

说明书

一种并行加扰的方法及装置

技术领域

本发明及对通信信号进行加扰的方法及装置，特别是涉及采用稀疏存储
5 技术对通信信号进行并行加扰的方法及装置。

背景技术

在通信系统中，加扰的目的主要在于将干扰信号随机化，在发送端用小
区专用扰码序列进行加扰，接收端再进行解扰，只有本小区内的 UE 才能根
据本小区的 ID 形成的小区专用扰码序列对接收到的本小区内的信息进行解
10 扰，这样可以在一定程度上减小相邻小区间的干扰。

现有的通信加扰技术可分为两种：串行加扰技术和并行加扰技术。

串行加扰就是用一个标准的伪随机序列与接收到信号串行加扰，伪随机
序列由一个标准的伪随机序列发生器生成，其中“0”与“1”出现的概率接
近 50%。由于二进制数值运算的特殊性质，用伪随机序列对输入的传送码流
15 进行扰乱后，无论原始传送码流是何种分布，扰乱后的数据码流中“0”与“1”
的出现概率都接近 50%。扰乱虽然改变了原始传送码流，但这种扰乱是有规
律的，因而也是可以解除的。串行加扰示意图如附图 1 所示。

但串行加扰存在的问题是由于受触发器翻转性能以及后续存储器读写操
作速度的限制。在这个速率等级上，对高速的串行数据流进行线速处理在硬
20 件实现上是不现实的。

传统的并行加扰技术是指扰码器同时给出多位伪随机码，与多路并行信
码进行逻辑运算，产生多位加扰的信码输出。对给定的信码流，进行并行加

扰后经并串转换所产生的码字,应当与直接串行加扰所产生的码字完全一致。实现并行加扰主要是伪随机序列发生器的并行化问题。即如何设计一种扰码器,使每个时钟周期所产生的并行码字符合规定的伪随机码顺序。因此,并行加扰技术就将串行的信码分块,不同的分块同时多个过程中进行加扰。

5 并行加扰示意图如附图 2 所示。

传统的并行加扰方法在并行宽度不大的情况下,复杂度与扰码生成多项式的阶数无关,而且不需要占用硬件的存储资源。

但是传统并行加扰方法也存在问题:

1、如果使用的扰码器复杂,传统并行加扰方法产生的伪随机序列长,对应寄存器的状态变化关系矩阵阶数高,即使使用 VHDL 语言在可编程逻辑器件上实现,也会造成庞大的运算量。

2、传统并行加扰方法的系统开销仍然很大,而且随着并行宽度的增大,扰码器并行输出逻辑的推导开始变复杂,需要占用的硬件资源开始变大。

发明内容

15 本发明的目的在于提供一种并行加扰的方法,以减少加扰的运算量和对存储空间的占用。

本发明的目的是这样实现的:一种并行加扰的方法,按以下步骤进行:

a) 信号分组

以扰码器状态变化关系矩阵的阶数 n 作为分割长度,对输入信号分割后
20 进行分组储存;

b) 加扰序列生成

根据加扰所需伪随机序列长度,确定扰码器的状态变化关系矩阵 T ;采

用以下三个数组对状态变化矩阵 T 进行稀疏存储：一维实型数组 A ：存储矩阵 T 的非零元素元值；一维实型数组 JA ：存储矩阵 T 的非零元素的列下标作为列值；一维实型数组 IA ：存储矩阵 T 的非零元素的行下标作为行标；设扰码器在初始时刻的状态 $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$ ， $Q^{n+k} = T^k \otimes Q^n$ ，式中， Q^{n+k} 为扰码器在 $n+k$ 时刻的状态， T^k 为状态变化矩阵 T 的 k 次方， $\otimes$ 符号表示矩阵与向量相乘过程中的加法采用模 2 加法；将 $Q_6^{n+1} Q_6^{n+2} \cdots Q_6^{n+k}$ 作为加扰序列；

c) 加扰处理

扰码器的并行输出数据 D_{out} 与并行输入数据 D_{in} 及扰码器初始状态之间的逻辑关系如下式： $D_{out}^{n+k} = D_{in}^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \cdots Q_6^{n+k})$ ，其中 $\oplus$ 表示模二加法，并对不同分组的输入信号采用加扰序列并行进行加扰处理；

d) 并串转换

在加扰后，不同分组的输入信号根据输入信号的时刻，按次序还原成串行的数据流。

上述阶数 n 等于 7 时，步骤 c) 中扰码器并行输出数据 D_{out} 与并行输入数据 D_{in} 及扰码器初始状态之间的逻辑关系式下：

$$D_{out}(n+i) = \begin{cases} D_{in}(n+i) \oplus Q_{6-i}^n & (0 \leq i \leq 6) \\ D_{in}(n+i) \oplus Q_{13-i}^n \oplus Q_{12-i}^n & (7 \leq i \leq 11) \\ D_{in}(n+12) \oplus Q_1^n \oplus Q_0^n & (i=12) \\ D_{in}(n+13) \oplus Q_0^n \oplus Q_5^n \oplus Q_6^n & (i=13) \\ D_{in}(n+14) \oplus Q_4^n \oplus Q_6^n & (i=14) \\ D_{in}(n+15) \oplus Q_3^n \oplus Q_5^n & (i=15) \end{cases}$$

其中， $\oplus$ 符号表示模二加法。

本发明的另一目的是提供一种能减少加扰的运算量和降低对存储空间占

用的并行加扰的装置。

本发明另一目的是这样实现的：一种并行加扰的装置，包括：

信号分组单元：以扰码器状态变化关系矩阵的阶数 n 作为分割长度，对输入信号分割后进行分组储存；

- 5 加扰序列生成单元：根据加扰所需伪随机序列长度，确定扰码器的状态变化关系矩阵，采用以下三个数组对状态变化矩阵 T 进行稀疏存储：一维实型数组 A ：存储矩阵 T 的非零之素元值；一维实型数组 JA ：存储矩阵 T 的非零元素的列下标作为列值，设扰码器在初始时刻的状态 $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$ ， $Q^{n+k} = T^k \otimes Q^n$ ，式中， Q^{n+k} 为扰码器在 $n+k$ 时刻的状态， T^k 为状态变化矩阵 T 的 k 次方， $\otimes$ 符号表示矩阵与向量相乘过程中的加法采用模 2 加法；将 $Q_6^{n+1} Q_6^{n+2} \cdots Q_6^{n+k}$ 作为加扰序列；
- 10 加扰单元：扰码器的并行输出数据与并行输入数据及扰码器初始状态之间的逻辑关系， $D_{out}^{n+k} = D_{in}^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \cdots Q_6^{n+k})$ ，将不同分组的输入信号并行进行加扰处理；

- 15 并串转换单元：在加扰后，不同分组的输入信号根据输入信号的时刻，按次序还原成串行的数据流。

本发明的有益效果是：本发明利用稀疏矩阵的特点，在伪随机序列较长、寄存器状态变化关系矩阵阶数较高时，能够减少加扰的运算量和对存储空间

的占用。

- 20 本发明的有益效果将结合具体实施方式加以进一步阐述。

附图说明

图 1 是传统串行加扰示意图；

图 2 是传统并行加扰示意图；

图 3 是扰码器示意图；

图 4 是稀疏矩阵储存示意图；

图 5 是本发明实施例提供的方法的程序流程图；

5 图 6 是本发明实施例提供的装置的示意图。

具体实施方式

下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员所获得的所有其他实施例，都属于本发明保护的范围。

在通信系统的加扰中，扰码器的状态不依赖于输入数据，这就为加扰与解扰的并行处理提供了可能。例如，扰码器示意图如附图 3 所示。扰码器中各寄存器的状态变化关系可用矩阵表示为：

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (1)$$

15 设扰码器在任意时刻 n 的状态为：

$$Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n] \quad (2)$$

则在 $n+1$ 时刻的状态为：

$$Q^{n+1} = T \otimes Q^n \quad (3)$$

其中 T 为变换矩阵， $\otimes$ 符号表示矩阵与向量相乘过程中的加法采用模 2

加法。

同样可以得出在 $n+k$ 时刻扰码器的状态为：

$$Q^{n+k} = T^k \otimes Q^n \quad (4)$$

其中 T 的上标为 k 次方，其他字母上下标无运算含义。

- 5 由以上分析可知， $n+k$ 时刻输入加扰器的信号 D_{in}^{n+k} ，通过加扰器后输出信号为：

$$D_{out}^{n+k} = D_{in}^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k}) \quad (5)$$

- 可以在同一时刻利用多个加扰器，并行对多个时刻信号进行加扰处理，从而解决于加扰过程中，由于硬件速度的限制所导致的整个系统的传输速度
10 瓶颈。即将一个速度为 M 的过程，分散到 N 个只能有 M/N 速度的过程，保证整体的速度 M 不降低。同时该电路可以方便的用 VHDL 语言在可编程逻辑器件上进行实现。使用该方法可以推导出任意位宽的并行加扰电路。

- 稀疏矩阵存储采用稀疏数据结构方式进行，即只存储矩阵中的非零元素，并且依这种结构能方便在进行通常的矩阵运算。稀疏矩阵最简单的存储格式
15 是用三个数组来存储稀疏矩阵，简单而灵活：

一维实型数组 **A**：存储矩阵 **A** 的非零元素之值；

一维实型数组 **JA**：存储非零元素的列下标；

一维实型数组 **IA**：存储非零元素的行下标；

例如：当

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (6)$$

按照矩阵的稀疏存储，A 可存储为：

A	1	1	1	1	1	1	1	1
JA	1	2	3	4	5	6	6	7
IA	2	3	4	5	6	1	7	1

表 1 稀疏矩阵存储参数表

稀疏矩阵储存示意图如附图 4 所示。

5 在使用本发明实施例提供的方法时，参见图 5，可以包括以下步骤：

S101：以扰码器状态变化关系矩阵的阶数 n 作为分割长度，对输入信号进行分割储存；

S102：首先根据加扰所需伪随机序列长度，确定扰码器的状态变化关系矩阵 T ：

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

按照矩阵的稀疏存储， T 可存储为：

T	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

JT	1	2	3	4	5	6	6	7
IT	2	3	4	5	6	1	7	1

表 2 寄存器状态变化关系稀疏矩阵

设定扰码器在初始时刻的状态，可得： $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$ ，
 $Q^{n+k} = T^k \otimes Q^n$ 。

将稀疏储存的 T 矩阵中的行值与列值对应相等的元素相乘。

5 例如 T^2 为：

A	1	1	1	1	1	1	1	1
JA	1	2	3	4	5	5	6	7
IA	3	4	5	6	7	1	1	2

表 3 寄存器状态变化关系稀疏矩阵的 2 次方

$Q^n = [1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1]^T$ 时，可储存为：

D	1	1	1	1	1
JD	1	1	1	1	1
ID	1	2	4	6	7

表 4 状态稀疏矩阵

由 $Q^{n+k} = T^k \otimes Q^n$ 可知，当 $k=2$ ，则可以转化为： $Q^{n+2} = T^2 \otimes Q^n$ ，则 Q^{n+2} 为：

D	1	1	1	1	1
JD	1	1	1	1	1
ID	1	2	3	6	4

表 5 更新状态系数矩阵

所以， $Q^{n+1} = [1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0]^T$ ；

S103: 根据 $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$, $Q^{n+k} = T^k \otimes Q^n$, 可以推导出扰码器的并行输出与并行输入数据及扰码器初始状态之间的逻辑关系:

$$D_{out}(n+i) = \begin{cases} D_{in}(n+i) \oplus Q_{6-i}^n & (0 \leq i \leq 6) \\ D_{in}(n+i) \oplus Q_{13-i}^n \oplus Q_{12-i}^n & (7 \leq i \leq 11) \\ D_{in}(n+12) \oplus Q_1^n \oplus Q_0^n & (i=12) \\ D_{in}(n+13) \oplus Q_0^n \oplus Q_5^n \oplus Q_6^n & (i=13) \\ D_{in}(n+14) \oplus Q_4^n \oplus Q_6^n & (i=14) \\ D_{in}(n+15) \oplus Q_3^n \oplus Q_5^n & (i=15) \end{cases}$$

将不同的分组的输入信号并行进行加扰处理;

- 5 S104: 将不同分组的输入信号根据输入信号的时刻, 按次序还原成串行的数据流。

本发明的创新点在于: 采用了稀疏矩阵储存和稀疏矩阵相乘的办法处理并行加扰过程中的运算。可以应用于大多数需要加扰的通信系统中。

本发明的优势在于: 大幅减少可编程逻辑器件的运算量。以矩阵 T 为例:

- 10 如果使用普通的矩阵相乘的办法, $D_{out} = D_{in} \otimes A$ 的过程中要进行 343 次两个实数相乘的运算。当使用稀疏矩阵相乘时, 仅需要进行 7 次实数相乘的运算。且从编程逻辑器件的结构考虑, 实数相乘是耗费编程逻辑器件运算时间较多的运算, 所以就上例而言, 运算效率提高了 97.9%。当伪随机序列较长, 对应的寄存器的状态变化关系矩阵阶数高时, 使用稀疏矩阵相乘的对运算效率的提升会更高。
- 15

与本发明实施例提供的并行加扰的方法相对应, 本发明实施例还提供了一种并行加扰的装置, 参见图 6, 该装置包括:

信号分组单元 701, 以扰码器状态变化关系矩阵的阶数 n 作为分割长度, 对输入信号进行分割储存;

加扰序列生成单元 702，根据加扰所需伪随机序列长度，确定扰码器的状态变化关系矩阵将稀疏储存 T 矩阵中行值与列值对应相等的元素相乘；

加扰单元 703，设定扰码器在初始时刻的状态，根据式：

$$Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n], \quad Q^{n+k} = T^k \otimes Q^n,$$

5 数据及扰码器初始状态之间的逻辑关系，将不同分组的输入信号并行进行加扰处理；

并串转换单元 704，在加扰后，不同分组的输入信号根据输入信号的时刻，按次序还原成串行的数据流。

需要说明的是，本发明实施例所描述的装置实施例是与前述方法实施例
10 对应的，因此，装置实施例中未详述部分，可以参照方法实施例中的介绍，这里不再赘述。

以上对本发明所提供的一种并行加扰的方法及装置，进行了详细介绍，本文中应用了具体个例对本发明的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本发明的方法及其核心思想；同时，对于本领域的一般技术人员，依据本发明的思想，在具体实施方式及应用范围上均会有改变之处。综上所述，本说明书内容不应理解为对本发明的限制。
15

权 利 要 求

1、一种并行加扰的方法，其特征在于，按以下步骤进行：

a) 信号分组：以扰码器状态变化关系矩阵的阶数 n 作为分割长度，对输入信号分割后进行分组储存；

5 b) 加扰序列生成：根据加扰所需伪随机序列长度，确定扰码器的状态变化关系矩阵 T ；采用以下三个数组对状态变化矩阵 T 进行稀疏存储：一维实型数组 A ：存储矩阵 T 的非零元素元值；一维实型数组 JA ：存储矩阵 T 的非零元素的列下标作为列值；一维实型数组 IA ：存储矩阵 T 的非零元素的行下标作为行标；设定扰码器在初始时刻的状态 Q^n ，根据式：
10 $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$ ， $Q^{n+k} = T^k \otimes Q^n$ ，式中， Q^{n+k} 为扰码器在 $n+k$ 时刻的状态， T^k 为状态变化矩阵 T 的 k 次方， $\otimes$ 符号表示矩阵与向量相乘过程中的加法采用模 2 加法，将 $Q_6^{n+1} Q_6^{n+2} \cdots Q_6^{n+k}$ 作为加扰序列；

 c) 加扰处理：扰码器的并行输出数据 D_{out} 与并行输入数据 D_{in} 及扰码器初始状态之间的逻辑关系如下式： $D_{out}^{n+k} = D_{in}^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \cdots Q_6^{n+k})$ ，其中 $\oplus$ 表示模二
15 加法，并对不同分组的输入信号采用加扰序列并行进行加扰处理；

 d) 并串转换：在加扰后，不同分组的输入信号根据输入信号的时刻，按次序还原成串行的数据流。

2、根据权利要求 1 所述的一种并行加扰的方法，其特征是，所述阶数 n 等于 7 时步骤 c) 中，扰码器并行输出数据 D_{out} 与并行输入数据 D_{in} 及扰码器
20 初始状态之间的逻辑关系式下：

$$D_{out}(n+i) = \begin{cases} D_{in}(n+i) \oplus Q_{6-i}^n & (0 \leq i \leq 6) \\ D_{in}(n+i) \oplus Q_{13-i}^n \oplus Q_{12-i}^n & (7 \leq i \leq 11) \\ D_{in}(n+12) \oplus Q_1^n \oplus Q_0^n & (i=12) \\ D_{in}(n+13) \oplus Q_0^n \oplus Q_5^n \oplus Q_6^n & (i=13) \\ D_{in}(n+14) \oplus Q_4^n \oplus Q_6^n & (i=14) \\ D_{in}(n+15) \oplus Q_3^n \oplus Q_5^n & (i=15) \end{cases}$$

其中， $\oplus$ 符号表示模二加法。

3、一种并行加扰的装置，其特征在于，包括：

信号分组单元：以扰码器状态变化关系矩阵的阶数 n 作为分割长度，对

5 输入信号分割后进行分组储存；

加扰序列生成单元：根据加扰所需伪随机序列长度，确定扰码器的状态变化关系矩阵，采用以下三个数组对状态变化矩阵 T 进行稀疏存储：一维实型数组 A ：存储矩阵 T 的非零之素元值；一维实型数组 JA ：存储矩阵 T 的非零元素的列下标作为列值，设扰码器在初始时刻的状态

10 $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$ ， $Q^{n+k} = T^k \otimes Q^n$ ，式中， Q^{n+k} 为扰码器在 $n+k$ 时刻的状态， T^k 为状态变化矩阵 T 的 k 次方， $\otimes$ 符号表示矩阵与向量相乘过程中的加法采用模2加法；将 $Q_6^{n+1} Q_6^{n+2} \cdots Q_6^{n+k}$ 作为加扰序列；

加扰单元：扰码器的并行输出数据与并行输入数据及扰码器初始状态之间的逻辑关系， $D_{out}^{n+k} = D_{in}^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \cdots Q_6^{n+k})$ ，将不同分组的输入信号并行进行

15 加扰处理；

并串转换单元：在加扰后，不同分组的输入信号根据输入信号的时刻，按次序还原成串行的数据流。