

SHOOTING THE BALL

SHUBHAM GUPTA (111601024 CSE)

ROSHIN RAPHEL (111601018)

CONTRIBUTUION:

1. **ROSHIN RAPHEL** – Idea for the topic and verilog coding,hardware implimentation.
2. **SHUBHAM GUPTA**- Hardware preparation,verilog code rectification,hardware implimentation.

ABSTRACT :

THE game is basically a shooting game.

As its name suggested that its a ball shooting game,there is a shooter at the bottom which can shoot the ball.

There is a 4x4 matrix for the ball and a 1x1 matrix for shooter,shooter can move from one column to other column by left and right switch button.

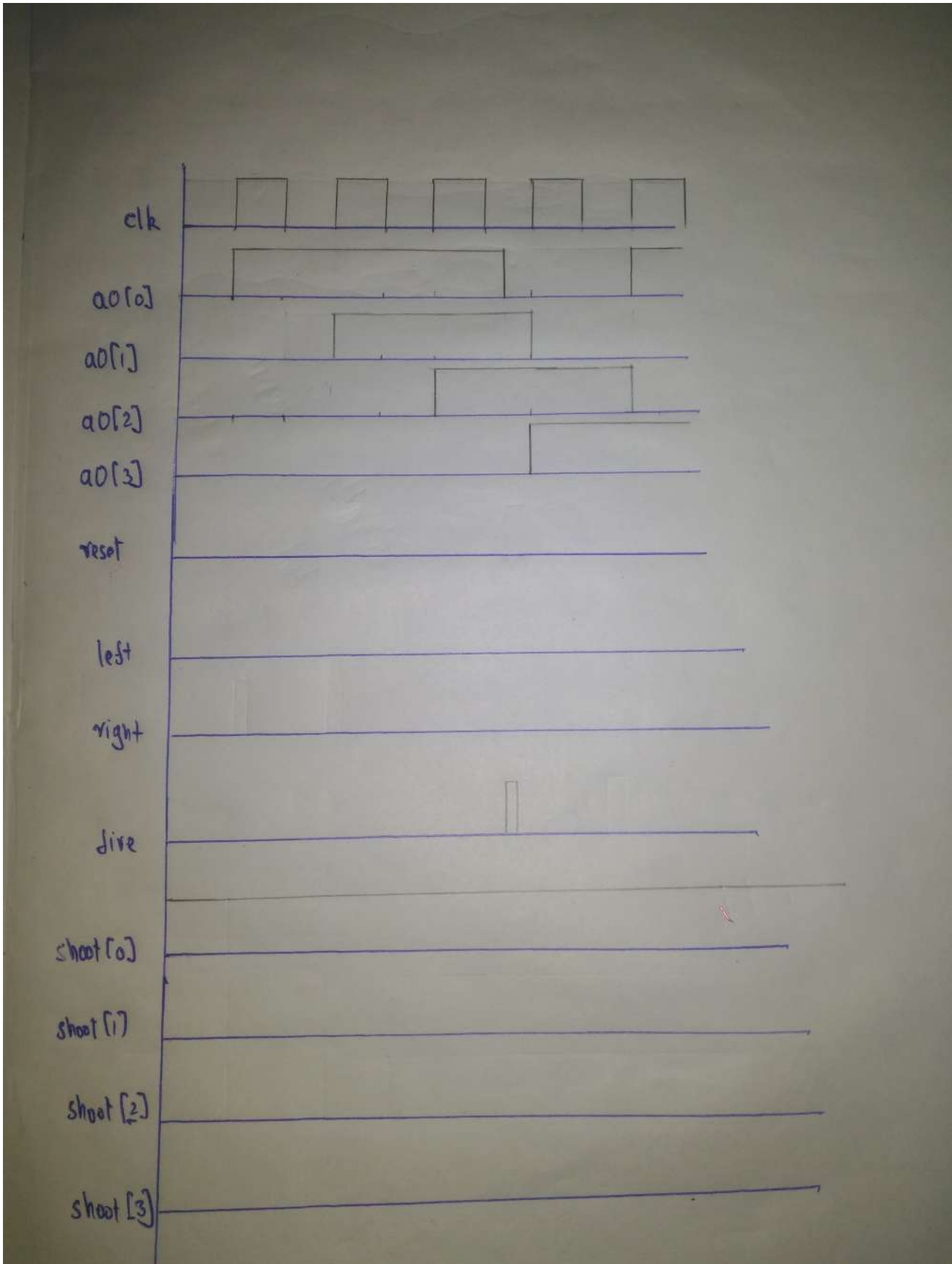
There are 4 switch buttons LEFT,RIGHT,SHOOT,RESET.

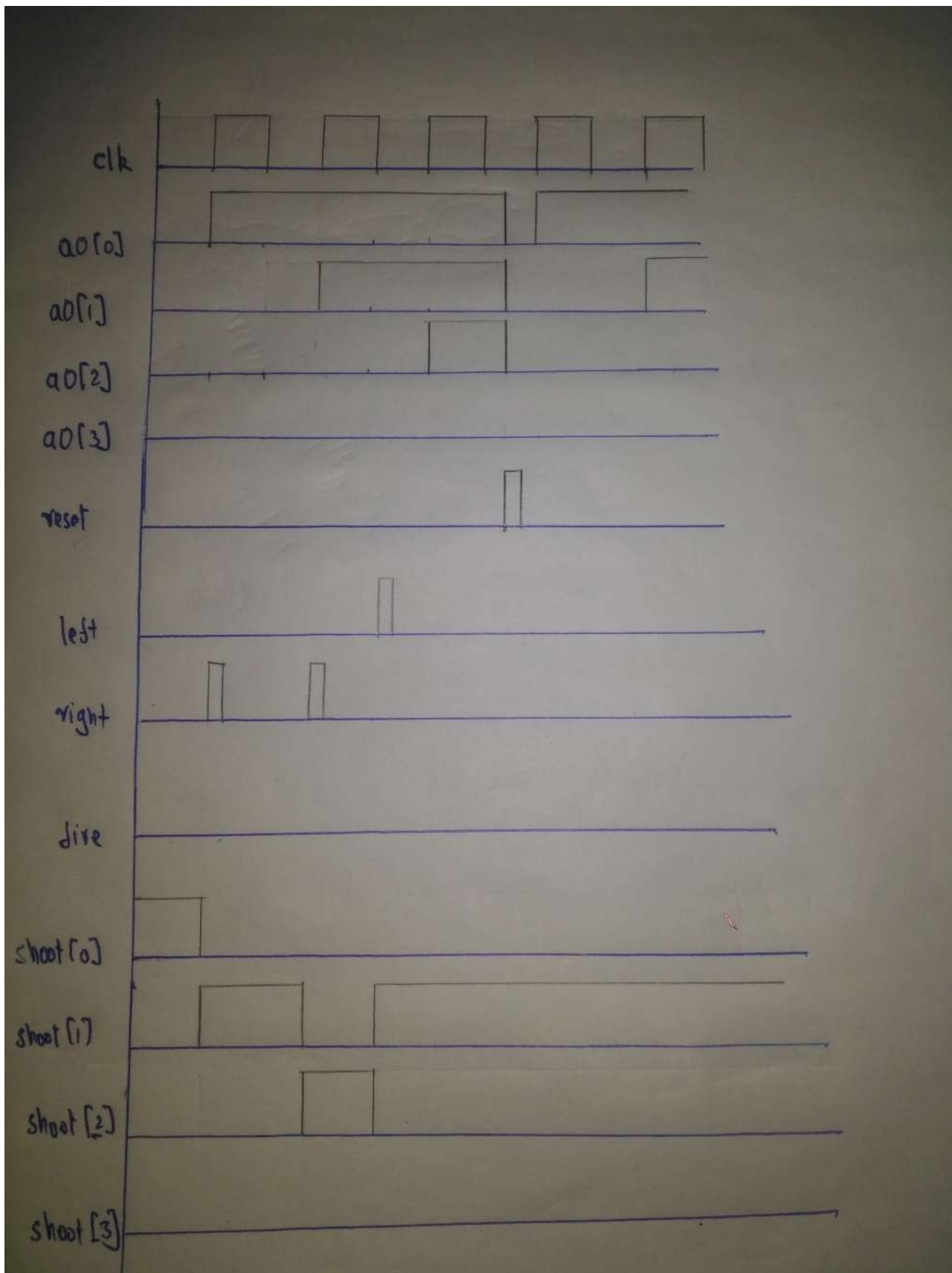
The ball (here LED) can glow randomly and then there is shooter at the bottom who will shoot the ball at a particular column,when it shoots at a particular column the whole column goes off.

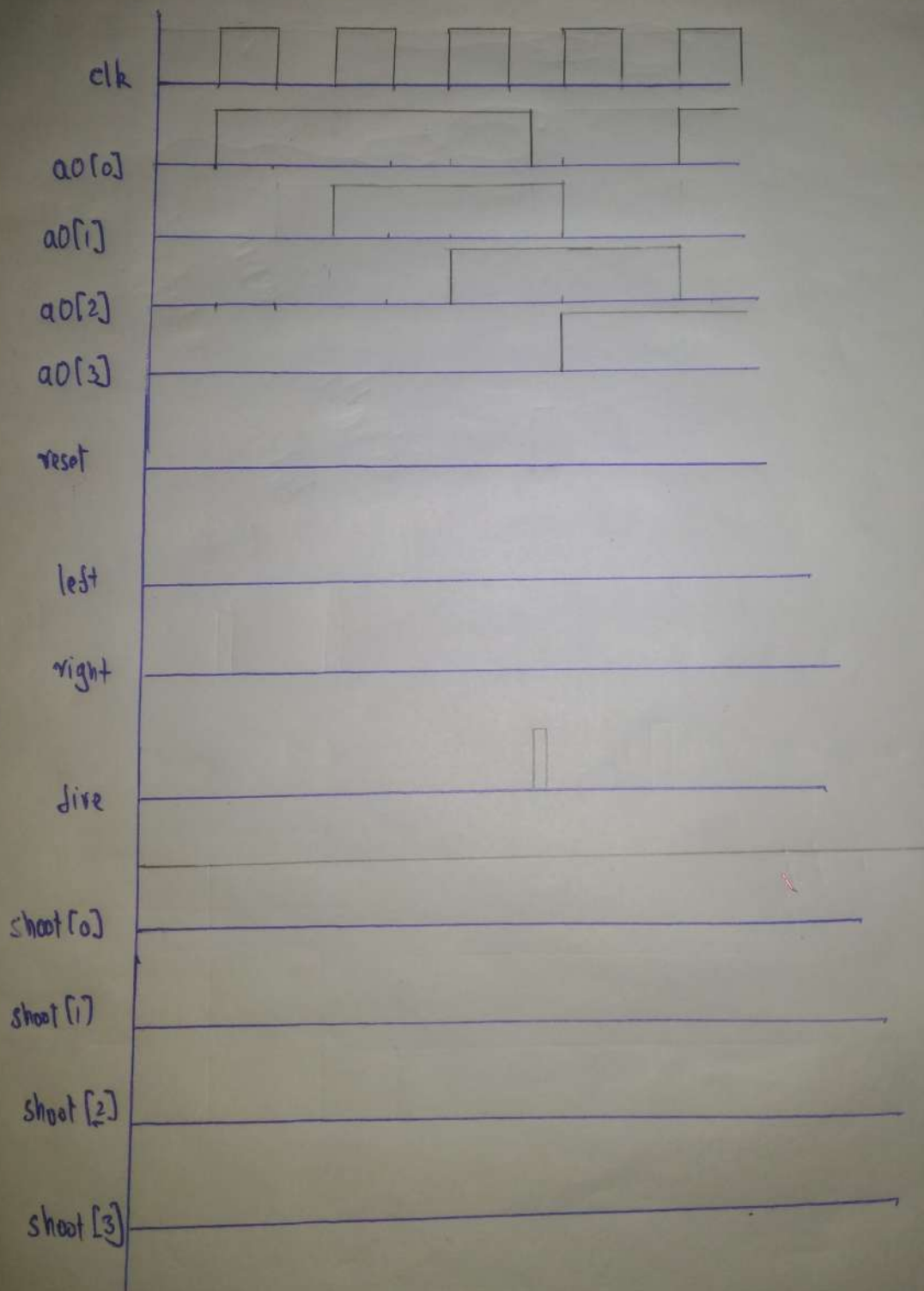
And if the ball reaches at the bottom and colloid with the shooter the game gets over.

INTRODUCTION

TIMING DIAGRAM:







Tri Canola	Maccanoe	1.00
------------	----------	------

IMPLEMENTATION

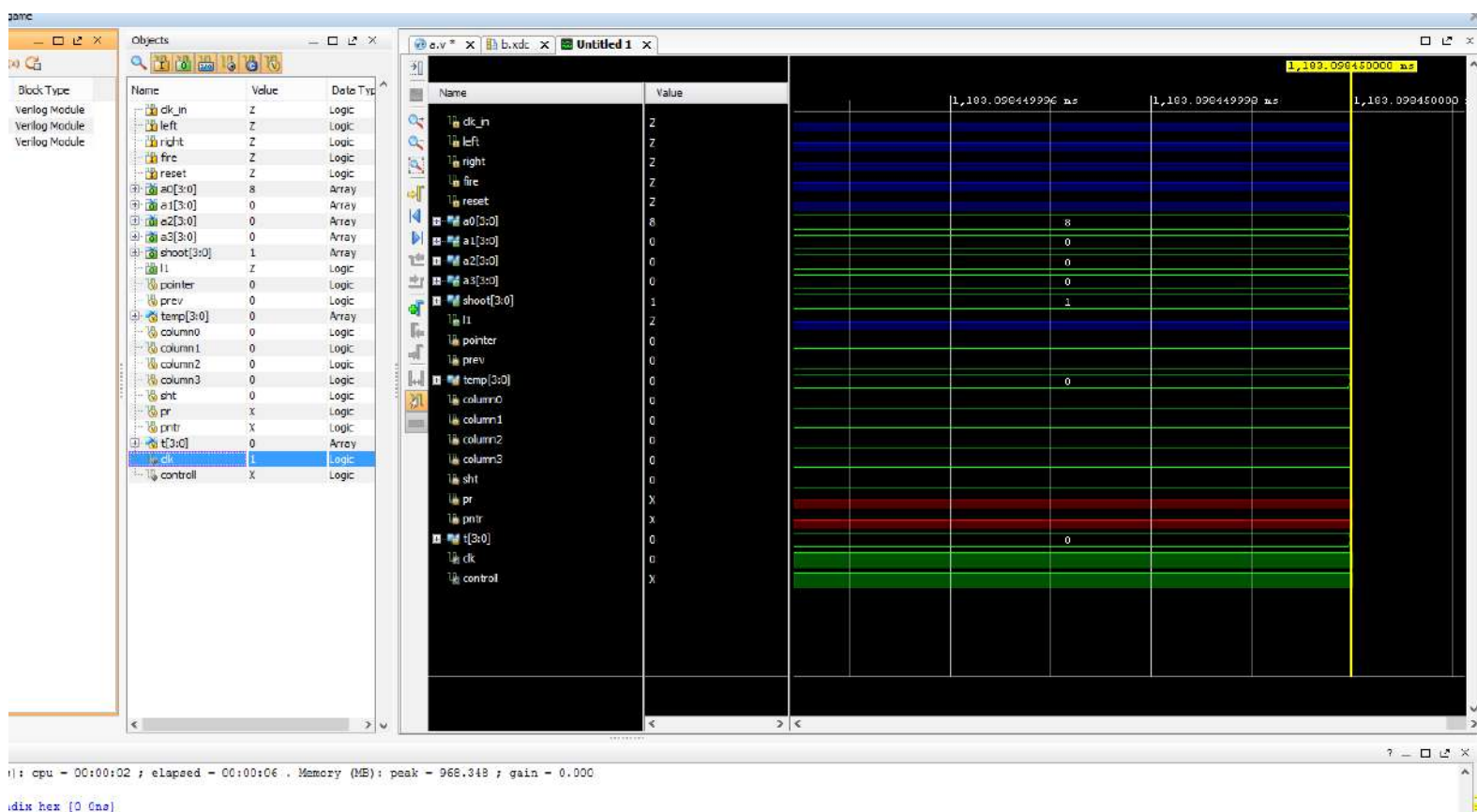
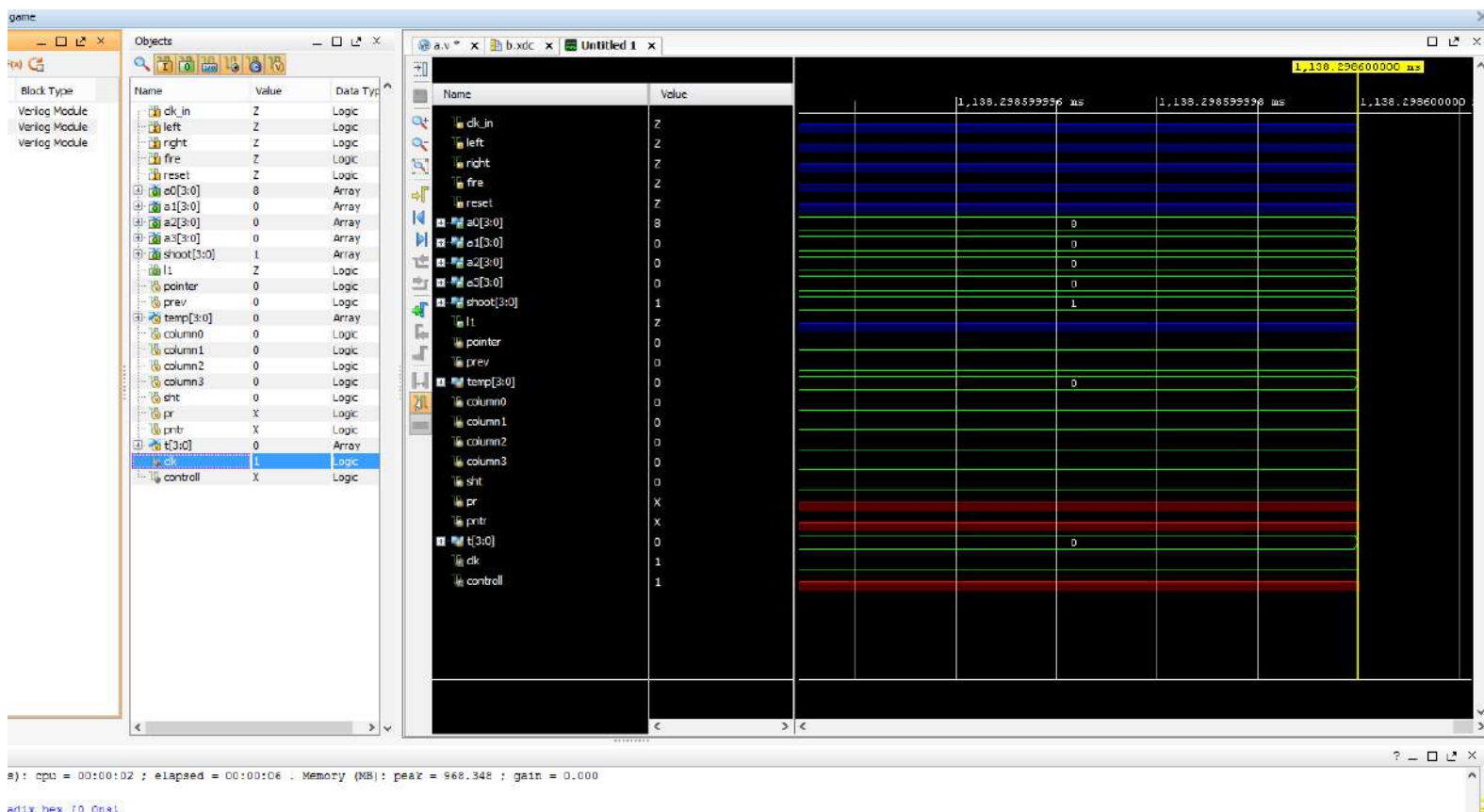
LOGIC:

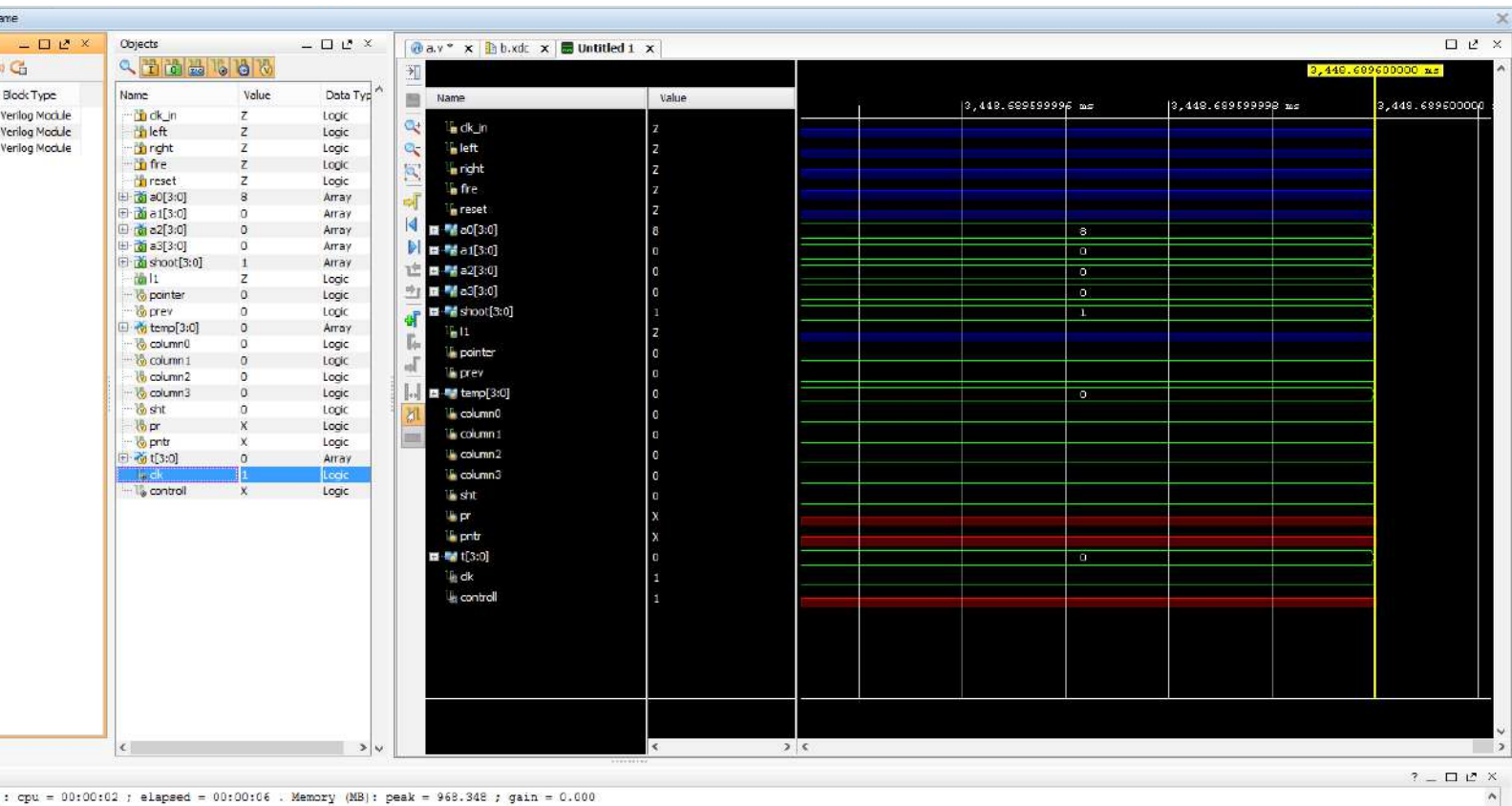
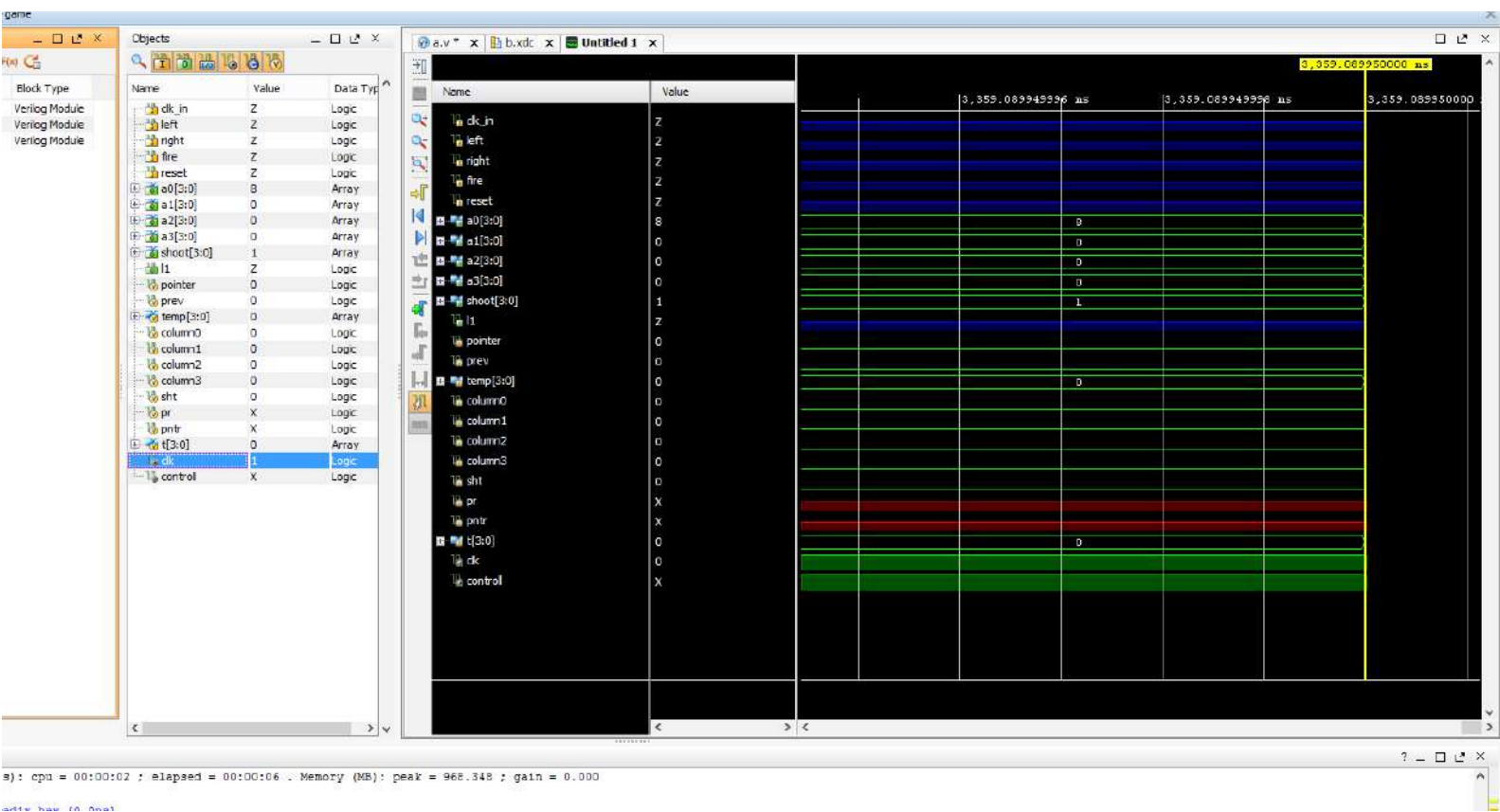
the logic is there is a random led glowing code which enable the led 4x4 matrix to glow in random pattern ,this apply for each and every column.

Then the led column will be taken care by the shooter led panel which is 1x4 matrix column at the bottom. The shooter led is moving by the push switches from zybo. The led is moving like 1000,0100,0010,0001 for left shift and for the right shift is like 0001,0010,0100,1000.

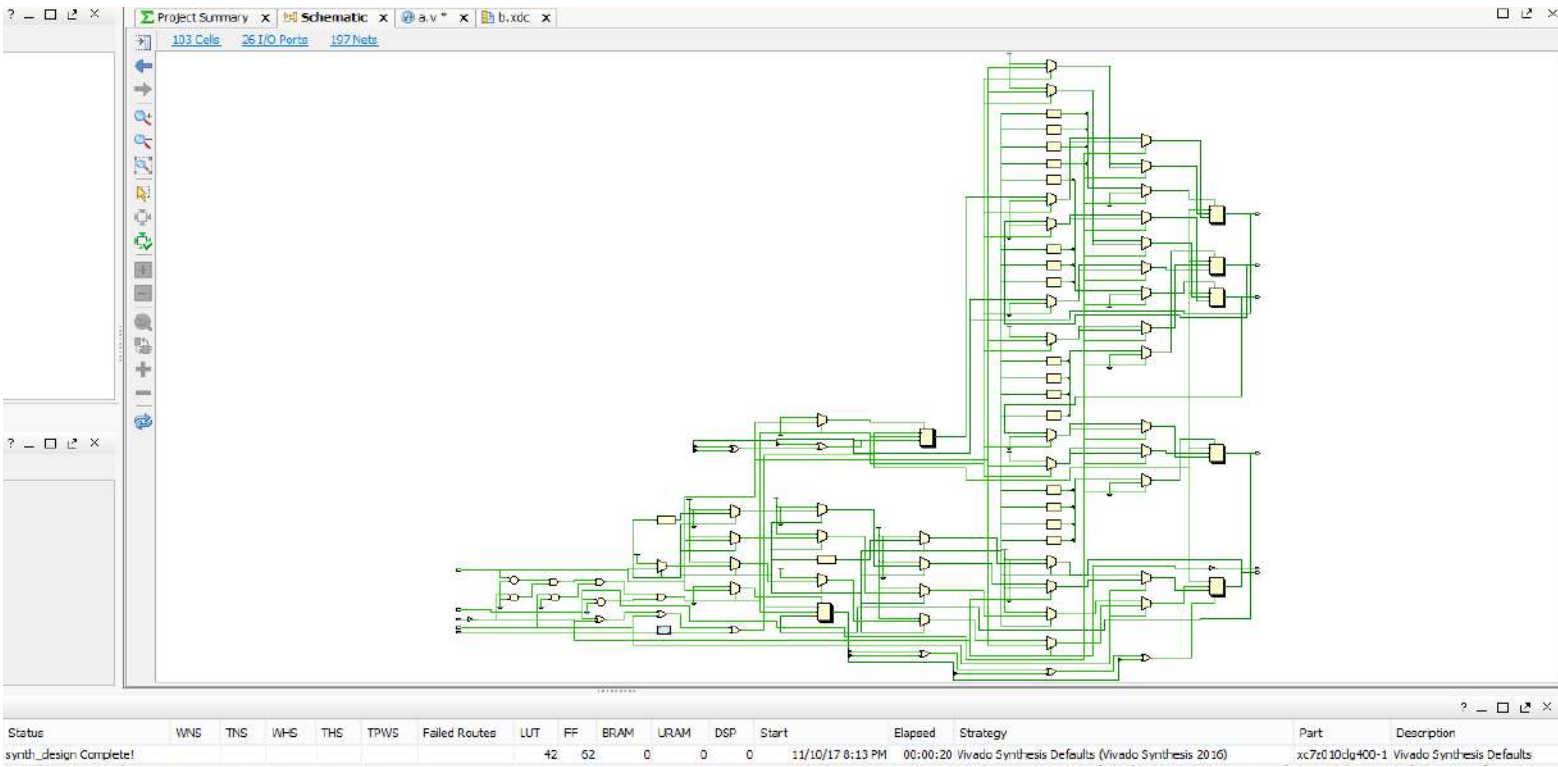
The shooting of the shooter will reset the whole particular column. By setting the value of whole column by the value 0000.

SIMULATION





RTL SCHEMATIC



CONCLUSION

in the conclusion we would like say that the project was quite fun.

Most of the time some errors are irritating and head busters,which took our most of the time though we have solved it. The implimentation and hardware assembly was awesome,first time uses glue gun.

Though we did some mistake but Arun Rahul sir guided us and we would like to thank him and also Gayatri mam who was there for even when we ask for a childish mistake.

At the end i would like to conclude that the game works good in the hardware.

APPENDIX

VERILOG CODE

```
module clkgen(a, b);
output reg a;
reg [32:0]clkvalue = 0;
input b;
initial
begin
a = 0;
end
always @(negedge b)
begin
if (clkvalue == 32'd800000000)
begin
a <= ~a;
clkvalue <= 0;
end
else
clkvalue <= clkvalue + 1;
end
endmodule
```

```
module game(                                     //game module
input clk_in,
input left,right,fire,reset,
output reg [3:0] a0,
output reg [3:0] a1,
output reg [3:0] a2,
output reg [3:0] a3,
output reg [3:0] shoot,
output l1
);
reg pointer,prev;
reg [3:0] temp;
reg column0,column1,column2,column
reg sht;
reg pr,pntr;
```

```

reg [3:0] t;
initial
begin
    shoot <= 4'b0001;
    a0 <= 4'b1000;
    a1 <= 4'b0000;
    a2 <= 4'b0000;
    a3 <= 4'b0000;
    pointer = 2'b00;
    prev = 2'b00;
    column0 = 2'b00;
    column1 = 2'b00;
    column2 = 2'b00;
    column3 = 2'b00;
    sht <= 2'b00;
    t[3] <= 1'b0;
    t[2] <= 1'b0;
    t[1] <= 1'b0;
    t[0] <= 1'b0;
end
wire clk;
wire controll;
assign controll = fire | left | right | clk;
clkgen generate_clock(clk,clk_in);           //randome clock generator
always @(posedge controll)
begin
    if(reset == 1)
    begin
                                                //Reset value

        temp[3] <= 1'b0;
        temp[2] <= 1'b0;
        temp[1] <= 1'b0;
        temp[0] <= 1'b0;
        a0 <= 4'b0000;
        a1 <= 4'b0000;
        a2 <= 4'b0000;
        a3 <= 4'b0000;
        t <= 4'b0000;
        shoot <= 4'b0001;
    end
end

```

```

if( reset == 0 & left == 0 & right == 0 & fire == 0)
begin
                                                                    //Random code generator

temp[3] <= a0[2] ;
temp[2] <= a0[1];
temp[1] <= a0[0];
temp[0] <= (~a0[3])|(a0[2]^a0[1]);
a0 <= temp;
a1 <= a0 ;
a2 <= a1 ;
a3 <= a2 ;
t <= a3 ;
end
if(fire == 1)
begin
if(shoot == 4'b0001)
begin
a0[0] <= 1'b0;
a1[0] <= 1'b0;
a2[0] <= 1'b0;
a3[0] <= 1'b0;
end
if(shoot == 4'b0010)
begin
a0[1] <= 1'b0;
a1[1] <= 1'b0;
a2[1] <= 1'b0;
a3[1] <= 1'b0;
end
if(shoot == 4'b0100)
begin
a0[2] <= 1'b0;
a1[2] <= 1'b0;
a2[2] <= 1'b0;
a3[2] <= 1'b0;
end
if(shoot == 4'b1000)
begin
a0[3] <= 1'b0;
a1[3] <= 1'b0;

```

```

        a2[3] <= 1'b0;
        a3[3] <= 1'b0;
    end
end
    if(right == 1'b1)
begin
    if(shoot == 4'b1000)
        shoot <= 4'b0001;
    if(shoot == 4'b0100)
        shoot <= 4'b1000;
    if(shoot == 4'b0010)
        shoot <= 4'b0100;
    if(shoot == 4'b0001)
        shoot <= 4'b0010;
end
    else if(left == 1'b1)
begin
    if(shoot == 4'b0001)
        shoot <= 4'b1000;
    if(shoot == 4'b0100)
        shoot <= 4'b0010;
    if(shoot == 4'b0010)
        shoot <= 4'b0001;
    if(shoot == 4'b1000)
        shoot <= 4'b0100;
end
    if(t[0] == 1'b1 | t[1] == 1'b1 | t[2] == 1'b1 | t[3] == 1'b1)
begin
shoot <= 4'b1111;
end
end
assign l1 = left;
endmodule

```

CONSTRAINTS FILE:

```
set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [get_ports {clk_in}]
set_property -dict {PACKAGE_PIN Y16 IOSTANDARD LVCMOS33 } [get_ports {left}]
set_property -dict {PACKAGE_PIN V16 IOSTANDARD LVCMOS33 } [get_ports {right}]
set_property -dict {PACKAGE_PIN P16 IOSTANDARD LVCMOS33 } [get_ports {fire}]
set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports {reset}]
```

```
set_property -dict {PACKAGE_PIN T14 IOSTANDARD LVCMOS33 } [get_ports {a0[0]}]
set_property -dict {PACKAGE_PIN T15 IOSTANDARD LVCMOS33 } [get_ports {a0[1]}]
set_property -dict {PACKAGE_PIN P14 IOSTANDARD LVCMOS33 } [get_ports {a0[2]}]
set_property -dict {PACKAGE_PIN R14 IOSTANDARD LVCMOS33 } [get_ports {a0[3]}]
```

```
set_property -dict {PACKAGE_PIN U14 IOSTANDARD LVCMOS33 } [get_ports {a1[0]}]
set_property -dict {PACKAGE_PIN U15 IOSTANDARD LVCMOS33 } [get_ports {a1[1]}]
set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVCMOS33 } [get_ports {a1[2]}]
set_property -dict {PACKAGE_PIN V18 IOSTANDARD LVCMOS33 } [get_ports {a1[3]}]
```

```
set_property -dict {PACKAGE_PIN V15 IOSTANDARD LVCMOS33 } [get_ports {a2[0]}]
set_property -dict {PACKAGE_PIN W15 IOSTANDARD LVCMOS33 } [get_ports {a2[1]}]
set_property -dict {PACKAGE_PIN T11 IOSTANDARD LVCMOS33 } [get_ports {a2[2]}]
set_property -dict {PACKAGE_PIN T10 IOSTANDARD LVCMOS33 } [get_ports {a2[3]}]
```

```
set_property -dict {PACKAGE_PIN W14 IOSTANDARD LVCMOS33 } [get_ports {a3[0]}]
set_property -dict {PACKAGE_PIN Y14 IOSTANDARD LVCMOS33 } [get_ports {a3[1]}]
set_property -dict {PACKAGE_PIN T12 IOSTANDARD LVCMOS33 } [get_ports {a3[2]}]
set_property -dict {PACKAGE_PIN U12 IOSTANDARD LVCMOS33 } [get_ports {a3[3]}]
```

```
set_property -dict {PACKAGE_PIN T20 IOSTANDARD LVCMOS33 } [get_ports {shoot[0]}]
set_property -dict {PACKAGE_PIN U20 IOSTANDARD LVCMOS33 } [get_ports {shoot[1]}]
set_property -dict {PACKAGE_PIN V20 IOSTANDARD LVCMOS33 } [get_ports {shoot[2]}]
set_property -dict {PACKAGE_PIN W20 IOSTANDARD LVCMOS33 } [get_ports {shoot[3]}]
```

```
set_property -dict {PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports {l1}]
```