Nolan Anderson

CPE 322

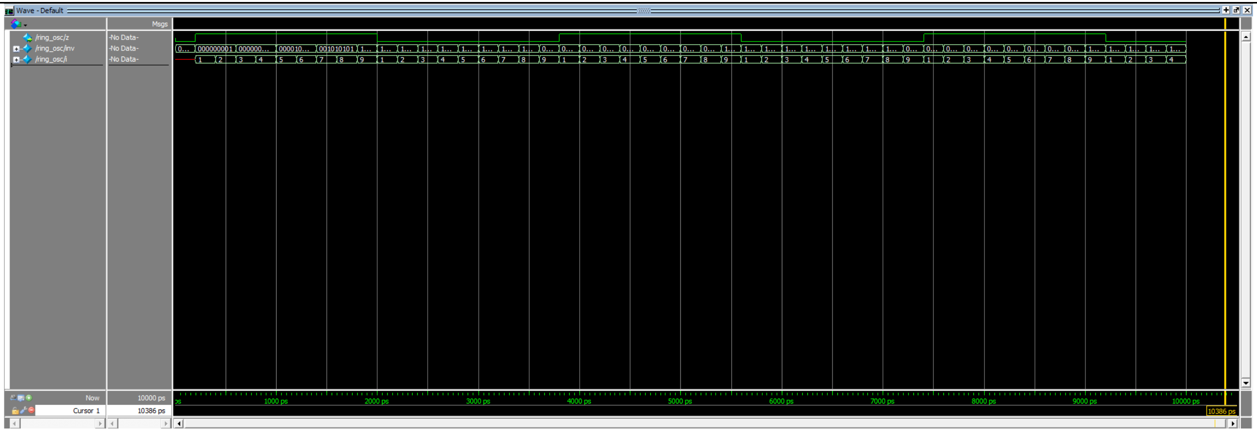
Dr. Detwiler

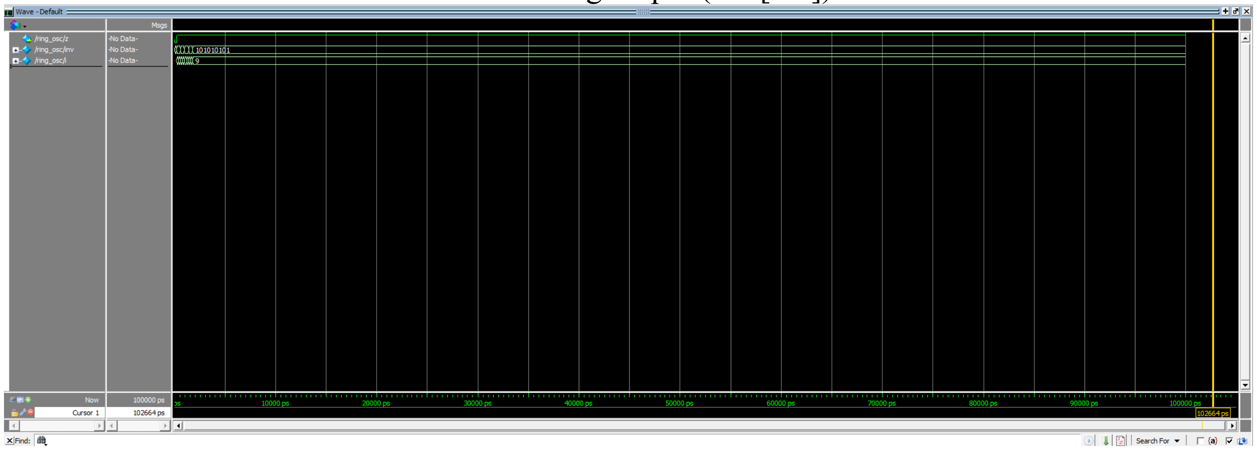
19 April 2021

Simulation Assignment

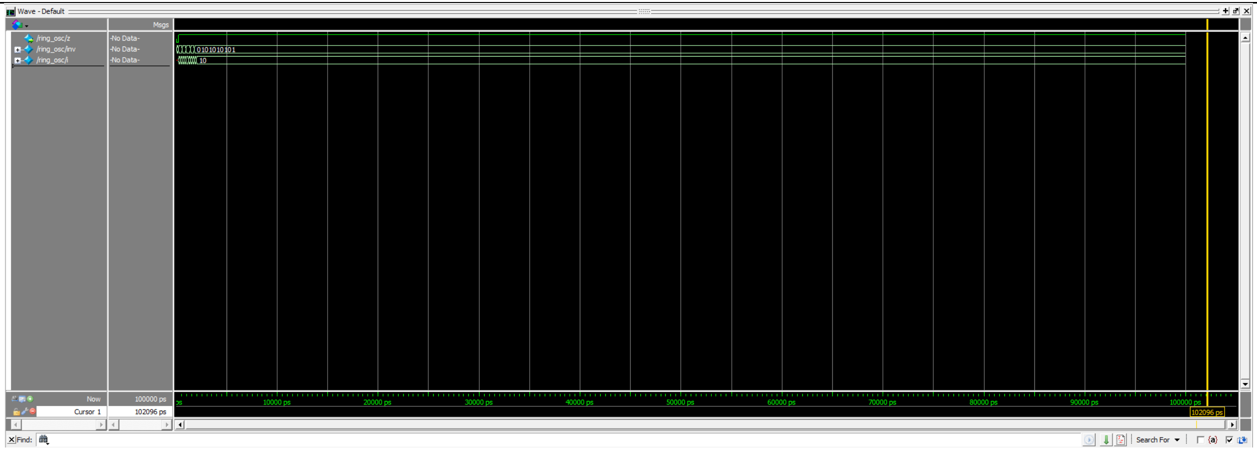
This document contains my submission for the simulation assignment, problems 1-6. They will be separated into 6 different sections with code and answers to the questions provided in the document.

Section 1 – Ring Oscillator

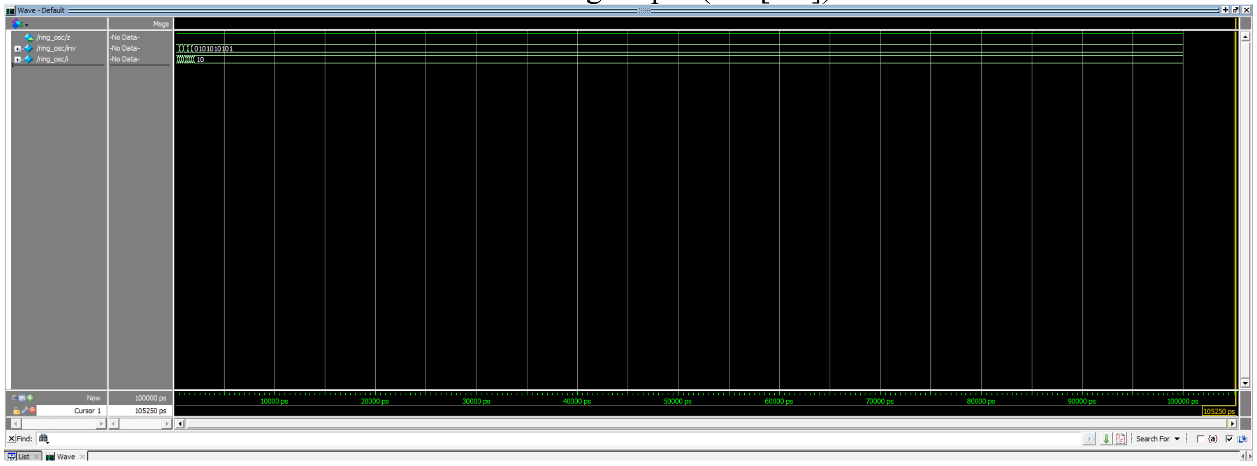

Non-blocking output (inv [8:0])


Blocking output (inv [8:0])

As you can see, the blocking output leads to no information being taken in after the first set. I stays the same the whole time.(code for this problem is on page 3).



Non-blocking output (inv [9:0])

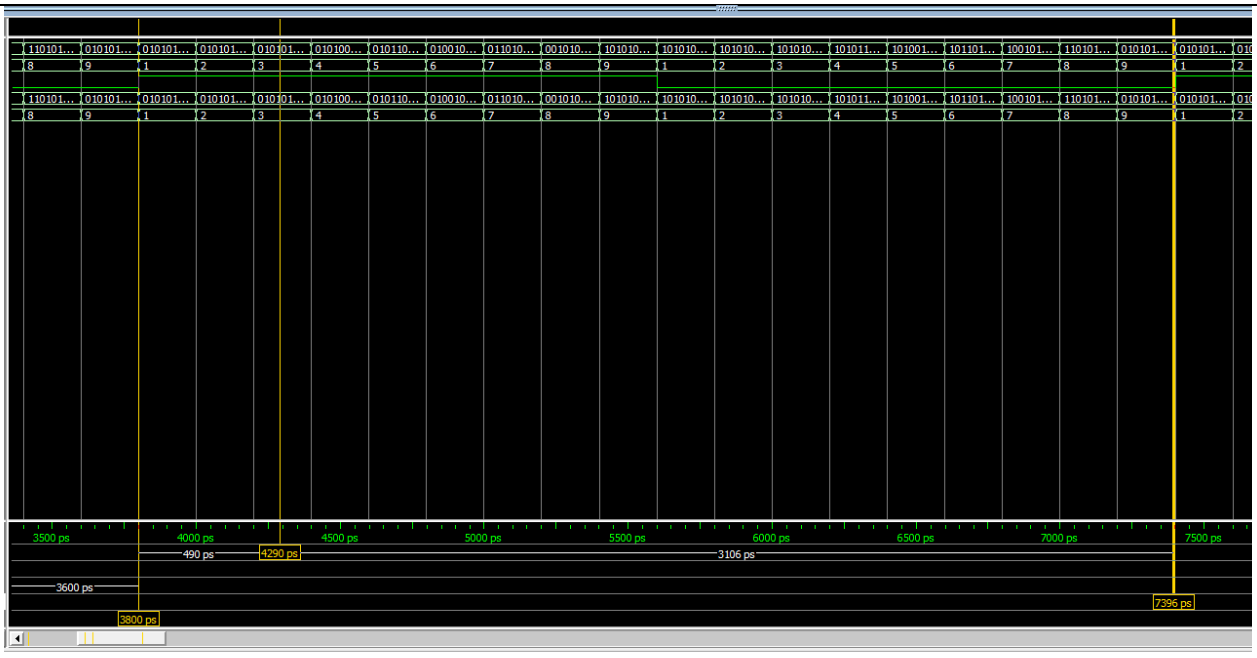


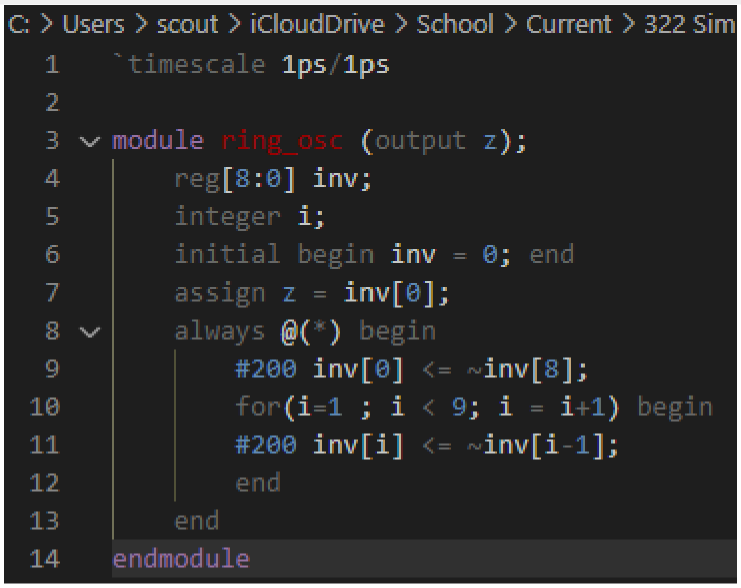
Blocking output (inv [9:0])

WHAT HAPPENS TO THE z OUTPUT WITH 10 INVERTERS?

I get the same output when I do 10 bits, doesn't seem to like it. This is because the number of inverters has to be even.

WHAT IS THE REPORTED DIFFERENCE BETWEEN CURSORS ON TWO CONSECUTIVE RISING EDGES? 3600ps

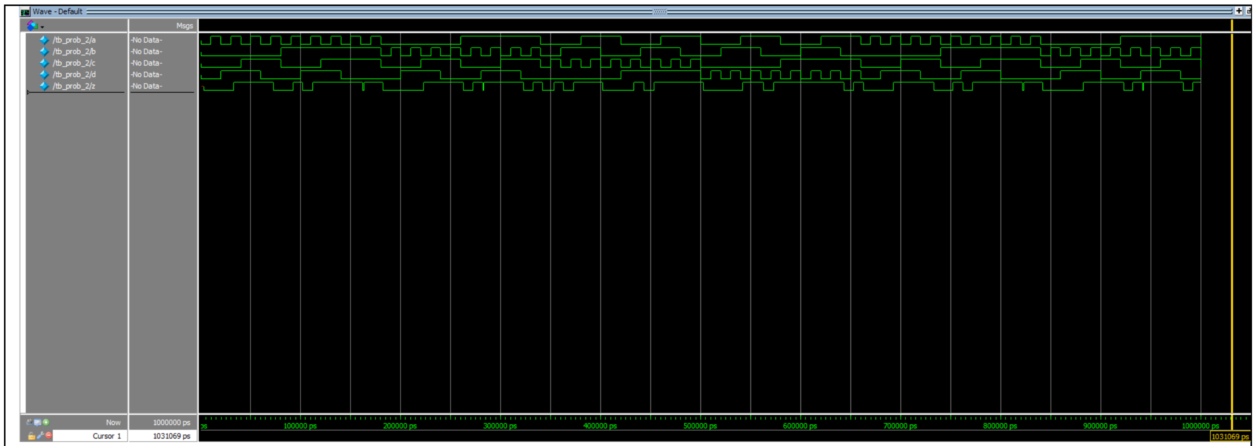



C: > Users > scout > iCloudDrive > School > Current > 322 Sim

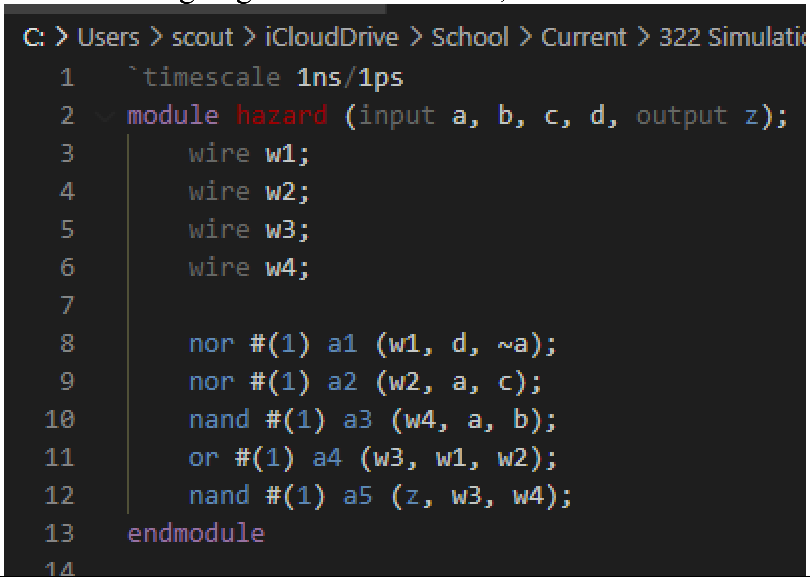
```verilog
`timescale 1ps/1ps

module ring_osc (output z);
    reg[8:0] inv;
    integer i;
    initial begin inv = 0; end
    assign z = inv[0];
    always @(*) begin
        #200 inv[0] <= ~inv[8];
        for(i=1 ; i < 9; i = i+1) begin
        #200 inv[i] <= ~inv[i-1];
        end
    end
endmodule
```

Finally, this is the code I used for this problem.

Section 2 – Hazard Detection

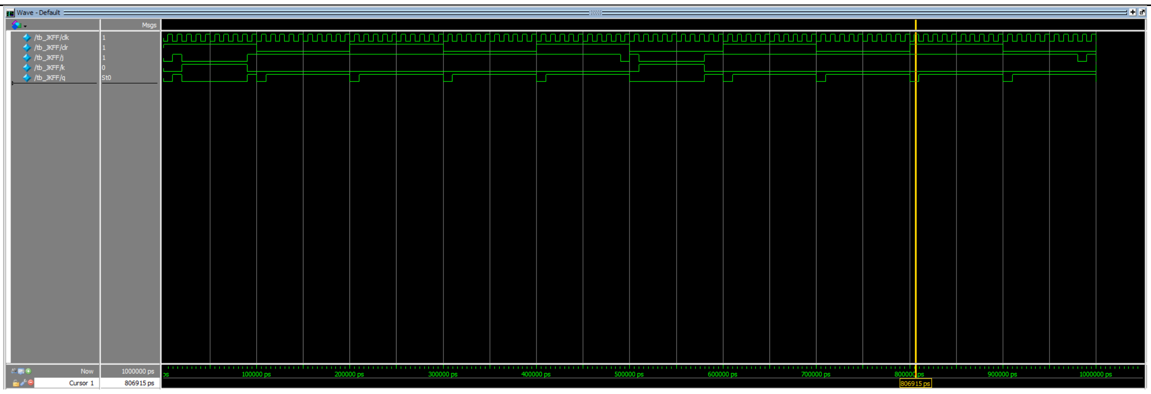


The bottom line is my output. There is one hazard right before 20000ps, and I used brute force. The hazard is on A when going from 1110 to 0110, but it does not work the other way.



```
C: > Users > scout > iCloudDrive > School > Current > 322 Simulati
`timescale 1ns/1ps
module hazard (input a, b, c, d, output z);
    wire w1;
    wire w2;
    wire w3;
    wire w4;

    nor #(1) a1 (w1, d, ~a);
    nor #(1) a2 (w2, a, c);
    nand #(1) a3 (w4, a, b);
    or #(1) a4 (w3, w1, w2);
    nand #(1) a5 (z, w3, w4);
endmodule
```

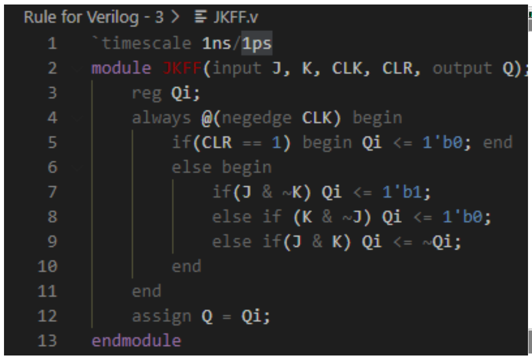
Section 3 – Rule for Verilog



Does it hold the Q output low, as required?

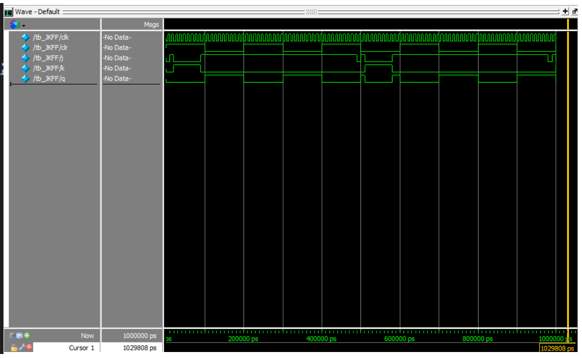
- No, as you can see, there are multiple examples where the Q output does not remain low.

Modified JKFF module: As you can see, the output remains low on clear being high.

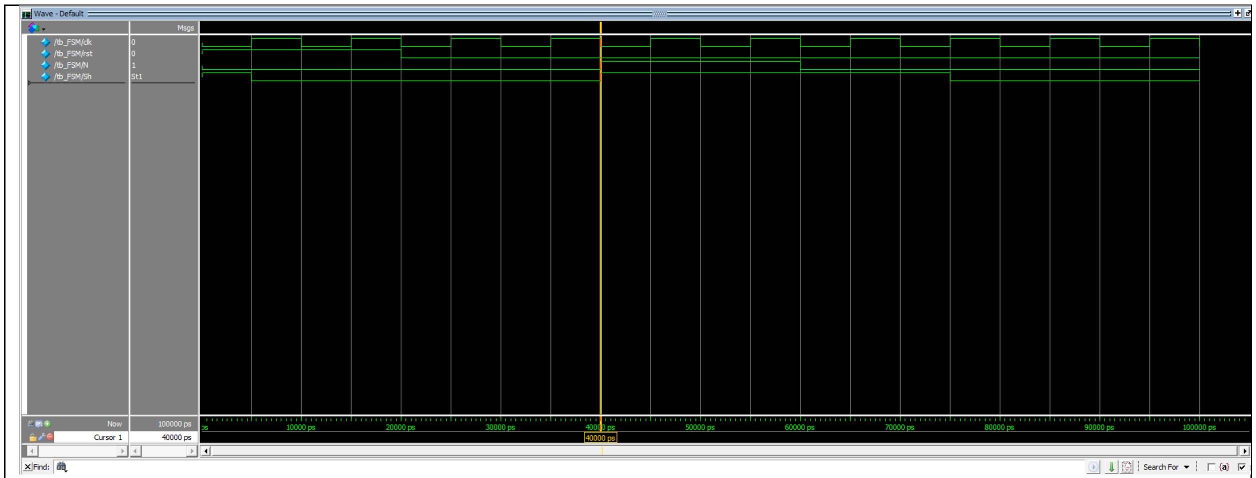


Rule for Verilog - 3 > JKFF.v

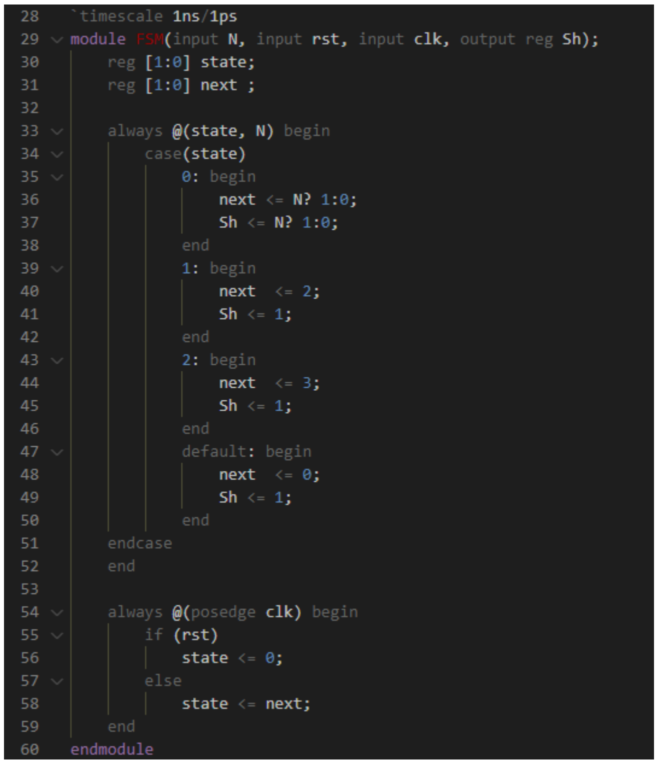
```verilog
`timescale 1ns/1ps
module JKFF(input J, K, CLK, CLR, output Q);
    reg Qi;
    always @(negedge CLK) begin
        if(CLR == 1) begin Qi <= 1'b0; end
        else begin
            if(J & ~K) Qi <= 1'b1;
            else if (K & ~J) Qi <= 1'b0;
            else if(J & K) Qi <= ~Qi;
        end
    end
    assign Q = Qi;
endmodule
```



Section 4 – FSM Coding



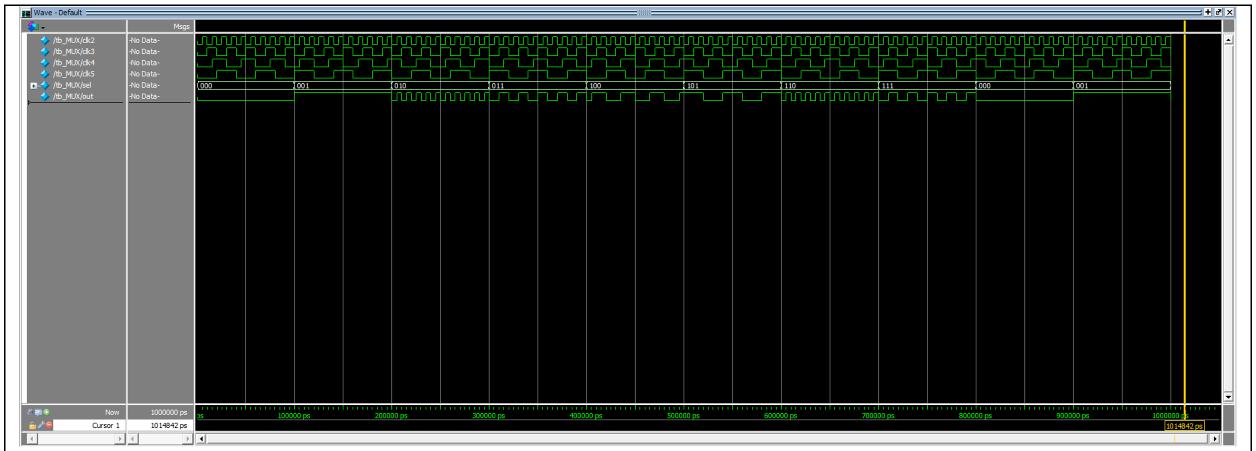
At 40nsec, Sh goes high when N goes high.



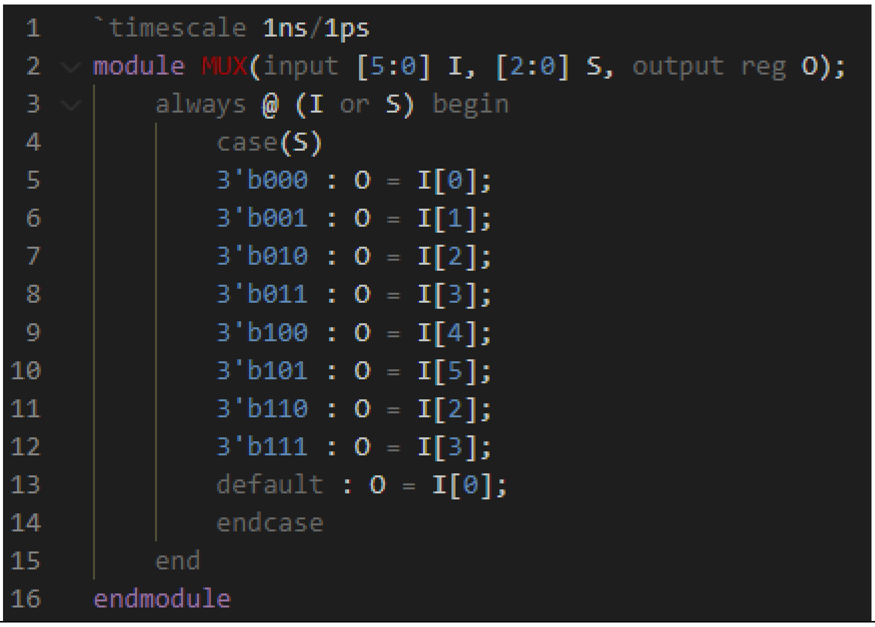
```verilog
`timescale 1ns/1ps
module FSM(input N, input rst, input clk, output reg Sh);
    reg [1:0] state;
    reg [1:0] next ;

    always @(state, N) begin
        case(state)
            0: begin
                next <= N? 1:0;
                Sh <= N? 1:0;
            end
            1: begin
                next  <= 2;
                Sh <= 1;
            end
            2: begin
                next  <= 3;
                Sh <= 1;
            end
            default: begin
                next  <= 0;
                Sh <= 1;
            end
    endcase
    end

    always @(posedge clk) begin
        if (rst)
            state <= 0;
        else
            state <= next;
    end
endmodule
```

This is the code I generated for #4

Section 5 – Verilog 6:1 MUX



As you can see, there are 6 distinct patters on the output.
Here is the code that I generated for the 6:1 MUX:



```verilog
`timescale 1ns/1ps
module MUX(input [5:0] I, [2:0] S, output reg O);
    always @ (I or S) begin
        case(S)
        3'b000 : O = I[0];
        3'b001 : O = I[1];
        3'b010 : O = I[2];
        3'b011 : O = I[3];
        3'b100 : O = I[4];
        3'b101 : O = I[5];
        3'b110 : O = I[2];
        3'b111 : O = I[3];
        default : O = I[0];
        endcase
    end
endmodule
```

Section 6 – Delays

Waveform screenshot at time t = 0 and 100 ns. I will refrain from providing code as it was provided in the assignment document.

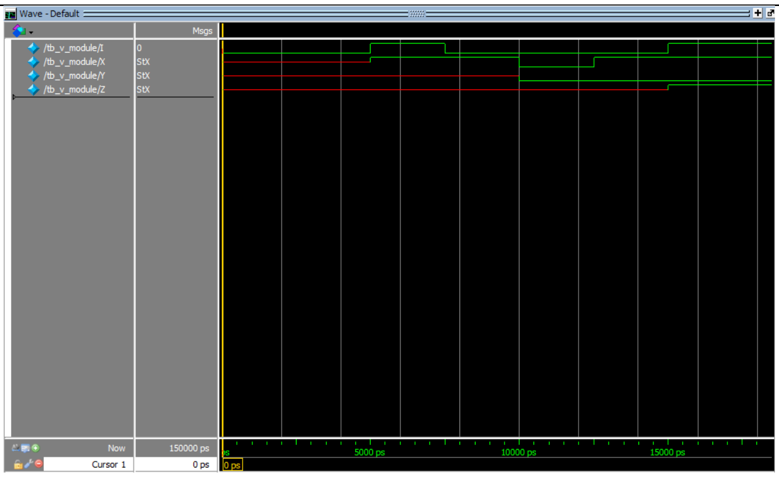


Figure 6.1: Output at t = 0ns, I = 0, X, Y, Z are unknown.

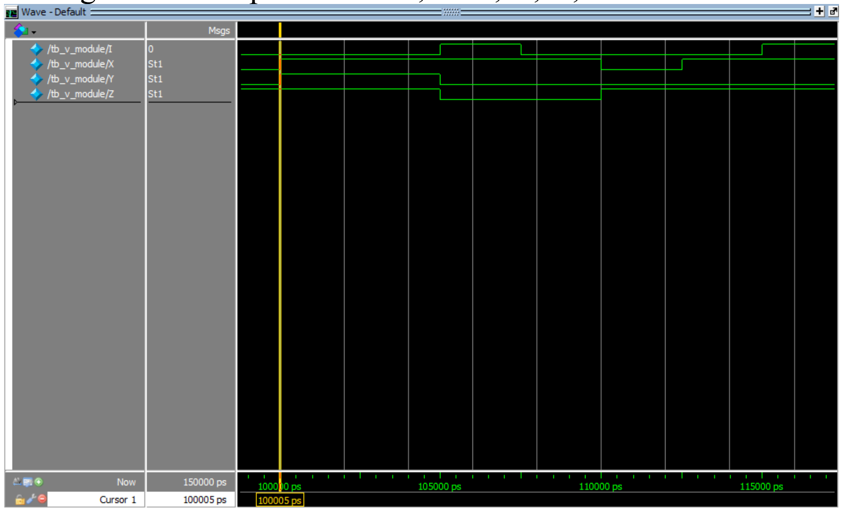


Figure 6.2: Output at 100ns, I = 0, X, Y, Z = 1.