(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6543246号
(P6543246)
(45)発行日　令和1年7月10日(2019.7.10)　(24)登録日　令和1年6月21日(2019.6.21)

(51)Int.Cl.　　F I
H04L 29/10 (2006.01)　　H04L 13/00 309C
H04L 13/08 (2006.01)　　H04L 13/08

請求項の数 15　(全 30 頁)

(21)出願番号　特願2016-523907(P2016-523907)
(86)(22)出願日　平成26年10月17日(2014.10.17)
(65)公表番号　特表2016-535483(P2016-535483A)
(43)公表日　平成28年11月10日(2016.11.10)
(86)国際出願番号　PCT/AU2014/000994
(87)国際公開番号　WO2015/054738
(87)国際公開日　平成27年4月23日(2015.4.23)
審査請求日　平成29年10月4日(2017.10.4)
(31)優先権主張番号　2013245529
(32)優先日　平成25年10月18日(2013.10.18)
(33)優先権主張国　オーストラリア(AU)

(73)特許権者　511113763
ゾモジョ・ピーティーワイ・リミテッド
オーストラリア国、ニューサウスウェールズ州　２０００、シドニー、クラレンス・ストリート　７６－８０、レベル　６
(74)代理人　100121083
弁理士　青木　宏義
(74)代理人　100083286
弁理士　三浦　邦夫
(74)代理人　100138391
弁理士　天田　昌行
(74)代理人　100137903
弁理士　菅野　亨

最終頁に続く

(54)【発明の名称】ネットワークインターフェイス

(57)【特許請求の範囲】
【請求項１】
ネットワークインターフェイスを介して外部ネットワークから複数のデータフレームを受信する工程と、
前記ネットワークインターフェイスからの前記データフレームを、内部システムバスを介して、ホストコンピュータシステムのメモリ内の受信バッファに転送する工程と、
前記ネットワークインターフェイスが、前記ホストコンピュータシステムからの介入なしに、前記受信バッファ内のデータフレームの配置を規定する工程と、
前記ネットワークインターフェイスからの各受信データフレームを、開ループ書き込みプロトコルを使って、前記受信バッファに転送する工程であって、前記受信データフレームは前記受信バッファ内の連続したメモリ範囲に連続的に書き込まれ、前記受信データフレームは、前記ホストコンピュータシステムからの読み出し確認無しに、前記受信バッファを繰り返し上書きする工程と、
を含むデータ受信方法。
【請求項２】
複数のデータフレームを保持できる保存容量を有する連続したメモリ内に、前記受信バッファを配置する工程を含む請求項１記載の方法。
【請求項３】
前記配置されたメモリを、個別にアクセスできる複数の連続したスロットであって、同じ保存容量を有するスロットに分割する工程を含む請求項２記載の方法。

【請求項４】
受信データフレームを、前記受信バッファ内の前記各スロットに連続的に書き込む工程を含む請求項３記載の方法。
【請求項５】
前記スロットの保存容量を超える大きさの前記受信データフレームをフレームフラグメントに分割し、前記フレームフラグメントを前記受信バッファ内の連続したスロットに割り当てる工程を含む請求項３及び４のいずれか記載の方法。
【請求項６】
連続した複数のスロットのフレームデータの関係を定義することにより、フラグメントに分割されたデータフレームを復元するための制御データを生成する工程を含む請求項５記載の方法。
【請求項７】
前記制御データが前記フレームフラグメントと同じバストランザクションの対応するスロットの最後に書き込まれるように、前記制御データを各フレームフラグメントの最後に付加する工程を含む請求項６記載の方法。
【請求項８】
前記制御データが、前記対応するスロットに割り当てられたフレームフラグメントの長さも定義する請求項７記載の方法。
【請求項９】
外部ネットワークデータから複数のデータフレームを受信するネットワーク面と、
ホスト内部システムバスを介してホストコンピュータシステムのメモリ内に配置される受信バッファに、前記データフレームを転送するホスト面と、
前記ホストコンピュータシステムからの介入なしに、前記受信バッファ内の前記データフレームの割り当てを規定する制御システムと、
開ループプロトコルを使って、前記受信データフレームのそれぞれを前記受信バッファに転送する書き込みコントローラと、を具備し、
前記書き込みコントローラが、前記受信バッファ内の連続したメモリ範囲に前記データフレームを連続的に書き込み、
前記開ループプロトコルは、前記ホストコンピュータシステムからの読み出し確認無しに、前記受信バッファを繰り返し上書きするネットワークインターフェイス。
【請求項１０】
個別にアクセスできる複数の連続したスロットであって同じ保存容量を有するスロットを、前記受信バッファに配置された連続したメモリ内に配置するバッファ管理モジュールを具備する請求項９記載のネットワークインターフェイス。
【請求項１１】
前記受信データフレームを前記受信バッファ内の各スロットに連続的に転送する書き込みコントローラを具備する請求項１０記載のネットワークインターフェイス。
【請求項１２】
前記スロットの保存容量を超える大きさの前記受信データフレームをフレームフラグメントに分割し、前記フレームフラグメントを前記受信バッファ内の連続したスロットに割り当てるフレーム管理モジュールを具備する請求項１１記載のネットワークインターフェイス。
【請求項１３】
前記フレーム管理モジュールは、連続した複数のスロットのフレームデータの関係を定義することにより、フラグメントに分割されたデータフレームを復元するための制御データを生成する請求項１２記載のネットワークインターフェイス。
【請求項１４】
前記書き込みコントローラは、前記制御データが前記フレームフラグメントと同じバストランザクションの対応するスロットの最後に書き込まれるように、前記制御データを各フレームフラグメントの最後に付加する請求項１３記載のネットワークインターフェイス

。
【請求項１５】
前記制御データが、前記対応するスロットに割り当てられたフレームフラグメントの長さも定義する請求項１４記載のネットワークインターフェイス。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、コンピュータシステムのネットワークインターフェイス及びデータ通信の管理方法に関する。
【背景技術】
【０００２】
ネットワークインターフェイスは、コンピュータシステムとデータネットワークとの間の通信を促進する。典型的なネットワークインターフェイスは、外部ネットワークとの通信リンクを確立する専用のハードウエアを有する。そのハードウエアはホストコンピュータシステムの他の構成部分と一体化されていてもよい。専用のファームウエア及び／又はソフトウエアがネットワークインターフェイスに組み込まれていてもよい。ネットワークインターフェイスの一般の名称には、ネットワークインターフェイスカード（ＮＩＣ）、ネットワークインターフェイスコントローラ、ネットワークインターフェイスアダプタ、及びネットワークインターフェイスユニットがある。
【０００３】
ネットワークケーブルコンピュータシステムは一般に少なくとも一つの専用ネットワークインターフェイスを有している。ネットワークインターフェイスは、ホストコンピュータシステムの内部システムバス（例えばＰＣＩ又はＰＣＩエキスプレスバス）と外部データネットワーク（例えばイーサネット（登録商標）ネットワーク）との橋渡しとなる。ネットワークインターフェイスは他のシステム構成部分と一体化されたり（例えばシステムマザーボードに組み込まれたオンボードネットワークインターフェイス）、モジュラ（例えばＰＣＩ－ｅカード）として供給されてもよい。
【０００４】
従来のネットワークインターフェイスは二つの主要な機能を有する。それは、外部ネットワークからデータを受信することと、ホストコンピュータシステムから外部ネットワークにデータを送信することである。ネットワークインターフェイスは、これらの動作の間、ホストコンピュータシステムのシステムバスとインタラクション（情報をやり取り）し、データ交換を管理する。これらのインタラクションはプロトコル（通常、ネットワークインターフェイスにより定義され、そのネットワークインターフェイスのドライバーにより実装される。）に従っており、そのプロトコルにより、ホストコンピュータシステムで実行されているソフトウエアアプリケーションがネットワークインターフェイスとインタラクションできる。
【０００５】
従来のインターフェイスドライバープロトコルは、一般的に通信遅延を犠牲にしてリソースの最適化を優先していた。従来のドライバープロトコルを用いたネットワークインターフェイスとホストコンピュータシステムとの間のデータのやり取りには、データ交換を調整するための制御メッセージの交換（例えば、データ配置のポインタや読取り確認の交換）が含まれている。これが、システム間のネットワークデータの転送を遅らせるような重大な遅延を導く。
【発明の概要】
【０００６】
この本発明の第一の態様によれば、ネットワークインターフェイスを介して外部ネットワークからデータを受信する工程と、そのデータを内部システムバスを介してホストコンピュータシステム内のメモリに転送する工程とを具備するデータ受信方法を提供する。ネットワークインターフェイスは、ホストコンピュータシステムの介入なしに、ホストコン

ピュータシステムのメモリ内のデータの配置を規定する。
【０００７】
この本発明の第二の態様によれば、データ受信方法は、
ネットワークインターフェイスを介して外部ネットワークから複数のデータフレームを受信する工程と、
ネットワークインターフェイスからのデータフレームを、内部システムバスを介して、ホストコンピュータシステムのメモリ内の受信バッファに転送する工程と、
ネットワークインターフェイスが、ホストコンピュータシステムからの介入なしに、受信バッファ内のデータフレームの配置を規定する工程と、を含む。
【０００８】
一実施例において、データ受信方法は、ネットワークインターフェイスからの各受信データフレームを、開ループ書き込みプロトコルを使って、受信バッファに転送する工程を含む。
【０００９】
一実施例において、データ受信方法は、受信データフレームを受信バッファ内の連続したメモリ範囲に連続的に書き込みする工程を含む。
【００１０】
一実施例において、データ受信方法は、複数のデータフレームを保持できる保存容量を有する連続したメモリ内に、受信バッファを配置する工程を含む。
【００１１】
一実施例において、データ受信方法は、配置されたメモリを、個別にアクセスできる複数の連続したスロットであって、同じ保存容量を有するスロットに分割する工程を含む。
【００１２】
一実施例において、データ受信方法は、受信データフレームを、前記受信バッファ内の前記各スロットに連続的に書き込む工程を含む。
【００１３】
一実施例において、データ受信方法は、受信バッファを、前記ホストコンピュータシステムからの読み出し確認無しに、繰り返し上書きする工程を含む。
【００１４】
一実施例において、データ受信方法は、スロットの保存容量を超える大きさの受信データフレームをフレームフラグメントに分割し、フレームフラグメントを受信バッファ内の連続したスロットに割り当てる工程を含む。
【００１５】
一実施例において、データ受信方法は、連続した複数のスロットのフレームデータの関係を定義することにより、フラグメントに分割されたデータフレームを復元するための制御データを生成する工程を含む。
【００１６】
一実施例において、データ受信方法は、制御データが前記フレームフラグメントと同じバストランザクションの対応するスロットの最後に書き込まれるように、制御データを各フレームフラグメントの最後に付加する工程を含む。
【００１７】
一実施例において、制御データが、対応するスロットに割り当てられたフレームフラグメントの長さも定義する。
【００１８】
この本発明の第三の態様によれば、ネットワークインターフェイスは、外部データネットワークからデータを受信するネットワーク面と、内部システムバスを介してホストコンピュータシステム内のメモリにデータを送信するホスト面と、ホストコンピュータシステムからの介入なしに、ホストコンピュータシステムのメモリ内のデータの配置を規定する制御システムを具備する。
【００１９】

この本発明の第四の態様によれば、ネットワークインターフェイスは、
外部ネットワークデータから複数のデータフレームを受信するネットワーク面と、
ホスト内部システムバスを介してホストコンピュータシステムのメモリ内に配置される受信バッファに、データフレームを転送するホスト面と、
前記ホストコンピュータシステムからの介入なしに、受信バッファ内の前記データフレームの割り当てを規定する制御システムと、を具備する。

【００２０】
一実施例において、ネットワークインターフェイスは、開ループプロトコルを使って、受信データフレームのそれぞれを受信バッファに転送する書き込みコントローラを具備する。

【００２１】
一実施例において、書き込みコントローラが受信バッファ内の連続したメモリ範囲にデータフレームを連続的に書き込む。

【００２２】
一実施例において、ネットワークインターフェイスは、個別にアクセスできる複数の連続したスロットであって同じ保存容量を有するスロットを、受信バッファに配置された連続したメモリ内に配置するバッファ管理モジュールを具備する。

【００２３】
一実施例において、ネットワークインターフェイスは、受信データフレームを受信バッファ内の各スロットに連続的に転送する書き込みコントローラを具備する。

【００２４】
一実施例において、書き込みコントローラが、ホストコンピュータシステムからの読み出し確認無しに、受信バッファを繰り返し上書きする。

【００２５】
一実施例において、ネットワークインターフェイスは、スロットの保存容量を超える大きさの受信データフレームをフレームフラグメントに分割し、フレームフラグメントを受信バッファ内の連続したスロットに割り当てるフレーム管理モジュールを具備する。

【００２６】
一実施例において、フレーム管理モジュールが、連続した複数のスロットのフレームデータの関係を定義することにより、フラグメントに分割されたデータフレームを復元するための制御データを生成する。

【００２７】
一実施例において、書き込みコントローラが、制御データがフレームフラグメントと同じバストランザクションの対応するスロットの最後に書き込まれるように、制御データを各フレームフラグメントの最後に付加する。

【００２８】
一実施例において、制御データが、対応するスロットに割り当てられたフレームフラグメントの長さも定義する。

【００２９】
この本発明の第五の態様によれば、バッファアクセスプロトコルは、
メモリバッファ内の複数のスロットから書き込みイタレーションカウンタを読み出す工程であって、書き込みイタレーションカウンタはスロットのそれぞれにいつデータが最後に書き込まれたかを定義する工程と、
データを受信する次のスロットを規定する書き込みアドレスを定義する工程であって、書き込みアドレスは書き込みイタレーションカウンタから定義される工程と、
スロットの書き込み更新を検出するために書き込みアドレスを繰り返しポーリングする工程と、を具備する。

【００３０】
この本発明の第六の態様によれば、バッファアクセスプロトコルは、
循環バッファ内の複数のデータスロットにアクセスし、アクセスしたスロットのそれぞ

れから書き込みイタレーションカウンタを読み出す工程であって、書き込みイタレーションカウンタはデータが各スロットに最後に書き込まれたときの書き込みイタレーションを規定する工程と、
データにより上書きされる次のスロットを定義する循環バッファの書き込みアドレスを定義する工程であって、書き込みアドレスは、隣接スロットに割り当てられた書き込みイタレーションカウンタへの移行により定義される工程と、
書き込み更新を検出するために書き込みアドレスを繰り返しポーリングする工程であって、書き込み更新は対応するスロットの書き込みイタレーションカウンタの変化から判定される工程と、を含む。

【００３１】
一実施例において、バッファアクセスプロトコルは、
循環バッファ内の所定のスロットから書き込みイタレーションカウンタをコピーし、コピーした書き込みイタレーションカウンタから読み出しイタレーションカウンタを作成する工程と、
読み出しイタレーションカウンタを循環バッファ内の連続したスロットに割り当てられた複数の書き込みイタレーションカウンタと比較して書き込みアドレスを特定する工程と、を含む。

【００３２】
一実施例において、バッファアクセスプロトコルは、
循環バッファ内の所定のスロットで読み出しポインタを初期化し、
所定のスロットに割り当てられた書き込みイタレーションカウンタを読み出しイタレーションカウンタと比較する工程と、
読み出しポインタが書き込みアドレスと一致するまで、読み出しポインタをインクリメントして、循環バッファ内の連続するスロットにアラインする工程と、を含む。

【００３３】
一実施例において、バッファアクセスプロトコルは、書き込み更新を検出した後に、読み出しポインタをインクリメントして連続するスロットにアラインする工程を含む。

【００３４】
一実施例において、バッファアクセスプロトコルは、現在の書き込みイタレーションを規定する循環バッファのループカウンタを保持する工程であって、ループカウンタは各書き込みイタレーションの最後にインクリメントされる工程を含む。

【００３５】
一実施例において、バッファアクセスプロトコルは、個々のスロットへの書き込みトランザクションと共に、循環バッファ内の個々のスロットにループカウンタを書き込む工程であって、ループカウンタは読み出し同期のための書き込みイタレーションカウンタを規定する工程を含む。

【００３６】
一実施例において、バッファアクセスプロトコルは、
書き込み更新が検出されたとき、書き込みアドレスと一致するスロットからデータを抽出する工程と、
データの抽出の後に書き込みイタレーションカウンタを確認し、抽出プロセスを有効化する工程と、
書き込み更新の検出の後に書き込みイタレーションカウンタが変わっていた場合に読み出しエラーを生成する工程と、を含む。

【００３７】
一実施例において、バッファアクセスプロトコルは、
書き込み更新の後に書き込みアドレスに対応する現在のスロットから長さリファレンスを読み出す工程と、
長さリファレンスが、現在のスロットがフレームフラグメントを含んでいることを示した場合、連続するスロットからのデータと現在のスロットとを結合する工程と、を含む。

【００３８】
この本発明の第七の態様によれば、バッファアクセスシステムは、
メモリバッファ内の複数のスロットから書き込みイタレーションカウンタを読み出す参照モジュールであって、書き込みイタレーションカウンタは、スロットのそれぞれにいつデータが最後に書き込まれたかを定義する参照モジュールと、
データを受信する次のスロットを規定する書き込みアドレスを定義する同期モジュールであって、書き込みアドレスは書き込みイタレーションカウンタから定義される同期モジュールと、
スロットの書き込み更新を検出するために書き込みアドレスを繰り返しポーリングする書き込みモニターと、を具備する。
【００３９】
この本発明の第八の態様によれば、バッファアクセスシステムは、
循環バッファ内の複数のデータスロットにアクセスし、アクセスしたスロットのそれぞれから書き込みイタレーションカウンタを読み出す参照モジュールであって、書き込みイタレーションカウンタは、データが各スロットに最後に書き込まれたときの書き込みイタレーションを規定する参照モジュールと、
データにより上書きされる次のスロットを定義する前記循環バッファの書き込みアドレスを定義する同期モジュールであって、書き込みアドレスは、隣接スロットに割り当てられた書き込みイタレーションカウンタへの移行により定義される同期モジュールと、
書き込み更新を検出するために書き込みアドレスを繰り返しポーリングする書き込みモニターであって、書き込み更新は対応するスロットの書き込みイタレーションカウンタの変化から判定される書き込みモニターと、を具備する。
【００４０】
一実施例において、参照モジュールが、循環バッファ内の所定のスロットから書き込みイタレーションカウンタをコピーし、コピーした書き込みイタレーションカウンタから読み出しイタレーションカウンタを作成し、
同期モジュールが、読み出しイタレーションカウンタを循環バッファ内の連続したスロットに割り当てられた複数の書き込みイタレーションカウンタと比較して書き込みアドレスを特定する。
【００４１】
一実施例において、同期モジュールが、循環バッファ内の所定のスロットで読み出しポインタを初期化し、所定のスロットに割り当てられた書き込みイタレーションカウンタを読み出しイタレーションカウンタと比較し、読み出しポインタが書き込みアドレスと一致するまで、読み出しポインタをインクリメントして、循環バッファ内の連続するスロットにアラインする。
【００４２】
一実施例において、書き込みモニターが書き込み更新を検出した後に、同期モジュールが読み出しポインタをインクリメントしてバッファ内の連続するスロットにアラインする。
【００４３】
一実施例において、バッファアクセスシステムは、書き込み更新が検出されたとき、書き込みアドレスと一致するスロットからデータを抽出し、データの抽出の後に書き込みイタレーションカウンタを確認して、抽出プロセスを有効化し、書き込み更新の検出の後に書き込みイタレーションカウンタが変わっていた場合に読み出しエラーを生成する、抽出モジュールを具備する。
【００４４】
一実施例において、バッファアクセスシステムは、書き込み更新に続き書き込みアドレスに対応する現在のスロットから長さリファレンスを読み出し、長さリファレンスが、現在のスロットのデータとそれに続くスロットのデータが一つのフレームのデータからなることを示している場合に、それらのデータを結合する、フレーム復元モジュールを具備す

る。
【００４５】
この本発明の第九の態様によれば、データ送信方法は、ホストコンピュータシステムのプロセッサ内の外部送信ネットワークデータをバッファへ移す工程と、バッファに移された外部送信ネットワークデータをネットワークインターフェイス内の所定のメモリであって、ホストコンピュータシステムのローカルメモリ階層にマッピングされたメモリに書き込む工程を具備する。
【００４６】
この本発明の第十の態様によれば、データ送信方法は、
ネットワークインターフェイスからの所定のメモリをホストコンピュータシステムのローカルメモリ階層（ヒエラルキー）にマッピングする工程と、
ホストコンピュータシステムのプロセッサの書き込み－結合バッファ内の外部送信ネットワークデータをバッファへ移す工程と、
書き込み－結合バッファからの外部送信ネットワークデータを所定のメモリに書き込む工程と、
ネットワークインターフェイスからの前記データを外部ネットワークに送信する工程とを含む。
【００４７】
一実施例において、データ送信方法は、外部ネットワークへのデータフレームの送信を開始するために、フレームリファレンスをネットワークインターフェイス内の離間した制御レジスタに書き込む工程であって、フレームレファレンスは、所定のメモリ内の対応するフレームの位置を規定する工程を含む。
【００４８】
一実施例において、データ送信方法は、離間した制御レジスタを、ホストコンピュータシステムの基本アドレスレジスタ０にマッピングする工程を含む。
【００４９】
一実施例において、データ送信方法は、所定のネットワークインターフェイスメモリをホストコンピュータシステムの基本アドレスレジスタ２にマッピングし、マッピングされたメモリの書き込み－結合可能にする。
【００５０】
一実施例において、データ送信方法は、外部ネットワークへ伝送するために所定のメモリから個別のデータフレームを抽出する工程であって、データフレームは、離間した制御レジスタへ書き込まれたフレームリファレンスと所定のメモリ内のデータフレームに書き込まれたフレーム長さにより定義される工程を含む。
【００５１】
一実施例において、データ送信方法は、外部送信ネットワークデータを含む所定のメモリに制御データを書き込む工程であって、制御データが対応するデータフレームのサイズを規定するフレーム長を含む工程を含む。
【００５２】
一実施例において、データ送信方法は、外部送信ネットワークデータを含む所定のメモリに書き込まれる制御データにフィードバック記述子を組み込む工程であって、フィードバック記述子はネットワークインターフェイスからのコンファメーションメッセージのホストコンピュータシステムへの送信を促進する工程を含む。
【００５３】
一実施例において、データ送信方法は、所定のメモリからのフレーム識別子をホストコンピュータシステムメモリ内に配置されたフィードバックレジスタに書き込み、対応するデータフレームのネットワークインターフェイスから外部ネットワークへの送信を報告する工程を含む。
【００５４】
一実施例において、データ送信方法は、対応するデータフレームの抽出を含む統合され

たメモリアクセストランザクションにおいて、ホストコンピュータシステムフィードバックレジスタを規定するフレーム識別子と参照アドレスとを所定のメモリから抽出する工程を含む。

【００５５】

この本発明の第十一の態様によれば、データ送信システムは、ネットワークインターフェイスからのメモリをホストコンピュータシステム内のローカルメモリ階層にマッピングするメモリ管理モジュールと、外部送信ネットワークデータを直接メモリにマッピングする前に、前記ホストコンピュータシステムのプロセッサ内の前記外部ネットワークデータをバッファに移す書き込みコントローラと、を具備する。

【００５６】

この本発明の第十二の態様によれば、データ送信システムは、

ネットワークインターフェイスからの所定のメモリをホストコンピュータシステムのローカルメモリ階層（ヒエラルキー）にマッピングするメモリ管理モジュールと、

外部送信ネットワークデータをネットワークインターフェイスからの所定のメモリに書き込む前に、書き込み－結合バッファ内の外部送信ネットワークデータをバッファへ移す書き込みコントローラと、

を具備する。

【００５７】

一実施例において、データ送信システムは、外部ネットワークへのデータフレームの送信を開始するために、フレームリファレンスをネットワークインターフェイス内の離間した制御レジスタに書き込む制御モジュールを具備し、フレームレファレンスは、所定のメモリ内の対応するフレームの位置を規定する。

【００５８】

一実施例において、メモリ管理モジュールが、離間した制御レジスタを、ホストコンピュータシステムの基本アドレスレジスタ０にマッピングする。

【００５９】

一実施例において、メモリ管理モジュールが、所定のネットワークインターフェイスメモリをホストコンピュータシステムの基本アドレスレジスタ２にマッピングし、マッピングされたメモリの書き込み－結合を可能にする。

【００６０】

一実施例において、データ送信システムは、外部ネットワークへ伝送するために前記所定のメモリから個別のデータフレームを抽出する送信エンジンを具備し、前記データフレームは、前記離間した制御レジスタへ書き込まれたフレームリファレンスと前記所定のメモリ内のデータフレームに書き込まれたフレーム長さにより定義される。

【００６１】

一実施例において、データ送信システムは、書き込みコントローラが前記ネットワークデータを前記所定のメモリに書き込む前に、制御データを外部送信ネットワークデータと結合する制御モジュールを具備し、前記制御データが対応するデータフレームのサイズを規定する。

【００６２】

一実施例において、制御モジュールが、外部送信ネットワークデータと結合された制御データにフィードバック記述子を組み込み、フィードバック記述子はネットワークインターフェイスからのコンファメーションメッセージのホストコンピュータシステムへの送信を促進する。

【００６３】

一実施例において、データ送信システムは、所定のメモリからのフレーム識別子をホストコンピュータシステムメモリ内に配置されたフィードバックレジスタに書き込み、対応するデータフレームのネットワークインターフェイスから外部ネットワークへの送信を報告する通知エンジンを具備する。

【００６４】

一実施例において、通知エンジンが、対応するデータフレームの抽出を含む統合されたメモリアクセストランザクションにおいて、前記ホストコンピュータシステムフィードバックレジスタを規定するフレーム識別子と参照アドレスとを前記所定のメモリから抽出する。

【００６５】

この本発明の第十三の態様によれば、ネットワークインターフェイスは、

外部データネットワークと接続する複数のデータポートと、前記データポートとホストコンピュータシステムの内部システムバスとの間のデータのやり取りを管理する制御システムを具備し、

前記制御システムは、前記システムバスと個々のデータポートとの間でやり取りしたデータを複製し、複製されたデータを所定のロギングポートに転送するロギングモジュールを有する。

【００６６】

この本発明の第十三の態様によれば、ネットワークインターフェイスは、

外部データネットワークと接続してデータのやり取りを促進するネットワーク面であって、通信のためのデータチャネルを規定する複数のデータポートを有するネットワーク面と、

ホストコンピュータシステムの内部システムバスと接続して前記ネットワークインターフェイスとホストコンピュータシステムとの間のデータのやり取りを促進するホスト面と、

ネットワークインターフェイスのネットワーク面とホスト面との間のデータのやり取りを管理する制御システムであって、個々のデータチャネルからのデータを複製し、複製されたデータを所定のロギングポートに転送するロギングモジュールを有する制御システムと、を具備する。

【００６７】

一実施例において、ネットワークインターフェイスは、ロギングモジュールの構成を促進するロギングインターフェイスであって、ロギングモジュールにより複製されたデータチャネルを規定する複数の制御レジスタを有するロギングインターフェイスを具備する。

【００６８】

一実施例において、ネットワークインターフェイスは、各データチャネルからのデータを受信ストリームと送信ストリームに分離するロギングコントローラを具備し、ロギングインターフェイスが前記ロギングモジュールにより複製されたデータストリームを規定する複数の制御レジスタを有する。

【００６９】

一実施例において、ネットワークインターフェイスは、外部ネットワークへ送信するために、複数のデータポートからのデータを統合されたデータストリームに結合するロギングユニットを具備する。

【００７０】

この本発明の第十五の態様によれば、データロギング方法は、コンピュータシステムネットワークインターフェイス内のネットワークデータを自律的に複製する工程と、複製されたデータを外部ロギングシステムへ送信するためにネットワークインターフェイスの所定のロギングポートに転送する工程とを具備する。

【００７１】

この本発明の第十六の態様によれば、データロギング方法は、

複数のデータポートを有するネットワークインターフェイスを使ってホストコンピュータシステムと外部データネットワークとの間でデータのやり取りをする工程であって、ネットワークインターフェイスはホストコンピュータシステムの内部システムバスに接続されている工程と、

個々のデータポートからのデータを複製し、複製されたデータをネットワークインターフェイスに組み込まれた所定のロギングポートに送信する工程と、を含む。

【００７２】
一実施例において、データロギング方法は、ロギングするための個別のデータポートを選択する工程であって、データポートはネットワークインターフェイス内に配置された所定の制御レジスタを使って選択される工程を含む。
【００７３】
一実施例において、データロギング方法は、
データポートのそれぞれで受信されたデータを受信ストリームと送信ストリームに分離する工程と、
個々のデータストリームをロギングのために選択する工程であって、そのストリームはネットワークインターフェイス内に配置された所定の制御レジスタを使って選択される工程と、を含む。
【００７４】
一実施例において、データロギング方法は、複数のデータポートからのデータを結合して、結合されたデータをホストコンピュータシステムから外部ネットワークに送信する工程を含む。
【００７５】
この本発明の第十七の態様によれば、ネットワークインターフェイスは、複数のデータポートと、データポートとホストコンピュータシステムの内部システムバスとの間のデータのやり取りを管理する制御システムを具備し、制御システムは、ホストコンピュータシステムからの介入なしに、離間したデータポート間でデータを転送する転送モジュールを有する。
【００７６】
この本発明の第十八の態様によれば、ネットワークインターフェイスは、
外部データネットワークと接続してデータのやり取りを促進するネットワーク面であって、複数のデータポートを有するネットワーク面と、
ホストコンピュータシステムの内部システムバスと接続してネットワークインターフェイスとホストコンピュータシステムとの間のデータのやり取りを促進するホスト面と、
ネットワークインターフェイスのネットワーク面とホスト面との間のデータのやり取りを管理する制御システムであって、ネットワークインターフェイスの規定されたデータポートからデータを直接受信し、ホストコンピュータシステムからの介入なしに、受信したデータをネットワークインターフェイスの他のデータポートを使って外部データネットワークへ送信する転送モジュールを有する制御システムと、を具備する。
【００７７】
一実施例において、ネットワークインターフェイスは、転送モジュールの構成を促進する転送インターフェイスであって、二つのポート間でデータ転送をできなくする制御レジスタを有する転送インターフェイスを具備する。
【００７８】
一実施例において、ネットワークインターフェイスは、転送モジュールにより受信されたデータから送り先アドレスを読み出し、ホストコンピュータシステムに関連するアドレスと一致する送り先アドレスのデータの転送を防ぐフィルタリングエンジンを具備する。
【００７９】
この本発明の第十九の態様によれば、データ転送方法は、コンピュータシステムネットワークインターフェイスの所定のデータポートでネットワークデータを受信する工程と、ホストコンピュータシステムからの介入なしに、受信されたデータを前記ネットワークインターフェイスの他のポートを介して外部データネットワークへ転送する工程と、を具備する。
【００８０】
この本発明の第二十の態様によれば、データ転送方法は、
複数のデータポートを有するネットワークインターフェイスを使って外部データネットワークからデータを受信する工程であって、ネットワークインターフェイスはホストコン

ピュータシステムの内部システムバスと接続されている工程と、
ホストコンピュータシステムからの介入なしに、ネットワークインターフェイスの所定のポートを介して受信したデータをネットワークインターフェイスの他のポートに転送する工程と、を含む。
【００８１】
一実施例において、データ転送方法は、ネットワークインターフェイス内の制御レジスタに書き込むことにより二つのネットワークインターフェイスポート間でのデータ転送をできなくする工程を含む。
【００８２】
一実施例において、データ転送方法は、転送モジュールにより受信されたデータから送り先アドレスを読み出し、ホストコンピュータシステムに関連するアドレスを一致する送り先アドレスのデータの転送を防ぐ工程を含む。
【００８３】
本発明の特徴及び利点は、下記の実施の形態における図面を参照した例示的説明で明らかになる。
【図面の簡単な説明】
【００８４】
【図１】相互に連結された複数のコンピュータシステムを含むネットワークの概略図を示す。
【図２】複数の機能モジュールを含むネットワークインターフェイスの概略図を示す。
【図３】受信バッファのメモリの配置を表したブロック図である。
【図４】ネットワークインターフェイスとのデータ交換を調整する複数の機能モジュールを含むホストコンピュータシステムの概略図を示す。
【図５】ネットワークインターフェイスの中のメモリの配置を示すブロック図である。
【図６】コンピュータシステムからネットワークインターフェイスへのデータ転送の送信プロセスのフローチャートである。
【図７】ネットワークインターフェイスのロギング機能のブロック図である。
【図８】ネットワークインターフェイスのポート伝送機能のブロック図である。
【発明を実施するための形態】
【００８５】
低遅延ネットワークインターフェイスとそれを補足するデータ管理プロトコルについての実施例をここで説明する。データ管理プロトコルは、制御データをネットワークデータと統合することにより、ネットワークインターフェイスとホストコンピュータシステムとの間の制御メッセージの交換を減少させる。
【００８６】
通信遅延を減少させることにより、通常、ネットワークの集中的なインタラクションの処理時間が改善する。下記で説明する送信プロトコルと受信プロトコルの実施例では、ネットワークインターフェイスとホストコンピュータシステムとの間のやり取りを制限して処理時間を減少させる。全体的な遅延を数分の一秒でも減少させることは、コンピュータアプリケーション（例えばオンラインゲームや金融トレーディング）の高性能化に深く関係する。
【００８７】
従来のネットワークアプリケーションでは、ネットワークデータは通常、個別のデータフレームを使ってコンピュータシステム間で伝送される。データフレームは、送信側により定義され、送信処理で使われる制御データ（例えば送信先のアドレス）を含んでいる。本明細書で説明されるネットワークインターフェイスの実施例は、主にフレームに基づくネットワークの実施（データがフレームで通信される）に関連する。しかしながら、一般のプロトコルも他のネットワーク規格で実行されてもよい。
【００８８】
遅延（レイテンシー）とは、データがシステムを通過するのにかかる時間の長さである

。ネットワークインターフェイスの「受信遅延」とは、ホストコンピュータシステムが外部ネットワークからデータフレームを受信してから、そのホストコンピュータシステムで実行されているソフトウエアアプリケーションでそのフレームが利用可能になるまでの時間を示す。ネットワークインターフェイスの「送信遅延」とは、フレームがホストコンピュータシステムで実行されているソフトウエアアプリケーションで利用可能となってから、そのフレームが外部ネットワークに伝送されるまでの時間を示す。受信遅延と送信遅延は主にネットワークインターフェイスに起因する。

【００８９】

データ管理プロトコル及びインターフェイスに関する実施例は、本明細書における個別の機能ユニット（典型的には、モジュール又はエンジン）で実現される。これらの機能ユニットは、それらが具現化する様々なインターフェイス要素（ハードウエア及びソフトウエア）の動作及び相互作用を明示する。それらは、厳格な動作区分又は特定のハードウエア／ソフトウエアの実行を表示するのではない。複数の機能ユニットに起因する動作は、一つのハードウエア（例えばマイクロプロセッサ又はフィールドプログラマブルゲートウエイ）、ソフトウエアアプリケーション（例えばドライバーソフトウエア又は組み込み（エンベデッド）ソフトウエア）、複数のハードウエア、又はハードウエアとソフトウエアの混合により実施されてもよい。

【００９０】

（外部ネットワークからのデータ受信）

従来のネットワークインターフェイスプロトコルは、ホストコンピュータシステムで実行されるドライバーソフトウエアにより調整される。ドライバーはホストメモリを個別のデータフレームに割り当て、ポインタ（受信ディスクリプタ（受信記述子））の配列をホストメモリの中で記録として維持する。ネットワークインターフェイスは、新しいフレームが受信されたときに受信ディスクリプタの配列にアクセスし、そのフレームを対応するポインタで定義されたメモリアドレスにコピーする。コピーが完了してホストシステムに新しいフレームが移管されたことを警告すると、インターフェイスはシステムインタラプト（割り込み）を発生させる。フレームが処理されると、ドライバーソフトウエアはポインタを次の使用から解放する。

【００９１】

ここで説明するネットワークインターフェイス及び受信プロトコルは、ホストコンピュータシステムとのいくつかのインタラクションを省略することにより、ネットワークインターフェイスに起因する受信遅延を減少させる。ここで開示されるネットワークインターフェイスは、ホストコンピュータシステムのメモリに配置された一つの受信バッファ（または少数の受信バッファ）を使って、ホストコンピュータシステムと受信ネットワークデータをやり取りする。各受信バッファは、外部ネットワークから受信されたデータフレームを複数記憶できる。

【００９２】

ネットワークインターフェイスは、ホストコンピュータシステムから介入されることなしに、受信バッファの中のデータの配置を自立的に規定する。典型的なインターフェイスは、循環した先入先出（ファーストインファーストアウト、ＦＩＦＯ）配列プロトコルを使って、受信バッファの中に受信データフレームを連続的に配置する。ホストコンピュータシステムは、ネットワークインターフェイスにより受信された個別のフレームに、書き込みアドレス又は専用バッファを割り当てない。

【００９３】

ネットワークインターフェイスは、ホストコンピュータシステムからのフィードバック（例えば、フレームが抽出されたことを示す読み出し確認）がない開ループ書き込みプロトコルを使って、データフレームを受信バッファに繰り返し書き込む。このことにより、ホストコンピュータシステムで実行されている複数のソフトウエアアプリケーションによる、受信バッファのデータへの組織的でないアクセスが促進される。ネットワークインターフェイスは、書き込みイタレーション（繰り返し）が間、受信バッファに新しいフレー

ムを連続的に書き込む。ネットワークインターフェイスは、通常、受信フレームが受信バッファに書き込まれる際に、その受信フレームに標準化された制御データ（現状の書き込みイタレーションを含む）を付加する。制御データは、ホストコンピュータシステムが受信バッファからフレームを抽出するのに用いられる。

【００９４】

ホストコンピュータシステムは受信プロトコルを実行して、受信バッファからのデータフレームを、ホストコンピュータシステムで実行されているソフトウエアアプリケーションがアクセスできるようにする。受信プロトコルは、ネットワークインターフェイスにより作成された制御データを使って、受信バッファに保存されているデータフレームの抽出と再構成を容易にする。

【００９５】

図１にコンピュータネットワーク１０を示す。ネットワーク１０は、互いに接続されている複数のコンピュータシステムを含む。それらのコンピュータシステムは共通のデータネットワーク１１（例えば、インターネット又はローカルエリアネットワーク）により接続されている。

【００９６】

図に示されたデータネットワーク１１は、ホストコンピュータシステム１５と複数の遠隔コンピュータシステム１２との間の通信を容易にする。ホストコンピュータシステム１５には、データネットワーク１１によるデータ交換を管理するネットワークインターフェイス２０が組み込まれている。

【００９７】

図２はネットワークインターフェイス１２の機能を示す図である。インターフェイス２０は、外部データネットワーク１１からデータを受信するネットワーク面２１と、受信したデータを、内部システムバスを介してホストコンピュータシステム１５内のメモリに転送するシステム面２２とを含む。

【００９８】

図２に示すネットワークインターフェイス２０のネットワーク面２１は、複数のデータポート１９を含む。各データポート１９は、ネットワークインターフェイス制御システム２５と外部データネットワークとの間のネットワークデータのやり取りのためのデータチャネルを定義する。データポート１９は物理的なコネクタ（例えば図２に示す小型フォームファクタモジュール（Ｓｍａｌｌ　Ｆｏｒｍ‐Ｆａｃｔｏｒ　Ｍｏｄｕｌｅｓ））で一体化されるか、又は外部ネットワークとの無線データチャネル接続であってもよい。

【００９９】

ネットワークインターフェイスのシステム面２２は、一般的に、ネットワークインターフェイスと内部システムバスとの機能的接合部となる（例えば、統合されたアプリケーションの共有されたマイクロコントローラ内の機能的部分）。インターフェイスのシステム面２２により定義されたこの機能的接合部は、物理的な接合点（例えば、バスコネクタ又はＰＣＢ上の有線インターフェイス）と同じであってもいてもよい。図２に示すネットワークインターフェイス２０のシステム面２２はホストコンピュータシステムのマザーボード２４のＰＣＩバススロット２３と同じである。インターフェイス制御システム２５はネットワークインターフェイス２０のネットワーク面２１とシステム面２２との間のデータ転送を管理する。

【０１００】

インターフェイス２０はネットワーク１１からフレーム単位でデータを受信する。典型的なイーサネット（登録商標）フレームの大きさは、６３バイトから１５１８バイトの間である。フレームの最大サイズより大きいデータ群は、データネットワーク１１に伝送される前に複数のフレームに分割される。ネットワークインターフェイスはネットワークから受信したフレームをホストコンピュータシステムの共有メモリに転送する。この手順は、従来の受信プロトコルにおいては、ホストコンピュータシステムにより管理されている。

【０１０１】
図２に示すネットワークインターフェイス２０は、ホストコンピュータシステムの介入なしに、受信したフレームをホストメモリの固定したバッファに転送する。これにより、時間的に集中するシステムインタラクションのいくつか（ホストコンピュータとのポインタ配置確認メッセージ及びシステム読み出し確認メッセージの交換を含む）を省略することで、受信処理のオーバーヘッドを減少させる。
【０１０２】
インターネット制御システム２５はインターフェイス２０のネットワーク面２１からホストコンピュータシステム１５内のシステムバス（例えばＰＣＩバス又はＰＣＩエクスプレスバス）へのフレーム転送を管理する。システムバスはフレームを、ホストコンピュータシステムプロセッサで実行されているソフトウエアアプリケーションがアクセス可能な受信バッファに送信する。
【０１０３】
ネットワークインターフェイスドライバーはホストコンピュータ１５内のメモリをネットワークインターフェイス２０に割り当てる。割り当てられたメモリは、ネットワークインターフェイス２０からのフレーム転送のためのバッファ（受信バッファ）に統合される。割り当てられたメモリアドレスは、一般的に、複数のデータフレームの保存に十分な容量の連続的なメモリの範囲を定義する（受信バッファのサイズは一般的に、ネットワークからのデータフレームの予想されるサイズより少なくとも一桁大きい）。ホストコンピュータシステム１５は、データ抽出を最適化するために、システムプロセッサにより効率的にアクセスできるメモリに受信バッファを割り当てる。
【０１０４】
ホストコンピュータ１５は、複数のデータフレームを受信できる容量を有する受信バッファのいくつかにメモリを割り当ててもよい。しかしながら、ホストコンピュータシステム１５により割り当てられる受信バッファの総数は、システムが処理することが予想されるデータフレームの数に比較して少ない。独立した受信バッファは一般的に、多数ポートネットワークインターフェイスの個々のポートに割り当てられる。単数ポートネットワークインターフェイスは、プログラム可能なフィルタを使って、フレームを複数の受信バッファに配分してもよい。個々の受信バッファは、ホストメモリの中で、連続（隣接）していなくてもよい（しかし一般的には、各バッファに割り当てられたメモリは連続（隣接）している）。
【０１０５】
ネットワークインターフェイス受信バッファは一般的に、システム起動の間に、専用のハードウエアドライバーにより割り当てられる（例えば、システムスタートアップ又は起動コマンドの後に）。図３に、受信バッファ３０内に割り当てられたデータの例示的な概略図を示す。
【０１０６】
ネットワークインターフェイス２０は、各受信バッファに割り当てられたメモリを複数の連続したスロット３１に分割する。統合されたバッファ管理モジュール２７は、割り当てられたメモリの分割を調整する。バッファ管理モジュール２７は、各受信バッファの中に、個別にアクセスできる複数の連続したスロット３１を生成する。バッファ管理モジュール２７により生成された複数のスロット３１は、同じ保存容量を有し、同じメモリアドレスオフセットを有する、均一のものである。スロット３１の保存容量は、典型的には、３２バイトと５１２バイトの間である。
【０１０７】
これらスロット３１によって、ネットワークインターフェイス２０は、連続した受信バッファ３０への書き込み操作を、ホストコンピュータシステムからの介入なしに、効率よく調整できる。例示されたネットワークインターフェイス２０は、割り当てられたメモリを循環バッファとして使う。つまり、特定のメモリアドレスへの書き込みシーケンスを再開する前に、特定のスロット３１のそれぞれに連続的に書き込む（典型的には、第一スロ

ットになる）。インターフェイス制御システム２５は、ホストコンピュータシステム（ホストコンピュータシステムで実行されているインターフェイスドライバーを含む）からの介入なしに、受信バッファ３０内でのデータフレームの割り当てを自律的に規定する。

【０１０８】

例示されたインターフェイス制御システム２５には、開ループ書き込みプロトコルを使って受信データフレームを受信バッファ３０に書き込む、書き込みコントローラ２６が組み込まれている。書き込みコントローラ２６は、連続した書き込みループの終了に続いて、受信バッファ３０内のスロット３１を繰り返し上書きする。受信バッファ３０の中のデータは、ホストコンピュータシステムからの読み出し確認（そのデータが処理済みであるとのフィードバック）なしに、上書きされる。ホストシステムからのフィードバックを省略することで、ネットワークインターフェイス２０は、中断なしにデータフレームを連続的にメモリに書き込みでき、データ転送の効率を高めることができる。

【０１０９】

ネットワークインターフェイス２０は、受信バッファ３０内の各スロット３１に書き込まれたネットワークデータ３２に、制御データ３３を結合させる。制御データ３３は、ホストコンピュータシステムにおいて受信バッファ３０からのデータ抽出を管理するのに使われる。例示されている制御システム２５には、ネットワークインターフェイス２０のための制御データ３３を作成するフレーム管理モジュール２８が組み込まれている。フレーム管理モジュール２８により作成された制御データ３３には以下のデータが含まれている。

タイムスタンプ（３４）	：３２ビット
フレーム状態（３５）	：８ビット
長さ（３６）	：８ビット
未使用（３７）	：８ビット
書き込みイタレーションカウンタ（３８）	：８ビット

【０１１０】

フレーム管理モジュール２８は、バッファへの書き込み操作の前に、制御データ３３とネットワークデータ３２とをひとまとめにする。このことにより、書き込みコントローラ２６は、ネットワークデータと同じバストランザクションで制御データ３３を受信バッファ３０に転送できる。ひとまとめになったデータパッケージは、ネットワークインターフェイス２０から受信バッファ３０へ、分離したスロットで転送される。

【０１１１】

例示したネットワークインターフェイスは、受信バッファ３０の各スロット３１に８バイト（６４ビット）の制御データを作成する。制御データ３３の１バイト（８ビット）は拡張のために割り当てられている（「未使用」データフィールド３７）。

【０１１２】

制御データ３３は、対応するネットワークデータを含む統合されたメモリトランザクションにおいて、受信バッファ３０の個々のスロット３１に書き込まれる。このことは、スロット３１の有効容量（ネットワークデータが利用できるメモリ）を減少させる。例えば、１２８バイトのスロットは、１２０バイトのネットワークデータ３２と８バイトの制御データ３３を保存できる。スロット３１内の、制御データ３３のネットワークデータ３２に対する比率は、スロット割り当てがより小さな場合に増加する。

【０１１３】

制御データ３３をネットワークデータ３２と同じスロット内に割り当てることで、制御データが別のレジスタに書き込まれている場合に一般的に発生する補助的な読み込み／書き込み操作を避けることができる。書き込みコントローラ２６は、典型的には、ネットワークデータと同じバストランザクションで、制御データ３３を受信スロットに書き込む。このことにより、フレームをホストコンピュータシステムに転送するのに必要な書き込み操作が減少する。例示した書き込みコントローラ２６は、制御データ３３がバッファ内の該当するスロットの最後に書き込まれるように、ネットワークデータ３２に制御データ３

３を付加する。このことにより、ホストコンピュータシステムは制御データ３３をチェックすることで、バッファ書き込み操作を確認できる。

【０１１４】

フレーム管理モジュール２８は、制御データ３３の４バイト（３２ビット）をシステムタイムスタンプ３４に割り当てている。タイムスタンプフィールド３４は、ネットワークインターフェイス２０におけるフレーム受信時間を記録する（典型的には、ネットワーク分析と診断に用いられる）。制御システム２５は、内部カウンタからタイムスタンプフィールド３４を引き出す。

【０１１５】

例示したフレーム管理モジュール２８は、フレームの第一バイトが外部ネットワークから受信されたときに、内部カウンタの値をタイムスタンプフィールド３４にコピーする。フレームがバッファ内で複数のスロット３１に分割された場合（つまり、データフレームが受信バッファ３０のスロットサイズより大きい場合）、同じタイムスタンプ３４が各フレームフラグメント（断片）に用いられる。

【０１１６】

フレーム状態フィールド３５は、受信フレームの一般的な状態の情報が含まれている。フレーム管理モジュール２８は、検知されたフレーム受信エラーをフレーム状態フィールド３５に記録する。例示されたフ制御システム２５により作成されるフレーム状態コードは、以下のとおりである。

０－正常受信
１－フレーム内遠隔送信者送信中止
２－フレームの完全性チェック失敗
３－フレーム受信中に内部メモリオーバーフロー

【０１１７】

バッファスロット３１の保存容量を超える受信フレームは、フレームフラグメント（断片）に分割され、複数のスロット３１に分配される。フレーム管理モジュール２８は、バッファの中で大きなデータフレームの分割と分配を調整する。フラグメントは、典型的には、受信バッファ３０の中の連続したスロット３１に分配される。

【０１１８】

フレーム管理モジュール２８は、制御データの長さフィールド３６を用いて、バッファ内の個々のフレームの分配を記録する。長さフィールド３６は、バッファ内の連続する複数のスロット３１間の関係を規定する。長さフィールド３６がゼロ以外の値の場合は、そのスロット内で、フレームが終わっていることを示す（そのスロットに、スロット分割と一致するフレームの端が含まれていることを示す）。長さフィールド３６の値がゼロの場合は、隣接するスロット３１に同じデータフレームのフラグメントが含まれていることを示す。ホストコンピュータシステムは長さフィールド３６を、断片化されたデータフレームを復元するのに用いる。

【０１１９】

一つのスロット３１内に含まれているフレームの長さフィールド３６はそのフレームのサイズ（典型的には、バイト数で）を示す。フレームの最初及び中間のフラグメントは長さフィールド３６の値にゼロを示す。このことは、同じフレームからのさらなるフラグメントが次の（連続した）スロットに保存されていることを示す。フレームの最後のフラグメントには、フラグメントのサイズ（スロット内でフラグメントが占める記憶スペース）に等しい長さが示されている。ゼロでない値の長さが入力されていることは、フレームのすべてのフラグメントがバッファから抽出され、次のスロットには異なるフレームが含まれていることを、ホストコンピュータシステムに示す。

【０１２０】

バッファ管理モジュール２７は、ネットワークインターフェイス２０における特定のレジスタの中の受信バッファ３０のジェネレーションカウンタを保持する。ジェネレーションカウンタは、循環する受信バッファ３０の現在の書き込みイタレーションを記録する。

バッファ管理モジュール２７は、書き込みループが完了するたびに（つまり、書き込みコントローラ２６が循環バッファ内の最後のスロットにデータを書き込むたびに）、ジェネレーションカウンタを単調にインクリメントする。

【０１２１】

フレーム管理モジュール２８は、書き込み操作の間に、バッファの現在のジェネレーションカウンタの値を制御データの書き込みイタレーションカウンタフィールド３８にコピーする。ホストコンピュータシステムで実行されているソフトウエアは書き込みイタレーションカウンタ３８を用いて受信バッファ３０からの読み出し操作の同期をとる。

【０１２２】

（バッファアクセスプロトコル）

ホストコンピュータシステムは、ネットワークインターフェイスからの書き込み更新のために受信バッファ３０をモニターし、新しいフレームが利用可能になったら、そのフレームを抽出する。このプロセスは、遅延しやすいアプリケーションにおいてネットワークインターフェイス書き込み操作に依存しない。ネットワークインターフェイスからの介入なしに受信バッファからネットワークデータを抽出することは、いくつかの時間を要するシステムインタラクション（従来のネットワークインターフェイスプロトコルで内部データの交換の調整に使われていた）を省略することになる。このことにより、ネットワークインターフェイスはより少ないオーバーヘッドで、データフレームをホストコンピュータシステムに転送できる。

【０１２３】

ホストコンピュータシステムには受信バッファ３０からのデータ抽出を調整するバッファアクセスシステムが組み込まれている。図４はバッファアクセスシステム４０を機能的に示す図である。例示されたバッファアクセスシステム４０は、複数の機能モジュールを含んでいる。それぞれのモジュールは、特定のドライバーソフトウエアにより、インプリメントされてもよいし、ホストコンピュータシステム１５で実行される独立のソフトウエアアプリケーションであってもよい。

【０１２４】

図４に例示されているバッファアクセスシステム４０には、データ抽出プロセスが始まったときに受信バッファ３０の状態を評価する、参照モジュール４１が組み込まれている。参照モジュール４１は、バッファの状態を判定するのに、書き込みイタレーションカウンタフィールド３８を用いる。各スロット３１の制御データ３３の中に配置された書き込みイタレーションカウンタ３８は、データが最後にスロットに書き込まれたときに書き込みイタレーションを定義する。参照モジュール４１は、循環する受信バッファ３０の中の複数のデータスロット３１にアクセスし、制御データ３３に含まれる書き込みイタレーションカウンタ３８を評価する。バッファアクセスシステム４０は、連続するスロット３１からの書き込みイタレーションカウンタ３８を使って、現在の書き込みイタレーションと以前の書き込みイタレーションとの境界を特定する。

【０１２５】

同期モジュール４２は、参照モジュール４１が定めた書き込みイタレーションカウンタ３８を使って、循環バッファの現在の書き込みアドレスを定義する。書き込みアドレスは、ネットワークインターフェイス２０によりデータが上書きされる次のスロット３１を受信バッファ３０の中で規定する。同期モジュール４２により実行される、バッファ書き込みアドレスを特定するプロセスは、ネットワークインターフェイスにより使用される書き込みイタレーションカウンタ３８のフォーマットに依存する。バイナリ（２進数）及び整数での書き込みイタレーションカウンタ３８のフォーマットでアドレス特定するプロセスは、本明細書で別途説明される。

【０１２６】

同期モジュール４２は書き込みアドレスを書き込みモニター４３に伝達する。書き込みモニター４３は、書き込みアドレスを繰り返しポーリング（順次問い合わせる）して、ネットワークインターフェイスからの書き込み更新を確認する。書き込みアドレスで定義さ

れたバッファスロット３１内の制御データ３３が変わったときに、書き込み更新が検出される。ネットワークインターフェイスは、書き込み更新の際に、ネットワークデータ３２と制御データ３３を共に上書きする。書き込みモニター４３は、モニターしているスロットの書き込みイタレーションカウンタ３８が変わったときに、スロット３１が書き込み更新を受けたかどうかを判定する。

【０１２７】

抽出モジュール４４は、書き込みモニター４３が、書き込み更新を検出したとき、書き込みアドレスに該当するスロット３１からデータを抽出する。抽出モジュール４４は、データをスロット３１から抽出した後に、いくつかのデータ有効確認を実行する。有効確認は、受信確認（制御データ３３のフレーム状態フィールド３５から導かれる）、データ完全性評価（書き込み更新の間、各スロット３１の最後に付加される制御データ３３の完全性から導かれる）、及び抽出評価を含む。

【０１２８】

抽出モジュール４４は、データ抽出に続いて、バッファスロット３１の書き込みイタレーションカウンタ３８を確認して、抽出プロセスを有効にする。書き込みモニター４３により特定された書き込み更新の検出の後に、書き込みイタレーションカウンタ３８が変わった場合、抽出モジュール４４は、読み出しエラーを生成する。読み出しエラーは、データが読み出される前に、スロット３１がネットワークインターフェイスにより上書きされているかもしれないことを示す（このことにより、抽出されたデータは信頼できないものとなる）。受信したソフトウエアアプリケーションは、一般的に、データを失ったフレームについて付加リクエストを発行することで、読み出しエラーに対処する。

【０１２９】

フラグメントデータフレームは、受信バッファ３０から抽出された後に、フレーム復元モジュール４５により、コンパイルされる。フレーム復元モジュール４５は、書き込み更新の後に抽出された制御データ３３から長さリファレンスフィールド３６を読み出し、該当するネットワークデータの状態を判定する。データ状態は、全データフレームからフレームフラグメントを識別し、同じデータフレームの複数のフラグメント間の関係を特定する。フレーム復元モジュール４５は、長さリファレンスフィールド３６が、各スロット３１が一つのフレームからのデータを含むことを示す場合に、連続する複数のスロット３１かあのネットワークデータ３２を結合させる。ネットワークインターフェイスは、フレーム復元モジュールが連続したフラグメントをバッファから直接コンパイルできるように、受信バッファ３０に書き込まれるフレームフラグメントの順序を規定する。

【０１３０】

ネットワークインターフェイスは、連続する書込みイタレーションを記録するために、バイナリ書き込みイタレーションカウンタと整数書き込みイタレーションカウンタを用いることができる。バイナリ書き込みイタレーションカウンタは最少のメモリ消費で連続する複数の書き込みイタレーションを区別する。整数書き込みイタレーションカウンタ（例えばフレーム管理モジュール２８により割り当てられた８ビットの書き込みイタレーションカウンタ）は、書き込みイタレーションカウンタの絶対値を維持できる。書き込みイタレーションカウンタ３８に用いられるデータタイプは、同期モジュールにより実行される評価プロセスに影響する。

【０１３１】

（書き込みアドレス：バイナリ書き込みイタレーションカウンタ）

バイナリ書き込みイタレーションカウンタの値は、連続するバッファ書き込みループにより書き換えられる。このことにより、同期モジュール４２は連続する複数の書き込みイタレーションを識別できる。しかし、抽出モジュールにより実施されるデータの有効性確認を損なうことにもなり得る。

【０１３２】

同期モジュール４２は、バッファ内の連続するスロット３１のバイナリ書き込みイタレーションカウンタを繰り返し比較することにより、バッファへの書き込みアドレスを判定

できる。バッファの隣接するスロットに異なる書き込みイタレーションカウンタを特定することにより、バッファ書き込みアドレスの位置が定まる。

【０１３３】

（書き込みアドレス：整数書き込みイタレーションカウンタ）

整数書き込みイタレーションカウンタは、書き込みイタレーションの蓄積を記録する。これにより、より多くのメモリを消費することになるが、抽出モジュールにより実施されるデータ認証確認の精度が改善される。図２に例示されるネットワークインターフェイス２０は、書込み操作の間に個々のスロット３１の書き込みイタレーションカウンタフィールドにコピーされるジェネレーションカウンタを保持する。

【０１３４】

バッファアクセスシステム４０は、補助的な読み出しイタレーションカウンタを作成することにより、整数書き込みイタレーションカウンタを適応させる。読み出しイタレーションカウンタは、受信バッファ３０の現在の書き込みアドレスを特定するために、連続したバッファスロット３１の制御データ３３と比較される。

【０１３５】

図４に例示されている参照モジュール４１は、同期モジュール４２のための読み出しイタレーションカウンタを作成する。読み出しイタレーションカウンタはバッファ管理モジュール２７で保持されているバッファジェネレーションカウンタを複製する。参照モジュール４１は、受信バッファ３０内の特定のスロット３１に書き込まれた制御データ３３から、読み出しイタレーションカウンタを引き出す。このことにより、ネットワークインターフェイスから直接ジェネレーションカウンタを検索することを避けられる。

【０１３６】

例示されている参照モジュール４１は、受信バッファ３０の最後のスロットからの書き込みイタレーションカウンタにより、読み出しイタレーションカウンタを初期化する。読み出しイタレーションカウンタは典型的には、バッファの最後のスロットから引き出される。ジェネレーションカウンタが各書き込みサイクルの完了時にのみインクリメントするので（最後のバッファスロットへの書き込み操作の後）、最後のスロットのそれぞれの書き込みイタレーションカウンタは、各書き込みサイクルの間、変わらずに残る。

【０１３７】

参照モジュール４１は、読み出しイタレーションカウンタを同期モジュール４２に転送する。同期モジュール４２は、読み出しイタレーションカウンタを、循環バッファ内の連続するスロット３１に割り当てられた複数の書き込みイタレーションカウンタと比較する。バッファの書き込みアドレスは、読み出しイタレーションカウンタ（バッファの最後から引き出されたもの）と一致する書き込みイタレーションカウンタを有する最初のスロット３１（バッファの最初に一番近いスロット３１）に一致する。

【０１３８】

比較プロセスは、受信バッファ３０内の連続する複数のスロット３１を循環する読み出しポインタを使って調整される。バッファアクセスシステム４０は、典型的に、受信バッファ３０内の最初のスロットから読み出しポインタを始める。読み出しポインタで特定されたスロットの書き込みイタレーションカウンタが、読み出しイタレーションカウンタと一致するまで、同期モジュール４２は読み出しポインタを規定されたオフセット（規定されたバッファスロットサイズに比例する）でインクリメントする。インクリメントのオフセットは、バッファ管理モジュール２７で割り当てられたバッファスロット３１のサイズに比例する。同期モジュール４２は、読み出しポインタをインクリメントして、検出された書き込み更新に続く連続するスロット３１にアライン（位置合わせ）する。

【０１３９】

（遅延集中データの受信警告プロトコル）

図２に例示されているネットワークインターフェイス２０は、遅延が集中してネットワークデータの受信に対して、警告プロトコルを実行できる。受信警告プロトコルは、ホストコンピュータシステムのモニタリングオーバーヘッドを縮小する。

【０１４０】
警告プロトコルは、ホストコンピュータシステム上で実行されているソフトウエアアプリケーションとインターフェイスするリザベーションモジュール（図示せず）により開始される。ソフトウエアアプリケーションは、遅延集中するネットワーク上のやり取りを特定し、それをリザベーションモジュールに伝達する。リザベーションモジュールは、これらのデータやり取りに帰するネットワークデータについて、一時的に書き込みモニター４３の記録機能を停止し、ネットワークインターフェイス警告プロトコルを開始する。これで、ホストコンピュータシステムプロセッサの負荷を緩和できる。
【０１４１】
リザベーションモジュールは、遅延が重大なことになるソフトウエアを全体的に優先することができる、個々のソフトウエアアプリケーションに対する設定変更可能な遅延指定を保持していてもよい。ソフトウエアインターフェイス（典型的には、ネットワークインターフェイスドライバーにより供給される）は、個々のソフトウエアアプリケーションの遅延指定の設定を促進する。
【０１４２】
リザベーションモジュールは、書き込みモニター４３を停止すると同時に、ネットワークインターフェイス制御システム２５に組み込まれている報告エンジンを初期化する。報告エンジンは、データが受信バッファに書き込まれているときに、システムの割込みを発生させる。この割込みは、リザベーションモジュールに、新しいデータが受信バッファの中で利用可能になっていることを通知する。
【０１４３】
受信警告も可能なネットワークインターフェイスには、リザベーションモジュールが警告プロトコルを初期化するのに使う、特定のしきい値レジスタが組み込まれている。リザベーションモジュールは、しきい値レジスタにスロットアドレスを書き込んで報告エンジンを起動する。そのスロットアドレスは、報告エンジンがモニターするスロットを受信バッファ内で特定する。スロットアドレスにより特定されたスロットが書き込み更新を受信したときに、報告エンジンは書き込みコントローラ２６とインターフェイスし、通知中断を発生させる。
【０１４４】
しきい値レジスタに書き込まれたスロットアドレスは、典型的には、受信バッファ内でのスロットオフセットを示す整数表示である。報告エンジンは、しきい値レジスタの中のスロットアドレスと、書き込みコントローラ２６で保持されている書き込みアドレスとを比較し、書き込みアドレスがスロットアドレスと等しいか、スロットアドレスより大きい場合に、システム割込みを発生させる。
【０１４５】
しきい値レジスタにスロット表示の絶対値を書き込むことは、同時に発生する書き込みコントローラ２６による書き込み操作を補う。同時に発生する書き込み操作は、リザベーションモジュールが、現状の書き込みアドレスの書き込み更新と同時に（又は部分的には後で）、しきい値レジスタへの書き込みを行う場合に発生する。報告エンジンは、書き込みコントローラのシーケンス番号がしきい値レジスタに書き込まれたスロットアドレスを超えることを確認することで、同時の書き込み更新を検出する。このことは、報告エンジンがバイナリ起動である場合には、不可能である。
【０１４６】
（外部ネットワークへのデータ送信）
ほとんどのネットワークインターフェイスは、バスマスタリング（ｂｕｓ　ｍａｓｔｅｒｉｎｇ）プロトコル又は、プログラムされた入力／出力（ｐｒｏｇｒａｍｅｄ　Ｉ／Ｏ）プロトコルを用いて、ホストコンピュータシステムの内部データ転送の調整を行う。これらのプロトコルは共に、標準化された内部システムバスを用いて実行される。
【０１４７】
早期のネットワークインターフェイスは、ホストコンピュータシステムとの通信するた

めに、プログラムされた入力／出力プロトコル（プログラムされたＩ／Ｏ）を実行した。プログラムされたＩ／Ｏプロトコルは、内部システムバス（例えばパラレルＡＴＡバス）を通じて、低い帯域幅のホスト制御の通信を促進する。外部送信するネットワークデータは、ホストコンピュータシステム内のＩ／Ｏアドレスに書き込むことにより、小さなパッケージ（通常、１６ビット又は３２ビット）でネットワークインターフェイスにプッシュされる。ホストコンピュータシステムのプロセッサはＩ／Ｏ書き込みを高周波で行う。このことは、過度のシステムバス帯域幅の消費や不均衡なプロセッサ負荷等の、リソース消費やホストコンピュータシステムにおけるオーバーヘッド発生を引き起こし得る。これらの制約は、バスマスタリングプロトコルにより、大部分対処される。

【０１４８】

ほとんどの従来のネットワークインターフェイスは、外部送信されるネットワークデータ（ホストコンピュータシステムから外部ネットワークへ送信されるデータ）の調整にバスマスタリングプロトコルを使用する。バスマスタリングにより、ネットワークインターフェイスは、バストランザクションを管理することができ、ホストコンピュータシステム内の外部送信ネットワークデータを、ホストコンピュータシステムプロセッサにとって最少のオーバーヘッドで移動できる。

【０１４９】

ホストコンピュータシステムで実行されているソフトウエアは、ネットワークインターフェイスに新しい外部送信ネットワークデータを通知することにより、転送プロセスを開始する。通知は、内部システムバスを用いて、プロセッサからネットワークインターフェイスに送られる。それからネットワークインターフェイスは、自律的に、ホストメモリから対応するデータを検索する。それには、ホストメモリへの読み出しリクエストと外部送信ネットワークデータの受信が含まれる（両方のやり取りには、システムバスが使われる）。

【０１５０】

ここで説明される伝送プロトコルの実施例では、ホストコンピュータシステムのプロセッサが外部送信ネットワークデータを、ネットワークインターフェイスのメモリ内の伝送バッファに直接書き込むことができる。これは、伝送バッファをホストコンピュータシステムのメモリ階層（ヒエラルキー）にマッピングすることにより、促進される。伝送バッファへの書き込みは、プログラムされたＩ／Ｏプロトコルに関連したいくつかの非効率を避けるために、バーストで放出される。これは、ネットワークインターフェイス内のアドレス指定可能なメモリ（伝送バッファ）に外部送信ネットワークデータを直接放出する前に、外部送信ネットワークデータを書き込み－結合バッファ（典型的にはホストコンピュータシステムのプロセッサ内にある）に集めることにより促進される。

【０１５１】

開示される伝送プロトコルは、バスマスタリングにおいて内部データ転送の調整に使われる、ホストプロセッサとネットワークインターフェイスとの間のいくつかのデータ管理のためのやり取りを省力する。省略されるやり取りは以下のとおりである。

・ネットワークインターフェイスからホストメモリへの、新しいデータを検索するための読み出しリクエスト。

・ホストコンピュータシステムメモリからネットワークインターフェイスへの新しいデータの転送。

【０１５２】

省略されるやり取りは、データがバスマスタリングプロトコルを使ってネットワークインターフェイスへ転送される前に実施される必要があるので、この省略により、開示されるネットワークインターフェイスの遅延動作が改善される。

【０１５３】

図４にデータ送信システム５０の機能的表示を示す。例示されたデータ伝送システム５０は、所定のネットワークインターフェイスメモリ内の伝送バッファ２９を管理するメモリ管理モジュール５１を含む。伝送バッファは典型的には、４キロバイトから２メガバイ

トの間の保存容量を有している（より大きな保存容量を使用してもよい）。図２に記載されているネットワークインターフェイス２０は、１６キロバイト、３２キロバイト、又は６４キロバイトの伝送バッファ２９を有していてもよい。

【０１５４】

メモリ管理モジュール５１は、所定のネットワークインターフェイスメモリを、ホストコンピュータシステム１５内のローカルメモリ階層６１にマッピングする。このことにより、マッピングされたメモリ（伝送バッファ２９に対応する）がホストコンピュータシステムのプロセッサから、直接アドレス指定可能になる。メモリ管理モジュール５１は、マッピングされたメモリの書き込み－結合も可能にする（これは、ＰＣＩバスアプリケーションに先読み可能属性を設定することにより達成される）。図５に、マッピングされたメモリ６１内に割り当てられたデータを図示する。

【０１５５】

伝送システム５０には、ネットワークインターフェイス２０へのデータ書き込みを調整する書き込みコントローラ５２が組み込まれている。書き込みコントローラ５２は、内部システムバスを使って、マッピングされたメモリ５５（伝送バッファ２９）に外部送信ネットワークデータ７４を書き込む。外部送信ネットワークデータ７４は、ネットワークインターフェイス２０の介入なしに、伝送バッファ２９に書き込まれる。書き込みコントローラは、ネットワークインターフェイスに書き込む前に、ホストコンピュータシステムプロセッサに割り当てられている書き込み－結合バッファ内の外部送信ネットワークデータ７４を集める。このことより、外部送信ネットワークデータ７４は伝送バッファ２９にバーストとして（小さなビットパッケージではなく）書き込まれる。

【０１５６】

書き込みコントローラ５２は、典型的には、外部ネットワークへ送信するために作成されたデータフレームで、外部送信ネットワークデータ７４を伝送バッファ２９に書き込む。このことにより、ネットワークインターフェイス２０は、最少の追加処理で、外部送信ネットワークデータ７４を伝送バッファ２９から抽出してデータ送信できる。データフレームは制御データ７３と結合されてもよい。そのことにより、伝送バッファ２９からの個々のフレーム抽出が容易になる。

【０１５７】

図４に例示されている伝送システム５０には、制御データ７３を伝送バッファ２９に書き込まれたデータフレームと結合する制御モジュール５３が組み込まれている。制御モジュール５３は、各フレームが書き込みコントローラ５２によりネットワークインターフェイス２０に転送される前に、各データフレームの前に制御データ７３を付加する。書き込みコントローラ５２は、対応するデータフレームと同じトランザクションで、制御データ７３を伝送バッファ２９に書き込む。制御データ７３は、通常、対応するデータフレームの前に付けられ、伝送バッファの中に外部送信ネットワークデータ７４と共に保存される。図４に例示された制御モジュール５３により生成された制御データ７３は、以下のデータを含む。

・フレーム識別子（７６）　：１６ビット
・フィードバックオフセット（７７）　：１６ビット
・フレーム長（７８）　：１６ビット
・未使用（７９）　：１６ビット

【０１５８】

制御モジュール５３は、各データフレームに対して８バイト（６４ビット）の制御データを生成する。制御データの２バイト（１６ビット）は、拡張のために割り当てられている（「未使用」データフィールド（７９））。

【０１５９】

制御モジュール５３により生成される制御データ７３は、フィードバック記述子を含み、それによりネットワークインターフェイスがフレーム伝送を報告できる。また制御データ７３は、伝送バッファ２９内の対応するフレームのサイズを定義する長さフィールドを

含む。各フレームの前に付加されている制御データ７３は標準化されている。

【０１６０】

制御モジュール５３は、フレームリファレンスをネットワークインターフェイス２０内の離間した制御レジスタ６２に書き込むことにより、外部ネットワークへのデータ伝送を開始する。フレームリファレンスは、所定のバッファメモリ内の対応するデータフレームの位置を規定する。制御モジュール５３は、典型的には、フレームリファレンスを所定のメモリバッファ２９におけるオフセットとして特定する。

【０１６１】

ネットワークインターフェイスに組み込まれている伝送エンジン（図示せず）は、外部ネットワークへ伝送するために、バッファ２９から個々のデータフレームを抽出する。フレームリファレンスとフレーム長（７８）フィールドは、所定のメモリバッファ内における対応するデータフレームの境界を規定する。伝送エンジンは、これらのフィールドを、伝送バッファ２９からフレームを抽出するために用いる。

【０１６２】

伝送エンジンは、フレームリファレンスにより規定されたバッファメモリアドレスでフレームポインタを初期化することにより、抽出プロセスを開始する。フレームポインタは典型的には、データフレームの前に付加された制御データ７３に一致する。伝送エンジンは、フレームの制御データ７３に含まれるフレーム長７８を読み出し、そのフレームの最後のアドレスを定義する。

【０１６３】

ネットワークインターフェイス２０は、伝送バッファ２９から抽出されたフレームを、外部ネットワークへ送信する。もしソースソフトウエアアプリケーションから要求される場合は、送信後に、コンファメーションメッセージがホストコンピュータシステムに送られてもよい。通知エンジン（図示せず）は、対応するフレームの制御データ７３に含まれているフィードバック記述子を使って、コンファメーションメッセージ送信を管理する。通知エンジンは、対応するデータフレームと同じトランザクションで、所定のネットワークインターフェイスメモリからフィードバック記述子を抽出する。フィードバック記述子は、フレームを生成したソフトウエアアプリケーションにより規定される。

【０１６４】

通知エンジンは、所定のネットワークインターフェイスメモリから抽出されたフレーム識別子７６を、ホストコンピュータシステムメモリ内に割り当てらえたフィードバックレジスタに書き込み、フレーム伝送を報告する。フィードバックレジスタは、制御データ７３に組み込まれたフィードバックオフセットフィールド７７により規定される。これは、典型的には、ソフトウエアアプリケーションが書き込み確認のためにモニターするフィードバックアレイ５４（ホストコンピュータシステムメモリ内に保持されている）内のオフセットを表示する。ソフトウエアアプリケーションは、フィードバックオフセットフィールド７７内で規定されているビットを設定することにより、コンファメーションメッセージを拒否できる。伝送プロトコルの実施例を図６にフローチャートで示す。

【０１６５】

書き込みコントローラ５２とメモリ管理モジュール５１を使って伝送プロセスを実行する特定プロトコルは、ホストプロセッサとネットワークインターフェイスとの間のデータ送受信をするシステムバスに依存する。ＰＣＩバス標準を用いて実行する例をここで簡単に説明する。伝送プロセスは、他のシステムバス標準を使っても実行できる。

【０１６６】

メモリ管理モジュール５１は、伝送バッファの所定のインターフェイスメモリを、ホストコンピュータシステムメモリ階層６１内の所定の基本アドレスレジスタ（ＢＡＲ）（例えばＰＣＩバスのＢＡＲ２（７２））にマッピングする。このことにより、書き込みコントローラ５２が、マッピングされたメモリ５５に対して書き込み－結合を使うことができる。メモリ管理モジュール５１も、マッピングされたメモリ５５に対して書き込み－結合することができる。フレーム伝送を開始するのに用いられる制御レジスタ６２は、メモリ

管理モジュール５１により、Ｉ／Ｏスペースに離間して（典型的には、基本アドレスレジスタ０（６０））マッピングされる。

【０１６７】

（ネットワークインターフェイス内でのネットワークデータの伝送）

図２に例示されているネットワークインターフェイス２０は、従来は専用のネットワークスイッチング装置により実行されていた、いくつかのデータ伝送機能を実行することができる。ネットワークインターフェイス２０は、ネットワークデータを内部のポート間で転送することにより、追加の伝送機能が対応可能になっている。このことにより、従来はネットワークデータを中間にあるスイッチに転送していたために発生していた遅延を避けることができる。

【０１６８】

ネットワークインターフェイスのネットワーク面２１は複数のデータポートを含む。これらのデータポートは、図２に例示されているように、互換性のある物理的媒体と物理的に接続できるハードウエアコネクタ（例えば小型フォームファクタモジュール（Ｓｍａｌｌ　Ｆｏｒｍ‐Ｆａｃｔｏｒ　Ｍｏｄｕｌｅｓ））で一体化されていてもよい。各データポートは、ネットワークインターフェイス制御システム２５と外部データネットワークとの間のネットワークデータのやり取りのためのデータチャネルを規定する。ネットワークインターフェイスには、外部ネットワークとの無線データチャネルを規定する無線データポート（図示せず）も組み込まれていてもよい。

【０１６９】

ネットワークインターフェイス制御システム２５は、個別のデータポートでやり取りしたネットワークデータをロギングポートにコピーすることにより、データログを取る（ｄａｔａ　ｌｏｇｇｉｎｇ）ことを容易にする。コピーされたネットワークデータは、外部ロギングシステムに送信するために、統合されたデータストリームにまとめられる。ネットワークインターフェイス制御システムは、ホストコンピュータシステムからの介入なしに、ロギングプロセスを促進する。

【０１７０】

ネットワークインターフェイス制御システム２５のデータロギングが可能な実施例には、ネットワークインターフェイス２０内でのデータロギングを調整するロギングモジュール８５が組み込まれている。ロギングモジュール８５は、個別のデータチャネル（ネットワークインターフェイスポートにより規定される）からのネットワークデータを複製し、その複製データを所定のロギングポートへ転送する。ロギングユニットは統合されたロギングデータストリームを生成し、それはロギングポートを介して外部データネットワークに送信される。

【０１７１】

ロギングモジュール８５は、動的なロギングリコンフィグレーションを促進するロギングインターフェイスを含む。ロギングインターフェイスは、ロギング（複製して遠隔のロギングサーバに送信する）のためのデータチャネルを指定する複数の制御レジスタを有する。制御レジスタは、システムアドミニストレータにより、個別のポートについてロギングできる／できないを、動的に設定されてもよい。

【０１７２】

ロギングモジュールはデータチャネルの中の離れたデータストリームに対して独立にロギングを促進してもよい。独立したデータストリームのロギングは、ロギングコントローラにより調整される。ロギングコントローラは、個別のデータストリームを一つのデータチャネルから分離し、指定されたストリームをロギングユニットに独立に転送する。ロギングインターフェイスは、所定の制御レジスタをデータチャネル内の個別のストリームに割り当てることにより、データストリームロギングを促進する。

【０１７３】

図７に、いくつかのデータチャネル内のデータストリームを送受信するための、独立したロギングについてのブロック図を示す。各データポートは、受信データストリーム８７

と送信データストリーム８８をそれぞれ伝搬する受信エンジン８１と送信エンジン８２とで表示される。各データストリームは、複製され、ロギングモジュール８５にコピーされる。

【０１７４】

所定のロギングユニット（図示せず）は、ネットワークインターフェイスから（ロギングポート送信エンジン８３を介して）送信するために、個々のデータフレームを結合させて、統合ロギングストリーム９０を作成する。統合ロギングストリームに組み込まれる個々のデータフレームは、動的に設定変更可能な制御レジスタを使って転送モジュール８５により選択される。

【０１７５】

図７に示すロギングポートは、ロギングモジュール８５により生成されたロギングストリーム９０に加えて、通常のネットワーク通信も管理する。マルチプレクサ８４は、ロギングポート送信ストリーム８９とロギングストリーム９０からの外部送信ネットワークデータを、ロギングポート送信エンジン８３に転送する。マルチプレクサ８４は、ロギングユニットに組み込まれていてもよい。

【０１７６】

ロギングポート送信ストリーム８９とロギングポート受信ストリーム８６は、複製され、ロギングモジュール８５に転送されて、他のデータポートからのデータストリームと共にロギングストリーム９０に含まれる。

【０１７７】

ネットワークインターフェイス制御システム２５は、ネットワークインターフェイス２０の複数のポート間のデータ伝送を促進してもよい。データ伝送は、ネットワークインターフェイスの所定のポート（転送ポート）から受信したデータを他のポート（送り先ポート）へ転送することにより、促進される。例示したネットワークインターフェイス２０は、ホストコンピュータシステムの介入なしに、転送ポートで受信したデータを自律的に伝送する。このことにより、ネットワークインターフェイス２０の複数のポート間のデータ転送が可能になる。

【０１７８】

伝送可能なネットワークインターフェイス制御システムは、転送ポートから直接データを受信し、受信したデータを他のポート（送り先ポート）を介して外部データネットワークに転送する転送モジュールを含む。転送モジュールは、受信したネットワークデータを、ホストコンピュータシステムの介入なしに、自律的に転送できる。このことにより、ホストコンピュータシステムのオペレーティングシステムへのデータ転送により遅延を避けることができる。

【０１７９】

フィルタリングモジュールは転送ポートで受信したネットワークデータをフィルタでき、ホストコンピュータシステムへ宛てられたデータではないデータを選択して伝送する。フィルタエンジン（通常転送モジュールに組み込まれている）は、転送ポートで受信したデータの送り先アドレスを読む。送り先アドレスは、典型的には、ネットワークデータに付随しているメタデータ（例えばパケットヘッダ）に規定されている。フィルタリングモジュールは、送り先アドレスが、ホストコンピュータシステムのアドレスと一致するデータを保持する。

【０１８０】

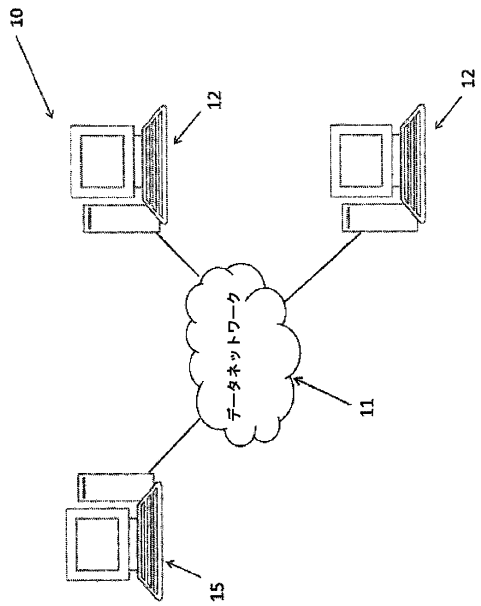
転送インターフェイスは、転送モジュールの構成を促進する（フィルタリングエンジンの起動を含む）。転送インターフェイスは、システム管理者による設定に応じて二つのポート間のデータ伝送をできなくする、制御レジスタを有する。

【０１８１】

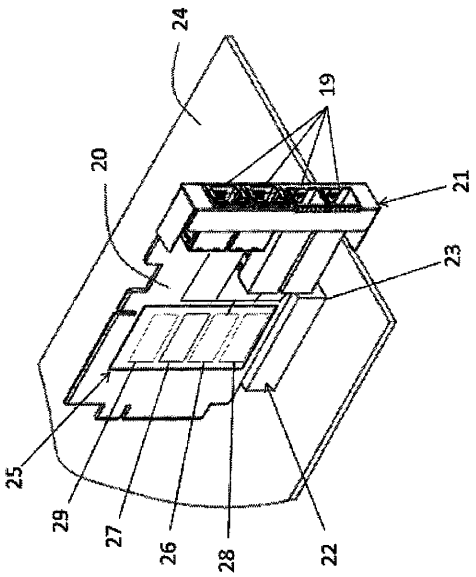
ネットワークインターフェイスに伝送機能があることにより、コンピュータシステムは、外部ネットワークスイッチや光学スプリッタを使うことなしに、共通のネットワーク接続を共有できる。

【０１８２】
図８に典型的なポート伝送の実施例９１を示す。図に示した実施例９１は、受信データを相互にやり取りする、連結された複数のデータポートを有し、シンメトリーなポート伝送構成となっている。各データポートは、外部ネットワークとの通信を促進する所定の受信エンジン９２と送信エンジン９３を含む。
【０１８３】
到来したネットワークデータフレームは受信エンジン９２により受信される。各ポートは、受信エンジン９２で受信されたすべてのフレームを、フィルタリングエンジン９７に転送する。各受信エンジン９２には、実施例に例示されているように、所定のフィルタリングエンジン９６がある。
【０１８４】
フィルタリングエンジン９７は、各受信データフレームから送り先アドレスを読み出す。送り先アドレスがホストコンピュータシステムのアドレスと一致するデータフレームは、ネットワークインターフェイス制御システムに転送して受信バッファ格納する。ホストコンピュータシステムのアドレスとは異なる送り先アドレスを有するデータフレームは、外部ネットワークは転送されるために抽出される。
【０１８５】
フィルタリングエンジンは、データフレームを転送ポートから複数の送り先（通常、送り先ポート及びホストコンピュータシステム）に配信する。このことにより、送り先アドレスがホストコンピュータシステムのアドレスを一致しないとき、フィルタリングエンジンがデータフレームをネットワークインターフェイス制御システム２５に転送できる（ロギングとアプリケーションに役立つ）。フィルタリングエンジンにより採用されるデータ配信の基準は、典型的には、転送インターフェイスを使って規定される。
【０１８６】
フィルタリングエンジンは、複数の送り先に配信するために転送ポートで受信したデータフレームのサブセットを選択してもよい。これは、望まれていないデータ（例えば放送データやマルチキャストデータ）を取り扱うときに良く使われる。フィルタリングエンジンは、無差別モードに設定さえた場合は、転送ポートで受信したデータフレームを無差別に配信してもよい。
【０１８７】
フィルタリングエンジン９７は、連結された送り先データポートの送信エンジン９５に、フィルタされたデータストリーム（ホストコンピュータシステムのアドレスとは異なるアドレスであるフレーム）を転送する。マルチプレクサ９６は、フィルタされたデータストリームをホストコンピュータシステムからの外部送信ネットワークデータと結合する。
【０１８８】
図８に示したポート伝送の実施例により、「マスター」コンピュータシステムは付加的なネットワークスイッチ無しで、「スレーブ」コンピュータシステムとネットワーク接続を共有することができる。例示された実施例は、「マスター」コンピュータシステム（外部ネットワークと直接接続されているコンピュータシステム）のためのポート伝送の構成を規定している。
【０１８９】
特許請求の範囲及びこの明細書において、「含む」またはその変形である「含み」、「含んで」等は、言葉の表現や必要なニュアンスにより前後の文脈から必要となる場合を除いて、「少なくとも含む」という意味で使われている。すなわち、述べられた特徴が存在することを述べているが、本発明の他の実施例における他のさらなる特徴の存在又は追加を排除するものではない。
【０１９０】
当業者であれば、本発明の範囲を逸脱することなく、本発明に多くの変形例が可能であることは理解できると考えられる。

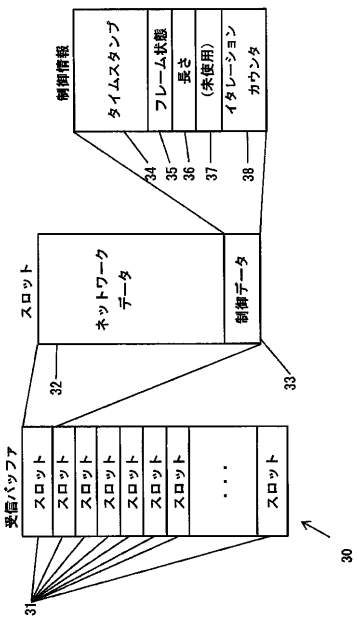
【図１】


10
12
12
データネットワーク
11
15

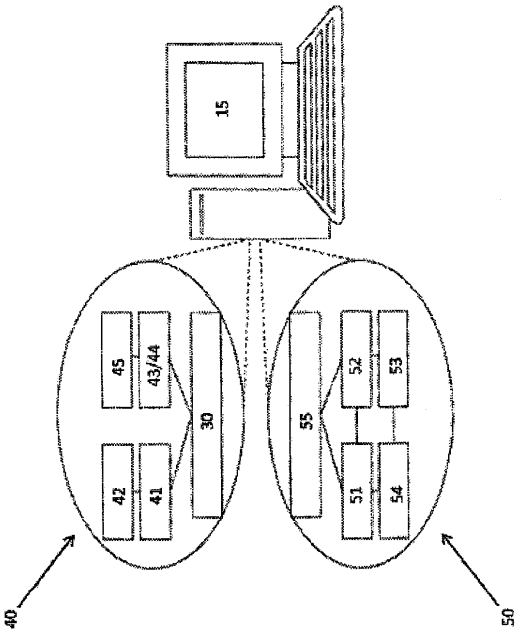
【図２】


24
19
20
21
23
25
29
27
26
28
22

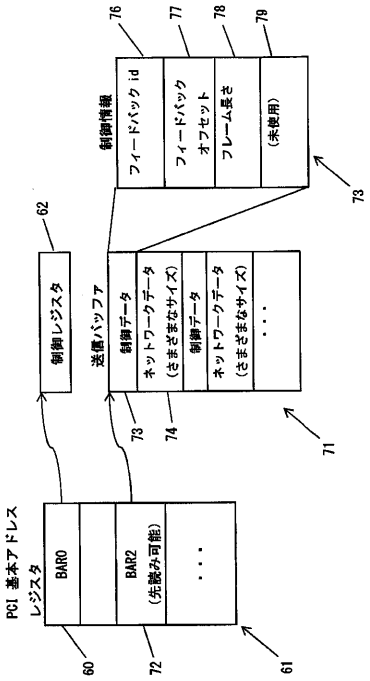
【図３】


制御情報
タイムスタンプ
フレーム状態
長さ
(未使用)
イタレーション
カウンタ
34
35
36
37
38
スロット
ネットワーク
データ
制御データ
32
33
受信バッファ
スロット
スロット
スロット
スロット
スロット
スロット
スロット
31
30

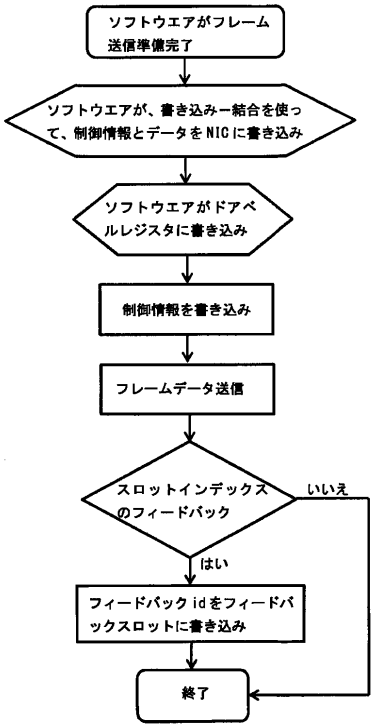
【図４】


15
45
43/44
30
42
41
55
52
53
51
54
40
50

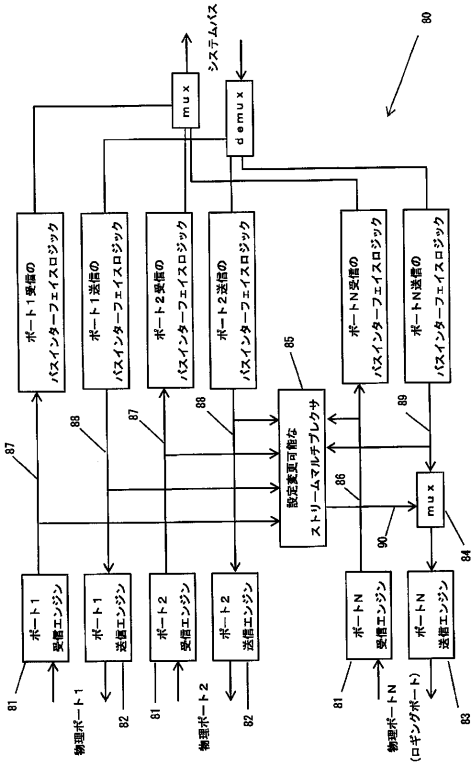
【図５】


PCI 基本アドレスレジスタ
BAR0
BAR2
(先読み可能)
制御レジスタ
送信バッファ
制御情報
フィードバック id
フィードバックオフセット
フレーム長さ
(未使用)
制御データ
ネットワークデータ
(さまざまなサイズ)
60
72
61
62
73
74
71
76
77
78
79

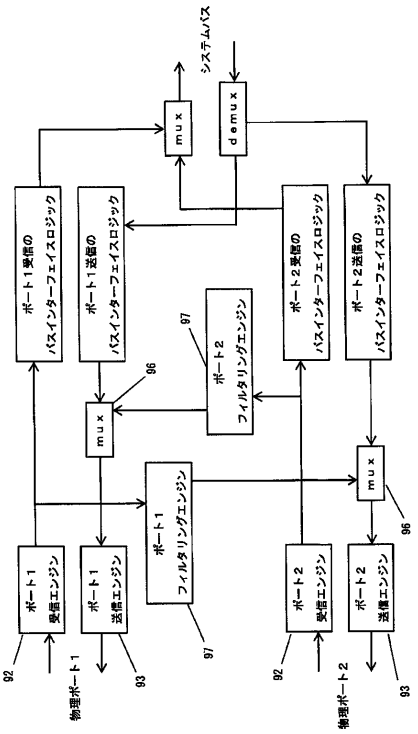
【図６】


ソフトウエアがフレーム送信準備完了
ソフトウエアが、書き込み一結合を使って、制御情報とデータをNICに書き込み
ソフトウエアがドアベルレジスタに書き込み
制御情報を書き込み
フレームデータ送信
スロットインデックスのフィードバック
いいえ
はい
フィードバックidをフィードバックスロットに書き込み
終了

【図７】


システムバス
mux
demux
ポート1受信のバスインターフェイスロジック
ポート1送信のバスインターフェイスロジック
ポート2受信のバスインターフェイスロジック
ポート2送信のバスインターフェイスロジック
ポートN受信のバスインターフェイスロジック
ポートN送信のバスインターフェイスロジック
設定変更可能なストリームマルチプレクサ
ポート1受信エンジン
ポート1送信エンジン
ポート2受信エンジン
ポート2送信エンジン
ポートN受信エンジン
ポートN送信エンジン
物理ポート1
物理ポート2
物理ポートN
(ロギングポート)
80
81
82
83
84
85
86
87
88
89
90

【図８】


システムバス
mux
demux
ポート1受信のバスインターフェイスロジック
ポート1送信のバスインターフェイスロジック
ポート2受信のバスインターフェイスロジック
ポート2送信のバスインターフェイスロジック
ポート2フィルタリングエンジン
ポート1フィルタリングエンジン
ポート1受信エンジン
ポート1送信エンジン
ポート2受信エンジン
ポート2送信エンジン
物理ポート1
物理ポート2
92
93
96
97

フロントページの続き

(72)発明者　チャップマン　マシュー
オーストラリア　ニュー　サウス　ウェールズ　２０３４，　クーギー，　３／２７　バイロン　ストリート

審査官　宮島　郁美

(56)参考文献　特開２００８－０９７２７３（ＪＰ，Ａ）
米国特許出願公開第２００７／０２６０７７７（ＵＳ，Ａ１）
特開平０６－１９７１３３（ＪＰ，Ａ）
米国特許出願公開第２００９／００３１０５８（ＵＳ，Ａ１）
米国特許出願公開第２００８／００２８１０３（ＵＳ，Ａ１）
特開平０６－２７４４２５（ＪＰ，Ａ）
米国特許出願公開第２００５／０２２０１２８（ＵＳ，Ａ１）
特開２００６－０２０３７１（ＪＰ，Ａ）
特表平０８－５０２６１３（ＪＰ，Ａ）
特開２００７－１２４３１３（ＪＰ，Ａ）
米国特許出願公開第２００６／００７５１１９（ＵＳ，Ａ１）
ＮＩＣまでのRMONエージェントをサポートする「Ｔｒａｎｓｃｅｎｄ　ｄRMON　Ｅｄｇｅ　Ｍｏｎｉｔｏｒ　Ｓｙｓｔｅｍ」，コンピュータ＆ネットワークＬＡＮ，日本，株式会社オーム社，１９９７年　９月　１日，第15巻　　第9号，112～115

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ１２／００－１２／９５５，２９／００
Ｇ０６Ｆ１３／００