



[12] 发明专利说明书

[21] ZL 专利号 99814039.2

[45] 授权公告日 2005 年 5 月 18 日

[11] 授权公告号 CN 1202625C

[22] 申请日 1999.12.3 [21] 申请号 99814039.2

[30] 优先权

[32] 1998.12.4 [33] US [31] 09/205,511

[86] 国际申请 PCT/US1999/028580 1999.12.3

[87] 国际公布 WO2000/035103 英 2000.6.15

[85] 进入国家阶段日期 2001.6.4

[71] 专利权人 高通股份有限公司

地址 美国加利福尼亚州

[72] 发明人 D·N·罗维奇 林福韵

审查员 王艳坤

[74] 专利代理机构 上海专利商标事务所有限公司

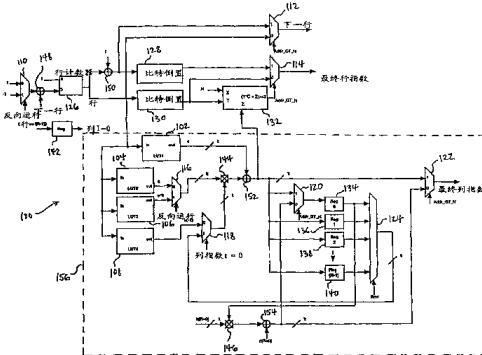
代理人 李玲

权利要求书 4 页 说明书 19 页 附图 3 页

[54] 发明名称 利用线性同余序列的 turbo 码交织器

[57] 摘要

一种利用线性同余序列的 turbo 码交织器 (100)，可以用作 turbo 编码器 (10) 中的两维交织器 (16)，turbo 编码器包括第一和第二组成部分编码器 (12, 14)。交织器 (16) 和第一编码器 (12) 各被配置成接收输入位。第一编码器 (12) 从其产生输出符号 (22, 24)。交织器 (16) 按行依次地接收输入位 (20)。交织器 (16) 内的线性同余序列递归算法的作用是对交织器 (16) 每一行中的位伪随机地重新排列或重排。然后按列依次地从交织器输出位 (26)。第二编码器 (14) 被配置成从交织器接收被交织的位。第二编码器 (14) 从其产生输出符号 (28)。两个输出符号 (22, 24) 流被一起复用，具有适当截去。如果需要的话，能够以逆向产生线性同余递归序列。此外如果需要的话，能够把比特倒置技术用在交织器 (16) 中，对交织器 (16) 的行进行重新排列或重排。



1. 一种 turbo 编码器，包括：

第一编码器，被配置成连续地接收多个输入位并从其产生第一组多个输出符号；

交织器，被配置成连续地接收多个输入位，该交织器包括被安排在具有多行和多列的矩阵中的多个位存储位置和一个线性同余序列发生器，被配置成伪随机地产生交织器每一行中的位重排的序列；以及

第二编码器，被配置成从交织器连续地接收多个被交织的位并从其产生第二组多个输出符号。

2. 如权利要求 1 所述的 turbo 编码器，其特征在于：对位作重排的序列包括按照以下方程式产生的线性同余序列递归：

$$x(n+1) = (ax(n) + c) \bmod M$$

这里 n 代表时间指数， $x(n)$ 代表在时间指数 n 的列指数，a、c 和 M 是整数，M 代表该序列的周期且满足以下条件：

- (i) c 是相对于 M 的素数；
- (ii) $a-1$ 是 p 的倍数，这里 p 代表除尽 M 的任何素数；
- (iii) 当 M 是 4 的倍数时， $a-1$ 必须是 4 的倍数；以及
- (iv) $x(0)$ 是一个整数种子列指数。

3. 如权利要求 2 所述的 turbo 编码器，其特征在于： $a=1$ 。

4. 如权利要求 2 所述的 turbo 编码器，其特征在于：周期 M 是 2 的幂。

5. 如权利要求 2 所述的 turbo 编码器，其特征在于：对于交织器中的每一行， $x(0)$ 等于 0。

6. 如权利要求 1 所述的 turbo 编码器，其特征在于：进一步包括一个复用器，它与第一和第二编码器耦合并被配置成分别从其接收第一和第二组多个输出符号。

7. 如权利要求 1 所述的 turbo 编码器，其特征在于：所示交织器进一步包括至少一个模块，用于按照预先限定的比特倒置算法对交织器的行进行重排。

8. 如权利要求 1 所述的 turbo 编码器，其特征在于：对位作重排的序列

包括按照以下方程式产生的线性同余序列递归：

$$x(n) = (a^{(M/2)-1} x(n+1) - c) \bmod M$$

这里 n 代表时间指数， $x(n)$ 代表在时间指数 n 的列指数，a、c 和 M 是整数，M 代表该序列的周期且满足以下条件：

- (i) c 是相对于 M 的素数；
- (ii) $a-1$ 是 p 的倍数，这里 p 代表除尽 M 的任何素数；
- (iii) 当 M 是 4 的倍数时， $a-1$ 必须是 4 的倍数；以及
- (iv) $x(0)$ 是一个整数种子列指数。

9. 如权利要求 8 所述的 turbo 编码器，其特征在于： $a=1$ 。

10. 如权利要求 8 所述的 turbo 编码器，其特征在于：周期 M 是 2 的幂。

11. 如权利要求 8 所述的 turbo 编码器，其特征在于：对于交织器中的每一行， $x(0)$ 等于 0。

12. 一种对数据元进行交织的方法，包括以下步骤：

通过按行依次地把数据元写到具有多个位存储位置的矩阵；

按照线性同余序列递归把每一行中的数据元伪随机地重新排列在位存储位置的矩阵中；以及

通过从位存储位置的矩阵按列依次地读出数据元。

13. 如权利要求 12 所述的方法，其特征在于：进一步包括按照以下方程式产生线性同余序列递归的步骤：

$$x(n+1) = (ax(n) + c) \bmod M$$

这里 n 代表时间指数， $x(n)$ 代表在时间指数 n 的列指数，a、c 和 M 是整数，M 代表该序列的周期且满足以下条件：

- (i) c 是相对于 M 的素数；
- (ii) $a-1$ 是 p 的倍数，这里 p 代表除尽 M 的任何素数；
- (iii) 当 M 是 4 的倍数时， $a-1$ 必须是 4 的倍数；以及
- (iv) $x(0)$ 是一个整数种子列指数。

14. 如权利要求 13 所述的方法，其特征在于： $a=1$ 。

15. 如权利要求 13 所述的方法，其特征在于：周期 M 是 2 的幂。

16. 如权利要求 13 所述的方法，其特征在于：对于每一行， $x(0)$ 等于 0。

17. 如权利要求 12 所述的方法，其特征在于：进一步按照预先限定的比

特倒置算法对比特存储位置的矩阵的行进行重新排列的步骤。

18. 如权利要求 12 所述的方法，其特征在于：进一步包括按照以下方程式产生线性同余序列递归的步骤：

$$x(n) = (a^{(M/2)-1}x(n+1) - c) \bmod M$$

这里 n 代表时间指数， $x(n)$ 代表在时间指数 n 的列指数，a、c 和 M 是整数，M 代表该序列的周期且满足以下条件：

- (i) c 是相对于 M 的素数；
- (ii) $a-1$ 是 p 的倍数，这里 p 代表除尽 M 的任何素数；
- (iii) 当 M 是 4 的倍数时， $a-1$ 必须是 4 的倍数；以及
- (iv) $x(0)$ 是一个整数种子列指数。

19. 如权利要求 18 所述的方法，其特征在于： $a=1$ 。

20. 如权利要求 18 所述的方法，其特征在于：周期 M 是 2 的幂。

21. 如权利要求 18 所述的方法，其特征在于：对于每一行， $x(0)$ 等于 0。

22. 一种交织器，包括：

通过按行依次地把数据元写到具有多个位存储位置的矩阵的装置；

按照线性同余序列递归把每一行中的数据元伪随机地重新排列在位存储位置的矩阵中的装置；以及

通过从位存储位置的阵列按列依次地读出数据元的装置。

23. 如权利要求 22 所述的交织器，其特征在于：按照以下方程式产生线性同余序列递归：

$$x(n+1) = (ax(n) + c) \bmod M$$

这里 n 代表时间指数， $x(n)$ 代表在时间指数 n 的列指数，a、c 和 M 是整数，M 代表该序列的周期且满足以下条件：

- (i) c 是相对于 M 的素数；
- (ii) $a-1$ 是 p 的倍数，这里 p 代表除尽 M 的任何素数；
- (iii) 当 M 是 4 的倍数时， $a-1$ 必须是 4 的倍数；以及
- (iv) $x(0)$ 是一个整数种子列指数。

24. 如权利要求 23 所述的交织器，其特征在于： $a=1$ 。

25. 如权利要求 23 所述的交织器，其特征在于：周期 M 是 2 的幂。

26. 如权利要求 23 所述的交织器，其特征在于：对于位存储位置的矩阵

中每一行， $x(0)$ 等于 0。

27. 如权利要求 22 所述的交织器，其特征在于：进一步按照预先定义的比特倒置算法对比特存储位置的矩阵的行进行排列的装置。

28. 如权利要求 22 所述的交织器，其特征在于：按照以下方程式产生线性同余序列递归：

$$x(n) = (a^{((M/2)-1)} x(n+1) - c) \bmod M$$

这里 n 代表时间指数， $x(n)$ 代表在时间指数 n 的列指数，a、c 和 M 是整数，M 代表该序列的周期且满足以下条件：

- (i) c 是相对于 M 的素数；
- (ii) $a-1$ 是 p 的倍数，这里 p 代表除尽 M 的任何素数；
- (iii) 当 M 是 4 的倍数时， $a-1$ 必须是 4 的倍数；以及
- (iv) $x(0)$ 是一个整数种子列指数。

29. 如权利要求 28 所述的交织器，其特征在于： $a=1$ 。

30. 如权利要求 28 所述的交织器，其特征在于：周期 M 是 2 的幂。

31. 如权利要求 28 所述的交织器，其特征在于：对于位存储位置的矩阵中每一行， $x(0)$ 等于 0。

利用线性同余序列的 turbo 码交织器

5

发明背景

I. 发明领域

本发明一般涉及通信系统的编码领域，尤其涉及 turbo 编码器的交织器。

II. 现有技术的描述

10 数字数据的传输固有地易于受干扰，这会把差错引入到被传输的数据中。已经提出各种差错检测方案，尽可能可靠地确定差错是否已经引入到被传输的数据中。例如，通常以分组发送数据并在每个分组中增加一个循环冗余校验(CRC)字段，例如 16 位的长度，它载有分组的数据的校验和。当接收机接收数据时，接收机计算接收到的数据上的相同校验和并验证计算的结果是否与 CRC
15 字段中的校验和相同。

当被发送的数据不是在线使用时，当检测到差错时可以请求重新发送有差错的数据。然而，当发送是在线进行时，例如在电话线、蜂窝电话、远程视频系统等上进行时，不可能请求重新发送。

20 卷积码已经被引入以允许数字数据的接收机即使在发送期间已经发生差错时也能够正确地确定被发送的数据。卷积码把冗余引入到被发送的数据中并把被发送的数据打包在分组中，其中每一位的值依赖于序列中的前面几位。因此，当差错发生时，接收机通过追溯收到数据中的可能序列仍然能够推导出原始数据。

25 为了进一步改进发送信道的性能，有些编码方案包括了交织器，它将编码期间分组中的位的次序混合。因此，当干扰毁坏发送期间某些相邻位时，干扰的影响在整个原始分组上被分散，通过解码过程能够更容易地被克服。其他的改进可能包括多分量编码，它以并联或串联对分组编码一次以上。例如，现有技术已知采用差错校正方法，该方法以并联方式使用至少两个卷积码。这种并联编码常常称为 turbo 编码。

30 对于多分量代码，最佳的解码常常是一项十分复杂的任务，可能需要相当

长的时间周期，对于在线解码通常是不能提供的。已经发展了迭代解码技术，以克服这个问题。不是立即确定收到的位是 0 还是 1，接收机每个位指定多级尺度上的一个值，代表该位是 1 的几率。常用标尺（称为对数似然比（LLR）几率）用某些范围内的整数代表每个位，例如 {-32, 31}。31 的值表明被发送的位 0
5 具有很高几率，-32 的值表明被发送的位 1 具有很高几率。0 的值表示该逻辑
位值是不确定的。

把多级尺度上所代表的数据称为“软数据”，迭代解码通常软进/软出，即解码过程接收对应于位值的几率的一个输入序列并提供输出校正的几率，考虑了代码的约束。通常，执行迭代解码的解码器使用来自以前多个迭代的软数据对接收机读出的软数据进行解码。在多分量代码的迭代解码中，解码器使用来自对一个代码的解码的结果来改进对第二个代码的解码。当使用并行解码器时，正如在 turbo 编码中一样，为此可以方便地并行使用两个相应的解码器。对于多次迭代进行这种迭代解码，直至相信软数据接近于代表被发送的数据为止。把具有表示它们接近于 1 的几率的这些位（例如，在上述尺度上的 0 与 31
10 之间）指定为二进制 0，把其余到位指定为二进制 1。
15

“turbo 编码”代表在前向纠错（FEC）方面的重要进展。存在许多种 turbo 编码，但是最常见的 turbo 编码采用多个被交织步骤分开的解码步骤，与迭代解码的使用相结合。这种组合对通信系统中的噪声容限提供了以前不能提供的性能。即，turbo 编码允许在利用现有的前向纠错技术以前是不能接受的每位能量与单位噪声功率谱密度之比 (E_b/N_0) 的电平上进行通信。
20

许多通信系统使用前向纠错技术，因此得益于 turbo 编码的使用。例如，turbo 码能够改善无线卫星链路的性能，其中卫星的有限的下行链路发送功率使接收机系统能够在低 E_b/N_0 电平上工作。

例如，诸如数字蜂窝和 PCS 电话系统的数字无线电信系统也使用前向纠错。例如，通信工业协会已经发布空中接口标准 TIA/EIA 临时标准 95，以及其派生标准，例如 IS-95B（以下共同地称为 IS-95），它定义数字无线通信系统使用卷积编码来提供编码增益从而增大系统的容量。美国专利 5,103,459 中描述了基本上按照使用 IS-95 标准处理射频信号的系统和方法，该专利转让给本发明的受让人，这里通过参考被全部引入。
25

30 通信工业中存在需要继续改善编码增益的推动力。在传统数字无线通信系

统中，已经发现，用同余随机序列可以有利地实施 turbo 编码的串联交织器。现有技术中发现，通过利用线性同余递归算法可以产生均匀随机序列。见例如 D. Knuth “计算机编程的技术” (1969) (描述具有线性同余递归的伪随机数的产生)。还发现采用两维交织器(即组织为包含行和列的矩形数据阵列的交织器) 5 的并联 turbo 编码器通常在编码增益上胜过具有一维交织器(即数据组织为单个线性阵列的交织器)的并联 turbo 编码器。

进一步增强 turbo 编码器的性能是有利的。另外，如果 turbo 编码器比传统编码器实施起来明显地更复杂的话，则需要提供降低复杂性的 turbo 编码器实施方案。因此，需要降低复杂性的使用多个线性同余序列的两维交织器。

10

发明概要

本发明是指使用多个线性同余序列的降低复杂性的两维同余序列。因此，在本发明的一个方面中，turbo 编码器有利地包括第一编码器，被配置成连续地接收多个输入位并从其产生第一组多个输出符号；交织器，被配置成连续地接收多个输入位，该交织器包括以行和列阵列排列的多个位存储位置和一个线性同余序列发生器，被配置成伪随机地产生交织器每一行中的位重排的序列；以及第二编码器，被配置成从交织器连续地接收多个被交织的位并从其产生第二组多个输出符号。
15

在本发明的另一个方面中，一种对数据元进行交织的方法有利地包括通过按行依次地把数据元写到位存储位置的矩阵；按照线性同余序列递归把每一行中的数据元伪随机地重新排列在位存储位置的矩阵中；以及通过从位存储位置的阵列按列依次地读出数据元的步骤。
20

在本发明的再一个方面中，一种交织器有利地包括通过按行依次地把数据元写到位存储位置的矩阵的装置；按照线性同余序列递归把每一行中的数据元伪随机地重新排列在位存储位置的矩阵中的装置；以及通过从位存储位置的阵列按列依次地读出数据元的装置。
25

附图简述

图 1 是并联 turbo 编码器的方框图。

30 图 2 是可以使用在图 1 的并联 turbo 编码器中的交织器的方框图。

图 3 是可以与图 2 交织器结合使用的组成部分编码器的方框图。

较佳实施例的详细描述

按照一个实施例，正如图 1 所示，并联 turbo 编码器 10、或 turbo 编码器 10 包括第一和第二编码器 12, 14、交织器 16 和复用器 18。第一编码器 12 和交织器 16 被配置成接收编码器输入数据 20，这通常是用户信息或控制数据。第一编码器 12 输出系统符号 22，这通常是原始输入位 20 的复制，和奇偶校验符号 24。第二编码器 14 被配置成接收交织器 16 的交织输出 26 和输出第二组奇偶校验符号 28。由第二编码器 14 产生的系统符号(未示出)被抑制，第一 10 和第二编码器 12, 14 的其余各输出 22、24、28 通过复用器 18 被复用为一个输出数据流。

可以并联地增加附加的编码器和交织器对以降低编码率，由此提供增强的前向纠错。另一方面，有些系统符号 22 和/或奇偶校验符号 24 可以被截去以增大编码率和提供改进的频谱效率。

15 第一和第二编码器 12, 14 可以是现有技术中已知的各种类型的编码器，包括块编码器和卷积编码器。在 Bernard Sklar 的“数字通信”245—380(1988)中描述了示例的块编码器和卷积编码器，这里将其引作参考。第一和第二编码器 12, 14 有利地是具有相对较小约束长度 K(例如 K=4)的卷积编码器，由此提供降低的复杂性，因为低的约束长度降低相应解码器(未示出)的复杂性。第一 20 和第二编码器 12, 14 还有利地是递归系统卷积(RSC)编码器，正如现有技术中已知的。交织器 16 有利地是二维交织器，正如以下描述的。

典型地，第一和第二编码器 12, 14 对接收到的每一位 20 输出两个奇偶校验符号 24, 28，对每个编码器 12, 14 产生编码率 R=1/2。然而，对于 turbo 编码器 10 的总编码率是 R=1/3，因为来自第二编码器 14 的系统位被截去。

25 正如图 2 所示，按照一个实施例的两维(2D)线性同余序列(LCS)交织器 100 包括 4 个查看表(LUT)102, 104, 106 和 108, 7 个两输入复用器(MUX)110, 112, 114, 116, 118, 120, 122, 1 个 R 输入 MUX124, 1 个行计数器 126, 第一和第二比特倒置逻辑块 128, 130, 1 个地址生效模块 132, 多个(R 个)列指数或行的寄存器 134, 136, 138, 140(为简单起见图中示出 4 个)，1 个标记列指数复 30 位的寄存器 142, 第一和第二 k 位复用器 144, 146, 以及 4 个 k 位加法器 148,

150, 152, 154。LCS 递归发生器 156 由虚线框表示。交织器 100 可以使用在图 1 的并联 turbo 编码器中，或者，另一方面交织器 100 可以使用在串联 turbo 编码器中，其中交织器 100 定位为外部或内部分量代码，正如本领域技术人员能够理解的。

5 交织器 100 的尺寸是 N，它小于或等于 2^r 且大于 2^{r-1} 。行 R 的数目乘以列 C 的数目等于 2^r 。列 C 的数目等于 2^k ，即 $k = \log_2 C$ 。行 R 的数目等于 2^r ，即 $r = \log_2 R$ 。

地址生效模块 132 可以有利地实施为分立门逻辑，被配置为移位寄存器和加法器。地址生效模块 132 的作用是校验 X 输入是否小于列数目 C 和 Y 输入（行指数）的乘积，与 Z 输入（列指数）的和，执行例如移位和加法功能。地址生效 10 模块 132 的作用是产生一个标记，该标记表示地址是否无效，即地址是否包含应当被放弃的超过 2 的幂的位（即交织器大小是否在相继的 2 的幂之间）。

LCS 递归发生器 156 的作用是通过在至 4 个 LUT102, 104, 106, 108 的输入上接收行数目值并产生一个列指数（Z 输入到地址生效模块 132）而伪随机地重新排列或重排交织器 100 包含的每一行中的位值，正如以下描述的。本领域 15 技术人员应当理解，在诸如图 1 中所示的并联 turbo 编码器中，数据元的物理重新排列可以被有利地回避，有利于利用由第二编码器寻址读出的伪随机产生的 LCS。第一和第二比特倒置逻辑块 128, 130 的作用是按照预先定义的比特倒置规则使交织器 100 中的行重新排列或重排，正如以下描述的和本领域所公知的。

20 LUT102, 104, 106, 108 可以实施为本领域已知的任何存储媒体。第一个 LUT102 用于存储系数 c 的值，第二个 LUT104 用于存储系数 a 的值。第三个 LUT106 用于存储系数 a 取为系数 b 的幂的值。第四个 LUT108 用于存储 $x(-1)$ 的值。每个 LUT102, 104, 106, 108 的大小是 $r \times k$ 位。交织器 100 的总存储器要求对于寄存器 134, 136, 138, 140 是 $4r \times k$ 加上 $r \times k$ 寄存器位。

25 寄存器 142 接收一个规定行数目的位值，初始将其设定为 R-1。采用每个处理循环，寄存器 142 输出一个规定列数目的位值，初始将其设定为不等于 0。由此寄存器 142 的作用是每次当行数目通过所有的行循环时使列指数复位。

采用每个处理循环，输入 MUX110 产生或是 1 或是 -1 的值，这取决于是否 30 设定 RunBackwards 标记。将该值提供给加法器 148，后者将该值加到由 NextRow 表示的位值上。将产生的和提供给行计数器 126 的数据输入端。1 的值提供给

行计数器 126 的第二输入端。行计数器 126 产生一个行值(作为 R-1 初始存储在寄存器 142 中), 将其提供给第二比特倒置逻辑块 130。还将该行值提供给每个 LUT102, 104, 106, 108。还将该行值提供给加法器 150, 后者将该行值加到 1 值上, 并把最终和提供给第一比特倒置逻辑块 128。还将该最终和提供给
5 MUX 的第一输入端。

采用每个处理循环, 第一比特倒置逻辑块 128 提供一个值给 MUX114 的第一输入端。第二比特倒置逻辑块 130 提供一个行指数值给 MUX114 的第二输入端, 以及还提供给地址生效模块 132 的 Y 输入端。地址生效模块 132 在 X 输入端接收值 N。地址生效模块 132 在 Z 输入端基于所存储的系数接收一个值。
10 LCS 地址生效模块 132 计算 C 和 Y 输入值的乘积, 将该乘积加到 Z 输入值, 以及校验结果是否大于或等于 X 输入值。如果计算值大于或等于 N, 地址生效模块 132 输出 1 的值。否则, 输出的值是 0。输出值是表示为 Addr_GT_N 的标记, 当设定为 1 时它表明交织器大小在相继的 2 的幂之间, 使得超出 2 的较低幂的过剩位应当被放弃。

15 Addr_GT_N 值作为选择器输入提供给 MUX112, 114, 120 和 122。如果 Addr_GT_N 值设定为 1, MUX112 选择其第一输入。从 MUX112 输出的所选输入是交织的 NextRow 值。如果 Addr_GT_N 值设定为 1, MUX114 选择其第一输入。从 MUX114 输出的所选输入代表是最后行指数值。

20 LCS 递归发生是按照如下进行的。采用每个处理循环, 代表系数 c 的 k 位值从第一 LUT102 送至数据路径 k 位加法器 152。代表 a 取为幂 b 的值从第三 LUT106 送至 MUX116 的第二输入端。MUX116 在选择器输入端上接收 RunBackwards 标记。如果 RunBackwards 值是 1, MUX116 选择其第二输入并将所选值 (k 位值) 提供给乘法器 144。否则, MUX116 提供其第一输入 (k 位值) 给乘法器 144。值 x (-1) 从第四 LUT108 送至 MUX118 的第一输入端。MUX118 在
25 第二输入端上接收从 MUX124 输出的 k 位值。MUX118 在选择器指数上接收列指数值。列指数值初始设定为不等于 0。如果列指数值设定为 1, MUX118 选择其第二输入。否则, MUX118 选择其第一输入。所选输入值 (k 位值) 提供给乘法器 144。乘法器 144 的所得乘积提供给 k 位加法器 152。有利地, 数据路径 k 位加法器 152 是一个可编程的加法器/减法器, 正如现有技术已知的。当交织器
30 100 反向运行时, 加法器 152 减去值 c。

k 位加法器 152 提供每个处理循环的输出值给地址生效模块 132 的 Z 输入。加法器 152 的输出还提供给 MUX120 的第一输入以及第一至第 (R-1) 行寄存器 136, 138, 140 中的每一个。加法器 152 的输出还作为 k 位输入值提供给 MUX122 的第一输入。

5 MUX120 从 k 位加法器 154 接收第二输入值。如果 MUX120 的选择器输入设定为 1, 那么 MUX120 选择其第一输入。否则, MUX120 选择其第二输入。将所选输入提供给第 0 行寄存器 134。每个行寄存器 134, 136, 138 和 140 提供输出值给 MUX124 的各个输入端。另外, 将第 0 行寄存器 134 的输出值提供给乘法器 146。MUX124 在选择器输入端上接收行值(行计数器 126 的输出)。由 MUX124 10 所选的行寄存器的输入依赖于选择器输入端上的行值的值。因此, 当行值等于各个行寄存器数时每个行寄存器 134, 136, 138 和 140 被更新, 当标记 Addr_GT_N 等于 0 时还使得第 0 行寄存器 134 被使能。

15 将 $R=0$ 的 k 位初始输入值 b 提供给乘法器 146。乘法器 146 还接收从第 0 行寄存器 134 输出的值。乘法器 146 使两个接收值相乘并将结果乘积提供给 k 位加法器 154。数据路径 k 位加法器 154 还接收 $R=0$ 的初始输入值 c 。有利地, 数据路径 k 位加法器 154 是一个可编程的加法器/减法器正如现有技术已知的。当交织器 100 在反向运行时, 加法器 154 减去初始值 c 。加法器 154 使两个接收值相加(或者按照编程相减)。产生的和, 即 k 位值提供给 MUX122 的第二输入端。

20 如果其选择器输入设定为 1, MUX122 选择其第一输入。否则, MUX122 选择其第二输入。MUX122 输出所选的输入, 作为最后列指数值。下一个位值的地址是 R 和从 MUX122 输出的最后行指数值的乘积, 与从 MUX144 输出的最后列指数值相加。

25 在一个实施例中, 按照以下恒等式递归地产生一个具有周期 M 的 LCS:

$$x(n+1) = (ax(n) + c) \bmod M$$

整数 a , c 和 M 满足以下三个条件: (1) c 必须相对于 M 是素数。(2) $a-1$ 必须是 p 的倍数, 这里 p 是除尽 M 的任何素数。当 M 是 4 的倍数时, $a-1$ 必须是 4 的倍数。(3) $x(0)$ 是种子值, 它可以是任何整数。为了简化实施方案, M 可以有利地选为是 2 的幂。因此, a 必须取 $4p+1$ 的形式, 而 c 可以取为任何奇数。30 应当注意, 虽然以上采用 $x(0)$ 来表示初始条件, 在结合图 2 描述的实施例中可

以采用 $x(-1)$ 来表示初始条件。对于所使用的不同数没有附加任何重要性。

按照一个实施例的 2D LCS 交织器作如下规定。让交织尺寸为 $K=2^n$, 交织器规定为具有 R 行和 C 列的矩形矩阵, 这里 R 和 C 二者均由 2 的幂次给出。将待交织的数据按行写到阵列中。按照任何传统交织规则首先对数据的行进行置换(即交织)。有利地, 按照应用于行指数的比特倒置规则对数据的行进行置换。在每一行中, 按照有关的 LCS 规定的规则对列(即数据元, 对于每一行, 每一列有一个数据元)进行置换。与两个不同行相关联的 LCS 有利地是不同的, 但是在另一方面可以是相同的。在对所有的行进行了置换后, 以按列方式读出输出, 产生一个交织序列。正如本领域技术人员应当理解的, 从长度 2^n 的交织器通过删除无效地址能够产生长度小于 2^n 且大于 2^{n-1} 的交织器。

在一个实施例中, 2D LCS 交织器包括以下技术规范。交织器尺寸为 32(即 $N=5$), 数据阵列定义为 $\{d(0), d(1), d(2) \dots d(31)\}$ 。交织器组织为具有 4 行和每一行具有 8 个元素的阵列。以以下方式按行地填入数据元:

$$\begin{pmatrix} d(0) & d(1) & d(2) & d(3) & d(4) & d(5) & d(6) & d(7) \\ d(8) & d(9) & d(10) & d(11) & d(12) & d(13) & d(14) & d(15) \\ d(16) & d(17) & d(18) & d(19) & d(20) & d(21) & d(22) & d(23) \\ d(24) & d(25) & d(26) & d(27) & d(28) & d(29) & d(30) & d(31) \end{pmatrix}.$$

二进制 $(00, 01, 10, 11)$ 中的行指数可以有利地被比特倒置(即 $00, 10, 01, 11$), 相应地对行置换以获得:

$$\begin{pmatrix} d(0) & d(1) & d(2) & d(3) & d(4) & d(5) & d(6) & d(7) \\ d(16) & d(17) & d(18) & d(19) & d(20) & d(21) & d(22) & d(23) \\ d(8) & d(9) & d(10) & d(11) & d(12) & d(13) & d(14) & d(15) \\ d(24) & d(25) & d(26) & d(27) & d(28) & d(29) & d(30) & d(31) \end{pmatrix}.$$

比特倒置的作用是按照预定的比特倒置算法重排交织器的行。比特倒置算法的应用在交织器的行之间提供所需的时间间隔。然而, 比特倒置不是实施交织器所必须的。

在一个特定实施例中, 按照以下方程式产生置换 LCS:

$$\begin{aligned} x_1(n+1) &= (5x_1(n)+7)_{\text{mod } 8}, \text{ with } x_1(0)=3, \\ x_2(n+1) &= (x_2(n)+5)_{\text{mod } 8}, \text{ with } x_2(0)=0, \\ x_3(n+1) &= (5x_3(n)+3)_{\text{mod } 8}, \text{ with } x_3(0)=4, \end{aligned}$$

和

$$x_4(n+1) = (x_4(n) + 3)_{\text{mod } 8}, \text{ with } x_4(0) = 3.$$

对于 4 行，置换图案分别由 $\{3, 6, 5, 0, 7, 2, 1, 4\}$ 、 $\{0, 5, 2, 7, 4, 1, 6, 3\}$ 、 $\{4, 7, 6, 1, 0, 3, 2, 5\}$ 和 $\{7, 2, 5, 0, 3, 6, 1, 4\}$ 给出。因此，在应用列置换后，第一行变为

5 $(d(3) \ d(6) \ d(5) \ d(0) \ d(7) \ d(2) \ d(1) \ d(4)),$

第二行变为

$$(d(16) \ d(21) \ d(18) \ d(23) \ d(20) \ d(17) \ d(22) \ d(19)),$$

第三行变为

$$(d(12) \ d(15) \ d(14) \ d(9) \ d(8) \ d(11) \ d(10) \ d(13)),$$

10 第四行变为

$$\{d(31) \ d(26) \ d(29) \ d(24) \ d(27) \ d(30) \ d(25) \ d(28)\}.$$

在具有各个行中对所有的列进行置换后，交织后的数据阵列具有以下形式：

$$\begin{pmatrix} d(3) & d(6) & d(5) & d(0) & d(7) & d(2) & d(1) & d(4) \\ d(16) & d(21) & d(18) & d(23) & d(20) & d(17) & d(22) & d(19) \\ d(12) & d(15) & d(14) & d(9) & d(8) & d(11) & d(10) & d(13) \\ d(31) & d(26) & d(29) & d(24) & d(27) & d(30) & d(25) & d(28) \end{pmatrix}.$$

15 交织后的矩阵中的数据是按列读出的，产生以下的交织序列：
 $\{d(3), d(16), d(12), d(31), d(6), d(21), d(15), d(26), d(5), d(18), d(14) \dots \dots$
 $d(11), d(30), d(1), d(22), d(10), d(25), d(4), d(19), d(13), d(28)\}$ 。如果需要
 长度 30 的交织器，通过删除数据元 $d(30)$ 和 $d(31)$ 能够缩短按照如上所述产生的
 的交织器，产生以下交织序列： $\{d(3), d(16), d(12), \cancel{d(31)}, d(6), d(21), d(15),$

20 $d(26), d(5), d(18), d(14) \dots \dots d(11), \cancel{d(30)}, d(1), d(22), d(10), d(25), d(4),$
 $d(19), d(13), d(28)\}$ 交织器构造中采用的 LCS 根据需要既能正向产生也能反
 向产生，供 turbo 解码中最佳使用 MAP 解码器。在一个实施例中，倒置序列产
 生是通过以下方程式给出的：

$$x(n) = (a^\beta x(n+1) - c)_{\text{mod } M}$$

25 这里

$$\beta = (M/2) - 1$$

应当注意，项 β 正如以上方程式使用的，代表结合图 2 实施例描述的系数 b。

因此，LCS 的产生需要每个交织器由 $3R$ 参数唯一地定义，这里 R 是行的数目。需要一个相对短的 $\log_2(C) \times \log_2(C)$ 乘法器。由于模量运算，不必产生 5 比特位置 $\log_2(C)$ 之上的各比特。需要一组 R 寄存器来保存 R 同余序列的中间结果。

对于每一行采用不同的参考 $x(0)$ 、 a 和 b ，对于置换序列存在许多不同可能性。需要进行一次搜索，使供特定 turbo 码一起使用的交织器参数最佳化。

在图 3 中，对于 CDMA 数字无线通信系统中使用的特定 turbo 码，使按照 10 一个实施例的组成部分编码器 200 最佳化。编码器 200 包括 7 个模 2 加法器 202、204、206、208、210、212、214 和 3 个比特位置 216、218、220。比特位置 216、218、220 可以作为一个 3 位寄存器或者另一方面作为 3 个 1 位寄存器实施。模 2 加法器 202、204、206、208、210、212、214 以精确方式被耦合到比特位置 216、218、220，从而产生一组所需反馈抽头。因此加法器 202 配置成接收一个输入位。加法器 202 还与比特位置 216 以及加法器 204 和 206 耦合。比特位置 216 与比特位置 218 以及加法器 204 和 206 耦合。比特位置 218 与比特位置 220 以及加法器 208 和 210 耦合。加法器 210 与加法器 202 耦合。加法器 204 与加法器 212 耦合。加法器 206 与加法器 208 耦合。加法器 208 与加法器 214 耦合。比特位置 220 与加法器 210、214 和 212 耦合。加法器 212，15 214 被配置为分别输出第一和第二符号。

现有技术中已知，用组成部分编码器（未示出）中差错事件的输入和输出权重能够表征差错性能。例如见 S. Benedetto & G. Montorsi “显露 turbo 码：并联编码方案的一些结果” 42 IEEE Trans. Info. Theory 409-28(1996 年 3 月)。差错事件的输入权重是位错的数目，而差错事件的输出权重是代码符号错的数目。输入权重 1 的差错事件清楚地将脱离全 0 状态且决不重新合并（1 将在移位 25 寄存器中无止境地循环，沿路累积越来越多的输出权重）。这是由于编码器的递归或反馈部分造成的。由于这一结构，已经表明，在高信噪比(SNR)下的 turbo 码的性能以具有输出权重 2 的输出差错事件为主。Turbo 码差错下界能够利用所谓的有效自由距离渐近线准确地预计。有效自由距离是输入权重 2 的所有差错事件的最小输出权重。在长度上短的输入权重 2 差错事件典型地将引起最小 30 差错事件的最小输出权重。

距离差错事件。对于图 3 的编码器 200，反馈多项式是 $1+D^2+D^3$ ，所有可能的输入权重 2 差错事件具有形式 $D^k(1+D^{7j})$ ，这里 $j=1, 2, \dots, k$ 是在范围 $0, \dots, K-7j$ 内的任意移位(假设 K 是交织器尺寸)。通过审查组成部分编码器的篱笆图能够方便地验证这点，正如本领域技术人员能够理解的。

5 应当注意，图 3 所示实施例的交织器尺寸表示为 K，而图 2 所示实施例的交织器尺寸表示为 N。本领域技术人员应当明白，对于使用不同字母的事实并不含有特别意义。

例如，假设 $D^k(1+D^7)$ 的差错图案引起第一解码器当中的最小距离差错事件。turbo 交织器将两个差错 (D^k, D^{k+7}) 映射到两个位置 (D^m, D^n) 。如果 $|m-n|=7$ 10 或者 7 的某个倍数，很可能是第二解码器当中的低距离差错事件。turbo 交织器的根本目的是阻止发生这种映射。即，交织器应当有利地把在第一维上倾向于低权重差错事件的位的集合映射到在第二维上产生大量输出权重的位的集合。因此，设计交织器的一种所需方法是试图防止位于指数 $(k, k+7j)$ 上的位对 15 映射到位于指数 $(s, s+7t)$ 上的位对，特别强调的是 j 和 t 具有较小值。这种输入差错事件列在以下的表 1 中。对于每个事件，利用速率 1/2、1/3 和 1/4 正向链路 turbo 码，列出了第一组成部分代码的输出奇偶校验权重，正如 IS-95 中规定的。

表 1 输入权重 2 差错事件

输入差错事件	输入长度	第 1 组成部分输出权重		
		速率 1/2	速率 1/3	速率 1/4
$D^k(1+D^7)$	8	3	6	8
$D^k(1+D^{14})$	15	6	10	16
$D^k(1+D^{21})$	22	7	14	20
$D^k(1+D^{28})$	28	10	18	28

20 如果一个给定交织器包括形式 $D^{k1}(1+D^7) \rightarrow D^{k2}(1+D^7)$ 的输入权重 2→2 的映射，那么产生的差错事件的复合输出权重对于速率 1/2 的 turbo 码将是 $2+3+3=8$ 。在以前的计算中，系统位(2)的权重与来自两个组成部分编码器(3 和 3)的各个奇偶校验权重求和。类似地，如果交织器包括形式

$D^{k1}(1+D^7) \rightarrow D^{k2}(1+D^4)$ 的输入权重 $2 \rightarrow 2$ 的映射，那么产生的差错事件的复合输出权重，对于速率 $1/2$ 的 turbo 码，将是 $2+3+6=11$ 或者是 $2+6+10=18$ ，对于速率 $1/3$ 的 turbo 码。

此外，低组成部分输出权重的输入权重 4 差错事件映射到第二维中低输出权重的 2 个输入权重 2 差错事件是可能的。这种映射由 $4 \rightarrow \{2, 2\}$ 表示。虽然对于输入权重 4 的组成部分差错事件不存在闭合形式解决办法，以下的表格包含组成部分编码器 200 的某些低输出权重差错事件，组成部分编码器 200 被用作 CDMA 数字无线通信系统中前向链路 turbo 码的第一组成部分编码器，该系统采用从 IS-95 导出的空中接口。

表 2：输入权重 4 差错事件

输入差错事件	输入长度	第1组成部分输出权重		
		速率1/2	速率1/3	速率1/4
$D^k(1+D^3+D^4+D^5)$	5	2	4	6
$D^k(1+D^1+D^2+D^6)$	6	2	4	8
$D^k(1+D^2+D^4+D^6)$	8	2	4	8
$D^k(1+D^1+D^4+D^7)$	9	2	4	8
$D^k(1+D^3+D^{10}+D^{11})$	11	2	8	12
$D^k(1+D^1+D^2+D^8)$	4	3	4	6
$D^k(1+D^2+D^3+D^9)$	6	3	4	8
$D^k(1+D^1+D^6+D^9)$	10	3	4	8
$D^k(1+D^4+D^6+D^9)$	9	3	8	10
$D^k(1+D^1+D^2+D^{10})$	13	3	8	12
$D^k(1+D^3+D^7+D^{10})$	13	3	8	12
$D^k(1+D^1+D^4+D^{10})$	15	3	8	12
$D^k(1+D^4+D^{10}+D^{11})$	15	3	8	12
$D^k(1+D^4+D^6+D^{11})$	12	3	8	14
$D^k(1+D^1+D^{12}+D^{13})$	13	3	8	14
$D^k(1+D^4+D^{10}+D^{13})$	15	3	8	14
$D^k(1+D^1+D^7+D^9)$	8	4	4	8
$D^k(1+D^3+D^4+D^9)$	9	4	8	10
$D^k(1+D^1+D^2+D^{11})$	11	4	8	10
$D^k(1+D^1+D^4+D^{11})$	11	4	8	10
$D^k(1+D^3+D^2+D^9)$	9	4	8	12
$D^k(1+D^3+D^7+D^{10})$	10	4	8	12
$D^k(1+D^4+D^6+D^{10})$	10	4	8	12
$D^k(1+D^3+D^5+D^{11})$	13	4	8	12
$D^k(1+D^3+D^6+D^{11})$	13	4	8	12
$D^k(1+D^4+D^7+D^{11})$	13	4	8	12
$D^k(1+D^1+D^6+D^{11})$	17	4	8	12
$D^k(1+D^8+D^{10}+D^{11})$	17	4	8	12
$D^k(1+D^2+D^4+D^{11})$	12	4	8	14
$D^k(1+D^3+D^7+D^{12})$	12	4	8	14
$D^k(1+D^1+D^6+D^{12})$	15	4	8	14
$D^k(1+D^3+D^6+D^{12})$	15	4	8	14
$D^k(1+D^6+D^{13}+D^{14})$	16	4	8	14
$D^k(1+D^3+D^{13}+D^{14})$	16	4	12	18
$D^k(1+D^3+D^{10}+D^{11})$	19	4	12	18
$D^k(1+D^4+D^6+D^{11})$	19	4	12	18

例如，如果交织器包括形式 $D^{k1}(1+D^3+D^4+D^5) \rightarrow \{D^{k2}(1+D^7) D^{k3}(1+D^7)\}$ 的输入权

重 $4 \rightarrow \{2, 2\}$ 映射，那么产生的差错事件的复合输出权重将是 $4+2+3+3=12$ ，对于速率 $1/2$ 的 turbo 码。这一复合差错事件不比由于映射 $D^{k_1}(1+D^7) \rightarrow D^{k_2}(1+D^{14})$ 造成的差错事件差很多，后者具有复合输出权重 11。因此，主要设计目的是使交织器的参数最佳化，从而避免上述类型的不良映射或者使其减至最低。理想 5 地，具有最低复合输出权重的不良映射是对避免或减至最小的最重要的映射。在设计特定尺寸（例如 1530）的交织器中，有可能使交织器参数最佳化从而使两者类型的映射（即权重 $2 \rightarrow 2$ 和权重 $4 \rightarrow \{2, 2\}$ ）减至最小。这种方法产生该特定尺寸的最佳交织器。应当指出，在尺寸 2^n 的交织器的设计中，该尺寸能够被稳健地截短到大于 2^{n-1} 的任何尺寸（将该交织器称为“截短友好”交织器），权重 10 $4 \rightarrow \{2, 2\}$ 映射的最佳化可能更难以实现。

进行搜索从按照图 3 所示实施例的 2D LCS 交织器获得尺寸 2^n 的截短友好的交织器。以下的表 3 包括搜索的原始结果。对于每个交织器尺寸，所用的行的数目和所用的列的数目随同 $x(0)$ 、 a 和 c 系数一起规定。为实施的简单和效率起见，对于所有交织器都使用 32 行。

表 3: 2D LCS 交织器系数

K=512, 32x16			K=1024, 32x32			K=2048, 32x64			K=4096, 32x128		
x(0)	A	c	x(0)	a	c	x(0)	a	c	x(0)	a	c
14	9	11	30	29	9	24	45	63	95	61	63
6	1	13	17	29	5	41	53	53	7	121	119
6	1	11	19	9	1	46	53	41	29	113	105
10	13	15	31	1	17	46	5	43	28	29	53
6	5	13	19	17	31	9	29	13	16	25	97
7	9	11	11	13	19	15	37	39	69	33	33
1	9	5	18	13	21	13	53	19	41	93	81
4	9	5	20	9	27	28	1	57	34	1	117
14	5	9	31	21	11	20	29	5	100	69	75
11	13	9	29	21	17	1	49	39	69	113	39
12	9	5	26	25	21	15	33	31	40	13	15
10	9	1	19	29	15	9	33	19	13	33	19
10	13	13	11	13	7	30	21	5	23	77	123
1	5	5	1	1	27	61	37	45	84	9	127
11	1	1	24	13	31	12	45	7	87	65	113
2	5	11	10	5	29	9	37	19	16	17	81
4	9	11	23	29	29	12	37	53	85	117	69
15	9	5	26	1	15	59	13	25	103	41	73
14	1	3	28	21	23	18	25	37	50	73	5
9	1	9	11	21	11	9	49	31	91	73	35
7	1	7	16	21	3	17	45	29	59	73	59
13	13	13	17	13	11	50	25	23	87	97	39
6	13	9	10	21	13	26	21	41	47	125	3
1	13	15	13	17	7	14	61	45	34	21	33
7	1	7	24	9	5	12	25	37	120	57	103
2	5	15	17	25	1	34	33	55	76	77	103
5	5	9	10	21	25	10	29	59	19	13	73
0	9	1	28	21	27	16	13	15	49	121	111
10	5	1	13	25	9	29	37	41	4	97	97
9	9	13	21	13	3	33	5	33	13	101	71
3	9	1	4	17	27	30	49	51	73	9	77
11	9	3	18	29	29	0	21	47	102	101	1

表 3: 2D LCS 交织器系数(续)

K=8192, 32x256			K=16384 32x512			K=32768, 32x1024		
X(0)	a	c	X(0)	a	c	X(0)	a	c
40	37	249	181	45	289	939	281	913
73	53	59	75	217	223	9	85	969
88	177	105	251	125	257	463	989	165
46	29	91	89	141	457	363	981	59
199	73	141	304	29	31	637	437	751
211	185	133	328	153	161	599	597	621
85	193	181	428	17	451	971	173	615
18	169	209	394	245	147	1007	157	79
246	81	89	178	321	111	444	693	415
192	85	71	186	217	227	339	205	331
144	189	223	438	357	423	382	293	565
6	129	73	465	85	261	505	981	259
118	69	119	24	185	109	728	845	447
221	217	127	415	417	465	139	953	383
241	109	27	225	137	33	682	537	461
235	209	133	77	61	265	487	293	903
181	189	235	158	45	211	453	9	885
126	245	37	278	373	255	352	729	619
129	173	191	410	117	175	442	81	315
222	9	43	275	253	429	537	945	519
89	113	159	35	85	289	201	13	175
21	13	205	496	249	15	459	745	97
75	253	183	79	365	221	592	69	829
195	149	71	257	449	337	803	909	385
45	101	209	232	81	137	133	425	201
246	193	159	385	121	87	965	109	1001
228	85	97	138	429	73	707	685	105
0	161	151	479	9	23	627	289	97
191	45	31	158	141	51	1019	805	753
162	165	35	442	481	71	101	69	573
196	149	191	464	269	157	521	753	183
94	157	197	413	505	237	290	221	181

在另一个实施例中，系数 a 可以设定等于 1，进行新的搜索以从 2D LCS

交织器获得尺寸 2^N 的截短友好的交织器。正向和反向 LCS 递归方程式分别简化为如下：

$$x(n+1) = (ax(n) + c) \bmod M$$

和

5 $x(n) = (ax(n+1) - c) \bmod M$

以下的表 4 包括搜索的原始结果。为了实施的简单和有效起见，对于所有的交织器都采用 32 行。从表 4 所示的结果本领域技术人员应当理解，在所有 LCS 递归中设定 a 等于 1，在产生的交织器的质量中不引起处罚。此外，从设定 a 等于 1 的简化中获得的复杂性增加是实质性的。例如，不需要在图 2 实施例中所描述的第二和第三 LUT(LUT 用于存储 a 和 a^b 的值)。也不需要在图 2 实施例中所描述的 k 位乘法器。正如从以下的表 4 看到的，对于交织器的每一行，仅仅需要规定原始条件 $x(-1)$ 和附加常数 c 。在该表中未包括较大尺寸的交织器的系数，该表仅仅规定原始搜索结果，因为它们在提交本申请时不能提供。

表4: 2D LCS交织器系数

K=512, 32x16		K=1024, 32x32		K=2048, 32x64		K=4096, 32x128		K=8192, 32x256		K=16384, 32x512		K=32768, 32x1024	
x(-1)	C	x(-1)	C	x(-1)	C	x(-1)	c	x(-1)	c	x(-1)	c	x(-1)	c
1	1	1	3	0	5	-	-	-	-	-	-	-	-
11	9	14	1	1	1	-	-	-	-	-	-	-	-
11	15	5	13	14	1	-	-	-	-	-	-	-	-
1	5	13	27	3	23	-	-	-	-	-	-	-	-
1	5	3	17	9	25	-	-	-	-	-	-	-	-
6	7	3	13	8	3	-	-	-	-	-	-	-	-
6	15	25	5	4	1	-	-	-	-	-	-	-	-
12	11	3	23	57	61	-	-	-	-	-	-	-	-
14	13	2	19	0	5	-	-	-	-	-	-	-	-
15	7	1	13	60	49	-	-	-	-	-	-	-	-
13	15	8	5	1	9	-	-	-	-	-	-	-	-
4	5	25	9	47	55	-	-	-	-	-	-	-	-
15	5	14	9	20	25	-	-	-	-	-	-	-	-
14	5	11	23	34	19	-	-	-	-	-	-	-	-
12	3	17	13	46	1	-	-	-	-	-	-	-	-
3	15	19	13	52	13	-	-	-	-	-	-	-	-
14	13	4	39	8	75	-	-	-	-	-	-	-	-
8	19	17	35	35	67	-	-	-	-	-	-	-	-
12	15	24	27	48	51	-	-	-	-	-	-	-	-
14	11	28	23	57	47	-	-	-	-	-	-	-	-
14	11	29	19	59	35	-	-	-	-	-	-	-	-
6	27	12	51	24	103	-	-	-	-	-	-	-	-
7	31	14	59	29	115	-	-	-	-	-	-	-	-
2	23	4	47	9	91	-	-	-	-	-	-	-	-
1	27	2	55	4	107	-	-	-	-	-	-	-	-
2	15	5	27	11	55	-	-	-	-	-	-	-	-
13	7	26	11	52	19	-	-	-	-	-	-	-	-
9	3	19	3	38	7	-	-	-	-	-	-	-	-
12	11	24	23	49	47	-	-	-	-	-	-	-	-
14	15	28	31	56	63	-	-	-	-	-	-	-	-
11	19	22	35	45	67	-	-	-	-	-	-	-	-
12	15	25	27	50	51	-	-	-	-	-	-	-	-

因此，已经描述利用线性同余序列的新颖和改进的 turbo 码交织器。本领域技术人员应当理解，虽然这里揭示的实施例已经描述了在蜂窝电话系统的方面中，但是实例发明的特征同样能很好地用于任何形式的通信系统中，包括例如卫星通信系统。本领域技术人员应当进一步理解，这里描述的实施例既可以 5 用于对数据通信编码也可以用于对话音通信编码。还应当理解，在以上整个描述中可以参考的数据、指令、命令、信息、信号、比特、符号和码片有利地由电压、电流、电磁波、磁场或粒子、光场或粒子或其任何组合来表示。

本领域技术人员应当进一步明白，结合这里揭示的实施例描述的各种说明性逻辑块和算法步骤可以用数字信号处理器 (DSP)、应用专用集成电路 10 (ASIC)、分立的门或晶体管逻辑、分立的硬件元件(如寄存器和 FIFO)、执行一组固件指令的处理器、或者任何传统可编程软件模块以及处理器来实施或进行。处理器可以有利地是微处理器，但是在另一方面，处理器可以是任何传统的处理器、控制器、微控制器、或者状态机。软件模块可以驻留在 RAM 存储器、快闪存储器、寄存器或本领域已知的任何其他形式的可读存储媒体中。

15 因此已经示出和描述了本发明的较佳实施例。然而应当明白，对于本领域一名技术人员而言，可以对这里揭示的这些实施例作出众多的改变，而不偏离本发明的精神和范围。因此，本发明并不局限于这些，除非按照以下的权利要求书。

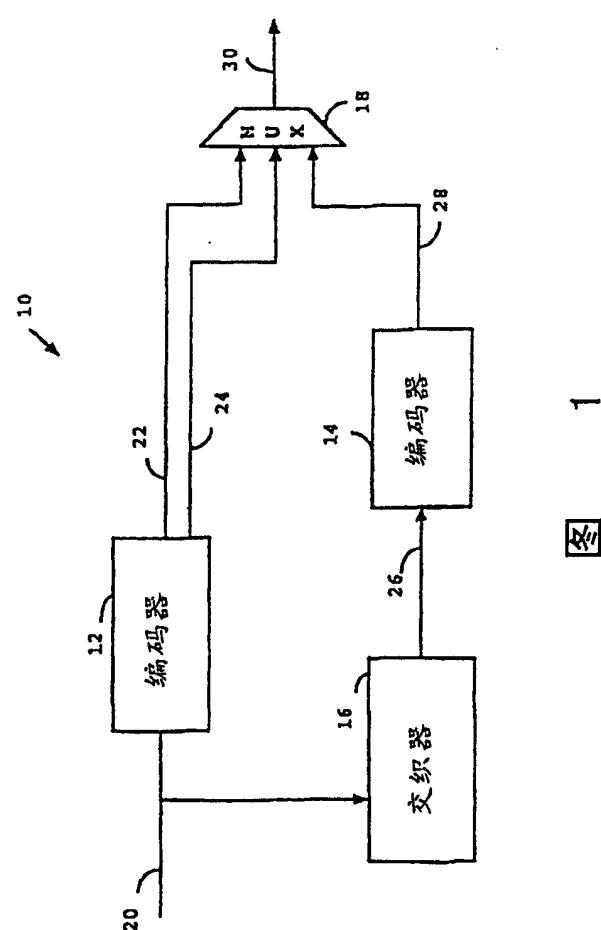
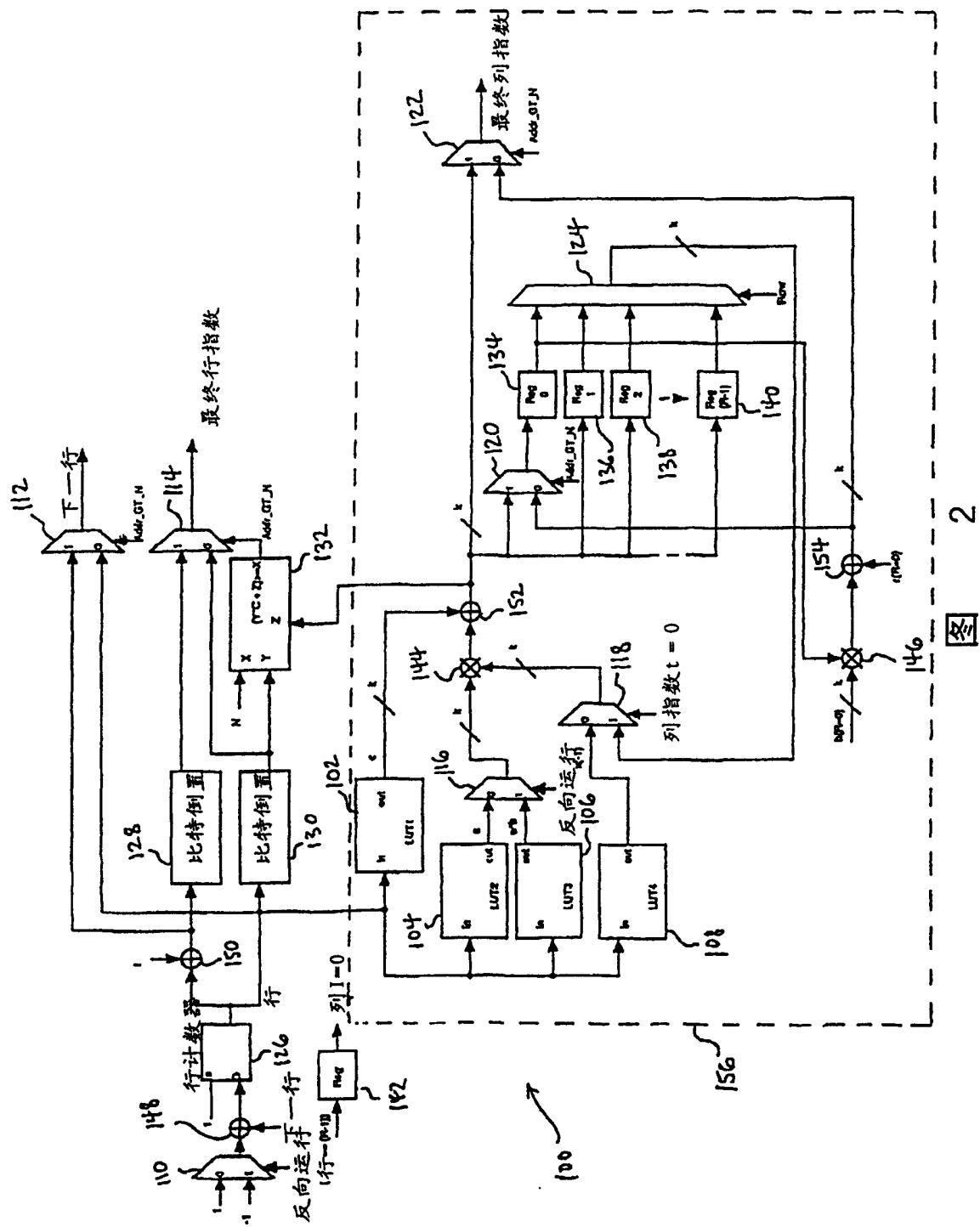


图 1



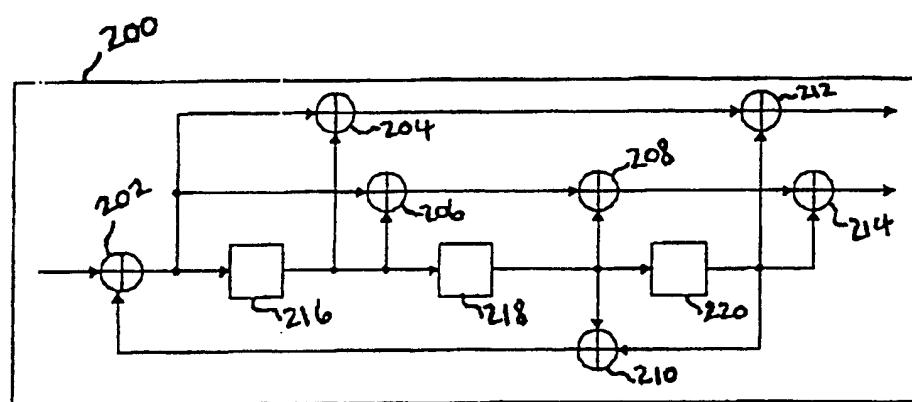


图 3