

(12) 发明专利申请

(10) 申请公布号 CN 103425610 A

(43) 申请公布日 2013. 12. 04

(21) 申请号 201310184803. 2

(22) 申请日 2013. 05. 17

(30) 优先权数据

13/475,973 2012. 05. 19 US

(71) 申请人 国际商业机器公司

地址 美国纽约

(72) 发明人 J·L·卡尔维尼亞克

C·M·德卡塞提斯 F·J·韦普兰肯

D·温德

(74) 专利代理机构 北京市中咨律师事务所

11247

代理人 张亚非 于静

(51) Int. Cl.

G06F 13/38 (2006. 01)

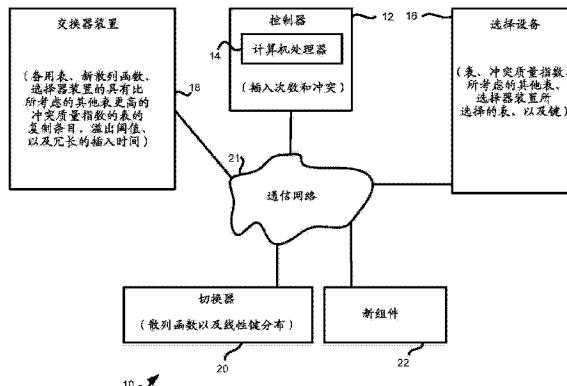
权利要求书1页 说明书6页 附图10页

(54) 发明名称

改进的计算机接口系统

(57) 摘要

一种改进的计算机系统可以包括控制器，其包含计算机处理器。该系统还可以包括与所述控制器通信的选择器装置，以由该选择器装置来选择具有比所考虑的其他表更高的冲突质量指数的表。该系统还可以包括交换器装置，以配置备用表，该备用表替换选择器装置所选择的表。该系统还可以包括切换器，其基于交换器装置对选中表的替换来改变散列函数，从而使得控制器在与引入到控制器的新组件相接口时减少插入次数和 / 或冲突中的至少一个。



1. 一种系统,包括:

控制器,其包含计算机处理器;

与所述控制器通信的选择器装置,以由该选择器装置来选择具有比所考虑的其他表更高的冲突质量指数的表;

交换器装置,其配置备用表,该备用表替换选择器装置所选择的表;以及

切换器,其基于交换器装置对选中表的替换来改变散列函数,从而使得控制器在与引入到控制器的新组件相接口时减少插入次数和冲突中的至少一个。

2. 如权利要求1所述的系统,其中所述选择器装置基于所考虑的表中的键布置来确定所考虑的表的冲突质量指数。

3. 如权利要求2所述的系统,其中键的分布作为散列函数效率的指示而被监视。

4. 如权利要求1所述的系统,其中所述交换器装置通过替换选择器装置所选择的表来降低冲突质量指数。

5. 如权利要求1所述的系统,其中所述交换器装置通过以下方式中的至少一个来配置备用表:使用新的散列函数,以及由选择器装置来复制具有比所考虑的其他表更高的冲突质量指数的表中的条目。

6. 如权利要求5所述的系统,其中,当接近达到溢出阈值时,所述交换器装置完成备用表配置。

7. 如权利要求6所述的系统,其中,所述溢出阈值基于冗长的插入时间。

8. 如权利要求7所述的系统,其中,通过对已解决了规定数量的冲突的插入的次数进行计数,来计算所述插入时间。

9. 如权利要求1所述的系统,其中所述散列函数包括线性键分布。

10. 一种方法,包括:

由与控制器通信的选择器装置来选择具有比所考虑的其他表更高的冲突质量指数的表;

用交换器装置来配置备用表,以替换选择器装置所选择的表;以及

基于交换器装置对选中表的替换,用切换器来改变散列函数,从而使得控制器在与引入到控制器的新组件相接口时减少插入次数和冲突中的至少一个。

11. 如权利要求10所述的方法,还包括基于所考虑的表中的键布置,由选择器装置来确定所考虑的表的冲突质量指数。

12. 如权利要求11所述的方法,还包括将键分布作为散列函数效率的指示来监视。

13. 如权利要求10所述的方法,还包括通过替换选择器装置所选择的表,由交换器装置来降低冲突质量指数。

14. 如权利要求10所述的方法,还包括由交换器装置通过以下方式中的至少一个来配置备用表:使用新的散列函数,以及由选择器装置来复制具有比所考虑的其他表更高的冲突质量指数的表的条目。

15. 如权利要求14所述的方法,还包括在接近达到溢出阈值时,由交换器装置来完成备用表配置。

改进的计算机接口系统

技术领域

[0001] 本发明涉及计算机系统领域,且更具体地,涉及用于有效散列和选择以用于冲突避免和最小插入时间的改进的计算机系统。

背景技术

[0002] 以太网端口可提供计算机和计算机网络间的连接。对于 100Gbps 以太网端口,最小的 64 字节的包持续时间是 6.7 纳秒,其仅给出在 500Mhz 的三个时钟周期以用于逻辑执行查找。

发明内容

[0003] 根据本发明的一个实施例,一种改进的计算机系统可包括控制器,其包含计算机处理器。该系统还可以包括与控制器通信的选择器装置,以由该选择器装置来选择具有比所考虑的其他表更高的冲突质量指数的表。该系统还可以包括交换器装置来配置备用表,该备用表替换选择器装置所选择的表。该系统还可以包括切换器,其基于交换器装置对选中表的替换来改变散列函数,从而使得控制器在与引入到控制器的新组件相接口时减少插入次数和 / 或冲突。

[0004] 选择器装置可以基于所考虑的表中的键布置来确定所考虑的表的冲突质量指数。键的分布可以作为散列函数效率的指示而被监视。

[0005] 交换器装置可以通过替换选择器装置所选择的表来降低冲突质量指数。交换器装置可以通过以下方式来配置备用表:使用新的散列函数,以及 / 或由选择器装置来复制具有比所考虑的其他表更高的冲突质量指数的表中的条目。

[0006] 当接近达到溢出阈值时,交换器装置可以完成备用表配置。溢出阈值可以基于冗长的插入时间。.

[0007] 可以通过对已解决了规定数量的冲突的插入的次数进行计数,来计算插入时间。散列函数可以包括线性键分布。

[0008] 本发明的另一方面是一种用于改进计算机系统的方法。该方法可以包括由与控制器通信的选择器装置来选择具有比所考虑的其他表更高的冲突质量指数的表。该方法还可以包括用交换器装置来配置备用表以替换选择器装置所选择的表。该方法还可以包括基于交换器装置对选中表的替换、用切换器来改变散列函数,从而使得控制器在与引入到控制器的新组件相接口时减少插入次数和 / 或冲突。

[0009] 该方法还可以包括基于所考虑的表中的键布置、由选择器装置来确定所考虑的表的冲突质量指数。该方法还可以包括将键分布作为散列函数效率的指示来监视。

[0010] 该方法还可以包括通过替换选择器装置所选择的表、由交换器装置来降低冲突质量指数。该方法还可以包括由交换器装置通过下列方式来配置备用表:使用新的散列函数,以及 / 或由选择器装置来复制具有比所考虑的其他表更高的冲突质量指数的表的条目。该方法还可以包括在接近达到溢出阈值时,由交换器装置来完成备用表配置。

[0011] 本发明的另一方面是一种与有形介质耦合以改进计算机系统的计算机可读程序代码。计算机可读程序代码可被配置为使得程序由与控制器通信的选择器装置来选择具有比所考虑的其他表更高的冲突质量指数的表。计算机可读程序代码还可以用交换器装置来配置备用表以替换选择器装置所选择的表。计算机可读程序代码还可以提供基于交换器装置对选中表的替换、用切换器来改变散列函数，从而使得控制器在与引入到控制器的新组件相接口时减少插入次数和 / 或冲突。

[0012] 计算机可读程序代码还可以基于所考虑的表中的键布置、由选择器装置来确定所考虑的表的冲突质量指数。计算机可读程序代码还可以将键分布作为散列函数效率的指示来监视。

[0013] 计算机可读程序代码还可以通过替换选择器装置所选择的表、由交换器装置来降低冲突质量指数。计算机可读程序代码还可以由交换器装置通过下列方式来配置备用表：使用新的散列函数，以及 / 或由选择器装置来复制具有比所考虑的其他表更高的冲突质量指数的表的条目。计算机可读程序代码还可以在接近达到溢出阈值时，由交换器装置来完成备用表配置。

附图说明

- [0014] 图 1 是示出根据本发明的数据库改进系统的框图。
- [0015] 图 2 是示出根据本发明的方法方面的流程图。
- [0016] 图 3 是示出根据图 2 的方法的方法方面的流程图。
- [0017] 图 4 是示出根据图 3 的方法的方法方面的流程图。
- [0018] 图 5 是示出根据图 2 的方法的方法方面的流程图。
- [0019] 图 6 是示出根据图 2 的方法的方法方面的流程图。
- [0020] 图 7 是示出根据图 6 的方法的方法方面的流程图。
- [0021] 图 8 是示出根据本发明的插入阈值的图。
- [0022] 图 9 是示出根据本发明的示例性键位(key bit)重新排序过程的一系列图。
- [0023] 图 10 和 11 是示出根据本发明的使用线性键分布的散列器(hasher)的图。
- [0024] 图 12 是示出根据本发明的具有位重新排序的重新配置。
- [0025] 图 13 和 14 是示出根据本发明的具有与线性键分布类似的性能的重新配置的散列器的图。

具体实施方式

[0026] 下面将参考附图来更完整地描述本发明，附图中示出了本发明的优选实施例。在全文中相同的标号表示相同的元件，并且具有字母下标的相同标号被用于表示单个实施例中的类似部件。

[0027] 现在参考图 1，开始描述了改进的计算机系统 10。在实施例中，系统 10 包括控制器 12，其包含计算机处理器 14。系统 10 还包括选择器装置 16，其与控制器 12 通信以选择具有比该选择器装置所考虑的其他表更高的冲突质量指数的表。系统 10 还包括交换器装置 18 来配置备用表，该备用表替换选择器装置 16 所选择的表。系统 10 还包括切换器 20，其基于交换器装置 18 对选中表的替换来改变散列函数，从而使得控制器 12 在与引入到控

制器的新组件 22 相接口(interface)时减少插入次数和 / 或冲突。如本领域技术人员所理解,系统 10 的组件通过通信网络 21 连接。

[0028] 在一个实施例中,选择器装置 16 基于所考虑的表中的键的布置来确定所考虑的表的冲突质量指数。在另一实施例中,键的分布可以作为散列函数效率的指示而被监视。

[0029] 在一个实施例中,交换器装置 18 通过替换选择器装置 16 所选择的表来降低冲突质量指数。在另一实施例中,交换器装置 18 通过以下方式来配置备用表:使用新的散列函数,以及 / 或由选择器装置复制具有比所考虑的其他表更高的冲突质量指数的表中的条目。

[0030] 在一个实施例中,当接近达到溢出阈值时,交换器装置 18 完成备用表配置。在另一个实施例中,溢出阈值基于冗长的插入时间。

[0031] 在一个实施例中,通过对已解决了规定数量的冲突的插入次数进行计数,来计算插入时间。在另一实施例中,散列函数包括线性键分布

[0032] 另一方面是一种改进计算机系统的方法,现在将参考图 2 的流程图 24 来描述该方法。该方法开始于框 26,并且在框 28 可以包括由与控制器通信的选择器装置来选择具有比所考虑的其他表更高的冲突质量指数的表。该方法在框 30 还可以包括用交换器装置来配置备用表以替换选择器装置所选择的表。该方法在框 32 还可以包括基于交换器装置对选中表的替换、用切换器来改变散列函数,从而使得控制器在与引入到控制器的新组件相接口时减少插入次数和 / 或冲突。该方法在框 34 结束。

[0033] 在现在将参考图 3 的流程图 36 来描述的另一方法实施例中,该方法开始于框 38。该方法可以包括框 28、30 和 32 处的图 2 的步骤。该方法在框 40 还可以包括基于所考虑的表中的键布置由选择器装置来确定所考虑的表的冲突质量指数。该方法在框 42 结束。

[0034] 在现在将参考图 4 的流程图 44 来描述的另一方法实施例中,该方法开始于框 46。该方法可以包括框 28、30、32 和 40 的图 3 的步骤。该方法在框 48 还可以包括将键分布作为散列函数效率的指示来监视。该方法在框 50 结束。

[0035] 在现在将参考图 5 的流程图 52 来描述的另一方法实施例中,该方法开始于框 54。该方法可以包括框 28、30 和 32 处的图 2 的步骤。该方法在框 56 还可以包括通过替换选择器装置所选择的表、由交换器装置来降低冲突质量指数。该方法在框 58 结束。

[0036] 在现在将参考图 6 的流程图 62 来描述的另一方法实施例中,该方法开始于框 62。该方法可以包括框 28、30 和 32 处的图 2 的步骤。该方法在框 64 还可以包括由交换器装置通过下列方式来配置备用表:使用新的散列函数,以及 / 或由选择器装置来复制具有比所考虑的其他表更高的冲突质量指数的表的条目。该方法在框 66 结束。

[0037] 在现在将参考图 7 的流程图 68 来描述的另一方法实施例中,该方法开始于框 70。该方法可以包括框 28、30、32 和 64 处的图 6 的步骤。该方法在框 72 还可以包括在接近达到溢出阈值时,由交换器装置来完成备用表配置。该方法在框 74 结束。

[0038] 本发明的另一方面是一种与有形介质耦合以改进计算机系统 10 的计算机可读程序代码。计算机可读程序代码可被配置为使得程序由与控制器 12 通信的选择器装置 16 来选择具有比所考虑的其他表更高的冲突质量指数的表。计算机可读程序代码还可以用交换器装置 18 来配置备用表以替换选择器装置 16 所选择的表。计算机可读程序代码还可以提供基于交换器装置 18 对选中表的替换、通过切换器 20 来改变散列函数,从而使得控制器在

与引入到控制器的新组件 22 相接口时减少插入次数和 / 或冲突。

[0039] 计算机可读程序代码还可以基于所考虑的表中的键布置、由选择器装置 16 来确定所考虑的表的冲突质量指数。计算机可读程序代码还可以将键分布作为散列函数效率的指示来监视。

[0040] 计算机可读程序代码还可以通过替换选择器装置 16 所选择的表、由交换器装置 18 来降低冲突质量指数。计算机可读程序代码还可以由交换器装置 18 通过下列方式来配置备用表：使用新的散列函数，以及 / 或由选择器装置 16 来复制具有比所考虑的其他表更高的冲突质量指数的表的条目。计算机可读程序代码还可以在接近达到溢出阈值时，由交换器装置 18 来完成备用表配置。

[0041] 鉴于以上所述，系统 10 提供了改进的计算机系统。例如，使用例如“Cuckoo 算法”的具有链(chain)的传统散列，在 100Gbps 以太网应用中使用时，可以在若干有限数量的表查找中进行搜索，但其缺陷是当非常大的表中存在很多散列冲突时具有长的插入时间。相反，系统 10 使得插入与搜索在相同的时间范围，当通过即插即用网络中的包学习来即时地 (on the fly) 进行新表插入时，这一点很重要。

[0042] 系统 10 使用高效的散列和 / 或算法选择，以用于冲突避免和减少的插入时间。系统 10 使用用于测量冲突质量指数且然后切换到将减少或最小化该指数的新表多项式的方法。

[0043] 系统 10 通过按照三个步骤基于效率较低的表的增量替换、使用具有多个散列器 / 表的查找机制以最小化冲突，来减少插入时间。首先，测量以找到效率较低的表。其次，配置最优的备用散列器并准备相应的备用表。以及第三，用备用表来即时地替换效率较低的表。

[0044] 当插入时间太长(达到溢出限制)，效率较低的表被新表替换。

[0045] 这被执行，而不用像经典 Cuckoo 算法解决方案中那样对三个其他表中的条目进行重新散列。

[0046] 插入到表是这样的，其使得可以针对每个表来计算冲突质量指数。使用新的散列器配置，在第 5 个表中复制效率较低的表中的条目(并且这可以在后台进行，以测试不同的散列器)。

[0047] 如果一个条目不能被放置于该第 5 个表中，它还没有被安装到三个其他表中(但被简单地记录以获得对新散列器的效率的概念)。当达到溢出时，进行效率较低的表中的条目到第 5 个表中的转移(如果在第 5 个表中不能这样的话，被安装在三个其他表中)。系统 10 向硬件发出原子操作以切换到新的查找表。

[0048] 系统 10 提供了冲突质量指数测量。插入到表是这样的，其使得可以针对每个表来计算冲突质量指数。

[0049] 系统 10 在插入过程中使用 N 个散列函数(N= 表的数量)来散列新的键。系统 10 进行对具有最多空条目的桶(bucket)的插入。系统 10 跟踪每个表的空桶的数量。结果，上述方法提供了所有表具有基本相同的负载。而且，当检测到溢出时，效率较低的表是具有最多空桶的表。

[0050] 另外参考图 8，在一个实施例中，当插入时间变得太长时，检测到溢出。这是通过对已解决了规定数量的冲突的插入的次数(长插入阈值)进行计数来实现的。当该计数达到规

定的阈值时,检测到溢出。此时,效率较低的表应被替换。

[0051] 系统 10 可以定义最合适的散列函数。如果过了一会儿达到溢出,这很可能是由于键分布的显著改变。

[0052] 系统 10 测量键熵(key entropy)并对此进行适当地反应。初始状态将是散列函数针对线性键分布被优化。然后,当检测到溢出时,系统 10 配置新的散列函数,例如异或矩阵,从而散列函数看到线性分布。

[0053] 另外参考图 9,系统 10 提供了键分布测量。在一个实施例中,系统 10 使用 64 个计数器(每个键位一个)加上一个全局计数器。在每次插入时,系统 10 :递增全局计数器;对于每个键位,如果 (bit=1),则递增 counter[bit];并且在每次键移除时,递减全局计数器,并且对每个键位,如果 (bit=1),则递减 counter[bit]。

[0054] 系统 10 还提供了新的算法选择。在一个实施例中,系统 10 为每个位计算切换率(toggling ratio),且如果 $\text{counter}[\text{bit}] / \text{GlobalCount} / 2$,则 $\text{counter}[\text{bit}] = (\text{GlobalCount} / 2) - (\text{Count}[\text{i}] - \text{GlobalCount} / 2)$ 。系统 10 还可以从较少切换位到最多切换位来对位重新排序,以恢复线性键分布。系统 10 还可以重新配置异或矩阵来选择位,同时考虑到重新排序。在另一实施例中,系统 10 使用每个字节一个计数器的方案,而不是每个位一个计数器的方案。

[0055] 另外参考图 10-14,其中,图 10 和 11 是异或矩阵重新配置的例子(8b 键到 2b 散列)。假设该初始散列器配置针对线性键分布被优化。

[0056] 图 12 示出了具有位重新排序的重新配置。并且,图 13 和 14 示出了重新配置的散列器具有相同的性能,就好像键分布是线性的。

[0057] 在一个实施例中,系统 10 提供了用于在硬件中执行单个散列函数以便将上亿个键中的一个映射到多个表中的一个中的槽的方法。该方法包括为所述多个表中的每个表确定冲突质量指数,并选择具有最高冲突质量指数的表。该方法还包括配置备用表并用备用表来替换具有最高冲突质量指数的表,从而降低表的冲突质量指数。该方法还包括通知切换管理元件具有最高冲突质量指数的表已被备用表替换,该切换管理元件由此将使用新的散列函数。

[0058] 在一个实施例中,系统 10 使用特定的插入算法,该算法基于尝试将相同数量的键放置于每个表中 = 多个表中的键的公平分布。在另一实施例中,系统 10 使用未被接触的桶的百分比以及 / 或表的加载率(良好喷射(spraying)),作为测量冲突质量指数的一种方法。

[0059] 在一个实施例中,如果键分布是线性的,系统 10 知道如何来选择具有良好崩塌(avalanche)且互相正交的不同的循环冗余校验(CRC)多项式。如果过了一会儿散列器性能降低,这是由于键分布的显著改变。因此,想法是测量键熵并重新排序位,从而在进入异或矩阵时它们看起来像线性分布。

[0060] 系统 10 确定较好的备用散列器。在一个实施例中,系统 10 提供了键的熵(安装的键的每个位)的连续测量。这使用每个位一个计数器以及一个全局计数器。

[0061] 对于键插入过程:递增全局加速器,对于每个键位:如果 (bit=1),则递增 counter[bit]。

[0062] 并且对于键移除过程:递减全局计数器。对于每个键位:如果 (bit=1),则递减

counter[bit]。

[0063] 系统 10 在异或矩阵之前对键位重新排序,从而 CRC 多项式产生良好的崩塌模式。

[0064] 系统 10 准备备用表。用新的散列器配置在第 5 个表中复制效率较低的表中的条目(并且这可以在后台进行,从而测试不同的散列器)。如果一个条目不能被放置于该第 5 个表中,它还没有被安装到三个其他表中(但被简单地记录在备份库中,以获得对新散列器的效率的概念)。

[0065] 系统 10 提供转接(switchover)。当达到溢出时,进行效率较低的表中的条目到第 5 个表中的转移(如果不能在第 5 个表中这样做,被安装到三个其他表中)。系统 10 向硬件发出原子操作以切换到新的查找表。

[0066] 在一个实施例中,系统 10 公开了用针对散列值的线性分布进行优化的散列函数来配置备用散列表,并用备用散列表来替换效率较低的散列表。

[0067] 应注意,在一些替换实施例中,方框中提到的功能可不按图中提到的顺序发生。例如,两个连续示出的方框实际上可以基本上以先后顺序执行,或者方框有时候可以相反的顺序被执行,这视涉及的功能而定,因为在此示出的流程图仅仅是例子。可对在此描述的这些图或步骤(或操作)进行各种修改,而不脱离本发明的精神。例如,步骤可被同时执行和 / 或以不同的顺序执行,或可添加、删除和 / 或修改步骤。所有这些变化都被认为是所要求保护的本发明的一部分。

[0068] 此处使用的术语仅是为了描述特定实施例,且不旨在限制本发明。如在此使用的,单数形式“一”、“一个”和“该”也旨在包括多数形式,除非上下文另有清楚的规定。还将理解,当在本说明中使用时,术语“包括”和 / 或“包含”明确说明存在所陈述的特点、整体、步骤、操作、元件和 / 或组件,但不排除存在或添加一个或多个其他的特点、整体、步骤、操作、元件、组件和 / 或其组。

[0069] 以下权利要求中的所有装置或步骤加功能元件的相应结构、材料、操作和等价物旨在包括用于结合在权利要求中特意阐明的元件而执行该功能的任何结构、材料或操作。本发明的说明已出于解释和描述的目的被展示,但不旨在是穷尽性的或将本发明限制在公开的形式。许多修改和变化对于本领域普通技术人员来说是明显的,且不脱离本发明的精神和范围。选择并描述实施例是为了最好地解释本发明的原理和实际应用,且使得本领域普通技术人员能针对具有适用于所打算的特定用途的各种修改的各种实施例来理解本发明。

[0070] 尽管已描述了本发明的优选实施例,应当理解本领域技术人员现在和将来都可以做出各种改善和改进,其都落在所附权利要求的范围内。这些权利要求应被理解为维护对于前面描述的本发明的适当保护。

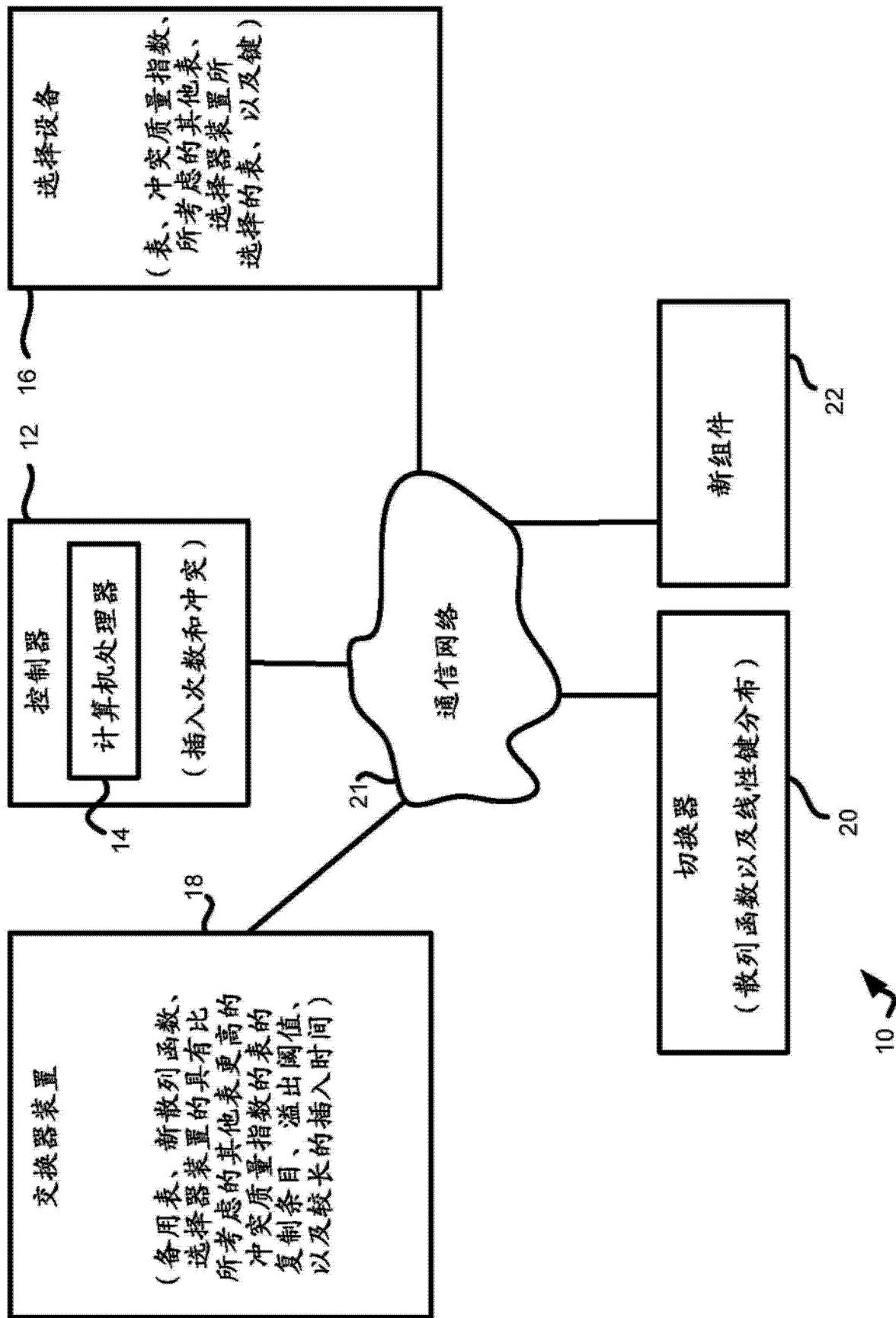


图 1

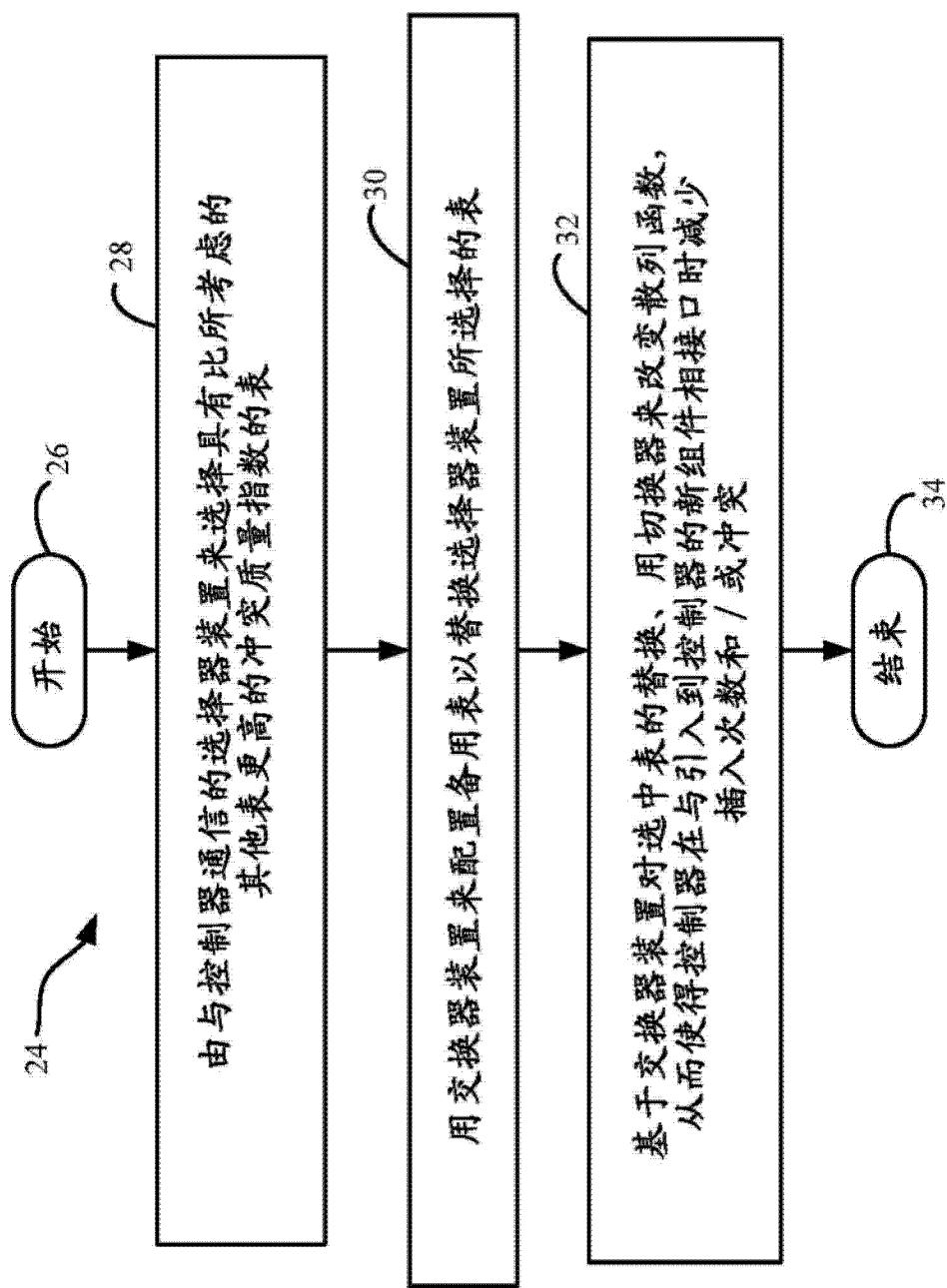


图 2

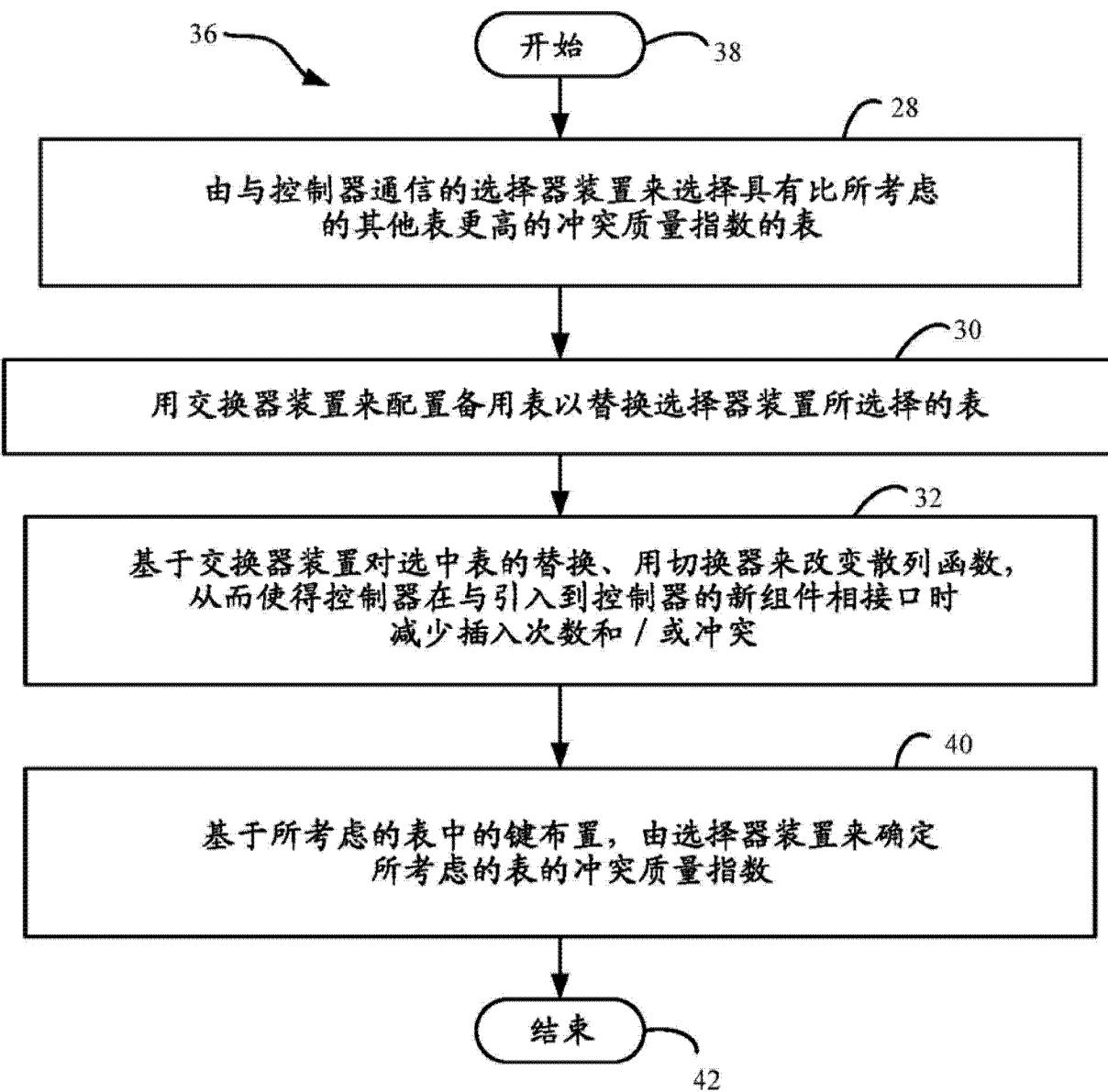


图 3

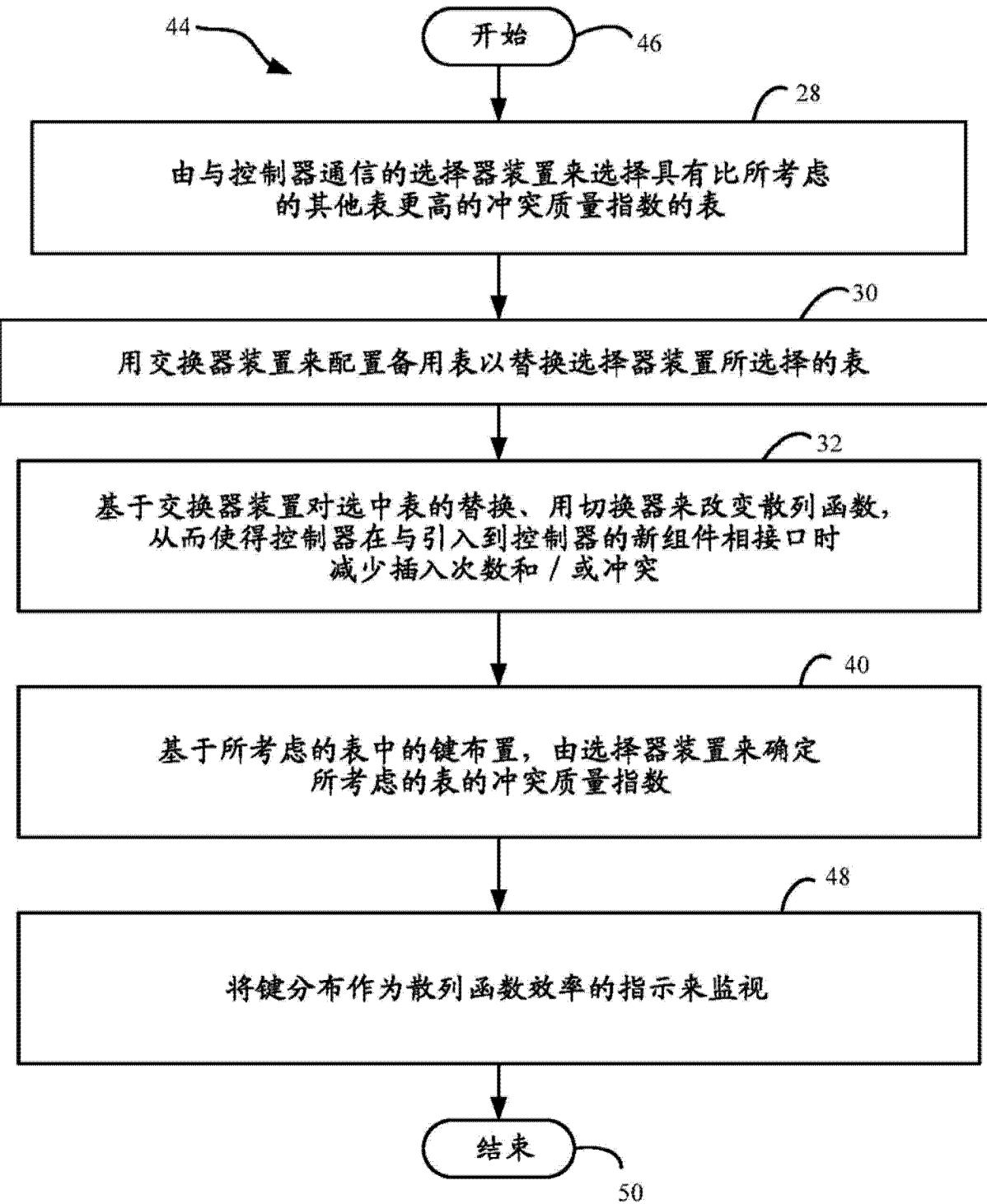


图 4

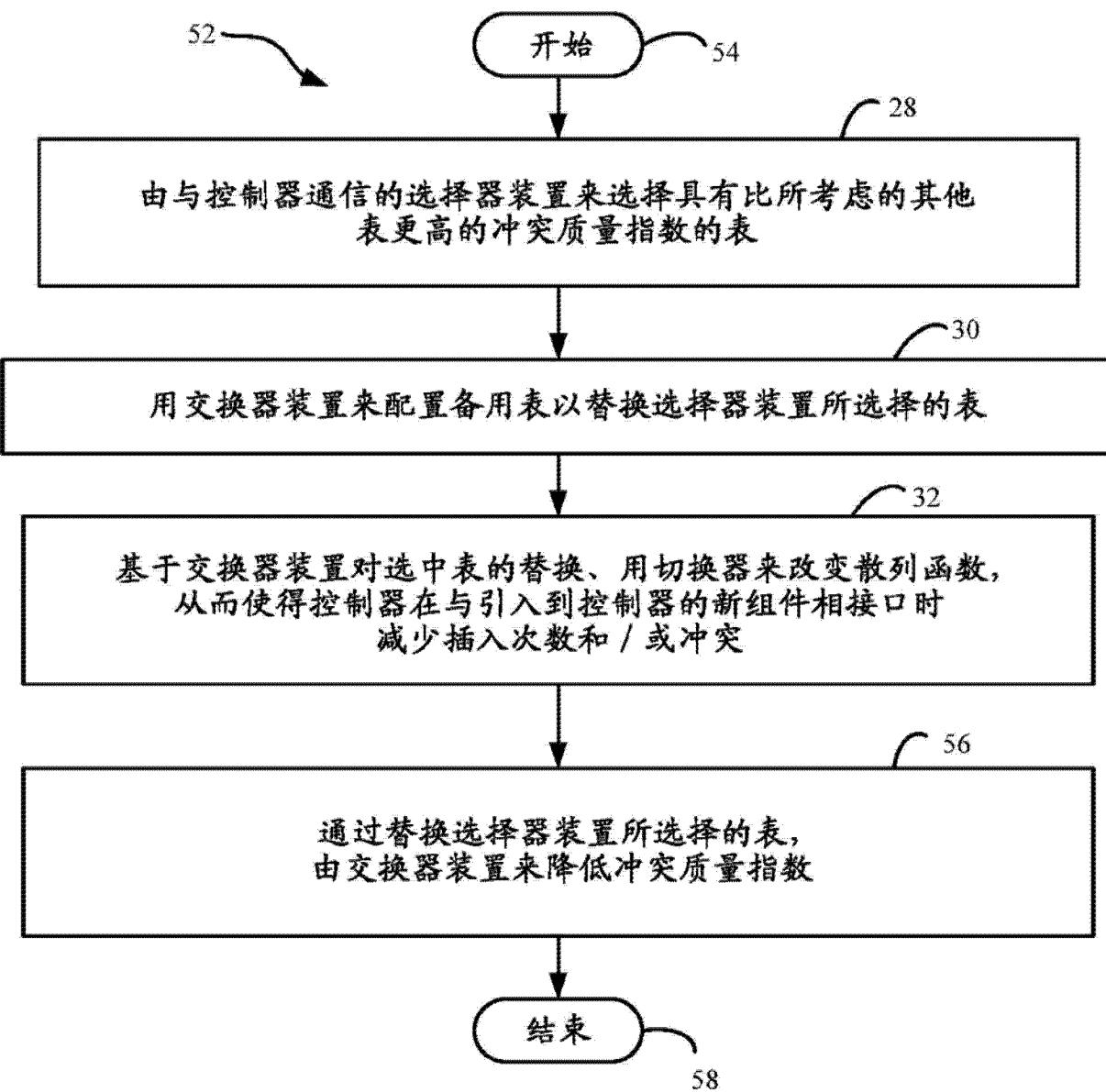


图 5

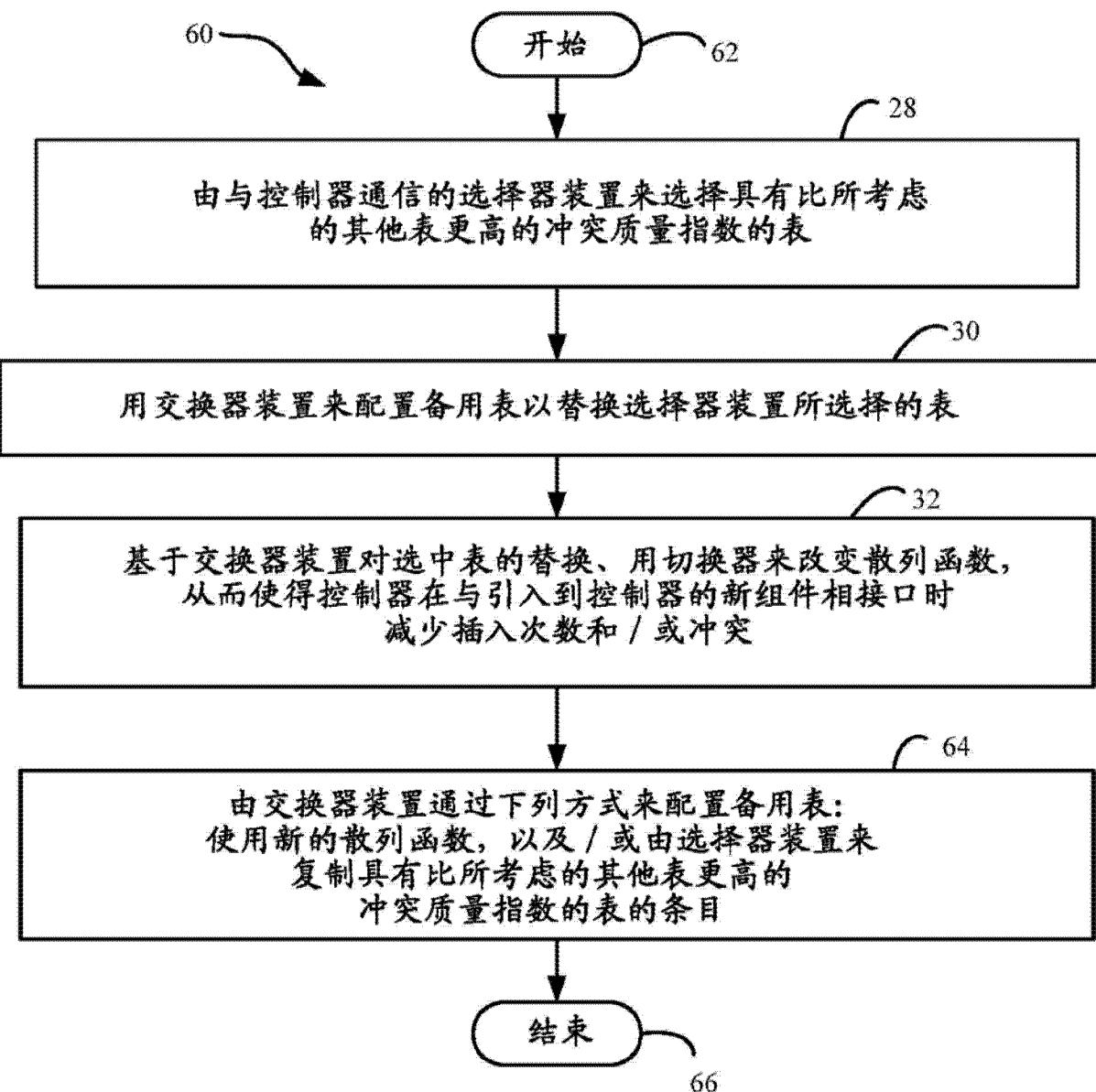


图 6

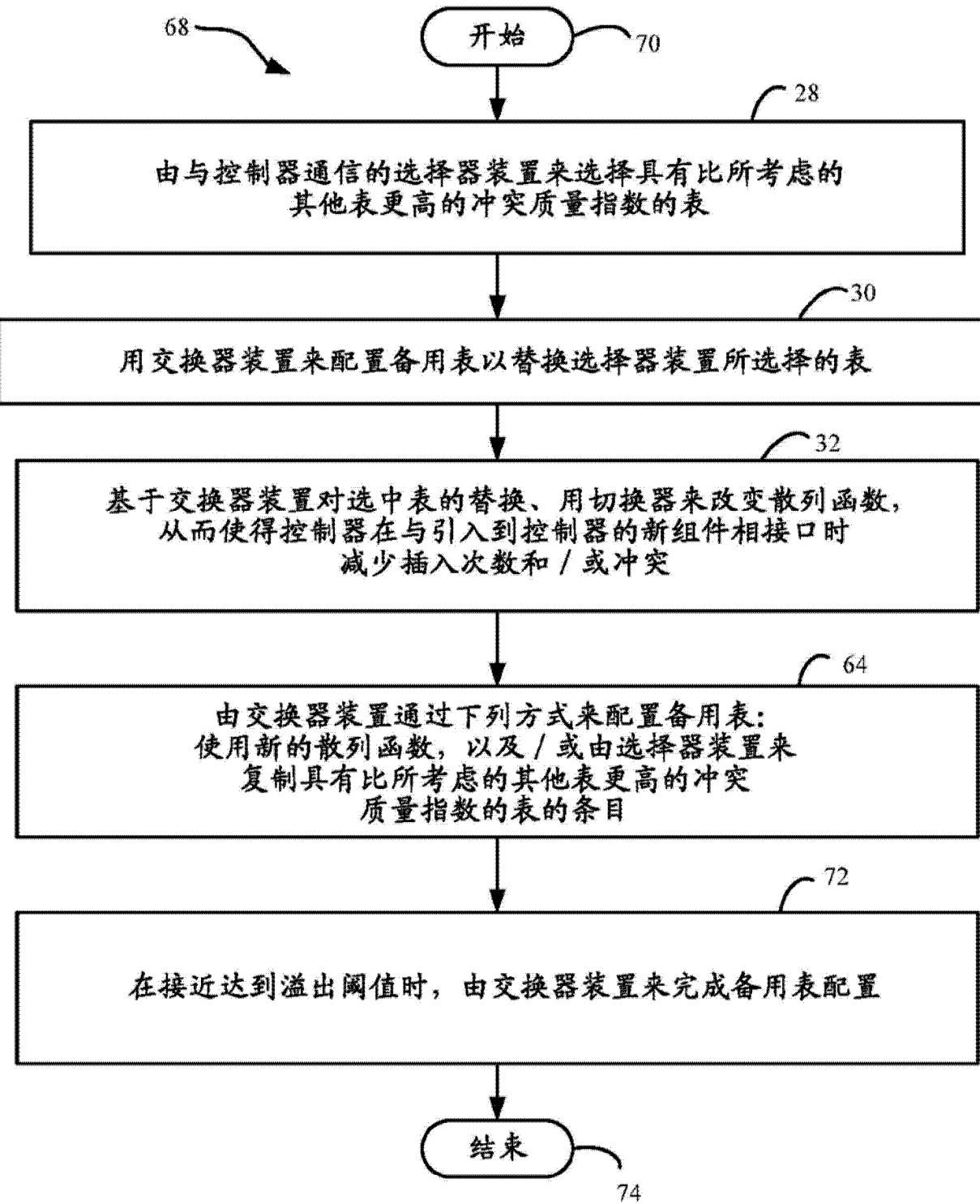


图 7

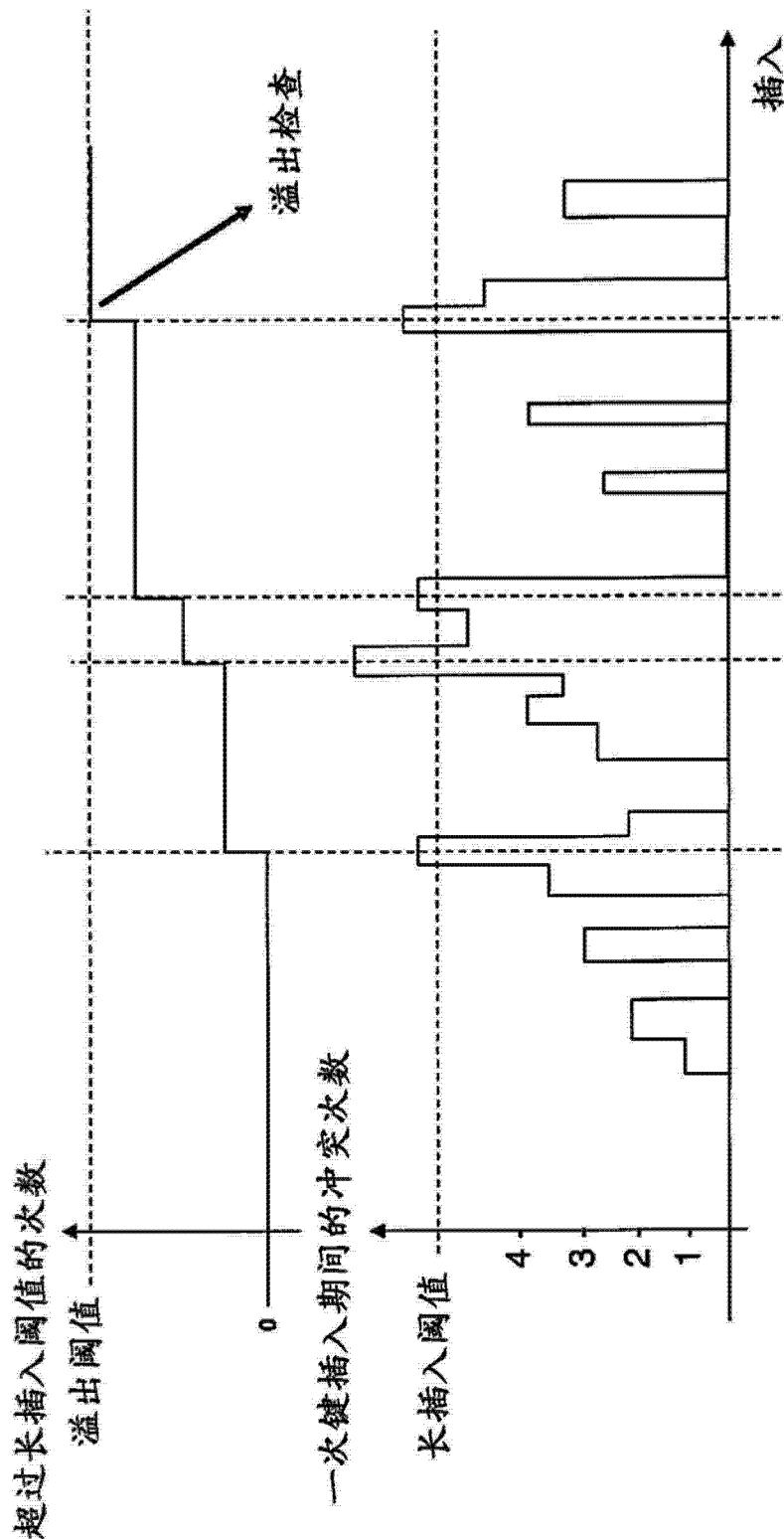


图 8

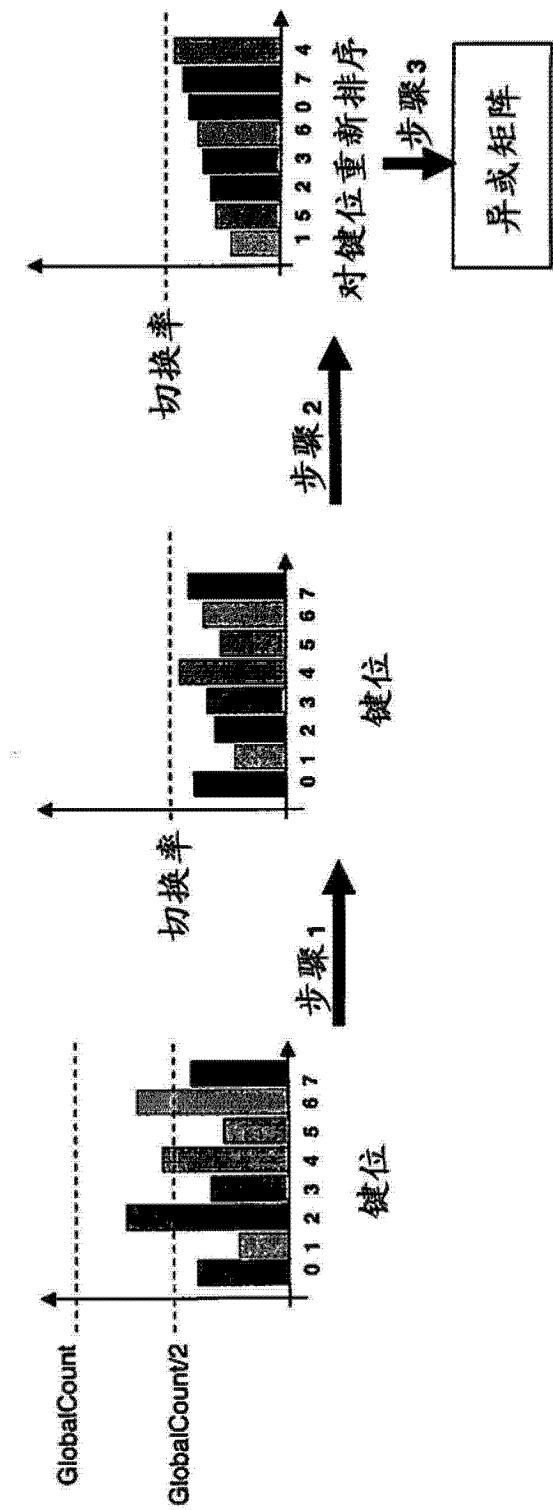
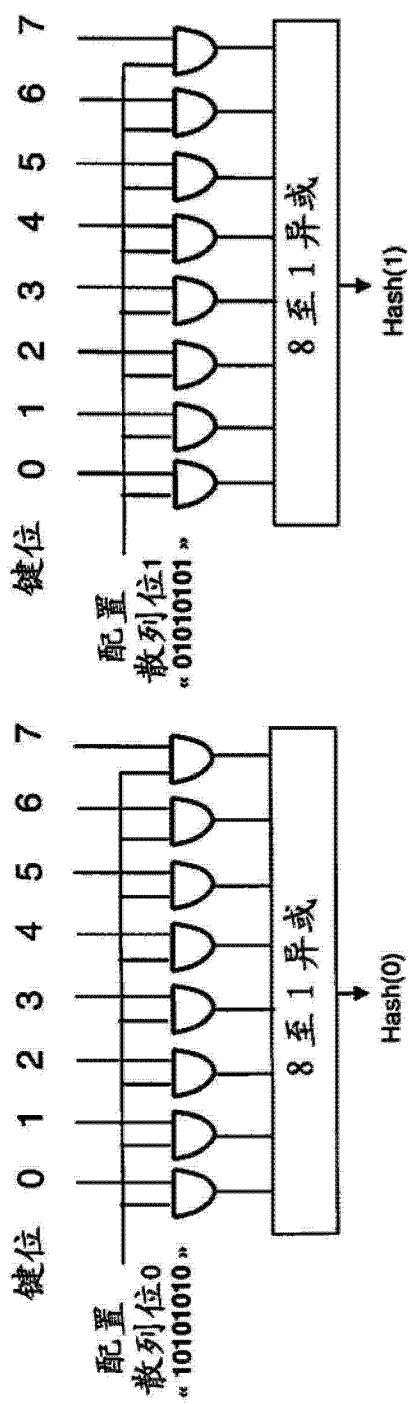
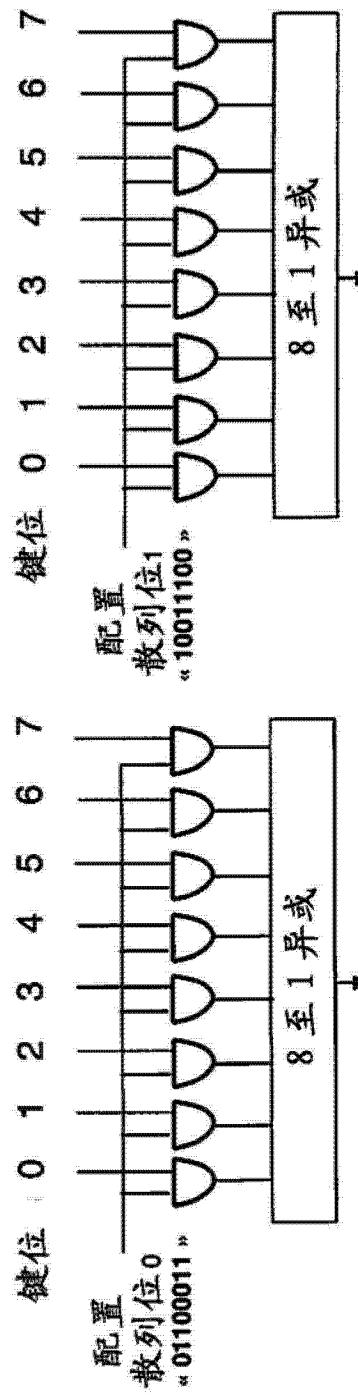


图 9

10
圖

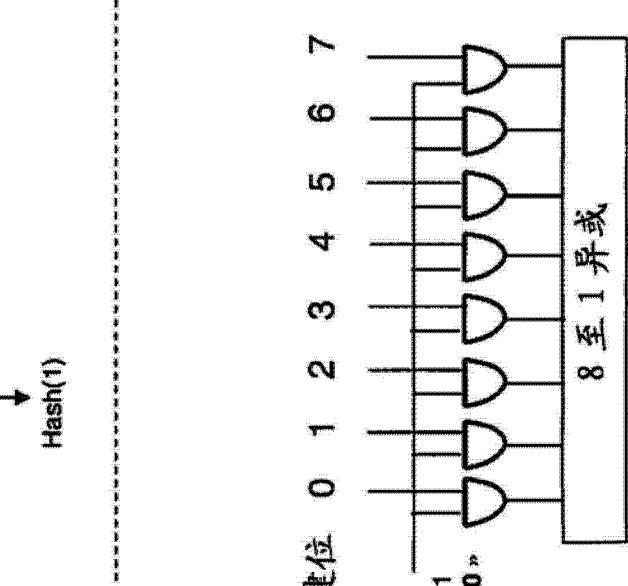


1



13

11



14