Proceedings of the 10th WSEAS International Conference on CIRCUITS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp17-21)

Flexible Digital Scrambler/De-Scrambler System

MUNIR A. AL-ABSI
Department of Electrical Engineering
King Fahd University of Petroleum and Minerals
P.O. Box 1139, Dhahran 31261
SAUDI ARABIA

Abstract: The structural design of a flexible Scramble/De-Scrambler that uses a programmable length shift register and modulo-2 adder is presented. The key feature of the proposed design is its flexibility in varying the length of the delay elements pseudo-randomly and hence the encryption code. In addition, the encryption code can be re-programmed by the manufacturer through the EPROM involved in the design and by the user through the loadable pseudo-random sequence generator.

I. Introduction

Scramblers are commonly used in data communication system either to secure data or re-code periodic sequences or strings of binary data. The basic structure of scrambler/de-scrambler system was presented in [1]. The drawback of this structure is the relative ease of decoding the scrambled output due to the fixed length of the delay element. A modified version of this scrambler utilises a tapped shift registers[2]. Tapping is done by multiplexer which can include or exclude an *n* bit delay line. Again, here in this approach, it is not difficult to decode the scrambled output, although more time may be needed. In this work, we present the design of a flexible scrambler/de-scrambler system that overcomes the above problems. In addition more feature are available in terms of reliability and field programmability. In Section II, a review of the basic concept of scrambler/de-Scrambler system is presented. In Section III we present the realisation of programmable length shift register to be used in our design based on the idea presented in [2]. The design of new flexible scrambler system is presented in Section IV.

II. Scrambler/De-Scrambler Fundamentals

The general structure of scrambler/de-scrambler system is shown in Figure 1. It consists of two registers R_1 and R_2 of

lengths L_1 and L_2 respectively in addition to two modulo-2 adders. S is the scrambler input and T_1 is the scrambler output. Where T_2 is the de-scrambler input and R is the de-scrambler output.

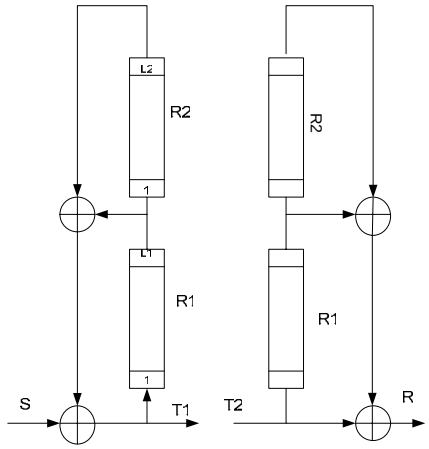

Figure1: Scrambler/De-Scrambler.

Notations

Let D represents the delay element, $D^k S$ represents a sequence S delayed by k bits.

Using the delay operators the following relations were reported in [1]

In the scrambler

$$S = T_1 \oplus T_1 D^{L1} \oplus T_1 D^{L1+L2}$$

$$S = T_1 \oplus F T_1$$

Where $F = D^{L1} \oplus D^{L1+L2}$

Or

$$T_1 = \frac{S}{1 \oplus F}$$

Where division stands for the inverse operation.

Similarly, in the de-scrambler

$$R = T_2 \oplus D^{L1} T_2 \oplus D^{L1+L2} T_2$$

Or

$$R = (1 \oplus F) T_2$$

In the absence of error, $T_1 = T_2$

The above equation can be written as

$$R = (1 \oplus F) T_2 = (1 \oplus F) T_1 = (1 \oplus F) \frac{S}{1 \oplus F} = S$$

From the above equation it is clear the de-scrambler output is the same as the scrambler input.

III. Flexible Structure Scrambler System

As can be seen from the above relations the scrambled data can be more encrypted if the length of the delay lines, $L_1$and L_2, are of variable sizes. Having such structure will make it difficult to decrypt the code. In what follows, we present the design of such a structure. The proposed system consists of an operational unit and control unit.

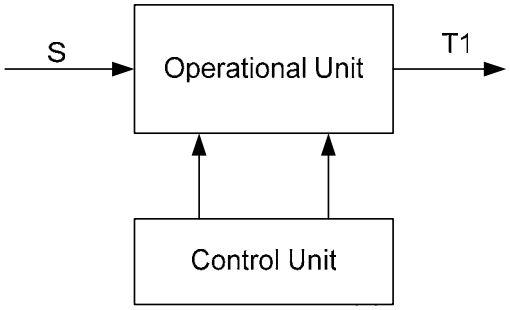

Figure 2: Flexible Scrambler block diagram

A- Operational Unit

The operational unit simply consists of a programmable length shift register (PLSR), and a Modulo-2 adder. The PLSR design is an adopted version of the one proposed in [2], in which PLSR of length N bits can be utilised using two N/2 bi-directional shift registers as shown in Figure 3. Half of the input data is streamed from left to right and the other half from right to left. The reader can refer to [2] for more details.

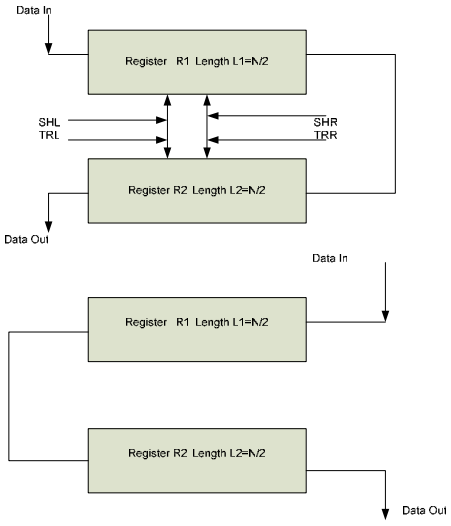

Figure3: Programmable shift register of length N bits

A suitable realisation of the PLSR in MOS technology was presented in [2]. We adopted the realisation to include the modulo-2 adders as shown in Figure 4 to implement the scrambler unit. In this adaptation we limit N (the total length of the PLSR) to even numbers for simplicity.

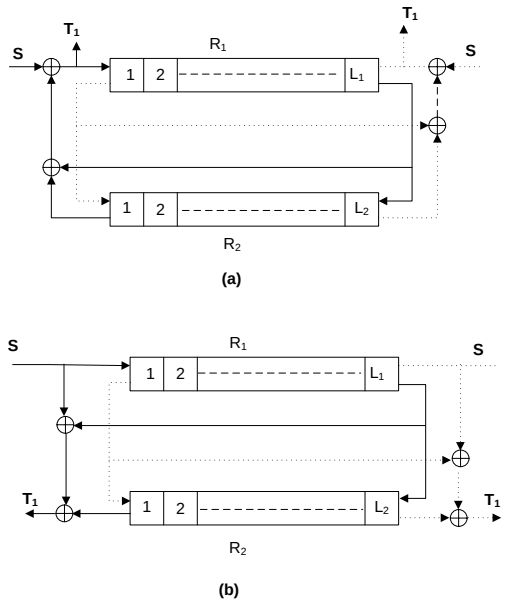

Figure 4: Scrambler unit configuration (a) Scrambler (b) de-scrambler

The operational unit consists of two bi-directional shift registers R_1 and R_2 each of length N/2 where N is the maximum delay required. During the first half of the input sequence, the input data bits are streamed through the system from left to right as indicated by the solid lines. During the second half, the route is changed and the data bits are streamed from right to left as indicated by the dashed lines. The control signals are to be received from the control unit, which will be described next.

B. Control unit

The control unit will generate the following control signal:

-SHR shift right

-TRR transfer right

-SHL shift left

-TRL transfer left

The control unit is designed in such a way that it can be programmed to suit different users with different codes and it consists of the following:

- A pseudo random sequence generator with loadable input
- Decoder with 128 outputs
- Counters with load enables
- EPROM
- Combinational circuit

The complete diagram of the scrambler systems is shown in Figure 5.

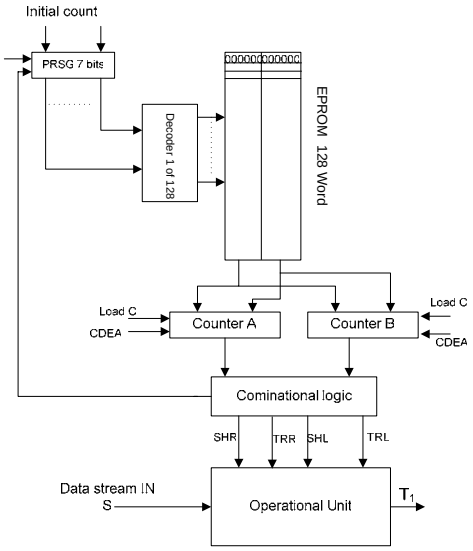

Figure5: Flexible scrambler system

Scrambler operation

The Scrambler operation can be described as follows:

Assume initially counters A and B are cleared and the pseudo random sequence generator (PRSG) is also cleared. At switch on the PRSG is loaded with an initial number. The1-out-of-128 decoder decodes the output of the PRSG. A memory location is addressed by the decoder output. The content of the memory location is loaded in counters A and B. A count down enable signal (CDEA) is activated and counter A is decremented till its count reaches zero. During this time the combinational logic circuit generates the shift right (SHR) and transfer right (TRR) signals and half of the input sequence is streamed into register R_1. A count down enable signal for counter B is activated (CDEB). Counter B is decremented till its count reaches zero. During this time, shift left (SHL) and transfer left (TRL) signals are activated and the second half of the input sequence is streamed into register R_2. When the content of counters A and B is zero, a Load c signal is activated and the two counters are re-loaded from the memory location again. The same operation is repeated till the message is finished. When new message has to be sent the PRSG is

enabled with (ENP) signal and new sequence is generated. The decoder will decode the new sequence and generate the equivalent address to new memory location. This new word sets new lengths for register R_1 and R_2. As a result, a new encryption code is generated and so that the scrambled output is different than the one before. The same operation is repeated over and over with different encryption code for each message.

Conclusion

We presented the design of flexible scrambler structure. The flexibility a rise from the fact that the controller is designed is such a way that it can be field programmable. Secondly, the same product can be sold to different customers without change in the design. All it need is to re-programme the EPROM to different words or change the sequence of words stored.

ACKNOWLEDGMENT

The author would like to thank King Fahd University of Petroleum and Minerals for supporting this work.

Reference

[1] K.Sam Shanmugam "Digital and analogue communication" John Wily Sons Inc, Canada 1979.

[2] Pere Daielson "A variable-length shift Register" IEEE Trans on Computers, Vol. C-32 No 11. November 1982.