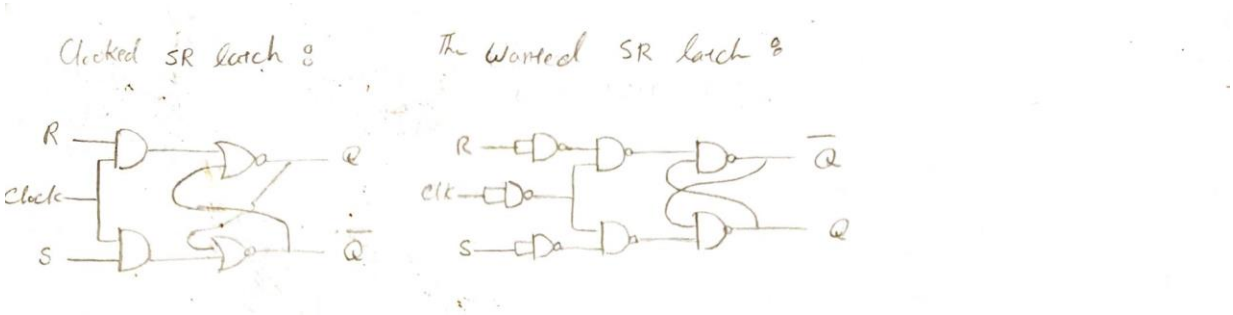
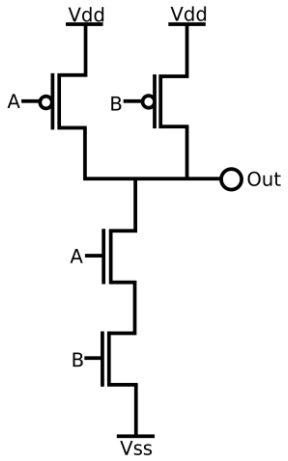
Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378
Question 1





Delays:

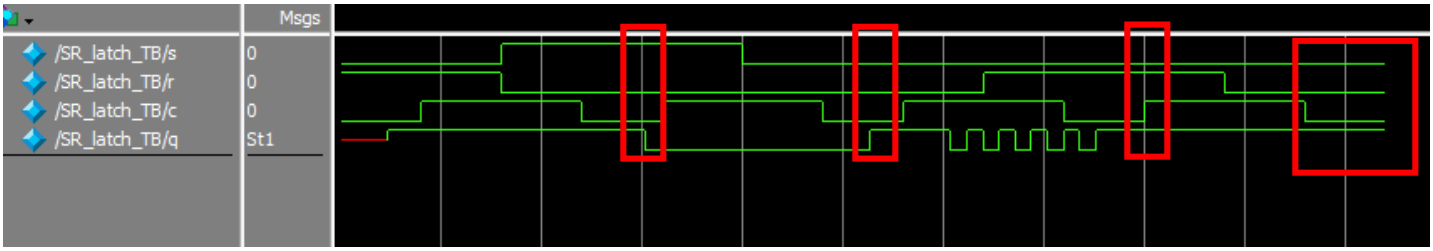
- Nmos: 4
- Pmos: 6
- Nand: 8
- SR latch: 32 (4 times of nand gate)

```
Ln#
 1   `timescale 1ns/1ns
 2
 3   module SR_latch(input S, R, Clk, output Q);
 4
 5           wire Rbar, Clkbar, Sbar, y1, y2, Qbar;
 6
 7
 8           nand #8 g1(Rbar, R, R), g2(Clkbar, Clk, Clk), g3(Sbar, S, S);
 9           nand #8 g4(y1, Rbar, Clkbar), g5(y2, Sbar, Clkbar);
10           nand #8 g6(Q, Qbar, y2), g7(Qbar, Q, y1);
11
12
13   endmodule
14
```

Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378

Question 2

```
Ln#
 1   `timescale 1ns/1ns
 2
 3   module SR_latch_TB();
 4
 5           logic s=0, r=1, c=0;
 6           wire q;
 7
 8           SR_latch l(s, r, c, q);
 9
10           initial begin
11           #40 c=1;
12           #40 s=1; r=0;
13           #40 c=0;
14           #40 c=1;
15           #40 s=0; r=0;
16           #40 c=0;
17           #40 c=1;
18           #40 s=0; r=1;
19           #40 c=0;
20           #40 c=1;
21           #40 s=0; r=0;
22           #40 c=0;
23           #40 $stop;
24           end
25
26   endmodule
27
```



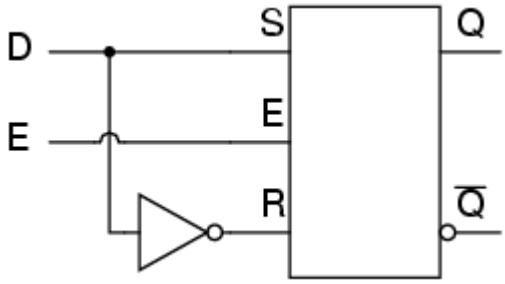
- First two rectangles: After turning on the active low input clock, the output drops to zero as a result of being reset. In the second rectangle we see after assigning 0 to S and R, and enabling the clock the output has changed to 1.
- Second two rectangles: In here the process of giving 0 value to S and R has been repeated but this time the output remains the same

From the two above conditions we understand that giving 0 value to both R and S simultaneously results in memory loss.

Digital Systems 1 – Computer Assignment 4

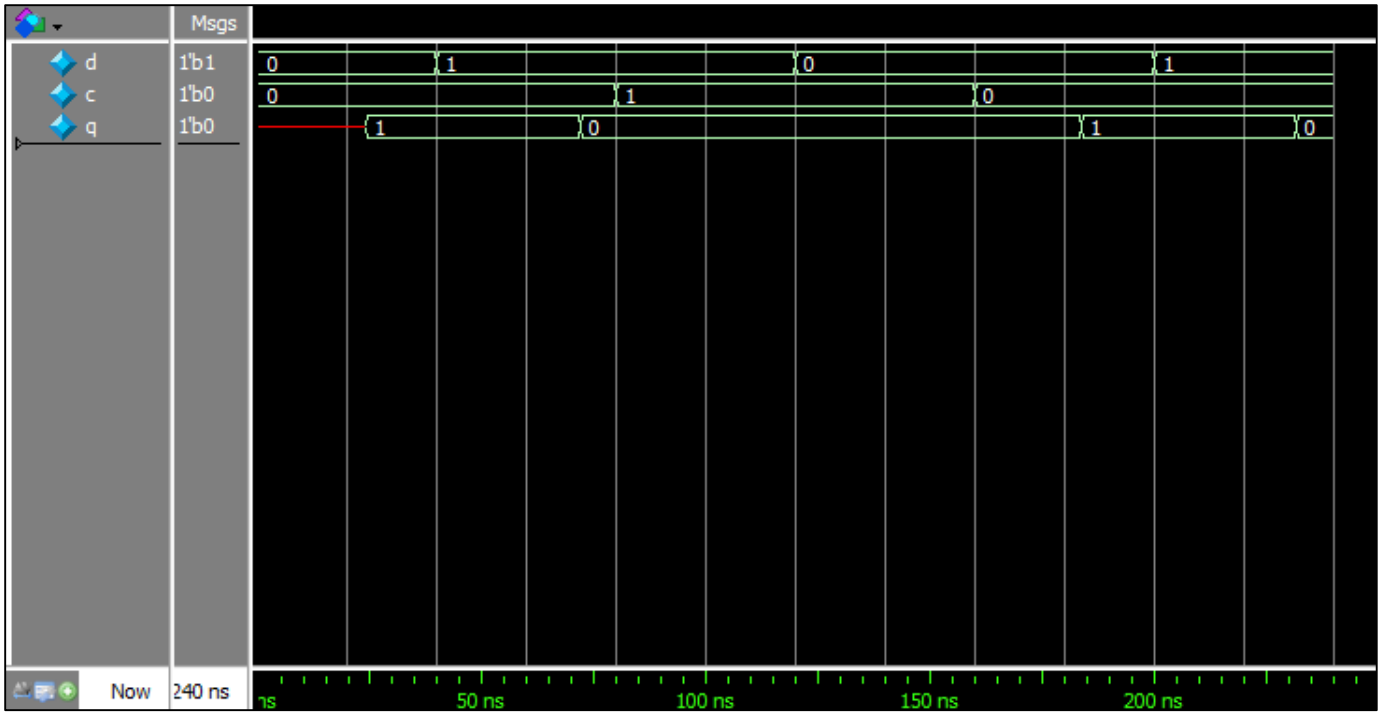
Samira Hajizadeh - 810198378

Question 3



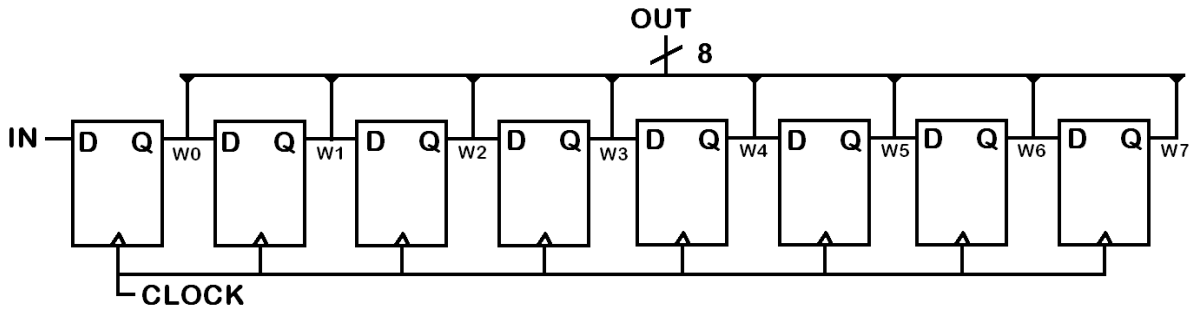

```
Ln#
 1   `timescale 1ns/1ns
 2
 3   module D_latch(input D, Clk, output Q);
 4
 5           wire dbar;
 6           not (dbar, D);
 7           SR_latch 1(D, dbar, Clk, Q);
 8
 9   endmodule
10
11
12   module D_latch_TB();
13
14          logic d=0;
15          logic c=0;
16          wire q;
17          D_latch 1(d, c, q);
18
19          initial begin
20          #40 d=1;
21          #40 c=1;
22          #40 d=0;
23          #40 c=0;
24          #40 d=1;
25          #40 $stop;
26          end
27
28  endmodule
```

Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378



As shown in the above picture, the circuit works as expected.

Question 4

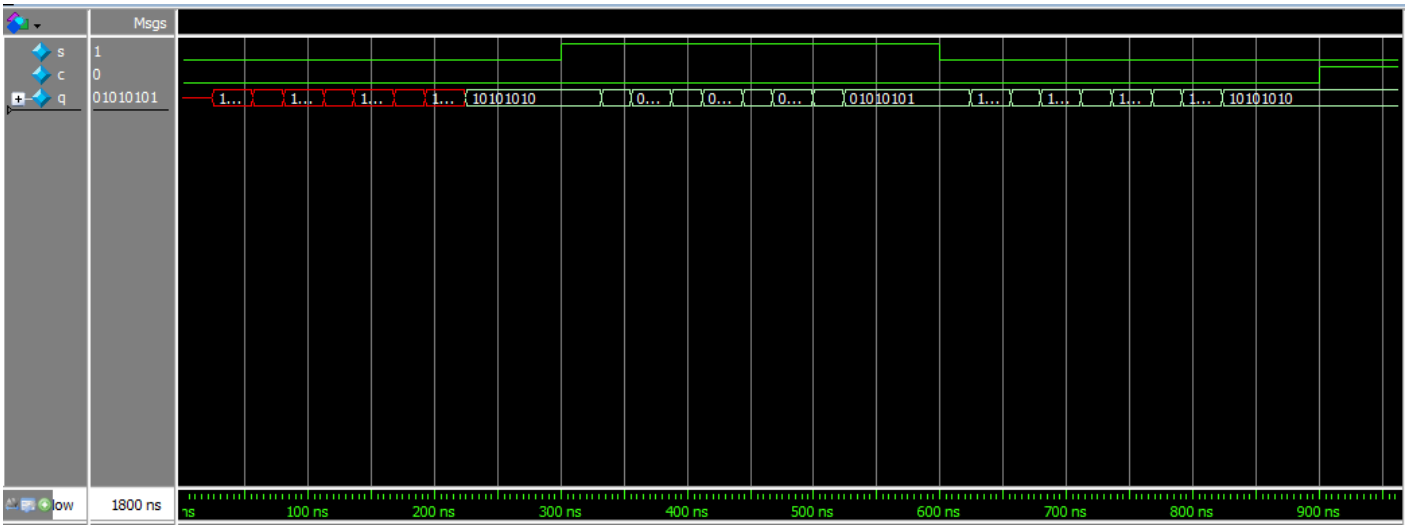


Question 5

Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378

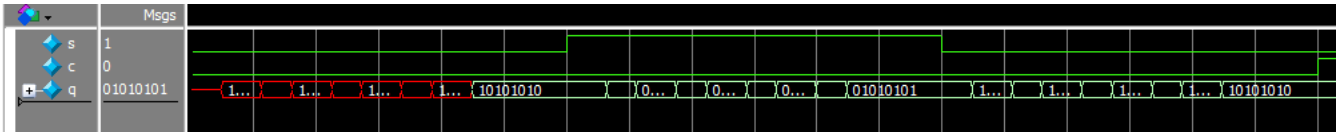
```
Ln#
 1  `timescale 1ns/1ns
 2
 3  module ShiftRegister8bit(input SID, Clk, output [7:0] Q);
 4
 5          wire [7:0] y;
 6          assign y[7] = SID;
 7          genvar i;
 8          generate
 9          for (i=7; i >= 0; i=i-1) begin
10                  D_latch l(y[i], Clk, Q[i]);
11                  if (i > 0) assign y[i-1] = Q[i];
12          end
13          endgenerate
14
15 endmodule
16
17 module SR_8bit_TB();
18
19          logic s=0, c=0;
20          wire [7:0] q;
21
22          ShiftRegister8bit SR(s, c, q);
23
24          initial begin
25          #300 s=1;
26          #300 s=0;
27          #300 c=1;
28          #300 s=1;
29          #300 c=0;
30          #300 $stop;
31          end
32
33 endmodule
```

Waveform:



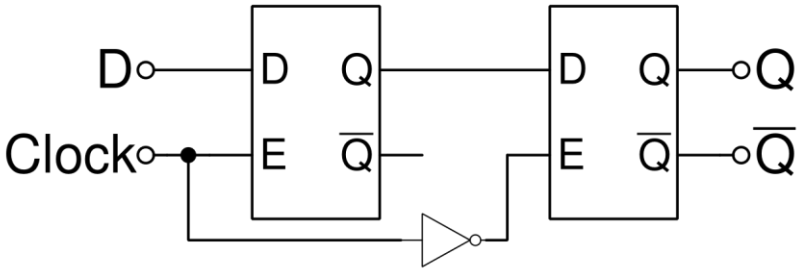
Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378

A part of the waveform:



Explanation: There is no barrier between each input and the one next to it therefore the SID effects all bits simultaneously.

Question 6




```
Ln#
1   `timescale 1ns/1ns
2
3   module D_mmff(input D, Clk, output Q);
4
5           wire y;
6           D_latch l1(D, Clk, y), l2(y, ~Clk, Q);
7
8   endmodule
9
```

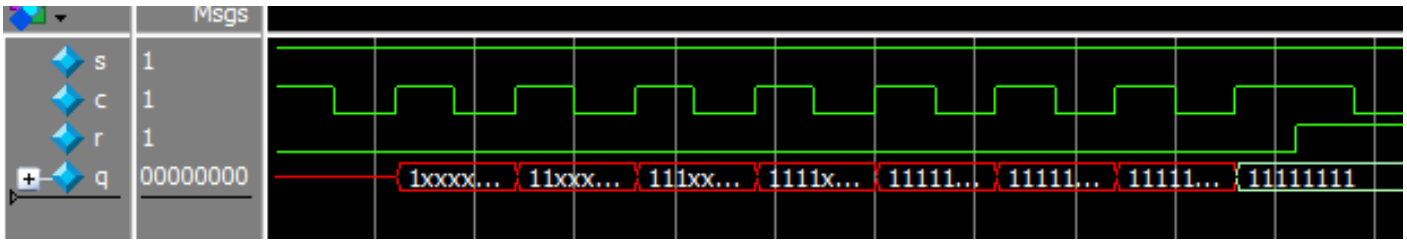
Question 7

```
Ln#
1   `timescale 1ns/1ns
2
3   module D_mmff_rs(input D, RS, Clk, output Q);
4
5           D_mmff d(D|RS, Clk, Q);
6   endmodule
```

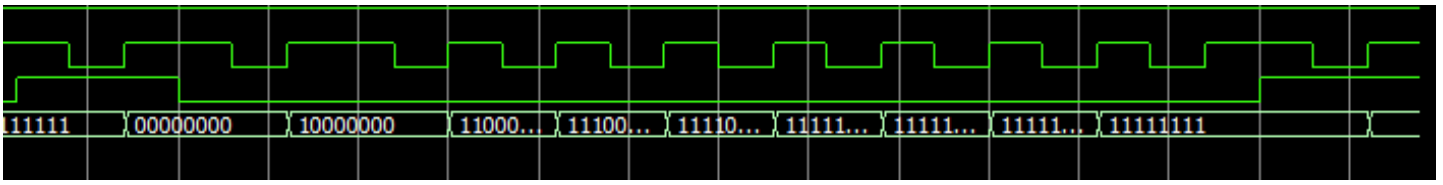
Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378
Question 8

```
Ln#
 1  `timescale 1ns/1ns
 2
 3  module ShiftRegister8bit_2(input SID, RS, Clk, output [7:0] Q);
 4
 5          wire [7:0] y;
 6          assign y[7] = SID;
 7          genvar i;
 8          generate
 9          for (i=7; i >= 0; i=i-1) begin
10                  D_mmff_rs 1(y[i], RS, Clk, Q[i]);
11                  if (i > 0) assign y[i-1] = Q[i];
12          end
13          endgenerate
14
15  endmodule
16
17  module SR_8bit_2_TB();
18
19          logic s=1, c=1, r=0;
20          wire [7:0] q;
21
22          ShiftRegister8bit_2 SR(s, r, c, q);
23
24          initial begin
25          repeat (16) #600 c=~c;
26          #600 r=1;
27          repeat (2) #600 c=~c;
28          #600 r=0;
29          repeat (2) #600 c=~c;
30          #600 s=1;
31          repeat (16) #600 c=~c;
32          #600 r=1;
33          repeat (2) #600 c=~c;
34          #600 $stop;
35          end
36
37  endmodule
```

Waveform:



Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378



Explanation: In spite of the circuit of problem 4, this shift register has the ability to be initially set to zero. In problem 4, this initialization was not approachable and as a result the shifter did not work well.

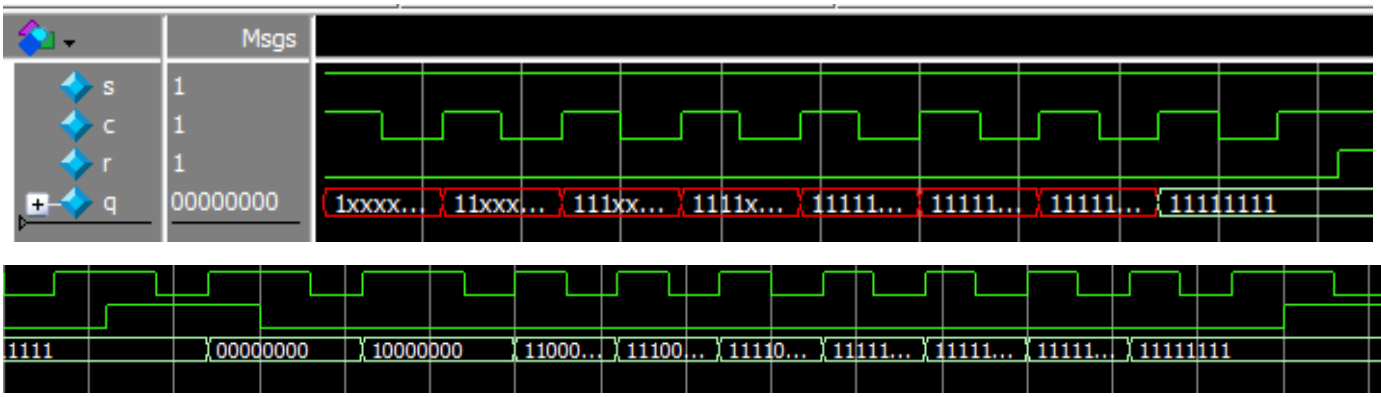
Question 9

```
Ln#
 1   `timescale 1ns/1ns
 2
 3   module ShiftRegister8bit_3(input SID, RS, Clk, output logic [7:0] Q);
 4
 5           always @(posedge Clk) begin
 6                   if (RS) assign Q = 7'b0;
 7                   else begin
 8                           assign Q = {SID, Q[7:1]};
 9                   end
10           end
11
12   endmodule
13
14   module SR_8bit_3_TB();
15
16           logic s=1, c=1, r=0;
17           wire [7:0] q;
18
19           ShiftRegister8bit_3 SR(s, r, c, q);
20
21           initial begin
22           repeat (16) #600 c=~c;
23           #600 r=1;
24           repeat (2) #600 c=~c;
25           #600 r=0;
26           repeat (2) #600 c=~c;
27           #600 s=1;
28           repeat (16) #600 c=~c;
29           #600 r=1;
30           repeat (2) #600 c=~c;
31           #600 $stop;
32           end
33
34   endmodule
35
```

Waveform:

Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378

Waveform:



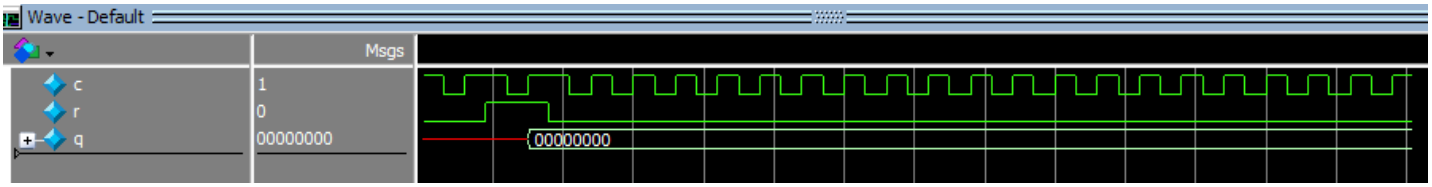
As seen above, the waveform is totally equal to the one of question 8.

Question 10

```
Ln#
 1  `timescale 1ns/1ns
 2
 3  module LFSR(input RS, Clk, output logic [7:0] Q);
 4
 5          ShiftRegister8bit_3 sr(SID, RS, Clk, Q);
 6          assign SID = Q[7]^Q[6]^Q[3]^Q[0];
 7
 8  endmodule
 9
10  module LFSR_TB();
11
12          logic c=1, r=0;
13          wire [7:0] q;
14
15          LFSR SR(r, c, q);
16
17          initial begin
18
19          repeat (2) #600 c=~c;
20          #600 r=1;
21          repeat (2) #600 c=~c;
22          #600 r=0;
23
24          repeat (40) #600 c=~c;
25          #600 $stop;
26          end
27
28  endmodule
29
```

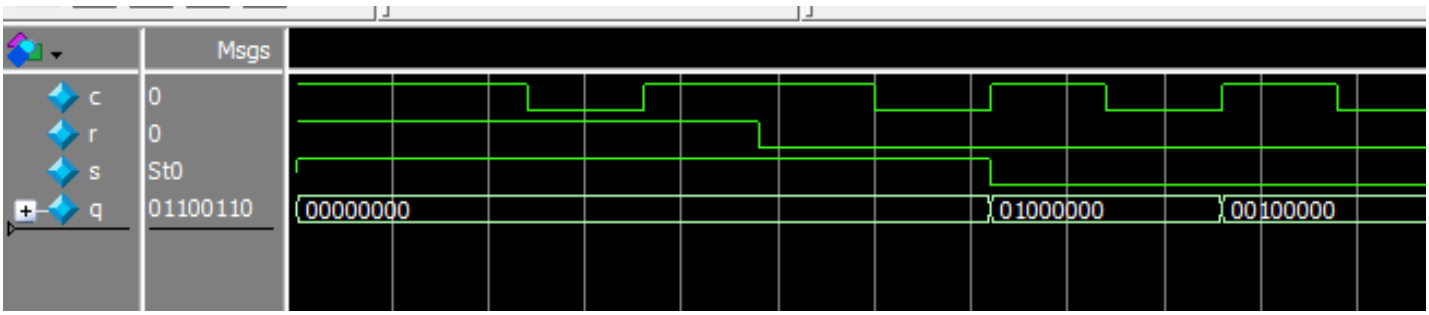
Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378

With the assumptions of the question, the output is always equal to zero.



The corrected circuit

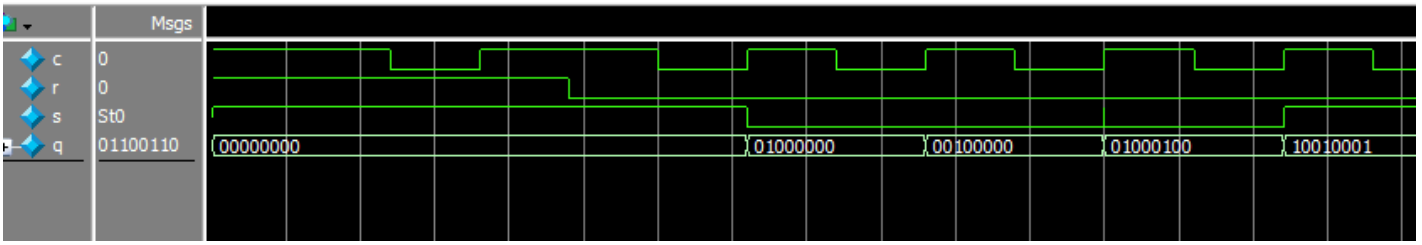
```
Ln#
1   `timescale 1ns/1ns
2
3   module LFSR(input SID, RS, Clk, output logic [7:0] Q);
4           xor g(SID, Q[7], Q[6], Q[5], Q[1], Q[0], 1'b1);
5           ShiftRegister8bit_3 sr(SID, RS, Clk, Q);
6
7   endmodule
8
9   module LFSR_TB();
10
11          logic c=1, r=1;
12          wire s;
13          wire [7:0] q;
14
15          LFSR SR(s, r, c, q);
16
17          initial begin
18          #600 r=1; c=1;
19          repeat (2) #600 c=~c;
20          #600 r=0;
21          repeat (50) #600 c=~c;
22          #600 $stop;
23          end
24
25  endmodule
26
```

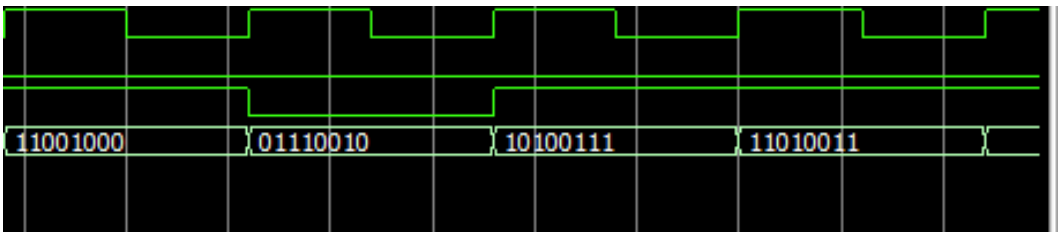


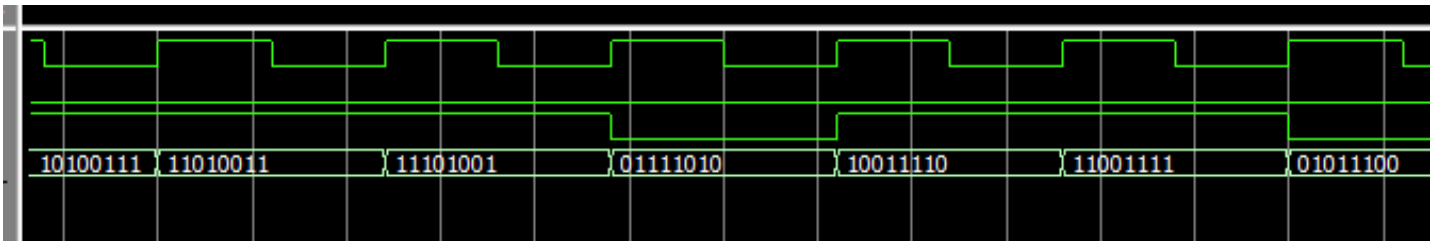
Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378

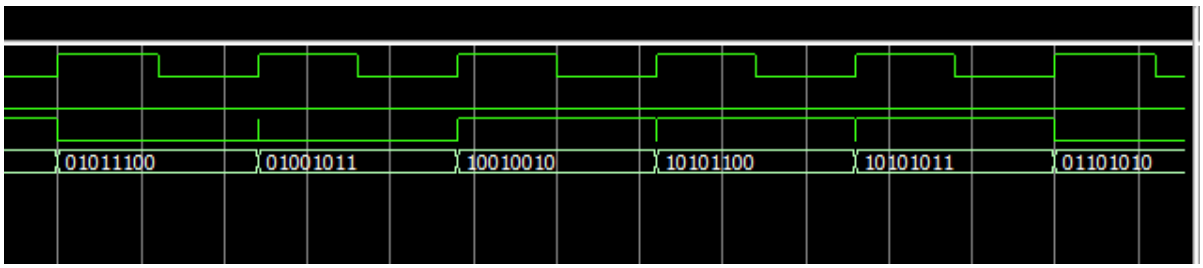
00000000 -> 01000000 -> 00100000 -> 01000100 -> 10010001 -> 11001000 - > 01110010 -> 10100111

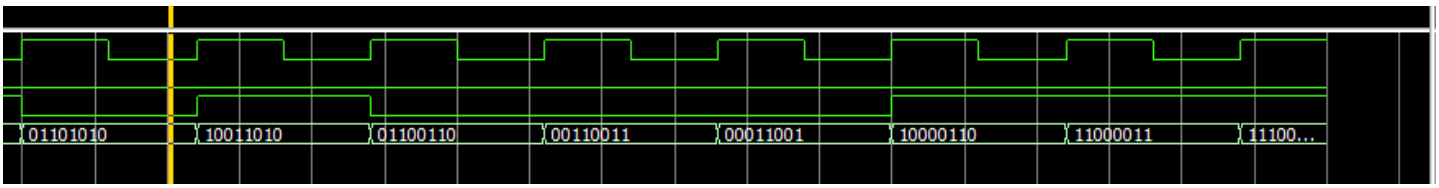
11010011 -> 11101001 -> 01111010 -> 10011110 -> 11001111 -> 01011100 -> 01011100 -> 01001011 -> 10010010 -> 10101100 -> 10101011 -> 01101010 -> 10011010 -> 01100110 -> 00110011 -> 00011001 ->10000110 -> 11000011 -> 11100001 -> 10111100 -> 1010111 -> 11011011 -> 11101101 -> 11110110 -> 01111101 -> 01001111 -> 00100111 -> 10001001 ->11000100



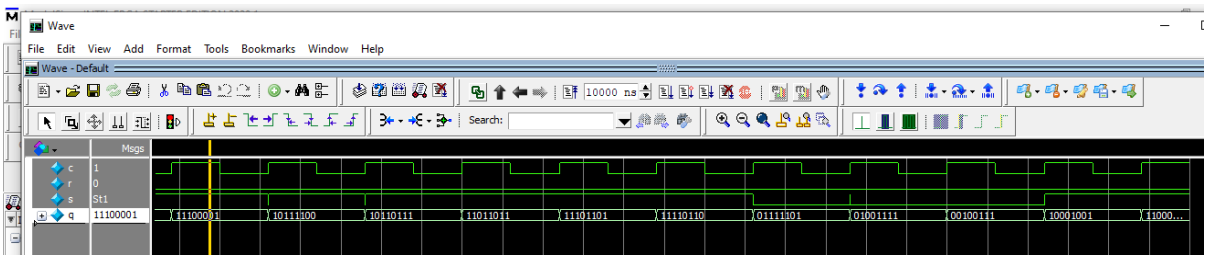








Digital Systems 1 – Computer Assignment 4
Samira Hajizadeh - 810198378



The period of this circuit is long enough to be used as a random code producer with a particular seed.