



[12] 发 明 专 利 说 明 书

[21] ZL 专利号 98801108.5

[45] 授权公告日 2005 年 6 月 15 日

[11] 授权公告号 CN 1206604C

[22] 申请日 1998.6.4 [21] 申请号 98801108.5

[30] 优先权

[32] 1997. 6. 5 [33] US [31] 08/869,913

[86] 国际申请 PCT/US1998/011680 1998.6.4

[87] 国际公布 WO1998/055929 英 1998.12.10

[85] 进入国家阶段日期 1999.4.2

[71] 专利权人 微软公司

地址 美国华盛顿

[72] 发明人 J·R·本内特

审查员 张雪梅

[74] 专利代理机构 中国专利代理(香港)有限公司

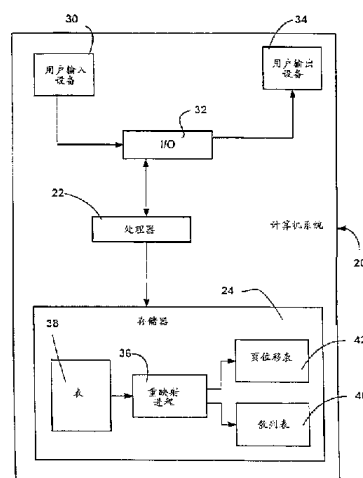
代理人 王 勇 王 岳

权利要求书 3 页 说明书 11 页 附图 11 页

[54] 发明名称 使用位移表建立一个完备的散列

[57] 摘要

变换大范围内一个不相邻数值子集诸如选择的单代码码点为一个具有完备散列的相邻的或大部分相邻的较小范围的方法和机构。该大范围数值组织为页和该页中的位移的一个二维位图矩阵(38)。如果该值在此子集内,则矩阵中的位等于1;如果不在,则为0。然后通过必要地移位每一页以避免与其它页冲突,而把这些页彼此叠放到一个一维位图(40)中。记录移位数量,并用于散列计算,其中大范围的一个值首先分割为其页号码和该页中的位移。然后通过查找该页移位数量并将该移位数量加到该页中的位移而将该值散列为密集子集范围的值。



1. 在包括一个数值范围的计算机系统中利用一个完备散列变换该范围内一个不相邻数值子集为一个相邻的或基本相邻的较小范围的方法, 包括步骤:

从该数值范围中选择数值子集;

组织该范围为一个二维矩阵, 该矩阵的一维表示页, 而另一维表示这些页中的位移, 在该矩阵中在一页和该页中的位移的每一交叉处包括映射信息, 该信息指示由该交叉表示的数值是否是在该子集内的一个所选择的数值; 以及

变换该矩阵为一个一维数组, 这是通过选择每一页、测试每一页上的映射信息与在该一维数组内的映射信息的冲突、移动每一页一定数量直到无冲突被检测到、在移位的位置上将该页中的映射信息叠加到一维数组上、以及记录该位移量而进行的。

2. 权利要求 1 的方法, 其中矩阵和一维数组是位图, 其中映射信息存储在该位的数值内。

3. 权利要求 1 的方法, 进一步包括将页排序为有序顺序的步骤, 其中选择每一页的步骤包括根据该有序顺序选择每一页的步骤。

4. 权利要求 3 的方法, 其中, 将页排序的步骤包括计数映射信息指示一页上的数值是在该子集中被选择的数值的次数, 以及从最高计数向最低计数将页排序的步骤。

5. 权利要求 3 的方法, 其中, 将页排序的步骤包括选择矩阵中的第一页作为排序顺序中的第一页的步骤。

6. 权利要求 1 的方法, 其中该数值范围的值等于包括由多个位构成的二进制数, 其中矩阵的维数相应于表示这些值的位的分割。

7. 权利要求 6 的方法, 进一步包括选择多种位分割的步骤, 及使用每一种位分割重复组织和变换, 为每一种位分割获得相应于子集数值的相邻范围的统计信息, 和根据该统计与对其它子集范围的统计的比较选择子集数值范围的步骤。

8. 权利要求 1 的方法, 其中该数值范围对应于一个信息表, 并进一步包括将表中信息写到一个散列表中对应于为选择的相应的数值所记录的页移位信息和该页中的位移的一个位置的步骤。

9. 权利要求1的方法, 进一步包括接收不相邻范围的一个值、及通过将该值分为页号码和该页中的一个位移、查找该页的移位量并把该移位量加到该页中的位移上而散列该值为一个子集范围的值的步骤。

5 1 0. 权利要求1的方法, 其中, 数值范围从0到6 5 5 3 5。

1 1. 权利要求1 0的方法, 其中数值表示可能的单代码码点, 其中数值子集表示所选择的单代码码点。

1 2. 在包括一个数值范围的计算机系统中利用一个完备散列变换该范围内一个不相邻数值子集为一个相邻的或基本相邻的较小范围
10 的设备, 包括:

从该范围中数值选择数值子集的装置;

组织该范围为一个二维矩阵的装置, 该矩阵的一维表示页, 而另一维表示对该页的位移, 在该矩阵中在一页和该页中的位移的每一交叉处包括映射信息, 该信息指示由该交叉表示的数值是否是在该子集
15 内的一个选择的数值; 以及

通过选择每一页变换该矩阵为一个一维数组的装置, 所述装置包括测试每一页上的映射信息与在该一维数组内的映射信息冲突的设备、移位每一页一定数量直到无冲突被检测到的装置、在该移位位置上将该页中的映射信息叠放到一维数组上的装置、以及记录该移位量
20 的装置。

1 3. 权利要求1 2的设备, 其中矩阵和一维数组是位图, 其中映射信息存储在该位的值内。

1 4. 权利要求1 2的设备, 进一步包括将页排序为有序顺序的装置, 其中选择每一页的装置包括根据排序顺序选择每一页的装置。

25 1 5. 权利要求1 4的设备, 其中, 将页排序的装置包括计数映射信息指示一页上的一个数值是在该子集中所选择的数值的次数的装置, 以及从最高计数向最低计数将页排序的装置。

1 6. 权利要求1 4的设备, 其中, 将页排序的装置包括选择矩阵中的第一页作为排序顺序中的第一页的装置。

30 1 7. 权利要求1 2的设备, 其中该数值范围的值等于包括多个位的二进制数, 其中矩阵的维数相应于表示这些值的位的分割。

1 8. 权利要求 1 7 的设备, 进一步包括选择多种位分割的装置, 使用每一种位分割重复地组织和变换该范围的装置, 为每一种位分割获得相应于数值子集的相邻范围的统计信息的装置, 和比较每一子集范围的统计信息以选择一个子集数值范围的装置。

5 1 9. 权利要求 1 2 的设备, 其中, 该数值范围对相应于一个信息表, 及进一步包括一个散列表, 将表中信息写到一个散列表中对应于为选择的相应的数值记录的页移位信息和该页中的位移的一个位置的装置。

10 2 0. 权利要求 1 2 的设备, 进一步包括接收不相邻范围的一个值的装置, 以及将该值散列为该子集范围一个值的装置包括把该值分为页号码和该页中的一个位移的装置、查找该页位移量的设备以及把该移位量加到该页中的位移上的装置。

 2 1. 权利要求 1 2 的设备, 其中, 该数值范围从 0 到 6 5 5 3 5。

15 2 2. 权利要求 2 1 的设备, 其中该数值表示可能的单代码码点, 其中数值子集表示选择的单代码码点。

使用位移表建立一个完备的散列

5 本发明一般来说涉及计算机系统存储器，更特别说涉及一个为存储与一个表关联的数据的改进的方法和系统。

像大小为64千字节且由16位键字索引的大条目表可能包含相对较小的非相邻条目子集，它由一个特殊的应用支持。例如，单代码(Unicode)是一个世界范围内基于16位编码单元的的字符编码标准。对于支持仅在特定语言中使用的码点(code point)的单代码配置的产品，使用该语言的码点的一个较小的、非相邻子集。作为一个例子，10 对于日语，在这种产品中通常使用日本工业标准(JIS)码，仅使用在由单代码支持的64千字节可能的符号中接近6000个符号。这6000左右符号广泛散布于整个单代码范围上的各种位置，而其余码点对这种产品没有兴趣。

15 用全部大小存储较少数目的字符属性等，64千字节表留下许多未用条目值。虽然这一存储方法允许使用单代码值作为键字的简单索引操作，从而对检索存储的信息来说是最快的方法，但是该表是属性大小的64K倍，浪费大多数空间。对许多系统和应用程序来说，快的检索速度不值为实现该方法需要的存储器数量，从而尝试其它存储方法。

20 对完全大小条目表的另一可选方案是使用对分检索。虽然在这样的对分表无浪费的位置，但是键字必需与每一条目存储，增加了存储感兴趣条目所用空间的量。此外，在执行查找时执行操作为 $O(\log(N))$ ，其中N为该表的大小，平均来说，相对其它存储方法这将提供较慢的检索。

可以使用一个散列表来存储用从码点计算的散列值索引的属性。一类散列表是带有冲突消解的散列表。这类散列表必须存储处理冲突的信息，通常由与每一条目存储键字实现。需要额外的时间来测试冲突，当25 检测到一个冲突时甚至需要更多的时间来消解。最后，这类散列表一般常常与一些额外的、不用的条目纠缠在一起，亦即该表不总是密集组合的。

30 最后，完备散列算法(无冲突)在许多应用中提供希望的性能，因为它可以提供在存储器使用和速度之间良好的平衡。此外，使用完备散列算法，键字不与条目存储在一起，这样节省空间。然而额外的、不用

的条目通常出现在完备散列中，从而浪费空间。另外，必须对计算散列值需要的成本（例如处理时间量）给以重要的考虑，因为检索时间在每次查阅操作中是一个重要的因素。此外，对一个较大范围值子集进行散列操作工作良好的对另一子集不一定好。因此，对完备散列算法存在的一个显著问题是，一般来说不存在良好的、迅速的和一致的方法为给定的值子集寻找一个完备的散列，它既非常快而又能把散列值密集组合在一起。

因此，本发明的一个总的目的是提供一种方法和机构来对一较大范围的给定不相邻值子集产生一个完备散列。

10 本发明的另一目的是提供具有上述特征的方法和机构，它把散列值密集组合在一起。

本发明的再一个目的是提供上述种类的方法和机构，它能非常快地计算散列值。

一个相关目的是提供一种以自动化方式工作的方法和机构。

15 在实现上述目的时，另一相关目的是提供一种具有上述特征的方法和机构，它对多个不相邻单代码码点（non-contiguous Unicode code point）子集工作得很好。

另一目的是提供快速、可靠、成本高效、灵活和可扩展到各种应用的上述方法和机构。

20 简而言之，本发明提供一种方法和机构来变换大范围的值不相邻子集为一个具有一个完备散列的相邻的或者大部分相邻的较小范围。值子集从该范围的值之内选择，该范围组织为一个二维矩阵。矩阵的一维是页，另一维是在该页中的位移。在每页和该页中的位移的交点，该矩阵包括映射信息，诸如指示由该交点表示的值是在该子集中选择的一个值的一位，例如如果是在该子集中选择，该位等于1，否则为0。

25 然后把该矩阵变换为一个一维数组（例如位图），这是通过选择每一页、测试每一页上的映射信息（例如一位）以发现与在一维数组中的映射信息的冲突、并移动每一页一个量、直到检测不到冲突而实现的。当检测不到冲突时，则把页映射信息覆盖在移位的位置上的一维数组上，并把移位量记录在一个页位移表中。

30

当接收到不相邻范围的一个值时，该值被分成一个页号码和其在该页中的位移，从而该值通过查阅该页的位移量并把该位移量加到该页中的位移而被散列为子集范围的值。

从下面详细的说明结合所附附图可明显看出其它的目的和优点，附图中：

图 1 是一个方框图，表示按照本发明建立一个完备散列的计算机系统；

图 2 是表示具有可能条目的示例表的方框图；

图 3 - 4 是表示以另一可选方式观察图 2 的表的选择条目的方框图；

图 5 - 6 是表示为存储关于选择条目的信息的位图矩阵的方框图；

图 7 - 10 包括一系列方框图，顺序说明根据本发明的一个方面从一个位图矩阵到一个一维结果位图的页的覆盖；

图 11 - 14 表示随着位图矩阵的页按照本发明的一个方面彼此覆盖时在一个页位移表中的位移值；

图 15 - 18 表示由条目信息正填充的一个散列表；

图 19 是一个流程图，一般表示按照本发明建立完备散列的步骤；

图 20 是一个流程图，一般表示按照本发明覆盖位图矩阵的页以产生完备散列的步骤；

图 21 是一个方框图，表示按照本发明的另一方面从键字计算一散列值的计算机系统；

图 22 是一个流程图，一般表示从一个键字计算散列值的步骤。

转向附图，首先参考图 1，图中表示一个计算机系统 20，它按照本发明使用一个位移表用于重映射非相邻数值。该计算机系统 20 包括一个与存储器 24 操作地连接的处理器 22，存储器 24 包括随机存取存储器（RAM）和非易失性存储器，诸如硬磁盘驱动器、光盘驱动器等。如所周知，非易失存储器可以与 RAM 结合使用通过公知的交换技术提供较大数量的虚存。十分明显，只读存储器（ROM）或者类似非易失性系统可以用于本发明。另外，嵌入式系统可能从本发明受益。

计算机系统 20 还包括至少一个输入设备 30，通常是一个键盘和/或一个用户指点设备例如鼠标，通过输入-输出电路（I/O）32 连接用以给处理器 22 发送用户命令。类似地，至少一个本地输出设备

3 4, 例如计算机显示监视器和扬声器连接到 I / O 3 2, 例如通过一个图形用户接口, 从处理器 2 2 给该系统 2 0 的用户传递信息。

根据本发明的一个方面, 在一个较大范围数值内给定不相邻位置的数值子集, 重映射进程 3 6 将该子集变换为一个相邻的或者大部分相邻的零基点范围。例如, 大范围数值可能对应于从 0 0 0 0 h (十六进制) 到 0 f f f f h 的索引, 以索引一个字符属性的大表 3 8, 例如该索引是单代码码点。容易理解, 需要相当大数量的空间来分配表 3 8。然而, 仅只表 3 8 中的某些条目对某个特定应用有意义, 例如, 仅相应于由一个应用程序或操作系统支持的那些字符的码点对该应用程序或操作系统有意义。在常规应用中, 重映射进程 3 6 基本取有意义的条目而重新组合它们到一个大部分密集的散列表 4 0 中, 其方式为允许原始表的键字能被接收并且被转换为对该大部分密集散列表 4 0 的一个散列值索引。作为重映射进程的一部分, 产生一个位移表 4 2, 它结合原始索引值用于计算大部分密集散列表 4 0 的散列值。位移表 4 2 在下面说明。

15 这样, 应该注意, 本发明有两个不同的方面。第一方面涉及非相邻数值子集诸如大表 3 8 的键字变换为位移表 4 2。本发明的第二方面涉及由操作系统、驱动器、应用程序等随后使用该位移表 4 2 来散列该大表 3 8 的键字为一散列值, 例如对大部分密集散列表 4 0 的一个散列值索引。通常的使用是大部分密集散列表 4 0 可以加载到实际上任何类型基于计算机的设备中以提供单代码字符属性表或类似物。在接收到相应于在大单代码表 3 8 中的一个位置处的一个单代码码点时, 计算一个对散列表 4 0 的散列值并从其中找出属性。

现在对本发明的操作进行说明。本发明对在 6 5 5 3 6 个单代码码点的范围内的日本字符子集提供实质好处。然而, 本发明不限于单代码码点子集, 而在处理较大范围的较小、不相邻子集时可能有许多潜在的应用, 这一点将十分明显。

以一简化的例子, 图 2 提供一个 1 6 个条目的表 3 8, 其中只有某些条目要被重映射到一个大部分密集散列表 4 0。更具体说, 由 0000b, 0001b, 0010b, 0110b, 1000b 和 1111b 索引的条目应用程序感兴趣。例如如图 3 原理表示 (以稍微不同的排列), 0 0 0 0 b 索引为图形显示字符 “A” 的属性, 0 0 0 1 b 索引字符 “B” 等, 直到 1 1 1 1 b 表示 “P”。其中只有 A、B、C、G、I 和 P 被支持。容易理

解, 这通常相应于从一个可能的单代码字符的 6 5 5 3 6 条目表中选择一组不相邻字符 (例如日本字符)。

为执行这一变换, 在步骤 8 0 0 - 8 0 4 选择较大表 3 8 的索引并布置在行 (页) 和列的一个位图矩阵中。图 5 表示一个为 1 6 个条目表的 4 行 X 4 列的矩阵。相应于图 3 中条目中感兴趣的条目的那些索引具有位图中一个相应位, 其设定为 1, 由图 5 位图矩阵 4 4 中的 “X” 表示, (为直观起见并为了避免与索引的 1 和 0 混淆), 而所有其它位为 0 (或空)。对于一个由 1 6 位索引的诸如 6 4 K 单代码索引的表来说, 一个初始矩阵例如由 2 5 6 行 (8 位) 和 2 5 6 列 (8 位) 组成。于是, 行 0 对任何希望的位于 0 0 0 0 h 到 0 0 F F h 之间的单代码字符其位设定为 1, 行 1 对位于范围 0 1 0 0 到 0 1 f f h 之间的希望的字符, 等等。

继续本例, 其中表 3 8 的索引布置在图 5 的位图矩阵 4 4 中, 每一行可以考虑为一页, 从 0 0 b 到 1 1 b 编号, 从而列号是从 0 0 b 到 1 1 b 的页中的位移。另一可选方案为列可以用作页, 行号表示页中的位移。确实, 当处理可能的单代码字符的 6 4 K 字符集合的子集时, 使用列作为页、行作为位移被发现能在大多数场合提供甚至更为密集的组合。然而如下所述, 可以尝试这两种方式, 选择具有最佳结果 (大部分密集的组合) 的一种。

在本例中, 如在图 6 中最佳所示, 图 5 原始的 4 X 4 矩阵 4 4 在概念上被重建为一个新矩阵 4 5, 使得图 5 中的列成为页, (图 6 中水平表示), 而行成为页中的位移 (图 6 中垂直表示)。

根据本发明的一个方面, 页一个摞一个每次一个叠放放入一个密集的或大部分密集的结果一维位图 4 6。因为所选择的索引 (其由在该位图中的 1 的位表示) 通常是不相邻的且散布范围很广, 因此在几页上的位趋向于无冲突交织, 至少是很稀少的冲突。每当表示一个索引的位确与另一位冲突时则被叠放在结果位图 4 6 上的当前页向右移动一个字符。当页无冲突完备地交织时, 位移量作为该页的一个位移记录在页位移表 4 2 中。

应该注意, 页彼此叠放的顺序影响页交织的好坏, 从而影响需要多少移动, 其最终决定散列表 4 0 的大小。因此, 寻找最佳组合的一种方式尝试所有页排序的组合, 选择最佳结果。可以理解, 这可能极为昂

贵, (例如分为 2 5 6 行和 2 5 6 列的 1 6 位索引, 可能有 2 5 6 的阶乘个组合)。

5 获得页覆盖合理顺序的一个不太贵的方式是把页这样排序, 即把其上具有较多 1 位的页在具有较少 1 位的页之前叠放放置在结果位图 4 6 中 (亦即从多到少的排序)。为此, 在步骤 8 0 6 调用一个排序进程计数各页上的 1 位, 并根据该计数将页排序。这种以 1 位加权的排序进程被发现能提供极为密集的组合。的确, 当把该排序进程与其它优化 (下面将会叙述) 结合时, 能迅速地获得百分之百 (或几近百分之百) 的组合。

10 然而, 尽管有这种通用排序技术, 但是优选选择开始散列表的第一页为在位图 4 5 矩阵中的第一页, 而不管其上的字符数目。虽然对本发明来说并非必需, 如下面显见, 选择 0 作为希望的索引并以该页作为 0 开始能保证较大表 3 8 中的 0 索引总映射为 0, 这在处理以空字符结束的字符串或类似情况十分有用。

15 图 7 表示相应于附图 2 0 的流程图的覆盖进程, 在步骤 9 0 0 开始图 6 的第一页, 页₀₀, 因为页₀₀包括 0 索引。如图 7 所示, 结果一维位图 4 6₀简单地成为该页, 而在步骤 9 1 0 把 0 位移写入页位移表 4 2, 因为在该页未发生移位。注意, 对该第一页, 步骤 9 0 0 直接继续到步骤 9 0 8, 因为假定选择 0 的话, 在该页无先导 0, 此时不可能有冲突。注意, 当该进程最终完成时, 如果选择该特殊位分割, 则任何相应于页₀₀上的每一个 1 的位的表数据将被写入散列表 4 0 (例如在步骤 20 8 1 4)。例如, “A” 属性数据将被写入由 0 索引的位置, 而 “I” 属性数据将被写入由 2 索引的位置, 如图 1 5 所示。步骤 9 1 4 测试另一页, 并因为可以获得一页, 所以返回步骤 9 0 0, 这里选择下一页, 页 25 10。

图 8 表示下一页, 页₁₀, 在先前结果位图 4 6₀上的覆盖, 这将产生一个新的结果位图 4 6₁。注意, 页₁₀是下一要被叠放的, 因为步骤 8 0 6 的排序进程决定, 该行与剩余其它行比较具有设定为 1 的最多位 (为 2)。

30 为叠放页₁₀, 在步骤 9 0 2 首先将页₁₀向左移动相等于任何先导 0 位的数量。在该位置, 在步骤 9 0 4 执行冲突测试。这样的一种冲突测试是把当前页 (页₁₀) 的位与先前结果位图 4 6₀相与, 测试 0 结果。

如果是0，则无冲突。如果非0，则检测到冲突，从而在步骤906把该页向右移动一位，重复冲突测试。继续移动该页，直到检测不到冲突为止。

在本例中，步骤904-906的冲突检测与消解进程把页₁₀在与
5 先前位图46组合之前向右移动3个0位置（图8清楚示出）。把该页的位移3记入页位移表42，其在图12中清楚示出。注意，在晚一些时候，（例如在步骤814），假定选择该位分割，矩阵数据，例如“C”和“G”属性，将加在散列表40适当位置，如图16所示。注意如图8所示，结果位图46现在长5位，而散列表40必须有至少分配5
10 个条目的空间。这样，结果位图46按需增长以容纳移位的位，散列表也一样，然而后缀0位（空位）不被考虑为结果位图46的一部分，因为仅只1表示（索引的）感兴趣的条目。注意，最终决定散列表40大小的结果位图46增长的数量依赖于对于集值的组合的密集程度。

下一个要叠放在结果位图46上的页（步骤914和900）是页
15 ₁₁。注意页₁₁和页₁₀每一个其中有一个位被设定，但是在本例中，假定步骤806的排序进程将页₁₁在页₁₀之前放置。如从图6的矩阵45和图9中看出，页₁₁具有3个先导0位（空位）。因此，在尝试叠放进程之前，在步骤902，在进行冲突测试前首先将页₁₁向左移位3次。显然，步骤904决定在该（负3）位移处存在一个冲突，从而在步骤
20 906将该页向右移动一个位置（位移是增量的），并重复冲突测试。这时在步骤904确定无冲突，在步骤908把页₁₁连同其负2的位移与先前的结果位图46组合成一个新位图46。（图9）。在步骤910，该页负2的位移记录在页位移表42中适当位置（位置3），其在图13中清楚示出。注意后来，例如在步骤814，相应于该索引的数据例如属性“P”保存在散列表40中相应于在结果位图46中的索引位置的位置，如图17所示。

剩余最后一个要叠放在结果位图46上的页是页₀₁。如从图6的矩阵45以及图10所示，页₀₁无先导0位（空位）。因此页₀₁在测试冲突前在步骤902不向左移位。显然，在步骤904在0位移处存在一个冲突，于是该页在步骤906向右移位一个位置，位移是增量的，并
30 在步骤904重复冲突测试。容易理解，在步骤904继续进行冲突检测，在步骤906移位该页，直到页₀₁位移正5个位置。其结果，具有

正 5 位移的页_{0,1}连同其正 5 位移与先前的位图 4 6₀组合成新的最终的结果位图 4 6 (图 1 0)。注意页_{0,1}移动了先前结果位图的全部大小。更特别的是,因为先前的位图 4 6₀具有百分之百组合,因此页_{0,1}在先前结果位图的末端开始叠放。注意,检测百分之百组合是可行的,而如
5 果检测到的话,立即在其末端叠放下页,而不在其开始处开始并重复移位到其末端。

在步骤 9 1 0,把为该页的位移 5 记录在页位移表 4 2,其在图 1 4 中清楚示出。现在该过程返回到图 1 9。再次注意,晚些时候在步骤 8 1 4,相应于该索引的属性数据例如属性“B”将保存在散列表 4 0
10 中相应于在结果位图 4 6 中该索引位置的位置,其在图 1 8 示出。

到此时,原始 1 6 个条目表 3 8 的 6 个散布的索引以及相应数据可以重映射到一个密集组合的散列表 4 0 和具有 4 个条目(例如字节)的页位移表中,散列表 4 0 由从 0 到 5 的数值索引。注意,即使是如此小的一个例子,存储器的节省也是相当可观的。例如,如果原始 1 6 条目
15 表 3 8 为每一索引分配 1 千字节的话,则该表需要 1 6 千字节存储器。然而,同样的数据可以存储在 6 千字节散列表加上页位移表的 4 个存储单元(例如字或字节)。的确,如下所述,只需要页位移表 4 2 将对原始大表 3 8 的一个键字变换为密集散列表 4 0 的一个散列值索引。更大数组的更大子集提供相似的节省,特别从自动重映射进程中受益。

20 虽然对本发明并非必需,但是为找出一个最佳密集组合,可以尝试各种位图矩阵,包括行列数不等的阵列。例如,对于 1 6 位索引,分为不等于 8 X 8 位也可以以上述方式用于重映射该数值。为找出最佳分割,在步骤 8 0 8,保存结果散列表 4 0 和页位移表 4 2 的大小作为与散列表 4 0 关联的统计。然后,该进程循环回到步骤 8 0 2 建立其它矩
25 阵,每一次改变数值分割为行和列的方式。例如,1 6 位索引可以分为具有 5 1 2 列的 1 2 8 页(7 X 9 位),具有 1 2 8 列的 5 1 2 页(9 X 7 位),具有 1 0 2 4 列的 6 4 页(6 X 1 0 位)等。如果某一循环重复改善前面的最佳分割,则该页位移表和相伴随的统计代替前面的一个,从而使用最小数目空间的分割在所有(合理的)分割尝试(步骤 8
30 1 2)后最终留下来。注意,上述映射操作足够快,使得可以尝试所有合理的组合,直到通过该方法识别出可用的最佳映射。此外,如上所述,一旦分割为行和列,则叠放的页既可以豎直观察,也可以水平观察,以

发现最好可能的组合。一旦发现最佳映射，就可以在步骤 8 1 4 使用其位移表 4 2 写原始表条目到一个密集的或大部分密集的散列表 4 0。

5 现在转向使用页位移表计算对散列表的一个散列值索引，为此在图 2 1 中表示另一个计算机系统。在图 2 1 中，对和图 1 中相同的部件重新编号，号码比在图 1 中高 1 0 0 0。该系统 1 0 2 0 可以例如是一个较低功能的手持式计算设备，其中的存储器 1 2 4 包括诸如为显示选择的字符诸如单代码字符的目的而完成的散列表 1 0 4 0 和页位移表 1 0 4 4 2。相应地，本发明的第二方面是处理原始的大表 3 8 的键字到一个散列值 1 0 4 4 的变换，该散列值用作对密集散列表 1 0 4 0 的一个索引。注意表 1 0 4 0 和 1 0 4 2 通常存储在非易失存储器中，例如烧制入一个 R O M 等中。

为实现散列计算，一个应用程序 1 0 4 8 或者类似程序首先加载（或者如果已经加载，则定位）页位移表 1 0 4 2 和散列表 1 0 4 0 到存储器 1 2 4（在图 2 2 的步骤 1 1 0 0）。应用程序 1 0 4 8 或类似程序收到一个键字 1 0 4 6（输入值），并将其供给散列计算进程 1 0 5 0，借此在步骤 1 1 0 4（图 2 2）首先把该值分成用于构建页位移表 1 0 4 2 的同样的位组。从页位移表 1 0 4 2 的大小可明显看出位分割，亦即页位移表 1 0 4 2 条目的数目等于页数，从而指示用以存储页位移信息所用的位数，例如 1 2 8 条目相应于 7 位。当然分割和其它信息（例如相应散列表的文件名）也可以保存在页位移表 1 0 4 2 的头标信息或类似信息中。

25 无论哪种情况，当接收到相应于原始大表 3 8 的一个输入值时，（此时竖直列成为页），该值的高位被作为步骤 1 1 0 4 的一部分的计算进程 1 0 4 8 屏蔽，留下表示页号码的适当数目的有效低位。在该简化的例子中，高位部分等于 4 个可能位中的两个，从而上两位被屏蔽。剩余的两位用作位移表 1 0 4 2 中的一个 0 基点的索引。例如，如果收到一个 0 1 1 0 b，例如为显示“G”字符，则上位被屏蔽，留下 1 0 b。类似地，在步骤 1 1 0 4 内，表示该页中的位移的上位通过向右移位非屏蔽上位页号码中同样的位数而获得。

30 在步骤 1 1 0 6，使用页号码作为在页位移表 1 0 4 2 中的一个索引，从而执行查阅操作以决定该相应页的位移值。在本例中，1 0 b 提供 3 的位移值。在步骤 1 1 0 8，该值加在该页中的位移上（等于 0 1

- b，其从在步骤 1 1 0 4 屏蔽和移动上位决定)。这一结果 (3 加 1 等于 4) 是计算的对散列表 1 0 4 0 的散列值索引，在步骤 1 1 1 0 将其返回给应用程序 1 0 4 8，应用程序 1 0 4 8 然后从散列表 1 0 4 0 获得适当数据。可以看出，在图 1 8 的散列表 4 0 中的第 4 (1 0 0 b) 5 条目恰当索引“G”字符。下表说明所有有效计算的结果，它们在这里不再详细叙述：

输入值	低位	位移值 (十进制)	高位	散列值 = 位移 + 高位 (十进制)	散列表 数据
0000b	00	0	00	0	A
0001b	01	5	00	5	B
0010b	10	3	00	3	C
0110b	10	3	01	4	G
1000b	00	0	10	2	I
1111b	11	-2	11	1	P

- 10 可以理解，该计算极快，只包括与 (为屏蔽)、移位、索引矩阵 (页位移表 1 0 4 2) 和加法。

- 应该注意，为提供正确的结果，散列计算需要一个有效输入值，亦即由使用该散列表的应用程序支持的一个有效值。可以在执行散列计算之前对该值执行有效性检验 (例如在步骤 1 1 0 2 所表示的) 以保证它 15 相应于一个有效输入。如果输入值有效，则该散列是完备的，因为任何有效键字提供一个无冲突的唯一结果。

- 此外，在散列表数据自身之外，仅只位移表需要维护以计算该散列值。因此，在构建位移表期间积累的任何其它信息，例如结果位图和统计数据可以不要。当然，如果希望的话，一些元数据和 / 或其它信息可以 20 与位移表和 / 或散列表一起保存，以便例如如果在系统中可用的散列表和相应位移表多于一个的话，可以使应用程序加载合适的散列表和相应位移表。

从前述中可看出，提供了一种方法和机构，用以对较大范围不相邻数值的给定子集产生一个完备散列。该方法和机构密集组合散列后的数值到一起，同时能对该散列值进行非常快的计算。该方法和机构以自动方式操作，而对不相邻单代码码点子集作用特别好。该方法和机构快速、可靠、成本高效、灵活和可扩展。

虽然本发明易于进行修改和具有另外可选择的结构，其中某些实施例已在附图中显示，并在上面详细叙述。然而应该理解，并不打算限制本发明在公开的具体形式，而是相反，本发明意在覆盖所有修改、其它可选择的结构以及落入本发明的精神和范围之内所有等价物。

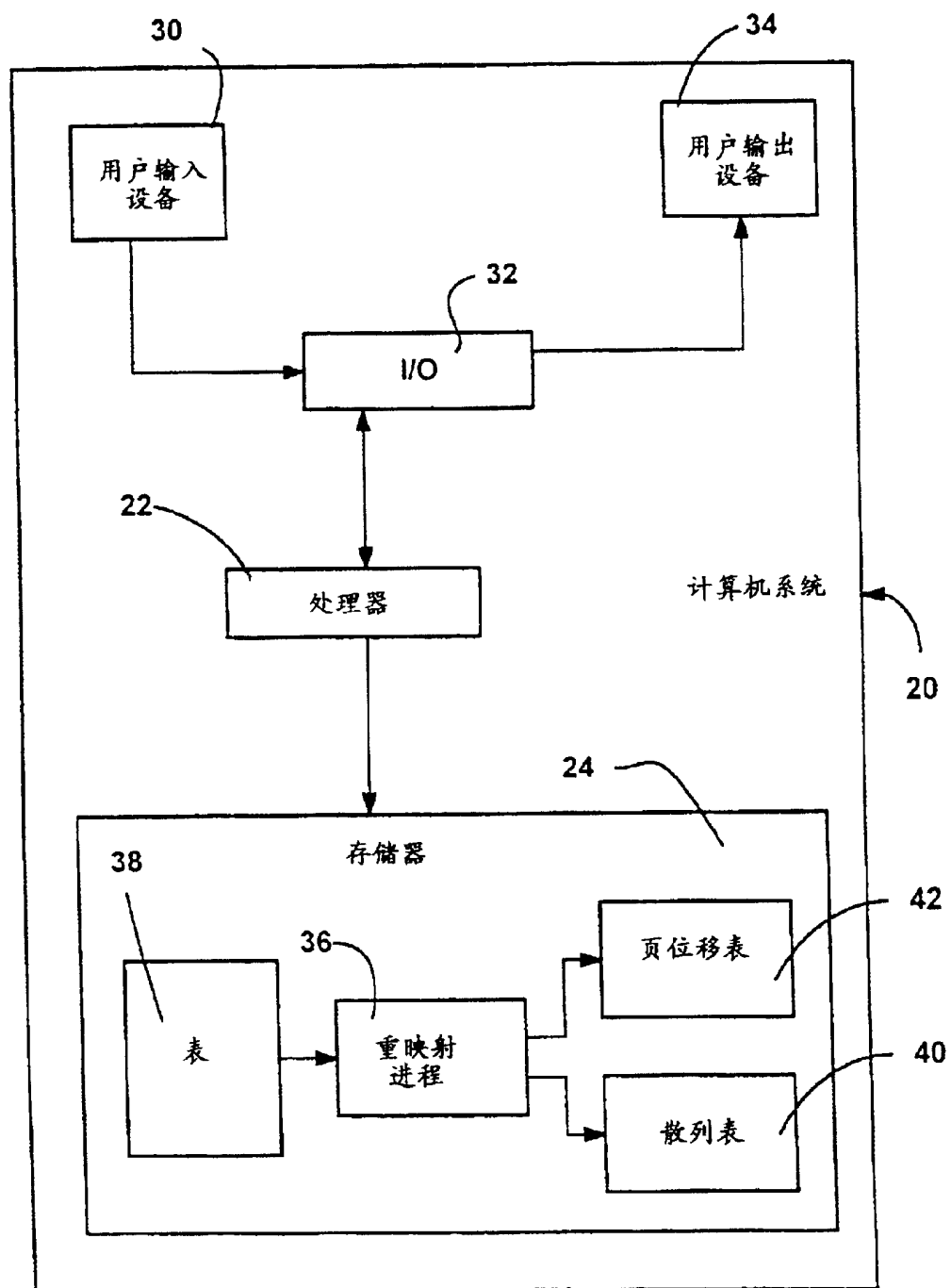


图 1

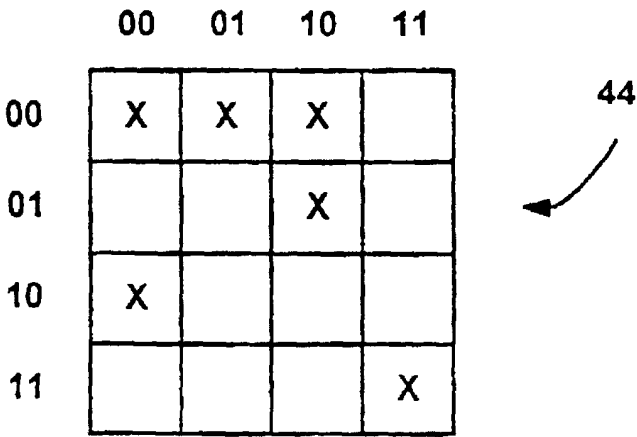


图 5

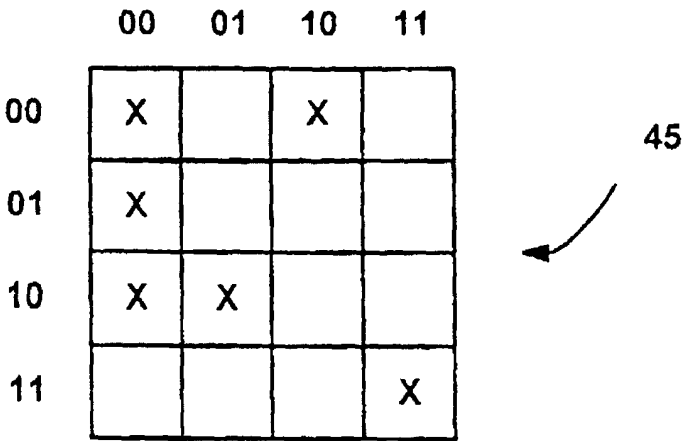


图 6

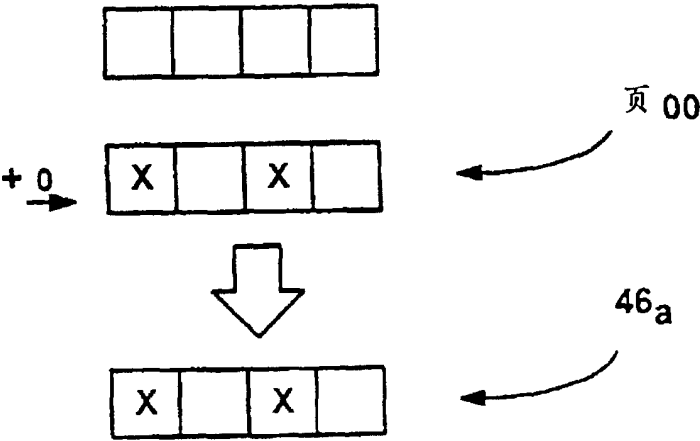


图 7

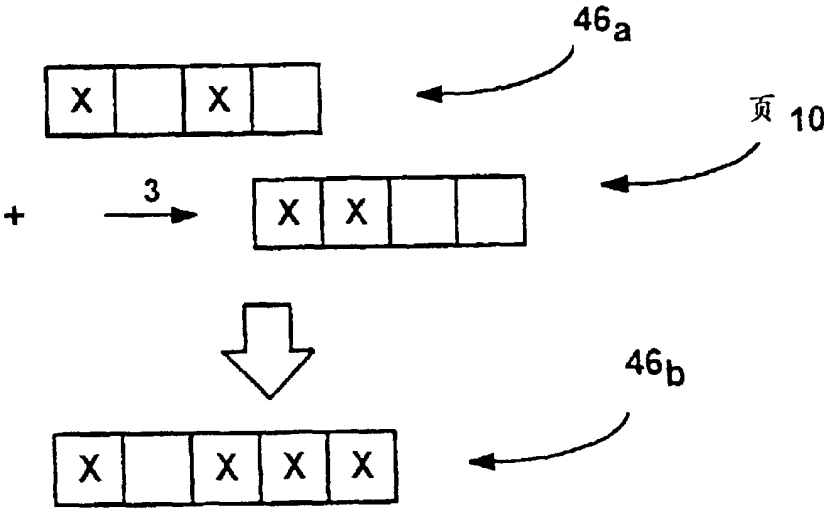


图 8

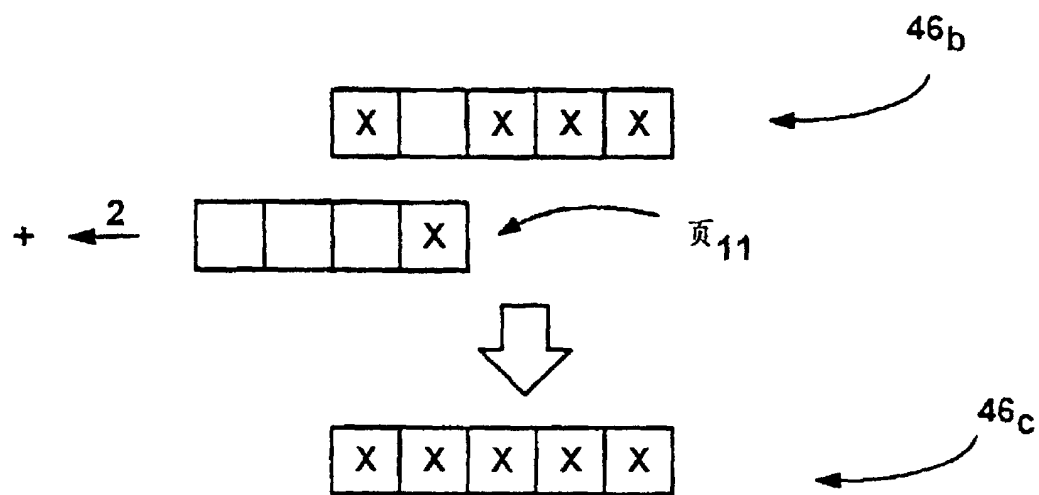


图 9

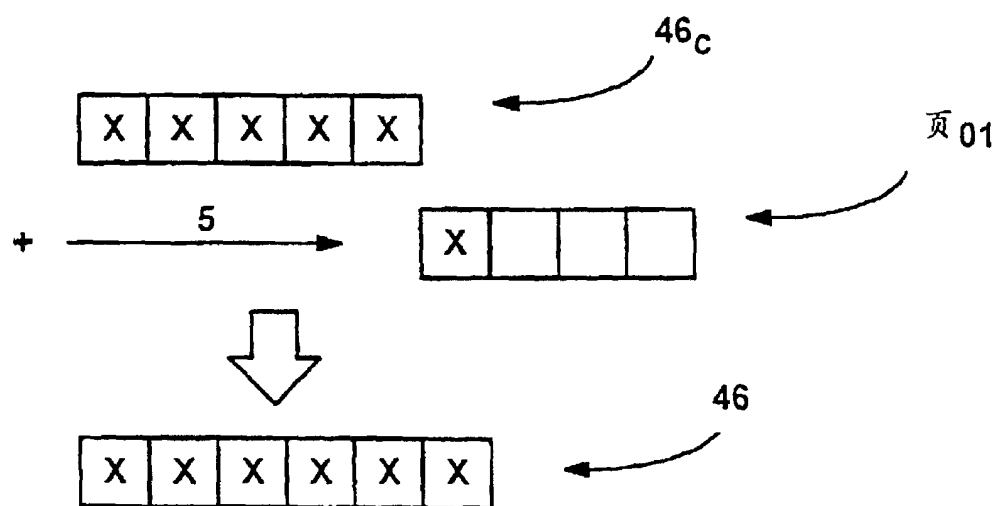


图 10

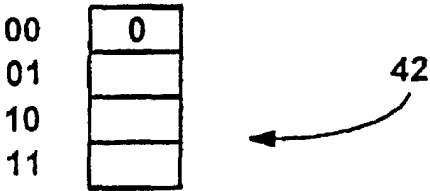


图 11

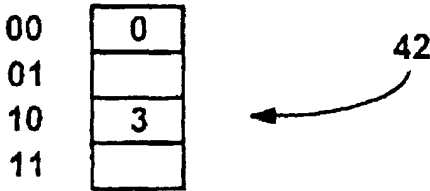


图 12

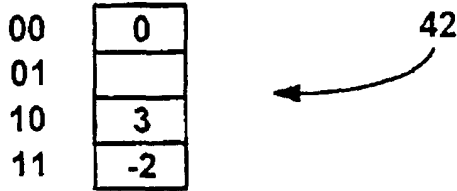


图 13

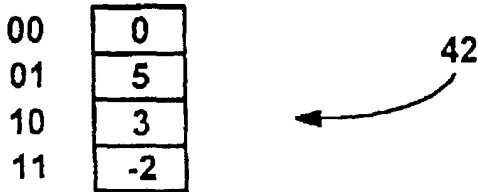


图 14

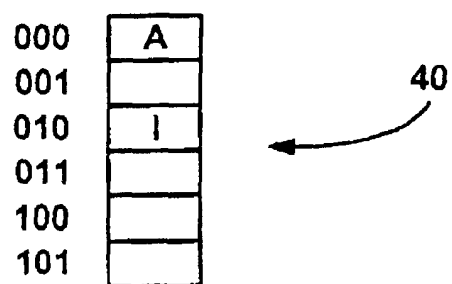


图 15

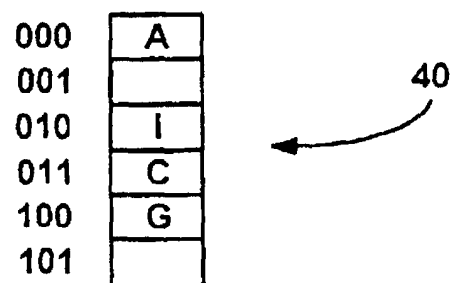


图 16

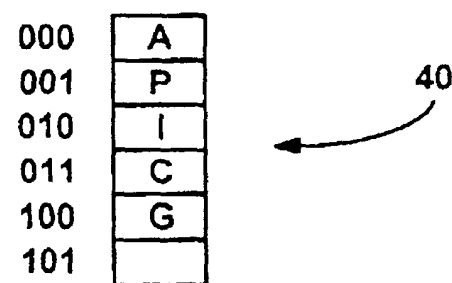


图 17

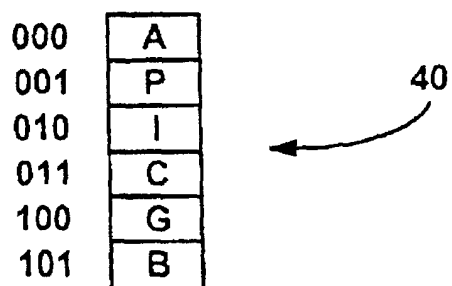


图 18

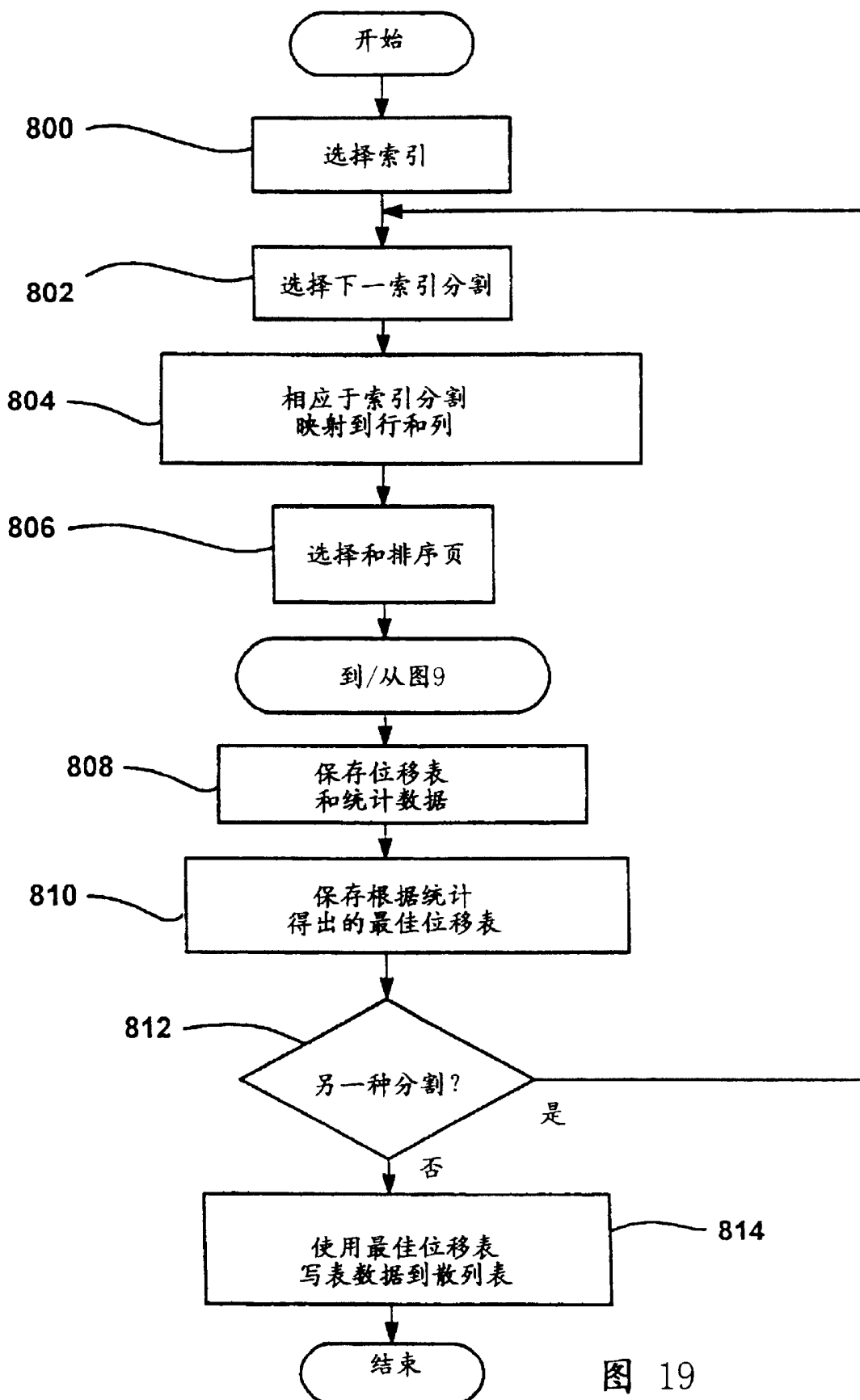


图 19

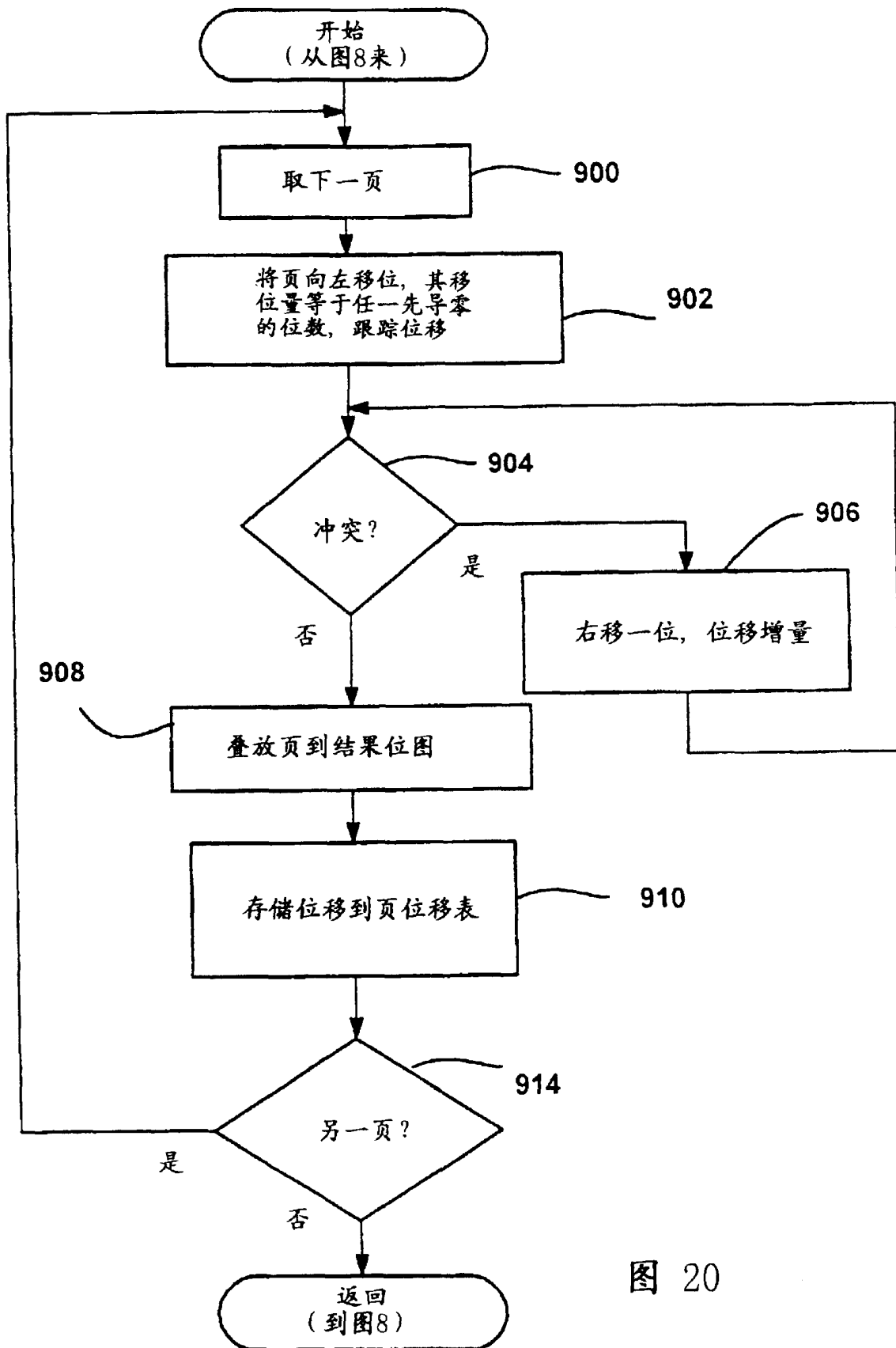
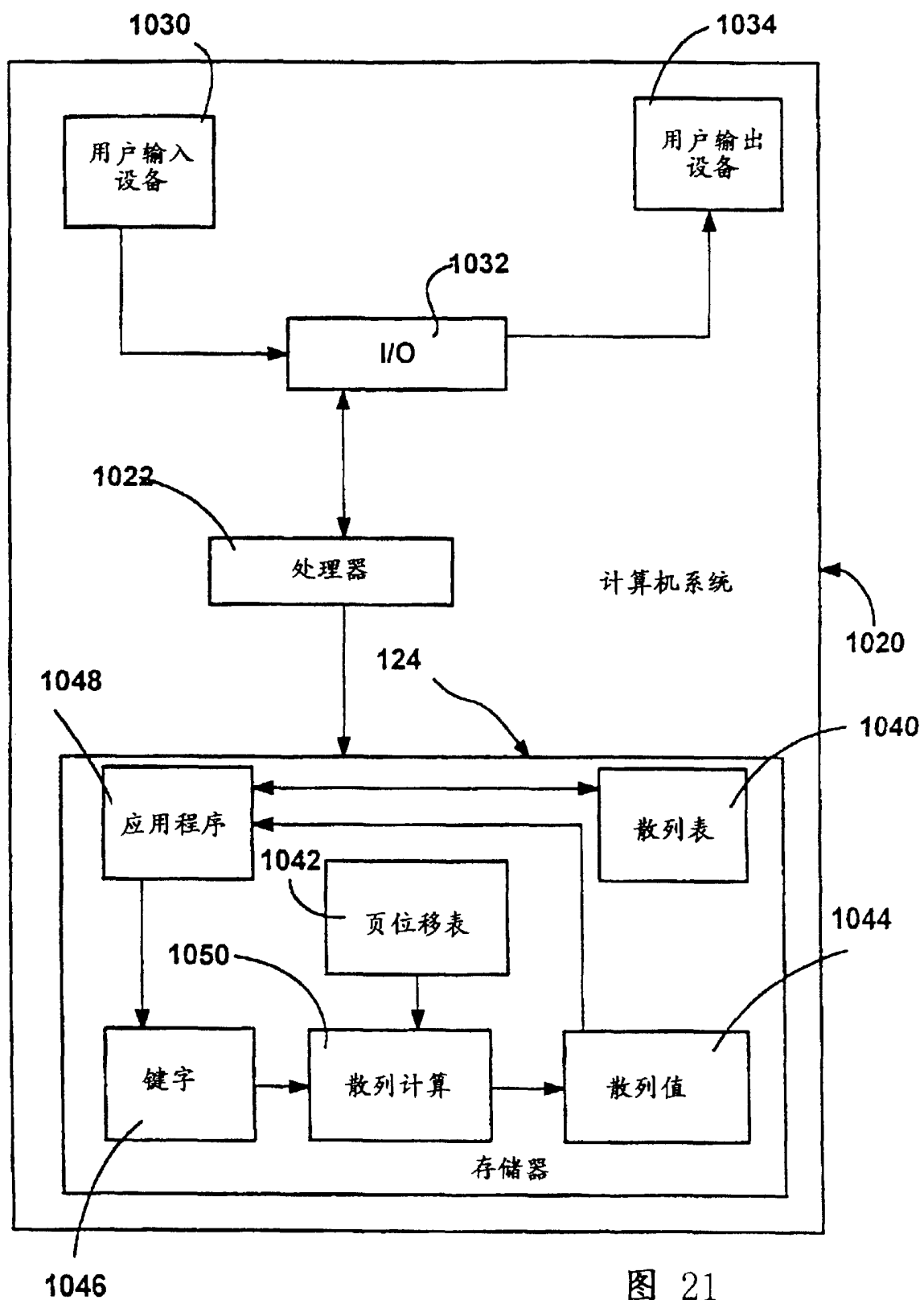


图 20



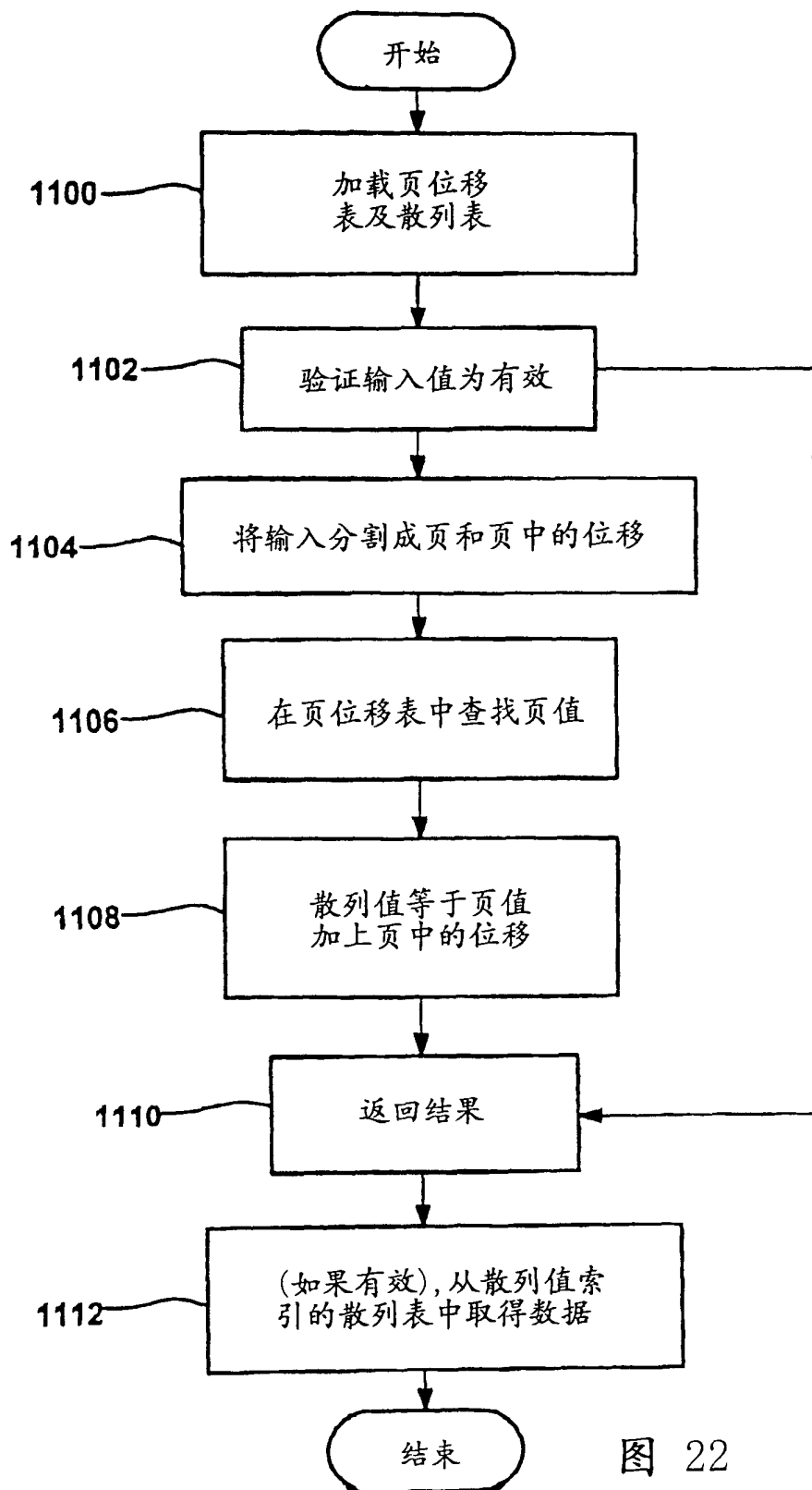


图 22