**1**

AMUSEMENT PARK WATER RIDES

DSD Project Report

Submitted in partial fulfillment of the requirements for the degree of BACHELOR OF TECHNOLOGY in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

**CHIKNIS TAPAN AVINASH**

(Roll. No. 211EC111)

**GANTA YASHWANTH KUMAR**

(Roll. No. 211EC112)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE -575025

January, 2023

# Abstract

“Big waves, crazy jumps, into the blue world we traverse …” How exciting it is to spend a day in an amusement park taking the rides filled with endless surprises, twists and turns and moreover accelerating fun! But ever wondered if we can make our own water way just by implementing few electronic circuits to work together? Well, this was the inspiration for our project. We have designed a unique key to enter into our classic and deluxe water rides.

In brief, our design is composed of a sequence detector and two pattern generators using counters. There are two sequences that are being detected, one will lead us to classic ride and the other takes us to deluxe ride. Along with the rides there’s a special pattern in which water sprinklers surprise you all the way. The classic and deluxe ride comprise of twisted ring counter pattern which activates the push button that lifts the seat up and the seat sprinklers on. The water sprinklers along the path are based on combination of up-down counters used to start every sprinkler at a specified time.

So, hope this ride gets you excited as the ride also waits for you.

|  |  |  |
| --- | --- | --- |
|  | **Table of Contents** |  |
| **Content** |  | **Page number** |
| 1. Introduction |  | 4 |
| 2. Design |  | 5 |
| 3. Implementation |  | 7 |
| 4. Results and discussions |  | 14 |
| 5. Conclusions |  | 14 |
| 6. References |  | 14 |

# Introduction

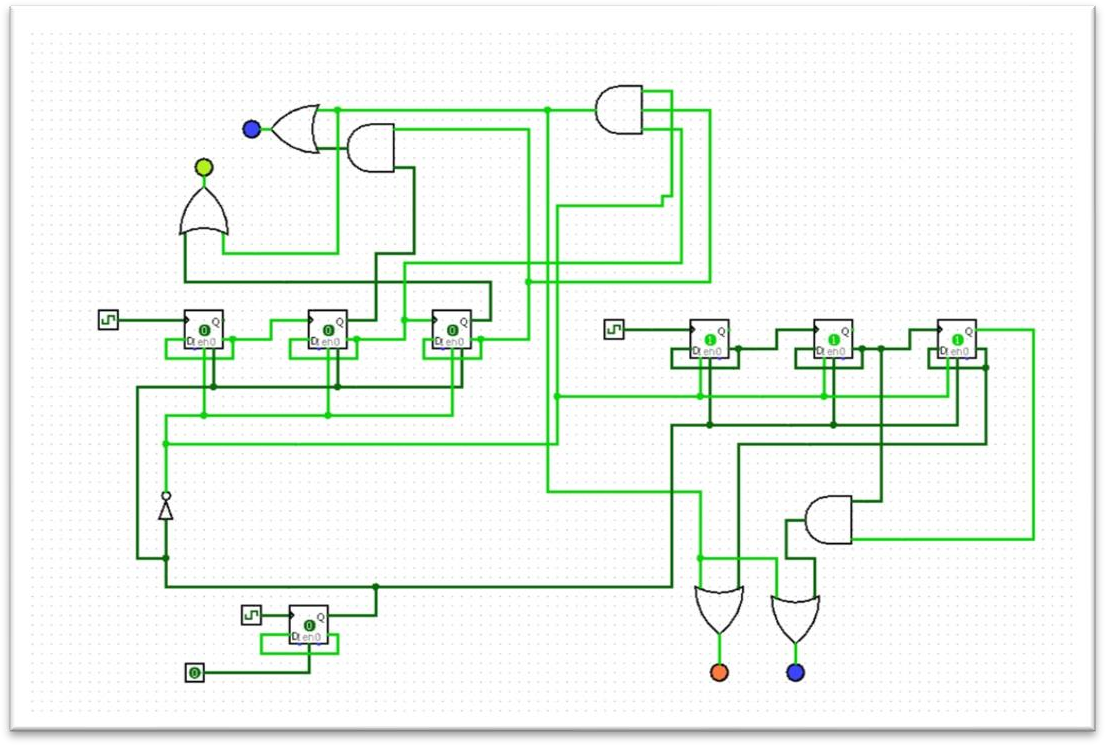
“Amusement Park water rides” project comprises of a sequence detector followed by two pattern generators. So, we begin with the tourist, who shall have to try his/her luck by solving the sequence puzzle. There are two sequences 101 and 111 that are being detected by this FSM. If the tourist enters 101, he/she shall be taken to the Deluxe ride and if 111 is entered, classic ride will be opened for him/her.

The deluxe ride uniqueness lies in the pattern in which the seats get lifted up. Basically, we have used Johnson counter pattern to lift the seats. So, for a three bit output circuit, all the bits with logic high will be interconnected to the seats and all the output pins with logic low will be connected to active low water sprinklers over the seat. This same design is followed in classic ride except for the chairs.

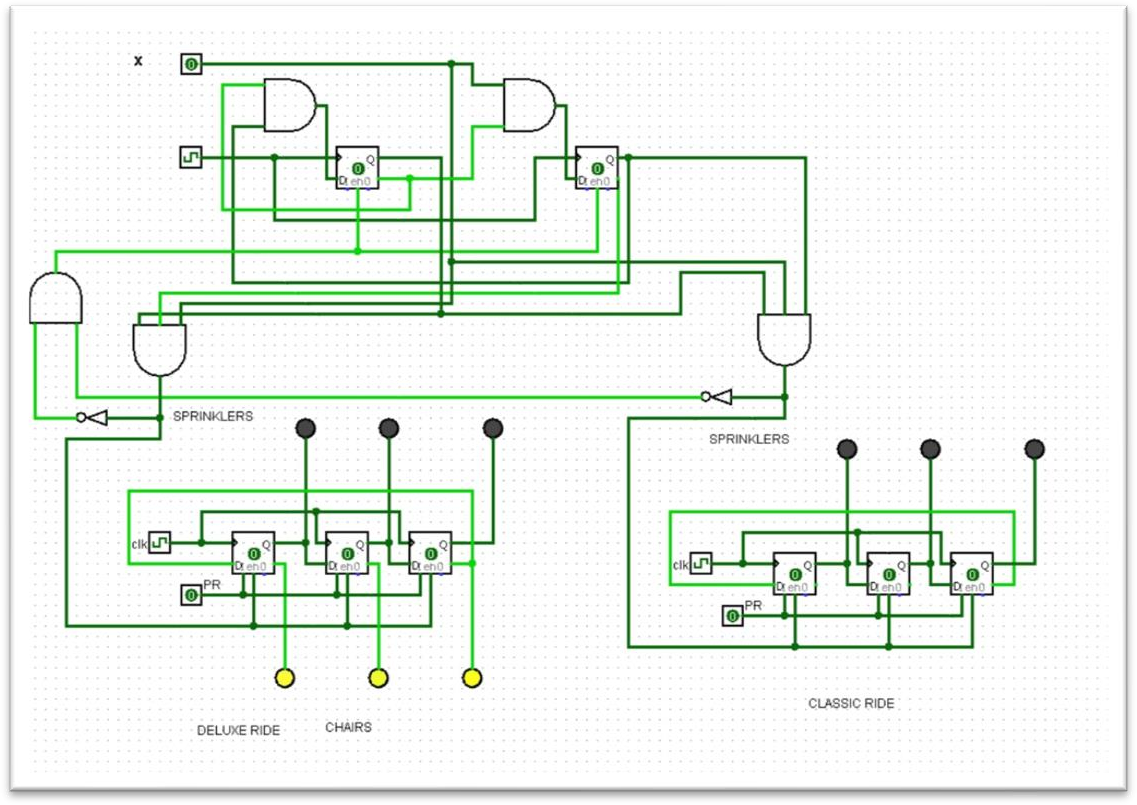
Now, apart from the seat patterns, we have water sprinklers along the two borders of the path. These water sprinklers are grouped in four and placed at equal distances. So, once the ride begins, the first sprinkler will be on for eight seconds followed by the next on for four seconds. After this all four sprinklers will be on for eight seconds and later the third sprinkler stays on for four seconds. This is followed by the last sprinkler on for eight seconds. As, the sprinklers are placed alternately on the two sides of path, we get a very beautiful zig-zag pattern.

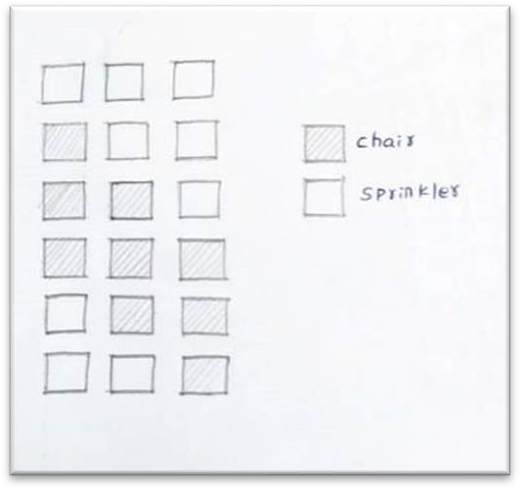
# Design

* Sideways sprinkler pattern using flip- flops:

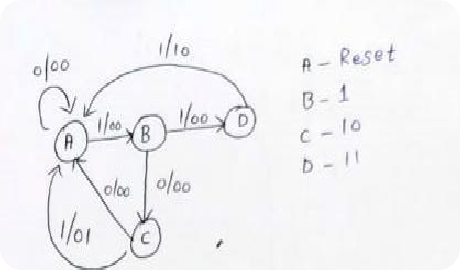
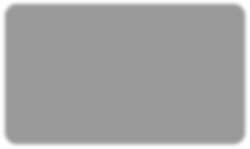


* Sequence Detector, twisted ring counter design:



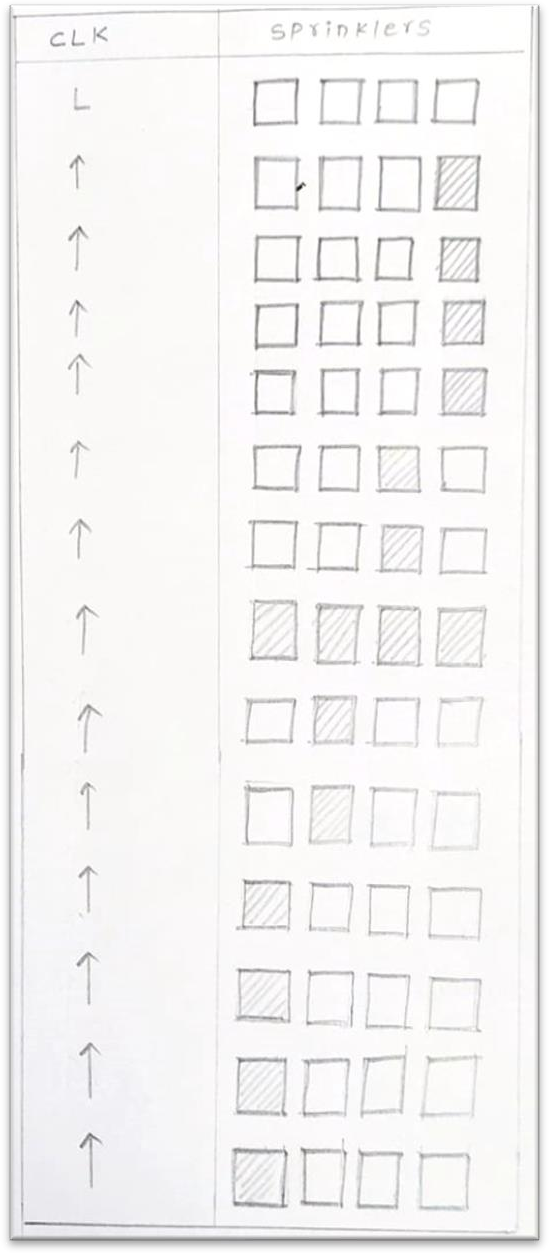


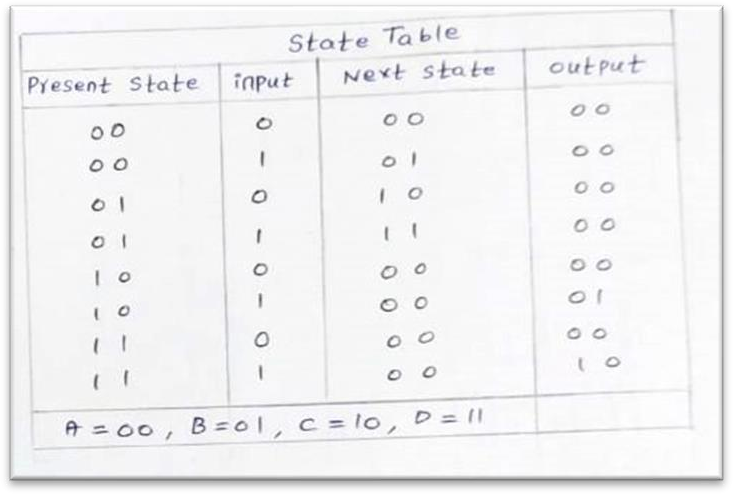
**Deluxe Ride Pattern**



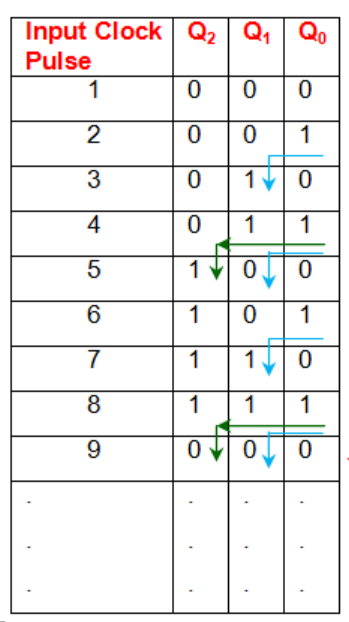
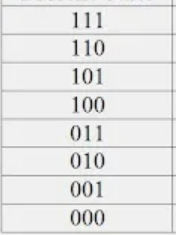
**State Diagram**

<=For 4 clocks

**Sprinklers Pattern**



**State Table**

****

**UP COUNTER DOWN COUNTER**

# Implementation

* Sequence Detector, twisted ring counter design:

timescale 1ns / 1ps

module Theme\_park\_enable(input x,clk,reset,output [1:0]Y,[2:0]Q,[2:0]Qb,[2:0]Q1);

reg [1:0]cs,nxs;

reg [1:0]S;

parameter A='b00;

parameter B='b01;

parameter C='b10;

parameter D='b11;

assign t=0;

always@(posedge(clk),reset)begin

if(reset==1)

cs<=A;

else

cs<=nxs;

end

always@(cs,x)begin

if(S=='b01 || S=='b10)begin

#130;

S<='b00;

end

case (cs)

A: begin

if(x==0) begin

nxs<=A;

S<='b00;

end

else begin

nxs<=B;

S<='b00;

end

end

B: begin

if(x==0) begin

nxs<=C;

S<='b00;

end

else begin

nxs<=D;

S<='b00;

end

end

C: begin

if(x==0) begin

nxs<=A;

S<='b00;

end

else begin

nxs<=A;

S<='b01;

$display("Be ready for deluxe ride");

end

end

D: begin

if(x==0) begin

nxs<=A;

S<='b00;

end

else begin

nxs<=A;

S<='b10;

$display("Be ready for classic ride");

end

end

endcase

end

assign Y=S;

Theme\_park\_chair d1(S[0],clk,Q[2:0],Qb[2:0]);

Theme\_park\_sprinklers d2(S[1],clk,Q1[2:0]);

endmodule

//Deluxe ride

module Theme\_park\_chair(input e,clk,output [2:0]Q,[2:0]Qb); reg [2:0]Y;

always@(posedge(clk),e)begin // 130ns if(e)begin

Y[2]<=~Y[0];

Y[1]<=Y[2];

Y[0]<=Y[1];

end

else begin Y<='b000;

end end

assign Q=Y; // Q stands for sprinklers assign Qb=~Y; // Qb stands for chairs endmodule

// Classic ride

module Theme\_park\_sprinklers(input e,clk,output [2:0]Q); reg [2:0]Y;

always@(posedge(clk),e)begin // 130ns if(e)begin

Y[2]<=~Y[0];

Y[1]<=Y[2];

Y[0]<=Y[1];

end

else begin Y<='b000;

end end

assign Q=Y; // Q stands for sprinklers endmodule

Test bench :

module Theme\_park\_enable\_tb(); reg x,clk,reset;

wire [1:0]Y;

wire [2:0]Q,Qb,Q1;

Theme\_park\_enable d1( x,clk,reset,Y[1:0],Q[2:0],Qb[2:0],Q1[2:0]); initial begin

clk=1;

forever #5 clk=~clk; end

initial begin

$display("Time to try your luck, dream world is waiting for you"); reset=1;

x=0; #10;

reset=0; x=1; #10;

reset=0; x=0; #10;

reset=0; x=1; #300;

reset=0; x=1; #10;

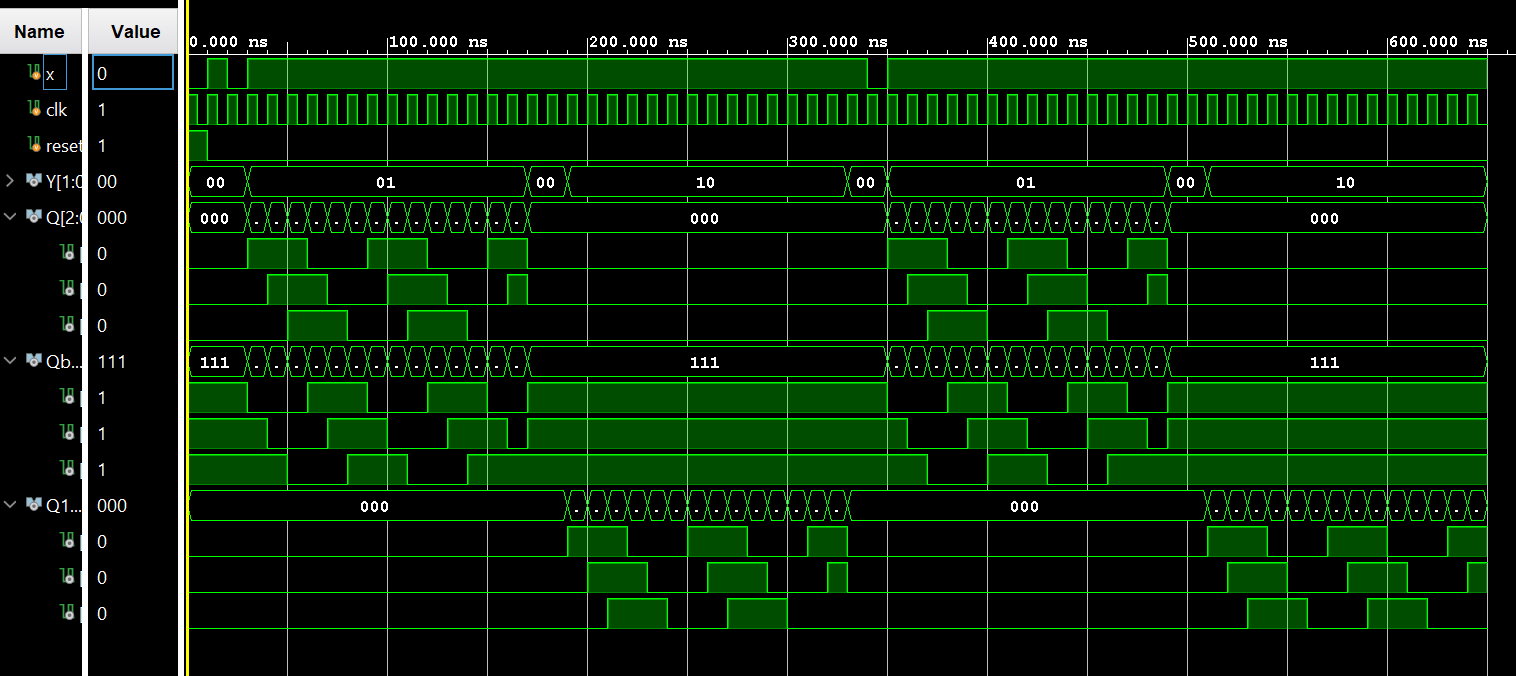
reset=0; x=0;

#10;

reset=0; x=1; #300;

$finish ; end endmodule

Output :



* Sideways sprinkler pattern using flip- flops:

module mod8up(input clk,preset,clr,output [2:0]Y); reg [2:0]Q;

always@(clk,clr,preset)begin if(clr==0)

Q<='b000;

else if(preset==0) Q<='b111;

else if(clk==1) begin Q[0]<=~Q[0];

if(Q[0]==1)begin Q[1]<=~Q[1];

if(Q[1]==1)

Q[2]<=~Q[2];

else Q[2]<=Q[2];

end else

Q[1]<=Q[1];

end end

assign Y=Q; endmodule

module mod8down(input clk,preset,clr,output [2:0]Y); reg [2:0]Q;

always@(clk,clr,preset)begin if(clr==0)

Q<='b000;

else if(preset==0) Q<='b111;

else if(clk==1) begin Q[0]<=~Q[0];

if(Q[0]==0)begin Q[1]<=~Q[1];

if(Q[1]==0)

Q[2]<=~Q[2];

else Q[2]<=Q[2];

end else

Q[1]<=Q[1];

end end

assign Y=Q; endmodule

module mod2(input clk,clr,preset,output Q); reg Y;

always@(clk,clr,preset)begin if(clr==0)

Y<='b0;

else if(preset==0) Y<='b1;

else if(clk==1) Y<=~Y;

#90;

end

assign Q=Y; endmodule

module Theme\_park\_path(input clk,clr,preset,output [3:0]Q); // Sideway sprinklers

reg [3:0]Y;

wire [2:0]Y1,Y2;

wire E;

reg clr2,preset1;

mod2 d1(clk,clr,preset,E);

mod8down d2(clk,preset1,1,Y1[2:0]);

mod8up d3(clk,1,clr2,Y2[2:0]);

always@(clk,preset,clr,E,clr2,preset1)begin

if(clr==0)begin

Y<='b0000; preset1<='b0; clr2<='b0;

end

else if(preset==0)

Y<='b1111;

else if(clk==1)begin

preset1<='b1; clr2<='b1;

if(E==0)begin

if(Y1=='b001 || Y1=='b000)

Y='b1111;

else begin

Y[0]<=Y1[2];

Y[1]<=~Y1[2] && Y1[1];

Y[2]<='b0000;

Y[3]<='b0000;

end

end

else begin

if(Y2=='b001 || Y2=='b000)

Y='b1111;

else begin

Y[2]<=~Y2[2] && Y2[1];

Y[3]<=Y2[2];

Y[1]<='b0000;

Y[0]<='b0000;

end

end

end

end

assign Q=Y;

endmodule

Test Bench :

module Theme\_park\_tb( );

reg clk,clr,preset;

wire [3:0]Y;

Theme\_park\_path d1(clk,clr,preset,Y[3:0]);

initial begin

clk=1;

forever #5 clk=~clk;

end

initial begin

clr=0;

preset=1;

#10;

clr=1;

preset=1;

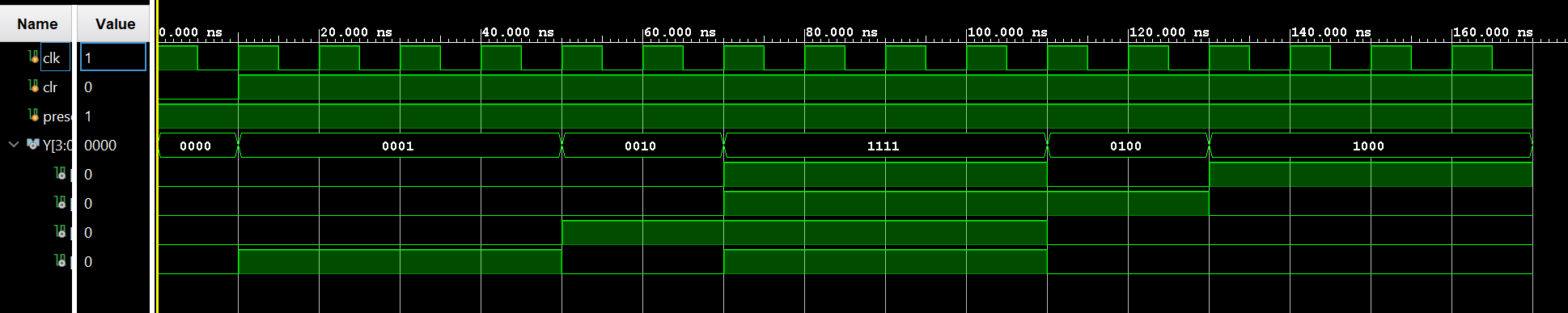
#160;

$finish ;

end

endmodule

Output waveform :



# Results and Discussions

So, describing the timeline of our project “Amusement Park water rides”, initially the topic was chosen as a result of group discussion among the group members. Secondly, the design was first checked at architectural level and later implemented at gate level in Logisim. On, completing this verification, the Verilog codes were written.

All in all, the project gave us a very good experience in understanding different parameters associated with Logisim, Verilog, any design implementation and also the direction to initiate with a project. Not only did it help us clear our concepts but also gave us the chance to work together which we may soon do as professionals.

# Conclusion

Thus, FSM and counter based design project is studied and implemented.

# References

DSD course material, fpga4student.com