



(12) 发明专利

(10) 授权公告号 CN 101432968 B

(45) 授权公告日 2012.01.11

(21) 申请号 200580031594.1

(51) Int. Cl.

(22) 申请日 2005.07.20

H03M 13/00(2006.01)

(30) 优先权数据

(56) 对比文件

10/895,547 2004.07.21 US

CN 1323463 A, 2001.11.21, 全文.

CN 1133512 A, 1996.10.16, 全文.

(85) PCT申请进入国家阶段日

2007.03.20

审查员 于春晖

(86) PCT申请的申请数据

PCT/US2005/025752 2005.07.20

(87) PCT申请的公布数据

W02006/014742 EN 2006.02.09

(73) 专利权人 高通股份有限公司

地址 美国加利福尼亚州

(72) 发明人 汤姆·理查森 金辉

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 王英

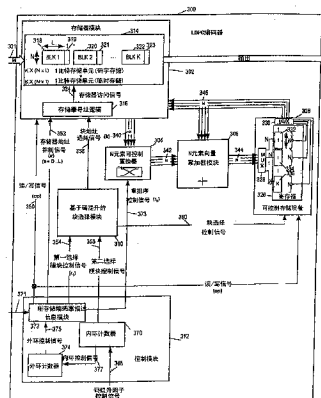
权利要求书 3 页 说明书 18 页 附图 5 页

(54) 发明名称

LDPC 编码方法和装置

(57) 摘要

描述了灵活并相对硬件高效的 LDPC 编码器。利用小于用于控制编码过程的码结构的全部并行度的并行度级别来实现所述编码器。每个用于描述码结构的相对简单微码指令能被存储并被执行多次以完成对码字的编码。使用相同组微码指令能支持不同的码字长度,但取决于所选将被使用的提升因子,码被实现的次数不同。在不需要改变储存的码描述信息的情况下,仅通过改变用于控制编码过程的码提升因子,所述 LDPC 编码器能在编码不同长度的码字间切换。当编码比最大可支持码字长度短的码字时,一些块存储单元和 / 或寄存器可以不使用。



1. 一种低密度奇偶校验 LDPC 编码器,包括:

存储器模块,其至少包括 $N \times L \times K$ 个存储单元,其中 N 和 L 都是正整数并且 K 是 > 1 的整数;

可控制置换器,其耦合到所述存储器模块,对至少 N 个元素执行元素重排序操作;

向量累加器模块,其包括 N 个并行排列的累加器,所述向量累加器模块包括:

i) 至少 N 个比特宽的第一输入端,其对应于所述可控制置换器的输出端;

ii) 至少 N 个比特宽的第二输入端,以及

iii) 至少 N 个比特宽的向量累加器输出端;

包括 $N \times K$ 个存储单元的可控制存储设备,所述可控制存储设备包括:块选择控制信号输入端,其用于接收指示一个包括将被访问的至少 N 个存储单元的块的信号;可控制存储设备输入端,其对应于所述向量累加器输出端;以及至少 N 个比特宽的可控制存储设备输出端,其用于输出从所述可控制存储设备读出的至少 N 个比特以作为所述第二输入端的输入;以及

其中,耦合到所述可控制存储设备的块选择模块用于向所述可控制存储设备提供作为所述信号的块选择控制信号。

2. 如权利要求 1 所述的编码器,所述编码器进一步包括:

控制模块,其用于根据将被执行的编码操作来生成第一选择模块控制信号,所述第一选择模块控制信号作为第一控制信号输入被提供给所述块选择模块。

3. 如权利要求 2 所述的编码器,

其中,所述块选择模块根据码提升因子生成所述块选择控制信号;并且

其中,所述控制模块进一步生成重排序控制信号,该重排序控制信号被提供给所述可控制置换器的重排序控制输入端。

4. 如权利要求 2 所述的编码器,进一步包括:

至少 N 个比特宽的总线,其用于将所述存储器模块耦合到所述可控制置换器。

5. 如权利要求 3 所述的编码器,其中,所述块选择模块进一步包括块地址选择输出端,该块地址选择输出端耦合到所述存储器模块的相应输入端。

6. 如权利要求 5 所述的编码器,其中,所述块选择模块进一步包括第二选择模块控制输入端,其用于接收指示将被使用的码提升因子的信号。

7. 如权利要求 6 所述的编码器,其中,所述控制模块进一步生成被提供给所述存储器模块的存储器地址控制信号。

8. 如权利要求 7 所述的编码器,其中,所述存储器模块包括寻址逻辑,其用于根据所述存储器地址控制信号和所述块地址选择输出端的块地址选择信号生成存储器访问信号。

9. 如权利要求 8 所述的编码器,

其中,所述可控制存储设备进一步包括读/写控制输入端;并且

其中,所述控制模块进一步包括读/写控制信号输出端,该读/写控制信号输出端耦合到所述可控制存储设备的所述读/写控制输入端。

10. 如权利要求 1 所述的编码器,其中,所述可控制存储设备的所述存储设备输出端耦合到所述向量累加器模块的所述第二输入端以及所述存储器模块的输入端。

11. 如权利要求 9 所述的编码器,其中,所述读/写控制信号输出端进一步耦合到所述

存储器模块的相应输入端。

12. 如权利要求 7 所述的编码器,其中,所述存储器地址控制信号是大于 0 并且小于 $L+1$ 的整数值,并且在编码操作期间通过每个所表示的整数值 1 到 L 来循环,其中 L 是正整数。

13. 如权利要求 6 所述的编码器,其中,将被使用的所述码提升因子是用户选择的控制值 SK ,其是 K 的整数因子。

14. 如权利要求 13 所述的编码器,其中,当所述码提升因子 SK 小于 K 时,在编码期间,所述存储器模块中的 $N \times L \times (K-SK)$ 个存储单元没有被使用。

15. 如权利要求 13 所述的编码器,其中,当所述码提升因子 SK 小于 K 时,在编码期间,所述可控制存储设备中的所述 $N \times K$ 个存储单元的一部分没有被使用。

16. 如权利要求 1 所述的编码器,

其中,在所述存储器模块中的所述 $N \times L \times K$ 个存储单元中的每一个均是一比特的存储单元;并且

其中,在所述可控制存储设备中的所述 $N \times K$ 个存储单元中的每一个均是一比特的存储单元。

17. 如权利要求 1 所述的编码器,其中,所述控制模块包括一组微码指令,其描述将被用于对数据编码的码结构,每个微码指令对应于一个码结构,该码结构被执行 K 次以编码具有总长 $K \times L \times N$ 比特的码字。

18. 一种执行低密度奇偶校验 LDPC 编码处理的方法,包括:

在存储器模块的每个存储单元存储多个比特,该存储器模块包括 $N \times L \times K$ 个存储单元,其中 N 和 L 都是正整数并且 K 是 > 1 的整数;

操作耦合到所述存储器模块的可控制置换器,对至少 N 个元素执行元素重排序操作;

操作耦合到所述可控制置换器的向量累加器模块所包括的 N 个并行排列的累加器,将从与所述可控制置换器的输出端对应的至少 N 个比特宽的第一输入端输入的至少 N 个比特和从至少 N 个比特宽的第二输入端输入的至少 N 个比特并行地合并成从至少 N 个比特宽的向量累加器模块输出端输出的至少 N 个累加值;

操作包括 $N \times K$ 个存储单元的可控制存储设备,将所述至少 N 个累加值写入到由在所述可控制存储设备的块选择控制信号输入端处接收的信号指示的一个包括将被访问的至少 N 个存储单元的块中,并从所述可控制存储设备的至少 N 个比特宽的输出端输出从所述可控制存储设备的所述块中读出的至少 N 个比特;以及

其中,所述信号是基于码提升的块选择模块通过根据存储的码描述和用于控制编码操作定时的时钟信号生成的第一选择模块控制信号而向所述可控制存储设备提供的块选择控制信号。

19. 如权利要求 18 所述的方法,进一步包括:

生成重排序控制信号;

将所述重排序控制信号提供给所述可控制置换器;以及

操作所述可控制置换器,以根据所述提供的重排序控制信号,执行消息重排序操作。

20. 如权利要求 19 所述的方法,进一步包括:

操作所述基于码提升的块选择模块,以根据存储的码描述信息生成块地址选择信号;

以及

将所述块地址选择信号提供给所述存储器模块,以用于确定将被访问的一组存储单元。

21. 如权利要求 20 所述的方法,进一步包括:

操作所述基于码提升的块选择模块,以接收指示将被使用的码提升因子的信号。

22. 如权利要求 21 所述的方法,进一步包括:

操作控制模块,以生成用于确定将被访问的所述一组存储单元的存储器地址控制信号;以及

将所述存储器地址控制信号提供给所述存储器模块。

23. 如权利要求 22 所述的方法,其中,所述存储器模块包括寻址模块,所述方法进一步包括:

操作所述寻址模块,以根据所述存储器地址控制信号和所述块地址选择信号生成存储器访问信号,所述存储器访问信号控制在一个时间点访问哪个特定存储单元块。

24. 如权利要求 23 所述的方法,进一步包括:

操作所述控制模块,以生成读/写控制信号,该读/写控制信号用于控制一个可控制存储访问操作是读还是写访问操作;以及

将所述生成的读/写控制信号提供给所述可控制存储设备。

25. 如权利要求 22 所述的编码器,其中,所述存储器地址控制信号是大于 0 并且小于 $L+1$ 的整数值,所述方法进一步包括:

当编码一组比特时,通过每个表示的整数值 1 到 L 来循环。

26. 如权利要求 21 所述的方法,其中,将被使用的所述码提升因子是用户选择的值 SK ,其是 K 的整数因子。

27. 如权利要求 26 所述的方法,进一步包括:

当所述码提升因子 SK 是小于 K 的整数时,在编码期间,所述存储器模块中的所述 $N \times L \times K$ 个存储单元中的一部分没有被使用。

28. 如权利要求 27 所述的方法,其中,所述存储器模块中的所述 $N \times L \times K$ 个存储单元中的每一个均是一比特的存储单元,并且其中,所述 $N \times L \times K$ 个存储单元中的一部分没有被使用包括 $K-SK$ 个存储单元没有被使用。

29. 如权利要求 26 所述的方法,进一步包括:

当所述码提升因子 SK 小于 K 时,在编码期间,所述可控制存储设备中的所述 $N \times K$ 个存储单元中的一部分没有被使用。

30. 如权利要求 29 所述的方法,其中,所述可控制存储设备中的所述 $N \times K$ 个存储单元中的每一个均是一比特的存储单元,并且其中,在编码期间所述可控制存储设备中的所述 $N \times K$ 个存储单元中的一部分没有被使用包括 $K-SK$ 个存储单元没有被使用。

LDPC 编码方法和装置

技术领域

[0001] 本发明涉及为了利于错误检测和 / 或纠正的目的,用于编码数据的方法和装置,例如,通过使用诸如低密度奇偶校验 (LDPC) 码的奇偶校验码。

背景技术

[0002] 在通信和数据存储系统中,纠错码是普遍存在的。通过在数据流中引入冗余,纠错码补偿在这些系统中信息传递的固有不可靠性。近来在已知为低密度奇偶校验 (LDPC) 码的码类中产生了相当大的兴趣。可证明 LDPC 码是好码。在各种信道上,已经证实 LDPC 码真正接近于信道容量—由克劳德·香农创建的传输上限。

[0003] 通常由称作 Tanner 图的二分图来表示 LDPC 码,其中,一组节点,即变量节点,对应于码字的比特,并且另一组节点,即约束节点,有时也称为校验节点,对应于一组用于定义码的奇偶校验约束。图中的边将变量节点连接到约束节点。如果变量节点与约束节点由图中的边连接,那么它们被称作邻居。

[0004] 当且仅当对于每个约束节点,与约束(经由与这些比特相关联的变量节点)相邻的比特的模二和为零时,即,它们包括偶数个一时,那么与变量节点一一相关联的比特序列是码的码字。

[0005] 图 1 示出了示例性二分图 100,其确定示例性的长为十以及二分之一速率的 (3,6) 规则 LDPC 码。长度十表明存在十个变量节点 V_1-V_{10} ,每个由码字 X_1-X_{10} 的一个比特标识。在图 1 中,通过参考数字 102 标识该组变量节点 V_1-V_{10} 。二分之一速率表明存在变量节点一半的校验节点,例如,存在通过参考数字 106 标识的五个校验节点 C_1-C_5 。二分之一速率进一步表明五个约束是线性独立的。示例性二分图 100 包括边 104,其中,示例性 (3,6) 规则 LDPC 码具有连接到每个变量节点的 3 条边和连接到每个约束节点的 6 条边以及任意两个节点之间的最多一条边。

[0006] 尽管图 1 说明了与长为 10 的码相关联的图,但是能理解,表示长为 1000 的码字的图将复杂 100 倍。

[0007] LDPC 码 Tanner 图表示的替代是诸如图 2 的图 200 中示出的奇偶校验矩阵表示。在这种码表示中,通常称为奇偶校验矩阵的矩阵 H 202 包括相应的边连接、变量节点以及约束节点信息。在矩阵 H 202 中,每列对应于变量节点之一,而每行对应于约束节点之一。因为在示例性码中存在 10 个变量节点和 5 个约束节点,因此矩阵 H 202 包括 10 列及 5 行。如果在图中出现边,即,如果两个节点是邻居,那么将对应于特定变量节点和特定约束节点的矩阵 202 中的项设置为 1,否则,设置为 0。例如,因为由边将变量节点 V_1 连到约束节点 C_1 ,因此,将一设置在矩阵 202 的最左上角。然而,变量节点 V_5 没有连到约束节点 C_1 ,因此将 0 设置在矩阵 202 的第一行的第五个位置,这表明没有连接对应的可变及约束节点。我们说,如果 H 202 的行在 GF[2] 上是线性独立的向量,那么约束是线性独立的,其中 GF[2] 是二进制伽罗瓦域。

[0008] 在矩阵表示的情况下,将被发送的码字 X 能表示为向量 204,向量 204 包括将被处

理的码字的比特 X_1-X_n 。当且仅当矩阵 202 与矩阵 204 的乘积等于零时,即 $Hx = 0$, 比特序列 X_1-X_n 是一个码字。

[0009] 编码 LDPC 码指的是从一组信息比特中产生码字的过程。通过预处理 LDPC 图表示或矩阵表示,在实际编码前能确定对应于信息比特的一组变量节点。

[0010] 为了建立通用 LDPC 码的编码器,第一步是要找到 H 的行和列的置换(permutation),因此,相当于重排序,我们能把 $m \times n$ 矩阵 H 划分成如下的子矩阵

$$[0011] \quad H = \begin{bmatrix} T & A & B \\ E & C & D \end{bmatrix}$$

[0012] 其中, T 是 $t \times t$ 上三角子矩阵,即,在主对角线以下的所有项都是零, E 是 $g \times t$ 子矩阵, A 是 $t \times g$, C 是 $g \times g$, B 是 $g \times (n-m)$, D 是 $g \times (n-m)$ 并且 $t+g = m$ 。此外, $g \times g$ 矩阵 $\phi = ET^{-1}A+C$ 是可逆的(我们这里假设 H 是行满秩)。

[0013] 然后如下继续编码。给定信息比特 x^s ,为了编码码字 $x = [x^{p1} \quad x^{p2} \quad x^s]$,使用回代,对于 y,我们首先解

$$[0014] \quad [T \ A \ B] [y \ 0 \ x^s]^T = 0$$

[0015] 对于 x^{p2} ,接下来我们解

[0016] $\phi x^{p2} = [E \ C \ D] [y \ 0 \ x^s]^T$ 对于这步,预先计算矩阵 ϕ^{-1} 。最后,使用回代,对于 x^{p1} ,解

$$[0017] \quad [T \ A \ B] [x^{p1} \quad x^{p2} \quad x^s]^T = 0 \text{ 向量 } [x^{p1} \quad x^{p2} \quad x^s]^T \text{ 组成码字。}$$

[0018] 尽管编码效率和高数据率是重要的,但对于在例如消费设备的广泛设备范围中使用的实用性的编码系统,编码器能够以合理的成本实现是重要的。因此,例如在硬件成本方面,高效地实现用于错误纠正和 / 或检测目的的编码方案的能力,是重要的考虑。

[0019] 考虑上述的讨论,应该理解,存在针对用于实现 LDPC 码高效体系结构的编码器装置和方法的需要。允许重用相同硬件以编码不同长度码字的装置和方法将是有益的和期望的。这是因为它将允许在编码器使用期间更大的灵活性,并且允许使用不同尺寸码字来编码不同组数据,因此在不需多个编码器支持这样灵活性的情况下,允许为特定的编码应用,例如,通信会话或数据存储应用,来选择码字尺寸。

发明内容

[0020] 本发明针对编码方法和装置,例如,用于实现低密度奇偶校验(LDPC)编码器的方法和装置。本发明的各个实施例针对允许使用单个编码器来编码不同长度码字的特别灵活的编码器。这允许本发明的 LDPC 编码器在编码用于例如第一通信应用或数据存储应用这样的第一应用的第一长度码字,和编码用于第二应用的第二长度码字之间切换。实际上,使用相同硬件能支持很多种码字长度,该相同硬件允许通过对编码器中使用的码字描述的简单修改来改变码字长度。在相对简单的微码中可以反映码字描述,该微码被执行来满足特定应用的需要。

[0021] 本发明的编码器能储存对应于不同码结构的多组微码。当选择码结构以用于编码时,就选择相应的微码,并且生成对应于所选码结构的特定长度的码字。对于给定的码结

构,通过选择不同的码提升因子,可以生成不同长度的码字。使用比最大可支持码提升因子小的码提升因子,通常会导致在编码器存储器中没有使用一些存储单元,例如块存储单元。通常由最大可支持码字尺寸确定在存储器中提供的块存储单元的数量。

[0022] 本发明的编码器能在不同的提升因子间切换,因此,这允许编码器在生成不同长度的码字间切换。因此,本发明的编码器能用于生成很适合于特定应用的长度的码字,并且不受硬件产生单一长度码字的约束。然而,在许多情况下,最大可支持码字长度取决于在编码期间可用于存储比特块的存储量。

[0023] 对于包括 $(N \times L)$ 比特整数倍直到最大值 K 倍的不同尺寸的码字,最大可支持码字尺寸可以是 $K \times N \times L$ 比特,其中 K 、 N 和 L 都是正整数。

[0024] 本发明的编码器特别适合编码依据用于定义码并拥有特定分层结构的 LDPC 图生成的码字,其中,在很大程度上,整个 LDPC 图看起来是由多个副本组成,例如 Z 个较小的图组成。 Z 个图副本可以是,但不需要是相同的。为了解释本发明的目的,我们将把所述较小的图称为投影图,把全图称为提升图,并把 Z 称为提升因子。因此,投影图用作生成或描述较大提升图的基础,其中,提升图描述了用于控制编码的码结构。

[0025] 考虑通过 $1, \dots, j, \dots, Z$ 来索引投影 LDPC 图,其中 j 是用作投影图索引的整数,并且 z 是用于定义提升图的投影图总数。在通过简单地复制投影图 Z 次来生成提升图的严格并行图中,图 j 中的变量节点仅连接到图 j 中的约束节点。即,在用于形成较大提升图的投影图的边之间,不存在相互影响。

[0026] 本发明的编码器不受限制于此。根据本发明,可能取一条向量边,其包括每条边来自 Z 个图副本的每一个的对应边,并允许在 Z 条边内的置换,例如,我们准许置换对应于该待置换向量边内的边的约束节点,例如,重排序。

[0027] 我们把置换限制在标记为 Ψ 的 $Z \times Z$ 置换矩阵集合(通常是群)空间内。因此,这里将使用 Ψ 来指通常是置换矩阵群的置换矩阵集合。我们假设 Ψ 中置换的逆还在 Ψ 中。通常能使用各种准则来选择 Ψ 集合。上面结构的主要目的之一是要简化解码器和编码器的硬件实现。因此,把 Ψ 限制为能在硬件中高效实现的置换是有益的,例如,在交换网中。

[0028] 根据本发明,并且在各个实施例中,LDPC 编码过程被展开为矩阵加法及乘法操作的有序序列,该 LDPC 编码过程可以转化为一系列命令。为了描述的方便,我们把示例性图 G 的这一系列编码命令标记为 G 的编码微码。然后,使用本发明的编码器,通过顺序执行 G 的微码来完成实际编码,该编码器根据在预先装载了例如将被编码的比特这样的信息比特的物理存储器上的微码,执行各种操作。每个命令包括操作符 op 和存储单元指示符。根据操作符 op ,编码器控制逻辑读出位于由存储单元指示符确定的存储器中的比特单元,并把它累加到寄存器,或者把寄存器值写到存储单元 a 并把寄存器值复位为零。微码的尺寸,即,其内的命令数,至少等于图 G 中的边数;通常,他们大致相同。

[0029] 考虑具有提升因子 Z 的提升 LDPC 图。给定将用于形成较大图的小的、投影的图,例如,投影图,通过用 $Z \times Z$ 矩阵替代 H 中的每个元素,我们能形成 Z 倍大的 LDPC 图。用标记为 0 的零矩阵替代 H 中的元素 0 。用来自 Ψ 的矩阵替代 H 中的每个元素 1 。在这种方式下,我们将 LDPC 图“提升”到 Z 倍大。表示的复杂度大致包括指定置换矩阵所需要的比特数, $|E_H| \log |\Psi|$ 加上表示 H 所需要的复杂度,其中 $|E_H|$ 标记 H 中 1 的数量,并且 $|\Psi|$

标记 Ψ 中不同置换数。例如,如果 Ψ 是循环置换的空间,那么 $|\Psi| = Z$ 。实际上,我们可以有,例如,对于 $n \approx 1000$, $Z = 16$,其中 n 是码字块长度。下面示出了提升小奇偶校验矩阵 H 的例子,其中,利用投影图来替换 H 中每个是一的元素,来产生在右面示出的较大的投影矩阵 H 。

$$[0030] \quad H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} \sigma_1 & 0 & \sigma_7 & \sigma_9 & \sigma_{11} & 0 & 0 \\ \sigma_2 & \sigma_4 & \sigma_8 & 0 & 0 & \sigma_{13} & 0 \\ \sigma_3 & \sigma_5 & 0 & \sigma_{10} & 0 & 0 & \sigma_{15} \\ 0 & \sigma_6 & 0 & 0 & \sigma_{12} & \sigma_{14} & \sigma_{16} \end{bmatrix}$$

[0031] 在矩阵 H 中, $\sigma_i, i = 1, \dots, 16$ 是这里示出的从变量节点侧索引的 Ψ 的元素(矩阵)。

[0032] 让我们回忆,当且仅当 $Hx = 0$ 时,向量 x 是码字。在提升矩阵表示中,把 x 看作 $GF(2^Z)$ 中元素的向量,替代二进制元素的向量,其中 $GF(2^Z)$ 是 2^Z 个元素的伽罗瓦域。就此而论,可以模仿作为在背景部分展开的矩阵-向量乘法和向量加法的编码过程:投影图中的矩阵中的每个非零元素 1 由它的相应 $Z \times Z$ 置换矩阵替代;向量中的每个比特由 Z 比特向量替代。

[0033] 使用 G 作为投影图的 LDPC 图的编码过程大部分被指定为上述用于投影图的编码过程的提升。这通过将原始算法中的比特操作替换为提升算法中的比特向量操作来完成。在编码处理中的一个或多个点,在从存储器读出后, Z 比特向量经受置换操作,例如,重排序操作。重排序操作可以是旋转操作,或简称旋转。这些旋转操作通常对应于与向量边相关联的旋转,该向量边互连投影图的 Z 个副本以形成单个大图。因此,在提升的微码中,每个命令包括操作符 op 、旋转数 r 、以及存储单元指示符。

[0034] 提升投影图的微码在很大程度上指定了用于提升图的编码。一个例外是,在投影图中存在矩阵求逆,即用于计算矩阵 Φ^{-1} 。在这样的情况下,该求逆不直接提升到提升图中的矩阵求逆。而是,在 $Z \times Z$ 置换矩阵的环中完成矩阵求逆,并且对应的编码命令产生一组指定矩阵求逆的新命令。在那些命令中,在适当的 LDPC 表示预处理之后,需要的旋转是显然的。

[0035] 因此,根据本发明用于描述提示图的微码尺寸大致等于投影图中边的数量加上通常很小的指定求逆的附加命令的数量。因此,对于相同块长度,增加提升因子通常会减少编码微码的尺寸。对于大的块长度,这可以显著节省微码描述存储器,例如,相对小的存储器可用于存储描述将用于编码的大图的信息。

[0036] 在实际通信或数据存储系统中,通常要求编码器的吞吐量匹配解码器的吞吐量。对于具有提升因子 Z 的提升图,对编码器和解码器实现硬件并行度 Z ,通常产生的编码器吞吐量比为了匹配解码器吞吐量所需的编码器吞吐量高,因为解码器需要多几个的迭代来收敛,例如,完成码字的解码。

[0037] 基于发明者提供的这个启示,应该理解,在给定时间周期处理的码字数量方面,期望编码器使用比解码器小的实现并行度,并达到相同吞吐量,利用本发明这是可能的。

[0038] 对于另一个实例,在描述大图时使用大的提升因子 Z 实现上面提到的节省存储器的好处的情况下,使用实现并行度 Z 引起高的并行单元数,生成过度的吞吐量。

[0039] 给定硬件复杂度正比于实现并行度 N 而图描述复杂度正比于提升因子 Z 的事实下,期望当使用以大提升因子 Z 描述的提升图时,由实现并行度得到的吞吐量匹配需求,但不过度。

[0040] 本发明的各种特征针对使用描述具有提升因子 Z 的提升图的微码,实现具有实现并行度 N 的向量 LDPC 编码器的方法和装置,其中, N 是 Z 的除数。选择实现并行度 N 以匹配所需吞吐量,因此使用最小硬件复杂性。

[0041] 此外,根据本发明,具有实现并行度 N 的向量 LDPC 编码器能够根据描述具有提升因子 Z 的提升图的相同微码,生成例如编码数据,以产生对应于一类具有相同速率但不同块尺寸的 LDPC 码的码字。特别地,作为实例,假定 Z 可用因子分解为 $K_1 \times K_2 \times N$, 并且投影图具有 n 个变量节点,则新颖编码器可以生成不同码字尺寸 $N \times n$, $K_2 \times N \times n$ 以及 $K_1 \times K_2 \times N \times n$ 的三个不同码。

[0042] 因此,使用本发明编码器的设备能,并且经常,根据应用,在使用存储的微码中描述的不同图进行编码之间切换,使得生成的码字是适于特定应用的块尺寸。例如,可以生成用于无线通信的第一尺寸的码字,而且可以生成用于光纤通信目的的第二尺寸的码字。

附图说明

[0043] 图 1 说明了长度为十的示例性规则 LDPC 码的二分图表示。

[0044] 图 2 是图 1 中以图说明的码的矩阵表示。

[0045] 图 3 说明了根据本发明实现的示例性编码器,并且也说明了根据本发明的编码方法生成的编码器内的各种信号和信息的流程。

[0046] 图 4 说明了例如移动节点的设备,其使用根据本发明实现的可编程 LDPC 编码器。

[0047] 图 5 包含图 5A 和图 5B 的结合,是运行根据本发明实现的示例性通信设备以执行根据本发明的编码和解码的示例性方法的流程图。

具体实施方式

[0048] 本发明针对用于实现 LDPC 编码器的方法和装置。本发明的各个实施例针对 LDPC 编码器和编码方法,该 LDPC 编码器和编码方法能以硬件高效的方式实现,同时关于能在编码期间使用的码结构和生成的码字尺寸相对灵活。

[0049] 图 3 说明了根据本发明实现的示例性的 LDPC 编码器 300。编码器包括图 3 中所示的耦合在一起的存储器模块 302、控制模块 312、基于码提升 (lifting) 的块选择模块 310、 N 元素可控制置换器 304、 N 元素向量累加器模块 306 以及可控制存储设备 308。注意,在本应用中术语置换器和变换器被交替使用来指相同的事物。以下将会详细地描述 LDPC 编码器 300 的各个单元和它们的功能。

[0050] 如上所讨论,使用不同的码描述和 / 或对于相同的码描述使用由不同提升因子指示的不同码字长度,本发明的编码器 300 能支持不同的码。存储器模块 302 包括一组 $K \times N \times L$ 存储单元 (318、320、322), 其中, $K \times N \times L$ 是最大可支持的码字尺寸。输入 301 允许将被编码的数据写入所述存储器。一旦编码完成时,输出 303 允许储存在存储器 314 中的码字被读出。存储器模块 302 还包括用于储存临时值的一组 $K \times N \times 1$ 个存储单元 (319、321、323)。其他实施例可以不需要和不使用临时存储值。使用存储器 314 也能支持较小尺

寸的码字。将存储器 314 中的存储单元排列成用于储存码字值的 K 个 $N \times L$ 块 Blk 1318、Blk 2320、Blk K 322, 以及用于储存临时值的 K 个 $N \times 1$ 块 Blk 1319、Blk 2321、Blk K 323。 $K \times N \times L$ 个存储单元中的每个存储单元通常是 1 比特。 $K \times N \times 1$ 个存储单元中的每个存储单元通常也是 1 比特。存储器寻址逻辑 316 控制从存储器 314 的存储单元读以及向存储器 314 的存储单元写, 该存储器寻址逻辑 316 响应于其他存储器组件生成的各种输入生成存储器访问信号 324 (地址和读 / 写信号)。通常一次从存储器模块 314 中读或写 N 个比特。 N 比特宽总线 340 将存储器模块 302 的 N 比特宽读输出耦合到 N 元素可控制置换器 304 的 N 比特宽输入, 其中, 在通过 N 比特宽总线 342 将比特提供给 N 元素向量累加器 306 前, N 元素可控制置换器 304 能重排序比特。 N 元素可控制置换器 304 接收重排序控制信号 $r2$ 373, 其是根据所储存的码描述信息, 例如, 诸如微码的控制码, 生成的。在比特被提供给 N 元素向量累加器模块 306 前, 如果需要, 信号 $r2$ 373 控制对从存储器获得的 N 个比特将要执行什么样的比特重排序。

[0051] N 元素向量累加器模块 306 包括并行排列的 N 个累加器电路。 N 个累加器电路中的每个生成来自 N 元素可控制置换器 304 的 N 个输入比特中的一个以及对应的从可控制存储设备 308 读出的 N 个比特中的一个的一比特模二和。这是执行 XOR 操作有效的方法。因此, 每个累加器电路执行一个 XOR 操作。在这种方式中, N 元素向量累加器 306 并行地生成 N 个累加值。由累加器模块 306 生成的 N 个值通过 N 比特宽总线 344 被并行地提供给可控制存储设备 308。可控制存储设备 308 包括输入 MUX 328、输出 MUX 308、以及一组 K 个 N 比特寄存器 326。当读 / 写控制信号 350 指示向量累加器模块的输出将要被储存到可控制存储设备 308 时, 由块选择控制信号 360 控制输入 MUX 328, 以确定将 N 比特块写入到 K 个 N 比特寄存器 332、334、336 中的哪一个。输出 MUX 330 耦合到 N 比特宽总线 346, 并且当读 / 写控制信号 350 指示将要执行读操作时, 输出 MUX 330 输出由块选择控制信号 360 指示的 N 比特块。将从可控制存储设备 308 读出的每组 N 个比特提供给存储器模块 302 和 N 元素向量累加器模块 306 的第二个输入。例如由存储的码描述确定的, 在累加器操作序列结束时将 N 比特写到存储器。

[0052] 控制模块 312 负责根据特定码描述生成多种控制信号, 所述特定码描述存储在编码器描述信息模块 372 中、被选择在特定时间点使用, 例如, 诸如微码的控制码。在可编程的实施例中, 例如, 经由输入 371, 从设备主存储器, 能将码描述信息装载到所存储编码器描述信息模块 372。在预装载和使用单一码描述的实施例中, 例如, 对于对应于相同码结构的不同长度的码字, 可以忽略输入 371。由外环计数器 374 生成的控制信号 375 来驱动由编码器描述信息模块 372 产生的信号的生成。由内环计数器 370 生成的内环控制信号 377 来驱动外环计数器 374。内环计数器 370 根据码提升因子控制信号 SK 348 生成第二选择模块控制信号 356 以及内环控制信号 377, 其中, 码提升因子控制信号 SK 348 作为控制值被提供给内环计数器 370。码提升因子控制信号用于指定将要生成的码字的长度, 并采取从 1 到 K 的值, 其中 K 指示在存储器 314 中的 $N \times L$ 比特块的总数。因此, 通过使用不同的码提升因子, 可以生成不同尺寸的码字, 其中每个不同可支持的码字尺寸将会是 $N \times L$ 的整数倍。在 $SK < K$ 的情况下, 通常不使用存储器 314 中的一个或多个块以及寄存器组 326 中的一个或多个寄存器。

[0053] 所存储编码器描述信息模块 372 包括例如微码的控制码。当响应于外环控制信号

375 执行该码时,这个码生成由在微码的执行行中包含的 op 值指定的读 / 写信号 350。将信号 350 提供给存储器模块 302 和可控制存储设备 308。所存储编码器描述信息模块 372 也生成;当要执行读 / 写操作时提供给存储器模块 302 的存储器地址控制信号 352、提供给基于码提升的块选择模块 310 的第一选择模块控制信号 r1354、以及提供给可控制置换器 304 以控制对从存储器模块 302 读出值的重排序的重排序控制信号 r2373。

[0054] 基于码提升的块选择模块 310 接收来自所存储编码器描述信息模块 372 的第一选择模块控制信号 r1354 和由内环计数器 370 生成的第二选择模块控制信号 356。基于码提升的块选择模块 310 生成提供给存储器寻址逻辑 316 的块地址选择信号 358,以指示将在特定时间点访问的存储器 314 的特定块。基于码提升的块选择模块 310 也生成块选择控制信号 360,其用于控制哪个信息块,例如,哪个寄存器 332、334、336 比特将在特定时间点可在控制存储设备 308 中被访问。

[0055] 由公共时钟信号驱动编码器 300 的各个组件,因此,各种操作和计数器 370、374 的递增以同步的方式工作。

[0056] 在多个处理相互作用后,作为在所存储编码器描述信息模块 372 指导下执行的编码器处理操作以及存储器访问操作的结果,最初存储在存储器模块 314 中的一组信息比特将被变换为码字。这个码字可以被读出,并且例如,被发送或存储。

[0057] 为了获得高错误复原级别,通常使用相对长的码字。例如,通过执行编码操作生成的单个码字可以包括总共 T 个比特,其中 T 可以是几百个或甚至几千个比特。为了解释本发明的目的,应该理解将被编码的比特可以被排列成 $K \times N \times L$ 比特向量,其中, N 是正整数,并且 K 是大于 1 的正整数。从存储器中读取每个 N 比特向量。然后能够并行地使用 N 个处理单元处理从存储器中读出的向量。现有系统在编码器中使用实现并行度 N 等于 Z,该编码器使用具有提升因子 Z 的特定提升 LDPC 码来编码码字,与现有系统相比,本发明允许编码器中的并行度级别不同于总的所支持的提升因子 Z。更特别地, $Z = K \times N$,其中 K 是大于 1 的整数。因此,根据本发明,在各种实现中,并行度 N 的级别低于提升因子 Z。此外,在一些实施例中,可以使用相同的码描述信息组能生成不同尺寸的码字。通过选择小于最大可支持提升因子控制值 K 的码提升因子控制值 SK,对于给定实现可以创建小于最大码字尺寸 ($L \times K \times N$) 的码字。不同尺寸的码字将会是 $N \times L$ 比特的倍数。

[0058] 2004 年 2 月 26 日提交的、名为“METHOD AND APPARATUS FOR PERFORMING LOW-DENSITY PARITY-CHECK (LDPC) CODE OPERATIONS USING A MULTI-LEVEL PERMUTATION”的美国专利申请 S. N. 10/788, 115 以及相应的具有相同标题和提交日期的 PCT 申请 PCT/US2004/005783,在此被明确地引入作为参考。这些专利申请描述了 LDPC 码的乘积提升方法。这些乘积提升方法将在提升中使用的 $Z \times Z$ 置换矩阵群限定为能被分解成为子群的直接乘积的群。例如,我们假设 Ψ 是三个子群的直接乘积,即, $\Psi = \Psi_1 \times \Psi_2 \times \Psi_3$ 。 Ψ 的维数等于 Ψ_i 维数的乘积,其中 Ψ_i 是 $K_i \times K_i$ 置换矩阵群。因此,大的提升可以作为多个小的顺序提升来实现。假设群 Ψ_i 的维数等于该群内矩阵的维数,因此, $Z = K_1 \times K_2 \times K_3$,其中, K_1, K_2, K_3 分别是 Ψ_1, Ψ_2, Ψ_3 的维数。

[0059] 根据本发明,我们将提升群 Ψ 限定为乘积提升群。如上面提到的,乘积提升能同等地被看作多维提升。因此,本发明的当前编码器 300 使用的提升能作为多维提升实现。假设投影码的尺寸是 P,即,具有 P 个变量节点。可以选择用于提升的尺寸为 64 的循环群。

根据本发明,备选是尺寸为 16 的循环群和尺寸为 4 的循环群的乘积(注意 $16 \times 4 = 64$)。这个群能由以下表示。使用对 (a, b) , $a = 0, \dots, 15$ 并且 $b = 0, \dots, 3$, 通过可逆映射 $L = 4a+b$, 考虑索引 $L = 0, \dots, 63$ 。这个乘积群的一个元素是一对 (c, d) , $c = 0, \dots, 15$ 并且 $d = 0, \dots, 3$ 。 (c, d) 对 (a, b) 的作用是把对 (a, b) 置换为 $(a+c \text{ 模 } 16, d+b \text{ 模 } 4)$ 。这个群也具有阶 64。然而,得到的提升图能被解释为尺寸为 $4P$ 的码乘以 16、或尺寸为 $16P$ 的码乘以 4、或尺寸为 P 的码乘以 64 的提升。

[0060] 在本发明的编码器和硬件实现的上下文中,实现由乘积提升提供的各种优点。在编码器中通过使用乘积提升增加的值是本发明许多特征之一。由不是乘积的群的提升,例如,由循环群的提升,允许任意尺寸的提升,但不能提供乘积提升的灵活性。

[0061] 名为“METHOD AND APPARATUS FOR PERFORMING LOW-DENSITY PARITY-CHECK (LDPC) CODE OPERATIONS USING A MULTI-LEVEL PERMUTATION”的美国专利申请 S. N. 10/788, 115 描述了乘积提升图和使用这些图的可能益处。

[0062] 通过描述使用提升因子 $Z = K \times N$ 的新颖编码器 300, 本发明扩展了在该申请中描述的一些基本概念。本发明的各个特征针对于以灵活但相对硬件高效的方式,利用实现并行度 N 来对图编码的方法和装置。当 N 是固定的时, K 能被用作提升控制因子,并能指示将要生成的码字的尺寸。

[0063] 我们假定具有提升因子 $Z = K \times N$ 的提升 LDPC 图。提升群 Ψ 是乘积提升群 $\Psi = \Psi_1 \times \Psi_2$, 其中 K 是群 Ψ_1 的维数,并且 N 是群 Ψ_2 的维数。我们能生成具有提升因子 Z 的提升图的微码,微码是一系列命令,其中每个命令包含操作符 op 、旋转数 r 、以及存储单元 a 。使用并行度 Z 实现的编码器 300 用以下方式执行每个命令:如果 op 指示读,那么控制器在存储单元 a 处从存储器读 Z 比特向量,用量 r 对其重排序,并将重排序的值累加到 Z 比特寄存器;如果 op 指示写,那么控制器将 Z 比特寄存器的值写入存储器的存储单元 a 处。并且通过执行整个系列的命令完成编码。

[0064] 执行的微码可以存储在编码器描述信息模块 372 中,除了存储要被使用的微码外,编码器描述信息模块 372 还负责访问并执行包含在微码中的指令。

[0065] 使用并行度 N , 即, N 个并行处理单元,而不是并行度 $Z = K \times N$, 能获得执行例如本发明命令的微码指令的相同结果。然而,在我们的 N 并行实现中,我们执行相同的基本命令 K 次,每次完成处理 Z 个比特工作的 $1/K$ 。

[0066] 为了更好地理解本发明的编码过程,让我们首先考虑一个命令,该命令从存储单元 a 中读出 Z 比特向量,并且通过量 r 重排序,并且然后将重排序的值累加到 Z 比特寄存器。我们写出原始数据向量 $d = (d_1, d_2, \dots, d_k)$, 每个 d_j 是 N 比特向量,其中 j 是用作索引的整数值。给定提升群是乘积提升 $\Psi = \Psi_1 \times \Psi_2$, 其中 Ψ_1 具有维数 K 并且 Ψ_2 具有维数 N , 让我们写出重排序量 $r = (r_1, r_2)$, 其中 r_1 是群 Ψ_1 中的重排序量,例如循环旋转量,并且 r_2 是群 Ψ_2 中的重排序量,例如循环旋转量。我们使用标记 $\Psi_1(d, r)$ 来表示通过量 r 对群 Ψ_1 中的向量 d (K 个元素) 的重排序,以及使用标记 $\Psi_2(d, r)$ 来表示通过量 r 对群 Ψ_2 中的向量 d (N 个元素) 的重排序。也能把重排序认为是位置置换,因此,在最初位置 j 中的元素 d_j 转到在重排序数据中标记为 $\Psi_{1,r}(j)$ 的新位置。那么能将重排序认为是 2 级重排序过程。第一级在群 Ψ_2 中对 N 个 (1 比特) 元素重排序来生成向量 $d' = (\Psi_2(d_1, r_2), \Psi_2(d_2, r_2), \dots, \Psi_2(d_k, r_2))$ 。然后第二级在群 Ψ_1 中对 K 个 (N 比特) 元素重排序来生成向量 d''

$= \Psi_1(d', r_1)$ 。然后将重排序的数据 d' 累加到 Z 比特寄存器。如以下将要讨论的,在图 3 的实现中, Z 比特寄存器是作为一组 K 个 N 比特寄存器 332、334、336 实现的。

[0067] 我们现在描述如何将上述提及的使用并行度 Z 的读-重排序-累加的单个步骤分解成为使用并行度 N 的读-重排序-累加的 K 个步骤,以实现诸如 LDPC 编码器 300 的 LDPC 编码器。它是由编码器 300 使用的一系列这样的 K 个步骤。假设我们有 K 个寄存器 332、334、336 并假设将存储单元 a 处的 Z 比特向量 d 物理排列为 K 个 N 比特向量 (d_1, d_2, \dots, d_k) ,其中 N 比特向量 d_j 存储在块 j 中的单元 a 处。如果在步骤 j ,我们在由 a 和 j 确定的地址读出数据 d_j ,并例如使用 N 元素可控制置换器 304,通过群 Ψ_2 中的量 r_2 重排序读出的数据,生成 $\Psi_2(d_j, r_2)$ 。然后,我们将重排序的数据累加到这 K 个寄存器 332、334、336 的第 $\Psi_{1,r_1}(j)$ 个寄存器中。这完成了第 j 步。通过运行 $j = 1, \dots, k$,我们获得与利用 Z 比特向量操作执行命令相同的结果,但是利用较低的编码器并行度 N 达到这个结果,其中, $N < Z$ 。

[0068] 实际上以非常简单的方式,根据本发明,把 Z 比特寄存器写到存储单元 a 并复位 Z 比特寄存器的命令也能被分解为 K 步。在步骤 j ,我们将 K 个 N 比特寄存器 332、334、336 的第 j 个寄存器写到由 j 和存储单元 a 确定的存储单元,并且复位该寄存器 332、334 或 336。通过运行 $j = 1, \dots, k$,我们获得与利用 Z 比特向量操作执行命令相同的结果。

[0069] 以上讨论在原理上描述了我们如何使用并行度 N 来实现对应于提升因子 Z 的微码,例如,储存在编码器描述信息模块 372 中的微码。

[0070] 参考图 3,我们现在将更加详细地描述用于实现 K 步编码过程的示例性编码器 300,当使用 $Z > N$ 的提升因子 Z 时,所述 K 步编码过程允许并行度 N 。

[0071] 在图 300 中,控制模块 312 通过内环计数器 370 来控制执行命令的步骤数。在每个步骤,计数器 370 增加 1 并且在达到由码提升因子控制信号 348 确定的最大计数时复位。每次内环计数器 370 到达最大值时,它触发外环计数器 374 增加 1。通过访问所存储编码器描述信息模块 372,外环计数器 374 确定当前所执行的编码命令。所存储编码器描述信息模块 372 以依据该命令生成的各种信号的形式输出该命令,该命令将被应用于由外环计数器 374 确定的存储单元。命令包含操作符 op 、重排序量 r 、以及存储单元 a 。操作 op 指定对存储器模块 302 的读/写信号 350,存储单元 a 确定耦合到存储器模块 302 的存储器地址控制信号 352,并且重排序量 r 分成两部分 (r_1, r_2) ,每部分分别指定在群 Ψ_1 和 Ψ_2 中的重排序元素。将信号 r_1 354 提供给用于生成块地址选择信号 358 的基于码提升的块选择模块 310,该块地址选择信号 358 用于控制访问存储器模块 302 中的哪个存储块,同时将信号 r_2 373 提供给置换器 304,以控制对从存储器 302 中读出块的 N 个元素的置换。

[0072] 为了接收信号 r_1 354,基于码提升的块选择模块 310 具有第一选择模块控制信号 354,其耦合到来自所述控制模块 312 的命令中的重排序量 r 的 r_1 部分。由控制模块 312 的内环计数器 370 生成的第二选择控制信号 356 驱动,并由控制信号 r_1 354 控制,基于码提升的块选择模块 310 输出取值从 1 到 K 的块地址选择信号 358,以及取值 $\Psi_{1,r_1}(1), \Psi_{1,r_1}(2), \dots, \Psi_{1,r_1}(K)$ 的块选择控制信号 360。

[0073] 存储器模块 302 具有耦合到所存储编码器描述信息的操作符 op 输出的用于接收读/写信号 350 的输入,以及用于接收存储器地址控制信号 352 的另一个输入,存储器地址控制信号 352 对应于在特定时间点被执行的存储在模块 372 中的微码指令中包含的存储单元 a 。

[0074] 存储器模块 302 包括存储器 314, 其排列成 $K \times (N \times L)$ 个 1 比特存储单元 318、320、322 以及 $K \times (N \times 1)$ 个 1 比特存储单元 319、321、323。为了方便, 我们把具有 K 块 $(N \times L)$ 个 1 比特单元的存储单元标识为用于码字存储的块 1, \dots , K , 并且我们把具有 K 块 $(N \times 1)$ 个 1 比特单元的存储单元标识为用于值的临时存储的块 1, \dots , K 。在作为存储地址控制信号 a 352 和块地址选择信号 k 358 的函数的存储单元处访问存储器 314。存储器寻址逻辑模块 316 实现这样的函数。给定 (a, k) , 根据读 / 写信号 350 指示应该执行读还是写操作, 存储器模块 302 在第 k 个块中的单元 a 上读或写 N 比特向量。

[0075] 存储器模块 302 的读出操作输出从存储器 314 中读出的 N 比特向量 340。将这 N 比特向量馈入到 N 元素可控制置换器模块 304。模块 304 实现群 Ψ_2 中的重排序; 它的重排序控制信号耦合到所存储编码器描述信息模块 372 的 r_2 信号输出。 r_2 信号导出自重排序数 r , 从在特定时间点执行的来自信息模块 372 的微码命令获得在该特定时间点使用的重排序量 r 。

[0076] 置换器模块 304 的重排序 N 比特向量输出耦合到 N 元素向量累加器模块 306 的第一个 N 比特向量输入 342。累加器模块 306 的第二个 N 比特向量输入 346 是从包括 K 个 N 比特寄存器 332、334、336 的可控制存储设备模块 308 馈入的。向量累加器模块 306 生成作为两个 N 比特向量输入的 XOR 和的 N 比特向量输出。在各个实施例中, 使用并行排列的 N 个 XOR 电路实现向量累加器模块 306, 其中, 每个 XOR 电路耦合到一个不同加法器, 该加法器用于将由 N 个 XOR 电路的特定一个产生的 XOR 操作结果与最近生成的 XOR 结果相加。将累加器模块的 N 比特宽输出耦合到可控制存储设备模块 308 的输入 344。

[0077] 可控制存储设备模块 308 包括 K 个寄存器, 每个寄存器存储 N 个比特。耦合到基于码提升的块选择模块 310 的块选择控制信号 360 确定在特定时间点要访问 K 个寄存器中的哪一个。耦合到来自控制模块 312 的命令中包含的操作符 op 的读 / 写控制信号 350 确定访问模式, 例如, 读或写访问模式。假定块选择控制信号 360 指示 j 。如果控制信号是读, 则来自可控制存储设备模块 308 的 N 比特输出向量取第 j 个寄存器的值, 并将来自 N 元素向量累加器模块 306 的累加值写到第 j 个寄存器。换句话说, 将来自 N 元素可控制置换器模块 304 的重排序值累加到由块选择控制信号 360 指示的第 j 个寄存器。如果读 / 写控制信号 350 是写, 输出向量再次假设第 j 个寄存器的值, 并且我们然后复位第 j 个寄存器为零。

[0078] 概括地, 给定具有提升因子 $Z = K \times N$ 的提升图的微码, 本发明的各个实施例针对执行 N 比特向量操作的编码器。每个 N 比特向量操作包括微码中的命令的执行, 该微码描述用于编码的码结构。为了实现包括 Z 比特的码字的编码, 在顺序的 K 个步骤中实现每个 N 比特命令, 该 K 个步骤由部分所存储微码命令信息和一个或多个计数器控制。

[0079] 在本发明的各个实施例中, 所提出的编码器能生成不同的码, 这些码共享与投影图相同的速率, 但具有不同码字长度。这通过使用选择的提升因子控制值 SK 作为对于每个命令执行的步骤数来完成, SK 是 K 的除数而不是 K 本身。更特别地, 在乘积提升中的群 Ψ_1 是两个群的直接乘积 $\Psi_1 = \Psi_{11} \times \Psi_{12}$, 并且 SK 是矩阵 Ψ_{12} 维数, 并且 J 是 Ψ_{11} 的维数, 因此, $K = J \times SK$ 。作为特殊情况, Ψ_{11} 可能是单一元素 1 的群以及 Ψ_{12} 是 Ψ_1 , 因此 $SK = K$ 并且 $J = 1$ 。无论如何, 在提升图中, 如果我们忽略提升图内的 Ψ_{11} 分量, 则我们有具有提升因子 $Z/J = SK \times N$ 的提升图。不同的理解方法是我们采用原始图并将它投影到提升群 Ψ_{11} 上, 因此, 在奇偶校验矩阵中, 指示 $Z \times Z$ 置换矩阵的每个非零项现在被投影到 $Z/J \times Z/J$ 置

换矩阵。这样,根据群论的基本原理,即使对于矩阵逆 Φ^{-1} ,在较大图中的与矩阵乘法相同的解码过程序列对于投影图仍然成立。

[0080] 因此,描述具有提升因子 Z 的较大图的微码也是描述具有提升因子 $Z/J = SK \times N$ 的投影图的微码。与以上提及的关于 Z 的情况相同的推理路线,通过按照由存储在所存储编码器描述信息模块 372 中的部分命令信息控制的顺序,在 SK 个步骤中执行微码内的每个命令,我们能使用具有 N 比特向量操作的相同编码器来编码具有提升因子 $SK \times N$ 的码。

[0081] 如果 Ψ_1 仍然能写为两个其他群的直接乘积 $\Psi_1 = \Psi_{11}' \times \Psi_{12}'$,那么存在共用相同微码的不同块长度的其他码,其是本发明的各个实现中的情况。根据本发明,通过指定相应的 SK ,具有并行度 N 的相同编码器能编码具有提升因子 Z/J' 的码,其中, J' 是 Ψ_{11}' 的维数。 Ψ_1 中更多的附加结构可以导致更多的不同块长度的码可在相同编码器硬件上编码。因此,通过依据群结构控制 SK ,编码器能生成具有不同块长度的一类 LDPC 码。

[0082] 在图 300 中,能通过码提升因子控制信号 348 指定选择的码提升因子,由于该选择的码提升因子将控制生成的码字的长度,因而能作为码字长度选择信号。信号 348 被提供给内环计数器 370 并确定内环计数器 370 的最大计数。

[0083] 本发明编码器的灵活性和它能通过使用控制码实现的方便性,将由于以下的控制码实例而显而易见。以下的码可作为码描述信息,码描述信息被存储在所存储编码器描述信息模块 372 中,并被执行,例如,每次一行,以实现编码操作。

[0084] 以下包含表 1A 与 1B 组合的表 1 列出了与具有最大提升因子 $Z = 64$ 的码结构对应的例如微码的示例性控制码的指令。在实例中,对于其中 $N = 16$, $K = 4$ 以及 $L = 10$ 的情况设计码。 $Z = K \times N$,并且因此 $Z = 64 = 4 \times 16$ 。最大可支持码字长度将是 $K \times N \times L$,其在这个实例中是 640。由具有 4 个校验节点和 10 个变量节点的码来描述投影图。当由最大提升因子 Z 提升时,这将导致具有 256 (64×4) 个校验节点和 640 (64×10) 个变量节点的码结构。将硬件设计为在并行度 N 的级别操作,其中 $N = 16$ 。在这个实施例中用于指定码字长度的提升因子控制信号 SK 能够是 K 的任何除数,其中如上提到 K 是 4 ($K = Z/N = 64/16$),即,最大可支持提升因子控制值。因此,通过选择不同的提升因子控制信号以控制表 1 的微码的重复数,编码对应于 3 个不同长度的码字是可能的,例如,当 $SK = 1$ 时,码字长度将等于 $(1 \times 16 \times 10)$ 160 比特,当 $SK = 2$ 时,码字长度将等于 $(2 \times 16 \times 10)$ 320 比特,并且当 $SK = K = 4$ 时,码字长度将等于 $(4 \times 16 \times 10)$ 640 比特。在表 1 中示出的微码中,在 op 列中,1 用于指示读指令而 0 用于指示写指令。控制值 $r1$ 和 $r2$ 都存储在值 r 中。从值 r 中确定 $r1$ 为当 r 被 N 除时的整除结果。即, $r1 = r$ 除以 N 。通过取 r/N 的模数从值 r 中确定 $r2$ 。在这个实例中 $N = 16$ 。考虑例如第一个指令 1434。这个指令应该解释为读 ($op = 1$) 指令, $r1 = (r \text{ 除以 } N) = (43 \text{ 除以 } 16) = 2$,而 $r2 = (r \text{ 模 } N) = (43 \text{ 模 } 16) = 11$ 。控制值 a 由表直接提供,并且在第一个指令的情况下是 4。当控制值在 $0, \dots, L-1$ 的范围内时,例如,对于该示例性微码为 $0 \dots 9$,被访问的存储器用于码字存储,例如,访问 K 个块 318、320、322 中的一个。当控制值在 $0, \dots, L-1$ 的范围外时,例如,对于该示例性微码为 10,被访问的存储器用于值的临时存储,例如,访问 K 个块 319、321、323 中的一个。

[0085] (开始)

[0086]

<u>op</u>	<u>r</u>	<u>a</u>
1	43	4
1	5	5
1	6	7
1	44	8
1	36	3
0	0	2
1	10	4
1	30	5
1	47	6
1	9	7
1	17	3
0	0	1
1	25	5
1	32	6
1	58	8
1	45	9
1	16	2
0	0	0
1	42	4
1	17	8
1	62	9
1	6	0

1	38	1
0	0	10
1	17	10
1	19	10
1	21	10
1	29	10
1	31	10

[0087] 表格 1A

[0088] (继续)

[0089]

<u>op</u>	<u>r</u>	<u>a</u>
1	49	10
1	50	10
1	51	10
1	52	10
1	53	10
1	54	10
1	55	10
1	56	10
1	58	10
1	63	10
0	0	3
1	43	4

1	5	5
1	6	7
1	44	8
1	36	3
0	0	2
1	10	4
1	30	5
1	47	6
1	9	7
1	17	3
0	0	1
1	25	5
1	32	6
1	58	8
1	45	9
1	16	2
0	0	0

[0090] 表格 1B

[0091] 本发明的方法和装置能用于实现广泛的各种设备,包括,例如,无线终端、基站、数据存储设备以及可能使用编码和 / 或解码数据以防止和 / 或纠正错误的其他类型设备。

[0092] 图 4 是根据使用本发明方法的 LDPC 编码器 / 解码器设备实现的例如移动节点的示例性无线终端 (WT) 1000 的图。示例性 WT 1000 包括接收机 1002、接收机天线 1004、可编程 LDPC 解码器 1006、发射机 1008、发射机天线 1010、可编程 LDPC 编码器 1012、处理器 1014、用户 I/O 设备 1015 以及存储器 1016。经由总线 1018 将可编程 LDPC 解码器 1006、可编程 LDPC 编码器 1012(其能使用图 3 的编码器 300 实现)、处理器 1014、用户 I/O 设备 1015 以及存储器 1016 耦合到一起,各种元件可以通过总线 1018 相互交换数据和信息。

[0093] 将接收机 1002 耦合到接收机天线 1004,经由接收机天线 1004,WT 1000 可以接收来自其他设备的信号,例如,来自基站的已编码下行链路信号。接收机 1002 也耦合到可编

程 LDPC 解码器 1006,其可以根据本发明解码所接收的下行链路信号。除了例如 LDPC 编码数据之外,接收的信号还可以包括例如控制信息的信号,该信号用于指示用于编码正被接收的数据的 LDPC 码结构和 / 或包括在接收数据中的码字的码字长度。接收的数据可以包括对应于不同应用的码字。根据本发明,解码器可以从解码对应于第一个码结构和码字长度的数据切换到解码对应于第二个码结构和第二个码字长度的数据。第一个和第二个码字结构可以不同,响应于包含在接收信息中的信息,用例如微码形式的控制码这样的适当码结构信息装载解码器。通常不使用 LDPC 码来编码控制信息,以便于控制信息的快速检测和解释。第一个和第二个码字长度也可以不同。在一些情况下,第一个和第二个码结构是相同的,但是对应于不同应用的数据的码字长度可以是不同的。在这样的情况下,不需要更新码结构信息来解码不同尺寸的码字,并且,随着接收数据的码字长度改变,仅需要将例如提升因子信息的码字长度信息提供给解码器。可以将码字长度信息指定为正在使用的码结构的码提升因子。如以下将要讨论的,例如控制码的码结构信息能用于控制可编程 LDPC 解码器,而码字长度信息能用于设置码字长度,用于解码目的。这样的信息经由总线 1018 能从存储器 1016 传送到解码器 1006。

[0094] 将发射机 1008 耦合到发射机天线 1010, WT 1000 可以通过发射机天线 1010 向基站发射包括已编码的上行链路信号的上行链路信号。将发射机 1008 耦合到可编程 LDPC 编码器 1012,可编程 LDPC 编码器 1012 在传输前编码各种上行链路信号,例如,对应于不同应用的数据信号。给编码器装载对应于不同码结构的不同组码描述信息,例如,诸如微码的不同组控制码。另外,将用于控制由编码器 1012 生成的码字的长度的码字长度信息,例如,以码提升因子信息的形式,提供给编码器 1012。可以从接收的信息中获得选择码字结构和 / 或码字长度的信息,例如,编码器可以使用与用于解码的相同码字结构和码字长度来编码由应用生成的数据,该相同码字结构和码字长度被用于解码为该生成数据的特定应用接收的数据。因此,可以编程编码器,以匹配由与无线终端相互作用的另一个设备使用的编码结构和码字长度。或者,设备用户可以指定使用特定的码字结构和 / 或码字长度,或者这样的信息可由通信程序或存储在无线终端中的其他程序指定。

[0095] 经由总线 1018,能将码结构信息和 / 或码字长度信息从存储器 1016 传送到可编程 LDPC 编码器 1012。例如键盘、扬声器、传声器、显示器等的用户 I/O 设备 1015 提供接口,以使用户输入数据和信息,例如,将被编码并传送到另一个 WT 的数据和信息,以及使用户输出和 / 或显示接收的数据 / 信息,例如,来自同等节点的已经被解码的接收数据和信息。用户 I/O 设备 1015 提供接口,以允许用户选择和 / 或指定将由可编程 LDPC 解码器 1006 和 / 或可编程 LDPC 编码器 1012 使用的与一组数据相关联的码、码长度指示符、和 / 或多组码描述信息。

[0096] 处理器 1014,例如 CPU,执行程序并使用存储器 1016 中的数据 / 信息,以控制无线终端 1000 的操作并实现本发明的方法。

[0097] 存储器 1016 包括编码器码描述信息组 1026、1028 的群 1025,以及解码器码描述信息组 1030、1032 的群 1029。每个编码器码描述信息组 1026、1028 包括反映将被用于对数据编码的码的码结构的控制码,例如微码。每组信息 1026、1028 对应不同的码结构。编码器码描述信息能被装载到可编程 LDPC 编码器 1012 的编码器控制模块,并例如作为存储的编码器描述信息,用于控制数据的编码。类似地,每个解码器码描述信息组 1030、1032 包括反

映将被用于对数据解码的码的码结构的控制码,例如微码。每组解码器码描述信息 1030、1032 对应不同码结构。解码器码描述信息能被装载到可编程 LDPC 解码器 1006 的控制模块,并例如作为存储的解码器描述信息,用于控制数据的解码。

[0098] 存储器 1016 包括通信程序 1020、编码器码和码字长度选择程序 1022、以及解码器码和码字长度选择程序 1024。通信程序 1020 可以控制与其他无线设备通常的通信和相互作用。当使用 LDPC 码编码和 / 或解码数据时,对于给定应用实现的通信程序可以指定将用于特定通信应用的码结构和 / 或码字长度。编码器码和码字选择程序 1022 负责选择将用于特定应用的码结构以及因此相应的编码器码描述信息 1026、1028。基于从通信程序 1020 接收的信息、经由接收机 1002 或从用户输入接收的信息来作出这个选择。如果编码器码和码字长度选择程序 1022 还没有被配置为根据选择的码和码字长度执行编码,那么它负责用选择的码描述信息装载可编程 LDPC 编码器 1012,并负责将例如选择的码提升因子的信息提供给可编程编码器 1012。如果解码器码和码字长度选择程序 1024 还没有被配置为根据选择的码和码字长度执行解码,那么它负责用选择的码描述信息装载可编程 LDPC 解码器 1006,并负责将例如选择的码提升因子的信息提供给可编程解码器 1006。

[0099] 除了以上讨论的涉及编码和解码的程序和信息外,存储器 1016 还可用于存储接收的解码器信息 1038,例如,由指示将用于解码的码结构和码字长度的解码器码和码字长度选择程序 1024 使用的接收信息。另外,接收的编码器信息 1044,例如,由指示将用于编码的码结构和码字长度的编码器码和码字长度选择程序 1022 使用的接收的信息,可存储在存储器 1016 中。涉及解码的用户输入信息 1036 以及涉及编码的用户输入信息 1042 也存储在存储器 1016 中。这样的信息与解码器信息 1038 和编码器信息 1044 相同或类似,但是它经由用户 I/O 设备 1015 而不是经由接收机 1002 从用户获得。

[0100] 对于能够并且在一些实施例中用作可编程 LDPC 解码器 1006 的示例性可编程 LDPC 解码器的详细讨论,见具有 2005 年 7 月 21 日的提交日期、发明者名字为 Tom Richardson、Hui Jin 和 Vladimir Novichkov、名为“LDPC DECODING METHODS AND APPARATUS”的美国专利申请 S. N. 10/895,645,在此被明确地引入作为参考。为了提供背景信息的目的,美国专利 NO. 6,633,856 也明确地引入作为参考。

[0101] 使用上述讨论的程序和存储的编码器 / 解码器信息,根据例如用户信息或经由接收机 1002 接收的信息这样的接收信息,为了编码和解码目的,无线终端能在使用不同码结构和码字长度间切换。也可由在特定时间点执行的特定通信程序 1020 来触发编码器 / 解码器改变。因此,利用通过码更新的使用而容易被修改的单个一组硬件,本发明在 LDPC 码的编码和解码中允许很大的灵活性,该码更新对应于不同码结构和 / 或码字长度控制参数的改变。

[0102] 涉及本发明依赖的 LDPC 码结构的各个概念在 2003 年 7 月 11 日提交的,名为“METHOD AND APPARATUS FOR ENCODING LDPC CODES”的美国专利申请 S. N. 10/618,325 中描述和讨论,并在此被明确地引入作为参考。当按照引入的专利申请中提供的讨论来看时,能获得对本发明的方法和装置的技术和好处的更好评价。

[0103] 包括图 5A 和图 5B 组合的图 5 是操作根据本发明实现的例如 WT1000 的示例性通信设备以执行根据本发明的编码和解码的示例性方法的流程图 1100。操作在步骤 1102 开始,在其中 WT 1000 接通电源并初始化。操作从步骤 1102 继续进行到步骤 1104、1106、以及

步骤 1108。

[0104] 在步骤 1104,操作 WT 1000 接收编码 / 解码信息和 / 或从接收的数据生成控制信息。可以经由通过接收机 1002 处理的接收信号和 / 或经由用户 I/O 设备 1015 接收的用户输入来接收编码 / 解码信息,例如,用于可编程 LDPC 编码器 1012 和 / 或可编程 LDPC 解码器的控制信息。另外,处理接收的编码数据以生成控制信息。例如,使用不同的码结构信息和 / 或不同码字长度能执行解码的多次尝试。一旦成功解码,在一些实施例中生成控制信息,该控制信息指示将用于解码到来数据以及在一些实施例中编码输出数据的码结构和 / 或码字长度。经由连接节点 A 1110,操作从步骤 1104 继续进行到步骤 1112。在步骤 1112,操作 WT 1000 以确定接收的编码 / 解码控制信息的类型。基于步骤 1112 的确定,操作继续进行到步骤 1114、1116、1118 或 1120。

[0105] 如果在步骤 1112 确定控制信息的类型是编码器码结构信息,则操作继续进行到步骤 1114。在步骤 1114,操作 WT 1000 来用一组码描述信息,例如对应于由控制信息指示的码结构信息的控制码,装载编码器 1012。操作从步骤 1114 继续进行到连接节点 B 1122。

[0106] 如果在步骤 1112 确定信息的类型是编码器码字长度信息,则操作继续进行到步骤 1116。在步骤 1116,操作 WT 1000 将对应于由控制信息指示的码字长度的码字长度指示符,例如选择的提升因子,提供给编码器 1012。操作从步骤 1116 继续进行到连接节点 B 1122。

[0107] 如果在步骤 1112 确定控制信息的类型是解码器码结构信息,则操作继续进行到步骤 1118。在步骤 1118,操作 WT 1000 来用一组码描述信息,例如对应于由控制信息指示的码结构的控制码,装载解码器 1006。操作从步骤 1118 继续进行到连接节点 B 1122。

[0108] 如果在步骤 1112 确定信息的类型是解码器码字长度信息,则操作继续进行到步骤 1120。在步骤 1120,操作 WT 1000 将对应于指示码字长度的码字长度指示符,例如选择的提升因子,提供给解码器 1006。操作从步骤 1120 继续进行到连接节点 B 1122。

[0109] 操作从连接节点 B 1122 返回到步骤 1104,其中,WT 1104 等待接收其他编码 / 解码信息,例如,完成可编程解码器 1006 和 / 或可编程编码器 1012 的配置的信息,和 / 或改变解码器 1006 和 / 或编码器 1012 的选择的设置的信息,例如,码字长度设置。

[0110] 在步骤 1106,操作包括先前配置的可编程解码器 1006 的 WT1000,通过接收机 1002 接收将被解码的数据,例如,来自 WT 1000 的同等节点的已编码用户数据。将接收的数据转发给解码器 1006。操作从步骤 1106 继续进行到步骤 1124。在步骤 1124,依据在解码器 1006 中的所存储码描述信息和已经被提供给解码器的码字长度指示符信息,操作解码器 1006 以解码数据。操作从步骤 1124 继续进行到步骤 1106,在步骤 1106 中接收将被解码的附加数据。

[0111] 在步骤 1108,操作包括先前配置的可编程编码器 1012 的 WT1000,通过用户 I/O 设备 1015 接收将被编码的数据,例如,来自 WT1000 用户的将要被编码并被传送给 WT 1000 的同等节点的输入数据。将接收的数据转发给编码器 1012。操作从步骤 1108 继续进行到步骤 1126。在步骤 1126,依据所存储码描述信息和被提供给编码器的码字长度指示符信息,操作编码器 1012 以编码数据。操作从步骤 1126 继续进行到步骤 1108,在步骤 1108 中接收将被编码的附加数据。

[0112] 在时间上,随着被装载到编码器 1012 和解码器 1006 的、对应于例如选择的提升因

子这样的码字长度信息的控制信息改变,码字长度将会改变。因此,随着无线终端从对应于第一个设备和 / 或应用而接收数据切换到对应于第二个设备和 / 或应用而处理数据,码字长度能,并且在各个实现中将改变。另外,在时间上随着无线终端与不同的设备相互作用和 / 或实现不同的应用,能改变由编码器 1012 和 / 或解码器 1006 使用的码结构信息。因此,在第一个时间点,编码器和解码器可以处理对应第一个长度和 / 或码结构的码字,并且在另一个时间处理对应于第二个长度和 / 或码结构的码字。在其他的时间点,本发明的可编程 LDPC 编码器 1012 和解码器 1006 可以使用其他的码结构和 / 或码字长度。各个可支持的码字长度通常将会达到由编码器 1012 和解码器 1006 中可用存储器量和 / 或可用寄存器数量和尺寸确定的最大尺寸。

[0113] 以下的专利申请和专利提供关于编码和 / 或解码 LDPC 码的信息,并在此被明确地引入作为参考。2004 年 2 月 26 日提交的美国专利申请 S. N. 10/788, 115 ;2002 年 4 月 4 日提交的美国专利申请 S. N. 10/177, 264 ;美国专利申请 S. N. 10/618, 325 以及美国专利 6, 633, 856。

[0114] 关于本发明的方法和装置的众多改变是可能的。因此,用于实现本发明的模块可以作为软件、硬件、或软件及硬件的组合实现。例如,可以用硬件和 / 或软件实现本发明的各种特征。例如,本发明的一些方面可以实现为处理器执行的程序指令。或者,或另外,本发明的一些方面可以实现为例如,诸如 ASIC 的集成电路。本发明的装置针对软件、硬件和 / 或软件与硬件的组合。设想了包括用于控制机器来实现根据本发明的一个或多个方法步骤的指令的机器可读媒介,其被认为在本发明的一些实施例的范围内。

[0115] 本发明尤其针对能用于控制处理器以执行根据本发明的编码和 / 或解码的软件。本发明的方法和装置能用于 OFDM 通信系统以及包括 CDMA 系统的其他类型通信系统。

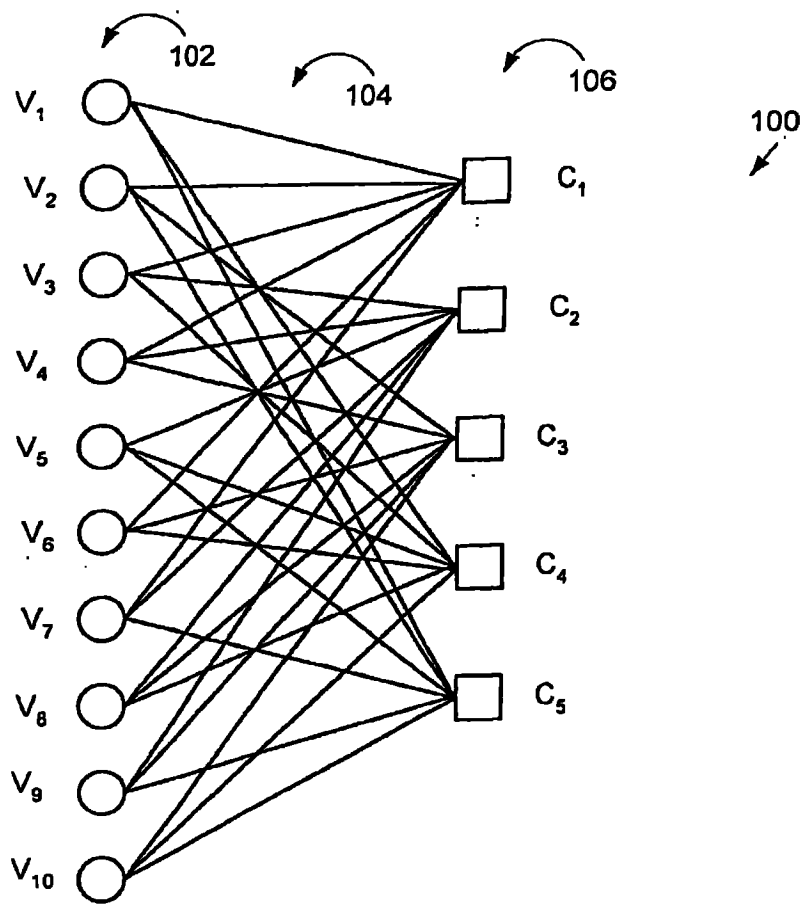


图 1

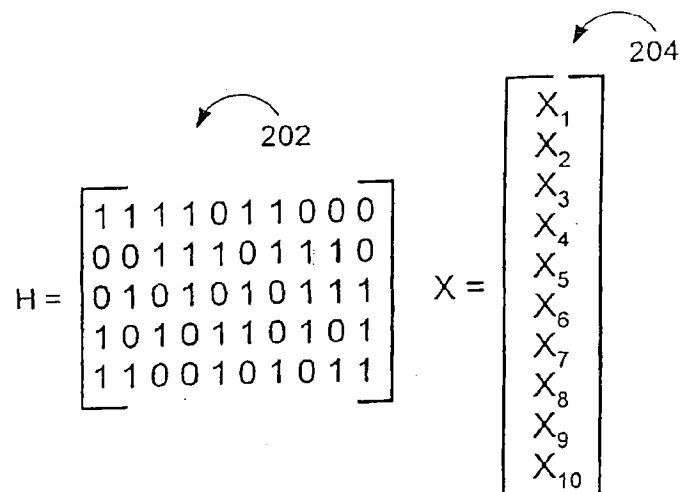


图 2

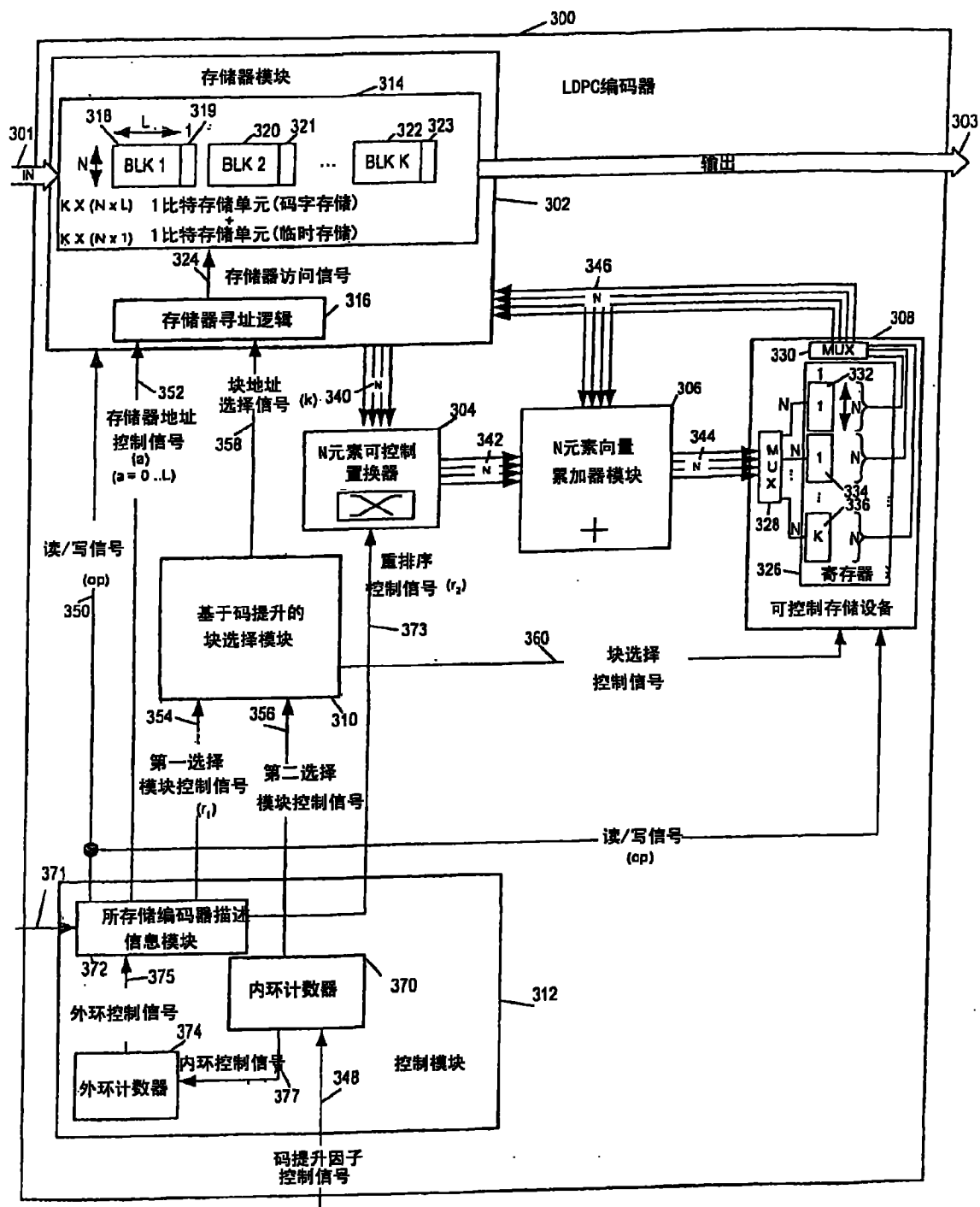


图 3

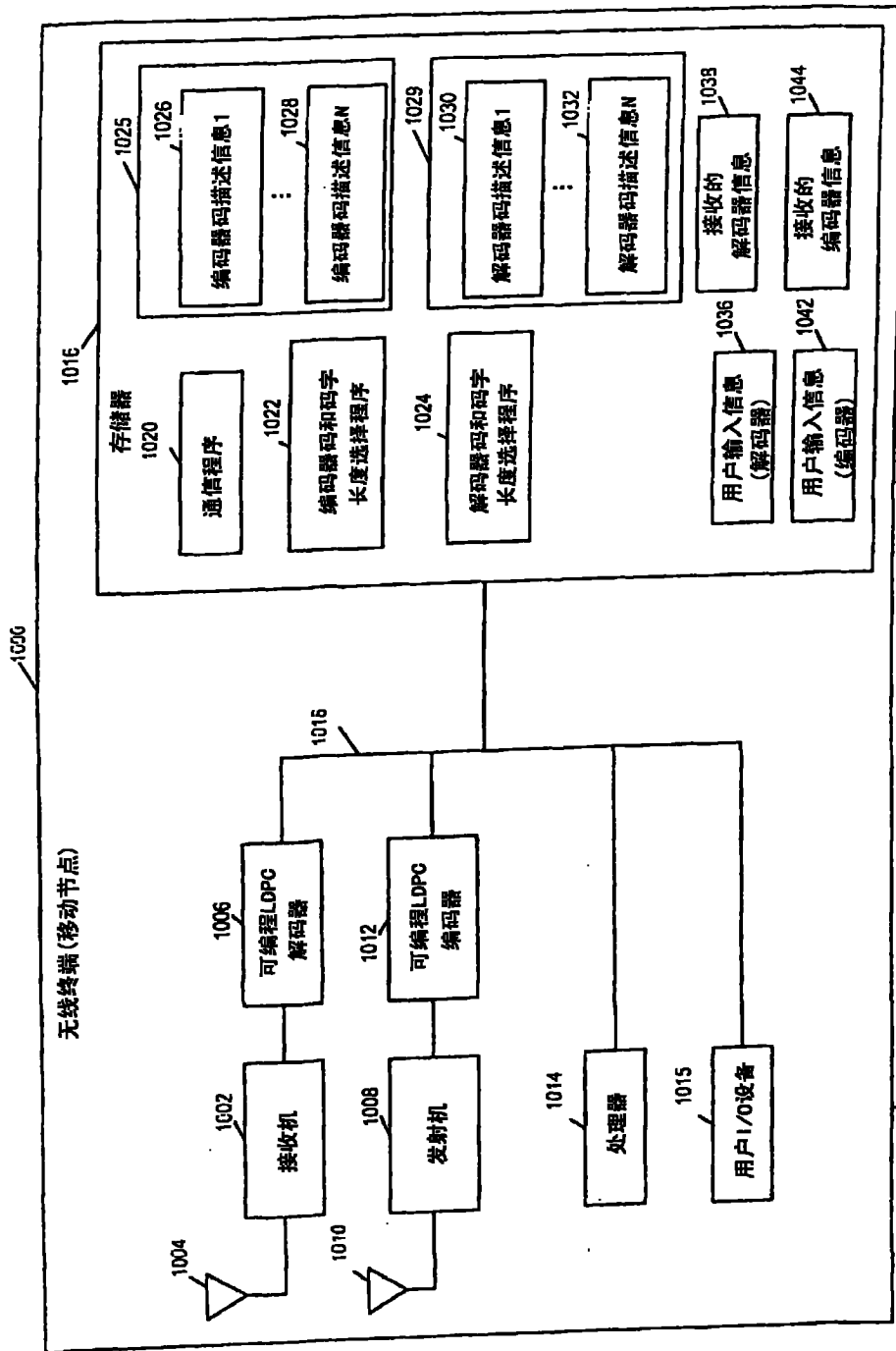


图4

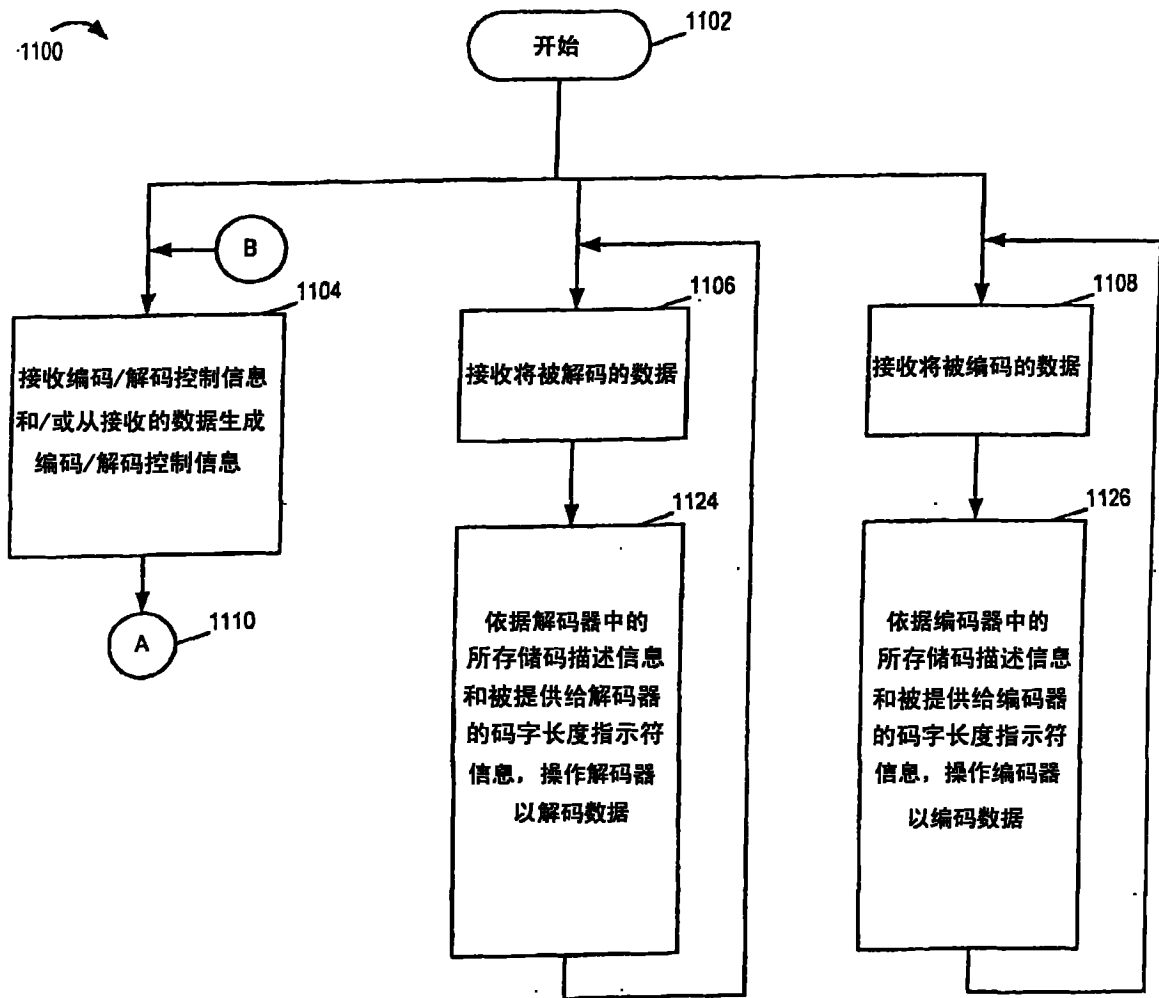


图 5A

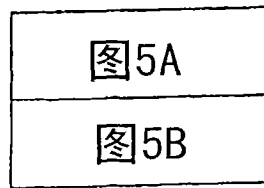


图 5

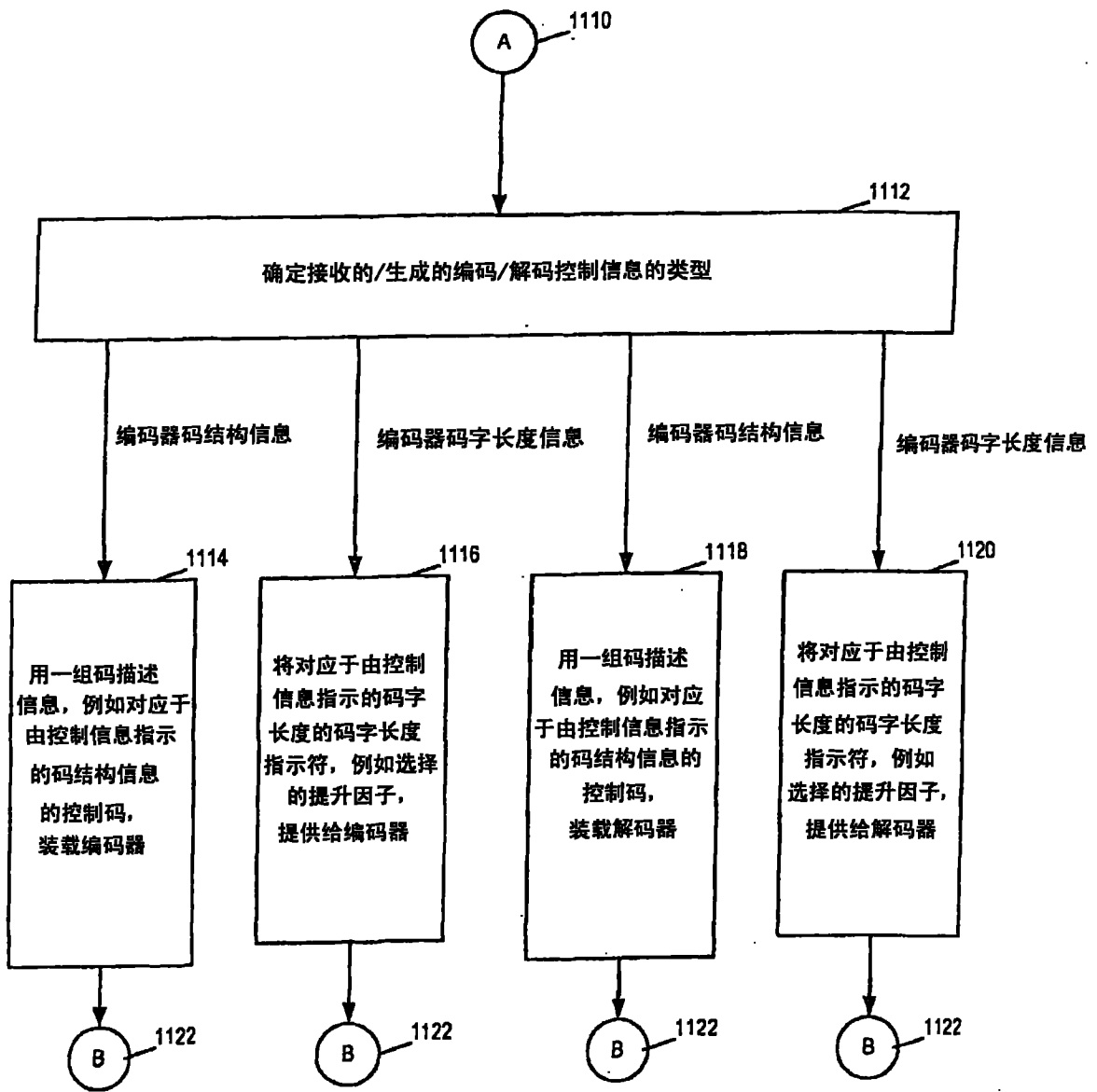


图 5B