



(12)发明专利申请

(10)申请公布号 CN 109474373 A
(43)申请公布日 2019.03.15

(21)申请号 201710806792.5

(22)申请日 2017.09.08

(71)申请人 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

(72)发明人 周悦 王桂杰 李榕 杜颖钢

(74)专利代理机构 北京龙双利达知识产权代理有限公司 11329

代理人 时林 毛威

(51) Int. Cl.

H04L 1/00(2006.01)

H03M 13/27(2006.01)

H03M 13/13(2006.01)

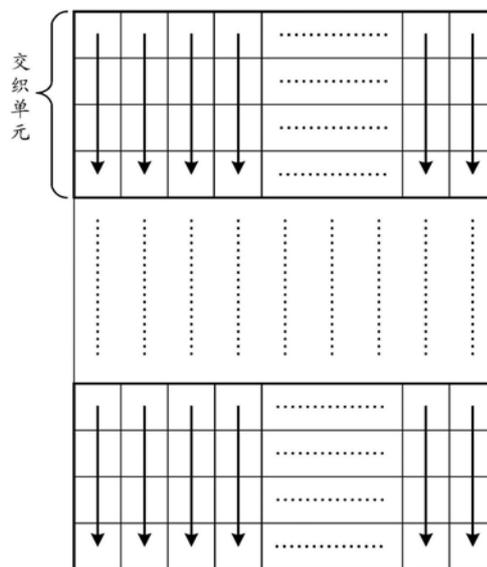
权利要求书3页 说明书27页 附图14页

(54)发明名称

交织方法和交织装置

(57)摘要

本申请提供一种交织方法,能够提高极化码的纠错性能。该方法包括:获取第一比特序列,第一比特序列包括L个比特,L为正整数;将所述L个比特按照预设的写入规则写入交织矩阵,交织矩阵包括C行R列,C和R为正整数;按照预设的读取规则从交织矩阵中读取所述L个比特,得到第二比特序列,第二比特序列包括L个比特;发送第二比特序列。



1. 一种交织方法,其特征在于,包括:

获取第一比特序列,所述第一比特序列包括L个比特,L为正整数;

将所述L个比特按照预设的写入规则写入交织矩阵,所述交织矩阵包括C行R列,其中,C和R为正整数;

按照预设的读取规则从所述交织矩阵中读取所述L个比特,得到第二比特序列,所述第二比特序列包括L个比特;

发送所述第二比特序列。

2. 根据权利要求1所述的方法,其特征在于,所述将所述L个比特按照所述写入规则写入交织矩阵,包括:

将所述L个比特按照每列B个比特逐列写入所述交织矩阵的至少一个交织单元,其中,每个交织单元包括B行R列,B为正整数。

3. 根据权利要求1所述的方法,其特征在于,所述将所述L个比特按照所述写入规则写入交织矩阵,包括:

将所述L个比特按照每列 B_i 个比特逐列写入所述交织矩阵的第i个交织单元,第i个交织单元包括 B_i 行R列, $i \geq 2$ 且为整数, B_i 为正整数,其中,所述交织矩阵包括至少两个交织单元,每个交织单元包括R列,且所述至少两个交织单元中的任意两个交织单元包括不同的行数。

4. 根据权利要求1所述的方法,其特征在于,所述将所述L个比特按照所述写入规则写入交织矩阵,包括:

将所述L个比特逐列写入所述交织矩阵的R列,其中,所述R列中的每列写入的比特数 B_j 不同,所述R列中的第一列至第R列中写入的比特数是随着列索引j的增大而递增或递减的,j遍历 $\{1, 2, \dots, R\}$ 中的取值, B_j 为正整数。

5. 根据权利要求4所述的方法,其特征在于,所述 B_j 为2的指数幂;或者,所述 B_j 为奇数或质数。

6. 根据权利要求1所述的方法,其特征在于,所述将所述L个比特按照所述写入规则写入交织矩阵,包括:

将所述L个比特分n轮逐列写入所述交织矩阵的R列,其中,每一轮写入所述R列中的至少一个列,且每一轮的所述至少一个列中写入的比特数 B_k 是随着列索引k的增大而递增的, $n \geq 2$ 且为整数, $1 \leq k \leq R$,且k和 B_k 为正整数。

7. 根据权利要求1至6中任一项所述的方法,其特征在于,将所述L个比特按照所述写入规则写入所述交织矩阵时,任意两次的写入方向可以相同或不同。

8. 根据权利要求1至7中任一项所述的方法,其特征在于,所述按照所述读取规则从所述交织矩阵中读出所述L个比特之前,所述方法还包括:

对所述交织矩阵进行列变换,其中,列变换的方式包括:奇偶对换、比特逆序或根据预定义的变换函数进行列变换。

9. 根据权利要求1至8中任一项所述的方法,其特征在于,所述读取规则包括如下至少一种:

从左至右读取;

从右至左读取;

奇数行从左向右读取,偶数行从右向左读取;

所述交织矩阵的所有行的读取方向相同且每一行的读取起点不同,其中,每一行的读取起点是根据预先定义的读取函数确定的。

10. 一种交织装置,其特征在于,包括:

收发单元,用于获取第一比特序列,所述第一比特序列包括L个比特,L为正整数;

处理单元,用于将所述L个比特按照预设的写入规则写入交织矩阵,再按照预设的读取规则从所述交织矩阵中读取所述L个比特,得到第二比特序列,C和R为正整数;

所述收发单元,还用于发送所述第二比特序列。

11. 根据权利要求10所述的装置,其特征在于,所述处理单元具体用于:

将所述L个比特按照每列B个比特逐列写入所述交织矩阵的至少一个交织单元,每个交织单元包括B行R列,其中,B为正整数。

12. 根据权利要求10所述的装置,其特征在于,所述处理单元具体用于:

将所述L个比特按照每列 B_i 个比特逐列写入所述交织矩阵的第i个交织单元,第i个交织单元包括 B_i 行R列, $i \geq 2$, B_i 和i为正整数,其中,所述交织矩阵包括至少两个交织单元,每个交织单元包括R列,且所述至少两个交织单元中的任意两个交织单元包括不同的行数。

13. 根据权利要求10所述的装置,其特征在于,所述处理单元具体用于:

将所述L个比特逐列写入所述交织矩阵的R列,其中,所述R列中的每列写入的比特数 B_j 不同,所述R列中的第一列至第R列中写入的比特数是随着列索引j的增大而递增或递减的,j遍历 $\{1, 2, \dots, R\}$ 中的取值, B_j 为正整数。

14. 根据权利要求13所述的装置,其特征在于,所述 B_j 为2的指数幂;或者,所述 B_j 为奇数或质数。

15. 根据权利要求10所述的装置,其特征在于,所述处理单元具体:

将所述L个比特分n轮逐列写入所述交织矩阵的R列,其中,每一轮写入所述R列中的至少一个列,且每一轮的所述至少一个列中写入的比特数 B_k 是随着列索引k的增大而递增的, $n \geq 2$ 且为整数, $1 \leq k \leq R$,k和 B_k 为正整数。

16. 根据权利要求10至15中任一项所述的装置,其特征在于,所述处理单元在将所述L个比特写入交织矩阵时,任意两次的写入方向可以相同或不同。

17. 根据权利要求10至16中任一项所述的装置,其特征在于,所述处理单元还用于从所述交织矩阵中读出所述L个比特之前,对所述交织矩阵进行列变换,其中,所述列变换包括奇偶对换、比特逆序或根据预定义的变换函数进行列变换。

18. 根据权利要求10至17中任一项所述的装置,其特征在于,所述读取规则包括如下至少一种:

从左至右读取;

从右至左读取;

奇数行从左向右读取,偶数行从右向左读取;

所述交织矩阵的所有行的读取方向相同且每一行的读取起点不同,其中,每一行的读取起点是根据预先定义的读取函数确定的。

19. 一种交织设备,其特征在于,包括:

收发器,用于获取第一比特序列,所述第一比特序列包括L个比特,L为正整数;

处理器,用于将所述L个比特按照预设的写入规则写入交织矩阵,再按照预设的读取规

则从所述交织矩阵中读取所述L个比特,得到第二比特序列,C和R为正整数;

所述收发器,还用于根据所述处理器的指示,发送所述处理器生成的所述第二比特序列。

20. 根据权利要求19所述的交织设备,其特征在于,所述处理器具体用于:

将所述L个比特按照每列B个比特逐列写入所述交织矩阵的至少一个交织单元,每个交织单元包括B行R列,其中,B为正整数。

21. 根据权利要求19所述的交织设备,其特征在于,所述处理器具体用于:

将所述L个比特按照每列 B_i 个比特逐列写入所述交织矩阵的第i个交织单元,第i个交织单元包括 B_i 行R列, $i \geq 2$, B_i 和i为正整数,其中,所述交织矩阵包括至少两个交织单元,每个交织单元包括R列,且所述至少两个交织单元中的任意两个交织单元包括不同的行数。

22. 根据权利要求19所述的交织设备,其特征在于,所述处理器具体用于:

将所述L个比特逐列写入所述交织矩阵的R列,其中,所述R列中的每列写入的比特数 B_j 不同,所述R列中的第一列至第R列中写入的比特数是随着列索引j的增大而递增或递减的,j遍历 $\{1, 2, \dots, R\}$ 中的取值, B_j 为正整数。

23. 根据权利要求22所述的交织设备,其特征在于,所述 B_j 为2的指数幂;或者,所述 B_j 为奇数或质数。

24. 根据权利要求19所述的交织设备,其特征在于,所述处理器具体用于:

将所述L个比特分n轮逐列写入所述交织矩阵的R列,其中,每一轮写入所述R列中的至少一个列,且每一轮的所述至少一个列中写入的比特数 B_k 是随着列索引k的增大而递增的, $n \geq 2$ 且为整数, $1 \leq k \leq R$,且k和 B_k 为整数。

25. 根据权利要求19至24中任一项所述的交织设备,其特征在于,所述处理器在将所述L个比特写入所述交织矩阵时,任意两次的写入方向可以相同或不同。

26. 根据权利要求19至25中任一项所述的交织设备,其特征在于,所述处理器还用于从所述交织矩阵中读出所述L个比特之前,对所述交织矩阵进行列变换,其中,所述列变换包括奇偶对换、比特逆序或根据预定义的变换函数进行列变换。

27. 根据权利要求19至26中任一项所述的交织设备,其特征在于,所述读取规则包括如下至少一项:

从左至右读取;

从右至左读取;

奇数行从左向右读取,偶数行从右向左读取;

所述交织矩阵的所有行的读取方向相同且每一行的读取起点不同,其中,每一行的读取起点是根据预先定义的读取函数确定的。

28. 一种芯片,其特征在于,包括存储器和处理器,所述存储器用于存储计算机程序,所述处理器用于从存储器中调用并运行所述计算机程序,使得安装有该芯片的通信设备执行权利要求1至9中任一项所述的方法。

29. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储有计算机程序,所述计算机程序在计算机上执行时,使得计算机执行权利要求1至9中任一项所述的方法。

交织方法和交织装置

技术领域

[0001] 本申请涉及信道编码领域,尤其涉及一种交织方法和交织装置。

背景技术

[0002] 通信系统通常采用信道编码提高数据传输的可靠性,保证通信质量。在衰落信道中,比特差错经常成串发生(即,突发错误),而信道编码仅在检测和纠正单个差错或不太长的差错串(即,随机错误)时才有效。为此,现有技术提出了交织技术。交织的作用是将原始数据序列打乱,使得交织前后的数据序列的相关性减弱,这样可以降低数据突发错误的概率,即使出现差错,也是单个差错或长度很短的差错串,这样,就可以利用信道编码的纠错能力纠正差错,从而恢复出原始数据序列。根据交织方式的不同,主要分为行列交织和随机交织。

[0003] 极化码(也即,Polar码)是第一个理论上证明可以取得香农容量且具有低编、译码复杂度的好码,从而得到了广泛应用。当采用Polar码进行信道编码时,如果采用随机交织方式,在离线计算交织序列时,需要存储置换序列供交织和解交织使用,在码长较长的情况下,随机交织所需的存储资源非常大,甚至不可接受。而如果采用行列交织方式,在高阶调制下,纠错性能较差。

发明内容

[0004] 本申请提供一种交织方法和交织装置,能够提高极化码在高阶调制下的纠错性能。

[0005] 第一方面,本申请提供了一种交织方法,该方法包括:获取第一比特序列,第一比特序列包括L个比特,L为正整数;将该L个比特按照预设的写入规则写入交织矩阵,交织矩阵包括C行R列,C和R为正整数;按照预设的读取规则从交织矩阵中读取该L个比特,得到第二比特序列,第二比特序列包括L个比特;发送第二比特序列。

[0006] 应理解,第一比特序列为待交织的比特序列。第二比特序列为交织后的序列。将第一比特序列中包括的L个比特按照写入规则写入交织矩阵,在按照读取规则读取该L个比特,得到第二比特序列的过程,实际上就是交织的过程。

[0007] 根据本申请实施例的写入规则,将待交织的比特序列写入交织矩阵中,使得交织矩阵的至少一行且至少一列中写入的比特在第一比特序列中不连续;或者,交织矩阵的至少一行中写入的比特在第一比特序列中不连续,且至少有两列中写入的比特数不相等;或者,交织矩阵的至少一行中写入的比特在第一比特序列中不连续,且至少有一列中写入的比特数在第一比特序列中不连续。

[0008] 可以理解的是,将第一比特序列中的L个比特写入交织矩阵中,使得交织矩阵满足上面描述的这些特点,是在本申请提出的交织方法的外在表现形式。本申请实施例利用极化码的准周期特性,在行列交织的基础上设计的交织方法,可以提高极化码用于信道编码时的纠错性能。

[0009] 结合第一方面,在第一方面的某些实现方式中,将该L个比特按照所述写入规则写入交织矩阵,包括:将该L个比特按照每列B个比特逐列写入交织矩阵的至少一个交织单元,每个交织单元包括B行R列,其中,B为正整数。

[0010] 结合第一方面,在第一方面的某些实现方式中,将该L个比特按照所述写入规则写入交织矩阵,包括:将该L个比特按照每列 B_i 个比特逐列写入交织矩阵的第i个交织单元,第i个交织单元包括 B_i 行R列, $i \geq 2$ 且为整数, B_i 为正整数,其中,交织矩阵包括至少两个交织单元,每个交织单元包括R列,且该至少两个交织单元中的任意两个交织单元包括不同的行数。

[0011] 结合第一方面,在第一方面的某些实现方式中,将该L个比特按照写入规则写入交织矩阵,包括:将该L个比特逐列写入交织矩阵的R列,其中,该R列中的每列写入的比特数 B_j 不同,该R列中的第一列至第R列中写入的比特数是随着列索引j的增大而递增或递减的,j遍历 $\{1, 2, \dots, R\}$ 中的取值, B_j 为正整数。

[0012] 结合第一方面,在第一方面的某些实现方式中, B_j 为2的指数幂;或者, B_j 为奇数或质数。

[0013] 结合第一方面,在第一方面的某些实现方式中,将该L个比特按照所述写入规则写入交织矩阵,包括:将该L个比特分n轮逐列写入交织矩阵的R列,其中,每一轮写入该R列中的至少一个列,且每一轮的该至少一个列中写入的比特数 B_k 是随着列索引k的增大而递增的, $n \geq 2$ 且为整数, $1 \leq k \leq R$,且k和 B_k 为正整数。

[0014] 结合第一方面,在第一方面的某些实现方式中,将该L个比特按照所述写入规则写入交织矩阵时,任意两次的写入方向可以相同或不同。

[0015] 结合第一方面,在第一方面的某些实现方式中,按照所述读取规则从交织矩阵中读出该L个比特之前,该方法还包括:对交织矩阵进行列变换,其中,列变换的方式包括奇偶对换、比特逆序或根据预定义的变换函数进行列变换。

[0016] 结合第一方面,在第一方面的某些实现方式中,所述读取规则包括如下至少一项:从左至右读取;从右至左读取;奇数行从左向右读取,偶数行从右向左读取;交织矩阵的所有行的读取方向相同且每一行的读取起点不同,其中,每一行的读取起点是根据预先定义的读取函数确定的。

[0017] 第二方面,本申请提供一种交织装置,用于执行第一方面或第一方面的任意可能的实现方式中的方法。具体地,该装置包括执行第一方面或第一方面的任意可能的实现方式中的方法的单元。

[0018] 本申请实施例提供的交织装置,利用极化码的准周期特征进行比特交织,可以提高极化码用于信道编码时的纠错性能。

[0019] 第三方面,本申请提供一种交织设备,该交织设备包括一个或多个处理器,一个或多个存储器,一个或多个收发器(每个收发器包括发射机和接收机)。发射机或接收机通过天线收发信号。存储器用于存储计算机程序指令(或者说,代码)。处理器用于执行存储器中存储的指令,当指令被执行时,处理器执行第一方面或第一方面的任意可能的实现方式中的方法。

[0020] 第四方面,本申请提供一种计算机可读存储介质,该计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述第一方面或第一方面的任意可能的

实现方式中的方法。

[0021] 第五方面,本申请提供一种芯片(或者说,芯片系统),包括存储器和处理器,存储器用于存储计算机程序,处理器用于从存储器中调用并运行该计算机程序,使得安装有该芯片的通信设备执行上述第一方面及其任意一种可能的实现方式中的方法。

[0022] 第六方面,本申请提供一种计算机程序产品,所述计算机程序产品包括:计算机程序代码,当所述计算机程序代码在计算机上运行时,使得计算机执行上述第一方面及其任意一种可能的实现方式中的方法。

[0023] 第七方面,本申请提供一种编码装置,该编码装置具有实现上述第一方面及其第一方面任意一种可能的实现方式中的方法的功能。所述功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的模块。

[0024] 在一个可能的设计中,当所述功能的部分或全部通过硬件实现时,所述编码装置包括:输入接口电路,用于获取第一比特比特;逻辑电路,用于执行上述第一方面及其第一方面的任意一种可能的设计中的交织方法;输出接口电路,用于输出第二比特序列。

[0025] 可选的,所述编码装置可以是芯片或者集成电路。

[0026] 在一个可能的设计中,当所述功能的部分或全部通过软件实现时,所述编码装置包括:存储器,用于存储程序;处理器,用于执行所述存储器存储的所述程序,当所述程序被执行时,所述编码装置可以实现上述第一方面及其第一方面的任意一种可能的设计中所述的交织方法。

[0027] 在一个可能的设计中,当所述功能的部分或全部通过软件实现时,所述编码装置包括处理器。用于存储程序的存储器位于所述编码装置之外,处理器通过电路/电线与存储器连接,用于读取并执行所述存储器中存储的程序。

[0028] 第八方面,本申请提供一种解交织的方法,该方法包括:获取待解交织的比特序列;按照预设的写入规则和读取规则,对待解交织的比特序列进行解交织,得到解交织后的比特序列。

[0029] 在理想无噪声的情况下,解交织后的比特序列与第一比特序列相同。

[0030] 需要说明的是,解交织是交织的逆过程。当译码端获取到待解交织的比特序列,执行与编码端的写入规则和读取规则的逆操作,对待解交织的比特序列进行解交织。本领域技术人员在第一方面描述的交织方法的基础上,容易得到解交织的方法,本文中不作详述。

[0031] 应理解,针对以上第一方面及其任意一种可能的实现方式中的交织方法,都会有对应的解交织的方法。这里不再一一说明。

[0032] 第九方面,本申请提供一种解交织装置,用于执行第八方面或第八方面的任意可能的实现方式中的解交织的方法。具体地,该装置包括执行第八方面或第八方面的任意可能的实现方式中的方法的单元。

[0033] 第十方面,本申请提供一种解交织设备,该解交织设备包括一个或多个处理器,一个或多个存储器,一个或多个收发器(每个收发器包括发射机和接收机)。发射机或接收机通过天线收发信号。存储器用于存储计算机程序指令(或者说,代码)。处理器用于执行存储器中存储的指令,当指令被执行时,处理器执行第八方面或第八方面的任意可能的实现方式中的方法。

[0034] 第十一方面,本申请提供一种计算机可读存储介质,该计算机可读存储介质中存

储有指令,当其在计算机上运行时,使得计算机执行上述第八方面或第八方面的任意可能的实现方式中的方法。

[0035] 第十二方面,本申请提供一种芯片(或者说,芯片系统),包括存储器和处理器,存储器用于存储计算机程序,处理器用于从存储器中调用并运行该计算机程序,使得安装有该芯片的通信设备执行上述第八方面及其任意八种可能的实现方式中的方法。

[0036] 第十三方面,本申请提供一种计算机程序产品,所述计算机程序产品包括:计算机程序代码,当所述计算机程序代码在计算机上运行时,使得计算机执行上述第八方面及其任意一种可能的实现方式中的方法。

[0037] 第十四方面,本申请提供一种译码装置,该译码装置具有实现上述第八方面及其第八方面任意一种可能的实现方式中的方法的功能。所述功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的模块。

[0038] 在一个可能的设计中,当所述功能的部分或全部通过硬件实现时,所述译码装置包括:输入接口电路,用于获取待解交织的比特序列;逻辑电路,用于执行上述第八方面及其第八方面的任意一种可能的设计中的解交织的方法;输出接口电路,用于输出解交织后的比特序列。

[0039] 可选的,译码装置可以是芯片或者集成电路。

[0040] 在一个可能的设计中,当所述功能的部分或全部通过软件实现时,译码装置包括:存储器,用于存储程序;处理器,用于执行存储器存储的所述程序,当程序被执行时,译码装置可以实现上述第八方面及其第八方面的任意一种可能的设计中所述的解交织的方法。

[0041] 在一个可能的设计中,当所述功能的部分或全部通过软件实现时,译码装置包括处理器。用于存储程序的存储器位于所述译码装置之外,处理器通过电路/电线与存储器连接,用于读取并执行存储器中存储的程序。

[0042] 可选的,上述存储器可以是物理上独立的单元,也可以与处理器集成在一起。

[0043] 在本申请实施例中,利用极化码的准周期特性,在行列交织的基础上设计的交织方法,其纠错性能接近甚至超过随机交织,可以提高极化码用于信道编码时的纠错性能。

附图说明

[0044] 图1为适用于本申请实施例的无线通信系统。

[0045] 图2是采用无线技术进行通信的基本流程图。

[0046] 图3是行列交织的示意图。

[0047] 图4是Polar码的准周期特性的示意图。

[0048] 图5是Polar码的准周期特性的另一个示意图。

[0049] 图6是交织和解交织的示意性流程图。

[0050] 图7是写入规则的一个示例。

[0051] 图8是写入规则的另一个示例。

[0052] 图9示出了两种交织深度的交织单元在交织矩阵中的一种排列形式。

[0053] 图10示出了两种交织深度的交织单元在交织矩阵中的另一种排列形式。

[0054] 图11是两种交织深度的交织单元按行写入的示意图。

- [0055] 图12为读取规则的一个示例。
- [0056] 图13是不同交织深度的交织单元的并行读取方式的示例。
- [0057] 图14是写入规则的再一个示例。
- [0058] 图15是写入规则的又一个示例。
- [0059] 图16是写入规则的又一个示例。
- [0060] 图17是写入方向的示意图。
- [0061] 图18为两种交织深度的交织单元的拼接示意图。
- [0062] 图19为本申请实施例提供的交织装置500的示意图。
- [0063] 图20为本申请实施例提供的交织设备700的示意性结构图。
- [0064] 图21为本申请实施例提供的终端设备800的结构图。
- [0065] 图22为本申请实施例提供的解交织装置900的示意图。
- [0066] 图23为本申请实施例提供的解交织设备1000的示意性结构图。

具体实施方式

[0067] 下面将结合附图,对本申请中的技术方案进行描述。

[0068] 图1为适用于本申请实施例的无线通信系统100。该无线通信系统中可以包括至少一个网络设备,该网络设备与一个或多个终端设备(例如,图1中所示的终端设备#1和终端设备#2)进行通信。网络设备可以是基站,也可以是基站与基站控制器集成后的设备,还可以是具有类似通信功能的其它设备。

[0069] 本申请实施例提及的无线通信系统包括但不限于:窄带物联网系统(Narrow Band-Internet of Things,NB-IoT)、全球移动通信系统(Global System for Mobile Communications,GSM)、增强型数据速率GSM演进系统(Enhanced Data rate for GSM Evolution,EDGE)、宽带码分多址系统(Wideband Code Division Multiple Access,WCDMA)、码分多址2000系统(Code Division Multiple Access,CDMA2000)、时分同步码分多址系统(Time Division-Synchronization Code Division Multiple Access,TD-SCDMA),长期演进系统(Long Term Evolution,LTE)、下一代5G移动通信系统的三大应用场景,即增强移动带宽(Enhance Mobile Broadband,eMBB),高可靠性低延迟通信(Ultra Reliable Low Latency Communication,URLLC)和增强海量机器连接通信(Massive Machine Type Communication,eMTC)或者将来出现的新的通信系统。

[0070] 本申请实施例中所涉及到的终端设备可以包括各种具有无线通信功能的手持设备、车载设备、可穿戴设备、计算设备或连接到无线调制解调器的其它处理设备。终端设备可以是移动台(Mobile Station,MS)、用户单元(subscriber unit)、蜂窝电话(cellular phone)、智能电话(smart phone)、无线数据卡、个人数字助理(Personal Digital Assistant,PDA)电脑、平板型电脑、无线调制解调器(modem)、手持设备(handset)、膝上型电脑(laptop computer)、机器类型通信(Machine Type Communication,MTC)终端等。

[0071] 图1中的网络设备与终端设备之间采用无线技术进行通信。当网络设备发送信号时,其为编码端,当网络设备接收信号时,其为译码端。终端设备也是一样的,当终端设备发送信号时,其为编码端,当终端设备接收信号时,其为译码端。

[0072] 另外,编码端也可以认为是发送端,译码端也可以认为是接收端。

[0073] 图2是采用无线技术进行通信的基本流程图。发送端的信源依次经过信源编码、信道编码、速率匹配和调制后在信道上发出,接收端收到信号后依次经过解调、解速率匹配、信道解码和信源解码后获得信宿。

[0074] 为了便于理解,首先对本申请涉及的相关技术作简单介绍。

[0075] (1) 信道编码

[0076] 信道编解码是无线通信领域的核心技术之一,其性能的改进将直接提升网络覆盖及用户传输速率。目前,极化码是可理论证明达到香农极限,并且具有可实用的线性复杂度编译码能力的信道编码技术。极化码构造的核心是通过“信道极化”的处理,在编码侧,采用编码的方法使各个子信道呈现出不同的可靠性,当码长持续增加时,一部分信道将趋向于容量接近于1的无噪信道,另一部分信道趋向于容量接近于0的全噪信道,选择在容量接近于1的信道上直接传输信息以逼近信道容量。

[0077] Polar码的编码策略正是应用了这种现象的特性,利用无噪信道传输用户有用的信息,全噪信道传输约定的信息或者不传信息。Polar码也是一种线性块码,其编码矩阵(也称为生成矩阵)为 F_N ,编码过程为 $x_1^N = \mu_1^N \cdot F_N$ 。其中, $\mu_1^N = (\mu_1, \mu_2, \dots, \mu_N)$ 是一个二进制的行矢量,长度为 N (即,码长),且 $N = 2^n$, n 为正整数。 F_N 是一个 $N \times N$ 的矩阵,且 $F_N = F_2^{\otimes(\log_2 N)}$ 。 $F_2^{\otimes(\log_2 N)}$ 定义为 $\log_2 N$ 个矩阵 F_2 的克罗内克(Kronecker)乘积, $F_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ 。以上各式中涉及到的加法、乘法操作均为二进制伽罗华域上的加法、乘法操作。

[0078] Polar码的编码过程中, u_1^N 中的一部分比特用来携带信息,称为信息比特集合。这些比特的索引的集合记作 A 。另外的一部分比特设置为接收端和发送端预先约定的固定值,称之为固定比特集合或冻结比特(frozen bits)集合,其索引的集合用 A 的补集 A^c 表示。Polar码的编码过程相当于 $x_1^N = \mu_A F_N(A) \oplus \mu_{A^c} F_N(A^c)$ 。这里, $F_N(A)$ 是 F_N 中由集合 A 中的索引对应的行得到的子矩阵。 $F_N(A^c)$ 是 F_N 中由集合 A^c 中的索引对应的行得到的子矩阵。 u_A 为 u_1^N 中的信息比特集合,数量为 K 。 u_{A^c} 为 u_1^N 中的固定比特集合,其数量为 $(N-K)$,是已知比特。这些固定比特通常被设置为0,但是只要接收端和发送端预先约定,固定比特可以被任意设置。从而,Polar码的编码输出可简化为 $x_1^N = \mu_A \cdot F_N(A)$ 。这里的 u_A 为 u_1^N 中的信息比特集合, u_A 为长度 K 的行矢量,即 $|A| = K$,符号 $| \cdot |$ 表示集合中元素的个数, K 为信息块大小, $F_N(A)$ 是矩阵 F_N 中由集合 A 中的索引对应的那些行得到的子矩阵, $F_N(A)$ 是一个 $N \times N$ 的矩阵。

[0079] (2) 高阶调制

[0080] 高阶调制的一个固有特性是比特可靠性依赖于条子星座图的映射关系,映射到同一个调制符号的比特具有不同的可靠性。换句话说,映射到同一个调制符号上的不同比特的受保护程度是不同的。例如,对于16QAM来说,映射到一个调制符号上的4个比特 b_i 的保护级别 $Q(b_i)$ 的排序是 $Q(b_1) = Q(b_2) > Q(b_3) = Q(b_4)$, $i \in \{1, 2, 3, 4\}$ 。又例如,对于64QAM来说,映射到一个调制符号上的6个比特 b_i 的保护级别 $Q(b_i)$ 的排序是 $Q(b_1) = Q(b_2) > Q(b_3) = Q(b_4) > Q(b_5) = Q(b_6)$ 。容易想到的是,保护级别较低的比特在传输过程中更容易发生失真,造成译码过程的突发错误。因此,高阶调制的这种特性对于纠错码(包括Polar码)来说显然是不利的。

[0081] (3) 交织技术

[0082] 在衰落信道中,错误常常是成串发生的。而信道编码仅能检测和校正单个差错和不太长的差错串。为了解决成串的比特差错问题,现有技术提出了交织技术。交织技术是将原始的数据序列打乱,使得交织前后的数据序列的相关性减弱。这样,即使在传输的过程中发生了成串的差错,在恢复交织前的数据序列时,差错也就变成单个的错误比特,并且,由于信道本身具有一定的纠错能力,恢复出原始数据序列的几率增大,降低了数据突发错误的概率,可以提高数据传输时的抗干扰能力。

[0083] 目前,常用的交织方法包括行列交织和随机交织。

[0084] 行列交织是根据编码后的码长产生一个 $Z_1 \times Z_2$ 的矩阵。将原始比特序列按照“行进列出”或“列进行出”的操作进行交织,得到交织后的比特序列。

[0085] 参见图3,图3是行列交织的示意图。图3中以“行进列出”为例,原始比特序列为 $[X_{11}, X_{12}, X_{13}, \dots, X_{1m}, X_{21}, X_{22}, X_{23}, \dots, X_{2m}, \dots, X_{n1}, X_{n2}, X_{n3}, \dots, X_{nm}]$,交织后的序列为 $[X_{11}, X_{21}, X_{31}, \dots, X_{n1}, X_{12}, X_{22}, X_{32}, \dots, X_{n2}, \dots, X_{1m}, X_{2m}, X_{3m}, \dots, X_{nm}]$ 。

[0086] 对于Polar码来讲,采用简单的行列交织,在高阶调制下,既不能随机化突发错误,也不能够抵消高阶调制下不同比特的保护级别的不同对译码性能带来的影响。

[0087] 为此,本申请提出一种交织方法,能够提高极化码在高阶调制性能下的纠错性能。

[0088] 为了更好地理解本申请的设计构思,首先对极化码的准周期特性进行说明。

[0089] 对于Polar的极化信道而言,极化信道的可靠度随着极化信道的序号呈现一种准周期特性。利用3GPP新无线(New Radio, NR)采纳的Polar码构造序列来描述极化信道的归一化可靠度,母码长度为1024的极化信道的可靠度随信道序号的变化趋势参见图4所示,母码长度为256的极化信道对信道序号的变化趋势参见图5所示。

[0090] 需要说明的是,归一化可靠度可以描述如下(以N个极化信道进行描述):计算N个极化信道的可靠度,将这N个极化信道的可靠度进行量化(或者说,归一化),得到每个极化信道的可靠度的量化值。例如 $N=1024$,将这N个量化值中的最大量化值赋值1023,再将次大的量化值赋值1022,依次类推,直至将最小的量化值赋值0。此时得到的可靠度的排序序列即为该N个极化信道的归一化可靠度的排序序列。又例如, $N=128$,将这N个量化值中的最大量化值赋值127,再将次大的量化值赋值126,依次类推,直至将最小的量化值赋值0。再例如, $N=1024$,将这N个量化值分别除以N个量化值中的最大量化值,此后N个量化值的最大值为1,各个信道可靠度的量化值分布在0到1的区间中。

[0091] 图5与图4中呈现的极化信道可靠度随信道序号的变化趋势是类似的。图4相当于4个图5构成。以长度等于1024的母码尺度为例,其中包含了4个周期,每个周期内包括的极化信道的个数为256,并且后一个周期内极化信道的平均可靠度高于前一个周期。参见图5可以发现,极化信道存在不同的尺度的周期,在每一个周期内,极化信道的可靠度均存在随信道序号增加而平均提升的变化,并且下一个周期的平均可靠度比上一周期的高。图5中以32个子信道为周期(用虚线标记了周期间隔),在每一个周期内,总的来讲,极化信道可靠度从小变大。如果以8个子信道为周期,例如 $[0, 7]$ 、 $[8, 15]$,每8个子信道的可靠度也存在从小变大的趋势。并且每一个周期的平均可靠度相对于上一周期的平均可靠度有所提升。该准周期性变化,最小可以细分至以2个极化信道为一个周期(或,称作极化周期或准极化周期)。

[0092] 考虑到极化码具有其它纠错码不具有的准周期特性,本申请提出一种交织方法,

可以针对高阶调制的调制符号对不同比特的保护级别的差异,提升极化码的纠错性能。

[0093] 根据上面的介绍已经知道,由于在高阶调制方式(例如16QAM、64QAM)下,连续输入的比特的调制保护级别是不同的。因此,在本申请实施例中,避免将一个极化周期内的比特映射到一个高阶调制符号内传输。以16QAM为例,每一个调制符号包括4个比特的信息,因此,要避免一个极化周期内极化程度由小到大或由大到小的4个连续的比特同时分配给一个调制符号。

[0094] 基于这样的设计构思,本申请提出了多种可能的交织方式。在本申请实施例中,我们将交织前的原始比特序列记作第一比特序列,将交织后的比特序列记作第二比特序列。第二比特序列也可以称作交织序列,或者交织后的比特序列。

[0095] 应理解,交织前后,比特序列中的包括的比特的数量是不会发生变化的。也就是说,如果第一比特序列包括L个比特,那么交织后得到的第二比特序列也包括L个比特,其中,L为正整数。

[0096] 参见图6,图6是交织和解交织的示意图。编码端获取第一比特序列,第一比特序列包括L个比特,L为正整数;将第一比特序列包括的L个比特按照预设的写入规则写入交织矩阵,再按照预设的读取规则将写入交织矩阵的L个比特读出,得到第二比特序列,完成交织过程。其中,第二比特序列包括L个比特,交织矩阵包括C行R列,C和R为正整数。发送端对第二比特序列进行调制、映射后发送。译码端进行逆映射和解调后,获取到待解交织的比特序列,对待解交织的比特序列进行解交织,得到解交织后的比特序列。

[0097] 在理想无噪声的情况下,解交织后的比特序列与第一比特序列相同。

[0098] 通常,我们将速率匹配之前的Polar码的码长称作母码长度,记作N。将速率匹配之后Polar码的码长记作M。在本申请提出的交织方法中,第一比特序列的长度(或者说L的取值)可以是Polar编码时速率匹配之后的长度M,也即 $L=M$ 。或者,也可以是速率匹配之前的母码长度N,此时 $L=N$ 。

[0099] 另外,本文中,一些实施例以“列进行出”作为示例进行说明,而另一些实施例以“行进列出”作为示例进行说明。可以理解的是,在行列交织的过程中,“行”和“列”是相对而言的,因此,根据一个“行进列出”的实施例,本领域技术人员容易得到“列进行出”的交织过程。同样地,根据“列进行出”的交织过程,也可以推导出“行进列出”的交织过程。因此,以下实施例中,不对每个实施例的“行进列出”和“列进行出”作详细描述。每个实施例中,通过描述其中一种交织方式,本领域技术人员在此基础上,可以得到另一种交织方式。

[0100] 本申请实施例中采用行列交织,以下将行列交织中的交织矩阵(也可以称作交织器)的行数记作C,列数记作R。C和R为正整数。

[0101] 本申请实施例中,写入规则包括多种方式,下文一一进行详细说明。

[0102] 方式1

[0103] 将L个比特按照每列B个比特写入交织矩阵的至少一个交织单元,每个交织单元包括B行R列。其中,B为正整数。

[0104] 在本申请实施例中,交织单元是交织矩阵的一部分,通常也是一个矩阵。需要说明的是,在不同的实施例中,交织单元的大小可能并不相等。或者,在一个实施例中,一个交织矩阵中可能包括几种大小不同的交织单元。交织单元的大小具体可以参见各实施例中所作的说明。

[0105] 其中,如果交织单元按行写入,每一行写入的比特数也可以称作该交织单元的交织深度。如果交织单元按列写入,每一列写入的比特数称作该交织单元的交织深度。

[0106] 另外,这L个比特即为待交织的比特。因此,在一些实施例中,将这L个比特描述为“待交织比特”。

[0107] 参见图7,图7是写入规则的一个示例。每一列写入B个比特,B为恒定值,然后换列。当写完交织矩阵的所有时,从第(B+1)行继续逐列写入,每列写入B个比特。

[0108] 在本实施例中,交织矩阵的行数C的取值等于星座点数C,其中星座点数 $C=2^M$,M为调制阶数,M为正整数。

[0109] 交织矩阵的列数R可以根据速率匹配之前的码长N计算,或者也可以根据速率匹配之后的码长M计算。

[0110] 如果是根据速率匹配之前的码长N(即,母码长度)计算,则, $R=\lceil N/C \rceil$;

[0111] 如果是根据速率匹配之后的码长M计算,则 $R=\lceil M/C \rceil$ 。

[0112] B的取值可以为极化周期的函数,例如, $B=2^n, n=1, 2, 3, \dots, 2^{\lceil \log_2 N \rceil}$ 。或者,

[0113] B可以为M或者C的函数。例如, $B=\lceil \alpha(\log_2 C) + \beta \rceil$ 、 $B=\lceil \alpha \cdot (\log_2 C + \beta) \rceil$ 、 $B=\lceil \alpha \cdot M + \beta \rceil$ 等。

[0114] 其中, α 的取值范围为(0, 2), β 的取值范围为(-2, +2)、 $\lceil \]$ 表示向上取整, $\lfloor \]$ 表示向下取整。

[0115] 例如, $\alpha=0.5, \beta=0$ 时,若 $M=4$,则B的取值等于8。若 $M=6$,则B的取值等于32。

[0116] 需要说明的是,在写入交织矩阵的过程中,当L个比特已经全部写入交织矩阵,而交织矩阵还未填满的情况下,未写入比特的位置写入NULL。

[0117] 在根据速率匹配之前的码长N计算B时,可以用如下代码实现:

```

B = 32; %设定分组最小单元大小
rowNum = ConsNum; %交织器行数等于调制方法星座点数
colNum = 向上取整(N/ConsNum); %根据编码后长度计算列数
groupNum =向上取整(N/B); %最小单元个数
lgroupNum = groupNum - (rowNum/-1)*colNum; %计算需要补 NULL 的单元个数
ind = 1:rowNum*colNum; %生成交织矩阵元素初始序号
indT = 将序号向量按列填充到 B 行 groupNum 列的矩阵中; %生成待交织矩阵
indTT = []; %生成交织矩阵初始值
for n = 1:rowNum %交织
    if n == rowNum %对最后一个单元进行补 NULL 处理
        lgroupInd = 生成含有 NULL 元素的单元;
        indTT = 将新生成的单元填充到交织矩阵的下方;
    else
        indTT = 将新生成的单元填充到交织矩阵的下方;
    end
end
end
seq = reshape( indTT', 1, rowNum*colNum);
seq = seq(seq<=N)

```

[0119] 在根据速率匹配之后的码长M计算B时,可以用如下代码实现:

```

B = 8; %设定最小单元大小
rowNum = ConsNum; %交织器行数等于调制方法星座点数
N0 = 2^(上取整(log2(N))); %获取母码长度 N0
colNum = 上取整(N0/ConsNum); %获取交织器列数
[0120] groupNum = 上取整(N0/); %获取最小单元 B 的个数
ind = 1:N0; %获取序号
ind(打孔/截短) = 0; %将速率匹配打孔、截短的信道置零
ind = 反转(ind);

indT = 将序号向量按列填充到 B 行 groupNum 列的矩阵中;
indTT = [];
for n = 1:rowNum %按单元分割填入交织矩阵
    [0121] indTT = 将新生成的单元填充到交织矩阵的下方;
end
seq = 按行读取交织矩阵;
seq = seq(seq>0); %完成速率匹配
[排序后序号,交织序列] = 升序排序(seq); %获取交织序列

```

[0122] 在方式1中所描述的写入规则下,将第一比特序列包括的L个比特写入交织矩阵后,交织矩阵中的任意一行中的写入的比特在第一比特序列中不连续,任意一列中写入的比特在第一比特序列中也不连续。

[0123] 需要说明的是,将第一比特序列包括的L个比特写入交织矩阵后,交织矩阵中所呈现出的特点实际上是本申请实施例提供的交织方法的一种外在表现。

[0124] 方式2

[0125] 将L个比特写入交织矩阵的至少两个交织单元,该至少两个交织单元分别包括不同的行数,且每个交织单元包括R列。

[0126] 参见图8,图8是写入规则的另一个示例。交织矩阵的第i个交织单元包括 B_i 行R列,写入第i个交织单元时,每列写入 B_i 个比特,其中, B_i 为正整数。

[0127] 例如,图8中示出了3个交织单元,第1个交织单元包括4行R列,第2个交织单元包括8行R列,第3个交织单元包括16行R列。即, $B_1=4, B_2=8, B_3=16$ 。

[0128] 在本实施例中,交织矩阵的总行数 $\sum_i B_i$ 满足关系式 $\sum_i B_i \leq C$,C为星座点数。

[0129] 列数R可以根据速率匹配之前的母码长度N计算,也或者可以根据速率匹配之后的码长M计算。类似地,如果是速率匹配之前的码长N,则 $R=\lceil N/C \rceil$,如果是速率匹配之后的码长M,则 $R=\lceil M/C \rceil$ 。

[0130] 同样地,如果L个比特已经全部写入交织矩阵,而交织矩阵还未填满,则将没有写入比特的位置写入NULL。

[0131] 可以看到,在方式2中,交织矩阵中存在至少两种深度的交织单元。按照方式2所描述的写入规则,将第一比特序列包括的L个比特写入交织矩阵后,交织矩阵中的任意一行中

写入的比特在第一比特序列中不连续,交织矩阵的任意一行中写入的比特在第一比特序列中也不连续。

[0132] 方式3

[0133] 将L个比特写入交织矩阵的第一交织单元和第二交织单元,交织矩阵中先排列第一交织单元,再排列第二交织单元。或者,也可以第一交织单元和第二交织单元交叉排列。

[0134] 这里所说的交叉排列包括第一交织单元和第二交织单元交替排列。或者,先排列 m_1 个第一交织单元,再排列 n_1 个第二交织单元。接着,再排列 m_2 个第一交织单元,再排列 n_2 个第二交织单元等。其中, m_1 、 m_2 、 n_1 和 n_2 均为正整数,各自的取值不作限定。

[0135] 参见图9,图9示出了两种交织深度的交织单元在交织矩阵中的一种排列形式。第一交织单元的交织深度记作 B_1 ,第二交织单元的交织深度记作 B_2 。其中, B_1 和 B_2 为正整数。图9中所示的排列方式中,交织矩阵中首先排列的是 U_1 个交织第一交织单元,之后排列的是 U_2 个第二交织单元,其中, U_1 和 U_2 为正整数。

[0136] 写入第一交织单元和第二交织单元的比特个数近似相等,即 $U_1 \times B_1 \times R = U_2 \times B_2 \times R$,其中,R为交织矩阵的列数。作为示例,两种交织深度可以取 $B_1 = 5, B_2 = 11$ 。或者 $B_1 = 3, B_2 = 5$ 或者 $B_1 = 3, B_2 = 7$ 或者 $B_1 = 3, B_2 = 11$ 或者 $B_1 = 5, B_2 = 7$ 或者 $B_1 = 7, B_2 = 11$ 。

[0137] 其中,一种确定 U_1 和 U_2 的方法为: $U_1 = B_2, U_2 = B_1$ 。此时, $R = \lceil M / (U_1 \cdot B_1 + U_2 \cdot B_2) \rceil$ 。

[0138] 分配给第一交织单元和第二交织单元的待交织比特的数量可以通过如下方法确定:分配给第一交织单元的比特长度 $M_1 = \lceil M / 2 \rceil$,分配给第二交织单元的比特长度 $M_2 = M - M_1$ 。或者,分配给第一交织单元的比特长度 $M_1 = \lceil M / 2 \rceil + 1$,分配给第二交织单元的比特长度 $M_2 = M - M_1$ 。

[0139] 交织过程可以表述为:

[0140] 将待交织比特 $[1, 2, 3, 4, \dots, M]$ 分为两组,第一组为 $[1, 2, 3, 4, \dots, M_1]$,第二组为 $[M_1 + 1, M_1 + 2, \dots, M_2]$ 。

[0141] 将第一组待交织比特逐列写入第1个第一交织单元,每一次写入的深度为 B_1 (级,每一列写入 B_1 个比特),每写完一列换列,当写满R列后跳转到第2个第一交织单元进行写入,重复这个过程直至将该 M_1 个比特全部写入。此时,如果 U_1 个第一交织单元还有未填满部分,则在未写入比特的位置的写入NULL,以将 U_1 个第一交织单元填满。

[0142] 将第二组待交织比特逐列写入第1个第二交织单元,每一次写入的深度为 B_2 ,每写完一列换列,当写满R列后跳转到第2个第二交织单元进行写入。重复这个过程直至将该 M_2 个比特全部写入。此时,如果第二交织单元还未填满,则在未写入比特的位置写入NULL,以将 U_2 个第二交织单元填满。

[0143] 可选地,在读取之前,可以进行交织器的行变换。例如,按照比特逆序的方法进行变换,或者按照奇偶顺序进行交换等。

[0144] 读取时,逐行读取交织矩阵中的比特,作为交织序列。

[0145] 或者,读取时也可以从不同的交织单元同时开始读取。例如,从第一交织单元的起点从左向右逐行向下读取。同时从第二交织单元的终点从右向左逐行向上读取。读取时遇到NULL跳过。从第一交织单元和第二交织单元读出的比特序列经过并串转换后作为最后输出的交织序列。

[0146] 按照方式3描述的写入规则将第一比特序列包括的L个比特写入交织矩阵后,交织矩阵的任意一行中写入的比特在第一比特序列中不连续,任意一列中写入的比特在第一比特序列中也不连续。

[0147] 方式4

[0148] 将L个比特写入交织矩阵的G种交织深度的交织单元,其中,交织矩阵中存在交织深度为 B_1, B_2, \dots, B_G 的交织单元各一个, B_1, B_2, \dots, B_G 均为正整数。

[0149] 在本实施例中,交织矩阵的列数 $R = \left\lceil M / \left(\sum_{i=1}^G B_i \right) \right\rceil$ 。其中,分配给交织深度为 B_i 的交织单元的比特个数 $M_i = R \times B_i$ 。当 $i = G$ 时,即交织深度为 B_G 的最后一个交织单元的比特个数 $M_G = M - \sum_{i=1}^{G-1} M_i$,B的取值可以为3,5,7,9,11等。

[0150] 为了便于理解,下面以 $G=2$,即交织矩阵中存在两种交织深度的两种交织单元为例,对本实施例的交织过程进行举例说明。

[0151] 方式4描述的写入规则可以参见图10,图10示出了两种交织深度的交织单元在交织矩阵中的另一种排列形式。如图10所示,交织矩阵中包括第一交织单元和第二交织单元,以下将第一交织单元的交织深度记作 B_1 ,第二交织单元的交织深度记作 B_2 。其中, B_1 和 B_2 为正整数。例如, $B_1=5, B_2=11$ 。如图10所示,交织矩阵中可以先排列第一交织单元,再排列第二交织单元。

[0152] 在本实施例中,交织矩阵的总列数R可以通过表达式 $R = \lceil M / (B_1 + B_2) \rceil$ 确定。其中,分配给第一交织单元的比特个数 $M_1 = R \times B_1$,分配给第二交织单元的比特个数 $M_2 = M - M_1$ 。

[0153] 具体地,交织过程可以表述为:

[0154] 将待交织比特 $[1, 2, 3, 4, \dots, M]$ 分为两组,第一组为 $[1, 2, 3, 4, \dots, M_1]$,第二组为 $[M_1+1, M_1+2, \dots, M_2]$ 。将第一组待交织比特逐列写入第一交织单元,每一次写入的深度为 B_1 (即,每一列写入 B_1 个比特),每写完一列换列,重复这个过程直至将该 M_1 个比特全部写入第一交织单元。此时,如果第一交织单元还有未填满部分,则在未写入比特的位置的写入NULL。

[0155] 将第二组待交织比特逐列写入第二交织单元,每一次写入的深度为 B_2 ,每写完一列换列。重复这个过程直至将该 M_2 个比特全部写入第二交织单元。此时,如果第二交织单元还未填满,则在未写入比特的位置写入NULL。

[0156] 读取时,逐行读取交织矩阵。或者,可以从不同的交织单元同时逐行读取,并将读取的比特做并串转换后输出,作为交织序列(即,第二比特序列)。

[0157] 逐行读取时,逐行读取的方向可以不同,可以一个从左向右读取、换行时是从上向下,另外一个从右向左读取,换行时从下向上。

[0158] 与方式3类似,在读取之前,可以对交织矩阵进行行变换。比如按照比特逆序的方法进行行交换,或者按照奇偶顺序交换的方式进行行交换。

[0159] 方式4中描述的写入规则和读取规则,可以对实现各个交织深度的交织单元并行写入和读出,可以提高交织速度。

[0160] 可以理解的是,方式4也可以按照“行进列出”的方式进行交织。参见图11,图11是

将第一比特序列按行写入两种交织深度的交织单元的示意图。

[0161] 按行写入的过程可以表述为：

[0162] 将待交织比特 $[1, 2, 3, 4, \dots, M]$ 分为两组，第一组为 $[1, 2, 3, 4, \dots, M_1]$ ，第二组为 $[M_1 + 1, M_1 + 2, \dots, M_2]$ 。

[0163] 将第一组待交织比特逐行写入第一交织单元，每一次写入的深度为 B_1 ，重复该过程直到完成该 M_1 个第一组待交织比特的写入。如果第一交织单元存在未填满的部分，则在为写入比特的位置补入NULL。

[0164] 将第二组待交织比特逐行写入第二交织单元，每一次写入的深度为 B_2 ，重复该过程直到完成 M_2 个第二组待交织比特的写入。如果第二交织单元存在未填满的部分，则在为写入比特的位置补填NULL。

[0165] 读取方式参见图12所示，图12为读取规则的一个示例。从左至右逐列读出。

[0166] 读取时，可以采用分别从两个不同交织深度的交织单元并行读取。其中一种实现方式如图13所示，图13是不同交织深度的交织单元的并行读取方式的示例。如图13所示，从第一交织单元的起点从上向下逐列向右读取，同时从第二交织单元的起点从下向上逐列向左读取。读取时跳过NULL。读取出的数据进行并串转换后作为输出的交织序列。

[0167] 可选地，在读取交织器之前，也可以对交织器执行列变换。列变换包括奇偶变换、比特逆序变换等。

[0168] 按照方式4中描述的写入规则，将第一比特序列包括的 L 个比特写入交织矩阵后，交织矩阵的任意一行中写入的比特在第一比特序列中不连续，且任意一列中写入的比特在第一比特序列中不连续。

[0169] 方式5

[0170] 将 L 个比特逐列写入交织矩阵，每一列写入的比特数都不相等。

[0171] 参见图14，图14是写入规则的再一个示例。如图14所示，将 L 个比特逐列写入交织矩阵，第一列写入2个比特，第二列写入4个比特，第三列写入8个比特，第四次写入16个比特，第五列写入32个比特。

[0172] 在本实施例中，交织矩阵的总行数为写入比特数最多的一列写入的比特数。例如，将 L 个写入交织矩阵时，第一列写入 B_1 个比特，第二列写入 B_2 个比特 \dots ，最后一列写入 B_c 个比特，则交织矩阵的行数就等于 B_c 。从另一个角度而言， $B_1 + B_2 + \dots + B_c = L$ 。其中， B_1, B_2, \dots, B_c 均为正整数。

[0173] 方式5中，交织矩阵的列数的确定方法为：

[0174] 如果写入交织矩阵的序列长度为速率匹配之后的长度 M ，则列数 $R = \min_C \left(\sum_{i=1}^C B_i \geq M \right)$ ；

[0175] 如果写入交织矩阵的序列长度是速率匹配之前的长度 N ，则列数 $R = \min_C \left(\sum_{i=1}^C B_i \geq N \right)$ 。

[0176] 在本实施例中，写入每列的比特数不相等，而写入交织矩阵的每一列的比特数可以取奇数序列、偶数序列或质数序列。

[0177] 按照方式5描述的写入规则将第一比特序列包括的 L 个比特写入交织矩阵后，交织矩阵中的至少一行中写入的比特在第一比特序列中不连续，且至少有两列中写入的比特数

不相等。

[0178] 方式6

[0179] 将L个比特分n轮写入交织矩阵,每一轮写入交织矩阵的至少一列,每一轮中写入各列的比特数不相等。

[0180] 参见图15,图15是写入规则的又一个示例。如图15所示,第1轮写入交织矩阵的5个列,第1列至第5列分别写入的比特数为2、4、8、16、32。第2轮写入交织矩阵的第1列至第4列,每列分别写入的比特数为2、4、8、16。第3轮写入交织矩阵的第1列至第3列,每列写入的比特数为4、8和16。第4轮写入交织矩阵的第1列和第2列,每列分别写入的比特数为8和16。第5轮写入交织矩阵的第1列,写入的比特数为16。

[0181] 在本实施例中,一个列中最多能够写入的比特数 $B_{\max} = 2^{\lceil \log_2 M \rceil - 1}$,交织矩阵的总列数 $R = \log_2 B_{\max}$ 。

[0182] 交织矩阵的行数C的确定方法为:

[0183] 如果写入交织矩阵的序列长度为速率匹配之后的长度M,则 $C = \max(B_{\max}, \lceil M / R \rceil)$;

[0184] 如果写入交织矩阵的序列长度为速率匹配之前的长度N,则 $C = \max(B_{\max}, \lceil M / R \rceil)$ 。

[0185] 按照方式6的写入规则将第一比特序列包括的L个比特写入交织矩阵后,交织矩阵中至少一行中写入的比特在第一比特序列中不连续,且至少有一列中写入的比特在第一比特序列中不连续。

[0186] 方式7

[0187] 将L个比特逐列写入交织矩阵,至少有两列写入的比特数不相等,且每列写入的比特数均为2的指数幂。

[0188] 参见图16,图16是写入规则的又一个示例。以L=128为例,第1列至第5列写入的比特数分别为2、2、4、8、16、32、32。

[0189] 在本实施例中,交织矩阵的总行数等于写入比特数最多的一列写入的比特个数,具体可以参见方式5中对确定交织矩阵的行数所作的说明。

[0190] 列数的确定方法如下:

[0191] 如果写入的序列长度为速率匹配之后的长度M,则列数 $R = \lceil \log_2(M) \rceil + 1$;如果写入的序列长度为速率匹配之前的长度N,则列数 $R = \lceil \log_2(N) \rceil + 1$ 。

[0192] 按照方式7描述的写入规则将第一比特序列包括的L个比特写入交织矩阵后,交织矩阵的至少一行中写入的比特在第一比特序列中不连续,且至少有两列中写入的比特数不相等。

[0193] 方式8

[0194] 将L个比特逐列写入交织矩阵,每列写入的比特个数不完全相等,写入交织矩阵各列的比特数可以为奇数序列或质数序列。

[0195] 在本实施例中,交织矩阵的总行数等于星座点数C,列数R的确定方法如下:

[0196] 如果写入的序列长度为速率匹配之后的长度M,则满足 $\sum_{i=1}^R B_i = M$;如果写入的序

列长度为速率匹配之前的长度 N ,则满足 $\sum_{i=1}^R B_i = N$ 。

[0197] 按照方式8描述的写入规则将第一比特序列包括的 L 个比特写入交织矩阵中,交织矩阵的至少一行中写入的比特在第一比特序列中不连续,且至少有两列中写入的比特数不相等。

[0198] 方式9

[0199] 将 L 个比特写入交织矩阵时,任意两次的写入方向可以相同或不同。

[0200] 其中,在按列写入的情况下,每向一个交织单元写入一行,认为是一次写入。在按行写入的情况下,每向一个交织单元写入一行认为是一次写入。

[0201] 参见图17,图17是写入方向的示意图。在图14中, B_1 、 B_2 和 B_3 对应的交织单元分别占据交织矩阵的4行8列。将 L 个比特写入交织矩阵时, B_1 对应的交织单元的每列的写入方向均是从上向下, B_2 对应的交织单元的每列的写入方向均是从下往上, B_3 对应的交织单元的每列的写入方向均是从上往下。其中, B_1 、 B_2 和 B_3 均为正整数。

[0202] 图17所示的写入方向仅是作为一个示例。可以理解的是,在以上描述的各种写入规则中,任意两次的写入方向可以是相同的,或者不同。

[0203] 方式9描述了将第一比特序列的 L 个比特写入交织矩阵时,不论是按行写入或是按列写入,任意两次的写入方向可以相同或者不同。通过这种方式,交织前后的比特序列的相关性进一步减弱,可以提高交织的性能。

[0204] 方式10

[0205] 将 L 个比特写入 Z 个交织单元分别进行交织,该 Z 个交织单元的输出合并得到第二比特序列。

[0206] 将待交织比特分为 Z 段, Z 段比特分别进入 Z 个不同交织深度的交织单元进行交织,随后将各个交织单元的输出进行合并,得到交织序列(即,第二比特序列)。

[0207] 其中,合并方法包括并串转换(Parallel-In,Serial-Out,PISO)寄存器、级联数据奇偶交叉、级联数据比特逆序交叉等。常用的交织深度包括5,7,9,11等。

[0208] 以 $Z=2$ 举例说明方式10的交织过程。以下,将这两种交织深度不同的交织单元分别记作第一交织单元和第二交织单元,交织深度分别记作 B_1 和 B_2 ,将这两段比特的长度分别记作 M_1 和 M_2 。 B_1 、 B_2 、 M_1 、 M_2 均为正整数。

[0209] 其中, $M_1 = \lceil M/2 \rceil$, $M_2 = M - M_1$ 。或者, $M_1 = \lceil M/2 \rceil + 1$, $M_2 = M - M_1$ 。第一交织单元的行数 $C_1 = \lceil M_1 / B_1 \rceil$,第二交织单元的行数 $C_2 = \lceil M_2 / B_2 \rceil$ 。

[0210] 其中,两个交织器的深度 B_1 、 B_2 可以分别为5、11或者5、7或者7、11。

[0211] 具体地,交织过程可以表述为:

[0212] 将待交织比特 $[1, 2, 3, 4, \dots, M]$ 分为两组,第一组为 $[1, 2, 3, 4, \dots, M_1]$,第二组为 $[M_1+1, M_1+2, \dots, M_2]$ 。第一组待交织比特逐行写入第一交织单元,未填满则填充NULL。第二组待交织比特逐行写入第二交织单元,未填满则填充NULL。

[0213] 从两个交织单元中分别逐列读出比特,随后两组读出的比特进行合并。读取过程中遇到NULL则跳过。合并的方式可以为:

[0214] (1) 将两组读出的比特拼接成 M 长的比特序列,然后通过序号比特逆序的方式完成

重排合并;或者

[0215] (2) 将两个交织单元每次交替输出1个比特,直至两个交织单元均完成输出;

[0216] (3) 直接采用PISO并串转换完成。

[0217] 按照方式10描述的写入规则将第一比特序列包括的L个比特写入交织矩阵中,交织矩阵的至少一行且至少一列中写入的比特在第一比特序列中不连续。

[0218] 方式11

[0219] 将L个比特分为n段,n段比特分别写入n个交织单元。随后将n个交织单元按照列方向进行拼接,最后读取按照列方向拼接的多个矩形拼接块。其中,n个交织单元的交织深度互不相同。

[0220] 以n=2作为示例说明本实施例。即,将待交织比特分别写入两个交织单元进行交织。以下,将这两种交织深度不同的交织单元分别记作第一交织单元和第二交织单元。

[0221] 交织过程可以表述为:

[0222] 将待交织比特分为2段,2段比特分别逐行写入第一交织单元和第二交织单元,每个交织单元内的未填满部分写入NULL。

[0223] 其中,两段比特的长度可以分别为 $M_1 = \lceil M/2 \rceil$, $M_2 = M - M_1$ 。或者, $M_1 = \lceil M/2 \rceil + 1$, $M_2 = M - M_1$ 。第一交织单元的行数 $C_1 = \lceil M_1 / B_1 \rceil$,第二交织单元的行数 $C_2 = \lceil M_2 / B_2 \rceil$ 。

[0224] 将两个交织单元按照列方向进行拼接。

[0225] 本实施例中的拼接过程可以参见图18,图18为两种交织深度的交织单元的拼接示意图。图18中,实线表示逐行写入,虚线表示逐列读出。

[0226] 按照方式11描述的写入规则,将第一比特序列写入不同交织深度的多个交织单元的拼接块,拼接块的至少一列中写入的比特在第一比特序列中不连续。

[0227] 可选地,读出前可以对交织矩阵进行行变换,比如比特逆序交换或者奇偶交换等。读取时遇到NULL跳过。

[0228] 可以理解的是,由于矩形拼接块的大小各不相同,因此,行变换可以仅对 $\min(R_1, R_2, \dots, R_n)$ 列进行。

[0229] 以上各实施例对如何将待交织的比特序列(即,第一比特序列)写入交织矩阵作了详细说明。下面说明本申请实施例中的读取规则,下面的读取规则适用于上述任意一种写入规则写入交织矩阵的实施例的读取。

[0230] 在本申请实施例中,读取规则也包括多种。例如:

[0231] (1) 从左至右逐行读取。

[0232] (2) 从右至左逐行读取。

[0233] (3) 奇数行从左至右读取,偶数行从右至左读取。

[0234] (4) 各行读取方向相同,但每行的读取起点不同。在这种情况下,每一行的读取起点可以根据一个读取函数确定,或者也可以由循环移位寄存器确定。

[0235] 此外,针对一些写入规则,为了方案描述上的清楚,将其适用的读取规则在各自的写入规则之后进行了说明。例如,方式3,方式4,方式10和方式11。换句话说,方式3,方式4,方式10和方式11除了适用这里所说的通用的读取规则,也可以针对各自的写入规则设计读取规则,以提升交织的性能。

[0236] 需要说明的是,在读取的过程中,如果写入交织矩阵的比特序列长度是速率匹配之前的长度N,则需要跳过NULL、打孔(Punctured)或者缩短(Shorten)的位置。如果写入交织矩阵的比特序列长度是速率匹配之后的长度M,则需要跳过NULL的位置。这对于上任意一种写入都是适用的。

[0237] 可选地,在本申请实施例中,将L个比特写入交织矩阵之后,还可以对交织矩阵进行列交换。列交换的方式可以有奇偶对换、比特逆序、固定列变换或者根据预定义的变换函数进行列变换等。

[0238] 例如,原始列序号的排序为[1 2 3 4 5 6 7 8],奇偶对换的结果为[2 1 4 3 6 5 8 7],比特逆序的结果为[1 5 3 7 2 6 4 8]。

[0239] 以上任意一种写入方式都可以在完成写入之后,进行列变换,再进行读取。

[0240] 下面给出根据本申请实施例的方法进行交织得到的交织序列,通过上述实施例的方式获得的交织序列不限于下面列举的序列。

[0241] 应理解,这里所说的交织序列即是按照读取规则,将写入交织矩阵的L个比特读出得到的第二比特序列。

[0242] 采用64QAM时,对于码长等于1024的极化码构造序列而言,其交织序列可以为如下所示的序列#1。

[0243] 其中,序列#1中的极化信道索引(或称作极化信道序号)从0开始。如果极化信道索引从1开始,则可以在序列#1的基础上整体加1得到。

[0244] 序列#1

[0245] [1023,1007,991,975,959,943,927,911,895,879,863,847,831,815,799,783,767,751,735,719,703,687,671,655,639,623,607,591,575,559,543,527,1015,999,983,967,951,935,919,903,887,871,855,839,823,807,791,775,759,743,727,711,695,679,663,647,631,615,599,583,567,551,535,519,1019,1003,987,971,955,939,923,907,891,875,859,843,827,811,795,779,763,747,731,715,699,683,667,651,635,619,603,587,571,555,539,523,1011,995,979,963,947,931,915,899,883,867,851,835,819,803,787,771,755,739,723,707,691,675,659,643,627,611,595,579,563,547,531,515,1021,1005,989,973,957,941,925,909,893,877,861,845,829,813,797,781,765,749,733,717,701,685,669,653,637,621,605,589,573,557,541,525,1013,997,981,965,949,933,917,901,885,869,853,837,821,805,789,773,757,741,725,709,693,677,661,645,629,613,597,581,565,549,533,517,1017,1001,985,969,953,937,921,905,889,873,857,841,825,809,793,777,761,745,729,713,697,681,665,649,633,617,601,585,569,553,537,521,1009,993,977,961,945,929,913,897,881,865,849,833,817,801,785,769,753,737,721,705,689,673,657,641,625,609,593,577,561,545,529,513,1022,1006,990,974,958,942,926,910,894,878,862,846,830,814,798,782,766,750,734,718,702,686,670,654,638,622,606,590,574,558,542,526,1014,998,982,966,950,934,918,902,886,870,854,838,822,806,790,774,758,742,726,710,694,678,662,646,630,614,598,582,566,550,534,518,1018,1002,986,970,954,938,922,906,890,874,858,842,826,810,794,778,762,746,730,714,698,682,666,650,634,618,602,586,570,554,538,522,1010,994,978,962,946,930,914,898,882,866,850,834,818,802,786,770,754,738,722,706,690,

674,658,642,626,610,594,578,562,546,530,514,1020,1004,988,972,956,940,924,908,892,876,860,844,828,812,796,780,764,748,732,716,700,684,668,652,636,620,604,588,572,556,540,524,1012,996,980,964,948,932,916,900,884,868,852,836,820,804,788,772,756,740,724,708,692,676,660,644,628,612,596,580,564,548,532,516,1016,1000,984,968,952,936,920,904,888,872,856,840,824,808,792,776,760,744,728,712,696,680,664,648,632,616,600,584,568,552,536,520,1008,992,976,960,944,928,912,896,880,864,848,832,816,800,784,768,752,736,720,704,688,672,656,640,624,608,592,576,560,544,528,512,511,495,479,463,447,431,415,399,383,367,351,335,319,303,287,271,255,239,223,207,191,175,159,143,127,111,95,79,63,47,31,15,503,487,471,455,439,423,407,391,375,359,343,327,311,295,279,263,247,231,215,199,183,167,151,135,119,103,87,71,55,39,23,7,507,491,475,459,443,427,411,395,379,363,347,331,315,299,283,267,251,235,219,203,187,171,155,139,123,107,91,75,59,43,27,11,499,483,467,451,435,419,403,387,371,355,339,323,307,291,275,259,243,227,211,195,179,163,147,131,115,99,83,67,51,35,19,3,509,493,477,461,445,429,413,397,381,365,349,333,317,301,285,269,253,237,221,205,189,173,157,141,125,109,93,77,61,45,29,13,501,485,469,453,437,421,405,389,373,357,341,325,309,293,277,261,245,229,213,197,181,165,149,133,117,101,85,69,53,37,21,5,505,489,473,457,441,425,409,393,377,361,345,329,313,297,281,265,249,233,217,201,185,169,153,137,121,105,89,73,57,41,25,9,497,481,465,449,433,417,401,385,369,353,337,321,305,289,273,257,241,225,209,193,177,161,145,129,113,97,81,65,49,33,17,1,510,494,478,462,446,430,414,398,382,366,350,334,318,302,286,270,254,238,222,206,190,174,158,142,126,110,94,78,62,46,30,14,502,486,470,454,438,422,406,390,374,358,342,326,310,294,278,262,246,230,214,198,182,166,150,134,118,102,86,70,54,38,22,6,506,490,474,458,442,426,410,394,378,362,346,330,314,298,282,266,250,234,218,202,186,170,154,138,122,106,90,74,58,42,26,10,498,482,466,450,434,418,402,386,370,354,338,322,306,290,274,258,242,226,210,194,178,162,146,130,114,98,82,66,50,34,18,2,508,492,476,460,444,428,412,396,380,364,348,332,316,300,284,268,252,236,220,204,188,172,156,140,124,108,92,76,60,44,28,12,500,484,468,452,436,420,404,388,372,356,340,324,308,292,276,260,244,228,212,196,180,164,148,132,116,100,84,68,52,36,20,4,504,488,472,456,440,424,408,392,376,360,344,328,312,296,280,264,248,232,216,200,184,168,152,136,120,104,88,72,56,40,24,8,496,480,464,448,432,416,400,384,368,352,336,320,304,288,272,256,240,224,208,192,176,160,144,128,112,96,80,64,48,32,16,0]

[0246] 序列#1左右对称,翻转后为相同的交织序列,例如如下所示的序列#2。

[0247] 序列#2

[0248] [0,16,32,48,64,80,96,112,128,144,160,176,192,208,224,240,256,272,288,304,320,336,352,368,384,400,416,432,448,464,480,496,8,24,40,56,72,88,104,120,136,152,168,184,200,216,232,248,264,280,296,312,328,344,360,376,392,408,424,440,456,472,488,504,4,20,36,52,68,84,100,116,132,148,164,180,196,212,228,244,

260, 276, 292, 308, 324, 340, 356, 372, 388, 404, 420, 436, 452, 468, 484, 500, 12, 28, 44, 60, 76, 92, 108, 124, 140, 156, 172, 188, 204, 220, 236, 252, 268, 284, 300, 316, 332, 348, 364, 380, 396, 412, 428, 444, 460, 476, 492, 508, 2, 18, 34, 50, 66, 82, 98, 114, 130, 146, 162, 178, 194, 210, 226, 242, 258, 274, 290, 306, 322, 338, 354, 370, 386, 402, 418, 434, 450, 466, 482, 498, 10, 26, 42, 58, 74, 90, 106, 122, 138, 154, 170, 186, 202, 218, 234, 250, 266, 282, 298, 314, 330, 346, 362, 378, 394, 410, 426, 442, 458, 474, 490, 506, 6, 22, 38, 54, 70, 86, 102, 118, 134, 150, 166, 182, 198, 214, 230, 246, 262, 278, 294, 310, 326, 342, 358, 374, 390, 406, 422, 438, 454, 470, 486, 502, 14, 30, 46, 62, 78, 94, 110, 126, 142, 158, 174, 190, 206, 222, 238, 254, 270, 286, 302, 318, 334, 350, 366, 382, 398, 414, 430, 446, 462, 478, 494, 510, 1, 17, 33, 49, 65, 81, 97, 113, 129, 145, 161, 177, 193, 209, 225, 241, 257, 273, 289, 305, 321, 337, 353, 369, 385, 401, 417, 433, 449, 465, 481, 497, 9, 25, 41, 57, 73, 89, 105, 121, 137, 153, 169, 185, 201, 217, 233, 249, 265, 281, 297, 313, 329, 345, 361, 377, 393, 409, 425, 441, 457, 473, 489, 505, 5, 21, 37, 53, 69, 85, 101, 117, 133, 149, 165, 181, 197, 213, 229, 245, 261, 277, 293, 309, 325, 341, 357, 373, 389, 405, 421, 437, 453, 469, 485, 501, 13, 29, 45, 61, 77, 93, 109, 125, 141, 157, 173, 189, 205, 221, 237, 253, 269, 285, 301, 317, 333, 349, 365, 381, 397, 413, 429, 445, 461, 477, 493, 509, 3, 19, 35, 51, 67, 83, 99, 115, 131, 147, 163, 179, 195, 211, 227, 243, 259, 275, 291, 307, 323, 339, 355, 371, 387, 403, 419, 435, 451, 467, 483, 499, 11, 27, 43, 59, 75, 91, 107, 123, 139, 155, 171, 187, 203, 219, 235, 251, 267, 283, 299, 315, 331, 347, 363, 379, 395, 411, 427, 443, 459, 475, 491, 507, 7, 23, 39, 55, 71, 87, 103, 119, 135, 151, 167, 183, 199, 215, 231, 247, 263, 279, 295, 311, 327, 343, 359, 375, 391, 407, 423, 439, 455, 471, 487, 503, 15, 31, 47, 63, 79, 95, 111, 127, 143, 159, 175, 191, 207, 223, 239, 255, 271, 287, 303, 319, 335, 351, 367, 383, 399, 415, 431, 447, 463, 479, 495, 511, 512, 528, 544, 560, 576, 592, 608, 624, 640, 656, 672, 688, 704, 720, 736, 752, 768, 784, 800, 816, 832, 848, 864, 880, 896, 912, 928, 944, 960, 976, 992, 1008, 520, 536, 552, 568, 584, 600, 616, 632, 648, 664, 680, 696, 712, 728, 744, 760, 776, 792, 808, 824, 840, 856, 872, 888, 904, 920, 936, 952, 968, 984, 1000, 1016, 516, 532, 548, 564, 580, 596, 612, 628, 644, 660, 676, 692, 708, 724, 740, 756, 772, 788, 804, 820, 836, 852, 868, 884, 900, 916, 932, 948, 964, 980, 996, 1012, 524, 540, 556, 572, 588, 604, 620, 636, 652, 668, 684, 700, 716, 732, 748, 764, 780, 796, 812, 828, 844, 860, 876, 892, 908, 924, 940, 956, 972, 988, 1004, 1020, 514, 530, 546, 562, 578, 594, 610, 626, 642, 658, 674, 690, 706, 722, 738, 754, 770, 786, 802, 818, 834, 850, 866, 882, 898, 914, 930, 946, 962, 978, 994, 1010, 522, 538, 554, 570, 586, 602, 618, 634, 650, 666, 682, 698, 714, 730, 746, 762, 778, 794, 810, 826, 842, 858, 874, 890, 906, 922, 938, 954, 970, 986, 1002, 1018, 518, 534, 550, 566, 582, 598, 614, 630, 646, 662, 678, 694, 710, 726, 742, 758, 774, 790, 806, 822, 838, 854, 870, 886, 902, 918, 934, 950, 966, 982, 998, 1014, 526, 542, 558, 574, 590, 606, 622, 638, 654, 670, 686, 702, 718, 734, 750, 766, 782, 798, 814, 830, 846, 862, 878, 894, 910, 926, 942, 958, 974, 990, 1006, 1022, 513, 529, 545, 561, 577, 593, 609, 625, 641, 657, 673, 689, 705, 721, 737, 753, 769, 785, 801, 817, 833, 849, 865, 881, 897, 913, 929, 945, 961, 977, 993, 1009, 521, 537, 553, 569, 585, 601, 617, 633, 649, 665, 681, 697, 713, 729, 745, 761, 777, 793, 809, 825, 841, 857, 873, 889, 905, 921, 937, 953, 969, 985, 1001, 1017, 517, 533, 549, 565, 581,

597,613,629,645,661,677,693,709,725,741,757,773,789,805,821,837,853,869,885,901,917,933,949,965,981,997,1013,525,541,557,573,589,605,621,637,653,669,685,701,717,733,749,765,781,797,813,829,845,861,877,893,909,925,941,957,973,989,1005,1021,515,531,547,563,579,595,611,627,643,659,675,691,707,723,739,755,771,787,803,819,835,851,867,883,899,915,931,947,963,979,995,1011,523,539,555,571,587,603,619,635,651,667,683,699,715,731,747,763,779,795,811,827,843,859,875,891,907,923,939,955,971,987,1003,1019,519,535,551,567,583,599,615,631,647,663,679,695,711,727,743,759,775,791,807,823,839,855,871,887,903,919,935,951,967,983,999,1015,527,543,559,575,591,607,623,639,655,671,687,703,719,735,751,767,783,799,815,831,847,863,879,895,911,927,943,959,975,991,1007,1023]

[0249] 对于任意母码长度的Polar码构造序列的交织序列,都可以从上述序列#1或序列#2中直接读取极化信道序号小于(极化信道索引从1开始时,取小于或等于)所需长度的索引即可。

[0250] 例如,母码长度 $N=512$ 的构造序列的交织序列如下面的序列#3所示。

[0251] 序列#3

[0252] [511,495,479,463,447,431,415,399,383,367,351,335,319,303,287,271,255,239,223,207,191,175,159,143,127,111,95,79,63,47,31,15,503,487,471,455,439,423,407,391,375,359,343,327,311,295,279,263,247,231,215,199,183,167,151,135,119,103,87,71,55,39,23,7,507,491,475,459,443,427,411,395,379,363,347,331,315,299,283,267,251,235,219,203,187,171,155,139,123,107,91,75,59,43,27,11,499,483,467,451,435,419,403,387,371,355,339,323,307,291,275,259,243,227,211,195,179,163,147,131,115,99,83,67,51,35,19,3,509,493,477,461,445,429,413,397,381,365,349,333,317,301,285,269,253,237,221,205,189,173,157,141,125,109,93,77,61,45,29,13,501,485,469,453,437,421,405,389,373,357,341,325,309,293,277,261,245,229,213,197,181,165,149,133,117,101,85,69,53,37,21,5,505,489,473,457,441,425,409,393,377,361,345,329,313,297,281,265,249,233,217,201,185,169,153,137,121,105,89,73,57,41,25,9,497,481,465,449,433,417,401,385,369,353,337,321,305,289,273,257,241,225,209,193,177,161,145,129,113,97,81,65,49,33,17,1,510,494,478,462,446,430,414,398,382,366,350,334,318,302,286,270,254,238,222,206,190,174,158,142,126,110,94,78,62,46,30,14,502,486,470,454,438,422,406,390,374,358,342,326,310,294,278,262,246,230,214,198,182,166,150,134,118,102,86,70,54,38,22,6,506,490,474,458,442,426,410,394,378,362,346,330,314,298,282,266,250,234,218,202,186,170,154,138,122,106,90,74,58,42,26,10,498,482,466,450,434,418,402,386,370,354,338,322,306,290,274,258,242,226,210,194,178,162,146,130,114,98,82,66,50,34,18,2,508,492,476,460,444,428,412,396,380,364,348,332,316,300,284,268,252,236,220,204,188,172,156,140,124,108,92,76,60,44,28,12,500,484,468,452,436,420,404,388,372,356,340,324,308,292,276,260,244,228,212,196,180,164,148,132,116,100,84,68,52,36,20,4,504,488,472,456,440,424,408,392,376,360,344,328,312,296,280,264,248,232,216,200,184,168,152,136,120,104,88,72,56,40,

24, 8, 496, 480, 464, 448, 432, 416, 400, 384, 368, 352, 336, 320, 304, 288, 272, 256, 240, 224, 208, 192, 176, 160, 144, 128, 112, 96, 80, 64, 48, 32, 16, 0]

[0253] 对于其它母码长度的交织序列可以采取相同方法获得,这里不作一一赘述。

[0254] 在获取到交织序列后,当编码长度不等于母码长度时,从交织序列中剔除需要打孔或者截短的极化信道序号即可。

[0255] 采用16QAM时,对于码长等于1024的极化码构造序列而言,其交织序列可以为如下所示的序列#4。

[0256] 序列#4

[0257] [1023, 959, 895, 831, 767, 703, 639, 575, 991, 927, 863, 799, 735, 671, 607, 543, 1007, 943, 879, 815, 751, 687, 623, 559, 975, 911, 847, 783, 719, 655, 591, 527, 1015, 951, 887, 823, 759, 695, 631, 567, 983, 919, 855, 791, 727, 663, 599, 535, 999, 935, 871, 807, 743, 679, 615, 551, 967, 903, 839, 775, 711, 647, 583, 519, 1019, 955, 891, 827, 763, 699, 635, 571, 987, 923, 859, 795, 731, 667, 603, 539, 1003, 939, 875, 811, 747, 683, 619, 555, 971, 907, 843, 779, 715, 651, 587, 523, 1011, 947, 883, 819, 755, 691, 627, 563, 979, 915, 851, 787, 723, 659, 595, 531, 995, 931, 867, 803, 739, 675, 611, 547, 963, 899, 835, 771, 707, 643, 579, 515, 1021, 957, 893, 829, 765, 701, 637, 573, 989, 925, 861, 797, 733, 669, 605, 541, 1005, 941, 877, 813, 749, 685, 621, 557, 973, 909, 845, 781, 717, 653, 589, 525, 1013, 949, 885, 821, 757, 693, 629, 565, 981, 917, 853, 789, 725, 661, 597, 533, 997, 933, 869, 805, 741, 677, 613, 549, 965, 901, 837, 773, 709, 645, 581, 517, 1017, 953, 889, 825, 761, 697, 633, 569, 985, 921, 857, 793, 729, 665, 601, 537, 1001, 937, 873, 809, 745, 681, 617, 553, 969, 905, 841, 777, 713, 649, 585, 521, 1009, 945, 881, 817, 753, 689, 625, 561, 977, 913, 849, 785, 721, 657, 593, 529, 993, 929, 865, 801, 737, 673, 609, 545, 961, 897, 833, 769, 705, 641, 577, 513, 1022, 958, 894, 830, 766, 702, 638, 574, 990, 926, 862, 798, 734, 670, 606, 542, 1006, 942, 878, 814, 750, 686, 622, 558, 974, 910, 846, 782, 718, 654, 590, 526, 1014, 950, 886, 822, 758, 694, 630, 566, 982, 918, 854, 790, 726, 662, 598, 534, 998, 934, 870, 806, 742, 678, 614, 550, 966, 902, 838, 774, 710, 646, 582, 518, 1018, 954, 890, 826, 762, 698, 634, 570, 986, 922, 858, 794, 730, 666, 602, 538, 1002, 938, 874, 810, 746, 682, 618, 554, 970, 906, 842, 778, 714, 650, 586, 522, 1010, 946, 882, 818, 754, 690, 626, 562, 978, 914, 850, 786, 722, 658, 594, 530, 994, 930, 866, 802, 738, 674, 610, 546, 962, 898, 834, 770, 706, 642, 578, 514, 1020, 956, 892, 828, 764, 700, 636, 572, 988, 924, 860, 796, 732, 668, 604, 540, 1004, 940, 876, 812, 748, 684, 620, 556, 972, 908, 844, 780, 716, 652, 588, 524, 1012, 948, 884, 820, 756, 692, 628, 564, 980, 916, 852, 788, 724, 660, 596, 532, 996, 932, 868, 804, 740, 676, 612, 548, 964, 900, 836, 772, 708, 644, 580, 516, 1016, 952, 888, 824, 760, 696, 632, 568, 984, 920, 856, 792, 728, 664, 600, 536, 1000, 936, 872, 808, 744, 680, 616, 552, 968, 904, 840, 776, 712, 648, 584, 520, 1008, 944, 880, 816, 752, 688, 624, 560, 976, 912, 848, 784, 720, 656, 592, 528, 992, 928, 864, 800, 736, 672, 608, 544, 960, 896, 832, 768, 704, 640, 576, 512, 511, 447, 383, 319, 255, 191, 127, 63, 479, 415, 351, 287, 223, 159, 95, 31, 495, 431, 367, 303, 239, 175, 111, 47, 463, 399, 335, 271, 207, 143, 79, 15, 503, 439, 375, 311, 247, 183, 119, 55, 471, 407, 343, 279, 215, 151, 87, 23, 487, 423, 359, 295, 231, 167, 103, 39, 455, 391, 327, 263, 199, 135, 71, 7, 507, 443, 379, 315, 251, 187, 123, 59, 475, 411, 347, 283,

219,155,91,27,491,427,363,299,235,171,107,43,459,395,331,267,203,139,75,11,499,435,371,307,243,179,115,51,467,403,339,275,211,147,83,19,483,419,355,291,227,163,99,35,451,387,323,259,195,131,67,3,509,445,381,317,253,189,125,61,477,413,349,285,221,157,93,29,493,429,365,301,237,173,109,45,461,397,333,269,205,141,77,13,501,437,373,309,245,181,117,53,469,405,341,277,213,149,85,21,485,421,357,293,229,165,101,37,453,389,325,261,197,133,69,5,505,441,377,313,249,185,121,57,473,409,345,281,217,153,89,25,489,425,361,297,233,169,105,41,457,393,329,265,201,137,73,9,497,433,369,305,241,177,113,49,465,401,337,273,209,145,81,17,481,417,353,289,225,161,97,33,449,385,321,257,193,129,65,1,510,446,382,318,254,190,126,62,478,414,350,286,222,158,94,30,494,430,366,302,238,174,110,46,462,398,334,270,206,142,78,14,502,438,374,310,246,182,118,54,470,406,342,278,214,150,86,22,486,422,358,294,230,166,102,38,454,390,326,262,198,134,70,6,506,442,378,314,250,186,122,58,474,410,346,282,218,154,90,26,490,426,362,298,234,170,106,42,458,394,330,266,202,138,74,10,498,434,370,306,242,178,114,50,466,402,338,274,210,146,82,18,482,418,354,290,226,162,98,34,450,386,322,258,194,130,66,2,508,444,380,316,252,188,124,60,476,412,348,284,220,156,92,28,492,428,364,300,236,172,108,44,460,396,332,268,204,140,76,12,500,436,372,308,244,180,116,52,468,404,340,276,212,148,84,20,484,420,356,292,228,164,100,36,452,388,324,260,196,132,68,4,504,440,376,312,248,184,120,56,472,408,344,280,216,152,88,24,488,424,360,296,232,168,104,40,456,392,328,264,200,136,72,8,496,432,368,304,240,176,112,48,464,400,336,272,208,144,80,16,480,416,352,288,224,160,96,32,448,384,320,256,192,128,64,0]

[0258] 与采用64QAM时类似,任意母码长度的构造序列的交织序列都可以从长度序列#4中读取。这里不在一一赘述。

[0259] 下面再给出长度 $N=1024$ 的构造序列的交织序列的一种可能,如下面的序列#5所示。

[0260] 序列#5

[0261] [1022,1019,1016,1013,1010,1007,1004,1001,998,995,992,989,986,983,980,977,974,971,968,965,962,959,956,953,950,947,944,941,938,935,932,929,926,923,920,917,914,911,908,905,902,899,896,893,890,887,884,881,878,875,872,869,866,863,860,857,854,851,848,845,842,839,836,833,830,827,824,821,818,815,812,809,806,803,800,797,794,791,788,785,782,779,776,773,770,767,764,761,758,755,752,749,746,743,740,737,734,731,728,725,722,719,716,713,710,707,704,701,698,695,692,689,686,683,680,677,674,671,668,665,662,659,656,653,650,647,644,641,637,633,629,625,621,617,613,609,605,601,597,593,589,585,581,577,573,569,565,561,557,553,549,545,541,537,533,529,525,521,517,513,509,505,501,497,493,489,485,481,477,473,469,465,461,457,453,449,445,441,437,433,429,425,421,417,413,409,405,401,397,393,389,385,380,375,370,365,360,355,350,345,340,335,330,325,320,315,310,305,300,295,290,285,280,275,270,265,260,255,250,245,240,235,230,225,

219, 213, 207, 201, 195, 189, 183, 177, 171, 165, 159, 153, 147, 141, 135, 129, 123, 116, 109, 102, 95, 88, 81, 74, 66, 58, 50, 42, 34, 25, 15, 4, 1023, 1020, 1017, 1014, 1011, 1008, 1005, 1002, 999, 996, 993, 990, 987, 984, 981, 978, 975, 972, 969, 966, 963, 960, 957, 954, 951, 948, 945, 942, 939, 936, 933, 930, 927, 924, 921, 918, 915, 912, 909, 906, 903, 900, 897, 894, 891, 888, 885, 882, 879, 876, 873, 870, 867, 864, 861, 858, 855, 852, 849, 846, 843, 840, 837, 834, 831, 828, 825, 822, 819, 816, 813, 810, 807, 804, 801, 798, 795, 792, 789, 786, 783, 780, 777, 774, 771, 768, 765, 762, 759, 756, 753, 750, 747, 744, 741, 738, 735, 732, 729, 726, 723, 720, 717, 714, 711, 708, 705, 702, 699, 696, 693, 690, 687, 684, 681, 678, 675, 672, 669, 666, 663, 660, 657, 654, 651, 648, 645, 642, 638, 634, 630, 626, 622, 618, 614, 610, 606, 602, 598, 594, 590, 586, 582, 578, 574, 570, 566, 562, 558, 554, 550, 546, 542, 538, 534, 530, 526, 522, 518, 514, 510, 506, 502, 498, 494, 490, 486, 482, 478, 474, 470, 466, 462, 458, 454, 450, 446, 442, 438, 434, 430, 426, 422, 418, 414, 410, 406, 402, 398, 394, 390, 386, 382, 377, 372, 367, 362, 357, 352, 347, 342, 337, 332, 327, 322, 317, 312, 307, 302, 297, 292, 287, 282, 277, 272, 267, 262, 257, 252, 247, 242, 237, 232, 227, 221, 215, 209, 203, 197, 191, 185, 179, 173, 167, 161, 155, 149, 143, 137, 131, 125, 118, 111, 104, 97, 90, 83, 76, 68, 60, 52, 44, 36, 27, 18, 7, 1021, 1018, 1015, 1012, 1009, 1006, 1003, 1000, 997, 994, 991, 988, 985, 982, 979, 976, 973, 970, 967, 964, 961, 958, 955, 952, 949, 946, 943, 940, 937, 934, 931, 928, 925, 922, 919, 916, 913, 910, 907, 904, 901, 898, 895, 892, 889, 886, 883, 880, 877, 874, 871, 868, 865, 862, 859, 856, 853, 850, 847, 844, 841, 838, 835, 832, 829, 826, 823, 820, 817, 814, 811, 808, 805, 802, 799, 796, 793, 790, 787, 784, 781, 778, 775, 772, 769, 766, 763, 760, 757, 754, 751, 748, 745, 742, 739, 736, 733, 730, 727, 724, 721, 718, 715, 712, 709, 706, 703, 700, 697, 694, 691, 688, 685, 682, 679, 676, 673, 670, 667, 664, 661, 658, 655, 652, 649, 646, 643, 640, 636, 632, 628, 624, 620, 616, 612, 608, 604, 600, 596, 592, 588, 584, 580, 576, 572, 568, 564, 560, 556, 552, 548, 544, 540, 536, 532, 528, 524, 520, 516, 512, 508, 504, 500, 496, 492, 488, 484, 480, 476, 472, 468, 464, 460, 456, 452, 448, 444, 440, 436, 432, 428, 424, 420, 416, 412, 408, 404, 400, 396, 392, 388, 384, 379, 374, 369, 364, 359, 354, 349, 344, 339, 334, 329, 324, 319, 314, 309, 304, 299, 294, 289, 284, 279, 274, 269, 264, 259, 254, 249, 244, 239, 234, 229, 224, 218, 212, 206, 200, 194, 188, 182, 176, 170, 164, 158, 152, 146, 140, 134, 128, 121, 114, 107, 100, 93, 86, 79, 72, 64, 56, 48, 40, 31, 22, 12, 1, 639, 635, 631, 627, 623, 619, 615, 611, 607, 603, 599, 595, 591, 587, 583, 579, 575, 571, 567, 563, 559, 555, 551, 547, 543, 539, 535, 531, 527, 523, 519, 515, 511, 507, 503, 499, 495, 491, 487, 483, 479, 475, 471, 467, 463, 459, 455, 451, 447, 443, 439, 435, 431, 427, 423, 419, 415, 411, 407, 403, 399, 395, 391, 387, 383, 378, 373, 368, 363, 358, 353, 348, 343, 338, 333, 328, 323, 318, 313, 308, 303, 298, 293, 288, 283, 278, 273, 268, 263, 258, 253, 248, 243, 238, 233, 228, 223, 217, 211, 205, 199, 193, 187, 181, 175, 169, 163, 157, 151, 145, 139, 133, 127, 120, 113, 106, 99, 92, 85, 78, 71, 63, 55, 47, 39, 30, 21, 10, 381, 376, 371, 366, 361, 356, 351, 346, 341, 336, 331, 326, 321, 316, 311, 306, 301, 296, 291, 286, 281, 276, 271, 266, 261, 256, 251, 246, 241, 236, 231, 226, 220, 214, 208, 202, 196, 190, 184, 178, 172, 166, 160, 154, 148, 142, 136, 130, 124, 117, 110, 103, 96, 89, 82, 75, 67, 59, 51, 43, 35, 26, 16, 5, 222, 216, 210, 204, 198, 192, 186, 180, 174, 168, 162, 156, 150, 144, 138, 132,

126,119,112,105,98,91,84,77,69,61,53,45,37,28,19,8,122,115,108,101,94,87,80,73,65,57,49,41,32,23,13,2,70,62,54,46,38,29,20,9,33,24,14,3,17,6,11,0]

[0262] 类似地,小于1024的母码序列的交织序列都可以从序列#5中读出,这里不再赘述。

[0263] 在本申请实施例中,利用极化码的准周期特性,在行列交织的基础上设计的交织序列,其纠错性能接近甚至超过随机交织,可以在不增加交织复杂度的情况下,提高极化码用于信道编码时的纠错性能。

[0264] 以上结合图1至图18对本申请提供的交织方法作了详细说明。下面说明本申请实施提供的交织装置。

[0265] 图19为本申请实施例提供的交织装置500的示意图。如图19所示,装置500包括收发单元510和处理单元520。其中,

[0266] 收发单元510,用于获取第一比特序列,第一比特序列包括L个比特,L为正整数;

[0267] 处理单元520,用于将所述L个比特按照预设的写入规则写入交织矩阵,交织矩阵包括C行R列,C和R为正整数;

[0268] 处理单元520,还用于按照预设的读取规则从交织矩阵中读取所述L个比特,得到第二比特序列;

[0269] 收发单元510,还用于发送第二比特序列。

[0270] 本申请实施例的装置500中的各单元和上述其它操作或功能分别为了实现本申请各实施例中的交织方法。为了简洁,此处不再赘述。

[0271] 本申请实施例提供的交织装置,利用极化码的准周期特性,可以在不增加交织复杂度的情况下,提高利用极化码进行信道编码时的纠错性能。

[0272] 图20为本申请实施例提供的交织设备700的示意性结构图。如图20所示,设备700包括:一个或多个处理器701,一个或多个存储器702,一个或多个收发器703。处理器701用于控制收发器703收发信号,存储器702用于存储计算机程序,处理器701用于从存储器702中调用并运行该计算机程序,使得设备700执行交织方法各实施例的相应流程和/或操作。为了简洁,此处不再赘述。

[0273] 需要说明的是,图19中所示的装置500可以通过图20中所示的设备700实现。例如,接收单元510可以由图20中的收发器703实现。处理单元520可以由处理器701实现等。

[0274] 这里的交织设备可以为图1中所示的网络设备或终端设备(例如,终端设备#1或终端设备#2)。具体地,在上行传输时,交织设备具体为终端设备,终端设备具有实现上述各实施例中描述的交织方法的功能。这些功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的单元。在下行传输时,交织设备具体为网络设备(例如,基站),网络设备具有实现上述各实施例中描述的交织方法的功能。同样地,这些功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的单元。

[0275] 当交织设备700具体为终端设备时,终端设备的结构可以如图21所示。图21为本申请实施例提供的终端设备800的示意性结构图。

[0276] 如图21所示,终端设备800包括:收发器808和处理器804。终端设备800还可以包括存储器819,其存储计算机执行指令。

[0277] 收发器808,用于获取第一比特序列,第一比特序列包括L个比特,L为正整数;

[0278] 处理器804,用于将第一比特序列按照预设的写入规则写入交织矩阵,再按照预设的读取规则将该L个比特读出,得到第二比特序列,第二比特序列包括L个比特,其中,交织矩阵包括C行R列,C和R为正整数;

[0279] 收发器808,用于根据处理器804的指示,输出第二比特序列。

[0280] 进一步地,上述处理器804可以用于执行前面方法实施例中描述的由交织设备内部实现的动作,而收发器808可以用于执行前面方法实施例中描述的交织设备的接收或发送动作。具体请见前面方法实施例中的描述,此处不再赘述。

[0281] 上述处理器804和存储器819可以集成为一个处理装置,处理器804用于执行存储器819中存储的程序代码来实现上述功能。具体实现时,该存储器819也可以集成在处理器804中。

[0282] 上述终端设备800还可以包括电源812,用于给终端设备800中的各种器件或电路提供电源。上述终端设备800可以包括天线810,用于将收发器808输出的数据或信息通过无线信号发送出去。

[0283] 除此之外,为了使得终端设备800的功能更加完善,该终端设备800还可以包括输入单元814,显示单元816,音频电路818,摄像头820和传感器822等中的一个或多个,所述音频电路还可以包括扬声器8182,麦克风8184等。

[0284] 此外,本申请提供一种计算机可读存储介质,该计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述各实施例中的交织方法。

[0285] 本申请还提供一种计算机程序产品,该计算机程序产品包括:计算机程序代码,当该计算机程序代码在计算机上运行时,使得计算机执行上述任意一个实施例中描述的交织方法。

[0286] 本申请还提供一种芯片(或者说,芯片系统),包括存储器和处理器,存储器用于存储计算机程序,处理器用于从存储器中调用并运行该计算机程序,使得安装有该芯片的通信设备执行本申请各方法实施例中的交织方法。

[0287] 其中,这里所说的通信设备可以为网络设备或终端设备。

[0288] 本申请还提供一种编码装置,该编码装置具有实现上述任意一个方法实施例中的交织方法的功能。所述功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的模块。除此之外,编码装置还具有实现Polar编码的相关功能。编码装置对待编码序列进行极化编码后,采用本申请实施例提供的交织方法,对编码后的序列进行交织,再进行后续的调制、映射、发送等。

[0289] 在一个可能的设计中,当所述功能的部分或全部通过硬件实现时,编码装置包括:

[0290] 输入接口电路,用于获取第一比特序列;

[0291] 逻辑电路,用于执行上述实施例中的任意一种可能的设计中的交织方法,对第一比特序列进行交织,得到第二比特序列;

[0292] 输出接口电路,用于输出第二比特序列。

[0293] 可选的,编码装置可以是芯片或者集成电路。

[0294] 在一个可能的设计中,当所述功能的部分或全部通过软件实现时,编码装置包括:存储器,用于存储程序;处理器,用于执行所述存储器存储的所述程序,当所述程序被执行时,编码装置可以实现上述实施例中任意一种可能的设计中的交织方法。

[0295] 在一个可能的设计中,当所述功能的部分或全部通过软件实现时,编码装置包括处理器。用于存储程序的存储器位于所述编码装置之外,处理器通过电路/电线与存储器连接,用于读取并执行所述存储器中存储的程序。

[0296] 可选的,上述的存储器与存储器可以是物理上相互独立的单元,或者,存储器也可以和处理器集成在一起。

[0297] 此外,本申请还提供一种解交织的装置900。参见图22,图22为本申请实施例提供的解交织的装置900的示意图。如图22所示,装置900包括收发单元910和处理单元920。其中,

[0298] 收发单元910,用于获取待解交织的比特序列;

[0299] 处理单元920,用于按照预设的写入规则和读取规则,对待解交织的比特序列进行解交织,得到解交织后的比特序列。

[0300] 具体地,解交织的装置900的各单元分别为了实现解交织的方法中的相应功能,这些功能的实现可以通过硬件实现,也可以通过硬件执行相应的软件实现。

[0301] 当解交织的方法通过硬件实现时,解交织的装置900包括:

[0302] 输入接口电路,用于获取待解交织的比特序列;

[0303] 逻辑电路,用于根据本申请实施例中描述的预设的写入规则和读取规则,对该待解交织的比特序列进行解交织,得到解交织后比特序列;

[0304] 输出接口电路,用于输出解交织后的比特序列。

[0305] 在理想无噪声的情况下,解交织后的比特序列与上文描述的第一比特序列相同。

[0306] 图23为本申请实施例提供的解交织的设备1000的示意性结构图。如图23所示,设备1000包括:一个或多个处理器1001,一个或多个存储器1002,一个或多个收发器1003。处理器1001用于控制收发器1003收发信号,存储器1002用于存储计算机程序,处理器1001用于从存储器1002中调用并运行该计算机程序,使得设备1000执行解交织的方法实施例的相应流程和/或操作。为了简洁,此处不再赘述。

[0307] 需要说明的是,图22中所示的装置900可以通过图23中所示的设备1000实现。例如,收发单元910可以由图23中的收发器1003实现。处理单元920可以由处理器1001实现等。

[0308] 在上行传输时,解交织的装置900具体为图1中所示的网络设备。在下行传输时,解交织的装置具体为图1中所示的终端设备(例如,终端设备#1或终端设备#2)。

[0309] 此外,解交织的装置900还可以为芯片,使得安装有该芯片的通信设备可以执行本申请实施例中的解交织的方法。其中,该芯片可以安装在网络设备上,使得网络设备具有实现解交织的方法的功能。或者,该芯片也可以安装在终端设备,使得终端设备具有实现解交织的方法的功能。

[0310] 此外,本申请提供一种计算机可读存储介质,该计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述各实施例中的解交织的方法。

[0311] 本申请还提供一种计算机程序产品,该计算机程序产品包括:计算机程序代码,当该计算机程序代码在计算机上运行时,使得计算机执行上述任意一个实施例中描述的解交织的方法。

[0312] 本申请还提供一种译码装置,该译码装置具有实现上述本申请实施例中的解交织的方法的功能。所述功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。除此之

外,编码装置还具有实现Polar码译码的相关功能,例如,解速率匹配、译码等。

[0313] 以上实施例中,处理器可以为中央处理器(Central Processing Unit,CPU)、微处理器、特定应用集成电路(Application-Specific Integrated Circuit,ASIC),或一个或多个用于控制本申请方案程序执行的集成电路等。例如,处理器可以包括数字信号处理设备、微处理器设备、模数转换器、数模转换器等。处理器可以根据这些设备各自的功能而在这些设备之间分配移动设备的控制和信号处理的功能。此外,处理器可以包括操作一个或多个软件程序的功能,软件程序可以存储在存储器中。

[0314] 处理器的所述功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的模块。

[0315] 存储器可以是只读存储器(Read-Only Memory,ROM)或可存储静态信息和指令的其他类型的静态存储设备,随机存取存储器(Random Access Memory,RAM)或者可存储信息和指令的其他类型的动态存储设备。也可以是电可擦可编程只读存储器(Electrically Erasable Programmable Read-Only Memory,EEPROM)、只读光盘(Compact Disc Read-Only Memory,CD-ROM)或其他光盘存储、光碟存储(包括压缩光碟、激光碟、光碟、数字通用光碟、蓝光光碟等)、磁盘存储介质或者其他磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。

[0316] 结合前面的描述,本领域的技术人员可以意识到,本文实施例的方法,可以通过硬件(例如,逻辑电路),或者软件,或者硬件与软件的结合来实现。这些方法究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0317] 当上述功能通过软件的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。在这种情况下,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0318] 以上所述,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以所述权利要求的保护范围为准。

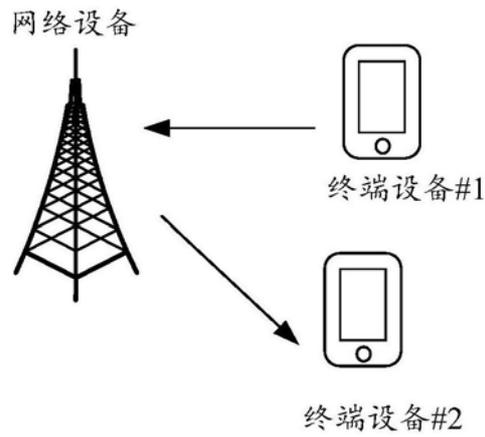


图1

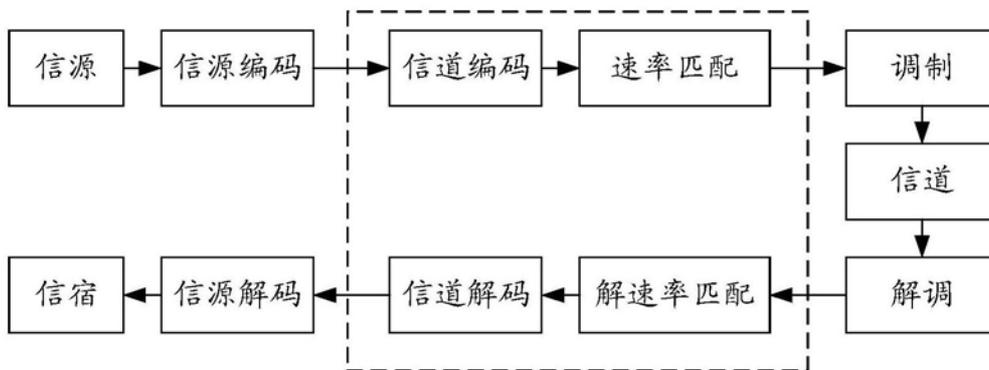


图2

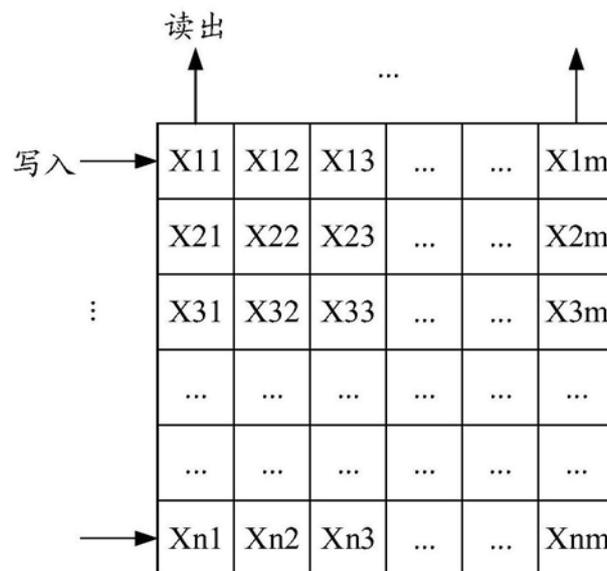


图3

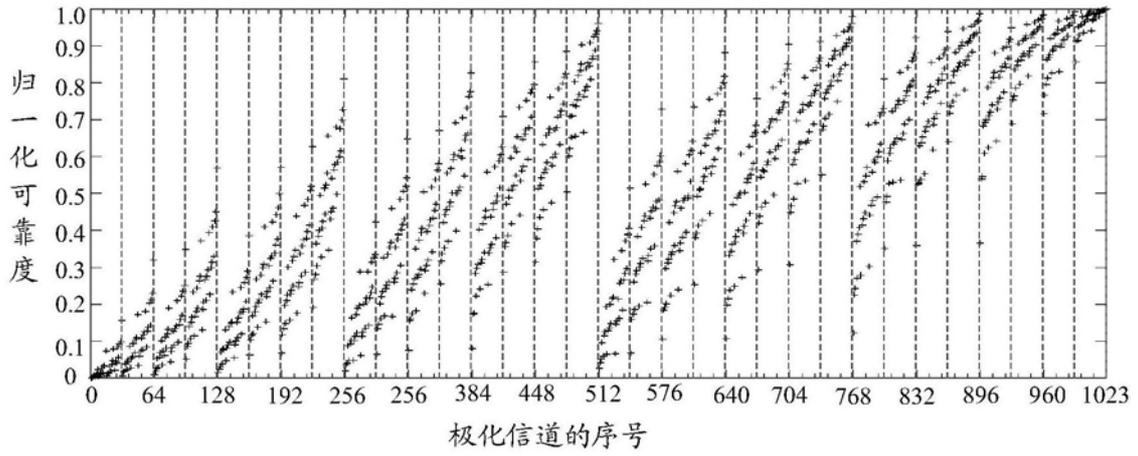


图4

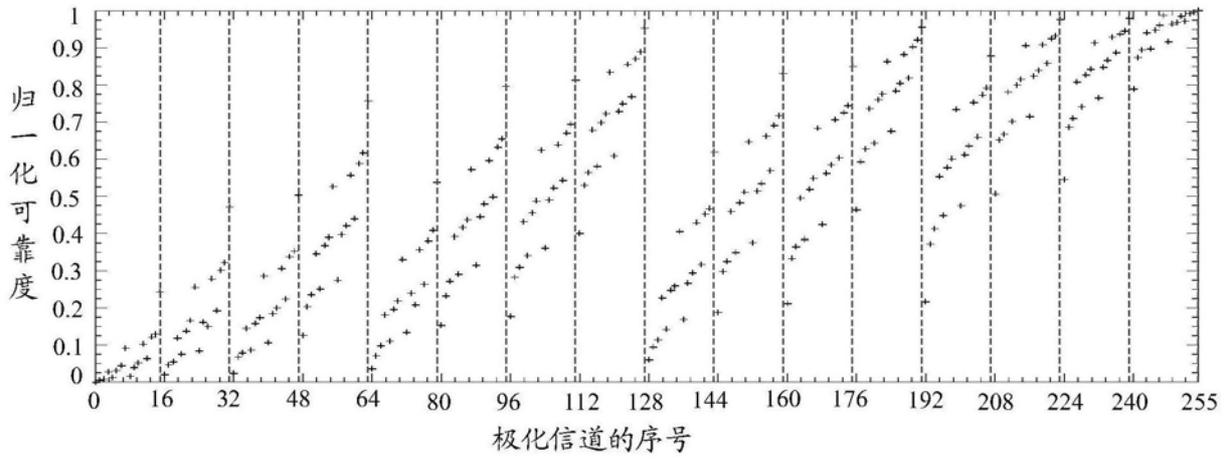


图5

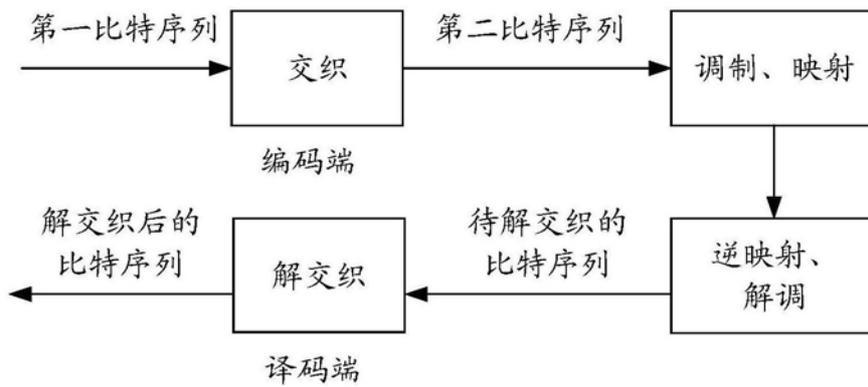


图6

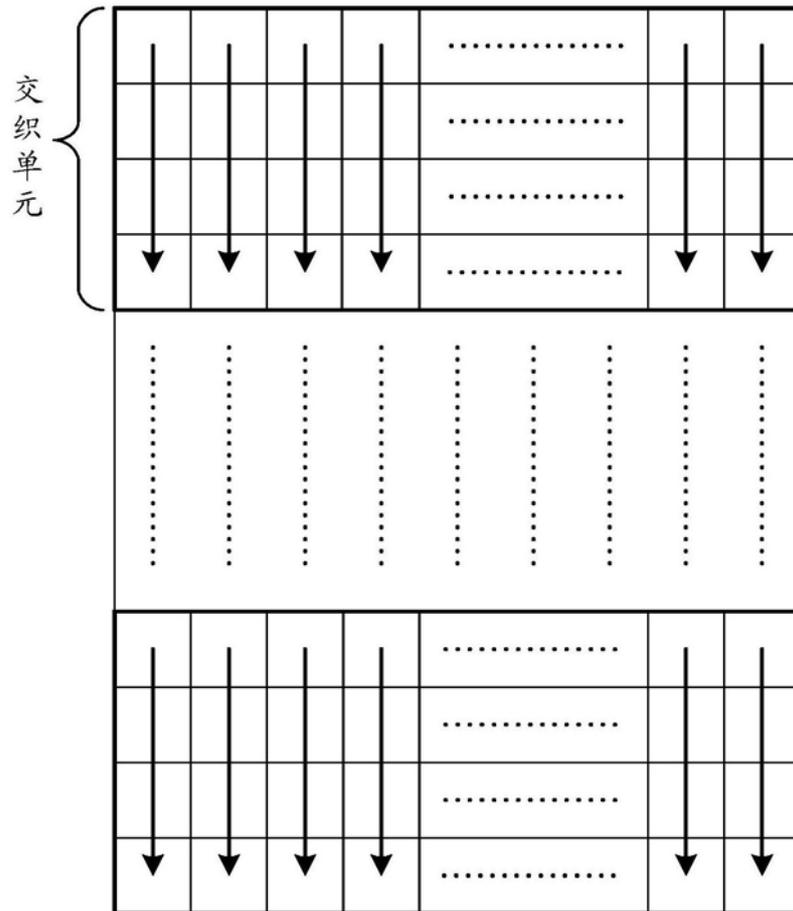


图7

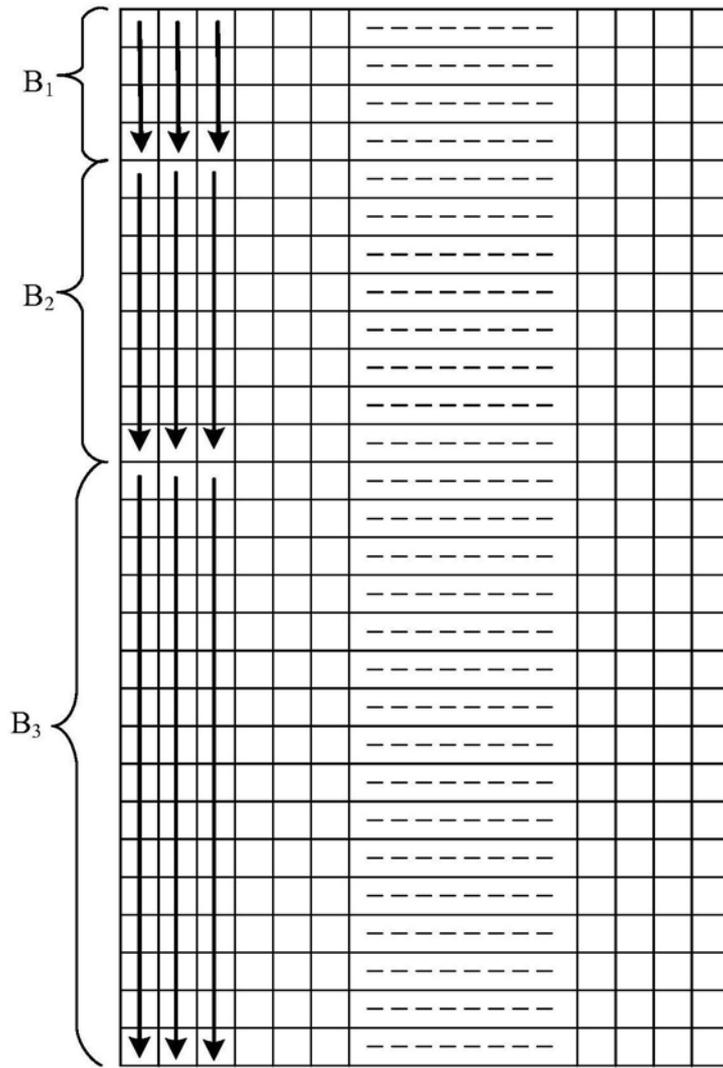


图8

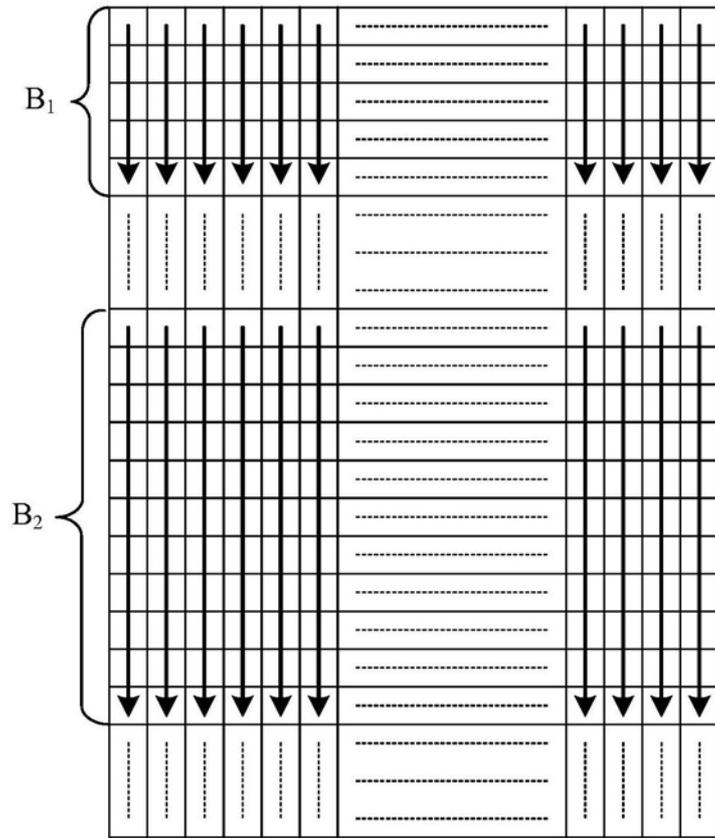


图9

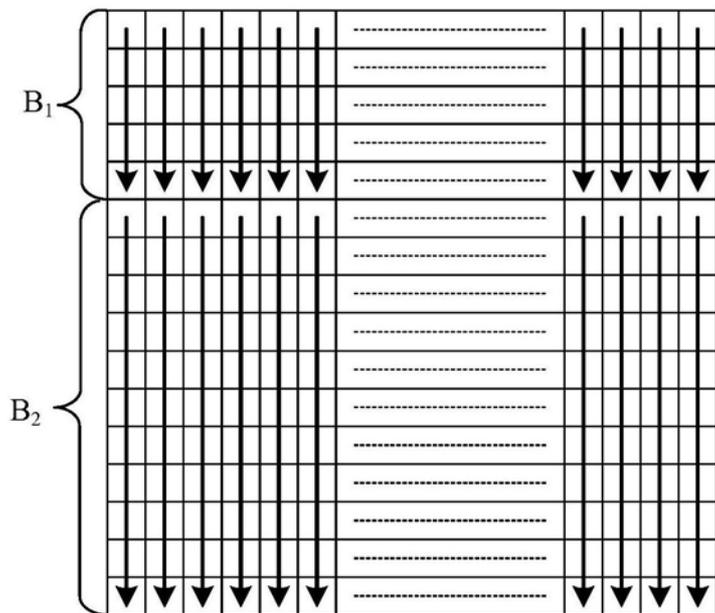


图10

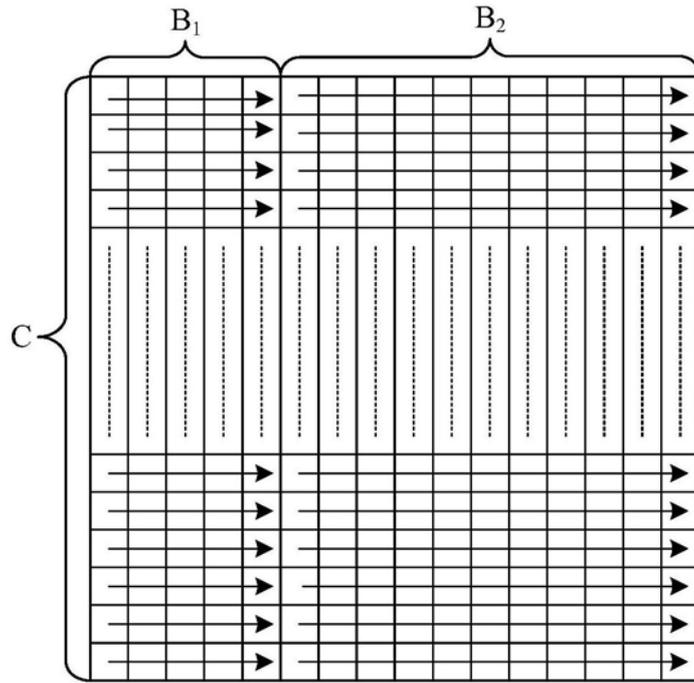


图11

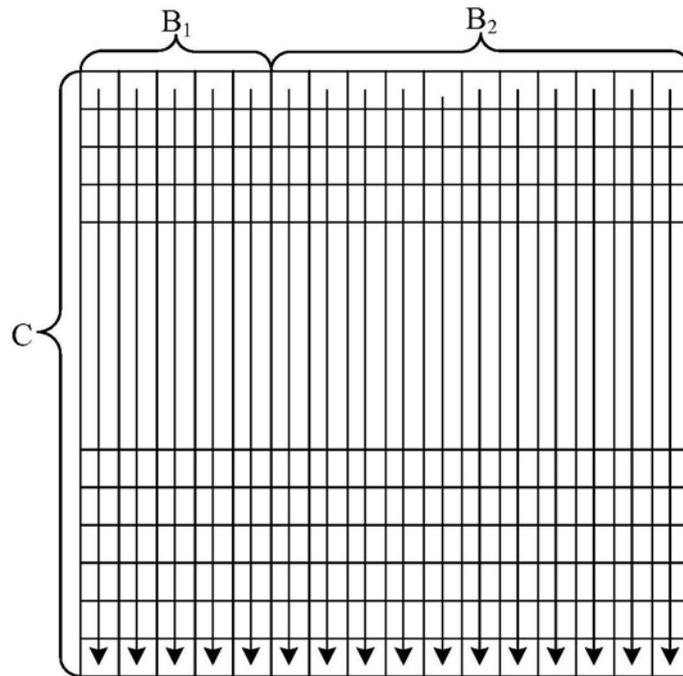


图12

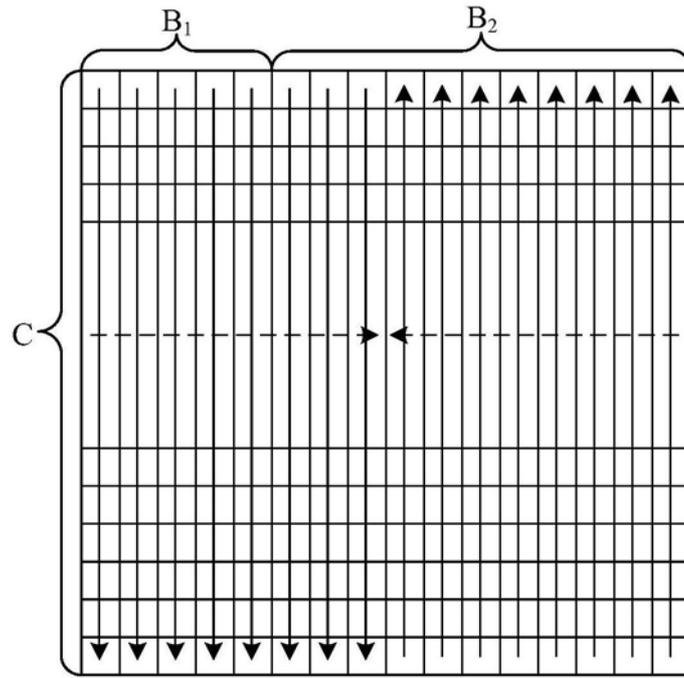


图13

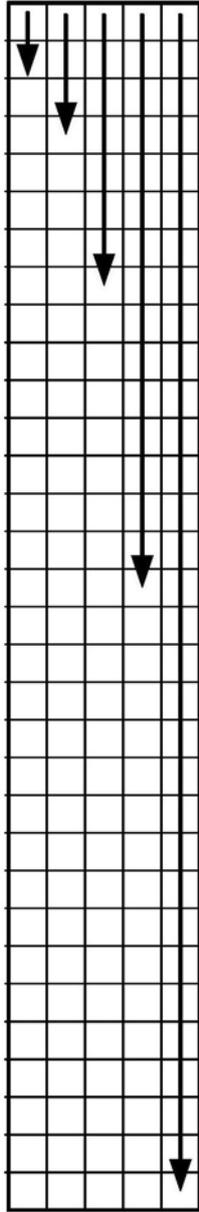


图14

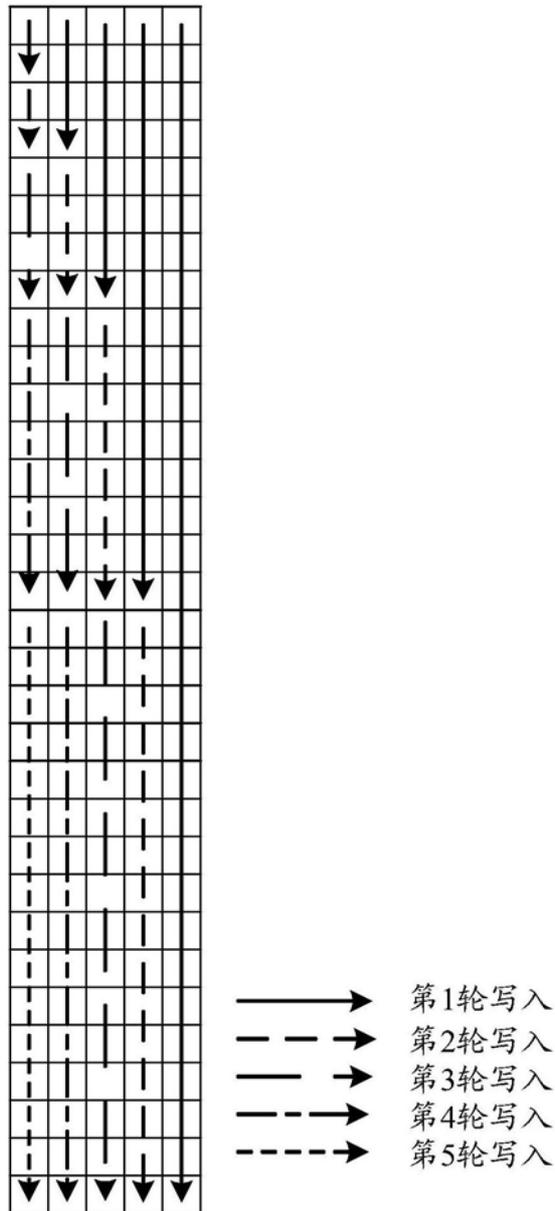


图15

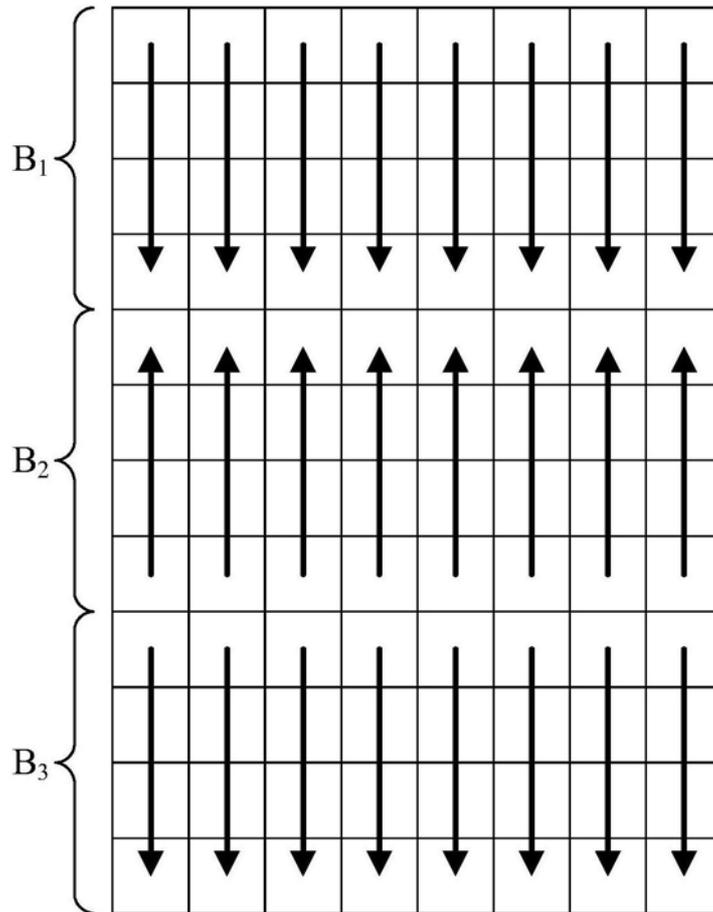


图17

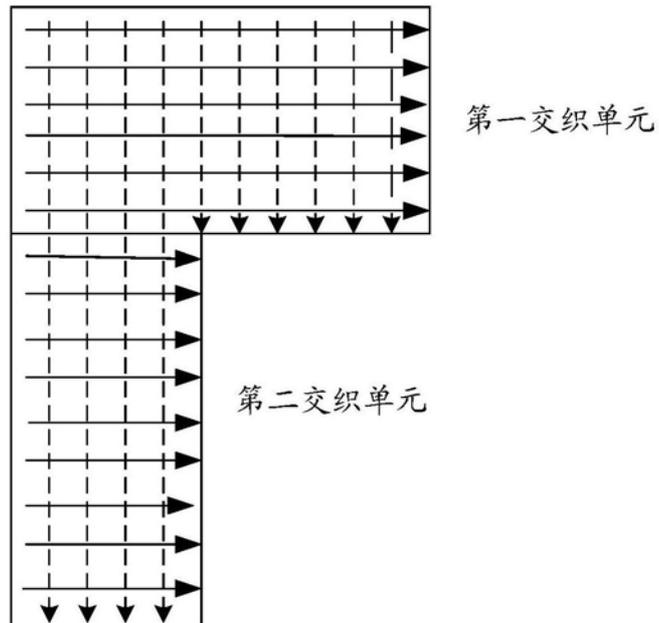


图18



图19

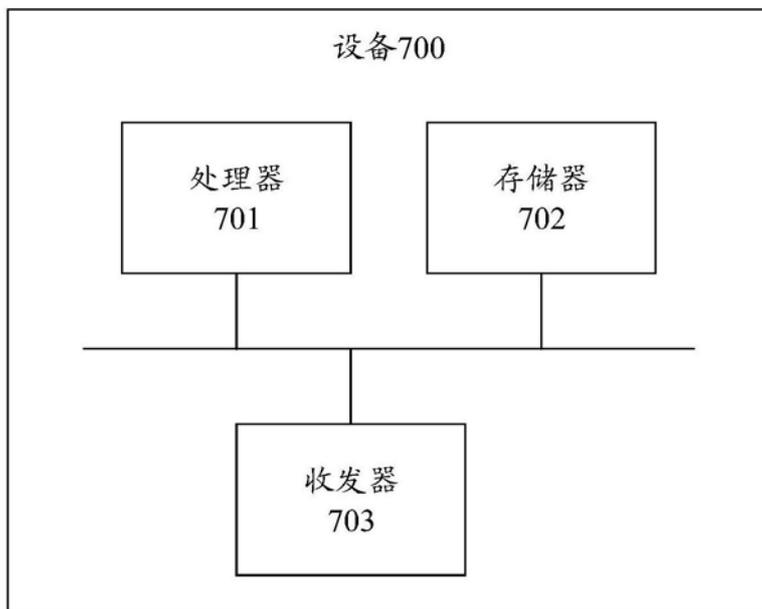


图20

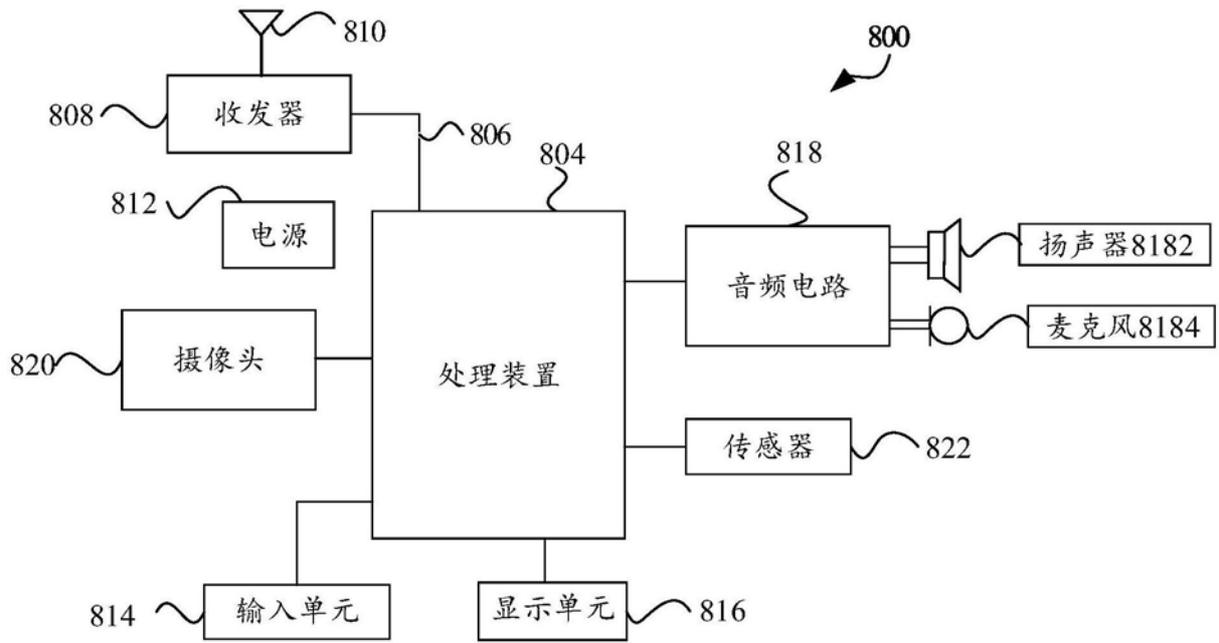


图21

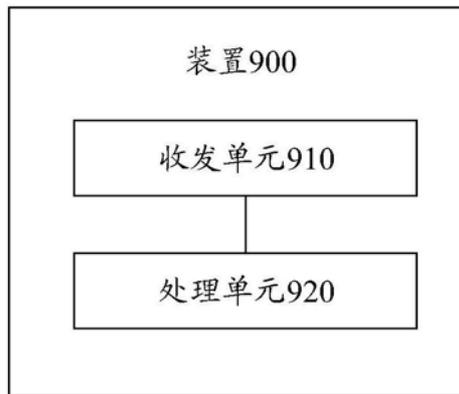


图22

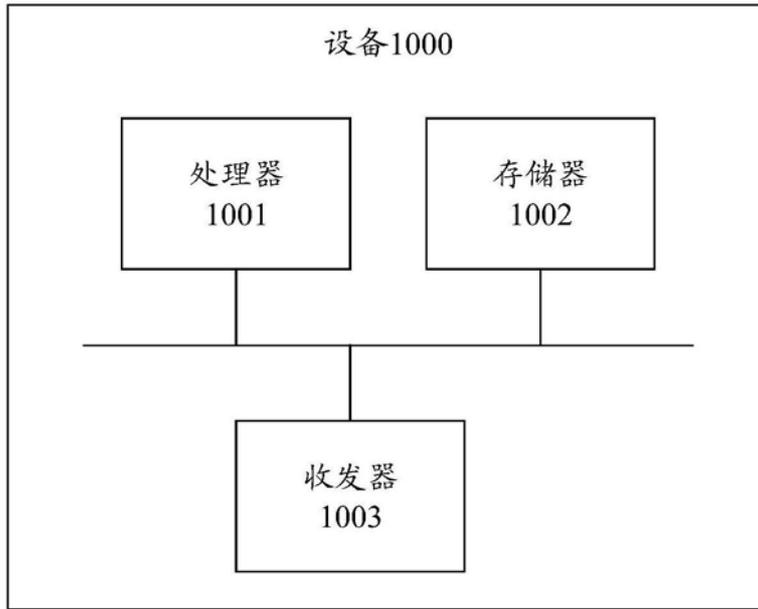


图23