Electronic Voting Machine

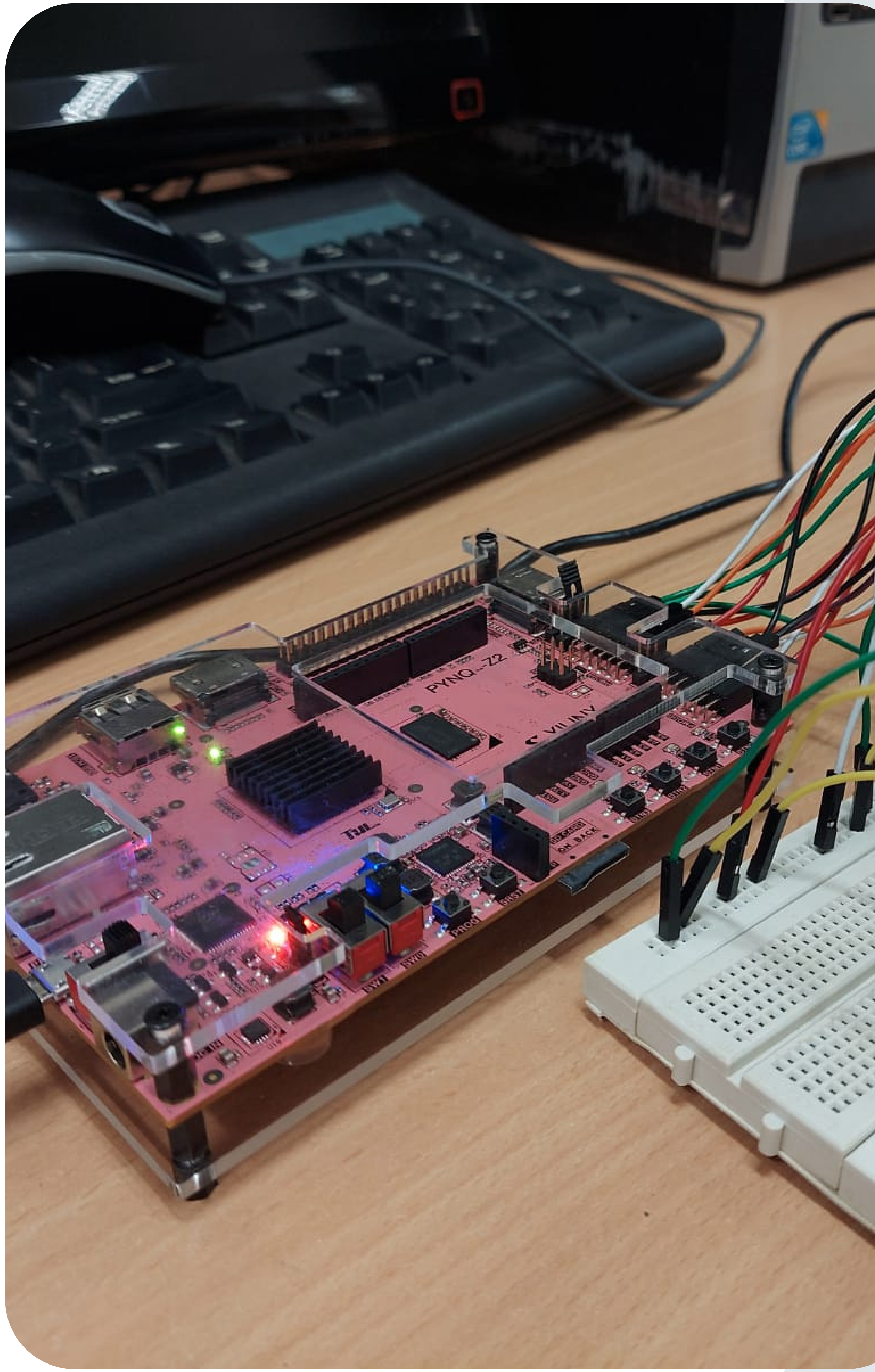


Table of Contents

Part 1:
Brief about EVM

Part 2:
Details of our work

Part 3:
Implementation

Part 4:
Block Diagram

Part 5:
Logic and explaination

Part 6:
FPGA board video

Brief about EVM

In this project, an Electronic Voting Machine (EVM) system is proposed which is in operation as transparent as the digital system. The Simplified Electronic Voting Machine (SEVM) responds on some flow of pulses coming from the switch operated by voter and produces the output of the counting values i.e. total casting votes of individual nominee and displays it. The machine is controlled both automatically and manually to operate the system for successive voters and to ensure that a voter can give only one vote to his/her chosen candidate of the same position

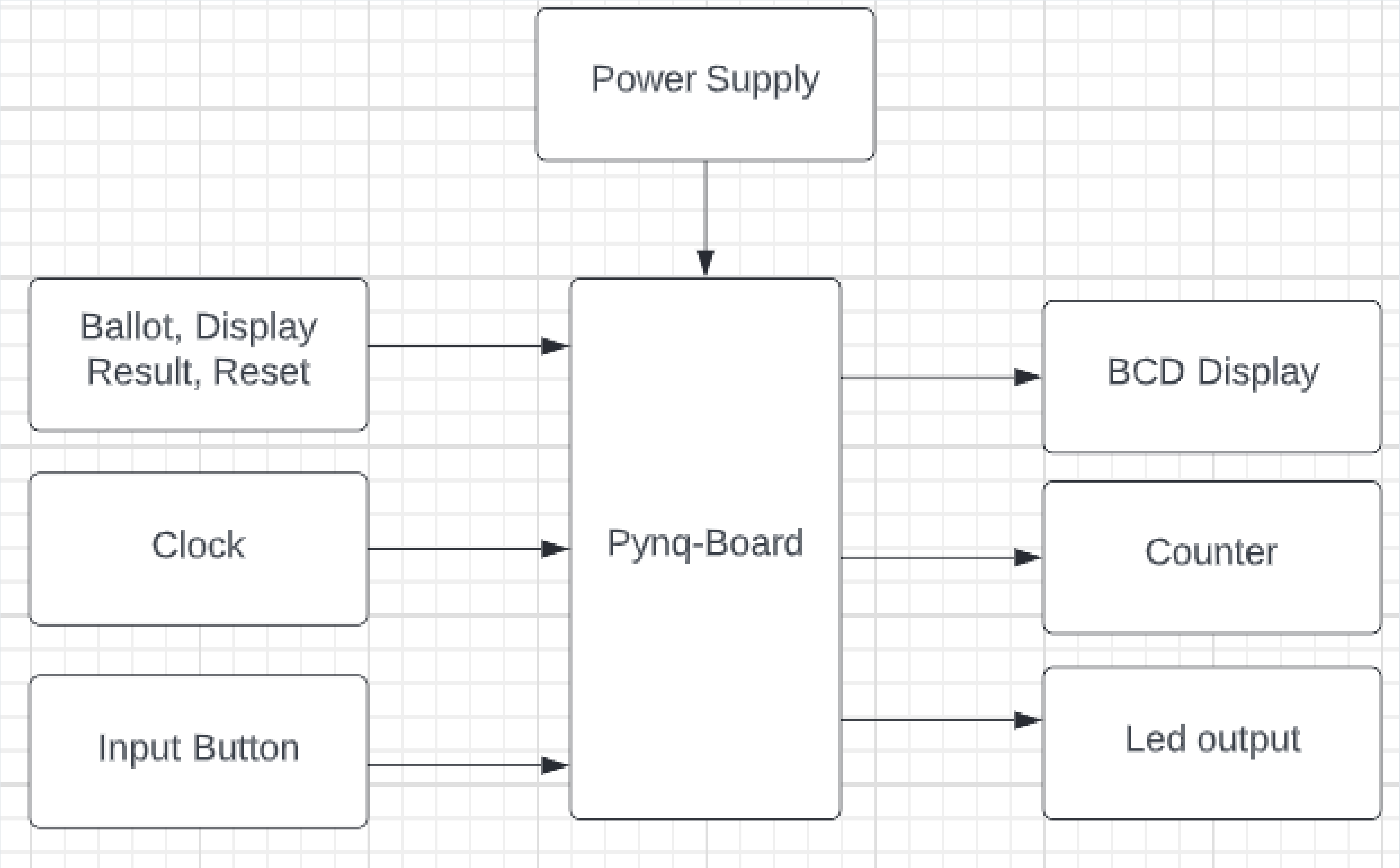
DETAILS OF OUR WORK - IMPLEMENTATION

Designed and implemented an electronic voting machine system that on the positive edge of the clock, checks if any of the buttons is pressed, if yes increases the count of votes of the candidate corresponding to the button, and blinks the LED in front of the candidate button. This entire functioning takes place only if the ballot is on. A Reset button to reset the counter of each candidate votes and a BCD to display the votes of each candidate turn by turn.

Major Functionalities

- Maximum 4 Candidates
- A ballot, Reset, display result button.
- Clock, BCD to display votes, etc.
- Buttons and LEDs correspond to each candidate.

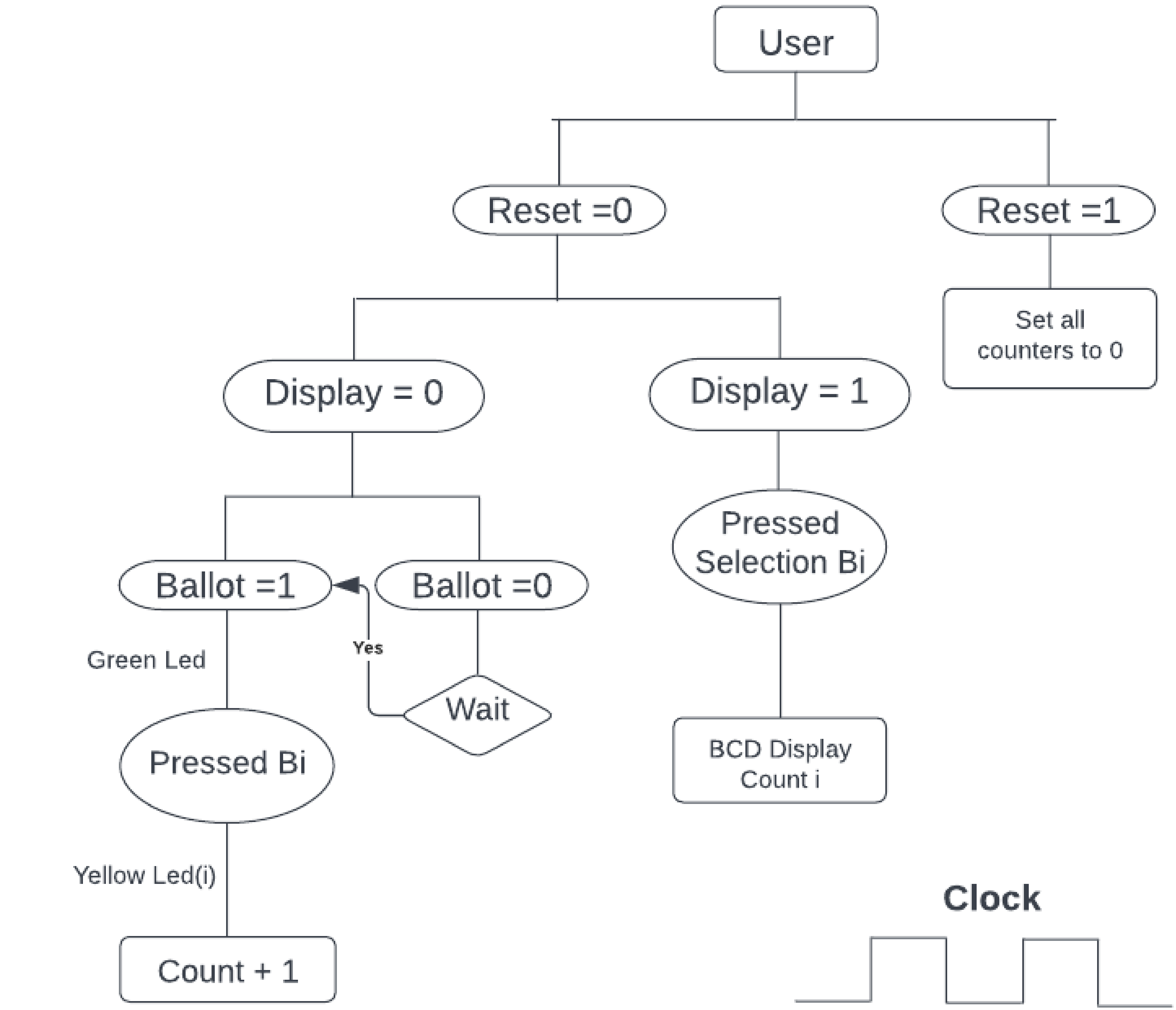
Block Diagram

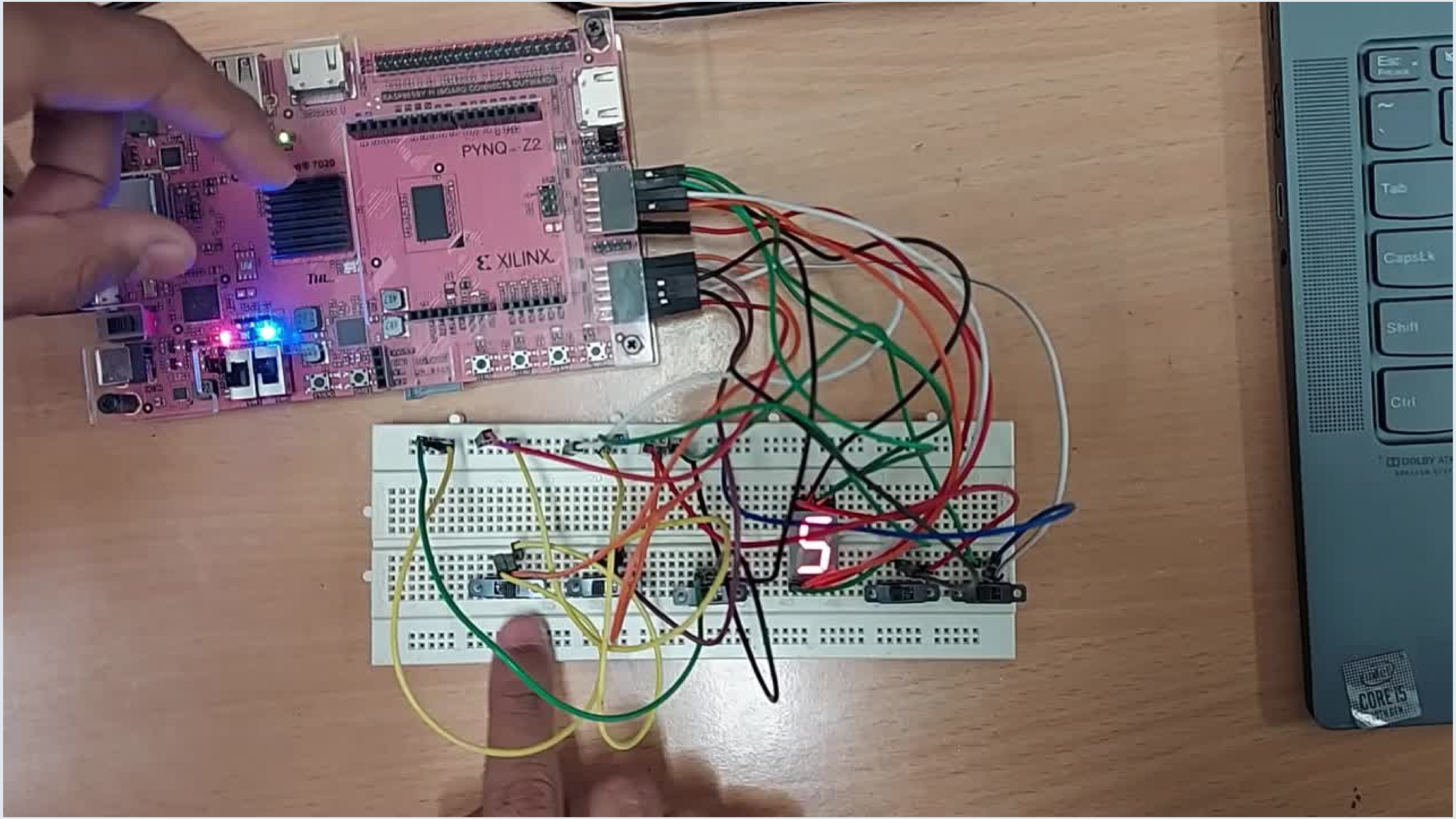


LOGIC AND EXPLANATION

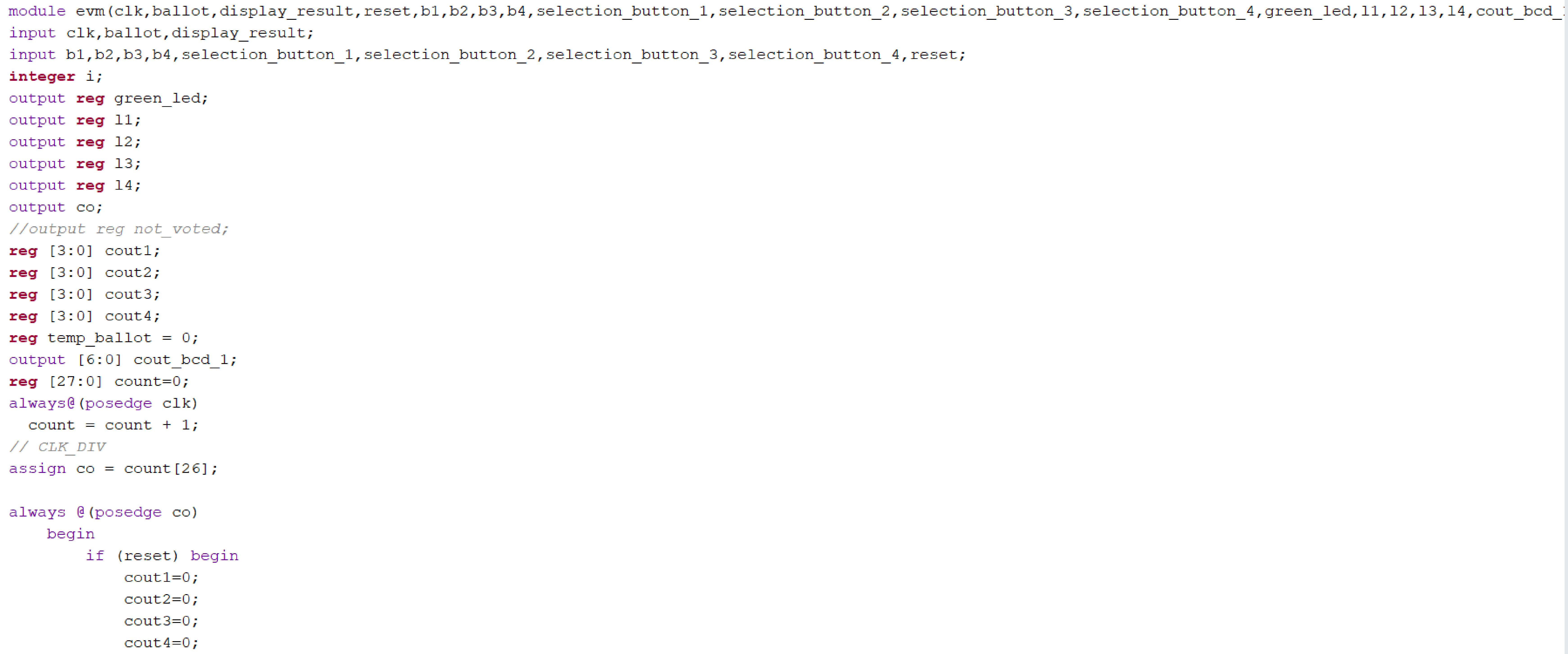
- We tried to keep the logic simple so that it can be easily implemented and explained.
- We used 1 button and LED corresponding to each candidate, and a clock to regulate the counter of the button.
- A ballot as mentioned, to give controls to the host and a display result button, to show the output results on BCD moderated by selection buttons to select which count to show.
- A reset button to reset the count to zero.
- When reset is off and the ballot is on and let's say button 1 is pressed, then on the positive edge of the clock led 1 blinks, and counter 1 is increased by 1, same for every button, led and counter.
- When the display result is on, and the selection button i is on the court of votes for candidate i is shown.

Flow chart

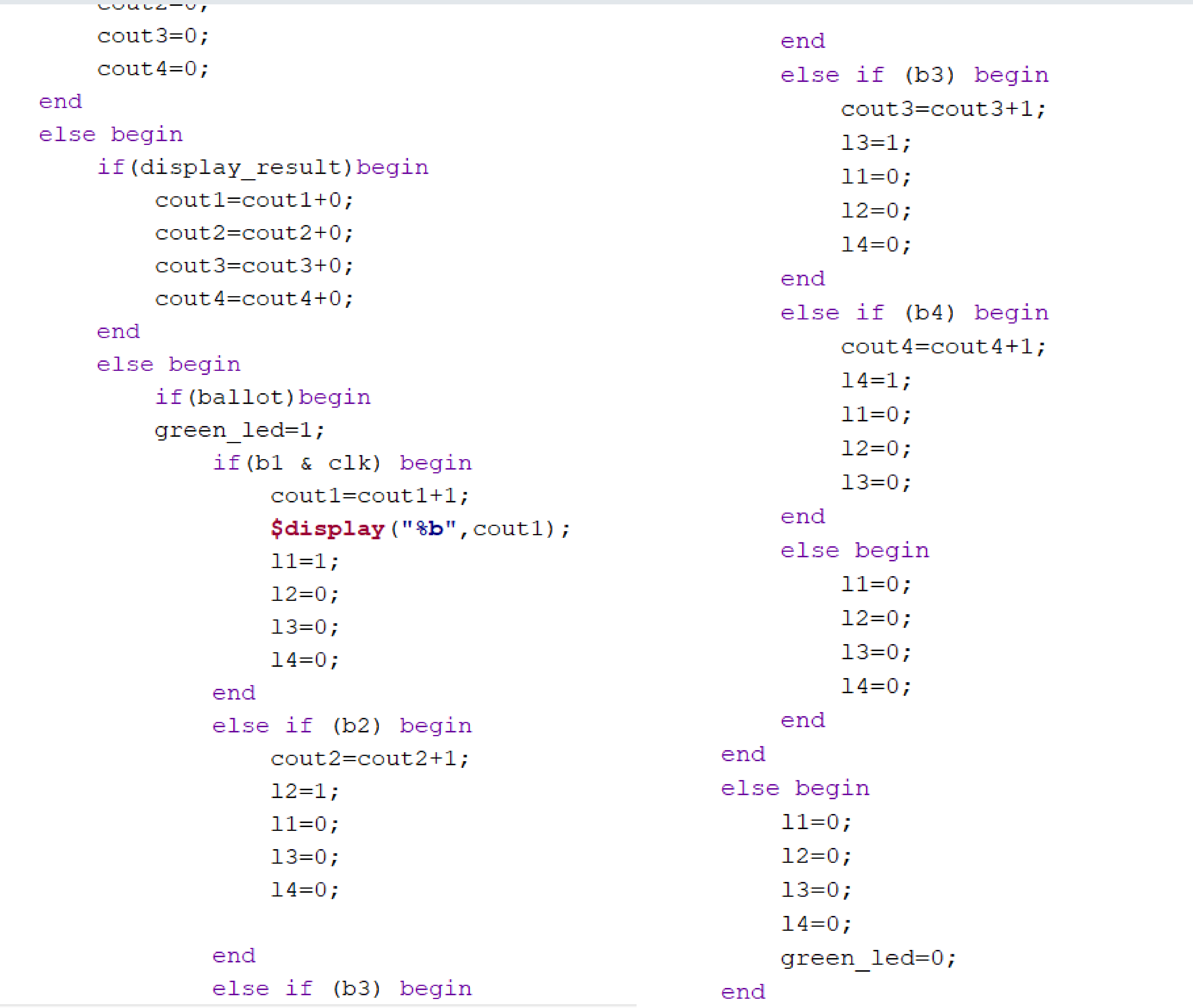

User
Reset =0
Reset =1
Set all counters to 0
Display = 0
Display = 1
Ballot =1
Ballot =0
Yes
Wait
Green Led
Pressed Bi
Yellow Led(i)
Count + 1
Pressed Selection Bi
BCD Display Count i
Clock


PYNQ-Z2
XILINX
5
Esc
Tab
CapsLk
Shift
Ctrl

Code Snippet - EVM Module

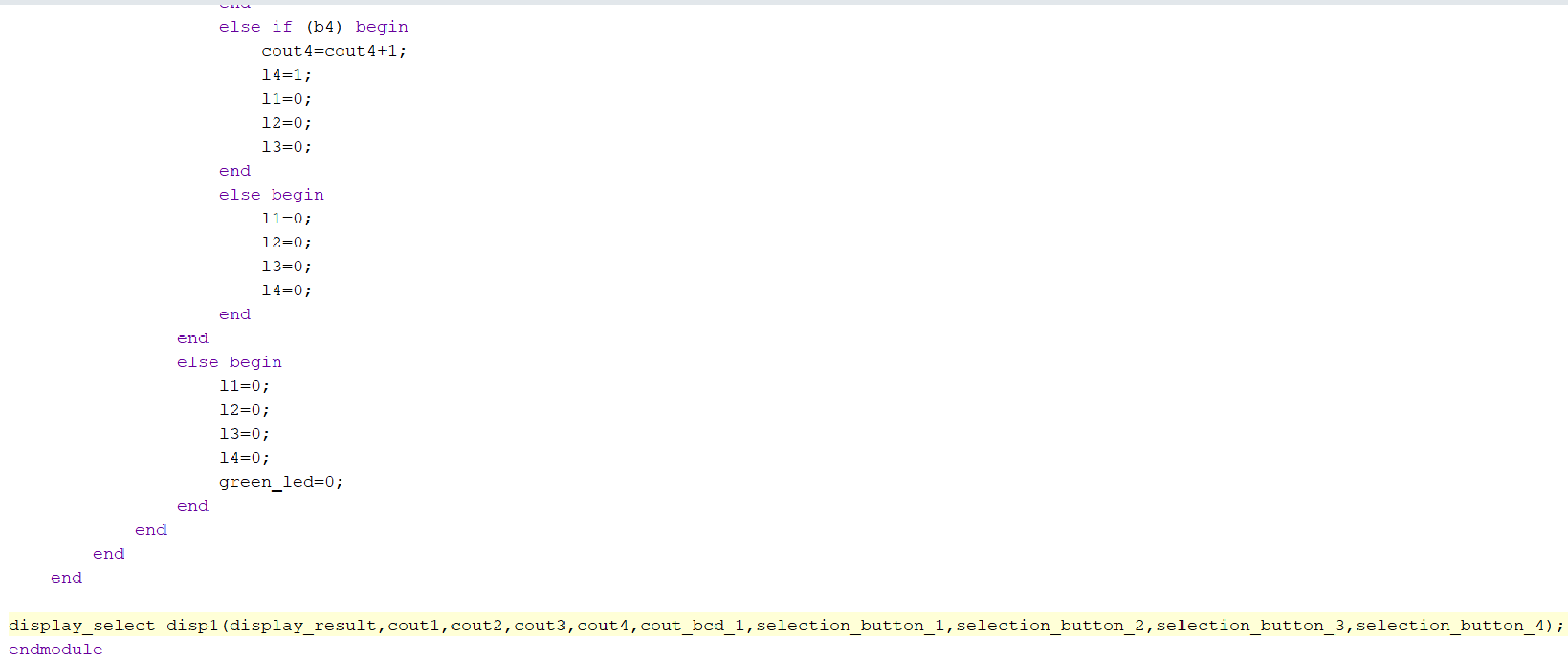


```verilog
module evm(clk,ballot,display_result,reset,b1,b2,b3,b4,selection_button_1,selection_button_2,selection_button_3,selection_button_4,green_led,l1,l2,l3,l4,cout_bcd_
input clk,ballot,display_result;
input b1,b2,b3,b4,selection_button_1,selection_button_2,selection_button_3,selection_button_4,reset;
integer i;
output reg green_led;
output reg l1;
output reg l2;
output reg l3;
output reg l4;
output co;
//output reg not_voted;
reg [3:0] cout1;
reg [3:0] cout2;
reg [3:0] cout3;
reg [3:0] cout4;
reg temp_ballot = 0;
output [6:0] cout_bcd_1;
reg [27:0] count=0;
always@(posedge clk)
  count = count + 1;
// CLK_DIV
assign co = count[26];

always @(posedge co)
    begin
        if (reset) begin
            cout1=0;
            cout2=0;
            cout3=0;
            cout4=0;
```



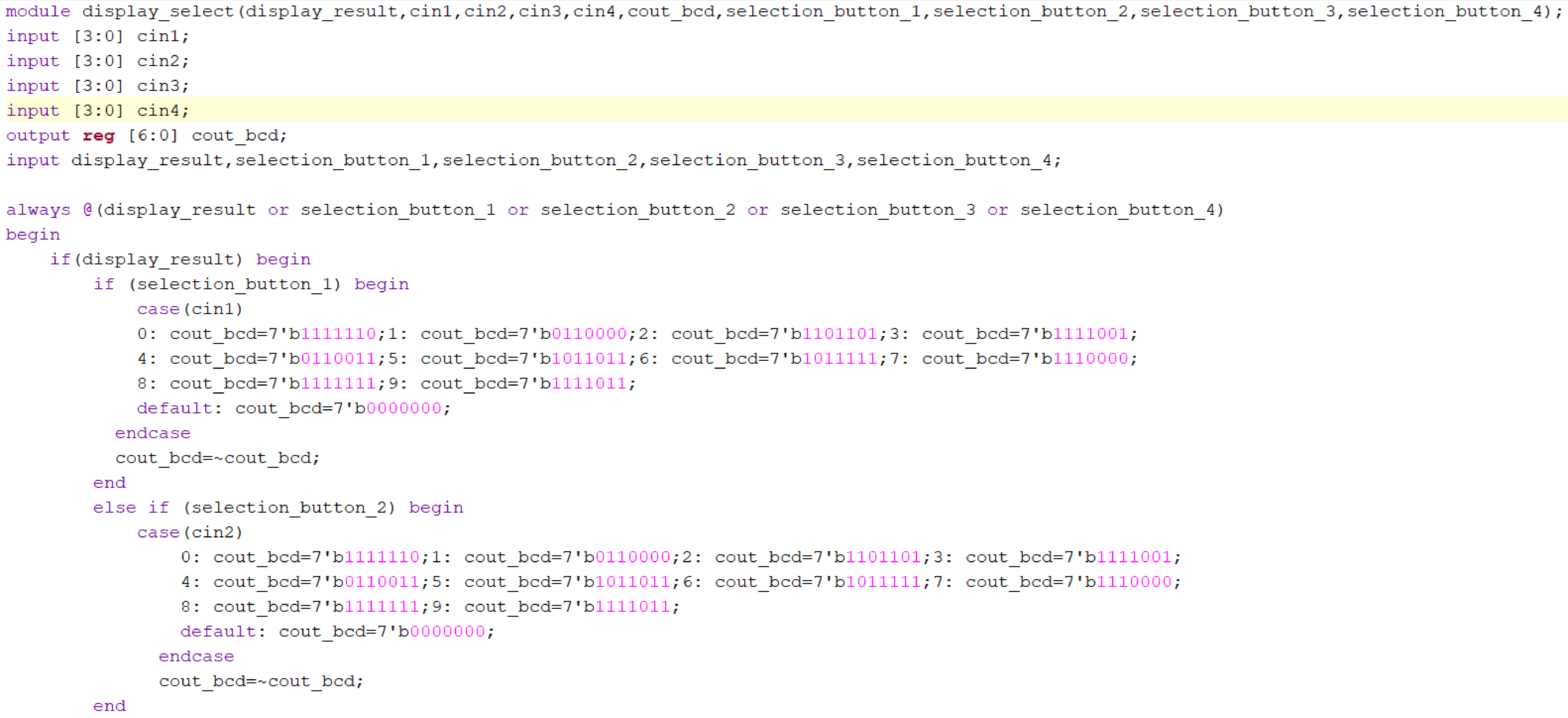
```
    cout2=0;
    cout3=0;
    cout4=0;
end
else begin
    if(display_result)begin
        cout1=cout1+0;
        cout2=cout2+0;
        cout3=cout3+0;
        cout4=cout4+0;
    end
    else begin
        if(ballot)begin
        green_led=1;
            if(b1 & clk) begin
                cout1=cout1+1;
                $display("%b",cout1);
                l1=1;
                l2=0;
                l3=0;
                l4=0;
            end
            else if (b2) begin
                cout2=cout2+1;
                l2=1;
                l1=0;
                l3=0;
                l4=0;

            end
            else if (b3) begin
```

```
            end
            else if (b3) begin
                cout3=cout3+1;
                l3=1;
                l1=0;
                l2=0;
                l4=0;
            end
            else if (b4) begin
                cout4=cout4+1;
                l4=1;
                l1=0;
                l2=0;
                l3=0;
            end
            else begin
                l1=0;
                l2=0;
                l3=0;
                l4=0;
            end
        end
        else begin
            l1=0;
            l2=0;
            l3=0;
            l4=0;
            green_led=0;
        end
```

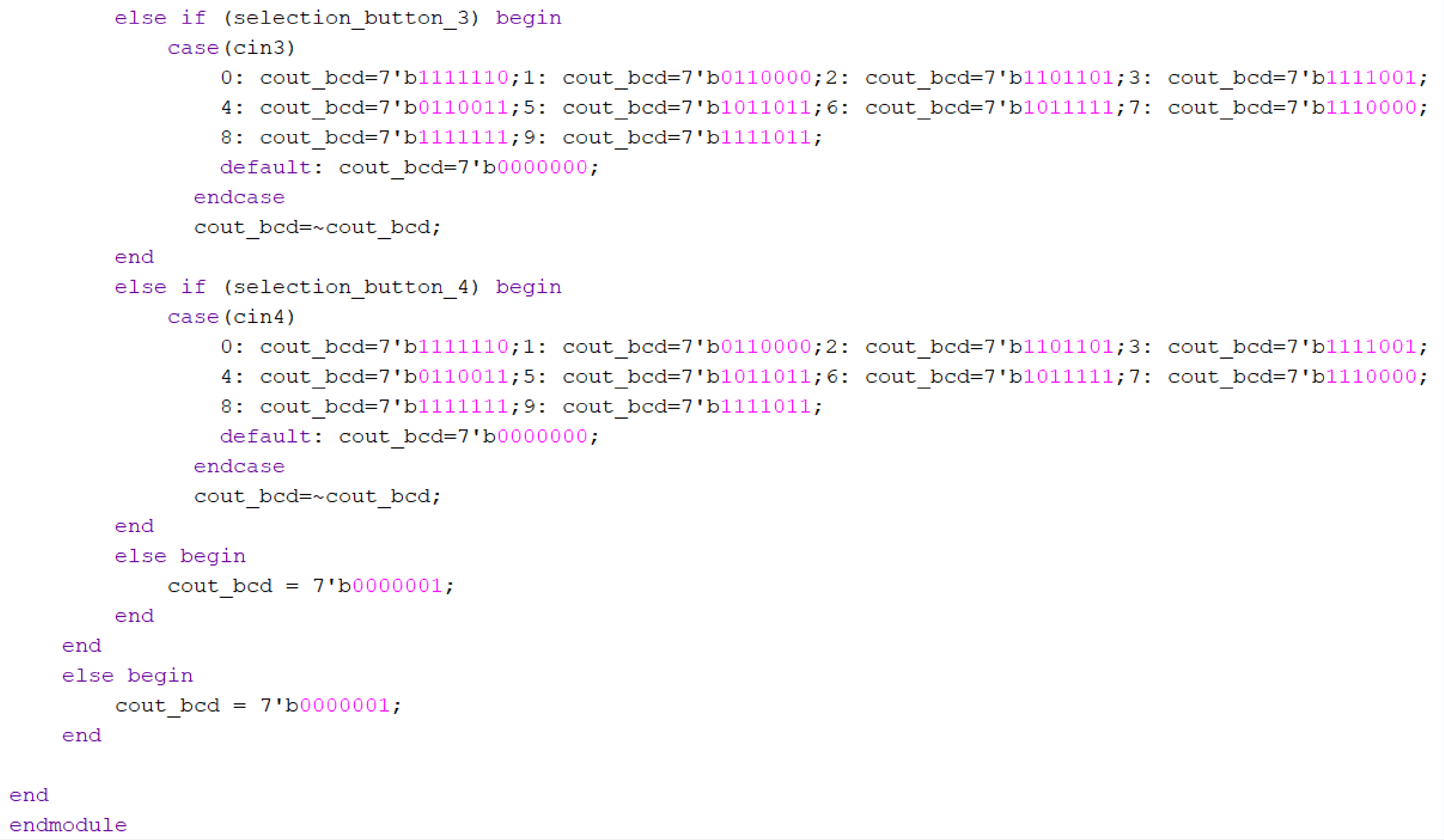


```
                    end
                    else if (b4) begin
                        cout4=cout4+1;
                        l4=1;
                        l1=0;
                        l2=0;
                        l3=0;
                    end
                    else begin
                        l1=0;
                        l2=0;
                        l3=0;
                        l4=0;
                    end
                end
                else begin
                    l1=0;
                    l2=0;
                    l3=0;
                    l4=0;
                    green_led=0;
                end
            end
        end
    end

display_select disp1(display_result,cout1,cout2,cout3,cout4,cout_bcd_1,selection_button_1,selection_button_2,selection_button_3,selection_button_4);
endmodule
```

Code Snippet - Display Select Module

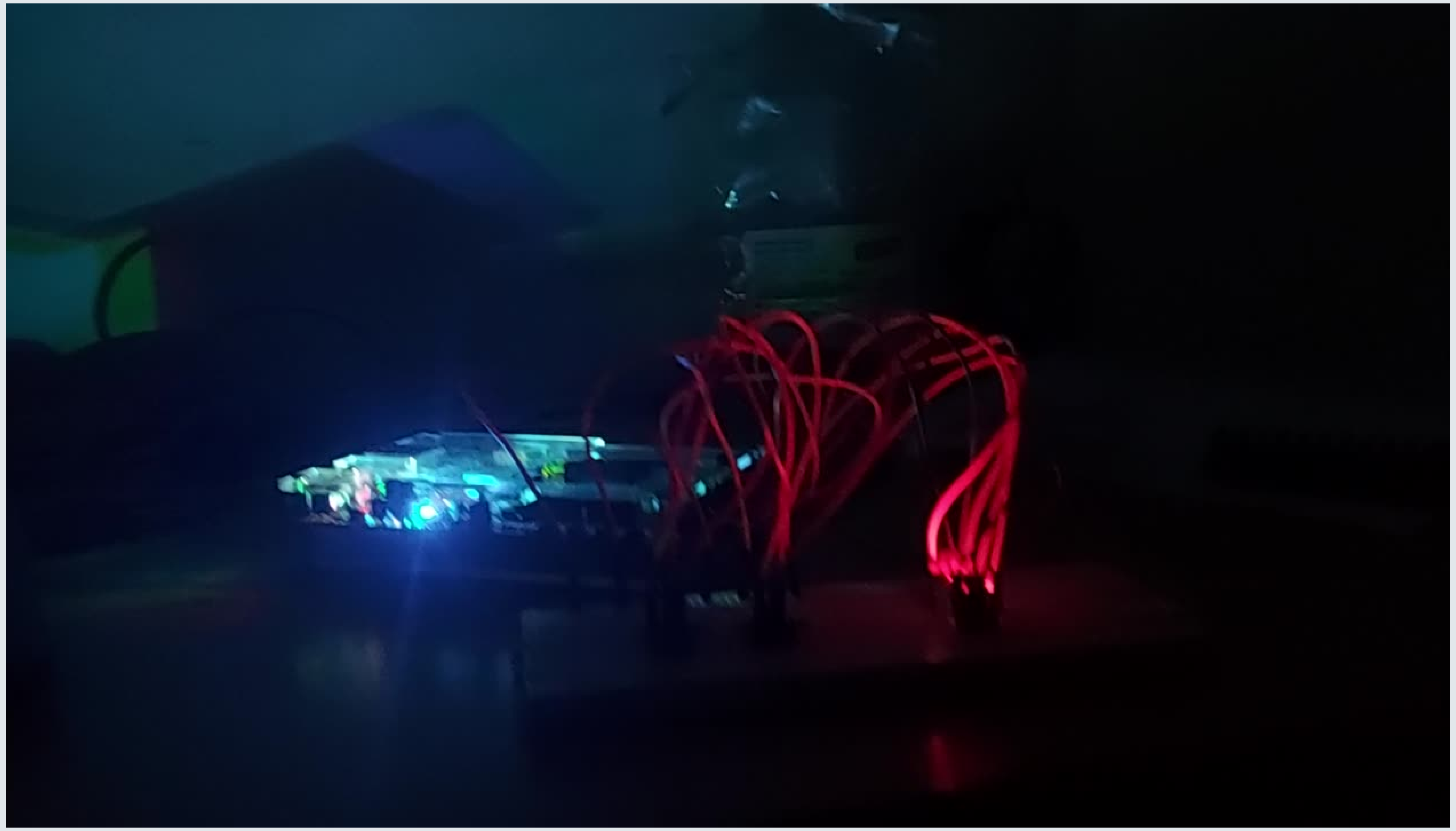


```
module display_select(display_result,cin1,cin2,cin3,cin4,cout_bcd,selection_button_1,selection_button_2,selection_button_3,selection_button_4);
input [3:0] cin1;
input [3:0] cin2;
input [3:0] cin3;
input [3:0] cin4;
output reg [6:0] cout_bcd;
input display_result,selection_button_1,selection_button_2,selection_button_3,selection_button_4;

always @(display_result or selection_button_1 or selection_button_2 or selection_button_3 or selection_button_4)
begin
    if(display_result) begin
        if (selection_button_1) begin
            case(cin1)
            0: cout_bcd=7'b1111110;1: cout_bcd=7'b0110000;2: cout_bcd=7'b1101101;3: cout_bcd=7'b1111001;
            4: cout_bcd=7'b0110011;5: cout_bcd=7'b1011011;6: cout_bcd=7'b1011111;7: cout_bcd=7'b1110000;
            8: cout_bcd=7'b1111111;9: cout_bcd=7'b1111011;
            default: cout_bcd=7'b0000000;
          endcase
          cout_bcd=~cout_bcd;
        end
        else if (selection_button_2) begin
            case(cin2)
                0: cout_bcd=7'b1111110;1: cout_bcd=7'b0110000;2: cout_bcd=7'b1101101;3: cout_bcd=7'b1111001;
                4: cout_bcd=7'b0110011;5: cout_bcd=7'b1011011;6: cout_bcd=7'b1011111;7: cout_bcd=7'b1110000;
                8: cout_bcd=7'b1111111;9: cout_bcd=7'b1111011;
                default: cout_bcd=7'b0000000;
              endcase
              cout_bcd=~cout_bcd;
        end
```



```
        else if (selection_button_3) begin
            case(cin3)
                0: cout_bcd=7'b1111110;1: cout_bcd=7'b0110000;2: cout_bcd=7'b1101101;3: cout_bcd=7'b1111001;
                4: cout_bcd=7'b0110011;5: cout_bcd=7'b1011011;6: cout_bcd=7'b1011111;7: cout_bcd=7'b1110000;
                8: cout_bcd=7'b1111111;9: cout_bcd=7'b1111011;
                default: cout_bcd=7'b0000000;
              endcase
              cout_bcd=~cout_bcd;
        end
        else if (selection_button_4) begin
            case(cin4)
                0: cout_bcd=7'b1111110;1: cout_bcd=7'b0110000;2: cout_bcd=7'b1101101;3: cout_bcd=7'b1111001;
                4: cout_bcd=7'b0110011;5: cout_bcd=7'b1011011;6: cout_bcd=7'b1011111;7: cout_bcd=7'b1110000;
                8: cout_bcd=7'b1111111;9: cout_bcd=7'b1111011;
                default: cout_bcd=7'b0000000;
              endcase
              cout_bcd=~cout_bcd;
        end
        else begin
            cout_bcd = 7'b0000001;
        end
    end
    else begin
        cout_bcd = 7'b0000001;
    end

end
endmodule
```

Behind The Scenes



Thank you!