LAB 11-12
SARTHAK BHAGAT

TOP

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 11/13/2017 12:43:13 AM
// Design Name:
// Module Name: gcd_top
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module gcd_top(
    input [5:0] xin,
    input [5:0] yin,
    input clk,
    input clr,
    input go,
    input value,
    output reg [7:0] cathode1,
    output reg [3:0] anode1,
    output reg [6:0] cathode2,
    output reg anode2
    );

    wire [3:0] anodea,anodeb,anodec,anoded;
wire [7:0] cathodea,cathodeb,cathodec,cathoded;
wire temp_anode2;
wire [6:0] cathode2a, cathode2b;
wire [3:0] ones1,ones2,ones3,tens1,tens2,tens3,hundreds1,hundreds2,hundreds3;
wire [5:0] d1;
wire [5:0] d2;

blk_mem_gen_0 bmg1 (.clka(clk),.ena(1'b1),.addra(xin),.douta(d1));
blk_mem_gen_0 bmg2 (.clka(clk),.ena(1'b1),.addra(yin),.douta(d2));

wire [5:0] xout, yout;

always @(posedge clk)
begin
   if (value == 1'b1)
   begin
      xout = xin;
      yout = yin;
   end
   else
   begin
      xout = d1;
      yout = d2;
   end
end

initial
begin
   cathode1 = cathodea;
end
initial
begin
   cathode2 = cathode2a;
end

debounce d1 (clk,clr,clrout);

fd #(18) f1 (clk,clk19);
fd #(2) f2 (clk,clk25);

clean_pulse cp1 (clk25,go,goout);

assign anodea = 4'b1110;
assign anodeb = 4'b1101;
assign anodec = 4'b1011;
assign anoded = 4'b0111;

wire [7:0] xs,ys;
assign xs = {2'b00,xout};
assign ys = {2'b00,yout};

b2b b1 (xs, ones1, tens1, hundreds1);
b2b b2 (ys, ones2, tens2, hundreds2);
seven_segment s1 (ones1, cathodea);
seven_segment s2 (tens1, cathodeb);
seven_segment s3 (ones2, cathodec);
seven_segment s4 (tens2, cathoded);

wire [5:0] gcd;
gcd_fsm g1 (clk25,goout,clrout,eqflg,ltflg,xmsel,ymsel,xld,yld,gld);
datapath d12 (clk25,clrout,xout,yout,xmsel,ymsel,xld,yld,gld,eqflg,ltflg,gcd);

wire [7:0] gcds;
assign gcds = {2'b00, gcd};
b2b b3 (gcds, ones3, tens3, hundreds3);

pmod p1 (ones3, cathode2a);
pmod p2 (tens3, cathode2b);

always @(posedge clk19)
begin
   if (anode1 == anodea)
   begin
      anode1 = 4'b1101;
      cathode1 = cathodeb;
   end
   else if (anode1 == anodeb)
   begin
      anode1 = 4'b1011;
      cathode1 = cathodec;
   end
   else if (anode1 == anodec)
   begin
      anode1 = 4'b0111;
      cathode1 = cathoded;
   end
```

```
        else
        begin
            anode1 = 4'b1110;
            cathode1 = cathodea;
        end
    end

    always @(posedge clk19)
    begin
    if (anode2 == 1'b0)
        begin
            anode2 = 1'b1;
            cathode2 = cathode2a;
        end
    else
        begin
            anode2 = 1'b0;
            cathode2 = cathode2b;
        end
    end

endmodule
```

DATA PATH

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 11/13/2017 06:36:35 PM
// Design Name:
// Module Name: datapath
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////


module datapath(
    input clk,
    input clr,
    input [5:0] xin,
    input [5:0] yin,
    input xmsel,
    input ymsel,
    input xld,
    input yld,
    input gld,
    output reg eqflg,
    output reg ltflg,
    output reg [5:0] gcd
    );

    reg [5:0] x;
    reg [5:0] y;

    assign xmy = x - y;
    assign ymx = y - x;

    always @(posedge clk)
    begin
        if (clr == 1'b1)
        begin
            x = 1'b0;
            y = 1'b0;
            gcd = 1'b0;
        end
        if (xld == 1'b1)
        begin
            if (xmsel == 1'b1)
            begin
                x = xin;
            end
            else
            begin
                x = xmy;
            end
        end
        if (yld == 1'b1)
        begin
            if (ymsel == 1'b1)
            begin
                y = yin;
            end
            else
            begin
                y = ymx;
            end
        end
        if (gld == 1'b1)
        begin
            gcd = x;
        end

    end

    always @(*)
    begin
        if (x == y)
            eqflg = 1'b1;
        else
            eqflg = 1'b0;
    end

    always @(*)
    begin
        if (x < y)
            ltflg = 1'b1;
        else
            ltflg = 1'b0;
    end


endmodule
```

CONTROL PATH

```
`timescale 1ns / 1ps
```

```
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 11/13/2017 02:27:03 PM
// Design Name:
// Module Name: gcd_fsm
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module gcd_fsm(
    input clk,
    input go,
    input clr,
    input eqflg,
    input ltflg,
    output reg xmsel,
    output reg ymsel,
    output reg xld,
    output reg yld,
    output reg gld
    );

    reg [2:0] start = 3'b000;
    reg [2:0] input1 = 3'b001;
    reg [2:0] test1 = 3'b010;
    reg [2:0] test2 = 3'b011;
    reg [2:0] update1 = 3'b100;
    reg [2:0] update2 = 3'b101;
    reg [2:0] done = 3'b110;

    reg [2:0] present = 3'b000;
    reg [2:0] next = 3'b000;

    initial
    begin
    present = start;
    end

    always @(posedge clk)
    begin
        if (present == start)
        begin
            gld <= 1'b0;
            xld <= 1'b0;
            yld <= 1'b0;
        end
        else if (present == input1)
        begin
            xmsel <= 1'b1;
            ymsel <= 1'b1;
            xld <= 1'b1;
            yld <= 1'b1;
            gld <= 1'b0;
        end
        else if (present == test1)
        begin
            xld <= 1'b0;
            yld <= 1'b0;
            gld <= 1'b0;
        end
        else if (present == test2)
        begin
            xld <= 1'b0;
            yld <= 1'b0;
            gld <= 1'b0;
        end
        else if (present == update1)
        begin
            yld <= 1'b1;
            ymsel <= 1'b0;
            gld <= 1'b0;
        end
        else if (present == update2)
        begin
```

```
        xld <= 1'b1;
        xmsel <= 1'b0;
        gld <= 1'b0;
    end
    else
    begin
        gld <= 1'b1;
        xld <= 1'b0;
        yld <= 1'b0;
    end
end

always @(posedge clk)
begin
    if (present == start)
    begin
    if (go == 1)
        next = input1;
    else
        next = start;
    end
    else if (present == input1)
        next = test1;
    else if (present == test1)
    begin
        if (eqflg == 1)
            next = done;
        else
            next = test2;
    end
    else if (present == test2)
    begin
        if (ltflg == 1)
            next = update1;
        else
            next = update2;
    end
    else if (present == update1)
        next = test1;
    else if (present == update2)
        next = test1;
    else
    begin
```

```
            if (clr == 1)
                next = start;
            else
                next = done;
        end
    end

    always @(*)
    begin
        present = next;
    end

endmodule
```