ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Universal Scrambler by using Verilog HDL

Kushan B. Vadwala
M.Tech in VLSI & Embedded Systems, Ganpat University, India
Kushan0711@gmail.com

Abstract

Universal Scrambler (Randomizer) is a device that is used for encoding the message at the transmitter to build the message at the receiver by using decoding techniques,where encryption is referred in the digital arena, scrambling is usually referred in the analog arena.

So, it is a type of coding and if you want to decode the scrambled message you have only the initial value because there is not a mathematical relationship between input and output data of randomizer and also scrambler increases the number of high-to-low and low-to-high transitions in a data stream.

Keywords: LFSR, PRBS, Cryptography, Encryption, Randomizer, Polynomial Equation

Introduction

Universal Scramblers are a class of substitution encrypts and have been found to be suitable for various security requirements such as cable and satellite TV operators and mobile phone service providers. A Universal Scrambler is a coding operation which basically randomizes the data streams. In addition to its use as a stream cipher, a scrambler is commonly used to avoid long strings of 0’s and 1’s which are responsible for DC wander and synchronization problems in communication circuits.

Scramblers are well-known for encrypting audio and video signals in broadcasting and many other applications.The main features of Universal Scramblers are:Low Cost, Easy to use and High Speed Of Operation.

The scrambler is built as a shift register with stages depending on the given characteristic equations. There is feedback taken from stages according to the polynomials listed in the characteristic equation. On the transmitter side, the scrambler may be started with a preload of a specific data content. This is important later on when we talk about testing. This preload is also called seed. With a known seed and a known input pattern the scrambler output is deterministic, which means the output of the scrambler can be calculated. On the receiver side, the descrambler synchronizes automatically from the incoming data.

Design of Universal Scrambler

A Universal Scrambler contains of X shift registers and N XORs. The length of Universal Scrambler is 2^M-1.So,it generates M-1 bit sequence for a given PRBS initial state. Each incoming data bit is XORed with the current bit in the M-1 bit sequence. These M bits are re-written with the initial state of the Universal Scrambler.So,Descrambling can be done at the receiver side.

If we want to implement various protocols;then they should be as flexible as possible. We attain it by defining the execution for data types and widths both.The Universal Scrambler randomizes the input bit stream by XORing each bit with a PRBS generated by a LFSR.

$\mathbf{Y(x) = 1 + x^{-M} + x^{-N}}$; where M and N are integers.

Firstly,we calculate the feedback bit by XORing the Mth and Nth bit of the random number in LFSR. Then, we render the output bit by XORing inData with the feedback bit. Finally, we work out the new random number by shifting the feedback bit into the current random number.We can process up to steps bits of inData for the higher performance.

Whenever the transmission frame starts, all registers of serial scrambler reset to initial condition and then scrambles with input data repeatedly. By this operation, the scrambled output data becomes pseudo-random data and transmitted to the channel.

ISSN: 2277-9655
Impact Factor: 1.852

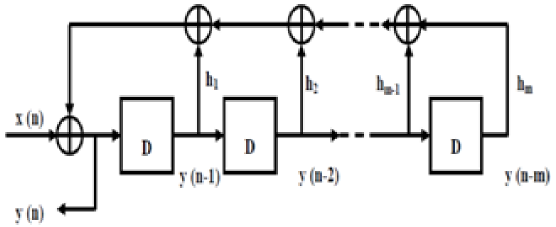


Fig 1. Universal Scrambler

Spread the Spectrum

There are many channels that are used to transmit data simultaneously in communication system. If one of the channels produces large amounts of noise to interfere other channels, it would be troublesome. Actually, if the channel transmitted repetitive sequences of data, from the spectrum point of view, it could have large amounts of energy concentrated at some frequencies. These large amounts of energy could lead to electromagnetic impulse (EMI) to affect other channels.

However, after scrambling, it becomes random data in time domain and the probability of 1 or 0 is equa1. On the other hand, the transmitted energy is spread more uniform across the transmission frequency band and minimizes the electromagnetic impulse (EMI) damage to other channels.

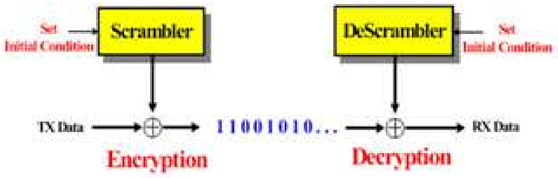
Data Transitions

The transmission format of signal can be classified into two categories: NRZ and RZ.When data is 1 in high level, it will not change level to low level until data is 0. The latter is on the contrary. When the data is 1 in high level, it will change level to low level after half interval of one bit time. Thus, the RZ format has more transitions than NRZ in the same transmitted data, and it is beneficial for receiver to synchronize with transmitter by these data transitions. However, the bandwidth requirement of RZ is twice that of NRZ.

Therefore, scrambler can be used to make the same effect on NRZ format for synchronization purpose. If the transmitted data bytes or control signals are long term 0 or 1 during transmission, after scrambling, the scrambled data will have more transitions and become more random. It can avoid the generation of transmitting a constant control signal for long periods.

Encryption

The scrambled data is the result of XOR operation on the input data and scrambler output sequences, the scrambled data is quite different from the input data. So, the data transmitted in channel is random and correlation of this sequence is very close to zero. It is random data until the receiver is descrambled, so it is like to encrypt the original data.



Advantages of Scrambling

The input data is scrambled by performing the XOR operation with the scrambler output sequences. Thus, the scrambled data becomes pseudo-random data what the original input data is. It is good to transmit pseudo-random data in the channel for the communication system. It has three advantages for transmitting pseudo-random data and using the scrambler circuit.

Simulation Results

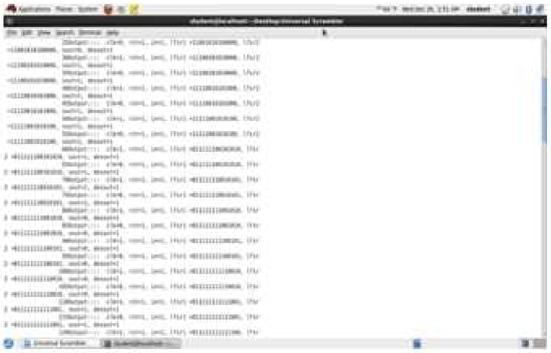


Fig 3. Scrambled Output

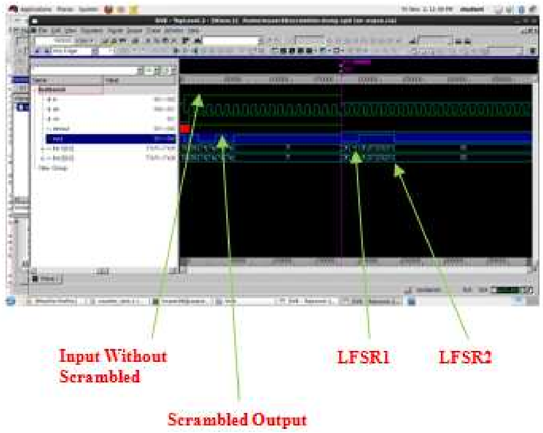


Fig 4. Waveform Of Scrambled Output

Conclusions

By using the Universal Scrambler, we can make Liner Feedback Shift Register (LFSR), Pseudo Random Binary Sequence Generator(PRBS),Cryptography, Randomizer, Encryption and Decryption.Scrambler is used for the security purposes.Scrambler (Randomizer) manipulates the data stream before transmitting.

Scrambler is used mainly in Digital Communications.

Acknowledgment

I wish to express our deep sense of gratitude to our supervisor **Mrs. Ekata Mehul**, Head, Department of VLSI and Embedded Systems, eiTRA who through their excellent guidance have enabled us to accomplish this work. They have been great source of inspiration to us. I am very grateful to them for guiding me how to conduct research and how to clearly & effectively present the work done.

I extend my sincere thanks to **Mrs. Ekata Mehul**, for providing the adequate means and support to pursue this work. Finally, I would like to add few heartfelt words for the people who were the part of the project in various ways, especially our friends and classmates who gave us unending support right from the beginning. Without their support and love we would not be where we are today.

References

[1] J. G. Proakis, M. Salehi. "Digital Communications", McGraw Hill, (2007).

[2] R. D. Gitlin, J. F. Hayes. "PCM jitter suppression by scrambling", Bell System Technical Journal 54(3): 569 – 593, 1975.

[3] H. Kasai, S. Senmoto and M. Matsushita, "On the timing information disappearance of digital transmission systems", IEEE Trans. on Communications, 22(8): 1114-1122, 1974.

[4] A. Huzii, S. Kondo. "Timing recovery and scramblers in data transmission", IEEE Trans. on Communications, 21(4):1072-1074, 1973.

[5] R.L. Freeman. "Reconstruction of a linear scrambler", John Wiley and Sons Inc., (1995).

[6] C. H. Lin et.al. "", IEEE Trans. on Circuit & Systems-II, 53(7): 558-562, 2006.

[7] M. Cluzeau. "Practical Data Communications", IEEE Trans. on Computers, 56(9): 1283-1291, 2007.

[8] "Parallel scrambler for high speed applications", IBM Technical Disclosure Bulletin, 28: 1063-1064, 1985.

[9] D. Y. Kim et. al. "Data Scrambler / Descrambler with Look Ahead", IEEE Trans. On Communications, 52(1): 54-61, 2004.