

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3926558号

(P3926558)

(45) 発行日 平成19年6月6日(2007.6.6)

(24) 登録日 平成19年3月9日(2007.3.9)

(51) Int. Cl.	F I
HO 4 L 12/56 (2006.01)	HO 4 L 12/56 B
GO 6 F 17/30 (2006.01)	GO 6 F 17/30 1 7 O Z
HO 4 L 12/28 (2006.01)	GO 6 F 17/30 4 1 3
	HO 4 L 12/28 2 O O A

請求項の数 2 (全 11 頁)

(21) 出願番号	特願2000-378532 (P2000-378532)	(73) 特許権者	500570195
(22) 出願日	平成12年12月13日(2000.12.13)		アセンド コミュニケーションズ インコ
(65) 公開番号	特開2001-223750 (P2001-223750A)		ーポレーテッド
(43) 公開日	平成13年8月17日(2001.8.17)		アメリカ合衆国 9 4 5 0 2 カリフォル
審査請求日	平成14年5月31日(2002.5.31)		ニア, アラミーダ, ハーバー ペイ パー
(31) 優先権主張番号	09/459441		クウェイ 1 7 0 1, アセンド プラザ
(32) 優先日	平成11年12月13日(1999.12.13)		1
(33) 優先権主張国	米国 (US)	(74) 代理人	100064447
前置審査			弁理士 岡部 正夫
		(74) 代理人	100085176
			弁理士 加藤 伸晃
		(74) 代理人	100096943
			弁理士 臼井 伸一
		(74) 代理人	100101498
			弁理士 越智 隆夫

最終頁に続く

(54) 【発明の名称】 ルートルックアップエンジン

(57) 【特許請求の範囲】

【請求項 1】

ルートルックアップエンジンの動作方法であって、

(a) タイプフラグ値、インダイレクトフラグ値、オーダデータ及びR L Eルートデータを含むリクエストブロックデータを含むルートルックアップリクエストを受取るステップ；

(b) 現オーダ値を「オーダデータ」にセットするステップ；

(c) タイプフラグを「葉」か否かについて評価するステップ；

(d) タイプフラグが「葉」を示していない場合にR L E D R A Mアドレスを増分し、タイプフラグが「葉」である場合にステップ (g) へ直接進むステップ；

(d 1) 増分された該R L E D R A Mアドレスに対応する次のブロックデータを読み取るステップであって、該次のブロックデータはタイプフラグ、インダイレクトフラグ、オーダデータ及びR L Eルートデータを含む、ステップ；

(d 2) 該次のブロックデータの該オーダデータに従って、該現オーダ値を増分するステップ；

(e) 増分された該現オーダ値を評価するステップ；

(e 1) 該増分された現オーダ値が6 4を越えている場合にステップ (i) へ進むステップ；

(e 2) 該増分された現オーダ値が3 2を越えている場合にLビットをセットするステップ；

10

20

(f) ステップ(c)に戻るステップ；

(g) 該インダイレクトフラグを評価するステップ；

(g 1) 該インダイレクトフラグが許可でない場合にステップ(j)へ直接進むステップ；

(g 2) 該インダイレクトフラグが許可の場合に該R L E D R A Mアドレスを該R L Eルートデータに等しくセットするステップ；

(g 3) 該R L E D R A Mアドレスに対応する次のブロックデータを読み取るステップであって、該次のブロックデータはタイプフラグ、インダイレクトフラグ、オーダデータ及びR L Eルートデータを含む、ステップ；

(h) 該インダイレクトフラグ及び該タイプフラグを評価するステップ；

10

(h 1) 該インダイレクトフラグが許可である場合、又は該タイプフラグが「ノード」を示している場合にステップ(i)へ直接進むステップ；

(h 2) 該インダイレクトフラグが許可でない場合、かつ該タイプフラグが「葉」を示している場合にステップ(j)へ直接進むステップ；

(i) 該方法を終了しエラーを報告するステップ；

(j) R L Eルートデータに対応するルートテーブルデータを記憶して該方法を終了するステップ

からなる方法。

【請求項 2】

ルートルックアップエンジンの動作方法であって、

20

ルートルックアップリクエストを受取るステップ、

ルートルックアップを実行するステップであって、

該ルックアップが該リクエストに対する最初のルックアップであるかの判断を行うステップ、

該ルックアップが該リクエストに対する最初のルックアップであるとき、該リクエストに対し現ブロックデータを使用し、該ルックアップが該リクエストに対して最初のルックアップでないとき、前回ブロックデータを使用するステップ、

該ブロックデータが「葉」を示しているかどうかの判断を行うステップであって、

該ブロックデータが「葉」を示している時、間接ビットが該ブロックデータに設定されているかどうかを判断し、該間接ビットが設定がされているとき、該ブロックデータからのルートテーブルルックアップエンジンルートから成るルートテーブルルックアップアドレスを供給し、該ルックアップテーブルから読取りを行い、該読取りは新規ブロックデータを供給し、該間接ビットが1に等しくなく、かつタイプビットがノードに等しくない場合に、結果を保存するこれらステップを実行するステップ；及び

30

該間接ビットが設定されていないとき、該結果を保存するステップを実行するステップからなる方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

データ転送においてルートルックアップを実行するルートルックアップエンジンに関する

40

【0002】

【従来の技術】

ルータのようなネットワークデバイスは処理動作のその一部としてルートルックアップを実行しなくてはならない。ルートルックアップは、次にパケットがどこに送られるべきかを判断するために、ルータにより受け取られる各パケットごとに必要となる。受信パケットは32ビット長の宛先アドレスを含む。宛先アドレスはパケットの転送がなされる固有のアドレスを識別する。ルートルックアップでは受信パケットの宛先アドレスを使用して、ルーティングテーブルにおける次ホップアドレスと出口ポートを識別する。宛先アドレスが32ビット長である場合、宛先可能数は10億以上となる。インターネットプロトコ

50

ル（IP）アドレスはネットワーク番号とホスト番号から構成されている。ネットワーク番号は可変長であり、固有のネットワークを識別する。ネットワーク番号に連結しているのがホスト番号であり、ホスト番号は特定のネットワーク内の固有のホストシステムを識別する。受信アドレスのネットワーク番号部分はルーティングテーブルにおける複数エントリーとマッチングが行われる。この最適マッチは、アドレスの最初の部分のビットが最大数マッチングするものである。

【0003】

【発明が解決しようとする課題】

しかしながら、この従来のルートルックアップにおいてマッチングを実行するために使用される従来技術による方法は複雑、かつ時間がかかる。よって、ルートルックアップを迅速、かつ効果的に実行するルートルックアップエンジンが必要とされる。

10

【0004】

【課題を解決するための手段】

上述した問題を解決するため、本発明は次ホップインデックスを判断するルートルックアップエンジンを開示する。ルートルックアップエンジンはルックアップキーを受け取り、プレフィックスエキスパンションと可変ストライド（stride）ツリーキャプチャによりマルチビット樹木探索を実行する。ルートルックアップエンジンが返送するデータは、次ホップ情報およびステータスフラグより構成される。ルートルックアップエンジンはコンパクトでフィールド再利用可能なデータ構造を使用する。ルートルックアップエンジンでは仮想プライベートネットワーク上にてユニキャストIPアドレスルックアップおよびマルチキャストIPアドレスルックアップの両方を実行する。ルートルックアップエンジンは別々のインデックスメモリーと転送メモリーを使用する。ルートルックアップエンジンのサーチタイムの上界はルートテーブルサイズに関係なく決定される。

20

【0005】

【発明の実施の形態】

ルートルックアップエンジンを開示する。ルートルックアップエンジンは多数アプリケーションにおいて使用が可能である。望ましい実施形態においては、ルートルックアップエンジンは非同期トランスファモード（ATM）スイッチのインターネットプロトコル（IP）ルーティングサブシステムの一部である。ATMスイッチのルーティングサブシステムは迅速かつ効果的なルートルックアップメカニズムを提供する。図1はATMスイッチのトップレベルブロック図である。ATMスイッチは入／出力アダプタ52を備えている。この入／出力アダプタはインターフェースオブティクス、同期光ネットワーク（SONET）フレーマ、およびハイレベルデータリンク制御手順（HDLC）コントローラを備えている。ATMスイッチはまた、入／出力モジュール10を備えている。入／出力モジュールは複数の転送エンジン20を備えている。さらに入／出力モジュールはタイミング制御ユニット30、DC電流変換／分配ユニット40、スイッチファブリックインターフェース50、および制御プロセッサ60を備えている。入／出力モジュールのタイミング制御ユニット30はローカルタイミングのシステムリファレンスを物理層に提供する。入／出力モジュールのDC電流変換／分配ユニット40は直流直流コンバータより成る。バックプレーンより可能な48ボルトはモジュールに必要な電圧に変換される。

30

40

【0006】

スイッチファブリックインターフェース50は、スイッチファブリック経由で各転送エンジンとスイッチにおける他のカードとを連結する。データはATMセル形式で搬送される。スイッチファブリック過密情報もまたファブリックから返送され、転送エンジン用に再フォーマット／サマライズされる。

【0007】

制御プロセッサ60は、バックグラウンド汎用プロセッサとしてペンティアムマイクロプロセッサにより構成される。転送エンジン20については以下に詳細に記載する。

【0008】

図2は入／出力モジュールの単一転送エンジン20のブロック図である。転送エンジン2

50

0 は、制御メモリ 160 およびパケットメモリ 170 と連動するインバウンドセグメンテーション／再アセンブリ（SAR）デバイス 150、制御メモリ 190 およびパケットメモリ 200 と連動するアウトバウンドセグメンテーション／再アセンブリ（SAR）180、パケットヘッダーディストリビュータ 210、パケット分類エンジン 110、ルートルックアップエンジン 100、転送プロセッサ 130、転送プロセッサと連動するキャッシュ 140、および DRAM／SRAM 120 から構成されている。

【0009】

転送エンジン 20 はインバウンドパケットおよびアウトバウンドパケットを処理する。インバウンドパケットに関して、パケットはインバウンド SAR 150 にトランスファされ、そこでパケットはパケットメモリ 170 に保存される。インバウンド SAR 150 は最高サービス品質パケットをパケットヘッダーディストリビュータ 210 に送る。転送プロセッサ 130 はパケットヘッダーディストリビュータのステータスチェックを行い、インバウンドパケットの受取りがいつ可能になるかを判断する。転送プロセッサ 130 は、パケットヘッダーディストリビュータ FIFO からのプライオリティの最も高いヘッダーを検索し、これをキャッシュ 140 に保存する。転送プロセッサ 130 はヘッダーを処理し、ルートルックアップエンジン 100 とパケット分類エンジン 110 の両方に要求されたヘッダーエレメントを同時に送信する。そこで、それらはイグレス（egress）キューに保存される。ルートルックアップエンジン 100 はパブリックあるいは仮想プライベートネットワークユニキャストルックアップを実行し、必要な場合、次のマルチキャストルックアップを実行する。一方で、パケット分類エンジン 110 はヘッダーエレメントに対してフィルタチェックを実行する。

10

20

【0010】

ルートルックアップエンジン 100 とパケット分類エンジン 110 のパケットヘッダーディストリビュータ 210 へのステータスパスが完了すると、転送プロセッサ 130 によりポーリングされるイグレスキューに結果を保存する。ルートルックアップエンジン 100 の動作については詳細を後述する。転送プロセッサ 130 はポーリングを行い、次に、新規ヘッダーが公式化されるルートルックアップと分類結果を得る。次に伝送情報に加えて新規ヘッダーがパケットヘッダーディストリビュータ 210 に書込まれる。インバウンド SAR 150 は新規にフォーマット化されたヘッダーを検索し、それをパケットの残りとともにパケットメモリ 170 に返送して、そして、セグメント化されスイッチファブリックに伝送されるパケットをスケジュールする。

30

【0011】

転送エンジン 20 はアウトバウンドパケットを次のように処理する。ATMセルはスイッチパケットからアウトバウンド SAR 180 にトランスファされ、そこで全体パケットが再アセンブリされ、パケットメモリ 200 に保存される。アウトバウンド SAR 180 は最高サービス品質パケットのヘッダーをパケットヘッダーディストリビュータ 210 に送信する。転送プロセッサ 130 はパケットヘッダーディストリビュータ 210 のステータスチェックを行い、アウトバウンドパケットの受け取りが可能であることを判断する。転送プロセッサ 130 はパケットヘッダーディストリビュータ FIFO からプライオリティの最も高いヘッダーを検索して、それをそれらのキャッシュに保存する。転送プロセッサ 130 はヘッダーを処理し、ルートルックアップエンジン 100 とパケット分類エンジン 110 の両方に要求されたヘッダーエレメントを同時に送信する。そこでそれらはイグレスキューに保存される。ルートルックアップエンジン 100 はパブリックあるいは仮想プライベートネットワークユニキャストルックアップを実行し、必要な場合、次のマルチキャストルックアップを実行する。一方で、パケット分類エンジン 110 はヘッダーエレメントに対してフィルタチェックを行う。

40

【0012】

ルートルックアップエンジン 100 とパケット分類エンジン 110 のパケットヘッダーディストリビュータ 210 へのステータスパスが完了すると、転送プロセッサ 130 によりポーリングが行われるイグレスキューとなる。転送プロセッサ 130 はポーリングを行い

50

、次に、新規ヘッダーが公式化されるルートルックアップと分類結果を得る。次に伝送情報に加えて新規ヘッダーがパケットヘッダーディストリビュータ210に書込まれる。アウトバウンドSAR180は新規にフォーマット化されたヘッダーを検索して、パケットの残りとともにパケットメモリ200にそれを返送し、物理フレームインターフェースに伝送されるパケットをスケジュールする。

【0013】

図3は転送エンジン20のルートルックアップエンジン100のブロック図である。ルートルックアップエンジンは転送プロセッサ130の次ホップインデックスを判断するハードウェアサーチエンジンである。ルートルックアップエンジン100は、転送プロセッサインターフェース65、入／出力FIFO40および入／出力FIFO50、ルックアップコントローラ20、ルックアップテーブルDRAM30、および制御プロセッサインターフェース70とにより構成される。転送プロセッサインターフェース65はサーチエンジンのキーをロードするために使用される。キーは、物理ポート／論理ポートに固有のツリー根のアドレス／仮想プライベートネットワークID、IP宛先アドレス、IPソースアドレス、ツリーの第一レベルのオーダ、および、シーケンスエラーチェック用にヘッダーといっしょに返送されるオプショナルリクエストIDとにより構成される。制御プロセッサインターフェース70はルックアップ間のテーブル更新に使用される。

10

【0014】

その動作において、転送プロセッサは、ルートルックアップエンジンとパケット分類エンジン両方の共通アドレスに1つあるいはそれ以上のキャッシュラインを同時に書込む。転送プロセッサインターフェース65はルートルックアップエンジンとパケット分類エンジン両方からの結果を同時に検索するために使用される。これらは、パケット分類エンジンより供給される結果部分とルートルックアップエンジンより供給される結果部分と共に同一アドレスから同時にリポートする。

20

【0015】

ルートルックアップエンジンはFIFOもしくは他のインターフェースを備え、スループットを最大限にすることを可能にする。入／出力FIFOは、転送プロセッサからの複数ルックアップリクエストを所定時間で明確にする。これらのパケットが異なるルックアップ時間を有する場合、FIFOはいくつかのヘッダーバーストをイン／アウトさせることによりルックアップレート平滑にする。ルートルックアップエンジンは仮想プライベートネットワーク(VPN)上でユニキャストIPアドレスルックアップおよびマルチキャストIPアドレスルックアップの両方を実行する。各仮想プライベートネットワークは、その仮想プライベートネットワークに特定のテーブル根に対するポインタとして使用される。多重仮想プライベートネットワークはソフトウェアの間接処理により同一ルートテーブルをマップする。

30

【0016】

IPパケットを転送するために、パワープロセッサコントローラはルートルックアップリクエストをパケットヘッダディストリビュータに提示する。ルートルックアップエンジンのハードウェア可変ストライドツリーテーブルは、ルートルックアップエンジンからの回答と共に含まれる結果インデックスによりリクエストを解決するために使用される。インデックスがパワープロセッサコントローラに対し使用可能となると、これらは転送テーブルからの転送情報への直接アクセスのために使用される。

40

【0017】

ルートルックアップからの結果は、次ホップインデックス、有効ルックアップ／ルート未発見ビット、ステータスフラグ、マルチキャストルックアップ実行ビット、統計情報、および、シーケンスエラーチェックのためのオプショナルリクエストIDとにより構成される。

【0018】

パケットがマルチキャストパケットである場合、ルートルックアップエンジンサーチはまず最初にIP宛先アドレスにおいて実行され、次にIPソースアドレスにおいて実行される。サーチにはプレフィックスエクスパンションおよびキャプチャによるマルチビット樹

50

木探索 (multi-bit tree search) を用いる。次ホップインデックスが発見されたかあるいはキー終了に達した場合にサーチは終了となる。この試みにおいて I P ソースアドレスの終了時に I P ヘッダーサーチの 64 ビット超を継続させると、ハードウェアエラーチェックはサーチを終了させ、エラーリポートを行う。ルーティングサブシステムのルートルックアップエンジンマネージャはルートルックアップエンジンメモリ管理を行う。ルートルックアップエンジンメモリは、デフォルト仮想プライベートネットワーク (V P N)、V P N 0 を含む、構成された仮想プライベートネットワーク (V P N) 各々に対するハードウェア可変ストライドツリー (V S T) ルートテーブルを保存するために使用される。各ハードウェア V S T は最小の近接メモリスペースが保証されている。残りのメモリスペースはメモリブロックのフリープールとして利用可能である。ルートが V S T テーブルに追加されると、また、これから削除されると、メモリブロックがこれらメモリプールから割当てされ、そして、これらメモリプールに解除される。ルーティングサブシステムはメモリの断片化を最小に保つよう管理する。

10

【 0 0 1 9 】

ルーティングサブシステムは、パワープロセッサコントローラのメモリ上の転送データベースレジデントとルートルックアップエンジンのメモリー上のハードウェア V S T テーブルレジデントとを確実に同期させておく。同様に、カーネルプロセッサのメモリ上のソフトウェア V S T テーブルレジデントはルートルックアップエンジンのハードウェア V S T テーブルと同期を保たなくてはならない。

【 0 0 2 0 】

20

ルートルックアップエンジンはローカルメモリに備わった V S T ベースルートルックアップテーブルを含む。これらの V S T テーブルは 1 つの特定ルートに一致する。ハードウェア V S T テーブルの各エントリは、パワープロセッサコントローラメモリ上の転送テーブルレジデント内における相当転送エントリに対するインデックスを有する。

【 0 0 2 1 】

各 V P N に関して、シングルソフトウェア V S T はカーナルプロセッサのメモリにて維持され、相当のハードウェア V S T はルートルックアップエンジンメモリにて維持される。V S T ルートテーブルは高レートにて最適マッチングプレフィックスをサポートする V S T データ構造にて構築される。V S T テーブルはサーチキーを別個のビットストリングに分割することで構築される。サーチキーは I P アドレスである。V S T レベル数はルートルックアップ中の最大メモリアクセス数を特定する。このように、ルックアップを完了する時間はバウンドされるが決定性を持つものではない。各 V S T レベル長はストライド長として知られ、その区分を表すノード内のエレメント数を判断する。

30

【 0 0 2 2 】

ハードウェア V S T データ構造は次のようになる。

エレメント

【 表 1 】

3 1	3 0	2 9	2 8 2 4	2 3 0
タイプ	インダイレクト	マルチキャスト	オーダ	ポインタ

40

タイプ : 0 = 葉, 1 = ノード

オーダ : ノードについて $\log_2$ 、葉についてプレフィックス長

ポインタ : 次ノードのベースアドレスに対するポインタ (無効の場合ゼロ、または転送エントリに対するインデックス (無効の場合 0 x F F F F F F)

ノード

【 表 2 】

エレメント[ビットストリング値=0]
エレメント[ビットストリング値=1]
.....
エレメント[ビットストリング値= $2^{\text{node order}} - 1$]

【0023】

ハードウェアルートサーチデータ構造はノードにより構成され、その各々はエレメントの
 アレーである。ノードにおけるエレメント数はノードオーダーにより決定する。ここでオー
 ダは、このノードの評価に使用されるサーチキー（宛先IPアドレス）のビット数である
 。各エレメントは、葉（leaf）か他のノードの根（root）のどちらかとして次ノード
 を指示するタイプインジケータを有する。現在のノードのタイプが「ノード」である場合
 、オーダーは次ノードのオーダーを示す。また、現在のノードのタイプが「葉」である場合、
 オーダーは葉のプレフィックス長を示す。ポインタフィールドはエレメントタイプにより決
 定する。タイプが「葉」である場合、ポインタはフォアグラウンドの転送テーブルのイン
 デックスであり、宛先アドレス／マスクペアの転送エントリを識別する。タイプが「ノード」
 である場合、ポインタは評価される次ノードのアドレスを識別するサーチスペースの
 ポインタである。次のテーブルはオーダーの意味と異なるタイプのポインタフィールドを示
 したものである。

【表3】

タイプ	オーダー	ポインタ
0(ノード)	Log_2 (ポインタにてノードにおけるエレメント数)	ノードに対するポインタ
1(葉)	エントリポインタの(プレフィックス長-1)	転送エントリに対するインデックス

【表4】

ルートルックアップエンジンテーブルエントリ：

ビット	31	30	29	28 24	23 0
フィールド	P (1)	T (1)	I(1)	オーダー(5)	RLE-Root/Flag.NHP

P：パリティビット。奇数パリティはハードウェアによりオプションにSDRAMに
 書込まれ、SDRAMより読取られる。

{1ビット}

T：タイプビット 0=ノード, 1=葉 {1ビット}

I：間接ビット。1=間接, 0=ダイレクト

{1ビット}

オーダー：1から32に相当する次レベルのオーダー {5ビット}

RLE-Root/Flag.NHP：（ルートルックアップエンジン（RLE）根／フラグ、次ホッ
 プポインタ）：Tビットによってこのデータフィールドに対し2コンテキストが存在する
 。ノードの場合、次ノードポインタ {24ビット}、または葉の場合、RLE次ホップポ
 インタ {16ビット} にプリペンドされたRLEフラグ（8ビット）。

【0024】

ルートルックアップエンジン根はサーチ開始に使用される。ルートルックアップエンジン
 根は根ポインタとツリーの第一レベルのオーダーを含む。第一ルックアップアドレスは根ポ

10

20

30

40

50

インタ、プラス、インデックスにより構成される。インデックスは宛先アドレスの上位ビットから導出される。ビット数は可変であり、第一レベルのオーダによって示される。このアドレスはテーブルからのエレメントをルックアップするために使用される。このエレメントが葉である場合、サーチは完了し、エレメントはパケット分類エンジン／ルートルックアップエンジンマネジャーにもどされる。エレメントがノードである場合、ポインタは次のルックアップに使用される。

【0025】

次ルックアップのアドレスは上記のポインタ、プラス、インデックスから成る。インデックスは宛先アドレスからの「次」ビットから導出される。正確なビットは、最終レベルのオーダ、プラス、このレベルのオーダから導出される。ゆえに、インデックスに使用される宛先アドレスの相当ビットを知る目的で、各レベルのオーダがトラックされなくてはならない。

10

【0026】

このアドレスはテーブルのエレメントをルックアップするために再度使用される。このエレメントが葉である場合、サーチは完了し、このエレメントはパケット分類エンジン／ルートルックアップエンジンマネジャーにもどされる。このエレメントがノードである場合、ポインタは次のルックアップに使用される。これは葉が見つけ出されるまで繰り返される。アクセス数やレベル数が最大数を超過した場合には、代わりにエラーが返送される。

【0027】

ある構成においてはルートルックアップに間接方法を有することが望ましい。これは間接ビットを用い達成される。間接ビットが設定されると、葉フラグは無視され、オーダフィールドは使用されず、そのインデックスロケーションから返送されたルート結果により、現24ビットポインタがダイレクトテーブルインデックスとして使用される。間接ファンクションはデプスをゼロに設定することにより呼び出すことも可能である。

20

【0028】

図4は、本発明によりルートルックアップを実行する方法300のフローチャートである。ルックアップはステップ310にてリクエストを受け取ることから開始する。ステップ320において、これがそのリクエストに対する最初のルックアップであるかどうかの判断が行われる。これがリクエストに対する最初のルックアップでない場合、ステップ330に示すように、TYPE、INDIRECT、ORDER、およびRLE ROOTの前回値が使用される。これがリクエストに対する最初のルックアップである場合、TYPE、INDIRECT、ORDER、およびRLE ROOTがリクエストブロックデータから取り出される。

30

【0029】

ステップ350において、TYPEビットが評価される。タイプが「葉」でない場合、ステップ420において、DRAMアドレスは、宛先アドレスとソースアドレスが繋がれたRLE ROOT+0に設定される。

【0030】

次にステップ430において、DRAMアドレスからの読取りが実行され、TYPE、INDIRECT、ORDER、およびRLE ROOTの新規値が得られる。ステップ440において、現在のオーダが、現オーダ、プラス、オーダに設定される。

40

【0031】

ステップ450において、現在のオーダが64よりも小さいかあるいは等しいかの判断がなされる。そうでない場合、エラーコンディションが示されてサーチは終了する。現在のオーダが64よりも小さいかあるいは等しい場合には、現在のオーダが32より大きいかどうかステップ470にて判断される。もしそうでない場合、ステップ320より開始する他のルックアップが実行される。現在のオーダが32により大きい場合、ステップ480が実行され、マルチキャストオペレーションを示すLビットが設定される。ステップ480の後、ステップ320より開始する他のルックアップが実行される。

【0032】

50

ステップ350に戻り、TYPEビットが「葉」でなかった場合、ステップ360にて間接ビットが設定されるかどうか判断される。間接ビットが設定されない場合、ステップ410で示すようにルックアップは完了し、その結果が保存される。

【0033】

間接ビットがステップ360にて設定されると、ステップ370において、DRAMアドレスがRLE ROOTに設定される。ステップ380において、RLE DRAMのアドレスが読取られ、TYPE、INDIRECT、ORDER、およびRLE ROOTの新規値が得られる。ステップ390において、INDIRECTビットが設定される場合、あるいはタイプが「ノード」の場合、エラーコンディションが示され、ステップ400で示すようにルックアップは終了する。INDIRECTビットが設定されないか、あるいはTYPEビットがノードを示さないかのどちらかである場合、ステップ410で示すようにルックアップは完了となり、その結果が保存される。

10

【0034】

以上、本発明の実施形態を詳細に説明したが、本発明の範囲を逸脱することなく他の方法でも具体化出来ることは当業者にとって明らかである。

【図面の簡単な説明】

【図1】 ATMスイッチのトップレベルブロック図である。

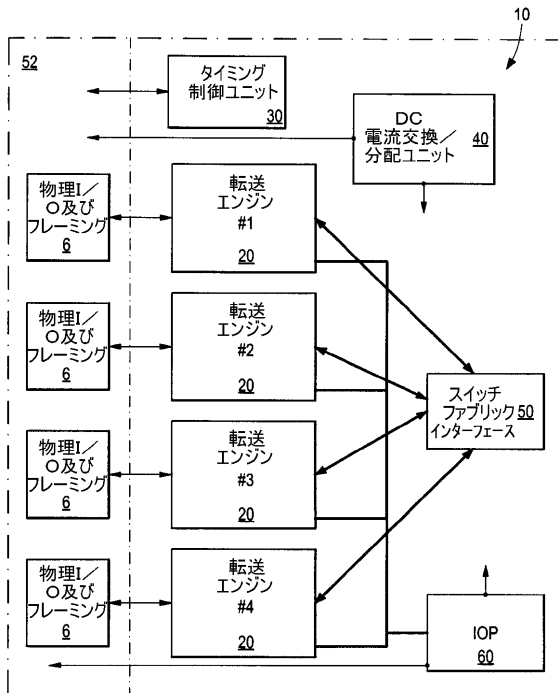
【図2】 転送エンジンのブロック図である。

【図3】 本発明によるルートルックアップエンジンのブロック図である。

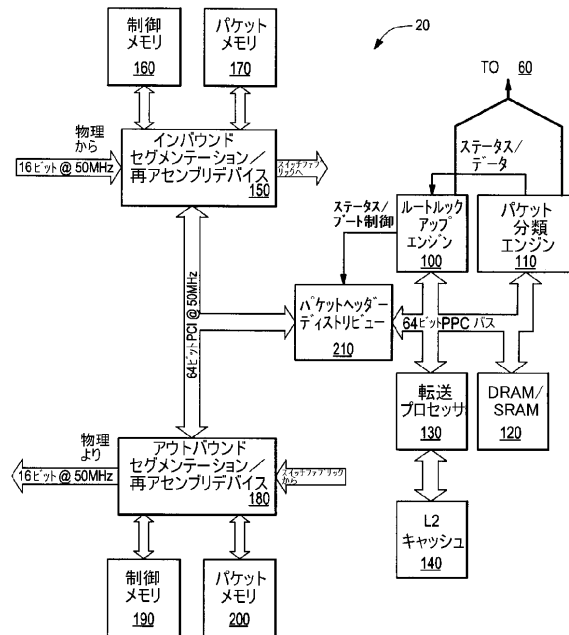
【図4】 本発明に基づいてルートルックアップを実行するフローチャートである。

20

【図1】



【図2】



フロントページの続き

(74)代理人 100104352

弁理士 朝日 伸光

(72)発明者 アンドリュー テー. ヘブ

アメリカ合衆国 01749 マサチューセッツ, ハドソン, レイクサイド アヴェニュー 62

(72)発明者 サンジェイ ジー. チェリアン

アメリカ合衆国 03033 ニューハンプシャー, ブルックリン, マックスウェル ドライヴ
6

審査官 清水 稔

(56)参考文献 特開平11-191781 (JP, A)

国際公開第96/000945 (WO, A1)

特表平09-511105 (JP, A)

V.Srinivasan, George Varghese, Faster IP lookups using controlled prefix expansion, Joint International Conference on Measurement and Modeling of Computer Systems Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems, 米国, ACM Press, 1998年, pp.1-10

(58)調査した分野(Int.Cl., DB名)

H04L 12/56

G06F 17/30

H04L 12/28