

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04L 12/56

G06F 12/10 G06F 17/30

H04L 29/06 G06F 12/08



[12] 发 明 专 利 说 明 书

[21] ZL 专利号 00813708.0

[45] 授权公告日 2004 年 12 月 8 日

[11] 授权公告号 CN 1179523C

[22] 申请日 2000.9.18 [21] 申请号 00813708.0

[30] 优先权

[32] 1999.9.30 [33] US [31] 09/409,184

[86] 国际申请 PCT/CA2000/001061 2000.9.18

[87] 国际公布 WO2001/024440 英 2001.4.5

[85] 进入国家阶段日期 2002.4.1

[71] 专利权人 睦塞德技术公司

地址 加拿大安大略省

[72] 发明人 戴维·A·布朗

审查员 陈 军

[74] 专利代理机构 中科专利商标代理有限责任公司

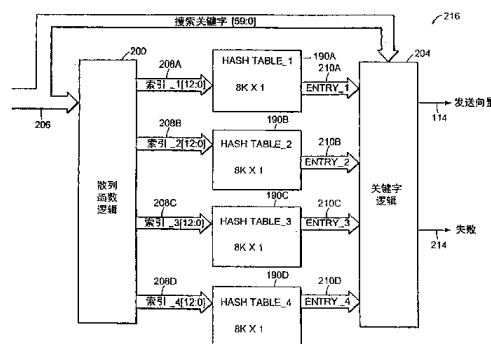
代理人 戎志敏

权利要求书 2 页 说明书 14 页 附图 11 页

[54] 发明名称 用于四路散列表的方法和装置

[57] 摘要

多个散列表被计算所得的每个散列表的不同索引同时访问。每个索引识别散列表之一的存储散列表项的位置。为了搜索发送项，由在搜索关键字上产生的散列函数计算的索引所识别的位置被搜索，以匹配搜索关键字。如果不要单一周期内插入，并且由在插入关键字上产生散列函数计算的索引识别的所有位置都已被占用，为了插入发送项，数据库则被重新排序，以便发送项能够插入由计算所得插入关键字的索引之一识别的位置。如果要求单一周期内插入，并且由计算的索引识别所有位置都已被占用，则由计算而得的插入关键字的索引之一识别的位置之一，可被发送项随机地重写。



1. 一种网络开关, 包括:

多个同样大小的散列表, 每个散列表包括多个位置, 每个位置能存
5 储可被索引访问的发送项;

散列函数逻辑执行装置, 根据依赖于目的地址的关键字, 利用每个
散列表对应的散列函数同时地为每个散列表计算独立的索引, 关键字的
数目大于多个散列表中的位置的总数, 与关键字有关的发送项被存储在
多个被索引的位置之一, 被索引的位置由其它关键字共享, 致使对发送
10 项的搜索在一个搜索周期中被完成;

发送项插入逻辑执行装置, 当检索到由第一关键字计算的索引识别
的散列表中的所有位置都被占用, 其中有一个或多个位置被具有第二关
键字的发送项占用, 将具有第二关键字的发送项之一移动到可用于第二
关键字的其它可用位置。

15 2. 如权利要求 1 所述的网络开关, 其特征在于, 发送项插入逻辑执
行装置, 通过对存储在散列表中的发送项递归地重新定位将发送项之一
移动到新位置, 以便为存储发送项的第一关键字提供计算的索引之一识
别的位置。

3. 如权利要求 1 所述的网络开关, 其特征在于, 发送项插入逻辑执
20 行装置, 在检测位置之一未被占用时将发送项插入为第一关键字计算的
索引识别的未被占用的位置。

4. 如权利要求 1 所述的网络开关, 其特征在于还包括:

关键字匹配逻辑执行装置, 为存储在某一发送项中的关键字搜索由
该关键字计算的索引识别的位置, 根据查找到的匹配关键字提供被存储
25 在该关键字所在位置上的发送向量。

5. 如权利要求 1 所述的网络开关, 其特征在于, 散列函数逻辑通过
产生关键字的 CRC 计算索引, 并映射 CRC 的一部分作为索引。

6. 如权利要求 1 所述的网络开关, 其特征在于, 散列表的数目为四。

7. 如权利要求 1 所述的网络开关, 其特征在于, 发送项插入逻辑不
30 移动, 而是任意地选择由关键字计算的索引识别的被占用的位置之一,

并以发送项重写这个被占用的位置。

8. 一种在网络开关中存储发送项的方法，包括：

提供多个散列表，每个散列表包括多个位置，每个位置能存储可被索引访问的发送项；

5 根据依赖于目的地址的关键字，通过使用单个散列函数的关键字同时地计算散列表的单个索引，关键字的数目大于多个散列表中的位置的总数，与关键字有关的发送项被存储在多个被索引的位置之一，被索引的位置由其它关键字共享，致使对发送项的搜索在一个搜索周期中被完成；

10 当检索到由第一关键字计算的索引识别的散列表中的所有位置都被占用，其中有一个或多个位置被具有第二关键字的发送项占用，将具有第二关键字的发送项之一移动到可用于第二关键字的其它可用位置，将发送项插入为第一关键字计算的索引识别的位置之一。

15 9. 如权利要求 8 所述的方法，其特征在于，通过对存储在散列表中的发送项递归地重新定位移动发送项之一，以便为存储发送项的第一关键字提供计算的索引之一识别的位置。

10. 如权利要求 8 所述的方法，其特征在于，在检测到位置之一未被占用时，将发送项插入为第一关键字计算的索引识别的未占用位置。

11. 如权利要求 8 所述的方法，其特征在于包括：

20 为存储在某一发送项中的关键字搜索由该关键字计算的索引识别的位置；

根据查找到的匹配关键字，提供被存储在该关键字所在位置上的发送向量。

12. 如权利要求 8 所述的方法，其特征在于，计算还包括：

25 产生关键字的 CRC；

映射 CRC 的一部分作为索引。

13. 如权利要求 8 所述的方法，其特征在于，散列表的数目为四。

14. 如权利要求 8 所述的方法，其特征在于，发送项插入逻辑不移动，通过由关键字计算的索引识别的被占用的位置中随机选择一个位置，
30 并以发送项重写所选被占用的位置。

用于四路散列表的方法和装置

5 技术领域

本发明涉及用于四路散列表的方法和装置。

背景技术

在计算机网络中，连网开关从连接于开关的若干入口接收数据包，并将数据包发送至连接于开关的若干出口。开关根据在入口之一接收到的数据包中所含的目的地址确定数据包将要被发送到哪个出口。

数据包将要按目的地址发往的出口被存储在网络开关的散列表的项中。在散列表中，对与具体的目的地址有关的发送项（forwarding entry）进行搜索。

一般来说，为通过开关访问每个可能的目的地址都在散列表中提供项是不行的。例如，为了 48 位以太网目的地址，即介质访问控制（“MAC”）地址，在散列表中要求有 2^{48} 个项，以便为全部可能的 MAC 地址存储发送信息。取代的办法是只有一部分 MAC 地址被存储在散列表中，然后在目的地址的基础上执行散列函数，以提供较小的数目，用来检索散列表。

作为在目的地址上执行散列函数的结果，多个目的地址可以映射到同一索引，即可能出现混淆。因此，向散列表中的一个位置插入项可能被阻断，如果这个位置正存有另一目的地址的发送项的话。

一种解决混淆问题的途径是产生目的地址的第二散列函数，以便在第一索引中找不到散列项时，提供第二索引。但是，由于搜索散列项增加延迟，多级散列法引起一种不确定的搜索。

一种减少混淆并提供确定的搜索的公知技术是在每个索引中提供多于一个的位置。这也被称为多存储桶技术（bucket technique）。在散列表中实施多存储桶，在每个索引中提供预定数的位置或存储桶。

图 1A 是说明现有技术的多存储桶散列表 136 的方框图。散列表 136

包括 2^{15} 个索引(经过索引位 14:0)134。在每个索引中有四个存储桶 138A-D。每个存储桶 138A-D 存储一个发送项。散列表 136 的规模由位置 138-D 的规模和索引 134 的总数确定。例如, 如果每个存储桶 138-D 是一字节宽, 则要求 128 千字节 ($324 \text{ 千字节} \times 4$) 的散列表。

- 5 但是, 多个存储桶的散列表引起存储器的无效使用, 并且, 如果在一个索引中的所有位置都被占用, 发送项的插入可能被阻断。因此, 对于映射到索引的目的地址来说, 没有更多的发送项能被存储在散列表的这个索引中, 即使散列表中别的索引有未被占用的位置可以使用。

10 发明内容

本发明的目的是提供一种用于四路散列表的方法和装置。

为实现上述目的, 一种网络开关, 包括:

多个同样大小的散列表, 每个散列表包括多个位置, 每个位置能存储可被索引访问的发送项;

- 15 散列函数逻辑执行装置, 根据依赖于目的地址的关键字, 利用每个散列表对应的散列函数同时地为每个散列表计算独立的索引, 关键字的数目大于多个散列表中的位置的总数, 与关键字有关的发送项被存储在多个被索引的位置之一, 被索引的位置由其它关键字共享, 致使对发送项的搜索在一个搜索周期中被完成;

- 20 发送项插入逻辑执行装置, 当检索到由第一关键字计算的索引识别的散列表中的所有位置都被占用, 其中有一个或多个位置被具有第二关键字的发送项占用, 将具有第二关键字的发送项之一移动到可用于第二关键字的其它可用位置。

- 25 在优选实施例中, 已确定由关键字计算的索引识别的全部位置被占用, 插入选择逻辑把具有另一关键字的一个发送项移动到可用于其它关键字的另一位置, 然后插入新的发送项。由于这个项的移动可能还引起另一项的移动, 所以发送项可以递归地重新排序。

- 30 当数据包被接收时, 关键字匹配逻辑在计算的索引所指定的地址之一中搜索用于存储在发送项中的搜索关键字的散列表。由于在插入过程中的重新排序, 可以确保在一个索引位置的适当发送项。

散列函数逻辑最好通过由关键字产生的 CRC 的方法为每个散列表计算索引，并选择 CRC 的一部分作为索引，散列表和相应索引的数目最好是 4 个。

5 附图说明

本发明的上述和其他目的、特点和优点从下面对本发明的优选实施例和参考附图的具体描述中将变得清晰，其中，所有不同图中的相同参考符号表示同一部件。附图的比例不是必要的，着重的是对本发明的原理的图解说明。

10 图 1A 是现有技术为提供发送项所使用的多个存储桶的散列表方框图；

图 1B 是根据本发明原理的为提供发送项所使用的散列表的方框图；

图 1C 是能使用图 1B 的散列表的无阻塞公用存储器开关的方框图；

图 1D 是现有技术在入口被接收的现有技术以太网数据包的方框图；

15 图 2 是包含发送逻辑的散列表的搜索逻辑的方框图，这个发送逻辑被表示在图 1C 中的无阻塞公用存储器中；

图 3 是图 2 中所示的散列函数逻辑的方框图；

图 4 是存储在图 1B 所示的任一散列表中的发送项格式的方框图；

图 5 是为搜索匹配一个散列表中的搜索关键字的发送项搜索逻辑的步骤流程图；

图 6 是为在散列表之一中插入新项的插入逻辑的方框图；

图 7 是图 6 中所示的插入逻辑的步骤流程图，用于把发送项插入重写所用项的一个散列表中；

图 8 是图 6 中所示的插入逻辑的步骤流程图，用于递归地重排存储在散列表中的发送项，以便为插入发送项而在散列表中释放一个位置。

具体实施方式

图 1B 是根据本发明原理在网络转接中用于存储发送项的散列表 190A-D 的方框图。每个散列表 190A-D 只提供 2^{13} 个索引，每个索引以 30 一个位置存储发送项。组合的散列表 190A-D 具有 2^{15} ($2^{13} \times 2^2$) 个

索引。因此，四个散列表 190A—D 提供与图 1A 中所示的现有技术多存储桶散列表相同数目的索引，一个散列表具有现有技术多存储桶散列表的四分之一规模。每个索引中的位置数和每个位置的比特数是一种设计选择，而非本发明的限制。

5 根据目的地址，用散列函数逻辑同时地计算四个不同的索引。对散列函数逻辑将参考图 3 作进一步描述。每个计算的索引被提供给不同的散列表 190A—D。例如，根据 Key_A 生成四个不同的索引，即 Key_A index 1—4 172A—D。关键字的发送项被插入由四个计算的索引之一识别的一个位置中，以便在对关键字的依次搜索中，关键字在一个搜索周期内被
10 定位。

 例如，为了在一个搜索周期内提供 Key_A 的发送项，Key_A 必须被存储在由 Key_A Index 1_4 172A—D 识别的位置 138, 146, 156, 164 之一中。如果所有的位置 138, 146, 156, 164 正存储着其他插入关键字的发送项，则发送项之一被移动到散列表 190A—D 之一的另一个位置，以
15 提供用于存储 Key_A 的位置 138, 146, 156, 164 之一。下面参考图 8 描述对存储在散列表 190A—D 中的发送项进行重排的方法，。

 如图 8 所示，四个 Key_A indices 172A—D 识别散列表_1 190A 中的位置 138，散列表_2 190B 中的位置 146，散列表_3 190C 中的位置 156，散列表_4 190D 中的位置 164。所有位置 138, 146, 156, 164 正存储着
20 不同于 Key_A 的关键字的发送项。Key_B 被存储在位置 138，Key_C 被存储在位置 146，Key_D 被存储在位置 156，Key_E 被存储在位置 164。未被占用或空的位置在散列表中可以被使用；例如，位置 140, 142, 144 在散列_1 190A 中是未被占用的。

 为了提供确定的搜索即不必排列的转接，Key_A 必须被存储在由
25 Key_A 172A—D 的索引识别的位置之一中。因此，已被存储在位置 138, 146, 156, 164 的关键字之一必须被移动到未被占用的位置，以便 Key_A 的发送项能被存储在空闲的位置，而被移动的关键字的发送项则被存储到由被移动关键字的其他索引之一识别的一个位置。

 Key_A 与 Key_B 共用位置 138。因此，存储在位置 138 的 Key_B 的
30 项能被移动到由 Key_B 174B—D 的任何其他索引识别的空闲位置。如图

8 所示, 由 Key_B 174B-D 的其他索引识别的其他位置 148, 156, 166 被占用; 所以, Key_B 不能被移动到其他位置 148, 158, 166 之一。但是, 由 Key_C 176A, 176D 的两个其他索引识别的两上位置 140, 168 是空闲的。因此, Key_C 的发送项可从 146 移动到位置 168 或 140。在移动 Key_C 的发送项以后, Key_A 的发送项被插入由 Key_A index_2 172B 识别的位置。对 Key_A 或 Key_C 的发送项的搜索是确定的, 也就是说每次搜索能在一个搜索周期内完成。类似地, Key_D 的发送项能从位置 156 移动到位置 142, 150 或 170, 以提供 Key_A 插入的位置 156, 或者, Key_E 能从位置 164 移动到位置 144, 152 或 162, 以提供 Key_A 插入的位置 164。

因此, 为了提供对存储在散列表 190A-D 中的每个发送项的确定的搜索, 存储在散列表 190A-D 中的每个发送项被重排, 以便与目的地地址有关的发送项被存储在目的地地址而计算的索引之一中。

为使本发明的方法与图 1A 中表示的现有技术方法的比较变得容易, 可考虑将图 1B 的各个的索引位置扩展为每个包括四个存储桶, 每个都如同图 1 所表示的那样。这种结构使用与图 1A 相同的 32K×4 的存储器。进入散列表的每个索引类似地提供四个发送项, 它们可以真正地与其目的地地址匹配, 但与图 1A 的单一索引相对的是, 对于每个关键字, 有四个索引进入散列表。

就本发明来说, 每个索引与其他的目的地地址有较多的共享性。当在图 1B 的方法中插入发送项时, 在任一索引上的冲突更像是由于较多的共享性, 但每个关键字有多个索引和多个存储桶的整个图 1B 的方法不会在插入过程中提供较多的初始冲突。从有利的方面说, 本方法附加地提供了将发送项移动(而不是阻止)到与具体的关键字不共享的其他位置的可能性, 因此, 在插入过程中提供了较大的灵活性。在具体的索引上有可能冲突的任何关键字不大会在发送项移动到的任何其他索引上发生冲突。这个优点能够得到而不必顾及多个存储桶是否被使用, 图 1B 的单个存储桶方法能做到这一点, 而所要求的存储区域减少到四分之一。

虽然插入过程可能需求附加的处理时间, 用于与具体实现有关的递归层次成正比的重新排位, 但处理一次插入所需求的时间大大少于在路

由处理期间的单个周期内寻求匹配所需要的时间。就本发明来说，这种匹配基本上是更为可能的。事实上，在整个存储器规模范围内匹配是肯定的。

图 1C 是公用存储器开关 100 的方框图，其中，图 1B 中所示的散列表可被使用。在入口 102 接收到的全部数据在被转接至一个或多个出口 112 之前被存储在分段缓冲存储器 108 中。数据包存储管理器 106 控制访问分段缓冲存储器 108 的写和读。

开关 100 包括入口引擎 104 和出口引擎 110。数据包串行地在入口 102 被接收。在入口引擎 104 的发送逻辑单元 128，入口引擎 104 检测和处理所接收的数据包的报头，并选择存储在入口引擎 104 中的发送逻辑 128 的用于数据包的发送项中的发送向量 114。发送向量是位映射，对应多个出口 112 的每一个的位，并指示数据包是否要被发送至依赖报头中的目的地址的那个出口 112。发送向量 114 被向前传递至数据包存储管理器 106。例如，目的地址 X 的数据包经出口_1 112A 被发送至目的地址为 136A 的网络节点；目的地址为 Y 的数据包经出口_2 112N 被发送至目的地址为 Y 136C 的网络节点；目的地址为 Z 的数据包经出口_N 112N 被发送至目的地址为 Z 136B 的网络节点。

数据包存储管理器 106 提供到分段缓冲存储器 108 的通路。数据包存储管理器 106 向分段缓冲存储器 108 提供由于读和写操作的分段缓冲存储器地址 122，并在管理器 106 中存储每个数据包在分段缓冲存储器 108 中的存储位置。出口引擎 110 通过对控制信号 120 的选择，选择要在那里发送一个数据包的多个出口 112 之一，并向所选的出口 112 提供所存储的数据包。

分段缓冲存储器 108 是由所有入口 102 和出口 112 共享的公用存储器。开关 100 是无阻塞的，就是说，在任何入口 102 到达的数据包无阻塞地被发送至任何出口 112。开关 100 提供并行处理，入口引擎 104 对入口 102 接收到的数据包进行处理，出口引擎 110 对被存储的数据包进行处理，以供应出口 112。

图 1D 是表示可在入口 102 接收的现有技术以太网数据包 120 的方框图。以太网数据包包括报头 122、数据段 134 和帧检验序列 132。报头

122 包括目的地址 122、源地址 126 和长度或类型段 130。源地址 126 和目的地址 124 是唯一的 48 位地址，分别识别源端和目的端的物理设备。数据包也可以包括 12 位 VLAN Identifier (“VID”) (未示)。数据包 120 的大小依赖于数据段 134 的大小，它可从 46 字节变化至 1500 字节。

5 图 2 是根据本发明的原理的图 1C 无阻塞公用存储器开关 100 的入口引擎 104 中所示的发送逻辑 128 中的搜索逻辑的方框图。搜索逻辑 216 包括四个散列表 190A-D。散列表 190A-D 的数目是一种设计选择，而非本发明的限制。

搜索逻辑 216 还包括散列函数逻辑 200 和关键字匹配逻辑 204。散列函数逻辑 200 根据搜索关键字并行地生成四个索引 208A-D。搜索关键字 206 类似于超高速存储器的标记。关键字匹配逻辑 204 确定由索引 208A-D 识别的四个位置中的那个位置存储与搜索关键字 206 有关的发送项。在找到匹配的发送项之后，这个发送项的发送向量 114 部分被发
10 送至数据包存储管理器 106 (图 1C)。

15 为搜索与数据包报头中的目的地址对应的发送项，随着数据包正在被入口 102 (图 1C) 之一接收，入口引擎 104 检查数据包 120 的报头 122 (图 1D)。48 位目的地址 124 (图 1D) 的复本与 12 位 VLAN Identifier (“VID”) 拼接，形成 60 位搜索关键字 206。目的地址与 VID 的拼接是一种设计选择，而非本发明的限制，因此，搜索关键字可以是目的地址
20 124 (图 1D)。搜索关键字 206 被发送至散列函数逻辑 200。散列函数逻辑 200 提供四个 13 位索引 208A-D，作为在搜索关键字上执行散列函数的结果。

图 1D 中所示的以太网数据包 120 描述了本发明。但本发明不局限于以太网数据包，它可被用于任何其他类型数据包的发送项的插入和搜索。
25

图 3 是说明图 2 中所示的散列函数逻辑 200 的方框图。散列函数逻辑 200 包括周期性冗余码检验 (“CRC”) 生成逻辑 300 和散列表索引逻辑 302。散列函数是 CRC 函数，它是在 CRC 生成逻辑 300 中在搜索关键字 206 上被完成的。CRC 生成逻辑是那些熟练的技术人员已知的。散列函数不限于 CRC 生成，被使用的散列函数可以是那些熟练的技术人员
30

所用的任何散列函数。CRC 生成逻辑从 60 位搜索关键字 206 生成 32 位 CRC 304。

32 位 CRC 304 被发送至散列表索引逻辑 302。散列表索引逻辑 302 从 32 位 CRC 304 生成四个索引 208A-D。四个索引 208A-D 每个有 13 位，因此提供 8K 地址，以在每个散列表 190A-D 中寻址 8K 位置。通过将 32 位 CRC 304 的位映射至四个索引 208A-D，生成四个索引 208A-D，方法如下：CRC 304 的位 12: 0 被映射至索引_1 208A 的位 12: 0；CRC 304 的位 18: 6 被映射至索引_2 208B 的位 12: 0；CRC 304 的位 25: 13 被映射至索引_3 208C 的位 12: 0；CRC 304 的位 31: 19 被映射至索引_4 208D 的位 12: 0。

回到图 2，在索引 208A-D 已被散列函数逻辑 200 生成以后，存储在由索引 208A-D 指定的散列表 190A-D 中的位置上的项 210A-D 被发送至关键字匹配逻辑 204。搜索关键字 206 也被发送至关键字匹配逻辑 204。

图 4 是被存储在图 2 中所示的散列表 190A-D 中任一位置上的发送项格式的方框图。发送项 210 包括指示数据包在哪里被发送的字段。发送项 210 包括下列字段：使用期限标志 402，远程超高速缓存更新标志 404，学习端口号码 406，地址相关的 QoS 有效 408，地址相关的 QoS 410，逻辑端口发送向量 412，静态项指示器标志 414，类型标志 416，有效项指示器标志 418，关键字 420。

一位使用期限标志 402 指明发送项 210 的使用期限。一位远程超高速缓存更新标志 404 指示超时是否被允许。学习端口号码 406 指示发送项中的关键字 420 中存储的 MAC 地址在哪个界面上被学习过。例如，如果 MAC 地址在局部外部访问端口被学习过，这就是实际的入口号码 104。关于地址的服务质量（“QoS”）字段 410，指示被指定的关于地址有服务质量管理。一位地址相关的 QoS 有效字段 408 指示 QoS 字段 410 是否被用于以后的排队考虑。32 位逻辑发送向量 412 指示数据包将被发送至哪个端口。一位静态项指示器标志 414 指明发送项是静态还是动态。一位类型标志 416 指示发送项是否能被用于 2 级发送项。一位有效端口指示器标志 418 指示发送项 210 是否有效。60 位关键字 420 存储与发送

项 210 有关的搜索关键字。

回到图 2，关键字匹配逻辑 204 将存储器在从散列表 190A-D 发送的发送项 210A-D 中的关键字项 420 与搜索关键字 206 进行比较。如果存在匹配，则 32 位逻辑端口发送向量 412（图 4），从发送向量 114 上，
5 发送项在发送向量 114 上被发送至数据包存储管理器 106（图 1）。如果不存在匹配，则失败信号 214 被生成，并由入口引擎 104 中的不匹配发现逻辑（未示）处理。

图 5 是一个流程图，用于搜索散列表 190A-D 中与搜索关键字匹配的发送项 210 的搜索逻辑 216 的步骤。

10 在步骤 500 中，搜索逻辑 216 等待搜索请求。如果有搜索请求，则处理过程继续至步骤 502。如果没有，则搜索逻辑 216 继续等待搜索请求。

在步骤 502 中，散列函数逻辑 200 从搜索关键字 200 生成四个索引 208A-D，并发送四个索引 208A-D 至散列表 A-D。处理过程继续至
15 步骤 504。

在步骤 504 中，存储在由索引 208A-D 指定的散列表 190A-D 的位置上的发送项 210A-D 被发送至关键字匹配逻辑 204。处理过程继续至步骤 506。

在步骤 506 中，关键字匹配逻辑 204 将存储在每个发送项 210A-D 中的关键字 420（图 4）与搜索关键字 206 进行比较，并确定是否匹配。
20 如果匹配，处理过程继续至步骤 508。如果不匹配，处理过程继续至步骤 510。

在步骤 508 中，用于搜索关键字 206 的发送项 210A-D 已被查找到。存储在发送项 210 中的逻辑端口发送向量 412（图 4）在发送向量 114 上
25 （图 1C）被发送至数据包存储管理器 106（图 1C）。数据包经过数据包存储管理器 106、分段缓冲存储器 108 和出口引擎 110 被发送至预定的出口 112。处理过程继续至步骤 500。

在步骤 510 中，没有查找到与搜索关键字 206 匹配的发送项 210A-D。入口引擎 104 用如扩散法处理这一次失败，即将数据包发送至所有
30 出口引擎 112。这种处理是桥式协议的一部分。处理过程继续至步骤 500。

图 6 是一个方框图，表示为在散列表 190A—D 之一中的位置上插入发送项 210，在发送逻辑 128（图 1C）中的插入逻辑 608。48 位目的地址 124（图 1D）的复本与 12 位 VLAN Identifier（“VID”）拼接，以形成 60 位插入关键字 606。目的地址和 VID 的拼接是一种设计选择，而非本发明的限制。同一散列函数在插入关键字 606 上被完成，如同在搜索关键字上被完成那样。搜索关键字 206 的散列函数已结合图 3 描述过。四个索引 208A—D 作为散列函数的结果被生成。每路索引识别在各自的散列表 190A—D 中可插入关键字 606 的位置。

存储在由索引 208A—D 识别的散列表 190A—D 的位置上的发送项 210A—D 被发送至发送项插入逻辑 600。发送项插入逻辑 600 根据散列表 210A—D 的内容和重写信号 602 的状态，确定在由索引 208A—D 识别的哪个位置上把插入关键字 606 插入。

发送项插入逻辑 600 确定由索引 208A—D 指定的位置中哪个位置未被占用。如果找到未被占用的位置，则包含插入关键字 606 的发送项 210 在由索引 208 识别的未被占用的位置上被插入。如果没有找到未被占用的位置，而且重写信号 602 指示允许重写，则索引 208A—D 被随机选择，并且包含插入关键字 606 的发送项被插入由所选择的索引 208 识别的位置上。

如果所有的位置都被占用，而且重写信号指示重写不被允许，则存储在散列表 190A—D 中的发送项被执行重排，以提供一个由索引 208A—D 识别的未被占用的位置，插入包含插入关键字 606 的发送项。将结合图 8 描述在散列表 190A—D 中重排发送项 210A—D 的方法。图 7 是表示图 6 所示插入逻辑 608 中的步骤流程图，用于将包含插入关键字 606 的发送项插入散列表 190A—D 之一中的一个位置。

在步骤 700 中，散列函数逻辑 200 接受关于将发送项插入散列表 190A—D 之一中的一个位置上的请求。处理过程继续至步骤 702。

在步骤 702 中，散列函数逻辑 200 根据插入关键字 606 同时生成并行的四个索引 208A—D。索引 208A—D 的生成已结合图 3 描述过。处理过程继续至步骤 704。

在步骤 704 中，由生成的索引 208A—D 识别的散列表 190A—D 中

的位置的内容被发送至发送项插入逻辑 600。处理过程继续至步骤 716。

在步骤 716 中，如果插入关键字 606 已被存储在由生成的索引 208A-D 识别的位置，则在这个位置上重写发送项插入。

在步骤 706 中，发送项插入逻辑 600（图 6），根据被发送的内容确定位置是否被占用。例如，用检查发送项 210 中的有效项字段 418（图 4）的方法。如果位置之一未被占用，处理过程继续至步骤 708。如果由生成的索引 208A-D 识别的所有位置已被占用，处理过程则继续至步骤 710。

在步骤 708 中，发送项插入逻辑 600（图 6）将包含插入关键字 606 的发送项 210 插入由生成的索引 208A-D 在识别一个未被占用的位置后识别的位置。

在步骤 710 中，发送项插入逻辑 600（图 6）根据重写信号 602（图 6）的状态确定重写是否被允许。如果单次周期插入被请求，并且散列表中的发送项连续被更新，例如，如果散列表 190A-D 正被用在以太网桥中，那么，重写被设置。如果重写被允许，处理过程则继续至步骤 712。如果重写不被允许，处理过程则继续至步骤 714。

在步骤 712 中，发送项插入逻辑 600（图 6），选择由任一个生成的索引 208A-D 识别的被占用的位置，在其中重写包含插入关键字 606 的发送项。

在步骤 714 中，发送项插入逻辑 600（图 6）执行对存储在由生成的索引 208A-D 识别的位置上的发送项 210 A-D 的重排，以便将发送项中的一个移动到由散列表 190 A-D 之一中的另一索引指定的位置。重排的方法将结合图 8 予以描述。在重排完成和由生成的索引 208 A-D 识别的位置中有一个未被占用的位置被提供以后，包含插入关键字 606 的发送项 210 被插入未被占用的位置。

图 8 是图 6 中所示的发送项插入逻辑 600 中的步骤流程图，用于重排在图 1B 中所示的散列表 190 A-D 中的发送项，以便将 Key_A 插入 Key_A 172A-D 的索引识别的位置。

在步骤 800 中，已经确定 Key_A 未被存储在由 Key_A 172A-D 的索引识别的位置，发送项插入逻辑 600（图 6）确定是否有任何由 Key_A

172A—D 的索引识别的位置是未被占用的。如果存在未被占用的位置，则无需重排发送项，处理过程继续至 814。如果全部位置都被占用，则请求重排，以提供由 Key_A 172A—D 的索引之一识别的未被占用的位置，在其中，存储 Key_A 并且处理过程继续至步骤 802。

5 在步骤 802 中，发送项插入逻辑 600（图 6）根据 Key_B 被存储在由 Key_A index_1 172A 和 Key_B index_1 172A 识别的位置 138，确定 Key_B 174B—D 的其他索引，其中 Key_B 可以被存储。通过在 Key_B 上执行散列函数，以生成 Key_B 174B—D 的索引，或者通过利用 Key_B 去索引递归的索引表（未示出），在 Key_B 被插入位置 138 时，Key_B
10 174B-D 的索引曾在这个递归的索引表中存储过。确定了与 Key_B 174B—D 有关的其他索引以后，处理过程继续至步骤 804。

在步骤 804 中，发送项插入逻辑 600（图 6）检查被存储在由 Key_B 174B—D 的索引识别的位置 148，158，166 上的发送项。如图 1B 中所示，Key_F 被存储在由 Key_B index_2 174B 和 Key_F index_2 182B 识别
15 的位置 148 上，Key_H 被存储在由 Key_B index_3 174C 和 Key_H index_3 186C 识别的位置 158 上，Key_I 被存储在 Key_B index_4 174D 和 Key_I index_4 188D 识别的位置 166 上。如果由 Key_B 174B—D 的其他索引识别的所有位置 148，158，166，如图 1B 所示都是在用的，处理过程则继续至步骤 806。如果发送项插入逻辑 600（图 6）发现由 Key_B 174B—D
20 的其他索引识别的所有位置 148，158，166 中的一个未被占用，处理过程则继续至步骤 816。

在步骤 806 中，发送项插入逻辑 600（图 6）根据 Key_C 被存储在位置 146 上确定 Key_C 176A，176C—D 的其他索引，其中，Key_C 可以被存储，这些位置不同于由 Key_A index_2 172B 和 Key_C index_2 176B
25 识别位置。确定了与 Key_C 有关的其他索引以后，处理过程继续至步骤 808。

在步骤 808 中，发送项插入逻辑 600（图 6）检查被存储在由 Key_C 176A，176C—D 的索引识别的位置上的发送项。如图 1B 中所示，Key_C index_1 172A 识别的位置 140 未被占用，Key_G 被存储在由 Key_C index_3
30 176C 和 Key_G index_3 184C 识别的位置 160 上，由 Key_C index_4 176D

识别的位置 168 未被占用。如果由 Key_C 176A, 176C—D 的索引识别的位置 140, 160, 168 中的任何位置为图 1B 中所示未被占用, 处理过程则继续至步骤 810。如果发送项插入逻辑 600 (图 6) 发现由 Key_C 176A, 176C—D 的索引识别的位置 140, 160, 168 没有未被占用的, 处理过程则继续至步骤 818。

在步骤 810 中, 发送项插入逻辑 600 (图 6) 将 Key_C 从由 Key_C index_2 176B 和 Key_A index_2 172B 识别的位置 146 移动到 Key_C index_1 176A 识别的位置 140。Key_A 插入由 Key_A index_2 172B 识别的位置 146。

在步骤 814 中, Key_A 被插入由 Key_A 索引 172A—D 之一识别的一个未被占用的位置。

在步骤 816 中, 发送项插入逻辑 600 (图 6) 将 Key_B 从由 Key_A index_1 172A 和 Key_B index_1 174A 识别的位置 138 移动到由 Key_B 174B—D 的索引之一识别的未被占用的位置。Key_A 被插入由 Key_A index_1 172A 和 Key_B index_1 174A 识别的位置 138。

在步骤 818 中, 发送项插入逻辑 600 (图 6) 根据 Key_D 被存储在由 Key_A index_3 172C 和 Key_D index_3 178C 识别的位置 156 上, 确定 Key_D 178A—B, 178D 的其他索引, 其中, Key_D 可以被存储。确定了 Key_D 的其他索引以后, 处理过程继续至步骤 820。

在步骤 820 中, 发送项插入逻辑 600 (图 6) 检查被存储在由 Key_D 178A—B, 178D 的其他索引识别的位置上的关键字。如图 1B 中所示, 由 Key_D index 178A—B, 178D 识别的所有位置 142, 150, 170 都是未被占用的。如果由 Key_D index 178A—B, 178D 的索引识别的任何位置 142, 150, 170 如图 1B 所示都是未被占用的, 处理过程继续至步骤 822。如果发送项插入逻辑 600 (图 6) 发现由 Key_D index 178A—B, 178D 识别的位置 142, 150, 170 没有未被占用的, 处理过程继续至步骤 824。

在步骤 822 中, 发送项插入逻辑 600 (图 6) 将 Key_D 从由 Key_A index_3 172C 识别的位置 156 移动到由 Key_D index 178A—B, 178D 识别的未被占用的位置 142, 150, 170 中的一个位置上。将 Key_A 插入由 Key_D index_3 178C 识别的位置 156。

在步骤 824 中, 发送项插入逻辑 600 (图 6) 根据 Key_E 被存储在由 Key_A index_4 172D 和 Key_E index_4 180D 的识别的位置 164 上, 确定 Key_E 180A—B 的其它索引, 其中, Key_E 可以被存储。确定了 Key_E 的其他索引以后, 处理过程继续至 826。

- 5 在步骤 826 中, 发送项插入逻辑 600 (图 6) 检查被存储在由 Key_E 180A—C 的其他索引识别的地址上的关键字。如图 1B 中所示, 由 Key_E 180A—C 的索引识别的位置 144, 152, 162 都未被占用。如果由 Key_E 180A—C 的索引识别的任何位置 144, 152, 162, 如图 1B 中所示都未被占用, 处理过程则继续至步骤 828。如果发送项插入逻辑 600 (图 6) 发现由 Key_E 10 180A—C 的索引识别的位置 144, 152, 162 没有未被占用的, 处理过程则继续至步骤 830。

- 在步骤 828 中, 发送项插入逻辑 600 (图 6) 将 Key_E 从由 Key_A index_4 172D 和 Key_E index_4 180D 识别的位置 164 移动到由 Key_E 180A—C 的其他索引识别的位置 144, 152, 162 中的一个位置上。最好是, Key_E 被插入第一个被检测到的空位, 如果 Key_E 180A—C 的索引 15 依 A—C 的次序被搜索, 那就是位置 144。Key_A 被插入由 Key_A index_4 172D 识别的位置 164。

- 在步骤 830 中, 发送项插入逻辑 600 (图 6) 通过检查被存储在由 Key_B, Key_C, Key_D 和 Key_E 共享的其他位置上的发送项, 考查所有与 Key_F, Key_G, Key_H 和 Key_I 有关的发送项, 继续搜索一个未被 20 占用的位置, 直至在 Key_A indices 172A—D 之一上的发送项卸空, 并将 Key_A 插入。

- 一般来说, 以预定的次序例如 A—D 的次序, 对由关键字的索引识别的位置进行搜索。关键字被插入第一被发现的空位。因此, 项被插入 25 由 index_1 208A 识别的位置, 随后是由 index_2 208B, index_3 208C 和 index_4 208D 识别的位置。

在散列表中重排发送项, 提供散列表的有效利用, 并增加这种可能性, 即当散列表几乎充满时, 发送项能被插入。

- 尽管本发明已参考优选实施例进行了具体地表示和描述, 熟练的技术员将会理解, 在不偏离所附权利要求包含的本发明范围的情况下, 在 30 形式和细节上的各种变化是能被做成的。

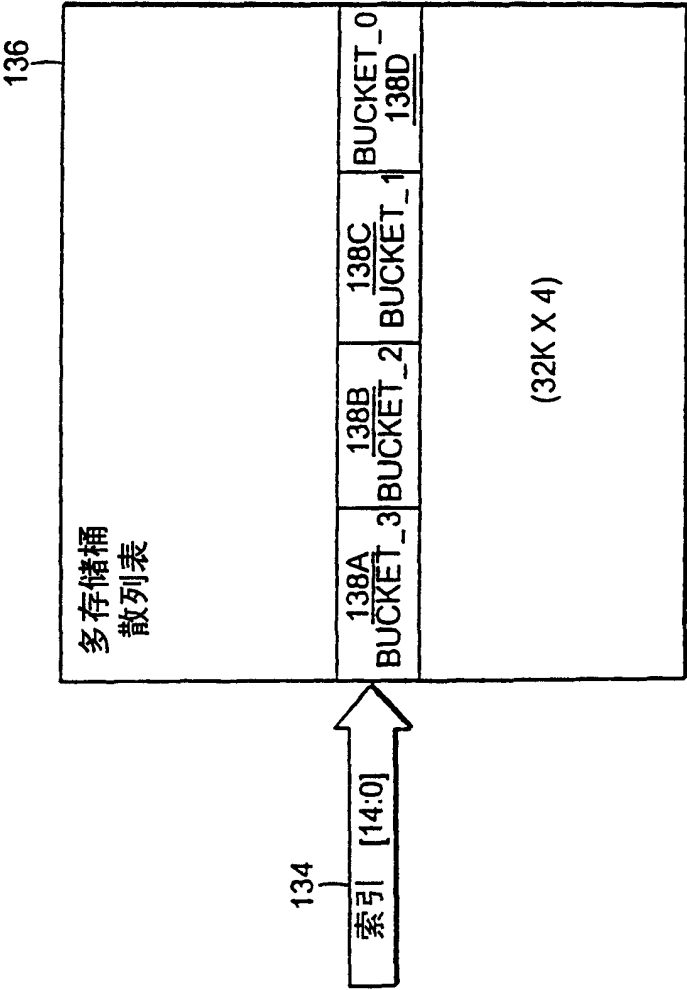


图 1A

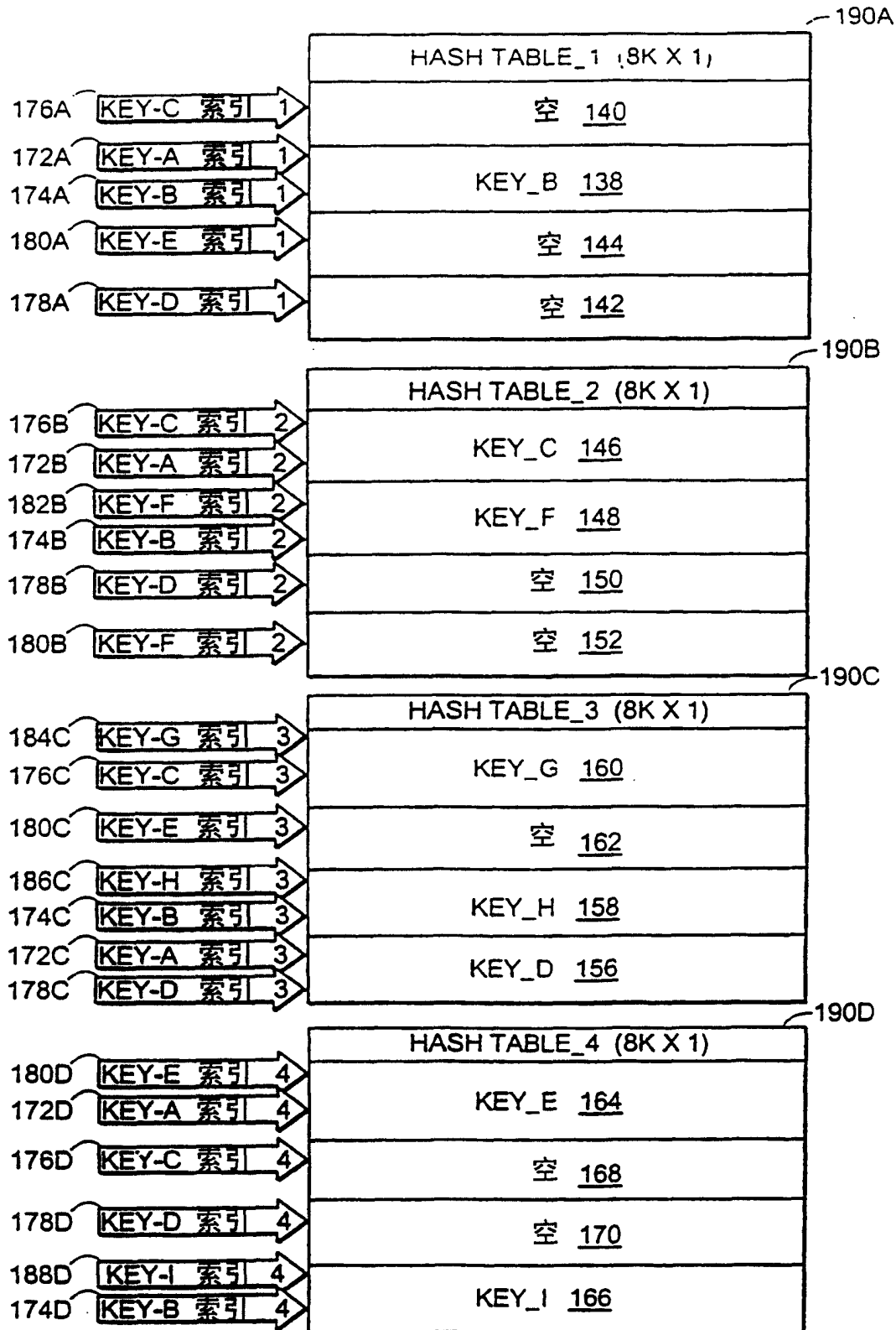


图 1B

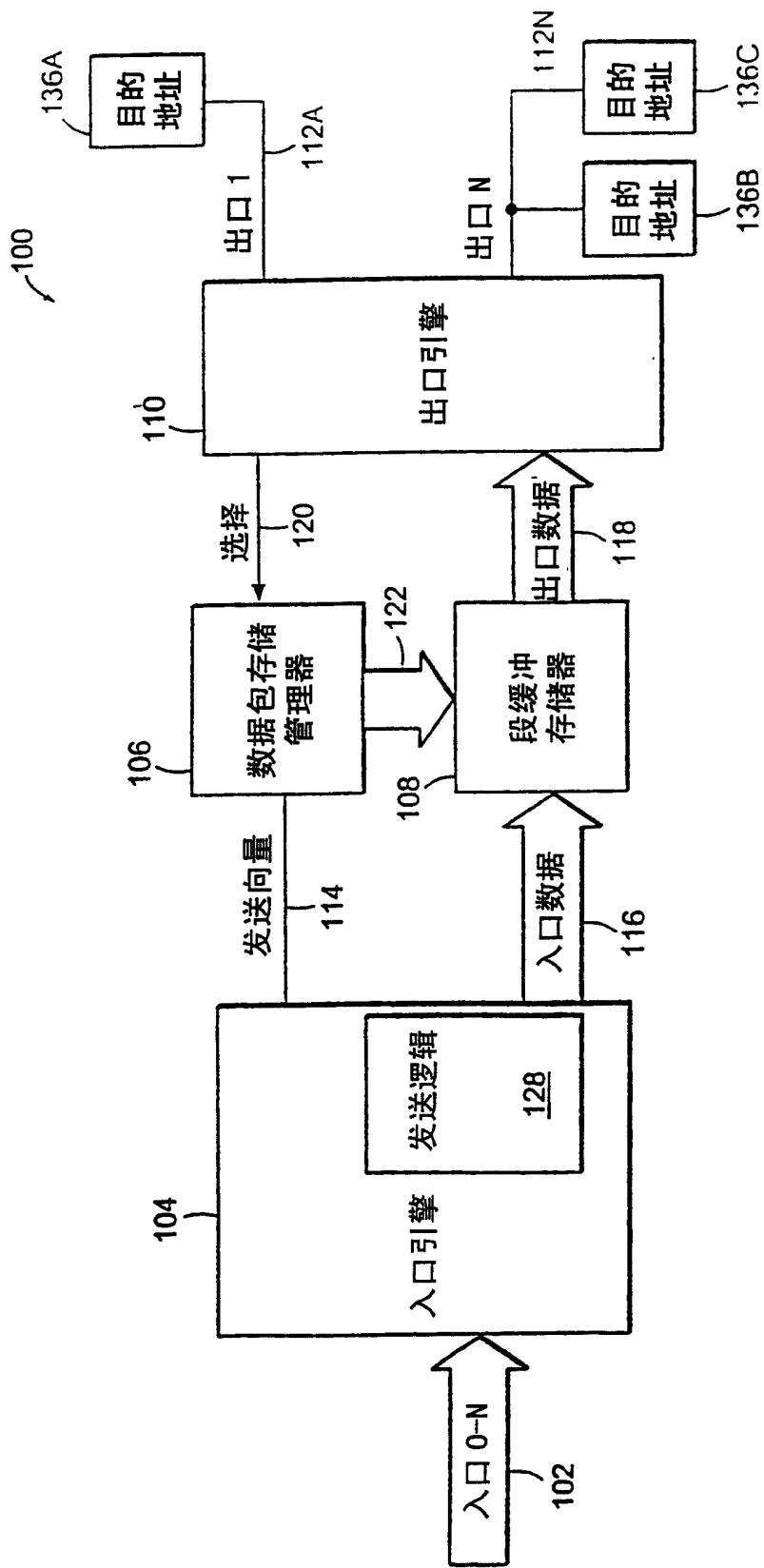


图 1C

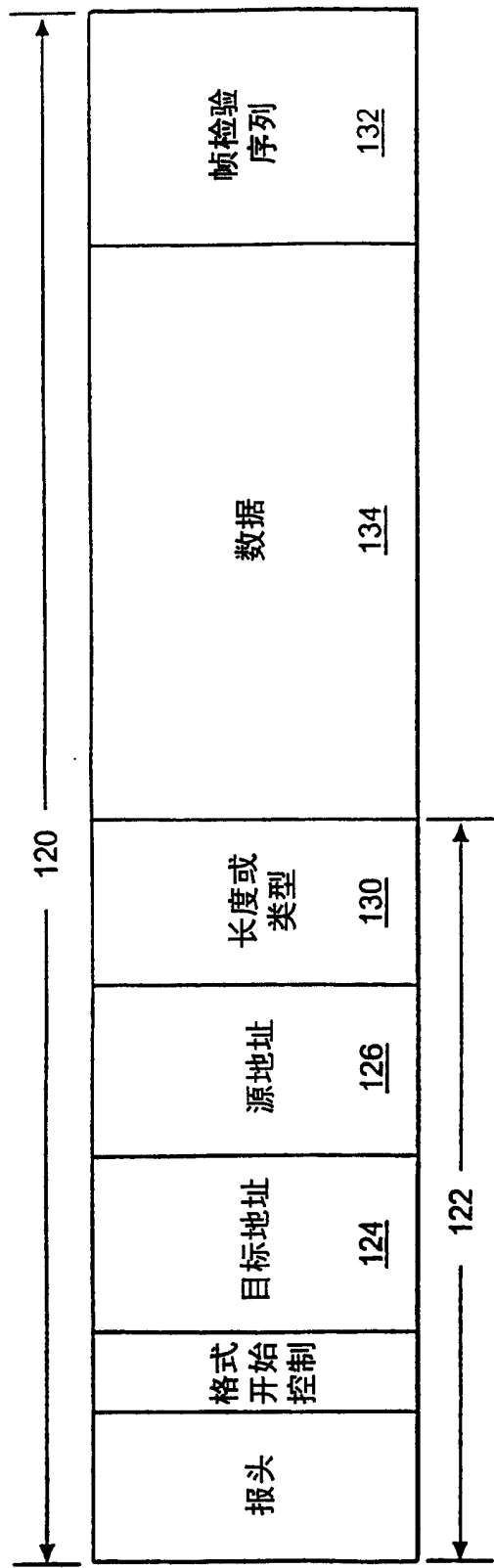


图 1D

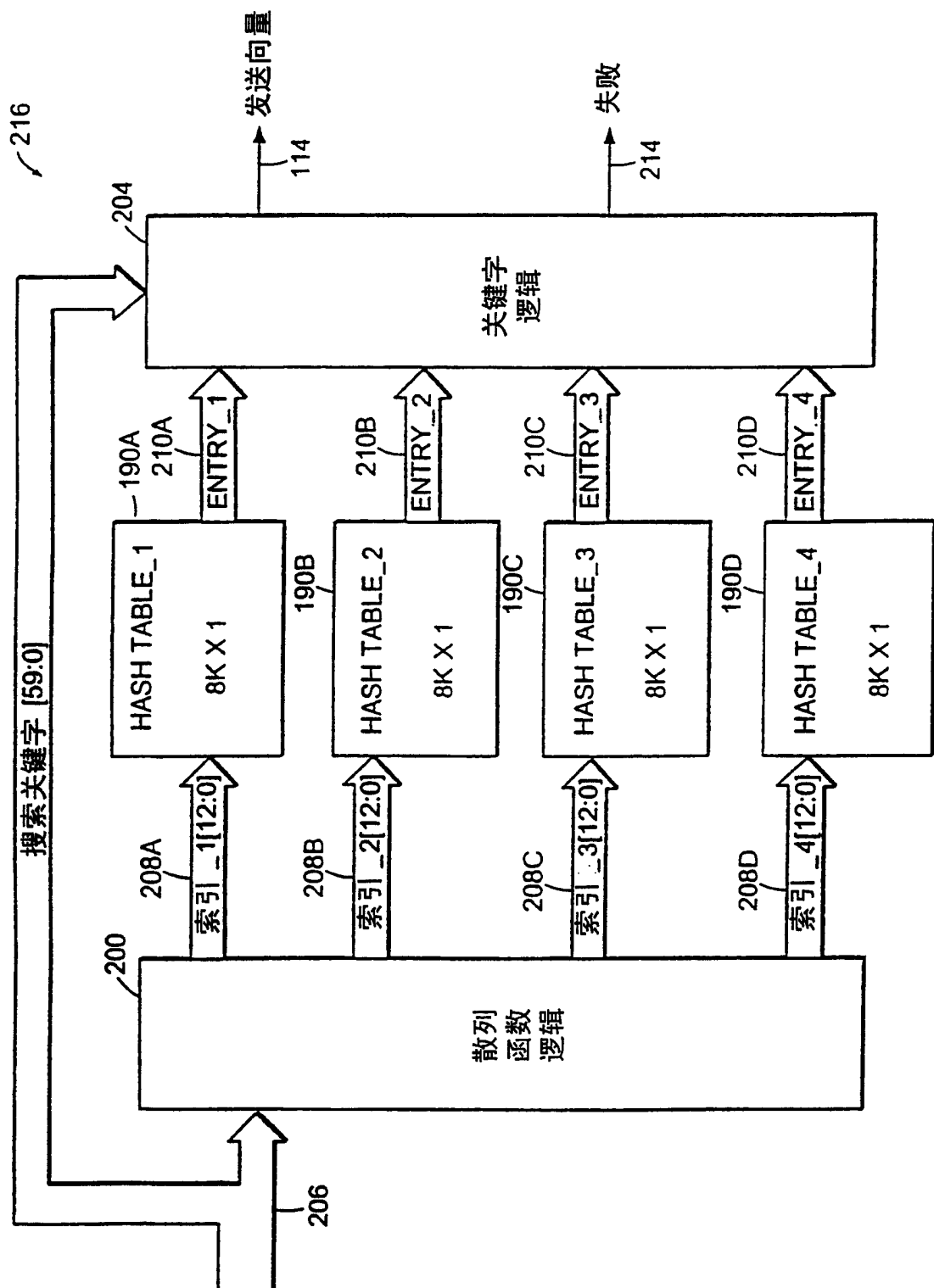


图 2

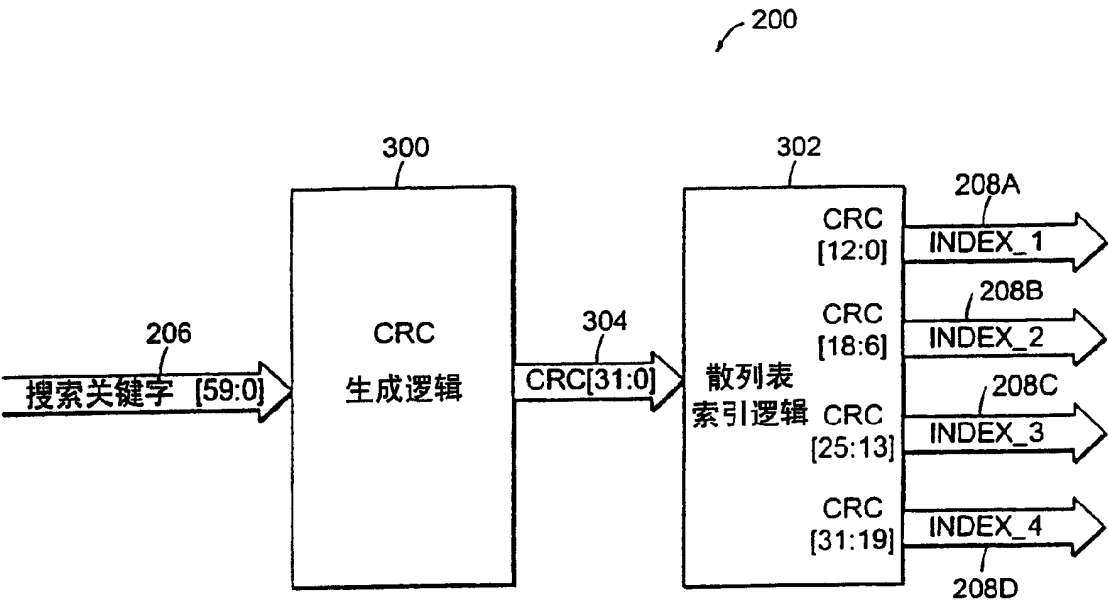


图 3

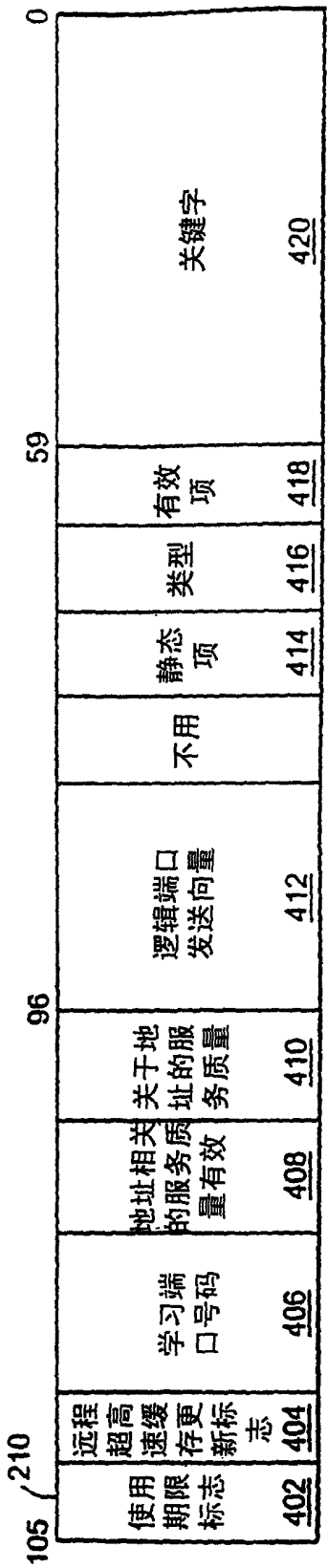


图 4

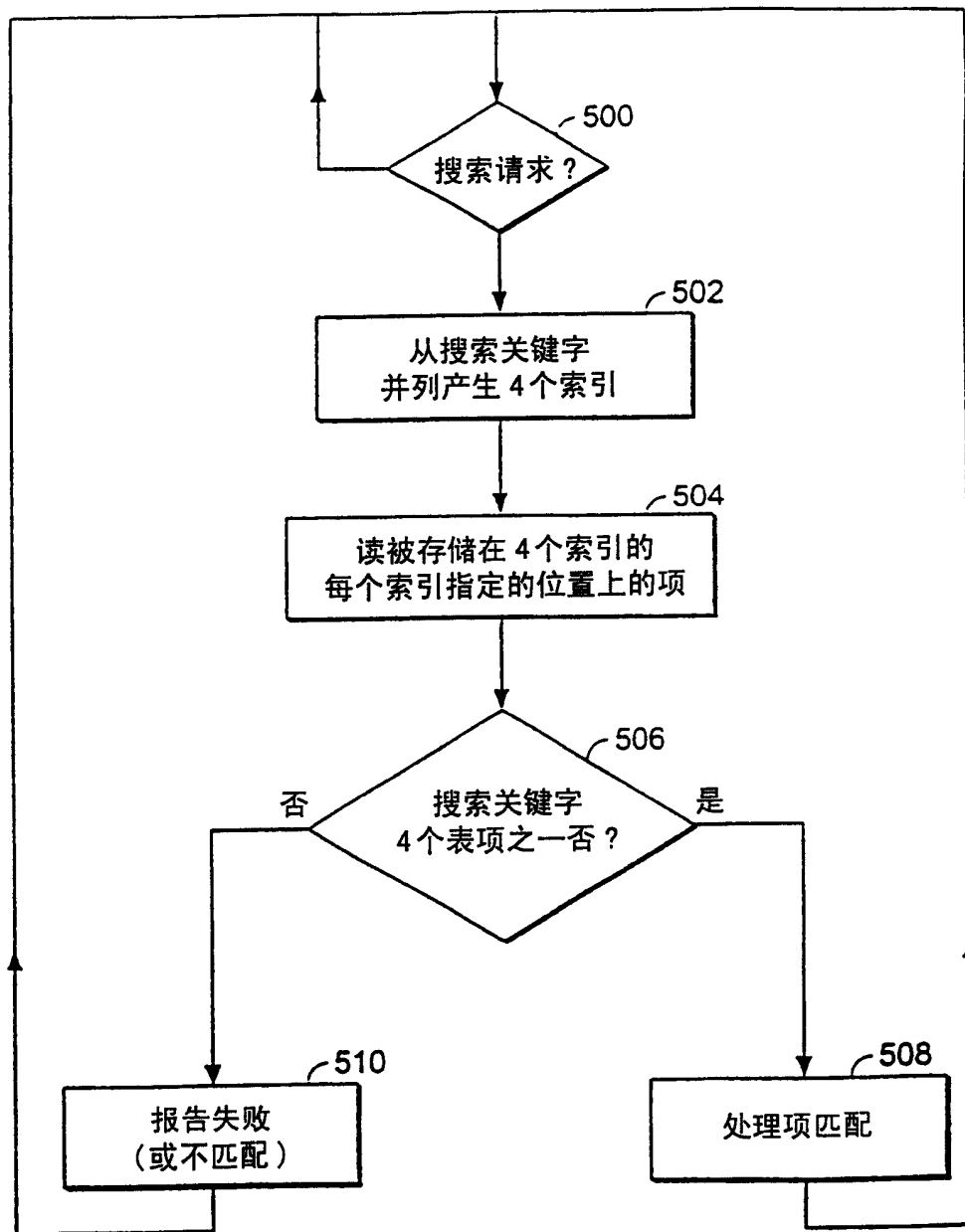


图 5

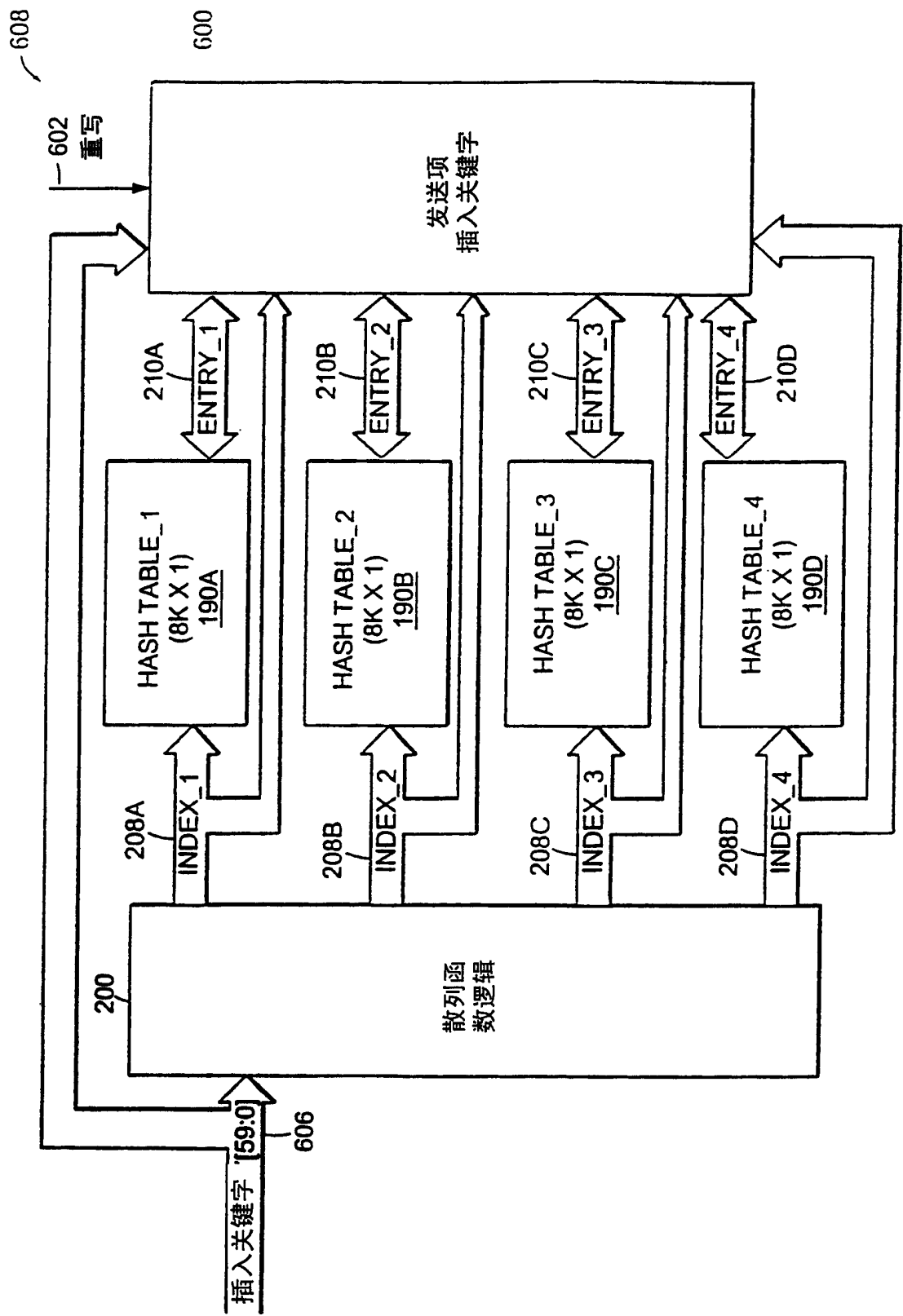


图 6

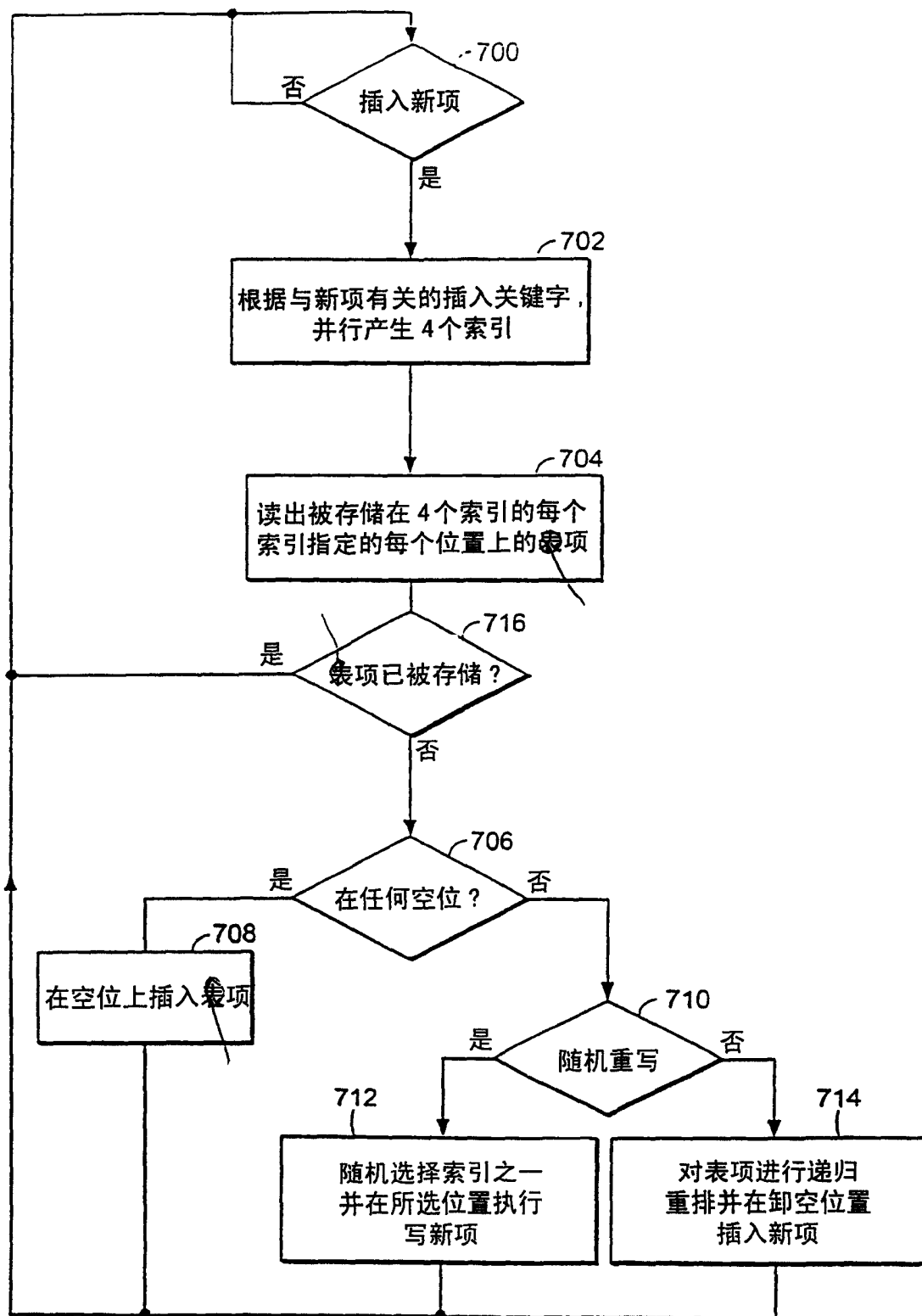


图 7

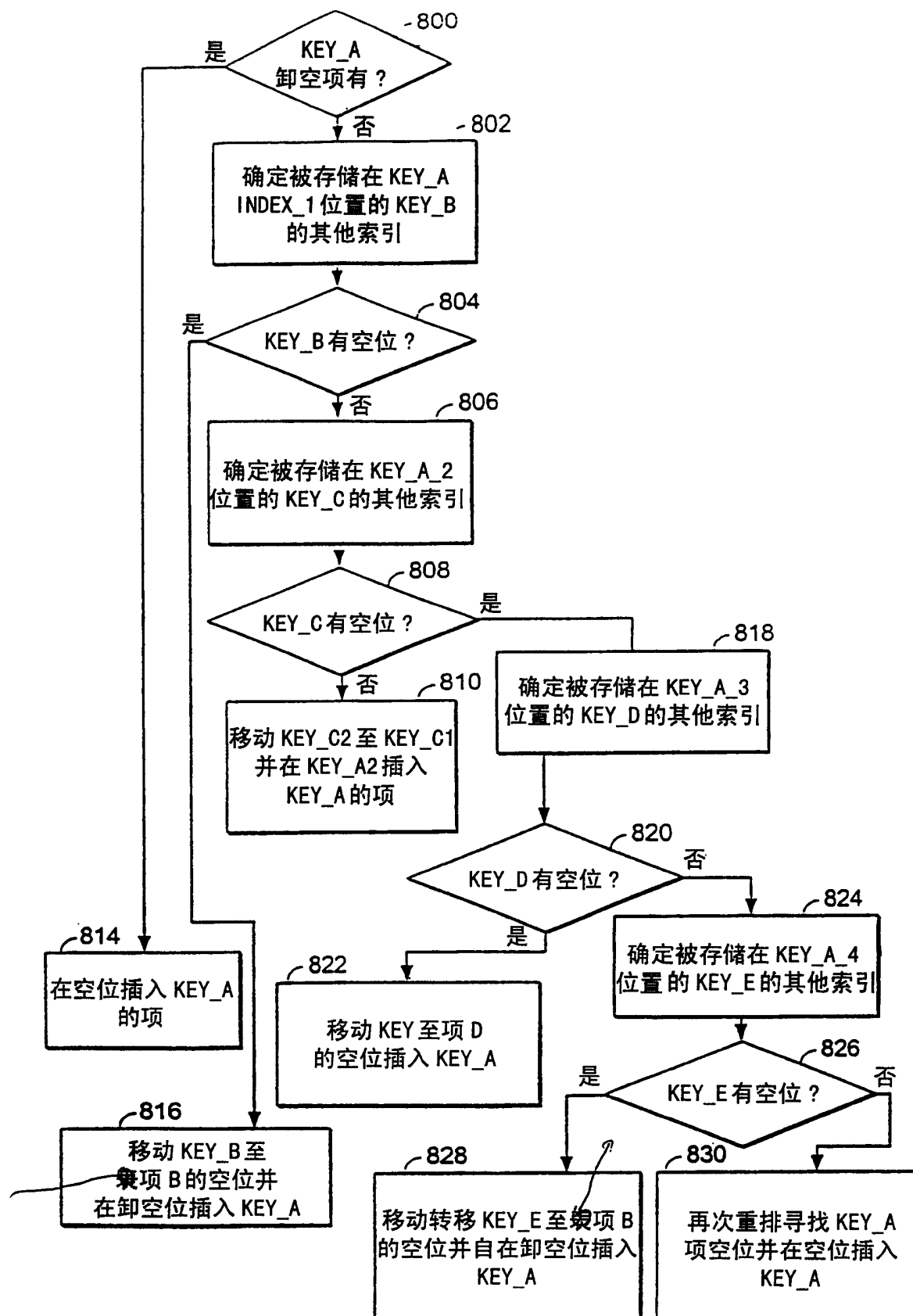


图 8