Hacettepe University

Computer Engineering Department

BM233 Logic Design Lab - 2021 Fall

Final Project

December 29, 2021

Student name:
Mert Ali Yalçın

Student Number:
21946682

1 Problem Definition

A finite-state machine (FSM) or simply a state machine is used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of user-defined states. The machine is in only one state at a time; the state it is in at any given time is called the current state. It can change from one state to another when initiated by a triggering event or condition; this is called a transition. A particular FSM is defined by a list of its states, and the triggering condition for each transition. The behavior of state machines can be observed in many devices in modern society performing a predetermined sequence of actions depending on a sequence of events with which they are presented. Simple examples are vending machines which dispense products when the proper combination of coins are deposited, elevators which drop riders off at upper floors before going down, traffic lights which change sequence when cars are waiting, and combination locks which require the input of combination numbers in the proper order. The state machines are modeled using two basic types of sequential networks- Mealy and Moore. In a Mealy machine, the output depends on both the present (current) state and the present (current) inputs. In Moore machine, the output depends only on the present state. A general model of a Mealy sequential machine consists of a combinatorial network, which generates the outputs and the next state, and a state register which holds the present state as shown below. The state register is normally modeled as D flip-flops. The state register must be sensitive to a clock edge. The other block(s) can be modeled either using the always procedural block or a mixture of the always procedural block and dataflow modeling statements; the always procedural block will have to be sensitive to all inputs being read into the block and must have all output defined for every branch in order to model it as a combinatorial block. In this final project, we should design a mealy machine as follows:
Inputs:

- SYSCLK: This input is the system clock signal provided to the chip from the armor system. The clock cycle is set to 10 ms.
- REBOOT: This input provides a reset signal (on HIGH) which resets the system to its default settings (idle state, full ammo).
- target_locked: This input is supplied to the chip from the helmet's target acquisition system. Once the gun has been locked on the target, this signal becomes HIGH. As soon as the target lock has been lost, the signal immediately becomes LOW.
- is_enemy: This is yet another input signal supplied from the helmet's target acquisition system. Since the helmet has a connection to KAM AI, an ultra-fast recognition of the targets is performed in real-time. This signal is HIGH when the target is recognized as a threat, and LOW otherwise.
- firing_mode: This signal is supplied from the combatant's neural-link. When the combatant wishes to shoot in the automatic firing mode, this signal becomes HIGH. Otherwise it is LOW by default, and in that case, the firing mode of the machine gun is set to firing a single shot.

- fire_command: This is yet another input signal supplied from the combatant's neural-link. When the combatant wishes to fire the gun (equivalent of pulling the trigger in traditional guns) this signal becomes HIGH. Otherwise, it is LOW by default.
- overheat_sensor: The gun is equipped with a high-tech cooling system which is triggered when the gun overheats. This overheating is detected by a temperature sensor mounted on the gun. When the gun overheats, the signal becomes HIGH. Otherwise, it is LOW by default.

Outputs:

2:0 current_state: A 3-bit wide output that shows the current state of the gun. It is supplied to the helmet's display system so that the combatant can be aware of the gun's status.

- fire_trigger: A 1-bit output signal which triggers the machine gun. Whenever it goes HIGH, a bullet is fired. For automatic firing mode, it should be set to HIGH then LOW in a cyclic manner to achieve automatic shooting.
- criticality_alert: Also a 1-bit output signal which goes HIGH when the last magazine has been loaded into the gun. This signal is also supplied to the helmet's display system so that the combatant can be aware that the ammo is about to become completely depleted.

States:

- idle (binary code 000): This is the start state. In this state, the machine gun is at rest. No shots are being fired.
- shoot_single (binary code 001): In this state, only a single shot is fired from the machine gun.
- shoot_auto (binary code 010): While in this state, the machine gun is in the automatic firing mode. In this state, the machine gun keeps firing at the rate of 6000 rpm until it has exhausted its ammunition supply or it has overheated.
- reload (binary code 011): When the machine gun depletes the ammunition from the magazine it enters this state in which it loads a new magazine. Reloading takes 50 ms.
- overheat (binary code 100): When the machine gun overheats from shooting, it has to be cooled down, otherwise it cannot resume shooting. The machine gun stays in this state until it is cooled down. This process takes 100 ms.
- downfall (binary code 101): When all of the magazines and ammo have been exhausted, the machine gun goes into this state, at which point it cannot shoot anymore until the system has been rebooted by technical service which replenishes the ammunition.

System Behavior — State Transitions
The SIGANFU_MACHINE_GUN has a capacity of carrying 4 magazines, each of which has a capacity of 25 .50 caliber rounds (a total of 100 bullets). After depleting a magazine, the machine gun should reload immediately until there are no spare magazines left. The reloading process takes

50 milliseconds. The user is notified that they are on their last magazine by setting the output signal criticality_alert to HIGH.

The shooting process is triggered by the fire_command input signal and is controlled by setting the output signal fire_trigger to HIGH, which operates as an electronic trigger to fire bullets towards the target.

The weapon has two firing modes defined as "Single Shooting" and "Auto Shooting". The firing mode of the weapon is specified using the firing_mode input signal. There is no direct transition between firing modes. The system must go into the IDLE state before switching between the firing modes for at least one clock cycle.

In automatic firing mode, the weapon fires a bullet every 10 milliseconds as long as the fire_command signal is HIGH (remember that the shooting rate of the SIGANFU_MACHINE_GUN is 6000 rpm). Firing a bullet is achieved by setting the fire_trigger output signal to HIGH for 5 ms and then to LOW for another 5 ms to achieve the specified firing rate.

In single firing mode, the weapon fires a single bullet using the same trigger delay. In this mode, the weapon cannot fire more than one round before going back to the IDLE state, and this is true even if the fire_command signal remains HIGH after the bullet has been fired. After the fire_command signal has become LOW, the weapon can return to the IDLE state. This mode is useful for precise shooting tasks such as sniper functionality.

The weapon has two fail-safe mechanisms which prevent its extremely destructive power to be used against allies or innocent civilians. Both mechanisms work by receiving input signals from the helmet's Target-Acquiring-System (TAS). By its specifications, the weapon cannot fire until a proper target is acquired (i.e., there is a precise lock on the target), which is specified by the input signal target_locked being HIGH, in order to prevent bystander casualties. The weapon also cannot fire upon allies or non-enemy parties, which is specified by the input signal is_enemy.

It also has an internal temperature sensor that prevents the weapon from overheating and eliminates the possibility of any malfunction caused by overheating. The sensor makes the overheat_sensor input signal HIGH when it detects temperatures above the operating range. When this signal is HIGH, the machine gun should stop firing immediately and wait for 100 milliseconds to cool itself down in the OVERHEAT state.

Idle

The initial state of the SIGANFU_MACHINE_GUN finite state machine (as well as the state when the system is rebooted) is the IDLE state. When in this state, the SIGANFU_MACHINE_GUN can transition into one of the shooting states (SHOOT_SINGLE or SHOOT_AUTO) only if all three of the following conditions are satisfied: The target is an enemy (if is_enemy is HIGH) TAS is locked on the target (if target_locked is HIGH) the user wants to fire the gun (if fire_command is HIGH) If all of the conditions for firing the gun are satisfied, the SIGANFU_MACHINE_GUN can proceed into one of two possible shooting states: either SHOOT_AUTO or SHOOT_SINGLE, based on the firing_mode input signal (LOW for SHOOT_SINGLE and HIGH for SHOOT_AUTO). If any one of the necessary conditions for shooting is not satisfied, then the SIGANFU_MACHINE_GUN should stay in the IDLE state.

SHOOT_AUTO

When in the SHOOT_AUTO state, the SIGANFU_MACHINE_GUN should first check if the gun has overheated (if overheat_sensor is HIGH). If this is the case, the SIGANFU_MACHINE_GUN should transition to the OVERHEAT state where it must stay for 100 milliseconds for the cooling to be completed.

Secondly, the SIGANFU_MACHINE_GUN should check if there are any bullets left in the currently used magazine. If there are not any bullets left, then the SIGANFU_MACHINE_GUN has two options based on the number of spare magazines left. If there aren't any magazines left to reload, the SIGANFU_MACHINE_GUN should proceed to the DOWNFALL state. If there are enough spare magazines to reload, the SIGANFU_MACHINE_GUN should proceed to the RELOAD state where it must stay for 50 milliseconds, the time it takes for reloading to complete.

Finally, the SIGANFU_MACHINE_GUN should check if the shooting is still the intended behavior in the same way as in the IDLE state (check all the conditions that must be satisfied for shooting). If not, the SIGANFU_MACHINE_GUN should transition to the IDLE state.

After making sure the SIGANFU_MACHINE_GUN can and should continue to fire (to stay in SHOOT_AUTO), the necessary steps for firing should be taken (continue firing, updating ammo, etc.).

SHOOT_SINGLE

Similarly to the SHOOT_AUTO state, the SIGANFU_MACHINE_GUN should make sure if it can and should fire in this state before taking any action and transition to the appropriate states accordingly.

The main difference of this state from the SHOOT_AUTO state is that the SIGANFU_MACHINE_GUN should only fire ONCE when in this state, and then transition to the IDLE state once the fire_command becomes LOW. Until fire_command becomes LOW, the SIGANFU_MACHINE_GUN should wait in this state without firing any more bullets.

RELOAD

In this state, the SIGANFU_MACHINE_GUN should change the current depleted magazine with one of the spare magazines. To do this, it should replenish the number of bullets in the current magazine (to 25) while decreasing the number of the available spare magazines. Before this operation, the SIGANFU_MACHINE_GUN should check if the reloading is actually possible by making sure it has a positive number of the remaining spare magazines. Once reloading is succesfully performed, if the user is on their last magazine (no more spare magazines left), then the SIGANFU_MACHINE_GUN should warn the user about this critical situation by making the output signal criticality_alert HIGH. The reloading process takes 50 milliseconds, so the SIGANFU_MACHINE_GUN must wait for this exact amount of time in this state.

After the reload operation is complete, the SIGANFU_MACHINE_GUN should check if it should go back to shooting or not, and transition to the appropriate state by checking the firing_mode input signal if all the shooting conditions are satisfied. If it is decided that the SIGANFU_MACHINE_GUN should not fire, it should then transition to the IDLE state.

OVERHEAT

In this state, the SIGANFU_MACHINE_GUN must wait for 100 milliseconds for the cooling process to complete. When the time has come for the SIGANFU_MACHINE_GUN to continue operating, the SIGANFU_MACHINE_GUN should make sure if it can and should transition back to the shooting state or not, and then proceed to the appropriate state accordingly (back to shooting if possible and wanted, to the DOWNFALL state if completely out of ammo, to the RELOAD state if needs to reload, or the IDLE state otherwise).

If it can and should fire, then it should transition to the appropriate state by checking the firing_mode input signal.

DOWNFALL

This is the failure state of the SIGANFU_MACHINE_GUN in which it cannot shoot anymore no matter what the inputs are, until it has been rebooted (ammo replenished and system reset via technical service). When this occurs, the user is out of bullets and should find other ways to fight vicious enemies. Only when the system gets rebooted (REBOOT signal becomes HIGH), the SIGANFU_MACHINE_GUN can exit this state and transition back to the IDLE state, but this time all ammo stats will also be reset (the current magazine will have 25 bullets, there will be 3 more full spare magazines available).

2 State Transition Diagram

1.

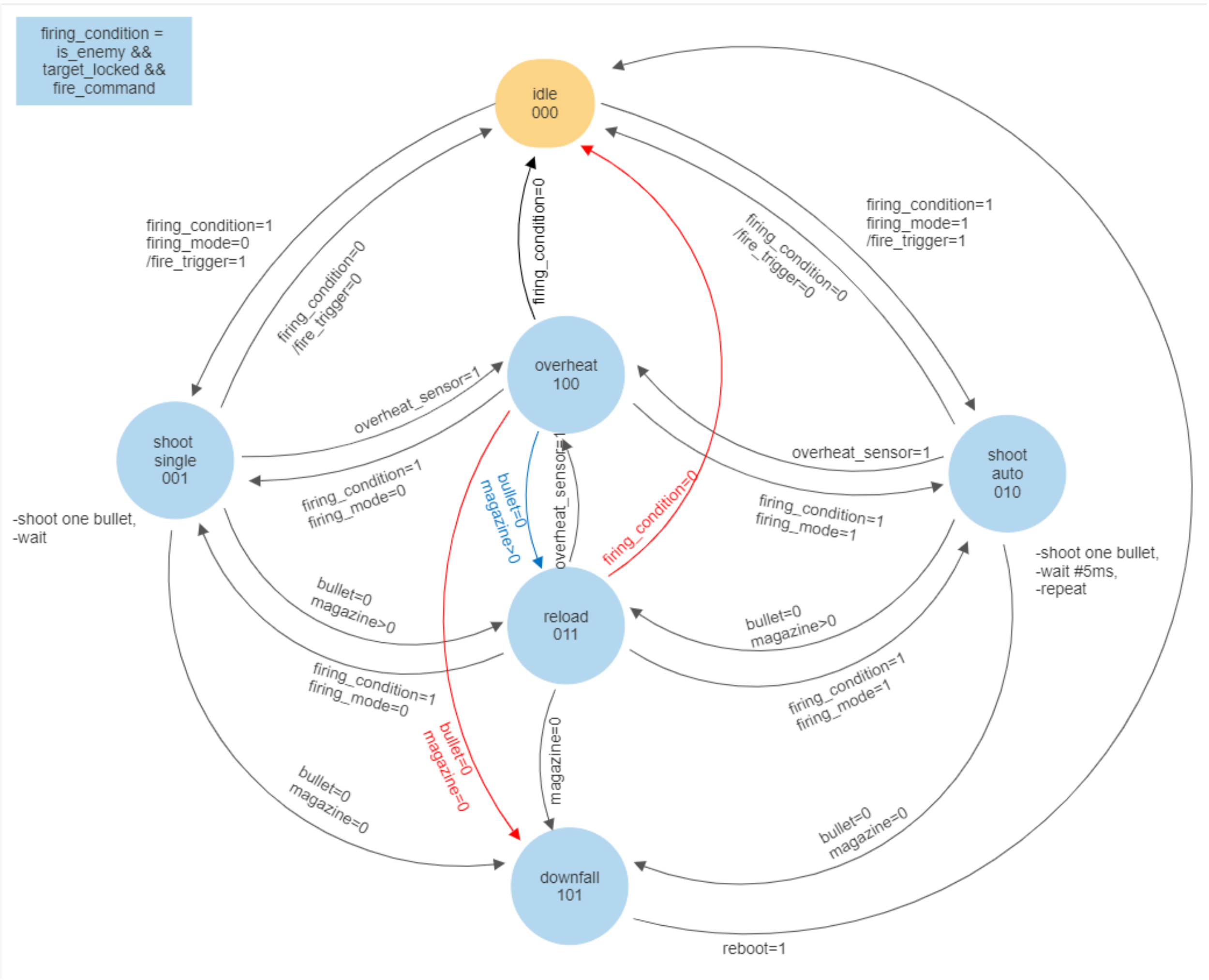


Figure 1: State Transition Diagram

3 Solution Code

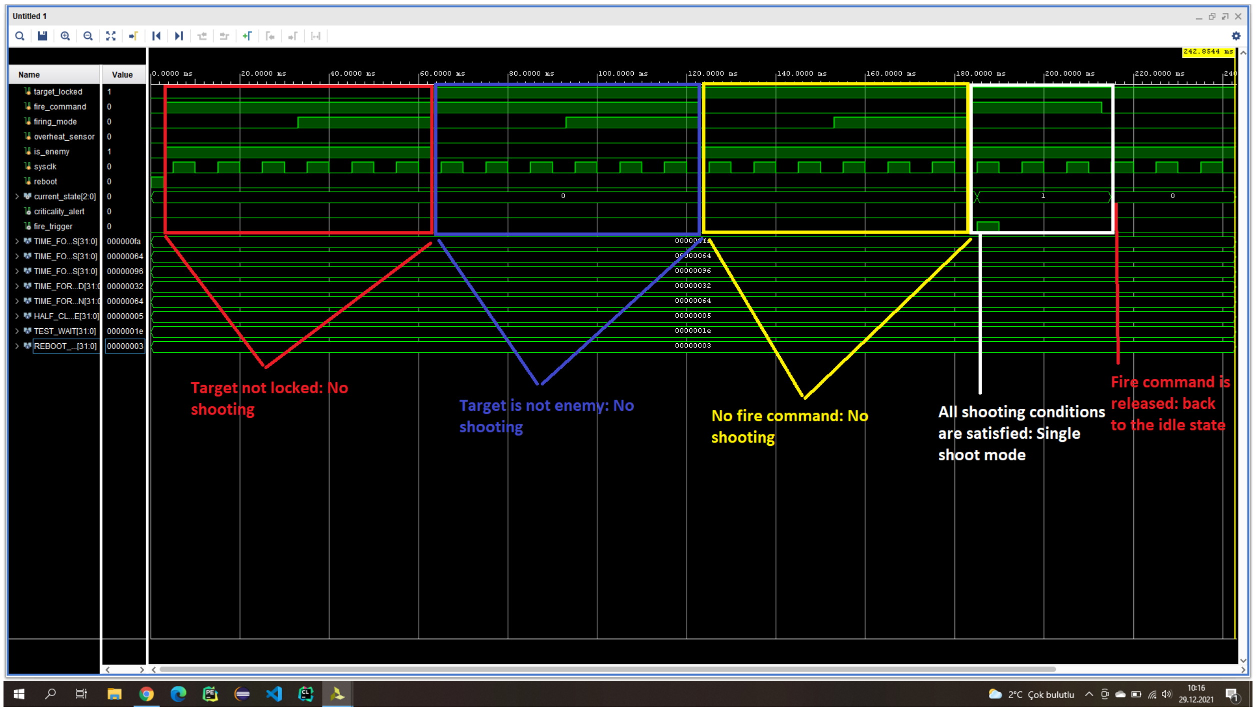
```
‘timescale 1ms / 100ns

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,
        output reg fire_trigger
);
reg[2:0] nextstate;
parameter idle=3’b000;
parameter shoot_single=3’b001;
parameter shoot_auto=3’b010;
parameter reload=3’b011;
parameter overheat=3’b100;
parameter downfall=3’b101;

assign firing_condition = is_enemy && target_locked && fire_command; //all of them mu

integer bullet = 25;
integer magazine = 3;
integer shot = 0; //checks if shoot_single has already been shot

always @(posedge sysclk or posedge reboot) // always block to update state
if (reboot)
 current_state <= idle;
else
 current_state <= nextstate;


 always @(current_state or firing_condition or reboot or firing_mode or overheat_sens
 begin
    case(current_state)
        idle:
        begin
             if(firing_condition && firing_mode == 0)
             begin
                 fire_trigger = 0;
                 nextstate = shoot_single;
             end
             else if(firing_condition && firing_mode == 1)
             begin
                 fire_trigger = 0;
                 nextstate = shoot_auto;
             end
             else
             begin
               fire_trigger=0;
               nextstate = idle;
             end
         end
        shoot_single:
            if(overheat_sensor)
            begin
                shot = 0;
                nextstate = overheat;
            end
            else if(bullet > 0)
            begin
                if(shot == 0)
                begin
                    fire_trigger = 1;
                    #5;
                    fire_trigger = 0;
                    bullet = bullet - 1;
                    shot = 1;
                end
                if(fire_command == 0)
                begin
                    shot = 0;
                    nextstate=idle;
                end
            else if(bullet == 0 && magazine > 0)
            begin
                shot = 0;
                nextstate=reload;
            end
            else if(bullet == 0 && magazine == 0)
            begin
                shot = 0;
                nextstate = downfall;
            end
            end
         shoot_auto:
         begin
           while((fire_command == 1) && (bullet>0) && (overheat_sensor == 0) && (firi
               begin
                   fire_trigger = 1;
                   bullet = bullet - 1;
                   #5;
                   fire_trigger = 0;
                   #5;
               end
            if(overheat_sensor)
                nextstate = overheat;
            else if(firing_condition == 0)
                nextstate=idle;
            else if(bullet == 0)
            begin
                if(magazine > 0)
                    nextstate=reload;
                else
                    nextstate=downfall;
            end
        end
        reload:
        begin
            if(magazine > 0)
            begin
                magazine = magazine - 1;
                bullet = 25;
                #50;
            end
            else if(magazine < 0)
                nextstate = downfall;
            if(firing_condition)
                begin
                    if(firing_mode == 1)
                        nextstate = shoot_auto;
                    else
                        nextstate = shoot_single;
                end
             else
                nextstate = idle;
        end
        overheat:
        begin
            #100;
            if(firing_condition)
            begin
                if(bullet == 0 && magazine < 0)
                    nextstate = downfall;
                else if(bullet == 0 && magazine >= 0)
                    nextstate = reload;
                else if(firing_mode == 0)
                    nextstate=shoot_single;
                else
```

```
                    nextstate = shoot_auto;
            end
            else if(bullet == 0 && magazine < 0)
                nextstate = downfall;
            else if(bullet == 0 && magazine >= 0)
                nextstate = reload;
            else
                nextstate=idle;
        end
        downfall:
        begin
            if(reboot == 1)
            begin
                bullet = 25;
                magazine = 3;
                nextstate = idle;
            end
        end
    endcase
 end

 always @(magazine) //always block to check the criticality alert
 begin
    if(magazine == 0)
        criticality_alert = 1;
    else
        criticality_alert = 0;
 end

endmodule
```

4 Results

TEST 1: Safety Test

Fig. 2.



Name	Value
target_locked	1
fire_command	0
firing_mode	0
overheat_sensor	0
is_enemy	1
sysclk	0
reboot	0
current_state[2:0]	0
criticality_alert	0
fire_trigger	0
TIME_FO...S[31:0]	000000fa
TIME_FO...S[31:0]	00000064
TIME_FO...S[31:0]	00000096
TIME_FOR...D[31:0	00000032
TIME_FOR...N[31:0	00000064
HALF_CL...E[31:0]	00000005
TEST_WAIT[31:0]	0000001e
REBOOT_...[31:0]	00000003

Target not locked: No shooting

Target is not enemy: No shooting

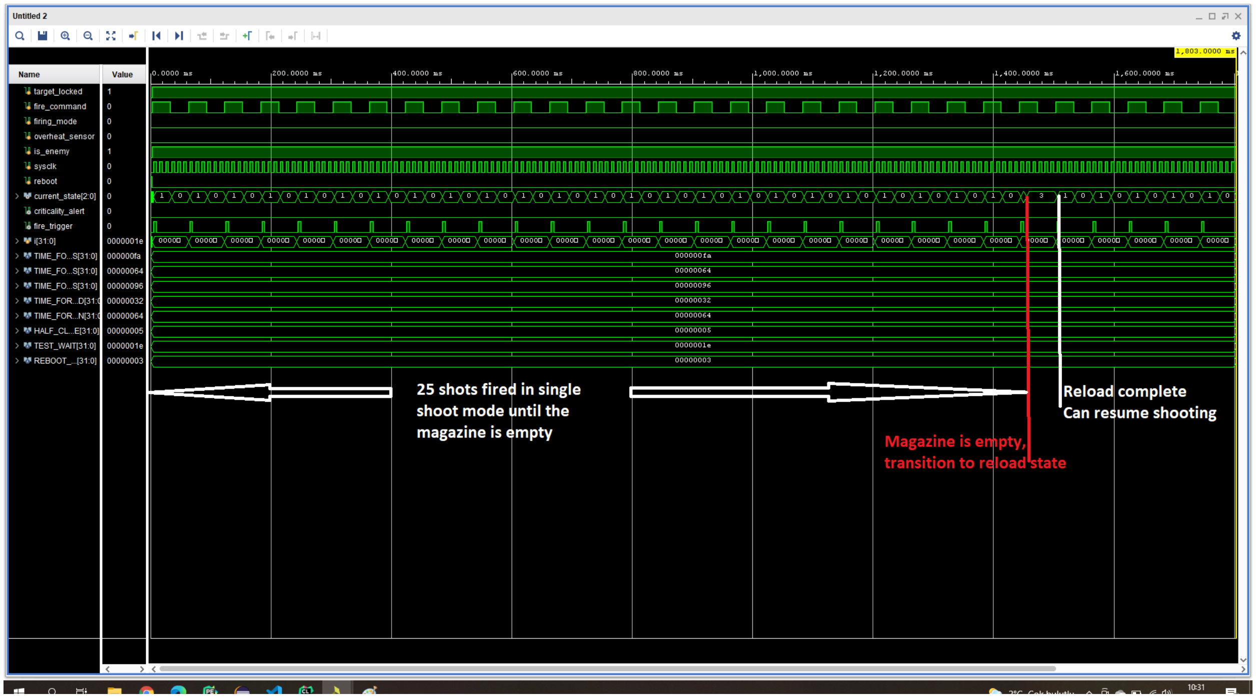
No fire command: No shooting

All shooting conditions are satisfied: Single shoot mode

Fire command is released: back to the idle state

Figure 2: Safety Test

TEST 2: Single Shooting Test
Fig. 3.



Name	Value
target_locked	1
fire_command	0
firing_mode	0
overheat_sensor	0
is_enemy	1
sysclk	0
reboot	0
current_state[2:0]	0
criticality_alert	0
fire_trigger	0
i[31:0]	0000001e
TIME_FO...S[31:0]	000000fa
TIME_FO...S[31:0]	00000064
TIME_FO...S[31:0]	00000096
TIME_FOR...D[31:0	00000032
TIME_FOR...N[31:0	00000064
HALF_CL...E[31:0]	00000005
TEST_WAIT[31:0]	0000001e
REBOOT_...[31:0]	00000003

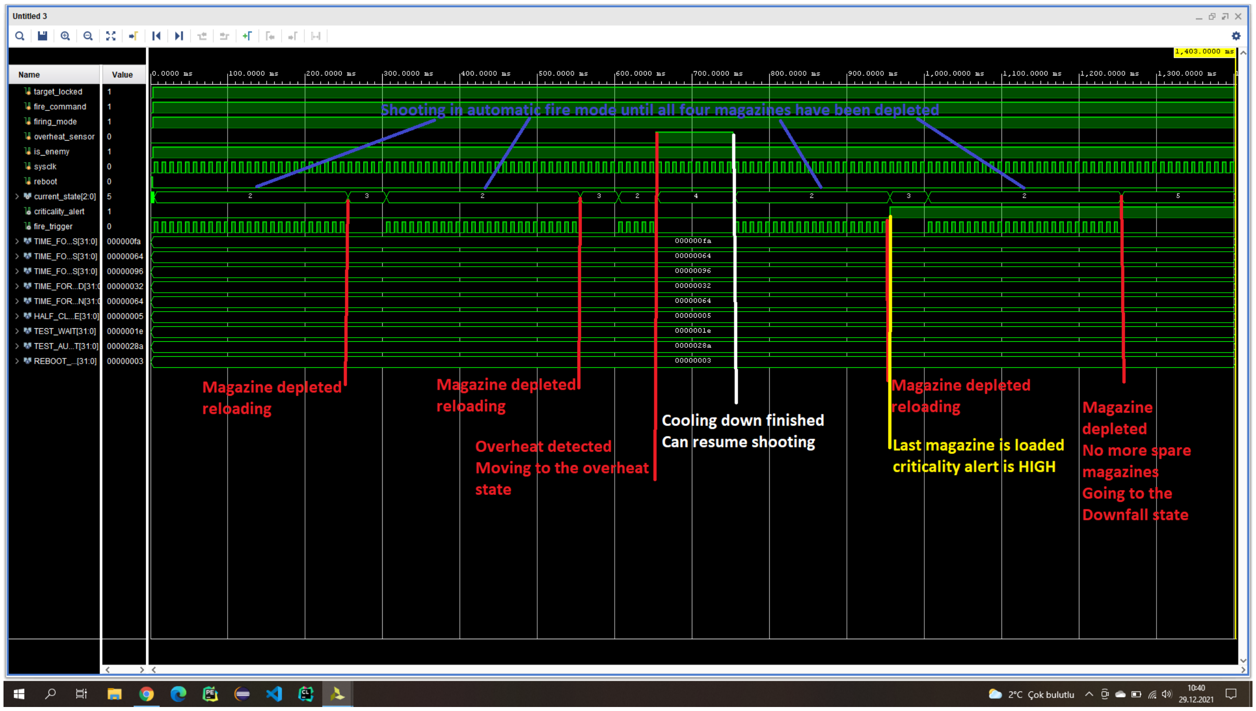
25 shots fired in single shoot mode until the magazine is empty

Magazine is empty, transition to reload state

Reload complete
Can resume shooting

Figure 3: Single Shooting Test

TEST 3: Automatic Shooting Test
Fig. 4.



Name	Value
target_locked	1
fire_command	1
firing_mode	1
overheat_sensor	0
is_enemy	1
sysclk	0
reboot	0
current_state[2:0]	5
criticality_alert	1
fire_trigger	0
TIME_FO...S[31:0]	000000fa
TIME_FO...S[31:0]	00000064
TIME_FO...S[31:0]	00000096
TIME_FOR...D[31:0	00000032
TIME_FOR...N[31:0	00000064
HALF_CL...E[31:0]	00000005
TEST_WAIT[31:0]	0000001e
TEST_AU...T[31:0]	0000028a
REBOOT_...[31:0]	00000003

Shooting in automatic fire mode until all four magazines have been depleted

Magazine depleted
reloading

Magazine depleted
reloading

Overheat detected
Moving to the overheat
state

Cooling down finished
Can resume shooting

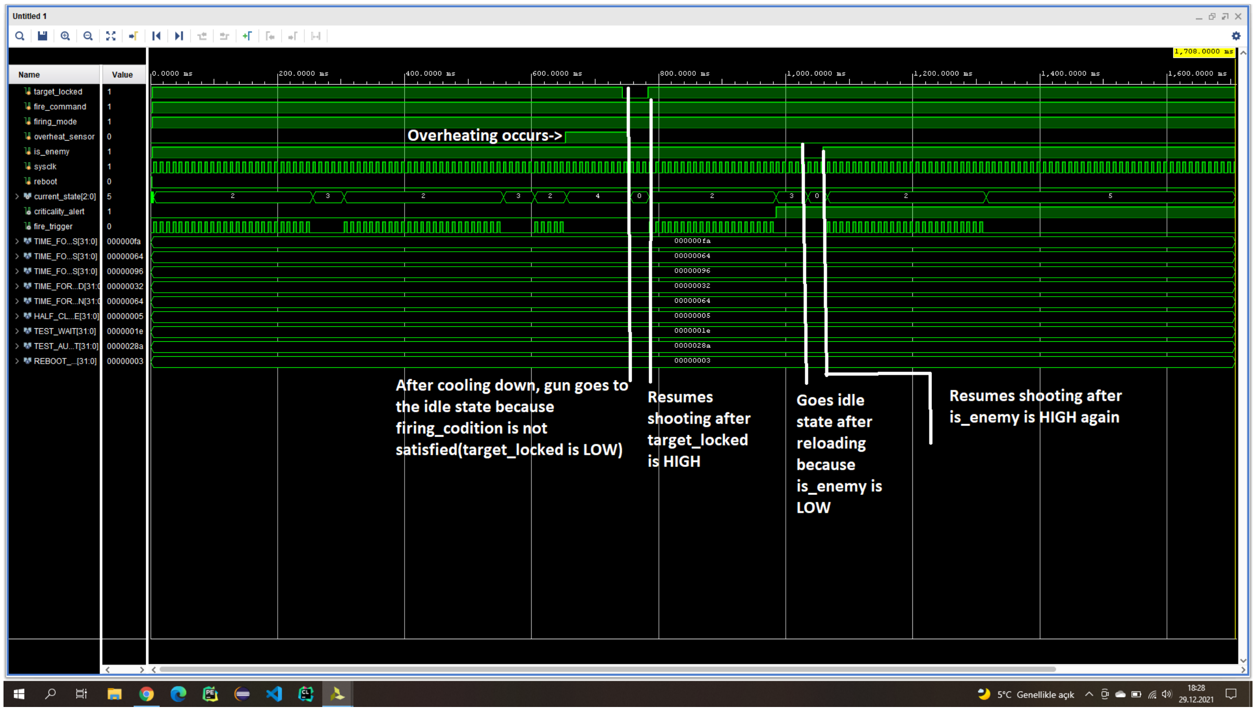
Magazine depleted
reloading

Last magazine is loaded
criticality alert is HIGH

Magazine
depleted
No more spare
magazines
Going to the
Downfall state

Figure 4: Automatic Shooting Test

TEST 4: Returning from RELOAD and OVERHEAT Test
Fig. 5.



Untitled 1

Name	Value
target_locked	1
fire_command	1
firing_mode	1
overheat_sensor	0
is_enemy	1
sysclk	1
reboot	0
current_state[2:0]	5
criticality_alert	1
fire_trigger	0
TIME_FO...S[31:0]	000000fa
TIME_FO...S[31:0]	00000064
TIME_FO...S[31:0]	00000096
TIME_FOR...D[31:0	00000032
TIME_FOR...N[31:0	00000064
HALF_CL...E[31:0]	00000005
TEST_WAIT[31:0]	0000001e
TEST_AU...T[31:0]	0000028a
REBOOT_...[31:0]	00000003

Overheating occurs->

After cooling down, gun goes to the idle state because firing_codition is not satisfied(target_locked is LOW)

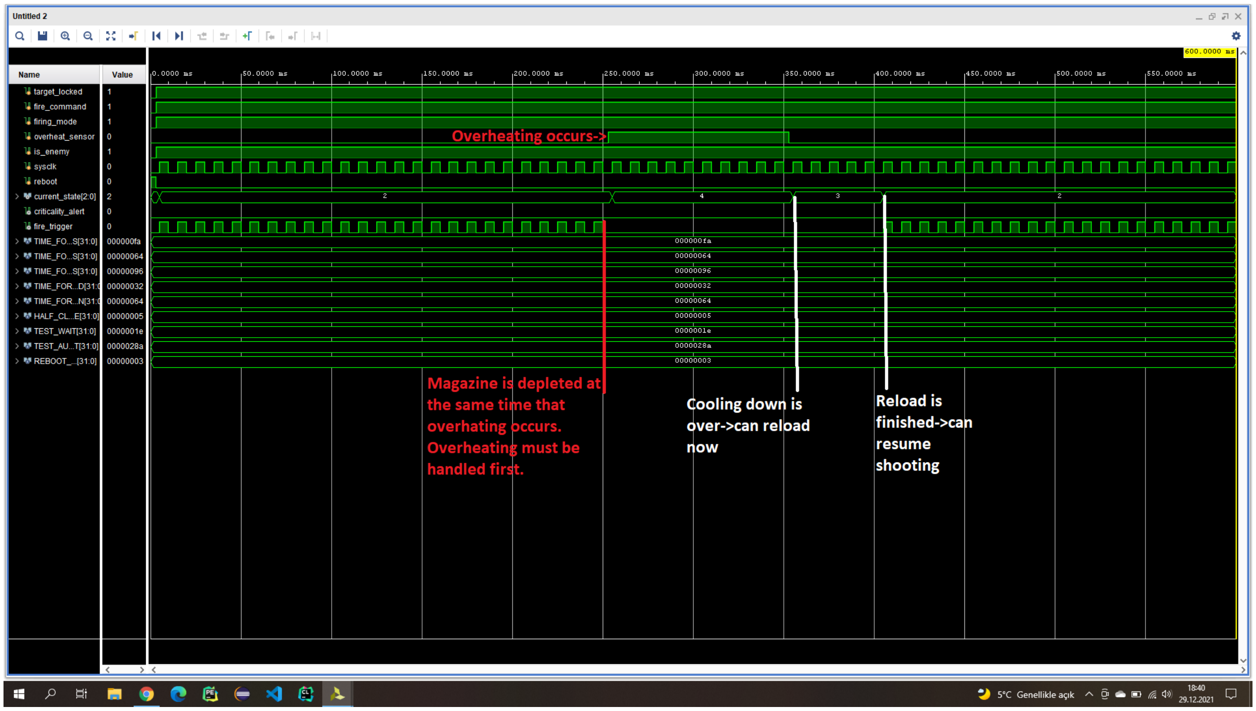
Resumes shooting after target_locked is HIGH

Goes idle state after reloading because is_enemy is LOW

Resumes shooting after is_enemy is HIGH again

Figure 5: Returning from RELOAD and OVERHEAT Test

TEST 5: RELOAD after OVERHEAT Test
Fig. 6.



Name	Value
target_locked	1
fire_command	1
firing_mode	1
overheat_sensor	0
is_enemy	1
sysclk	0
reboot	0
current_state[2:0]	2
criticality_alert	0
fire_trigger	0
TIME_FO...S[31:0]	000000fa
TIME_FO...S[31:0]	00000064
TIME_FO...S[31:0]	00000096
TIME_FOR...D[31:0	00000032
TIME_FOR...N[31:0	00000064
HALF_CL...E[31:0]	00000005
TEST_WAIT[31:0]	0000001e
TEST_AU...T[31:0]	0000028a
REBOOT_...[31:0]	00000003

Overheating occurs->

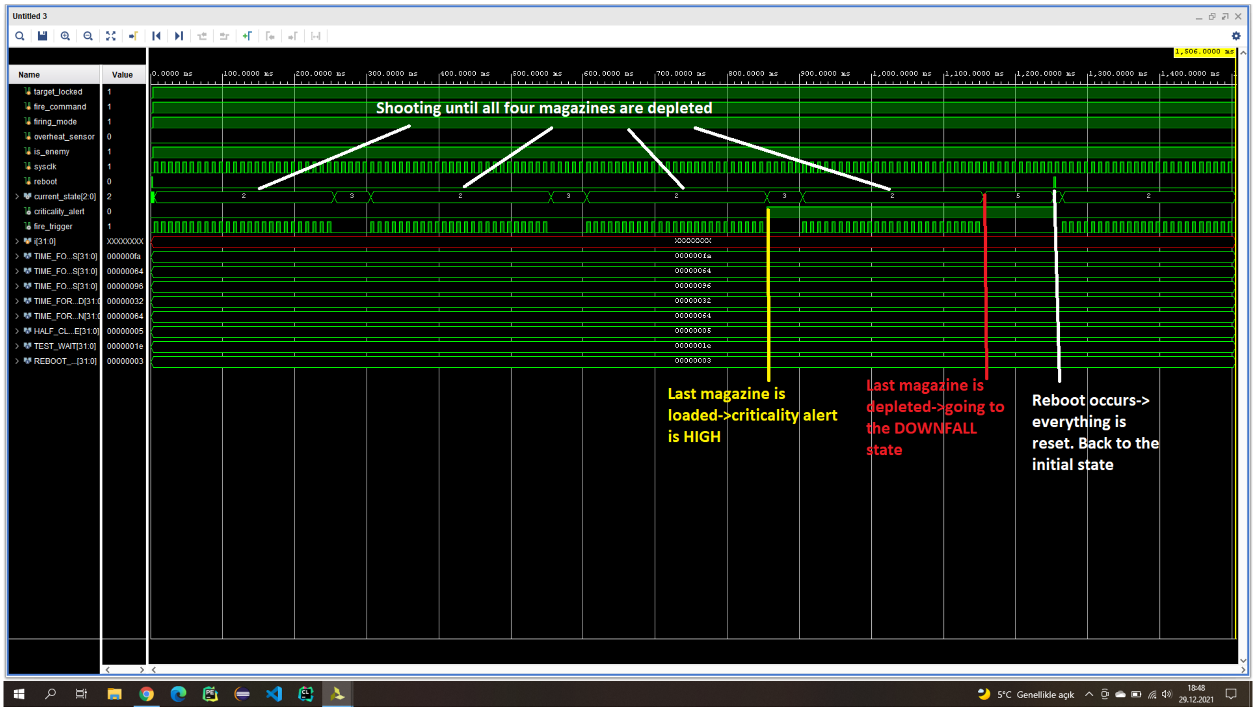
Magazine is depleted at the same time that overhating occurs. Overheating must be handled first.

Cooling down is over->can reload now

Reload is finished->can resume shooting

Figure 6: RELOAD after OVERHEAT Test

TEST 6: REBOOT Test
Fig. 7.



Name	Value
target_locked	1
fire_command	1
firing_mode	1
overheat_sensor	0
is_enemy	1
sysclk	1
reboot	0
current_state[2:0]	2
criticality_alert	0
fire_trigger	1
i[31:0]	XXXXXXXX
TIME_FO...S[31:0]	000000fa
TIME_FO...S[31:0]	00000064
TIME_FO...S[31:0]	00000096
TIME_FOR...D[31:0	00000032
TIME_FOR...N[31:0	00000064
HALF_CL...E[31:0]	00000005
TEST_WAIT[31:0]	0000001e
REBOOT_...[31:0]	00000003

Shooting until all four magazines are depleted

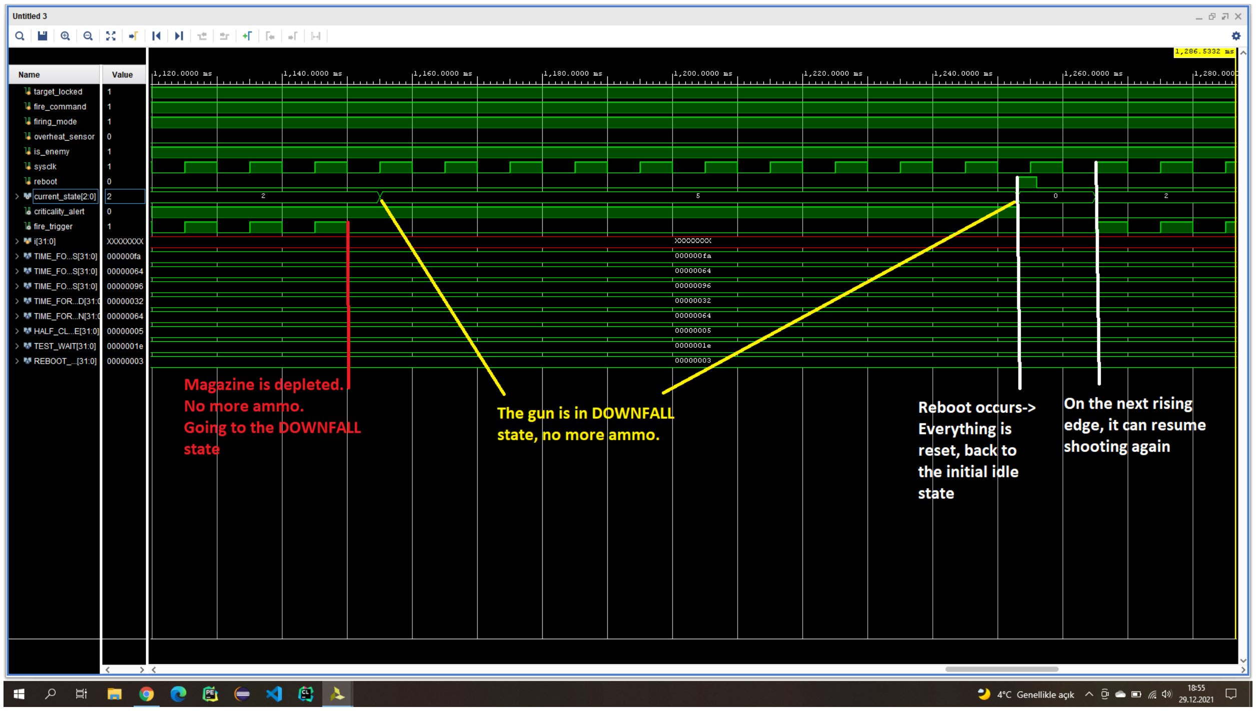
Last magazine is loaded->criticality alert is HIGH

Last magazine is depleted->going to the DOWNFALL state

Reboot occurs-> everything is reset. Back to the initial state

Figure 7: Reboot test

Fig. 8.



Name	Value
target_locked	1
fire_command	1
firing_mode	1
overheat_sensor	0
is_enemy	1
sysclk	1
reboot	0
current_state[2:0]	2
criticality_alert	0
fire_trigger	1
i[31:0]	XXXXXXXX
TIME_FO...S[31:0]	000000fa
TIME_FO...S[31:0]	00000064
TIME_FO...S[31:0]	00000096
TIME_FOR...D[31:0	00000032
TIME_FOR...N[31:0	00000064
HALF_CL...E[31:0]	00000005
TEST_WAIT[31:0]	0000001e
REBOOT_...[31:0]	00000003

Magazine is depleted.
No more ammo.
Going to the DOWNFALL state

The gun is in DOWNFALL state, no more ammo.

Reboot occurs-> Everything is reset, back to the initial idle state

On the next rising edge, it can resume shooting again

Figure 8: Reboot test(zoomed in downfall state)

References

- BBM231 Lecture Notes
- BBM233 Lecture Notes
- https://www.xilinx.com/support/documentation/university/ISE-Teaching/HDL-Design/14x/Nexys3/Verilog/docs-pdf/lab10.pdf
- https://www.smartdraw.com/software/drawing-software.htm?id=361780&gclid=CjwKCAiAiKuOBhBQ
  sK8SnD1V_1AnNXwQ1hbEYASKw2LNZy7AAkNqZWeZdYgCWAPeeST4sxRoCq4QQAvD_BwE