EC-204

<LAB - 8>

NITK SURATHKAL

INBASEKARAN.P

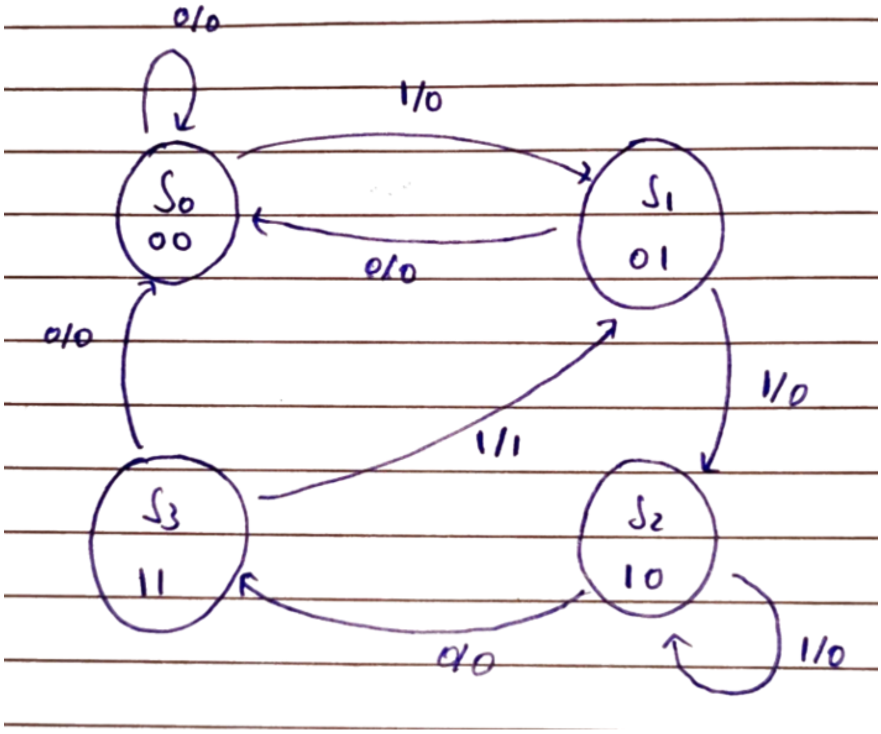
201EC226

Prof: Sumam S

1. Sequence 1101 detector

Solution:

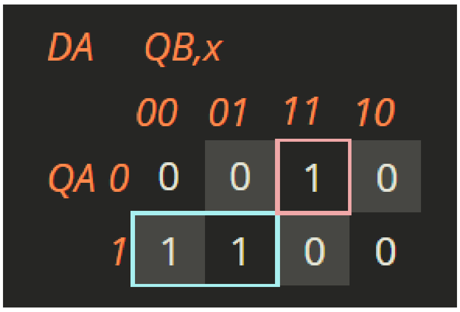
State Diagram



State Table

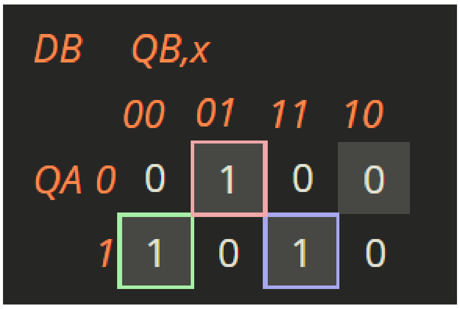
Present State		i/p	Next State		o/p	Excitation (DFF)		Excitation (JKFF)	
Q_A	Q_B	x	Q_A^+	Q_B^+	y	D_A	D_B	J_AK_A (T_A)	J_BK_B (T_B)
0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1	0	1
0	1	0	0	0	0	0	0	0	1
0	1	1	1	0	0	1	0	1	1
1	0	0	1	1	0	1	1	0	1
1	0	1	1	0	0	1	0	0	0
1	1	0	0	0	0	0	0	1	1
1	1	1	0	1	1	0	1	1	0

K Map



DA	QB,x			
	00	01	11	10
QA 0	0	0	1	0
1	1	1	0	0

DA (QA, QB, x) = QA'QBx + QAQB'

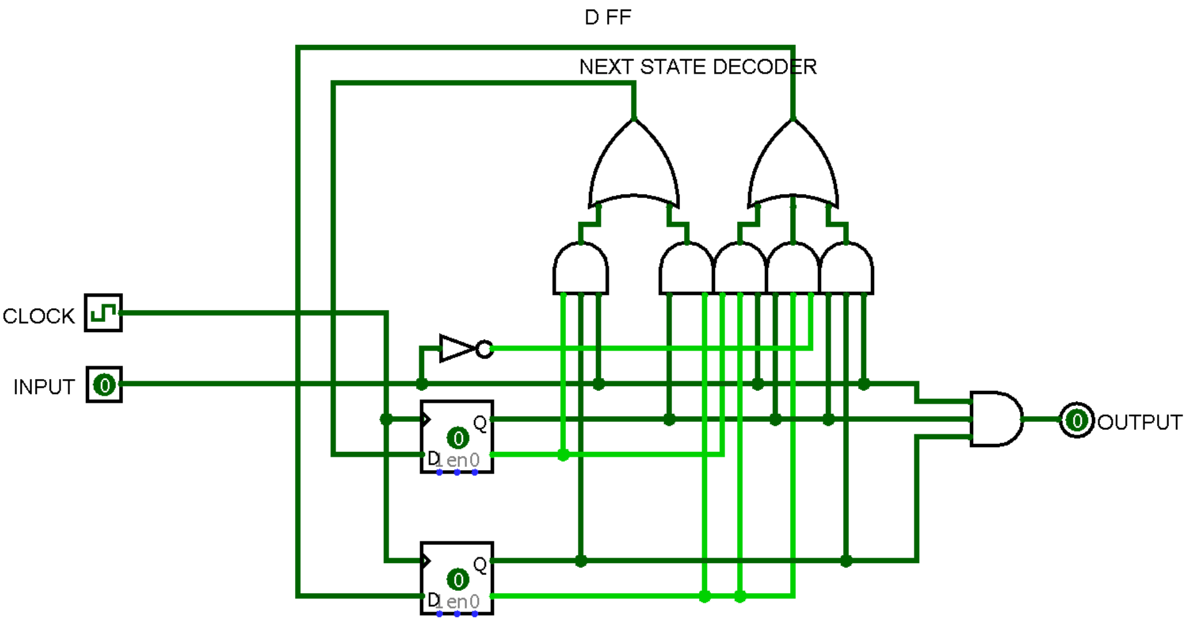


DB	QB,x			
	00	01	11	10
QA 0	0	1	0	0
1	1	0	1	0

DB (QA, QB, x) = QA'QB'x + QAQB'x' + QAQBx

a) Using D FF

Circuit diagram

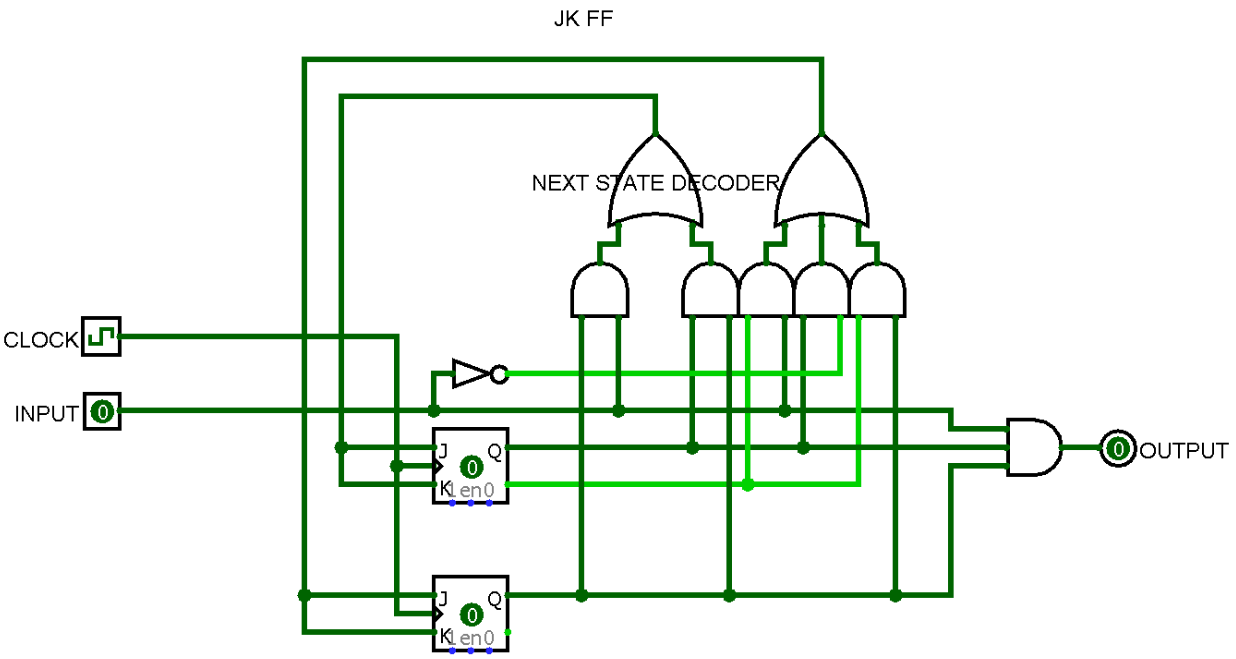


Simulation

Video uploaded on Moodle

b) Using JK FF

Circuit diagram



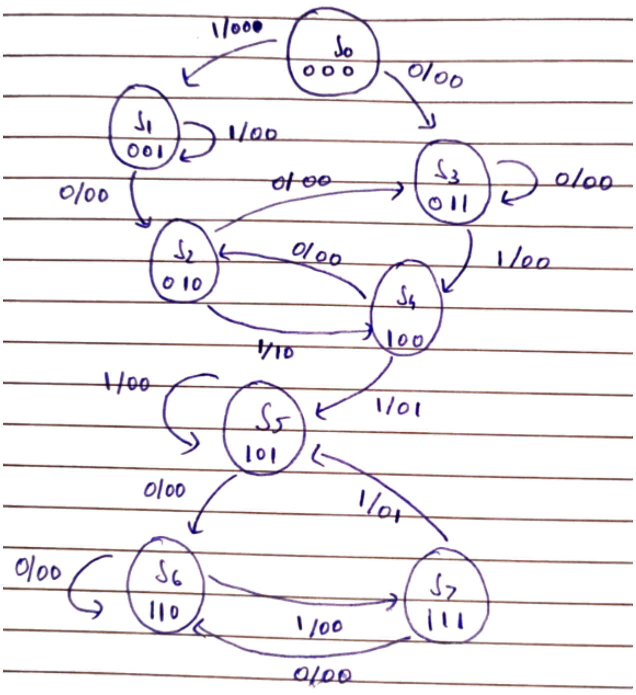
Simulation

Video uploaded on Moodle

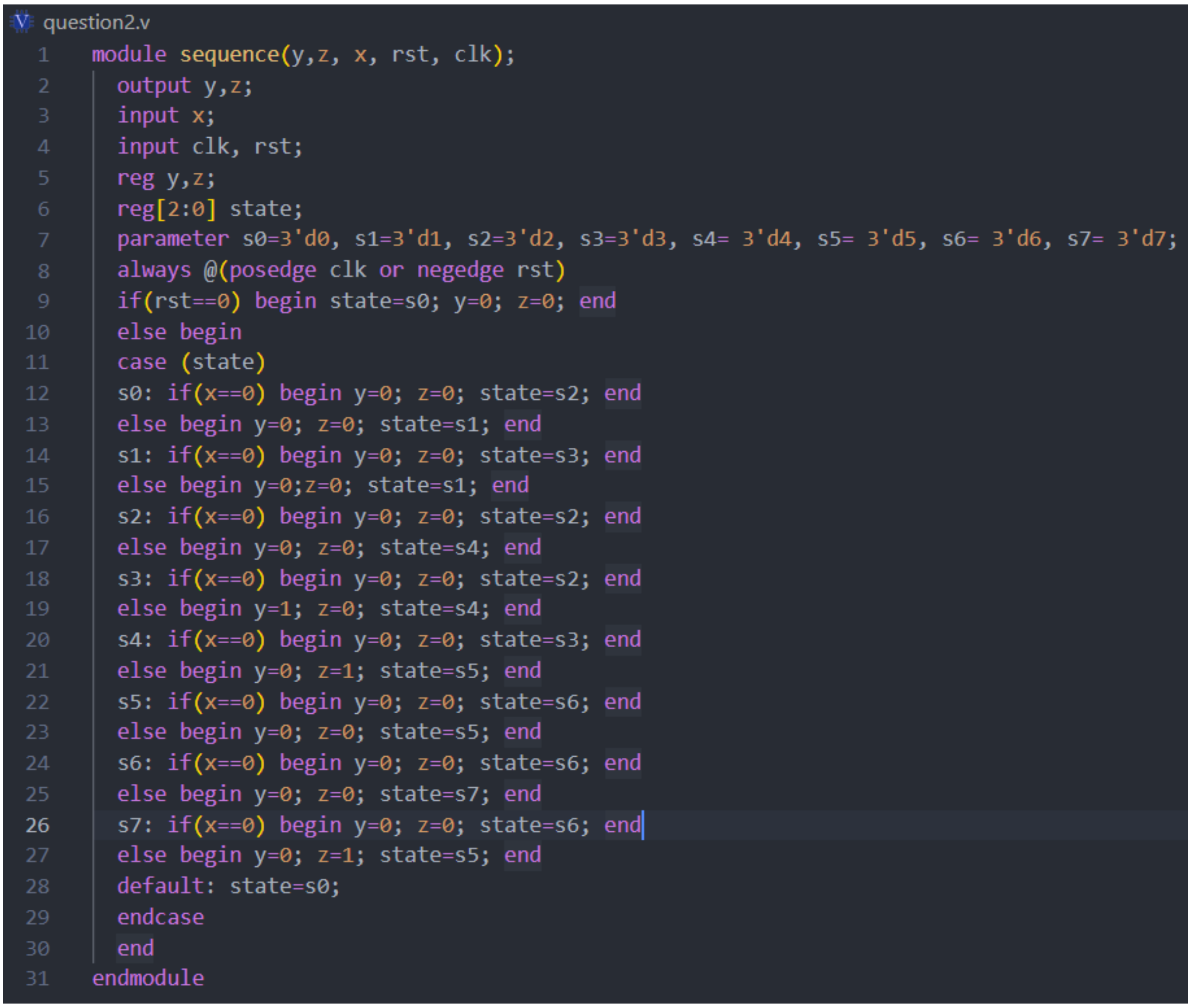
2. A digital system has one input X and two outputs Y and Z.

Solution:

State Diagram



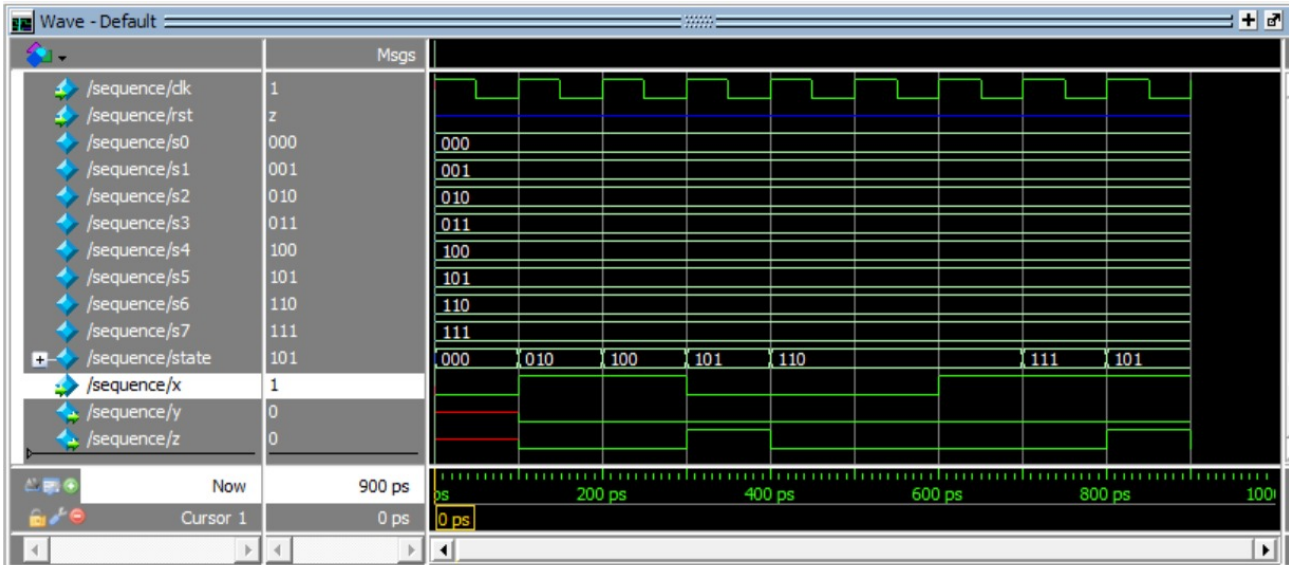
Verilog code



question2.v

```verilog
module sequence(y,z, x, rst, clk);
  output y,z;
  input x;
  input clk, rst;
  reg y,z;
  reg[2:0] state;
  parameter s0=3'd0, s1=3'd1, s2=3'd2, s3=3'd3, s4= 3'd4, s5= 3'd5, s6= 3'd6, s7= 3'd7;
  always @(posedge clk or negedge rst)
  if(rst==0) begin state=s0; y=0; z=0; end
  else begin
  case (state)
  s0: if(x==0) begin y=0; z=0; state=s2; end
  else begin y=0; z=0; state=s1; end
  s1: if(x==0) begin y=0; z=0; state=s3; end
  else begin y=0;z=0; state=s1; end
  s2: if(x==0) begin y=0; z=0; state=s2; end
  else begin y=0; z=0; state=s4; end
  s3: if(x==0) begin y=0; z=0; state=s2; end
  else begin y=1; z=0; state=s4; end
  s4: if(x==0) begin y=0; z=0; state=s3; end
  else begin y=0; z=1; state=s5; end
  s5: if(x==0) begin y=0; z=0; state=s6; end
  else begin y=0; z=0; state=s5; end
  s6: if(x==0) begin y=0; z=0; state=s6; end
  else begin y=0; z=0; state=s7; end
  s7: if(x==0) begin y=0; z=0; state=s6; end
  else begin y=0; z=1; state=s5; end
  default: state=s0;
  endcase
  end
endmodule
```

Simulation



Wave - Default

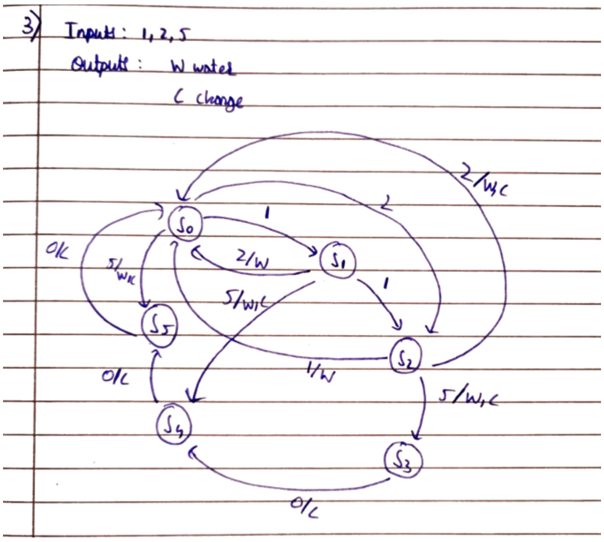
```
                   Msgs
/sequence/clk      1
/sequence/rst      z
/sequence/s0       000
/sequence/s1       001
/sequence/s2       010
/sequence/s3       011
/sequence/s4       100
/sequence/s5       101
/sequence/s6       110
/sequence/s7       111
/sequence/state    101
/sequence/x        1
/sequence/y        0
/sequence/z        0

Now                900 ps
Cursor 1           0 ps
```

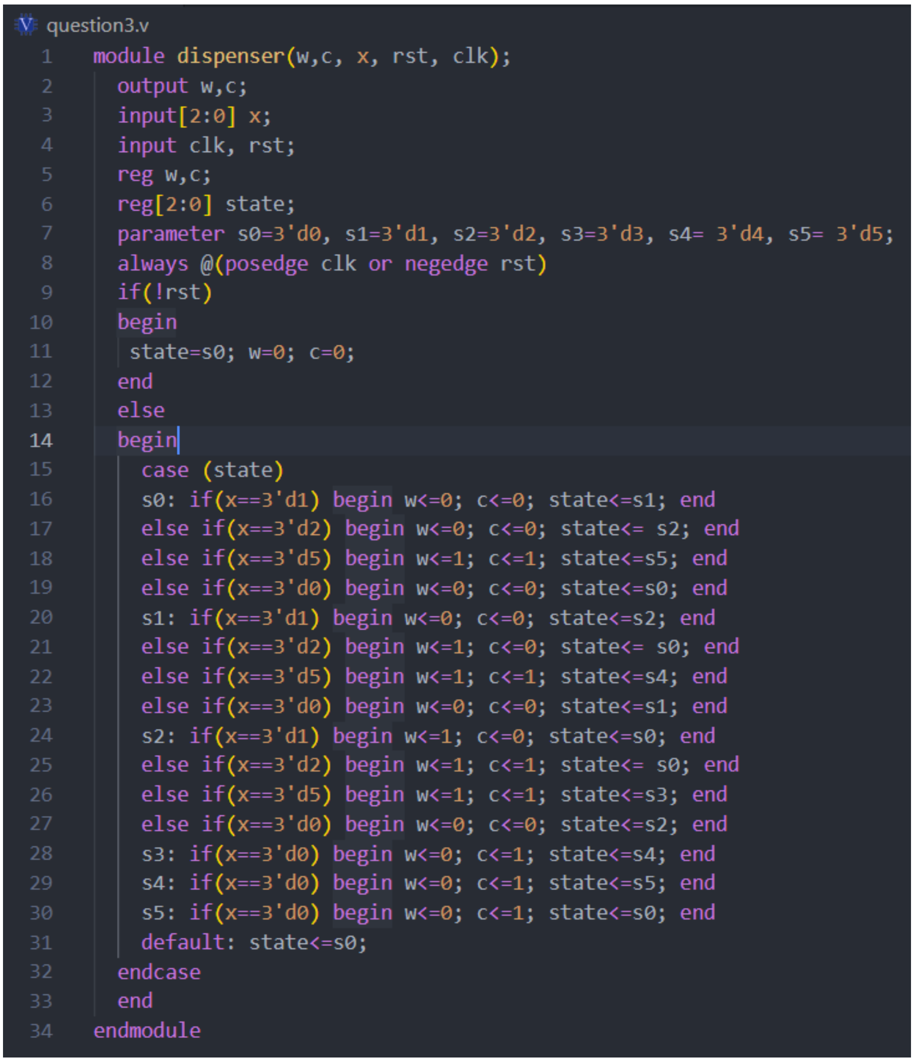
3. Vending machine

Solution:

State Diagram



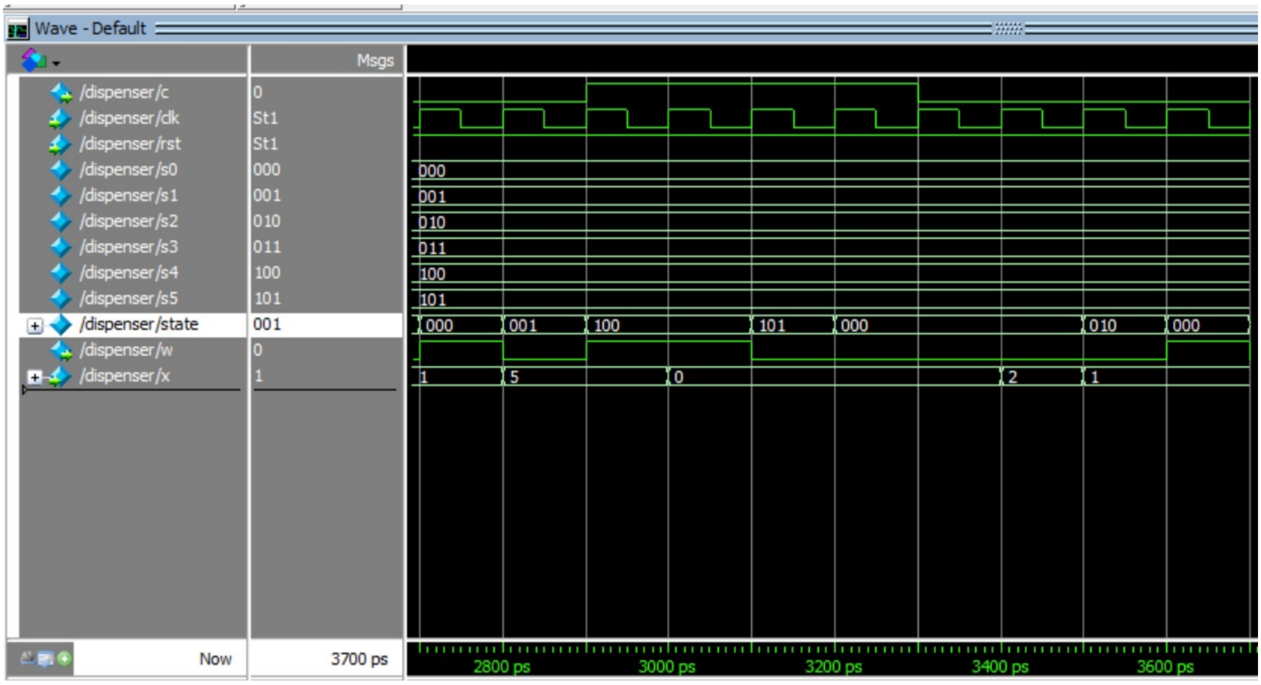
Verilog code



question3.v

```
module dispenser(w,c, x, rst, clk);
  output w,c;
  input[2:0] x;
  input clk, rst;
  reg w,c;
  reg[2:0] state;
  parameter s0=3'd0, s1=3'd1, s2=3'd2, s3=3'd3, s4= 3'd4, s5= 3'd5;
  always @(posedge clk or negedge rst)
  if(!rst)
  begin
   state=s0; w=0; c=0;
  end
  else
  begin
    case (state)
    s0: if(x==3'd1) begin w<=0; c<=0; state<=s1; end
    else if(x==3'd2) begin w<=0; c<=0; state<= s2; end
    else if(x==3'd5) begin w<=1; c<=1; state<=s5; end
    else if(x==3'd0) begin w<=0; c<=0; state<=s0; end
    s1: if(x==3'd1) begin w<=0; c<=0; state<=s2; end
    else if(x==3'd2) begin w<=1; c<=0; state<= s0; end
    else if(x==3'd5) begin w<=1; c<=1; state<=s4; end
    else if(x==3'd0) begin w<=0; c<=0; state<=s1; end
    s2: if(x==3'd1) begin w<=1; c<=0; state<=s0; end
    else if(x==3'd2) begin w<=1; c<=1; state<= s0; end
    else if(x==3'd5) begin w<=1; c<=1; state<=s3; end
    else if(x==3'd0) begin w<=0; c<=0; state<=s2; end
    s3: if(x==3'd0) begin w<=0; c<=1; state<=s4; end
    s4: if(x==3'd0) begin w<=0; c<=1; state<=s5; end
    s5: if(x==3'd0) begin w<=0; c<=1; state<=s0; end
    default: state<=s0;
  endcase
  end
endmodule
```

Simulation waveform

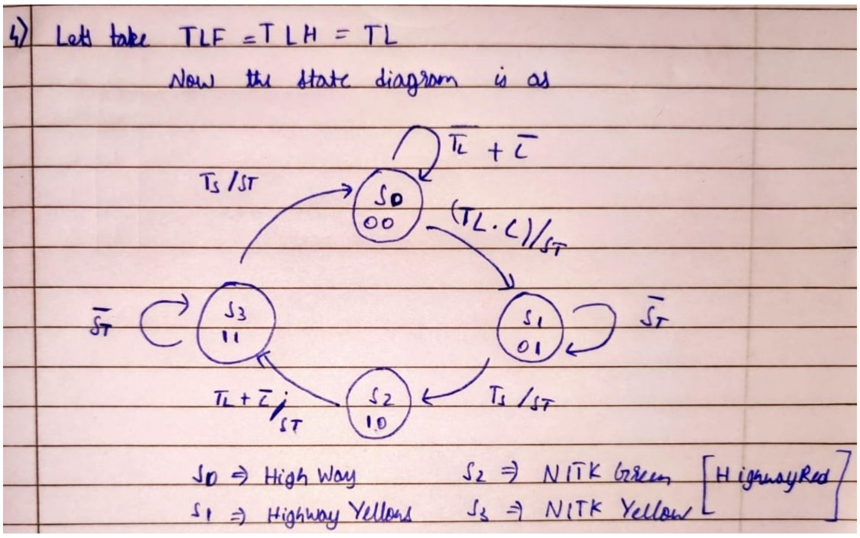


```
Wave - Default
                    Msgs
/dispenser/c        0
/dispenser/clk      St1
/dispenser/rst      St1
/dispenser/s0       000
/dispenser/s1       001
/dispenser/s2       010
/dispenser/s3       011
/dispenser/s4       100
/dispenser/s5       101
/dispenser/state    001
/dispenser/w        0
/dispenser/x        1
Now                 3700 ps
```

4. Traffic controller

Solution:

State Diagram



4) Lets take TLF = TLH = TL

Now the state diagram is as

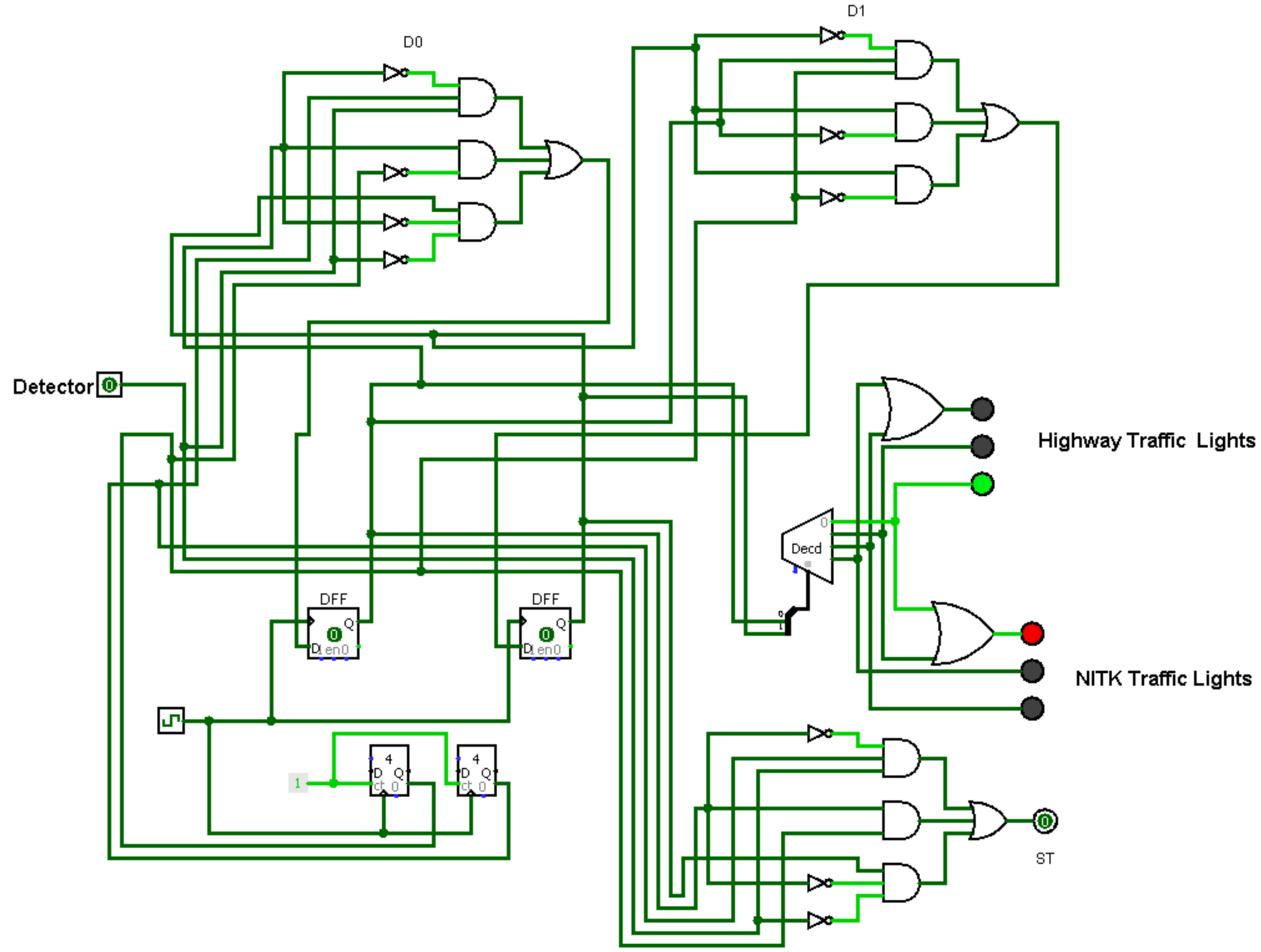
S_0 ⇒ High Way S_2 ⇒ NITK Green [Highway Red]

S_1 ⇒ Highway Yellows S_3 ⇒ NITK Yellow

State Table

Present State		i/p			Next State		o/p	Excitation (DFF)	
Q_A	Q_B	TL	C	TS	Q_A^+	Q_B^+	ST	D_A	D_B
0	0	0	0	x	0	0	0	0	0
0	0	0	1	x	0	0	0	0	0
0	0	1	0	x	0	0	0	0	0
0	0	1	1	x	0	1	1	0	1
0	1	x	x	0	0	1	0	0	1
0	1	x	x	1	1	0	1	1	0
1	0	0	0	x	1	1	1	1	1
1	0	0	1	x	1	0	0	1	0
1	0	1	0	x	1	1	1	1	1
1	0	1	1	x	1	1	1	1	1
1	1	x	x	0	1	1	0	1	1
1	1	x	x	1	1	0	1	1	0

Circuit diagram



Simulation

Video uploaded on Moodle