Problem-1

- Source code

```
 //module declaration with inputs and outputs
module p1(Y,present,clock,reset,input_bit);
//Conventionally output is declared first, then input.
        output Y;
        input clock, reset,input_bit;
        output [2:0] present;
        reg [2:0] next;
        reg[2:0] present;

        parameter [2:0] S0=3'b000;
        parameter [2:0] S1=3'b001;
        parameter [2:0] S2=3'b010;
        parameter [2:0] S3=3'b011;
        parameter [2:0] S4=3'b100;

        initial
        begin
                present = 3'b000;
                next = 3'b000;
        end

        always @(posedge clock)
 // sequential logic
        begin
                if (reset) present <= S0;
                else present <= next;
```

```
                end
  // combinational logic
                always @(present or input_bit) // combinational logic
                begin
                        case(present)
                                S0: if (reset) next = S0;
                                                else if(input_bit) next = S1;
                                                else next = S0;
                                S1: if (reset) next = S0;
                                                else if(input_bit) next = S2;
                                                else next = S0;
                                S2: if (reset) next = S0;
                                                else if(input_bit) next = S2;
                                                else next = S3;
                                S3: if (reset) next = S0;
                                                else if(input_bit) next = S4;
                                                else next = S0;
                                S4: if (reset) next = S0;
                                                else if(input_bit) next = S2;
                                                else next = S0;
                        endcase

        end
        // output logic described using continuous assignment
        assign Y = (present == S4);
endmodule
```

- Test bench for p1.v

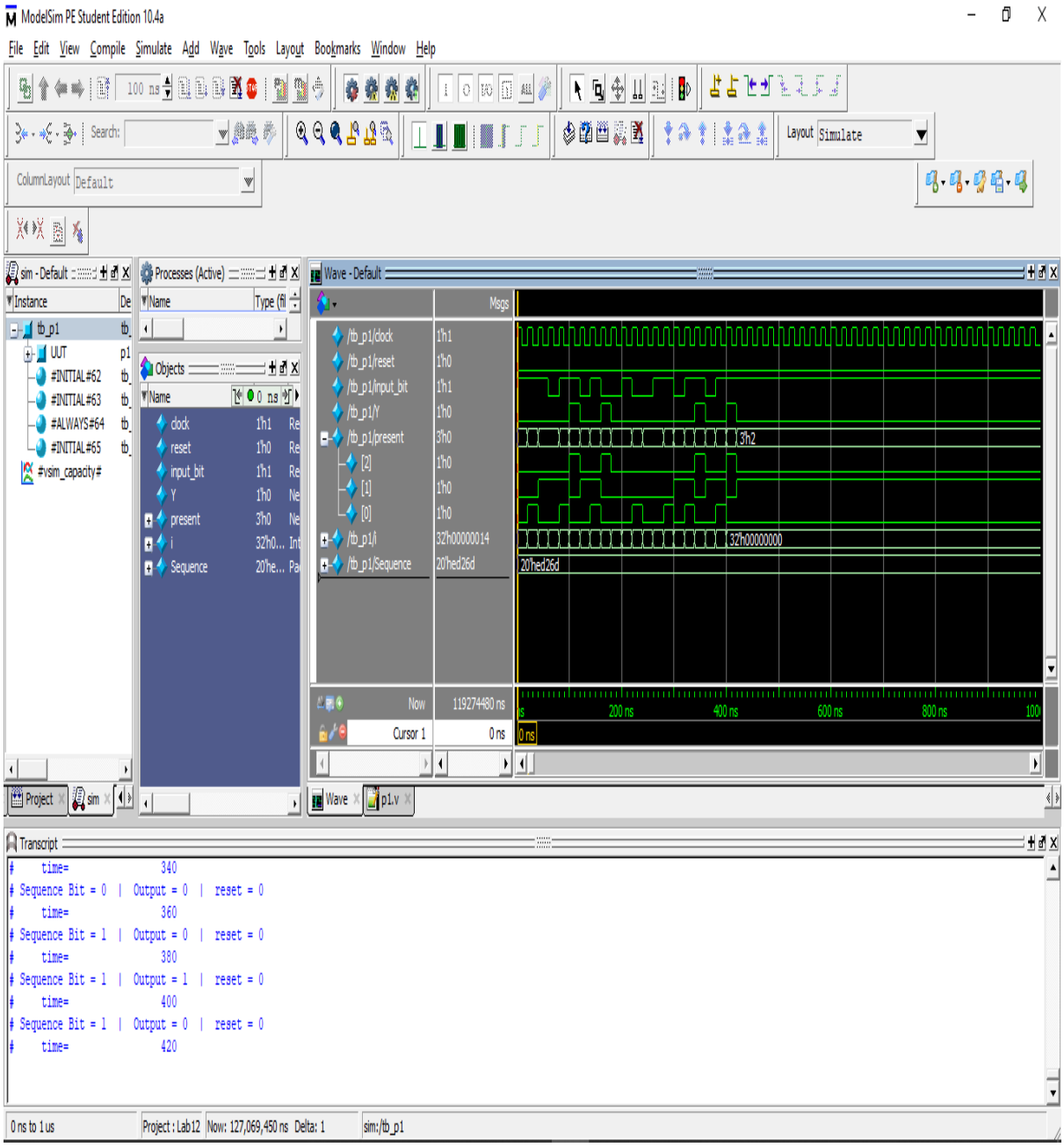
```
module tb_p1();
        reg clock, reset,input_bit;
        wire Y;
```

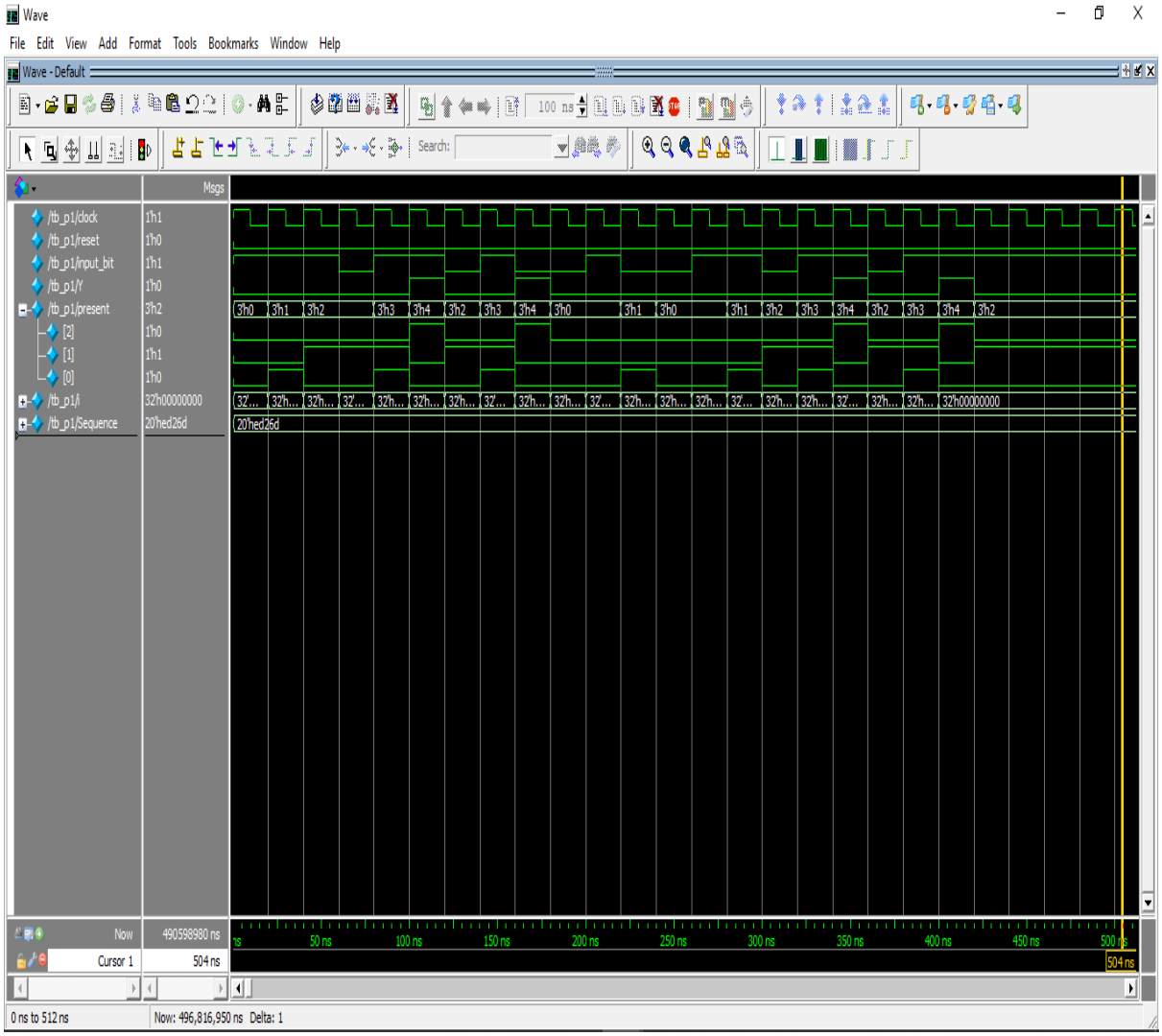
```
        wire[2:0] present;
        integer i;
        p1 UUT(Y,present,clock,reset,input_bit);
        reg[20:1] Sequence = 20'b11101101001001101101;
        initial reset = 1'b1;
        always #10 clock = ~clock;
        initial
                begin
                clock = 1'b1;
                reset = 1'b0;
                for (i=20;i>0;i=i-1)
                begin
                        input_bit = Sequence[i]; #20
                        $monitor("Sequence Bit = %b  |  Output = %b  |  reset = %b\n    time=",
input_bit, Y, reset, $time);
                end
        end
endmodule
```

- Transcript

```
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=              20
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=              40
# Sequence Bit = 0  |  Output = 0  |  reset = 0
#    time=              60
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=              80
# Sequence Bit = 1  |  Output = 1  |  reset = 0
#    time=             100
# Sequence Bit = 0  |  Output = 0  |  reset = 0
```

```
#    time=            120
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=            140
# Sequence Bit = 0  |  Output = 1  |  reset = 0
#    time=            160
# Sequence Bit = 0  |  Output = 0  |  reset = 0
#    time=            180
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=            200
# Sequence Bit = 0  |  Output = 0  |  reset = 0
#    time=            220
# Sequence Bit = 0  |  Output = 0  |  reset = 0
#    time=            240
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=            260
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=            280
# Sequence Bit = 0  |  Output = 0  |  reset = 0
#    time=            300
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=            320
# Sequence Bit = 1  |  Output = 1  |  reset = 0
#    time=            340
# Sequence Bit = 0  |  Output = 0  |  reset = 0
#    time=            360
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=            380
# Sequence Bit = 1  |  Output = 1  |  reset = 0
#    time=            400
# Sequence Bit = 1  |  Output = 0  |  reset = 0
#    time=            420
```


ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
sim - Default
Instance
tb_p1
UUT
#INITIAL#62
#INITIAL#63
#ALWAYS#64
#INITIAL#65
#vsim_capacity#
Processes (Active)
Objects
clock
reset
input_bit
Y
present
i
Sequence
Wave - Default
Msgs
/tb_p1/clock
/tb_p1/reset
/tb_p1/input_bit
/tb_p1/Y
/tb_p1/present
[2]
[1]
[0]
/tb_p1/i
/tb_p1/Sequence
3'h2
32'h00000000
20'hed26d
Now
119274480 ns
Cursor 1
0 ns
200 ns
400 ns
600 ns
800 ns
Transcript
time= 340
Sequence Bit = 0 | Output = 0 | reset = 0
time= 360
Sequence Bit = 1 | Output = 0 | reset = 0
time= 380
Sequence Bit = 1 | Output = 1 | reset = 0
time= 400
Sequence Bit = 1 | Output = 0 | reset = 0
time= 420
0 ns to 1 us
Project : Lab12
Now: 127,069,450 ns Delta: 1
sim:/tb_p1



Problem 2

- Source Code

```
module p2(Q,upper,reset,clk); //counter module with inputs and outputs
//conventionally outputs are declared first, then input.
        output reg [3:0]Q;
        output upper;
        input reset;
        input clk;
```

```
	always@(posedge clk)
	begin
		if(reset) //reset
			Q = 4'b0000;
		else
			Q = Q + 4'b0001; //increment by 1
	end
	assign upper = Q[3];
endmodule
```

- Test Bench

```
module tb_p2(); //Test Bench
	reg clk, reset;
	wire upper;
	wire [3:0]Q;
	p2 UUT(Q, upper, reset, clk);
		initial begin //clock definition
			clk = 0;
			forever #5 clk = ~clk;
		end
		initial begin //simulation
			$monitor("COUNT  = %b", Q, " | UPPER = %b", upper);
			reset = 1;
			#20;
			reset = 0;
    //stop after 500 time units
    #500 $finish;
		end
endmodule
```

- Transcript

COUNT = xxxx | UPPER = x

COUNT = 0000 | UPPER = 0

COUNT = 0001 | UPPER = 0

COUNT = 0010 | UPPER = 0

COUNT = 0011 | UPPER = 0

COUNT = 0100 | UPPER = 0

COUNT = 0101 | UPPER = 0

COUNT = 0110 | UPPER = 0

COUNT = 0111 | UPPER = 0

COUNT = 1000 | UPPER = 1

COUNT = 1001 | UPPER = 1

COUNT = 1010 | UPPER = 1

COUNT = 1011 | UPPER = 1

COUNT = 1100 | UPPER = 1

COUNT = 1101 | UPPER = 1

COUNT = 1110 | UPPER = 1

COUNT = 1111 | UPPER = 1

COUNT = 0000 | UPPER = 0

COUNT = 0001 | UPPER = 0

COUNT = 0010 | UPPER = 0

COUNT = 0011 | UPPER = 0

COUNT = 0100 | UPPER = 0

COUNT = 0101 | UPPER = 0

COUNT = 0110 | UPPER = 0

COUNT = 0111 | UPPER = 0

COUNT = 1000 | UPPER = 1

COUNT = 1001 | UPPER = 1

COUNT = 1010 | UPPER = 1

COUNT = 1011 | UPPER = 1

COUNT = 1100 | UPPER = 1

COUNT = 1101 | UPPER = 1

COUNT = 1110 | UPPER = 1

COUNT = 1111 | UPPER = 1

COUNT = 0000 | UPPER = 0

COUNT = 0001 | UPPER = 0

COUNT = 0010 | UPPER = 0

COUNT = 0011 | UPPER = 0

COUNT = 0100 | UPPER = 0

COUNT = 0101 | UPPER = 0

COUNT = 0110 | UPPER = 0

COUNT = 0111 | UPPER = 0

COUNT = 1000 | UPPER = 1

COUNT = 1001 | UPPER = 1

COUNT = 1010 | UPPER = 1

COUNT = 1011 | UPPER = 1

COUNT = 1100 | UPPER = 1

COUNT = 1101 | UPPER = 1

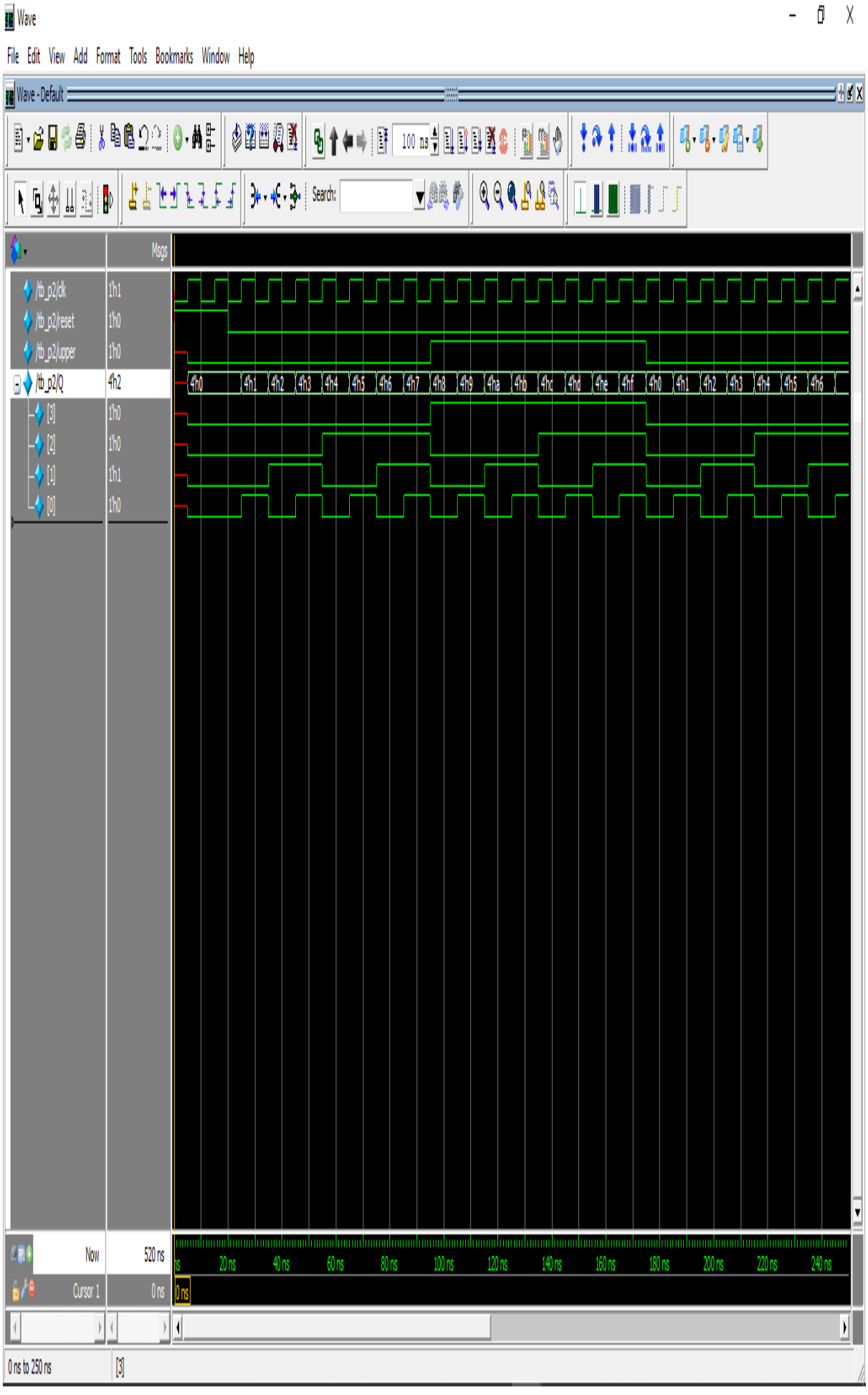
COUNT = 1110 | UPPER = 1

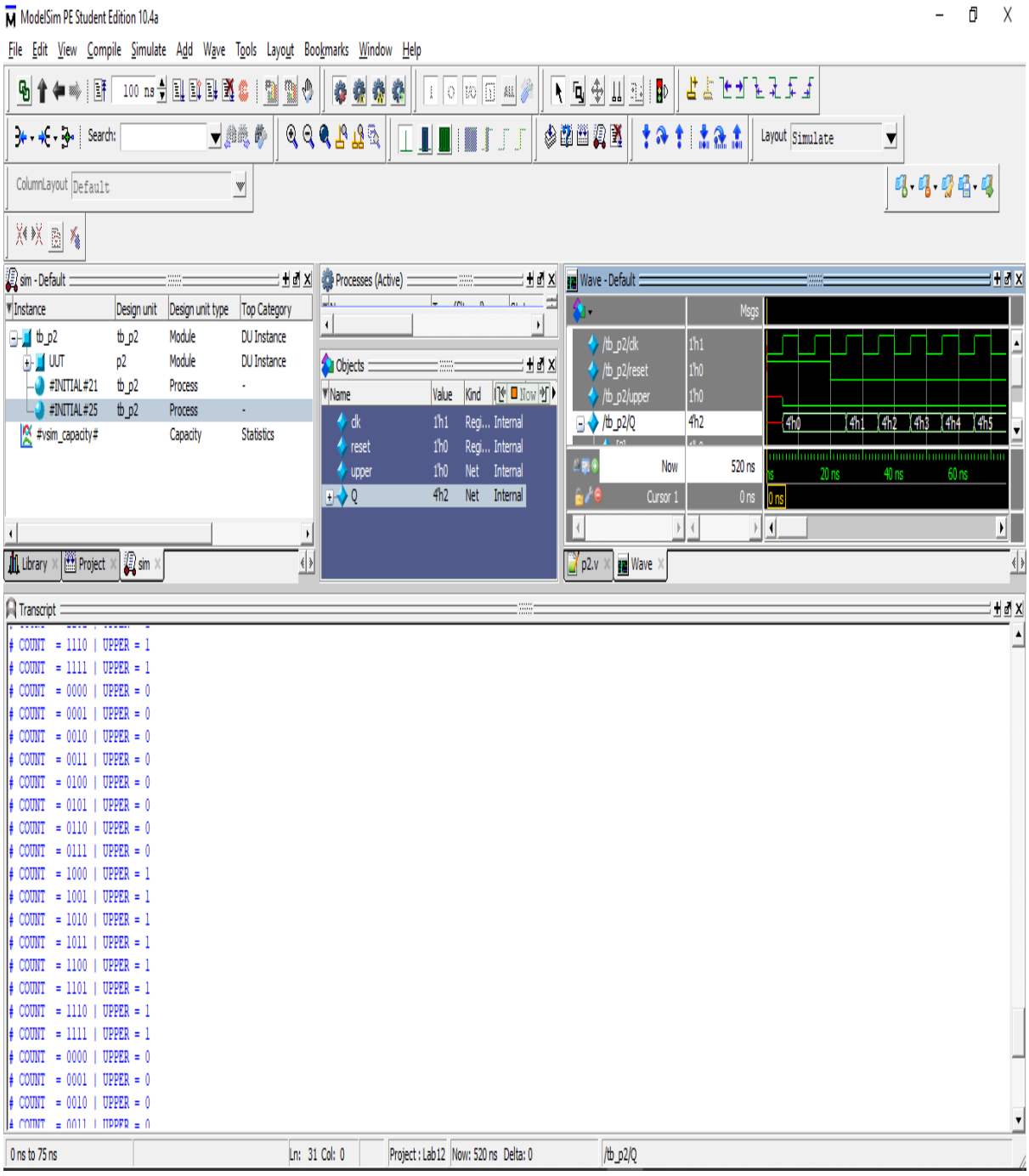
COUNT = 1111 | UPPER = 1

COUNT = 0000 | UPPER = 0

COUNT = 0001 | UPPER = 0

COUNT = 0010 | UPPER = 0


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/tb_p2/clk
/tb_p2/reset
/tb_p2/upper
/tb_p2/Q
[3]
[2]
[1]
[0]
1'h1
1'h0
1'h0
4'h2
1'h0
1'h0
1'h1
1'h0
4'h0 4'h1 4'h2 4'h3 4'h4 4'h5 4'h6 4'h7 4'h8 4'h9 4'ha 4'hb 4'hc 4'hd 4'he 4'hf 4'h0 4'h1 4'h2 4'h3 4'h4 4'h5 4'h6
Now
520 ns
Cursor 1
0 ns
0 ns
20 ns 40 ns 60 ns 80 ns 100 ns 120 ns 140 ns 160 ns 180 ns 200 ns 220 ns 240 ns
0 ns to 250 ns
[3]



Problem-3

- Source code

```
module p3(open,clk,reset, c0,c1);

 output[2:0] open;

 input clk, reset;
```

```
input c0,c1;
reg [2:0] Next;
reg[2:0]  state;
parameter [2:0] S0= 3'b000;
parameter [2:0] S1= 3'b001;
parameter [2:0] S2 = 3'b010;
parameter [2:0] S3=3'b011;
parameter [2:0] S4=3'b100;
parameter [2:0] S5=3'b101;
parameter [2:0] S6=3'b110;
parameter [2:0] S7=3'b111;

always @(posedge clk) // sequential
begin
if (reset) state <= S0;
else state <= Next;
end

always @(state,c0,c1) // combinational
begin
case(state)
S0: if (!c0 && c1)
                Next = S1;
        else if (c0 && !c1)
                Next = S7;
        else if(c0 && c1)
                Next = S2;
S1: if (!c0 && c1)
                Next = S2;
        else if (c0 && !c1)
                Next = S0;
```

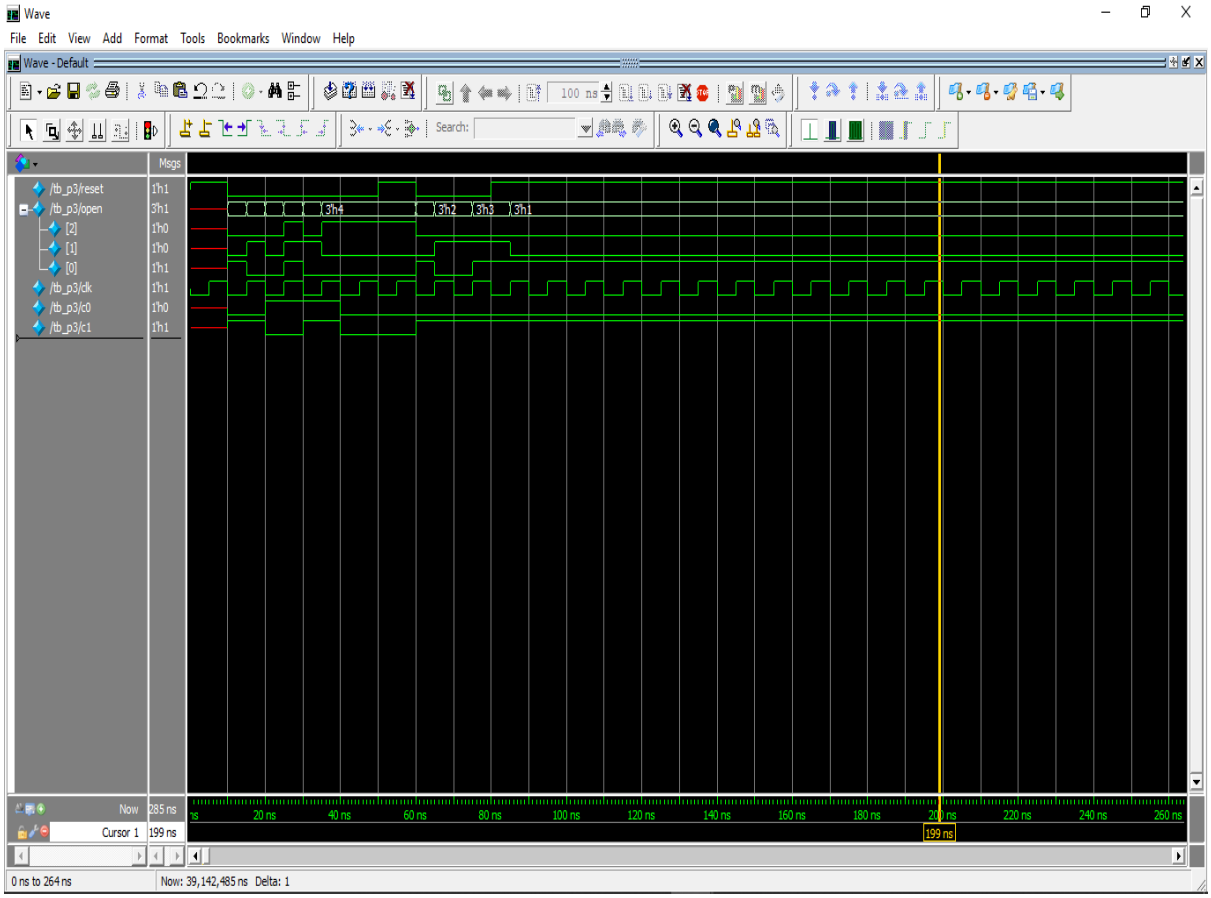
```
        else if(c0 && c1)
                Next = S3;

S2: if (!c0 && c1)
                Next = S3;
        else if (c0 && !c1)
                Next = S1;
        else if(c0 && c1)
                Next = S4;

S3: if (!c0 && c1)
                Next = S4;
        else if (c0 && !c1)
                Next = S2;
        else if(c0 && c1)
                Next = S5;

S4:if (!c0 && c1)
                Next = S5;
        else if (c0 && !c1)
                Next = S3;
        else if(c0 && c1)
                Next = S6;


S5:if (!c0 && c1)
                Next = S6;
        else if (c0 && !c1)
                Next = S4;
        else if(c0 && c1)
                Next = S7;
```

```
 S6:if (!c0 && c1)
                    Next = S7;
          else if (c0 && !c1)
                    Next = S5;
          else if(c0 && c1)
                    Next = S0;


 S7:if (!c0 && c1)
                    Next = S0;
          else if (c0 && !c1)
                    Next = S6;
          else if(c0 && c1)
                    Next = S1;

          endcase end
 // output logic described using continuous assignment

 assign open = Next;


endmodule
```

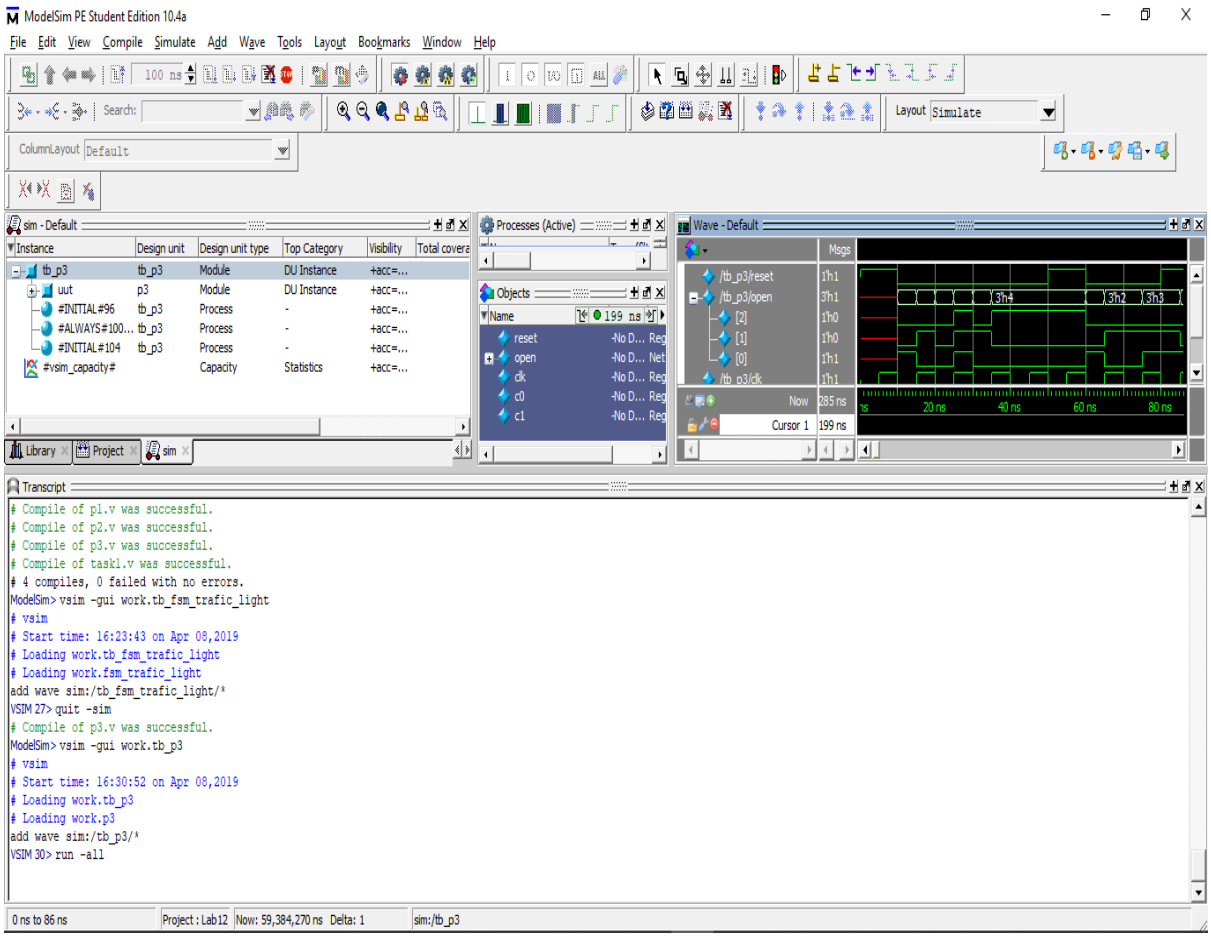
- Test Bench

```
module tb_p3();
 reg reset;
 wire[2:0] open;
 reg clk=0;
```

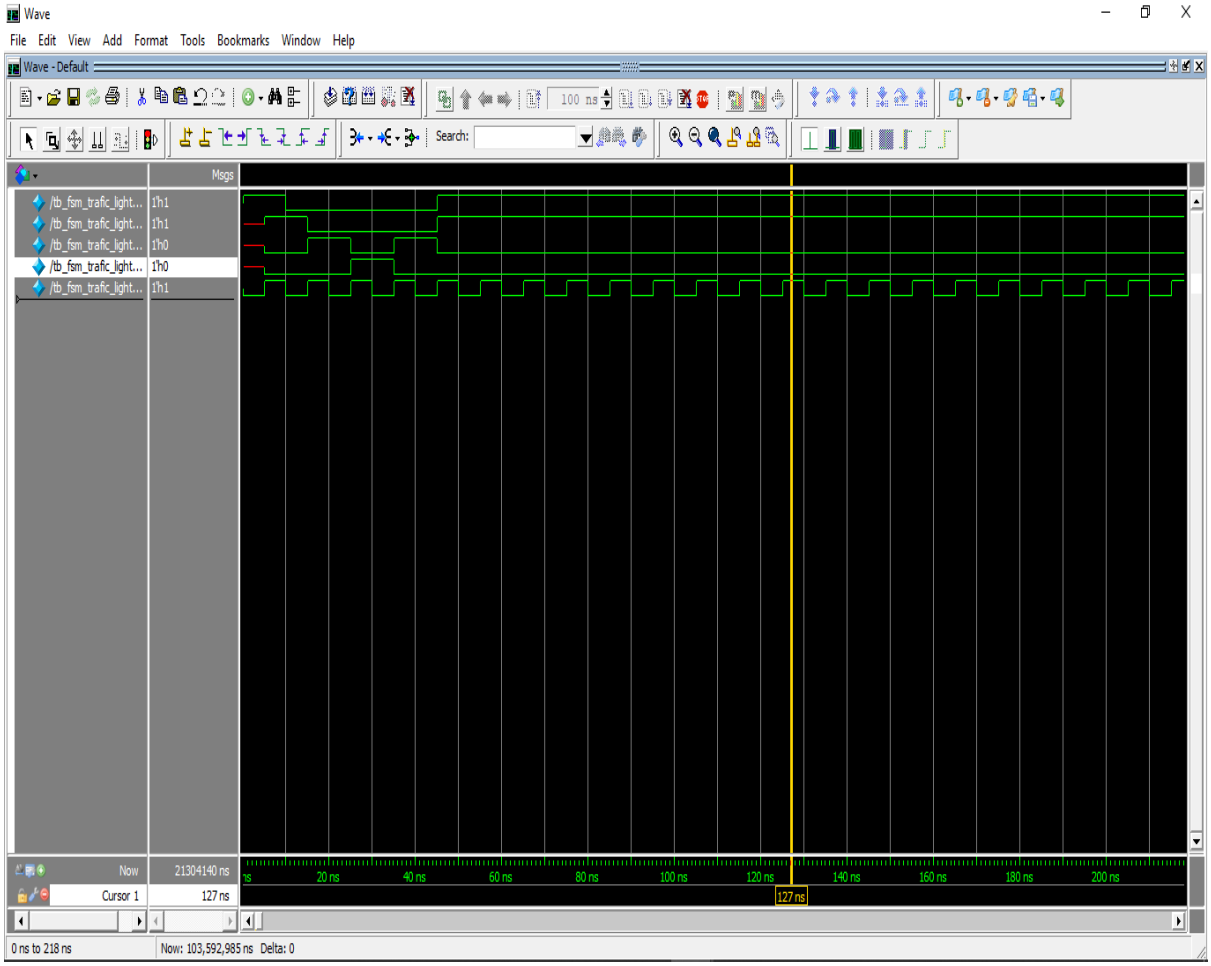
```
reg c0,c1;
p3 uut (open,clk,reset, c0,c1);

always #5 clk = ~clk;

initial
        begin
                reset=1'b1; #10;
                reset=1'b0; c0=1'b0;c1=1'b1;#10;
                reset=1'b0; c0=1'b1;c1=1'b0;#10;
                reset=1'b0; c0=1'b1;c1=1'b1;#10;
                reset=1'b0; c0=1'b0;c1=1'b0;#10;
                reset=1'b1; #10;
                reset=1'b0; c0=1'b0;c1=1'b1;#10;
                reset=1'b0; c0=1'b0;c1=1'b1;#10;
                reset=1'b1; #10;
        end
endmodule
```


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/tb_p3/reset
/tb_p3/open
[2]
[1]
[0]
/tb_p3/clk
/tb_p3/c0
/tb_p3/c1
1'h1
3'h1
1'h0
1'h0
1'h1
1'h1
1'h0
1'h1
3'h4
3'h2
3'h3
3'h1
Now
285 ns
Cursor 1
199 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
160 ns
180 ns
200 ns
220 ns
240 ns
260 ns
199 ns
0 ns to 264 ns
Now: 39,142,485 ns Delta: 1

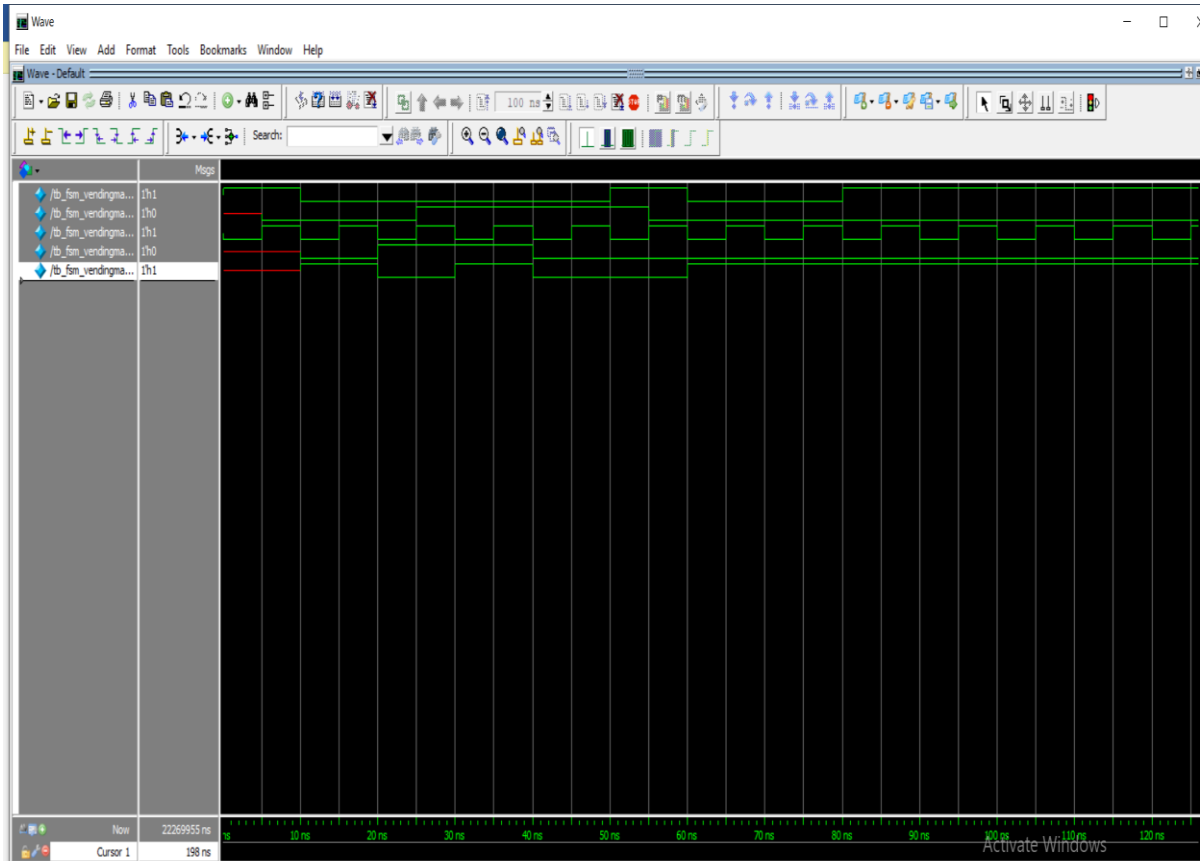
-


ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type Top Category Visibility
tb_p3 tb_p3 Module DU Instance +acc=...
uut p3 Module DU Instance +acc=...
#INITIAL#96 tb_p3 Process - +acc=...
#ALWAYS#100... tb_p3 Process - +acc=...
#INITIAL#104 tb_p3 Process - +acc=...
#vsim_capacity# Capacity Statistics +acc=...
Processes (Active)
Objects
Name
reset
open
clk
c0
c1
Wave - Default
Msgs
/tb_p3/reset
/tb_p3/open
[2]
[1]
[0]
/tb_p3/clk
Now
Cursor 1
20 ns
40 ns
60 ns
80 ns
Transcript
Compile of p1.v was successful.
Compile of p2.v was successful.
Compile of p3.v was successful.
Compile of task1.v was successful.
4 compiles, 0 failed with no errors.
ModelSim> vsim -gui work.tb_fsm_trafic_light
vsim
Start time: 16:23:43 on Apr 08,2019
Loading work.tb_fsm_trafic_light
Loading work.fsm_trafic_light
add wave sim:/tb_fsm_trafic_light/*
VSIM 27> quit -sim
Compile of p3.v was successful.
ModelSim> vsim -gui work.tb_p3
vsim
Start time: 16:30:52 on Apr 08,2019
Loading work.tb_p3
Loading work.p3
add wave sim:/tb_p3/*
VSIM 30> run -all
0 ns to 86 ns
Project : Lab12
Now: 59,384,270 ns Delta: 1
sim:/tb_p3

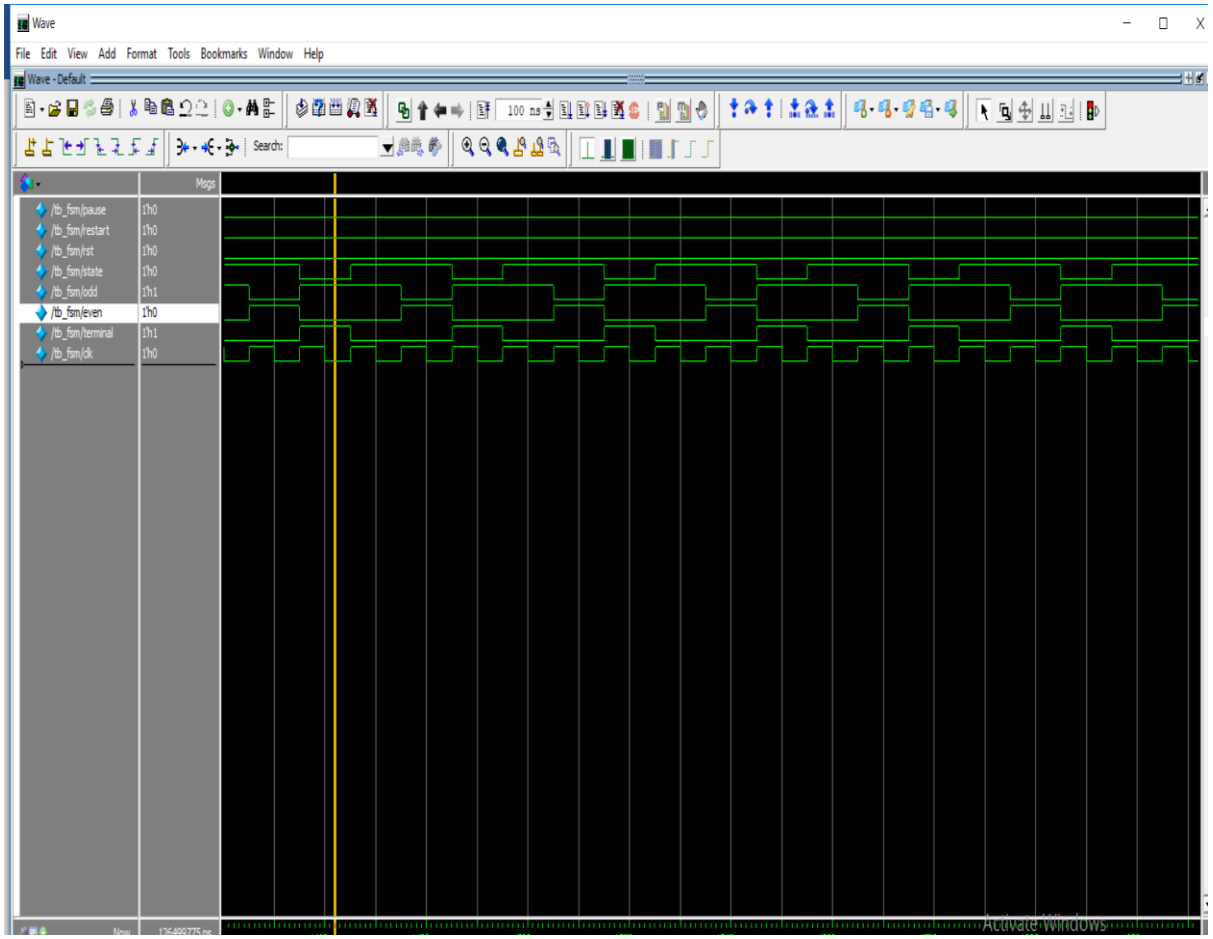
Task-1


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/tb_fsm_trafic_light...
1'h1
1'h0
Now
21304140 ns
Cursor 1
127 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
160 ns
180 ns
200 ns
0 ns to 218 ns
Now: 103,592,985 ns Delta: 0

TASK-2


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/tb_fsm_vendingma... 1'h1
/tb_fsm_vendingma... 1'h0
/tb_fsm_vendingma... 1'h1
/tb_fsm_vendingma... 1'h0
/tb_fsm_vendingma... 1'h1
Now 22269955 ns
Cursor 1 198 ns
10 ns 20 ns 30 ns 40 ns 50 ns 60 ns 70 ns 80 ns 90 ns 120 ns
Activate Windows

TASK-3


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/tb_fsm/pause
/tb_fsm/restart
/tb_fsm/rst
/tb_fsm/state
/tb_fsm/odd
/tb_fsm/even
/tb_fsm/terminal
/tb_fsm/clk
1'h0
1'h0
1'h0
1'h0
1'h1
1'h0
1'h1
1'h0
Activate Windows