

module lab2\_2(clk,rst\_n,en,dir,gray,cout);

    input clk,rst\_n,en,dir;

    output [7:0] gray;

    output cout;

    wire tmpcout;

    counter\_1 first(.clk(clk),.rst\_n(rst\_n),.en(en),.dir(dir),.gray(gray[3:0]),.cout(cout));

   //counter\_1 second(.clk(clk),.rst\_n(rst\_n),.en(tmpcout),.dir(dir),.gray(gray[7:4]),.cout(cout));

endmodule

module counter\_1(dir,clk,en,cout,gray,rst\_n);

    input dir,clk,rst\_n,en;

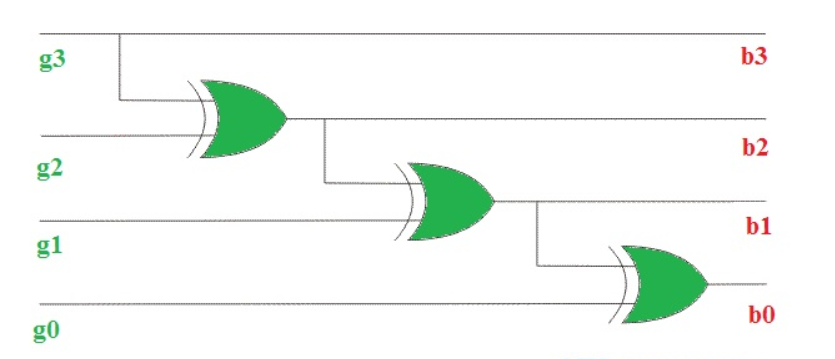
    output cout;

    output [3:0] gray;

    wire [3:0] tmp1,tmp2;

    Gray gray1(.dir(dir),.en(en),.cout(cout),.inputs(tmp2),.outputs(tmp1));

    my\_DFF dff1(.D(tmp1),.clk(clk),.Q(tmp2),.rst\_n(rst\_n));

    assign gray=tmp2; 

endmodule

module Gray(dir,en,cout,inputs,outputs);

    input dir,en;

    output cout;

    output [3:0] outputs;

    input [3:0] inputs;

    reg [3:0] bnum;

    reg g1,g2,g3;

    reg [3:0] outputs;

    reg cout;

    always@(\*)

        begin

            $display("%b",inputs);

            bnum={inputs[3],inputs[3]^inputs[2],inputs[2]^inputs[1],inputs[1]^inputs[0]};

            if(en==1'b0)

            begin

            outputs=inputs;

            end

            else if(en==1'b1)

                begin

                    if(dir==1'b1)

                        begin

                            if(inputs==4'b1000)

                                begin

                                    outputs=4'b0000;

                                    cout=1'b1;

                                end

                            else

                                begin

                                    cout=1'b0;

                                    bnum=bnum+4'b0001;

                                    g3=bnum[3]^bnum[2];

                                    g2=bnum[2]^bnum[1];

                                    g1=bnum[1]^bnum[0];

                                    outputs={bnum[3],g3,g2,g1};

                                end

                        end

                    else if(dir==1'b0)

                        begin

                            if(inputs==4'b0000)

                                begin

                                    outputs=4'b1000;

                                    cout=1'b1;

                                end

                            else

                                begin

                                    cout=1'b0;

                                    bnum=bnum-4'b0001;

                                    g3=bnum[3]^bnum[2];

                                    g2=bnum[2]^bnum[1];

                                    g1=bnum[1]^bnum[0];

                                    outputs={bnum[3],g3,g2,g1};

                                end

                        end

                end

        end

endmodule

module my\_DFF(D,clk,Q,rst\_n);

    input clk,rst\_n;

    input [3:0] D;

    output [3:0]Q;

    reg[3:0] Q;

    always@(negedge clk or negedge rst\_n)

        begin

            if(rst\_n==1'b0) Q=4'b0000;

            else Q<=D;

        end

endmodule

