DIGITAL DESIGN
LAB 8

DIVYAM PATEL

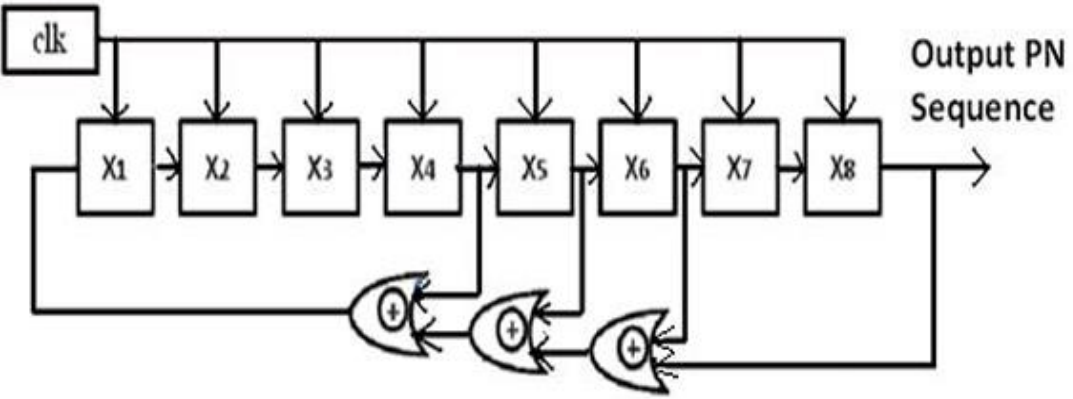
B20EE082

1. Create the 8bit MLS counter using LFSR.

CODE

```
module mls_counter(A,reset,clk,O);
input [1:8] A;
input reset,clk;
output O;
reg [1:8] X;
wire V;

always @(posedge reset)
begin
    X[1]<=A[1];
    X[2]<=A[2];
    X[3]<=A[3];
    X[4]<=A[4];
    X[5]<=A[5];
    X[6]<=A[6];
    X[7]<=A[7];
    X[8]<=A[8];
end

always @(posedge clk)
begin
    X[1]<=V;
    X[2]<=X[1];
    X[3]<=X[2];
    X[4]<=X[3];
    X[5]<=X[4];
    X[6]<=X[5];
    X[7]<=X[6];
    X[8]<=X[7];
end

assign V=X[8]^X[6]^X[5]^X[4];
assign O=X[8];
endmodule
```

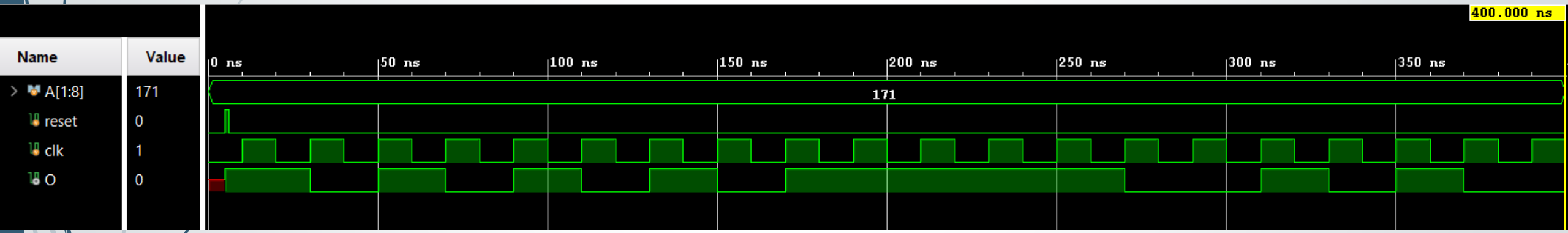
8 BIT LFSR



TESTBENCH

```
module test_mls();
reg [1:8] A;
reg reset,clk;
wire O;
mls_counter MLS(A,reset,clk,O);

initial
begin
    A=8'b10101011; reset=0; clk=0;
    #5 reset=1; #1 reset=0; #4 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
    #10 clk=0; #10 clk=1;
end
initial #400 $finish;
endmodule
```

SIMULATION


Name
Value
A[1:8]
171
reset
0
clk
1
O
0
400.000 ns
0 ns
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
350 ns
171

2. Write a program to implement 2’s complement of a running bitstream using

a) Melay Machine

b) Moore Machine

Use LFSR for generating input bitstream.

MELAY MACHINE

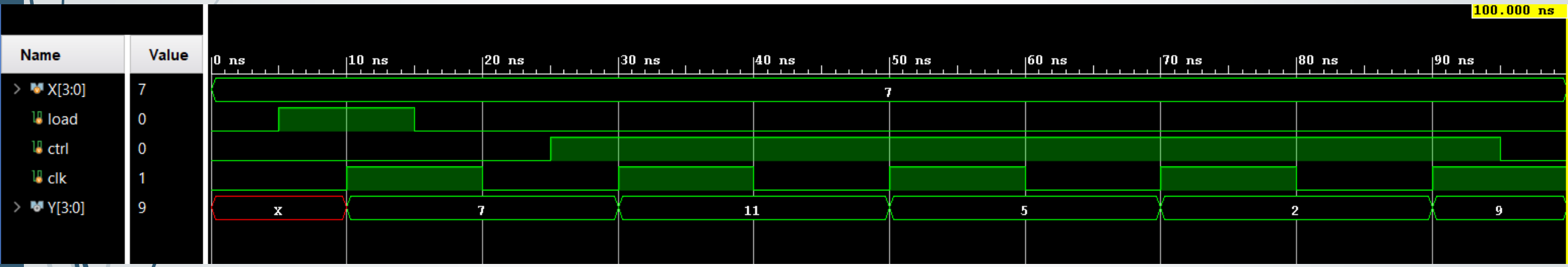
CODE

```
module melay(X,load,ctrl,clk,Y);
input [3:0]X;
input load,ctrl,clk;
output reg [3:0] Y;
wire D,V;
reg A=0;

always @(posedge clk)
begin
    if(load)
    begin
        Y[0]=X[0];
        Y[1]=X[1];
        Y[2]=X[2];
        Y[3]=X[3];
    end
    if(ctrl)
    begin
        A<=D;
        Y[0]<=Y[1];
        Y[1]<=Y[2];
        Y[2]<=Y[3];
        Y[3]<=V;
    end
end

assign D= A | Y[0];
assign V= Y[0]^A;
endmodule
```

TESTBENCH

```
module test_melay();
reg [3:0] X;
reg load,ctrl,clk;
wire [3:0] Y;
melay me(X,load,ctrl,clk,Y);

initial
begin
    X=4'b0111; load =0; ctrl=0; clk=0;
    #5 load=1;
    #5 clk=1; #5 load=0;
    #5 clk=0; #5 ctrl=1;
    #5 clk=1;
    #10 clk=0;
    #10 clk=1;
    #10 clk=0;
    #10 clk=1;
    #10 clk=0;
    #10 clk=1; #5 ctrl=0;
end
initial #100 $finish;
endmodule
```

SIMULATION


Name
Value
X[3:0]
7
load
0
ctrl
0
clk
1
Y[3:0]
9
100.000 ns
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
7
x
7
11
5
2
9

MOORE MACHINE

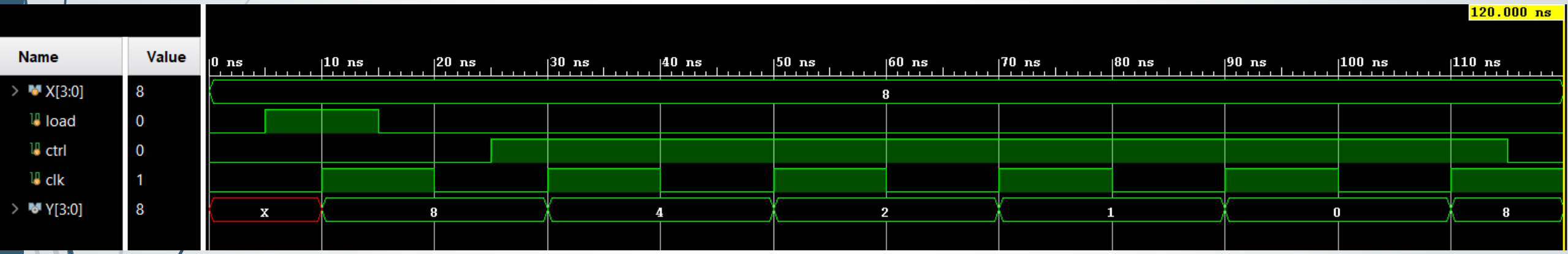
CODE

```
module moore(X,load,ctrl,clk,Y);
input [3:0]X;
input load,ctrl,clk;
output reg [3:0] Y;
wire D1,D2;
reg A=0,B=0;

always @(posedge clk)
begin
    if(load)
    begin
        Y[0]=X[0];
        Y[1]=X[1];
        Y[2]=X[2];
        Y[3]=X[3];
    end

    if(ctrl)
    begin
        A<=D1;
        B<=D2;
        Y[0]<=Y[1];
        Y[1]<=Y[2];
        Y[2]<=Y[3];
        Y[3]<=B;
    end
end

assign D1=Y[0] & (A | B);
assign D2=Y[0] ^ (A | B);
endmodule
```

TESTBENCH

```
module test_moore();
reg [3:0] X;
reg load,ctrl,clk;
wire [3:0] Y;
moore mo(X,load,ctrl,clk,Y);

initial
begin
    X=4'd8; load =0; ctrl=0; clk=0;
    #5 load=1;
    #5 clk=1; #5 load=0;
    #5 clk=0; #5 ctrl=1;
    #5 clk=1;
    #10 clk=0;
    #10 clk=1;
    #10 clk=0;
    #10 clk=1;
    #10 clk=0;
    #10 clk=1;
    #10 clk=0;
    #10 clk=1; #5 ctrl=0;
end
initial #120 $finish;
endmodule
```

SIMULATION


Name
Value
X[3:0]
8
load
0
ctrl
0
clk
1
Y[3:0]
8
120.000 ns
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
100 ns
110 ns
8
X
8
4
2
1
0
8

THANK YOU