



(12)发明专利

(10)授权公告号 CN 108292967 B

(45)授权公告日 2020.07.07

(21)申请号 201580083408.2

(22)申请日 2015.09.30

(65)同一申请的已公布的文献号
申请公布号 CN 108292967 A

(43)申请公布日 2018.07.17

(85)PCT国际申请进入国家阶段日
2018.03.28

(86)PCT国际申请的申请数据
PCT/CN2015/091202 2015.09.30

(87)PCT国际申请的公布数据
W02017/054164 ZH 2017.04.06

(73)专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼
专利权人 香港理工大学

(72)发明人 金丽丽 刘重明

(74)专利代理机构 北京同立钧成知识产权代理有限公司 11205

代理人 马爽

(51)Int.Cl.
H04L 1/00(2006.01)

(56)对比文件
CN 104219019 A, 2014.12.17,
CN 103825669 A, 2014.05.28,
CN 103281166 A, 2013.09.04,
KR 101496182 B1, 2015.03.09,
审查员 魏玉翀

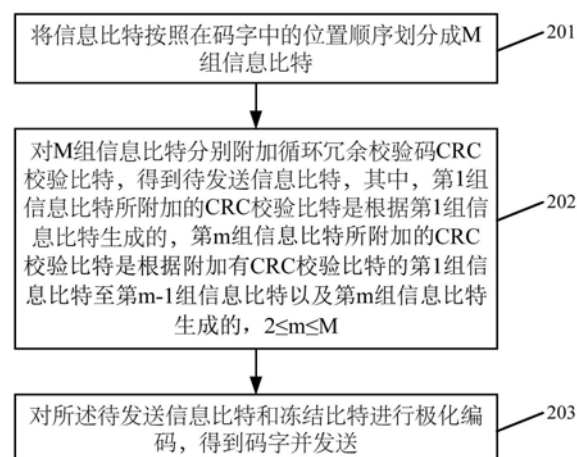
权利要求书4页 说明书15页 附图6页

(54)发明名称

极化码的编译码方法及其装置

(57)摘要

本发明实施例提供一种极化码的编译码方法及其装置。极化码的译码方法,包括:接收码字,所述码字包含接收比特和冻结比特;从所述码字中提取接收比特,并对所述接收比特按照在码字中的位置顺序划分成M组接收比特,其中,各组接收比特均包含循环冗余校验码CRC校验比特,M为大于等于2的整数;对所述M组接收比特进行SCL译码处理,并将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;其中,所述译码处理包括:对第m-1组接收比特进行L条路径的SCL译码处理,并对各L条路径的译码结果分别与第1至第m-2组接收比特的最终译码结果一同进行CRC校验,若L条路径的译码结果中存在能够通过CRC校验的路径,则开始对第m组接收比特进行SCL译码处理;否则对L翻倍,并从第1组开始重新进行SCL译码处理,直到L达到路径数上限 L_{\max} 且m达到M。



1. 一种极化码的编码方法,其特征在于,包括:

将信息比特按照在码字中的位置顺序划分成M组信息比特,其中M为大于等于2的整数;

对M组信息比特分别附加循环冗余校验码CRC校验比特,得到待发送信息比特,其中,第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的,第m组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第m-1组信息比特以及第m组信息比特生成的, $2 \leq m \leq M$;

对所述待发送信息比特和冻结比特进行极化编码,得到码字并发送。

2. 根据权利要求1所述的方法,其特征在于,所述将信息比特按照在码字中的位置顺序划分成M组信息比特,包括:

将信息比特按照在码字中的位置顺序等分成M组,得到M组信息比特。

3. 根据权利要求1或2所述的方法,其特征在于,所述对M组信息比特分别附加循环冗余校验码CRC校验比特,包括:

将所述CRC校验比特分别附加在M组信息比特中各组信息比特的尾部。

4. 一种极化码的译码方法,其特征在于,包括:

接收码字,所述码字包含接收比特和冻结比特;

从所述码字中提取接收比特,并对所述接收比特按照在码字中的位置顺序划分成M组接收比特,其中,各组接收比特均包含循环冗余校验码CRC校验比特,M为大于等于2的整数;

对所述M组接收比特进行SCL译码处理,并将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;其中,所述译码处理包括:对第m-1组接收比特进行L条路径的SCL译码处理,并对各L条路径的译码结果分别与第1至第m-2组接收比特的最终译码结果一同进行CRC校验,若L条路径的译码结果中存在能够通过CRC校验的路径,则开始对第m组接收比特进行SCL译码处理;否则对L翻倍,并从第1组开始重新进行SCL译码处理,直到L达到路径数上限 L_{\max} 且m达到M。

5. 根据权利要求4所述的方法,其特征在于,所述对所述M组接收比特进行SCL译码处理,并将M组接收比特对应的最终译码结果与冻结比特进行组合并输出,包括:

S601、初始化处理,其中, $L=1, L_{\max}=L_S$;

S602、令 $m=1$;

S603、对第m组接收比特进行SCL译码,得到L条路径的译码结果;

S604、对L条路径的译码结果分别进行CRC校验,得到与各条路径的译码结果对应的CRC校验结果,其中,对第m组接收比特的L条路径的译码结果进行CRC校验时,将各条路径的译码结果与已经完成译码的第1~m-1组接收比特的译码结果一起进行CRC校验,得到与L条路径的译码结果分别对应的CRC校验结果;

S605、根据各CRC校验结果,判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验;若是,则执行S606,否则执行S610;

S606、将通过校验的译码结果中最优的译码结果作为第m组接收比特对应的最终译码结果;

S607、判断m是否等于M,若是,则执行S609,否则执行S608;

S608、令 $m=m+1$,并执行S603;

S609、将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;

S610、判断 L 是否等于 L_{\max} ，若是，则执行S611，否则执行S612；

S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第 m 组接收比特对应的最终译码结果，并执行S607；

S612、令 $L=2L$ ，并执行S602。

6. 根据权利要求4或5所述的方法，其特征在于，所述对所述 M 组接收比特进行SCL译码处理之前，还包括：

根据当前信息比特的接收速度和/或接收缓冲器的剩余空间，对所述 L_{\max} 进行调整。

7. 一种编码设备，其特征在于，包括：

分组模块，用于将信息比特按照在码字中的位置顺序划分成 M 组信息比特，其中 M 为大于等于2的整数；

编码处理模块，用于对 M 组信息比特分别附加循环冗余校验码CRC校验比特，得到待发送信息比特，其中，第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的，第 m 组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第 $m-1$ 组信息比特以及第 m 组信息比特生成的， $2 \leq m \leq M$ ；

编码发送模块，用于对所述待发送信息比特和冻结比特进行极化编码，得到码字并发送。

8. 根据权利要求7所述的设备，其特征在于，所述分组模块，具体用于将信息比特按照在码字中的位置顺序等分成 M 组，得到 M 组信息比特。

9. 一种编码设备，其特征在于，包括：分组器、 M 个CRC生成器以及编码器；其中，分组器与所述 M 个CRC生成器的输入端连接，第 m 个CRC生成器的输出端与所述编码器和第 $m+1$ 至第 M 个CRC生成器的输入端连接，第 M 个CRC生成器的输出端与所述编码器连接，其中 M 为大于等于2的整数；

所述分组器，用于将信息比特按照在码字中的位置顺序划分成 M 组信息比特；

所述 M 个CRC生成器，用于对 M 组信息比特分别附加循环冗余校验码CRC校验比特，得到待发送信息比特，其中，第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的，第 m 组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第 $m-1$ 组信息比特以及第 m 组信息比特生成的， $2 \leq m \leq M$ ；

所述编码器，用于对所述待发送信息比特和冻结比特进行极化编码，得到码字并发送。

10. 一种译码设备，其特征在于，包括：

分组模块，用于接收码字，所述码字包含接收比特和冻结比特；从所述码字中提取接收比特，并对所述接收比特按照在码字中的位置顺序划分成 M 组接收比特，其中，各组接收比特均包含循环冗余校验码CRC校验比特， M 为大于等于2的整数；

译码处理模块，用于对所述 M 组接收比特进行SCL译码处理，并将 M 组接收比特对应的最终译码结果与冻结比特进行组合并输出；其中，所述译码处理包括：对第 $m-1$ 组接收比特进行 L 条路径的SCL译码处理，并对各 L 条路径的译码结果分别与第1至第 $m-2$ 组接收比特的最终译码结果一同进行CRC校验，若 L 条路径的译码结果中存在能够通过CRC校验的路径，则开始对第 m 组接收比特进行SCL译码处理；否则对 L 翻倍，并从第1组开始重新进行SCL译码处理，直到 L 达到路径数上限 L_{\max} 且 m 达到 M 。

11. 根据权利要求10所述的设备，其特征在于，所述译码处理模块，具体用于：

S601、初始化处理,其中, $L=1$, $L_{\max}=L_s$;

S602、令 $m=1$;

S603、对第 m 组接收比特进行SCL译码,得到 L 条路径的译码结果;

S604、对 L 条路径的译码结果分别进行CRC校验,得到与各条路径的译码结果对应的CRC校验结果,其中,对第 m 组接收比特的 L 条路径的译码结果进行CRC校验时,将各条路径的译码结果与已经完成译码的第 $1\sim m-1$ 组接收比特的译码结果一起进行CRC校验,得到与 L 条路径的译码结果分别对应的CRC校验结果;

S605、根据各CRC校验结果,判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验;若是,则执行S606,否则执行S610;

S606、将通过校验的译码结果中最优的译码结果作为第 m 组接收比特对应的最终译码结果;

S607、判断 m 是否等于 M ,若是,则执行S609,否则执行S608;

S608、令 $m=m+1$,并执行S603;

S609、将 M 组接收比特对应的最终译码结果与冻结比特进行组合并输出;

S610、判断 L 是否等于 L_{\max} ,若是,则执行S611,否则执行S612;

S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第 m 组接收比特对应的最终译码结果,并执行S607;

S612、令 $L=2L$,并执行S602。

12. 一种译码设备,其特征在于,包括:分组器、SCL译码器、 M 个CRC校验器、控制器以及存储器;其中,所述分组器与所述SCL译码器连接,所述SCL译码器的第 m 路输出与第 $m+1$ 至第 M 个CRC校验器连接,所述存储器与所述控制器和所述SCL译码器,所述控制器与所述 M 个CRC校验器和所述SCL译码器连接,其中 M 为大于等于2的整数;

所述分组器,用于接收码字,所述码字包含接收比特和冻结比特;从所述码字中提取接收比特,并对所述接收比特按照在码字中的位置顺序划分成 M 组接收比特,其中,各组接收比特均包含CRC校验比特;

所述SCL译码器,用于对所述 M 组接收比特分别进行SCL译码处理,并将译码处理结果对应发送给CRC校验器进行CRC校验;

所述 M 个CRC校验器,用于分别对所述 M 组接收比特的译码结果进行CRC校验得到校验结果;

所述控制器,用于根据所述校验结果确定各组接收比特的最终译码结果,并将最终译码结果存储到所述存储器中,并且在得到全部接收比特的译码结果之后将 M 组接收比特对应的最终译码结果与冻结比特进行组合并输出;

所述存储器,用于存储控制器确定的各组接收比特的最终译码结果,并将各组接收比特的最终译码结果反馈给所述SCL译码器;

其中,所述SCL译码处理和所述CRC校验处理,包括:对第 $m-1$ 组接收比特进行 L 条路径的SCL译码处理,并对各 L 条路径的译码结果分别与第1至第 $m-2$ 组接收比特的最终译码结果一同进行CRC校验,若 L 条路径的译码结果中存在能够通过CRC校验的路径,则开始对第 m 组接收比特进行SCL译码处理;否则对 L 翻倍,并从第1组开始重新进行SCL译码处理,直到 L 达到路径数上限 L_{\max} 且 m 达到 M 。

13. 根据权利要求12所述的设备, 其特征在于, 所述SCL译码处理和所述CRC校验处理, 具体包括:

S601、初始化处理, 其中, $L=1$, $L_{\max}=L_S$;

S602、令 $m=1$;

S603、对第 m 组接收比特进行SCL译码, 得到 L 条路径的译码结果;

S604、对 L 条路径的译码结果分别进行CRC校验, 得到与各条路径的译码结果对应的CRC校验结果, 其中, 对第 m 组接收比特的 L 条路径的译码结果进行CRC校验时, 将各条路径的译码结果与已经完成译码的第 $1 \sim m-1$ 组接收比特的译码结果一起进行CRC校验, 得到与 L 条路径的译码结果分别对应的CRC校验结果;

S605、根据各CRC校验结果, 判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验; 若是, 则执行S606, 否则执行S610;

S606、将通过校验的译码结果中最优的译码结果作为第 m 组接收比特对应的最终译码结果;

S607、判断 m 是否等于 M , 若是, 则执行S609, 否则执行S608;

S608、令 $m=m+1$, 并执行S603;

S609、将 M 组接收比特对应的最终译码结果与冻结比特进行组合并输出;

S610、判断 L 是否等于 L_{\max} , 若是, 则执行S611, 否则执行S612;

S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第 m 组接收比特对应的最终译码结果, 并执行S607;

S612、令 $L=2L$, 并执行S602。

极化码的编译码方法及其装置

技术领域

[0001] 本发明涉及通信技术,尤其涉及一种极化码的编译码方法及其装置。

背景技术

[0002] 前向纠错 (Forward Error Correction,以下简称:FEC) 技术是通信系统的一个关键技术,可以通过牺牲一定的传输带宽来大幅提升系统性能。

[0003] 极化码(以下简称:Polar码)是FEC技术的一种,是由Erdal Arikan于2007年提出的一种信道编码方法,在二进制离散无记忆信道(Binary Discrete Memoryless Channel,以下简称:BDMC)下,这种编码方法理论上可以达到香农极限,并且具有较低的编译码复杂度。连续删除(Successive Cancellation,以下简称:SC)译码算法是针对Polar码最常见的译码算法。但针对中长码,SC译码性能并不理想。为解决这一问题,现有技术在SC译码算法基础上,提出了序列连续删除(SC List,以下简称:SCL)+循环冗余校验码(Cyclic Redundancy Check,以下简称:CRC)译码算法。SCL+CRC算法是在每一次的SC译码之后,进行路径分裂,允许有 L_{max} 条译码结果路径,从这 L_{max} 条译码结果路径中选出能够通过CRC校验且概率乘积最大的一条路径上的译码结果作为译码输出,从而显著提高译码准确度。

[0004] 但是,现有的SCL+CRC译码算法,其译码速度慢,延时大,无法满足高效的处理需求。

发明内容

[0005] 本发明实施例提供一种极化码的编译码方法及其装置,以提高译码速度。

[0006] 第一方面,本发明实施例提供一种极化码的编码方法,包括:

[0007] 将信息比特按照在码字中的位置顺序划分成M组信息比特,其中M为大于等于2的整数;

[0008] 对M组信息比特分别附加循环冗余校验码CRC校验比特,得到待发送信息比特,其中,第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的,第m组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第m-1组信息比特以及第m组信息比特生成的, $2 \leq m \leq M$;

[0009] 对所述待发送信息比特和冻结比特进行极化编码,得到码字并发送。

[0010] 可选的,所述将信息比特按照在码字中的位置顺序划分成M组信息比特,包括:

[0011] 将信息比特按照在码字中的位置顺序等分成M组,得到M组信息比特。

[0012] 可选的,所述对M组信息比特分别附加循环冗余校验码CRC校验比特,包括:

[0013] 将所述CRC校验比特分别附加在M组信息比特中各组信息比特的尾部。

[0014] 第二方面,本发明实施例提供一种极化码的译码方法,包括:

[0015] 接收码字,所述码字包含接收比特和冻结比特;

[0016] 从所述码字中提取接收比特,并对所述接收比特按照在码字中的位置顺序划分成M组接收比特,其中,各组接收比特均包含循环冗余校验码CRC校验比特,M为大于等于2的整

数;

[0017] 对所述M组接收比特进行SCL译码处理,并将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;其中,所述译码处理包括:对第m-1组接收比特进行L条路径的SCL译码处理,并对各L条路径的译码结果分别与第1至第m-2组接收比特的最终译码结果一同进行CRC校验,若L条路径的译码结果中存在能够通过CRC校验的路径,则开始对第m组接收比特进行SCL译码处理;否则对L翻倍,并从第1组开始重新进行SCL译码处理,直到L达到路径数上限 L_{\max} 且m达到M。

[0018] 可选的,所述对所述M组接收比特进行SCL译码处理,并将M组接收比特对应的最终译码结果与冻结比特进行组合并输出,包括:

[0019] S601、初始化处理,其中, $L=1$, $L_{\max}=L_s$;

[0020] S602、令 $m=1$;

[0021] S603、对第m组接收比特进行SCL译码,得到L条路径的译码结果;

[0022] S604、对L条路径的译码结果分别进行CRC校验,得到与各条路径的译码结果对应的CRC校验结果,其中,对第m组接收比特的L条路径的译码结果进行CRC校验时,将各条路径的译码结果与已经完成译码的第1~m-1组接收比特的译码结果一起进行CRC校验,得到与L条路径的译码结果分别对应的CRC校验结果;

[0023] S605、根据各CRC校验结果,判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验;若是,则执行S606,否则执行S610;

[0024] S606、将通过校验的译码结果中最优的译码结果作为第m组接收比特对应的最终译码结果;

[0025] S607、判断m是否等于M,若是,则执行S609,否则执行S608;

[0026] S608、令 $m=m+1$,并执行S603;

[0027] S609、将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;

[0028] S610、判断L是否等于 L_{\max} ,若是,则执行S611,否则执行S612;

[0029] S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第m组接收比特对应的最终译码结果,并执行S607;

[0030] S612、令 $L=2L$,并执行S602。

[0031] 可选的,所述对所述M组接收比特进行SCL译码处理之前,还包括:

[0032] 根据当前信息比特的接收速度和/或接收缓冲器的剩余空间,对所述 L_{\max} 进行调整。

[0033] 第三方面,本发明实施例提供一种编码设备,包括:

[0034] 分组模块,用于将信息比特按照在码字中的位置顺序划分成M组信息比特,其中M为大于等于2的整数;

[0035] 编码处理模块,用于对M组信息比特分别附加循环冗余校验码CRC校验比特,得到待发送信息比特,其中,第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的,第m组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第m-1组信息比特以及第m组信息比特生成的, $2 \leq m \leq M$;

[0036] 编码发送模块,用于对所述待发送信息比特和冻结比特进行极化编码,得到码字并发送。

[0037] 可选的,所述分组模块,具体用于将信息比特按照在码字中的位置顺序等分成M组,得到M组信息比特。

[0038] 第四方面,本发明实施例提供一种编码设备,包括:分组器、M个CRC生成器以及编码器;其中,分组器与所述M个CRC生成器的输入端连接,第m个CRC生成器的输出端与所述编码器和第m+1至第M个CRC生成器的输入端连接,第M个CRC生成器的输出端与所述编码器连接,其中M为大于等于2的整数;

[0039] 所述分组器,用于将信息比特按照在码字中的位置顺序划分成M组信息比特;

[0040] 所述M个CRC生成器,用于对M组信息比特分别附加循环冗余校验码CRC校验比特,得到待发送信息比特,其中,第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的,第m组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第m-1组信息比特以及第m组信息比特生成的, $2 \leq m \leq M$;

[0041] 所述编码器,用于对所述待发送信息比特和冻结比特进行极化编码,得到码字并发送。

[0042] 第五方面,本发明实施例提供一种译码设备,包括:

[0043] 分组模块,用于接收码字,所述码字包含接收比特和冻结比特;从所述码字中提取接收比特,并对所述接收比特按照在码字中的位置顺序划分成M组接收比特,其中,各组接收比特均包含循环冗余校验码CRC校验比特,M为大于等于2的整数;

[0044] 译码处理模块,用于对所述M组接收比特进行SCL译码处理,并将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;其中,所述译码处理包括:对第m-1组接收比特进行L条路径的SCL译码处理,并对各L条路径的译码结果分别与第1至第m-2组接收比特的最终译码结果一同进行CRC校验,若L条路径的译码结果中存在能够通过CRC校验的路径,则开始对第m组接收比特进行SCL译码处理;否则对L翻倍,并从第1组开始重新进行SCL译码处理,直到L达到路径数上限 L_{\max} 且m达到M。

[0045] 可选的,所述译码处理模块,具体用于:

[0046] S601、初始化处理,其中, $L=1$, $L_{\max}=L_s$;

[0047] S602、令 $m=1$;

[0048] S603、对第m组接收比特进行SCL译码,得到L条路径的译码结果;

[0049] S604、对L条路径的译码结果分别进行CRC校验,得到与各条路径的译码结果对应的CRC校验结果,其中,对第m组接收比特的L条路径的译码结果进行CRC校验时,将各条路径的译码结果与已经完成译码的第1~m-1组接收比特的译码结果一起进行CRC校验,得到与L条路径的译码结果分别对应的CRC校验结果;

[0050] S605、根据各CRC校验结果,判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验;若是,则执行S606,否则执行S610;

[0051] S606、将通过校验的译码结果中最优的译码结果作为第m组接收比特对应的最终译码结果;

[0052] S607、判断m是否等于M,若是,则执行S609,否则执行S608;

[0053] S608、令 $m=m+1$,并执行S603;

[0054] S609、将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;

[0055] S610、判断L是否等于 L_{\max} ,若是,则执行S611,否则执行S612;

[0056] S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第m组接收比特对应的最终译码结果,并执行S607;

[0057] S612、令 $L=2L$,并执行S602。

[0058] 第六方面,本发明实施例提供一种译码设备,包括:分组器、SCL译码器、M个CRC校验器、控制器以及存储器;其中,所述分组器与所述SCL译码器连接,所述SCL译码器的第m路输出与第m+1至第M个CRC校验器连接,所述存储器与所述控制器和所述SCL译码器,所述控制器与所述M个CRC校验器和所述SCL译码器连接,其中M为大于等于2的整数;

[0059] 所述分组器,用于接收码字,所述码字包含接收比特和冻结比特;从所述码字中提取接收比特,并对所述接收比特按照在码字中的位置顺序划分成M组接收比特,其中,各组接收比特均包含CRC校验比特;

[0060] 所述SCL译码器,用于对所述M组接收比特分别进行SCL译码处理,并将译码处理结果对应发送给CRC校验器进行CRC校验;

[0061] 所述M个CRC校验器,用于分别对所述M组接收比特的译码结果进行CRC校验得到校验结果;

[0062] 所述控制器,用于根据所述校验结果确定各组接收比特的最终译码结果,并将最终译码结果存储到所述存储器中,并且在得到全部接收比特的译码结果之后将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;

[0063] 所述存储器,用于存储控制器确定的各组接收比特的最终译码结果,并将各组接收比特的最终译码结果反馈给所述SCL校验器;

[0064] 其中,所述SCL译码处理和所述CRC校验处理,包括:对第m-1组接收比特进行L条路径的SCL译码处理,并对各L条路径的译码结果分别与第1至第m-2组接收比特的最终译码结果一同进行CRC校验,若L条路径的译码结果中存在能够通过CRC校验的路径,则开始对第m组接收比特进行SCL译码处理;否则对L翻倍,并从第1组开始重新进行SCL译码处理,直到L达到路径数上限 L_{\max} 且m达到M。

[0065] 可选的,所述SCL译码处理和所述CRC校验处理,具体包括:

[0066] S601、初始化处理,其中, $L=1$, $L_{\max}=L_s$;

[0067] S602、令 $m=1$;

[0068] S603、对第m组接收比特进行SCL译码,得到L条路径的译码结果;

[0069] S604、对L条路径的译码结果分别进行CRC校验,得到与各条路径的译码结果对应的CRC校验结果,其中,对第m组接收比特的L条路径的译码结果进行CRC校验时,将各条路径的译码结果与已经完成译码的第1~m-1组接收比特的译码结果一起进行CRC校验,得到与L条路径的译码结果分别对应的CRC校验结果;

[0070] S605、根据各CRC校验结果,判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验;若是,则执行S606,否则执行S610;

[0071] S606、将通过校验的译码结果中最优的译码结果作为第m组接收比特对应的最终译码结果;

[0072] S607、判断m是否等于M,若是,则执行S609,否则执行S608;

[0073] S608、令 $m=m+1$,并执行S603;

[0074] S609、将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;

- [0075] S610、判断L是否等于 L_{\max} ，若是，则执行S611，否则执行S612；
- [0076] S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第m组接收比特对应的最终译码结果，并执行S607；
- [0077] S612、令 $L=2L$ ，并执行S602。
- [0078] 本发明实施例通过在编码端对待发送的信息比特进行分段编码，在译码端对接收到的信息比特进行分段译码，且在SCL译码过程中，分裂路径自适应调整，相对于现有技术来说，其译码时间显著减小，译码速度快，延时低，可以提高2-M倍的译码速度。而且，本发明实施例在编码端对第m组信息比特生成CRC校验比特时，将第1~m-1组信息比特均引入一同进行CRC计算，保证了分段编码中各组信息比特的关联性，在译码端对第m组译码结果生成CRC校验比特时，同样将已经译码得到的第1~m-1组译码结果均引入一同进行CRC计算，保证了译码结果的关联性和准确性。另外，本发明实施例还可以动态调整路径数量上限 L_{\max} ，避免了译码设备闲置或浪费，合理分配系统资源；此外，本发明实施例的译码能力与传统的SCL+CRC器译码器相当，译码能力无损。

附图说明

- [0079] 图1为现有极化码的SCL+CRC的编码处理示意图；
- [0080] 图2为本发明极化码的编码方法实施例的流程图；
- [0081] 图3为图2所示方法实施例所采用的编码设备的结构示意图；
- [0082] 图4为 $N=8$ 的Polar码路径分裂过程的示意图；
- [0083] 图5为本发明中极化码的译码设备的结构示意图；
- [0084] 图6为本发明极化码的译码方法实施例的流程图；
- [0085] 图7为图5所示译码设备中控制器的处理状态迁移示意图；
- [0086] 图8为本发明实施例的译码处理过程和现有SCL译码处理过程的对比示意图；
- [0087] 图9为本发明编码设备实施例的结构示意图；
- [0088] 图10为本发明译码设备实施例的结构示意图。

具体实施方式

[0089] 本发明实施例针对Polar码的编译码处理。Polar码是基于信道极化理论完成的，将信道极化以后，一部分信道趋于无噪信道，另外一部分信道趋于全噪信道。基于这个现象，Polar码在进行编码时，可以将要传送的信息比特放在无噪信道上传输，而在全噪信道上传输冻结比特。因此，当码长 N 趋于无穷大时，系统容量可以达到香农极限。

[0090] Polar码的编码又称为 G_N 陪集编码。Polar码的码字长度 N 被严格定义为2的幂次方，即对于任意 $n \geq 0$ 都有 $N=2^n$ 。Polar码是线性分组码，其编码公式为： $y_1^N = x_1^N G_N$ 式中，向量 x_1^N 是所需传送的信息比特， G_N 是 N 阶次的生成矩阵， y_1^N 是编码后的码字。

[0091] 图1为现有极化码的SCL+CRC的编码处理示意图，如图1所示，以Polar码长2048，码率 $R=0.5$ 为例，其信息比特为1024bits，经过16bits的CRC校验编码以后，与1008bits的冻结比特一起，输到Polar码(2048,1024)编码器进行编码，输出编码后的2048比特。在译码端，SCL译码器首先会对接收到的这2048个比特进行SC译码处理，输出 L 条路径，然后分别对这 L 条路径进行CRC校验，选满足CRC校验，且概率积最大的路径做为译码输出。

[0092] 由图1可以看出,SCL+CRC算法是在对完整的信息比特进行一次SC译码之后,进行路径分裂,并在已经找到能够通过CRC校验的路径或者路径数达到所设定上限 L_{\max} 时结束译码,也即允许有 L_{\max} 条路径进行后续的操作,从中选出能够通过CRC校验,并且概率乘积最大的一条路径,作为译码输出。因此,对于一次SC译码来说,该算法需要基于全部信息比特进行路径分裂并判断各路径中是否存在通过CRC校验的路径,若不存在,则需要增大输出的路径数 L ,再进行一次SC译码过程,直到路径数量 L 达到最大值 L_{\max} 。该过程处理复杂,耗时较长。

[0093] 图2为本发明极化码的编码方法实施例的流程图,图3为图2所示方法实施例所采用的编码设备的结构示意图,如图2和3所示,本实施例的编码方法包括:

[0094] S201、将信息比特按照在码字中的位置顺序划分成 M 组信息比特;其中 M 为大于等于2的整数;

[0095] S202、对 M 组信息比特分别附加循环冗余校验码CRC校验比特,得到待发送信息比特,其中,第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的,第 m 组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第 $m-1$ 组信息比特以及第 m 组信息比特生成的, $2 \leq m \leq M$;

[0096] S203、对所述待发送信息比特和冻结比特进行极化编码,得到码字并发送。

[0097] 结合图3具体来说,编码设备例如可以包括 M 个CRC生成器和一个Polar编码器,Polar编码器最终要编码得到 N 位比特,其中包含 K 位信息比特和 $N-K$ 位冻结比特。冻结比特对于译码端来说,其发送内容以及在码字中的位置均是已知的。针对 K 位信息比特来说,编码设备需要对原始信息比特附加CRC校验比特来形成。具体来说,编码设备可以按照原始信息比特在码字中的位置顺序将原始信息比特划分成 M 组,每一组的原始信息比特为 n_1, n_2, \dots, n_M 位,第一组原始信息比特为 x_1, x_2, \dots, x_{n_1} ,经过CRC生成器1校验以后变成 x_1, x_2, \dots, x_{k_1} ,第二组原始信息比特为 $x_{k_1+1}, x_{k_1+2}, \dots, x_{k_1+n_2}$,第二组原始信息比特与第一组的CRC校验输出 x_1, x_2, \dots, x_{k_1} 共同作为CRC生成器2的输入,CRC校验后的比特为 $x_{k_1+1}, x_{k_1+2}, \dots, x_{k_2}$ 。以此类推,第 m 组($1 \leq m \leq M$)包含 n_m 个原始信息比特和 r_m 个校验比特,其末位比特的位置为 k_m ,所以 $k_M = K$ 。

[0098] 由上述过程可知,第 m 组的校验比特为:

[0099] $x_{k_m-r_m+1}, x_{k_m-r_m+2}, \dots, x_{k_m} = \text{CRC}(x_1, x_2, \dots, x_{k_{m-1}}, x_{k_{m-1}+1}, \dots, x_{k_{m-1}+n_m})$ 。

[0100] 基于所需生成的校验比特位数,本领域技术人员可以对 M 个CRC生成器进行设计,此处不再赘述。

[0101] 然后,Polar编码器可以采用 G_N 陪集编码对 N 位比特 x_1^N 进行编码,最终形成 N 比特的码字 X_1^N 。

[0102] 需要说明的是,在上述编码过程中, M 组信息比特中,各组信息比特的长度可以是不相同的,优选的,各组信息比特的也可以是将全部信息比特按照位置顺序等分成 M 组后得到的各组等长的信息比特。

[0103] 另外,在对 M 组信息比特分别附加CRC校验比特时,可以优选地将CRC校验比特分别附加在 M 组信息比特中各组信息比特的尾部。

[0104] 由上述编码过程可知,本发明实施例的编码过程是待发送的原始信息比特进行

分组编码,并且,在生成当前组信息比特的CRC校验比特时,需要参考其之前各组的信息比特以及校验比特,从而在分组的过程中,不会对各组的信息比特进行隔离,保证了信息比特之间的相关性。

[0105] 下面详细介绍,在上述编码过程的基础上,译码端的具体译码过程。在对该译码过程进行说明之前,首先对SCL译码算法进行说明。发送信号 X_1^N 在经过信道传输后,译码端从信道上接收的信号为 y_1^N 。针对该 y_1^N 可以进行如下译码处理过程:

[0106] 第一步:似然比计算。

[0107] 采用公式(1)进行似然比计算,得到信道上各接收信号 y_i 的似然比 $L_N^{(i)}(y_1^N, \hat{x}_1^{i-1})$ 。

$$[0108] \quad L_N^{(i)}(y_1^N, \hat{x}_1^{i-1}) = e^{2y_i/\sigma^2} \quad (1)$$

[0109] 其中, \hat{x}_1^{i-1} 表征对前 $1 \sim i-1$ 个信号 x_1^{i-1} 的估计结果, σ 为与信道的信噪比相关的参数。

[0110] 第二步:路径分裂。

[0111] 图4为 $N=8$ 的Polar码路径分裂过程的示意图,如图4所示,其中, $\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_5$ 是冻结比特的估计结果, $\hat{x}_4, \hat{x}_6, \hat{x}_7, \hat{x}_8$ 是信息比特的估计结果。

[0112] 对于信息比特 $\hat{x}_4, \hat{x}_6, \hat{x}_7, \hat{x}_8$ 来说,可以将当前的译码路径分裂成两条,即 $\hat{x}_i=0$ 和 $\hat{x}_i=1$;对于冻结比特 $\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_5$ 来说,由于其对于译码端来说是已知的,因此,不进行路径分裂,保持路径数不变。

[0113] 在具体的分裂过程中,可以预先设定幸存路径条数 L 的值。例如,对图4所给出的4位信息比特的Polar码路径分裂来说,最多可分裂得到 $2^4=16$ 条路径,路径分裂到 \hat{x}_6 时,将得到4条分裂路径;路径分裂到 \hat{x}_7 时,将得到8条路径;路径分裂到 \hat{x}_8 时,将得到16条路径。

[0114] 第三步:计算分裂得到的路径上各节点为0和为1的概率。

[0115] 该节点为0和为1的概率分别采用下述公式(2)和公式(3)计算得到:

$$[0116] \quad p(\hat{x}_i=0 | \hat{x}_1^{i-1}) = 1 - \frac{1}{L_N^{(i)}(y_1^N, \hat{x}_1^{i-1}) + 1} \quad (2)$$

$$[0117] \quad p(\hat{x}_i=1 | \hat{x}_1^{i-1}) = \frac{1}{L_N^{(i)}(y_1^N, \hat{x}_1^{i-1}) + 1} \quad (3)$$

[0118] 式中的 $L_N^{(i)}(y_1^N, \hat{x}_1^{i-1})$ 可由公式(1)求出。

[0119] 第四步:确定当前分裂得到的路径数是否超过预设幸存路径数 L ,若未超过,则执行第二步和第三步,继续进行路径分裂和各节点概率计算,若超过,则计算当前分裂得到的各路径的概率乘积,并从中选择 L 条概率乘积最大的路径,并基于所选择的路径继续执行第二步和第三步,直到完成全部节点的路径分裂过程。

[0120] 以 $L=4$ 举例来说, $\hat{x}_1, \hat{x}_2, \hat{x}_3$ 均为冻结比特,无需路径分裂且对于译码端来说其均为已知,因此这3个节点的概率均为1,对于 \hat{x}_4 来说,其为信息比特,分裂成2条路径,可以采用公式(2)和(3)分别计算 $\hat{x}_4=0$ 和 $\hat{x}_4=1$ 的概率,对于 \hat{x}_5 来说,其为冻结比特无需路径分裂且对于译码端来说其均为已知,因此该节点的概率为1,且路径依然为2条,未超过 L ,可以继续分裂。对于 \hat{x}_6 来说,其为信息比特,继续分裂得到4条路径,可以采用公式(2)和(3)分别计算 $\hat{x}_6=0$ 和 $\hat{x}_6=1$ 的概率,此时路径为4条,未超过 L ,可以继续分裂。对 \hat{x}_7 来说,其为信息比特,继续分裂得到8条路径,可以采用公式(2)和(3)分别计算 $\hat{x}_7=0$ 和 $\hat{x}_7=1$ 的概率,而且,因为此

时的路径数超过了L,需要通过计算这8条路径的概率乘积。概率乘积具体可以采用公式(4)计算得到:

$$[0121] \quad P(\hat{x}_1^i) = P(\hat{x}_1^{i-1}) p(\hat{x}_1 = b | \hat{x}_1^{i-1}) \quad (4)$$

[0122] 其中, $b \in \{0, 1\}$ 。

[0123] 然后,可以从这8条路径中选择出概率乘积最大的4条路径,并丢弃剩余的4条路径。在所选择的这4条路径的基础上,对信息比特 \hat{x}_7 继续进行路径分裂,依然得到8条路径,可以采用公式(2)和(3)分别计算 $\hat{x}_8 = 0$ 和 $\hat{x}_8 = 1$ 的概率,并且需要采用公式(4)计算这8条路径的概率乘积,进而从这8条路径中选择出概率乘积最大的4条路径,并丢弃剩余的4条路径。至此,即完成了全部节点的路径分裂,并且得到了L条幸存路径。

[0124] 第五步,从这L条幸存路径中选出概率乘积 $P(\hat{x}_1^i)$ 最大的那条路径上的译码结果做为最终的译码输出。

[0125] 而对于SCL+CRC译码算法来说,第五步是对L条幸存路径上的译码结果均进行CRC校验,将CRC校验通过的译码结果作为最终的译码输出,或者在有多条路径均通过CRC校验时,将CRC校验通过的路径中概率乘积最大的译码结果作为最终的译码输出。

[0126] 图5为本发明中极化码的译码设备的结构示意图,如图5所示,该译码设备可以包括SCL译码器、M个CRC校验器、控制器以及存储器。其中,SCL译码器用于对M组接收比特分别进行SCL译码,M个CRC校验器分别与SCL译码器中各组译码结果对应,对每组的译码结果分别进行CRC校验,控制器负责整个译码过程的集中控制,在确定存在CRC校验通过的译码结果时,可以将该译码结果存入存储器,在完成各组译码之后,可以根据存储器中存储的各组译码结果以及冻结码字,生成最终的完整译码结果并进行输出。

[0127] 译码设备可以预先获知接收的码字上哪些位置传输的是信息比特,哪些位置传输的是冻结比特,也可以预先获知编码设备的CRC校验比特的传输位置,而且还可以预先获知编码设备对原始信息比特进行M组的分组划分方式。因此,译码设备可以按照编码设备的分组划分方式,对接收到的N个比特位中除N-K个冻结比特之外的K个接收比特进行分组,得到M组接收比特,每组接收比特中包含原始信息比特对应的接收比特以及相应的CRC校验比特。即如图5中所示,译码设备划分得到的每一组接收比特为 K_1, K_2, \dots, K_M 位,第1组接收比特为 y_1, y_2, \dots, y_{k_1} ,经过SCL译码以后变成 $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{k_1}$,第2组接收比特为 $y_{k_1+1}, y_{k_1+2}, \dots, y_{k_2}$,经过SCL译码以后变成 $\hat{x}_{k_1+1}, \hat{x}_{k_1+2}, \dots, \hat{x}_{k_2}$ 。以此类推,第m组接收比特为 $y_{k_{m-1}+1}, y_{k_{m-1}+2}, \dots, y_{k_m}$,经过SCL译码以后变成 $\hat{x}_{k_{m-1}+1}, \hat{x}_{k_{m-1}+2}, \dots, \hat{x}_{k_m}$,第M组接收比特为 $y_{k_{M-1}+1}, y_{k_{M-1}+2}, \dots, y_{k_M}$,经过SCL译码以后变成 $\hat{x}_{k_{M-1}+1}, \hat{x}_{k_{M-1}+2}, \dots, \hat{x}_{k_M}$ 。其中 K_m 为第m组包含的比特数, $K_1 = k_1, K_M = k_M - k_{M-1}$ 。

[0128] 图6为本发明极化码的译码方法实施例的流程图,图7为图5所示译码设备中控制器的处理状态迁移示意图,结合图5~图7所示,本实施例的译码方法在接收到包含接收比特和冻结比特的码字后,可以从码字中提取接收比特,并对接收比特按照在码字中的位置顺序划分成M组接收比特,其中,各组接收比特均包含图2所示编码设备所生成的CRC校验比特,M为大于等于2的整数;然后译码设备可以对M组接收比特进行SCL译码处理,其中,对第m-1组接收比特进行L条路径的SCL译码处理,并对各L条路径的译码结果分别与第1至第m-2组接收比特的最终译码结果一同进行CRC校验,若L条路径的译码结果中存在能够通过CRC

校验的路径,则开始对第m组接收比特进行SCL译码处理;否则对L翻倍,并从第1组开始重新进行SCL译码处理,直到L达到路径数上限 L_{\max} 且m达到M。

[0129] 具体的,本实施例的译码方法可以包括:

[0130] S601、初始化处理,其中, $L=1, L_{\max}=L_s$ 。

[0131] S602、令 $m=1$ 。

[0132] 具体来说,译码处理过程均是从对第1组接收比特进行1条路径分裂开始的,路径数上限可以设定为 L_s 。

[0133] 优选的, L_{\max} 的取值 L_s 可以动态地调整,例如在对一个码字译码开始前,可以根据当前信息比特的接收速度和/或缓冲器的剩余空间动态地调整。在一个码字译码结束,下一个码字译码开始之前,根据当前输入缓冲器的剩余空间可以计算出下一个码字译码时所允许的路径数上限 L_{\max} ,并存入控制器。假设D为缓冲器剩余空间数,t为相邻码字信道信息输入的间隔时间,那么所允许的路径数上限 L_{\max} 应满足条件使译码过程中缓冲器不会溢出。根据前面的译码过程分析,译码过程理论上可能占用的最长时间与传统自适应SCL相同,即 $T_{\max}=(2L_0-1)KT_0$ 。其中, L_0 为路径分裂个数,K为待译码的比特数, T_0 为分裂一个比特所需的时间。因此,为了保证缓冲器不会溢出,要求 $T_{\max}<Dt$ 。所以 L_{\max} 为满足该条件的最大值即:

$$[0134] \quad L_{\max} = 2^{\left\lceil \log_2 \left(\frac{Dt}{2KT_0} + \frac{1}{2} \right) \right\rceil}$$

[0135] S603、对第m组接收比特进行SCL译码,得到L条路径的译码结果。

[0136] S604、对L条路径的译码结果分别进行CRC校验,得到与各条路径的译码结果对应的CRC校验结果。

[0137] 其中,与图2所示编码过程对应的,对第m组接收比特的L条路径的译码结果进行CRC校验时,需要将各条路径的译码结果与已经完成译码的第1~m-1组接收比特的译码结果一起进行CRC校验,得到与L条路径的译码结果分别对应的CRC校验结果。

[0138] S605、根据各CRC校验结果,判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验;若是,则执行S606,否则执行S610;

[0139] S606、将通过校验的译码结果中最优的译码结果作为第m组接收比特对应的最终译码结果;

[0140] S607、判断m是否等于M,若是,则执行S609,否则执行S608。

[0141] S608、令 $m=m+1$,并执行S603。

[0142] S609、将M组接收比特对应的最终译码结果与冻结比特进行组合并输出。

[0143] S610、判断L是否等于 L_{\max} ,若是,则执行S611,否则执行S612;

[0144] S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第m组接收比特对应的最终译码结果,并执行S607;

[0145] S612、令 $L=2L$,并执行S602。

[0146] 结合图5和图7具体来说,译码过程是对M组接收比特按照从1至M的顺序依次进行SCL译码。对于每组接收比特的SCL译码过程,其可以参考前述已说明的SCL译码过程。

[0147] 控制器包括 $ML_{\max}+1$ 个状态的状态机,除了最后一个状态外,每个状态取决于两个参数m和L,其中状态(L,m)持续 $K_m L$ 个时钟。

[0148] 从(L=1,m=1)这一初始状态开始,SCL译码器可以对第1组接收比特 y_1, y_2, \dots, y_k

进行路径数 L 为1的译码处理,因此,SCL译码算法可以经过路径分裂得到1条路径,该条路径上的译码结果 $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{k_1}$ 被送入CRC校验器1进行CRC校验,得到该译码结果的CRC校验结果CRC#1,该CRC校验结果CRC#1被送入控制器中判决是通过校验还是没通过校验,例如CRC#1=1则表征通过校验,CRC#1=0则表征未通过校验。

[0149] 若CRC#1=1校验通过,则控制器可以将该路径的译码结果输出给存储器,存储器即可存储该第1组接收比特的最终译码结果。在此基础上,控制器可以将状态迁移至($L=1, m=2$),即开始对第2组接收比特进行SCL译码处理。在对第2组接收比特进行译码处理时,存储器可以将第1组接收比特的最终译码结果再反馈给SCL译码器,从而使得SCL译码器可以将第1组接收比特的最终译码结果作为对第2组接收比特的译码结果进行CRC校验时的参考。具体来说,译码设备可以对第2组接收比特进行SCL译码。也即SCL译码器可以对第2组接收比特进行路径分裂得到1条路径,该条路径上的译码结果 $\hat{x}_{k_1+1}, \hat{x}_{k_1+2}, \dots, \hat{x}_{k_2}$ 以及第1组接收比特的最终译码结果 $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{k_1}$ 均被送入CRC校验器2进行CRC校验,得到与该路径的译码结果 $\hat{x}_{k_1+1}, \hat{x}_{k_1+2}, \dots, \hat{x}_{k_2}$ 对应的CRC校验结果CRC#2,该CRC校验结果CRC#2被送入控制器中判决是通过校验还是没通过校验,例如CRC#2=1则表征通过校验,CRC#2=0则表征未通过校验。若第2组接收比特 $y_{k_1+1}, y_{k_1+2}, \dots, y_{k_2}$ 进行路径数 L 为1的SCL译码处理也通过了CRC校验,则控制器可以将该路径的译码结果输出给存储器,存储器即可存储该第2组接收比特的最终译码结果,控制器可以将状态迁移至($L=1, m=3$),即开始对第3组接收比特进行SCL译码处理。在对第3组接收比特进行译码处理时,SCL译码器可以获得第1组接收比特的最终译码结果和第2组接收比特的最终译码结果,从而使得SCL译码器可以将第1组接收比特的最终译码结果和第2组接收比特的最终译码结果作为对第3组接收比特的译码结果进行CRC校验时的参考。具体来说,译码设备可以对第3组接收比特进行SCL译码。也即SCL译码器可以对第3组接收比特进行路径分裂得到1条路径,该条路径上的译码结果 $\hat{x}_{k_2+1}, \hat{x}_{k_2+2}, \dots, \hat{x}_{k_3}$ 、第1组接收比特的最终译码结果 $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{k_1}$ 以及第2组接收比特的最终译码结果 $\hat{x}_{k_1+1}, \hat{x}_{k_1+2}, \dots, \hat{x}_{k_2}$ 均被送入CRC校验器3进行CRC校验,得到与该路径的译码结果 $\hat{x}_{k_2+1}, \hat{x}_{k_2+2}, \dots, \hat{x}_{k_3}$ 对应的CRC校验结果CRC#3,该CRC校验结果CRC#3被送入控制器中判决是通过校验还是没通过校验,例如CRC#3=1则表征通过校验,CRC#3=0则表征未通过校验,依次类推。最快的处理过程是,在 $L=1$ 的情况下,控制器可以一直将状态迁移至($L=1, m=M$),即开始对第 M 组接收比特 $y_{k_{M-1}+1}, y_{k_{M-1}+2}, \dots, y_{k_M}$ 进行路径数 L 为1的SCL译码处理,若该路径的译码结果也通过了CRC校验,则可以得到全部信息比特的最终译码结果。

[0150] 若CRC#1=0校验不通过,且此时路径数 L 未达到路径数上限 L_{\max} ,则控制器可以将路径数 L 翻倍,即调整为 $L=2$,此时控制器的状态迁移至($L=2, m=1$)。控制器因此可以控制SCL译码器重新对第1组接收比特 y_1, y_2, \dots, y_{k_1} 进行路径分裂,得到2条路径,这2条路径的译码结果 $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{k_1}$ 均被送入CRC校验器1进行CRC校验,分别得到2条路径的译码结果的CRC校验结果,该2个CRC校验结果被送入控制器中判决是通过校验还是没通过校验。

[0151] 针对状态($L=2, m=1$),若2条路径中有至少1条路径的译码结果校验通过,则控制器可以将该校验通过的译码结果中的最优译码结果输出给存储器,存储器即可存储该第1组接收比特的最终译码结果。需要说明的是,若只有1条路径的译码结果通过校验,则该条路径的译码结果就是最优译码结果,若2条路径的译码结果均通过校验,则控制器可以从这

2条路径的译码结果中选出各节点概率乘积最高的译码结果作为最优译码结果。之后,控制器的状态即可迁移至($L=2, m=2$),从而可以对第2组接收比特进行路径数为2的SCL译码处理。在得到2条路径上的译码结果 $\hat{x}_{k_1+1}, \hat{x}_{k_1+2}, \dots, \hat{x}_{k_2}$ 之后,这2条路径上的译码结果可以分别与第1组接收比特的最终译码结果 $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{k_1}$ 一同被送入CRC校验器2进行CRC校验,得到与2条路径的译码结果 $\hat{x}_{k_1+1}, \hat{x}_{k_1+2}, \dots, \hat{x}_{k_2}$ 分别对应的CRC校验结果CRC#2,该2条路径的CRC校验结果CRC#2被送入控制器中判决是通过校验还是没通过校验,例如CRC#2=1则表征通过校验,CRC#2=0则表征未通过校验。若第2组接收比特 $y_{k_1+1}, y_{k_1+2}, \dots, y_{k_2}$ 进行路径数L为2的译码结果中至少有1条路径的译码结果也通过了CRC校验,则控制器可以存储第2组接收比特的最终译码结果,并且可以将状态迁移至($L=2, m=3$),即开始对第3组接收比特进行SCL译码处理,依次类推,控制器可以将状态迁移至($L=2, m=M$),即开始对第M组接收比特 $y_{k_{M-1}+1}, y_{k_{M-1}+2}, \dots, y_{k_M}$ 进行路径数L为2的SCL译码处理,若也通过了CRC校验,就可以得到全部信息比特的最终译码结果。

[0152] 针对状态($L=2, m=1$),若2条路径的译码结果均未通过CRC校验且路径数L未达到路径数上限 L_{\max} ,则控制器可以将路径数L再翻倍,即调整为 $L=4$,此时控制器的状态迁移至($L=4, m=1$)。控制器因此可以控制SCL译码器重新对第1组接收比特 y_1, y_2, \dots, y_{k_1} 进行路径分裂,得到4条路径,这4条路径的译码结果 $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{k_1}$ 均被送入CRC校验器1进行CRC校验,分别得到4条路径的译码结果的CRC校验结果,后续的处理过程与前述对2条路径的处理过程是类似的,以此类推。

[0153] 综上,以 $L=L_0 < L_{\max}$ 为例来说:

[0154] 对于状态($L=L_0, m=1$),若这 L_0 条路径中至少有1条路径的译码结果通过了CRC校验,则控制器的状态从($L=L_0, m=1$)迁移至($L=L_0, m=2$),若 L_0 条路径的译码结果均未通过CRC校验,则控制器的状态从($L=L_0, m=2$)迁移至($L=2L_0, m=1$)从第1组接收比特重新开始SCL译码处理。

[0155] 对于状态($L=L_0, m=2$),若这 L_0 条路径中至少有1条路径的译码结果通过了CRC校验,则控制器的状态从($L=L_0, m=2$)迁移至($L=L_0, m=3$),若 L_0 条路径的译码结果均未通过CRC校验且 $L_0 \neq L_{\max}$,则控制器的状态从($L=L_0, m=3$)迁移至($L=2L_0, m=1$)。

[0156] 以此类推,最慢的处理过程是,控制器需要遍历全部状态,在 $L=L_{\max}, m=M$ 后,结束译码处理过程。

[0157] 需要说明的是, L_{\max} 可以是预设值,是根据系统对复杂度的需求以及对性能的需求,设置的幸存路径的门限值,一般情况下,在 $L \leq L_{\max}$ 时,均可以找到校验通过的最优路径。但也存在直到 $L=L_{\max}$ 时也无法找到校验通过路径的极端情况。对这种极端情况,本实施例可以从 L_{\max} 条分裂路径中选择各节点概率乘积最高的译码结果输出给存储器。

[0158] 下面再举例对上述译码过程所需时间进行说明。

[0159] 假设预先可以获知 $L=L_0$ 时才能够找到正确路径。那么在译码过程中:

[0160] $L=1 < L_0$ 且 $m=1$ 时CRC#1=0(不能通过校验),此时 $L \rightarrow 2$ 并重新从第 $m=1$ 组开始译码。用 T_0 表示分裂一个比特所需要的时间,那么此过程需要时间 $T(1,1)=K_1T_0$ 。

[0161] $L=2 < L_0$ 且 $m=1$ 时CRC#1=0(仍不能通过校验),此时 $L \rightarrow 4$ 并重新从第 $m=1$ 组开始译码。此过程需要时间 $T(2,1)=2K_1T_0$ 。

[0162] $L=L_0$ 且 $m=1$ 时 $CRC\#1=1$ (能通过校验), 此时 L 不变并开始第 $m=2$ 组译码。此过程需要时间 $T(L_0, 1)=L_0K_1T_0$ 。

[0163] $L=L_0$ 且 $m=2$ 时 $CRC\#2=1$ (能通过校验), 此时 L 不变并开始第 $m=3$ 组译码。此过程需要时间 $T(L_0, 2)=L_0K_2T_0$ 。

[0164] $L=L_0$ 且 $m=m$ 时 $CRC\#2=1$ (能通过校验), 此时 L 不变并开始第 $m=m+1$ 组译码。此过程需要时间 $T(L_0, m)=L_0K_mT_0$ 。

[0165] $L=L_0$ 且 $m=M$ 时 $CRC\#M=1$ (能通过校验), 此时结束译码, 可以输出译码结果。此过程需要时间 $T(L_0, M)=L_0K_MT_0$ 。

[0166] 整个译码过程所需总时间为:

$$[0167] \sum_{L=1}^{L_0} \sum_{m=1}^M T(L, m) = (L_0 - 1)K_1T_0 + L_0KT_0。$$

[0168] 若 M 组信息比特平均划分, 即 $K_i=K/M$, 那么译码总时间为:

$$[0169] T = (L_0 - 1)\frac{K}{M}T_0 + L_0KT_0。$$

[0170] 特殊情况, 在 $L < L_0$ 时, 有很小的可能 $CRC\#1=1$ (能通过校验), 那么 L 不变并对第2、3、4、...组译码, 直到第 m 组满足 $CRC\#m=0$ (不能通过校验), 此时 $L \rightarrow 2L$ 并重新从第1组开始译码, 此过程需要时间 $T(L, 0) + T(L, 1) + \dots + T(L, m) = Lk_mT_0$ 。

[0171] 综上, 图8为本发明实施例的译码处理过程和现有SCL译码处理过程的对比示意图, 如图8所示, 其中粗线条表征本发明实施例的译码处理过程中, 虚线条表征现有SCL译码处理过程。由该图8所示过程可知, 本发明实施例通过在编码端对待发送的信息比特进行分段编码, 在译码端对接收到的信息比特进行分段译码, 且在SCL译码过程中, 分裂路径自适应调整, 相对于现有技术来说, 其译码时间显著减小, 译码速度快, 延时低, 可以提高2-M倍的译码速度。而且, 本发明实施例在编码端对第 m 组信息比特生成CRC校验比特时, 将第1~ $m-1$ 组信息比特均引入一同进行CRC计算, 保证了分段编码中各组信息比特的关联性, 在译码端对第 m 组译码结果生成CRC校验比特时, 同样将已经译码得到的第1~ $m-1$ 组译码结果均引入一同进行CRC计算, 保证了译码结果的关联性和准确性。另外, 本发明实施例还可以动态调整路径数量上限 L_{max} , 避免了译码设备闲置或浪费, 合理分配系统资源; 此外, 本发明实施例的译码能力与传统的SCL+CRC器译码器相当, 译码能力无损。

[0172] 图9为本发明编码设备实施例的结构示意图, 如图9所示, 本实施例的编码设备可以包括:

[0173] 分组模块91, 用于将信息比特按照在码字中的位置顺序划分成 M 组信息比特, 其中 M 为大于等于2的整数;

[0174] 编码处理模块92, 用于对 M 组信息比特分别附加循环冗余校验码CRC校验比特, 得到待发送信息比特, 其中, 第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的, 第 m 组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第 $m-1$ 组信息比特以及第 m 组信息比特生成的, $2 \leq m \leq M$;

[0175] 编码发送模块93, 用于对所述待发送信息比特和冻结比特进行极化编码, 得到码字并发送。

[0176] 可选的, 分组模块91, 具体用于将信息比特按照在码字中的位置顺序等分成 M 组,

得到M组信息比特。

[0177] 本发明还提供一种编码设备的硬件结构实现,该硬件结构图如3所示,其包括:分组器、M个CRC生成器以及编码器;其中,分组器与M个CRC生成器的输入端连接,第m个CRC生成器的输出端与编码器和第m+1至第M个CRC生成器的输入端连接,第M个CRC生成器的输出端与所述编码器连接,其中M为大于等于2的整数;

[0178] 分组器,用于将信息比特按照在码字中的位置顺序划分成M组信息比特;

[0179] M个CRC生成器,用于对M组信息比特分别附加循环冗余校验码CRC校验比特,得到待发送信息比特,其中,第1组信息比特所附加的CRC校验比特是根据第1组信息比特生成的,第m组信息比特所附加的CRC校验比特是根据附加有CRC校验比特的第1组信息比特至第m-1组信息比特以及第m组信息比特生成的, $2 \leq m \leq M$;

[0180] 编码器,用于对所述待发送信息比特和冻结比特进行极化编码,得到码字并发送。

[0181] 本实施例的编码设备可以用于执行上述编码端所执行的操作,其原理和技术效果类似,此处不再赘述。

[0182] 图10为本发明译码设备实施例的结构示意图,如图10所示,本实施例的译码设备可以包括:

[0183] 分组模块10,用于接收码字,所述码字包含接收比特和冻结比特;从所述码字中提取接收比特,并对所述接收比特按照在码字中的位置顺序划分成M组接收比特,其中,各组接收比特均包含循环冗余校验码CRC校验比特,M为大于等于2的整数;

[0184] 译码处理模块11,用于对所述M组接收比特进行SCL译码处理,并将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;其中,所述译码处理包括:对第m-1组接收比特进行L条路径的SCL译码处理,并对各L条路径的译码结果分别与第1至第m-2组接收比特的最终译码结果一同进行CRC校验,若L条路径的译码结果中存在能够通过CRC校验的路径,则开始对第m组接收比特进行SCL译码处理;否则对L翻倍,并从第1组开始重新进行SCL译码处理,直到L达到路径数上限 L_{\max} 且m达到M。

[0185] 可选的,译码处理模块11,具体用于:

[0186] S601、初始化处理,其中, $L=1, L_{\max}=L_s$;

[0187] S602、令 $m=1$;

[0188] S603、对第m组接收比特进行SCL译码,得到L条路径的译码结果;

[0189] S604、对L条路径的译码结果分别进行CRC校验,得到与各条路径的译码结果对应的CRC校验结果,其中,对第m组接收比特的L条路径的译码结果进行CRC校验时,将各条路径的译码结果与已经完成译码的第1~m-1组接收比特的译码结果一起进行CRC校验,得到与L条路径的译码结果分别对应的CRC校验结果;

[0190] S605、根据各CRC校验结果,判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验;若是,则执行S606,否则执行S610;

[0191] S606、将通过校验的译码结果中最优的译码结果作为第m组接收比特对应的最终译码结果;

[0192] S607、判断m是否等于M,若是,则执行S609,否则执行S608;

[0193] S608、令 $m=m+1$,并执行S603;

[0194] S609、将M组接收比特对应的最终译码结果与冻结比特进行组合并输出;

- [0195] S610、判断 L 是否等于 L_{\max} ，若是，则执行S611，否则执行S612；
- [0196] S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第 m 组接收比特对应的最终译码结果，并执行S607；
- [0197] S612、令 $L=2L$ ，并执行S602。
- [0198] 另外，本发明还提供一种译码设备的硬件结构实现，该硬件结构图如5所示，该译码设备可以包括：分组器、SCL译码器、 M 个CRC校验器、控制器以及存储器；其中，分组器与SCL译码器连接，SCL译码器的第 m 路输出与第 $m+1$ 至第 M 个CRC校验器连接，存储器与控制器和SCL译码器，控制器与 M 个CRC校验器和SCL译码器连接，其中 M 为大于等于2的整数；
- [0199] 分组器，用于接收码字，所述码字包含接收比特和冻结比特；从所述码字中提取接收比特，并对所述接收比特按照在码字中的位置顺序划分成 M 组接收比特，其中，各组接收比特均包含CRC校验比特；
- [0200] SCL译码器，用于对所述 M 组接收比特分别进行SCL译码处理，并将译码处理结果对应发送给CRC校验器进行CRC校验；
- [0201] M 个CRC校验器，用于分别对所述 M 组接收比特的译码结果进行CRC校验得到校验结果；
- [0202] 控制器，用于根据所述校验结果确定各组接收比特的最终译码结果，并将最终译码结果存储到所述存储器中，并且在得到全部接收比特的译码结果之后将 M 组接收比特对应的最终译码结果与冻结比特进行组合并输出；
- [0203] 存储器，用于存储控制器确定的各组接收比特的最终译码结果，并将各组接收比特的最终译码结果反馈给所述SCL校验器；
- [0204] 其中，SCL译码处理和所述CRC校验处理，包括：对第 $m-1$ 组接收比特进行 L 条路径的SCL译码处理，并对各 L 条路径的译码结果分别与第1至第 $m-2$ 组接收比特的最终译码结果一同进行CRC校验，若 L 条路径的译码结果中存在能够通过CRC校验的路径，则开始对第 m 组接收比特进行SCL译码处理；否则对 L 翻倍，并从第1组开始重新进行SCL译码处理，直到 L 达到路径数上限 L_{\max} 且 m 达到 M 。
- [0205] 优选的，上述SCL译码处理和所述CRC校验处理，具体包括：
- [0206] S601、初始化处理，其中， $L=1$ ， $L_{\max}=L_s$ ；
- [0207] S602、令 $m=1$ ；
- [0208] S603、对第 m 组接收比特进行SCL译码，得到 L 条路径的译码结果；
- [0209] S604、对 L 条路径的译码结果分别进行CRC校验，得到与各条路径的译码结果对应的CRC校验结果，其中，对第 m 组接收比特的 L 条路径的译码结果进行CRC校验时，将各条路径的译码结果与已经完成译码的第1~ $m-1$ 组接收比特的译码结果一起进行CRC校验，得到与 L 条路径的译码结果分别对应的CRC校验结果；
- [0210] S605、根据各CRC校验结果，判断各条路径的译码结果中是否存在至少一条路径的译码结果通过校验；若是，则执行S606，否则执行S610；
- [0211] S606、将通过校验的译码结果中最优的译码结果作为第 m 组接收比特对应的最终译码结果；
- [0212] S607、判断 m 是否等于 M ，若是，则执行S609，否则执行S608；
- [0213] S608、令 $m=m+1$ ，并执行S603；

- [0214] S609、将M组接收比特对应的最终译码结果与冻结比特进行组合并输出；
- [0215] S610、判断L是否等于 L_{\max} ，若是，则执行S611，否则执行S612；
- [0216] S611、将各条路径的译码结果中节点概率乘积最大的路径的译码结果作为第m组接收比特对应的最终译码结果，并执行S607；
- [0217] S612、令 $L=2L$ ，并执行S602。
- [0218] 本实施例的译码设备可以用于执行上述译码端所执行的操作，其原理和技术效果类似，此处不再赘述。
- [0219] 本领域普通技术人员可以理解：实现上述方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成，前述的程序可以存储于一计算机可读取存储介质中，该程序在执行时，执行包括上述方法实施例的步骤；而前述的存储介质包括：ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。
- [0220] 最后应说明的是：以上各实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述各实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分或者全部技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的范围。

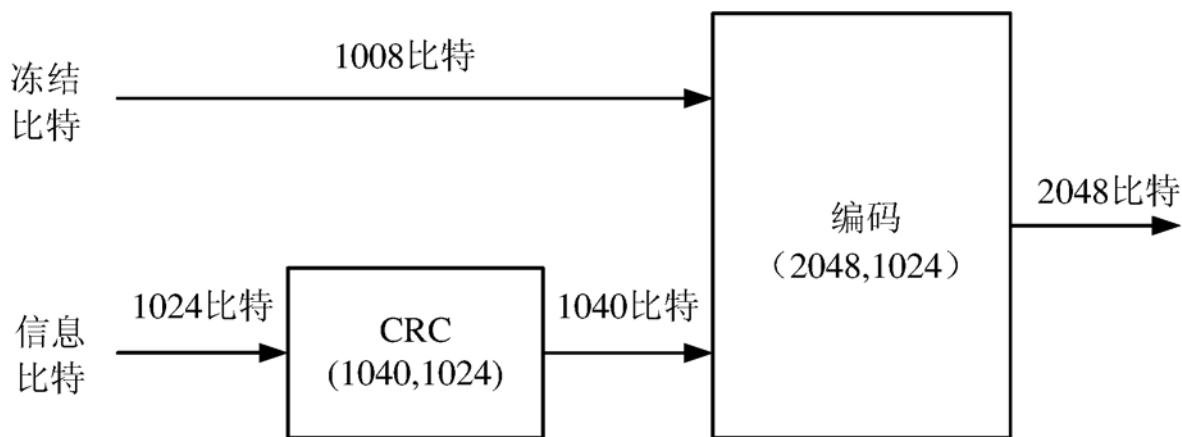


图1

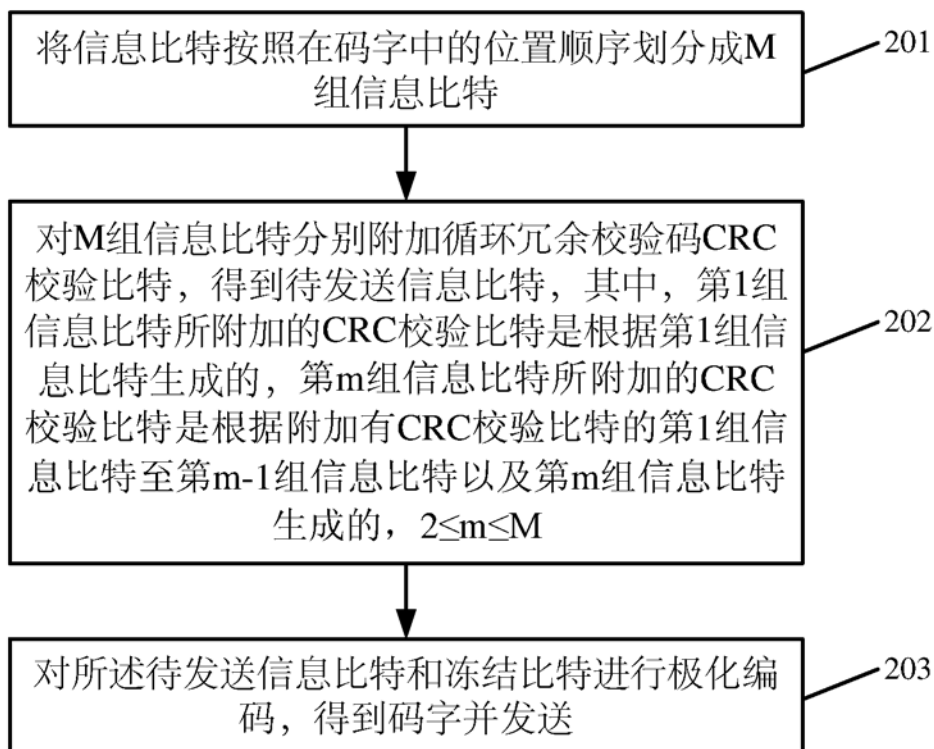


图2

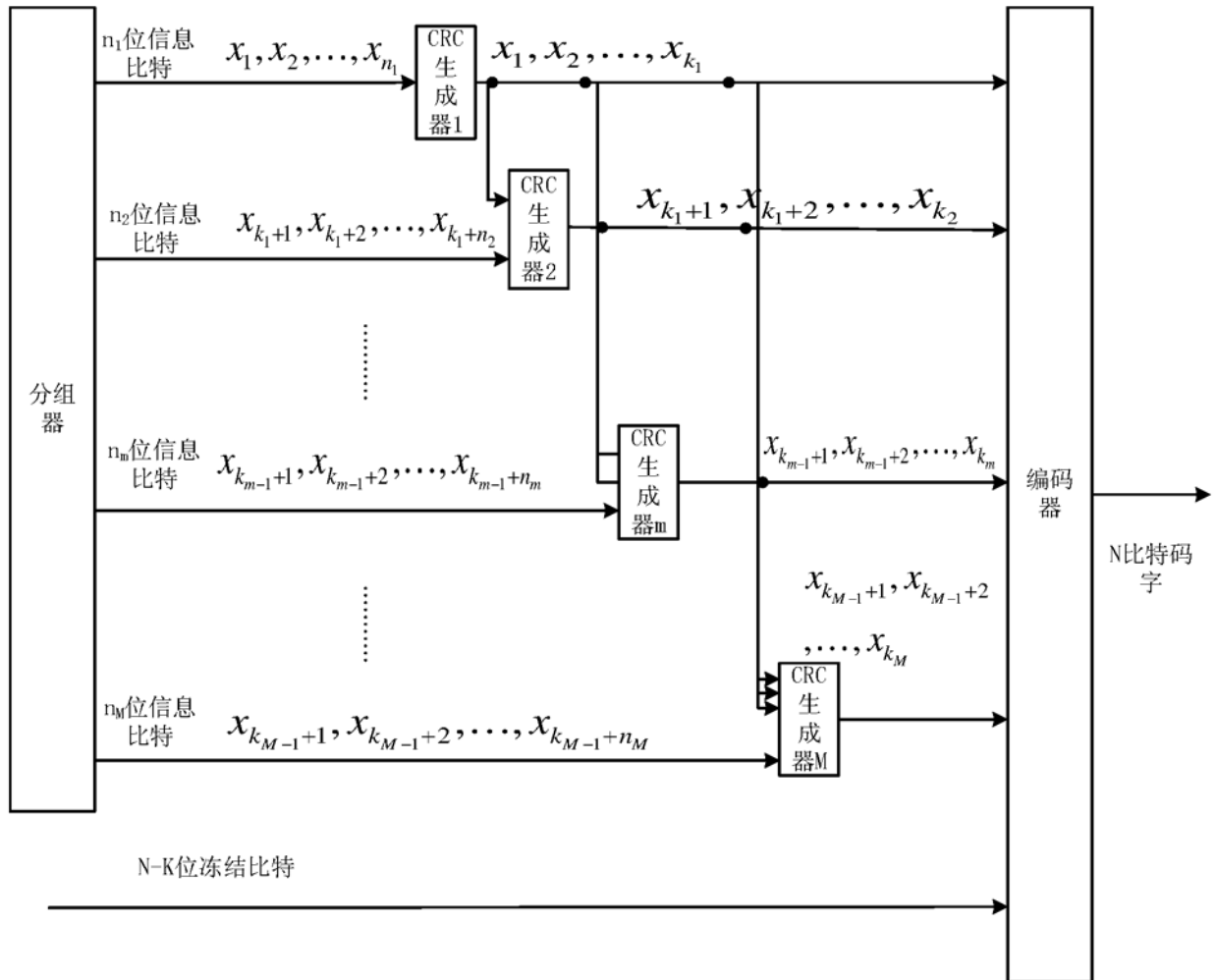


图3

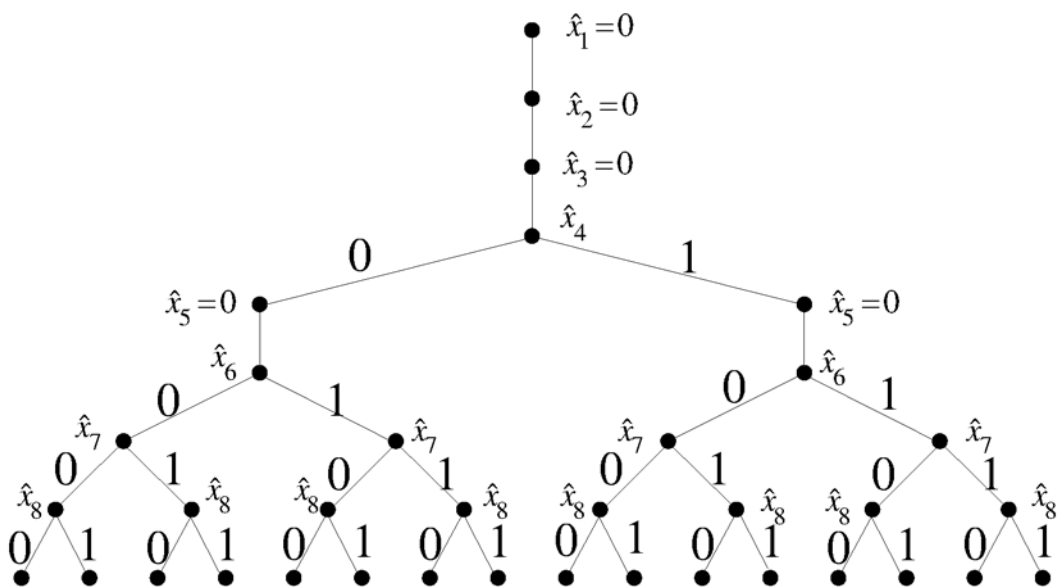


图4

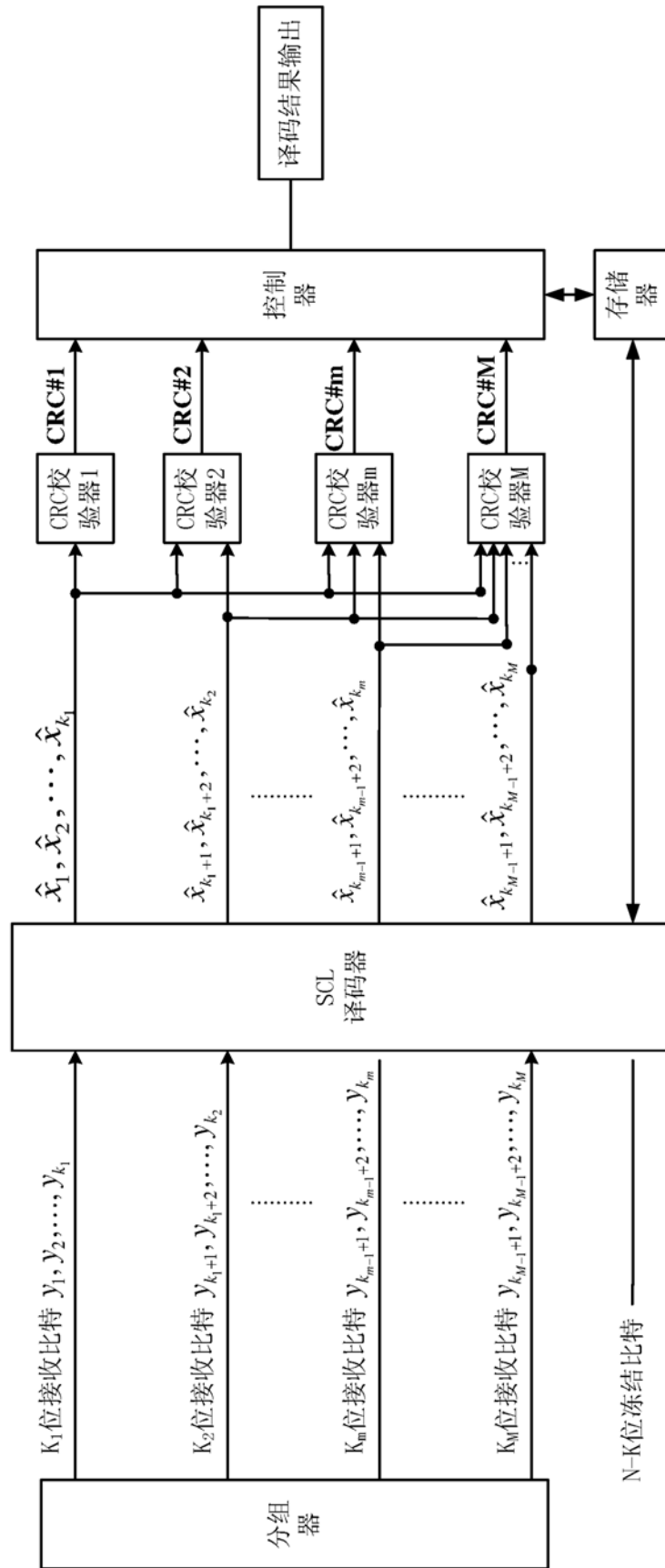


图5

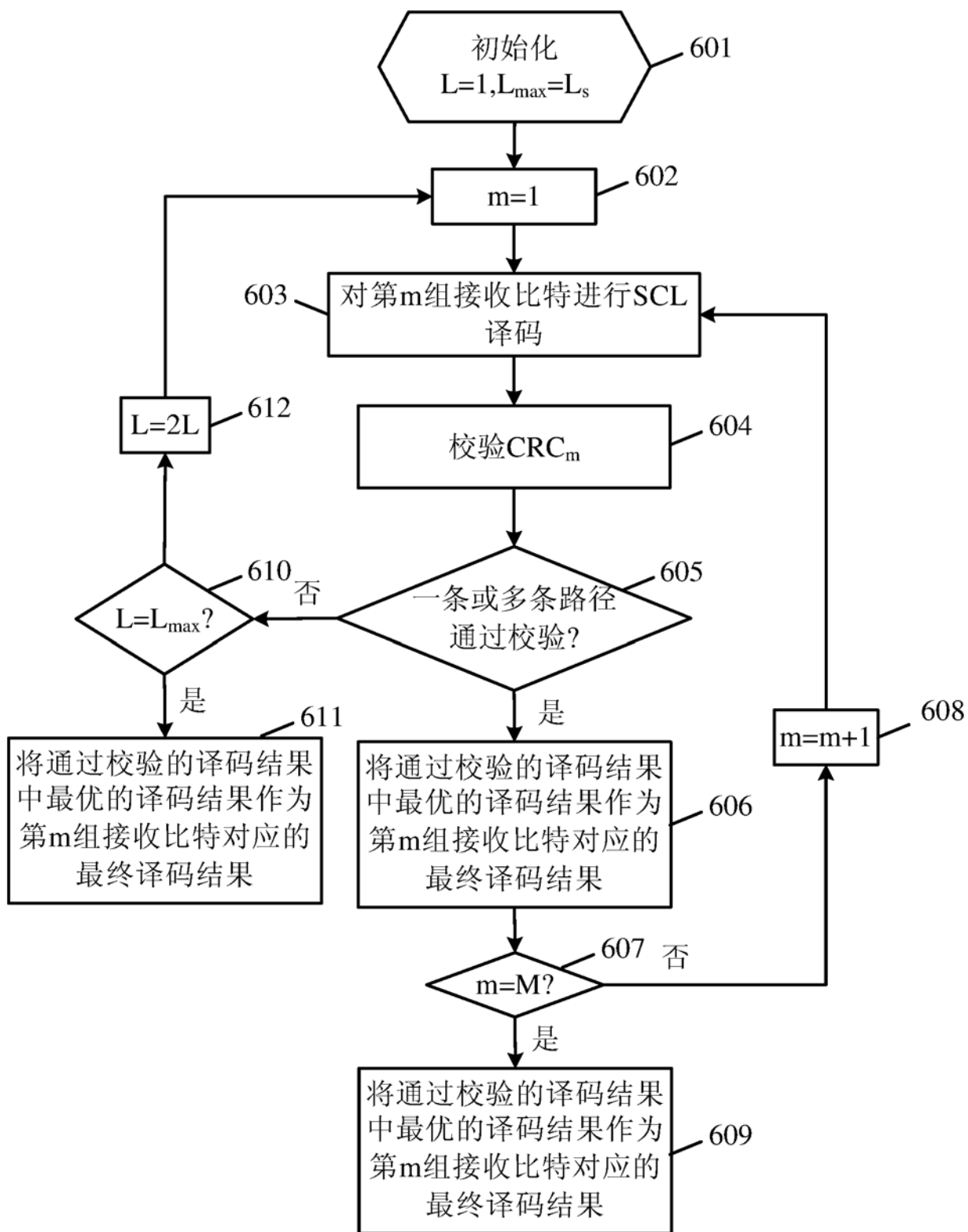


图6

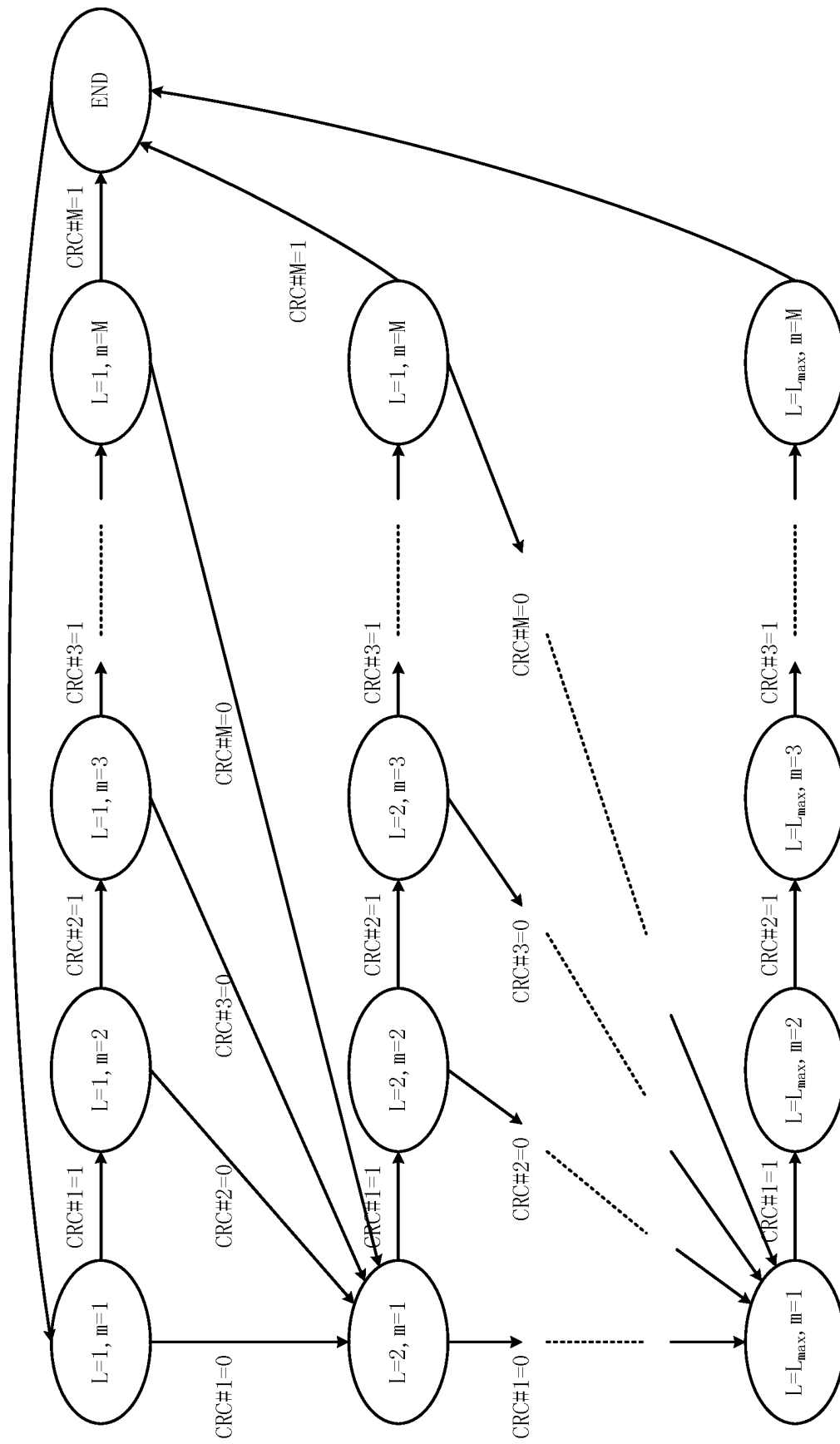


图7

$\begin{matrix} m \\ L \end{matrix}$	1	2	3	4	5	...	M
1	$T(1,1)$	$T(1,2)$	$T(1,3)$	$T(1,4)$	$T(1,5)$...	$T(1,M)$
2	$T(2,1)$	$T(2,2)$	$T(2,3)$	$T(2,4)$	$T(2,5)$...	$T(2,M)$
4	$T(4,1)$	$T(4,2)$	$T(4,3)$	$T(4,4)$	$T(4,5)$...	$T(4,M)$
8	$T(8,1)$	$T(8,2)$	$T(8,3)$	$T(8,4)$	$T(8,5)$...	$T(8,M)$
16	$T(16,1)$	$T(16,2)$	$T(16,3)$	$T(16,4)$	$T(16,5)$...	$T(16,M)$
...	$T(1,M)$
L_0	$T(L_0,1)$	$T(L_0,2)$	$T(L_0,3)$	$T(L_0,4)$	$T(L_0,5)$...	$T(L_0,M)$

图8

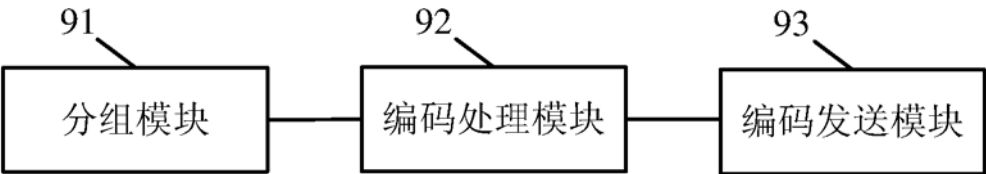


图9

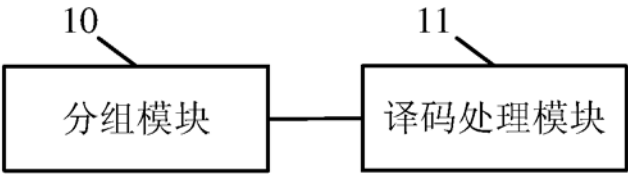


图10