Hacettepe University

Computer Engineering Department

BM233 Logic Design Lab - 2021 Fall

Verilog Final Assignment

January 9, 2022

Student name:	*Student Number:*
Metin Eren Oktay	2210356137

1 Problem Definition of The Implementation with D Flip-Flops

For this project, we are expected to design a controller module for the SIGANFU_MACHINE_GUN using Verilog and behavioral design.

2 Mealy State Diagram

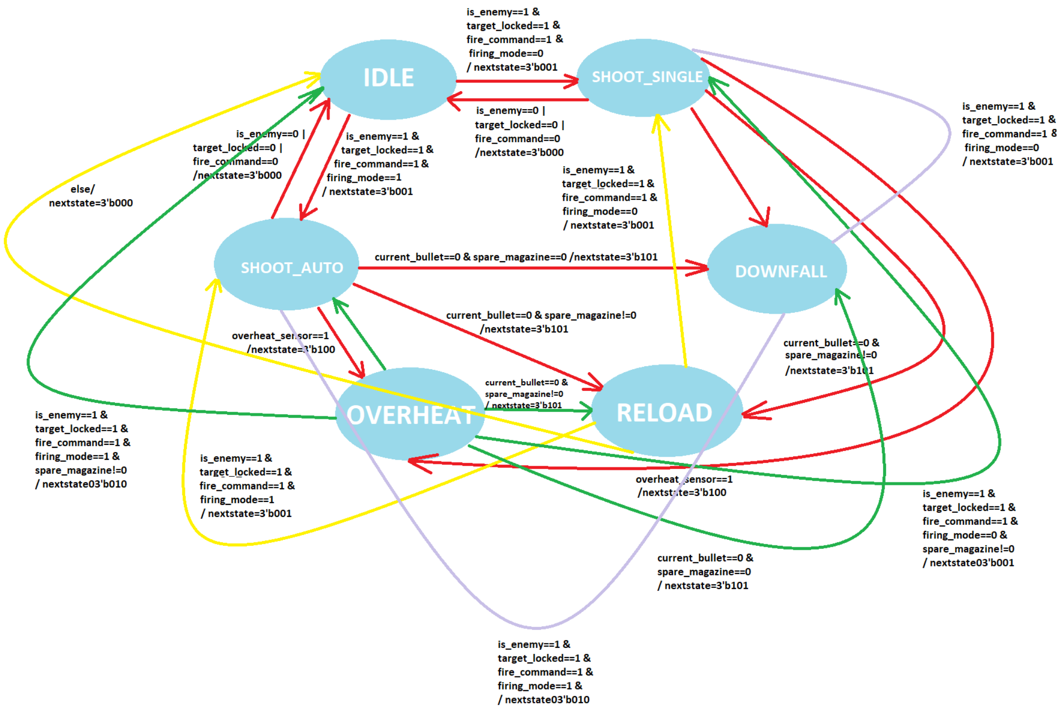

IDLE
SHOOT_SINGLE
SHOOT_AUTO
DOWNFALL
OVERHEAT
RELOAD
is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==0 / nextstate=3'b001
is_enemy==0 | target_locked==0 | fire_command==0 /nextstate=3'b000
is_enemy==0 | target_locked==0 | fire_command==0 /nextstate=3'b000
is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==1 / nextstate=3'b001
else/ nextstate=3'b000
is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==0 / nextstate=3'b001
is_enemy==1 & target_locked==1 & fire_command==1 8 firing_mode==0 / nextstate=3'b001
current_bullet==0 & spare_magazine==0 /nextstate=3'b101
current_bullet==0 & spare_magazine!=0 /nextstate=3'b101
overheat_sensor==1 /nextstate=3'b100
current_bullet==0 & spare_magazine!=0 /nextstate=3'b101
current_bullet==0 & spare_magazine!=0 / nextstate=3'b101
is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==1 & spare_magazine!=0 / nextstate03'b010
is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==1 / nextstate=3'b001
overheat_sensor==1 /nextstate=3'b100
is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==0 & spare_magazine!=0 / nextstate03'b001
current_bullet==0 & spare_magazine==0 / nextstate=3'b101
is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==1 & / nextstate03'b010

Figure 1: Transition Table for the Circuit

3 Verilog Code Solution

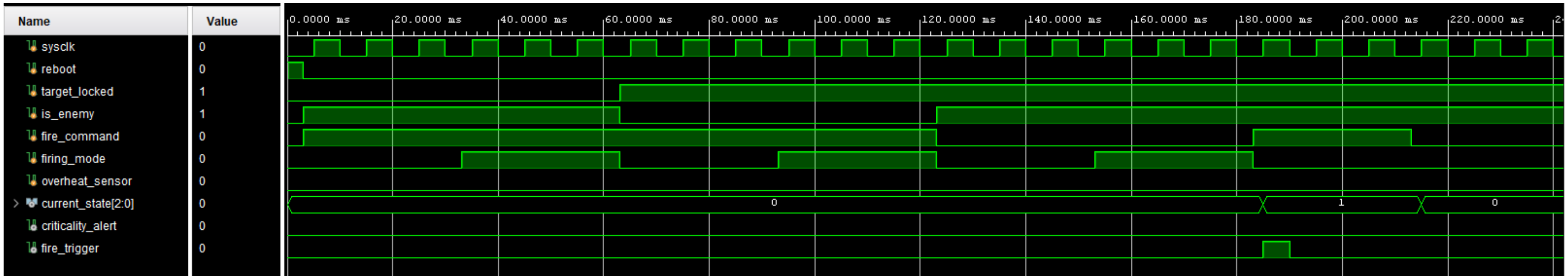
```
`timescale 1ms / 100ns

```

```
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
);
//single_previous is used for not shooting more than one bullet for the state shoot_single
reg [2:0] nextstate;
integer single_previous, current_bullet, spare_magazine, i;
parameter idle=3'b000, shoot_single=3'b001, shoot_auto=3'b010, reload=3'b011, overheat=3'b100, downfall=3'b101;

initial begin
criticality_alert=0; fire_trigger=0; single_previous=0; current_bullet=25; spare_magazine=3; end

always @(posedge sysclk or posedge reboot ) // always block to update state
if(reboot==1) begin
    current_state <= idle;
    if (sysclk==1) begin
    current_state <= nextstate;
    end end
else
    current_state <= nextstate;
always @(current_state or target_locked or is_enemy or fire_command or firing_mode or overheat_sensor)
// always block to compute both output & nextstate
begin
    case(current_state)
        idle: if(is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==0) begin
                    nextstate = shoot_single; end
                else if(is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==1) begin
                    nextstate = shoot_auto; end
                else begin
                    nextstate = idle; end
        shoot_auto:
            if(overheat_sensor==1) begin
                nextstate = overheat;  end
            else begin
                if(current_bullet==0 & spare_magazine==0) begin
                    nextstate = downfall; end
                else if(current_bullet==0 & spare_magazine!=0) begin
                    nextstate = reload; end
                else begin
                    if(is_enemy==0 | target_locked==0 | fire_command==0 ) begin
                        nextstate = idle; end
                    else begin
                        if(single_previous==0) begin
                        for(i=current_bullet; i>0; i=i-1) begin
                            if(overheat_sensor==1) begin nextstate = overheat; single_previous=0; end
                            else if(current_bullet==0) begin nextstate = reload; single_previous=0; end
                            else if (current_bullet==0 & spare_magazine==0) begin nextstate = downfall; end
                            else begin
                            fire_trigger=1; #5; fire_trigger=0; #5;
                            single_previous=single_previous+1; current_bullet=current_bullet-1;
                            if(overheat_sensor==1) begin nextstate = overheat; single_previous=0; end
                            else if(current_bullet==0) begin nextstate = reload; single_previous=0; end
                            else if (current_bullet==0 & spare_magazine==0) begin nextstate = downfall; end
                         end end

                        end
                        if(fire_command==0) begin nextstate <= idle; single_previous=0;end
                        else if(overheat_sensor==1) begin nextstate = overheat; single_previous=0; end
                        else if(current_bullet==0 & spare_magazine!=0) begin nextstate = reload;
                        single_previous=0; end
                        else if (current_bullet==0 & spare_magazine==0) begin nextstate = downfall; end
                        else begin nextstate = shoot_auto; single_previous=0; end
                    end
                end
            end
        shoot_single:
            if(overheat_sensor==1) begin
                nextstate = overheat; #100; end
            else begin
                if(current_bullet==0 & spare_magazine==0) begin
```

```
                    nextstate = downfall; end
                else if(current_bullet==0 & spare_magazine!=0) begin
                    nextstate = reload;  end
                else begin
                    if(is_enemy==0 | target_locked==0 | fire_command==0 ) begin
                        nextstate = idle; end
                    else begin
                        if(single_previous==0) begin
                        fire_trigger=1; #5; fire_trigger=0; #5; single_previous=single_previous+1;
                        current_bullet=current_bullet-1; end
                        if(overheat_sensor==1) begin nextstate = overheat; single_previous=0; end
                        else if(current_bullet==0 & spare_magazine!=0) begin nextstate = reload;
                        single_previous=0; end
                        else if (current_bullet==0 & spare_magazine==0) begin nextstate = downfall; end
                        if(fire_command==0)begin nextstate = idle; single_previous=0;end
                        else begin nextstate = shoot_single;  end
                    end if(current_bullet!=0) begin single_previous=0; end
                end
            end
        reload:
            if(spare_magazine!=0) begin
                #40; current_bullet=25; spare_magazine=spare_magazine-1;
                if(spare_magazine==0) begin criticality_alert=1; end #10;
                if(is_enemy==1 & target_locked==1 & firing_mode==0) begin
                    nextstate = shoot_single;  end
                else if(is_enemy==1 & target_locked==1 & firing_mode==1) begin
                    nextstate = shoot_auto;  end
                else begin
                    nextstate = idle; end
                end
            else begin criticality_alert=1; end
        overheat:
        if(1>0) begin #100;
            if (current_bullet==0 & spare_magazine==0) begin
                nextstate = downfall; end
            else if(current_bullet==0 & spare_magazine!=0) begin
                nextstate = reload; end
            else if(is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==0 & spare_magazine!=0) begin
                nextstate = shoot_single; end
            else if(is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==1 & spare_magazine!=0) begin
                nextstate = shoot_auto; end
            else begin
                nextstate = idle; end
        end
        downfall:
            if (1>0) begin
               #100;  current_bullet=25; spare_magazine=3;
               if(is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==0) begin
                nextstate = shoot_single; end
            else if(is_enemy==1 & target_locked==1 & fire_command==1 & firing_mode==1) begin
                nextstate = shoot_auto; end
              end
        default:
            nextstate <= idle;
    endcase
end
endmodule
```

4 Waveforms of Tests

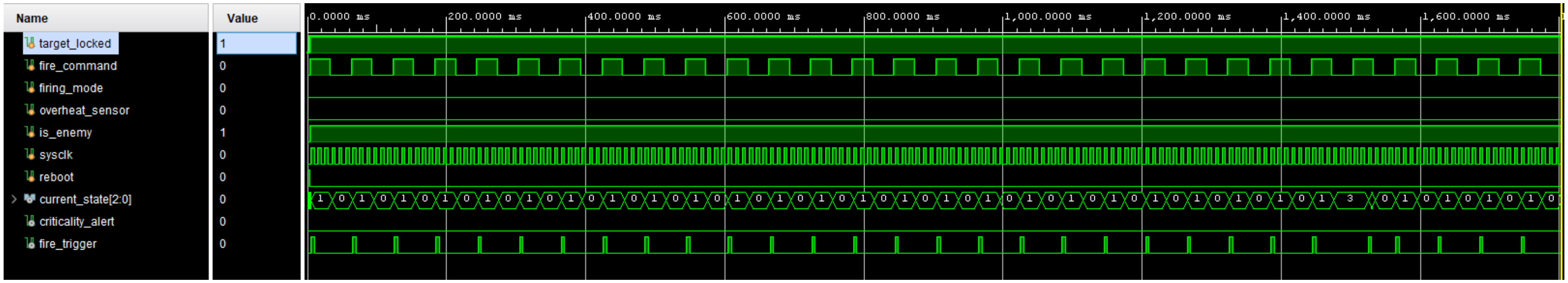
4.1 TEST 1: Safety Test



Name	Value
sysclk	0
reboot	0
target_locked	1
is_enemy	1
fire_command	0
firing_mode	0
overheat_sensor	0
current_state[2:0]	0
criticality_alert	0
fire_trigger	0

Figure 2: Waveform of the test 1

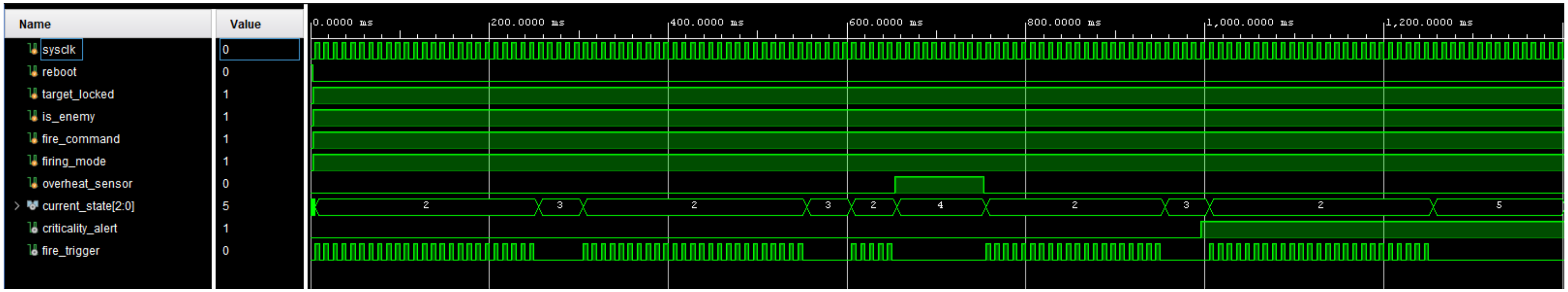
4.2 TEST 2: Single Shooting Test



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

Figure 3: Waveform of the test 2

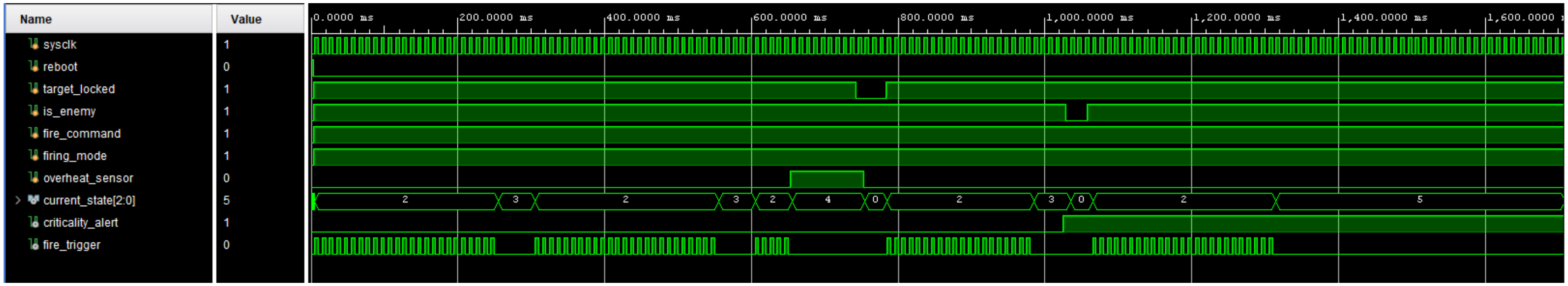
4.3 TEST 3: Automatic Shooting Test



Name	Value
sysclk	0
reboot	0
target_locked	1
is_enemy	1
fire_command	1
firing_mode	1
overheat_sensor	0
current_state[2:0]	5
criticality_alert	1
fire_trigger	0

Figure 4: Waveform of the test 3

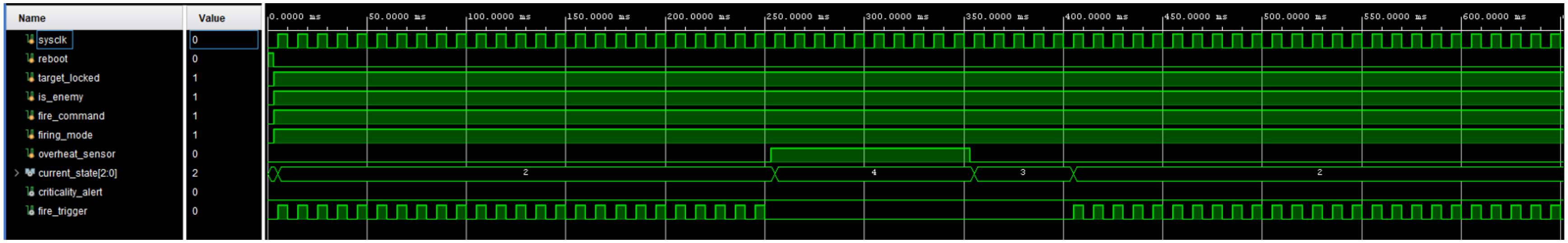
4.4 TEST 4: Returning from RELOAD and OVERHEAT Test



Name	Value
sysclk	1
reboot	0
target_locked	1
is_enemy	1
fire_command	1
firing_mode	1
overheat_sensor	0
current_state[2:0]	5
criticality_alert	1
fire_trigger	0

Figure 5: Waveform of the test 4

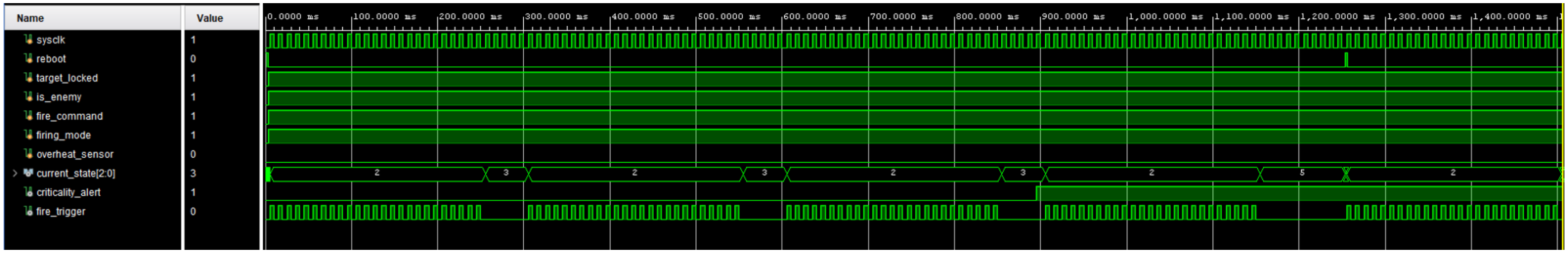
4.5 TEST 5: RELOAD after OVERHEAT Test



Name	Value
sysclk	0
reboot	0
target_locked	1
is_enemy	1
fire_command	1
firing_mode	1
overheat_sensor	0
current_state[2:0]	2
criticality_alert	0
fire_trigger	0

Figure 6: Waveform of the test 5

4.6 TEST 6: REBOOT Test



Name	Value
sysclk	1
reboot	0
target_locked	1
is_enemy	1
fire_command	1
firing_mode	1
overheat_sensor	0
current_state[2:0]	3
criticality_alert	1
fire_trigger	0

Figure 7: Waveform of the test 6

References

- One and Only Mentor