Lab 8

Submitted By :
Snehal Gupta
2016201

Top module :

```
module top(
   input clk,
   input [3:0] inp,
   input clr,
   input btn1,
   output [6:0] cathode,
   output anode,
   output [3:0] col
   );
   wire a1,a2;
   wire [3:0] ones,tens,num;
   wire [7:0] dout_dsp2,dout_dsp3;
   wire [6:0] dout_dsp1;
   reg [4:0] din_dsp2,din_dsp3;
   wire [1:0] weight;
   assign weight = 2'b01;
   always @( posedge out_pulse or posedge clr)
      begin
         if (clr == 1)begin
            din_dsp2 <= 5'b0000;
            din_dsp3 <= 5'b0000;
         end
         else begin
            din_dsp2 <= {1'b0,num};
            din_dsp3 <= din_dsp2;
         end
      end
   Decoder d2(.clk(clk),.Row(inp),.Col(col),.DecodeOut(num));
   xbip_dsp48_macro_0 dsp1(.A({1'b0,num}),.B(weight),.P(dout_dsp1));
   xbip_dsp48_macro_1 dsp2(.A(din_dsp2),.B(weight),.C(dout_dsp1),.P(dout_dsp2));
   xbip_dsp48_macro_2 dsp3(.A(din_dsp3),.B(weight),.C(dout_dsp2),.P(dout_dsp3));
   clk_div CD1(.mclk(clk),.clk190(clk_190));
   debounce D1(.clk_in(clk_190),.clr_in(clr),.clr_out(clr_de));
   debounce D2(.clk_in(clk_190),.clr_in(btn1),.clr_out(btn));
   clk_pulse cp1(.inp(btn),.cclk(clk_190),.clr(clr_de),.outp(out_pulse));
   binary2bcd(.number(dout_dsp3),.ones(ones),.tens(tens));
   pmodco p1(.ones(ones),.tens(tens),.cathode(cathode),.anode(anode),.clock_in(clk));
```

```
endmodule
Decoder :
module Decoder(
    clk,
    Row,
    Col,
    DecodeOut
    );


    input clk;
    input [3:0] Row;
    output [3:0] Col;
    output [3:0] DecodeOut;

        reg [3:0] Col=0;
        reg [3:0] DecodeOut=0;


        reg [19:0] sclk=0;



        always @(posedge clk) begin


                    if (sclk == 20'b00011000000000000000) begin

                        Col <= 4'b0111;
                        sclk <= sclk + 1'b1;
                    end


                    else if(sclk == 20'b00011000000000001000) begin

                        if (Row == 4'b0111) begin
                            DecodeOut <= 4'b0001;
                        end

                        else if(Row == 4'b1011) begin
                            DecodeOut <= 4'b0100;
                        end
```

```
        else if(Row == 4'b1101) begin
                DecodeOut <= 4'b0111;
        end

        else if(Row == 4'b1110) begin
                DecodeOut <= 4'b0000;
        end
        sclk <= sclk + 1'b1;
end


else if(sclk == 20'b00110000000000000000) begin

        Col<= 4'b1011;
        sclk <= sclk + 1'b1;
end


else if(sclk == 20'b00110000000000001000) begin

        if (Row == 4'b0111) begin
                DecodeOut <= 4'b0010;
        end

        else if(Row == 4'b1011) begin
                DecodeOut <= 4'b0101;
        end

        else if(Row == 4'b1101) begin
                DecodeOut <= 4'b1000;
        end

        else if(Row == 4'b1110) begin
                DecodeOut <= 4'b1111;
        end
        sclk <= sclk + 1'b1;
end


else if(sclk == 20'b01100000000000000000) begin

        Col<= 4'b1101;
        sclk <= sclk + 1'b1;
```

```
end


else if(sclk == 20'b01100000000000001000) begin

        if(Row == 4'b0111) begin
                DecodeOut <= 4'b0011;
        end

        else if(Row == 4'b1011) begin
                DecodeOut <= 4'b0110;
        end

        else if(Row == 4'b1101) begin
                DecodeOut <= 4'b1001;
        end

        else if(Row == 4'b1110) begin
                DecodeOut <= 4'b1110;
        end

        sclk <= sclk + 1'b1;
end

else if(sclk == 20'b11000000000000000000) begin

        Col<= 4'b1110;
        sclk <= sclk + 1'b1;
end

else if(sclk == 20'b11000000000000001000) begin

        if(Row == 4'b0111) begin
                DecodeOut <= 4'b1010;
        end

        else if(Row == 4'b1011) begin
                DecodeOut <= 4'b1011;
        end

        else if(Row == 4'b1101) begin
                DecodeOut <= 4'b1100;
        end
```

```
                        else if(Row == 4'b1110) begin
                                DecodeOut <= 4'b1101;
                        end
                        sclk <= 20'b00000000000000000000;
                end


                else begin
                        sclk <= sclk + 1'b1;
                end

        end

endmodule

//pmodco.v
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 26.10.2017 01:05:57
// Design Name:
// Module Name: pmodco
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module pmodco(
    input clock_in,
    input [3:0] ones,
    input [3:0] tens,
```

```
output [6:0] cathode,
output anode
);
reg [6:0] sseg_temp;
reg [6:0]sseg;
reg an_temp=1'b0;
reg [17:0] count;
    always@(posedge clock_in)
    begin
    count <= count+1;
    end
    always @ (*)
        begin
         case(count[17])
          1'b0 :
           begin
            sseg = ones;
            an_temp = 1'b1;
           end
          1'b1:
           begin
            sseg = tens;
            an_temp = 1'b0;
           end
         endcase
        end
assign anode=an_temp;
always @ (*)
     begin
      case(sseg)
       4'd0 : sseg_temp = 7'b0000001;
       4'd1 : sseg_temp = 7'b1001111;
       4'd2 : sseg_temp = 7'b0010010;
       4'd3 : sseg_temp = 7'b0000110;
       4'd4 : sseg_temp = 7'b1001100;
       4'd5 : sseg_temp = 7'b0100100;
       4'd6 : sseg_temp = 7'b0100000;
       4'd7 : sseg_temp = 7'b0001111;
       4'd8 : sseg_temp = 7'b0000000;
       4'd9 : sseg_temp = 7'b0000100;
       default : sseg_temp = 7'b1111110;
      endcase
     end
```

```
        assign cathode = ~sseg_temp;
Endmodule
```

//clk_div.v

```
`timescale 1ns / 1ps
module clk_div(
input wire mclk,
input wire clr,
output wire clk190);
reg [18:0] q;
always @(posedge mclk)
begin
if(clr == 1)
q <= 0;
else
q <= q+1;
end
assign clk190 = q[18];
endmodule
```

```
//binary2bcd.v
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 26.09.2017 17:01:34
// Design Name:
// Module Name: bcd_adder
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
```

```
module binary2bcd(
    number,hundreds,tens,ones
    );
    input[7:0] number;
    output reg [3:0]hundreds;
    output reg [3:0]tens;
    output reg [3:0]ones;
    // double dabble algorithm
    reg[19:0] shift;
    integer  i;
    always @(number)
    begin
        shift[19:8] =0;
        shift[7:0] = number;
        for (i=0;i<8;i=i+1)
        begin
            if(shift[11:8] >= 5)
                shift[11:8] = shift[11:8] + 3;
            if(shift[15:12] >= 5)
                shift[15:12] = shift[15:12] + 3;
            if(shift[19:16] >= 5)
                shift[19:16] = shift[19:16] + 3;
            shift = shift <<1;
         end

         hundreds = shift[19:16];
         tens     = shift[15:12];
         ones     = shift[11:8];
    end



endmodule
```

Clk_pulse.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 25.10.2017 21:11:14
// Design Name:
```

```
// Module Name: clk_pulse
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module clk_pulse(
input wire inp,
input wire cclk,
input wire clr,
output wire outp);
reg delay1;
reg delay2;
reg delay3;
always@(posedge cclk or posedge clr)
begin
if(clr == 1)
begin
delay1 <= 1'b0;
delay2 <= 1'b0;
delay3 <= 1'b0;
end
else
begin
delay1 <= inp;
delay2 <= delay1;
delay3 <= delay2;
end
end
assign outp = delay1 & delay2 & delay3;
endmodule

//debounce.v
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
```

```
// Company:
// Engineer:
//
// Create Date: 04.10.2017 16:46:57
// Design Name:
// Module Name: debounce
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module debounce(
    input clk_in,
    input clr_in,
    output clr_out
    );
    reg D1,D2,D3;
    always@(posedge clk_in)
    begin
    D1 <= clr_in;
    D2 <= D1;
    D3 <= D2;
    end
    assign clr_out = D1 && D2 && D3;
endmodule

//const.xdc
set_property PACKAGE_PIN W5 [get_ports {clk}]
set_property IOSTANDARD LVCMOS33 [get_ports {clk}]
set_property PACKAGE_PIN T18 [get_ports {clr}]
set_property IOSTANDARD LVCMOS33 [get_ports {clr}]
set_property PACKAGE_PIN T17 [get_ports {btn1}]
set_property IOSTANDARD LVCMOS33 [get_ports {btn1}]
set_property PACKAGE_PIN G3 [get_ports {inp[3]}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {inp[3]}]
set_property PACKAGE_PIN H2 [get_ports {inp[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {inp[2]}]
set_property PACKAGE_PIN K2 [get_ports {inp[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {inp[1]}]
set_property PACKAGE_PIN H1 [get_ports {inp[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {inp[0]}]
set_property PACKAGE_PIN G2 [get_ports {col[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {col[3]}]
set_property PACKAGE_PIN J2 [get_ports {col[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {col[2]}]
set_property PACKAGE_PIN L2 [get_ports {col[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {col[1]}]
set_property PACKAGE_PIN J1 [get_ports {col[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {col[0]}]

set_property PACKAGE_PIN A14 [get_ports {cathode[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {cathode[3]}]
set_property PACKAGE_PIN A16 [get_ports {cathode[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {cathode[2]}]
set_property PACKAGE_PIN B15 [get_ports {cathode[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {cathode[1]}]
set_property PACKAGE_PIN B16 [get_ports {cathode[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {cathode[6]}]
set_property PACKAGE_PIN K17 [get_ports {cathode[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {cathode[5]}]
set_property PACKAGE_PIN M18 [get_ports {cathode[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {cathode[4]}]
set_property PACKAGE_PIN N17 [get_ports {cathode[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {cathode[0]}]
set_property PACKAGE_PIN P18 [get_ports {anode}]
set_property IOSTANDARD LVCMOS33 [get_ports {anode}]
```