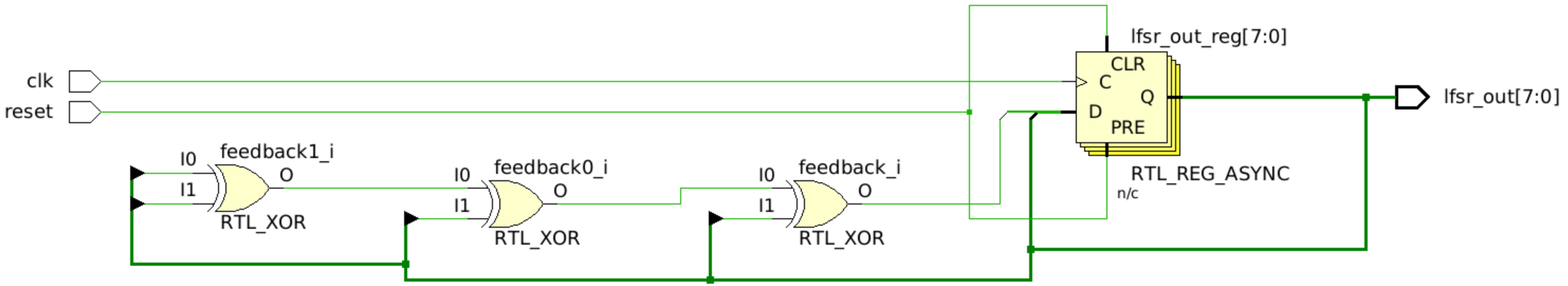
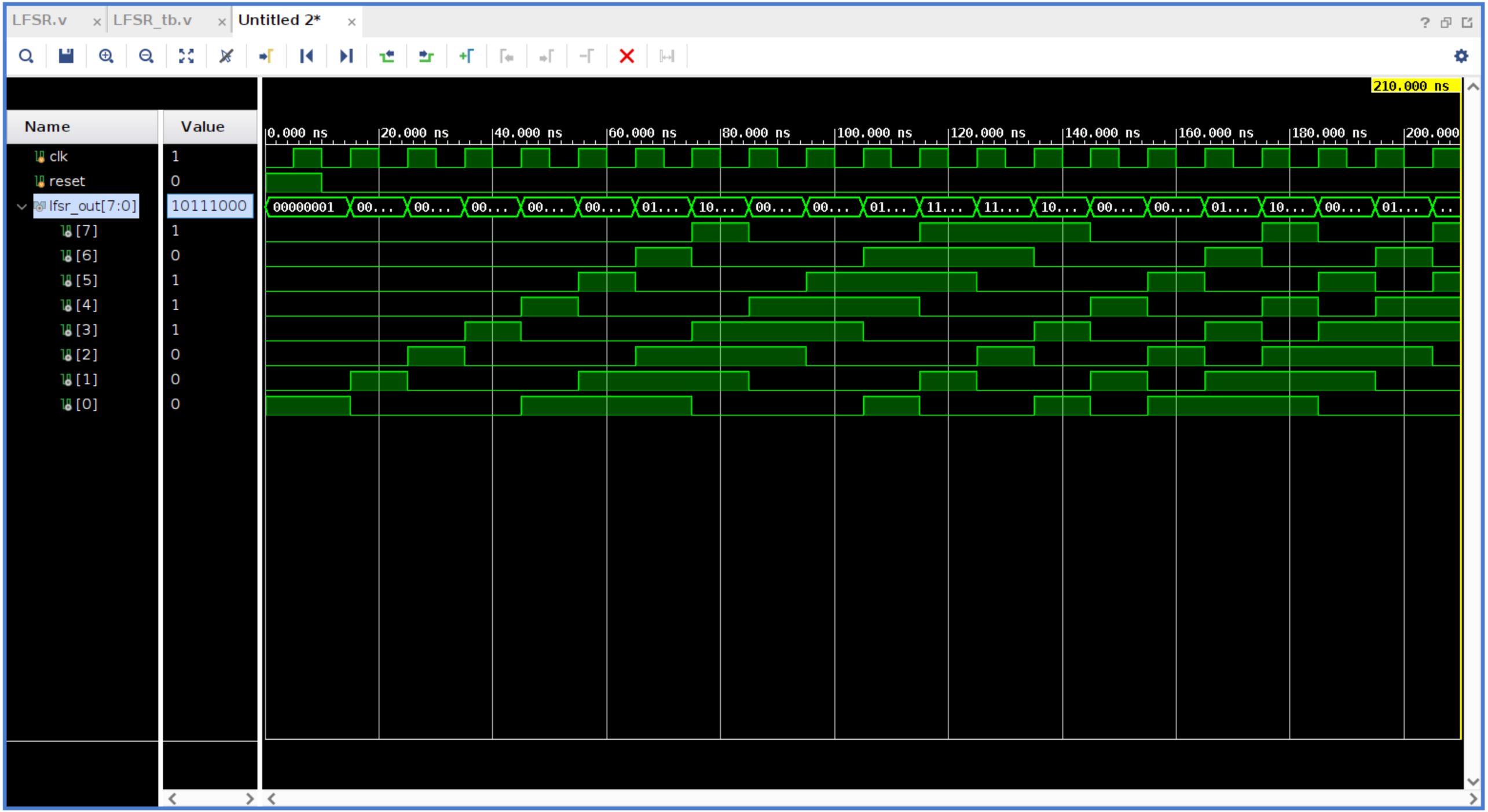
Project Summary | LFSR.v | Schematic | LFSR_tb.v | Schematic (2)

11 Cells 10 I/O Ports 13 Nets




LFSR.v
LFSR_tb.v
Untitled 2*
Name
Value
clk
1
reset
0
lfsr_out[7:0]
10111000
[7]
1
[6]
0
[5]
1
[4]
1
[3]
1
[2]
0
[1]
0
[0]
0
210.000 ns
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
140.000 ns
160.000 ns
180.000 ns
200.000
00000001
00...
01...
10...
11...

LFSR.v | LFSR_tb.v | Untitled 2*

/home/itzzinfinity/Cozy Drive/100daysofRTL/day_060/project_1/project_1.srcs/sources_1/new/LFSR.v

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Engineer: Anjan Prasad
4  // Create Date: 11/20/2024 08:29:10 AM
5  // Module Name: LFSR
6  //////////////////////////////////////////////////////////////////////////////////
7
8
9  module LFSR(
10     input clk, reset,
11     output reg [7:0] lfsr_out
12 );
13     wire feedback;
14
15     // Feedback polynomial taps: x^8 + x^6 + x^5 + x^4 + 1
16     assign feedback = lfsr_out[7] ^ lfsr_out[5] ^ lfsr_out[4] ^ lfsr_out[3];
17
18     always @(posedge clk or posedge reset) begin
19         if (reset)
20             lfsr_out <= 8'h1;  // Non-zero seed value
21         else
22             lfsr_out <= {lfsr_out[6:0], feedback};  // Shift left and feed back
23     end
24 endmodule
25
26
```

LFSR.v | LFSR_tb.v | Untitled 2*

/home/itzzinfinity/Cozy Drive/100daysofRTL/day_060/project_1/project_1.srcs/sim_1/new/LFSR_tb.v

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Engineer: Anjan Prasad
4  // Create Date: 11/20/2024 08:31:43 AM
5  // Module Name: LFSR_tb
6  //////////////////////////////////////////////////////////////////////////////////
7
8
9  module LFSR_tb;
10     reg clk, reset;
11     wire [7:0] lfsr_out;
12
13     // Instantiate the LFSR module
14     LFSR DUT(.clk(clk), .reset(reset), .lfsr_out(lfsr_out));
15
16     // Clock generation
17     initial begin
18         clk = 0;
19         forever #5 clk = ~clk;  // 10 ns clock period
20     end
21
22     // Test stimulus
23     initial begin
24         reset = 1;  // Apply reset
25         #10 reset = 0;  // Release reset
26
27         // Observe the output over several clock cycles
28         #200;
29         $stop;
30     end
31
32     // Monitor for debugging
33     initial begin
34         $monitor("Time: %0dns, LFSR Output: %b", $time, lfsr_out);
35     end
36 endmodule
37
38
```