[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 1/20 (2006.01)
H04L 25/49 (2006.01)
H03M 13/11 (2006.01)

[12] 发明专利说明书

专利号 ZL 200410092122.4

[45] 授权公告日 2007 年 9 月 5 日

[11] 授权公告号 CN 100336334C

[22] 申请日 2004.7.2
[21] 申请号 200410092122.4
[30] 优先权
[32] 2003.7.3 [33] US [31] 60/484,974
[73] 专利权人 直视集团公司
地址 美国加利福尼亚
[72] 发明人 孙凤文 穆斯塔法·埃勒兹 李琳南
[56] 参考文献
CN 1405981 A 2003.3.26
US 2003/0033575 A1 2003.2.13
WO 01/97387 A1 2001.12.20
Architecture – aware Low – density Parity – check Codes Mansour M M et al, IEEE International Symposium on Circuits and Systems, Vol. 2 of 5 2003
Deocoder – First Code Design Boutillon E et al, International Symposium on Turbo Codes and Related Topics 2000
低密度校验码 BP 译码算法中量化问题的研究 孙韶辉等,电子学报,第 31 卷第 2 期 2003
审查员 李晓莉
[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
代理人 李德山

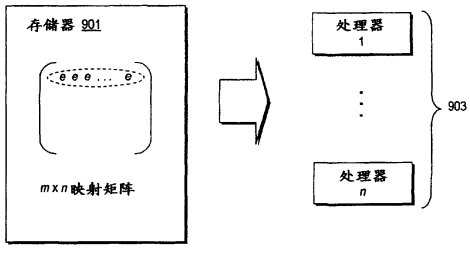
权利要求书 2 页 说明书 16 页 附图 10 页

[54] 发明名称

产生并行可解码的低密度奇偶校验码的方法和系统

[57] 摘要

提供一种用于有效解码低密度奇偶校验(LDPC)码的方法和解码装置。该方法包括：基于与所述低密度奇偶校验码相关联的奇偶校验矩阵构建 m×n 的映射矩阵，其中该映射矩阵满足允许集总存储器结构(901，1101)的多个并行可解码条件，该存储器结构可以由 n 个并行操作的处理器(903，1103)进行访问，m 和 n 是整数，n>1；和在存储器(901，1101)中存储该映射矩阵，以通过处理器(903，1103)对低密度奇偶校验码进行解码。



1. 一种用于支持对低密度奇偶校验码进行解码的方法，该方法包括:

基于与所述低密度奇偶校验码相关联的奇偶校验矩阵，通过将映射矩阵的值存储在存储器中来构建 m×n 的映射矩阵，其中该映射矩阵满足允许集总存储器结构（901，1101）的多个并行可解码条件，该存储器结构可以由 n 个并行操作的处理器（903，1103）进行访问，m 和 n 是整数，

其中所述并行可解码条件包括:

确定所述映射矩阵的同一行中没有任何两个不同项连接到相同的比特节点或相同的校验节点;

如果比特节点组 BG_i 表示连接到第 i 行边界的所有比特节点，则对于 $i \neq j$、BG_i 和 BG_j，验证 BG_i 和 BG_j 或者没有任何公共节点或者相同；和

如果校验节点组 CG_i 表示连接到第 i 行边界的所有校验节点，则对于 $i \neq j$、CG_i 和 CG_j，验证 CG_i 和 CG_j 或者没有任何公共节点或者相同；和

在存储器（901，1101）中存储该映射矩阵，以通过处理器（903，1103）对低密度奇偶校验码进行解码。

2. 根据权利要求 1 的方法，其中所述奇偶校验矩阵被等同置换成一个具有 n×n 密度对称矩阵的项的分块矩阵。

3. 根据权利要求 2 的方法，其中一个所述 n×n 密度对称矩阵项中的每一项是一个 m×m 密度对称矩阵，而其它所述 n×n 密度对称矩阵项被分解成分块密度对称子矩阵。

4. 根据权利要求 1 的方法，其中所述低密度奇偶校验码由根据一种信号星座调制的信号表示，该信号星座是 8-相移键控、16-正交幅度调制、16-振幅相移键控、32-振幅相移键控和正交相移键控中的一种。

5. 根据权利要求 4 的方法，其中所述信号通过卫星链路按照数字视频广播来传输。

6. 一种用于对低密度奇偶校验码进行解码的解码装置，该装置包括:

集总存储器（901，1101），配置成存储 m×n 的映射矩阵，该映射矩阵满足多个并行可解码条件，m 和 n 是整数，其中，基于与所述低密度奇偶校验码相关联的奇偶校验矩阵，通过将映射矩阵的值存储在存储器中来构建 m×n 的映射

矩阵；和

n 个并行操作的处理器（903，1103），该处理器（903，1103）访问集总存储器（901，1101）以对低密度奇偶校验码进行解码，

其中所述并行可解码条件包括：

确定所述映射矩阵的同一行中没有任何两个不同项连接到相同的比特节点或相同的校验节点；

如果比特节点组 BG_i 表示连接到第 i 行边界的所有比特节点，则对于 $i \neq j$、BG_i 和 BG_j，验证 BG_i 和 BG_j 或者没有任何公共节点或者相同；和

如果校验节点组 CG_i 表示连接到第 i 行边界的所有校验节点，则对于 $i \neq j$、CG_i 和 CG_j，验证 CG_i 和 CG_j 或者没有任何公共节点或者相同。

7. 根据权利要求 6 的装置，其中，存储器（901，1101）存储基于所述映射矩阵通过删除所述映射矩阵内的边界值构成的导出映射矩阵，其中所述导出映射矩阵被 m 个并行处理器（903，1103）使用，m 小于 n。

8. 根据权利要求 7 的装置，其中所述奇偶校验矩阵被等同置换成一个具有 $n \times n$ 密度对称矩阵的项的分块矩阵。

9. 根据权利要求 8 的装置，其中一个所述 $n \times n$ 密度对称矩阵项中的每一项是一个 $m \times m$ 密度对称矩阵，而其它所述 $n \times n$ 密度对称矩阵项被分解成分块密度对称子矩阵。

10. 根据权利要求 6 的装置，其中所述低密度奇偶校验码由根据一种信号星座调制的信号表示，该信号星座是 8-相移键控、16-正交幅度调制、16-振幅相移键控、32-振幅相移键控和正交相移键控中的一种。

11. 根据权利要求 10 的装置，其中所述信号通过卫星链路按照数字视频广播来传输。

产生并行可解码的低密度奇偶校验码的方法和系统

相关申请

本申请涉及并根据 35U.S.C.§119(e)要求在 2003 年 7 月 3 日申请的标题为“一般可并行解码的 LDPC 码”的序列号为 60/484，974 的美国临时专利申请的较早申请日的权益，其全文在此合并以作参考。

技术领域

本发明涉及通信系统，更确切地涉及编码系统。

背景技术

通信系统采用编码来确保通过有噪声通信信道的通信可靠性。这些通信信道显示出在某个信噪比（SNR）下按照每个码元的比特来表示的固定容量，定义一个理论上的上限（公知的香农极限）。结果，编码方案旨在获得接近此香农极限的速率。其中一类这样的接近香农极限的码就是低密度奇偶校验（LDPC）码。

通常，LDPC 码因为有许多缺点还没有被广泛使用。其中一个缺点是 LDPC 编码技术太复杂。用它的生成矩阵对一个 LDPC 码进行编码需要存储一个非常大的、非稀疏矩阵。还有，LDPC 码需要大的块来实现；因此，即使 LDPC 码的奇偶校验矩阵稀疏，存储这些矩阵也存在问题。

从实现的角度看要面临许多问题。例如，LDPC 码没有在实际应用中被广泛应用的一个重要原因是因为存储。在 LDPC 解码中的一个重要问题是存储器的组织。认识到存储器的容量越大、每比特的成本就越低的事实，就有了研究在 LDPC 解码器中对大量边界值允许有效集总（lumped）的存储器结构的码结构的动机。

同样的，在实施 LDPC 编码中的一个关键问题是怎样实现该解码器内几个处理引擎（节点）之间的连接网络。此外，解码处理中的计算负载，特别是校

验节点操作成了问题。

人们认识到，一般来说，对应固定数目的并行引擎的码，不能被重新配置以容纳不同数目的并行引擎，从而维护有效的存储器结构。因为不同的应用需要不同的解码速度，这种不变性就显示了其严重的缺陷。同样的，这种约束阻碍了利用和开发半导体性能的进步的能力。例如，当处理能力增加时，一个给定解码速度所需的并行引擎的数量会减少；然而，固定数目引擎的设计不允许使用的处理器数目上的直接降低。

所以，需要一种使用有效解码处理的 LDPC 通信系统。也需要将所需用于执行 LDPC 编码的存储器的数量减到最少。还需要能够更容易地适应将来技术发展的解码方案。

发明内容

通过本发明解决了这样和那样的需求，其中提供了一种用于结构化低密度奇偶校验（LDPC）码的解码方法。建立并行可解码代码的准则，以生成对应奇偶校验矩阵的一个映射矩阵。第一准则是映射矩阵的同一行中没有任何两个不同项连接到相同的比特节点或相同的校验节点。第二准则是，如果比特节点组 i，i=0,1,...m-1（表示为 BG_i），表示连接到第 i 行边界的所有比特节点，对于 $i\neq j$，BG_i 和 BG_j，保证它们或者没有任何公共节点或者相同。这个准则也加到校验节点上，从而校验节点组 i，i=0,1,...m-1（表示为 CG_i），表示连接到第 i 行边界的所有校验节点，对于 $i\neq j$，CG_i 和 CG_j。验证 CG_i 或者没有任何公共节点或者相同。这些准则导致产生利用 LDPC 码的并行解码结构并提供有效的存储器结构的 LDPC 码。为了对少于 n 个处理器（即，m 个处理器，使得 m 是 n 的整数部分（integer fraction））的有效并行处理保持这三个条件，利用分块矩阵 A_{ij} 构成一个奇偶校验矩阵，A_{ij} 是一个 $n\times n$ 密度对称矩阵。该分块矩阵在行和列置换下被分解成一个有 B_{ij} 项的矩阵，这是 $m\times m$ 密度对称矩阵。根据二分图，通过删除该映射矩阵中的一些边界可得到导出码。在上述安排下，可以使用 m 个并行引擎，从而增强了实现的灵活性。

依照本发明一个实施例的一方面，披露了一种用于支持对低密度奇偶校验码进行解码的方法，该方法包括：基于与所述低密度奇偶校验码相关联的奇偶校验

矩阵构建 m×n 的映射矩阵，其中该映射矩阵满足允许集总存储器结构的多个并行可解码条件，该存储器结构可以由 n 个并行操作的处理器进行访问，m 和 n 是整数，n>1；和在存储器中存储该映射矩阵，以通过处理器对低密度奇偶校验码进行解码。

依照本发明一个实施例的另一方面，披露了一种用于对低密度奇偶校验码进行解码的解码装置，该装置包括：存储器，配置成存储 m×n 的映射矩阵，该映射矩阵满足用于允许集总存储器结构的多个并行可解码条件，m 和 n 是整数，n>1；和 n 个并行操作的处理器，该处理器访问存储器以对低密度奇偶校验码进行解码。

依照本发明一个实施例的另一方面，披露了一种用于支持对低密度奇偶校验码进行解码的方法，该方法包括：基于对应于低密度奇偶校验码的奇偶校验矩阵存储 m×n 的映射矩阵，m 和 n 是整数，n>1；确定所述映射矩阵的同一行中没有任何两个不同项连接到相同的比特节点或相同的校验节点；如果比特节点组 BG_i，表示连接到第 i 行边界的所有比特节点，则对于 $i\neq j$，BG_i 和 BG_j，验证 BG_i 或者没有任何公共节点或者相同；如果校验节点组 CG_i，表示连接到第 i 行边界的所有校验节点，则对于 $i\neq j$，CG_i 和 CG_j，验证 CG_i 或者没有任何公共节点或者相同；和输出所述映射矩阵以存储在存储器中，该存储器可以由 n 个并行操作的处理器访问，以依照所述存储的映射矩阵对低密度奇偶校验码解码。

通过列举许多特定实施例和实施方案，包括期望实现本发明的最佳模式，本发明的其它的方面、特征、和优势很容易地从下面的详细描述中体现。本发明还可以有不背离本发明精神和范围的其它的和不同的实施例，它的一些细节可以在多方面进行修改。因此，认为附图和描述实际上是阐述性的，不是限制性的。

附图说明

本发明是通过例子的方式进行说明的，并且不是用各附图中的图限制的，以及图中相同标号代表相同部件，其中：

图 1 示出了一个依照本发明实施例的配置成采用低密度奇偶校验（LDPC）码的通信系统；

图 2 示出了图 1 的系统中的示范性发射机；

图 3 示出了图 1 的系统中的示范性接收机；

图 4 示出了一个依照本发明实施例的稀疏奇偶校验矩阵;

图 5 示出了图 4 的矩阵的 LDPC 码的二分图;

图 6 是依照本发明实施例的图 3 的 LDPC 解码器的操作流程图;

图 7 示出了依照本发明实施例的 LDPC 解码器的结构;

图 8 是依照本发明不同实施例的生成一个用于有效并行解码的映射矩阵的处理流程图;

图 9 示出了一个依照本发明实施例的具有存储用于支持并行解码的边界值的映射矩阵的存储器的 LDPC 解码器;

图 10 是依照本发明不同实施例的生成一个导出映射矩阵以提供并行解码的处理流程图;

图 11 示出了一个依照本发明实施例的具有储存用于支持并行解码的边界值的导出映射矩阵的存储器的 LDPC 解码器; 和

图 12 示出了一个依照本发明实施例的能够进行 LDPC 码的编码和解码的计算机系统。

具体实施方式

描述一个用于对结构化的低密度奇偶校验（LDPC）码进行有效解码的系统、方法、和软件。在下面的描述中，为了说明的目的，阐述了许多特定细节以提供对本发明的彻底理解。然而，显然的，对本领域技术人员来说可以不采用这些特定细节或采用等同配置来实现本发明。另外，以方块图显示众所周知的结构和设备，目的是避免对本发明不必要的混淆。

图 1 示出了一个依照本发明实施例配置成采用低密度奇偶校验（LDPC）码的通信系统。数字通信系统 100 包括产生信号波形的发射机 101，信号通过通信信道 103 到达接收机 105。在该离散通信系统 100 中，发射机 101 具有产生一组离散可能消息的消息源，每个可能消息有一个相应的信号波形。这些信号波形被通信信道 103 削弱或相反改变。为了对付噪音信道 103，采用 LDPC 码。

由发射机 101 生成的 LDPC 码能够高速实现而不会引起任何性能损失。这样的 LDPC 码有一个可并行解码算法(不同于特播码),该算法有利地包括如加法、比较、查表的简单操作。使用该并行结构以最小化存储器需求，并提高在使用不同数量的并行引擎（如处理器）方面的灵活性; 后面将参照图 7-11 对此做更

详细的说明。此外，仔细设计的LDPC码不会出现利害范围内的误差底限的任何迹象。

依照本发明的一个实施例，发射机101基于奇偶校验矩阵生成LDPC码，这些矩阵促进了解码期间有效的存储器访问，以与接收机105进行通信。

图2示出了图1的系统中的示范性发射机。发射机200装备有LDPC编码器203，接受从信息源201的输入，并输出适于在接收机105进行纠错处理的冗余编码流。信息源201从一个离散字母表X生成k个信号，用奇偶校验矩阵指定LDPC码。

调制器205将来自编码器203的编码消息映射成信号波形，传送到发射天线207，天线207通过通信信道103发射这些波形。因此，该编码信息被调制和分发到发射天线207。来自发射天线207的发射传播到一个接收机，如下所述。

图3示出了图1的系统中的示范接收机。在接收端，接收机300包括解调器301，对从发射机200接收的信号进行解调。这些在接收天线303收到的信号用于解调。解调后，将接收的信号传送到解码器305，解码器305与比特矩阵生成器307一起通过生成消息X′试图重新构成原始的源消息。在解码处理过程中比特矩阵生成器307可以与解码器305来回（反复地）交换信息。这些解码方法在共同未决的标题为"用于低密度奇偶校验（LDPC）解码器中的路由的方法和系统"的申请中有更详细的描述，该申请于2003年7月3日申请（序列号为10/613824；代理案卷号PD-203009），其整个内容在此合并。为了理解本发明所具有的优势，研究LDPC码是怎样生成的是有用的，如对图4的讨论。

图4示出了一个依照本发明实施例的稀疏奇偶校验矩阵。LDPC码为具有稀疏奇偶校验矩阵 $H_{(n-k)xn}$ 的长，线性块码。典型地，块长n是从数千到数万个比特。例如，图4显示了一个用于长度n=8和比率为1/2的LDPC码的奇偶校验矩阵。同样的码可以通过图5的二分图等同表示。

图5示出了图4的LDPC码矩阵的二分图。奇偶校验等式表示了对于每个校验点，所有相邻比特节点的总和（通过GF（伽罗瓦域）（2））等于0。如该图所示，比特节点占据图左侧，并依照预定关系与一个或多个校验节点关联。例如，对应于校验节点 m_1，对于比特节点存在下面的表达式 $n_1+n_4+n_5+n_8=0$。

LDPC解码器305被认为是一个消息通过解码器，借此解码器305旨在寻找比特节点的值。为了实现这个任务，比特节点和校验节点相互之间反复通信，

该通信的特性将在下面进行描述。

从校验节点到比特节点，基于来自于其它相邻比特节点的信息，每个校验节点给该相邻比特节点提供关于该比特节点的值的估算（“意见”）。例如，在上述例子中，如果对 m_1、n_4、n_5和 n_8的总和“像”0，则 m_1将指示 n_1，n_1的值认为是 0（既然 $n_1+n_4+n_5+n_8=0$）；否则 m_1指示 n_1，n_1的值认为是 1。另外对软判决解码增加了一个可靠性测量。

从比特节点到校验节点，每个比特节点将一个基于来自其它相邻校验节点的反馈的关于它自己的值的估算传递到一个相邻校验节点。在上述例子中，n_1只有两个相邻校验节点 m_1和 m_3。如果从 m_3到 n_1的反馈指示 n_1的值可能是 0，则 n_1将通知 m_1，n_1自己的估算值是 0。在比特节点有两个以上的相邻校验节点的情况下，在将该判定报告给与之通信的校验节点之前，比特节点对来自其它相邻节点的反馈执行一个多数判决（软判断）。上述处理重复进行，直到所有比特节点被认为是正确的（即，所有奇偶校验等式成立），或直到达到了一个预定的最大重复次数，从而宣告解码失败。

图 6 是依照本发明实施例的图 3 的 LDPC 解码器的操作流程图。认识到 LDPC 的一个最突出的优点是并行解码的潜力。通过查看 LDPC 码的二分图和解码算法可以看出这是显然的。特别地，既然在处理过程中没有中间数据进行交换，当处理比特节点（或校验节点）时，所有比特节点（或校验节点）能够被同时处理，这是很清楚的。对每个比特节点 i，有一个接收值 u_i，表示特定比特的对数似然比是逻辑 0 而非逻辑 1，它被称作“信道值”。通过举例，对解码 LDPC 码的消息分析算法作解释。

在步骤 601 中，对解码器 305 进行初始化，在二分图（图 5）中给每个边界分配一个值。这个值等于与该边界连接的比特节点所关联的信道值。接下来，通过步骤 603 更新校验码。也就是说，在校验节点处理的最后，所有边界值被更新。令 $e_1,e_2,...,e_v$表示连接到一个特定校验节点的所有边界的所有值，则更新的边界值如下所示：

$e_i=g(e_1,e_2,...,e_{i-1},e_b...,e_v)$ 其中 $i=1,2,...v$,

$g(a,b)=\ln\frac{1+e^{a+b}}{e^a+e^b}$ 。函数 $g(a,b,c)=g(g(a,b),c)$可以递归计算。

所有校验节点经过处理后，比特节点用更新的边界值进行处理，如步骤 605 所示。比特节点处理的目的是进一步更新边界值。假设 $e_1,e_2,...,e_v$表示连接到该特

定比特节点的所有边界的边界值，按照如下方式更新边界值:

$e_1=e_1+e_2+...,e_{t-1}+e_t+...e_v$ 。

通过步骤 607，该处理输出一个后验概率信息，并确定奇偶校验等式是否成立，或反复次数是否已经达到了一个可配置的最大数目（步骤 609）。如果这些等式不成立，则重复进行步骤 603-607。通过步骤 611，在最后一次重复中，与一个比特节点相关联的所有边界值被加在一起，并进一步和信道值相加，以形成最后的判决度量。

因为 LDPC 解码器 305 可以实现为高并行系统，存在两种实现校验节点和比特节点之间的互连的方法:（1）一个全并行方法，和（2）一个部分并行方法。在全并行结构中，所有节点和它们的互连是物理实现的。这种结构的优势是速度快。然而，该全并行结构在实现所有节点和它们的连接上会更复杂。所以，需要一个小的块尺寸来减少全并行结构的复杂性。

第二种实现 LDPC 码的方法是仅仅物理实现节点总数的一个子集，仅用这些有限数目的"物理"节点来处理码的所有"功能"节点。即使 LDPC 解码器 305 操作能够做得极其简单且能够并行执行，该设计中的进一步的挑战涉及怎样建立"随机"分配的比特节点和校验节点之间的通信。

图 7 示出了依照本发明实施例的 LDPC 解码器的组件的框图。解码器 305 通过利用并行处理器执行图 7 的解码处理。如图所示，解码器 305 包括比特节点处理器 701 和校验节点处理器 703，它们相应地执行比特节点和校验节点操作。比特节点或校验节点更新的中间边界值缓存在边界存储器 705 中。同样的，在一个比特节点处理过程中形成新边界值的部分和也被缓存，通常与处理器本地存储。一个信道存储器 707 保留来自信道的可靠性信息。如附图中所示，信道可靠性存储器 707 仅用于比特节点处理中。

解码器 305 的设计需要是许多工程上的折衷：并行处理器的数目与解码速度，以及存储器容量。一种极端是，一个单独的处理器能够被用来顺序从一个节点到另一个节点对全部码进行解码。然而，由于大量的节点和对一个典型的好 LDPC 码的多次重复，这种方法是不实用的。另一种极端是使用在数量上和节点数量相同的处理器。这样可以得到最高的解码速度，但遗憾的是成本也最高。在这两种极端方案之间设计实用的解码器。

明显地，在解码速度和比特节点和/或校验节点处理器数目之间存在一个折

衷。虽然所示的比特节点处理器 701 和校验节点处理器 703 是分离组件，期望比特节点处理器 701 能够与校验节点处理器 703 组合，使它们能够共享部分数据通路。可选地，在处理之前通过对边界值执行转换，比特节点处理器 701 可以和校验节点处理器 703 相同。同样，值得注意的，比特节点处理器 701 的数量不需要与校验节点处理器 703 的数量相同。

同样，一个有效结构不能随意处理并行处理引擎的数量。期望在并行操作数量上具有一些自由度以提供适应技术发展进步的实现灵活性。例如，如果一个码被设计为有 360 个并行引擎，随着集成电路技术的改进，电路能够最快以两倍速度操作。因此期望重新配置相同码以具有 180 个并行引擎，同时仍保持相同的解码结构。

关于存储器结构，LDPC 解码器 305 将边界存储器 705 放置在每个处理器 701、703 附近。对于存储器 705 可以利用寄存器或随机存取存储器（RAM）。对于长码，用寄存器比用 RAM 昂贵得多。大家知道，对于 RAM 来说，随着 RAM 尺寸的增加每个比特的成本降低。所以，将尽可能多的 RAM 集总在一起是有利的，这样所有边界值在功能上被集总到一个 RAM 上。

大家知道，一个任意的 LDPC 码不能有效地支持图 7 的解码器结构。通过检查标准 RAM 结构就能很显然的看出来。注意到以下情况。一个 RAM 是一个有预定宽度和深度的存储器，能够按照行的规则来读和写。这意味着在一个访问循环中，只有相同行内的值可以被访问。“随机”简单的含义就是可以以任意顺序从一个访问循环到下一个访问循环来访问行。所以，从效率的观点出发，显然在任何给定的访问循环，每个比特节点处理器 701（或当在校验节点侧进行处理时就是校验节点处理器 703）必须被提供以相同数量的边界值输入。从而，在任何给定时刻，一个有效解码器结构不应当使处理引擎数据量不足。

而且，期望一个特定比特节点（或校验节点）在任一给定时间有一个专用的处理器。换句话说，在处理器完成更新与该一个比特节点相关联的边界之前，处理器不会切换到另一个比特节点（或校验节点）。

本发明引入了可并行解码的码的概念以说明图 7 的结构的各种特性，如下所述。

图 8 是依照本发明不同实施例，生成一个用于有效并行解码的映射矩阵的处理流程图。可并行解码的码被定义为展示一个有效的存储器结构。如果存在

一个一一映射，将奇偶校验矩阵的非零项映射到一个满足许多预定特性的 m×n 矩阵（映射矩阵）上，一个 LDPC 码可以用具有集总存储器的 n 个并行引擎进行解码。用于生成该映射矩阵的处理过程在下面进行解释。

在步骤 801 中，基于奇偶校验矩阵生成一个映射矩阵。然后校验该映射矩阵以保证用于有效并行解码的各种条件或特性。通过步骤 803 校验该矩阵，使得映射矩阵的同一行中没有任何两个不同项连接到相同的比特节点或相同的校验节点。令比特节点组 i，i=0,1,...m-1，表示为 BG_i，为连接到第 i 行边界的所有比特节点，对于 i≠j，BG_i 和 BG_j，则该过程保证它们或者没有任何公共节点或者相同（步骤 805）。同样的，通过步骤 807，令校验节点组 i，i=0,1,...m-1，表示为 CG_i，为连接到第 i 行边界的所有校验节点，对于 i≠j，CG_i 和 CG_j，则验证 CG_i 或者没有任何公共节点或者相同。

步骤 803-807 的三个条件是实现有效并行处理所必需的。在步骤 809 中，映射矩阵被输出并存储在边界存储器 705 中（通过步骤 811）。在一个示范实施例中，映射矩阵代表存储器结构。该存储器可以是功能上等同于映射矩阵的物理上不同的存储设备。步骤 803 的特性保证在每个访问循环中，每个处理器（如处理器 701、703）在每个访问循环中从边界存储器 705 收到一个而且只一个值。

利用 n 个并行处理引擎，所有处理器将被均匀加载。第二个特性（通过步骤 805）是把比特节点划分成大小为 n 的组。只要存在与节点相关联的边界被映射到相同行，两个比特节点就在同一组中。此外，在主动处理每个比特节点（或校验节点）的处理过程中保留一些中间结果是必需的。从而，一组比特节点（或校验节点）的处理在处理新的比特节点（或校验节点）之前要完成；这是通过步骤 805 和 807 的第二和第三特性来保证的。

图 9 示出了一个依照本发明实施例的具有存储用于支持并行解码的边界值的映射矩阵的存储器的 LDPC 解码器。一个 m×n 映射矩阵存储在存储器 901 中，其中边界值被提供给 n 个并行处理器 903。为了说明的目的，用下面的奇偶校验矩阵来说明映射矩阵的应用：

$$\begin{bmatrix}1&1&0&0&0&0&0&1&0&0&1&0&0&0&0&0\\0&1&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\0&0&1&1&0&1&0&0&1&0&0&0&0&0&0&0\\1&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0\\1&0&0&0&0&1&0&0&1&0&0&0&0&1&1&0\\0&1&0&0&0&0&1&0&0&1&0&0&0&0&1&1\\0&0&1&0&0&0&0&1&0&0&1&0&1&0&0&1\\0&0&0&1&1&0&0&0&0&0&0&1&1&1&0&0\end{bmatrix}$$

在解码期间，该奇偶校验矩阵的每个非零项与一个边界值相关联。这些边界值可以如下标记:

$$\begin{bmatrix}e_1&e_4&0&0&0&0&0&e_{19}&0&0&e_{25}&0&0&0&0&0\\0&e_5&e_7&0&0&0&e_{17}&0&0&0&0&e_{27}&0&0&0&0\\0&0&e_8&e_{10}&0&e_{15}&0&0&e_{21}&0&0&0&0&0&0&0\\e_2&0&0&e_{11}&e_{13}&0&0&0&0&e_{23}&0&0&0&0&0&0\\e_3&0&0&0&0&e_{16}&0&0&e_{22}&0&0&0&0&e_{31}&e_{33}&0\\0&e_6&0&0&0&0&e_{18}&0&0&e_{24}&0&0&0&0&e_{34}&e_{35}\\0&0&e_9&0&0&0&0&e_{20}&0&0&e_{26}&0&e_{29}&0&0&e_{36}\\0&0&0&e_{12}&e_{14}&0&0&0&0&0&0&e_{28}&e_{30}&e_{32}&0&0\end{bmatrix}$$

边界值可以被一一映射到下面的9×4矩阵。

$$\begin{bmatrix}e_1&e_5&e_8&e_{11}\\e_2&e_4&e_7&e_{10}\\e_3&e_6&e_9&e_{12}\\e_{13}&e_{15}&e_{17}&e_{19}\\e_{14}&e_{16}&e_{18}&e_{20}\\e_{21}&e_{23}&e_{25}&e_{27}\\e_{22}&e_{24}&e_{26}&e_{28}\\e_{29}&e_{32}&e_{33}&e_{35}\\e_{30}&e_{31}&e_{34}&e_{36}\end{bmatrix}$$

假设矩阵映射存储器，上述矩阵是一个100%利用率的稠密矩阵。换句话说，存储器的所有单元都被完全利用。很容易验证，这个映射满足有效并行解码的上述所有三个特性。

在这个例子中，比特节点被划分成四个组，每组包括四个比特节点，从而比特节点i是与原始奇偶校验矩阵第i列相关联的比特节点。这些比特节点组是{1，2，3，4}，{5，6，7，8}，{9，10，11，12}，和{13，14，15，16}。

同样地，校验节点i是与原始奇偶校验矩阵第i行相关联的校验节点。校验

节点被划分成两个组：{1，2，3，4}和{5，6，7，8}。在这个例子中，利用四个并行处理器（n=4）。对于比特节点处理，第一行被读入到这四个并行处理器903，由于最终目标是更新与比特节点{1，2，3，4}相关联的边界，因此将产生中间结果并将其缓存在一个单独的存储器中。这是在{1，2，3，4}的所有原始边界值都经过处理后实现的。

值得注意的是，所有与比特节点{1，2，3，4}相关联的边界值都驻留在边界矩阵的前三行。这样，在下面的访问循环中，第二行或第三行将继续被访问。如果在任意其它的行开始该处理，就需要缓存另外的中间结果。当所有与比特节点{1，2，3，4}相关联的行都经过处理并且边界值被更新后，处理比特节点的其它组。

类似的，上述过程在校验节点执行。也就是说，校验节点在组中进行处理，当所有与该组相关联的边界都被更新后，进行下一组的处理。在每个访问循环中，对于每个正被主动处理的节点，提取边界矩阵中的正好一个值，从而保证处理器903中没有一个超载（或为此利用不够）。

人们知道，有效并行处理的三个条件（如图8的步骤803-807所述）可能有些限制，从而一个任意的LDPC码一般不能满足这些条件。这些限制通过从映射矩阵导出一个不同的矩阵可以放松，如下所述。

图10是依照本发明的各个实施例，生成一个导出映射矩阵以提供并行解码的处理流程图。为了更好的理解导出映射矩阵的生成，下面先设定一些术语和法则。一个n×n矩阵称作“一个密度对称矩阵”，如果矩阵的每一行和每一列有完全相同数量的非零项的话。密度对称矩阵的一个例子是全0矩阵；该矩阵被广泛应用到LDPC码结构上。

密度对称矩阵的另一个例子是有下述特性的矩阵。假定$b_0b_1...b_{n-1}$是一个二进制矢量，而a与n互素（relatively prime），则在第i行和第j列的项等于b_{ai+j}的矩阵是一个密度对称矩阵。特别的，当a=1时，产生一个轮换矩阵。

值得注意的，一个密度对称矩阵中任何行和/或列置换产生一个密度对称矩阵。然而，由于不同分矩阵可以有不同的置换，因此通过利用不同置换的构件块产生根本不同的LDPC码。对一个密度对称矩阵来说，如果一行中非零项的数量是k，则通过一次从一行中取出一个非零项可以把所有非零项映射（一对一）到一个k×n稠密矩阵。

建议下面的法则(法则1): 当且仅当该LDPC码的奇偶校验矩阵是等效于分块矩阵的置换时，该LDPC码满足对n个有效并行解码的所有三个条件，

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & \cdots & A_{1u} \\ A_{21} & A_{22} & \cdots & \cdots & A_{2u} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{v1} & A_{v2} & \cdots & \cdots & A_{vu} \end{bmatrix}$$

这样，每个分量分块矩阵 A_{ij} 是一个 $n \times n$ 密度对称矩阵。值得注意的，如此构成的矩阵不一定产生一个规则的LDPC码。

假定该法则对奇偶校验矩阵的结构给出了限定条件，显然的，一个一般的奇偶校验矩阵将不能满足这些条件。因此，设计具有这种结构限制的码从而能用有效存储器访问进行解码是必需的。

人们知道，一个能用n个并行引擎进行有效并行解码的码可能对其它数量（即，引擎数不得为 n）不是有效并行解码。如前面所提到的，为适应将来的发展，希望将来的接收机上用更少的并行处理。然而，甚至在这样一种减少并行操作的情况下还是需要满足所有关于有效并行处理的条件。

为了对少于n个处理器的有效并行处理保持三个条件，建议另一个法则(法则2)。令下面的矩阵为一个奇偶校验矩阵，从而每个分块矩阵 A_{ij} 是 $n \times n$ 密度对称矩阵:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & \cdots & A_{1u} \\ A_{21} & A_{22} & \cdots & \cdots & A_{2u} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{v1} & A_{v2} & \cdots & \cdots & A_{vu} \end{bmatrix}$$

如果 A_{11} 在行和列置换下可以分解成一个矩阵，

$$A_{11}=\begin{bmatrix} B_{11} & B_{12} & \cdots & \cdots & B_{1\rho} \\ B_{21} & B_{22} & \cdots & \cdots & B_{2\rho} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B_{q1} & B_{q2} & \cdots & \cdots & B_{q\rho} \end{bmatrix}$$

使得每个 B_{ij} 是一个 $m \times m$ 密度对称矩阵，且对于所有其它 $A_{ij} \neq A_{11}$，分块矩阵可以在如同 A_{11} 的相同行和列置换下分解成分块密度对称子矩阵，则，LDPC码通过如下来定义:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & \cdots & A_{1u} \\ A_{21} & A_{22} & \cdots & \cdots & A_{2u} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{v1} & A_{v2} & \cdots & \cdots & A_{vu} \end{bmatrix}$$

上述矩阵能满足关于 n 和 m 两种并行操作的有效并行处理的条件。

通过举例进行下面的定义：A_{ij} 是 n×n 的轮换矩阵，m 是 n 的任何整数分数。A_{ij} 的行和列标记为 0，1，...n。A_{ij} 的子矩阵由行和列的下标都等于一个常量模 m 的所有元素组成，它也是一个轮换矩阵。从而，在相同的行和列置换下，所有 A_{ij}' 再次被分解成轮换子矩阵。这意味着一个由 A_{ij}' 作为分量码的 LDPC 码能满足关于 n 和 m 两种并行操作的有效并行处理条件（这里 m 是 n 的任意整数部分）。

给定任何具有奇偶校验矩阵 H 的 LDPC 码，可以通过将 H 的一些非零项转换成 0 从 H 导出其它 LDPC 码。任何以这种方式获得的码被称作"原始 LDPC 码的导出码"。在步骤 1001 中，检索到用于 n 个可并行解码的码的映射矩阵。如果原始码是用 n 个并行引擎校验矩阵的全密度存储器进行 n 个并行解码的，则任何导出码也可以用减少的密度存储器矩阵进行 n 个并行解码。

从二分图来看，可以通过删除一些边界得到导出码（步骤 1003）。导出码的映射矩阵与基础码（base code）的映射矩阵本质上是相同的；然而，对应所删除的边界的项可以标记为"DON'TCARE"（即，空值）。这些存储器单元因此被浪费，而且访问循环在提取空值时被消耗。然而，只要被边界删除的总数相对较少，是不至于显著影响效率的。这样的导出码可以显示其它期望的特性，例如易于解码和改善距离特性。在步骤 1005 中，导出映射矩阵被输出。

图 11 示出了一个依照本发明的实施例，具有存储器存储用于支持并行解码的边界值的导出映射矩阵的 LDPC 解码器。该矩阵于是被存储在存储器 1101 中，通过 m 个处理器 1103 进行访问。在这种安排下，在使用的并行引擎的数量上有很大的灵活性，同时保持了有效的存储器结构。

图 12 示出了一个能够在其上实现本发明实施例的计算机系统。计算机系统 1200 包括总线 1201 或其它的用于传递信息的通信机制，和连接到总线 1201 用于处理信息的处理器 1203。该计算机系统 1200 也包括主存储器 1205，如一个随机访问存储器（RAM）或其它的动态存储设备，它们连接到总线 1201 用于存储信息和由处理器 1203 执行的指令。主存储器 1205 也可以在可由处理器 1203

执行的指令执行期间用于存储临时变量或其它中间信息。计算机系统1200进一步包括一个只读存储器（ROM）1207或其它的静态存储设备，它们连接到总线1201用于存储静态信息和存储器1203的指令。存储设备1209，如磁盘或光盘，被附加连接到总线1201，用于存储信息和指令。

该计算机系统1200可以通过总线1201连接到显示器1211，例如阴极射线管（CRT）、液晶显示器、有源矩阵显示器、或等离子体显示器，将信息显示给计算机用户。输入设备1213，如一个包括字母数字和其它键的键盘，连接到总线1201，用来把信息和命令选择传送到处理器1203。另一种类型的用户输入设备是光标控制1215，例如鼠标、跟踪球、或光标方向键，把方向信息和命令选择传送到处理器1203，并控制显示器1211上的光标运动。

依照本发明的一个实施例，计算机系统1200响应处理器1203执行一个包含在主存储器1205中的指令排列，提供LDPC码的并行解码。这个指令可以从诸如存储设备1209的另一个计算机可读介质读入主存储器1205。执行包含在主存储器1205中的指令排列，使处理器1203执行在这里所描述的处理步骤。为了执行包含在主存储器1205中的指令也可以采用一个多处理装置中的一个或多个处理器。在可选实施例中，硬件电路可以代替软件指令或与之相结合来执行本发明的实施例。这样，本发明的实施例不限于任何硬件电路和软件的特定组合。

该计算机系统1200也包括连接到总线1201的通信接口1217。该通信接口1217提供耦合到网络链路1219的双向数据通信，该网络链路与本地网络1221连接。例如，通信接口1217可以是数字用户线（DSL）卡或调制解调器、综合业务数字网（ISDN）卡、电缆调制解调器、或提供到相应类型电话线的数据通信连接的电话调制解调器。另一个例子，通信接口1217可以是局域网（LAN）卡（如，以太网（Ethernet™）或异步传输模式（ATM）网）提供到兼容LAN的数据通信连接。也可以用无线链路来执行。在任何一种实现方式中，通信接口1217发送和接收承载代表不同信息类型的数字数据流的电、电磁、或光信号。更进一步的，通信接口1217可以包括外围接口设备，如通用串行总线（USB）接口、PCMCIA（个人计算机存储卡国际协会）接口等。

通常，网络链路1219提供穿过一个或多个网络到其它的数据设备的数据通信。例如，网络链路1219可以提供穿过局域网1221到主机1223的连接，它具有到网络1225（例如广域网络（WAN）或全球分组数据通信网络，现在通常称

为“因特网”）或到由服务提供商操作的数据设备的连通性。本地网 1221 和网络 1225 都利用电、电磁、光信号来传送信息和指令。穿越不同网络的信号以及在网络链路 1219 并通过通信接口 1217，该接口与计算机系统 1200 传送数字数据，的信号是承载信息和指令的载波的示范形式。

计算机系统 1200 能够通过网络、网络链路 1219、和通信接口 1217 发送消息和接收包括程序码的数据。在因特网的例子中，一个服务器（未示出）可以通过网络 1225、本地网 1221 和通信接口 1217 发送属于执行本发明的一个实施例的应用程序的请求码。处理器 1203 可以执行该传输码同时接收和/或把该码存储在存储设备 129，或存储在其它非易失性存储器中，用于后面的运行。如此，计算机系统 1200 可以得到载波形式的应用程序代码。

在这里所用的“可读计算机介质”术语指任何提供用于执行的指令给处理器 1203 的介质。这样一种介质可以有多种形式，包括但是不限于非易失性介质、易失性介质、传输介质。非易失性介质包括例如象存储设备 1209 的光盘或磁盘，易失性介质包括象主存储器 1205 的动态存储器。

传输介质包括同轴电缆、铜线和光纤，包括包含总线 1201 的线路。传输介质也能采用声学的、光学的、或电磁波的形式，如那些在射频（RF）和红外（IR）数据通信中产生的。可读计算机介质的通用形式包括例如软盘、软碟、硬盘、磁带，任何其它的磁介质， CD-ROM、CDRW、DVD，任何其它的光介质，穿孔卡片、纸带、光学侧标纸。任何带洞的或带可辨认标记的物理介质，RAM、PROM、和 EPROM、FLASH-EPROM，任何其它的存储片或卡带，载波、或任何其它计算机可读的介质。

不同形式的计算机可读介质可用于给处理器提供用于执行的指令。例如，用于实现至少本发明的部分的指令可以最初产生在一个远程计算机的磁盘上。在这样的情况下，远程计算机将指令加载到主存储器并通过使用调制解调器的电话线来发送。一个本地计算机系统的调制解调器在电话线上接收数据，用红外发射机把数据转换成一个红外信号，并把红外信号发射到例如个人数字助理（PDA）和膝上型电脑的便携式计算设备。便携式计算设备上的红外探测器收到红外信号产生的信息和指令，把数据放在总线上。总线把数据传递到主存储器，处理器从该主存储器中检索并执行该指令。由主存储器接收的指令可选的可在处理器执行该命令之前或之后在存储设备中储存。

从而，本发明的不同实施例提供了一种对结构化低密度奇偶校验（LDPC）码进行解码的方法。建立用于可并行解码的码的准则，以产生对应于一个奇偶校验矩阵的一个映射矩阵。这些准则产生使用 LDPC 码的并行解码结构，并提供有效存储结构的 LDPC 码。本方法也允许对于少于 n 个处理器（如 m 个处理器，m 是 n 的整数分数）也保持用于有效并行处理的这些并行解码条件。在上述安排下，诸如存储器和处理引擎的解码器资源被有效利用。同样的，实现的灵活性得以加强，从而能够采用 m 个并行引擎。

虽然本发明已经用许多实施例和实施方案进行了描述，但本发明不仅限于此，而是覆盖所有落入所附权利要求的范围之内的多种明显的修改和等同的配置。

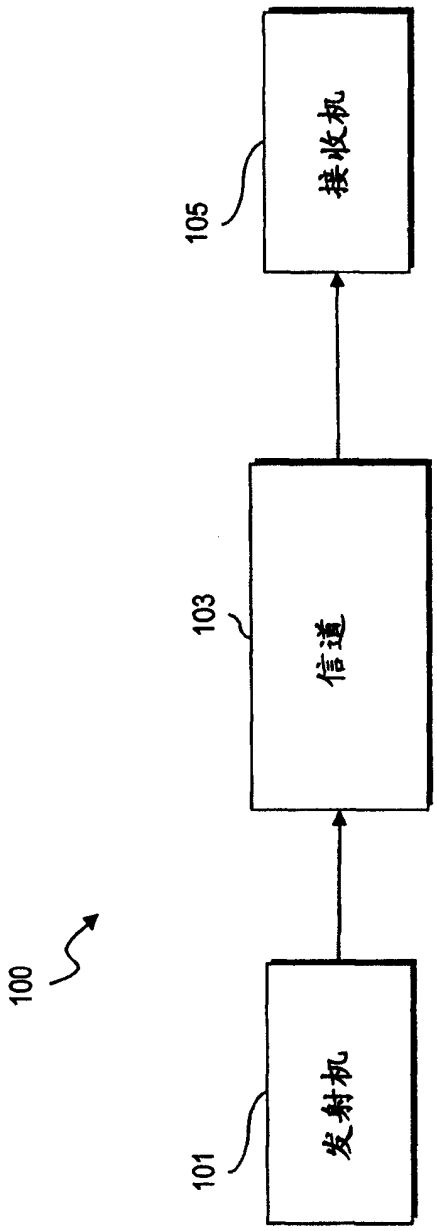


图1

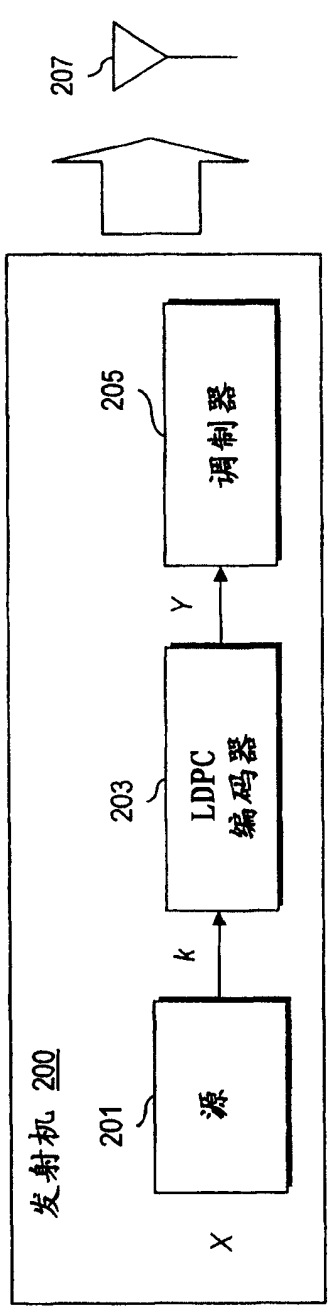

发射机 200
201
源
X
k
203
LDPC
编码器
Y
205
调制器
207

图2

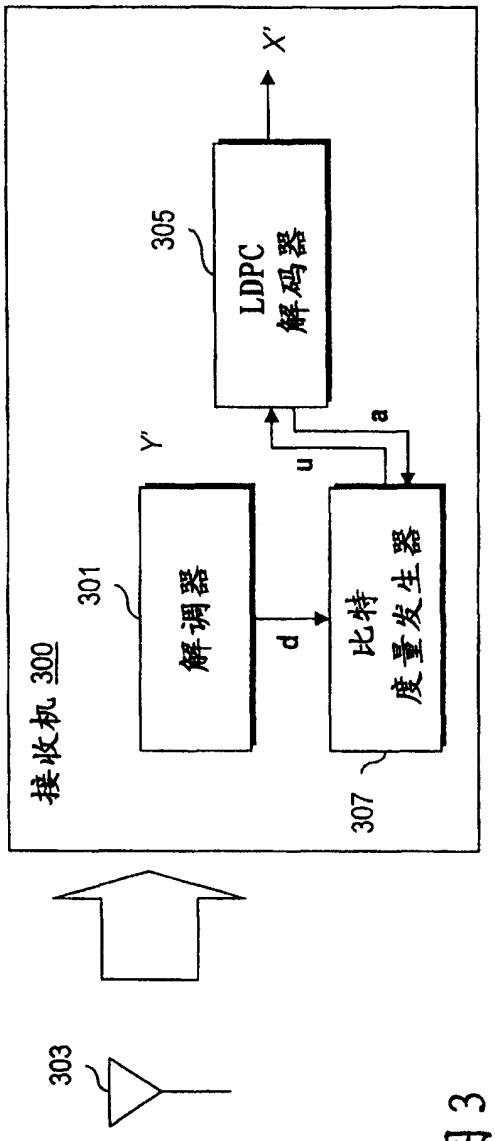
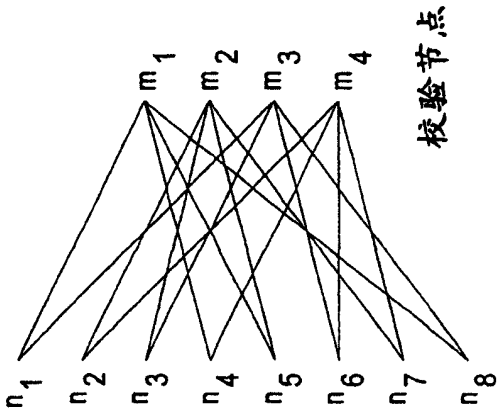

接收机 300
301
解调器
Y'
d
比特
度量发生器
307
u
a
305
LDPC
解码器
X'
303

图3

图4

$$H=\begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{array}\begin{array}{c} \begin{array}{cccccccc} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 & n_7 & n_8 \end{array} \\ \left[\begin{array}{cccccccc} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}\right] \end{array}$$

图5



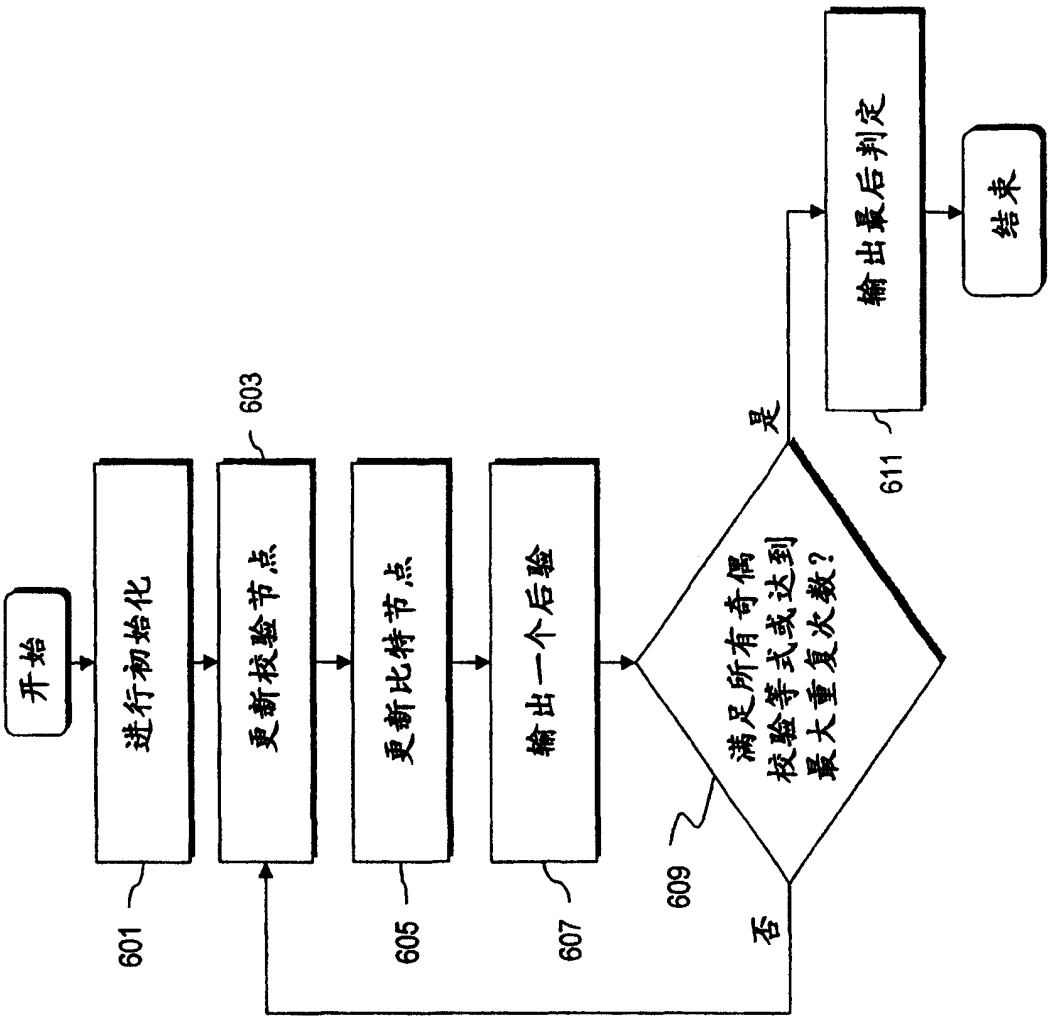
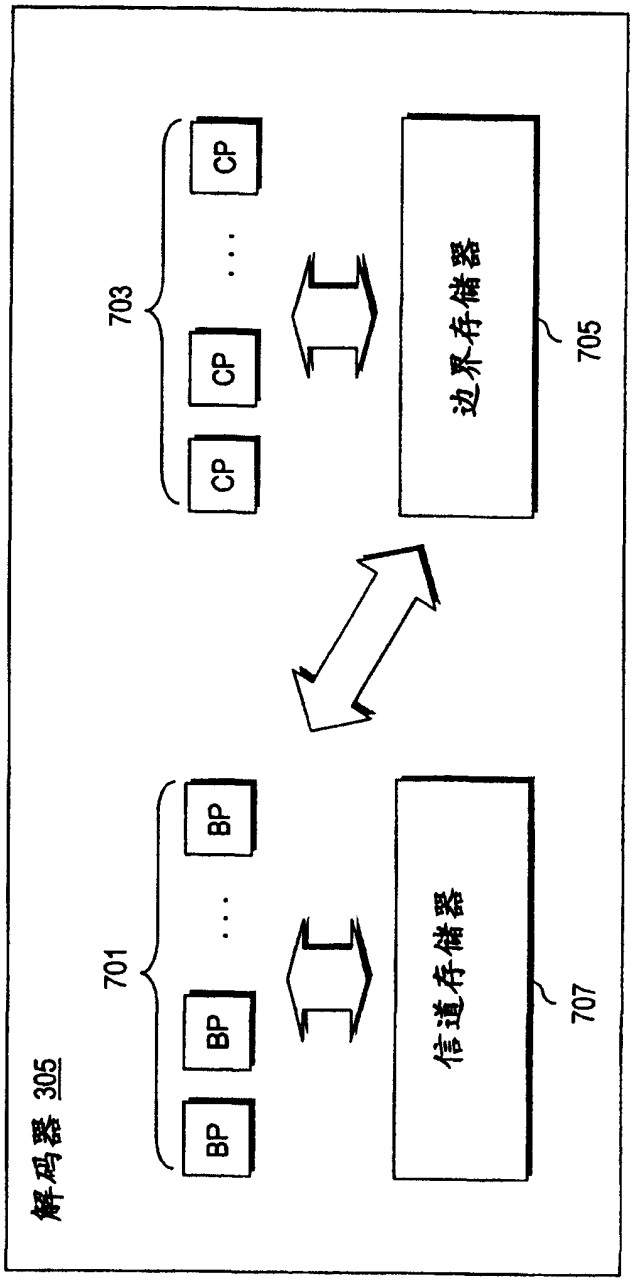

开始
进行初始化
601
更新校验节点
603
更新比特节点
605
输出一个后验
607
满足所有奇偶校验等式或达到最大重复次数?
609
否
是
输出最后判定
611
结束

图6

图7



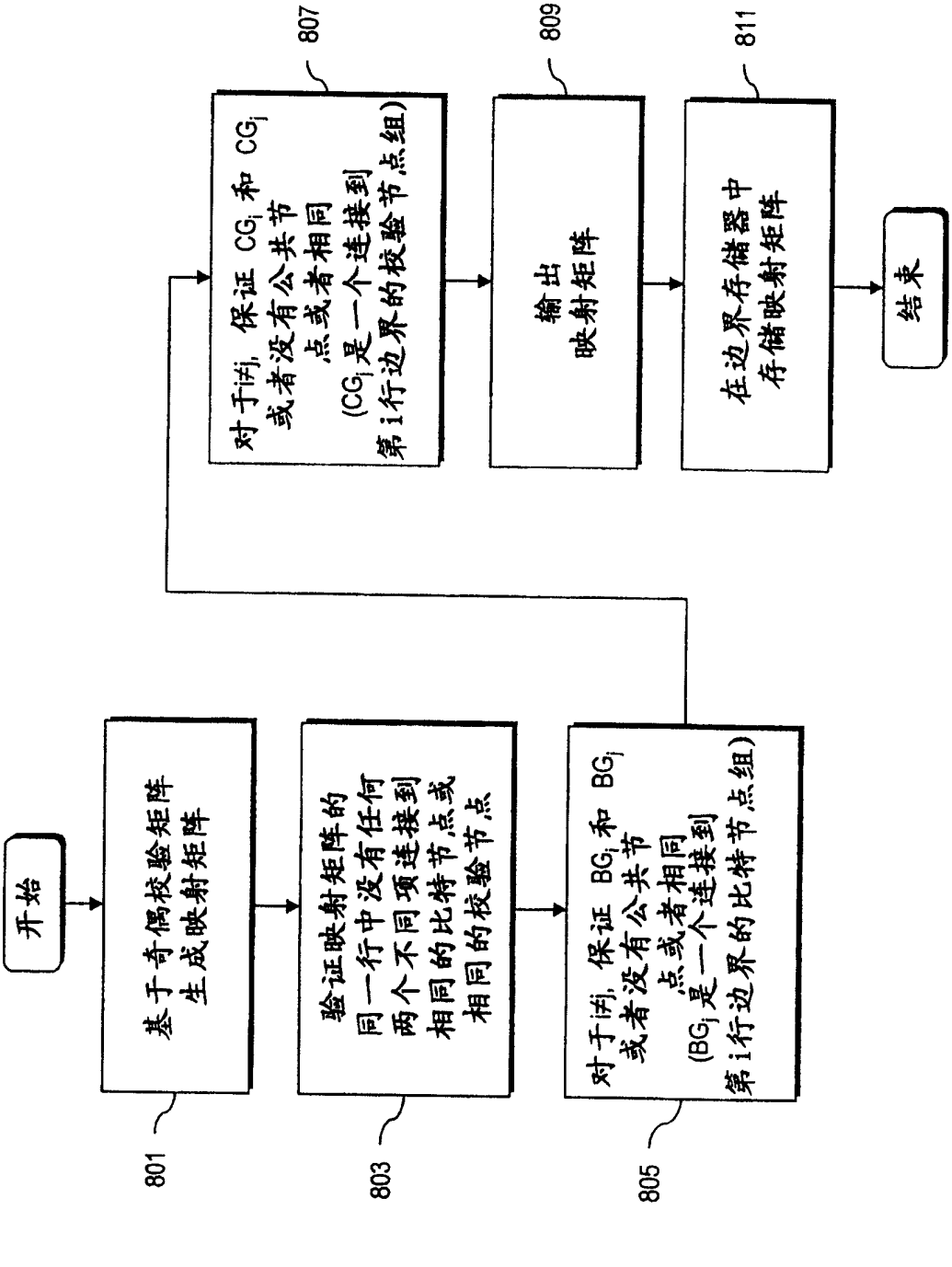

开始
基于奇偶校验矩阵
生成映射矩阵
801
验证映射矩阵的
同一行中没有任何
两个不同项连接到
相同的比特节点或
相同的校验节点
803
对于i和j，保证 BG_i 和 BG_j
或者没有公共节
点或者相同
(BG_j 是一个连接到
第i行边界的比特节点组)
805
对于i和j，保证 CG_i 和 CG_j
或者没有公共节
点或者相同
(CG_j 是一个连接到
第i行边界的校验节点组)
807
输出
映射矩阵
809
在边界存储器中
存储映射矩阵
811
结束

图8

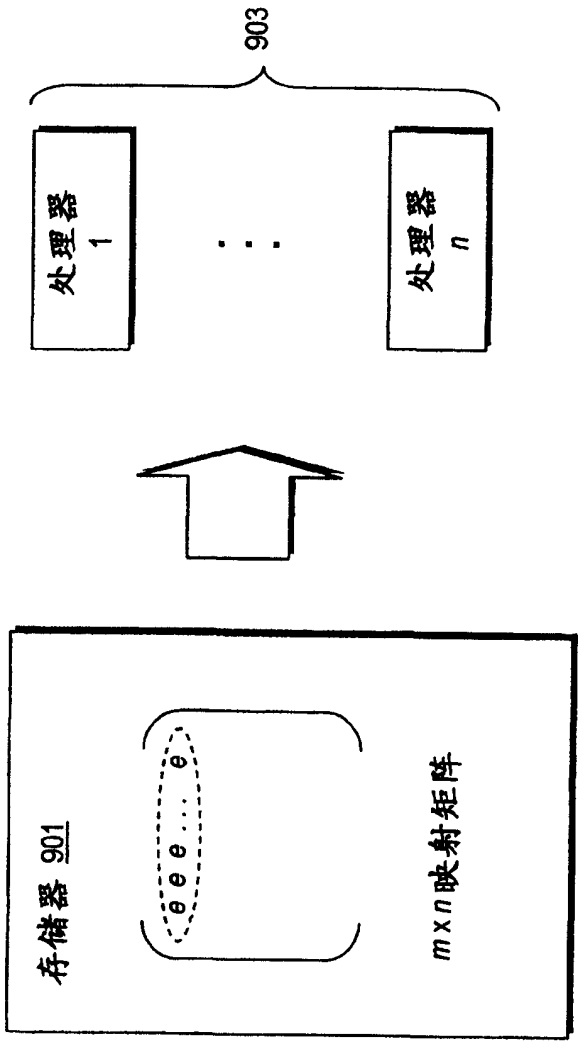


图 9

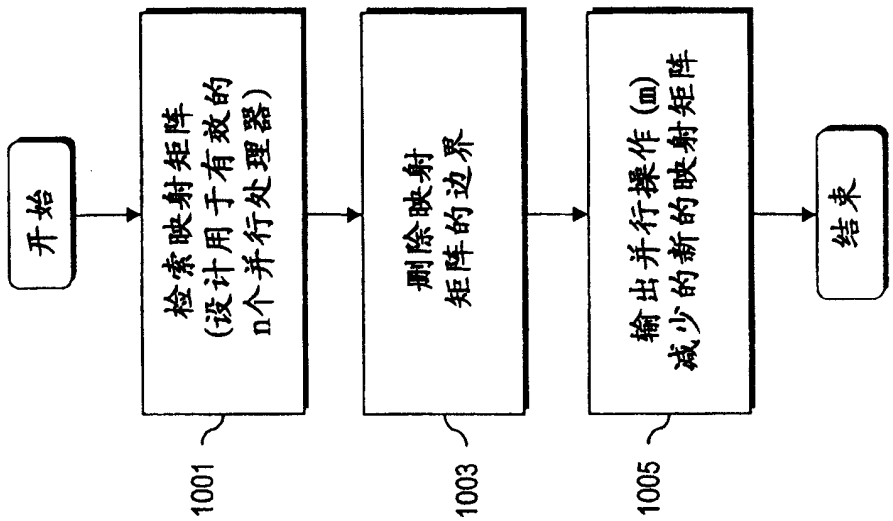

开始
检索映射矩阵
(设计用于有效的
n个并行处理器)
删除映射
矩阵的边界
输出并行操作(m)
减少的新的映射矩阵
结束
1001
1003
1005

图10

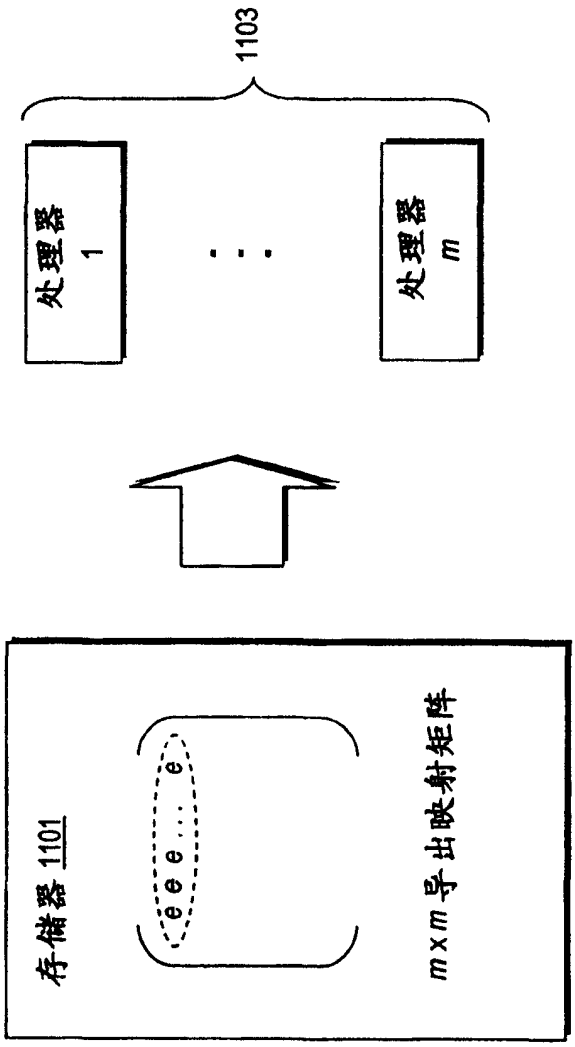


图 11

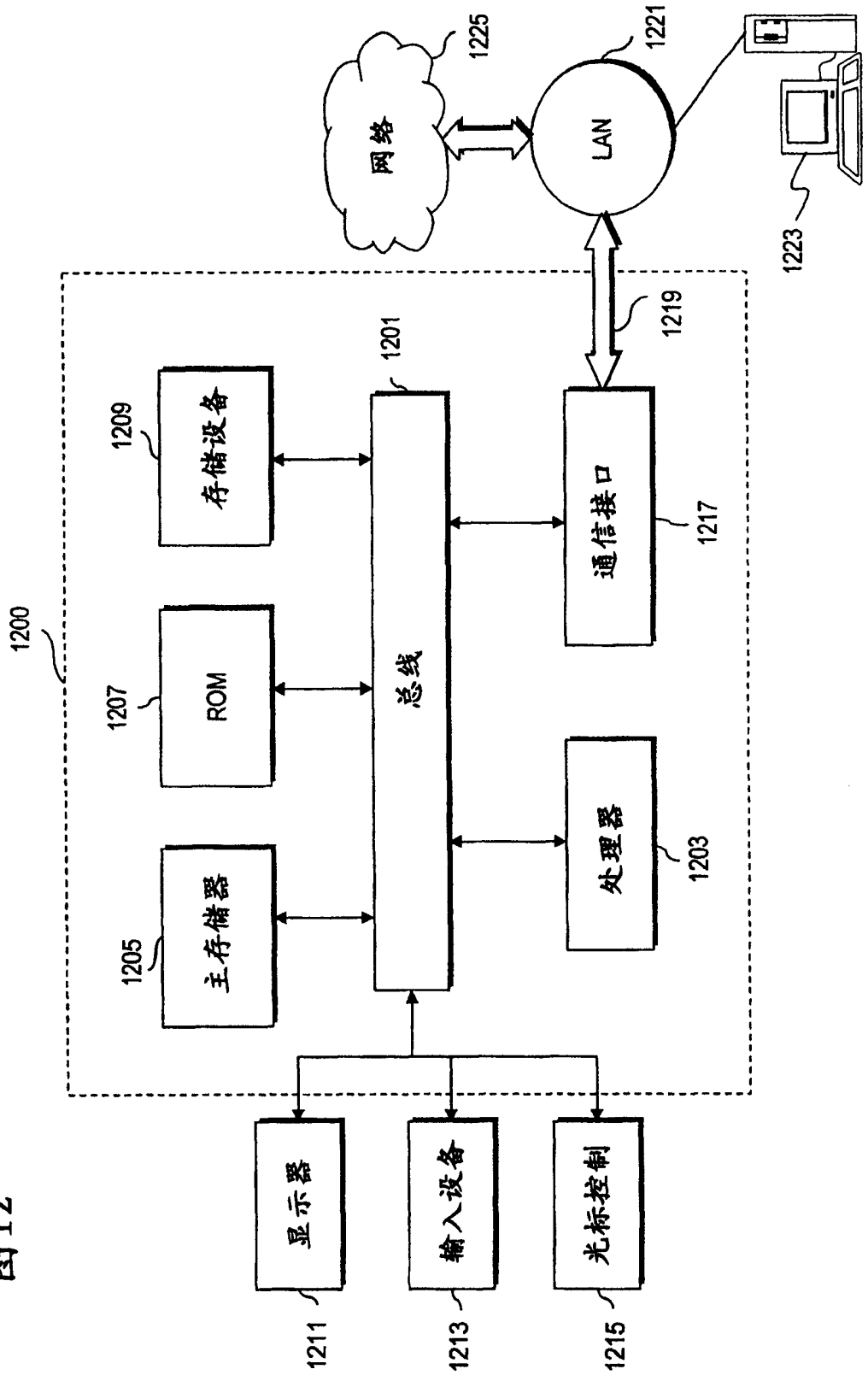

网络
1225
LAN
1221
1223
1219
1200
1201
1209
存储设备
1207
ROM
1205
主存储器
总线
通信接口
1217
处理器
1203
显示器
1211
输入设备
1213
光标控制
1215

图12