PostLab 7

```
module NESInterface(clock, data, latch, pulse, buttons, egg_led);

        input clock, data;

        output[6:0] egg_led;

        output reg latch, pulse;

        output reg[7:0] buttons;

        reg[19:0] count;

        //counts from 1 - 2^(19+1), approx 47 Hz cloc. Counts up every 20 nanoseconds.


        always@(posedge clock)

                begin

                count <= count + 1;

                latch <= count<=600;                                                    //The first 12 us of each 47 Hz polling

                pulse <= (count>900  && count <=1200) //First 6 us pulse, 6 us after latch

                                        ||(count>1500 && count <=1800)       //Second pulse, etc.

                                        ||(count>2100 && count <=2400)

                                        ||(count>2700 && count <=3000)

                                        ||(count>3300 && count <=3600)

                                        ||(count>3900 && count <=4200)

                                        ||(count>4500 && count <=4800)

                                        ||(count>5100 && count <=5400);

                if(count==600)

                        buttons[7] <= !data; //A

                if(count==1200)

                        buttons[6] <= !data; //B

                if(count==1800)

                        buttons[5] <= !data;    //select

                if(count==2400)
```

```
            buttons[4] <= !data;      //start
        if(count==3000)
            buttons[3] <= !data;      //up
        if(count==3600)
            buttons[2] <= !data; //down
        if(count==4200)
            buttons[1] <= !data; //left
        if(count==4800)
            buttons[0] <= !data;      //right
        end

        easter_egg my_egg(clock, buttons, egg_led);


endmodule

module easter_egg(clock, buttons, egg_led);
    input clock;
    input[7:0] buttons;
    output reg [6:0] egg_led;

    //CODE: OFF RIGHT OFF UP OFF B OFF A OFF A OFF LEFT -> egg_led HIGH
    //STATE:         1     2     3    4    5    6

    parameter ALL_OFF =           8'b00000000;
    parameter ONLY_A =            8'b10000000;
    parameter ONLY_B =            8'b01000000;
    parameter ONLY_SELECT = 8'b00100000;
    parameter ONLY_START =        8'b00010000;
    parameter ONLY_UP =         8'b00001000;
    parameter ONLY_DOWN =       8'b00000100;
    parameter ONLY_LEFT =       8'b00000010;
    parameter ONLY_RIGHT =      8'b00000001;

    reg[3:0] S, NS;
    //NEXT STATE LOGIC
    always@*
        begin
            if(S == 0)
                begin
                    if(buttons == ONLY_RIGHT)
                            NS = S+1;
                    else if(buttons == ALL_OFF)
                            NS = S;
                    else
                            NS = 0;
                end
            else if(S == 1)
                begin
                    if(buttons == ONLY_UP)
                            NS = S+1;
                    else if(buttons == ALL_OFF || buttons == ONLY_RIGHT)
                            NS = S;
                    else
                            NS = 0;
                end
            else if(S == 2)
                begin
            if(buttons == ONLY_B)
                    NS = S+1;
            else if(buttons == ALL_OFF|| buttons == ONLY_UP)
                    NS = S;
            else
                    NS = 0;
        end
    else if(S == 3)
        begin
            if(buttons == ONLY_A)
                    NS = S+1;
            else if(buttons == ALL_OFF|| buttons == ONLY_B)
                    NS = S;
            else
                    NS = 0;
        end
    else if(S == 4)
        begin
            if(buttons == ALL_OFF)
                    NS = S+1;
            else if(buttons == ONLY_A)
                    NS = S;
            else
                    NS = 0;
        end
    else if(S == 5)
        begin
            if(buttons == ONLY_A)
                    NS = S+1;
                    else if(buttons == ALL_OFF)
                            NS = S;
                    else
                            NS = 0;
                end
            else if(S == 6)
                begin
                    if(buttons == ONLY_LEFT)
                            NS = S+1;
                    else if(buttons == ALL_OFF|| buttons == ONLY_A)
                            NS = S;
                    else
                            NS = 0;
                end
            else if(S == 7)
                begin
                    if(buttons == ONLY_START)
                            NS = 0;
                    else
                            NS = S;
                end
            else
                NS = 0;
        end
    //NEXT STATE ASSIGNMENT
    always@(posedge clock)
        begin
            S <= NS;
        end
        //OUTPUT LOGIC
        always@*
            begin
                if(S==0)
                    egg_led = 7'b0000000;
                else if(S==1)
                    egg_led = 7'b0000001;
                else if(S==2)
                    egg_led = 7'b0000011;
                else if(S==3)
                    egg_led = 7'b0000111;
                else if(S==4)
                    egg_led = 7'b0001111;
                else if(S==5)
                    egg_led = 7'b0011111;
                else if(S==6)
                    egg_led = 7'b0111111;
                else if(S==7)
                    egg_led = 7'b1111111;
                else
                    egg_led = 7'b0101010;
            end
    endmodule
```