LAB 4 Report

Name: Aryan Agarwal

UT EID: aab5473

Section: MW 10:30

Name: Sanjay Sivakumar

UT EID:

Section:

Checklist:

Part 1 -

i. Simulation waveform of the Flight Attendant Call System for behavioral as well as dataflow modelling

ii. K-map for minimizing the expression for next_state for dataflow modelling

iii. Boolean expression for next_state for dataflow modelling

iv. Completed design file (.v) for dataflow modelling

Part 2 -

v. State Diagram of the Rising Edge Detector

vi. Completed design files (.v) including the top module and clock divider

vii. Test-bench of the system

viii. Simulation waveform

ix. Constraints File (Just the uncommented portion)

Part 3 -

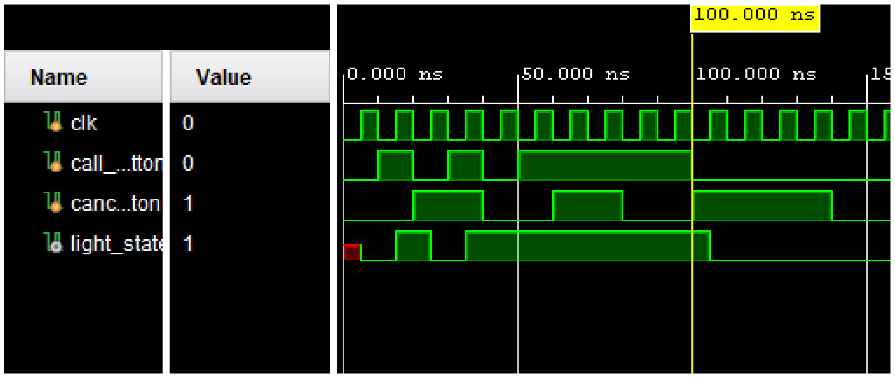
x. Completed design files (.v) of all the modules in the system

xi. Test-bench of the system

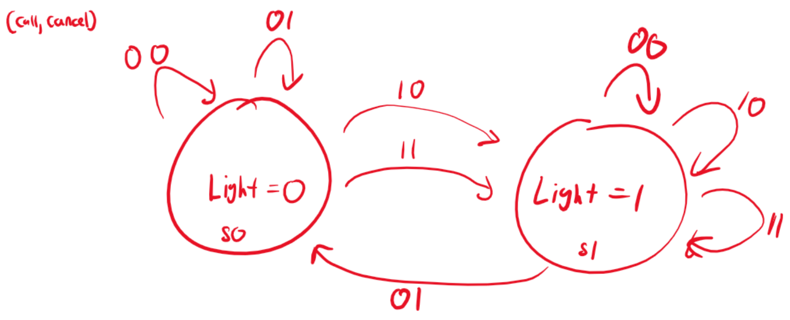
xii. Simulation waveform
xiii. Constraints File (Just the uncommented portion)

Note *→ The Verilog codes and the uncommented portions of the constraint files should be copied in your lab report and the* ***actual Verilog (.v), Constraint (.xdc) files and Bitstream (.bit) files*** *need to be zipped and submitted as well on Canvas. You are not allowed to change your codes after final submission as the TAs may download the submitted codes or bitstream files from Canvas during checkouts. For the truth Table, K-maps minimizations and algebraic expressions, you are free to draw them on paper and then put the pictures in your lab report, but please make sure it is legible for the TAs to grade it properly.*

Part 1:



```
`timescale 1ns / 1ps

module flight_attendant_call_system(

    input wire clk,

    input wire call_button,

input wire cancel_button,

    output reg light_state

    );

wire next_state ;

assign next_state = (call_button) || (!(cancel_button) && light_state);

always @( posedge clk ) begin

    light_state <= next_state;

end

endmodule
```

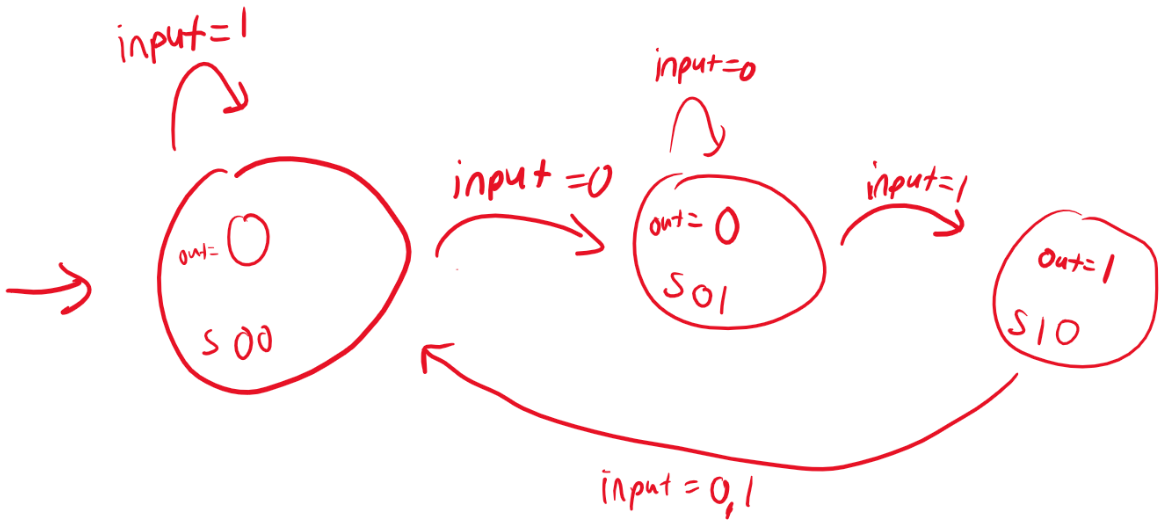


call	cancel	s	out
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

call \ cancel/s	00	01	11	10
0	0	1	0	0
1	1	1	1	1

next = call + cancel's

Part 2:



Verilog

```
`timescale 1ns / 1ps
module rising_edge_detector(
   input clk,
   input signal,
   input reset,
   output reg outedge
   );
   wire slow_clk ;
   reg [1:0] state ;
   reg [1:0] next_state ;
   clkdiv cl(clk, rest, slow_clk);
   always @(*) begin
      case (state)
         2'b00   : begin
            outedge = 1'b0;
            if (~signal)
               next_state = 2'b01;
            else
```

```
                next_state = 2'b00;
             end
          2'b01   : begin
             outedge = 1'b0;
             if (~signal)
                next_state = 2'b01;
             else
                next_state = 2'b10;
             end
          2'b10   : begin
             outedge = 1'b1;
             next_state = 2'b00;
             end
          default : begin
             next_state = 2'b00 ;
             outedge = 1'b0;
             end
       endcase
    end
    always @(posedge slow_clk) begin
       if (reset)
          state <= 1'b0;
       else
          state <= next_state;
    end
endmodule
```

<u>Clock</u>

```
`timescale 1ns / 1ps
module clkdiv(
    input clk,
    input reset,
    output clk_out
    );
    reg [25:0] COUNT;

    assign clk_out = COUNT[25]

    always @(posedge clk)

    begin

    if ( reset )

       COUNT = 0;

    else

       COUNT = COUNT + 1;

    end

endmodule
```

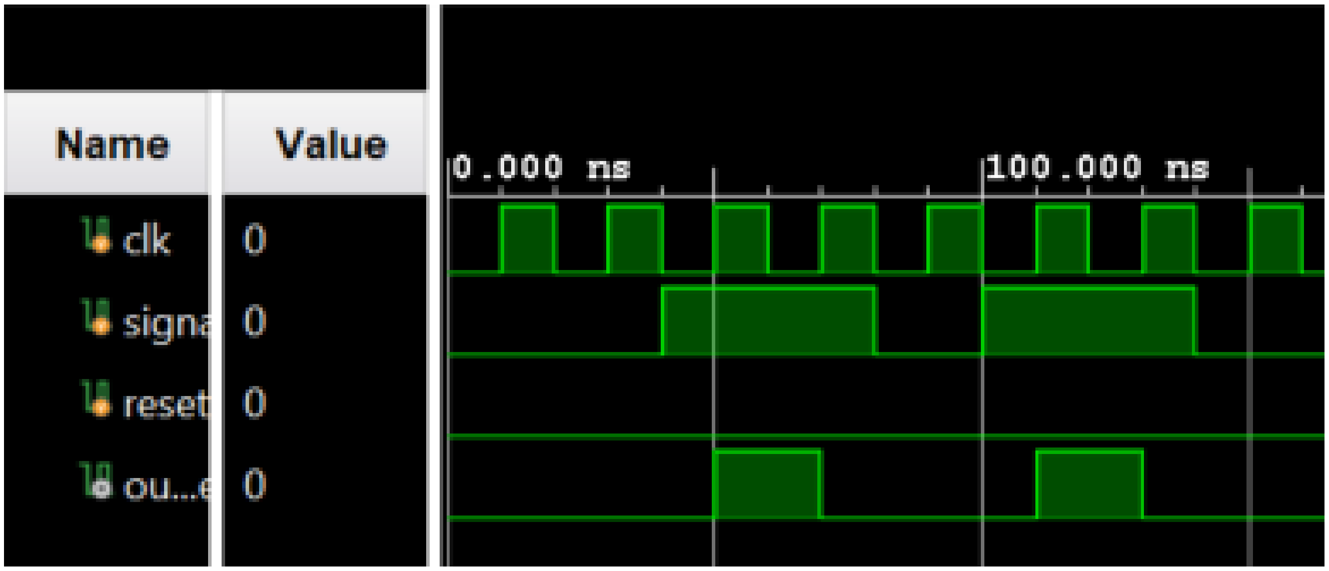
<u>Testbench</u>

```
`timescale 1ns / 1ps

module tb_rising_edge_detector;

reg clk;

reg signal;

reg reset;

wire outedge;


rising_edge_detector u1 (

    .clk(clk),

    .signal(signal),

    .reset(reset),

    .outedge(outedge)

);


initial

begin


clk = 0;

reset = 0;

signal = 0;
```

```
#20;

signal = 0;

#20;

signal = 1;

#20;

signal = 1;

#20;

signal = 0;

#20;

signal = 1;

#20;

signal = 1;

#20;

signal = 0;

end

always
#10 clk = ~clk;
```

endmodule

Simulation



Constraints

```
set_property PACKAGE_PIN W5 [get_ports {clk}]

set_property IOSTANDARD LVCMOS33 [get_ports {clk}]

create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {clk}]

set_property PACKAGE_PIN V17 [get_ports {signal}]

## Sets the switch to use 3.3V logic

        set_property IOSTANDARD LVCMOS33 [get_ports {signal}]

set_property PACKAGE_PIN U18 [get_ports reset]

        set_property IOSTANDARD LVCMOS33 [get_ports reset]

set_property PACKAGE_PIN U16 [get_ports {outedge}]

## Sets the LED to use 3.3V logic

        set_property IOSTANDARD LVCMOS33 [get_ports {outedge}]
```

Part 3:

Verilog

```
`timescale 1ns / 1ps

module time_multiplexing_main(

   input clk,

   input reset,

   input [15:0] sw,

   output [3:0] an,

   output [6:0] sseg

   );


   wire [6:0] in0, in1, in2, in3;

   wire slow_clk;


   hexto7segment c1 (.x(sw[3:0]), .r(in0));

   hexto7segment c2 (.x(sw[7:4]), .r(in1));

   hexto7segment c3 (.x(sw[11:8]), .r(in2));

   hexto7segment c4 (.x(sw[15:12]), .r(in3));


   clkdiv cl(clk, reset, slow_clk);



   time_mux_state_machine c6(

      .clk (clk),

      .reset (reset),

      .in0 (in0),

      .in1 (in1),

      .in2 (in2),

      .in3 (in3),

      .an (an),

      .sseg (sseg));

Endmodule
```

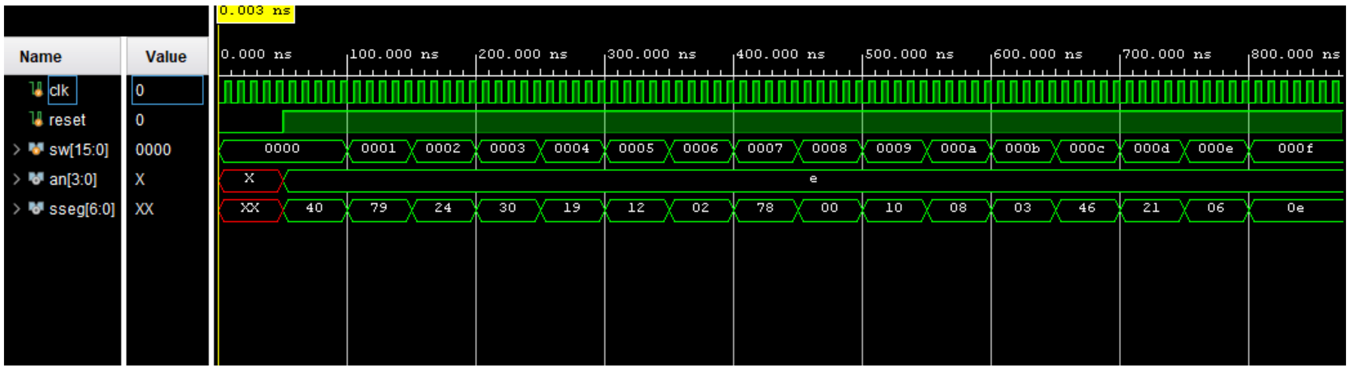
```
`timescale 1ns / 1ps

module time_mux_state_machine(

   input clk,

   input reset,

   input [6:0] in0,

   input [6:0] in1,

   input [6:0] in2,

   input [6:0] in3,

   output reg [3:0] an,

   output reg [6:0] sseg

   );


   reg [1:0] state;

   reg [1:0] next_state;


   always @ (*) begin

      case(state)

         2'b00:  next_state = 2'b01;

         2'b01:  next_state = 2'b10;

         2'b10:  next_state = 2'b11;

         2'b11:  next_state = 2'b00;

      endcase

   end


   always @(*) begin

      case (state)

         2'b00   :   sseg = in0;

         2'b01   :   sseg = in1;

         2'b10   :   sseg = in2;

         2'b11   :   sseg = in3;

      endcase


      case (state)
```

```
            2'b00:  an = 4'b1110;
            2'b01:  an = 4'b1101;
            2'b10:  an = 4'b1011;
            2'b11:  an = 4'b0111;
        endcase
     end
     always @(posedge clk or posedge reset) begin
        if(reset)
            state <= 2'b00;
        else
            state <= next_state;
     end
endmodule

`timescale 1ns / 1ps
module clkdiv(
    input clk,
    input reset,
    output slow_clk
    );
   reg [16:0] COUNT;
    assign slow_clk = COUNT[16];
    always @(posedge clk)
    begin
    if ( reset )
       COUNT = 0;
    else
       COUNT = COUNT + 1;
    end
endmodule
```

Testbench

```
`timescale 1ns / 1ps


module tb_time_multiplexing_main;
   reg clk;
   reg reset;
   reg [15:0] sw;
   wire [3:0] an;
   wire[6:0] sseg;

   time_multiplexing_main ul (
   .clk(clk),
   .reset(reset),
   .sw(sw),
   .an(an),
   .sseg(sseg)
   );

      initial
      begin
      clk = 0;
      reset = 0;
      sw = 0;

      #10;
      reset = 1;
      sw = 16'h0000;
      #10;
      sw = 16'h00A1;
      #10;
      sw = 16'h0B02;
      #10;
      sw = 16'hC003;
```

```
        #10;

        sw = 16'h00D4;

        #10;

        sw = 16'h0E05;

    end


    always #5 clk = ~clk;

endmodule
```



Constraints

```
set_property PACKAGE_PIN W5 [get_ports clk]

                set_property IOSTANDARD LVCMOS33 [get_ports clk]

                create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]


set_property PACKAGE_PIN V17 [get_ports {sw[0]}]

## Sets the switch to use 3.3V logic

                set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]

## Connects pin V16 (SW1 on the board) to input b in our gate module

set_property PACKAGE_PIN V16 [get_ports {sw[1]}]

## Sets the switch to use 3.3V logic

                set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]

set_property PACKAGE_PIN W16 [get_ports {sw[2]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]

set_property PACKAGE_PIN W17 [get_ports {sw[3]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]

set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
```

```
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]

#7 segment display
set_property PACKAGE_PIN W7 [get_ports {sseg[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[0]}]
set_property PACKAGE_PIN W6 [get_ports {sseg[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[1]}]
set_property PACKAGE_PIN U8 [get_ports {sseg[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[2]}]
set_property PACKAGE_PIN V8 [get_ports {sseg[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[3]}]
set_property PACKAGE_PIN U5 [get_ports {sseg[4]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {sseg[4]}]

set_property PACKAGE_PIN V5 [get_ports {sseg[5]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {sseg[5]}]

set_property PACKAGE_PIN U7 [get_ports {sseg[6]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {sseg[6]}]


set_property PACKAGE_PIN U2 [get_ports {an[0]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]

set_property PACKAGE_PIN U4 [get_ports {an[1]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]

set_property PACKAGE_PIN V4 [get_ports {an[2]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]

set_property PACKAGE_PIN W4 [get_ports {an[3]}]

                set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]


#Buttons

set_property PACKAGE_PIN U18 [get_ports reset]

                set_property IOSTANDARD LVCMOS33 [get_ports reset]
```