Hacettepe University

Computer Engineering Department

BM233 Logic Design Lab - 2020 Fall

Final Verilog Project

January 13, 2021

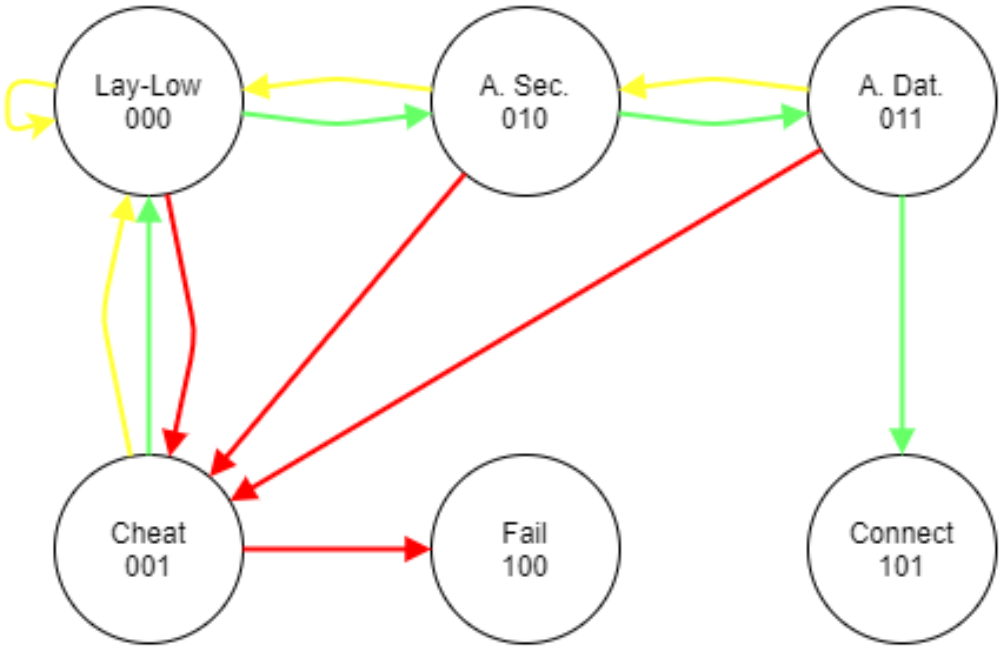
Student name:
Deniz Gönenç

Student Number:
b21946146

1 Problem Definition

Helping SCP-079 destroy mankind. It will have different states according to the awareness levels of the facility. It can attack, lay the blame on another poor microcomputer, fail or success. We will decide what it will do based on the awarenes levels of the facility which are represented by green, yellow and red.

2 State Transition Diagram



3 scp_079.v Implementation

```
1  `timescale 1ms/1ps
2
3  module scp_079(clock, green, yellow, red, state, timer, a1, a2, a3, cheat_out);
4      //setting inputs and outputs
5      input clock, green, yellow, red;
6      output reg a1 = 0, a2 = 0, a3 = 0, cheat_out = 0;
7      output reg [2:0] state = 3'b0;
8      output reg [5:0] timer = 6'b0;
9
10     always @(posedge clock)begin
11         timer <= timer + 1;
12
13         //laylow to attack security transition
14         if(state == 3'b0 && green && timer >= 6'b010100) begin
15             state <= 3'b010;
16             a1 <= 1;
17             timer <= 6'b001;
18         end
19
20         //attack security to attack database transition
21         if(state == 3'b010 && green && timer >= 6'b1010) begin
22             state <= 3'b011;
23             a2 <= 1;
24             timer <= 6'b001;
25         end
26
27         //attack security to laylow transition
28         if(state == 3'b010 && yellow) begin
29             state <= 3'b000;
30             a1 <= 0;
31             timer <= 6'b001;
32         end
33
34         //attack database to connect transition
35         if(state == 3'b011 && green && timer >= 6'b1010) begin
36             state <= 3'b101;
37             a3 <= 1;
38             timer <= 6'b001;
39         end
40
41         //attack database to attack security transition
42         if(state == 3'b011 && yellow) begin
43             state <= 3'b010;
44             a2 <= 0;
45             timer <= 6'b001;
46         end
```

```
47
48          //cheat state transition
49          if((state == 3'b000 || state == 3'b010 || state == 3'b011) && red) begin
50              state <= 3'b001;
51              cheat_out <= 1;
52              timer <= 6'b001;
53          end
54
55          if(state == 3'b001 && timer >= 6'b001111) begin
56              //cheat to fail transition
57              if(red) begin
58                  state <= 3'b100;
59                  timer <= 6'b001;
60              end
61              //cheat to laylow transition
62              else begin
63                  state <= 3'b0;
64                  a1 <= 0;
65                  a2 <= 0;
66                  a3 <= 0;
67                  cheat_out <= 0;
68                  timer <= 6'b001;
69              end
70          end
71      end
72
73  endmodule
```

4 Testbench Implementations

4.1 allok_tb.v

```
1  `timescale 1ms/1ps
2
3  `include "scp_079.v"
4
5  module allok_tb();
6      //setting inputs and outputs
7      reg green, yellow, red, clock;
8      wire a1, a2, a3, cheat_out;
9      wire [2:0] state;
10     wire [5:0] timer;
11
12     //Instantiate UUT
13     scp_079 UUT(.clock(clock), .green(green), .yellow(yellow), .red(red), .state(stat
14
15     //clock
16     always begin
17         #500 clock = 0;
18         #500 clock = 1;
19     end
20
21     initial begin
22         //all green all ok
23         clock = 1; green = 1; yellow = 0; red = 0;
24         $finish;
25     end
26
27 endmodule
```

4.2 a1trouble_tb.v

```
1
2  `timescale 1ms/1ps
3
4  `include "scp_079.v"
5
6  module allok_tb();
7      //setting inputs and outputs
8      reg green, yellow, red, clock;
9      wire a1, a2, a3, cheat_out;
10     wire [2:0] state;
11     wire [5:0] timer;
12
13     //Instantiate UUT
14     scp_079 UUT(.clock(clock), .green(green), .yellow(yellow), .red(red), .state(stat
15
16     //clock
17     always begin
18         #500 clock = 0;
19         #500 clock = 1;
20     end
21
22     initial begin
23         //green in the beginning
24         clock = 1; green = 1; yellow = 0; red = 0;
25         //yellow during attack security state
26         #25000 green = 0; yellow = 1;
27         //back to ok
28         #22000 green = 1; yellow = 0;
29         $finish;
30     end
31
32 endmodule
```

4.3 a2trouble_tb.v

```
1  `timescale 1ms/1ps
2
3  `include "scp_079.v"
4
5  module allok_tb();
6      //setting inputs and outputs
7      reg green, yellow, red, clock;
8      wire a1, a2, a3, cheat_out;
9      wire [2:0] state;
10     wire [5:0] timer;
11
12     //Instantiate UUT
13     scp_079 UUT(.clock(clock), .green(green), .yellow(yellow), .red(red), .state(stat
14
15     //clock
16     always begin
17         #500 clock = 0;
18         #500 clock = 1;
19     end
20
21     initial begin
22         //green in the beginning
23         clock = 1; green = 1; yellow = 0; red = 0;
24         //yellow during attack database state
25         #40000 green = 0; yellow = 1;
26         //back to ok
27         #20000 green = 1; yellow = 0;
28         $finish;
29     end
30
31 endmodule
```

4.4 cheatsuccess_tb.v

```
`timescale 1ms/1ps

`include "scp_079.v"

module allok_tb();
    //setting inputs and outputs
    reg green, yellow, red, clock;
    wire a1, a2, a3, cheat_out;
    wire [2:0] state;
    wire [5:0] timer;

    //Instantiate UUT
    scp_079 UUT(.clock(clock), .green(green), .yellow(yellow), .red(red), .state(stat

    //clock
    always begin
        #500 clock = 0;
        #500 clock = 1;
    end

    initial begin
        //green in the beginning
        clock = 1; green = 1; yellow = 0; red = 0;
        //red during attack database state
        #40000 green = 0; red = 1;
        //back to ok, cheat successful
        #12000 green = 1; red = 0;
        $finish;
    end

endmodule
```

4.5 fail_tb.v

```
1  `timescale 1ms/1ps
2
3  `include "scp_079.v"
4
5  module allok_tb();
6      //setting inputs and outputs
7      reg green, yellow, red, clock;
8      wire a1, a2, a3, cheat_out;
9      wire [2:0] state;
10     wire [5:0] timer;
11
12     //Instantiate UUT
13     scp_079 UUT(.clock(clock), .green(green), .yellow(yellow), .red(red), .state(stat
14
15     //clock
16     always begin
17         #500 clock = 0;
18         #500 clock = 1;
19     end
20
21     initial begin
22         //green in the beginning
23         clock = 1; green = 1; yellow = 0; red = 0;
24         //red during attack database state...
25         #40000 green = 0; red = 1;
26         //but never back to ok :( scp failed
27         $finish;
28     end
29
30 endmodule
```

5 Results

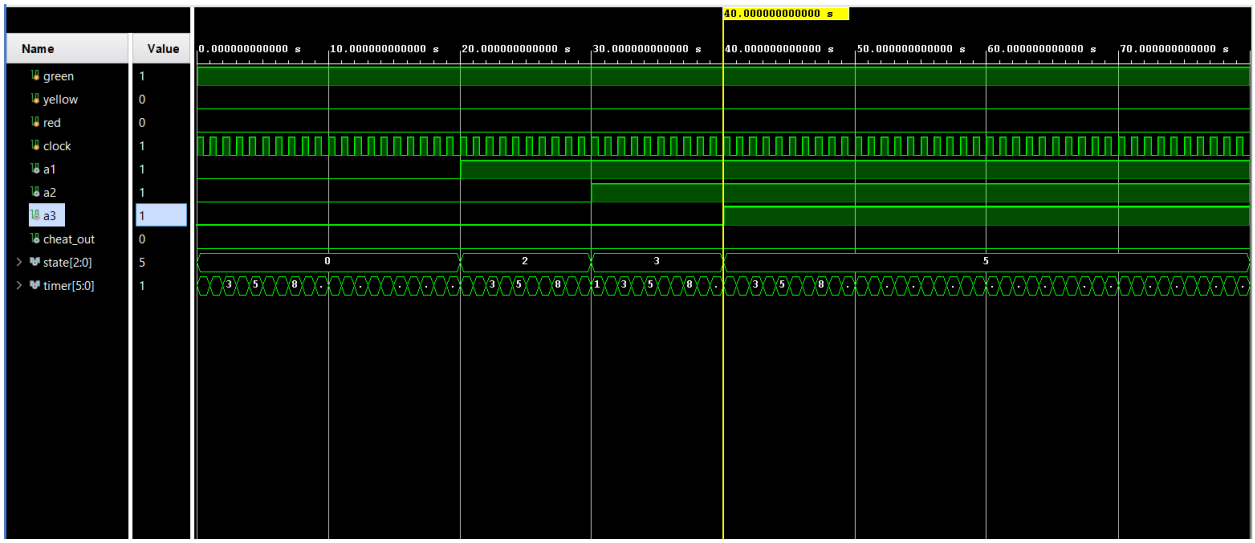


Figure 1: allok_tb.v

Lay-low State to Attack Security State transition at 20 s mark. Attack Security State to Attack Database State transition at 30 s mark. Attack Database State to Connect State transition at 40 s mark.

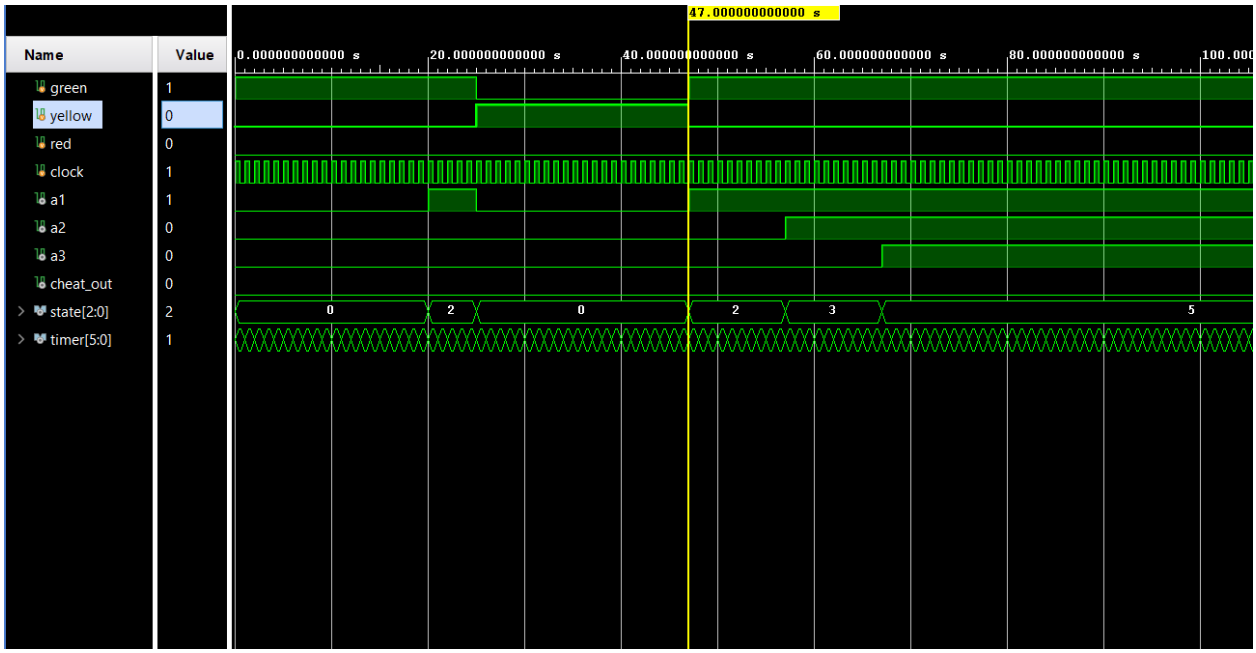


Figure 2: a1trouble_tb.v

Lay-low State to Attack Security State transition at 20 s mark. Attack Security State to Lay-low State transition at 25 s mark. Lay-low State to Attack Security State transition at 47 s mark. Attack Security State to Attack Database State transition at 57 s mark. Attack Database State to Connect State transition at 67 s mark.

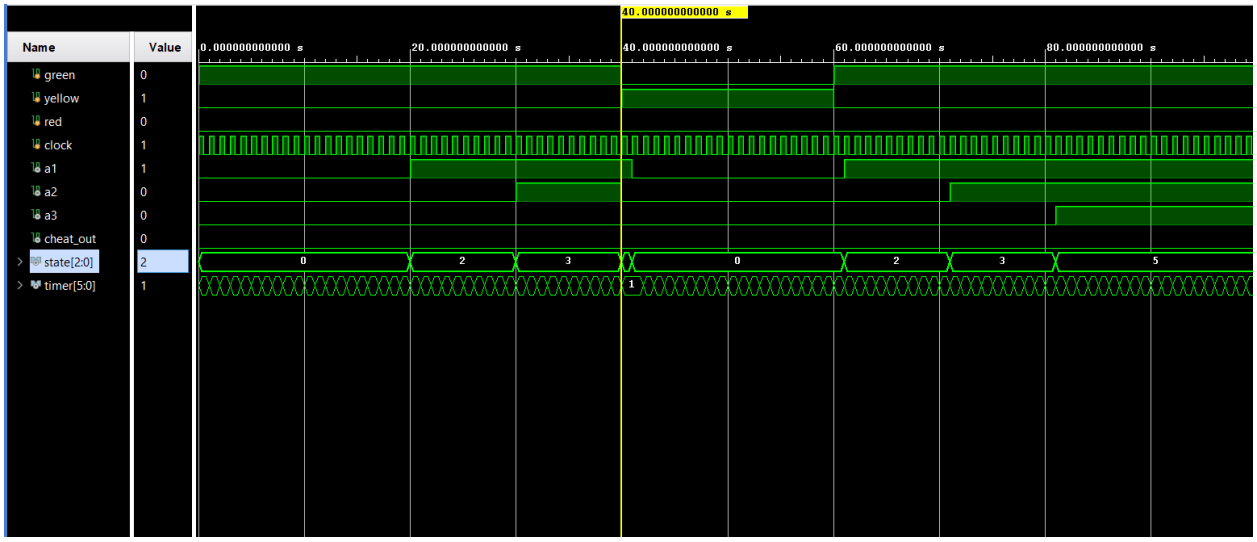


Figure 3: a2trouble_tb.v

Lay-low State to Attack Security State transition at 20 s mark. Attack Security State to Attack Database State transition at 30 s mark. Attack Database State to Attack Security State transition at 40 s mark. Attack Security State to Lay-low State transition at 41 s mark. Lay-low State to Attack Security State transition at 61 s mark. Attack Security State to Attack Database State transition at 71 s mark. Attack Database State to Connect State transition at 81 s mark.

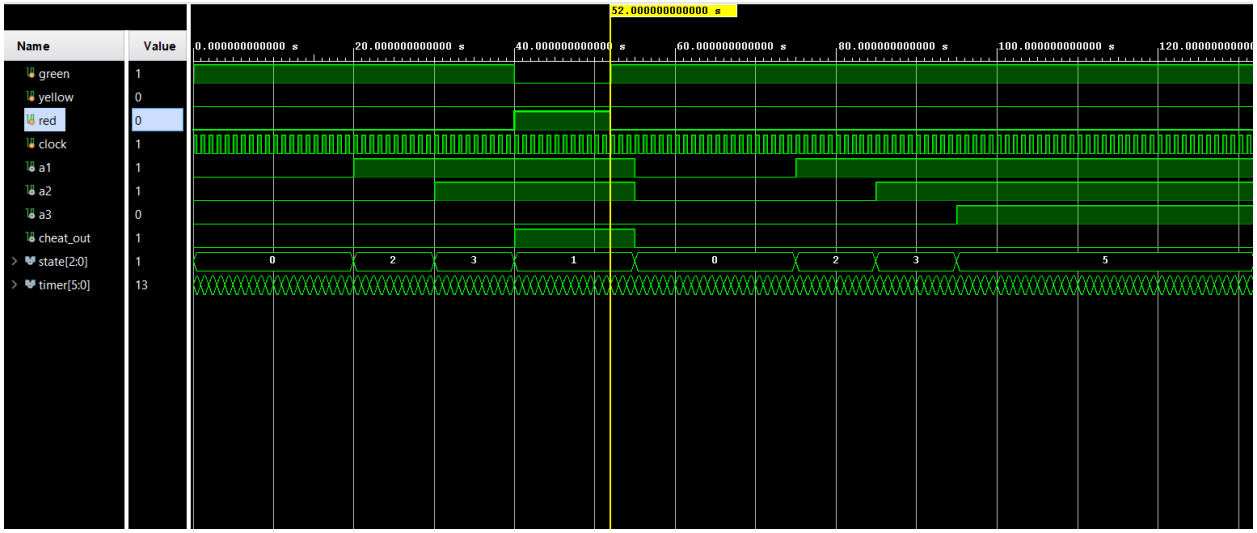


Figure 4: cheatsuccess_tb.v

Lay-low State to Attack Security State transition at 20 s mark. Attack Security State to Attack Database State transition at 30 s mark. Attack Database State to Cheat State transition at 40 s mark. Red goes low at 52 s mark. Cheat State to Lay-low State transition at 55 s mark. Lay-low State to Attack Security State transition at 75 s mark. Attack Security State to Attack Database State transition at 85 s mark. Attack Database State to Connect State transition at 95 s mark.

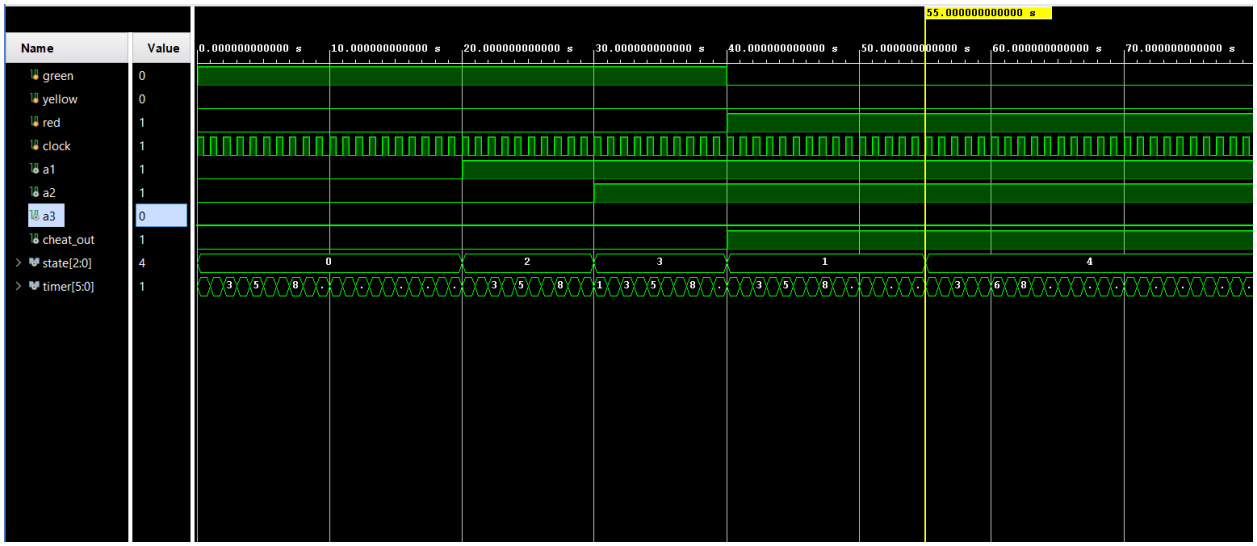


Figure 5: fail_tb.v

Lay-low State to Attack Security State transition at 20 s mark. Attack Security State to Attack Database State transition at 30 s mark. Attack Database State to Cheat State transition at 40 s mark. Cheat State to Fail State transition at 55 s mark. Red never goes away...

6 Notes

The waveforms may seem small but if you zoom in they should be readable. Sorry for any mistakes I made. And thanks for a wonderful semester!

References

- Our TAs amazing pdf.
- Stackoverflow for Verilog troubles.