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

特許第4334760号
(P4334760)

(45)発行日　平成21年9月30日(2009.9.30)　(24)登録日　平成21年7月3日(2009.7.3)

(51)Int.Cl.　　F I
G06F 13/12 (2006.01)　　G06F 13/12 330A
H04L 12/46 (2006.01)　　H04L 12/46 100Z

請求項の数 20　(全 15 頁)

(21)出願番号　特願2000-504685(P2000-504685)
(86)(22)出願日　平成10年7月21日(1998.7.21)
(65)公表番号　特表2001-511575(P2001-511575A)
(43)公表日　平成13年8月14日(2001.8.14)
(86)国際出願番号　PCT/IB1998/001117
(87)国際公開番号　WO1999/005826
(87)国際公開日　平成11年2月4日(1999.2.4)
審査請求日　平成16年7月26日(2004.7.26)
(31)優先権主張番号　08/900,757
(32)優先日　平成9年7月25日(1997.7.25)
(33)優先権主張国　米国(US)

前置審査

(73)特許権者　500035465
ネグザビット・ネットワークス，リミテッド・ライアビリティー・カンパニー
アメリカ合衆国マサチューセッツ州０１５８１，ウエストボロー，ウエスト・パーク・ドライブ・１７００，スイート・３９０
(74)代理人　100087642
弁理士　古谷　聡
(74)代理人　100076680
弁理士　溝部　孝彦
(74)代理人　100121061
弁理士　西山　清春

最終頁に続く

(54)【発明の名称】ネットワーキングシステム

(57)【特許請求の範囲】

【請求項１】

共通メモリと、パケット／セルの受信、前記共通メモリへのパケット／セルの書き込み、及び該共通メモリからのパケット／セルの除去を行う複数のI/0モジュールとに接続された共通バスを介してデータがインターフェイスされる、データ制御システムにおいて、メモリ及びバスアクセスの競合及びその結果として生じるシステムの待ち時間を低減させるための方法であって、該方法が、各I/0モジュール（#1-#n）により受信されたパケット／セルからの制御情報の抽出及び該制御情報の送出エンジン（FE）への提供を別個のパスを介して行い、パケット／セルの前記抽出された制御情報を前記送出エンジン（FE）で処理して、そのデータがメモリに書き込まれている間にスイッチング、ルーティング、及び／又はフィルタリングの決定を行い、該送出エンジンによる処理結果を待ち行列マネージャ（QM）に送って各パケット／セルの受信待ち行列及び送信待ち行列に対する待ち行列の出し入れを行い、対応するI/0モジュールの送信待ち行列手段を介して、共通メモリとパケット／セルデータを送信すべき適当な出口I/0モジュールとのインターフェイスを制御し、その全てを、メモリとの間でのパケット／セルデータの伝送と競合することなく及び該データ伝送とは独立して行う、という各ステップを有する、システムの待ち時間を低減させるための方法。

【請求項２】

前記制御情報がパケット／セルデータのヘッダから抽出され、そのバッファアドレス情報が待ち行列マネージャ（QM）により処理される、請求項１に記載の方法。

【請求項３】
パケット／セルデータがメモリに完全に書き込まれる前に前記送出エンジンの結果が前記待ち行列マネージャ（QM）に送られ、これによりパケット／セルデータがメモリに完全に書き込まれた直後にデータの読み出しを開始させることが可能となる、請求項２に記載の方法。
【請求項４】
前記パケット／セルデータの読み出しが、該データがメモリに完全に書き込まれる前に開始される、請求項２に記載の方法。
【請求項５】
データ入口とデータ出口との間で同期を行って、パケット／セルの読み出しの開始が早すぎないことを確実にする、請求項４に記載の方法。
【請求項６】
前記送出エンジン（FE）が、全てのパケット／セルデータがメモリへ書き込まれたことが検出された際に、その処理結果を前記待ち行列マネージャ（QE）に送る、請求項５に記載の方法。
【請求項７】
受信したパケット／セルのバッファアドレスが適当な出口I/0モジュールに送られる、請求項２に記載の方法。
【請求項８】
待ち行列が分散され、前記待ち行列マネージャ（QM）により読み出される待ち行列の数が、システム全体ではなく特定のI/0モジュールをサポートするのに必要な待ち行列の数まで低減される、請求項７に記載の方法。
【請求項９】
各I/0ポート毎に多数の待ち行列が維持され、前記待ち行列マネージャ（QM）が複数の独立したサブ待ち行列マネージャへと分割され、その各サブ待ち行列マネージャがそれに関連するI/0モジュールの待ち行列を処理する、請求項７に記載の方法。
【請求項１０】
マルチキャスト情報が各サブ待ち行列マネージャへ同時に供給される、請求項９に記載の方法。
【請求項１１】
ＣＰＵ又はそれに類似したデータ制御システムにおけるメモリ及びバスアクセスの競合及びその結果として生じるシステムの待ち時間を低減させるための装置であって、共通メモリと、パケット／セルデータの受信、前記共通メモリへのパケット／セルデータの書き込み、及び該共通メモリからのパケット／セルデータの除去を行う複数のI/0モジュール（#1-#n）とに接続された共通バスを介してデータがインターフェイスされ、該装置が、前記I/0モジュールにそれぞれ配設された複数の送出エンジン（FE）及び送信待ち行列手段と、前記I/0モジュールにより受信されたパケット／セルから制御情報を抽出して該制御情報を前記送出エンジン（FE）に提供するための別個のパスとを備えており、前記送出エンジン（FE）の各々が、対応するI/0モジュールにより受信されたパケット／セルから抽出された制御情報を処理して、そのデータがメモリに書き込まれている間にスイッチング、ルーティング、及び／又はフィルタリングの決定を行い、該装置が更に、前記送出エンジン（FE）による処理結果を待ち行列マネージャ（QM）に送って各パケット／セルの受信待ち行列及び送信待ち行列に対する待ち行列の出し入れを行う手段であって、対応するI/0モジュールの送信待ち行列手段を介して、共通メモリとパケット／セルデータを送信すべき適当な出口I/0モジュールとのインターフェイスを制御し、その全てを、メモリとの間でのパケット／セルデータの伝送と競合することなく及び該データ伝送とは独立して行う手段を備えている、システムの待ち時間を低減させるための装置。
【請求項１２】
前記制御情報がパケット／セルデータのヘッダから抽出される、請求項１１に記載の装置。

【請求項１３】
前記送出エンジン（FE）が、前記待ち行列マネージャ（QM）にバッファアドレス情報を提供する、請求項１２に記載の装置。
【請求項１４】
パケット／セルデータがメモリに完全に書き込まれる前に前記送出エンジン（FE）の結果を前記待ち行列マネージャ（QM）に送る手段を備えており、これによりパケット／セルデータがメモリに完全に書き込まれた直後にデータの読み出しを開始させることが可能となる、請求項１３に記載の装置。
【請求項１５】
パケット／セルデータがメモリに完全に書き込まれる前に該データの読み出しを開始させる手段を備えている、請求項１３に記載の装置。
【請求項１６】
データ入口とデータ出口との間で同期を行ってパケット／セルの読み出しの開始が早すぎないことを確実にする手段を備えている、請求項１５に記載の装置。
【請求項１７】
パケット／セルデータの最後のビットがメモリへ書き込まれたことが検出された際に前記送出エンジン（FE）にその処理結果を前記待ち行列マネージャ（QM）へ送らせる手段を備えている、請求項１６に記載の装置。
【請求項１８】
受信したパケット／セルのバッファアドレスが適当な出口I/Oモジュールに送られる、請求項１３に記載の装置。
【請求項１９】
待ち行列が分散され、前記待ち行列マネージャ（QM）により読み出される待ち行列の数が、システム全体ではなく特定のI/Oモジュールをサポートするのに必要な待ち行列の数まで低減される、請求項１８に記載の装置。
【請求項２０】
各I/Oポート毎に多数の待ち行列が維持され、前記待ち行列マネージャ（QM）が複数の独立したサブ待ち行列マネージャへと分割され、その各サブ待ち行列マネージャがそれに関連するI/Oモジュールの待ち行列を処理する、請求項１８に記載の装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、ネットワーキングシステム及びその内部における情報の送出及びルーティングに関し、特に共有メモリへのアクセスの競合や送出及び／又はルーティングの決定に要する時間に起因するシステムの待ち時間の問題に関し、とりわけかかる待ち時間の最小化に関する。
【０００２】
【従来の技術】
ネットワーキングシステムにおけるシステム性能に要求される主な要因及び関心事のうちの２つとして、帯域幅能力及びシステム動作待ち時間が一般に挙げられる。帯域幅は、システムを介して伝送できるデータの量を反映し、待ち時間は、データがシステム中に「滞在する」時間の量に関係するものとなる。
【０００３】
本発明は、待ち時間の最小化に関するものである。本出願人の「High Performance Universal Multi-Port Pnternally Cached Dynamic Random Access Memory System, Architecture and Method」と題する同時係属中の米国特許出願第581,467号（1995年12月29日出願）では、帯域幅を最大化するための前途有望な解決策が提供されている。
【０００４】
ネットワークシステムの待ち時間は、幾つかの要因により決定され、その主な１つが、データパケット又はセルの始まりにおいて制御情報を検査する結果として送出又はルーティ

ングの決定を行うために要する時間の量である。該制御情報は、セル又はパケットを伴うか否かによって異なる。セルの場合には、システム内の出口(egress)インターフェイスに対するセルのマッピングに使用することが可能なＶＣＩ／ＶＰＩ情報に基づいてスイッチング決定が行われる。一方、パケットの場合には、出口インターフェイスに対するパケットのマッピングに使用することが可能なデスティネーション（即ち宛先）アドレスに基づいてルーティング決定が行われる。更に、パケットの場合には、ソース（即ち発信元）アドレスを使用して、複数対のソース／デスティネーションアドレス対に基づく１レベルのフィルタリングを提供することが可能であり、この場合には、どのソース／デスティネーションアドレス対の通信が可能であるかを決定するために多数のルールが設定される。かかるルールに適合しないパケットが受信された場合には、該パケットは破棄される。例えば、典型的には、この種のネットワークにおけるデータは、セルの場合には53byte、パケットの場合には64～64Kbyteとなる。

【０００５】

従来のシステムでは、制御情報の処理は、中央処理装置（ＣＰＵ）により実行され、セル／パケット全体が受信されるまで開始されない。かかるシステムの待ち時間は、I/0ポートからメモリへのデータの伝送、データの最初に配置された制御情報のアクセス、該制御情報の更新、及びメモリからI/0ポートへのデータの伝送によって決まる。これらの共有メモリへのアクセスの全ての結果として、かなりのバス及びメモリ競合が生じることになり、これにより待ち時間が増大することになる。該待ち時間は、この種のアーキテクチャではかなり重大なものとなる。これは、パケット／セル全体が受信されるまで制御情報の処理を開始することができないからである。待ち時間を増大させることになる他の要因としては、ＱＯＳ(Quality of Service)及びマルチキャストのサポートが挙げられる。ＱＯＳは、各I/0ポート毎に複数の待ち行列を維持することを必要とし、これにより、既に過剰負荷となっているメモリへのアクセス数が増大する。また、マルチキャストは、複数のI/0ポートに対する同一パケット／セルの送信を必要とし、これもまた、過剰負荷となっているメモリへのアクセス数を増大させるものとなる。

【０００６】

システムの待ち時間を決定する更に別の要因は、共有メモリのスループットである。共有メモリのスループットがあまり高くない場合、それに従って待ち時間も増大する。一般に、フル帯域幅をサポートするには、メモリのスループットは、ポート速度にポート数を乗じた値の２倍に等しい必要がある。しかし、これは、同一の共有メモリに対して行わなければならない他のアクセス全てを考慮したものではなく、このため、待ち時間を最小限にすると共にシステムを介した高い帯域幅を達成するためには、メモリのスループットを一層高くする必要がある。更に、一層多数のポートが追加され、及び各ポートの速度が増大されると、それに比例して待ち時間が増大する。したがって、共有メモリシステムのスループットの増大は極めて困難な問題となる。

【０００７】

後に例証するように、殆どの従来のネットワーキングシステムの動作は、ゼロ又はほぼゼロの待ち時間の達成を許さないものである。一方、本発明によれば、新規のパケット／セルデュアルパスデータ処理及び管理アーキテクチャを用いることにより、結果的に最適に最小化された待ち時間を達成することができる。

【０００８】

【発明が解決しようとする課題】

したがって、本発明の目的は、待ち時間を劇的に低減させる、データパケット及び／又はセル等のデュアルパスデータ処理及び管理に関する新規のシステムアーキテクチャ及びその方法を提供することにある。

【０００９】

本発明の更なる目的は、待ち時間を最小限にするこの新規の結果を、各パケット／セルの制御情報を処理している間に他のリソースと競合することなく達成することにある。

【００１０】

本発明の更なる目的については、以下で説明すると共に、特許請求の範囲に一同詳細に表すこととする。

【００１１】

【課題を解決するための手段】

要するに、本発明は、その重要な１つの観点からすれば、ＣＰＵ又はデータ制御システムに包含されるものであり、この場合、データは共通バスに沿ってインターフェイスされ、該共通バスは、共通メモリと、データパケット／セルの受信及び前記メモリへの書き込み及び該メモリからの除去を行う複数のI/0モジュールとに接続される。本発明はまた、メモリアクセスの競合及びその結果として生じるシステムの待ち時間を低減させるための方法を包含し、該方法は、各I/0モジュールに、対応する送出エンジン、送信待ち行列手段、及び該I/0モジュールにより受信されたパケット／セルから制御情報を抽出して該制御情報を前記送出エンジンに提供するための別個のパスを設け、前記抽出されたパケット／セルの制御情報を前記送出エンジンで処理して、そのデータをメモリに書き込んでいる際にスイッチング、ルーティング、及び／又はフィルタリングの決定を行い、送出エンジンによる処理結果を待ち行列マネージャに送って各パケット／セルの受信待ち行列及び送信待ち行列に対する待ち行列の出し入れを行い、対応するI/0モジュールの送信待ち行列手段を介して、パケット／セルデータを送信すべき適当な出口I/0モジュールとのインターフェイスを制御し、その全てを、メモリとの間でのパケット／セルデータの伝送と競合することなく及び該データ伝送とは独立して行う、という各ステップを有するものである。

【００１２】

好適でベストモードの構成及び技法について以下に詳述する。

【００１３】

【発明の実施の形態】

従来技術による現行ネットワークシステムにおける待ち時間の制限

既述のように、ネットワークシステムと共に使用される典型的なパケット／セル構成では、制御情報は、図１に概略的に示すように、パケットの最初に配置される。既述のように、システム内の出口インターフェイスに対してセルをマッピングするために使用されるＶＣＩ／ＶＰＩ情報に基づくスイッチング決定が示されている。パケットのルーティング決定は、出口インターフェイスに対するパケットのマッピングに使用されるデスティネーションアドレスに基づいて行われる。

【００１４】

図２の従来のシステムでは、ＣＰＵは、共通バスを介して、メモリアクセス、及び複数のデータ受信及び除去I/0ポート#1,#2等とのインターフェイスを行い、周知のように、様々な点線及び破線は、共有メモリとのインターフェイスパスを示している。上記で指摘したように、共有メモリの様々なアクセスの結果として、かなりの競合が生じて、待ち時間が増大する。パケット／セル全体が受信されるまで制御情報の処理を開始することができないため、この種のアーキテクチャでは、該待ち時間の増大はかなり重大なものとなっている。

【００１５】

更に、図３から分かるように、共有メモリへのアクセスが増大すると、競合も増大し、競合が増大すると、システムの待ち時間が増大することになる。図３には（１回の読み出し又は書き込みについてのアクセスタイムがＭに等しく、１回のメモリアクセスについてのビット数がＷであり）以下の機能が示されている。

【００１６】

Ａ．受信ポート#1から共有メモリへのデータ書き込み。１パケット又はセルを伝送するための時間は（(Ｂ×８)／Ｗ）×Ｍに等しい（Ｂは１パケット又はセルのバイト数）。パケットが大きくなると、該パケットをメモリへ書き込むための時間も長くなる。

【００１７】

Ｂ．受信ポート#2から共有メモリへのデータ書き込み。１パケット又はセルを伝送するた

めの時間は（(Ｂ×８)／Ｗ）×Ｍに等しい（Ｂは１パケット又はセルのバイト数）。パケットが大きくなると、該パケットをメモリへ書き込むための時間も長くなる。

【００１８】

Ｃ．ポート#1から共有メモリへ書き込まれたパケット／セルからの制御情報の読み出し。これに要する時間は、読み出すべき制御情報の量によって決まる。該制御情報の量は、典型的には、パケットの場合には約24～28byte、セルの場合には5byteとなる。読み出すべきバイト数はＮに等しく、したがって読み出し時間は（(Ｎ×８)／Ｗ）×Ｍとなる。他のインターフェイスが同一の共有メモリについて競合するため、ポート#2が現在メモリへデータを書き込んでいることに起因して、該アクセスは一層長いものとなる、ということが理解されよう。これは、ポート#1で受信されたパケット／セルの待ち時間を増大させるものとなる。

【００１９】

Ｄ．ポート#1から受信されたパケット／セルのバッファアドレスの適当な待ち行列への書き込み。これは典型的には8～12byteとなる。該待ち行列を更新するための時間は（(Ｐ×８)／Ｗ）×Ｍとなる（Ｐは適当な待ち行列に書き込まれるべき待ち行列情報の長さ）。他のインターフェイスが同一の共有メモリについて競合するため、この場合も、該アクセスが一層長い時間を要し、ポート#1で受信されたパケット／セルの待ち時間が増大する。

【００２０】

Ｅ．どの待ち行列が送信に利用可能なデータを有しているかを判定するための異なる待ち行列の読み出し。これは、ポート#1からのパケット／セルのバッファアドレスが送信可能な状態になるまで多数の待ち行列の読み出しを行うことからなる。読み出される各待ち行列エントリは、典型的には8～12byteとなる。待ち行列を更新するための時間は(Ｑ＋１)((Ｐ×８)／Ｗ)×Ｍ)となる（Ｑは最後にパケットが待ち行列から出される前に読み出された待ち行列の数）。この場合も、他のインターフェイスが同一の共有メモリについて競合するため、該アクセスが一層長い時間を要し、ポート#1で受信されたパケット／セルの待ち時間が増大する。

【００２１】

Ｆ．共有メモリから受信ポート#2へのデータの読み出し。１パケット又はセルを伝送するための時間は（(Ｂ×８)／Ｗ）×Ｍに等しい（Ｂは１パケット又はセルのバイト数）。パケットが大きくなると、該パケットをメモリから読み出すための時間も長くなる。

【００２２】

勿論、本システムの目標は、待ち時間をゼロ（又はほぼゼロ）にすることにある。待ち時間がゼロになると、出口インターフェイスからメモリへのパケット／セルの書き込みと、該出口インターフェイスに関するメモリからの前記パケット／セルの読み出しとの間の時間がなくなることになる。実際に、出口インターフェイスが決定され、及びパケット／セルがメモリに完全に書き込まれる前にバッファアドレスが待ち行列から出された場合には、競合条件が存在し得る。該競合条件の結果として、メモリへのデータの書き込みが完了する前に該データの読み出しが開始されることになり、これにより不正データが伝送されることになる。共有メモリシステムでは、前述のように、パケット／セルがメモリに完全に書き込まれるまで制御情報や待ち行列を処理することができないため、ゼロ待ち時間を達成することは不可能である。

【００２３】

今日の典型的なシステムは、システムの一層高いスループットを提供することによりシステムの待ち時間を低減させようとするものであり、これは、待ち時間の低減に漸進的な利益を提供するものである。しかし、システム内の高いスループットの提供は、コスト及び複雑性を犠牲にすることによってしか達成することができない。ボトムライン（最低のライン即ち損益を示す最低の数字）は、I/0ポートのデータ伝送速度及び密度を増大させることであり、かかるシステムの待ち時間は削減されることはなく、実際には増大する。

図４のタイプのネットワーク

スイッチ、ルータ、ブリッジ、ハブ、ルーティングスイッチ、及びスイッチングルータ等

の典型的なネットワーク機器は、図４に示すように、ＡＴＭ、トークンリング、ＦＤＤＩ、Ehternet、及びSonetといった複数のネットワークと相互接続される。これらのネットワークの相互接続は、システムが受信した各パケット又はセルを参照してどのポートからパケット／セルを送信すべきかを決定するためにＣＰＵ又は送出エンジン（ＦＥ:Forwarding Engine）を必要とする。既述のように、ＣＰＵ／ＦＥは、各パケット／セルの最初の部分にアクセスして、該パケット／セルが如何なるタイプのデータであるか、及びそのデスティネーションが何処であるかを判定しなければならない。データ伝送速度及びI/0ポートが増大すると、同一メモリリソースに対する競合も増大し、これにより既述のようにシステムの待ち時間が増大する。待ち時間を低減させるための唯一の解決策は、メモリアクセスタイムの短縮であるが、これはコスト及び複雑性を高めるものとなる。共有メモリシステムの場合、該メモリシステムの性能は、最小限でも、全てのポートの帯域幅の２倍よりも大きくなければならない。例えば、システムがＮ個のポートを有しており、その各ポートがデータ伝送速度Ｖを有している場合には、該メモリシステムの総帯域幅は、２ＮＶよりも大きくなければならない。メモリシステムは、制御情報のルックアップ及び修正、並びに考え得る待ち行列の管理及びルーティングテーブルのルックアップ等もサポートしなければならないため、該総帯域幅は２ＮＶよりも大きくなければならない。メモリシステムが２ＮＶよりも大きい総帯域幅を有していなければならないため、これにより、性能のスケーラビリティ（即ち拡張性及び縮小性）が抑制され、その結果として待ち時間の低減におけるスケーラビリティが制限されることになる。

【００２４】

ＱＯＳを提供する場合には、１ポートにつき多数の待ち行列を維持することが必要となり、該アーキテクチャの結果として、待ち行列のアクセスの競合が増大することに起因して待ち時間が増大することになる。勿論、この場合には、一層高いメモリスループットが必要となり、コスト及び複雑性が増すことになる。

【００２５】

また、マルチキャストサポートを提供する場合には、やはり、共有メモリへのアクセス数が劇的に増大し、競合及び待ち時間が大幅に増大し、一層高いスループットのメモリシステムを設計する必要が生じる。

【００２６】

また、この種のシステムでは、パケット／セルがメモリに完全に書き込まれるまで制御情報及び待ち行列の処理を開始することができないため、ゼロ待ち時間を達成することも不可能である。

図５のタイプのネットワーク

図５は、図４と類似したものであるが、ＣＰＵ／ＦＥの前部にヘッダキャッシュが追加されている。パケット／セルは、各インターフェイスで送受信される際に、共有メモリ構造中にリード／ライトされるが、この場合には、該パケットの最初の64byteがヘッダキャッシュ内に「ミラー(mirror)」される。最初の64byteがヘッダキャッシュ中にコピーされ、１セルが53byteであるため、このアーキテクチャは、パケットベースのシステムにのみ適用可能なものとなる。ＣＰＵ／ＦＥは、パケットの制御情報にアクセスする場合に、実際には共有メモリからではなくヘッダキャッシュからデータを取り出す。これにより、制御情報に対するアクセスのみではあるが、共有メモリに対する競合が低減される。このため、このアーキテクチャは、図４に示す従来のアーキテクチャと比較して漸進的な改善を提供するものとなる。

【００２７】

しかし、従来のアーキテクチャの場合のように、メモリシステムは、各ポートについてのスループットの２倍よりも大きいことを依然として必要とする。共有メモリへのデータの書き込み、制御情報の処理、各ポート毎の待ち行列の設定及び維持、及び共有メモリからのデータの読み出しが、依然として一組をなす一連のアクセスとなっている。ポート及びデータ伝送速度が増大すると、それに従って同一の待ち時間を提供するためにメモリスループットをスケーリングする（即ち拡張する）必要がある。これは、コスト及び複雑性を

増大させることによってのみ達成することができ、該増大は、コストの制限上実施不能となるポイントに達するものとなる。

【００２８】

各ポート毎に多数の待ち行列を維持することを必要とするＱＯＳを提供する場合、このアーキテクチャは、待ち行列にアクセスするための競合の増大に起因して待ち時間を増大させるものとなる。この場合もやはりコスト及び複雑性を増大させるものとなる一層高いメモリスループットが必要となる。

【００２９】

マルチキャストサポートを提供する場合には、このアーキテクチャは、共有メモリへのアクセスを増大させ、競合及び待ち時間を大幅に増大させ、より高いスループットのメモリシステムを設計することが必要となる。

【００３０】

このシステムの場合も、パケットがメモリ中に完全に書き込まれるまで制御情報及び待ち行列の処理を開始することができないため、ゼロ待ち時間を達成することは不可能である。

図６のタイプのネットワーク

このシステムは、図４のシステムと類似したものである。この図６のシステムにおいて、パケット／セルが受信されると、該パケット／セルは、各I/0モジュール上のメモリ並びにＣＰＵ／ＦＥメモリ内に格納される。ＣＰＵ／ＦＥによりアクセスすることが可能なメモリ内にデータが受信されると、該ＣＰＵ／ＦＥが制御情報を読み出してデータのデスティネーションとなるポートを決定する。該ポートが決定されると、それが各I/0モジュール上に書き込まれて、該データを棄却すべきか維持すべきかが示される。したがって、このアーキテクチャは、ＣＰＵ／ＦＥメモリに関する幾分かの競合を軽減させ、これにより待ち時間が幾分か低減されるが、その代わりにメインシステムバスに多くの競合が生成される。これは、各I/0モジュールが他のI/0モジュールに関するデータをその必要性の有無に関わらず送信しなければならないからである。ポート及びデータ伝送速度が増大すると、ＣＰＵ／ＦＥモジュールのメモリスループットは、システム中の全ポートのデータ伝送速度よりも大きくなければならない。これは、図４及び図５に示す従来の２つの例と比較してメモリシステムのコスト及び複雑性を低減させるものであるが、あらゆるI/0モジュール上に一層複雑で高コストの「モジュール」相互接続バス及びメモリを必要とし、このため、システム全体のコスト及び複雑性が増大することになる。コスト及び複雑性を増大させる他の要因として、あらゆるI/0モジュールが、他のあらゆるI/0モジュールからのデータを受信することができるよう十分なメモリを有していなければならないことが挙げられる。一般に、I/0モジュール上のメモリスループットは、前述のように、そのポートのデータ伝送速度の２倍をサポートしていればよく、及びその待ち行列の管理をサポートしていればよい。このアーキテクチャでは、そのポートのデータ伝送速度の２倍及びその待ち行列の管理をサポートしていなければならず、更に、システム内の他のポートの全てのデータ伝送速度、及びこのデータに関する付加的な受信待ち行列の管理をサポートしていなければならない。ポート及びデータ伝送速度が増大すると、それに従って、あらゆるI/0モジュール及びＣＰＵ／ＦＥモジュール上のメモリシステムのスループットが増大しなければならず、このため、この種のアーキテクチャのスケーラビリティが制限されることになる。

【００３１】

各ポート毎に多数の待ち行列を維持することを必要とするＱＯＳを提供する場合、このアーキテクチャは、待ち行列にアクセスするための競合の増大に起因して待ち時間を増大させるものとなる。

【００３２】

マルチキャストサポートを提供する場合には、このアーキテクチャは、同一のパケットを各I/0モジュールに同時に送信することができるという点では従来の例よりも良好なものであるが、１回のアクセスでパケット全体を送信することができないため、それに従って

待ち時間が増大する。したがって、このアーキテクチャは、待ち時間の幾分かの内部的な低減を提供するものとなるが、これは、一層高いメモリスループット、１I/0モジュールあたりのコスト及び複雑性の増大という犠牲の下で実現されるものである。

【００３３】

更に、このシステムの場合も、パケット／セルがメモリ中に完全に書き込まれるまで制御情報及び待ち行列の処理を開始することができないため、ゼロ待ち時間の達成が不可能となる。

図７のタイプのネットワーク

図７に示すようなクロスバーを用いる場合には、セルは、典型的には入力で処理され、及び該セルがクロスバー内で効率的にスイッチングされることを可能にする内部ヘッダが与えられる。クロスバーは、該内部ヘッダを用いてセルをどの出力ポートにスイッチングすべきかを決定することになる。複数のセルが同一の出力を宛先とする場合には、クロスバー内の出力又は入力において更なるバッファリングが必要となる。殆どのクロスバーアーキテクチャは、セルのみと共に用いられる。これは、かかるアーキテクチャが、単一の出力ポートを宛先とする複数の入力ポートやマルチキャストを含む幾つかの要因に起因するブロック化(blocking)という問題を一般に有するという事実に起因する。パケットが使用される場合、該パケットのサイズは64byteから64,000byteまで変化する可能性があり、これらのブロック化の問題は大きな問題となり、一般に該アーキテクチャを使用不能にするものとなる。

【００３４】

入力ポートにおける初期のルックアップは、図４に関して説明した競合の問題を依然として有するものであり、セル全体が受信されるのを待った後にルックアップを実行しなければならず、この場合もシステムの待ち時間が増大する。セルがスイッチ内に入ると、待ち時間は、実施されるクロスバーのタイプによって決まるが、一般には、シリコンベースのクロスバーにおける多数のホップのトラバース(traversing)、又はメモリベースのクロスバーにおける共有メモリに関する競合から構成される。クロスバー内で内部的なブロック化が生じた場合には、更なる待ち時間が生じる可能性がある。

【００３５】

ＱＯＳを提供する場合には、典型的には入力ポート又は出力ポートの何れかに待ち行列が設けられる。該待ち行列は、あらゆるポートに必要なものではなく、競合及び維持すべき待ち行列の数が低減され、待ち時間もまた低減される。

【００３６】

マルチキャストサポートを提供する場合には、典型的にはクロスバー内でセルが複製され、その結果として、内部的に又は出力ポートでブロック化状況が発生し（及び待ち時間が増大し）、また入力ポートでバックプレッシャー(backpressure)が生じ、このため、入力ポートが更なるバッファ空間を提供することが必要となる。ポート及びデータ伝送速度が増大した場合、このアーキテクチャは、スケーリングすることができない。これは、マルチキャストにより、ブロック化の問題が悪化し、システムのコスト及び複雑性が一層増大するからである。

【００３７】

このシステムの場合も、パケット／セルがメモリ中に完全に書き込まれるまで制御情報及び待ち行列の処理を開始することができないため、ゼロ待ち時間の達成が不可能となる。

本発明の好適実施例

図８に例示する従来のものとは異なる本発明は、待ち時間を最小限にするようネットワーキングシステムを最適化し、実際に、データ伝送速度及びポート密度が増大された場合でさえ、ゼロ待ち時間を達成することが可能なものである。更に、本発明は、53byteのセル又は64～64Kbyteのパケットについてもこれを達成する。これは、パケット／セルがメモリに書き込まれている際に該パケット／セルから制御情報を抽出し、及びメモリへのデータの書き込み時にスイッチング、ルーティング、及び／又はフィルタリングの決定を行う送出エンジンＦＥに前記制御情報を提供することにより、達成される。

【００３８】
送出エンジンＦＥがそのタスクを終えた後、次いで、その結果が、待ち行列に対する出し入れを行うために待ち行列マネージャＱＭに与えられる。本発明によれば、これら全てが発生するのは、パケット／セルがメモリに完全に書き込まれる前であり、これにより、パケット／セルがメモリに完全に書き込まれた直後にデータの読み出しを開始させることが可能となる。実際に、パケットがメモリに完全に書き込まれる前に該パケットの読み出しを開始させることが可能であるが、これは、これまでメモリからの読み出し時に行うことはできなかった。不正なデータがメモリから読み出されることになるからである。パケット／セルの読み出しの開始が早すぎないことを保証するためには、図８に示すように出口ポートと入口ポートとの間の同期Ｓが必要となる。これは、送出エンジンＦＥが、最後のデータがメモリ中にあることを検出し、次いで意図される出口ポートについて待ち行列のアドレス情報を待ち行列マネージャＱＭに送る場合に達成される。これは、本発明と他の全てのネットワーキングアーキテクチャとの間の重要な相違点である。かかる他のアーキテクチャを用いる場合には、この競合条件は存在せず、したがって、既に指摘したように、それらでゼロ待ち時間を達成することは不可能である。

【００３９】
該競合条件を防止するために、出口ポートと入口ポートとの間の同期Ｓが送出エンジンの出力で実施される（送出エンジンは、パケット／セルがメモリに完全に書き込まれるまで、その出力に結果を保持し、次いで、該結果を一点鎖線のフローラインで示すように待ち行列マネージャに送ることができる）。

【００４０】
本発明では制御情報のために別個のパス又はデュアルパスが使用され、このため、各I/0モジュール（#1～#n）がそのポートのみについてデータを（該データが送信データであろうと受信データであろうと）処理することが可能となり、これにより、本発明のシステムは、今日の既存のシステムと比較して、必要となるロジックの複雑性がより低下し、一層単純で安価に実施することが可能なものとなる。前述のように、殆どのシステムは、システム内のあらゆるポートのデータ伝送速度の２倍よりも遙かに大きいデータ伝送速度をサポートするために、制御情報を格納するために使用されるメモリアーキテクチャを必要とする。一方、本発明は、そのI/0モジュール上のあらゆるポートのデータ伝送速度に対する要件を低減させるものとなる（これは著しい低減となる）。本発明はまた、高コストで複雑な解決策を必要とすることなく、システムのポート及びデータ伝送速度の増大を可能にするものである。

【００４１】
殆どの他のアーキテクチャでは、既述のように、パケット又はセルを格納するために使用される同一のメモリにアクセスするためにＦＥ及び待ち行列マネージャが必要となり、その結果、待ち時間が増大することになる。これは、ＦＥ及び待ち行列マネージャがメモリアクセスに関して各ポートと競合しなければならないからである。しかし、本発明を用いた場合には、ＦＥ及び待ち行列マネージャは、制御情報を処理するための別個のパスＰを有しており、これにより、それら２つのエンティティが最大限の性能で動作することが可能になり、メモリとの間のパケット／セルの伝送による干渉を受けることもなくなる。これは、実際に、ゼロ待ち時間を達成するために生じなければならないことである。

【００４２】
図９から分かるように、本発明のシステムの待ち時間は、制御情報及び待ち行列の処理がメモリへのデータ書き込みから独立して行われる場合には、ゼロへと低減することが可能である。図９において（この場合も、メモリに対する１回の読み出し又は書き込みについてのアクセスタイムはＭに等しく、１回のメモリアクセスについてのビット数はＷである）、図８に示す本発明のデュアルパス処理により以下の事象が生じ、該図９は、特に前述の図３のＡ-Ｅ動作と対照をなすような形式をとったものである。

【００４３】
Ａ．受信ポート#1からメモリへのデータの書き込み。１パケット又はセルを伝送するため

の時間は（(Ｂ×８)／Ｗ）×Ｍに等しい（Ｂは１パケット又はセルのバイト数）。パケットが大きくなると、該パケットをメモリへ書き込むための時間も長くなる。

【００４４】

Ｂ．受信ポート#2からメモリへのデータの書き込み。１パケット又はセルを伝送するための時間は（(Ｃ×８)／Ｗ）×Ｍに等しい（Ｃは１パケット又はセルのバイト数）。パケットが大きくなると、該パケットをメモリへ書き込むための時間も長くなる。

【００４５】

Ｃ．メモリへのパケット又はセルの書き込み中に該パケット又はセルから制御情報が抽出される。該制御情報の処理は直ちに開始される。この処理の結果は、待ち行列マネージャに与えられる。該制御情報は、パケット／セルのヘッダから抽出されるので、必要となる情報のみが抽出される。これは、典型的には、パケット又はセルについて4～10byteである。該制御情報の抽出に要する時間の量は（(Ｙ×８)／Ｗ）×Ｍとなる。ここで、Ｙは、制御情報がヘッダ内でまたがるバイト数（典型的には4～24byte）である。送出エンジンは他の如何なるデバイスとも競合する必要がないので、制御情報の処理の開始時における遅延は全く存在しない、ということが理解されよう。この結果は、受信したばかりのパケット／セルの待ち時間に影響を与えるものではない。

【００４６】

Ｄ．ポート#1から受信されたパケット／セルのバッファアドレスの適当な出口I/0モジュールへの送出。該I/0モジュール上で、待ち行列マネージャがバッファアドレスを適当な待ち行列に入れ、該待ち行列の最上部に現れた際に抽出することになる。この送出結果は典型的には4～10byteとなる。該結果を待ち行列マネージャに送るための時間は（(Ｚ×８)／Ｒ）×Ｓとなる。ここで、Ｚは送出結果の長さ、Ｒは該結果をＦＥから待ち行列マネージャに送るために使用されるバスの幅、及びＳは同じバスのクロック速度である。メモリとの間でのパケット／セルの伝送は送出結果の送信に干渉しないものであることが理解されよう。この結果は、受信したばかりのパケット／セルの待ち時間に影響を与えるものではない。

【００４７】

Ｅ．どの待ち行列が送信に利用可能なデータを有しているかを判定するための異なる待ち行列の読み出し。これは、ポート#1からのパケット／セルのバッファアドレスが送信可能な状態になるまで多数の待ち行列の読み出しを行うことからなる。読み出される各待ち行列エントリは典型的には8～12byteとなる。待ち行列を更新するための時間は(Ｆ＋１)(((Ｐ×８)／Ｒ)×Ｓ)となる（Ｆは最後にパケットが待ち行列から出される前に読み出された待ち行列の数）。待ち行列は分散され、読み出すべき待ち行列の数は、特定のI/0モジュール（システム全体ではない）をサポートするための待ち行列の数まで低減される。これは、異なる待ち行列をスキャンするために要する時間量を低減させ、したがってゼロ待ち時間の達成を支援するものとなる。メモリとの間でのパケット／セルの伝送は、待ち行列からのバッファアドレスの取り出しに干渉しないものであることが理解されよう。この結果は、受信したばかりのパケット／セルの待ち時間に影響を与えるものではない。

【００４８】

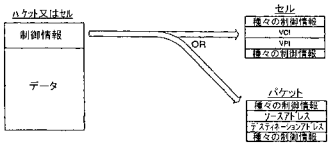
Ｆ．メモリから受信ポート#2へのデータの読み出し。１パケット又はセルを伝送するための時間は（(Ｂ×８)／Ｗ）×Ｍに等しい（Ｂは１パケット又はセルのバイト数）。パケットが大きくなると、該パケットをメモリから読み出すための時間も長くなる。

【００４９】

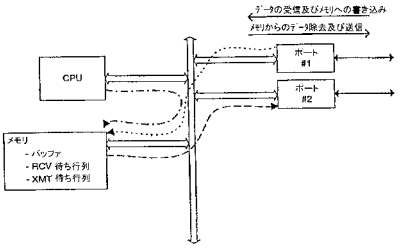
前述のように１ポートにつき多数の待ち行列を維持することを必要とするＱＯＳを提供する場合には、本アーキテクチャは、待ち行列マネージャをＮ個の独立したサブ待ち行列マネージャへと分割することを可能にする。この場合、各サブ待ち行列マネージャは、関連するI/0モジュールについて待ち行列を処理することを責務とするものとなる。これにより、システムのポート及びデータ伝送速度が増大された場合に拡張することが可能な単純で安価な実施態様が可能となり、これもまた、システムのゼロ待ち時間の達成を可能にするものとなる。

【００５０】
また、マルチキャストサポートを提供する場合には、本発明のアーキテクチャは、単純で安価な実施態様を用いることができるように送出及び待ち行列の決定を行うために必要となる最小限の量の情報を送るという点で最適な解決策を提供するものとなる。待ち行列マネージャが「サブ待ち行列マネージャ」から構成されるため、マルチキャスト情報を各サブ待ち行列マネージャへ同時に与えることができ、これにより、競合が排除され、ゼロ待ち時間が達成される。
【００５１】
したがって、本発明は、待ち時間を最小限にするための最適な解決策を提供し、該解決策は、ポート及びデータ伝送速度の増大に伴って拡張可能である一方、単純で安価な実施形態しか必要としないものとなる。
【００５２】
本発明の最終的な結果は、送出エンジンは、各パケット／セルの制御情報を処理する際に他のリソースと競合する必要がない、ということであり、実際に、送出エンジンは、そのI/0モジュール内のデータを処理するだけでよく、一層単純で複雑性の低いものとすることができる。待ち行列マネージャもまた、各パケット／セルの待ち行列の送受信を行う処理する際に他のリソースと競合する必要がなく、実際に、待ち行列マネージャは、そのI/0モジュール内のデータを処理するだけでよく、やはり一層単純で複雑性の低いものとすることが可能となる。
【００５３】
更に、本発明の場合には、データと制御情報との間に競合が存在せず、待ち行列が効率的に処理され、並びにマルチキャストサポートが提供される。本最終的な結果は、待ち時間を劇的に低減させるアーキテクチャとなる。本発明を前記の同時係属中の米国特許出願第581,467号の帯域幅最適化構造と組み合わせると、最適帯域幅及び最小待ち時間を有するネットワークが得られることになる。
【００５４】
当業者であれば更なる修正例を実施することが可能であり、かかる実施は、特許請求の範囲に係る本発明の思想及び範囲内に含まれるものである。
【図面の簡単な説明】
【図１】　従来の典型的なパケット／セル構造及び現行のネットワーキングシステムを示すブロック図である。
【図２】　例示としての典型的な従来のネットワーク中のシステムを示す同様のブロック図である。
【図３】　競合が如何にして待ち時間を生成するかを示す説明図である。
【図４】　典型的な従来の多数のI/0ポートを有する共有メモリシステムを示すブロック図である。
【図５】　「ヘッダ」キャッシュを有する修正された共有メモリシステムを示すブロック図である。
【図６】　従来の典型的な分散型メモリシステム及び現行のネットワーキングシステムを示す同様のブロック図である。
【図７】　ネットワーキングシステムで使用される典型的なクロスバーシステムを示すブロック図である。
【図８】　本発明の好適なパケット／セルデュアルパスデータ処理及び管理アーキテクチャを示すブロック図である。
【図９】　本発明により如何に待ち時間が低減されるかを示す説明図である。

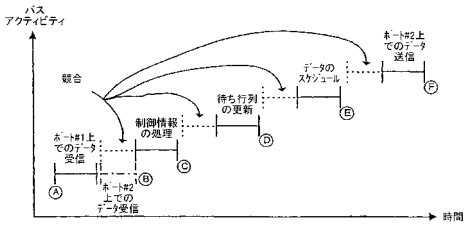
【図１】



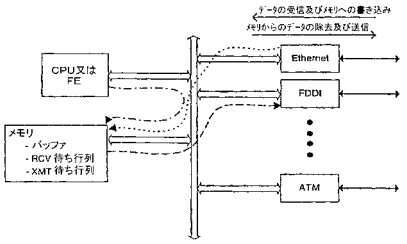

【図２】



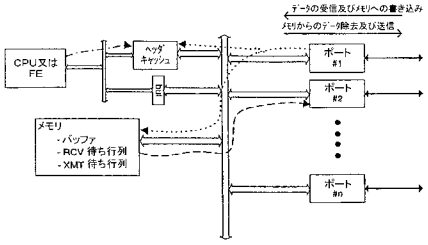

【図３】



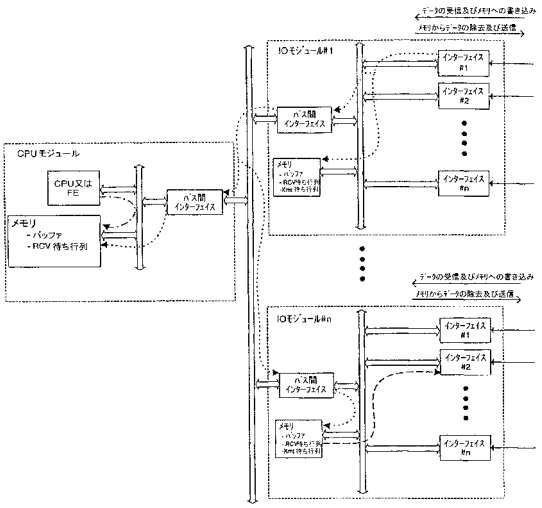

【図４】



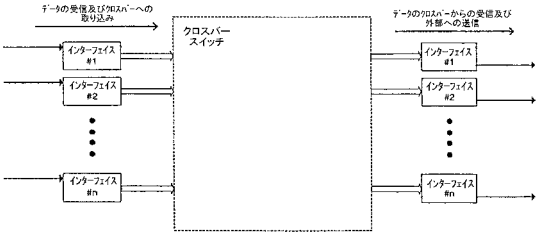

【図５】



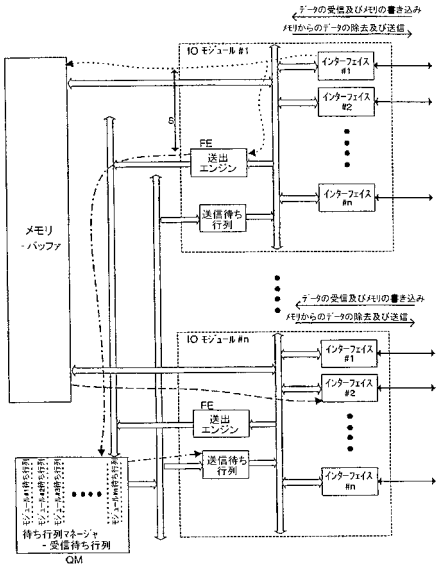

【図６】



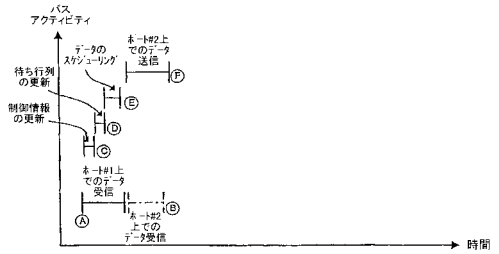
【図７】


データの受信及びクロスバーへの取り込み
クロスバースイッチ
データのクロスバーからの受信及び外部への送信
インターフェイス #1
インターフェイス #2
インターフェイス #n

【図８】


データの受信及びメモリの書き込み
メモリからのデータの除去及び送信
IO モジュール #1
IO モジュール #n
インターフェイス #1
インターフェイス #2
インターフェイス #n
FE
送出エンジン
送信待ち行列
メモリ
・バッファ
待ち行列マネージャ
・受信待ち行列
QM

【図９】


バス
アクティビティ
データのスケジューリング
ポート#2上でのデータ送信
待ち行列の更新
制御情報の更新
ポート#1上でのデータ受信
ポート#2上でのデータ受信
時間
A
B
C
D
E
F

フロントページの続き

(72)発明者　ライト，ティム
アメリカ合衆国マサチューセッツ州０１７０１，フレミンガム，オークス・ロード・７７
(72)発明者　マルコニ，ピーター
アメリカ合衆国マサチューセッツ州０１７０１，フランクリン，オーク・ツリー・レーン・５
(72)発明者　コンリン，リチャード
アメリカ合衆国マサチューセッツ州０２０３８，フランクリン，エイム・ストリート・３２
(72)発明者　オパルカ，ビグニュー
アメリカ合衆国マサチューセッツ州０１４５１，ハーバード，クアリー・レーン・２５

審査官　横山　佳弘

(56)参考文献　特開平０７－２７３８０１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 13/12
H04L 12/46