



(12)发明专利

(10)授权公告号 CN 105808209 B

(45)授权公告日 2018.10.23

(21)申请号 201610127442.1

(22)申请日 2012.05.02

(65)同一申请的已公布的文献号

申请公布号 CN 105808209 A

(43)申请公布日 2016.07.27

(30)优先权数据

13/099,463 2011.05.03 US

(62)分案原申请数据

201280026722.3 2012.05.02

(73)专利权人 高通股份有限公司

地址 美国加利福尼亚州

(72)发明人 谢尔盖·拉林

卢西恩·科德雷斯库

安舒曼·达斯古普塔

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 宋献涛

(51)Int.Cl.

G06F 9/30(2006.01)

(56)对比文件

CN 101075903 A, 2007.11.21,

CN 1531235 A, 2004.09.22,

CN 101114217 A, 2008.01.30,

US 2010/0223237 A1, 2010.09.02,

US 6199126 B1, 2001.03.06,

US 2006/0265572 A1, 2006.11.23,

US 2010/0083001 A1, 2010.04.01,

US 2011/0083001 A1, 2011.04.07,

审查员 张力

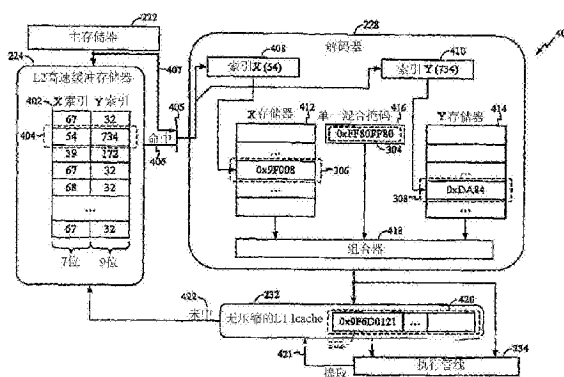
权利要求书2页 说明书17页 附图21页

(54)发明名称

用于内嵌在存储器层级内的经熵编码软件的存储和翻译的方法和装置

(57)摘要

本发明涉及用于内嵌在存储器层级内的经熵编码软件的存储和翻译的方法和装置。针对程序代码的片段产生X索引和Y索引格式的定制熵界限编码,连同由X样式存储器和Y样式存储器组成的定制解码词典。在运行时解码中,混合掩码与根据所述X索引选自所述X样式存储器的X样式以及根据所述Y索引选自所述Y样式存储器的Y样式一起使用,以确定可执行指令。所述混合掩码从所述X样式和所述Y样式识别要组合的位的次序。描述在经编码代码片段的执行期间解码机制和地址翻译的适当硬件实施和放置。还描述用以确定所述X索引、所述Y索引、所述X样式、所述Y样式,以及一个或一个以上混合掩码的包含一般过程的方法。



1. 一种确定用于有效地翻译压缩指令的混合掩码的方法,所述方法包括:

对第一母代混合掩码(960)和第二母代混合掩码(962)执行培育(938、946),其中所述培育包括:

组合所述第一母代混合掩码的开始部分(970)与所述第二母代混合掩码的尾部部分(971)以产生第一后代混合掩码(966),以及

组合所述第二母代混合掩码的开始部分与所述第一母代混合掩码的尾部部分以产生第二后代混合掩码(968);

通过半随机翻转个别位(976、977、978)使所述第一后代混合掩码和所述第二后代混合掩码变异(940、948)以产生更新混合掩码(942)的种子群集的变异后代混合掩码(979),其中所述半随机翻转个别位包括在所述第一后代混合掩码和所述第二后代混合掩码中将“0”改变为“1”或将“1”改变为“0”;

基于所述变异后代混合掩码更新所述种子群集;以及

基于X索引大小、Y索引大小、X样式存储器解压缩表大小和Y样式存储器解压缩表大小确定翻译的成本,其中所述成本包括外部存储装置的大小以及翻译词典的大小,所述X样式表示对于一特定指令所属的指令群组为共同的位样式,所述Y样式表示相同指令群组中的一组唯一位样式中的一者,且其中所述X索引指示所述X样式存储器中的位置的地址,所述Y索引指示所述Y样式存储器中的位置的地址;

根据所述翻译的所述成本从更新的所述种子群集确定混合掩码(957),其中根据所述所确定的混合掩码来组合压缩指令的位样式以将程序的所述压缩指令翻译为可执行形式。

2. 根据权利要求1所述的方法,其中所述培育进一步包括:

通过半随机选择过程随机选择交叉点,其中所述交叉点表示所述第一母代混合掩码的所述开始部分与所述第二母代混合掩码的所述尾部部分相组合的位置,且其中所述半随机选择为由伪随机数产生器引导的随机选择;以及

当所述选定交叉点是每一母代混合掩码的终点时,产生继承所述第一母代混合掩码的所述第一后代混合掩码和继承所述第二母代混合掩码的所述第二后代混合掩码。

3. 根据权利要求1所述的方法,其进一步包括:

根据半随机选择频率基于混合掩码对中的半随机选择的交叉点随机选择培育,其中根据每一母代混合掩码内的所述选定交叉点确定每一母代混合掩码的所述开始部分和所述尾部部分,且其中所述半随机选择为由伪随机数产生器引导的随机选择。

4. 根据权利要求1所述的方法,其中变异进一步包括:

将所述后代混合掩码的一者中的个别位半随机翻转以确定变异后代混合掩码;以及
用所述变异后代混合掩码代替混合掩码的所述种子群集中的混合掩码。

5. 根据权利要求1所述的方法,其中所述程序包含启动代码、操作系统代码和多个应用程序。

6. 根据权利要求1所述的方法,其进一步包括:

使用与每一混合掩码相关联的翻译硬件的当前成本确定所述经更新的种子群集中的每一混合掩码的配合度等级,其中混合掩码的配合度等级表示用于提供高压缩水平的益处水平。

7. 根据权利要求1所述的方法,其进一步包括:

选择种子混合掩码的初始种子群集以包含针对具有类似于所述程序中使用的指令的指令的先前程序具有高配合度等级的半随机选择的混合掩码和先前使用的混合掩码。

用于内嵌在存储器层级内的经熵编码软件的存储和翻译的方法和设备

[0001] 分案申请的相关信息

[0002] 本案是分案申请。该分案的母案是申请日为2012年5月2日、申请号为201280026722.3、发明名称为“相用于内嵌在存储器层级内的经熵编码软件的存储和翻译的方法和设备”的发明专利申请案。

技术领域

[0003] 本发明大体涉及具有用于改进内嵌软件中的代码密度的压缩指令集的处理器,且更特定来说涉及用于产生压缩指令、存储所述压缩指令且翻译所述压缩指令的技术。

背景技术

[0004] 例如蜂窝式电话、膝上型计算机、个人数字助理(PDA)等许多便携式产品需要使用执行程序支持通信和多媒体应用的处理器。此类产品的处理系统包含一个或一个以上处理器,每一者具有针对指令、输入运算数和执行结果的存储装置。举例来说,针对处理器的指令、输入运算数和执行结果可存储在由通用寄存器文件、多层指令高速缓冲存储器、数据高速缓冲存储器和系统存储器组成的层级存储器子系统中。

[0005] 为了提供高代码密度,可使用本机指令集架构(ISA),其具有两个指令格式,例如作为32位指令格式的子集的16位指令格式。在许多情况下,所提取的16位指令在允许执行硬件被设计为仅支持32位指令格式的解码过程之前或之中由处理器变换为32位指令。作为32位指令的子集的16位指令的使用是限制可编码为16位格式的信息的量的约束。举例来说,16位指令格式可限制可指定的可寻址源运算数寄存器和目的地寄存器的数目。16位指令格式例如可使用3位或4位寄存器文件地址字段,而32位指令可使用5位字段。如果两个格式在程序中部分归因于指令寻址约束(例如,形成分支到16位和32位指令)而混合,那么处理器管线复杂性也可能增加。并且,代码压缩的要求依据不同程序而变化,从而使针对一个程序选择的固定16位指令格式不太有利于由不同程序使用。在此方面,现有处理器的传统代码不能有效地利用两个指令格式来显著改进代码密度且在许多情况下满足实时要求。这些和其它约束限制了具有作为标准大小指令中使用的字段的子集的字段的缩减大小指令的效力。

发明内容

[0006] 在本发明的若干方面中,本发明解决将使用本机指令集的程序与使用压缩指令集的压缩程序解耦的需要。本文提出的技术允许针对内嵌软件极其有效地利用存储装置和发射管道,而不影响软件的执行时间和效率。

[0007] 为实现此些目的,本发明的一实施例应用一种对经压缩可变长度指令解压缩的方法。紧密固定长度指令从系统存储器提取。所述系统存储器经配置以存储经压缩可变长度指令,其中每一紧密固定长度指令包括X索引和Y索引。对于每一紧密固定长度指令,使用X

索引从X存储器提取X位样式,且使用Y索引从Y存储器提取Y位样式。基于与每一紧密固定长度指令相关联的混合掩码将X位样式与Y位样式组合为经解压缩可变长度指令,其中经解压缩可变长度指令的格式长度由所述相关联混合掩码确定。

[0008] 本发明的另一实施例提出一种用于翻译存储在虚拟存储器系统中的压缩指令的设备。分页指令高速缓冲存储器经配置以存储与未压缩指令的页混合的压缩指令的页。指令翻译后备缓冲器(TLB)经配置以存储将分页高速缓冲存储器中的页识别为存储压缩指令的地址翻译条目。

[0009] 本发明的另一实施例提出一种用于将压缩指令翻译为可执行格式的系统。层2高速缓冲存储器经配置以存储针对每一压缩指令的X索引和Y索引。翻译单元经配置以从层2高速缓冲存储器接收压缩指令,针对每一所接收的压缩指令使用X索引和Y索引从翻译存储器选择X位样式和Y位样式,且使用程序指定的混合掩码用于将选定的X位样式和Y位样式组合为本机指令格式。层1高速缓冲存储器经配置以针对每一压缩指令存储本机指令格式。

[0010] 本发明的另一实施例提出一种确定用于有效地翻译压缩指令的混合掩码的方法。培育表示为来自混合掩码的种子群集的基因的混合掩码对以产生后代混合掩码对。后代混合掩码变异以产生更新种子群集的变异后代混合掩码。从经更新种子群集确定提供高压压缩水平的混合掩码,其中根据所确定的混合掩码来组合位样式以将程序的压缩指令翻译为可执行形式。

[0011] 从以下具体实施方式和附图将了解对本发明的更完整理解以及本发明的更多特征和优点。

附图说明

[0012] 图1是其中可有利地采用本发明的实施例的示范性无线通信系统的框图;

[0013] 图2是根据本发明经设计用于运行时解压缩的代码压缩系统;

[0014] 图3说明根据本发明将基于混合掩码的指令分裂为具有字节重叠填补位的X样式和Y样式的指令分割过程的示范性元件;

[0015] 图4是根据本发明具有以压缩形式存储在主存储器和层2高速缓冲存储器中的程序的解压缩器系统,所述程序经解压缩以存储在层1高速缓冲存储器中;

[0016] 图5A说明根据本发明将基于第二混合掩码的第二指令分裂为具有字节重叠填补位的第二X样式和第二Y样式的第二指令分割过程的示范性元件;

[0017] 图5B说明根据本发明在具有三个不同掩码 $A_{mm} \neq B_{mm} \neq C_{mm}$ 的X存储器和Y存储器中图3的 A_x 、 A_y 样式以及图5A的 B_x 、 B_y 样式的示范性存储布置;

[0018] 图5C是根据本发明具有以压缩形式存储在主存储器和层2高速缓冲存储器中的程序的解压缩器系统,所述程序使用多个混合掩码和索引压缩经解压缩从而以未压缩形式存储在层1高速缓冲存储器中;

[0019] 图6说明根据本发明的VLIW包压缩格式;

[0020] 图7说明根据本发明具有压缩页和未压缩页的分页指令翻译后备缓冲器(ITLB)和存储器组织;

[0021] 图8A说明根据本发明用于解压缩压缩指令页和存取未压缩指令页的分页解压缩器系统;

- [0022] 图8B说明示范性解压缩状态图,其说明用于在处理器管线上执行的L2高速缓冲存储器压缩页指令和L1 Icache解压缩指令的状态;
- [0023] 图9A-9C说明根据本发明基于一般算法的变型的一般混合掩码确定过程;
- [0024] 图9D说明根据本发明的交叉算法的实施方案;
- [0025] 图9E说明根据本发明的变异算法的实施方案;
- [0026] 图10说明根据本发明的示范性当前成本过程;
- [0027] 图11说明根据本发明的经加权汉明启发式;
- [0028] 图12说明根据本发明具有填补-分类的穷举搜索X/Y表压紧过程;
- [0029] 图13说明根据本发明到压缩存储过程中的符号插入;以及
- [0030] 图14说明根据本发明具有填补-分类过程的三角X/Y表压紧。

具体实施方式

[0031] 现将参看附图更充分描述本发明,附图中展示本发明的若干实施例。然而,本发明可体现为各种形式且不应解释为限于本文陈述的实施例。事实上,提供这些实施例使得本发明将为详尽和完整的,且将把本发明的范围完全传达给所属领域的技术人员。

[0032] 供在上面操作或用于实行根据本发明的教示的操作的计算机程序代码或“程序代码”可初始以例如C、C++、JAVA®、Smalltalk、JavaScript®、Visual Basic®、TSQL、Perl等高级编程语言或以各种其它编程语言编写。通过使用以本机指令格式编码的指令将高级程序代码转换为本机汇编程序而将以这些语言中的一者编写的源程序或源代码编译为目标处理器架构。目标处理器架构的程序还可直接以本机汇编语言编写。本机汇编程序使用机器级二进制指令的指令助记符表示。如本文使用的编译器或人类程序员产生的程序代码或计算机可读媒体指代例如格式可由处理器理解的目标代码等机器语言代码。

[0033] 图1说明其中可有利地采用本发明的实施例的示范性无线通信系统100。出于说明的目的,图1展示三个远程单元120、130和150以及两个基站140。将认识到,常见无线通信系统可具有更多远程单元和基站。包含如分别由组件125A、125C、125B和125D表示的硬件组件、软件组件或两者的远程单元120、130、150以及基站140已适于体现如下文进一步论述的本发明。图1展示从基站140到远程单元120、130和150的前向链路信号180,以及从远程单元120、130和150到基站140的反向链路信号190。

[0034] 在图1中,远程单元120展示为移动电话,远程单元130展示为便携式计算机,且远程单元150展示为无线本地回路系统中的固定位置远程单元。借助实例,远程单元或者可为蜂窝式电话、寻呼机、对讲机、手持式个人通信系统(PCS)单元、例如个人数字助理等便携式数据单元,或例如仪表读取设备等固定位置数据单元。尽管图1说明根据本发明的教示的远程单元,但本发明不限于这些示范性的所说明的单元。本发明的实施例可适当地用于任何处理器系统中。

[0035] 图2是根据本发明经设计用于运行时解压缩的用于代码压缩的压缩系统200。压缩系统200包含如上文描述的源代码和源代码中呈未压缩形式的二进制库文件,以及包括正编译的当前程序应用的库文件204。压缩系统200还包含编译器和链接器206、任选的简档反馈信息208,其用于基于本机指令集架构(ISA)格式和支持的数据区段212产生链接的可执行代码210。本机ISA由固定未压缩格式表示且可表示多种方法,包含(例如)固定64或32或

16位编码以及这些编码的混合。本机ISA是针对一般效用开发的且不特定为手头的当前应用定制。通过维持固定字边界(例如,32位指令字边界),尽管16位和32位指令可一起混合在代码中,也可使用仅支持分支、调用、返回等的固定字地址的寻址模型。

[0036] 从此ISA选择的指令可依据当前应用而压缩和定制,同时维持代码的可寻址性且保证快速、固定等待解压缩时间。此压缩可自动化以在线性时间内产生经压缩代码。原始ISA依据其极其一般化性质而具有低信息熵,其根据本发明通过针对给定源代码和库文件204产生定制熵边界编码而增加。提供链接的可执行代码210作为到翻译工具216的输入,翻译工具216产生经压缩代码218和解码表220。经压缩代码218和支持的数据区段212存储在静态存储装置214中,例如硬盘、光盘、内嵌式装置的快闪存储器,或可从其将选定代码下载到处理器复合体203以供执行的其它此类存储媒体。处理器复合体203包含主存储器222、层2高速缓冲存储器(L2高速缓冲存储器)和处理器核心226。处理器核心226包含具有根据本发明的翻译存储器(TM)230、层1指令高速缓冲存储器(L1 Icache)232和执行管线234的解码器228。经压缩代码存储在静态存储装置214、主存储器222和L2高速缓冲存储器224中。经解压缩代码存储在L1高速缓冲存储器中且由执行管线234执行。下文更详细描述用于产生经压缩代码218且用于在解码器228中解码经压缩指令的翻译工具216的各个实施例。

[0037] 处理器复合体203可适当地用于图1的硬件组件125A-125D中,用于执行以未压缩形式存储在L1 Icache 232中以及以压缩形式存储在L2高速缓冲存储器224和主存储器222中的程序代码。为了论述的清晰未展示可连接到处理器复合体的外围装置。处理器核心226可为通用处理器、数字信号处理器(DSP)、专用处理器(ASP)等。处理复合体203的各个组件可使用专用集成电路(ASIC)技术、现场可编程门阵列(FPGA)技术或其它可编程逻辑、离散门或晶体管逻辑,或适于既定应用的任何其它可用的技术实施。尽管展示单一处理器核心226,但本发明的压缩指令的处理适用于超标量设计和实施并行管线的其它架构,例如多线程、多核心,和极长指令字(VLIW)设计。

[0038] 图3说明根据本发明将基于二进制混合掩码(MM)304的原始ISA固定大小指令A302分裂为具有重叠填补位310和312的Ax样式306和Ay样式308的指令分割过程的示范性元件300。填补位归因于现代存储器系统强加的要求而产生,以至少以字节粒度片段表示指令和数据。具有字节粒度片段的格式的使用用于提供新颖的压紧表示,从而允许位粒度上的存储重叠,同时满足存储系统的字节粒度要求。

[0039] 为压缩原始ISA代码片段,将代码片段分割为指令群组,每一群组贡献单一共享X样式和一组唯一Y样式。Ax样式306表示对于指令A所属的指令群组为共同的位样式。Ay样式308表示相同指令群组中的一组唯一位样式中的一者。注意,代码片段可分割为1与N之间的任何数目的群组,其中N为代码片段中的原始指令的数目。用以确定群组的最佳或近似最佳数目以及最佳或近似最佳混合掩码的过程是下文进一步描述的非平凡过程。代码片段的X样式存储在包括X存储器的X词典中,且代码片段的Ay样式存储在包括Y存储器的Y词典中。X索引是X存储器中的位置的地址,且Y索引是Y存储器中的位置的地址。这两个索引、来自X和Y词典的样式以及二进制混合掩码的组合确切地表示原始指令。字节可寻址X/Y词典样式以压紧形式存储在X/Y存储器中,仍可存取而不需要可变长度解压缩。可变长度解压缩基于将可变长度压缩符号反向映射到本机固定大小字母表中的过程。固定大小索引压缩用于此表示且在稍后论述。压缩操作314使用代码片段的至少一个混合掩码从指令302选择Ax样式

306和Ay样式308。在以下实例中,十六进制数字或Hex数字用‘0x’前缀表示。举例来说,指令302[0x9F6D0121]与混合掩码304[0xFF80FF80]组合以产生Ax样式306[0x9F00(8,9,A,B)]和Ay样式308[0xDA8(4,5,6,7)]。解码器228解压缩操作316使用代码片段的所述至少一个混合掩码、X索引提取的X存储器样式和Y索引提取的Y存储器样式来解压缩经压缩指令。举例来说,混合掩码304[0xFF80FF80]与从X存储器提取的Ax样式306[0x9F00(8,9,A,B)]以及从Y存储器提取的Ay样式308[0xDA8(4,5,6,7)]组合,以产生原始指令302[0x9F6D0121]。

[0040] 如上文描述,X样式和Y样式存储在字节可寻址X存储器和字节可寻址Y存储器中。X样式、Y样式或两者的索引压缩利用一过程,所述过程消除副本、减少物理存储的双重利用且使填补位(例如,重叠填补位310和312)以及字节可寻址位置重叠。可通过将第一X样式的末尾位与第二X样式的开始位共享而使第一X样式与第二X样式重叠。在此比较中,具有重叠填补位的字节允许例如图中指示的另一选项范围。举例来说,Ax样式306包括三个字节,包含第一字节0x9F、第二字节0x00和第三字节,第三字节包括具有集合{8,9,A,B}中的一者的值(归因于重叠填补位310)的第一半字节,以及可为从0x00到0xFF且包含0x00和0xFF在内的任何数字的第二半字节。待与第一Ax样式306重叠的第二X样式将具有开始位的第一半字节中的集合{8,9,A,B}中的一者,以及开始位的第二半字节中的从0x00到0xFF且包含0x00和0xFF在内的任何数字。待与第一Ax样式306重叠的第三X样式的另一可能重叠样式具有开始位的第一字节中的0x00、开始位的下一半字节中的集合{8,9,A,B}中的一者,以及开始位的下一连续半字节中的从0x00到0xFF且包含0x00和0xFF在内的任何数字。对于Y样式,Ay样式308包括两个字节,包含第一字节0xDA和第二字节,所述第二字节包括第一半字节0x8以及归因于重叠填补位312而可为集合{4,5,6,7}中的一者的第二半字节。利用此些能力,有可能在128位线中存储四个未压缩指令,且对于具有填补和重叠的32位到16位压缩能够在同一空间中存储八个以上压缩指令。举例来说,如果压缩指令中的四个可以字节重叠存储,那么十个16位压缩指令可存储在128位线中。因此,与无填补的方法相比,寻址空间被压缩,且索引寻址也将被压缩。待与第一Ay样式308重叠的第二Y样式将具有开始位的第一半字节中的‘0x8’,以及作为集合{4,5,6,7}中的一者的第二半字节。利用单一混合掩码,所有X样式具有相同数目的位,且所有Y样式具有相同数目的位。在较大数目的X和Y样式的情况下,有可能具有多种映射,其中仅少数映射占据最少存储,其被视为近似最佳或可接受映射。最佳或近似最佳映射的选择是NP完全问题,且不能针对任何显著数目的X和Y样式实际上得以解决。无论如何,本发明使用在线性时间内产生可接受映射的启发式。

[0041] 图4是根据本发明具有以压缩形式存储在主存储器222和L2高速缓冲存储器224中的程序的解压缩器系统400,所述程序经解压缩以存储在L1 Icache 232中。L2高速缓冲存储器224包含XY索引存储器402,其将X索引和Y索引对存储在可寻址位置中,例如具有7位X索引值0x54和9位Y索引值0x734的XY条目404。多路复用器405用于选择关于L2高速缓冲存储器命中406的XY条目,或关于L2高速缓冲存储器中的未命中的来自主存储器222的XY值407。在具有索引X寄存器408、索引Y寄存器410、X存储器412、Y存储器414、单一MM寄存器416和组合器418的解码器228中实现解压缩操作。L1 Icache 232包含多个高速缓冲存储器线,例如保持解压缩指令的高速缓冲存储器线420。

[0042] 在程序加载时或在内嵌式系统启动过程中,主存储器222加载有压缩代码,X存储器412和Y存储器414加载有相关联X和Y词典语境,且单一二进制混合掩码在MM寄存器416中

设定。注意,X和Y存储器语境以及混合掩码可视需要在执行期间再加载。此再加载可构成更小区段中的原始代码片段的进一步粒度,每一区段具有其定制编码。举例来说,一些复杂的内嵌式系统(例如,智能电话)可从主应用调用多个独立的子应用,所述子应用不共享代码空间且为自含式的。每一此类应用可具有其自身的包括X/Y词典和MM的定制编码,其在子过程启动时加载。

[0043] 此情境的较好实例将是开始电子邮件处理应用的智能电话操作系统(OS),所述电子邮件处理应用在其执行持续时间期间占据大部分系统资源,且执行仅属于电子邮件处理应用的代码持续一段较长时间。电子邮件应用的定制编码在启动时加载,且仅在电子邮件应用已完成操作时被不同OS定制编码替代。此外,可能的情境是,单一词典用于OS和电子邮件处理应用,但经由将索引X(408)和/或索引Y(410)寄存器内容调整为偏移到X/Y词典的适当区段中来利用不同区段。索引X(408)和/或索引Y(410)寄存器内容的加载可经由通过例如OS中断的系统层级请求或经由连同应用代码一起编码的特殊指令而实现,所述特殊指令被处理器理解为对于再加载索引X(408)和/或索引Y(410)寄存器内容的请求。在此情况下,解码表220的数目等于针对L2中的同时驻留可能具有不同编码的程序片段的数目。使用以上智能电话实例,如果OS专用于其自身的具有始终保持OS特定编码的翻译存储器(TM)230的解码器228,那么系统可并入有另一应用特定TM230,其保持针对所利用的系统应用定制的不同编码。确定哪一翻译存储器230为适当的可以每代码页为基础经由TLB条目进行。相比之下,如果系统中仅设计一个TM230,那么依据所安装TM230的容量,以下程序可为适当的。如果仍需要超出OS TM230的容量的多个独立编码,那么如在以上实例中,一旦OS开始电子邮件处理应用,就将新的X/Y词典放置到X/Y存储器(412、414)中。MM寄存器416可更新,且发布L2/TLB刷新请求。L2/TLB刷新请求使对应于L2/TLB中的OS的经压缩代码无效。对于描述的剩余部分,使用具有待用于包含任何应用代码的整个系统的单一TM230和单一编码的解码器228来描述解压缩器系统400。

[0044] 接下来,执行管线234开始从L1 Icache 232提取指令。初始,对L1 Icache的每次存取产生未中指示422,从而导致对L2高速缓冲存储器224的存取。初始,对L2高速缓冲存储器224的存取也产生未中,从而导致对主存储器222的存取,主存储器222以在L2高速缓冲存储器224中加载且经由多路复用器405转发到解码器228的压缩指令响应。解码器228将XY索引压缩指令解压缩为未压缩格式以供存储在L1 Icache 232中且供在执行管线234中执行,如关于图3的解压缩操作316描述。在较短操作周期之后,L1 Icache 232和L2高速缓冲存储器224将已达到稳定状态。

[0045] 从处理器的角度来说,执行管线234尝试提取操作,在L1 Icache 232中搜索指令的提取地址和控制信号421。初始,L1 Icache 232确定指令不存在且将未中指示422发布给L2高速缓冲存储器224。L2高速缓冲存储器提取操作例如是针对XY条目404,其是L2高速缓冲存储器224中的命中,从而致使XY条目404经由多路复用器405传递到解码器228。XY条目404分裂,其中X索引值0x54在索引X寄存器408中接收,且Y索引值0x734在索引Y寄存器410中接收。在地址0x54处从X存储器412提取的X样式306施加到组合器418。在地址0x734处从Y存储器414提取的Y样式308也施加到组合器418。存储在MM寄存器416中的单一混合掩码(MM)304[0xFF80FF80]进一步施加到组合器418。组合器418根据MM304将来自X样式306的适当位与来自Y样式308的适当位组合以产生原始指令302,原始指令302存储在高速缓冲存

储器线420中且传递到执行管线234。

[0046] 此系统的另一额外特征是以隐式加密格式存储的程序内容。尽管不对指令流执行特定加密类型的数据加扰,但程序代码以应用特定和压缩形式存储在静态存储装置214、主存储器222和L2高速缓冲存储器224中。由于程序代码的经编码状态的一部分驻留在TM 230中的处理器核心226(其在最终产品中不容易存取)内部,所以静态存储装置214以及上部存储器层级222和224内容不足以恢复原始程序,从而使得难以分析或复制。

[0047] 处理器核心226处理多个地址空间。执行管线234在虚拟地址空间中操作,虚拟地址空间不同于贯穿整个存储器层级使用的物理地址空间。一个地址空间翻译到另一地址空间的任務通常由翻译后备缓冲器(TLB)执行,且在物理地址空间含有相同大小的指令的情况下具有减小的复杂性。因此,根据本发明,执行代码片段以加起来占据固定字节对准空间的X和Y索引的固定大小对表示。此方法(如下文更详细描述)允许基于物理寻址空间分页过程而混合压缩和未压缩指令,其中代码页表示由TLB的单一条目处理的代码的原子单位。

[0048] 图5A说明利用填补位将X和Y索引放置到X和Y存储器中的情况。根据本发明将基于第二混合掩码504的指令B 502分裂为具有字节重叠填补位的第二X样式和第二Y样式的另一指令B分割过程的示范性元件500。为压缩第二代代码片段,将第二代代码片段分割为指令群组,每一群组贡献Bx样式和一组唯一By样式。Bx样式506表示对于来自第二代代码片段的指令群组为共同的位样式。By样式508表示相同指令群组中的一组唯一位样式中的一者。代码片段的Bx样式存储在包括X存储器的X词典中,且代码片段的By样式存储在包括Y存储器的Y词典中。X索引是X存储器中的位置的地址,且Y索引是Y存储器中的位置的地址。在压缩过程中,举例来说,指令502[0xBAFF0FEB]与混合掩码504[0xFFC0FF80]组合以产生Bx样式506[0xBAC3(E,F)]和By样式508[0xFF5(8,9,A,B,C,D,E,F)]。在解压缩过程中,举例来说,混合掩码504[0xFFC0FF80]与从X存储器提取的Bx样式506[0xBAC3(E,F)]以及从Y存储器提取的By样式508[0xFF5(8,9,A,B,C,D,E,F)]组合,以产生原始指令502[0xBAFF0FEB]。

[0049] 图5B说明图3的Ax、Ay样式、图5A的Bx、By样式以及此处由X存储器522和Y存储器528中的Cx和Cy表示的又一指令C的示范性存储布置520。根据本发明,三个指令A、B和C具有不同掩码 $A_{mm} \neq B_{mm} \neq C_{mm}$ 。举例来说,X存储器522和Y存储器528为字节可寻址的且利用64位长存取线。Ax样式306[0x9F00(8,9,A,B)]以二进制编码为Ax 523[1001 1111 0000 0000 10],By样式506[0xBAC3(E,F)]以二进制编码为Bx 524[1011 1010 1100 0011 111],且示范性Cx 525以二进制编码为Cx 525[1110 0000 0000 0000 1]。在示范性存储布置520中,混合掩码 A_{mm} 304[0xFF80FF80]、 B_{mm} 504[0xFFC0FF80]和示范性 C_{mm} [FF80FF00]不相等。因此,与每一混合掩码相关联的X样式是不同数目的位,且同样与每一混合掩码相关联的Y样式是不同数目的位。

[0050] X样式、Y样式或两者的索引压缩利用重叠填补位(例如,重叠填补位310、312、510和512)以及字节可寻址位置。可通过将第一X样式的末尾位与第二X样式的开始位比较而使第一X样式与第二X样式重叠。在此比较中,具有重叠填补位的字节允许例如图5B中指示的另一选项范围。举例来说,Ax样式523包括三个字节,包含第一字节0x9F、第二字节0x00和第三字节,第三字节包括具有为集合{8,9,A,B}中的一者的值(归因于重叠填补位310)的第一半字节,以及可为从0x00到0xFF且包含0x00和0xFF在内的任何数字的第二半字节。Bx样式524与第一Ax样式523重叠且在Bx样式524的第一半字节中具有值0xB,其是集合{8,9,A,B}

中的一者。Bx样式524的第二半字节具有值0xA,其是从0x00到0xFF的数字。Cx样式525的开始位与Bx样式524的末尾位重叠。对于Y样式,Y存储器528中未展示重叠。样式之间的位(例如,位534和535)可为任何二进制值,因为其不在压缩或解压缩过程中使用。如果归因于代码改变或添加,产生具有第一字节0x58的新Dy样式,那么新Dy样式可与By样式532重叠,如图5B所示。并且,具有第一字节0x5和作为集合{8,9,A,B,C,D,E,F}中的一者的第二字节的其它样式可与By样式532的末尾位重叠。

[0051] 图5C说明根据本发明具有以压缩形式存储在主存储器542和层2高速缓冲存储器544中的程序的解压缩器系统540,所述程序在解码器546中使用多个混合掩码和索引压缩而解压缩从而以未压缩形式存储在层1指令高速缓冲存储器(L1 Icache) 548中。L2高速缓冲存储器544包含XYM索引存储器552,其将X索引、Y索引和混合掩码索引存储在可寻址位置中,例如具有9位X索引值0x102、8位Y索引值0xA9和2位M索引值0x2的XYM条目554。多路复用器555用于选择关于L2高速缓冲存储器命中556的XYM条目或来自主存储器542的XYM值557。在解码器546中实现解压缩操作,所述解码器具有索引X寄存器558、索引Y寄存器560、MM索引寄存器561、X存储器562、Y存储器564、多个MM存储器566、双线X寄存器568、双线Y寄存器570和组合器572。L1 Icache 548包含多个高速缓冲存储器线,例如高速缓冲存储器线580。在初始操作周期之后,L2高速缓冲存储器544和主存储器542加载有压缩代码,且L1 Icache 548加载有解压缩代码。

[0052] 当执行管线550尝试从L1 Icache 548提取581指令且确定指令不存在时,发布未命中指示582,且将提取重定向到L2高速缓冲存储器544。举例来说,所述提取针对作为L2高速缓冲存储器544中的命中的XYM条目554,从而致使XY条目554经由多路复用器555传递到解码器546。XYM条目554分裂,其中X索引值0x102在索引X寄存器558中接收,Y索引值0xA9在索引Y寄存器560中接收。且M索引值0x2在索引MM寄存器561中接收。在地址0x0x100处从X存储器562提取含有X样式506[0xBAC3E]的线,且将其加载到双线X寄存器568。在地址0xA8处从Y存储器564提取含有Y样式508[0xFF58]的线,且将其加载到双线Y寄存器570。在地址0x2处从多个MM存储器566提取混合掩码504[0xFFC0FF80],且将其加载到MM寄存器571。基于X索引值0x102从双线X寄存器568选择X样式506[0xBAC3E]且将其施加到组合器572。基于Y索引值0xA9从双线Y寄存器570选择Y样式508[0xFF58]且将其施加到组合器572。将存储在MM寄存器571中的单一混合掩码(MM) 504[0xFFC0FF80]进一步施加到组合器572。组合器572根据MM 504将来自X样式506的适当位与来自Y样式508的适当位组合以产生原始指令502,原始指令502存储在高速缓冲存储器线580中且传递到执行管线550。连同XYM条目554一起存储在双线X寄存器和双线Y寄存器中的其它压缩指令(尤其是使用相同混合掩码504的情况下)可依据指令序列与XYM条目554的解压缩并行或以串行方式解压缩。解压缩可变长度指令的格式长度由与压缩指令相关联的混合掩码确定。

[0053] 图5B和图5C中展示的X和Y存储器内的个别样式的放置的特性是,放置不受约束且由最左字节对准控制。具有填补位的高利用率的可接受、最佳或近似最佳放置可潜在地使字节对准碎片最小化或消除。对符号放置的任何约束(例如,左侧调整对准要求)潜在地产生碎片。举例来说,在随机大小对象的字节对准存储中,最差情况情境为每所存储符号7位被浪费且除特定符号放置担保外不携带任何有用目的。一般来说在索引压缩的帮助下,且特定来说在使用填补位的情况下,这7位可用于存储有用数据。来自被浪费的位利用的存储

节省的上限为7位乘以存储装置中的符号数目。来自索引压缩的总节省一般来说较高,且自然仅受内容的信息熵限定。还可存在占据相同物理位且仅在存在适当MM(其隐含地确定符号长度)的情况下正确提取的不同大小的符号。如果使用单一MM,那么X/Y词典符号条目具有相同大小。如果利用多个混合掩码,那么可存在与存在的MM一样多的符号大小的变化。举例来说,可变长度指令格式可包含32位格式化指令和64位格式化指令。无论如何,符号的最佳放置的任务是NP完全问题且不能预期在合理线性时间内实际上完美解决。为克服此限制,使用启发式算法在线性时间内产生近似最佳放置,如下文参看图12-14更详细描述。

[0054] 压缩指令和未压缩指令可在同一可执行代码片段中混合,而不需要在操作中进行模式切换。在分页虚拟存储器系统中,存储器的页可含有压缩指令或未压缩指令。图6说明根据本发明具有压缩页和未压缩页的分页虚拟高速缓冲存储器组织600。虚拟地址通常以两个部分进行编码。地址位的上部字段通常表示基于选定页大小(例如,4K字节页)编码的虚拟页数。地址位的下部字段是识别所寻址页内的地址的页偏移。在虚拟到物理地址翻译中,虚拟页数翻译为物理页数。页偏移对于虚拟地址和物理地址两者来说是相同的且不进行翻译。

[0055] 虚拟到物理地址翻译系统可包含一个或一个以上翻译后备缓冲器(TLB),例如指令和数据TLB,以改进翻译过程的性能。指令TLB(ITLB)是存储最近的虚拟到物理地址翻译连同所存储页的属性(例如,条目验证和存取许可)的小高速缓冲存储器。ITLB通常包含与随机存取存储器(RAM)电路耦合的内容可寻址存储器(CAM)电路,且相对较小(例如,具有32或64个条目)。每一ITLB条目包含CAM电路中的标签,其具有与RAM电路中的所翻译物理页数目相关联的最近使用的虚拟页数。举例来说,分页虚拟高速缓冲存储器组织600使用ITLB 602和物理存储器604,所述物理存储器具有与压缩页608混合的未压缩页606。ITLB 602的每一条目具有虚拟地址标签610、条目旗标612(例如,有效(V)旗标、读取(R)旗标、写入(W)旗标)、物理页(P页)地址614和压缩(C)字段616。举例来说,C字段616可为适于针对一系统将页识别为压缩或未压缩的单一位,所述系统对于所有压缩页具有单一混合掩码。或者,C字段616可为两个或两个以上位,其对于2位字段可指示“00”未压缩、“01”以第一混合掩码压缩、“10”以第二混合掩码压缩,且“11”以第三混合掩码压缩。是否压缩指令或代码块的决策在编译和代码压缩时间静态地进行且可能取决于多种因素。举例来说,如果系统的实际实施方案无论如何都会对解码的等待时间敏感,那么应用的性能关键部分可能保持为原始未压缩形式,而较不频繁执行的代码可经压缩。频繁与不频繁执行的代码部分的确定由编译器和链接器206以任选简档定向反馈信息208或以编译时间启发式基于代码的控制结构进行。然而,解码器228在L2高速缓冲存储器224与L1高速缓冲存储器232之间的放置有效地将解码器从系统的性能关键路径移除。从解码器的此放置获得的额外益处包含执行管线234不需要改变,以及归因于增加的L2容量的潜在功率节省,且借此使对主存储器的存取最小化。这些压缩决策基于翻译工具216与编译器和链接器206的紧密交互,且可利用简档反馈信息208,简档反馈信息208可能识别(例如)将不压缩的传统代码和待压缩的新功能代码。

[0056] 图7说明系统700,其以每物理存储器页为基础在类似于图6的分页虚拟高速缓冲存储器组织600的系统中混合压缩和未压缩代码。针对具有4千字节(4KB)页(具有4字节(4B)条目)的实例主存储器物理地址空间展示压缩页701和未压缩页702的组成。压缩指令

的每一页和未压缩指令的每一页具有拥有相同固定数目的字节的相同容量。然而,存储在压缩页701中的指令的数目为存储在未压缩页702中的指令数目的两倍,其使用例如为未压缩指令格式中使用的位数的一半的压缩指令格式。举例来说,压缩页701存储压缩指令d、e、f、g等,作为16位或2字节的一对X/Y索引。因此,为4K页的压缩页701含有2048个指令,而未压缩页702仅含有1024个指令。由于页边界不受页保持压缩指令还是保持未压缩指令影响,所以页的物理位置的地址翻译不变,但压缩页保持更多个别指令。

[0057] 图8A说明根据本发明用于解压缩压缩指令页和存取未压缩指令页的分页解压缩器系统800。分页解压缩器系统800包括处理器管线802、ITLB 804、物理地址缓冲器806、L1 Icache 808、L2高速缓冲存储器电路810,和压缩指令解码器电路812。翻译过程通过将从虚拟地址803选择的虚拟页数805施加到CAM电路而开始,CAM电路通常进行所施加虚拟页数与同CAM标签819中的条目标签一起存储的所有所存储的最近使用的虚拟页数目的并行比较。如果存在匹配,那么CAM电路存取RAM电路中的对应条目820,其作为存储在物理地址缓冲器806中的经翻译物理页地址815而输出。经翻译物理地址809包括与来自虚拟地址803的页偏移817连接的经翻译物理页地址815。

[0058] 举例来说,在具有4千兆字节(4GB)和4K字节页的虚拟地址空间的内嵌式系统中,虚拟地址803包括具有位[31:12]的虚拟页数805和具有位[11:0]的页偏移807。在相同内嵌式系统中,高速缓冲存储器和主存储器的存储器层级可涵盖512K字节和4K字节页的物理存储器空间。在ITLB 804中的命中的情况下,虚拟地址803翻译为物理地址809。物理地址809包括具有位[28:12]的物理页数815和具有位[11:0]的页偏移817。在此系统中,虚拟到物理翻译系统将把以位[31:12]编码的虚拟页数805翻译为以位[28:12]编码的物理页数815。并且,在命中的情况下,还输出压缩位字段821以作为C位字段822存储在物理地址缓冲器806中。压缩位字段821和C位字段822的放置是示范性的。

[0059] 物理地址809用于搜索L1 Icache 808以寻找匹配的条目。如果找到匹配的条目,那么其是与匹配的条目相关联的解压缩指令且经选择以经由L1/L2多路复用器824传递到处理器管线802。在L1 Icache 808中的未命中的情况下,物理地址被导向L2高速缓冲存储器810以搜索匹配的条目。在L2高速缓冲存储器810中的命中的情况下(其中C位字段822指示来自压缩页的压缩指令),提取具有相关联匹配条目的压缩指令的线且将其存储在L2读取缓冲器826中。在L2高速缓冲存储器810中的未命中的情况下(其中C位字段822指示来自未压缩页的未压缩指令),提取具有相关联匹配条目的未压缩指令的线且将其存储在L2读取缓冲器826中。未压缩指令绕过压缩指令解码器812,且使其在L2读取多路复用器828处可用以便存储在L1 Icache 808中并经选择以经由L1/L2多路复用器824传递到处理器管线802。

[0060] 在L2高速缓冲存储器810中的命中的情况下,L2读取缓冲器826中的所提取的压缩指令在压缩指令解码器812中基于指示到达控制电路830的压缩指令的C位字段822来解压缩。下文参看图8B更详细描述解压缩过程。解压缩指令存储在解压缩缓冲器832中,其可由L2读取多路复用器828选择以供存储在L1 Icache 808中,且经选择以经由L1/L2多路复用器824传递到处理器管线802。

[0061] 如早先在图5C中说明,本机应用的压缩可指定多个混合掩码的使用。如果所要粒度下降到单一指令,如图5C中指示,那么每一X/Y索引对必须能够选择适当掩码。如果混合掩码可逐指令地选择,那么混合掩码的识别需要额外存储位,例如比如XYN条目554中展示

的2位混合掩码索引。举例来说,另一方法允许每代码页选择一混合掩码,这消除了对来自X/Y索引对554的混合掩码标记的需要且将所述混合掩码标记作为C字段的一部分放置在TLB页描述符中。

[0062] 图8B说明示范性解压缩状态图850,其说明用于在处理器管线上执行的L2高速缓冲存储器压缩页指令和L1 Icache解压缩指令的状态。图8B展示L2高速缓冲存储器852、L2读取缓冲器854、压缩指令解码器(CID)856、解压缩缓冲器858、L1 Icache 860、物理地址缓冲器862和处理器管线864。L2高速缓冲存储器852具有未压缩线866和压缩线868作为初始状态。未压缩指令是32位本机指令,且压缩指令中的每一者为由X索引和Y索引对组成的16位,如上文例如参看图4描述。压缩线868保持未压缩线866所保持指令的两倍的指令。对指令M的提取由处理器管线864进行,指令M由ITLB(例如,图8A的ITLB 804)从虚拟地址翻译为物理地址,且存储在物理地址缓冲器862中。初始,L1 Icache 860不含有指令M,且产生未中,这引起到存储器层级的上部层的提取。首先检查解压缩缓冲器858,在此示范性情境中未找到指令M。接下来检查L2高速缓冲存储器852,且发现压缩线868含有压缩形式“m”870的指令M。提取压缩线868且将其存储在L2读取缓冲器854中。CID 856接收C位字段872和提取地址从而允许从压缩指令“m”870开始循序地提取来自L2读取缓冲器854的压缩指令。注意,可存取提取序列中的第一指令,首先解压缩且快速转发到处理器执行管线以使定时要求最小化。由于如本文描述的压缩指令在解压缩之前具有固定长度和已知存储地址位置,所以所提取序列中的第一压缩指令可在压缩指令的线中识别。由于L1 Icache具有八个未压缩指令的线长度,所以压缩指令从L2读取缓冲器854在L1 Icache线长度上存取,从“m、n、o、p”开始,接着为“i、j、k和l”。接下来存取L2读取缓冲器854的第二半,从“q、r、s、t、u、v、w和x”开始。因此,从L2读取缓冲器854中的任意开始点存取压缩指令。

[0063] 由于首先提取指令M,所以其在CID 856中首先解压缩且接着可转发到处理器管线864。当其它指令“n、o、p接着i、j、k和l”被解压缩时,其与解压缩指令M组合且加载在解压缩缓冲器858中。一旦整个指令线已解压缩,其就加载到L1 Icache 860中。或者,个别解压缩指令或解压缩指令对例如可个别地在L1 Icache 860中更新且视需要转发到处理器管线864。应注意,如果存储在解压缩缓冲器858中的解压缩指令线在L1 Icache 860中复制,那么保持解压缩指令线的解压缩缓冲器可用作临时存储从L1逐出的最后指令块的牺牲品高速缓冲存储器。

[0064] X索引、Y索引、X样式、Y样式以及一个或一个以上混合掩码的确定取决于本机指令架构和特定代码片段中指令的使用。当分析代码片段(例如,智能电话中的代码片段)时,单一混合掩码可提供与经由使用多个混合掩码获得的压缩可比较的压缩。单一混合掩码的使用可视为使用单一指令群组且有效地将代码片段中使用的本机指令分离为两个部分,X样式和Y样式。群组中的固定(X)和易变(Y)部分的原始区别变得不太重要且可互换地看待。并且,单一混合掩码的使用简化了组合器电路的设计,例如图4的组合器418。举例来说,在具有16,542个本机32位指令的实验代码片段中,仅存在6,464个唯一指令和总共0.76的熵。使用单一混合掩码,这6,464个指令分裂为1,351个X样式和1,345个Y样式。使用本发明的技术,16,542个本机32位指令中的每一者由23位X/Y索引对代替,从而提供例如图2所示的系统中的L2高速缓冲存储器和主存储器的存储容量的28.1%压缩。经编码数据的所得熵从0.76增加到0.89。然而,理论上可能的1.0信息熵量度中的0.89熵测量值说明,不同混合掩

码可提供明显较好的结果。确定最佳或近似最佳单一混合掩码以提供多个代码片段上的最佳可能压缩是一个复杂的问题。

[0065] 图9A-9C说明根据本发明基于一般算法的变型的自动混合掩码确定过程900。例如图2A的翻译工具216等翻译工具实施过程900。在框902处,过程900开始。在框904处,获得待压紧的代码片段P。举例来说,代码片段P可包含启动代码、操作系统代码以及如特定产品中利用的多个功能程序。在框906处,代码片段P中的唯一指令收集在P哈希表中(P_unique)。在框908处,P_unique指令以语义次序(例如,位样式升序)基于指令格式而分类。举例来说,P哈希表中的每一指令使用32位解译为整数且在列表中以升序分类。此保证了相邻指令之间的最小汉明距离且实现稍后步骤中的指令分组。在框910处,以唯一指令的分类次序收集位切换频率。举例来说,如果位置p中的位从“1”改变为“0”或从“0”改变为“1”,那么bit_toggle[p]值递增1。bit_toggle阵列大小等于以位表达的原始ISA中的最大指令的宽度,例如对于32位指令为32。在框912处,使用若干已知的良好MM、若干随机产生的MM和若干从所收集位切换计数产生的MM选择种子混合掩码的集合(种子MM群集)。已知的良好掩码是从来自类似应用的先前获得的MM预先填充而成且特定针对任何给定架构。所收集的位切换计数充当估计位在实际代码中的给定位置中改变其值的概率的良好启发式。依据经验,根据给定位置的阈值而选择混合掩码会产生良好种子掩码。在框912处,种子MM群集是10与30个个体之间的大小的个别MM的阵列,且可选择为偶数数目的个别MM。较大群集可产生较好结果,但所选择的MM的数目的线性增加需要计算时间的对应增加。可选择特定种子MM群集以配合特定计算平台。20MM的种子群集值已展示为实验设置中的良好折衷。在框914处,种子MM群集中的每一选定MM的配合度值(例如)设定为低但为非零值。所述配合度值为例如表示MM提供的益处水平的双精度浮点数字。过程900继续到图9A的连接符A 916,其是到图9B中的连接符A 916的链路。

[0066] 在图9B中,含有过程900的重复部分。提供20MM的种子群集作为到过程900的输入。在决策框920处,作出关于是否将发生另一反复的确定。将阈值与数值退出条件比较,且如果未达到退出条件(如阈值小于或等于退出条件时的情况),那么开始另一反复。

[0067] 在框918处,确定具有最佳配合度值的个体为最佳个体,且记录其配合度值(成本函数)。并且,记录确定为群集中的个体的成本函数的总和的total_fitness的值,其从图10的过程1000获得,如下文更详细描述。total_fitness值稍后用作变异算法的参数,从而基于较快改变速率合乎需要的假设在总体压缩效力为低的情况下引起较高变异速率。在决策框924处,作出关于完成过程900的进度的确定。如果最后反复的最佳配合度值与先前反复相比未改变,那么过程900进行到框926。在框926处,增加阈值。如果最后反复的最佳配合度值不同于先前反复的最佳配合度值,那么过程900进行到框928。在框928处,依据配合度值将群集分类。在决策框930处,作出关于以决策框920开始的过程900是否是在第一反复时的确定。如果确定为第一反复,那么过程900进行到框932。否则,过程900在MM的当前群集的情况下进行到框934-952以产生新的群集。

[0068] 在决策框934处,作出关于是否设定精英指示符的确定。精英指示符指示是否将保留多个最佳等级MM。如果设定精英指示符,那么过程900进行到框936。在框936处,将来自先前反复的两个最佳执行者MM直接复制到新的群集中。在框938处,从两个最佳执行者产生两个后代MM,如下文参看图9C更详细描述。在框940处,使两个后代MM随机变异以产生两个经

变异后代MM,如下文参看图9D更详细描述。在框942处,当前群集的前四个成员由两个最佳执行者MM和两个经变异后代MM代替。过程900接着进行到框946。

[0069] 返回到决策框934,如果未设定精英指示符从而指示未选择精英,那么过程900进行到框946。在框946和948处,如框938处描述的杂交和如框940处描述的变异的类似程序针对来自先前反复群集的剩余个别MM的未经处理对重复。在框946处,使用搭档选择算法选择随机个体,选择几率与个体的配合度成比例。较好执行的个体更有可能(但不保证)产生后代。在框948结束时,确定新的反复群集。在框950处,新反复群集中的任何副本变异以确保新反复群集中不存在副本。在框952处,重置新反复群集中的个别MM的配合度值。在此点处,新反复群集准备好进行成本计算。为了节省计算时间,维持先前处理的MM(个体)的先前所计算成本的列表,且简单地预先填充而不计算(在框932中)已确定的成本。过程900接着进行到图9C上的连接符B 922。

[0070] 返回到框938处的培育操作且参看图9B,选择两个最佳执行者MM作为图9D中的母A 960和母B 962,其中混合掩码(MM)由32位数字表示,且出于此步骤的目的,MM表示为基因。对于框946处实施的图9D的培育过程959,从输入的MM群集选择两个个别MM,且其也可表示为母A 960和母B 962。在框938和946处,如图9D所示在随机频率下在基因中的随机点处经由交换基因过程964培育两个母代,从而产生两个后代A 966和B 968。随机选择可由伪随机数产生器引导且可称为半随机选择。一个母代(母代A 960)提供一个后代(后代A 966)新基因的开始部分970,且另一母代(母代B 962)提供所述后代基因的尾部部分971。另一后代B 968是第一后代的补充。产生后代A 966的尾部部分971的母代B 962产生后代B 968的开始部分。产生后代A 966的开始部分970的母代A 960产生后代B 968的尾部部分。在范围0-32内随机选择交叉点972。快速算法收敛的潜在原因可视为归因于从母代的一者到后代的一者中的成功适应特性的保留,其稍后通过配合率计算确认。图9D中说明的培育过程959本身是伪随机事件且受制于识别为CROSSOVER_RATE阈值的阈值。举例来说,在两个母代之间交换基因的100次尝试中,仅将发生CROSSOVER_RATE乘以交换尝试次数。

[0071] 返回到图9B的框940和948,新产生的后代MM群集通过半随机翻转个别位而变异,从而在每一后代基因中将“0”改变为“1”或将“1”改变为“0”。此变异过程974在图9E中说明,其中个别MM A 975通过与位976、977和978相关联的位翻转而变异。以位翻转的结果展示经变异的个别MM A 979。在变异过程974中,伪随机变异用以将先前不可用特性引入到MM群集中。举例来说,产生新的先前未尝试的MM,但在可控速率下,使得可快速删除不需要的变异。在框942处,将MM群集四个成员设定为来自框936的所保留的两个最佳执行者和来自框940的两个新的经变异后代。在框946处,剩余MM群集成对培育,如参看图9C的过程955描述,以获得数对新的后代。在框948处,遵循图9D的变异过程974以产生两个新后代。在框950处,使任何副本MM变异。在框952处,重置每一MM个体的配合度值。在框932处,再使用重复MM的先前所计算成本。过程900接着进行到连接符B 922,其是到图9C中的连接符B 922的链路。

[0072] 在图9C中的框955处,当前个别MM的成本函数计算为选定MM和所分类P_unique列表的动态压缩函数。成本函数包括两个主要分量:外部存储装置的大小和翻译词典的大小。经组合的这两个分量反映了所获得的压缩程度或所实现的熵水平。外部存储装置大小与内部X/Y存储器解压缩表大小的确切关系由指示两个分量的相对重要性的参数确定。举例来说,成本函数等于x和y索引大小总和乘以X/Y词典大小(以千字节计)的乘积。因此,成本等

于 X/Y 词典大小 $\times(x_index_size+y_index_size)$ 。此公式反映评估内部词典大小与外部存储装置大小的相对重要性的平衡方法。可在成本函数中通过引入相对重要性加权常数而引入偏好。举例来说,成本等于 $(weight_constant \times X/Y_dictionary_size) \times (x_index_size + y_index_size)$ 。在当前实施方案中, $weight_constant$ 设定为1.00,但其可从0.01变化到1.00。

[0073] 在决策框956处,作出关于过程900的运行时是否已超过期限的确定。如果运行时已超过期限,那么过程900进行到框958且停止。如果运行时尚未超过期限,那么过程900进行到决策框957。在决策框957处,作出关于是否已找到最低当前成本MM的确定。举例来说,仅找到产生小于阈值的当前成本的单一MM可能是终止过程900的不充分原因。另一方面,有可能检测到已达到理论上最低熵界限表示,且不必要进行任何进一步搜索。在另一实例中,可找到若干MM,每一者产生与选择具有最低当前成本的MM或选择各自具有相同可接受最低当前成本的多个MM中的一者近似相同的小于阈值的当前成本(这可为在框958处终止过程900的充分原因)。否则,过程900继续到图9B的连接符A 916。

[0074] 图10说明根据本发明的示范性动态压缩和当前成本计算,作为过程1000。在框1002处,输入当前成本值,例如可从图9B的框952获得的MM,其在确定其先前在决策框932中未评估之后选择,且输入经分类的 P_unique 指令。在框1004处,根据针对选定MM的当前分组启发式基于具有代码片段的单一MM或多个MM而对经分类的 P_unique 指令分组。举例来说,针对单一MM的分组是将唯一指令分离为X和Y部分(例如,图5A的示范性元件500)以及构成初始未压缩X和Y表的过程。在一个实例中,整个本机指令程序可视为单一群组。另一方面,针对多个MM的分组涉及将唯一指令分割为若干群组,针对每一群组具有唯一MM。在此情况下,X样式表示群组的固定样式,其中整个群组共享单一固定X样式,且每群组的多个Y样式表示群组的易变部分。虽然对于到群组的近似最佳分割的若干启发式是可能的,但一个特定启发式的选择可由另一启发式驱动,所述另一启发式可基于唯一指令的输入列表的大小及其初始熵。以下启发式已通过实验证明是可行的,包含汉明、加权汉明、1的数目以及1的加权数目。

[0075] 图11说明用于当在经分类唯一指令列表中的相邻条目之间超过汉明距离的指定阈值时产生新群组的汉明启发式1100。经加权汉明启发式1100在计算与阈值相比的改变计时向某些位字段位置指派比其它位字段位置多的权数,如图11所示。在过程1100中的框1102处,获得经分类输入原始指令样式的完全列表(经分类 P_unique)。其接着通过分析两个相邻条目之间的汉明距离而同时处理所述两个相邻条目。在框1104处,将可变汉明距离值($hamm_dist$)计算为两个相邻条目的XOR。举例来说,如果原始指令的列表仅含有三个条目0、1和2,那么 $hamm_dist$ 计算两次,一次计算为第0与第1条目之间的XOR,且接着计算为第1与第2条目之间的XOR。在框1106处,以一次一个位的方式解析 $hamm_dist$,且针对每一非零位递增 sum_weight 值。框1106可视为对每一 $hamm_dist$ 中的非零位计数。递增量可依据位位置而改变。从 $weight_array$ 读取确切递增量,且又从图9A的框910中计算的 Bit_Toggle 阵列计算或静态地预先计算 $weight_array$ 。在决策框1108处,作出关于 sum_weight 值是否超过 $weight_threshold$ 的确定。肯定结果指示将形成新指令群组。在框1110处,形成新群组,且过程1100进行到决策框1112。决策框1108的否定输出致使过程1100进行到决策框1112。在决策框1112处,作出关于是否已到达经分类 P_unique 列表的末尾的确定。如果尚未达到列

表的末尾,那么过程1100进行到框1114,框1114递增循环变量“i”且进行到框1104。如果已到达列表的末尾,那么过程1100进行到框1116且停止。

[0076] 可使用其它启发式,例如1的数目启发式和1的经加权数目启发式,其两者均基于测量指令中的位字段的易变性。举例来说,MM中的较低1的数目指示每群组中较大X样式和较少Y样式。MM中的较高1的数目指示每群组中较小X样式和较多Y样式。存在对于用于选择产生最高压缩的1的数目的单一点的非线性相依性。在当前方法中,在框1004处选择单一MM分割策略。在框1006处继续,分别确定例如存储在图4的X存储器412中的X样式和存储在Y存储器414中的Y样式等X存储器和Y存储器表,如上文参看图4描述。在决策框1008处,使用过滤启发式滤出标准以下的MM,其表示不配合的个体,其在过程900中是可能的。进行此过滤以节省计算时间且不会影响过程1000的效力。如果从掩码产生的未压缩X/Y表的字节大小超过输入的原始指令列表大小,那么确定所述掩码为明显不配合。在决策框1008处,作出关于X存储器大小+Y存储器大小是否大于P_unique指令的数目乘以本机指令宽度(其在此示范性情况下为32位)的确定。在决策框1008处,如果所述确定是肯定的,那么过程1000进行到框1010以拒绝混合掩码。在框1010处,将1008处拒绝的MM的配合度值设定为零,且过程1000进行到图9C上的连接符B 922。如果所述确定为否定的,那么过程1000进行到框1012。

[0077] 在框1012处,压紧X存储器和Y存储器表。框1012直接影响X和Y表的大小,这又决定寻址这些表所需的索引的大小,且最终决定外部压缩代码片段的大小。X和Y表使用如参看图5B和图12-14描述的索引压缩而压紧。在框1014处,产生当前成本,其等于常数C1乘以X存储器大小加上Y存储器大小的总和,其接着乘以X索引大小加上Y索引大小的总和。此过程1000提供用于计算由当前成本表示的熵值的有利启发式,所述熵值接近现分离为X/Y索引和X/Y压紧表的外部集合的压缩代码的实际熵。任选地,在框1014处,可确定理论最小熵量度,以及是否需要进一步搜索。在框1016处,将所产生的当前成本输出到图9C中的决策框956。

[0078] 如上所述,X/Y表中的条目的近似最佳选择与索引压缩组合可显著减小X/Y表的大小。可基于压缩程度与执行时间之间的折衷使用两个示范性放置算法。第一启发式是根据本发明具有图12处呈现的填补分类的近似穷尽搜索X/Y表压紧算法。处理在框1202处通过获得待压缩的符号的列表而开始。此为待压紧的X/Y表符号的当前集合的列表。表中存在“table_size”个符号。在框1204处,以使填补重叠的概率最大化的次序将此列表分类。并且,在框1204处,将反复索引“i”重置到0。

[0079] 框1204处的填补分类操作是在实际压紧过程起始之前分类X/Y表中的符号以增加填补重叠的概率的指定方式。填补分类操作通过快速分类算法实施,其中在一个符号的一个或一个以上左侧最高有效字节与另一符号的一个或一个以上右侧最高有效字节之间存在匹配的情况下比较函数要求两个符号的交换。此处,符号的长度以完全字节测量,且左侧调整。右侧最高有效字节可仅具有若干有效位,但对于分类的目的这并不重要。

[0080] 在决策框1206处,作出关于反复索引“i”是否小于“table_size”的确定。如果反复索引“i”大于或等于“table_size”,那么过程1200进行到框1232且停止。否则,过程1200进行到决策框1208,因为X/Y表的第i符号尚未经处理。在决策框1208处,作出关于参数seen[i]是否等于“1”从而指示第i符号已经处理的确定。如果seen[i]=1,那么过程1200进行到框1210,其中反复索引“i”递增1。如果符号尚未经处理,seen[i]≠1,那么过程1200进行到

框1212。在框1212处,将第i符号插入到压缩表中。插入过程的细节在下文参看图13描述。同样在框1212处,通过将参数seen[i]设定为1、将best_score变量设定为0,且将新反复索引“k”初始化为值i+1而指出插入的事实。

[0081] 在决策框1214处,作出关于新反复索引“k”是否小于待压缩的符号数目的确定。如果新反复索引“k”小于“table_size”,那么过程1200进行到框决策框1216,因为第k符号尚未经处理。在决策框1216处,作出关于参数seen[k]是否等于“1”从而指示第k符号已经处理的确定。如果seen[k]=1,那么过程1200进行到框1218,其中新反复索引“k”递增1。如果符号尚未经处理,seen[k]≠1,那么过程1200进行到框1220。在框1220处,针对第k符号计算匹配计分。

[0082] 在框1220处,计分匹配计算模拟符号插入过程,但不插入任何符号。事实上,计分匹配计算返回与通过将当前符号以表的当前形式放置到压缩表中而预期的节省量成比例的计分。此计算过程在X/Y未压缩列表中的剩余未处理符号之间定位潜在最佳匹配候选者。计分匹配返回匹配指示或极高计分,例如无穷大(举例来说,如果符号可在现有内容内完全匹配),或提供若干重叠字节(如果符号在压缩存储器的末尾匹配)。如果未找到重叠,那么返回零计分。

[0083] 所计算的计分反映待通过将第k符号在其当前状态中插入到压缩表中而预期的节省程度。在决策框1224处,作出关于计分是否等效于无穷大的确定。如上所述,例如无穷大计分等匹配指示意味着第k符号与压缩表的当前状态完全匹配,将其插入不会增加压缩表的大小,且过程1200直接进行到框1228中的插入步骤。注意,在此情况下的插入过程是从压缩X/Y表的开始的当前符号偏移的记录,其稍后变为其在外部存储器中的表示。还应注意,一般来说,当使用多个混合掩码时,共享共同前缀的多个符号具有相同偏移,但不同长度。否则,在框1226处,在所计算计分与迄今最佳计分之间进行比较。如果新计算的计分超过迄今最佳计分,那么过程1200进行到框1230。在框1230处,当前最佳计分设定为等于新计算的计分,且最佳索引设定为“k”。在框1218处,新反复索引“k”递增1,且过程1200进行到框1214。一旦“k”到达列表末尾(table_size),如框1214处确定,那么过程1200进行到框1222。在框1222处,作出关于是否确定了best_score的确定。如果最佳计分尚未确定,那么反复索引“i”在框1210处递增,且过程1200从框1206继续。在框1222处,非零best_score指示找到最佳配合符号,且过程1200进行到框1228。在框1228处,将最佳配合符号插入到压缩表中,且通过将参数seen[best_index]阵列设定为等于“1”指出插入的事实。一旦反复索引“i”达到如框1206处确定的table_size,过程就在框1232处终止。在此点处,已将所评估符号插入到压缩表中。

[0084] 图13说明根据本发明的符号插入过程1300。在框1302处,获得压缩存储器的当前状态和待插入的新符号。在决策框1304处,作出关于新符号是否为零的确定。如果新符号为零,那么在框1320处将符号映射到压缩存储器中的指定位置,且仅存储一个字节而不管符号大小如何。在决策框1304处,如果新符号确定为非零,那么过程1300进行到框1306。在框1306处,计算当前pad_mask.pad_mask是用于掩蔽符号中的最后有效字节的非有效部分的二进制变量。举例来说,如果正插入的符号具有15位大小,那么最后字节的最后位非有效且在针对放置的比较期间需要忽略。此情况下的pad_mask是0xfe(pad_mask始终仅施加到最后有效字节)。在决策框1308处,作出关于反复索引“i”是否小于当前压缩表大小减去新符

号大小加上1的确定。如果所述确定为肯定的,那么过程1300进行到框1310。在框1310处,重复尝试将新符号放置在压缩存储器中,检查每一符号位置以便仅以字节步长将有效位与当前内容匹配。此操作通过将新符号的第一有效字节重复地与已在存储器中的每一字节比较来实现,且如果找到针对第一有效字节的匹配,那么新符号中的其它剩余位匹配。所述匹配仅考虑符号中的有效位,如上文所述经由施加pad_mask。如果在决策框1312处找到此位置,那么过程1300进行到框1318。在框1318处,将符号放置在所确定位置处,且反复索引“i”变为从压缩表的开始的符号偏移。此索引将最终存储在外部存储器中且用于将原始X/Y样式定位在X/Y压缩表中。如果到达框1316,那么这意味着当前符号可能尚未与压缩存储器的现有内容完全匹配,且其需要放置在列表的末尾。在所述情况下,对符号的每一字节执行最大重叠检查。如果未发现重叠,那么简单地将新符号放置在压缩表的末尾。

[0085] 图14说明根据本发明具有填补-分类过程1400的三角X/Y表压紧。过程1400是穷尽搜索X/Y表压紧过程1200的替代。启发式过程1400快于过程1200,但不能产生同样良好的压缩。然而,过程1400的压缩结果对于若干实际应用可为可接受的。此外,两个过程的组合可能在实际实施方案中可实现,图14所示的较快过程用于确定最佳掩码候选者,且图12所示的较慢过程用于产生最终编码。过程1400以与图12的步骤1202、1204、1206、1208和1212相同的步骤1402、1404、1406、1408和1412开始。在框1412处,执行与框1212处进行的类似的第i元件的插入,但以框1414开始,不存在与以图12的框1214和1222所进行相同的所使用的best_score。代替于使用最佳计分评估,在图14的剩余框处,i+1与table_size之间的符号被遍历一次,且在框1426处将被发现具有如决策框1424处确定的非零计分的每个元件插入到压缩表中。

[0086] 结合本文揭示的实施例描述的方法可体现在硬件的组合中,以及体现在存储由处理器执行的非暂时信号的软件模块中。软件模块可驻留在随机存取存储器(RAM)、快闪存储器、只读存储器(ROM)、电可编程只读存储器(EPROM)、硬盘、可拆卸盘、磁带、压缩盘只读存储器(CD-ROM),或此项技术中已知或将来将提出的任何其它形式的存储媒体中。存储媒体可耦合到处理器使得处理器可从存储媒体读取信息以及在一些情况下向存储媒体写入信息。耦合到处理器的存储媒体可为与电路实施方案成一体的直接耦合,或可利用一个或一个以上接口,从而支持直接存取或使用下载技术的数据串流传输。

[0087] 虽然在用于处理器中的说明性实施例的上下文中揭示本发明,但将认识到,所属领域的一般技术人员可依据以上论述和所附权利要求书采用广泛多种实施方案。

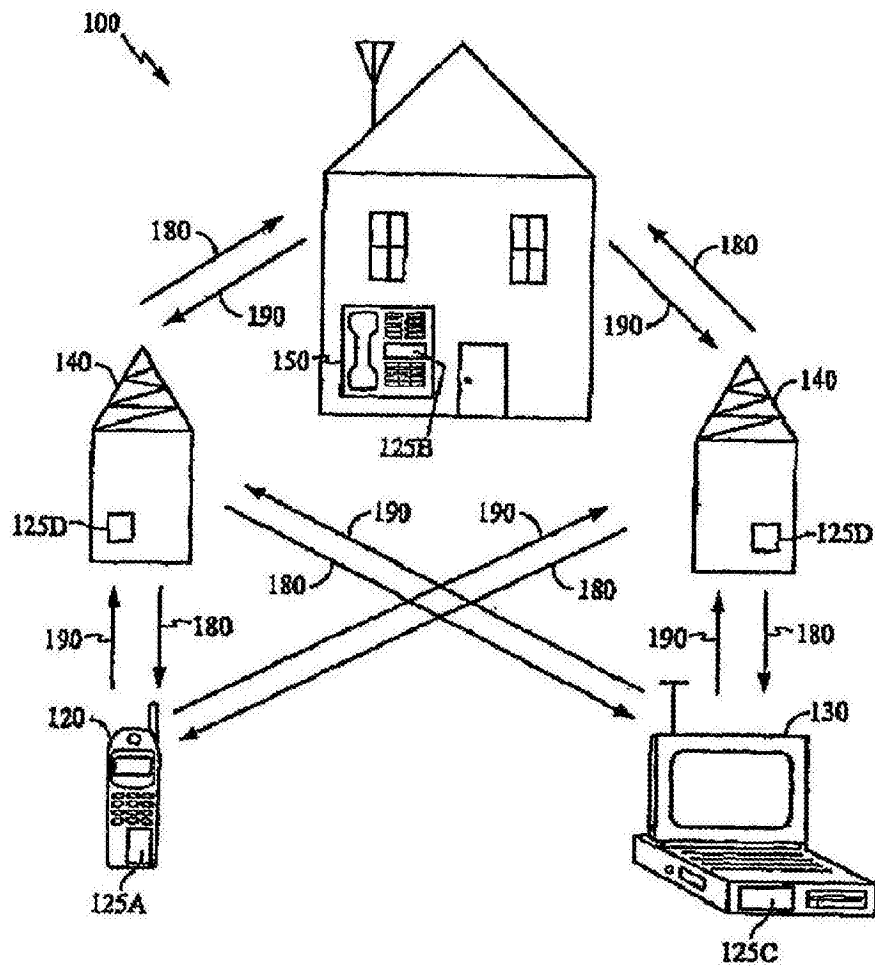


图1

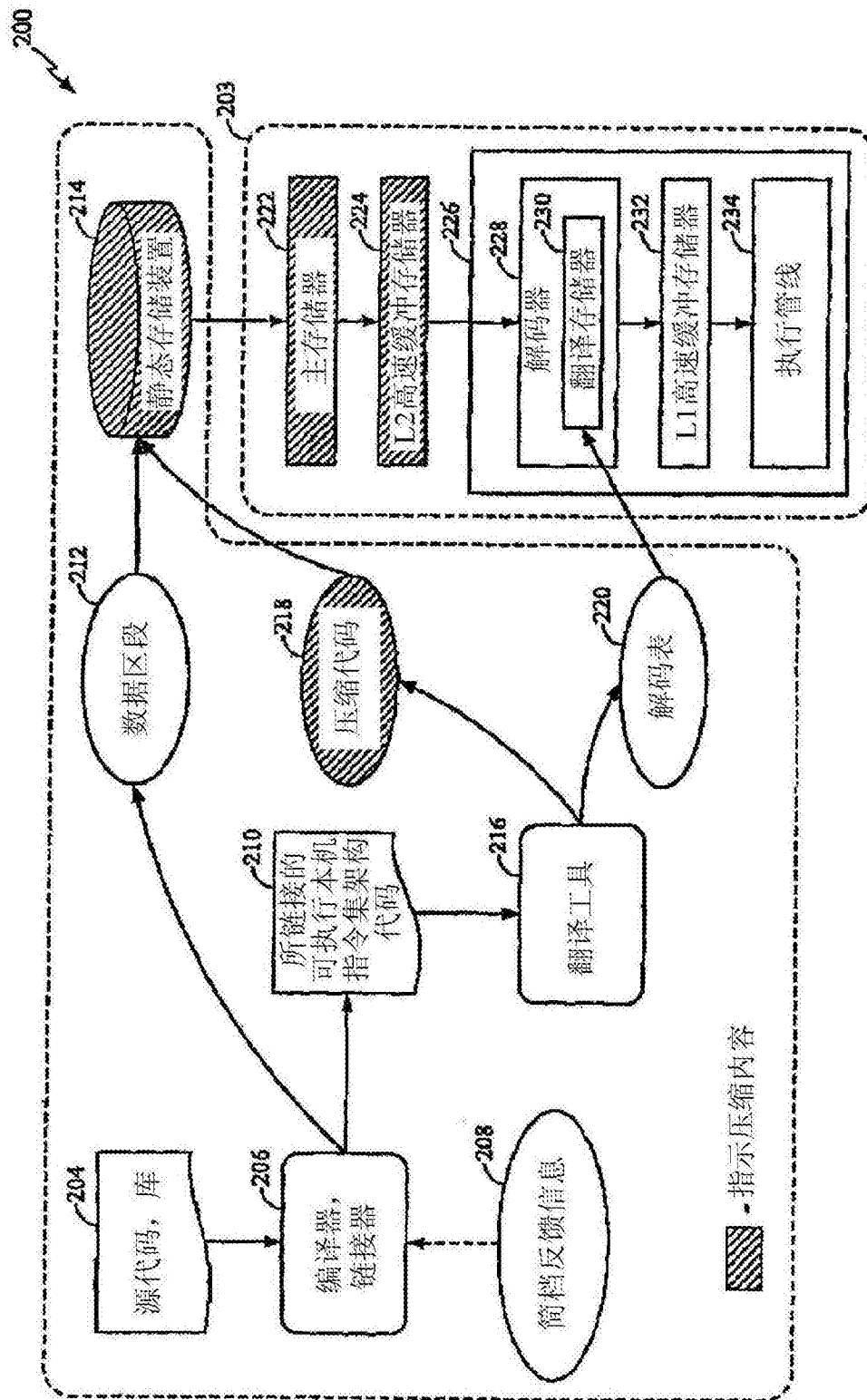


图2

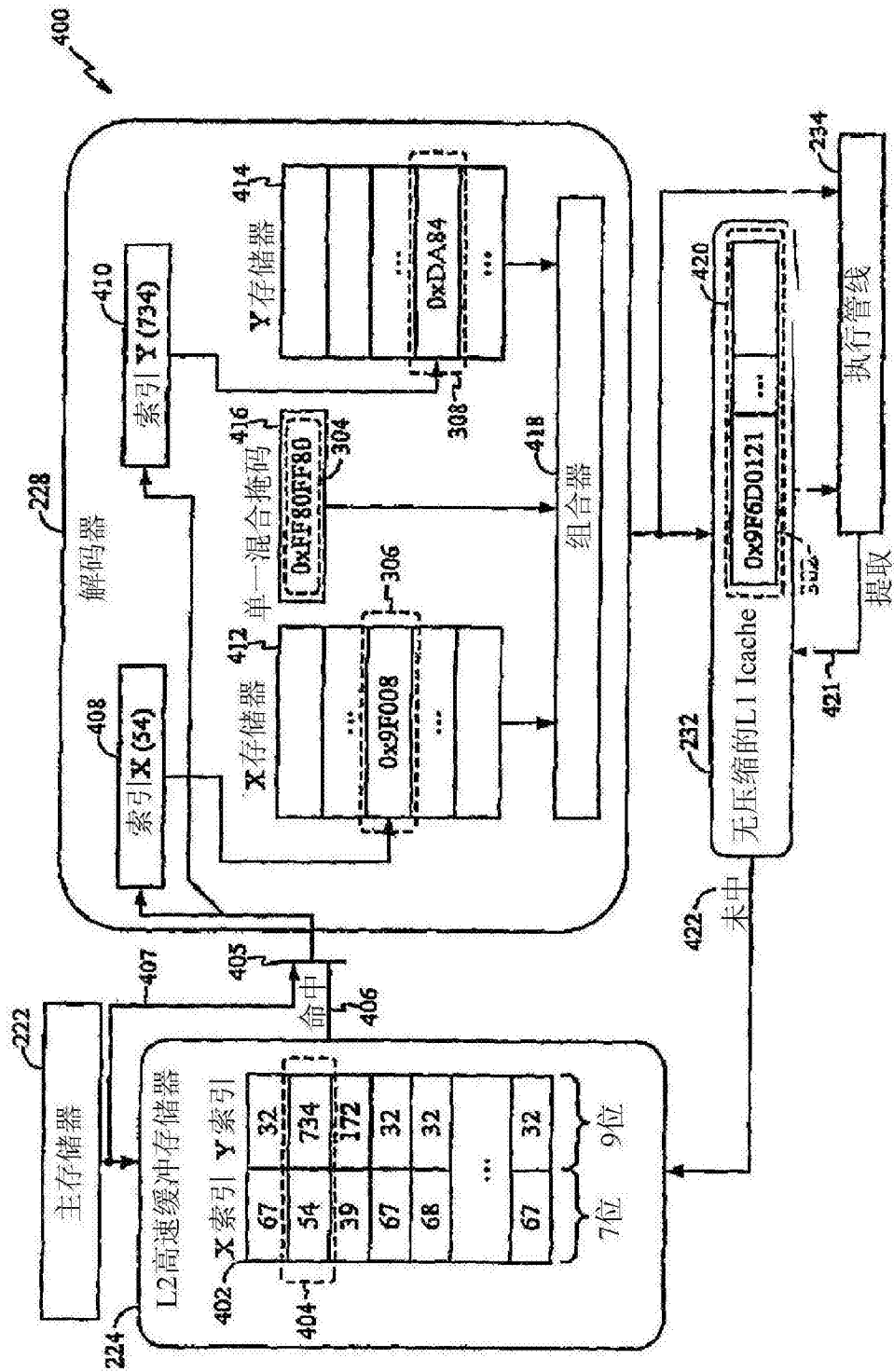


图4

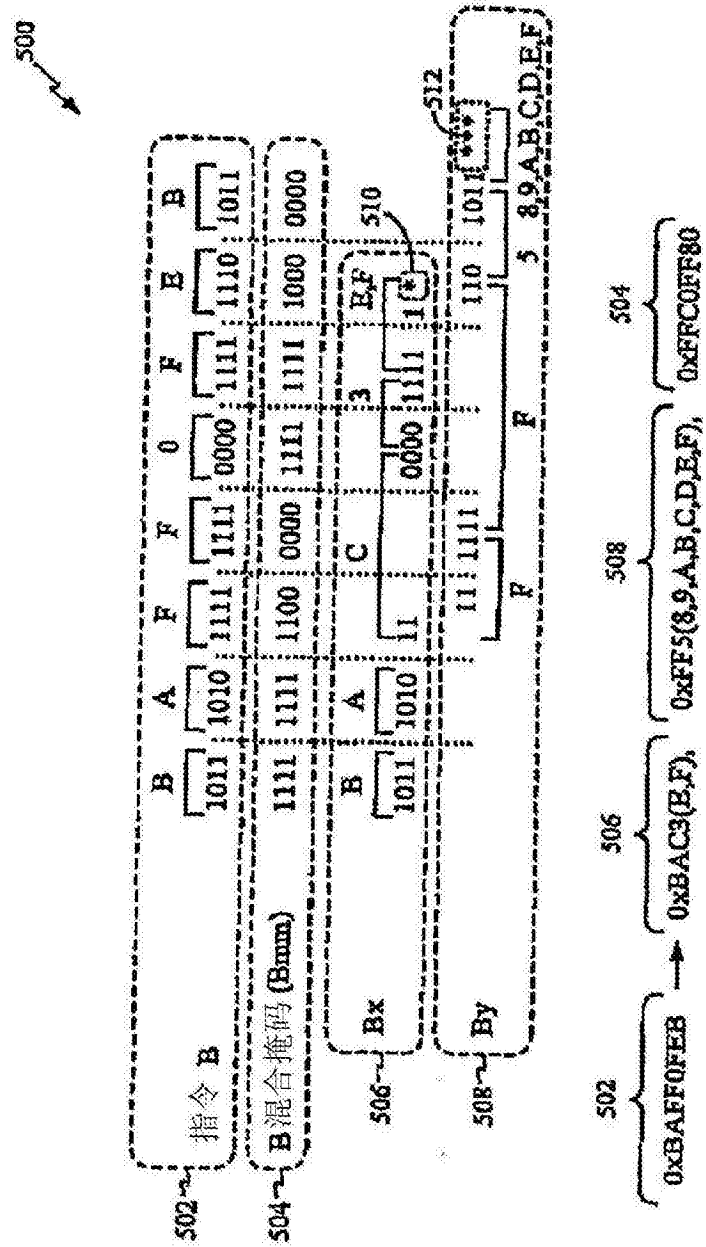


图5A

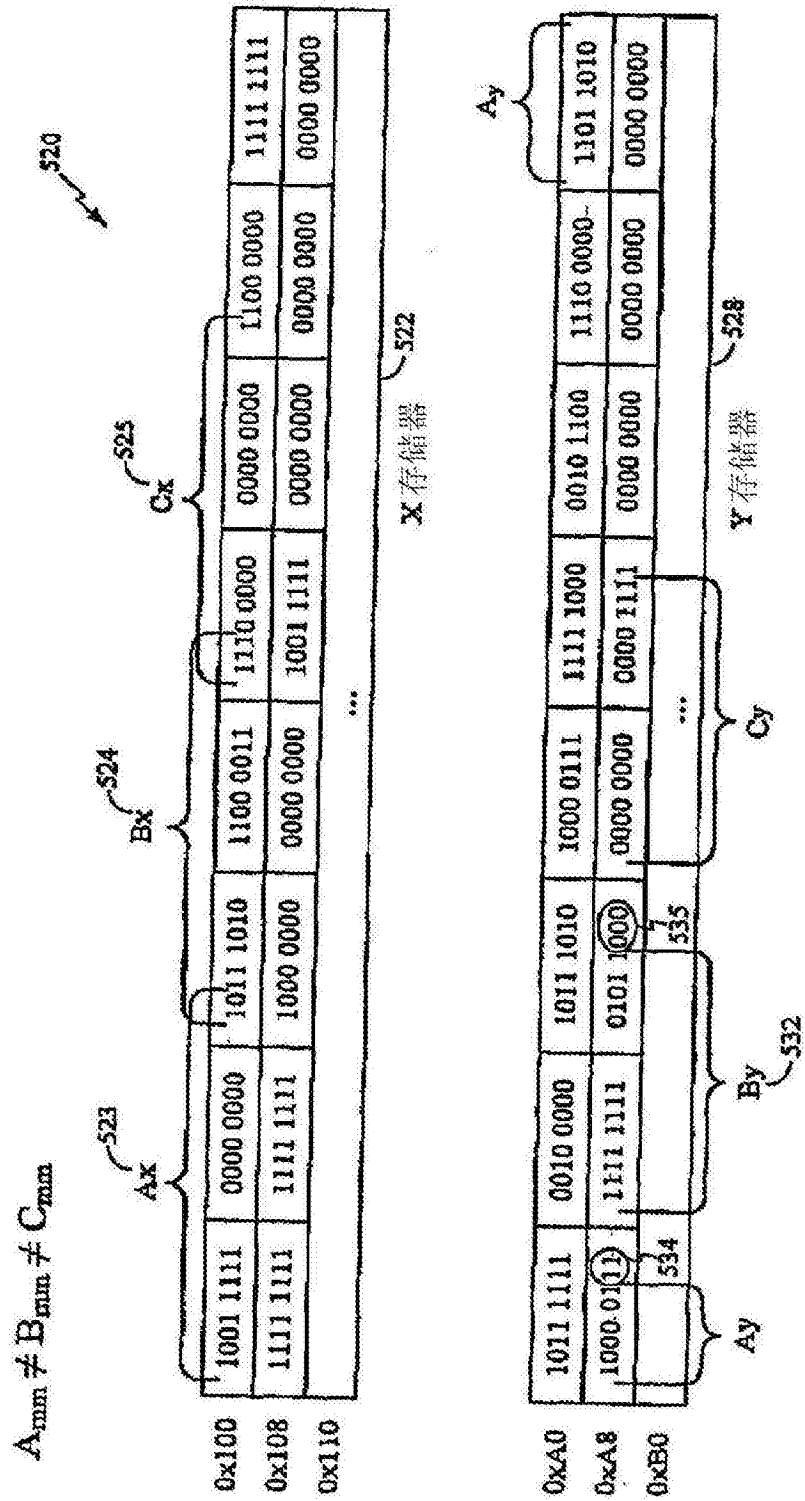


图5B

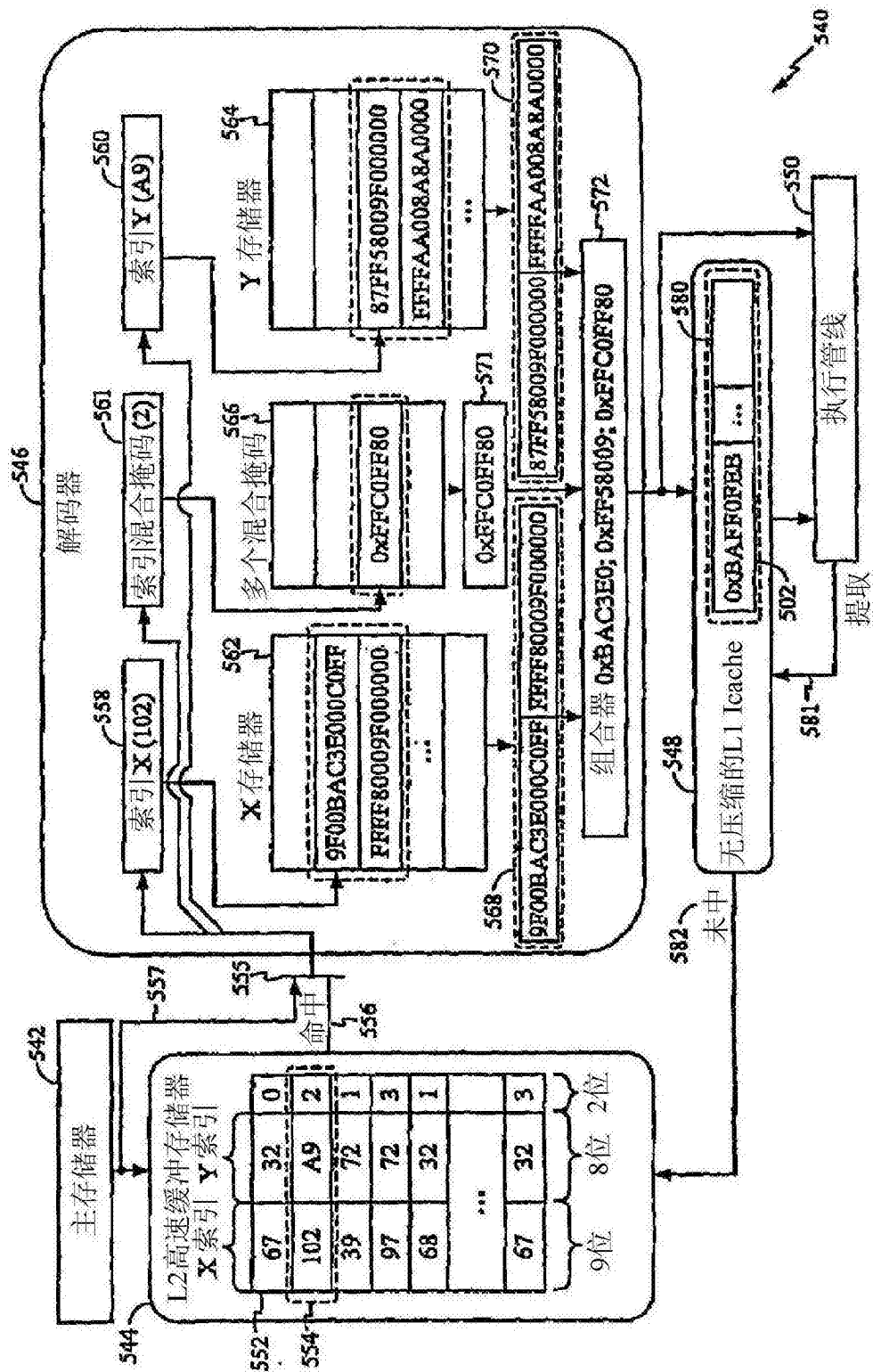


图5C

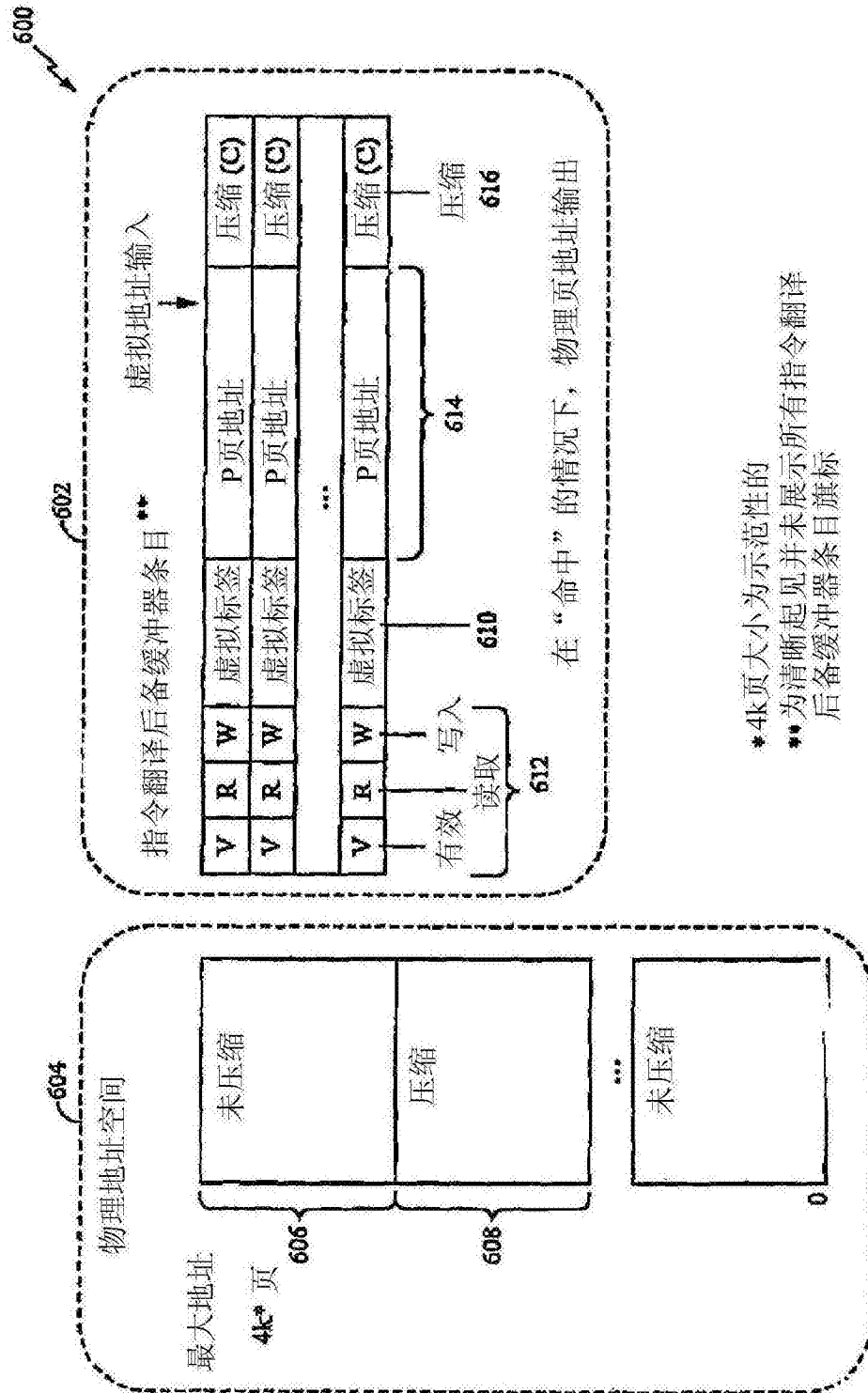


图6

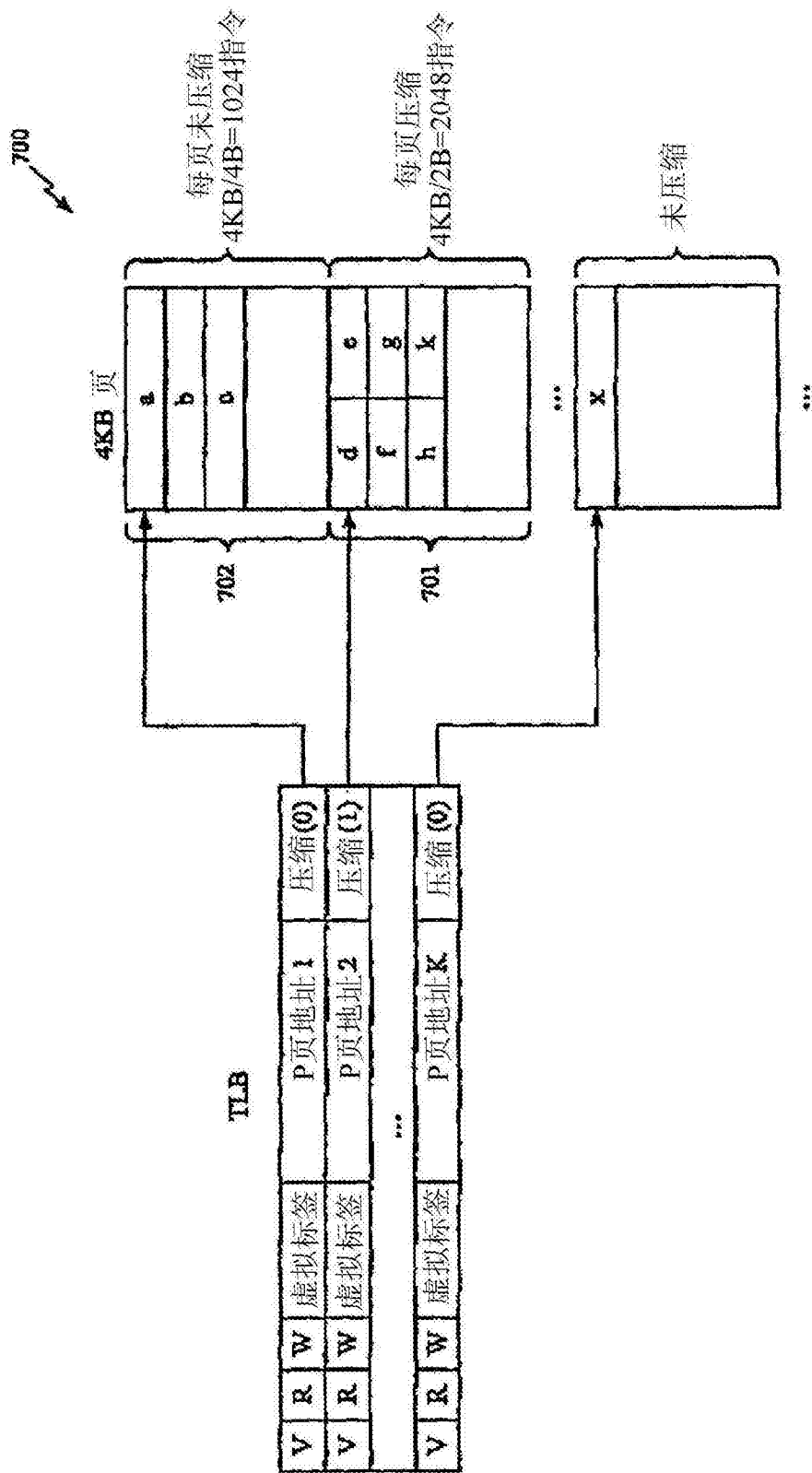


图7

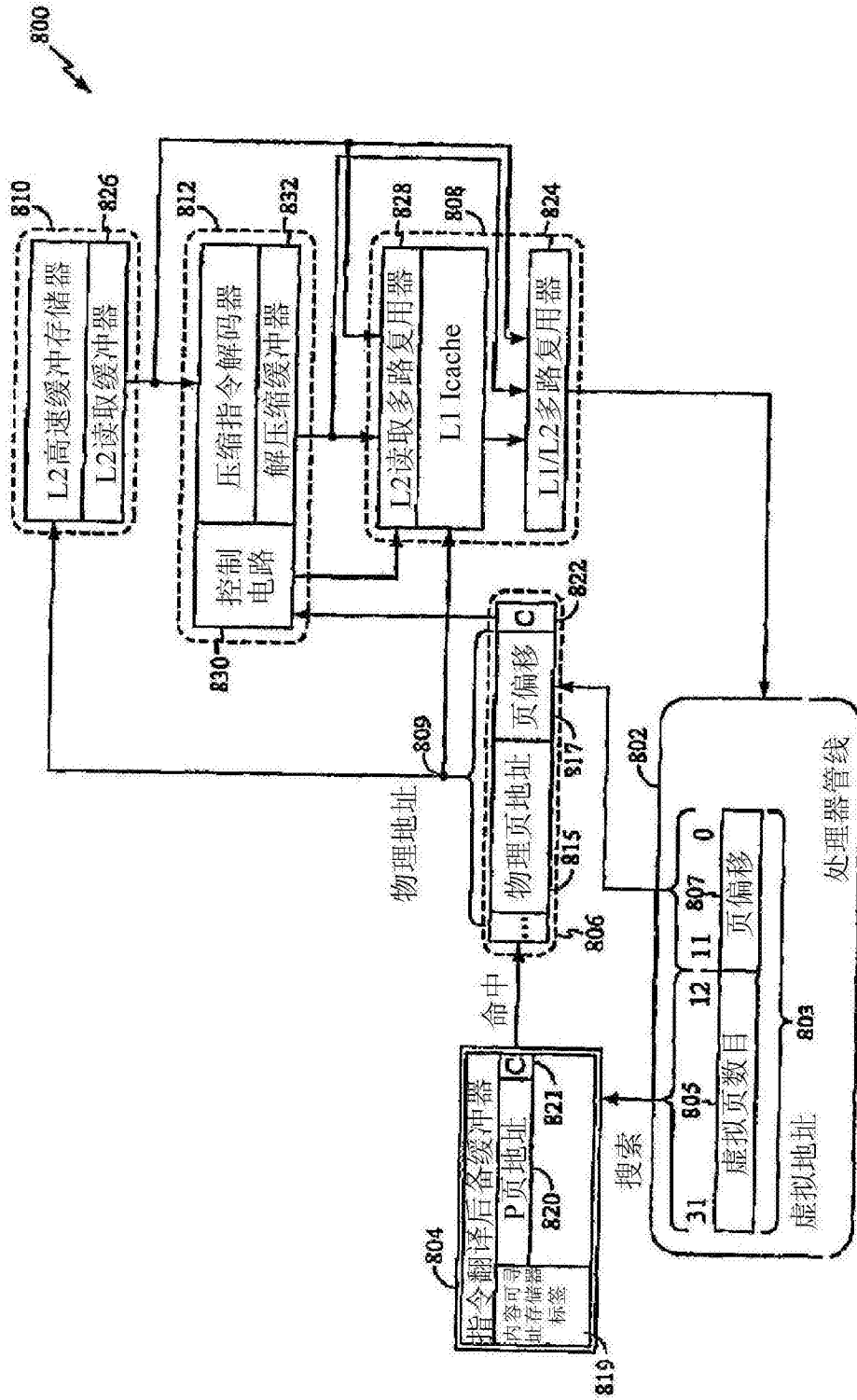


图8A

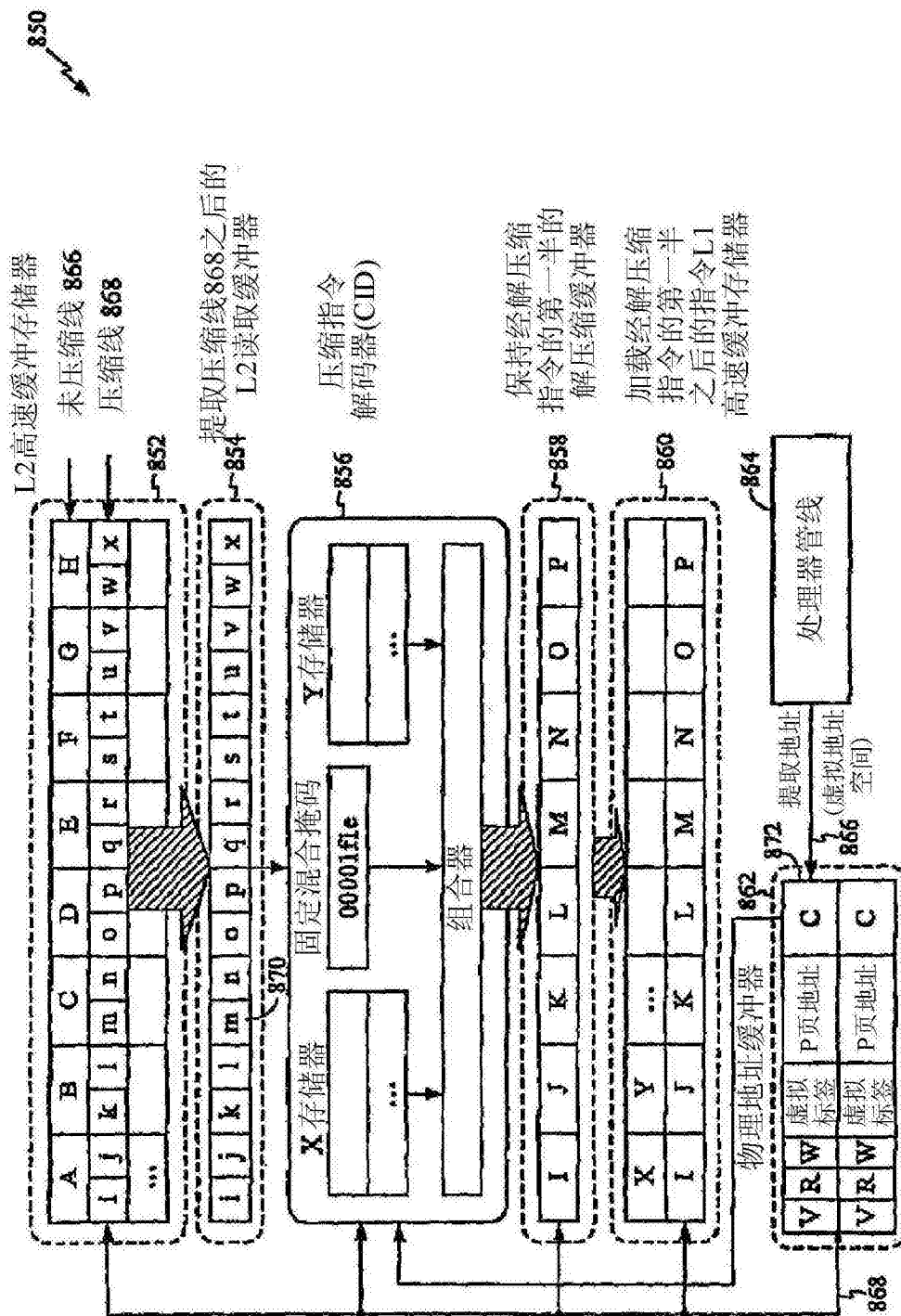


图8B

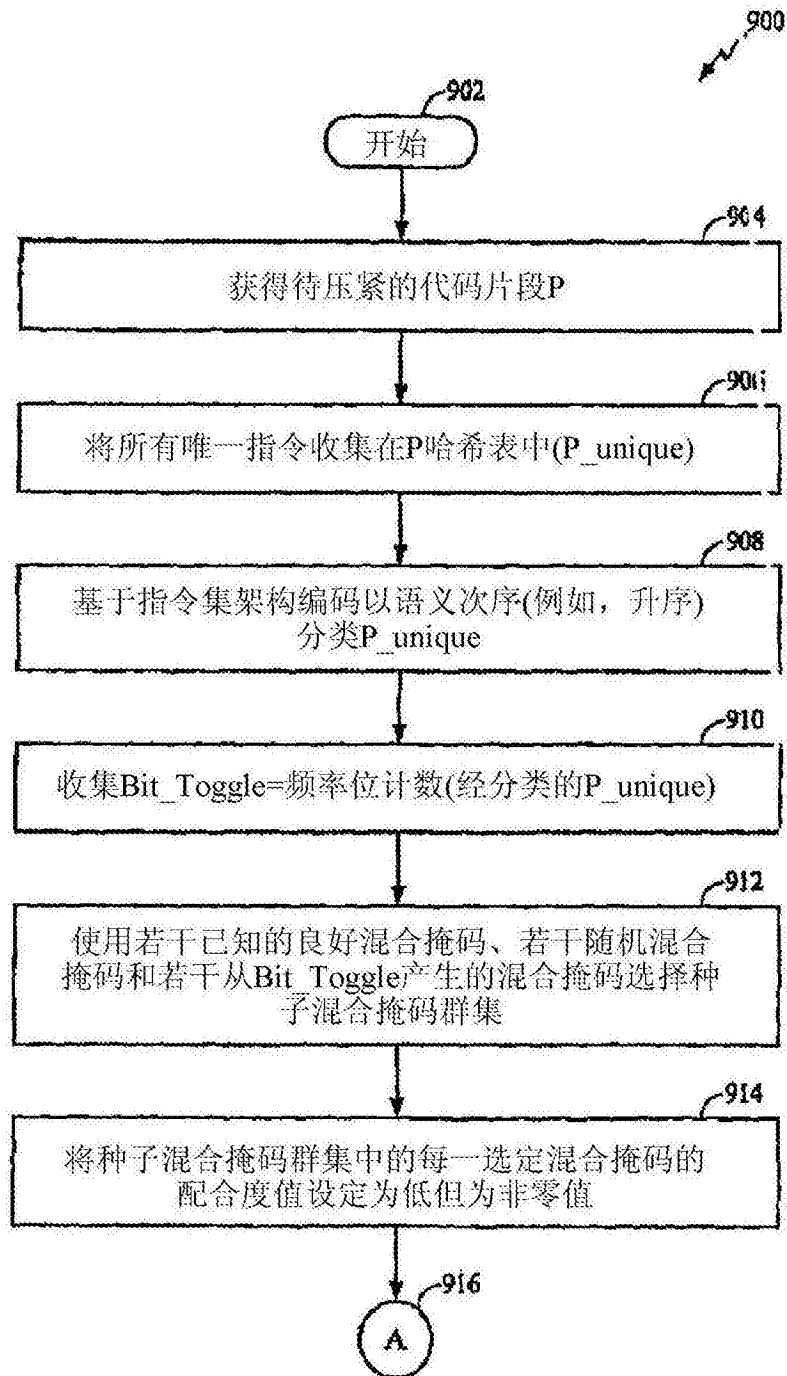


图9A

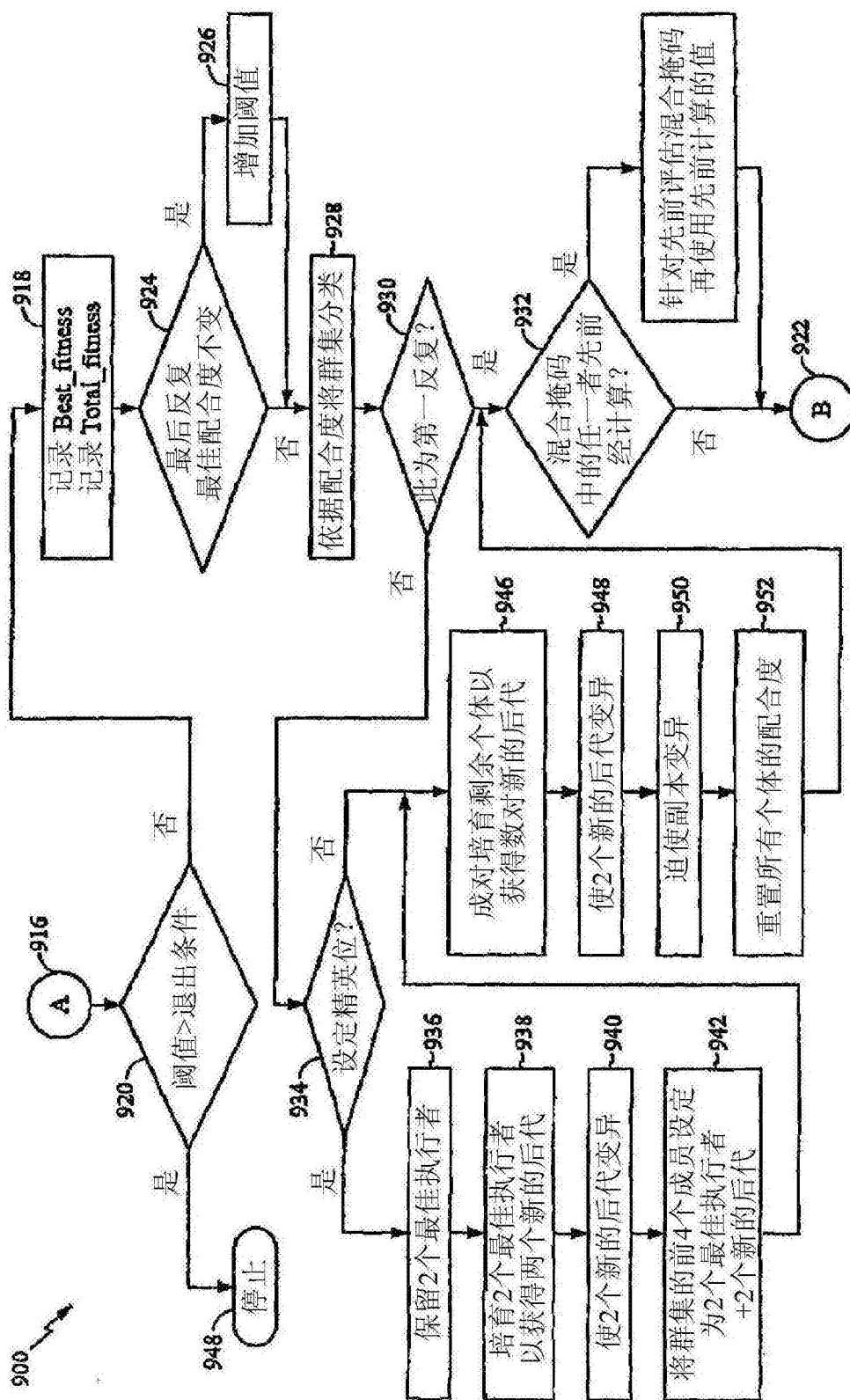


图9B

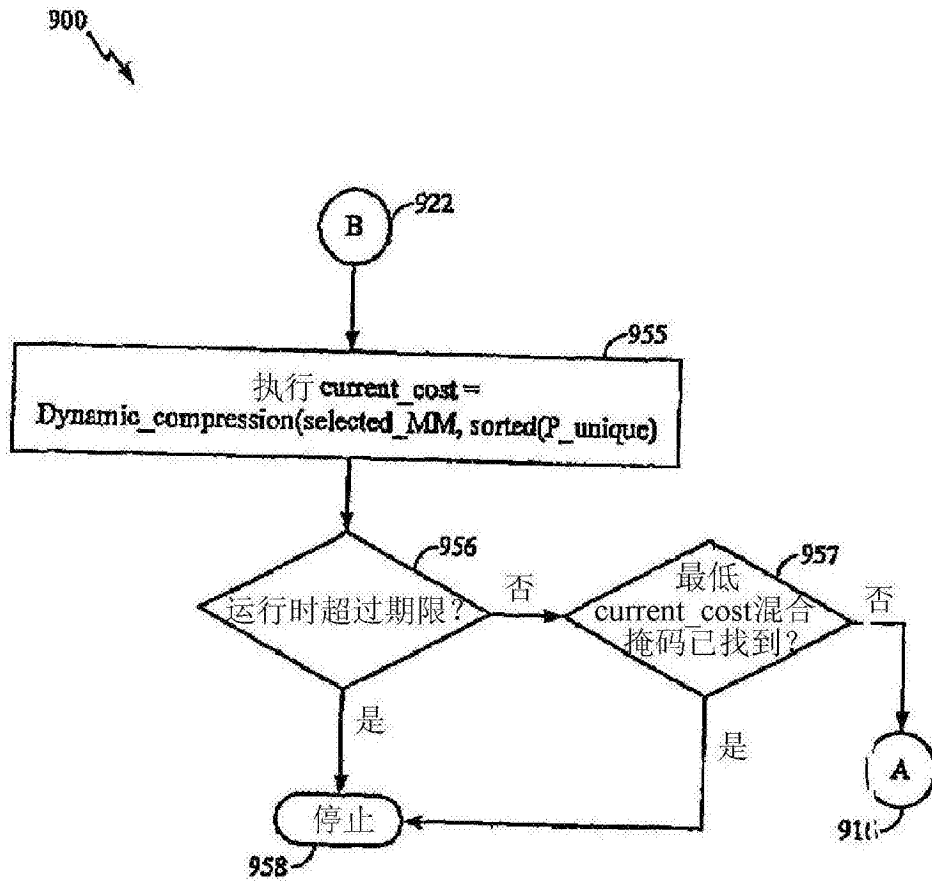


图9C

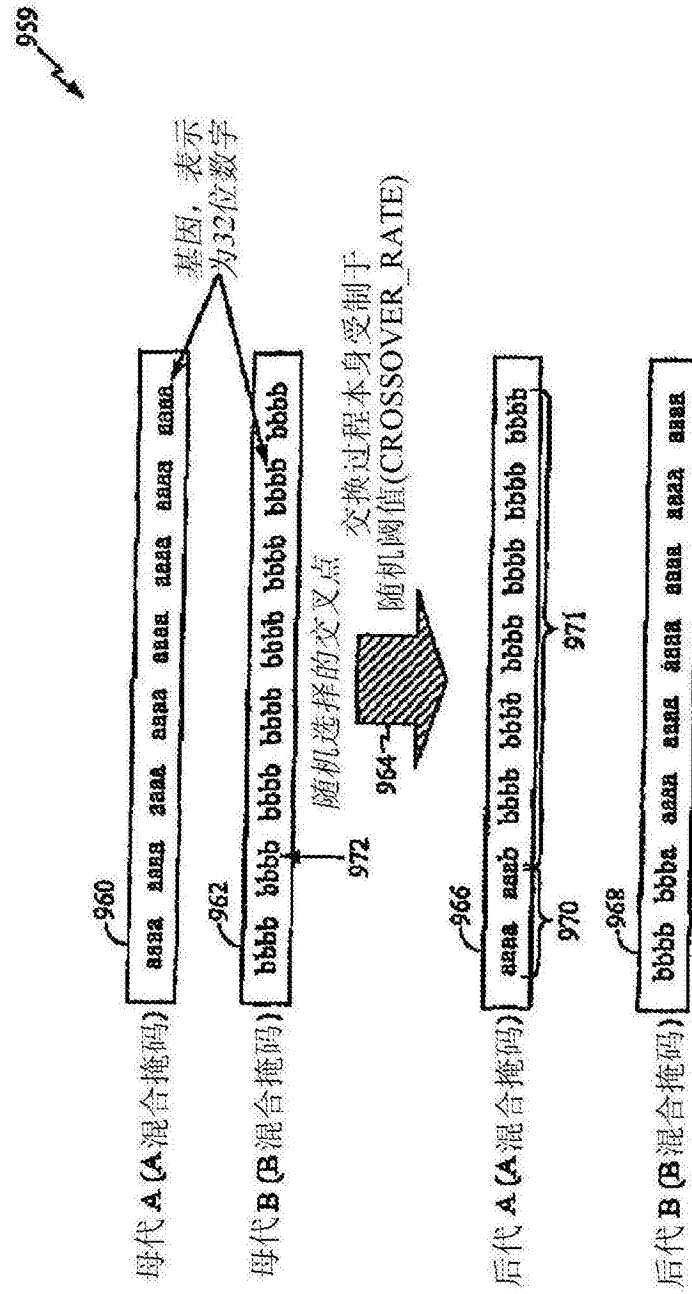


图9D

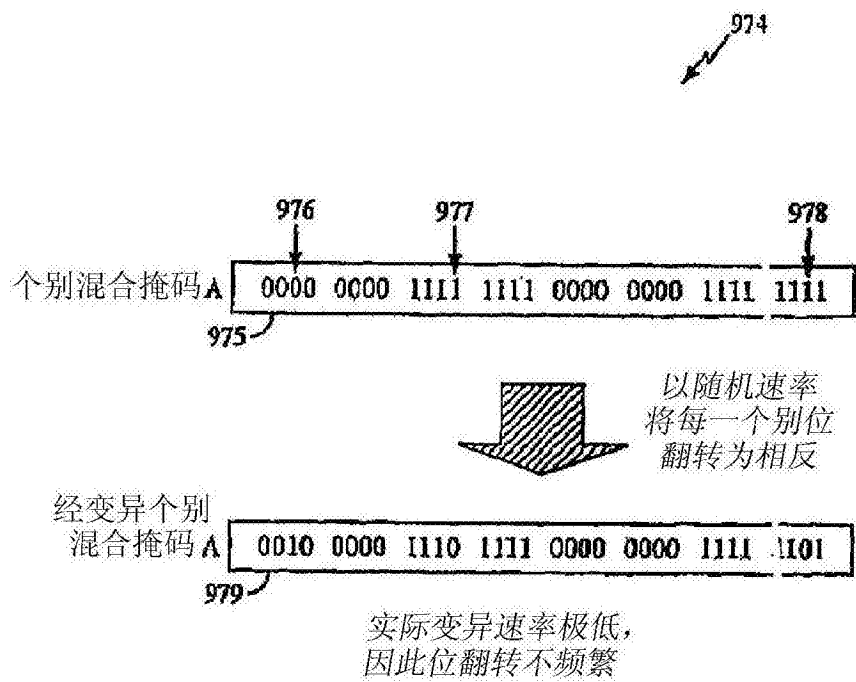


图9E

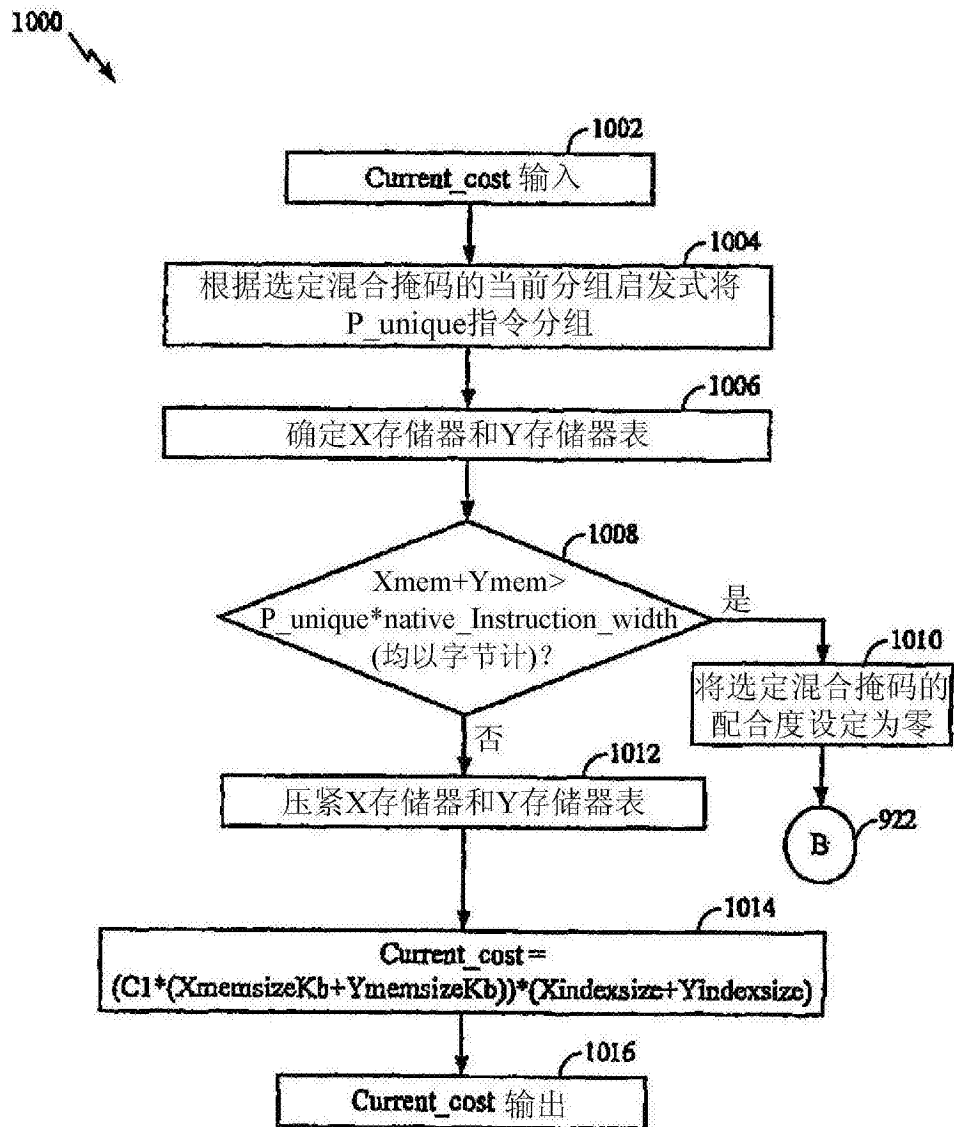


图10

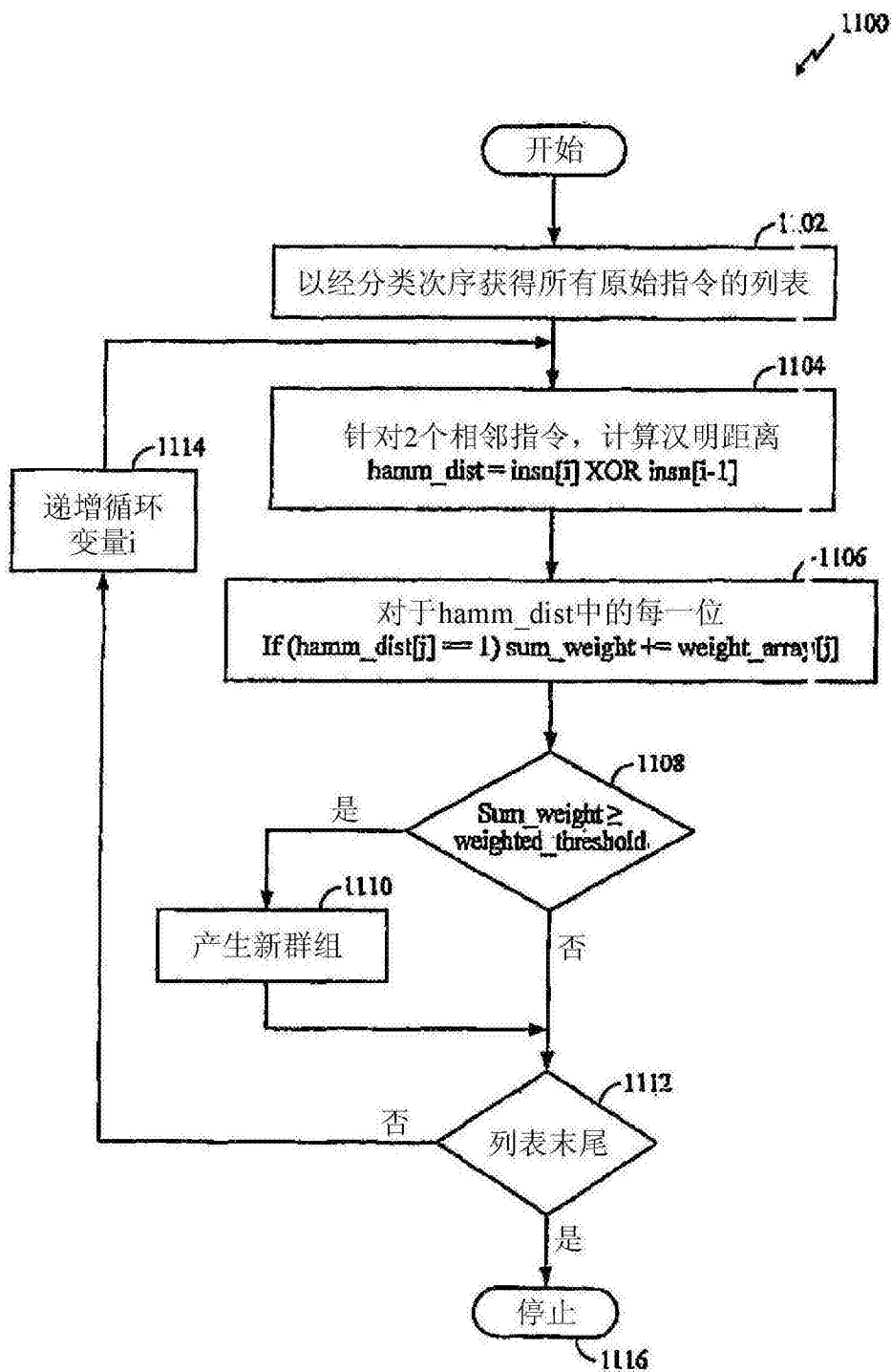


图11

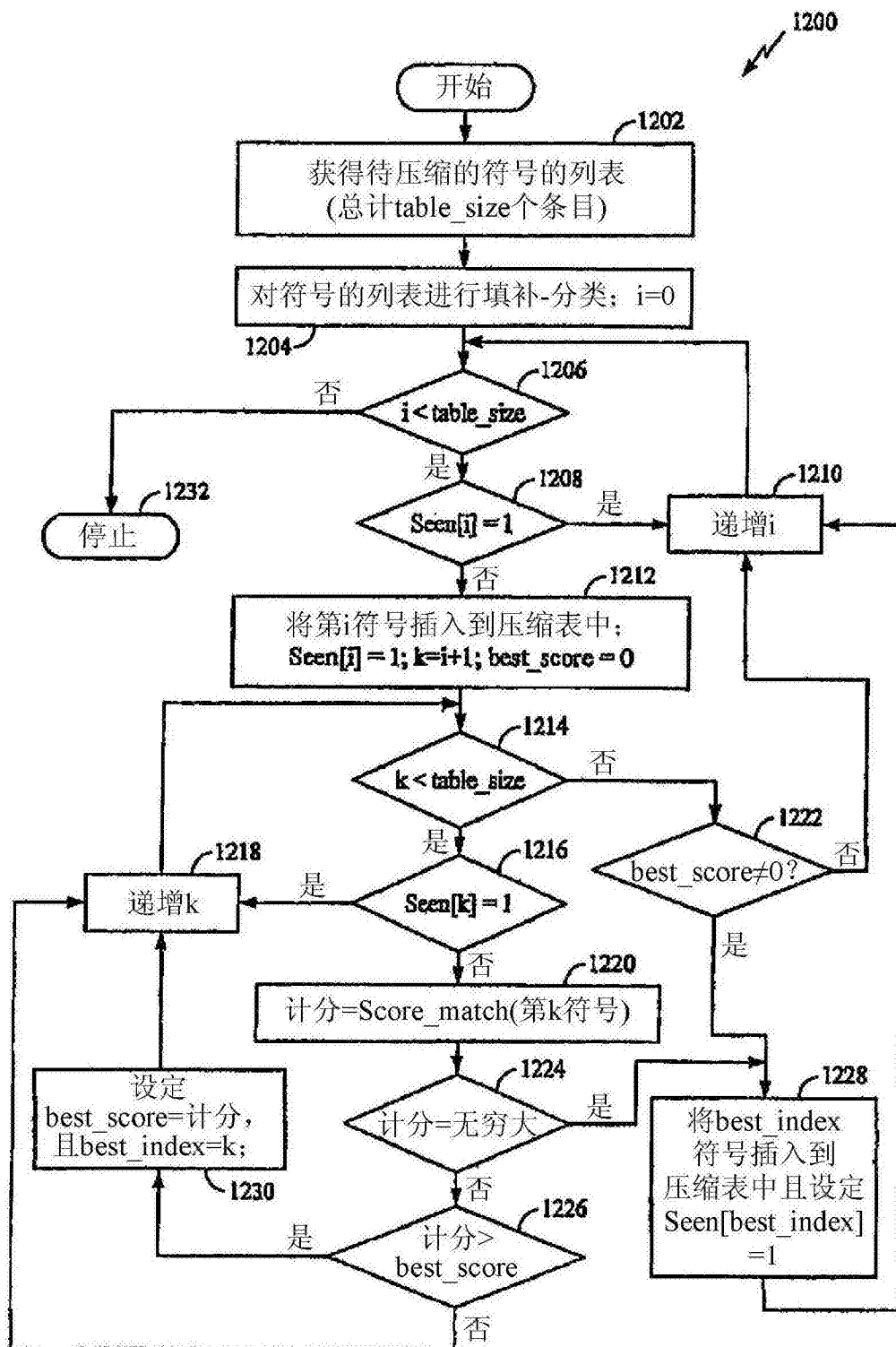


图12

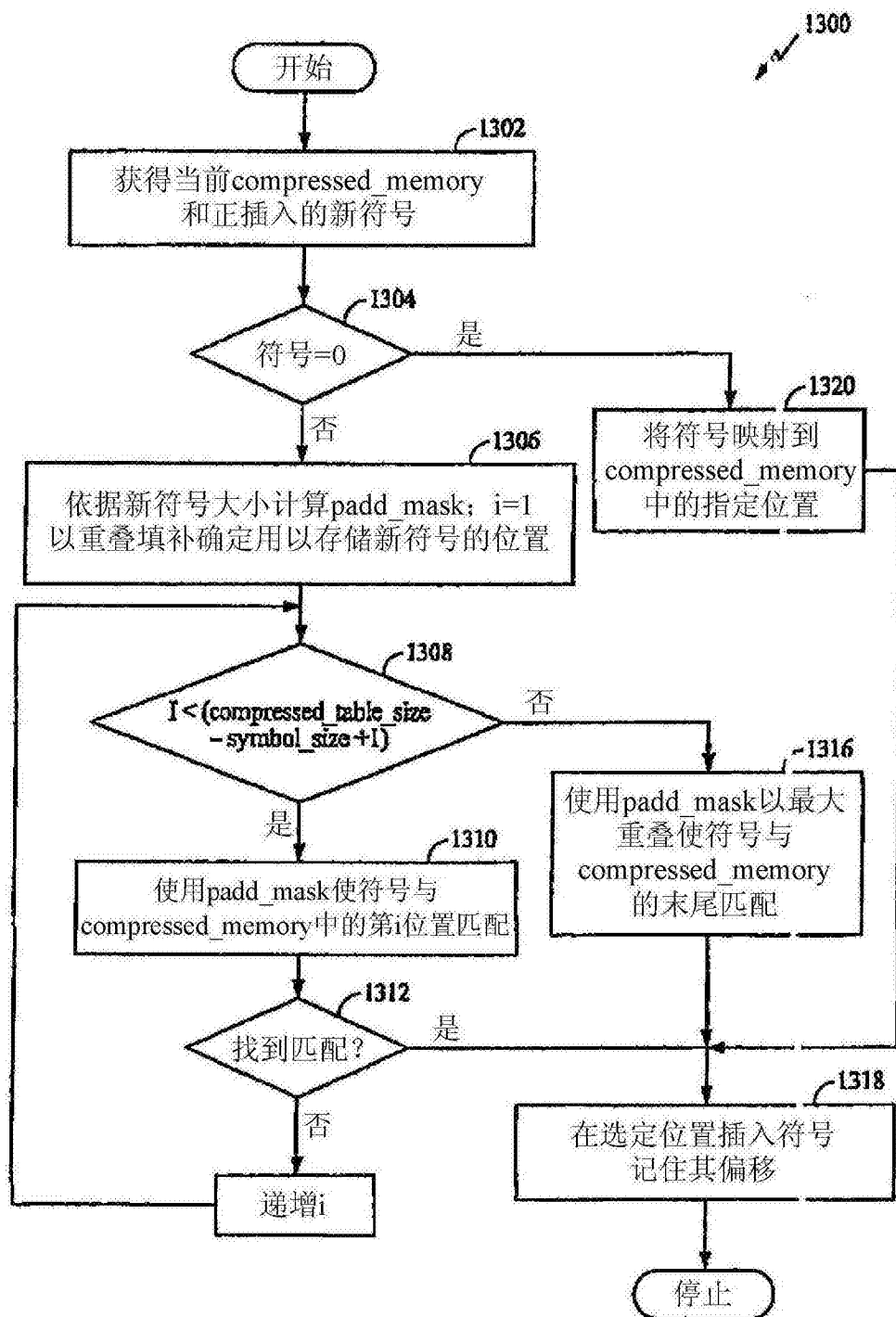


图13

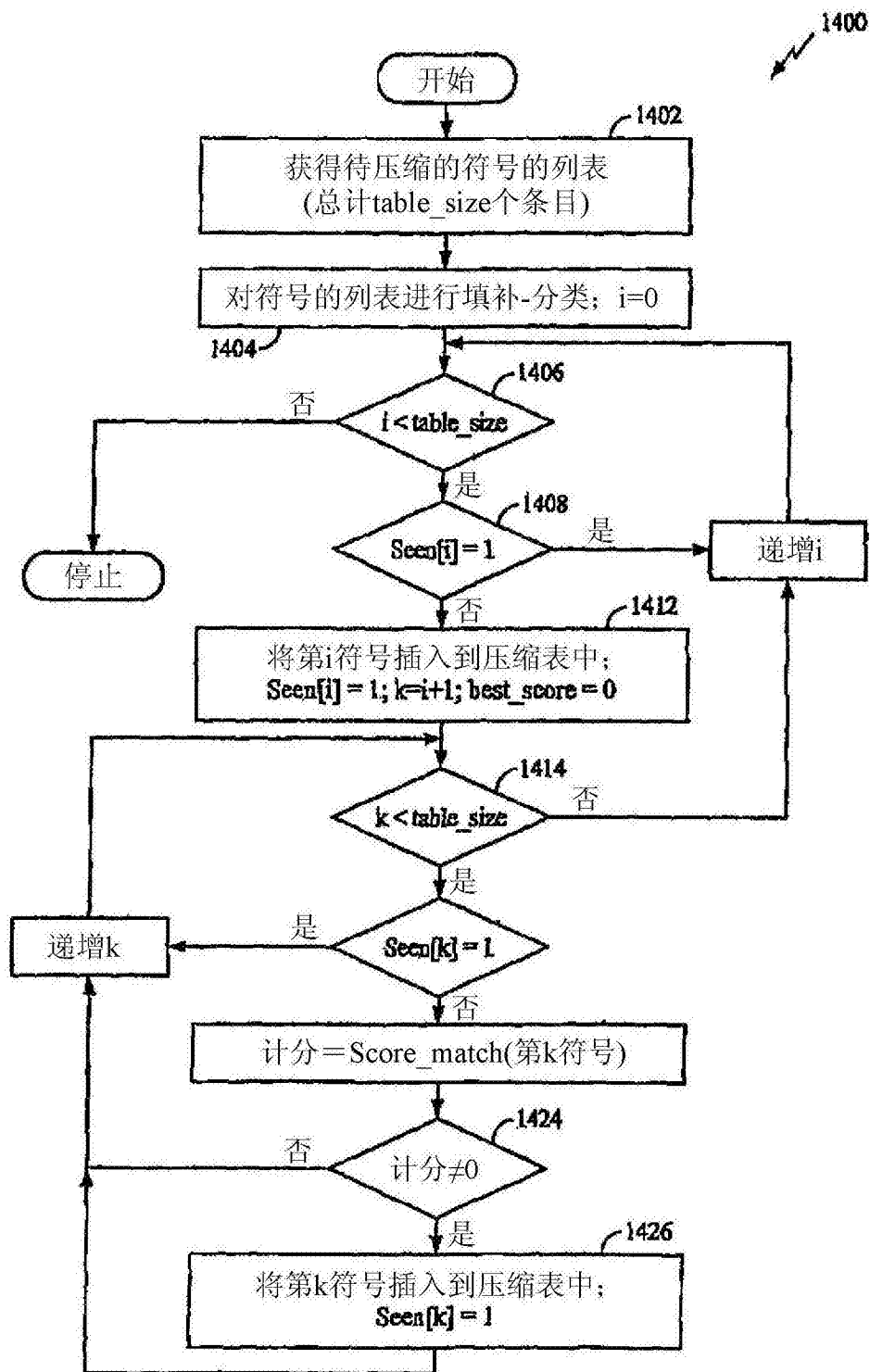


图14