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

特許第5066702号
(P5066702)

(45)発行日　平成24年11月7日(2012.11.7)　(24)登録日　平成24年8月24日(2012.8.24)

(51)Int.Cl.		F I		
G06F 13/38	(2006.01)	G06F	13/38	350
G06F 12/00	(2006.01)	G06F	12/00	545M
G06F 13/10	(2006.01)	G06F	13/10	340B

請求項の数 35　(全 50 頁)

(21)出願番号	特願2002-531029 (P2002-531029)
(86)(22)出願日	平成13年9月24日 (2001.9.24)
(65)公表番号	特表2004-510252 (P2004-510252A)
(43)公表日	平成16年4月2日 (2004.4.2)
(86)国際出願番号	PCT/US2001/030150
(87)国際公開番号	WO2002/027519
(87)国際公開日	平成14年4月4日 (2002.4.4)
審査請求日	平成20年9月24日 (2008.9.24)
(31)優先権主張番号	09/675,700
(32)優先日	平成12年9月29日 (2000.9.29)
(33)優先権主張国	米国(US)
(31)優先権主張番号	09/675,484
(32)優先日	平成12年9月29日 (2000.9.29)
(33)優先権主張国	米国(US)

(73)特許権者　501082978
アラクリテック・インコーポレイテッド
アメリカ合衆国，カリフォルニア州・９５０１２，サン・ノゼ，イースト・ギッシュ・ロード・２３４
(74)代理人　100098062
弁理士　梅田　明彦
(72)発明者　スター，ダリル，ディー
アメリカ合衆国，カリフォルニア州・９５０３５，ミルピタス，フォルソム・コート・４４６
(72)発明者　フィルブリック，クライヴ，エム
アメリカ合衆国，カリフォルニア州・９５１２５，サン・ノゼ，ロイコット・ウェイ・１１７０

最終頁に続く

(54)【発明の名称】インテリジェントネットワークストレージインタフェースシステム及びデバイス

(57)【特許請求の範囲】

【請求項１】

ネットワークと記憶装置間で情報を転送するための装置であって、

ファイルシステムを動作させるＣＰＵと、ホストメモリバスにより前記ＣＰＵに接続されたホストメモリとを有するホストコンピュータと、

前記ホストコンピュータと前記ネットワークと前記記憶装置とに接続され、前記ネットワークと記憶装置間で通信されるデータを格納するのに適したインタフェースファイルキャッシュを含むインタフェースメモリを有するインタフェース装置とを備え、

前記インタフェースファイルキャッシュが前記ファイルシステムにより制御され、

前記ホストコンピュータが、通信制御ブロックを作成しかつ前記通信制御ブロックを前記インタフェース装置へ送るように構成され、

前記インタフェース装置が、前記通信制御ブロックに従って前記ネットワークと前記ファイルキャッシュ間で前記データを通信するように構成されていることを特徴とする装置。

【請求項２】

前記ホストコンピュータが、通信制御ブロックを作成しかつ前記通信制御ブロックを前記インタフェース装置へ送るように構成され、かつ

前記インタフェース装置が、前記通信制御ブロックに従って前記記憶装置を前記ファイルキャッシュ間で前記データを通信するように構成されていることを特徴とする請求項１に記載の装置。

【請求項３】
前記データが前記ホストコンピュータに入力しないことを特徴とする請求項１に記載の装置。
【請求項４】
前記データが、前記ネットワークと前記インタフェース装置間をファイル形式で通信され、かつ
前記データが前記インタフェース装置と前記記憶装置間をブロック形式で通信されることを特徴とする請求項１に記載の装置。
【請求項５】
前記データがトランスポートレイヤヘッダを含むヘッダに添付され、かつ前記インタフェース装置が前記トランスポートレイヤヘッダを処理するように構成されたメカニズムを有することを特徴とする請求項１に記載の装置。
【請求項６】
前記インタフェース装置が、前記ホストを有する第１パスを介して、または前記ホストを有しない第２パスを介して前記ネットワークと前記記憶装置間で前記データを転送するかどうかを選択するように構成されたメカニズムを有することを特徴とする請求項１に記載の装置。
【請求項７】
前記インタフェース装置が、前記記憶装置に接続されたＳＣＳＩコントローラを有することを特徴とする請求項１に記載の装置。
【請求項８】
前記インタフェース装置が、前記記憶装置に接続されたファイバチャネルコントローラを有することを特徴とする請求項１に記載の装置。
【請求項９】
前記インタフェース装置が、前記記憶装置に接続されたＲＡＩＤコントローラを有することを特徴とする請求項１に記載の装置。
【請求項１０】
前記インタフェース装置が、前記ネットワーク及び前記記憶装置の少なくとも一方に接続されたネットワークポートを有することを特徴とする請求項１に記載の装置。
【請求項１１】
前記ホストコンピュータ、前記ネットワーク及び第２記憶装置に接続された第２インタフェース装置を更に備え、
前記第２インタフェース装置が、前記ネットワークと前記第２記憶装置間で通信されるデータを格納するのに適した第２インタフェースファイルキャッシュを含む第２インタフェースメモリを有し、
前記第２インタフェースファイルキャッシュが前記ファイルシステムにより制御されることを特徴とする請求項１に記載の装置。
【請求項１２】
ネットワークと記憶装置間で情報を転送するための装置であって、
ホストメモリバスによりホストメモリに接続されたプロセッサを有し、前記ホストメモリが、ネットワーク接続を定義する通信制御ブロックを作成するように前記プロセッサにより動作可能なプロトコルスタックを有するホストコンピュータと、
前記ホストコンピュータに接続されかつ前記ネットワークと前記記憶装置間に結合されたインタフェース装置とを備え、前記インタフェース装置が、前記通信制御ブロックを格納するのに適したインタフェースメモリと、データを前記通信制御ブロックと関連付け、それにより前記データが前記ホストコンピュータに出会うことなく前記ネットワークと前記記憶装置間で通信されるように構成されたメカニズムとを有することを特徴とする装置。
【請求項１３】
前記ホストコンピュータがファイルシステムを有し、かつ前記インタフェースメモリが

前記データを格納するのに適したファイルキャッシュを有し、前記ファイルシステムが前記データの前記ファイルキャッシュへの格納を管理することを特徴とする請求項１２に記載の装置。

【請求項１４】

前記データが、トランスポートレイヤヘッダを含む少なくとも１つのパケットで前記ネットワーク上を移動し、かつ前記インタフェース装置が前記ヘッダを処理するための回路を有することを特徴とする請求項１２に記載の装置。

【請求項１５】

前記トランスポートレイヤヘッダがＴＣＰヘッダであることを特徴とする請求項１４に記載の装置。

【請求項１６】

前記トランスポートレイヤヘッダがＵＤＰヘッダであることを特徴とする請求項１４に記載の装置。

【請求項１７】

前記インタフェース装置が入力／出力バスにより前記ホストコンピュータに接続され、かつ前記通信コントロールブロックが前記入力／出力バス上で前記インタフェース装置と前記ホストコンピュータ間を移動することを特徴とする請求項１２に記載の装置。

【請求項１８】

前記インタフェース装置が更に、前記記憶装置に接続されたＳＣＳＩコントローラを有することを特徴とする請求項１２に記載の装置。

【請求項１９】

前記インタフェース装置が更に、前記記憶装置に接続されたファイバチャネルコントローラを有することを特徴とする請求項１２に記載の装置。

【請求項２０】

前記インタフェース装置が更に、前記記憶装置に接続されたＲＡＩＤコントローラを有することを特徴とする請求項１２に記載の装置。

【請求項２１】

前記インタフェース装置が前記記憶装置と、少なくとも１つのネットワークポートにより前記ネットワークに接続されていることを特徴とする請求項１２に記載の装置。

【請求項２２】

前記通信制御ブロックに対応する先のまたは後のデータが前記ホストコンピュータを通過することを特徴とする請求項１２に記載の装置。

【請求項２３】

前記ホストコンピュータが、前記インタフェース装置によってアクセス可能なＵＤＰソケットを指定するように構成され、かつ前記インタフェース装置が、前記ＵＤＰソケットに従って前記ネットワークと前記ファイルキャッシュ間で前記データを通信するように構成されていることを特徴とする請求項１２に記載の装置。

【請求項２４】

ネットワークと記憶装置間で情報を転送するための装置であって、

ファイルシステムを動作させるＣＰＵと、ホストバスにより前記ＣＰＵに接続されたホストメモリとを有し、前記ファイルシステムにより情報がファイルの階層構造として論理的に編成されるホストコンピュータと、

前記ホストコンピュータ、前記ネットワーク及び前記記憶装置に接続され、前記ファイルシステムの制御下で前記ネットワークと前記記憶装置間で通信されるデータを格納するのに適したインタフェースファイルキャッシュを含むインタフェースメモリを有し、前記ファイルシステムによって前記ファイルキャッシュ内の情報が前記ファイルの階層構造の一部として論理的に編成されるようにしたインタフェース装置とを備え、

前記ホストコンピュータが、前記インタフェース装置によりアクセス可能なＵＤＰソケットを指定するように構成され、かつ、

前記インタフェース装置が、前記ＵＤＰソケットに従って前記ネットワークと前記ファ

イルキャッシュ間で前記データを通信するように構成されていることを特徴とする装置。
【請求項２５】
前記ホストコンピュータが、前記インタフェース装置によりアクセス可能なアプリケーションレイヤヘッダを作成するように構成され、かつ前記インタフェース装置が前記アプリケーションレイヤヘッダを前記データにプリペンドするように構成されていることを特徴とする請求項２４に記載の装置。
【請求項２６】
前記ホストコンピュータが、前記インタフェース装置によりアクセス可能なリアルタイムトランスポートプロトコルヘッダを作成するように構成され、かつ前記インタフェース装置が前記リアルタイムトランスポートプロトコルヘッダを前記データにプリペンドするように構成されていることを特徴とする請求項２４に記載の装置。
【請求項２７】
前記データが関連するＵＤＰヘッダと共に格納され、かつ前記インタフェース装置が前記ＵＤＰヘッダを処理するように構成されたメカニズムを有することを特徴とする請求項２４に記載の装置。
【請求項２８】
前記データが前記インタフェース装置によりＵＤＰヘッダでプリペンドされてＵＤＰデータグラムを作成し、かつ前記インタフェース装置が前記データグラムを複数のフラグメントに分割するように構成されたメカニズムを有することを特徴とする請求項２４に記載の装置。
【請求項２９】
前記データが複数のフラグメントに配置され、かつ前記インタフェース装置が、ＵＤＰヘッダに対応する前記フラグメントを連結するように構成されたメカニズムを有することを特徴とする請求項２４に記載の装置。
【請求項３０】
前記データが前記ホストコンピュータに入力しないことを特徴とする請求項２４に記載の装置。
【請求項３１】
前記データが音声データを含むことを特徴とする請求項２４に記載の装置。
【請求項３２】
前記データが映像データを含むことを特徴とする請求項２４に記載の装置。
【請求項３３】
前記データがリアルタイム通信の一部であることを特徴とする請求項２４に記載の装置。
【請求項３４】
ネットワークと記憶装置間で情報を転送するための装置であって、
ファイルシステムを動作させるＣＰＵと、メモリバスにより前記ＣＰＵに接続されたメモリとを有するコンピュータと、
前記コンピュータ、前記ネットワーク及び前記記憶装置に接続され、前記ネットワークと記憶装置間で通信されるデータを格納するインタフェースファイルキャッシュを含むインタフェースメモリを有するインタフェース装置とを備え、
前記インタフェースファイルキャッシュが前記ファイルシステムにより制御され、
前記データが前記インタフェース装置と前記記憶装置間での前記データの転送の際にｉＳＣＳＩヘッダに添付され、
前記コンピュータが、通信制御ブロックを作成し、かつ前記通信制御ブロックを前記インタフェース装置へ送るように構成され、
前記インタフェース装置が、前記通信制御ブロックに従って前記ネットワークと前記ファイルキャッシュ間で前記データを通信するように構成されていることを特徴とする装置。
【請求項３５】

前記インタフェース装置がギガビットイーサネットネットワークにより前記記憶装置に接続されていることを特徴とする請求項３４に記載の装置。

【発明の詳細な説明】

【０００１】

【技術分野】

本発明は、ネットワーク通信及びストレージに関する。

【０００２】

【背景技術】

過去数１０年以上に亘って、ネットワークコンピューティングの利益及び進歩がコンピュータネットワークの著しい成長を促しており、これがより一層の進歩、成長及び利益に拍車をかけている。しかしながら、この成長と共に従来のネットワークデバイスを用いる上で混乱及び障害が起こっている。例えば、ネットワークに接続されたコンピュータのＣＰＵが、そのネットワーク通信を処理するのに使用する時間の部分が増加して、他の作業に使うことができる時間が少なくなっている場合がある。特に、ネットワークとディスクドライブのようなコンピュータの記憶装置との間でファイルデータを移動させることへの要求が加速度的に増えている。従来、このようなデータはネットワーク上での移動のためにパケットに分割され、各パケットは受信コンピュータのＣＰＵによって一度に１つのレイヤが処理される制御情報のレイヤにカプセル化される。ＣＰＵの速度が一定の割合で増加しているにも拘わらず、このファイル転送のようなネットワークメッセージのプロトコル処理は、市販の最速ＣＰＵの利用可能な処理能力の大部分を消費する場合がある。

【０００３】

このような状況は、その主な機能がネットワーク上でファイルデータを転送することによってその接続型ディスクまたはテープドライブ上でファイルを格納しかつ検索することであるネットワークファイルサーバにとって、より一層挑戦的なものになる場合がある。ネットワーク及びデータベースが成長するにつれて、そのようなサーバに格納される情報の量が爆発的に増加し、そのようなサーバ接続型記憶装置の限界を露呈している。ホストＣＰＵにより処理されるプロトコルの上述した問題に加えて、従来の小型コンピュータシステムインタフェース（ＳＣＳＩ）インタフェースのようなパラレルデータチャネルの限界が、ストレージの必要性が増加するにつれて明らかになっている。例えば、パラレルＳＣＳＩインタフェースは、サーバに接続することができるストレージデバイスの数及びストレージデバイスとサーバ間の距離を制限する。

【０００４】

Ｔｏｍ　Ｃｌａｒｋ著の書籍「Designing Storage Area Networks」（著作権１９９９年）に記載されるように、サーバ接続型のパラレルＳＣＳＩストレージデバイスの限界に対する１つの解決策は、ネットワークサーバの正面にある既存のローカルエリアネットワーク（ＬＡＮ）に別のファイルサーバを接続することを必要とする。このネットワーク接続型ストレージ（ＮＡＳ）によって、ネットワーク上の他のサーバ及びクライアントからＮＡＳファイルサーバへのアクセスが可能になるが、元のネットワークサーバに専用の記憶容量が増加するものではない。逆に、ＮＡＳは、元のネットワークサーバが様々なＮＡＳファイルサーバと通信する必要があることから、そのサーバが必要とするプロトコル処理が増える場合がある。更に、ＮＡＳファイルサーバのそれぞれが、次に、プロトコル処理の負担及びストレージインタフェースの限界を被ることになる。

【０００５】

ストレージエリアネットワーキング（ＳＡＮ）が、サーバの背後に接続されたストレージデバイスのネットワークでデイジーチェーンにより接続されたＳＣＳＩストレージデバイスを置き換えることによって、ネットワーク上でのファイル転送及び記憶に対する必要性の増加に対する別の解決策を提供している。イーサネットまたはファーストイーサネットのような従来のネットワーク規格の代わりに、ＳＡＮは、ファイバチャネル（ＦＣ）と呼ばれる新しいネットワーキング規格を使用する。しかしながら、それが比較的最近の導入であるため、多くの市販されているＦＣデバイスは互いに互換性がない。また、ＦＣネッ

トワークはサーバと記憶装置のようなネットワーク上の２点間の通信のためにバンド幅を専用にすることがあり、前記２点が通信していない時はバンド幅が無駄になる。

【０００６】

今や公知のＮＡＳ及びＳＡＮは、転送されかつ格納されるデータの形によって区別することができる。ＮＡＳデバイスは一般にデータファイルを別のファイルサーバまたはクライアントへ及びそこから転送するのに対し、ＳＡＮ上ではデバイスレベルのデータのブロックが転送される。このため、ＮＡＳデバイスは通常記憶のためにファイルとブロック間で変換するためのファイルシステムを有するが、ＳＡＮはそのようなファイルシステムを持たないストレージデバイスを備えることができる。

【０００７】

これに代えて、ＮＡＳファイルサーバは、イーサネットＳＡＮの一部分として、サーバに専用のイーサネットベースのネットワークに接続することができる。Ｍａｒｃ　Ｆａｒｌｅｙは、その書籍「Building Storage Networks」（著作権２０００年）において、イーサネット上でストレージプロトコルを実行することが可能であり、それによってファイバチャネルの非互換性の問題を解消できると述べている。しかしながら、ＳＡＮのようなネットワークトポロジを使用することによってサーバに接続されたストレージデバイスの数の増加によって、そのサーバが実行しなければならないプロトコル処理の量が増加する。上述したように、このようなプロトコルの処理は既に最新型のサーバを乱用している。

【０００８】

ファイル転送のようなネットワークメッセージの従来の処理の一例は、ネットワークデータの格納を遅くするいくつかの処理ステップを説明している。ネットワークインタフェースカード（ＮＩＣ）は一般に、ホストがネットワークにアクセスできるようにする媒体アクセスコントロール（ＭＡＣ）機能を提供することに加えて、ホストとネットワーク間の物理的接続を提供する。ホストに送られたネットワークメッセージのパケットがＮＩＣに到着すると、そのパケットのＭＡＣレイヤヘッダは処理されかつ前記パケットはＮＩＣにおいて巡回冗長検査（ＣＲＣ）を受ける。次に前記パケットは、周辺装置相互接続（ＰＣＩ）のような入力／出力（Ｉ／Ｏ）バスを通ってホストに送られかつホストメモリに格納される。次に、ＣＰＵが、プロトコルスタックからの命令を実行することによって、前記パケットのヘッダレイヤのそれぞれを順に処理する。これには、最初に前記パケットを格納するためにホストメモリバスを通る移動と、次に各ヘッダレイヤを順に処理するためにホストメモリバスを通る移動とが必要である。そのパケットの全ヘッダレイヤが処理された後、該パケットからのペイロードデータが、前記メッセージの他の同様に処理されたペイロードパケットと共にファイルキャッシュ内にグループ分けされる。前記データは、ディスク上に格納するためのファイルブロックとして前記ファイルシステムに従ってＣＰＵにより再アセンブリされる。全パケットが処理されかつメッセージがファイルキャッシュにファイルブロックとして再アセンブリされた後、前記ファイルは、それぞれに数個のペイロードパケットから構築することができるデータブロックにして、ホストメモリバス及びＩ／Ｏバス上をホストストレージへディスク上で長期間格納するために、一般にＩ／ＯバスにブリッジされたＳＣＳＩバスを介して送られる。

【０００９】

これに代えて、ＳＡＮ上でファイルを格納するために、ファイルキャッシュの再アセンブリされたファイルがブロックでホストメモリバス及びＩ／Ｏバス上を前記ＳＡＮ用に構成されたＩ／Ｏコントローラへ送られる。ＳＡＮがＦＣネットワークである場合には、ファイバチャネルプロトコル（ＦＣＰ）に従ってファイルブロックをＳＡＮ上のストレージデバイスに送ることができる特別のＦＣコントローラが設けられる。ファイルがＮＡＳデバイス上で格納されることになっている場合には、該ファイルはＮＡＳデバイスへ送りまたは再送することができ、これがパケットを上述したとほぼ同様に処理するが、ＮＡＳデバイスのＣＰＵ、プロトコルスタック及びファイルシステムを使用し、ＮＡＳデバイスの記憶装置に前記ファイルのブロックを格納する。

【００１０】

このように、ホストに接続されたＮＡＳまたはＳＡＮでの格納のためにネットワークから該ホストに送られるファイルは、一般に前記ファイルの各メッセージパケットについてＩ／Ｏバスを通る２つの移動が必要である。更に、各パケットのヘッダレイヤの制御情報は、一時的に格納され、一度に１つのレイヤが処理されかつ次にＩ／Ｏバスへ送り戻されるので、ホストメモリバスを繰り返し通過することになる。このようなファイルをクライアントからのリクエストに応答してＳＡＮ上のストレージから検索することはまた、従来ホストＣＰＵ及びファイルシステムによる相当量の処理を必要とする。

【００１１】

【発明の開示】

データ転送の制御をホストに残しつつ、ネットワークと記憶装置間のデータ転送を加速するためのハードウェア及び処理メカニズムを提供する、ローカルホストのためのインテリジェントネットワークインタフェースカード（ＩＮＩＣ）のようなインタフェース装置が開示されている。このインタフェース装置は、ネットワークパケットヘッダを処理するためのハードウェア回路を有し、ネットワークと記憶装置間でのデータ転送のためのホストにより設定された専用の高速パスを使用することができる。ホストＣＰＵ及びプロトコルスタックにより、高速パス上でのデータ転送のためのプロトコル処理が回避され、ネットワーク及びストレージサブシステムの多くの要求からホストバスのバンド幅を解放する。独立ディスクの冗長アレイ（ＲＡＩＤ）または複数のドライブからなる他の構成を有することができる記憶装置が、ＳＣＳＩのようなパラレルチャネルによって、またはイーサネットもしくはファイバチャネルのようなシリアルチャネルによってインタフェース装置に接続することができ、かつインタフェース装置はＰＣＩバスのようなＩ／Ｏバスによってローカルホストに接続することができる。追加の記憶装置を、ＳＣＳＩのようなパラレルインタフェースによりローカルホストに接続することができる。

【００１２】

ファイルキャッシュがホストをバイパスすることができるデータを格納するためにインタフェース装置上に設けられ、インタフェース装置のファイルキャッシュにおけるデータの編成が、ホスト上のファイルシステムによって制御される。この構成によって、リモートホストと記憶装置間のデータ転送が、データがインタフェース装置とＩ／Ｏバス上のローカルホストの間を通過することなく、インタフェース装置の高速パス上で処理することができる。また、従来の通信プロトコル処理と対照的に、高速パスデータのための制御信号がホストメモリバス上を繰り返し移動して一時的に格納され、かつ次にホストＣＰＵにより一度に１レイヤが処理されるということがない。従って、ホストは、ホスト制御型記憶装置上でのファイル読出しまたは書込みのためのデータトラフィックの大部分に関与することから解放することができる。

【００１３】

追加のインタフェース装置をＩ／Ｏバスを介してホストに接続することができ、追加の各インタフェース装置はホストファイルシステムにより制御されるファイルキャッシュを有し、かつ追加のネットワーク接続を提供しかつ／または追加の記憶装置に接続される。複数のインタフェース装置が単一のホストに接続されることによって、該ホストは複数のストレージネットワークを制御することができ、ホスト制御型ネットワークとの間のデータフローの大部分がホストプロトコル処理、Ｉ／Ｏバスを通る移動、ホストバスを通る移動、ホストメモリへの格納をバイパスする。ある実施例では、記憶装置がそのようなインタフェース装置にギガビットイーサネットネットワークにより接続することができ、前記欠陥なしでファイバチャネルの速度及びバンド幅を提供し、かつイーサネットベースのネットワークの大規模導入ベース及び互換性の利益が得られる。

【００１４】

【発明を実施するための最良の形態】

本発明によるネットワークデータ通信システムの概要が図１に示されている。ホストコンピュータ２０が、ローカルまたはワイドエリアネットワーク２５もしくはインターネット２８のようなネットワークに接続するための１つまたは複数のポートを有することができ

るインテリジェントネットワークインタフェースカード（ＩＮＩＣ）２２のようなインタフェース装置に接続されている。ホスト２０は、ホストバス３５によりホストメモリ３３に接続された中央処理装置（ＣＰＵ）３０のようなプロセッサを有し、図示しないオペレーティングシステムが、ファイルシステム２３を含む様々なタスク及び装置を監督するためにメモリ３３に存在する。同様にホストメモリ３３に格納されているのは、ネットワーク通信を処理するためのインストラクションのプロトコルスタック３８、及びＩＮＩＣ２２とプロトコルスタック３８間を通信するＩＮＩＣドライバ３９である。キャッシュマネージャ２６は、ファイルシステム２３及びＷｉｎｄｏｗｓ（登録商標）ＮＴまたは２０００の仮想メモリマネージャのような任意のメモリマネージャ２７の制御下で動作して、ファイルストリームと称されるファイル部分をホストファイルキャッシュ２４上に格納しかつ検索する。

【００１５】

ホスト２０は、ＰＣＩバスのようなＩ／Ｏバス４０によってＩＮＩＣ２２に接続され、これはホストＩ／Ｏブリッジ４２によりホストバス３５に結合されている。前記ＩＮＩＣは、ＩＮＩＣバス４８により相互に接続されたインタフェースプロセッサ４４及びメモリ４６を有する。ＩＮＩＣバス４８は、ＩＮＩＣ・Ｉ／Ｏブリッジ５０を有するＩ／Ｏバス４０に結合されている。同様にＩＮＩＣバス４８に接続されているのは、ネットワークメッセージの上位レイヤプロセッシングを行うハードウェアシーケンサ５２のセット即ち集合である。ＬＡＮ／ＷＡＮ２５及びインターネット２８への物理的接続は、従来の物理的レイヤハードウェアＰＨＹ５８によって提供される。各ＰＨＹ５８装置は、媒体アクセスコントロール（ＭＡＣ）６０の対応する装置に接続され、各ＭＡＣ装置によって、前記ＩＮＩＣと前記ネットワークの１つとの間に従来型のデータリンクレイヤ接続が提供される。

【００１６】

ディスクドライブまたはディスクドライブの集合と対応するコントローラのようなホスト記憶装置６６が、ＳＣＳＩアダプタのような従来のＩ／Ｏコントローラ６４によりＩ／Ｏバス４０と接続することができる。パラレルデータチャネル６２がコントローラ６４をホスト記憶装置６６に接続している。これに代えて、ホスト記憶装置６６は、独立ディスクの冗長アレイ（ＲＡＩＤ）とすることができ、かつＩ／Ｏコントローラ６４はＲＡＩＤコントローラとすることができる。ファイルシステム２３の命令下で動作するＩ／Ｏドライバ６７、例えばＳＣＳＩドライバモジュールがコントローラ６４と対話して、ホスト記憶装置６６へのデータの書込み及び読出しを行う。好ましくは、ホスト記憶装置６６が、ホストメモリ３３にキャッシュすることができる、ファイルシステム２３を含むホスト２０のオペレーティングシステムコードを有する。

【００１７】

ディスクドライブまたはディスクドライブ及び対応するコントローラの集合のようなＩＮＩＣ記憶装置７０が、マッチングインタフェースコントローラ、ＩＮＩＣ・Ｉ／Ｏコントローラ７２を介してＩＮＩＣバス４８に接続され、これが次にパラレルデータチャネル７５により前記ＩＮＩＣ記憶装置に接続されている。ＩＮＩＣ・Ｉ／Ｏコントローラ７２は、ＳＣＳＩコントローラとすることができ、これはパラレルデータチャネル７５によりＩＮＩＣ記憶装置７０に接続される。これに代えて、ＩＮＩＣ記憶装置７０はＲＡＩＤシステムとすることができ、かつＩ／Ｏコントローラ７２をＲＡＩＤコントローラとし、多数のまたは分岐したデータチャネル７５とすることができる。同様に、Ｉ／Ｏコントローラ７２は、ＩＮＩＣ記憶装置７０のためのＲＡＩＤコントローラに接続されたＳＣＳＩコントローラとすることができる。別の実施例では、ＩＮＩＣ記憶装置７０は、ファイバチャネル（ＦＣ）ネットワーク７５に接続され、かつＩ／Ｏコントローラ７２はＦＣコントローラである。ＩＮＩＣ・Ｉ／Ｏコントローラ７２はＩＮＩＣバス４８に接続されたように図示されているが、Ｉ／Ｏコントローラ７２は、これに代えてＩ／Ｏバス４０に接続することができる。ＩＮＩＣ記憶装置７０は、任意により、そこからオペレーティングシステムの核がロードされる、ホスト２０のルートディスクを有することができる。ＩＮＩＣメモリ４６は、ＬＡＮ／ＷＮ２５のようなネットワークから受信しまたはそれに送信される

パケットを一時的に格納するためのフレームバッファ７７を有する。また、ＩＮＩＣメモリ４６は、ＩＮＩＣ記憶装置７０に格納しまたはそこから検索するデータを一時的に格納するために、インタフェースファイルキャッシュ、ＩＮＩＣファイルキャッシュ８０を有する。ＩＮＩＣメモリ４６が図１においてそれを簡単にするために１個のブロックで記載されているが、メモリ４６はＩＮＩＣ２２の様々な位置に配置された別個の装置で形成することができ、かつダイナミックランダムアクセスメモリ（ＤＲＡＭ）、スタティックランダムアクセスメモリ（ＳＲＡＭ）、リードオンリメモリ（ＲＯＭ）及び他の形態のメモリで構成することができる。

【００１８】

ファイルシステム２３は、記憶装置６６、７０及びファイルキャッシュ２４、８０の情報の編成に関する一般的な知識を有する高レベルのソフトウェアエンティティであり、ストレージアーキテクチュアの特性及び性能を実行するアルゴリズムを提供する。ファイルシステム２３は、ファイルの階層構造として、記憶装置６６、７０及び各ファイルキャッシュ２４、８０に格納されている情報を論理的に編成するが、このような論理ファイルは、記憶装置６６または７０の異なるディスクの本質的に異なるブロックに物理的に配置できるものである。また、ファイルシステム２３は、記憶装置６６、７０及びファイルキャッシュ２４、８０上でのファイルデータの格納及び検索を管理する。前記ファイルシステム下でホスト２０上で動作するＩ／Ｏドライバ６７ソフトウェアが、各記憶装置６６、７０のコントローラ６４、７２と対話して、データのブロックを操作し、即ちそれらの記憶装置からデータブロックを読み出しまたはそれらに書き込む。ホストファイルキャッシュ２４及びＩＮＩＣファイルキャッシュ８０によって、記憶装置６６、７０から読み出されまたはそれらに書き込まれるデータの記憶空間が提供され、前記データは記憶装置６６、７０の物理的ブロック形式とアプリケーションについて使用される論理的ファイル形式との間でファイルシステム２３によりマッピングされる。ファイルに関連しかつホストファイルキャッシュ２４及びＩＮＩＣファイルキャッシュ８０に格納されるバイトの線形ストリームをファイルストリームと称する。ホストファイルキャッシュ２４及びＩＮＩＣファイルキャッシュ８０はそれぞれ、その対応するキャッシュに保持されるファイルストリームをリストするインデックスを有する。

【００１９】

ファイルシステム２３は、記憶装置６６、７０上のファイルブロックのアドレスを決定するのに使用することができるメタデータを有し、最近アクセスされたファイルブロックのアドレスへのポインタがメタデータキャッシュにキャッシュされる。ファイルブロックへのアクセスが、例えばＬＡＮ／ＷＡＮ２５上のリモートホストにより要求されると、ホストファイルキャッシュ２４及びＩＮＩＣファイルキャッシュ８０のインデックスが最初に参照されて、前記ブロックに対応するファイルストリームがそれらのキャッシュに格納されているかどうかを見る。前記ファイルストリームがファイルキャッシュ２４または８０に見当たらない場合には、そのブロックへのリクエストが、前記メタデータにより示された適当な記憶装置のアドレスへ送られる。１つまたは複数の従来のキャッシングアルゴリズムをファイルキャッシュ２４及び８０のキャッシュマネージャ２６が用いて、前記キャッシュが空で新しいデータをキャッシュすべき場合にどのデータを捨てるべきかを選択する。ＩＮＩＣファイルキャッシュ８０上でのファイルストリームのキャッシングによって、ＩＮＩＣ記憶装置７０に格納されたファイルブロックのＩ／Ｏバス４０及びデータチャネル７５双方におけるトラフィックが大幅に減少する。

【００２０】

ホスト２０に送られたネットワークパケットがＩＮＩＣ２２に到着すると、そのパケットのヘッダがシーケンサ５２により処理されて、前記パケットを有効にしかつ該パケットのサマリまたは記述子が作成され、該サマリは前記パケットにプリペンドされかつフレームバッファ７７に格納され、前記パケットへのポインタが９に記憶される。前記サマリは、前記パケットヘッダのプロトコルタイプ及びチェックサムの結果を表す状態語である。この状態語に含まれるものは、前記フレームが高速パスデータフローの対象であるかどうか

を示すものである。従来の方法と異なり、トランスポート及びセッションレイヤの情報を含むプロトコル情報を有する上位レイヤヘッダが、シーケンサ５２のハードウェア論理により処理されて、サマリが作成される。前記シーケンサの専用論理回路によって、パケットヘッダを、該パケットがネットワークから到着するのと同様に高速で仮想的に処理することが可能になる。

【００２１】

次に、前記ＩＮＩＣは、ＣＰＵ３０実行プロトコルスタック３８によるヘッダの「低速パス」プロセッシングのために前記パケットをホストメモリ３３に送るか、または「高速パス」に従って前記パケットデータを直接ＩＮＩＣファイルキャッシュ８０またはホストファイルキャッシュ２４のいずれかに送るかを選択する。前記高速パスは、シーケンシャルでエラーのないメッセージ毎に複数のパケットを有するデータトラフィックの相当大部分について選択することができ、データを繰り返しコピーしたり、ホストメモリバス３５を繰り返し往来するような、前記ＣＰＵによる各パケットの時間のかかるプロトコルプロセッシングが回避される。パケットがＩＮＩＣファイルキャッシュ８０内に直接移される高速パス状況の場合、ホストバス３５とＩ／Ｏバス４０とにおける追加の往来が同様に回避される。低速パスプロセッシングは、ＩＮＩＣ２２の高速パスによって従来は転送されないあらゆるパケットをホスト２０により従来通り処理できるようにする。

【００２２】

ホスト２０における高速パス能力を与えるためにリモートホストとの接続が最初に設定され、それにはハンドシェーク、認証及び他の接続初期化の手順を含むことができる。通信制御ブロック（ＣＣＢ）が、ＴＣＰ／ＩＰまたはＳＰＸ／ＩＰＸプロトコルにより代表されるような接続ベースのメッセージのための接続初期設定手順の際にプロトコルスタック３８により作成される。前記ＣＣＢは、ソース及びデスティネーションアドレス並びにポートのような接続情報を有する。ＴＣＰ接続の場合、ＣＣＢは、ソース及びデスティネーション媒体アクセスコントロール（ＭＡＣ）アドレス、ソース及びデスティネーションＩＰアドレス、ソース及びデスティネーションＴＣＰポート、並びにスライディングウィンドウプロトコルのための送信受信ウィンドウ及びタイマのようなＴＣＰ変数からなる。接続が設定された後、前記ＣＣＢはＩＮＩＣドライバ３９により前記ホストからＩＮＩＣメモリ４６へ、そのメモリ４６内にコマンドレジスタを書き込むことによって送られ、そこでは他のＣＣＢと共にＣＣＢキャッシュ７４に格納することができる。また、前記ＩＮＩＣは、前記ＣＣＢをパケットサマリと加速して突き合わせるためにキャッシュしたＣＣＢに対応するハッシュテーブルを作成する。

【００２３】

ファイル書込みのような前記ＣＣＢに対応するメッセージを前記ＩＮＯＣが受け取ると、該メッセージの最初のパケットのヘッダ部分がホスト２０に送られて、ＣＰＵ及びプロトコルスタック３８により処理される。この前記ホストに送られたヘッダ部分は、前記パケットの一定のオフセットで始まることが知られている、前記メッセージのセッションレイヤヘッダを有し、かつ任意により前記パケットからいくつかのデータを含んでいる。プロトコルスタック３８のセッションレイヤによる前記セッションレイヤヘッダのプロセッシングによって、前記ファイルに属するものとして前記データが認識されかつ前記メッセージのサイズが表示され、これらが前記ファイルシステムにより使用されて、前記メッセージデータをホストファイルキャッシュ２４またはＩＮＩＣファイルキャッシュ８０にキャッシュするかどうか、及び選択されたファイルキャッシュ内の前記データのデスティネーションを確保するかどうかが決定される。前記ホストに送られたヘッダ部分に何らかのデータが含まれていた場合には、それは前記デスティネーションに格納される。選択された前記ファイルキャッシュの前記デスティネーションのバッファアドレスのリストがＩＮＩＣ２２に送られて、前記ＣＣＢにまたはそれと共に格納される。また、前記ＣＣＢは、メッセージの長さ及び処理されたパケットの数並びに順序のような、前記メッセージに関する状態情報を保持し、どのユーザが関与しているか及び転送情報毎の記憶空間を含む、各プロトコルレイヤに関するプロトコル及びステータス情報を提供する。

【００２４】
前記ＣＣＢが一旦デスティネーションを示すと、該ＣＣＢに対応するパケットの高速パスプロセッシングが可能になる。次に受け取ったパケットをシーケンサ５２により上述したようにプロセッシングしてパケットサマリを生成した後、前記パケットサマリのハッシュが前記ハッシュテーブルと比較され、かつ必要な場合には前記ＣＣＢキャッシュ７４に格納されたＣＣＢと比較され、高速パス接続が設定されているメッセージに前記パッケージが属するかどうかを決定する。パケットサマリがＣＣＢに整合すると、他の例外的な条件が存在しないと仮定して、ネットワークまたはトランスポートレイヤヘッダ無しで、前記パケットのデータがダイレクトメモリアクセス（ＤＭＡ）装置６８により、前記ＣＣＢにより示されたファイルキャッシュ８０またはファイルキャッシュ２４のデスティネーションに送られる。

【００２５】
前記メッセージからの全てのデータがファイルストリームとしてＩＮＩＣファイルキャッシュ８０またはホストファイルキャッシュ２４内にキャッシュされた後のいくつかの時点において、データのファイルストリームが、ファイルシステム２３の制御下にあるＤＭＡ装置６８により、そのファイルキャッシュからＩＮＩＣ記憶装置７０またはホスト記憶装置６６に、前記ファイルシステムの制御下で送られる。通例、ホストファイルキャッシュ２４内にキャッシュされたファイルストリームは、ＩＮＩＣ記憶装置６６上に格納されるのに対し、ＩＮＩＣファイルキャッシュ８０内にキャッシュされたファイルストリームは、ＩＮＩＣ記憶装置７０上に格納されるが、このような構成は必ずしも必要でない。ファイル転送に関する後のリクエストは、そのリクエストが同一のソース及びデスティネーションＩＰアドレス及びポートを含んでいると仮定して、同じＣＣＢにより取り扱うことができ、書込みリクエストの最初のパケットは前記ホストＣＰＵにより処理されて、前記メッセージを格納するためのホストファイルキャッシュ２４またはＩＮＩＣファイルキャッシュ８０内の位置を決定する。また、前記ファイルキャッシュをバイパスしてリモートホストから受け取ったメッセージからのデータを格納するためのデスティネーションとしてＩＮＩＣ記憶装置７０またはホスト記憶装置６６上の位置を予定するように前記ファイルシステムを構成することが可能である。

【００２６】
本発明の基本的な理解を促すための概要が図２に記載されており、同図では、各パスについての情報についての基本タイプを示すことによって、図１のネットワークデータ記憶システムのための情報の流れに関する主要なパスを分離している。図２は、細い矢印を有する基本的に制御情報からなる情報のフローパス、太く白い矢印を有する基本的にデータからなる情報のフローパス、及び太く黒い矢印を有する制御情報及びデータ双方からなる情報のフローパスを示している。ホスト２０が基本的に制御情報の流れに関係しているのに対して、ＩＮＩＣ記憶装置７０は基本的にデータ転送に関連していることに注意する。

【００２７】
ＬＡＮ／ＷＡＮ２５のようなネットワークとＩＮＩＣ２２間の情報の流れは制御情報及びデータを含むことができ、従って太く黒い矢印８５で示されている。ＬＡＮ／ＷＡＮ２５のようなネットワークとＩＮＩＣ２２間の情報の流れ８１の例は、ファイルの読出しまたは書込みに加えて、接続初期設定の対話及び肯定応答のような、制御情報に包含されるようなファイルデータを含むパケットとして送られる制御情報が含まれる。シーケンサ５２がファイル書込みからの制御情報を処理し、かつデータ及び制御情報をＩＮＩＣフレームバッファ７７との間でやりとりするが、従ってそれららの転送は太く黒い矢印８８で表されている。フレームバッファ７７に格納されるデータに関する制御情報は、細い矢印９０で示すようにプロセッサ４４により処理され、かつネットワーク接続初期設定パケット及びセッションレイヤヘッダのような制御情報は、太い矢印９２で示すように、プロトコルスタック３８に送られる。前記ホストにより接続が設定されると、その接続に関する、ＣＣＢのような制御情報を、細い矢印９４で示すようにホストプロトコルスタック３８とＩＮＩＣメモリ４６との間で送ることができる。ＩＮＩＣ記憶装置７０から読み出されまた

は書き込まれるデータの一時的な格納が、太く白い矢印９６、９８で示されるように、ＩＮＩＣファイルキャッシュ８０及びフレームバッファ７７により行われる。ＩＮＩＣファイルキャッシュ８０に格納されている全てのファイルストリームの制御及び知識が、細い矢印９１で示すように、ファイルシステム２３によって提供される。ホスト記憶装置６６がネットワークアクセス可能なデータを格納しない実施例では、ファイルシステムの情報が、矢印８１で示すように、ホストファイルキャッシュ２４とホスト記憶装置６６との間で送られる。同図に記載されていない別の実施例では、ホスト記憶装置を備えておらず、または基本的にネットワークファイル転送のためにホスト記憶装置及びホストファイルキャッシュを用いることができる。

【００２８】

図２からわかるように、ネットワークファイル読出しまたは書込みのデータは基本的にＩＮＩＣ２２を通過しかつホスト２０を避けており、制御情報が基本的に前記ホストとＩＮＩＣ間で送られる。このネットワークと記憶装置間のファイル転送に関する制御情報の分離によって、前記ホストは、前記ネットワークと記憶装置との間で前記ＩＮＩＣを流れるファイル転送を管理することが可能になり、他方前記ＩＮＩＣがそれらデータ転送のための高速パスを提供し、それによってデータ処理能力が加速される。前記ＩＮＩＣのデータ高速パスにより得られる処理能力の増加によって、ホスト及びＩＮＩＣは、ファイルサーバとして機能することに加えて、例えば映像のような高バンド幅アプリケーションのデータベースサーバとして機能することができる。

【００２９】

図３は、ネットワークから受け取ったメッセージを格納するために図１のシステムが実行するいくつかのステップを示している。ＬＡＮ／ＷＡＮ２５のようなネットワークから送られたパケットは、最初にＰＨＹ装置５８によりＩＮＩＣ２２において受け取られ１００、かつＭＡＣ装置６０が前記パケットがホスト２０にアドレス指定されていることを確認するようにリンクレイヤプロセッシングを実行する。次に、ネットワーク、前記パケットのトランスポート及び任意によりセッションレイヤヘッダがシーケンサ５２により処理され１０２、それにより前記パケットが有効になりかつそれらヘッダのサマリが作成される。次に、前記サマリをパケットに追加して、フレームバッファ７７の１つに格納する。次にプロセッサ４４が、前記パケットサマリを検査することにより、前記パケットが高速パスプロセッシングの対象であるかどうかを決定する１０６。パケットが高速パスの対象であるかどうかは、単にサマリ内に示されているパケットのプロトコルによって決定することができる。パケットが高速パス対象でない場合には、そのパケットはプロトコルスタック３８からのＣＰＵ３０実行命令によって前記パケットのヘッダを処理するために、Ｉ／Ｏバス４０を介してホストメモリ３３に送られる１０８。

【００３０】

前記パケットは高速パス対象である場合には、前記パケットサマリが、それぞれにＣＣＢとして表される、前記カードによって取り扱われる高速パス接続の集合と前記サマリをＣＣＢハッシュ及びＣＣＢキャッシュと突き合わせることによって比較する。前記サマリが前記ＩＮＩＣメモリに保持されているＣＣＢと整合しない場合には、前記パケットは、プロトコルスタックからのＣＰＵ実行命令により前記パケットのヘッダを処理するためにホストメモリに送られる１１２。パケットが接続初期設定対話の一部分である場合には、前記パケットを用いて前記メッセージについてＣＣＢを作成することができる１１５。前記パケットサマリがむしろ前記ＩＮＩＣメモリに保持されているＣＣＢに整合する場合には、前記プロセッサは、例えば断片化されたまたは壊れたパケットを含む例外状態を検討し、そのような例外状態が見つかった場合には、プロトコルプロセッシングのためにＣＣＢ及び前記パケットをホストプロトコルスタック３８にフラッシュする。パケットサマリはＣＣＢに整合しているが該パケットのデスティネーションがＣＣＢで示されていない場合には、前記パケットのセッションレイヤヘッダをホストプロトコルスタック３８に送って、前記ファイルシステムに従って、ホストファイルキャッシュまたはＩＮＩＣファイルキャッシュ内のデスティネーションを決定し１２２、そのデスティネーションのキャッシュ

アドレスのリストをＣＣＢと共にＩＮＩＣに格納する。また、前記ＩＮＩＣはＣＣＢをホスト１１６にフラッシュさせかつパケットをスタックによるプロセッシングのためにホストに送らせることになる例外状態について前記パケットサマリを検査する１１４。

【００３１】

パケットサマリがＣＣＢに整合しかつ該パケットのデスティネーションが該ＣＣＢと共に格納され、かつ例外状態が存在しない場合には、前記パケットからのデータがＤＭＡにより、ＣＣＢにより示されたホストファイルキャッシュまたはＩＮＩＣファイルキャッシュのデスティネーションに送られる１２５。この場合におけるメッセージパケットはホストプロトコルプロセッシングスタックによるヘッダのプロセッシングをバイパスし、高速パスデータ転送を提供する。パケットのデータが高速パスを介してＩＮＩＣファイルキャッシュ及びＩＮＩＣストレージに送られているような場合には、前記パケットはホストによるプロトコルプロセッシングを避けるだけでなく、前記Ｉ／Ｏバスまたはホストメモリバスを横切らないので、従来のネットワークストレージと比較して大幅に時間を節約するＣＰＵプロセッシング及びバストラフィックが得られる。

【００３２】

【発明の実施の形態】

図４は、ＬＡＮ／ＷＡＮ２５のようなネットワークからのリクエスト２００に応答して、ホスト記憶装置６６またはＩＮＩＣ記憶装置７０からファイルまたはファイルの一部分を検索するために図１のシステムによって実行されるいくつかのステップを示している。まず、リクエストパケットをプロトコルスタックにより処理し、それにより前記リクエストをファイルシステムに送る。前記ファイルシステムは前記リクエストに示されたファイルを、該ファイルに対応するファイルストリームがＩＮＩＣファイルキャッシュまたはホストファイルキャッシュにキャッシュされているかどうかを決定すること、及び前記ファイルストリームが前記キャッシュのいずれかにも配置されていない場合には前記ファイルに対応するブロックが前記ホスト記憶装置またはＩＮＩＣ記憶装置に格納されているかどうかを決定することを含めて、位置付けする。ファイルストリームが前記キャッシュのいずれにも配置されていないと仮定した場合、前記ファイルブロックはホストファイルキャッシュに読み取られ２０４、またはＩＮＩＣファイルキャッシュに読み取られる２０６。大抵の場合には、前記ホスト記憶装置に格納されたファイルブロックはホストファイルキャッシュ内に読み取られ、かつＩＮＩＣ記憶装置に格納されたファイルブロックはＩＮＩＣファイルに読み取られることになるが、このマッピングは必然ではない。例えば、ホスト記憶装置に格納されたファイルブロックをＩＮＩＣファイルキャッシュ内に読み取り、それによってホストメモリバス上のトラフィックを減少させることが好ましい場合がある。

【００３３】

ファイルブロックがホストファイルキャッシュにキャッシュされていない場合には、前記ホストは前記リクエストに対応するＣＣＢが前記ＩＮＩＣにより保持されているかどうかを注意することによって、ファイルを高速パスプロセッシングにより送るかどうかを決定する２１０。前記ホストが高速パスを使うのではなく、低速パスによって前記ファイルをホストから送ることを選択した場合、前記ホストはプロトコルスタックを実行して、ホストファイルキャッシュに保持されているデータについてヘッダを作成し、かつ次に前記ヘッダ及びチェックサムをデータに追加して、従来と同様にＩＮＩＣによりネットワーク上を送信するためにネットワークフレームを作成する２１２。次に、ＩＮＩＣはＤＭＡを用いてホストから前記フレームを入手し２１４、かつ次にＩＮＩＣは前記フレームをネットワーク上に送る２０８。そうしないで前記ファイルを高速パスで送る場合には、ＩＮＩＣプロセッサはＣＣＢを用いてヘッダ及びチェックサムを作成し、ホストファイルキャッシュからのデータのフレームサイズのセグメントをＤＭＡし、かつ次に前記ヘッダ及びチェックサムを前記データセグメントにプリペンドしてネットワークフレームを作成し２１８、前記ホストをプロトコルプロセッシングから解放する。

【００３４】

同様にして、ファイルブロックがＩＮＩＣファイルキャッシュ内にキャッシュされた場合

には、前記ホストがＣＣＢがＩＮＩＣにより保持されているかどうかを注意することによって、高速パスプロセッシングにより前記ファイルを送るかどうかを決定する２２０。前記ホストが高速パスを使用しないことを選択した場合２２２、ホストＣＰＵは、ファイルブロックデータについてヘッダ及びチェックサムを準備し、前記ヘッダをホストメモリに格納する。次に前記ホストは、ＩＮＩＣに指示して、ホストメモリからのヘッダをＩＮＩＣメモリ内のデータにプリペンドすることによってネットワークフレームをアセンブリし、次にＩＮＩＣによりネットワークを介して送られるメッセージフレームを作成する。この高速パスを使用しない場合でさえ、データはＩ／Ｏバスを通ってホストに送られたりＩＮＩＣから戻されることが無く、Ｉ／Ｏバスまたはネットワークによりホストに接続された記憶装置上に配置されたファイルブロックの送信という従来方法と比較して、Ｉ／Ｏトラフィックが少なくなる。その代わりに高速パスが選択された場合２２５、ＩＮＩＣプロセッサはＣＣＢに対応するヘッダ及びチェックサムを作成し、かつ前記ヘッダ及びチェックサムをＩＮＩＣファイルキャッシュからのデータセグメントにプリペンドしてネットワークフレームを作成し、これは次にＩＮＩＣによってネットワーク上を送信される。この高速パスの場合には、前記ホストが、Ｉ／Ｏバストラフィックから解放されることに加えて、プロトコルプロセッシング及びホストメモリバストラフィックから解放される。

【００３５】

図５は、ホスト２０がＩ／Ｏバス４０を介して、第１ＩＮＩＣ２２に加えていくつかのＩ／Ｏ・ＩＮＩＣに接続されたネットワーク記憶システムを示している。本実施例の各ＩＮＩＣは少なくとも１つのネットワークに接続されかつ少なくとも１つの記憶装置が接続されている。かくして第２ＩＮＩＣ３０３はＩ／Ｏバス４０に接続され、第２ネットワーク３０５への前記ホストのインタフェースを提供する。第１ＩＮＩＣ２２は、上述したように実際はいくつかのネットワークに接続するためにいくつかのネットワークポートを持つことができ、同様に第２ＩＮＩＣ３０３も１つのネットワーク３０５以外のネットワークに接続することができるが、この図面では簡単化するために、各ＩＮＩＣについて１つのネットワーク接続のみが示されている。第２ＩＮＩＣ３０３は、第２ＩＮＩＣ記憶装置３０８に接続されたＳＣＳＩアダプタのようなＩ／Ｏコントローラを有する。これに代えて、第２ＩＮＩＣ記憶装置３０８はＲＡＩＤシステムとすることができ、かつ第２ＩＮＩＣ３０３はＲＡＩＤコントローラを有しまたはそれと接続することができる。別の実施例では、第２ＩＮＩＣ３０３が、ＦＣネットワークループまたはＦＣアダプタ及びネットワークラインによって第２ＩＮＩＣ記憶装置３０８に接続されたＦＣコントローラを有する。第Ｎ・ＩＮＩＣ３１０で示されるように、Ｎ個のＩＮＩＣをホスト２０にＩ／Ｏバスを介して接続することができる。第Ｎ・ＩＮＩＣ３１０は、第Ｎネットワーク３１３及び第Ｎ記憶装置３１５にそれぞれ接続されたネットワークインタフェース及びストレージインタフェースを提供する回路及び制御インストラクションを有する。第Ｎ・ＩＮＩＣ３１０はいくつかのネットワークに接続するためにいくつかのネットワークポートを持つことができ、かつ第２の第Ｎ・ＩＮＩＣ３１０が同様に２以上のネットワークに接続することができる。第Ｎ・ＩＮＩＣ３１０のストレージインタフェース回路及び制御インストラクションは、例えば、ＳＣＳＩケーブルによって第Ｎ・ＩＮＩＣ記憶装置３１５に接続されたＳＣＳＩコントローラを含むことができる。これに代えて、第Ｎ・ＩＮＩＣ記憶装置３１５はＲＡＩＤシステムとすることができ、かつ第Ｎ・ＩＮＩＣ３１５はＲＡＩＤコントローラを有しまたはそれに接続することができる。さらに別の実施例では、第Ｎ・ＩＮＩＣ３１０が、ＦＣアダプタ及びＦＣネットワークラインによって第Ｎ・ＩＮＩＣ記憶装置３１５に接続されたＦＣコントローラを有する。

【００３６】

前記ファイルシステムは、ネットワークアクセス可能ファイルが、ホスト記憶装置６６ではなく、いずれか１つのネットワーク記憶装置６６、３０５、３１５に格納されるように構成することができ、その代わりにそれはファイルシステムコード及びプロトコルスタックを含み、それらの複製が前記ホストにキャッシュされる。この構成によって、ホスト２０はネットワークファイル転送を制御するが、それらファイル内のデータの相当大部分は

、前記ホストに入ることすらなく、ＩＮＩＣ２２、２０３、３１０を通って高速パスにより転送することができる。ファイルブロックがネットワークと同じＩＮＩＣに接続された記憶装置との間で転送される場合には、ファイルデータは決してＩ／Ｏバスまたはホストメモリバスを横切ることがない。ファイルがネットワークと異なるＩＮＩＣに接続された記憶装置との間で転送される場合には、ファイルブロックはＤＭＡによって、記憶装置に接続されたＩＮＩＣ上のファイルキャッシュとの間でＩ／Ｏバスを通して送ることができ、依然としてホストメモリバスを回避する。この通常は回避される最悪の場合では、データはあるＩＮＩＣから別のものに転送することができ、それには、２回のＩ／Ｏバス転送や従来通りの繰り返しのホストメモリバスへのアクセスではなく、それに伴うＩ／Ｏバスの転送が１回であり、同様にホストメモリバスが回避される。

【００３７】

図６は、上述したＩＮＩＣ・Ｉ／Ｏコントローラ７２を必要とすることなく、ネットワーク通信接続及びネットワークストレージ接続を提供するＩＮＩＣ４００を含むネットワークストレージシステムを示している。簡潔にするために、ホスト２０及び関連する要素は、図１から変更はないものとして図示されているが、これは必ずしも当てはまらない。本実施例のＩＮＩＣは、第１ＬＡＮ４１４、第２ＬＡＮ４１６、第１ＳＡＮ４１８及び第２ＳＡＮ４２０に接続されたネットワーク接続またはポートを有する。ネットワーク４１４、４１６、４１８、４２０のいずれかまたは全部は、イーサネット、ファーストイーサネットまたはギガビットイーサネット規格に従って動作することができる。例えば８０２．３ｚ及び８０２．３ａｂ規格で記述されるギガビットイーサネットは、１ギガビット／秒または１０ギガビット／秒のデータ転送速度、もしくは将来的におそらくはより速い速度を提供する。ＳＡＮ４１８、４２０は、ＴＣＰ／ＩＰまたはＳＣＳＩカプセル化プロトコルにおいてＳＣＳＩのようなストレージプロトコルを実行することができる。そのようなストレージプロトコルの１つが、Ｊ．Ｓａｔｒａｎ他によりthe Internet Engineering Task Force（ＩＥＴＦ）のthe Internet-Draftにおいて「ｉＳＣＳＩ（ＩｎｔｅｒｎｅｔＳＣＳＩ）」の表題を付して２０００年６月に、それより早い時期のＩｎｔｅｒｎｅｔ－Ｄｒａｆｔでは「ＳＣＳＩ／ＴＣＰ（ＳＣＳＩ　ｏｖｅｒ　ＴＣＰ）」の表題を付して２０００年２月に記載されている。イーサネットストレージと称される、このような別のプロトコルがセッションレイヤにおいてＳＣＳＩカプセル化プロトコル（ＳＥＰ）を使用し、かつ基本的にＴＣＰが使用されているＷＡＮまたはインターネット上でデータが転送されているかどうか、またはＳＴＰが使用されるＬＡＮまたはＳＡＮ上でデータが転送されているかによって、トランスポートレイヤにおいてＴＣＰまたはＳＡＮトランスポートプロトコル（ＳＴＰ）を用いることを提案している。

【００３８】

ホスト２０は、ＰＣＩバスのようなＩ／Ｏバス４０によってＩＮＩＣ４００に接続され、これはＰＣＩバスインタフェースのようなＩＮＩＣ・Ｉ／Ｏブリッジ４０６によってＩＮＩＣバス４０４に接続されている。ＩＮＩＣ４００は、Ｉ／Ｏバス４０によってＩＮＩＣメモリ４１０に接続された専用プロセッサ４０８を有する。ＩＮＩＣメモリ４１０は、フレームバッファ４３０及びＩＮＩＣファイルキャッシュ４３３を有する。また、ＩＮＩＣバス４０４には、ネットワーク、トランスポート及びセッションレイヤプロセッシングを含むネットワークメッセージのプロセッシングを提供するハードウェアシーケンサ４１２の集合が接続されている。ＬＡＮ４１４、４１６及びＳＡＮ４１８、４２０への物理的接続は、従来の物理的レイヤハードウェアＰＨＹ４２２によって提供される。各ＰＨＹ４２２装置は、媒体アクセスコントロール（ＭＡＣ）４２４の対応する装置と接続され、ＭＡＣ装置はそれぞれＩＮＩＣ４００と１つのネットワークとの間にデータリンクレイヤ接続を提供する。

【００３９】

図７は、ＩＮＩＣ４００と第１イーサネット－ＳＣＳＩアダプタ４５２、第２イーサネット－ＳＣＳＩアダプタ４５４、及び第３イーサネット－ＳＣＳＩアダプタ４５６との間に接続されたギガビットイーサネットライン４５０を有するＳＡＮ４１８を示している。イ

ーサネット－ＳＣＳＩアダプタ４５２、４５４、４５６はＴＣＰ接続を形成しかつ遮断することができ、ＩＮＩＣ４００にＳＣＳＩコマンドを送りまたはそれから受け取り、かつ４５０を介してＩＮＩＣ４００にデータを送りまたはそれから受け取ることができる。第１記憶装置４６２が第１ＳＣＳＩケーブル４５８によって第１イーサネット－ＳＣＳＩアダプタ４５２に接続されている。同様に、第２記憶装置４６４が第２ＳＣＳＩケーブル４５９によって第２イーサネット－ＳＣＳＩアダプタ４５４に接続され、かつ第３記憶装置４６６が第３ＳＣＳＩケーブル４６０によって第３イーサネット－ＳＣＳＩアダプタ４５６に接続されている。記憶装置４６２、４６４、４６６はＳＣＳＩ規格に従って各アダプタ４５２、４５４、４５６によって動作する。各記憶装置は、それぞれのアダプタにデイジーチェーン式に接続された多数のディスクドライブを含むことができる。

【００４０】

図８は、第１イーサネット－ＳＣＳＩアダプタ４５２を詳細に示しており、これは本実施例において図１に示すものと類似するＩＮＩＣである。アダプタ４５２は、従来のＰＨＹ４７０によって提供されるネットワークライン４５０との物理的レイヤ接続及び従来のＭＡＣ４７２によって提供される媒体アクセスを有する１個のネットワークポートを有する。上位レイヤプロセッシングを含むネットワークメッセージのパケットヘッダのプロセッシングは、アダプタバス４７７を介してプロセッサ４８０及びアダプタメモリ４８２に接続されたシーケンサ４７５によって提供される。アダプタメモリ４８２はフレームバッファ４８４及びファイルキャッシュ４８６を有する。また、アダプタバス４７７にはＳＣＳＩコントローラが接続され、かつこれはＳＣＳＩチャネル４５８によって第１記憶装置４６２に接続されている。アダプタ４５２とＩＮＩＣ２０との１つの違いは、アダプタ４５２が低速パスメッセージを処理するためにＣＰＵ及びプロトコルスタックを有するホストに必ずしも接続されていないことである。本実施例において接続の設定は、例えば接続初期設定対話の際にアダプタ４５２に初期パケットをＩＮＩＣ４００が送ることによって、アダプタ４５２により行われ、前記パケットはシーケンサ４７５により処理されかつプロセッサ４８０に送られてＣＣＢを作成する。ソフトウェアプロトコルスタックを実行するＣＰＵによる低速パスプロセッシングを必要とする一定の状態は、このようなアダプタ４５２とＩＮＩＣ４００間の通信環境ではなお一層起こりにくいものである。アダプタ４５２とＩＮＩＣ４００間で送られるメッセージは、ＳＣＳＩ／ＴＣＰ及び簡易ネットワーク管理プロトコル（ＳＮＭＰ）のような単一のまたは限られたプロトコルレイヤの集合に従って構成することができ、かつ単一のソースへまたはそれから単一のもしくは制限された数のデスティネーションに送られる。従来の通信ネットワークに複雑な状態を生じさせる変数の多くを減らすことによって、高速パスプロセッシングの使用を増やすことができ、エラープロセッシングのためのアダプタ４５２の必要性が少なくなる。アダプタ４５２は、ＩＰ及びＴＣＰ上でいくつかのタイプのストレージプロトコルを処理する能力を持たせることができるが、それはアダプタ４５２が、ＩＮＩＣ４００に接続される代わりに、ネットワークストレージのためにそのようなプロトコルの１つを使用するホストに接続されるような場合である。ネットワーク４５０がストレージ転送に専用のＳＡＮではないが通信トラフィックを取り扱うような場合、低速パスパケットのためにプロトコルスタックを実行するＣＰＵを有するホストに接続されたＩＮＩＣは、アダプタ４５２の代わりに用いることができる。

【００４１】

図９に示すように、ＩＮＩＣ４００に類似する追加のＩＮＩＣが、Ｉ／Ｏバス４０を介してホスト２０に接続され、追加の各ＩＮＩＣによって追加のＬＡＮ接続が提供されかつ／または追加のＳＡＮに接続されている。複数のＩＮＩＣは、第Ｎ・ＩＮＩＣ４９０で表されており、これは第Ｎ・ＳＡＮ４９２及び第Ｎ・ＬＡＮ４９４に接続されている。ホスト２０に接続された複数のＩＮＩＣによって、前記ホストは複数のストレージネットワークを制御することができ、ホスト制御ネットワークとの間で行き来するデータの流れの相当大部分がホストプロトコルプロセッシング、Ｉ／Ｏバスを介しての移動、ホストバスを介しての移動及びホストメモリへの格納をバイパスする。

【００４２】
図１のＩＮＩＣ２２がネットワーク２５のようなネットワークから受信するメッセージパケットのプロセッシングが、図１０により詳細に示されている。受け取ったメッセージパケットは、まず媒体アクセスコントローラ６０に入り、これがネットワークへのＩＮＩＣアクセス及びパケットの受け取りを制御しかつネットワークプロトコル管理のための統計的情報を提供することができる。そこから、データは一度に１バイトづつ、本実施例では１２８ビットの幅を有するアセンブリレジスタ５００内に流れる。このデータは、図１１に関してより詳細に説明するように、フライバイシーケンサ５０２によって分類され、これはパケットのバイトをそれらが近くを飛ぶごとに調査し、それらのバイトから前記パケットを要約されるのに用いられることになるステータスを生成する。このように作成されたステータスは、マルチプレクサ５０５により前記データと併合され、その結果得られたデータがＳＲＡＭ５０８に格納される。パケット制御シーケンサ５１０がフライバイシーケンサ５０２を監督し、媒体アクセスコントローラ６０からの情報を調べ、データのバイトをカウントし、アドレスを生成し、かつアセンブリレジスタ５００からＳＲＡＭ５０８及び最終的にＤＲＡＭ５１２へのデータの移動を管理する。パケット制御シーケンサ５１０は、ＳＲＡＭコントローラ５１５を介してＳＲＡＭ５０８のバッファを管理し、またデータをＳＲＡＭ５０８からＤＲＡＭ５１２のバッファに移動させる必要がある時、ＤＲＡＭコントローラ５１８に知らせる。前記パケットに関するデータの移動が一旦完了しかつ全データがＤＲＡＭ５１２のバッファに移動すると、パケットコントロールシーケンサ５１０は、フライバイシーケンサ５０２によって生成されたステータスを、パケットデータにプリペンドされるようにＳＲＡＭ５０８に及びＤＲＡＭ５１２バッファの開始部分に移動させる。次に、パケット制御シーケンサ５１０は、キューマネージャ５２０に受信バッファ記述子を受信９に入力するように要求し、これが次にプロセッサ４４にパケットがハードウェア論理及びその要約されたステータスによって処理されたことを通知する。
【００４３】
図１１は、フライバイシーケンサ５０２がいくつかの階層を有することを示しており、各階層は一般にパケットヘッダの特定部分及び特定のプロトコルレイヤに、そのレイヤに関連するステータスを生成するために、焦点を合わせている。本実施例のフライバイシーケンサ５０２は、媒体アクセス制御シーケンサ５４０、ネットワークシーケンサ５４２、トランスポートシーケンサ５４６及びセッションシーケンサ５４８を有する。より上位のプロトコルレイヤに関係するシーケンサを追加で設けることができる。フライバイシーケンサ５０２は、パケット制御シーケンサ５１０によってリセットされ、かつ一定のバイトがアセンブリレジスタ５００から利用可能であるかをフライバイシーケンサに知らせるパケット制御シーケンサによってポインタが与えられる。媒体アクセス制御シーケンサ５４０は、バイト０～５を見ることによってパケットが別のホストではなくまたは別のホストに加えてホスト２０にアドレス指定されていることを判断する。また、パケットのオフセット１２、１３が媒体アクセス制御シーケンサ５４０により処理されて、タイプフィールド、例えば前記パケットがイーサネットまたは８０２．３であるかどうかを決定する。タイプフィールドがイーサネットの場合、それらのバイトは更に前記パケットのネットワークプロトコルタイプを媒体アクセス制御シーケンサ５４０に知らせる。８０２．３の場合には、そうではなく、それらのバイトはフレーム全体の長さを示し、媒体アクセス制御シーケンサ５４０が前記パケット内に更に８バイトを調べて、ネットワークレイヤのタイプを決定することになる。
【００４４】
大部分のパケットについて、ネットワークシーケンサ５４２が、受け取ったヘッダの長さが正しい長さであるかを検証し、かつネットワークレイヤのヘッドをチェックサムする。高速パスの対象について、ネットワークレイヤのヘッダは、媒体アクセス制御シーケンサ５４０により行われる分析から、ＩＰまたはＩＰＸであることがわかる。例えば、タイプフィールドが８０２．３でありかつネットワークプロトコルがＩＰであると仮定すると、ネットワークシーケンサ５４２は、ＩＰタイプを決定するために、バイト２２で始まるこ

とになるネットワークレイヤヘッダの最初のバイトを分析する。ＩＰヘッダの最初のバイトは、ネットワークシーケンサにより処理されて、前記パケットが関連するＩＰタイプを決定することになる。前記パケットが例えばＩＰバージョン４に関するものであることを決定することで、ネットワークシーケンサ５４２による処理が更に行われ、該ネットワークシーケンサは前記パケットのトランスポートヘッダプロトコルを表示するためにＩＰヘッダ内への１０バイトに配置されたプロトコルタイプをも見る。例えば、イーサネット上でのＩＰの場合、ＩＰヘッダはオフセット１４で始まり、かつプロトコルタイプバイトはオフセット２３であり、これはネットワーク論理により処理されて、トランスポートレイヤプロトコルが例えばＴＣＰであるかどうかを決定する。一般に２０～４０バイトであるネットワークレイヤヘッダの長さから、ネットワークシーケンサ５４２は、トランスポートレイヤヘッダを検証するためにパケットのトランスポートレイヤヘッダの開始部分を決定する。トランスポートシーケンサ５４６は、トランスポートレイヤヘッダ及びデータのチェックサムを生成することができ、これには、少なくともＴＣＰの場合にＩＰヘッダからの情報を含むことができる。

【００４５】

ＴＣＰパケットの例について続けて説明すると、トランスポートシーケンサ５４６はまた、ヘッダのトランスポートレイヤ部分の最初のいくつかのバイトを分析して、前記パケットがＮｅｔＢｉｏｓまたは別のプロトコルであるかどうかのようなメッセージについてＴＣＰソース及びデスティネーションポートを部分的に決定する。ＴＣＰヘッダのバイト１２をトランスポートシーケンサ５４６により処理されて、ＴＣＰヘッダの長さを決定しかつ検証する。ＴＣＰヘッダのバイト１３は、認証フラグ及びプッシュフラグは別として、プロセッサにこのパケットを例外として分類させることがある、リセット及び終了のような予期しないオプションを示すことがあるフラグを有する。ＴＣＰオフセットバイト１６、１７は、ハードウェア論理によって取り出されかつ格納されるチェックサムであり、前記フレームの残りの部分は該チェックサムに対して検証される。

【００４６】

セッションシーケンサ５４８は、セッションレイヤヘッダの長さを決定し、これは、ＮｅｔＢｉｏｓの場合にはたった４つのバイトであり、その内の２つがＮｅｔＢｉｏｓペイロードデータの長さを知らせるが、これは他のプロトコルについてより大きくすることができる。また、セッションシーケンサ５４８を用いて、例えば、特に高速パスが有利であるような読出しまたは書込みとしてメッセージのタイプを分類することができる。メッセージのタイプによって、更に上位レイヤの論理プロセッシングが、パケット制御シーケンサ５１０及びフライバイシーケンサ５０２のハードウェア論理によって実行することができる。このようにして、シーケンサ５２は、バイトの単一のストリームから選択したバイトの分類によってヘッダのハードウェアプロセッシングを知的に行い、パケットのステータスがオンザフライで決定された分類から構築される。全てのパケットがフライバイシーケンサ５０２により処理されたことをパケット制御シーケンサ５１０が一旦検出すると、パケット制御シーケンサ５１０は、フライバイシーケンサ５０２により生成されたステータス情報及びパケット制御シーケンサ５１０により生成されたあらゆるステータス情報を付け加え、かつプロセッサ４４によるパケットの取り扱いに好都合なように、それらのステータス情報をパケットにプリペンド（前部に追加）する。パケット制御シーケンサ５１０により生成された追加のステータス情報には、媒体アクセスコントローラ６０のステータス情報及び発見されたあらゆるエラー、またはアセンブリレジスタまたはＤＲＡＭバッファにおけるデータのオーバフロー、または前記パケットに関する他の様々な情報が含まれる。また、パケット制御シーケンサ５１０は、キューマネージャ５２０を介して受信バッファ９及び受信統計９への入力を記憶する。

【００４７】

ハードウェア論理によりパケットを処理することの利点は、従来のシーケンシャルソフトウェアプロトコルプロセッシングと対照的に、パケットを格納したり、移動したり、コピーしたり、各プロトコルレイヤヘッダを処理するためにストレージから引き抜いたりする

必要がなく、処理効率の大幅な増加及び各パケットについて処理時間の節約が得られることである。パケットは、ネットワークから受け取ったレートビットで、例えば１００ベースＴ接続の場合に１００メガビット／秒で処理することができる。このレートで受信されかつ６０バイトの長さを有するパケットを分類するための時間は、従って約５ミリ秒である。このパケットをハードウェア論理で処理しかつパケットデータを高速パスを介してそのホストデスティネーションに送るための合計時間は、ＣＰＵの割込みの減少で得られる追加の時間の節約及びホストバスバンド幅の節約を考慮することさえ無く、従来のシーケンシャルソフトウェアプロトコルプロセッシングを用いる従来のＣＰＵが必要とするものよりも、大幅に少なくすることができる。デスティネーションがＩＮＩＣキャッシュにあるような場合には、ホストバス３５及びＩ／Ｏバス４０について追加のバンド幅の節約が得られる。

【００４８】

プロセッサ４４は、フレームバッファ７７に保持される各受信メッセージパケットについて、そのパケットが高速パスの対象であるかどうかを選択し、かつそうである場合には、前記パケットが属する接続について高速パスが既に接続されているかどうかを調べる。これをするために、プロセッサ４４はまずヘッダのステータスサマリを調べて、パケットのヘッダが高速パスの対象について定義されたプロトコルからなるものであるかどうかを決定する。そうでない場合には、プロセッサ４４は、ＩＮＩＣ２２内のＤＭＡコントローラに命令して、前記パケットを低速パスプロセッシングのためのホストに送る。メッセージの低速パスプロセッシングの場合でさえ、このようにＩＮＩＣ２２は、メッセージタイプの検証及び決定のような初期処理を実行し、かつ検証したメッセージを少なくともホストのデータリンクレイヤに送る。

【００４９】

高速パス対象の場合、プロセッサ４４は、ヘッダのステータスサマリがＩＮＩＣにより保持されてるＣＣＢと一致するかどうかを調べる。そうである場合、前記パケットからのデータは、高速パスに沿って前記ホストのデスティネーション１６８に送られる。高速パス対象のパケットサマリがＩＮＩＣにより保持されるＣＣＢと一致しない場合には、前記パケットは低速パスプロセッシングのために前記ホストに送られ、該メッセージのＣＣＢを作成する。また、断片化されたメッセージや他の複雑さがある場合には高速パスが用いられないことがある。しかしながら、相当大部分のメッセージについては、ＩＮＩＣ高速パスによってメッセージのプロセッシングを大幅に加速することができる。このようにＩＮＩＣ２２は、与えられたパケットの運命を決定するためにいくつかのプロトコルレイヤのそれぞれにステートマシンを用いる従来のものと対照的に、オンザフライで集めた情報に基づいて、データをそのデスティネーションに直接送るかどうかを決定する単一のステートマシンプロセッサ４４が提供される。

【００５０】

ＣＣＢをＩＮＩＣのハッシュテーブルにキャッシングすることによって、到来するパケットを要約する語との素早い比較が行われ、パケットを高速パスを介して処理できるかどうかが決定される一方、一杯のＣＣＢがプロセッシングのためにＩＮＩＣに保持される。この比較を加速する別の方法には、Ｂ－ツリーのようなソフトウェア処理またはコンテンツアドレス可能メモリ（ＣＡＭ）のようなハードウェアアシストが含まれる。ＩＮＩＣマイクロコードまたはコンパレータ回路がＣＣＢとの整合を検出すると、ＤＭＡコントローラが前記パケットからのデータをホストメモリ３３またはＩＮＩＣファイルキャッシュ８０のデスティネーションにＣＰＵによる割込み、プロトコルプロセッシングまたは複製なしで配置する。受信したメッセージのタイプによって、前記データのデスティネーションはホスト２０、またはホストファイルキャッシュ２４、ＩＮＩＣファイルキャッシュ８０のセッションレイヤ、プレゼンテーションレイヤまたはアプリケーションレイヤであったりする。

【００５１】

ファイル転送のような大きなメッセージについて最も一般的に使用されているネットワー

クプロトコルの１つは、ＴＣＰ／ＩＰ上のサーバメッセージブロック（ＳＭＢ）である。ＳＭＢは、ファイルが書き込まれるプリンタまたはディスクのような特定の動作のために必要なリソースが前記動作が行われたホストに存在するかまたはそれに関連するかもしくはファイルサーバのようなネットワークに接続された別室のホストに配置されているかどうかを決定するリダイレクタソフトウェアと協力して動作することができる。ＳＭＢ及びサーバフラッシュリダイレクタは、通常トランスポートレイヤからサービスを受ける。本発明では、ＳＭＢ及びリダイレクタはそうではなく、ＩＮＩＣからサービスを受けることができる。この場合、大きなＳＭＢトランズアクションを受け取った時、ＩＮＩＣバッファからＤＭＡコントローラによってデータを送ることは、ホストが取り扱わなければならない割込みを大幅に少なくすることができる。更に、このＤＭＡは、通常データをそのホストファイルキャッシュ２４またはＩＮＩＣファイルキャッシュ８０内のデスティネーションに移動させ、そこから次にブロックでホスト記憶装置６６またはＩＮＩＣ記憶装置７０へそれぞれフラッシュされる。

【００５２】

ＳＭＢ高速パス電送は一般に上述したＳＭＢ高速パス受信を逆にし、データのブロックがホスト記憶装置６６またはＩＮＩＣ記憶装置７０からホストファイルキャッシュ２４またはＩＮＩＣファイルキャッシュ８０へそれぞれ読み取られる一方、関連するプロトコルヘッダが、ネットワークラインを介してリモートホストへの伝送のために、ＩＮＩＣにより前記データにプリペンドされる。カスタムハードウェア経由かつホストの繰り返し割込み無しにＩＮＩＣにより複数パケット及び複数ＴＣＰ、ＩＰ、ＮｅｔＢｉｏｓ及びＳＭＢプロトコルレイヤのプロセッシングによって、ＳＭＢメッセージをネットワークラインに伝送する速度を大幅に増加させることができる。図４に関連して上述したように、送信したファイルブロックがＩＮＩＣ記憶装置７０上に格納される場合には、ホストバス３５バンド幅及びＩ／Ｏバスバンド幅４０における追加の節約を得ることができる。

【００５３】

図１２は、そのいずれも同図には記載されていないが、ネットワークメッセージを処理するために、ＩＮＩＣと共にホストにより使用されるAlacritechのプロトコルスタック３８を示している。ＩＮＩＣドライバ５６０はＩＮＩＣをホストのオペレーティングシステムにリンクさせ、ＩＮＩＣとプロトコルスタック３８間で通信を送ることができる。本実施例のプロトコルスタック３８はデータリンクレイヤ５６２、ネットワークレイヤ５６４、トランスポートレイヤ５６６、上位レイヤインタフェース５６８及び上位レイヤ５７０を有する。上位レイヤ５７０は、使用される特定のプロトコル及び通信されるメッセージによって、セッションレイヤ、プレゼンテーションレイヤ及び／またはアプリケーションレイヤを表すことができる。プロトコルスタック３８は低速パスのパケットヘッダを処理し、接続を作りかつ壊し、ＩＮＩＣへの高速パス接続のためにＣＣＢを分配し、かつＩＮＩＣからホスト２０へフラッシュされる高速パス接続のためにＣＣＢを受け取る。上位レイヤインタフェース５６８は一般に、データリンクレイヤ５６２、ネットワークレイヤ５６４及びトランスポートレイヤ５６６により作成された接続及びステータス情報に基づいてＣＣＢを組み立てること、及びＩＮＩＣデバイスドライバ５６０経由ＣＣＢをＩＮＩＣに分配すること、またはＩＮＩＣデバイスドライバ５６０経由ＩＮＩＣからフラッシュされたＣＣＢを受け取ることに責任を有する。

【００５４】

図１３は、マイクロソフト（登録商標）のオペレーティングシステムと協力してネットワーク通信を処理するための複数のプロトコルスタックを有するAlacrytechのプロトコルスタック３８の別の実施例を示している。従来のマイクロソフトのＴＣＰ／ＩＰプロトコルスタック５８０はＭＡＣレイヤ５８２、ＩＰレイヤ５８４及びＴＣＰレイヤ５８６を有する。コマンドドライバ５９０がホストスタック５８０と協力して動作し、ネットワークメッセージを処理する。コマンドドライバ５９０はＭＡＣレイヤ５９２、ＩＰ５９４及びAlacritechのＴＣＰ（ＡＰＣＰ）レイヤ５９６を有する。従来のスタック５８０とコマンドドライバ５９０とが、ＩＮＩＣデバイスドライバ５７０と対話するネットワークドライバ

インタフェース仕様（ＮＤＩＳ）レイヤ５９８を共有する。ＩＮＩＣデバイスドライバ５７０は、従来のホストスタック５８０またはＡＴＣＰドライバ５９０によって処理するために受信表示をソートする。ＰＤＩフィルタドライバ及び上位レイヤインタフェース５７２は同様に、ＴＤＩユーザ５７５からネットワークに送られるメッセージがコマンドドライバにかつ恐らくはＩＮＩＣの高速パスへ向きを変えられるか、またはホストスタックにより処理されるかを決定する。

【００５５】

図１４は、共にＩＮＩＣを有するサーバ６００とクライアント６０２との間のＳＭＢエクスチェンジを示しており、各ＩＮＩＣは、ギガビットイーサネットコンプライアントで有り得るネットワーク６０４上でのデータの高速パス移動のためにその接続及びステータスを定義するＣＣＢを有する。クライアント６０２はＩＮＩＣ６０６、８０２．３コンプライアントデータリンクレイヤ６０８、ＩＰレイヤ６１０、ＴＣＰレイヤ６１１、ＡＴＣＰレイヤ６１２、ＮｅｔＢｉｏｓレイヤ６１４及びＳＭＢレイヤ６１６を有する。前記クライアントは、通信処理のために低速パス６１８及び高速パス６２０を有する。同様に、サーバ６００はＩＮＩＣ６２２、８０２．３コンプライアントデータリンクレイヤ６２４、ＩＰレイヤ６２６、ＴＣＰレイヤ６２７、ＡＴＣＰレイヤ６２８、ＮｅｔＢｉｏｓレイヤ６３０及びＳＭＢレイヤ６３２を有する。サーバ直接接続型記憶装置６３４が、ＳＣＳＩチャネルのようなパラレルチャネル６３８上でサーバと接続される。これはＩＮＩＣ６２２にも接続されたＩ／Ｏバス６３９に接続されている。ネットワーク記憶装置６４０がネットワークライン６４４上でＩＮＩＣ６２２に接続され、かつＮＡＳ記憶装置６４２が、ギガビットイーサネットコンプライアントで有り得る同じネットワーク６４４に接続されている。サーバ６００は、図１４に示されてないファイルキャッシュとＩＮＩＣ６２２間でＩ／Ｏバス６３８上を通過する通信プロセッシングのための低速パス６４６及び高速パス６４８を有する。

【００５６】

ネットワーク記憶装置６４０または６４２とクライアント６０２との間で転送される、Ｉ／Ｏバスを通らないデータのために、サーバの制御下でＩＮＩＣ６２２によりストレージ高速パスが提供される。データはＳＣＳＩ／ＴＣＰまたはＩＳＣＳＩのようなブロック形式に従ってＩＮＩＣ６２２とネットワーク記憶装置６４０との間で通信され、データはＴＣＰ／ＮｅｔＢｉｏｓ／ＳＭＢのようなファイル形式に従ってＩＮＩＣ６２２とＮＡＳ記憶装置６４２との間で送信される。いずれのストレージ高速パスについても、ＩＮＩＣ６２２が、記憶装置６４０または６４２との接続を定義する別のＣＣＢを保持することができる。以下の説明を簡便にするために、そのサーバ６００とのネットワーク６０４上での接続を定義するＩＮＩＣ６０６により保持されるＣＣＢをクライアントＣＣＢと称し、そのクライアント６０２とのネットワーク６０４上での接続を定義するＩＮＩＣ６２２により保持されるＣＣＢをサーバＣＣＢと称する。そのネットワーク記憶装置６４０とのネットワーク６４４上での接続を定義するＩＮＩＣ６２２により保持されたＣＣＢをＳＡＮ・ＣＣＢと称し、そのＮＡＳ記憶装置６４２とのネットワーク６４４上での接続を定義するＩＮＩＣ６２２により保持されたＣＣＢをＮＡＳ・ＣＣＢと称する。追加のネットワークライン６５０、６５２を別の通信及び／またはストレージネットワークと接続することができる。

【００５７】

クライアント６０２が、ネットワーク記憶装置６４０上にブロックで格納されたサーバ６００上の１００ＫＢファイルを読み出したいと仮定すると、前記クライアントは、ネットワーク６０４を通ってＳＭＢ読出しリクエストを送ることによって開始し、前記サーバ上のそのファイルの最初の６４ＫＢを要求することができる。この要求は、例えば７６バイトだけであってもよく、前記サーバ上のＩＮＩＣ６２２はメッセージのタイプ（ＳＭＢ）及び比較的小さいメッセージのサイズを認識し、前記７６バイトを直接ＡＴＣＰフィルタレイヤ６２８に送り、それが前記リクエストをサーバのＮｅｔＢｉｏｓ６３０に送信する。ＮｅｔＢｉｏｓ６３０はセッションヘッダをＳＭＢ６３２に送り、これが前記読出しリ

クエストを処理して、要求されたデータがホストまたはＩＮＩＣファイルキャッシュ上に保持されているかどうかを決定する。要求されたデータがファイルキャッシュにより保持されていない場合、ＳＭＢはファイルシステムに読出しリクエストを出して、ネットワーク記憶装置６４０からのデータをＩＮＩＣ６２２ファイルキャッシュ内に書き込む。

【００５８】

これを実行するために、ファイルシステムはＩＮＩＣ６２２に、ネットワーク記憶装置６４０から６４ＫＢのデータをＩＮＩＣ６２２ファイルキャッシュ内にフェッチするように指示する。次にＩＮＩＣ６２２は、前記データへのリクエストをネットワーク記憶装置６４０へネットワーク６４４で送信する。このリクエストは前記ブロックを読取る記憶装置６４０への１つまたは複数のＳＣＳＩコマンドの形を取ることができ、前記コマンドにはＩＳＣＳＩまたは類似のプロトコルに従って、ＴＣＰ／ＩＰヘッダが付加されている。記憶装置６４０上のコントローラがそのディスクドライブから要求されたブロックを読出し、ＩＳＣＳＩまたは類似のプロトコルヘッダを前記ブロックまたは該ブロックのフレームサイズの部分に付け加え、かつその結果得られるフレームをネットワーク６４４上でＩＮＩＣ６２２に送ることによって、前記コマンドに応答する。前記フレームはＩＮＩＣ６２２により受け取られ、ＩＮＩＣ６２２シーケンサにより処理され、ストレージＣＣＢと整合され、かつ要求された１００ＫＢファイルの一部分を形成するＩＮＩＣファイルキャッシュの６４ＫＢファイルストリームとして再び組立てられる。前記ファイルストリームがＩＮＩＣ６２２ファイルキャッシュ上に一旦格納されると、ＳＭＢは読出し応答を組立てかつそのファイルストリームを示すスキャッタギャザリストをＩＮＩＣ６２２に送り、かつ前記応答をＩＮＩＣ６２２に送って、サーバＣＣＢに従ってネットワーク上で前記データを送る。ＩＮＩＣ６２２は前記スキャッタギャザリストを用いて、そのファイルキャッシュからデータパケットを読出し、これは、サーバＣＣＢに基づいてＩＮＩＣにより作成されたＩＰ／ＴＣＰ／ＮｅｔＢｉｏｓ／ＳＭＢヘッダとプリペンドされ、かつその結果のフレームをネットワーク６０４上に送る。前記ファイルの残りの３６ＫＢは、同様の手法で送られる。このようにして、ネットワーク記憶装置上のファイルは、サーバの制御下で、Ｉ／Ｏバスまたはサーバプロトコルスタックに出会うファイルからのデータ無しで転送することができる。

【００５９】

クライアント６０２により要求されたデータがＮＡＳ記憶装置６４２上に格納される場合には、リクエストはサーバ６００からその記憶装置６４２へ送ることができ、これはクライアント６０２にアドレス指定されたヘッダを有するデータを送ることによって応答し、サーバ６００がルータとして機能する。サーバ６００がプロキシサーバまたはウェブキャッシュサーバとして実行されるような実施例では、ＮＡＳ記憶装置からのデータをサーバ６００に送ることができ、これが前記データをそのファイルキャッシュに格納して、そのデータに関する将来のリクエストに対するより早い応答を提供する。この実施例では、サーバ６００上のファイルシステムが、ＮＡＳ記憶装置６４２上のファイルデータを要求するようにＩＮＩＣを指示し、これがファイルデータの最初の６４ＫＢを含む多数の約１．５ＫＢパケットを送ることによって応答する。前記ファイルデータを含むパケットはＩＮＩＣ６２２が受信し、ＩＮＩＣ受信シーケンサにより分類されかつＮＡＳ・ＣＣＢと整合され、最初のパケットからのセッションレイヤヘッダがホストスタックにより処理され、これがファイルシステムからＩＮＩＣ６２２ファイルキャッシュのアドレスのスキャッタギャザリストを入手して、前記パケットからのデータを格納する。前記スキャッタギャザリストはホストスタックによりＩＮＩＣ６２２に送られ、かつＮＡＳ・ＣＣＢと共に格納され、ＩＮＩＣ６２２が、ＮＡＳ・ＣＣＢに対応するあらゆる蓄積パケット及びその後に続くパケットからのデータを、スキャッタギャザリストに従ってファイルストリームとしてＩＮＩＣ６２２ファイルキャッシュ内へＤＭＡし始める。次に、ホストファイルシステムがＩＮＩＣ６２２に指示して、クライアントＣＣＢに基づいてヘッダを作成し、かつ前記ファイルストリームから読み出されたデータのパケットに、該データをクライアント６０２に送るためにヘッダをプリペンドする。前記ファイルの残りの３６ＫＢは同様の手法

により送られ、かつ別のファイルストリームとしてＩＮＩＣ６２２ファイルキャッシュにキャッシュすることができる。ファイルストリームがＩＮＩＣ６２２ファイルキャッシュに保持されることによって、クライアント６０６のようなクライアントへの前記ファイルへの後から来るリクエストが、より早く処理することができる。

【００６０】

クライアント６０２により要求されたファイルがキャッシュに存在せず、その代わりにファイルブロックとしてサーバ接続記憶装置６３４上に格納されていた場合には、サーバ６２２ファイルシステムがホストＳＣＳＩドライバを指示して、サーバ接続記憶装置６３４から１００ＫＢのデータをサーバ６００ファイルキャッシュ内にフェッチさせる（前記ファイルシステムはＩＮＩＣ６２２ファイルキャッシュ上のデータをキャッシュしたくないと仮定する）。次に、ホストＳＣＳＩドライバがＳＣＳＩチャネル６３８上で前記データへのＳＣＳＩリクエストをサーバ直接接続型記憶装置６３４に送る。サーバ直接接続型記憶装置６３４上のコントローラが、要求されたブロックをそのディスクドライブから読出しかつ該ブロックをＳＣＳＩチャネル６３９上で前記ＳＣＳＩドライバに送ることによって前記コマンドに応答し、これが前記ファイルシステムの命令下でキャッシュマネージャと対話して、前記ブロックをファイルストリームとしてサーバ６００ファイルキャッシュに格納する。次にファイルシステムリダイレクタがＳＭＢを指示して、 前記ファイルストリームのスキャッタギャザリストをＩＮＩＣ６２２に送り、これをＩＮＩＣ６２２が用いて、サーバ６００ファイルストリームからデータパケットを読み出す。ＩＮＩＣ６２２は、サーバＣＣＢに基づいて作成されたヘッダで前記データパケットをプリペンドしかつその結果のフレームをネットワーク６０４上に送る。

【００６１】

この応答が到着したときクライアント６０２上でＩＮＩＣ６０６が動作し、ＩＮＩＣ６０６は受け取った第１フレームからこの接続が高速パス６２０プロセッシング（ＴＣＰ／ＩＰ、ＮｅｔＢｉｏｓ、ＣＣＢをマッチング）を受け取っていることを認識し、ＳＭＢ６１６がこの第１フレームを用いてメッセージのためのバッファステージを獲得する。バッファの割り当ては、あらゆるＮｅｔＢｉｏｓ／ＳＭＢヘッダを含む前記フレームの最初の１９２バイトをＡＴＣＰ高速パス６２０を介してクライアントＮｅｔＢｉｏｓ６１４に直接送り、適当なヘッダをＮｅｔＢｉｏｓ／ＳＭＢに与えることによって提供される。ＮｅｔＢｉｏｓ／ＳＭＢはこれらのヘッダを分析し、リクエストＩＤとの整合によって、これが元の読出し接続への応答であることを認識し、かつその中にデータが配置されるクライアントファイルキャッシュのバッファの６４Ｋリストを ＡＴＣＰコマンドドライバに与える。この段階ではただ１つのフレームが到着しているが、このプロセッシングが行なわれている間により多くのフレームが到着し得る。前記クライアントバッファリストがＡＴＣＰコマンドドライバ６２８に与えられるとすぐに、これはその転送情報をＩＮＩＣ６０６に送り、ＩＮＩＣ６０６はＤＭＡによりそれらのバッファ内に蓄積されている全てのフレームデータの送信を開始する。

【００６２】

クライアント６０２がＳＭＢファイルをサーバ６００に書き込みたい場合には、書込みリクエストが、ＩＮＩＣ６２２により保持されているＣＣＢとマッチングするであろうネットワーク６０４上を送信される。ファイル書込みの最初のパケットからのセッションレイヤヘッダがサーバＳＭＢ６３２により処理されて、サーバ６００またはＩＮＩＣ６２２ファイルキャッシュ内にバッファが割り当てられ、それらバッファに関するアドレスのスキャッタギャザリストが、高速パスプロセッシングが適当であると仮定した場合、ＩＮＩＣ６２２へ戻される。ＳＭＢファイルデータを含むパケットがＩＮＩＣ６２２により受け取られ、ＩＮＩＣ受信シーケンサにより分類されかつキューの中に配置される。ＩＮＩＣ６２２プロセッサは、パケットがサーバＣＣＢに対応していることを認識しかつ前記パケットからのデータをＩＮＩＣ６２２またはサーバ６００ファイルキャッシュバッファ内にスキャッタギャザリストに従ってＤＭＡして、ファイルストリームを形成する。

【００６３】

次に、前記ファイルシステムが、前記ファイルストリームのサーバ記憶装置６３４、ネットワーク記憶装置６４０またはＮＡＳ記憶装置６４２への送信を編成する。ファイルストリームをサーバ記憶装置６３４に送るために、ファイルシステムはサーバ６００内のＳＣＳＩドライバに命令して、前記ファイルストリームをファイルブロックとして記憶装置６３４に送る。ファイルストリームをネットワーク記憶装置６４０に送るために、ファイルシステムはＩＮＩＣに命令して、ＳＡＮ・ＣＣＢに基づいてＩＳＣＳＩまたは類似のヘッダを作成し、それらのヘッダを前記スキャッタギャザリストに従って前記ファイルストリームから読み出したパケットにプリペンドし、その結果のフレームをネットワーク６４４上で記憶装置６４０に送る。例えば分散ファイルキャッシュまたはプロキシサーバの実行に有用で有り得るＮＡＳ記憶装置６４２に前記ファイルストリームを送るために、ファイルシステムリダイレクタは適当なＮｅｔＢｉｏｓ／ＳＭＢヘッダをプリペンドし、かつＩＮＩＣに命令して、ＮＡＳ・ＣＣＢに基づいてＩＰ／ＴＣＰヘッダを作成し、かつそれらのヘッダを、スキャッタギャザリストに従って前記ファイルストリームから読み出されたパケットにプリペンドし、その結果のフレームをネットワーク６４４上で記憶装置６４２に送る。

【００６４】

図１５は図１４に示すものと類似のものを示しているが、図１５は、特にリアルタイムトランスポートプロトコル（ＲＴＰ）及びリアルタイムトランスポートコントロールプロトコル（ＲＴＣＰ）が関係し得る音声及び映像通信のコンテキストにおいて、本発明のユーザデータグラムプロトコル（ＵＤＰ）に焦点を合わせている。サーバ６００は、図１４に示すプロトコルレイヤに加えて、ＵＤＰレイヤ６５４、ＡＵＤＰレイヤ６５５、ＲＴＰ／ＲＴＣＰレイヤ６５６及びアプリケーションレイヤ６５７を有する。同様に、クライアント６０２は、図１４に示すプロトコルレイヤに加えてＵＤＰレイヤ６６０、ＡＵＤＰレイヤ６６１、ＲＴＰ／ＲＴＣＰレイヤ６６２及びアプリケーションレイヤ６６３を有する。図１５にＲＴＰ／ＲＴＣＰレイヤ６５６、６６２は図示されているが、セッション初期設定プロトコル（ＳＩＰ）または媒体ゲートウェイコントロールプロトコル（ＭＧＣＰ）のような他のセッション及びアプリケーションプロトコルをＵＤＰの上に用いることができる。

【００６５】

オーディオ／ビデオインタフェース（ＡＶＩ）６６６または類似の周辺装置がサーバ６００のＩ／Ｏバス６３９に接続され、かつＡＶＩ６６６にはスピーカ６６８、マイク６７０、ディスプレイ６７２及びビデオカメラで有り得るカメラ６７４が接続されている。簡単にするために１つのインタフェースとして図示されているが、これに変えてＡＶＩ６６６は、サウンドカードまたはビデオカードのようなそれぞれＩ／Ｏバス６３９に接続される別個のインタフェースを有することができる。別の実施例では、ＡＶＩインタフェースがＩ／Ｏバスではなくホストバスに接続され、かつホストＣＰＵのメモリコントローラまたはＩ／Ｏコントローラと一体にすることができる。ＡＶＩ６６６は、音声または映像データのための一時的記憶装置として機能するメモリを有することができる。また、クライアント６０２はＩ／Ｏバス６７５を有し、ＡＶＩ６７７はＩ／Ｏバス６７５に接続されている。ＡＶＩ６７７に接続されているのは、スピーカ６７８、マイク６８０、ディスプレイ６８２、及びビデオカメラで有り得るカメラ６８４である。簡単にするために１つのインタフェースとして図示されているが、これに代えてＡＶＩ６６６は、サウンドカード及びビデオカードのようなそれぞれＩ／Ｏバス６７５に接続された別個のインタフェースを有することができる。

【００６６】

ＴＣＰと異なり、ＵＤＰは従属性の接続を提供しない。その代わりに、ＵＤＰパケットはベストエフォートベースで送られ、かつ失われたまたは壊れたパケットはＵＤＰより上位のレイヤがそのようなサービスを提供しない限り、再送されない。ＵＤＰは、接続を確立する必要なくＩＰを介してアプリケーションがデータを送る方法を提供する。しかしながら、ＵＤＰまたはネットワークファイルシステム（ＮＦＦ）、ＴＣＰ、ＲＴＣＰ、ＳＩＰ

またはＭＧＣＰのような別のプロトコルによるリクエストに応答してソケットが指定され、そのソケットがＵＤＰにより送られたメッセージを受け入れることができる受信装置上のポートを割り当てる。ソケットは、ソース及びデスティネーションＩＰアドレス並びにソース及びデスティネーションＵＤＰポートを示すＵＤＰにより用いられるアプリケーションプログラミングインタフェース（ＡＰＩ）である。

【００６７】

ＵＤＰを介して従来どおりに通信を送るために、セッションレイヤまたはアプリケーションレイヤのヘッダを含むことができ、かつ通常約８ＫＢまでのＮＦＳデフォルトサイズのデータグラムである６４ＫＢまでのデータグラムを、ホストコンピュータによりＵＤＰヘッダでプリペンドしてＩＰレイヤに渡す。ＵＤＰヘッダは、ソース及びデスティネーションポート並びに任意のチェックサムを含む。イーサネット伝送の場合、ＵＤＰダイヤグラムは、必要に応じて、ＩＰレイヤにより約１．５ＫＢのフラグメントに分割され、ＩＰヘッダでプリペンドされかつイーサネットヘッダ及びトレーラでのカプセル化のためにＭＡＣレイヤに与えられ、かつネットワーク上に送られる。ＩＰヘッダは、ＵＤＰデータグラムに固有のＩＰ識別フィールド（ＩＰ・ＩＤ）と、そのダイヤグラムのより多くのフラグメントが続いているかどうかを示すフラグフィールドと、前記フラグメントを正しい順序で再アセンブリするために、前記ダイヤグラムのどの１．５ＫＢフラグメントがＩＰヘッダに接続されたかを示すフラグメントオフセットフィールドを有する。

【００６８】

クライアント６０２からサーバ６００への高速パスＵＤＰデータ転送の場合、例えばクライアントアプリケーション６６３またはオーディオ／ビデオインタフェース６７７からのデータのファイルストリームは、ファイルシステムの命令下でＩＮＩＣ６０６のメモリ上に格納することができる。アプリケーション６６３は、例えば約８ＫＢを有するように前記ファイルストリームを構成することができ、それはアプリケーション６６３の命令下でＩＮＩＣ６０６により獲得される上位レイヤヘッダを含むことができ、ＩＮＩＣ６０６により受け取られる各ファイルストリームは、指定されたソケットに従ってＩＮＩＣ６０６によりＵＤＰヘッダでプリペンドされ、ＵＤＰデータグラムを作成する。ＵＤＰデータグラムは、ＩＮＩＣ６０６により６つの１．５ＫＢメッセージフラグメントに分割され、それはそれぞれＩＰ及びＭＡＣレイヤヘッダでプリペンドされて、ネットワーク６０４上に伝送されるＩＰパケットを作成する。

【００６９】

ＩＮＩＣ６２２は、ＩＮＩＣ６０６から送られたイーサネットフレームをネットワーク６０４から受け取り、ヘッダをチェックサムしかつ処理して、関連するプロトコルを決定し、かつＵＤＰ及び上位レイヤヘッダをＡＵＤＰレイヤ６５５に送って、そのＵＤＰデータグラムのパケットから前記データのデスティネーションアドレスのリストを入手する。ＵＤＰ及び上位レイヤヘッダはＵＤＰデータグラムからの最大６つまでのパケットの１つに含まれ、かつそのパケットは通常そのデータグラムからの他のパケットより先に受け取られる。前記パケットが間違った順序で到着する場合には、ＩＮＩＣ６２２が再アセンブリバッファ内でＵＤＰデータグラムからの前記最大６つのパケットを列に並ばせる。ＵＤＰデータグラムに対応するパケットは、そのＩＰ・ＩＤに基づいて識別され、フラグメントオフセットに基づいて連結される。ＵＤＰ及び上位レイヤヘッダを含むパケットが処理されてデスティネーションアドレスが獲得された後、並べられた前記データをそれらのアドレスに書き込むことができる。ＵＤＰデータグラムからの全てのパケットが来ない可能性を説明するために、ＩＮＩＣ６２２は、受け取ったデータの落とし込み（dropping）をトリガするタイマを用いることができる。

【００７０】

リアルタイム音声または映像通信の場合、電気通信接続がサーバ６００とクライアント６０２との間で最初に設定される。国際電気通信連合（ＩＴＵ）Ｈ．３２３規格による通信では、電気通信接続の設定が、ＲＴＰを用いたデータの流れのためにソース及びデスティネーションＵＤＰポートを指定するＴＣＰダイヤログで実行される。電気通信接続設定の

ための別のＴＣＰダイヤログが、ＲＴＣＰを用いたデータフローをモニタするためにソース及びデスティネーションＵＤＰポートを指定する。ＳＩＰによって電気通信接続を初期設定するための別のメカニズムが提供される。ＵＤＰソケットが指定された後、本発明によるサーバ６００からクライアント６０２への音声または映像データの伝送を開始することができる。

【００７１】

例えば、オーディオ／ビデオインタフェース６６６は、マイク６７０及びカメラ６７４からの音声及びイメージを、ＩＮＩＣ６２２に利用可能なオーディオ／ビデオデータに変換する。サーバ６００ファイルシステムの指示のもとで、ＩＮＩＣ６２２は、ＲＴＰヘッダを含むＡＶデータの８ＫＢファイルストリームを獲得し、かつそのデータをＩＮＩＣファイルキャッシュに格納することができる。電気通信接続に従って、ＩＮＩＣ６２２はＵＤＰヘッダを各ファイルストリームにプリペンドすることができ、次にこれが１．５ＫＢのフラグメントに断片化され、そのそれぞれがＩＰ及びイーサネットヘッダでプリペンドされて、ネットワーク６０４上に伝送される。これに変えて、ＩＮＩＣ６２２はＩＮＩＣメモリ上に格納されたＡＶデータから１．５ＫＢのＵＤＰデータグラムを作成することができ、断片化を回避しかつＵＤＰ及びＩＰ及びＭＡＣレイヤヘッダを同時に前記電気通信接続に対応するＩＮＩＣ６２２上に保持されたテンプレートから作成できるようにする。また、ＡＶデータで８ＫＢより大きいＵＤＰデータグラムを作成することが可能であり、それが追加の分割を作成するが、データグラムごとにより大きなブロックのデータを転送することができる。ＲＴＰヘッダは、ＡＶデータがオーディオ／ビデオインタフェース６６６によりパケット化される相対時間を表示するタイムスタンプを有する。

【００７２】

ホストＣＰＵによるＡＶデータの従来のプロトコルプロセッシングと対照的にＩＮＩＣ６２２は、ＵＤＰヘッダ作成、ＩＰ断片化及びＩＰヘッダ作成のタスクをオフロードすることができ、ホストＣＰＵの複数のプロセッシングサイクル、複数の割り込みを不要にしかつホストパストラフィックを大幅に少なくすることができる。ＩＮＩＣ６２２は、複数のヘッダを１度にプリペンドすることによって、より効率的にヘッダ作成タスクを実行することができる。更に、ＡＶデータは、最初にＩ／Ｏバス６３９上をプロトコルプロセッシングスタックへ送られかつ次にＲＴＰ／ＵＤＰ／ＩＰパケットとしてＩ／Ｏバス６３９上を送り返されるよりはむしろ、Ｉ／Ｏバス６３９上でＩＮＩＣ６２２により直接アクセスすることができる。またこれらの効率上の利点は、従来のプロトコルプロセッシングに比較して、ＡＶデータの送信における遅れを大幅に小さくすることができる。

【００７３】

リアルタイム音声及び映像通信システムの対象は、通信における遅れやジッタが該システムを介して通信する人々に実質的に知覚できないことである。ジッタは、パケットを受信する遅れの変化によって生じ、これが、音や景色が記録されたテンポと比較してそれらが表示されるテンポに変動を生じさせることになる。上述したパケット化された音声及び／または映像を伝送する際の遅れを減少させる利点は、受信装置でのバッファリングを介してジッタもまた減少させるのに用いることができるが、これは遅れの減少によってジッタを円滑化するための受信装置における時間が増加するからである。ＡＶデータを通信する際の遅れ及びジッタを更に減らすために、ＩＮＩＣ６２２によりＵＤＰデータグラムフラグメントにプリペンドされたＩＰまたはＭＡＣレイヤヘッダは、そのサービス分野での高品質のサービス（ＱＯＳ）表示を含んでいる。この表示は、ネットワーク６０４により用いられて、例えば優先的にバンド幅を割り当てかつそれらのフレームについてキューに入れることによって、高ＱＯＳフレームの伝送を早めることができる。

【００７４】

ＡＶデータを含むフレームがＩＮＩＣ６０６により受信されると、ＱＯＳの表示がＩＮＩＣ６０６受信論理によってフラグを立てられ、かつパケットがＩＮＩＣ６０６の高い優先順位の受信キューにバッファされる。ＩＮＩＣ６０６のハードウェア論理によってパケットのヘッダを分類しかつ検証することによって、従来のＩＰ／ＵＤＰヘッダのプロセッシ

ングに比較して、ＡＶデータを受け取る際の遅れが減少する。各データグラムのＵＤＰヘッダはＡＵＤＰレイヤに送ることができ、これが前記ヘッダを処理して、関連するソケットをアプリケーションに表示する。次に、前記アプリケーションはＩＮＩＣ６０６に指示して、そのＵＤＰヘッダに対応するデータをオーディオ／ビデオインタフェース６７７のデスティネーションに送る。これに代えて、ＵＤＰデータグラムに対応する全てのフラグメントが前記受信キューに連結された後、前記データグラムをＵＤＰヘッダ無しでファイルストリームとして前記ソケットに対応するオーディオ／ビデオインタフェース６７７のデスティネーションに書き込まれる。任意によって、ＵＤＰデータグラムは最初にＩＮＩＣメモリの別の部分に格納することができる。ＡＶデータ及びＲＴＰヘッダは、次にオーディオ／ビデオインタフェース６７７に送られ、そこで復号されてスピーカ６７８及びディスプレイ６８２で再生される。

【００７５】

ＡＶデータフロー全体及び受信したＩＰ／ＵＤＰパケットの大部分のプロトコルプロセッシングはＩＮＩＣ６０６により取扱うことができ、クライアント６０２のＣＰＵの複数のプロフェッシングサイクルを省略してホストバストラフィック及び割り込みを大幅に減少できることに注意する。更に、ＩＮＩＣ６０６の専用ハードウェア論理によって、汎用ＣＰＵよりもより効率的にヘッダを分類することができる。また、ＡＶデータは、ＩＮＩＣ６０２により、最初にＩ／Ｏバス６７５上でプロトコルプロセッシングスタックに送られかつ次にＩ／Ｏバス６７５上でオーディオ／ビデオインタフェース６７７に送られるのではなく、Ｉ／Ｏバス６７５上をオーディオ／ビデオインタフェース６７７に直接提供することができる。

【００７６】

図１６は、ネットワークインタフェース、ストレージコントローラ及びプロトコルプロセッサの機能を１つのＡＳＩＣチップ７０３に結合させたＩＮＩＣ２２の図を示している。本実施例におけるＩＮＩＣ２２は、サーバ及びネットワークストレージアプリケーションのための高速プロトコルプロセッシング用に設計された全２重４チャネル１０／１００メガビット毎秒（Ｍｂｐｓ）のインテリジェントネットワークインタフェースコントローラが提供される。また、ＩＮＩＣ２２は、パーソナルコンピュータ、ワークステーション、ルータまたはＴＣＰ／ＩＰ、ＰＰＣＰ／ＩＰまたはＳＰＸ／ＩＰＸプロトコルが使用されている他のホストに接続することができる。このようなＩＮＩＣの説明は、本願明細書の冒頭部分に挙げた関連特許出願に記載されている。

【００７７】

ＩＮＩＣ２２はネットワークコネクタによって４つのネットワークライン７０２、７０４、７０６、７０８に接続され、これらはツイストペア同軸ケーブルまたは光ファイバのような、それぞれの接続によって、米国カリフォルニア州９４５３８、フリーモント、レイサイドパークウェイ４７２００のＳＥＥＱ　Ｔｅｃｈｎｏｌｏｇｙ　Ｉｎｃｏｒｐｏｒａｔｅｄから市販されているモデル８０２２０／８０２２１イーサネットメディアインタフェースアダプタのような物理的レイヤチェック７１２、７１４、７１６、７１８を介して媒体独立インタフェース（ＭＩＩ）を提供する多数の異なるコンジットに沿ってデータを運ぶことができる。前記ラインは好ましくは８０２．３コンプライアントでありかつＩＮＩＣに関連して４つの完全なイーサネットノード、ＩＮＩＣ支援の１０ベースＴ、１０ベースＴ２、１００ベースＴＸ、１００ベースＦＸ及び１００ベースＴ４に加えて将来のインタフェース標準を構成する。物理的レイヤの識別及び初期設定が、ホストドライバの初期設定ルーチンを介して実行される。ネットワークライン７０２、７０４、７０６、７０８とＩＮＩＣ２２との間の接続は、ＭＡＣサブレイヤの基本機能を実行し、基本的にＩＮＩＣがネットワークライン７０２、７０４、７０６、７０８にアクセスする時に制御する論理回路を含むＭＡＣ装置ＭＡＣ－Ａ７２２、ＭＡＣ－Ｂ７２４、ＭＡＣ－Ｃ７２６、ＭＡＣ－Ｄ７２８によって制御される。ＭＡＣ装置７２２、７２４、７２６、７２８は、無差別モード、マルチキャストモードまたはユニキャストモードで動作することができ、ＩＮＩＣがネットワークモニタとして機能し、ブロードキャストパケット及びマルチキャス

トパケットを受信し、かつ各ノードについて複数のＭＡＣアドレスを実行できるようにする。また、ＭＡＣ装置７２２、７２４、７２６、７２８によって、単一のネットワーク管理プロトコル（ＳＮＭＰ）について使用可能な統計的情報が得られる。

【００７８】

ＭＡＣ装置７２２、７２４、７２６、７２８はそれぞれ、送受信シーケンサＸＭＴ＆ＲＣＶ－Ａ７３２、ＸＭＴ＆ＲＣＶ－Ｂ７３４、ＸＭＴ＆ＲＣＶ－Ｃ７３６、ＸＭＴ＆ＲＣＶ－Ｄ７３８に接続される。各送受信シーケンサは、メッセージフレームがそのシーケンサを通過する時、幾つかのプロトコルプロセッシングのステップをオンザフライで実行することができる。ＭＡＣ装置と組み合わせて、送受信シーケンサ７３２、７３４、７３６、７３８は、ハードウェアにおけるデータリンク、ネットワーク、トランスポート、セッション及び適当な場合にはプレゼンテーション及びアプリケーションレイヤのプロトコルに関するパケットステータスを集めることができ、従来のシーケンシャルソフトウェアエンジンと比較してそのようなプロトコルプロセッシングのための時間を大幅に減少させることができる。送受信シーケンサ７３２、７３４、７３６、７３８はＳＲＡＭ及びＤＭＡコントローラ７４０に接続され、それらはＤＭＡコントローラ７４２及びＳＲＡＭ７４４を有し、これがスタティックランダムアクセスメモリ（ＳＲＡＭ）バッファ７４８を制御する。ＳＲＡＭ及びＤＭＡコントローラ７４０は外部メモリコントロール７５０と対話して、外部メモリバス７５２を介して、ＩＣチップ７００の近くに配置されたダイナミックランダムアクセスメモリ（ＤＲＡＭ）バッファ７５５との間でフレームを送りかつ受信する。ＤＲＡＭバッファ７５５は、４ＭＢ、８ＭＢ、１６ＭＢまたは３２ＭＢとして構成することができ、かつ任意によりチップ上に配置することができる。ＳＲＡＭ及びＤＭＡコントローラ７４０は、本実施例ではＩＮＩＣ２２とＰＣＩインタフェースバス７５７との間のインタフェースを管理するＰＣＩバスインタフェース装置（ＢＩＵ）７５６であるＩ／Ｏブリッジに接続されている。６４ビットの多重化ＢＩＵ７５６によって、スレイブ及びマスタ機能双方についてＰＣＩバス７５７との直接のインタフェースが得られる。ＩＮＩＣ２２は６４ビットまたは３２ビットのＰＣＩ環境で動作することができ、そのいずれの構成においても６４ビットのアドレス指定をサポートする。

【００７９】

マイクロプロセッサ７８０がＳＲＡＭ及びＤＭＡコントローラ７４０とＰＣＩ・ＢＩＵ７５６に接続されているマイクロプロセッサ７８０の命令及びレジスタファイルはＳＲＡＭの書込み可能なオンチップコントロールストア（ＷＣＳ）及びリードオンリメモリ（ＲＯＭ）を有するオンチップコントロールストア７８０に存在する。マイクロプロセッサ７８０は、入力するフレームを処理し、ホストコマンドを処理し、ネットワークトラフィックを指示しかつＰＣＩバストラフィックを指示することができるプログラマブルステートマシンを提供する。３つのプロセッサが、クロックサイクルごとに１つの命令を発しかつ完了する３段階パイプラインアーキテクチュアの共有ハードウェアを用いて実行される。受信プロセッサ７８２は、基本的に通信を受け取るために用いられるのに対し、送信プロセッサ７８４は、基本的に全２重通信を容易にする目的で通信を送るために用いられ、ユーティリティプロセッサ７８６は、ＰＣＩレジスタのアクセスを監督しかつ制御することを含む様々な機能を提供する。

【００８０】

プロセッサ７８２、７８４、７８６のための命令がオンチップコントールストア７８１に存在するので、前記３つのプロセッサの機能は容易に再定義することができ、それによってマイクロプロセッサ７８０は所定の環境に適応することができる。例えば、受信機能に必要なプロセッシングの量が送信またはユーティリティ機能に必要なそれを上回ることがある。そのような場合、幾つかの受信機能が送信プロセッサ７８４及び／またはユーティリティプロセッサ７８６によって実行することができる。それに代えて、追加レベルのパイプライン方式を作って３つではなく４つまたはそれ以上の仮想プロセッサを設けることができ、その追加レベルに受信機能が指示される。

【００８１】

本実施例のＩＮＩＣ２２は、ＤＲＡＭ７５５のテーブルに保持される最大２５６のＣＣＢをサポートすることができる。また、しかしながら、順次検索を省略するためにＳＲＡＭ７４８には順番がハッシュ状態のＣＣＢインデックスが存在する。一旦ハッシュが生成されると、ＣＣＢがＳＲＡＭにキャッシュされ、本実施例では６０までのＣＣＢがＳＲＡＭにキャッシュされる。ＳＲＡＭにキャッシュされた６０のＣＣＢの割り当ては、以下に説明するように、最小使用頻度のレジスタによって操作される。これらキャッシュの配置は、送信プロセッサ７８４と受信プロセッサ７８６との間で共有され、それによってより大きなロードを有するプロセッサがより多くのキャッシュバッファを使用することができる。また、前記シーケンサ間で共有されるべき８つのヘッダバッファ及び８つのコマンドバッファが存在する。所定のヘッダまたはコマンドバッファは、特定のＣＣＢバッファに統計的にリンクされないが、そのリンクは各フレームごとにダイナミックだからである。

【００８２】

図１７はパイプラインマイクロプロセッサ７８０の概観を示しており、受信、送信及びユーティリティプロセッサへの命令がクロックインクリメントＩ、II及びIIIによって交互に変わるそれぞれにパイプラインの段階に対応する３つのフェーズで実行される。各フェーズは異なる機能について責任を有し、前記３つのプロセッサのそれぞれが各クロックインクレメントの際に異なるフェーズを占有する。各プロセッサは通常コントロールストア７８１からの異なる命令ストリームに基づいて動作し、それぞれに前記フェーズのそれぞれを通してそれ自身のプログラムカウンタ及びステータスを有する。

【００８３】

一般にパイプラインマイクロプロセッサの第１インストラクションフェーズ８００は、命令を完了しかつ結果をデスティネーションオペランドに格納し、次の命令をフェッチし、かつその次の命令をインストラクションレジスタに格納する。第１のレジスタセット７９０によって、命令レジスタを含む多数のレジスタが提供され、かつ前記第１レジスタセットのためのコントロール７９２セットによって、第１レジスタセット７９０への記憶のためのコントロールが提供される。幾つかのアイテムがコントロール７９０による変更なしに第１フェーズを通過し、かつその代わりに単に第１レジスタセット７９０またはＲＡＭファイルレジスタ８３３内にコピーされる。第２インストラクションフェーズ８６０はインストラクションデコーダ及びオペランドマルチプレクサ７９８を有し、これは一般に第１レジスタセット４９０のインストラクションレジスタ内に格納された命令をデコードし、かつ生成されている全てのオペランドを集め、これは次に第２のレジスタセット７９６のデコードレジスタに格納される。第３インストラクションフェーズ９００で使用される第１レジスタセット７９０、第２レジスタセット７９６及び第３レジスタセット８０１は、図１８Ａ～Ｃにより詳細に示すように、多数の同じレジスタを有する。インストラクションデコーダ及びオペランドマルチプレクサ７９８は、第１フェーズ８０６及び第２フェーズ８６０の双方で動作するＲＡＭファイルレジスタ８３３の２つのアドレスおよびデータポートから読み出すことができる。プロセッサ７８０の第３フェーズ９００は、演算論理装置（ＡＬＵ）９０２を有し、これは一般に第２レジスタセットからのオペランドについて全てのＡＬＵ動作を実行し、結果を第３レジスタセット８０１に含まれるリザルトレジスタに格納する。スタックエクスチェンジ８０８がレジスタスタックを順序付けすることができ、かつキューマネージャ８０３がプロセッサ７８０のキューを編成し、その結果が第３レジスタセットに格納される。

【００８４】

インストラクションは、循環パイプライン８０５に示されるように第１フェーズに続き、第３フェーズが続く。全ての所定のフェーズにおける組み合わせ遅れを最小にするために、様々な機能が命令実行の３つのフェーズに渡って分散されていることに注意する。本実施例では６６ＭＨＺの周波数で、各クロックインクレメントは１５ナノ秒で完了し、３つの前記プロセッサのそれぞれについて１つの命令を完了するのに合計４５ナノ秒を要する。インストラクションフェーズが循環する様子が図１８Ａ～Ｃにより詳細に示されており、各フェーズが異なる図で示されている。

【００８５】
より詳細に言えば、図１８Ａは、第１フェーズ８００の幾つかの特定のハードウェア機能を示しており、一般に第１レジスタセット７９０及び関連するコントロール７９２を有する。第１レジスタセット７９２のコントロールには、アドレス及び書込みデータをＳＲＡＭアドレス及びデータレジスタ８２０内にロードするための論理コントロールである、ＳＲＡＭコントロール８２０を有する。従って、第３フェーズ９００からのＡＬＵ９０２の出力は、ＳＲＡＭコントロール８０２によってＳＲＡＭアドレス及びデータレジスタ８２０のアドレスレジスタまたはデータレジスタ内に配置される。同様にロードコントロール８０４がコンテキストレジスタ８２２にファイルするファイルのためのコンテキストを書き込むためのコントロールを提供し、かつ別のロードコントロール８０６が、様々な雑データをフリップフロップレジスタ８２５に格納するためのコントロールを提供する。キャリービットが設定されているかどうかというようなＡＬＵ状態コードが、第１フェーズ８００において実行される動作無しで、ＡＬＵ状態コードレジスタ８２８内にクロックされる。フラグレコード８０８がロックを設定するような、フラグレジスタ８３０内に格納される様々な機能を実行する。
【００８６】
ＲＡＭファイルレジスタ８３３は、アドレス及びデータのための１つの書込みポートとアドレス及びデータのための２つの読出しポートとを有し、それによって２以上のレジスタが同時に読み出すことができる。上述したように、ＲＡＭファイルレジスタ３３は基本的に第１及び第２フェーズにまたがるが、それは第１フェーズ８００において書き込まれかつ第２フェーズ８６０において読み出されるからである。コントロールストアインストラクション８１０によって、この図面には記載されていないコントロールストア７８１からの新しいデータによる前記プロセッサの再プログラミングが可能になり、前記インストラクションはインストラクションレジスタ８３０のみ格納される。このためのアドレスがフェッチコントロールレジスタ８１１において生成され、それによりどのアドレスにフェッチするかが決定され、アドレスがフェッチアドレスレジスタ８３８に格納される。ロードコントロール８３１がプログラムカウンタ８４０の命令を出し、これはコントロールストアのためのフェッチアドレスに良く似たように動作する。３つのレジスタからなる後入れ先出し方式のスタック８４４が、このフェーズで他の動作を受けることなく、第１レジスタセットにコピーされる。最後にデバッグアドレス８４８のためのロードコントロール８１７が任意により含まれ、それによって起こり得るエラーの修正が可能になる。
【００８７】
図１８Ｂは、ＲＡＭファイルレジスタ８３３からのアドレス及びデータの読出しを含む第２マイクロプロセッサフェーズ８６０を示している。スクラッチＳＲＡＭ８６５が、最初の２つのフェーズを通過して第３で増分されるレジスタを有する、前記第１レジスタセットのＳＲＡＭアドレス及びデータレジスタ８２０から書き込まれる。スクラッチＳＲＡＭ８６５は、スタック８４４、デバッグアドレス８４８、及び上述したＳＲＡＭアドレス及びデータレジスタを除いて、前記第１レジスタセットからの大部分のレジスタがそうであるように、インストラクションデコーダ及びオペランドマルチプレクサ７９８によって読み出される。インストラクションレコーダ及びオペランドマルチプレクサ７９８は、レジスタセット７９０の各レジスタ及びＳＲＡＭ８６５を見て、インストラクションをデコードし、かつ次のフェーズにおける動作のためのオペランドを集め、特に以下のＡＬＵ９０２に提供するオペランドを決定する。インストラクションデコーダ及びイペランドマルチプレクサ７９８の成果は、ＡＬＵオペランド８７９、８８２、ＡＬＵ状態コードレジスタ８８０、及び本実施例では３２のキューを制御することができるキューチャネル及びコマンド８８７レジスタを含む第２レジスタセット７９６の多数のレジスタに格納される。レジスタセット７９６のレジスタの幾つかは、デコーダ７９８により実質的にデコードすることなく、上述したインストラクションレジスタ８３５からほぼ直接にロードされ、プログラムコントロール８９０、リテラルフィールド８８９、テストセレクト８８４及びフラグ８８５が含まれる。第１フェーズ８００のファイルコンテキスト８２２のような他のレ

ジスタは、常に第２フェーズ８６０のファイルコンテキスト８７７に格納されるが、マルチプレクサ８７０により集められたオペランドとして取扱うこともできる。スタックレジスタ８４４は、単にスタックレジスタ８９４にコピーされる。プログラムカウンタ８４０はこのフェーズで増分され８６８、かつレジスタ８９２に格納される。また、任意のデバッグアドレス８４８が増分され、８７０、かつロードコントロール８７５が各フェーズでのエラーを制御できるようにする目的で、この時点でパイプライン８０５から供給することができ、その結果がデバッグアドレス８９８に格納される。

【００８８】

図１８Ｃは、ＡＬＵ及びキュー動作を含む第３のマイクロプロセッサフェーズ９００を示している。ＡＬＵ９０２は加算器、優先順位エンコーダ及び他の標準的な論理機能を有する。ＡＬＵの結果はレジスタＡＬＵ出力９１８、ＡＬＵ状態コード９２０及びデスティネーションオペランドリザルト９２２に格納される。ファイルコンテキストレジスタ９２５、フラグセレクトレジスタ９２６及びリテラルフィールドレジスタ９３０は、先のフェーズ８６０から単にコピーされるだけである。テストマルチプレクサが設けられて、条件付ジャンプが結果としてジャンプになるかどうかが決定され、その結果がテストリザルトレジスタ９２４に格納される。テストマルチプレクサ９０４は、そうではなくフェッチコントロール８１１のような類似の決定と共に第１フェーズ８００で実行することができる。スタックエクスチェンジ８０８が、プログラムカウンタをスタック８７９からフェッチし、またはそのスタック上にプログラムカウンタを置くことによってスタックを上下にシフトさせ、その結果がプログラムコントロール９３４、プログラムカウンタ９３８及びスタック９４０レジスタに格納される。ＳＲＡＭアドレスは、任意によりこのフェーズ９００で増分させることができる。このフェーズでもエラーを制御できるようにするために、別のデバッグアドレス９４２の別のロードコントロール９１０をこの時点でパイプライン８０５から強制することができる。この図面では一体に表されているＱＲＡＭ＆ＱＡＲＵ９０６が、キューチャネル及びコマンドレジスタ８８７から読出し、ＳＲＡＭに格納し、かつキューを再編成し、データのキューを管理するのに必要なようにデータ及びポインタを追加しまたは取り外し、結果をテストマルチプレクサ９０４及びキューフラグ及びキューアドレスレジスタ９２８に送る。このようにＱＲＡＭ＆ＱＡＲＵ９０６は、従来はＣＰＵ上のソフトウェアにより順に実行されていたタスク、前記３つのプロセッサのためのキューを管理する義務を引き受け、代わりにキューマネージャ９０６によって加速されかつ実質的にパラレルなハードウェアキューイングが行なわれる。

【００８９】

図１９は、キューマネージャ９０６によって管理される３２のハードウェアキューの内の２つを示しており、前記キューのそれぞれはＳＲＡＭヘッド、ＳＲＡＭ末尾及びＴＲＡＭ本体内で情報を列に並べる能力をも有し、各キューの拡張及び個々の構成を可能にしている。このように、ＦＩＦＯ１０００はＳＲＡＭ記憶装置１００５、１００７、１００９、１０１１を有し、それぞれが８バイトで合計３２バイトを有するが、これら装置の数及び能力は、他の実施例において変えることができる。同様に、ＦＩＦＯ１００２はＳＲＡＭ記憶装置１０１３、１０１５、１０１７、１０１９を有する。ＳＲＡＭ装置１００５、１００７はＦＩＦＯ１０００のヘッドであり、かつ装置１００９、１０１１はそのＦＩＦＯの末尾であるのに対し、装置１１３、１１５はＦＩＦＯ１００２のヘッドでありかつ装置１１７、１１９はそのＦＩＦＯの末尾である。ＦＩＦＯ１０００の情報は矢印１０２２で示すように、ヘッドの装置１００５または１００７内に書き込むことができ、かつ矢印１０２５で示すように、末尾の装置１０１１または１００９から読み出される。しかしながら、特定の項目はヘッドの装置１００５または１００７への書込み及び読出し両方を行なうことができ、または末尾の装置１００９または１０１１への書込み及び読出し双方を行なうことができ、データの移動及び待ち時間を最小にする。同様に、ＦＩＦＯ１００２の情報は一般に、矢印１０３３で示すように、ヘッドの装置１０１３または１０１５に書き込まれ、かつ矢印１０３９で示すように、末尾の装置１０１７または１０１９から読み出されるが、そうではなく、書き込まれた同じヘッドまたは末尾の装置から読み出すことが

できる。

【００９０】

ＳＲＡＭ・ＦＩＦＯ１０００、１００２は双方共ＤＲＡＭ７５５に接続されており、それらＦＩＦＯが実質的に無限に拡張してＳＲＡＭヘッド及び末尾部分が一杯であるような状況を処理できるようにしている。例えば、Ｑ０とラベル付けされた３２のキューの最初のものは、矢印１０２７で示すように、ＦＩＦＯ７００のヘッド部分または末尾部分で待ち行列に入れられる代わりに、キューマネージャの指示下で動作するＤＭＡ装置によって、ＤＲＡＭ７５５のエントリを待ち行列に入れることができる。ＤＲＡＭ７５５に格納されたエントリは、矢印１０３０で示すように、ＳＲＡＭ装置１００９に戻り、そのＦＩＦＯの長さ及びフォールスルー時間を延長させる。ＳＲＡＭからＤＲＡＭへの転換は、ＤＲＡＭの方がより遅くかつＤＭＡの動作が別の待ち時間を生じさせることから、一般にＳＲＡＭが一杯である場合のために取っておかれる。このようにＱ０は、キューマネージャ８０３によってＦＩＦＯ１０００及びＤＲＡＭ７５５の双方に格納されたエントリで構成することができる。同様に、例えばＱ２７に対応するＦＩＦＯ１０００に行きの情報は、矢印１０３５で示すように、ＤＭＡによってＤＲＡＭ７５５内に動かすことができる。遅いとはいえコスト効果のよいＤＲＡＭ８０３で待ち行列に入れる能力は、初期設定の際にユーザ定義可能なものであり、前記キューを所望のサイズに変更できるようにしている。ＤＲＡＭ７５５にキューされた情報は、矢印１０３７で示すように、ＳＲＡＭ装置１０１７に戻される。

【００９１】

３２個のハードウェアキューのそれぞれのステータスが、図２０に示すように、４つの３２ビットのレジスタからなる組み１０４０に保持されかつそれからアクセスされ、各レジスタの特定のビットが特定のキューに対応していると好都合である。前記レジスタには、Ｑ－Ｏｕｔ＿Ｒｅａｄｙ１０４５、Ｑ－Ｉｎ＿Ｒｅａｄｙ１０５０、Ｑ－Ｅｍｐｔｙ１０５５及びＱ－Ｆｕｌｌ１０６０のラベルが付されている。特定のビットがＱ－Ｏｕｔ＿Ｒｅａｄｙレジスタ１０５０で設定されると、そのビットに対応するキューは、読み出される準備のできた情報を有するのに対し、同じビットのＱ－Ｉｎ＿Ｒｅａｄｙ１０５２レジスタにおける設定は、前記キューが書き込まれる準備ができていることを意味している。同様に、特定のビットのＱ－Ｅｍｐｔｙレジスタ１０５５の肯定的設定がそのビットに対応するキューが空であることを意味するのに対し、特定のビットのＱ－Ｆｕｌｌレジスタ１０６０における肯定的設定は、そのビットに対応するキューが一杯であることを意味する。このようにＱ－Ｏｕｔ＿Ｒｅａｄｙ１０４５は、ビット２７・１０５２、２８・１０５４、２９・１０５６及び３０・１０５８を含むビット０・１０４６から３１・１０４８を有する。Ｑ－Ｉｎ＿Ｒｅａｄｙ１０５０は、、ビット２７・１０６６、２８・１０６８、２９・１０７０及び３０・１０７２を含めて、ビット０・１０６２～３１・１０６４を有する。Ｑ－Ｅｍｐｔｙ１０５５は、ビット２７・１０７８、２８・１０８０、２９・１０８２及び３０・１０８４を含めて、ビット０・１０７４～３１・１０７６を有し、Ｑ－Ｆｕｌｌ１０７０は、ビット２７・１０９０、２８・１０９２、２９・１０９４及び３０・１０９６を含めて、ビット０・１０８６～３１・１０８８を有する。

【００９２】

ＦＩＦＯ１０００に対応するＱ０はフリーバッファキューであり、全ての使用可能なバッファのアドレスのリストを保持している。このキューは、前記マイクロプロセッサまたは他のデバイスがフリーバッファのアドレスを必要とする時にアドレス指定され、従って一般に分かる程度のＤＲＡＭ７５５を有する。従って、フリーバッファのアドレスを必要とするデバイスは、Ｑ－０を調べてそのアドレスを得ることになる。ＦＩＦＯ１００２に対応するＱ－２７は、受信バッファ記述子のキューである。受け取ったフレームを受信シーケンサにより処理した後、シーケンサはＱ－２７に前記フレームの記述子を格納すると思われる。そのような記述子のロケーションがＳＲＡＭにおいてすぐに使用可能な場合には、Ｑ－Ｉｎ＿Ｒｅａｄｙ１０５０のビット２７・１０６６が設定されることになる。そうでない場合には、前記シーケンサはキューマネージャがＳＲＡＭからＤＲＡＭへＤＭＡの

移動を開始させるのを待たなければならず、それにより空間を解放して受信記述子を格納する。

【００９３】

ＳＲＡＭと前記プロセッサ間、送信シーケンサと受信シーケンサ間、及びＳＲＡＭとＤＲＡＭ間におけるキューエントリの動きを管理するキューマネージャの動作が、図２１により詳細に示されている。前記キューを利用するリクエストは、プロセッサリクエスト１１０２、送信シーケンサリクエスト１１０４、及び受信シーケンサリクエスト１１０６を含む。前記キューへの他のリクエストは、ＤＲＡＭ対ＳＲＡＭリクエスト１０８６及びＳＲＡＭ対ＤＲＡＭリクエスト１１１０であり、これはＤＲＡＭと前記キューのＳＲＡＭヘッド部分または末尾部分との間でデータを行ったり来たり動かす際にキューマネージャのために動作する。これら様々なリクエストのどれが次のサイクルでキューマネージャを使用することになるかを決定することは、優先順位論理アービタ１１１５によって処理される。高周波数動作を可能にするために、キューマネージャはパイプライン化され、レジスタＡ１１１８及びレジスタＢ１１２０によって一時的ストレージが提供されるのに対し、ステータスレジスタ１１１２は、次の更新までステータスを維持する。キューマネージャは、送受信シーケンサが要求するＤＭＡについて偶数サイクルを、プロセッサのリクエストのために奇数サイクルを確保する。デュアルコープのＱＲＡＭ１１２５が前記キューのそれぞれに関する変数、前記キューのＳＲＡＭ状態に対応するヘッドライトポインタ、ヘッドリードポインタ、テールライトポインタ及びテールリードポインタ、並びに前記キューのＤＲＡＭ状態及び前記キューのサイズに対応するボディライトポインタ及びボディリードポインタを含む各キューの変数を格納する。

【００９４】

アービタ１１１５が実行すべき次の動作を選択した後ＱＲＡＭ８２５の変数はＱＡＬＵ１１２８により選択された動作に従って設置されかつ変更され、ＳＲＡＭ読出しリクエスト１１３０またはＳＲＡＭ書込みリクエスト１１４０が生成される。前記変数が更新されかつ更新されたステータスがＱＲＡＭ１１２５に加えてステータスレジスタ１１２２に格納される。前記ステータスはまたアービタ１１１５に供給されて、先に要求された動作が完了したという信号を送り、リクエストの重複を抑制する。ステータスレジスタ１１２２は、４つのキューレジスタＱ－Ｏｕｔ＿Ｒｅａｄｙ１０４５、Ｑ－Ｉｎ＿Ｒｅａｄｙ１０５０、Ｑ－Ｅｍｐｔｙ１０５５及びＱ－Ｆｕｌｌ１０６０を更新して、アクセスされた前記キューの新しいステータスを反映する。同様に、ＤＭＡを介してそのキューについてＳＲＡＭヘッド部分及び末尾部分にアクセスしかつそれからアクセスされたＳＲＡＭアドレス１１３３、ボディ書込みリクエスト１１３５及びボディ読出しリクエスト１１３８が更新される。これに代えて、Ｑ書込みデータ１１４４で示すように、マルチプラクサ１１４６により選択されかつＳＲＡＭ書込みリクエスト１１４０にパイプライン化されたキューに様々な処理を書き込みたい場合がある。ＳＲＡＭコントローラは、アクセスされた前記キューの末尾部分に書き込みまたはヘッド部分を読み出し、かつ応答を返すことによって、読出し及び書込みリクエストを処理する。このようにして、様々なキューが用いられかつそれらのステータスが更新される。

【００９５】

図２２Ａ～Ｄは、どのコンテキストまたはＣＣＢをＩＮＩＣキャッシュメモリ内に保持するかを選択するために使用される最小使用頻度レジスタ１２００を示している本実施例のＩＮＩＣは、所定の時間で最大６０のＣＣＢをＳＲＡＭにキャッシュすることができ、従って新しいＣＣＢがキャッシュされると、古いものは多くの場合捨てなければならず、捨てられるＣＣＢは通常このレジスタ１２００によって、その時点で最も使用されていないＣＣＢが選択される。本実施例では、最大２５６のＣＣＢのハッシュテーブルがＳＲＡＭに保持されるのに対し、最大２５６の満杯のＣＣＢがＤＲＡＭに保持される。最小使用頻度レジスタ１２００は、Ｒ０～Ｒ１５とラベル付けされた１６の４ビットブロックを有し、そのそれぞれがＳＲＡＭキャッシュ装置に対応している。初期設定時、前記ブロックには０～１５の番号が付され、番号０が前記最小使用頻度（ＬＲＵ）キャッシュ装置を表す

ブロックに任意に格納され、かつ番号１５が最大使用頻度（ＭＲＵ）キャッシュ装置を表すブロックに格納される。図２２Ａは、ＬＲＵブロックＲ０が番号９を保持し、かつＭＲＵブロックＲ１５が番号６を保持する任意の時点におけるレジスタ１２００を示している。現在ＳＲＡＭに保持されている別のＣＣＢをキャッシュしようとする場合、ＬＲＵブロックＲ０が読み出されこれは図２２Ａにおいて番号９を保持し、かつ新しいＣＣＢが番号９に対応するＳＲＡＭキャッシュ装置に格納される。番号９に対応する新しいＣＣＢはこの時点で最大使用頻度ＣＣＢであるから、番号９が、図２２Ｂに示すように、ＭＲＵブロックに格納される。他の番号は、番号１をＬＲＵブロックに残して、全て１レジスタブロックずつ左へシフトされる。番号９に対応してＳＲＡＭ装置に以前キャッシュされていたＣＣＢは、速度は遅いがよりコスト効率的なＤＲＡＭに移動される。

【００９６】

図２２Ｃは、使用された次のＣＣＢがすでにＳＲＡＭにキャッシュされた結果を示している。この実施例では、番号１０に対応するＳＲＡＭ装置にＣＣＢがキャッシュされており、従ってそのＣＣＢを用いた後、番号１０がＭＲＵブロックに格納される。以前により多く使用されていた番号１０以外の番号のみ（レジスタブロックＲ９～Ｒ１５）が、番号１をＬＲＵブロックに残して左側へシフトされる。このようにして、ＩＮＩＣはＳＲＡＭキャッシュ内に最もアクティブなＣＣＢを保持する。

【００９７】

ある場合には、使用されているＣＣＢは、制限されたキャッシュメモリ内に保持することが望ましくないものである。例えば、閉じつつあることが分かっているコンテキストについてＣＣＢをキャッシュしないことが好ましく、従って他のキャッシュされたＣＣＢをＳＲＡＭにより長く留めることことができる。この場合には、デキャッシュ可能なＣＣＢを保持するキャッシュ装置を表す番号が、ＭＲＵブロックＲ１５ではなくむしろＬＲＵブロックＲ０に格納され、従ってデキャッシュ可能なＣＣＢは、ＬＲＵブロックＲ０に保持される番号に対応するＳＲＡＭ装置内にキャッシュされた新しいＣＣＢの使用の直後に置き換えられることになる。図２２Ｄは、番号８（図２２Ｃにおいてブロック Ｒ９にあったもの）が使用されかつその後閉じることになっているＣＣＢに対応する場合を示している。この場合、番号８はブロックＲ９から取り外され、かつＬＲＵブロックＲ０内に格納される。先にブロックＲ９の左側に格納されていた全ての数（Ｒ１～Ｒ８）は、ここで１ブロック右側へシフトされる。

【００９８】

図２３は、最小使用頻度レジスタ１２００を動作するのに用いられる論理装置のいくつかを示している。１６個の３または４入力マルチプレクサ１２１０のアレイ、その内のマルチプレクサＭＵＸ０、ＭＵＸ７、ＭＵＸ８、ＭＵＸ９及びＭＵＸ１５のみが簡単にするために示されているが、最小使用頻度レジスタ１２００の対応する１６個のブロックに接続された出力を有する。例えば、ＭＵＸ０の出力はブロックＲ０に格納され、ＭＵＸ７の出力はブロックＲ７に格納される等である。各レジスタブロックの値は、ブロック番号をシフトする際に使用するために、その対応するマルチプレクサの入力及び両側に隣接するマルチプレクサの入力に接続されている。例えば、Ｒ８に格納される番号はＭＵＸ７、ＭＵＸ８及びＭＵＸ９の入力に供給される。ＭＵＸ０及びＭＵＸ１５はそれぞれ隣接するブロックを１つだけ有し、それらマルチプレクサの特別の入力がそれぞれＬＲＵ及びＭＲＵブロックの選択のために使用される。ＭＵＸ１５は４入力マルチプレクサとして図示されており、入力１２１５がＲ０に格納された番号を供給する。

【００９９】

１６個のコンパレータ１２２０からアレイはそれぞれ、最少使用頻度レジスタ１２００の対応するブロックに格納された値を受け取る。また、各コンパレータは、ライン１２３５に沿ってプロセッサ４７０から信号を受け取り、それによってプロセッサ４７０により送られたものと整合する番号を有するレジスタブロックが、論理回路１２３０に真を出力するのに対し、他の１５個のコンパレータが偽を出力する。論理回路１２３０が、マルチプレクサへの入力を選択しかつ従ってレジスタブロック番号のシフトを制御するために、

前記マルチプレクサのされぞれに繋がる一対の選択ラインを制御する。このように、選択ライン１２３９がＭＵＸ０を制御し、選択ライン１２４４がＭＵＸ７を制御し、選択ライン１２４９がＭＵＸ８を制御し、選択ライン１２５４がＭＵＸ９を制御し、かつ選択ライン１２５９がＭＵＸ１５を制御する。

【０１００】

ＣＣＢを使用しようとする時、プロセッサ４７０は、ＣＣＢが現在１６個のキャッシュ装置の１つに保持されているＣＣＢと整合するかどうかを調べる。整合が認められると、前記プロセッサは、そのキャッシュ装置に対応するブロック番号、例えば番号１２と共に信号をライン１２３５に沿って送る。コンパレータ１２２０は、そのライン１２３５からの前記信号をブロック番号と比較し、かつコンパレータＣ８が前記信号に整合するブロックＲ８について真を出力するのに対し、他の全てのコンパレータは偽を出力する。論理回路１２３０は、プロセッサ４７０からの制御下で、選択ライン１２５９を用いて、ＭＵＸ１５についてライン１２３５からの入力を選択し、番号１２をＭＲＵブロックＲ１５に格納する。また、論理回路１２３０は、ＭＵＸ１５は別にして、ＭＵＸ８及びそれより上位のマルチプレクサに関する各対の選択線に信号を送り、ＭＵＸ８及びそれより上位の各マルチプレクサへの入力として、１ブロック右側のレジスタブロック（Ｒ９～Ｒ１５）に格納されていた値を選択することによって、その出力を１ブロック左側へシフトする。ＭＵＸ８の左側にあるマルチプレクサの出力は、一定に選択される。

【０１０１】

他方、プロセッサ４７０が１６個のキャッシュ装置の中で前記ＣＣＢとの整合を見つけることができない場合には、前記プロセッサはライン１２６６に沿ってＬＲＵブロックＲ０から読出しを行い、ＬＲＵブロックに対応するキャッシュを識別し、かつそのキャッシュに格納されていたデータをＤＲＡＭに書き込む。Ｒ０に格納されていた番号、本実施例では番号３、がＭＲＵブロックＲ１５に格納するためにＭＵＸ１５への入力１２１５として選択ライン１２５９により選択される。他の１５のマルチプレクサは、そのそれぞれのレジスタブロックに、各レジスタブロックのすぐ右側に格納されていた番号を出力する。

【０１０２】

前記プロセッサがＣＣＢを使用後にキャッシュから取り外したいような場合には、ＭＲＵブロックＲ１５よりもむしろＬＲＵブロックＲ０が、そのＣＣＢを保持するキャッシュ装置に対応する番号を置くために選択される。ＳＲＡＭから取り外すためにＬＲＵブロックＲ０に置かれるべきＣＣＢに対応する数（例えば、ブロックＲ９に保持される番号１）は、プロセッサ４７０によりライン１２３５に沿って送られ、これはコンパレータＣ９によって整合される。前記プロセッサは論理回路１２３０に指示して、ライン１２３９でＭＵＸ０への入力１２３５を選択することによって、番号１をＲ０に入力させる。ＭＵＸ９への選択ライン１２５４が入力として、レジスタブロックＲ８に保持されている番号を選択し、それによってＲ８からの番号がＲ９に格納される。Ｒ０とＲ９間の他のレジスタブロックにより保持されている番号が同様にして右側へシフトされ、Ｒ９の右側のレジスタブロックの番号が一定のまま残される。これによって、多くのサイクルについて閉じたＣＣＢを保持することから不十分なキャッシュメモリが解放され、それらの識別番号がレジスタブロックを介してＭＲＵからＬＲＵブロックへ移動する。

【０１０３】

図２４は、ＩＮＩＣの詳細を追加して示しており、この記載では単一のネットワークシステムに焦点を置いている。ＩＮＩＣ２２は、ＰＨＹチップ７１２、ＡＳＩＣチップ７００及びＤＲＡＭ７５５を有する。ＰＨＹチップ７１２はＩＮＩＣカード２２をネットワークコネクタ２１０１を介してネットワークライン２１０５に接続している。ＩＮＩＣ２２は、カードエッジコネクタ２１０７及びＰＣＩバス７５７を介してホストのＣＰＵ（例えば、図１のホスト２０のＣＰＵ３０）に接続されている。ＡＳＩＣチップ７００は、媒体アクセスコントロール（ＭＡＣ）装置７２２、シーケンサブロック７３２、ＳＲＡＭコントロール７４４、ＳＲＡＭ７４８、ＤＲＡＭコントロール７４２、キューマネージャ８０３、プロセッサ７８０及びＰＣＩバスインタフェース装置７５６を有する。シーケンサブロ

ック７３２は、送信シーケンサ２１０４、受信シーケンサ２１０５、及びコンフィグレーションレジスタ２１０６を有する。ＭＡＣデスティネーションアドレスはコンフィグレーションレジスタ２１０６に格納される。プロセッサ７８０により実行されるプログラムコードの一部分がＲＯＭ（図示せず）に保有され、かつ一部分が書込み可能なコントロールストアＳＲＡＭ（図示せず）に送られている。前記プログラムは、初期設定時にホスト２０から書込み可能コントロールストアＳＲＡＭにダウンロードすることができる。

【０１０４】

図２５は、図２４の受信シーケンサ２１０５のより詳細な線図である。受信シーケンサ２１０５は、データ同期バッファ２２００、パケット同期シーケンサ２２０１、データアセンブリレジスタ２２０２、プロトコルアナライザ２２０３、パケットプロセッシングシーケンサ２２０４、キューマネージャインタフェース２２０５、及びダイレクトメモリアクセス（ＤＭＡ）コントロールブロック２２０６を有する。パケット同期シーケンサ２２０１及びデータ同期バッファ２２００はＭＡＣ７２２のネットワーク同期式クロックを利用し、受信シーケンサ２１０５の残りの部分は固定周波数のクロックを利用する。破線２２２１は、クロックのドメイン境界を示している。

【０１０５】

図２４及び図２５の受信シーケンサ２１０５の動作について、ネットワークライン７０２からのＴＣＰ／ＩＰパケットのＩＮＩＣ２２上での受信に関連して説明する。初期設定時には、プロセッサ７８０がＤＲＡＭ７５５をバッファに区分する。受信シーケンサ２１０５がＤＲＡＭ７５５のバッファを用いて、到来するネットワークパケットデータを前記パケットのステータス情報と共に格納する。プロセッサ７８０は各バッファについて３２ビットのバッファ記述子を作成する。バッファ記述子は、その関連するバッファのＤＲＡＭにおけるサイズ及びロケーションを示す。プロセッサ７８０は、これらのバッファ記述子をキューマネージャ８０３に書き込むことによって、「フリーバッファキュー」２１０８上にバッファ記述子を置く。キューマネージャ８０３は、「フリーバッファキュー」２１０８を含む複数のキューを保持する。この実施例では、様々なキューのヘッド部分及び末尾部分がＳＲＡＭ７４８に置かれ、前記キューの中間部分がＤＲＡＭ７５５に置かれる。

【０１０６】

ライン２２２９は、リクエストライン及びアドレスラインに関連するリクエストメカニズムを有する。同様にライン２２３０は、リクエストライン及びアドレスラインに関係するリクエストメカニズムを有する。キューマネージャ８０３はライン２２２９、２２３０を用いて、ＤＲＡＭからＳＲＡＭへまたはＳＲＡＭからＤＲＡＭへキュー情報を転送するリクエストを出す。

【０１０７】

受信シーケンサのキューマネージャインタフェース２２０５は、パケットプロセッシングシーケンサ２２０４による使用のために、常にフリーバッファ記述子２２０７を保持しようとする。ビット２２０８は、フリーバッファ記述子２２０７がパケットプロセッシングシーケンサ２２０４により使用可能であることを表示するレディビットである。キューマネージャインタフェース２２０５がフリーバッファ記述子を持っていない（ビット２２０８が設定されていない）場合には、キューマネージャインタフェース２２０５はリクエストライン２２０５を介してキューマネージャ８０３から１を要求する。（リクエストライン２２０９は実際、動作がキューへの書込みである場合、リクエスト、キューＩＤ、読出し／書込み信号及びデータを通信するバスである。）

応答の際、キューマネージャ８０３は「フリーバッファキュー」２１０８の末尾部分からフリーバッファ記述子を検索し、次に応答ナイン２２１０上の応答信号を介してキューマネージャインタフェース２２０５に知らせる。キューマネージャインタフェース２２０５が応答信号を受け取ると、キューマネージャインタフェース２２０５はフリーバッファ記述子２２０７をロードし、かつレディビット２２０８を設定する。前記フリーバッファ記述子がＳＲＡＭ７４８のフリーバッファキューの末尾部分にあったことから、キューマネージャインタフェース２２０５は、実際はＳＲＡＭコントロールブロック７４４の読出し

データバス２２２８からフリーバッファ記述子２２０７を受け取る。パケットプロセッシングシーケンサ２２０４は、リクエストライン２２１１を介してフリーバッファ記述子２２０７を要求する。キューマネージャインタフェース２２０５がフリーバッファ記述子２２０７を検索し、かつフリーバッファ記述子２２０７がパケットプロセッシングシーケンサによる利用が可能である場合、キューマネージャインタフェース２２０５はグラントライン２２１２を介してパケットプロセッシングシーケンサ２２０４に知らせる。この処理によって、フリーバッファ記述子はパケットプロセッシングシーケンサ２２０４による使用が可能になり、かつ受信シーケンサ２１０５が到来するパケットを処理できる状態となる。

【０１０８】

次に、ＴＣＰ／ＩＰパケットがネットワークコネクタ２１０１及び物理的レイヤインタフェース（ＰＨＹ）７１２を介してネットワークライン２１０５から受け取られる。ＰＨＹ７１２は、前記パケットをＭＡＣ７２２に媒体独立インタフェース（ＭＩＩ）パラレルバス２１０９を介して供給する。ＭＡＣ７２２は前記パケットの処理を開始し、かつパケットの開始部分が受信されていることを示す「パケットの開始」信号をライン２２１３上にアサートする。データのバイトがＭＡＣに受信されかつＭＡＣ出力２２１５において利用可能になると、ＭＡＣ７２２は「データ有効」信号をライン２２１４上にアサートする。前記「データ有効」信号を受け取ると、パケット同期シーケンサ２２０１がロード信号ライン２２２２を介してデータ同期バッファ２２００を指示し、データライン２２１５から受け取ったバイトをロードする。データ同期バッファ２２００は４バイトの深さを有する。次に、パケット同期シーケンサ２２０１がデータ同期バッファの書き込みポインタを増分させる。このデータ同期バッファの書き込みポインタは、ライン２２１６を介してパケットプロセッシングシーケンサ２２０４に利用可能となる。データライン２２１５から連続するデータのバイトは、このようにしてデータ同期バッファ２２００内に刻時される。

【０１０９】

ライン２２１９上で利用可能なデータ同期バッファ読出しポインタがパケットプロセッシングシーケンサ２２０４によって保持される。パケットプロセッシングシーケンサ２２０４は、ライン２２１６上のデータ同期バッファ書き込みポインタをライン２２１９上のデータ同期バッファ読出しポインタと比較することによって、データ同期バッファ２２００においてデータが使用可能であることを決定する。

【０１１０】

データアセンブリレジスタ２２０２は１６バイト長のシフトレジスタ２２１７を有する。このレジスタ２２１７は、一度に１バイトが順にロードされ、かつパラレルにアンロードされる。データがレジスタ２２１７にロードされると、書込みポインタが増分される。この書込みポインタが、ライン２２１８を介してパケットプロセッシングシーケンサ２２０４に利用可能となる。同様に、データがレジスタ２２１７からアンロードされると、パケットプロセッシングシーケンサ２２０４により保持された読出しポインタが増分される。この読出しポインタは、ライン２２２０を介してデータアセンブリレジスタ２２０２に利用可能である。従って、パケットプロセッシングシーケンサ２２０４は、ライン２２１８上の書込みポインタをライン２２２０上の読出しポインタと比較することによって、レジスタ２２１７に利用可能な余裕があるかどうかを決定することができる。

【０１１１】

レジスタ２２１７に利用可能な余裕があるとパケットプロセッシングシーケンサ２２０４が決定した場合、パケットプロセッシングシーケンサ２２０４はデータアセンブリレジスタ２２０２に命令して、データ同期バッファ２２００からデータのバイトをロードする。データアセンブリレジスタ２２０２は、ライン２２１８上のデータアセンブリレジスタ書込みポインタを増分し、かつパケットプロセッシングシーケンサ２２０４がライン２２１９上のデータ同期バッファ読出しポインタを増分する。レジスタ２２１７にシフトされたデータは、チェックサムを検証しかつ「ステータス」情報２２２３を生成するプロトコルアナライザ２２０３によってレジスタ出力で調べられる。

【０１１２】
ＤＭＡコントロールブロック２２０６は、６４バイト受信ＦＩＦＯ２１１０を介して情報をレジスタ２２１７からバッファ２１１４へ動かす責任を有する。ＤＭＡコントロールブロック２２０６は、６４バイトのＳＲＡＭ７４８を用いて受信ＦＩＦＯ２１１０を２つの３２バイトピンポンバッファとして実行する。ＤＭＡコントロールブロック２２０６は、書込みポインタ及び読出しポインタを用いて前記受信ＦＩＦＯを実行する。転送されるべきデータがレジスタ２２１７において利用可能でありかつ空間がＦＩＦＯ２１１０において利用可能であると、ＤＭＡコントロールブロック２２０６は、ライン２２２５を介してＳＲＡＭコントローラ７４４にＳＲＡＭ書込みリクエストをアサートする。次にＳＲＡＭコントローラ７４４がデータをレジスタ２２１７からＦＩＦＯ２１１０へ移動させ、かつライン２２２５を介して応答信号をＤＭＡコントロールブロック２２０６へアサートする。ここで、ＤＭＡコントロールブロック２２０６が受信ＦＩＦＯ書込みポインタを増分させ、かつデータアセンブリレジスタ読出しポインタを増分させる。

【０１１３】
３２バイトのデータが受信ＦＩＦＯ２１１０に預けられていると、ＤＭＡコントロールブロック２２０６はライン２２２６を介してＤＲＡＭコントローラ７４２にＤＲＡＭ書込みリクエストを示す。この書込みリクエストは、ＤＲＡＭリクエストアドレスに関して「バッファロードカウント」でＯＲ演算したフリーバッファ記述子２２０７と、ＳＲＡＭ読出しアドレスに関する受信ＦＩＦＯ読出しポインタとから構成される。受信ＦＩＦＯ読出しポインタを用いて、ＤＲＡＭコントローラ７４２はＳＲＡＭ７４４に読出しリクエストをアサートする。ＳＲＡＭコントローラ７４４は、ＳＲＡＭ７４８の受信ＦＩＦＯ２１１０から表示されたデータを戻しかつ応答信号をアサートすることによって、ＤＲＡＭコントローラ７４２に応答する。ＤＲＡＭコントローラ７４２は、前記データをＤＲＡＭ書込みデータレジスタに格納し、ＤＲＡＭリクエストアドレスをＤＲＡＭアドレスレジスタに格納し、かつ応答ＤＭＡコントロールブロック２２０６にアサートする。次に、ＤＭＡコントロールブロック２２０６は、受信ＦＩＦＯ読出しポインタをデクリメントする。次に、ＤＲＡＭコントローラ７４２は、前記データをＤＲＡＭ書込みデータレジスタからバッファ２１１４へ移動させる。このようにして、連続的な３２バイトのチャンクからなるデータがＳＲＡＭ７４８に格納されるので、ＤＲＡＭコントロールブロック２２０６は、これら３２バイトのチャンクからなるデータを一度にＳＲＡＭ７４８からＤＲＡＭ７５５のバッファ２２１４に移動させる。３２バイトチャンクのデータをこのようにしてＤＲＡＭ７５５に転送することによって、ＤＲＡＭの比較的効率的なバーストモードを用いて、データをＤＲＡＭに書き込むことができる。

【０１１４】
パケットデータは、全てのパケットデータが受け取られるまで、ネットワークライン２１０５からバッファ２１１４へ流れ続ける。次に、ＭＡＣ７２２は、ライン２２２７上に「フレーム終了」（即ちパケットの終了）信号をアサートすることによって、かつ最後のパケットステータス（ＭＡＣパケットステータス）をパケット同期シーケンサ２２０４に示すことによって、到来するパケットが完了したことを表す。次にパケットプロセッシングシーケンサ２２０４が、最後にバッファ２１１４に転送するためにステータス２２２３（「プロトコルアナライザステータス」とも称する）及びＭＡＣパケットステータスをレジスタ２２１７へ移動させる。前記パケットの全データがバッファ２２１４内に置かれた後、ステータス２２２３及びＭＡＣパケットステータスは、図２３に示すように関連するデータにプリペンドされて格納されるように、バッファ２２１４に転送される。

【０１１５】
全データ及びステータスがバッファ２１１４に転送された後、パケットプロセッシングシーケンサ２２０４が、フリーバッファ記述子２２０７、バッファロードカウント、ＭＡＣ・ＩＤ及びステータスビット（「アテンションビット」とも称する）を連結することによって、サマリ２２２４（「受信パケット記述子」とも称する）を作成する。前記アテンションビットが１である場合、前記パケットは「高速パス対象」でなく、前記アテンション

ビットが０である場合には、前記パケットは「高速パス対象」となる。アテンションビットの値は、さもなければプロセッサ７８０が前記パケットが「高速パス対象」であるかどうかを決定するためにしなければならない多大な量の処理の結果を表す。例えば、アテンションビットが０であることは、前記パケットがＴＣＰプロトコル及びＩＰプロトコルの両方を用いていることを示している。この多大な量の処理を事前にハードウェアで実行しかつ次に結果をアテンションビットにエンコードすることによって、パケットが本当の「高速パスパケット」であるかどうかに関してプロセッサ７８０が次に行なう決定が加速される。

【０１１６】

次に、パケットプロセッシングシーケンサ２２０４がサマリ２２２４に関連するレディビット（図示せず）を設定し、かつサマリ２２２４をキューマネージャインタフェース２２０５に提示する。次にキューマネージャインタフェース２２０５が、「サマリキュー」２１１２（「受信記述子キュー」とも称する）のヘッド部分に書込みを要求する。キューマネージャ８０３はこのリクエストを受け取り、サマリ２２２４をサマリキュー２２１２のヘッドに書き込み、かつライン２２１０を介してキューマネージャインタフェースに応答信号をアサートする。キューマネージャインタフェース２２０５が前記応答を受け取ると、キューマネージャインタフェース２２０５は、前記サマリに関連するレディビットをクリアすることによって、サマリ２２２４がサマリキュー２２１２にあることをパケットプロセッシングシーケンサ２２０４に知らせる。また、パケットプロセッシングシーケンサ２２０４は、ＭＡＣパケットステータスとＭＡＣ・ＩＤを連結することによって、前記パケットに関する追加のステータス情報（「ベクタ」とも称する）を生成する。パケットプロセッシングシーケンサ２２０４は、このベクタに関連するレディビット（図示せず）を設定し、かつこのベクタをキューマネージャインタフェース２２０５に提示する。キューマネージャインタフェース２２０５及びキューマネージャ８０３は、上述したようにサマリ２２２４がサマリキュー２１１２のヘッドに書き込まれたのと同様にして、このベクタを協同して「ベクタキュー」２１１３のヘッドに書き込む。前記パケットのベクタがベクタキュー２１１３に書き込まれると、キューマネージャインタフェース２２０５は、前記ベクタに関連するレディビットをリセットする。

【０１１７】

サマリ２２２４（バッファ２１１４を指すバッファ記述子を含む）が一旦サマリキュー２１１２に置かれかつパケットデータがバッファ２１４４に置かれると、プロセッサ７８０はサマリ２２２４をサマリキュー２１１２から検索して、前記「アテンションビット」を調べることができる。

【０１１８】

サマリ２２２４からのアテンションビットが数字の１である場合、プロセッサ７８０は、前記パケットが「高速パス対象」でないと決定し、プロセッサ７８０はパケットヘッダを調べる必要がない。バッファ２１１４からのステータス２２２３（最初の１６バイト）のみがＳＲＡＭにＤＭＡ転送され、従ってプロセッサ７８０がこれを調べることができる。ステータス２２２３は、前記パケットがホストに転送されることになっていない種類のパケットであること（例えば、ホストが受け取るようにレジスタされていないマルチパストフレーム）を示す場合には、前記パケットは捨てられる（即ち、ホストに送られない）。ステータス２２２３が、前記パケットがホストに転送されるべきでないタイプのパケットであることを示していない場合には、パケット全体（ヘッダ及びデータ）が、「低速パス」移動及びホスト２０のプロトコルスタックによるネットワークレイヤプロセッシングのためにホスト２０のバッファに送られる。

【０１１９】

前記アテンションビットが０である場合、プロセッサ７８０は、前記パケットが「高速パス対象」であることを決定する。プロセッサ７８０が前記パケットを「高速パス対象」と決定した場合、プロセッサ７８０は前記サマリからのバッファ記述子を用いて、バッファ２１１４からの最初の約９６バイトの情報をＤＲＡＭ７５５からＳＲＡＭ７４８の部分内

にＤＭＡ転送し、それによってプロセッサ７８０がこれを調べることができる。この最初の約９６バイトは、ＩＰヘッダのＩＰソースアドレス、ＩＰヘッダのＩＰデスティネーションアドレス、ＴＣＰヘッダのＴＣＰソースアドレス及びＴＣＰヘッダのＴＣＰデスティネーションアドレスと共に、ステータス２２２３を含む。ＩＰヘッダのＩＰソースアドレス、ＩＰヘッダのＩＰデスティネーションアドレス、ＴＣＰヘッダのＴＣＰソースアドレス及びＴＣＰヘッダのＴＣＰデスティネーションアドレスは一緒になって、前記パケットが関連する１つの接続コンテキスト（ＴＣＢ）を独自に定義する。プロセッサ７８０は、これらのＴＣＰ及びＩＰヘッダのアドレスを調べ、前記パケットの接続コンテキストを決定する。次に、プロセッサ７８０は、ＩＮＩＣ２２の制御下にある接続コンテキストのリストを調査し、前記パケットがＩＮＩＣ２２の制御下で接続コンテキスト（ＴＣＢ）に関連しているかどうかを決定する。

【０１２０】

前記接続コンテキストが前記リストにない場合、前記「高速パス対象」パケットは「高速パスパケット」でないと決定される。そのような場合、パケット全体（ヘッダ及びデータ）は、ホスト２０のプロトコルスタックによる「低速パス」処理のためにホスト２０のバッファに転送される。

【０１２１】

他方、前記接続コンテキストが前記リストにある場合、ソフトウェアステートマシン２２３１、２２３２を含むプロセッサ７８０により実行されるソフトウェアが、多くの例外状態の中の１つを調べて、前記パケットが「高速パスパケット」であるか「高速パスパケット」でないかを決定する。これらの例外状態に含まれるのは、１）ＩＰ断片化が検出されたこと、２）ＩＰオプションが検出されたこと、３）予期しないＴＣＰフラグ（緊急ビットセット、リセットビットセット、ＳＹＮビットセットまたはＦＩＮビットセット）が検出されたこと、４）ＴＣＰヘッダのＡＣＫフィールドがＴＣＰウィンドウの前にあり、またはＴＣＰＡＣＫフィールドがＴＣＰウィンドウの後にあり、またはＴＣＰヘッダのＡＣＫフィールドがＴＣＰウィンドウをシュリンクさせていること、５）ＴＣＰヘッダのＡＣＫフィールドが複製のＡＣＫでありかつＡＣＫフィールドが前記複製ＡＣＫのカウントを越えている（前記複製ＡＣＫのカウントがユーザが設定可能な値であること）、及び６）ＴＣＰヘッダのシーケンス番号が順番になっていない（パケットが順番どおりに受け取られていない）ことである。プロセッサ７８０により実行されるソフトウェアがこれら例外状態の１つを検出した場合、プロセッサ７８０は前記「高速パス対象」が「高速パスパケット」でないと決定する。このような場合、前記パケットの接続コンテキストは「フラッシュ」され（接続コンテキストはホストに戻される）、それによって前記接続コンテキストはもはやＩＮＩＣ２２の制御下で接続コンテキストのリストに存在しないことになる。パケット全体（ヘッダ及びデータ）は、ホスト２０のプロトコルスタックによる「低速パス」トランスポートレイヤ及びネットワークレイヤの処理のためにホスト２０のバッファへ転送される。

【０１２２】

他方、プロセッサ７８０がそのような例外状態を発見しない場合、前記「高速パス対象」のパケットは本当の「高速パスパケット」であると決定される。次に、受信ステートマシン２２３２がＴＣＰを介して前記パケットの処理を行なう。次にバッファ２１１４の前記パケットのデータ部分が別のＤＭＡコントローラ（図２１には図示せず）によって、バッファ２１１４からホストストレージ３５のホスト割り当て型ファイルキャッシュに転送される。ある実施例では、ホスト２０は「高速パスパケット」のＴＣＰ及びＩＰヘッダの分析を行なわない。「高速パスパケット」のＴＣＰ及びＩＰヘッダの全ての分析はＩＮＩＣカード２０上で行なわれる。

【０１２３】

図２６は、「高速パスパケット」（６４ｋバイトセッションレイヤメッセージ２３００のパケット）のデータのＩＮＩＣ２２からホスト２０への転送を示す線図である。破線２３０１の左側の線図の部分はＩＮＩＣ２２を表し、破線２３０１の右側の線図の部分はホス

ト２０を表す。６４ｋバイトのセッションレイヤメッセージ２３００は約４５個のパケットを有し、図２４ではその内の４つ（２３０２、２３０３、２３０４、２３０５）にラベルが付されている。第１パケット２３０２は、トランスポート及びネットワークレイヤヘッダ（例えば、ＴＣＰ及びＩＰヘッダ）を含む部分２３０６と、セッションレイヤヘッドを含む部分２３０７と、データを含む部分２３０８とを有する。最初のステップでは、部分２３０７、部分２３０８からの最初の数バイトのデータ、及びパケット２３００の接続コンテキスト識別子がＩＮＩＣ２２からホスト２０の２５６バイトのバッファ２３０９に転送される。第２ステップでは、この情報をホスト２０が調査し、かつＩＮＩＣ２２に前記データのデスティネーション（例えばストレージ３５におけるファイルキャッシュ２３１１のロケーション）を戻す。また、ホスト２０は、バッファ２３０９からのデータの最初の数バイトをファイルキャッシュ２３１１の第１部分２３１２の開始部分にコピーする。第３ステップでは、ＩＮＩＣ２２が部分２３０８からの前記データの残りの部分をホスト２０へ、前記データの残りの部分がファイルキャッシュ２３１１の第１部分２３１２の残りの部分に格納されるように転送する。ネットワーク、トランスポートまたはセッションレイヤヘッダはファイルキャッシュ２３１１の第１部分２３１２に格納されない。次に、第２パケット２３０３のデータ部分２３１３がホスト２０に、第２パケット２３０３のデータ部分２３１３がファイルキャッシュ２３１１の第２部分２３１４に格納されるように転送する。第２パケット２３０３のトランスポートレイヤ及びネットワークレイヤヘッダ部分２３１５はホスト２０に転送されない。第１パケット２３０２のデータ部分と第２パケット２３０３のデータ部分との間でファイルキャッシュ２３１１に格納されるネットワーク、トランスポートまたはセッションレイヤヘッダはない。同様に、第２セッションレイヤメッセージの次のパケット２３０４のデータ部分２３１６はファイルキャッシュ２３１１に、第２パケット２３０３のデータ部分とファイルキャッシュ２３１１の第３パケット２３０４のデータ部分との間でネットワーク、トランスポートまたはセッションレイヤヘッダが存在しないように転送される。このようにして、前記セッションレイヤメッセージのパケットのデータ部分のみがファイルキャッシュ２３１１に置かれる。セッションレイヤメッセージ２３００からのデータはブロックとしてファイルキャッシュ２３１１内に存在し、それによってこのブロックはネットワーク、トランスポートまたはセッションレイヤヘッダを含まない。

【０１２４】

より短い単一パケットのセッションレイヤメッセージの場合、前記セッションレイヤメッセージの部分２３０７及び２３０８は、上述した長いセッションレイヤメッセージの場合のように、接続コンテキスト識別子２３１０と共にホスト２０の２５６バイトバッファ２３０９に転送される。単一パケットセッションレイヤメッセージの場合、しかしながら、この時点で転送が完了する。ホスト２０はデスティネーションをＩＮＩＣ２２に戻さず、かつＩＮＩＣ２２はそのようなデスティネーションに後のデータを転送しない。

【０１２５】

全体として、上述したデータ通信を処理するためのデバイス及びシステムによって大きな接続ベースのメッセージを処理するために必要な時間及びホストリソースが大幅に低減される。プロトコル処理速度及び効率は従来のプロトコルソフトウェアを実行する汎用ＣＰＵと比較して、特別に設計されたプロトコル処理ハードウェアを有するインテリジェントネットワークインタフェースカード（ＩＮＩＣ）により著しく加速され、かつホストＣＰＵへの割り込みが大幅に減少する。これらの利益は、ネットワークストレージの用途の場合に大きく、そのような場合、ファイル転送からのデータはホストによりファイル転送の制御が維持されて、ホストメモリバス及びホストメモリＩ／Ｏバス双方を回避することができる。

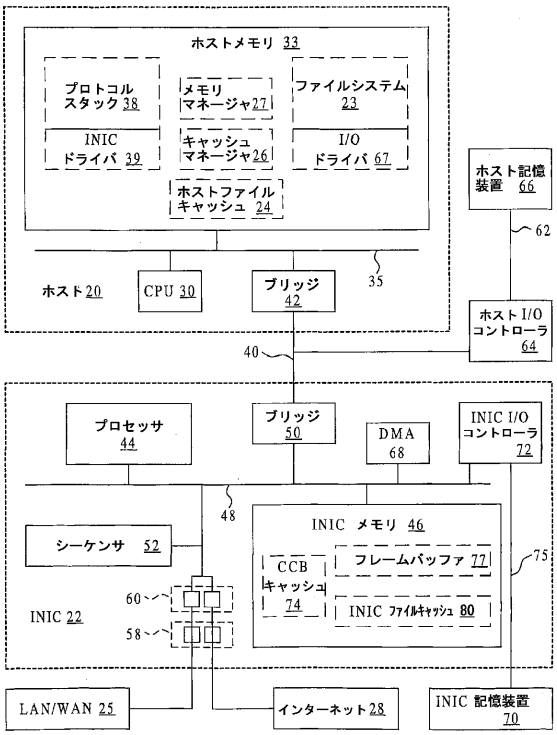
【図面の簡単な説明】

【図１】　ＩＮＩＣに接続された記憶装置のためのＩ／Ｏコントローラ及びファイルキャッシュを有するインテリジェントネットワークインタフェースカード（ＩＮＩＣ）により複数のネットワークにより接続されたホストコンピュータを有するネットワークストレー

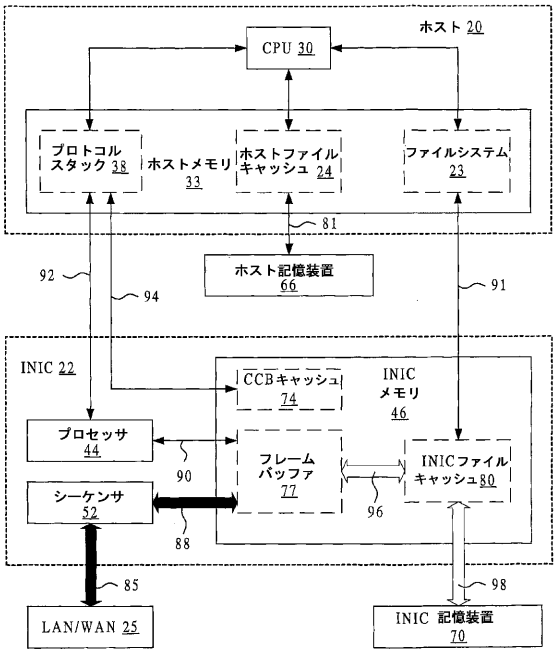
ジシステムを示す図である。
【図２】　本発明による複数のネットワーク間でデータを転送する際のＩＮＩＣ及びホストコンピュータの機能を示す図である。
【図３】　図１のシステムによりネットワークからメッセージパケットを受信する際に必要なステップの順序を示すフロー図である。
【図４】　図１のシステムによりネットワークからのリクエストに応答してメッセージパケットをネットワークに送る際に必要なステップの順序を示すフロー図である。
【図５】　ホストコンピュータにより管理される複数のＩＮＩＣにより複数のネットワーク及び複数の記憶装置に接続されたホストコンピュータを有するネットワークストレージシステムを示す図である。
【図６】　Ｉ／Ｏコントローラ無しでインテリジェントネットワークインタフェースカード（ＩＮＩＣ）により複数のＬＡＮ及び複数のＳＡＮに接続されたホストコンピュータを有するネットワークストレージシステムを示す図である。
【図７】　ネットワークラインと記憶装置間に接続されたイーサネットＳＣＳＩアダプタを有する、図６のＳＡＮの１つを示す図である。
【図８】　図６のイーサネット－ＳＣＳＩアダプタの１つを示す図である。
【図９】　ホストコンピュータにより管理される複数のＩＮＩＣにより複数のＬＡＮ及び複数のＳＡＮに接続されたホストコンピュータを有するネットワークストレージシステムを示す図である。
【図１０】　パケットコントロールシーケンサ及びフライバイシーケンサを有する、図１に示すＩＮＩＣの実施例のためのハードウェア論理を示す図である。
【図１１】　ＩＮＩＣにより受け取られるヘッダバイトを分析するための図１０のフライバイシーケンサを示す図である。
【図１２】　低速パスでパケットを処理することに加え、高速パスのために通信制御ブロックを作成しかつ制御するための図１の専用ホストプロトコルスタックを示す図である。
【図１３】　ＮｅｔＢｉｏｓ通信用に構成されたマイクロソフトのＴＣＰ／ＩＰスタック及びAlacrytechのコマンドドライバを示す図である。
【図１４】　ネットワーク記憶装置を有するサーバとクライアント間でのＮｅｔＢｉｏｓ通信エクスチェンジを示す図である。
【図１５】　図１４のサーバとクライアント間で音声または映像データを転送することができるユーザデータグラムプロトコル（ＵＤＰ）エクスチェンジを示す図である。
【図１６】　図１のＩＮＩＣに含まれるハードウェア機能を示す図である。
【図１７】　それぞれにプロセッサを有する３つのフェーズを含む、図１６のＩＮＩＣに含まれる３つ組のパイプライン化マイクロプロセッサを示す図である。
【図１８Ａ】　図１７のパイプライン化マイクロプロセッサの第１フェーズを示す図である。
【図１８Ｂ】　図１７のパイプライン化マイクロプロセッサの第２フェーズを示す図である。
【図１８Ｃ】　図１７のパイプライン化マイクロプロセッサの第３フェーズを示す図である。
【図１９】　図１７のマイクロプロセッサと対話しかつＳＲＡＭ及びＤＲＡＭを有する複数のキュー記憶装置を示す図である。
【図２０】　図１９のキュー記憶装置のためのステータスレジスタセットを示す図である。
【図２１】　図１９及び図２０のキュー記憶装置及びステータスレジスタと対話するキューマネージャを示す図である。
【図２２】　Ａ図～Ｄ図は、キャッシュメモリを割り当てるために用いられる最小使用頻度レジスタの様々な段階をそれぞれ示す図である。
【図２３】　図２２Ａ～Ｄの最小使用頻度レジスタを動作するために用いられるデバイスを示す図である。

【図２４】　図１６のＩＮＩＣの別の実施例を示す図である。
【図２５】　図２４の受信シーケンサ２１０５を詳細に示す図である。
【図２６】　ネットワークから受け取ったデータのためのキャッシュファイルにデスティネーションを獲得するためのメカニズムを示す図である。

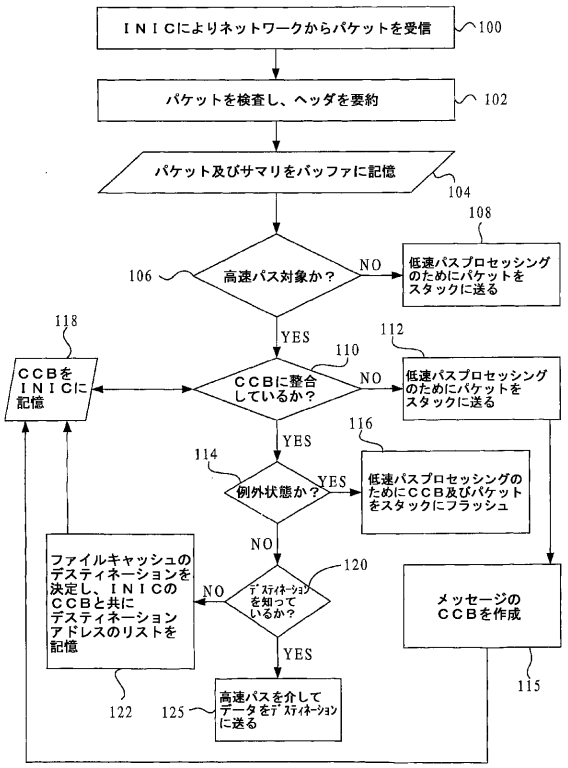
【図１】



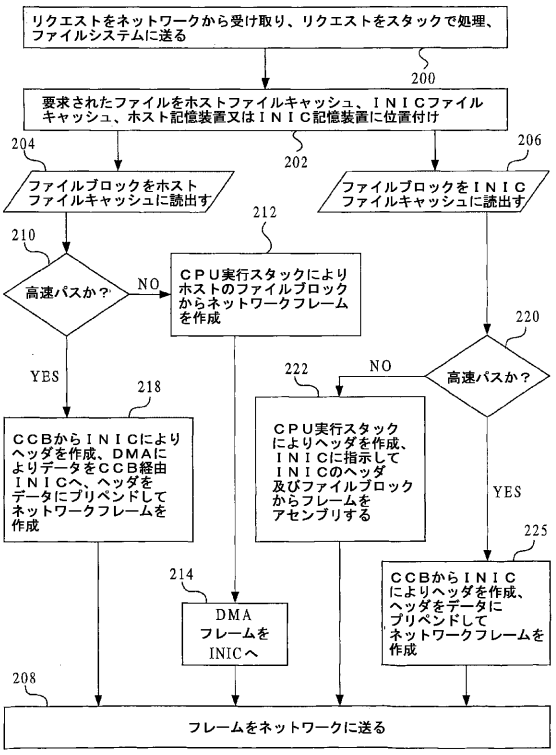
【図２】



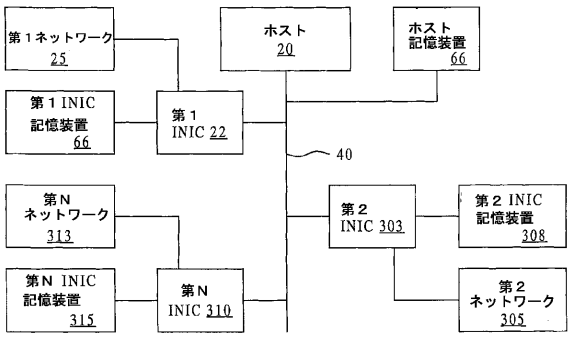
【図３】


ＩＮＩＣによりネットワークからパケットを受信
100
パケットを検査し、ヘッダを要約
102
パケット及びサマリをバッファに記憶
104
108
高速パス対象か？
106
NO
低速パスプロセッシングのためにパケットをスタックに送る
YES
118
ＣＣＢをＩＮＩＣに記憶
110
ＣＣＢに整合しているか？
NO
112
低速パスプロセッシングのためにパケットをスタックに送る
YES
116
例外状態か？
114
YES
低速パスプロセッシングのためにＣＣＢ及びパケットをスタックにフラッシュ
NO
ファイルキャッシュのデスティネーションを決定し、ＩＮＩＣのＣＣＢと共にデスティネーションアドレスのリストを記憶
NO
デスティネーションを知っているか？
120
YES
メッセージのＣＣＢを作成
115
122
125
高速パスを介してデータをデスティネーションに送る

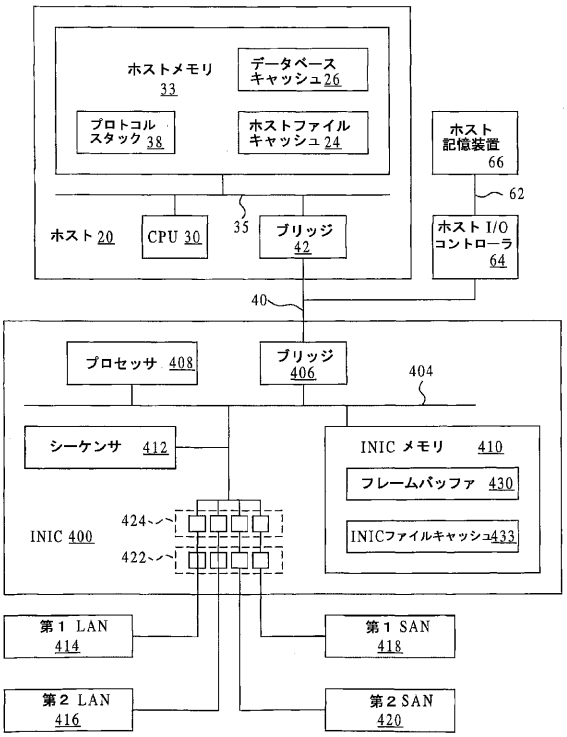
【図４】


リクエストをネットワークから受け取り、リクエストをスタックで処理、ファイルシステムに送る
200
要求されたファイルをホストファイルキャッシュ、ＩＮＩＣファイルキャッシュ、ホスト記憶装置又はＩＮＩＣ記憶装置に位置付け
202
204
ファイルブロックをホストファイルキャッシュに読出す
206
ファイルブロックをＩＮＩＣファイルキャッシュに読出す
210
高速パスか？
NO
212
ＣＰＵ実行スタックによりホストのファイルブロックからネットワークフレームを作成
YES
218
ＣＣＢからＩＮＩＣによりヘッダを作成、ＤＭＡによりデータをＣＣＢ経由ＩＮＩＣへ、ヘッダをデータにプリペンドしてネットワークフレームを作成
222
NO
高速パスか？
220
ＣＰＵ実行スタックによりヘッダを作成、ＩＮＩＣに指示してＩＮＩＣのヘッダ及びファイルブロックからフレームをアセンブリする
YES
225
ＣＣＢからＩＮＩＣによりヘッダを作成、ヘッダをデータにプリペンドしてネットワークフレームを作成
214
DMA フレームをINICへ
208
フレームをネットワークに送る

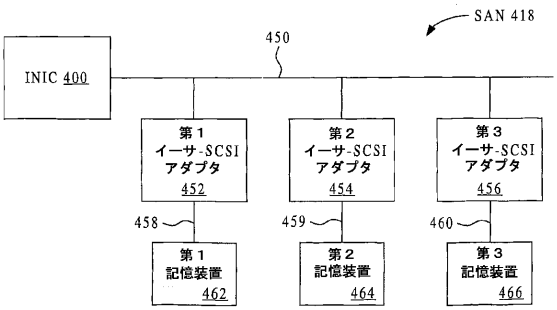
【図５】


第１ネットワーク 25
ホスト 20
ホスト記憶装置 66
第１INIC記憶装置 66
第１ INIC 22
40
第Ｎネットワーク 313
第２ INIC 303
第２ INIC記憶装置 308
第Ｎ INIC記憶装置 315
第Ｎ INIC 310
第２ネットワーク 305

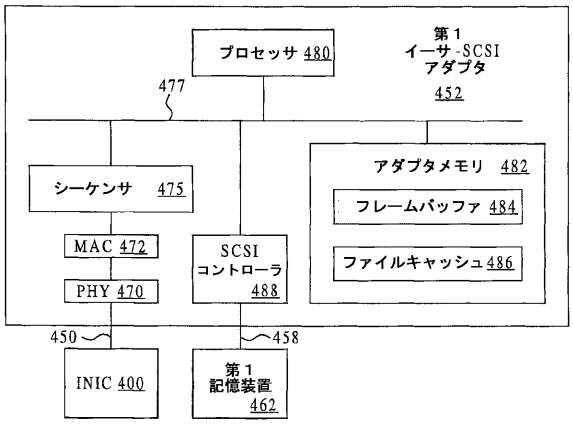
【図６】


ホストメモリ 33
データベースキャッシュ26
プロトコルスタック 38
ホストファイルキャッシュ 24
ホスト記憶装置 66
62
ホスト 20
CPU 30
35
ブリッジ 42
ホスト I/O コントローラ 64
40
プロセッサ 408
ブリッジ 406
404
シーケンサ 412
INIC メモリ 410
フレームバッファ 430
INIC 400
424
422
INICファイルキャッシュ433
第１ LAN 414
第１ SAN 418
第２ LAN 416
第２ SAN 420

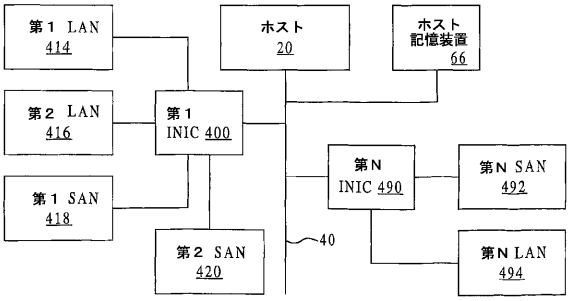
【図７】



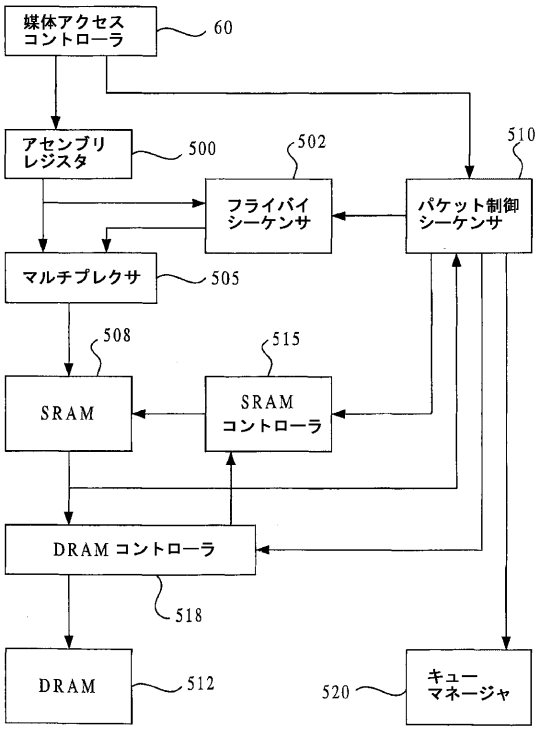
【図８】



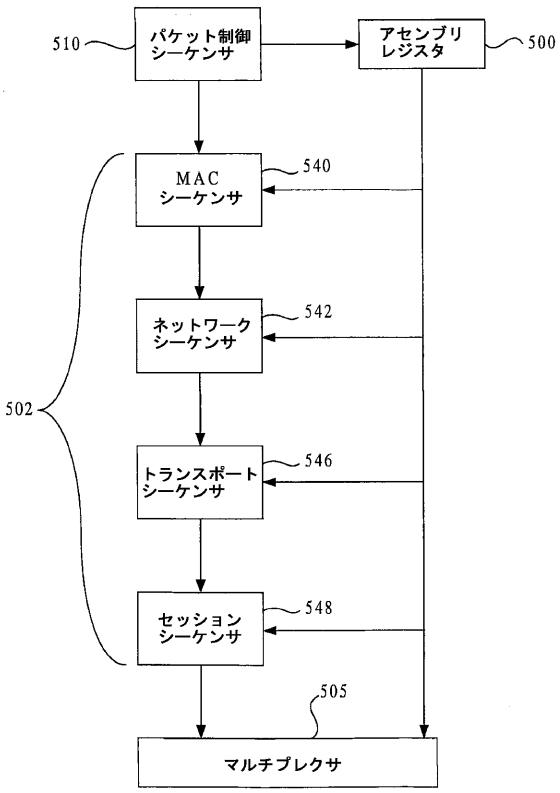
【図９】



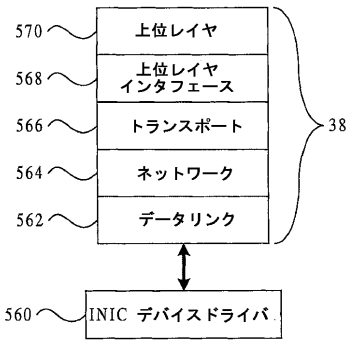
【図１０】



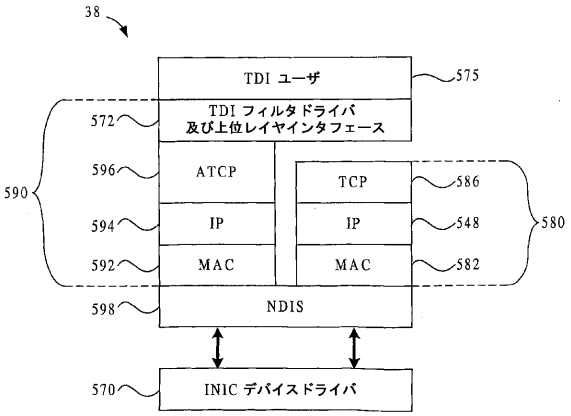
【図１１】



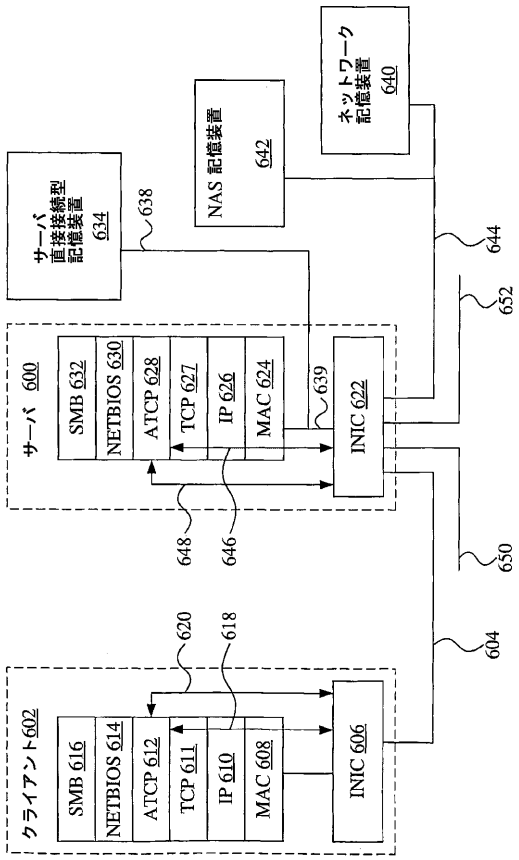
【図１２】



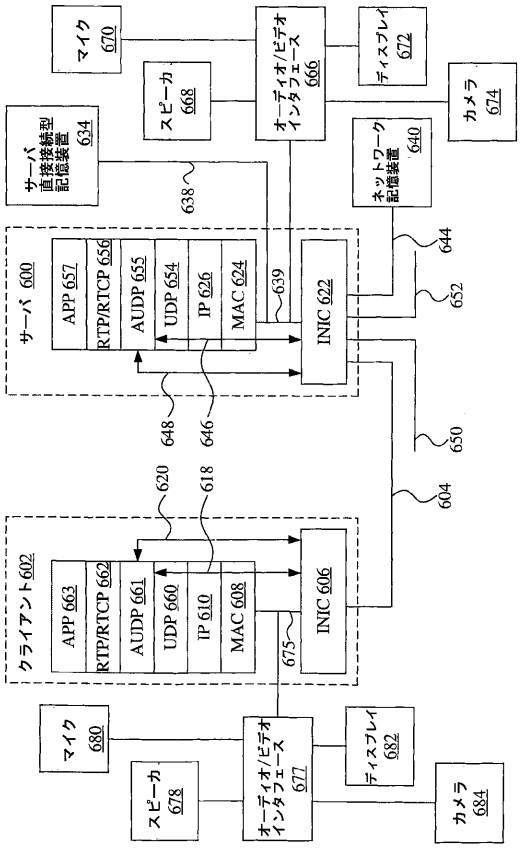
【図１３】



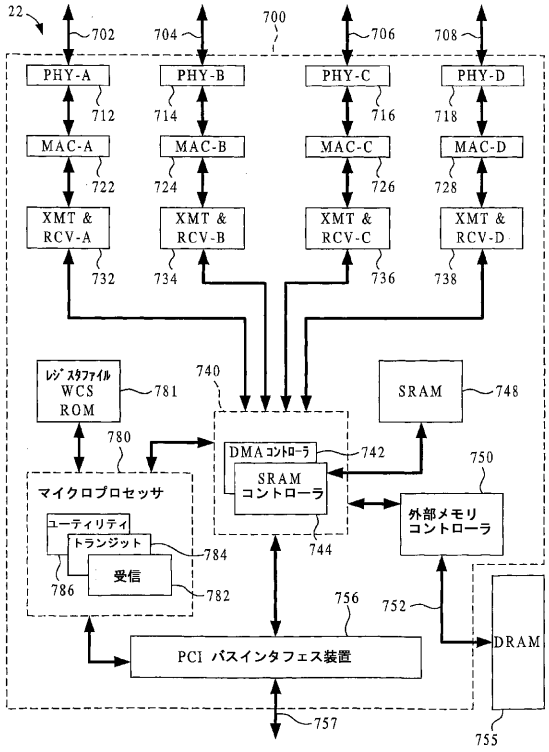
【図１４】



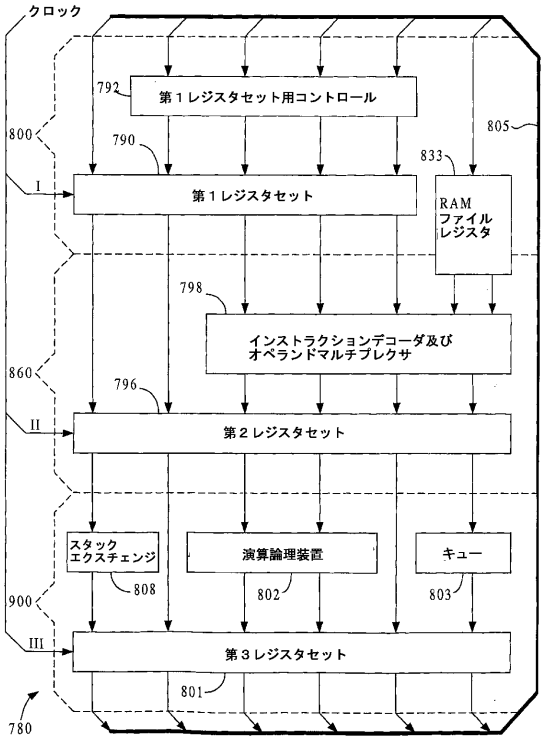
【図１５】



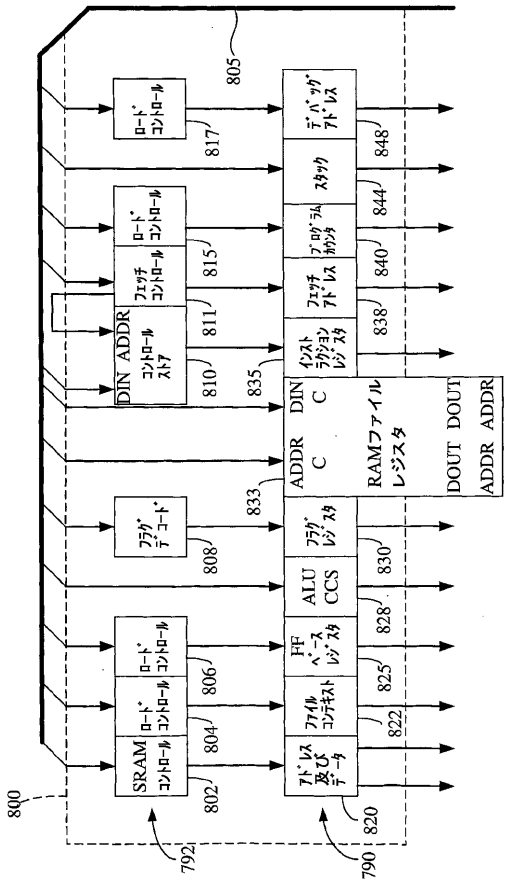
【図１６】



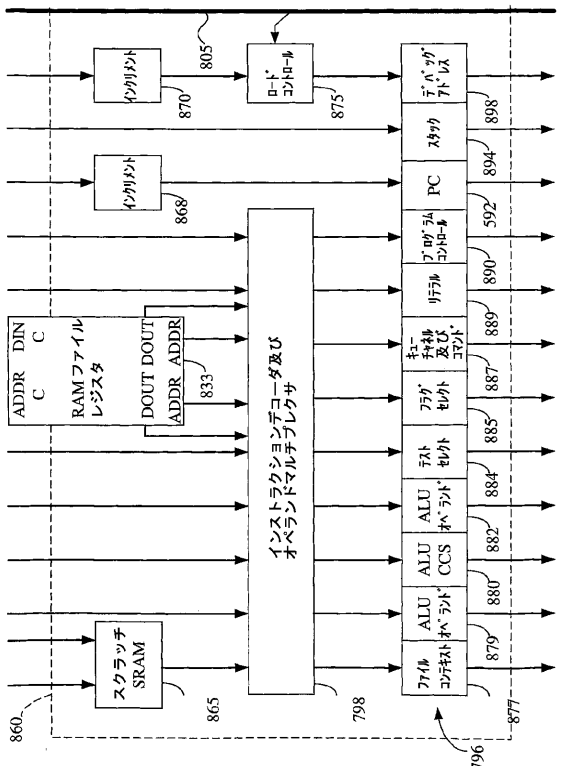
【図１７】



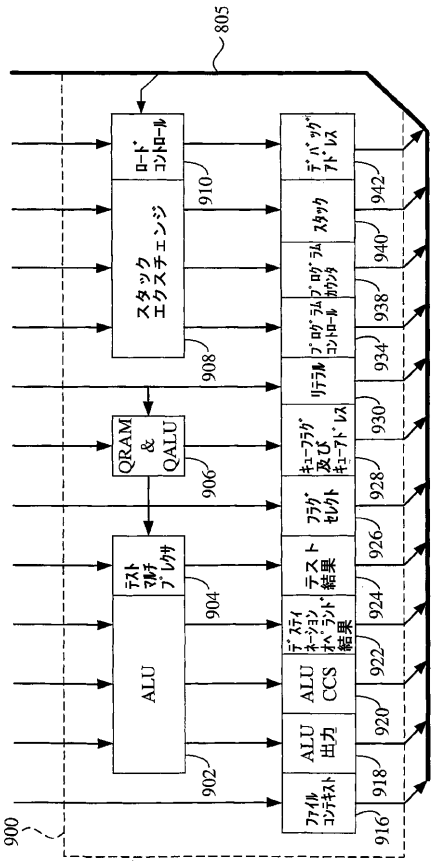
【図１８－Ａ】



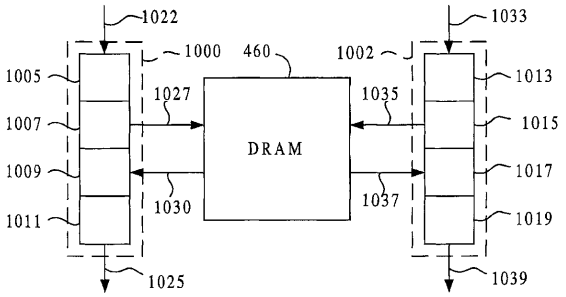
【図１８－Ｂ】



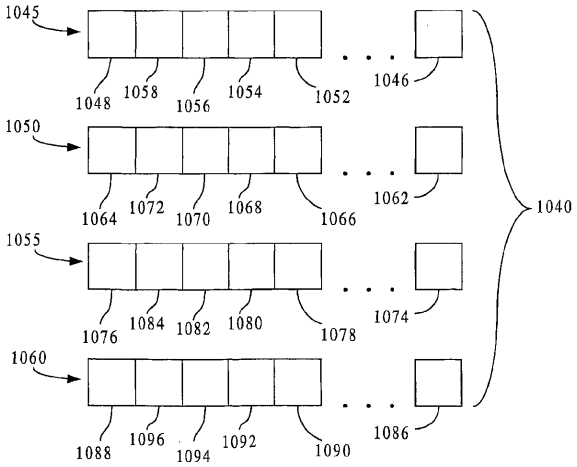
【図１８－Ｃ】



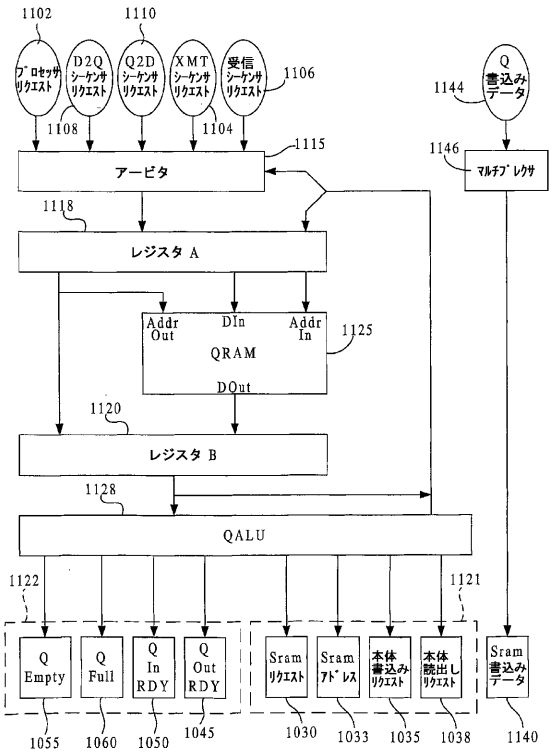
【図１９】



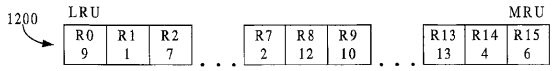
【図２０】



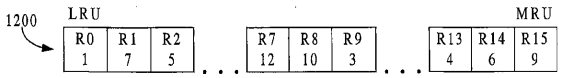
【図２１】



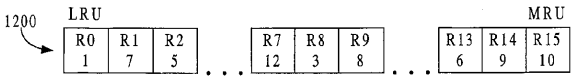
【図２２－Ａ】



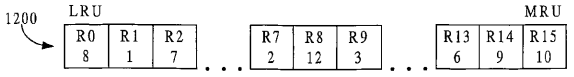
【図２２－Ｂ】



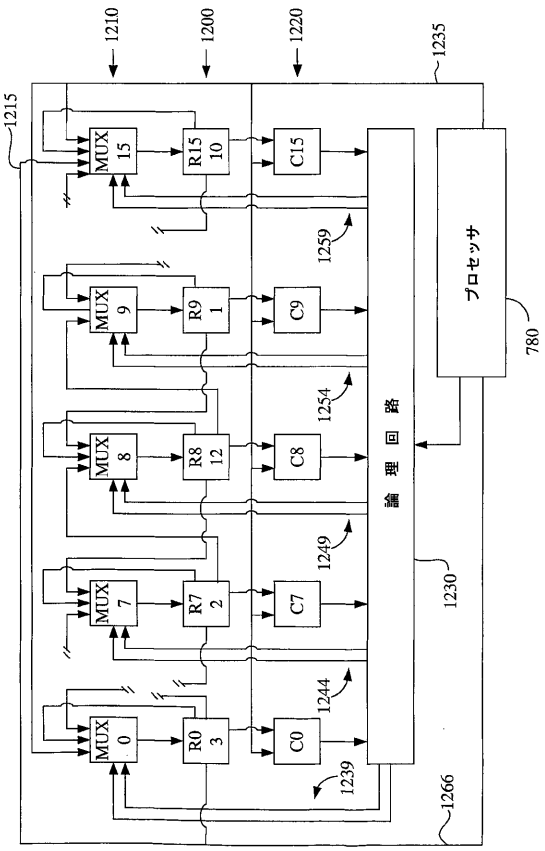
【図２２－Ｃ】



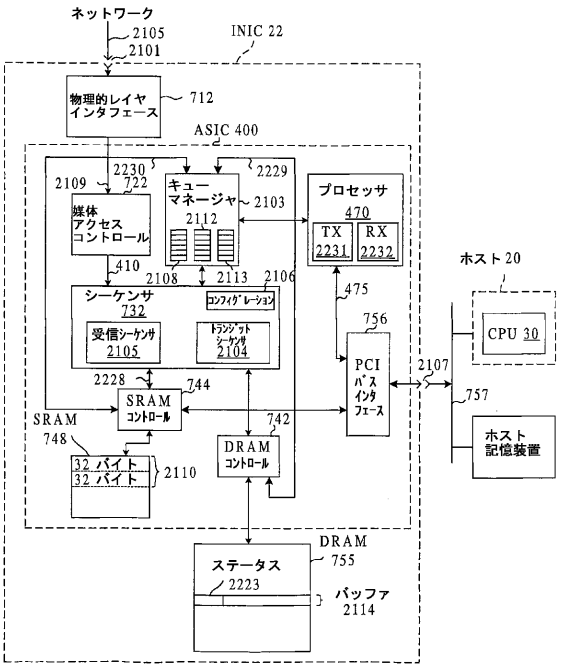
【図２２－Ｄ】



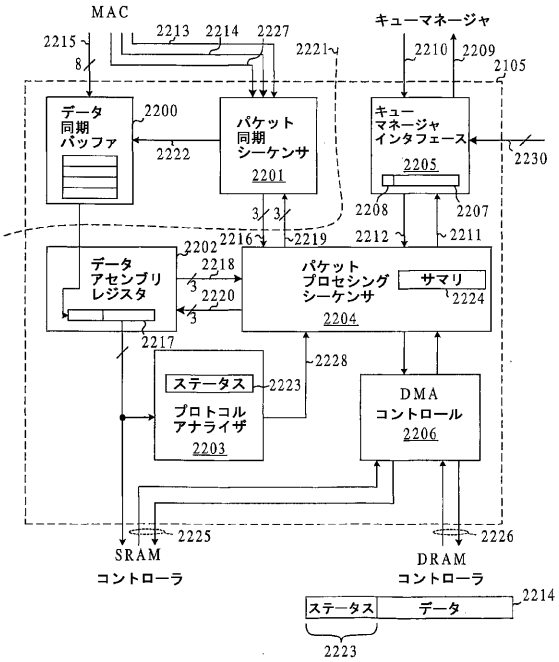
【図２３】



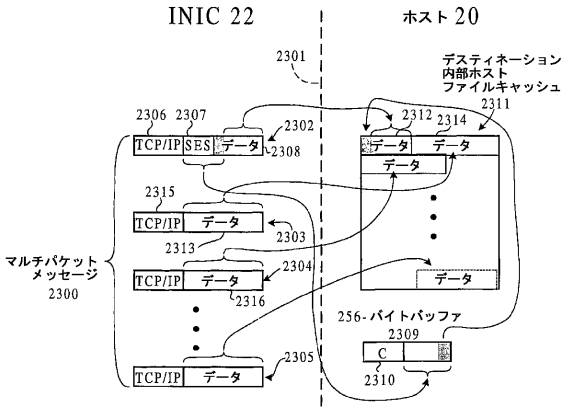
【図２４】


ネットワーク
2105
2101
INIC 22
物理的レイヤ
インタフェース
712
ASIC 400
2109
2230
722
2229
媒体
アクセス
コントロール
キュー
マネージャ
2103
2112
プロセッサ
470
TX
2231
RX
2232
410
2108
2113
2106
シーケンサ
732
コンフィグレーション
受信シーケンサ
2105
トランジット
シーケンサ
2104
475
ホスト 20
CPU 30
756
PCI
バス
インタ
フェース
2107
2228
744
SRAM
コントロール
SRAM
748
742
757
ホスト
記憶装置
32 バイト
32 バイト
2110
DRAM
コントロール
DRAM
755
ステータス
2223
バッファ
2114

【図２５】


MAC
2215
8
2213
2214
2227
2221
キューマネージャ
2210
2209
2105
データ
同期
バッファ
2200
2222
パケット
同期
シーケンサ
2201
キュー
マネージャ
インタフェース
2205
2230
2208
2207
3
3
2216
2219
2212
2211
2202
データ
アセンブリ
レジスタ
2218
3
2220
3
パケット
プロセシング
シーケンサ
2204
サマリ
2224
2217
2228
ステータス
2223
プロトコル
アナライザ
2203
DMA
コントロール
2206
2225
2226
SRAM
コントローラ
DRAM
コントローラ
ステータス
データ
2214
2223

【図２６】


INIC 22
ホスト 20
2301
デスティネーション
内部ホスト
ファイルキャッシュ
2311
2306
2307
2302
2312
2314
TCP/IP
SES
データ
2308
データ
データ
データ
2315
TCP/IP
データ
2303
2313
マルチパケット
メッセージ
2300
TCP/IP
データ
2304
データ
2316
256-バイトバッファ
2309
TCP/IP
データ
2305
C
2310

フロントページの続き

(31)優先権主張番号 09/802,551
(32)優先日 平成13年3月9日(2001.3.9)
(33)優先権主張国 米国(US)

(72)発明者 ブーシェ，ローレンス，ビー
アメリカ合衆国，カリフォルニア州・９５０７０，サラトガ，モンタルボ・ハイツ・ドライブ・２０６０５

審査官 横山 佳弘

(56)参考文献 国際公開第２０００／０１３０９１（WO，Ａ１）
特開平１１－０４５２０３（ＪＰ，Ａ）
特開平０９－１２８３１４（ＪＰ，Ａ）
国際公開第００／００４４５３（WO，Ａ１）
特開平１１－１８４６３９（ＪＰ，Ａ）
特開２０００－１０１５３７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 13/38
G06F 12/00
G06F 13/10
G06F 13/00