Lab 8

Top module

```
module top(
    input clk,
    input sw,
    input [1:0] press,      // input sequence
    input rst,
    output led,
    output [7:0] cathode,
    output [3:0] anode
    );

    wire clk_190,push_clk,pb_press;
    wire [2:0] ps1,ps2;
    wire led1,led2;

    reg led=0;
    reg [2:0] ps=3'b0;

    assign pb_press= press[0] | press[1];

    frq_div #(20) fun1(clk,clk_190);
    pulse fun2(clk_190,push_clk,pb_press);
    overlap_10101 fun3(push_clk,press[1],rst,led1,ps1);
    non_overlap_10101 fun4(push_clk,press[1],rst,led2,ps2);

    always @ (posedge clk)
```

```
    begin
        if(sw==0)     // For overlapping   // 101 10101
        begin
            led<=led1;
            ps<=ps1;
        end
        else if(sw==1)    // For non-overlapping
        begin
            led<=led2;
            ps<=ps2;
        end
    end

    sevenseg
fun5(.clk(clk),.clr(rst),.ones(ps),.tens(4'd5),.cathode(cathode),.ano
de(anode));

Endmodule
```

Frequency Divider

```
module frq_div(
    input clk,
    output div_clk
    );

parameter n=25;
```

```
reg [n-1:0] q=0;

always @ (posedge clk)
begin

q<=q+1;

end

assign div_clk=q[n-1];

endmodule
```

Pulse

```
module pulse(
    input clk,
    output deb_out,
    input deb_in
    );

    reg D1,D2,D3;
    always@(posedge clk)
    begin
        D1<=deb_in;
        D2<=D1;
```

```
        D3<=~D2; // inversion is done to generate a pulse
    end

    assign deb_out=D1&&D2&&D3;

endmodule
```

Overlapping

```
module overlap_10101(
    input fsm_clk,
    input din,
    input rst,
    output reg seq_det=0,
    output reg [2:0] ps=0   // present state
    );

    reg [2:0] ns;   // next state
    parameter [2:0]
s0=3'd0,s1=3'd1,s2=3'd2,s3=3'd3,s4=3'd4,s5=3'd5,s6=3'd6,s7=3'
d7;

    always @ (posedge fsm_clk or posedge rst)  // this is for
present state
    begin
        if(rst==1)
            ps<=s0;
```

```
    else
        ps<=ns;
end

always @ (*)    // for next state
begin
    case(ps)
        s0: if(din==1)
                ns=s1;
            else
                ns=s0;
        s1: if(din==0)
                ns=s2;
            else
                ns=s1;
        s2: if(din==0)
                ns=s0;
            else
                ns=s3;
        s3: if(din==0)
                ns=s4;
            else
                ns=s1;
        s4: if(din==1)
                ns=s3;
            else
                ns=s0;
     endcase
end
```

```
always @ (posedge fsm_clk or posedge rst)  // this is for output
begin
    if(rst==1)
        seq_det<=0;
    else if(ps==s4 && din==1)
        seq_det<=1;
    else
        seq_det<=0;
end

endmodule
```

Non-Overlapping

```
module non_overlap_10101(
    input fsm_clk,
    input din,
    input rst,
    output reg seq_det=0,
    output reg [2:0] ps=0   // present state
    );

    reg [2:0] ns;   // next state
    parameter [2:0]
s0=3'd0,s1=3'd1,s2=3'd2,s3=3'd3,s4=3'd4,s5=3'd5,s6=3'd6,s7=3'
d7;
```

```
    always @ (posedge fsm_clk or posedge rst)  // this is for
present state
    begin
        if(rst==1)
            ps<=s0;
        else
            ps<=ns;
    end

    always @ (*)    // for next state
    begin
        case(ps)
            s0: if(din==1)
                    ns=s1;
                else
                    ns=s0;
            s1: if(din==0)
                    ns=s2;
                else
                    ns=s1;
            s2: if(din==0)
                    ns=s0;
                else
                    ns=s3;
            s3: if(din==0)
                    ns=s4;
                else
                    ns=s0;
```

```
            s4: if(din==1)
                    ns=s5;
                else
                    ns=s0;
            s5: ns=s0;
          endcase
    end

always @ (posedge fsm_clk or posedge rst)  // this is for output
begin
    if(rst==1)
        seq_det<=0;
    else if(ps==s4 && din==1)
        seq_det<=1;
    else
        seq_det<=0;
end

endmodule
```

XDC File

```
# Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]

        set_property IOSTANDARD LVCMOS33 [get_ports clk]

## Switches
```

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

        set_property IOSTANDARD LVCMOS33 [get_ports {sw}]

## LEDs
set_property PACKAGE_PIN U16 [get_ports {led}]

        set_property IOSTANDARD LVCMOS33 [get_ports {led}]

#7 segment display
            set_property PACKAGE_PIN W7 [get_ports {cathode[7]}]
                set_property IOSTANDARD LVCMOS33 [get_ports
{cathode[7]}]
            set_property PACKAGE_PIN W6 [get_ports {cathode[6]}]
                set_property IOSTANDARD LVCMOS33 [get_ports
{cathode[6]}]
            set_property PACKAGE_PIN U8 [get_ports {cathode[5]}]
                set_property IOSTANDARD LVCMOS33 [get_ports
{cathode[5]}]
            set_property PACKAGE_PIN V8 [get_ports {cathode[4]}]
                set_property IOSTANDARD LVCMOS33 [get_ports
{cathode[4]}]
            set_property PACKAGE_PIN U5 [get_ports {cathode[3]}]
                set_property IOSTANDARD LVCMOS33 [get_ports
{cathode[3]}]
            set_property PACKAGE_PIN V5 [get_ports {cathode[2]}]
                set_property IOSTANDARD LVCMOS33 [get_ports
{cathode[2]}]
            set_property PACKAGE_PIN U7 [get_ports {cathode[1]}]
```

```
            set_property IOSTANDARD LVCMOS33 [get_ports
{cathode[1]}]
        set_property PACKAGE_PIN V7 [get_ports cathode[0]]
            set_property IOSTANDARD LVCMOS33 [get_ports
cathode[0]]

        set_property PACKAGE_PIN U2 [get_ports {anode[0]}]
            set_property IOSTANDARD LVCMOS33 [get_ports
{anode[0]}]
        set_property PACKAGE_PIN U4 [get_ports {anode[1]}]
            set_property IOSTANDARD LVCMOS33 [get_ports
{anode[1]}]
        set_property PACKAGE_PIN V4 [get_ports {anode[2]}]
            set_property IOSTANDARD LVCMOS33 [get_ports
{anode[2]}]
        set_property PACKAGE_PIN W4 [get_ports {anode[3]}]
            set_property IOSTANDARD LVCMOS33 [get_ports
{anode[3]}]

##Buttons
set_property PACKAGE_PIN U18 [get_ports press[0]]

      set_property IOSTANDARD LVCMOS33 [get_ports press[0]]
set_property PACKAGE_PIN T18 [get_ports press[1]]

      set_property IOSTANDARD LVCMOS33 [get_ports press[1]]
set_property PACKAGE_PIN W19 [get_ports rst]

      set_property IOSTANDARD LVCMOS33 [get_ports rst]
```

Output waveform