Electrical and Computer Engineering Department

Project Assessment

Advanced Digital Design (ENCS3310)

Prepared By: *Lana Aref Batnij* ***ID:****1200308*

Instructor: Dr. Abdellatif Abu-Issa

Date :*1/26/2023* ***Section :****1*

Table of content:

- ***Table of figures:***

- ***Introduction:***

In this project, we were required to construct and design a traffic Lights system, where in this system we had a highway and farm road intersection, and it was my job to create a controller for this traffic.

The controller traffic light had an output Signal consisting of 2 bits, these bits represent the colors that will be shown in this traffic light. if the value of the bit is 00 then the light will be green, else if it is 01 it will display Yellow, but it is the yellow that came after the green, otherwise, if it is 10 it will be colored in Red, last but not least if the value of the bits equal 11 then it will be Yellow that came after the red.

There is a State consistent of 18 state table that illustrates each light had a color and the time delay for each state.

State	Highway TL1	Highway TL2	Farm TL1	Farm TL2	Delay [Sec]
S0	Red	Red	Red	Red	1
S1	Red-Yellow	Red-Yellow	Red	Red	2
S2	Green	Green	Red	Red	30
S3	Green	Yellow	Red	Red	2
S4	Green	Red	Red	Red	10
S5	Yellow	Red	Red	Red	2
S6	Red	Red	Red	Red	1
S7	Red	Red	Red-Yellow	Red-Yellow	2
S8	Red	Red	Green	Green	15
S9	Red	Red	Green	Yellow	2
S10	Red	Red	Green	Red	5
S11	Red	Red	Yellow	Red-Yellow	2
S12	Red	Red	Red	Green	10
S13	Red	Red	Red	Yellow	2
S14	Red	Red	Red	Red	1
S15	Red	Red-Yellow	Red	Red	2
S16	Red	Green	Red	Red	15
S17	Red	Yellow	Red	Red	3

Figure 1: State Table of the Traffic Light System

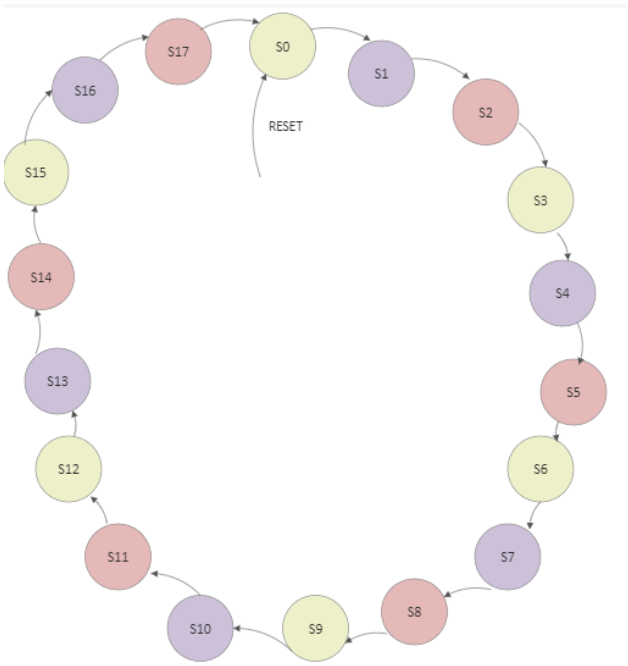

Figure 2: The State diagram

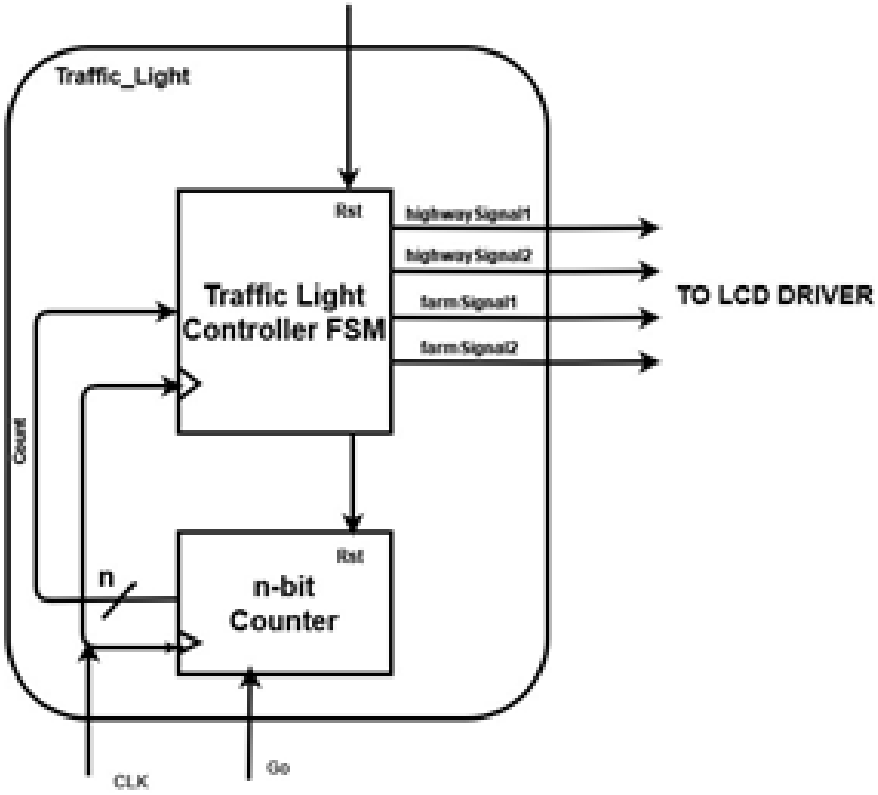

Figure 3: the design diagram

- ***The design philosophy:***

My philosophy is to Follow the Design Diagram and the states table while building the System, first of all, I Start by creating a counter with n-bit, where I decide that n should equal 5 bits because we have 18 states, and 2^5 =32 state and this would be enough to cover the number of our states.

The counter counts from 0 to 17 and it has 3 Signals, the first one is the clock, which is the Signal of the Synchronous system, and the clock that I use here is a positive edge clock, on the other hand, the counter has a Reset Signal, and it is also working on a positive edge, that's mean when the Reset =1, go back to the initial state which is S0 and let the counter Start over again, the third signal is called Go, which refers to, if go=1, then let the counter count normally and change between different states, and if they go =0 then Freeze the system.

```
module Counter_n_bit(go,clk,reset,out);
    input go,clk,reset;
    output reg[4:0] out;
    integer i;

    always @(posedge clk or posedge reset)
        begin
        if(reset!=1)begin
            if (go==1)begin
               for(i=0;i<18;i=i+1)begin
                   out[i]=out[i]+5'b00001;
                    $monitor("the counter value %d",out[i]);
                end
            end
        end
        else if (reset ==1)begin
            i=0;
        end
    end

endmodule
```

Figure 4: the counter Verilog Code

After I finish the counter, I Start implementing the Traffic Light System depending on our state diagram. This module has 3 input signals which are the clock reset and the counter output will be inputted here, I n addition we also have 4 outputs from this system, each of them representing one of the traffic lights, which is the first highway, the sconed one the first Farm and the second one.

When I first implement the module, I just define the inputs and the outputs, then I create an instance from the counter module, the counter results can be used correctly in this module, I preferred here to use the sequential Way of writing the code, that means I just used the (Always statement) and the condition was on the positive edge of the clock or the positive edge of the Reset, go in the statement and start working.

I used the for loop that counts from 0 to 17, which is the number of states, inside the for there are multiple if-else statements, and each if a check is a counter equal to a number this number represents the state I am in, and according to it I assigned a value for the output.

For example, if the counter value is 2, then the System is in the second state, and the lights would be Green, Green for the highways, and Red, Red for the farm.

```
module Traffic_Light(clk,reset,count_out,Hs1,Hs2,Fs1,Fs2);
    parameter Green=2'b00,Yellow_g=2'b01,Red=2'b10,Yellow_r=2'b11;
    input clk,reset;
    input [4:0] count_out;
    output reg Hs1,Hs2,Fs1,Fs2;
    integer i;

    Counter_n_bit My_Counter(1,clk,reset,count_out);

    always @(posedge clk or posedge reset)
        begin
            for(i=0;i<18;i=i+1)
                begin
                    if(count_out[i]==0)
                        begin
                            Hs1=Red;
                            Hs2=Red;
                            Fs1=Red;
                            Fs2=Red;
                            $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
                        end
                    else if(count_out[i]==1)
                        begin
                            Hs1=Yellow_r;
                            Hs2=Yellow_r;
                            Fs1=Red;
                            Fs2=Red;
                            $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
                        end
                    else if(count_out[i]==2)
                        begin
                            Hs1=Green;
                            Hs2=Green;
                            Fs1=Red;
                            Fs2=Red;
                            $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
                        end
```

Figure 5.1: The System code

```
                    else if(count_out[i]==4)
                        begin
                            Hs1=Green;
                            Hs2=Red;
                            Fs1=Red;
                            Fs2=Red;
                            $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
                        end
                    else if(count_out[i]==5)
                        begin
                            Hs1=Yellow_g;
                            Hs2=Red;
                            Fs1=Red;
                            Fs2=Red;
                            $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
                        end
                    else if(count_out[i]==6)
                        begin
                            Hs1=Red;
                            Hs2=Red;
                            Fs1=Red;
                            Fs2=Red;
                            $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
                        end
                    else if(count_out[i]==7)
                        begin
                            Hs1=Red;
                            Hs2=Red;
                            Fs1=Yellow_r;
                            Fs2=Yellow_r;
                            $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
                        end
                    else if(count_out[i]==8)
                        begin
                            Hs1=Red;
                            Hs2=Red;
                            Fs1=Green;
                            Fs2=Green;
                            $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
```

Figure 5.2: The System code

```
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end
        else if(count_out[i]==11)
            begin
                Hs1=Red;
                Hs2=Red;
                Fs1=Yellow_g;
                Fs2=Yellow_r;
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end
        else if(count_out[i]==12)
            begin
                Hs1=Red;
                Hs2=Red;
                Fs1=Red;
                Fs2=Green;
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end
        else if(count_out[i]==13)
            begin
                Hs1=Red;
                Hs2=Red;
                Fs1=Red;
                Fs2=Yellow_g;
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end

        else if(count_out[i]==14)
            begin
                Hs1=Red;
                Hs2=Red;
                Fs1=Red;
                Fs2=Red;
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end
        else if(count_out[i]==15)
            begin
                Hs1=Red;
                Hs2=Yellow_r;
```

Figure 5.3: The System code

```
                Fs1=Red;
                Fs2=Red;
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end
        else if(count_out[i]==15)
            begin
                Hs1=Red;
                Hs2=Yellow_r;
                Fs1=Red;
                Fs2=Red;
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end
        else if(count_out[i]==16)
            begin
                Hs1=Red;
                Hs2=Green;
                Fs1=Red;
                Fs2=Red;
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end
        else if(count_out[i]==17)
            begin
                Hs1=Red;
                Hs2=Yellow_r;
                Fs1=Red;
                Fs2=Red;
                $display("THE VALUES OF THE OF THE Highway1=%d,Highway2=%d,Farm1=%d, Farm2=%d",Hs1,Hs2,Fs1,Fs2);
            end

    end

end

endmodule
```

Figure 5.4: The System code

After implementing this system I started the Verification for my design, instead of building a generator module I apply it in the testbench module, and I create a module that is the reference and in the same module there is an analyzer, and his job just to see if the values of the System generator is equal to the reference values, which is the right values, then it will print a message on the console that everything is fine, and the values are right, otherwise it will print the error.

```
module reference_And_Analysiser(clk,reset,count_out,Hs1,Hs2,Fs1,Fs2,mem_Hs1,mem_Hs2,mem_Fs1,mem_Fs2);
    parameter Green=2'b00,Yellow_g=2'b01,Red=2'b10,Yellow_r=2'b11;

    input clk,Hs1,Hs2,Fs1,Fs2;
    input [4:0] count_out;
    integer i;

    Counter_n_bit counter1(1,clk,reset,count_out);
    Traffic_Light mySystem(clk,reset,count_out,Hs1,Hs2,Fs1,Fs2);

    output reg [1:0] mem_Hs1[0:17] ={Red,Yellow_r,Green,Green,Green,Yellow_g
    ,Red,Red,Red,Red,Red,Red,Red,Red,Red,Red,Red,Red};

    output reg [1:0] mem_Hs2[0:17] ={Red,Yellow_r,Green,Yellow_g,Red,Red
    ,Red,Red,Red,Red,Red,Red,Red,Red,Red,Yellow_r,Green,Yellow_g};

    output reg [1:0] mem_Fs1[0:17] ={Red,Red,Red,Red,Red,Red
    ,Red,Yellow_r,Green,Green,Green,Yellow_g,Red,Red,Red,Red,Red,Red};

    output reg [1:0] mem_Fs2[0:17] ={Red,Red,Red,Red,Red,Red
    ,Red,Yellow_r,Green,Yellow_g,Red,Yellow_r,Green,Yellow_g,Red,Red,Red,Red};

    for(i=0;i<18;i=i+1)
        begin
            if(Hs1==mem_Hs1[i])
                $display("THE VALUSE OF THE FIRST HIGHWAY ARE TRUE %d",Hs1[i]);
            else
                $display("THE VALUSE OF THE FIRST HIGHWAY ARE FALSE %d",Hs1[i]);
        end

    for(i=0;i<18;i=i+1)
        begin
            if(Hs2==mem_Hs2[i])
                $display("THE VALUSE OF THE SECOND HIGHWAY ARE TRUE %d",Hs2[i]);
            else
                $display("THE VALUSE OF THE SECOND HIGHWAY ARE FALSE %d",Hs2[i]);
        end
```

Figure 6.1: The Analyzer and reference code

```
    output reg [1:0] mem_Fs1[0:17] ={Red,Red,Red,Red,Red,Red
    ,Red,Yellow_r,Green,Green,Green,Yellow_g,Red,Red,Red,Red,Red,Red};

    output reg [1:0] mem_Fs2[0:17] ={Red,Red,Red,Red,Red,Red
    ,Red,Yellow_r,Green,Yellow_g,Red,Yellow_r,Green,Yellow_g,Red,Red,Red,Red};

    for(i=0;i<18;i=i+1)
        begin
            if(Hs1==mem_Hs1[i])
                $display("THE VALUSE OF THE FIRST HIGHWAY ARE TRUE %d",Hs1[i]);
            else
                $display("THE VALUSE OF THE FIRST HIGHWAY ARE FALSE %d",Hs1[i]);
        end

    for(i=0;i<18;i=i+1)
        begin
            if(Hs2==mem_Hs2[i])
                $display("THE VALUSE OF THE SECOND HIGHWAY ARE TRUE %d",Hs2[i]);
            else
                $display("THE VALUSE OF THE SECOND HIGHWAY ARE FALSE %d",Hs2[i]);
        end
    for(i=0;i<18;i=i+1)
        begin
            if(Fs1==mem_Fs1[i])
                $display("THE VALUSE OF THE FIRST FARM ARE TRUE %d ",Fs1[i]);
            else
                $display("THE VALUSE OF THE FIRST FARM ARE FALSE %d" ,Fs1[i]);
        end
    for(i=0;i<18;i=i+1)
        begin
            if(Fs2==mem_Fs2[i])
                $display("THE VALUSE OF THE SECOND FARM ARE TRUE %d" ,Fs2[i]);
            else
                $display("THE VALUSE OF THE SECOND FARM ARE FALSE %d" ,Fs2[i]);
        end


endmodule
```

Figure 6.2: The Analyzer and reference code

Finally, I implement the testbench for the system, at the beginning of this I define the inputs and outputs of the systems as registers and wires, then I make an instance of the counter the system, and the analyzer, and then I apply the delay on each, and what I realized that the state that has an odd number then its delay is 2 seconds while the even ones each have different delays.

```
//_____________________________________________________________________

module System_TB;//the generator module inside the Testbench
    reg go,clk,reset;
    reg [4:0] Fs2,mem_Hs1,mem_Hs2,mem_Fs1,mem_Fs2,count_ou;
    wire Hs1,Hs2,Fs1,Fs2;
    integer i;

    Counter_n_bit counter1(go,clk,reset,count_out);
    Traffic_Light mySystem(clk,reset,count_out,Hs1,Hs2,Fs1,Fs2);
    reference_And_Analysiser(clk,reset,count_out,Hs1,Hs2,Fs1,Fs2,mem_Hs1,mem_Hs2,mem_Fs1,mem_Fs2);

    always
        begin
            repeat(200);
            #5s clk=~clk;
        end
    initial
        begin
            for(i=0;i<18;i=i+1)
                begin
                    if((i%2!=0)&(i!=17))
                        begin
                            #2s clk=~clk;
                        end
                    else if(i==0)
                        begin
                            #1s clk=~clk;
                        end
                    else if(i==2)
                        begin
                            #30s clk=~clk;
                        end
                    else if(i==4)
                        begin
                            #10s clk=~clk;
```

Figure 7.1: The Testbench code

```
                        begin
                            #15s clk=~clk;
                        end
                    else if(i==10)
                        begin
                            #5s clk=~clk;
                        end
                    else if(i==12)
                        begin
                            #10s clk=~clk;
                        end
                    else if(i==14)
                        begin
                            #1s clk=~clk;
                        end
                    else if(i==16)
                        begin
                            #15s clk=~clk;
                        end
                    else if(i==17)
                        begin
                            #3s clk=~clk;
                        end

            end
```

Figure 7.2: The Testbench code

- ***Results:***

Since my code of mine does not work properly, because I had some Kind of error, I don't have clear results of this System, but if it was working as it supposes then the colors of the lights would be displayed as the states table explains, and the result would be noticed by the waveform and the consoles.

- ***Conclusion and future work:***

To Sum up, I just implement a System that works as a traffic light controller, and it was a simple type of controller that works (you can walk only one way), in the future, we can improve it to be more realistic, that would be like, the car that stops at this traffic light would go to any of three other directions.