



(12)发明专利

(10)授权公告号 CN 104917587 B

(45)授权公告日 2018.08.14

(21)申请号 201410093531.X

H03M 13/27(2006.01)

(22)申请日 2014.03.13

(56)对比文件

(65)同一申请的已公布的文献号
申请公布号 CN 104917587 A

EP 1463255 A1,2004.09.29,
WO 2010086874 A8,2012.04.19,
CN 101034951 A,2007.09.12,
CN 101662333 A,2010.03.03,
CN 101636914 A,2010.01.27,
CN 103560993 A,2014.02.05,

(43)申请公布日 2015.09.16

(73)专利权人 钜泉光电科技(上海)股份有限公司
地址 201203 上海市浦东新区张江高科技
园区张东路1388号16栋

审查员 吴龙

(72)发明人 张旭明

(74)专利代理机构 上海一平知识产权代理有限公司 31266
代理人 成春荣 竺云

(51)Int.Cl.

H04L 1/00(2006.01)

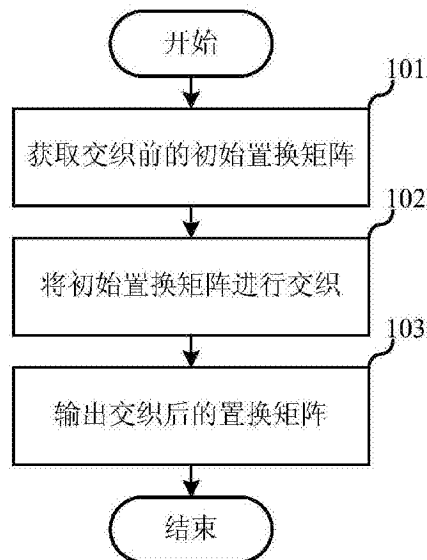
权利要求书5页 说明书17页 附图8页

(54)发明名称

通信设备中的数据块交织和解交织方法及其装置

(57)摘要

本发明涉及通信领域,公开了一种通信设备中的数据块交织和解交织方法及其装置。本发明中,提出了与现有通信标准的交织方式等价的新计算方法,新计算方式巧妙地把原有的取模运算以迭代的形式化解为简单的先加,然后比较,最后做最多两次减法的运算,从而以加法取代了现有算法中的乘法,以条件减法取代了取模运算,从而大大节省了硬件开销,加快了运算速度。



1. 一种通信设备中的数据块交织方法,其特征在于,包括以下步骤:

获取交织前的初始置换矩阵;

将初始置换矩阵进行交织,其中,初始置换矩阵中任意一个比特的位坐标 (i, j) 与该比特在交织后的置换矩阵中的位坐标 (I, J) 的关系为:

如果 $I=0$ 则

$$i(0, J) = (i(m-1, J-1) + m_j) \bmod m$$

$$j(0, J) = \begin{cases} (j(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(m-1, J-1) + m_j \geq m \\ (j(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i(m-1, J-1) + m_j < m \end{cases}$$

如果 $I \neq 0$ 则

$$i(I, J) = (i(I-1, J) + m_I) \bmod m,$$

$$j(I, J) = \begin{cases} (j(I-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(I-1, J) + m_I \geq m \\ (j(I-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i(I-1, J) + m_I < m \end{cases}$$

其中, $i(I, J)$ 为交织后的置换矩阵中位坐标为 (I, J) 的比特在初始置换矩阵中的行坐标, $j(I, J)$ 为交织后的置换矩阵中位坐标为 (I, J) 的比特在初始置换矩阵中的列坐标, $i=0, 1, \dots, m-1, j=0, 1, \dots, n-1, I=0, 1, \dots, m-1, J=0, 1, \dots, n-1, m$ 和 n 分别为初始置换矩阵的列数和行数, \bmod 表示取模运算,

且 m_I, m_J, n_I 和 n_J 由下列各式确定:

$$m_I = m_i^{-1} \bmod m$$

$$m_J = ((m-m_j) \times m_I) \bmod m$$

$$n_I = ((n-n_j) \times n_J) \bmod n$$

$$n_J = n_j^{-1} \bmod n$$

其中, m_i 和 m_j 为指定的与 m 互质的正整数, n_j 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定:

$$(m_i m_i^{-1}) \bmod m = 1$$

$$(n_j n_j^{-1}) \bmod n = 1,$$

而 \tilde{n}_I, \tilde{n}_J , 和 n_q 分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

$$\tilde{n}_J = (m_I n_I + n_J) \bmod n$$

$$n_q = (m \times (n-n_I)) \bmod n;$$

输出交织后的置换矩阵;

其中,所述取模运算通过以下方式实现:

将 \bmod 两边的数据进行比较,如果 \bmod 左边的数据大于或等于 \bmod 右边的数据,则用左边的数据减去右边的数据,将所得的差值作为取模运算的结果;

如果 \bmod 左边的数据小于 \bmod 右边的数据,则将左边的数据作为取模运算的结果。

2. 根据权利要求1所述的通信设备中的数据块交织方法,其特征在于,所述数据块包括n个正交频分复用符号,每个正交频复用符号包含m个子载波。

3. 根据权利要求1或2所述的通信设备中的数据块交织方法,其特征在于,该数据块交织方法用于ITU G.9902、ITU G.9903、IEEE P1901.2或G3-PLC前向纠错交织标准。

4. 一种通信设备中的数据块解交织方法,其特征在于,包括以下步骤:

获取已交织的置换矩阵;

将已交织的置换矩阵进行解交织,其中,已交织的置换矩阵中任意一个比特的位坐标(I,J)与该比特在解交织后的置换矩阵中的位坐标(i',j')的关系为:

如果I=0则

$$i'(0, J) = (i'(m-1, J-1) + m_j) \bmod m$$

$$j'(0, J) = \begin{cases} (j'(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(m-1, J-1) + m_j \geq m \\ (j'(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(m-1, J-1) + m_j < m \end{cases}$$

如果I≠0则

$$i'(I, J) = (i'(I-1, J) + m_I) \bmod m,$$

$$j'(I, J) = \begin{cases} (j'(I-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(I-1, J) + m_I \geq m \\ (j'(I-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(I-1, J) + m_I < m \end{cases}$$

其中, $i'(I, J)$ 为交织后的置换矩阵中位坐标为(I, J)的比特在解交织后的置换矩阵中的行坐标, $j'(I, J)$ 为交织后的置换矩阵中位坐标为(I, J)的比特在解交织后的置换矩阵中的列坐标, $i' = 0, 1, \dots, m-1$, $j' = 0, 1, \dots, n-1$, $I = 0, 1, \dots, m-1$, $J = 0, 1, \dots, n-1$, m和n分别为初始置换矩阵的列数和行数, mod表示取模运算,

且 m_I 、 m_J 、 n_I 和 n_J 由下列各式确定:

$$m_I = m_i^{-1} \bmod m$$

$$m_J = ((m - m_{j'}) \times m_I) \bmod m$$

$$n_I = ((n - n_{j'}) \times n_J) \bmod n$$

$$n_J = n_j^{-1} \bmod n$$

其中, m_i 和 $m_{j'}$ 为指定的与m互质的正整数, $n_{j'}$ 为指定的与n互质的正整数, m_i^{-1} 为 m_i 模m条件下的倒数, n_j^{-1} 为 n_j 模n条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定:

$$(m_i m_i^{-1}) \bmod m = 1$$

$$(n_j n_j^{-1}) \bmod n = 1,$$

而 \tilde{n}_I , \tilde{n}_J , 和 n_q 分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

$$\tilde{n}_J = (m_I n_I + n_J) \bmod n$$

$$n_q = (m \times (n - n_I)) \bmod n;$$

输出解交织后的置换矩阵;

其中,所述取模运算通过以下方式实现:

将mod两边的数据进行比较,如果mod左边的数据大于或等于mod右边的数据,则用左边的数据减去右边的数据,将所得的差值作为取模运算的结果;

如果mod左边的数据小于mod右边的数据,则将左边的数据作为取模运算的结果。

5.根据权利要求4中任一项所述的通信设备中的数据块解交织方法,其特征在于,所述数据块包括n个正交频分复用符号,每个正交频复用符号包含m个子载波。

6.根据权利要求4或5所述的通信设备中的数据块解交织方法,其特征在于,该数据块解交织方法用于ITU G.9902,ITU G.9903,IEEE P1901.2或G3-PLC前向纠错交织标准。

7.一种通信设备中的数据块交织装置,其特征在于,包括以下单元:

第一获取单元,用于获取交织前的初始置换矩阵;

交织单元,用于将初始置换矩阵进行交织,其中,初始置换矩阵中任意一个比特的的位置坐标(i,j)与该比特在交织后的置换矩阵中的位置坐标(I,J)的关系为:

如果I=0则

$$i(0,J) = (i(m-1,J-1) + m_j) \bmod m$$

$$j(0,J) = \begin{cases} (j(m-1,J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(m-1,J-1) + m_j \geq m \\ (j(m-1,J-1) + \tilde{n}_j) \bmod n & \text{若 } i(m-1,J-1) + m_j < m \end{cases}$$

如果I≠0则

$$i(I,J) = (i(I-1,J) + m_I) \bmod m,$$

$$j(I,J) = \begin{cases} (j(I-1,J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(I-1,J) + m_I \geq m \\ (j(I-1,J-1) + \tilde{n}_j) \bmod n & \text{若 } i(I-1,J) + m_I < m \end{cases}$$

其中,i(I,J)为交织后的置换矩阵中位置坐标为(I,J)的比特在初始置换矩阵中的行坐标,j(I,J)为交织后的置换矩阵中位置坐标为(I,J)的比特在初始置换矩阵中的列坐标,i=0,1,⋯,m-1,j=0,1,⋯,n-1,I=0,1,⋯,m-1,J=0,1,⋯,n-1,m和n分别为初始置换矩阵的列数和行数,mod表示取模运算,

且m_I、m_J、n_I和n_J由下列各式确定:

$$m_I = m_i^{-1} \bmod m$$

$$m_J = ((m-m_j) \times m_I) \bmod m$$

$$n_I = ((n-n_j) \times n_J) \bmod n$$

$$n_J = n_j^{-1} \bmod n$$

其中,m_i和m_j为指定的与m互质的正整数,n_j为指定的与n互质的正整数,m_i⁻¹为m_i模m条件下的倒数,n_j⁻¹为n_j模n条件下的倒数,且m_i⁻¹和n_j⁻¹分别由下式确定:

$$(m_i m_i^{-1}) \bmod m = 1$$

$$(n_j n_j^{-1}) \bmod n = 1,$$

而 \tilde{n}_I , \tilde{n}_J ,和n_q分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

$$\tilde{n}_j = (m_i n_i + n_j) \bmod n$$

$$n_q = (m \times (n - n_i)) \bmod n;$$

第一输出单元,用于输出交织后的置换矩阵;

其中,所述交织单元包括以下子单元:

第一取模子单元,用于将mod两边的数据进行比较,如果mod左边的数据大于或等于mod右边的数据,则用左边的数据减去右边的数据,将所得的差值作为取模运算的结果,如果mod左边的数据小于mod右边的数据,则将左边的数据作为取模运算的结果。

8. 一种通信设备中的数据块解交织装置,其特征在于,包括以下单元:

第二获取单元,用于获取已交织的置换矩阵;

解交织单元,用于将已交织的置换矩阵进行解交织,其中,已交织的置换矩阵中任意一个比特的位坐标(I, J)与该比特在解交织后的置换矩阵中的位坐标(i', j')的关系为:

如果I=0则

$$i'(0, J) = (i'(m-1, J-1) + m_j) \bmod m$$

$$j'(0, J) = \begin{cases} (j'(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(m-1, J-1) + m_j \geq m \\ (j'(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(m-1, J-1) + m_j < m \end{cases}$$

如果I≠0则

$$i'(I, J) = (i'(I-1, J) + m_i) \bmod m,$$

$$j'(I, J) = \begin{cases} (j'(I-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(I-1, J) + m_i \geq m \\ (j'(I-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(I-1, J) + m_i < m \end{cases}$$

其中, i'(I, J) 为交织后的置换矩阵中位坐标为(I, J)的比特在解交织后的置换矩阵中的行坐标, j'(I, J) 为交织后的置换矩阵中位坐标为(I, J)的比特在解交织后的置换矩阵中的列坐标, i'=0, 1, ..., m-1, j'=0, 1, ..., n-1, I=0, 1, ..., m-1, J=0, 1, ..., n-1, m和n分别为初始置换矩阵的列数和行数, mod表示取模运算,

且m_i、m_j、n_i和n_j由下列各式确定:

$$m_i = m_i^{-1} \bmod m$$

$$m_j = ((m - m_{j'}) \times m_i) \bmod m$$

$$n_i = ((n - n_{j'}) \times n_j) \bmod n$$

$$n_j = n_j^{-1} \bmod n$$

其中, m_i'和m_j'为指定的与m互质的正整数, n_j'为指定的与n互质的正整数, m_i⁻¹为m_i模m条件下的倒数, n_j⁻¹为n_j模n条件下的倒数, 且m_i⁻¹和n_j⁻¹分别由下式确定:

$$(m_i m_i^{-1}) \bmod m = 1$$

$$(n_j n_j^{-1}) \bmod n = 1,$$

而 \tilde{n}_i , \tilde{n}_j , 和n_q分别由下列各式确定:

$$\tilde{n}_i = (m_i n_i) \bmod n$$

$$\tilde{n}_j = (m_I n_I + n_j) \bmod n$$

$$n_q = (m \times (n - n_I)) \bmod n;$$

第二输出单元,用于输出解交织后的置换矩阵;

其中,所述解交织单元包括以下子单元:

第二取模子单元,用于将mod两边的数据进行比较,如果mod左边的数据大于或等于mod右边的数据,则用左边的数据减去右边的数据,将所得的差值作为取模运算的结果,如果mod左边的数据小于mod右边的数据,则将左边的数据作为取模运算的结果。

9. 一种正交频分复用通信设备中的数据发送方法,其特征在于,包括如权利要求1-3中任一项所述的数据块交织步骤。

10. 一种正交频分复用通信设备中的数据接收方法,其特征在于,包括如权利要求4-6中任一项所述的数据块解交织步骤。

通信设备中的数据块交织和解交织方法及其装置

技术领域

[0001] 本发明涉及通信领域,特别涉及一种通信设备中的数据块交织和解交织技术。

背景技术

[0002] 在陆地移动通信这种变参信道上,比特差错经常是成串发生的。这是由于持续较长的深衰落会影响到相继一串的比特。然而,一般的信道编码仅在检测和校正单个差错和不太长的差错串时才有效。为了解决这一问题,希望能找到把一条消息中的相继比特分散开的方法,即一条消息中的相继比特以非相继方式被发送。这样,在传输过程中即使发生了成串差错,恢复成一条相继比特串的消息时,差错也就变成单个(或长度很短),这时再用信道编码纠错功能纠正差错,恢复原消息。这种方法就是交织技术。

[0003] ITU G.9902,ITU G.9903,IEEE P1901.2,以及G3-PLC是基于正交频分复用(Orthogonal Frequency Division Multiplexing,简称“OFDM”)的电力载波通信(Power Line Communication,PLC)国际标准,其中采用的前向纠错(Forward Error Control,简称“FEC”)交织器的设计能够为对抗以下两种不同误码源提供保护:

[0004] 由强脉冲干扰引起的连续几个正交频分复用(OFDM)符号误码(时域);

[0005] 由强频率选择性衰落或窄带干扰引起的连续几个正交频分复用(OFDM)子载波误码(频域)。

[0006] 为了提高通信的鲁棒性,上述PLC标准还采用了重叠码(repetitioncode),交织是在重叠复制过程后完成的,可以同时对抗上述两个问题,并提供时域和频域分集增益。信道交织器根据有效子载波数(m)把需要交织的比特排成m列乘n行的矩阵,n是OFDM的符号数。交织分两步完成。在第一步中,交织矩阵中的每列被循环移位不同的次数。因而,错误的正交频分复用符号被分散在了不同的符号中。在第二步中,交织矩阵中的每一行被循环移位不同的次数以防止强频率选择性衰落或窄带干扰破坏整列数据。循环移位的总次数由参数 m_i 、 m_j 、 n_i 和 n_j 确定,这些参数基于每个OFDM符号的子载波数(m)和OFDM的符号数(n)来确定。

[0007] 图1示出了应用上述标准的交织器的缓冲器中初始置换矩阵的比特排列。其中,初始置换矩阵中的初始比特位置为(i,j),其中, $i=0,1,\dots,m-1$, $j=0,1,\dots,n-1$ 。交织后的置换矩阵中比特的位置为(I,J),两者关系如下式: $I=(i \cdot m_i + j \cdot m_j) \bmod m$ $J=(j \cdot n_j + i \cdot n_i) \bmod n$

[0008] 其中 (m_i, m_j) 和 (n_i, n_j) 由下式选定: $\text{GCD}(m_i, m) = \text{GCD}(m_j, m) = \text{GCD}(n_i, n) = \text{GCD}(n_j, n) = 1$

[0009] $\text{GCD}(a, b)$ 表示a和b两个正整数的最大公约数。执行简单的搜索便可基于以下两个参数m和n找出一组适合的上述参数,并且通信协议或标准会规定产生这些参数的规约,其中,m为每个正交复用(OFDM)符号包含的子载波数,n为交织的数据块包含的正交复用符号的数目。

[0010] ITU G.9902标准或G3-PLC标准都提出一种查表法来实现交织器的双循环置换,但是,对于G3-PLC交织器,允许的帧长($K=n \cdot m$)范围很大,加上此表依赖于n和m,所以交织表必须在FCH(帧控制头frame controlheader,简称“FCH”)收到后才能生成。因此,这种交织方

法使得调制解调器在交织前后的运算量较大,需要较大的存储空间,增加了硬件开销,降低了计算效率。

[0011] 关于本发明的研究背景,具体可参考如下资料:

[0012] 1.G3-PLC的物理层规范(G3-PLC Physical Layer Specification),ERDF (Electricite Reseau Distribution France,法国电网输送公司),2009年8月;

[0013] 2.低频(小于500kHz)窄带电力线通讯的智能电网应用(Low Frequency (less than 500kHz) Narrowband Power Line Communications for Smart Grid Applications), IEEE P1901.2标准,2013年8月;

[0014] 3.用于ITU-T G.hnem网络的窄带正交频分复用电力线通讯收发器(Narrowband orthogonal frequency division multiplexing power line communication transceivers for ITU-T G.hnem networks),ITU G.9902标准,2012年10月;

[0015] 4.用于G3-PLC的窄带正交频分复用电力线通讯接收器(Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks),ITU G.9903标准,2012年10月。

发明内容

[0016] 本发明的目的在于提供一种通信设备中的数据块交织和解交织方法及其装置,在计算交织或解交织前后数据的对应关系时,把原有的取模运算以迭代的方式化解为简单的先加,然后比较,最后做最多两次减法的运算,从而以加法取代了现有算法中的乘法,以条件减法取代了取模运算,大大节省了硬件开销,加快了运算速度。

[0017] 为解决上述技术问题,本发明的实施方式公开了一种通信设备中的数据块交织方法,包括以下步骤:

[0018] 获取交织前的初始置换矩阵;

[0019] 将初始置换矩阵进行交织,其中,初始置换矩阵中任意一个比特的位坐标(i, j)与该比特在交织后的置换矩阵中的位坐标(I, J)的关系为:

[0020] 如果I=0则 $i(0, J) = (i(m-1, J-1) + m_j) \bmod m$

$$[0021] \quad j(0, J) = \begin{cases} (j(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(m-1, J-1) + m_j \geq m \\ (j(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i(m-1, J-1) + m_j < m \end{cases}$$

$$i(I, J) = (i(I-1, J) + m_I) \bmod m,$$

$$[0022] \quad j(I, J) = \begin{cases} (j(I-1, J-1) + \tilde{n}_I + n_q) \bmod n & \text{若 } i(I-1, J) + m_I \geq m \\ (j(I-1, J-1) + \tilde{n}_I) \bmod n & \text{若 } i(I-1, J) + m_I < m \end{cases}$$

[0023] 其中, $i(I, J)$ 为交织后的置换矩阵中位坐标为(I, J)的比特在初始置换矩阵中的行坐标, $j(I, J)$ 为交织后的置换矩阵中位坐标为(I, J)的比特在初始置换矩阵中的列坐标, $i=0, 1, \dots, m-1, j=0, 1, \dots, n-1, I=0, 1, \dots, m-1, J=0, 1, \dots, n-1, m$ 和 n 分别为初始置换矩阵的列数和行数, mod表示取模运算,

[0024] 且 m_I, m_J, n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \bmod m, m_J = ((m - m_j) \times m_I) \bmod m, n_I = ((n - n_j) \times n_J) \bmod n, n_J = n_j^{-1} \bmod n$

[0025] 其中, m_i 和 m_j 为指定的与 m 互质的正整数, n_j 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定: $(m_i m_i^{-1}) \bmod m = 1$
 $(n_j n_j^{-1}) \bmod n = 1$,

[0026] 而 \tilde{n}_I , \tilde{n}_J , 和 n_q 分别由下列各式确定: $\tilde{n}_I = (m_I n_I) \bmod n$, $\tilde{n}_J = (m_J n_J + n_J) \bmod n$
 $n_q = (m \times (n - n_I)) \bmod n$;

[0027] 输出交织后的置换矩阵。

[0028] 本发明的实施方式还公开了一种通信设备中的数据块解交织方法, 包括以下步骤:

[0029] 获取已交织的置换矩阵;

[0030] 将已交织的置换矩阵进行解交织, 其中, 已交织的置换矩阵中任意一个比特的位
 置坐标 (I, J) 与该比特在解交织后的置换矩阵中的位置坐标 (i', j') 的关系为:

[0031] 如果 $I=0$ 则 $i'(0, J) = (i'(m-1, J-1) + m_J) \bmod m$

[0032]
$$j'(0, J) = \begin{cases} (j'(m-1, J-1) + \tilde{n}_J + n_q) \bmod n & \text{若 } i'(m-1, J-1) + m_J \geq m \\ (j'(m-1, J-1) + \tilde{n}_J) \bmod n & \text{若 } i'(m-1, J-1) + m_J < m \end{cases}$$

[0033] 如果 $I \neq 0$ 则 $i'(I, J) = (i'(I-1, J) + m_I) \bmod m$,

[0034]
$$j'(I, J) = \begin{cases} (j'(I-1, J-1) + \tilde{n}_I + n_q) \bmod n & \text{若 } i'(I-1, J) + m_I \geq m \\ (j'(I-1, J-1) + \tilde{n}_I) \bmod n & \text{若 } i'(I-1, J) + m_I < m \end{cases}$$

[0035] 其中, $i'(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在解交织后的置
 换矩阵中的行坐标, $j'(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在解交织后的置
 换矩阵中的列坐标, $i' = 0, 1, \dots, m-1$, $j' = 0, 1, \dots, n-1$, $I = 0, 1, \dots, m-1$, $J = 0,$
 $1, \dots, n-1$, m 和 n 分别为初始置换矩阵的列数和行数, \bmod 表示取模运算,

[0036] 且 m_I , m_J , n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \bmod m$, $m_J = ((m - m_j') \times m_I) \bmod m$, $n_I = ((n -$
 $n_{j'}) \times n_j) \bmod n$, $n_J = n_j^{-1} \bmod n$

[0037] 其中, m_i' 和 m_j' 为指定的与 m 互质的正整数, n_j' 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i
 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定: $(m_i m_i^{-1}) \bmod m = 1$
 $(n_j n_j^{-1}) \bmod n = 1$,

[0038] 而 \tilde{n}_I , \tilde{n}_J , 和 n_q 分别由下列各式确定: $\tilde{n}_I = (m_I n_I) \bmod n$, $\tilde{n}_J = (m_J n_J + n_J) \bmod n$
 $n_q = (m \times (n - n_I)) \bmod n$;

[0039] 输出解交织后的置换矩阵。

[0040] 本发明的实施方式还公开了一种通信设备中的数据块交织装置, 包括以下单元:

[0041] 第一获取单元, 用于获取交织前的初始置换矩阵;

[0042] 交织单元, 用于将初始置换矩阵进行交织, 其中, 初始置换矩阵中任意一个比特的
 位置坐标 (i, j) 与该比特在交织后的置换矩阵中的位置坐标 (I, J) 的关系为:

[0043] 如果 $I=0$ 则

[0044] $i(0, J) = (i(m-1, J-1) + m_J) \bmod m$

$$[0045] \quad j(0, J) = \begin{cases} (j(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(m-1, J-1) + m_j \geq m \\ (j(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i(m-1, J-1) + m_j < m \end{cases}$$

[0046] 如果 $I \neq 0$ 则

$$[0047] \quad i(I, J) = (i(I-1, J) + m_I) \bmod m,$$

$$[0048] \quad j(I, J) = \begin{cases} (j(I-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(I-1, J) + m_I \geq m \\ (j(I-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i(I-1, J) + m_I < m \end{cases}$$

[0049] 其中, $i(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在初始置换矩阵中的行坐标, $j(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在初始置换矩阵中的列坐标, $i=0, 1, \dots, m-1, j=0, 1, \dots, n-1, I=0, 1, \dots, m-1, J=0, 1, \dots, n-1, m$ 和 n 分别为初始置换矩阵的列数和行数, \bmod 表示取模运算,

[0050] 且 m_I, m_J, n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \bmod m, m_J = ((m - m_j) \times m_I) \bmod m, n_I = ((n - n_j) \times n_J) \bmod n, n_J = n_j^{-1} \bmod n$

[0051] 其中, m_i 和 m_j 为指定的与 m 互质的正整数, n_j 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定: $(m_i m_i^{-1}) \bmod m = 1$ $(n_j n_j^{-1}) \bmod n = 1$

[0052] 而 \tilde{n}_I, \tilde{n}_J , 和 n_q 分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

$$[0053] \quad \tilde{n}_J = (m_J n_J + n_q) \bmod n$$

$$n_q = (m \times (n - n_I)) \bmod n;$$

[0054] 第一输出单元, 用于输出交织后的置换矩阵。

[0055] 本发明的实施方式还公开了一种通信设备中的数据块解交织装置, 包括以下单元:

[0056] 第二获取单元, 用于获取已交织的置换矩阵;

[0057] 解交织单元, 用于将已交织的置换矩阵进行解交织, 其中, 已交织的置换矩阵中任意一个比特的位坐标 (I, J) 与该比特在解交织后的置换矩阵中的位坐标 (i', j') 的关系为:

[0058] 如果 $I=0$ 则

$$[0059] \quad i'(0, J) = (i'(m-1, J-1) + m_J) \bmod m$$

$$[0060] \quad j'(0, J) = \begin{cases} (j'(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(m-1, J-1) + m_J \geq m \\ (j'(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(m-1, J-1) + m_J < m \end{cases}$$

[0061] 如果 $I \neq 0$ 则

$$[0062] \quad i'(I, J) = (i'(I-1, J) + m_I) \bmod m,$$

$$[0063] \quad j'(I, J) = \begin{cases} (j'(I-1, J-1) + \tilde{n}_I + n_q) \bmod n & \text{若 } i'(I-1, J) + m_I \geq m \\ (j'(I-1, J-1) + \tilde{n}_I) \bmod n & \text{若 } i'(I-1, J) + m_I < m \end{cases}$$

[0064] 其中, $i'(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在解交织后的置换矩阵中的行坐标, $j'(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在解交织后的置换矩阵中的列坐标, $i'=0, 1, \dots, m-1, j'=0, 1, \dots, n-1, I=0, 1, \dots, m-1, J=0, 1, \dots, n-1, m$ 和 n 分别为初始置换矩阵的列数和行数, \bmod 表示取模运算,

[0065] 且 m_I, m_J, n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \bmod m, m_J = ((m - m_{j'}) \times m_I) \bmod m, n_I = ((n - n_{j'}) \times n_J) \bmod n, n_J = n_j^{-1} \bmod n$

[0066] 其中, m_i 和 $m_{j'}$ 为指定的与 m 互质的正整数, n_j 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定:

$$[0067] \quad (m_i m_i^{-1}) \bmod m = 1$$

$$[0068] \quad (n_j n_j^{-1}) \bmod n = 1$$

[0069] 而 \tilde{n}_I, \tilde{n}_J , 和 n_q 分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

$$[0070] \quad \tilde{n}_J = (m_J n_I + n_J) \bmod n$$

$$n_q = (m \times (n - n_I)) \bmod n;$$

[0071] 第二输出单元, 用于输出解交织后的置换矩阵。

[0072] 本发明的实施方式还公开了一种正交频分复用通信设备中的数据发送方法, 包括上述一种通信设备中的数据块交织方法中的数据块交织步骤。

[0073] 本发明的实施方式还公开了一种正交频分复用通信设备中的数据接收方法, 包括上述一种通信设备中的数据块解交织方法中的数据块解交织步骤。

[0074] 本发明实施方式与现有技术相比, 主要区别及其效果在于:

[0075] 该交织(或解交织)方法在计算交织(或解交织)前后数据的对应关系时, 利用一种迭代算法来取代原有的取模运算, 以加法取代了原先算法中的乘法, 因为加法运算所需要的硬件开销远小于乘法运算, 加法运算的速度也远快于乘法运算, 所以大大节省了硬件开销, 加快了运算速度。并通过使用简单的硬件在传输过程中分别计算数据地址, 消除了与可编程处理器和预先计算的程序地址存储器相关的开销(如查表法), 极大的节省了硬件实现在交织过程中的计算量, 提高了计算效率。

[0076] 进一步地, 本申请的迭代计算方法保证了取模(mod)运算的左侧的变量小于右侧变量的两倍, 所以可以用简单的比较和减法运算取代复杂的标准mod运算(通常需要乘法或除法), 从而大大减少了硬件的开销, 加快了运算速度。

附图说明

[0077] 图1是现有技术中ITU G.9902, ITU G.9902, IEEE P1901.2或G3-PLC标准交织器的缓冲器中初始置换矩阵的比特排列;

[0078] 图2是本发明第一实施方式中一种通信设备中的数据块交织方法的流程示意图;

- [0079] 图3是本发明第二实施方式中一种通信设备中的数据块解交织方法的流程示意图；
- [0080] 图4示出了本发明第一实施方式中G.9902交织器生成的已编码的有效荷载块；
- [0081] 图5示出了本发明第一实施方式中当k=4时将比特插入置换矩阵的过程；
- [0082] 图6示出了本发明第一实施方式中比特被输入G3-PLC或P1901.2,G.9903交织器的缓冲器的顺序；
- [0083] 图7示出了本发明第一实施方式中G3-PLC RC4模式下置换矩阵的写入和读取顺序；
- [0084] 图8是本发明第一实施方式中交织器的输入和输出向量指数生成的流程图；
- [0085] 图9(a)是本发明第一实施方式中执行函数 $(i, j) = \text{IVL_IDX_INC}(i, j, m, n, \text{step_mod_i}, \text{step_j})$ 的流程图；
- [0086] 图9(b)是本发明第一实施方式中执行函数 $\text{ACS}(j, \text{step_i}, j)$ 的流程图；
- [0087] 图10是本发明第三实施方式中一种通信设备中的数据块交织装置的结构示意图；
- [0088] 图11是本发明第三实施方式中的一实际应用的交织器的工作流程示意图；
- [0089] 图12是本发明第四实施方式中一种通信设备中的数据块解交织装置的结构示意图；
- [0090] 图13是本发明第四实施方式中的一个实际应用的解交织器的工作流程示意图。

具体实施方式

[0091] 在以下的叙述中,为了使读者更好地理解本申请而提出了许多技术细节。但是,本领域的普通技术人员可以理解,即使没有这些技术细节和基于以下各实施方式的种种变化和修改,也可以实现本申请各权利要求所要求保护的技术方案。在本发明的各实施方式中,相同的符号代表同一部件或具有同样的意义。

[0092] 为使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明的实施方式作进一步地详细描述。

[0093] 本发明第一实施方式涉及一种通信设备中的数据块交织方法。图2是该通信设备中的数据块交织方法的流程示意图。

[0094] 具体地说,如图2所示,该通信设备中的数据块交织方法包括以下步骤:

[0095] 在步骤101中,获取交织前的初始置换矩阵。

[0096] 此后进入步骤102,将初始置换矩阵进行交织,其中,初始置换矩阵中任意一个比特的位坐标 (i, j) 与该比特在交织后的置换矩阵中的位坐标 (I, J) 的关系为:

[0097] 如果 $I=0$ 则

$$[0098] \quad i(0, J) = (i(m-1, J-1) + m_j) \bmod m$$

$$[0099] \quad j(0, J) = \begin{cases} (j(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(m-1, J-1) + m_j \geq m \\ (j(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i(m-1, J-1) + m_j < m \end{cases}$$

[0100] 如果 $I \neq 0$ 则

$$[0101] \quad i(I, J) = (i(I-1, J) + m_I) \bmod m,$$

$$[0102] \quad j(I, J) = \begin{cases} (j(I-1, J-1) + \tilde{n}_I + n_q) \bmod n & \text{若 } i(I-1, J) + m_I \geq m \\ (j(I-1, J-1) + \tilde{n}_I) \bmod n & \text{若 } i(I-1, J) + m_I < m \end{cases}$$

[0103] 其中, $i(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在初始置换矩阵中的行坐标, $j(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在初始置换矩阵中的列坐标, $i(0, J)$ 表示交织后的置换矩阵中位置坐标为 $(0, J)$ 的比特在初始置换矩阵中的行坐标, $j(0, J)$ 表示交织后的置换矩阵中位置坐标为 $(0, J)$ 的比特在初始置换矩阵中的列坐标, $i=0, 1, \dots, m-1, j=0, 1, \dots, n-1, I=0, 1, \dots, m-1, J=0, 1, \dots, n-1, i(m-1, J-1)$ 和 $j(m-1, J-1)$ 分别表示在交织后的置换矩阵中位置坐标为 $(m-1, J-1)$ 的比特在初始置换矩阵中的行坐标和列坐标, $i(I-1, J)$ 表示在交织后的置换矩阵中位置坐标为 $(I-1, J)$ 的比特在初始置换矩阵中的行坐标, $j(I-1, J-1)$ 表示在交织后的置换矩阵中位置坐标为 $(I-1, J-1)$ 的比特在初始置换矩阵中的列坐标, 相对于交织后在置换矩阵中位置为 (I, J) 的比特, 在对其进行交织时, 这四个坐标已知; m 和 n 分别为初始置换矩阵的列数和行数, \bmod 表示取模运算, m_I, m_J, n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \bmod m, m_J = ((m - m_j) \times m_I) \bmod m, n_I = ((n - n_j) \times n_J) \bmod n, n_J = n_j^{-1} \bmod n$

[0104] 其中, m_i 和 m_j 为指定的与 m 互质的正整数, n_j 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数 (为正整数), n_j^{-1} 为 n_j 模 n 条件下的倒数 (为正整数), 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定: $(m_i m_i^{-1}) \bmod m = 1, (n_j n_j^{-1}) \bmod n = 1,$

[0105] $\tilde{n}_I, \tilde{n}_J,$ 和 n_q 分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

$$[0106] \quad n_q = (m \times (n - n_I)) \bmod n。$$

$$\tilde{n}_J = (m_I n_I + n_J) \bmod n$$

[0107] 在本实施方式中, 取模运算通过以下方式实现:

[0108] 将 \bmod 两边的数据进行比较, 如果 \bmod 左边的数据大于或等于 \bmod 右边的数据, 则用左边的数据减去右边的数据, 将所得的差值作为取模运算的结果; 如果 \bmod 左边的数据小于 \bmod 右边的数据, 则将左边的数据作为取模运算的结果。

[0109] 此外, 在本实施方式中, 上述数据块包括 n 个正交频分复用符号, 每个正交频复用符号包含 m 个子载波。

[0110] 在本发明中, 上述计算方法保证了取模运算 (\bmod) 的左侧变量小于右侧变量的两倍, 所以可以用简单的比较和减法运算取代复杂的标准 \bmod 运算, 从而大大减少了硬件的开销, 大大加快了运算速度。

[0111] 此后进入步骤 103, 输出交织后的置换矩阵。

[0112] 此后, 结束本流程。

[0113] 在本实施方式中, 上述数据块交织方法用于 ITU G.9902、ITU G.9903、IEEE P1901.2 或 G3-PLC 前向纠错交织标准。

[0114] 下面, 以 ITU G.9902 交织标准和 ITU G.9903 (或 G3-PLC) 交织标准为例说明本交织方法一实际的实现过程。应用两种交织标准的交织器的基本操作如下:

[0115] (1) ITU G.9902 交织器

[0116] ITU G.9902交织器的被设计为能够同时抵抗频域和时域误码,包括从1/2交流电(Alternating Current,简称“AC”)周期(50Hz或60Hz)持续到1/4AC周期期间的周期性误码,同时能增强接收器灵敏度以对抗加性高斯白噪声(Additive White Gaussian Noise,简称“AWGN”)。对于有效荷载,交织器首先将其分为片段(fragment)。然后将每个片段复制2,4,6或12次以增强其鲁棒性,并用片段交织(IoF)模式和AC循环交织(IoAC)模式中的一种将其交织。在片段交织(IoF)模式中,每个片段被正常交织。AC循环交织(IoAC)模式是专为具有严重周期性干扰的信道设计的,比如电力载波信道。在这一模式中,交织前,每个片段被额外的片段拷贝填充以接近1/4AC周期的倍数。在两种模式中,如果使用了片段拷贝,则来自每个片段拷贝的比特除了第一位,其余的相对于前一个片段拷贝被循环移位以分散频域误码。下文将为该片段拷贝的数据头确定更加鲁棒的交织器。

[0117] 图4示出了本实例中G.9902交织器生成的已编码有效荷载块。如图4所示,定义m为每个OFDM符号的有效数据荷载数,n为每帧所用的OFDM符号的数目, B_0 为片段的长度。片段重复编码器(FRE)以复制率R提供片段的拷贝(如图4中所示的Rep1、Rep2、...、Rep R等)。每个片段会被复制R次并且所有的片段拷贝会被串连接到片段缓冲器(FB)中,从而使每个片段拷贝的第一个比特连接前一个片段拷贝的最后一个比特,参见图4。片段缓冲器FB总的大小为 $B_0 \times R$ 比特(如图4中的缓冲器FB1)。片段重复编码器应当支持的R值为R=1、2、4、6、12(R=1对应于操作的正常模式)。如果R=1,则一个FB相应地应包含一个 B_0 比特的单独片段。 B_0 的输出比特被写成n行m列的置换矩阵。利用下列方程将比特插入到矩阵中: $q = \text{floor}(p / (k \times m))$
 $r = \text{mod}(p, k \times m)$
 $i = \text{mod}(r, m)$
 $j = k \times q + \text{floor}(r / k)$

[0118] 其中,p是输入数列(输入向量)中的比特序号,变化范围为0到 B_1-1 ;k取决于所用的调制,(k=1为BPSK调制,k=2是QPSK的调制,k=4为16QAM的调制,等等);i是置换矩阵的列指数,变化范围为0~m-1,j是行指数,变化范围为0~n-1(m列n行); $\text{floor}(x)$ 表示对x的取整运算。

[0119] 图5示出了当k=4(16QAM)时利用上述方程将比特插入置换矩阵的过程。如图5所示,图中每个方格表示一个比特。每个方格中的数字分别表示输入的比特在比特输入数列(输入向量)和比特输出数列(输出向量)中的位置。整个过程是将输入比特按输入向量序列排列,并按图示组成交织输入矩阵进行交织,然后将交织输出矩阵中的元素按输出向量顺序输出。

[0120] 在本实例中,无需进行交织器的插入和提取操作,而是基于交织器的输出(已交织的)矩阵指数(I,J)(即比特在交织后的置换矩阵中的坐标)计算交织器的输入(交织中的)矩阵指数(i,j)(即比特在初始置换矩阵中的坐标)。用(i,j)可以根据下式计算输入向量序列号p:

$$[0121] \quad p = M2V(i, j, m, k) = \text{floor}(j/k) \times m \times k + \text{mod}(j, k) + i \times m$$

[0122] 由于在G.9902交织器中k=1、2或4, $\text{floor}(j/k)$ 可以通过分别向右移位0、1或2位比特来实现,而且 $\text{mod}(j, k)$ 也可以被简化为 $\text{mod}(j, k) = j \text{ mod } k = j \& (k-1)$,其中“&”是“逐比特求与”的操作符号。类似的,交织器输出向量序列号P可以通过 $P = M2V(I, J, m, k)$ 来计算。

[0123] 在片段的重复编码中,交织前,从第二个片段拷贝开始每个片段拷贝的比特(如图4中的“Rep2”)将相对于前一个片段拷贝在LSB(最低有效位)至MSB(最高有效位)的方向上被循环移位 $M = \text{ceiling}(B_0/R_r)$ 位,即拷贝“Rep(d+1)”相对于拷贝“Rep1”将被移位 $d \times M$ 位,从

而使拷贝“Rep1”的LSB在拷贝“Rep(d+1)”中具有位数 $d \times M$ 。值 $R_T \geq R$ 是拷贝的总次数,包括填充拷贝,其依赖于交织模式。上面 M 表示大于或等于 B_0/R_T 的最小整数。

[0124] (2) ITU G.9903(或G3-PLC)交织器

[0125] 在本实例中,对于ITU G.9903(或G3-PLC)交织器,其鲁棒(ROBO)模式(因重复4次,故简称“RC4”)和超级鲁棒(Super ROBO)模式(因重复6次,故简称“RC6”)使用重叠码(Repetition code),并且基本调制模式为DBPSK(Differential BPSK,差分二进制相移键控)。在鲁棒模式,卷积编码器输出的每个比特被复制四次后作为交织器的输入。该重叠码编码器(RC4)仅在鲁棒模式下是被触发的。

[0126] 定义 m 为每个OFDM符号中的有效数据荷载数, n 为每帧中的OFDM符号数(即交织的数据块中的OFDM符号数),而 $Total_num_of_bits$ 为包括填补比特在内的已编码的比特总数。

$$[0127] \quad n = \text{ceil} \left(\frac{\text{Total_num_of_bits}}{4 \cdot m \cdot k} \right) \times 4$$

[0128] 其中 $k=1, 2, 3, 4$ 为调制指数,即每个星座符号的比特数,函数 $\text{ceil}(x)$ 表示取大于或等于 x 的最小整数。

[0129] 在ITU G.9903,IEEE P1901.2或G3-PLC标准中,DBPSK调制对应置换矩阵定义为基本置换矩阵,而DQPSK(Differential Quadrature Phase ShiftKeying,差分四相相移键控)调制和D8PSK(差分八相相移键控)调制则分别使用两次和三次该基本置换矩阵。因此,用于DQPSK和D8PSK调制的置换矩阵的维度为 m 列 $n \cdot k$ 行,图6示出了比特被输入G3-PLC标准交织器的缓冲器的顺序(即逐行输入)。如图6所示,被交织的数据存储在输入缓冲器中且维度为 m 列 $n \cdot k$ 行。

[0130] 交织后,用于调制的映射功能逐行读取缓冲器的输出。每个 k 比特作为一组被映射到一个OFDM子载波。

[0131] 图7是G3-PLC RC4模式下置换矩阵的写入和读取顺序的示意图。如图7所示,图中每个方格表示一个比特。每个方格中的数字分别表示输入的比特在输入数列(输入向量)和输出数列(输出向量)中的位置。整个过程是将输入比特按输入向量序列排列,并按图示将相同的比特重复4次后组成交织输入矩阵进行交织,然后将交织输出矩阵中的元素按输出向量顺序输出。这样,输入数列只有 L 个比特,经过重复四次和交织后,输出数列将变为 $4L$ 个比特。

[0132] 在本发明中,交织器引擎直接完成RC4(重复4次)操作而无需实际上复制每个比特四次,但对于编码器RC6(FCH)则需要在物理上复制每个比特6次以形成交织器输入向量来执行交织置换。在RC4模式中,交织器输入向量的序号可用 $p = \text{floor}((j \times m + i) / 4)$ 来算($k=1$, DBPSK),而且 m 必须是4的倍数,如 $m=36$ 或 72 。

[0133] 下面将描述ITU G.9902交织器和ITU G.9903(或G3-PLC)交织器交织指数的生成。

[0134] ITU G.9902和G3-PLC交织器共用相同的初始置换矩阵,即 $n \cdot m$ 的置换矩阵。其输入(交织前)和输出(交织后)指数间的关系由下式决定:

$$I = (i \cdot m_i + J \cdot m_j) \bmod mJ = (j \cdot n_j + i \cdot n_i) \bmod n$$

[0135] 式中的 (m_i, m_j) 和 (n_i, n_j) 的选择如下: $GC^D(m_i, m) = GC^D(m_j, m) = GC^D(n_i, n) = GC^D(n_j,$

$n) = 1$

[0136] 式中的GCD(a,b)表示a和b两个正整数的最大公约数。执行简单的搜索便可基于以下两个参数m和n发现一组适合的上述参数,其中,m为每个正交复用符号包含的子载波数,n为交织的数据块中的正交复用符号的数目。

[0137] 在发射机中,需要逐个符号地将已经交织的比特映射为子载波星座,给出(I,J)。为了实施置位(in-place)交织器,需要获得如下的反向交织器的公式: $i = (I \times m_I + J \times m_J) \bmod m$
 $j = (i \times n_I + J \times n_J) \bmod n$

[0138] 式中 m_I 、 m_J 、 n_I 和 n_J 可由如前所述的公式确定。

[0139] 为了避免在取模操作中计算(i,j),需要获得从 $J=0, I=0, 1, \dots, m-1$ 开始计算的递归(迭代)算法,然后计算 $J=1, 2, \dots, n-1$,以确保在每次迭代期间计算的指数i或j只需确定其是否超出各自的取模界限。如果超出了,则执行一次减法。最后,从上述初始置换矩阵中任意一个比特的的位置坐标(i,j)与该比特在交织后的置换矩阵中的位置坐标(I,J)的关系式可获得i和j的迭代算法。

[0140] 在本发明的另一实例中,上述交织器输入和输出向量指数的生成过程如图8所示:

[0141] 首先,进行初始化,使得 $(i,j)=(0,0)$, $(I,J)=(0,0)$, $B=0$;然后将Bit赋予0值,设置交织器输出向量序号的值为 $P=B+Bit$;交织器输入向量序号 p 可以用 $p=M2V(i,j,m,k)$ 来计算;如果这是PLC-G3RC4模式,则将 p 向右移位两位比特(除4);如果交织器是ITU G.9902模式,则将对 p 循环操作。到此,交织器已算好了输出向量序号P和与其对应的输入向量序号 p ,从而可以输出一个交织好的比特,然后判断是否有更多的OFDM符号要处理,如果是的,执行函数 $(i,j)=IVL_IDX_INC(i,j,m,n,step_mod_j,step_row_i,step_row_j)$ 来进行迭代运算得到(i,j),该函数的具体运算如图9所示,并更新交织器输出向量序号P,使得 $P=P+k$ (如上所示,k为每个子载波星座符号的比特数),将子载波指数I自动加1后,判断其是否还小于m,如果判断结果为是,则返回重新将其值赋予交织器的输入向量序号 p ,然后重复上述后续运算;如果为否,则说明 $J=J+1$,则执行函数 $(i,j)=IVL_IDX_INC(i,j,m,n,step_mod_j,step_col_i,step_col_j)$,更新(i,j)的值;接着将Bit自动加1,判断其是否还小于k,如果小于k,则返回重新将其值赋予交织器的输出向量P,然后重复上述后续运算,如果自动加1后的Bit值大于等于k,则更新交织器的输出向量的下一块的起始地址 $B=B+m*k$,重复整个过程求该符号的(i,j)值。如果所有的OFDM符号都已计算,则结束本流程。

[0142] 其中,m为上述初始矩阵的列数(即子载波数),n为其行数(即正交频分复用符号数); $step_col_i=m_I$,是i对应的每个I增量的增量步长(Incremental step for each I increment for i index); $step_col_j=m_J$,是j对应的每个J增量的增量步长(Incremental step for each J increment for i index); $step_row_i=\tilde{n}_I$,是j对应的每个I增量的增量步长(Incremental step for each I increment for j index); $step_row_j=\tilde{n}_J$,是j对应的每个J增量的增量步长(Incremental step for each J increment for j index); $step_mod_j=n_q$,是由于j的迭代溢出引入的修正系数(Modification factor due to i iteration overflow for j index)。

[0143] 在本实施方式中,函数 $(i,j)=IVL_IDX_INC$ 的具体执行过程如图9(a)所示(以参数(i,j,m,n,step_mod_j,step_row_i,step_row_j)为例),首先调用函数ACS(j,step_row_j,n),将输入的参数j、step_row_j和n进行运算(如图9(b)所示),得到符合条件的 $j < n$ 的j值,

然后判断i值加i的步长(step_row_i)后是否大于等于m值,如果判断结果为是,则将i值减去m值,再次调用函数ACS,输入新的j以及step_mod_j和n,然后返回函数(i,j)=IVL_IDX_INC,该函数执行下一步,如果判断结果为否,则直接返回函数(i,j)=IVL_IDX_INC,该函数执行下一步。

[0144] 该交织方法在计算交织前后数据的对应关系时,以加法取代了原先算法中的乘法,因为加法运算所需要的硬件开销远小于乘法运算,加法运算的速度也远快于乘法运算,所以大大节省了硬件开销,加快了运算速度。并通过使用简单的硬件在传输过程中分别计算数据地址,消除了与可编程处理器和预先计算的程序地址存储器相关的开销(如查表法),极大的节省了计算机在交织过程中的计算量,提高了计算效率。

[0145] 本发明第二实施方式涉及一种通信设备中的数据块解交织方法。图3是该通信设备中的数据块解交织方法的流程示意图。该通信设备中的数据块解交织方法包括以下步骤:

[0146] 在步骤201中,获取已交织的置换矩阵。

[0147] 此后进入步骤201,将已交织的置换矩阵进行解交织,其中,已交织的置换矩阵中任意一个比特的位坐标(I,J)与该比特在解交织后的置换矩阵中的位坐标(i',j')的关系为:

[0148] 如果I=0则 $i'(0,J) = (i'(m-1,J-1) + m_j) \bmod m$

$$[0149] \quad j'(0,J) = \begin{cases} (j'(m-1,J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(m-1,J-1) + m_j \geq m \\ (j'(m-1,J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(m-1,J-1) + m_j < m \end{cases}$$

[0150] 如果I≠0则 $i'(I,J) = (i'(I-1,J) + m_I) \bmod m$,

$$[0151] \quad j'(I,J) = \begin{cases} (j'(I-1,J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(I-1,J) + m_I \geq m \\ (j'(I-1,J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(I-1,J) + m_I < m \end{cases}$$

[0152] 其中,i'(I,J)为交织后的置换矩阵中位坐标为(I,J)的比特在解交织后的置换矩阵中的行坐标,j'(I,J)为交织后的置换矩阵中位坐标为(I,J)的比特在解交织后的置换矩阵中的列坐标,i'(0,J)表示已交织的置换矩阵中位坐标为(0,J)的比特在解交织后的置换矩阵中的行坐标,j'(0,J)表示已交织的置换矩阵中位坐标为(0,J)的比特在解交织后的置换矩阵中的列坐标,i'=0,1,⋯,m-1,j'=0,1,⋯,n-1,I=0,1,⋯,m-1,J=0,1,⋯,n-1;i'(m-1,J-1)和j'(m-1,J-1)分别表示在已交织的置换矩阵中位坐标为(m-1,J-1)的比特在解交织后的置换矩阵中的行坐标和列坐标,i'(I-1,J)表示在已交织的置换矩阵中位坐标为(I-1,J)的比特在解交织后的置换矩阵中的行坐标,j'(I-1,J-1)表示在已交织的置换矩阵中位坐标为(I-1,J-1)的比特在解交织后的置换矩阵中的列坐标,相对于已交织的置换矩阵中位坐标为(I,J)的比特,在对其进行解交织时,这四个坐标已知;m和n分别为初始置换矩阵的列数和行数,mod表示取模运算,m_I、m_J、n_I和n_J由下列各式确定:m_I=m_i⁻¹ mod m, m_J=(m-m_j') mod m, n_I=(n-n_j') mod n, n_J=n_j⁻¹ mod n

[0153] 其中,m_i'和m_j'为指定的与m互质的正整数,n_j'为指定的与n互质的正整数,m_i⁻¹为m_i模m条件下的倒数,n_j⁻¹为n_j模n条件下的倒数,且m_i⁻¹和n_j⁻¹分别由下式确定:

$$[0154] \quad (m_i m_i^{-1}) \bmod m = 1$$

[0155] $(n_j n_j^{-1}) \bmod n = 1,$

[0156] 而 $\tilde{n}_I, \tilde{n}_J,$ 和 n_q 分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

[0157] $\tilde{n}_J = (m_J n_J + n_J) \bmod n$

$$n_q = (m \times (n - n_I)) \bmod n.$$

[0158] 在本实施方式中,取模运算通过以下方式实现:

[0159] 将mod两边的数据进行比较,如果mod左边的数据大于或等于mod右边的数据,则用左边的数据减去右边的数据,将所得的差值作为取模运算的结果;如果mod左边的数据小于mod右边的数据,则将左边的数据作为取模运算的结果。

[0160] 此外,在本实施方式中,上述数据块包括n个正交频分复用符号,每个正交频复用符号包含m个子载波。

[0161] 此后进入步骤203,输出解交织后的置换矩阵。

[0162] 此后,结束本流程。

[0163] 在本实施方式中,上述数据块解交织方法用于ITU G.9902,ITU G.9903,IEEE P1901.2或G3-PLC前向纠错标准。

[0164] 该解交织方法在计算解交织前后数据的对应关系时,以加法取代了原先算法中的乘法,因为加法运算所地而要硬件开销远小于乘法运算,加法运算的速度也远快于乘法运算,所以大大节省了硬件开销,加快了运算速度。通过使用简单的硬件在传输过程中分别计算数据地址,消除了与可编程处理器和预先计算的程序地址存储器相关的开销(如查表法),极大的节省了计算机在解交织过程中的计算量,提高了计算效率。

[0165] 本发明第三实施方式涉及一种通信设备中的数据块交织装置。图10是该通信设备中的数据块交织装置的结构示意图。

[0166] 具体地说,如图10所示,该数据块交织装置包括以下单元:

[0167] 第一获取单元,用于获取交织前的初始置换矩阵。

[0168] 交织单元,用于将初始置换矩阵进行交织,其中,初始置换矩阵中任意一个比特的位坐标(i, j)与该比特在交织后的置换矩阵中的位坐标(I, J)的关系为:

[0169] 如果 $I=0$ 则 $i(0, J) = (i(m-1, J-1) + m_J) \bmod m$

$$j(0, J) = \begin{cases} (j(m-1, J-1) + \tilde{n}_J + n_q) \bmod n & \text{若 } i(m-1, J-1) + m_J \geq m \\ (j(m-1, J-1) + \tilde{n}_J) \bmod n & \text{若 } i(m-1, J-1) + m_J < m \end{cases}$$

[0171] 如果 $I \neq 0$ 则 $i(I, J) = (i(I-1, J) + m_I) \bmod m,$

$$j(I, J) = \begin{cases} (j(I-1, J-1) + \tilde{n}_I + n_q) \bmod n & \text{若 } i(I-1, J) + m_I \geq m \\ (j(I-1, J-1) + \tilde{n}_I) \bmod n & \text{若 } i(I-1, J) + m_I < m \end{cases}$$

[0173] 其中, $i(I, J)$ 为交织后的置换矩阵中位坐标为(I, J)的比特在初始置换矩阵中的行坐标, $j(I, J)$ 为交织后的置换矩阵中位坐标为(I, J)的比特在初始置换矩阵中的列坐标, $i=0, 1, \dots, m-1, j=0, 1, \dots, n-1, I=0, 1, \dots, m-1, J=0, 1, \dots, n-1, m$ 和 n

为初始置换矩阵的列数和行数, mod表示取模运算,

[0174] 且 m_I, m_J, n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \bmod m, m_J = ((m - m_j) \times m_I) \bmod m, n_I = ((n - n_j) \times n_J) \bmod n, n_J = n_j^{-1} \bmod n$

[0175] 其中, m_i 和 m_j 为指定的与 m 互质的正整数, n_j 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数,且 m_i^{-1} 和 n_j^{-1} 分别由下式确定: $(m_i m_i^{-1}) \bmod m = 1$
 $(n_j n_j^{-1}) \bmod n = 1$,

[0176] 而 \tilde{n}_I, \tilde{n}_J , 和 n_q 分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

[0177] $\tilde{n}_J = (m_I n_I + n_J) \bmod n$

$$n_q = (m \times (n - n_I)) \bmod n$$

[0178] 在本实施方式中,该交织单元还包括以下子单元:

[0179] 第一取模子单元,用于将mod两边的数据进行比较,如果mod左边的数据大于或等于mod右边的数据,则用左边的数据减去右边的数据,将所得的差值作为取模运算的结果;如果mod左边的数据小于mod右边的数据,则将左边的数据作为取模运算的结果。

[0180] 第一输出单元,用于输出交织后的置换矩阵。

[0181] 优选地,图11是本实施方式中的一个实际应用的交织器的工作流程示意图。图中虚线代表比特读取地址 (i, j) 产生的流程,实线代表数据比特的流程。当OFDM映射(mapper)单元在处理第 I 个OFDM符号中的第 J 子载波符号 S_{IJ} 时,交织器地址生成单元算出 S_{IJ} 所对应数据的初始置换矩阵位置坐标 (i, j) ,也就是其在输入数列中的位置 $(i + mj)$;交织器读取单元读出比特 b_{ij} ,再由OFDM映射单元转换为子载波符号 S_{IJ} ;当第 I 个OFDM符号的全部子载波映射完之后,就可以送到OFDM调制器进行下一步处理。

[0182] 第一实施方式是与本实施方式相对应的方法实施方式,本实施方式可与第一实施方式互相配合实施。第一实施方式中提到的相关技术细节在本实施方式中依然有效,为了减少重复,这里不再赘述。相应地,本实施方式中提到的相关技术细节也可应用在第一实施方式中。

[0183] 本发明第四实施方式涉及一种通信设备中的数据块解交织装置。图11是该通信设备中的数据块解交织装置的结构示意图。

[0184] 具体地说,如图12所示,该数据块解交织装置包括以下单元:

[0185] 第二获取单元,用于获取已交织的置换矩阵。

[0186] 解交织单元,用于将已交织的置换矩阵进行解交织,已交织的置换矩阵中任意一个比特的的位置坐标 (I, J) 与该比特在解交织后的置换矩阵中的位置坐标 (i', j') 的关系为:

[0187] 如果 $I=0$ 则 $i'(0, J) = (i'(m-1, J-1) + m_J) \bmod m$

$$[0188] \quad j'(0, J) = \begin{cases} (j'(m-1, J-1) + \tilde{n}_J + n_q) \bmod n & \text{若 } i'(m-1, J-1) + m_J \geq m \\ (j'(m-1, J-1) + \tilde{n}_J) \bmod n & \text{若 } i'(m-1, J-1) + m_J < m \end{cases}$$

[0189] 如果 $I \neq 0$ 则 $i'(I, J) = (i'(I-1, J) + m_I) \bmod m$,

$$[0190] \quad j'(I, J) = \begin{cases} (j'(I-1, J-1) + \tilde{n}_I + n_q) \bmod n & \text{若 } i'(I-1, J) + m_I \geq m \\ (j'(I-1, J-1) + \tilde{n}_I) \bmod n & \text{若 } i'(I-1, J) + m_I < m \end{cases}$$

[0191] 其中, $i'(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在解交织后的置换矩阵中的行坐标, $j'(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在解交织后的置换矩阵中的列坐标, $i' = 0, 1, \dots, m-1, j' = 0, 1, \dots, n-1, I = 0, 1, \dots, m-1, J = 0, 1, \dots, n-1, m$ 和 n 为初始置换矩阵的列数和行数, \bmod 表示取模运算,

[0192] 且 m_I, m_J, n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \bmod m, m_J = ((m - m_j') \times m_I) \bmod m, n_I = ((n - n_j') \times n_J) \bmod n, n_J = n_j^{-1} \bmod n$

[0193] 其中, m_i 和 m_j' 为指定的与 m 互质的正整数, n_j' 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定: $(m_i m_i^{-1}) \bmod m = 1, (n_j n_j^{-1}) \bmod n = 1,$

[0194] 而 $\tilde{n}_I, \tilde{n}_J,$ 和 n_q 分别由下列各式确定:

$$\tilde{n}_I = (m_I n_I) \bmod n$$

$$[0195] \quad \tilde{n}_J = (m_J n_J + n_J) \bmod n$$

$$n_q = (m \times (n - n_I)) \bmod n。$$

[0196] 该解交织单元还包括以下单元:

[0197] 第二取模子单元, 用于将 \bmod 两边的数据进行比较, 如果 \bmod 左边的数据大于或等于 \bmod 右边的数据, 则用左边的数据减去右边的数据, 将所得的差值作为取模运算的结果; 如果 \bmod 左边的数据小于 \bmod 右边的数据, 则将左边的数据作为取模运算的结果。

[0198] 第二输出单元, 用于输出解交织后的置换矩阵。

[0199] 优选地, 图13是本实施方式中的一个实际应用的解交织器的工作流程示意图。图中虚线代表比特写入地址 (i, j) 的流程, 实线代表解调的软判决对数似然比 (LLR, log-likelihood ratio) 数据的流程。当OFDM解调器的反射射 (demapper) 单元在处理完第 I 个OFDM符号中的第 J 子载波符号 S_{IJ} 后, 交织器地址生成单元算出 S_{IJ} 所对应数据的初始置换矩阵位置坐标 (i, j) , 解交织器存储单元把 LLR_{IJ} 写入到输出数列位置 $(i+m_j)$ 中; 当整个交织块的数据全部解调和解交织后, 可以进行维特比 (Viterbi) 译码处理。

[0200] 第二实施方式是与本实施方式相对应的方法实施方式, 本实施方式可与第二实施方式互相配合实施。第二实施方式中提到的相关技术细节在本实施方式中依然有效, 为了减少重复, 这里不再赘述。相应地, 本实施方式中提到的相关技术细节也可应用在第二实施方式中。

[0201] 本发明第五实施方式涉及一种正交频分复用通信设备中的数据发送方法。该正交频分复用通信设备中的数据发送方法包括如下的数据块交织步骤:

[0202] 首先, 获取交织前的初始置换矩阵。

[0203] 其次, 将初始置换矩阵进行交织, 其中初始置换矩阵中任意一个比特的的位置坐标 (i, j) 与该比特在交织后的置换矩阵中的位置坐标 (I, J) 的关系为:

[0204] 如果 $I=0$ 则 $i(0, J) = (i(m-1, J-1) + m_j) \bmod m$

$$[0205] \quad j(0, J) = \begin{cases} (j(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(m-1, J-1) + m_j \geq m \\ (j(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i(m-1, J-1) + m_j < m \end{cases}$$

[0206] 如果 $I \neq 0$ 则 $i(I, J) = (i(I-1, J) + m_I) \bmod m$,

$$[0207] \quad j(I, J) = \begin{cases} (j(I-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i(I-1, J) + m_I \geq m \\ (j(I-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i(I-1, J) + m_I < m \end{cases}$$

[0208] 其中, $i(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在初始置换矩阵中的行坐标, $j(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在初始置换矩阵中的列坐标, $i(0, J)$ 表示交织后的置换矩阵中位置坐标为 $(0, J)$ 的比特在初始置换矩阵中的行坐标, $j(0, J)$ 表示交织后的置换矩阵中位置坐标为 $(0, J)$ 的比特在初始置换矩阵中的列坐标, $i=0, 1, \dots, m-1, j=0, 1, \dots, n-1, I=0, 1, \dots, m-1, J=0, 1, \dots, n-1, i(m-1, J-1)$ 和 $j(m-1, J-1)$ 分别表示在交织后的置换矩阵中位置坐标为 $(m-1, J-1)$ 的比特在初始置换矩阵中的行坐标和列坐标, $i(I-1, J)$ 表示在交织后的置换矩阵中位置坐标为 $(I-1, J)$ 的比特在初始置换矩阵中的行坐标, $j(I-1, J-1)$ 表示在交织后的置换矩阵中位置坐标为 $(I-1, J-1)$ 的比特在初始置换矩阵中的列坐标, 相对于交织后在置换矩阵中位置为 (I, J) 的比特, 在对其进行交织时, 这四个坐标已知; m 和 n 分别为初始置换矩阵的列数和行数, \bmod 表示取模运算, m_I, m_J, n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \bmod m, m_J = ((m - m_j) \times m_I) \bmod m, n_I = ((n - n_j) \times n_J) \bmod n, n_J = n_j^{-1} \bmod n$

[0209] 其中, m_i 和 m_j 为指定的与 m 互质的正整数, n_j 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定:

$$[0210] \quad (m_i m_i^{-1}) \bmod m = 1$$

$$[0211] \quad (n_j n_j^{-1}) \bmod n = 1, \tilde{n}_I = (m_I n_I) \bmod n, \tilde{n}_J = (m_J n_J) \bmod n, n_q = (m \times (n - n_I)) \bmod n.$$

[0212] 最后, 输出交织后的置换矩阵。

[0213] 本发明第六实施方式涉及一种正交频分复用通信设备中的数据接收方法。该正交频分复用通信设备中的数据接收方法包括如下数据块解交织步骤:

[0214] 首先, 获取已交织的置换矩阵。

[0215] 然后, 将已交织的置换矩阵进行解交织, 其中, 已交织的置换矩阵中任意一个比特的位坐标 (I, J) 与该比特在解交织后的置换矩阵中的位坐标 (i', j') 的关系为:

[0216] 如果 $I=0$ 则

$$[0217] \quad i'(0, J) = (i'(m-1, J-1) + m_J) \bmod m$$

$$j'(0, J) = \begin{cases} (j'(m-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(m-1, J-1) + m_J \geq m \\ (j'(m-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(m-1, J-1) + m_J < m \end{cases}$$

[0218] 如果 $I \neq 0$ 则 $i'(I, J) = (i'(I-1, J) + m_I) \bmod m$,

$$j'(I, J) = \begin{cases} (j'(I-1, J-1) + \tilde{n}_j + n_q) \bmod n & \text{若 } i'(I-1, J) + m_I \geq m \\ (j'(I-1, J-1) + \tilde{n}_j) \bmod n & \text{若 } i'(I-1, J) + m_I < m \end{cases}$$

[0219] 其中, $i'(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在解交织后的置换矩阵中的行坐标, $j'(I, J)$ 为交织后的置换矩阵中位置坐标为 (I, J) 的比特在解交织后的置

换矩阵中的列坐标, $i'(0, J)$ 表示已交织的置换矩阵中位置坐标为 $(0, J)$ 的比特在解交织后的置换矩阵中的行坐标, $j'(0, J)$ 表示已交织的置换矩阵中位置坐标为 $(0, J)$ 的比特在解交织后的置换矩阵中的列坐标, $i'=0, 1, \dots, m-1, j'=0, 1, \dots, n-1, I=0, 1, \dots, m-1, J=0, 1, \dots, n-1$; $i'(m-1, J-1)$ 和 $j'(m-1, J-1)$ 分别表示在已交织的置换矩阵中位置坐标为 $(m-1, J-1)$ 的比特在解交织后的置换矩阵中的行坐标和列坐标, $i'(I-1, J)$ 表示在已交织的置换矩阵中位置坐标为 $(I-1, J)$ 的比特在解交织后的置换矩阵中的行坐标, $j'(I-1, J-1)$ 表示在已交织的置换矩阵中位置坐标为 $(I-1, J-1)$ 的比特在解交织后的置换矩阵中的列坐标, 相对于已交织的置换矩阵中位置为 (I, J) 的比特, 在对其进行解交织时, 这四个坐标已知; m 和 n 分别为初始置换矩阵的列数和行数, mod 表示取模运算, m_I, m_J, n_I 和 n_J 由下列各式确定: $m_I = m_i^{-1} \text{mod} m, m_J = ((m - m_{j'}) \times m_I) \text{mod} m, n_I = ((n - n_{j'}) \times n_J) \text{mod} n, n_J = n_j^{-1} \text{mod} n$

[0220] 其中, m_i 和 $m_{j'}$ 为指定的与 m 互质的正整数, $n_{j'}$ 为指定的与 n 互质的正整数, m_i^{-1} 为 m_i 模 m 条件下的倒数, n_j^{-1} 为 n_j 模 n 条件下的倒数, 且 m_i^{-1} 和 n_j^{-1} 分别由下式确定: $(m_i m_i^{-1}) \text{mod} m = 1, (n_j n_j^{-1}) \text{mod} n = 1$;

[0221] 而 \tilde{n}_I, \tilde{n}_J 和 n_q 分别由下列各式确定: $\tilde{n}_I = (m_I n_I) \text{mod} n, \tilde{n}_J = (m_J n_J + n_J) \text{mod} n, n_q = (m \times (n - n_I)) \text{mod} n$ 。

[0222] 最后, 输出解交织后的置换矩阵。

[0223] 通过将交织过程分为两个递归阶段, 本发明的交织及解交织方法具有如下特征:

[0224] a) 同时支持 ITU G.9902 和 G.9903 (G3-PLC) 标准的基本交织和解交织功能并将其整合为 DMA (直接存储器访问) 线程;

[0225] b) 支持所有用于 G.9902 的重叠码和用于 G.9903 (G3-PLC) 的 RC4 (ROBO, 鲁棒) 模式;

[0226] c) 支持 G.9902 高阶调制 (QPSK 和 16QAM) 的交织/解交织功能;

[0227] d) 在发送方向上, 来自卷积编码器的编码比特没有被打包而交织后的比特被打包为星座字节 (2 比特用于 QPSK, 4 比特用于 16QAM) 以便于实行星座映射;

[0228] d) 在接收方向, 来自解映射器的软判决对数似然比 (LLR) 没有被打包而保持 8 位比特的长度。解交织后, 对数似然比保持不打包的状态 (比如: 8 比特)。对于重叠码的对数似然比 LLRs, 硬件将组合所有属于同一比特的 LLRs (合并, 如: 作平均或作最佳信噪比合并), 并且维特比译码器 (Viterbi) 将从 LLR 字 (8 个比特) 中取出 6 个比特作为输入进行解码。

[0229] 需要说明的是, 本发明各设备实施方式中提到的各单元都是逻辑单元, 在物理上, 一个逻辑单元可以是一个物理单元, 也可以是一个物理单元的一部分, 还可以以多个物理单元的组合实现, 这些逻辑单元本身的物理实现方式并不是最重要的, 这些逻辑单元所实现的功能的组合才是解决本发明所提出的技术问题的关键。此外, 为了突出本发明的创新部分, 本发明上述各设备实施方式并没有将与解决本发明所提出的技术问题关系不太密切的单元引入, 这并不表明上述设备实施方式并不存在其它的单元。

[0230] 需要说明的是, 在本专利的权利要求和说明书中, 诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来, 而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且, 术语“包括”、“包含”或者其他任何变体意在涵盖非排他性的包含, 从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素, 而且还包括没有明确列出的其他要素, 或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下, 由语句“包括一个”限定的要

素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0231] 虽然通过参照本发明的某些优选实施方式,已经对本发明进行了图示和描述,但本领域的普通技术人员应该明白,可以在形式上和细节上对其作各种改变,而不偏离本发明的精神和范围。

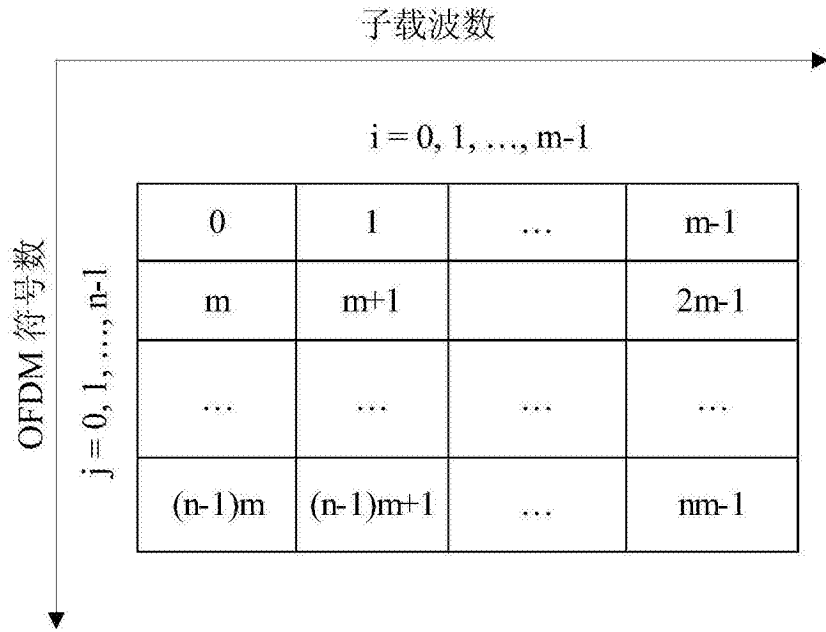


图1

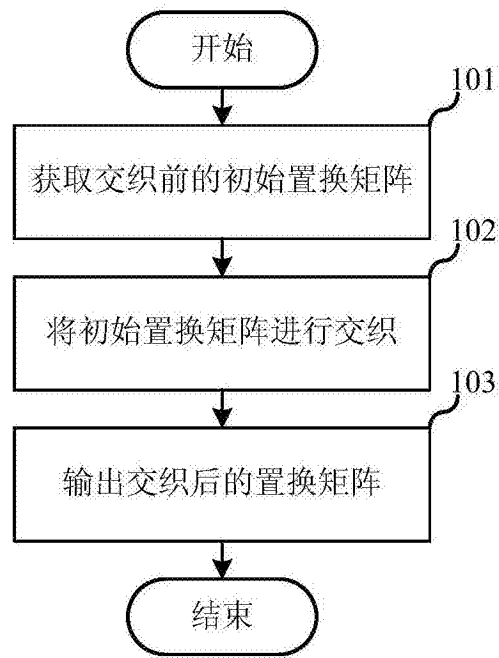


图2

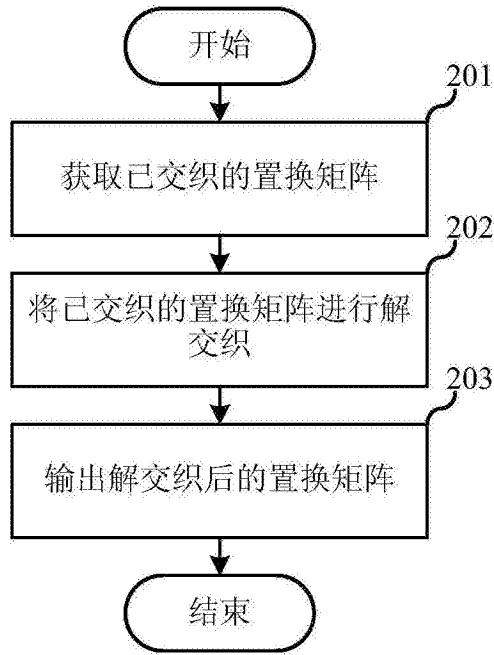


图3

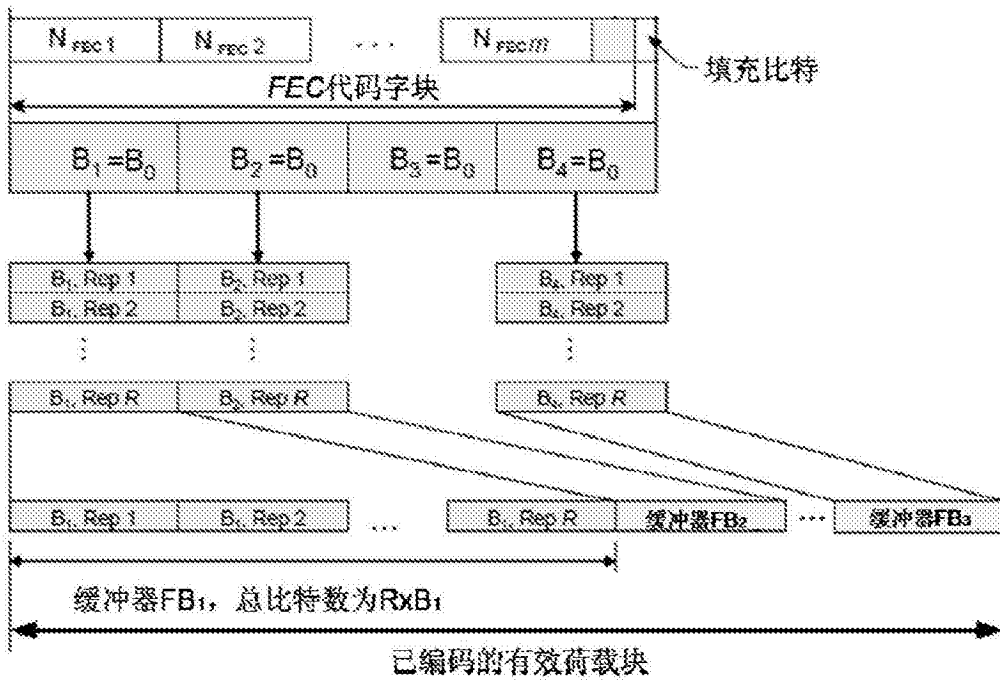


图4

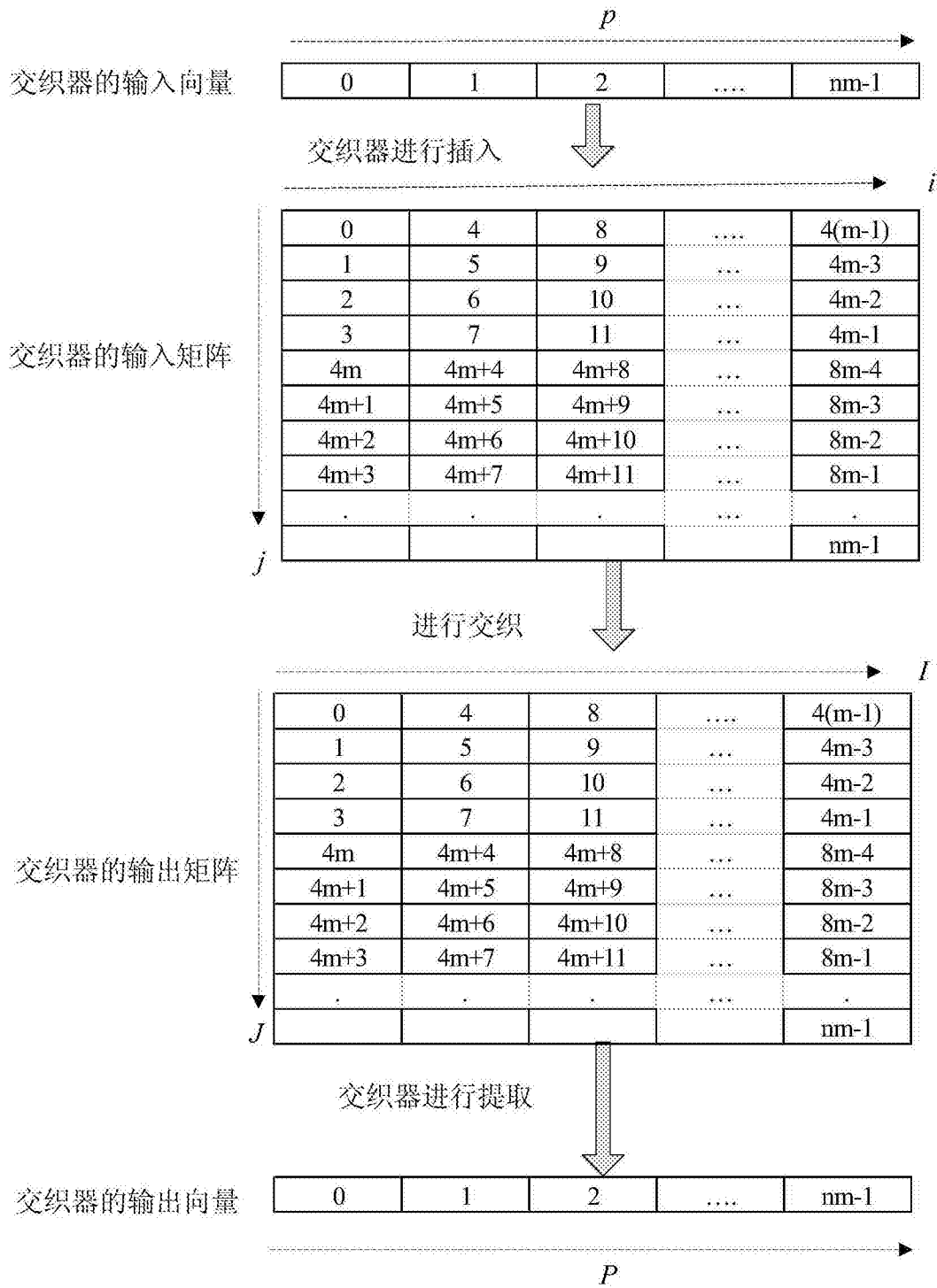


图5

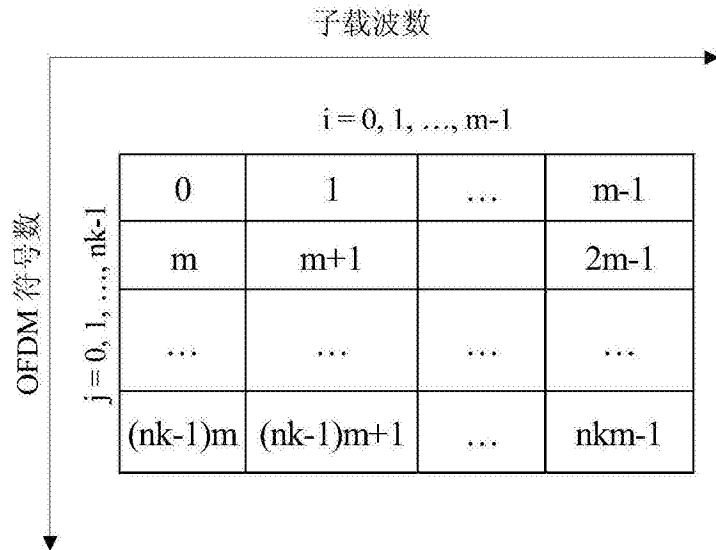


图6

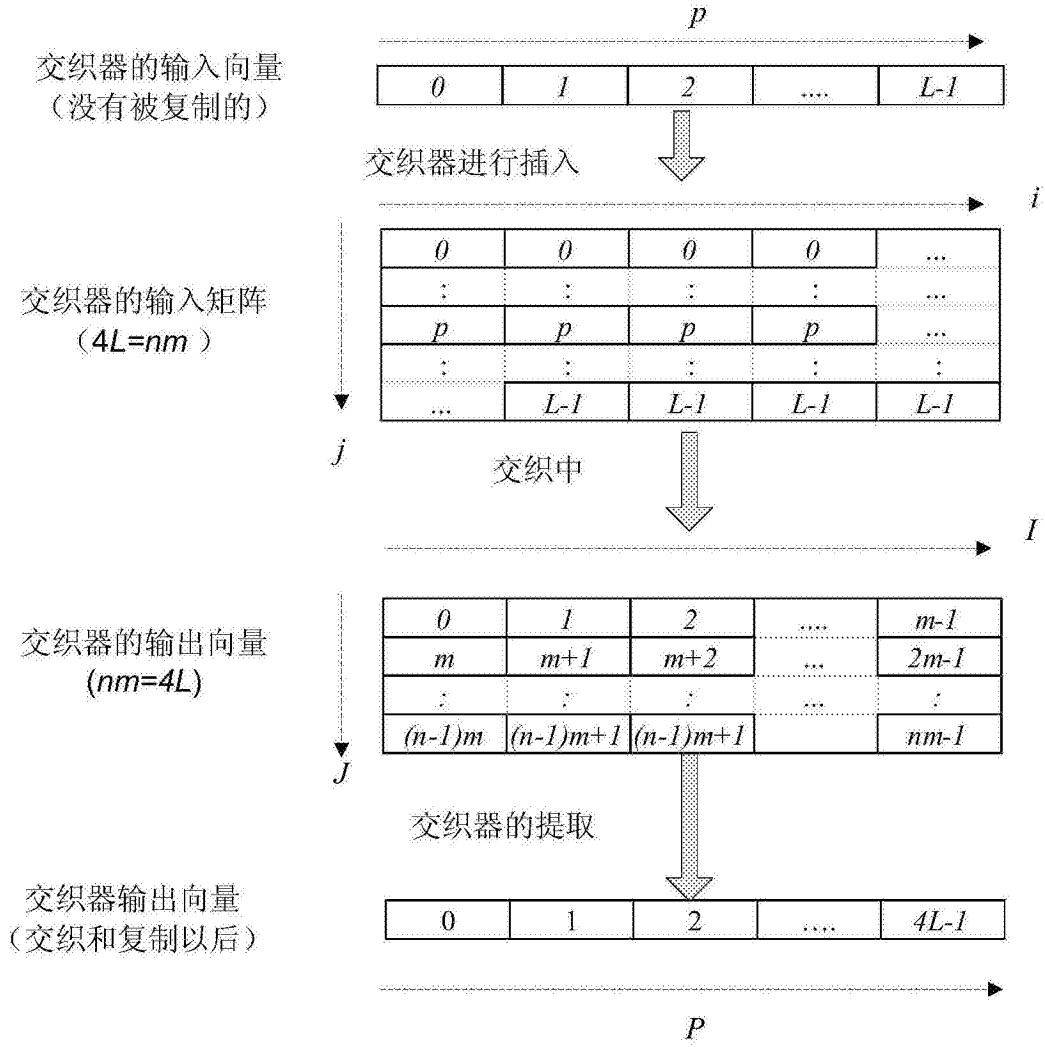


图7

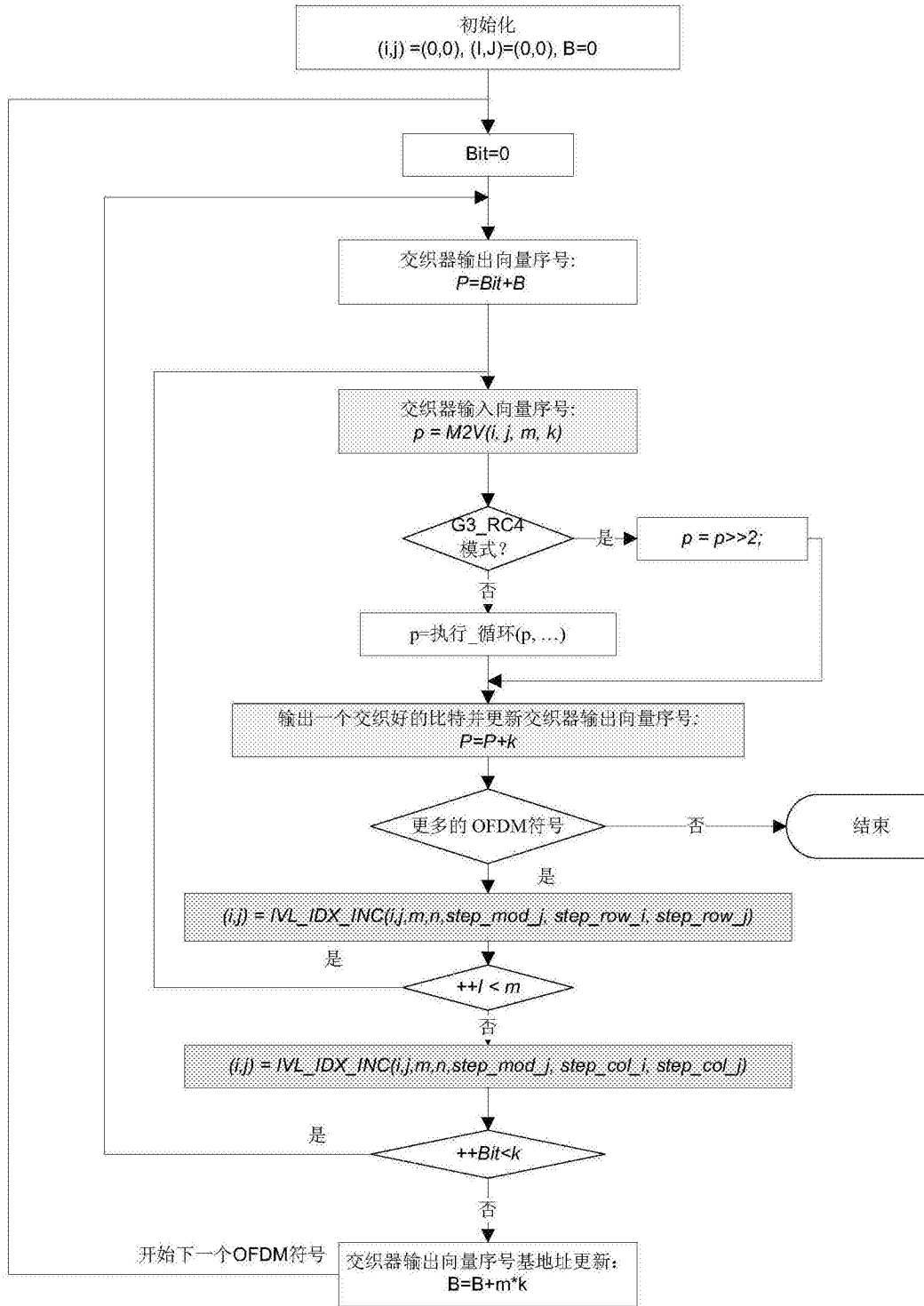


图8

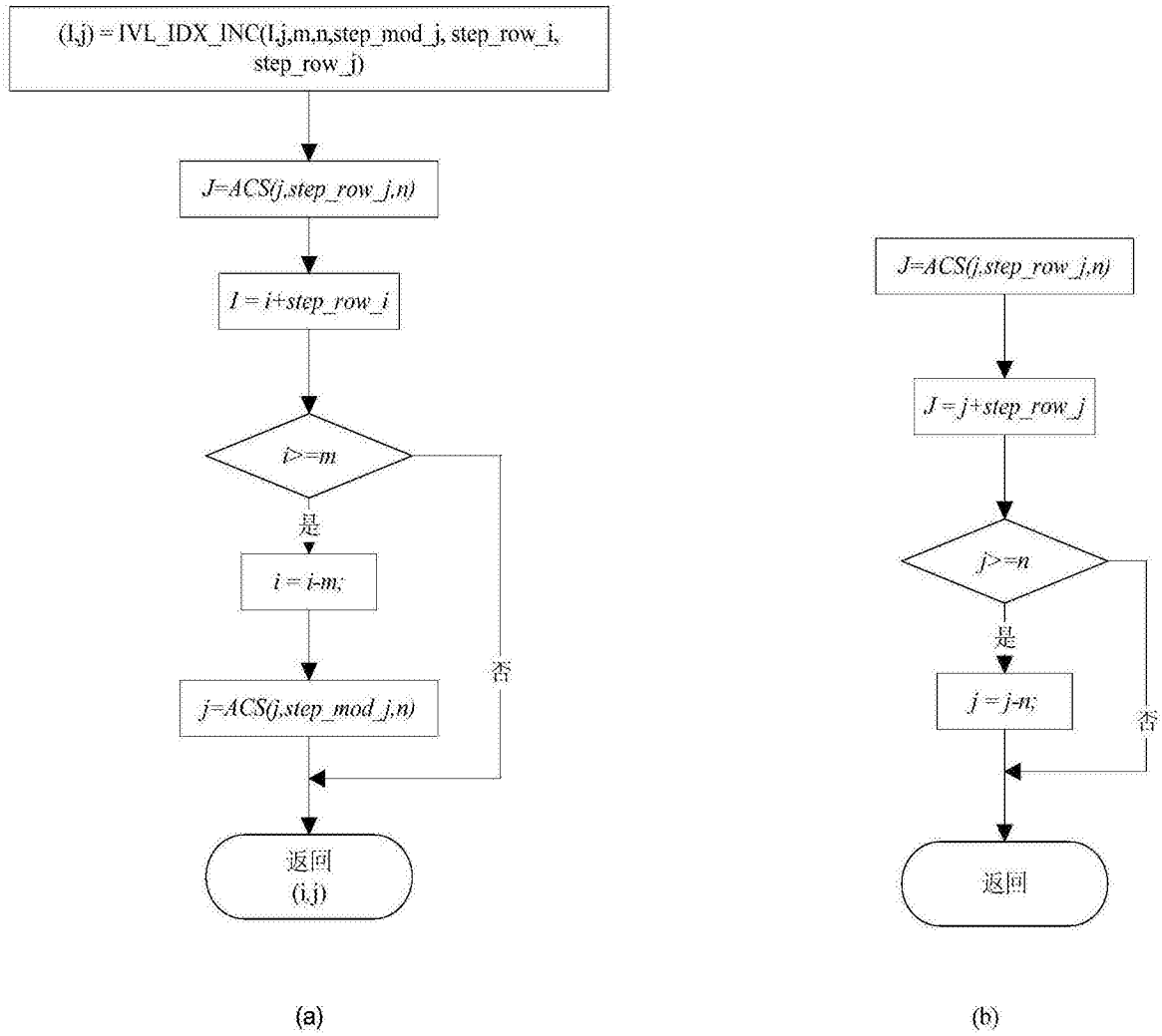


图9

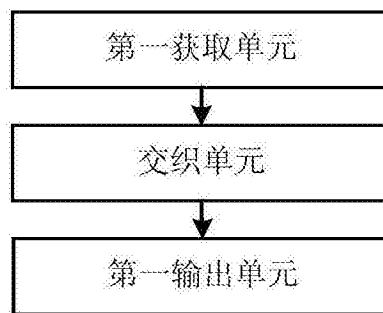


图10

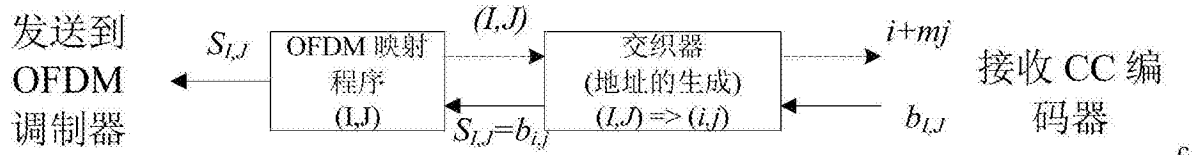


图11

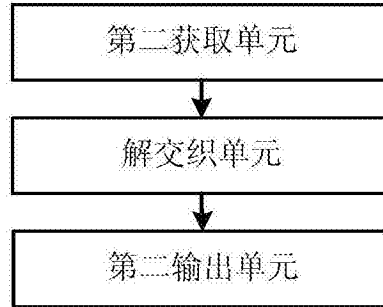


图12

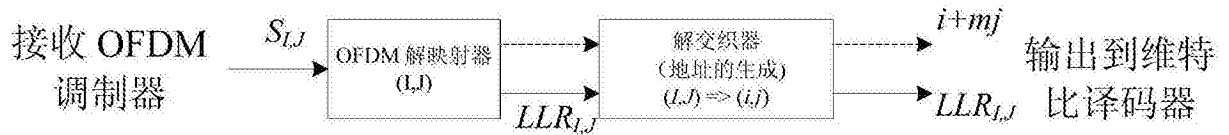


图13