(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112470405 A
(43)申请公布日 2021.03.09

(21)申请号 201980045194.8

(22)申请日 2019.07.04

(30)优先权数据
18305888.2 2018.07.05 EP

(85)PCT国际申请进入国家阶段日
2021.01.05

(86)PCT国际申请的申请数据
PCT/EP2019/068040 2019.07.04

(87)PCT国际申请的公布数据
WO2020/008005 EN 2020.01.09

(71)申请人 南布列塔尼大学
地址 法国洛里昂

(72)发明人 E·布蒂永 C·马尔尚 H·哈博

(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 刘瑜

(51)Int.Cl.
H03M 13/11(2006.01)

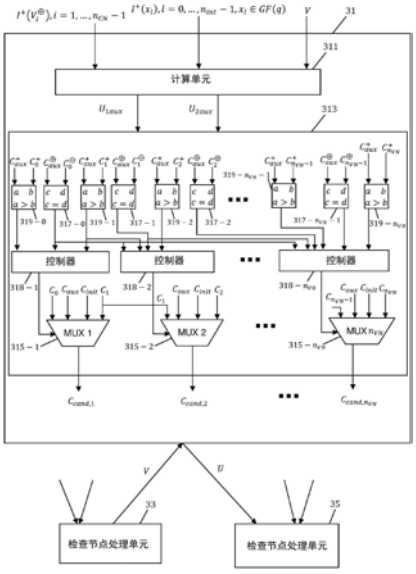
权利要求书3页 说明书25页 附图5页

(54)发明名称

非二进制码的消息传递解码的可变节点处理方法和设备

(57)摘要

本发明的实施例提供了一种用于非二进制纠错码解码器的可变节点处理单元(31),该可变节点处理单元(31)被配置为:接收一个检查节点消息和固有可靠性度量,并且根据从所述一个检查节点消息和固有可靠性度量导出的辅助分量来生成一个可变节点消息,固有可靠性度量是从接收到的信号导出的,辅助分量包括辅助符号和关联于所述辅助符号的辅助可靠性度量,其中可变节点处理单元(31)包括:-排序和冗余消除单元(313),其被配置为迭代地处理辅助分量并通过以下操作来确定可变节点消息的分量:根据辅助可靠性度量的给定顺序来对辅助分量进行迭代地排序,并保留包括最可靠且全部互不相同的辅助符号的预定义数量的辅助分量。



CN 112470405 A

1.一种用于非二进制纠错码解码器的可变节点处理单元(31),所述可变节点处理单元(31)被配置为:接收一个检查节点消息和固有可靠性度量,并且根据从所述一个检查节点消息和所述固有可靠性度量导出的辅助分量来生成一个可变节点消息,所述固有可靠性度量从接收到的信号导出,辅助分量包括辅助符号和关联于所述辅助符号的辅助可靠性度量,其中所述可变节点处理单元(31)包括:

-排序和冗余消除单元(313),其被配置为迭代地处理所述辅助分量并且通过以下操作来确定所述可变节点消息的分量:根据所述辅助可靠性度量的给定顺序来对所述辅助分量进行迭代地排序,并且保留包括最可靠且全部互不相同的辅助符号的预定义数量的辅助分量。

2.根据权利要求1所述的可变节点处理单元(31),其中所述可变节点处理单元(31)包括计算单元(311),所述计算单元(311)被配置为:根据所述检查节点消息和所述固有可靠性度量来确定第一辅助消息,并且根据所述固有可靠性度量来确定第二辅助消息,所述固有可靠性度量与预定义数量的最可靠固有符号以及参考可靠性度量相关联,所述辅助分量是从所述第一辅助消息或所述第二辅助消息中提取的。

3.根据权利要求2所述的可变节点处理单元(31),其中所述检查节点消息包括检查节点分量,每个检查节点分量包括符号和关联于所述符号的可靠性度量,所述计算单元(311)被配置为:

-根据所述检查节点消息中包括的符号来确定所述第一辅助消息中包括的辅助符号,辅助符号从所述检查节点分量中包括的符号中被选择,以及

-通过在预定义的代数结构上应用加法运算来确定与所述第一辅助消息中包括的辅助符号相关联的每个辅助可靠性度量,所述加法运算与所选择的辅助符号关联地被应用于与所述辅助符号相关联的固有可靠性度量以及被包括在检查节点分量中的可靠性度量。

4.根据前述权利要求2或3中的任一项所述的可变节点处理单元(31),进一步被配置为:接收在所述非二进制纠错码解码器中根据偏移值和包括最不可靠符号的所述检查节点分量确定的参考可靠性度量,所述计算单元(311)被配置为:

-根据与最低固有可靠性度量或最高固有可靠性度量相关联的预定义数量的最可靠固有符号,确定在所述第二辅助消息中包括的所述辅助符号,辅助符号等于所述最可靠固有符号之中的固有符号;以及

-通过在预定义的代数结构上应用加法运算来确定与在所述第二辅助消息中包括的辅助符号相关联的所述辅助可靠性度量,所述加法运算被应用于所述参考可靠性度量以及所述固有可靠性度量,所述固有可靠性度量与等于所述辅助符号的所述固有符号相关联。

5.根据权利要求3至4中的任一项所述的可变节点处理单元(31),其中所述预定义的代数结构是在包括实数域、整数域和自然数域的组中选择的。

6.根据权利要求4至5中的任一项所述的可变节点处理单元(31),其中所述偏移值是根据预定义的解码性能标准来在所述非二进制纠错码解码器中确定的。

7.根据前述权利要求4至6中的任一项所述的可变节点处理单元(31),其中所述可变节点处理单元(31)被配置为:在多次迭代期间与至少一个检查节点处理单元(33)交换可变节点消息和检查节点消息,所述偏移值取决于所述迭代的次数在所述非二进制纠错码解码器中被确定,不同的偏移值在所述迭代的每一次时被动态地确定。

8.根据任何前述权利要求所述的可变节点处理单元(31),其中所述排序和冗余消除单元(313)包括:

-多个复用器(315-j);

-多个可靠性度量比较器(319-j),以及

-多个符号比较器(317-j);

每个复用器(315-j)与可靠性度量比较器(319-j)以及符号比较器(317-j)相关联,每个复用器(315-j)被配置为:在每次迭代时取决于由所述多个符号比较器(317-j)和所述多个可靠性度量比较器(319-j)执行的比较来确定当前候选分量,所述排序和冗余消除单元(313)被配置为在处理所述第一辅助消息和所述第二辅助消息中包括的所有所述辅助分量之后,根据包括最可靠且全部互不相同的所述符号的候选分量来生成可变节点分量。

9.根据权利要求8所述的可变节点处理单元(31),其中候选分量包括候选符号和关联于所述候选符号的候选可靠性度量,每个复用器(315-j)被配置为:在每次迭代时接收从所述第一辅助消息或所述第二辅助消息中提取的至少辅助分量以及预定义的初始化分量,所述预定义的初始化分量包括初始化符号和关联于所述初始化符号的初始化可靠性度量,每个复用器(315-j)被配置为在每次迭代时通过在以下各项之中选择分量来确定当前候选分量:所述至少辅助分量、所述预定义的初始化分量、以及与由所述复用器(315-j)在先前迭代时确定的候选分量相对应的分量,以及:

-与每个复用器(315-j)相关联的所述可靠性度量比较器(319-j)被配置为:执行在所述至少辅助分量中包括的所述辅助可靠性度量与由每个复用器(315-j)在先前迭代时确定的所述候选分量中包括的所述候选可靠性度量之间的比较,以及

-与每个复用器(315-j)相关联的所述符号比较器(317-j)被配置为:通过执行在所述至少辅助分量中包括的所述辅助符号与在由所述每个复用器(315-j)在先前迭代时确定的所述候选分量中包括的所述候选符号之间的比较,来检查符号的冗余。

10.根据权利要求8至9中的任一项所述的可变节点处理单元(31),其中在每次迭代时,每个复用器(315-j)被配置为:

-通过将所述当前候选分量设置为等于在先前迭代时确定的所述候选分量来执行保留动作:

*如果确定在所述至少辅助分量中包括的所述辅助符号不冗余,其中没有在先前迭代时确定所述候选分量中的任何一个,并且确定在所述至少辅助分量中包括的所述辅助符号比在由所述复用器315-j在先前迭代时确定的所述候选分量中包括的所述符号不可靠;或者

*如果确定在所述至少辅助分量中包括的所述辅助符号冗余,其中由所述复用器(315-m)中的至少一个在先前迭代时确定所述候选分量中的一个中包括的所述符号中的至少一个,其中m在1和j-1之间变化;

-如果确定在所述辅助分量中包括的所述辅助符号不冗余,其中没有由先前迭代时的所述复用器(315-m)在先前迭代时确定的所述候选分量中包括的所述符号中的任何一个,其中m在1和j-1之间变化,并且确定在所述辅助分量中包括的所述辅助符号比在由所述复用器(315-j)在先前迭代时确定的所述候选分量中包括的所述符号可靠,并且所述辅助符号比由所述复用器(315-(j-1))在先前迭代时确定的所述候选分量中包括的所述符号不可

靠,则通过将所述当前候选分量设置为等于所述辅助分量来执行插入动作;

-如果确定在所述辅助分量中包括的所述辅助符号不冗余,其中没有由所述复用器(315-m)在先前迭代时确定的候选分量中包括的所述符号中的任何一个,其中m=1,…,j-1,并且确定在所述辅助分量中包括的所述辅助符号比由所述复用器(315-j)在先前迭代时确定的所述候选分量中包括的所述符号可靠,并且所述辅助符号比由所述复用器(315-(j-1))在先前迭代时确定的所述候选分量中包括的所述符号可靠,则通过将所述当前候选分量设置为等于由所述复用器(315-(j-1))在先前迭代时确定的所述候选分量来执行移位动作。

11.根据权利要求9所述的可变节点处理单元(31),其中所述多个复用器(315-j)中的每一个被配置为:通过将所述当前候选分量设置为等于所述初始化分量来在第一迭代时执行初始化动作。

12.根据任何前述权利要求所述的可变节点处理单元(31),进一步被配置为:响应于由所述可变节点处理单元(31)递送的固有可靠性度量生成请求来从所述非二进制纠错码解码器中的数据处理单元(23)在线接收所述固有可靠性度量。

13.根据任何前述权利要求所述的可变节点处理单元(31),进一步被配置为:在每次迭代时从存储单元(21)加载至少一部分所述固有可靠性度量。

14.一种用于根据从一个检查节点消息和固有可靠性度量导出的辅助分量来生成一个可变节点消息的方法,所述固有可靠性度量是从接收到的信号中导出的,辅助分量包括辅助符号和关联于所述辅助符号的辅助可靠性度量,其中所述方法包括:

-通过以下操作来确定所述可变节点消息的分量:对所述辅助分量进行迭代地处理,以用于根据所述辅助可靠性度量的给定顺序来对所述辅助分量进行排序,并且保留包括最可靠且全部互不相同的所述辅助符号的预定义数量的辅助分量。

非二进制码的消息传递解码的可变节点处理方法和设备

技术领域

[0001] 本发明总体上涉及数字通信,并且特别地涉及用于对使用非二进制纠错码编码的信号进行解码的方法和设备。

背景技术

[0002] 在各种数据处理系统和设备中实现纠错码,以用于确保保护数据免受在存在噪声和/或干扰的情况下在数据传输或存储期间引入的错误。编码包括将冗余数据添加到原始数据,该冗余数据实现错误的检测和/或纠正。

[0003] 例如在数据通信和/或存储中应用的多种设备和系统中实现纠错码。示例性应用包括例如在无线自组织网络(wireless ad-hoc network)中(例如,用Wi-FiTM802.11标准化)、在无线电通信系统中(例如,用3G、4G/LTE、5G等)、在光学通信系统中、以及在数字视频广播中(例如,用DVB-C2、DVB-S2X和DVB-T2标准化)的语音和多媒体传输。

[0004] 鉴于线性纠错码比非线性纠错码要求更低的实施方式复杂性,因此它们是有优势的。示例性线性纠错码包括卷积码和线性块码,例如Hamming码、Reed-Solomon码、Turbo码、Polar码、以及低密度奇偶校验(LDPC)码。

[0005] 特别是LDPC码是非常高效的。非二进制LDPC码的性能被证明接近香农极限;它们提供接近在传输信道上可以被传送的最大信息量的高传输速率。非二进制LDPC码在提供高频谱效率编码时是非常高效的,并且性能比二进制LDPC码更好。然而,为了达到这样的增益,要求开发用于非二进制码的低复杂性解码算法,特别是针对高频谱效率通信。

[0006] 对通常使用线性块码(并且特别是LDPC码)编码的信号进行的解码可以使用迭代消息传递算法来执行。

[0007] 用于非二进制码(例如,非二进制LDPC码)的示例性解码算法包括:

[0008] -例如在“L.Barnault和D.Declercq,Fast decoding algorithm for LDPC over GF(q),In Proceedings of IEEE Information Theory Workshop,70-73页,2003年四月”中公开的“q阵列总和-乘积”算法;

[0009] -例如在“V.Savin,Min-max decoding for non-binary LDPC codes,In Proceedings of IEEE International Symposium on Information Theory,960-964页,2008年七月”中公开的“最小-最大”算法;

[0010] -例如,在“D.Declercq和M.Fossorier,Decoding algorithms for non-binary LDPC codes over GF,IEEE Transactions on Communications,卷55,第4期,633-643页,2007年四月”中公开的“扩展的最小-最大”(EMS)算法,以及

[0011] -在“J.O.Lacruz,F.García-Herrero,J.Valls和D.Declercq,One Minimum Only Trellis Decoder for Non-Binary Low-Density Parity-Check Codes,in IEEE Transactions on Circuits and Systems I:Regular Papers,卷62,第1期,177-184页,2015年一月”和“E.Li,F.García-Herrero,D.Declercq,K.Gunnam,J.O.Lacruz和J.Valls,“Low latency T-EMS decoder for non-binary LDPC codes”,2013Asilomar Conference

on Signals,Systems and Computers,Pacific Grove,CA,2013,831-835页”中公开的“网格(Trellis)EMS解码器”(T-EMS)。

[0012] EMS算法基于对数尺度计算,以用于通过将乘积运算变换为简单求和运算来降低计算复杂性。与其他现有的迭代解码算法相比,EMS算法给出了在硬件复杂性和通信性能之间的良好的折衷。尽管如此,EMS算法仍然要求大量的计算和存储资源,这些资源在实际设备或系统中可能不可用,例如在实时应用中或在需要高吞吐量和显著减少的延时的系统中使用。

[0013] 消息传递算法基于在检查节点处理单元和关联于所用码的图形表示的可变节点处理单元之间交换表示编码数据的消息。解码过程包括:通过计算可变节点消息进行的可变节点更新、通过计算检查节点消息进行的检查节点更新、以及码字决策。每个检查节点和可变节点消息可以包括一个或多个分量,一个分量包括符号以及与该符号相关联的可靠性度量。在解码过程的开始处,发送到检查节点处理单元的可变节点消息可以包括固有消息,固有消息包括固有可靠性度量和固有符号。固有可靠性度量仅根据信道输出/观察来计算。因此,用词“固有”是指固有可靠性度量仅依赖于(传输)信道。与固有可靠性度量相关联的每个符号都被称为固有符号。

[0014] 在使用EMS算法的解码过程期间涉及的最大计算复杂性源自在检查节点处理单元和可变节点处理单元处执行的计算。

[0015] 关于非二进制纠错码解码架构的大多数现有工作都关注于检查节点而不是可变节点处的处理。文献中很少有著作致力于在可变节点处理单元处执行的处理。

[0016] 可变节点处理单元处的处理包括三个主要步骤:根据接收到的信号生成固有可靠性度量,根据固有可靠性度量和由可变节点处理单元接收的检查节点消息计算可变节点消息,以及计算局部决策。计算可变节点消息的步骤包括中间步骤,该中间步骤包括根据固有可靠性度量和检查节点消息来计算被称为“辅助分量”的分量。辅助分量是被用于生成可变节点消息的分量的中间分量。因此,通过在辅助分量之中提取包括不同且最可靠的符号的分量,来确定可变节点消息的分量。为此,计算可变节点消息的步骤涉及排序子步骤和冗余消除子步骤。排序实现根据其中包括的可靠性度量的给定顺序来对辅助分量进行排序。冗余消除实现移除冗余并且在排序的辅助分量之中保留包括不同且最可靠的符号的分量。可变节点处理的这种架构特别在以下文献中被详细地描述:

[0017] -C.L.Lin,S.W.Tu,C.L.Chen,H.C.Chang,和C.Y.Lee,“An Efficient Decoder Architecture for non-binary LDPC Codes with Extended Min-Sum Algorithm,”IEEE Transactions on Circuits and Systems II:Express Briefs,卷63,第9期,863-867页,2016年九月。

[0018] -E.Boutillon,L.Conde-Canencia,和A.A.Ghouwayel,“Design of a GF(64)-LDPC decoder based on the EMS algorithm,”IEEE Transactions on Circuits and Systems I:Regular Papers,卷60,第10期,2644-2656页,2013年十月。

[0019] 在现有的可变节点架构中,排序和冗余消除子步骤以单独的方式被连续执行,这要求大量的内存和计算能力,这特别是在对在高阶代数结构上构建的纠错码进行解码的情况下可能是负担不起的。现有的可变节点架构是复杂的,并且仍然要求改进以便降低可变节点消息计算的复杂性。因此,存在低复杂性和低延时的可变节点处理技术的需要。

发明内容

[0020] 为了解决这些和其他问题，提供了一种在非二进制纠错码解码器中实现的低复杂性且低延时的可变节点处理单元。可变节点被配置为：接收一个检查节点消息和从接收到的信号导出的固有可靠性度量，并且根据从一个检查节点消息和固有可靠性度量导出的辅助分量来生成一个可变节点消息，辅助分量包括辅助符号和关联于所述辅助符号的辅助可靠性度量。可变节点处理单元包括：

[0021] -排序和冗余消除单元，其被配置为：迭代地处理辅助分量并且通过以下操作来确定可变节点消息的分量：根据辅助可靠性度量的给定顺序来对辅助分量进行迭代地排序，并且保留包括最可靠且全部互不相同的辅助符号的预定义数量的辅助分量。

[0022] 根据一些实施例，可变节点处理单元包括计算单元，该计算单元被配置为：根据检查节点消息和固有可靠性度量来确定第一辅助消息，并且根据固有可靠性度量来确定第二辅助消息，所述固有可靠性度量与预定义数量的最可靠固有符号以及参考可靠性度量相关联，所述辅助分量是从第一辅助消息或第二辅助消息中提取的。

[0023] 根据一些实施例，检查节点消息包括检查节点分量，每个检查节点分量包括符号和关联于所述符号的可靠性度量。在这样的实施例中，计算单元可以被配置为：

[0024] -根据在检查节点消息中包括的符号来确定在第一辅助消息中包括的辅助符号，辅助符号从在检查节点分量中包括的符号中被选择，以及

[0025] -通过在预定义的代数结构上应用加法运算来确定与在第一辅助消息中包括的辅助符号相关联的每个辅助可靠性度量，加法运算与所选择的辅助符号关联地被应用于与所述辅助符号相关联的固有可靠性度量以及被包括在检查节点分量中的可靠性度量。

[0026] 根据一些实施例，可变节点处理单元被进一步配置为：接收在非二进制纠错码解码器中根据偏移值和包括最不可靠符号的检查节点分量确定的参考可靠性度量。在这样的实施例中，计算单元可以被配置为：

[0027] -根据与最低或最高固有可靠性度量相关联的预定义数量的最可靠固有符号，确定在第二辅助消息中包括的辅助符号，辅助符号等于在最可靠固有符号之中的固有符号；以及

[0028] -通过在预定义的代数结构上应用加法运算来确定与第二辅助消息中包括的辅助符号相关联的辅助可靠性度量，该加法运算被应用于所述参考可靠性度量以及所述固有可靠性度量，所述固有可靠性度量与等于所述辅助符号的固有符号相关联。

[0029] 根据一些实施例，预定义的代数结构是在包括实数域、整数域和自然数域的组中选择的。

[0030] 根据一些实施例，偏移值是根据预定义的解码性能标准来在非二进制纠错码解码器中确定的。

[0031] 根据一些实施例，可变节点处理单元被配置为：在多次迭代期间与至少一个检查节点处理单元交换可变节点消息和检查节点消息，偏移值取决于所述迭代次数在非二进制纠错码解码器中被确定，不同的偏移值在所述迭代的每一次中被动态地确定。

[0032] 根据一些实施例，排序和冗余消除单元包括：

[0033] -多个复用器；

[0034] -多个可靠性度量比较器；以及

[0035] -多个符号比较器;

[0036] 每个复用器与可靠性度量比较器以及符号比较器相关联,每个复用器被配置为:在每次迭代时取决于由多个符号比较器和多个可靠性度量比较器执行的比较来确定当前候选分量,排序和冗余消除单元被配置为在处理第一和第二辅助消息中包括的所有辅助分量之后,根据包括最可靠且全部互不相同的符号的候选分量来生成可变节点分量。

[0037] 根据一些实施例,候选分量包括候选符号和关联于所述候选符号的候选可靠性度量。每个复用器可以被配置为:在每次迭代时接收从第一或第二辅助消息中提取的至少辅助分量以及预定义的初始化分量,该预定义的初始化分量包括初始化符号和关联于初始化符号的初始化可靠性度量。在这样的实施例中,每个复用器可以被配置为在每次迭代时通过在以下各项之中选择分量确定当前候选分量:至少辅助分量、预定义的初始化分量、以及与由所述复用器在先前迭代时确定的候选分量相对应的分量;以及:

[0038] -与每个复用器相关联的可靠性度量比较器可以被配置为:执行在所述至少辅助分量中包括的辅助可靠性度量与由每个复用器在先前迭代时确定的候选分量中包括的候选可靠性度量之间的比较,以及

[0039] -与每个复用器相关联的符号比较器可以被配置为:通过执行在至少辅助分量中包括的辅助符号与在由所述每个复用器在先前迭代时确定的候选分量中包括的候选符号之间的比较,来检查符号的冗余。

[0040] 根据一些实施例,在每次迭代时,每个复用器315-j可以被配置为:

[0041] -通过将当前候选分量设置为等于在先前迭代时确定的候选分量来执行保留动作:

[0042] *如果确定至少辅助分量中包括的辅助符号不冗余,其中没有在先前迭代时确定的候选分量中的任何一个,并且确定至少辅助分量中包括的辅助符号比由复用器315-j在先前迭代时确定的候选分量中包括的符号不可靠;或者

[0043] *如果确定至少辅助分量中包括的辅助符号冗余,其中由复用器315-m中的至少一个在先前迭代时确定候选分量中的一个中包括的符号中的至少一个,其中m在1和j-1之间变化;

[0044] -如果确定在辅助分量中包括的辅助符号不冗余,其中没有由在先前迭代时的复用器315-m在先前迭代时确定的辅助分量中包括的符号中的任何一个,其中m在1和j-1之间变化,并且确定在辅助分量中包括的辅助符号比在由复用器315-j在先前迭代时确定的候选分量中包括的符号可靠,并且该辅助符号比由复用器315-(j-1)在先前迭代时确定的候选分量中包括的符号不可靠,则通过将当前候选分量设置为等于辅助分量来执行插入动作;

[0045] -如果确定辅助分量中包括的辅助符号不冗余,其中没有由复用器315-m在先前迭代时确定的候选分量中包括的符号中的任何一个,其中m=1,…,j-1,并且确定在辅助分量中包括的辅助符号比由复用器315-j在先前迭代时确定的候选分量中包括的符号可靠,并且辅助符号比由复用器315-(j-1)在先前迭代时确定的候选分量中包括的符号可靠,则通过将当前候选分量设置为等于由复用器315-(j-1)在先前迭代时确定的候选分量来执行移位动作。

[0046] 根据一些实施例,多个复用器中的每一个被配置为:通过将当前候选分量设置为

等于初始化分量来在第一迭代时执行初始化动作。

[0047] 根据一些实施例,可变节点处理单元被进一步配置为:响应于由可变节点处理单元递送的固有可靠性度量生成请求来从非二进制纠错码解码器中的数据处理单元在线接收固有可靠性度量

[0048] 根据一些实施例,可变节点处理单元被进一步配置为:在每次迭代时从存储单元加载至少一部分固有可靠性度量。

[0049] 还提供了一种用于根据从一个检查节点消息和固有可靠性度量导出的辅助分量来生成一个可变节点消息的方法,所述固有可靠性度量是从接收到的信号中导出的,该辅助分量包括辅助符号和关联于辅助符号的辅助可靠性度量。该方法包括:

[0050] -通过以下操作来确定可变节点消息的分量:对辅助分量进行迭代地处理,以用于根据辅助可靠性度量的给定顺序来以对辅助分量进行排序,并且保留包括最可靠且全部互不相同的辅助符号的预定义数量的辅助分量。

[0051] 有利地,根据本发明的各种实施例的可变节点架构与现有架构相比具有减少的延时并且要求减少的存储和计算资源。

附图说明

[0052] 并入本说明书并构成本说明书的一部分的附图仅出于说明目的被提供,并且附图与上面给出的本发明的一般描述以及下面给出的实施例的具体实施方式一起示出了本发明的各种实施例。

[0053] 图1是根据一些实施例的本发明对通信系统的示例性应用的框图;

[0054] 图2是根据一些实施例的非二进制纠错码解码器的框图,在这些实施例中考虑了EMS解码算法;

[0055] 图3是示出根据一些实施例的可变节点处理单元的结构的框图,在这些实施例中可变节点处理单元的度(degree)为二;

[0056] 图4是示出根据一些实施例的对使用非二进制纠错码编码的信号进行解码的方法的流程图,在这些实施例中使用了EMS解码算法;

[0057] 图5是示出根据本发明的一些实施例的计算可变节点消息的方法的流程图,在这些实施例中根据一个检查节点消息和固有可靠性度量来迭代地确定可变节点消息,以及

[0058] 图6是示出根据本发明的一些实施例的确定包括不同符号的排序的可变节点分量的方法的流程图,其中执行了迭代排序和冗余消除。

具体实施方式

[0059] 本公开的实施例提供了用于对使用非二进制纠错码编码的信号进行解码的具有降低的计算复杂性和降低的延时的设备、方法和计算机程序产品。特别地,它们提供了在迭代消息传递解码器中实现的高效的低延时、低内存和低复杂性的可变节点处理架构,以用于对使用非二进制纠错码编码的信号进行解码。

[0060] 可以在若干类型的应用中使用的若干类型的数字数据传输和存储设备以及系统中实现根据各种实施例的方法、设备和计算机程序产品。示例性设备和系统包括但不限于计算机、磁盘、膝上型计算机、电话、智能电话、录音机、基站、无人机、卫星等。示例性应用包

括磁性和光学记录、数字电视和视频广播、数字通信等。

[0061] 仅出于说明目的，将参考数字通信系统对本公开的一些实施例进行以下描述。然而，技术人员将容易理解的是，本公开的各种实施例可以被集成在用于其他应用的其他类型的系统中，例如定位系统存储器和存储系统以及航天器系统。

[0062] 参考图1，示出了本公开在数字通信系统100中的示例性应用。通信系统100可以是例如：

[0063] -有线的；

[0064] -无线的(例如，无线电或可见光通信系统)；

[0065] -光学的(例如，基于光纤的、基于激光的)；

[0066] -声学的(例如，水下声学通信系统)；

[0067] -分子的(例如，在地下结构中(例如，隧道和管线)或在水下环境中使用)。

[0068] 通信系统100可以包括至少发送器设备10和接收器设备12。发送器设备10(在下文中也被称为“发送器”)可以被配置为将数据信息经由传输信道11传送到接收器设备12(在下文中也被称为“接收器”)。

[0069] 在本公开在诸如计算机联网系统之类的有线通信系统的应用中，发送器10和/或接收器12可以是被配置为在有线网络中操作的任何设备。在这样的应用中的示例性设备包括连接到小区域或大区域有线网络的计算机、路由器或交换机。在这种情况下，传输信道11可以是用于确保不同连接设备之间的数据传输的任何类型的物理电缆。

[0070] 在本公开在诸如自组织无线网络、无线传感器网络和无线电通信系统之类的无线通信系统的应用中，发送器10和接收器12可以是被配置为在无线环境中操作的任何类型的固定或移动无线设备。在这样的应用中的示例性设备包括膝上型计算机、移动电话和基站。在这种情况下，传输信道11可以是任何无线传播介质。此外，传输信道11可以容纳若干个发送器10和/或若干个接收器12。在这样的实施例中，可以结合纠错码使用多种接入技术和/或网络编码技术。示例性多址技术包括时分多址(TDMA)、频分多址(FDMA)、码分多址(CDMA)和空分多址(SDMA)。

[0071] 在本公开在诸如基于光纤的系统之类的光学通信系统的应用中，发送器10和接收器12可以是被配置为分别发送和接收在光学链路上传播的数据信息的任何光学收发器设备。传输信道11可以是被设计为在短距离或长距离上承载数据的任何光纤链路。在短距离上使用光纤链路的示例性应用包括诸如数据中心互连之类的高容量网络。在长距离上使用光纤链路的示例性应用包括陆地的和越洋的传输。在这样的实施例中，由发送器10传达的信息符号可以由根据光纤的不同偏振状态进行偏振的光学信号承载。光学信号根据一种或多种传播模式来沿着基于光纤的传输信道11进行传播，直到到达接收器12为止。示例性光学通信系统包括偏振分割复用(PDM)和模式分割复用(MDM)系统。

[0072] 针对任何类型的有线、无线或光学通信系统，传输信道11可以是任何噪声信道。噪声可能是由于系统分量的热噪声或者由天线截获的干扰辐射引起的。其他示例性噪声源包括开关、手动中断、电火花和闪电。在一些实施例中，总噪声可以通过加性高斯白噪声(AWGN)来建模。

[0073] 此外，根据应用于数字大容量存储的一些实施例，可以例如通过擦除信道、二进制对称信道或高斯信道来对传输信道11进行建模。在这样的实施例中，传输信道11可以是可

以被发送到的(写入)和从其接收(读取)的任何类型的存储设备。

[0074] 另外,发送器10和接收器12可以被配备有单个或多个天线。特别地,在存在多个发送和/或接收天线的情况下,可以结合纠错编码和解码使用空时编码和解码技术。

[0075] 此外,可以在一个或多个频带上发送编码数据。当在多个频带上传送编码数据时,调制器105可以使用诸如OFDM(正交频分复用的缩写)和FBMC(滤波器组多载波的缩写)之类的多载波调制格式。

[0076] 根据本公开的一些实施例,发送器10可以包括纠错码(ECC)编码器103,其被配置为使用非二进制纠错码将表示为u的数字输入数据块101编码成表示为c的码字向量。接收器12可以被配置为通过传输信道11接收码字向量或编码数据的噪声副本p。接收器12可以包括纠错码解码器123,其被配置为递送数字输出数据块125作为原始数字输入数据块101的估计$\hat{u}$。

[0077] 数字输入数据101可以在被ECC编码器103编码之前被预先压缩。适用于增加信息吞吐量的任何源编码方案(图1中未示出)可以用于执行压缩。由ECC编码器103编码的数据可以被调制器105进一步调制。调制器105可以被配置为将编码数据映射到模拟信号s上并且通过传输信道11发送编码数据。

[0078] 接收器12可以包括被配置为执行反向功能的相应处理块。它可以包括解调器121,该解调器121被配置为在由ECC解码器123进行ECC解码之前,通过对来自传输信道的接收到的信号p执行解调来生成信号y。解调器121可以被配置为将接收到的信号或信道输出移回到基带中并且执行低通滤波、采样和量化。由ECC解码器123解码的数据可以使用任何源解码器(图1中未示出)来进一步解压缩。ECC解码器123可以被配置为实现根据本公开的各种实施例的涉及检查节点处理单元和可变节点处理单元的迭代解码器(被称为“迭代解码算法”)。

[0079] 仅出于说明目的,将参考线性块非二进制纠错码来对本公开的一些实施例进行以下描述。然而,技术人员将容易理解的是,本公开的各种实施例适用于包括非二进制卷积码的任何线性非二进制纠错码。

[0080] 因此,ECC编码器103可以实现由$\mathcal{C}(n,K)$指定的线性块非二进制纠错码;n和K分别是指码字向量的长度和编码数据块的长度。ECC编码器103相应地将长度为K的数据块u编码成码字向量c,$c=(c_1,\cdots,c_n)$是长度为n的向量,该向量包括纠错码$\mathcal{C}(n,K)$的构造的代数结构的n个元素,也被称为“符号”。

[0081] 线性码$\mathcal{C}(n,K)$可以使用G表示的生成器矩阵和H表示的奇偶校验矩阵来以矩阵形式表示。使用向量的行表示法,生成器矩阵G的维度为K x n,而奇偶校验矩阵的维度为(n-K)×n。这两个矩阵通过关系$G.H^t=0$链接起来。另外,两个矩阵的条目都属于代数结构,在该代数结构上构造了纠错码。使用矩阵表示,任何码字向量c都满足等式$c.H^t=0$。此等式还被称为“奇偶校验等式”。它定义了n-K个奇偶校验约束,旨在被任何码字向量满足。

[0082] 与矩阵表示相关联,线性码$\mathcal{C}(n,K)$可以使用被称为“Tanner图”的二部图$\mathcal{H}$来表示。此图包括n个可变节点和n-K个检查节点。

[0083] 每个可变节点$vn\in\{1,2,\cdots,n\}$对应于奇偶校验矩阵的一列。每个检查节点$cn\in\{1,2,\cdots,n-K\}$对应于奇偶校验矩阵的一行,即,对应于奇偶校验等式。如果奇偶校验矩阵的

条目$H_{vn,cn}$等于码$\mathcal{C}(n,K)$的构造的代数结构的元素，则将可变节点vn被连接到检查节点cn。

[0084] $\mathcal{H}_v(vn)$表示被连接到可变节点vn的检查节点的集合。类似地，$\mathcal{H}_c(cn)$表示被连接到检查节点cn的可变节点的集合。

[0085] 可变节点vn的度d_{VN}(或检查节点cn的d_{CN})对应于集合$\mathcal{H}_v(vn)$的基数(或对应于集合$\mathcal{H}_c(cn)$的基数)。

[0086] 根据一些实施例，非二进制纠错码$\mathcal{C}(n,K)$的构造的代数结构可以是任何非零交换除环(commutative division ring)，例如域。示例性域包括有限域(也被称为“Galois域”)。

[0087] 仅出于说明目的，将参考有限域对一些实施例进行以下描述。然而，技术人员将容易理解的是，本公开可以应用于诸如非零交换除环之类的任何类似除环的代数结构以及应用于诸如有限除法拟环(finite division near-ring)的任何拟环。可以在文章“Non-binary LDPC codes over finite division near rings，第23届International Conference on Telecommunications(ICT)，1-7页，Thessaloniki，2016”中找到有关在有限除法拟环上的非二进制纠错码的设计的见解。

[0088] 针对在Galois域(通常由GF(q)表示，其中q>2指定码的基数)上构造的非二进制线性码，这些符号取GF(q)中的值。码字向量c因此是n个符号的向量，每个符号都属于GF(q)。

[0089] 仅出于说明目的，将参考使用非二进制LDPC码对数据进行编码的ECC编码器103对一些实施例进行以下描述。然而，技术人员将容易理解的是，本公开的各种实施例还适用于其他非二进制码，例如非二进制极性码、非二进制卷积码、非二进制turbo码、以及一般地非二进制线性块纠错码。非二进制纠错码可以有利地用于高频谱效率编码。

[0090] 出于示例性的目的，在一个实施例中，ECC解码器123实现非二进制LDPC码解码器，以用于使用非二进制LDPC码对由ECC编码器103编码的数据进行解码。

[0091] 参考EMS算法对本公开进行描述。然而，技术人员将容易理解的是，各种实施例适用于任何迭代的非二进制LDPC码解码器，例如最小-最大和T-EMS。

[0092] 参考图2，示出了根据使用EMS算法的一些实施例的迭代非二进制LDPC解码器123的结构。

[0093] 迭代解码器123可以被配置为根据由向量$y=(y_1,\cdots,y_n)$表示的接收到的噪声序列来确定发送器10对被传送的码字c的估计$\hat{u}$。码字$c=(c_1,\cdots,c_n)$可能已经在发送器处使用由Galois域GF(q)(其中q>2)上构造的$\mathcal{C}(n,K)$指定的非二进制LDPC码进行了编码。它可以在若干次迭代时处理信号，以尝试在每次迭代时减少剩余的解码错误。

[0094] 迭代解码器123可以被配置为基于在发送器10处使用的码$\mathcal{C}(n,K)$的Tanner图表示来确定估计$\hat{u}$。Tanner图中的每个可变节点映射到可变节点处理单元。Tanner图中的每个检查节点都映射到检查节点处理单元。

[0095] 因此，迭代解码器123可以包括n个可变节点处理单元27(也由27-1至27-n或者27-vn(其中vn=1,…,n)表示)和n-K个检查节点处理单元25(也由25-1到25-(n-K)或者25-cn

(其中cn＝1,…,n-K)表示)。

[0096] 可变节点处理单元27-vn(其中vn＝1,…,n)和检查节点处理单元25-cn(其中cn＝1,…,n-K)可以被配置为迭代地交换消息以根据噪声序列y估计最可靠的码字$\hat{u}$。

[0097] 由可变节点处理单元27-vn生成的消息被称为“可变节点消息”。类似地,由检查节点处理单元25-cn生成的消息被称为“检查节点消息”。

[0098] 可变节点处理单元27-vn和检查节点处理单元25-cn之间的交换消息可以承载表示符号的数据。

[0099] 根据一些实施例,可变节点消息和检查节点消息可以承载符号的值和测量其可靠性的度量(在下文中还被称为“可靠性度量”)。可靠性度量的值与符号的可靠性有关。在这样的实施例中,每个可变节点消息和检查节点消息可以包括q个分量,分量包括:

[0100] -GF(q)中符号的值,以及

[0101] -与符号相关联的可靠性度量。

[0102] 在一些实施例中,符号的可靠性度量可以对应于符号的所估计的概率密度函数,所估计的概率密度函数表示符号正确的概率。特别地,可靠性度量可以在对数域中由对数似然比(LLR)值表示。

[0103] 取决于可靠性度量的表达式,最可靠的符号可以与可靠性度量的最小值或最大值相关联。

[0104] LLR度量的一种定义涉及由β_i表示的固定符号,并且对应于满足以下等式的最可靠的符号:

[0105] $\beta_i＝\mathrm{argmax}_{t=0,\cdots,q-1}p(\alpha_t|y_i)$ (1)

[0106] 在等式(1)中,α_t指示Galois域上的符号。

[0107] 因此,利用在Galois域中的符号是等概率的假设,对于第i个符号c_i,针对等于α_t的此符号的LLR值被记为$LLR_t(c_i)$,并且可以被表示为:

[0108] $$LLR_t(c_i) = -\log\left(\frac{p(c_i=\alpha_t|y_i)}{p(c_i=\beta_i|y_i)}\right) \qquad (2)$$

[0109] 使用LLR度量的这种定义,最可靠的符号是根据等式(2)与最小LLR值相关联的那些符号。

[0110] 仅出于说明目的,将参考在对数域中由对数似然比(LLR)值表示的可靠性度量来对一些实施例进行以下描述。然而,技术人员将容易理解的是,其他类型的解码和可靠性度量可以用于测量符号的可靠性。例如,可靠性度量可以是符号的概率密度函数的二次距离或任何单调函数。

[0111] 由各种可变节点处理单元27-vn和检查节点处理单元25-cn执行的处理可以根据若干种调度机制来实现,包括但不限于以下内容所描述的三个示例。

[0112] 根据第一实施方式,所有可变节点处理单元27-vn(其中vn＝1,…,n)可以被配置为在第一轮中操作,并且然后,所有的检查节点处理单元25-cn(其中cn＝1,…,n-K)可以被配置为更新要在其对应的集合$\mathcal{H}_c(cn)$中递送给可变节点处理单元的检查节点消息。这种特定调度被称为“泛洪调度(flooding scheduling)”。特别地,检查节点处理单元25-cn可以被配置为串行或并行地操作,其中从2至(n-K)个检查节点处理单元25-cn可以同时操作。

[0113] 根据基于“水平调度”的第二实施方式,检查节点处理单元25-cn(其中cn＝1,…,

n-K)可以被配置为串行地操作,以更新与它们连接的所有可变节点处理单元27-vn。特别地,一组检查节点处理单元25-cn可以被配置为并行地操作,以更新所有连接的可变节点处理单元27-vn,前提是不存在冲突的可变节点处理单元27-vn(例如,当两个检查节点处理单元25-cn被连接到相同的可变节点处理单元27-vn时)。

[0114] 根据基于“垂直调度”的第三实施方式,可变节点处理单元27-vn可以被配置为串行地操作,以更新与它们连接的所有检查节点处理单元25-cn。

[0115] 迭代解码器123可以包括数据处理单元23,该数据处理单元23被配置为根据接收到的信号确定固有符号和固有可靠性度量,表示为$x\in GF(q)$并且属于Galois域(通常属于非二进制纠错码的代数结构)的固有符号与表示为$I^{+}(x)$的固有可靠性度量相关联。更具体地,数据处理单元23可以包括:

[0116] -符号生成器231,其被配置为生成属于代数结构的固有符号x,在该代数结构上构造了非二进制纠错码,以及

[0117] -固有可靠性度量生成器233,其被配置为从接收到的信号生成与每个固有符号x相关联的固有可靠性度量$I^{+}(x)$。使用等式(2)的LLR表示法,可以根据以下等式以对数尺度确定与符号x相关联的固有可靠性度量:

[0118] $$I^{+}(x)=-\log\left(\frac{p(x=\alpha_t|y_i)}{p(x=\beta_i|y_i)}\right) \qquad (3)$$

[0119] 在等式(3)中,β_i是指先前在等式(1)中定义的最可靠的符号。

[0120] 根据一些实施例,固有可靠性度量生成器单元233可以被配置为从存储单元21加载接收到的信号y,存储单元21被配置为存储接收到的信号y。

[0121] 在图2所示的一些实施例中,存储单元21可以被包括在数据处理单元23内。

[0122] 在其他实施例中(图2中未示出),存储单元21可以被包括在数据处理单元23外部的迭代解码器123中。

[0123] 根据一些实施例,固有可靠性度量生成器单元233可以被配置为:在迭代解码过程开始之前,离线确定至少一部分固有可靠性度量。在这样的实施例中,固有可靠性度量生成器单元233可以被进一步配置为:将计算出固有可靠性度量的至少一部分存储在存储单元21中。这样的实施例可以被有利地实现用于对在小阶代数结构上构建的码进行解码。

[0124] 在一些实施例中,固有可靠性度量生成器单元233可以被配置为在迭代解码过程期间在线确定固有可靠性度量。可以有利地实现这样的实施例以用于对高阶非二进制纠错码进行解码,以实现内存节省。

[0125] 在一些实施例中,固有可靠性度量生成器单元233可以被配置为离线确定固有可靠性度量中的至少一些,并且在线确定剩余的固有可靠性度量。

[0126] 根据一些实施例(图2中示出),迭代解码器123可以包括单个数据处理单元23,其被配置为确定要被递送到所有可变节点处理单元27-vn(其中vn=1,…,n)的固有可靠性度量。

[0127] 在其他实施例中(这里未示出),迭代解码器123可以包括多个数据处理单元23-vn(其中vn=1,…,n),每个数据处理单元23-vn被包括在可变节点处理单元23-vn(其中vn=1,…,n)中。

[0128] 可变节点处理单元27-vn可以被配置为接收从接收到的信号y导出的固有可靠性

度量，并且从对应于集合$\mathcal{H}_v(vn)$的检查节点处理单元25-cn接收检查节点消息。可变节点处理单元27-vn可以被进一步配置为：处理接收到的检查节点消息和固有可靠性度量，计算局部决策，以及将可变节点消息递送到与集合$\mathcal{H}_v(vn)$中的检查节点相对应的至少一个检查节点处理单元25-cn。

[0129] 类似地，检查节点处理单元25-cn可以被配置为处理由对应于集合$\mathcal{H}_c(cn)$的可变节点处理单元27-vn发送的可变节点消息。检查节点处理单元25-cn可以被进一步配置为将检查节点消息递送到与集合$\mathcal{H}_c(cn)$中的可变节点相对应的至少一个可变节点处理单元27-vn。

[0130] 消息的交换可以由可变节点处理单元27-vn开始。如果处理后的信号满足奇偶校验等式，或者如果在不满足所有奇偶校验约束的情况下达到最大迭代次数，则消息的交换可以终止。

[0131] 因此，迭代解码器123可以包括解码决策单元29，其被配置为在解码过程的每次迭代时接收由可变节点处理单元27-vn计算的局部决策，并且被配置为：

[0132] -如果处理后的信号满足奇偶校验等式，则将处理后的信号作为原始码字向量的估计来递送；或者

[0133] -如果在不满足所有奇偶校验约束的情况下达到最大迭代次数，则声明解码失败，但是输出在最后一次迭代时估计的码字向量。

[0134] 本公开的各种实施例提供了用于在迭代的非二进制纠错码解码器中实现的度为二($d_{VN}=2$)的可变节点处理的高效的低复杂性和低延时技术，该可变节点处理单元被配置为：

[0135] -从集合$\mathcal{H}_v(vn)$中的检查节点处理单元接收从接收到的信号y和一个检查节点消息导出的固有可靠性度量，以及

[0136] -根据检查节点消息和固有可靠性度量生成一个可变节点消息。

[0137] 图3是示出度$d_{VN}=2$的可变节点处理单元31的结构的框图，该可变节点处理单元31被配置为：从检查节点处理单元33接收表示为V的一个检查节点消息以及固有可靠性度量，以及确定表示为U的可变节点消息，并且将可变节点消息U递送给不同于检查节点处理单元33的检查节点处理单元35。

[0138] 在以下内容中，在可变节点消息中包括的分量被称为“可变节点分量”，而在检查节点消息中包括的分量被称为“检查节点分量”。

[0139] 根据使用EMS算法的一些实施例，可变节点消息和/或检查节点消息可以根据其可靠性来排序，即，根据其中包括的可靠性度量的给定顺序(例如，递增或递减)来排序。此外，可变节点消息可以被截断以便仅保留预定数量n_{VN}的最可靠的可变节点分量，其中n_{VN}严格小于q($n_{VN}<q$)。类似地，可以将检查节点消息截断，以便仅保留预定义数量n_{CN}的最可靠的检查节点分量，其中n_{CN}严格小于q($n_{CN}<q$)。此外，可以由可变节点处理单元31或者由检查节点处理单元33和/或检查节点处理单元35执行排序和截断操作。

[0140] 在实施例中，预定义数量的可变节点分量n_{VN}和预定义数量的检查节点分量n_{CN}可以是不同的，即，$n_{VN}\neq n_{CN}$。

[0141] 在另一实施例中，预定义数量的可变节点分量n_{VN}和预定义数量的检查节点分量n_{CN}可以是相同的，即，$n_{VN}=n_{CN}$。

[0142] 出于清楚和简化的目的，将参考排序和截断的检查节点消息V和可变节点消息U对本公开的实施例进行以下描述，检查节点分量和可变节点分量的数量是相等或不同的。但是技术人员将容易理解的是，本公开适用于包括未排序的分量的可变节点消息和检查节点消息。

[0143] 因此，检查节点消息V可以包括由$V_i=(V_i^{\oplus},V_i^{+})$表示的$n_{CN}$个检查节点分量(其中$i=0,\cdots,n_{CN}-1$)，检查节点分量$V_i=(V_i^{\oplus},V_i^{+})$包括属于Galois域GF(q)的符号$V_i^{\oplus}$和关联于该符号$V_i^{\oplus}$的可靠性度量$V_i^{+}$。可以根据与可靠性量度$V_i^{+}$的给定顺序(递增或递减)相对应的减少的可靠性来对检查节点分量进行排序，使得$V_i^{+}\leq V_{i+1}^{+}$(其中$i=0,\cdots,n_{CN}-2$)并且包括最可靠符号的分量是$V_0=\left(V_0^{\oplus},V_0^{+}\right)$。

[0144] 可变节点消息U可以包括由$U_t=\left(U_t^{\oplus},U_t^{+}\right)$表示的$n_{VN}$个可变节点分量(对$t=0,\cdots,n_{VN}-1$)，可变节点分量$U_t=\left(U_t^{\oplus},U_t^{+}\right)$包括属于Galois域GF(q)的符号$U_t^{\oplus}$和关联于符号$U_t^{\oplus}$的可靠性度量$U_t^{+}$。可以根据与可靠性度量$U_t^{+}$的给定顺序(递增或递减)相对应的减少的可靠性来对可变节点分量进行排序，使得$U_t^{+}\leq U_{t+1}^{+}$(其中$t=0,\cdots,n_{VN}-2$)并且包括最可靠符号的分量为$U_0=\left(U_0^{\oplus},U_0^{+}\right)$。

[0145] 此外，在以下描述中将使用以下符号：

[0146] -n_{int}表示预定义数量的最可靠的固有符号，它们属于非二进制纠错码的构造的代数结构；

[0147] -$x_l\in GF(q)$(其中$l=0,\cdots,n_{int}-1$)是指n_{int}个最可靠的固有符号，它们属于非二进制纠错码的构造的代数结构，使得最可靠的固有符号是x_0；

[0148] -$I^{+}(x_l)$是指与属于非二进制纠错码的代数结构的固有符号x_l关联的固有可靠性度量，固有可靠性度量$I^{+}(x_l)$是从接收到的序列/信号y导出的；

[0149] -U_{1aux}是指第一辅助消息，其包括表示为$U_{1aux,i}=\left(U_{1aux,i}^{\oplus},U_{1aux,i}^{+}\right)$的$n_{CN}$个辅助分量，其中$i=0,\cdots,n_{CN}-1$，辅助分量$U_{1aux,i}$包括辅助符号$U_{1aux,i}^{\oplus}$和关联于符号$U_{1aux,i}^{\oplus}$的辅助可靠性度量$U_{1aux,i}^{+}$。

[0150] -U_{2aux}是指第二辅助消息，其包括表示为$U_{2aux,l}=\left(U_{2aux,l}^{\oplus},U_{2aux,l}^{+}\right)$的$n_{int}$个辅助分量，其中$l=0,\cdots,n_{int}-1$，辅助分量$U_{2aux,l}$包括辅助符号$U_{2aux,l}^{\oplus}$和关联于符号$U_{2aux,l}^{\oplus}$的辅助可靠性度量$U_{2aux,l}^{+}$；

[0151] -$U_{aux,r}=\left(U_{aux,r}^{\oplus},U_{aux,r}^{+}\right)$(其中r取介于0和$n_{int}+n_{CN}-1$之间的值)是指从第一辅

助消息或第二辅助消息提取的辅助分量。辅助分量可以包括辅助符号$U_{aux,r}^{\oplus}$和关联于符号$U_{aux,r}^{\oplus}$的辅助可靠性度量$U_{aux,r}^{+}$。

[0152] -J是指由迭代解码器123确定的偏移值，该偏移值J可以是大于或等于零的整数(J≥0)；

[0153] -C_{ref}^{+}是指根据以下等式的参考可靠性度量(也被称为“默认可靠性度量”)，其根据可靠性度量$V_{n_{CN}-1}^{+}$以及偏移值J来确定，可靠性度量$V_{n_{CN}-1}^{+}$被包括在包含最不可靠符号的检查节点分量$V_{n_{CN}-1}$(也被称为“参考检查节点分量”或“默认检查节点分量”)中：

[0154] $$C_{ref}^{+}=J+V_{n_{CN}-1}^{+} \qquad (4)$$

[0155] 参考可靠性度量是指其计算，该计算涉及与包括最不可靠性符号的参考检查节点分量相关联的可靠性度量；

[0156] -$C_0=\left(C_0^{\oplus},C_0^{+}\right)$是指包括预定义候选符号$C_0^{\oplus}$和预定义候选可靠性度量$C_0^{+}$的预定义候选分量。在一些实施例中，预定义候选符号$C_0^{\oplus}$可以选自与非二进制纠错码的构造的代数结构不同的代数结构。在一些实施例中，预定义候选可靠性度量C_0^{+}可以等于预定义的最小值；

[0157] -$C_j=\left(C_j^{\oplus},C_j^{+}\right)$(其中j取介于1到$n_{VN}$之间的值)是指包括候选符号$C_j^{\oplus}$和关联于候选符号$C_j^{\oplus}$的候选可靠性度量$C_j^{+}$的候选分量，候选分量对应于在迭代排序和冗余消除操作处确定的分量；

[0158] -$C_{init}=\left(C_{init}^{\oplus},C_{init}^{+}\right)$是指预定义的“初始化分量”，其包括初始化符号$C_{int}^{\oplus}$和关联于初始化符号$C_{init}^{\oplus}$的初始化可靠性度量$C_{init}^{+}$；

[0159] -$C_{cand,j}=\left(C_{cand,j}^{\oplus},C_{cand,j}^{+}\right)$(其中j取介于1到$n_{VN}$之间的值)是指在迭代排序和冗余消除过程的当前迭代时确定的当前候选分量，该当前候选分量包括当前候选符号$C_{cand,j}^{\oplus}$和关联于当前候选符号$C_{cand,j}^{\oplus}$的当前候选可靠性度量$C_{cand,j}^{+}$。

[0160] 根据一些实施例，可变节点处理单元31可以被配置为接收与n_{int}个最可靠的固有符号相关联的预定义数量n_{int}的固有可靠性度量$I^{+}(x_l)$，l＝0,…,n_{int}-1，以及与检查节点消息V中包括的符号$V_i^{\oplus}$相关联的n_{CN}个固有可靠性度量$I^{+}\left(V_i^{\oplus}\right)$，i＝0,…,$n_{CN}$-1。可变节点处理单元31可以被配置为接收来自数据处理单元23的$n_{int}+n_{CN}$个固有可靠性度量。

[0161] 考虑到接收到的检查节点消息V和$n_{int}+n_{CN}$个固有可靠性度量(从存储单元21加载或者从固有可靠性度量生成器单元233在线接收)，可变节点处理单元31可以被配置为根据$n_{CN}+n_{int}$个辅助分量$U_{aux,r}=\left(U_{aux,r}^{\oplus},U_{aux,r}^{+}\right)$(其中r＝0,…,$n_{int}+n_{CN}$-1)确定$n_{VN}$个可变节

点分量$U_t = \left(U_t^{\oplus}, U_t^{+}\right)$，t＝0，…，$n_{VN}$-1，辅助分量是从检查节点消息V和$n_{int}+n_{CN}$个固有可靠性度量$I^{+}(x_l)$（l＝0，…，$n_{int}$-1）和$I^{+}\left(V_i^{\oplus}\right)$（i＝0，…，$n_{CN}$-1）导出的。更具体地，可变节点处理单元31可以包括：

[0162] -计算单元311，其被配置为根据检查节点消息V和关联于检查节点消息V中包括的符号的固有可靠性度量$I^{+}\left(V_i^{\oplus}\right)$，i＝0，…，$n_{CN}$-1来计算第一辅助消息$U_{1aux}$，并且根据与预定义数量$n_{int}$的$n_{int}$个最可靠的固有符号$x_l \in GF(q)$相关联的固有可靠性度量$I^{+}(x_l)$，l＝0，…，$n_{int}$-1和参考可靠性度量$C_{ref}^{+}$来计算第二辅助消息$U_{2aux}$。第一辅助消息$U_{1aux}$可以包括$n_{CN}$个辅助分量$U_{1aux,i} = \left(U_{1aux,i}^{\oplus}, U_{1aux,i}^{+}\right)$（其中i＝0，…，$n_{CN}$-1），并且第二辅助消息$U_{2aux}$可以包括$n_{int}$个辅助分量$U_{2aux,l} = \left(U_{2aux,l}^{\oplus}, U_{2aux,l}^{+}\right)$（其中l＝0，…，$n_{int}$-1）；

[0163] -排序和冗余消除单元313，其被配置为迭代地处理从第一或第二辅助消息中提取的至少辅助分量$U_{aux,r}$，冗余消除和排序单元313被配置为通过迭代地对辅助分量$U_{1aux,i}$和$U_{2aux,l}$进行排序并且保留预定义数量n_{VN}的辅助分量来确定可变节点消息的分量，这些辅助分量包括最可靠且全部互不相同的辅助符号。迭代排序和冗余消除操作是指以这样的方式对第一和第二辅助消息中包括的辅助分量进行迭代地处理：在每次迭代时处理至少辅助分量。迭代可以对应于处理时间间隔（例如，时钟周期）。

[0164] 将参考在每次迭代时处理辅助分量来进行以下描述，根据该辅助分量，排序和冗余消除单元313可以被配置为在每次迭代时处理辅助分量。然而，技术人员将容易理解的是，本公开适用于在一个或多个迭代的每一个时处理两个或更多个辅助分量。

[0165] 因此，计算单元311可以被配置为：

[0166] -根据检查节点消息V中包括的符号$V_i^{\oplus}$确定第一辅助消息U_{1aux}中包括的辅助符号$U_{1aux,i}^{\oplus}$（其中i＝0，…，n_{CN}-1），根据以下等式，辅助符号选自检查节点分量中包括的符号：

[0167] $$U_{1aux,i}^{\oplus} = V_i^{\oplus}, i = 0, \ldots, n_{CN} - 1 \qquad (5)$$

[0168] -通过在包括实数域、整数域和自然数域的组中选择的预定义代数结构上应用加法运算，确定与第一辅助消息U_{1aux}中包括的辅助符号$U_{1aux,i}^{\oplus}$相关联的每个辅助可靠性度量$U_{1aux,i}^{+}$，加法运算根据以下等式被应用于：与辅助符号$U_{1aux,i}^{\oplus}$相关联的固有可靠性度量$I^{+}\left(V_i^{\oplus}\right)$以及与等于所选择的辅助符号$U_{1aux,i}^{\oplus} = V_i^{\oplus}$的符号$V_i^{\oplus}$关联地被包括在检查节点分量$V_i = \left(V_i^{\oplus}, V_i^{+}\right)$中的可靠性度量$V_i^{+}$：

[0169] $$U_{1aux,i}^{+} = I^{+}\left(V_i^{\oplus}\right) + V_i^{+}, i = 0, \ldots, n_{CN} - 1 \qquad (6)$$

[0170] 根据一些实施例，非二进制纠错码解码器123可以根据等式(4)被进一步配置为：确定偏移值J，并且根据包括最不可靠符号和偏移值的检查节点分量来确定参考可靠性度

量C_{ref}^{+}。在这样的实施例中，计算单元311可以被配置为：

[0171] -根据与最高或最低固有可靠性度量$I^{+}(x_l)$(其中l＝0,…,n_{int}-1)相关联的预定义数量n_{int}的最可靠固有符号x_l，确定第二辅助消息U_{2aux}中包括的辅助符号$U_{2aux,l}^{\oplus}$，辅助符号$U_{2aux,l}^{\oplus}$根据以下等式等于固有符号x_l，其中l＝0,…,n_{int}-1：

[0172] $$U_{2aux,l}^{\oplus} = x_l, l = 0, \ldots, n_{int} - 1 \qquad (7)$$

[0173] -通过在预定义的代数结构上应用加法运算来确定与第二辅助消息U_{2aux}中包括的辅助符号$U_{2aux,l}^{\oplus}$相关联的辅助可靠性度量$U_{2aux,l}^{+}$，加法运算根据以下等式被应用于参考可靠性度量C_{ref}^{+}和关联于固有符号x_l的固有可靠性度量$I^{+}\left(U_{2aux,l}^{\oplus}\right)$，固有符号$x_l$等于所述辅助符号$x_l = U_{2aux,l}^{\oplus}$：

[0174] $$U_{2aux,l}^{+} = C_{ref}^{+} + I^{+}(x_l), l = 0, \ldots, n_{int} - 1 \qquad (8)$$

[0175] 根据一些实施例，解码器123可以被配置为：首先取决于在包括信噪比、解码过程的迭代次数、以及非二进制纠错码的代数结构的组中选择的一个或多个参数来确定与最可靠的固有符号相关联的固有可靠性度量的数量n_{int}。

[0176] 根据一些实施例，解码器123可以被配置为：取决于检查节点分量n_{CN}的数量来确定与最可靠的固有符号相关联的固有可靠性度量的数量n_{int}，该固有可靠性度量的数量严格小于检查节点分量的数量$n_{int}<n_{CN}$。

[0177] 根据一些实施例，非二进制纠错码解码器123可以被配置为根据预定义解码性能标准来确定偏移值J。

[0178] 根据一些实施例，非二进制纠错码解码器123可以被配置为：取决于可变节点处理单元31与检查节点处理单元33和/或检查节点处理单元35之间的消息交换的迭代次数来确定偏移值J，不同的偏移值J在解码处理的每次迭代时被动态地确定。

[0179] 此外，根据参考图2的一些实施例，非二进制纠错码解码器123可以被配置为针对每个可变节点处理单元27-vn(其中vn＝1,…,n)确定不同的偏移值J。

[0180] 根据图3所示的一些实施例，排序和冗余消除单元313可以被配置为执行迭代排序和冗余消除，以通过复用器和比较器的实施方式来生成可变节点分量。因此，在一些实施例中，排序和冗余消除单元313可以包括：

[0181] -多个复用器315-j，其中j＝1,…,n_{VN}；

[0182] -多个可靠性度量比较器319-j，其中j＝0,…,n_{VN}，其包括第一可靠性度量比较器319-0和一个或多个可靠性度量比较器319-j，其中j＝1,…,n_{VN}，以及

[0183] -多个符号比较器317-j，其中j＝0,…,n_{VN}-1，其包括第一符号比较器317-0和一个或多个符号比较器317-j，其中j＝1,…,n_{VN}-1。

[0184] 每个复用器315-j可以与可靠性度量比较器319-j和符号比较器317-j相关联。特别地，复用器315-1可以与可靠性度量比较器319-0和319-1以及符号比较器317-0相关联，复用器315-n_{VN}可以仅与可靠性度量比较器319-n_{VN}相关联。每个复用器315-j(其中j＝

1,…,n_{VN})可以被配置为:取决于由多个符号比较器317-j和多个可靠性度量比较器319-j执行的比较,在每次迭代时(例如,在每个时钟周期时)确定当前候选分量。特别地,排序和冗余消除单元313可以包括多个控制器318-j,其中j=1,…,n_{VN},每个控制器318-j与复用器315-j相关联并且被配置为在每次迭代时执行对复用器315-j的控制动作(也被称为"动作"),这取决于由符号比较器317-m执行的比较,其中m=0,…,j-1。

[0185] 即使检查节点分量被排序,第一辅助消息的辅助分量也可能不会被排序,这是由于加法运算涉及与检查节点消息中包括的符号相关联的固有可靠性度量。排序和冗余消除单元313因此用于根据辅助可靠性度量的给定顺序来对辅助分量进行排序,并且实现生成从排序的辅助分量导出的排序的可变节点分量。

[0186] 此外,在第一辅助消息和第二辅助消息中包括的辅助分量可以包括冗余符号,即,可以包括两个或更多个辅助分量,这些辅助分量包括一个或多个辅助符号的相同值。符号比较器317-j用于冗余消除,以便在辅助分量之中保留包括最可靠的不同符号的分量,并且实现生成包括不同符号的最可靠的可变节点分量。

[0187] 在每次迭代时(例如,在每个时钟周期时),由控制器318-j对复用器315-j执行的控制动作可以由初始化动作、保留动作、插入动作或移位动作组成。

[0188] 同时于处理在第一和第二辅助消息中包括的所有$n_{CN}+n_{int}$个辅助分量(即,同时于处理$n_{CN}+n_{int}$次迭代)(在第一辅助消息中包括的n_{CN}个辅助分量和在第二辅助消息中包括的n_{int}个辅助分量),排序和冗余消除单元313可以被配置为根据包括最可靠的不同符号的候选分量生成可变节点分量。

[0189] 根据其中在每次迭代时处理一个或多个辅助分量的一些实施例,每个复用器315-j(其中j=1,…,n_{VN})可以被配置为在每次迭代时(例如,在每个时钟周期时)接收从第一辅助消息或第二辅助消息中提取的至少辅助分量$C_{aux}=\left(C_{aux}^{\oplus},C_{aux}^{+}\right)=U_{aux,r}=\left(U_{aux,r}^{\oplus},U_{aux,r}^{+}\right)$以及预定义的初始化分量$C_{init}$,每个复用器315-j(其中j=1,…,$n_{VN}$)被配置为通过在至少辅助分量$C_{aux}$、预定义的初始化分量$C_{init}$、以及与由复用器315-j在先前迭代时确定的候选分量相对应的分量$C_j=\left(C_j^{\oplus},C_j^{+}\right)$之中选择分量来确定当前候选分量$C_{cand,j}$。在这样的实施例中:

[0190] -与复用器315-j相关联的可靠性度量比较器319-j可以被配置为执行在辅助分量C_{aux}中包括的至少一个辅助可靠性度量C_{aux}^{+}与由复用器315-j在先前的迭代时确定的候选分量C_j中包括的可靠性度量C_j^{+}之间的比较,以及

[0191] -与每个复用器315-j相关联的符号比较器317-j可以被配置为通过执行在辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$与由复用器315-j在先前的迭代时确定的候选分量C_j中包括的符号$C_j^{\oplus}$之间的比较来检查符号的冗余。

[0192] 在其中在每次迭代时处理辅助分量的一些实施例中,每个复用器315-j(其中j=1,…,n_{VN})可以被配置为:在每次迭代时(例如,在每个时钟周期时)接收从第一辅助消息或第二辅助消息提取的辅助分量以及预定义的初始化分量,并且通过在辅助分量C_{aux}、预定的初始化分量C_{init}和与由复用器315-j在先前的迭代时确定的候选分量相对应的分量

$C_j = \left(C_j^{\oplus}, C_j^{+}\right)$之中选择分量来确定当前候选分量$C_{cand,j}$。

[0193] 仅出于说明目的，将参考可靠性度量的升序并参考在每次迭代时处理辅助分量对以下实施例进行描述，根据该可靠性度量，最可靠的符号与最小的可靠性度量相关联。

[0194] 因此，在一些实施例中，接收辅助分量C_{aux}的每个复用器315-j(其中j＝1,…,n_{VN})可以被配置为：

[0195] -通过将当前候选分量$C_{cand,j}$设置为等于在先前的迭代时确定的候选分量C_j来执行保留动作：

[0196] *如果在辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$不冗余，其中没有由复用器315-m(其中m＝1,…,j-1)在先前迭代时确定的任何候选分量C_m(其中m＝1,…,j-1)(即，如果其中每个m在1和j-1之间变化，与每个复用器315-m相关联的每个符号比较器317-m确定$C_{aux}^{\oplus} \neq C_m^{\oplus}$)，并且在辅助分量$C_{aux}$中包括的辅助符号$C_{aux}^{\oplus}$比由复用器315-j在先前迭代时确定的候选分量$C_j$中包括的符号$C_j^{\oplus}$不可靠(即，如果对于可靠性度量的升序，与复用器315-j相关联的可靠性度量比较器319-j确定$C_{aux}^{+} \geq C_j^{+}$；或者

[0197] *如果在辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$冗余，其中由复用器315-1中的至少一个在先前迭代时确定的候选分量C_1的一个中包括的符号$C_l^{\oplus}$的至少一个，其中1在1与j-1之间变化(即，与复用器315-1相关联的至少一个符号比较器317-1确定对于取介于1与j-1之间的值的至少一个索引1，$C_{aux}^{\oplus} = C_l^{\oplus}$)；根据保留动作，复用器315-j不执行任何特定动作，并且保留先前在先前迭代期间选择的候选分量作为当前候选分量。因此，$C_{cand,j}^{\oplus} = C_j^{\oplus}$并且$C_{cand,j}^{+} = C_j^{+}$；

[0198] -如果在辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$不冗余，其中没有在先前迭代时确定的候选分量C_1(其中1＝1,…,j-1)中包括的任何符号$C_l^{\oplus}$(即，如果符号比较器317-1(其中1＝1,…,j-1)的每一个确定$C_{aux}^{\oplus} \neq C_l^{\oplus}$)，并且辅助分量$C_{aux}$中包括的辅助符号$C_{aux}^{\oplus}$比由复用器315-j在先前迭代时确定的候选分量$C_j$中包括的符号$C_j^{\oplus}$可靠并且辅助符号$C_{aux}^{\oplus}$比由复用器315-j-1在先前迭代时确定的候选分量C_{j-1}中包括的符号$C_{j-1}^{\oplus}$不可靠(即，如果与复用器315-j相关联的可靠性度量比较器319-j确定$C_{aux}^{+} < C_j^{+}$并且与复用器315-j-1相关联的可靠性度量比较器319-j-1确定$C_{aux}^{+} \geq C_{j-1}^{+}$)，则通过将当前候选分量$C_{cand,j}$设置为等于辅助分量$C_{aux}$来执行插入动作。根据插入动作，复用器315-j可以被配置为选择辅助分量作为当前候选分量，因此用辅助分量替换先前在先前迭代期间选择的候选分量，使得

$C_{cand,j}^{\oplus} = C_{aux}^{\oplus}$且$C_{cand,j}^{+} = C_{aux}^{+}$；

[0199] -如果在辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$不冗余，其中没有在先前迭代时确定的候选分量C_1中包括的任何符号$C_l^{\oplus}$，其中l＝1，…，j-1(即，如果每个符号比较器317-l(其中l＝1，…，j-1)确定$C_{aux}^{\oplus} \neq C_l^{\oplus}$)并且在辅助分量$C_{aux}$中包括的辅助符号$C_{aux}^{\oplus}$比由复用器315-j在先前迭代时确定的候选分量$C_j$中包括的符号$C_j^{\oplus}$可靠并且辅助符号$C_{aux}^{\oplus}$比由复用器315-j-1在先前迭代时确定的候选分量C_{j-1}中包括的符号$C_{j-1}^{\oplus}$可靠(即，如果与复用器315-j相关联的可靠性度量比较器319-j确定$C_{aux}^{+} < C_j^{+}$，并且与复用器315-j-1相关联的可靠性度量比较器319-j-1确定$C_{aux}^{+} < C_{j-1}^{+}$)，则通过将当前候选分量$C_{cand,j}$设置为等于由复用器315-(j-1)在先前迭代时确定的候选分量C_{j-1}来执行移位动作。根据移位动作，每个复用器315-j可以被配置为选择由复用器315-(j-1)确定的候选分量C_{j-1}作为当前候选分量$C_{cand,j}$，使得$C_{cand,j}^{\oplus} = C_{j-1}^{\oplus}$，$C_{cand,j}^{+} = C_{j-1}^{+}$。

[0200] 应当注意的是，对于与复用器315-1关联地执行的比较，在符号和可靠性度量比较期间考虑了预定义的候选分量C_0。技术人员将容易理解的是，鉴于预定义的候选分量包括预定义的候选符号和预定义的可靠性度量的恒定值，在这样的实施例中由控制器315-1确定的动作可以被有利地简化。

[0201] 根据一些实施例，为了在第一迭代时初始化候选分量，每个复用器315-j(其中j＝1，…，n_{VN})可以被配置为：通过将当前候选分量$C_{cand,j}$设置为等于初始化分量$C_{init} = (C_{init}^{\oplus}, C_{ref}^{+})$来执行初始化动作，使得$C_{cand,j}^{\oplus} = C_{init}^{\oplus}$且$C_{cand,j}^{+} = C_{init}^{+}$。初始化分量可以包括初始化符号和初始化可靠性度量的恒定值。因此，多个复用器315-j(其中j＝1，…，n_{VN})中的每一个可以被配置为根据所述初始化分量确定初始候选分量。因此，在开始处理从第一或第二辅助消息中提取的辅助分量之前，第j个复用器315-j可以被配置为将候选分量C_j和当前候选分量$C_{cand,j}$初始化为初始化分量C_{init}，使得候选符号和当前候选符号是根据$C_j^{\oplus} = C_{cand,j}^{\oplus} = C_{init}^{\oplus}$等于初始化符号的，并且与该候选符号$C_j^{\oplus}$相关联的可靠性度量$C_j^{+}$和关联于当前候选符号$C_{cand,j}^{\oplus}$的可靠性度量是根据$C_j^{+} = C_{cand,j}^{+} = C_{init}^{+}$等于初始化可靠性度量的。

[0202] 在一些实施例中，可以从不同于非二进制纠错码的构造的代数结构的代数结构中选择初始化符号。在一些实施例中，初始化可靠性度量可以是预定义的最大值。

[0203] 根据一些实施例，可变节点处理单元31可以被配置为从固有可靠性度量生成单元233接收固有可靠性度量，数据处理单元23被配置为在解码过程期间的每次迭代期间在线生成固有可靠性度量；例如，响应于由可变节点处理单元31递送的用于计算第一或第二辅助消息的辅助分量的固有可靠性度量生成请求。每当固有可靠性度量被要求用于可变节点

处理单元更新时，在线进行固有可靠性度量的计算实现了避免存储(memorization)所有固有可靠性度量，使得有利地节省了大量的内存和面积，特别是在考虑高阶Galois域时。

[0204] 在一些实施例中，可变节点处理单元31可以被配置为在每次迭代时从存储单元21加载固有可靠性度量，数据处理单元23被配置为预先离线计算至少一部分固有可靠性度量，并且将固有可靠性度量的计算出的部分存储在存储单元21中。

[0205] 图4是示出对在通信信道上接收的信号进行解码的方法的流程图，该信号使用在Galois域GF(q)上构造的非二进制纠错码$\mathcal{C}(n,K)$进行编码并且根据本公开的一些实施例进行解码，其中例如应用了使用泛洪调度的EMS算法。

[0206] 因此，解码过程可以包括两个阶段：初始化阶段和解码阶段。

[0207] 在步骤401处，接收到的信号y可以被接收。接收到的信号可以根据以下等式来写作被传送的码字的函数：

[0208] $y=c+w$ (9)

[0209] 在等式(9)中，w指定长度为n的向量，该向量对更改通信信道的噪声进行建模。

[0210] 在步骤403处，可以确定一个或多个解码参数。解码参数可以包括EMS迭代解码过程的最大迭代次数N_{max}、检查节点分量的数量n_{CN}、可变节点分量的数量n_{VN}、最可靠符号的预定义数量n_{int}、以及偏移值J。

[0211] 最大迭代次数N_{max}可以对解码性能和复杂性有影响。如果可用的计算和内存资源受到限制，或者如果解码延迟受到限制，则少量的迭代可以是优选的，以实现吞吐量增加。相反，例如在存在高噪声水平时，即在低信噪比的情况下，实现更好的解码性能的大量迭代可以是优选的。在一些实施例中，最大迭代次数可以因此根据噪声水平和/或系统可操作性的其他参数(例如，解码器存储器单元/缓冲器的大小)来调节。

[0212] 在被应用于EMS算法的一些实施例中，可以将检查节点分量的数量和可变节点分量的数量设置为严格小于Galois域的阶，即，$n_{CN}<q$且$n_{VN}<q$。

[0213] 在实施例中，可变节点分量的数量n_{VN}和检查节点分量的数量n_{CN}可以是不同的，即，$n_{VN}\neq n_{CN}$。

[0214] 在另一实施例中，可变节点分量的数量n_{VN}和检查节点分量的数量n_{CN}可以是相同的，即$n_{VN}=n_{CN}$。

[0215] 根据一些实施例，检查节点分量的数量n_{CN}可以进一步取决于非二进制纠错码(例如，在其上构造码的代数结构)和/或取决于迭代解码过程的解码迭代的顺序和/或取决于信噪比和/或取决于解码器的计算和存储能力和/或取决于解码过程的最大迭代次数。

[0216] 根据一些实施例，可以取决于在包括信噪比、解码过程的迭代次数、以及非二进制纠错码的代数结构的组中选择的一个或多个参数来确定固有可靠性度量的数量n_{int}。

[0217] 根据一些实施例，可以取决于检查节点分量的数量n_{CN}来确定最可靠的固有符号的数量n_{int}或等同地与最可靠的符号相关联的固有可靠性度量的数量，最可靠的固有符号的数量严格地小于检查节点分量的数量$n_{int}<n_{CN}$。

[0218] 根据一些实施例，可以根据预定义的解码性能标准来确定偏移值。

[0219] 根据一些实施例，可以取决于消息交换的迭代次数来确定偏移值，不同的偏移值在解码过程的每次迭代时被动态地确定(即，被更新)。

[0220] 在步骤405处，可以根据等式(3)来根据接收到的信号计算出与属于Galois域的最

可靠的固有符号x_l相关联的n_{int}个固有可靠性度量$I^+(x_l)$，l＝0,…,n_{int}-1，以及与在检查节点分量V_i中包括的符号$V_i^{\oplus}$相关联的n_{CN}个固有可靠性度量$I^+\left(V_i^{\oplus}\right)$，i＝0,…,$n_{CN}$-1，固有可靠性度量要被用于初始化可变节点消息。

[0221] 计算出的固有可靠性度量在迭代解码过程期间被用作要由可变节点处理单元递送的初始可变节点消息。在多次迭代期间交换承载这样的可靠性度量的可变节点消息和检查节点消息实现处理信号$z＝(z_1,z_2,\cdots,z_n)$，以用于生成估计$\hat{u}$，使其更接近MAP估计$\hat{u}_{MAP}$。对于要成为码字的z，它应当满足所有奇偶校验等式，即，$H.z^t＝0$。可以使用最可靠的符号β_i将解码的消息z初始化为$z＝(\beta_1,\beta_2,\cdots,\beta_n)$。

[0222] 因此，在步骤409处，迭代次数可以被初始化，即，设置为零，$N_{iter}＝0$。

[0223] 可以执行步骤411以检查是否满足奇偶校验约束(即，奇偶校验等式)。

[0224] 如果在步骤411处确定处理后的信号z满足奇偶校验等式并且满足$H.z^t＝0$，则可以停止解码处理，并且可以在步骤413处递送解码的码字$\hat{u}＝z$。

[0225] 如果在步骤411处确定不满足奇偶校验约束，则可以执行步骤415以检查是否达到最大迭代次数N_{max}，即，检查是否$N_{iter}＝N_{max}$。如果确定被执行的迭代次数达到最大迭代次数，则在步骤417处可以停止解码过程并且可以声明解码失败。如果确定尚未达到最大迭代次数，则可以执行步骤419至425以运行检查节点消息和可变节点消息的交换的一次迭代，检查节点消息包括n_{CN}个检查节点分量，而可变节点消息包括n_{VN}个可变节点分量。

[0226] 在一些实施例中，检查节点分量的数量和可变节点分量的数量可以是不同的。

[0227] 在步骤419处，可以根据可变节点消息来计算出检查节点消息。可以根据可靠性度量的给定顺序(递增或递减)来对每个检查节点消息中包括的检查节点分量进行排序。

[0228] 在步骤421处，可以计算可变节点消息，每个可变节点消息根据一个检查节点消息和$n_{int}+n_{VN}$个固有可靠性度量来计算。可以根据可靠性度量的给定顺序(递增或递减)来对每个可变节点消息中包括的可变节点分量进行排序。

[0229] 在步骤423处，可以根据在可变节点分量中包括的可靠性度量来更新处理后的信号z。

[0230] 在步骤425处，可以根据$N_{iter}＝N_{iter}+1$来递增消息交换的迭代次数。

[0231] 还提供了一种方法，该方法根据一个检查节点消息和从接收到的信号导出的$n_{int}+n_{VN}$个固有可靠性度量来确定一个可变节点消息。所提供的方法对应于在步骤421处执行的一个可变节点消息的计算。

[0232] 图5是描绘根据本公开的一些实施例的计算一个可变节点消息U的可变节点分量$U_t=\left(U_t^{\oplus},U_t^{+}\right)$(其中$t＝0,\cdots,n_{VN}$-1)的方法的流程图，其中迭代地确定可变节点分量，根据一个检查节点消息V和$n_{int}+n_{VN}$个固有可靠性度量在每次迭代时确定一个可变节点分量。

[0233] 在步骤501处，可以接收一个检查节点消息V，其包括：n_{CN}个检查节点分量$V_i=\left(V_i^{\oplus},V_i^{+}\right)$(其中$i＝0,\cdots,n_{CN}$-1)、与$n_{int}$个最可靠的固有符号$x_l\in GF(q)$相关联的$n_{int}$个固有可靠性度量$I^+(x_l)$、以及与检查节点消息V中包括的符号相关联的$n_{CN}$个固有可靠性度量$I^+\left(V_i^{\oplus}\right)$。

[0234] 根据一些实施例，可以串行地或顺序地接收n_{CN}个检查节点分量，使得在每次迭代

时(例如,在每个时钟周期时)接收检查节点分量。

[0235] 可以根据从检查节点消息V和固有可靠性度量$I^+(x_1)$和$I^+\left(V_i^{\oplus}\right)$导出的$U_{aux,r}=\left(U_{aux,r}^{\oplus},U_{aux,r}^{+}\right)$表示的$n_{int}+n_{VN}$个辅助分量来确定可变节点分量,一个或多个辅助分量在每次迭代时被处理。辅助分量$U_{aux,r}=\left(U_{aux,r}^{\oplus},U_{aux,r}^{+}\right)$可以包括辅助符号$U_{aux,r}^{\oplus}$和关联于符号$U_{aux,r}^{\oplus}$的辅助可靠性度量$U_{aux,r}^{+}$。

[0236] 在步骤503处,可以根据检查节点消息V和关联于检查节点消息V中包括的符号的固有可靠性度量$I^+\left(V_i^{\oplus}\right)$确定第一辅助消息$U_{1aux}$。

[0237] 在一些实施例中,第一辅助消息U_{1aux}可以包括n_{CN}个辅助分量$U_{1aux,i}=\left(U_{1aux,i}^{\oplus},U_{1aux,i}^{+}\right)$,其中$i=0,\cdots,n_{CN}-1$,根据等式(5),辅助符号$U_{1aux,i}^{\oplus}$根据检查节点消息中包括的符号来确定,并且根据等式(6),与该辅助符号$U_{1aux,i}^{\oplus}$相关联的辅助可靠性度量$U_{1aux,i}^{+}$通过对与辅助符号$V_i^{\oplus}$相关联的固有可靠性度量$I^+\left(V_i^{\oplus}\right)$和包括在检查节点分量$V_i=\left(V_i^{\oplus},V_i^{+}\right)$中的可靠性度量$V_i^{+}$应用加法运算来与等于辅助符号$U_{1aux,i}^{\oplus}=V_i^{\oplus}$的符号$V_i^{\oplus}$关联地确定。

[0238] 在步骤505处,可以根据与预定义数量n_{int}的最可靠的n_{int}个符号$x_1\in GF(q)$相关联的固有可靠性度量$I^+(x_1)$来确定第二辅助消息U_{2aux}。

[0239] 在一些实施例中,第二辅助消息U_{2aux}可以包括n_{int}个辅助分量$U_{2aux,l}=\left(U_{2aux,l}^{\oplus},U_{2aux,l}^{+}\right)$(其中$l=0,\cdots,n_{int}-1$),根据等式(7),第二辅助消息$U_{2aux}$中包括的辅助符号$U_{2aux,l}^{\oplus}$根据与最高或最低固有可靠性度量$I^+(x_1)$(其中$l=0,\cdots,n_{int}-1$)相关联的最可靠固有符号$x_1$来确定,并且根据等式(8),与在第二辅助消息$U_{2aux}$中包括的辅助符号$U_{2aux,l}^{\oplus}$相关联的可靠性度量$U_{2aux,l}^{+}$通过将加法运算应用于参考可靠性度量以及与固有符号$x_1$相关联的固有可靠性度量$I^+(x_1)$来确定。

[0240] 根据一些实施例,其中已经预先计算和存储了至少一部分固有可靠性度量$I^+(x_1)$(其中$x_1\in GF(q)$)(例如在图4所示的初始化阶段期间被离线计算出),可以从存储单元加载固有可靠性度量,固有可靠性度量每当要求计算第一或第二辅助消息的辅助分量时就被加载。

[0241] 在其他实施例中,响应于每当计算第一或第二辅助消息的辅助分量而要求固有可靠性度量时递送的固有可靠性度量请求,固有可靠性度量可以被在线(即,在解码阶段期间)计算。

[0242] 在其他实施例中,至少一部分固有可靠性度量可以被离线计算,并且至少一部分固有可靠性度量可以响应于递送的固有可靠性度量请求而被在线计算。

[0243] 重新组合第一和第二辅助消息的辅助分量可以不被排序(由于涉及固有可靠性度量的计算),并且可以包括冗余符号。因此,要求排序和冗余消除操作,以便根据辅助分量生

成包括最可靠的不同符号的可变节点分量。

[0244] 在步骤507处，可以根据在第一和第二辅助消息中包括的辅助分量迭代地确定n_{VN}个可变节点分量，可变节点分量通过在每次迭代时处理从第一或第二辅助消息中提取的至少辅助分量来确定，该处理包括根据辅助可靠性度量的给定顺序来对辅助分量进行迭代地排序，并且保留包括全部互不相同且最可靠的辅助符号的预定义数量n_{VN}的辅助分量。

[0245] 图6是示出根据本公开的一些实施例的在步骤507处使用的排序和冗余消除方法的流程图，该排序和冗余消除方法用于根据第一和第二辅助消息中包括的辅助分量来确定n_{VN}个可变节点分量，其中在预定义次数的迭代期间同时地执行排序和冗余消除。

[0246] 在时钟周期期间执行迭代的实施例中，排序和冗余消除要求$\Delta=n_{CN}+n_{int}$个时钟周期的延时，并减少了计算可变节点分量所要求的内存，因此带来在计算负载、延时和存储方面针对可变节点处理的更高效的架构。

[0247] 此外，在流水线式的实施例中，其中检查节点分量被在线连续接收(即，在排序和冗余消除过程的迭代的每次迭代时接收检查节点分量)，可变节点消息生成在时钟周期数方面的延时可以有利地减少到$\Delta=1+n_{int}$。

[0248] 根据一些实施例，排序和冗余消除方法可以基于在每次迭代时对n_{VN}个当前候选分量$C_{cand,j}=\left(C_{cand,j}^{\oplus},C_{cand,j}^{+}\right)$(其中$j=1,\cdots,n_{VN}$)的确定，该确定取决于：

[0249] -在从第一或第二辅助消息中提取的至少辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$与在先前迭代时确定的候选分量C_j的至少一个中包括的候选符号$C_j^{\oplus}$之间执行的比较，以及

[0250] -在从第一或第二辅助消息中提取的至少辅助分量C_{aux}中包括的辅助可靠性度量C_{aux}^{+}与在先前迭代时确定的候选分量C_j的至少一个中包括的候选可靠性C_j^{+}之间执行的比较。

[0251] 取决于比较结果，可以在每次迭代时决定控制动作，以通过更新在先前迭代时预先确定的候选分量来确定当前候选分量。控制动作可以包括初始化动作、保留动作、插入动作或移位动作。

[0252] 在步骤601处，可以根据初始化分量C_{init}来执行用于初始化n_{VN}个候选分量C_j(其中$j=1,\cdots,n_{VN}$)和n_{VN}个当前候选分量$C_{cand,j}$的初始化动作，使得候选符号和当前候选符号根据$C_j^{\oplus}=C_{cand,j}^{\oplus}=C_{init}^{\oplus}$等于初始化符号，并且与候选符号$C_j^{\oplus}$相关联的可靠性度量$C_j^{+}$和关联于当前候选符号$C_{cand,j}^{\oplus}$的可靠性度量根据$C_j^{+}=C_{cand,j}^{+}=C_{init}^{+}$等于初始化可靠性度量。

[0253] 在步骤603处，可以根据$\Delta=0$来执行排序和冗余消除过程的迭代次数的初始化。在时钟周期期间执行迭代的实施例中，排序和冗余消除过程的迭代次数可以对应于排序和冗余消除处理时间段，即，在时钟周期数方面的延时。

[0254] 在步骤605处，可以通过检查排序和冗余消除过程的迭代次数Δ是否达到$n_{CN}+n_{int}$来控制在第一和第二辅助分量中包括的辅助分量的总数$n_{CN}+n_{int}$的处理。

[0255] 如果在步骤605处确定$\Delta=n_{CN}+n_{int}$，则这意味着第一和第二辅助消息中包括的所

有辅助分量已经被处理。因此，在步骤607处，可以停止排序和冗余消除操作，并且可以将n_{VN}个可变节点分量设置为当前候选分量(即，最后更新的候选分量)。

[0256] 如果在步骤605处确定$\Delta \neq n_{CN}+n_{int}$，更精确地说$\Delta < n_{CN}+n_{int}$，则存在第一辅助消息或第二辅助消息中包括的在排序和冗余消除处理期间尚未被处理的至少一个辅助分量。因此，在步骤609处，可以接收从第一或第二辅助消息中提取的至少辅助分量$C_{aux} = (C_{aux}^{\oplus}, C_{aux}^{+})$(未被处理的辅助分量)。步骤611至623可以被执行以确定n_{VN}个当前候选分量$C_{cand,j}$，其中$j=1,\cdots,n_{VN}$。

[0257] 将参考在每次迭代时处理辅助分量来进行以下描述。技术人员将容易理解的是，本公开的各种实施例适用于在一次或多次迭代的每一次时对两个或更多个辅助分量的处理。

[0258] 在步骤611处，可以接收预定义的候选分量C_0，并且可以执行以下比较：在从第一或第二辅助分量中提取的接收到的辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$与在候选分量C_j(其中$j=1,\cdots,n_{VN}$)中包括的候选符号$C_j^{\oplus}$的每一个之间的符号比较，以及辅助可靠性度量C_{aux}^{+}+与候选分量C_j(其中$j=1,\cdots,n_{VN}$)中包括的候选可靠性度量C_j^{+}的每一个之间的可靠性度量比较。

[0259] 可以在步骤617、621和623处执行控制动作，以取决于符号和可靠性度量比较的结果并行地确定n_{VN}个当前候选分量$C_{cand,j}$(其中$j=1,\cdots,n_{VN}$)。更具体地：

[0260] -在以下条件下在步骤621处可以执行插入动作以将当前候选分量$C_{cand,j}$设置为等于辅助分量C_{aux}：如果在步骤613处确定辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$不冗余，其中在先前迭代时确定的候选分量C_m中不包括任何符号$C_m^{\oplus}$(其中$m=1,\cdots,j-1$)(即，如果在步骤613处确定$C_{aux}^{\oplus} \neq C_m^{\oplus}$，其中$m=1,\cdots,j-1$)，并且在步骤615处确定辅助分量$C_{aux}$中包括的辅助符号$C_{aux}^{\oplus}$比在先前迭代时确定的候选分量$C_{j-1}$中包括的符号$C_{j-1}^{\oplus}$可靠，并且在步骤619处确定辅助符号$C_{aux}^{\oplus}$比在先前迭代时确定的候选分量$C_j$中包括的符号$C_j^{\oplus}$不可靠(即，如果在当前迭代的步骤615处确定$C_{aux}^{+} < C_{j-1}^{+}$，并且在步骤619处确定$C_{aux}^{+} < C_j^{+}$)。在涉及索引$j=1$的比较中，考虑了预定义的候选分量$C_0$。根据插入动作，可以通过用辅助分量替换在先前迭代期间预先选择的候选分量来确定当前候选分量$C_{cand,j}$，使得$C_{cand,j}^{\oplus} = C_{aux}^{\oplus}$且$C_{cand,j}^{+} = C_{aux}^{+}$；

[0261] -在以下条件下可以在步骤615处执行移位动作以将当前候选分量$C_{cand,j}$设置为等于在先前迭代时确定的候选分量C_{j-1}：如果在步骤613处确定辅助分量C_{aux}中包括的辅助符$C_{aux}^{\oplus}$不冗余，其中在先前迭代时确定的候选分量C_m(其中$m=1,\cdots,j-1$)中不包括任何符号$C_m^{\oplus}$(即，如果在步骤613处确定$C_{aux}^{\oplus} \neq C_m^{\oplus}$，其中$m=1,\cdots,j-1$)，并且在步骤615处确定在

辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$比在先前迭代时确定的候选分量C_{j-1}中包括的符号$C_{j-1}^{\oplus}$可靠(即,如果在当前迭代的步骤615处确定$C_{aux}^{+} < C_{j-1}^{+}$)。根据移位动作,可以确定当前候选分量$C_{cand,j}$,使得$C_{cand,j}^{\oplus} = C_{j-1}^{\oplus}$,$C_{cand,j}^{+} = C_{j-1}^{+}$,以及

[0262] -可以在步骤623处执行保留动作,以通过将当前候选分量$C_{cand,j}$设置为等于在先前迭代时确定的候选分量C_j来确定当前候选分量:

[0263] *如果在步骤613处确定在辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$不冗余,其中没有任何在先前迭代时确定的候选分量C_m(其中m=1,…,j-1)(即,如果在1与j-1之间变化的索引m的先前迭代的每一个的每个步骤611处,确定$C_{aux}^{\oplus} \neq C_m^{\oplus}$),并且在步骤619处确定在辅助分量$C_{aux}$中包括的辅助符号$C_{aux}^{\oplus}$比在先前迭代时确定的候选分量$C_j$中包括的符号$C_j^{\oplus}$不可靠(即,如果对于可靠性度量的升序,在先前迭代的步骤615处确定$C_{aux}^{+} \geq C_j^{+}$);或者

[0264] *如果在步骤613处确定在辅助分量C_{aux}中包括的辅助符号$C_{aux}^{\oplus}$冗余,其中在先前迭代时确定的候选分量C_m中的一个中包括的符号$C_m^{\oplus}$中的至少一个,其中m在1与j-1之间变化(即,如果在至少一个先前迭代的步骤613处确定对于取介于1与j-1之间的值的至少一个索引m,$C_{aux}^{\oplus} = C_m^{\oplus}$)。

[0265] 根据保留动作,当前候选分量$C_{cand,j}$可以被设置为等于在先前迭代期间预先选择的候选分量,使得$C_{cand,j}^{\oplus} = C_j^{\oplus}$且$C_{cand,j}^{+} = C_j^{+}$。

[0266] 可以执行步骤625以根据Δ=Δ+1来递增被执行的迭代的次数。

[0267] 本文所描述的方法和设备可以通过各种手段来实现。例如,可以以硬件、软件或其组合来实现这些技术。针对硬件实施方式,可以例如根据仅硬件的配置(例如,在具有对应的存储器的一个或多个FPGA、ASIC或VLSI集成电路中)或者根据使用VLSI和DSP二者的配置来实现迭代解码器123的处理元件。

[0268] 尽管已经通过各种示例的描述示出了本公开的实施例,并且尽管已经相当详细地描述了这些实施例,但是申请人的意图不是将所附权利要求的范围约束或以任何方式限制到这样的细节。

[0269] 特别地,尽管已经参考EMS算法的特定的实施方式执行了本公开的一些实施例的描述,但是应当注意的是,本公开还可以应用于诸如最小-最大算法之类的其他迭代解码算法。

[0270] 此外,尽管已经参考在Galois域上构造的纠错码描述了本公开的一些实施例,但是本领域技术人员将容易理解的是,所提出的实施例也可以应用于任何非二进制LDPC码和在非交换组上构造的任何非二进制图纠错码,例如多项式码(例如,循环码)。

[0271] 此外,即使本公开在对通信系统的应用中具有一些优点,也应当注意,本公开不限于这样的通信设备,并且可以被集成在诸如数据存储设备之类的许多设备中。

[0272] 可以通过被供应给任何类型的计算机的处理器的计算机程序指令来实现本文所

描述的方法，以产生具有处理器的机器，该处理器执行指令以实现本文所指定的功能/动作。这些计算机程序指令还可以被存储在计算机可读介质中，该计算机可读介质可以指导计算机以特定的方式运作。为此，可以将计算机程序指令加载到计算机上以引起一系列操作步骤的执行，并且从而产生计算机实现的过程，使得被执行的指令提供用于实现本文所指定的功能的过程。

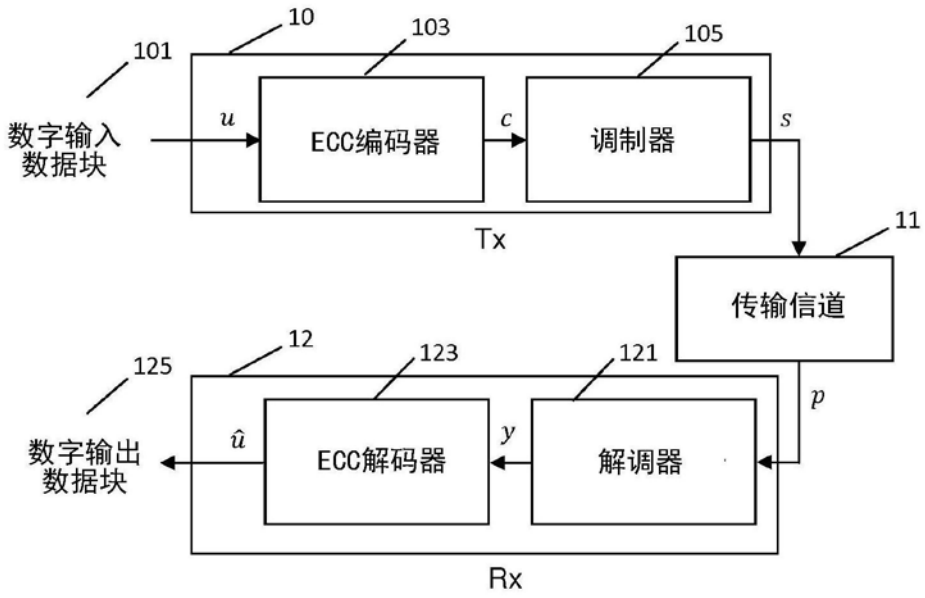

10
101
103
105
数字输入
数据块
u
ECC编码器
c
调制器
s
Tx
11
传输信道
12
125
123
121
p
数字输出
数据块
û
ECC解码器
y
解调器
Rx

图1

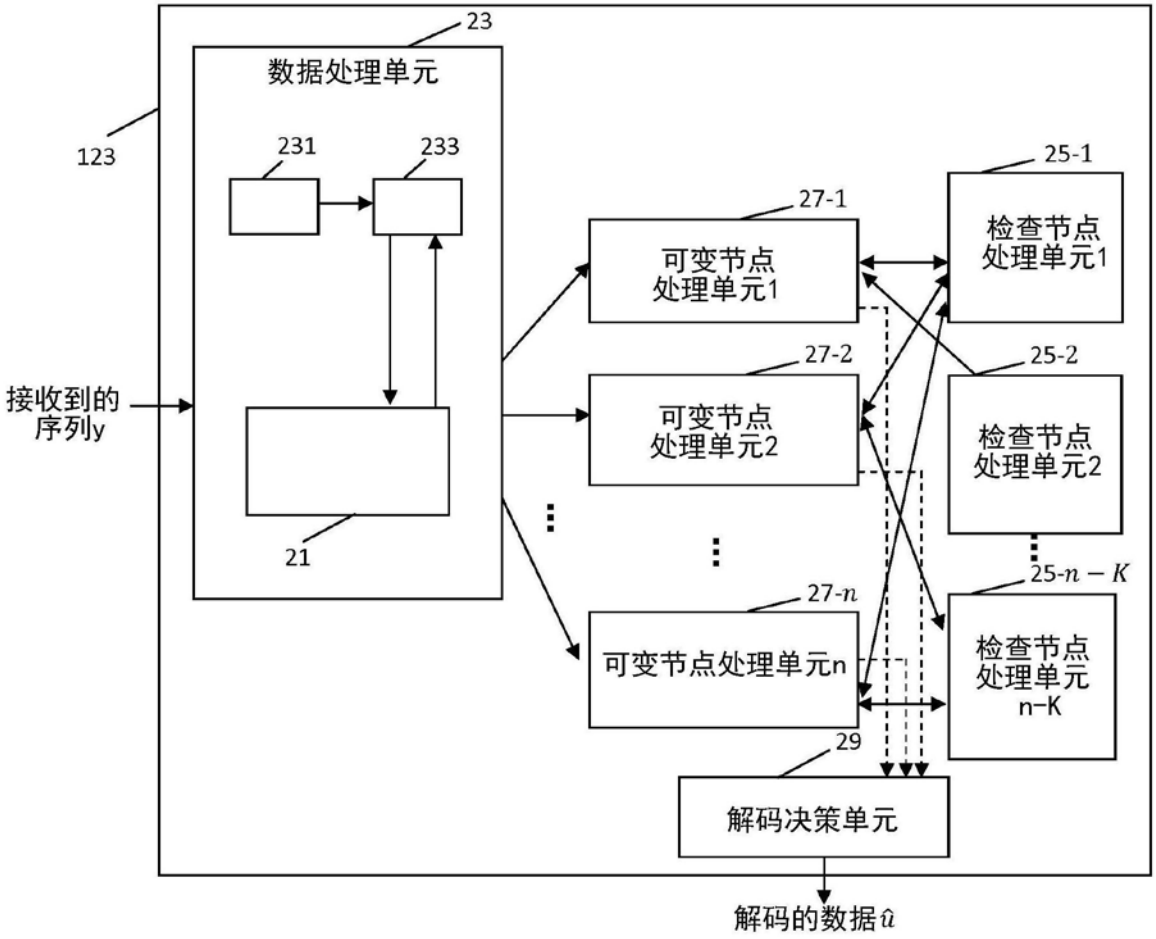

23
数据处理单元
123
231
233
21
接收到的
序列y
27-1
可变节点
处理单元1
25-1
检查节点
处理单元1
27-2
可变节点
处理单元2
25-2
检查节点
处理单元2
25-n − K
27-n
可变节点处理单元n
检查节点
处理单元
n-K
29
解码决策单元
解码的数据û

图2

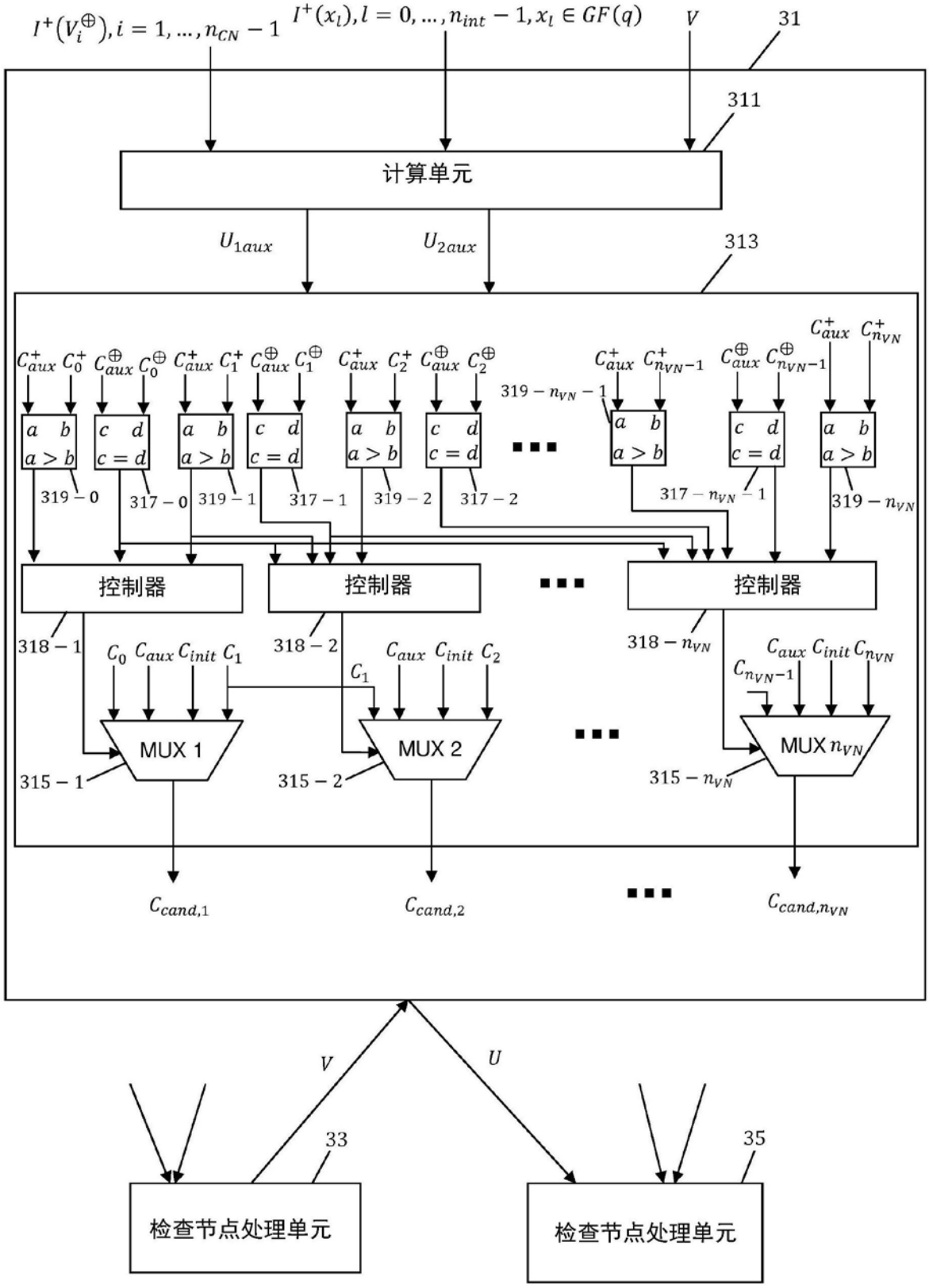

$I^+(V_i^{\oplus}), i = 1, \ldots, n_{CN} - 1$
$I^+(x_l), l = 0, \ldots, n_{int} - 1, x_l \in GF(q)$
V
31
311
计算单元
U_{1aux}
U_{2aux}
313
C_{aux}^+ C_0^+ $C_{aux}^{\oplus}$ $C_0^{\oplus}$ C_{aux}^+ C_1^+ $C_{aux}^{\oplus}$ $C_1^{\oplus}$ C_{aux}^+ C_2^+ $C_{aux}^{\oplus}$ $C_2^{\oplus}$
C_{aux}^+ $C_{n_{VN}-1}^+$ $C_{aux}^{\oplus}$ $C_{n_{VN}-1}^{\oplus}$ C_{aux}^+ $C_{n_{VN}}^+$
$319 - n_{VN} - 1$
$a\ b$ $a > b$
$c\ d$ $c = d$
$319 - 0$
$317 - 0$
$319 - 1$
$317 - 1$
$319 - 2$
$317 - 2$
$317 - n_{VN} - 1$
$319 - n_{VN}$
控制器
控制器
控制器
$318 - 1$
$318 - 2$
$318 - n_{VN}$
C_0 C_{aux} C_{init} C_1
C_1
C_{aux} C_{init} C_2
C_{aux} C_{init} $C_{n_{VN}}$
$C_{n_{VN}-1}$
MUX 1
MUX 2
MUX n_{VN}
$315 - 1$
$315 - 2$
$315 - n_{VN}$
$C_{cand,1}$
$C_{cand,2}$
$C_{cand,n_{VN}}$
V
U
33
35
检查节点处理单元
检查节点处理单元

图3

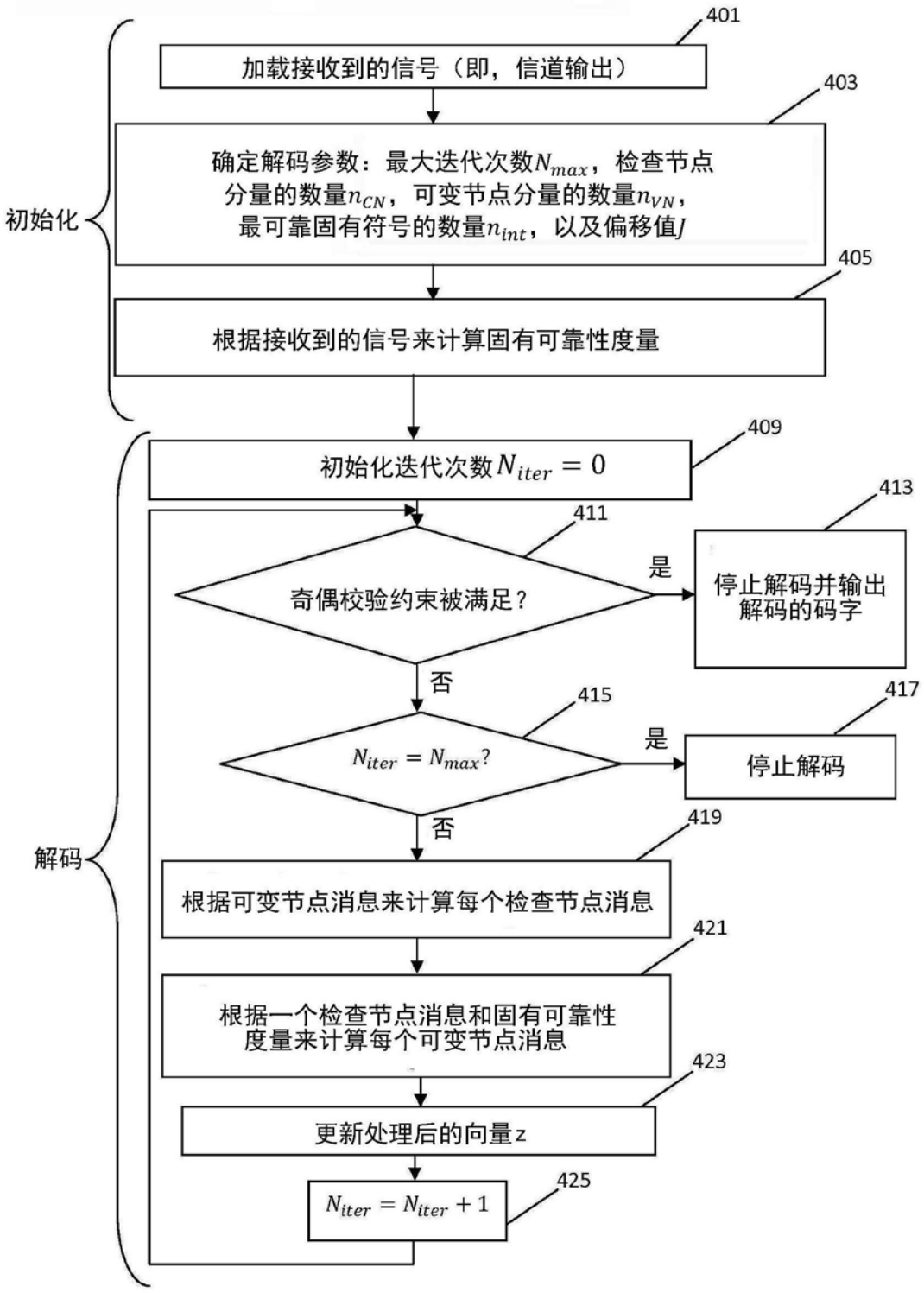

401
加载接收到的信号（即，信道输出）
403
确定解码参数：最大迭代次数N_{max}，检查节点分量的数量n_{CN}，可变节点分量的数量n_{VN}，最可靠固有符号的数量n_{int}，以及偏移值J
初始化
405
根据接收到的信号来计算固有可靠性度量
409
初始化迭代次数$N_{iter}=0$
411
奇偶校验约束被满足？
是
413
停止解码并输出解码的码字
否
415
$N_{iter}=N_{max}$?
是
417
停止解码
否
419
根据可变节点消息来计算每个检查节点消息
421
根据一个检查节点消息和固有可靠性度量来计算每个可变节点消息
423
更新处理后的向量z
$N_{iter}=N_{iter}+1$
425
解码

图4

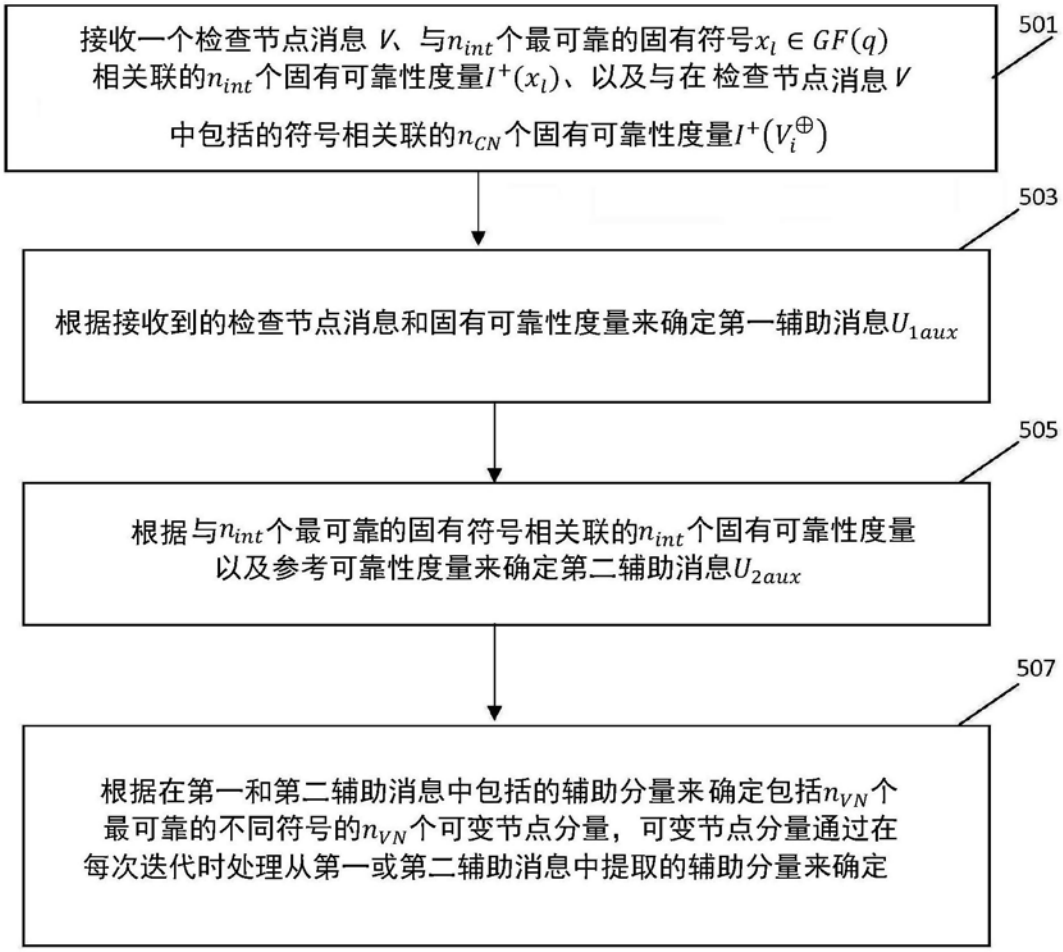

接收一个检查节点消息 V、与n_{int}个最可靠的固有符号$x_l \in GF(q)$相关联的n_{int}个固有可靠性度量$I^+(x_l)$、以及与在 检查节点消息 V 中包括的符号相关联的n_{CN}个固有可靠性度量$I^+(V_i^{\oplus})$
501
根据接收到的检查节点消息和固有可靠性度量来确定第一辅助消息U_{1aux}
503
根据与n_{int}个最可靠的固有符号相关联的n_{int}个固有可靠性度量以及参考可靠性度量来确定第二辅助消息U_{2aux}
505
根据在第一和第二辅助消息中包括的辅助分量来确定包括n_{VN}个最可靠的不同符号的n_{VN}个可变节点分量，可变节点分量通过在每次迭代时处理从第一或第二辅助消息中提取的辅助分量来确定
507

图5

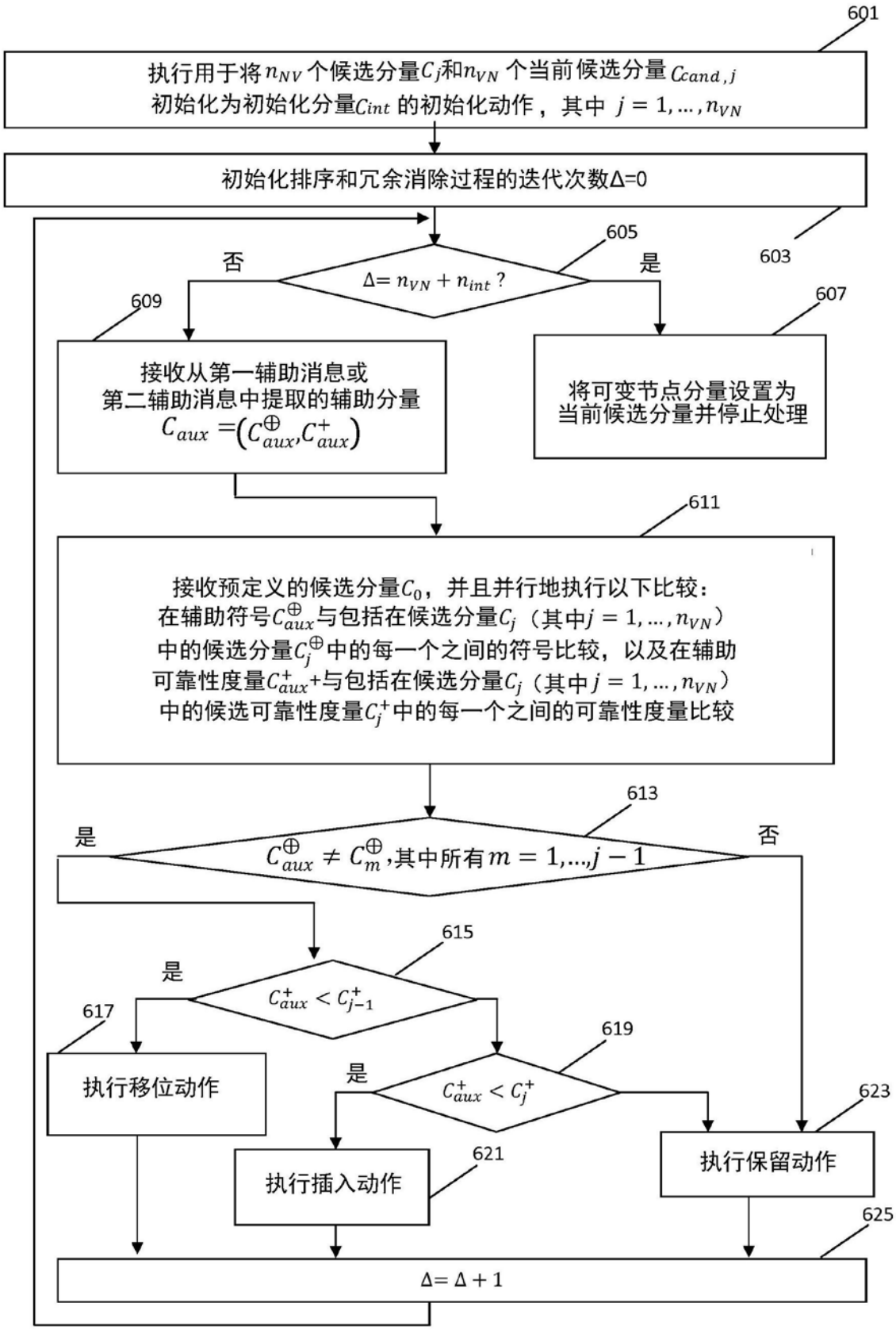

601
执行用于将n_{NV}个候选分量C_j和n_{VN}个当前候选分量$C_{cand,j}$初始化为初始化分量C_{int}的初始化动作，其中$j=1,\ldots,n_{VN}$
初始化排序和冗余消除过程的迭代次数Δ=0
603
605
$\Delta=n_{VN}+n_{int}$?
否
是
609
接收从第一辅助消息或第二辅助消息中提取的辅助分量$C_{aux}=(C_{aux}^{\oplus},C_{aux}^{+})$
607
将可变节点分量设置为当前候选分量并停止处理
611
接收预定义的候选分量C_0，并且并行地执行以下比较：在辅助符号$C_{aux}^{\oplus}$与包括在候选分量C_j（其中$j=1,\ldots,n_{VN}$）中的候选分量$C_j^{\oplus}$中的每一个之间的符号比较，以及在辅助可靠性度量C_{aux}^{+}+与包括在候选分量C_j（其中$j=1,\ldots,n_{VN}$）中的候选可靠性度量C_j^{+}中的每一个之间的可靠性度量比较
613
$C_{aux}^{\oplus}\neq C_m^{\oplus}$,其中所有$m=1,\ldots,j-1$
是
否
615
$C_{aux}^{+}<C_{j-1}^{+}$
是
617
执行移位动作
619
$C_{aux}^{+}<C_j^{+}$
是
621
执行插入动作
623
执行保留动作
625
Δ=Δ+1

图6