(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 101803206 B
(45)授权公告日 2013.09.04

(21)申请号 200880106457.3
(22)申请日 2008.12.12
(30)优先权数据
61/089,297 2008.08.15 US
(85)PCT申请进入国家阶段日
2010.03.10
(86)PCT申请的申请数据
PCT/US2008/086537 2008.12.12
(87)PCT申请的公布数据
WO2010/019169 EN 2010.02.18
(73)专利权人 LSI 公司
地址 美国加利福尼亚
(72)发明人 K·关纳姆
(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 申发振
(51)Int.Cl.
H03M 13/00(2006.01)

(56)对比文件
US 2006/0015802 A1,2006.01.19, 全文.
CN 101032082 A,2007.09.05, 全文.
US 2008/0148129 A1,2008.06.19, 全文.
US 2008/0163032 A1,2008.07.03, 全文.

审查员 陈毅强

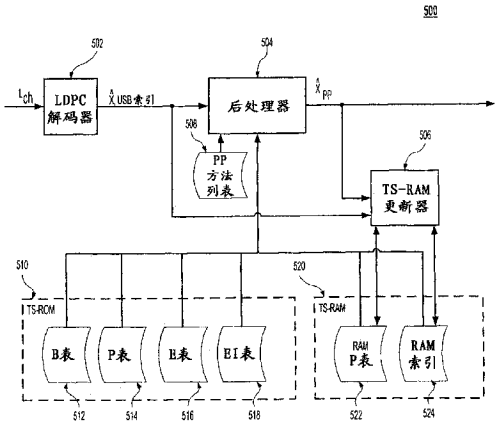
权利要求书2页 说明书17页 附图16页

(54)发明名称

近码字的ROM列表解码

(57)摘要

本发明的某些实施例是用于在ROM中组织陷阱集合简档，以及用于在(LDPC)列表解码过程中搜索这些简档的方法。按照优势，即，按照其对解码器的错误平台特性的影响给简档评分。更具优势的陷阱集合简档包含关于不满足校验节点(USC)和误满足校验节点(MSC)两者的信息，而优势较少的陷阱集合简档仅包含关于USC的信息。陷阱集合简档被组织到若干链接的分层数据表内，这些数据表允许使用指针跟随搜索迅速定位和检索最具优势的匹配的陷阱集合简档。



CN 101803206 B

1. 一种对使用基于图的码编码的编码数据进行解码的方法，该方法包括以下步骤：

(a) 对所述编码数据解码以产生候选解码码字，和

(b) 如果所述候选解码码字不是解码正确的码字，执行陷阱集合 TS-ROM 列表解码方法，以便试图产生解码正确的码字，其中：

所述候选解码码字具有至少一个不满足校验节点，其中不满足校验节点是奇偶校验失败的校验节点；

所述 TS-ROM 列表解码方法访问存储在 ROM 存储器内的一个或更多个 TS 简档；

第一个存储的 TS 简档包括至少一个不满足校验 USC 节点的存储信息以及至少一个误满足校验 MSC 节点的存储信息，其中 MSC 节点是与错误位节点 EBN 相关联、并且满足奇偶校验的校验节点；以及

第二个存储的 TS 简档与具有一个或更多个 USC 节点和一个或更多个 MSC 节点的陷阱集合相关联，其中第二个存储的 TS 简档包括一个或更多个 USC 节点的存储信息，但是不包含关于所述一个或更多个 MSC 节点的信息。

2. 如权利要求 1 所述的方法，其中第一个存储的 TS 简档与比第二个存储的 TS 简档的陷阱集合更具优势的陷阱集合相关联。

3. 如权利要求 1 所述的方法，其中所述 TS-ROM 列表解码方法包括以下步骤：

(b1) 识别与所述候选解码码字中的所述至少一个 USC 节点相关联的一个或更多个 EBN；

(b2) 如果 (i) 所述候选解码码字具有一个或更多个 MSC 节点，并且 (ii) 所述 ROM 存储器包含关于所述一个或更多个 MSC 节点的信息，则识别与所述一个或更多个 MSC 相关联的一个或更多个 EBN；

(b3) 修改识别出的 EBN；和

(b4) 执行进一步的处理，其中：

如果步骤 (b2) 识别出一个或更多个 EBN，则所述进一步的处理包括对修改后的候选解码码字执行校验子校验；和

如果步骤 (b2) 未识别出一个或更多个 EBN，则所述进一步的处理包括解码修改后的编码数据。

4. 如权利要求 1 所述的方法，其中：

对于 TS 简档中的每个不满足校验节点，所述 TS 简档包括：

所述不满足校验节点所在的解码层；

所述不满足校验节点 USC 在所述解码层内的索引；和

与所述不满足校验节点相关联的一个或更多个错误位节点 EBN 的一个或更多个索引；和

对于 TS 简档中的每个误满足校验节点，所述 TS 简档包括：

与所述误满足校验节点相关联的一个或更多个错误位节点的位置信息。

5. 如权利要求 4 所述的方法，其中：

所述 ROM 存储器包括多个存储的 TS 简档；

用于所述多个存储的 TS 简档的解码层、USC 索引和 EBN 索引存储在第一个表内；和

与误满足校验节点相关联的错误位节点的位置信息存储在第二个表内。

6. 如权利要求 5 所述的方法，其中：

所述存储的 TS 简档被基于所述存储的 TS 内的不满足校验节点的数目在第一个表内分组；

具有相同不满足校验节点数目的每组存储的 TS 简档中的存储的 TS 简档被按照 TS 简档的优势排序，其中 TS 简档的优势取决于相关联的陷阱集合对步骤 (a) 的解码的错误平台特性的影响。

7. 如权利要求 6 所述的方法，其中对于具有多个不满足校验节点的每个 TS 简档：

按解码层在所述 TS 简档中给所述不满足校验节点分组；和

按照 USC 索引排列具有相同解码层的不满足校验节点组内的不满足校验节点。

8. 如权利要求 5 所述的方法，其中所述 ROM 存储器还包括第三个表和第四个表，其中：

第三个表标识匹配的 TS 简档的与误满足校验节点相关联的错误位节点的位置信息在第二个表中的地址；

对于第一个表中具有相同数目的不满足校验节点的每组存储的 TS 简档，第四个表标识：

(i) 该组存储的 TS 简档在第一个表中的开始地址；

(ii) 该组中具有关于误满足校验节点的位置信息的 TS 简档的数目；和

(iii) 该组存储的 TS 简档在第三个表中的开始地址。

9. 如权利要求 1 所述的方法，其中所述基于图的码是低密度奇偶校验 LDPC 码。

10. 一种用于对使用基于图的码编码的编码数据进行解码的装置，该装置包括：

(a) 用于对编码数据解码以产生候选解码码字的装置，和

(b) 用于如果候选解码码字不是解码正确的码字则执行陷阱集合 TS-ROM 列表解码方法以便试图产生解码正确的码字的装置，其中：

所述候选解码码字具有至少一个不满足校验节点，其中不满足校验节点是奇偶校验失败的校验节点；

所述 TS-ROM 列表解码方法访问存储在 ROM 存储器内的一个或更多个 TS 简档；

第一个存储的 TS 简档包括至少一个不满足校验 USC 节点的存储信息，以及用于至少一个误满足校验 MSC 节点的存储信息，其中 MSC 节点是与错误位节点 EBN 相关联、并且满足奇偶校验的校验节点；以及

第二个存储的 TS 简档与具有一个或更多个 USC 节点和一个或更多个 MSC 节点的陷阱集合相关联，其中第二个存储的 TS 简档包括所述一个或更多个 USC 节点的存储信息，但是不包含关于所述一个或更多个 MSC 节点的信息。

近码字的 ROM 列表解码

[0001] 与相关申请的交叉引用

[0002] 本申请要求提交于 2008 年 8 月 15 日的代理卷号为 No. 08-0241-PR 的美国临时申请 No. 61/089,297 的优先权，通过引用将该申请的教导完整结合在此。

[0003] 本申请的主题内容涉及与本申请同日提交的代理卷号为 No. 08-0241 的美国专利申请 No. XX/xxx，xxx，通过引用将该申请的教导完整结合在此。

技术领域

[0004] 本发明涉及数字信号处理，并且更具体地，涉及被称为低密度奇偶校验 (LDPC) 编码的数据编码方法。

背景技术

[0005] 通信是由传输方在通信信道上向接收方传输信息。在现实世界中，通信信道是向接收方输出从传输方接收的信息的失真版本的有噪信道。硬盘 (HD) 驱动器就是这样一种有噪信道，其从传输方接收信息，存储该信息，然后可能将该信息的或多或少地失真的拷贝传输给接收方。

[0006] 由诸如 HD 驱动器的通信信道引入的失真可能大到足以引起信道错误，即，当信道输入信号是 0 时，接收方将信道输出信号解释为 1，或者反之。信道错误减小了吞吐率，并且因此是不希望的。因此，存在对检测和 / 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。LDPC 码是可以为低信噪比 (SNR) 应用实现非常低的位错误率 (BER) 的已知的近 Shannon 极限编码之一。LDPC 解码以其并行潜力、低实现复杂度、低解码延迟以及在高 SNR 下的不很严重的错误平台 (error-floor) 而著称。实际上认为 LDPC 码将用于所有下一代通信标准。

发明内容

[0007] 在某些实施例中，本发明包括用于对使用基于图的码编码的编码数据进行解码的方法。该方法包括 (a) 对编码数据解码，以便产生候选解码码字，和 (b) 如果候选解码码字不是解码正确的码字，执行陷阱集合 (TS)-ROM 列表解码方法，以便试图产生解码正确的码字。候选解码码字具有至少一个不满足校验节点，其中该不满足校验节点是奇偶校验失败的校验节点。TS-ROM 列表解码方法访问存储在 ROM 存储器内的一个或更多个 TS 简档。第一个存储的 TS 简档包括用于至少一个不满足校验 (USC) 节点的存储信息，以及用于至少一个误满足校验 (MSC) 节点的存储信息。MSC 节点是 (1) 与错误位节点 (EBN) 相关联并且 (2) 满足奇偶校验的校验节点。第二个存储的 TS 简档与具有一个或更多个 USC 节点和一个或更多个 MSC 节点的陷阱集合相关联，其中第二个存储的 TS 简档包括一个或更多个 USC 节点的存储信息，但是不包含关于一个或更多个 MSC 节点的信息。

[0008] 在其它实施例中，本发明是用于对使用基于图的码字编码的编码数据进行解码的装置。该装置包括 :(a) 被配置为对编码数据解码，以便产生候选解码码字的解码器，和 (b)

后处理器，被配置为如果候选解码码字不是解码正确的码字，则执行陷阱集合 (TS)-ROM 列表解码方法，以便试图产生解码正确的码字。候选解码码字具有至少一个不满足校验节点，其中不满足校验节点是奇偶校验失败的校验节点。TS-ROM 列表解码方法访问存储在 ROM 存储器内的一个或更多个 TS 简档。第一个存储的 TS 简档包括用于至少一个不满足校验 (USC) 节点的存储信息，以及用于至少一个误满足校验 (MSC) 节点的存储信息。MSC 节点是 (1) 与错误位节点 (EBN) 相关联并且 (2) 满足奇偶校验的校验节点。第二个存储的 TS 简档与具有一个或更多个 USC 节点和一个或更多个 MSC 节点的陷阱集合相关联，其中第二个存储的 TS 简档包括一个或更多个 USC 节点的存储信息，但是不包含关于一个或更多个 MSC 节点的信息。

附图说明

[0009] 从下面的详细描述、所附的权利要求书以及附图中，将会完全明了本发明的其它方面、特征和优点，在附图中相似的附图标记指示相似或相同的元件。

[0010] 图 1 是使用 LDPC 编码的典型硬盘 (HD) 驱动器 100 的一部分的方框图；

[0011] 图 2(A) 示出了 LDPC H 矩阵 200，图 2(B) 是 H 矩阵 200 的 Tanner 图；

[0012] 图 3 是由解码器 112 使用的典型 LDPC 解码方法 300 的流程图；

[0013] 图 4 是用于识别陷阱集合和记录关于这些陷阱集合的各种信息的离线陷阱集合 (TS) 仿真工具 400 的方框图；

[0014] 图 5 是根据本发明的一个实施例的 LDPC 解码方法 500 的方框图；

[0015] 图 6 是图 5 的 ROM P 表 514 的示例布局；

[0016] 图 7 是图 5 的 B 表 512 的示例布局；

[0017] 图 8 是图 5 的 E 表 516 的示例布局；

[0018] 图 9 是图 5 的 EI 表 518 的示例布局；

[0019] 图 10 是图 5 的 RAM P 表 522 的示例布局；

[0020] 图 11 是图 5 的 RAM 索引表 524 的示例布局；

[0021] 图 12 是图 5 的 LDPC 解码系统 500 所使用的示例处理 1200 的流程图；

[0022] 图 13 是由图 5 的后处理器 504 实施的图 12 的示例 TS-ROM 列表解码处理 1206 的流程图；

[0023] 图 14 是图 13 的示例 TS-ROM 搜索处理 1314 的流程图；

[0024] 图 15 是图 12 的示例 TS-RAM 列表解码处理 1208 的流程图；和

[0025] 图 16 是图 12 的示例 TS-RAM 更新处理 1216 的流程图。

具体实施方式

[0026] 图 1 是使用 LDPC 编码的典型硬盘 (HD) 驱动器 100 的一部分的方框图。HD 驱动器 100 包括盘片 102 和读通道 104。读通道 104 包括 LDPC 编码器 106、写处理器 108、读处理器 110 和 LDPC 解码器 112。路径 114 是 LDPC 编码器 106 和 LDPC 解码器 112 之间的有噪信道。

[0027] 由 LDPC 编码器 106 处理将被写到盘片 102 的信息字以便产生 LDPC 码字。LDPC 码字被发送到写处理器 108，写处理器 108 包括若干模块，例如，BPSK（二进制相移键控）编码

器、数字到模拟转换器等。写处理器 108 的输出 116 被写到盘片 102。

[0028] 从盘片 102 读取的信号 118 被发送到读处理器 110，读处理器 110 包括若干模块，例如，前置放大器、连续时间滤波器、固定激励响应滤波器、解码器、模拟到数字转换器等。读处理器 110 向 LDPC 解码器 112 输出对数似然比 (LLR) 值 L_{ch}，LDPC 解码器 112 又输出解码的信息字。另外，LDPC 解码器 112 将 E_{LDPC} 值送回读处理器 110。E_{LDPC} 被以下面的等式 6 定义，并且表示中间计算 LLR 值。读处理器 110 使用 E_{LDPC} 值调整其性能，这是被称为 turbo 解码的处理。

[0029] LDPC 编码

[0030] LDPC 编码器 106 给信息字的位附加由 LDPC 码指定的若干奇偶位，以便产生码字。信息字中的位被称为变量位，并且这些变量位的数目被表示为 K。LDPC 码字中的位的总数被表示为 N。因此，由 N-K 给出奇偶位的数目。特定 LDPC 码的比率是 K/N，即，信息字长度与码字长度的比。因此，给每 3 位信息字附加 6 个奇偶位以便产生 9 位码字的 LPDC 码具有 1/3 的比率。在典型的 HD 驱动器的情况下，对于 4506 位的码字长度和 0.9 的比率，信息字长度 K 是 4096 位（典型 HD 驱动器扇区的长度），并且奇偶位的数目近似为 410 位。

[0031] LDPC 码字中的每个奇偶位以由特定 LDPC 码所指定的特定方式与码字中的一个或更多个其它（变量或奇偶）位相关联，并且分配给奇偶位的值被设置为满足 LDPC 码。典型 LPDC 码规定相关联的位满足奇偶校验约束，例如，相关联的位的和是偶数，即，和模 2 = 0。

[0032] LDPC 码

[0033] 由被称为奇偶校验矩阵或 H 矩阵或简称为 H 的 1 和 0 的二维矩阵定义特定的 LDPC 码。LDPC 编码器和解码器事先知道 H。H 包括 N 列和 N-K 行，即，码字的每个位针对一列，并且每个奇偶位针对一行。H 中的每个 1 表示列的码字位和行的奇偶位之间的关联。例如，H 的第 3 行第 7 列处的 1 意味着第 3 个奇偶校验位与码字的第 7 个位相关联。校验位与和该校验位相关联的所有变量位的值的和模 2 应当为 0。

[0034] H 的一列中的 1 的数目被称为该列的权重 w_c。类似地，H 的一行中的 1 的数目被称为该行的权重 w_r。由所有列具有相同的 w_c 并且所有行具有相同 w_r 的 H 定义的 LDPC 码被称为规则 LDPC 码。由分别在所有列和 / 或行上不全相同的 w_c 和 / 或 w_r 的 H 定义的 LDPC 码被称为不规则 LDPC 码。

[0035] 典型 LDPC 码的一种定义特性是 H 是“稀疏的”，即，H 的元素大部分为 0，只有少数为 1。研究显示 H 矩阵通常需要 $w_c \geq 3$ 以便有良好表现，并且不规则 LDPC 码优于规则 LDPC 码。

[0036] 图 2(A) 示出了 LDPC H 矩阵 200。H 矩阵 200 包括 N = 9 列和 N-K = 6 行。因此，H 矩阵 200 定义接受 3 位信息字、附加 6 个奇偶位、并且输出 9 位码字的 LDPC 码。因此，这种特定 LDPC 码的比率是 3/9 或 1/3。由 H 矩阵 200 定义的 LDPC 码是规则的，其 w_c 为 2 并且 w_r 为 3。

[0037] 信道输出 :对数似然比

[0038] 返回图 1，LDPC 编码器 106 和 LDPC 解码器 112 之间的路径 114 是有噪信道，并且从而，解码器 112 不能接收由 LDPC 编码器 106 输出的码字的完美拷贝。而是，读处理器 110 输出一个或更多个 L_{ch} 值，其中每个 L_{ch} 值与信道输入码字中的一个位相对应。

[0039] 每个 L_{ch} 值是一个对数似然比 (LLR)。LLR 是包括若干位的数据结构，其中一个单

个的符号位指示硬判决（即，读处理器 110 关于原始位是 1 还是 0 的最佳猜测），并且剩余的量值位指示读处理器 110 对该硬判决的置信程度。更准确地，LLR 表示$\log\frac{p_0}{p_1}$，其中 p_0 是样本表示 0 的概率，并且 p_1 是样本表示 1 的概率。

[0040] 例如，读处理器 110 可以用 5 位数据结构输出每个 L_{ch} 值，其中最高位是指示硬判决值的符号位，并且 4 个量值位的 16 个值指示硬判决的置信度。因此，例如，在一种典型的方案中，二进制 00000 的 LLR 值指示具有最低置信度的硬判决值 0，二进制 01111 的 LLR 值指示具有最高置信度的硬判决值 0，二进制 10000 未使用，二进制 10001 指示具有最低置信度的硬判决值 1，并且二进制 11111 指示具有最高置信度的硬判决值 1。

[0041] LDPC 解码 :信任传播

[0042] 图 3 是由解码器 112 使用的典型 LDPC 解码方法 300 的流程图。LDPC 解码器 112 接收 N 个 L_{ch} 值，并且输出解码的信息字。解码方法 300 的核心是被称为信任传播的迭代的两阶段消息传递算法。通过使用被称为 Tanner 图的视觉表示最好地解释信任传播。

[0043] 图 2(B) 是 H 矩阵 200 的 Tanner 图。一般地，Tanner 图包括 :1) 等于 H 中的列数的位节点数目 n（并且因此等于变量位的数目），2) 等于 H 中的行数的校验节点数目 m（并且因此等于奇偶位的数目），3) 线，也被称为边，每个边将单个位节点连接到单个校验节点，4) 对于每个位节点 n，从接收方接收的原始 L_{ch} 值，和 5) 对于每个位节点 n，计算的硬判决输出值$\hat{x}_n$。图 2(B) 的 Tanner 图包括 9 个位节点 n_0-n_8，6 个校验节点 m_0-m_5，18 个将位节点连接到校验节点的边 202，9 个 L_{ch} 值和 9 个$\hat{x}_n$值。

[0044] Tanner 图中的边表示（即，变量）位节点 n 和校验节点 m 之间的关系，即，边表示 H 中的 1。例如，在图 2(B) 中，边 202 将第 1 个位节点 n_0 连接到第 4 个校验节点 m_3，这意味着在图 2(A) 的 H 矩阵 200 的第 1 列第 4 行中存在 1。

[0045] Tanner 图是二分图，即，一个边仅能将一个位节点连接到一个校验节点，而不能将一个位节点连接到另一个位节点，或者将一个校验节点连接到另一个校验节点。以 N(m) 表示被以边连接到特定校验节点 m 的所有位节点 n 的集合。以 M(n) 表示被以边连接到特定位节点 n 的所有校验节点 m 的集合。

[0046] 特定（位或校验）节点的索引是其在图中的序数序列。（位或校验）节点的度是连接到该节点的边的数目。因此，Tanner 图中位节点 n 的度等于相应 H 矩阵中的列 n 的权重 w_c，并且 Tanner 图中校验节点 m 的度等于相应 H 矩阵中的行 m 的权重 w_r。

[0047] 返回图 3，处理开始于步骤 302，并且进入步骤 304，进行解码器初始化。解码器初始化 304 包括将与每个位节点 n 连接的所有边（例如，图 2(B) 的 202）设置为与位节点 n 相关相应 L_{ch} 值，并且将位节点 n 的$\hat{x}_n$值设置为位节点 n 的 L_{ch} 的硬判决值。因此，例如，在图 2(B) 中，如果与位节点 n_0 相关联的 L_{ch} 值是 +5，则在步骤 304，将使位节点 n_0 连接到校验节点 m_0 和 m_3 的两个边 202 设置为 +5，并且位节点 n 的$\hat{x}_n$值被设置为 1。表述这个步骤的第一部分的另一种方式是位节点 n_0 向集合 $M(n_0)$ 中的每个校验节点 m 发送消息 +5。以 Q_{nm} 表示从位节点 n 发送到校验节点 m 的消息，其中 Q_{nm} 为 LLR 的形式。解码器刚被初始化后的状态被称为状态 0。

[0048] 随后，步骤 304 向校验子校验步骤 306 发送包括 N 个$\hat{x}_n$值的矢量$\hat{x}$。矢量$\hat{x}$是码字候选。校验子校验步骤 306 使用下面的等式 (1) 计算校验子矢量 z :

[0049]

$$z = \hat{x}H^T \qquad (1)$$

[0050] 其中 H^T 是 H 矩阵的转置。如果 z 是 0 矢量，则矢量$\hat{x}$满足由 H 定义的所有奇偶校验约束，即，$\hat{x}$是有效的解码码字。在该情况下，处理进入循环冗余校验 (CRC)318。

[0051] 相反，如果 z 不是 0 矢量，则矢量$\hat{x}$未通过一个或更多个奇偶校验约束，这通常被称为不满足校验节点或 USC。校验子矢量 z 中不为 0 标量值的元素的数目是矢量$\hat{x}$中的 USC 的数目 b。另外，校验子矢量 z 的非 0 标量元素的索引是 USC 在矢量$\hat{x}$中的索引。

[0052] 如果矢量$\hat{x}$未通过校验子校验 306，则处理继续一个或更多个解码迭代 308 中的第一个迭代。解码迭代 308 包括三个步骤 :1) 信任传播校验节点更新步骤 310，2) 信任传播位节点更新步骤 312 和 3) 与步骤 306 相同的校验子校验步骤 314。

[0053] 在信任传播校验节点更新步骤 310 中，每个校验节点 m 根据下面的等式 2、3 和 4 使用从集合 N(m) 中的所有位节点 n 接收的 Q_{nm} 消息计算以 R_{mn} 表示的消息 :

[0054] $$R_{mn}^{(i)} = \delta_{mn}^{(i)} \max\left(\kappa_{mn}^{(i)} - \beta, 0\right) \qquad (2)$$

[0055] $$\kappa_{mn}^{(i)} = \left|R_{mn}^{(i)}\right| = \min_{n' \in N(m)\backslash n} \left|Q_{n'm}^{(i-1)}\right| \qquad (3)$$

[0056] $$\delta_{mn}^{(i)} = \left(\prod_{n' \in N(m)\backslash n} \operatorname{sgn}\left(Q_{n'm}^{(i-1)}\right)\right) \qquad (4)$$

[0057] 其中 i 是解码迭代，N(m)\n 是除去位节点 n 之外的集合 N(m)，并且 β 是正的常数，其值取决于码参数。然后计算的 R_{mn} 消息被沿着相同的边送回集合 N(m) 中的所有位节点 n。类似于 Q_{nm} 消息，R_{mn} 消息是 LLR。

[0058] 接着，在信任传播位节点更新步骤 312，每个位节点 n 根据下面的等式 5 计算 Q_{nm} 消息 :

[0059] $$Q_{nm}^{(i)} = L_n^{(0)} + \sum_{m' \in M(n)\backslash m} R_{m'n}^{(i)} \qquad (5)$$

[0060] 其中 $L_n^{(0)}$ 是位节点 n 的 L_{ch} 值，并且 M(n)\m 是除去校验节点 m 之外的集合 M(n)。然后位节点 n 将计算的 Q_{nm} 消息发送到集合 M(n) 中的所有校验节点 m。

[0061] 并且，在位节点更新步骤 312 中，每个位节点 n 根据下面的等式 6 和 7 更新其$\hat{x}_n$:

[0062] $$E_n^{(i)} = \sum_{m \in M(n)} R_{mn}^{(i)} \qquad (6)$$

[0063] $$P_n = L_n^{(0)} + E_n^{(i)} \qquad (7)$$

[0064] 如果 $P_n \geq 0$，则$\hat{x}_n = 0$，并且如果 $P_n \leq 0$，则$\hat{x}_n = 1$。由等式 6 产生的值也称为 E 值或 E_{LDPC} 值。典型地，E_{LDPC} 值被作为称为 turbo 解码的调整处理的一部分被发送回读处理器 (例如，图 1 的读处理器 110)。由等式 (7) 产生的值被称为 P 值。以等式 (2)-(7) 表示的特定信任传播算法被称为最小和算法。

[0065] 注意，$\hat{x}_n$被在每个解码迭代 308 中更新，并且最后由解码处理 300 输出。原始 LLR 值 L_{ch} 在解码处理 300 中保持不变。换言之，在每个解码迭代 308 中，每个位节点 n 对其通过校验节点 m 相关联的所有其它位节点 n 的正确值进行投票。例如，在图 2(B) 中，位节点 n_0 与校验节点 m_0 和 m_3 相关联。因此，n_0 对与校验节点 m_0 和 m_3 相关联的位节点，即，n_3、n_5、

n_6 和 n_7 的正确值投票。位节点 n 的 L_{ch} 值的量值越大（即，置信度越大），位节点 n 的投票计数越多。这种投票的最终作用是具有低 L_{ch} 量值（即，置信度）的位节点的 $\hat{x}_n$ 值将发生改变，并且顺应与该位节点相关联的高置信度位节点的信任。换言之，如果位节点的 L_{ch} 值包含错误的硬判决值和低量值，则在一个或更多个迭代之后，其它位节点的组合投票趋于校正该错误的硬判决值。

[0066] 位节点更新步骤 312 将由解码器的当前 $\hat{x}_n$ 值构成的矢量 $\hat{x}$ 发送到校验子校验步骤 314。校验子校验步骤 314 与上面讨论的步骤 306 的校验子校验相同。如果矢量 $\hat{x}$ 通过了校验子校验 314，则矢量 $\hat{x}$ 被发送到 CRC 步骤 318。

[0067] LDPC 解码 :循环冗余校验和误满足校验节点

[0068] 通过校验子校验 306 或 314 意味着矢量 $\hat{x}$ 是有效的解码码字，但是不必然是解码正确的码字 (DCCW)。LDPC 解码器可能产生不是 DCCW 的有效码字。在该情况下，矢量 $\hat{x}$ 中不存在 USC，但是存在误满足校验节点 (MSC)。因此，为了确保有效的矢量 $\hat{x}$ 是 DCCW，处理 300 将矢量 $\hat{x}$ 传递到循环冗余校验 (CRC)318。CRC 校验是校验和运算，其可以检测传输或存储过程中的数据改变。

[0069] 如果矢量 $\hat{x}$ 通过了 CRC 校验，则矢量 $\hat{x}$ 是 DCCW，并且处理 300 将全局变量 DCCW 设置为真，输出矢量 $\hat{x}$ ，并且在步骤 320 处终止。否则，矢量 $\hat{x}$ 不是 DCCW，并且处理 300 将全局变量 DCCW 设置为假，输出矢量 $\hat{x}$ ，并且在步骤 320 处终止。全局变量 DCCW 通知其它解码处理（例如，下面讨论的图 12 的 TS-ROM 列表解码处理 1206）是否产生了 DCCW。

[0070] 返回步骤 314，如果矢量 $\hat{x}$ 未通过校验子校验，则矢量 $\hat{x}$ 仍然包含一个或更多个 USC。解决 USC 的典型方法是执行另一个解码迭代 308。然而，特定矢量 $\hat{x}$ 中可能存在不能在合理的时间量中被满足的一个或更多个 USC。因此，LDPC 解码器通常受限于它们可以对特定矢量 $\hat{x}$ 执行多少次解码迭代。迭代的最大数目的典型值的范围从 50 到 200。

[0071] 在图 3 中，步骤 316 确定是否达到了迭代的最大数目。如果未达到，执行另一个解码迭代 308。反之，如果达到了迭代的最大数目，则解码器处理 300 失败，即，解码器是“失败的解码器”。在该情况下，处理 300 将全局变量 DCCW 设置为假，输出矢量 $\hat{x}$ ，并且在步骤 320 终止。

[0072] 如果失败的解码器的矢量 $\hat{x}$ 包含小数目（例如，小于 16）的 USC，则矢量 $\hat{x}$ 被称为近码字 (NCW)。如果失败的解码器的矢量 $\hat{x}$ 包含大数目（例如，大于 15）的 USC，则矢量 $\hat{x}$ 被称为无效码字 (ICW)。

[0073] 处理失败的解码处理的两个典型方法是 1) 请求重新发送相应数据，或 2) 将矢量 $\hat{x}$ 传递到一个或更多个后处理 (PP) 方法。典型地，矢量 $\hat{x}$ 中的 USC 的数目 b 指示使用这两种方法中的哪一种。通常通过重发或其它后处理方法处理大的 b（例如，大于 16），而通过错误平台减轻后处理方法处理小的 b 值。

[0074] BER、SNR 和错误平台

[0075] LDPC 解码器的位错误率 (BER) 是表达对于处理的 x 个位将产生多少个错误解码位的比率。因此，例如，具有 10^{-9} 的 BER 的解码器对于处理的每十亿个解码位平均产生 1 个错误位。BER 越小，解码器越好。当解码器失败，即，解码器终止而没有收敛到解码正确的码字 DCCW 时，LDPC 解码器的 BER 增大（恶化）。

[0076] LDPC 解码器的 BER 受解码器的输入信号的信噪比 (SNR) 的强烈影响。作为 SNR 的

函数的 BER 图通常包括两个不同区域：初始“瀑布”区域，其中对于 SNR 的单位增加，BER 迅速改善（下降）；以及随后的“错误平台”区域，其中 SNR 的单位增加仅产生 BER 的略微改善。因此，在错误平台区域中实现显著的 BER 改进需要 SNR 增加之外的方法。

[0077] 一种改善 LDPC 解码的错误平台特性的方法是增加码字长度。然而，增加码字长度也增加 LDPC 解码所需的存储器和其它计算资源。因此，如果这些资源是紧缺的，HD 驱动器上的读通道设备通常就是这种情况，必须寻找其它方法产生必要的错误平台改进。

[0078] 另一种稀缺的资源是处理周期。通常，为了实现指定的吞吐率，HD 驱动器为解码一个码字安排固定数目的读通道处理周期预算。超出该预算的方法（即，计划外 (off-the-fly) 的方法）将减小吞吐率。更希望在时钟周期分配内恢复 DCCW，并且因此不减小吞吐率的计划内的方法。

[0079] 陷阱集合和优势陷阱集合

[0080] (a，b) 陷阱集合是解码器不能在最大数目的迭代内满足的 b 个 USC 以及与这些 USC 相关联的 a 个错误位节点 (EBN) 的集合。大部分陷阱集合包括少于 5 个 USC 和少于 10 个 EBN。陷阱集合对 LDPC 解码器的错误平台特性具有显著影响，即，当 LDPC 解码器未能收敛到 DCCW 时，通常是因为陷阱集合。

[0081] 改进 LDPC 解码器的错误平台特性的一种方法是：(i) 校验失败的解码器的$\hat{x}$矢量中的 USC，并且识别陷阱集合（如果有的话），(ii) 识别与这些 USC 相关联的 EBN，(iii) 翻转与这些陷阱集合相关联的一个或更多个 EBN，和 (iV) 重新启动解码器。在一种可能的实现中，如果 LDPC 解码器刚被初始化，即，解码器处于状态 0，翻转 EBN 包括：(i) 逆转 EBN 的 L_{ch} 值的硬判决值，即，1 变为 0，并且反之亦然，和 (ii) 将量值位，即，相同 L_{ch} 值的置信度设置为最大，例如，全为 1。如果解码器处于不是状态 0 的某个状态，则翻转 EBN 包括：(i) 确定 EBN 的 P 值（以上面的等式 7 定义）的硬判决值，(ii) 将 EBN 的 L_{ch} 值、P 值和所有相关联的 Q_{nm}LLR 的硬判决值设置为与步骤 (i) 的硬判决值相反，和 (iii) 将 EBN 的 L_{ch} 值、P 值和所有相关联的 Q_{nm}LLR 的量值位设置为最大。通常，翻转一个或两个 EBN 将“中断”陷阱集合，并且重新启动的解码器将收敛到 DCCW。

[0082] 当被中断时，不同的陷阱集合产生错误平台特性的不同改进。优势陷阱集合指其中断产生 EBR/ 错误平台特性的指定改进的最小陷阱集合。例如，DTS-1 指产生 BER 的单个数量级改进，例如，从 10^{-9} 到 10^{-10} 的最小陷阱集合，而 DTS-3 指产生 BER 的三个数量级改进，例如，从 10^{-10} 到 10^{-13} 的最小陷阱集合。

[0083] 近码字的列表解码

[0084] 列表解码是用于检测和中断陷阱集合的一种后处理方法。在列表解码中，将矢量$\hat{x}$中的观察到的陷阱集合与已知陷阱集合的一个列表或多个列表进行匹配。陷阱集合列表通常包括列表中的每个陷阱集合内的所有 USC 的索引，以及与这些 USC 相关联的一个或更多个 EBN 的索引。如果发现了该列表中的与观察到的陷阱集合匹配的陷阱集合，则从列表中检索 EBN 索引值（多个）。然后，翻转这些位节点，并且重新开始图 3 的解码处理 300。

[0085] 陷阱集合仿真

[0086] 通常使用软件和硬件仿真工具离线产生列表解码所需的陷阱集合列表。图 4 是用于识别陷阱集合并且记录关于这些陷阱集合的各种信息的离线陷阱集合 (TS) 仿真工具 400 的方框图。工具 400 可被实现在例如现场可编程门阵列 (FPGA) 内。LDPC 正确码字

(CCW)402 被发送到信道和信号模块 404，信道和信号模块 404 模拟图 1 的有噪信道 114 的行为。信道和信号模块 404 向 LDPC 解码器 408 输出 L_{ch} 值 406。如果 LDPC 解码器 408 产生近码字 (NCW)410，则 NCW410 被发送到校验子校验模块 412 和误匹配位置记录器 414。校验子校验模块 412 输出 NCW410 中的所有 USC 的索引 416。误匹配位置记录器 414 比较 NCW410 和 CCW402，并且输出 NCW410 中的所有 EBN 的索引 418。USC 索引 416 加上 EBN 索引 418 构成陷阱集合 (TS) 信息 420。

[0087] 对于给定的 LDPC 实现，所有可能的陷阱集合可能有数以百万。然而，实现该实现的错误平台的显著（即，一个数量级或更多）改进通常仅需要所有可能的陷阱集合的一个子集，即，优势陷阱集合 (DTS)。因此，离线 TS 仿真工具 400 包括 DTS-N 汇编器 422，它获取输入的 TS 信息 420，并且产生 DTS-N 信息 424。

[0088] DTS-N 汇编器 422 使用三个步骤的处理：收集、评分和评估。陷阱集合收集方法使用基于码图结构的确定性噪声激励，以便检测陷阱集合。然后，按照距错误边界的距离 (DEB) 值给收集的陷阱集合评分，其中具有低 DEB 值的陷阱集合对错误平台贡献更大。然后使用重要性采样评估陷阱集合，并且确认预测的评分。

[0089] 实际上，这种基于 FPGA 的离线仿真可能要被执行一年。例如，为了识别对于 4Gb/s 的 HD 驱动器将产生 10^{-15}BER 的陷阱集合，需要将离线仿真工具（例如，图 4 的工具 400）运行大约 289 天。通常，由于在 HD 驱动器读通道的最终设计和芯片的量产之间通常有 1 到 2 年的延迟，这种时间限制不是问题。

[0090] 陷阱集合只读存储器 (TS-ROM)

[0091] 因此，使用图 4 的离线 TS 仿真工具 400，可以事先识别对于特定的 LDPC 实现，将产生错误平台特性的改进的一个或更多个陷阱集合或优势陷阱集合。一种在运行时环境中实现列表解码的方法是在陷阱集合只读存储器 (TS-ROM) 中存储图 4 的离线产生的陷阱集合信息 420，并且将 TS-ROM 与列表解码器程序耦联。TS-ROM 列表解码器程序对 $\hat{x}$ 中观察到的 USC 和存储在 TS-ROM 中的陷阱集合进行比较。如果发现匹配，则 TS-ROM 列表解码器程序在 LDPC 解码器中翻转适当的位节点值，并且重新启动解码器。

[0092] 通常，TS-ROM 信息随机地存储在单向或双向链表内，其中使用暴力顺序搜索搜索该表。通常，每个 (a,b) 陷阱集合在 TS-ROM 列表中占用 (2+a+b) 个记录。因此，对于 (4,4) 陷阱集合简档（即，4 个 USC 和 4 个 EBN），将有 10 个记录：一个指示存在 4 个 USC 的记录，之后是 4 个单独的 USC 记录，然后一个指示存在 4 个 EBN 的记录，之后是 4 个单独的 EBN 记录。典型的 TS-ROM 列表实现大约存储 100 个陷阱集合，并且不存储关于误满足校验节点的任何信息。

[0093] 为了经济地实现这种 TS-ROM 实现，单个 TS-ROM 必需能够实现大量实现中所需的错误平台改进。然而，陷阱集合随着实现的不同而不同，甚至当实施相同的 LDPC 码时也是如此。例如，即使在两个 HD 驱动器上使用的 LDPC 码相同，与 HD 驱动器相关联的陷阱集合也可能不同。具体地，研究发现陷阱集合受 HD 驱动器的抖动曲线、符号间干扰特性和脉冲成形方案的影响。这些因素不仅可以在不同制造商的 HD 驱动器之间改变，而且还可以在来自相同制造商的不同 HD 驱动器型号之间，甚至在相同型号的不同产品批次改变。因此，即使在两个相同型号的硬盘驱动器之间，陷阱集合也可能改变。仿真如此众多的不同 HD 驱动器的 LDPC 陷阱集合是不现实的。另外，当与特定 HD 驱动器配对使用时，仅装载了大量 HD

驱动器公共的陷阱集合的 TS-ROM 可能不能产生所需的错误平台改进级别。

[0094] 一种改进 TS-ROM 的性能的方法是给由基于 FPGA 的离线仿真工具（例如，图 4 的工具 400）产生的信息补充从制造的设备的测试模型获得的结果。典型地，一旦结束了电路设计，将制造并且分发该设计的有限数目的测试模型，以便在开始大量生产之前进行测试。虽然可能要花费一年以便确定对于特定的 HD 驱动器实现产生 10^{-15} 的 BER 的陷阱集合，但仅用一天就可以确定产生 10^{-12} 的 BER 的陷阱集合。因此，测试模型被以 LDPC 测试模式运行有限的时间段，并且存储任意发现的陷阱集合。将未在 TS-ROM 中的任意被发现的陷阱集合添加到 TS-ROM 内。通过使用分发给消费者的实际设备，这种方法可以捕捉可能逃过了基于 FPGA 的离线仿真工具（例如，图 4 的工具 400）的陷阱集合。

[0095] 陷阱集合随机访问存储器 (TS-RAM)

[0096] 一种对 TS-ROM 的静态陷阱集合列表的运行时替换是在陷阱集合随机访问存储器 (TS-RAM) 中存储陷阱集合信息，并且将图 4 的离线陷阱集合仿真工具 400 变为运行在实际的单个设备（例如，HD 驱动器）上的运行时陷阱集合收集和分析工具。取代从信道和信号模块（例如，图 4 的模块 404）接收初始值，运行时工具处理特定设备的实际信号。运行时工具包括列表解码器功能，即，它将尝试匹配观察到的 USC 和 TS-RAM 中存储的陷阱集合信息，并且如果发现了匹配，使用存储的信息改变解码器的位节点值，并且重新启动解码器。如果未发现匹配，则运行时工具分析观察到的陷阱集合，即，识别与该 USC 相关联的 EBN，并且确定观察到的陷阱集合是否满足存储在 TS-RAM 中的阈值要求（例如，DTS-N 中的成员资格）。

[0097] 在理论上，上述的 TS-RAM 工具适合于任意实现的陷阱集合简档。现实是由图 4 的离线仿真工具 400 执行的陷阱集合 / 优势陷阱集合仿真是计算复杂的。特别地，从可能的数十亿个陷阱集合构建优势陷阱集合尤其复杂。这种复杂性使得上述 TS-RAM 工具不适合于大多数 HD 驱动器。通常，HD 驱动器高速输出数据（每秒 4 吉比特），并且需要非常低的 BER/ 错误平台率（例如，10^{-13} 到 10^{-15}），但是仅在其固件中提供不多的计算资源。

[0098] 另外，TS-RAM 工具，例如，图 4 的离线仿真工具 400，需要正确码字 (CCW) 以便产生 EBN 索引。容易在离线仿真环境中获得 CCW，但是在运行时环境中不易获得。

[0099] 根据本发明的某些实施例，执行用于在 ROM 中组织存储的陷阱集合简档的方法。按照优势，即，按照其对 LDPC 解码器的错误平台特性的影响给陷阱集合简档评分。更具优势的陷阱集合简档包含关于不满足校验节点 (USC) 和误满足校验节点 (MSC) 两者的信息，而具有较少优势的陷阱集合简档仅包含关于 USC 的信息。然后，将陷阱集合简档信息组织到若干链接的分层数据表内，这些表允许使用指针跟随搜索迅速定位并且检索优势最大的匹配的陷阱集合简档。

[0100] 根据本发明的某些实施例，执行用于在 RAM 中收集和识别优势陷阱集合的高效的运行时方法。如果可能，新发现的陷阱集合被存储在 RAM 中，并且然后根据一个或更多个下列因素排序或评分：自最后匹配一个陷阱集合以来，RAM 被搜索的次数；自被添加到 RAM 内以来，陷阱集合被匹配的总次数；不满足校验节点的数目；和错误位节点的数目。评分低的陷阱集合简档被从 RAM 中删除，并且给新发现的陷阱集合简档留出空间。因此，除了或取代使用诸如图 4 的 DTS-N 汇编器 422 中所使用的用于事先识别优势陷阱集合的高计算复杂度的离线方法，本发明的这些实施例执行低计算复杂度的在后方法，其中尽可能多地存储新发现的陷阱集合，并且通过周期评分和删除去除非优势陷阱集合简档。

[0101] 本发明的实施例通常是计划内 (on-the-fly) 的方法，即，它们能够在LDPC解码的时钟周期预算内恢复DCCW，并且因此不会负面影响系统的吞吐率。

[0102] 图5是根据本发明的一个实施例的LDPC解码系统500的方框图。在与图1的现有技术的HD驱动器100类似的本发明的HD驱动器中，图5的LDPC解码系统500将被实现为与图1的LDPC解码器112类似的LDPC解码器的一部分。对此而言，图5的输入 L_{ch} 值类似于图1的解码器输入 L_{ch} 值，并且图5的输出 $\hat{x}_{pp}$ 矢量类似于图1的解码信息字。

[0103] LDPC解码器502接收 L_{ch} 值，执行图3的LDPC解码处理300，并且向后处理器504和TS-RAM更新器506输出矢量 $\hat{x}$ 。后处理器504被连接到后处理 (PP) 方法列表508，后处理 (PP) 方法列表508是包含表示后处理方法，例如，TS-ROM列表解码、TS-RAM列表解码等的一个或更多个可执行程序的存储器。如果后处理器504需要执行特定的PP方法，后处理器504从PP方法列表508中读取可执行程序，并且运行该程序。后处理器504可以并行或串行执行任意数目的这些PP方法。后处理器504输出矢量 $\hat{x}_{pp}$ ，除了作为LDPC解码系统500的输出之外，矢量 $\hat{x}_{pp}$ 还被发送到TS-RAM更新器506。

[0104] 数据表

[0105] 在执行过程中，特定PP方法可能需要访问与PP方法的可执行程序代码相分离的数据结构。具体地，TS-ROM和TS-RAM列表解码方法分别访问存储在TS-ROM510和TS-RAM520中的陷阱集合信息的一个或更多个列表。

[0106] 在图5的示例实施例中，TS-ROM510包括4个表：B表512、P表514、E表516和EI表518。TS-RAM520包括2个表，RAM P表522和RAM索引524。表是被组织在一个或更多个相等大小的行（记录）和一个或更多个相等大小的列（字段）内的数字数据的二维矩阵。表的记录被从上到下从0开始顺序编号。这个编号是记录号。

[0107] P表514和522包含关于USC和其相关EBN的信息。B表512包含用于ROM P表514的指针信息。EI表518包含关于MSC的信息，并且E表516包含用于EI表518的指针信息。RAM索引表524包含用于RAM B表522的指针信息。

[0108] 图6是图5的ROM P表514的示例布局。ROM P表514包含陷阱集合简档信息，即，USC和EBN索引。ROM P表514包含若干记录（行），每一个记录用于一个存储的陷阱集合的一个USC。记录号602是记录在P表514中的从0开始的顺序位置。

[0109] ROM P表514中的每个记录包括3个字段：LAYER 604，USC_INDEX 606和EBN_INDEX 608。某些LDPC解码器被配置为并行地执行被另称为层的一组更新操作。LAYER 604指示包含在USC中的解码层的数目。USC_INDEX 606包含USC的索引。EBN_INDEX 608包含与该USC相关联的一个或两个EBN的索引。

[0110] ROM P表514首先被按照b（即，$\hat{x}$ 中USC的数目）排序，例如，首先是b＝2的所有陷阱集合，后面是所有b＝3的陷阱集合等。因此，存在用于b＝2值域中的每个陷阱集合（例如，610、612）的两个记录，最后跟着用于b＝3的陷阱集合的三记录集合（例如，614、616），用于b＝4的陷阱集合的四记录集合（例如，618、620）等等。

[0111] 在每个b值域中，按照优势，即，陷阱集合对错误平台特性的影响对陷阱集合排序。对错误平台特性具有更大影响的陷阱集合出现在该b值域的开头处，并且具有较少影响的陷阱集合出现在靠近结尾处。然后按照USC_INDEX 606给特定陷阱集合的记录排序。

[0112] 图7是图5的B表512的示例布局。对于ROM P表514中的每个b值，B表512包

含指向该 b 值在 ROM P 表 514 和 E 表 516 中的首次出现的指针，以及该 b 值在 E 表 516 中的记录的数目。因此，B 表 512 中存在用于每个 b 值的单个记录，其中由记录号 702 指示 b 值。然而，记录号 702 从 0 开始，而 b 值通常从 2 或更大开始。因此，在本发明的这种示例实施例中，给记录号 702 添加一个偏移，以便产生相应的 b 值。例如，如果仅存储 $b \geq 2$ 的陷阱集合，则给每个记录号添加偏移 2，以便得到相应的 b 值。

[0113] 字段 PTABLE_START_OFFSET 704 包含特定 b 值在 ROM P 表 514 中首次出现的位置。字段 ETABLE_START_OFFSET 706 包含特定 b 值在 E 表 516 中首次出现的位置。字段 NUM_ETABLE_ENTRIES 708 包含这个特定 b 值在 E 表 516 中的记录数目。

[0114] 图 8 是图 5 的 E 表 516 的示例布局。E 表 516 包含指向 EI 表 518 中的 MSC 记录的指针。E 表 516 中的每个记录具有记录号 802、EITABLE_START_ADDRESS 字段 804 和 EITABLE_END_ADDRESS 字段 806。EITABLE_START_ADDRESS 字段 804 包含指向相应数据在 EI 表 518 中的首次出现的指针，并且 EITABLE_END_ADDRESS 字段 806 包含指向相应数据在 EI 表 518 中的最后出现的指针。

[0115] 图 9 是图 5 的 EI 表 518 的示例布局。EI 表 518 存储与 MSC 相关联的 EBN 的索引。EI 表 518 中的每个记录具有记录号 902，并且包含两个字段：BLOCK_COLUMN 字段 904 和 B_INDEX 字段 906。BLOCK_COLUMN 字段 904 指示 EBN 所在的块列，并且 B_INDEX 字段 906 是 EBN 的索引。

[0116] 图 10 是图 5 的 RAM P 表 522 的示例布局。RAM P 表 522 存储不能在 ROM P 表 514 中找到的新识别出的陷阱集合的简档。RAM P 表 522 中的每行（即，记录）具有记录号 1002，并且包括两个字段：两位 TAG 字段 1004 和 R_WORD 字段 1006。TAG 字段 1004 的 4 个可能的值指示记录类型和 R_WORD 字段 1006 内的数据的结构。如果 TAG 字段 1004 具有值 11，则该记录是包含关于整个陷阱集合的信息的主记录。如果 TAG 字段 1004 具有值 10，则该记录是包含关于陷阱集合简档内的特定 USC 的信息的副记录。如果 TAG 字段 1004 具有值 00 或 01，则 R_WORD 字段 1006 为空，即，这个记录是可用于存储新识别的陷阱集合的简档信息。陷阱集合简档通常包括单个主记录，后面跟有 b 个副记录。

[0117] 如果记录是主记录，则 R_WORD 字段 1006 包含 4 个子字段：(i)b 值子字段 1008，(ii)a 值子字段 1010，其记录陷阱集合 EBN 的数目，(iii)LAST_HIT_NUM 子字段 1012，其指示最后匹配这个陷阱集合的 TS-RAM 搜索的数目，和 (iV)HIT_COUNTER 子字段 1014，其记录自从这个陷阱集合被存储在 TS-RAM 内以来，这个特定的陷阱集合简档已经与观察到的陷阱集合匹配了多少次。

[0118] 如果记录是副记录，R_WORD 字段 1006 包含陷阱集合内的单个 USC 的层 (LAYER 字段 1016) 和索引 (USC_INDEX 字段 1018)，以及与该 USC 相关联的一个或更多个 EBN 的索引 (EBN_INDEX 字段 1020)。

[0119] 图 11 是图 5 的 RAM 索引表 524 的示例布局。RAM 索引表 524 中有用于 RAM P 表 522 中的每个陷阱集合简档的记录。RAM 索引表 524 中的每个记录包括单个字段，RAM_PTABLE_OFFSET 1102。RAM_PTABLE_OFFSET 1102 是指向 RAM 索引表 524 中的特定陷阱集合简档的开始的指针，即，RAM_PTABLE_OFFSET 1102 包含陷阱集合简档主记录的图 10 的记录号 1002。RAM 索引表 524 中的记录被按照优势排序，从而 RAM 索引表 524 中最后的记录指向 RAM P 表 522 中最不具优势的记录。

[0120] 图 12 是由图 5 的 LDPC 解码系统 500 使用的示例处理 1200 的流程图。处理在步骤 1202 处开始，并且进入步骤 1204，由图 5 的 LDPC 解码器对 L_{ch} 值进行 LDPC 解码。如果步骤 1204 的 LDPC 解码产生 DCCW，则处理 1200 在步骤 1218 处终止。否则，处理进入步骤 1206 的 TS-ROM 列表解码（由图 5 的后处理器 504 执行）。

[0121] 如果步骤 1206 产生 DCCW，则处理 1200 在步骤 1218 处终止。否则，处理进入步骤 1208 的 TS-RAM 列表解码（由图 5 的后处理器 504 执行）。如果步骤 1208 产生 DCCW，则处理 1200 在步骤 1218 处终止。否则，处理进入以类似方式操作的一个或更多个附加的后处理方法 1210、1212、...、1214（由图 5 的后处理器 504 执行）。

[0122] 如果 TS-RAM 列表解码 1208 或附加的后处理方法 1210、1212...、1214 产生 DCCW，则处理进入步骤 1216，其中可能以 TS-RAM 更新器 508 更新图 5 的 TS-RAM520。然后，处理在步骤 1218 处终止。

[0123] 图 12 显示了后处理方法 1207-1214 的一种可能的顺序。可以采用后处理方法的几乎任意顺序，虽然某些顺序比其它顺序更实用。例如，希望将 TS-ROM 列表解码安排在 TS-RAM 列表解码之前，以便确保已经存储在 ROM 中的陷阱集合不会在 RAM 中重复。

[0124] TS-ROM 列表解码

[0125] 图 13 是由图 5 的后处理器 504 执行的图 12 的示例 TS-ROM 列表解码处理 1206 的流程图。处理在步骤 1302 处开始，并且进入步骤 1304，其中处理 1206 确定从图 12 的 LDPC 解码器 1204 接收的矢量$\hat{\boldsymbol{x}}$中的观察到的 USC 的数目 $b_{observed}$ 是否大于 0 并且小于可由处理 1206 有效地处理的 USC 的最大数目 b_{max}。如果 $b_{observed} = 0$，则矢量$\hat{\boldsymbol{x}}$中没有 USC（即，$\hat{\boldsymbol{x}}$是近码字，误校正），并且因此，不存在要匹配的陷阱集合。如果步骤 1304 的估计为假，则处理 1206 终止。否则，处理继续到步骤 1306。

[0126] 在步骤 1306，存储解码器的当前状态，并且将其标记为状态 1。然后处理继续到步骤 1308，其中对观察到的 USC 排序，首先按照解码层排序，然后按照索引排序。接着，在步骤 1310，从图 5 的 B 表 512 中取出并且存储下面 4 个值：

[0127] (1) 对于 $b = b_{observed}$，图 7 的 PTABLE_START_OFFSET 字段 704；

[0128] (2) 对于 $b = b_{observed}+1$，图 7 的 PTABLE_START_OFFSET 字段 704；

[0129] (3) 对于 $b = b_{observed}$，图 7 的 ETABLE_START_OFFSET 字段 706；和

[0130] (4) 对于 $b = b_{observed}$，图 7 的 NUM_ETABLE_ENTRIES 字段 708。

[0131] 第一个值指示处理 1206 从哪里开始其对图 5 的 P 表 514 中的匹配陷阱集合信息（即，USC 和 EBN 索引）的搜索，并且第二个值指示处理 1206 何时结束其搜索。类似地，第三个值和第四个值指示处理 1206 从哪里开始和结束其对扩展信息（即，MSC 索引）的搜索。

[0132] 因此，例如，如果 $b_{observed} = 5$，则处理 1206 取回对于 $b = 5$ 和 $b = 6$ 的图 7 的 PTABLE_START_OFFSET 字段 704，以及对于 $b = 5$ 的图 7 的 ETABLE_START_OFFSET 字段 706 和图 7 的 NUM_ETABLE_ENTRIES 字段 708。

[0133] 接着，在步骤 1312，处理 1206 选择图 5 的 P 表 514，并且进入由 $b = b_{observed}$ 的 PTABLE_START_OFFSET 的存储值所指示的地址。接着，在步骤 1314，在 TS-ROM 中搜索与观察到的 USC 匹配的陷阱集合。

[0134] 图 14 是图 13 的示例 TS-ROM 搜索处理 1314 的流程图。处理 1314 在步骤 1402 开始，并且在步骤 1404 搜索图 5 的 P 表 514 中的作为观察到的 USC 的同构匹配的下一个记录。

观察到的 USC 的特定集合的同构匹配是陷阱集合，其中 USC 的数目和这些 USC 之间的距离与观察到的 USC 相同。因此，如果观察到的 USC 是 [1,3,10]，则 [1,3,10] 是一种匹配，并且 [2,4,11] 是一种同构匹配，同样还有 [3,5,12]，[4,6,13] 等等。如果未发现匹配，则处理 1314 以无匹配状态 1406 终止。

[0135] 相反，如果在步骤 1404 发现匹配，则在步骤 1408，存储匹配的 P 表记录的图 6 的 EBN_INDEX 字段 608 的值。EBN_INDEX 字段包含与这个匹配的陷阱集合相关联的一个或可能两个错误位节点的索引。

[0136] 接着，处理 1314 试图定位任何扩展信息，即，与这个匹配的陷阱集合中的误满足校验节点 (MSC) 相关联的 EBN 的索引。扩展信息保持在 EI 表 518 中。然而，不是为存储在 P 表 514 内的所有陷阱集合保持扩展信息，而是仅为每个 b 值域中的陷阱集合的一个子集保持。该子集与特定 b 值域中更有优势的陷阱集合相对应，即，对错误平台特性具有更显著影响的那些陷阱集合。如上所述，在 P 表 514 内，特定 b 值域内的陷阱集合按优势排序；因此，仅为 b 值域内的前 x 个记录保持扩展信息。以 B 表 512 中的字段 ETABLE_START_OFFSET706 和 NUM_ETABLE_ENTR1ES 708 指示 EI 表 518 中每个 b 值域的开始和结束。

[0137] 处理 1314 在搜索 ROM P 表 514 时保持陷阱集合的内部计数。因此，例如，如果处理 1314 正在搜索图 6 的 P 表 514 中的 b ＝ 2 的陷阱集合，处理 1314 将记录 0 和 1 识别为陷阱集合 0（例如，图 6 的陷阱集合 610），将记录 2 和 3 识别为陷阱集合 1（例如，图 6 的陷阱集合 612）并且依此类推。这个陷阱集合号被称为 TSNUM。

[0138] 在步骤 1410，对 TSNUM 和在图 13 的步骤 1310 存储的 NUM_ETABLE_ENTRIES 字段的值进行比较。如果 TSNUM 大于 NUM_ETABLE_ENTRIES 的值，则没有可获得的扩展信息，并且处理 1314 以没有扩展信息 1412 的匹配状态终止。

[0139] 相反，如果在步骤 1410 发现 TSNUM 小于或等于 NUM_ETABLE_ENTRIES 的存储值，则存在这个匹配的陷阱集合的扩展信息。在步骤 1414，将 TSNUM 加到 ETABLE_START_OFFSET 的存储值，以便产生变量 ETABLE_ENTRY_ADDRESS。

[0140] 接着，在步骤 1416，处理 1314 选择图 5 的 E 表 516，进入具有等于变量 ETABLE_ENTRY_ADDRESS 的值的地址的记录，并且存储图 8 的 EITABLE_START_ADDRESS 字段 804 和图 8 的 EITABLE_END_ADDRESS 字段 806 的值。

[0141] 接着，在步骤 1418，处理 1314 选择图 5 的 EI 表 518，并且存储在所存储的 EITABLE_START_ADDRESS 和 EITABLE_END_ADDRESS 的值之间的每个记录的图 9 的 BLOCK_COLUMN 字段 904 和图 9 的 B_INDEX 字段 906 的值。最后，处理 1314 以具有扩展信息 1420 的匹配状态退出。

[0142] 返回图 13，如果步骤 1314 以无匹配状态终止，则处理 1206 在步骤 1316 终止。

[0143] 如果步骤 1314 以无扩展信息匹配状态终止，则处理 1206 拥有 USC 索引以及与这个陷阱集合相关联的某些 EBN 索引，但是没有扩展信息（即，与 MSC 相关联的 EBN 的索引）。在这种情况下，步骤 1318 翻转这些 EBN 索引处的位节点，并且在步骤 1320 执行迭代 LDPC 解码。除了跳过解码器初始化步骤 304 和校验子校验步骤 306 之外，步骤 1320 的处理与图 3 的处理 300 相同。如果步骤 1320 收敛到 DCCW，则处理 1206 在步骤 1316 终止。否则，在步骤 1322，解码器恢复到状态 1，并且在步骤 1314 寻找下一个匹配的陷阱集合。

[0144] 如果步骤 1314 以具有扩展信息的匹配状态终止，则处理 1206 拥有与匹配的陷阱

集合相关联的所有 EBN 的索引。在该情况下，不必执行信任传播（例如，图 3 的步骤 310 和 312）。相反，在步骤 1324，翻转 EBN，并且将得到的矢量$\hat{x}$提交校验子校验 1326。如果矢量$\hat{x}$在步骤 1326 未通过校验子校验，则处理 1206 进入步骤 1322。相反，如果矢量$\hat{x}$在步骤 1326 通过了校验子校验（即，矢量$\hat{x}$是有效的码字），则在步骤 1328，对矢量$\hat{x}$执行 CRC 校验，以便确定它实际上是否是正确的码字。如果矢量$\hat{x}$通过了 CRC 校验 1328（即，矢量$\hat{x}$是 DCCW），则处理 1206 在步骤 1316 终止。如果矢量$\hat{x}$未通过 CRC 校验 1328，则处理 1206 进入步骤 1322。

[0145] TS-RAM 列表解码

[0146] 由图 5 的后处理器 504 使用的另一种 PP 方法是图 12 的 TS-RAM 列表解码 1208，即，使用存储在易失存储器诸如随机访问存储器内的陷阱集合信息对陷阱集合解码。TS-RAM 列表解码类似于 TS-ROM 列表解码，即，图 5 的 RAM P 表 522 存储所选择的陷阱集合的 USC 和 EBN 信息。然而，与 ROM P 表 514 不同，在运行时过程中由图 5 的 TS-RAM 更新器 506 改变 RAM P 表 522。因此，仅存储最重要的信息，例如，USC 和 EBN 索引。不在 TS-RAM 中保持扩展信息（例如，图 5 的 EI 表 518）。

[0147] 不以任意方式对 RAM P 表 522 中简档排序。相反，一个单独的图 16 的 RAM 索引表 524 保持按优势排序的存储在 RAM P 表 522 内的陷阱集合简档的地址列表。

[0148] 图 15 是图 12 的示例 TS-RAM 列表解码处理 1208 的流程图。处理在步骤 1502 开始，并且进入步骤 1504，步骤 1504 的目的和操作与图 13 的步骤 1304 相同。如果步骤 1504 估计为假，处理 1208 在步骤 1506 终止；否则，处理继续到步骤 1508，其中记录当前解码器状态，并且将其标记为状态 1。然后处理继续到步骤 1510。

[0149] 在步骤 1510，处理 1208 进入图 5 的 RAM P 表 522 中的最具优势的陷阱集合简档。具体地，由于 RAM 索引表 524 按照优势给 RAMP 表 522 中的简档评分，处理 1208 进入 RAM 索引表 524 中的第一个记录，并且检索图 11 的 RAM_PTABLE_OFFSET 字段 1102 的值。然后，处理 1208 将 RAM P 表 522 中的指针移动到存储的偏移值。

[0150] 在步骤 1512，处理 1208 递增全局变量 RAM_SEARCH_COUNT，全局变量 RAM_SEARCH_COUNT 跟踪执行的 TS-RAM 搜索的总数，即，已经执行处理 1208 的总次数。另外，在步骤 1512，处理 1208 以由 RAM 索引表 524 指定的顺序，即，以下降的优势的顺序在 RAMP 表 522 内的简档中校验观察到的 USC 的同构匹配。如果未发现匹配，则处理继续到步骤 1514。

[0151] 相反，如果在步骤 1512 发现了匹配，则步骤 1516，匹配的简档的图 10 的 LAST_HIT_NUM 字段 1012 被设置为全局变量 RAM_SEARCH_COUNT 的值，并且图 10 的 HIT_COUNTER 字段 1014 增加 1。然后，在步骤 1518，存储图 10 的 EBN_INDEX 字段 1020 的值。步骤 1520 翻转位于 EBN_INDEX 值处的位节点，并且在步骤 1522，执行 LDPC 解码。步骤 1522 与图 13 的步骤 1320 相同。如果解码处理 1522 收敛到 DCCW，则处理继续到步骤 1514；否则，在步骤 1524，解码器被重置为状态 1，并且然后处理继续到步骤 1512，其中在 P 表 522 中寻找另一个同构匹配。

[0152] 在步骤 1514，处理 1208 更新图 11 的 RAM 索引表 524。具体地，步骤 1514 按照 RAM P 表 522 中的字段，例如，LAST_HIT_NUM1012、HIT_COUNTER 字段 1014、USC 节点数目字段 1008 和 EBN 数目字段 1010 的任意组合对 RAM P 表 522 中的所有 TS-RAM 简档排序。然后所有有序简档的地址，即，所有主记录的记录号 1002 被存储为图 11 的 RAM 索引表 524 中的记录。简档地址被以在步骤 1514 中排序的相同顺序存储在 RAM 索引表 524 中（例如，从最具

优势到最不具优势）。

[0153] 一旦完成了步骤 1514，处理在步骤 1506 终止。

[0154] TS-RAM 更新器

[0155] 如在对图 12 的处理 1200 的讨论中解释的，如果除了 TS-ROM 列表解码 1206 之外的任意后处理方法获得了 DCCW，则可能意味着发现了新的陷阱集合。如果是这样，步骤 1216 可以尝试将新的陷阱集合添加到图 5 的 RAM P 表 522。

[0156] 在一个实施例中，步骤 1216 是用于保留 TS-RAM520 中的优势陷阱集合的低复杂度处理。具体地，在这个实施例中，步骤 1216 不执行穷尽性计算，以便事先确定优势陷阱集合，诸如由图 4 的 DTS-N 汇编器 422 执行的计算，而是相反 (i) 按照一个或更多个因素，例如，自从陷阱集合被最后匹配以来，TS-RAM 已被搜索了多少次、陷阱集合已被匹配的总次数、USC 数目、EBN 数目等的任意组合给 TS-RAM 陷阱集合评分；并且然后 (ii) 清除评分最低的陷阱集合，以便给新发现的陷阱集合留出空间。使用这些因素给 TS-RAM520 中的陷阱集合评分通常比由离线仿真工具（例如，图 4 的汇编器 422）执行的分析的复杂度低得多。

[0157] 图 16 是图 12 的示例 TS-RAM 更新处理 1216 的流程图。处理在步骤 1602 开始，并且进入步骤 1604，其中确定是否由图 12 的 TS-RAM 列表解码处理 1208 产生了 DCCW。如果为是，则不需要向 TS-RAM 中添加陷阱集合简档，并且处理 1216 在步骤 1608 终止。

[0158] 相反，如果步骤 1604 估计为否 / 假，则意味着不同于 TS-ROM 或 TS-RAM 列表解码的某个后处理方法获得了 DCCW，因此已经发现了新的陷阱集合，并且应当将其附加到 RAM。在步骤 1610，产生陷阱集合简档。陷阱集合简档包括陷阱集合 USB 的索引和与这些 USB 相关联的 EBN 的索引。已经由图 5 的 LDPC 解码器 502 产生了 USB 索引。为了产生 EBN 索引，步骤 1610 对由后处理器 504 产生的 DCCW $\hat{x}_{pp}$ 和由 LDPC 解码器 502 产生的矢量 $\hat{x}$ 进行比较。

[0159] 在步骤 1612，确定是否存在足够的空间以便附加新的陷阱集合简档。如果是这样，则在步骤 1614，新的陷阱集合简档被附加到 RAMP 表 522，并且处理 1216 进入步骤 1616。

[0160] 然而，如果在步骤 1612，RAM P 表 522 中没有足够的空间以便附加新的陷阱集合简档，则在步骤 1618，清除评分最低的符合清除条件的陷阱集合简档。在这个例子中，按照自从该简档被最后匹配以来 RAM 被搜索的次数，即，全局变量 RAM_SEARCH_COUNT 的值减去简档的 LAST_HIT_NUM 字段的值确定简档对清除条件的符合性。如果插入搜索的数目大于指定阈值，则简档符合清除条件。假设所有简档首先按照对清除条件的符合性评分，所有符合清除条件的记录在图 11 的 RAM 索引表 524 的结尾处。

[0161] 因此，在步骤 1618，选择 RAM 索引表 524 中的最后记录，并且检索 RAM_PTABLE_OFFSET 字段 1102 的值。如果 RAM P 表 522 内的位于存储的偏移值（由检索到的 RAM_PTABLE_OFFSET 值指示）处的简档符合清除条件，则删除 RAM P 表 522 中的位于存储的偏移值处的主记录和相关联的副记录。在步骤 1620，更新 RAM 索引表 MSB，并且控制循环回到步骤 1612，其中确定 RAM P 表 522 内是否存在足够的空间以便附加新的陷阱集合简档。步骤 1620 的处理与图 15 的步骤 1514 的处理相同。

[0162] 相反，如果评分最低的简档不符合清除条件，则处理继续到步骤 1616。在步骤 1616，更新 RAM 索引表 523，并且处理在步骤 1608 终止。步骤 1616 的处理与图 15 的步骤 1514 的处理相同。

[0163] 虽然以实施 LDPC 编码和解码的硬盘驱动器的上下文描述了本发明，但本发明不

限于此。一般地，本发明可被实现在涉及 LPDC 编码和解码的任意适合的通信路径中。

[0164] 另外，虽然上面使用的示例信任传播算法是偏移量最小和算法 (OMS)，但本发明不限于此，而可以用于任意信任传播变体，例如，和积算法 (SPA) 或 Bahl-Cocke-Jelinek-Raviv(BCJR) 算法。

[0165] 另外，虽然上面使用的信任传播例子使用特定的解码调度（泛滥调度），其中在单个校验节点更新步骤中更新所有校验节点，之后在单个位节点更新步骤中更新所有位节点，但本发明不限于此，而可以用于任意解码调度，例如，行串行调度、列串行调度和行 - 列串行调度。

[0166] 另外，虽然上面使用的示例 LDPC 解码器是不分层的解码器，但本发明不限于此，而可以使用分层解码器和不分层解码器。

[0167] 另外，虽然上面给出的示例 TS-RAM 实现假设在 HD 驱动器的读通道内的 RAM 中存储陷阱集合简档，但本发明不限于此。RAM P 表（例如，图 5 的 522）还可以存储在 HD 驱动器的盘片上，或存储在诸如闪存的单独存储器内。

[0168] 另外，虽然以只读存储器的上下文描述了上面给出的示例 TS-ROM 实现，但本发明不限于此。一般地，说明书和权利要求书使用的术语“ROM”应当被解释为指存储静态 TS 简档数据的任意数据存储设备，不论该设备内的数据是否能够被修改。

[0169] 另外，虽然以 LDPC 码的上下文描述了本发明的实施例，但本发明不限于此。可以为可由图定义的任意码实现本发明的实施例，例如，tornado 码、结构化 IRA 码，因为它们都是具有陷阱集合问题的以图定义的码。

[0170] 虽然以接收对数似然比描述了本发明，但本发明不限于此。可设想处理诸如似然比的其它软值或硬判决值的本发明的实施例。

[0171] 本发明可被表达为方法和用于实现这些方法的装置的形式。本发明可被表达为包含在有形介质内的程序代码的形式，所述有形介质诸如是磁记录介质、光学记录介质、固态存储器、软盘、CD-ROM、硬盘驱动器或任意其它机器可读的存储介质，其中当程序代码被装入诸如计算机的机器并且由其执行时，该机器成为实现本发明的装置。本发明还可被表达为，例如，存储在存储介质内的、被装入和 / 或由机器执行的、或在某种传输介质或载体上传输的程序代码的形式，诸如在电线或电缆上传输、通过光纤传输或通过电磁辐射传输，其中当程序代码被装入诸如计算机的机器并且由其执行时，该机器成为实现本发明的装置。当在通用处理器上实现时，该程序代码段与处理器相结合，以便提供类似于专用逻辑电路操作的独特设备。

[0172] 除非明确说明，每个数值和范围应被解释为是近似的，就如同在该值或范围前面加有单词“大约”或“近似”一样。

[0173] 还应当理解，本领域的技术人员可以对为了解释本发明的属性而描述和示出的部件的细节、材料和布置进行各种改变，而不脱离由所附的权利要求书表述的本发明的范围。

[0174] 权利要求书中使用的附图号和 / 或附图参考标记旨在标识所要求的主题内容的一个或更多个可能的实施例，以便于解释权利要求。这种使用不能被认为是将这些权利要求的范围必然地限制为相应附图中示出的实施例。

[0175] 应当理解，此处提出的示例方法的步骤不必然需要以描述的顺序执行，并且这些方法的步骤的顺序应当被理解为仅是例子。同样，这些方法中可以包括附加步骤，并且可以

在根据本发明的各种实施例的方法中忽略或组合某些步骤。

[0176] 虽然以相应的标记以特定顺序叙述，如果有的话，下面的方法权利要求中的元素，除非权利要求表述暗示着实现某些或全部这些元素的特定顺序，这些元素不必然旨在局限于以该特定顺序实现。

[0177] 此处对“一个实施例”或“实施例”的引用意味着结合该实施例描述的特定特征、结构或特性可被包括在本发明的至少一个实施例中。在说明书各处出现的短语“在一个实施例中”不必然全部指相同的实施例，不同的或可替换的实施例也不必然互相排除在其它实施例之外。这同样适用于术语“实现”。

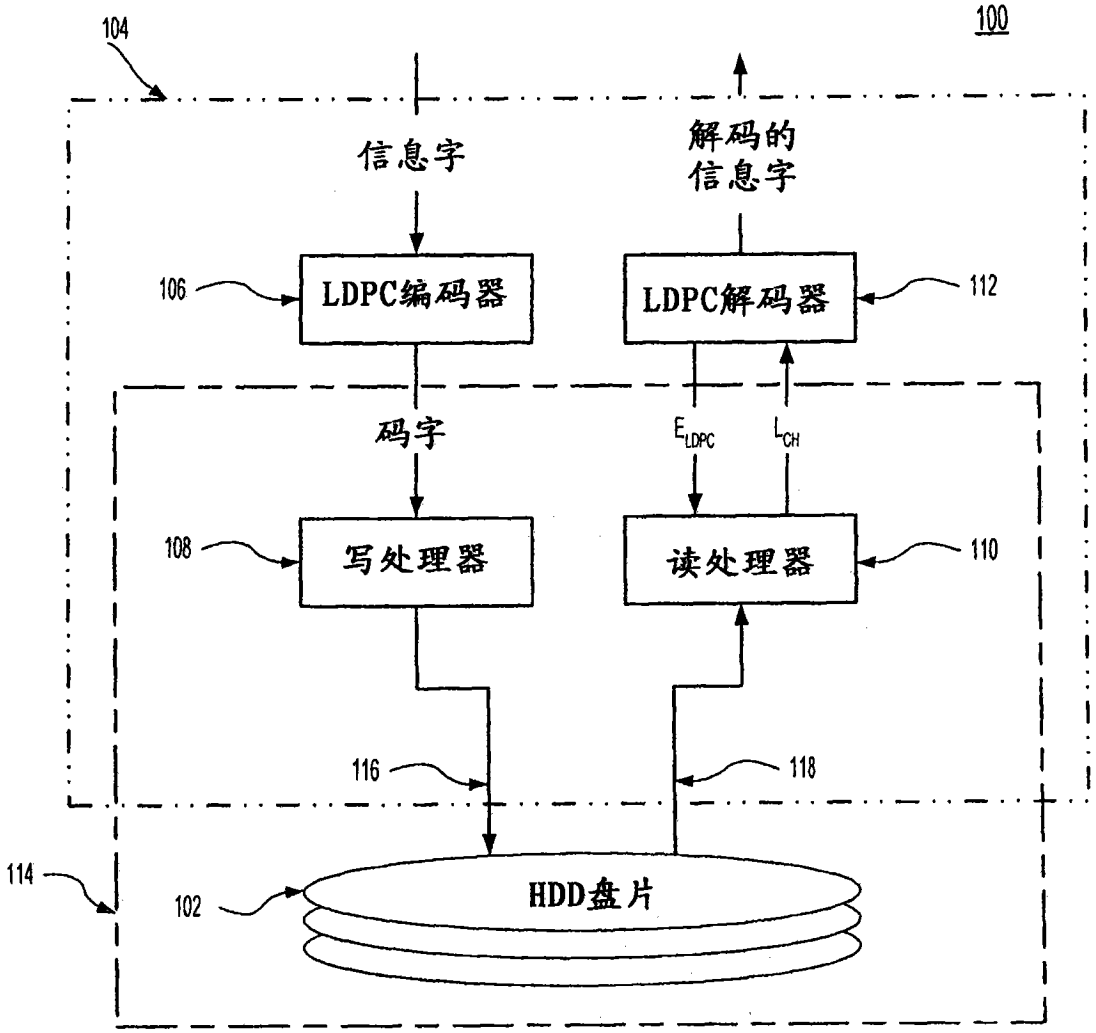

104
100
信息字
解码的
信息字
106
LDPC编码器
LDPC解码器
112
码字
E_{LDPC}
L_{CH}
108
写处理器
读处理器
110
116
118
114
102
HDD盘片

图 1 现有技术

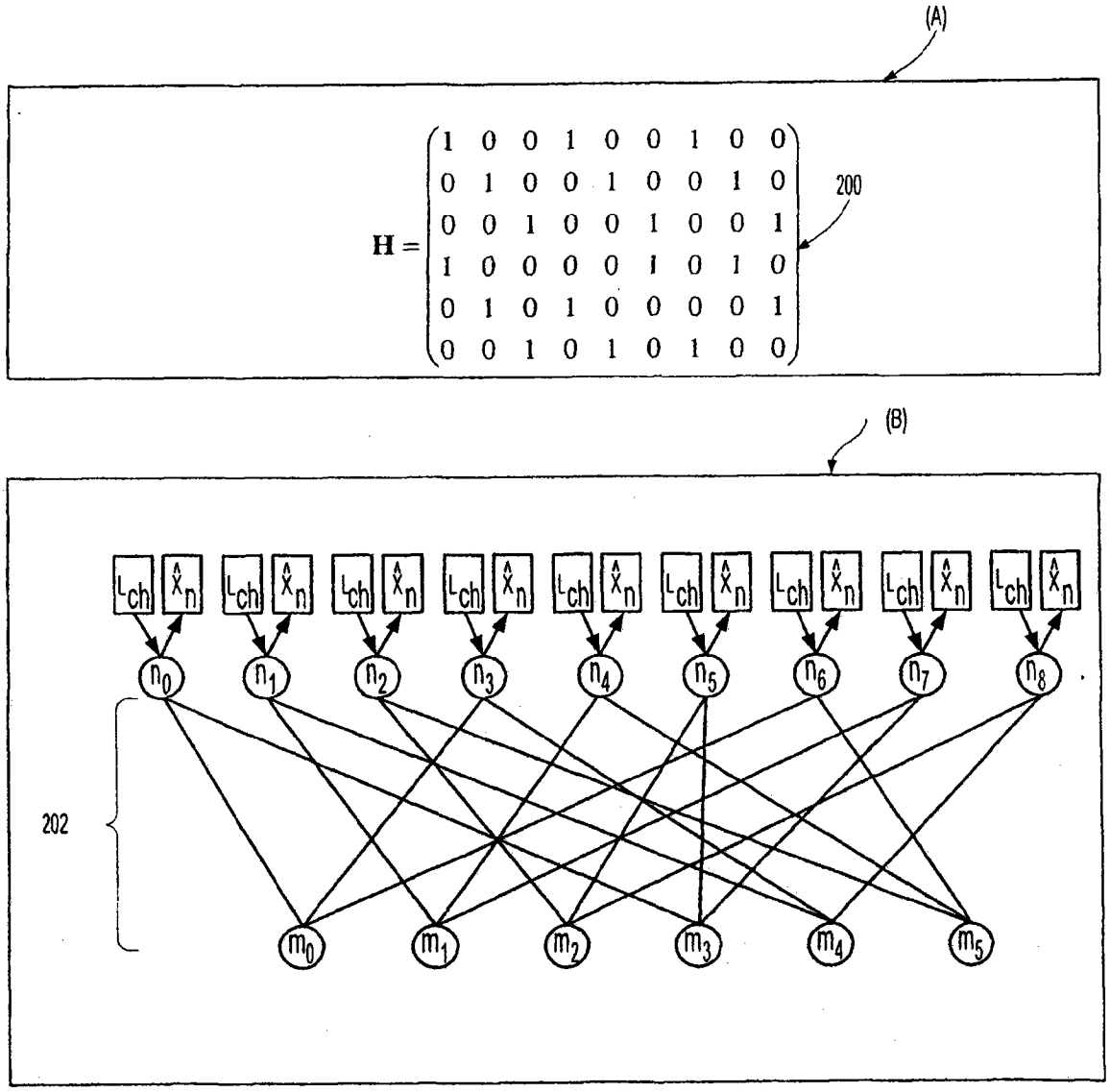

(A)
H =
1 0 0 1 0 0 1 0 0
0 1 0 0 1 0 0 1 0
0 0 1 0 0 1 0 0 1
1 0 0 0 0 1 0 1 0
0 1 0 1 0 0 0 0 1
0 0 1 0 1 0 1 0 0
200
(B)
Lch
n0 n1 n2 n3 n4 n5 n6 n7 n8
202
m0 m1 m2 m3 m4 m5

图 2 现有技术

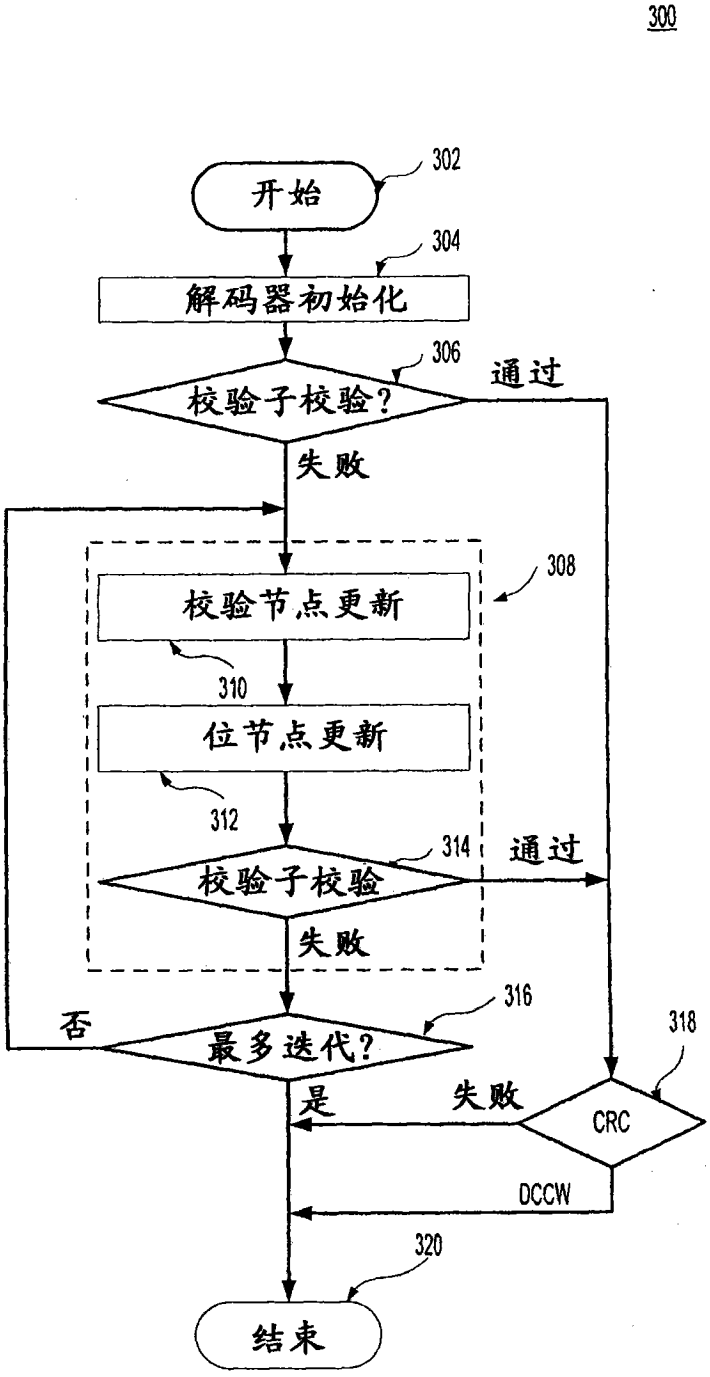

300
302
开始
304
解码器初始化
306
校验子校验?
通过
失败
308
校验节点更新
310
位节点更新
312
314
校验子校验
通过
失败
316
否
最多迭代?
318
是
失败
CRC
DCCW
320
结束

图 3 现有技术

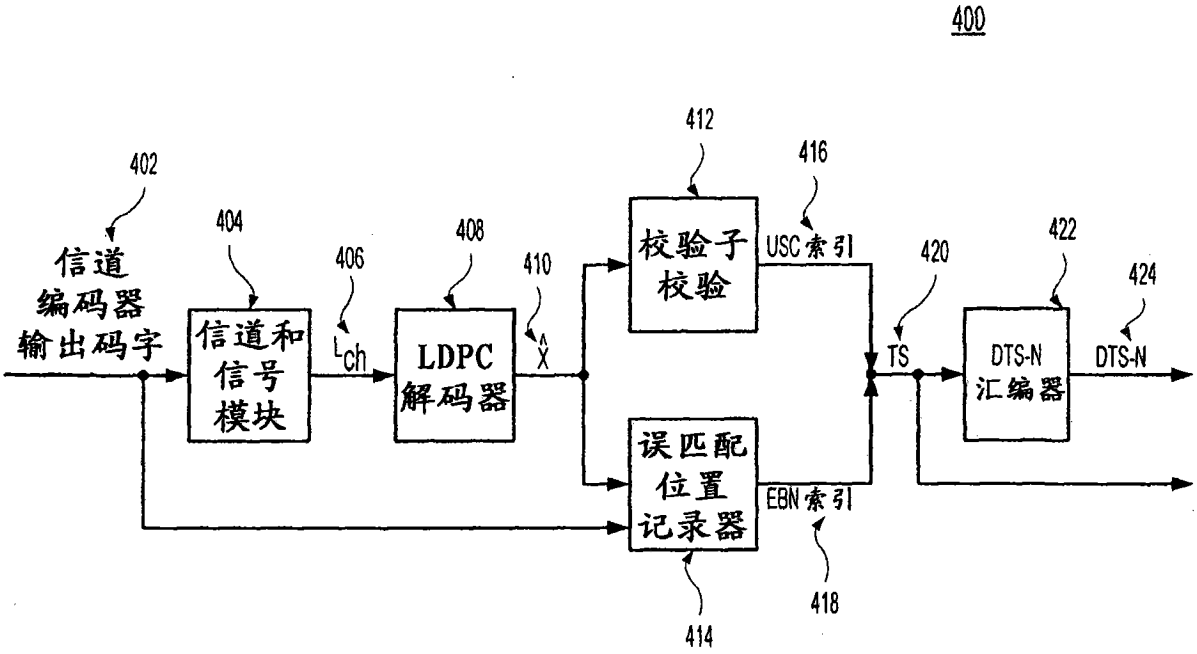

400
402
信道
编码器
输出码字
404
信道和
信号
模块
406
Lch
408
LDPC
解码器
410
X
412
校验子
校验
414
误匹配
位置
记录器
416
USC索引
418
EBN索引
420
TS
422
DTS-N
汇编器
424
DTS-N

图 4 现有技术

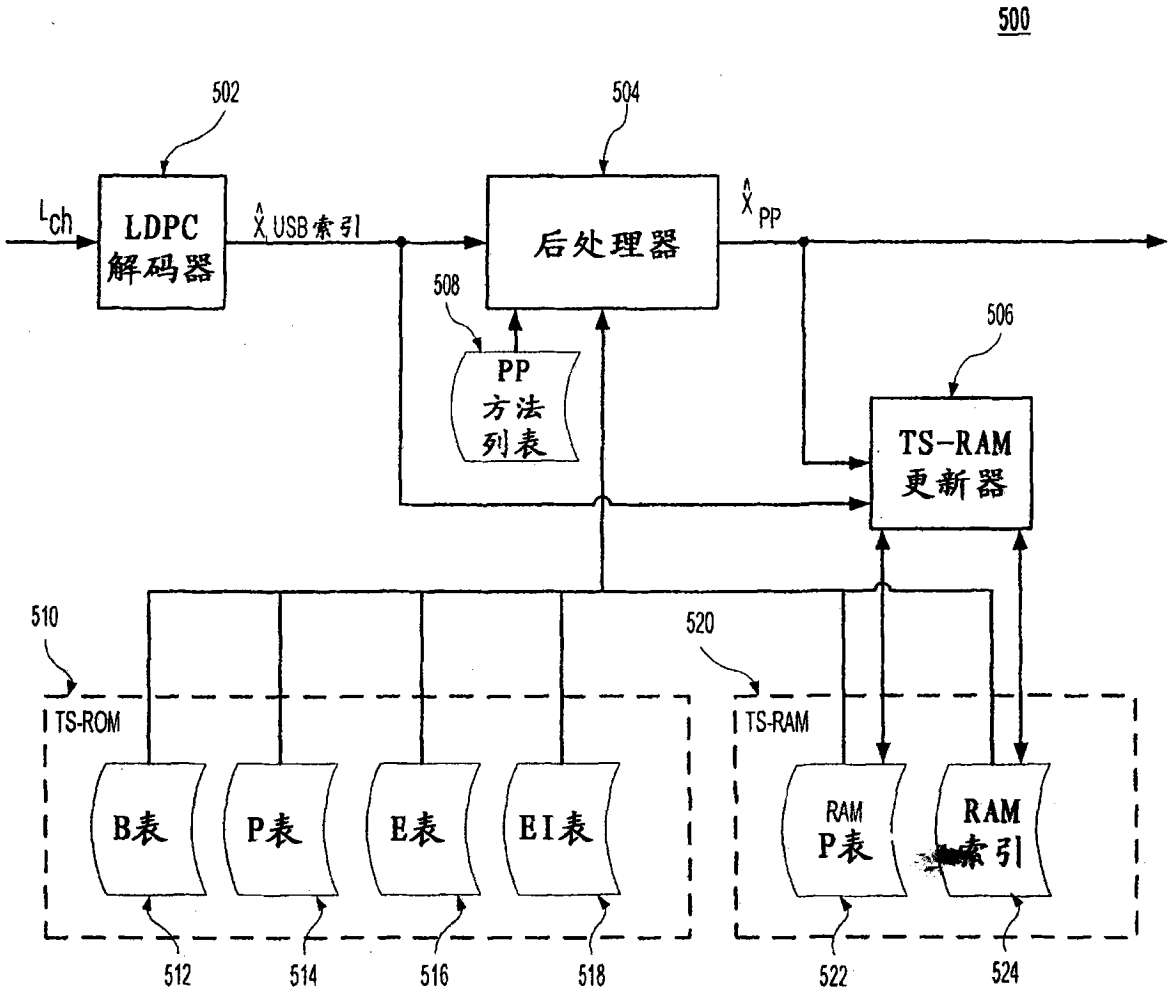

500
502
504
L_{ch}
LDPC
解码器
$\hat{X}$ USB索引
后处理器
$\hat{X}$ PP
508
PP
方法
列表
506
TS-RAM
更新器
510
520
TS-ROM
TS-RAM
B表
P表
E表
EI表
RAM
P表
RAM
索引
512
514
516
518
522
524

图 5

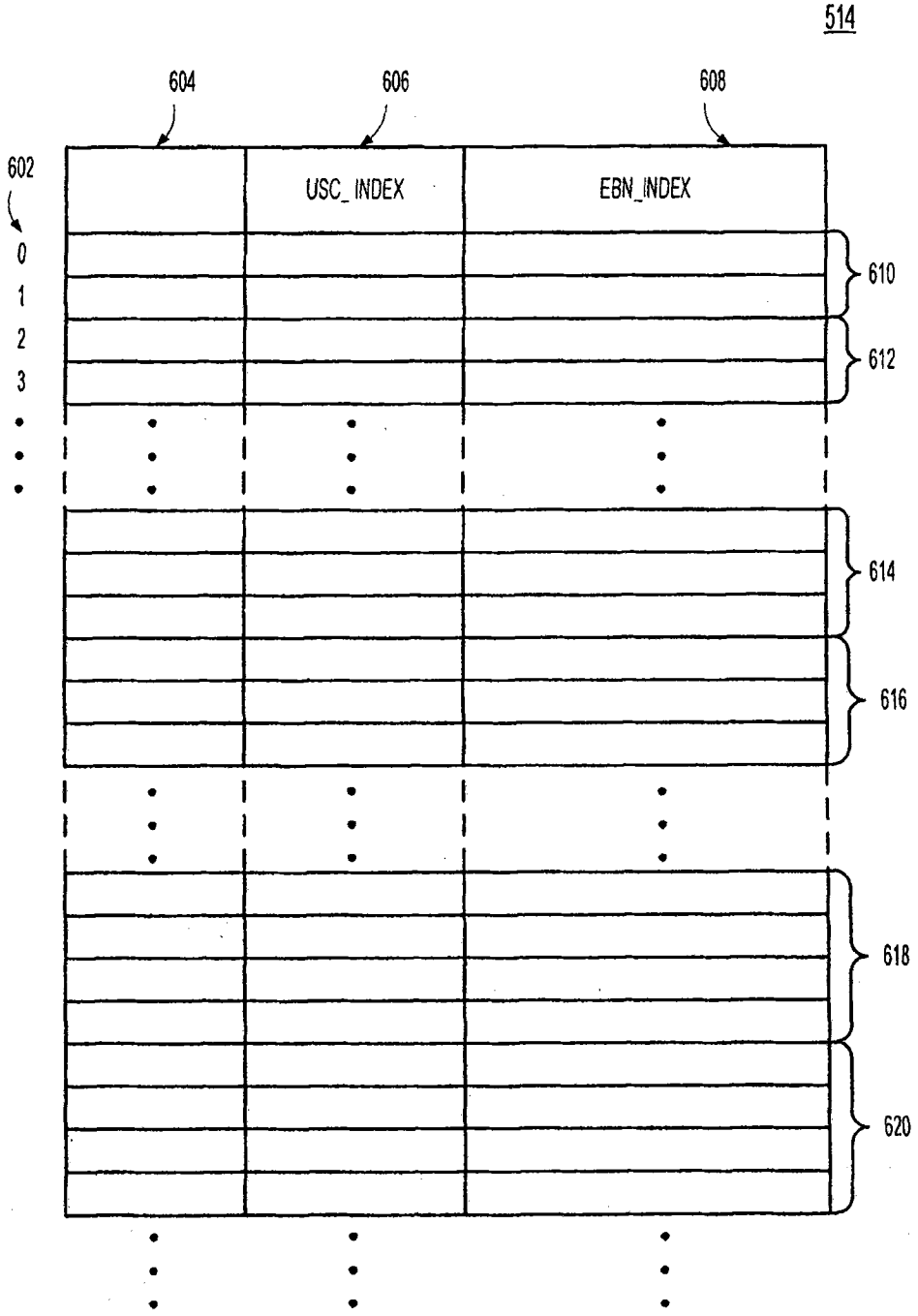

514
604
606
608
602
USC_INDEX
EBN_INDEX
0
1
2
3
610
612
614
616
618
620

图 6

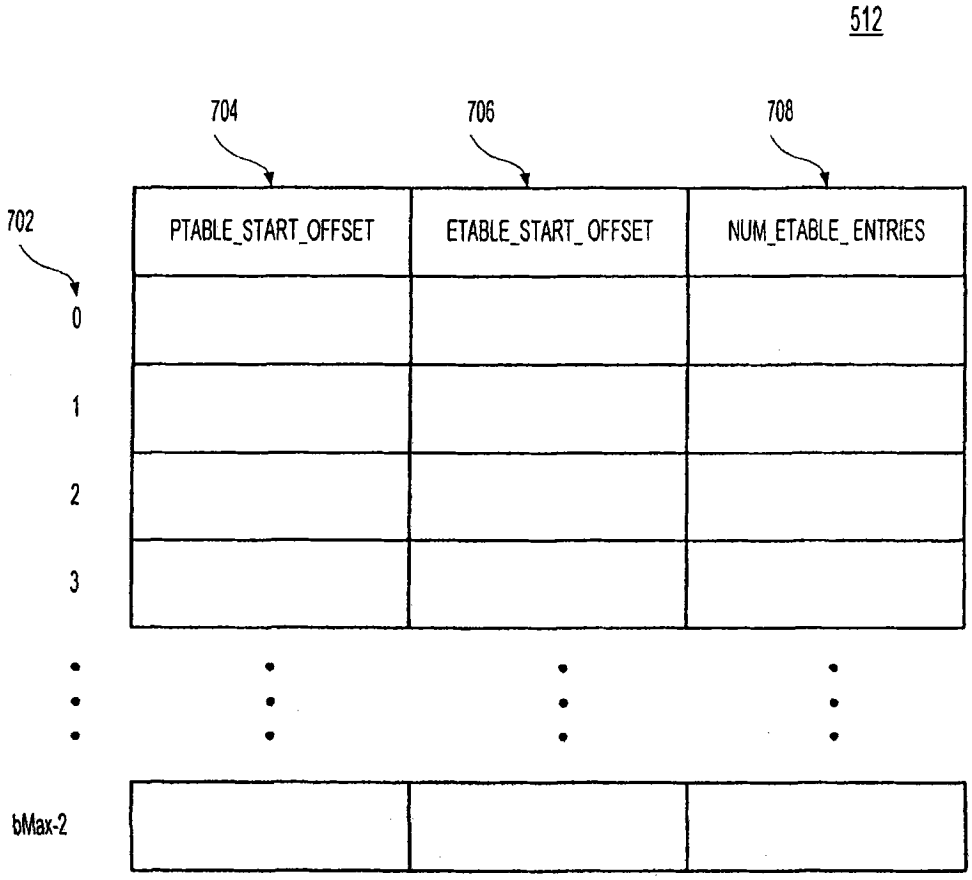

512
704
706
708
702
PTABLE_START_OFFSET
ETABLE_START_OFFSET
NUM_ETABLE_ENTRIES
0
1
2
3
bMax-2

图 7

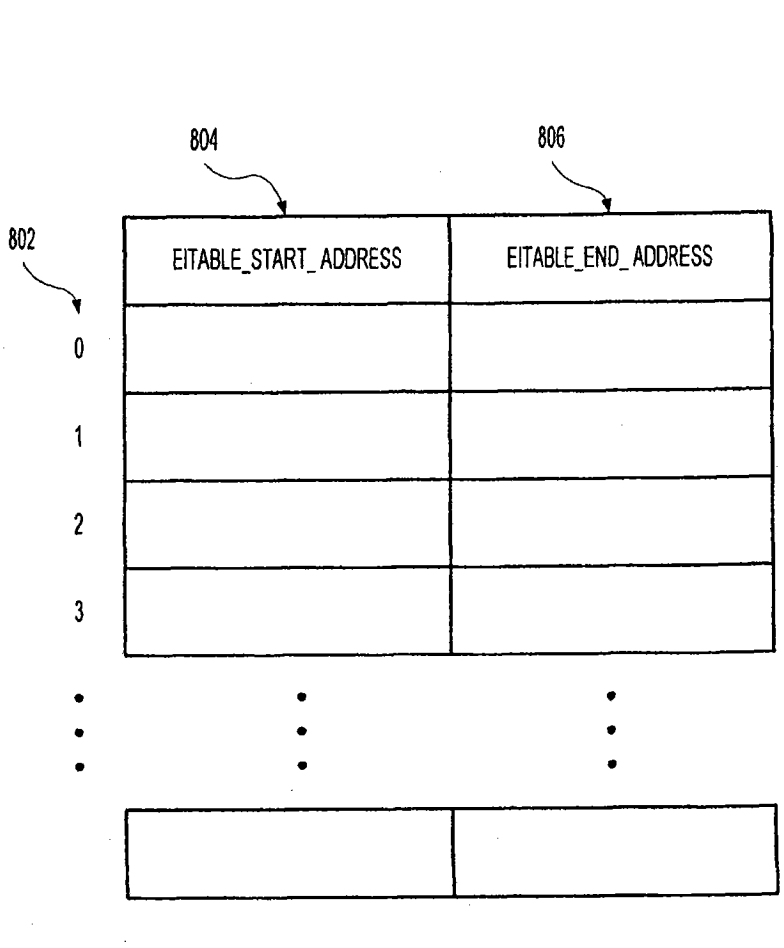

516
804
806
802
EITABLE_START_ADDRESS
EITABLE_END_ADDRESS
0
1
2
3

图 8

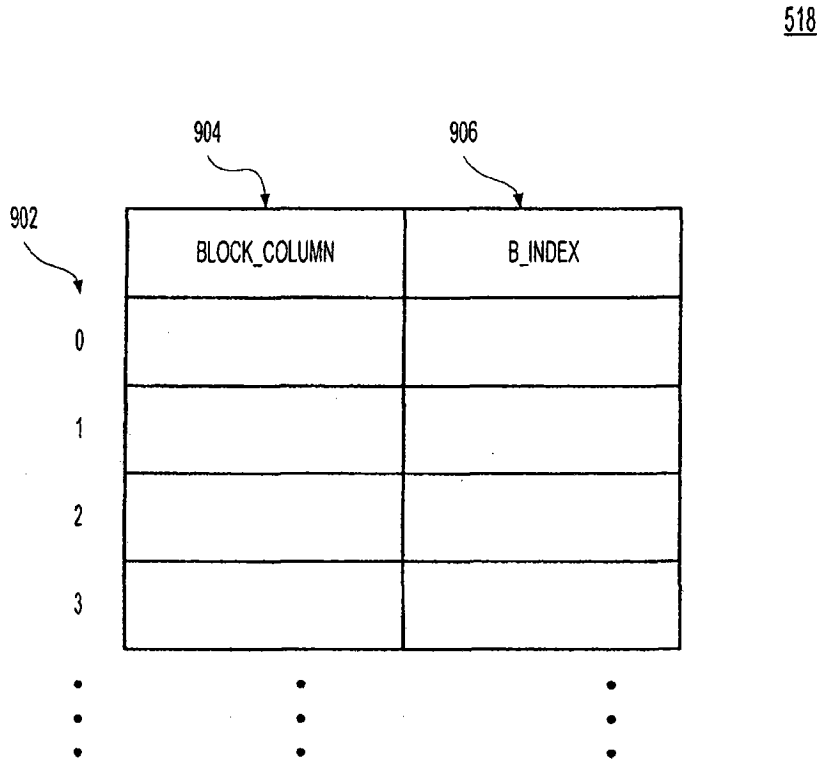


图 9

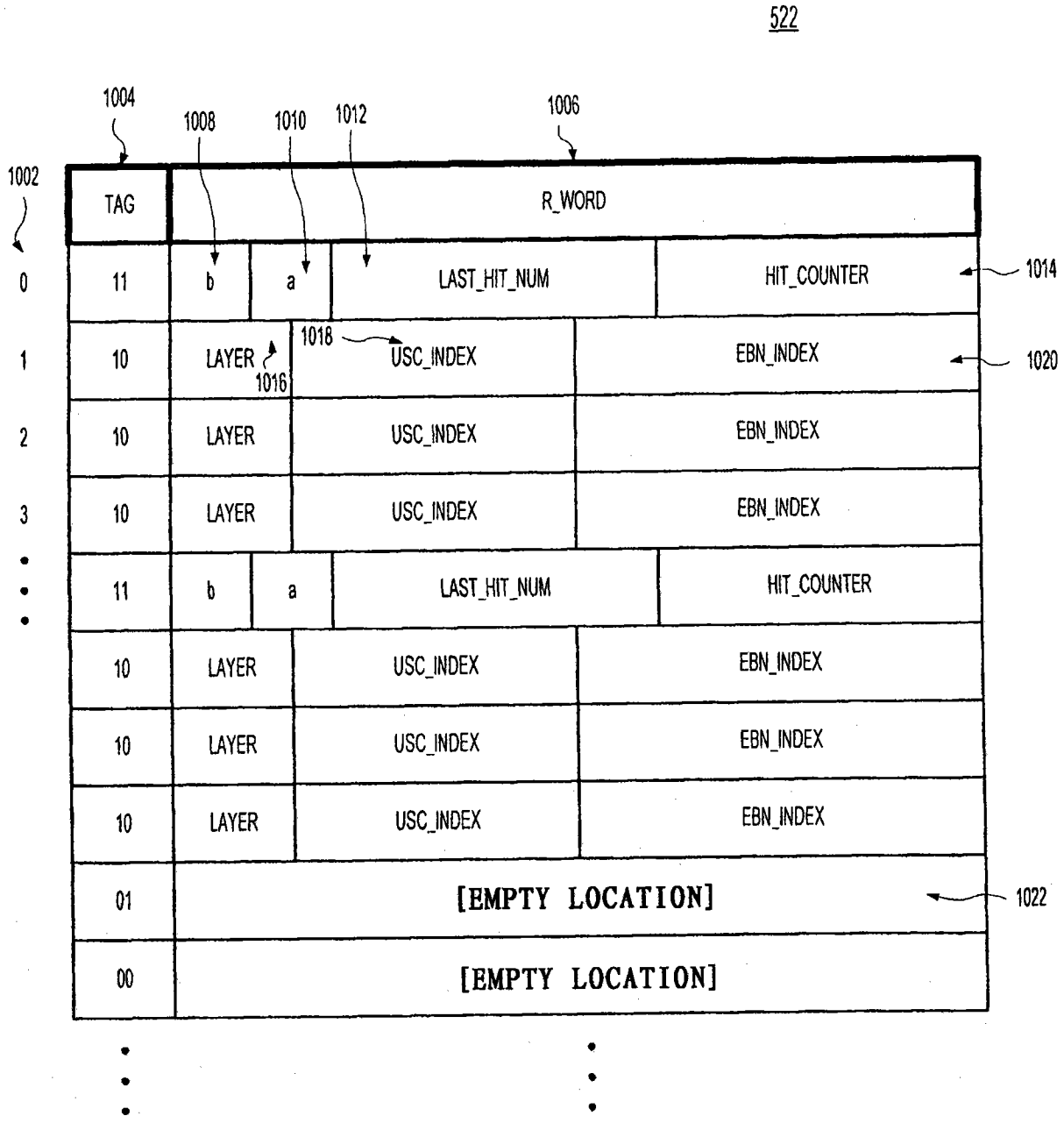

522
1004
1008
1010
1012
1006
1002
TAG
R_WORD
0
11
b
a
LAST_HIT_NUM
HIT_COUNTER
1014
1
10
LAYER
1016
1018
USC_INDEX
EBN_INDEX
1020
2
10
LAYER
USC_INDEX
EBN_INDEX
3
10
LAYER
USC_INDEX
EBN_INDEX
11
b
a
LAST_HIT_NUM
HIT_COUNTER
10
LAYER
USC_INDEX
EBN_INDEX
10
LAYER
USC_INDEX
EBN_INDEX
10
LAYER
USC_INDEX
EBN_INDEX
01
[EMPTY LOCATION]
1022
00
[EMPTY LOCATION]

图 10

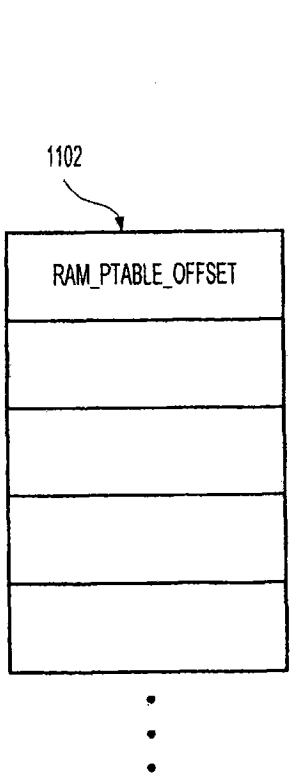


图 11

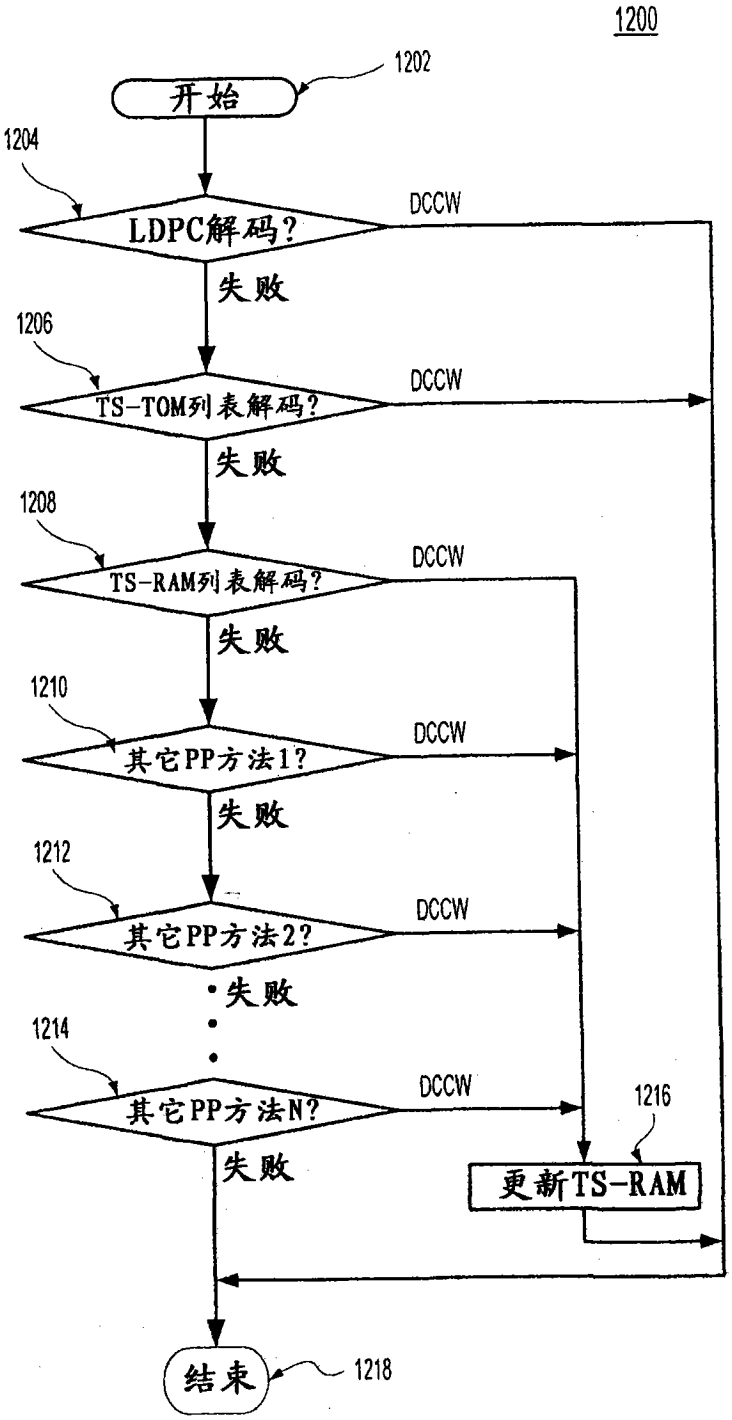

1200
1202
开始
1204
LDPC解码?
DCCW
失败
1206
TS-TOM列表解码?
DCCW
失败
1208
TS-RAM列表解码?
DCCW
失败
1210
其它PP方法1?
DCCW
失败
1212
其它PP方法2?
DCCW
失败
1214
其它PP方法N?
DCCW
失败
1216
更新TS-RAM
结束
1218

图 12

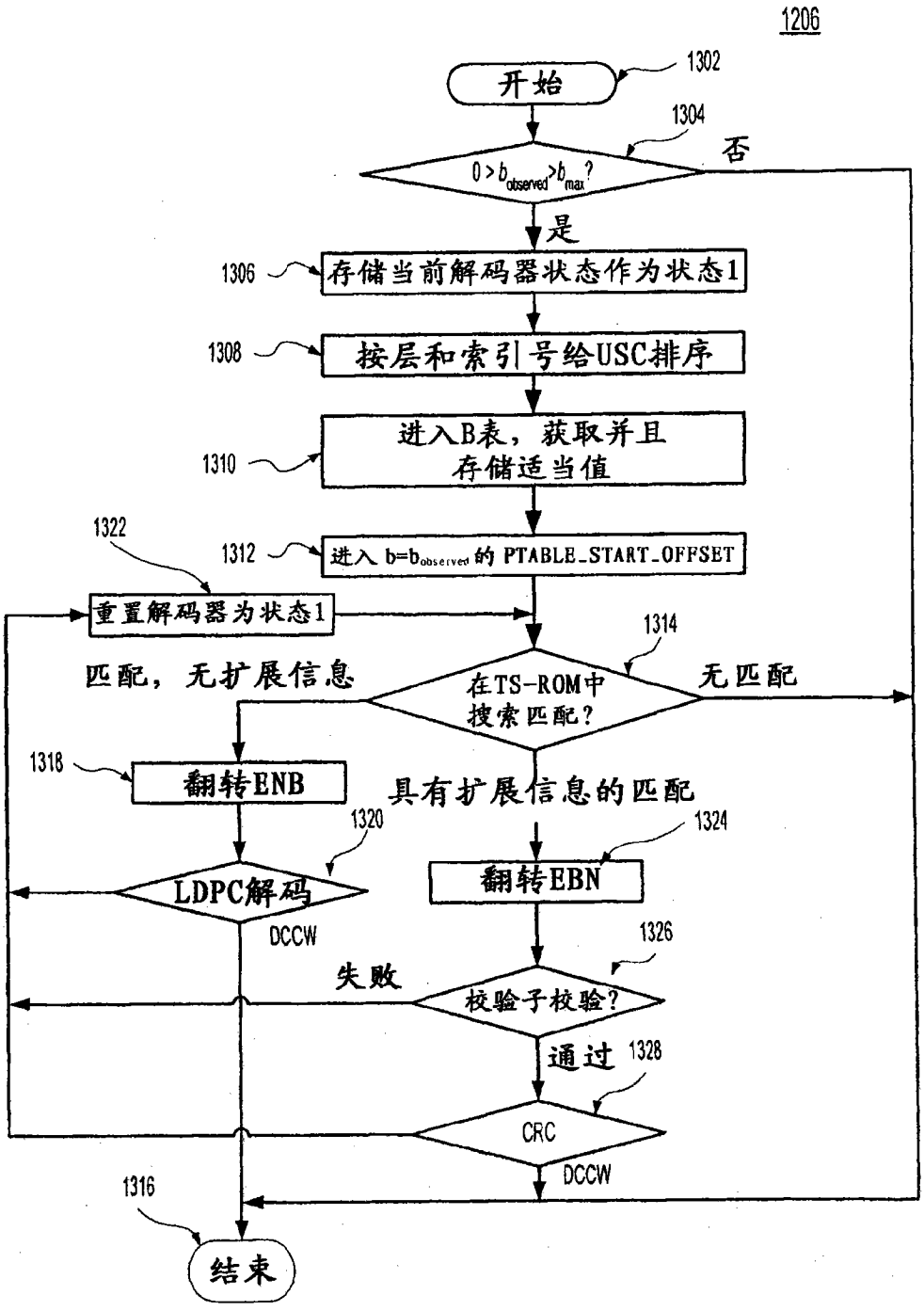

1206
开始
1302
1304
$0>b_{observed}>b_{max}$?
否
是
1306
存储当前解码器状态作为状态1
1308
按层和索引号给USC排序
1310
进入B表，获取并且
存储适当值
1322
1312
进入 $b=b_{observed}$ 的 PTABLE_START_OFFSET
重置解码器为状态1
1314
匹配，无扩展信息
在TS-ROM中
搜索匹配?
无匹配
1318
翻转ENB
具有扩展信息的匹配
1320
1324
LDPC解码
翻转EBN
DCCW
1326
失败
校验子校验?
通过
1328
CRC
DCCW
1316
结束

图 13

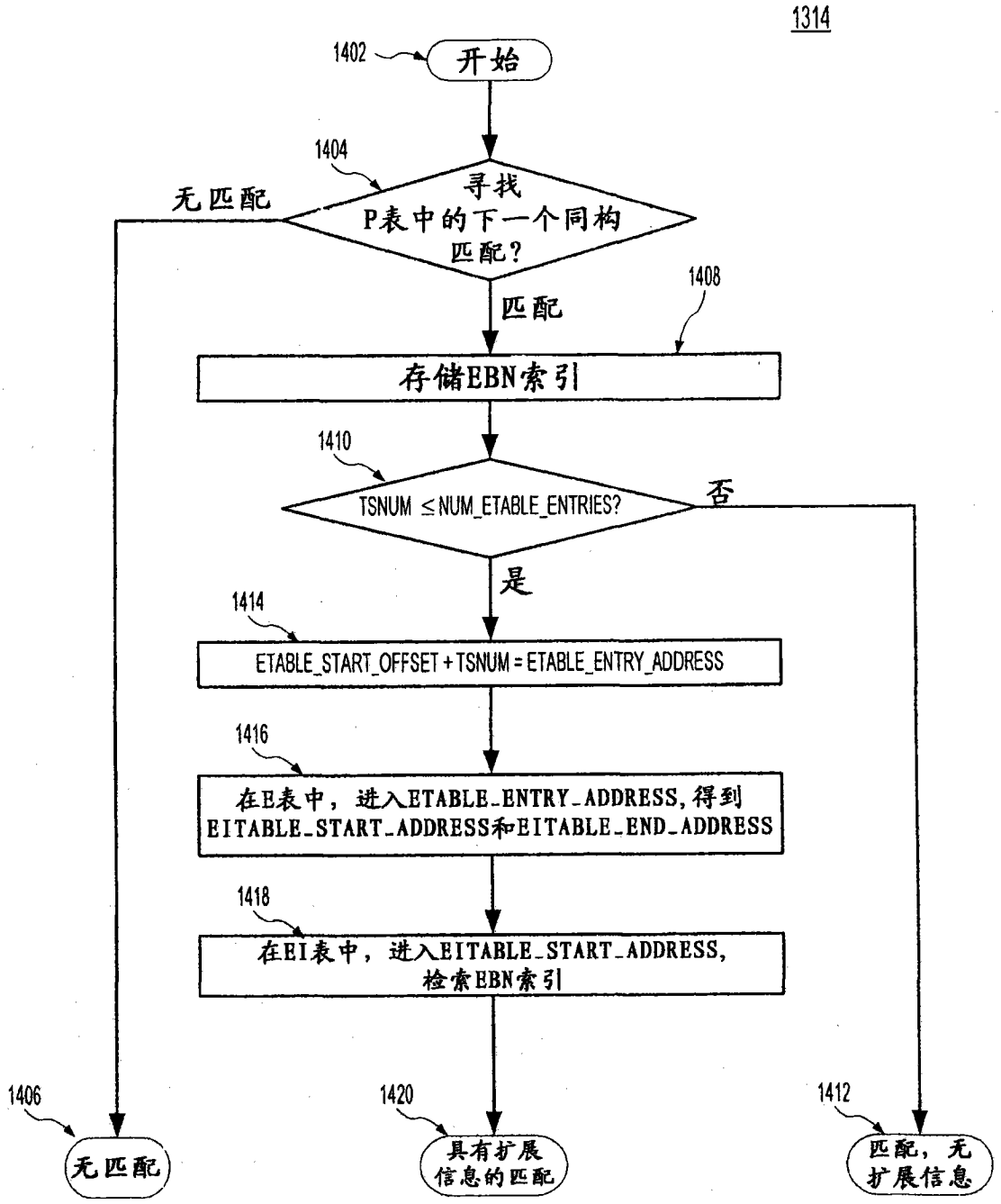

1314
1402
开始
1404
寻找
P表中的下一个同构
匹配?
无匹配
匹配
1408
存储EBN索引
1410
TSNUM ≤NUM_ETABLE_ENTRIES?
否
是
1414
ETABLE_START_OFFSET + TSNUM = ETABLE_ENTRY_ADDRESS
1416
在E表中，进入ETABLE_ENTRY_ADDRESS,得到
EITABLE_START_ADDRESS和EITABLE_END_ADDRESS
1418
在EI表中，进入EITABLE_START_ADDRESS,
检索EBN索引
1406
无匹配
1420
具有扩展
信息的匹配
1412
匹配，无
扩展信息

图 14

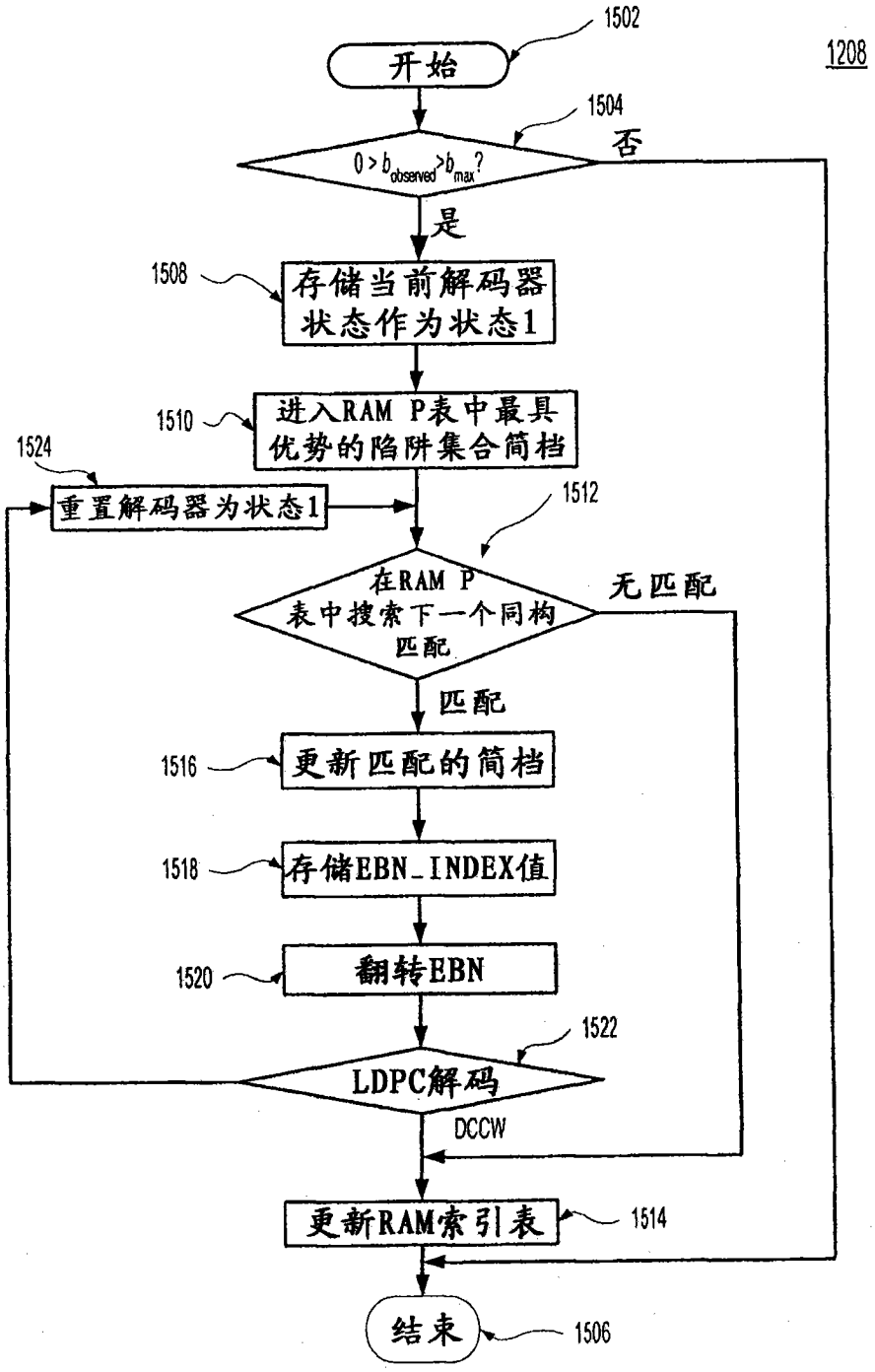

1208
1502
开始
1504
$0 > b_{observed} > b_{max}$?
否
是
1508
存储当前解码器状态作为状态1
1510
进入RAM P表中最具优势的陷阱集合简档
1524
重置解码器为状态1
1512
在RAM P表中搜索下一个同构匹配
无匹配
匹配
1516
更新匹配的简档
1518
存储EBN_INDEX值
1520
翻转EBN
1522
LDPC解码
DCCW
更新RAM索引表
1514
结束
1506

图 15

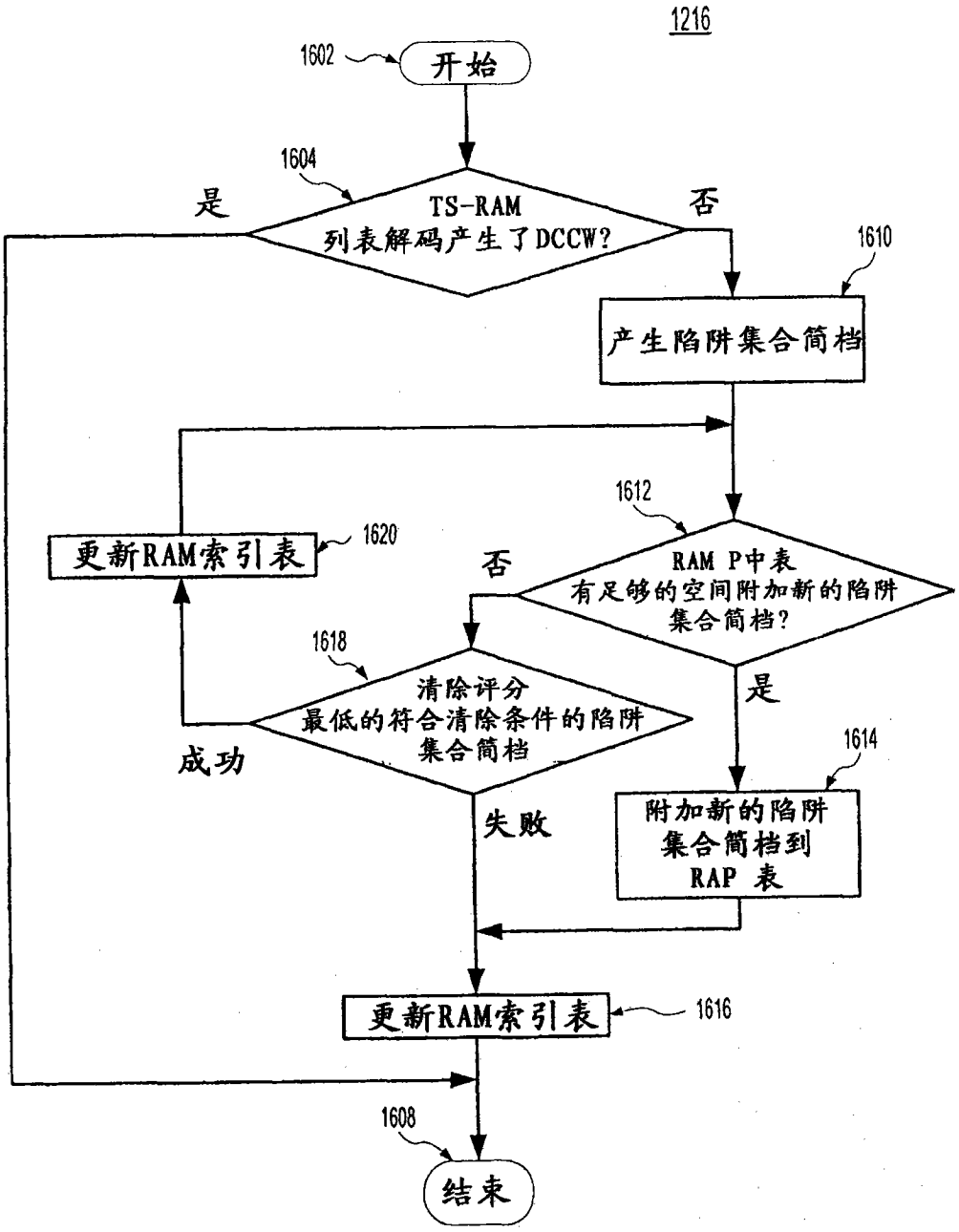

1216
1602
开始
1604
TS-RAM
列表解码产生了DCCW?
是
否
1610
产生陷阱集合简档
1612
RAM P中表
有足够的空间附加新的陷阱
集合简档?
否
是
1620
更新RAM索引表
1618
清除评分
最低的符合清除条件的陷阱
集合简档
成功
失败
1614
附加新的陷阱
集合简档到
RAP 表
更新RAM索引表
1616
1608
结束

图 16