DESIGN & VERIFICATION OF PRBS FOR MAXIMAL LENGTH USING VHDL

Manikandan. S, Meena Priya Dharshini
CMR Institute of Technology, Bangalore
EmIL: msv29june@gmail.com, meenunandu@gmail.com

Abstract- Pseudo random binary sequence is essentially a random sequence of binary numbers. So PRBS generator is nothing but random binary number generator. It is ‘random’ in a sense that the value of an element of the sequence is independent of the values of any of the other elements. It is ‘pseudo’ because it is deterministic and after N elements it starts to repeat itself, unlike real random sequences. The implementation of PRBS generator is based on the linear feedback shift register (LFSR). The PRBS generator produces a predefined sequence of 1’s and 0’s, with 1 and 0 occurring with the same probability. A sequence of consecutive n*(2^n -1) bits comprise one data pattern, and this pattern will repeat itself over time.

Keywords: LFSR, Correlation, Random, Deterministic.

I. INTRODUCTION

Resilience:
The word “resilience” means the ability to adapt well to stress. It means that, overall you remain stable and maintain healthy levels of physical functioning in the face of disruption or chaos.

A resilient network is a network, which does not fail under any circumstances. Failure refers to a situation where the observed behavior of a system differs from its specified behavior. A failure occurs due to an error, caused by a fault. Faults can be hard or soft. For example a cable break is a hard failure whereas an intermittent noise in the network is a soft failure. Resilient implies flexible and adaptive yet at the same time fortified against all types of threats.

II. PRBS BASIC TECHNIQUES

Pseudo Random Binary Sequence is essentially a random sequence of binary numbers. It is random in a sense that the value of an element of the sequence is independent of the values of any of the other elements. It is pseudo because it is deterministic and after N elements it starts to repeat itself, unlike real random sequences.

A binary sequence (BS) is a sequence of N bits, a_j for j = 0, 1, ..., N - 1, i.e. m ones and N – m zeros. A binary sequence is pseudo-random (PRBS) if its autocorrelation function, the implementation of PRBS generator is based on the linear feedback shift register, which consists of ‘n’ master slave flip-flops. The PRBS generator produces a predefined sequence of 1’s and 0’s, with 1 and 0 occurring with the same probability.

Implementation

One of the two main parts of an LFSR is the shift register (the other being the feedback function). A shift register is a device whose identifying function is to shift its contents into adjacent positions within the register or, in the case of the position on the end, out of the register. The position on the other end is left empty unless some new content is shifted into the register. The contents of a shift register are usually thought of as being binary, that is, ones and zeros. If a shift register contains the bit pattern 1101, a shift (to the right in this case) would result in the contents being 0110; another shift yields 0011. After two more shifts, things tend to get boring since the shift register will never contain anything other than zeros. The conversion function can go either way -- fill the shift register positions all at once (parallel) and then shift them out (serial) or shift the contents into the register bit by bit (serial) and then read the contents after the register is full (parallel). The delay function simply shifts the bits from one end of the shift register to the other, providing a delay equal to the length of the shift register.

Figure 1 Shift Register
• **Feedback Action:**

In an LFSR, the bits contained in selected positions in the shift register are combined in some sort of function and the result is fed back into the register's input bit. By definition, the selected bit values are collected before the register is clocked and the result of the feedback function is inserted into the shift register during the shift, filling the position that is emptied as a result of the shift. Feedback around an LFSR's shift register comes from a selection of points (taps) in the register chain and constitutes XORing these taps to provide tap(s) back into the register. Register bits that do not need an input tap, operate as a standard shift register. It is this feedback that causes the register to loop through repetitive sequences of pseudo-random value. The choice of taps determines how many values there are in a given sequence before the sequence repeats. The implemented LFSR uses a one-to-many structure, rather than a many-to-one structure, since this structure always has the shortest clock-to-clock delay path. The feedback is done so as to make the system more stable and free from errors. Specific taps are taken from the tapping points and then by using the XOR operation on them they are feedback into the registers. The bit positions selected for use in the feedback function are called "taps". The list of the taps is known as the "tap sequence". By convention, the output bit of an LFSR that is n bits long is the nth bit; the input bit of an LFSR is bit 1

Tapping Actions:

An LFSR is one of a class of devices known as state machines. The contents of the register, the bits tapped for the feedback function, and the output of the feedback function together describe the state of the LFSR. With each shift, the LFSR moves to a new state. (There is one exception to this -- when the contents of the register are all zeroes, the LFSR will never change state.) For any given state, there can be only one succeeding state. The reverse is also true: any given state can have only one preceding state. For the rest of this discussion, only the contents of the register will be used to describe the state of the LFSR. The state of an LFSR that is n bits long can be any one of 2^n different values. The largest state space possible for such an LFSR will be 2^n - 1 (all possible values minus the zero state). Because each state can have only once succeeding state, an LFSR with a maximal length tap sequence will pass through every non-zero state once and only once before repeating a state. The period of an LFSR is defined as the length of the stream before it repeats. The period, like the state space, is tied to the tap sequence and the starting value. As a matter of fact, the period is equal to the size of the state space. The longest period possible corresponds to the largest possible state space, which is produced by a maximal length tap sequence. (Hence "maximal length")

LFSR's can have multiple maximal length tap sequences. A maximal length tap sequence also describes the exponents in what is known as a primitive polynomial mod 2.

• **Characteristics of output stream:**

By definition, the period of an LFSR is the length of the output stream before it repeats. Besides being non-repetitive, a period of a maximal length stream has other features that are characteristic of random streams. In one period of a maximal length stream, the sum of all ones will be one greater than the sum of all zeroes. In a random stream, the difference between the two sums will tend to grow progressively smaller in proportion to the length of the stream as the stream gets longer. In an infinite random stream, the sums will be equal. A run is a pattern of equal values in the bit stream. A bit stream like 10110100 has six runs of the following lengths in order: 1, 1, 2, 1, 1, 2. One period of an n-bit LFSR with a maximal length tap sequence will have 2^n(n-1) runs. A random stream of sufficient length shows similar behavior statistically.Take the stream of bits in one period of an LFSR with a maximal length tap sequence and circularly shift it any number of bits less than the total length. One characteristic of the LFSR output not shared with a random stream is that the LFSR stream is deterministic.

III. SIMULATION RESULTS

A separate program module for D-Flip-Flop was written and this module was called 16 times in the main program to get the 16-bit shift register. Now the taps 1, 2, 4 and 15 were taken out and XORed together and then was fed back to the first bit as an input to the shift register. The output to the PRBS generator was taken from all the 16-bits of the shift register.
IV. CONCLUSION

In the code for the PRBS tapings are taken so as to get the maximum range of the binary numbers generated. In the developed code tapings are taken from 1, 2, 4 and 15 taps so as to obtain the maximum length of binary digits produced. Initially when the reset is kept at zero the outputs of each of the registers is uninitialized and hence the output is uninitialized as well. However as soon as the reset is made high the output of all the registers start coming out. A dead lock condition arises in the case when the initial input into the first register as output of the XOR gate are all 0’s. Under this condition the output of all the register of the PRBS Generator remains as 0 at all instants of time. Therefore it is necessary that the initial input to the PRBS Generator be equal to 1, the output of the XOR gate. The code for implementing the above circuit was written and hence the simulation results were generated and tested.

REFERENCES