Hacettepe University

Computer Engineering Department

BM233 Logic Design Lab - 2020 Fall

Verilog Project - Resit

February 1, 2021

Student name:
İlayda Atmaca

Student Number:
b21827101

1 Problem Definition

In this experiment, SCP-079 is a microcomputer and at first it was just a student's project. but it gained sentience with time, reached an uncontrollable point, now it has a purpose .SCP-079 needs the attack system in verilog to achieve its malicious goals. and we're designing this verilog. While carrying out this plan, we take our steps by thinking about the possible good and bad scenarios.

2 Draw and include the Mealy state transition diagram

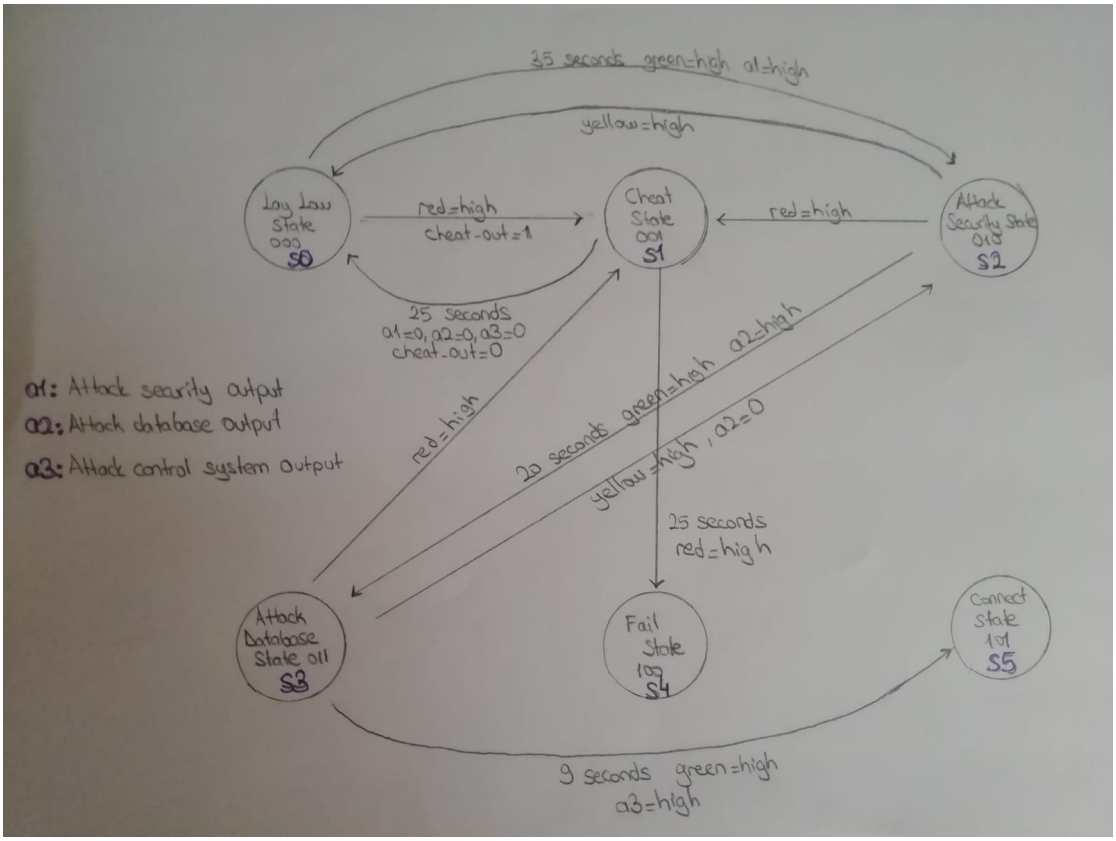

Figure 1: Mealy Diagram

3 Verilog code

Include the Verilog code for the system with clear comments and explanations of the code parts.

```
1
2
3  `timescale 1 s / 1s
4
5  module scp_079(input green,input yellow,input red,input clock,
6  output [2:0] state,input [5:0]timer,
7  output reg a1,output reg a2,output reg  a3,output reg cheat_out);
8  //a1:attack security output
9  //a2:attack database output
10 //a3:attack control system output
11 reg [2:0]state;
12 reg [2:0]next_state=3'b000;
13
14 parameter s0 =3'b000, s1 = 3'b001 , s2 =3'b010,
15           s3 =3'b011, s4 = 3'b100, s5= 3'b101;
16           //s0:lay low state s1:cheat state s2:attack security state
17           //s3:attack database state s4:fail state s5:connect state
18
19
20 always@(posedge clock)begin
21 state<=next_state;
22 end
23
24 always @(state,red,green,yellow,timer)begin
25 case(state)
26
27 s0:
28 if(red ==1) begin
29 next_state =s1;
30 cheat_out=1;
31 end
32
33 else if(green == 1 && red==0 && yellow==0 && timer>=6'b100011)begin
34 next_state = s2;
35 a1=1 ;
36 end
37
38 else if(yellow == 1 && red==0) begin
39 a1=0;a2=0;a3=0;
40 next_state = s0;
41 end
42
43 s1: if(red ==1 && timer>=6'b011001) begin
44 next_state = s4;
```

```
45  end
46
47  else if(red==0 && timer>=6'b011001)begin
48      next_state = s0;
49      a1=0; a2=0; a3=0;
50      cheat_out=0;
51      end
52
53  s2: if(green == 1 && red==0 && yellow==0 && timer==6'b010100) begin
54  next_state = s3;
55  a2 = 1;
56  end
57  else if(red ==1) begin
58  cheat_out=1;
59  next_state =s1;
60  end
61  else if(yellow ==1 && red==0) begin
62  next_state =s0;
63  a1 =0;
64  end
65
66  s3:
67  if(red==1) begin
68   cheat_out=1;
69   next_state=s1;
70  end
71
72  else if(yellow == 1 && red==0) begin
73  a2 =1;
74  next_state = s2;
75  end
76
77  else if(green == 1 && red==0 && yellow==0 && timer == 6'b001001)begin
78   a3 =1;
79   next_state = s5;
80  end
81  s4:begin end
82
83  s5:begin end
84  default begin a1=0; a2=0; a3=0; cheat_out =0; state = 3'b000; end
85
86  endcase
87  end
88  endmodule
```

```
1
2  `timescale 1 s / 1 s
3  module allok_tb;
4  reg green;
5  reg yellow;
6  reg red;
7  reg clock;
8  wire [2:0] state;
9  reg [5:0]timer;
10 wire a1;
11 wire a2;
12 wire a3;
13 wire cheat_out;
14
15 //Instantiate scp079
16 scp_079 s(.green(green),.yellow(yellow),.red(red),.clock(clock),.state(state),
17 .a1(a1),.a2(a2),.a3(a3),.timer(timer),.cheat_out(cheat_out));
18
19
20 initial begin
21     green=1;yellow=0;red=0; timer=0;
22     //Initial color situations
23     clock=1;
24     #0.5 clock  = 1;
25
26  end
27 //Generate a clock with period
28   always begin clock = ~clock;
29    #0.5;
30    end
31
32 initial begin//Test sequence
33 for(integer i=0;i<35;i=i+1) //state0
34     begin timer=timer+1; #1;end
35
36     timer=0;
37
38 for(integer i=0;i<20;i=i+1)//state2
39     begin timer=timer+1; #1;end
40
41     timer=0;
42 for(integer i=0;i<9;i=i+1)//state3
43     begin timer=timer+1; #1 ; end
44
45 timer=0;
46 for(integer i=0;i<11;i=i+1)//state5
47     begin timer=timer+1;#1;end
48     $finish;
```

```
end
endmodule
```

```


module a1trouble_tb;

initial begin

  green=1;yellow=0;red=0; timer=0;//Initial color situations
  clock=1;
  #0.5 clock  = 1;

 end

//Generate a clock with period
  always begin clock = ~clock;
   #0.5;
   end

initial begin//Test sequence
for(integer i=0;i<35;i=i+1) //state0
    begin timer=timer+1; #1;end

timer=0;
for(integer i=0;i<5;i=i+1)//state2
    begin timer=timer+1; #1;end

 green=0; red =0; yellow=1;timer=0;
for(integer i=0;i<35;i=i+1)//state0
    begin timer=timer+1; #1 ; end

green=1; red =0; yellow=0;#1; timer=0;

for(integer i=0;i<20;i=i+1)//state2
    begin timer=timer+1;#1;end

timer=0;
for(integer i=0;i<9;i=i+1)//state3
    begin timer=timer+1; #1 ; end

timer=0;
for(integer i=0;i<8;i=i+1)//state5
    begin timer=timer+1; #1 ; end


$finish;
```

```
 end
endmodule


module a2trouble_tb;

initial begin
  green=1;yellow=0;red=0; timer=0;//Initial color situations
  clock=1;
  #0.5 clock  = 1;

 end

//Generate a clock with period
  always begin clock = ~clock;
   #0.5;
   end

initial begin  //Test sequence
for(integer i=0;i<35;i=i+1) //state0
    begin timer=timer+1; #1;end
    timer=0;

for(integer i=0;i<20;i=i+1)//state2
    begin timer=timer+1; #1;end

    timer=0;
for(integer i=0;i<5;i=i+1)//state3
    begin timer=timer+1; #1 ; end

timer=0;green=0;yellow=1;
for(integer i=0;i<35;i=i+1)//state0
    begin timer=timer+1; #1;end

green=1;yellow=0; #1;timer=0;
for(integer i=0;i<20;i=i+1)//state2
    begin timer=timer+1; #1;end

    timer=0;
for(integer i=0;i<9;i=i+1)//state3
    begin timer=timer+1; #1 ; end

timer=0;
for(integer i=0;i<4;i=i+1)//state5
    begin timer=timer+1;#1;end

$finish;
```

```
 end
endmodule
```

```
module cheatsuccess_tb;

initial begin
  green=1;yellow=0;red=0; timer=0;//Initial color situations
  clock=1;
  #0.5 clock  = 1;

 end

//Generate a clock with period
  always begin clock = ~clock;
   #0.5;
   end

initial begin//Test sequence
for(integer i=0;i<35;i=i+1) //state0
    begin timer=timer+1; #1;end
    timer=0;

for(integer i=0;i<5;i=i+1)//state2
    begin timer=timer+1; #1;end

 green=0;yellow=0;red=1; timer=0;
for(integer i=0;i<25;i=i+1)begin//state1
if(timer==24)begin
red=0;
end
timer=timer+1;#1;end

green=1;yellow=0;red=0; timer=0;
for(integer i=0;i<35;i=i+1)begin//state0
if(timer==34)begin
green=1;end
timer=timer+1;#1;
end

 green=1;yellow=0;red=0; timer=0;
for(integer i=0;i<20;i=i+1)//state2
    begin timer=timer+1; #1;end
    timer=0;
for(integer i=0;i<9;i=i+1)//state3
    begin timer=timer+1; #1;end
    timer=0;
for(integer i=0;i<36;i=i+1)//state5
```

```
46      begin timer=timer+1; #1;end
47
48      $finish;
49  end
50  endmodule
```

```
1
2
3   module fail_tb;
4
5   initial begin
6
7       green=1;yellow=0;red=0; timer=0;//Initial color situations
8       clock=1;
9       #0.5 clock  = 1;
10   end
11
12  //Generate a clock with period
13     always begin clock = ~clock;
14     #0.5;
15     end
16
17  initial begin//Test sequence
18  for(integer i=0;i<35;i=i+1)//state0
19      begin timer=timer+1; #1;
20      end
21   timer=0;
22  for(integer i=0;i<5;i=i+1)//state2
23      begin timer=timer+1; #1;end
24
25  timer=0;green=0;red=1;
26  for(integer i=0;i<25;i=i+1)//state1
27      begin timer=timer+1; #1 ; end
28
29  timer=0;
30  for(integer i=0;i<12;i=i+1)//state4
31      begin timer=timer+1;#1;end
32
33  $finish;
34  end
35  endmodule
```

4 Waveforms

We did not encounter any danger in this attempt and we realized the plan of SCP-079.

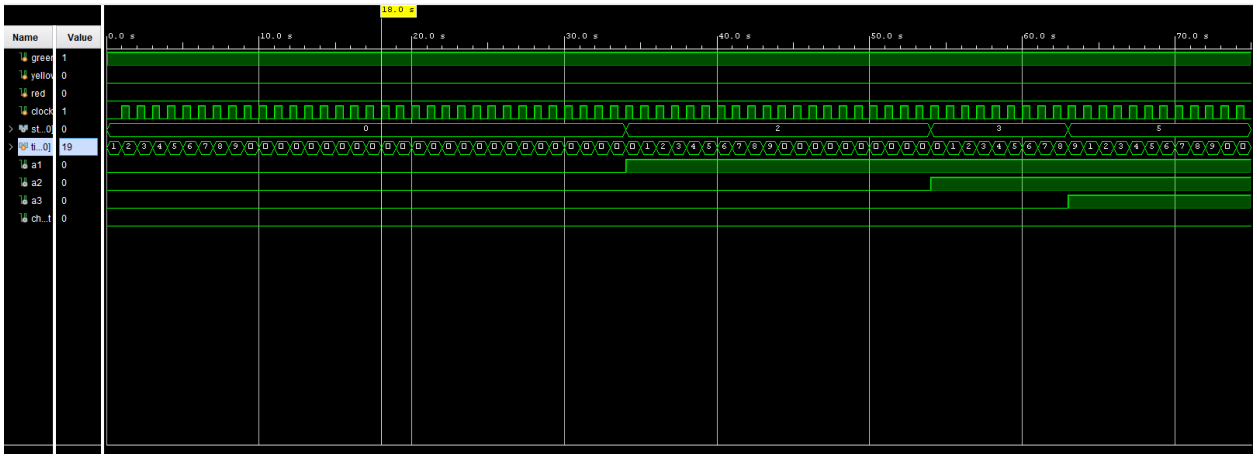

Figure 2: Allok Waveform

We encountered a small danger and the system that we designed reacted to it , but with the measures that we took, our plan was successfully realized.

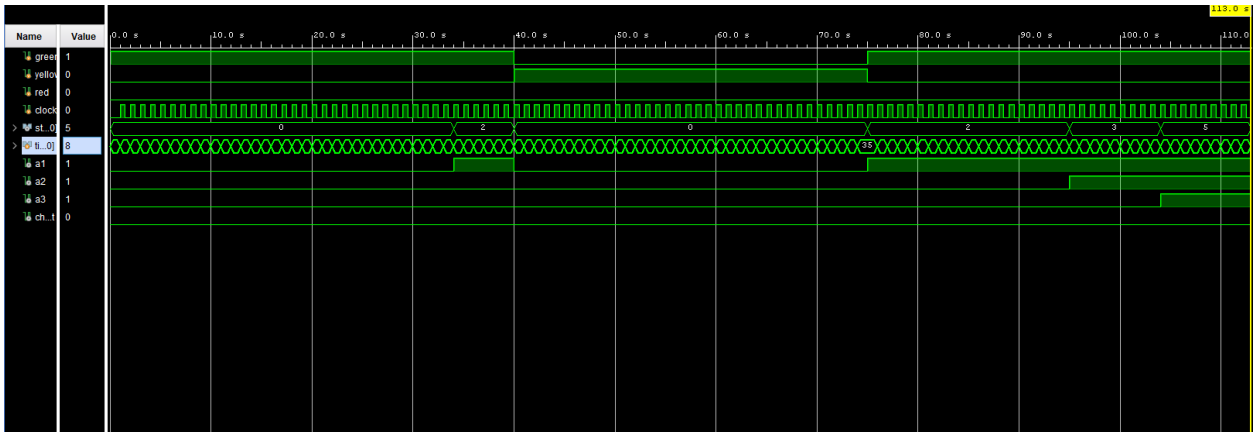

Figure 3: A1trouble Waveform

We encountered more than one danger in this scenario, but we successfully completed this project with our precautions.

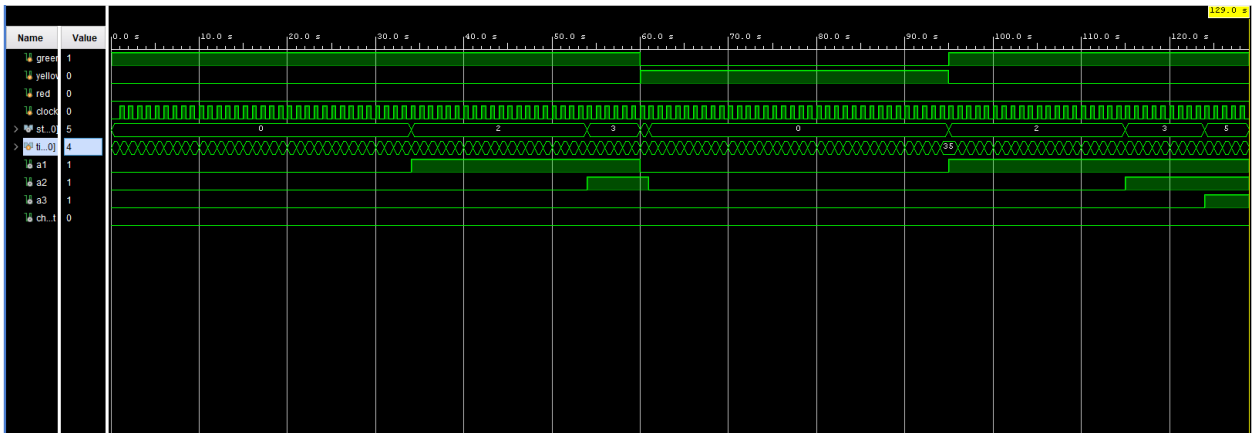

Figure 4: A2trouble Waveform

this scenario represents a cheat situation. but here we achieved our goal again by being successful

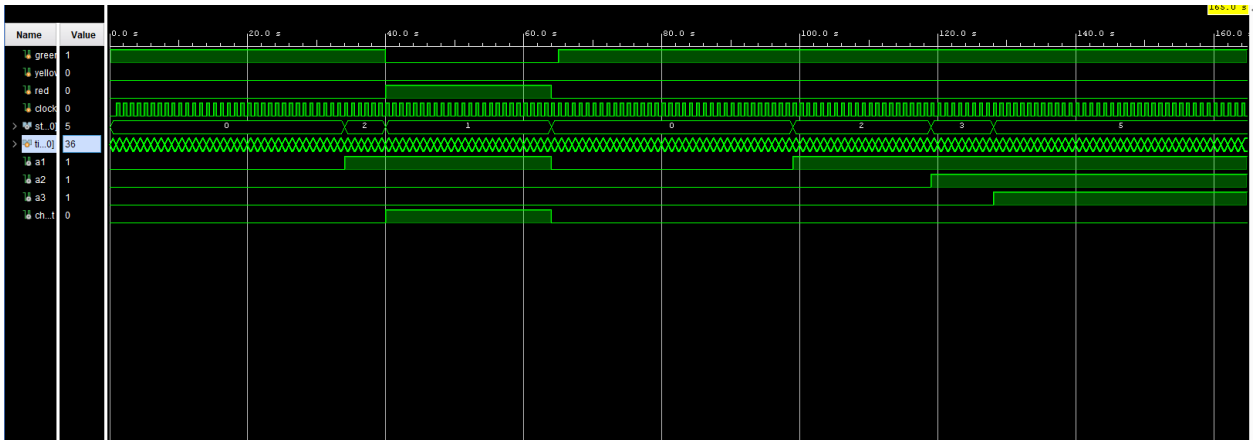

Figure 5: Cheatsuccess Waveform

In this scenario, we were unable to overcome the dangers and our plan failed.

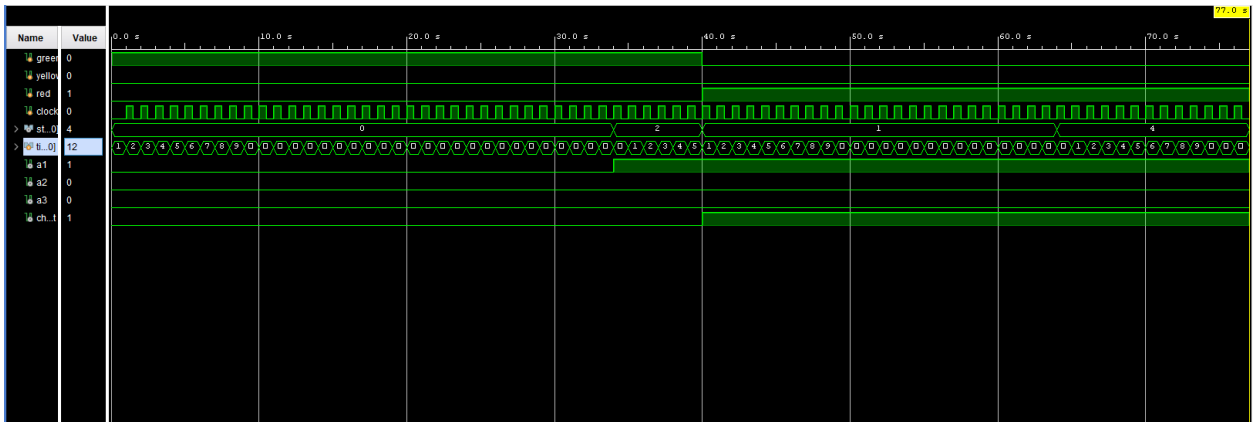

Figure 6: Fail Waveform

References

- https://scpcb.gamepedia.com/SCP-079: :text=References-,Description,attempt