Lab 05

Design of a Randomizer

1 Design Specification

- Using a linear feedback shift register (LFSR), design a pseudorandom binary sequence (PRBS) generator that implements the polynomial $1 + X^{14} + X^{15}$ shown in Figure 1 which is used to randomize a sequence of binary inputs.
- The randomizer is initialized with the vector:
  - [LSB] 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 [MSB].
- In addition to clock (`clk`) and asynchronous reset (`reset`), include a synchronous seed load (`load`) and clock enable (`en`) inputs
- Validation: using the following input data sequence and the corresponding data output to validate your design
  - Input Data (Hex):
    - AC BC D2 11 4D AE 15 77 C6 DB F4 C9
  - Randomized Data (Hex):
    - 55 8A C4 A5 3A 17 24 E1 63 AC 2B F9

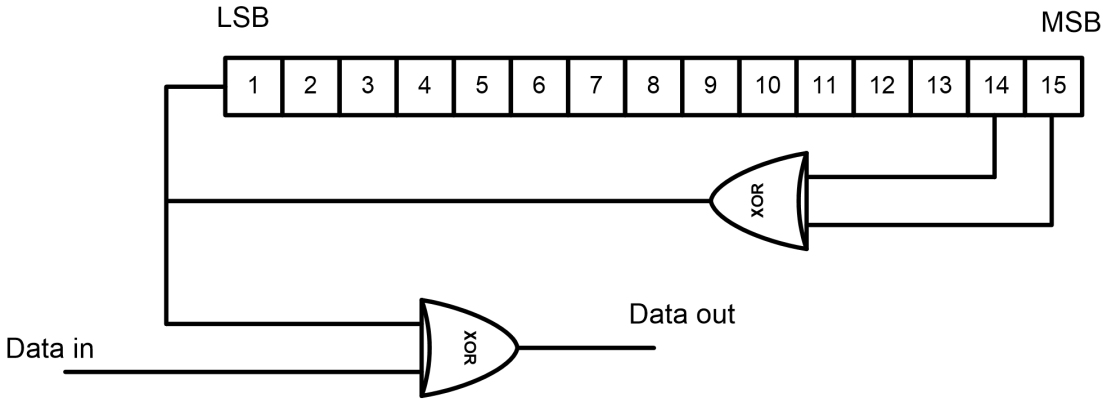

Figure 5: PRBS generator for Data randomization

Table 1: Intermediate data values for validation

Initialized vector 011 0111 0001 0101

No	Shift Register															XOR #1	Data in		Data out	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		0x	0b	0b	0x
1	**0**	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	1		**1**	**0**	
2	1	0	1	1	0	1	1	1	0	0	0	1	0	1	0	1	**A**	**0**	**1**	**5**
3	1	1	0	1	1	0	1	1	1	0	0	0	1	0	1	1		**1**	**0**	
4	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	1		**0**	**1**	
5	1	1	1	1	0	1	1	0	1	1	1	0	0	0	1	1	**C**	**1**	**0**	**5**
6	1	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0		**1**	**1**	

7	0	1	1	1	1	1	0	1	1	0	1	1	1	0	0	0		**0**	**0**	
8	0	0	1	1	1	1	1	0	1	1	0	1	1	1	0	1		**0**	**1**	
9	1	0	0	1	1	1	1	1	0	1	1	0	1	1	1	0	**B**	**1**	**1**	**8**
10	0	1	0	0	1	1	1	1	1	0	1	1	0	1	1	0		**0**	**0**	
11	0	0	1	0	0	1	1	1	1	1	0	1	1	0	1	1		**1**	**0**	
12	1	0	0	1	0	0	1	1	1	1	1	0	1	1	0	1		**1**	**0**	
13	1	1	0	0	1	0	0	1	1	1	1	1	0	1	1	0	**C**	**1**	**1**	**A**
14	0	1	1	0	0	1	0	0	1	1	1	1	1	0	1	1		**1**	**0**	
15	1	0	1	1	0	0	1	0	0	1	1	1	1	1	0	1		**0**	**1**	
16	1	1	0	1	1	0	0	1	0	0	1	1	1	1	1	0		**0**	**0**	
17	0	1	1	0	1	1	0	0	1	0	0	1	1	1	1	0	**D**	**1**	**1**	**C**
18	0	0	1	1	0	1	1	0	0	1	0	0	1	1	1	0		**1**	**1**	
19	0	0	0	1	1	0	1	1	0	0	1	0	0	1	1	0		**0**	**0**	
20	0	0	0	0	1	1	0	1	1	0	0	1	0	0	1	1		**1**	**0**	
21	1	0	0	0	0	1	1	0	1	1	0	0	1	0	0	0	**2**	**0**	**0**	**4**
22	0	1	0	0	0	0	1	1	0	1	1	0	0	1	0	1		**0**	**1**	
23	1	0	1	0	0	0	0	1	1	0	1	1	0	0	1	1		**1**	**0**	
24	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0		**0**	**0**	
25	0	1	1	0	1	0	0	0	0	1	1	0	1	1	0	1	**1**	**0**	**1**	**A**
26	1	0	1	1	0	1	0	0	0	0	1	1	0	1	1	0		**0**	**0**	
27	0	1	0	1	1	0	1	0	0	0	0	1	1	0	1	1		**0**	**1**	
28	1	0	1	0	1	1	0	1	0	0	0	0	1	1	0	1		**1**	**0**	
29	1	1	0	1	0	1	1	0	1	0	0	0	0	1	1	0	**1**	**0**	**0**	**5**
30	0	1	1	0	1	0	1	1	0	1	0	0	0	0	1	1		**0**	**1**	
31	1	0	1	1	0	1	0	1	1	0	1	0	0	0	0	0		**0**	**0**	
32	0	1	0	1	1	0	1	0	1	1	0	1	0	0	0	0		**1**	**1**	
33	0	0	1	0	1	1	0	1	0	1	1	0	1	0	0	0	**4**	**0**	**0**	**3**
34	0	0	0	1	0	1	1	0	1	0	1	1	0	1	0	1		**1**	**0**	
35	1	0	0	0	1	0	1	1	0	1	0	1	1	0	1	1		**0**	**1**	
36	1	1	0	0	0	1	0	1	1	0	1	0	1	1	0	1		**0**	**1**	
37	1	1	1	0	0	0	1	0	1	1	0	1	0	1	1	0	**D**	**1**	**1**	**A**
38	0	1	1	1	0	0	0	1	0	1	1	0	1	0	1	1		**1**	**0**	
39	1	0	1	1	1	0	0	0	1	0	1	1	0	1	0	1		**0**	**1**	
40	1	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1		**1**	**0**	
41	1	1	1	0	1	1	1	0	0	0	1	0	1	1	0	1	**A**	**1**	**0**	**1**
42	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1	0		**0**	**0**	
43	0	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1		**1**	**0**	
44	1	0	1	1	1	1	0	1	1	1	0	0	0	1	0	1		**0**	**1**	
45	1	1	0	1	1	1	1	0	1	1	1	0	0	0	1	1	**E**	**1**	**0**	**7**
46	1	1	1	0	1	1	1	1	0	1	1	1	0	0	0	0		**1**	**1**	
47	0	1	1	1	0	1	1	1	1	0	1	1	1	0	0	0		**1**	**1**	
48	0	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1		**0**	**1**	
49	1	0	0	1	1	1	0	1	1	1	1	0	1	1	1	0	**1**	**0**	**0**	**2**
50	0	1	0	0	1	1	1	0	1	1	1	1	0	1	1	0		**0**	**0**	
51	0	0	1	0	0	1	1	1	0	1	1	1	1	0	1	1		**0**	**1**	
52	1	0	0	1	0	0	1	1	1	0	1	1	1	1	0	1		**1**	**0**	
53	1	1	0	0	1	0	0	1	1	1	0	1	1	1	1	0	**5**	**0**	**0**	**4**
54	0	1	1	0	0	1	0	0	1	1	1	0	1	1	1	0		**1**	**1**	

55	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	0		**0**	**0**	
56	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1		**1**	**0**	
57	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1		**0**	**1**	
58	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	**7**	**1**	**1**	**E**
59	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	0		**1**	**1**	
60	0	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1		**1**	**0**	
61	1	0	0	1	1	0	0	0	1	1	0	0	1	0	0	0		**0**	**0**	
62	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	1	**7**	**1**	**0**	**1**
63	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	1		**1**	**0**	
64	1	1	0	1	0	0	1	1	0	0	0	1	1	0	0	0		**1**	**1**	
65	0	1	1	0	1	0	0	1	1	0	0	0	1	1	0	1		**1**	**0**	
66	1	0	1	1	0	1	0	0	1	1	0	0	0	1	1	0	**C**	**1**	**1**	**6**
67	0	1	0	1	1	0	1	0	0	1	1	0	0	0	1	1		**0**	**1**	
68	1	0	1	0	1	1	0	1	0	0	1	1	0	0	0	0		**0**	**0**	
69	0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	0		**0**	**0**	
70	0	0	1	0	1	0	1	1	0	1	0	0	1	1	0	1	**6**	**1**	**0**	**3**
71	1	0	0	1	0	1	0	1	1	0	1	0	0	1	1	0		**1**	**1**	
72	0	1	0	0	1	0	1	0	1	1	0	1	0	0	1	1		**0**	**1**	
73	1	0	1	0	0	1	0	1	0	1	1	0	1	0	0	0		**1**	**1**	
74	0	1	0	1	0	0	1	0	1	0	1	1	0	1	0	1	**D**	**1**	**0**	**A**
75	1	0	1	0	1	0	0	1	0	1	0	1	1	0	1	1		**0**	**1**	
76	1	1	0	1	0	1	0	0	1	0	1	0	1	1	0	1		**1**	**0**	
77	1	1	1	0	1	0	1	0	0	1	0	1	0	1	1	0		**1**	**1**	
78	0	1	1	1	0	1	0	1	0	0	1	0	1	0	1	1	**B**	**0**	**1**	**C**
79	1	0	1	1	1	0	1	0	1	0	0	1	0	1	0	1		**1**	**0**	
80	1	1	0	1	1	1	0	1	0	1	0	0	1	0	1	1		**1**	**0**	
81	1	1	1	0	1	1	1	0	1	0	1	0	0	1	0	1		**1**	**0**	
82	1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	**F**	**1**	**0**	**2**
83	1	1	1	1	1	0	1	1	1	0	1	0	1	0	0	0		**1**	**1**	
84	0	1	1	1	1	1	0	1	1	1	0	1	0	1	0	1		**1**	**0**	
85	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1		**0**	**1**	
86	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	**4**	**1**	**0**	**B**
87	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	1		**0**	**1**	
88	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1		**0**	**1**	
89	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0		**1**	**1**	
90	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	0	**C**	**1**	**1**	**F**
91	0	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1		**0**	**1**	
92	1	0	0	1	1	1	1	1	0	1	1	1	1	1	0	1		**0**	**1**	
93	1	1	0	0	1	1	1	1	1	0	1	1	1	1	1	0		**1**	**1**	
94	0	1	1	0	0	1	1	1	1	1	0	1	1	1	1	0	**9**	**0**	**0**	**9**
95	0	0	1	1	0	0	1	1	1	1	1	0	1	1	1	0		**0**	**0**	
96	0	0	0	1	1	0	0	1	1	1	1	1	0	1	1	0		**1**	**1**	

2 Requirements

1. Using RTL verilog, implement the above randomizer using synchronous sequential logic.

2. Write a testbench that verifies the RTL code
3. Compile and simulate your design