

Method and Device for Parallel Scrambling

Technical Field

The present invention relates to a method and apparatus for scrambling communication signal, specifically to a method and apparatus for parallel scrambling communication signal by sparse storage technique.

Background Art

In the communication system, the main purpose of scrambling is to randomize interference signal through scrambling by the community-specific scrambling sequence at the transmission terminal and descrambling at the receiving terminal so that only the UE within the community can descramble the information received in the community according to the community-specific scrambling sequence formed based on the community ID, thereby reducing inter-community interference to some extent.

Existing communication scrambling technology can be divided into two types: serial scrambling technology and parallel scrambling technology.

Serial scrambling is realized by a standard pseudo-random sequence and the received signal. The pseudo-random sequence is generated by a standard pseudo-random sequence generator, wherein the probability of appearance of "0" and "1" is close to 50%. Due to the special nature of the operation of the binary values, after the input transmission code is disrupted by pseudo-random sequence, regardless of the distribution of the original transmission code stream, the probability of appearance of "0" and "1" is close to 50%. Scrambling changes the original transmission code stream but it is regular and therefore can be descrambled. Serial scrambling is illustrated in Figure 1.

However, serial scrambling is limited by the turnover performance of the trigger and the reading and writing speed of subsequent memory. Thus, at this speed grade, it is unrealistic to perform line speed processing on high-speed serial data stream in terms of hardware.

Traditional parallel scrambling technology refers to that the scrambler gives a number of pseudo-random codes for parallel logic operation of multiple channel codes so as to generate multi-scrambling code output. The code generated by a given code

after scrambling and parallel and serial conversion should be fully consistent with that generated from direct serial scrambling. Parallel scrambling mainly refers to parallel operation of pseudo-random sequence generator. Namely, how to design a scrambler, which can enable the parallel code generated from each clock cycle to comply with the regulated pseudo-random code sequence. Therefore, the parallel scrambling technology divides serial codes into different blocks, which are scrambled simultaneously in a plurality of processes. The parallel scrambling is as shown in Figure 2.

As for traditional parallel scrambling method, in the case of small parallel width, complexity is unrelated to the exponent number of the polynomial generated from the scrambling code and does not take up hardware storage resources.

However, traditional parallel scrambling method also has some problems:

1. If the used scrambler is complex, the pseudo-random sequence generated from traditional parallel scrambling method is long and the exponent number of corresponding register state change relationship matrix is high, even if VHDL language is used for realization on programmable logic devices, it will also cause enormous computation.

2. The traditional parallel scrambling method results in large system expenditure. With increase of the parallel width, derivation of parallel output logics of the scrambler becomes complicated and needs more and more hardware resources.

Summary of the Invention

The present invention aims to provide a parallel scrambling method to reduce the amount of scrambling calculation and occupancy of storage space.

The object of the present invention is achieved in a parallel scrambling method, including the following steps:

- a) Signal grouping

Dividing the input signal for grouped storage with the exponent number n of scrambler change relationship matrix as the division length;

- b) Generating scrambling sequence

Determining scrambler change relationship matrix T according to the pseudo-random sequence length needed for scrambling; performing sparse storage of the state change matrix T by the following three arrays: one-dimensional real array

A for storing non-zero element values of the matrix T ; one-dimensional real array JA for storing the column index of non-zero elements of the matrix T as the column value; and one-dimensional real array IA for storing the row index of non-zero elements of the matrix T as the row index; setting the state of the scrambler at the initial moment according to the formula: $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$ $Q^{n+k} = T^k \otimes Q^n$, wherein Q^{n+k} represents the state of the scrambler at the time of $n+k$, T^k represents the k^{th} power of state change matrix T , $\otimes$ means that the addition during the process of multiplying the matrix by the vector is modulo-2 addition, and $Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k}$ is scrambling sequence;

c) Scrambling process

The relationship among the parallel output data D_{out} of the scrambler, parallel input data D_{in} of the scrambler and the initial state of the scrambler is shown in the following formula: $D_{out}^{n+k} = D_{in}^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k})$, wherein $\oplus$ represents modulo-2 addition, and different groups of input signal is subject to parallel scrambling process by scrambling sequence;

d) Parallel and serial conversion

After scrambling, different groups of input signal is recovered to serial data stream based on the time of the input signal.

When the exponent number n is equal to 7, the logic relationship among the parallel output data D_{out} of the scrambler, parallel input data D_{in} of the scrambler and the initial state of the scrambler is shown in the following formula:

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

Wherein, $\oplus$ represents modulo-2 addition.

The present invention aims to provide a parallel scrambling apparatus which can reduce the amount of scrambling calculation and occupancy of storage space.

The other object of the present invention is achieved by a parallel scrambling apparatus, comprising:

A signal grouping unit: dividing the input signal for grouped storage with the exponent number n of scrambler change relationship matrix as the division length;

A scrambling sequence generating unit: Determining scrambler change relationship matrix T according to the pseudo-random sequence length needed for scrambling; performing sparse storage of the state change matrix T by the following three arrays: one-dimensional real array A for storing non-zero element values of the matrix T ; and one-dimensional real array JA for storing the column index of non-zero elements of the matrix T as the column value; setting the state Q^n of the scrambler at the initial moment according to the formula: $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$, $Q^{n+k} = T^k \otimes Q^n$, wherein Q^{n+k} represents the state of the scrambler at the time of $n+k$, T^k represents the k^{th} power of state change matrix T , $\otimes$ means that the addition during the process of multiplying the matrix by the vector is modulo-2 addition, and $Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k}$ is scrambling sequence;

A scrambling unit: the logic relationship among the parallel output data of the scrambler, parallel input data of the scrambler and the initial state of the scrambler is shown in the following formula: $D_{out}^{n+k} = D_{in}^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k})$; wherein different groups of input signal is subject to parallel scrambling process;

A parallel-serial conversion unit: After scrambling, different groups of input signal is recovered to serial data stream based on the time of the input signal.

The beneficial effects of the present invention are as follows: the present invention makes use of the characteristics of sparse matrix to reduce the amount of scrambling calculation and occupancy of storage space in case of long pseudo-random

sequence and high exponent number of register state change relationship matrix.

The beneficial effects of the present invention will be further stated in connection with specific embodiments.

Brief Description of Drawings

Figure 1 is the schematic diagram of the traditional serial scrambling

Figure 2 is the schematic diagram of the traditional parallel scrambling

Figure 3 is the schematic diagram of the scrambler.

Figure 4 is the schematic diagram of sparse matrix storage.

Figure 5 is the flow chart of the method according to the embodiments of the present invention.

Figure 6 is the schematic diagram of the apparatus according to the embodiments of the present invention.

Embodiments

The embodiments of the present invention will be described clearly and completely in conjunction with the accompanying drawings of embodiments. It is clear that the described embodiments are merely part of embodiments of the present invention but not all embodiments. All other embodiments obtained by the ordinary technicians of this field based on the embodiments of the present invention shall be within the scope of protection of the present invention.

In scrambling of communication system, the state of the scrambler does not depend on the input data, which makes parallel processing of scrambling and descrambling possible. For example, the schematic diagram of the scrambler is as shown in Figure 3. The state change relationship of each register of scrambler can be expressed as the following matrix:

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

The state of the scrambler at any time is set to be:

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

The state at the moment of $n+1$ is:

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

Wherein, T is the conversion matrix, $\otimes$ means that the addition during the process of multiplying the matrix by the vector is modulo-2 addition.

The state of the scrambler at the moment of $n+k$ can be obtained:

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

Wherein, the upper index of T is the k^{th} power, and upper and lower index of other letters have no operational meaning.

From the above analysis, as for the signal D_m^{n+k} input into the scrambler at the moment of $n+k$, the output signal through the scrambler is:

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

A plurality of scramblers can be used at the same time for parallel scrambling processing of a plurality of timing signal in order to address the transmission speed bottleneck of the whole system due to limitation of hardware speed in scrambling process. Namely, the process at speed M is dispersed into N processes at M/N speed so as to ensure overall speed M does not decrease. Meanwhile, the circuit can easily apply VHDL language for programmable logic devices. The method can be used to derive parallel scrambling circuit of any bit width.

Sparse matrix storage is performed by sparse data structure. Namely, only the non-zero elements in the matrix can be stored and such structure make usual matrix calculation easy. The simplest storage format of sparse matrix is to use three arrays to store the sparse matrix, making it simple and flexible:

One-dimensional real array A for storing non-zero element values of the matrix A ;

One-dimensional real array JA for storing the column index of non-zero elements;

One-dimensional real array IA for storing the row index of non-zero elements;

For example, when:

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

In accordance with the sparse storage of the matrix, A can be stored as:

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

Table 1 Sparse Matrix Storage Parameters

The schematic diagram of sparse matrix storage is as shown in Figure 4.

When the method provided by the embodiments of the present invention is used, please refer to Figure 5. It may include the following steps:

S101: Dividing the input signal for grouped storage with the exponent number n of scrambler change relationship matrix as the division length;

S102: Determining scrambler change relationship matrix T according to the pseudo-random sequence length needed for scrambling;

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

In accordance with the sparse storage of the matrix, T can be stored as:

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

Table 2 Register state change relationship sparse matrix

Set the state of the scrambler at the initial moment, you can get:

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

The corresponding equal elements of the row and column values of sparse storage matrix T are multiplied.

For example, T^2 is:

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

Table 3 The 2nd power of register state change relationship sparse matrix

When $Q^n = [1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1]^T$, It can be stored as:

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

Table 4 State sparse matrix

According to $Q^{n+k} = T^k \otimes Q^n$, when $k=2$, it can be converted into:

$Q^{n+2} = T^2 \otimes Q^n$, and Q^{n+2} is:

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

Table 5 Update state coefficient matrix

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

S103: Deriving the logic relationship among the parallel output data of the scrambler, parallel input data of the scrambler and the initial state of the scrambler according to

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

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

Parallel scrambling processing is performed on different groups of input signals;

S104: After scrambling, different groups of input signal is recovered to serial data stream based on the time of the input signal.

The innovation of the present invention is as follows: parallel scrambling process is computed by the use of sparse matrix storage and sparse matrix multiplication approach. It can be applied in most communication systems needing scrambling.

The advantages of the present invention are as follows: it can substantially reduce computation of programmable logic devices. Take matrix T as an example: in case of ordinary matrix multiplication approach, in the $D_{out} = D_{in} \otimes A$ process, two real numbers should be multiplied for 343 times. In case of sparse matrix multiplication, it only requires seven times of multiplication of the real numbers. From the perspective of the structure of programmable logic devices, multiplication of real numbers takes much time for programming logic devices. Thus, for the above example, the operation efficiency is improved by about 97.9%. In case of long pseudo-random sequence and high exponent number of corresponding register change relationship matrix, sparse matrix multiplication will further improve the efficiency of operation.

Corresponding to the parallel scrambling method of the embodiments of the present invention, the present invention further provides a parallel scrambling apparatus as shown in Figure 6. It comprises:

A signal grouping unit 701: dividing the input signal for grouped storage with the exponent number n of scrambler change relationship matrix as the division length;

A scrambling sequence generating unit 702: Determining scrambler state change relationship matrix according to the pseudo-random sequence length needed for scrambling to multiple the corresponding equal elements of the row and column values of sparse storage matrix T ;

A scrambling unit 703: setting the initial state of the scrambler according to the formula: $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$, $Q^{n+k} = T^k \otimes Q^n$ so as to get the logic relationship among the parallel output data of the scrambler, parallel input data of the scrambler and the initial state of the scrambler; wherein different groups of input signal is subject to parallel scrambling process;

A parallel-serial conversion unit 704: After scrambling, different groups of input signal is recovered to serial data stream based on the time of the input signal.

It needs to be noted that the embodiments of the present invention are corresponding to the embodiments of the above method. Therefore, for those not described in details in embodiment of the apparatus, please refer to the embodiment of the method and it is omitted here.

The contents above describe in detail the parallel scrambling method and apparatus provided by the present invention. Specific examples are applied to state the principles and embodiments of the present invention. The description of the above embodiments is only to help the understanding of the method and core idea of the present invention. Meanwhile, according to the ideas of the present invention, the specific embodiments and applications may be different for ordinary technicians in this field. In summary, the specification shall not be understood as limitation to the present invention.

Claims

1. Method for parallel scrambling, wherein it includes the following steps:

a) Signal grouping

Dividing the input signal for a grouped storage with the exponent number n of scrambler change relationship matrix as the division length;

b) Generating scrambling sequence

Determining scrambler change relationship matrix T according to the pseudo-random sequence length needed for scrambling; performing sparse storage of the state change matrix T by the following three arrays: one-dimensional real array A for storing non-zero element values of the matrix T ; one-dimensional real array JA for storing the column index of non-zero elements of the matrix T as the column value; and one-dimensional real array IA for storing the row index of non-zero elements of the matrix T as the row index; setting the state Q^n of the scrambler at the initial moment according to the formula: $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$, $Q^{n+k} = T^k \otimes Q^n$, wherein Q^{n+k} represents the state of the scrambler at the time of $n+k$, T^k represents the k^{th} power of state change matrix T , $\otimes$ means that the addition during the process of multiplying the matrix by the vector is modulo-2 addition, and $Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k}$ is scrambling sequence;

c) Scrambling process

The relationship among the parallel output data D_{out} of the scrambler, parallel input data D_{in} of the scrambler and the initial state of the scrambler is shown in the following formula: $D_{out}^{n+k} = D_{in}^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k})$, wherein $\oplus$ represents modulo-2 addition, and different groups of input signal is subject to parallel scrambling process by scrambling sequence;

d) Parallel and serial conversion

After scrambling, different groups of input signal is recovered to serial data stream based on the time of the input signal.

2. The method for parallel scrambling according to claim 1, wherein when the exponent number n is equal to 7, in step c), the logic relationship among the parallel output data D_{out} of the scrambler, parallel input data D_{in} of the scrambler and the initial state of the scrambler is shown in the following formula:

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

Wherein, $\oplus$ represents modulo-2 addition.

3. Device for parallel scrambling, comprising:

A signal grouping unit: dividing the input signal for grouped storage with the exponent number n of scrambler change relationship matrix as the division length;

A scrambling sequence generating unit: Determining scrambler change relationship matrix T according to the pseudo-random sequence length needed for scrambling; performing sparse storage of the state change matrix T by the following three arrays: one-dimensional real array A for storing non-zero element values of the matrix T ; and one-dimensional real array JA for storing the column index of non-zero elements of the matrix T as the column value; setting the state Q^n of the scrambler at the initial moment according to the formula: $Q^n = [Q_0^n Q_1^n Q_2^n Q_3^n Q_4^n Q_5^n Q_6^n]$, $Q^{n+k} = T^k \otimes Q^n$, wherein Q^{n+k} represents the state of the scrambler at the time of $n+k$, T^k represents the k^{th} power of state change matrix T , $\otimes$ means that the addition during the process of multiplying the matrix by the vector is modulo-2 addition, and $Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k}$ is scrambling sequence;

A scrambling unit: the logic relationship among the parallel output data of the scrambler, parallel input data of the scrambler and the initial state of the scrambler is shown in the following formula: $D_{out}^{n+k} = D_m^{n+k} \oplus (Q_6^{n+1} Q_6^{n+2} \dots Q_6^{n+k})$; wherein different groups of input signal is subject to parallel scrambling process;

A parallel-serial conversion unit: After scrambling, different groups of input signal is recovered to serial data stream based on the time of the input signal.