

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H03M 13/27
H04L 1/00

[12] 发明专利申请公开说明书

[21] 申请号 00137726.4

[43] 公开日 2001 年 8 月 1 日

[11] 公开号 CN 1306348A

[22] 申请日 2000. 12. 27 [21] 申请号 00137726.4
[30] 优先权
[32] 1999. 12. 28 [33] US [31] 09/473, 155
[71] 申请人 德克萨斯仪器股份有限公司
地址 美国得克萨斯州
[72] 发明人 A·雅基尔

[74] 专利代理机构 上海专利商标事务所
代理人 钱慰民

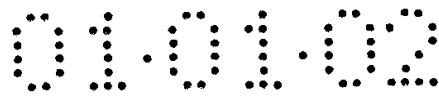
权利要求书 1 页 说明书 4 页 附图页数 0 页

[54] 发明名称 对长度可变的数据分组进行有效交织的方法

[57] 摘要

提供了一种对长度可变的数据分组进行有效交织的方法。在数据分组网 络中,当交织器的块大小固定且小于数据分组大小时,对长度可变数据分组 的块交织效率很差。对于某些数据分组长度,最后一块的剩余单元数(即、字 节或码元)非常少。本发明提供了一种为预选数据分组长度和预选交织器宽度 计算最佳交织器块大小的方法。使块大小最大,同时使其大小几乎相等。

I S S N 1 0 0 8 - 4 2 7 4



权 利 要 求 书

1. 一种为具有已知长度的数据分组计算交织器最佳块大小的方法，其特征在于，包括以下步骤：

使块大小最大，同时使所述块的所述大小大致相等。

2. 如权利要求 1 所述的方法，其特征在于，还包括：
选择交织器宽度等于码字长度。

3. 一种用于数据分组的交织器，其特征在于，包括：
输入电路，用于为所述数据分组提供至少一个长度；
存储器，用于存储数据分组信息；
处理器，用于根据所述数据分组信息确定最大块大小；和
输出电路，用于提供交织块中的所述数据分组信息。

4. 如权利要求 3 所述的交织器，其特征在于，所述处理器确定所述块的所述最大尺寸，以便每个所述块的大小大致相等。

5. 一种具有已知长度的数据分组计算交织器最佳块大小的方法，其特征在于，包括以下步骤：

首先，提供以下数据：

提供用单元表示的总数据分组长度(例如，位、字节、码元等)，用 N_f 表示；

提供交织器宽度，用 N_r 表示；

提供用单元表示的最大块大小，用 B_r 表示；

提供所述数据分组中交织器的总行数，用 $I_{tot}^0 = \text{ceil}(N_f/N_r)$ 表示；

其中“ $\text{ceil}(x)$ ”是一函数，其值为沿正无限大方向最接近 x 的整数；

提供每段的最大行数，使得交织器块不超过 B_r ，用 $I_{r,max} = \text{floor}(B_r/N_r)$ 表示，其中“ $\text{floor}(x)$ ”是一函数，其值为沿负无限大方向最接近 x 的整数；

提供一个分段数目，用 $N_s^0 = \text{ceil}(I_{tot}/I_{r,max})$ 表示；以及

其次，用以下方式处理所述数据：

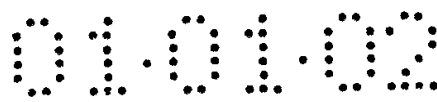
对于一个分段 m ，按以下公式确定最大行数 I_r^m ：

$$m=1, \dots, N_s^0:$$

$$I_r^m = \text{ceil}(I_{tot}^{(m-1)}/N_s^{(m-1)})$$

$$N_s^m = N_s^{(m-1)} - 1$$

$$I_{tot}^m = I_{tot}^{(m-1)} - I_r^m$$



说明书

对长度可变的数据分组进行有效交织的方法

本发明涉及数字通信信道，尤其涉及一种对长度可变的数据分组进行有效交织的方法。

在许多通信信道中，差错以突发方式出现，即单个差错事件具有不止一个差错数据单元(位、字节、码元或其它单元)。突发差错因各种原因而产生，原因有信道中的突发噪声，维特比解码器输出中的突发差错，以及其它形式。因为大多数解码器不能有效地处理许多相邻的差错，所以通常不希望有突发差错。交织器是一种通过对数据单元的置换将来自一个脉冲串的差错分散成分立的多个“单”差错的器件。例如，当使用分组编码(诸如，Reed-solomon)时，交织器可以在几个码字之间分散突发差错，从而放宽对该码的纠错要求。

一种非常普通而简单的交织器是块交织器。在此交织器中，将数据单元(位、字节、码元或其它单元)放在二维表的行方向，然后按表的列方向读取。块的大小(行数乘以列数)一般受交织器存储资源的限制。

在数据分组网络中，通过通信信道传送长度可变的数据分组。数据分组的大小通常大于块的大小。在此情况下，将数据分组成许多段，段的大小小于或等于交织器块的大小。然后对每段分别交织。一种寻常的分段方式是使最后一段以外的所有分段都等于交织器块的大小。于是，最后一段将小于块的大小。此方法的缺点是，对于某些数据分组长度来说，最后一段会非常小。这将导致交织器的性能很差。

本发明提供了一种用于对长度可变的数据分组进行有效交织的方法。在数据分组网络中，当交织器的块大小固定且小于数据分组大小时，对长度可变数据分组的块交织效率很差。原因是，对于某些数据分组长度，最后一块的剩余单元数(即、字节或码元)非常少。

本发明的方法可以为已知的数据分组长度(可变)和已知的(或选定的)交织器宽度计算最佳的交织器块大小。

交织器是一种通过对数据分组中的数据单元进行排列将来自一个脉冲串的差错分散成分立的多个“单”差错的器件。例如，当使用分组编码(诸如，Reed-solomon)时，交织器可以在几个码字之间分散突发差错，从而放宽对该

码的纠错要求。

一种非常普通而简单的交织器是块交织器。在此交织器中，将数据单元（位、字节、码元或其它单元）放在二维表的行方向，然后按表的列方向读取。块的大小（行数乘以列数）一般受交织器存储资源的限制。

在数据分组网络中，通过通信信道传送长度可变的数据分组。数据分组的大小通常大于块的大小。在此情况下，将数据分组分成许多段，段的大小小于或等于交织器块的大小。然后对每段分别交织。一种寻常的分段方式是使最后一段以外的所有分段都等于交织器块的大小。于是，最后一段将小于块的大小。此方法的缺点是，对于某些数据分组长度来说，最后一段会非常小。这将导致交织器的性能很差。

计算块大小更有效于仅仅挑选块大小并让尾部（最后一块）为任意获得的大小，其理由如下：一个块的脉冲串容限（即可以攻击块的最长噪声脉冲串，由此最长噪声脉冲串仍可以恢复数据）是 $T \cdot I$ 字节，其中 T 是 Reed-Solomon (RS) 校正因数，而 I 是块的交织深度。一个数据分组的脉冲串容限由其“最弱”的块，即交织深度最小的块决定。因此，希望最后一块的交织深度不太小（可能是 1）。这可以用本发明的方法来实现。

本发明计算数据分组的各段大小，使得所有分段的大小都几乎相等，并仍然尽可能的大。列的数目是已知的或者固定的，并且最好等于码字长度。使用公式计算每段的行数，它是数据分组长度、码字长度和最大块大小的函数。

本发明提供了一种对给定数据分组长度计算最佳交织器块大小并使交织器块大小最大的简单计算方法。也就是说，本发明是将数据分散到大致相等的交织器块中，而不是具有几个大小固定的块以及一个位于尾部的小块，且小块包含少量的数据（例如，较小的交织深度），因此本发明可以使脉冲串噪声容限最大。

在本发明的公式中使用了以下符号：

用单元表示的总数据分组长度（例如，位、字节、码元等）： N_f

交织器宽度（一般是码字长度）： N_r

最大块大小（单元数）： B_r

数据分组中交织器的总行数： $I_{tot}^0 = \text{ceil}(N_f/N_r)$ ，

其中“ $\text{ceil}(x)$ ”是一函数，其值为沿正无限大方向最接近 x 的整数。

每段的最大行数(使得交织器块不超过 B_r): $I_{r,\max} = \text{floor}(B_r/N_r)$,

其中“ $\text{floor}(x)$ ”是一函数,其值为沿负无限大方向最接近 x 的整数。

分段数: $N_s^0 = \text{ceil}(I_{\text{tot}}/I_{r,\max})$ 。

于是,对于一个分段 m ,用于计算最大行数 I_r^m 的公式是:

$$m=1, \dots, N_s^0:$$

$$I_r^m = \text{ceil}(I_{\text{tot}}^{(m-1)}/N_s^{(m-1)})$$

$$N_s^m = N_s^{(m-1)} - 1$$

$$I_{\text{tot}}^m = I_{\text{tot}}^{(m-1)} - I_r^m$$

使用此公式的一个例子是:

$$N_f=2100, N_r=200, B_r=2048,$$

$$I_{\text{tot}}^0 = \text{ceil}(N_f/N_r) = 11$$

$$I_{r,\max} = \text{floor}(B_r/N_r) = 10$$

$$N_s^0 = \text{ceil}(I_{\text{tot}}^0/I_{r,\max}) = 2$$

$$I_r^1 = \text{ceil}(I_{\text{tot}}^0/N_s^0) = 6$$

$$N_s^1 = N_s^0 - 1 = 1$$

$$I_{\text{tot}}^1 = I_{\text{tot}}^0 - I_r^1 = 5$$

$$I_r^2 = \text{ceil}(I_{\text{tot}}^1/N_s^1) = 5$$

$$N_s^2 = N_s^1 - 1 = 0$$

$$I_{\text{tot}}^2 = I_{\text{tot}}^1 - I_r^2 = 0$$

在此例中,已获得了最佳分段。也就是说,对于此例,计算得到均具有 5 行的两个块(交织深度为 5)。尽管选择了最大块大小 B_r ,但这些公式还可以用来优化块大小,或者可以使用几种块大小并选择一个具有最佳(最大)交织深度进行使用。

以下是另一个公式。此公式提供相等的段大小,但次序不同。

同样使用先前的符号:其中

数据分组长度

N_f

码字长度

N_r

最大块大小

B_r

交织器总行数

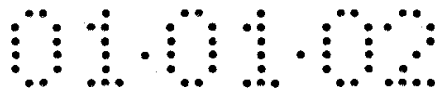
$I_{\text{tot}}^0 = \text{ceil}(N_f/N_r)$

每段的最大行数

$I_{r,\max} = \text{floor}(B_r/N_r)$

分段数

$N_s^0 = \text{ceil}(I_{\text{tot}}^0/I_{r,\max})$



第一块的交织深度

$$I_r^1 = \text{ceil}(I_{\text{tot}}^0 / N_s^0)$$

深度为 I_r^1 的块的数目

$$M = N_s^0 \cdot (I_r^1 + 1) - I_{\text{tot}}^0$$

于是, 对于分段 i , 用于计算最大行数 $I_r^{(i)}$ 的公式为:

$$(i=1, \dots, N_s^0):$$

$$I_r^{(i)} = \begin{cases} I_r^1, & i=1, \dots, M \\ I_r^1 + 1, & i=M+1, \dots, N_s^0 \end{cases}$$

数据分组由 N_s^0 个交织段(块)组成, 每段的宽度(列数)为 N_r 。最前面的 M 个分段的深度(行数)为 I_r^1 。剩余分段的深度为 $I_r^1 + 1$ 。

使用此公式的例子如下:

$$N_f=2100, N_r=200, B_r=2048,$$

$$I_{\text{tot}}^0 = \text{ceil}(N_f / N_r) = 11$$

$$I_{r, \max} = \text{floor}(B_r / N_r) = 10$$

$$N_s^0 = \text{ceil}(I_{\text{tot}}^0 / I_{r, \max}) = 2$$

$$I_r^1 = \text{floor}(I_{\text{tot}}^0 / N_s^0) = 5$$

$$M = N_s^0 - (I_{\text{tot}}^0 - N_s^0 \cdot I_r^1) = 1$$

此例的结果是:

$$I_r^1 = 5, I_r^2 = 6$$

也就是说, 在此例中, 计算得到两个块, 一块具有 5 行, 另一块具有 6 行。同样, 可以对块大小进行优化。用类似的方式可以产生其它公式, 用于计算一给定数据分组长度的交织器块大小, 同时使块大小最大。

本发明可以用软件、硬件或软件与硬件的组合来实现。对于本领域的熟练技术人员来说, 阅读了上述对本发明的几个特定实施例之后, 很容易进行各种改变、修改和改进。这种根据本申请揭示内容为显而易见的改变、修改和改进是本说明书的一部分, 尽管没有在此表述, 并且它们落在本发明的精神和范围内。因此, 上述描述只是举例, 没有限制作用。本发明只被后附权利要求及其等效物的限制。