



(12)发明专利申请

(10)申请公布号 CN 111181568 A

(43)申请公布日 2020.05.19

(21)申请号 202010028350.4

(22)申请日 2020.01.10

(71)申请人 深圳花果公社商业服务有限公司

地址 518000 广东省深圳市前海深港合作
区前湾一路1号A栋201室(入驻深圳市
前海商务秘书有限公司)

(72)发明人 刘必明

(74)专利代理机构 深圳市道臻知识产权代理有
限公司 44360

代理人 陈嘉琪

(51)Int.Cl.

H03M 7/30(2006.01)

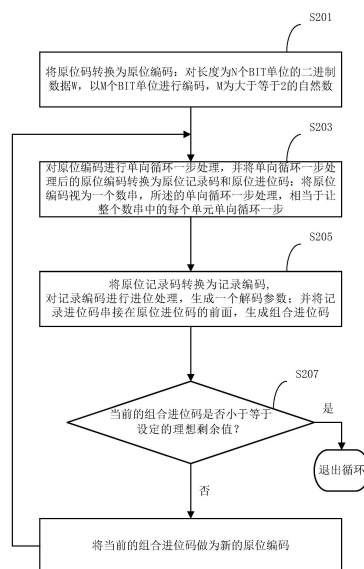
权利要求书2页 说明书10页 附图4页

(54)发明名称

数据压缩装置及方法、数据解压装置及方法

(57)摘要

本发明提出一种数据压缩方法,包括:将原位码转换为原位编码;对长度为N个BIT单位的二进制数据W,以M个BIT单位进行编码,M为大于等于2的自然数;对原位编码进行单向循环一步处理,并将单向循环一步处理后的原位编码转换为原位记录码和原位进位码;将原位编码视为一个数串;将原位记录码转换为记录编码,对记录编码进行进位处理,生成一个解码参数;并将记录进位码串接在原位进位码的前面,生成组合进位码;判断当前的组合进位码是否小于等于设定的理想剩余值,不是的话,将当前的组合进位码做为新的原位编码,进行循环处理;是的话,退出循环。实施本发明实施例,可达到压缩数据的目的,能够大大提升无损压缩的效率。



1. 一种数据压缩方法,其特征在于,包括以下步骤:

将原位码转换为原位编码:对长度为N个BIT单位的二进制数据W,以M个BIT单位进行编码,M为大于等于2的自然数;

对原位编码进行单向循环一步处理,并将单向循环一步处理后的原位编码转换为原位记录码和原位进位码:将原位编码视为一个数串,所述的单向循环一步处理,相当于让整个数串中的每个单元单向循环一步;

将原位记录码转换为记录编码,对记录编码进行进位处理,生成一个解码参数;并将记录进位码串接在原位进位码的前面,生成组合进位码;

判断当前的组合进位码是否小于等于设定的理想剩余值,不是的话,将当前的组合进位码做为新的原位编码,返回到对原位编码进行单向循环一步处理,进行循环处理;是的话,退出循环。

2. 根据权利要求1所述的数据压缩方法,其特征在于,所述的原位记录码,是按照原位码中的单个编码是否为循环终止码,对应地以逻辑代码“0”和“1”分别进行记录而得到的二进制串;所述的原位进位码,是原位码中的单个不是循环终止码的编码均单方向循环一步后所对应的编码得到的M个BIT单位的编码串。

3. 根据权利要求1所述的数据压缩方法,其特征在于,在将记录进位码串接在原位进位码的前面,生成组合进位码时,若在组合进位码的尾巴为循环终止码时,将这循环终止码予以丢弃。

4. 根据权利要求1所述的数据压缩方法,其特征在于,所述的循环终止码可以指定为M个BIT编码中的任意一个编码,作为进位或者退位循环的一个终止位置指示。

5. 根据权利要求1所述的数据压缩方法,其特征在于,所述的单向循环一步是指加1的进位循环或者减1的退位循环。

6. 一种计算设备,其特征在于,包括:处理器、存储器和总线;该存储器通过该总线连接该处理器,该处理器被配置为执行存储器中存储的指令;该处理器通过执行所述存储器的指令实现权利要求1至5任一所述的数据压缩方法。

7. 一种数据压缩装置,用于计算设备,其特征在于,包括:

第一单元,用于将原位码转换为原位编码:对长度为N个BIT单位的二进制数据W,以M个BIT单位进行编码,M为大于等于2的自然数;

第二单元,用于对原位编码进行单向循环一步处理,并将单向循环一步处理后的原位编码转换为原位记录码和原位进位码:将原位编码视为一个数串,所述的单向循环一步处理,相当于让整个数串中的每个单元单向循环一步;

第三单元,用于将原位记录码转换为记录编码,对记录编码进行进位处理,生成一个解码参数;并将记录进位码串接在原位进位码的前面,生成组合进位码;

第四单元,用于判断当前的组合进位码是否小于等于设定的理想剩余值,不是的话,将当前的组合进位码做为新的原位编码,返回到对原位编码进行单向循环一步处理,进行循环处理;是的话,退出循环。

8. 一种数据解压方法,其特征在于,包括以下步骤:

按第n个解码参数Cn的数,将第n次计算后得到的组合进位码Qn的头部分离出来的记录进位码,进行单方向循环一步处理;

将M个BIT编码转为1个BIT的记录代码,如果有余码,将余码添加在尾端;

记录代码与组合进位码 Q_n 分离出来的原位进位码相对应,进行单方向循环一步处理,如果对应不到,默认为循环终止码,得到当前的组合进位码 Q_{n-1} ;

判断当前的组合进位码 Q_{n-1} 是否根据第一个解码参数 C_1 运算得到,不是的话,将当前的组合进位码 Q_{n-1} 作为当前要参加运算的组合进位码 Q_n ;提取解码参数 C_{n-1} 作为当前要参加运算的解码参数 C_n ,返回到按第n个解码参数 C_n 的数,将第n次计算后得到的组合进位码 Q_n 的头部分离出来的记录进位码,进行单方向循环一步处理,进行循环处理;是的话,退出循环。

9. 一种计算设备,其特征在于,包括:处理器、存储器和总线;该存储器通过该总线连接该处理器,该处理器被配置为执行存储器中存储的指令;该处理器通过执行所述存储器的指令实现权利要求8所述的数据解压方法。

10. 一种数据解压装置,用于计算设备,其特征在于,包括:

第一单元,用于按第n个解码参数 C_n 的数,将第n次计算后得到的组合进位码 Q_n 的头部分离出来的记录进位码,进行单方向循环一步处理;

第二单元,用于将M个BIT编码转为1个BIT的记录代码,如果有余码,将余码添加在尾端;

第三单元,用于记录代码与组合进位码 Q_n 分离出来的原位进位码相对应,进行单方向循环一步处理,如果对应不到,默认为循环终止码,得到当前的组合进位码 Q_{n-1} ;

第四单元,用于判断当前的组合进位码 Q_{n-1} 是否根据第一个解码参数 C_1 运算得到,不是的话,将当前的组合进位码 Q_{n-1} 作为当前要参加运算的组合进位码 Q_n ;提取解码参数 C_{n-1} 作为当前要参加运算的解码参数 C_n ,返回到按第n个解码参数 C_n 的数,将第n次计算后得到的组合进位码 Q_n 的头部分离出来的记录进位码,进行单方向循环一步处理,进行循环处理;是的话,退出循环。

数据压缩装置及方法、数据解压装置及方法

技术领域

[0001] 本发明涉及二进制数据的处理,尤其涉及二进制数据的压缩。

背景技术

[0002] 随着信息时代的发展,数据量呈爆发式增长。除了硬件技术遵循摩尔定律快速更新迭代应对外,软件技术也是解决方案之一,尤其是数据压缩技术,最直接有效地减少信息数据的体量,不仅节省存储空间和传输时间,而且相对于硬件更经济更安全、可靠。

[0003] 现在使用的数据无损压缩算法,主要有两大类:1、熵编码类,如:霍夫曼编码,算术编码等;2、字典编码类,如:RLE编码,LZ编码等。它们基于不同的理念,设计出不同的算法,针对不同类型的数据,产生不同的压缩效果,但它们的共同点:都是把数据看作信息的载体,减少信息的冗余度,用最少的BIT单位承载原数据信息内容,以达到压缩数据的目的,所以它们都受到了信息熵理论的限制,无损压缩比一般都在2:1到5:1的范围内,可压缩空间非常有限。

发明内容

[0004] 本发明要解决的技术问题在于,针对现有技术的上述缺陷,提出一种数据压缩方法,能够大大提升无损压缩的效率。

[0005] 本发明解决其技术问题所采用的技术方案包括:提供一种数据压缩方法,包括以下步骤:

[0006] 将原位码转换为原位编码:对长度为N个BIT单位的二进制数据W,以M个BIT单位进行编码,M为大于等于2的自然数;

[0007] 对原位编码进行单向循环一步处理,并将单向循环一步处理后的原位编码转换为原位记录码和原位进位码;

[0008] 将原位编码视为一个数串,所述的单向循环一步处理,相当于让整个数串中的每个单元单向循环一步;

[0009] 将原位记录码转换为记录编码,对记录编码进行进位处理,生成一个解码参数;并将记录进位码串接在原位进位码的前面,生成组合进位码;判断当前的组合进位码是否小于等于设定的理想剩余值,不是的话,将当前的组合进位码做为新的原位编码,返回到对原位编码进行单向循环一步处理,进行循环处理;是的话,退出循环。

[0010] 在一些实施例中,所述的原位记录码,是按照原位码中的单个编码是否为循环终止码,对应地以逻辑代码“0”和“1”分别进行记录而得到的二进制串;所述的原位进位码,是原位码中的单个不是循环终止码的编码均单方向循环一步后所对应的编码儿得到的M个BIT单位的编码串。

[0011] 在一些实施例中,在将记录进位码串接在原位进位码的前面,生成组合进位码时,若在组合进位码的尾巴为循环终止码时,将这循环终止码予以丢弃。

[0012] 在一些实施例中,所述的循环终止码可以指定为M个BIT编码中的任意一个编码,

作为进位或者退位循环的一个终止位置指示。

[0013] 在一些实施例中,所述的单向循环一步是指加1的进位循环或者减1的退位循环。

[0014] 本发明解决其技术问题所采用的技术方案还包括:提供一种计算设备,其包括:处理器、存储器和总线;该存储器通过该总线连接该处理器,该处理器被配置为执行存储器中存储的指令;该处理器通过执行所述存储器的指令实现前述的数据压缩方法。

[0015] 本发明解决其技术问题所采用的技术方案还包括:提供一种数据压缩装置,用于计算设备,其包括:

[0016] 第一单元,用于将原位码转换为原位编码:对长度为N个BIT单位的二进制数据W,以M个BIT单位进行编码,M为大于等于2的自然数;

[0017] 第二单元,用于对原位编码进行单向循环一步处理,并将单向循环一步处理后的原位编码转换为原位记录码和原位进位码:将原位编码视为一个数串,所述的单向循环一步处理,相当于让整个数串中的每个单元单向循环一步;

[0018] 第三单元,用于将原位记录码转换为记录编码,对记录编码进行进位处理,生成一个解码参数;并将记录进位码串接在原位进位码的前面,生成组合进位码;

[0019] 第四单元,用于判断当前的组合进位码是否小于等于设定的理想剩余值,不是的话,将当前的组合进位码做为新的原位编码,返回到对原位编码进行单向循环一步处理(也即返回到第二单元),进行循环处理;是的话,退出循环。

[0020] 本发明解决其技术问题所采用的技术方案还包括:提供一种数据解压方法,其包括以下步骤:

[0021] 按第n个解码参数 C_n 的数,将第n次计算后得到的组合进位码 Q_n 的头部分离出来的记录进位码,进行单方向循环一步处理;

[0022] 将M个BIT编码转为1个BIT的记录代码,如果有余码,将余码添加在尾端;

[0023] 记录代码与组合进位码 Q_n 分离出来的原位进位码相对应,进行单方向循环一步处理,如果对应不到,默认为循环终止码,得到当前的组合进位码 Q_{n-1} ;

[0024] 判断当前的组合进位码 Q_{n-1} 是否根据第一个解码参数 C_1 运算得到,不是的话,将当前的组合进位码 Q_{n-1} 作为当前要参加运算的组合进位码 Q_n ;提取解码参数 C_{n-1} 作为当前要参加运算的解码参数 C_n ,返回到按第n个解码参数 C_n 的数,将第n次计算后得到的组合进位码 Q_n 的头部分离出来的记录进位码,进行单方向循环一步处理,进行循环处理;是的话,退出循环。

[0025] 本发明解决其技术问题所采用的技术方案还包括:提供一种计算设备,其包括:处理器、存储器和总线;该存储器通过该总线连接该处理器,该处理器被配置为执行存储器中存储的指令;该处理器通过执行所述存储器的指令实现前述的数据解压方法。

[0026] 本发明解决其技术问题所采用的技术方案还包括:提供一种数据解压装置,用于计算设备,其包括:

[0027] 第一单元,用于按第n个解码参数 C_n 的数,将第n次计算后得到的组合进位码 Q_n 的头部分离出来的记录进位码,进行单方向循环一步处理;

[0028] 第二单元,用于将M个BIT编码转为1个BIT的记录代码,如果有余码,将余码添加在尾端;

[0029] 第三单元,用于记录代码与组合进位码 Q_n 分离出来的原位进位码相对应,进行单

方向循环一步处理,如果对应不到,默认为循环终止码,得到当前的组合进位码 Q_{n-1} ;

[0030] 第四单元,用于判断当前的组合进位码 Q_{n-1} 是否根据第一个解码参数 C_1 运算得到,不是的话,将当前的组合进位码 Q_{n-1} 当前要参加运算的组合进位码 Q_n ;提取解码参数 C_{n-1} 作为当前要参加运算的解码参数 C_n ,返回到按第 n 个解码参数 C_n 的数,将第 n 次计算后得到的组合进位码 Q_n 的头部分离出来的记录进位码,进行单方向循环一步处理(也即返回到第一单元),进行循环处理;是的话,退出循环。

[0031] 与现有技术相比,本发明的数据压缩方法,通过将二进制的数视为一个十进制的大整数,经过简单的运算处理,表述为若干个解码参数和一个组合进位码,可以大大提升无损压缩的效率,尤其针对存储量在上GBIT的大数据。

附图说明

[0032] 图1示意出本发明的数学二叉树的模型的结构。

[0033] 图2示意出本发明的数据压缩方法的流程。

[0034] 图3示意出本发明的数据解压方法的流程。

[0035] 图4示意出本发明的计算设备的结构框图。

[0036] 图5示意出本发明的数据压缩装置的框图。

[0037] 图6示意出本发明的数据解压装置的框图。

具体实施方式

[0038] 为了详细说明本发明的构造及特点所在,兹举以下较佳实施例并配合附图说明如下。

[0039] 本发明提出一种新的数据压缩方法理念,将数据(二进制串)整体看成一个十进制的大整数,因为所有二进制数据都可以计算出一个十进制的整数。如何简化表达这个大整数,以达到压缩数据的效果,这是一个纯数学问题,与该数据承载的信息内容无关。本发明的压缩方法可以突破原数据信息熵值的限制,对于大数据(例如:存储量在MBIT数量级的MP3音乐,存储量在GBIT数量级的DVD视频),压缩比可达千分之一甚至万分之一以上。

[0040] 本发明的压缩方法通过对原数据,以2个BIT单位进行编码进行编码,形成数据编码链,经过数学矩阵行列、逻辑等运算,输出运算过程得到的系列解码参数,用系列解码参数和计算剩余的数据来替代原数据,达到压缩数据的目的。由于本发明的压缩方法可以对数据进行重复压缩计算,任何二进制数据都可以被压缩到只剩下2个BIT单位和系列解码参数,具有数据越大,压缩比越大;压缩解压速度快;无损压缩;能够通用于任何类型的数据等优点。

[0041] 参见图1,图1示意出本发明的数学二叉树的模型的结构。将由“0”或“1”构成的二进制数据,以一个BIT为单位,放置到数学二叉树模型上,数据的第一位BIT,放在二叉树顶上的1位置(注图中的数字1为数学二叉树的顺序编号),数据的第二位BIT如果是“0”则放在左下叉点,即第二层的2位置上;如果是“1”则放在右下叉点,即第二层的3位置上。第三层根据第二层是2位置还是3位置,以及数据的第三位BIT是“0”还是“1”,决定第三层叉点是4、5还是6、7的位置。依此类推,按照上述规则,二叉树的一层对应放数据的一个BIT位,直到最后一个BIT位于对应的二叉树层的交叉点位置,这个位置的数学二叉树顺序编码就是该二

进制数据的十进制数。如果把这个数学二叉树的模型予以展开,就会发现:数学二叉树上的编号就是从1到无穷大的自然数。

[0042] 由此,可以得到一个结论:除了0和1之外的任何一个自然数,都对应或两个二进制数据,这两个二进制数据的唯一不同之处在于:一个是从“0”开始的数据(也即数学二叉树的第一层放“0”),另一个是从“1”开始的数据(也即数学二叉树的第一层放“1”)。可以理解的是,只要在图1中的1位置,可以放置“0”和“1”;在其他位置,遵照前述的左下叉放“0”、右下叉放“1”的规则,只能相应地限定为“0”或“1”。

[0043] 任何二进制数据所对应的自然数,在理论上都可以被计算出来,但是由于数据量都很大,通常是以万BIT位甚至亿BIT位计的大数,根本无法进一步分解简化表达。

[0044] 从前述的数学二叉树的模型可以看出,只有当这个大数落在二叉树的左右两边上的叉点时,才可以被简化表达。根据这个大数是由数据的每个BIT位一步一步构成的,可以得出:只要让每一层的BIT位都归“0”或归“1”就可以让这个大数归边,而且可以用每层BIT归“0”或归“1”过程的记录,简化表达这个大数。据此,本发明提出一种新的数据压缩方法,也称进位循环归零的算法,或者,进位循环归一的算法。

[0045] 参见图2,图2示意出本发明的数据压缩方法的流程。本发明的压缩方法具体包括以下步骤:

[0046] S201、将原位码转换为原位编码。

[0047] 对长度为N个BIT单位的二进制数据W(也称原位码),以M个BIT单位进行编码(M为大于等于2的自然数);以下以M=2并且进行单方向循环一步处理为进位处理为例,进行说明。

[0048] 以2个BIT单位进行编码,即编成:“00”、“01”、“10”和“11”四种码(以下为表述方便,用字母a代表编码“01”;字母b代表编码“10”;字母c代表编码“11”;字母d代表编码“00”)。原位码在进行编码之后,形成N/2个2个BIT编码链(也称原位编码),如果N为奇数,最后一个BIT位作为余码处理,用(0/1)表示,(0)代表余码为“0”,(1)代表余码为“1”。

[0049] S203、对原位编码进行单方向循环一步处理,并将单方向循环一步处理后的原位编码转换为原位记录码和原位进位码。

[0050] 具体地,将原位编码视为一个数串;所述的进位处理,相当于让整个数串中的每个单元前进一步,即:a进位成b,b进位成c,c进位成d,d进位成a;可以理解的是,每个单元在进位之前与进位之后是相差1的循环关系。比如:表述为[cbcd]的数串,经进位处理后,变换成表述为[dcda]的数串;表述为[aaaa]的数串,经进位处理后,变换成表述为[bbbb]的数串;表述为[dddd]的数串,经进位处理后,变换成表述为[aaaa]的数串。

[0051] 在a、b、c、d四个编码之中选定一个为循环终止码,其不发生进位,其他三个可以发生进位。例如:将d设为循环终止码,则a、b、c三个编码可以发生进位。在原位码中的单元(即一个编码)是可以发生进位的,在记录码中以逻辑代码“1”进行记录,在原位码中的单元是循环终止码的,在记录码中以逻辑代码“0”进行记录。在原位进位码中保存发生进位后的编码。

[0052] 举例而言,原位码为[adcb],其记录码为[1011],其进位码为[bdc]。也即,在原位码中,a进位成b,记录代码为“1”;b进位成c,记录代码为“1”;c进位成d,记录代码为“1”;d不进位,记录代码为“0”。

[0053] S205、将原位记录码转换为记录编码,对记录编码进行单方向循环一步处理,生成一个解码参数;并将记录进位码串接在原位进位码的前面,生成组合进位码 Q_1 。

[0054] 参照前述的步骤S201,将得到的记录码当作新生成的数据,并按2个BIT单位进行编码,形成记录编码,例如:记录码为[1110000],则记录编码为[cbd]+(0),注:(0)为余码。

[0055] 将记录编码视为一个数串;所述的进位处理,相当于让整个数串中的每个单元前进一步,即:a进位成b,b进位成c,c进位成d,d进位成a;统计这个数串中的单元总数,得到记录编码数,在结合记录编码中的余码,做为一个解码参数 C_1 输出;在记录进位码中保存发生进位后的编码。

[0056] 值得一提的是,在前述的原位进位码中,对于来源于循环终止码的进位要求是予以丢弃的;在这里的记录进位码中,对于来源于循环终止码的进位要求是予以保留的,也即是不区分循环终止码的全进位。

[0057] 举例而言,前述的记录编码:[cbd]+(0)经过进位处理后,得到的记录进位码为[dca],记录编码数为3,解码参数 C_1 为3(0)。

[0058] 具体地,将记录进位码对应的数串排列在原位进位码对应的数串的左侧,所得到的数串即为组合进位码 Q_1 。

[0059] 值得一提的是,在组合进位码 Q_1 的数串的尾巴为前述的循环终止码d时,将这循环终止码d予以丢弃,可以提高压缩效率。相应地,在解码时,凡是遇到解码参数和记录代码的1对应不到组合进位码时,默认为循环终止码d,还原为与循环终止码退一步相对应的编码c。

[0060] 步骤S207、判断当前的组合进位码 Q_1 是否小于等于Q,是的话,退出循环;否则的话,将组合进位码 Q_1 视为新的原位编码,重复前述的步骤S203和步骤S205的处理过程,可以得到新的解码参数 C_2 和新的组合进位码 Q_2 ,如此类推,可以计算出 $C_3(0/1)C_4(0/1)\cdots C_n(0/1)Q_n$,直到 Q_n 小于等于Q为止,Q为设定的理想剩余值,例如:Q=2个BIT。

[0061] 综上,前述处理过程可以描述为公式一: $W=P(0/1)C_1(0/1)C_2(0/1)\cdots C_n(0/1)Q_n$,其中:W为数据;P为重复压缩次数;(0/1)为余码; C_n 为第n个解码参数; Q_n 为第n次计算后得到的组合进位码。

[0062] 可以理解的是,本发明的压缩方法,通过将数据W视为一个整数,而无论数据本身的结构,进而通过简单运算,用n个解码参数: $C_1(0/1)C_2(0/1)\cdots C_n(0/1)$ 和一个组合进位码 Q_n 来表述这个数据W。由于n个解码参数和一个组合进位码的BIT量可以远小于数据W本身的BIT量,从而能够得到很大数值的压缩比。

[0063] 举例而言,对于一个GBIT(8.59×10^9 次方BIT)的数据W,经压缩后,解码参数的总数约为300个,以每个参数平均占用40BIT计算(考虑到:解码参数的最大值约为N/4,N为数据W的二进制位数,平均按10位十进制数,每4BIT相当于一位十进制数),可以压缩到12000BIT以内,压缩比大致为百万分之十四。如果需要可以对压缩结果12000BIT进行二次或多次重复压缩。

[0064] 对应于上述的压缩方法,本发明提出一种解压方法,表述为公式二: $Q_{n-1}=C_nQ_n$ 。参见图3,图3示意出本发明的数据解压方法的流程。本发明的解压方法具体包括以下步骤:

[0065] S301、按第n个解码参数 C_n 的数,将第n次计算后得到的组合进位码 Q_n 的头部分离出来的记录进位码(如果 Q_n 中不够数,默认为循环终止码d,补上),从上往下(或者说,从头往

尾,从左往右),a码解码为d码;b码解码为a码;c码解码为b码;d码解码为c码;可以理解的是,与压缩过程的进位处理相对应,在解码过程中,是进行了退位处理;同理,与压缩过程的退位处理相对应,在解码过程中,是进行了进位处理。

[0066] S303、将步骤S301解码后的2个BIT编码转为1BIT的记录代码,如果有余码(0/1),将余码添加在尾端。

[0067] S305、将步骤S303还原的记录代码与组合进位码 Q_n 分离出来的原位进位码相对应,从上往下(或者说,从头往尾,从左往右),遇到记录代码中代码为“0”时,解码为编码d;遇到记录代码中代码为“1”时,对应于原位进位码:b码解码为a码;c码解码为b码;d码解码为c码;如果对应不到,默认为编码d,解码为c码。如此,得到当前的组合进位码 Q_{n-1} 。

[0068] S307、判断当前的组合进位码 Q_{n-1} 是否根据第一个解码参数 C_1 运算得到,是的话,退出循环;否则的话,将当前的组合进位码 Q_{n-1} 当前要参加运算的组合进位码 Q_n ;提取解码参数 C_{n-1} 作为当前要参加运算的解码参数 C_n 。

[0069] 重复前述的步骤S301、步骤S303和步骤S305的小循环的处理过程,可以完成解码的一个大循环,得到组合进位码 Q_0 ,即为数据的原码,如果有余码(0/1),随尾加上。

[0070] 值得一提的是,如果 $P \geq 1$,说明是多次重复压缩,则需要重复前述的解码过程的多个大循环,每完成一个大循环,做 $i = i - 1$ 的递减处理,直至 $i = 0$,其中, i 的初始值为 $P + 1$,也即 P 为0时表示只进行了一次压缩, P 为1时表示进行了两次压缩, P 为2时表示进行了叁次压缩。

[0071] 下面,以二进制数据 $W = 100111010000101100101011001$ 为例,对上述的压缩方法和解压方法进行演示。

[0072] A、压缩过程

[0073] 一、将原位码转换为原位编码(对应于前述的步骤S201)。

[0074] $100111010000101100101011001 = > 10, 01, 11, 01, 00, 00, 10, 11, 00, 10, 10, 11, 00, 1 = > [\text{bacaddbcbdbbcd}] (1)$ 。

[0075] 二、对原位编码进行进位处理,并将进位处理后的原位编码转换为原位记录码和原位进位码(对应于前述的步骤S203)。

[0076] 原位记录码为 $[1111001101110]$;原位进位码为 $[cbdbcdcccd]$ 。

[0077] 三、将原位记录码转换为记录编码,对记录编码进行进位处理,生成一个解码参数,并得到记录进位码,将记录进位码与原位进位码叠加,生成组合进位码 Q_1 (对应于前述的步骤S205)。

[0078] 记录编码为 $1111001101110 = > 11, 11, 00, 11, 01, 11, 0 = > [\text{ccdcac}] (0)$;

[0079] 第一个解码参数 C_1 为6(0);

[0080] 记录进位码为 $[\text{ddadbd}]$ 。

[0081] Q_1 为 $[\text{ddadbd}] + [\text{cbdbcdcccd}] = > [\text{ddadbdcdbdbcdcccd}]$,再进行丢弃尾部的循环终止码d, Q_1 最后表述为 $[\text{ddadbdcdbdbcdcc}]$

[0082] 重复前述的步骤三步骤的小循环运算,可以依次计算出:第二个解码参数 C_2 为7,第三个解码参数 C_3 为6,第四个解码参数 C_4 为6,第五个解码参数 C_5 为5,第六个解码参数 C_6 为5(1),第七个解码参数 C_7 为3(1),第八个解码参数 C_8 为3(1),第九个解码参数 C_9 为2(1),第十个解码参数 Q_{10} 为2,第十组合进位码 Q_{10} 为a,也即“01”。

[0083] 由于只进行了一次压缩处理,可以将整个压缩过程表述为: $W=0(1), 6(0), 7, 6, 6, 5, 5(1), 3(1), 3(1), 2(1), 2, a$ 。其中,等号右边的首位的0表示只进行了一次压缩处理,没有进行重复压缩;接下来的(1)表示余码为“1”;接下来的6(0), 7, 6, 6, 5, 5(1), 3(1), 3(1), 2(1), 2分别表示解码参数 $C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}$;最后的编码a表示第十次计算后得到的组合进位码。

[0084] B、解压过程

[0085] 第1次小循环:

[0086] $Q_9 = C_{10}Q_{10} = 2a = > [11][a-] = > [11][ad] = > [dc]$;对应于前述的步骤S301。

[0087] $[dc] = > [0011]$;对应于前述的步骤S303。

[0088] $[0011][] = > [0011][] = > [ddcc] = Q_9$ 。对应于前述的步骤S305。

[0089] 第2次小循环:

[0090] $Q_8 = C_9Q_9 = 2(1)[ddcc] = > [11][dd][cc](1) = > [cc](1)$;对应于前述的步骤S301。

[0091] $[cc](1) = > [11111]$;对应于前述的步骤S303。

[0092] $[11111][cc] = > [bbccc] = Q_8$ 。对应于前述的步骤S305。

[0093] 第3次小循环:

[0094] $Q_7 = C_8Q_8 = 3(1)[bbccc] = > [111][abb][cc](1) = > [aab](1)$;对应于前述的步骤S301。

[0095] $[aab](1) = > [0101101]$;对应于前述的步骤S303。

[0096] $[0101101][cc] = > [dbdbcdc] = Q_7$ 。对应于前述的步骤S305。

[0097] 第4次小循环:

[0098] $Q_6 = C_7Q_7 = 3(1)[dbdbcdc] = > [111][dbd][bcd](1) = > [cac](1)$;对应于前述的步骤S301。

[0099] $[cac](1) = > [1101111]$;对应于前述的步骤S303。

[0100] $[1101111][bcd] = > [abdcbbc] = Q_6$ 。对应于前述的步骤S305。

[0101] 第5次小循环:

[0102] $Q_5 = C_6Q_6 = 5(1)[abdcbbc] = > [11111][abdc][cc](1) = > [dacba](1)$;对应于前述的步骤S301。

[0103] $[dacba](1) = > [00011110011]$;对应于前述的步骤S303。

[0104] $[00011110011][cc] = > [dddbbccddcc] = Q_5$ 。对应于前述的步骤S305。

[0105] 第6次小循环:

[0106] $Q_4 = C_5Q_5 = 5[dddbbccddcc] = > [11111][dddbb][ccddcc] = > [cccaa]$;对应于前述的步骤S301。

[0107] $[cccaa] = > [1111110101]$;对应于前述的步骤S303。

[0108] $[1111110101][ccddcc] = > [bbccbbddcdc] = Q_4$ 。对应于前述的步骤S305。

[0109] 第7次小循环:

[0110] $Q_3 = C_4Q_4 = 6[bbccbbddcdc] = > [111111][bbccbb][ddcdc] = > [aabbba]$;对应于前述的步骤S301。

[0111] $[aabbba] = > [010110100101]$;对应于前述的步骤S303。

[0112] $[010110100101][ddcdc] = > [dcdbcdbddcdc] = Q_3$ 。对应于前述的步骤S305。

[0113] 第8次小循环:

[0114] $Q_2 = C_3 Q_3 = 6 [dcdcbcd bddcdc] \Rightarrow [111111] [dcdcbd] [bddcdc] \Rightarrow [cbcabc]$; 对应于前述的步骤S301。

[0115] $[cbcabc] \Rightarrow [111011011011]$; 对应于前述的步骤S303。

[0116] $[111011011011] [bddcdc] \Rightarrow [accdbcd bcdcc] = Q_2$ 。对应于前述的步骤S305。

[0117] 第9次小循环:

[0118] $Q_1 = C_2 Q_2 = 7 [accdbcd bcdcc] \Rightarrow [1111111] [accdbcd] [bcdcc] \Rightarrow [dbbcabc]$; 对应于前述的步骤S301。

[0119] $[dbbcabc] \Rightarrow [00101011011011]$; 对应于前述的步骤S303。

[0120] $[00101011011011] [bcdcc] \Rightarrow [ddadbcd bcdcc] = Q_1$ 。对应于前述的步骤S305。

[0121] 第10次小循环:

[0122] $Q_0 = C_1 Q_1 = 6 (0) [ddadbcd bcdcc] \Rightarrow [111111] [ddadbd] [cbdbcdcc] \Rightarrow [ccdcac] (0)$; 对应于前述的步骤S301。

[0123] $[ccdcac] (0) \Rightarrow [1111001101110]$; 对应于前述的步骤S303。

[0124] $[1111001101110] [cbdbcdcc] \Rightarrow [bacad bcd bbd] = Q_0$ 。对应于前述的步骤S305。

[0125] 补上0 (1) 的余码, 得 $W = 0 (1) Q_0 = 100111010000101100101011001$ 。

[0126] 综上, 本发明的压缩方法, 具有以下特点:

[0127] 1、每运行三次 (即前述的压缩过程的三个步骤S201、S203和S205) 为一个小周期, 每个小周期的平均压缩率, 按照发明人的估算, 在18%以上。

[0128] 2、在前几个小周期的运算过程, 会出现数据量增大的现象, 这是由于记录进位码与原位进位码串接 (也称叠加) 在一起的缘故。

[0129] 3、对数据量越大的压缩效果越好, 压缩比越大; 任何数据都可以压缩到只剩下2个BIT和系列解码参数。

[0130] 4、2个BIT编码是最佳的, 也可以是采用3个BIT ($M=3$)、4个BIT ($M=4$) 等其他位数的编码; 可以理解的是, 采用3个BIT编码的话, 任何数据都可以压缩到只剩下3个BIT和系列解码参数; 采用4个BIT编码的话, 任何数据都可以压缩到只剩下4个BIT和系列解码参数。

[0131] 5、前述的进位运算可以替换为退位运算; 编码a、b、c、d是相差1的循环关系, 编码a、b、c、d具体对应哪个数都可以; 前述的进位运算是指加1的循环一步的运算; 前述的退位运算是指减1的循环一步的运算;

[0132] 6、前述的循环终止码d可以指定为M个BIT编码中的任意一个编码, 作为进位或者退位循环的一个终止位置指示; 在实际应用中, 可以选用最多数量的编码来作为循环终止码d, 做诸如不进位、默认或者丢弃的特别对待, 以减少运算量。

[0133] 7、对于原始码及记录代码, 可以将本发明的压缩方法与行程码压缩方法相结合, 例如: 先用行程码压缩技术, 先压缩“00”和“11”两组码, 再用本发明的压缩方法, 如此可以进一步提高压缩效率。

[0134] 8、设定的理想剩余值Q, 可以按照压缩、解压效率及实际应用的需要, 选取合理的值。

[0135] 值得一提的是, 本发明的数据压缩方法和数据解压方法, 可以不区分数据类型地, 广泛应用于对诸如文字电子文档、音乐电子文档和视频电子文档等各类数据进行压缩和解

压处理。

[0136] 可以理解的是,本发明的数据压缩方法和数据解压方法,可以运行在诸如智能手机、平板电脑、个人电脑、云计算、区块链、数据库服务器和智能家用电器等各类计算设备上的处理器上的软件程序予以实现。

[0137] 参见图4,图4示意出本发明的计算设备的结构框图。举例而言,计算设备400包括:处理器401、存储器402和总线403,存储器402通过总线403连接处理器401,处理器401被配置为执行存储器402中存储的指令,处理器401通过执行存储器402中存储的指令实现本发明的数据压缩方法和/或数据解压方法。

[0138] 处理器401包括运算逻辑部件、寄存器部件以及控制部件等,其可以是独立的中央处理器,或者也可以是嵌入式处理器,比如微处理器(英文:Micro Processor Unit,缩写:MPU)、微控制器(英文:Microcontroller Unit,缩写:MCU)或者数字信号处理器(英文:Embedded Digital Signal Processor,缩写:EDSP)等。

[0139] 存储器402由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(英文:Static Random Access Memory,简称:SRAM),电可擦除可编程只读存储器(英文:Electrically Erasable Programmable Read-Only Memory,简称:EEPROM),可擦除可编程只读存储器(英文:Erasable Programmable Read Only Memory,简称:EPROM),可编程只读存储器(英文:Programmable Read-Only Memory,简称:PROM),只读存储器(英文:Read Only Memory,简称:ROM),磁存储器,快闪存储器,磁盘或光盘。存储器可用于存储指令,该指令可以实现为软件程序或者软件模块。

[0140] 其中,处理器401可以通过执行存储器402中存储的指令来实现图2所示的数据压缩方法的全部或者部分步骤,和/或,图3所示的数据解压方法的全部或者部分步骤。

[0141] 计算设备400还可以包括连接在总线403上的缓存器404和通信组件405。其中,缓存器404通过总线403与处理器401和/或存储器402配合,用于缓存处理器401在计算过程中用到/产生的一些中间数据。

[0142] 通信组件405用于对外通信,包括对外部网络通信和/或与其他计算设备/存储设备通信,其可以包括多种类型的通信模块,例如:实现有线通讯用的以太网收发器,和/或,实现无线通讯用的无线收发器。

[0143] 可以理解的是,本发明的数据压缩方法和数据解压方法,还可以采用诸如专用的集成电路或可编程的集成电路之类的硬件电路予以实现;或者,可以一部分采用软件程序、另一部分采用硬件电路的软硬混合方式予以实现。

[0144] 可以理解的是,本发明提出了一种数据压缩装置,用于计算设备,实现了上述的数据压缩方法。该数据压缩装置可以通过软件、硬件或者两者的结合实现成为上述图4所示计算设备400的全部或者一部分。该数据压缩装置可以通过专用集成电路(英文:application-specific integrated circuit,缩写:ASIC)实现,或可编程逻辑器件(英文:programmable logic device,缩写:PLD)实现。上述PLD可以是复杂可编程逻辑器件(英文:complex programmable logic device,缩写:CPLD),现场可编程逻辑门阵列(英文:field-programmable gate array,缩写:FPGA),通用阵列逻辑(英文:generic array logic,缩写:GAL)或其任意组合。当图2所示的数据压缩方法通过软件实现时,该数据压缩装置中的各个单元也可以为软件模块。

[0145] 参见图5,图5示意出本发明的数据压缩装置的框图。该数据压缩装置500包括对应于图2示出的S201、S203、S205和S207四个步骤的四个单元501、503、505和507,每个单元用于对应实现其中一个步骤,也即:第一单元501对应实现步骤S201,第二单元503对应实现步骤S203,第三单元505对应实现步骤S205,第四单元507对应实现步骤S207。

[0146] 可以理解的是,本发明提出了一种数据解压装置,用于计算设备,实现了上述的数据解压方法。该数据解压装置与前述的数据压缩装置类似,可以通过软件、硬件或者两者的结合实现成为上述图4所示计算设备400的全部或者一部分。该数据解压装置可以通过ASIC实现,或PLD实现。上述PLD可以是CPLD,FPGA,GAL或其任意组合。当图3所示的数据解压方法通过软件实现时,该数据解压装置中的各个单元也可以为软件模块。

[0147] 参见图6,图6示意出本发明的数据解压装置的框图。该数据解压装置600包括对应于图3示出的S301、S303、S305和S307四个步骤的四个单元601、603、605和607,每个单元用于对应实现其中一个步骤,也即:第一单元601对应实现步骤S301,第二单元603对应实现步骤S303,第三单元605对应实现步骤S305,第四单元607对应实现步骤S307。

[0148] 可以理解的是,本发明提出了一种计算机可读存储介质,其以诸如光盘和硬盘之类的存储介质,保存有实现了上述的数据压缩方法的软件程序;或者,其以诸如光盘和硬盘之类的存储介质,保存有实现了上述的数据解压方法的软件程序。

[0149] 以上,仅为本发明之较佳实施例,意在进一步说明本发明,而非对其进行限定。凡根据上述之文字和附图所公开的内容进行的简单的替换,都在本专利的权利保护范围之内。

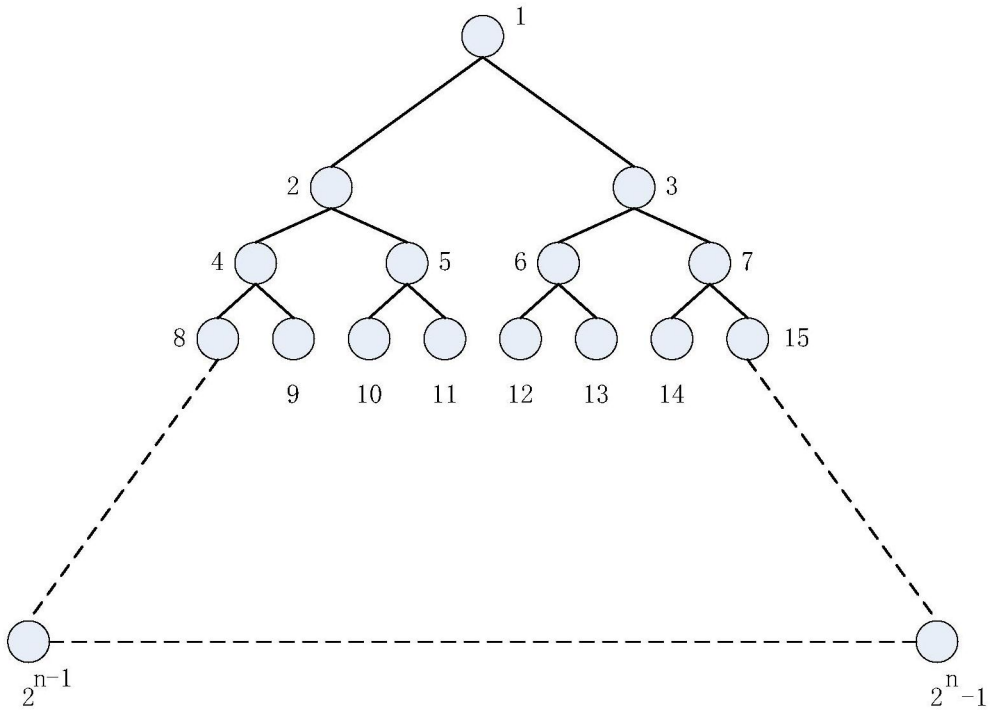


图1

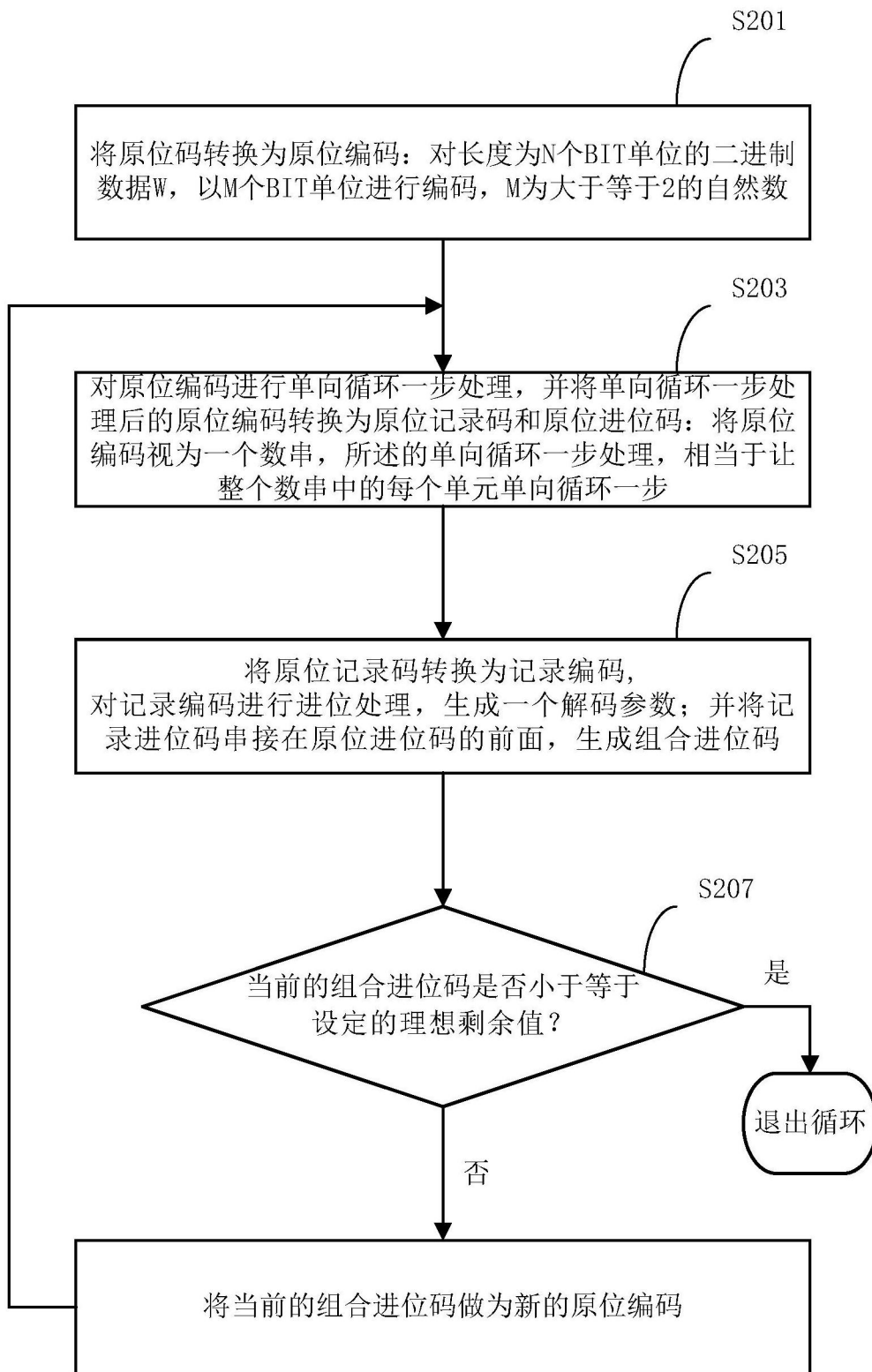


图2

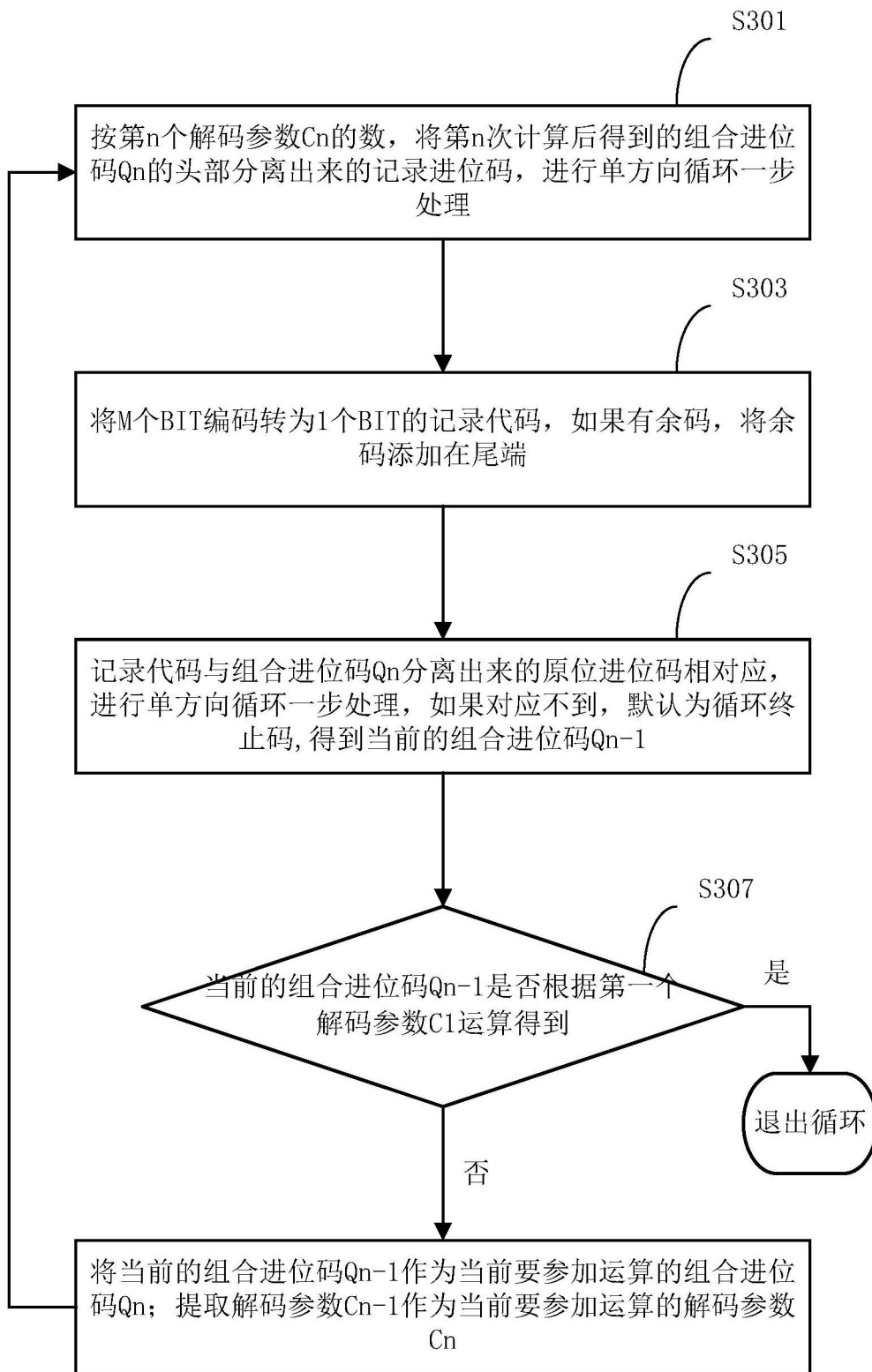


图3

400

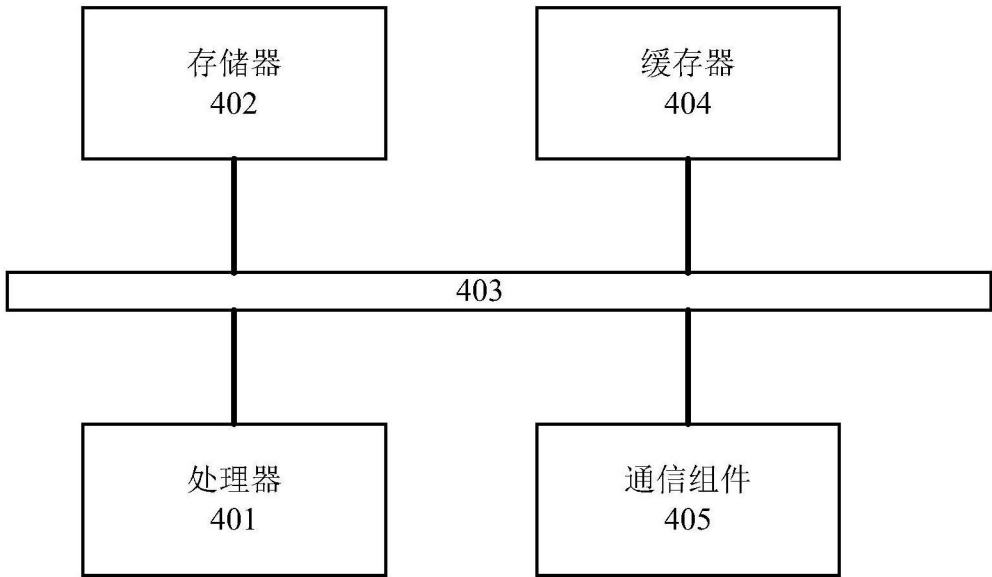


图4

500

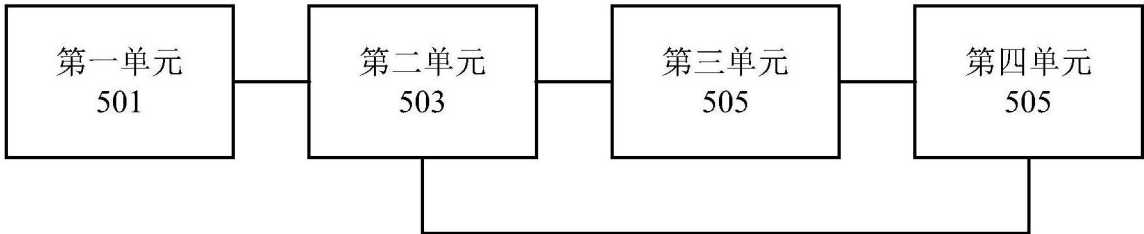


图5

600

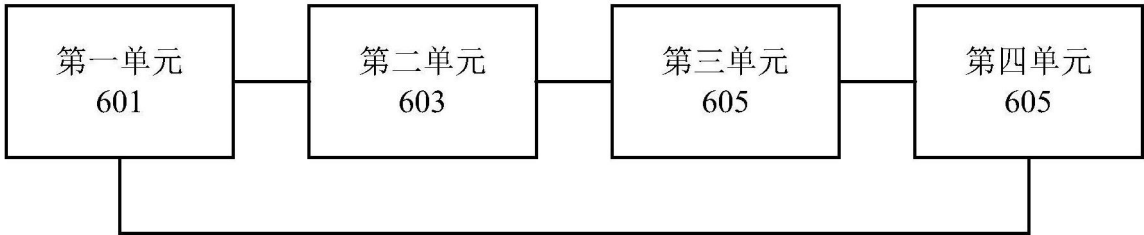


图6