノードがアドレス指定されたメモリ位置の変更されていないキャッシュされたコピーを有することを（ソース処理ノード７２に）示す。別の実施例においては、クリアされたヒットビットが同様の表示を示し得る。こうして、ソースノード７０は、ターゲットノード７２から受取ったデータのブロックを（そのキャッシュ内に）どのようにマークするかについての必要な情報を得る。たとえば、もし残りのノード７４または７６のうちの１つが変更されていない（すなわちクリーンである）アドレス指定されたメモリ位置のコピーを有すれば、ソースノード７０はターゲットメモリコントローラ（図示せず）から受取ったデータブロックをクリーン／共用であるとマークするであろう。一方で、もしこれがサイズ指定された読出動作であれば、ソースノード７０は、読出したデータがキャッシュブロックよりもサイ

ズが小さいために、受取ったデータに関連するそのキャッシュタグを変える必要はない。これは残りのノードを参照した先の説明（図１０Ｂ）と極めて似ている。

【００５９】

図１３に、残りのノードの１つ（ここではノード７６）がそのキャッシュ内にターゲットメモリ位置の変更されたコピー（すなわちダーティコピー）を有する場合のパケットのフローの例、すなわち配置５４を示す。上述のように、ターゲットノードメモリコントローラ（図示せず）は、ソースノード７０から読出コマンドを受取ると、Probe／SrcCommand（プローブコマンド）とRdResponseとを送る。ここで、ターゲットノード７２は関連のキャッシュメモリを有すると想定し、よって、ターゲットノードキャッシュコントローラ（図示せず）は上述のようにソースノード７０にプローブ応答を送る。ターゲットノード７２もまた要求されたデータのキャッシュされたコピーを有する場合には、ターゲットノードキャッシュコントローラ（図示せず）は上述のように、要求されたデータと併せて読出応答パケットをも送る。関連のキャッシュがない場合には、ターゲットノード７２はプローブ応答パケットを送り得ない。

【００６０】

プローブコマンドパケットおよび読出応答パケットの一実施例は、それぞれ図１０Ａおよび図１１Ａに関して先に説明した。しかしながら、図１３の実施例においては、応答ノード７６はプローブコマンドに応答して、そのキャッシュコントローラを介して２つのパケットを送るよう構成される。すなわち、RdRespパケットをソースノード７０内のプロセッサに送り、MemCancel応答をターゲットノードメモリコントローラ（図示せず）に送る。残りのノード７６からの読出応答の後に、プローブコマンドパケット内のＤＭ（データ移動）ビット（図１０Ａ）から要求されて、変更されたキャッシュブロックを含むデータパケットが続く。図１１Ａを参照して先に説明したように、非ターゲットノードからのRdResponseは、そのProbeビットをセットにしてデータブロックのソースがターゲットノード７２ではないことを示し得る。応答ノード７６からのこのRdResponseパケットを介して、ソースノード７０は受取ったデータのキャッシュブロックの状態を（その内部キャッシュ内に）変更／共有としてマークするための表示を得る。

【００６１】

残りのノード７６からのRdResponseパケットは、読出コマンドがサイズ指定された読出トランザクションを識別した場合でも、対応するキャッシュブロックの全体を（変更された状態で）含む。異なった実施例においては、応答非ターゲットノード（ここではノード７６）は、要求されたデータのみを直接ソースノードに送るよう構成されてもよい。この実施例においては、ソースノードに転送されるべきデータのサイズは、プローブコマンドの一部として符号化してもよい。さらなる別の実施例においては、応答ノード７６は要求されたデータのみをターゲットノード７２内のメモリコントローラ（図示せず）に送ってもよく、その後で、ターゲットノードメモリコントローラはデータをソースノード７０に返す。

【００６２】

応答ノード７６からのMemCancel（メモリキャンセル）応答は、ターゲット処理ノード７２のメモリコントローラにソースノード７０からの読出コマンドのさらなる処理を打ち切らせる。言い換えると、MemCancel応答は、ターゲットノードメモリコントローラからのRdResponseパケット（および要求されたデータ）の送信をキャンセルし、かつターゲットノードメモリコントローラによる先行のメモリ読出サイクルさえもキャンセルする効果を有するが、該メモリ読出サイクルとは、もしターゲットノード読出応答バッファからのRdResponseパケットの解放の前、またはメモリ読出サイクルの完了の前に、ターゲットノードメモリコントローラがMemCancel応答をそれぞれ受取っていればソース７０からの読出コマンドに応答して開始されているであろうものである。こうしてMemCancel応答は、２つの主要な目的を達成する。（１）システムメモリ（たとえば、メモリ４２１）が失効したデータを有する場合、比較的長いメモリアクセスを可能な限りなくすことにより、システムメモリバス帯域幅を節約する。これはまたコヒーレントなリンク上の不必要なデータ

転送をも減じる。（２）処理ノード間で最新のキャッシュデータの転送を可能にすることにより、マルチプロセッシングコンピュータシステムにおいてさまざまな処理ノード間のキャッシュコヒーレンシを維持する。

【００６３】

図１の回路構成に含まれる経路制御のために、応答ノード７６からのMemCancel応答パケットのターゲットノード７２への到達は、ターゲットノードメモリコントローラの読出応答パケットの送信または比較的長いメモリ読出サイクルを打ち切らせるのに間に合わないおそれがあることに留意されたい。そのような状況においては、ターゲット処理ノード７２は、読出応答送信またはシステムメモリ読出サイクルに遅すぎる場合には、遅れて到着したMemCancel応答を単に無視し得る。トランザクションが打ち切られる正確な時点は、回路構成、実現化された経路制御、オペレーティングソフトウェア、さまざまな処理ノードを構成するハードウェアなどに依存し得る。ソースノードが、ターゲットノードメモリコントローラからRdResponseを受取るとき、これはこのRdResponse（およびその関連のデータパケット）を単に無視し、代わりに、残りのノード７６からRdResponseパケットと併せて供給されたキャッシュブロックからの要求されたデータを受取る。

【００６４】

MemCancel応答を受取ると、ターゲットノードメモリコントローラは、TgtDone（ターゲット終了）応答をソース処理ノード７０に送信する。TgtDone応答は、それ以前にターゲットノードがRdResponseパケット（および要求されたデータ）をソースノード７０に送ったかどうかにかかわらず、送信される。もしターゲットノードメモリコントローラがそれ以前にRdResponseパケットを送っていなければ、これはRdResponseパケット（および要求されたデータ）の送信をキャンセルし、代わりに、TgtDone応答をソースノード７０に送る。TgtDone応答は、ソースノード７０にキャッシュブロックフィルのソースを伝える機能を果たす。TgtDone応答の存在は、ターゲットノードメモリ４２１またはターゲットノード内部キャッシュ（図示せず）が要求されたデータの失効した状態のものを有することを、ソースノードに示し、よってソースノード７０は残りのノードのうちの１つ（たとえばノード７４または７６）からのキャッシュブロックの変更されたコピーを待たねばならない。

【００６５】

ソースノードプロセッサは、TgtDone応答の受取の前に、応答ノード７６からのRdResponseパケットと併せて送信される変更されたキャッシュブロックを用い得る。しかしながら、ソースノード７０は、SrcDone応答を送る前にそのソースタグ（図６の読出コマンドパケット内のSrcTagフィールド）を再使用し得ないが、これは読出コマンドパケットによって開始されたトランザクション、すなわち読出動作は、読出トランザクションの開始によって生成されたすべての応答をソースノード７０が受取るまで、完了しないおそれがあるためである。したがってソースノード７０は、（送信されていれば）ターゲットノード７２からのRdResponse、ターゲットノードからのTgtDone応答、および他の残りのノードからの他の応答のいずれか(図１４を参照して後に説明）を受取るまで待機し、その後でターゲットノードメモリコントローラへのSrcDone応答を送信する。図９を参照した説明と同様に、図１３におけるSrcDone応答は、ソースノードによって開始されたメモリ読出トランザクションの完了の信号を、ターゲットノードに送る。ターゲットノード７２がRdResponseとTgtDone応答とを送信すると、ソースノードはこれら両方の応答を待ってから、SrcDone応答を介した読出トランザクションの完了に対して肯定応答しなければならないであろう。SrcDone応答はこうして、メモリ読出トランザクションの間のキャッシュブロックのフィループローブ順序の維持を助けるが、これは要求されたデータのソースがターゲットノードメモリコントローラであるか、ターゲットノード内部（または外部）キャッシュであるか、または要求されたデータを含むキャッシュブロックのダーティコピーを有する残りのノードのうちの１つであるかにかかわらない。

【００６６】

図１４を参照すると、ソースノード７０によって開始されるメモリ読出トランザクション

に関するパケットフロー構成５６を示す。この実施例は１つ以上の残りのノード、すなわちノード７４および７６を示し、残りのノードのうちの１つである７６はそのキャッシュ内に要求されたデータを含むメモリブロックのダーティ（変更された）コピーを有すると想定する。図１４に示すさまざまなコマンドおよび応答パケットは、図９から図１３を参照して先に説明したものと同様である。ソースプロセッサは、ノード７６からRdResponseと併せて受取ったデータを、システム内の他の残りのノードのすべて（ここではノード７４のみ）からのプローブ応答をも受取るまで、使用し得ない。図１３を参照して説明したように、ソースノードは開始されたトランザクション、すなわちメモリ読出動作がSrcDone応答の送信によって完全に確立されるまで、SrcTagを再使用し得ない。応答ノード７６からのRdResponse、すべての残りの処理ノードからのプローブ応答、ターゲットノード７２からのTgtDone応答、および（既に送信されていれば）ターゲットノードからのRdResponseが、ソースノード７０によって受取られたときにSrcDone応答が送信される。（図９、図１３、図１４、図１５Ａおよび図１５Ｂの）SrcDoneおよびTgtDone応答は、こうして用いられてコマンドと応答との間のエンドツーエンド肯定応答を提供する。

【００６７】

図１５Ａは、ダーティヴィクティムブロック書込動作の間のパケット５８の例示的なフローを示す。ダーティヴィクティムブロックとは一般的には、ヴィクティムブロック書込動作を発信する処理ノード、すなわちソースノード７０内にあるキャッシュ（図示せず）から排除された、変更されたキャッシュブロックであって、好適なキャッシュブロック置き換えアルゴリズムのいずれかに従って置き換えられる。置き換えのためにダーティヴィクティムブロックが選択されると、ここではターゲットノード７２に関連するメモリ４２１である、対応のシステムメモリ内にVicBlkコマンドを用いてライトバックされる。メモリライトバック動作は、VicBlkパケットを用いて開始され、その後に変更されたヴィクティムキャッシュブロックを含むデータパケットが続く。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受取ったヴィクティムブロックデータをメモリ４２１にコミットするよう準備されると、ターゲットメモリコントローラはTgtDoneパケットをソースノードプロセッサに送る。ソースノードプロセッサは、SrcDoneパケットで応答してデータがコミットされるべきことを表示するか、またはMemCancelパケットで応答してデータがVicBlkコマンドの送信とTgtDoneパケットの受信との間で（たとえば、介入するプローブに応答して）無効化されたことを表示する。

【００６８】

ソースノード７０は、ヴィクティムブロックがシステムメモリ４２１内の適切なメモリ位置に書込まれるためにターゲットノードメモリコントローラ（図示せず）によって受取られるまで、該ヴィクティムブロックを所有することに留意されたい。ターゲットノード７２は、受取ったヴィクティムブロックをそのコマンドデータバッファ内に配置し、ソースノードプロセッサ（図示せず）にTgtDone応答を送り返して、ヴィクティムブロックの受取を表示し得る。ソースノード７０は、TgtDone応答を受取るまで、ヴィクティムブロック内に含まれるデータを含む他のトランザクションを処理し続ける。

【００６９】

図１５Ｂを参照すると、TgtDone応答の前のソースノード７０による無効化プローブの受取を示す、パケット５９の詳細なフローを示す。上述のように、制御またはデータパケットのターゲットノードへの引渡しは、システム１０内に含まれる経路制御に依存する。図１５Ｂに示すように、ソースノード７０からのVicBlkコマンドおよびヴィクティムブロックデータパケット（図１５Ｂにおいてデーター１と示す）は、残りのノード７４のうちの１つを含む経路を通過し得る。システム１０内のパケット伝搬にかかわる時間は、一般的には経路内に介在する処理ノードの数と、介入するノードが受取ったコマンドおよびデータパケットを経路上の他の処理ノードに、またはこの場合のようにターゲットノード７２に、伝送するためにかかる時間とに依存する。

【００７０】

図１５Ｂは、ソースノード７０がVicBlkコマンドとヴィクティムブロック（データ－１）

とを送信した後であって、このVicBlkコマンドがターゲットノードメモリコントローラ（図示せず）によって受取られる前に、残りの処理ノードのうちの１つ（ここではノード７６）がRdBlkModコマンドをターゲットノード７２に送る１つの例を示す。ノード７６からのRdBlkModコマンドは、ソースノード７０からのヴィクティムブロックに対する宛先である、メモリ４２１内の同一のメモリ位置を特定し得る。先に簡単に説明したように、キャッシュブロックの書込可能なコピーが所望である場合にはRdBlkModコマンドを用い得る。RdBlkModコマンドは、読出コマンドの１種であることから、RdBlkModコマンド実行の間に図９から図１４を参照して示し説明したさまざまな信号フローパターンが生じ得る。

【００７１】

RdBlkModコマンドに応答して、図９を参照して説明したように、ターゲットノード７２はプローブコマンドパケット（図１０Ａ）をソースノード７０と他の残りのノード７４とに送信し得る。ソースノード７０は、要求されたデータすなわちヴィクティムブロックと併せて読出応答パケット（図１１Ａ）を送ることにより、プローブコマンド（無効化プローブとしても知られる）に応答するが、これは（ｉ）ソースノードは指定されたメモリ位置の変更されたコピー（すなわちヴィクティムブロック）を有し、かつ（ii）ソースノードは、先行して送信したヴィクティムブロックのターゲットノードによる受取と受認とを表示する、ターゲットノード７２からのターゲット終了応答をまだ受取っていないためである。ソースノード７０はまた、図１３を参照して先に説明したように、メモリキャンセル応答（図１５Ｂには図示せず）をターゲットノード７２に送ってもよい。他の残りのノード７４から読出コマンドのソースノード７６へのプローブ応答は、より明確にするため図１５Ｂには示さない。

【００７２】

処理ノード７６はまた、ソースノード７０からの受取ったヴィクティムブロックを変更し、かつ変更されたデータ（データ－２）をターゲットノード７２に送信して、データ－２をシステムメモリ４２１内の対応するメモリ位置にコミットさせることができる。図１５に示す状況においては、VicBlkコマンドおよびもとのヴィクティムブロック（データ－１）は、ターゲットノードが変更されたヴィクティムブロック（データ－２）を受取った後で、ターゲットノードに到着する。もとのヴィクティムブロック（データ－１）を受取ると、ターゲットノードメモリコントローラ（図示せず）は、ターゲット終了応答をソースノード７０に送信して、ヴィクティムブロックデータパケット（データ－１）の受認を表示する。ターゲットノード７２はデータ送信事象の履歴を追跡し得ないために、ターゲットノード７２が、変更されたヴィクティムブロック（データ－２）を含むメモリ位置に、（より早く送信されたが）遅れて到着した、失効したヴィクティムブロック（データ－１）を上書きすることを防ぐことが望ましい。この場合ソースノード７０は、ソースノードがターゲットノード７２からターゲット終了応答を受取ったときに、SrcDone応答の代わりにMemCancel応答をターゲットノードメモリコントローラに送信する。ソースノード７０からのMemCancel応答はこうして、ターゲットノード７２が失効したデータ（データ－１）を共通のメモリ位置に上書きすることを防ぐ。

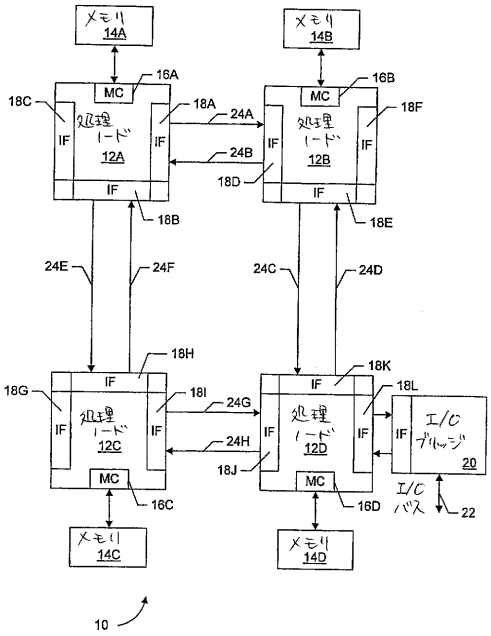
【００７３】

一般的に、ソースノード７０は、ソースノードが無効化プローブをTgtDoneメッセージを受取る前であって、VicBlkコマンドおよびヴィクティムブロックデータパケットを送信した後であればいつでも、ターゲットノードからのTgtDoneメッセージに応答してメモリキャンセルメッセージ（MemCancel）を送る。メモリキャンセル応答はこうして、たとえば、ソースノード７０以外の処理ノード（ここではノード７６）が、図１５Ｂに示されるように、ソースノード７０によって先立って送られたヴィクティムブロック（データ－１）内に含まれるデータを変更する意図を表示する場合の、システム内のさまざまな処理ノード間のキャッシュコヒーレンシを維持する。メモリキャンセル応答はまた、システムメモリ４２１にコミットされるべきデータがもはや有効でない場合に、ターゲットノードメモリコントローラが長いメモリ書込動作を開始することを防ぐことにより、システムメモリ帯域幅を節約し得る。

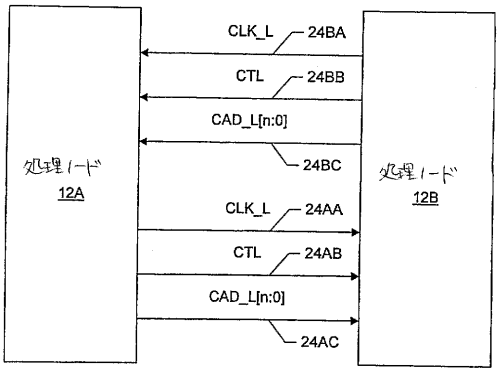
【００７４】
これに代えてソースノードプロセッサは、TgtDone応答が無効化プローブに先立って受取られた場合、図１５Ｂに点線の矢印によって示すように、SrcDoneパケットをターゲットノードメモリコントローラに送ってもよい。言い換えると、ソースノードは、ヴィクティムブロックがまだ有効であれば、TgtDoneメッセージを受取った後にSrcDone応答をターゲットノードに送ってもよい。図１５Ｂに示す状況においては、ソースノードは、プローブコマンドがターゲット終了応答の後に到着したときに、読出応答パケットの代わりにプローブ応答パケット（図１２）を送ってもよいが、これはソースノードがソース終了メッセージを送ることによりヴィクティムブロックをターゲットノードに解放すると、もはやヴィクティムブロックを有し得ないためである。SrcDone応答は、ソースノードプロセッサによって開始されたダーティヴィクティムブロック（データ－１）書込動作の終了信号を送る。メモリキャンセル応答は必要ではないが、これはたとえば、変更されたヴィクティムブロック（データ－２）を含む同じメモリ位置への後の書込動作が、先行の（よって失効した）ヴィクティムブロック（データ－１）を正確に上書きするためである。処理ノード間のキャッシュコヒーレンシはこうして適切に維持される。
【００７５】
ヴィクティムブロックコマンド（VicBlk）はシステムメモリに対してのみ向けられ、かつコヒーレントな処理ノード（すなわち図１における処理ノード１２Ａ－１２Ｄのうちの１つ）によってのみ生成されることができ、たとえばＩ／Ｏブリッジ２０によっては生成されないことに留意されたい。SrcDoneおよびTgtDone応答とは、上述のようにコマンドと応答とのエンドツーエンド肯定応答を提供するために用いられる。
【００７６】
最後に、図１６Ａはメモリ書込動作に含まれるトランザクション（サイズ指定された読出またはブロック読出動作）に対する例示的なフローチャート６０を示す。さらに、図１６Ｂはダーティヴィクティムブロック書込動作に関連するトランザクションに対する例示的なフローチャート６２を示す。図１６Ａおよび図１６Ｂのフローチャート内のさまざまなブロックに関連の詳細のすべては、図９から図１５Ｂを参照に先に説明した。（コマンドパケットと応答パケットとを含む）さまざまな制御パケットとデータパケットとが、図３から図８および図１０から図１２において例示的な実施例を用いて示された。システムは同様の目的のために他の制御およびデータパケットを実現化し得るが、異なったフォーマットおよび符号化を使用する。図１のシステム構成におけるコマンドおよび応答パケットを含むこのメッセージ通信方式は、他のシステム構成においても実現化し得る。
【００７７】
先の説明は、マルチプロセッシングコンピュータシステム環境におけるキャッシュコヒーレントなデータ転送通信方式を開示する。データ転送通信方式は、ターゲット処理ノードにより遅いシステムメモリバスでの比較的長いメモリ読出または書込動作を打切らせることにより、システムメモリ帯域幅を節約し得る。コマンドと応答とのエンドツーエンド肯定応答は、マルチプロセッシングシステムを通してキャッシュコヒーレンシを維持し得る。
【００７８】
この発明はさまざまな変形および代替形に対処するものであるが、その特定の実施例は本明細書と図面中に例示の目的でのみ示された。しかしながら、図面と詳細な説明とはこの発明の範囲を開示された特定の形に限定するものではなく、反対に、すべてのそのような変形、等価物および代替物を、前掲の特許請求の範囲に規定されるこの発明の精神および範囲に入れるものであることを理解されたい。
【００７９】
【産業的用途】
この発明は一般的にコンピュータシステムに適する。
【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からの、１対の処理ノードの間の相互接続の一実施例の詳細な図である。
【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】　コマンドパケットの一実施例のブロック図である。
【図７】　データパケットの一実施例のブロック図である。
【図８】　図１のコンピュータシステムにおい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９】　メモリ読出動作に対応するパケットのフローの例を示す図である。
【図１０Ａ】　プローブコマンドパケットの一実施例のブロック図である。
【図１０Ｂ】　図１０Ａのプローブコマンドパケットにおける NextState フィールドに対する符号化の一実施例のブロック図である。
【図１１Ａ】　読出応答パケットの一実施例のブロック図である。
【図１１Ｂ】　一実施例において図１１Ａの読出応答パケットの Probe、Tgt および Type フィールドの間の関係を示す図である。
【図１２】　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１３】　メモリキャンセル応答にかかわる、パケットのフローの例を示す図である。
【図１４】　プローブコマンドとメモリキャンセル応答とを組合せるメッセージ通信方式を示すパケットのフローの例を示す図である。
【図１５Ａ】　ヴィクティムブロック書込動作の間のパケットのフローの例を一般的に示す図である。
【図１５Ｂ】　ヴィクティムブロック書込動作の間の無効化プローブとメモリキャンセル応答とを示すパケットのフローを詳細に示す図である。
【図１６Ａ】　メモリ読出動作に含まれるトランザクションに対する例示的なフローチャートの図である。
【図１６Ｂ】　ヴィクティムブロック書込動作に含まれるトランザクションに対する例示的なフローチャートの図である。

【図１】



【図２】



【図３】

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

30

【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	TgtNode [1:0]		CMD[5:0]					
2	SrcTag [1:0]		SrcNode[3:0]				TgtNode [3:2]	
3	Count		SrcTag[6:2]					
4	Addr[7:0]							
5	Addr[15:8]							
6	Addr[23:16]							
7	Addr[31:24]							
8	Addr[39:32]							

36

【図７】

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]							

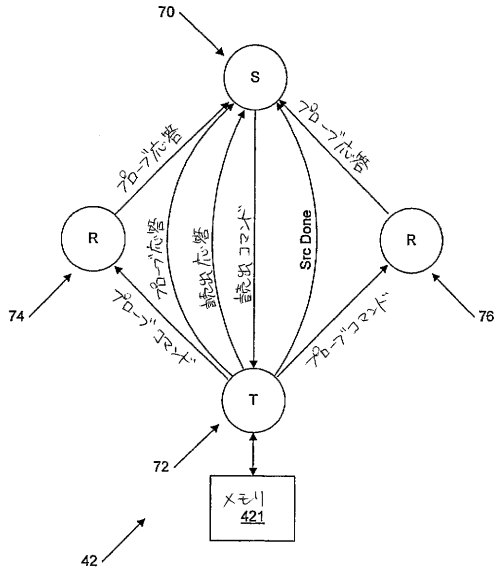
38

【図８】

CMD符号	コマンド	パケットタイプ
000000	Nop	Info
000001	Interrupt Broadcast	Command
000011	Reserved	-
000100	Probe/Src	Command/Address
000101	Probe/Tgt	Command/Address
00011x	Reserved	-
001000	RdBlkS	Command/Address
001001	RdBlk	Command/Address
001010	RdBlkMod	Command/Address
001011	ChangeToDirty	Command/Address
001100	ValidateBlk	Command/Address
001110	CleanVicBlk	Command/Address
001101	Interrupt Target	Command
001111	VicBlk	Command/Address/Data
01xxxx	Read(Sized)	Command/Address
10xxxx	Write(Sized)	Command/Address/Data
11000x	RdResponse	Rd 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

40

【図９】



【図１０Ａ】

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]							

44

【図１０Ｂ】

Next State [1:0]	ネクストステート
0	変化なし
1	クリーン → 共用 ダーティ → 共用/ダーティ
2	無効

46

【図１１Ａ】

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]				TgtNode [3:2]	
3	Count		SrcTag[6:2]					
4	Rsv						Probe	Type

48

【図１１Ｂ】

Probe	Probe	Type, Count
0	X	データパケットの符号化長さ
1	0	キャッシュブロック： Typeは1, Countは7であるべき
1	1	キャッシュブロック： Typeは1, Countは7であるべき

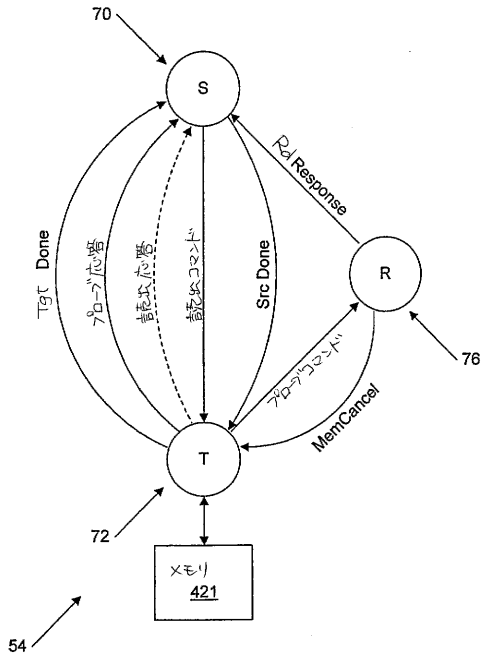
50

【図１２】

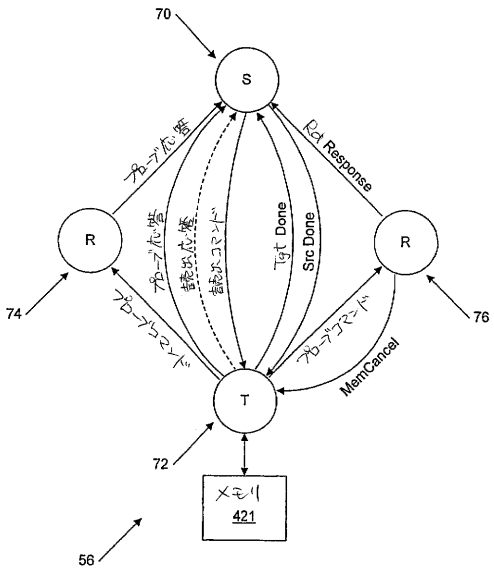
ビット時間	7	6	5	4	3	2	1	0
1	RespNode [1:0]		CMD[5:0]					
2	SrcTag [1:0]		SrcNode[3:0]				RespNode [3:2]	
3	Rsv		Hit	SrcTag[6:2]				
4								

52

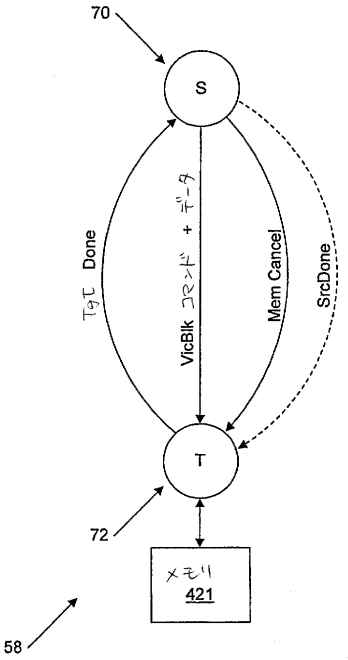
【図１３】



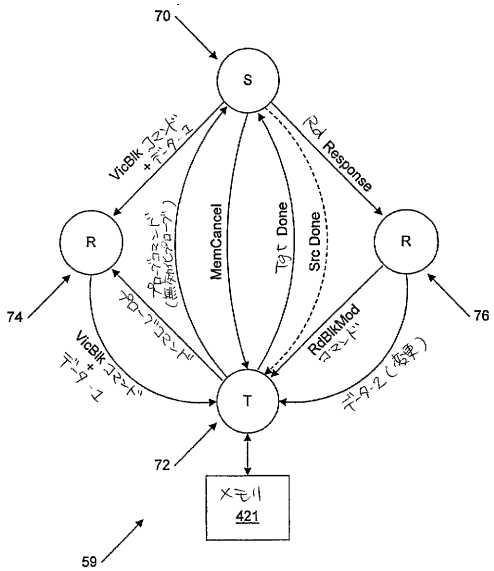
【図１４】



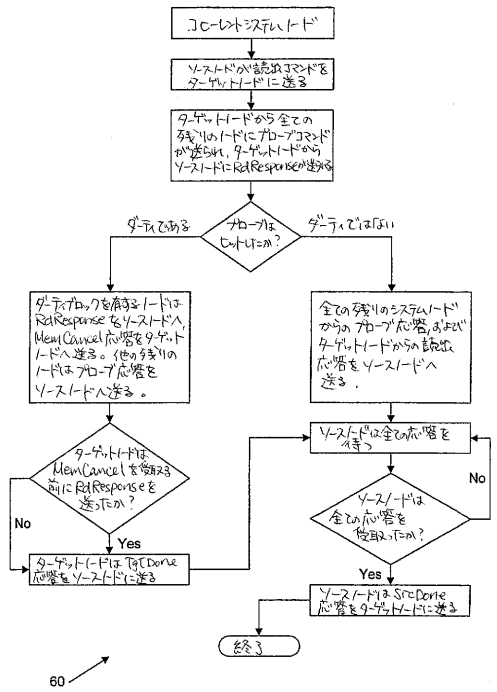
【図１５Ａ】



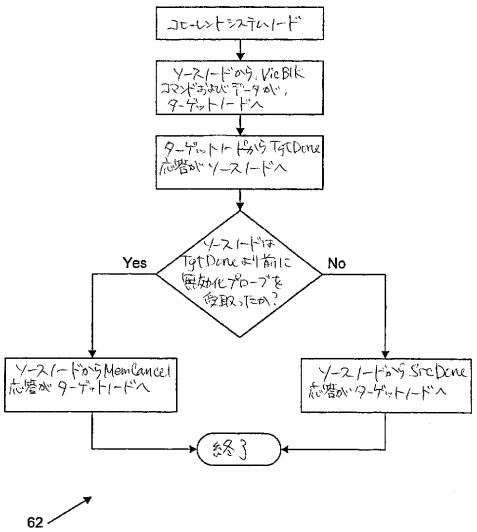
【図１５Ｂ】



【図１６Ａ】



【図１６Ｂ】



フロントページの続き

(31)優先権主張番号 09/217,649
(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)
(31)優先権主張番号 09/370,970
(32)優先日 平成11年8月10日(1999.8.10)
(33)優先権主張国 米国(US)

(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、イリス・ウェイ、２１０

審査官 高瀬 勤

(56)参考文献 特開平０６－１１０８４４（ＪＰ，Ａ）
特開平０６－２７４４６１（ＪＰ，Ａ）
特表２００２－５３３８１２（ＪＰ，Ａ）
特開２０００－１３２５３１（ＪＰ，Ａ）
特開２０００－０７６１３０（ＪＰ，Ａ）
特開平１０－１０５４６４（ＪＰ，Ａ）
特開平０８－０１６４７４（ＪＰ，Ａ）
特開平０６－３１４２３９（ＪＰ，Ａ）
特開平０５－１３４９９１（ＪＰ，Ａ）
特開平０２－２０５９６３（ＪＰ，Ａ）
特開昭６０－２００３５１（ＪＰ，Ａ）
特表平０６－５０９６７１（ＪＰ，Ａ）
特開平０４－１１３４４４（ＪＰ，Ａ）
特開昭６３－２２３９４８（ＪＰ，Ａ）
特開昭５８－０６４８４６（ＪＰ，Ａ）
特開平０９－０９１２６２（ＪＰ，Ａ）
特開平０５－２１０５８４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 12/08