Chapter 1

Binary Source

The class *Binary Source* corresponds to a PRBS (*pseudo-random binary sequence*) generator. This is the way as it is possible to recreate a real data sequence, which is random in fact. A random binary sequence is a statistically independent sequence of 0's and 1's each occurring with probability of 50% [1].

One of the upgrades in the *C++* virtual transmitter consisted of adding a feature that allows the user to make the bit stream setting, in other words make the manual initialization for the PRBS. Obviously, the introduction of such feature implied a validation and error functions. The other upgrade consisted of improving and ensure the proper functioning of the PRBS generator, that is described in detail in the section below.

1.1 Pseudo-Random Test Patterns

In order to validate the algorithm which simulates the PRBS generator, there were made several tests over it, such as sequences validation and periodicity test.

A pseudo-random binary sequence is supposed to be a periodic binary sequence with an autocorrelation function that resembles the autocorrelation function of a random binary sequence. This PRBS is generated by a LFSR (*linear feedback shift register*), represented on figure 1.1), whose the arrangement consists of binary storage elements and feedback logic. Binary sequences are shifted through the shift register in response to clock pulses. The contents are logically combined, by a *XOR* in this case, to produce the input to the first stage, which means the initial contents and feedback logic determine the successive contents of the shift register [1].

For simulation effects, the considered initial contents consisted on the first state set to '1' and the following ones set to '0'. If a LFSR contains all zeros at any time, it will be stuck in that state forever. Since there are exactly $2^m - 1$ non-zero states, the maximum period cannot exceed $2^m - 1$, as

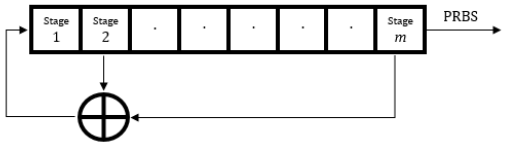

Figure 1.1: m-stage linear feedback shift register for generating PRBS

well [1]. Moreover, the number of distinct states in an m-state shift register is 2^m, so the sequence of states and the output sequences must be periodic, with a period of at most 2^m.

The following table 1.1 is result of various simulations in order to find the respective feedback polynomials to each sequence pattern length, which equals the LFSR length, or number of stages. The polynomial coefficients were taken from [1], [2] and [3].

Right after, it is presented an exampling table 1.2 with some binary sequences generated by applying the previous polynomials. The table includes the pattern length, m, the period length of the binary sequence, $2^m - 1$, and also the data length, $2^m + m - 2$, which equals the period length plus the necessary margin to ensure the validation of the last sequence values.

m	feedback polynomial	ref.	m	feedback polynomial	ref.
1	[1]	[1]	14	[2,12,13,14]	[2]
2	[1,2]	[1]	15	[1,15]	[2]
3	[1,3]	[1]	16	[2,3,5,16]	[2]
4	[1,4]	[1]	17	[14,17]	[2]
5	[1,2,4,5]	[1]	18	[1,2,5,18]	[2]
6	[2,3,5,6]	[1]	19	[1,2,5,19]	[2]
7	[1,7]	[2]	20	[8,9,17,20]	[3]
8	[2,3,4,8]	[1]	21	[19,21]	[2]
9	[3,4,6,9]	[1]	22	[1,22]	[2]
10	[3,10]	[1]	23	[5,23]	[2]
11	[2,5,8,11]	[1]	24	[1,3,4,24]	[2]
12	[1,4,6,12]	[1]	25	[3,25]	[2]
13	[1,3,4,13]	[1]	26	[8,24,25,26]	[3]

Table 1.1: feedback polynomials for each shift register stage

Finally, it was proceeded to the validation of binary sequences of any length. The table 1.3 exemplifies the validation process, presenting all the possible values inside a binary sequence generated with a certain pattern length. In this case, pattern length equals 5, whose the feedback polynomial is $(x^5 + x^4 + x^2 + x^1 + 1)$ according to table 1.1.

m	$2^m - 1$	$2^m + m - 2$	binary sequence
1	1	1	1
2	3	4	0110
3	7	9	101001110
4	15	18	000111101011001000
5	31	35	00101011101100011111001101001000010
6	63	68	10000111000001011111100101010001100...
⋮	⋮	⋮	⋮
26	67108863	67108888	...

Table 1.2: binary sequences generated for different pattern lengths, from 1 to 26

state	binary value	decimal value
1	01010	5
2	01010	10
3	10101	21
4	01011	11
5	10111	23
6	01110	14
7	11101	29
8	11011	27
9	10110	22
10	01100	12
11	11000	24
12	10001	17
13	00011	3
14	00111	7
15	01111	15
16	11111	31
17	11110	30
18	11100	28
19	11001	25
20	10011	19
21	00110	6
22	01101	13
23	11010	26
24	10100	20
25	01001	9
26	10010	18
27	00100	4
28	01000	8
29	10000	16
30	00001	1
31	00010	2

Table 1.3: validation for generated sequence with pattern length of 5

Bibliography

[1] P. S. K. Jeruchim, M.C.; Balaban, *Simulation of Communication Systems*, ch. 3, pp. 284–285. Plenum, 1992.

[2] P. Alfke, "Efficient shift registers, lfsr counters, and long pseudo-random sequence generators," *http://www. xilinx. com/bvdocs/appnotes/xapp052. pdf*, 1998.

[3] C. . . N. W. Instruments, "Tables of m-sequence feedback taps." $www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm$