Lab 3

Verilog code- Top Module

```
module top(
   input ori_clk,
   input rst,
   output reg [4:0] q=0,
   input [1:0] sel_line
   );

wire [4:0] out1;
wire [4:0] out2;
wire [4:0] out3;              // four 4-bit wire variables to store the values for 1,2,4 and 8 sec rates
wire [4:0] out4;

wire f1,f2,f3;
wire clk,deb_out;

frq_one fun1(.clk(ori_clk),.out_clk(clk));                  // frequency divider to get 1 Hz

debounce fun2(.clk_deb(clk),.deb_out(deb_out),.pb_press(rst));          // Debounce
dff fun3(.clk(clk),.rst(deb_out),.q(out1));                             // 0-31 Counter for 1 Hz

counter #(1) fun4(.clk(clk),.div_clk(f1));                 // frequency divider to get 0.5 Hz
dff fun5(.clk(f1),.rst(deb_out),.q(out2));                 // 0-31 Counter for 0.5 Hz

counter #(2) fun6(.clk(clk),.div_clk(f2));                 // frequency divider to get 0.25 Hz
dff fun7(.clk(f2),.rst(deb_out),.q(out3));                 // 0-31 Counter for 0.5 Hz

counter #(3) fun8(.clk(clk),.div_clk(f3));                 // frequency divider to get 0.125 Hz
dff fun9(.clk(f3),.rst(deb_out),.q(out4));                 // 0-31 Counter for 0.125 Hz

always @ (posedge clk)
begin
   if(sel_line==2'b00)                          // if sel_line is 0, choose at the rate of 1 sec
   begin
      q<=out1;
   end
   else if(sel_line==2'b01)                     // if sel_line is 1, choose at the rate of 2 sec
   begin
      q<=out2;
       end
   else if(sel_line==2'b10)                     // if sel_line is 2, choose 1 at the rate of 4 sec
   begin
      q<=out3;
```

```
    end
    else if(sel_line==2'b11)              // if sel_line is 3, choose at the rate of 8 sec
    begin
        q<=out4;
    end
end

Endmodule
```

Verilog code- Counter

```
module dff(
    input clk,
    output reg [4:0] q=0,     // 32 is of 5 bits, Therefore 4:0
    input rst
    );

// always @ (posedge clk | rst) // asynchronous clk
always @ (posedge clk)  // synchronous clk
begin
    if(rst | q==31)         // for a 0-31 counter, reset it after 31 or if reset is pressed.
        q<=5'b0;           // making it zero
    else
        q<=q+1;         // increment counter by 1.
end

Endmodule
```

Verilog code- Frequency divider

```
module counter(
    input clk,
    output div_clk
    );

parameter n=25;
```

```
reg [n-1:0] q=0;      // parameter

always @ (posedge clk)
begin

q<=q+1;     // incrementing

end

assign div_clk=q[n-1];     // taking the (n-1)th bit from the incremented q.

Endmodule
```

Verilog code- Debounce

```
module debounce(
   input clk_deb,
   output deb_out,
   input pb_press
   );

   reg [2:0] ff;

   always @ (posedge clk_deb)
   begin
      ff[2]<=pb_press;
      ff[1]<=ff[2];
      ff[0]<=ff[1];
   end

   assign deb_out=ff[0]&ff[1]&ff[2];

Endmodule
```

Verilog code- 1 hz frequency generator

```
module frq_one(
input clk,
output reg out_clk=0
);

reg [26:0] count=0;

always @ (posedge clk)
begin
        if(count==49999999)
        begin
                count<=0;               // resetting its value
                out_clk<=~out_clk;
        end
        else
        Begin
                count<=count+1;     // Incrementing the counter
        end
end

endmodule
```

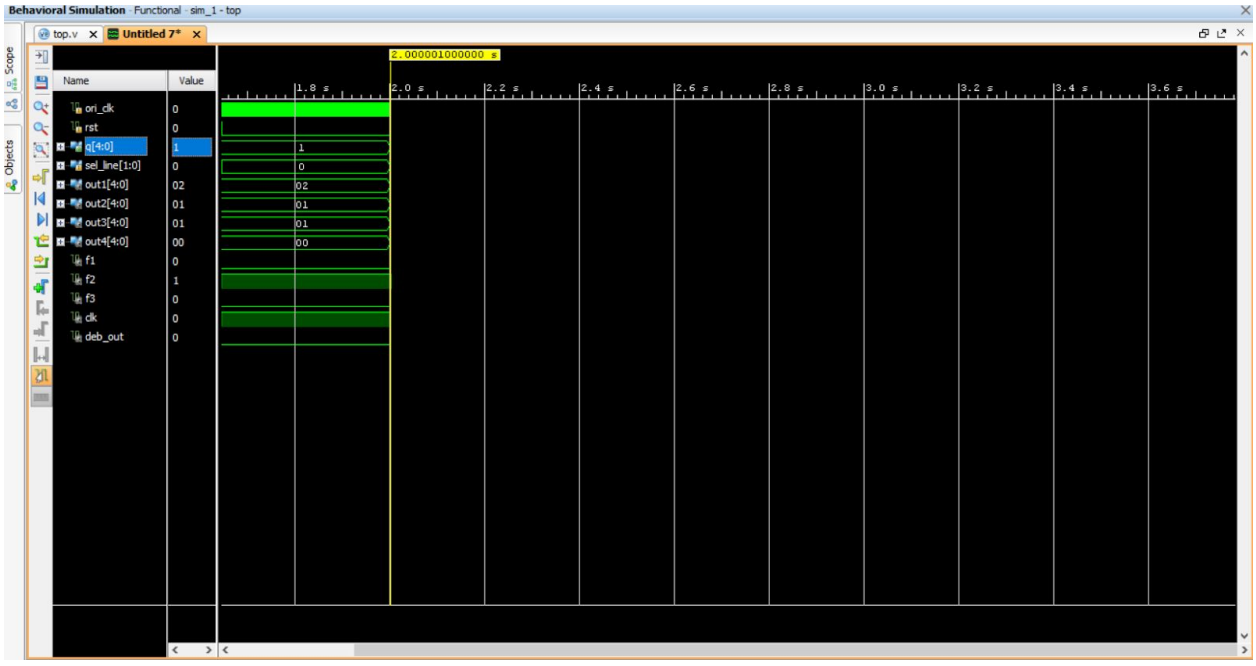
// we know that frequency of basys3 board is 100M hz. Therefore in order to get 1 hz out of it we did the calculation which goes like this, **(10^8) / (2^n) = 1** . And after solving this the value of ‘n’ was approximately 26. Therefore the frequency of the output waveform should be 2^26 times lower than the clock frequency of input frequency.

Xdc File

```
# Switch
set_property PACKAGE_PIN W5 [get_ports ori_clk]
        set_property IOSTANDARD LVCMOS33 [get_ports ori_clk]
# Inputs
set_property PACKAGE_PIN v16 [get_ports {sel_line[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sel_line}]
set_property PACKAGE_PIN W17 [get_ports {sel_line[1]}]
# Buttons
```

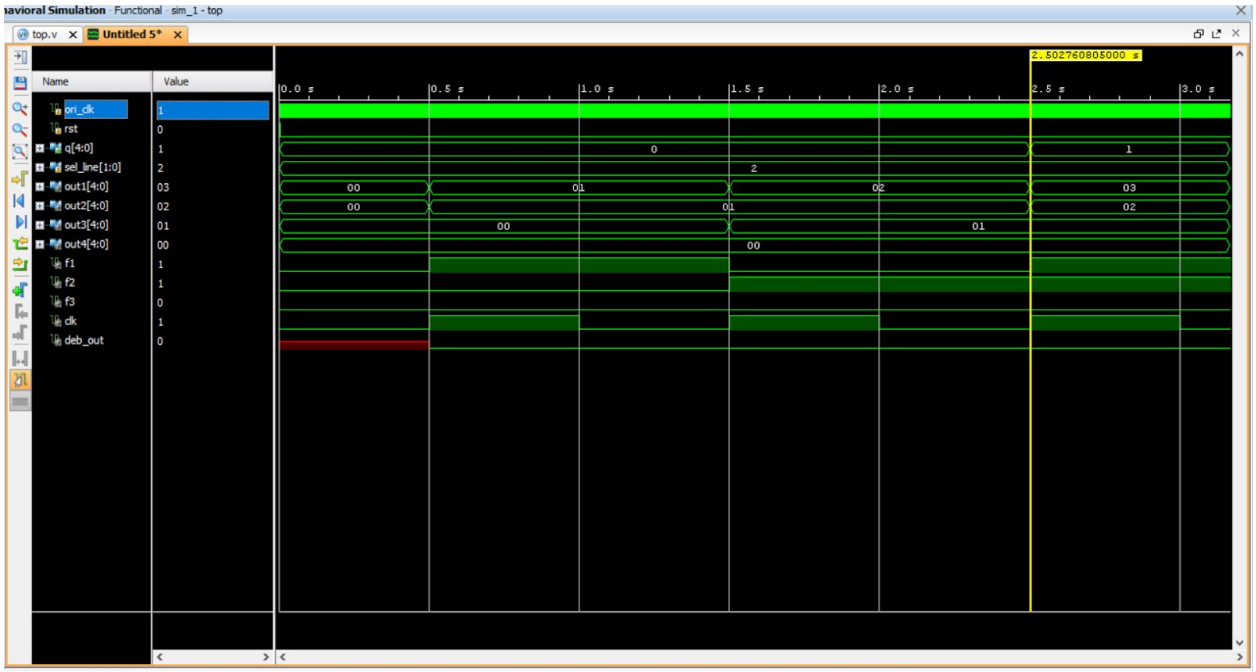
```
set_property PACKAGE_PIN U18 [get_ports rst]
    set_property IOSTANDARD LVCMOS33 [get_ports rst]
# LED's
set_property PACKAGE_PIN U16 [get_ports {q[0]}]
            set_property IOSTANDARD LVCMOS33 [get_ports {q}]
set_property PACKAGE_PIN E19 [get_ports {q[1]}]
set_property PACKAGE_PIN U19 [get_ports {q[2]}]
set_property PACKAGE_PIN V19 [get_ports {q[3]}]
set_property PACKAGE_PIN W18 [get_ports {q[4]}]
```

Outputs

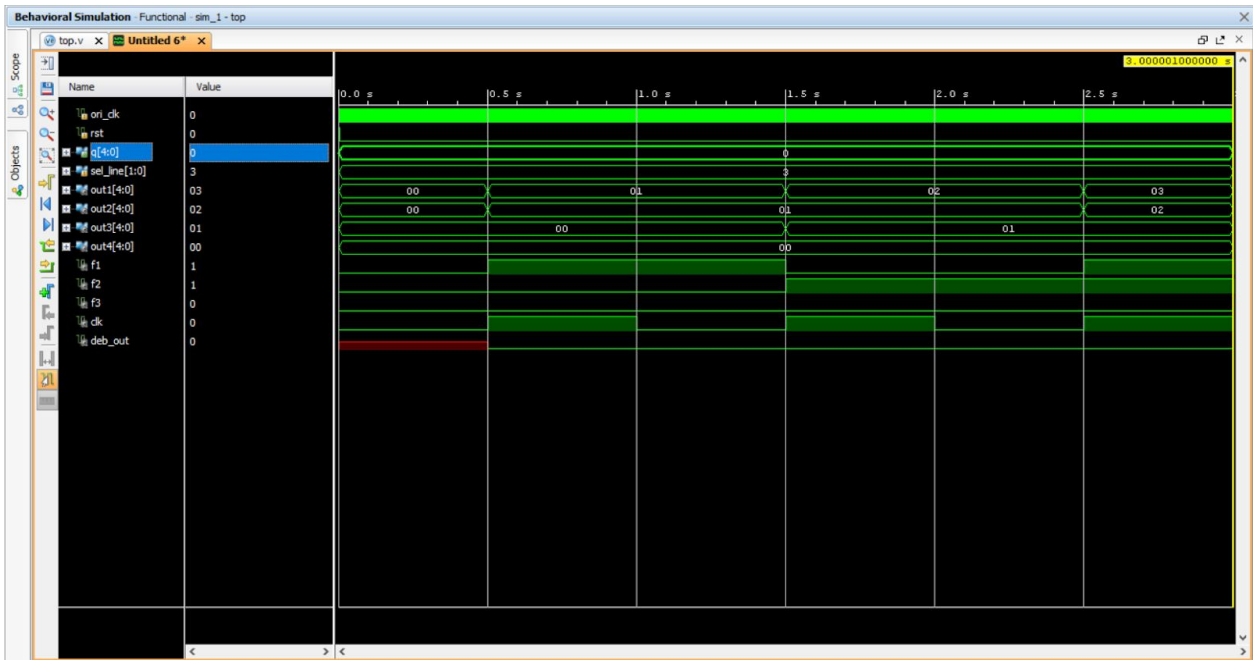


sel_line= 00 (0)

sel_line= 01 (1)



sel_line= 10 (2)


Behavioral Simulation - Functional - sim_1 - top
top.v
Untitled 6*
Scope
Objects
Name
Value
ori_clk
rst
q[4:0]
sel_line[1:0]
out1[4:0]
out2[4:0]
out3[4:0]
out4[4:0]
f1
f2
f3
clk
deb_out
0
0
0
3
03
02
01
00
1
1
0
0
0
0.0 s
0.5 s
1.0 s
1.5 s
2.0 s
2.5 s
3.000001000000 s

sel_line= 11 (3)