(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4406604号
(P4406604)
(45)発行日　平成22年2月3日(2010.2.3)　(24)登録日　平成21年11月13日(2009.11.13)

(51)Int.Cl.　F I
G06F 13/10 (2006.01)　G06F 13/10 340B

請求項の数 10　(全 57 頁)

(21)出願番号	特願2004-511951 (P2004-511951)
(86)(22)出願日	平成15年6月10日 (2003.6.10)
(65)公表番号	特表2006-516054 (P2006-516054A)
(43)公表日	平成18年6月15日 (2006.6.15)
(86)国際出願番号	PCT/US2003/018386
(87)国際公開番号	WO2003/104943
(87)国際公開日	平成15年12月18日 (2003.12.18)
審査請求日	平成18年6月7日 (2006.6.7)
(31)優先権主張番号	60/388,407
(32)優先日	平成14年6月11日 (2002.6.11)
(33)優先権主張国	米国 (US)

(73)特許権者　504457223
パンドヤ　アシシュ　エイ
アメリカ合衆国　カリフォルニア州　９５７６２　エル　ドラド　ヒルズ　ラファイエット　ドライヴ　４３１８
(74)代理人　100082005
弁理士　熊倉　禎男
(74)代理人　100067013
弁理士　大塚　文昭
(74)代理人　100074228
弁理士　今城　俊夫
(74)代理人　100086771
弁理士　西島　孝喜

最終頁に続く

(54)【発明の名称】ＴＣＰ／ＩＰ、ＲＤＭＡ、及びＩＰストレージアプリケーションのための高性能ＩＰプロセッサ

(57)【特許請求の範囲】
【請求項１】
トランスミッションコントロールプロトコル（ＴＣＰ）又はストリームコントロールトランスミッションプロトコル（ＳＣＴＰ）又はユーザデータグラムプロトコル（ＵＤＰ）オペレーション、又はその任意の組合せをインターネットプロトコル（ＩＰ）上で提供するハードウエアプロセッサであって、前記オペレーションは、イニシエータから又はイニシエータへ、及びターゲットへ又はターゲットからネットワーク上でデータを転送するリモートダイレクトメモリアクセス（ＲＤＭＡ）機能を含み、スモールコンピュータシステムインタフェース（ＳＣＳＩ）コマンド層及びＩＰストレージドライバを有するホストプロセッサによって要求され、前記ハードウエアプロセッサは、
ａ．ＲＤＭＡ機構、
ｂ．プロセッサ内の作動のためのホストプロセッサのコマンド層からのコマンドをスケジュールに入れるためのコマンドスケジューラ、
ｃ．前記コマンドスケジューラを経由する既存セッションのための前記ホストプロセッサからのスケジュールされたコマンドを待ち行列に入れるための第１のコマンド待ち行列、
ｄ．前記コマンドスケジューラを経由する現在存在しないセッションのための前記ホストプロセッサからのスケジュールされたコマンドを待ち行列に入れるための第２のコマンド待ち行列、
ｅ．前記コマンドが移送されたセッションの状態を記録するための、また、該コマンド

のＲＤＭＡを使用するものに対するＲＤＭＡの進行を記録するためのデータベース、
ｆ．コマンド実行の状況を処理のために前記ＳＣＳＩ層に通信するためのプロセッサと前記ＳＣＳＩ層の間の通信パス、及び
ｇ．コマンドを解釈し、前記ハードウエアプロセッサと前記ターゲット及び前記イニシエータの少なくとも１つとの間でデータを転送するためのＲＤＭＡオペレーションを実行するために協働するように連結した少なくとも１つの送信／受信エンジン及び少なくとも１つのコマンドエンジン、
を含むことを特徴とするプロセッサ。
【請求項２】
少なくとも１つのデータプロセッサを複数のインターネットプロトコル（ＩＰ）プロセッサにインタフェースで接続するための複数のＩＰプロセッサに連結した少なくとも１つのデータプロセッサを含むマルチプロセッサシステムであって、
前記複数のＩＰプロセッサの各々は、リモートダイレクトメモリアクセス（ＲＤＭＡ）プロトコルと、トランスミッションコントロールプロトコル（ＴＣＰ）、ストリームコントロールトランスミッションプロトコル（ＳＣＴＰ）、及びユーザデータグラムプロトコル（ＵＤＰ）のうちの１つのプロトコルとを実行するよう構成され、データをＩＰネットワークのセット上に転送し、前記ＩＰプロセッサの１つは前記少なくとも１つのプロセッサからのコマンドを受信するよう構成され、 前記複数のＩＰプロセッサの１つは、
ａ．ＲＤＭＡを介してデータを転送するためのＲＤＭＡ機構、
ｂ．前記データを含む複数のＩＰパケットを処理するための少なくとも１つのパケットプロセッサ、
ｃ．前記データの転送に使用されるセッションに関するＴＣＰ／ＩＰセッション情報を保存するためのセッションメモリ、
ｄ．前記データを保存するメモリに対するメモリアクセスを制御するための少なくとも１つのメモリコントローラ、
ｅ．前記ＩＰネットワークの少なくとも１つに連結し、リモートホストからの要求を受信するための少なくとも１つのメディアインタフェース、及び
ｆ．前記少なくとも１つのデータプロセッサに連結するためのホストインタフェース、又は前記受信した要求に応答してファブリックを介してストレージシステムにアクセスするためのファブリックインタフェース、
ｇ．前記コマンドの状態を記録し、前記コマンドのターミネーションに基づいて前記コマンドの状態を更新するよう構成されたコマンド状態テーブル、
を含む、
ことを特徴とするシステム。
【請求項３】
前記ＩＰプロセッサの１つは、
ａ．ＩＰストレージセッション情報を保存するためのＩＰストレージセッションメモリ、
ｂ．ＩＰパケットを分類するための分類プロセッサ、及び、前記少なくとも１つのパケットプロセッサによって使用される分類タグと前記ＩＰパケットとの対応付け、
ｃ．前記データのフローを制御するためのフローコントローラ、
ｄ．ポリシーを適用するためのポリシープロセッサ、
ｅ．認証、暗号化、及び復号化のうちの１又は複数のセキュリティ作動を実行するためのセキュリティプロセッサ、
ｆ．パケットを保存するためのパケットメモリ、
ｇ．プレーン処理を制御するためのコントローラ、
ｈ．パケットスケジューラ、
ｉ．ピアプロセッサにインタフェース接続するための共用プロセッサインタフェース、又は
ｊ．これらの任意の組合せ、

のうちの少なくとも１つを含むことを特徴とする請求項２に記載のマルチプロセッサシステム。

【請求項４】

トランスミッションコントロールプロトコル（ＴＣＰ）又はインターネットプロトコル（ＩＰ）ネットワーク上の他の接続指向プロトコルを使用しリモートダイレクトメモリアクセス（ＲＤＭＡ）を介してデータ転送が可能なハードウエアプロセッサを有するストレージエリアネットワークであって、

前記ハードウエアプロセッサはプログラム可能なように構成され、ポリシー処理エンジンを含み、前記ポリシー処理エンジンは、パケット基準毎、フロー基準毎、及びコマンド基準毎の１又は複数に基づくポリシーを適用するものであって、

前記ハードウエアプロセッサは、

ホストプロセッサから受信したデータ転送に対応するコマンドを実行し、前記受信したコマンドに基づいて前記ホストプロセッサのアドレスを決定し、前記コマンドの完了に際して前記決定されたアドレスにおいて前記ホストプロセッサのみに割り込みを行うようプログラム可能である

ことを特徴とするストレージエリアネットワーク。

【請求項５】

前記少なくとも１つの送信／受信エンジンは、分離された送信エンジン及び分離された受信エンジンとして実装されることを特徴とする、請求項１に記載のハードウエアプロセッサ。

【請求項６】

前記少なくとも１つの送信／受信エンジン及び少なくとも１つのコマンドエンジンは、送信、受信、及びコマンドエンジン機能を与える少なくとも１つの複合エンジンとして実装されることを特徴とする、請求項１に記載のハードウエアプロセッサ。

【請求項７】

前記ハードウエアプロセッサは、イニシエータからターゲットは、あるいはターゲットからイニシエータへのデータ転送についてのｉＳＣＳＩコマンドをｉＳＣＳＩセッションのいて受信し、前記セッションの各々の状態を前記データベースにセッションエントリとして記録することを特徴とする、請求項１に記載のハードウエアプロセッサ。

【請求項８】

前記複数のＩＰプロセッサの２以上のものが互いに結合されることを特徴とする、請求項２に記載のマルチプロセッサシステム。

【請求項９】

前記ハードウエアプロセッサは深いパケット分類エンジンを更に含むことを特徴とする、請求項４に記載のストレージエリアネットワーク。

【請求項１０】

前記ハードウエアプロセッサは、サーバ、ホストバスアダプタ、スイッチ、スイッチラインカード、ゲートウェイ、ゲートウェイのラインカード、ストレージエリアネットワークアプライアンス、アプライアンスのラインカード、ストレージシステム、及びストレージシステムのラインカード、のうちの１又は複数において配置されることを特徴とする、請求項４に記載のストレージエリアネットワーク。

【発明の詳細な説明】

【技術分野】

【０００１】

関連出願

２００２年６月１１日出願の米国特許仮出願出願番号第６０／３８８，４０７号に対して優先権を請求する。２００３年６月１０日出願の「ＲＤＭＡを使用する高性能ＩＰプロセッサ」という名称の米国特許出願（出願番号未定）、２００３年６月１０日出願の「ＲＤＭＡを使用するＴＣＰ／ＩＰプロセッサ及びエンジン」という名称の米国特許出願（出願番号未定）、２００３年６月１０日出願の「ＲＤＭＡを使用するＩＰストレージプロセ

ッサ及びそのためのエンジン」という名称の米国特許出願（出願番号未定）、２００３年６月１０日出願の「高性能ＩＰプロセッサ用メモリシステム」という名称の米国特許出願（出願番号未定）、２００３年６月１０日出願の「インターネットプロトコル及びＲＤＭＡを使用するデータ処理システム」という名称の米国特許出願（出願番号未定）、２００３年６月１０日出願の「高性能ＩＰプロセッサ」という名称の米国特許出願（出願番号未定）、及び２００３年６月１０日出願の「インターネットプロトコルを使用するデータ処理システム」という名称の米国特許出願（出願番号未定）は、上記の仮出願に関連する。

本発明は、一般的に、ストレージネットワーキング半導体に関し、より詳細には、インターネットプロトコル（ＩＰ）ベースのストレージネットワークを作り出すために使用される高性能ネットワークストレージプロセッサに関する。

【背景技術】

【０００２】

インターネットプロトコル（ＩＰ）は、ローカルエリアネットワーク（ＬＡＮ）、メトロエリアネットワーク（ＭＡＮ）、ワイドエリアネットワーク（ＷＡＮ）などの様々なネットワークに亘って配備される最も普及しているネットワーキングプロトコルである。ストレージエリアネットワーク（ＳＡＮ）は、主にファイバチャンネル（ＦＣ）技術に基づいている。ＩＰベースのストレージネットワークを作り出す必要性が存在する。

【０００３】

データストリームを移送するように設計されたＩＰ上でブロックストレージトラフィックを移送する時に、データストリームは、ＩＰの上で実行されるように階層化された伝送制御プロトコル（ＴＣＰ）を使用して移送される。ＴＣＰ／ＩＰは、オペレーティングシステム内のソフトウエアで実行される確実な接続／セッション志向プロトコルである。ＴＣＰ／ＩＰソフトウエアスタックは、将来配備されることになる高回線速度を扱うには非常に低速である。現在、１Ｇｂｐｓネットワーク接続を使用してＴＣＰ／ＩＰスタックを実行する１ＧＨｚプロセッサベースサーバは、５０～７０％又はそれ以上のプロセッササイクルを使用し、プロセッサがサーバ上で実行されるアプリケーションに割り当てるのに利用可能な最小限のサイクルを残す。このオーバーヘッドは、高性能ＩＰネットワークに対する場合と同じく、ストレージデータをＴＣＰ／ＩＰ上で移送する場合は許容できないことである。従って、新しいハードウエアソリューションは、ストレージ及びデータトラフィックを運び、ＦＣベースソリューションと競合するようにＴＣＰ／ＩＰスタックを加速するであろう。ＴＣＰプロトコルに加えて、ＳＣＴＰ及びＵＤＰプロトコルのような他のプロトコル、及びデータストリームを移送するのに適切な他のプロトコルを使用することもできる。

【０００４】

【特許文献１】米国特許仮出願出願番号第６０／３８８，４０７号

【特許文献２】２００３年６月１０日出願「ＲＤＭＡを使用する高性能ＩＰプロセッサ」米国特許出願（出願番号未定）

【特許文献３】２００３年６月１０日出願「ＲＤＭＡを使用するＴＣＰ／ＩＰプロセッサ及びエンジン」米国特許出願（出願番号未定）

【特許文献４】２００３年６月１０日出願「ＲＤＭＡを使用するＩＰストレージプロセッサ及びそのためのエンジン」米国特許出願（出願番号未定）

【特許文献５】２００３年６月１０日出願「高性能ＩＰプロセッサ用メモリシステム」米国特許出願（出願番号未定）

【特許文献６】２００３年６月１０日出願「インターネットプロトコル及びＲＤＭＡを使用するデータ処理システム」米国特許出願（出願番号未定）

【特許文献７】２００３年６月１０日出願「高性能ＩＰプロセッサ」米国特許出願（出願番号未定）

【特許文献８】２００３年６月１０日出願「インターネットプロトコルを使用するデータ処理システム」米国特許出願（出願番号未定）

【発明の開示】

【０００５】
本発明人は、ホストプロセッサからＴＣＰ／ＩＰプロトコルスタックのオーバーヘッドを大幅に低減し、ＩＰに基づく高回線速度のストレージ及びデータ移送ソリューションを可能にする高性能ハードウエアプロセッサについて説明する。
従来的に、ＴＣＰ／ＩＰネットワーキングスタックは、ソフトウエアスタックとしてオペレーティングシステムカーネルの内部で実行されている。ソフトウエアＴＣＰ／ＩＰスタックの実行は、上述のように、１Ｇｂｐｓネットワークに情報提供する時に、１ＧＨｚプロセッサで使用可能な処理サイクルの５０％よりも多くを費やす。オーバーヘッドは、チェックサム計算、メモリバッファコピー、パケット到着時のプロセッサ割り込み、セッションの確立、セッションの取外し、及び他の確実な移送サービスを含むソフトウエアＴＣＰ／ＩＰスタックの様々な局面から生じる。ソフトウエアスタックのオーバーヘッドは、回線速度がより高い時は法外に大きくなる。より性能が低いホストプロセッサを使用する無線ネットワークのような回線速度がより低いネットワークでも同様の問題が発生する。ハードウエアの実施によってホストプロセッサからオーバーヘッドを取り除くことができる。
【０００６】
オペレーティングシステムから提供されたソフトウエアＴＣＰ／ＩＰネットワーキングスタックは、ホストプロセッササイクルの大部分を使い果たす。ＴＣＰ／ＩＰは、不安定なデータリンク上で実行することができる確実な移送手段である。従って、ネットワークパケットが欠落するか又はエラーを有する場合、ＴＣＰは、パケットの再送信を行う。パケットのエラーは、パケット内に有するチェックサムを使用して検出される。ＴＣＰパケットの受け取り側は、受け取ったパケットのチェックサムを実行し、それを受け取ったチェックサムと比較する。これは、パケット内の各受信バイトに関わる各パケット上で実行される高価な計算集約的作業である。発信元と宛先の間のパケットは、順序が狂って到着する場合があり、ＴＣＰ層は、それをより上の層に呈示する前にデータストリームの順序付けを行う。ＩＰパケットはまた、リンク層の最大移送ユニット（ＭＴＵ）に基づいてフラグメント化される場合があり、従って、受け取り側は、パケットを非フラグメント化することを期待される。これらの機能は、順序の狂ったパケット、フラグメント化したパケット、又は肯定応答されなかったパケットの例えばネットワークカード上のメモリへの一時的保存をもたらす。回線速度が１Ｇｂｐｓを超えて増加すると、これらによって生じるメモリサイズオーバーヘッド及びメモリ速度の妨害により、ネットワークカードにかなりのコストが追加され、同じく莫大な性能的オーバーヘッドを引き起こす。多くのプロセッサのリソースを費やす別の機能は、ネットワークカードバッファ、カーネルバッファ、及びアプリケーションバッファへの／からのデータのコピーである。
【０００７】
マイクロプロセッサは、深いパイプライン化とスーパースカラーアーキテクチャを使用して、益々高性能と高速度を実現している。小さなパケットの到着時にこれらのプロセッサに割り込むと、コンテキスト切替オーバーヘッド、パイプラインフラッシュ、及びパイプラインの補充のために厳しい性能劣化を引き起こすことになる。従って、プロセッサの割り込みは、最も基本的な割り込みだけにして最小限にすべきである。ブロックストレージトラフィックがＴＣＰ／ＩＰネットワーク上で移送される時、これらの性能の問題が重大になり、ストレージトラフィックの処理能力と待ち時間に大幅な影響を与える。従って、ストレージトラフィックの移送の処理全体におけるプロセッサの介入は、ＩＰベースストレージソリューションが、ハードウエア実行を目的として特化されたファイバチャンネルのような他の特化されたネットワークアーキテクチャと同等な性能と待ち時間を有するために最小にする必要がある。ｉＳＣＳＩ、ＦＣＩＰ、ｉＦＣＰ、及び他のもの（ＮＦＳ、ＣＩＦＳ、ＤＡＦＳ、ＨＴＴＰ、ＸＭＬ、ＸＭＬ派生物（音声ＸＭＬ、ＥＢＸＭＬ、マイクロソフトＳＯＡＰ、及びその他など）、ＳＧＭＬ、及びＨＴＭＬフォーマットのような）のような出現してきたＩＰベースストレージ基準は、ＴＣＰ／ＩＰセグメント内のストレージ及びデータトラフィックをカプセル化する。しかし、通常は、ＴＣＰセグメント

とＴＣＰパケットによってカプセル化されたプロトコルデータユニットとの間には、アラインメント関係が存在しない。これは、今日のネットワークでは非常に頻繁なイベントであるパケットの順序が狂って到着した時に問題となる。ストレージ及びデータブロックは、ストリーム内の中間パケットが到着するまで、使用のための抽出を順序が狂ったパケットから行うことができず、この中間パケットは、ネットワークアダプタにこれらのパケットをメモリに保存させ、中間パケットの到着時にそれらを検索させ、それらを順序化させることになる。これは、要求されるメモリストレージのサイズから、及びメモリサブシステムがサポートすることを期待される性能からも、特に回線速度が１Ｇｂｐｓを超える時に高価になる可能性がある。各ＴＣＰセグメントがプロトコルデータユニット及びそのシーケンスを一意的に識別することができる場合、このオーバーヘッドを排除することができる。これは、パケットをホストシステム内のそれらの末端メモリ位置に直接移送することを可能にすることができる。ホストプロセッサの介入はまた、ストレージサブシステムに転送される場合があるか又はクラスター化した環境又は他のクライアントサーバ環境で他のプロセッサと共有される大きなデータブロックの転送では最小限にされるべきである。プロセッサは、影響を最小限にするためにストレージコマンド境界上でのみ割り込まれるべきである。

【０００８】

本明細書に示すＩＰプロセッサは、革新的な構造的特徴と設計により、以上概説した様々な問題の影響を除去するか又は著しく低減する。説明したプロセッサのアーキテクチャは、ストレージ及びデータペイロードを運ぶＴＣＰトラフィックを終了させる機能を提供し、それによってホストプロセッサ上のＴＣＰ／ＩＰネットワーキングスタックのオーバーヘッドを除去するか又は大幅に低減し、パケットを入力から出力まで最小の待ち時間で通過させるパケットストリームアーキテクチャをもたらす。高回線速度のストレージ又はデータトラフィックがＩＰ上で搬送されることを可能にするために、従来はホストカーネル又はドライバソフトウエアによって維持されていた様々な接続（セッション）に対する伝送制御ブロック情報を維持することが必要になる。本特許で使用されるように、用語「ＩＰセッション」は、ＩＰ上で実行されるセッション志向プロトコルに対するセッションを意味する。例としては、ＴＣＰ／ＩＰ及びＳＣＴＰ／ＩＰなどがある。各パケットに対するセッション情報へのアクセスは、大幅な処理オーバーヘッドを付加する。説明したアーキテクチャは、このオーバーヘッドを大幅に低減する高性能メモリサブシステムを作り出す。このプロセッサのアーキテクチャは、主としてコマンド又はデータ転送終了境界でのホストプロセッサに対する割り込みを最小限にする知的フロー制御の機能を提供する。

【０００９】

今日、セキュリティを有するＴＣＰ／ＩＰプロセッサは提供されていない。

説明したプロセッサのアーキテクチャはまた、一体化されたセキュリティ機能を提供する。ストレージトラフィックが、ネットワーク上でサーバからＳＡＮ又は他のストレージシステム内のストレージアレイまで搬送される時、それは、直接取付けられたストレージシステムが処理する必要のない様々なセキュリティの脆弱性に露出される。このプロセッサは、ストリーム内でストレージトラフィックの暗号化と暗号解読を可能にするので、高回線速度を可能にし、同時にストレージデータトラフィックの機密性を提供する。

【００１０】

ネットワークトラフィックの分類は、深いパケットの検査及び処理のための僅かなサイクルを残してパケットプロセッサ上で利用可能な処理サイクルの半分までを消費する別のタスクである。ＩＰベースのストレージトラフィックは、プロトコルの性質上、高速で待ち時間が少ない深いパケット処理を要求する。説明したＩＰプロセッサは、プログラマブル分類エンジンを提供することにより、分類のオーバーヘッドを大幅に低減する。

【００１１】

ストレージ容量とストレージネットワークにおける大きな成長は、ＩＴ部門の主なコストアイテムとしてストレージ領域管理を作り出している。管理コストを抑えるためにポリシーベースのストレージ管理が要求される。説明したプログラマブル分類エンジンは、パ

ケット、トランザクション、フロー、及びコマンド境界に対して強制することができるストレージポリシーの配備を可能にする。これは、ストレージ領域の管理コストを大幅に改善することになる。

【００１２】

プログラマブルＩＰプロセッサアーキテクチャはまた、顧客特定アプリケーションの配備を可能にするのに十分なヘッドルームを提供する。これらのアプリケーションは、例えば、ネットワーク管理、ストレージファイアウォール又は他のセキュリティ機能、帯域幅管理、サービス品質、仮想化、性能モニタリング、ゾーニング、及びＬＵＮマスキングなどの複数のカテゴリに属するであろう。

【発明を実施するための最良の形態】

【００１３】

本発明人は、ソフトウエアＴＣＰ／ＩＰスタックの厳しい性能の影響からホストプロセッサを解放するために、ハードウエアにＴＣＰ／ＩＰスタックを実施する新しい高性能で待ち時間が短い方法を提供する。このハードウエアＴＣＰ／ＩＰスタックは、次に、付加的な処理要素とインタフェースで接続され、高性能で待ち時間が短いＩＰベースのストレージアプリケーションを使用可能にする。

これは、ＴＣＰ／ＩＰの終了、高性能で待ち時間が短いＩＰストレージ機能、遠隔ＤＭＡ（ＲＤＭＡ）機能、セキュリティ機能、及びプログラマブル分類及びポリシー処理機能などの恩典をもたらすために、様々な形式で実行することができる。以下は、これを実行することができるいくつかの実施形態である。

【００１４】

サーバ

説明するアーキテクチャは、ＴＣＰ／ＩＰソフトウエアの１つ又は複数のホストサーバプロセッサ及び性能オーバーヘッドを緩和するハードウエアベースのＴＣＰ／ＩＰ機能を提供する高性能サーバ環境に組み込むことができる。ＩＰプロセッサは、ハードウエアＴＣＰ／ＩＰとの高性能ネットワーキングインタフェースを提供する、サーバチップセットに対する対照プロセッサとすることができる。サーバは、ブレードサーバ、アプライアンスサーバ、ファイルサーバ、シンサーバ、クラスター化サーバ、データベースサーバ、ゲームサーバ、グリッド計算サーバ、ＶＯＩＰサーバ、無線ゲートウェイサーバ、セキュリティサーバ、ネットワーク取付ストレージサーバ、又は従来のサーバのような様々な形状因子にすることができる。本実施形態は、サーバマザーボード上に高性能ネットワークインタフェースの作成を可能にすると考えられる。

【００１５】

サーバチップセットに対する対照プロセッサ

サーバ環境はまた、高性能ＴＣＰ／ＩＰ及び／又はＲＤＭＡ機能に加えて、説明したプロセッサの高性能ＩＰストレージ処理性能に影響を及ぼすことがある。そのような実施形態では、プロセッサは、サーバプロセッサからのＴＣＰ／ＩＰオフロードに加えて高性能ネットワークストレージＩ／Ｏ性能を提供する、サーバチップセットに対する対照プロセッサとすることができる。この実施形態は、マザーボード上に高性能ＩＰベースのネットワークストレージＩ／Ｏの作成を可能にすると考えられる。換言すれば、それは、マザーボード上の「ＩＰ　ＳＡＮ」を使用可能にするであろう。

【００１６】

ストレージシステムチップセット

プロセッサはまた、ストレージシステム内のチップセットの対照として使用することができ、これは、ストレージネットワーキング環境でストレージデータサーバの機能を実行するストレージアレイ（又は、他の適切なストレージシステム又はサブシステム）コントローラとすることができる。ＩＰベースＳＡＮにおいてネットワーキングするために、プロセッサは、ストレージアレイコントローラに対してＩＰネットワークストレージ機能を提供すると考えられる。その構成は、ストレージアレイにアクセスして別のストレージ中枢機能を提供するために、システム内に付加的な機能を持ったサーバ環境の構成と同等に

することができる。

【００１７】

サーバ／ストレージホストアダプタカード

ＩＰプロセッサはまた、高速度ＴＣＰ／ＩＰネットワーキングを提供するサーバホストアダプタカードに組み込むことができる。同じアダプタカードはまた、ＩＰベースストレージネットワークに対して、高速度ネットワークストレージ機能を提供することができる。アダプタカードは、従来のサーバで使用することができ、同様にブレードサーバ構成においてブレードとして使用することもできる。プロセッサはまた、ＩＰベースストレージネットワーキング機能を提供するストレージアレイ（又は、別のストレージシステム又はサブシステム）のフロントエンド内のアダプタで使用することができる。

【００１８】

プロセッサチップセット構成要素

ＴＣＰ／ＩＰプロセッサは、ＴＣＰ／ＩＰオフロード機能を提供するプロセッサチップセットの内部に組み込むことができる。そのような構成は、ハイエンドサーバ、ワークステーション、又は高速ネットワークとインタフェースで接続する高性能パーソナルコンピュータに使用することができる。そのような実施形態はまた、ＩＰストレージ又はＲＤＭＡ機能、又は本発明の組合せを含むことができ、チップセットに組み込まれたＩＰベースストレージネットワーキング、及び／又はＲＤＭＡ機能を有するＴＣＰ／ＩＰを提供する。説明したアーキテクチャの複数の機能の使用は、機能要件、開発時間とコスト、及びシリコンダイのコストなどのトレードオフとして、上記又は他の実施形態の他の機能の使用と独立して行うことができる。

【００１９】

ストレージ又はＳＡＮシステム又はサブシステム切替回線カード

ＩＰプロセッサはまた、高性能で待ち時間が少ないＳＡＮ切換システム（又は、別のシステム又はサブシステム）回線カードを作り出すために使用することができる。プロセッサは、回線カードへの／からのＩＰベースストレージトラフィックを終了させたり開始させたりするメインプロセッサとして使用することができる。このプロセッサは、ホストのように作動することができる切換システムのファブリックコントローラを使用して作動し、切換システム内に存在する転送情報ベースによって判断された適切な切替回線カードに対するＩＰ宛先に基づいて、終了したストレージトラフィックを移送する。そのような切換システムは、純粋にＩＰベースネットワーキングをサポートすることができ、又は、マルチプロトコルサポートをサポートすることもでき、ファイバチャンネルのような他のデータセンターのＳＡＮファブリックに加えて、ＩＰベースＳＡＮとインタフェースで接続することを可能にする。ゲートウェイコントローラシステムの内部には、ＬＡＮ又はＷＡＮからＩＰストレージトラフィックを終了させ、ストレージトラフィックをＳＡＮに搬送するために新しいセッションを開始させる非常に類似の構成が存在することができ、上述のＳＡＮは、ＩＰベースＳＡＮか、又は、より可能性があるのは、ファイバチャンネルのようなデータセンター内部の他のファブリックから構成されたＳＡＮである。プロセッサはまた、ＳＡＮゲートウェイコントローラに組み込むことができる。

【００２０】

ストレージアプライアンス

ストレージネットワークの管理コストは急速に増加している。ネットワーク及びストレージ容量の大幅な成長を管理する能力は、ストレージ領域の管理機能をもたらす特別なアプライアンスを作り出すことを要求する。高性能ＩＰベースＳＡＮに対して説明した管理アプライアンスは、その機能を達成することができるように、ＴＣＰ／ＩＰパケット内部で移送されたストレージトラフィック上で高性能ＩＰプロセッサを実行する。これらのシステムは、ポリシーベースの管理及び実施機能を提供するために、深いパケットの検査を実行してＩＰトラフィック内のストレージペイロードを抽出するための高性能プロセッサを要求する。セキュリティを有するプログラマブル分類及びポリシーエンジンは、説明した高速ＴＣＰ／ＩＰ及びＩＰストレージエンジンと共に、本特許に説明するこれらのアプ

ライアンスと他の実施形態が、高回線速度かつ短い待ち時間で深いパケット検査及び分類を実行し、パケット毎のベースで必要なポリシーを適用することを可能にする。更に、これらの機能は、仮想化、ポリシーベース管理、セキュリティ実施、アクセス制御、侵入検出、帯域幅管理、トラフィックシェーピング、サービス品質、迷惑メール防止、ウイルス検出、暗号化、暗号解読、ＬＵＮマスキング、ゾーニング、リンク集約、及びストレージ領域ネットワークトラフィックに対してインバンドである類似のもののような、その機能を実行することができるストレージ管理アプライアンスの作成を可能にする。同様のポリシーベースの管理、及びセキュリティ作動又は機能性はまた、本特許に説明する他の実施形態の内部でサポートすることができる。

【００２１】

クラスター化された環境

システム性能と、クラスター化データベースなどのようなアプリケーションに対する拡張容易性とを増強するために、クラスター化環境ではサーバシステムが使用される。高性能クラスターサーバ上で作動しているアプリケーションは、プロセス間通信に対して高速でデータを共有する能力を要求する。従来のソフトウエアＴＣＰ／ＩＰネットワーク上のクラスタープロセッサ間でのこのプロセス間通信トラフィックの移送は、厳しい性能のオーバーヘッドを受ける。従って、ファイバチャンネルのような専用ファブリックがそのような構成では使用されてきた。しかし、通信プロセスメモリ間で直接のメモリアクセスを可能にするＴＣＰ／ＩＰベースのファブリックは、ファイバチャンネルのような専用ファブリックに変換することなく、任意のＴＣＰ／ＩＰネットワーク上で作動するアプリケーションによって使用される。説明したＩＰプロセッサは、その高性能ＴＣＰ／ＩＰ処理機能とＲＤＭＡ機能を使用してクラスターサーバ環境に組み込むことが可能であり、高性能で短い待ち時間の直接メモリという恩典をメモリデータ転送にもたらす。この実施形態はまた、グローバルなクラスター化を作り出すために使用することができ、同様にグリッドコンピュータ及びグリッドネットワーク内のデータ転送を可能にするために使用することができる。

【００２２】

付加的な実施形態

プロセッサアーキテクチャは、部分的にソフトウエアで、また部分的にハードウエアで実施することができる。性能の必要性とコストの推測は、本発明のシステムアーキテクチャ全体のハードウエア及びソフトウエア分割に対するトレードオフを推し進める可能性がある。このアーキテクチャをハードウエア及びソフトウエア分割と共にチップセットの組み合わせとして、又は分割とは独立して実行することも可能である。例えば、セキュリティプロセッサと分類エンジンは、別々のチップ上にあり、同様の機能を提供することができる。これは、開発及び製造コストを含めて、シリコンコストの廉価なＩＰプロセッサをもたらすことができるが、いくつかの例では、システムも含めた要素を増加させ、設置面積とソリューション全体のコストを増大させる可能性がある。セキュリティ及び分類エンジンも別々のチップとすることができるであろう。本明細書で使用されるように、チップセットは、複数チップのチップセット、又は単一チップだけを含むチップセットを意味することができ、それはアプリケーションに依存する。

【００２３】

ストレージフローコントローラと待ち行列は、ホスト上のソフトウエア内に維持することができ、又はチップセット内の別のチップの一部になることもできる。従って、高性能プロセッサの将来の出現に伴って要求される高性能ＩＰベースストレージとＴＣＰ／ＩＰオフロードアプリケーションとを達成するために、このアーキテクチャを分割する複数の方法が実現可能である。ｉＳＣＳＩと関連してストレージエンジンが説明されたが、制御プロセッサと共に、ＴＣＰ／ＩＰ及びストレージエンジンのプログラム可能性、分類子のプログラム可能性、及びストレージフローコントローラを使用して、ｉＦＣＰ、ＦＣＩＰ、及びその他のような他のＩＰストレージプロトコルを適切なファームウエアで実施することができる。ｉＳＣＳＩ作動はまた、ＩＰストレージ作動とすることができる。高性能

ＩＰプロセッサコアは、より低い回線速度の複数の入力出力ポートと連結し、全処理量を適合させてマルチポートＩＰプロセッサ実施形態を同様に作り出すことができる。

【００２４】

ストレージエンジンを使用しないでメインプロセッサからオフロードする高性能ＴＣＰ／ＩＰに対してこのアーキテクチャを使用することが実現可能である。これは、データ及び遠距離通信アプリケーションの次世代の高性能ネットワークに対してシリコンとシステムのソリューションをもたらすことができる。ＴＣＰ／ＩＰエンジンは、アプリケーション特定のパケット加速器を使用して増強し、コアのアーキテクチャを利用してこのプロセッサの新しい味を引き出すことができる。本発明の他の機能に加えて、ストレージエンジンをファイアウォールエンジン、又はルート検索エンジン、又は遠距離通信／ネットワーク加速エンジンのような別のアプリケーション特定の加速器と取り替え、このプロセッサアーキテクチャのターゲットを遠距離通信／ネットワーキング及び他のアプリケーションにすることができる。

【００２５】

詳細な説明

ストレージのコストと需要は、過去数年間に亘って速いペースで増加してきた。これは、近い将来に亘って同じ割合で成長することが見込まれる。ｅビジネスの出現に伴って、サーバ又はシステムの機能停止時間に関係なく、任意の時間の任意の場所でのデータ使用可能度が重要になっている。これは、大きな必要性を駆り立て、サーバ接続ストレージをネットワークに移動させ、ストレージの統合、データの利用可能性、及びデータ管理の容易性をもたらしている。ストレージエリアネットワーク（ＳＡＮ）は、現在では大部分がファイバチャンネル技術に基づいており、ＴＣＰ／ＩＰ技術と比べてハードウエア志向のスタックを使用した短い待ち時間と高性能というような様々な恩典をもたらしている。

【００２６】

一部のシステムは、データストリームを移送するように設計されたＩＰ上でブロックストレージトラフィックを移送する。データストリームは、ＩＰの上で実行されるように階層化された伝送制御プロトコル（ＴＣＰ）を使用して移送される。ＴＣＰは、オペレーティングシステム内のソフトウエアで実施される確実な接続志向プロトコルである。ＴＣＰ／ＩＰソフトウエアスタックは、将来配置されるであろう高回線速度を処理するには低速である。新しいハードウエアのソリューションは、ストレージとネットワークトラフィックを搬送するためにＴＣＰ／ＩＰスタックを加速し、ＦＣベースのソリューションに対して競争力があることになる。

【００２７】

高性能サーバ、ワークステーション、ストレージコントローラ、及びネットワークトラフィックで普及しているストレージプロトコルは、約２０年前から存在するＳＣＳＩプロトコルである。ＳＣＳＩアーキテクチャは、階層化プロトコルアーキテクチャとして構築されている。図１は、イニシエータ（ブロック１０１）とターゲットサブシステム（ブロック１０２）内の様々なＳＣＳＩアーキテクチャ階層を示す。本特許で使用されるように、用語「イニシエータ」及び「ターゲット」は、データ処理装置又はそれらを含むサブシステム又はシステムを意味する。用語「イニシエータ」及び「ターゲット」はまた、クライアント又はサーバ又はピアを意味することができる。同様に、用語「ピア」は、対等の処理装置又はそれらのサブシステム又はシステムを意味することができる。「遠隔ピア」は、世界中に亘って又は部屋中に亘って配置されたピアとすることができる。

【００２８】

図１のイニシエータとターゲットサブシステムは、クライアントサーバ要求と応答処理を提供するために使用されるＳＣＳＩアプリケーションプロトコル層（ブロック１０３）を使用して互いに対話する。それは、ディスクアレイやテープドライブなどのような多くの形態をとることができるイニシエータ及びターゲットの大容量ストレージ間で装置サービス要求と応答を同様に提供する。従来、ターゲットとイニシエータは、ＳＣＳＩプロトコル（ブロック１０４）を搬送するＳＣＳＩバスアーキテクチャを使用して相互接続され

た。ＳＣＳＩプロトコル層は、ＳＣＳＩアプリケーションプロトコルを使用して、クライアントとサーバの対話を互いに可能にする移送層である。移送層は、上部層のプロトコルとアプリケーションが移送プロトコルを独立させておくことができるように、上部層に対して同じ意味を呈するべきである。

【００２９】

図２は、ベースとなる移送層に加えてＳＣＳＩアプリケーション層を示す。ＩＥＴＦ基準の追跡プロトコル、すなわち、ｉＳＣＳＩ（ＩＰ上のＳＣＳＩ）は、ＩＰベースのストレージ移送プロトコルを提供するための試みである。ＦＣＩＰ（ＩＰにカプセル化されたＦＣ）、ｉＦＣＩＰ（ＩＰ上のＦＣ）などを含む別の同様の試みがある。移送機構としてのＴＣＰ／ＩＰに加えて、これらのプロトコルの多くは、図２に示す方法と同様の方法で階層化される。図２に示すように、ｉＳＣＳＩプロトコルサービス層（ブロック２０４）は、ＳＣＳＩアプリケーション層（ブロック２０３）に対して層状インタフェースを形成する。ｉＳＣＳＩは、基準によって定められるように、ｉＳＣＳＩプロトコルデータユニット（ＰＤＵ）としてＳＣＳＩコマンド及びデータを搬送する。これらのプロトコルデータユニットは、次に、ＴＣＰ／ＩＰ（ブロック２０５）などを使用してネットワーク上で移送することができる。この基準は、「ｉＳＣＳＩ　ＰＤＵ」を運ぶ基本的な移送を実施するための手段を特定しない。図２は、「ｉＳＣＳＩ　ＰＤＵ」に対して移送を提供するＴＣＰ／ＩＰ上で階層化されたｉＳＣＳＩを示す。

【００３０】

ｉＳＣＳＩのようなＩＰベースのストレージプロトコルは、ソフトウエアベースのＴＣＰ／ＩＰスタックに加えて、ソフトウエア内で階層化することができる。しかし、そのような実施は、それに加えてソフトウエアＴＣＰ／ＩＰと階層化されたストレージプロトコルから発生する厳しい性能ペナルティに苦しむことになる。それらの実施は、ホストプロセッサの性能に厳しく影響し、１Ｇｂｐｓを超える回線速度での他の全てのタスクに対してプロセッサを使用不能にするであろう。従って、ハードウエア内でＴＣＰ／ＩＰスタックを実施し、ホストプロセッサを解放し、その上にストレージプロトコルを構築することができるであろう。ｉＳＣＳＩと同様にストレージプロトコルは、ホストプロセッサ上で実施されるソフトウエア内に構築することができるか、又は本特許に説明するようにハードウエアの実施によって加速することができる。ソフトウエアｉＳＣＳＩスタックは、受信ＴＣＰセグメントからＰＤＵを抽出してそれらに作用することができるように、ホストプロセッサに対して多くの割り込みを呈することになる。そのような実施は、ソフトウエアベースＴＣＰスタックが実施したであろう同様の理由で、厳しい性能ペナルティに苦しむことになる。説明したプロセッサは、ホストプロセッサに対する性能ペナルティと得られる待ち時間の影響とを排除又は大幅に低減する、ＴＣＰ／ＩＰベースネットワーク上でストレージプロトコルを移送するための高性能で待ち時間の短いアーキテクチャを提供する。

【００３１】

図３は、ＯＳＩネットワーキングスタックとして参照されたファイバチャンネルに対するＴＣＰ／ＩＰスタックの比較を示す。ＴＣＰ／ＩＰスタック（ブロック３０３）には、本特許の「発明の開示」の節で上述したように、ホスト上のソフトウエアの実施に起因する性能上の問題がある。それに比べて、ファイバチャンネル（ブロック３０４）のような又は他の専用ネットワーキングプロトコルは、ハードウエア内で実施されるように設計される。ハードウエアでの実施により、ネットワーキングソリューションがＩＰベースソリューションよりも高性能になる。しかし、ＩＰのユビキタス性とＩＴユーザ及び開発者の観点からのＩＰの習熟性とは、広範囲の開発に対してＩＰを更に適切なものにする。これは、ＴＣＰ／ＩＰに起因する性能ペナルティが、他の競争力のある専用プロトコルに相当するまで減少した場合に達成することができる。図４は、図示の他の専用プロトコルと競合することができるようになるために、ＴＣＰ／ＩＰ（ブロック４０３）に使用されるハードウエア及びソフトウエア内で階層化するプロトコルのレベルを示す。

【００３２】

図５は、ハードウエアベースのＴＣＰ／ＩＰと、本特許のストレージプロトコルの実行を使用するホストオペレーティングシステムスタックとを示す。プロトコルは、それがホストオペレーティングシステムスタック（ブロック５１３）内にオペレーティングシステム層がその上で不変のまま導入されように実施される。これは、ＳＣＳＩアプリケーションプロトコルを一切変えないで作動させる。ドライバ層（ブロック５１５）と、ＩＰベースのストレージインタフェースのための下のスタック（ブロック５０１）とは、非ネットワーキングＳＣＳＩインタフェース（ブロック５０６及び５０３）又はファイバチャンネルインタフェース（ブロック５０２）と同等のインタフェースを表すことになる。

【００３３】

図６は、ソフトウエアＴＣＰ／ＩＰスタックに包含されたデータ転送を示す。ＴＣＰ／ＩＰスタックのそのような実施は、データ転送のメモリコピーからの非常に大きな性能ペナルティを有する。図は、クライアントとサーバネットワーキングスタックの間のデータ転送を表す。クライアントからユーザまで又はその反対に移送する必要があるユーザレベルのアプリケーションバッファ（ブロック６０１）は、図示の様々なレベルのデータ転送を通過する。発信元上のユーザアプリケーションバッファは、ＯＳカーネルのスペースバッファ（ブロック６０２）にコピーされる。このデータは、次に、ネットワークドライバのバッファ（ブロック６０３）にコピーされ、そこからネットワークインタフェースカード（ＮＩＣ）又はホストバスアダプタ（ＨＢＡ）にＤＭＡ転送される。バッファコピー作動は、ホストプロセッサを伴い、大事なプロセッササイクルを使い果たす。更に、転送中のデータは、ホストからの付加的な計算サイクルを使い果たし、ホスト上のチェックサム計算を通って進行する。ホスト上のシステムメモリへの又はからのデータの複数回の移動も、同じくメモリ帯域幅の妨害を作り出す。ＮＩＣ／ＨＢＡに移送されたデータは、次にネットワーク上に送られ（ブロック６０９）、宛先システムに到着する。宛先システムでは、データパケットは、ホストと反対方向であるが同様のバッファコピーとチェックサム作動に従って、ソフトウエアネットワーキングスタックを通ってトラバースする。ＴＣＰ／ＩＰスタックのそのような実施は、ブロックストレージデータの移送に対して、及び発信元と宛先の間で大量のデータが移送されるような場合のクラスター化アプリケーションに対しては非能率である。

【００３４】

図７は、本特許に説明するアーキテクチャの遠隔の直接メモリアクセス（ＲＤＭＡ）機能を可能にするという特徴を有するイニシエータ及びターゲットでのネットワーキングスタックを示す。以下は、ＲＤＭＡ能力又はＲＤＭＡ機構又はＲＤＭＡ機能と呼ぶことができる。そのようなシステムでは、イニシエータ又はターゲット上で実行されるアプリケーションは、メモリ領域（ブロック７０２）を登録し、この領域は、実質的にホストの介入なしでそのピアにＮＩＣ／ＨＢＡから直接利用可能にされる。これらのアプリケーションはまた、それらのピアにＲＤＭＡ（ブロック７０８）に関して利用可能であるメモリ領域について知らせるであろう。通信の両方のピアがＲＤＭＡ機構を使用する準備ができた状態で、ＲＤＭＡ領域からのデータの移送は、ピア内のＮＩＣ／ＨＢＡハードウエアがＲＤＭＡ機能を実行すれば、実質的なホストの介在がなくても発信元から宛先まで基本的にゼロコピーのオーバーヘッドで起こり得る。発信元、すなわちイニシエータは、その要求のそのピアに特定ＲＤＭＡ使用可能バッファを読み取るか又は書き込むように通知し、次に、宛先、すなわちターゲットにそのＲＤＭＡバッファへ／からデータをプッシュ又はプルさせる。イニシエータ及びターゲットＮＩＣ／ＨＢＡは、次に、本特許に説明するＴＣＰ／ＩＰハードウエア実施、ＲＤＭＡ７０３、ＴＣＰ／ＩＰオフロード７０４、ＲＤＭＡ７０８、及びオフロード７０９を使用して、ホストプロセッサの実質的な記入なしで互いの間でデータを移送し、それによってプロセッサのオーバーヘッドを大幅に低減するであろう。この機構は、ホストプロセッサ上のＴＣＰ／ＩＰ処理のオーバーヘッドを大幅に低減し、図６に示すデータ転送のための複数バッファのコピーの必要性を排除すると考えられる。ＲＤＭＡ実行可能システムは、従って、高速又は低速の如何に関わらず、システムがそのピアの機能の妨害を作成せずにデータ転送を実行することを可能にする。ＩＰソリュ

ーションの上でストレージ内のこのプロセッサに実施されたＲＤＭＡ機能は、通常はデータ転送の開始時と終了時を除いて、ホストの介在を排除する。これは、ターゲット及びイニシエータの両方のシステムでホストプロセッサを解放し、各パケット到着又は転送時に介入されることなく有用なタスクを実行する。ＲＤＭＡの実行はまた、システムを安全にし、認可されていないアクセスを阻止することを可能にする。これは、領域のＩＤと共にアクセス制御キーを有するＨＢＡ／ＮＩＣを使用してエクスポートされたメモリを登録することによって達成される。ＨＢＡ／ＮＩＣは、遠隔のホストからＲＤＭＡバッファへのメモリ領域要求のアドレス変換を実行し、セキュリティキー検証のような安全作動を実行し、次にデータを移送させる。この処理は、ＨＢＡ／ＮＩＣに常駐するか、又は、例えばマザーボード上のホストプロセッサに対するコンパニオンプロセッサとして、本発明のプロセッサのホストプロセッサから切り離されて実行される。この機能はまた、クライアントサーバアプリケーションと同様に、厳しいクラスター化アプリケーションに対する大量のデータ転送に使用することができる。発信元すなわちイニシエータと宛先すなわちターゲットの間で大量のデータを移送するリアルタイムのメディアアプリケーションは、これによって恩典を得ることができる。

【００３５】

図８は、ソフトウエアで実行されたホストファイルシステムとＳＣＳＩスタックとを示す。上述の通り、ＩＰベースのストレージスタック（ブロック８０５、８０６、８０７、８０８、及び８０９）は、ＳＣＳＩ層（ブロック８０３及び８０４）に対して、ＳＣＳＩ移送層（ブロック８１１）又はファイバチャンネル移送（ブロック８１０）によって提供されたような一貫したインタフェースを呈示すべきである。この図は、性能に影響されやすいブロックデータを移送するように設定されていないＩＰの様々な問題によって課せられるもの以外に、システムレベルからＩＰベースのストレージの実行に課せられる高レベルの要求を示す。

【００３６】

図９は、図８に示すｉＳＣＳＩスタックをより詳細に示す。ｉＳＣＳＩスタック（ブロック８０５から８０９）は、ＳＣＳＩコマンド統合層（ブロック８０３及び８０４）に対して、この層とその上の他の層の挙動が変わらないように、ＯＳで定められたドライバインタフェースレベルの機能性を提供すべきである。図９は、ＩＰストレージ機能を提供するために実施されると考えられる一組の機能を示す。ｉＳＣＳＩの機能性を提供する機能は、関連する組の機能に分類されるが、どの当業者も認めるように、これらの多くの変形が存在する可能性がある。標準的（例えば、ターゲット及びイニシエータのログイン及びログアウト）機能（ブロック９１６）や接続の確立及び取外し機能（ブロック９０５）を満足するように要求される一組の機能がある。図は、「ＯＳ　ＳＣＳＩ」ソフトウエアスタックがｉＳＣＳＩ装置（ブロック９１６）を見つけ、オプション／パラメータ（ブロック９０３及び９０９）を設定して取得し、装置を始動して（ブロック９１３）装置を解放する（ブロック９１１）ことを可能にする機能を示す。上述の制御機能以外に、ｉＳＣＳＩの実施は、待ち行列（９１２及び９１７）を通じて大量のデータの移送機能を提供し、ｉＳＣＳＩ基準によって規定されたＰＤＵを移送する。ｉＳＣＳＩスタックはまた、直接データ転送／配置（ＤＤＴ）又はＲＤＭＡ機能又はその組合せを含むことができ（ブロック９１８）、これらは、イニシエータ及びターゲットシステムによって使用することができ、ストレージ及び他の大量のブロックデータ転送を含めて、実質的にゼロバッファのコピーとホスト介入のないデータ転送とを実行する。ＳＣＳＩコマンド及びこれらに関連するブロックデータ転送は、説明したプロセッサ上で実行されるコマンド待ち行列（ブロック９１２及び９１７）として実施される。ホストは、主にコマンドの終了時に割り込まれる。終了したコマンドは、ホストの都合のよい時に作用するように、ホストに対する待ち行列に入れられる。図は、ｉＳＣＳＩプロトコル層と、同じくプロセッサから切り離されて実行される本明細書に説明したＩＰプロセッサシステム上のＴＣＰ／ＩＰスタック（ブロック９０７及び９０８）上で階層化されたドライブ層とを示す。

【００３７】

図１０は、説明したＩＰプロセッサシステムで実施されるＴＣＰ／ＩＰスタック機能性を示す。これらの機能は、このプロセッサの高性能ＴＣＰ／ＩＰ実施を利用するために、上部層プロトコル機能に対するインタフェースを提供し、直接「ＯＳ　ＴＣＰ／ＩＰ」バイパス、ＲＤＭＡ又はネットワークソケット直接機能、又はその組合せから恩典を得ることができる他のアプリケーションと共にＩＰストレージトラフィックを搬送する。ＴＣＰ／ＩＰスタックは、上部層データ（ブロック１０１７及び１０３１）とコマンドＰＤＵを送受信し、移送接続と取外し機能（ブロック１０２１）を確立し、データ転送機能やチェックサム機能（ブロック１０１９）、並びにエラー処理機能（ブロック１０２２）やセグメント化及び順序付け及びウィンドウ操作作動（ブロック１０１３）を送受信するための機能を提供する。チェックサム確認／作成のような特定の機能は、データ転送の全てのバイトに関連するのに対して、データパケットを移送して伝送制御ブロック又はセッションデータベースを更新するいくつかの機能は、データ転送の各パケットのために呼び出される。セッションＤＢ（ブロック１０２５）は、ＴＣＰ／ＩＰ状態情報と共に、アクティブセッション／接続に関する様々な情報を維持するために使用される。ＴＣＰ層は、基準による要求に応じてＩＰ機能を提供するＩＰ層に加えて構築される。この層は、パスＭＴＵによってパケットをフラグメント化／フラグメント解消する機能（ブロック１０３３）を提供し、例えば、ＩＣＭＰ（ブロック１０２９）のようなエラーを通信するために必要な他の機能とのインタフェースと同様に、ルート及び転送情報（ブロック１０３２）を提供する。ＩＰ層は、イーサネット層又は他のメディアアクセス層技術とインタフェースで接続し、ネットワーク上にＴＣＰ／ＩＰパケットを移送する。この説明の様々な図には、より低い層がイーサネットとして示されているが、例えば、ＭＡＮ／ＷＡＮ上のＳＯＮＥＴに亘ってパケットを移送するためのＳＯＮＥＴのような別の技術とすることができる。イーサネットはまた、類似のアプリケーションで使用されるが、例えば、ＬＡＮ及び専用ローカルＳＡＮ環境内で更に多く使用することができる。

【００３８】

図１１は、ｉＳＣＳＩデータフローを示す。図は、データフローの受信及び送信パスを示す。ホストのＳＣＳＩコマンド層は、ｉＳＣＳＩドライバと協働し、両方ともブロック１１０１に示されているが、図２６により詳細に見られるストレージフローコントローラ内のコマンドスケジューラ（ブロック１１０８）に対して処理されるようにコマンドをスケジュールに入れるであろう。コマンドスケジューラ１１０８は、図１７により詳細に示すプロセッサ内で作動するために、新しいコマンドをスケジュールに入れる。既存の接続を用いてターゲット装置と定められた新しいコマンドは、その既存の接続に対する待ち行列に入れられる（ブロック１１１１）。ターゲット装置に対する接続が存在しない場合は、新しいコマンドは、割り当てられていないコマンド待ち行列に対してエンキューされる（ブロック１１０２）。図４７とブロック９０５及び１００６に示すようなセッション／接続の確立処理が次に呼び出され、ターゲットに接続される。接続が確立された状態で、待ち行列１１０２からの対応するコマンドは、図に示すように、コマンドスケジューラ１１０８により新しく作り出された接続コマンド待ち行列１１１１に対してエンキューされる。コマンドが実行ステージに達した状態で、コマンドが読取又は書込トランザクション次第で、受信１１０７又は送信１１０９パスがアクティブにされる。コマンドが移送される接続／セッションの状態は、次に説明するように、セッションデータベース内でのコマンド実行の進行を記録するために使用される。データ転送に関連するバッファは、転送が終了する時までロックすることができる。イニシエータとターゲットの間でデータを送信するためにＲＤＭＡ機構が使用される場合、適切な領域バッファ識別子、アクセス制御キー、及び関連ＲＤＭＡ状態データは、選択された実施次第でプロセッサ上のメモリに維持され、同じくオフチップメモリにも維持することができる。複数のＴＣＰセグメントに亘ることがあるコマンドに関連するデータ転送が終了すると、コマンド実行のステータスは、次に適切な処理を行うホストのＳＣＳＩ層に転送される。これは、アプリケーションや統計データの更新などに対するデータ転送のために使用されているバッファの解放を含むことができる。転送中は、「ｉＳＣＳＩ　ＰＤＵ」は、伝送コマンドエンジン（ブロック

１１１０）と協働する伝送エンジン（ブロック１１０９）によって伝送され、このコマンドエンジンは、ＰＤＵを変換し、ストレージプロセッサに対するＤＭＡを使用し、ホストメモリからのアプリケーションバッファを検索するなどの適切な作動を実行し、進行に伴って更新されたｉＳＣＳＩ接続データベース内のストレージコマンドフロー情報を維持する。本特許で使用されるように、用語「エンジン」は、エンジンの機能又は使用に適するデータプロセッサ又はデータプロセッサの一部とすることができる。同様に、受信エンジン（ブロック１１０７）は、受信コマンドを新しい要求、応答、エラー、又は他のコマンド、又は適切に作用されるべきデータＰＤＵに翻訳する。コマンドエンジン（ブロック１１０６）と協働するこれらの受信エンジンは、読取データ又は受信データを、直接データ転送／配置、又はｉＳＣＳＩセッションテーブル内のセッションのために維持されたＲＤＭＡ制御情報を通じて、適切に割り当てられたアプリケーションバッファにルーティングする。コマンドが終了すると、それぞれのバッファに対する制御（ブロック１１０３及び１１１２）は、使用するアプリケーションのために解放される。受信及び送信エンジンは、図２３のＴＣＰ／ＩＰプロセッサ又は図２４のＩＰプロセッサから見る時にグローバルメモリとして見ることができる、セッションデータベースエントリ１７０４に記録されたセッション情報と協働するこのＩＰプロセッサの図１７のＳＡＮパケットプロセッサ１７０６（ａ）から１７０６（ｎ）とすることができる。ブロック１７０４と図１７の１７０８のセッションデータベースの一部分に関連して以下により詳細に説明される、セッションデータベースによって提供された適切なストレージフローコンテキストを使用して、同じエンジンを異なるパケットとコマンドのために再使用することができる。明確にするために、ＩＰネットワークアプリケーションプロセッサ、ＩＰストレージプロセッサ、ネＩＰストレージネットワークアプリケーションプロセッサ、及びＩＰプロセッサという用語は、アプリケーション次第で同じエンティティとすることができる。ＩＰネットワークアプリケーションプロセッサコア及びＩＰストレージネットワークアプリケーションプロセッサコアは、アプリケーション次第で同じエンティティとすることができる。

【００３９】

同様に、制御コマンドは、送信パスを使用することができ、一方、受信応答は、受信パスを使用するであろう。類似のエンジンは、ターゲットと同様にイニシエータ上に存在することができる。データフローの方向は、それがイニシエータかターゲットかによって異なる。しかし、ターゲットでは付加的な段階を伴うが、本質的にはイニシエータとターゲットの両方に類似のデータフローが存在する。例えば、書込コマンドデータを取得するのに必要なバッファを保証するために、ターゲットは、付加的な作動を実行する必要があるか又はイニシエータにデータが供給される前に読取データを準備する必要があるであろう。中間装置の場合は同様の例が存在するであろうが、そのようなスイッチ又はアプライアンスとすることができる装置では、あるレベルの仮想化又はフレーム濾過、又は、一方でセッションの終了を他方でセッションの開始を要求する場合がある類似の別の作動を実施することができる。この機能性は、このアーキテクチャによってサポートされるが、当業者には公知の範囲なので図には明確には示されていない。

【００４０】

図１２から図１５は、移送セッションに関する特定のプロトコル情報と、その情報をメモリ内のデータベースに保存することができる方法とを示す。

図１２は、ｉＳＣＳＩプロトコル及び関連ＴＣＰ／ＩＰ接続のために維持されたデータ構造を示す。各ｉＳＣＳＩセッション（ブロック１２０１）に属するデータは、基本的にイニシエータとターゲット接続のネクサスであり、適切な接続（ブロック１２０２）上で搬送される。従属コマンドは、同じ接続の待ち行列上でスケジュールに入れられ、コマンドの順序を維持する（ブロック１２０３）。しかし、関連のないコマンドは、別の移送接続に割り当てることができる。実施がセッション当たり１つの接続だけをサポートする場合、全てのコマンドを同じ接続の待ち行列に入れることが可能である。しかし、イニシエータとターゲットの間の回線中継をサポートするために、セッション当たり複数の接続が実現可能である。例えば、いくつかの実施形態では、イニシエータとターゲットは、互い

に通信が可能であり、複数接続を受け入れるための交渉を通じて決めることになる。別の実施形態では、イニシエータとターゲットは、１つのセッション又は接続だけを通じて通信する。図１３と図１４は、ＴＣＰ／ＩＰ及びｉＳＣＳＩセッションデータベース、又は、セッション及び接続毎の伝送制御ブロックを示す。これらのエントリは、別々のテーブルとして搬送することができ、又は、次に図２３、２４、２６、及び２９に関連して見られるように、複合テーブルとして一緒に搬送することもできるが、ＴＣＰ／ＩＰだけ、ＲＤＭＡを有するＴＣＰ／ＩＰ、ＩＰストレージだけ、ＴＣＰ／ＩＰを有するＩＰストレージ、及びＲＤＭＡを有するＩＰストレージなど、選択された実施と実施された機能性とに依存する。ＴＣＰ／ＩＰ及びストレージフロー制御を実施する様々なエンジンは、これらのフィールドの全部又はいくつか、又は、示されていないより多くのフィールドを使用して、ＴＣＰ／ＩＰに亘ってブロックデータ転送を方向付けする。データ転送過程の間に複数の状態を通じて接続が進行する時、適切なフィールドが更新される。図１５は、オンチップセッションキャッシュ（ブロック１５０１及び１５０２）、及びデータ転送の連続した進行のために必要な状態情報を保持するオフチップセッションメモリ（ブロック１５０３、１５０４、１５０５、１５０６、及び１５０７）から成るメモリサブシステムに転送制御エントリを保存する一方法を示す。

【００４１】

図１６は、抽象化レベルの高いＩＰプロセッサアーキテクチャを示す。プロセッサは、モジュールの拡張可能ＩＰネットワークアプリケーションプロセッサコア（ブロック１６０３）から成る。その機能性ブロックは、ＩＰネットワーク上で高速保存とデータ転送を利用可能にする機能性を提供する。プロセッサコアは、インテリジェントフローコントローラプログラマブル分類エンジン、及びストレージ／ネットワークポリシーエンジンを含むことができる。それぞれは、独立したプロセッサと考えることができ、それらの任意の組合せは、単一プロセッサとして実行することができる。開示されたプロセッサはまた、セキュリティ処理ブロックを含み、ネットワークパケットに対して高回線速度の暗号化及び暗号解読機能を提供する。同様に、これは、単一プロセッサにすることができ、又は上述の他のものと組み合わせることができる。開示されたプロセッサは、メモリサブシステムを含み、このサブシステムは、オンチップセッションキャッシュ／メモリを管理するメモリコントローラインタフェースと、オフチップメモリへのアクセスを管理するメモリコントローラ（ブロック１６０２）とを含み、このオフチップメモリは、ＳＲＡＭ、ＤＲＡＭ、ＦＬＡＳＨ、ＲＯＭ、ＥＥＰＲＯＭ、ＤＤＲ　ＳＤＲＡＭ、ＲＤＲＡＭ、ＦＣＲＡＭ、ＱＤＲ　ＳＲＡＭ、又は、静的又は動的ランダムアクセスメモリの他の派生物、又はその組合せとすることができる。ＩＰプロセッサは、適切なシステムインタフェースを含み、それを目標のマーケットセグメントで使用することができるようにし、ＬＡＮ、ＳＡＮ、ＷＡＮ、及びＭＡＮネットワーク、並びに同様のネットワーク、並びに適切なホストインタフェース（ブロック１６０６）に対して、適正なメディアインタフェース（ブロック１６０１）を提供する。メディアインタフェースブロックとホストインタフェースブロックとは、マルチポートの形態で存在することができ、そこでは、いくつかのポートは、開示されたプロセッサが使用されるネットワーク及びシステムで冗長性及びフェイルオーバー機能を供給することができる。プロセッサはまた、例えば、複数プロセッサシステムを作り出してメインプロセッサの機能を拡張するために、共用プロセッサインタフェース（ブロック１６０５）を含むことができる。ブロック１６０４のシステムコントローラインタフェースは、このプロセッサが汎用マイクロコントローラとインタフェースで接続することを可能にし、このマイクロプロコントローラは、開示されたプロセッサを使用することができるシステムに対して、システムコントローラとして作動することができる。プロセッサアーキテクチャはまた、システムコントローラ又はセッションマネージャとして作動することができる搭載された制御プレーンプロセッサをサポートする。システムコントローラインタフェースを依然として提供し、外部プロセッサの使用を可能にすることができる。このプロセッサのこのようなバージョンは、ダイのコストの理由から制御プロセッサを含むことができない。例えば、サーバアダプタ、又はストレージコントローラ、又は

スイッチ回線カード、又は他のネットワーキングシステムのような特定のシステム要件に的を絞って作り出すことができる様々な形式のコアアーキテクチャがある。主な違いは、本特許の前出の節で上述したようなことであろう。これらのプロセッサブロックは、ｉＳＣＳＩ、ＦＣＩＰ、ｉＦＣＩＰなどのような標準的プロトコルを使用し、高性能ＩＰベースのストレージを達成するための機能と性能をもたらす。これらのブロックの詳細なアーキテクチャを以下に説明する。

【００４２】

図１７は、ＩＰプロセッサアーキテクチャをより詳細に示す。上述のアーキテクチャは、完全なＴＣＰ／ＩＰの終了と深いパケットの検査とを通じてメディアアクセス制御装置（ＭＡＣ）又は他の適切な装置からの着信ＩＰパケットを処理するための機能を提供する。この図表は、図１６のＭＡＣ層のブロック１６０１、又はブロック１６０２、１６０４、又は１６０５を示していない。メディアインタフェース（ブロック１６０１）内のプロセッサの入力待ち行列（ブロック１７０１）と出力待ち行列（ブロック１７１２）とに対するＭＡＣ層インタフェースブロックは、図１６に示されている。ＭＡＣの機能性は、ネットワークに依存する特定の形式を有する標準ベースとすることができるであろう。イーサネット及びＳＯＮＥＴ上パケットは、今日最も広範囲に使用されるインタフェースの例であり、それらは、同じシリコン上、又は各々を使用して作り出されたプロセッサの異なるバージョンに含めることができる。

【００４３】

図１７のブロック図は、入力待ち行列及び出力待ち行列ブロック１７０１及び１７１２を２つの別のブロックとして示す。機能性は、組み合わせられたブロックを使用して提供することができる。入力待ち行列１７０１は、ＭＡＣインタフェースブロックからの着信パケットを検索するために、論理、制御、及びストレージから成る。ブロック１７０１は、パケットスケジューラ１７０２及び分類エンジン１７０３と協働してインタフェースから到着するパケットを待ち行列に入れ、パケットの最初、パケットの最後、及び、フラグメント化したパケット又は安全なパケットのような他の属性を識別するために適切な標識を作り出す。パケットスケジューラ１７０２は、入力待ち行列コントローラからパケットを検索し、分類のためにそれらを分類エンジンに転送することができる。分類ブロック１７０３は、スケジューラに続くように示されているが、論理的な観点から、分類エンジンが入力待ち行列からパケットを受け取り、パケットを分類して分類タグをパケットに提供し、そのパケットが次にスケジューラによってプロセッサアレイ１７０６（ａ）．．．１７０６（ｎ）に対してスケジュールに入れられる。従って、分類エンジンは、通過分類エンジンとして作用することができ、その構成を最高回線速度で使用してパケットフローを維持する。分類エンジンは、プログラマブルエンジンであり、様々なカテゴリのネットワークからパケットを受け取り、識別の結果を使用して、スケジューラと、使用すべき他のパケットプロセッサとのためにパケットにタグ付けする。ネットワークトラフィックの分類は、非常に計算集約的なタスクであり、パケットプロセッサで利用可能なプロセッササイクルの半分を占める。この一体化された分類エンジンは、第２層から第７層の検査を実行するためにプログラム可能である。分類されるフィールドは、適合するものがあれば、それらに対応する比較と処置のための期待値を使用してプログラムされる。図３０に関連して次に見られるように、分類子は、分類ウォークの結果を収集し、分類結果を識別するパケットに対するタグとしてこれらを呈示することができる。これは、ツリー構造によく似ており、「ウォーク」として理解されている。分類されたパケットは、次に、処理のパイプラインの次の段階としてスケジューラ１７０２に供給される。

【００４４】

パケットスケジューラブロック１７０２は、状態コントローラと、開示されたプロセッサ上の適切な実行エンジンにパケットを割り当てるシーケンサとを含む。実行エンジンは、所望する実施次第で、ＴＣＰ／ＩＰ、及び／又はストレージエンジン、並びにストレージフロー／ＲＤＭＡコントローラ（ブロック１７０８）、又はホストバイパス、及び／又は他の適切なプロセッサを含むＳＡＮパケットプロセッサ（ブロック１７０６（ａ）から

１７０６（ｎ））である。明確にするために、記号「／」は、本特許でハードウエア構成要素を指す時は、適宜「及び／又は」を意味することができる。例えば、構成要素「ストレージフロー／ＲＤＭＡコントローラ」は、実行に対して適宜、ストレージフロー及びＲＤＭＡコントローラ、ストレージフローコントローラ、又はＲＤＭＡコントローラとすることができる。スケジューラはまた、パケットから同じ接続／セッション上のパケットへの状態の従属性が着信パケットの正しい処理のために重要である場合、プロセッサを通じてパケット順序化を維持する。パケットのリタイヤメントまで、プロセッサを使用してスケジュールに入れたパケットの進行を追跡するために、スケジューラは、様々なテーブルを維持する。スケジューラはまた、出ていくコマンド上のパケットプロセッサと、ホストプロセッサ又は切替ファブリックコントローラ又はインタフェースからのパケットとに対してスケジュールに入れる必要があるコマンドを受け取る。

【００４５】

プログラマブルパケットプロセッサに加えて、ＴＣＰ／ＩＰ及びストレージエンジンは、図１７のＳＡＮパケットプロセッサ１７０６（ａ）から１７０６（ｎ）として一緒にラベル付けされる。これらのパケットプロセッサは、独立のプログラマブル構成要素であるエンジンであり、特定の役割を果たす。代替的に、２つ又はそれ以上のそれらを望ましい実行次第で単一プロセッサとして実行することができる。図２３のＴＣＰ／ＩＰエンジンと図２４のストレージエンジンは、この例で、図２１のプログラマブルパケットプロセッサエンジンブロック２１０１に対する共有プロセッサとして構成される。このアーキテクチャは、コスト、製造、マーケットセグメントのような理由からストレージエンジンを置換／除去することにより、ストレージ以外のアプリケーションに比較的容易に適用することができる。純粋なネットワーキング環境では、ストレージエンジンは、専用のＴＣＰ／ＩＰエンジンを有するパケットプロセッサを残して除去され、ネットワーキングトラフィックの代わりに適用されることがあり、ＴＣＰ／ＩＰソフトウエアスタックによる同様の処理のオーバーヘッドに直面する。代替として、１つ又はそれ以上のエンジンを望ましい実施のために落としてもよく、例えば、ＩＰストレージ機能だけをサポートするプロセッサに対して、別のチップ内にあるＴＣＰ／ＩＰエンジン及び／又はパケットエンジンを落とすことができる。従って、コアの拡張可能モジュラーアーキテクチャの複数の変形が実現可能である。従って、コアのアーキテクチャは、例えば、アプリケーション専用の加速度のためにサポートを提供する高性能ネットワークセキュリティ及びポリシーエンジン、高性能ルーティングエンジン、高性能ネットワーク管理エンジン、ストリング検索を提供する深いパケット検査エンジン、ＸＭＬのエンジン、仮想化のためのエンジンのような他の専用エンジンでストレージエンジンを置換することにより、ＩＰアプリケーション上のストレージに加えて、アプリケーション内で強化することができる。このＩＰプロセッサの処理機能は、ネットワークインタフェースの回線速度の要求に合致するように、チップ内のＳＡＮパケットプロセッサブロック１７０６（ａ）から１７０６（ｎ）の数を増減することにより、スケーリングすることができる。拡張容易性からの根本的な制約は、要求されるシリコンのスペースとシリコン処理技術によって課せられる制約から生じる。根本的にこのアーキテクチャは、より多くのＳＡＮパケットプロセッサブロックを追加して処理機能を向上させることにより、かなりの高回線速度まで拡張することができる。同様の結果を達成する他の手段は、プロセッサの作動のクロック周波数を処理技術の限界内で実現可能な程度まで上げることである。

【００４６】

図１７はまた、ＩＰセッションキャッシュ／メモリ及びメモリコントローラブロック１７０４を示す。このキャッシュは、内部メモリ又はローカルセッションのデータベースキャッシュと見ることができる。このブロックは、ＴＣＰ／ＩＰセッションデータベースと、同様に特定数のアクティブセッションに対するストレージセッションデータベースとをキャッシュして保存するために使用される。キャッシュされるセッションの数は、選択されたシリコンスペースと、製造するための経済的な可能性との直接の結果である。オンチップでないセッションは、ブロック１７０４の一部か又は別様の高性能メモリコントロー

ラブロックを使用して、外部メモリとして見られるオフチップメモリへ／から保存及び検索される。このプロセッサの様々な処理要素は、高速内部バスを使用してこのコントローラを共有し、セッション情報を保存して検索する。メモリコントローラはまた、フラグメント化したパケットを一時的に保存するために、又はホストインタフェース又は外部行きの待ち行列がバックアップされる時に使用することができる。コントローラはまた、統計データ情報、又は、開示プロセッサか又は開示又はホストプロセッサ上で作動しているアプリケーションによって収集することができる他の任意の情報を保存するために使用することができる。

【００４７】

図１７のプロセッサのブロック図はまた、ホスト入力待ち行列（ブロック１７０７）とホスト出力待ち行列（ブロック１７０９）、並びにストレージフロー／ＲＤＭＡコントローラ（ブロック１７０８）を示す。これらのブロックは、ホスト（「ピア」とも呼ばれる）メモリ又は切替ファブリックへ及びそこからデータを転送するために必要な機能を提供する。これらのブロックはまた、ホストベースのドライバがコマンドをスケジュールに入れ、着信ステータスを検索し、セッションデータベースのエントリを検索し、開示されたプロセッサをプログラムすることができるようにする機能を提供し、ソケット直接アーキテクチャ、ＴＣＰ／ＩＰの完全終了、ＩＰストレージオフロードのような機能をＲＤＭＡを使用して、又は使用しないで実行することができるようにする。図２７により詳細に見られるホストインタフェースコントローラ１７１０は、構成レジスタ、メモリ対メモリの直接データ転送のためのＤＭＡエンジン、及び上述のいくつかのタスクを実行するホストコマンドブロックをホストインタフェース処理コントローラ及びホスト割り込みコントローラと共に提供する。ホスト入力及び出力待ち行列１７０７及び１７０９は、着信パケットと発信パケットに待ち行列を提供する。ストレージフローとＲＤＭＡコントローラブロック１７０８は、開示されたプロセッサに対してホストがコマンドを待ち行列に入れるために必要な機能性を提供し、このプロセッサは、次にこれらのコマンドを取得してそれらを実行し、コマンドが終了するとホストプロセッサに割り込む。ブロック１７０８のＲＤＭＡコントローラ部分は、遠隔の直接メモリアクセスを可能にするために必要な様々な機能を提供する。それは、ＲＤＭＡ領域、アクセスキー、及び仮想アドレス変換機能性のような情報を含むテーブルを有する。このブロック内部のＲＤＭＡエンジンは、データ転送を実行し、受け取ったＲＤＭＡコマンドを変換し、実行が許可されている場合に処理を実行する。ブロック１７０８のストレージフローコントローラはまた、ターゲットとイニシエータの間でデータ転送が発生した時に、スケジュールに入れられた様々なコマンドの進行状態の追跡を保存する。ストレージフローコントローラは、実行するためのコマンドをスケジュールに入れ、コマンド終了情報をホストドライバに提供する。以上はＲＤＭＡ機能と考えられ、説明したように又は個々のプロセッサを実施することにより、設計者の選択次第で実施することができる。また、本特許の精神及び範囲から逸脱することなく、これらの説明に対して付加的な機能を追加するか又は排除することができる。

【００４８】

このプロセッサの制御プレーンプロセッサブロック１７１１は、ＩＣＭＰプロトコルによるエラー処理、ネーム解決、及びアドレス解決プロトコルを含むことができるＴＣＰ／ＩＰ及び／又はストレージプロトコルに対する比較的低速のルーティング機能を提供するのに使用され、それはまた、プログラムされてセッションコントローラ／接続マネージャとしてのセッション開始／取外し作動、ログイン及びパラメータ交換などを実行することができる。この制御プレーンプロセッサは、システム開発者に制御プレーンプロセッサの選択を提供するためにオフチップにするか、又は統合されたソリューションを提供するためにオンチップにすることができる。制御プレーンプロセッサがオフチップの場合、インタフェースブロックがそこで作り出されるか又は統合され、このプロセッサが制御プレーンプロセッサとインタフェースで接続することができるようになり、データとコマンドの転送を実行する。内部バス構造及び機能ブロック相互接続は、性能やダイのコスト要件などに関する全ての詳細図に対して図示と異なることがあり、それらは、本発明の精神及び

範囲から逸脱しないものである。
図１７のブロックに関して上述した機能は、以下により詳細に説明するように、開示されたプロセッサの様々な処理リソースによるインストリーム処理を使用して、パケットが最低限の待ち時間で入力から出力まで通過することができるパケットストリームアーキテクチャを実行可能にする。
【００４９】
図１８は、一般的に、詳細には図１７の１７０１で表される入力待ち行列とコントローラブロックを示す。このブロックの核になる機能性は、ブロック１８０１及び１８０２（ｉ）から１８０２（ｎ）において複数の入力ポートであるポート１からＮからの着信パケットを受け入れ、パケットが分類子、スケジューラ、及びスケジューラＩ／Ｆブロック１８０７～１８１４を通した更に別の処理に対してデキューされた入力パケット待ち行列（ブロック１８１０）に対する固定又はプログラマブル優先度を使用して、それらを待ち行列に入れることである。入力待ち行列コントローラは、各入力ポート（マルチポート実行のポート１からポートＮ）とインタフェースで接続し、入力パケット待ち行列上１８１０に対してパケットを待ち行列に入れる。パケット待ちコントローラとマーカーブロック１８０４は、ポート速度、ポートのネットワークインタフェース、ポート優先度、及び他の適切な様々な特徴に基づいて、固定された優先度機能を提供することができ、又は、別々のインタフェースに対して別々のポリシーが適用されるようにプログラムすることができる。ラウンドロビンや重み付けラウンドロビンなどのような様々なモードの優先度をプログラムすることができる。入力パケットデキューコントローラ（１８１２）は、パケットをデキューし、それらをスケジューラＩ／Ｆ１８１４を通じてパケットスケジューラ（図１７のブロック１７０２）に供給する。スケジューラは、図１７の分類エンジン１７０３によってパケットが分類された状態で、ＳＡＮパケットプロセッサ１７０６（ａ）～１７０６（ｎ）に対してパケットをスケジュールに入れる。暗号化されたパケットは、最初に暗号化される時に分類することができ、実施がセキュリティ処理を含む場合、認証及び／又は暗号解読のために、図１８の安全なパケットインタフェースブロック１８１３により、図１７のセキュリティエンジン１７０５に転送されるが、そうでない場合は、セキュリティインタフェースは存在しないことがあり、同様の機能を実行するために外部セキュリティプロセッサが使用されるであろう。クリアなパケットインタフェース（ブロック１８１１）から暗号解読されたパケットは、次に、パケットがクリアなパケットと同じパスを追従するブロック１８１２を通じて入力待ち行列に供給される。
【００５０】
フラグメント化したＩＰパケットは、フラグメント化パケット保存制御バッファ（ブロック１８０６）内のオンチップに保存することができるか、又は内部又は外部メモリに保存することができる。最後のフラグメントが到着すると、ブロック１８０６のフラグメント化コントローラは、図１７の分類エンジン及びスケジューラと協働して、これらのフラグメントを組み合わせて完全なパケットを組み立てる。フラグメント化したパケットが組み合わせられて完全なパケットを形成した状態で、パケットは、ブロック１８０４を通じて入力待ち行列に対してスケジュールに入れられ、次に、パケットデキューコントローラ（ブロック１８１２）によって処理され、このプロセッサの様々な他の処理ステージに通される。図１８の入力待ち行列コントローラは、ブロック１８０８のパケット属性及びタグアレイに保存された、分類エンジンからの他の任意のセキュリティ情報と共にパケット開始、サイズ、バッファアドレスのようなパケット記述子フィールドを使用するブロック１８０９の属性マネージャによって管理されるパケットタグ／記述子を各着信パケットに割り当てる。パケットタグ及び属性は、プロセッサのスケジューラと他の構成要素により、インタフェース８０７、１８１１、１８１３，及び１８１４を通じて、プロセッサを通過するパケットのフローを効率的に制御するために使用される。
【００５１】
図１９は、図１７のパケットスケジューラとシーケンサ１７０２をより詳細に示す。このブロックは、パケットのスケジュール設定とこのプロセッサの実行リソースに対するタ

スクとを担い、従って、負荷バランサとしても作用する。スケジューラは、入力待ち行列コントローラ１９０１から、ヘッダの待ち行列（ブロック１９０２）のパケットのヘッダを検索し、それらを図１７の分類エンジン１７０３に通し、このエンジンは、次に、残りのプロセッサによって使用される分類子待ち行列（ブロック１９０９）の分類結果を戻す。分類エンジンは、最初はヘッダと共に呈示することができるが、深いパケット検査もプログラムされる場合は、分類エンジンは、分類の後でスケジューラにルーティングする完全なパケットを受け取ることができる。スケジューラは、分類エンジンを通じてパケットの実行を管理する分類コントローラ／スケジューラ（ブロック１９０８）を有する。図１９のこのブロックは、フラグメント化したパケット又は安全なパケットの場合は、そのようなパケットに対して適切な作動を行うために、例えば、図１７のセキュリティエンジンに対して暗号化されたパケットをスケジュールに入れるために、入力待ち行列コントローラ（ブロック１９０１）にコマンドを供給する。スケジューラ状態制御及びシーケンサ（ブロック１９１６）は、プロセッサ内部でアクティブな様々な処理／作動の状態情報を受け取り、次の作動のセットに対する命令を提供する。例えば、スケジューラは、ブロック１９０３の入力パケット待ち行列からパケットを検索し、分類子から受け取った分類の結果に基づいて、適切なリソース待ち行列に対してこれらのパケットをスケジュールに入れるか、又はパケットをパケットメモリ（ブロック１９１３又は１７０４から１９０６）に導き、スケジュール設定時又はその後に作動を実行するために、適切なリソースがそれを必要とする時にパケットを検索するために使用することができるパケット記述子／タグを作り出す。状態制御及びシーケンサブロック１９１６は、分類結果（ブロック１９１４）を有するパケットが、パケットが作動のためにスケジュールに入れられる時に検索されるパケットメモリ（ブロック１９１３）に保存されるように命令／誘導する。状態コントローラ及びシーケンサは、作動のためのパケットを受け取るべき実行リソースを識別し、コマンドを作り出し、このコマンドをパケットタグと共に図１９のリソース待ち行列ブロック１９１７（制御プレーン）、１９１８（ポートｉ～ポートｎ）、１９１９（バイパス）、及び１９２０（ホスト）に割り当てる。優先度セレクタ１９２１は、割り当てられた優先度に基づいて、それぞれの待ち行列からコマンドとパケットタグを検索し、これをパケットフェッチとコマンドコントローラ（ブロック１９２２）に転送するプログラマブルブロックである。このブロックは、分類結果と共にパケットメモリストア１９１３からパケットを検索し、例えば、１９２６でリソースが作動待機中などの時に、高性能プロセッサのコマンド上の適切なリソースとパケットバスに対してパケット転送をスケジュールに入れる。それぞれの受け取り側のコマンドバスインタフェースコントローラ１９０５のようなバスインタフェースブロックは、コマンドを解釈してパケットと作動のための分類タグを受け入れる。これらの実行エンジンは、何時パケット作動が終了し、何時パケットが最終の宛先（ホストバスインタフェース、又は出力インタフェース、又は制御プレーンインタフェースなどのいずれか）に対してスケジュールに入れられるかをスケジューラに通知する。これによって、スケジューラは、ブロック１９０４のリタイヤメントエンジンの助けを借りて、パケットをその状態から退去させることができ、リソース割当テーブル（ブロック１９２３）のこのセッションのためにリソースエントリを解放する。リソース割当テーブルは、例えば、ＳＡＮパケットプロセッサエンジンのバッファに入れられたセッションデータベースキャッシュエントリや発信元内で実行されている現在のパケットの接続ＩＤなどのこれらのリソースの内部状態の現在の状態により、受け取ったパケットを特定のリソースに割り当てるためにシーケンサによって使用される。従って、順序付けられた実施に依存するパケットは、最初に同じリソースに割り当てられ、プロセッサのセッションメモリの現在のＤＢ状態を使用してメモリのトラフィックと性能を改善し、従って、新しいセッションのエントリを検索しなくてもよい。シーケンサはまた、パケットを待ち行列に入れるためにメモリコントローラ（ブロック１９０６）とのインタフェースを有し、これらのパケットは、フラグメント化されたパケットであり、及び／又は、最大回線速度性能を維持するために割り当てられたものよりも多くの時間が掛かり、パケット上で実行された特定のアプリケーションが原因となることがある下流のパケット処理の妨害のため

にスケジューラの待ち行列が渋滞する場合のものであり、又は、他の任意の下流のシステムが一杯になって回線速度が維持できない場合のものである。

【００５２】

分類エンジンが入力待ち行列からパケットを受け取る図１７に関連して上述したように、分類子がスケジューラの前で実施される場合、アイテム１９０１、１９０２、１９０８、１９０９、及び１９１０は、個々の設計次第で分類子内にあるか又は必要ないこともある。分類子からスケジューラブロック１９０３、１９０７、１９１４、及び１９１５まで／からの適切な連結は、そのようなシナリオで作り出すことができ、分類子は、図１８の入力待ち行列ブロックに直接連結される。

【００５３】

図２０は、通常は図１７の１７０３で表されるパケット分類エンジンを示す。パケットのそれらの様々な属性への分類は、非常に計算集約的な作動である。分類子は、パケット形式、例えば、ＩＰ、ＩＣＭＰ、ＴＣＰ、ＵＤＰのようなプロトコル形式、ポートアドレス、発信元及び宛先フィールドなどを識別するために、受け取ったパケットの様々なフィールドを試験するプログラマブルプロセッサとすることができる。分類子は、ヘッダ又はペイロード内の特定フィールド又は一組のフィールドを試験するために使用することができる。ブロック図は、内容アドレスメモリベースの分類子を示す。しかし、先に説明したように、これもまた、プログラマブルプロセッサとすることができる。主な違いは、性能とエンジンの実施の複雑さである。分類子は、図２０の入力待ち行列（ブロック２００５及び２００４）からスケジューラを通じて入力パケットを取得する。入力バッファ２００４は、分類される必要があるパケット／記述子及び／又はパケットヘッダを待ち行列に入れる。次に、分類シーケンサ２００３は、待ち行列内の利用可能なパケットをフェッチし、プログラムされるか又はプログラムすることができるグローバルフィールド記述子の組（ブロック２００７）に基づいて、適切なパケットフィールドを抽出する。次に、分類子は、内容アドレスメモリ（ＣＡＭ）アレイ（ブロック２００９）にこれらのフィールドを通し、分類を実行する。フィールドは、ＣＡＭアレイを通過するので、これらのフィールドの符合は、比較されるべきフィールドの次の組と、場合によってはそれらのビットフィールド位置とを識別する。ＣＡＭアレイでの符合は、アクション／イベントタグをもたらし、このタグは、結果コンパイラ（ブロック２０１４）から収集され（「コンパイル」が「収集」の意味で使用される場合）、また、特定のＣＡＭ条件又は規則の符合に関連するメモリアレイ（ブロック２０１３）のデータを更新することができるアクションとしても作用する。これは、演算論理装置（ＡＬＵ）作動（ブロック２０１７）の実行を含むことができ、これは、例えば条件符合の増分又は減分などのこのフィールド上の実行リソースの一例と考えられる。ＣＡＭアレイは、ホスト又は制御プレーンプロセッサインタフェース１７１０及び１７１１を通してプログラミングのためにアクセス可能なデータベース初期化ブロック２０１１を通じて、比較すべき次のフィールドを含むフィールド、それらの期待値、及び符合に対するアクションを使用してプログラムされる。分類がリーフノードに達した状態で、分類は終了し、同じ分類タスクを実行するのを回避するためにＩＰプロセッサの他のエンジンによって次に使用することができる、トラバースされたパスを識別する分類タグが生成される。例えば、分類タグは、フロー又はセッションＩＤ、ＴＣＰ／ＵＤＰ／ＩＣＭＰのようなプロトコル形式の表示、処理、バイパス、パケットのドロップ、セッションのドロップなどをするか否かを示す値を含むことができ、又は実行リソースがパケット処理を開始するための特定のファームウエアコードのルーチンポインタを含むことができ、又はトラバースされた分類パスの署名などを含むこともできる。分類タグフィールドは、プロセッサの実行と機能性に基づいて選択される。分類子リタイヤメント待ち行列（ブロック２０１５）は、分類されたパケットのパケット／記述子と分類タグを保持し、スケジューラによって検索されるのを待機する。分類データベースは、データベース拡張インタフェースとパイプライン制御論理ブロック２００６とを使用して拡張することができる。これによって、より大きな分類データベースのための拡張性を必要とするシステムの構築が可能になる。アクションインタープリタ、ＡＬＵ、及び範囲符合ブロック

２０１２を有する分類エンジンはまた、ストレージ／ネットワークポリシー／特定のポリシーが満足された場合に取る必要があるアクションをプログラムする機能を提供する。ポリシーは、規則及びアクションテーブルの形態で実行することができる。ポリシーは、分類テーブルと共にホストインタフェースを使用して分類エンジン内でコンパイルされ、プログラムされる。データベースインタフェースとパイプライン制御２００６は、分類／ポリシーエンジンを拡張するためにコンパニオンプロセッサと連結するように実施することができる。

【００５４】

図２１は、通常は図７の１７０６（ａ）から１７０６（ｎ）で示すようなＳＡＮパケットプロセッサを示す。パケットプロセッサは、特別に設計されたパケットプロセッサか、又はＡＲＭ、ＭＩＰＳ、ＳｔｒｏｎｇＡＲＭ、Ｘ８６、ＰｏｗｅｒＰＣ、Ｐｅｎｔｉｕｍプロセッサのような任意の適切なプロセッサか、又は本明細書に説明した機能を満足させる他の任意のプロセッサとすることができる。これは、本特許の様々な節においてパケットプロセッサ複合システムとも呼ぶ。このパケットプロセッサは、パケット処理のためのターゲット命令、又はＴＣＰ／ＩＰエンジン（ブロック２１０２）、又はＩＰストレージエンジン（ブロック２１０３）、又はその組合せを備えた、一般的にＲＩＳＣマシンであるパケットエンジン（ブロック２１０１）を有する。これらのエンジンは、パケットエンジンに対する共有エンジンとして、又は独立したエンジンとして構成することができる。図２２は、パケットエンジンをより詳細に示す。パケットエンジンは、通常は上述のようにパケット処理マイクロルーチン及びパケット及び中間ストレージを保持するために使用される命令メモリ（ブロック２２０２）及びデータメモリ（ブロック２２０６）（両方ともＲＡＭである）を備えたＲＩＳＣマシンである。命令メモリ２２０２は、本特許の全てのこのようなメモリと同様にＲＡＭ又は他の適切なストレージであり、パケット処理中に実行されるコードを使用して初期化される。パケット処理コードは、割り当てられたメモリ内に適合する緊密なマイクロルーチンとして編成される。命令復号器及びシーケンサ（ブロック２２０４）は、命令メモリ２２０２からの命令をフェッチしてそれらを復号化し、ＡＬＵ（ブロック２２０８）内に包含された実行ブロックを通じてそれらを順序付けする。このマシンは、単純なパイプラインエンジンか、又はパケット志向命令セットを提供するように設計されたより複雑な深いパイプラインマシンとすることができる。ＤＭＡエンジン（ブロック２２０５）及びバスコントローラ（ブロック２２０１）により、パケットエンジンがデータパケットを図１９のスケジューラから移動し、ホストインタフェースを作動のためにデータメモリ２２０６に移動させることが可能になる。ＤＭＡエンジンは、パケット／データをメモリ／パケットメモリへ／から保存／検索するために複数のメモリ記述子を保持することができる。これによって、メモリアクセスが、プロセッサエンジン作動と平行して発生するようになる。ＤＭＡエンジン２２０５はまた、データパケットをＴＣＰとストレージエンジン２２１０及び２２１１の間で移動させるために使用することができる。パケットの実行が終了した状態で、抽出されたデータ又は新しく生成されたパケットは、メディアインタフェース又はホストインタフェースのいずれかに向けて出力インタフェースに転送される。

【００５５】

図２３は、通常は図２２の２２１０に見られるプログラマブルＴＣＰ／ＩＰパケットプロセッサエンジンをより詳細に示す。このエンジンは、一般的に、様々なＴＣＰ／ＩＰ志向命令及び実行エンジンと共に共有ＲＩＳＣ命令を有するプログラマブルプロセッサであるが、本特許に説明する適切な実行エンジンを備えたマイクロコード化又は状態機械駆動プロセッサとすることもできるであろう。ＴＣＰプロセッサは、ＴＣＰチェックサムの確認、及びプロセッサ上のこれらの命令の実行による新しいチェックサム生成のためのチェックサムブロック２３１１を含む。チェックサムブロックは、パケットバッファメモリ２３０９（データＲＡＭがそのようなメモリの一例である）からデータを抽出し、チェックサムの生成又は確認を行う。パケット検索インタフェースブロック２３１０は、実行エンジンと命令シーケンサ２３０５を助け、様々なデータパケットフィールド又は全データパ

ケットにアクセスを提供する。分類タグインタープリタ２３１３は、そのような実行が選択された場合、分類の結果に基づいてプログラムフローを誘導するために命令復号器２３０４によって使用される。プロセッサは、例えば、次の予想されるシーケンス番号を検索し、受け取ったものが合意されたスライドウインドウの範囲内にあるか否か、また、パケットが属する接続に対してどのスライドウインドウがＴＣＰプロトコルの公知の部分であるかを見るために、特定のシーケンスとＴＣＰ／ＩＰデータ順序付け計算で使用するための分割（ブロック２３１５）を含むウィンドウ作動を提供する。この要素２３１５は、図２４の２４１３で表すような分割コントローラを含むことができる。代替的に、当業者は、本特許の教示を使用して、この図２３のＴＣＰ／ＩＰプロセッサ上のどこにでも分割コントローラを容易に実施することができる。このプロセッサは、ハッシュエンジン（ブロック２３１７）を提供し、このエンジンは、パケットの特定フィールドに対してハッシュ作動を実行するために使用され、パケットに対して正しいセッションエントリを取得するために必要なハッシュテーブルウォークを実行する。プロセッサはまた、データの発信元及び宛先のためのポインタレジスタ、コンテキストレジスタセット、及び汎用レジスタファイルに加えてＴＣＰ状態を保持するレジスタと共に、ＴＣＰ処理のための様々な共通に使用されるヘッダフィールドを抽出するレジスタファイル（ブロック２３１６）を含む。ＴＣＰ／ＩＰプロセッサは、パケットの実行に対して複数のコンテキストを有することができ、そのために、例えばメモリアクセスのような何らかの理由で所定パケットの実行が機能停止すると、別のコンテキストが喚起され、プロセッサは、ごく僅かの効率ロスで別のパケットストリームの実行を継続することができる。ＴＣＰ／ＩＰプロセッサエンジンはまた、ローカルセッションキャッシュ（ブロック２３２０）を維持し、このキャッシュは、最も最近使用された又は最も頻繁に使用されたエントリを保持し、グローバルセッションメモリから検索する必要なしに局所的に使用することができる。ローカルセッションキャッシュは、パケットプロセッサとすることができるＴＣＰ／ＩＰプロセッサの内部メモリと考えることができる。セッション又はグローバルメモリから付加的なものを検索することなく、より多くのエントリが使用されて内部メモリに保存することができるほど、処理がより効率的になるのは勿論である。図１９のパケットスケジューラは、ＴＣＰ／ＩＰリソース毎にキャッシュされた接続ＩＤを通知され、従って、同じパケットプロセッサ複合システムに対して同じセッションに属するパケットをスケジュールに入れることができる。特定の接続に対してパケットプロセッサがセッションエントリを保持しない場合、ＴＣＰセッションデータベース検索エンジン（ブロック２３１９）は、セッションマネージャ（ブロック２３２１）及びハッシュエンジンと協働し、グローバルセッションメモリからメモリコントローラインタフェース（ブロック２３２３）を通じて対応するエントリを検索する。ハッシュエンジンと共に作用して、対応するセッションエントリ又はそのフィールドをセッションデータベースに対して保存／検索するためのセッション識別子を生成するセッションエントリ又はセッションエントリのフィールドのアクセスを可能にするセッションマネージャ内の論理回路のような手段がある。これは、それらのフィールド又はエントリをパケット処理の結果として更新するために使用することができる。新しいエントリがフェッチされると、配置されているエントリは、グローバルセッションメモリに保存される。ローカルセッションキャッシュは、キャッシュ原理に排他的に従うことができ、従って、複数プロセッサの複合システムは、セッションの状態に損害を与えるいかなる競合条件ももたらすことはない。ＭＥＳＩプロトコルのような他のキャッシュプロトコルは、類似の結果を達成するのに使用することができる。プロセッサ複合システムでセッションエントリがキャッシュされ、別のプロセッサ複合システムがそのエントリを必要とする場合、アルゴリズムに基づいて独占的アクセス又は適切なキャッシュ状態を使用し、このエントリは、新しいプロセッサに転送される。セッションエントリはまた、グローバルセッションメモリに書き込まれる場合がある。ＴＣＰ／ＩＰプロセッサはまた、作動されている接続に対してＴＣＰ状態をウォークスルーするために、ＴＣＰ状態機械（ブロック２３２２）を含む。この状態機械は、新しく受け取ったパケットから、状態に影響する適切なフィールドと共にセッションエントリに保存された同じ情報を受け取る。これによ

って、状態移行があり、情報がセッションテーブルエントリで更新されている場合は、状態機械は、次の状態を生成することができるようになる。ＴＣＰ／ＩＰプロセッサはまた、フレーム情報を抽出して順序不揃いのパケット実行の作動を行うために使用されるフレームコントローラ／順序不揃いマネージャブロック２３１８を含む。このブロックはまた、図２４の２４１７で表されるＲＤＭＡ機構を含むことができるが、非ストレージデータ転送のために使用される。当業者はまた、本特許の教示を使用して、ＴＣＰ／ＩＰプロセッサのどこにでもＲＤＭＡ機構を実施することができる。このアーキテクチャは、上部層フレーム指示機構を作り出し、この機構は、プログラマブルフレームコントローラによって使用されるフレーム指示キー又は他のキーとしてパケットＣＲＣを使用することができ、パケットの順序が狂って到着した時でも埋め込まれたＰＤＵを抽出し、それらが最後のバッファの宛先に導かれるようにする。このユニットは、セッションデータベースと対話し、順序が狂って到着した情報を処理し、この情報は記録され、従って、中間セグメントが到着した状態で再送信が回避される。パケットがＴＣＰ／ＩＰプロセッサを通じて処理された状態で、パケットがストレージデータ転送に属し、特定の実施がストレージエンジンを含む場合は、それは、作動のためにストレージエンジンに配信され、そうでなければ、最終のバッファの宛先に対するＤＭＡの処理のために、パケットは、ホストプロセッサのインタフェースか、又はブロック１７０８のストレージフロー／ＲＤＭＡコントローラに通される。パケットはまた、パケット上の任意の付加的な処理のために、パケットプロセッサブロックに転送することができる。これは、ＴＣＰ／ＩＰプロセッサ及びストレージプロセッサにより、処理の前か後にパケット上で実行することができるアプリケーション及びカスタムアプリケーションコードを含むことができる。ホストから出力メディアインタフェースへのデータ転送はまた、ＴＣＰ／ＩＰプロセッサを通過し、データ周辺に作り出されるべき適切なヘッダを形成し、フレームコントローラ及び／又はストレージプロセッサと協働し、適切なデータ分割を実行し、同じくセッション状態を更新する。このデータは、作動のためにスケジューラによってパケットプロセッサに対してスケジュールに入れられたホストコマンド又は受信ネットワークパケットの結果として検索することができる。内部バス構造及び機能性ブロック相互接続は、性能やダイコスト要件などに関して説明したものと異なる場合がある。例えば、ホストコントローラインタフェース２３０１、スケジューラインタフェース２３０７、及びメモリコントローラインタフェース２３２３は、バスコントローラの一部とすることができ、このバスコントローラは、スケジューラ、又はストレージフロー／ＲＤＭＡコントローラ、又はホスト、又はセッションコントローラ、又は、以下に限定されるものではないが、キュリティプロセッサ、メディアインタフェースユニット、ホストインタフェース、スケジューラ、分類プロセッサ、パケットバッファ又はコントローラプロセッサ、又は以上の任意の組合せのような他のリソースへの又はからのデータパケット又は状態情報又はコマンド又はその組合せの転送を可能にするものである。

【００５６】

図２４は、図２２のＩＰストレージプロセッサをより詳細に示す。ストレージエンジンは、一般的に、通常のＲＩＳＣ様のパケット処理用命令セットと共にＩＰベースのストレージを意図した命令セットを有するプログラマブルエンジンである。ＩＰストレージプロセッサエンジンは、ＣＲＣ作動を実行するためのブロック２４１１を含む。このブロックは、ＣＲＣの生成と確認を可能にする。ＩＰストレージを使用して着信パケットは、ＴＣＰ／ＩＰエンジンからＤＭＡ（ブロック２４０２及び２４０８）を通って、データメモリ（そのようなメモリの例がデータＲＡＭである）（ブロック２４０９）に転送される。実施がＴＣＰ／ＩＰエンジン又はパケットプロセッサエンジン又はその組合せを含まない場合は、パケットを例えばスケジューラから直接受け取ることができる。接続に関連するＴＣＰセッションデータベース情報は、必要に応じてローカルセッションキャッシュから検索することができるか、又はＴＣＰ／ＩＰエンジンからパケットを使用して受け取ることができる。ストレージＰＤＵは、ＰＤＵ分類子エンジン（ブロック２４１８）に提供され、このエンジンは、ＰＤＵを適切なコマンドに分類し、このコマンドは、次に、適切なス

トレージコマンド実行エンジン（ブロック２４１２）を呼び出すために使用される。コマンド実行は、ＲＩＳＣ又はピア又は命令セットを使用し、又は専用ハードウエアエンジンを使用して達成することができる。コマンド実行エンジンは、ＰＤＵで受け取ったコマンドを実行する。受け取ったＰＤＵは、保留書込コマンド又はＩＰストレージプロトコルによって要求された他のコマンドに対して、読取コマンドデータ又はＲ２Ｔを含むことができる。これらのエンジンは、書込データをホストインタフェースから検索するか、又は読取データを宛先バッファに誘導する。ストレージセッションデータベースは、プロセッサによって供給された最近の又は頻繁な接続に対してローカルメモリ（ブロック２４２０）として見なされるものにローカルにキャッシュされる。コマンド実行エンジンは、コマンドを実行し、ストレージ状態機械（ブロック２４２２）及びセッションマネージャ（ブロック２４２１）と協働してストレージデータベースエントリの更新を行う。接続ＩＤは、セッションを識別するために使用され、セッションがキャッシュに存在しない場合は、ストレージセッション検索エンジン（ブロック２４１９）により、図１７のグローバルセッションメモリ１７０４から検索される。イニシエータからターゲットへのデータ転送のために、プロセッサは、分割コントローラ（ブロック２４１３）を使用し、パスＭＴＵなどのような様々なネットワーク制約毎にデータユニットをセグメントに分割する。分割コントローラは、発信ＰＤＵが接続に対して最適なサイズになることを保証するように試みる。要求されたデータ転送が分割の最大有効サイズよりも大きい場合は、分割コントローラは、データを複数のパケットに圧縮し、シーケンスマネージャ（ブロック２４１５）と協働してシーケンス番号を適切に割り当てる。分割コントローラ２４１３はまた、図２３のＴＣＰ／ＩＰプロセッサ内で実行することができる。換言すれば、分割コントローラは、このプロセッサがＴＣＰ／ＩＰ作動のために使用されてストレージ作動でない時には、図２３のシーケンス／ウインドウ作動マネージャ２３１５の一部である。当業者は、本特許の教示を使用して、分割コントローラをＴＣＰ／ＩＰプロセッサに組み込むための代替実施形態を容易に示唆することができる。図２４のストレージプロセッサ（又は、図２３のＴＣＰ／ＩＰプロセッサ）はまた、このＲＤＭＡ機構を使用して実施されるストレージ又はネットワークデータ転送のためにＰＤＵ内で受け取った遠隔の直接アクセス命令を変換するＲＤＭＡエンジンを含むこともできる。例えば、図２４では、それはＲＤＭＡエンジン２４１７である。図２３のＴＣＰ／ＩＰプロセッサでは、ＲＤＭＡエンジンは、フレームコントローラ及び順序不揃いマネージャ２３１８、又は他の適切な要素の一部とすることができる。接続の両端がデータ転送のＲＤＭＡモードに合致する場合は、実質的なホストの介在なしにターゲット及びイニシエータ間のデータ転送をスケジュールに入れるためにＲＤＭＡエンジンが使用される。ＲＤＭＡ転送状態は、セッションデータベースエントリに維持される。このブロックは、データの周囲で階層化されたＲＤＭＡのヘッダを作り出し、ＲＤＭＡ使用可能の接続上で受け取った受信パケットからこれらのヘッダを抽出するためにも使用される。ＲＤＭＡエンジンは、メッセージ／命令を通過させることにより、ストレージフロー／ＲＤＭＡコントローラ（１７０８）及びホストインタフェースコントローラ（１７１０）と協働し、実質的なホストの介入なしに大量のブロックデータの転送を達成する。ＩＰプロセッサのストレージフロー／ＲＤＭＡコントローラブロック（１７０８）のＲＤＭＡエンジンは、要求された作動に対して保護検査を実行し、また、ＲＤＭＡ領域識別子からホスト空間の物理的又は仮想アドレスへの変換も提供する。この機能性はまた、選択された実行に基づくＳＡＮパケットプロセッサのストレージエンジンのＲＤＭＡエンジン（ブロック２４１７）によって提供することができる。２４１７と１７０８、及び他の同様のエンジンの間のＲＤＭＡ機能の分布は、本特許を使用して当業者が行うことができる実施の選択肢である。発信データは、ＰＤＵクリエータ（ブロック２４２５）によって標準ベースのＰＤＵにパッケージ化される。ＰＤＵのフォーマット設定はまた、パケット処理命令を使用することによって達成することができる。図２４のストレージエンジンは、図２３のＴＣＰ／ＩＰエンジン及び図１７のパケットプロセッサエンジンと協働し、両方の方向、すなわちイニシエータからターゲット、ターゲットからホスト、及びその逆にデータ及びコマンド転送を伴うＩＰストレージ作動を実行する。換言すれば

、ホストコントローラインタフェース２４０１及び２４０７は、コマンド又はデータ又はその組合せをホストプロセッサへ保存し、又はそれから検索する。これらのインタフェースは、直接又は中間の接続を通じてホストに接続される。２つの装置として示されているが、インタフェース２４０１及び２４０７は、単一の装置として実施することができる。これらのブロックを通るデータフローは、その転送の方向に基づいて異なるであろう。例えば、コマンド又はデータがホストからターゲットに送られる場合、ストレージ処理エンジンは、ＰＤＵをフォーマット設定するために最初に呼び出されることになり、次に、このＰＤＵがＴＣＰプロセッサに通され、ＰＤＵを有効なＴＣＰ／ＩＰセグメントにパッケージ化する。しかし、受け取ったパケットは、ストレージプロセッサエンジンに対してスケジュールに入れられる前に、ＴＣＰ／ＩＰエンジンを通って進むことになる。内部バス構造及び機能性ブロック相互接続は、性能やダイコスト要件などのために図示のものと異なる場合がある。例えば、図２３と同様に、ホストコントローラインタフェース２４０１及び２４０７、及びメモリコントローラインタフェース２４２３は、バスコントローラの一部とすることができ、このバスコントローラは、スケジューラ、又はホスト、又はストレージフロー／ＲＤＭＡコントローラ、又はセッションコントローラ、又は、以下に限定されるものではないが、セキュリティプロセッサ、又はメディアインタフェースユニット、ホストインタフェース、スケジューラ、分類プロセッサ、パケットバッファ、又はコントローラプロセッサ、又は以上の任意の組合せのような他のリソースへ又はそれからのデータパケット又は状態情報又はコマンド又はその組合せの転送を可能にするものである。

【００５７】

一例として図２４のｉＳＣＳＩプロセッサのようなＩＰストレージプロセッサにより、図２３のＴＣＰ／ＩＰプロセッサを組み込まないチップ上でストレージが行われる用途では、ＴＣＰ／ＩＰインタフェース２４０６は、ＩＰストレージプロセッサによって処理するＩＰストレージパケットをスケジュールに入れるためのスケジューラに対するインタフェースとして機能するであろう。本特許の開示を鑑みる時、同様の変形は、十分に当業者の知識の範囲内である。

【００５８】

図２５は、図１７の出力待ち行列コントローラブロック１７１２をより詳細に示す。このブロックは、図１６のネットワークメディアの独立インタフェース１６０１に送る必要があるパケットを受け取る。パケットは、送出される前に暗号化する必要があるか否かを示すためにタグ付けすることができる。コントローラは、待ち行列２５１１及びセキュリティエンジンインタフェース２５１０を通じてセキュリティエンジンに保護する必要があるパケットを待ち行列に入れる。暗号化されたパケットは、セキュリティエンジンから受け取られ、ブロック２５０９で待ち行列に入れられ、それらの宛先に送られる。このような機構がサポートされている場合は、出力待ち行列コントローラは、それらの各サービス品質（ＱＯＳ）待ち行列にパケットを割り当てることができる。プログラマブルパケット優先度セレクタ（ブロック２５０４）は、送られる次のパケットを選択し、そのパケットを適切なポート、すなわち、ポート１．．．ポートＮに対してスケジュールに入れる。ポートに付随するメディアコントローラブロック１６０１は、パケットを受け入れ、それらをその宛先に送る。

【００５９】

図２６は、一般的に、図１７の１７０８で表すストレージフロー／ＲＤＭＡコントローラブロックをより詳細に示す。ストレージフロー及びＲＤＭＡコントローラブロックは、コマンド（ストレージ、又はＲＤＭＡ、又はソケット直接、又はその組合せ）をこのプロセッサに対して待ち行列に入れるためにホストに対して必要な機能性を提供し、このプロセッサは、次に、これらのコマンドを取り出してそれらを実行し、主にコマンドが終了する時にホストプロセッサに割り込む。新しいアクティブなコマンド待ち行列（ブロック２６１１及び２６１０）と終了待ち行列（ブロック２６１２）とは、部分的にオンチップにすることができ、又は部分的にホストメモリ領域又はＩＰプロセッサに付随するメモリに入れることができ、それらからコマンドがフェッチされるか又は終了ステータスが保存さ

れる。ＲＤＭＡエンジン（ブロック２６０２）は、遠隔の直接メモリアクセスを使用可能にするのに必要な様々な機能を提供する。それは、ＲＤＭＡ領域及びアクセスキーのような情報と仮想アドレス変換機能性とを含むＲＤＭＡ検索テーブル２６０８のようなテーブルを有する。このブロック２６０２内部のＲＤＭＡエンジンは、データ転送を実行し、受け取ったＲＤＭＡコマンドを解釈して、可能であればトランザクションを実行する。ストレージフローコントローラはまた、ターゲットとイニシエータの間でデータ転送が発生した時にスケジュールに入れられた様々なコマンドの進行状態の追跡を維持する。ストレージフローコントローラは、実行のためのコマンドをスケジュールに入れ、また、ホストドライバにコマンド終了情報も提供する。ストレージフローコントローラは、ホストからの新しい要求が入っており、同様にアクティブなコマンドがアクティブコマンド待ち行列に保持されたコマンド待ち行列を提供する。ブロック２６０１のコマンドスケジューラは、接続が存在しないターゲットのために受信した新しいコマンドを、新しい接続を開始するためにスケジューラに割り当てる。スケジューラ１７０２は、図１７の１７１１で表される制御プレーンプロセッサを使用して、一般的に図１５と図１７の１７０４で表されるセッションキャッシュに接続エントリが移動される点で接続を確立し、ストレージフローコントローラブロック２６０１内の状態コントローラは、新しいコマンドをアクティブコマンドに移動して、そのコマンドを適切な接続に関連付ける。ブロック２６１０内のアクティブコマンドは、パケットプロセッサによる作動のために、検索されてスケジューラ（ブロック１７０２）に送られる。コマンドステータスに対する更新は、再度フローコントローラに提供され、このコントローラは、次に、それをブロック２６０３を通じてアクセスされるコマンド状態テーブル（ブロック２６０７）に保存する。２６０１のシーケンサは、コマンドスケジュール設定に対してプログラマブル優先度を適用し、従って、アクティブコマンドと新しいコマンドからスケジュールに入れられるべき次のコマンドを選択する。フローコントローラはまた、着信コマンド（ブロック２６１３）のための新しい要求の待ち行列を含む。新しい要求は、ホスト上でホストドライバによって適切な処理とバッファ予約が行われた状態で、アクティブコマンド待ち行列に転送される。実行のためにコマンドがスケジュールに入れられると、状態コントローラ２６０１は、ホストデータ事前取り出しマネージャ（ブロック２６１７）により、ホストインタフェースブロック２７０７のＤＭＡエンジンを使用して、ホストメモリからデータの事前取り出しを開始し、従って、コマンドが実行される時に、パケットプロセッサ複合システムに対して提供する準備が完了したデータを保存する。出力待ち行列コントローラ（ブロック２６１６）は、ホストコントローラインタフェース（ブロック２６１４）と協働して、データ転送を可能にする。ストレージフロー／ＲＤＭＡコントローラは、解決されたターゲット／イニシエータと、迅速な検索のため及びアクティブ接続にコマンドを関付けるために確立された接続とを関連付けるターゲット－イニシエータテーブル（ブロック２６０９）を維持する。コマンドシーケンサはまた、実行されているコマンドがＲＤＭＡコマンドの場合か、又はストレージ転送が接続開始時にＲＤＭＡ機構を通じて実行されるように交渉された場合は、ＲＤＭＡエンジンと協働することができる。ＲＤＭＡエンジン２６０２は、上述のように、複数ＲＤＭＡ領域、アクセス制御キー、及び仮想アドレス変換ポインタを受け入れるための機能性を提供する。ホストアプリケーション（ユーザアプリケーション又はＯＳカーネル機能、ストレージ、又は、ウェブページ又はビデオファイルなどのダウンロードのような非ストレージとすることができる）は、それが、開示されたプロセッサを用い、関連ホストドライバによって提供されたサービスを通じてＲＤＭＡトランザクションでの使用を希望するメモリ領域を登録する。これが行われた状態で、ホストアプリケーションは、その遠隔端末のピアにこの情報を通信する。この時点で、遠隔マシン又はホストは、実質的にホストの介在なしに、両端のＲＤＭＡブロックによって供給されたＲＤＭＡコマンドを実行することができる。ＲＤＭＡ転送は、領域からの読取、特定オフセットを有するある一定数のバイト、又は属性が類似の書込のような作動を含むことができる。ＲＤＭＡ機構はまた、２つの端末ノード間での通信パイプの作成のために有用な送信機能性を含むことができる。これらの機能は、クラスター内のサーバ、又は、更に可能性があるのは、２つの

異なるクラスターのサーバ内のサーバ、又は他のそのようなクラスター化システムで実行される２つのアプリケーションのバッファ間で大量のデータ転送が要求される、アプリケーションのクラスター化において有用である。ストレージデータの転送は、それが実質的にホストの介在なしに大きなブロックのデータ転送を可能にするので、ＲＤＭＡ機構を使用しても達成される。両端のホストは、ＲＤＭＡ転送を行なうこと、及び共有されたアクセス制御キーを通じてメモリ領域と認可を割り当てることに同意するために最初に関わり合っている。更に、２つのノード間のデータ転送は、利用可能なバッファスペースとバッファ転送クレジットが２つの端部ノードによって維持されている限りは、ホストプロセッサの介入がなくても継続することができる。ストレージデータ転送プロトコルは、ＲＤＭＡプロトコルを使用することに同意し、それを両端で使用可能にすることにより、ＲＤＭＡの上で実行されるであろう。図２６のストレージフローコントローラ及びＲＤＭＡコントローラは、次に、ＲＤＭＡコマンドを使用して、ストレージコマンド実行とデータ転送を遂行することができる。予想されたデータ転送が終了すると、終了待ち行列２６１２を使用して、ストレージコマンド終了ステータスがホストに通信される。ネットワークから到着した着信データパケットは、図１７のパケットプロセッサ複合システムによって処理され、次にＰＤＵが抽出され、ストレージ／ＲＤＭＡデータパケットの場合、図２６のフローコントローラに呈示される。これらは、次に着信待ち行列ブロック２６０４に割り当てられ、受信バッファのメモリ記述子の検索によって端末宛先バッファに転送され、次に、ホストインタフェースブロック２７０７のＤＭＡエンジンを使用してＤＭＡを実行する。ＲＤＭＡコマンドはまた、ＲＤＭＡ初期化のように保護キー検索及びアドレス変換を通過することができる。

以上の内容はまた、ＲＤＭＡ能力、又はＲＤＭＡ機構、又はＲＤＭＡ機能の一部と考えることができる。

【００６０】

図２７は、図１７の１７１０で表されるホストインタフェースコントローラをより詳細に示す。ホストインタフェースブロックは、ホストバスに対して物理的なインタフェースを提供するホストバスインタフェースコントローラ（ブロック２７０９）を含む。ホストインタフェースブロックは、スイッチ又はゲートウェイ又は同様な構成に組み込まれる時は、システムアーキテクチャに依存し、ファブリックインタフェースとして又はメディア独立インタフェースとして実行することができ、仮想出力待ち行列及び／又は他の品質のサービス機能として提供することができる。ブロック２７０８の処理コントローラ部分は、様々なバス処理を実行し、それらのステータスを維持して終了するために要求された処理を行う。ホストコマンドユニット（ブロック２７１０）は、ホストによって配送されたコマンドを実行するために、ホストバス構成レジスタと１つ又はそれ以上のインタープリタとを含む。ホストドライバは、ホスト出力待ち行列インタフェース２７０３上でこれらのコマンドをこのプロセッサに提供する。コマンドは、構成レジスタやスケジュールＤＭＡ転送の設定、必要に応じてＤＭＡ領域及び許可の設定、セッションエントリや検索セッションデータベースの設定、及びＲＤＭＡエンジンの構成などのような様々な機能を提供する。ストレージ及び他のコマンドはまた、ＩＰプロセッサによる実行のためにこのインタフェースを使用して転送することができる。

【００６１】

図２８は、図１７のセキュリティエンジン１７０５をより詳細に示す。図示のセキュリティエンジンは、例えば、ＩＰＳＥＣのような基準によって要求されるような認証及び暗号化及び暗号解読サービスを提供する。セキュリティエンジンによって提供されるサービスは、複数の認証及びセキュリティアルゴリズムを含むことができる。セキュリティエンジンは、上述のように、プロセッサをオンボードにすることができ、又は別のシリコンチップの一部とすることができる。ＩＰセキュリティサービスを提供する外部セキュリティエンジンは、着信パケットのパケット処理の最初のステージの１つとして、かつ、出ていくパケットの最後のステージの１つとして、データフロー内の同様な位置に配置される。図示のセキュリティエンジンは、セキュリティ基準として適合した非常にハードウエア性

能効率的なアルゴリズムの最新暗号化基準（ＡＥＳ）ベースの暗号化及び暗号解読サービスを提供する。このブロックは、一例としてＤＥＳや３ＤＥＳのような他のセキュリティ機能を同様に提供することができる。セキュリティと認証のためにサポートされたアルゴリズムと機能は、シリコンコストと開発コストから影響されたものである。選択されたアルゴリズムはまた、ＩＰストレージ基準によって要求されたものである。認証エンジン（ブロック２８０３）は、利用可能なアルゴリズムの一例としてＳＨＡ－１を含むように示されている。このブロックは、ＩＰセキュリティ基準で規定されたようなメッセージのダイジェストと認証機能を提供する。セキュリティ及びメッセージ認証サービスが要求される時、データは、これらのブロックを通過する。ターゲットに向けて出ていく途中のクリアなパケットは暗号化され、次に、必要に応じて適切なエンジンを使用して認証される。受け取った安全なパケットは、反対の順序で同じ段階を進む。安全なパケットは認証され、次に、このブロックのエンジン２８０３及び２８０４を使用して暗号解読される。セキュリティエンジンはまた、接続のために確立されたセキュリティコンテキストメモリ（ブロック２８０９）にセキュリティアソシエーションを維持する。セキュリティアソシエーションは、（安全なセッションのインデックス、セキュリティキー、使用されるアルゴリズム、セッションの現在の状態などを含むことができ）、メッセージ認証及び暗号化／暗号解読サービスを実行するために使用される。メッセージ認証サービス及び暗号化／暗号解読サービスを互いに独立して使用することも可能である。

【００６２】

図２９は、全体的に図１７の１７０４で表されるセッションキャッシュ及びメモリコントローラ複合システムをより詳細に示す。メモリ複合システムは、キャッシュ又はメモリ又はその組合せとして、本特許ではセッション／グローバルセッションメモリ又はセッションキャッシュと呼ばれるＴＣＰ／ＩＰセッションデータベースのためのキャッシュ／メモリアーキテクチャを含む。セッションキャッシュ検索エンジン（ブロック２９０４）は、特定のセッションキャッシュエントリを検索するための機能性を提供する。この検索ブロックは、提供されたフィールドの外にハッシュインデックスを作り出すか、又はハッシュキーを受け入れてセッションキャッシュエントリを検索することができる。キャッシュアレイ内にハッシュインデックスとのタグの符合がない場合、検索ブロックは、このキーを使用して外部メモリからセッションエントリを見つけ、現在のセッションキャッシュエントリをそのセッションエントリと取り替える。それは、要求のパケットプロセッサ複合システムにセッションエントリフィールドを提供する。ローカルプロセッサ複合システムキャッシュ内に存在するキャッシュエントリは、グローバルキャッシュに共有されるようにマーク付けされる。従って、任意のプロセッサがこのキャッシュエントリを要求すると、それは、グローバルキャッシュと要求しているプロセッサとに転送され、グローバルキャッシュにそのようにマーク付けされる。セッションメモリコントローラはまた、退去させられたキャッシュエントリをこのブロック内部のグローバルキャッシュに移動させる役割を果たす。従って、セッションエントリの任意の要求者に対して、最新のセッション状態だけがいつでも利用可能である。セッションキャッシュが満杯の場合、新しいエントリにより最低使用頻度のエントリを退去させることができる。セッションメモリは、所定の処理技術で使用することができるシリコンスペースに応じて、単一回路又は複数回路キャッシュ又はハッシュインデックスメモリ、又はその組合せにすることができる。ネットワークスイッチ又はルータのためのネットワーキングアプリケーションでは、一般的にパケット間で利用可能な基準特性の局所性があまり多くなく、従って、キャッシュの使用はキャッシュミスのために多くの性能改善をもたらさないという点で、セッションデータベースエントリを保存するためのキャッシュの使用は独特である。しかし、ストレージ処理は、２つの端部システム間の持続時間が長い処理であり、大量のデータを交換することができる。クラスター化又はメディアサーバのように２つのノード間で大量のデータ転送が発生するこのシナリオ又はケースでは、キャッシュベースのセッションメモリアーキテクチャは、オフチップメモリからのデータ転送を大幅に低減することにより、性能的な恩典を達成することになる。セッションキャッシュのサイズは、利用可能なシリコンダイ面積の

関数であり、トレードオフによって性能に影響を及ぼすことがある。メモリコントローラブロックはまた、パケット、パケットのフラグメント、又はメモリ内のオペレーティングデータを保存する必要がある他のブロックにサービスを提供する。メモリインタフェースは、サポートする必要のある予想データ帯域幅に依存して、単一又は複数の外部メモリコントローラ（ブロック２９０１）を提供する。これは、ＤＲＡＭ又はＳＲＡＭ又はＲＤＲＡＭ、又は他の動的又は静的ＲＡＭ、又はその組合せのための１つ又は複数の２倍データ速度コントローラとすることができる。図は、複数コントローラを表すが、その数は必要な帯域幅とコストに応じて変動する。メモリ複合システムはまた、セッションデータベースメモリブロックによって維持された再発信待ち行列上でセッション自体を待ち行列に入れるセッションに対する再発信タイムアウトに使用するためのタイマ機能性を提供することができる。

【００６３】

図３０は、分類エンジンのデータ構造の詳細を示す。これは、分類エンジンのデータ構造を編成するための１つの方法である。分類データベースは、ツリー構造（ブロック３００１）として表され、ツリー内のノード（ブロック３００３）とノードに関連したアクション（ブロック３００８）とは、分類エンジンがツリーの下方にウォークし、特定のノード値の比較を行うことを可能にする。ノード値とそれらが呈示するフィールドは、プログラム可能である。アクションフィールドは、フィールドが特定のノード値に符合する時に抽出される。アクションアイテムは、次の段階を規定し、この段階は、新しいフィールドの抽出と比較を含むことができ、このノード値の対に関連する特定のデータフィールド上でＡＬＵ作動のような他の作動を実行するか、又は端末ノードを表示することができ、その時点で特定パケットの分類が終了する。データ構造は、それがパケットスケジューラから受け取るパケットを分類するために分類エンジンによって使用される。符合する値を使用して検索されたアクションアイテムは、パケットの異なるフィールドを繰り返す一方で、結果コンパイラによって使用され、一般的にパケットのヘッダの前でパケットに接続される分類タグを生成する。分類タグは、次に、プロセッサの残りによって基準として使用され、分類の結果に基づいて取る必要があるアクションを決める。プログラマブルな特性を有する分類子は、分類ツリー構造のシステム変更を可能にし、分類の必要性の異なるシステムでプロセッサの使用を可能にする。分類エンジンはまた、分類のツリーノード値アクションの構造の一部としてプログラムすることができるストレージ／ネットワークポリシーの作成を可能にし、ＩＰベースのストレージシステムにおいて非常に強力な機能を提供する。ポリシーは、このプロセッサを使用するシステムの管理を増強し、特定のポリシー又は規則が符合した時又は侵害された時の強化機能を可能にする。分類エンジンは、特定のシステム制約によって要求された時に、外部の要素を使用して分類データベースの拡張を可能にする。ツリーとノードの数は、シリコン面積と性能トレードオフとに基づいて決まる。データ構造の構成要素は、分類エンジンの様々なブロックに維持され、分類シーケンサによって使用され、構造によってパケット分類の方向付けが行われる。分類データ構造は、ターゲットソリューションに応じて、表示されたものよりも多いか又は少ないフィールドを要求することができる。従って、分類のコアの機能性は、基本となるアーキテクチャから離れることなく、より少ない要素と構造を使用して達成することができる。分類処理は、プログラムに従ってツリーとノードを通ってウォークする。特定ノードアクションにより、分類の残りのフィールドのために新しいツリーの使用を可能にすることができる。従って、分類処理は、ツリールートで始まり、ノードを通って進行し、最後はリーフノードに到達する。

【００６４】

図３１は、イニシエータとターゲットの間の読取作動を示す。イニシエータは、処理を開始するためにターゲットにＲＥＡＤコマンド要求（ブロック３１０１）を送信する。これは、アプリケーション層の要求であり、特定のＳＣＳＩプロトコルコマンドにマップされ、ＩＰベースのストレージネットワーク内でＲＥＡＤプロトコルデータユニット（ブロック３１０２）として移送されるものである。ターゲットは、要求されたデータ（ブロッ

ク３１０３）を準備し、最大転送ユニット制約を満足するように分割された読取応答ＰＤＵ（ブロック３１０５）を提供する。イニシエータは、次に、ＩＰパケットからデータを検索（ブロック３１０６）し、それは、次に、この作動のために割り当てられた読取バッファに保存される。全てのデータが転送された状態で、ターゲットは、コマンド終了の応答を行ってステータスを感知する（ブロック３１０７）。完全な転送が終了する（ブロック３１０９）と、イニシエータは、次にコマンドを退去させる。ターゲットに何らかのエラーがあり、何らかの理由でコマンドが中途終了している時は、イニシエータによって単独で回復の手順を開始することができる。このトランザクションは、プロトコルのＰＤＵとしてのｉＳＣＳＩのようなＩＰベースのストレージプロトコル上のデータ移送を有する標準的ＳＣＳＩのＲＥＡＤトランザクションである。

【００６５】

図３２は、図３１に示すトランザクションの受信「ＲＥＡＤ　ＰＤＵ」の１つに対する本発明のＩＰプロセッサ内部のデータフローを示す。内部データフローは、イニシエータ端末上のＩＰプロセッサによって受信した読取データＰＤＵに対して示される。この図は、パケットが通過する作動の様々なステージを表す。ステージは、パケットがトラバースするパイプラインステージと考えることができる。トラバースするパイプのステージの数は、受け取ったパケットの種類に依存する。図は、確立された接続上で受け取ったパケットに対するパイプステージを表す。パケットは、以下の主要なパイプステージを通ってトラバースする。

【００６６】

１．ブロック３２０７に示す主要な段階を有するブロック３２０１の受信パイプステージ。パケットは、メディアアクセスコントローラによって受信される。パケットが検出され、プリアンブル／トレーラーが除去され、パケットは、第２層ヘッダとペイロードと共に抽出される。これは、第２層確認、及び全てのエラー検出が意図する受取人に対して発生するステージである。確立されたポリシー毎に適用されるサービス品質検査がある場合もある。パケット確認がクリアされた状態で、パケットは、入力待ち行列に入れられる。

２．ブロック３２０８に示す主要な段階を有するブロック３２０２のセキュリティパイプステージ。パケットは、入力待ち行列から分類エンジンまで動かされ、そこでセキュリティ処理のための迅速な判断が行われ、パケットがセキュリティ処理を通過する必要がある場合、セキュリティパイプステージに入る。パケットがクリアなテキストで受信されて認証を必要としない場合、セキュリティパイプステージは省略される。セキュリティパイプステージはまた、セキュリティエンジンがＩＰプロセッサと統合されていない場合は、省略することができる。パケットは、第１にこの接続に対するセキュリティアソシエーションがメモリから検索され、パケットが選択されたメッセージ認証アルゴリズムを使用して認証されるセキュリティエンジンの様々なステージを通過する。パケットは、次に、セッションのために確立されたセキュリティキーを使用して暗号解読される。パケットがクリアなテキストになった状態で、それは、入力待ち行列コントローラに対して再度待ち行列に入れられる。

３．ブロック３２０９に示す主要な段階を有するブロック３２０３の分類パイプステージ。スケジューラは、入力待ち行列からクリアなパケットを検索し、分類のためにパケットをスケジュールに入れる。分類エンジンは、第３層及びより高い層の分類のためにパケットから関連フィールドを抽出するような様々なタスクを実行し、ＴＣＰ／ＩＰ／ストレージプロトコルなどを識別し、それらの分類タグを生成し、分類エンジンでプログラムされたポリシーに応じて、パケットを排除したりバイパスのためにパケットにタグ付けするようなアクションを取ることができる。分類エンジンはまた、実行を容易にするためのパケットヘッダ及びペイロードのマーク付けと共に、セッション又はこのエンジンが属するフローでパケットにタグ付けすることができる。分類エンジンのプログラムに依存して、列挙されたタスクのいくつかを実行することができ、又は実行されなくてもよく、他のタスクが実行されてもよい。分類が実行されると、パケットに分類タグが追加され、処理するためにスケジューラに対してパケットが待ち行列に入れられる。

４．ブロック３２１０に示す主要な段階を有するブロック３２０４のスケジュールパイプステージ。分類パケットが、分類エンジン待ち行列から検索され、処理のためにスケジューラ内に保存される。スケジューラは、パケットヘッダから発信元及び宛先フィールドのハッシュを実行し、分類子によって実行されない場合にパケットが属するフローを識別する。フローの識別が行われた状態で、パケットは、フローの従属性に基づいて実行リソース待ち行列に割り当てられる。新しいパケットを受け入れるためにリソースが利用可能になると、待ち行列の次のパケットが、そのリソースに対する実行のために割り当てられる。

５．ブロック３２１１に示す主要な段階を有するブロック３２０５の実行パイプステージ。パケットは、このパケットを実行するためにリソースが利用可能になると、実行パイプステージに入る。パケットは、パケットを実行するために想定されたパケットプロセッサ複合システムに転送される。プロセッサは、パケットに接続された分類タグを見て、パケットのために要求される処理段階を決める。これがＩＰベースのストレージパケットの場合、このセッションのためのセッションデータベースエントリが検索される。ローカルセッションキャッシュが既にセッションエントリを保持する場合は、データベースアクセスは要求されない。フローに基づいてパケット割当が行われた場合は、グローバルセッションメモリからセッションエントリを検索する必要はない。パケットプロセッサは、次に、ＴＣＰエンジン／ストレージエンジンを開始し、それらの作動を実行する。ＴＣＰエンジンは、チェックサム、シーケンス番号検査、必要なＣＲＣ作動を使用したフレーム指示検査、及びＴＣＰ状態更新を含む様々なＴＣＰ検査を実行する。次に、ストレージＰＤＵが抽出され、実行のためにストレージエンジンに割り当てられる。ストレージエンジンは、ＰＤＵ内のコマンドを変換し、この特定のケースでは、それをアクティブなセッションに対する読取応答として識別する。次に、それは、ペイロードの整合性とシーケンスの整合性を確認し、次に、セッションデータベースエントリ内のストレージフロー状態を更新する。宛先バッファのメモリ記述子はまた、セッションデータベースエントリから検索され、抽出されたＰＤＵペイロードはストレージフロー／ＲＤＭＡコントローラに対して、それらのホストインタフェースブロックはＤＭＡに対して、データは最終のバッファ宛先に対して、待ち行列に入れられる。データは、メモリ記述子と実行するコマンド／作動と共に、フローコントローラに配信することができる。この場合、このアクティブ読取コマンドに対してデータを入れる。ストレージフローコントローラは、そのアクティブコマンドデータベースを更新する。実行エンジンは、パケットが既に退去しており、パケットプロセッサ複合システムがそのシステムの次のコマンドを受け取る準備ができていることをスケジューラに示す。

６．ブロック３２１２に示す主要な段階を有するブロック３２０６のＤＭＡパイプステージ。ストレージフローコントローラがメモリ記述子、コマンド、及びフロー状態の適切な確認を行った状態で、このコントローラは、ホストメモリに対する転送のためにＤＭＡエンジンにデータブロックを送る。ＤＭＡエンジンは、そのようなＱＯＳ機構がプログラムされるか又は実行されると、優先度ベースの待ち行列を実行することができる。データは、ＤＭＡを通じてホストメモリ位置に転送される。これがコマンドの最後の作動の場合、ホストドライバにコマンド実行の終了が表示される。これがコマンドに対する最後の作動であり、コマンドが終了待ち行列に対して待ち行列に入れられている場合、コマンドに対して割り当てられたリソースは、新しいコマンドを受け入れるために解放される。コマンド統計データは、性能分析、ポリシー管理、他のネットワーク管理、又は統計目的のために要求される場合があるので、終了ステータスと共に収集して転送することができる。

【００６７】

図３３は、イニシエータとターゲットの間の書込コマンド作動を示す。イニシエータは、トランザクションを開始するためにターゲットにＷＲＩＴＥコマンド（３３０１）を送る。このコマンドは、ＩＰストレージネットワーク上で「ＷＲＩＴＥ　ＰＤＵ」（ブロック３３０２）として移送される。レシーバは、新しい要求待ち行列に受け取ったコマンドを入れる。作動の古いコマンドが終了した状態で（ブロック３３０４）、レシーバは、リ

ソースを割り当ててコマンド（ブロック３３０５）に対応するＷＲＩＴＥデータを受け入れる。このステージで、レシーバは、受け取る予定のデータ量とどの位置から受け取るかの指示を使用して、イニシエータに転送待機中（Ｒ２Ｔ）ＰＤＵ（ブロック３３０６）を出す。イニシエータは、Ｒ２Ｔ要求のフィールドを変換し、データパケット（ブロック３３０７）を、受け取ったＲ２Ｔ毎にレシーバに送る。イニシエータとターゲットの間のこの交換のシーケンスは、コマンドが終了されるまで続く。成功したコマンドの終了又はエラー条件は、応答ＰＤＵとしてターゲットによってイニシエータに通信され、それが次にコマンドを終了させる。イニシエータは、エラーの場合に回復処理を開始するように要求することができる。これは、図３３の交換には示されていない。

【００６８】

図３４は、「Ｒ２Ｔ　ＰＤＵ」の１つに対する本発明のＩＰプロセッサ内部のデータフローと、図３３に示す書込トランザクションの追随する書込データとを示す。イニシエータは、ネットワークメディアインタフェースを通じてＲ２Ｔパケットを受け取る。パケットは、受信、セキュリティ、分類、スケジュール、及び実行を含む図３２の「ＲＥＡＤ　ＰＤＵ」と同様の対応するブロック３４１５、３４１６、３４０９、及び３４１０内の詳細な主要段階を使用して、全てのステージ（ブロック３４０１、３４０２、３４０３、及び３４０４）を通過する。セキュリティ処理は、この図には表示されていない。これらのステージに続いてＲ２Ｔは、図３４に示すＤＭＡステージを使用して、書込データフェッチを開始する（ブロック３４０５及び３４１１）。書込データは、次に分割され、実行ステージ（ブロック３４０６及び３４１２）を通じてＴＣＰ／ＩＰパケットに置かれる。ＴＣＰ及びストレージセッションＤＢエントリは、Ｒ２Ｔに応答して転送されたデータを用いてＷＲＩＴＥコマンドに対して更新される。パケットは、次に、出力待ち行列コントローラに対して待ち行列に入れられる。接続セキュリティの合意に基づいて、パケットは、セキュリティパイプステージ（ブロック３４０７及び３４１３）に入ることができる。パケットが暗号化されてメッセージ認証コードが生成された状態で、パケットは、宛先への送信のためにメディアインタフェースに対して待ち行列に入れられる。このステージ（ブロック３４０８及び３４１４）の間に、パケットは、パケットプロセッサによってまだ為されていなければ、第２層ヘッダ内にカプセル化され、かつ送信される。パイプラインの各ステージで従うべき段階は、上述の「ＲＥＡＤ　ＰＤＵ」パイプステージと類似であり、この図に示されている書込データパケットステージのための付加的なステージを有する。各ステージで実行される特定の作動は、コマンドの種類、セッションの状態、コマンドの状態、及び設定することができるポリシーに対する他の様々な構成に依存する。

【００６９】

図３５は、イニシエータとターゲットの間のＲＤＭＡ機構を使用したＲＥＡＤデータ転送を示す。イニシエータとターゲットは、ＲＤＭＡデータ転送を開始させる前にＲＤＭＡバッファを登録する（ブロック３５０１、３５０２、及び３５０３）。イニシエータは、予想される受け取り側としてのＲＤＭＡを使用してＲＥＡＤコマンド（ブロック３５１０）を出す。このコマンドは、ターゲット（ブロック３５１１）に移送される。ターゲットは、読み込まれるデータを準備し（ブロック３５０４）、次に、ＲＤＭＡ書込作動（ブロック３５０５）を実行し、書込データをイニシエータにおいてホストの介在なしに直接ＲＤＭＡバッファに保存する。作動の終了は、コマンド終了応答を使用して示される。

【００７０】

図３６は、ＲＥＡＤコマンドフローを実行するＲＤＭＡ書込パケットの内部アーキテクチャデータフローを示す。ＲＤＭＡ書込パケットはまた、ネットワークインタフェース上で受け取った他の任意の有効なパケットと同じパイプステージに従う。このパケットは、受信パイプステージ（ブロック３６０１及び３６０７）内の第２層処理を通過し、そこから、それは、セキュリティ処理の必要性を検出するためにスケジューラに対して待ち行列に入れられる。パケットが暗号解読される必要がある場合、それは、セキュリティパイプステージに入る（ブロック３６０２及び３６０８）。暗号解読されたパケットは、次に、プログラムされていた分類タスクを実行するために、分類エンジンに対してスケジュール

に入れられる（ブロック３６０３及び３６０９）。分類が終了した状態で、タグ付けされたパケットは、スケジュールパイプステージ（ブロック３６０４及び３６１０）に入り、そこで、スケジューラは、フローベースのスケジュール設定に依存して、このパケットをリソース特定の待ち行列に割り当てる。意図されたリソースがこのパケットを実行する準備が完了している場合、それは、そのパケットプロセッサ複合システム（ブロック３６０５及び３６１１）に転送され、そこで、全てのＴＣＰ／ＩＰ確認、検査、及び状態更新が行われ、ＰＤＵが抽出される。次に、ストレージエンジンは、ＲＤＭＡを使用して実施されたストレージＰＤＵに対するストレージフローに属するとしてＰＤＵを識別し、ＲＤＭＡコマンドを解釈する。この場合、特定のＲＤＭＡバッファに書き込むのはＲＤＭＡである。このデータは、抽出されて、ＲＤＭＡ領域変換及び保護検査を行うストレージフロー／ＲＤＭＡコントローラブロックに通され、パケットは、ホストインタフェース（ブロック３６０６及び３６１２）を通じてＤＭＡに対して待ち行列に入れられる。パケットがパケットプロセッサ複合システムを通じた作動を終了した状態で、スケジューラに通知され、パケットは、スケジューラ内に保持された状態から退去させられる。ＤＭＡステージでＲＤＭＡデータ転送が終了した状態で、これがストレージコマンド実行を終了させる最後のデータ転送の場合、そのコマンドは退去させられ、コマンド終了待ち行列に割り当てられる。

【００７１】

図３７は、ＲＤＭＡ読取作動を使用するストレージ書込コマンド実行を示す。イニシエータ及びターゲットは、ＲＤＭＡコントローラを使用して最初にそれらのＲＤＭＡバッファを登録し、次に、同じくバッファをピアに通知する。次に、イニシエータは、ターゲットに書込コマンド（ブロック３７０１）を出し、それは、ＩＰストレージＰＤＵを使用して移送される。受け取り側は、最初にＲＤＭＡバッファを割当てることによって書込コマンドを実行し、書込を受け取り、次にイニシエータに対してＲＤＭＡ読取を要求する（ブロック３７０７及び３７０８）。イニシエータから書き出されるデータは、次に、ＲＤＭＡ読取応答パケット（ブロック３７０７及び３７０８）として供給される。受け取り側は、いかなるホストの介入もなく、パケットを直接ＲＤＭＡバッファに保存する。読取要求がセグメントサイズよりも大きなデータに対するものだった場合、ＲＥＡＤ要求に応答して、複数のＲＥＡＤ応答ＰＤＵがイニシエータに送られるであろう。データ転送が終了した状態で、イニシエータに終了ステータスが移送され、ホストに対してコマンド終了が示される。

【００７２】

図３８は、図３７に示すフロー処理の１つのセクションに対するＲＤＭＡ読取要求及び得られる書込データ転送のデータフローを示す。データフローは、図３４の書込データフローと非常に似ている。ＲＤＭＡ読取要求パケットは、受信、分類、スケジュール、及び実行（ブロック３８０１、３８０２、３８０３、３８０４、３８１５、３８１６、３８０９、及び３８１０）を含む様々な処理パイプステージを通って流れる。この要求が実行された状態で、それは、ＲＤＭＡ読取応答パケットを生成する。ＲＤＭＡ応答は、最初にシステムメモリから要求されたデータのＤＭＡ（ブロック３８０５及び３８１１）を行うことによって生成され、次に、実行ステージ（ブロック３８０６及び３８１２）を通してセグメントとパケットを生成する。適切なセッションデータベースエントリが更新され、データパケットは、必要に応じてセキュリティステージに進む（ブロック３８０７及び３８１３）。安全又はクリアなパケットは、次に、送信ステージに対して待ち行列に入れられ（ブロック３８０８及び３８１４）、このステージは、適切な第２層更新を実行してパケットをターゲットに送信する。

【００７３】

図３９は、イニシエータから開始されたストレージコマンドに対するイニシエータコマンドフローをより詳細に示す。図示のように、以下は、コマンドが従う主要段階のいくつかである。

１．ホストドライバは、コマンドをストレージフロー／ＲＤＭＡコントローラのプロセ

ッサコマンド待ち行列に入れる。
２．ホストは、作動のため及びリソースを予約するためにコマンドが正常にスケジュールに入れられたかを通知される。
３．ストレージフロー／ＲＤＭＡコントローラは、ターゲットに対する接続が確立された場合、作動のためのコマンドをパケットスケジューラに対してスケジュールに入れる。そうでなければ、コントローラは、ターゲットセッション開始を開始し、セッションが確立された状態で、コマンドは、パケットスケジューラに対してスケジュールに入れられる。
４．スケジューラは、このコマンドを受け入れる準備が完了したＳＡＮパケットプロセッサの１つにコマンドを割り当てる。
５．プロセッサ複合システムは、セッションエントリのために、要求をセッションコントローラに送る。
６．セッションエントリが、パケットプロセッサ複合システムに供給される。
７．パケットプロセッサは、ＰＤＵとしてコマンドを運び、出力待ち行列に対してスケジュールに入れられるパケットを形成する。
８．コマンドＰＤＵは、それをターゲットに送るネットワークメディアインタフェースに与えられる。
これは、イニシエータとターゲットの間で接続が確立された時に、主にイニシエータからターゲットまで大部分のコマンドが追従する高レベルのフローである。

【００７４】

図４０は、読取パケットデータフローをより詳細に示す。ここでは、読取コマンドは、最初に図３９に示すフローと同様のフローを使用し、イニシエータからターゲットに送られる。ターゲットは、図４０に示すようなフローに従う読取応答ＰＤＵをイニシエータに送る。図示のように、読取データパケットは、以下の主要段階を通過する。
１．入力パケットは、ネットワークメディアインタフェースブロックから受信される。
２．パケットスケジューラは、入力待ち行列からパケットを検索する。
３．パケットは、分類のためにスケジュールに入れられる。
４．分類されたパケットは、分類タグと共に分類子から戻る。
５．分類とフローベースのリソース割当とに基づいて、パケットは、パケット上で作動するパケットプロセッサ複合システムに割り当てられる。
６．パケットプロセッサ複合システムは、セッションキャッシュ内のセッションエントリを検索する（ローカルに存在しない場合）。
７．セッションキャッシュエントリが、パケットプロセッサ複合システムに戻される。
８．パケットプロセッサ複合システムは、ＴＣＰ／ＩＰ作動／ＩＰストレージ作動を実行し、ペイロード内の読取データを抽出する。ＭＤＬ（メモリ記述子リスト）のような適切な宛先タグを有する読取データが、ホストインタフェース出力コントローラに提供される。
９．ホストＤＭＡエンジンは、システムバッファメモリに読取データを転送する。
これらの段階のいくつかは、安全なパケットフローが表された図３２により詳細に与えられ、一方、図４０は、クリアテキスト読取パケットフローを表す。本特許に示すこのフロー及び他のフローは、本特許の教示により当業者が容易に実施することができる開示されたプロセッサの適切なリソースを使用することにより、ストレージ及び非ストレージデータ転送に適用可能である。

【００７５】

図４１は、書込データフローをより詳細に示す。書込コマンドは、図３９のフローと同様のフローに従う。イニシエータは、書込コマンドをターゲットに送る。ターゲットは、ターゲットが指定量のデータを受け取る準備が完了していることをイニシエータに示す転送待機中（Ｒ２Ｔ）ＰＤＵによりイニシエータに応答する。イニシエータは、次に、要求されたデータをターゲットに送る。図４１は、イニシエータからターゲットまで要求された書込データパケットが従うＲ２Ｔを示す。このフローで追従される主な段階は、以下の

通りである。
１．入力パケットが、ネットワークメディアインタフェースブロックから受信される。
２．パケットスケジューラは、入力待ち行列からパケットを検索する。
３．パケットは、分類のためにスケジュールに入れられる。
４．分類されたパケットは、分類タグと共に分類子から戻る。
ａ．分類とフローベースのリソース割当とに依存して、パケットは、パケット上で作動するパケットプロセッサ複合システムに割り当てられる。
５．パケットプロセッサ複合システムは、セッションキャッシュ内のセッションエントリを検索する（ローカルに存在しない場合）。
６．セッションキャッシュエントリが、パケットプロセッサ複合システムに戻される。
７．パケットプロセッサは、「Ｒ２Ｔ　ＰＤＵ」を判断し、ストレージフロー／ＲＤＭＡコントローラへの要求で書込データを要求する。
８．フローコントローラは、ホストインタフェースに対してＤＭＡを開始する。
９．ホストインタフェースは、ＤＭＡを実行し、ホスト入力待ち行列にデータを戻す。
１０．パケットプロセッサ複合システムは、ホスト入力待ち行列からデータを受け取る。
１１．パケットプロセッサ複合システムは、有効なＰＤＵとパケットをデータの周りに形成し、適切なセッションエントリを更新し、出力待ち行列にパケットを転送する。
１２．パケットは、データパケットを宛先に送信する出力ネットワークメディアインタフェースブロックに転送される。

【００７６】
図４１のフローは、クリアテキストデータ転送を示す。データ転送が安全である必要がある場合は、フローは、出力データパケットが１１ａ及び１１ｂとラベル付けされた矢印で示すような安全パケットを通るようにルーティングされた図４３に示すフローと同様である。入力Ｒ２Ｔパケットはまた、安全である場合に、セキュリティエンジンを通るようにルーティングされるであろう（これは、図示されていない）。

【００７７】
図４２は、パケットが暗号テキストか又は安全である場合の読取パケットフローを示す。このフローは、上述の関連する説明を使用して図３２により詳細に示されている。安全な読取フローとクリアな読取フローの主な相違点は、パケットが分類子によって安全パケットとして最初に分類され、従って、セキュリティエンジンにルーティングされることである。これらの段階は、２ａ、２ｂ、及び２ｃとラベル付けされた矢印で示されている。セキュリティエンジンは、パケットを暗号解読し、メッセージの認証を実行し、矢印２ｄで示すような更なる処理のために入力待ち行列にクリアパケットを転送する。クリアパケットは、次に、スケジューラによって検索され、図４２の２ｅ及び３とラベル付けされた矢印で示すように、分類エンジンに供給される。残りの段階と作動は、上述の図４０のものと同様である。

【００７８】
図４４は、ＲＤＭＡバッファ通知フローを示す。このフローは、図３９に示すような他の全てのストレージコマンドフローと非常に似ているように示されている。主な段階で取られる詳細なアクションは、コマンドによって異なる。ＲＤＭＡバッファ通知とレジストレーションに対して、ＲＤＭＡ領域ＩＤが作成され、この領域に対するアドレス変換機構と共に記録される。ＲＤＭＡレジストレーションはまた、アクセス制御のための保護キーを含み、ＲＤＭＡ転送のために必要な別のフィールドを含むことができる。コマンドのためのパケットを作り出す段階は、図３９の段階と類似である。

【００７９】
図４５は、ＲＤＭＡ書込フローをより詳細に示す。ＲＤＭＡ書込は、ＲＤＭＡ書込を受け取るイニシエータに対する通常の読取ＰＤＵのように見える。ＲＤＭＡ書込パケットは、図４０に示す読取ＰＤＵのような同じ主要フローの段階に従う。ＲＤＭＡ転送は、ＲＤＭＡアドレス変換と領域アクセス制御キー検査とを伴い、他のセッションエントリ以外に

ＲＤＭＡデータベースエントリを更新する。主要なフローの段階は、通常の読取応答ＰＤＵと同じである。

【００８０】

図４６は、ＲＤＭＡ読取データフローをより詳細に示す。この図は、イニシエータによってターゲットから受け取られるＲＤＭＡ読取要求と、イニシエータからターゲットに書き出されるＲＤＭＡ読取データとを示す。このフローは、ストレージ書込コマンドが従うＲ２Ｔ応答と非常に似ている。このフローでは、ストレージ書込コマンドは、ＲＤＭＡ読取を使用して達成される。パケットが従う主な段階は、図４１に示すＲ２Ｔ書込データフローと基本的に同じである。

【００８１】

図４７は、セッション作成フローの主要な段階を示す。この図は、イニシエータとターゲットの間のセッション開始で要求される、この低速パス作動のための制御プレーンプロセッサの使用を表す。この機能性は、パケットプロセッサ複合システムを通して実施することが可能である。しかし、ここでは、制御プレーンプロセッサを使用して実施されるものとして示されている。どちらの手法も許容することができる。以下は、セッション作成中の主要な段階である。

１．コマンドが、ホストドライバによってスケジュールに入れられる。

２．ホストドライバは、コマンドがスケジュールに入れられ、ホストによって要求された全ての制御情報が通されることを通知される。

３．ストレージフロー／ＲＤＭＡコントローラは、セッションが存在しないターゲットにコマンドを送る要求を検出し、従って、それは、その要求を制御プレーンプロセッサに通して移送セッションを確立する。

４．制御プレーンプロセッサは、出力待ち行列に「ＴＣＰ　ＳＹＮ」パケットを送る。

５．ＳＹＮパケットは、ネットワークメディアインタフェースに送信され、そこから宛先に送信される。

６．宛先は、ＳＹＮパケットを受け取った後に、ＳＹＮ－ＡＣＫ応答を使用して応答し、そのパケットは、ネットワークメディアインタフェースから受け取り次第入力待ち行列に入れられる。

７．パケットが、パケットスケジューラによって検索される。

８．パケットが、分類エンジンに通される。

９．タグ付けされて分類されたパケットが、スケジューラに戻される。

１０．スケジューラは、分類に基づいてこのパケットを制御プレーンプロセッサに転送する。

１１．プロセッサは、次に、ＡＣＫパケットを使用して出力待ち行列に応答する。

１２．パケットは、次に、最終の宛先に送信され、従って、セッション確立ハンドシェークを終了する。

１３．セッションが確立された状態で、この状態は、ストレージフローコントローラに供給される。セッションエントリは、このように作り出され、次に、セッションメモリコントローラに通される（この部分は、図に示されていない）。

セッションを段階１３のような確立状態にする前に、制御プレーンプロセッサは、ストレージプロトコルの完全なログイン段階を実行し、パラメータを交換し、それがストレージデータ転送接続の場合は、それらを特定接続のために記録するように要求されることがある。ログインが認証されてパラメータ変換が終了した状態で、セッションは、上述の段階１３で示されたセッション確立状態に入る。

【００８２】

図４８は、セッション取外しフローの主要な段階を示す。このフローの段階は、図４７の段階と非常に似ている。２つのフローの主な違いは、セッション作成のためのＳＹＮ、ＳＹＮ－ＡＣＫ、及びＡＣＫパケットの代わりに、イニシエータとターゲットの間で、ＦＩＮ、ＦＩＮ－ＡＣＫ、及びＡＣＫパケットが転送されることである。それ以外は、主要段階は、非常に類似している。ここでの別の主な相違点は、適切なセッションエントリが

作り出されず、セッションキャッシュとセッションメモリから除去されることである。接続の作動統計データは、図に示されていないが、記録されてホストドライバに供給することができる。

【００８３】

図４９は、ターゲットの観点からのセッション作成とセッション取外しの段階を示す。以下は、セッション作成のために従うべき段階である。

１．イニシエータからのＳＹＮ要求が、ネットワークメディアインタフェース上で受信される。

２．スケジューラは、入力待ち行列からＳＹＮパケットを検索する。

３．スケジューラは、分類のためにこのパケットを分類エンジンに送る。

４．分類エンジンは、適切なタグを付けて分類されたパケットを戻す。

５．スケジューラは、ＳＹＮパケットとしての分類に基づいて、このパケットを制御プレーンプロセッサに転送する。

６．制御プレーンプロセッサは、ＳＹＮ－ＡＣＫ肯定応答パケットを使用して応答する。それはまた、イニシエータからの非送信請求データ転送のために適切なバッファスペースを割り当てるようにホストに要求する（この部分は、図示されていない）。

７．ＳＹＮ－ＡＣＫパケットが、イニシエータに送られる。

８．イニシエータは、次に、ＡＣＫパケットを使用してＳＹＮ－ＡＣＫパケットに肯定応答し、３方向ハンドシェークを終了する。このパケットは、ネットワークメディアインタフェースで受信され、第２層処理の後で入力待ち行列に入れられる。

９．スケジューラは、このパケットを検索する。

１０．パケットが、分類子に送られる。

１１．分類されたパケットは、スケジューラに戻され、３方向ハンドシェークを終了するために、制御プロセッサに提供されるべくスケジュールに入れられる。

１２．コントローラは、ＡＣＫパケットを取得する。

１３．制御プレーンプロセッサは、この時点で確立状態での接続を有し、セッションキャッシュ内にエントリを作り出すストレージフローコントローラに送る。

１４．ホストドライバは、終了したセッション作成を通知される。

セッション確立はまた、図４９に示されていないが、ログイン段階を伴う場合がある。しかし、ログイン段階とパラメータ変換は、完全に構成されて確立された状態にセッションが入る前に起きる。これらのデータ転送とハンドシェークは、主として制御プロセッサによって行うことができる。これらの段階が行われた状態で、上述のフローの残りの段階を実行することができる。

【００８４】

図５０及び５１は、ターゲットサブシステムの書込データフローを示す。図５０は、イニシエータからのデータ書込を受け入れる準備が完了していることをイニシエータに通知するためにターゲットによって使用されるＲ２Ｔコマンドフローを示す。イニシエータは、次に、ターゲットで受け取った書込を送り、その内部データフローは、図５１に示されている。２つの図は、両方で１つのＲ２Ｔ及びデータ書込の対を示す。以下は、図４０及び５１の両方で示すように追従される主な段階である。

１．ターゲットホストシステムは、図３３に示すような書込要求の受信に応答して、書込データを受け入れるために適切なバッファを準備し、ストレージフローコントローラに準備完了したらそれを通知し、転送待機中要求をイニシエータに送る。

２．フローコントローラは、ホストドライバに対して、要求とＤＭＡのためのバッファポインタとの受領を肯定応答する。

３．フローコントローラは、次に、実行されるＲ２Ｔコマンドをスケジューラに対してスケジュールに入れる。

４．スケジューラは、コマンドをこのコマンドを実行する準備が完了しているパケットプロセッサ複合システムの１つに出す。

５．パケットプロセッサは、セッションキャッシュコントローラからセッションエント

リを要求する。
６．セッションエントリが、パケットプロセッサに戻される。
７．パケットプロセッサは、ＴＣＰパケットを形成し、Ｒ２Ｔコマンドをカプセル化し、出力待ち行列にそれを送る。
８．パケットは、次に、ネットワークメディアインタフェースに送られ、それは、次に、イニシエータにこのパケットを送る。転送が安全転送である必要がある場合は、セキュリティエンジンを伴うことができるであろう。
９．次に、図５１に示すように、イニシエータは、書込データをターゲットに送ることによってＲ２Ｔに応答する。ネットワークメディアインタフェースは、パケットを受け取り、入力待ち行列にそれを入れる。
１０．パケットスケジューラは、入力待ち行列からパケットを検索する。
１１．パケットは、分類エンジンに対してスケジュールに入れられる。
１２．分類エンジンは、分類タグと共に分類されたパケットをスケジューラに供給する。図示のフローは、暗号化されていないパケットのためであり、従って、セキュリティエンジンは実行されない。
１３．スケジューラは、パケットプロセッサ待ち行列に対するフローベースのリソース割当待ち行列に基づいてパケットを割り当てる。パケットは、次に、パケットプロセッサがこのパケットの実行を待機中の時に、パケットプロセッサ復号システムに転送される。
１４．パケットプロセッサは、セッションキャッシュエントリを要求する（それがそのローカルキャッシュにまだそれを有していない場合）。
１５．セッションエントリが、要求するパケットプロセッサに戻される。
１６．パケットプロセッサは、全てのＴＣＰ／ＩＰ機能を実行し、セッションエントリを更新し、ストレージエンジンは、以前のＲ２Ｔに応答して書込コマンドとしてＰＤＵを抽出する。それは、ストレージセッションエントリを更新し、パケットをそれがホストバッファに転送されるようにホスト出力待ち行列にルーティングする。パケットは、ホスト割当の宛先バッファ内へのこのパケットのＤＭＡの実行に使用することができるメモリ記述子又はメモリ記述子リストを使用してタグ付けすることができる。
１７．ホストインタフェースブロックは、ＤＭＡを実行し、書込データコマンドのこのセグメントを終了する。

【００８５】

図５２は、ターゲット読取データフローを示す。このフローは、図４１に示すイニシエータＲ２Ｔ及び書込データフローに非常に似ている。このフローで従うべき主な段階は、次の通りである。
１．ネットワークメディアインタフェースブロックから入力パケットが受信される。
２．パケットスケジューラは、入力待ち行列からパケットを検索する。
３．分類のためにパケットがスケジュールに入れられる。
４．分類されたパケットが、分類タグを付けて分類子から戻る。
ａ．分類とフローベースのリソース割当とにより、パケットは、パケット上で作動するパケットプロセッサ複合システムに割り当てられる。
５．パケットプロセッサ複合システムは、セッションキャッシュ内のセッションエントリを検索する（ローカルに存在しない場合）。
６．セッションキャッシュエントリが、パケットプロセッサ複合システムに戻される。
７．パケットプロセッサは、読取コマンドＰＤＵを判断し、フローコントローラに対する要求を用いて読取データを要求する。
８．フローコントローラは、ホストインタフェースに対してＤＭＡを開始する。
９．ホストインタフェースは、ＤＭＡを実行し、ホスト入力待ち行列にデータを戻す。
１０．パケットプロセッサ複合システムは、ホスト入力待ち行列からデータを受け取る。
１１．パケットプロセッサ複合システムは、データの周囲に有効なＰＤＵとパケットを形成し、適切なセッションエントリを更新し、出力待ち行列にパケットを転送する。

１２．パケットは、宛先にデータパケットを送信する出力ネットワークメディアインタフェースブロックに転送される。

【００８６】

上述のフローの説明は、高帯域幅データ転送に伴ういくつかの主なフローの一例である。図示していないが本発明のＩＰプロセッサによってサポートされるフラグメント化データフロー、複数の異なる種類のエラーを有するエラーフロー、ネーム解決サービスフロー、アドレス解決フロー、及び、ログイン及びログアウトフローなどのようないくつかのフローが存在する。

本発明のＩＰプロセッサは、限定する意味でなく以下に概説する製造工程を使用して、様々な実現可能な実施形態の選択されたものにおけるハードウエア製品に製造することができる。プロセッサは、特定ターゲット製造工程技術に対する機能性、タイミング、及び他の設計及び製造制約に対するＲＴＬレベル、回路／回路図／ゲートレベル、レイアウトレベルなどのような様々なレベルのチップ設計抽象概念で設計及び検証することができる。適切な物理的／レイアウトレベルでのプロセッサ設計を用いて、ターゲット処理技術においてチップを製造するのに使用されるマスクセットを作り出すことができる。次に、マスクセットを用いて、選択された処理技術に対して使用される段階を通してプロセッサチップが構築される。プロセッサチップは、次に、製造されたプロセッサ製品の品質を保証するのに適切な場合は、試験／包装工程を通過することができる。

上述の事項は本発明の特定の実施形態に関連したものであるが、当業者は、本発明の原理及び精神から逸脱することなく、これらの実施形態に変更を行うことができることを認めるであろう。

【図面の簡単な説明】

【００８７】

【図１】階層化されたＳＣＳＩアーキテクチャと、イニシエータとターゲットシステムの間に位置するそれぞれの層の間の対話とを示す図である。

【図２】イニシエータとターゲットシステムの間のｉＳＣＳＩ及びＴＣＰ／ＩＰベースの移送を有する階層化されたＳＣＳＩアーキテクチャを示す図である。

【図３】ファイバチャンネルのようなハードウエア志向プロトコルを有するソフトウエアベースＴＣＰ／ＩＰスタックのＯＳＩスタック比較を示す図である。

【図４】他の非ＩＰハードウエア志向プロトコルとの性能パリティを提供するためのハードウエアベースＴＣＰ／ＩＰ実施によるＯＳＩスタックを示す図である。

【図５】ネットワーキング及びストレージスタックを実行するオペレーティングシステム層を表すホストソフトウエアスタックを示す図である。

【図６】ソフトウエアＴＣＰスタックデータ転送を示す図である。

【図７】本特許に説明するようなホストプロセッサからのＴＣＰ／ＩＰオフロードを使用する遠隔の直接メモリアクセスデータ転送を示す図である。

【図８】ＩＰネットワーク上でブロックストレージデータを移送するためのホストソフトウエアＳＣＳＩストレージスタック層を示す図である。

【図９】本発明の実施形態の特定のｉＳＣＳＩストレージネットワーク層スタックの詳細を示す図である。

【図１０】本発明の実施形態のＴＣＰ／ＩＰネットワークスタックの機能的詳細を示す図である。

【図１１】本発明の実施形態の様々な要素を通じたｉＳＣＳＩストレージデータフローを示す図である。

【図１２】本発明に有用なｉＳＣＳＩストレージデータ構造を示す図である。

【図１３】本発明の実施形態に有用なセッションデータベースエントリのためのＴＣＰ／ＩＰ伝送制御ブロックデータ構造を示す図である。

【図１４】本発明の実施形態に有用なｉＳＣＳＩセッションデータベース構造を示す図である。

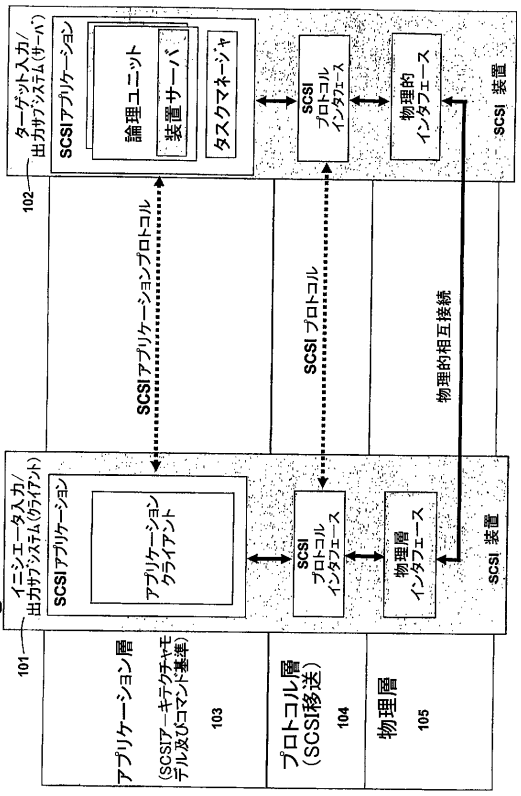
【図１５】本発明の実施形態に有用なｉＳＣＳＩセッションメモリ構造を示す図である。

【図１６】本発明の実施形態に有用なＩＰネットワークアプリケーションプロセッサの高レベルアーキテクチャブロック図である。
【図１７】図１６のＩＰネットワークアプリケーションプロセッサのアーキテクチャブロック図の詳細図である。
【図１８】ＩＰプロセッサの一実施形態の入力待ち行列とコントローラを示す図である。
【図１９】ＩＰプロセッサの一実施形態に有用なパケットスケジューラ、シーケンサ、及び負荷バランサを示す図である。
【図２０】ＩＰストレージプロセッサの一実施形態のポリシーエンジンブロックを含むパケット分類エンジンを示す図である。
【図２１】高レベルでのＩＰプロセッサの一実施形態のＳＡＮパケットプロセッサブロックの実施形態を概略で示す図である。
【図２２】説明したＩＰプロセッサのＳＡＮパケットプロセッサブロックの実施形態をより詳細に示す図である。
【図２３】説明したＳＡＮパケットプロセッサの一部として使用することができるプログラマブルＴＣＰ／ＩＰプロセッサエンジンの実施形態を示す図である。
【図２４】説明したＳＡＮパケットプロセッサの一部として使用することができるプログラマブルＩＰストレージプロセッサエンジンの実施形態を示す図である。
【図２５】図１７のプログラマブルＩＰプロセッサの出力待ち行列ブロックの実施形態を示す図である。
【図２６】ストレージフローコントローラ及びＲＤＭＡコントローラの実施形態を示す図である。
【図２７】本発明の実施形態に有用なＩＰプロセッサのホストインタフェースコントローラブロックの実施形態を示す図である。
【図２８】セキュリティエンジンの実施形態を示す図である。
【図２９】説明したプロセッサに有用なメモリ及びコントローラの実施形態を示す図である。
【図３０】説明した分類エンジンの実施形態に利用可能なデータ構造を示す図である。
【図３１】イニシエータとターゲットの間のストレージ読取フローを示す図である。
【図３２】説明したプロセッサのパイプラインステージを通じた読取データパケットフローを示す図である。
【図３３】イニシエータとターゲットの間のストレージ書込作動フローを示す図である。
【図３４】説明したプロセッサのパイプラインステージを通じた書込データパケットフローを示す図である。
【図３５】イニシエータとターゲットの間の遠隔ＤＭＡ（ＲＤＭＡ）機能を使用したイニシエータとターゲットの間のストレージ読取フローを示す図である。
【図３６】説明したプロセッサのパイプラインステージを通じてＲＤＭＡを使用したイニシエータとターゲットの間の読取データパケットフローを示す図である。
【図３７】ＲＤＭＡ機能を使用したイニシエータとターゲットの間のストレージ書込フローを示す図である。
【図３８】説明したプロセッサのパイプラインステージを通じてＲＤＭＡを使用した書込データパケットフローを示す図である。
【図３９】説明したプロセッサのパイプラインステージを通じたイニシエータコマンドフローをより詳細に示す図である。
【図４０】説明したプロセッサのパイプラインステージを通じた読取パケットデータフローをより詳細に示す図である。
【図４１】説明したプロセッサのパイプラインステージを通じた書込データフローをより詳細に示す図である。
【図４２】パケットがチップセットテキストか又はそうでなければ説明されたプロセッサのパイプラインステージを通じた安全なパケットの時の読取データパケットフローを示す図である。

【図４３】パケットがチップセットテキストか又はそうでなければ説明されたプロセッサのパイプラインステージを通じた安全なパケットの時の書込データパケットフローを示す図である。
【図４４】説明したプロセッサのパイプラインステージを通じたＲＤＭＡバッファ通知フローを示す図である。
【図４５】説明したプロセッサのパイプラインステージを通じたＲＤＭＡ書込フローをより詳細に示す図である。
【図４６】説明したプロセッサのパイプラインステージを通じたＲＤＭＡ読取データフローをより詳細に示す図である。
【図４７】説明したプロセッサのパイプラインステージを通じたセッション作成フローの段階を示す図である。
【図４８】説明したプロセッサのパイプラインステージを通じたセッション取外しフローの段階を示す図である。
【図４９】説明したプロセッサのパイプラインステージを通じてターゲットの展望で見たセッション作成及びセッション取外しの段階を示す図である。
【図５０】説明したプロセッサのパイプラインステージを通じたターゲットサブシステムのＲ２Ｔコマンドフローを示す図である。
【図５１】説明したプロセッサのパイプラインステージを通じたターゲットサブシステムの書込データフローを示す図である。
【図５２】説明したプロセッサのパイプラインステージを通じたターゲット読取データフローを示す図である。
【符号の説明】
【００８８】
１７０１　入力待ち行列
１７０２　パケットスケジューラ
１７０３　分類エンジン
１７０５　セキュリティエンジン
１７０６　ストレージエリアネットワーク（ＳＡＮ）パケットプロセッサ
１７０７　ホスト入力待ち行列
１７０９　ホスト出力待ち行列
１７１２　出力待ち行列

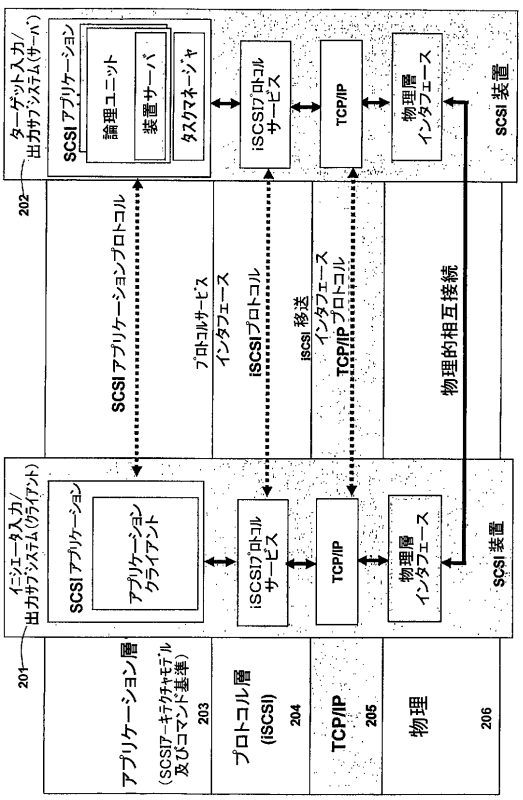
【図１】

Fig. 1 アーキテクチャ層



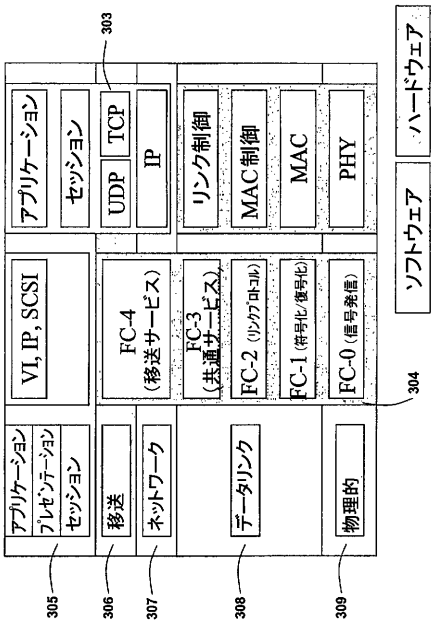
【図２】

Fig. 2 iSCSI上のSCSIアーキテクチャ層



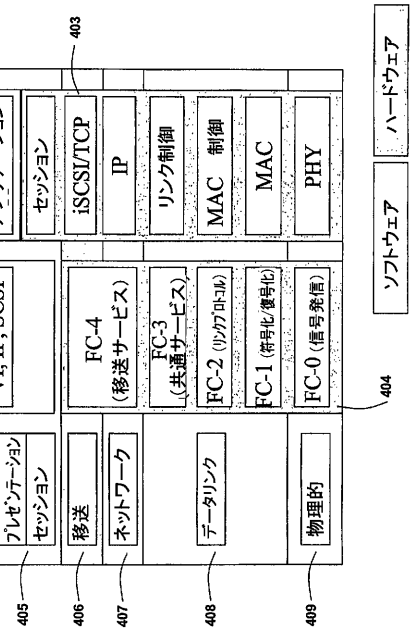
【図３】

Fig. 3 OSIスタック



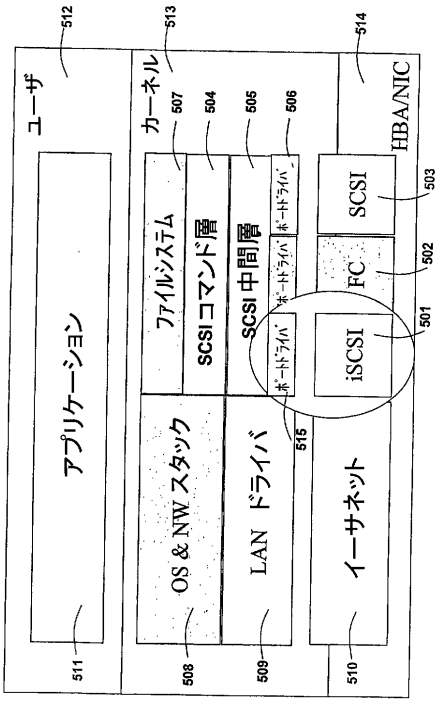
【図４】

Fig. 4 スタック(ハードウェアTCP/IP付き)



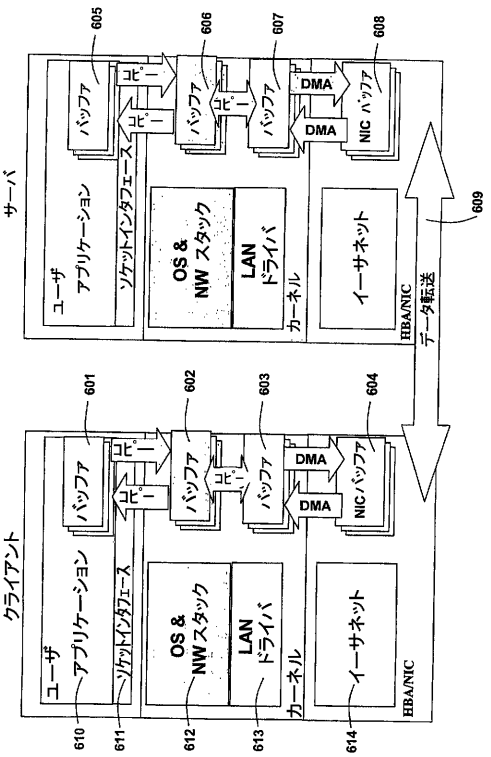
【図５】

Fig. 5 ホストソフトウエアスタック (iSCSI)



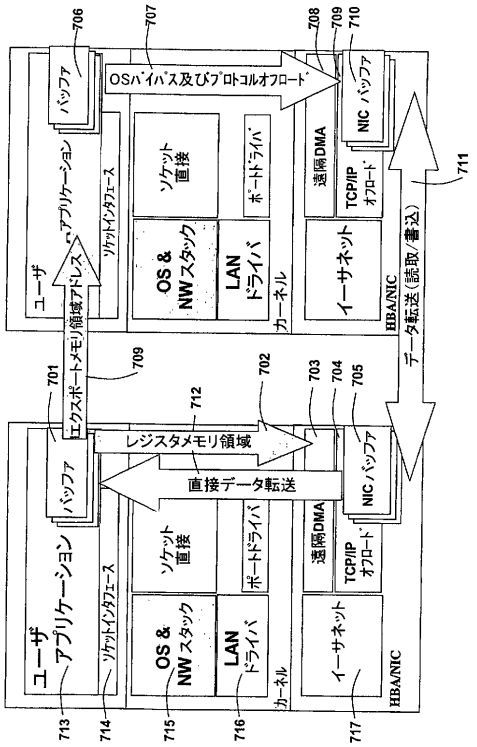
【図６】

Fig. 6 S/W TCP スタックデータ転送



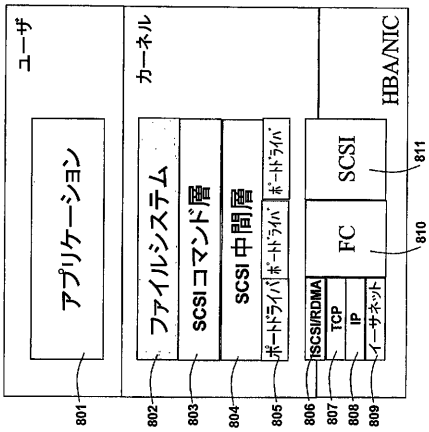
【図７】

Fig. 7 遠隔直接メモリアクセス



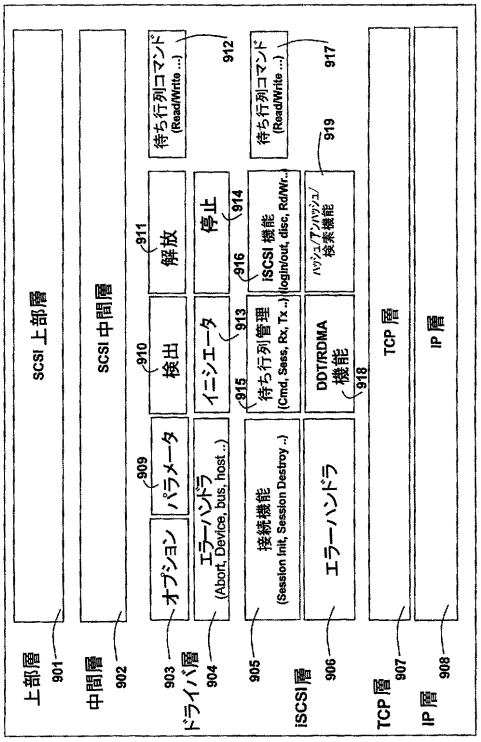
【図８】

Fig. 8 ホストソフトウエアスタック (SCSI)



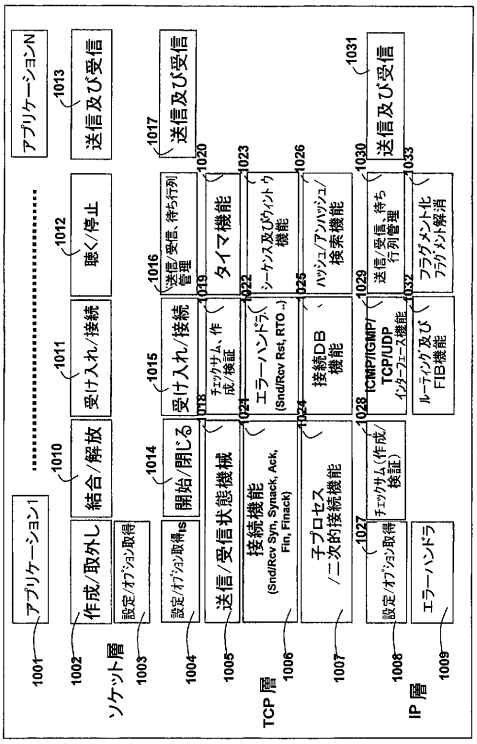
【図９】

Fig. 9 iSCSIスタック



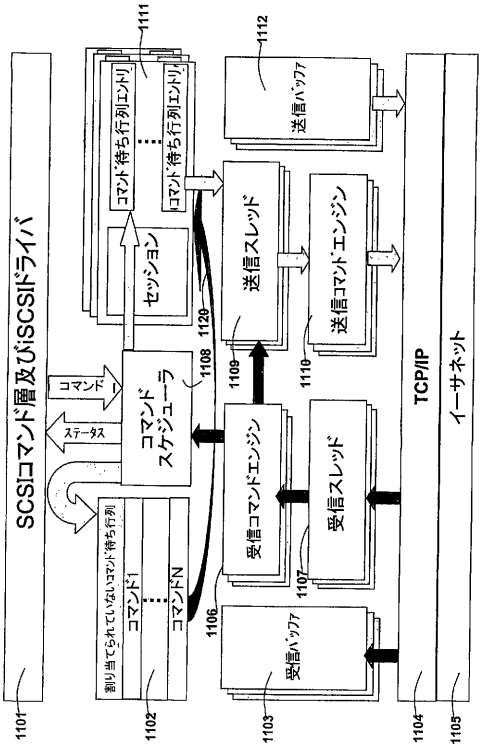
【図１０】

Fig. 10 TCP/IPスタック



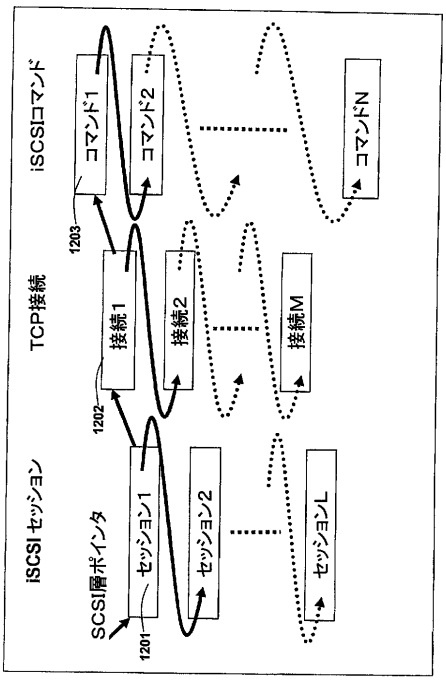
【図１１】

Fig. 11 iSCSI データフロー



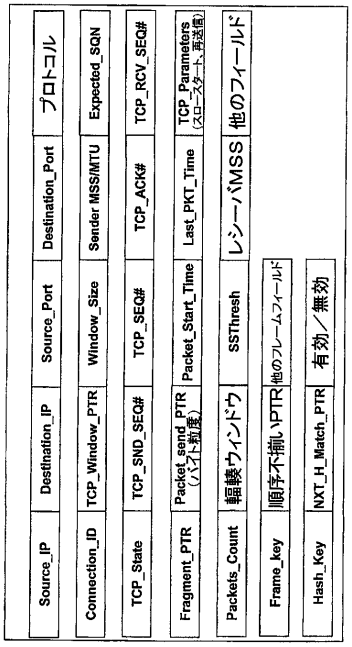
【図１２】

Fig. 12 iSCSIデータ構造



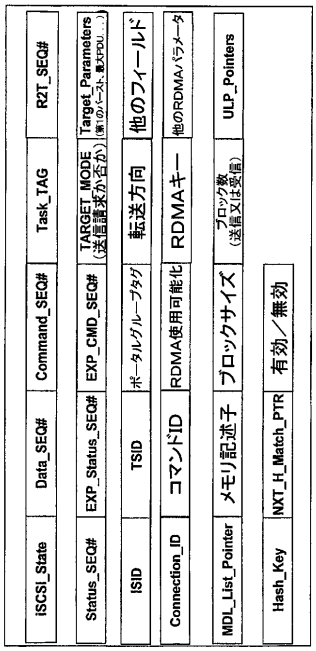
【図１３】

Fig. 13 TCP／IPセッションDBエントリ



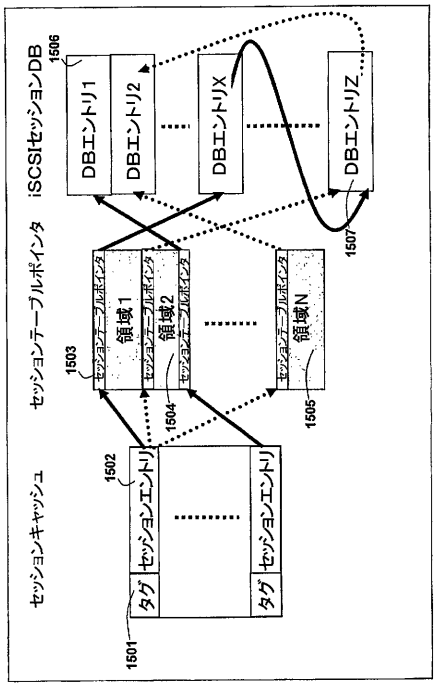
【図１４】

Fig. 14 iSCSIセッションDBエントリ



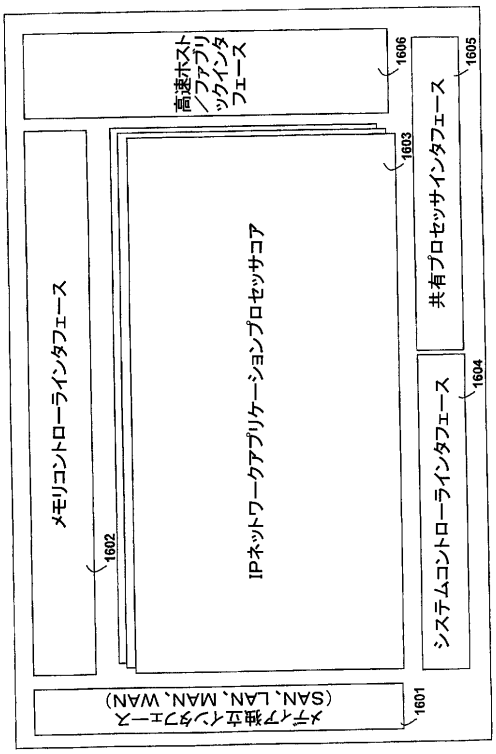
【図１５】

Fig. 15 iSCSIセッションメモリ



【図１６】

Fig. 16 IPネットワークアプリケーションプロセッサ



【図１７】

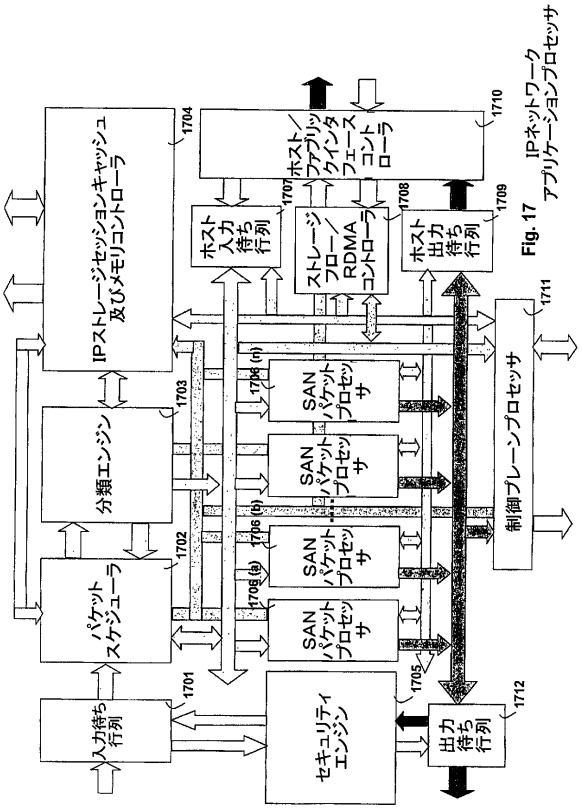


Fig. 17

【図１８】

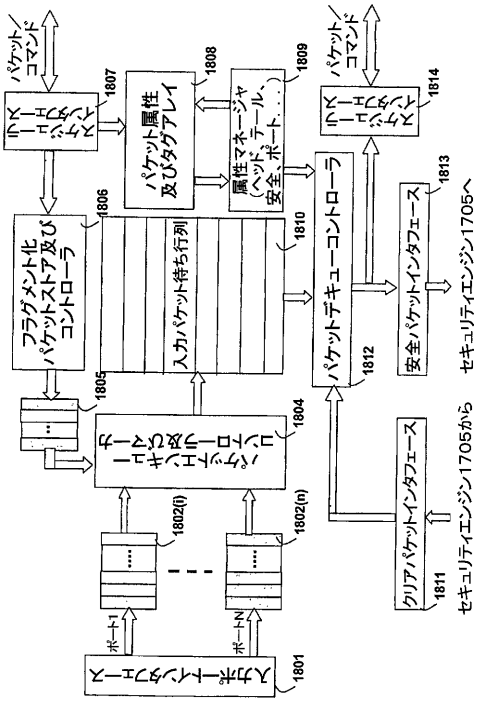


Fig. 18 入力待ち行列及びコントローラ

【図１９】

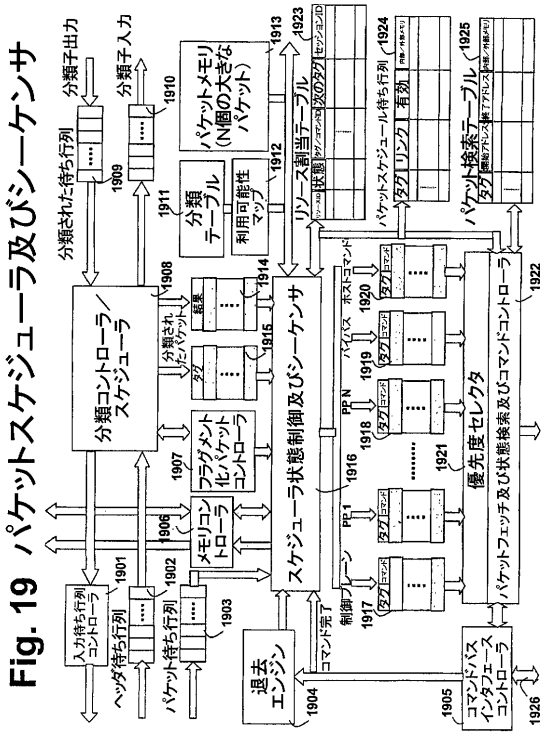


Fig. 19 パケットスケジューラ及びシーケンサ

【図２０】

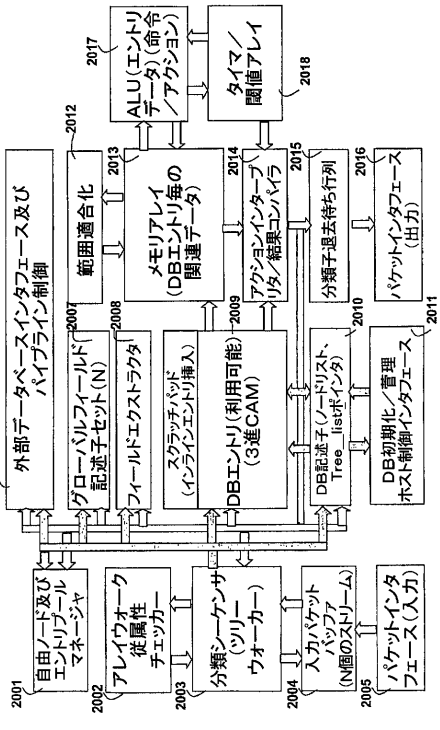
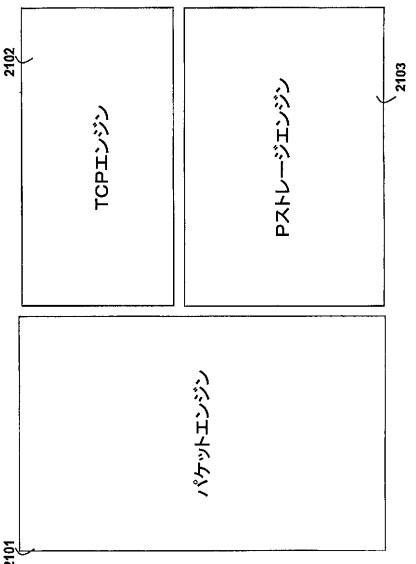


Fig. 20 パケット分類／ポリシーエンジン

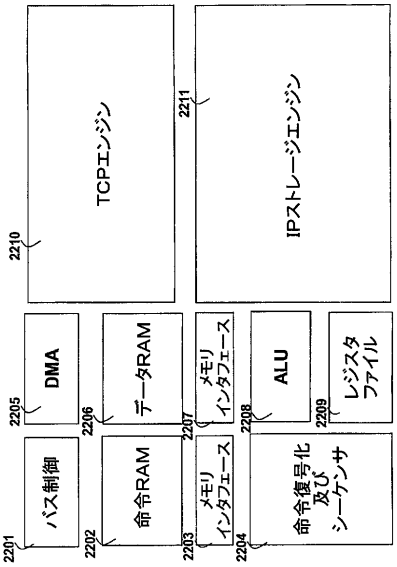
【図２１】

Fig. 21 SANパケットプロセッサ



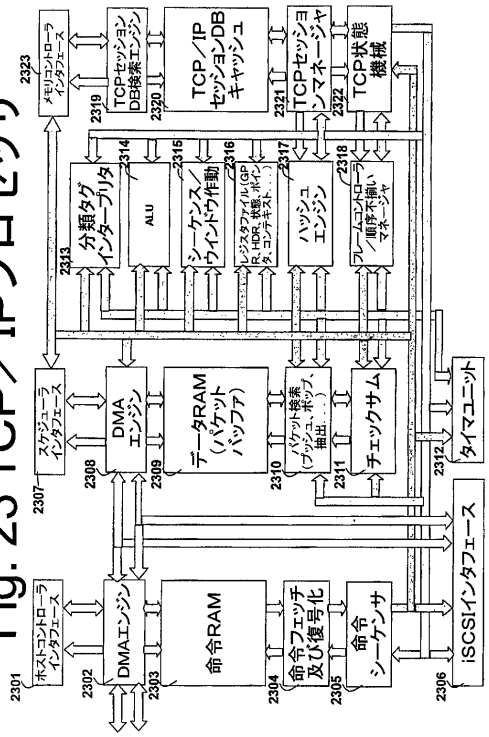
【図２２】

Fig. 22 SANパケットプロセッサ



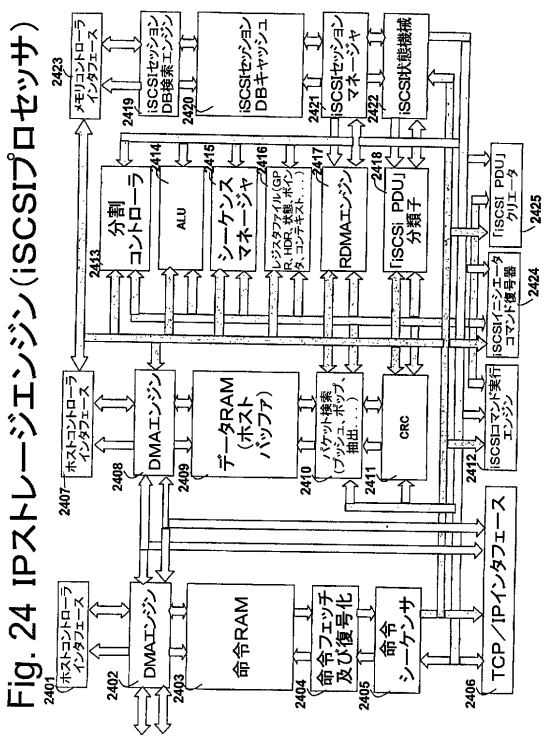
【図２３】

Fig. 23 TCP／IPプロセッサ



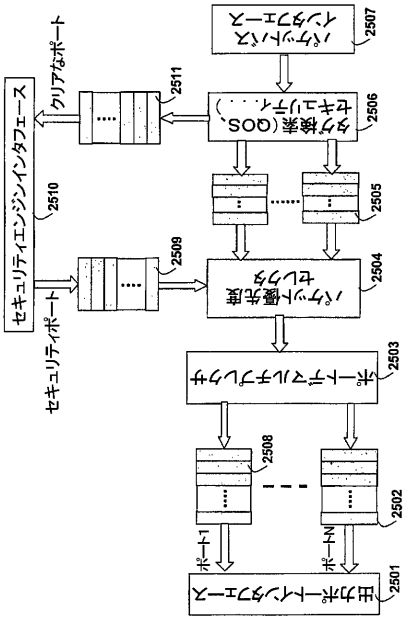
【図２４】

Fig. 24 IPストレージエンジン（iSCSIプロセッサ）



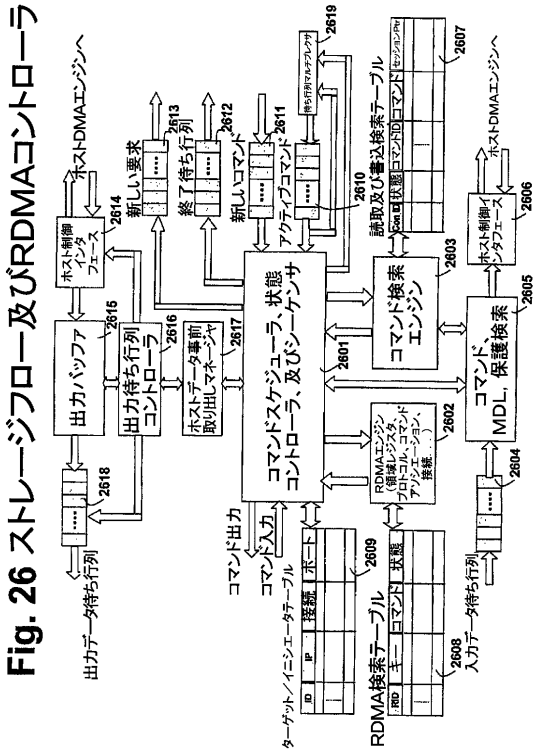
【図２５】

Fig. 25 出力待ち行列



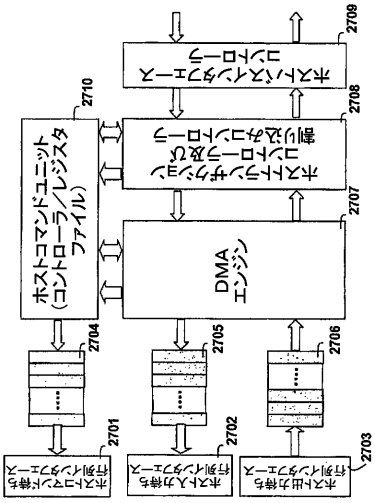
【図２６】

Fig. 26 ストレージフロー及びRDMAコントローラ



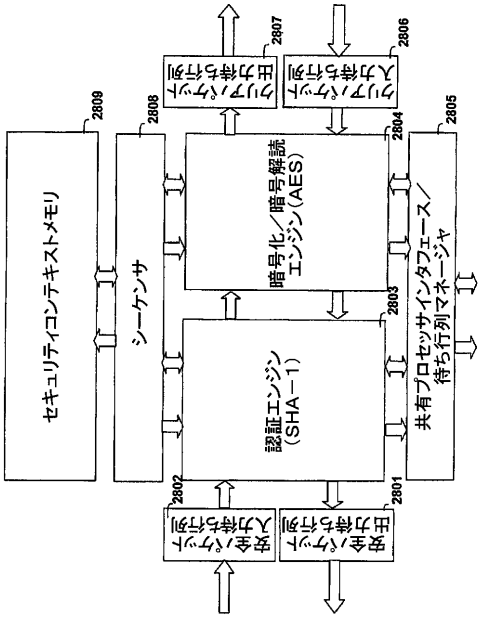
【図２７】

Fig. 27 ホストインタフェースコントローラ



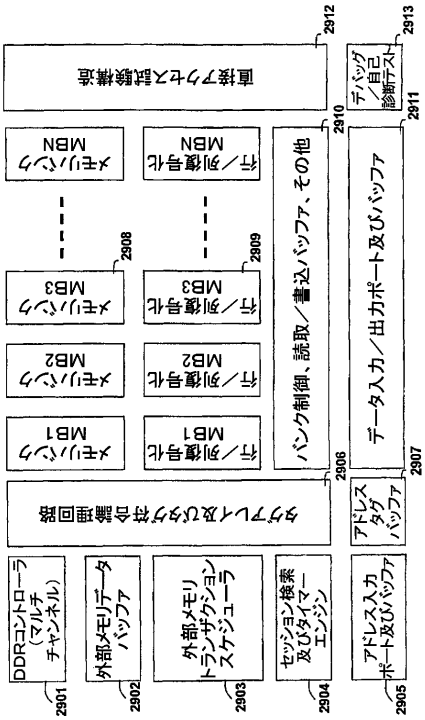
【図２８】

Fig. 28 セキュリティエンジン



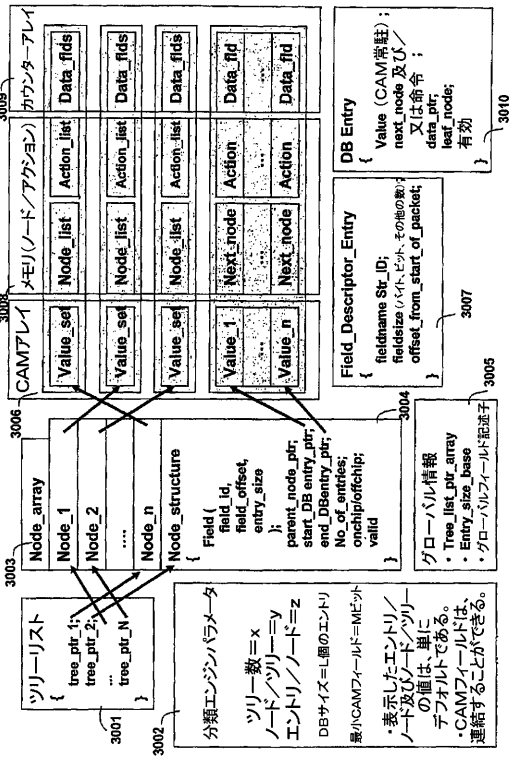
【図２９】

Fig. 29 IPストレージセッションキャッシュ及びメモリコントローラ複合システム



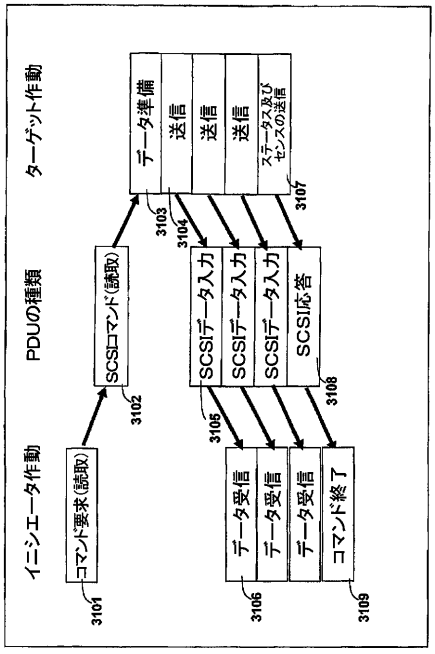
【図３０】

Fig. 30 分類エンジンデータ構造



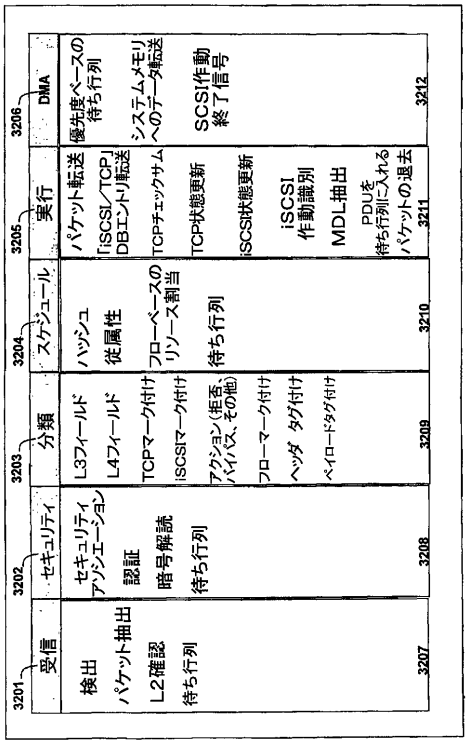
【図３１】

Fig. 31 読取作動



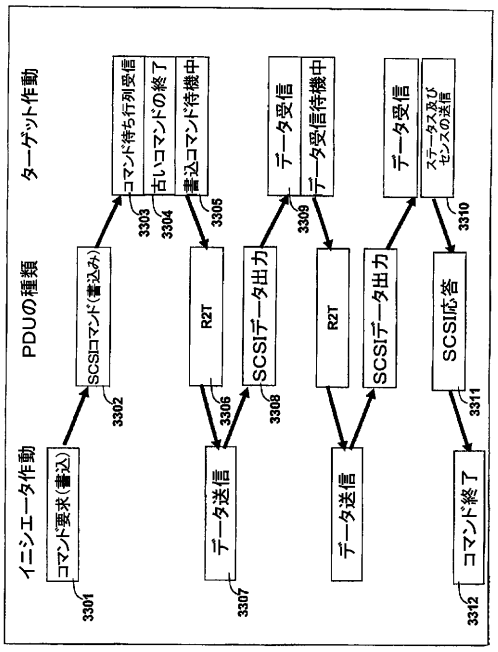
【図３２】

Fig. 32 読取データパケットフロー



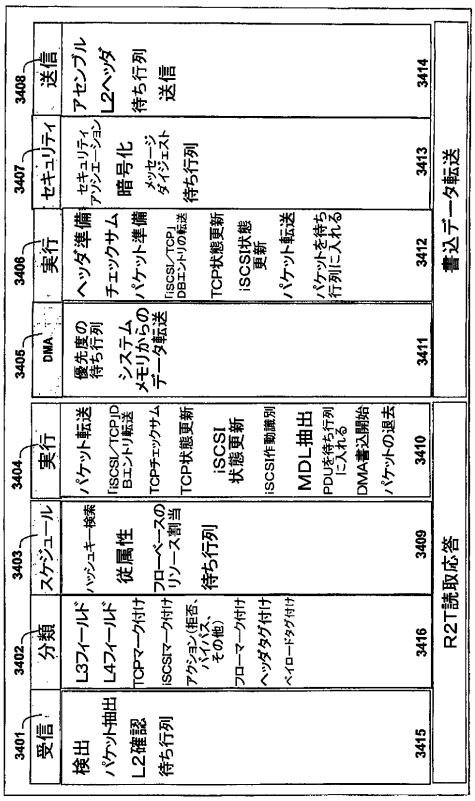
【図３３】

Fig. 33 書込作動



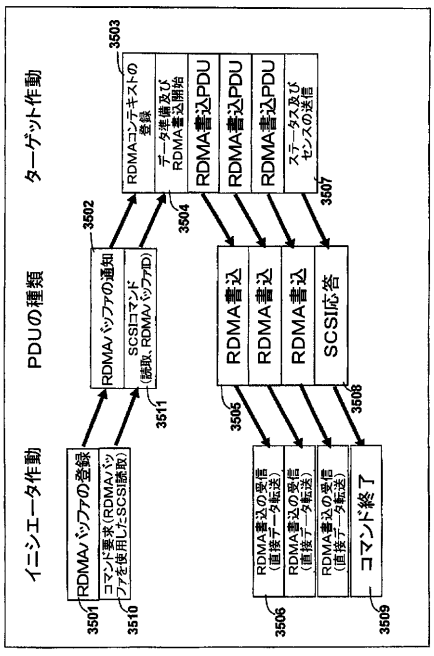
【図３４】

Fig. 34 書込データパケットフロー



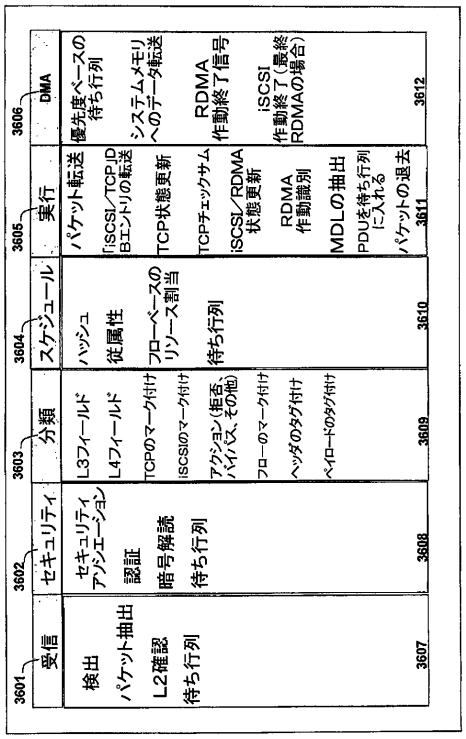
【図３５】

Fig. 35 RDMA書込を使用したiSCSI読取



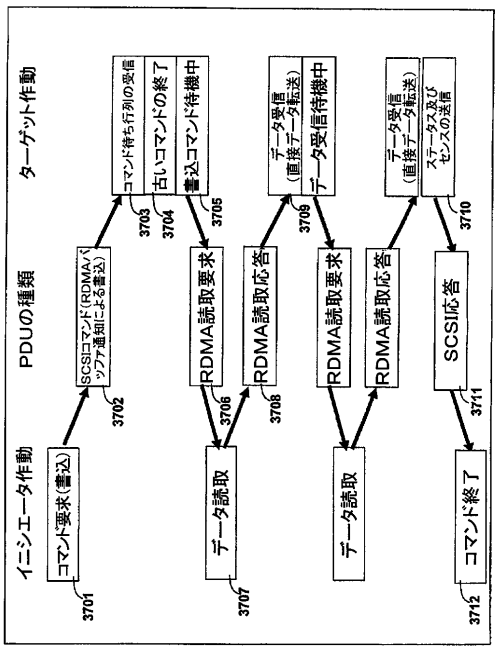
【図３６】

Fig. 36 読取データパケットフロー(RDMA)



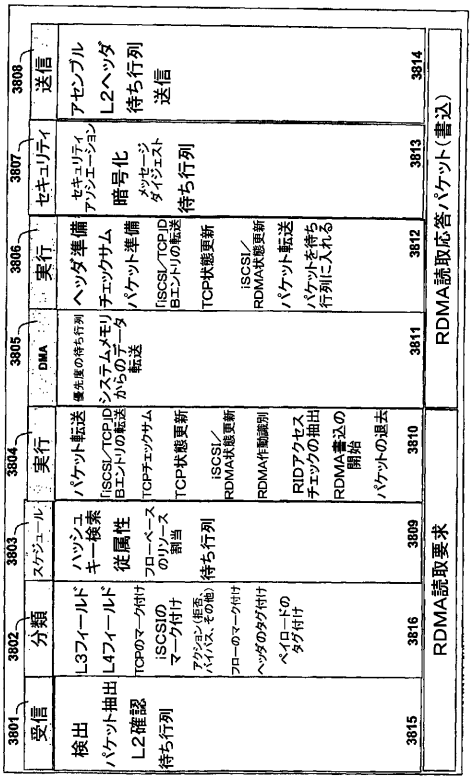
【図３７】

Fig. 37 RDMA読取を使用したiSCSI書込



【図３８】

Fig. 38 書込データパケットフロー(RDMA)



【図３９】

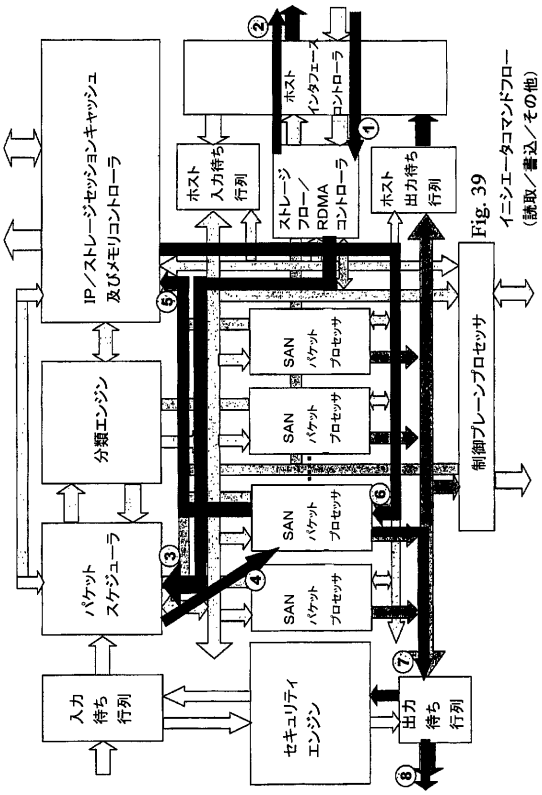


Fig. 39
イニシエータコマンドフロー
(読取/書込/その他)

【図４０】

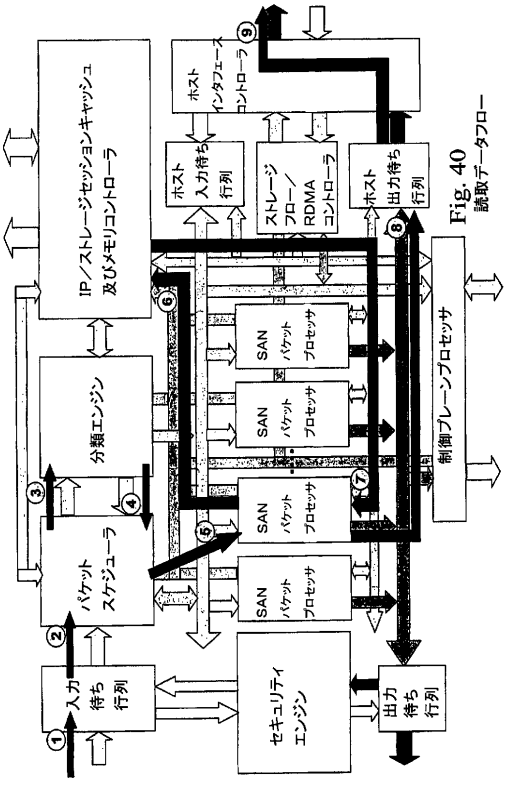


Fig. 40
読取データフロー

【図４１】

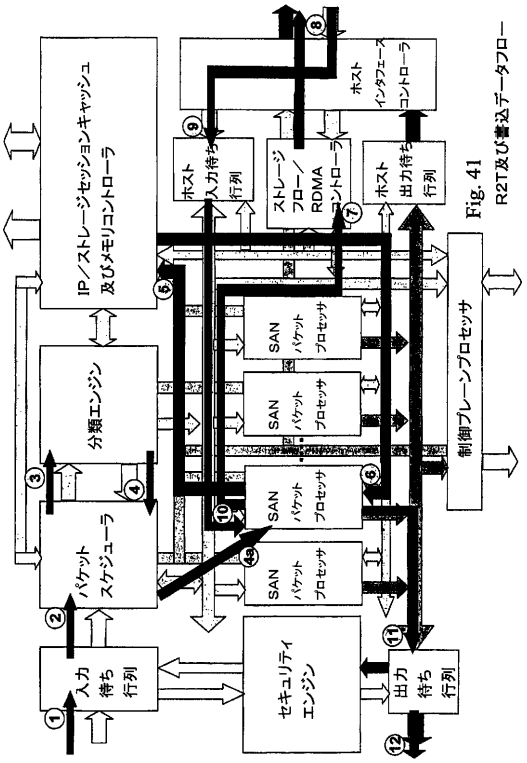


Fig. 41 R2T及び書込データフロー

【図４２】

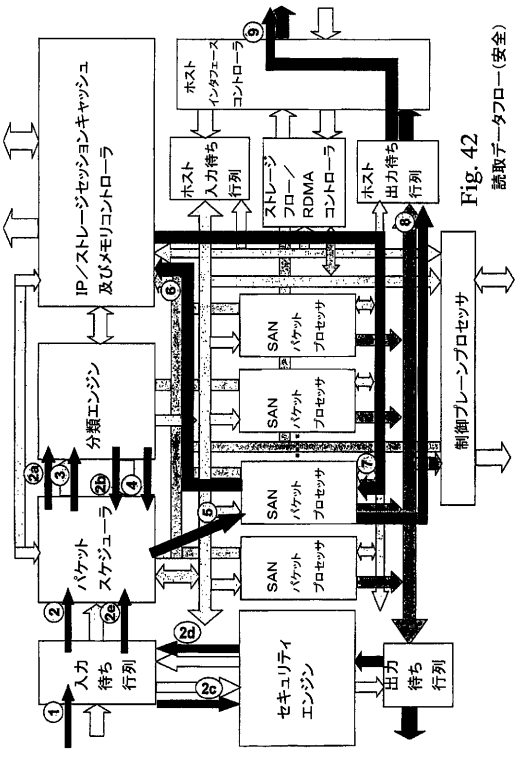


Fig. 42 読取データフロー（安全）

【図４３】

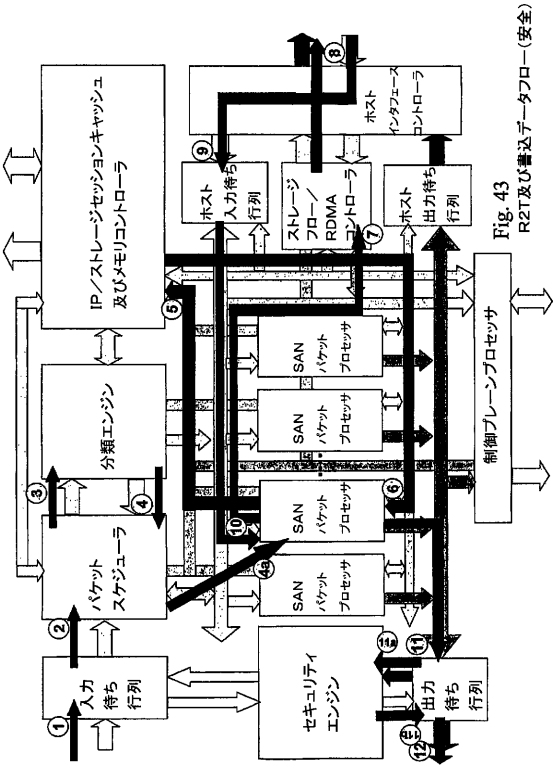


Fig. 43 R2T及び書込データフロー（安全）

【図４４】

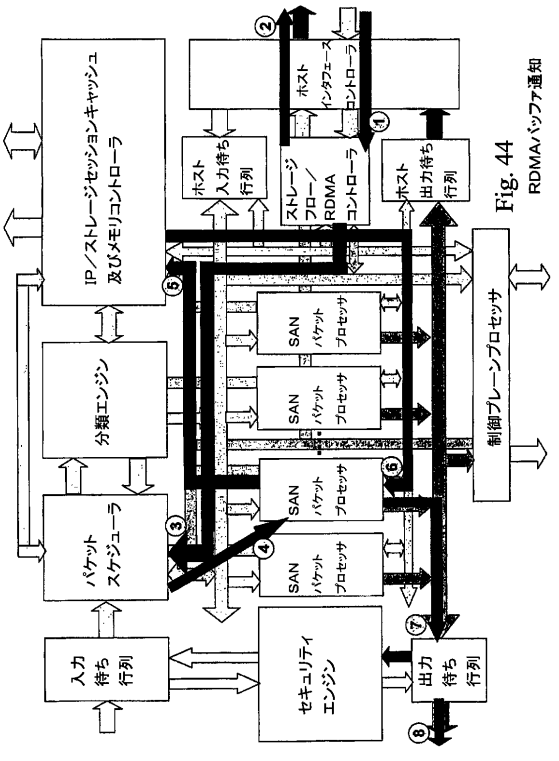


Fig. 44 RDMAバッファ通知

【図４５】

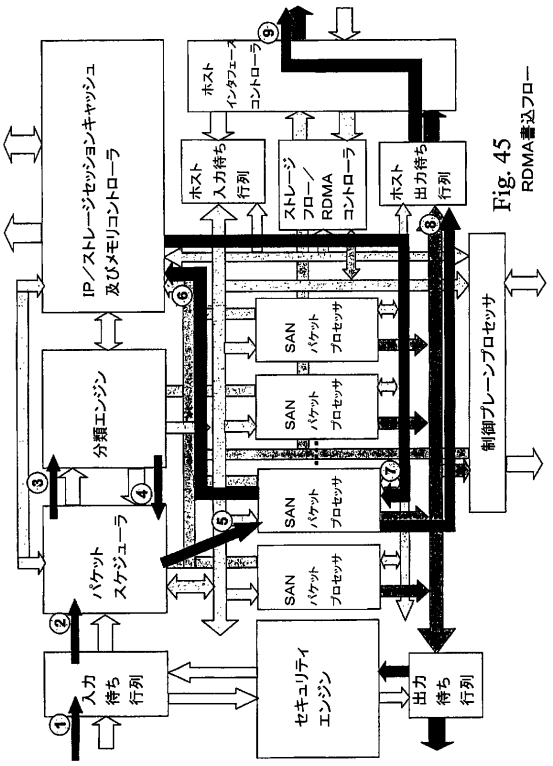


Fig. 45 RDMA書込フロー

【図４６】

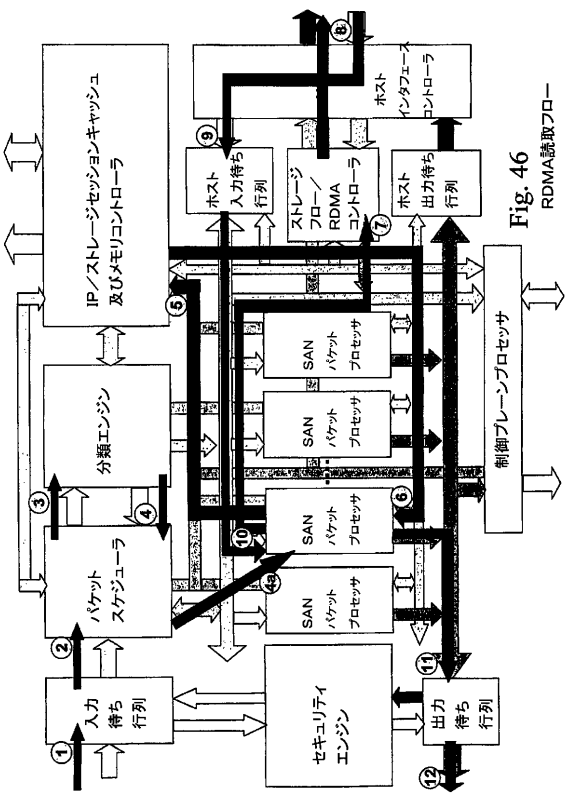


Fig. 46 RDMA読取フロー

【図４７】

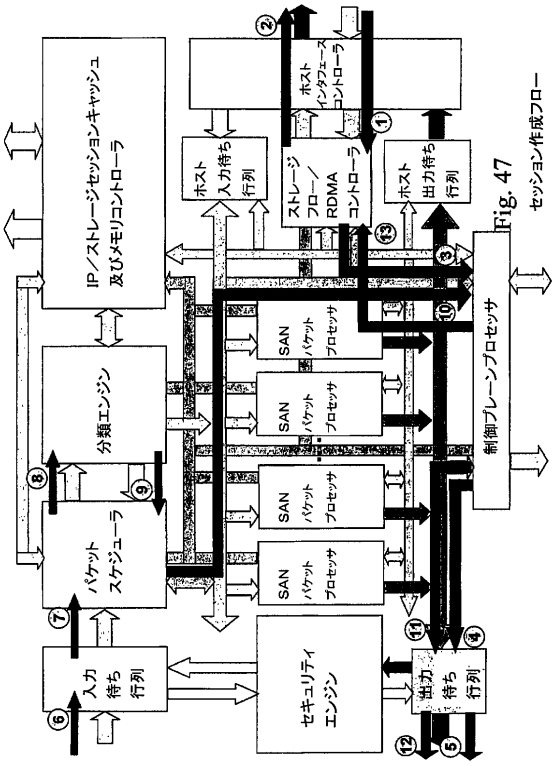


Fig. 47 セッション作成フロー

【図４８】

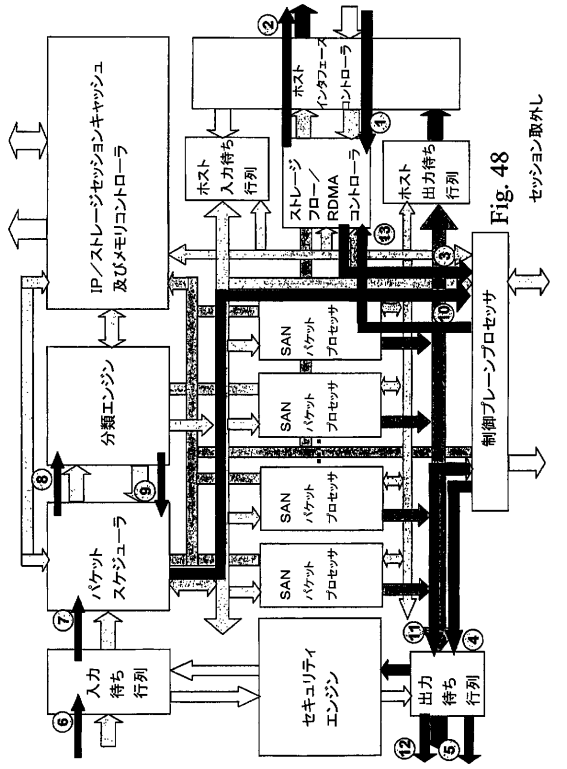


Fig. 48 セッション取外し

【図４９】

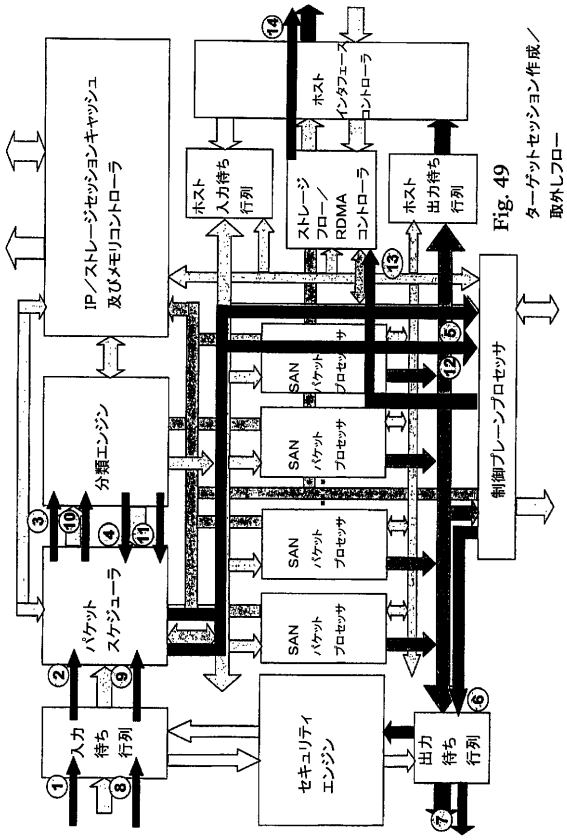


Fig. 49 ターゲットセッション作成／取外しフロー

【図５０】

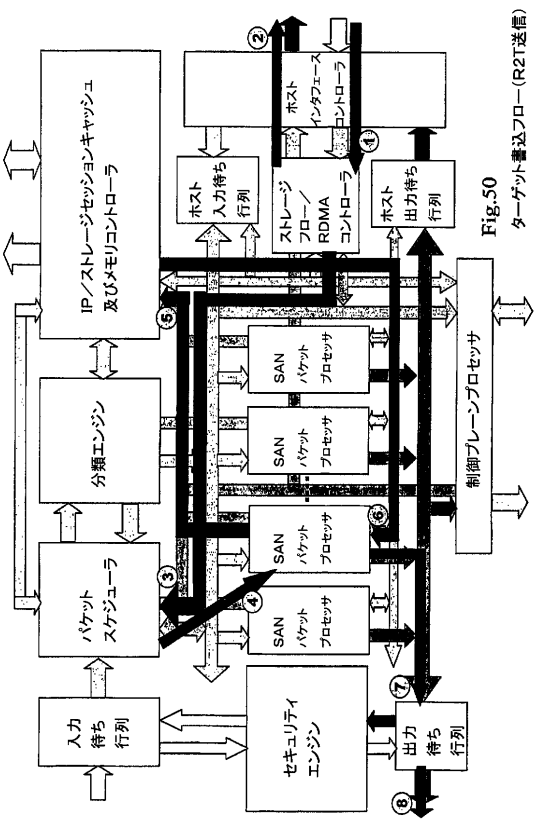


Fig.50 ターゲット書込フロー（R2T送信）

【図５１】

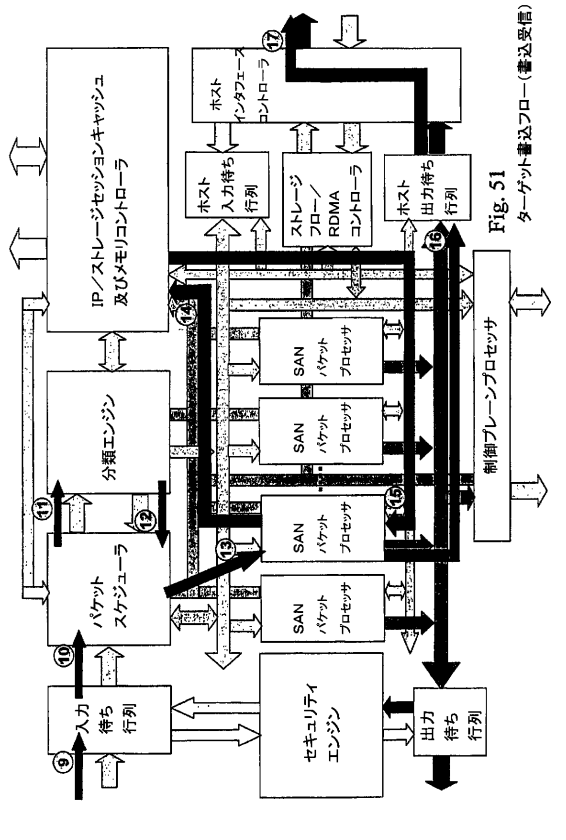


Fig. 51 ターゲット書込フロー（書込受信）

【図５２】

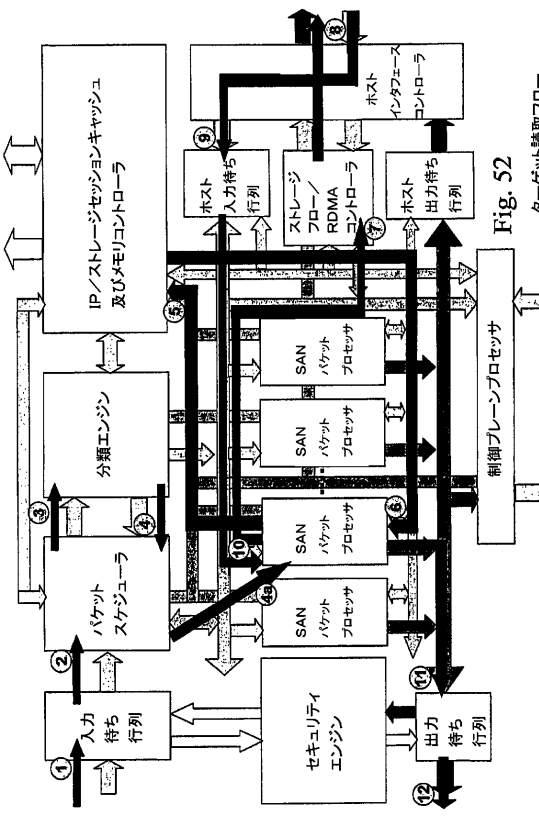


Fig. 52 ターゲット読取フロー

フロントページの続き

(72)発明者　パンドヤ　アシシュ　エイ
アメリカ合衆国　カリフォルニア州　９５７６２　エル　ドラド　ヒルズ　ラファイエット　ドライヴ　４３１８

審査官　横山　佳弘

(56)参考文献　特開２００２－０６３０６０（ＪＰ，Ａ）
特開２００１－２６８１５９（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　13/10
G06F　13/00
G06F　3/06