Giao Tran

N-bit Linear Feedback Shift Register (LFSR)

1. SystemVerilog RTL model

```
1   //RTL Model for Linear Feedback Shift Register
2   module lfsr
3   #(parameter N = 4) // Number of bits for LFSR, could be 2,3,4,5,6,7,8
4   (
5     input logic clk, reset, load_seed,
6     input logic[N-1:0] seed_data,
7     output logic lfsr_done,
8     output logic[N-1:0] lfsr_data
9   );
10
11  reg XOR;
12  reg [N:1]shift;
13
14  always @ (posedge clk)
15      begin
16          if(reset==1)
17              begin
18                  if(load_seed==1)
19                      shift<=seed_data;
20
21                  else if (load_seed==0)
22                      shift<={shift[N-1:1], XOR};
23              end
24      end
25
26  always @ (*)
27      begin
28          XOR=shift[N]^shift[N-1]; //the value from register N and N-1 are what being fed
            in XOR
29      end
30
31  //alternatively I also can do what is shown below for N=2 to N=8
32  /*always @ (*) begin
33      case (N)
34      2: begin
35              XOR=shift[2]^shift[1];
36          end
37      3: begin
38              XOR=shift[3]^shift[2];
39          end
40      4: begin
41              XOR=shift[4]^shift[3];
42          end
43      5: begin
44              XOR=shift[5]^shift[4];
45          end
46      6: begin
47              XOR=shift[6]^shift[5];
48          end
49      7: begin
50              XOR=shift[7]^shift[6];
51          end
52      8: begin
53              XOR=shift[8]^shift[7];
54          end
55      endcase
56  end*/
57
58
59  assign lfsr_data=shift[N:1];
60  assign lfsr_done=(shift[N:1]==load_seed);
61
62  endmodule: lfsr
```

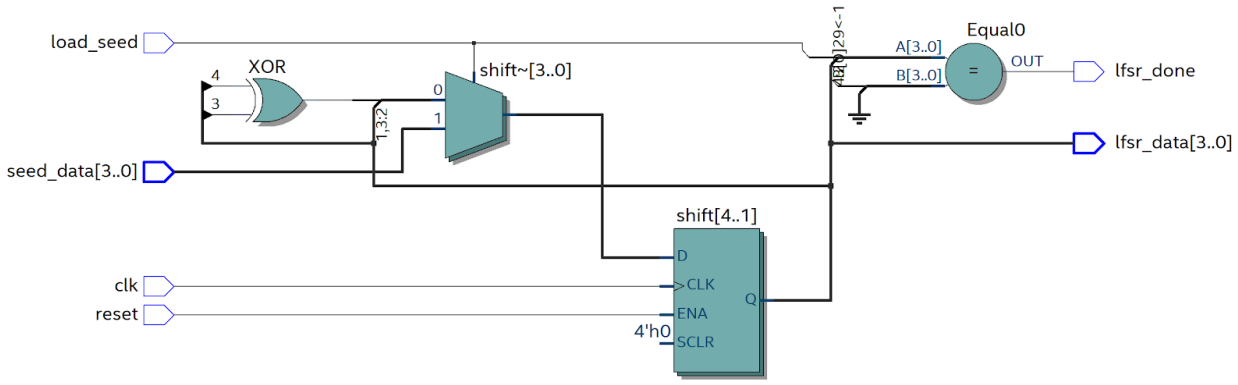
2. Resource usage table for N=4

	Resource	Usage
1	Estimated ALUTs Used	5
1	-- Combinational ALUTs	5
2	-- Memory ALUTs	0
3	-- LUT_REGs	0
2	Dedicated logic registers	4
3		
4	Estimated ALUTs Unavailable	0
1	-- Due to unpartnered combinational logic	0
2	-- Due to Memory ALUTs	0
5		
6	Total combinational functions	5
7	Combinational ALUT usage by number of inputs	
1	-- 7 input functions	0
2	-- 6 input functions	0
3	-- 5 input functions	1
4	-- 4 input functions	1
5	-- <=3 input functions	3
8		
9	Combinational ALUTs by mode	
1	-- normal mode	5
2	-- extended LUT mode	0
3	-- arithmetic mode	0
4	-- shared arithmetic mode	0
10		
11	Estimated ALUT/register pairs used	5
12		
13	Total registers	4
1	-- Dedicated logic registers	4
2	-- I/O registers	0
3	-- LUT_REGs	0
14		
15		
16	I/O pins	12
17		
18	DSP block 18-bit elements	0
19		
20	Maximum fan-out node	load_seed~input
21	Maximum fan-out	5
22	Total fan-out	47
23	Average fan-out	1.42

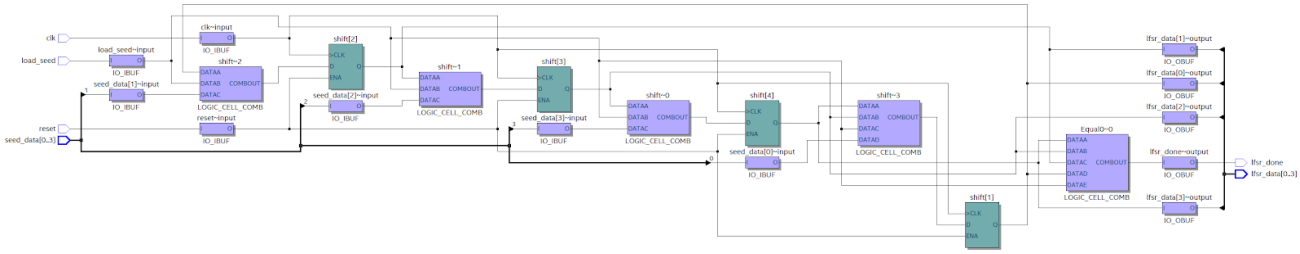
Number of ALUT: 5 (12 I/O pins)
Number of Functions: 5

1 5 input functions
1 4 input functions
3 3 input functions

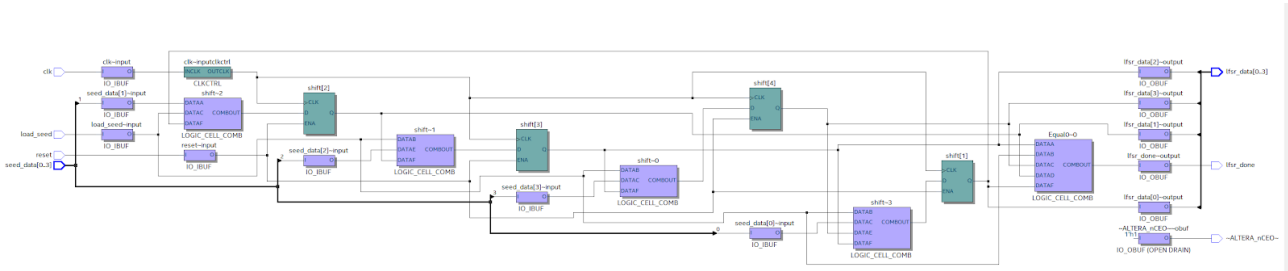
3. RTL schematic for N=4



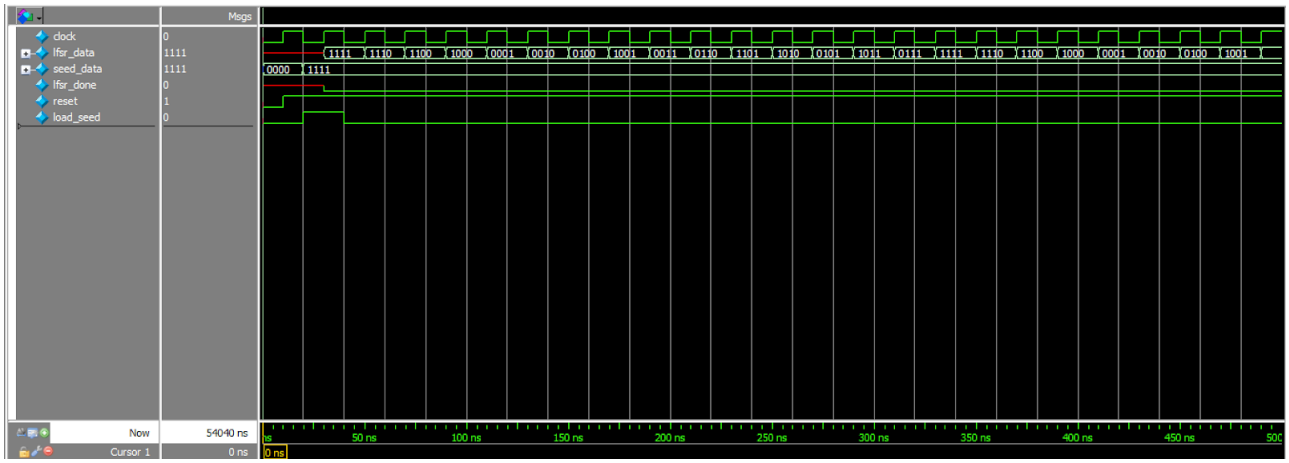
4. Post mapping schematic for N=4



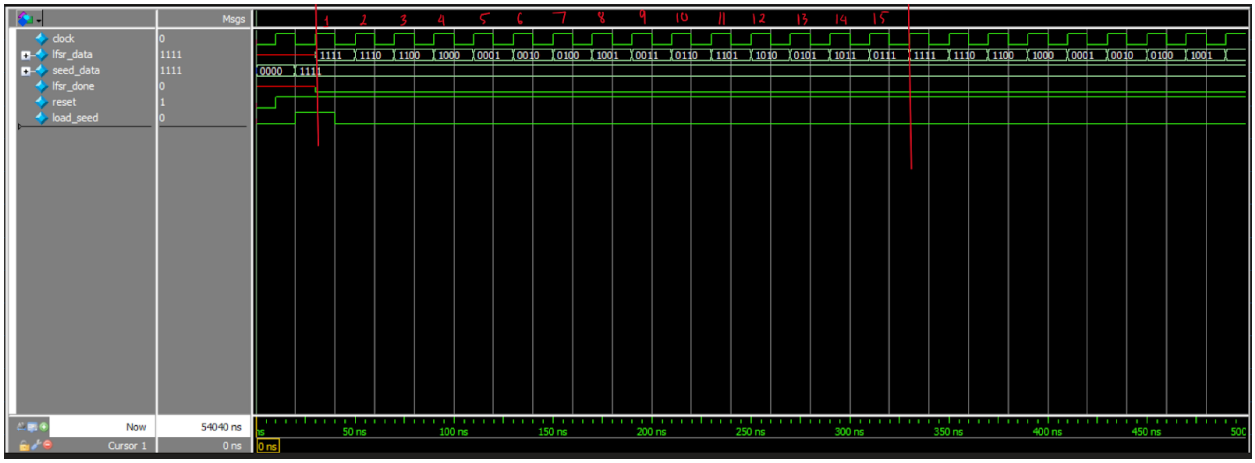
5. Post fitting schematic for N=4



6. Waveform simulation for N=4



Closer analysis of waveform



We can see that the pattern repeats every 15 cycles as stated by the polynomial equation **period= 2^N-1= 2^4-1=15 cycles**.

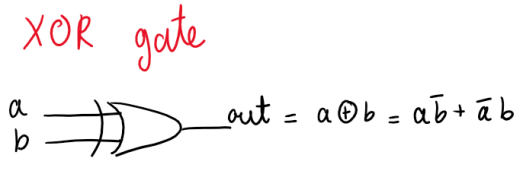
7. Transcript for N=4

```
# -- Compiling module lfsr_testbench
#
# Top level modules:
#       lfsr_testbench
# End time: 21:36:38 on Feb 10,2021, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
ModelSim> vsim work.lfsr_testbench
# vsim work.lfsr_testbench
# Start time: 21:36:46 on Feb 10,2021
# Loading sv_std.std
# Loading work.lfsr_testbench
# Loading work.lfsr
```

```
#  time=11410,  reset=1  clk=1  load_seed=0  count= 7
#  time=11420,  reset=1  clk=0  load_seed=0  count= 7
#  time=11430,  reset=1  clk=1  load_seed=0  count=15
#  time=11440,  reset=1  clk=0  load_seed=0  count=15
#  time=11450,  reset=1  clk=1  load_seed=0  count=14
#  time=11460,  reset=1  clk=0  load_seed=0  count=14
#  time=11470,  reset=1  clk=1  load_seed=0  count=12
#  time=11480,  reset=1  clk=0  load_seed=0  count=12
#  time=11490,  reset=1  clk=1  load_seed=0  count= 8
#  time=11500,  reset=1  clk=0  load_seed=0  count= 8
#  time=11510,  reset=1  clk=1  load_seed=0  count= 1
#  time=11520,  reset=1  clk=0  load_seed=0  count= 1
#  time=11530,  reset=1  clk=1  load_seed=0  count= 2
#  time=11540,  reset=1  clk=0  load_seed=0  count= 2
#  time=11550,  reset=1  clk=1  load_seed=0  count= 4
#  time=11560,  reset=1  clk=0  load_seed=0  count= 4
#  time=11570,  reset=1  clk=1  load_seed=0  count= 9
#  time=11580,  reset=1  clk=0  load_seed=0  count= 9
#  time=11590,  reset=1  clk=1  load_seed=0  count= 3
#  time=11600,  reset=1  clk=0  load_seed=0  count= 3
#  time=11610,  reset=1  clk=1  load_seed=0  count= 6
#  time=11620,  reset=1  clk=0  load_seed=0  count= 6
#  time=11630,  reset=1  clk=1  load_seed=0  count=13
#  time=11640,  reset=1  clk=0  load_seed=0  count=13
#  time=11650,  reset=1  clk=1  load_seed=0  count=10
#  time=11660,  reset=1  clk=0  load_seed=0  count=10
#  time=11670,  reset=1  clk=1  load_seed=0  count= 5
#  time=11680,  reset=1  clk=0  load_seed=0  count= 5
#  time=11690,  reset=1  clk=1  load_seed=0  count=11
#  time=11700,  reset=1  clk=0  load_seed=0  count=11
#  time=11710,  reset=1  clk=1  load_seed=0  count= 7
#  time=11720,  reset=1  clk=0  load_seed=0  count= 7
#  time=11730,  reset=1  clk=1  load_seed=0  count=15
#  time=11740,  reset=1  clk=0  load_seed=0  count=15
#  time=11750,  reset=1  clk=1  load_seed=0  count=14
#  time=11760,  reset=1  clk=0  load_seed=0  count=14
```

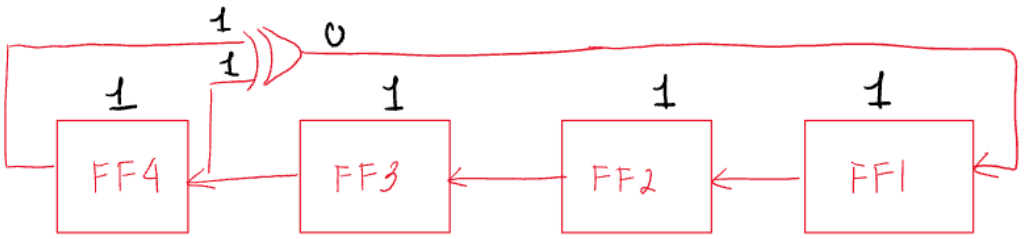
LFSR analysis:

For N=4, the LFSR has 4 shift registers with a XOR gate that receives the most left (FF4) and second to most (FF3) values as inputs, while this XOR's output is fed into the rightmost shift register (FF1) as the bits continue to shift left. The truth table for the XOR gate is shown below:

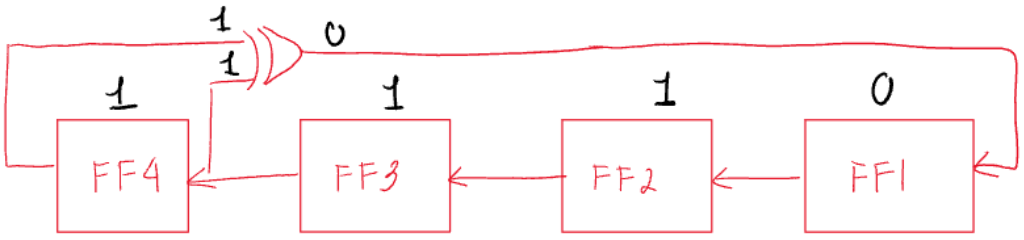


a	b	a⊕b
0	0	0
0	1	1
1	0	1
1	1	0

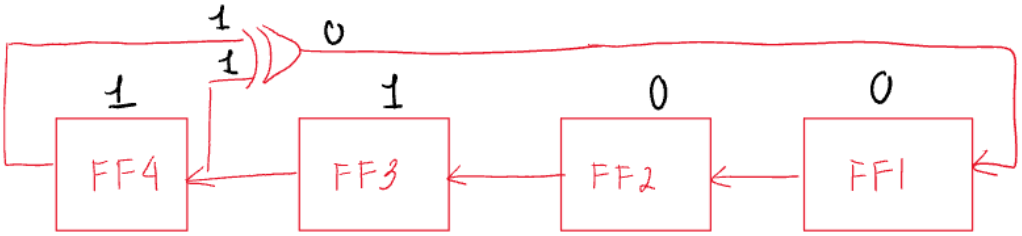
The truth table explains the output lfsr_data from the simulation as shown in the diagram below:



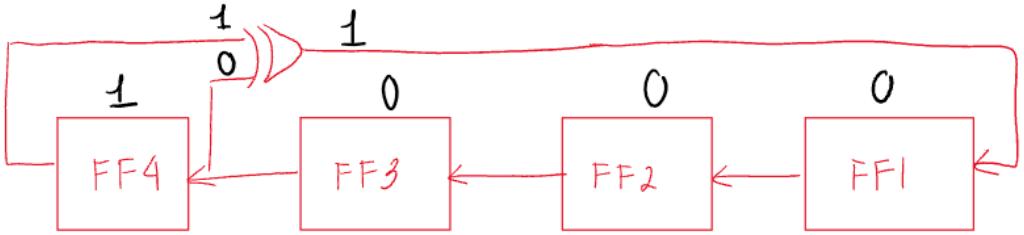
⇒ lsfr_data = 1110



⇒ lsfr_data = 1100



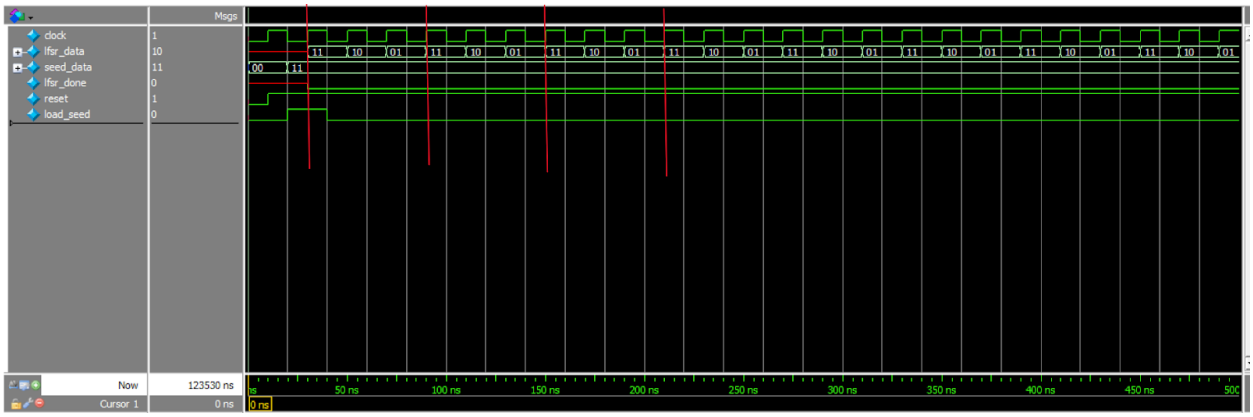
⇒ lsfr_data = 1000



⇒ lsfr_data = 0001

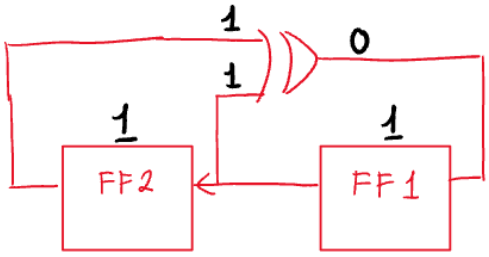
. . . so on

8. Waveform simulation of N=2

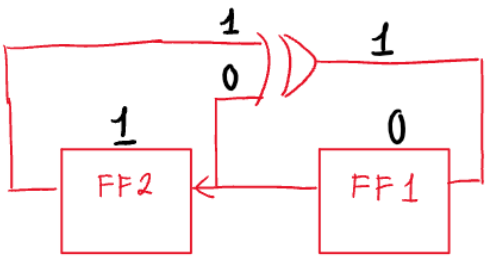


Repeats every **Period= 2^N-1= 2^2-1 =3 cycles**

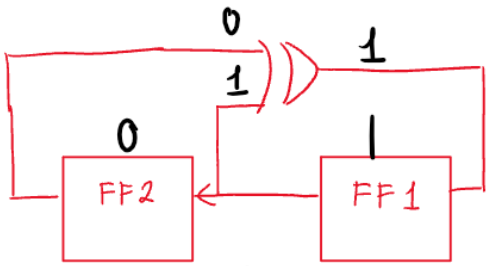
Diagram to explain how the bits are shifted and changed due to the XOR.



⇒ lsfr_ data = 10

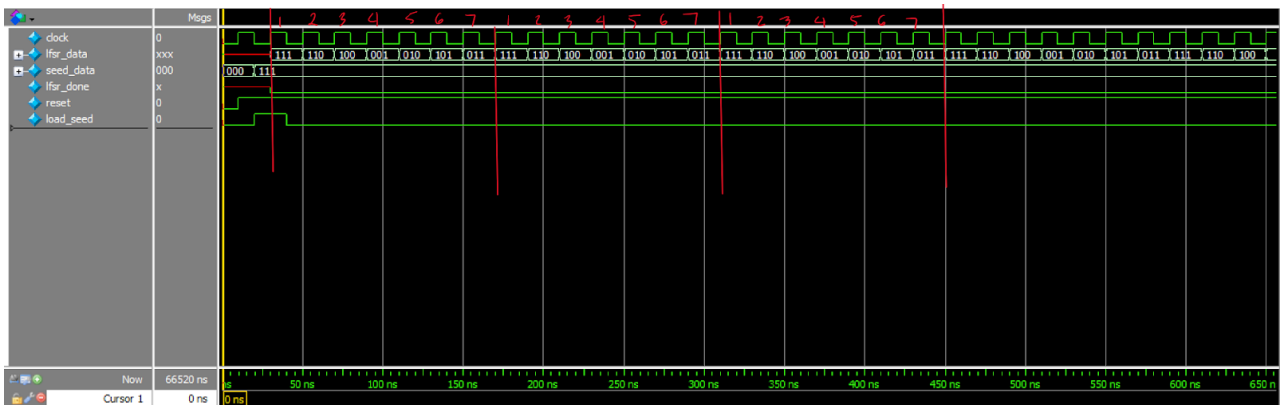


⇒ lsfr_ data = 0 1



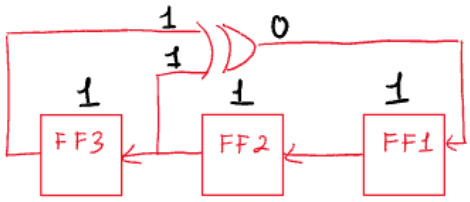
⇒ lsfr_ data = 1 1

9. Waveform simulation for N=3

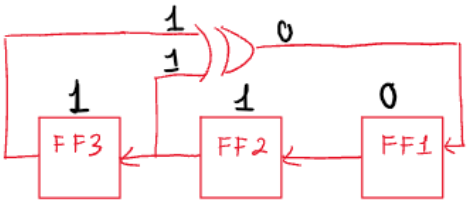


Repeats every **Period= 2^N-1= 2^3-1= 7 cycles**

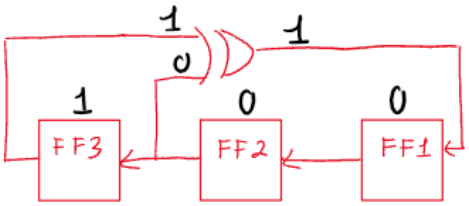
Diagram to explain how the bits are shifted and changed due to the XOR.



⇒ lsfr_ data = 110



⇒ lsfr_ data = 100



⇒ lsfr_ data = 001

. . . so on