

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4741134号  
(P4741134)

(45) 発行日 平成23年8月3日(2011.8.3)

(24) 登録日 平成23年5月13日(2011.5.13)

(51) Int.Cl.

HO4L 12/56 (2006.01)

F 1

HO4L 12/56 100Z

請求項の数 42 (全 50 頁)

(21) 出願番号 特願2001-542932 (P2001-542932)  
 (86) (22) 出願日 平成12年12月8日 (2000.12.8)  
 (65) 公表番号 特表2003-516660 (P2003-516660A)  
 (43) 公表日 平成15年5月13日 (2003.5.13)  
 (86) 國際出願番号 PCT/CA2000/001443  
 (87) 國際公開番号 WO2001/043345  
 (87) 國際公開日 平成13年6月14日 (2001.6.14)  
 審査請求日 平成19年12月7日 (2007.12.7)  
 (31) 優先権主張番号 60/170,232  
 (32) 優先日 平成11年12月10日 (1999.12.10)  
 (33) 優先権主張国 米国(US)  
 (31) 優先権主張番号 60/212,966  
 (32) 優先日 平成12年6月21日 (2000.6.21)  
 (33) 優先権主張国 米国(US)

(73) 特許権者 509196822  
 エスエイテック・グループ・エー・ビー・  
 リミテッド・ライアビリティ・カンパニー  
 SATECH GROUP A. B. L  
 LIMITED LIABILITY CO  
 MPANY  
 アメリカ合衆国、19808 デラウェア  
 州、ウィルミントン、センタービル・ロー  
 ド、2711、スイート・400  
 (74) 代理人 100079108  
 弁理士 稲葉 良幸  
 (74) 代理人 100109346  
 弁理士 大貫 敏史

最終頁に続く

(54) 【発明の名称】最長一致アドレスルックアップのための方法および装置

## (57) 【特許請求の範囲】

## 【請求項 1】

サブツリーインデックスによってアクセスされる複数個のサブツリーエントリと、  
 第1のサブツリーエントリに記憶される稠密サブツリー記述子とを含み、前記稠密サブ  
 ツリー記述子は、稠密サブツリーの各葉ごとに値の表示を含み、さらに、  
 第2のサブツリーエントリに記憶される複数個の疎サブツリー記述子を含み、各疎サブ  
 ツリー記述子は少なくとも1つのノード記述子を含む、サブツリーの葉のための値を与える  
 ルックアップテーブル。

## 【請求項 2】

前記ノード記述子は、疎サブツリーの一組の葉のための共通値を識別する、請求項1に記載のルックアップテーブル。

## 【請求項 3】

前記稠密サブツリー記述子において「1」にセットされたビットの数に依存して前記稠密サブツリーの葉に対応する値を選択するマッパ論理をさらに含む、請求項1に記載のルックアップテーブル。

## 【請求項 4】

サブツリーエントリに複数個の疎サブツリー記述子を記憶するために設けられたビットの数は、前記稠密サブツリー記述子におけるビットの数に等しい、請求項1に記載のルックアップテーブル。

## 【請求項 5】

10

20

前記ノード記述子は、ランレンジス符号化を用いて符号化される、請求項1に記載のルックアップテーブル。

【請求項6】

前記ノード記述子におけるビットの数は、サブツリー葉におけるビットの数に1を加えたものに等しい、請求項5に記載のルックアップテーブル。

【請求項7】

複数のサブツリーの葉のための値は、「1」にセットされた最初の最下位ビットの位置に依存して前記ノード記述子に符号化される、請求項6に記載のルックアップテーブル。

【請求項8】

前記ノード記述子におけるビットの数は9であり、前記サブツリー葉におけるビットの数は8である、請求項6に記載のルックアップテーブル。

10

【請求項9】

前記サブツリー葉によってサーチ可能な連想メモリと、

選択された疎サブツリー記述子において前記ノード記述子を選択し、前記ノード記述子を8ビット符号化値に変換し、前記8ビット符号化値を前記連想メモリに記憶する変換論理とをさらに含む、請求項8に記載のルックアップテーブル。

【請求項10】

前記ノード記述子は、「1」にセットされた前記最初の最下位ビットの右側の最下位ビットを「ドントケア」にセットし、「1」にセットされた前記最初の最下位ビットを除去することにより変換される、請求項9に記載のルックアップテーブル。

20

【請求項11】

サブツリーインデックスによってアクセスされる複数個のサブツリーエントリと、

第1のサブツリーエントリに記憶される稠密サブツリー記述子とを含み、前記稠密サブツリー記述子は、稠密サブツリーの各葉ごとに値の表示を含み、さらに、

第2のサブツリーエントリに複数個の疎サブツリー記述子を記憶するための手段を含み、各疎サブツリー記述子は少なくとも1つのノード記述子を含む、サブツリーの葉のための値を与えるルックアップテーブル。

【請求項12】

前記ノード記述子は、疎サブツリーの一組の葉のための共通値を識別する、請求項11に記載のルックアップテーブル。

30

【請求項13】

前記稠密サブツリー記述子において「1」にセットされたビットの数に依存して前記稠密サブツリーの葉に対応する値を選択するための手段をさらに含む、請求項12に記載のルックアップテーブル。

【請求項14】

サブツリーエントリに複数個の疎サブツリー記述子を記憶するために設けられたビットの数は、前記稠密サブツリー記述子におけるビットの数に等しい、請求項12に記載のルックアップテーブル。

【請求項15】

前記ノード記述子は、ランレンジス符号化を用いて符号化される、請求項12に記載のルックアップテーブル。

40

【請求項16】

前記ノード記述子におけるビットの数は、サブツリー葉におけるビットの数に1を加えたものに等しい、請求項15に記載のルックアップテーブル。

【請求項17】

複数のサブツリーの葉のための値は、「1」にセットされた最初の最下位ビットの位置に依存して前記ノード記述子に符号化される、請求項16に記載のルックアップテーブル。

【請求項18】

前記ノード記述子におけるビットの数は9であり、前記サブツリー葉におけるビットの

50

数は 8 である、請求項 17 に記載のルックアップテーブル。

【請求項 19】

前記サブツリー葉によってサーチ可能な連想メモリと、

選択された疎サブツリー記述子において前記ノード記述子を選択し、前記ノード記述子を 8 ビット符号化値に変換し、前記 8 ビット符号化値を前記連想メモリに記憶する変換するための手段とをさらに含む、請求項 18 に記載のルックアップテーブル。

【請求項 20】

前記ノード記述子は、「1」にセットされた前記最初の最下位ビットの右側の最下位ビットを「ドントケア」にセットし、「1」にセットされた前記最初の最下位ビットを除去することにより変換される、請求項 19 に記載のルックアップテーブル。

10

【請求項 21】

サブツリーインデックスによって複数個のサブツリーエントリにアクセスするステップと、

第 1 のサブツリーエントリに稠密サブツリー記述子を記憶するステップとを含み、前記稠密サブツリー記述子は、稠密サブツリーの各葉ごとに値の表示を含み、さらに、

第 2 のサブツリーエントリに複数個の疎サブツリーを記憶するステップを含み、各疎サブツリー記述子は少なくとも 1 つのノード記述子を含む、ルックアップテーブルに実現されるサブツリーの葉のための値を与えるための方法。

【請求項 22】

前記ノード記述子は、疎サブツリーの一組の葉のための共通値を識別する、請求項 21 に記載の方法。

20

【請求項 23】

前記稠密サブツリー記述子において「1」にセットされたビットの数に依存して前記稠密サブツリーの葉に対応する値を選択するステップをさらに含む、請求項 21 に記載の方法。

【請求項 24】

サブツリーエントリに複数個の疎サブツリー記述子を記憶するために設けられたビットの数は、前記稠密サブツリー記述子におけるビットの数に等しい、請求項 21 に記載の方法。

【請求項 25】

30

前記ノード記述子は、ランレンジス符号化を用いて符号化される、請求項 21 に記載の方法。

【請求項 26】

前記ノード記述子におけるビットの数は、サブツリー葉におけるビットの数に 1 を加えたものに等しい、請求項 25 に記載の方法。

【請求項 27】

複数のサブツリー葉のための値は、「1」にセットされた最初の最下位ビットの位置に依存して前記ノード記述子に符号化される、請求項 26 に記載の方法。

【請求項 28】

前記ノード記述子におけるビットの数は 9 であり、前記サブツリー葉におけるビットの数は 8 である、請求項 27 に記載の方法。

40

【請求項 29】

前記サブツリー葉によってサーチ可能な連想メモリを設けるステップと、

選択された疎サブツリー記述子において前記ノード記述子を選択するステップと、

前記ノード記述子を 8 ビット符号化値に変換するステップと、

前記 8 ビット符号化値を前記連想メモリに記憶するステップとをさらに含む、請求項 28 に記載の方法。

【請求項 30】

前記ノード記述子は、「1」にセットされた前記最初の最下位ビットの右側の最下位ビット「ドントケア」にセットし、「1」にセットされた前記最初の最下位ビットを除去す

50

ることにより変換される、請求項 2 9 に記載の方法。

【請求項 3 1】

複数レベルルックアップテーブル中のキーに対応するルートをサーチするための方法であって、

サブツリーインデックスによってサブツリーエントリを選択するステップを含み、前記サブツリーエントリは複数個の疎サブツリー記述子を含み、さらに

選択されたサブツリーエントリ中の前記複数個の疎サブツリー記述子のうち 1 つを選択するステップを含み、前記疎サブツリー記述子は複数個のノード記述子を含み、さらに

選択された前記疎サブツリー記述子中の前記ノード記述子をマスクされた値に変換するステップと、

10

前記マスクされた値を最長プレフィックスの順序でメモリにロードするステップと、前記キーの部分の最長プレフィックス一致を求めて前記メモリをサーチするステップと、

前記最長プレフィックス一致のアドレスに依存して別のメモリに記憶される前記ルートを選択するステップとを含む、方法。

【請求項 3 2】

前記メモリは連想メモリである、請求項 3 1 に記載の方法。

【請求項 3 3】

前記ノード記述子は、疎サブツリーの一組の葉のための共通値を識別する、請求項 3 2 に記載の方法。

20

【請求項 3 4】

前記ノード記述子は、ランレンジス符号化を用いて符号化される、請求項 3 3 に記載の方法。

【請求項 3 5】

前記ノード記述子におけるビットの数は、サブツリー葉におけるビットの数に 1 を加えたものに等しい、請求項 3 4 に記載の方法。

【請求項 3 6】

複数のサブツリー葉のためのマスクされた値は、「1」にセットされた最初の最下位ビットの位置に依存して前記ノード記述子に符号化される、請求項 3 5 に記載の方法。

【請求項 3 7】

前記ノード記述子におけるビットの数は 9 であり、前記マスクされた値におけるビットの数は 8 である、請求項 3 6 に記載の方法。

30

【請求項 3 8】

前記ノード記述子は、「1」にセットされた前記最初の最下位ビットの右側の最下位ビットを「ドントケア」にセットし、「1」にセットされた前記最初の最下位ビットを除去することにより変換される、請求項 3 7 に記載の方法。

【請求項 3 9】

前記ルックアップテーブルは、少なくとも 1 つの稠密サブツリーと、複数個の疎サブツリーとを含み、各疎サブツリー記述子に含まれるノード記述子の数は前記疎サブツリーの葉の数より少ない、請求項 1 に記載のルックアップテーブル。

40

【請求項 4 0】

前記ルックアップテーブルは、少なくとも 1 つの稠密サブツリーと、複数個の疎サブツリーとを含み、各疎サブツリー記述子に含まれるノード記述子の数は前記疎サブツリーの葉の数より少ない、請求項 1 に記載のルックアップテーブル。

【請求項 4 1】

前記ルックアップテーブルは、少なくとも 1 つの稠密サブツリーと、複数個の疎サブツリーとを含み、各疎サブツリー記述子に含まれるノード記述子の数は前記疎サブツリーの葉の数より少ない、請求項 2 1 に記載の方法。

【請求項 4 2】

各疎サブツリー記述子に含まれるノード記述子の数は前記疎サブツリーの葉の数より少

50

ない、請求項 3 1 に記載の方法。

## 【発明の詳細な説明】

## 【0001】

## 【発明の背景】

インターネットはルータで接続された 1 セットのネットワークである。ルータは、受信したデータパケットを転送すべき次のホップを、各々の可能なデスティネーションネットワークごとに示すルーティングテーブルを維持する。次のホップとは、別のルータまたは最終的なデスティネーションであろう。

## 【0002】

ルータのポートで受信されるインターネットプロトコル (IP) データパケットは IP デ 10  
スティネーションアドレスを含む。IP デスティネーションアドレスは IP データパケットの最終デスティネーションである。現在、IP バージョン 4 (IPv4) および IP バージョン 6 (IPv6) の 2 つの IP バージョンが存在する。IPv4 は、IP デスティネーションアドレスを記憶する、データパケットに含まれる IP ヘッダに 32 ビットのフィールドを設ける。ルータは、IP ヘッダに記憶された IP デスティネーションアドレスに依存して、接続された受信データパケットを次のループのルータかまたは、デスティネーションがローカルネットワークであれば最終的なデスティネーションに転送する。

## 【0003】

32 ビット IPv4 デスティネーションアドレスは 40 億の可能なルートをもたらす。イ 20  
ンターネットルータは典型的に、40 億の可能なルートのうち 5 万を記憶する。しかしながら、記憶されるルートの数は、インターネットの発展および IPv6 の広がりによって増加している。

## 【0004】

もともと、IP アドレス空間は 3 つのクラスの IP アドレス、すなわち A, B および C に分割されていた。各々の IP アドレス空間はネットワークアドレスおよびホストアドレスに分割された。クラス A は、126 のネットワークおよびネットワーク当たり 1600 万個のホストを許容した。クラス B は、ネットワーク当たり 64,000 個のホストを備えて、16382 個のネットワークを許容し、クラス C は、ネットワーク当たり 256 個のホストを備えて、200 万個のネットワークを許容した。しかしながら、IP アドレス空間を異なるクラスに分割することにより、利用可能な IP アドレスの数が減少してしまった。クラス C はネットワーク当たり最大で 256 個のホストしか許容しないが、これは大抵の組織にとっては少なすぎる。したがって、大抵の組織にクラス B アドレスを割当てて、64,000 個のホストアドレスをとった。このホストアドレスは、それらを割当てられた組織が使用しなくても、他の組織が使用不可能なものである。クラス B の IP アドレスを有する組織のホストのすべては、たとえば、27.32.x.x などの、16 の最上位ビット (MBS) に同じネットワークアドレスを記憶する。

## 【0005】

クラスレス・インターネットドメイン・ルーティング (CIDR) は、未使用の IP ホストアドレスを自由化するために導入された。残余の未使用のネットワークは、サイズ可変のブロックで組織に割当てられる。500 個のアドレスを必要とする組織は 500 個の連続するアドレスを入手する。たとえば、ある組織に、128.32.x.x で始まる 500 個の利用可能なアドレスを割当て可能である。ルータが記憶するルートの数は、クラスレス・インターネットドメイン・ルーティングの導入以来増加した。クラスレス・インターネットドメイン・ルーティングは、一致するネットワークアドレスをサーチして IP デスティネーションアドレスの対応する次のホップを見出す代わりに、対応するルートを見出すのに最長プレフィックス一致を必要とする。たとえば、サーチは、たとえば 128.x.x.x などのクラス B の IP アドレスの 16 MBS の後はもはやストップできない。なぜなら、128.32.4.x は別の組織に割当てられ得るからであり、これには次の異なるホップを必要とする。

## 【0006】

10

20

30

40

50

キーに対して最長プレフィックス一致をサーチするための1つの方法は、バイナリツリーサーチの使用によるものである。バイナリツリーサーチは、1ビットずつ32のレベルまで32ビット入力を一致させ、32ビットのキーに一致するエントリを見出すには32のサーチが必要である。一致をサーチするための別の方法はパトリシアツリーの使用によるものである。パトリシアツリーは、バイナリツリーの葉までにエントリが存在しなければ、必要なサーチ数を減らす。

【0007】

IPデスティネーションアドレスと関連する次のホップを効率的にサーチするためのまた別の方法が、1998年5月11日に出願された、プロドニックら(Brodnick et al.)による「高速ルーティングルックアップのための方法およびシステム」(“Method and System for Fast Routing Lookups”)と題された、PCT出願連続番号PCT/SE98/00854に記載されている。プロドニックが記載する方法は、重複するルートを記憶しないことで、記憶される次のホップの数を減じる。次のホップの数を減じることにより、メモリの必要量を低減し、それによりルートルックアップテーブルを高速キャッシュメモリに記憶できるようになる。

10

【0008】

プロドニックらは、バイナリツリーを3レベルに分割する。バイナリツリーを3つのレベルに分割することにより、サーチ数を3に減らした。第1のレベルのインデックス付けされたエントリは、ルートをエントリからとる場合、サーチを第1のレベルで終了可能であるかまたは、IPデスティネーションアドレスのさらなる部分を用いて、その後のレベルまでサーチを継続しなければならないかを示す。

20

【0009】

図1Aは、バイナリツリーの第1のレベルを表わす先行技術の64K(65536)ビットマップを図示する。64Kビットマップ30は、深度16のバイナリツリーの葉またはノード44を表わし、ノード44当り1ビットである。ビットマップは長さ16のビットマスクに分割される。64Kビットマップには $2^{12} = 4096$ のビットマスクが存在する。図1Aに1つのビットマスクが示される。ビットマップ30中のビットは、ノード44に対応するポインタのアレイに記憶されたルートインデックスまたはサブツリーが存在すれば、「1」にセットされる。ノードが前のノード44とルートエントリを共有していれば、ビットマップ30中のビットは、「0」にセットされる。

30

【0010】

図1Bは、キャッシュメモリに実現される先行技術のルックアップテーブルを図示する。ルックアップテーブルは、コードワード36のアレイ、ベースインデックス34のアレイおよびマップテーブル40を含む。図1Bには32ビットのIPアドレス38も示される。コードワード46は、ビットマップ30(図1A)の各ビットマスクごとにコードワード36のアレイに記憶される。コードワード46は6ビットの値46aおよび10ビットのオフセット46bを含む。ベースインデックス42は、コードワード36のアレイ中の4つのコードワード46ごとに、ベースインデックス34のアレイに記憶される。

【0011】

コードワード36のアレイ、ベースインデックス34のアレイおよびマップテーブル40を用いて、ポインタ(図示せず)のアレイ中のポインタを選択する。ポインタは、ルートインデックスまたはインデックスを記憶してさらなるサーチを行なう。

40

【0012】

ポインタのアレイ中のポインタ群は、コードワード36のアレイ中のコードワード46およびベースインデックス34のアレイ中のベースインデックス42を選択することによって選択される。コードワード46は、IPアドレス38の最初の12ビット50を用いて選択される。ベースインデックス42は、IPアドレス38の最初の10ビット48を用いて選択される。マップテーブル32を用いて、ポインタ群の中の正しいポインタが選択される。

【0013】

50

選択されたコードワード 3 6 中の 10 ビットの値 4 6 b は、マップテーブル 3 2 へのインデックスである。マップテーブル 3 2 は、ビットマスク内のビット数を 4 ビットのオフセットにマッピングする。オフセットは、ポインタのアレイの中の選択されたポインタ群内のポインタを特定する。10 ビットの値 4 6 b はマップテーブル 3 2 中の列を選択し、IP アドレス 5 2 のビット 19 : 16 は 4 ビットのオフセット 5 4 を選択する。

#### 【0014】

このように、ポインタのサーチには、以下のキャッシュメモリアクセスが必要である。すなわち、(1) 16 ビットのコードワード 4 6 を読出す；(2) 16 ビットのベースアドレス 4 2 を読出す；(3) マップテーブル 3 2 から 4 ビットのオフセット 5 4 を読出す；(4) ポインタインデックスがベースアドレス 4 2 、コードワードオフセット 4 6 a および 4 ビットのオフセット 5 4 の和であるポインタインデックスでポインタを読出す。

10

#### 【0015】

バイナリツリーの各レベルで同じメモリアクセスが必要である。したがって、3 レベルのサーチには 12 のメモリアクセスが必要である。

#### 【0016】

IP 宛先アドレスと関連付けられる次のホップのために効率よくサーチを行なうための方法が、1998年5月11日に出願されたプロドニックらによる「高速ルーティングルックアップのための方法およびシステム」と題する PCT 出願連続番号第 PCT/SE98/00854 号に記載される。プロドニックらは、疎であるサブツリー；すなわち、8 個未満のルートが記憶されるサブツリーに必要とされるバイトの数を低減するための方法を記載する。24 バイトのメモリが、疎サブツリーのための 8 個のルートを記憶するために割当てられる。24 バイトは、サーチキーに対応する値を記憶する 8 個の 8 ビットルートエントリのアレイと、各ルートエントリごとのそれぞれの 16 ビットのポインタとを含む。

20

#### 【0017】

値は、降順にメモリのアレイに記憶される。アレイは、サーチキーより小さいまたは等しい値を見つけるためにサーチされる。メモリへの複数アクセスが必要とされる。多くとも 7 個のメモリアクセスが、サーチキーに対応する値を見つけるために必要とされ得る。

#### 【0018】

#### 【発明の概要】

30

複数の疎サブツリー記述子がメモリのサブツリーエントリに記憶されることを可能にするルックアップテーブルが提供される。ルックアップテーブルは、サブツリーインデックスによってアクセスされる複数個のサブツリーエントリを含む。稠密サブツリー記述子が、第 1 のサブツリーエントリに記憶される。稠密サブツリー記述子は、稠密サブツリーの各葉ごとに値の表示を含む。複数個の疎サブツリー記述子が、第 2 のサブツリーエントリに記憶される。疎サブツリー記述子の各々は、少なくとも 1 つのノード記述子を含む。

#### 【0019】

ノード記述子は、疎サブツリーの一組の葉のための共通値を識別する。メモリは、疎サブツリー記述子および稠密サブツリー記述子の両方を記憶する。稠密サブツリー記述子が第 1 のサブツリーエントリに記憶される。稠密サブツリー記述子は、稠密サブツリーの葉当たり 1 ビットを含む。稠密サブツリー記述子は、稠密サブツリーの葉のための値を示す。複数個の疎サブツリー記述子が第 2 のサブツリーエントリに記憶される。各疎サブツリー記述子は、少なくとも 1 つのノード記述子を含む。ノード記述子は、疎サブツリーの一組の葉のための共通値を識別する。

40

#### 【0020】

マッパ論理が、稠密サブツリー記述子において「1」にセットされたビットの数に依存して稠密サブツリーの葉に対応する値を選択する。サブツリーエントリに疎サブツリー記述子を記憶するために設けられたビットの数は、稠密サブツリー記述子におけるビットの数に等しい。

#### 【0021】

50

ノード記述子は、ランレンゲス符号化を用いて符号化される。ノード記述子におけるビットの数は、サブツリー葉におけるビットの数に1を加えたものに等しい。複数のサブツリー葉のための値は、「1」にセットされた最初の最下位ビットの位置に依存してノード記述子に符号化される。ノード記述子におけるビットの数は9であり、サブツリー葉におけるビットの数は8である。

【0022】

サブツリー葉によってサーチ可能な連想メモリが設けられる。変換論理が、選択された疎サブツリー記述子においてノード記述子を選択し、ノード記述子を8ビット符号化値に変換し、8ビット符号化値を連想メモリに記憶する。ノード記述子は、「1」にセットされた最初の最下位ビットの右側の最下位ビットを「ドントケア」にセットし、「1」にセットされた最初の最下位ビットを除去することにより変換される。

10

【0023】

この発明の以上およびその他の目的、特徴および利点は、添付の図面に図示されるように、この発明の好ましい実施例の以下のより特定的な説明から明らかになるであろう。図面中の同じ参照番号は異なる図面を通じて同じ部分を示す。図面は、この発明の原則を図示するのに供される代わりに、縮尺を変更したり、強調したりということを必ずしも行なわなくてよい。

【0024】

【詳細な説明】

この発明の好ましい実施例の説明は以下のとおりである。

20

【0025】

デンスモード

図2Aは、この発明の原則に従う最長一致プレフィックスルックアップテーブル100を図示する。ルックアップテーブル100はキー104のルートインデックス102を与える。ルートインデックス102を用いて、IPデスティネーションアドレスに向けた次のホップにアクセスする。図2Aに示された実施例では、キー104は32ビット幅であるが、キー104は32ビットに限定されるものではない。ルックアップテーブル100は3つのマッパ106a-cを含む。各々のマッパ106a-cは別個にアドレス指定されたメモリを含む。キー104に対応するデフォルトルートインデックスまたはルートインデックス102は、マッパ106a-cのうち1つの場所に記憶される。各マッパからのマッパ出力110a-cは、多数のマッパをサーチする必要がある場合、キーについてすべてのマッパ106a-cにアクセスするまで、遅延メモリ150a-cに記憶される。

30

【0026】

マルチプレクサ108は、マルチプレクサ108の入力に転送されるマッパ出力110a-cの1つをルートインデックス102として選択する。マッパ出力110a-cは、マッパ出力110a-cの最上位ビット(MSB)に依存して選択される。マッパ出力110a-cがルートインデックス102を含む場合のみ、マッパ出力110a-cのMSBが‘1’となる。

【0027】

図2Bは、図2Aに示されたルックアップテーブル100のマッパ106a-cに記憶されたエントリのバイナリツリー表示を図示する。図2Bは図2Aと関連して説明される。32ビットのキー104は32レベルのバイナリツリーとして表わすことができる。バイナリツリー実現例は、1ビットずつ32レベルまでサーチするには、32のサーチを必要とする。サーチ数を減らすため、バイナリツリーの32のレベルは3つのマッパレベル112a-cに分割され、各マッパレベル112a-cはマッパ106a-c(図2A)に対応する。マッパレベル\_1 112aは、バイナリツリーの32のレベルのうち最初の16個を含む。しかしながら、簡潔さのため、図2Bには16のレベルのうち5つしか示されない。マッパレベル\_2 112bは、32レベルのバイナリツリーの次の8レベルを含み、図2Bには8つのレベルのうち3つが示される。マッパレベル\_3は、32レベル

40

50

ルのバイナリツリーのうち最後の 8 レベルを含み、図 1 B には 8 つのレベルのうち 3 つが示される。各マッパレベル 1 1 2 a - c は複数のノードを含む。1 6 レベル (キー 1 0 4 の 1 6 の M S B ) がマッパレベル - 1 1 1 2 a にあり、8 レベルがマッパレベル\_2 1 1 2 b にありかつ 8 レベルがマッパレベル\_3 にあるように 3 2 レベルを分割することが、現在のメモリ技術においては最適であるように思われるが、この発明はこの構成に限定されるものではない。

#### 【 0 0 2 8 】

キー 1 0 4 の最初の 1 6 ビットに対して 1 ビットずつ別個の 1 6 のサーチを行なう代わりに、キー 1 0 4 の最初の 1 6 ビットと関連するルートインデックス 1 0 2 をマッパ 1 0 6 a (図 2 A) に記憶する。マッパ 1 0 6 a (図 2 A) は、キー 1 0 4 の最初の 1 6 ビットの M S B によって直接にインデックス付けされる。次のマッパ 1 0 6 b は、前のマッパ 1 0 6 a が、キー 1 0 4 と関連する次のホップ情報にアクセスするのに用いるルートインデックス 1 0 2 を記憶しているか否かに依存して、サーチされる。

#### 【 0 0 2 9 】

図 2 B に示されるように、マッパレベル\_1 1 1 2 a に示されるノードまたは葉は、それぞれ r 0 および r 1 と標識付けられた 2 つのルート 1 1 4 、 1 1 6 と、それぞれ s 0 および s 1 と標識付けられたマッパレベル\_2 1 1 2 b への 2 つのポインタ 1 3 0 <sup>4</sup> および 1 3 0 <sup>23</sup> を含む。各ルート 1 1 4 、 1 1 6 ごとのルートインデックス 1 0 2 は L 1 マッパ 1 0 6 a に記憶される。また、 L 2 マッパ 1 0 6 b に向けたアドレスポインタ 1 2 0 はサブツリーインデックス 1 3 0 <sup>4</sup> に対して記憶され、 L 2 マッパ 1 0 6 b に向けたアドレスポインタ (図示せず) はサブツリー 1 3 0 <sup>23</sup> に対して記憶される。サブツリーインデックス 1 3 0 <sup>4</sup> に対して、マッパ 1 0 6 a 中のマッパエントリ 1 4 0 <sup>4</sup> に記憶されるアドレスポインタ 1 2 0 は、キー 1 0 4 と関連のルートインデックス 1 0 2 を見出すためには、次のレベルのサーチが必要であることを示す。

#### 【 0 0 3 0 】

ツリー中のいずれのノードの値も、根 1 1 4 からの経路をトレースすることによって判定することができる。バイナリツリー中の各ノードは、右側の子および左側の子の 2 つの子を有して示される。親ノードが '1' ならば右側の子が選択される。親ノードが '0' ならば左側の子が選択される。根 1 1 4 からノード 1 1 6 への経路をトレースすると、 r 1 は、 '0 1 0' にセットされた M S B を有するすべてのキーに対して、 L 1 マッパ 1 0 6 a 中にルートインデックス 1 0 2 として記憶される。根ノード 1 1 4 から s 0 ノード 1 3 0 <sup>4</sup> への経路をトレースすると、 s 0 は、 '0 0 0 1 1' にセットされた M S B を有するすべてのキーに対して L 1 マッパ 1 0 6 a に記憶される。

#### 【 0 0 3 1 】

L 1 マッパ 1 0 6 a はダイレクト・マップト・マッパであり、マッパレベル\_1 1 1 2 a のボトムレベルの各ボトムレベルノードまたは葉ごとにルートインデックス 1 0 2 を記憶する。マッパレベル\_1 1 1 2 a のボトムレベルは、 3 2 レベルバイナリツリーの 1 6 番目のレベルである。1 6 番目のレベルは 6 4 K のノードを有する。しかしながら、例示の目的のため、マッパレベル\_1 1 1 2 a のボトムレベルは、 3 2 レベルのバイナリツリーのレベル - 5 として示される。L 1 マッパ 1 0 6 a に示されるルートインデックス 1 0 2 は、マッパレベル\_1 1 1 2 a のレベル - 5 の 1 3 0 <sup>1</sup> - 1 3 0 <sup>32</sup> ノードに対応する。根ノード 1 1 4 からレベル\_5 のノード 1 3 0 <sup>1</sup> 、 1 3 0 <sup>2</sup> 、 1 3 0 <sup>3</sup> への経路をトレースすると、ルートインデックス 1 0 2 は r 0 である。したがって、 r 0 は、 L 1 マッパ 1 0 6 a の場所 1 4 0 <sup>1</sup> 、 1 4 0 <sup>2</sup> 、 1 4 0 <sup>3</sup> に、すなわち、インデックス 0 0 0 0 0 、 0 0 0 0 1 および 0 0 0 1 0 に記憶される。ノード 1 3 0 <sup>4</sup> はサブツリーインデックス s 0 を記憶し、したがって、 s 0 は、アドレス 0 0 0 1 1 に、 L 1 マッパ 1 0 6 a の中の場所 1 4 0 <sup>4</sup> に記憶される。同様に、レベル - 5 のノード 1 3 0 <sup>5</sup> - 1 3 0 <sup>8</sup> に対するルートインデックス 1 0 2 は r 0 であり、したがって、 L 1 マッパ 1 0 6 a の中のアドレス 0 0 1 0 0 、 0 0 1 0 1 、 0 0 1 1 0 および 0 0 1 1 1 の場所 1 4 0 <sup>5</sup> 、 1 4 0 <sup>6</sup> 、 1 4 0 <sup>7</sup> 、 1 4 0 <sup>8</sup> は r 0 を記憶する。レベル - 5 のノード 1 3 0 <sup>9</sup> - 1 3 0 <sup>12</sup> に対するルートインデ

10

20

30

40

50

ツクス102はr1であり、したがって、L1マッパ106aの中のアドレス01000および010001の場所140<sup>9</sup>、140<sup>10</sup>はr1を記憶する。

【0032】

L1マッパ106a中の各々の場所は、直接にもしくはレベル-5のノード300<sup>1-32</sup>の親を介してレベル-5のノード300<sup>1-300<sup>32</sup></sup>に割当てられるルートインデックス102または次のマッパ106b-cへのアドレスポインタを記憶する。マッパレベル\_3 106cは、32レベルのバイナリツリーのボトムレベルで、ノード138のh0およびノード140のh1の2つのホストノードを含む。ホストノードのサーチは、キー104のすべてのビットのサーチを必要とする。図2Aに示されるように、h0に対するルートインデックス102は、L3\_マッパ106c中の場所146<sup>4</sup>に記憶される。L1マッパ106aと異なり、L2マッパ106bおよびL3マッパ106cは直接にマッピングされない。

【0033】

マッパ106bおよび106cでは、ルートインデックス102は各々の可能な入力ごとに記憶されない。ノードに対するルートインデックス102がマッパ106b-cに以前に記憶されたルートインデックス102と異なる場合にのみ、ルートインデックス102が記憶される。マッパレベル\_2 112bに示される第1のサブツリーAのレベル-3ノードを見ると、ノード132<sup>1</sup>およびノード132<sup>2</sup>に対するルートインデックス102はr0であり、したがって、r0に対するルートインデックスは、L2マッパ106bの中のノード132<sup>1</sup>およびノード132<sup>2</sup>の両者について、場所142<sup>1</sup>に記憶される。ノード132<sup>2</sup>に対するサブツリーインデックスs0は場所142<sup>2</sup>に記憶される。レベル-3のノード132<sup>4</sup>およびレベル-3のノード132<sup>5</sup>および132<sup>6</sup>と関連のルートインデックス102はr0であり、これは、前のノード132<sup>2</sup>に対して記憶されたs0とは異なる。したがって、r0はL2マッパ106b中の次の場所142<sup>3</sup>に記憶される。ルートr2は、ノード132<sup>7</sup>に対して、L2マッパ106b中の場所142<sup>4</sup>に記憶される。なぜなら、ノード132<sup>7</sup>は前のノード132<sup>6</sup>と同じルートを共有しないからである。サブツリーインデックスs3は次のレベル-3のノード132<sup>7</sup>に対して記憶される。したがって、s3はL2マッパ106bの中の場所142<sup>5</sup>に記憶される。前のノードからルート変更があるときにだけルートインデックスを記憶することにより、ルートインデックス102を記憶するのに必要なメモリが低減される。示されるように、マッパレベル\_2 112bの第1のサブツリーAのレベル-3の8つのノード132<sup>1-8</sup>についてルートインデックスを記憶するためには、L2マッパ106bの中の場所は5つしか必要でない。ノンダイレクトマッパ106b、106cは、図5と関連してより詳細に後述される。

【0034】

図3は、この発明の原則に従う40ビットのキー210のための最長一致プレフィックスルックアップテーブル200を図示する。1つの実施例では、40ビットのキーは、8ビットのプレフィックスおよび32ビットのIPアドレスを含む。8ビットのプレフィックスは、32ビットのIPアドレスと関連する仮想私設網(VPN)識別子であり得る。ルックアップテーブル200は4つのマッパ106a-dを含む。マッパ106aは、図2Aと関連して説明されたように、ダイレクト・マップト・マッパである。マッパ106b-dはインダイレクトマッパである。マッパ106aは、40ビットのキー210の16のMSBに対応する、L2マッパ106bに対するサブツリーインデックスまたはルートインデックス102を記憶する。したがって、L1マッパは、第1のマッパレベル112a(図2B)中の64Kのノードの各々ごとに1つの、64Kの可能な場所を有する。L1マッパ106a中の対応する場所に記憶されるL1マッパエントリデータ220aは、パイプライン208およびL2インダイレクトマッパ106bに転送される。L1マッパエントリデータ220aが、キー210bの次の8ビットを用いる次のレベルのサーチが必要であることを示せば、キー210bの次の8ビットおよびL1マッパエントリデータ220aに依存して、L2インダイレクトマッパ106bでサーチが行なわれる。

## 【0035】

第2のレベルのサーチの結果がL2マッパエントリデータ220b上に与えられ、これはパイプライン208およびL3インダイレクトマッパ106cに転送される。第3のレベルのサーチは、キー210cの次の8ビットおよびL2マッパエントリデータ220bに依存して、L3インダイレクトマッパ106cで行なわれる。

## 【0036】

サーチの結果は、L3マッパエントリデータ220c上を、パイプライン208およびL4インダイレクトマッパ106dに与えられる。L3マッパエントリデータ220cは、キー210dの最後の8ビットおよびL3マッパエントリデータ220cに依存して、L4インダイレクトマッパ106dで行なわれる。

10

## 【0037】

第4のサーチの結果はL4マッパエントリデータ220d上に与えられる。キー210に対する最長一致プレフィックスと関連するルートインデックス102は、マッパ106a-dのうちの1つの中のただ1つの場所に記憶される。したがって、パイプライン208に転送されるルートインデックス102は、ただ1つのマッパエントリデータ220a-dに含まれる。マッパ106a-dのうち1つ、たとえばマッパ106bにルートインデックス102を見出すと、残余のマッパ106c-dのサーチは必要ではなく、マッパ106c-dはアクセスされない。パイプライン208は、マッパエントリデータ220a-dのうち1つに含まれるルートインデックス102を選択するためのマルチプレクサ108(図2A)を含む。たとえば、マッパエントリデータ220a-dのMSBは、ルートインデックスが含まれているか否かの表示を与えることができる。

20

## 【0038】

マッパ106a-dと関連してパイプライン208を用いることにより、異なるキー210による最長一致プレフィックステーブル200の複数サーチを並行して行なうことができる。パイプライン208は、40ビットのルックアップテーブル200の複数のサーチを、必要に応じて、他のマッパ106a-dの各々のサーチが完了するまで、40ビットのキー210と関連の各マッパ106a-dごとにマッパエントリデータ220a-dを記憶することによって並行して行なって、40ビットキー210に対応するルートインデックスを見出すのを可能にする。したがって、ダイレクト・マップト・マッパ106aへの単一のメモリアクセスを行なうことにより、受信したIPアドレスに対応するルートインデックスに対するサーチリクエストがルックアップテーブル200に発せられる。別のキーに対応するルートインデックスに対するその後のサーチは、ダイレクト・マップト・マッパ106aのための次のメモリアクセスサイクルの中でルックアップテーブル200に発せられ得る。

30

## 【0039】

図4は、図3に示されたダイレクト・マップト・マッパ106aに記憶可能なマッパエントリのタイプを図示する。図2Bに示されたバイナリツリー中のいずれのノードに対するマッパエントリも、ノーエントリ300、ルートエントリ302またはサブツリーエントリ記述子304を記憶することができる。マッパエントリ300、302、304の各タイプは、サブツリーフラグ306を含む。サブツリーフラグ306の状態は、マッパエントリがサブツリーエントリ記述子304であるか否かを示す。サブツリーフラグ306が‘1’にセットされれば、マッパエントリはサブツリーエントリ記述子304であり、サブツリーインデックス312を含む。サブツリーインデックス312は、次のノンダイレクト・マップト・マッパ106b-dに記憶されるサブツリーエントリ記述子304のアドレスである。サブツリーエントリは図4と関連して後述される。サブツリーフラグ306が‘0’ならば、ノーエントリフラグ314をチェックして、マッパエントリがノーエントリ300であるかまたはルートエントリ302であるかを判定する。ノーエントリフラグ314が‘0’ならば、エントリはノーエントリ300である。ノーエントリフラグ314が‘1’ならばエントリはルートエントリ302であり、ルートインデックスフィールド310中に、キー104と関連のルートインデックス102(図3)を記憶する。

40

50

マルチプレクサ 108 (図 2A) は、サブツリーフラグ 306 を用いて、ルートインデックス 102 (図 3) を含むマッパエントリデータ 220a-d を選択する。

#### 【0040】

図 5 は、図 2B に示されたマッパレベル\_2\_112b 中のノードに対応するマッパ 106b を図示する。マッパ 106b は、サブツリーメモリ 400、マッパアドレス論理 402 およびサブツリーマッパ 418 を含む。マッパ 106a に記憶されるキー 210a の最初の部分によって選択されたサブツリーラインデックス 312 は、サブツリーメモリ 400 に転送される。サブツリーメモリ 400 は、サブツリーラインデックス 312 が選択するサブツリーエントリ 404 を含む。サブツリーエントリ 404 はデータフィールド 406 およびポインタフィールド 408 を含む。

10

#### 【0041】

図 2B に戻って、サブツリーエントリ 404 は、マッパレベル\_2\_112b に示されるサブツリーのうちの 1 つのボトムレベルに対応する。マッパレベル\_2\_112b が 8 つのレベルを有すれば、各サブツリー (図示せず) のボトムレベルは、各ノードに 1 つの、最大 256 のルートを有する。

#### 【0042】

続いて図 5 で、サブツリーエントリ 404 は、サブツリーのボトムレベル上の各ノードに対応する 256 の可能なルートインデックス 102 (図 3) へのアクセスを提供する。ルートインデックス 102 (図 3) はサブツリーマッパ 418 に記憶される。256 の可能なルートインデックスへのアクセスを提供するため、稠密サブツリー記述子がデータフィールド 406 に記憶される。データフィールド 406 は 256 ビット幅であり、サブツリーのボトムレベルで各ノードごとに 1 ビットを与える。データフィールド 406 は、図 6A および図 6B と関連してより詳細に後述される。ポインタフィールド 408 は 256 ビット幅であり、16 の 16 ビットポインタの記憶を可能にする。各ポインタは、サブツリーマッパ 418 中に、連続した 16 のマッパエントリに対するベースアドレスを記憶して、256 のルートインデックスへのアクセスを与える。したがって、ポインタフィールド 408 は、サブツリーのボトムレベルの各ノードごとにサブツリーマッパ 418 中のマッパエントリへのポインタを間接的に与えることができる。ポインタフィールド 408 は図 6 と関連してより詳細に説明される。

20

#### 【0043】

データフィールド 406 中の稠密サブツリー記述子に記憶されたサブツリーデータ 412 とポインタフィールド 408 に記憶されたサブツリーポインタ 414 とはマッパアドレス論理 402 に転送される。マッパアドレス論理 402 は、キー 210b の次の部分 (次の 8 ビット) も受ける。

30

#### 【0044】

マッパアドレス論理 402 は、キー 212b の次の 8 ビットに依存するサブツリーのボトムレベルのノードと関連のマッパエントリ、すなわちサブツリーデータ 412 およびサブツリーと関連のサブツリーポインタ 414 のマッパアドレス 416 を判定する。マッパアドレス 416 は、サブツリーマッパ 418 中のマッパエントリを選択する。サブツリーマッパ 418 は、ダイレクト・マップト・マッパ 106a に対して、図 4 と関連して説明されたのと同じタイプのマッパエントリを含む。マッパデータエントリ 220b の中身が、その後のサーチが必要であるか否かを決定する。マッパエントリデータ 220b が、次のマッパレベル 112c (図 2B) 中に別のサブツリーエントリ 404 が存在することを示すサブツリーラインデックス 312 (図 4) を含めば、その後のサーチが必要である。

40

#### 【0045】

キー 210b の第 2 の部分は、選択されたサブツリーのボトムレベル中のノードを選択する。サブツリーポインタ 414 は、サブツリー中のノードと関連のベースアドレスを選択し、サブツリーデータ 412 は、ベースアドレスと関連のマッパエントリのブロック内のオフセットを選択する。マッパアドレス論理 402 は、図 7 と関連して後述される。

#### 【0046】

50

図 6 A はサブツリーのバイナリツリー表示である。示されたサブツリーは 5 つのレベルを含む。サブツリーは、3 つのルートインデックス  $r_1$ 、 $r_2$  および  $r_3$  ならびに 2 つのサブツリーインデックス  $s_0$  および  $s_1$  を含む。サブツリーのボトムレベルには 32 個のノード  $500^1 - 500^{32}$  が存在する。ボトムレベルの各ノード  $500^1 - 500^{32}$  と関連のルートインデックスまたはサブツリーインデックスが以下の表 1 に示される。

【0047】

【表 1】

サブツリー・ビット	ルート / サブツリー	
00000	$r_0$	10
00001	$r_0$	
00010	$r_0$	
00011	$r_0$	
00100	$r_1$	
00101	$r_1$	
00110	$r_0$	
00111	$r_0$	
01000	$r_2$	
01001	$s_0$	
01010	$r_2$	
01011	$r_2$	20
01100	$r_2$	
01110	$r_2$	
01111	$r_2$	
1xxxx	$r_3$	

表 1

【0048】

図 6 B は、図 6 A に示されたサブツリーのボトムレベル中のノードに対応する、図 5 に示されたデータフィールド 406 に記憶された稠密サブツリー記述子を図示する。データフィールド 406 は、図 6 A に示されたサブツリーのボトムレベル中の各ノード  $500$  ごとに 1 ビットの、32 ビットを含む。データフィールド 406 中のビット  $502^1 - 502^{32}$  は以下のように割当てられる。前のノードのルートインデックスを用いるならば、データフィールド 406 中のビットを '0' にセットし、サブツリーマッパ 418 (図 5) に記憶された次のルートインデックスを用いるならば、'1' にセットして次のマッパエンタリアドレスを増分する。ルートが特定されなければ、データフィールド 402 中の最初のビットは、マッパエンタリ  $504^1$  に記憶されたデフォルトルート  $r_0$  を選択する。特定されたルートがないため、こうして、ビット  $502^1$  を '0' にセットしてデフォルトルートを選択する。マッパエンタリ  $504^1$  に記憶されたデフォルトルート  $r_0$  は、次の 3 つのノード  $500^2 - 500^4$  に対して選択され、すなわち、対応するビット  $502^2 - 500^4$  はデータフィールド 406 中で '0' にセットされて、 $502^1$  が用いた前のルートインデックスを用いる。ノード  $500^5$  ではルート変更がある。

【0049】

マッパエンタリ  $504^2$  に記憶される、ノード  $500^5$  に用いられるルート  $r_1$  はノード  $506^6$  と共有される。したがって、ビット  $502^5$  は、ルート変更を示す '1' であり、サブツリーマッパ 418 (図 5) 中のマッパエンタリ  $504^2$  を選択する。ビット  $502^6$  は、 $502^5$  に記憶されるルートインデックスをこのノードに用いるべきであることを示す '0' である。ノード  $500^7$  に対してはいずれのルートも与えられないため、ルート変更があり、デフォルトルート  $r_0$  を記憶するサブツリーマッパ 418 (図 5) 中のマッパエンタリ  $504^3$  を必要として、ビット  $502^7$  に '1' が記憶される。

【0050】

ノード $500^8$ は前のノード $500^7$ と同じルートを共有し、サブツリーマッパ418(図5)で新たなマッパエントリは必要ない。ノード $500^8$ に対応するビット $502^8$ は‘0’にセットされる。ノード $500^9$ は前のノード $500^8$ とは異なるルートを有し、サブツリーマッパ418(図5)で新たなマッパエントリが必要である。ノード $500^9$ に対応するビット $502^9$ は‘1’にセットされ、次の連続するメモリ場所中のサブツリーマッパ418(図5)に、 $r_2$ を記憶するマッパエントリ $504^4$ が加えられる。

#### 【0051】

ノード $500^{10}$ は、前のノード $500^9$ とは異なるルートを有し、サブツリーマッパ418(図5)で新たなルートエントリが必要である。ノード $500^{10}$ に対応するビット $502^{10}$ は‘1’にセットされ、 $s_0$ を記憶するマッパエントリ $504^5$ が、次の連続するメモリ場所中のサブツリーマッパ418(図5)に加えられる。10

#### 【0052】

ノード $500^{11}$ は、前のノード $500^{10}$ とは異なるルートを有し、サブツリーマッパ418(図5)で新たなマッパエントリが必要である。ノード $500^{11}$ に対応するビット $502^{11}$ は‘1’にセットされ、 $r_2$ を記憶するマッパエントリ $504^6$ が、次の連続するメモリ場所中のサブツリーマッパ418(図5)に加えられる。

#### 【0053】

ノード $500^{12}$ および $500^{13}$ は、前のノード $500^{11}$ と同じルートを共有し、サブツリーマッパ418(図5)で新たなマッパエントリは必要ない。ノード $500^{12}$ に対応するビット $502^{12}$ およびノード $500^{13}$ に対応するビット $502^{13}$ はデータフィールド406中で‘0’にセットされる。20

#### 【0054】

ノード $500^{14}$ は前のノード $500^{13}$ とは異なるルートを有し、サブツリーマッパ418(図5)で新たなマッパエントリが必要である。ノード $500^{14}$ に対応するデータフィールド406中のビット $502^{14}$ は‘1’にセットされ、 $s_1$ を記憶するマッパエントリ $504^7$ がサブツリーマッパ418(図5)に加えられる。ノード $500^{15}$ は前のノード $500^{14}$ とは異なるルートを有し、サブツリーマッパ418(図5)で新たなマッパエントリが必要である。ノード $500^{15}$ に対応する、データフィールド中のビット $502^{15}$ は‘1’にセットされ、サブツリーマッパ418(図5)に、 $r_2$ を記憶するマッパエントリ $504^8$ が加えられる。ノード $500^{16}$ は前のノード $500^{15}$ と同じルートを共有し、サブツリーマッパ418(図5)で新たなマッパエントリは必要ない。30

#### 【0055】

ノード $500^{17}$ は前のノード $500^{16}$ とは異なるルートを有し、サブツリーマッパ418(図5)で新たなマッパエントリが必要である。ノード $500^{17}$ に対応する、データフィールド406中のビット $502^{17}$ は‘1’にセットされ、 $r_3$ を記憶するマッパエントリ $504^9$ がサブツリーマッパ418(図5)に加えられる。

#### 【0056】

ノード $500^{18}$  -  $500^{32}$ はすべてノード $500^{17}$ と同じルートを共有し、サブツリーマッパ418(図5)で新たなマッパエントリは必要ない。対応するビット $502^{18}$  -  $502^{32}$ は‘0’にセットされる。したがって、32個のノード $500^1$  -  $500^{32}$ について、ルートエントリ302(図4)またはサブツリーエントリ304(図4)を記憶するには、マッパエントリ $504^{1-9}$ が9つ必要である。40

#### 【0057】

ノード $500^1$  -  $500^{32}$ に対応するマッパエントリ $504^1$  -  $504^9$ は、データフィールド406に記憶される稠密サブツリー記述子に記憶される‘1’の数を計算することにより、サブツリーマッパ418(図5)中でインデックス付けされる。たとえば、ノード $500^{28}$ に対応するマッパエントリ $504^1$  -  $504^9$ を見出すには、データフィールド406のビット $502^1$  -  $502^{28}$ に記憶される‘1’の数をカウントする。‘1’の数が8であると、対応するマッパエントリはデフォルトルートから8番目の場所、すなわち、マッパエントリ $504^9$ である。50

## 【0058】

ルート変更がある場合にのみマッパエントリを記憶することにより、サブツリー・マッパ418(図5)中のサブツリー当りのマッパエントリ $504^1 - 504^9$ の数が減少する。

## 【0059】

図7は図5に示されたポインタフィールド408を図示する。ポインタフィールド408は、サブツリー・マッパ418(図5)中の連続する16のマッパエントリ場所 $504^1 - 504^{16}$ (図6B)のブロックのベースアドレスを記憶するためのブロックベースアドレスフィールド $600^1, 600^2$ を含む。連続する16のマッパエントリのブロック $602^1, 602^2$ 中のサブツリー・マッパ418(図5)にメモリが割当てられる。8レベルのサブツリーは異なるルートを256まで有することができ、256のすべてのルートを記憶するためには、ブロック $602^1, 602^2$ が16個必要である。必要なブロック $602$ の数は、サブツリーに対するルートの数に依存する。ブロックベースアドレス $602^1, 602^2$ をブロックベースアドレス(図示せず)のフリーリストから除くことにより、ブロック $602$ を特定のサブツリーに割当てる。メモリにアドレスのフリーリストを与えるための方法は技術分野では周知である。10

## 【0060】

16のマッパエントリ $504^{1-16}$ のメモリブロックを割当てることにより、サブツリー・マッパ418(図5)中のメモリの扱いがより容易になる。なぜなら割当てられた16個の場所は連続しているからである。

## 【0061】

図8は、図5に示されたマッパアドレス論理402を図示する。マッパアドレス論理402は、オフセット論理700、ポインタ論理702およびアダー論理704を含む。オフセット論理700はノードセレクト論理706および1カウント論理708を含む。ポインタ論理はベースアドレスセレクト論理710を含む。20

## 【0062】

ノードセレクト論理706は、キー $210b$ の8ビットに対応する、サブツリーデータ412中のノード500(図6B)を選択する。対応するノード番号はノードセレクト718上を1カウント論理708に転送される。1カウント論理708は、サブツリーデータフィールド406に記憶された‘1’の数を、選択されたノード500に対応するビットまでカウントする。1の数はブロックセレクト712上をポインタ論理702へおよびブロックオフセット714上をアダー論理704へ転送される。30

## 【0063】

8ビット幅のカウントフィールドを必要とする256ビットのサブツリーデータフィールド406に記憶される‘1’は最大で256個であり得る。8ビットカウントフィールドは2つのフィールドに分割され、4つのMSBがブロックセレクト712を与え、4つの最下位ビット(LSB)がブロックオフセット714を与える。

## 【0064】

たとえば、8ビットキー $210b$ が‘01000100’である場合、ノード番号68を選択し、サブツリーデータ412の最初の68ビットに記憶される‘1’が27個存在すれば、カウントは1C Hex(00011100)、MSB(0001)である。すなわち、ブロックセレクト714、セレクトブロック $602^1$ (図6)およびLSB(1100)である。すなわち、ベースブロックオフセットセレクトマッパエントリ $504^{11}$ (図6)であり、すなわちブロック $502^1$ 中の12番目のエントリである。40

## 【0065】

ベースアドレスセレクト論理710は、オフセット論理700から転送されるブロックセレクト712に依存して、サブツリー・ポインタ414からベースアドレス716を選択する。アダー論理704は、オフセット論理700から転送されるブロックオフセット714をベースアドレス716に加算し、マッパアドレス416を与える。マッパアドレス416はマッパ $106b - d$ 中のマッパエントリ $504$ (図6B)のインデックスである。

## 【0066】

図9は、この発明の原則に従う、ルックアップテーブル200中のキー210(図3)に対して最長一致プレフィックスをサーチするためのステップを図示するフローチャートである。

【0067】

ステップ800で、キー210a(図3)の最初の部分がマッパ106aへのインデックスとして転送される。処理はステップ802に継続する。

【0068】

ステップ802で、キー210a(図3)の最初の部分によってインデックス付けされた第1のレベルのマッパ中のマッパエントリ504(図6B)に記憶されたマッパエントリデータ220a(図3)は、次のレベルのさらなるサーチが必要であるか否かを判定する。必要ならば、処理はステップ804に継続する。必要なければ、第1のレベルのマッパ中のインデックス付けされたマッパエントリ504(図6B)中のルートエントリ302(図4)が、キーに対して、対応する最長プレフィックスルートを記憶し、処理はステップ808に継続する。

10

【0069】

ステップ804で、次のレベルのマッパ106b-dがサーチされる。次のレベルのマッパのためのインデックスは、前のレベルのマッパ中のインデックス付けされたマッパエントリ504(図6B)中のサブツリーエントリ記述子304(図4)に記憶されたサブツリーインデックス312と、キー210b-dの次の部分とに依存する。処理はステップ806に継続する。

20

【0070】

ステップ806で、次のレベルのマッパ中のインデックス付けされたマッパエントリ504(図6B)は、キーに対する、対応する最長プレフィックスルートインデックスまたはさらなるサーチが必要であることを示すサブツリーインデックスを記憶する。さらなるサーチが必要ならば、処理はステップ804に継続する。必要なければ、処理はステップ808に継続する。

【0071】

ステップ808で、マッパ106a-dのうち1つの中のマッパエントリ504(図6B)に記憶されたルートインデックス102(図3)は、ルートインデックス102(図3)としてルックアップテーブル200から転送される。処理は完了する。

30

【0072】

深度拡張(depth expansion)

図3に示されたルックアップテーブル200に記憶可能なルートインデックス102(図3)の数は、サブツリーマッパ418(図5)中の利用可能なマッパエントリ504(図6B)の数によって限定される。たとえば、各々のサブツリーマッパ418(図5)が128Kのマッパエントリを含みかつルックアップテーブル中にサブツリーマッパ418(図5)が2つ存在すれば、ルックアップテーブル200中に最大で256Kのルートインデックス102(図3)を記憶可能である。128Kのマッパエントリを有するサブツリーマッパ418(図5)は17ビットのインデックスを必要とする。512Kのマッパエントリを有するサブツリーマッパ418(図5)は19ビットのインデックスを必要とする。ルックアップテーブル200中の2つの512Kサブツリーマッパ418(図5)は、32ビットIPv4デスティネーションアドレスのために、可能な40億個のルートインデックス102(図3)のうち100万個のための記憶を提供する。

40

【0073】

ルートインデックス102(図3)を記憶するためのマッパエントリ504(図6B)の数は、複数のルックアップテーブル200を設けることによって増やすことができる。複数のルックアップテーブルは、ルックアップテーブル200のうち1つの中のサブツリーマッパ418(図5)のマッパエントリ504(図6B)に記憶された、サーチキー210に対応する値を求めて並行してサーチされる。

【0074】

50

図 10 A は深度拡張のための実施例を図示する。2 つのルックアップテーブル、すなわちマスタルックアップテーブル 200 a およびスレーブルックアップテーブル 200 b が示される。しかしながら、ルックアップテーブルの数は示された 2 つに限定されるのではなく、1 つよりも多くのスレーブルックアップテーブル 200 b を加えることができる。

#### 【 0075 】

ルックアップテーブル 200 a - b の各々は、同じサーチキー 210 によって並行してサーチされる。サーチキー 210 に対応するルートインデックス 102 (図 3) は、ルックアップテーブル 200 a - b のうち 1 つの中のサブツリーマッパ 418 (図 5) またはルックアップテーブル 200 a、200 b の両者の中のダイレクトマッパ 106 a に記憶される。最終ルートインデックス 900 は、両者のルックアップテーブル 200 a、200 b の並行したサーチの後に見出される。 10

#### 【 0076 】

図 10 B は、図 10 A に示された実施例のルックアップテーブル 200 a の 1 つを図示する。ルックアップテーブル 200 a - b の各々は、図 3 と関連してルックアップテーブル 200 について既に説明されたようなマッパ 106 a - d およびパイプライン 208 ならびにドライバ論理 902 を含む。ルックアップテーブル 200 a は、サーチキーに対応するルートインデックスを求めて、マッパ 106 a - d 中の複数レベルサーチを行なう。各レベルのサーチ結果は、マッパエントリデータ 220 a - d 上をパイプライン 208 に転送される。パイプライン 208 は、サーチ 904 の結果をドライバ論理 902 に転送する。ルックアップテーブル 200 a - b の各々の中のマッパ 106 a のマッパエントリ 50 20 4 (図 6 B) はサブツリーエントリ 304 (図 4) を記憶するが、ルートエントリ 302 (図 4) は、マスタルックアップテーブル 200 a 中のマッパ 106 a にしか記憶されない。ルートエントリ 302 の代わりに、スレーブルックアップテーブル 200 b 中のマッパ 106 a のマッパエントリ 504 (図 6 B) にノーエントリ 300 (図 4) が記憶される。マッパ 106 a 中でただ 1 つのルックアップテーブルにルートインデックスを配することにより、ルックアップテーブルを選択して最終ルートインデックス 900 を与えることを回避する。この結果、スレーブルックアップテーブル 200 b 中のメモリ 64 K が与えられる。これは、ルートインデックスを記憶するのに用いることはできないが、図 3 と関連して説明されたように、同じルックアップテーブルをマスタルックアップテーブルとしてまたはスレーブルックアップテーブルとして構成できるようにする。代替的な実施例では、マッパ 106 a を有しないスレーブルックアップデバイスを設けることができる。ルートインデックス 102 (図 3) をマッパ 106 a 中のルートエントリ 302 (図 4) に記憶すれば、サーチは、マスタルックアップテーブル 200 a 中のマッパ 106 a で終了する。 30

#### 【 0077 】

図 10 A に示されたように、マスタルックアップテーブル 200 a およびスレーブルックアップテーブル 200 b は最終ルートインデックス 900 を共有する。最終ルートインデックス 900 を記憶するルックアップテーブル 200 a、200 b はルートインデックス 102 (図 3) を与える。ルックアップテーブル 200 a、200 b の各々が別個のデバイスならば、最終ルートインデックス 900 を共有することにより、各デバイスの外部ピンのカウントが減少する。いかなるときにも、最終ルートインデックス 900 を駆動するのは、ルックアップテーブル 200 a、b のうち 1 つだけである。 40

#### 【 0078 】

サーチキー 210 と関連のルートインデックスをルックアップテーブル 200 a、200 b の両者に記憶して、ルックアップテーブルの両者が同時に最終ルートインデックス 900 を駆動するエラー状態を回避するため、各ルックアップテーブル 200 a、b はデバイスコード 906 を記憶する。3 ビットのデバイスコードにより、拡張ルックアップテーブルが 8 つのデバイスを含むことが可能になる。

#### 【 0079 】

ドライバ論理 902 は、サーチ 904 の結果がルートインデックス 102 (図 3) を含む 50

か否かを判定する。含んでいれば、ルックアップテーブル 200a 中のドライバ論理 902 は、バスリクエスト信号（図示せず）上に、最終ルートインデックス 900 を駆動する意図を信号で伝える。2つ以上のルックアップテーブル 200a、200b が同時にルートインデックス信号を駆動する意図を信号で伝えれば、ルートインデックスは、ルックアップテーブル 200a、200b により、最も低いデバイスコードを与えられる。バスリクエスト信号を用いることによりバス競合を解決するための方法は技術分野で周知である。

#### 【0080】

図 10C は、サーチキー 210 に対応する値を記憶するのに利用可能なマッパエントリの数を増やすために深度拡張を与える別の実施例を図示する。図 10C に示される実施例では、2つのルックアップテーブル 200c-d、すなわちマスタルックアップテーブル 200c およびスレーブルックアップテーブル 200d が値を記憶するために設けられる。しかしながら、ルックアップテーブルの数は示された2つに限定されるものではなく、マッパエントリの数は、より多くのスレーブルックアップテーブル 200d を加えることによって増やすことができる。サーチキー [39:0] 210 に対応するルックアップテーブル 200c-d のうち1つの中のマッパエントリに記憶される値に対するサーチは、ルックアップテーブル 200c-d で並行して行なわれる。

#### 【0081】

図 10D は、図 10C に示された実施例のスレーブルックアップテーブル 200d を図示する。各々のルックアップテーブルは、図 3 と関連してルックアップテーブル 200 について説明されたような、マッパ 106a-d を含む。ルックアップテーブル 200c-d の各々の中のマッパ 106a のマッパエントリは、サブツリーエントリ 304（図 4）を記憶する。各々のルックアップテーブル c-d は、マッパ 106a 中のマッパエントリ 504（図 6B）に記憶されたサブツリーエントリ記述子 304（図 4）に記憶されたサブツリーアインデックス 312 を次のマッパ 106b-d に転送する。しかしながら、ルートインデックス 102（図 3）は、マスタルックアップテーブル 200c 中のマッパ 106a にのみ記憶される。スレーブルックアップテーブル 200d 中のマッパ 106a にノーエントリを記憶して、キーに対応するルートインデックスを1つよりも多くのルックアップテーブル 200b、200d に記憶しないようにする。

#### 【0082】

スレーブルックアップ 200d の複数レベルのサーチ 904 の結果は最終インデックス論理 1004 に転送される。最終インデックス論理 1004 は、複数レベルサーチ 904 の結果およびマスタルックアップテーブル 200c から転送された入結果（incoming result）1000a を出結果（outgoing result）1002a として転送する。ルートインデックス 102（図 3）が複数レベルサーチ 904 の結果に含まれれば、複数レベルサーチの結果が出結果 1002a として転送される。ルートインデックス 102（図 3）が入結果 1000a に含まれれば、入結果 1000a が出結果 1002a として転送される。ルートインデックス 102（図 3）が入結果 1000a または複数レベルサーチ 904 の結果のいずれにも含まれなければ、複数レベルサーチ 904 の結果が出結果 1002a として転送される。

#### 【0083】

図 10C に示されるように、マスタルックアップテーブル 200c およびスレーブルックアップテーブル 200d は、入結果 1000a で標識付けられた共通のバスを介して接続される。ルートインデックス 102（図 3）は、出結果 1002a の上を、スレーブルックアップテーブル 200d から転送されるのみである。1つよりも多くのスレーブルックアップテーブル 200d が存在する場合、最後のスレーブルックアップテーブルにより、拡張ルックアップテーブルに対するルートインデックス 102（図 3）が与えられる。この実施例は、図 10A と関連して説明されたマルチドライバ最終ルートインデックス 900 の実現例を回避するが、入結果 1000a 用に、より多くのデバイス外部ピンが必要である。

10

20

30

40

50

## 【0084】

図11A-Bは、ルックアップテーブル200a-b(図10A)または200c-d(図10C)中の、図2Bに示されたルートの分布のバイナリツリー表示を図示する。

## 【0085】

図11Aは、マスタルックアップテーブル200a(図10A)または200c(図10C)に記憶されたルートのバイナリツリー表示を図示する。図2Bに示されたルートのバイナリツリー表示に示されるサブツリーBは、マスタルックアップテーブル200aには含まれない。ノード $130^1 - 130^2$ および $130^{24-32}$ は、図3と関連して説明されたように、ルックアップテーブル200a中のマッパ106aでコード化される。サブツリーBがインデックス付けされるノードは、それがマスタルックアップテーブル200aに記憶されれば、Xでグラフィックに表わされ、ブルーニングされたサブツリーを示す。マスタルックアップテーブル200a中のノード $130^3$ に対応するマッパエントリ504(図6B)は、もはやサブツリーBへのサブツリーインデックス312(図4)を記憶しない。代わりに、ノード $130^3$ に対応するマッパエントリが別のスレーブルックアップテーブル200b中のサブツリーマッパ418(図5)に記憶されることを示すマスタルックアップテーブル200a中の、ノード $130^3$ に対応するマッパエントリ504(図6B)に、ノーエントリ300(図4)が記憶される。

## 【0086】

図11Bは、スレーブルックアップテーブル200b(図10A)または200d(図10C)中のサブツリーマッパ418(図5)中のマッパエントリ504(図6B)に記憶されるルートのバイナリツリー表示を図示する。スレーブルックアップテーブル200bに記憶されるルートのバイナリツリー表示は、サブツリーAが含まれないという点で、図2Bに示されたバイナリツリー表示と異なっている。したがって、図2Bと関連して説明されたように、ノード $130^1 - 130^3$ および $130^5 - 130^{32}$ がコード化される。スレーブルックアップテーブル200b中のノード $130^4$ に対応するマッパエントリ504(図6B)は、もはやサブツリーAへのサブツリーインデックス312(図4)を記憶しない。代わりに、スレーブルックアップテーブル200b中のノード $130^4$ に対応するマッパエントリは、ノード $130^4$ に対応するマッパエントリが別のルックアップテーブルに記憶されることを示すノーエントリ300(図4)を記憶する。サブツリーAへのサブツリーインデックスおよびしたがってホスト138(図11A)へのルートインデックスはマスタルックアップテーブル200aに記憶され、サブツリーBへのサブツリーインデックスおよびしたがってホスト140へのルートインデックスはスレーブルックアップテーブル200bに記憶される。スレーブルックアップテーブル200b、200dはサブツリーに対する結果のみを記憶する。すなわち、スレーブルックアップテーブル200b、200dは第1のレベルのマッパ106aの結果を記憶しない。

## 【0087】

図11Aおよび図11Bを参照して、キー210aの最初の部分による、スレーブマッパレベル\_1\_1104aまたはマスタマッパレベル\_1\_1102(図3)中のノード $130^9 - 130^{12}$ のいずれに対するサーチの結果も、マスタルックアップテーブル200a、200c中のマッパ106aのマッパエントリ504(図6B)中のルートエントリ302(図4)と、スレーブルックアップテーブル200b、200d中のマッパ106aのマッパエントリ504(図6B)に記憶されるノーエントリ300(図4)とに記憶される $r_1_116$ となる。マスタルックアップテーブル200a、200cに記憶されるルートエントリ302(図4)は、入結果1000a上をスレーブルックアップテーブル200b、200dに転送され、スレーブルックアップテーブル200b、200dにより出結果1002a上を転送される。

## 【0088】

キー210aの最初の部分による、ノード $130^4$ に対するサーチの結果は、マスタルックアップテーブル200a中のマッパ106aのマッパエントリ504(図6B)中にサブツリーエントリ記述子304(図4)に記憶されたサブツリーAに対するサブツリーイ

10

20

30

40

50

ンデックス312(図4)となる。サブツリーインデックス312はマスタルックアップテーブル200a中のマッパ106bに転送され、マスタルックアップテーブル200aに記憶されるルートエントリ302(図4)に対するサーチを継続する。

#### 【0089】

キー210の最初の部分による、ノード130<sup>23</sup>に対するサーチの結果は、マスタルックアップテーブル200a中のマッパ106aのマッパエントリ504(図6B)に記憶されるサブツリー-エントリ記述子304(図4)と、スレーブルックアップテーブル200b中のマッパ106aのマッパエントリ504(図6B)に記憶されるノーエントリ300(図4)とである。したがって、ルートエントリ302(図4)に対するサーチは、スレーブルックアップテーブル200b中のマッパ106bで、キー210bの次の部分を用いて継続する。

10

#### 【0090】

図12は、図10Aに示されたルックアップテーブル200a-b中のマッパエントリ504(図6B)に記憶されるべきルートエントリ302(図4)を分散するための方法を図示するフローチャートである。同じ方法は、図10Cに示されたルックアップテーブル200c-dに当てはまる。マッパエントリに記憶されるべきルートエントリ302(図4)はまず、ルックアップテーブル200a-bに記憶される前に、プロセッサ(図示せず)によってメモリに記憶される。

#### 【0091】

ルートエントリ302(図4)がメモリに記憶される間、ルックアップテーブル200a-b(図10A)の各々に記憶されるべきルートエントリ302(図4)の数がカウントされる。マッパレベル\_1\_1104a(図11B)に対するルートエントリ302(図4)は、ルックアップテーブル200a中のマッパ106aに記憶される。マッパレベル\_1\_1104a(図11B)に対するサブツリー-エントリ(図4)は、ルックアップテーブル200a-200bの各々の中のマッパ106aに記憶される。

20

#### 【0092】

ステップ1200で、ルックアップテーブル200a-200b(図10A)の各々の中のマッパ106aのサブツリー-エントリ304(図4)の各々ごとの、記憶されるべきルートエントリ302(図4)の数を計算して、どのようにルートエントリ302(図4)をルックアップテーブル200a-b(図10A)の中に分散すべきかを判定する。ルートエントリ302(図4)を記憶するのに必要なマッパエントリ504(図6B)の総数を判定した後、処理はステップ1202に継続する。

30

#### 【0093】

ステップ1202で、サブツリーに対して記憶されるべきマッパエントリ504(図6B)の総数を、ルックアップテーブル200a-b(図10A)の数で除算して、各ルックアップテーブル200a-b(図10A)に記憶すべきルートエントリ302(図4)の数を定める。処理はステップ1204に継続する。

#### 【0094】

ステップ1204で、ルートエントリ302(図4)は、選択されたルックアップテーブル200a-b中のサブツリーマッパ418(図5)中のマッパエントリ504(図6B)に記憶される。処理はステップ1206に継続する。

40

#### 【0095】

ステップ1206で、選択されたルックアップテーブル200a-b(図10A)中のサブツリーマッパ418(図5)のマッパエントリ504(図6B)に記憶されたルートエントリの数が1/n未満ならば、なおここでnは利用可能なルックアップテーブル200a-b(図10A)の数であるが、プロセスはステップ1204に続く。それ以上ならば、選択されたルックアップテーブル200a-bはマッパエントリの総数の1/nを記憶しており、処理はステップ1208に継続する。

#### 【0096】

ステップ1208で、選択されたルックアップテーブル200a-bはマッパエントリの

50

総数の  $1/n$  を記憶し、選択されたルックアップテーブル 200a - b のいずれの残余のサブツリーノードに対しても、ノーエントリ 300 (図 4) を記憶する。それぞれのサブツリーに対するルートインデックスは、現在選択されたルックアップテーブルに記憶されないからである。処理はステップ 1210 に続く。

【0097】

ステップ 1210 で、すべてのルートエントリが記憶されれば、処理は完了する。記憶されていなければ、処理はステップ 1212 に継続する。

【0098】

ステップ 1212 で、次のルックアップテーブル 200a - b (図 10A) が選択される。処理はステップ 1204 に続く。

10

【0099】

ルートエントリは、IP アドレスに対応するルートインデックスに対するサーチの前に、ルックアップテーブル 200a - b (図 10A) の中に分散される。サーチは、ルックアップテーブル 200a - b (図 10A) の各々の中で並行して行なわれる。ルックアップテーブルの各々の中で並行して実現されるサーチのための方法が、ルックアップテーブル 200a - b (図 10A) の 1 つのために説明される。

【0100】

図 13 は、図 10C に示されたルックアップテーブル 200c - d のいずれの 1 つにも記憶されるサーチキーに対応する値に対する、サーチキーによるサーチのための方法を図示するフローチャートである。

20

【0101】

ステップ 1300 で、ルックアップテーブル 200c - d (図 10C) の各々はサーチキー 210 を受ける。ルックアップテーブル 200c - d の各々の中のマッパ 106a は、キー 210a の最初の部分に対応する値を求めてサーチされる。処理はステップ 1302 に継続する。

【0102】

ステップ 1302 において、マッパ 106a 内のマッパエントリ 504 (図 6B) に記憶されたエントリが読み出される。マスタルックアップテーブル 200c 中のマッパエントリ 504 (図 6B) は、ノーエントリ 300 (図 4)、ルートエントリ 302 (図 4) またはサブツリーエントリ記述子 304 (図 4) を記憶し得る。スレーブルックアップテーブル 200d 内のマッパエントリ 504 (図 6B) は、ノーエントリ 300 (図 4) およびサブツリーエントリ記述子 304 (図 4) を記憶し得る。それぞれのルックアップテーブル 200 内のマッパエントリがルートエントリ 302 (図 4) を記憶していれば、そのエントリは有効な値であり、ルックアップテーブル 200c - 200d 内の後続のマッパ 106b - d のさらなるサーチは必要はなく、処理はステップ 1310 へ進む。そうでなければ、処理はステップ 1304 へ進む。

30

【0103】

ステップ 1304 において、このエントリがサブツリーエントリ記述子 304 (図 4) を記憶していれば、ルックアップテーブル 200c - d 内のさらなるサーチが必要となり、処理はステップ 1306 へ進む。そうでなければ、このエントリはノーエントリを記憶しており、これがさらなるサーチが必要でないことを示しているので、処理はステップ 1310 へ進む。

40

【0104】

ステップ 1306 において、選択されたサブツリーにおいてサーチが継続される。キー 210b - d の次の部分および前のレベルのサーチの結果得られたサブツリーインデックス 312 (図 4) に依存して、次のレベルのマッパ 106b - d (図 3) がサーチされる。処理はステップ 1308 へ進む。

【0105】

ステップ 1308 において、現在のレベルのマッパ 106b - d のサーチの結果得られるマッパエントリに依存して、サーチを継続するか否かが判定される。マッパエントリ 50

50

4 (図6B) がサブツリーエントリ記述子304 (図4) を記憶していれば、次のレベルのマッパ106b-dでサーチが継続され、処理はステップ1306へ進む。マッパエントリ504 (図6B) がサブツリーエントリ記述子304 (図4) を記憶していなければ、さらなるサーチは必要なく、処理はステップ1310へ進む。

#### 【0106】

ステップ1310において、サーチの結果が別のルックアップテーブルから受けた入結果1000aと比較される。たとえば、ルックアップテーブルがスレーブルックアップテーブル200dの場合、マスタルックアップテーブル200cでのサーチからの入結果は、入結果1000a上でルックアップテーブル200dへ転送され、スレーブルックアップテーブル200dにおけるサーチの結果と比較される。処理はステップ1312へ進む。

10

#### 【0107】

ステップ1312において、入結果1000aと現在のルックアップテーブル200dのサーチ結果が異なる場合、処理はステップ1314へ進む。入結果1000aと現在のルックアップテーブル200dのサーチ結果との両者が同じであれば、別々のルックアップテーブル200c-d中のマッパエントリ504 (図6B) に2つの有効な結果が記憶されている。同じキー210に対して2つの有効な結果を記憶すべきでなく、処理はステップ1316へ進む。

#### 【0108】

ステップ1314において、入結果1000aがチェックされ、それが有効であるか否かが判定される。入結果1000aは、それがルートエントリ302 (図4) であれば有効である。入結果1000aは、それがノーエントリ300 (図4) またはサブツリーエントリ記述子304 (図4) であれば無効である。サブツリーエントリ記述子304 (図4) 、ルートエントリ302 (図4) およびノーエントリ300 (図4) については図4に関連して既に説明した。入結果1000aが無効であれば、処理はステップ1318へ進む。そうでなければ、処理はステップ1320へ進む。

20

#### 【0109】

ステップ1318において、入結果1000aは有効であり、現在のルックアップテーブル200dのサーチ結果は無効である。入結果1000aは現在のルックアップテーブル200dから出結果(出て行く結果)1002a上で転送される。入結果1000aは、現在のルックアップテーブル200dが最後のルックアップテーブルであればルートインデックス102 (図3) として転送され、または、入結果1000aとして次のルックアップテーブルへ転送される。処理が完了する。

30

#### 【0110】

ステップ1316において、2つの有効な結果の値がキーに対して別々のルックアップテーブルに記憶される。ルックアップテーブル200c-d中のルートエントリの記憶中にエラーが起こる。エラーコードが生成されるのでこのエラーは修正され得る。処理が完了する。

#### 【0111】

ステップ1320において、現在のルックアップテーブル200dのサーチ結果も入結果1000aも有効でない。現在のルックアップテーブル200dのサーチ結果は、たとえ無効であっても、入結果1000aとして次のルックアップテーブルへ転送される。処理が完了する。

40

#### 【0112】

図14は、図10Aに示すルックアップテーブル200a-dのうちの1つに記憶されたサーチキーに対応する値をサーチするための方法を表わすフローチャートである。

#### 【0113】

ステップ1340において、ルックアップテーブル200aおよび200bの両者における第1のレベルのマッパ106aが、キー210aの第1の部分に対応する値についてサーチされる。処理はステップ1342へ進む。

#### 【0114】

50

ステップ1342において、キー210aの第1の部分で第1のレベルのマッパ106aをサーチした後に有効な結果の値が見つかれば、処理はステップ1352へ進む。そうでなければ、処理はステップ1344へ進む。

【0115】

ステップ1344において、キー210aの第1の部分で第1のレベルのマッパ106aをサーチした結果の値がサブツリーエントリ記述子304(図4)であれば、処理はステップ1346へ進む。そうでなければ、そのキーに対する有効な値は現在のルックアップテーブルには記憶されておらず、処理は完了する。

【0116】

ステップ1346において、有効な値のサーチは、前のレベルのマッパのサーチ中に見つかったサブツリーエントリ記述子304(図4)において識別されたサブツリーで継続される。次のレベルのマッパが、キー210b-cの次の部分および次のレベルのサーチの結果得られるサブツリーセレクトに依存する値に対してサーチされる。処理はステップ1348へ進む。

10

【0117】

ステップ1348において、このサーチ結果により、次のレベルのマッパのサーチが要求されるか否かが判定される。現在のサーチの結果得られるエントリは、ルートエントリ302、ノーエントリ300(図4)またはサブツリーエントリ記述子304(図4)を記憶し得る。このエントリがサブツリーエントリ記述子304(図4)を記憶していれば、さらなるサーチが必要であり、処理はステップ1346へ進む。このエントリがサブツリーエントリ記述子304(図4)を記憶していなければ、処理はステップ1350へ進む。

20

【0118】

ステップ1350において、このエントリがルートインデックス102(図3)を記憶していれば、処理はステップ1352へ進む。そうでなければ、このエントリは別のルックアップテーブルに記憶されている。処理が完了する。

【0119】

ステップ1352において、キーに対応する有効な値が現在のルックアップテーブルに記憶されている。この有効な値は、キーに対応するルートインデックス102(図3)として転送される。処理が完了する。

30

スペースモード

図5を参照して、サブツリーエントリ404は256までの可能なルートインデックスのアクセスを、256のノードサブツリーにおいて各ノードに1つずつ提供する。これらのルートインデックスはサブツリーマッパ418(図5)におけるマッパエントリ504<sup>1</sup>-504<sup>n</sup>に記憶される。サブツリーマッパ418(図5)内のマッパエントリ504(図6B)に対するマッパアドレス416は、データフィールド406に記憶された稠密サブツリー記述子と、サブツリーエントリ404内のポインタフィールド408に記憶されたサブツリーポインタとに依存して決定される。稠密サブツリー記述子の形式については図6A-6Bに関連して既に説明した。稠密サブツリー記述子は、256のノードサブツリーにおける各ノードに対してノードビット502(図6B)を記憶する。しかしながら、すべてのサブツリーが、256個のノードの各々に対してある別個のルートインデックスを有する。たとえば、1つのサブツリーは1つのルートインデックスのみを有し得る。

40

【0120】

図15は、第1のマッパレベル112aにおけるサブツリーエントリ304(図4)によってインデックス付けされた第2のマッパレベル112bにおける疎サブツリーAおよび稠密サブツリーBのバイナリツリーを表わした図である。マッパ106a内のs1に対するサブツリーエントリ記述子304(図4)はサブツリーAのサブツリーエントリ404に対するサブツリーアインデックス312を記憶する。マッパ106a内のs0に対するサブツリーエントリ記述子304(図4)は、サブツリーBのサブツリーエントリ404に対するサブツリーアインデックス312を記憶する。

50

## 【0121】

密に配置されたサブツリーBは11個のルートインデックス、すなわちr6からr16と、6個のサブツリーエントリ、すなわちs2からs7を有する。サブツリーBに対してルートエントリ302(図4)とサブツリーエントリ304(図4)とを記憶するマッパエントリ504(図6B)に対応するマッパアドレス416は、図6Bに関連して既に述べたように、稠密サブツリー記述子においてコード化される。

## 【0122】

疎に配置されたサブツリーAは2個のルートインデックス、すなわちr1およびr2を記憶する。これらが稠密サブツリー記述子に記憶されていれば、サブツリーエントリ404全体を用いてマッパエントリ504(図6B)に対する3つのマッパアドレス416、すなわちr0、r1およびr2が提供される。

10

## 【0123】

ルックアップテーブル200に記憶されたルートの数は、複数の疎サブツリー記述子のうちの1つにある疎サブツリーをコード化し、サブツリーエントリ404内の稠密サブツリー記述子にある密に配置されたサブツリーをコード化することによって、増加し得る。

## 【0124】

密に配置されたサブツリーは16以上のマッパエントリ504(図6B)を有し、サブツリーエントリ404内のデータフィールド406は、図6A-6Bに関して述べたように、稠密サブツリー記述子を記憶する。疎に配置されたサブツリーは15以下のマッパエントリ504(図6B)を有し、サブツリーエントリ404内のデータフィールドは複数の疎サブツリー記述子を記憶する。疎に配置されたサブツリーを疎サブツリー記述子に記憶する能力を与えることにより、サブツリーメモリ400により多くのサブツリーを記憶することができ、よって、ルックアップテーブル200中により多くのルートエントリを記憶することができる。

20

## 【0125】

図16A-Cは、図5に示すサブツリーエントリ404内のデータフィールド406およびポインタフィールド408と、図4に示すサブツリーエントリ記述子304(図4)とを、サブツリーエントリ404内に複数の疎サブツリー記述子を記憶できるよう変形した例を表わす。

## 【0126】

30

図16Aを参照して、スペースモードで構成されたサブツリーエントリ404内のデータフィールド406は、図6Bに関連して述べたサブツリーの各ノードごとに1ビットを有する稠密サブツリー記述子の代わりに、複数の疎サブツリー記述子 $1400^1 - 1400^n$ を含む。疎サブツリー記述子 $1400^1 - 1400^n$ の各々は、ノード記述子 $1402^1 - 1402^n$ を含む。ノード記述子 $1402^1 - 1402^n$ は、サブツリーにおいて完全に符号化されたルートを表わす9ビットの値である。ノード記述子 $1402^1 - 1402^n$ は、サブツリー内の単一のノードまたは複数のノードを表わす。

## 【0127】

図16Bを参照して、疎サブツリー記述子の記憶をサポートするために、モードフィールド1404がサブツリーエントリ404内のポインタフィールド408に加えられる。ポインタフィールド408は、ロックベースアドレス $600^1$ およびロックベースアドレス $600^2$ も記憶しており、各ロックが16個の割当てられたマッパアドレス416を含み、これらのアドレスは各サブツリーエントリ404ごとに合計32のマッパアドレス416を提供する。モードフィールド1404はモード値を記憶する。モードフィールド1404に記憶されたモード値は、サブツリーエントリ404に記憶された疎サブツリー記述子 $1400^1 - 1400^n$ の数と、疎サブツリー記述子 $1400^1 - 1400^n$ の各々に記憶されたノード記述子 $1402^1 - 1402^n$ の数とを示す。表2は、各モードについてのサブツリーエントリ404の構成を表わす。

40

## 【0128】

## 【表2】

50

	ノード/ サブツリー		ビット	サブツリー/ サブツリー エントリ	ルート/ サブツリー-エントリ		未使用	1サブツリー-エントリ に要求される マッパエントリ	
モード	最大	最小			最大	最小		最大	最小
0	1	1	9	16	16	16	112	32	32
1	2	2	18	10	20	20	76	30	30
2	3	3	27	8	24	24	40	32	32
3	4	4	36	6	24	24	40	30	30
4	7	5	63	4	28	20	4	32	24
5	15	8	135	2	30	16	-14	32	18

表2

## 【0129】

表2を参照して、たとえば、サブツリー-エントリ404内のポインタフィールド408におけるモードフィールド1404に記憶されたモード値が「4」にセットされる場合、サブツリー-エントリ404の疎サブツリー記述子1400の各々は5から7のノード記述子1402<sup>1</sup> - 1402<sup>n</sup>を記憶する。各ノード記述子1402<sup>1</sup> - 1402<sup>n</sup>は9ビットを記憶する。疎サブツリー記述子1400に記憶されたビットの総数は、疎サブツリー記述子1400<sup>1</sup> - 1400<sup>n</sup>ごとのノード記述子1402<sup>1</sup> - 1402<sup>n</sup>の数を9（ノード記述子1402<sup>1</sup> - 1402<sup>n</sup>ごとのビット数）で乗じることにより計算される。モード4に対して疎サブツリー記述子1400ごとのビット数を計算すると、7個のノード記述子1402を有する疎サブツリー記述子1400は63ビットを記憶している（7ノード記述子\*9ビット = 63）。

## 【0130】

サブツリー-エントリ404ごとの疎サブツリー記述子1400の数は、データフィールド406内のビット数を疎サブツリー記述子1400<sup>1</sup> - 1400<sup>n</sup>内のビット数で除することにより、算出される。モード4について、データフィールド406内のビット数は256であり、疎サブツリー記述子内のビット数は63である。したがって、疎サブツリー記述子1400<sup>1</sup> - 1400<sup>n</sup>の数は4である（int（256 / 63）= 4）。

## 【0131】

サブツリー-エントリ404ごとのノード記述子1402<sup>1</sup> - 1402<sup>n</sup>の総数は、各サブツリーごとのノードの数に各サブツリー-エントリ404ごとのサブツリーの数を乗じたものである。モード4について計算すると、サブツリー-エントリ404ごとのノード記述子1402の総数は、疎サブツリー記述子1400<sup>1</sup> - 1400<sup>n</sup>に記憶されたノード記述子1402<sup>1</sup> - 1402<sup>n</sup>が7個ある場合は28であり（7 \* 4 = 28）、疎サブツリー記述子1400<sup>1</sup> - 1400<sup>n</sup>ごとのノード記述子1402が5個ある場合は20である（5 \* 4 = 20）。

## 【0132】

表2のマッパエントリの列には、サブツリーマッパ418（図5）内のマッパエントリ504（図6B）がいくつサブツリー-エントリ404によって使用されているかが示される。マッパの値は、サブツリーごとのノードを1ずつ増分し、かつ疎サブツリー記述子のサブツリーの数で乗じることにより、算出される。サブツリーのデフォルトエントリを記憶するためには各サブツリーごとのノードの数より1以上多いマッパエントリが要求される

ので、サブツリーごとのノードは1ずつ増分される。

【0133】

表2のモード4の行を参照して、疎サブツリー記述子1400ごとに7個のノード記述子1402があれば、各サブツリーエントリ404ごとに32((7+1)\*4=32)のマッパエントリが要求され、疎サブツリー記述子1400ごとに5個のノード記述子1402があれば、疎サブツリー記述子1400ごとに24((5+1)\*4=24)のノード記述子1402が要求される。サブツリーごとのノードの数およびサブツリーエントリ404ごとのサブツリーの数は、サブツリーエントリ404ごとのノード記述子の最大数が30を超えないように選択される。これは、サブツリーマッパ418(図5)内のマッパアドレス416が16ロック刻みで割当てられるからである。ポインタフィールド408に記憶された2つのロックベースアドレス600<sup>1</sup>および600<sup>2</sup>を記憶することにより、32のマッパアドレス416が提供される。

【0134】

図16Cを参照して、サブツリーメモリ400における各サブツリーエントリ404は、図6Bに関連して述べたようなデンス(稠密)モードで、またはスパース(疎)モードで構成され得る。デンスモードについて図4に関して述べたサブツリーマッパ418(図5)に記憶されるサブツリーエントリ記述子304(図4)は、サブツリーエントリ404がデンスモードでコード化されるかまたはスパースモードでコード化されるかの指示を提供することによりスパースモードを可能にするよう変更される。この指示(indicator)は、タイプフィールド1406により提供される。

【0135】

タイプフィールド1406の状態により、サブツリーエントリ404がデンスモードまたはスパースモードかのいずれで構成されているかが示される。サブツリーエントリ404がスパースモードで構成されている場合、スパースサブツリー記述子セレクトフィールド1408に記憶された値とサブツリーインデックス314とが用いられて、スパースサブツリー記述子1400が選択される。スパースサブツリー記述子セレクト1408については、図16を参照して後により詳細に説明する。

【0136】

図17は、疎に配置されたサブツリーにおけるノードに対してマッパエントリ504(図6B)を選択するようにロックオフセット714を与えるための、図8に示すオフセット論理700のスパースモード論理1502を表わす。スパースモード論理1502は、サブツリーエントリ404中の疎サブツリー記述子1400に記憶されたノード記述子1402に依存して、ロックオフセット714を与える。オフセット論理700はまた、デンスモード論理1500も含む。デンスモード論理1500は、密に配置されたサブツリーにおけるルートに対してロックオフセット714を与えるためのノードセレクト706および1カウント論理708を含む。デンスモード論理1500については、図8に関連して既に説明した。

【0137】

タイプフィールド1406の状態がサブツリーエントリ404がスパースモードで構成されていると示す場合、サブツリーエントリ404からのサブツリーデータ412はスパースモード論理1502へ転送される。スパースモードサブツリー論理1502については図18に関連して説明する。

【0138】

図18は、図17のオフセット論理700に示されるスパースモード論理1502を表わす。スパースモード論理1502は、サブツリーセレクト論理1600、マルチプレクサ1602、連想メモリ(「CAM」)1606および変換論理1604を含む。選択されたサブツリーエントリ404内のデータフィールド406に記憶された疎サブツリー記述子1400<sup>1</sup>-1400<sup>n</sup>は、サブツリーデータ412上でオフセット論理700へ転送される。オフセット論理700は疎サブツリー記述子1400<sup>1</sup>-1400<sup>n</sup>をスパースモード論理1502におけるマルチプレクサ1602へ転送する。サブツリーデータ412中

の疎サブツリー記述子 1400<sup>1</sup>のうちの 1 つが、サブツリーセレクト論理 1600 により生成されたセレクト 1614 によって選択される。

【0139】

サブツリーセレクト論理 1600 はセレクト 1614 を生成し、前のマッパレベルにおいて選択されたマッパエントリから転送された疎サブツリー記述子セレクト 1408 の状態と、選択されたサブツリーエントリ 404 内のポインタフィールド 408 に記憶されたモード 1404 とに依存して、疎サブツリー記述子 1400<sup>1</sup>を選択する。表 3 は、モード 4 サブツリーエントリ 404 について、マルチプレクサ 1602 から、選択された疎サブツリー記述子 1610 上で転送された、選択された疎サブツリー記述子 1400<sup>1</sup>およびそれぞれのサブツリーデータビット 412 を表わす。表 2 のモード 4 の行を参照して、モード 4 サブツリーエントリ 404 には 4 つの疎サブツリー記述子が記憶され得る。4 つの疎サブツリー記述子 1400 の各々は 63 ビットであり、7 から 5 個のノード記述子 1402 を記憶し得る。よって、これら 4 つの疎サブツリー記述子 1400 の各々は 63 ビット境界から始まる。第 1 の疎サブツリー記述子 1400<sup>1</sup>はデータフィールド 406 のビット 62:0 に記憶される。第 2 の疎サブツリー記述子 1400<sup>2</sup>はデータフィールド 406 のビット 125:63 に記憶される。第 3 の疎サブツリー記述子 1400<sup>3</sup>はデータフィールド 406 のビット 188:126 に記憶され、第 4 の疎サブツリー記述子 1400<sup>4</sup>はデータフィールド 406 におけるそれぞれのビットは、疎サブツリー記述子セレクト 1408 により選択される。たとえば、表 3 を見ると、疎サブツリー記述子セレクト 1408 が「0001」であれば、第 2 の疎サブツリー記述子 1400<sup>2</sup>が選択され、256 ビットのサブツリーデータ 412 のビット 125:63 が、マルチプレクサ 1602 を介して、選択された疎サブツリー記述子 1610 上で変換論理 1604 へ転送される。

【0140】

【表 3】

疎サブツリー記述子セレクト	サブツリーデータ	疎サブツリー記述子
0000	ビット 62:0	1
0001	ビット 125:63	2
0010	ビット 188:126	3
0011	ビット 251:189	4

表 3

【0141】

サブツリーメモリ 400 内の各サブツリーエントリ 404 はスペースモードまたはデンスマードで構成され得る。スペースモードで構成された各サブツリーエントリ 404 は、モード 1404 を介して疎サブツリー記述子 1400 ごとに異なる数のノード記述子 1402 を記憶するよう構成され得る。スペースモードで構成されたサブツリーエントリ 404 内の疎サブツリー記述子 1400 はすべて、疎サブツリー記述子 1400 ごとに同数のノード記述子 1402 を記憶する。

【0142】

ノード記述子 1402 はサブツリーにおける複数のノードを表わすようにコード化され得る。ノード記述子 1402 によって表わされる複数の 8 ビットノードは、8 ビットのうちのいくつかをマスキングすることにより識別される。マスクビットを各ノード記述子 1402 で記憶する代わりに、9 個のビットノード記述子 1402 を用いてノード記述子 1402 が表わす 8 ビット幅のノードを完全に符号化する。この 8 ビット幅のノードは、ランビットレンジス符号化を用いて 9 ビット幅のノード記述子 1402 において符号化される

。ランビットレンジス符号化により、そのノードの8ビットのうちのいずれがマスキングされるかが識別できるようになる。

【0143】

変換論理1604は、選択された疎サブツリー記述子1400に記憶された9ビット幅のノード記述子1402<sup>1</sup> - 1402<sup>n</sup>を、「X」(ドントケア)にセットされたビットを含む8ビットのCAM値1612へ変換し、この8ビットのCAM値1612をCAM1606へロードする。9ビットのノード記述子1402を変換論理1604により8ビットのCAM値1612へ変換する例を以下の表4に示す。

【0144】

【表4】

10

8ビット値	9ビットコード
101100XX	101100100
100XXXXX	100100000
001111XX	001111100
0011XXXX	001110000

表4

20

【0145】

9ビットコードの列は、ノード記述子1402に記憶された値を表わす。表4の第1行目を見ると、ノード記述子1402に記憶された9ビットコードは「101100100」であり、対応する8ビット値「1011100XX」がCAM1606に記憶される。変換論理1604は、9ビットのコードを「1」にセットされた第1のビットに対して右から左へサーチすることにより、この9ビットコードを変換する。9ビットコード中のビットを右から左へ見ると、最初の2つのビットが「0」にセットされ、3番目のビットが「1」にセットされている。変換論理1604は、最初の「1」の右側に「0」が2つあるので、「100」を2つのドントケア(「XX」)に変換する。1つ目の「1」は無視され、残りのビットが8ビット値の次の各ビットへと直接コピーされる。

30

【0146】

表4の第2行目を見ると、ノード記述子1402に記憶された9ビットコードは「100100000」である。変換論理1604は、最初の「1」についてこの9ビットコードを右から左へサーチすることにより、この9ビットコードを変換する。5番目の数字が「1」を記憶している。この9ビットコードは、5つの最下位ビット(「LSB」)が「ドントケア」(「X」)にセットされた8ビット値に変換される。9ビットのランビットレンジス符号化を用いてノード記述子1402を記憶することにより、各ノード記述子1402ごとに要求されるビット数は最小になり、よって、ルックアップテーブル200中に記憶され得るノード記述子1402の数が増加する。

40

【0147】

9ビットのノード記述子1402を8ビット値に変換した後、変換論理1604はこの8ビット値をCAM1606内へロードする。この8ビット値は、ノード記述子1402が選択された疎サブツリー記述子1400に記憶されたのと同じ順序で、すなわち最短一致から最長一致の順序で、CAM1606内にロードされる。CAM1606は、疎サブツリー記述子1400ごとのノード記述子1402の最大数を記憶するための記憶部を提供する。したがって、CAM1606は8ビット幅×16エントリの深さであり、デフォルトマッパアドレスおよびモード5疎サブツリー記述子1400に対する最大数のノード記述子1402を記憶するために15のエントリを提供する。CAM1606にはターナリの能力(ternary capability)と複数一致リゾルバ(multi-match resolver)とが内蔵されている。真の連想メモリを提供するのとは対照的に、小さなサイズのCAM1606が

50

、ゲートの形態で実現され得る。すなわち、CAM1606は、CAMをエミュレートするハードウェア回路で実現され得る。

【0148】

疎サブツリー記述子1400に記憶されたノード記述子1402の数は、疎サブツリー記述子1400が記憶されるサブツリーエントリ404を決定する。特定のモードの範囲内のノード記述子1402を記憶する疎サブツリー記述子1400は、同じサブツリーエントリ404に記憶される。各サブツリーのデフォルトルートに対して、デフォルトマッパアドレスが算出される。デフォルトの8ビット値は、CAM1606内の第1の場所に永久に記憶され、デフォルトマッパアドレスを算出する。

【0149】

選択された疎サブツリー1400に対する8ビット値がCAM1606内へロードされた後、CAM1606はキー210bの次の部分でサーチされる。キー210bの次の部分において最大数のビットに一致するCAM1606内のエントリが選択される。CAMのサーチの結果得られた一致アドレスは、ロックオフセット714として転送される。ロックオフセット714を用いて、サブツリーマッパ418(図5)に記憶されたルートに対応するマッパエントリのマッパアドレス416が決定される。

10

【0150】

図19A-Dは、疎に配置されたサブツリー1700内のあるノードに対するロックオフセット714の選択を図示する。図17Aは疎に配置されたサブツリー1700におけるルートを図示したものである。サブツリー1700内のノードは3つのルートr0、r1およびr2のうちの1つに対応しており、r0はサブツリー1700のデフォルトルートである。2つのルートr1およびr2は、疎サブツリー記述子1400のノード記述子1402<sup>1</sup>および1402<sup>2</sup>で符号化される。デフォルトルートr0の値は、CAM1606内の第1のエントリ1702に永久に記憶される。表2を参照して、2つのノード記述子1402を備える疎サブツリー記述子1400は、サブツリーエントリ404に記憶され、モードフィールド1404は「1」にセットされる。

20

【0151】

サブツリー1700を見ると、r2は10XXXXXXに一致するすべてのノードに対応し、r1は010XXXXXに一致するすべてのノードに対応する。各ノード記述子1402<sup>1</sup>および1402<sup>2</sup>により要求されるビット数を最小にして疎サブツリー記述子1400中の各ルートを記述するために、ノード記述子1402<sup>1</sup>および1402<sup>2</sup>は、ランビットレンジス符号化を用いてコード化される。コード化の方法は、そのノードを完全に符号化するのに用いられるビット数より1ビット多いものを用いる。最初の「X」(「ドントケア」)の場所に「1」が挿入され、残りのXは0としてコード化される。すなわち、ルート10XXXXXXは10100000に翻訳され、010XXXXXは01010000に翻訳される。

30

【0152】

図19Bは、疎サブツリー記述子1400中のノード記述子1402<sup>1</sup>および1402<sup>2</sup>の記憶部を表わす。ノード記述子1402<sup>1</sup>および1402<sup>2</sup>はサブツリーエントリ404に記憶され、モードフィールド1404は「1」にセットされる。これは、疎サブツリー記述子1400に記憶された2つのノード記述子1402<sup>1</sup>および1402<sup>2</sup>があるからである。サブツリーの最長一致はr1である。なぜなら、r1は最初の3ビットの一致を要求し、r2は最初の2ビットの一致を要求するからである。ノード記述子1402<sup>1</sup>および1402<sup>2</sup>は疎サブツリー記述子1400において最短一致から最長一致の順序で記憶され、最初にr2のノード記述子1402<sup>1</sup>が記憶されて次にr1のノード記述子1402<sup>2</sup>が次に記憶される。

40

【0153】

図19Cは、ノード記述子1402<sup>2</sup>の8ビットのマスキングされた値1706への変換を表わす。ノード記述子ビット1708<sup>1</sup>-1708<sup>9</sup>を左から右へ見ると、最初の「1」がビット1708<sup>6</sup>に記憶され、これは8ビットのマスキング値1706のマスクビット

50

の終わりを記している。ノード記述子 1402<sup>2</sup>を 8 ビットのマスキングされた値 1706 に変換するために、以下のビット変換が行なわれる。ノード記述子のビット 1708<sup>1</sup> に記憶された「0」は「X」に変換され、8 ビットのマスキングされた値のビット 1710<sup>1</sup> に記憶される。ノード記述子のビット 1708<sup>2</sup> に記憶された「0」は「X」に変換され、8 ビットのマスキングされた値のビット 1710<sup>2</sup> に記憶される。ノード記述子のビット 1708<sup>3</sup> に記憶された「0」は、「X」に変換され、8 ビットのマスキングされた値のビット 1710<sup>3</sup> に記憶される。ノード記述子のビット 1708<sup>4</sup> に記憶された「0」は「X」に変換され、8 ビットマスキングされた値のビット 1710<sup>4</sup> に記憶される。ノード記述子のビット 1708<sup>5</sup> に記憶された「0」は「X」に変換され、8 ビットのマスキングされた値のビット 1710<sup>5</sup> に記憶される。ノード記述子のビット 1708<sup>6</sup> に記憶された「1」は無視される。ノード記述子のビット 1708<sup>7</sup> に記憶された「0」は 8 ビットのマスキングされた値のビット 1710<sup>6</sup> に記憶される。ノード記述子のビット 1708<sup>8</sup> に記憶された「1」は 8 ビットのマスキングされた値のビット 1710<sup>7</sup> に記憶される。ノード記述子のビット 1708<sup>9</sup> に記憶された「0」は 8 ビットのマスキングされた値のビット 1710<sup>8</sup> に記憶される。  
10

#### 【0154】

図 19D は、CAM1606 内のノード記述子 1402<sup>1</sup> および 1402<sup>2</sup> の記憶部と、選択された疎サブツリー記述子 1400 に対してサブツリーマッパ 418 (図 5) に記憶された対応のマッパエントリ 504<sup>1</sup> - 504<sup>3</sup> とを表わす。選択されたサブツリー記述子 1400 に記憶された 9 ビットのノード記述子 1402<sup>1</sup> および 1402<sup>2</sup> は変換論理 1604 (図 18) に変換され、CAM1606 内へロードされる。CAM1606 内の第 1 のエントリ 1702 は、図 19A のサブツリー 1700 に示す r0 に対するデフォルトエントリである。第 2 のエントリ 1704 は、選択された疎サブツリー記述子 1400 に記憶された第 1 のノード記述子 1402<sup>1</sup> から変換される。第 2 のエントリ 1704 は、r2 に対して変換される最短の一一致である。選択されたサブツリー記述子 1400 に記憶された第 2 のノード記述子 1402<sup>2</sup> は、0101000000 から 010XXXXX へ変換され、CAM1606 内の第 3 のエントリ 1706 に記憶される。  
20

#### 【0155】

CAM1606 のサーチの結果、ロックオフセット 714 (図 18) が得られる。このロックオフセット 714 は、サブツリーマッパ 418 (図 5) に記憶されたマッパエントリ 504<sup>1</sup> - 504<sup>3</sup> のマッパアドレス 416 を決定するのに用いられる。CAM1606 は、最長一致を記憶するエントリ 1702、1704 および 1706 についてキー 210b の第 2 の部分でサーチされる。CAM1606 により提供されたロックオフセット 714 は、選択されたサブツリーエントリ 404 内のポインタフィールド 408 に記憶されたロックベースアドレス 600<sup>1</sup>、600<sup>1</sup> のうちの 1 つに依存するサブツリーベースアドレスと組合される。  
30

#### 【0156】

図 20 は、図 8 に示すポインタ論理 702 中のスペースモードベースセレクト論理 1800 を表わすブロック図である。ポインタロジック 702 は、サブツリーマッパ 418 (図 5) 内のマッパエントリ 504 (図 6B) のマッパアドレス 416 を算出するのに用いられるベースアドレス 716 を選択する。ポインタ論理 702 はデンスモードベースセレクト論理 710 およびスペースモードベースセレクト論理 1800 を含み、それらのうちの 1 つが、前のマッパレベルから転送された、サブツリーエントリ記述子 304 (図 4) に記憶されたタイプ 1406 の状態に依存して、選択される。既に述べたように、タイプ 1406 の状態は、サブツリーエントリ 404 がデンスモードで構成されるか否かを示す。  
40

#### 【0157】

スペースモードベースセレクト論理 1800 は、サブツリーエントリ 404 が複数の疎サブツリー記述子 1400 を記憶していれば、疎サブツリー記述子 1400 のベースアドレス 716 を算出する。スペースモードベースセレクト論理 1800 は、モードフィールド 1404 に記憶されたモード値 1608 と、サブツリーエントリ 404 内のブロックベー  
50

アドレスフィールド 600<sup>1</sup> および 600<sup>2</sup> に記憶されたサブツリー・ポインタ 414 と、前のマッパレベルから転送されたサブツリー・エントリ記述子 304 (図 4) に記憶された疎サブツリー記述子セレクト 1408 とを用いて、ベースアドレス 716 を算出する。ベースアドレス 716 は以下のように算出される。

【0158】

(疎サブツリー記述子の) ベースアドレス = ブロックベースアドレス + ベースオフセット  
ここでベースオフセット = ((1 + ノード / サブツリー) \* 疎サブツリー記述子セレクト)

たとえば、スペースモード 4 で構成されたサブツリー・エントリ 404 におけるサブツリー番号 2 の開始時のベースアドレス 716 を見つけるためには、最初にベースオフセットが計算される。サブツリー番号 2 に対する疎サブツリー記述子セレクト 1408 は「1」であり、ノード / サブツリーの数は 7 である (表 2 参照)。ベースオフセットは 8 ((1 + 7) \* 1) である。ブロックベースアドレス 600<sup>1</sup> および 600<sup>2</sup> の各々が、サブツリー・エントリ 404 に対して割当てられた 16 のマッパアドレスのブロックのベースアドレスである。サブツリー番号 2 のベースオフセットは 8 であり、これは 16 より小さいので、サブツリー 2 のブロックベースアドレスはブロックベースアドレス 600<sup>1</sup> になり、疎サブツリー記述子のベースアドレス 716 はブロックベースアドレス 600<sup>1</sup> + 8 になる。以下に表 5 に、モード 4 で構成されたサブツリー・エントリ 404 における 4 つのサブツリーの各々についてのサブツリー・ベースアドレスを表わす。

【0159】

【表 5】

サブツリー・ベースアドレス	サブツリー
ブロックベースアドレス 1+0	1
ブロックベースアドレス 2+8	2
ブロックベースアドレス 2+0	3
ブロックベースアドレス 2+8	4

表 5

【0160】

図 21 は、サブツリームリ 400 に記憶された稠密サブツリー記述子および疎サブツリー記述子を表わす。図 21 は図 15 と関連して説明する。サブツリー B (図 21) の稠密サブツリー記述子がサブツリー・エントリ 404<sup>1</sup> 内のデータフィールド 406<sup>1</sup> に記憶される。サブツリー A (図 21) の疎サブツリー記述子 1400<sup>1</sup> は、サブツリー・エントリ 404<sup>2</sup> 内のデータフィールド 406<sup>2</sup> に記憶される。稠密サブツリー記述子は、図 6B に関連して既に述べたように、サブツリー B の最下レベルにおいて各ノードに対してノードビットを記憶する。スペースモード記述子 1400<sup>1</sup> は、図 19B に関連して既に述べたように、ルート r4 および r5 に対応するノード記述子 1402<sup>1</sup> および 1402<sup>2</sup> を含む。サブツリーインデックス 312 はサブツリー・エントリ 404<sup>1</sup> および 404<sup>2</sup> を選択する。

【0161】

s0 (図 15) に対してマッパ 106a 内のマッパ・エントリ 504 中のサブツリー・エントリ記述子 304 (図 4) に記憶されたサブツリー・インデックス 312 がサブツリー・エントリ 404<sup>1</sup> を選択する。s1 (図 15) に対するマッパ 106a 内のマッパ・エントリ 504 (図 6B) におけるサブツリー・エントリ記述子 304 (図 4) に記憶されたサブツリー・インデックス 312 は、サブツリー・エントリ 404<sup>2</sup> を選択する。このように、サブツリームリ 400 は、疎サブツリーおよび稠密サブツリーに対してサブツリー・エントリ 404<sup>1</sup> および 404<sup>2</sup> を記憶し得る。

10

20

30

40

50

## 【0162】

図22は、疎に配置されたサブツリーおよび密に配置されたサブツリーにおけるあるノードに対するルートを記憶するサブツリーマッパ418(図5)内のマッパエントリ504(図6B)についてのマッパアドレス416(図5)を提供するための方法を図示するフローチャートである。いずれのサブツリーエントリ404も、複数の疎サブツリー記述子または単一の稠密サブツリー記述子を記憶し得る。バイナリツリー内でルートがいかに分散されるかに依存して、疎サブツリー記述子と密サブツリー記述子とのいかなる組合せも可能である。サブツリーメモリ400内のサブツリーエントリ404でスパースモードと稠密サブツリー記述子とを混合させ一致させる柔軟性により、サブツリーメモリ400をより有用化することが可能になる。

10

## 【0163】

ステップ1900において、選択されたサブツリーエントリ404の構成は、前のマッパレベルで選択されたサブツリーエントリ記述子304(図4)に記憶されたタイプ1406(図16C)の状態から決定される。サブツリーエントリ404タイプがスパースモードで構成される場合、処理はステップ1902へ進む。そうでなければ、処理はステップ1914へ進む。

## 【0164】

ステップ1902において、サブツリーエントリ404はスパースモードで構成される。スパースモードで構成されるサブツリーエントリ404は、複数の疎サブツリー記述子1400を記憶する。サブツリーエントリ404に記憶された疎サブツリー記述子1400の数は、モードフィールド1404の状態に依存する。オフセット論理700におけるスパースモード論理1502により、図14に関連して前に述べたように前のマッパレベルから転送されたサブツリーエントリ記述子304(図4)に記憶された疎サブツリー記述子セレクト1408とモードフィールド1404の内容とに依存して、サブツリーエントリ404から疎サブツリー記述子1400が選択される。処理はステップ1904へ進む。

20

## 【0165】

ステップ1904において、選択された疎サブツリー記述子1400内のノード記述子1402に記憶された9ビットコード化された値は8ビット値に変換され、最短一致から最長一致の順序で、CAM1606に記憶される。処理はステップ1906へ進む。

30

## 【0166】

ステップ1906において、CAM1606が、最長の一致を記憶するCAMエントリについてキー210bの次の部分でサーチされる。処理はステップ1908へ進む。

## 【0167】

ステップ1908において、キー210bの次の部分の最長一致を記憶するCAM1606における場所のアドレスが、ロックオフセット714として転送される。ロックオフセット714を用いて、サブツリーマッパ418(図5)内のマッパエントリ504(図6B)のマッパアドレス416(図5)が算出される。処理はステップ1910へ進む。

40

## 【0168】

ステップ1910において、選択された疎サブツリー記述子1400のベースアドレス716(図20)は、前のマッパレベルから転送されたサブツリーエントリ記述子304(図4)に記憶された疎サブツリー記述子セレクト1408と、選択されたサブツリーエントリ404に記憶されたモードフィールド1404の内容とに依存して、算出される。処理はステップ1912へ進む。

## 【0169】

ステップ1912において、マッパアドレス416は、アダーランジメント704(図8)内でロックオフセット714とベースアドレス716とを加算することにより算出される。サブツリーマッパ418(図5)内のマッパアドレス416により識別されたマッパエントリ504(図6B)は、ルートエントリ302(図4)またはサブツリーエントリ記述子

50

304(図4)のいずれかを記憶する。マッパエントリ504(図6B)がルートエントリ302(図4)を記憶する場合、サーチは完了する。マッパエントリ504(図6B)がサブツリーエントリ記述子304(図4)を記憶する場合、キー210に対応する値のサーチが次のマッパレベルにおいて継続される。

#### 【0170】

ステップ1914において、サブツリーエントリ404はデンスモードで構成され、データフィールド406に単一の稠密サブツリー記述子を記憶する。ロックオフセット714は、図6Bに関連して前述したように、サブツリーエントリ404内のデータフィールド406に記憶された稠密サブツリー記述子に記憶された「1」の数をカウントすることにより、算出される。処理はステップ1916へ進む。

10

#### 【0171】

ステップ1916において、サブツリーエントリ404は、サブツリーエントリ404内のポインタフィールド408に16個のブロックベースアドレス600を記憶する。ロックベースポインタ600のうちの1つが、図8と関連して前述したポインタ論理702におけるデンスモードベースセレクト論理710により選択される。処理はステップ1912へ進む。

#### インクリメンタルな更新

図23は、ルックアップテーブル200に追加される新たなルートのバイナリツリーを表わす図である。このバイナリツリーは、マッパレベル\_1 2000、マッパレベル\_2 2002およびマッパレベル\_3 2004について、ルックアップテーブル200に記憶されたルートを表わす。マッパレベル\_2 2002は、サブツリーAおよびBに対するルートを記憶する。マッパレベル\_3 2004はサブツリーA<sub>1</sub>、A<sub>2</sub>、B<sub>1</sub>およびB<sub>2</sub>に対するルートを記憶する。s5は、サブツリーマッパ418(図5)に記憶されたサブツリーエントリ記述子304(図4)を表わす。s5に対するサブツリーエントリ記述子304(図4)はサブツリーB<sub>2</sub>に対するポインタを記憶し、キー210に対する最長一致のルートのサーチをマッパレベル\_3 2004で継続できるようにする。

20

#### 【0172】

サブツリーB<sub>2</sub> 2006は、2つのルート、r6およびh1のみを有するので、疎サブツリーである。したがって、ノードr6およびh1のノード記述子1402(図16A)は、図14Aに関連して既に述べたように、疎サブツリー記述子1400に記憶される。サブツリーB<sub>2</sub> 2006の疎サブツリー記述子1400はサブツリーエントリ404に記憶され、サブツリーメモリ400においてモードフィールド1404は1にセットされる。これは、疎サブツリー記述子1400に記憶されたノード記述子1402が2つあるためである。

30

#### 【0173】

サブツリーB<sub>2</sub> 2008に示す新たなルートh2は、ルックアップテーブル200に追加されることになる。新たなルートh1をルックアップテーブル内のサブツリーB<sub>2</sub> 2006に直接追加することはできない。なぜなら、サブツリーB<sub>2</sub> 2006に対するルートを追加することにより、疎サブツリー記述子1400に記憶されたノード記述子1402の数が2から3へ増加するからである。疎サブツリー記述子1400へノード記述子1402を追加するとモードフィールド1404が「2」にセットされたサブツリーエントリ404において新たな疎サブツリー記述子1400を割当てる必要が生じる。すなわち、新たなルートh1の追加により、サブツリーB<sub>2</sub> 2006をサブツリーB<sub>2</sub> 2008で置き換える必要が生じる。

40

#### 【0174】

図24は、プロセッサメモリ2400に記憶された更新ルートを表わす。ルックアップテーブル200に記憶されたバイナリツリーのコピーもまた、ルックアップテーブル200とは別のプロセッサメモリ2400に記憶される。サブツリーB<sub>2</sub> 2006について記憶されたルートはプロセッサメモリ2400内のサブツリーB<sub>2</sub> 2008にコピーされ、新たなルートh2がサブツリーB<sub>2</sub> 2008に追加される。

50

## 【0175】

ルート更新ルーチン 2402 は、ルート更新命令 2404 のシーケンスを生成してサブツリー B<sub>2</sub> 2008 をルックアップテーブル 200 に追加し、ルート更新 2404 をテーブル更新ルーチン 2406 へ転送する。テーブル更新ルーチン 2406 はルート更新 2402 に対してテーブル更新 2410 を生成し、更新サイクル 2412 を転送し、ルックアップテーブル 200 をルート更新 2404 で更新する。更新サイクル 2412 は、サブツリーマッパ 418 (図 5) およびサブツリーメモリ 400 (図 5) における適切なメモリ場所にルート更新を書き込む。

## 【0176】

図 23 を参照して、更新サイクル 2412 はサブツリーマッパ 418 (図 5) の一部分を割当て、新たなサブツリー B<sub>2</sub> 2008 のルートをマッパエントリ 504 (図 6B) に記憶するための命令を含む。サブツリー B<sub>2</sub> 2008 は、ルート h1 および r6 ならびに新たなルート h2 に対してマッパエントリ 504 (図 6B) に記憶されたルートエントリを含む。サブツリー B<sub>2</sub> 2008 に対するルートエントリがサブツリーマッパ 418 (図 5) 内のマッパエントリ 504 (図 6B) に記憶された後、ルートに対するノード記述子 1402 が作成され、疎サブツリー記述子 1400 に記憶される。疎サブツリー記述子 1400 はサブツリーエントリ 404 に記憶される。サブツリーエントリ 404 のモード 1404 は、疎サブツリー記述子 1400 に記憶されたノード記述子 1402 の数に関連する。

## 【0177】

サブツリー B<sub>2</sub> 2008 の疎サブツリー記述子 1400 がルックアップテーブル 200 中のサブツリーメモリ 400 のサブツリーエントリ 404 に記憶された後、s5 で表わされるサブツリーエントリ記述子 304 (図 4) は、サブツリー B<sub>2</sub> 2006 の代わりにサブツリー B<sub>2</sub> 2008 を指すように変更される。サブツリー B<sub>2</sub> 2008 がルックアップテーブルに追加される一方、サブツリー B<sub>2</sub> 2006 に記憶されたルート r6 および h1 は s5 を介してアクセスされ得る。サブツリー B<sub>2</sub> 2008 がルックアップテーブルに記憶された後、s5 は、サブツリー B<sub>2</sub> 2008 およびルート r6 を指すように変更され、新たなルート h2 がアクセスされ得る。したがって、サブツリー B<sub>2</sub> 2006 の、ルート r6 および h1 に対応するルートインデックスについてのサーチを継続することができ、一方、新たなルート h2 はルックアップテーブル 200 に追加される。

## 【0178】

図 25 は、ルックアップテーブル 200 中のサブツリーマッパ 418b におけるマッパエントリ 504c<sup>4</sup> に記憶された、図 23 に示す新たなルート h2 を表わす。図 25 は、図 24 に示すバイナリツリーの図に関連して説明する。

## 【0179】

マッパレベル\_2 2002 におけるサブツリー B は 3 つのルート、すなわち r3、s4 および s5 を有する。サブツリー B は 16 未満のルートを有するので、疎サブツリーである。サブツリー B の r3、s4 および s5 に対するノード記述子 1402a<sup>1</sup> - 1402a<sup>3</sup> は、サブツリーメモリ 400a 内のサブツリーエントリ 404a における疎サブツリー記述子 1400a<sup>1</sup> に記憶される。マッパエントリ 504a<sup>2</sup> - 504a<sup>4</sup> は、サブツリー B 内の各ルートに対してサブツリーマッパ 418a に記憶される。サブツリー B のデフォルトルートは、サブツリーマッパ 418a 内のマッパエントリ 504a<sup>1</sup> に記憶される。マッパエントリ 504a<sup>2</sup> - 504a<sup>4</sup> の各々は、そのノードに対するルートエントリ 302 (図 4) またはサブツリーエントリ記述子 304 (図 4) を記憶する。サブツリーエントリ記述子 304 (図 4) は、ルート s4 については 504a<sup>3</sup> に、s5 については 504a<sup>4</sup> に記憶される。s5 についてマッパエントリ 504a<sup>4</sup> に記憶されたサブツリーエントリ記述子 304 (図 4) は、サブツリーメモリ 400b にサブツリーインデックス 312b を与え、次のレベルのサーチ、すなわちマッパレベル\_3 2004 に対するサーチを開始する。

## 【0180】

10

20

30

40

50

サブツリー  $B_2$  は、2つのルートすなわち  $h_1$  および  $r_6$  を有するので、これもまた疎サブツリーである。ノード記述子  $1402b^1 - 1402b^2$  は、サブツリーメモリ  $400b$  内のサブツリーエントリ  $404b$  における疎サブツリー記述子  $1400b^1$  に記憶される。サブツリー  $B_2$  内の各ルートは、マッパエントリ  $504b^2 - 504b^3$  に記憶され、サブツリー  $B_2$  のデフォルトルートがマッパエントリ  $504b^1$  に記憶される。

#### 【0181】

サブツリー  $B_2$   $2006$  内でルート  $h_1$  についてサーチするために、ルート  $s_5$  についてノード記述子  $1402$  を記憶する疎サブツリー記述子  $1400a$  を記憶しているサブツリーエントリ  $404a$  のアドレスが、サブツリーインデックス  $312a$  上でサブツリーメモリ  $400a$  へ転送される。選択されたサブツリーエントリ  $404a$  に記憶されたデータフィールド  $406$  およびポインタフィールド  $408$  は、サブツリーデータ  $412a$  およびサブツリーポインタ  $414a$  上でマッパアドレス論理  $402a$  へ転送される。マッパアドレス論理  $402a$  は、 $s_5$  についてサブツリーエントリを記憶しているマッパエントリ  $504a^4$  のマッパアドレス  $416a$  を生成する。マッパアドレス  $416a$  は、サブツリーデータ  $412a$ 、サブツリーポインタ  $414a$  およびキー  $210b$  の次の部分に依存する。 $s_5$  のサブツリーエントリは、サブツリーインデックス  $312b$  上でサブツリーメモリ  $400b$  へ転送される。

#### 【0182】

サブツリーメモリ  $400b$  は、サブツリー  $B_2$   $2006$  についてノード記述子  $1402b^2$  および  $1402b^1$  を記憶する。 $B_2$  についての疎サブツリー記述子  $1400b^1$  がサブツリーエントリ  $404b$  に記憶される。サブツリーエントリ  $404b$  に記憶されたデータフィールド  $406$  およびポインタフィールド  $408$  は、サブツリーデータ  $412b$  およびサブツリーポインタ  $414b$  上でマッパアドレス論理  $402b$  へ転送される。マッパアドレス論理  $402b$  は、 $h_1$  についてルートエントリを記憶しているマッパエントリ  $504b^3$  のマッパアドレス  $416b$  を生成する。マッパアドレス  $416b$  は、サブツリーデータ  $412b$ 、サブツリーポインタ  $414b$  およびキー  $210c$  の次の部分に依存している。

#### 【0183】

ルート  $h_2$  をサブツリー  $B_2$   $2006$  に追加するためには、サブツリーマッパ  $418b$  内のこれまでに未使用のマッパエントリ  $602b$  のブロックが割当てられ、サブツリー  $B_2$   $2008$  についてルート  $r_6$ 、 $h_1$  および  $h_2$  を記憶しているマッパエントリ  $504c^2 - 504c^4$  を記憶する。マッパエントリ  $504c^1$  は、サブツリー  $B_2$   $2008$  のデフォルトルートエントリ、すなわち、マッパエントリ  $504b^1$  に記憶されたものと同じ値を記憶する。マッパエントリ  $504c^2$  はルート  $r_6$  のルートエントリ、すなわちマッパエントリ  $504b^2$  に記憶されたものと同じ値を記憶する。マッパエントリ  $504c^3$  はルート  $h_1$  のルートエントリ、すなわちマッパエントリ  $504b^3$  に記憶されたものと同じ値を記憶する。マッパエントリ  $504c^4$  は、新たなルート  $h_2$  に対するルートエントリを記憶する。マッパエントリ  $504c^{1-4}$  のブロックが書込まれている間に、マッパエントリ  $504b^1 - 504b^3$  に記憶されたルートエントリは、サブツリーマッパ  $418a$  内の  $504a^4$  にルート  $s_5$  について記憶されたサブツリーエントリを介して、アクセスされ得る。

#### 【0184】

サブツリーマッパ  $418b$  にはサブツリー  $B_2$   $2008$  についてのマッパエントリ  $504c^{1-4}$  が記憶されており、疎サブツリー記述子  $1400c^1$  がサブツリーメモリ  $400b$  に追加される。ノード記述子  $1402c^{1-3}$  の数は 16 未満であるので、ノード記述子  $1402c^{1-3}$  は疎サブツリー記述子  $1400c^1$  に記憶される。サブツリーメモリ  $400b$  内のサブツリー記述子  $1400^1$  の場所は、疎サブツリー記述子  $1400c^1$  に関連付けられるノード記述子  $1402c^{1-3}$  の数に依存する。サブツリー  $B_2$   $2006$  に新たなルートを追加することにより、疎サブツリー記述子  $1400c^1$  に対して記憶されるノード記述子  $1402c^1 - 1402c^3$  の数は 2 から 3 へ増加する。疎サブツリー記述子  $1400$

10

20

30

40

50

$c^1$  はサブツリーエントリ 404c に記憶され、各疎サブツリー記述子ごとに 3 つのノード記述子があり、モードフィールド 1404 は「2」にセットされる。疎サブツリー記述子 1400c $^1$  は、利用可能なスペースがあれば現在のモード 3 サブツリーエントリ 404c に記憶されるか、または新たなモード 3 サブツリーエントリが割当てられる。B<sub>2</sub> 2008 内のルートについてのノード記述子は、モード 3 サブツリーエントリ 404c の疎サブツリー記述子 1400c $^1$  におけるノード記述子 1402c $^{1-3}$  に記憶される。

【0185】

疎サブツリー記述子 1400c $^1$  およびノード記述子 1402c $^{1-3}$  がサブツリーメモリ 400b に記憶された後、サブツリー B<sub>2</sub> 2008 がアクセスされ得る。B<sub>2</sub> 2008 にアクセスを提供するために、サブツリーエントリ 504a $^4$  は、サブツリーエントリ 404b 内の疎サブツリー記述子 1400b $^1$  の代わりにサブツリーエントリ 404c 内の疎サブツリー記述子 1400c $^1$  をインデックス付けするよう変更される。マッパエントリ 504c $^4$  に記憶されたルート h<sub>2</sub> についてのルートエントリならびにそれぞれのマッパエントリ 504c $^2$  および 504c $^3$  に記憶されたルート r<sub>6</sub> および h<sub>1</sub> がアクセスされ得る。

【0186】

マッパエントリ 504b $^1$  - 504b $^3$  はもはやアクセス不可能であるので、割当解除され、将来の割当のためにフリーリスト（図示せず）上に置かれる。また疎サブツリー記述子 1400b $^1$  はもはやアクセス不可能である。こうして、疎サブツリー記述子 1400b $^1$  は割当解除され、将来の割当のためにフリーリスト（図示せず）に置かれる。

【0187】

疎サブツリーにルートを追加することについて説明した。ルートは、新たに割当てられたサブツリーエントリ 404 に新たな稠密サブツリー記述子を記憶し、対応のマッパエントリをサブツリーマッパ 418 に記憶することにより、また、マッパエントリ 504a $^4$  に記憶されたサブツリーエントリを新たに割当てられたサブツリーエントリ 404 のインデックス付けを行なうように変更することにより、追加することもできる。

【0188】

図 26 は、図 25 に示すルックアップテーブル 200 にあるルートを追加するためにインクリメンタルな更新を行なうためのステップを表わすフローチャートである。

【0189】

ステップ 2200 において、サブツリーごとのルートの数が算出され、ルート更新結果が疎サブツリーであるか稠密サブツリーであるかが判定される。ルート更新後のサブツリーが稠密であれば、処理はステップ 2218 へ進む。ルート更新後のサブツリーが疎であれば、処理はステップ 2202 へ進む。

【0190】

ステップ 2202 において、サブツリーは疎である。疎サブツリーモードが決定される。処理は 2204 へ進む。

【0191】

ステップ 2204 において、サブツリーマッパ 418（図 5）に記憶された部分的にフルにされたサブツリーエントリ 404 のリストがサーチされ、新たな疎サブツリー記述子 1400c $^1$  が前に割当てられたサブツリーエントリ 404 に記憶され得るか否かが判定される。たとえば、4 つの疎サブツリー記述子 1400c $^1$  - 1400c $^4$  がモード 4 サブツリーエントリ 404 に記憶され得る。3 つのみが記憶される場合、サブツリーエントリ 404 は部分的にフルにされており、部分的に満たされたサブツリーエントリ 404 のリストに記憶される。部分的にフルなサブツリーエントリ 404 が利用可能であれば、処理はステップ 2208 へ進む。そうでなければ、処理はステップ 2206 へ進む。

【0192】

ステップ 2206 において、新たなサブツリーエントリ 404c が疎サブツリー記述子 1400c $^1$  を記憶するために割当てられ、マッパエントリ 504c $^1$  - 504c $^4$  が、新たに割当てられたサブツリーエントリ 404c 内の疎サブツリー記述子 1400c $^1$  に記憶

10

20

30

40

50

されたノード記述子  $1\ 4\ 0\ 2\ c^1 - 1\ 4\ 0\ 2\ c^3$  のマッパエントリ  $5\ 0\ 4$  (図 6 B) を記憶するために、サブツリー・マッパに割当てられる。サブツリー・マッパ  $4\ 1\ 8$  (図 5) 内のマッパエントリ  $5\ 0\ 4\ c^1 - 5\ 0\ 4\ c^4$  の割当てられたブロックに対するポインタは、新たなサブツリー・エントリ  $4\ 0\ 4\ c$  内のポインタフィールド  $4\ 0\ 8$  に記憶される。処理はステップ 2208 へ進む。

#### 【0193】

ステップ 2208 において、疎サブツリー記述子  $1\ 4\ 0\ 0\ c^1$  に対するサブツリー・マッパ内の第 1 のマッパエントリ  $5\ 0\ 4\ c^1$  の場所が、サブツリー・エントリ  $4\ 0\ 4\ c$  内のポインタフィールド  $4\ 0\ 8$  に記憶されたポインタおよびサブツリー・エントリ  $4\ 0\ 4\ c$  内のモードフィールド  $1\ 4\ 0\ 4$  に記憶されたモードから判定される。処理はステップ 2210 へ進む 10。

#### 【0194】

ステップ 2210 において、疎サブツリーのルートエントリが、サブツリー・マッパ  $4\ 1\ 8\ b$  内のマッパエントリ  $5\ 0\ 4\ c^1 - 5\ 0\ 4\ c^4$  に記憶される。処理はステップ 2212 へ進む。

#### 【0195】

ステップ 2212 において、ノード記述子  $1\ 4\ 0\ 2\ c^1 - 1\ 4\ 0\ 2\ c^3$  を記憶している疎サブツリー記述子  $1\ 4\ 0\ 0\ c^1$  がサブツリー・エントリ  $4\ 0\ 4\ c$  に記憶される。処理はステップ 2214 へ進む。

#### 【0196】

ステップ 2214 において、マッパエントリ  $5\ 0\ 4\ a^4$  に記憶されたサブツリー・エントリ記述子  $3\ 0\ 4$  (図 4) は、サブツリー・エントリ  $4\ 0\ 4\ c$  に記憶された新たな疎サブツリー記述子  $1\ 4\ 0\ 0\ c^1$  のインデックス付けを行なうよう変更される。マッパエントリ  $5\ 0\ 4\ c^4$  に記憶された  $h\ 2$  のルートエントリがここでアクセス可能にある。処理はステップ 2216 へ進む。 20

#### 【0197】

ステップ 2216 において、マッパエントリ  $5\ 0\ 4\ b^1 - 5\ 0\ 4\ b^3$  および疎サブツリー記述子  $1\ 4\ 0\ 0\ b$  はもはやアクセス不可能である。マッパエントリ  $5\ 0\ 4\ b^1 - 5\ 0\ 4\ b^3$  は、サブツリー・マッパ  $4\ 1\ 8\ b$  のマッパエントリ  $5\ 0\ 4$  (図 6 B) のフリーリストに置かれ、他のルートを記憶するために割当てられ得る。サブツリー・エントリ  $4\ 0\ 4\ b$  内の第 1 の利用可能な場所は、部分的にフルなサブツリー・エントリのリストにおいて更新される。処理が完了する。 30

#### 【0198】

ステップ 2218 において、新たなサブツリー・エントリ  $4\ 0\ 4$  がプロセッサメモリ  $2\ 4\ 0\ 0$  (図 24) に記憶されたフリーのサブツリー・エントリ  $4\ 0\ 4$  のリストから割当てられる。この新たなサブツリー・エントリ  $4\ 0\ 4$  は、新たな稠密サブツリー記述子を記憶するために割当てられる。サブツリー・マッパ  $4\ 1\ 8\ b$  におけるマッパエントリ  $5\ 0\ 4$  (図 6 B) のブロックが、ルートを記憶するために割当てられる。割当てられたマッパエントリ  $5\ 0\ 4$  (図 6 B) のブロックに対するポインタは、サブツリー・エントリ  $4\ 0\ 4$  (図 5) 内のポインタフィールド  $4\ 0\ 8$  (図 7) に記憶される。処理はステップ 2220 へ進む。 40

#### 【0199】

ステップ 2220 において、新たな稠密サブツリー記述子が、図 6 A および 6 B に関連して前述したように、新たなサブツリー・エントリ  $4\ 0\ 4$  内のデータフィールド  $4\ 0\ 6$  に書き込まれる。処理はステップ 2222 へ進む。

#### 【0200】

ステップ 2222 において、稠密サブツリーのルートエントリが、サブツリー・エントリ  $4\ 0\ 4$  内のポインタフィールド  $4\ 0\ 8$  に記憶されたポインタにより識別されたサブツリー・マッパ  $4\ 1\ 8$  (図 5) 内のマッパエントリ  $5\ 0\ 4$  (図 6 B) に記憶される。処理はステップ 2224 へ進む。

#### 【0201】

10

20

30

40

50

ステップ 2224において、マッパエントリ 504a<sup>4</sup>に記憶されたサブツリーエントリ記述子 304(図4)は、新たなサブツリーエントリ 404cに記憶された新たな稠密サブツリー記述子をインデックス付けするよう変更される。マッパエントリ 504c<sup>4</sup>に記憶された h2に対するルートエントリがここでアクセス可能になる。処理はステップ 2226へ進む。

【0202】

ステップ 2226において、古いサブツリーエントリ 404内のポインタフィールド 408に記憶されたポインタによりインデックス付けされたマッパエントリ 504(図6B)は、プロセッサメモリ 2400(図24)に記憶されたマッパエントリのフリーリストへ戻される。この古いサブツリーエントリ 404bは、プロセッサメモリ 2400(図24)に記憶されたサブツリーエントリのフリーリストに追加される。

10

【0203】

ルートをルックアップテーブルに追加する処理について述べてきた。ルートをルックアップテーブルから削除するためにも同様の処理が行なわれる。たとえば、h2 504c<sup>4</sup>をサブツリー B<sub>2</sub>から削除するためには、ルート r6 および h1 の 2つのノード記述子を有する新たな疎サブツリー記述子を記憶し、モード 2 サブツリーエントリに疎サブツリー記述子を記憶し、対応のサブツリーマッパを更新し、マッパエントリ 504a<sup>4</sup>に記憶されたサブツリーエントリ記述子 304(図4)を、新たなサブツリーエントリ 404に記憶された更新されたサブツリー記述子のインデックス付けを行なうよう変更する必要がある。

20

【0204】

この発明はその好ましい実施例を参照して具体的に図示し説明してきたが、当業者には、前掲の請求項に包含される本発明の範囲から離れることなく、形式および内容の点で種々の変更をなすことができるようわかるであろう。

【図面の簡単な説明】

【図1A】バイナリツリーの第1のレベルを表わす先行技術のビットマップを示す図である。

【図1B】キャッシュメモリで実現される先行技術のルックアップテーブルを示す図である。

【図2A】この発明の原則に従う最長一致プレフィックスルックアップテーブルを示す図である。

30

【図2B】図2Aに示されるルックアップテーブルに記憶されるルートインデックスのバイナリツリー表示を示す図である。

【図3】この発明の原則に従う40ビットキーに対する最長一致プレフィックスルックアップテーブルを示す図である。

【図4】図2Aに示されるダイレクト・マップト・マッパに記憶可能なマッパエントリのタイプを示す図である。

【図5】図2Bに示されるマッパレベル\_2 112b中のノードに対応するマッパを示す図である。

【図6A】サブツリーのバイナリツリー表示の図である。

40

【図6B】図6Aに示されたサブツリーのボトムレベルのノードに対応する、図5に示されたデータフィールドに記憶される稠密サブツリー記述子を示す図である。

【図7】図5に示された p t r フィールドを示す図である。

【図8】図5に示されたマッパアドレス論理を示す図である。

【図9】最長一致をサーチするためのステップを図示するフローチャートの図である。

【図10A】深度拡張を与えるための実施例を示す図である。

【図10B】図10Aに示された実施例のルックアップテーブルの1つを示す図である。

【図10C】値を記憶するのに利用可能なマッパエントリの数を増やすために深度拡張を与えるための別の実施例を示す図である。

50

【図10D】 図10Cに示された実施例のスレーブルックアップテーブルを示す図である。

【図11A】 図10Aおよび図10Cに示されたルックアップテーブル中のマッパエントリの中の、図2Bに示されたルートインデックスのバイナリツリー表示のノードの分布のバイナリツリー表示を示す図である。

【図11B】 図10Aおよび図10Cに示されたルックアップテーブル中のマッパエントリの中の、図2Bに示されたルートインデックスのバイナリツリー表示のノードの分布のバイナリツリー表示を示す図である。

【図12】 図10Aおよび図10Cに示されたルックアップテーブル中のマッパエントリの中に値を分散するための方法を図示するフローチャートの図である。 10

【図13】 図10Cに示されたルックアップテーブル中のマッパエントリのうち1つに記憶されるサーチキーに対応する値をサーチするための方法を図示するフローチャートの図である。

【図14】 図10Aに示されたルックアップテーブル中のマッパエントリのうち1つに記憶されるサーチキーに対応する値をサーチするための方法を図示するフローチャートの図である。

【図15】 第1のマッパレベルによってインデックス付けされる第2のマッパレベル中の疎らなサブツリーおよび稠密なサブツリーのバイナリツリー表示を示す図である。

【図16A】 図5に示されるサブツリーエントリおよび図4に示されるサブツリーエントリのデータフィールドおよびポインタフィールドに対して変更を加えて、サブツリーエントリ中への複数の疎らなサブツリー記述子の記憶を可能にするのを示す図である。 20

【図16B】 図5に示されるサブツリーエントリおよび図4に示されるサブツリーエントリのデータフィールドおよびポインタフィールドに対する変更を加えてサブツリーエントリ中への複数の疎らなサブツリー記述子の記憶を可能にするのを示す図である。

【図16C】 図5に示されるサブツリーエントリおよび図4に示されるサブツリーエントリのデータフィールドおよびポインタフィールドに対する変更を加えてサブツリーエントリ中への複数の疎らなサブツリー記述子の記憶を可能にするのを示す図である。

【図17】 疎らに配されたサブツリー中のノードに対してブロックオフセットを選択するための、図8に示されたオフセット論理中のスペースモードサブツリー論理を示す図である。 30

【図18】 図17のオフセット論理に示されたスペースモード論理を示す図である。

【図19A】 疎らに配されたサブツリー中のノードに対するブロックオフセットの選択を示す図である。

【図19B】 疎らに配されたサブツリー中のノードに対するブロックオフセットの選択を示す図である。

【図19C】 疎らに配されたサブツリー中のノードに対するブロックオフセットの選択を示す図である。

【図19D】 疎らに配されたサブツリー中のノードに対するブロックオフセットの選択を示す図である。

【図20】 図8に示されたポインタ論理中のスペースモードベースセレクト論理を示すブロック図である。 40

【図21】 サブツリーメモリに記憶される、稠密サブツリー記述子および疎サブツリー記述子を示す図である。

【図22】 疎らに配されたサブツリーおよび密に配されたサブツリー中のノードに対してルートを記憶するサブツリーマッパ中のマッパエントリに対してマッパアドレスを与えるための方法を図示するフローチャートの図である。

【図23】 ルックアップテーブルに加えるべき新たなルートのバイナリツリー表示を示す図である。

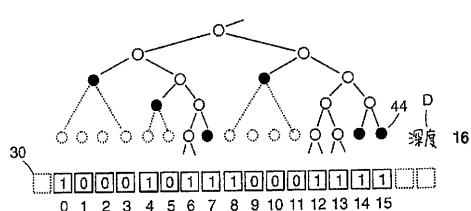
【図24】 プロセッサメモリに記憶される更新ルートを示す図である。

【図25】 ルックアップテーブルに記憶される、図23に示された新たなルートを示す 50

図である。

【図26】 図25に示されたルックアップテーブルに新たなルートを加えるためのステップを図示するフローチャートの図である。

【 図 1 A 】



**FIG. 1A**  
**PRIOR ART**

【図1B】

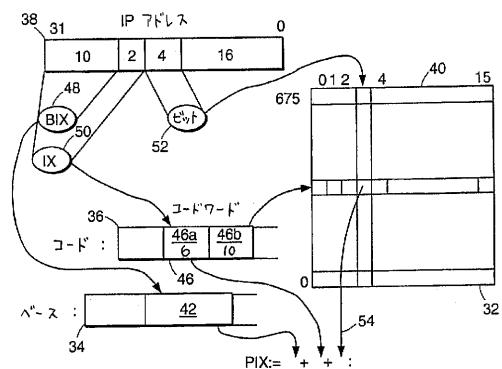


FIG. 1B  
PRIOR ART

【 図 2 A 】

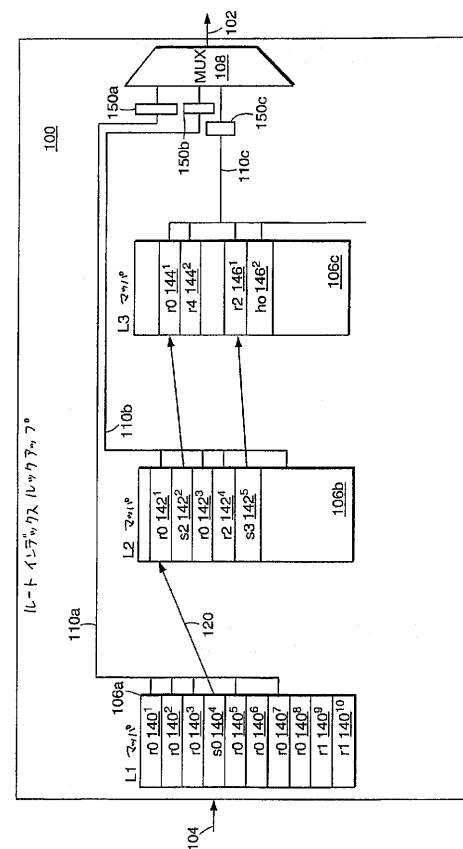


FIG. 2A

## 【図2B】

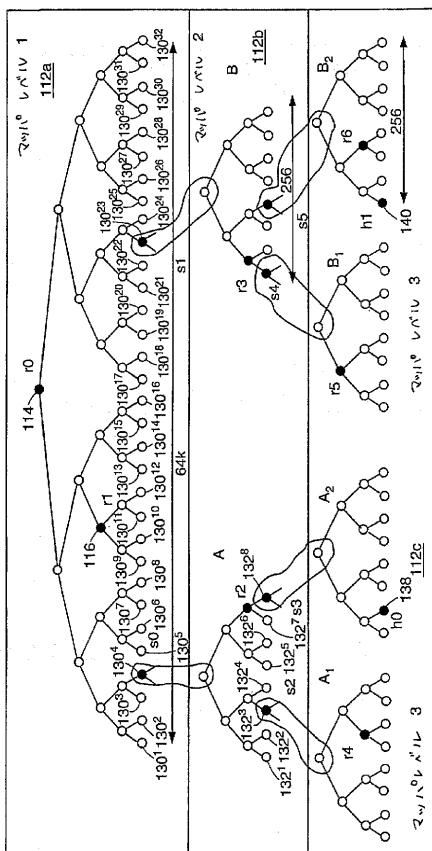


FIG. 2B

【図3】

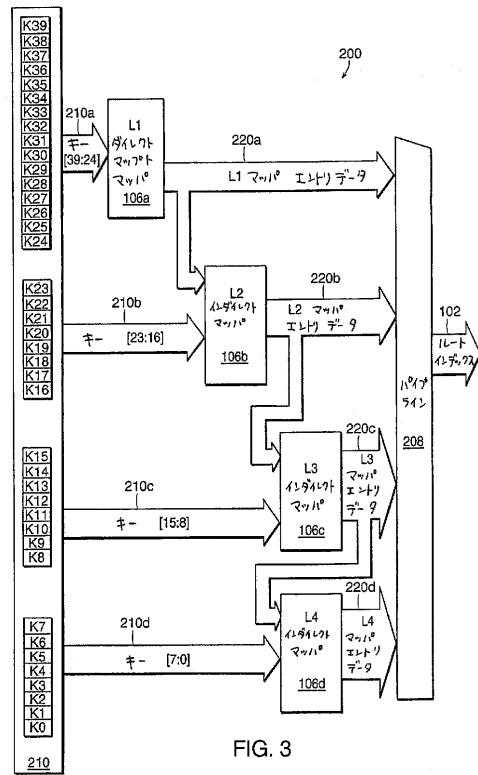


FIG. 3

【図4】

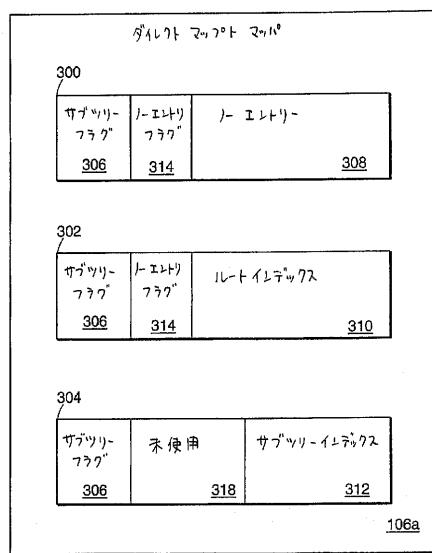


FIG. 4

【図5】

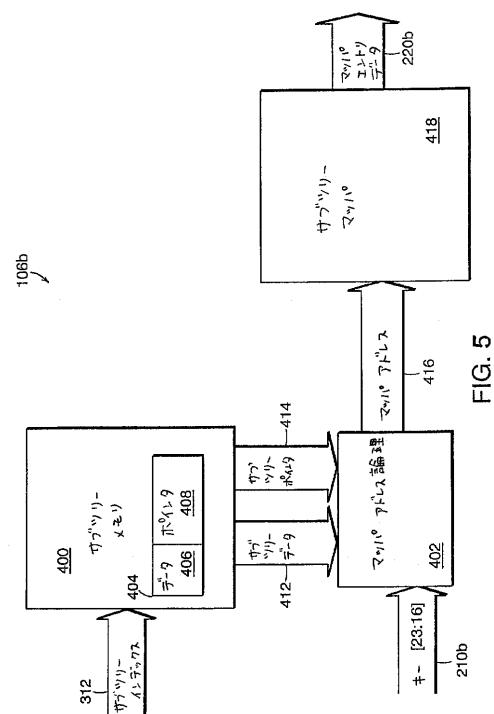


FIG. 5

【図 6 A】

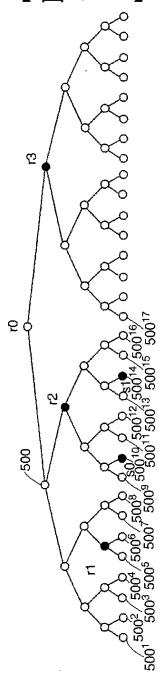


FIG. 6A

【図 6 B】

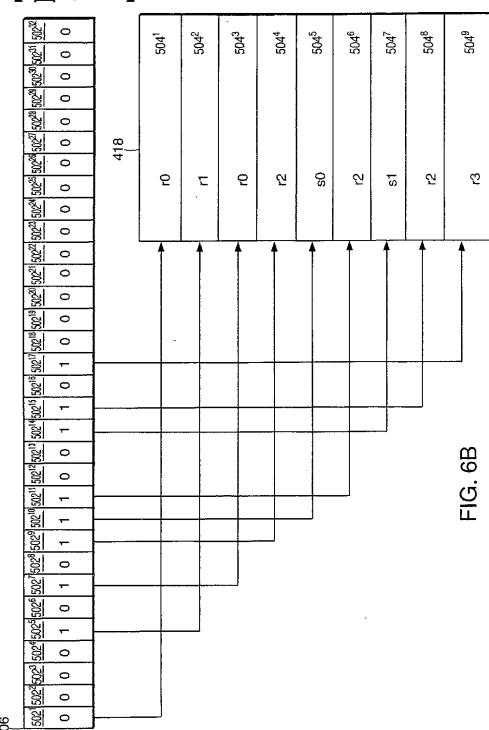


FIG. 6B

【 四 7 】

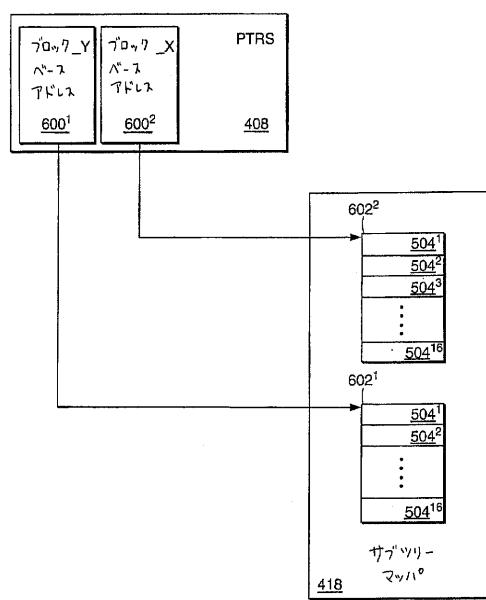


FIG. 7

【図8】

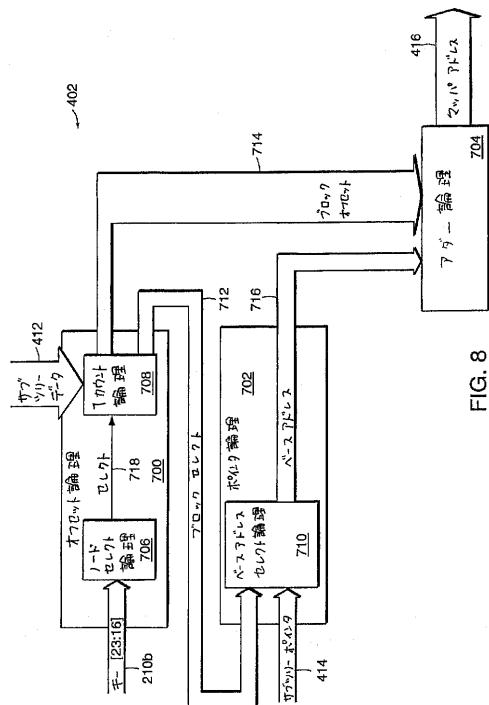


FIG. 8

【図9】

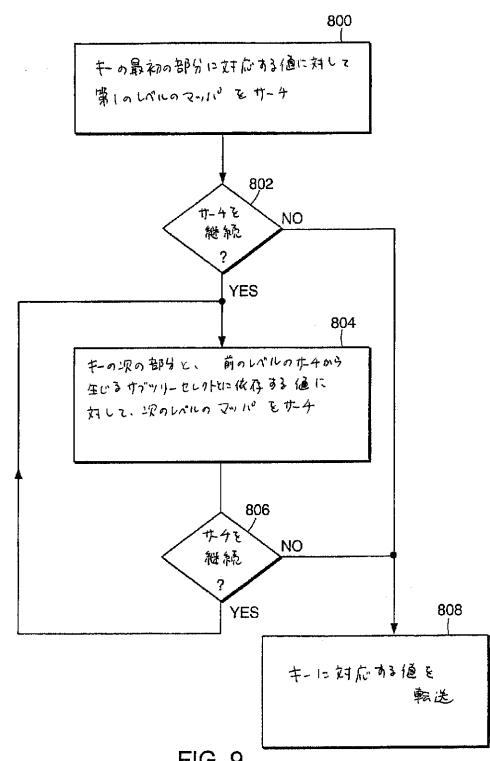


FIG. 9

【図10A】

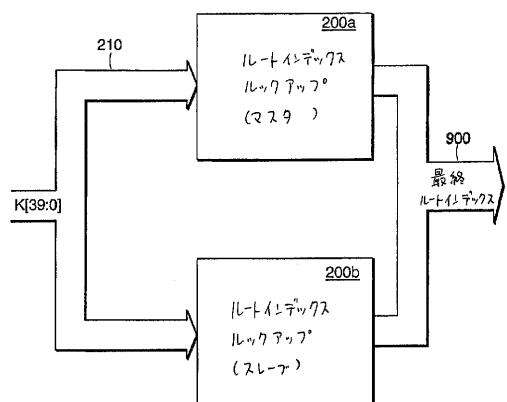


FIG. 10A

【図10B】

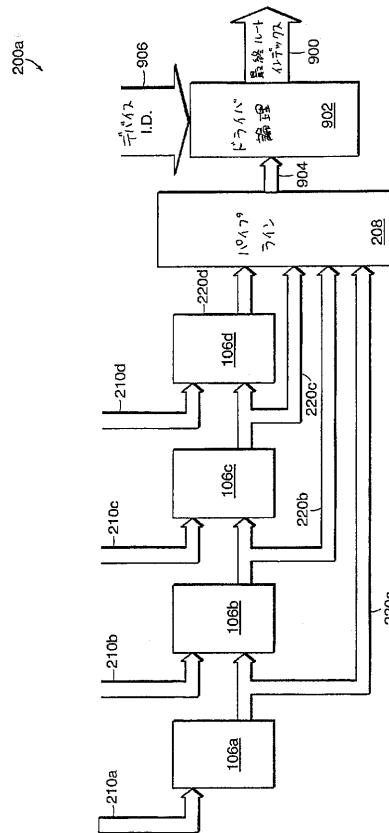


FIG. 10B

【図10C】

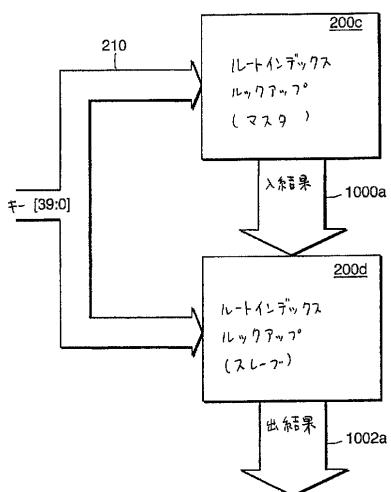


FIG. 10C

【図10D】

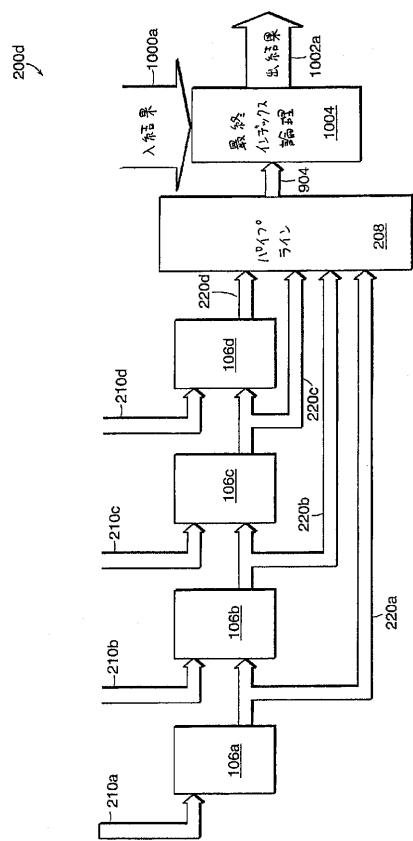


FIG. 10D

【図 1 1 A】

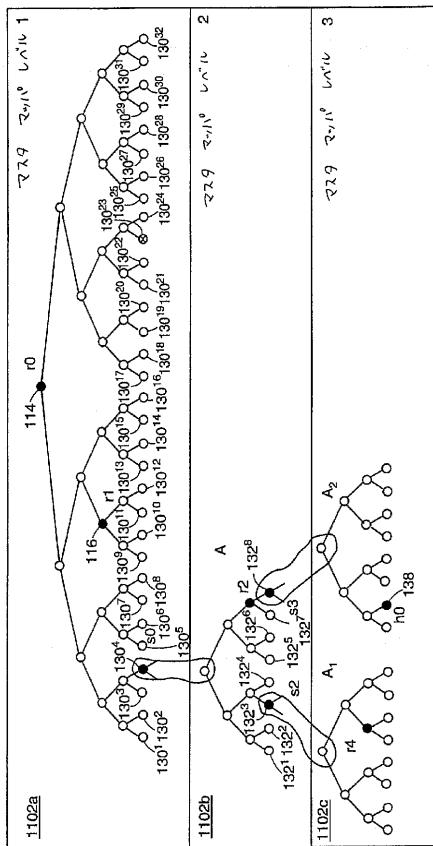


FIG. 11A

【図11B】

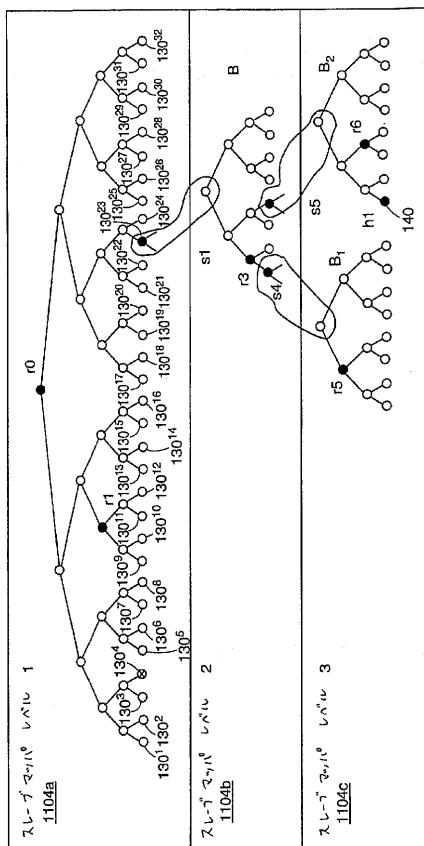


FIG. 11B

### 【図12】

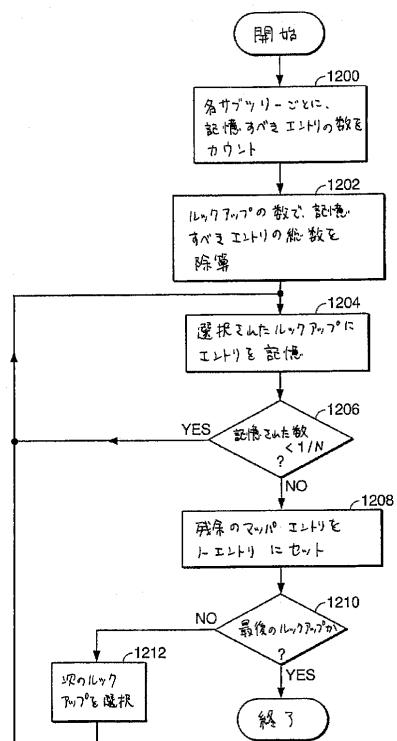


FIG. 12

【図 13】

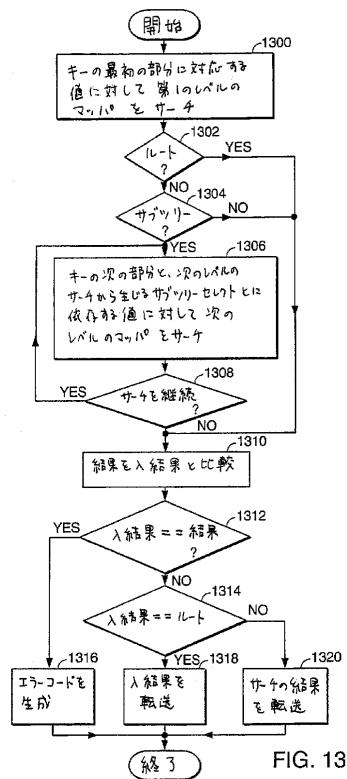


FIG. 13

【図 14】

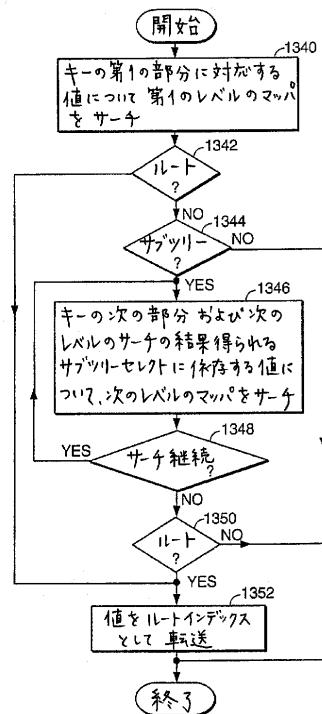
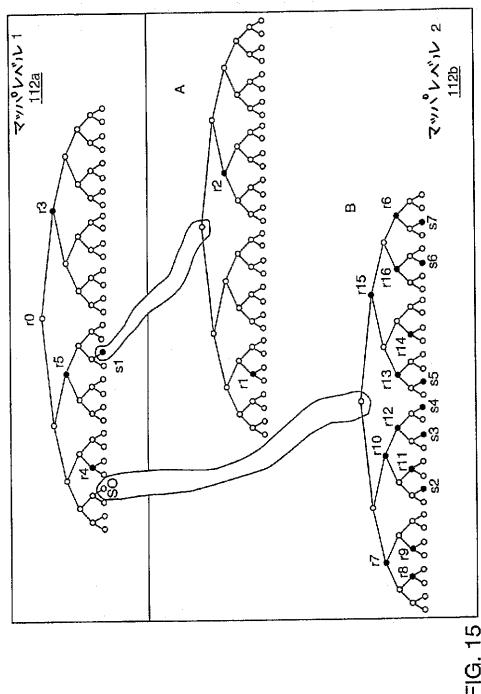


FIG. 14

【図 15】



【図16B】

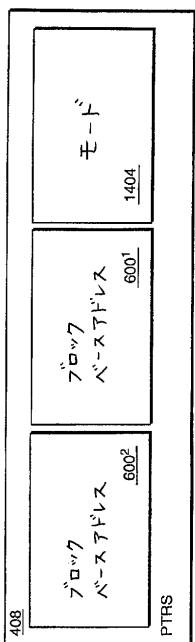


FIG. 16B

【図 1 6 C】

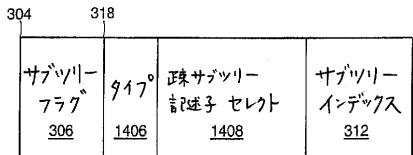


FIG. 16C

【 図 1 8 】

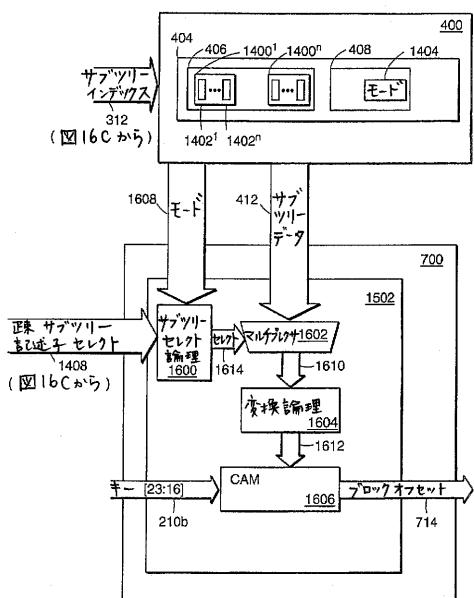


FIG. 18

【 図 17 】

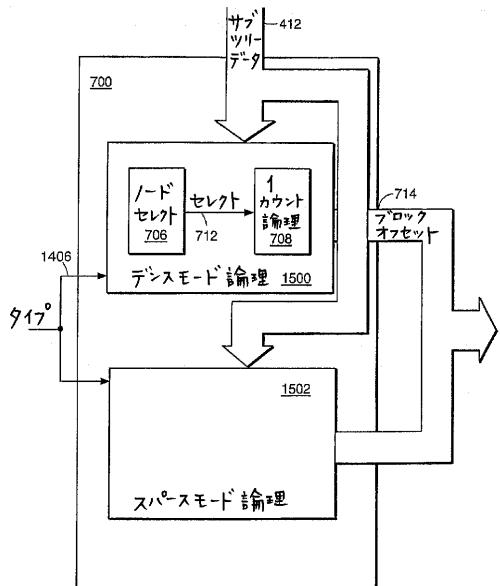
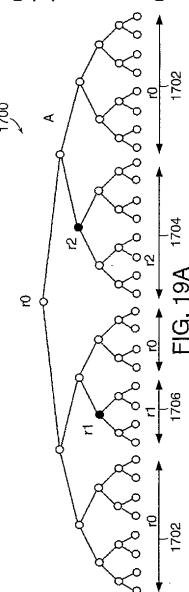


FIG. 17

【 19A 】



- 1 -

【図19B】

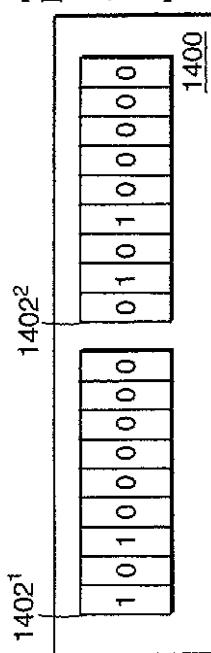


FIG. 19B

### 【図19c】

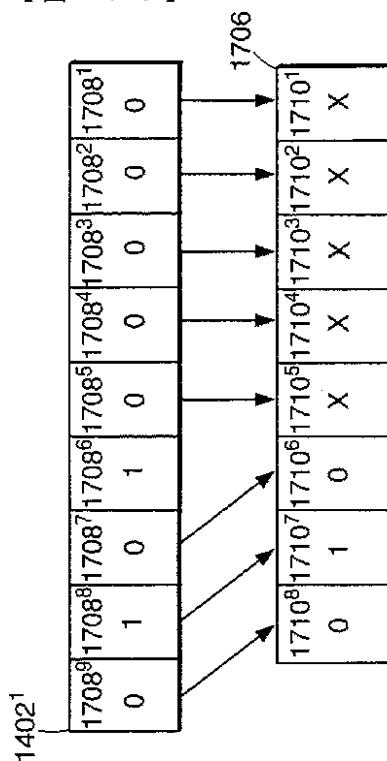


FIG. 19C

【図19D】

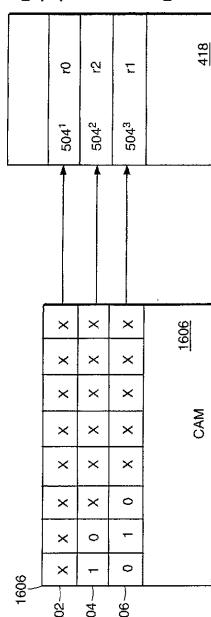


FIG. 19D

【 図 20 】

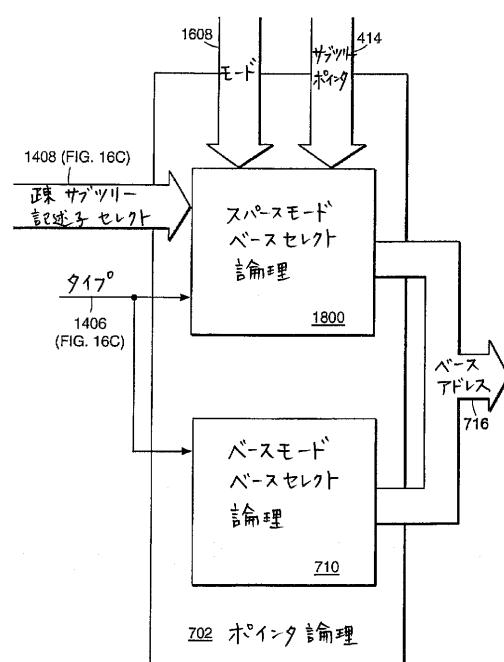


FIG. 20

【図21】

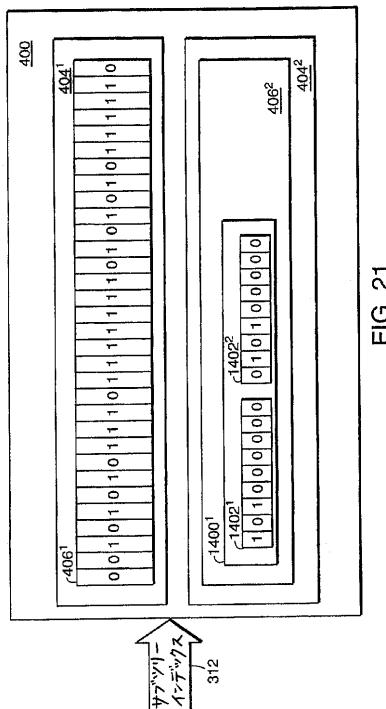


FIG. 21

【 図 2 2 】

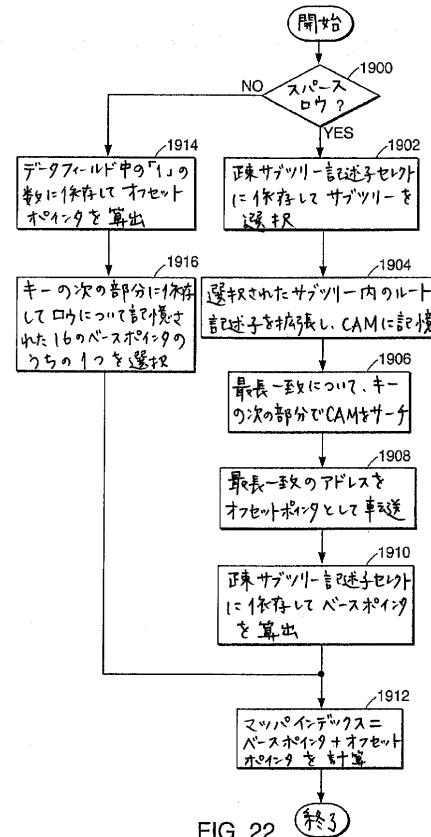


FIG. 22

【図23】

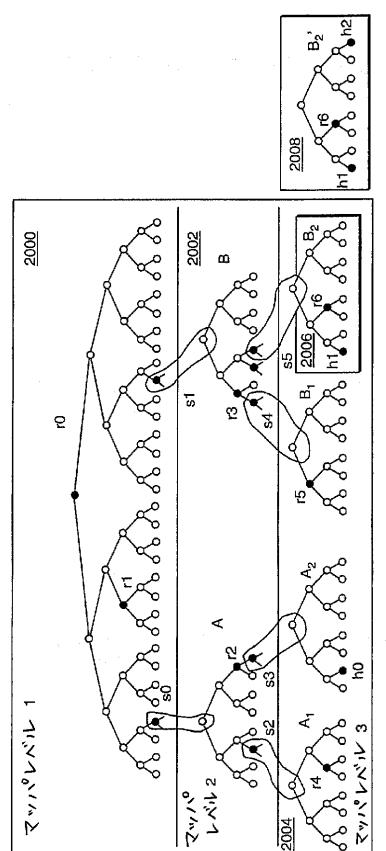


FIG. 23

【 図 2 4 】

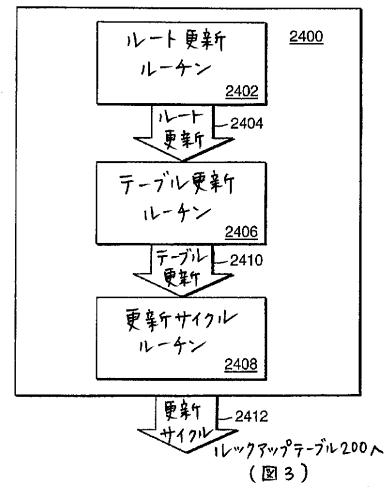


FIG. 24

【図25】

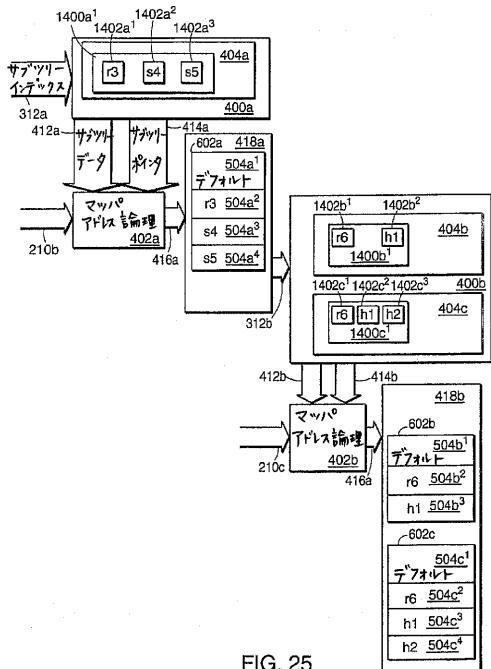


FIG. 25

【 図 26 】

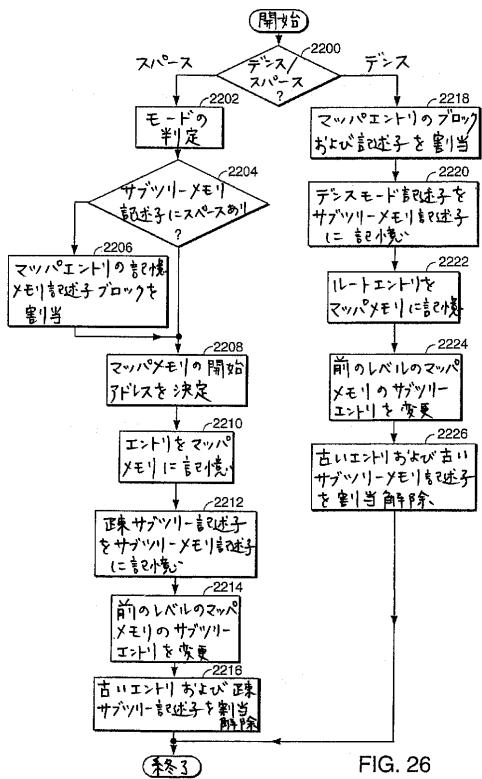


FIG. 26

---

フロントページの続き

(72)発明者 ブラウン, デイビッド・エイ  
カナダ、ケイ・0・エイ 1・エル・0 オンタリオ州、カーブ、アリーシア・クレセント、11  
0

審査官 斎藤 浩兵

(56)参考文献 特開平10-257066 (JP, A)  
国際公開第99/014906 (WO, A1)  
特開平11-191781 (JP, A)  
特開平11-066096 (JP, A)  
特開平11-232285 (JP, A)  
ポインタキャッシュを用いた最長一致に基づくルーティングテーブル検索法, 信学技報 Vol  
. 99 No. 135, 1999年 6月22日, p.7~12  
用語解説, 人工知能学会誌 第11巻 第2号, 社団法人人工知能学会, 1996年 3月 1日  
, p.337~339

(58)調査した分野(Int.Cl., DB名)

H04L 12/56