



[12] 发明专利申请公开说明书

[21] 申请号 00818944.7

[43] 公开日 2003 年 8 月 6 日

[11] 公开号 CN 1434950A

[22] 申请日 2000.12.8 [21] 申请号 00818944.7

[30] 优先权

[32] 1999.12.10 [33] US [31] 60/170,232

[32] 2000.6.21 [33] US [31] 60/212,966

[86] 国际申请 PCT/CA00/01444 2000.12.8

[87] 国际公布 WO01/43346 英 2001.6.14

[85] 进入国家阶段日期 2002.8.12

[71] 申请人 睦塞德技术公司

地址 加拿大安大略省

[72] 发明人 大卫·A·布朗

[74] 专利代理机构 中科专利商标代理有限责任公
司

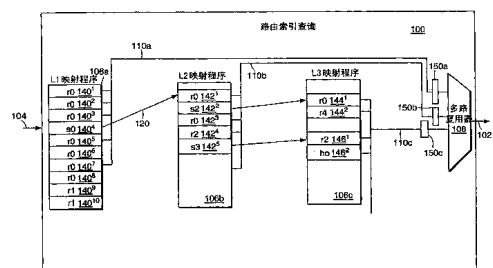
代理人 王 玮

权利要求书 2 页 说明书 41 页 附图 34 页

[54] 发明名称 最长匹配地址查询的方法和装置

[57] 摘要

一种在给出可用于搜索的查询表时对查询表进行增量更新的方法和装置。为了增加或删除查询表，在提供对查询表中的第一存储空间中存储的第一组路由的访问时，把第二组路由存储在查询表中的第二存储空间中。通过子树入口中存储的第一指针提供对第一存储空间的访问。将第二组路由存储到第二存储空间后，通过用指向第二存储空间的第二指针替换子树入口中存储的第一指针，将访问切换到第一存储空间中的第一组路由。



ISSN 1008-4274

1. 一种更新查询表中的子树的方法，包括步骤：
 - 5 通过指向第一子树入口的第一指针提供对查询表中第一存储空间中存储的第一组路由和相关的第二子树入口的访问；
将第二组路由和相关的第二子树入口存储到查询表中的第二存储空间；和
通过用指向第二子树入口的第二指针替换指向第一子树入口的第一
 - 10 指针使访问切换到第二存储空间中存储的第二组路由。
 2. 根据权利要求1所述的方法，进一步包括步骤：
切换访问后释放第一存储空间。
 3. 根据权利要求1所述的方法，其特征在于第一组路由中的路由数量小于第二组路由中的路由数量。
 - 15 4. 根据权利要求1所述的方法，其特征在于第一组路由中的路由数量大于第二组路由中的路由数量。
 5. 一种更新查询表的装置，包括：
指向第一子树入口的第一指针，第一子树入口提供对第一存储空间中存储的第一组路由的访问；和
 - 20 当通过第一指针提供对第一存储空间中存储的第一组路由的访问时，用来将第二组路由和相关的第二子树入口存储到第二存储空间以及通过用指向第二子树入口的第二指针替换第一指针使访问切换到第二组路由的装置，第二子树入口提供对第二存储空间的访问。
 6. 根据权利要求5所述的装置，进一步包括：
在切换访问后释放第一存储空间的装置。
 - 25 7. 根据权利要求5所述的装置，其特征在于第一组路由中的路由数量小于第二组路由中的路由数量。
 8. 根据权利要求5所述的装置，其特征在于第一组路由中的路由数量大于第二组路由中的路由数量。
 - 30 9. 一种更新查询表的装置，包括：

指向第一子树入口的第一指针，第一子树入口提供对第一存储空间中存储的第一组路由的访问；

当通过第一指针提供对第一存储空间中存储的第一组路由的访问时，用来存储第二组路由和相关的第二子树入口的第二存储空间；和

- 5 通过用指向第二子树入口的第二指针替换第一指针来提供对第二组路由的访问的逻辑，在第二组路由存储到第二存储空间中之后，第二子树入口提供对第二存储空间的访问。

10. 根据权利要求 9 所述的装置，进一步包括：

在替换第一指针后，用于释放第一存储空间的释放逻辑。

- 10 11. 根据权利要求 9 所述的装置，其特征在于第一组路由中的路由数量小于第二组路由中的路由数量。

12. 根据权利要求 9 所述的装置，其特征在于第一组路由中的路由数量大于第二组路由中的路由数量。

最长匹配地址查询的方法和装置

5

技术领域

因特网是由路由器连接的一组网络。路由器维护表示每个可能的目的网络，即接收的数据包应被转发到的下一跳的路由表。下一跳可以是另一个路由器或最终的目的地。

10

背景技术

在路由器中的端口接收的网际协议（“IP”）数据包包括一个 IP 目的地址。该 IP 目的地址是该 IP 数据包的最终目的地。目前，存在两种版本的 IP，IP 版本 4（“IPv4”）和 IP 版本 6（“IPv6”）。IPv4 提供包含在用来存储 IP 目的地址的数据包中 IP 首部中的 32 位字段。根据存储在 IP 首部中的 IP 目的地址，如果目的地在本地网，路由器转发接收到的与下一跳路由器或最终目的地相连的数据包。

一个 32 位的 IPv4 目的地址提供 40 亿个可能的路由。一个互连网路由器通常存储 40 亿个可能的路由中的 5 万个路由。然而，存储的路由数量会随着互连网的发展和 IPv6 的普及而增加。

起初，IP 地址空间分为 A，B 和 C 三类 IP 地址。每个 IP 地址空间分为网络地址和主机地址。A 类可容纳 126 个网络，每个网络可包含 16,000,000 台主机。B 类可容纳 16382 个网络，每个网络可包含 64,000 台主机，以及 C 类可容纳 2,000,000 个网络，每个网络可包含 256 台主机。然而，将 IP 地址空间划分为不同的类减少了可用 IP 地址的数量。C 类只允许每个网络最多 256 台主机对于大多数组织来说太少了。因此，大多数组织分配到一个 B 类地址，多达 64,000 个主机地址，这些地址即使该组织用不了也不能由其它组织使用。具有 B 类 IP 地址的组织的主机都在 16 位最高有效位（“MSBs”）中存储相同的网络地址，例如 27.32.xx.xx。

引入无级域间路由（“CIDR”）以便释放未使用的 IP 主机地址。剩余的未使用网络分配给规模不同的组织。一个需要 500 个地址的组织得到 500 个连续的地址。例如，一个组织可以分配到以 128.32.xx 为起始的 500 个可用地址。自从引入了无级域间路由，增加了存储在路由器中的路由数量。无级域间路由需要最长前缀匹配来寻找相应路由来代替为找到相应的 IP 目的地址的下一跳而寻找匹配的网络地址。例如，因为 128.32.4.xx 可能已经分配给其它需要不同下一跳的组织，所以在 B 类 IP 地址的 16 MSB，例如 128.xx.xx 之后，不再停止寻找。

一种用来查询关键字最长前缀匹配的方法是使用二叉树搜索。二叉树搜索与输入的 32 位逐位匹配到 32 层，需要 32 次搜索才能找到与 32 位关键字匹配的入口。另一种搜索匹配的方法是使用 Patricia 树。如果二叉树没有入口的叶子，Patricia 树减少所需的搜索次数。

Brodnick 等人于 1998 年 5 月 11 日提交的题为“快速路由查询的方法和系统”，序号为 PCT/SE98/00854 的 PCT 申请中描述了有效地搜索与 IP 目的地址有关的下一跳的另一种方法。Brodnick 描述的方法通过不再存储重复路由来减少所存储的下一跳的数量。通过减少下一跳的数量，减小了对内存的需求，因此路由查询表可以存储在快速超高速缓冲存储器中。

Brodnick 等人将二叉树划分为三层。将二叉树划分为三层使搜索次数减少到三。第一层中的索引入口表示搜索是否可以利用从入口获得的路由在第一层结束，或者该搜索必须利用 IP 目的地址另外的部分将继续到下一层。

图 1A 说明了表示二叉树第一层的现有技术的 64K (65536) 位表。64K 位表 30 代表在深度 16 层的二叉树的叶子或节点 44，每个节点 44 对应一位。位表被划分为长度为 16 的位屏蔽。在 64K 位表中共有 $2^{12}=4096$ 个位屏蔽。图 1A 中显示了一个位屏蔽。如果有子树或路由索引存储在节点 44 对应的指针数组中，则位表 30 中的一位赋值为“1”。如果该节点与前一节点 44 共享路由入口，则位表 30 中的一位赋值为“0”。

图 1B 说明了在超高速缓冲存储器中实现查询表的当前技术。查询表包括码字数组 36，基索引数组 34 和映射表 40。图 1B 中还显示了 32 位

IP 地址 38。针对位表 30（图 1A）中的每个位屏蔽，将码字 46 存储在码字数组 36 中。码字 46 包括一个 6 位的值 46a 和一个 10 位的位移 46b。针对码字数组 36 的数组中的每四个码字 46，将基索引 42 存储在基索引数组 34 中。

5 码字数组 36，基索引数组 34 和映射表 40 用来在指针数组（未显示）中选择指针。指针存储一个路由索引或一个索引以便执行进一步的搜索。

通过在码字数组 36 中选择码字 46 和在基索引数组 34 中选择基索引 42 来在指针数组中选择一组指针。利用 IP 地址 38 的前 12 位 50 来选择码字 46。利用 IP 地址 38 的前 10 位来选择基索引 42。利用映射表 32 在
10 一组指针中选择正确的指针。

所选码字 36 中 10 位的值 46b 作为到映射表 32 的索引。映射表 32 将一个位屏蔽中的位数映射到 4 位的位移。该位移指定在指针数组中所选的一组指针中的指针。10 位的值 46b 选择映射表 32 中的行，并且 IP 地址 52 的位 19: 16 选择 4 位的位移 54。

15 这样，搜索指针需要下面的超高速缓冲存储器访问：（1）读取 16 位码字 46；（2）读取 16 位基地址 42；（3）从映射表 32 中读取 4 位位移 54；（4）从指针索引读取指针，其中指针索引是基地址 42，码字位移 46a 和 4 位位移 54 之和。

20 二叉树的每一层需要同样的存储器访问。这样，3 层的搜索需要访问存储器 12 次。

发明内容

David A. Brown 提交的代理案编号 2037.2004-001，题为“长匹配地址查询的方法和装置”描述了用来执行多层搜索符合关键字的值的
25 查询表。能够存储在查询表中的路由索引的数量由查询表中存储单元的数量来限制。

根据本发明的原理，给出包括第一查询单元和第二查询单元的查询表。第一查询单元接收关键字并且通过第一单元中的多层搜索提供第一
30 结果。第二查询单元接收关键字并且通过第二单元中的多层搜索提供第二结果。第一查询单元和第二查询单元二者都并行地接收关键字，但根

据第一结果和第二结果，只有一个单元提供只存储在一个查询单元中的关键字的最终结果。

根据存储在查询表中的最终结果的总数将最终结果分配到表中。关键字可以是 IP 地址，最终结果是 IP 地址的路由索引。

- 5 由关键字索引的第一查询单元中的第一存储单元存储最终结果，由关键字索引的第二查询单元中的第二存储单元存储最终结果存储在第一个查询单元的指示。第一查询单元和第二查询单元都包括多个映射程序层并且每一个映射程序层包括用于存储结果的存储单元。

10 在一个实施例中，把与存储在第一个查询单元中的多个映射程序层的任何一层中的单元中的关键词对应的最终结果转发到第二查询单元，并由第一个查询单元提供。

在另一个实施例中，由存储最终结果的查询单元提供与该关键字对应的最终结果。

15 附图说明

图 1A 示出代表二叉树第一层的现有技术的位表；

图 1B 示出在超高速缓冲存储器中实现的现有技术的查询表；

图 2A 示出根据本发明原理的最长匹配前缀查询表；

图 2B 示出存储在图 2A 所示的查询表中的路由索引的二叉树表示；

- 20 图 3 示出根据本发明原理的 40 位关键字的最长匹配前缀查询表；

图 4 示出能够存储在图 2A 所示的直接映射的映射程序的映射程序入口的类型；

图 5 示出对应图 2B 所示的映射程序层 2 112b 中的节点的映射程序；

图 6A 是子树的二叉树表示；

- 25 图 6B 示出对应图 6A 所示的子树底层中的节点的图 5 所示的数据字段中存储的密集子树描述符。

图 7 示出图 5 所示的 ptr 字段；

图 8 示出图 5 所示的映射程序地址逻辑；

图 9 是说明搜索最长匹配的步骤的流程图；

- 30 图 10A 示出提供深度扩展的实施例；

图 10B 示出图 10A 所示的实施例中的一个查询表；

图 10C 示出通过提供深度扩展来增加可用于存储值的映射程序入口数量的另一个实施例；

图 10D 示出图 10C 所示的实施例中的副查询表；

5 图 11A—B 示出在图 10A 和 10C 所示查询表中的映射程序入口中图 2B 所示路由索引的二叉树表示中节点分配的二叉树表示。

图 12 是说明在图 10A 和 10C 所示的查询表中的映射程序入口中分配值的方法的流程图；

10 图 13 是说明搜索与图 10C 所示的查询表中一个映射程序入口中存储的搜索关键字对应的值的方法的流程图；

图 14 是说明搜索与图 10A 所示的查询表中一个映射程序入口中存储的搜索关键字对应的值的方法的流程图；

图 15 示出由第一映射程序层索引的第二映射程序层中稀疏子树和密集子树的二叉树表示；

15 图 16A—C 示出在图 5 所示子树入口和图 4 所示子树入口中数据字段和指针字段的修改以便在子树入口中允许存储多个稀疏子树描述符；

图 17 示出图 8 所示位移逻辑中稀疏模式子树逻辑以选择稀疏居住子树中节点的块位移；

图 18 示出图 17 中位移逻辑中所示的稀疏模式逻辑；

20 图 19A—D 示出在稀疏居住子树中为节点选择块位移；

图 20 是说明图 8 所示指针逻辑中稀疏模式基本选择逻辑的方框图；

图 21 示出在子树存储器中存储的密集子树描述符和稀疏子树描述符；

25 图 22 是说明向为稀疏居住子树和密集居住子树中的节点存储路由的子树映射程序中的映射程序入口提供映射程序地址的方法的流程图；

图 23 示出将要加到查询表的新路由的二叉树表示；

图 24 示出更新处理器存储器中存储的路由；

图 25 示出查询表中存储的图 23 所示的新路由；

图 26 是说明向图 25 所示查询表中增加新路由的步骤的流程图。

30 本发明前面所述的及其它目的，特征和优点将通过下面参考本发明

实施例的具体描述更加显而易见，不同附图中的相同部分用同一参考符号表示。附图不必放大，强调只起说明本发明原理的作用。

具体实施方式

5 下面描述本发明的优选实施例。

密集模式

图 2A 示出根据本发明原理的最长匹配前缀查询表 100。查询表 100 提供对应关键字 104 的路由索引 102。路由索引 102 用来访问对应 IP 目的地址的下一跳。在图 2A 所示的实施例中，关键字 104 为 32 位，但是
10 关键字 104 不必限制在 32 位。查询表 100 包括 3 个映射程序 106a-c。每个映射程序 106a-c 包括独立的地址存储器。对应关键字 104 的路由索引 102 或缺省路由索引存储在映射程序 106a-c 之一的一个单元中。如果需要搜索多个映射程序，来自每个映射程序的映射程序输出 110a-c 存储在延迟存储器 150 a-c 中，直到所有映射程序 106 a-c 已访问
15 了关键字。

多路复用器 108 选择转发到多路复用器 108 输入端中的一个映射程序输出 110 a-c 作为路由索引 102。根据映射程序输出 110 a-c 的最高有效位 (“MSB”) 来选择映射程序输出 110 a-c。仅当映射程序输出 110 a-c 包括路由索引 102 时，将映射程序输出 110 a-c 的 MSB 置为 “1”。

20 图 2B 说明了图 2A 所示的查询表 100 中映射程序 106 a-c 中存储的入口的二叉树表示。结合图 2A 来描述图 2B。32 位关键字 104 可以表示为 32 层二叉树。二叉树实现需要 32 次搜索，以便按位向下搜索到 32 层。为减少搜索次数，将 32 层的二叉树分为每个映射程序层 112 a-c 对应一个映射程序 106 a-c (图 2A) 的 3 个映射程序层 112 a-c。映射程序层 1 112a 包括 32 层二叉树的前 16 层。然而，为简单起见，图 2B 中只显示了 16 层中的 5 层。映射程序层 2 112b 包括 32 层二叉树接下来的 8 层，图 2B 显示了 8 层中的 3 层。映射程序层 3 包括 32 层二叉树的最后 8 层，图 1B 显示了 8 层中的 3 层。每个映射程序层 112a-c 包括多个
25 节点。将 32 层这样分配，使映射程序层 1 112a 中包括 16 层 (关键字 104 的 16MSB)，映射程序层 2 112b 中包括 8 层以及映射程序层 3 中包括 8
30 的 16MSB)，映射程序层 2 112b 中包括 8 层以及映射程序层 3 中包括 8

层对当前存储器技术显然是最适宜的；然而，本发明不受这种配置的限制。

取代对关键字 104 的前 16 位进行 16 次分别的逐位搜索，而是将与关键字 104 的前 16 位有关的路由索引 102 存储在映射程序 106a 中（图 5 2A）。映射程序 106a（图 2A）直接由关键字 104 的前 16 位 MSBs 索引。根据前一个映射程序 106a 是否存储了用于访问与关键字 104 有关的下一跳信息的路由索引 102 来搜索接下来的映射程序 106b。

如图 2B 所示，映射程序层 1 112a 所示的节点或叶子包括分别标为 r0 和 r1 的两个路由 114, 116, 以及分别标为 s0 和 s1 的指向映射程序层 2 112b 中 130^4 和 130^{23} 的两个指针。对应每个路由 114, 116 的路由索引 102 存储在 L1 映射程序 106a 中。另外，为子树索引 130^4 存储 L2 映射程序 106b 的地址指针 120, 为子树索引 130^{23} 存储 L2 映射 106b 的地址指针（未示出）。在映射程序 106a 中的映射程序入口 140^4 中为子树索引 130^4 存储的地址指针 120 表示为了寻找与关键字 104 有关的路由索引 102 所需的下一层的搜索。

树中任何节点的值可以通过跟踪从根 114 开始的路径来确定。所示二叉树中的每个节点有两个子节点，右子节点和左子节点。如果父节点为“1”，选择右子节点。如果父节点为“0”，选择左子节点。跟踪从根 114 到节点 116 的路径，对于所有 MSB 置为“010”的关键字，r1 作为路由索引 102 存储在 L1 映射程序 106a 中。跟踪从根 114 到 s0 节点 130^4 的路径，对于所有 MSBs 置为“00011”的关键字，s0 存储在 L1 映射程序 106a 中。

L1 映射 106a 是直接映射的映射程序并且存储映射程序层 1 112a 的底层的每个底层节点或叶子的路由索引 102。映射程序层 1 112a 的底层是 32 层二叉树的第 16 层。第 16 层有 64K 个节点。但是，为了说明的目的，所示映射程序层 1 112a 的底层作为 32 层二叉树的第五层显示。L1 映射程序 106a 所示的路由索引 102 对应映射程序层 1 112a 的第五层节点 130^1-130^{32} 。跟踪从根节点 114 到第五层节点 $130^1, 130^2, 130^3$ 的路径，路由索引 102 为 r0。因此 r0 存储在 L1 映射程序 106a 的位置 $140^1, 140^2, 140^3$ 中；即在索引 00000, 00001 和 00010。节点 130^4 存储子树索

引 s0, 因此 s0 存储在 L1 映射程序 106a 中地址为 00011 的位置 140⁴。
类似地, 第五层节点 130⁵—130⁸ 的路由索引 102 是 r0, 因此 r0 存储在 L1
映射程序 106a 中地址为 00100, 00101, 00110 和 00111 的位置 140⁵, 140⁶,
140⁷, 140⁸。对应第五层节点 130⁹—130¹² 的路由索引 102 是 r1, 因此 r1
5 存储在 L1 映射程序 106a 中地址为 01000 和 01001 的位置 140⁹, 140¹⁰。

在 L1 映射程序 106a 的每个位置存储直接分配给第五层节点 300¹—
300³²或通过第五层节点 300¹⁻³²的父节点分配或指向下一个映射程序 106b
—c 的地址指针的路由索引 102。映射程序层 3 106c 包括在 32 层二叉树
底层的节点 138 的主节点 h0 和在节点 140 的主节点 h1。对主节点的搜索
10 需要搜索关键字 104 的所有位。如图 2A 所示, h0 的路由索引 102 存
储在 L3 映射程序 106c 中的位置 146⁴中。与 L1 映射程序 106a 不同, L2
映射程序 106b 和 L3 映射程序 106c 不是直接映射。

在映射程序 106b 和 106c 中, 不为每个可能的输入而存储路由索引
102。仅当节点的路由索引 102 与映射程序 106b—c 中存储的前一个路由
15 索引 102 不同时才存储路由索引 102。请看映射程序层 2 112b 中所示的
第一子树 A 中的第三层节点, 节点 132¹和节点 132²的路由索引 102 是 r0,
因此 r0 对应的路由索引存储在 L2 映射程序 106b 中的节点 132¹和节点
132²对应的位置 142¹中。节点 132²的子树索引存储在位置 142²中。与
第三层节点 132⁴和第三层节点 132⁵和 132⁶有关的路由索引 102 为 r0,
20 与存储前一个节点 132²的 s0 不同, r0 存储在 L2 映射程序 106b 中的下
一个位置 142³。因为节点 132⁷不与前一节点 132⁶共享同一路由, 所以对
应节点 132⁷的路由 r2 存储在 L2 映射程序 106b 中的位置 142⁴中。为下
一个第三层的节点 132⁷存储子树索引 s3, 因此, s3 存储在 L2 映射程序
106b 中的位置 142⁵中。由于仅当与前一节点的路由不同时才存储路由索
25 引, 因此减少了用于存储路由索引 102 所需的内存。如图所示, 只需要
L2 映射程序 106b 中的 5 个位置来存储映射层 2 112b 中第一子树 A 的 8
个第三层节点 132¹⁻⁸对应的路由索引。对于非直接映射 106b、106c 将在
后面结合图 5 详细描述。

图 3 说明了根据本发明原理的对应 40 位关键字 210 的最长匹配前缀
30 查询表 200。在一个实施例中, 40 位关键字包括 8 位前缀和 32 位 IP 地

址。8 位前缀可以是与 32 位 IP 地址有关的虚拟专用网（“VPN”）的标识符。查询表 200 包括 4 个映射程序 106a—d。如结合图 2A 所描述过的，映射程序 106a 是直接映射的映射程序。映射程序 106b—d 是间接映射。映射程序 106a 存储路由索引 102 或对应 40 位关键字 210 的 16 个 MSB 的 L2 映射程序 106b 的子树索引。这样，L1 映射程序具有 64K 个可能的位置，第一映射程序层 112a（图 2B）的 64K 个节点中的每一个节点对应一个位置。存储在 L1 映射程序 106a 中对应位置的 L1 映射程序入口数据 220a 转发到管道 208 和 L2 间接映射程序 106b。如果 L1 映射程序入口数据 220a 表明下一层的搜索需要利用关键字 210b 接下来的 8 位，那么根据关键字 210b 接下来的 8 位和 L1 映射程序入口数据 220a 在 L2 间接映射程序 106b 中执行搜索。

第二层搜索的结果提供在转发到管道 208 和 L3 间接映射程序 106c 的 L2 映射程序入口数据 220b 上。根据关键字 210c 接下来的 8 位和 L2 映射程序入口数据 220b，在 L3 间接映射程序 106c 中执行第三层搜索。

搜索结果提供在转发到管道 208 和 L4 间接映射程序 106d 的 L3 映射程序入口数据 220c 上。L3 映射程序入口数据 220c 根据关键字 210d 的最后 8 位和 L3 映射程序入口数据 220c 来确定是否在 L4 间接映射程序 106d 中执行第四层搜索。

第四层的搜索结果提供在 L4 映射程序入口数据 220d 上。与关键字 210 的最长匹配前缀有关的路由索引 102 只存储在映射 106a—d 中的一个的一个位置。因此，转发到管道 208 的路由索引 102 只包含在映射程序入口数据 220a—d 之一中。如果在映射程序 106a—d 的其中一个中找到路由索引 102，例如在 106b 中，则不需要对余下的映射程序 106c—d 进行搜索和访问。管道 208 包括用来选择映射程序入口数据 220a—d 之一中包含的路由索引 102 的多路复用器 108（图 2A）。例如，映射程序入口数据 220a—d 的 MSB 能够提供是否包含路由索引的指示。

通过将管道 208 与映射程序 106a—d 结合使用，可以并行执行对具有不同关键字 210 的最长匹配前缀表 200 的多重搜索。管道 208 通过存储与 40 位关键字 210 有关的每个映射程序 106a—d 的映射程序入口数据 220a—d，允许并行执行 40 位查询表 200 的多重搜索，直到映射 106a—

d 的每个搜索都已完成，如果必要，寻找对应 40 位关键字 210 的路由索引。因此，对应所接收的 IP 地址的路由索引的搜索请求通过对直接映射的映射程序 106a 执行一次内存访问来发送给查询表 200。对应于另一个关键字的路由索引的后继搜索可以在直接映射的映射程序 106a 的下一个
5 内存访问循环时发送给查询表 200。

图 4 说明了能够存储在图 3 所示的直接映射的映射程序 106a 中的映射程序入口的类型。图 2B 所示的二叉树中任何节点对应的映射程序入口能够存储非入口 300，路由入口 302 或子树入口描述符 304。映射程序入口 300，302，304 中每一种类型包括一个子树标记 306。子树标记 306
10 的状态表示映射程序入口是否为子树入口描述符 304。如果子树标记 306 置为“1”，映射程序入口是子树描述符 304 并且包括子树索引 312。子树索引 312 是下一个非直接映射的映射程序 106b-d 中存储的子树入口描述符 304 的地址。子树入口将在后面结合图 4 进行描述。如果子树标记 306 为“0”，将检查非入口标记 314 来确定映射程序入口是非入口 300
15 还是路由入口 302。如果非入口标记 314 是“0”，该入口则是非入口 300。如果非入口标记 314 为“1”，该入口则是路由入口 302，并且在路由索引字段 310 存储中与关键字 104 有关的路由索引 102（图 3）。多路复用器 108（图 2A）利用子树标记 306 来选择包括路由索引 102（图 3）的映射程序入口数据 220a-d。

图 5 说明了对应图 2B 所示的映射层 2 112b 中的节点的映射程序
20 106b。映射 106b 包括子树内存 400，映射程序地址逻辑 402 和子树映射程序 418。把由存储在映射程序 106a 中的关键字 210a 的第一部分选择的子树索引 312 转发到子树内存 400。子树内存 400 包括由子树索引 312 选择的子树入口 404。子树入口 404 包括数据字段 406 和指针字段 408。

返回图 2B，子树入口 404 对应映射程序层 2 112b 所示的一个子树的
25 底层。如果映射程序层 2 112b 有 8 层，每个子树（未示出）的底层具有最多 256 条路由，每个节点一个路由。

继续到图 5，子树入口 404 提供对与子树底层上的每个节点对应的 256
30 个可能的路由索引 102（图 3）的访问。路由索引 102（图 3）存储在子树映射程序 418 中。为提供对 256 个可能的路由索引的访问，密集子树

描述符存储在数据字段 406 中。数据字段 406 有 256 位宽，为在子树底层的每个节点提供一位。数据字段将在后面结合图 6A 和图 6B 详细描述。指针字段 408 有 256 位宽，允许存储 16 个 16 位指针，每个指针存储子树映射程序 418 中 16 个相邻映射程序入口的基地址，以便提供对 256 个路由索引的访问。因此，指针字段 408 可以为子树底层中的每个节点间接地提供到子树映射程序 418 中的映射程序入口的指针。将结合图 6 详细描述指针字段 408。

把存储在数据字段 406 的密集子树描述符中的子树数据 412 和存储在指针字段 408 中的子树指针 414 转发到映射程序地址逻辑 402。映射程序地址逻辑 402 还接收关键字 210b 的下一部分（接下来的 8 位）。

映射程序地址逻辑 402 根据与子树有关的关键字 212b 接下来的 8 位，子树数据 412 和子树指针 414 来确定与子树底层中的节点有关的映射程序入口的映射程序地址 416。映射程序地址 416 在子树映射程序 418 中选择映射程序入口。子树映射程序 418 包括直接映射的映射程序 106a 的与结合图 4 所述类型相同的映射程序入口。映射程序数据入口 220b 的内容确定是否需要后继搜索。如果映射程序入口数据 220b 包括表明在下一个映射层 112c（图 2B）还有另一个子树入口 404 的子树索引 312（图 4），则需要后继搜索。

关键字 210b 的第二部分选择所选子树底层中的节点。子树指针 414 选择与子树中的节点有关的基地址，子树数据 412 选择与基地址有关的映射入口模块中的位移。映射程序地址逻辑 402 将结合图 7 在后面描述。

图 6A 是子树的二叉树表示。所示子树包括 5 层。该子树包括三个路由索引 r_1 ， r_2 和 r_3 以及两个子树索引 s_0 和 s_1 。子树底层上有 32 个节点 500^1-500^{32} 。下面的表 1 显示了与底层中的每个节点 500^1-500^{32} 有关的路由索引或子树索引。

	子树位	路由/子树
	00000	r0
	00001	r0
5	00010	r0
	00011	r0
	00100	r1
	00101	r1
	00110	r0
10	00111	r0
	01000	r2
	01001	s0
	01010	r2
	01011	r2
15	01100	r2
	01110	r2
	01111	r2
	1xxxx	r3

表 1

20

图 6 说明了对应图 6A 所示的子树底层中的节点的图 5 所示的数据字段 406 中存储的密集子树描述符。数据字段 406 包括 32 位，图 6A 所示的子树底层中的每个节点 500 对应一位。位 502^1-502^{32} 在数据字段 406 中是这样分配的：如果要使用前一个节点的路由索引，那么数据字段中的一位设定为 ‘0’，如果使用存储在子树映射程序 418（图 5）中的下一个路由索引，那么设定 ‘1’ 递增到下一个映射入口地址。除非指定了路由，否则数据字段 402 中的第一位选择存储在映射程序入口 504^1 中的缺省路由 r0。这样，由于没有指定的路由，把位 502^1 设定为 ‘0’ 来选择缺省路由。为接下来三个节点 500^2-500^4 选择映射程序入口 504^1 中

25

30 存储的缺省路由 r0，因此，在数据字段 406 中的对应位 502^2-502^4 被设

定为“0”，以使用 502¹ 使用的前一个路由索引。在节点 500⁵ 存在路由改变。

节点 500⁶ 共享用于映射程序入口 504² 中存储的节点 500⁵ 的路由 r1。因此位 502⁵ 是‘1’表示路由改变，以便在子树映射程序 418（图 5）中选择映射程序入口 504²。位 502⁶ 是‘0’表示把 502⁵ 中存储的路由索引用于该节点。不为节点 500⁷ 提供路由，因此，存在着路由变化，并且在位 502⁷ 中存储‘1’来表示需要在子树映射程序 418（图 5）中的映射程序入口 504³ 存储缺省路由 r0。

节点 500⁸ 与前一个节点 500⁷ 共享同一个路由，在子树映射程序 418（图 5）中不需要新的映射程序入口。对应节点 500⁸ 的位 502⁸ 设定为“0”。节点 500⁹ 与前一个节点 500⁸ 具有不同的路由，因此在子树映射程序 418（图 5）中需要新的映射程序入口。对应节点 500⁹ 的位 502⁹ 设定为“1”，并且将存储 r2 的映射程序入口 504⁴ 添加到下一个相邻存储单元中的子树映射程序 418（图 5）。

节点 500¹⁰ 与前一个节点 500⁹ 具有不同的路由，在子树映射程序 418（图 5）中需要新的路由入口。对应节点 500¹⁰ 的位 502¹⁰ 设定为“1”，并且将存储 s0 的映射程序入口 504⁵ 添加到下一个相邻存储单元中的子树映射程序 418（图 5）。

节点 500¹¹ 与前一个节点 500¹⁰ 具有不同的路由，在子树映射程序 418（图 5）中需要新的映射程序入口。对应节点 500¹¹ 的位 502¹¹ 设定为“1”，并且将存储 r2 的映射程序入口 504⁶ 添加到下一个相邻存储单元中的子树映射程序 418（图 5）。

节点 500¹² 和 500¹³ 共享与前一个节点 500¹¹ 相同的路由，在子树映射程序 418（图 5）中不需要新的映射程序入口。在数据字段 406 中把对应于节点 500¹² 的位 502¹² 和对应于节点 500¹³ 的位 502¹³ 设定为‘0’。

节点 500¹⁴ 与前一个节点 500¹³ 具有不同的路由，在子树映射程序 418（图 5）中需要新的映射程序入口。数据字段 406 中对应节点 500¹⁴ 的位 502¹⁴ 设定为‘1’，并且把存储 s1 的映射程序入口 504⁷ 添加到下一个相邻存储单元中的子树映射程序 418（图 5）。节点 500¹⁵ 与前一个节点 500¹⁴ 具有不同的路由，在子树映射程序 418（图 5）中需要新的映射程序入口。

数据字段 406 中对应节点 500^{15} 的位 502^{15} 设定为 ‘1’，并且将存储 $r2$ 的映射程序入口 504^8 添加到下一个相邻存储单元中的子树映射程序 418（图 5）。节点 500^{16} 与前一个节点 500^{15} 共享同一个路由，在子树映射程序 418（图 5）中不需要新的映射程序入口。

- 5 节点 500^{17} 与前一个节点 500^{16} 具有不同的路由，在子树映射程序 418（图 5）中需要新的映射程序入口。将数据字段 406 中对应节点 500^{17} 的位 502^{17} 设定为 ‘1’，并且将存储 $r3$ 的映射程序入口 504^9 添加到下一个相邻存储单元中的子树映射 418（图 5）。

10 节点 $500^{18}-500^{32}$ 全部与前一个节点 500^{17} 共享相同的路由，在子树映射程序 418（图 5）中不需要新的映射程序入口。将对应的位 $502^{18}-502^{32}$ 设定为 ‘0’。这样就需要 9 个映射程序入口 504^{1-9} 来存储 32 个节点 500^1-500^{32} 的路由入口 302（图 4）或子树入口 304（图 4）。

15 通过计算在数据字段 406 中存储的密集子树描述符中存储的 ‘1’ 的数量在子树映射程序 418（图 5）中索引与节点 500^1-500^{32} 对应的映射程序入口 504^1-504^9 。例如，为寻找对应节点 500^{28} 的映射程序入口 504^1-504^9 ，计算数据字段 406 的位 502^1-502^{28} 中存储 ‘1’ 的数量。‘1’ 的数量是 8，并且对应的映射程序入口是从缺省路由开始的第八个位置，即映射程序入口 504^9 。

20 仅当路由发生变化时才存储映射程序入口的方法减少了子树映射程序 418（图 5）中每个子树映射程序入口 504^1-504^9 的数量。

25 图 7 说明了图 5 所示的指针字段 408。指针字段 408 包括用于存储子树映射程序 418（图 5）中几组 16 个相邻映射程序入口位置 504^1-504^{16} 组（图 6B）的基地址的块基地址字段 $600^1, 600^2$ 。在 16 个相邻映射程序入口的组 $602^1, 602^2$ 的子树映射程序 418（图 5）中分配内存。8 层的子树可以具有多达 256 个不同路由，需要 16 个组 $602^1, 602^2$ 来存储所有 256 个路由。根据子树的路由数量来确定所需组 602 的数量。通过从块基地址的空闲列表中（未显示）删除块基地址 $602^1, 602^2$ 来将组 602 分配给特定的子树。为内存提供地址空闲列表的方法是已知的技术。

30 通过分配 16 个映射程序入口 504^{1-16} 的内存块，由于所分配的 16 个位置是相邻的，所以子树映射程序 418（图 5）中的内存易于管理。

图 8 说明了图 5 所示的映射程序地址逻辑 402。映射程序地址逻辑 402 包括位移逻辑 700，指针逻辑 702 和加法器逻辑 704。位移逻辑 700 包括节点选择逻辑 706 和“1”计数逻辑 708。指针逻辑包括基地址选择逻辑 710。

5 节点选择逻辑 706 选择在子树数据 412 中对应 8 位关键字 210b 的节点 500 (图 6B)。相应的节点数通过节点选择 718 转发到‘1’计数逻辑 708。‘1’计数逻辑 708 累计子树数据字段 406 中存储的‘1’的数量，直到与所选节点 500 对应的位。‘1’的数量通过块选择 712 转发到指针逻辑 702 并且通过块位移 714 转发到加法器逻辑 704。

10 需要 8 位宽的计数字段的 256 位子树数据字段 406 中最大可存储 256 个“1”。8 位计数字段分为两个字段，提供块选择 712 的 4 个 MSB 和提供块位移 714 的 4 个最低有效位 (“LSB”)。

例如，如果 8 位的关键字 210b 为‘01000100’，为选择节点数 68 并且共有 27 个“1”存储在子树数据 42 的前 68 位，计数结果为 IC Hex
15 (0001 1100)，MSB (0001)；即块选择 714 选择块 602¹ (图 6) 和 LSB (1100)；即基块位移选择映射程序入口 504¹¹ (图 6)，即块 502¹ 中的第 12 个入口。

基地址选择逻辑 710 根据从位移逻辑 700 转发的块选择 712 从子树指针 414 中选择基地址 716。加法器逻辑 704 将从位移逻辑 700 转发的
20 块位移 714 累加到基地址 716，并且提供映射程序地址 416。映射程序地址 416 是映射程序 106b-d 中映射程序入口 504 (图 6B) 的索引。

图 9 是根据本发明原理用于在查询表 200 中搜索关键字 210 (图 3) 的最长匹配前缀的步骤的流程图。

25 在步骤 800 中，把关键字 210a (图 3) 的第一部分作为索引转发到映射程序 106a。处理过程继续到步骤 802。

在步骤 802 中，由关键字 210a (图 3) 的第一部分索引的第一层映射程序中的映射程序入口 504 (图 6B) 中存储的映射程序入口数据 220a (图 3) 确定是否需要继续对下一层进行搜索。如果需要，处理过程继续到步骤 804。否则，第一层映射程序中索引的映射程序入口 504 (图 6B)
30 中的路由入口 302 (图 4) 存储该关键字的最长前缀路由并且处理过程继

续到步骤 808。

在步骤 804，搜索下一层映射程序 106b-d。下一层映射程序的索引取决于前一层映射程序中索引的映射程序入口 504（图 6B）中子树入口描述符 304（图 4）中存储的子树索引 312 和关键字 210b-d 的下一部分。

5 处理过程继续到步骤 806。

在步骤 806 中，下一层映射程序中的索引映射程序入口 504（图 6B）存储该关键字对应的最长前缀路由索引或表示需要进一步搜索的子树索引。如果需要进一步搜索，处理过程继续到步骤 804。否则，处理过程继续到步骤 808。

10 在步骤 808，把映射程序 106a-d 之一中的一个映射程序入口 504（图 6B）中存储的路由索引 102（图 3）作为路由索引 102（图 3）从查询表 200 转发。处理过程结束。

深度扩展

15 能够存储在图 3 所示查询表 200 中的路由索引 102（图 3）的数量受子树映射程序 418（图 5）中可用映射程序入口 504（图 6B）的数量的限制。例如，如果每个子树映射程序 418（图 5）包括 128K 个映射程序入口并且在查询表中有两个子树映射程序 418（图 5），那么查询表 200 中最多可以存储 256K 个路由索引 102（图 3）。具有 128K 个映射程序入口的子树映射程序 418（图 5）需要一个 17 位的索引。具有 512K 个映射程序入口的子树映射程序 418（图 5）需要一个 19 位的索引。查询表 200 中 2 个 512K 的子树映射程序 418（图 5）为 32 位 IPv4 目的地址的 40 亿个可能的路由索引 102（图 3）中 1 百万个提供了存储空间。

25 通过提供多个查询表 200 可以增加用于存储路由索引 102（图 3）的映射程序入口 504（图 6B）的数量。可以在多个查询表中并行地搜索与查询表 200 之一中的子树映射程序 418（图 5）中的映射程序入口 504（图 6B）中存储的搜索关键字对应的值。

30 图 10A 说明了深度扩展的实施例。示出两个查询表，一个为主查询表 200a，另一个为副查询表 200b。但是，查询表的数量不只限于所示的两个，可以增加一个以上的副查询表 200b。

以相同的搜索关键字 210 并行搜索查询表 200a-d 中的每一个。对应于搜索关键字 210 的路由索引 102 (图 3) 存储在查询表 200a-d 之一中的子树映射程序 418 (图 5) 中, 或查询表 200a 和 200b 二者中的直接映射程序 106a 中。在完成对查询表 200a 和 200b 的并行搜索后, 找到最终路由索引 900。

图 10B 说明了图 10A 所示实施例中的查询表 200a 中的一个。查询表 200a-b 中的每一个包括映射程序 106a-d 和已经结合图 3 针对查询表 200 描述过的管道 208, 以及驱动器逻辑 902。查询表 200a 为与搜索关键字对应的路由索引在映射程序 106a-d 中执行多层搜索。每一层的搜索结果通过映射程序入口数据 220a-d 转发到管道 208。管道 208 再将搜索结果 904 转发到驱动器逻辑 902。每个查询表 200a-b 中映射程序 106a 中的映射程序入口 504 (图 6B) 存储子树入口 304 (图 4), 但只有主查询表 200a 的映射程序 106a 存储路由入口 302 (图 4)。副查询表 200b 中映射程序 106a 中的映射程序入口 504 (图 6B) 代替路由入口 302 存储非入口 300 (图 4)。将映射程序 106a 中的路由索引集中在一个查询表避免了为提供最终路由索引 900 的查询表的选择过程。这导致了副查询表 200b 中不能用于存储路由索引而只允许存储相同查询表的 64K 内存配置成主查询表或结合图 3 所述的副查询表。在另一个实施例中, 可以提供没有映射程序 106a 的副查询设备。如果路由索引 102 (图 3) 存储在映射程序 106a 中的路由入口 302 (图 4), 那么搜索在主查询表 200a 中的映射程序 106a 中结束。

如图 10A 所示, 主查询表 200a 和副查询表 200b 共享最终路由索引 900。存储了最终路由索引 900 的查询表 200a, 200b 提供路由索引 102 (图 3)。如果每个查询表 200a, 200b 是独立的设备, 共享最终路由索引 900 减小了每个设备的外部引脚数量。只有查询表 200a, 200b 中的一个随时驱动最终路由索引。

为避免两个查询表 200a, 200b 中存储与搜索关键字 210 有关的路由索引而会使两个查询表同时驱动最终路由索引 900 的错误状态, 每个查询表 200a, 200b 存储一个设备代码。3 位的设备代码允许扩展的查询表包括 8 个设备。

驱动器逻辑 902 确定搜索结果 904 是否包括路由索引 102 (图 3)。
如果包括, 查询表 200a 中的驱动器逻辑 902 发出试图通过总线请求信号
(未示出) 驱动最终路由索引 900 的信号。如果两个或更多的查询表
200a, 200b 同时发出试图驱动路由索引信号的信号, 那么由具有最低设
5 备代码的查询表 200a, 200b 中提供路由索引。通过使用总线请求信号来
解决总线冲突的方法是熟知的技术。

图 10C 说明了用来提供深度扩展以增加可用于存储与搜索关键字 210
对应的值的映射程序入口数量的另一个实施例。在图 10C 所示的实施例
中, 提供两个查询表 200c-d 来存储值, 主表 200c 和副表 200d。但是
10 查询表的数量不只限于所示的两个, 可以通过加入更多的副表 200d 来增
加映射程序入口的数量。在查询表 200c-d 中并行执行对与搜索关键字
[39: 0]210 对应的查询表 200c-d 之一中的映射程序入口中所存储的值的
搜索。

图 10D 说明了图 10C 所示的实施例中的副表 200d。每个查询表包括
15 结合图 3 针对查询表 200 描述的映射程序 106a-d。每个查询表 200c-d
中映射程序 106a 中的映射程序入口存储子树入口 304 (图 4)。每个查
询表 200c-d 将映射程序 106a 中映射程序入口 504 (图 6B) 里存储的子
树入口描述符 304 (图 4) 中存储的子树索引 312 转发到下一个映射程序
106b-d。但是, 路由索引 102 (图 3) 只存储在主查询表 200c 的映射程
20 序 106a 中。无入口存储在副查询表 200d 中的映射程序 106a 中以避免在
一个以上的查询表 200b, 200d 中存储与该关键字对应的路由索引。

副查询表 200d 中的多层搜索结果 904 转发到最终索引逻辑 1004。
最终索引逻辑 1004 将多层搜索结果 904 或从主查询表 200c 转发来的输
入结果 1000a 作为输出结果 1002a 转发。如果路由索引 102 (图 3) 包括
25 在多层搜索结果 904 中, 那么将多层搜索的结果作为输出结果 1002a 转
发。如果路由索引 102 (图 3) 包括在输入结果 1000a 中, 那么将输入结
果 1000a 作为输出结果 1002a 转发。如果路由索引 102 (图 3) 既不包
括在输入结果 1000a, 也不包括在多层搜索结果 904 中, 那么将多层搜索
结果 904 作为输出结果 1002a 转发。

30 如图 10C 所示, 主查询表 200c 和副查询表 200d 通过标有输入结果

1000a 的公共总线相连。路由索引 102 (图 3) 只通过输出结果 1002a 从副查询表 200d 转发。如果有一个以上的副查询表 200d, 最后一个副查询表提供扩展查询表的路由索引 102 (图 3)。本实施例可以避免实现结合图 10A 所述的多驱动器最终路由索引 900 的出现, 但需要更多用于输入结果 1000a 的设备外部引脚。

图 11A—B 说明了在查询表 200a—b (图 10A) 或 200c—d (图 10C) 中分配图 2B 所示路由的二叉树表示。

图 11A 说明了主查询表 200a (图 10A) 或 200c (图 10C) 中存储的路由的二叉树表示。主查询表 200a 中不包括图 2B 所示路由的二叉树表示中所示的子树 B。节点 130^1-130^{22} 和 130^{24-32} 在结合图 3 所述的查询表 200a 中的映射程序 106a 中编码。如果存储在主查询表 200a 中, 那么在那里索引子树 B 的节点用“X”图形化地表示为已修剪子树。在主查询表 200a 中对应节点 130^{23} 的映射程序入口 504 (图 6B) 不再存储指向子树 B 的子树索引 312 (图 4)。取而代之的是非入口 300 (图 4) 存储在主查询表 200a 中对应节点 130^{23} 的映射程序入口 504 (图 6B) 中, 以表示对应节点 130^{23} 的映射程序入口存储在另一个副查询表 200b 的子树映射程序 418 (图 5) 中。

图 11B 说明了存储在副查询表 200b (图 10A) 或 200d (图 10C) 中子树映射程序 418 (图 5) 中的映射程序入口 504 (图 6B) 的路由的二叉树表示。与图 2B 所示二叉树表示不同, 存储在副查询表 200b 中的路由的二叉树表示不包括子树 A。因此, 节点 130^1-130^3 和 130^5-130^{32} 以如结合图 2B 所述的那样进行编码。对应副查询表 200b 中节点 130^4 的映射程序入口 504 (图 6B) 不再存储指向子树 A 的子树索引 312 (图 4)。取而代之的是对应副查询表 200b 中节点 130^4 的映射程序入口存储非入口 300 (图 4) 以表示对应节点 130^4 的映射程序入口存储在另一个查询表中。子树 A 对应的子树索引和主机 138 (图 11A) 对应的路由索引存储在主查询表 200a 中, 并且子树 B 对应的子树索引和主机 140 (图 11A) 对应的路由索引存储在副查询表 200b 中。副查询表 200b, 200d 只存储子树的结果; 就是说, 副查询表 200b, 200d 不存储第一层映射程序 106a 中的结果。

参考图 11A 和图 11B, 利用关键字 210a 的第一部分对主映射程序层 1 1102a (图 3) 或副映射程序层 1 1104a 中任何节点 130^9-130^{12} 的搜索产生了主查询表 200a, 200c 中映射程序 106a 中映射程序入口 504 (图 6B) 中路由入口 302 (图 4) 中存储的 r1 116 以及副查询表 200b, 200d 5 中映射程序 106a 中映射程序入口 504 (图 6B) 中存储的非入口 300 (图 4)。主查询表 200a, 200c 中存储的路由入口 302 (图 4) 通过输入结果 1000a 转发到副查询表 200b, 200d 并且通过输出结果 1002a 由副查询表 200b, 200d 转发。

利用关键字 210a 的第一部分对节点 130^4 的搜索产生了主查询表 200a 10 中映射程序 106a 的映射程序入口 504 (图 6B) 中子树入口描述符 304 (图 4) 中存储的子树 A 的子树索引 312 (图 4)。子树索引 312 转发到主查询表 200a 中的映射程序 106b 来继续对主查询表 200a 中存储的路由入口 302 (图 4) 进行搜索。

利用关键字 210 的第一部分对节点 130^{23} 的搜索产生主查询表 200a 15 中映射程序 106a 的映射程序入口 504 (图 6B) 中存储的非入口 300 (图 4) 以及副查询表 200b 中映射程序 106a 的映射程序入口 504 (图 6B) 中存储子的树入口描述符 304 (图 4)。因此, 继续利用关键字 210b 的下一部分在副查询表 200b 的映射程序 106b 中搜索路由入口 302 (图 4)。

图 12 是说明图 10A 所示查询表 200a-b 中映射程序入口 504 (图 6B) 20 中存储的路由入口 302 (图 4) 的分配方法的流程图。同样的方法用于图 10C 所示的查询表 200c-d。在查询表 200a-b 中存储之前, 最初由处理器 (未示出) 在存储器中存储映射程序入口中要存储的路由入口 302 (图 4)。

当路由入口 302 (图 4) 存储在存储器中时, 计算每个查询表 200a 25 -b (图 10A) 中要存储的路由入口 302 (图 4) 的数量。映射程序层 1 1104a (图 11B) 的路由入口 302 (图 4) 存储在查询表 200a 的映射程序 106a 中。映射程序层 1 1104a (图 11B) 的子树入口 304 (图 4) 存储在每个查询表 200a-200b 的映射程序 106a 中。

在步骤 1200 中, 为确定如何在查询表 200a-b (图 10A) 中分配路 30 由入口 302 (图 4), 需要计算每个查询表 200a-200b (图 10A) 中映射

程序 106a 的每个子树入口 304 (图 4) 中要存储的路由入口 302 (图 4) 的数量。当确定了存储路由入口 302 (图 4) 所需的映射程序入口 504 (图 6B) 的总数之后, 处理过程继续到步骤 1202。

在步骤 1202 中, 将要存储的子树的映射程序入口 504 (图 6B) 的总数除以查询表 200a-b (图 10A) 的数量以确定存储在每个查询表 200a-b (图 10A) 中的路由入口 302 (图 4) 的数量。处理过程继续到步骤 1204。

在步骤 1204 中, 在所选查询表 200a-b 中子树映射程序 418 (图 5) 的映射程序入口 504 (图 6B) 中存储路由入口 302 (图 4)。处理过程继续到步骤 1206。

10 在步骤 1206 中, 如果在所选查询表 200a-b 中子树映射程序 418 (图 5) 的映射程序入口 504 (图 6B) 中存储的路由入口的数量小于 $1/n$, 其中 n 表示可用查询表 200a-b (图 10A) 的数量, 那么处理过程继续到步骤 1208。否则, 所选查询表 200a-b 已经存储了映射程序入口总数的 $1/n$, 并且处理过程继续到步骤 1208。

15 在步骤 1208 中, 所选查询表 200a-b 存储映射程序入口总数的 $1/n$, 因为相应子树的路由索引不存储在当前所选的查询表中, 所以任何剩余子树节点的非入口 300 (图 4) 存储在所选择的查询表 200a-b 中。处理过程继续到步骤 1210。

在步骤 1210 中, 如果已经存储了全部路由入口, 处理过程结束。否则, 处理过程继续到步骤 1212。

在步骤 1212 中, 选择下一个查询表 200a-b (图 10A)。处理过程继续到步骤 1204。

在搜索对应于 IP 地址的路由索引之前, 在查询表 200a-b (图 10A) 中分配路由入口。搜索是在查询表 200a-b (图 10A) 中的每一个中并行执行的。用查询表 200a-b (图 10A) 中的一个来描述实现对每个查询表并行搜索的方法。

图 13 是说明利用搜索关键字搜索与图 10C 所示的查询表 200c-d 中的任何一个存储的搜索关键字对应值的方法的流程图。

在步骤 1300 中, 查询表 200c-d (图 10C) 中的每一个接收搜索关键字 210。查询表 200c-d 中的每一个中的映射程序 106a 搜索于关键字

210a 的第一部分对应的值。处理过程继续到步骤 1302。

在步骤 1302 中，读取映射程序 106a 中映射程序入口 504（图 6B）中存储的入口。主查询表 200c 中的映射程序入口 504（图 6B）可以存储非入口 300（图 4），路由入口 302（图 4）或子树入口描述符 304（图 4）。副查询表 200d 中的映射程序入口 504 可以存储非入口 300（图 4）和子树入口描述符 304（图 4）。如果相应的查询表 200 中的映射程序入口存储路由入口 302（图 4），该入口则是有效值，不需要进一步搜索查询表 200c-d 中的后继映射程序 106b-d，处理过程继续到步骤 1310。否则，处理过程继续到步骤 1304。

在步骤 1304 中，如果入口存储了子树入口描述符 304（图 4），那么需要进一步搜索查询表 200c-d 并且处理过程继续到步骤 1306。否则，该入口存储非入口，表示不需要进一步的搜索，处理过程继续到步骤 1310。

在步骤 1306 中，在所选子树中继续搜索。根据关键字 210b-d 的下一部分和从上一层的搜索得到的子树索引 312（图 4）来搜索下一层映射程序 106b-d（图 3）。处理过程继续到步骤 1308。

在步骤 1308 中，根据从当前层映射程序 106b-d 中搜索得到的映射程序入口来确定是否继续执行搜索。如果映射程序入口 504（图 6B）存储了子树入口描述符 304（图 4），搜索继续到下一层映射程序 106b-d 并且处理过程继续到步骤 1306。如果映射程序入口 504（图 6B）没有存储子树入口描述符 304（图 4），则不需要作进一步搜索，处理过程继续到步骤 1310。

在步骤 1310 中，把搜索结果与从另一个查询表接收的输入结果作比较。例如，如果查询表是副查询表 200d，那么把从主查询表 200c 中搜索产生的输入结果通过输入结果 1000a 转发到查询表 200d，并且与副查询表 200d 的搜索结果作比较。处理过程继续到步骤 1312。

在步骤 1312 中，如果输入结果 1000a 与当前查询表 200d 中的搜索结果不同，处理过程继续到步骤 1314。如果输入结果 1000a 与当前查询表 200d 中的搜索结果相同，在分开的查询表 200c-d 中的映射程序入口 504（图 6B）中已经存储这两个有效的结果。不应该相同的关键字 210

来存储两个有效结果，处理过程继续到步骤 1316。

在步骤 1314 中，检查输入结果 1000a 来确定它是否有效。如果输入结果 1000a 是路由入口 302(图 4)，那么它是有效的。如果输入结果 1000a 是非入口 300 (图 4) 或子树入口描述符 304 (图 4)，那么它是无效的。

5 子树入口描述符 304 (图 4)，路由入口 302 (图 4) 和非入口 300 (图 4) 已经结合图 4 描述过了。如果输入结果 1000a 无效，处理过程继续到步骤 1318。否则，处理过程继续到步骤 1320。

在步骤 1318 中，输入结果 1000a 有效，而从当前查询表 200d 中搜索得到的结果无效。输入结果 1000a 通过输出结果 1002a 从当前查询表
10 200d 转发。如果当前查询表 200d 是最后一个查询表，那么输入结果 1000a 作为路由索引 102 (图 3) 转发，或者作为输入结果 1000a 转发到下一个查询表。处理过程结束。

在步骤 1316 中，在不同的查询表中存储关键字的两个有效的结果
15 值。在查询表 200c—d 中存储路由入口期间发生错误时，生成错误代码以便能够纠错。处理过程结束。

在步骤 1320 中，来自当前查询表 200d 的搜索结果和输入结果 1000a 都无效。即使无效，也将当前查询表 200d 的搜索结果作为输入结果 1000a 转发到下一个查询表。处理过程结束。

图 14 是说明搜索与图 10A 所示的查询表 200a—b 之一中存储的搜索
20 关键字对应的值的方法的流程图。

在步骤 1340 中，在两个查询表 200a—b 中的第一层映射程序 106a 搜索与关键字 210a 的第一部分对应的值。处理过程继续到步骤 1342。

在步骤 1342 中，如果利用关键字 210a 的第一部分搜索第一层映射
25 程序 106a 后找到了有效的结果值，则处理过程继续到步骤 1352。否则，处理过程继续到步骤 1344。

在步骤 1344 中，如果利用关键字 210a 的第一部分搜索第一层映射
程序 106a 得到的值是子树入口描述符 304 (图 4)，处理过程继续到步
骤 1346。否则，当前查询表中不存储关键字的有效值，处理过程结束。

在步骤 1346 中，继续在以搜索上一层映射程序期间找到的子树入口
30 描述符 304 (图 4) 中标识的子树中搜索有效值。根据关键字 210b—c 的

下一部分和从搜索下一层得到的子树选择在下一层映射程序中搜索一个值。处理过程继续到步骤 1348。

在步骤 1348 中，搜索结果确定是否需要到下一层进行搜索。从当前搜索得到的入口可以存储路由入口 302，非入口 300（图 4）或子树入口描述符 304（图 4）。如果入口存储了子树入口描述符 304（图 4），则需要进一步的搜索并且处理过程继续到步骤 1346。如果入口没有存储子树入口描述符 304（图 4），处理过程继续到步骤 1350。

在步骤 1350 中，如果入口存储了路由索引 102（图 3），处理过程继续到步骤 1352。否则，将入口存储在另一个查询表中。处理过程结束。

在步骤 1352 中，与该关键字对应的有效值存储在当前查询表中。有效值作为对应于该关键字的路由索引 102（图 3）转发。处理过程结束。

稀疏模式

返回图 5，子树入口 404 对多达 256 个可能的路由索引中的每一个提供访问，256 个节点字树中每个节点一个。路由索引存储在子树映射程序 418（图 5）中的映射程序入口 504^1-504^n 中。根据子树入口 404 中数据字段 406 中存储的密集子树描述符和指针字段 408 中存储的子树指针确定在子树映射程序 418（图 5）中映射程序入口 504（图 6B）的映射程序地址 416。密集子树描述符的格式已经结合图 6A-6B 描述过了。密集子树描述符为 256 个节点的子树中每个节点存储一个节点位 502（图 6B）。但是，所有子树针对 256 个节点中的每一个具有不同的路由索引，例如，一个子树可能只有一个路由索引。

图 15 说明了由第一映射程序层 112a 中的子树入口 304 索引的第二层映射程序层 112b 中稀疏子树 A 和密集子树 B 的二叉树表示。映射程序 106a 中 s1 的子树入口描述符 304（图 4）存储子树 A 的子树入口 404 的子树索引 312。映射程序 106a 中 s0 的子树入口描述符 304（图 4）存储子树 B 的子树入口 404 的子树索引 312。

密集集中子树 B 有 11 个路由索引；即，r6-r16 和 6 个子树入口；即，s2-s7。与存储路由入口 302（图 4）的映射程序入口 504（图 6B）和子树 B 的子树入口 304（图 4）对应的映射程序地址 416 在密集子树描

述符中的编码已经结合图 6B 描述过了。

稀疏集中子树存储两个路由索引；即 r_1 和 r_2 。如果它们存储在密集子树描述符中，那么整个子树入口 404 用来提供映射程序入口 504（图 6B）的三个映射程序地址 416；即， r_0 ， r_1 和 r_2 。

5 查询表 200 中存储的路由数量可以通过对多个稀疏子树描述符的其中一个稀疏子树描述符编码以及对子树入口 404 中密集子树描述符中的密集集中子树描述符编码来增加。

密集集中子树有 16 个或更多个映射程序入口 504（图 6B），子树入口 404 中的数据字段 406 存储结合图 6A—6B 所述的密集子树描述符。稀疏集中子树有 15 个或更少个映射程序入口 504（图 6B）；子树入口 404 中的数据字段存储了多个稀疏子树描述符。通过提供在稀疏子树描述符中存储稀疏集中子树的能力，子树存储器 400 可以存储更多的子树并且查询表 200 中可以存储更多的路由入口。

图 16A—C 说明了对图 5 所示子树入口 404 中数据字段 406 和指针字段 408 以及图 4 所示子树入口描述符 304（图 4）的更改，以便允许在子树入口 404 中存储多个稀疏子树描述符。

转到图 16A，以稀疏模式配置的子树入口 404 中的数据字段 406 包括多个稀疏子树描述符 1400^1-1400^n 来代替结合图 6B 所述的子树的每个节点对应一位的密集子树描述符。每个稀疏子树描述符 1400^1-1400^n 包括一个节点描述符 1402^1-1402^n 。一个节点描述符 1400^1-1402^n 是代表子树中完全已编码路由的一个 9 位值。节点描述符 1402^1-1402^n 描述了子树中一个节点或多个节点。

转到图 16B，为支持存储稀疏子树描述符，在子树入口 404 的指针字段 408 增加模式字段 1404。指针字段 408 还存储块基地址 600^1 和块基地址 600^2 ，每个块包括 16 个为每个子树入口 404 提供总共 32 个映射程序地址 416 的已分配映射程序地址 416。模式字段 1404 存储模式值。模式字段 1404 中存储的模式值表示子树入口 404 中存储的稀疏子树描述符 1400^1-1400^n 的数量和每个稀疏子树描述符 1400^1-1400^n 中存储的节点描述符 1402^1-1402^n 的数量。表 2 说明了每个模式对应的子树入口 404 的配置。

模式	节点/子树		位	子树/ 子树 入口	路由/子树入 口		未 用	子树入口需要的 映射程序入口	
	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

参见表 2，例如，如果设置子树入口 404 中指针字段 408 的模式字
 5 段 1404 中存储的模式值为“4”，那么子树入口 404 中的每个稀疏子树
 描述符 1400 存储 5 到 7 个节点描述符 1402^1-1402^n 。每个节点描述符 1402^1-1402^n
 存储 9 位。通过每个稀疏子树描述符 1400^1-1400^n 的节点描述符
 1402¹-1402ⁿ 的数量乘以 9（每个节点描述符 1402^1-1402^n 的位数）来计
 10 算存储在稀疏子树描述符 1400 中的总位数。计算模式 4 的每个稀疏子树
 描述符 1400 的位数后，具有 7 个节点描述符 1402 的稀疏子树描述符 1400
 存储 63 位（7 个节点描述符 × 9 位 = 63）。

数据字段 406 中的位数除以稀疏子树描述符 1400^1-1400^n 的位数可
 以计算出每个子树入口 404 的稀疏子树描述符 1400 的数量。对于模式 4，
 数据字段 406 中的位数为 256 并且稀疏子树描述符中的位数为 63。因此
 15 稀疏子树描述符 1400^1-1400^n 的数量为 4（ $\text{int}(256/63)=4$ ）。

每个子树的节点数乘以每个子树入口 404 的子树数量可以得到每个
 子树入口 404 的节点描述符 1402^1-1402^n 的总数。计算模式 4，如果在
 稀疏子树描述符 1400^1-1400^n 中存储了 7 个节点描述符 1402^1-1402^n ，
 那么每个子树入口 404 的节点描述符 1402 的数量为 28（ $7 \times 4 = 28$ ），

并且如果每个子树描述符 1400^1-1400^n 有 5 个节点描述符 1402, 则结果为 20 ($5 \times 4 = 20$)。

表 2 中的映射程序入口列表示子树入口 404 使用了多少子树映射程序 418 (图 5) 中的映射程序入口 504 (图 6B)。通过将每个子树的节点
5 加 1 并通过乘以稀疏子树描述符中的子树数量来计算映射程序值。每个子树节点加 1 是因为需要比每个子树节点数量多 1 个映射程序入口, 以便存储子树的缺省入口。

参见表 2 中模式 4 的一行, 如果每个稀疏子树描述符 1400 有 7 个节点描述符 1402, 那么每个子树入口 404 需要 32 ($(7+1) \times 4 = 32$) 个
10 映射程序入口, 并且如果每个稀疏子树描述符 1400 有 5 个节点描述符 1402, 那么每个稀疏子树描述符 1400 需要 24 ($(5+1) \times 4 = 24$) 个节点描述符 1402。因为在 16 个块递增中分配了子树映射程序 418 (图 5) 中的映射程序地址 416, 所以选择每个子树节点的数量和每个子树入口 404 的子树数量, 以便每个子树入口 404 的节点描述符的最大数量不超
15 过 30。通过在指针字段 408 中存储两个块基地址 $600^1, 600^2$ 来提供 32 个映射程序地址 416。

转到图 16C, 子树存储器 400 中的每个子树入口 404 可以配置成如结合图 6B 所述的密集模式或稀疏模式。通过提供表示子树入口 404 是否
20 以密集模式或稀疏模式编码的标志, 将结合图 4 所述的子树映射程序 418 (图 5) 中存储的密集模式的子树入口描述符 304 (图 4) 修改为稀疏模式。类型字段 1406 提供指示符。

类型字段 1406 的状态表示子树入口 404 是按密集模式还是按稀疏模式配置。如果子树入口 404 以稀疏模式配置, 那么稀疏子树描述符选择
25 字段 1408 中存储的值和子树索引 312 用于选择稀疏子树描述符 1400。稀疏子树描述符选择 1408 将结合图 16 在后面作详细描述。

图 17 说明了用于提供块位移 714 来选择稀疏集中子树的节点的映射程序入口 504 (图 6B) 的图 8 所示的位移逻辑 700 中稀疏模式逻辑 1502。稀疏模式逻辑 1502 根据子树入口 404 中稀疏子树描述符 1400 存储的节点描述符 1402 来提供块位移 714。位移逻辑 700 还包括密集模式逻辑
30 1500。密集模式逻辑 1500 包括节点选择 706 和用来提供密集集中子树中

用于路由的块位移 714 的“1”计数逻辑 708。密集模式逻辑 1500 已经结合图 8 描述过了。

如果类型字段 1406 的状态表示子树入口 404 以稀疏模式配置，那么来自子树入口 404 的子树数据 412 转发到稀疏模式逻辑 1502。结合图 18
5 描述稀疏模式子树逻辑 1502。

图 18 说明了图 17 中位移逻辑 700 所示的稀疏模式逻辑 1502。稀疏模式逻辑 1502 包括子树选择逻辑 1600，多路复用器 1602，内容可编址存储器 (“CAM”) 1606 以及转换逻辑 1604。所选子树入口 404 中的数据字段 406 中存储的稀疏子树描述符 1400^1-1400^n 通过子树数据 412 转发到
10 位移逻辑 700。位移逻辑 700 将稀疏子树描述符 1400^1-1400^n 转发到稀疏模式逻辑 1502 中的多路复用器 1602。由子树选择逻辑 1600 生成的选择 1614 来选择子树数据 412 中的一个稀疏子树描述符 1400^1 。

子树选择逻辑 1600 根据从前一个映射程序层选择的映射程序入口转发的稀疏子树描述符选择 1408 的状态和所选子树入口 404 中的指针字段
15 408 中存储的模式 1404 生成选择 1614，以便选择稀疏子树描述符 1400^1 。表 3 说明了所选稀疏子树描述符 1400^1 和从多路复用器 1602 通过所选择的稀疏子树描述符 1610 从对应模式 4 的子树入口 404 的多路复用器 1602 转发的相应的子树数据位 412。参见表 2 中模式 4 的一行，在模式 4 中，子树入口 404 可以存储 4 个稀疏子树描述符。这 4 个稀疏子树描述符中的
20 的每一个可以以子树入口 404 的模式 4 存储。这 4 个稀疏子树描述符 1400 中的每一个有 63 位并且可以存储 7 到 5 个节点描述符 1402。因此，这 4 个稀疏子树描述符 1400 中的每一个从 63 位的边界开始。第一个稀疏子树描述符 1400^1 存储在数据字段 406 的位 62: 0，第二个稀疏子树描述符 1400^2 存储在数据字段 406 的位 125: 63，第三个稀疏子树描述符 1400^3
25 存储在数据字段 406 的位 188: 126 以及第四个稀疏子树描述符 1400^4 存储在数据字段 406 的位 251: 189。稀疏子树描述符选择 1408 选择数据字段 406 中的相应位。例如，看表 3，如果稀疏子树描述符选择 1408 是“0001”，那么选择第二个稀疏子树描述符 1400^2 并且通过多路复用器 1602 将 256 位子树数据 412 的位 125:63 从所选择的稀疏子树描述符 1610
30 转发到转换逻辑 1604。

稀疏子树描述符选择	子树数据	稀疏子树描述符
0000	Bits 62:0	1
0001	Bits 125:63	2
0010	Bits 188:126	3
0011	Bits 251:189	4

表 3

子树存储器 400 中的每个子树入口 404 可以配置成稀疏模式或密集
5 模式。以稀疏模式配置的每个子树入口 404 可以配置为能够通过模式 1404
存储每个稀疏子树描述符 1400 的不同数量的节点描述符 1402。以稀疏
模式配置的子树入口 404 中所有稀疏子树描述符 1400 存储每个稀疏子树
描述符 1400 相同数量的节点描述符 1402。

可以对节点描述符 1402 编码以表示子树中的多个节点。节点描述符
10 1402 表示的多个 8 位节点通过屏蔽 8 位中的一些位来识别。为取代利用
每个节点描述符 1402 存储屏蔽位，使用 9 位节点描述符 1402 对节点描
述符 1402 表示的 8 位宽的节点进行完全编码。使用运行比特长度编码来
以 9 位宽的节点描述符 1402 对 8 位宽的节点编码。运行比特长度编码可
以识别 8 位节点中的哪些位被屏蔽了。

15 转换逻辑 1604 将所选稀疏子树描述符 1400 中存储的 9 位宽的节点
描述符 1402¹—1402ⁿ 转换为包括置为“X”位（任意值）的 8 位 CAM 值 1612
并且将 8 位 CAM 值 1612 装载到 CAM 1606。表 4 显示了通过转换逻辑 1604
将 9 位节点描述符 1402 转换为 8 位 CAM 值 1612 的例子。

8 位值	9 位代码
101100XX	101100100
100XXXXX	100100000
001111XX	001111100
0011XXXX	001110000

表 4

9 位代码列说明了节点描述符 1402 中存储的值。看表 4 的第一行，
 5 节点描述符 1402 中存储的 9 位代码为“101100100”并且对应的 8 位值
 “101100XX”存储在 CAM 1606 中。转换逻辑 1604 通过在 9 位代码中从
 右到左搜索第一个为“1”的位来转换 9 位代码。在 9 位代码从右到左的
 位中，前两位为“0”，第三位为“1”。由于在第一个“1”的右边有两个“0”，
 因此转换逻辑 1604 将“100”转换为两个任意位（“XX”）。第一个“1”
 10 被忽略，余下的位直接复制到 8 位值的后几位。

在表 4 中的第二行，节点描述符 1402 存储的 9 位代码为
 “100100000”。转换逻辑 1604 通过在 9 位代码中从右到左搜索第一个
 为“1”的位来转换 9 位代码。第 5 位存储“1”。9 位代码被转换为其 5
 位最低有效位（“LSBS”）设置为“任意值”（“X”）的 8 位值。通过
 15 使用 9 位运行比特长度编码存储节点描述符 1402，使每个节点描述符 1402
 所需的位数最少，因此增加了可以存储在查询表 200 中的节点描述符 1402
 的数量。

在 9 位节点描述符 1402 转换为 8 位值之后，转换逻辑 1604 将 8 位
 值载入 CAM 1606。8 位值以与所选稀疏子树描述符 1400 中存储的节点描
 20 述符 1402 相同的顺序载入 CAM 1606，即，从最短到最长的匹配。CAM 1606
 提供用来存储每个稀疏子树描述符 1400 的最大编号节点描述符 1402 的
 存储空间。因此，CAM 1606 为 8 位宽 16 个入口深，以便能提供 15 个入
 口来存储模式 5 的稀疏子树描述符 1400 的节点描述符 1402 的最大编号
 以及缺省的映射程序地址。CAM 1606 具有三进制功能和内置的多匹配分
 25 解器。相对于提供真正的内容地址存储器，小容量的 CAM 1606 可以利用

逻辑门来实现，即，CAM 1606 可以通过仿真 CAM 的硬件电路来实现。

稀疏子树描述符 1400 中存储的节点描述符 1402 的数量确定了存储稀疏子树描述符 1400 的子树入口 404。在特定模式范围内存储节点描述符 1402 的稀疏子树描述符 1400 存储在相同的子树入口 404 中。为每个子树的缺省路由计算缺省映射程序地址。缺省的 8 位值永久地存储在 CAM 1606 的第一单元以计算缺省的映射程序地址。

在所选稀疏子树 1400 对应的 8 位值载入 CAM 1606 之后，利用关键字 210b 的下一部分搜索 CAM 1606。选定与关键字 210b 的下一部分中最大位数匹配的 CAM 1606 中的入口。从搜索 CAM 得到的匹配地址作为块位移 714 转发。块位移 714 用来确定与子树映射程序 418（图 5）中所存储路由对应的映射程序入口的映射程序地址 416。

图 19A—D 说明了对稀疏集中子树 1700 中一个节点的块位移 714 的选择。图 17A 用图形表示了稀疏集中子树 1700 中的路由。子树 1700 中的节点对应三个路由 r0, r1 和 r2 之一，r0 是子树 1700 的缺省路由。以稀疏子树描述符 1400 中的节点描述符 1402¹ 和 1402² 编码 r1, r2 两个路由。缺省路由 r0 的值永久保存在 CAM 1606 中的第一入口 1702 中。参见表 2，具有两个节点描述符 1402 的稀疏子树描述符 1400 存储在模式字段 1404 为“1”的子树入口 404 中。

在子树 1700 中，r2 对应所有与 10xxxxxx 匹配的节点，r1 对应所有与 010xxxxx 匹配的节点。为了最小化用来描述稀疏子树描述符 1400 中每个路由的每个节点描述符 1402¹, 1402² 所需的位数，使用运行比特长度编码对节点描述符 1402¹, 1402² 进行编码。用于编码的方法比用于对节点进行完全编码所用位数多 1 位。在第一个“X”（“任意值”）的位置插入“1”并且其余 X 位编码为“0”。这样，路由 10xxxxxx 被翻译为 10100000，路由 010xxxxx 被翻译为 010100000。

图 19B 说明了节点描述符 1402¹ 和 1402² 在稀疏子树描述符 1400 中的存储。因为有两个节点描述符 1402¹ 和 1402² 存储在稀疏子树描述符 1400 中，所以节点描述符 1402¹ 和 1402² 存储在模式字段 1404 设置为“1”的子树入口 404 中。因为 r1 需要与前三位匹配，而 r2 需要与前两位匹配，所以子树的最长匹配为 r1。节点 1402¹ 和 1402² 按从最短到最长的

匹配顺序存储在稀疏子树描述符 1400 中，其中 r2 的节点描述符 1402¹ 存储在第一位，r1 的节点描述符 1402² 存储在下一位。

图 19C 说明了节点描述符 1402² 向 8 位屏蔽值 1706 的转换。在从左到右的节点描述符位 1708¹—1708⁹ 中，第一个“1”存储在位 1708⁶，它
5 标志着 8 位被屏蔽值 1706 的屏蔽位的结束。执行下面的位转换来实现节点描述符 1402² 向 8 位被屏蔽值 1706 的转换。存储在节点描述符 1708¹ 位的“0”转换为“X”并且存储在 8 位被屏蔽值 1710¹ 位。存储在节点描述符 1708² 位的“0”转换为“X”并且存储在 8 位被屏蔽值 1710² 位。存储在节点描述符 1708³ 位的“0”转换为“X”并且存储在 8 位被屏蔽值 1710³ 位。存储在节点描述符 1708⁴ 位的“0”转换为“X”并且存储在 8 位被屏蔽值 1710⁴ 位。存储在节点描述符 1708⁵ 位的“0”转换为“X”并且存储在 8 位被屏蔽值 1710⁵ 位。存储在节点描述符 1708⁶ 位的“1”被忽略。存储在节点描述符 1708⁷ 位的“0”存储在 8 位被屏蔽值 1710⁶ 位。存储在节点描述符 1708⁸ 位的“1”存储在 8 位被屏蔽值 1710⁷ 位。
10 存储在节点描述符 1708⁹ 位的“0”存储在 8 位被屏蔽值 1710⁸ 位。

图 19D 说明了存储 CAM 1606 中的节点描述符 1402¹ 和 1402² 以及所选择的稀疏子树描述符 1400 的子树映射程序 418（图 5）中存储的对应的映射程序入口 504¹—504³。将所选择的子树描述符 1400 中存储的 9 位节点描述符 1402¹ 和 1402² 转换到转换逻辑 1604（图 18）中，并且载入
20 CAM 1606。CAM 1606 中第一个入口 1702 是图 19A 中子树 1700 所示的 r0 的缺省入口。第二个入口 1704 从所选择的稀疏子树描述符 1400 中存储的第一节点描述符 1402¹ 转换。第二个入口 1704 是为 r2 转换的最短匹配。把所选择的子树描述符 1400 中存储的第二节点描述符 1402² 从 010100000 转换为 010XXXXX 并且存储在 CAM 1606 的第三入口 1706 中。

对 CAM 1606 的搜索产生块位移 714（图 18）。块位移 714 用来确定子树映射程序 418（图 5）中存储的映射程序入口 504¹—504³ 的映射程序地址 416。利用存储最长匹配的入口 1702，1704，1706 的关键字 210b 的第二部分来搜索 CAM 1606。根据所选择的子树入口 404 中的指针字段 408 中存储的块基地址 600¹，600¹ 之一将 CAM 1606 提供的块位移 714 与
30 子树基地址结合。

图 20 是说明图 8 所示指针逻辑 702 中稀疏模式基选择逻辑 1800 的方框图。指针逻辑 702 选择用于计算子树映射程序 418 (图 5) 中映射程序入口 504 (图 6B) 的映射程序地址 416 的基地址 716。指针逻辑 702 包括密集模式基选择逻辑 710 和稀疏模式基选择逻辑 1800, 根据从上一层映射程序层转发的子树入口描述符 304 (图 4) 中存储的类型 1406 的状态来选择其一。如前所述, 类型 1406 的状态表示是否将子树配置为密集模式。

如果子树入口 404 存储了多个稀疏子树描述符 1400, 稀疏模式基选择逻辑 1800 计算稀疏子树描述符 1400 的基地址 716。稀疏模式基选择逻辑 1800 利用模式字段 1404 中存储的模式值 1608 和子树入口 404 中块基地址域 600^1 , 600^2 中存储的子树指针 414 以及上一层映射程序层转发的子树入口描述符 304 (图 4) 中存储的稀疏子树描述符选择 1408 来计算基地址 716。基地址 716 的计算如下:

$$\text{基地址 (稀疏子树描述符)} = \text{块基地址} + \text{基位移}$$

其中, $\text{基位移} = ((1 + \text{节点/子树}) \times \text{稀疏子树描述符选择})$

例如, 为查找以稀疏模式 4 配置的子树入口 404 中 2 号子树的起始基地址 716, 首先计算基位移。2 号子树对应的稀疏子树描述符选择 1408 为“1”并且节点/子树数为 7 (见表 2)。基位移为 $8((1+7) \times 1)$ 。每个块基地址 600^1 , 600^2 是分配给子树入口 404 的 16 个映射程序地址块的基地址。2 号子树对应的基位移是小于 16 的 8, 因此 2 号子树的块基地址是块基地址 600^1 并且稀疏子树描述符的基地址 716 是块基地址 $600^1 + 8$ 。表 5 说明了以模式 4 配置的子树入口 404 中四个子树的每个子树基地址。

子树基地址	子树
块基地址 $1+0$	1
块基地址 $2+8$	2
块基地址 $2+0$	3
块基地址 $2+8$	4

表 5

图 21 说明了存储在子树存储器 400 中的密集子树描述符和稀疏子树描述符。结合图 15 来描述图 21。子树 B (图 21) 的密集子树描述符存储在子树入口 404¹ 的数据字段 406¹。子树 A (图 21) 的稀疏子树描述符 1400¹ 存储在子树入口 404² 的数据字段 406²。密集子树描述符存储结合图 6B 描述过的子树 B 底层中每个节点的节点位。稀疏模式描述符 1400¹ 包括对应结合图 19B 描述过的路由 r4 和 r5 的节点描述符 1402¹ 和 1402²。子树索引 312 选择子树入口 404¹, 404²。

s0 (图 15) 的映射程序 106a 中映射程序入口 504 (图 6B) 中子树入口描述符 304 (图 4) 存储的子树索引 312 选择子树入口 404¹。s1 (图 15) 的映射程序 106a 中映射程序入口 504 (图 6B) 中子树入口描述符 304 (图 4) 存储的子树索引 312 选择子树入口 404²。因此, 子树存储器 400 可以存储稀疏子树和密集子树对应的子树入口 404¹, 404²。

图 22 是说明用来向存储稀疏集中子树和密集集中子树中的节点的路由的子树映射程序 418 (图 5) 中的映射程序入口 504 (图 6B) 提供映射程序地址 416 (图 5) 的方法的流程图。任何子树入口 404 都可以存储多个稀疏子树描述符或一个密集子树描述符。根据路由在二叉树中的如何分布, 可以对稀疏子树描述符和密集子树描述符进行任意组合。灵活地混合和匹配在子树存储器 400 中子树入口 404 中的稀疏模式和密集子树描述符可以更好地利用子树存储器 400。

在步骤 1900 中, 从在上一映射程序层中选择的子树入口描述符 304 (图 4) 中存储的类型 1406 的状态来确定所选择的子树入口 404 的配置。如果子树入口 404 的类型配置为稀疏模式, 处理过程继续到步骤 1902。否则, 处理过程继续到步骤 1914。

在步骤 1902 中, 子树入口 404 的配置是稀疏模式。以稀疏模式配置的子树入口 404 存储多个稀疏子树描述符 1400。根据模式字段 1404 的状态确定子树入口 404 中存储的稀疏子树描述符 1400 的数量。位移逻辑 700 中的稀疏模式逻辑 1502 根据从上一映射程序层转发的子树入口描述符 304 (图 4) 中存储的稀疏子树描述符选择 1408 和前面结合图 14 描述过的模式字段 1404 的内容来从子树入口 404 中选择稀疏子树描述符

1400。处理过程继续到步骤 1904。

在步骤 1904，所选择的稀疏子树描述符 1400 中的节点描述符 1402 中存储的 9 位编码值被转换为 8 位值并且按从最短到最长匹配的顺序存储在 CAM 1606 中。处理过程继续到步骤 1906。

- 5 在步骤 1906 中，利用关键字 210b 的下一部分在 CAM 1606 中搜索存储最长匹配的 CAM 入口。处理过程继续到步骤 1908。

在步骤 1908 中，把存储关键字 210b 下一部分的最长匹配的 CAM 1606 中的位置的地址作为块位移 714 转发。块位移 714 用来计算子树映射程序 418 (图 5) 中映射程序入口 504 (图 6B) 的映射程序地址 416 (图 5B)。

- 10 处理过程继续到步骤 1910。

在步骤 1910 中，根据从上一映射程序层转发的子树入口描述符 304 (图 4) 中存储的稀疏子树描述符选择 1408 和所选择的子树入口 404 中存储的模式字段 1404 的内容来计算所选稀疏子树描述符 1400 对应的基地址 716。处理过程继续到步骤 1912。

- 15 在步骤 1912 中，通过在加法器逻辑 704 (图 8) 中将块位移 714 和基地址 716 相加来计算映射程序地址 416。通过子树映射程序 418 (图 5) 中的映射程序地址 416 识别的映射程序入口 504 存储路由入口 302 (图 4) 或子树入口描述符 304 (图 4)。如果映射程序入口 504 (图 6B) 存储路由入口 302 (图 4)，那么搜索结束。如果映射程序入口 504 (图 6B) 存储子树入口描述符 304 (图 4)，那么继续在下一映射程序层搜索对应关键字 210b 的值。
- 20

- 在步骤 1914 中，子树入口 404 以密集模式配置并且在数据字段 406 中存储一个密集子树描述符。通过统计前面结合图 6B 描述过的子树入口 404 中数据字段 406 中存储的密集子树描述符中“1”的数量来计算块位移 714。处理过程继续到步骤 1916。
- 25

在步骤 1916 中，子树入口 404 在子树入口 404 中的指针字段 408 存储 16 个块基地址 600。前面结合图 8 描述过的指针逻辑 702 中的密集模式基选择逻辑 710 选择块基指针 600 中的一个。处理过程继续到步骤 1912。

递增更新

图 23 说明了在查询表 200 中添加新路由的二叉树表示。二叉树说明了查询表 200 中存储的对应映射程序层 1 2000, 映射程序层 2 2002 和映射程序层 3 2004 的路由。映射程序层 2 2002 存储子树 A 和 B 的路由。5 映射程序层 3 2004 存储子树 A_1 , A_2 , B_1 和 B_2 的路由。s5 表示子树映射程序 418 (图 5) 中存储的子树入口描述符 304 (图 4)。s5 对应的子树入口描述符 304 (图 4) 存储指向子树 B_2 的指针, 子树 B_2 允许继续在映射程序层 3 2004 搜索关键字 210 的最长匹配路由。

10 因为子树 B_2 2006 只有两个路由, r6 和 h1, 所以它是稀疏子树。因此, 节点 r6 和 h1 对应的节点描述符 1402 (图 16A) 存储在已经结合图 14A 描述过的稀疏子树描述符 1400 中。因为在稀疏子树描述符 1400 中存储了两个节点描述符 1402, 所以子树 B_2 2006 对应的稀疏子树描述符 1400 存储在子树存储器 400 中模式字段 1404 设置为“1”的子树入口 404 中。

15 子树 B_2' 2008 所示的新节点 h2 添加到查询表 200 中。因为子树 B_2 2006 的新增路由使稀疏子树描述符 1400 中存储的节点描述符 1402 的数量从 2 增加到 3, 所以新路由 h1 不能直接添加到查询表的子树 B_2 2006 中。向稀疏子树描述符 1400 增加节点描述符 1402 需要在模式字段 1404 设置为“2”的子树入口 404 中分配新的稀疏子树描述符 1400。因此, 增加20 路由 h1 需要用子树 B_2' 2008 代替子树 B_2 2006。

图 24 说明了处理器存储器 2400 中存储的更新路由。查询表 200 中存储的二叉树的拷贝也存储在与查询表 200 分离的处理器存储器 2400 中。存储子树 B_2 2006 的路由复制到处理器存储器 2400 的子树 B_2' 2008 并且向子树 B_2' 2008 增加新路由 h2。

25 路由更新程序 2402 生成一系列路由更新指令 2404, 以便将子树 B_2' 2008 添加到查询表 200 中并且将路由更新 2404 转发到表更新程序 2406。表更新程序 2406 生成路由更新 2402 对应的表更新 2410 并且转发更新周期 2412, 以便利用路由更新 2404 来更新查询表 200。更新周期 2412 将路由更新写入子树存储器 400 (图 5) 中相应的存储器位置和子树映射程30 序 418 中 (图 5)。

返回图 23,更新周期 2412 包括用来分配一部分子树映射程序 418(图 5)以存储映射程序入口 504(图 6B)中新子树 B_2' 2008 对应的路由的指令。子树 B_2' 2008 包括路由 h1 和 r6 以及新路由 h2 的映射程序入口 504(图 6B)中存储的路由入口。在子树 B_2' 2008 的路由入口存储在子树映射程序 418(图 5)中的映射程序入口 504(图 6B)后,产生了这些路由对应的节点描述符 1402 并且存储在稀疏子树描述符 1400 中。稀疏子树描述符 1400 存储在子树入口 404。子树入口 404 的模式 1404 与稀疏子树描述符 1400 中存储的节点描述符 1402 的数量有关。

在子树 B_2' 2008 的稀疏子树描述符 1400 存储在查询表 200 的子树存储器 400 的子树入口 404 中之后, s5 代表的子树入口描述符 304(图 4)由指向子树 B_2 2006 改为指向子树 B_2' 2008。当把子树 B_2' 2008 添加到查询表时,可以通过 s5 访问子树 B_2 2006 中存储的路由 r6 和 h1。在子树 B_2' 2008 存储到查询表中后, s5 改为指向子树 B_2' 2008 并且可以访问路由 h1 和 r6 和新路由 h2。因此,当把新路由 h2 添加到查询表 200 中时,可以在子树 B_2 2006 中继续搜索对应于路由 h1 和 r6 的路由索引。

图 25 说明了查询表 200 中子树映射程序 418b 中的映射程序入口 $504c^4$ 中存储的图 23 所示的新路由 h2。结合图 24 所示的二叉树表示来描述图 25。

映射程序层 2 2002 中的子树 B 有 3 个路由;即 r3, s4 和 s5。因为子树 B 的路由少于 16 个,所以它是稀疏子树。子树 B r3, s4 和 s5 的节点描述符 $1402a^1-1402a^3$ 存储在子树存储器 400a 中子树入口 404a 中的稀疏子树描述符 $1400a^1$ 中。子树映射程序 418a 中存储子树 B 中每个路由对应的映射程序入口 $504a^2-504a^4$ 。子树映射程序 418a 中的映射程序入口 $504a^1$ 存储子树 B 的缺省路由。每个映射程序入口 $504a^2-504a^4$ 存储该节点的路由入口 302(图 4)或子树入口描述符 304(图 4)。 $504a^3$ 中存储路由 s4 的子树入口描述符 304(图 4), $504a^4$ 中存储 s5 的子树入口描述符 304(图 4)。映射程序入口 $504a^4$ 中存储的 s5 为子树存储器 400b 提供子树索引 312b,以便开始下一层搜索;即对映射程序层 3 2004 的搜索。

子树 B_2 有两个路由,即 h1 和 r6,所以它也是稀疏子树。节点描述

符 $1402b^1-1402b^2$ 存储在子树存储器 400b 中的子树入口 404b 中的稀疏子树描述符 $1400b^1$ 中。子树 B_2 的每个路由存储在映射程序入口 $504b^2-504b^3$ 中并且子树 B_2 的缺省路由存储在映射程序入口 $504b^1$ 中。

为在子树 B_2 2006 中搜索路由 h1, 将保存了用来存储路由 s5 的节点描述符 1402 的稀疏子树描述符 $1400a$ 的子树入口 404a 的地址通过子树索引 312a 转发到子树存储器 400a。所选择的子树入口 404a 中存储的数据字段 406 和指针字段 408 通过子树数据 412a 和子树指针 414a 转发到映射程序地址逻辑 402a。映射程序地址逻辑 402a 生成存储了 s5 的子树入口的映射程序入口 $504a^4$ 的映射程序地址 416a。映射程序地址 416a 取决于子树数据 412a, 子树指针 414a 和关键字 210b 的下一部分。把 s5 的子树入口通过子树索引 312b 转发到子树存储器 400b。

子树存储器 400b 存储子树 B_2 2006 的节点描述符 $1402b^2, 1402b^1$ 。子树 B_2 的稀疏子树描述符 $1400b^1$ 存储在子树入口 404b 中。子树入口 404b 中存储的数据字段 406 和指针字段 408 通过子树数据 412b 和子树指针 414b 转发到映射程序地址逻辑 402b。映射程序地址逻辑 402b 生成存储了 h1 的路由入口的映射程序入口 $504b^3$ 的映射程序地址 416b。映射程序地址 416b 取决于子树数据 412b, 子树指针 414b 和关键字 210c 的下一部分确定。

为向子树 B_2 2006 添加路由 h2, 分配子树映射程序 418b 中一组前面未使用过的映射程序 602c, 以便存储保存了子树 B_2' 2008 的路由 r6, h1 和 h2 的映射程序入口 $504c^2-504c^4$ 。映射程序入口 $504c^1$ 存储子树 B_2' 2008 的缺省入口; 即, 映射程序入口 $504b^1$ 中存储的相同的值。映射程序入口 $504c^2$ 存储路由 r6 的路由入口; 即, 映射程序入口 $504b^2$ 中存储的相同的值。映射程序入口 $504c^3$ 存储路由 h1 的路由入口; 即, 映射程序入口 $504b^3$ 中存储的相同的值。映射程序入口 $504c^4$ 存储路由 h2 的路由入口。当写入映射程序入口 $504c^{1-4}$ 块时, 可以通过子树映射程序 418a 中的 $504a^4$ 中为路由 s5 存储的子树入口来访问映射程序入口 $504b^1-504b^3$ 中存储的路由入口。

在子树映射程序 418b 中已存储了子树 B_2' 2008 的映射程序入口 $504c^{1-4}$ 后, 把稀疏子树描述符 $1400c^1$ 添加到子树存储器 400b。节点描述符

1402c¹⁻³的数量少于16,因此,节点描述符1402c¹⁻³存储在稀疏子树描述符1400c¹中。子树描述符1400¹在子树存储器400b中的位置取决于与稀疏子树描述符1400c¹有关的节点描述符1402c¹⁻³的熟练。通过在子树B₂ 2006中增加一个新路由,要为稀疏子树描述符1400c¹存储的节点描述符1402c¹⁻³的数量从2增加到3。稀疏子树描述符1400c¹存储在每个稀疏子树描述符含有3个节点描述符的子树入口404c中,并且将模式字段1404设置为“2”。如果有可用空间或分配了新模式3的子树入口,那么稀疏子树描述符1400c¹存储在当前模式3的子树入口404c中。子树B₂' 2008中路由的节点描述符存储在模式3的子树入口404c中稀疏子树描述符1400c¹中的节点描述符1402c¹⁻³中。

把稀疏子树描述符1400c¹和节点描述符1402c¹⁻³存储到子树存储器400b后,可以访问子树B₂' 2008。为提供对子树B₂' 2008的访问,修改子树入口504a⁴在索引子树入口404c中索引稀疏子树描述符1400b¹而不是在子树入口404b中索引稀疏子树描述符1400c¹。可以访问映射程序入口504c⁴中存储的路由h2的路由入口和相应映射程序入口504c²和504c³中分别存储的路由r6和h1。

由于不能再访问映射程序入口504b¹⁻³,所以将它们释放并放置在空闲列表(未示出)中等待以后分配。另外,也不能再访问稀疏子树描述符1400b¹。因此,稀疏子树描述符1400b¹被释放并放置在空闲列表(未显示)中等待以后分配。

已经描述了向稀疏子树中增加路由。也可以通过在新分配的子树入口404中存储新的密集子树描述符以及在子树映射程序418中存储对应的映射程序入口,并且将映射程序入口504a⁴中存储的子树入口改为索引新分配的子树入口404来向密集子树中增加路由。

图26是说明在图25所示查询表200中执行增加路由的递增更新步骤的流程图。

在步骤2200中,计算每个子树的路由数量以确定路由更新产生稀疏子树还是密集子树。如果路由更新后为密集子树,处理过程继续到步骤2218。如果路由更新后为稀疏子树,处理过程继续到步骤2202。

在步骤2202中,子树是稀疏的,因此确定为稀疏子树模式。处理过

程继续到步骤 2204。

在步骤 2204 中，搜索子树映射程序 418（图 5）中存储的部分填充的子树入口 404 的列表以确定是否可将新的稀疏子树描述符 $1400c^1$ 存储在前面分配的子树入口 404 中。例如，可将 4 个稀疏子树描述符 $1400c^1$ — $1400c^4$ 存储在模式 4 的子树入口 404 中。如果只存储了 3 个，子树入口 404 被部分填充并且存储在部分填充的子树入口 404 的列表中。如果有可用的部分填充的子树入口 404，处理过程继续到步骤 2208。否则，处理过程继续到步骤 2206。

在步骤 2206 中，分配新的子树入口 $404c$ 以便存储稀疏子树描述符 $1400c^1$ ，并且在为新分配的子树入口 $404c$ 中的稀疏子树描述符 $1400c^1$ 中存储的节点描述符 $1402c^1$ — $1402c^3$ 存储映射程序入口 504（图 6B）的子树映射程序中分配映射程序入口 $504c^1$ — $504c^4$ 。把指向子树映射程序 418（图 5）中分配的映射程序入口 $504c^1$ — $504c^4$ 块的指针存储在新子树入口 $404c$ 的指针字段 408 中。处理过程继续到步骤 2208。

在步骤 2208 中，从子树入口 $404c$ 中指针字段 408 中存储的指针和子树入口 $404c$ 中模式字段 1404 中存储的模式来确定稀疏子树描述符 $1400c^1$ 的子树映射程序中第一个映射程序入口 $504c^1$ 的位置。处理过程继续到步骤 2210。

在步骤 2210 中，将稀疏子树的路由入口存储在子树映射程序 418b 中的映射程序入口 $504c^1$ — $504c^4$ 中。处理过程继续到步骤 2212。

在步骤 2212 中，把存储节点描述符 $1402c^1$ — $1402c^3$ 的稀疏子树描述符 $1400c^1$ 存储在子树入口 $404c$ 中。处理过程继续到步骤 2214。

在步骤 2214 中，映射程序入口 $504a^4$ 中存储的子树入口描述符 304（图 4）被修改为索引子树入口 $404c$ 中存储的新稀疏子树描述符 $1400c^1$ 。现在可以访问映射程序入口 $504c^4$ 中存储的 h2 的路由入口。处理过程继续到步骤 2216。

在步骤 2216 中，不能再访问映射程序入口 $504b^1$ — $504b^3$ 和稀疏子树描述符 1400b。将映射程序入口 $504b^1$ — $504b^3$ 放置在子树映射程序 418b 的映射程序入口 504（图 6B）的空闲列表中并且分配用来存储其它路由。在部分填充的子树入口的列表中更新子树入口 $404b$ 中的第一个可用位

置。处理过程结束。

在步骤 2218 中，从处理器存储器 2400（图 24）中存储的空闲子树入口 404 的列表分配一个新的子树入口 404。分配新的子树入口 404 用来存储新的密集子树描述符。分配子树映射程序 418b 中的映射程序入口组 504（图 6B）用于存储这些路由。指向若干块已分配的映射程序入口 504（图 6B）的指针存储在子树入口 404（图 5）中的指针字段 408（图 7）中。处理过程继续到步骤 2220。

在步骤 2220 中，如前面结合图 6A—B 描述的，将新的密集子树描述符写入新子树入口 404 中的数据字段 406。处理过程继续到步骤 2222。

在步骤 2222 中，把密集子树的路由入口存储在由子树入口 404 中的指针字段 408 中存储的指针所识别的子树映射程序 418（图 5）中的映射程序入口 504（图 6B）中。处理过程继续到步骤 2224。

在步骤 2224 中，修改映射程序入口 504a⁴ 中存储的子树入口描述符 304（图 4）以指向新的子树入口 404c 中存储的新密集子树描述符。现在可以访问映射程序入口 504c⁴ 中存储的 h2 的路由入口。处理过程继续到步骤 2226。

在步骤 2226 中，把由旧子树入口 404 中的指针字段 408 中存储的指针指向的映射程序入口 504（图 6B）返回到处理器存储器 2400（图 24）中存储的映射程序入口的空闲列表。旧的子树入口 404b 添加到处理器存储器 2400（图 24）中存储的空闲子树入口列表中。

已经描述了向查询表添加路由的过程。执行类似的过程以便从查询表删除路由。例如，从子树 B₂' 2008 中删除 h2 504c⁴ 需要存储含有路由 r6 和 h1 的两个节点描述符的新稀疏子树描述符，在模式 2 的子树入口中存储稀疏子树描述符，更新对应的子树映射程序并且习惯映射程序入口 504a⁴ 中存储的子树入口描述符 304（图 4）指向新子树入口 404 中存储的更新的子树描述符。

虽然已经参考优选实施例特别给出和描述了本发明，本领域技术人员可以理解，在不脱离本发明所附权利要求的范围的情况下可以对本发明的形式和细节进行改变。

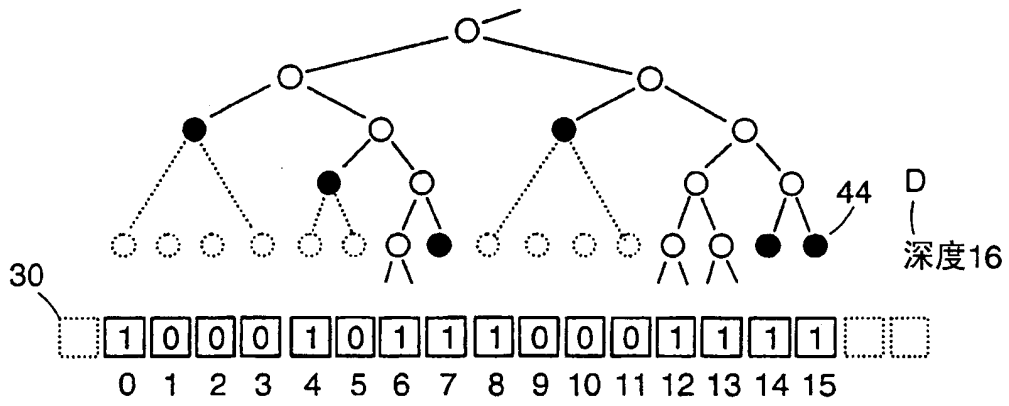


图 1A

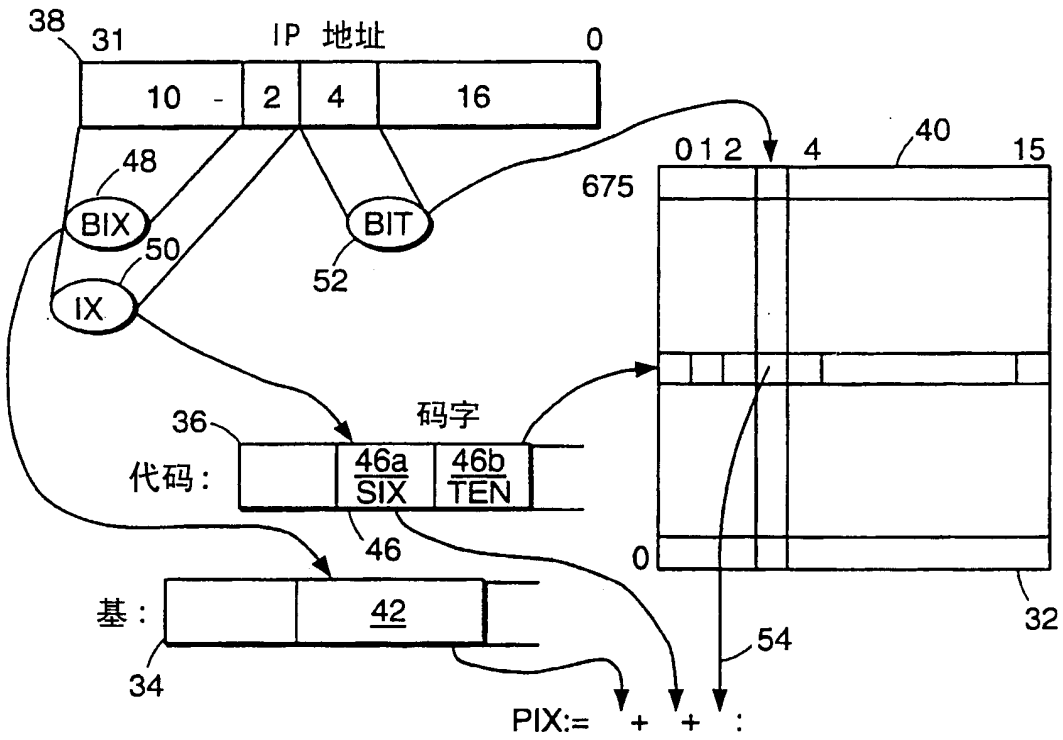


图 1B

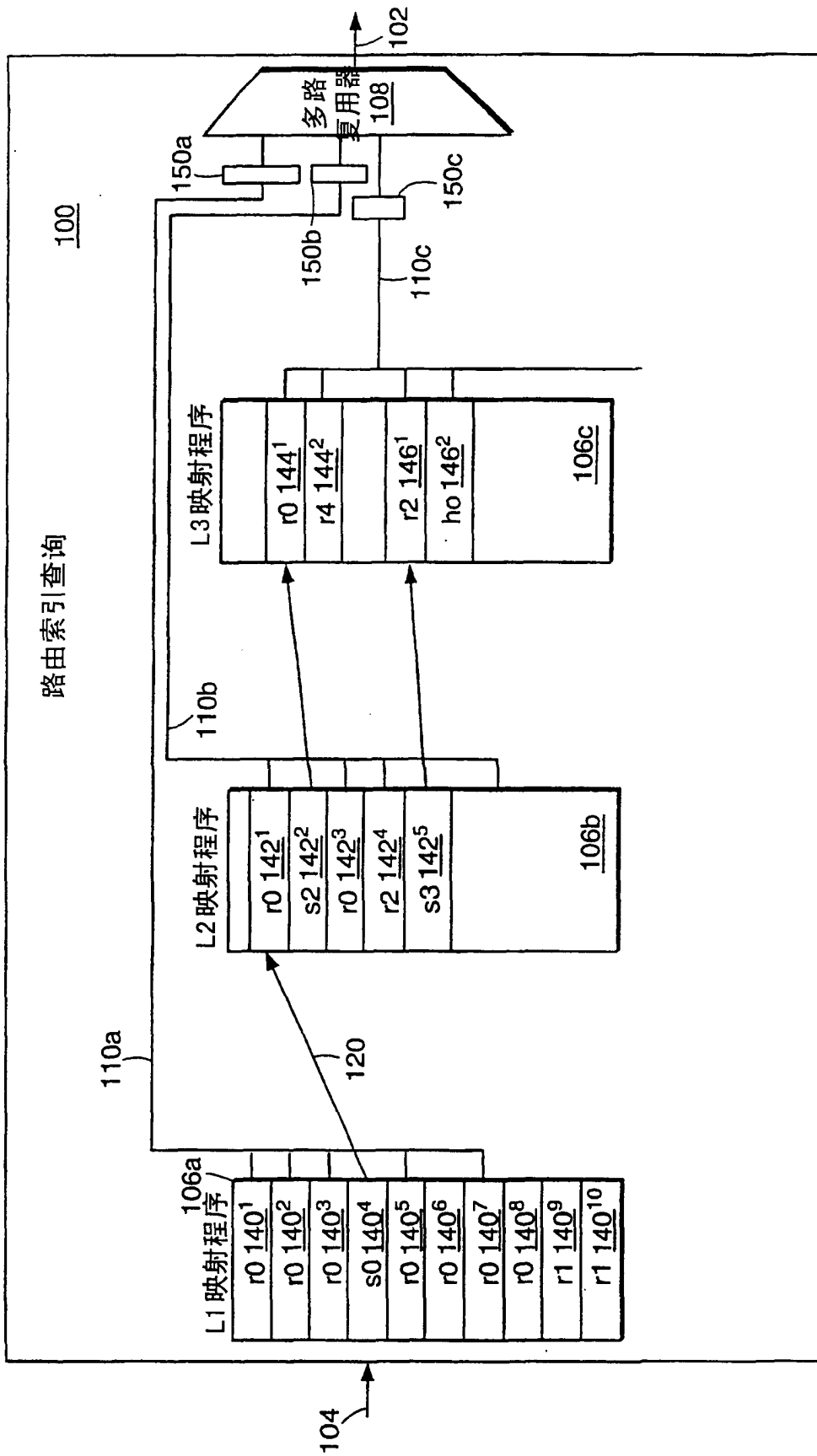


图 2A

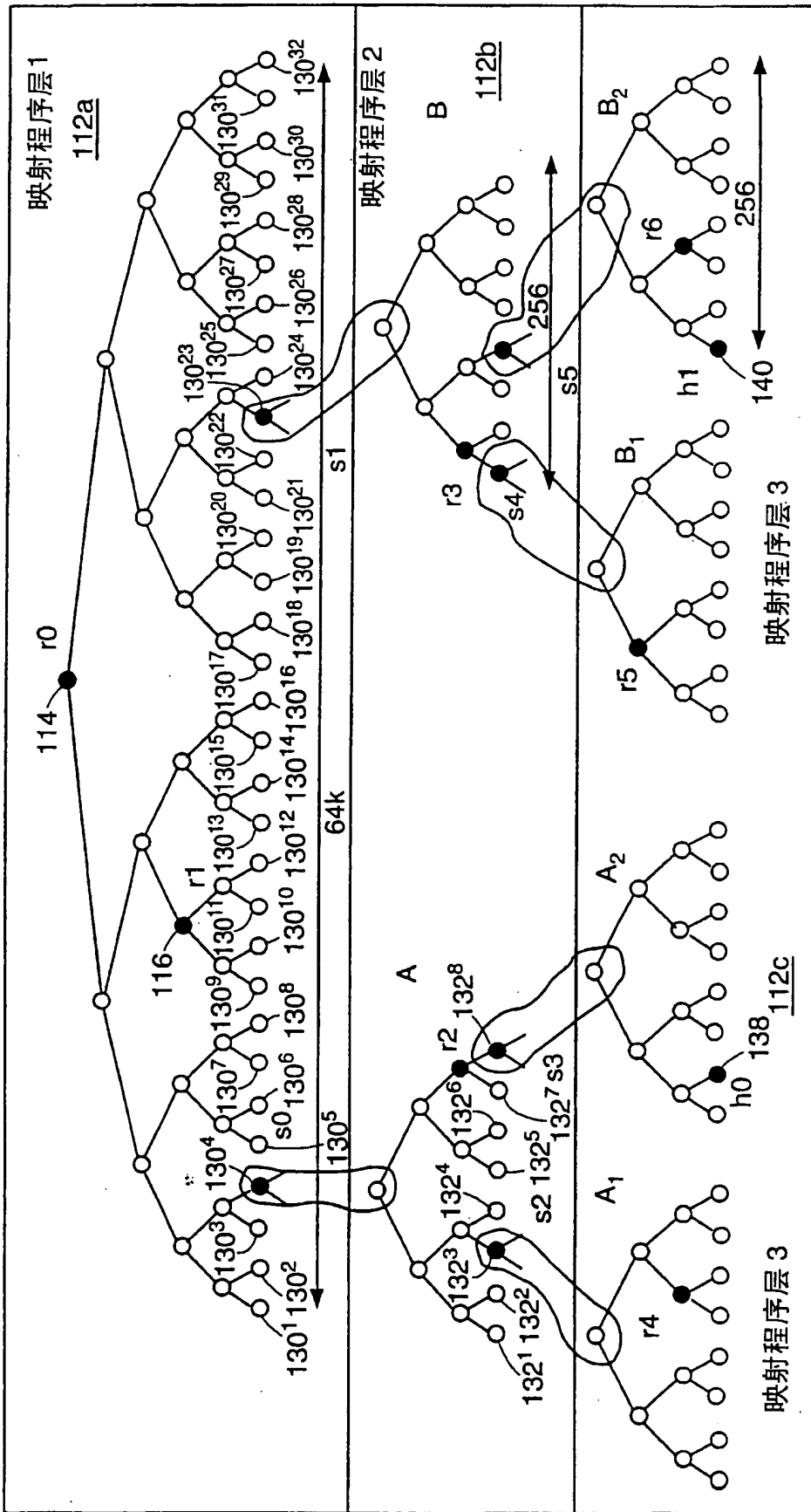
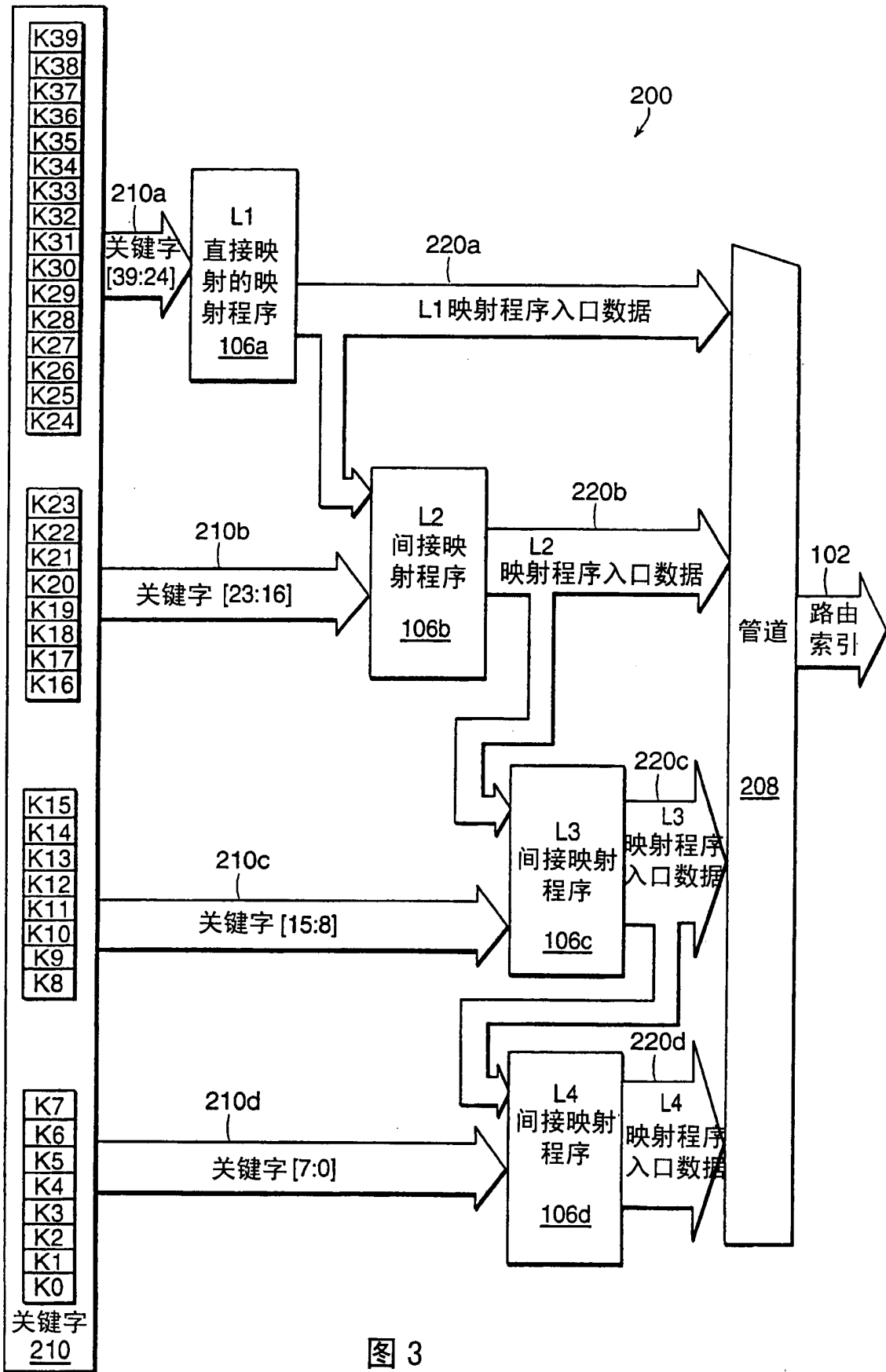


图 2B



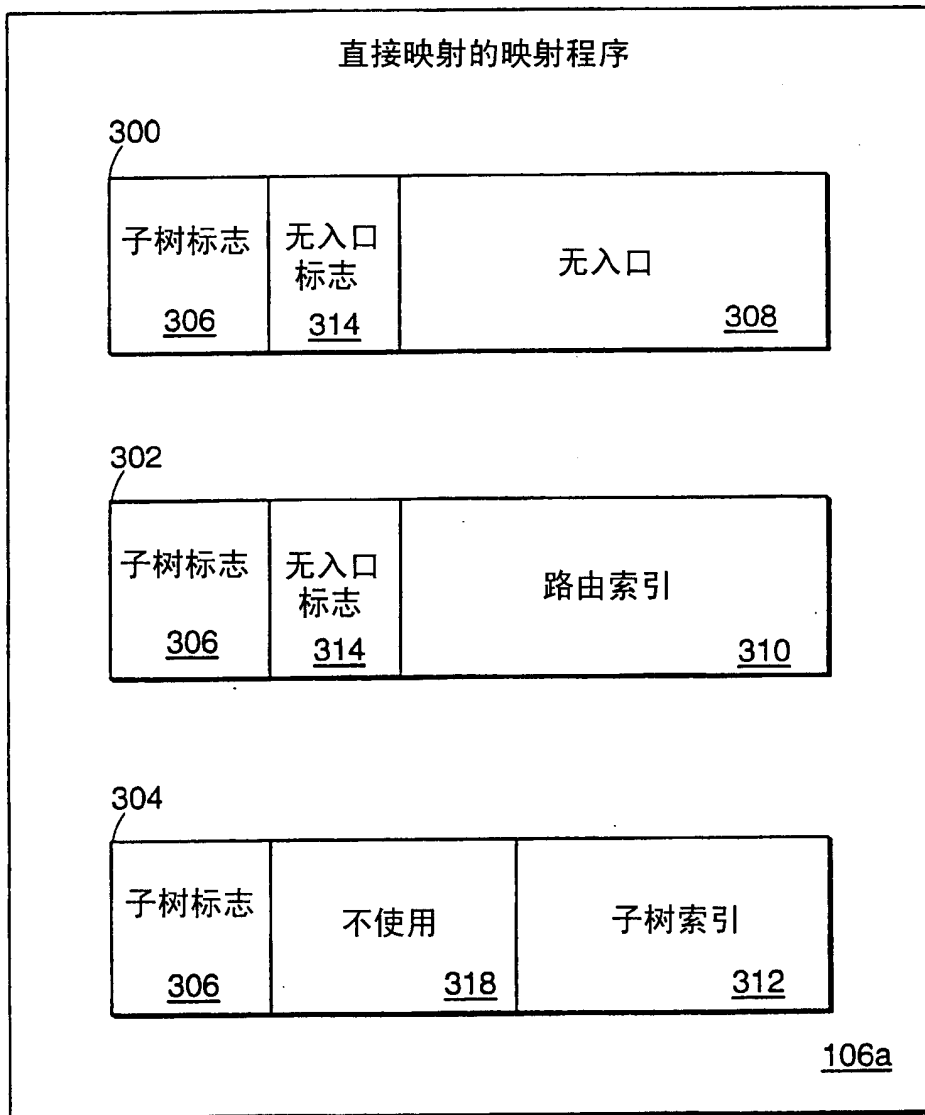


图 4

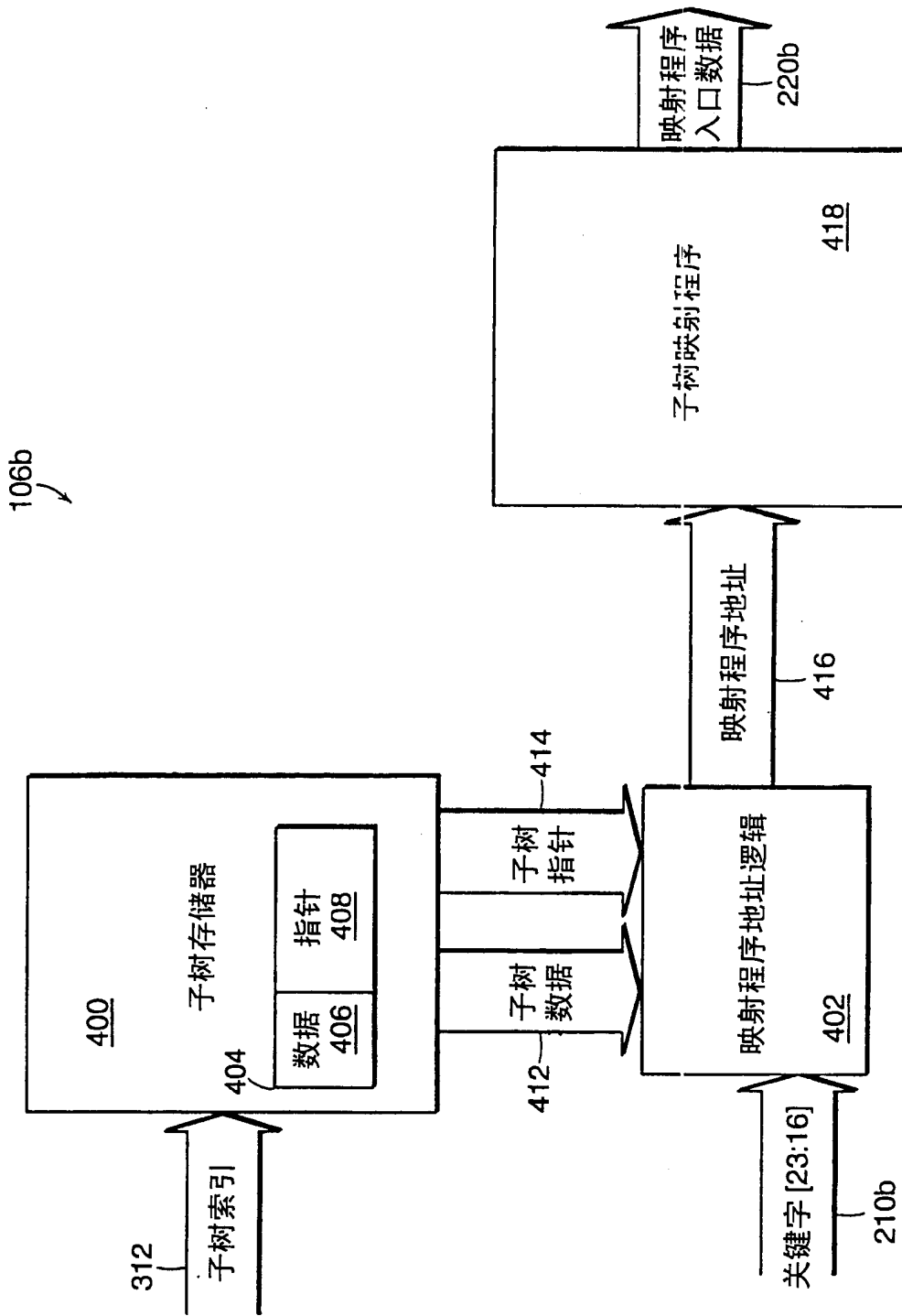


图 5

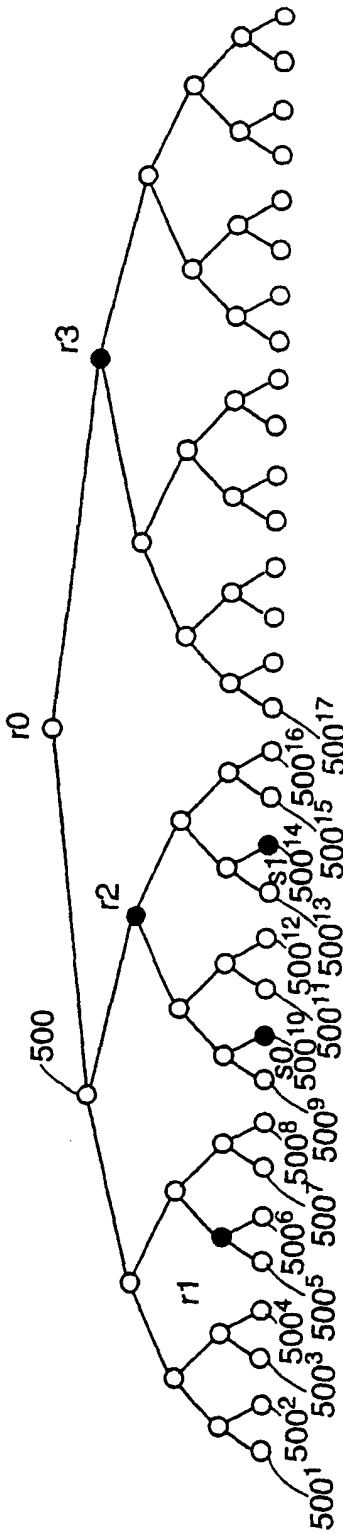


图 6A

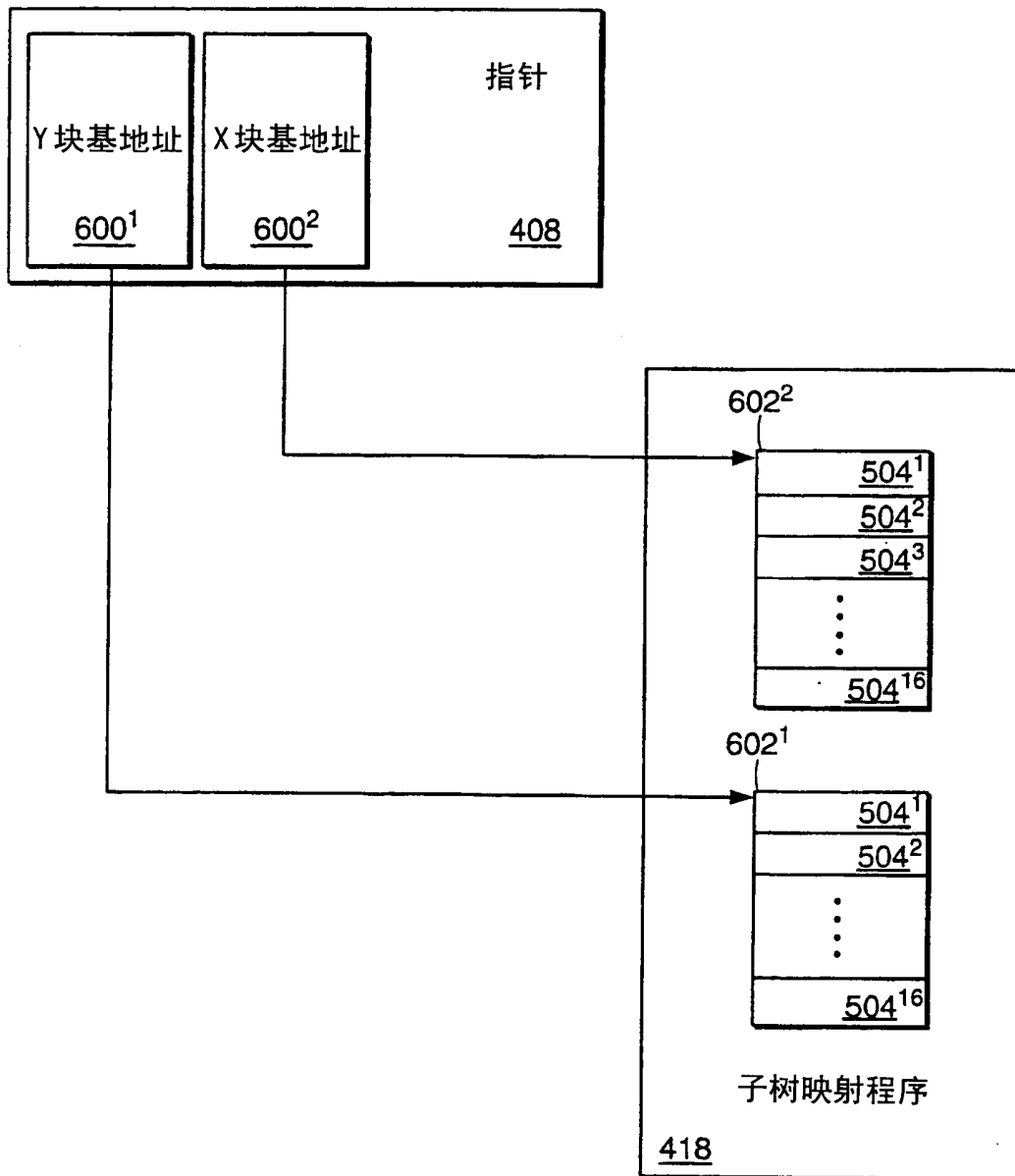


图 7

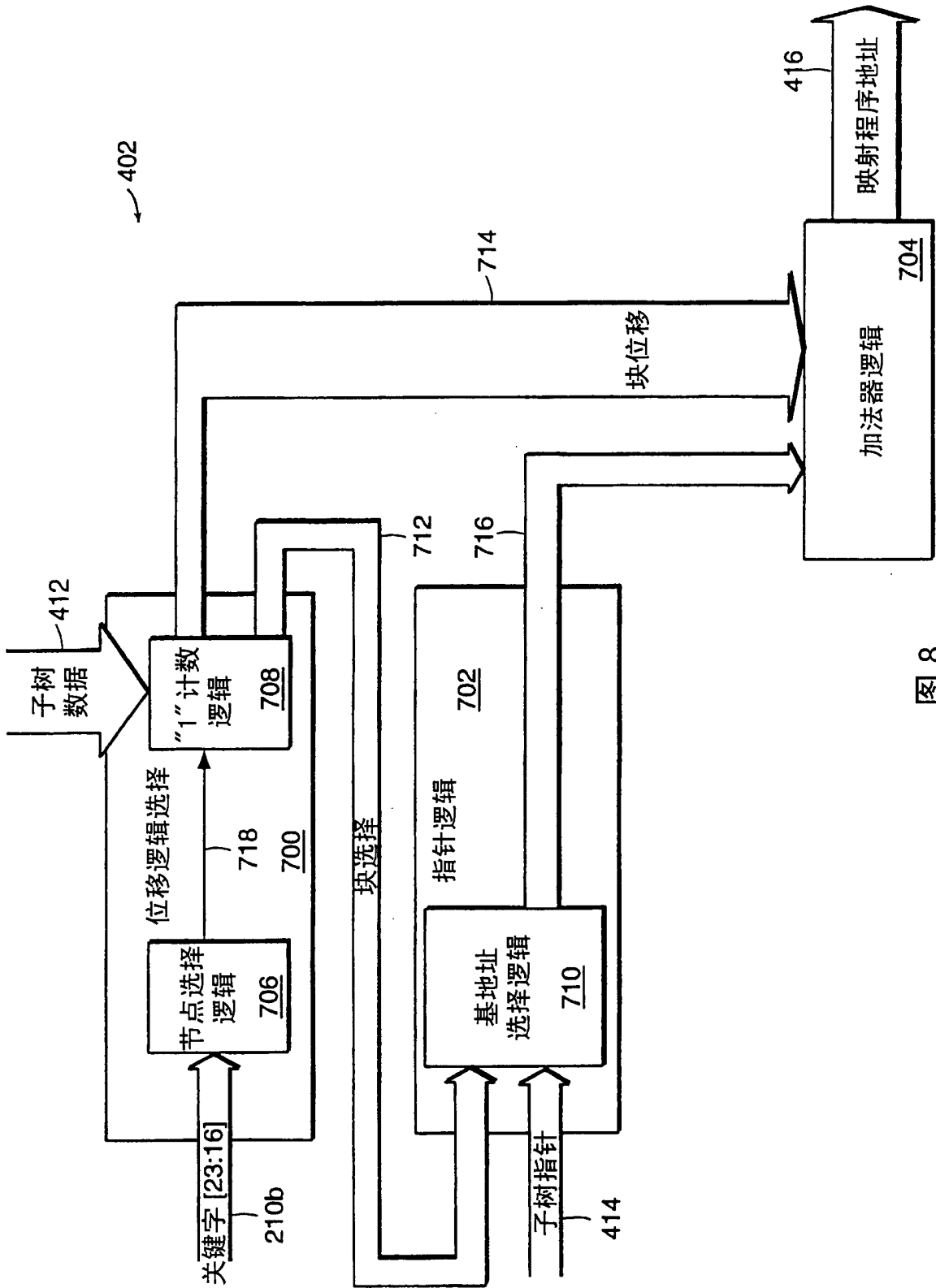


图 8

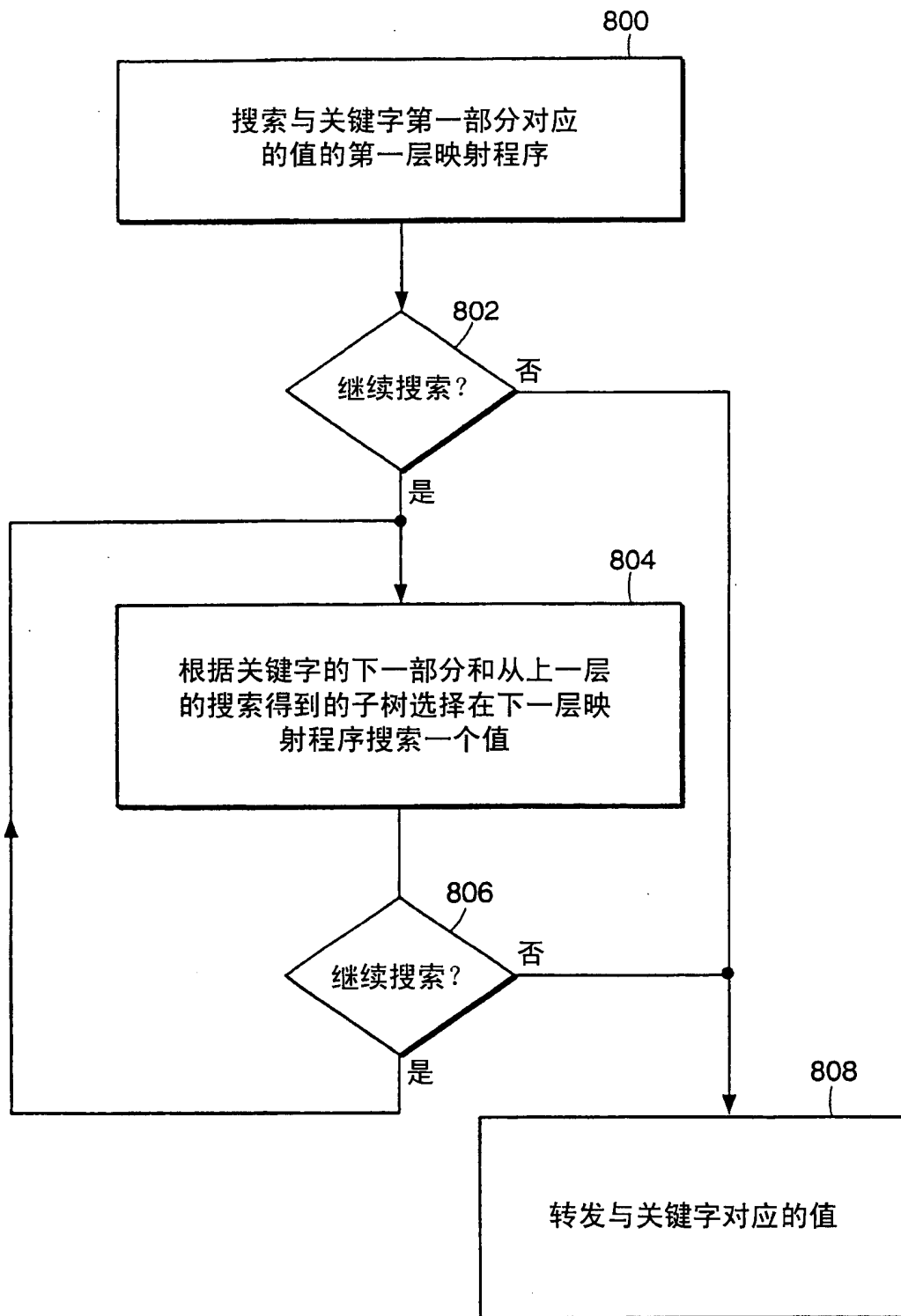


图 9

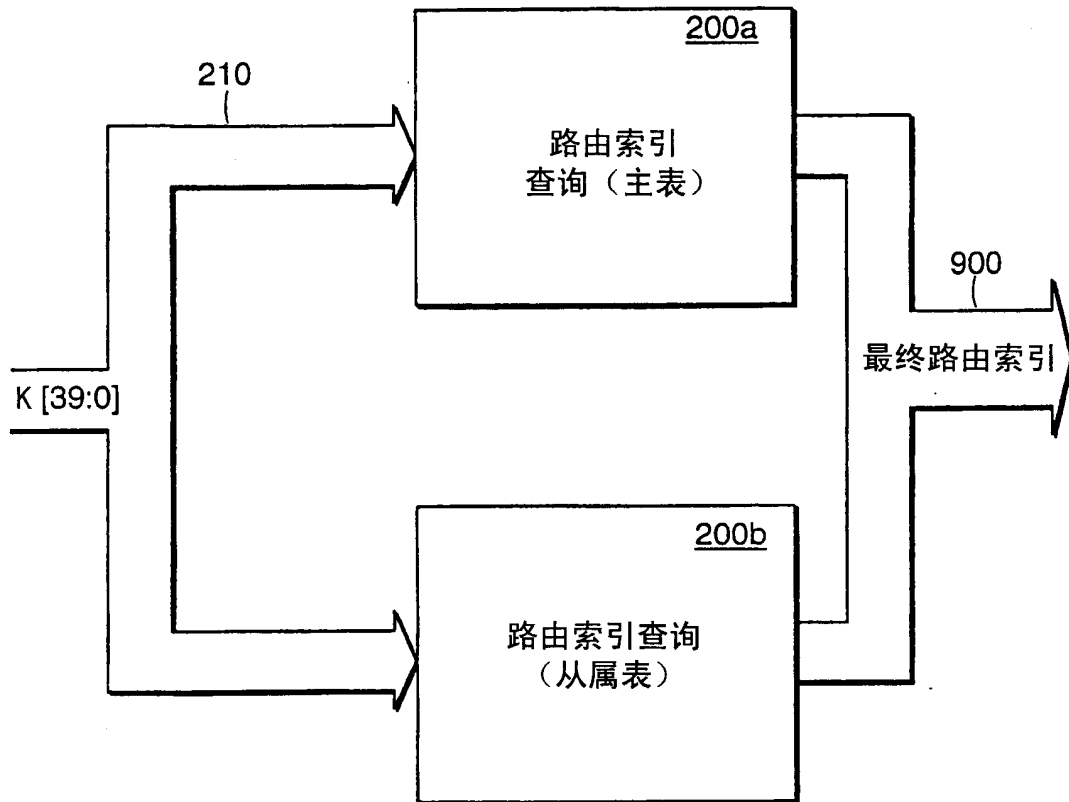


图 10A

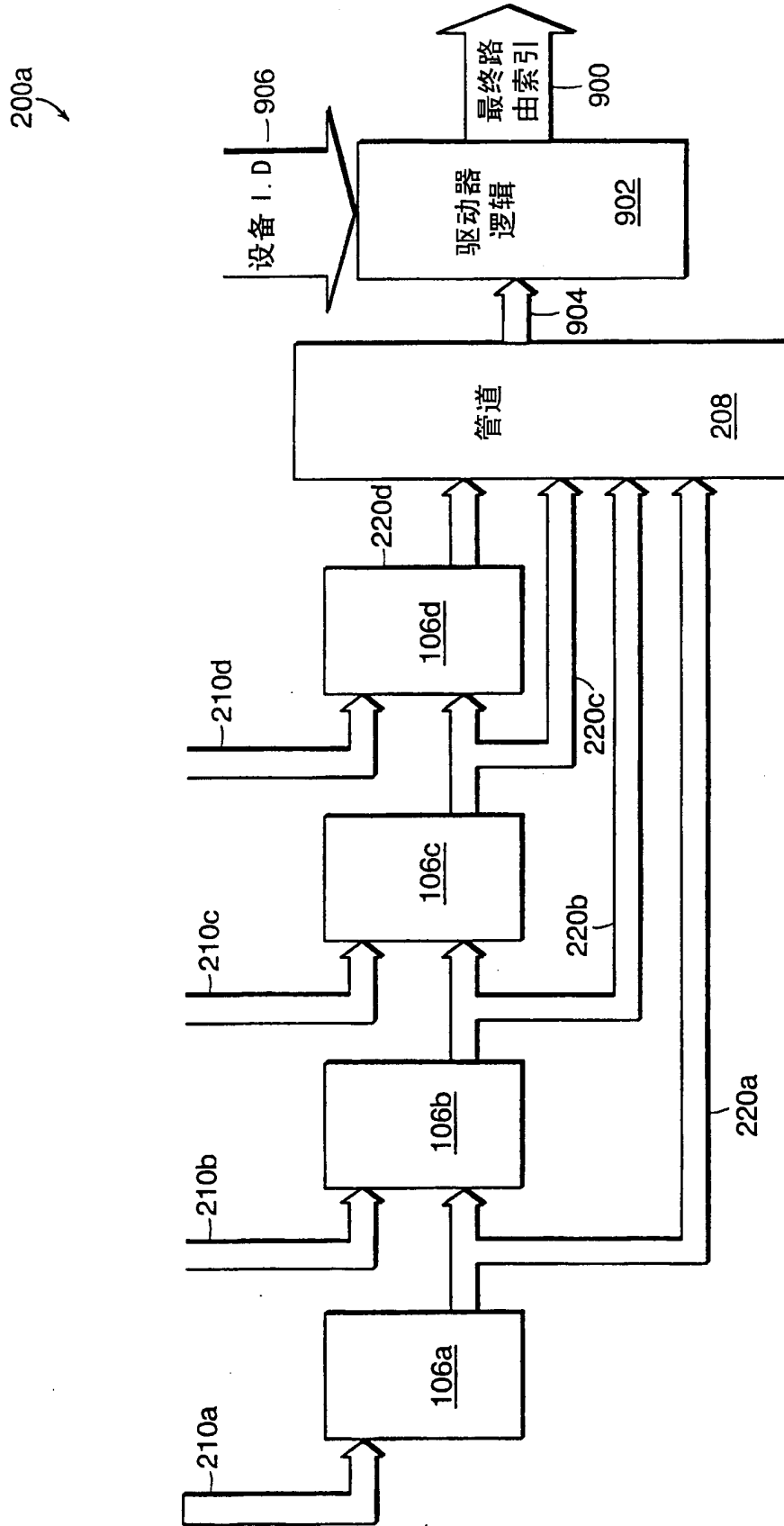


图 10B

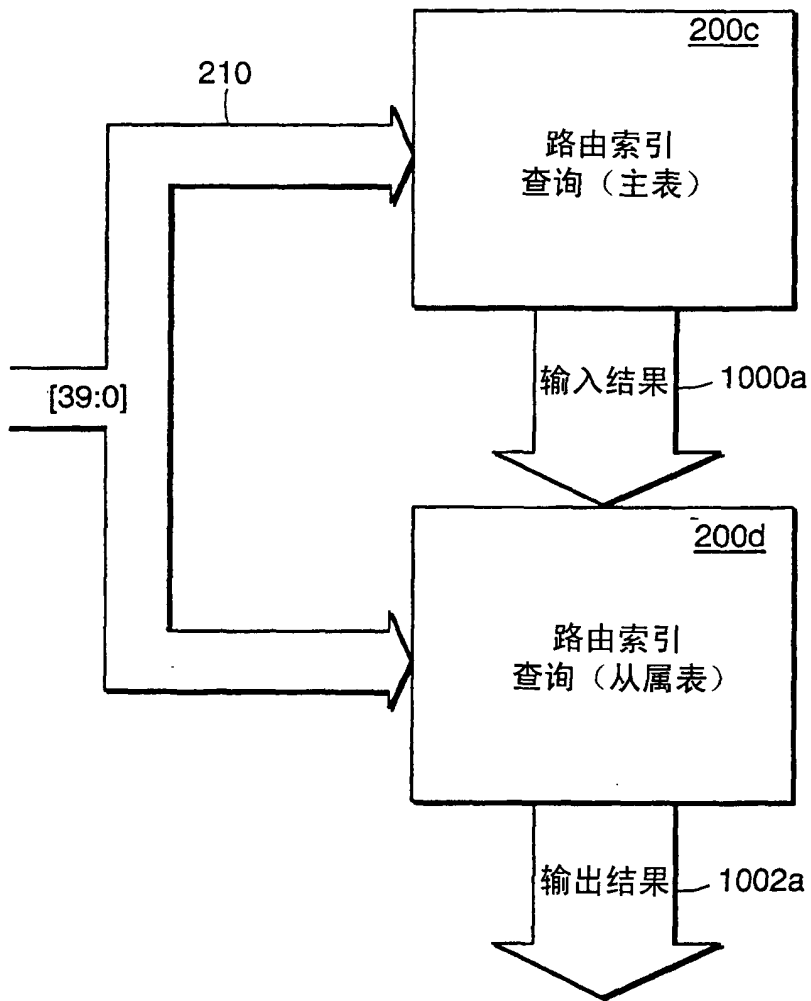


图 10C

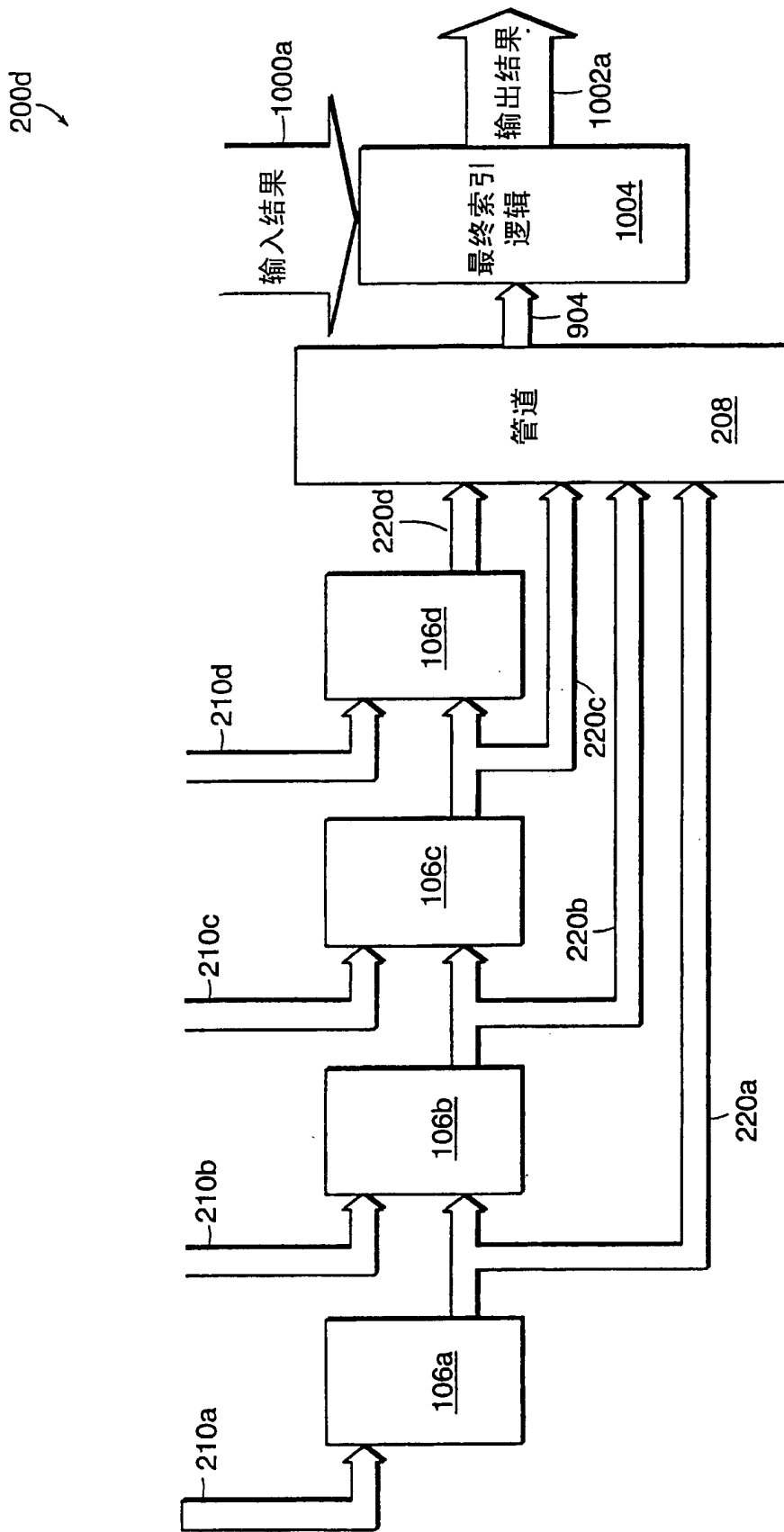


图 10D

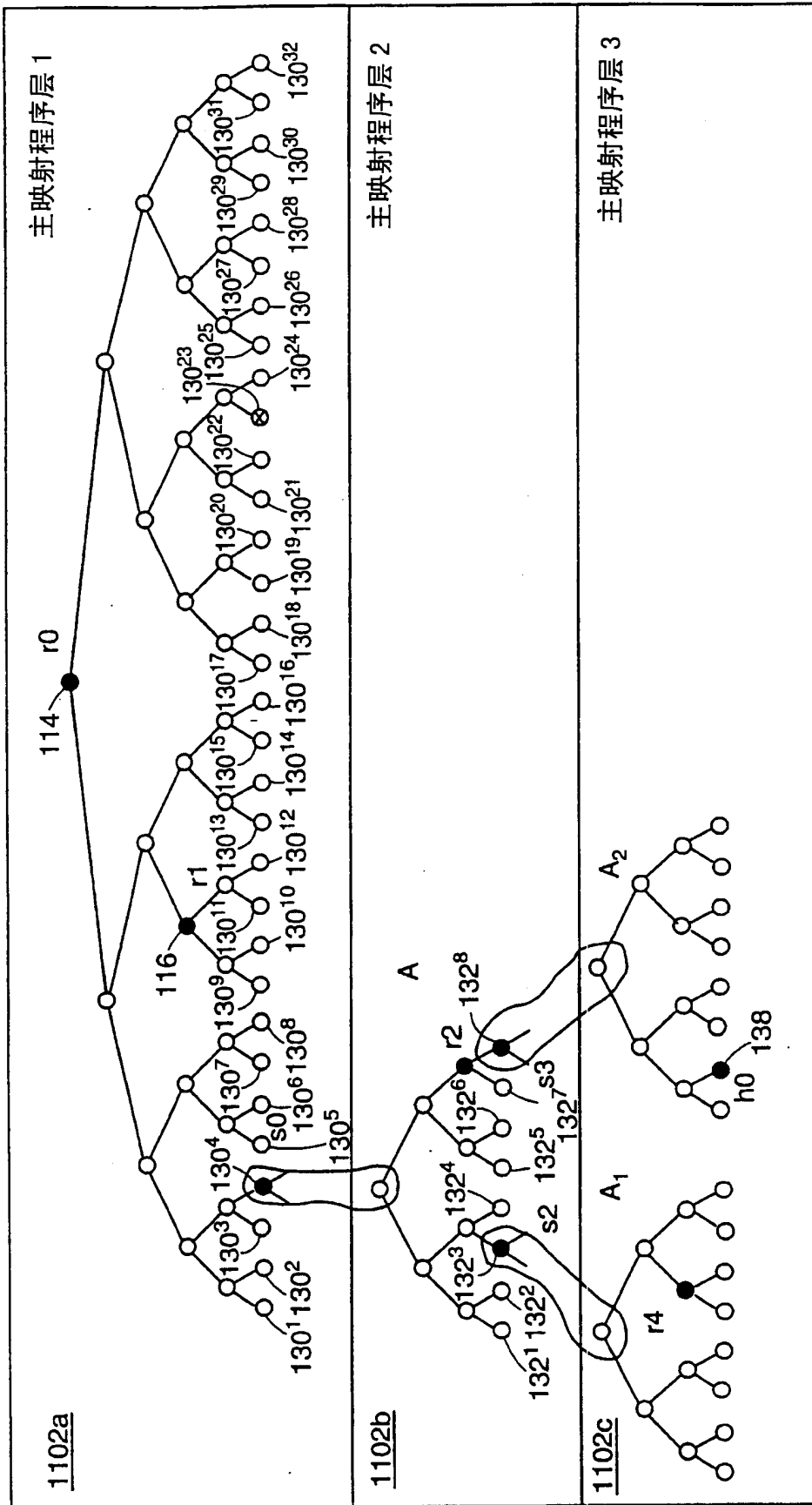


图 11A

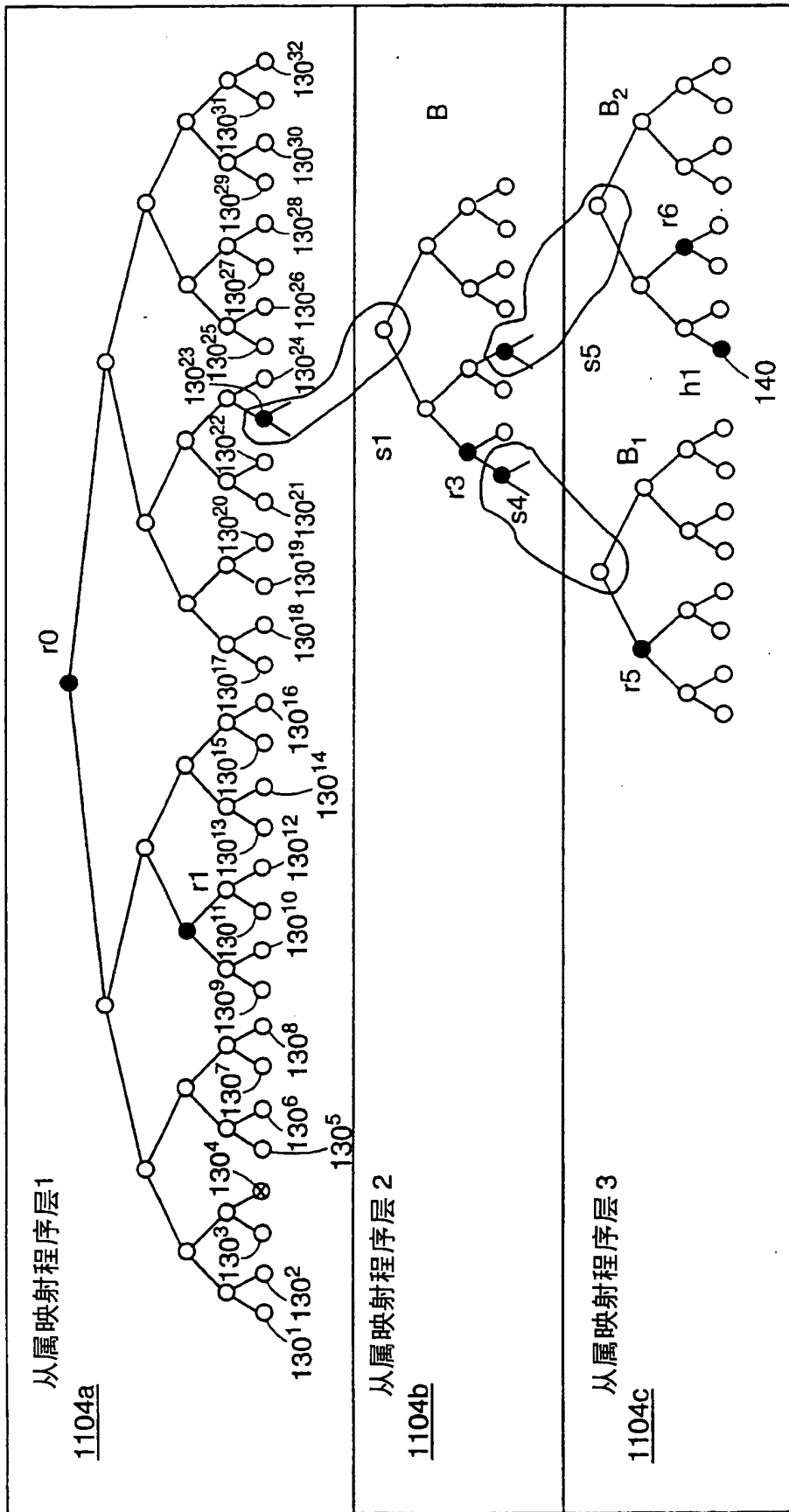


图 11B

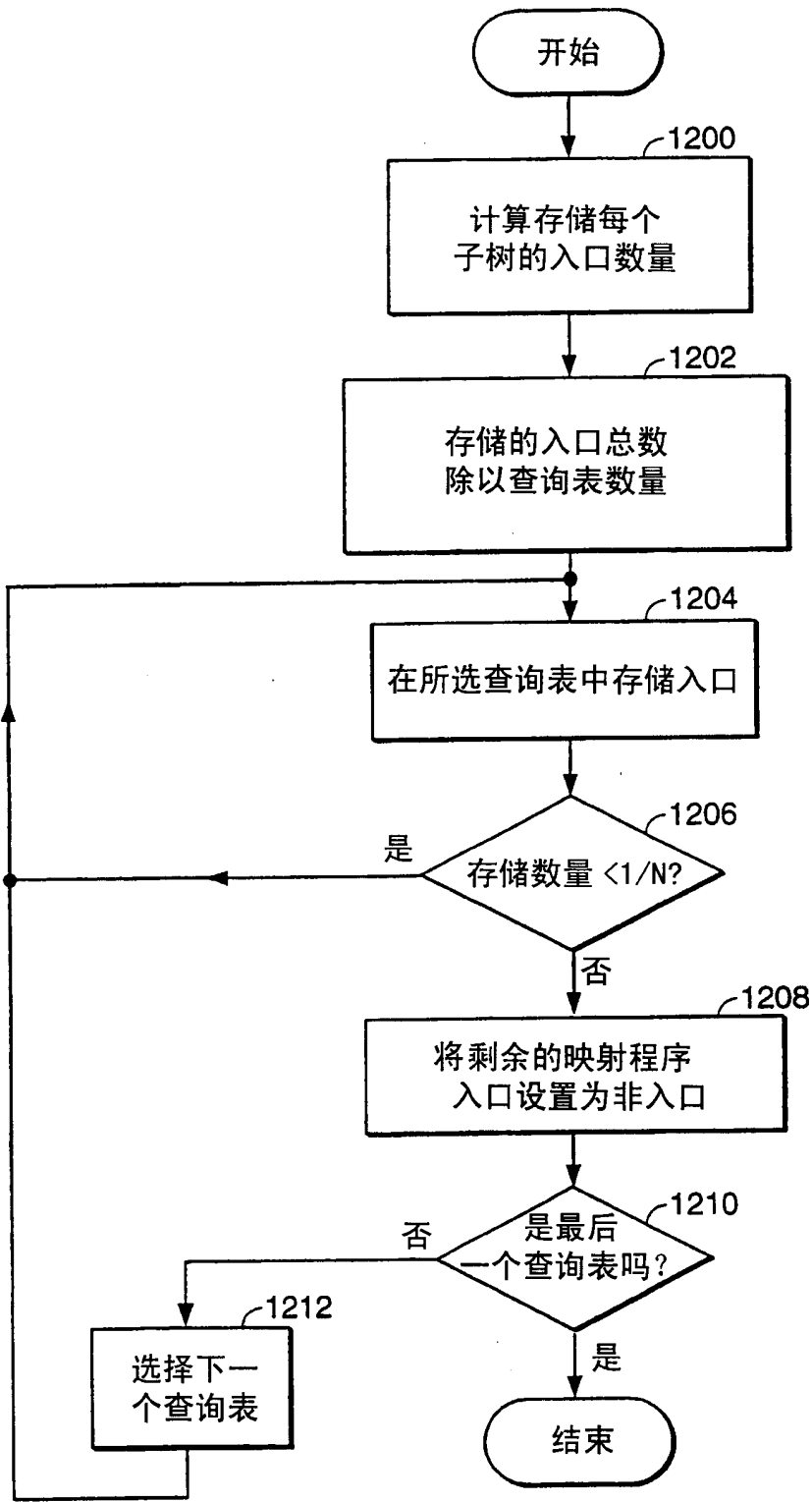


图 12

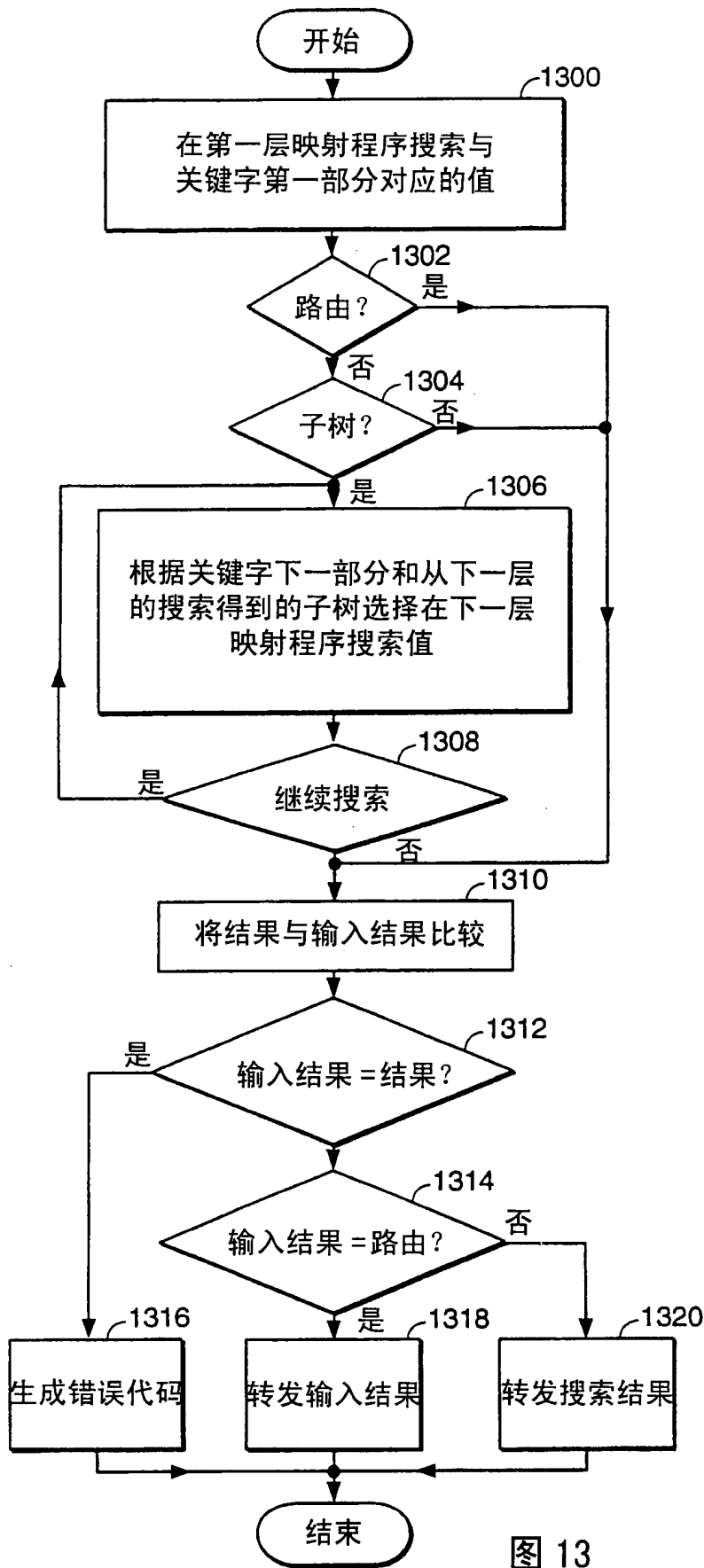


图 13

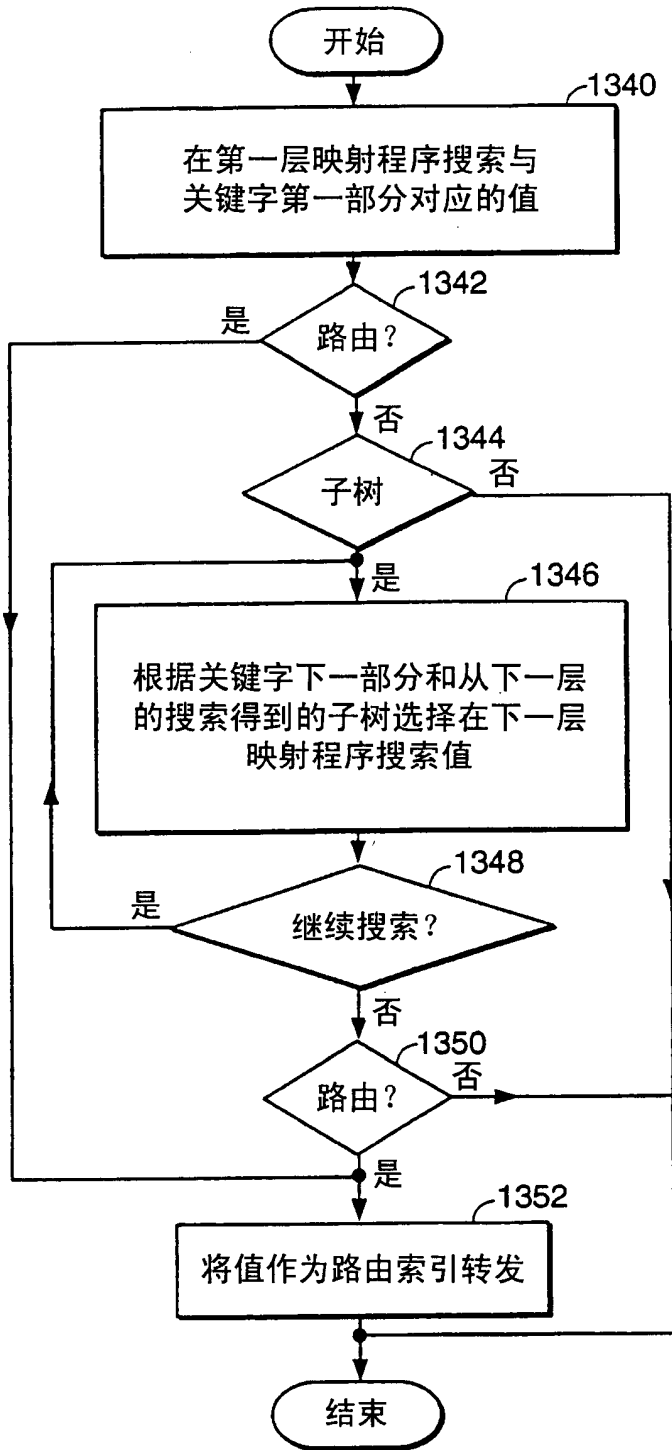


图 14

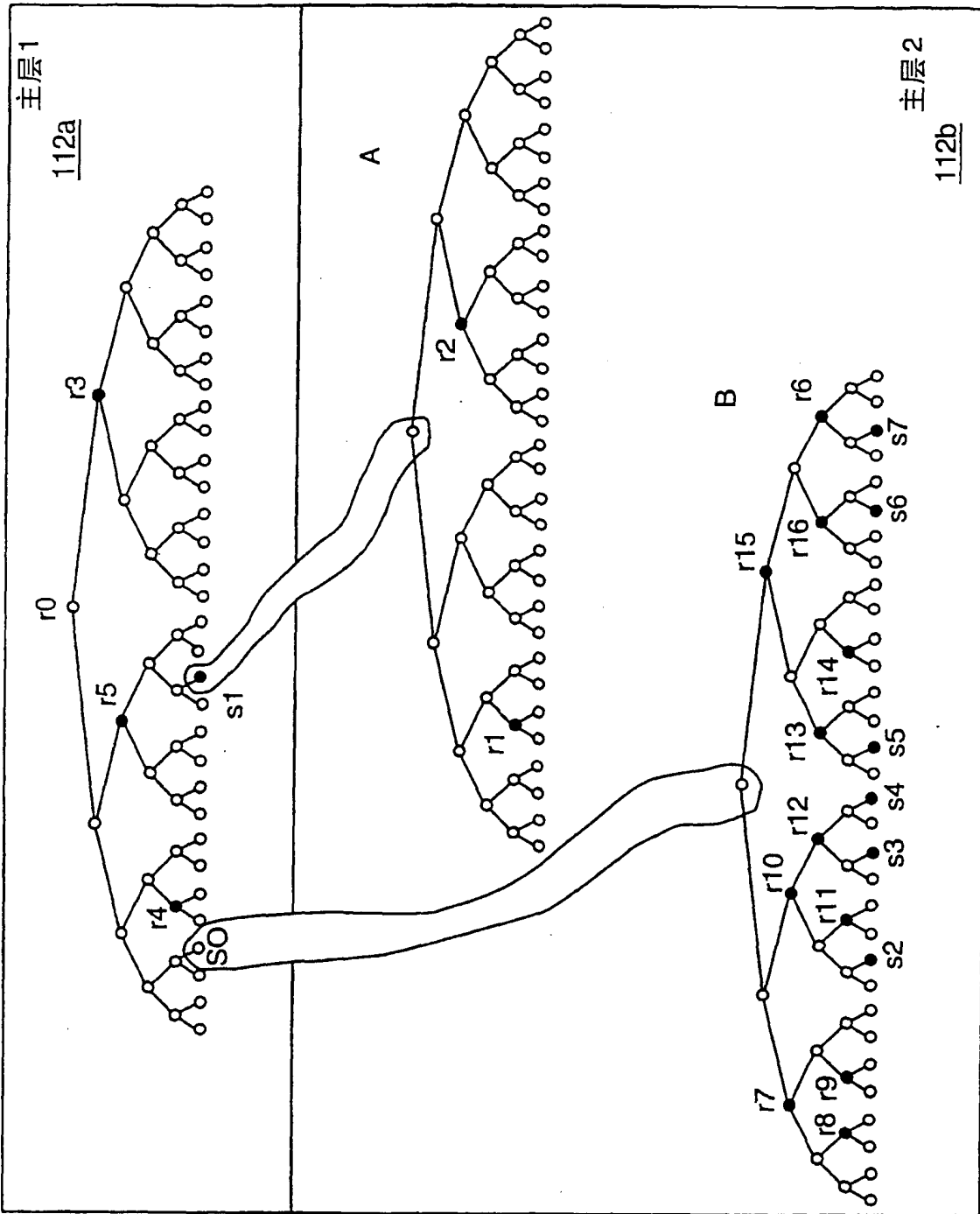


图 15

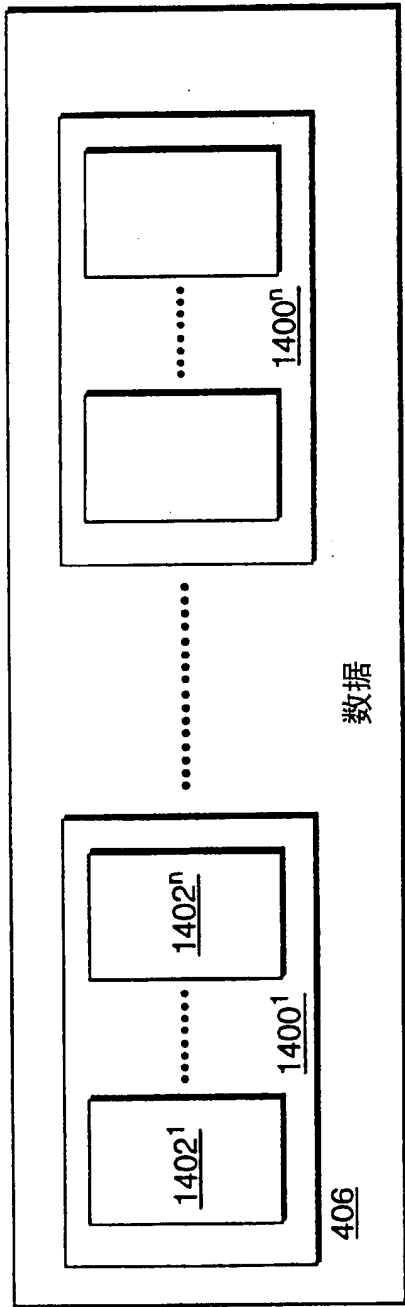


图 16A

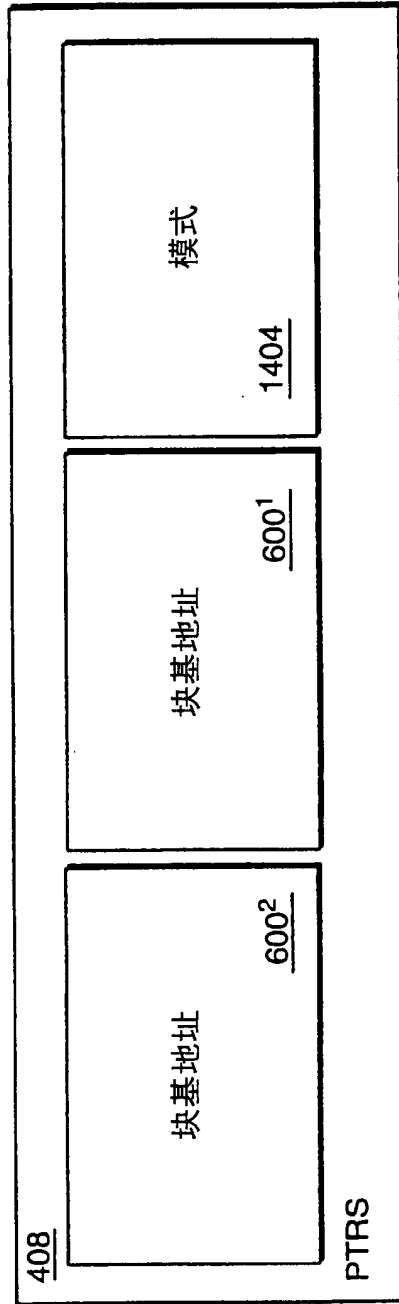


图 16B

子树标志 <u>306</u>	类型 <u>1406</u>	稀疏子树 描述符选择 <u>1408</u>	子树索引 <u>312</u>
--------------------	-------------------	------------------------------	--------------------

图 16C

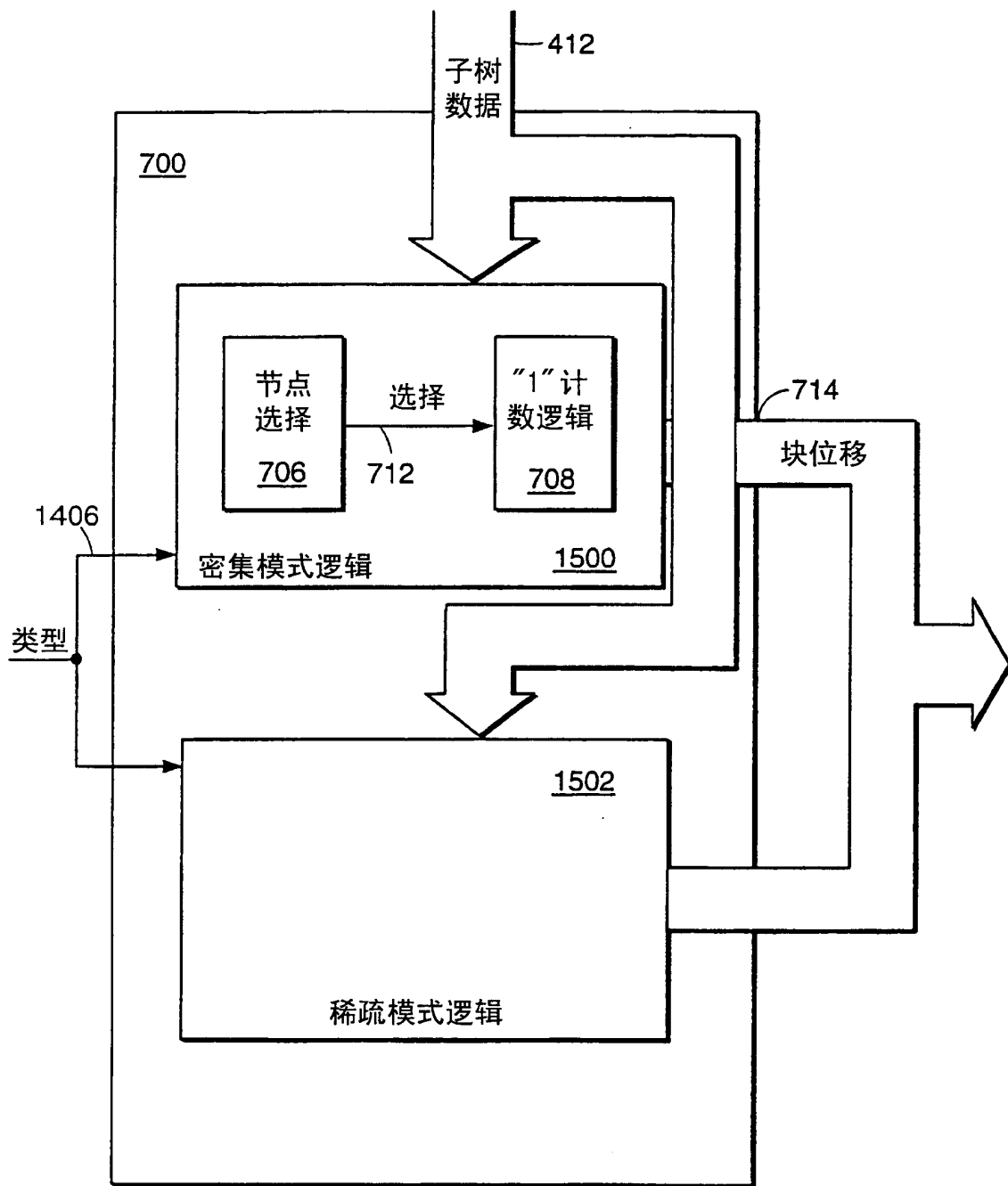


图 17

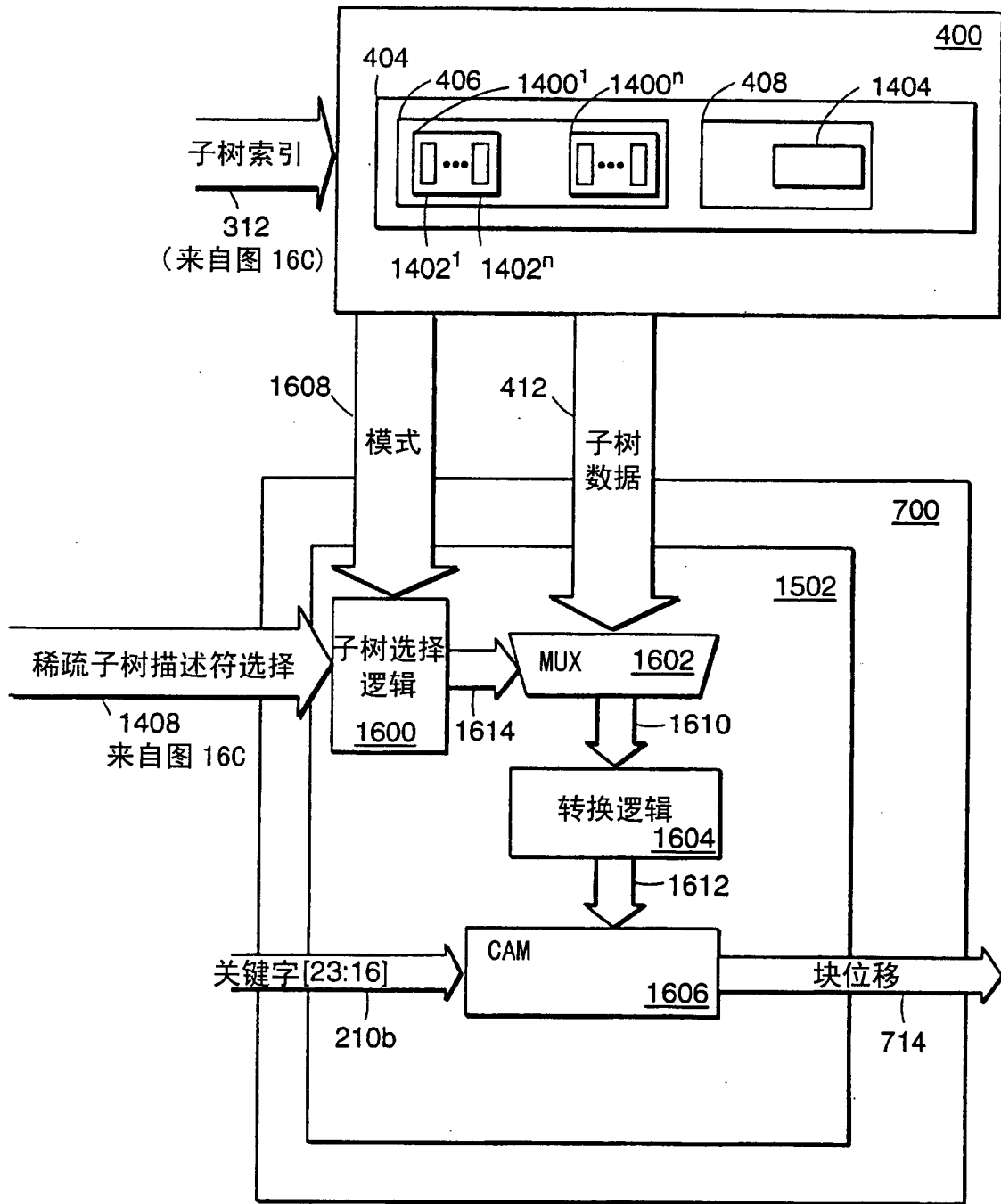


图 18

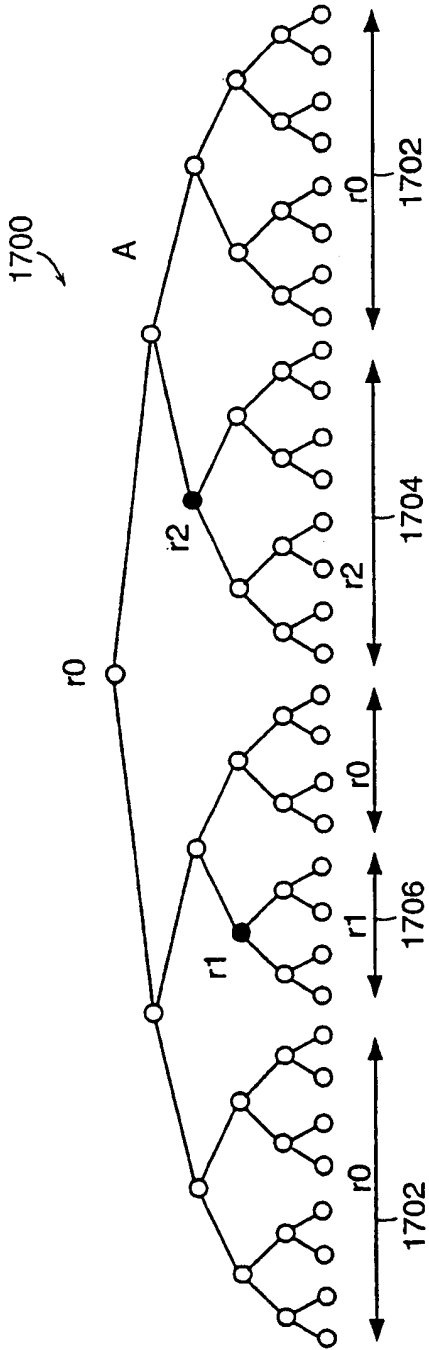


图 19A

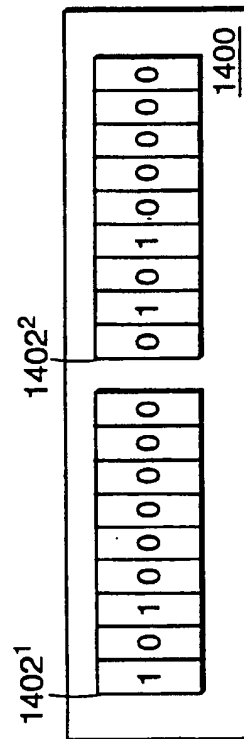


图 19B

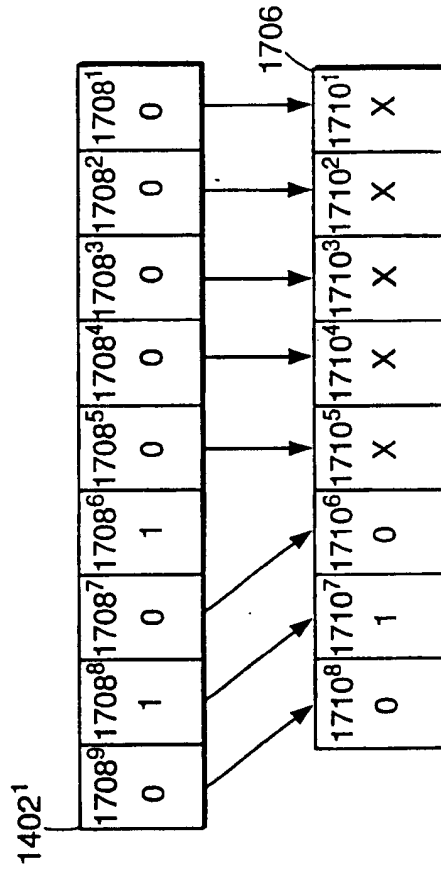


图 19C

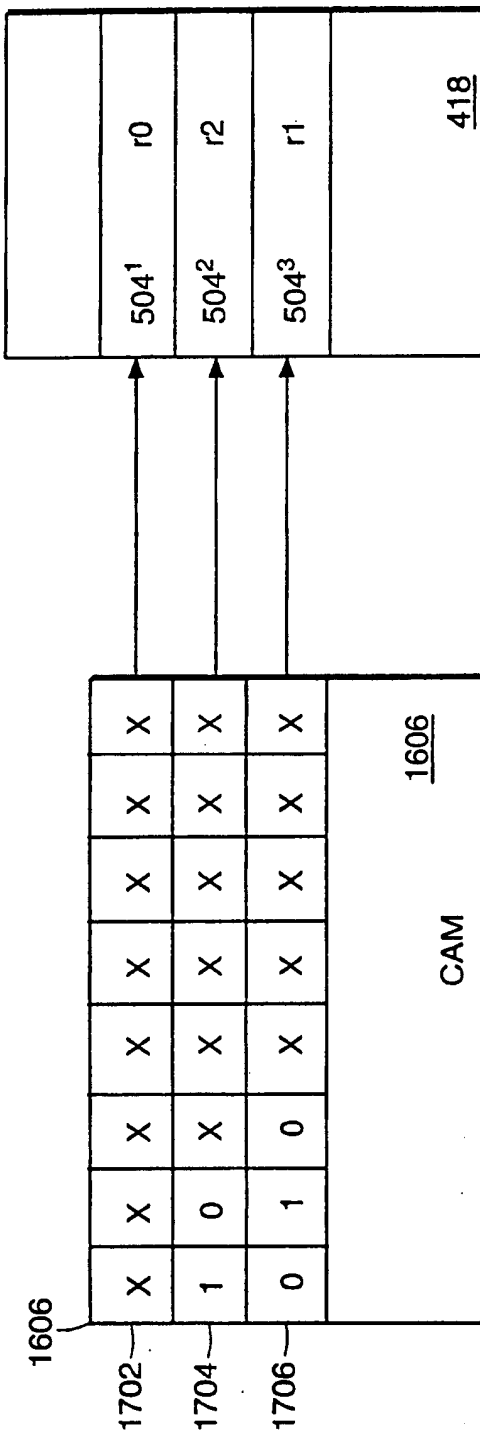


图 19D

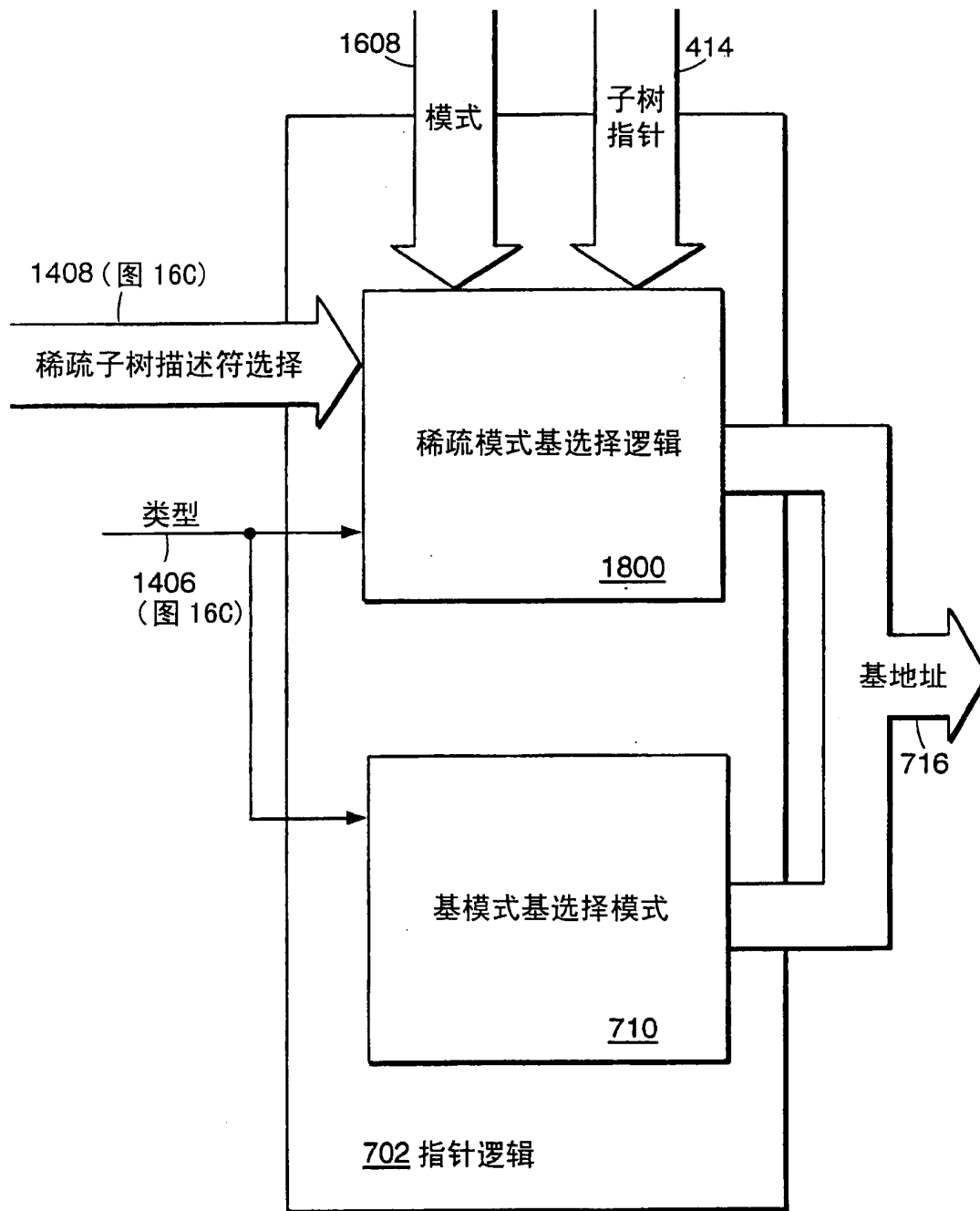


图 20

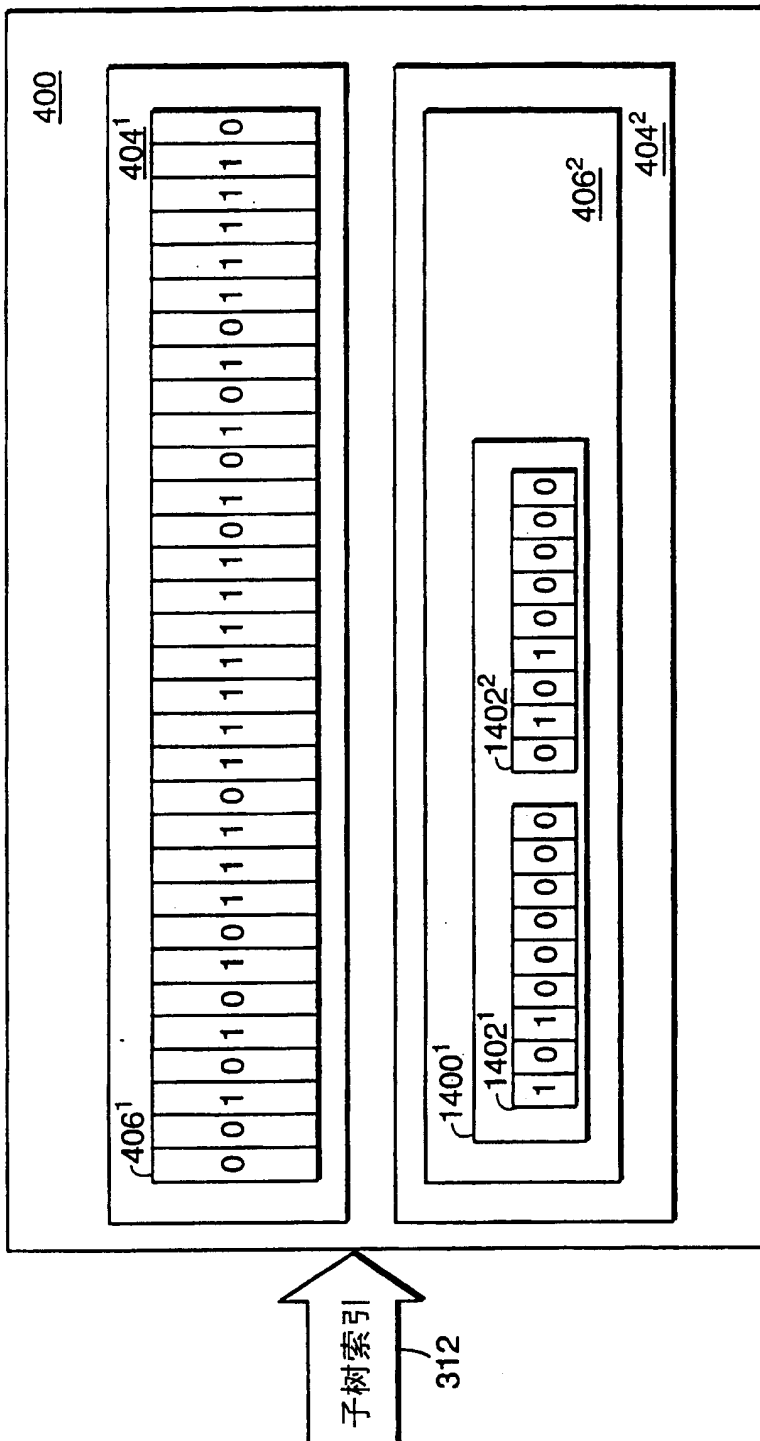


图 21

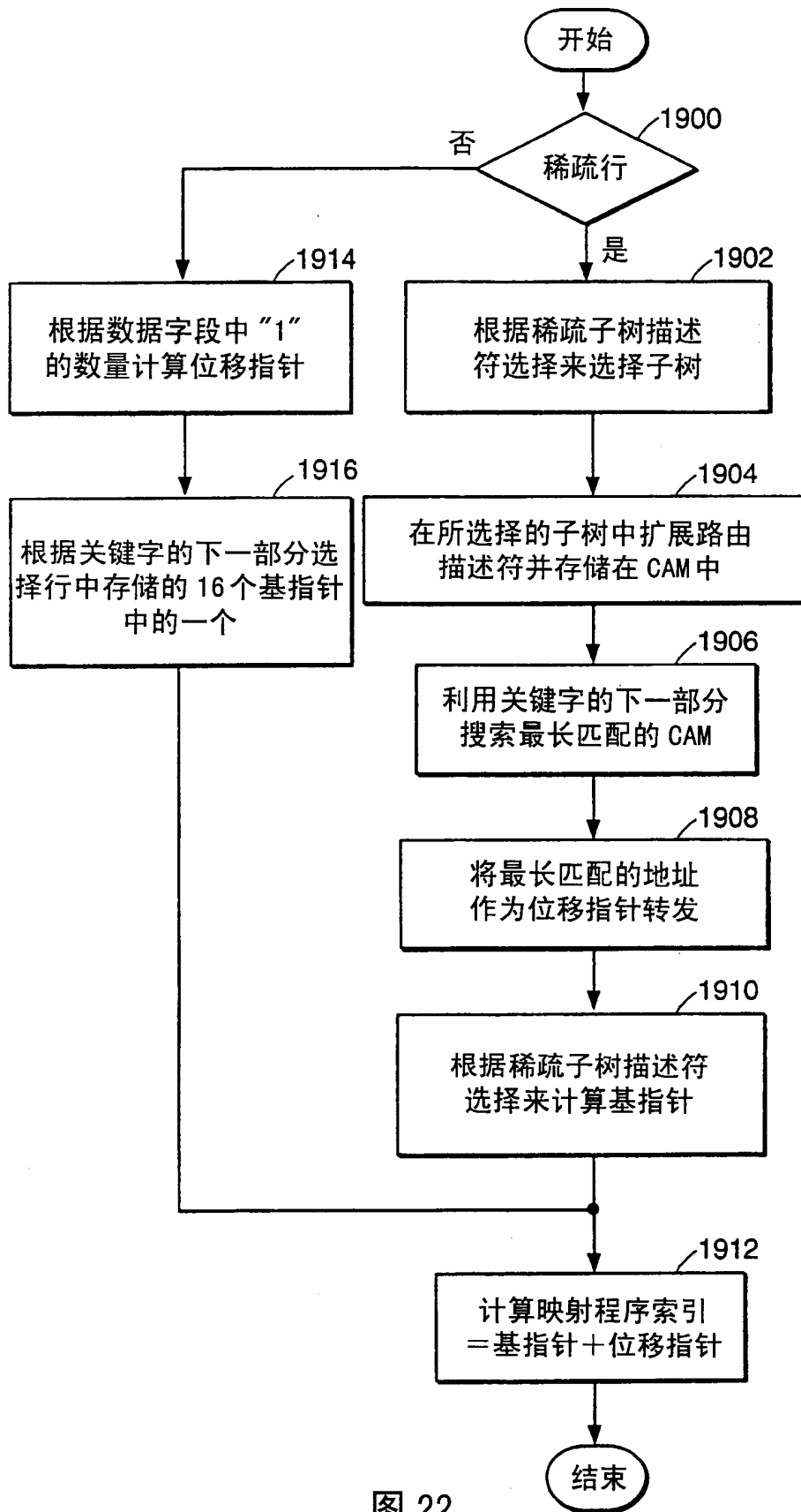


图 22

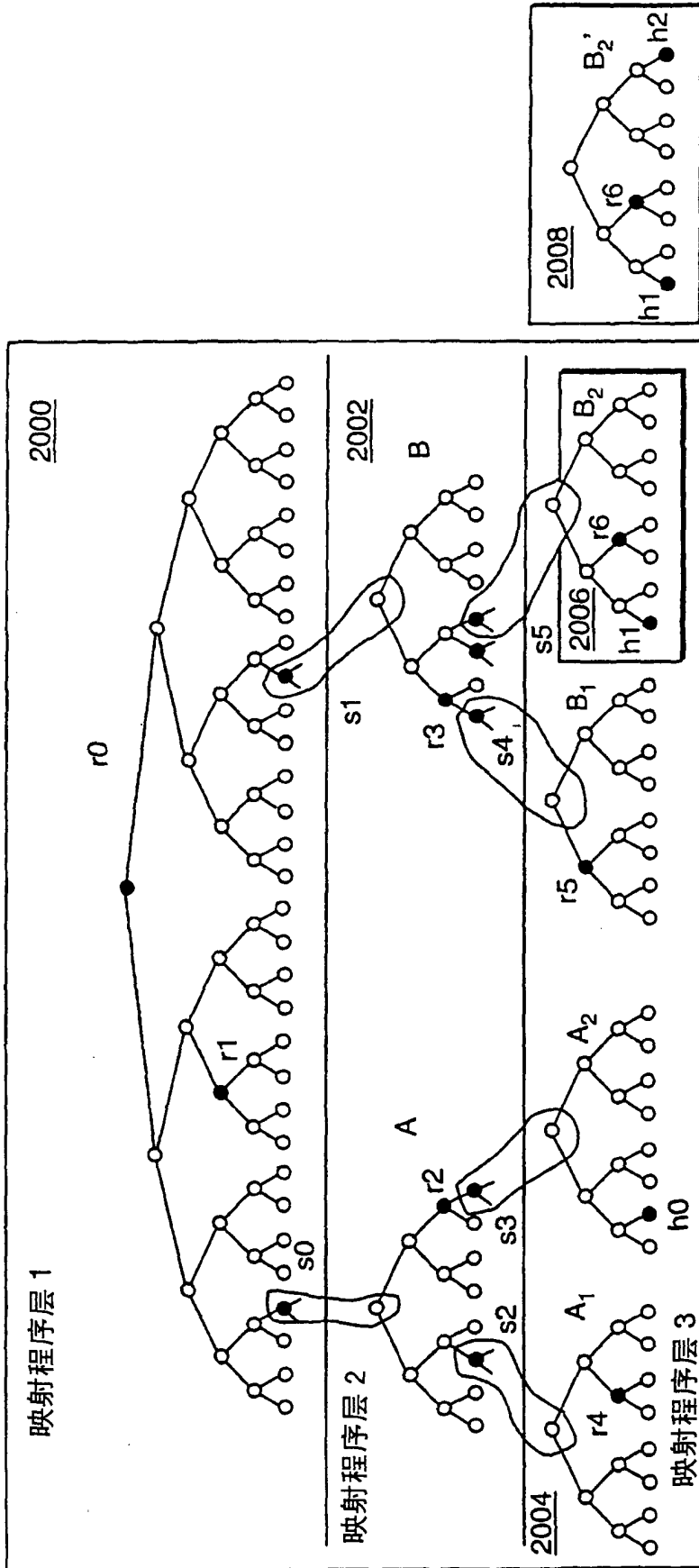


图 23

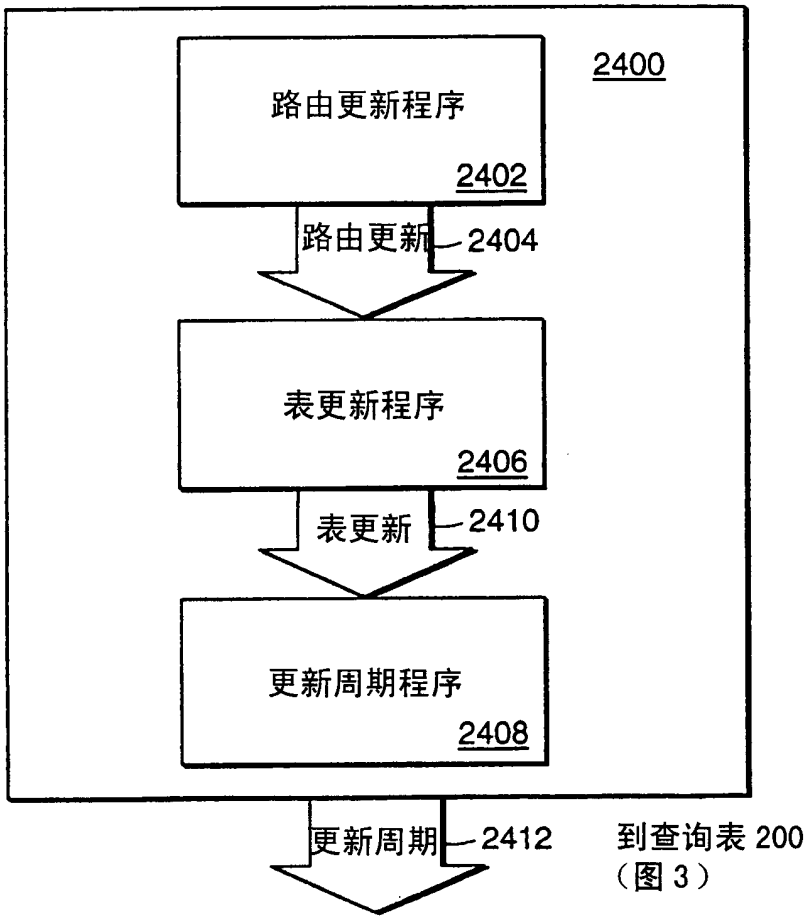


图 24

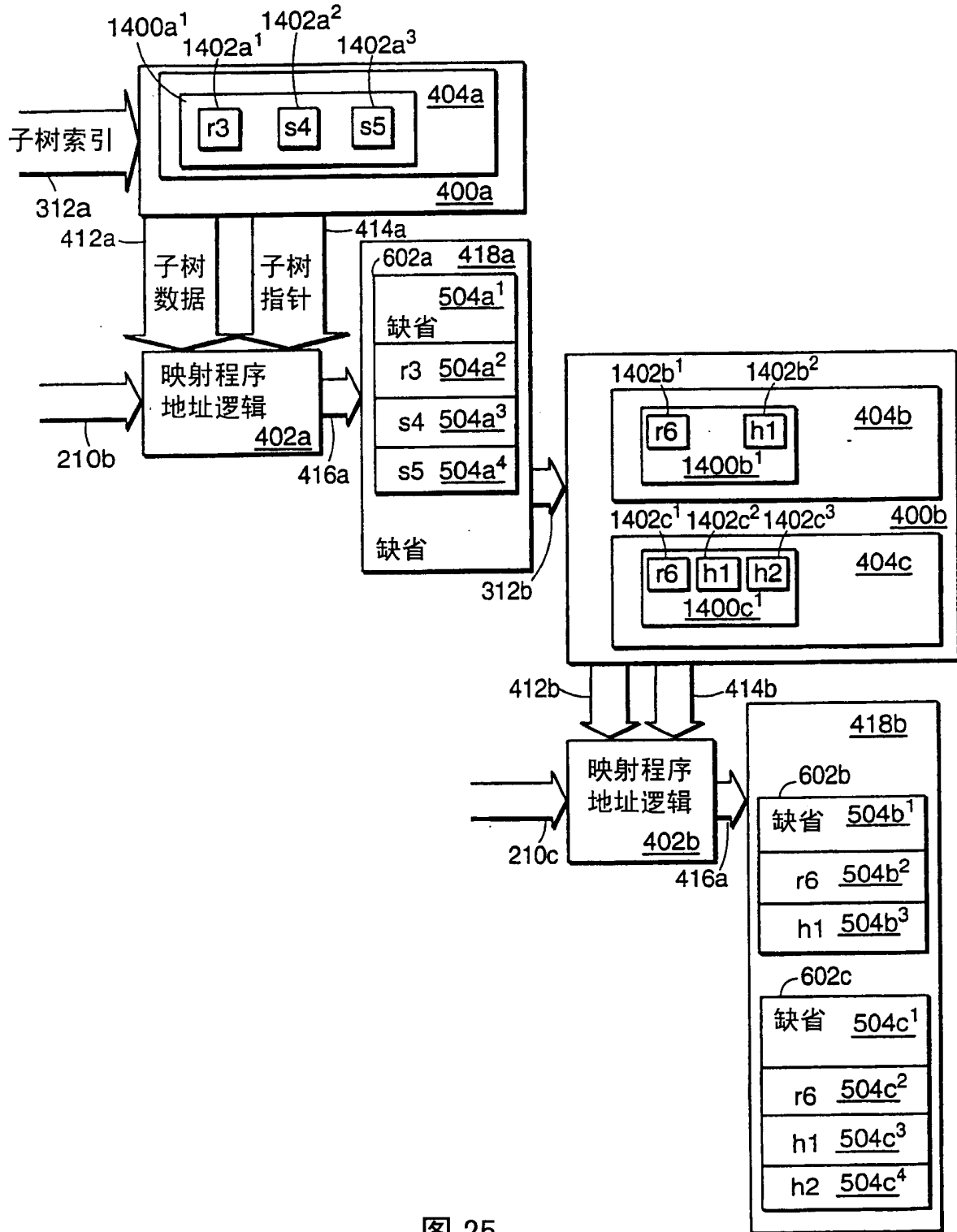


图 25

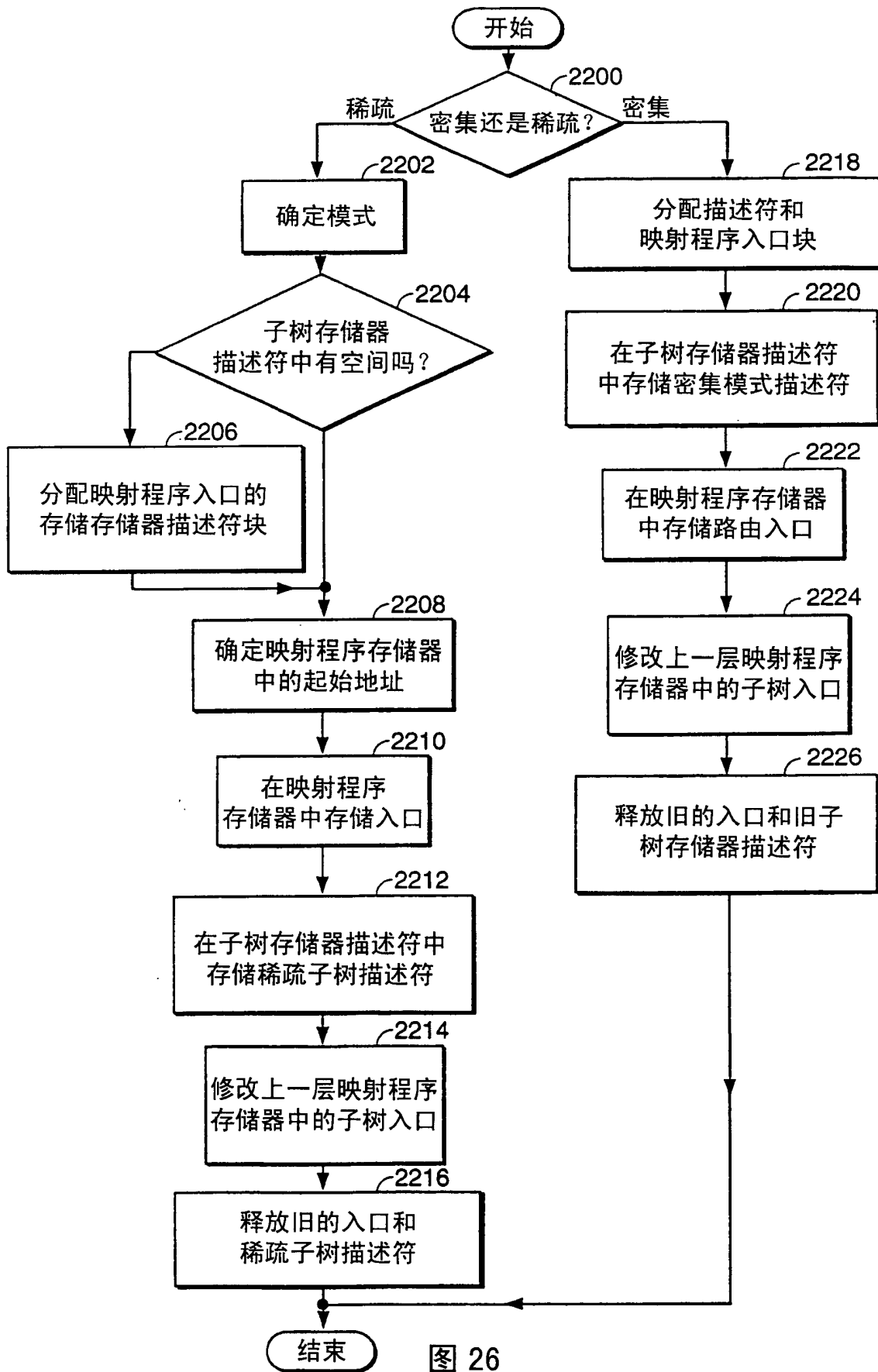


图 26