



(12) 发明专利

(10) 授权公告号 CN 102077173 B

(45) 授权公告日 2015.06.24

(21) 申请号 200980124441.X

0022 段至 0081 段、图 2-8.

(22) 申请日 2009.04.21

US 2007/0234184 A1, 2007.10.04, 全文.

CN 101174839 A, 2008.05.07, 全文.

(85) PCT国际申请进入国家阶段日
2010.12.28

CN 101174838 A, 2008.05.07, 全文.

US 2005/0204264 A1, 2005.09.15, 全文.

(86) PCT国际申请的申请数据

PCT/US2009/041215 2009.04.21

审查员 杨莹莹

(87) PCT国际申请的公布数据

W02010/123493 EN 2010.10.28

(73) 专利权人 艾格瑞系统有限责任公司

地址 美国特拉华州

(72) 发明人 N·格拉菲 K·冈曼

(74) 专利代理机构 北京律盟知识产权代理有限
责任公司 11287

代理人 林斯凯

(51) Int. Cl.

G06F 11/00(2006.01)

(56) 对比文件

US 2008/0104485 A1, 2008.05.01, 说明书第

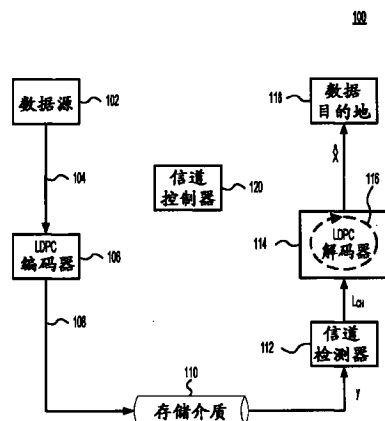
权利要求书2页 说明书12页 附图7页

(54) 发明名称

利用写入验证减轻代码的误码平层

(57) 摘要

在将信道输入(例如,LDPC)码字写入至存储介质时所执行的写入验证方法:(i)比较信道输入码字与写入码字,(ii)识别任何错误位,以及(iii)将针对记录的错误位索引存储在表中。在以后的某一时间,读取该写入码字并将其发送至解码器。如果该解码器无法利用附近码字,则写入错误恢复处理搜索该表并且检索错误位信息。调节在那些索引处的码字位,并且提交修正的码字以供进一步的处理。



1. 一种用于将原始编码码字存储至存储介质的机器实施的方法,该方法包括:
 - (a) 将所述原始编码码字写入至所述存储介质,作为写入编码码字;
 - (b) 通过从所述存储介质读取所述写入编码码字来生成信道输出码字;
 - (c) 将所述原始编码码字与基于所述信道输出码字的导出码字进行比较,以识别所述写入编码码字中的一个或多个错误位的第一集合;
 - (d) 通过在所述第一集合中选择一个或多个错误位来生成所述写入编码码字中的一个或多个错误位的第二集合;
 - (e) 将与所述第二集合中的一个或多个错误位相对应的错误位信息写入至所述存储介质;
 - (f) 通过从所述存储介质读取所述写入编码码字来生成解码器输入码字;
 - (g) 针对所述解码器输入码字执行解码以生成解码码字;以及
 - (h) 如果所述解码码字不是正确的解码码字,则:
 - (h1) 从所述存储介质读取所述错误位信息;
 - (h2) 基于所述错误位信息生成修正的码字;以及
 - (h3) 对所述修正的码字执行进一步的处理;其中,所述步骤 (h3) 包括:
 - (h3i) 确定所述错误位信息是否对应于所述第一集合中的所有错误位;
 - (h3ii) 如果所述错误位信息对应于所述第一集合中的所有错误位,则所述进一步的处理包括对所述修正的码字执行出错校验和循环冗余校验 (CRC) 中的一个或多个;以及
 - (h3iii) 如果所述错误位信息不对应于所述第一集合中的所有错误位,则所述进一步的处理包括对所述修正的码字执行解码。
2. 根据权利要求 1 所述的方法,其中,所述原始编码码字是 LDPC 码字。
3. 根据权利要求 1 所述的方法,其中,所述步骤 (c) 还包括针对所述信道输出码字执行解码来生成所述导出码字。
4. 根据权利要求 1 所述的方法,其中,所述第二集合等于所述第一集合。
5. 根据权利要求 1 所述的方法,其中,所述第二集合是所述第一集合的适当子集。
6. 根据权利要求 1 所述的方法,其中,所述步骤 (h2) 包括基于所述错误位信息来修正所述解码码字的一个或多个位以生成所述修正的码字。
7. 根据权利要求 1 所述的方法,其中,所述错误位信息包括:
 - 针对所述写入编码码字的特有标识符;
 - 针对所述第二集合中的每一个错误位的索引值;以及
 - 对所述第二集合是否对应于所述第一集合中的所有错误位的指示。
8. 根据权利要求 1 所述的方法,其中,所述步骤 (d) 包括:
 - (d1) 在所述第一集合中选择一个或多个错误位;
 - (d2) 基于所选择的一个或多个错误位生成修正的码字;
 - (d3) 对所述修正的码字执行解码,以生成候选解码码字;
 - (d4) 确定所述候选解码码字是否为正确的解码码字;
 - (d5) 如果所述候选解码码字是正确的解码码字,则基于所选择的一个或多个错误位来生成所述第二集合;以及

(d6) 如果所述候选解码码字不是正确的解码码字,则针对所述第一集合中的其它选择的一个或多个错误位重复步骤 (d1)-(d4)。

9. 一种用于针对存储在存储介质中的写入编码码字生成解码码字的机器实施的方法,所述写入编码码字具有一个或多个错误位,所述方法包括:

(a) 通过从所述存储介质读取所述写入编码码字来生成信道输出码字;

(b) 基于所述信道输出码字来生成导出码字;

(c) 从所述存储介质读取错误位信息,其中,所述错误位信息对应于所述写入编码码字中的一个或多个错误位;

(d) 基于所述错误位信息生成修正的码字;以及

(e) 针对所述修正的码字执行处理,以生成所述解码码字

其中,所述步骤 (e) 包括:

(e1) 确定所述错误位信息是否对应于所述写入编码码字中的所有错误位;

(e2) 如果所述错误位信息对应于所述写入编码码字中的所有错误位,则所述处理包括对所述修正的码字执行出错校验和循环冗余校验 (CRC) 中的一个或多个;以及

(e3) 如果所述错误位信息不对应于所述写入编码码字中的所有错误位,则所述处理包括针对所述修正的码字执行解码。

10. 根据权利要求 9 所述的方法,其中,所述写入编码码字是 LDPC 码字。

11. 根据权利要求 9 所述的方法,其中,所述步骤 (b) 包括:

(b1) 针对所述信道输出码字执行解码,以生成所述导出码字;以及

(b2) 确定所述导出码字不是正确的解码码字。

12. 根据权利要求 9 所述的方法,其中,所述步骤 (d) 包括基于所述错误位信息来修正所述导出码字中的一个或多个位,以生成所述修正的码字。

13. 根据权利要求 9 所述的方法,其中,所述错误位信息包括:

针对所述写入编码码字的特有标识符;

针对所述写入编码码字中的每一个错误位的索引值;以及

对所述错误位信息是否对应于所述写入编码码字中的所有错误位的指示。

利用写入验证减轻代码的误码平层

[0001] 对相关申请的交叉引用

[0002] 本申请的主题涉及：(1) 代理案卷 no. 08-0241 并于 2008 年 12 月 12 日提交的 PCT 申请 no. PCT/US08/86523 的主题，(2) 代理案卷 no. 08-1293 并于 2008 年 12 月 12 日提交的 PCT 申请 no. PCT/US08/86537 的主题，(3) 代理案卷 no. 08-0248 并于 2009 年 3 月 10 日提交的美国申请 no. 12/401116 的主题，(4) 代理案卷 no. 08-1057 并于 2009 年 4 月 2 日提交的 PCT 申请 no. PCT/US09/39279 的主题，(5) 代理案卷 no. 08-0242 并于 2009 年 4 月 8 日提交的 US 申请 no. 12/420535 的主题，(6) 2008 年 1 月 5 日提交的美国专利申请 no. 12/113729 的主题，(7) 2008 年 5 月 1 日提交的 US 专利申请 no. 12/113755 的主题，以及 (8) 代理案卷 no. 08-0243 并于 2009 年 4 月 8 日提交的 PCT 申请 no. PCT/US09/39918 的主题，其全部教导通过引用其全部内容而包含于此。

技术领域

[0003] 本发明涉及数字信号处理，并且更具体地说，涉及诸如低密度奇偶校验 (LDPC) 编码的数据编码方法。

背景技术

[0004] 通信是通过通信信道从发送器向接收器传送信息。在真实世界中，通信信道是有噪声的信道，向接收器提供从发送器发送来的失真形式的信息。存储装置（例如，硬盘 (HD) 驱动器、闪存驱动器）就是一种这样的有噪声信道，接受来自发送器的信息、存储该信息，并接着向接收器提供或多或少失真形式的该信息。

[0005] 由诸如存储装置的通信信道引入的失真可能大到足以导致信道错误，即，当信道输入信号为 0 时，接收器将信道输出信号解译为 1，或反之亦然。信道错误降低了吞吐量，并因此不合乎需要。因此，存在对于检测和 / 或校正信道错误的工具的持续需要。低密度奇偶校验 (LDPC) 编码是用于检测和校正信道错误的一种方法。

[0006] LDPC 码是可以针对低信噪比 (SNR) 应用实现非常低的位出错率 (BER) 的已知近 Shannon 极限码之一。LDPC 解码由于其并行化的潜力、实施复杂性低、解码等待时间少以及高 SNR 下误码平层 (error floor) 不太严重而著名。LDPC 码实质上考虑到了所有下一代通信标准。

发明内容

[0007] 在一个实施例中，本发明是一种用于将原始编码码字存储至存储介质的机器实施的方法。所述原始编码码字被写入至所述存储介质，作为写入编码码字。通过从所述存储介质读取所述写入编码码字来生成信道输出码字。将所述原始编码码字与基于所述信道输出码字的导出码字进行比较，以识别所述写入编码码字中的一个或多个错误位的第一集合。通过在所述第一集合中选择一个或多个错误位来生成所述写入编码码字中的一个或多个错误位的第二集合。将与所述第二集合中的一个或多个错误位相对应的错误位信息写入

至所述存储介质。

[0008] 在另一实施例中,本发明是一种用于针对存储在存储介质中的写入编码码字生成解码码字的机器实施方法,所述写入编码码字具有一个或多个错误位。通过从所述存储介质读取所述写入编码码字来生成信道输出码字。从所述信道输出码字生成导出码字。从所述存储介质读取错误位信息,其中,所述错误位信息对应于所述写入编码码字中的一个或多个错误位。使用所述错误位信息来生成修正的码字。对所述修正的码字执行进一步的处理,以生成所述解码码字。

附图说明

[0009] 根据下面的详细描述、附属权利要求书以及附图,本发明的其它方面、特征以及优点将更充分清楚,在附图中,相同标号标识相似或相同部件。

[0010] 图 1 是利用 LDPC 编码的通信系统 100 的框图。

[0011] 图 2(A) 描绘了 LDPC H 矩阵 200, 而图 2(B) 是 H 矩阵 200 的 Tanner 图形。

[0012] 图 3 是图 A 的解码器 AC 所使用的 LDPC 解码方法 300 的流程图。

[0013] 图 4 是根据本发明一实施例的通过信道控制器 120 控制的、图 1 的通信系统 100 所实施的 LDPC 编码 / 解码处理 400 的流程图。

[0014] 图 5 是错误位表的一个实施例的图。

[0015] 图 6 是根据本发明一个实施例的、图 4 的步骤 410 的流程图,即,写入错误验证处理的流程图。

[0016] 图 7 是图 6 的步骤 612 的流程图。

[0017] 图 8 是根据本发明一个实施例的图 4 的写入错误恢复处理 418 的流程图。

具体实施方式

[0018] 图 1 是利用 LDPC 编码的通信系统 100 的框图。数据源 102 生成称为原始信息字 104 的多个位的集合。LDPC 编码器 106 编码原始信息字 104, 以生成原始编码码字 108。下面,更详细讨论 LDPC 编码。将原始编码码字 108(还称为信道输入码字)写入至存储介质 110(例如, 闪存驱动器、硬盘驱动器盘片等), 作为写入编码码字。

[0019] 在以后的某一时间, 存储介质 110 读取写入编码码字并向信道检测器 112 输出一组值 y (即信道输出码字)。信道输出码字和根据该信道输出码字导出的任何码字称为导出码字。信道检测器 112 将接收到的值 y 转换成一组对数似然比 (LLR) 值 L_{ch} 。LLR 值包括: (i) 表示解码器关于由对应值 y 所表示的一位硬判定值的最佳推测的符号位, 和 (ii) 表示硬判定中解码器的置信度的一个或多个量值位 (magnitude bit)。例如, 信道检测器 112 可能输出每一个 LLR 值 L_{ch} , 作为五位值, 其中, 最高有效位是表示硬判定的符号位, 而四个量值位的值表示硬判定的置信度。因而, 在一种可能 LLR 方案中, 二进制 00000 的 LLR 值表示具有最小置信度的 0 的硬判定, 二进制 01111 的 LLR 值表示具有最大置信度的 0 的硬判定, 二进制 10001 的 LLR 值表示具有最小置信度的 1 的硬判定, 而二进制 11111 的 LLR 值将表示具有最大置信度的 1 的硬判定, 其中, 二进制 10000 未使用。

[0020] 信道检测器 112 向 LDPC 解码器 114 发送 L_{ch} 值, 其中, 它们变为解码器输入码字。接着, LDPC 解码器 114 针对一组 L_{ch} 值执行一次或多次解码迭代 116(“局部迭代”), 以生

成解码码字 \hat{x} 。当 (i) LDPC 解码器 114 得出正确的解码码字 (DCCW), 即 \hat{x} 和信道输入码字 108 相同, 或者 (ii) LDPC 解码器 AC 执行了最大可允许数量的局部迭代而没有得出 DCCW, 即 LDPC 解码器 114 失败时, LDPC 解码器 AC 终止。当解码器 114 终止时, 其向数据目的地 118 输出解码码字 \hat{x} 。下面更详细地描述 LDPC 解码。

[0021] 信道控制器 120 至少控制 LDPC 编码器 106、信道检测器 112, 以及 LDPC 解码器 114 的操作。信道控制器典型为 ARM(高级 RISC(精简指令集代码)机) 处理器。

[0022] LDPC 编码

[0023] 为了创建码字 108, LDPC 编码器 106 向信息字 104 的多个位附加由 LDPC 代码指定的多个奇偶位。信息字 104 中的位数由 K 表示。编码码字中的位称为变量位, 并且那些变量位的数量由 N 表示。因而, 奇偶位的数量由 N-K 给出。

[0024] 按由特定 LDPC 代码所指定的特定方式, LDPC 码字中的每一个奇偶位与该码字中的一个或多个其它位相关联, 并且设置指配给奇偶位的值, 以满足 LDPC 代码。典型 LDPC 代码指定奇偶位, 并且其关联位满足奇偶检验约束, 例如, 位的总和是偶数, 即, 和模 $2 = 0$ 。

[0025] LDPC 代码

[0026] 特定的 LDPC 代码由称为奇偶校验矩阵或 H 矩阵 (或简称为 H) 的 1 和 0 的二维矩阵来定义。H 被 LDPC 编码器和解码器推理而知。H 包括 N 列和 N-K 行, 即, 针对码字的每一列, 和针对每一个奇偶位的行。H 中的每一个 1 表示列的码字位与行的奇偶位之间的关联。例如, H 的第三行、第七列处的 1 意指第三奇偶校验位与码字的第七位相关联。检验位与和该检验位相关联的所有变量位的值的和模 2 应当为 0。典型 LDPC 代码的定义特征是, H 为“稀疏的”, 即, H 的元素大多为 0, 而 1 相对较少。

[0027] 图 2A 描绘了 LDPC H 矩阵 200。H 矩阵 200 包括 $N = 9$ 列和 $N-K = 6$ 行。因而, H 矩阵 200 定义了接受三位信息字的 LDPC 代码, 附加六个奇偶位, 并且输出九位码字。在存储介质是硬盘驱动器或闪速驱动器的一种实现方式中, 每一个信息字的长度为 4096 位, 并且每一个码字的长度为 4552 位。其它实现方式可以包括具有其它位长度的信息字和 / 或码字。

[0028] LDPC 解码: 置信传播 (Belief Propagation)

[0029] 图 3 是图 1 的解码器 114 所使用的 LDPC 解码方法 300 的流程图。解码方法 300 的核心是称作置信传播的迭代、两相消息传递算法。置信传播可以使用 Tanner 图形来说明。

[0030] 图 2(B) 是 H 矩阵 200 的 Tanner 图形。一般来说, Tanner 图形包括: 1) 等于 H 中的列数 (并因此等于变量位数 N) 的多个位节点 (还称为变量节点) n , 2) 等于 H 中的行数 (并因此等于奇偶位数) 的多个校验节点 m , 3) 边 202, 每一条边将单个位节点 n_i 连接至单个校验节点 m_j , 4) 针对每一个位节点 n_i 的原始 L_{ch} 值, 以及 5) 针对每一个位节点 n_i 的所计算的硬判定输出值 \hat{x}_n 。图 2(B) 的 Tanner 图形包括: 九个位节点 n_0-n_8 、六个校验节点 m_0-m_5 、将位节点连接至校验节点的 18 条边 202、九个 L_{ch} 值, 以及九个 \hat{x}_n 值。

[0031] Tanner 图形中的边表示位节点 n 与校验节点 m 之间的关系, 其中, 边表示 H 中的 1。例如, 在图 2(B) 中, 边 202 将第一位节点 n_0 连接至第四校验节点 m_3 , 因为在图 2(A) 中的 H 矩阵 200 的第一列、第四行中存在 1。

[0032] Tanner 图形是双向图形, 即, 一条边可以将一个位节点仅连接至一个校验节点, 而

不能将一个位节点连接至另一位节点,或者不能将一个校验节点连接至另一校验节点。通过边连接至特定校验节点 m 的所有位节点 n 的集合表示为 $N(m)$ 。通过边连接至特定位节点 n 的所有校验节点 m 的集合表示为 $M(n)$ 。特定(位或校验)节点的索引在该图形中是其序数系列。

[0033] 返回至图 3,在步骤 302 开始处理并且前进至步骤 304,解码器初始化。解码器初始化 304 包括针对与位节点 n 相关联的对应 L_{ch} 值来设置连接至每一个位节点 n 的所有边 n (例如,图 2(B) 的边 202),并且针对位节点 n 的 L_{ch} 的硬判定值(即,MSB)的来设置位节点 n 的 \hat{x}_n 值。因而,例如,在图 2(B) 中,如果与位节点 n_0 相关联的 L_{ch} 值是十进制值 +5,则在步骤 304,将位节点 n_0 连接至校验节点 m_0 和 m_3 的两条边 202 被设置成 +5,并且位节点 n_0 的 \hat{x}_n 值被设置成 1。表达这个步骤的第一部分的另选方式是,位节点 n_0 向集合 $M(n_0)$ 中的每一个校验节点 m 发送消息 +5。从位节点 n 向校验节点 M 发送的消息被称作位节点或 Q 消息,并且被表示为 Q_{nm} 。

[0034] 接着,步骤 304 向出错校验步骤(syndrome check step)306 发送包括 N 个 \hat{x}_n 值的候选解码码字矢量 \hat{x} 。出错校验步骤 306 利用以下等式(1)来计算出错矢量(syndrome vector) z :

$$[0035] \quad z = \hat{x}H^T \quad (1)$$

[0036] 其中, H^T 是 H 矩阵的转置。如果出错矢量 z 是 0 矢量,则矢量 \hat{x} 满足由 H 定义的所有奇偶检验约束,即, \hat{x} 是有效解码码字。在这种情况下,处理前进至循环冗余校验(CRC)校验 318。

[0037] 相反,如果出错矢量 z 不是 0 矢量,则矢量 \hat{x} 无法满足所述奇偶检验约束中的一个或多个。出错矢量 z 中的每一个非零元素表示一无法满足的奇偶检验约束,其还称为未满足校验节点(USC)。出错矢量 z 中的非零元素的数量是矢量 \hat{x} 中的 USC 的数量 b 。此外,出错矢量 z 中的非零元素的索引是矢量 \hat{x} 中的 USC 的索引。

[0038] 如果矢量 \hat{x} 无法通过出错校验 306,则处理继续一个或多个解码迭代 308 中的第一个迭代(称作“局部迭代”)。解码迭代 308 包括三个步骤:1) 置信传播校验节点更新步骤 310,2) 置信传播位节点更新步骤 312,以及 3) 出错校验步骤 314,其与步骤 306 相同。

[0039] 在置信传播校验节点更新步骤 310 中,每一个校验节点 m 使用从集合 $N(m)$ 中的所有位节点 n 接收到的 Q_{nm} 消息,以根据以下的等式(2)、(3)以及(4)来计算表示为 R_{mn} 的一个或多个校验节点或 R 消息:

$$[0040] \quad R_{mn}^{(i)} = \delta_{mn}^{(i)} \max(\kappa_{mn}^{(i)} - \beta, 0) \quad (2)$$

$$[0041] \quad \kappa_{mn}^{(i)} = \left| R_{mn}^{(i)} \right| = \min_{n' \in N(m) \setminus n} \left| Q_{n'm}^{(i-1)} \right| \quad (3)$$

$$[0042] \quad \delta_{mn}^{(i)} = \left(\prod_{n' \in N(m) \setminus n} \text{sgn}(Q_{n'm}^{(i-1)}) \right) \quad (4)$$

[0043] 其中, i 是解码迭代, $N(m) \setminus n$ 是不包括位节点 n 的集合 $N(m)$, 函数 sgn 返回其运算数的符号,而 β 是正常数,其值取决于码参数。每一个校验节点 m 都将计算出的 R_{mn} 消息连

同那些相同的边发送回集合 $N(m)$ 中的所有位节点 n 。

[0044] 接下来,在置信传播位节点更新步骤 312 中,每一个位节点 n 根据以下等式 (5) 来计算一个或多个 Q_{nm} 消息:

$$[0045] \quad Q_{nm}^{(i)} = L_n^{(0)} + \sum_{m' \in M(n) \setminus m} R_{m'n}^{(i)} \quad (5)$$

[0046] 其中, L_n^0 是位节点 n 的原始 L_{ch} 值,而 $M(n) \setminus m$ 是不包括校验节点 m 的集合 $M(n)$ 。接着,每一个位节点 n 将计算出的 Q_{nm} 消息发送至集合 $M(n)$ 中的所有校验节点 m 。

[0047] 此外,在位节点更新步骤 312 期间,每一个位节点 n 根据以下等式 (6) 和 (7) 来更新其 \hat{x}_n 值:

$$[0048] \quad E_n^{(i)} = \sum_{m' \in M(n)} R_{m'n}^{(i)} \quad (6)$$

$$[0049] \quad P_n = L_n^0 + E_n^{(i)} \quad (7)$$

[0050] 如果 $P_n \geq 0$, 则 $\hat{x}_n = \mathbf{0}$, 而如果 $P_n < 0$, 则 $\hat{x}_n = \mathbf{1}$ 。由等式 (6) 生成的值还称为非固有或 E 值, 并且表示为 E_{LDPC} 。根据等式 (7) 生成的值称为 P 值。由等式 (2)-(7) 表示的特定置信传播算法称为最小和算法。应注意, \hat{x}_n 值在每一次解码迭代 308 期间更新并且最后通过解码处理 300 输出。原始 LLR 值 L_{ch} 在解码处理 300 期间保持不变。

[0051] 位节点更新步骤 312 向出错校验步骤 314 发送由解码器的当前 \hat{x}_n 值之外的其它值构造的矢量 \hat{x} 。步骤 314 的出错校验与上述步骤 306 的出错校验相同。如果矢量 \hat{x} 通过出错校验 314, 则将矢量 \hat{x} 发送至 CRC 步骤 318。

[0052] LDPC 解码: 循环冗余校验和不满足校验节点

[0053] 通过出错校验 306 或 314 意指矢量 \hat{x} 是有效解码码字, 但不一定是正确的解码码字 (DCCW)。对于 LDPC 解码器来说, 可能生成不是 DCCW 的有效解码码字。在这种情况下, 矢量 \hat{x} 中不存在 USC, 但存在不满足校验节点 (MSC)。不满足校验节点是与偶数数量的错误位节点相关联的校验节点。

[0054] 因而, 为了确保有效矢量 \hat{x} 是 DCCW, 处理 300 向循环冗余校验 (CRC) 318 传递矢量 \hat{x} 。CRC 校验是可以检测传送或存储期间数据的变化了的校验和运算。具体来说, 编码器计算要发送的码字的第一 CRC 校验和, 并将第一 CRC 校验和和该码字都发送至解码器。解码器在接收到该码字和第一 CRC 校验和时, 利用接收到的码字来计算第二 CRC 校验和, 并且比较其与第一 CRC 校验和。如果两个 CRC 校验和不匹配, 则存在接收到的码字中存在错误的概率高。

[0055] 如果矢量 \hat{x} 通过 CRC 检验, 则矢量 \hat{x} 是 DCCW, 并且处理 300 将全局变量 DCCW 设置成真, 输出矢量 \hat{x} , 并且在步骤 320 终止。否则, 矢量 \hat{x} 不是 DCCW, 并且处理 300 将全局变量 DCCW 设置成假, 输出矢量 \hat{x} , 并且在步骤 320 终止。全局变量 DCCW 向其它解码处理通知是否已经生成 DCCW。

[0056] 返回至步骤 314, 如果矢量 \hat{x} 无法通过出错校验, 则矢量 \hat{x} 中存在一个或多个 USC。用于解决 USC 的典型方法是执行另一解码迭代 308。然而, 在特定解码会话 (decoding

session) 中,在合理的时间量中可能存在从未满足的一个或多个 USC(下面,参见俘获集合(trapping set)的讨论)。因而,LDPC 解码器典型地受限于它们可以执行多少解码迭代。最大迭代次数的典型值的范围为 50 到 200。

[0057] 在图 3 中,步骤 316 确定是否已经达到指定的最大迭代次数。如果没有达到,则执行另一解码迭代 308。相反,如果已经达到最大迭代次数,则解码器处理 300 失败。在这种情况下,处理 300 将全局变量 DCCW 设置成假,输出矢量 \hat{x} ,并且在步骤 320 终止。

[0058] 处理 300 的完整执行(具有一次或更多次局部解码迭代 308)称为解码会话。

[0059] BER、SNR,以及误码平层(error floor)

[0060] LDPC 解码器的位出错率(BER)表示解码位具有错误值的概率。因而,例如,具有 10^{-9} 的 BER 的解码器平均起来对于每百万解码位将生成一个错误位。LDPC 解码会话无法收敛到 DCCW 促成导致了(Contribute to)解码器的 BER。

[0061] LDPC 解码器的 BER 受解码器输入信号的信噪比(SNR)的强烈影响。作为 SNR 的函数的 BER 的图形典型地包括两个不同的区域:其中在 SNR 单位增加的情况下 BER 快速改进(降低)的初始“瀑布”区,和其中 SNR 的增加仅产生 BER 的适度改进的后续“误码平层”区。因而,实现误码平层区域中的 BER 的显著改进需要除了 SNR 增加以外的其它方法。

[0062] 用于改进 LDPC 解码的误码平层特性的一种方法是增加码字长度。然而,增加码字长度还增加了 LDPC 解码所需的存储器和其它计算资源。因而,如果这种资源严格受限,就像存储装置上的读取信道装置的典型情况一样,则必须寻找其它方法来产生所需的误码平层改进。

[0063] 另一不充足的资源是处理周期。典型地讲,为了实现指定的吞吐量,存储装置预算了固定数量的读取信道处理周期来解码码字。超出该预算的方法(例如,非传输过程中处理(off-the-fly)方法)降低了吞吐量。更合乎需要的是传输过程中处理(on-the-fly)方法,其在时钟周期分配之内恢复 DCCW,并且由此不减少吞吐量。

[0064] 用于改进 LDPC 解码器的误码平层特性的另一方法是使用一种或多种后处理方法。后处理方法在解码会话在允许的最大数量迭代内无法收敛到 DCCW 时被调用。后处理方法调节与解码处理相关联的一个或多个变量(例如, γ 值、 L_{cn} 值以及 / 或者解码器的运算参数),并且重新开始解码。像解码本身一样,对于解码处理的输入来说,后处理方法通常是迭代的、多重的、顺序的变化。

[0065] 在典型 LDPC 解码会话中,解码器在最初几次解码迭代中收敛到 DCCW。相反,当 LDPC 解码器在指定的最大数量的迭代中无法收敛到 DCCW 时,其已知为失败的解码器,并且由失败的解码器生成的解码码字是失败的解码码字。

[0066] 失败的码字典典型地根据它们所包含的 USC 节点数来分类。无效解码码字(ICW)是具有大量 USC 节点的失败码字(例如,对于大约 5000 位的码字来说,USC 节点数大于 16)。ICW 典型地起因于解码器输入码字包含了解码器不能够校正所有位错误的很多位错误(即,很少正确值)。用于处理 ICW 的典型后处理方法是请求重新发送输入码字。尽管重新发送是非传输过程中处理方法,并因此是不希望的,但其通常对于校正 ICW 来说是唯一可靠选择。

[0067] 附近码字(NCW)是失败码字,其具有少量 USC(例如,对于大约 5000 位的码字来说,USC 数量为 16 或更少)。有时,NCW 中的 USC 形成称为俘获集合的稳定构造,对于其来

说,进一步的解码迭代不生成 DCCW。俘获集合对 LDPC 解码器的误码平层特性具有显著影响。

[0068] 俘获集合的标志为 (a, b), 其中, b 是俘获集合中的 USC 数, 而 a 是与那些 USC 相关联的错误位节点的数量。因而, (8, 2) 俘获集合包括两个 USC 和与那两个 USC 相关联的八个错误位节点 (EBN)。多数俘获集合包括少于五个的 USC 和少于 10 个的 EBN。

[0069] 翻转位节点指用于改变与位节点相关联的一个或多个值的特定处理。在翻转期间改变哪些值取决于 LDPC 解码器的状态。在一种可能的实现方式中, 如果 LDPC 解码器刚被初始化, 则翻转位节点包括: (i) 反转该位节点的 L_{ch} 值的硬判定值, 即, 1 变为 0, 或反之亦然, (ii) 将同一 L_{ch} 值的量值位 (即, 置信度) 设置成最大, 以及 (iii) 将所有其它 L_{ch} 值的量值位限制成最大可允许量值的值的至多 15%。

[0070] 例如, 假定具有 4 位 L_{ch} 量值的值的系统, 其中, 最大可允许正量值为 +15, 而最大可允许负量值为 -16, 并且其中, 最大可允许值的 15% 分别为 +2 和 -2。进一步假定对应于四个位节点的四个 L_{ch} 值: +2、-11、+1、+13。在这个示例中, 翻转第一个位节点包括: (i) 反转第一个位节点的 L_{ch} 值的符号, 即, +2 变为 -2, (ii) 将第一个位节点的 L_{ch} 值的量值设置成最大可允许值, 即, -2 变为 -16, 以及 (iii) 将其它三个位节点的 L_{ch} 值的量值设置成最大可允许值的至多 15%, 即, -11、+1 以及 +13 分别变为 -2、+1 以及 +2。将 L_{ch} 量值设置成较低值使得迭代解码处理能够更快速地收敛到 DCCW, 并且降低解码器失败的机会。

[0071] 根据同一实现方式, 如果解码器处于除了初始化以外的其它某一状态下, 则翻转位节点包括: (i) 确定位节点的 P 值的硬判定值 (通过上述等式 7 定义), (ii) 将该位节点的 L_{ch} 值、P 值以及所有关联 Q_{mm} 消息的硬判定值设置成 P 值硬判定值的相反数, (iii) 将该位节点的 L_{ch} 值、P 值以及所有关联 Q_{mm} 消息的量值位设置成最大, 以及 (iv) 将所有其它位的 L_{ch} 、P 以及 Q_{mm} 消息值的量值限制成最大可允许值的 15%。应注意到, 仅限制了初始量值。随着解码会话的前进, P 和 Q_{mm} 消息值更新并且可以假定任何可允许值。另一方面, L_{ch} 值是只读的, 并由此在该解码会话的持续时间内保持它们的受限量值的值。

[0072] 擦除是用于改变位节点值的另一特定处理。擦除位节点包括: (i) 将该位节点的 L_{ch} 值的硬判定值设置成 0, 以及 (ii) 将同一 L_{ch} 值的量值位 (即, 置信度) 设置成 0, 即, 无置信度。

[0073] 如果调节与俘获集合相对应的附近码字的多个 EBN 中的一个或多个 (例如, 翻转或擦除), 则对于所得修正码字重新执行 LDPC 解码可能收敛到 DCCW。如果成功, 则将这个处理称为打破俘获集合。因而, 用于改进 LDPC 解码器的误码平层特性的另一方法是采用失败解码器的附近码字 (NCW), 识别该 NCW 中的可能的 EBN, 翻转或擦除那些 EBN 中的一个或多个, 以及提交修正的 NCW 以供进一步 LDPC 处理。

[0074] 某些俘获集合可以通过翻转或擦除单一 EBN 来打破。在其它俘获集合中, 翻转或擦除单一 EBN 可以减少 USC 的数量, 但不完全打破俘获集合, 产生可以接着通过翻转或擦除另一 EBN 来打破的第二个、不同的俘获集合。其它俘获集合可以仅通过同时翻转或擦除两个或更多个 EBN 来打破。

[0075] 俘获集合基于以下情况而广泛地改变: (i) 解码器的操作条件 (例如, 解码器字母表、解码器算法、解码器校验节点更新方法), (ii) 写入错误, 以及 (iii) 读取错误。当通信信道是存储装置时, 写入错误是将码字的位不正确地写入至存储装置的情况, 即, 所写入的

编码码字中的位的值不匹配对应信道输入码字中的对应位的值。读取错误是从存储装置不正确地读取码字的正确写入位的情况,即,信道输出码字中的位的值不匹配所写入的编码码字中的对应位的值。例如通过多次重新读取扇区,接着平均化多次读取的样本而可以从读取错误恢复。然而,不能通过重新读取扇区而从写入错误恢复。

[0076] 本发明的实施方式是用于估计并校正 LDPC 解码器输入码字中的、因写入错误而造成的那些 EBN 的方法。该方法典型地包括一对处理:写入错误验证处理和写入错误恢复处理。写入错误验证处理在将码字写入至存储介质时执行。当从存储介质读取码字并解码时,在之后的时间执行写入错误恢复处理。

[0077] 图 4 是根据本发明一实施例的、通过信道控制器 120 控制的、图 1 的通信系统 100 所实施的 LDPC 编码/解码处理 400 的流程图。在步骤 402 开始处理,并且继续至步骤 404,其中,数据源 102 向 LDPC 编码器 106 提供原始信息字 104。接下来,在步骤 406,LDPC 编码器 106 编码原始信息字 104,以生成信道输入码字 108。接下来,在步骤 408,将信道输入码字 108 写入至存储介质 110(例如,闪速驱动器)。接下来,在步骤 410,执行写入错误验证处理,下面,在图 6 和 7 的讨论中,对该处理进行更详细的描述。

[0078] 在以后的某一时间,在步骤 412,从存储介质 110 读取所存储的码字,通过信道检测器 112 检测,并且将对应解码器输入码字的 L_{ch} 值发送至 LDPC 解码器 114。接下来,在步骤 414,LDPC 解码器 114 处理解码器输入码字以生成解码码字。该处理包括 LDPC 解码和各种读取事件方法,如误码平层减轻方法、介质缺陷检测方法以及标准信道重试方法。步骤 414 的处理的净效果是,校正了因读取错误而造成的所有 EBN,而保留在解码码字中的任何 ENB 是写入错误。

[0079] 接下来,在步骤 416,确定解码码字是否具有任何错误位。如果没有,则该解码码字是 DCCW,并且处理在步骤 420 结束。相反,如果该解码码字具有一个或多个错误位,则在步骤 418,针对该解码码字执行写入错误恢复处理,下面,在图 8 的讨论中,对该处理进行更详细的描述。

[0080] 写入错误验证处理 410 在将信道输入码字初始地写入至存储介质(例如,闪速驱动器)作为写入编码码字时执行。写入错误验证处理读取回该写入编码码字,并且生成对应的解码器输入码字。接着,该处理针对该解码器输入码字执行 LDPC 解码,以生成解码码字。接着,该处理比较信道输入码字与解码码字,并且确定任何错误位的数量和索引。如果错误位的数量大于 0 并且小于指定阈值,则该处理将错误位信息写入至称作错误位表的数据结构。在本发明的一个实施例中,该错误位信息包括码字标识符(例如,闪速驱动器页面 ID)和一个或多个错误位的索引。错误位的索引识别解码码字中的错误位的位置。错误位表典型地存储在存储介质上。

[0081] 图 5 是错误位表的一个实施例的图。表 500 包括三列(字段)和任何数量的行(记录)。列 CODEWORD_IDENTIFIER 包含码字标识符(例如,闪速驱动器页面 ID)。列 EB_INDICES 包含由写入错误验证处理识别的一个或多个错误位的索引。列 COMPLETE_SET 是指列 EB_INDICES 中识别的错误位索引是表示由写入错误验证处理所检测到的所有错误位索引(即,真或 1)还是适当子集(即,假或 0)的一位真/假字段。

[0082] 图 6 是根据本发明一个实施例的图 4 的步骤 410 的流程图,即,写入错误验证处理的流程图。在步骤 602 开始处理,并且前进至步骤 604,其中,从存储介质读取写入编码码

字,并且生成解码器输入码字。

[0083] 接下来,在步骤 606,对解码器输入码字执行 LDPC 解码,以及各种读取事件方法(例如,误码平层减轻方法、介质缺陷检测方法、标准信道重试方法),并且生成解码码字。步骤 606 的处理的净效果是,校正了因读取错误而造成的任何以及所有错误位,保留在解码码字中的任何错误位是由于写入错误所导致的。接下来,在步骤 608,以位的方式来比较解码码字与信道输入码字,并且确定任何错误位(对应于位失配)的数量和索引。

[0084] 在步骤 610,如果错误位的数量为 0(即,在步骤 606,解码器收敛到 DCCW),则写入验证处理 410 在步骤 616 终止。相反,如果错误位的数量超出指定阈值(例如,32),则在步骤 614,产生针对图 1 的信道控制器 120 的控制,其可以执行其自身的方法,以校正错误。例如,信道控制器 120 可以进行判定以将存储介质(例如,闪速驱动器页面)上的当前位置标记为坏的,并且尝试将信道输入码字 108 写入至另一位置。信道控制器方法往往具体到特定信道控制器和信道控制器制造方。相反,如果错误位的数量大于 0 并且小于或等于指定阈值,则在步骤 612,将一个或多个错误位索引写入至错误位表。

[0085] 表的限制 L 指定对于任意码字可以存储在错误位表中的错误位索引的最大数量。不必总是存储关于所识别的所有错误位的信息。研究已经显示,对于 5000 位 LDPC 码字来说,针对具有三个以上 EBN 的俘获集合,翻转这些 EBN 中的任何三个将打破该俘获集合。因而,通常不需要将每码字多于三个的错误位存储在错误位表中。在这种情况下,3 是该表的限制 L。

[0086] 而且,大多数俘获集合可以通过翻转一个或两个 EBN 打破。如果进行判定以仅存储一个或两个错误位索引,即, L 为 1 或 2,并且所识别的错误位的数量大于 L,则测试 L 个选定索引,以确保按那些索引翻转位允许解码器收敛到 DCCW。否则,选择 L 个索引的其它集合,并且对其进行测试直到 (i) 解码器收敛到 DCCW 或 (ii) 没有更多的子集可用为止。

[0087] 图 7 是图 6 的步骤 612 的流程图。处理 612 选择图 6 的步骤 607 中检测到的错误位索引中的、将写入至错误位表的那一个(若有的话)。

[0088] 在步骤 702 开始处理,并且前进至步骤 704,其中,确定错误位的数量是否小于或等于表的限制 L。

[0089] 在步骤 704,如果错误位的数量小于或等于 L,则在步骤 706 选择所有的错误位索引,并且将变量 VARCOMPLETESET 的值设置成 1(真)。接下来,在步骤 708,将一记录附加至错误位表(例如,图 5 的 500)。所附加记录的字段设置如下。字段 CODEWORD_IDENTIFIER 被设置成等于特有的码字标识符(例如,闪速驱动器页面 ID)。字段 COMPLETE_SET 被设置成 VARCOMPLETESET 的值。将所选错误位索引存储至 EB_INDICES。

[0090] 相反,在步骤 704,如果错误位的数量大于 L,那么在步骤 712,确定 L 是否大于或等于 3。如果是,则在步骤 714,随机选择 L 个错误位索引,并且将 VARCOMPLETESET 设置成 0。接着,处理继续至步骤 716,其中,将变量 VARCOMPLETESET 设置成 0(即,将少于所有错误位的错误位存储至错误位表)。接着,处理进行至步骤 708(上述),并且在步骤 710 终止。

[0091] 相反,在步骤 712,如果 L 为 1 或 2,则处理继续至步骤 718,其中选择 L 个错误位索引。接下来,在步骤 720,翻转所选索引处的原始解码器输入码字中的那些位,以产生修正的解码器输入码字。接下来,在步骤 722,针对修正的解码器输入码字执行 LDPC 解码,以生成解码码字。接着,在步骤 724,比较解码码字与信道输入码字。如果在解码器输出码字中不

存在错误位（步骤 726），那么在步骤 716，将 VARCOMPLETESET 设置成 0，并且处理进行至步骤 708 和 710（上述）。

[0092] 相反，在步骤 726，如果确定解码码字包含任何错误位，那么在步骤 728，确定 L 个错误位索引的另一集合是否存在。如果存在，则在步骤 732，选择另一集合。接着，在步骤 734，将解码器输入码字重置成原始解码器输入码字（即，取消步骤 720 中进行的变化），并且处理循环至步骤 720。

[0093] 相反，在步骤 728，如果 L 个错误位索引的其它集合不存在，则在步骤 730，处理 612 产生针对信道控制器的控制，并且在步骤 710 终止。参见上面图 6 的步骤 614 的、针对信道控制器的说明。

[0094] 当系统从存储介质读取写入编码码字，并且 LDPC 解码器在指定最大数量的解码迭代内无法收敛到 DCCW 时，在以后的时间执行图 4 的写入错误恢复处理 418。写入错误恢复处理使用特有的码字标识符（例如，闪速驱动器页面 ID），以在错误位表中搜索匹配失败的解码码字的记录。如果找到匹配，则从该匹配记录检索错误位索引，并且翻转所检索索引处的失败的解码码字位的值，以产生修正的码字。

[0095] 如果所存储的错误位索引表示在写入验证处理期间检测到的所有错误位索引（即，COMPLETE_SET = 1），则典型地仅将修正的码字提交给出错校验和循环冗余校验。出错校验确定修正的解码器输入码字是否为有效 LDPC 码字，而循环冗余校验确定该有效 LDPC 码字是否为 DCCW。

[0096] 相反，如果所存储的错误位索引表示比写入验证处理期间所检测到的所有错误位索引少的错误位索引（即，COMPLETE_SET = 0），则提交修正的解码码字以进行进一步的 LDPC 解码（例如，图 3 的处理 300，而不需要初始化步骤 304）。当 (i) 进一步的解码收敛到 DCCW 或者 (ii) 解码器在指定的最大数量的解码迭代内无法收敛到 DCCW 时，写入错误恢复处理终止。

[0097] 图 8 是根据本发明一实施例的图 4 的写入错误恢复处理 418 的流程图。在步骤 802 开始处理，并且进行至步骤 804，其中，从 LDPC 解码器接收失败的解码码字。接下来，在步骤 806，处理 418 在错误位表（例如，图 5 中的 500）中搜索针对解码码字的匹配，例如，通过闪速驱动器页面 ID 来搜索。如果未发现匹配，则在步骤 808，处理 418 产生针对信道控制器的控制并接着在步骤 810 终止；否则，在步骤 812，从位错位表检索 EB_INDICES 和 COMPLETE_SET 的值。

[0098] 接下来，在步骤 814，处理 418 翻转所检索的错误位索引处的解码码字位的值，以产生修正的码字。上面在段落 [0060] 到 [0063] 中对翻转进行了描述。接下来，在步骤 816，如果 COMPLETE_SET 是 0（即，仅将错误位索引的适当子集存储在错误位表中），则在步骤 818 提交修正的码字，以进行 LDPC 解码。如果步骤 818 产生 DCCW（步骤 820），则处理 418 在步骤 810 终止。如果步骤 818 没有产生 DCCW，那么在步骤 808，产生针对信道控制器的控制，并且处理 418 在步骤 810 终止。

[0099] 另一方面，在步骤 816，如果 COMPLETE_SET 为 1（即，将所有错误位索引存储在错误位表中），则在步骤 822 提交修正的码字以进行出错校验。如果修正的码字对出错校验失败，则处理 418 产生针对信道控制器的控制（步骤 808），并且在步骤 810 终止。相反，在步骤 822，如果修正的解码器输入码字通过出错校验，则在步骤 824 执行 CRC 校验。如果修正

的解码器输入码字通过 CRC 校验,则处理在步骤 810 终止。相反,在步骤 824,如果修正的码字未通过 CRC 校验,则产生针对信道控制器的控制(步骤 808),并且处理在步骤 810 终止。

[0100] 在写入恢复处理 410 的另一实施例中,步骤 814 翻转解码器输入码字中的多个位,而非解码码字中的多个位。

[0101] 总之,本发明的实施例是用于根据包含写入错误的 NCW 来生成 DCCW 的方法。该方法典型地包括写入验证处理和写入错误恢复处理。在将信道输入码字写入至存储介质时所执行的写入验证处理识别写入码字中的写入错误(例如,错误位),并将一个或多个错误位的索引存储至错误位表。当解码器尝试解码码字并且因 NCW 而失败时,在以后某一时间所执行的写入恢复处理在错误位表中搜索与 NCW 相关联的错误位索引。如果找到关联的错误位索引,则翻转 NCW 中的对应位,并且提交 NCW,以进行进一步的 LDPC 解码。

[0102] 尽管本发明在硬盘驱动器和闪速驱动器的背景下进行了描述,但本发明不限于此。通常来说,本发明可以利用任何适当存储介质来实现。

[0103] 更进一步地,尽管本发明的实施例已经在 LDPC 代码的背景下进行了描述,但本发明不限于此。本发明的实施例可以针对可以通过图形定义的任何代码(例如,tornado 码、结构化 IRA 码)来实施,因为其是受到俘获集合影响的图形定义代码。

[0104] 本发明可以采用用于具体实践那些方法的方法和装置的形式来具体实施。本发明还可以采用在有形介质(如磁记录介质、光记录介质、固态存储器、软盘、CD-ROM、硬盘或任何其它机器可读存储介质)中包含的程序代码的形式来具体实施,其中,当通过机器(如计算机)加载该程序代码并执行时,该机器变为用于具体实践本发明的装置。本发明还可以采用例如或者存储在存储介质中或者通过机器加载和/或执行的程序代码的形式来具体实施,其中,当通过机器(如计算机)加载该程序代码并执行时,该机器变为用于实践本发明的装置。当在通用处理器上实施时,该程序代码段与该处理器组合,以提供与专用逻辑电路类似地操作的特有装置。

[0105] 除非另外明确规定,每一个数字值和范围都应被解译为近似的,如同该值或范围的值之前的单词“大约”或“近似”一样。

[0106] 还应明白,在不脱离如下列权利要求书中所表达的本发明的范围的情况下,本领域技术人员可以对已经描述和例示以便说明本发明的性质的那些部分的细节、材料和布置进行各种改变。

[0107] 权利要求书中使用的图号和/或附图标记旨在识别要求保护的主题的一个或多个可能的实施例,以便方便对权利要求书的诠释。这种使用不应解释为必须将那些权利要求的范围限制成对应图形中示出的实施例。

[0108] 应当明白,在此阐述的示例性方法的步骤不必要求按所述次序执行,并且应当明白这种方法的步骤的次序仅仅是示例性的。同样地,可以将附加步骤包括在这种方法中,并且可以在与本发明的各个实施例一致的方法中省略或组合特定的步骤。

[0109] 尽管下列方法权利要求中的要素(若有的话)按具有对应标号的特定顺序进行陈述,但除非权利要求的陈述另外暗示了用于实施这些要素中的一些或全部的特定顺序,那些要素都不必受限于按该特定顺序来实施。

[0110] 在此,引用“一个实施例”或“一实施例”意指可以将关于该实施例描述的特定特征、结构或特性包括在本发明的至少一个实施例中。在本说明书中的多个地方出现的短语

“在一个实施例中”不必全部指代同一实施例,也不是必需相互排除其它实施例的单独或另选实施例。这同样应用于术语“实现方式”。

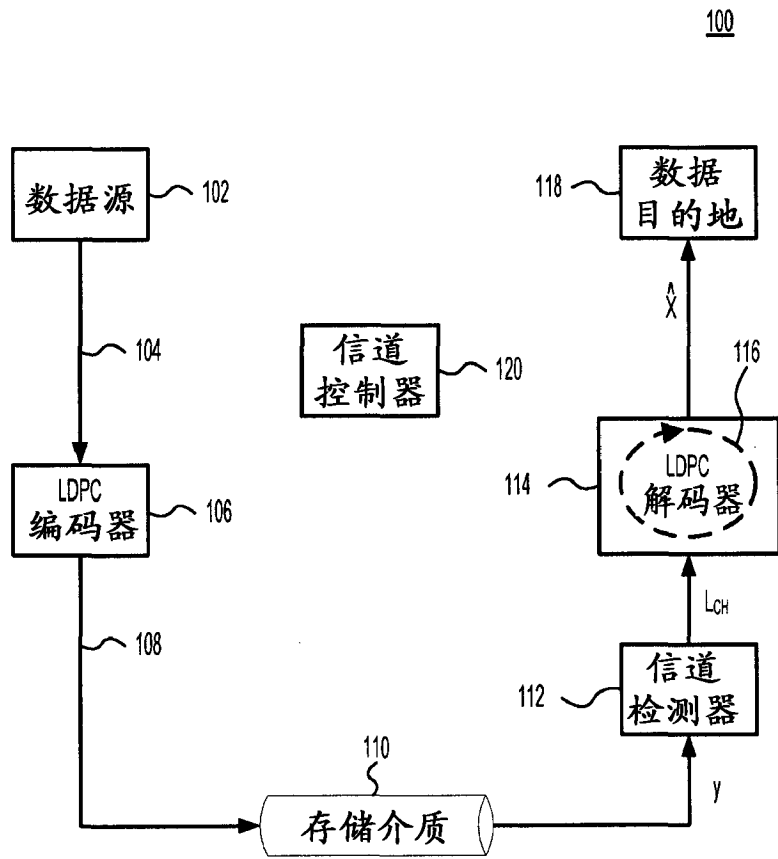


图 1

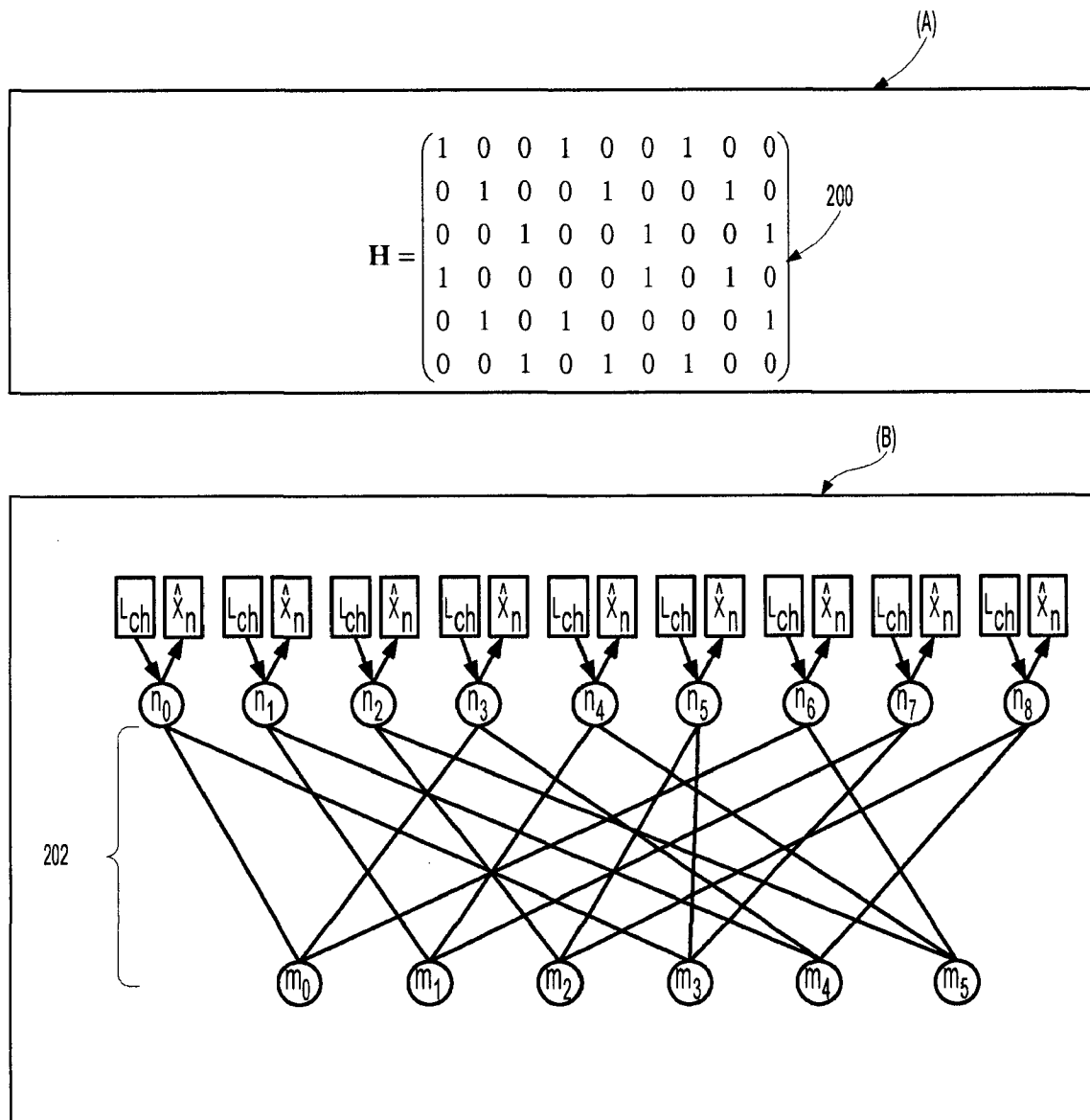


图 2

300

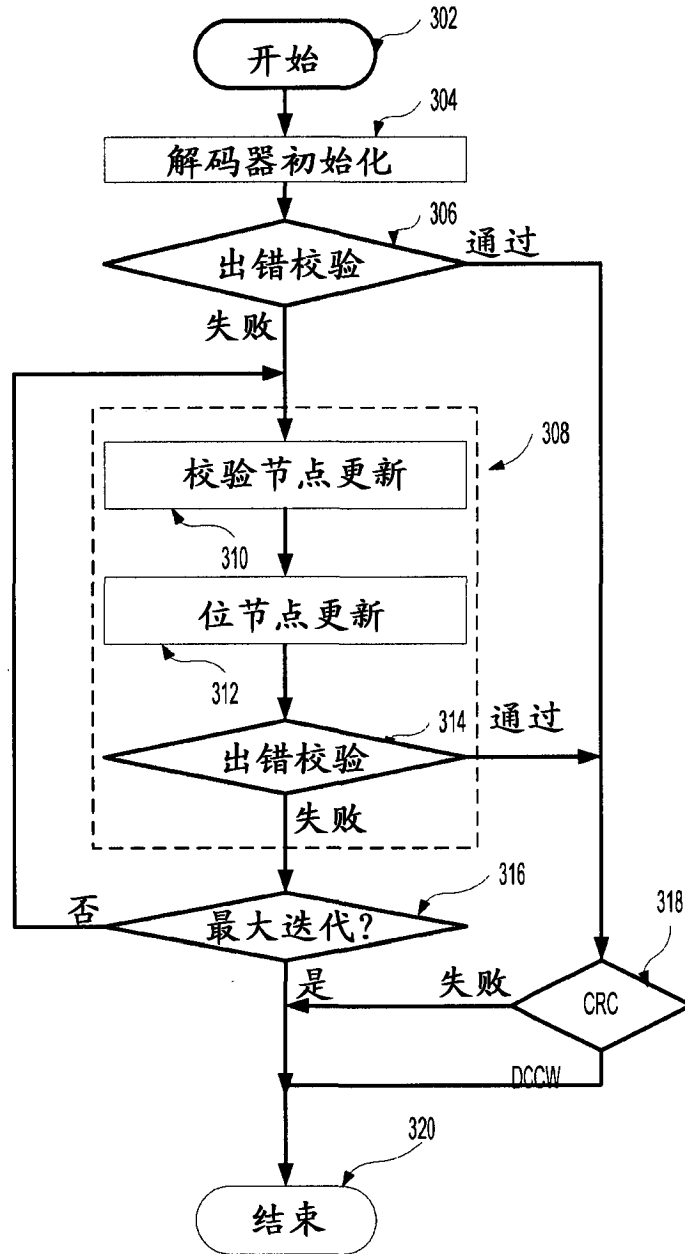


图 3

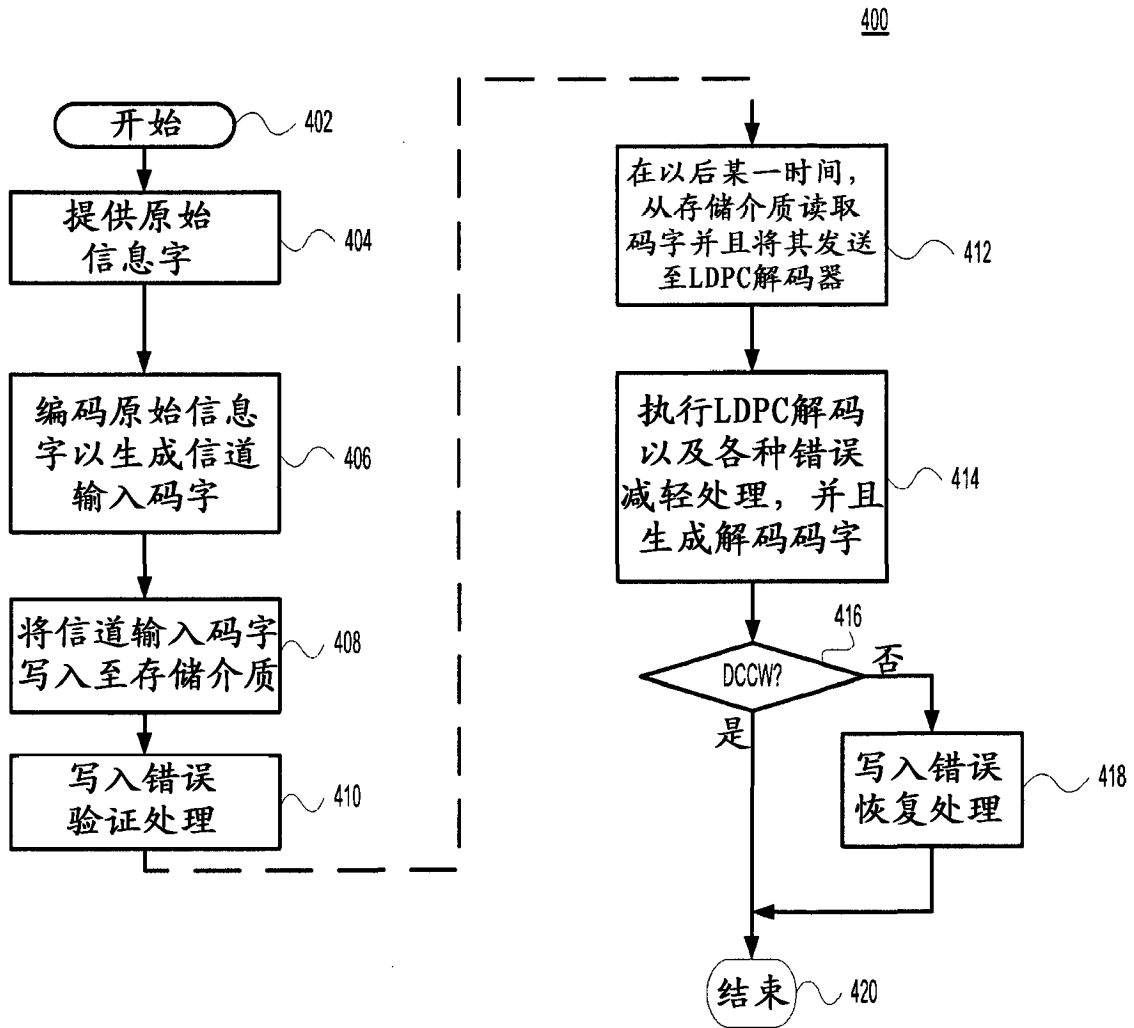


图 4

500

CODEWORD_IDENTIFIER	EB_INDICES	COMPLETE_SET
4002	3, 42, 101	0
391	30, 1007	1
.	.	.
.	.	.
.	.	.

图 5

410

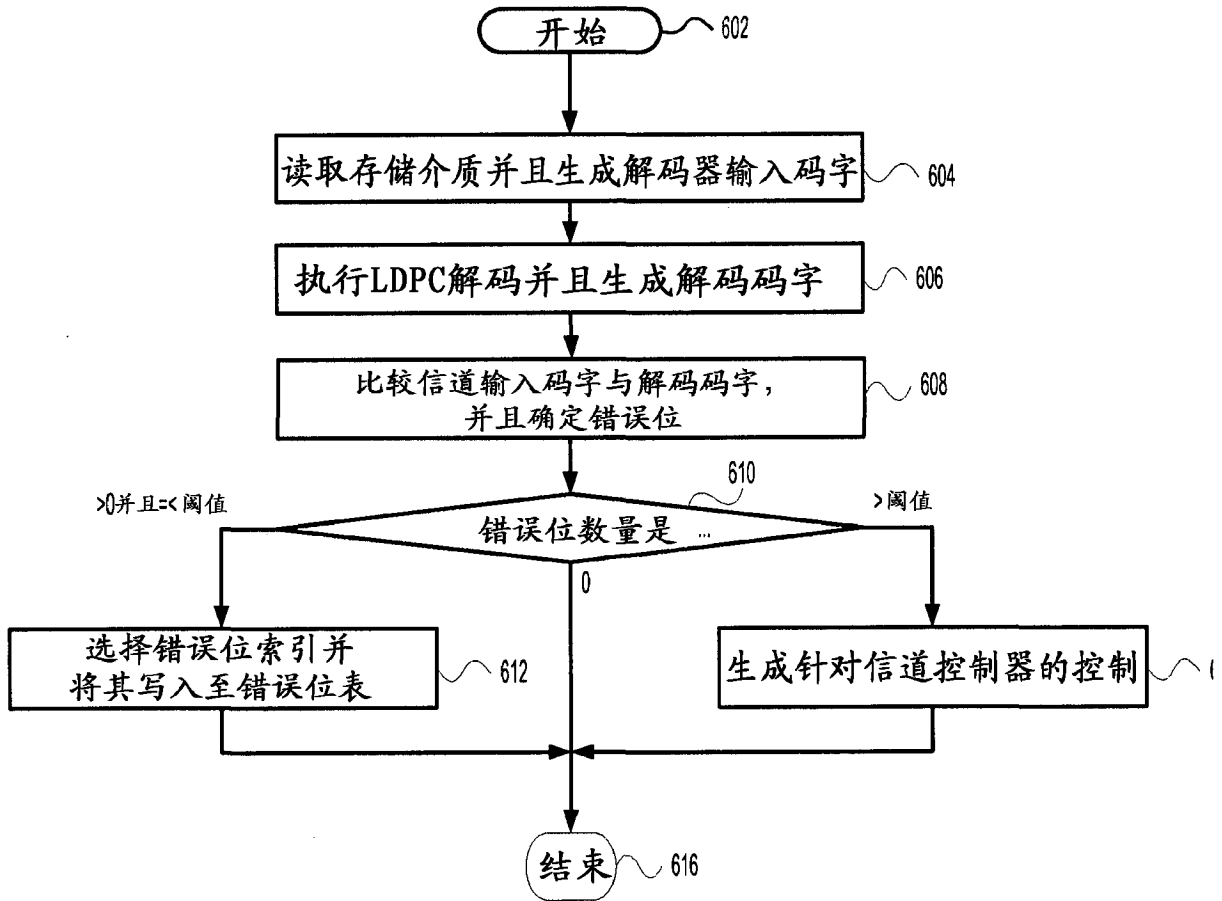


图 6

612

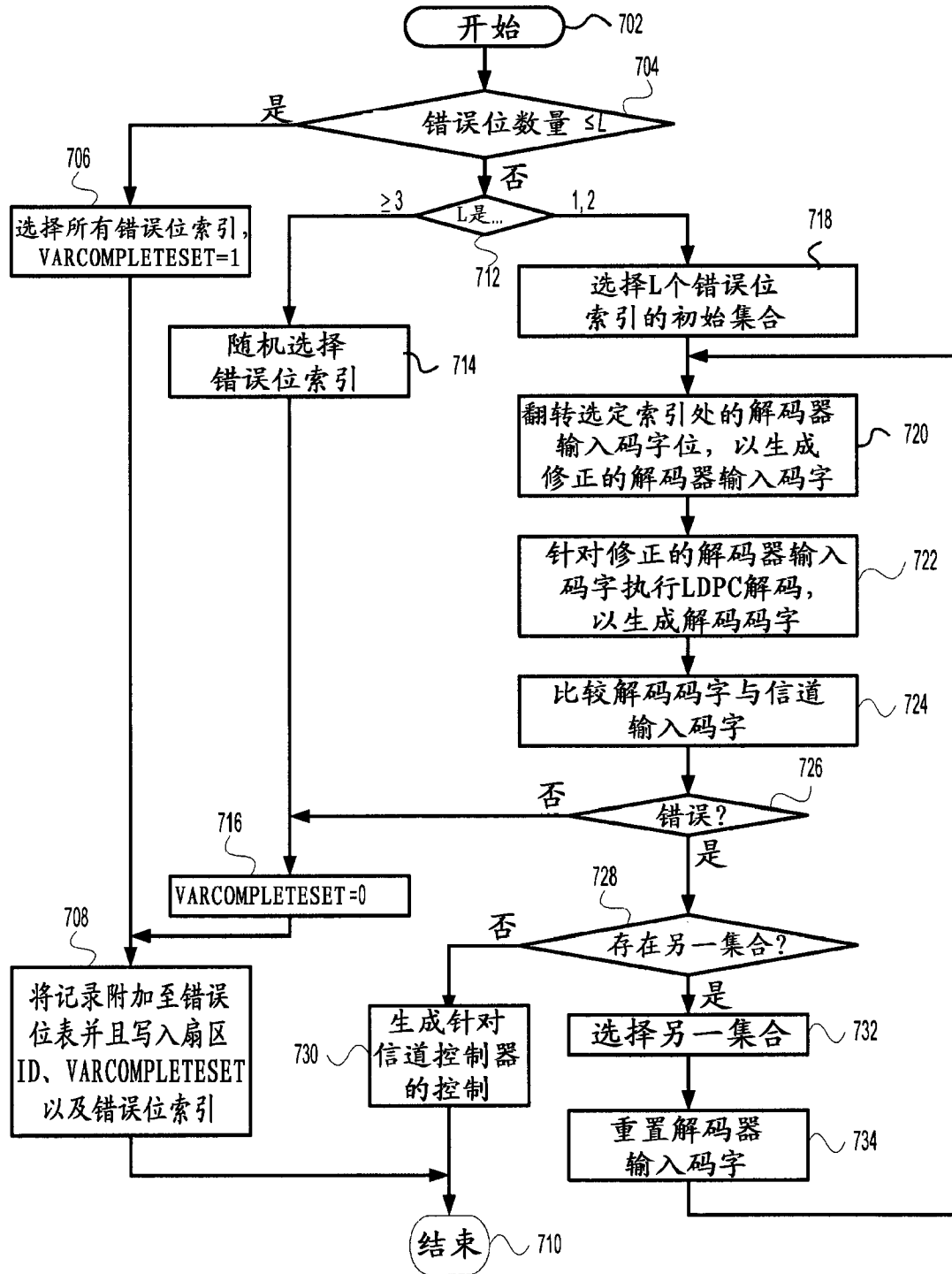


图 7

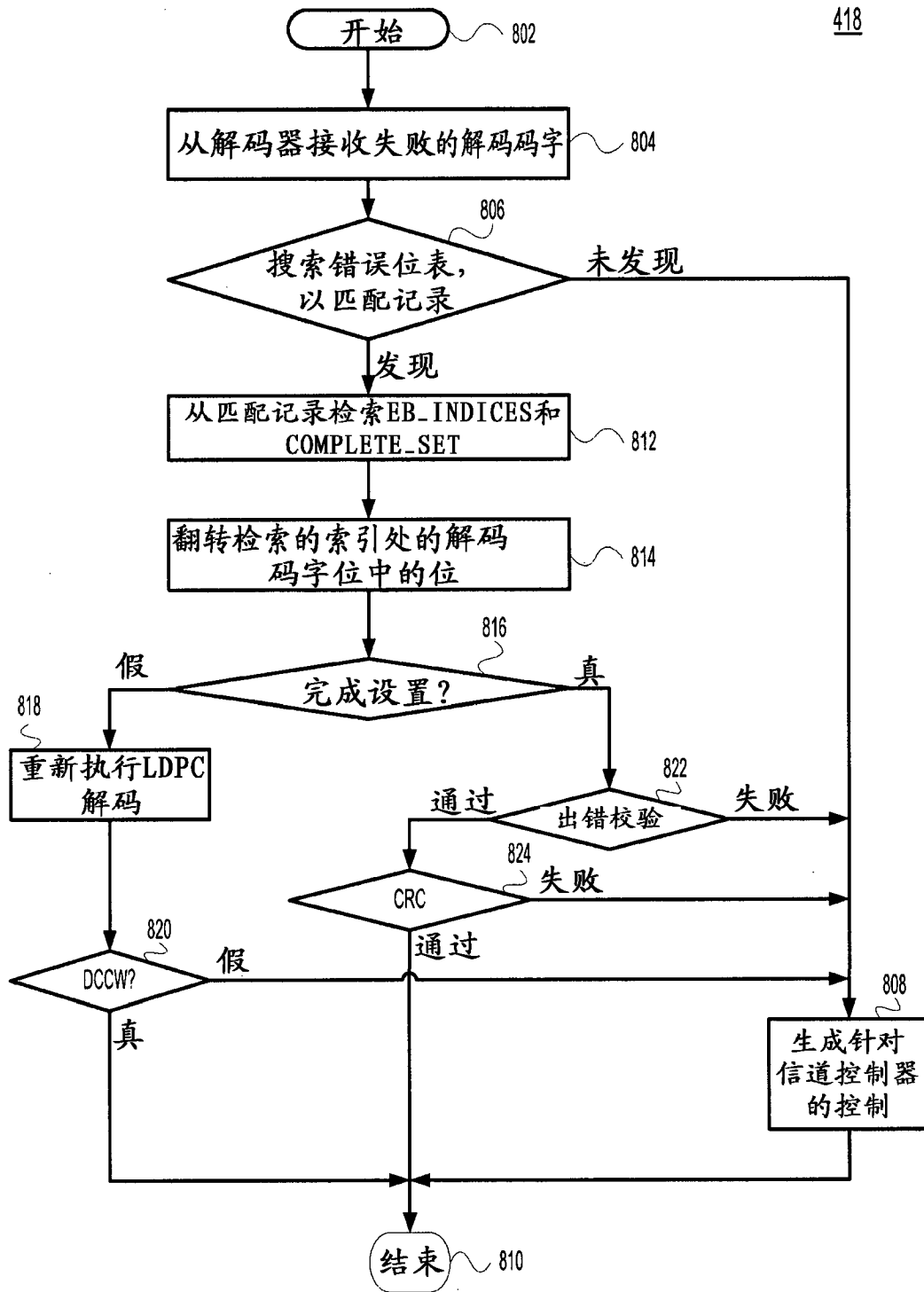


图 8