(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103124182 B
(45)授权公告日 2017.05.10

(21)申请号 201310014760.3

(22)申请日 2005.05.09

(65)同一申请的已公布的文献号
申请公布号 CN 103124182 A

(43)申请公布日 2013.05.29

(30)优先权数据
60/569,127 2004.05.07 US

(62)分案原申请数据
200580001026.7 2005.05.09

(73)专利权人 数字方敦股份有限公司
地址 美国加利福尼亚州

(72)发明人 M·G·路比 M·A·肖克罗拉希 M·华生

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 亓云

(51)Int.Cl.
H03M 13/29(2006.01)
H03M 13/37(2006.01)
H04L 1/00(2006.01)

(56)对比文件
US 2003058958 A1,2003.03.27,
US 6373406 B1,2002.04.16,
US 6320520 B1,2001.11.20,
US 5432787 A,1995.07.11,
US 6486803 B1,2002.11.26,
US 6307487 B1,2001.10.23,

审查员 刘蕾蕾

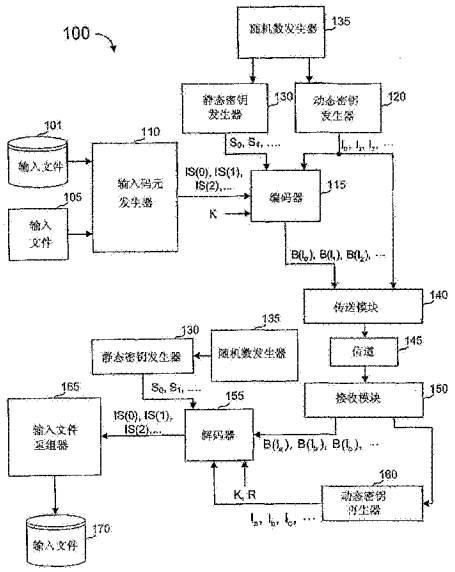
权利要求书4页 说明书39页 附图17页

(54)发明名称

文件下载和流系统

(57)摘要

提供了一种编码用于在通信信道上从源传送到目标的数据的方法。该方法在输入码元的有序集上运算,并包括从输入码元产生多个冗余码元。该方法还包括从包含输入码元和冗余码元的码元的组合集中产生多个输出码元,其中可能的输出码元的数量比码元的组合集中码元的数量多得多,其中至少一个输出码元是从码元的组合集中的一个以上码元、并从码元的组合集中的少于全部码元的码元中产生的,从而输入码元的有序集可从任何预定数量的输出码元中以期望准确度再生。多个冗余码元从输入码元的有序集中产生以以下确定过程传送,即使用第一输入码元计算的第一静态码元集与使用不同于第一输入码元的第二输入码元计算的第二静态码元集具有较少的共同成员。



CN 103124182 B

1.一种编码用于在通信信道上从源传送到目标的数据的方法，其中所述用于传送的数据是由输入码元的有序集表示的，所述方法包括：

从所述输入码元的有序集中生成多个冗余码元，其中所述多个冗余码元中的每个冗余码元使用所述输入码元的有序集的一个或多个来计算；以及

从包含所述输入码元和所述冗余码元的码元的组合集中生成多个输出码元，其中可能的输出码元的数量比所述码元的组合集中的码元的数量大得多，其中至少一个输出码元是从所述码元的组合集中的一个以上码元、并从所述码元的组合集中的少于全部码元的码元中生成的，

其中生成所述多个冗余码元是使用确保冗余码元中较少共同成员的确定性过程来完成的，以及

其中对于每个输入码元，存在取决于所述输入码元的预定数量的冗余码元，

其中使用所述输入码元的有序集的一个或多个输入码元生成所述多个冗余码元包括：

初始化矩阵C[K]，…C[K+S-1]，使得每个矩阵元素为零，其中K是输入码元的数量而S是将产生的冗余码元的数量，C[0]，…C[K-1]对应于前K个中间码元，其中所述中间码元是使用逆编码过程从源码元中产生的码元；

执行以下步骤，使计数器i＝0，其中a和b是中间变量，“%”表示模运算，floor()是表示小于或等于函数自变量的最高正整数值的函数，而“^”表示逐位的异或运算：

(1)a＝1+(floor(i/S)%(S-1))

(2)b＝i%S

(3)C[K+b]＝C[K+b]^C[i]

(4)b＝(b+a)%S

(5)C[K+b]＝C[K+b]^C[i]

(6)b＝(b+a)%S

(7)C[K+b]＝C[K+b]^C[i]

对计数器i从1到K-1的每个值重复这些步骤；以及

至少输出结果产生的矩阵C[K]，…C[K+S-1]，作为所述S个冗余码元，其中S是使得使S≥ceil(0.01·K)+X的最小质数，其中X是使得X·(X-1)＝2·K的最小正整数。

2.如权利要求1所述的方法，其特征在于，还包括在所述通信信道上传送所述多个输出码元。

3.如权利要求1所述的方法，其特征在于，在存储介质上存储所述多个输出码元。

4.如权利要求1所述的方法，其特征在于，所述多个冗余码元是根据LDPC码生成的。

5.如权利要求1所述的方法，其特征在于，输出码元使得所述输入码元的有序集能从任何预定数量N的所述输出码元中再生，其中N稍大于所述输入码元的数量。

6.如权利要求1所述的方法，其特征在于，所述输出码元使得所述输入码元的有序集能以高概率从N个输出码元中再生，其中N至少和所述输入码元的数量一样大。

7.如权利要求1所述的方法，其特征在于，所述输出码元使得所述输入码元的有序集的G个输入码元能从K个输出码元中再生，其中K是输入码元的数量，而G小于K。

8.如权利要求1所述的方法，其特征在于，至多G个输入码元可从任意数量的输出码元中再生，其中G比输入码元的有序集中的输入码元的数量少。

9.如权利要求1所述的方法，其特征在于，生成多个冗余码元包括，对于每个冗余码元：

根据权重分布确定t个不同的输入码元；以及

按照t个不同输入码元的异或来计算每个冗余码元。

10.如权利要求1所述的方法，还包括在所述通信信道上传送所述多个输出码元，其中生成所述多个输出码元的步骤与传送所述多个输出码元的步骤实质上并发地进行。

11.如权利要求1所述的方法，其特征在于，所述多个冗余码元包括静态码元、汉明码元和填充码元，其中码元数量之和被选为质数。

12.一种编码用于在通信信道上从源传送到目标的数据的系统，其中所述用于传送的数据是由输入码元的有序集表示的，所述系统包括：

静态编码器，配置成从所述输入码元的有序集中生成多个冗余码元，其中所述多个冗余码元中的每个冗余码元使用所述输入码元的有序集的一个或多个来计算；以及

动态编码器，可通信地耦合至所述静态编码器并配置成从包含所述输入码元和所述冗余码元的码元的组合集中生成多个输出码元，其中可能的输出码元的数量比所述码元的组合集中的码元的数量大得多，其中至少一个输出码元是从所述码元的组合集中的一个以上码元、并从所述码元的组合集中的少于全部码元的码元中生成的，

其中所述静态编码器配置成使用确保冗余码元中较少共同成员的确定性过程来生成所述多个冗余码元，以及

其中对于每个输入码元，存在取决于所述输入码元的预定数量的冗余码元，

其中所述静态编码器配置成使用所述输入码元的有序集的一个或多个输入码元生成所述多个冗余码元包括：

初始化矩阵C[K]，…C[K+S-1]，使得每个矩阵元素为零，其中K是输入码元的数量而S是将产生的冗余码元的数量，C[0]，…C[K-1]对应于前K个中间码元，其中所述中间码元是使用逆编码过程从源码元中产生的码元；

执行以下步骤，使计数器i＝0，其中a和b是中间变量，“%”表示模运算，floor()是表示小于或等于函数自变量的最高正整数值的函数，而“^”表示逐位的异或运算：

(1)a＝1+(floor(i/S)%(S-1))

(2)b＝i%S

(3)C[K+b]＝C[K+b]^C[i]

(4)b＝(b+a)%S

(5)C[K+b]＝C[K+b]^C[i]

(6)b＝(b+a)%S

(7)C[K+b]＝C[K+b]^C[i]

对计数器i从1到K-1的每个值重复这些步骤；以及

至少输出结果产生的矩阵C[K]，…C[K+S-1]，作为所述S个冗余码元，其中S是使得使S≥ceil(0.01·K)+X的最小质数，其中X是使得X·(X-1)＝2·K的最小正整数。

13.如权利要求12所述的系统，其特征在于，还包括传送模块，其可通信地耦合至所述动态编码器并配置成在所述通信信道上传送所述多个输出码元。

14.一种用于编码用于在通信信道上从源传送到目标的数据的系统，其中所述用于传送的数据是由输入码元的有序集表示的，所述系统包括：

用于从所述输入码元的有序集中生成多个冗余码元的装置，其中所述多个冗余码元中的每个冗余码元使用所述输入码元的有序集的一个或多个来计算；以及

用于从包含所述输入码元和所述冗余码元的码元的组合集中生成多个输出码元的装置，其中可能的输出码元的数量比所述码元的组合集中的码元的数量大得多，其中至少一个输出码元是从所述码元的组合集中的一个以上码元、并从所述码元的组合集中的少于全部码元的码元中生成的，

其中所述用于生成所述多个冗余码元的装置使用确保冗余码元中较少共同成员的确定性过程，以及

其中对于每个输入码元，存在取决于所述输入码元的预定数量的冗余码元，

其中所述用于使用所述输入码元的有序集的一个或多个输入码元生成所述多个冗余码元的装置包括：

用于初始化矩阵C[K]，…C[K+S-1]，使得每个矩阵元素为零的装置，其中K是输入码元的数量而S是将产生的冗余码元的数量，C[0]，…C[K-1]对应于前K个中间码元，其中所述中间码元是使用逆编码过程从源码元中产生的码元；

用于执行以下步骤，使计数器i＝0的装置，其中a和b是中间变量，“%”表示模运算，floor()是表示小于或等于函数自变量的最高正整数值的函数，而“^”表示逐位的异或运算：

(1)a＝1+(floor(i/S)％(S-1))

(2)b＝i％S

(3)C[K+b]＝C[K+b]^C[i]

(4)b＝(b+a)％S

(5)C[K+b]＝C[K+b]^C[i]

(6)b＝(b+a)％S

(7)C[K+b]＝C[K+b]^C[i]

对计数器i从1到K-1的每个值重复这些步骤；以及

至少输出结果产生的矩阵C[K]，…C[K+S-1]，作为所述S个冗余码元，其中S是使得使S≥ceil(0.01·K)+X的最小质数，其中X是使得X·(X-1)＝2·K的最小正整数。

15.一种用于解码在通信信道上收到的从源传送到目标的已编码数据的系统，所述系统包括：

接收模块，配置成接收预定数量N个码元，其中收到码元包括收到源码元和从K个源码元的有序集的多个源码元中生成的收到修复码元的组合，其中所述修复码元是众多中间码元的异或；以及

解码器，可通信地耦合至所述接收模块并配置成生成K个源码元的有序集的一个或多个未收到源码元的期望准确度，

其中每个收到码元具有由系统索引J(K)确定的相关联的码元关系，其中J(K)由K确定，其中所述系统索引管理中间码元根据所述源码元的构建规则，

其中每个未收到源码元的值由所述相关联的码元关系和多个L中间码元值确定，其中所述中间码元使用反向编码过程从所述源码元产生并且L至少为K，

其中L个中间码元值由K个源码元值和关联于K个源码元的K个码元关系以及由L-K个预

编码关系确定，以及

其中L个中间码元值可从N个收到源码元和修复码元生成期望准确度。

16.如权利要求15所述的系统，其特征在于，每个源码元具有相关联的标识所述源码元的编码码元标识符ESI，其中所述系统索引J(K)和值X，其中X是有效ESI，确定由ESI X所标识的源码元的码元关系。

17.如权利要求15所述的系统，其特征在于，L-K数量个预编码关系包括S个预编码关系的第一集合和H个预编码关系的第二集合，且其中所述L个中间码元包括K个中间码元的第一集合、S个中间码元的第二集合以及H个中间码元的第三集合。

18.如权利要求15所述的系统，其特征在于，K个源码元对应于源块，其中所述源块是由用于流送数据的传输协议定义的。

19.一种用于解码在通信信道上收到的从源传送到目标的已编码数据的方法，所述方法包括：

接收预定数量N个码元，其中收到码元包括收到源码元和从K个源码元的有序集的多个源码元中生成的收到修复码元的组合，其中所述修复码元是众多中间码元的异或；以及

生成K个源码元的有序集的一个或多个未收到源码元的期望准确度，

其中每个收到码元具有由系统索引J(K)确定的相关联的码元关系，其中J(K)由K确定，其中所述系统索引管理中间码元根据所述源码元的构建规则，

其中每个未收到源码元的值由所述相关联的码元关系和多个L中间码元值确定，其中所述中间码元是使用逆编码过程从源码元中产生的码元，并且其中L至少为K，

其中L个中间码元值由K个源码元值和关联于K个源码元的K个码元关系以及由L-K个预编码关系确定，以及

其中L个中间码元值可从N个收到源码元和修复码元生成期望准确度。

20.一种用于解码在通信信道上收到的从源传送到目标的已编码数据的系统，所述系统包括：

用于接收预定数量N个码元的装置，其中收到码元包括收到源码元和从K个源码元的有序集的多个源码元中生成的收到修复码元的组合，其中所述修复码元是众多中间码元的异或；以及

用于生成K个源码元的有序集的一个或多个未收到源码元的期望准确度的装置，

其中每个收到码元具有由系统索引J(K)确定的相关联的码元关系，其中J(K)由K确定，其中所述系统索引管理中间码元根据所述源码元的构建规则，

其中每个未收到源码元的值由所述相关联的码元关系和多个L中间码元值确定，其中所述中间码元使用反向编码过程从所述源码元产生并且L至少为K，

其中L个中间码元值由K个源码元值和关联于K个源码元的K个码元关系以及由L-K个预编码关系确定，以及

其中L个中间码元值可从N个收到源码元和修复码元生成期望准确度。

文件下载和流系统

[0001] 本发明专利申请是国际申请号为PCT/US2005/016334，国际申请日为2005年5月9日，进入中国国家阶段的申请号为200580001026.7，名称为“文件下载和流系统”的发明专利申请的分案申请。

[0002] 相关申请

[0003] 本申请要求对共同待批的于2004年5月7日提交的题为“File Download and Streaming System”的美国临时专利申请60/569,127的优先权，该申请如在本文档中完全陈述地那样通过引用结合于此。

技术领域

[0004] 本发明涉及在通信系统中编码和解码数据，尤其涉及编码和解码数据以说明传送的数据中的差错和间隔的通信系统。在各实施例中，数据通过广播和/或多点传送无线网络被传送给接收器。

背景技术

[0005] 发送器和接收器之间文件和流在通信信道上的传送已经成为许多著作的主题。较佳地，接收器想要有一定确定性地接收发送器在信道上传送的数据的准确副本。在信道不具有完美保真度时(包括大多数可全部物理实现的系统)，所关心的一方面是如何处理传输中丢失或出错的数据。丢失的数据(删除)常常比遭破坏的数据(差错)更易于处理，因为接收器常常无法分清何时遭破坏的数据是错误接收的数据。已经开发了许多纠错码，用来校正删除和/或差错。通常，所使用的特定代码是基于有关传送数据的信道的失真和所传送的数据的性质的一些信息来选择的。例如，当已知信道具有较长时段的失真时，则突发差错码(burst error code)可能最为适合该应用。在预期到仅仅较短、较少发生的差错的情况下，简单的奇偶检验码可能是最好的。

[0006] 在传送器和接收器具有通信所需的全部计算能力和电能，且传送器和接收器之间的信道足够干净以供相对无差错的通信使用时，数据传送是直接的。数据传送的问题在信道处于不利环境或者传送器和/或接收器具有有限能力时变得更加困难。

[0007] 一种方案是使用前向纠错(FEC)技术，其中数据在传送器上编码从而接收器能从传送删除和差错中恢复。在可行时，从接收器到传送器的反向信道供接收器用于与传送器进行有关差错的通信，后者可相应地调整其传送过程。然而，反向信道常常并不可用或可行。例如，当传送器向大量接收器传送时，传送器可能不能处理来自全部这些接收器的反向信道。结果，通信协议常常需要被设计成没有反向信道，从而传送器可能不得不在没有那些信道条件的全视图的情况下广泛地改变信道条件。

[0008] 传送器和接收器之间的数据传送问题在接收器需要是低功率、可便携或移动的小设备并需要以高带宽接收数据时，变得更加困难。例如，无线网络可设置成从一固定传送器向大量或不确定数量的便携式或移动接收器按照广播或者多点传送来传送文件或流，这些接收器受限于其计算能力、存储器大小、可用电功率、天线大小、设备大小和其它设计约束。

[0009] 在这种系统中，要作的考虑包括几乎没有或没有反向信道、有限的存储器、有限的计算周期、移动性和定时。较好地，设计应当最小化将数据传送给可能极多接收器所需的传送时间量，其中各个接收器可在不可预期的时间上打开和关闭、移入和移出范围、因链接错误而导致损失、单元(cell)改变、单元中的拥塞，从而使低优先级的文件或流分组暂时中断等。

[0010] 在用于数据传送的分组协议的情形中，要在分组网络上传送的文件、流、或其它数据块被分成相等大小的输入码元，且输入码元被置入连续的分组中。输入码元的“大小”可用比特来度量，而不管输入码元是否实际上被分成比特流，其中当在从2^M个码元的字母表中选择输入码元时，输入码元具有M比特的大小。在这种基于分组的通信系统中，面向分组的编码方案可能是适当的。如果允许预期的接收者即使在面对网络中的删除时也能恢复原始文件的准确副本，则文件传输被称为是可靠的。如果允许预期的接收者即使在面对网络中的删除时也能及时地恢复流的每一部分的准确副本，则流传输被称为是可靠的。在文件或流的一部分不可恢复，或者对于流而言如果流的某些部分不能及时恢复的意义上，文件传输或流传输也可以是有几分不可靠的。因为零星的拥塞使路由器中的缓存机制达到其容量限度从而迫使它丢弃输入的分组，所以常常发生分组丢失。对传输期间免遭删除的保护已是许多研究的主题。

[0011] 众所周知的是，使用连锁反应码来供从文件或流的输入码元中产生任意数量的输出码元。这具有许多用途，包括用与信息复制方法相反的信息附加方法产生输出码元，其中信息复制方法是接收器接收复制接收器已知数据的附加数据。用于产生、使用和操作连锁反应码的各种新技术在例如以下专利(或专利申请)中示出：颁发给Luby的题为“Information Additive Code Generator and Decoder for Communication Systems”(用于通信系统的信息附加码发生器和解码器)的美国专利6,307,487(“Luby I”)，颁发给Luby等人的题为“Information Additive Group Code Generator and Decoder for Communication Systems”(用于通信系统的信息附加组码发生器和解码器)的美国专利6,320,520(下文中称为“Luby II”)，以及Shokrollahi等人的于2003年3月27日公布的题为“Multi-stage Code Generator and Decoder for Communication Systems”(用于通信系统的多级码发生器和解码器)的申请号为2003/0058958的美国已公布的专利申请(下文中称为“Shokrollahi”)。在所许可的范围内，其全部揭示内容通过引用结合于此。

[0012] 连锁反应编码器所产生的输出码元的一个属性是：一接收到足够的输出码元，接收器就能恢复原始文件或原始流的块。具体地，为了较大可能地恢复原始的K个输入码元，接收器需要约K+A个输出码元。比例A/K称为“相对接收开销”。相对接收开销取决于输入码元的数量K，并取决于解码器的可靠性。例如，在一个特定实施例中，且当K等于60,000时，5%的相对接收开销确保解码器以至少$1-10^{-8}$的概率成功解码所输入的文件或流块，且当K等于10,000时，15%的相对接收开销确保解码器相同的成功概率。在一实施例中，连锁反应码的相对接收开销按照(13*sqrt(K)+200)/K来计算，其中sqrt(K)是输入码元数量K的平方根。在该实施例中，对较小值K，连锁反应码的相对接收开销倾向于较大。

[0013] Luby I,Luby II和Shokrollahi提供了可在根据本发明的某些实施例中采用的系统和方法的教示。然而，可以理解，这些系统和方法并非是本发明所必需的，并可使用许多其它变体、更改或替换。

[0014] 还众所周知的是使用多级连锁反应(“MSCR”)码,诸如在Shokrollahi中描述的和由Digital Fountain Inc.开发的商品名为“Raptor”的码。多级连锁反应码用于例如从源文件或源流中接收输入码元、从中产生中间码元、并使用连锁反应码编码中间码元的编码器。更特别地,从输入码元的有序集中产生要传送的多个冗余码元。多个输出码元从包括输入码元和冗余码元的码元的组合集中产生,其中可能的输出码元的数量比码元的组合集中码元的数量多得多,其中至少一个输出码元是从码元的组合集中的一个以上的码元、以及少于码元的组合集的全部码元的码元中产生的,从而输入码元的有序集可从任何预定数量N的输出码元中以所需准确度而重新产生。

[0015] 对于一些应用程序,码的其它变体可能更适合或更佳。

发明内容

[0016] 根据本发明一实施例,提供了编码用于在通信信道上从源传送到目标的数据的方法。该方法对输入码元的有序集操作,并包括从这些输入码元中产生多个冗余码元。该方法还包括从包含输入码元和冗余码元的码元的组合集中产生多个输出码元,其中可能的输出码元的数量比码元的组合集中码元的数量多得多,其中至少一个输出码元是从码元的组合集中的一个以上码元、以及少于码元的组合集的全部码元的码元中产生的,从而输入码元的有序集可从任何预定数量的输出码元中以所需准确度而重新产生。多个冗余码元从输入码元的有序集中产生以以下确定性过程传输,即使用第一输入码元计算的第一个静态码元集与使用不同于第一输入码元的第二输入码元计算的第二个静态码元集具有较少的共同成员。

[0017] 根据本发明的又一实施例,提供了使用相似技术接收在通信信道上从源传送的数据的系统。该系统包括与的通信信道相耦合用于接收在通信信道上传送的输出码元的接收模块,其中每个输出码元是从输入码元和冗余码元的组合集中的至少一个码元产生的,其中至少一个输出码元是从组合集中的一个以上码元、以及少于组合集的全部码元的码元中产生的,其中可能的输出码元的数量比组合集中码元的数量多得多,其中输入码元来自输入码元的有序集,其中冗余码元从输入码元中产生,且其中所述多个冗余码元是从输入码元的有序集中产生以以下确定性过程传输,即使用第一输入码元计算的第一个静态码元集与使用不同于第一输入码元的第二输入码元计算的第二个静态码元集具有较少的共同成员。

[0018] 根据本发明的再一实施例,提供了用载波实现的计算机数据信号。

[0019] 众多优点通过本发明来实现。例如,在一特定实施例中,编码用于在信道上传输的数据的计算开销下降。在另一特定实施例中,解码这种数据的计算开销下降。取决于该实施例,可实现这些优点中的一个或多个。这些和其它优点可通过本说明书在下面更详细地提供。

[0020] 对在此揭示的本发明性质和优点的进一步理解可通过参照本说明书的剩余部分和附图来实现。

附图说明

[0021] 图1是根据本发明一实施例的通信系统的框图。

[0022] 图2是根据本发明一实施例的编码器的框图。

[0023] 图3是根据本发明一实施例的产生冗余码元的方法的简化框图。

[0024] 图4是根据本发明一实施例的静态编码器的基本操作的简化框图。

[0025] 图5是根据本发明一实施例的动态编码器的简化框图。

[0026] 图6是根据本发明一实施例的动态编码器的基本操作的简化框图。

[0027] 图7是根据本发明一实施例的静态编码器的简化框图。

[0028] 图8是根据本发明一实施例的静态编码器的基本操作的简化框图。

[0029] 图9是根据静态编码器的一特定实施例的用于计算编码参数的方法的简化图。

[0030] 图10是根据本发明另一实施例的解码器的简化流程图。

[0031] 图11是根据本发明一实施例的解码器的简化框图。

[0032] 图12是根据本发明一实施例的解码器的操作的简化流程图。

[0033] 图13是根据本发明另一实施例的解码器的操作的简化流程图。

[0034] 图14是根据本发明又一实施例的解码器的操作的简化流程图。

[0035] 图15是根据本发明一实施例的动态解码器的简化框图。

[0036] 图16是根据本发明一实施例的静态解码器的简化框图。

[0037] 图17示出来自子码元映射的源码元。

[0038] 图18示出用于各种文件大小的文件下载参数的可能设置。

[0039] 图19示出用于各种源块大小的流参数的可能设置。

[0040] 图20示出表示源和中间码元之间关系的矩阵的形式。

[0041] 图21示出次数发生器的次数分布。

[0042] 图22示出可用于解码的矩阵A的形式。

[0043] 详细描述后有三个附录：附录A包含用于系统索引J(K)的示例值；附录B.1包含用于表V_0的示例值；以及附录B.2包含用于表V_1的示例值。

具体实施方式

[0044] 在以下所述的各个特定实施例中，描述表示为“多级编码”的编码方案，其各实施例在Shokrollahi中提供。

[0045] 在此所述的多级编码，以多级来编码数据。通常(但并非总是如此)，第一级向数据添加预定量的冗余。然后第二级使用连锁反应码等以从原始数据产生输出码元以及通过第一级编码计算出的冗余码元。在本发明的一特定实施例中，所接收的数据首先使用连锁反应解码过程来解码。如果该过程不能成功地完全恢复原始数据，则可应用第二个解码步骤。

[0046] 在多级编码的各个实施例中，在第一级编码期间从输入文件或流块中产生冗余码元。在这些实施例中，在第二级编码中，输出码元从输入文件或流块与冗余码元的组合中产生。在一部分这些实施例中，输出码元可按需产生。在第二级包括连锁反应编码的实施例中，每个输出码元可与其它输出码元如何产生无关地生成。一旦生成后，这些输出码元就可置入分组中并传送到其目标，其中每个分组包含一个或多个输出码元。作为替代，也可使用非分组化的传输技术。

[0047] 当在此使用时，术语“文件”指在一个或多个源上存储并作为一个单元向一个或多个目标传送的任何数据。从而，来自文件服务器或计算机存储设备的文档、图像和文件都是

能够被传送的“文件”的示例。文件可以是已知大小(诸如存储在硬盘上的一兆字节的图像)、或未知大小(诸如从流源的输出中取得的文件)。不管是哪种情形,文件都是一系列的输入码元,其中每个输入码元具有文件中的位置和值。

[0048] 当在此使用时,术语“流”涉及在一个或多个源上存储或产生、并按所产生的顺序在每个时间点上以指定速率向一个或多个目标传送的任何数据。流可以是固定速率或可变速率。因而,MPEG视频流、AMR音频流、以及用于控制远程设备的数据流都是可被传送的“流”的示例。流在每个时间点上的速率可以是已知的(诸如每秒4兆比特)或未知的(诸如每个时间点上的速率事先未知的可变速率流)。不管是哪种情形,流是一系列输入码元,其中每个输入码元都具有流中的位置和值。

[0049] 传输是为了传送文件或流而通过信道从一个或多个发送器向一个或多个接收器传送数据的过程。发送器有时也称为编码器。如果一个发送器通过完美的信道与任何数量的接收器相连,则由于将准确接收所有数据,则所接收的数据可以是输入文件或流的准确副本。在此,我们假设信道作为大多数实际信道的情形不是完美的。对于许多信道缺陷,两种感兴趣的缺陷是数据删除和数据不完整(可视为是数据删除的特定情形)。数据删除在信道丢失或丢弃数据时发生。直到一部分数据已在下列情形中传送时接收器尚未开始接收数据时,数据不完整才发生:在传输结束之前接收器停止接收数据,接收器选择仅接收所传送数据的一部分、和/或接收器间歇地停止和开始接收数据。作为数据不完整的一个示例,移动卫星发送器可传送表示输入文件或流的数据,并在接收器处于范围内之前开始该传输。一旦该接收器在范围内,则可接收数据直到该卫星移出范围,此时接收器可改变其卫星接收器的方向(在此期间它不接收数据)以开始接收由已移入范围内的另一卫星正在传送的相同输入文件或流。从阅读该说明书可显而易见,数据不完整是数据删除的特定情形,因为接收器可将数据不完整(接收器具有相同的问题)视为:好像接收器全部时间都在范围内,但信道丢失所有数据直到接收器开始接收数据。此外,如在通信系统设计中众所周知的,通过简单地丢弃具有可检测的差错的所有数据块或码元,可检测差的错可被视为等同于删除。

[0050] 在某些通信系统中,接收器接收由多个发送器(或由使用多个连接的一个发送器)产生的数据。例如,为加快下载,接收器可同时与一个以上传送关于同一文件的数据的发送器相连。作为另一示例,在多点传送传输中,可传送多个多点传送数据流以允许接收者与这些流中的一个或多个相连,从而使得合计传输速率匹配使它们与发送器相连的信道的带宽。在所有这些情形中,所关心的是确保:即使当传输速率对不同流都有很大不同时,并且当有任意模式的损失时,所有传送的数据对接收者都是独立使用的,即,多源数据在各个流之间不是冗余的。

[0051] 一般而言,通信信道连接发送器和接收器用于数据传送。该通信信道可以是实时信道,其中当信道得到数据时该信道将数据从发送器移到接收器,或者通信信道可以是存储在其从发送器到接收器的传送中的部分或全部数据的存储信道。后者的一个示例是盘存储或其他存储设备。在该示例中,产生数据的程序或设备可被视为发送器,向存储设备传送数据。接收器是从存储设备中读取数据的程序或设备。发送器用来将数据置入存储设备上的机制、存储设备本身、以及接收器用来从存储设备得到数据的机制共同形成了所述信道。如果那些机制或存储设备有丢失数据的机会,则将被视为通信信道中的数据删除。

[0052] 当发送器和接收器由其中可能删除码元的通信信道分隔时，则最好不传送输入文件或流的准确副本，而是传送从有助于恢复删除的输入文件或流(可包括输入文件或流本身的全部或部分)所产生的数据。编码器是处理该任务的电路、设备、模块或代码段。一种查看编码器的操作的方法是编码器从输入码元中产生输出码元，其中输入码元值的序列表示输入文件或流块。每个输入码元因而会具有输入文件或流块中的位置和值。解码器是从接收器接收的输出码元中重构输入码元的电路、设备、模块或代码段。在多级编码中，编码器和解码器进一步细分成执行不同任务的多个子模块。

[0053] 在多级编码系统的各个实施例中，编码器和解码器可进一步细分成子模块，每个子模块执行不同任务。例如，在一些实施例中，编码器包括在此称为静态编码器和动态编码器。当在此使用时，“静态编码器”是从输入码元集中产生若干冗余码元的编码器，其中冗余码元的数目在编码之前确定。静态编码码的示例包括里德索罗门(Reed-Solomon)码、龙卷风(Tornado)码、汉明(Hamming)码、低密度奇偶校验(LDPC)码等。术语“静态解码器”在此用来指可解码由静态编码器编码的数据的解码器。

[0054] 当在此使用时，“动态解码器”是从输入码元集产生输出码元的编码器，其中可能的输出码元的数量的数量级大于输入码元的数量，且其中要产生的输出码元的数量不需要固定。动态编码器的一个示例是连锁反应编码器，诸如Luby I和Luby II中所述的编码器。术语“动态解码器”在此用来指可解码由动态编码器编码的数据的解码器。

[0055] 多级编码的各个实施例无需受限于任何特定类型的输入码元。通常，输入码元的值是从2^M个码元的字母表中选择的，M为某一正整数。在这些情形中，输入码元可由来自输入文件或流的M比特数据序列表示。M的值常常基于例如应用的用途、通信信道、和/或输出码元的大小来确定。此外，输出码元的大小常常基于应用、信道和/或输入码元的大小确定。在一些情形中，如果输出码元值和输入码元值的大小相同(即可由相同数量的比特表示，或从同一字母表中选择)，则编码过程能得到简化。如果是这种情形，则所输入码元值大小在输出码元值大小受限时受到限制。例如，可能需要将输出码元置入有限大小的分组中。如果有关关联于输出码元的密钥的某些数据被传送以便于在接收器上恢复该密钥，则输出码元最好足够小以在一个分组中容纳输出码元值和有关密钥的数据。

[0056] 作为示例，如果输入文件是一多兆字节文件，则该输入文件可分成数千、数万、或数十万个输入码元，其中每个输入码元编码数千、数百、或仅数个字节。作为另一示例，对于基于分组的因特网信道，具有1024字节大小的有效载荷的分组可能是适当的(一个字节是8比特)。在该示例中，假设每个分组包含一个输出码元和8个字节的辅助信息，8128比特的输出码元大小((1024-8)*8)会是适当的。因而，输入码元大小可被选为M＝(1024-8)*8或8128比特。作为又一示例，一些卫星系统使用MPEG分组标准，其中每个分组的有效载荷包括188个字节。在该示例中，假设每个分组包含一个输出码元和4个字节的辅助信息，1472比特的输出码元大小((188-4)*8)会是适当的。因而，输入码元大小可被选为M＝(188-4)*8或1472比特。在使用多级编码的通用通信系统中，诸如输入码元大小(即由输入码元编码的比特数量M)的应用特定(application-specific)的参数可以是由应用设置的变量。

[0057] 作为另一示例，对于使用可变大小的源分组的流，码元大小可被选成相当小，从而每个源分组可用整数数量的输入码元覆盖，这些输入码元具有比源分组稍大一些的合计大小。

[0058] 每个输出码元都具有值。在以下考虑的一个较佳实施例中，每个输出码元还与称为其“密钥”的标识符相关联。较佳地，每个输出码元的密钥可由接收器来简便地确定，以使接收器能从其他输出码元中辨认一个输出码元。较佳地，输出码元的密钥不同于所有其他输出码元的密钥。本领域中所讨论的有各种形式的密钥方法。例如，Luby I描述可在本发明各个实施例中采用的各种形式的密钥方法。

[0059] 在预期有数据删除时或当接收器没有正好在传送开始和结束时开始和结束接收时，多级编码特别有用。后一条件在此被称为“数据不完整”。对于删除事件，多级编码享有Luby I中所述的连锁反应编码的许多优点。特别地，多级输出码元是信息附加的，因此任何适当数量的分组可用来将输入文件或流恢复到所需准确度。这些条件在使用多级编码时不会不利地影响通信过程，因为用多级编码生成的输出码元是信息附加的。例如，如果100个分组因为噪音突发而丢失从而导致数据删除，则可在该突发后拾取额外的100个分组以替换被删除分组的丢失。如果数千分组因为接收器未在开始传送时调入传送器而丢失，则接收器可从任何其它传输时段、甚至从另一传送器中拾取该数千分组。使用多级编码，接收器未受限于拾取任何特定的分组集，所以它可从一个传送器中接收一些分组、切换到另一个传送器、丢失一些分组、丢失给定传输的开始或结束、并仍然恢复输入文件或流块。无需接收器-传送器合作而加入和离开传输的能力有助于简化该通信过程。

[0060] 在一些实施例中，使用多级编码传送文件或流可包括从输入文件或流块中产生、形成或提取输入码元，计算冗余码元，将输入和冗余码元编码成一个或多个输出码元，其中每个输出码元是基于其密钥而独立于所有其它输出码元而产生的，以及在信道上将输出码元传送给一个或多个接收器。此外，在一些实施例中，使用多级编码接收(和重构)输入文件或流块的副本可包括从一个或多个数据流中接收输出码元的某个集或子集，并从接收到的输出码元的值和密钥中解码输入码元。

[0061] 在此所述的适当的FEC删除码可用来克服上述困难，并会在包括多媒体广播和多点传送系统及服务的众多领域中找到用途。下文称为“多级连锁反应码”的FEC删除码具有符合这些系统和服务的许多当前和将来的要求的属性。

[0062] 多级连锁反应码的一些基本属性是，对于任何分组丢失情形以及传送任何相关大小的源文件或任何相关速率的流：(a)每个接收器设备(“RD”)的接收开销最小化；(b)将源文件传送给任何数量的RD所需的总传输时间可最小化；(c)通过适当地选择传输时间表，传送给任何数量RD的流的质量可为相对于输入码元的数量而发送的输出码元的数量最大化。RD可以是手持式设备、可装在车辆中、可以是便携式(即可移动但使用时通常不移动)或固定在某一位置。

[0063] 解码所需的工作存储器的数量较少并仍然能提供以上属性，并且编码和解码所需的计算量是最少的。在该文档中，我们提供简便的方法来实现多级连锁反应码的一些变体的描述。

[0064] 多级连锁反应码是源泉码(fountain code)，即所需的编码分组可在工作中产生，每一个都包含等效用于恢复源文件或流块的唯一编码码元。使用源泉码而非其它类型的FEC码有许多优点。一个优点是：不管分组丢失情形和RD可用性如何，源泉码最小化每个RD需要接收以重构源文件或流块的编码分组的数量。即使在苛刻的分组丢失情形中，且当例如移动RD在长文件下载会话上仅间歇地打开或可用时，这也是真的。

[0065] 另一个优点是按需准确地产生编码分组，判断在传输进行时要在工作中产生多少编码分组。如果例如有来自RD反馈指示是否它们接收了足够的编码分组来回复源文件或流块，则这可能是有用的。当分组丢失情形没有预期的严重时，则该传输可及早予以结束。当分组丢失情形比预期的更严重时，或者RD比预期更频繁地不可用时，则该传输可无缝地延长。

[0066] 另一个优点是逆复用的能力。逆复用是当RD能组合接收到的在独立发送器上产生的编码分组时，重构源文件或流块。逆复用的一种实际用途如下参照从不同接收器接收编码分组来描述。

[0067] 在将来的分组丢失、RD可用性和应用情形难以预测的情况下，重要的是选择尽可能灵活的FEC方案以在不可预测情形下良好地工作。多级连锁反应码提供了其它类型的FEC码难以匹敌的灵活度。

[0068] 本发明的各个方面将参照附图进行描述。

[0069] 系统综述

[0070] 图1是使用多级编码的通信系统100的框图。在通信系统100中，输入文件101或输入流105被提供给输入码元发生器110。输入码元发生器110从输入文件或流中产生一个或多个输入码元(IS(0)，IS(1)，IS(2)，…)的一个序列，其中每个输入码元具有值和位置(在图1中示为加括号的整数)。如上所述，输入码元的可能值，即其字母表，通常是2^M个码元的字母表，从而每个输入码元对输入文件或流的M个比特编码。M的值通常通过使用通信系统100确定，但通用系统可包括对输入码元发生器110的码元大小输入，从而M可在使用时变化。输入码元发生器110的输出被提供给编码器115。

[0071] 静态密钥发生器130产生静态密钥S_0、S_1……的流。所产生静态密钥的数量通常受到限制并取决于编码器115的特定实施例。静态密钥的产生随后将作更详细的描述。动态密钥发生器120对要由编码器115产生的每个输出码元产生动态密钥。每个动态密钥被生成为同一输入文件或流块的大部分动态密钥都是唯一的。例如，Luby I描述了可使用的密钥发生器的各个实施例。动态密钥发生器120和静态密钥发生器130的输出被提供给编码器115。

[0072] 根据动态密钥发生器120提供的每个密钥I，编码器115从输入码元发生器提供的输入码元中产生值为B(I)的输出码元。编码器115的操作将在后面详细描述。根据每个输出码元的密钥、一个或多个输入码元的某一函数、以及可能根据已从输入码元计算出的一个或多个冗余码元，产生每个输出码元的值。产生特定输出码元的输入码元和冗余码元的集合在此被称为输出码元的“关联码元”或仅称为其“关联”。函数(“值函数”)和关联的选择是根据以下更详细描述的过程来完成的。通常，但并非总是这样，M对输入码元和输出码元是相同的，即它们都对相同数量的比特编码。

[0073] 在一些实施例中，输入码元的数量K由编码器115用来选择关联。如果K为事先未知，诸如输入为流式文件的情况，则K仅是估计值。值K还可由编码器115用来分配对输入码元和由编码器115产生的任何中间码元的存储。

[0074] 编码器115将输出码元提供给传送模块140。传送模块140还被供以来自动态密钥发生器120的每个输出码元的密钥。传送模块140传送输出码元，并取决于所使用的密钥方法，传送模块140还可在信道145上向接收模块150传送有关所传送的输出码元的密钥的一些数据。信道145假设为删除信道，但并非是通信系统100的正常操作所必需的。模块140、

145和150可以是任何适当硬件组件、软件组件、物理介质、或其任何组合，只要传送模块140适于向信道145传送输出码元和有关其密钥的任何所需数据，而接收模块150适于从信道145接收码元和可能的有关其密钥的一些数据。K的值，如果用来确定关联，则可在信道145上发送，或者它按照编码器115和解码器155达成的协议预先设置。

[0075] 如上所述，信道145可以是实时信道，诸如通过因特网的路径、或者从电视传送器到电视接收器的广播链接、或者从一个点到另一个点的电话连接，或者信道145可以是存储信道，诸如CD-ROM、盘驱动器、网站等。信道145甚至可以是实时信道和存储信道的组合，诸如当一个人通过电话线将输入文件从个人计算机传送到因特网服务供应商(ISP)、输入文件存储在Web服务器上并随后通过因特网传送给了接收器时形成的信道。

[0076] 因为信道145被假设为删除信道，所以通信系统100不假定从接收模块150出来的输出码元与进入传送模块140的输出码元之间的一对一对应性。实际上，当信道145包括分组网络时，通信系统100甚至可能不能假设任意的两个或多个分组的相对顺序在通过信道145传送时得以保留。因此，输出码元的密钥是使用上述一个或多个密钥方案来确定的，并且无需按照输出码元从接收模块150出来的顺序来确定。

[0077] 接收模块150向解码器155提供输出码元，并且接收模块150接收的有关这些输出码元的密钥的任何数据被提供给动态密钥再生器160。动态密钥再生器160重新产生接收到的输出码元的动态密钥，并将这些动态密钥提供给解码器155。静态密钥发生器163重新产生静态密钥S_0、S_1……并将它们提供给解码器155。静态密钥发生器可访问在编码和解码过程期间使用的随机数发生器135。这可以是如果随机数在物理设备上产生则访问该相同设备的形式，或者访问产生随机数的相同算法以实现同样动作的形式。解码器155使用动态密钥再生器160和静态密钥发生器163提供的密钥、以及相应的输出码元，以恢复输入码元(再一次为IS(0)，IS(1)，IS(2)，…)。解码器155向产生输入文件101或输入流105的副本170的输入文件重组器165提供恢复的输入码元。

[0078] 编码器

[0079] 图2是图1中示出的编码器115的一个特定实施例的框图。编码器115包括静态编码器210、动态编码器220、和冗余计算器230。静态编码器210接收以下输入：a)由输入信号发生器110提供的并存储在输入码元缓冲器205中的原始输入码元IS(0)，IS(1)，……IS(K-1)；b)原始输入码元的数量K；c)由静态密钥发生器130提供的静态密钥S_0，S_1，……；以及d)R个冗余码元。在接收这些输入之后，静态编码器205如以下所述地计算R个冗余码元RE(0)，RE(1)，……，RE(R-1)。通常，但并非总是如此，冗余码元具有与输入码元相同的大小。在一特定实施例中，静态编码器210产生的冗余码元存储在输入码元缓冲器205中。输入码元缓冲器205可以只是逻辑的，即，文件或流块可物理地存储在一个地方，且输入码元在码元缓冲器205内的位置只是这些码元在原始文件或流块内位置的重命名。

[0080] 动态编码器接收输入码元和冗余码元并产生输出码元，如下更详细地描述。在冗余码元存储于输入码元缓冲器205的一个实施例中，动态编码器220从输入码元缓冲器205中接收输入码元和冗余码元。

[0081] 冗余计算器230从K个输入码元中计算R个冗余码元。该计算在后面进行更详细的描述。

[0082] 静态编码器的综述

[0083] 静态编码器210的一般操作参照图3和4示出。图3是示出静态编码方法的一个实施例的简化流程图。在步骤305，跟踪已产生多少冗余码元的变量j被设置为0。然后，在步骤310，码元按照至少一些输入码元IS(0)，……IS(K-1)的函数F_0计算第一冗余码元RE(0)。然后，在步骤315，变量j增1。然后，在步骤320，测试是否已产生了全部的冗余码元(即j>R-1?)如果是，则流程结束。否则，流程继续到步骤325。在步骤325，按照输入码元IS(0)，……，IS(K-1)和先前产生的冗余码元RE(0)，……，RE(j-1)的函数F_j来计算RE(j)，其中F_j无需是依赖于每一个输入码元或每一个冗余码元的函数。重复步骤315,320和325直到已计算了R个冗余码元。

[0084] 再参看图1和2，在一些实施例中，静态编码器210从静态密钥发生器130中接收一个或多个静态密钥S_0,S_1,……。在这些实施例中，静态编码器210使用静态密钥来确定部分或全部函数F_0,F_1,……,F_{j-1}。例如，静态密钥S_0可用来确定函数F_0，静态密钥S_1可用来确定函数F_1等。或者，一个或多个静态密钥S_0,S_1,……可用来确定函数F_0，一个或多个静态密钥S_0,S_1,……可用来确定函数F_1等。在其它实施例中，不需要静态密钥，因而也不需要静态密钥发生器130。

[0085] 现在参看图2和3，在一些实施例中，静态编码器210产生的冗余码元可存储在输入码元缓冲器205中。图4是静态编码器210的一个实施例的操作的简化视图。特别地，静态编码器210产生码元按照从输入码元缓冲器205接收的输入码元IS(0)，……，IS(K-1)，RE(0)，……，RE(j-1)的函数来产生冗余码元RE(j)，并将其存储回输入码元缓冲器205。函数F_0,F_1,……,F_{R-1}的准确形式取决于特定应用。通常，但并非总是如此，函数F_0,F_1,……,F_{R-1}包括部分或全部相应自变量的异或。如上所述，这些函数实际上可采用或不采用图1的静态密钥发生器130所产生的静态密钥。例如，在以下所述的一个特定实施例中，前几个函数实现汉明码而不使用静态密钥S_0,S_1……，而剩下的函数实现低密度奇偶校验码并使用静态密钥。

[0086] 多级编码器的综述

[0087] 再参看图2，动态编码器220接收输入码元IS(0)，……，IS(K-1)和冗余码元RE(0)，……，RE(R-1)，以及它要对每个输出码元产生的密钥I。包括原始输入码元和冗余码元的集合将在下文中称为“动态输入码元”的集合。图5是动态编码器的一个实施例的简化框图，包括权重选择器510、关联器515、值函数选择器520和计算器525。如图5所示，K+R个动态输入码元被存储在动态码元缓冲器505中。实际上，动态编码器500执行图6所示的动作，即，按照选定输入码元的一些值函数产生输出码元值B(I)。

[0088] 图7是根据本发明的静态编码器的一特定实施例的简化框图。静态编码器600包括参数计算器605、汉明编码器610、以及低密度奇偶校验(LPDC)编码器620。汉明编码器610被耦合成接收来自输入码元缓冲器625的输入码元IS(0)，……，IS(K-1)、输入码元个数K、和参数D。作为响应，汉明编码器610根据汉明码产生D+1个冗余码元HA(0)，HA(1)，……，HA(D)。

[0089] 图8示出采用图7中示出的静态编码器的本发明的一个实施例的操作。

[0090] 图9是示出参数计算器的一个实施例的简化流程图，诸如图7的如上所述计算参数D和E的参数计算器605。首先，在步骤705，参数D被初始化为1。然后，在步骤710，确定2^D-D-1是否小于K。如果不是，则流程继续到步骤730。如果是，则流程继续到步骤720，其中参数D增

1。然后，流程回到步骤710。一旦已确定D，则在步骤730，参数按照R-D-1计算参数E。

[0091] 图10是根据本发明一实施例的这种编码器的简化流程图，如下所述。首先，在步骤805，变量i被初始化为0。变量i跟踪已产生冗余码元的数量。在步骤810，按照大于或等于K/2的最小奇数来计算数字t。在步骤815，值P_1，P_2，……，P_t基于K、t和静态密钥S_i产生。值P_1，P_2，……，P_t指示将用来产生冗余码元的输入码元的位置。在一特定实施例中，诸如图5的关联器515的关联器被用来产生P_1，P_2，……，P_t。特别地，值t可被提供为W(I)输入，值K可被提供为K+R输入，而静态密钥S_i可被提供为密钥I输入。应注意，许多不同的值t会产生相似的编码效果，因而该特定选择仅是一示例。在步骤820中，按照值IS(P_1)，IS(P_2)，……，IS(P_t)的XOR来计算RE(i)。在步骤825，变量i增1以准备计算下一冗余码元，且在步骤830确定是否已计算了全部冗余码元。如果不是，则流程返回步骤815。

[0092] 图11是示出根据本发明的解码器的一个实施例的简化框图。解码器900可用来例如实现图1的解码器155。

[0093] 解码器900包括动态解码器905和静态解码器910。动态解码器905所恢复的输入码元和冗余码元被存储在重组缓冲器915中。在完成动态解码之后，静态解码器910尝试恢复动态解码器905尚未恢复的任何输入码元(如果有的话)。特别地，静态解码器910从重组缓冲器915中接收输入码元和冗余码元。

[0094] 图12是示出根据本发明的用于解码的方法的一个实施例的简化流程图。在步骤1005，Q个输出码元由解码器接收。Q的值取决于输入码元的数量和所使用的特定动态编码器。Q的值还取决于解码器能恢复输入码元的所期望的准确度。例如，如果希望解码器可较大可能地恢复全部输入码元，则Q应选为比输入码元的数量大。特别地，在一些应用中，当输入码元的数量较大时，Q可比原始输入码元的数量大3％不到。在其它应用中，当输入码元的数量较小时，Q可比输入码元的数量大至少10％。具体地，Q可选为输入码元的数量K加上数A，其中A被选为确保解码器可较大可能地再生全部输入码元。数A的确定在下面更详细地描述。如果对解码器而言(有时或总是)不能解码全部输入码元是可接受的，则Q可小于K+A、等于K、或者甚至小于K。显然，整体编码系统的一个目的常常是尽可能地减小数Q，同时保持对参照所需准确度的解码过程的成功的较大概率保证。

[0095] 在步骤1010，动态解码器905从所接收的Q个输出码元中重新产生输入码元和冗余码元。可以理解，步骤1005和1010实际上可并发执行。例如，动态解码器905可在解码器接收Q个输出码元之前再生输入码元和冗余码元。

[0096] 在动态解码器905已处理Q个输出码元之后，确定输入码元是否已恢复到期望准确度。期望准确度可以是例如全部输入码元、或少于全部输入码元的部分数量或百分比等。如果是，则该流程结束。如果不是，则流程继续到步骤1020。在步骤1020，静态解码器910尝试恢复动态解码器905不能恢复的任何输入码元。在静态编码器910已处理了动态编码器905所恢复的输入码元和冗余码元之后，流程结束。

[0097] 图13是示出根据本发明的用于解码的方法的另一实施例的简化流程图。该实施例与参照图11所述的相似，并包括共同步骤1005、1010、1015和1025。但是，在步骤1025之后，流程继续到步骤1030，其中确定输入码元是否已恢复到期望准确度。如果是，则流程结束。如果不是，则流程继续到步骤1035。在步骤1035，接收一个或多个额外的输出码元。然后，流程返回步骤1010，从而动态解码器905和/或静态解码器910可尝试恢复剩余的未恢复的输

入码元。

[0098] 图14是根据本发明的用于解码的方法的又一实施例的简化流程图。在步骤1055，输出码元由解码器接收，而在步骤1060，动态解码器905从所接收的输出码元中再生输入码元和冗余码元。然后，在步骤1065，确定动态解码是否应结束。该确定可基于所处理的输出码元的数量、所恢复的输入码元的数量、其它输入码元当前的恢复速率、处理输出码元所花的时间等的一个或多个。

[0099] 在步骤1065，如果确定不要终止动态解码，则流程继续回到步骤1055。但是，如果在步骤1065确定结束动态解码，则流程继续到步骤1070。在步骤1070，确定输入码元是否已恢复到期望准确度。如果是，则流程结束。如果不是，则流程继续到步骤1075。在步骤1075，静态解码器910尝试恢复动态解码器905未能恢复的任何输入码元。在静态解码器910已处理了动态解码器905所恢复的输入码元和冗余码元之后，流程结束。

[0100] 图15示出根据本发明的动态解码器的一个实施例。动态解码器1100包括与图5所示的动态编码器相似的组件。解码器1100与Luby I和Luby II中所述的连锁反应解码器的实施例相似。动态解码器1100包括权重选择器510、关联器515、值函数选择器520、输出码元缓冲器1105、缩减器1115、重组器1120和重组缓冲器1125。

[0101] 图16是示出静态解码器的一个实施例的简化框图。该实施例可在数据用诸如参照图7所述的静态编码器编码时使用。静态解码器1200包括LDPC解码器1205和汉明解码器1210。LDPC解码器1205从重组缓冲器1215中接收输入码元和冗余码元，并尝试重组在动态解码器的解码步骤之后重组缓冲器1215未恢复的那些码元。在一些实施例中，重组缓冲器1215是重组缓冲器1125(图15)。

[0102] LDPC解码器和汉明解码器的许多变体都是本领域技术人员所众所周知的，并可在根据本发明的各个实施例中采用。在一特定实施例中，汉明解码器使用高斯消元算法来实现。高斯消元算法的许多变体是本领域技术人员众所周知的，并可在根据本发明的各个实施例中采用。

[0103] 变体

[0104] 如上所述的多级连锁反应码不是系统码，即，源块的全部原始源码元没有必要在所发送的编码码元中。然而，系统FEC码对文件下载系统或服务有用，并且对流系统或服务是非常重要的。如在以下实现中所示，更改后的码可变成是系统的，并仍然保持源泉码和其它所述属性。

[0105] 使用多级码便于构建多种补充服务的一个原因是：它可组合接收到的来自多个发送器的编码码元以重构源文件或流，而无需在各发送器之间进行协调。唯一的要求是发送器使用不同的密钥集来产生它们用编码分组发送给码的编码码元。实现方法包括：指定每个发送器要使用的不同的密钥空间范围，或在每个发送器上随机产生密钥。

[0106] 作为使用该能力的一个示例，考虑向文件下载服务提供一补充服务，以允许未从文件下载会话中接收足够的重构源文件的编码分组的多级连锁反应码请求例如通过HTTP会话从组成发送器中发送其它编码分组。组成发送器从源文件中产生编码码元并例如使用HTTP发送它们，并且所有这些编码码元都能与从文件下载会话中接收的编码码元相组合，以恢复该源文件。使用该方法使不同的发送器能提供增量式源文件传送服务而无需在发送器之间进行协调，并确保每个接收器仅需要接收最少量的编码分组，以恢复每个源文件。

[0107] 多级码的各级的实现

[0108] FEC方案定义

[0109] 使用这些技术的分组可呈现为具有标头信息，诸如包括源块号码(SBN)(分组内编码码元与之相关的源块的16比特整数标识符)、以及编码码元ID(ESI)(分组内编码码元的16比特整数标识符)的4个八位字节的FEC有效负载ID。源块号码和编码码元标识符的一个适当解释在以下小节B中定义。FEC对象传输信息可包括FEC编码ID、传送长度(F)和以下定义的参数T、Z、N和A。参数T和Z是16比特的无符号整数，N和A是8比特的无符号整数。

[0110] 用于MBMS前向纠错的FEC编码方案在以下各小节中定义。它定义两种不同的FEC有效载荷ID格式，一种用于FEC源分组而另一种用于FEC修复分组，但非系统码的变体也是可能的。

[0111] 源FEC有效载荷ID可包括源块号码(SBN)(分组内编码码元与之相关的源块的16比特整数标识符)、以及编码码元ID(ESI)(分组内编码码元的16比特整数标识符)，而修复FEC有效载荷ID可包括源块号码(SBN)(分组内修复码元与之相关的源块的16比特整数标识符)、编码码元ID(ESI)(分组内修复码元的16比特整数标识符)、以及源块长度(SBL)(16比特，表示源码元在源块中的数量)。源块号码、编码码元标识符和源块长度如下作定义。

[0112] FEC对象传输信息可包括FEC编码ID、以码元为单位的最大源块长度、以及以字节为单位的码元大小。码元大小和最大源块长度可包括4个八位字节的码元大小(T)(表示以字节为单位的编码码元的大小的16比特)、以及最大源块长度(表示以码元为单位的源块的最大长度的16比特)。

[0113] 以下小节指定系统MSCR前向纠错码及其在MBMS中的应用和其它用途。MSCR是源泉码，即，按所需那么多的编码码元可由编码器在工作中从块的源码元中产生。解码器能从数量上仅比源码元数量略多一点的任何编码码元集中恢复源块。在该文档中描述的码是系统码，即原始源码元以及众多修复码元未经更改地从发送器发送到接收器。

[0114] B.1.定义、码元和缩写

[0115] B.11定义

[0116] 为了本说明书的目的，应用以下术语和定义。

[0117] 源块：为MSCR编码目的考虑的K个源码元的块。

[0118] 源码元：在编码过程期间使用的最小单位的数据。源块内的所有源码元都具有相同的大小。

[0119] 编码码元：包括在数据分组中的码元。编码码元包括源码元和修复码元。从源块产生的修复码元与该源块的源码元大小相同。

[0120] 系统码：其中源码元被包括作为对源块发送的编码码元的一部分的码。

[0121] 修复码元：为不是源码元的对源块发送的编码码元。修复码元基于源码元产生。

[0122] 中间码元：使用逆编码过程从源码元中产生的码元。然后修复码元直接从中间码元中产生。编码码元不包括中间码元，即，中间码元不包括在数据分组中。

[0123] 码元：数据单位。以字节为单位的码元大小被称为码元大小。

[0124] 编码码元组：一起发送(即在同一分组内)的编码码元组，其与源码元的关系可从单个编码码元ID中导出。

[0125] 编码码元ID：定义编码码元组的各个码元和源码元之间关系的信息。

[0126] 编码分组：包含编码码元的数据分组。

[0127] 子块：源块有时被分成子块，每个子块都足够小以在工作存储器中被解码。对于包括K个源码元的源块，每个子块包括K个子码元，源块的每个码元由来自每个子块的一个子码元构成。

[0128] 子码元：码元的一部分。每个源码元由像源块中的子块一样多的子码元构成。

[0129] 源分组：包含源码元的数据分组。

[0130] 修复分组：包含修复码元的数据分组。

[0131] B.1.2符号

i,j,x,h,a,b,d,v,m	表示正整数
ceil(x)	表示大于或等于 x 的最小正整数
choose(i,j)	表示可从 i 个对象中不重复地选择 j 个对象的方法的数量
floor(x)	表示小于或等于 x 的最大正整数
i%j	表示 i 模 j
X^Y	表示对于相等长度的比特串 X 和 Y，逐个比特地异或 X 和 Y。
A	表示码元对齐参数。码元和子码元的大小都受限于 A 的倍数。
A^T	表示矩阵 A 的转置矩阵
A^{-1}	表示矩阵 A 的逆矩阵
K	表示单个源块中的码元数量
K_{MAX}	表示可在单个源块中的源码元的最大数量。设置成 8192。
L	表示单个源块的预编码码元的数量。
S	表示单个源块的 LDPC 码元的数量。
H	表示单个源块的半码元的数量。
C	表示中间码元的数组，C[0], C[1], C[2], ……, C[L-1]
C’	表示源码元的数组，C’[0], C’[1], C’[2], ……, C’[K-1]
X	非负整数值
V_0,V_1	4-字节整数的两个数组，V_0[0], V_0[1], V_0[2], ……, V_0[255]和 V_1[0], V_1[1], V_1[2], ……, V_1[255]

[0132]

Rand[X, i, m]	伪随机数发生器
Deg[v]	次数发生器
LTEnc[K, C, (d, a, b)]	LT 编码码元发生器
Trip[K, X]	三元发生器函数
G	编码码元组内码元的数量
N	源块内子块的数量
T	以字节为单位的码元大小。如果源块不分成子块，则 T=T’·N。
T’	以字节为单位的子码元大小。如果源块不分成子块，则 T’是不相关的。
F	以字节为单位的用于文件下载的文件大小
I	以字节为单位的子块大小
P	对文件下载，以字节为单位的每个分组的有效载荷大小，用于文件下载传输参数的推荐导出。对于流，以字节为单位的每个修复分组的有效载荷大小，用于流传输参数的推荐导出。
Q	Q=65521，Q 是比 2^{16} 小的最大质数。
Z	用于文件下载的源块数量
J(K)	与 K 相关联的系统索引
G	表示任何发生器矩阵
I_S	表示 SxS 单位矩阵
0_{SxH}	表示 SxH 零矩阵

[0133] B.1.3缩写

[0134] 以下缩写用于本文档：

[0135]

ESI	编码码元ID
LDPC	低密度奇偶校验
LT	Luby变换
SBN	源块号码

[0136]

SBL	源块长度(以码元为单位)

[0137] B.2.综述

[0138] MSCR前向纠错码可应用于MBMS文件传送和MBMS流应用。对每个这些应用特定的MSCR码方面在本文档的小节B.3和B.4中讨论。

[0139] 系统MSCR码的构成是小节B.5中所述的基本编码器。首先，描述如何从原始源码元中导出中间码元集的值，使得知道了中间码元就足以重构源码元。其次，编码器产生修复码元，它们是众多中间码元的异或。编码码元是源和修复码元的组合。修复码元被生成为中间码元和因此源码元可从任何足够大的编码码元集中恢复。

[0140] 本文档定义系统MSCR码编码器。众多可能的解码算法是可能的。一种有效的解码算法在小节B.6中提供。

[0141] 中间和修复码元的构建部分地基于小节B.5中所述的伪随机数发生器。该发生器基于可用于发送器和接收器的512个随机数的一个固定集。一个示例数集如附录B.1中所提供。

[0142] 最终，中间码元根据源码元的构建受“系统索引”管理。系统索引的值的示例集如附录A中所示，用于从4个源码元到K_{MAX}＝8192个源码元的源块大小。

[0143] B.3.文件下载

[0144] B.3.1源块构建

[0145] B.3.1.1概述

[0146] 为了将MSCR编码器应用于源文件，文件可分成Z≥1块，称为源块。MSCR编码器独立地应用于每个源块。每个源块由唯一的整数源块号码(SBN)标识，其中第一源块具有SBN 0，第二源块具有SBN 1等等。每个源块被分成K个大小为T个字节的源码元。每个源码元由唯一的整数编码码元标识符(ESI)标识，其中源块的第一源码元具有ESI 0，第二源码元具有ESI 1等。

[0147] 每个具有K个源码元的源块被分成N≥1个子块，子块足够小以在工作存储器中被解码。每个子块被分成大小为T’的K个子码元。

[0148] 注意，K值对文件的每个源块没有必要是相同的，而T’值对源块的每个子块也没有必要是相同的。然而，码元大小T对文件的所有源块都是相同的，且码元数K对源块的每个子块都相同。文件到源块和子块的确切划分如下在B.3.1.2中所述。

[0149] 图17示出置入两维数组的一示例源块，其中每个条目是T’字节的子码元，每行是子块而每列是源码元。在该示例中，T’值对每个子块都相同。每个子码元条目中所示的数字指示其在源块内的原始顺序。例如，标号为K的子码元包含该源块的T’·K～T’·(K+1)-1字

节。然后，源码元i是来自每个子块的第i个子码元的串联拼接，它对应于标号为i,K+i,2·K+i,……,(N-1)·K+i的源块的子码元。

[0150] B.3.1.2源块和子块划分

[0151] 源块和子块的构建基于5个输入参数F、A、T、Z和N以及函数Partition[]来确定。5个输入参数定义如下：

[0152] F 以字节为单位的文件的大小

[0153] A 以字节为单位的码元对齐参数

[0154] T 以字节为单位的必须是A的倍数的码元大小

[0155] Z 源块的数量

[0156] N 每个源块中子块的数量

[0157] 这些参数可设置成使ceil(ceil(F/T)/Z)≤K_{MAX}。推导这些参数的推荐在小节B.3.4中提供。

[0158] 函数Partition[]取一对整数(I,J)为输入，并导出4个整数(I_L,I_S,J_L,J_S)作为输出。具体地，Partition[I,J]是4个整数(I_L,I_S,J_L,J_S)的一个序列，其中I_L=ceil(I/J)，I_S=floor(I/J)，J_L=I-I_S·J和J_S=J-J_L。Partition[]导出用于将块的大小I分成J个大小近似相等的块的参数。具体地，J_L个长度为I_L的块以及J_S个长度为I_S的块。

[0159] 源文件可如下分成多个源块和子块：

[0160] 设K_t=ceil(F/T)

[0161] (K_L,K_S,Z_L,Z_S)=Partition[K_t,Z]

[0162] (T_L,T_S,N_L,N_S)=Partition[T/A,N]

[0163] 然后，文件可被分成Z=Z_L+Z_S个连续源块，前Z_L个源块每个都具有长度K_L·T字节，而剩下的Z_S个源块每个都具有K_S·T字节。

[0164] 如果K_t·T>F，则出于编码目的，最后一个码元可在末端用K_t·T-F个零字节填充。

[0165] 然后，每个源块可分成N=N_L+N_S个连续子块，前N_L个子块每个都包括大小为T_L·A的K个连续子码元，而剩下的N_S个子块每个都包括大小为T_S·A的K个连续子码元。码元对齐参数A确保子码元总是A的倍数字节。

[0166] 最后，源块的第m个码元包括来自N个子块的每一个的第m个子码元的串联拼接。

[0167] B.3.2编码分组构建

[0168] B.3.2.1概论

[0169] 每个编码分组包含以下信息：

[0170] 源块号码(SBN)

[0171] 编码码元ID(ESI)

[0172] 编码码元(s)

[0173] 每个源块独立于其它源块编码。源块从0开始连续编号。

[0174] 从0到K-1的编码码元ID值标识各源码元。K以上的编码码元ID标识修复码元。

[0175] B.3.2.2编码分组构建

[0176] 每个编码分组最好完全由源码元构成(源分组)，或者完全由修复码元构成(修复分组)。分组可包含来自相同源块的任何数量的码元。在分组中最后一个码元包括为FEC编码目的添加的填充字节的情形中，这些字节不需要包括在分组中。否则，只可包括完整的码

元。

[0177] 在每个源分组中携带的编码码元ID X是在该分组中携带的第一源码元的编码码元ID。该分组中的后续的源码元具有按顺序排列的编码码元ID X+1～X+G-1，其中G是分组中码元的数量。

[0178] 类似地，置入修复分组中的编码码元ID X是修复分组中第一修复码元的编码码元ID，且该分组中的后续的修复码元具有按顺序排列的编码码元ID X+1～X+G-1，其中G是分组中码元的数量。

[0179] 注意，接收器不需要知道修复分组的总数量。置入具有ESI X的修复分组的修复码元的G个修复码元三元组(d[0],a[0],b[0]),……,(d[G-1],a[G-1],b[G-1])使用B.5.3.4中定义的三元发生器计算如下：

[0180] 对于每个i＝0,……,G-1

[0181] (d[i],a[i],b[i])＝Trip[K,X+i]

[0182] 要置入具有ESI X的修复分组的G个修复码元使用中间码元C和LT编码器LTenc[K,C,(d[i],a[i],b[i])]如小节B.5.3所述基于修复码元三元组来计算。

[0183] B.3.3传输

[0184] 本节描述使用对文件传送的MSCR前向纠错在MSCR编码器/解码器和任何传输协议之间的信息交换。

[0185] 用于文件传送的MSCR编码器和解码器需要来自传输协议的以下信息：以字节为单位的文件大小F、码元对齐参数A、以字节为单位的是A倍数的码元大小T、源块数量Z、每个源块中的子块数量N。用于文件传送的MSCR编码器还需要要被编码的文件，F字节。

[0186] MSCR编码器向传输协议提供编码分组信息，包括每个分组的SBN、ESI和编码码元。传输协议可将该信息透明地传送给MSCR解码器。

[0187] B.3.4推荐参数(提供信息的)

[0188] B.3.4.1参数导出算法

[0189] 本节提供对导出4个传送参数A、T、Z和N的推荐。该推荐基于以下输入参数：

[0190] F 以字节为单位的文件大小

[0191] W 以字节为单位的对子块大小的目标

[0192] P 以字节为单位的最大分组有效载荷大小，假设为A的倍数

[0193] A 以字节为单位的码元对齐因数

[0194] K_{MAX} 每个源块源码元的最多数量

[0195] K_{MIN} 对每个源块码元数量的最小目标

[0196] G_{MAX} 每个分组码元的最多目标数量

[0197] 基于以上输入，传送参数T、Z和N计算如下：

[0198] 设G＝min{ceil(P·K_{MIN}/F),P/A,G_{MAX}}-每个分组码元的大约数量

[0199] T＝floor(P/(A·G))·A

[0200] K＝ceil(F/T)-文件中码元的总量

[0201] Z＝ceil(K_t/K_{MAX})

[0202] N＝min{ceil(ceil(K_t/Z)·T/W),T/A}

[0203] 以上导出的G和N值都应视为下限。将这些值增大为例如最接近的2的幂是有利的。

特别地，以上算法不能确保码元大小T能被最大分组大小P除尽，因此不可能使用大小精确为P的分组。如果相反，G被选为能被P/A除尽的值，则码元大小T将是P的除数并可使用大小为P的分组。

[0204] 输入参数W、A、K_{MIN}和G_{MAX}的推荐设置如下：

[0205] W＝256KB A＝4 K_{MIN}＝1024 G_{MAX}＝10

[0206] B.3.4.2示例

[0207] 以上算法导出如图18所示的传送参数，假设W、A、K_{MIN}和G_{MAX}的推荐值且P＝512。

[0208] B.4.流

[0209] B.4.1源块构建

[0210] 源块由例如本文档中定义的传送协议使用系统MSCR前向纠错码构建。用于源块构建和修复码元构建的码元大小T由传送协议提供。参数T可设置成任何源块中源码元的数量最多为K_{MAX}。

[0211] 推荐参数如小节B.4.4所示。

[0212] B.4.2编码分组构建

[0213] 如B.4.3中所述，每个修复分组都包含SBN、ESI、SBL和修复码元。包含在修复分组内的修复码元的数量根据分组长度计算。置入修复分组的ESI值和用来产生修复码元的修复码元三元组如小节B.3.2.2所述进行计算。

[0214] B.4.3传送

[0215] 本节描述MSCR编码器/解码器和任何对流使用MSCR前向纠错的传输协议之间的信息交换。用于流的MSCR编码器可对每个源块使用以下来自传输协议的信息：以字节为单位的码元大小T、源块中的码元数量K、源块号码(SBN)和要编码的源码元K·T字节。MSCR编码器向传输协议提供编码分组信息，包括每个修复分组的SBN、ESI、SBL和修复码元。该传输协议可将该信息透明地传送给MSCR解码器。

[0216] B.4.4.推荐参数

[0217] ___B.4.4.1参数导出算法

[0218] 本节提供对传送参数T的推导的推荐。该推荐基于以下输入参数：

[0219]

B	以字节为单位的最大源块大小
P	以字节为单位的最大修复分组有效载荷大小，为A的倍数
A	以字节为单位的码元对齐因数
K_{MAX}	每个源块中的源码元的最大数量
K_{MIN}	对每个源块中码元数量的最小目标
G_{MAX}	每个修复分组中码元的最大目标数量

[0220] 对这些参数的要求是$ceil(B/P) \leq K_{MAX}$。基于以上输入，传送参数T计算如下：

[0221] 设$G=\min\{ceil(P \cdot K_{MIN}/B), P/A, G_{MAX}\}$-每个分组码元的大约数量

[0222] $T=floor(P/(A \cdot G)) \cdot A$

[0223] 以上导出的T值应被视为所使用实际T值的基准。确保T能被P除尽是有利的，或者将T值设置得更小以在全尺寸大小的修复码元用来在丢失的源分组的末端恢复部分源码元时使浪费最少(只要源块中源码元的最多数量不超过K_{MAX})是有利的。此外，对T的选择可取

决于源分组大小的分布，即如果全部源分组大小都相同，则将T选成修复分组P'的实际有效载荷大小(其中P'是T的倍数)等于每个源分组占据源块的字节数(或者比之大尽可能少的字节)是有利的。

[0224] 输入参数A、K_{MIN}和G_{MAX}的推荐设置如下：

[0225] A＝4 K_{MIN}＝1024 G_{MAX}＝10

[0226] B.4.4.2示例

[0227] 以上算法导出如图19所示的传送参数，假设A、K_{MIN}和G_{MAX}的推荐值且P＝512。

[0228] B.5.系统MSCR编码器

[0229] B.5.1.编码综述

[0230] 系统MSCR编码器用来从包括K个源码元的源块中产生修复码元。

[0231] 码元是编码和解码过程的基本数据单位。对于每个源块(子块)，所有码元(子码元)大小都相同。对码元(子码元)执行的编码和解码的原子运算是异或运算。

[0232] 设C'[0]，……，C'[K-1]表示K个源码元。

[0233] 设C[0]，……，C[L-1]表示L个中间码元。

[0234] 编码的第一个步骤是从K个源码元中产生L>K个中间码元。在该步骤中，K个源三元组(d[0]，a[0]，b[0])，……，(d[K-1]，a[K-1]，b[K-1])使用小节B.5.4.4中所述的Trip[]发生器产生。K个源三元组与K个源码元相关联，然后使用逆编码过程用来从源码元中确定L个中间码元C[0]，……，C[L-1]。该过程可由MSCR解码过程实现。

[0235] 一定的“预编码关系”必需在L个中间码元内保持。小节B.5.2描述这些关系以及中间码元如何从源码元中产生。

[0236] 一旦中间码元已经产生，修复码元就产生且一个或多个修复码元作按照组被置入单个数据分组。每个修复码元组与编码码元ID(ESI)和G个编码码元相关联。ESI用来产生用于每个修复码元的三个整数的三元组(d，a，b)，再次使用小节B.5.4.4中所述的Trip[]发生器。这使用如小节B.5.4所述的发生器如小节B.3和B.4所述地完成。然后，使用小节B.5.4.3中所述的LTEnc[K，C[0]，……，C[L-1]，(d，a，b)]发生器，每个(d，a，b)三元组被用来从中间码元中产生相应的修复码元。

[0237] B.5.2.第一编码步骤：中间码元产生

[0238] B.5.2.1概述

[0239] 第一编码步骤是从源码元C'[0]，……，C'[K-1]中产生L个中间码元C[0]，……，C[L-1]的预编码步骤。中间码元由两套约束条件唯一限定：

[0240] 1.中间码元通过源码元三元组集与源码元相关。源码元三元组的产生使用小节B.5.4.4中所述的Trip[]发生器在小节B.5.2.2中定义。

[0241] 2.预编码关系集在各中间码元本身内保持。这些在小节B.5.2.3中定义。

[0242] 然后L个中间码元的产生在小节5.2.4中定义。

[0243] B.5.2.2源码元三元组

[0244] K个源码元的每一个都与三元组(d[i]，a[i]，b[i])(0≤i≤K)相关联。源码元三元组使用小节B.5.4.4中定义的三元组发生器确定为：

[0245] 对于每个i，0≤i<K

[0246] (d[i]，a[i]，b[i])＝Trip[K，i]

[0247] B.5.2.3预编码关系

[0248] L个中间码元之间的预编码关系通过根据前K个中间码元表达后面L-K个中间码元来定义。

[0249] 后面L-K个中间码元C[K],……,C[L-1]包括S个LDPC码元和H个半码元(Half码元)。S和H的值都如下所述根据K来确定。然后L=K+S+H。

[0250] 设

[0251] X 使X·(X-1)=2·K的最小正整数

[0252] S 使S≥ceil(0.01·K)+X的最小质数

[0253] H 使choose(H,ceil(H/2))≥K+S的最小整数

[0254] H’ =ceil(H/2))L=K+S+H

[0255] C[0],……,C[K-1]表示前K个中间码元

[0256] C[K],……,C[K+S-1]表示S个LDPC码元,初始化为零

[0257] C[K+S],……,C[L-1]表示H个半码元,初始化为零

[0258] S个LDPC码元被定义为以下过程结束时的C[K],……,C[K+S-1]值:

[0259] For i=0,……,K-1do

[0260] a=1+(floor(i/S)%(S-1))

[0261] b=i%S

[0262] C[K+b]=C[K+b]^C[i]

[0263] b=(b+a)%S

[0264] C[K+b]=C[K+b]^C[i]

[0265] b=(b+a)%S

[0266] C[K+b]=C[K+b]^C[i]

[0267] H个半码元被定义如下:

[0268] 设

[0269] 对于所有的正整数i,g[i]=i^(floor(i/2))

[0270] 注意:g[i]是Gray序列,其中每个元素与前一个元素有单个比特位置的不同

[0271] g[j,k]表示g[i]子序列的第j个元素,j=0,1,2,……,其元素在其二进制表示中有正好k个非零比特。

[0272] 然后,半码元被定义为以下过程后的C[K+S],……,C[L-1]值:

[0273] for h=0,……,H-1do

[0274] for j=0,……,K+S-1do

[0275] if g[j,H’]的比特h等于1,then C[h+K+S]=C[h+K+S]^C[j]。

[0276] B.5.2.4中间码元

[0277] B.5.2.4.1定义

[0278] 给定K个源码元C’[0],C’[1],……,C’[K-1],L个中间码元C[0],C[1],……,C[L-1]是满足以下条件的唯一定义的码元值:

[0279] 1.K个源码元C’[0],C’[1],……,C’[K-1]满足K个约束条件

[0280] C’[i]≡LTEnc[K,(C[0],……,C[L-1]),(d[i],a[i],b[i])],对于全部i,0≤i<K

[0281] 2.L个中间码元C[0],……,C[L-1]满足B.5.2.3中定义的预编码关系。

[0282] B.5.2.4.2计算中间码元

[0283] 本小节描述计算满足B.5.2.4.1中约束条件的L个中间码元C[0],……,C[L-1]的一种可能方法。

[0284] 用于从K个输入码元中产生N个输出码元的码的发生器矩阵G是GF(2)上的N x K矩阵,其中每行对应于输出码元之一而每列对应于输入码元之一,且第i个输出码元等于其列包含行i中非零条目的那些列的输入码元之和。

[0285] 然后,L个中间码元可如下计算:

[0286] 设

[0287] C表示L个中间码元C[0],……,C[L-1]的列向量

[0288] D表示包括S+H个零码元后跟K个源码元C'[0],C'[1]……,C'[K-1]的列向量

[0289] 则以上约束条件将GF(2)上的LxL矩阵A定义为:

[0290] $A \cdot C = D$

[0291] 矩阵A可构建如下:

[0292] 设:

[0293] G_{LDPC}是LDPC码元的S x K发生器矩阵。所以

[0294] $G_{LDPC} \cdot (C[0],\ldots\ldots,C[K-1])^T = (C[K],\ldots\ldots,C[K+S-1])^T$

[0295] G_{Half}是半码元的H x(K+S)发生器矩阵。所以

[0296] $G_{Half} \cdot (C[0],\ldots\ldots,C[K-1])^T = (C[K],\ldots\ldots,C[K+S+H-1])^T$

[0297] I_{SS}是SxS单位矩阵。

[0298] I_{HH}是HxH单位矩阵。

[0299] 0_{SSxxHH}是SxH零矩阵。

[0300] G_{LT}是由LT编码器产生的编码码元的KxL发生器矩阵。

[0301] 所以,

[0302] $G_{LT} \cdot (C[0],\ldots\ldots,C[L-1])^T = (C'[0],C'[1],\ldots\ldots,C'[K-1])^T$

[0303] 即,当且仅当C[i]包括在相异或产生LTEnc[K,(C[0],……,C[L-1]),(d[i],a[i],b[i])],$G_{Lti,j}=1$。

[0304] 则:

[0305] A的前S行等于$G_{LDPC}|I_S|Z_{SxH}$

[0306] A接下来的H行等于$G_{Half} \cdot |I_H$

[0307] A剩下的K行等于G_{LT}。

[0308] 矩阵A如图20所示。然后中间码元可计算为:

[0309] $C = A^{-1} \cdot D$

[0310] 源三元组被生成为:对于任何K,矩阵A都满秩,因此可逆。该计算可通过将MSCR解码过程应用于K个源码元C'[0],C'[1],……,C'[K-1]来实现,以产生L个中间码元C[0],……,C[L-1]。

[0311] 为了有效地从源码元产生中间码元,建议使用诸如小节B.6所述的有效解码器实现。源码元三元组被设计成使用该算法来便于源码元的有效解码。

[0312] B.5.3.第二个编码步骤:LT编码

[0313] 在第二个编码步骤中,使用根据小节B.3.2.2和B.4.2产生的三元组(d,a,b)=

Trip[K,X]，具有ESI X的修复码元通过将在小节B.5.4中定义的发生器LTEnc[K,(C[0],……,C[L-1]),(d,a,b)]应用于L个中间码元C[0],……,C[L-1]来产生。

[0314] B.5.4发生器

[0315] B.5.4.1随机发生器

[0316] 随机数发生器Rand[X,i,m]定义如下，其中X是非负整数，i是非负整数，且m是正整数，所产生的值是0到m-1之间的整数。设V_0和V_1都是256个条目的数组，其中每个条目都是4-字节的无符号整数。这些数组在小节B.7中提供。

[0317] 然后，Rand[X,i,m]＝(V_0[X+i]％256)^V_1[(floor(X/256)+i)％256]％m

[0318] B.5.4.2次数发生器

[0319] 次数发生器Deg[v]定义如下，其中v是至少为0并小于2^{20}＝1048576的整数。

[0320] 在图21中，寻找索引j使f[j-1]≤v<f[j]

[0321] Deg[v]＝d[j]

[0322] B.5.4.3LT编码码元发生器

[0323] 编码码元发生器LTEnc[K,(C[0],……,C[L-1]),(d,a,b)]取以下输入：

[0324] K是源块(子块)的源码元(或子码元)的数量。设L根据小节B.5.2所述的K导出，并设L'为大于或等于L的最小质数。

[0325] (C[0],……,C[L-1])是如小节B.5.2所述地产生的L个中间码元(子码元)的数组。(d,a,b)是使用小节B.5.3.4中定义的三元组发生器确定的源三元组，而d是表示编码码元次数的整数，a是1到L'-1(包括1和L'-1)之间的整数，b是0到L'-1(包括1和L'-1)之间的整数。

[0326] 编码码元发生器根据以下算法，产生单个编码码元作为输出：

[0327] while(b≥L)do b＝(b+a)％L'

[0328] LTEnc[K,(C[0],……,C[L-1]),(d,a,b)]＝C[b]

[0329] For j＝1,……,min(d-1,L-1)do

[0330] b＝(b+a)％L'

[0331] while(b≥L)do b＝(b+a)％L'

[0332] LTEnc[K,(C[0],……,C[L-1]),(d,a,b)]＝LTEnc[K,(C[0],……,C[L-1]),(d,a,b)]^C[b]

[0333] B.5.4.4三元组发生器

[0334] 三元组发生器Trip[K,X]取以下输入：

[0335] K源码元的数量

[0336] X编码码元ID

[0337] 设

[0338] L根据如小节B.5.2中所述的K确定

[0339] L'是大于或等于L的最小质数

[0340] Q＝65521，小于2^{16}的最大质数

[0341] J(K)是与K相关联的系统索引，如附录A中所定义。

[0342] 三元组发生器的输出是如下确定的三元组(d,a,b)：

[0343] 1.A＝(53591+J(K)•997)％Q

[0344] 2.B＝10267 • (J(K)+1)％Q

[0345] 3.Y＝(B+X • A)％Q

[0346] 4.V＝Rand[Y,0,2^{20}]

[0347] 5.d＝Deg[v]

[0348] 6.a＝1+Rand[Y,1,L'-1]

[0349] 7.b＝Rand[Y,2,L']

[0350] B.6 FEC解码器实现

[0351] B.6.1概论

[0352] 本节描述用于本说明书所述的MSCR码的有效解码算法。注意：每个所接收的编码码元可视为中间码元在其中的方程的值。根据这些联立方程，以及中间码元之间的已知预编码关系，任何用于解联立方程的算法可成功地解码中间码元和源码元。然而，所选择的算法对解码的计算效率有主要影响。

[0353] B.6.2解码源块

[0354] B.6.2.1概论

[0355] 假设解码器知道它要解码的源块的结构，包括码元大小T、以及源块中的码元数量K。

[0356] 根据小节B.5中所述的算法，MSCR解码器可计算预编码码元的总量达L＝K+S+H，并确定它们如何从要被解码的源块中产生。在本说明书中，假设接收到的要被解码的源块的编码码元被传递给了解码器。此外，对于每个这种编码码元，假设其异或等于编码码元的中间码元的数量和集被传递给了解码器。在源码元的情形中，小节B.5.2.2中所述的源码元三元组表示其和给出每个源码元的中间码元的数量和集。

[0357] 设N≥K为所接收的源块的编码码元的数量，并设M＝S+H+N。随后MxL比特矩阵A可从传递给解码器的要解码的源块的信息中导出。设C是L个中间码元的列向量，并设D是M个其值为接收器已知的码元的列向量，其中M个码元的前S+H个是与LDPC和半码元相对应的零值码元(这些是LDPC和半码元的校验码元，而不是LDPC和半码元本身)，并且M个码元的剩余N个是接收到的源块的编码码元。然后，A是满足A • C＝D的比特矩阵，其中 • 表示GF[2]上的矩阵乘法。特别地，如果与索引j相对应的中间码元异或成对应于编码中索引i的LDPC、半或编码码元，或者如果索引i对应于LDPC或半码元而索引j对应于同一LDPC或半码元，则A[i,j]＝1。对于其它的i和j，则A[i,j]＝0。

[0358] 解码源块等效于根据已知的A和D来解码C。显然当且仅当GF[2]上A的秩为L时C可被解码。一旦C已被解码，则遗漏的源码元就可通过使用源码元三元组来获取，以确定要异或以获得每个遗漏的源码元的中间码元的数量和集。

[0359] 解码C的第一个步骤是形成解码时间表。在该步骤中，使用高斯消元算法(使用行运算以及行和列的重新排序)并去掉M-L行之后，A被转换成LxL单元矩阵。解码时间表包括在高斯消元过程期间的行运算的序列、以及行和列的重新排序，并仅依赖于A而非D。根据D对C的解码可与形成解码时间表并发地进行，或者解码可基于解码时间表在之后进行。

[0360] 解码时间表和C的解码之间对应如下。开始时设c[0]＝0,c[1]＝1,……,c[L-1]＝L-1,并设d[0]＝0,d[1]＝1,……,d[M-1]＝M-1。

[0361] -每当A的行i被异或成解码时间表中的行i'时，在解码过程中码元D[d[i]]被异或

成码元D[d[i']]。

[0362] -每当行i与解码时间表中的行i'交换时，在解码过程中d[i]的值与d[i']的值交换。

[0363] -每当列j与解码时间表中的列j'交换时，在解码过程中c[j]的值与c[j']的值交换。

[0364] 根据该对应关系，显然解码源块时码元异或的总数量是高斯消元中行运算的数量(未作交换)。因为A是高斯消元之后并去掉最后M-L行之后的LxL单位矩阵，所以显然在成功解码结束时，L个码元D(d[0])，D(d[1])，……，D(d[L-1])是L个码元C(c[0])，C(c[1])，……，C(c[L-1])的值。

[0365] 执行高斯消元以形成解码时间表的顺序对解码是否成功没有关系。然而，解码速度较强地取决于执行高斯消元的顺序(此外，保持A的稀疏表示是重要的，尽管并未在此描述)。本节的剩下部分描述相对高效的执行高斯消元的顺序。

[0366] B.6.2.2第一阶段

[0367] 高斯消元的第一阶段：矩阵A在概念上被分成多个子矩阵。子矩阵的大小非负整数i和u为参数，初始化为0。A的子矩阵为：

[0368] (1)由前i行和前i列的交点限定的子矩阵I。这是该阶段中每个步骤结束时的单位矩阵。

[0369] (2)由前i行与除前i列和最后u列之外的所有列的交点限定的子矩阵。该子矩阵的所有条目都为零。

[0370] (3)由前i列与除前i行之外的所有行的交点限定的子矩阵。该子矩阵的所有条目都为零。

[0371] (4)由所有行和最后u列的交点限定的子矩阵U。

[0372] (5)由除前i列和最后u列之外的全部列与除前i行之外的所有行的交点形成的子矩阵V。

[0373] 图22示出A的子矩阵。在第一阶段开始时，V＝A。在每个步骤中，选择A的一行。由V的结构限定的跟踪图用于确定选择A的哪一行。与V相交的各个列为图中的节点，并且V中具有正好2个1的行是在该两个1的位置上连接两列(节点)的图的边。该图中的一个组成是节点(列)和边(行)的最大集，从而图中每对节点/边之间都有路径。组成的大小是该组成中节点(列)的数量。

[0374] 在第一阶段中最多有L个步骤。当i+u＝L时，即当V和V上所有零矩阵消失且A包括I、I下的所有零子矩阵和U时，该阶段成功结束。如果在V消失前的某一步骤上V中有非零行待选，则该阶段不成功地结束，解码失败。在每个步骤中，如下选择行A：

[0375] 如果V的所有条目都为零，则不选择行，且解码失败。

[0376] 设r为使A的至少一行在V中恰好有r个1的最小整数。

[0377] 如果r≠2，则选择V中正好有r个1的行，该行是所有这些行中具有最小原始次数的行。

[0378] 如果r＝2，则选择V中正好有2个1的任一行，它是由X限定的图中最大尺寸组成的一部分。

[0379] 在本步骤中选择该行后，与V相交的A的第一行与选定行交换，使该选定行是与V相

交的第一行。A中与V相交的各列重新排序，使得选定行中r个1之一出现在V的第一列，并使剩下的r-1个1出现在V后面的列中。然后，选定行被异或成V的第一列中有1的选定行之下A的所有其它行。最后，i增1而u增r-1，从而完成该步骤。

[0380] B.6.2.3第二个步骤

[0381] 子矩阵U进一步被分成前i行U_{upper}，以及剩下的M-i行U_{lower}。在第二阶段对U_{lower}执行高斯消元，以确定其秩小于u(解码失败)、或将其转换成前u行为单位矩阵的矩阵(第二阶段成功)。称该u x u单位矩阵为I_u。与$U_{lower}-I_u$相交的A的M-L行被去除。该阶段之后A具有L行和L列。

[0382] B.6.2.4第三阶段

[0383] 在第二阶段之后，需要清零去除以将A转换成LxL单位矩阵的A的那部分是U_{upper}。子矩阵U_{upper}的行数i通常比U_{upper}的列数u大得多。为了有效地对U_{upper}清零去除，随后在第三阶段基于I_u计算预算矩阵U'，并在第四阶段将U'用于清零去除U_{upper}。I_u的u个行被分成ceil(u/8)个每个为8行的组。然后，对于每个8行的组，计算该8个行的所有非零组合，结果为$2^8-1=255$行(这可通过每组各行的$2^8-8-1=247$次异或来完成，因为I_u中出现的汉明权1的组合不需要重新计算)。因而，最后的预算矩阵U'具有ceil(u/8)·255行和u列。注意，U'形式上并非是矩阵A的一部分，但将用于第四阶段以清零去除U_{upper}。

[0384] B.6.2.5第四阶段

[0385] 对于A的前I行，对该行的U_{upper}子矩阵中每个8列组，如果U_{upper}中8列条目的集并非全为零，则与8列中模式相匹配的预算矩阵U'的行被异或到该行中，从而以将U'的一行异或到该行的代价清零去除该行中的那8列。

[0386] 在该阶段之后A为LxL的单位矩阵，并已成功形成完整的解码时间表。然后，包括异或已知编码码元的相应解码可被执行，以基于解码时间表恢复中间码元。

[0387] 与全部源码元相关联的三元组根据B.5.2.2计算。接收到的源码元的三元组在解码中使用。遗漏源码元的三元组用来确定哪些中间码元需要异或，以恢复所遗漏的源码元。

[0388] 一些多级码的部分属性

[0389] 在上述大多数示例中，输入和输出码元对相同数量的比特编码，且每个输出码元都被置入一分组中(分组是全部接收或全部丢失的传送单位)。在一些实施例中，通信系统被更改使每个分组都包含若干输出码元。输出码元值的大小基于众多因素，被设置成由在开始将文件或流块分成输入码元时输入码元值的大小确定的大小。解码过程保持基本不变，除非接收每个分组时输出码元成串到达。

[0390] 输入码元和输出码元大小的设置通常由文件或流块的大小、以及输出码元要在其上传送的通信系统命令。例如，如果通信系统将数据比特分成限定大小的分组、或用其它方法来分组比特，则码元大小的设计从分组或组的大小开始。然后，设计者将确定在一个分组或组中携带多少输出码元，并确定输出码元大小。为了简便，设计者可能将输入码元大小设置成等于输出码元大小，但如果输入数据使用不同的输入码元大小会更方便，则可使用之。

[0391] 上述编码过程基于原始文件或流块产生包含输出码元的分组流。流中每个输出码元独立于所有其它输出码元产生，且对可创建的输出码元数量没有上限和下限。密钥与每个输出码元相关联。该密钥，以及输入文件或流块的部分内容确定输出码元的值。连续产生的输出码元不需要具有连续密钥，且在某些应用中，最好随机产生密钥序列，或伪随机地产

生该序列。

[0392] 多级解码具有一属性，即，如果原始文件或流块可分成K个大小相等的输入码元，且每个输出码元值都具有与输入码元值相同的长度，则该文件或块平均地可从K+A个输出码元中恢复，其中A比K小的概率极高。例如，对于以上介绍的权重分布，如果K大于19,681，则A值超过α*K的概率最高为10^{-12}；对于任何K值，概率最高为10^{-10}。因为特定输出码元以随机顺序或伪随机顺序产生，且在传输中特定输出码元的丢失假设为随机，则恢复输入文件或块所需的输出码元的施加数量存在一些较小变化。在一些情形中，当K+A个分组的特定集合不足以解码逐个输入文件或块时，如果接收器能从输入分组的一个或多个源中收集更多分组时，该输入文件或块仍然是可恢复的。

[0393] 因为输出码元的数量仅受I的解的限制，所以可产生多于K+A个的输出码元。例如，如果I是32比特数，则可产生40亿个不同的输出码元，而文件或流块可包括K＝50,000个输入码元。在一些应用中，只可产生并传送40亿个输出码元的一小部分，并且用极少量的可能输出码元可恢复输入文件或流块是几乎肯定的，用略多于K个输出码元就可恢复输入文件或块也是相当可能的(假设输入码元大小与输出码元大小相同)。

[0394] 在一些应用中，能解码全部输入码元或不能解码全部输入码元都是可以接受的，只是概率相对较低。在这种应用中，接收器可在接收K+A个输出码元之后停止解码全部输入码元。或者，接收器在接收少于K+A个输出码元之后停止接收输出码元。在一些应用中，接收器甚至只接收K个或更少输出码元。因而，可以理解，在本发明的一些实施例中，所需准确度不需要完全恢复所有输入码元。

[0395] 此外，在一些可接受不完全恢复的应用中，数据可编码成不能恢复所有输入码元，或者完全恢复输入码元将需要接收比输入码元数量多得多的输出码元。这种编码通常需要较少的计算成本，因而是降低编码计算成本的可接受方法。

[0396] 可以理解，上述各图中的各个功能块可通过硬件和/或软件的组合实现，并且在特定实现中，一些块的部分或全部功能可加以组合。类似地，也可以理解，在此所述的各种方法可通过硬件和/或软件的组合实现。

[0397] 以上描述是说明性而非限制性的。对本领域技术人员而言，在阅读本说明书之后本发明的许多变体变得显而易见。因此，本发明的范围不应参照以上描述确定，相反应参照所附权利要求以及等效方案的全部范围来确定。

[0398] 附录A.系统索引J(K)的值

[0399] 对于每个值K，系统索引J(K)被设计成具有以下属性：源码元三元组(d[0]，a[0]，b[0])，……，(d[L-1]，a[L-1]，b[L-1])的集使L个中间码元唯一限定，即小节B.5.2.4.2中的矩阵A具有满秩并因此可逆。下面是介于4和8192(包括4和8192)的K值的系统索引的列表。各值的顺序是读取顺序，即从第一行的第一个数到第一行的最后一个数，然后是第二行的第一个数，依此类推。

[0400] 18,14,61,46,14,22,20,40,48,1,29,40,43,46,18,8,20,2,61,26,13,29,36,19,58,5,58,0,54,56,24,14,5,67,39,31,25,29,24,19,14,56,49,49,63,30,4,39,2,1,20,19,61,4,54,70,25,52,9,26,55,69,27,68,75,19,64,57,45,3,37,31,100,41,25,41,53,23,9,31,26,30,30,46,90,50,13,90,77,61,31,54,54,3,21,66,21,11,23,11,29,21,7,1,27,4,34,17,85,69,17,75,93,57,0,53,71,88,119,88,90,22,0,58,41,22,96,26,79,118,

19,3,81,72,50,0,32,79,28,25,12,25,29,3,37,30,30,41,84,32,31,61,32,61,7,56,54,
39,33,66,29,3,14,75,75,78,84,75,84,25,54,25,25,107,78,27,73,0,49,96,53,50,21,
10,73,58,65,27,3,27,18,54,45,69,29,3,65,31,71,76,56,54,76,54,13,5,18,142,17,
3,37,114,41,25,56,0,23,3,41,22,22,31,18,48,31,58,37,75,88,3,56,1,95,19,73,52,
52,4,75,26,1,25,10,1,70,31,31,12,10,54,46,11,74,84,74,8,58,23,74,8,36,11,16,
94,76,14,57,65,8,22,10,36,36,96,62,103,6,75,103,58,10,15,41,75,125,58,15,10,
34,29,34,4,16,29,18,18,28,71,28,43,77,18,41,41,41,62,29,96,15,106,43,15,3,43,
61,3,18,103,77,29,103,19,58,84,58,1,146,32,3,70,52,54,29,70,69,124,62,1,26,
38,26,3,16,26,5,51,120,41,16,1,43,34,34,29,37,56,29,96,86,54,25,84,50,34,34,
93,84,96,29,29,50,50,6,1,105,78,15,37,19,50,71,36,6,54,8,28,54,75,75,16,75,
131,5,25,16,69,17,69,6,96,53,96,41,119,6,6,88,50,88,52,37,0,124,73,73,7,14,
36,69,79,6,114,40,79,17,77,24,44,37,69,27,37,29,33,37,50,31,69,29,101,7,61,
45,17,73,37,34,18,94,22,22,63,3,25,25,17,3,90,34,34,41,34,41,54,41,54,41,41,
41,163,143,96,18,32,39,86,104,11,17,17,11,86,104,78,70,52,78,17,73,91,62,7,
128,50,124,18,101,46,10,75,104,73,58,132,34,13,4,95,88,33,76,74,54,62,113,
114,103,32,103,69,54,53,3,11,72,31,53,102,37,53,11,81,41,10,164,10,41,31,36,
113,82,3,125,62,16,4,41,41,4,128,49,138,128,74,103,0,6,101,41,142,171,39,105,
121,81,62,41,81,37,3,81,69,62,3,69,70,21,29,4,91,87,37,79,36,21,71,37,41,75,
128,128,15,25,3,108,73,91,62,114,62,62,36,36,15,58,114,61,114,58,105,114,41,
61,176,145,46,37,30,220,77,138,15,1,128,53,50,50,58,8,91,114,105,63,91,37,37,
13,169,51,102,6,102,23,105,23,58,6,29,29,19,82,29,13,36,27,29,61,12,18,127,
127,12,44,102,18,4,15,206,53,127,53,17,69,69,69,29,29,109,25,102,25,53,62,99,
62,62,29,62,62,45,91,125,29,29,29,4,117,72,4,30,71,71,95,79,179,71,30,53,32,
32,49,25,91,25,26,26,103,123,26,41,162,78,52,103,25,6,142,94,45,45,94,127,94,
94,94,47,209,138,39,39,19,154,73,67,91,27,91,84,4,84,91,12,14,165,142,54,69,
192,157,185,8,95,25,62,103,103,95,71,97,62,128,0,29,51,16,94,16,16,51,0,29,
85,10,105,16,29,29,13,29,4,4,132,23,95,25,54,41,29,50,70,58,142,72,70,15,72,
54,29,22,145,29,127,29,85,58,101,34,165,91,46,46,25,185,25,77,128,46,128,46,
188,114,46,25,45,45,114,145,114,15,102,142,8,73,31,139,157,13,79,13,114,150,
8,90,91,123,69,82,132,8,18,10,102,103,114,103,8,103,13,115,55,62,3,8,154,114,
99,19,8,31,73,19,99,10,6,121,32,13,32,119,32,29,145,30,13,13,114,145,32,1,
123,39,29,31,69,31,140,72,72,25,25,123,25,123,8,4,85,8,25,39,25,39,85,138,25,
138,25,33,102,70,25,25,31,25,25,192,69,69,114,145,120,120,8,33,98,15,212,155,
8,101,8,8,98,68,155,102,132,120,30,25,123,123,101,25,123,32,24,94,145,32,24,
94,118,145,101,53,53,25,128,173,142,81,81,69,33,33,125,4,1,17,27,4,17,102,27,
13,25,128,71,13,39,53,13,53,47,39,23,128,53,39,47,39,135,158,136,36,36,27,
157,47,76,213,47,156,25,25,53,25,53,25,86,27,159,25,62,79,39,79,25,145,49,25,
143,13,114,150,130,94,102,39,4,39,61,77,228,22,25,47,119,205,122,119,205,119,
22,119,258,143,22,81,179,22,22,143,25,65,53,168,36,79,175,37,79,70,79,103,70,

25,175,4,96,96,49,128,138,96,22,62,47,95,105,95,62,95,62,142,103,69,103,30,103,34,173,127,70,127,132,18,85,22,71,18,206,206,18,128,145,70,193,188,8,125,114,70,128,114,145,102,25,12,108,102,94,10,102,1,102,124,22,22,118,132,22,116,75,41,63,41,189,208,55,85,69,8,71,53,71,69,102,165,41,99,69,33,33,29,156,102,13,251,102,25,13,109,102,164,102,164,102,25,29,228,29,259,179,222,95,94,30,30,30,142,55,142,72,55,102,128,17,69,164,165,3,164,36,165,27,27,45,21,21,237,113,83,231,106,13,154,13,154,128,154,148,258,25,154,128,3,27,10,145,145,21,146,25,1,185,121,0,1,95,55,95,95,30,0,27,95,0,95,8,222,27,121,30,95,121,0,98,94,131,55,95,95,30,98,30,0,91,145,66,179,66,58,175,29,0,31,173,146,160,39,53,28,123,199,123,175,146,156,54,54,149,25,70,178,128,25,70,70,94,224,54,4,54,54,25,228,160,206,165,143,206,108,220,234,160,13,169,103,103,103,91,213,222,91,103,91,103,31,30,123,13,62,103,50,106,42,13,145,114,220,65,8,8,175,11,104,94,118,132,27,118,193,27,128,127,127,183,33,30,29,103,128,61,234,165,41,29,193,33,207,41,165,165,55,81,157,157,8,81,11,27,8,8,98,96,142,145,41,179,112,62,180,206,206,165,39,241,45,151,26,197,102,192,125,128,67,128,69,128,197,33,125,102,13,103,25,30,12,30,12,30,25,77,12,25,180,27,10,69,235,228,343,118,69,41,8,69,175,25,69,25,125,41,25,41,8,155,146,155,146,155,206,168,128,157,27,273,211,211,168,11,173,154,77,173,77,102,102,102,8,85,95,102,157,28,122,234,122,157,235,222,241,10,91,179,25,13,25,41,25,206,41,6,41,158,206,206,33,296,296,33,228,69,8,114,148,33,29,66,27,27,30,233,54,173,108,106,108,108,53,103,33,33,33,176,27,27,205,164,105,237,41,27,72,165,29,29,259,132,132,132,364,71,71,27,94,160,127,51,234,55,27,95,94,165,55,55,41,0,41,128,4,123,173,6,164,157,121,121,154,86,164,164,25,93,164,25,164,210,284,62,93,30,25,25,30,30,260,130,25,125,57,53,166,166,166,185,166,158,94,113,215,159,62,99,21,172,99,184,62,259,4,21,21,77,62,173,41,146,6,41,128,121,41,11,121,103,159,164,175,206,91,103,164,72,25,129,72,206,129,33,103,102,102,29,13,11,251,234,135,31,8,123,65,91,121,129,65,243,10,91,8,65,70,228,220,243,91,10,10,30,178,91,178,33,21,25,235,165,11,161,158,27,27,30,128,75,36,30,36,36,173,25,33,178,112,162,112,112,112,162,33,33,178,123,123,39,106,91,106,106,158,106,106,284,39,230,21,228,11,21,228,159,241,62,10,62,10,68,234,39,39,138,62,22,27,183,22,215,10,175,175,353,228,42,193,175,175,27,98,27,193,150,27,173,17,233,233,25,102,123,152,242,108,4,94,176,13,41,219,17,151,22,103,103,53,128,233,284,25,265,128,39,39,138,42,39,21,86,95,127,29,91,46,103,103,215,25,123,123,230,25,193,180,30,60,30,242,136,180,193,30,206,180,60,165,206,193,165,123,164,103,68,25,70,91,25,82,53,82,186,53,82,53,25,30,282,91,13,234,160,160,126,149,36,36,160,149,178,160,39,294,149,149,160,39,95,221,186,106,178,316,267,53,53,164,159,164,165,94,228,53,52,178,183,53,294,128,55,140,294,25,95,366,15,304,13,183,77,230,6,136,235,121,311,273,36,158,235,230,98,201,165,165,165,91,175,248,39,185,128,39,39,128,313,91,36,219,130,25,130,234,234,130,234,121,205,304,94,77,

64,259,60,60,60,77,242,60,145,95,270,18,91,199,159,91,235,58,249,26,123,114,29,15,191,15,30,55,55,347,4,29,15,4,341,93,7,30,23,7,121,266,178,261,70,169,25,25,158,169,25,169,270,270,13,128,327,103,55,128,103,136,159,103,327,41,32,111,111,114,173,215,173,25,173,180,114,173,173,98,93,25,160,157,159,160,159,159,160,320,35,193,221,33,36,136,248,91,215,125,215,156,68,125,125,1,287,123,94,30,184,13,30,94,123,206,12,206,289,128,122,184,128,289,178,29,26,206,178,65,206,128,192,102,197,36,94,94,155,10,36,121,280,121,368,192,121,121,179,121,36,54,192,121,192,197,118,123,224,118,10,192,10,91,269,91,49,206,184,185,62,8,49,289,30,5,55,30,42,39,220,298,42,347,42,234,42,70,42,55,321,129,172,173,172,13,98,129,325,235,284,362,129,233,345,175,261,175,60,261,58,289,99,99,99,206,99,36,175,29,25,432,125,264,168,173,69,158,273,179,164,69,158,69,8,95,192,30,164,101,44,53,273,335,273,53,45,128,45,234,123,105,103,103,224,36,90,211,282,264,91,228,91,166,264,228,398,50,101,91,264,73,36,25,73,50,50,242,36,36,58,165,204,353,165,125,320,128,298,298,180,128,60,102,30,30,53,179,234,325,234,175,21,250,215,103,21,21,250,91,211,91,313,301,323,215,228,160,29,29,81,53,180,146,248,66,159,39,98,323,98,36,95,218,234,39,82,82,230,62,13,62,230,13,30,98,0,8,98,8,98,91,267,121,197,30,78,27,78,102,27,298,160,103,264,264,264,175,17,273,273,165,31,160,17,99,17,99,234,31,17,99,36,26,128,29,214,353,264,102,36,102,264,264,273,273,4,16,138,138,264,128,313,25,420,60,10,280,264,60,60,103,178,125,178,29,327,29,36,30,36,4,52,183,183,173,52,31,173,31,158,31,158,31,9,31,31,353,31,353,173,415,9,17,222,31,103,31,165,27,31,31,165,27,27,206,31,31,4,4,30,4,4,264,185,159,310,273,310,173,40,4,173,4,173,4,250,250,62,188,119,250,233,62,121,105,105,54,103,111,291,236,236,103,297,36,26,316,69,183,158,206,129,160,129,184,55,179,279,11,179,347,160,184,129,179,351,179,353,179,129,129,351,11,111,93,93,235,103,173,53,93,50,111,86,123,94,36,183,60,55,55,178,219,253,321,178,235,235,183,183,204,321,219,160,193,335,121,70,69,295,159,297,231,121,231,136,353,136,121,279,215,366,215,353,159,353,353,103,31,31,298,298,30,30,165,273,25,219,35,165,259,54,36,54,54,165,71,250,327,13,289,165,196,165,165,94,233,165,94,60,165,96,220,166,271,158,397,122,53,53,137,280,272,62,30,30,30,105,102,67,140,8,67,21,270,298,69,173,298,91,179,327,86,179,88,179,179,55,123,220,233,94,94,175,13,53,13,154,191,74,83,83,325,207,83,74,83,325,74,316,388,55,55,364,55,183,434,273,273,273,164,213,11,213,327,321,21,352,185,103,13,13,55,30,323,123,178,435,178,30,175,175,30,481,527,175,125,232,306,232,206,306,364,206,270,206,232,10,30,130,160,130,347,240,30,136,130,347,136,279,298,206,30,103,273,241,70,206,306,434,206,94,94,156,161,321,321,64,161,13,183,183,83,161,13,169,13,159,36,173,159,36,36,230,235,235,159,159,335,312,42,342,264,39,39,39,34,298,36,36,252,164,29,493,29,387,387,435,493,132,273,105,132,74,73,206,234,273,206,95,15,280,280,280,280,397,273,273,242,397,280,397,397,397,273,397,280,230,137,353,67,81,137,137,

353,259,312,114,164,164,25,77,21,77,165,30,30,231,234,121,234,312,121,364,136,123,123,136,123,136,150,264,285,30,166,93,30,39,224,136,39,355,355,397,67,67,25,67,25,298,11,67,264,374,99,150,321,67,70,67,295,150,29,321,150,70,29,142,355,311,173,13,253,103,114,114,70,192,22,128,128,183,184,70,77,215,102,292,30,123,279,292,142,33,215,102,468,123,468,473,30,292,215,30,213,443,473,215,234,279,279,279,279,265,443,206,66,313,34,30,206,30,51,15,206,41,434,41,398,67,30,301,67,36,3,285,437,136,136,22,136,145,365,323,323,145,136,22,453,99,323,353,9,258,323,231,128,231,382,150,420,39,94,29,29,353,22,22,347,353,39,29,22,183,8,284,355,388,284,60,64,99,60,64,150,95,150,364,150,95,150,6,236,383,544,81,206,388,206,58,159,99,231,228,363,363,121,99,121,121,99,422,544,273,173,121,427,102,121,235,284,179,25,197,25,179,511,70,368,70,25,388,123,368,159,213,410,159,236,127,159,21,373,184,424,327,250,176,176,175,284,316,176,284,327,111,250,284,175,175,264,111,176,219,111,427,427,176,284,427,353,428,55,184,493,158,136,99,287,264,334,264,213,213,292,481,93,264,292,295,295,6,367,279,173,308,285,158,308,335,299,137,137,572,41,137,137,41,94,335,220,36,224,420,36,265,265,91,91,71,123,264,91,91,123,107,30,22,292,35,241,356,298,14,298,441,35,121,71,63,130,63,488,363,71,63,307,194,71,71,220,121,125,71,220,71,71,71,71,235,265,353,128,155,128,420,400,130,173,183,183,184,130,173,183,13,183,130,130,183,183,353,353,183,242,183,183,306,324,324,321,306,321,6,6,128,306,242,242,306,183,183,6,183,321,486,183,164,30,78,138,158,138,34,206,362,55,70,67,21,375,136,298,81,298,298,298,230,121,30,230,311,240,311,311,158,204,136,136,184,136,264,311,311,312,312,72,311,175,264,91,175,264,121,461,312,312,238,475,350,512,350,312,313,350,312,366,294,30,253,253,253,388,158,388,22,388,22,388,103,321,321,253,7,437,103,114,242,114,114,242,114,114,242,242,242,306,242,114,7,353,335,27,241,299,312,364,506,409,94,462,230,462,243,230,175,175,462,461,230,428,426,175,175,165,175,175,372,183,572,102,85,102,538,206,376,85,85,284,85,85,284,398,83,160,265,308,398,310,583,289,279,273,285,490,490,211,292,292,158,398,30,220,169,368,368,368,169,159,368,93,368,368,93,169,368,368,443,368,298,443,368,298,538,345,345,311,178,54,311,215,178,175,222,264,475,264,264,475,478,289,63,236,63,299,231,296,397,299,158,36,164,164,21,492,21,164,21,164,403,26,26,588,179,234,169,465,295,67,41,353,295,538,161,185,306,323,68,420,323,82,241,241,36,53,493,301,292,241,250,63,63,103,442,353,185,353,321,353,185,353,353,185,409,353,589,34,271,271,34,86,34,34,353,353,39,414,4,95,95,4,225,95,4,121,30,552,136,159,159,514,159,159,54,514,206,136,206,159,74,235,235,312,54,312,42,156,422,629,54,465,265,165,250,35,165,175,659,175,175,8,8,8,8,206,206,206,50,435,206,432,230,230,234,230,94,299,299,285,184,41,93,299,299,285,41,285,158,285,206,299,41,36,396,364,364,120,396,514,91,382,538,807,717,22,93,412,54,215,54,298,308,148,298,148,298,308,102,656,6,148,745,128,298,64,407,273,41,172,64,234,250,398,

181,445,95,236,441,477,504,102,196,137,364,60,453,137,364,367,334,364,299,196,397,630,589,589,196,646,337,235,128,128,343,289,235,324,427,324,58,215,215,461,425,461,387,440,285,440,440,285,387,632,325,325,440,461,425,425,387,627,191,285,440,308,55,219,280,308,265,538,183,121,30,236,206,30,455,236,30,30,705,83,228,280,468,132,8,132,132,128,409,173,353,132,409,35,128,450,137,398,67,432,423,235,235,388,306,93,93,452,300,190,13,452,388,30,452,13,30,13,30,306,362,234,721,635,809,784,67,498,498,67,353,635,67,183,159,445,285,183,53,183,445,265,432,57,420,432,420,477,327,55,60,105,183,218,104,104,475,239,582,151,239,104,732,41,26,784,86,300,215,36,64,86,86,675,294,64,86,528,550,493,565,298,230,312,295,538,298,295,230,54,374,516,441,54,54,323,401,401,382,159,837,159,54,401,592,159,401,417,610,264,150,323,452,185,323,323,185,403,185,423,165,425,219,407,270,231,99,93,231,631,756,71,364,434,213,86,102,434,102,86,23,71,335,164,323,409,381,4,124,41,424,206,41,124,41,41,703,635,124,493,41,41,487,492,124,175,124,261,600,488,261,488,261,206,677,261,308,723,908,704,691,723,488,488,441,136,476,312,136,550,572,728,550,22,312,312,22,55,413,183,280,593,191,36,36,427,36,695,592,19,544,13,468,13,544,72,437,321,266,461,266,441,230,409,93,521,521,345,235,22,142,150,102,569,235,264,91,521,264,7,102,7,498,521,235,537,235,6,241,420,420,631,41,527,103,67,337,62,264,527,131,67,174,263,264,36,36,263,581,253,465,160,286,91,160,55,4,4,631,631,608,365,465,294,427,427,335,669,669,129,93,93,93,93,74,66,758,504,347,130,505,504,143,505,550,222,13,352,529,291,538,50,68,269,130,295,130,511,295,295,130,486,132,61,206,185,368,669,22,175,492,207,373,452,432,327,89,550,496,611,527,89,527,496,550,516,516,91,136,538,264,264,124,264,264,264,264,264,535,264,150,285,398,285,582,398,475,81,694,694,64,81,694,234,607,723,513,234,64,581,64,124,64,607,234,723,717,367,64,513,607,488,183,488,450,183,550,286,183,363,286,414,67,449,449,366,215,235,95,295,295,41,335,21,445,225,21,295,372,749,461,53,481,397,427,427,427,714,481,714,427,717,165,245,486,415,245,415,486,274,415,441,456,300,548,300,422,422,757,11,74,430,430,136,409,430,749,191,819,592,136,364,465,231,231,918,160,589,160,160,465,465,231,157,538,538,259,538,326,22,22,22,179,22,22,550,179,287,287,417,327,498,498,287,488,327,538,488,583,488,287,335,287,335,287,41,287,335,287,327,441,335,287,488,538,327,498,8,8,374,8,64,427,8,374,417,760,409,373,160,423,206,160,106,499,160,271,235,160,590,353,695,478,619,590,353,13,63,189,420,605,427,643,121,280,415,121,415,595,417,121,398,55,330,463,463,123,353,330,582,309,582,582,405,330,550,405,582,353,309,308,60,353,7,60,71,353,189,183,183,183,582,755,189,437,287,189,183,668,481,384,384,481,481,481,477,582,582,499,650,481,121,461,231,36,235,36,413,235,209,36,689,114,353,353,235,592,36,353,413,209,70,308,70,699,308,70,213,292,86,689,465,55,508,128,452,29,41,681,573,352,21,21,648,648,69,509,409,21,264,21,509,514,514,409,21,264,443,443,427,160,433,663,433,231,

646,185,482,646,433,13,398,172,234,42,491,172,234,234,832,775,172,196,335,822,461,298,461,364,1120,537,169,169,364,694,219,612,231,740,42,235,321,279,960,279,353,492,159,572,321,159,287,353,287,287,206,206,321,287,159,321,492,159,55,572,600,270,492,784,173,91,91,443,443,582,261,497,572,91,555,352,206,261,555,285,91,555,497,83,91,619,353,488,112,4,592,295,295,488,235,231,769,568,581,671,451,451,483,299,1011,432,422,207,106,701,508,555,508,555,125,870,555,589,508,125,749,482,125,125,130,544,643,643,544,488,22,643,130,335,544,22,130,544,544,488,426,426,4,180,4,695,35,54,433,500,592,433,262,94,401,401,106,216,216,106,521,102,462,518,271,475,365,193,648,206,424,206,193,206,206,424,299,590,590,364,621,67,538,488,567,51,51,513,194,81,488,486,289,567,563,749,563,338,338,502,563,822,338,563,338,502,201,230,201,533,445,175,201,175,13,85,960,103,85,175,30,445,445,175,573,196,877,287,356,678,235,489,312,572,264,717,138,295,6,295,523,55,165,165,295,138,663,6,295,6,353,138,6,138,169,129,784,12,129,194,605,784,445,234,627,563,689,627,647,570,627,570,647,206,234,215,234,816,627,816,234,627,215,234,627,264,427,427,30,424,161,161,916,740,180,616,481,514,383,265,481,164,650,121,582,689,420,669,589,420,788,549,165,734,280,224,146,681,788,184,398,784,4,398,417,417,398,636,784,417,81,398,417,81,185,827,420,241,420,41,185,185,718,241,101,185,185,241,241,241,241,241,185,324,420,420,1011,420,827,241,184,563,241,183,285,529,285,808,822,891,822,488,285,486,619,55,869,39,567,39,289,203,158,289,710,818,158,818,355,29,409,203,308,648,792,308,308,91,308,6,592,792,106,106,308,41,178,91,751,91,259,734,166,36,327,166,230,205,205,172,128,230,432,623,838,623,432,278,432,42,916,432,694,623,352,452,93,314,93,93,641,88,970,914,230,61,159,270,159,493,159,755,159,409,30,30,836,128,241,99,102,984,538,102,102,273,639,838,102,102,136,637,508,627,285,465,327,327,21,749,327,749,21,845,21,21,409,749,1367,806,616,714,253,616,714,714,112,375,21,112,375,375,51,51,51,51,393,206,870,713,193,802,21,1061,42,382,42,543,876,42,876,382,696,543,635,490,353,353,417,64,1257,271,64,377,127,127,537,417,905,353,538,465,605,876,427,324,514,852,427,53,427,557,173,173,7,1274,563,31,31,31,745,392,289,230,230,230,91,218,327,420,420,128,901,552,420,230,608,552,476,347,476,231,159,137,716,648,716,627,740,718,679,679,6,718,740,6,189,679,125,159,757,1191,409,175,250,409,67,324,681,605,550,398,550,931,478,174,21,316,91,316,654,409,425,425,699,61,699,321,698,321,698,61,425,699,321,409,699,299,335,321,335,61,698,699,654,698,299,425,231,14,121,515,121,14,165,81,409,189,81,373,465,463,1055,507,81,81,189,1246,321,409,886,104,842,689,300,740,380,656,656,832,656,380,300,300,206,187,175,142,465,206,271,468,215,560,83,215,83,215,215,83,175,215,83,83,111,206,756,559,756,1367,206,559,1015,559,559,946,1015,548,559,756,1043,756,698,159,414,308,458,997,663,663,347,39,755,838,323,755,323,159,159,717,159,21,41,128,516,159,717,71,870,755,159,740,717,374,516,740,51,148,335,148,335,791,

120,364,335,335,51,120,251,538,251,971,1395,538,78,178,538,538,918,129,918,129,538,538,656,129,538,538,129,538,1051,538,128,838,931,998,823,1095,334,870,334,367,550,1061,498,745,832,498,745,716,498,498,128,997,832,716,832,130,642,616,497,432,432,432,432,642,159,432,46,230,788,160,230,478,46,693,103,920,230,589,643,160,616,432,165,165,583,592,838,784,583,710,6,583,583,6,35,230,838,592,710,6,589,230,838,30,592,583,6,583,6,6,583,30,30,6,375,375,99,36,1158,425,662,417,681,364,375,1025,538,822,669,893,538,538,450,409,632,527,632,563,632,527,550,71,698,550,39,550,514,537,514,537,111,41,173,592,173,648,173,173,173,1011,514,173,173,514,166,648,355,161,166,648,497,327,327,550,650,21,425,605,555,103,425,605,842,836,1011,636,138,756,836,756,756,353,1011,636,636,1158,741,741,842,756,741,1011,677,1011,770,366,306,488,920,920,665,775,502,500,775,775,648,364,833,207,13,93,500,364,500,665,500,93,295,183,1293,313,272,313,279,303,93,516,93,1013,381,6,93,93,303,259,643,168,673,230,1261,230,230,673,1060,1079,1079,550,741,741,590,527,741,741,442,741,442,848,741,590,925,219,527,925,335,442,590,239,590,590,590,239,527,239,1033,230,734,241,741,230,549,548,1015,1015,32,36,433,465,724,465,73,73,73,465,808,73,592,1430,250,154,154,250,538,353,353,353,353,353,175,194,206,538,632,1163,960,175,175,538,452,632,1163,175,538,960,194,175,194,632,960,632,94,632,461,960,1163,1163,461,632,960,755,707,105,382,625,382,382,784,707,871,559,387,387,871,784,559,784,88,36,570,314,1028,975,335,335,398,573,573,573,21,215,562,738,612,424,21,103,788,870,912,23,186,757,73,818,23,73,563,952,262,563,137,262,1022,952,137,1273,442,952,604,137,308,384,913,235,325,695,398,95,668,776,713,309,691,22,10,364,682,682,578,481,1252,1072,1252,825,578,825,1072,1149,592,273,387,273,427,155,1204,50,452,50,1142,50,367,452,1142,611,367,50,50,367,50,1675,99,367,50,1501,1099,830,681,689,917,1089,453,425,235,918,538,550,335,161,387,859,324,21,838,859,1123,21,723,21,335,335,206,21,364,1426,21,838,838,335,364,21,21,859,920,838,838,397,81,639,397,397,588,933,933,784,222,830,36,36,222,1251,266,36,146,266,366,581,605,366,22,966,681,681,433,730,1013,550,21,21,938,488,516,21,21,656,420,323,323,323,327,323,918,581,581,830,361,830,364,259,364,496,496,364,691,705,691,475,427,1145,600,179,427,527,749,869,689,335,347,220,298,689,1426,183,554,55,832,550,550,165,770,957,67,1386,219,683,683,355,683,355,355,738,355,842,931,266,325,349,256,1113,256,423,960,554,554,325,554,508,22,142,22,508,916,767,55,1529,767,55,1286,93,972,550,931,1286,1286,972,93,1286,1392,890,93,1286,93,1286,972,374,931,890,808,779,975,975,175,173,4,681,383,1367,173,383,1367,383,173,175,69,238,146,238,36,148,888,238,173,238,148,238,888,185,925,925,797,925,815,925,469,784,289,784,925,797,925,925,1093,925,925,925,1163,797,797,815,925,1093,784,636,663,925,187,922,316,1380,709,916,916,187,355,948,916,187,916,916,948,948,916,355,316,316,334,300,1461,36,583,1179,699,235,858,583,699,858,699,1189,1256,1189,699,797,699,

699,699,699,427,488,427,488,175,815,656,656,150,322,465,322,870,465,1099,582,665,767,749,635,749,600,1448,36,502,235,502,355,502,355,355,355,172,355,355,95,866,425,393,1165,42,42,42,393,939,909,909,836,552,424,1333,852,897,1426,1333,1446,1426,997,1011,852,1198,55,32,239,588,681,681,239,1401,32,588,239,462,286,1260,984,1160,960,960,486,828,462,960,1199,581,850,663,581,751,581,581,1571,252,252,1283,264,430,264,430,430,842,252,745,21,307,681,1592,488,857,857,1161,857,857,857,138,374,374,1196,374,1903,1782,1626,414,112,1477,1040,356,775,414,414,112,356,775,435,338,1066,689,689,1501,689,1249,205,689,765,220,308,917,308,308,220,327,387,838,917,917,917,220,662,308,220,387,387,220,220,308,308,308,387,1009,1745,822,279,554,1129,543,383,870,1425,241,870,241,383,716,592,21,21,592,425,550,550,550,427,230,57,483,784,860,57,308,57,486,870,447,486,433,433,870,433,997,486,443,433,433,997,486,1292,47,708,81,895,394,81,935,81,81,81,374,986,916,1103,1095,465,495,916,667,1745,518,220,1338,220,734,1294,741,166,828,741,741,1165,1371,1371,471,1371,647,1142,1878,1878,1371,1371,822,66,327,158,427,427,465,465,676,676,30,30,676,676,893,1592,93,455,308,582,695,582,629,582,85,1179,85,85,1592,1179,280,1027,681,398,1027,398,295,784,740,509,425,968,509,46,833,842,401,184,401,464,6,1501,1501,550,538,883,538,883,883,883,1129,550,550,333,689,948,21,21,241,2557,2094,273,308,58,863,893,1086,409,136,1086,592,592,830,830,883,830,277,68,689,902,277,453,507,129,689,630,664,550,128,1626,1626,128,902,312,589,755,755,589,755,407,1782,589,784,1516,1118,407,407,1447,589,235,755,1191,235,235,407,128,589,1118,21,383,1331,691,481,383,1129,1129,1261,1104,1378,1129,784,1129,1261,1129,947,1129,784,784,1129,1129,35,1104,35,866,1129,1129,64,481,730,1260,481,970,481,481,481,481,863,481,681,699,863,486,681,481,481,55,55,235,1364,944,632,822,401,822,952,822,822,99,550,2240,550,70,891,860,860,550,550,916,1176,1530,425,1530,916,628,1583,916,628,916,916,628,628,425,916,1062,1265,916,916,916,280,461,916,916,1583,628,1062,916,916,677,1297,924,1260,83,1260,482,433,234,462,323,1656,997,323,323,931,838,931,1933,1391,367,323,931,1391,1391,103,1116,1116,1116,769,1195,1218,312,791,312,741,791,997,312,334,334,312,287,287,633,1397,1426,605,1431,327,592,705,1194,592,1097,1118,1503,1267,1267,1267,618,1229,734,1089,785,1089,1129,1148,1148,1089,915,1148,1129,1148,1011,1011,1229,871,1560,1560,1560,563,1537,1009,1560,632,985,592,1308,592,882,145,145,397,837,383,592,592,832,36,2714,2107,1588,1347,36,36,1443,1453,334,2230,1588,1169,650,1169,2107,425,425,891,891,425,2532,679,274,274,274,325,274,1297,194,1297,627,314,917,314,314,1501,414,1490,1036,592,1036,1025,901,1218,1025,901,280,592,592,901,1461,159,159,159,2076,1066,1176,1176,516,327,516,1179,1176,899,1176,1176,323,1187,1229,663,1229,504,1229,916,1229,916,1661,41,36,278,1027,648,648,648,1626,648,646,1179,1580,1061,1514,1008,1741,2076,1514,1008,952,1089,427,952,427,1083,425,427,1089,1083,425,427,425,230,920,1678,920,

1678,189,189,953,189,133,189,1075,189,189,133,1264,725,189,1629,189,808,230,230,2179,770,230,770,230,21,21,784,1118,230,230,230,770,1118,986,808,916,30,327,918,679,414,916,1165,1355,916,755,733,433,1490,433,433,433,605,433,433,433,1446,679,206,433,21,2452,206,206,433,1894,206,822,206,2073,206,206,21,822,21,206,206,21,383,1513,375,1347,432,1589,172,954,242,1256,1256,1248,1256,1256,1248,1248,1256,842,13,592,13,842,1291,592,21,175,13,592,13,13,1426,13,1541,445,808,808,863,647,219,1592,1029,1225,917,1963,1129,555,1313,550,660,550,220,660,552,663,220,533,220,383,550,1278,1495,636,842,1036,425,842,425,1537,1278,842,554,1508,636,554,301,842,792,1392,1021,284,1172,997,1021,103,1316,308,1210,848,848,1089,1089,848,848,67,1029,827,1029,2078,827,1312,1029,827,590,872,1312,427,67,67,67,67,872,827,872,2126,1436,26,2126,67,1072,2126,1610,872,1620,883,883,1397,1189,555,555,563,1189,555,640,555,640,1089,1089,610,610,1585,610,1355,610,1015,616,925,1015,482,230,707,231,888,1355,589,1379,151,931,1486,1486,393,235,960,590,235,960,422,142,285,285,327,327,442,2009,822,445,822,567,888,2611,1537,323,55,1537,323,888,2611,323,1537,323,58,445,593,2045,593,58,47,770,842,47,47,842,842,648,2557,173,689,2291,1446,2085,2557,2557,2291,1780,1535,2291,2391,808,691,1295,1165,983,948,2000,948,983,983,2225,2000,983,983,705,948,2000,1795,1592,478,592,1795,1795,663,478,1790,478,592,1592,173,901,312,4,1606,173,838,754,754,128,550,1166,551,1480,550,550,1875,1957,1166,902,1875,550,550,551,2632,551,1875,1875,551,2891,2159,2632,3231,551,815,150,1654,1059,1059,734,770,555,1592,555,2059,770,770,1803,627,627,627,2059,931,1272,427,1606,1272,1606,1187,1204,397,822,21,1645,263,263,822,263,1645,280,263,605,1645,2014,21,21,1029,263,1916,2291,397,397,496,270,270,1319,264,1638,264,986,1278,1397,1278,1191,409,1191,740,1191,754,754,387,63,948,666,666,1198,548,63,1248,285,1248,169,1248,1248,285,918,224,285,1426,1671,514,514,717,514,51,1521,1745,51,605,1191,51,128,1191,51,51,1521,267,513,952,966,1671,897,51,71,592,986,986,1121,592,280,2000,2000,1165,1165,1165,1818,222,1818,1165,1252,506,327,443,432,1291,1291,2755,1413,520,1318,227,1047,828,520,347,1364,136,136,452,457,457,132,457,488,1087,1013,2225,32,1571,2009,483,67,483,740,740,1013,2854,866,32,2861,866,887,32,2444,740,32,32,866,2225,866,32,1571,2627,32,850,1675,569,1158,32,1158,1797,2641,1565,1158,569,1797,1158,1797,55,1703,42,55,2562,675,1703,42,55,749,488,488,347,1206,1286,1286,488,488,1206,1286,1206,1286,550,550,1790,860,550,2452,550,550,2765,1089,1633,797,2244,1313,194,2129,194,194,194,818,32,194,450,1313,2387,194,1227,2387,308,2232,526,476,278,830,830,194,830,194,278,194,714,476,830,714,830,278,830,2532,1218,1759,1446,960,1747,187,1446,1759,960,105,1446,1446,1271,1446,960,960,1218,1446,1446,105,1446,960,488,1446,427,534,842,1969,2460,1969,842,842,1969,427,941,2160,427,230,938,2075,1675,1675,895,1675,34,129,1811,239,749,1957,2271,749,1908,129,239,239,129,129,2271,2426,1355,1756,194,

1583,194,194,1583,194,1355,194,1628,2221,1269,2425,1756,1355,1355,1583,1033,427,582,30,582,582,935,1444,1962,915,733,915,938,1962,767,353,1630,1962,1962,563,733,563,733,353,822,1630,740,2076,2076,2076,589,589,2636,866,589,947,1528,125,273,1058,1058,1161,1635,1355,1161,1161,1355,1355,650,1206,1206,784,784,784,784,784,412,461,412,2240,412,679,891,461,679,679,189,189,1933,1651,2515,189,1386,538,1386,1386,1187,1386,2423,2601,2285,175,175,2331,194,3079,384,538,2365,2294,538,2166,1841,3326,1256,3923,976,85,550,550,1295,863,863,550,1249,550,1759,146,1069,920,2633,885,885,1514,1489,166,1514,2041,885,2456,885,2041,1081,1948,362,550,94,324,2308,94,2386,94,550,874,1329,1759,2280,1487,493,493,2099,2599,1431,1086,1514,1086,2099,1858,368,1330,2599,1858,2846,2846,2907,2846,713,713,1854,1123,713,713,3010,1123,3010,538,713,1123,447,822,555,2011,493,508,2292,555,1736,2135,2704,555,2814,555,2000,555,555,822,914,327,679,327,648,537,2263,931,1496,537,1296,1745,1592,1658,1795,650,1592,1745,1745,1658,1592,1745,1592,1745,1658,1338,2124,1592,1745,1745,1745,837,1726,2897,1118,1118,230,1118,1118,1118,1388,1748,514,128,1165,931,514,2974,2041,2387,2041,979,185,36,1269,550,173,812,36,1165,2676,2562,1473,2885,1982,1578,1578,383,383,2360,383,1578,2360,1584,1982,1578,1578,1578,2019,1036,355,724,2023,205,303,355,1036,1966,355,1036,401,401,401,830,401,849,578,401,849,849,578,1776,1123,552,2632,808,1446,1120,373,1529,1483,1057,893,1284,1430,1529,1529,2632,1352,2063,1606,1352,1606,2291,3079,2291,1529,506,838,1606,1606,1352,1529,1529,1483,1529,1606,1529,259,902,259,902,612,612,284,398,2991,1534,1118,1118,1118,1118,1118,734,284,2224,398,734,284,734,398,3031,398,734,1707,2643,1344,1477,475,1818,194,1894,691,1528,1184,1207,1501,6,2069,871,2069,3548,1443,2069,2685,3265,1350,3265,2069,2069,128,1313,128,663,414,1313,414,2000,128,2000,663,1313,699,1797,550,327,550,1526,699,327,1797,1526,550,550,327,550,1426,1426,1426,2285,1123,890,728,1707,728,728,327,253,1187,1281,1364,1571,2170,755,3232,925,1496,2170,2170,1125,443,902,902,925,755,2078,2457,902,2059,2170,1643,1129,902,902,1643,1129,606,36,103,338,338,1089,338,338,338,1089,338,36,340,1206,1176,2041,833,1854,1916,1916,1501,2132,1736,3065,367,1934,833,833,833,2041,3017,2147,818,1397,828,2147,398,828,818,1158,818,689,327,36,1745,2132,582,1475,189,582,2132,1191,582,2132,1176,1176,516,2610,2230,2230,64,1501,537,1501,173,2230,2988,1501,2694,2694,537,537,173,173,1501,537,64,173,173,64,2230,537,2230,537,2230,2230,2069,3142,1645,689,1165,1165,1963,514,488,1963,1145,235,1145,1078,1145,231,2405,552,21,57,57,57,1297,1455,1988,2310,1885,2854,2014,734,1705,734,2854,734,677,1988,1660,734,677,734,677,677,734,2854,1355,677,1397,2947,2386,1698,128,1698,3028,2386,2437,2947,2386,2643,2386,2804,1188,335,746,1187,1187,861,2519,1917,2842,1917,675,1308,234,1917,314,314,2339,2339,2592,2576,902,916,2339,916,2339,916,2339,916,1089,1089,2644,1221,1221,2446,308,308,2225,2225,3192,2225,555,1592,1592,555,893,555,

550,770,3622,2291,2291,3419,465,250,2842,2291,2291,2291,935,160,1271,308,325,935,1799,1799,1891,2227,1799,1598,112,1415,1840,2014,1822,2014,677,1822,1415,1415,1822,2014,2386,2159,1822,1415,1822,179,1976,1033,179,1840,2014,1415,1970,1970,1501,563,563,563,462,563,1970,1158,563,563,1541,1238,383,235,1158,383,1278,383,1898,2938,21,2938,1313,2201,2059,423,2059,1313,872,1313,2044,89,173,3327,1660,2044,1623,173,1114,1114,1592,1868,1651,1811,383,3469,1811,1651,869,383,383,1651,1651,3223,2166,3469,767,383,1811,767,2323,3355,1457,3341,2640,2976,2323,3341,2323,2640,103,103,1161,1080,2429,370,2018,2854,2429,2166,2429,2094,2207,871,1963,1963,2023,2023,2336,663,2893,1580,691,663,705,2046,2599,409,2295,1118,2494,1118,1950,549,2494,2453,2046,2494,2453,2046,2453,2046,409,1118,4952,2291,2225,1894,1423,2498,567,4129,1475,1501,795,463,2084,828,828,232,828,232,232,1818,1818,666,463,232,220,220,2162,2162,833,4336,913,35,913,21,2927,886,3037,383,886,876,1747,383,916,916,916,2927,916,1747,837,1894,717,423,481,1894,1059,2262,3206,4700,1059,3304,2262,871,1831,871,3304,1059,1158,1934,1158,756,1511,41,978,1934,2603,720,41,756,41,325,2611,1158,173,1123,1934,1934,1511,2045,2045,2045,1423,3206,3691,2512,3206,2512,2000,1811,2504,2504,2611,2437,2437,2437,1455,893,150,2665,1966,605,398,2331,1177,516,1962,4241,94,1252,760,1292,1962,1373,2000,1990,3684,42,1868,3779,1811,1811,2041,3010,5436,1780,2041,1868,1811,1780,1811,1868,1811,2041,1868,1811,5627,4274,1811,1868,4602,1811,1811,1474,2665,235,1474,2665

[0401] 附录B.1用于表V_0的值

[0402] 这些值表示用于上述应用中小节B.5.4.1所述的表V_0的示例值集。每个条目是10进制表示的32-比特整数。值的顺序是从第一列的顶部到第一列的底部，然后是第二列的顶部，依此类推。

[0403]

251291136	3676575285	1255426612	138853680	2213589897
3952231631	2581671944	4213663089	4160398285	2734638932
3370958628	3312220480	1521339547	3322241130	1129721478
4070167936	681232419	3041843489	2068983570	3187422815
123631495	307306866	420130494	2247491078	2859178611
3351110283	4112503940	10677091	3669524410	3284308411
3218676425	1158111502	515623176	1575146607	3819792700
2011642291	709227802	3457502702	828029864	3557526733
774603218	2724140433	2115821274	3732001371	451874476
2402805061	4201101115	2720124766	3422026452	1740576081
1004366930	4215970289	3242576090	3370954177	3592838701
1843948209	4048876515	854310108	4006626915	1709429513
428891132	3031661061	425973987	543812220	3702918379
3746331984	1909085522	325832382	1243116171	3533351328
1591258008	510985033	1796851292	3928372514	1641660745
3067016507	1361682810	2462744411	2791443445	179350258
1433388735	129243379	1976681690	4081325272	2380520112
504005498	3142379587	1408671665	2280435605	3936163904
2032657933	2569842483	1228817808	885616073	3685256204
3419319784	3033268270	3917210003	616452097	3156252216
2805686246	1658118006	263976645	3188863436	1854258901
3102436986	932109358	2593736473	2780382310	2861641019
3808671154	1982290045	2471651269	2340014831	3176611298
2501582075	2983082771	4291353919	1208439576	834787554
3978944421	3007670818	650792940	258356309	331353807
246043949	3448104768	1191583883	3837963200	517858103
4016898363	683749698	3046561335	2075009450	3010168884
649743608	778296777	2466530435	3214181212	4012642001
1974987508	1399125101	2545983082	3303882142	2217188075
2651273766	1939403708	969168436	880813252	3756943137
2357956801	1692176003	2019348792	1355575717	3077882590
689605112	3868299200	2268075521	207231484	2054995199
715807172	1422476658	1169345068	2420803184	3081443129
2722736134	593093658	3250240009	358923368	3895398812
191939188	1878973865	3963499681	1617557768	1141097543
3535520147	2526292949	2560755113	3272161958	2376261053
3277019569	1591602827	911182396	1771154147	2626898255
1470435941	3986158854	760842409	2842106362	2554703076
3763101702	3964389521	3569308693	1751209208	401233789
3232409631	2695031039	2687243553	1421030790	1460049922
122701163	1942050155	381854665	658316681	678083952
3920852693	424618399	2613828404	194065839	1064990737
782246947	1347204291	2761078866	3241510581	940909784
372121310	2669179716	1456668111	38625260	1673396780
2995604341	2434425874	883760091	301875395	528881783
2045698575	2540801947	3294951678	4176141739	1712547446
2332962102	1384069776	1604598575	297312930	3629685652
4005368743	4123580443	1985308198	2137802113	1358307511
218596347	1523670218	1014570543	1502984205	
3415381967	2708475297	2724959607	3669376622	
4207612806	1046771089	3062518035	3728477036	
861117671	2229796016	3115293053	234652930	

[0404] 附录B.2用于表V_1的值

[0405] 这些值表示用于上述应用中小节B.5.4.1所述的表V_1的示例值集。每个条目是10进制表示的32-比特整数。值的顺序是从第一列的顶部到第一列的底部，然后是第二列的顶部，依此类推。

[0406]

807385413	3843885867	3545667983	644189126	922673938
2043073223	4201106668	332038910	226475395	3877430102
3336749796	415906198	976628269	307789415	3422391938
1302105833	19296841	3123492423	1196105631	1414347295
2278607931	2402488407	3041418372	3191691839	1971054608
541015020	2137119134	2258059298	782852669	3061798054
1684564270	1744097284	2139377204	1608507813	830555096
372709334	579965637	3243642973	1847685900	2822905141
3508252125	2037662632	3226247917	4069766876	167033190
1768346005	852173610	3674004636	3931548641	1079139428
1270451292	2681403713	2698992189	2526471011	4210126723
2603029534	1047144830	3453843574	766865139	3593797804
2049387273	2982173936	1963216666	2115084288	429192890
3891424859	910285038	3509855005	4259411376	372093950
2152948345	4187576520	2358481858	3323683436	1779187770
4114760273	2589870048	747331248	568512177	3312189287
915180310	989448887	1957348676	3736601419	204349348
3754787998	3292758024	1097574450	1800276898	452421568
700503826	506322719	2435697214	4012458395	2800540462
2131559305	176010738	3870972145	1823982	3733109044
1308908630	1865471968	1888833893	27980198	1235082423
224437350	2619324712	2914085525	2023839966	1765319556
4065424007	564829442	4161315584	869505096	3174729780
3638665944	1996870325	1273113343	431161506	3762994475
1679385496	339697593	3269644828	1024804023	3171962488
3431345226	4071072948	3681293816	1853869307	442160826
1779595665	3618966336	412536684	3393537983	198349622
3068494238	2111320126	1156034077	1500703614	45942637
1424062773	1093955153	3823026442	3019471560	1324086311
1033448464	957978696	1066971017	1351086955	2901868599
4050396853	892010560	3598330293	3096933631	678860040
3302235057	1854601078	1979273937	3034634988	3812229107
420600373	1873407527	2079029895	2544598006	19936821
2868446243	2498544695	1195045909	1230942551	1119590141
311689386	2694156259	1071986421	3362230798	3640121682
259047959	1927339682	2712821515	159984793	3545931032
4057180909	1650555729	3377754595	491590373	2102949142
1575367248	183933047	2184151095	3993872886	2828208598
4151214153	3061444337	750918864	3681855622	3603378023
110249784	2067387204	2585729879	903593547	4135048896
3006865921	228962564	4249895712	3533062472	
4293710613	3904109414	1832579367	1799803217	
3501256572	1595995433	1192240192	772984149	
998007483	1780701372	946734366	895863112	
499288295	2463145963	31230688	1899036275	
1205710710	307281463	3174399083	4187322100	
2997199489	3237929991	3549375728	101856048	
640417429	3852995239	1642430184	234650315	
3044194711	2398693510	1904857554	3183125617	
486690751	3754138664	861877404	3190039692	
2686640734	522074127	3277825584	525584357	
2394526209	146352474	4267074718	1286834489	
2521660077	4104915256	3122860549	455810374	
49993987	3029415884	666423581	1869181575	

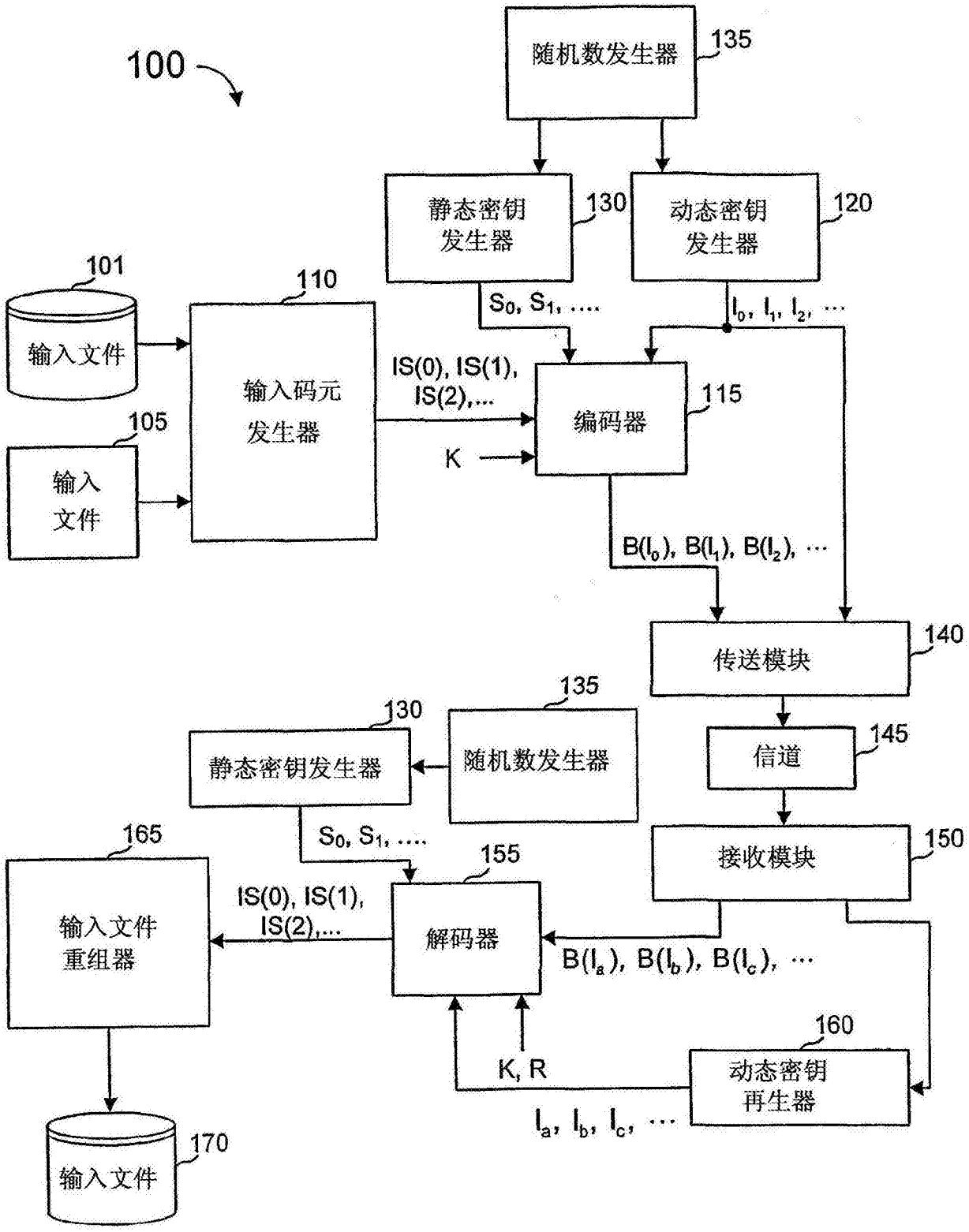

100
随机数发生器
135
静态密钥发生器
130
动态密钥发生器
120
101
输入文件
110
输入码元发生器
105
输入文件
$S_0, S_1, \ldots$
$I_0, I_1, I_2, \ldots$
IS(0), IS(1), IS(2),...
K
编码器
115
$B(I_0), B(I_1), B(I_2), \ldots$
传送模块
140
135
随机数发生器
130
静态密钥发生器
信道
145
165
输入文件重组器
$S_0, S_1, \ldots$
155
解码器
IS(0), IS(1), IS(2),...
接收模块
150
$B(I_a), B(I_b), B(I_c), \ldots$
160
动态密钥再生器
K, R
$I_a, I_b, I_c, \ldots$
170
输入文件

图1

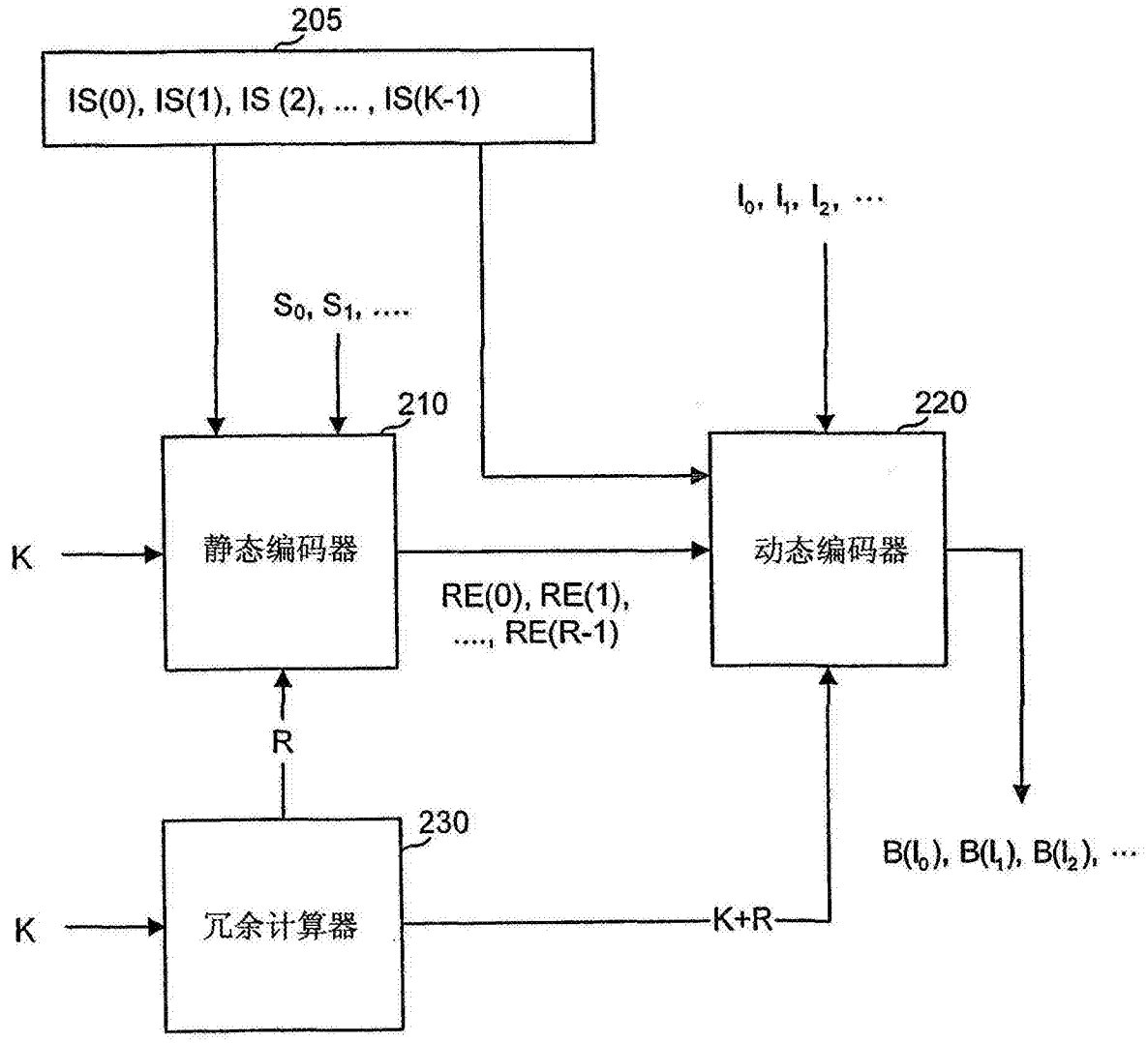

205
IS(0), IS(1), IS (2), ... , IS(K-1)
$I_0, I_1, I_2, \cdots$
$S_0, S_1, \ldots$
210
220
K
静态编码器
动态编码器
RE(0), RE(1),
...., RE(R-1)
R
230
$B(I_0), B(I_1), B(I_2), \cdots$
K
冗余计算器
K+R

图2

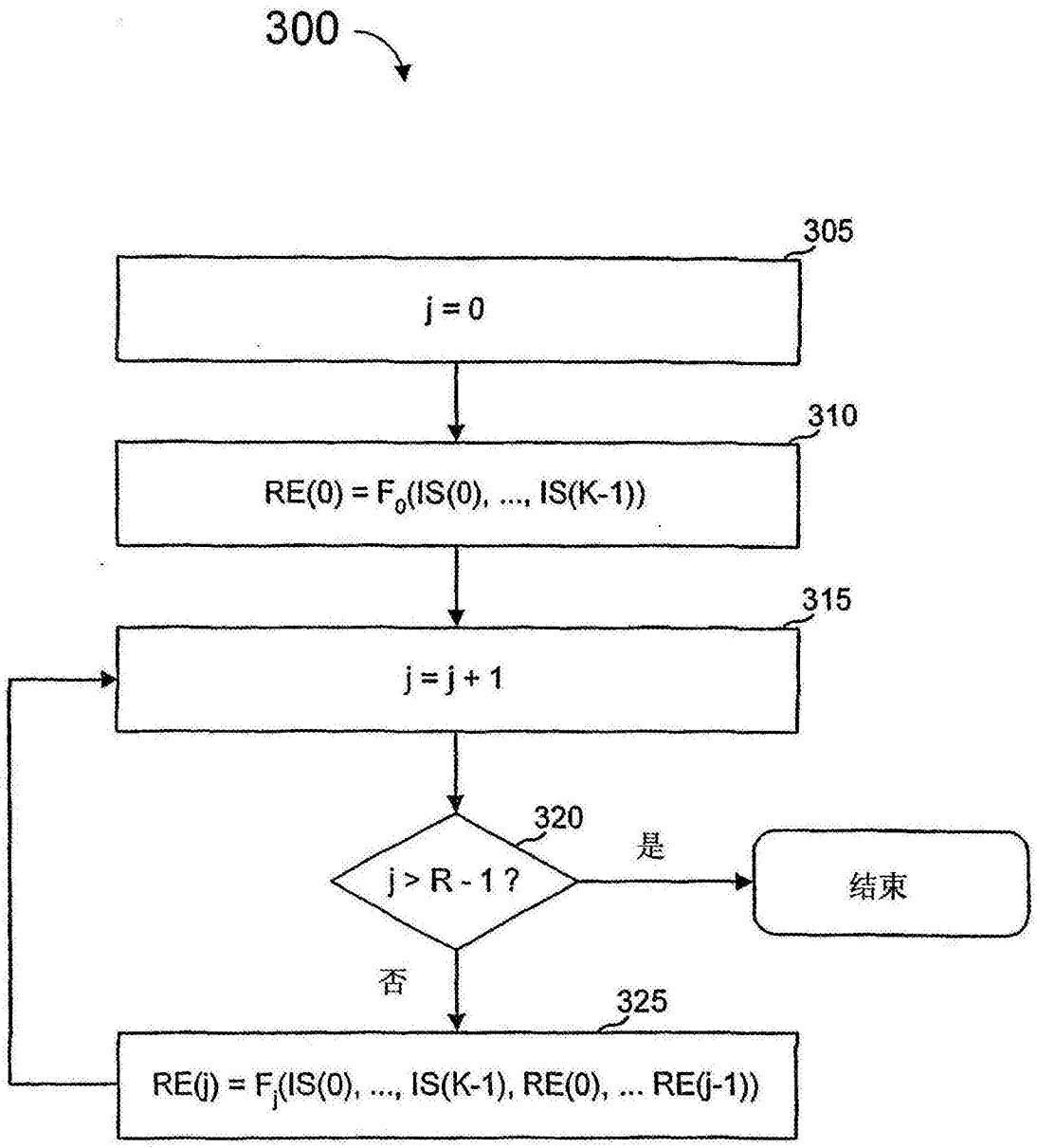

300
305
j = 0
310
RE(0) = F0(IS(0), ..., IS(K-1))
315
j = j + 1
320
j > R - 1 ?
是
结束
否
325
RE(j) = Fj(IS(0), ..., IS(K-1), RE(0), ... RE(j-1))

图3

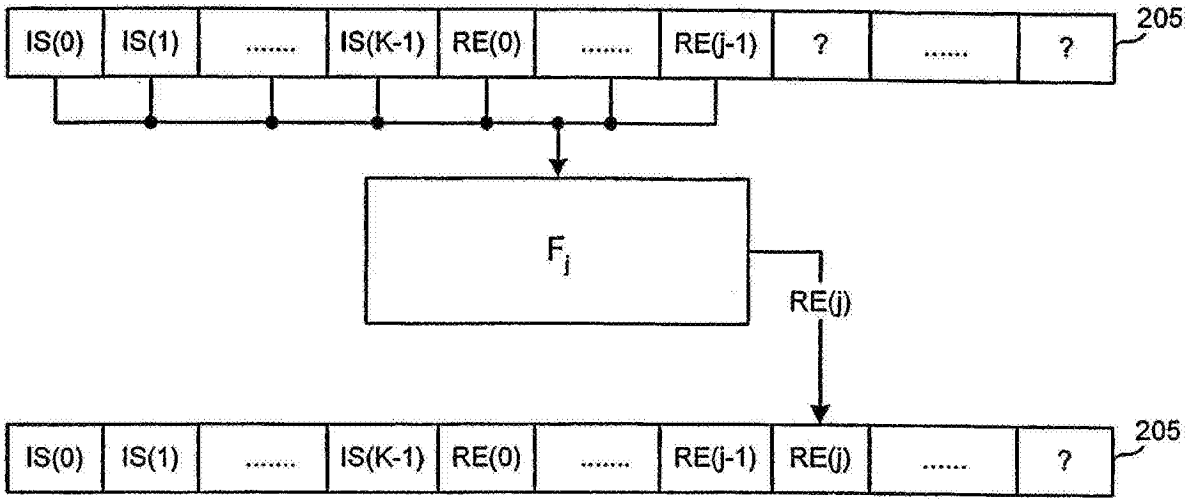

IS(0)
IS(1)
.....
IS(K-1)
RE(0)
.....
RE(j-1)
?
.....
?
205
F_j
RE(j)
IS(0)
IS(1)
.....
IS(K-1)
RE(0)
.....
RE(j-1)
RE(j)
.....
?
205

图4

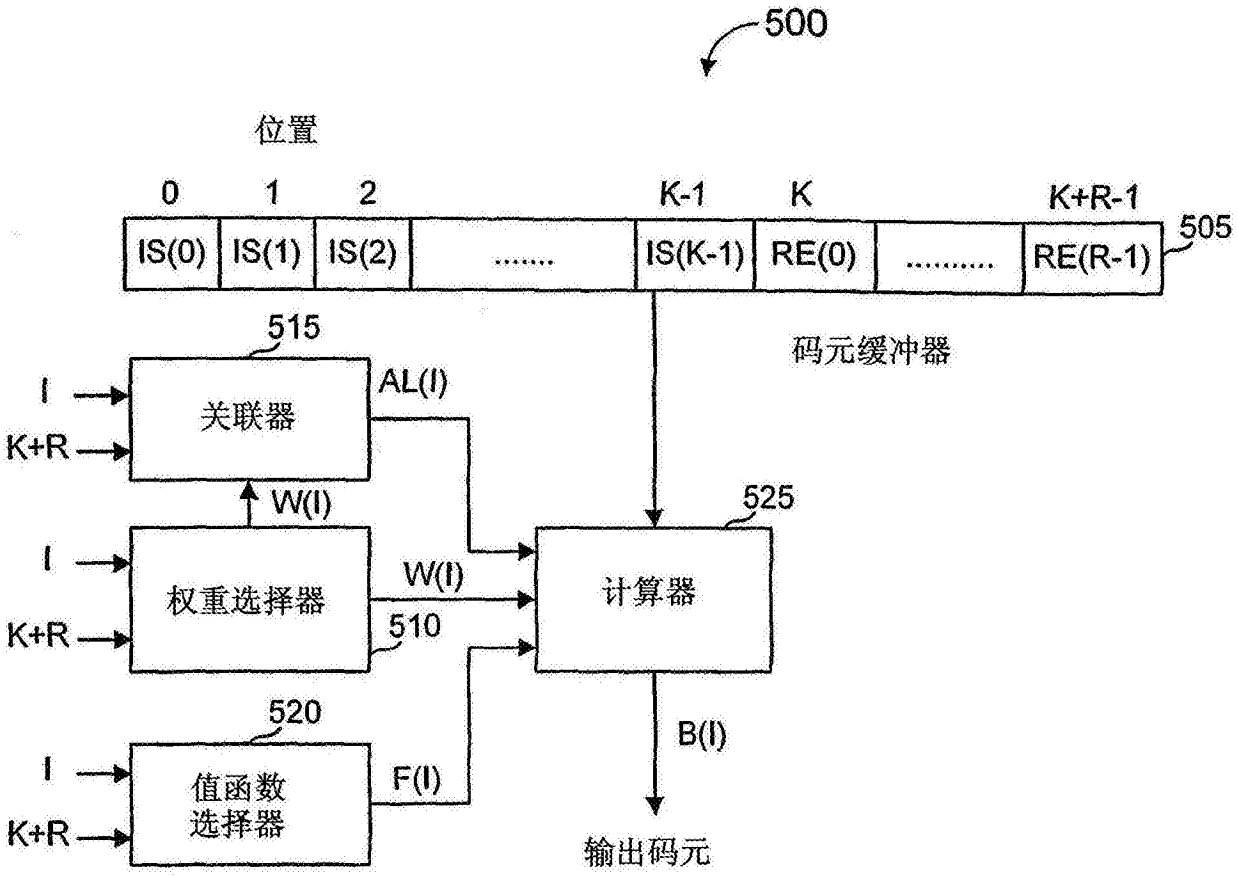

500
位置
0
1
2
K-1
K
K+R-1
IS(0)
IS(1)
IS(2)
.....
IS(K-1)
RE(0)
.....
RE(R-1)
505
码元缓冲器
515
关联器
AL(I)
I
K+R
W(I)
525
权重选择器
W(I)
计算器
510
520
值函数
选择器
F(I)
B(I)
输出码元

图5

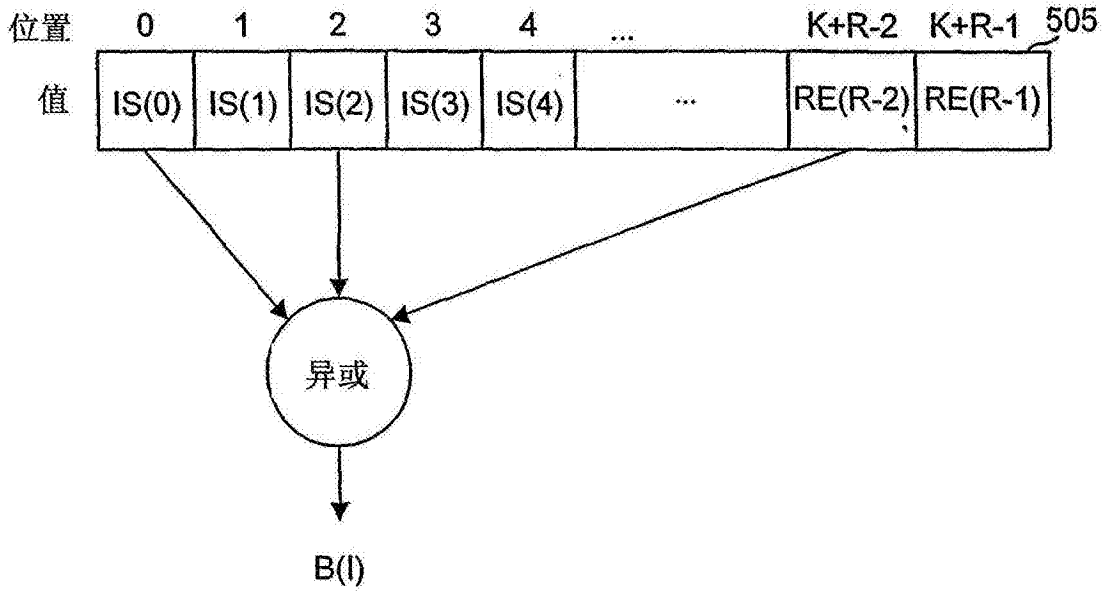


图6

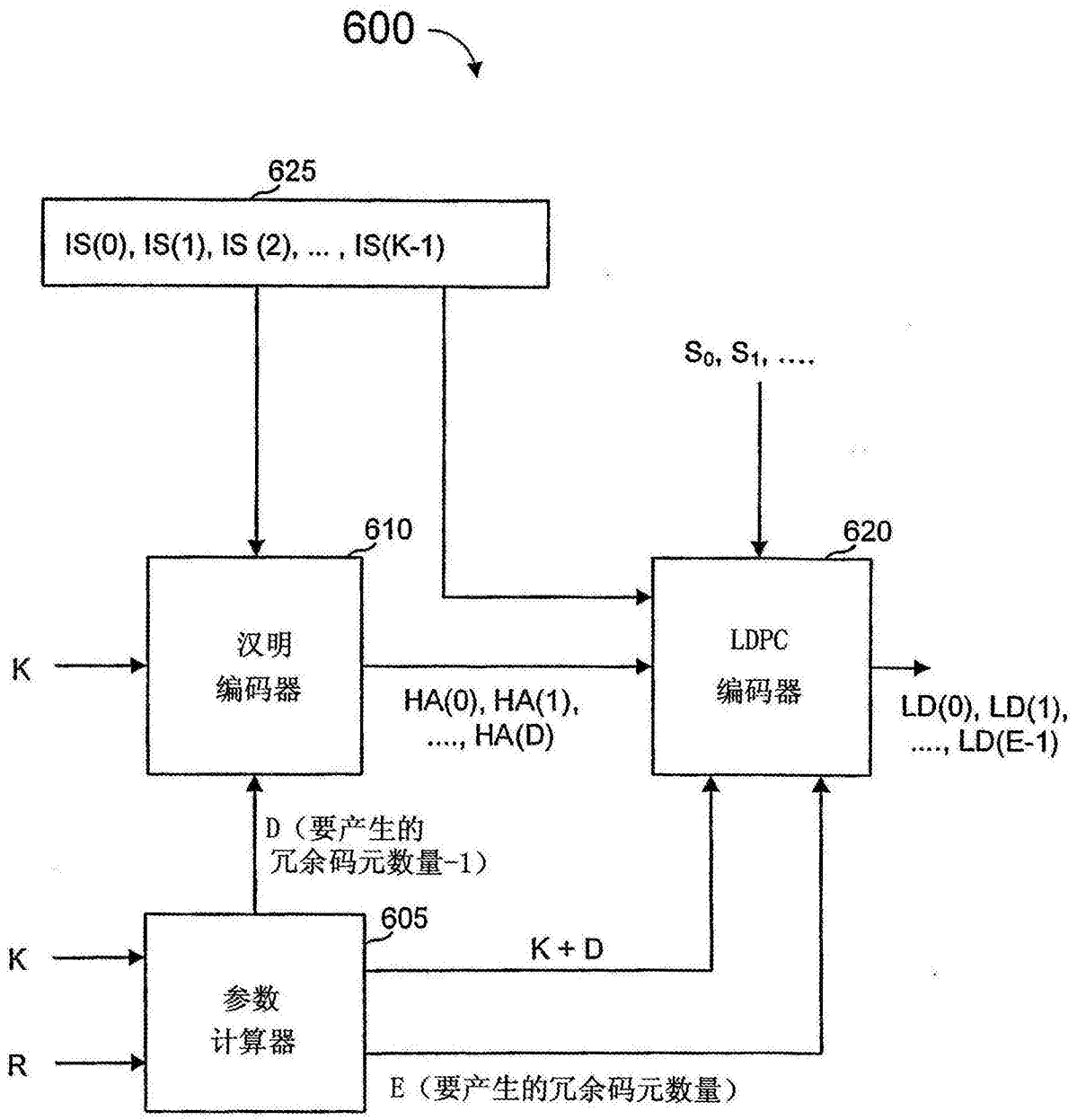

600
625
IS(0), IS(1), IS (2), ... , IS(K-1)
$S_0, S_1, \ldots$
610
620
K
汉明
编码器
HA(0), HA(1),
...., HA(D)
LDPC
编码器
LD(0), LD(1),
...., LD(E-1)
D（要产生的
冗余码元数量-1）
605
K + D
K
参数
计算器
R
E（要产生的冗余码元数量）

图7

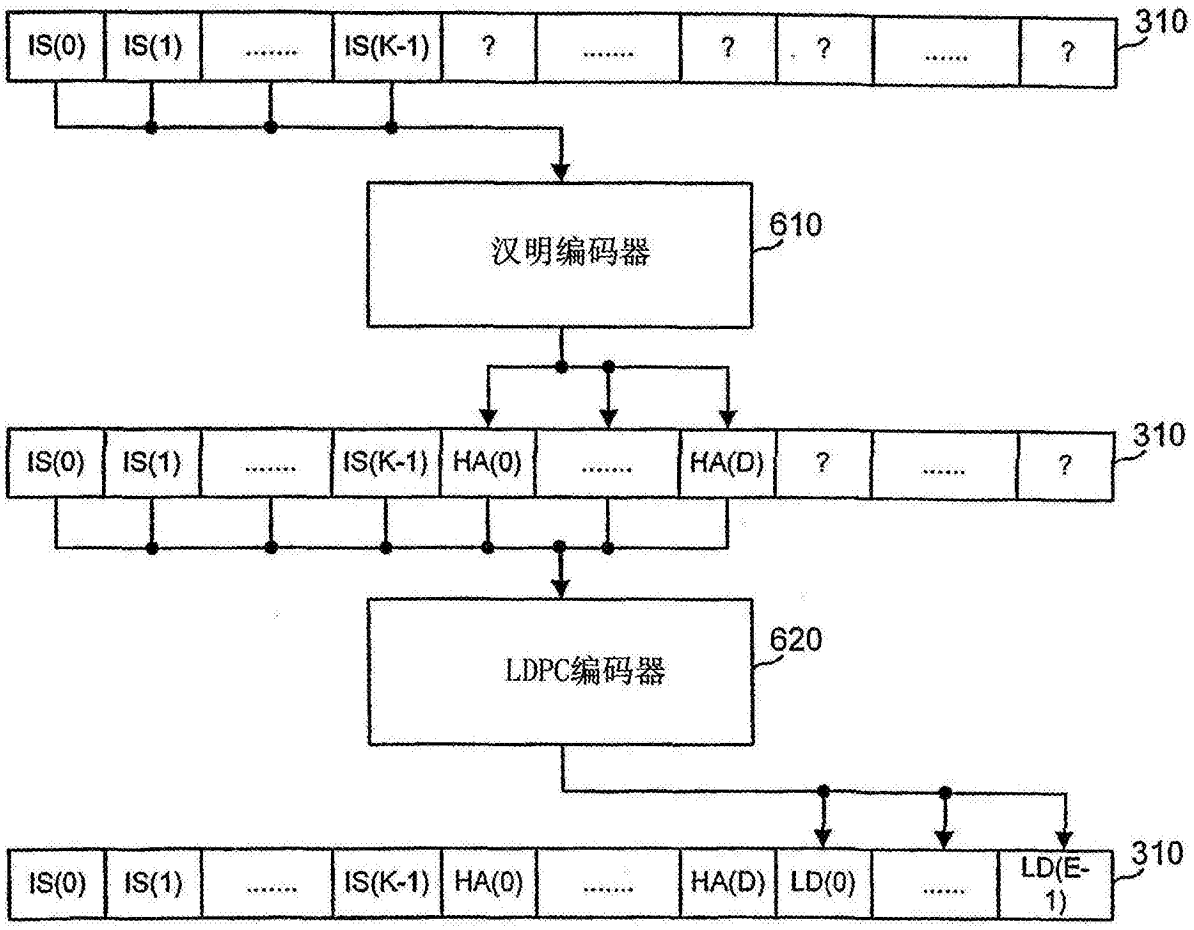

IS(0)
IS(1)
.....
IS(K-1)
?
.....
?
?
.....
?
310
汉明编码器
610
IS(0)
IS(1)
.....
IS(K-1)
HA(0)
.....
HA(D)
?
.....
?
310
LDPC编码器
620
IS(0)
IS(1)
.....
IS(K-1)
HA(0)
.....
HA(D)
LD(0)
.....
LD(E-1)
310

图8

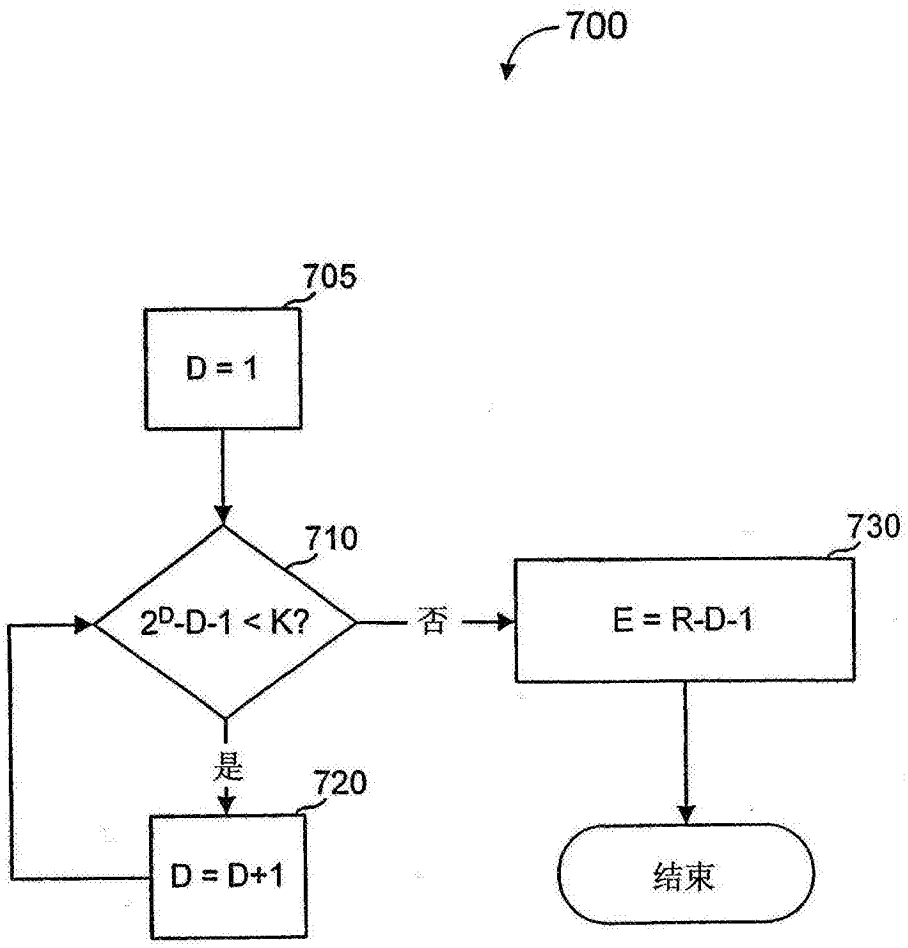


图9

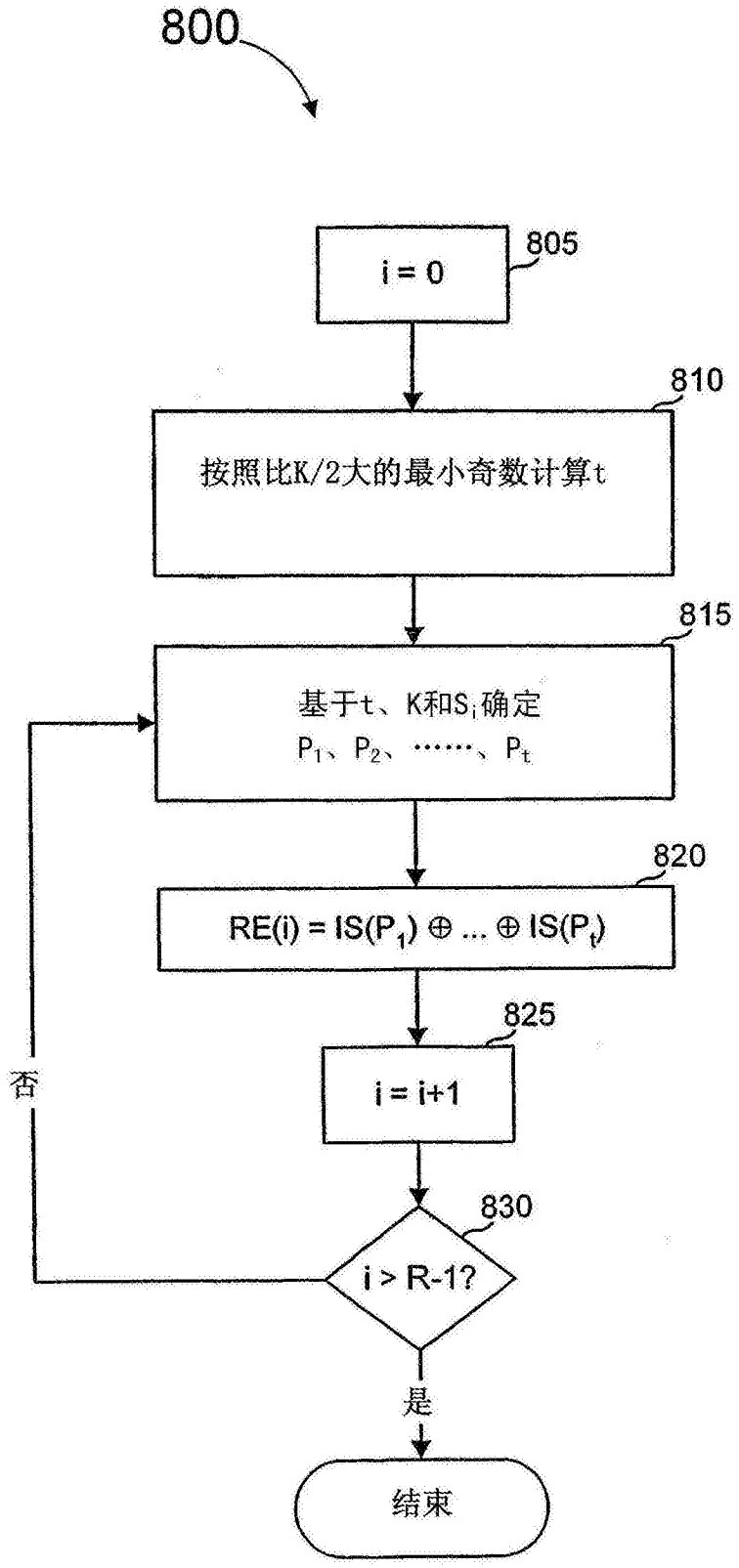

800
i = 0
805
810
按照比K/2大的最小奇数计算t
815
基于t、K和S_i确定
P_1、P_2、……、P_t
820
RE(i) = IS(P_1) ⊕ ... ⊕ IS(P_t)
825
i = i+1
830
i > R-1?
否
是
结束

图10

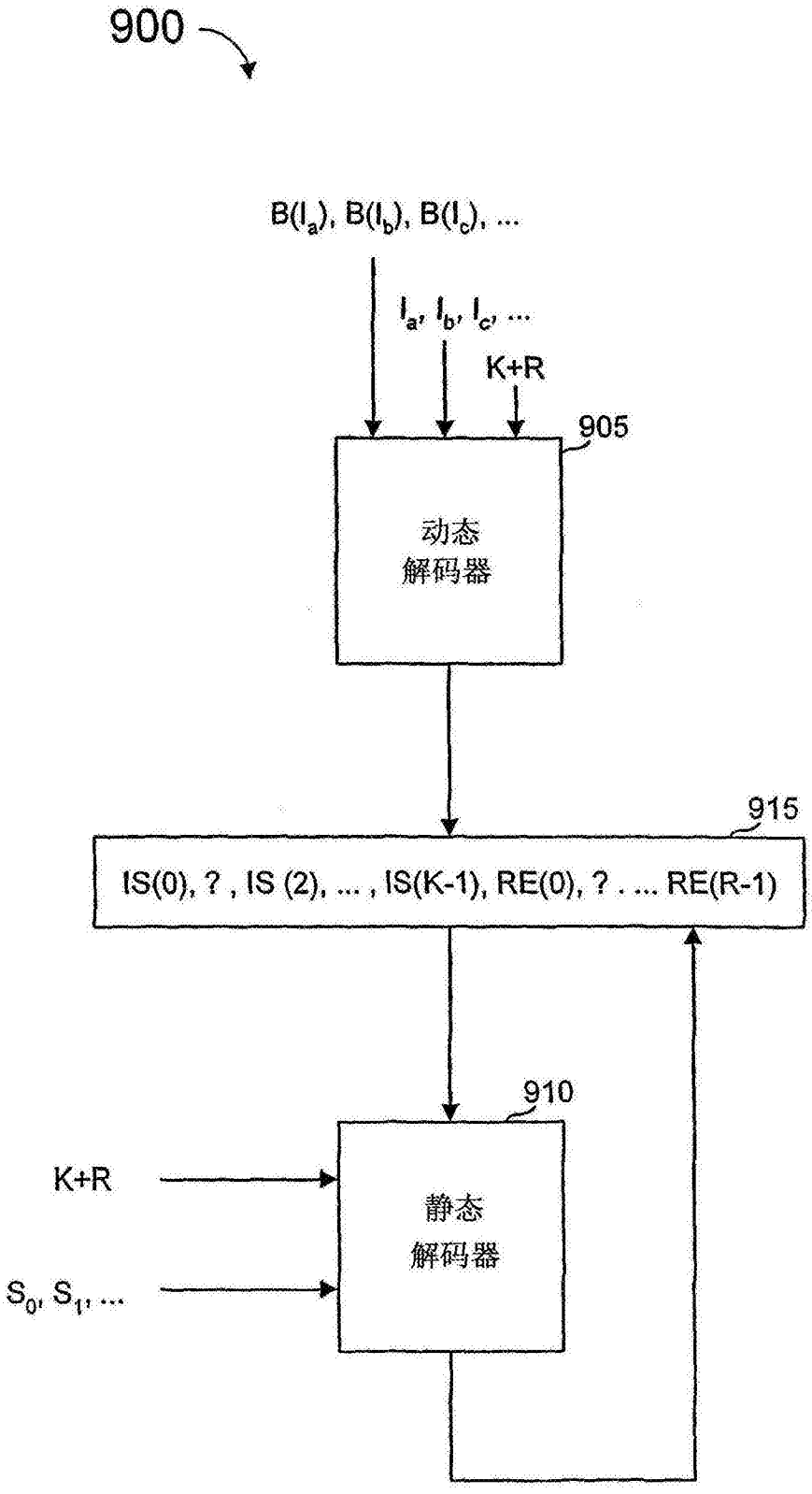

900
B(Ia), B(Ib), B(Ic), ...
Ia, Ib, Ic, ...
K+R
905
动态
解码器
915
IS(0), ? , IS (2), ... , IS(K-1), RE(0), ? . ... RE(R-1)
910
K+R
静态
解码器
S0, S1, ...

图11

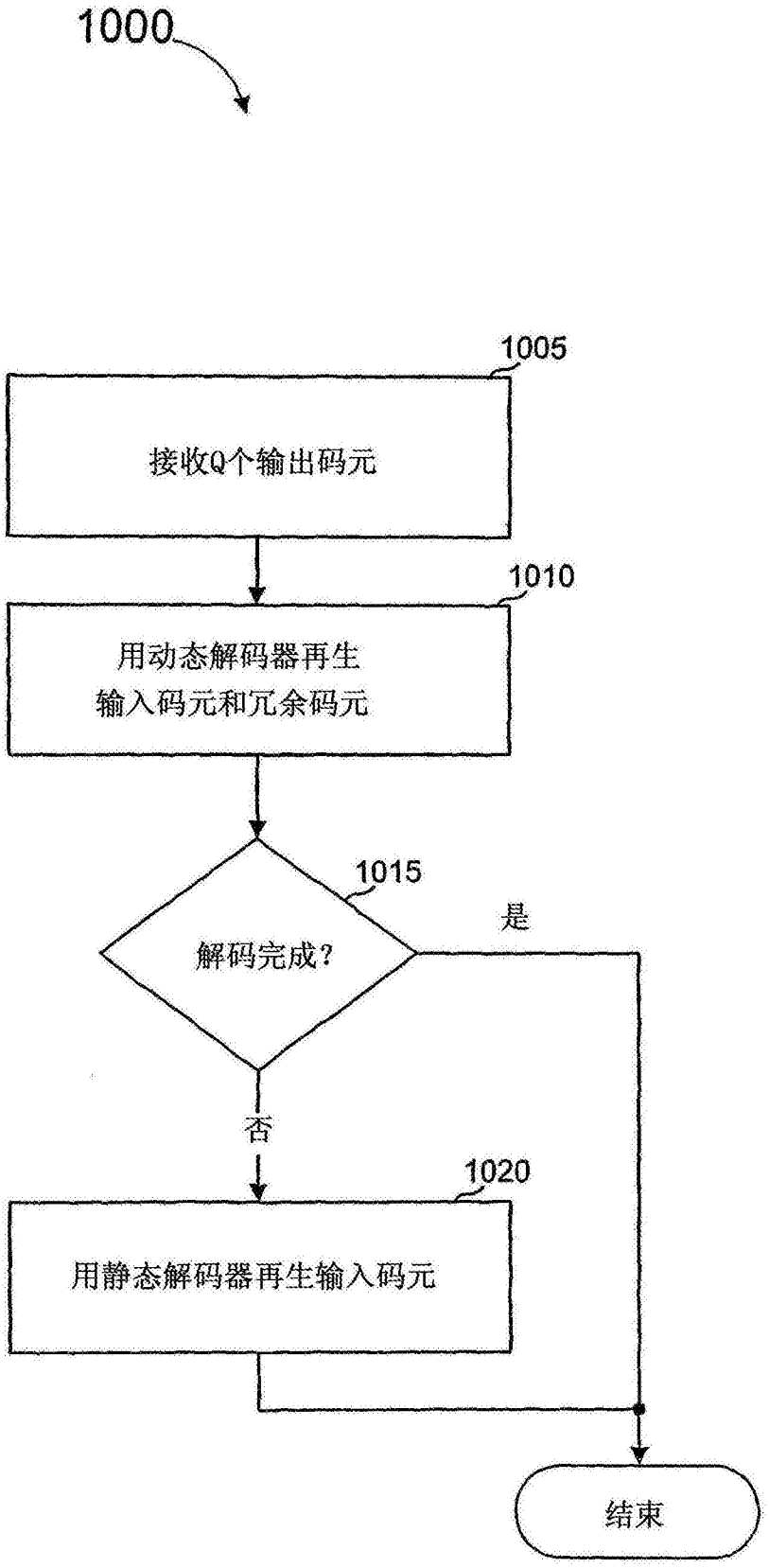

1000
1005
接收Q个输出码元
1010
用动态解码器再生
输入码元和冗余码元
1015
解码完成?
是
否
1020
用静态解码器再生输入码元
结束

图12

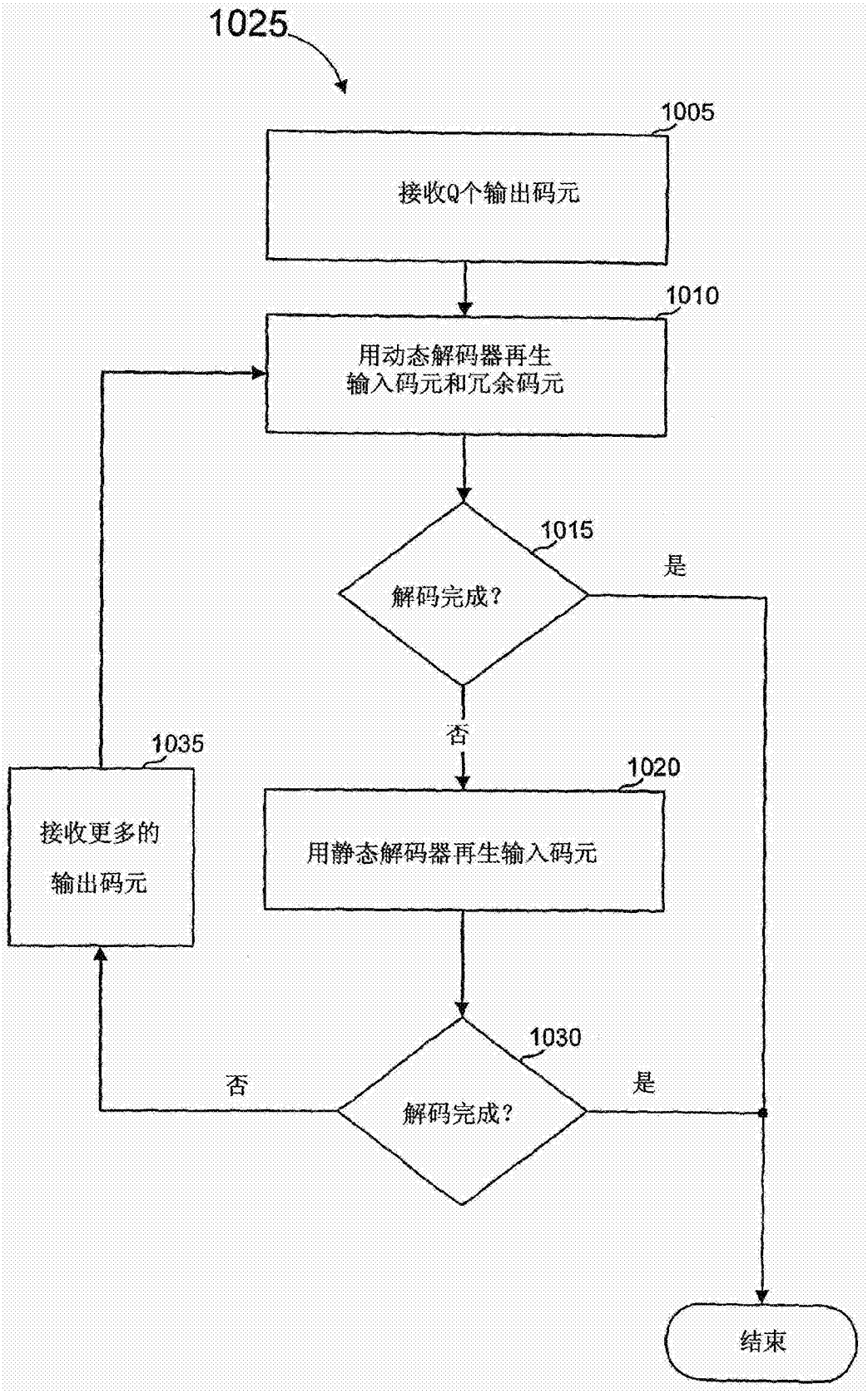

1025
1005
接收Q个输出码元
1010
用动态解码器再生
输入码元和冗余码元
1015
解码完成?
是
否
1035
接收更多的
输出码元
1020
用静态解码器再生输入码元
1030
否
解码完成?
是
结束

图13

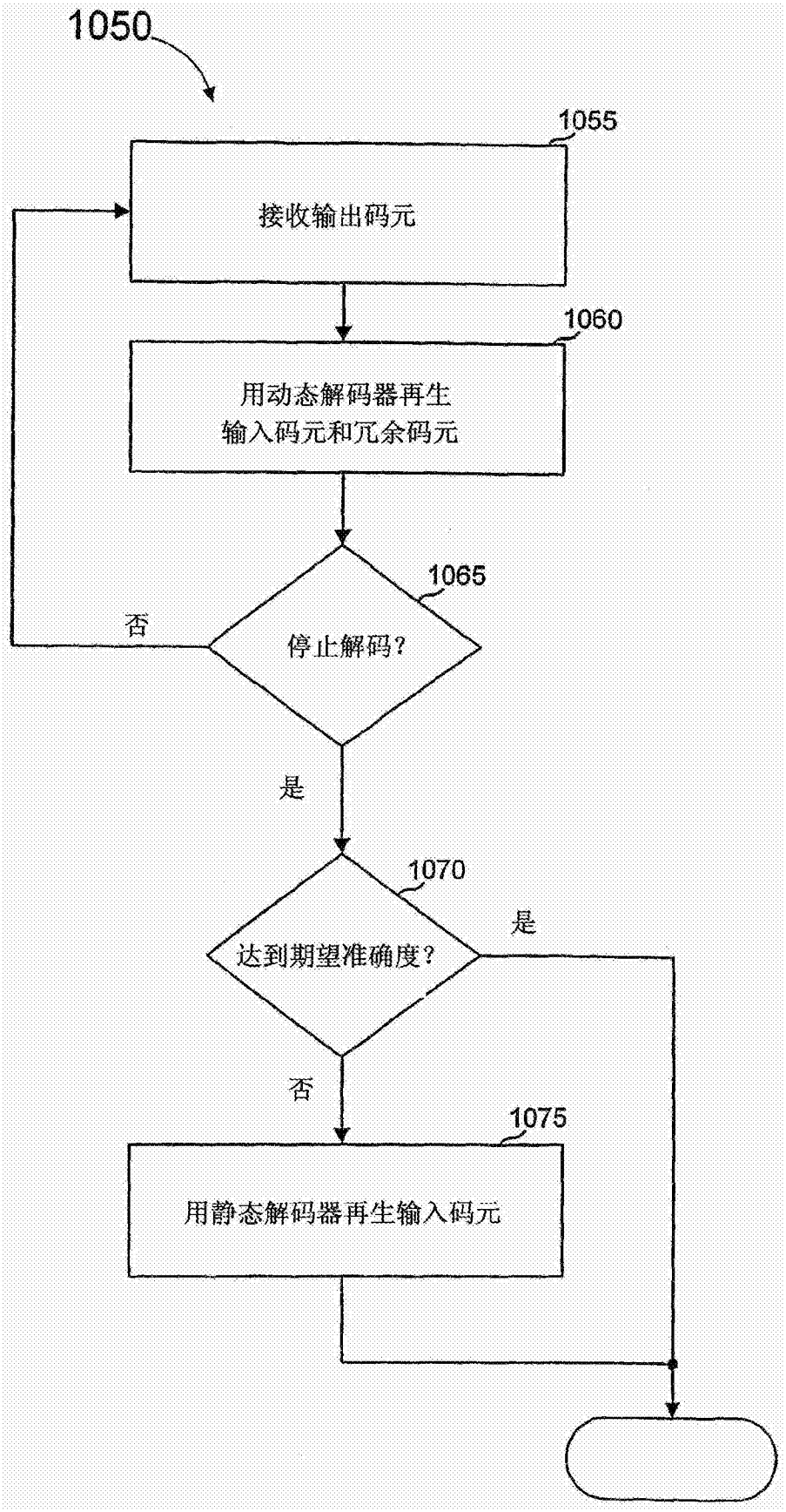

1050
1055
接收输出码元
1060
用动态解码器再生
输入码元和冗余码元
1065
停止解码？
否
是
1070
达到期望准确度？
是
否
1075
用静态解码器再生输入码元

图14

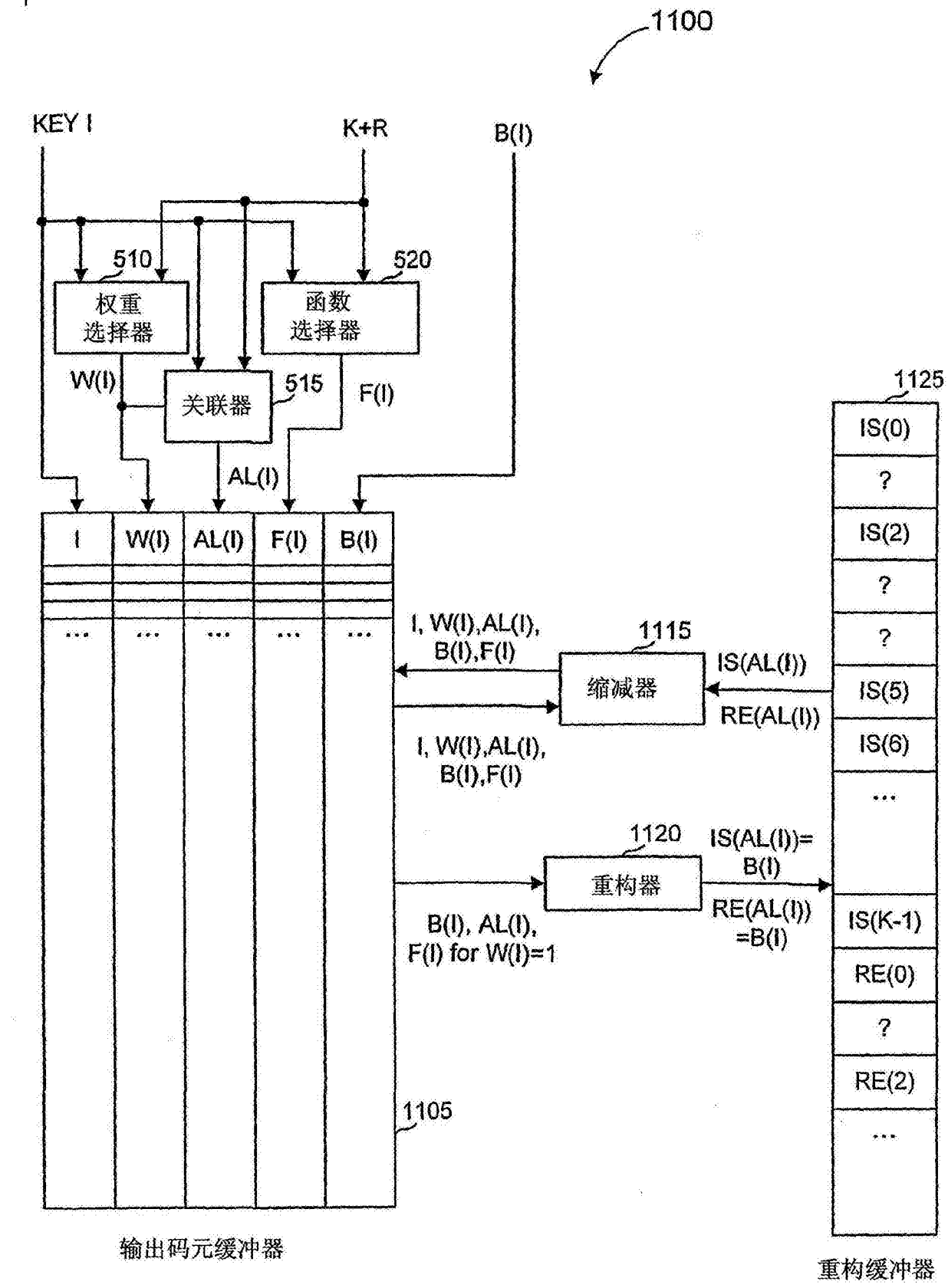

1100
KEY I
K+R
B(I)
510
520
权重选择器
函数选择器
W(I)
关联器
515
F(I)
AL(I)
I
W(I)
AL(I)
F(I)
B(I)
I, W(I),AL(I), B(I),F(I)
1115
缩减器
IS(AL(I))
RE(AL(I))
I, W(I),AL(I), B(I),F(I)
1120
重构器
IS(AL(I))= B(I)
RE(AL(I)) =B(I)
B(I), AL(I), F(I) for W(I)=1
1105
输出码元缓冲器
1125
IS(0)
?
IS(2)
?
?
IS(5)
IS(6)
IS(K-1)
RE(0)
?
RE(2)
重构缓冲器

图15

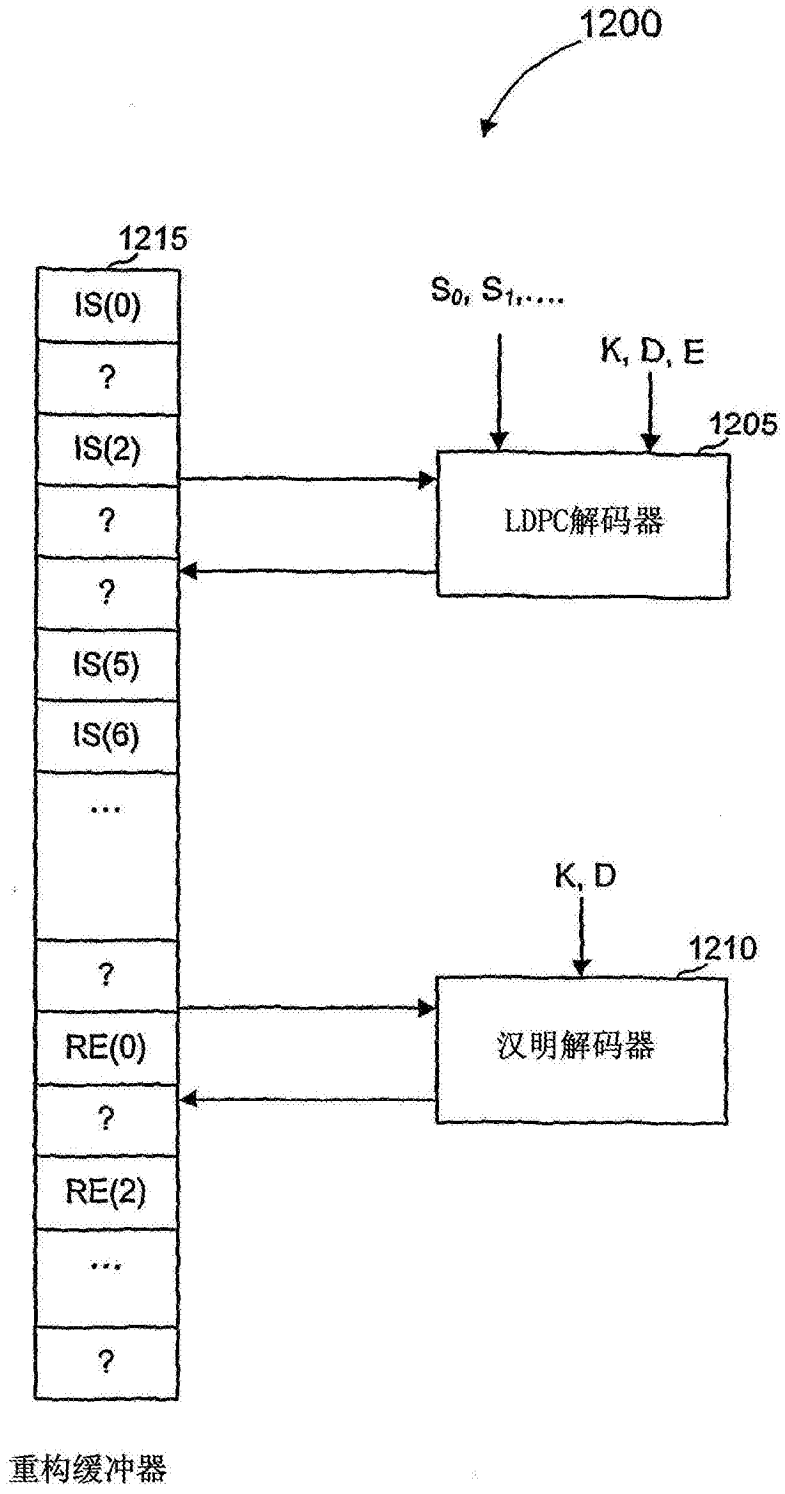

1200
1215
IS(0)
?
IS(2)
?
?
IS(5)
IS(6)
...
?
RE(0)
?
RE(2)
...
?
$S_0, S_1, \ldots$
K, D, E
1205
LDPC解码器
K, D
1210
汉明解码器
重构缓冲器

图16

0	1	2	...	...	...	...	K-1
K	K+1	K+2	...	...	...	...	2·K-1
2·K	2·K+1	2·K+2	...	...	...	...	3·K-1
...	...	...	...	...	...	...	
(N-1)·K	...	...	...	...	...	...	N·K-1

图17

文件大小F	G	码元大小T	G*T	K_t	源块 Z	子块 N	K_L	K_S	$T_L \cdot A$	$T_S \cdot A$
100 KB	6	84	504	1,220	1	1	1,220	1,220	N/A	N/A
100 KB	8	64	512	1,600	1	1	1,600	1,600	N/A	N/A
300 KB	2	256	512	1,200	1	2	1,200	1,200	128	128
1,000 KB	1	512	512	2,000	1	5	2,000	2,000	104	100
3,000 KB	1	512	512	6,000	1	12	6,000	6,000	44	40
10,000 KB	1	512	512	20,000	3	14	6,666	6,667	40	36

图18

最大源块大小B	G	码元大小T	G·T
40 KB	10	48	480
160 KB	4	128	512
640 KB	1	512	512

图19

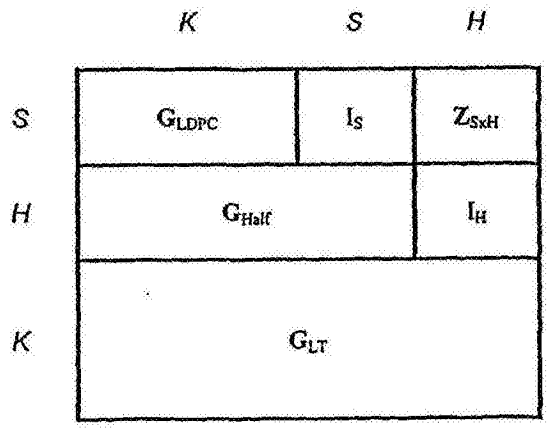


图20

索引 J	f[J]	d[J]
0	0	--
1	10241	1
2	491582	2
3	712794	3
4	831695	4
5	948446	10
6	1032189	11
7	1048576	40

图21

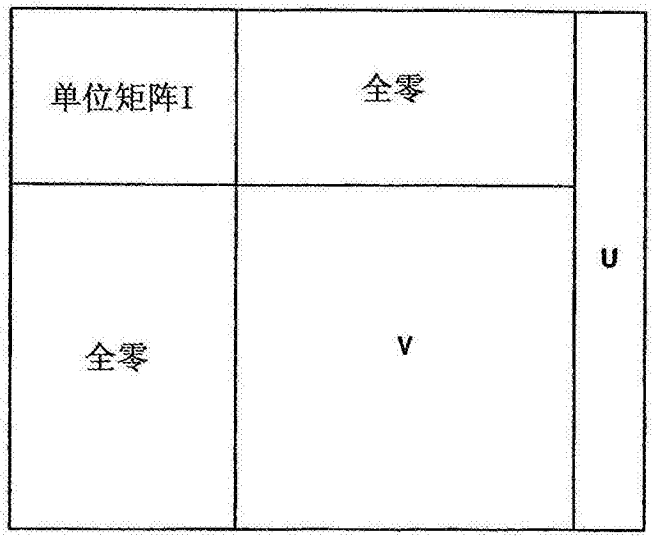


图22