HACETTEPE UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BBM 233 LOGIC DESIGN LAB – 2021 FALL

Final Project

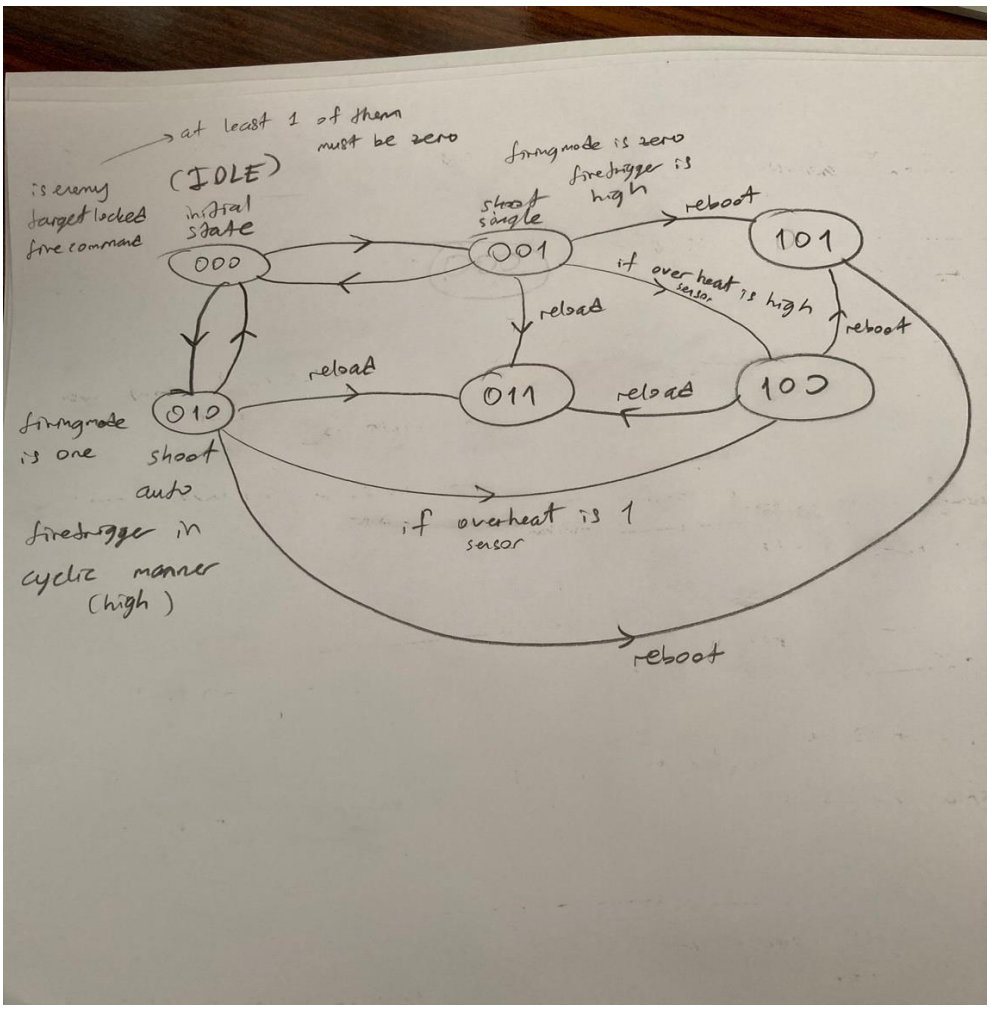
January 9, 2022

Student Name: Umut GÜNGÖR
Student Number: b21946198

Problem Definition

In this project, we are asked to design a machine gun for Turkish Special Forces BÖRÜ team. This gun uses last techonology. It has different modes and also some signals for user. I hope this gun will help BÖRÜ team.

State Diagram



Verilog Code

```
`timescale 1ms / 100ns

module siganfu_machine_gun (
    input sysclk,
    input reboot,
    input target_locked,
    input is_enemy,
    input fire_command,
    input firing_mode, // 0 single, 1 auto
    input overheat_sensor,
    output reg[2:0] current_state,
    output reg criticality_alert = 0,          // initially this is 0
    output reg fire_trigger = 0                // initially this is 0
);

    reg[2:0] next_state;

    parameter s0 = 3'b000, s1 = 3'b001, s2 = 3'b010,        // 6 states for system specifications
              s3 = 3'b011, s4 = 3'b100, s5 = 3'b101;

    integer bullet = 25, magazine = 3;          // 25 bullets and 3 magazines
    integer flag = 1;                           // this is for break the loop

    always@ (posedge sysclk, posedge reboot)    // posedge clock

        if (reboot == 1) begin                  // set current state to initial state

            current_state <= s0;
            magazine = 3;
            bullet = 25;
            next_state = s0;
            criticality_alert = 0;
            flag = 1;
            #5;

        end

        else begin current_state <= next_state; end                // update the current state


```

```
    always@ (current_state, target_locked, is_enemy, fire_command, firing_mode, overheat_sensor, fire_trigger) begin

      case (current_state)

        s0: if (target_locked & is_enemy & fire_command & ~firing_mode & ~overheat_sensor) begin        // IDLE state no shooting
              fire_trigger = 0;
              next_state = s1;
              flag = 1;
          end

            else if (target_locked & is_enemy & fire_command & firing_mode & ~overheat_sensor) begin
                fire_trigger = 0;
                next_state = s2;
                flag = 1;
            end

            else begin
                next_state = s0;
                flag = 1;
            end


        s1: if (overheat_sensor) next_state = s4;                   // single shooting state
            else if (bullet == 0) begin
                flag = 1;

                if (magazine > 0) next_state = s3;
                else next_state = s5;

            end

            else if (~target_locked | ~is_enemy | ~fire_command) begin      // if these 3 conditions are not satisfied back to thte initial state
                next_state = s0;
                flag = 1;
            end


            else if (target_locked & is_enemy & fire_command & ~firing_mode & ~overheat_sensor) begin

                fire_trigger = 1;
                bullet = bullet - 1;
                #5;
                next_state = s0;
                flag = 1;

            end
```

```
s2: if (overheat_sensor) next_state = s4;                          // auto shooting state

    else if (bullet == 0) begin

        if (magazine > 0) next_state = s3;
        else next_state = s5;
    end

    else if (~target_locked | ~is_enemy | ~fire_command) next_state = s0;       // if these 3 conditions are not satisfied back to thte initial state

    else if (target_locked & is_enemy & fire_command & firing_mode & ~overheat_sensor) begin

        while(bullet > 0 && flag) begin                            // gun shoots until it has no bullet
            fire_trigger <= ~fire_trigger;

            if(fire_trigger) bullet = bullet - 1;

            if(bullet == 1) fire_trigger = 0;
            #5;

            if (overheat_sensor) begin                             // if overheat happens first go to that state
                next_state = s4;
                flag = 0;
              end

            else if (bullet == 0) begin                            // otherwise if it has magazine go to the reload state if not go to the reboot state

                if (magazine > 0) next_state = s3;
                else next_state = s5;
                flag = 0;
            end

            else if (~target_locked | ~is_enemy | ~fire_command) begin
                next_state = s0;
                flag = 0;
            end

        end

    end
```

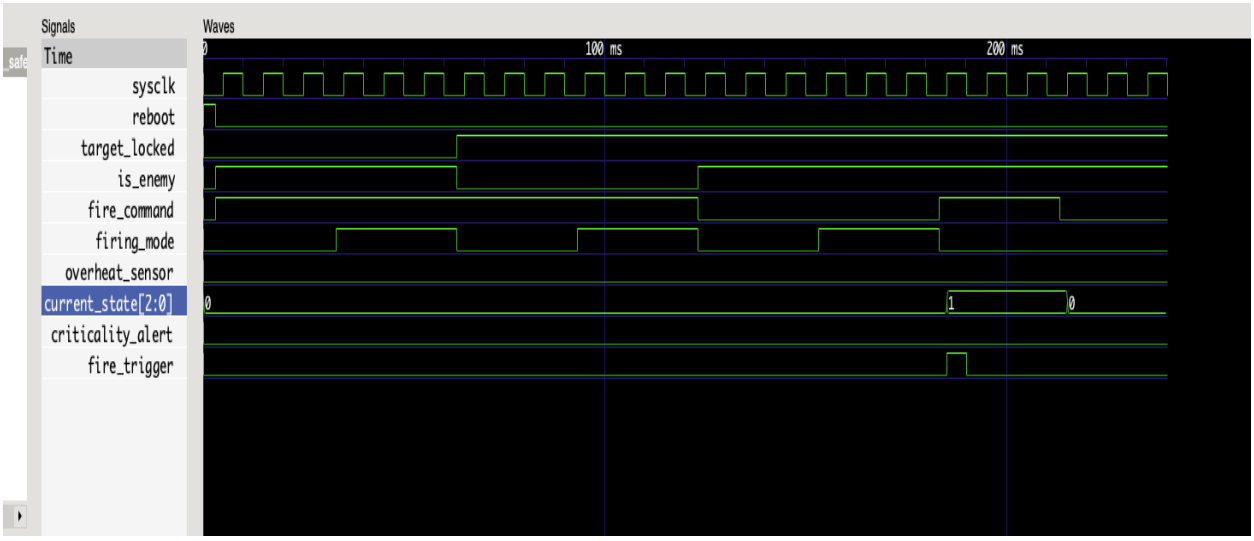
```
s3: if (overheat_sensor) next_state = s4;                  // reload state

    else if (magazine == 1) begin

        #50;
        bullet = 25;                    // reload bullets
        magazine = magazine - 1;        // decreases the magazine
        criticality_alert = 1;          // if there is not another magazine criticality alert is goes to high
        flag = 1;

        if (~target_locked | ~is_enemy | ~fire_command) next_state = s0;
        else if (target_locked & is_enemy & fire_command & ~firing_mode & ~overheat_sensor) next_state = s1;
        else if (target_locked & is_enemy & fire_command & firing_mode & ~overheat_sensor) next_state = s2;
    end

    else if (magazine > 1) begin

        #50;                    // wait 50 ms
        bullet = 25;
        magazine = magazine - 1;
        flag = 1;

        if (~target_locked | ~is_enemy | ~fire_command) next_state = s0;                                    // conditions are not satisfied go to the IDLE state
        else if (target_locked & is_enemy & fire_command & ~firing_mode & ~overheat_sensor) next_state = s1;       // firing mode is low go to the single shoot state
        else if (target_locked & is_enemy & fire_command & firing_mode & ~overheat_sensor) next_state = s2;        // firing mode is high go to the auto shoot state

    end
```

```
157         s4: if (bullet == 0 && magazine == 0) begin                         // overheat state
158             #100;                   // wait 100 ms
159             next_state = s5;
160             flag = 1;
161         end
162
163         else if (bullet == 0 && magazine > 0) begin                         // if reload needed go to the reload state
164             #100;
165             next_state = s3;
166             flag = 1;
167         end
168
169         else if (!target_locked | !is_enemy | !fire_command) begin
170             #10;
171             next_state = s0;
172             flag = 1;
173         end
174
175         else if (target_locked & is_enemy & fire_command & ~firing_mode) begin
176             #100;
177             next_state = s1;
178             flag = 1;
179         end
180
181         else if (target_locked & is_enemy & fire_command & firing_mode) begin
182             #100;
183
184                 if(!target_locked | !is_enemy | !fire_command) begin
185                   next_state = s0;
186                   flag = 1;
187                 end
188
189                 else begin
190                     next_state = s2;
191                     flag = 1;
192                 end
193         end
```

```
194
195       s5: if (reboot == 1) begin                // if reboot is high reload magazines bullets and go to the initial state to start over again
196         magazine = 3;
197         bullet = 25;
198         next_state = s0;
199         criticality_alert = 0;
200         flag = 1;
201         #5;
202
203       end
204           else begin                            // if reboot signal is low and gun in this state it remains in this state until reboot signal is high
205               next_state = s5;
206               flag = 1;
207           end
208
209       endcase
210
211   end
212
213 endmodule
214
```

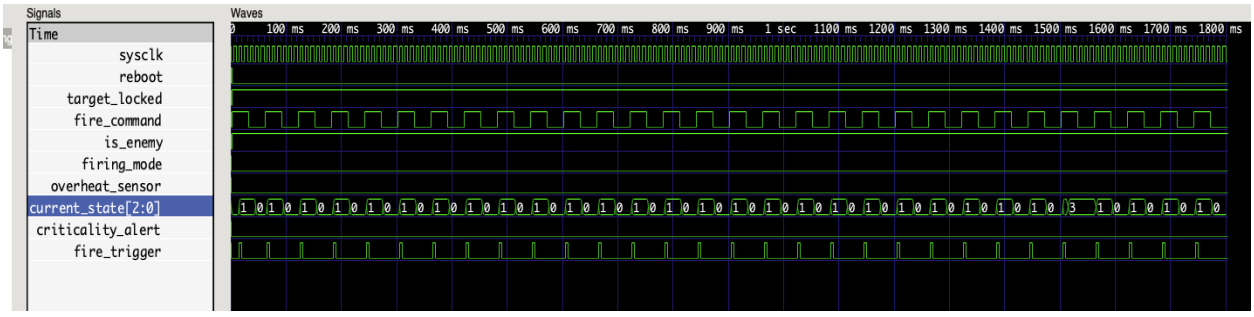
Waveforms

Safety Test



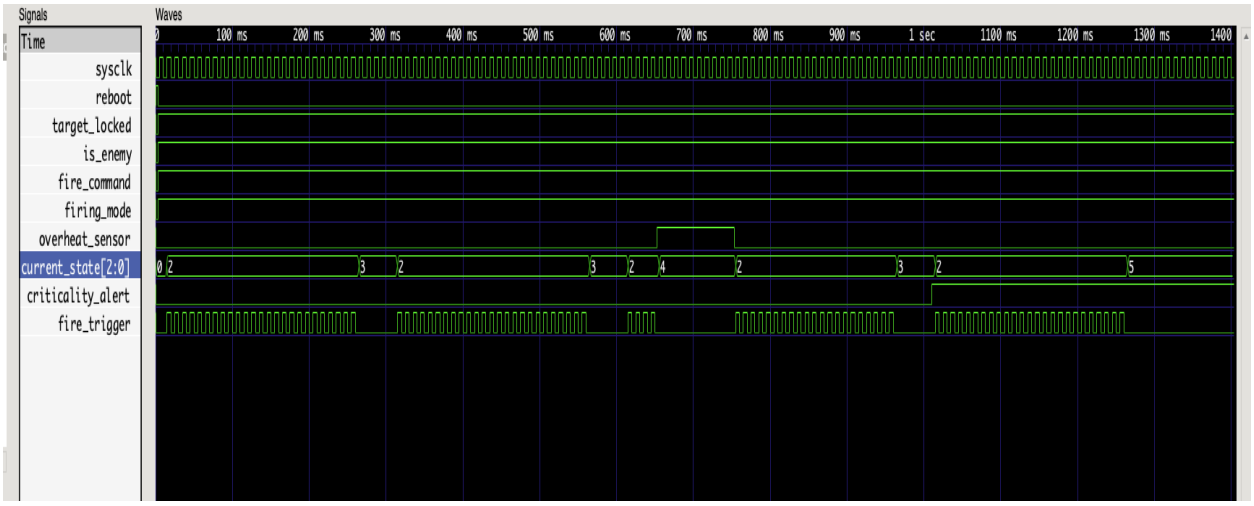
At first, reboot signal is active so machine is at rest. Until target_locked, is_enemy and fire_command signals are high machine remains at IDLE state. When they are all high and firing_mode signal is low machine goes to the single shoot state.

Single Shooting Test



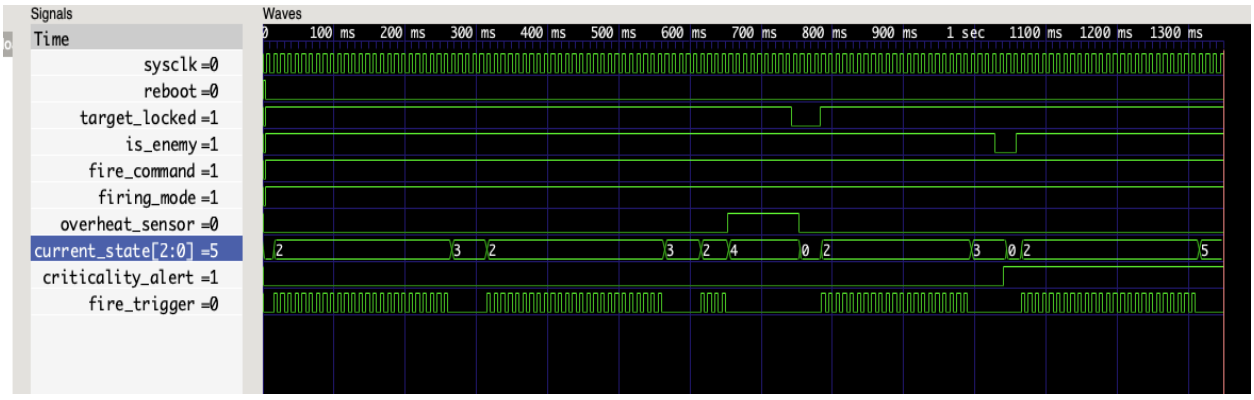
While gun has bullet, it shoots in single mode fire_trigger changes in cyclic manner. When there is no bullet in magazine machine goes to reload state. Reload the bullets and continue shooting.

Automatic Shooting Test



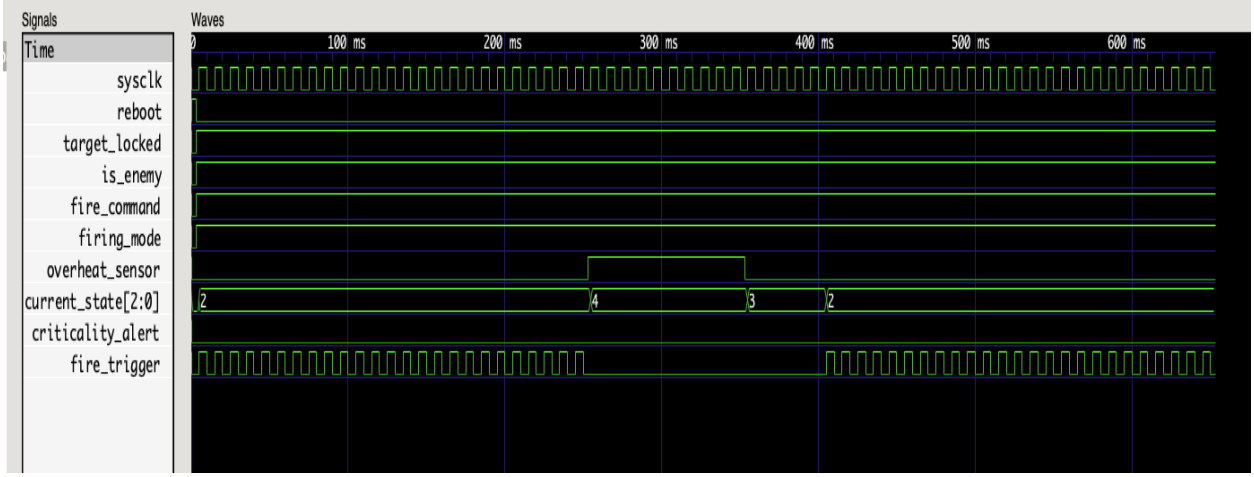
As long as magazine has bullet, machine shoots and remains at automatic shoot state. When magazine is out of bullet machine goes to the reload state and reloads. When there is only one magazine left criticality_alert signal goes to high. Finally, when last magazine has no bullet machine goes to the downfall state and remains there until reboot signal.

Return from Reload and Overheat Test



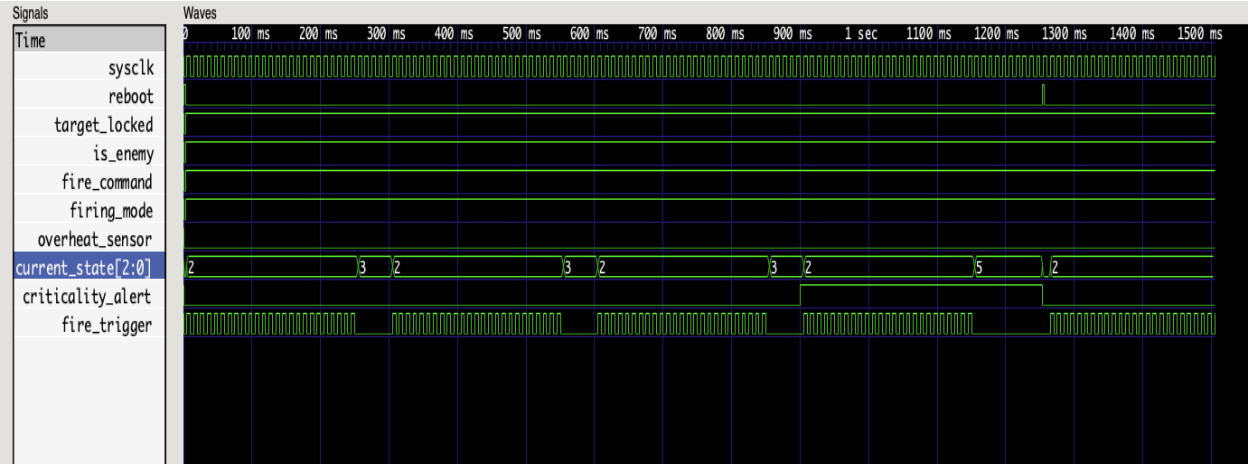
In this test, when gun returns from reload or overheat states it checks target_locked, is_enemy and fire_command signals. If one of those is low machine goes to the initial state.

Reload After Overheat Test



After reboot signal is low, machine starts shooting. When it runs out of bullet at the same time overheat sensor is going to be high. Firstly it goes to the overheat state and waits a hundred miliseconds after that it goes to the reload state and waits fifty miliseconds reload the magazine and continue shooting.

Reboot Test



If there are not any bullets and magazines in the gun, machine goes to the downfall state. However, if reboot signal is high machine restarts itself. It refills its ammo and continue shooting.

References

- chipverify.com
- stackoverflow.com
- javatpoint.com