实验一~实验四报告

实验一:

Led灯控制电路:

激励文件:

module led\_tb;

    reg CLK;

    wire [1:0] led\_out;

*// 被测试的模块实例化*

    led led\_587 (

        .CLK(CLK),

        .led\_out(led\_out)

    );

*// 生成时钟信号*

    initial begin

        CLK = 0;

        forever #5.87 CLK = ~CLK;  *// 使用学号后三位作为时钟周期的一半*

    end

    initial begin

        $monitor("Time = %t, CLK = %b, led\_out = %b", $time, CLK, led\_out);

        #1000000000;

        $finish;

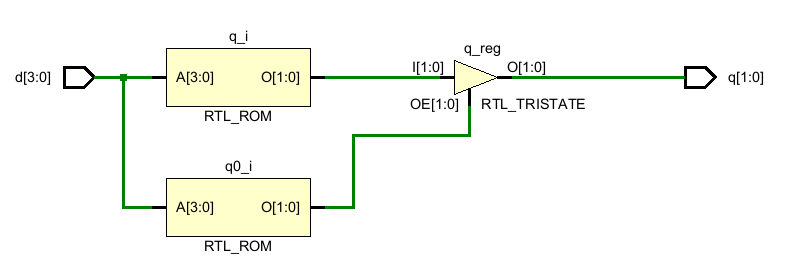
    end

endmodule

仿真结果截图:



电路图:



编码器:

激励文件:

module encoder4\_2\_tb(

    );

reg [3:0] d;

wire [1:0] q;

encoder4\_2 encoder(

    .d(d),

    .q(q)

);

initial begin

    d <= 4'b0111;

    #10

    d <= 4'b1011;

    #10

    d <= 4'b1101;

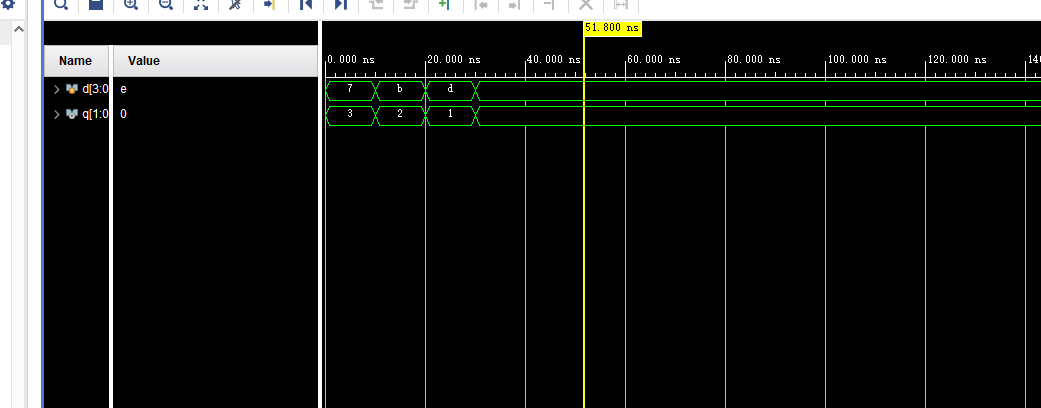
    #10

    d <= 4'b1110;

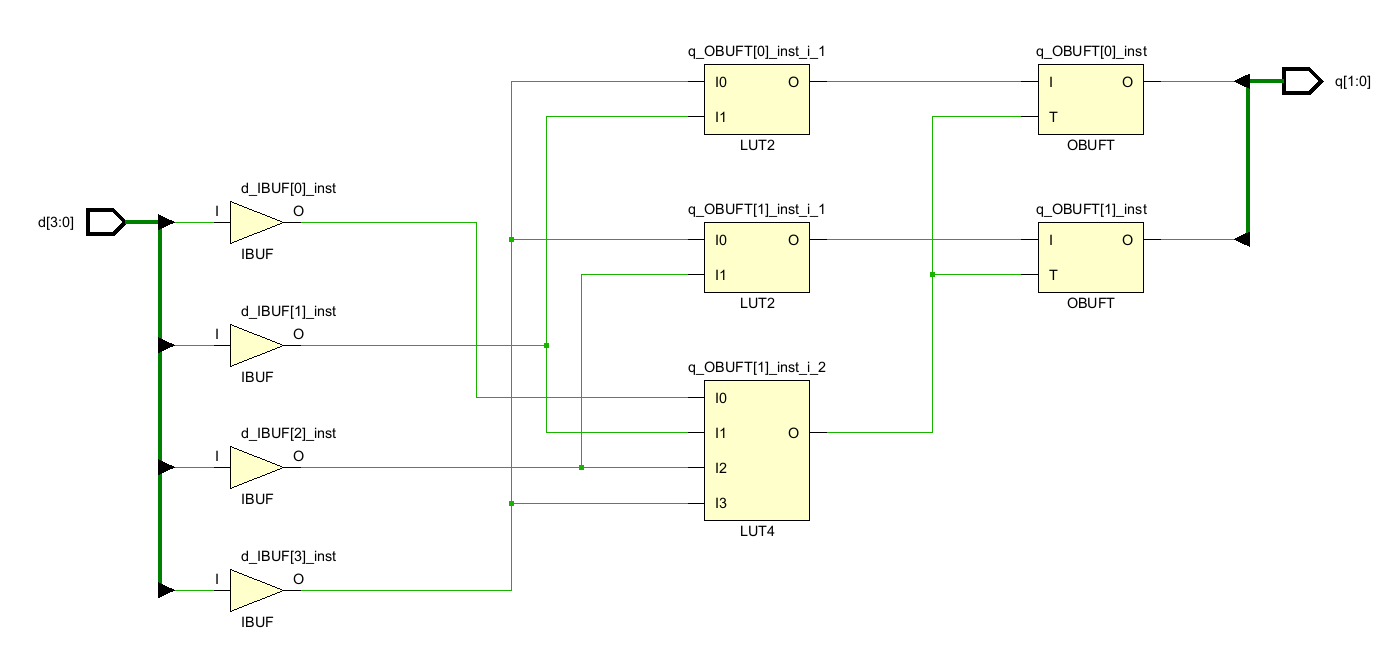
end

endmodule

仿真结果截图:



电路图:



比较器:

激励文件:

module comp\_tb(

    );

    reg CLK,RST;

    reg [1:0] A,B;

    wire AEQB,AGTB,ALTB;

    comp comp2022217587(.CLK(CLK),.RST(RST),.A(A),.B(B),.AGTB(AGTB),.ALTB(ALTB),

    .AEQB(AEQB));

    initial begin

        CLK <=0;

        RST <=0;

        A <=2'b00;

        B <=2'b00;

        #10

        RST <=1;

        #10

        A <=2'b01;

        #10

        B <=2'b01;

        #10

        B <=2'b11;

        #10

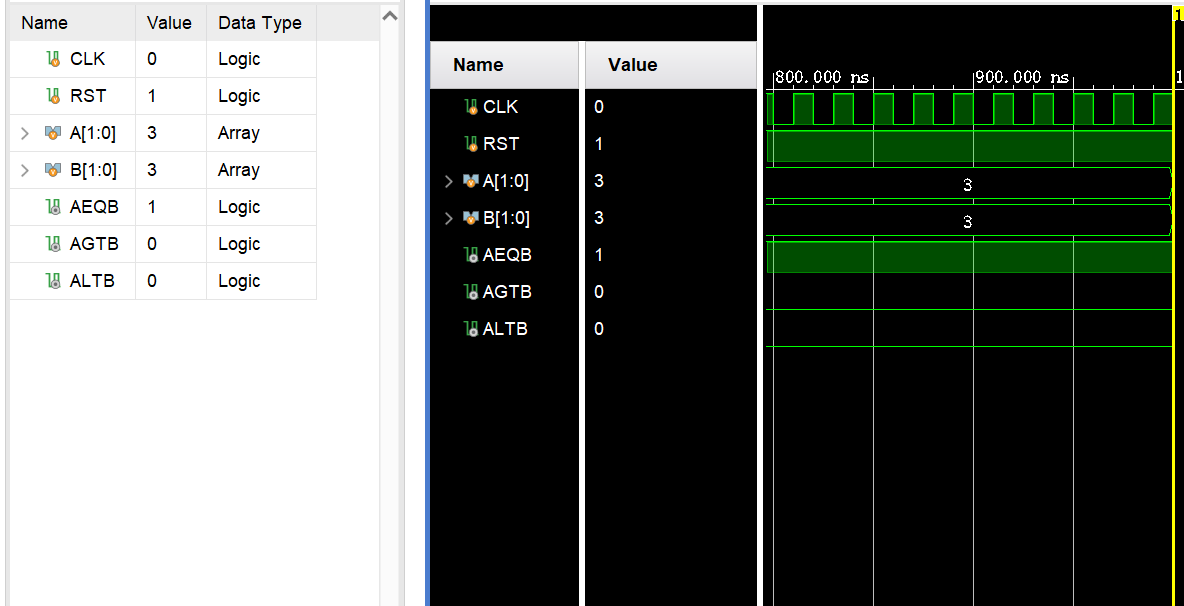
        A <=2'b11;

    end

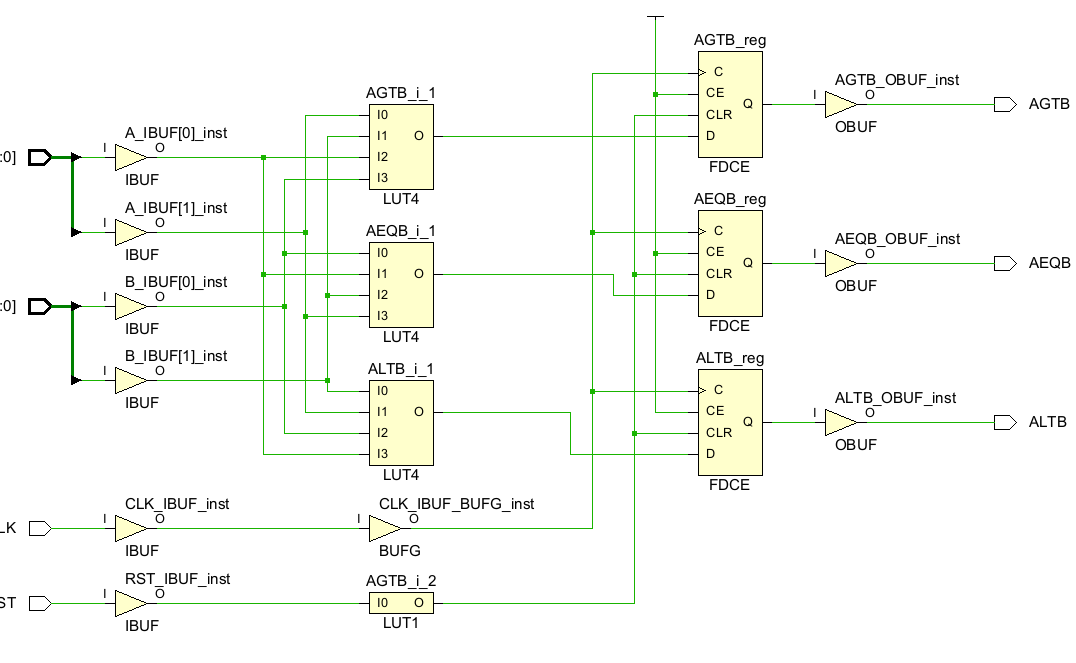
    always #10 CLK =~CLK;

endmodule

仿真结果截图:



电路图:



全加器:

激励文件:

module add\_tb();

    reg a, b, cin;

    wire sum, cout;

    add add587(.cout(cout), .sum(sum), .a(a), .b(b), .cin(cin));

    initial begin

        a = 0;

        b = 0;

        cin = 0;

        repeat (10) begin

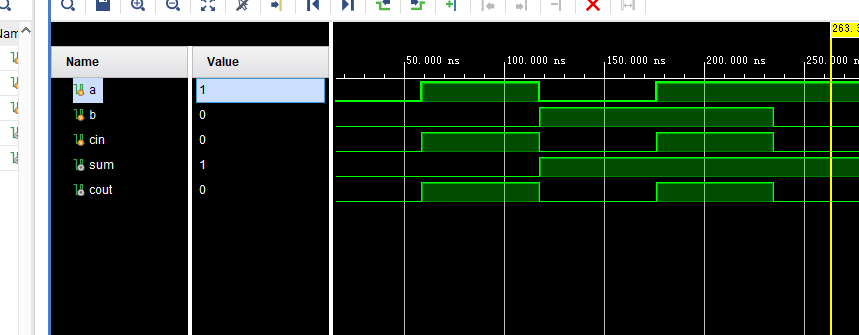
            #58.7 {a, b, cin} = {a, b, cin} + 5;*//我的学号是2022217587 设置时间为58.7毫秒*

        end

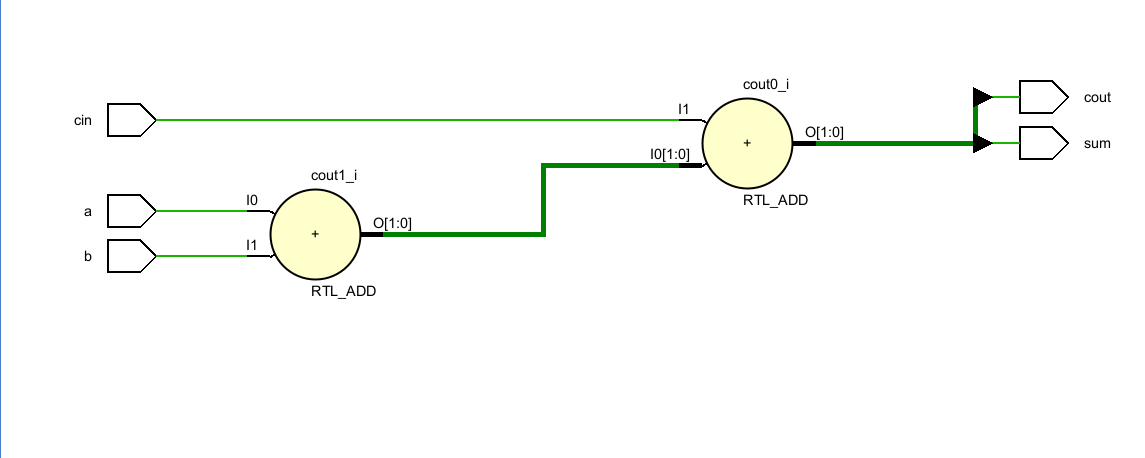
    end

endmodule

仿真结果截图:



电路图:



实验二:

加法计数器:

激励文件:

module addcount\_tb();

    reg clk;

    reg reset;

    wire [3:0] Q;

    addcount addcount587(.clk(clk), .Q(Q), .reset(reset));

    initial begin

        clk = 0;

        reset = 0;

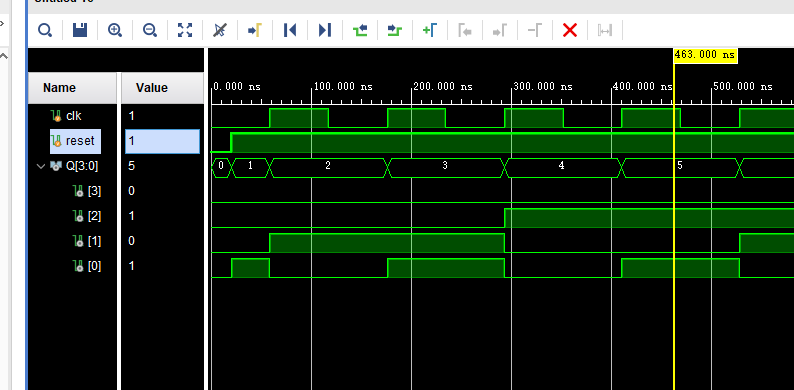
        #20 reset = 1;

    end

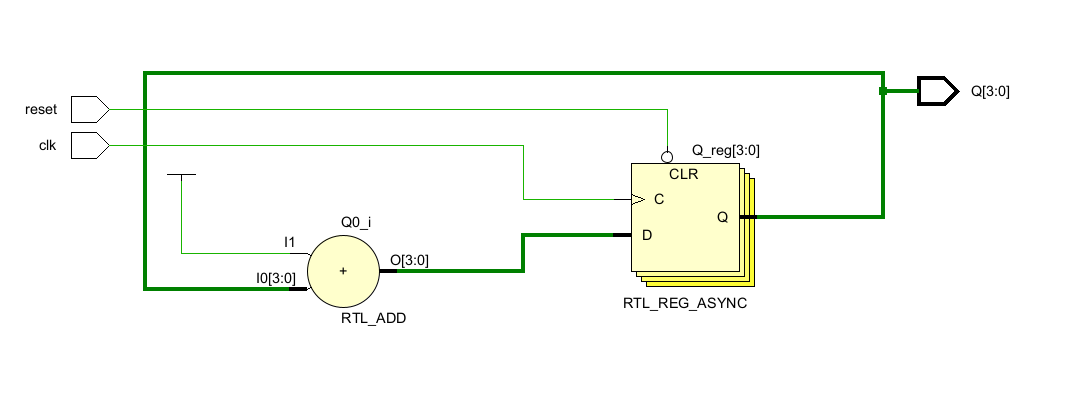
    always #58.7 clk = ~clk;  *//学号后三位作为时钟信号*

endmodule

仿真结果截图:



电路图:



减法计数器:

激励文件:

module reducecounter\_tb(

    );

reg clk;

reg reset;

wire [3:0] Q;

reducecounter reducecount587(.clk(clk), .Q(Q), .reset(reset));

initial begin

    clk =0;

    reset = 0;

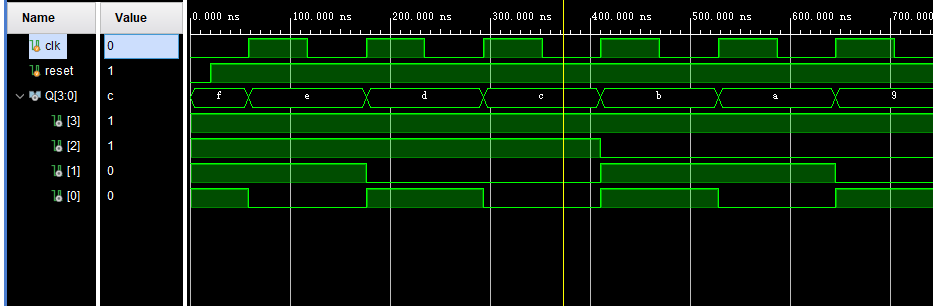
    #20 reset =1;

end

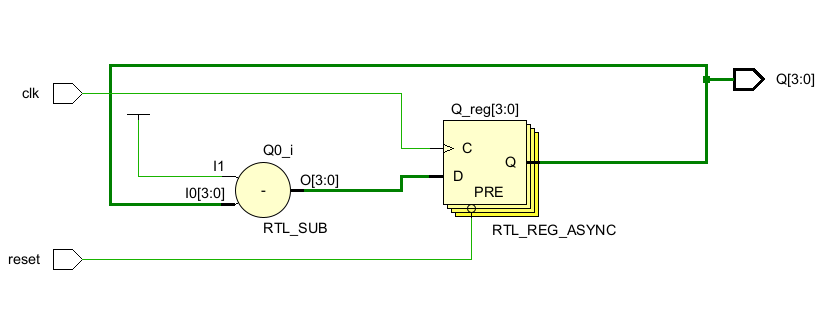
always #58.7 clk =~clk; *//学号后三位作为时钟信号*

endmodule

仿真结果截图:



电路图:



状态机:

激励文件:

module FSM\_tb();

reg Clock;

reg Reset;

reg A;

wire F;

wire G;

fsm Fsm\_587(

    .Clock(Clock),

    .Reset(Reset),

    .A(A),

    .F(F),

    .G(G)

);

always #58.7 Clock = ~Clock; *//学号后三位*

initial begin

    Clock <=0;

    Reset <=1;

    A <=0;

    #5.87 Reset<=0; *//当第一个Clock上升沿出现时state为1000 F为0 G也为0*

*//之后将A设置为1在下一个*

    #117.4 Reset<=1; *//保证后面的状态可以全部展现*

    A<=1; #117.4;

    A<=0; #117.4;

    A<=1; #117.4; *//学号\*2*

    A<=0; #117.4;

    #117.4

    $finish;

end

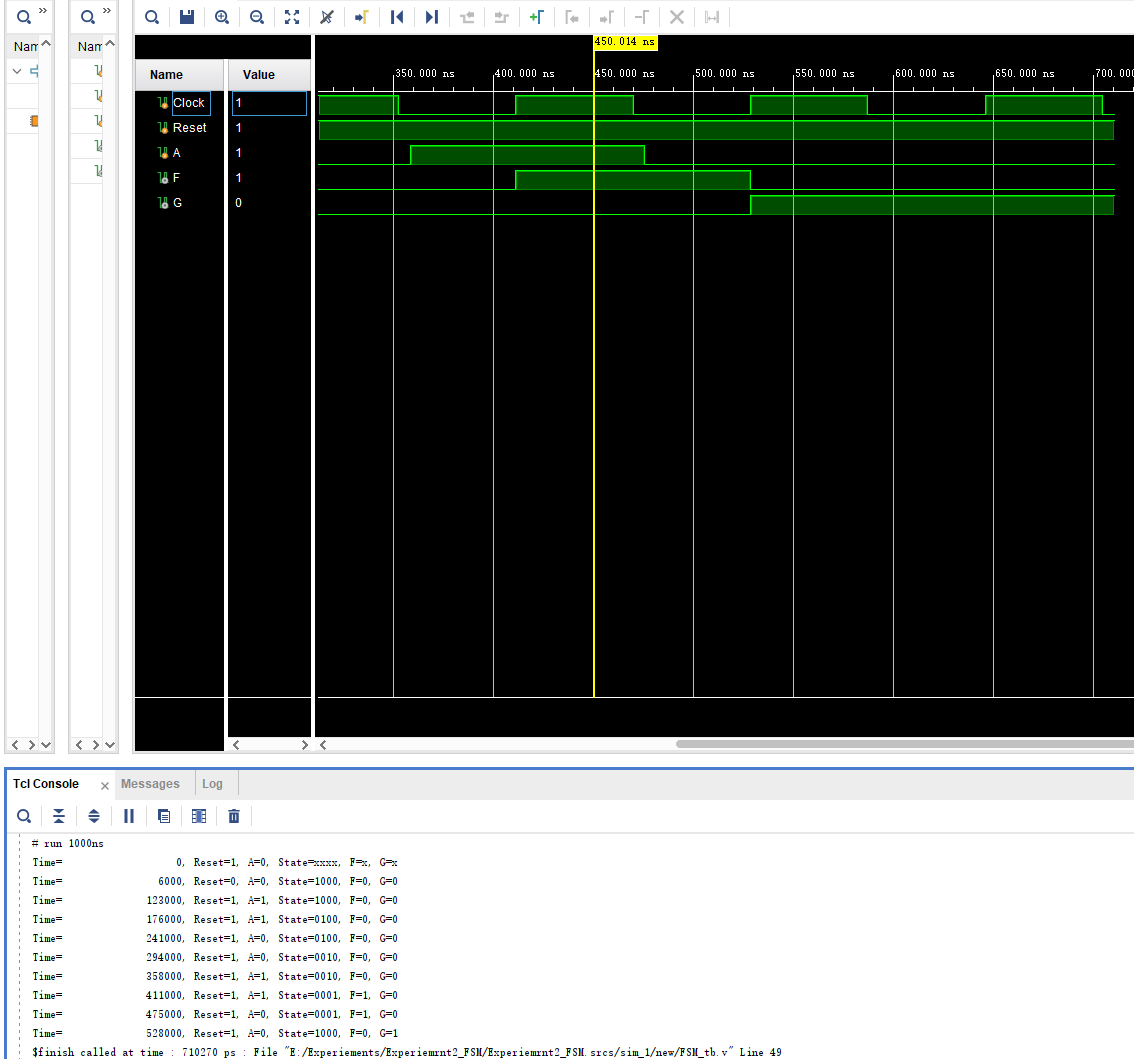
 initial begin

        $monitor("Time=%t, Reset=%b, A=%b, State=%b, F=%b, G=%b", $time, Reset, A,Fsm\_587.state, F, G);

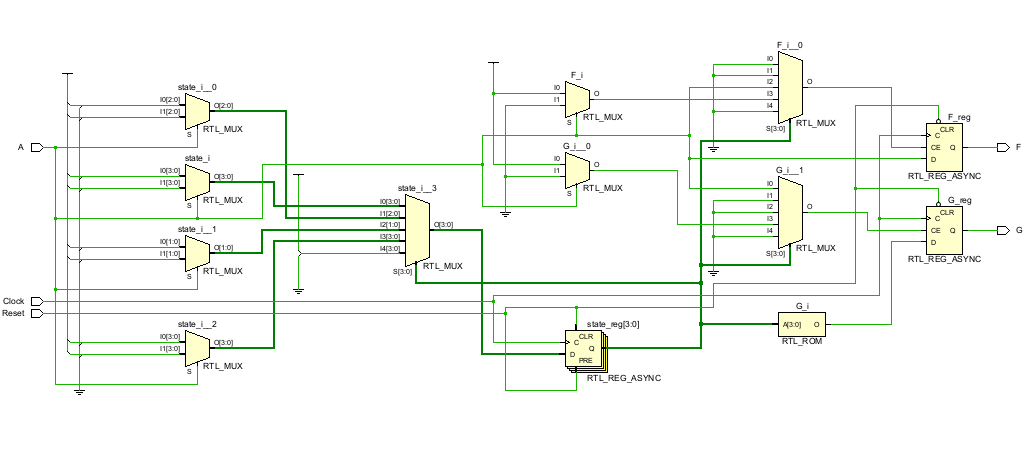
    end

endmodule

仿真结果截图:



电路图:



流水灯:

激励文件:

module led\_tb;

    reg clk;

    reg sw;

    wire [3:0] data;

    led led\_587 (

        .clk(clk),

        .sw(sw),

        .data(data)

    );

    initial begin

        clk = 0;

        sw = 0;

        #5.87;

        sw = 1;

    end

    always #0.587 clk = ~clk; *// 学号后三位*

    initial begin

        $monitor("Time = %t, clk = %b, sw = %b, data = %b", $time, clk, sw, data);

    end

    initial begin

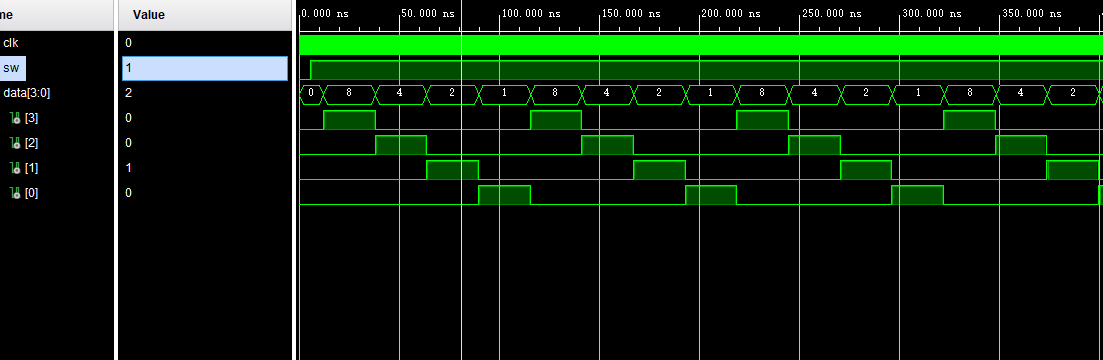
        #1000; *// 运行仿真一段时间，确保分频器有足够的时钟周期工作*

        $finish;

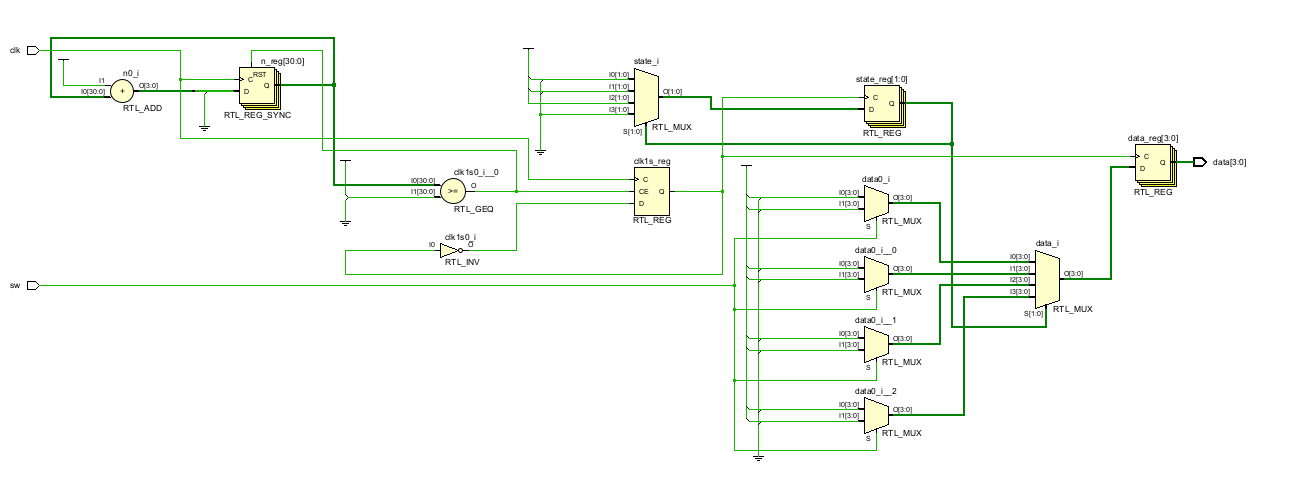
    end

endmodule

仿真结果截图:



电路图:



实验三:

超前进位加法器:

激励文件:

module ahead\_adder\_tb;

    reg [3:0] A;

    reg [3:0] B;

    reg c\_in;

    wire [3:0] F;

    wire c\_out;

    ahead\_adder ahead\_adder\_587 (

        .A(A),

        .B(B),

        .c\_in(c\_in),

        .F(F),

        .c\_out(c\_out)

    );

    initial begin

        A = 4'b0000;

        B = 4'b0000;

        c\_in = 0;

        #587;   *//学号后三位*

        A = 4'b1010;

        B = 4'b1001;

        c\_in = 1;

        #587;

        A = 4'b0010;

        B = 4'b1110;

        c\_in = 0;

        #587;

        A = 4'b0110;

        B = 4'b1101;

        c\_in = 1;

        #587;

        A = 4'b1010;

        B = 4'b1001;

        c\_in = 0;

        #587;

        $finish;

    end

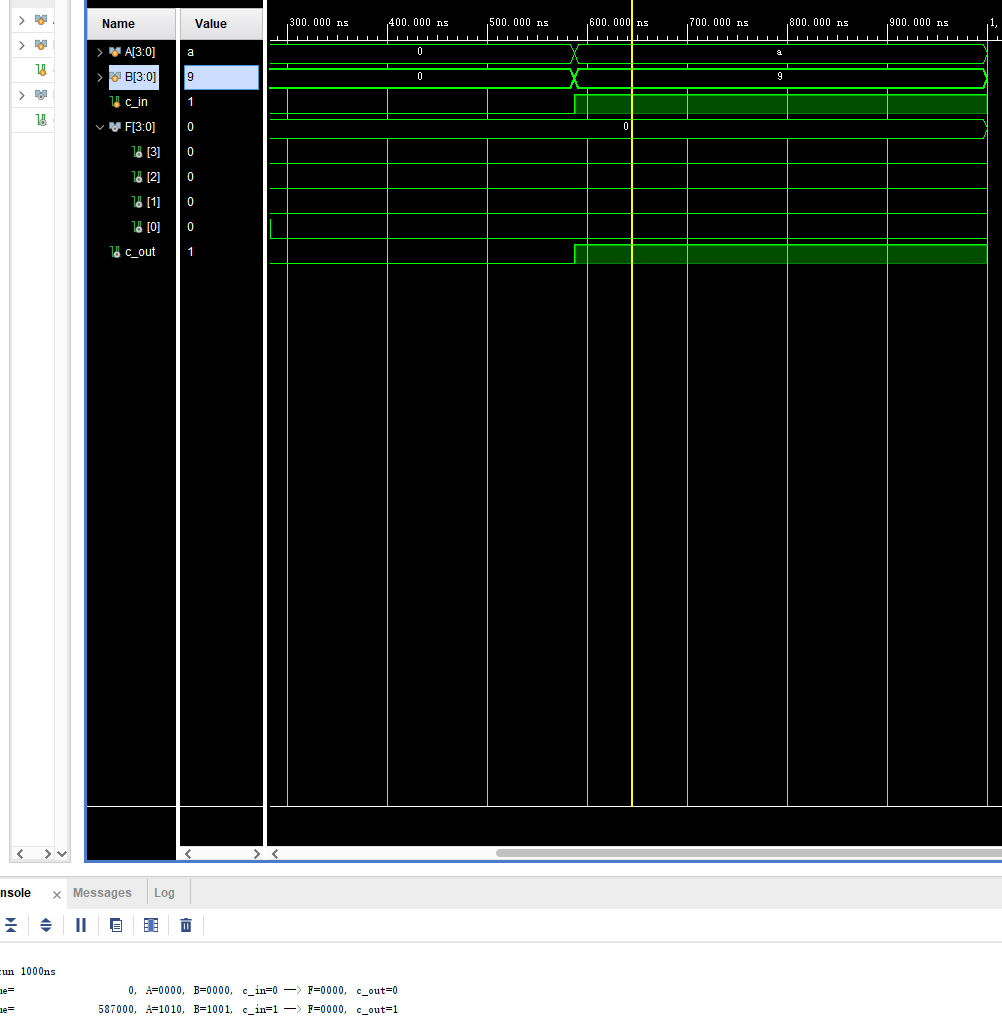
    initial begin

        $monitor("Time=%t, A=%b, B=%b, c\_in=%b --> F=%b, c\_out=%b", $time, A, B, c\_in, F, c\_out);

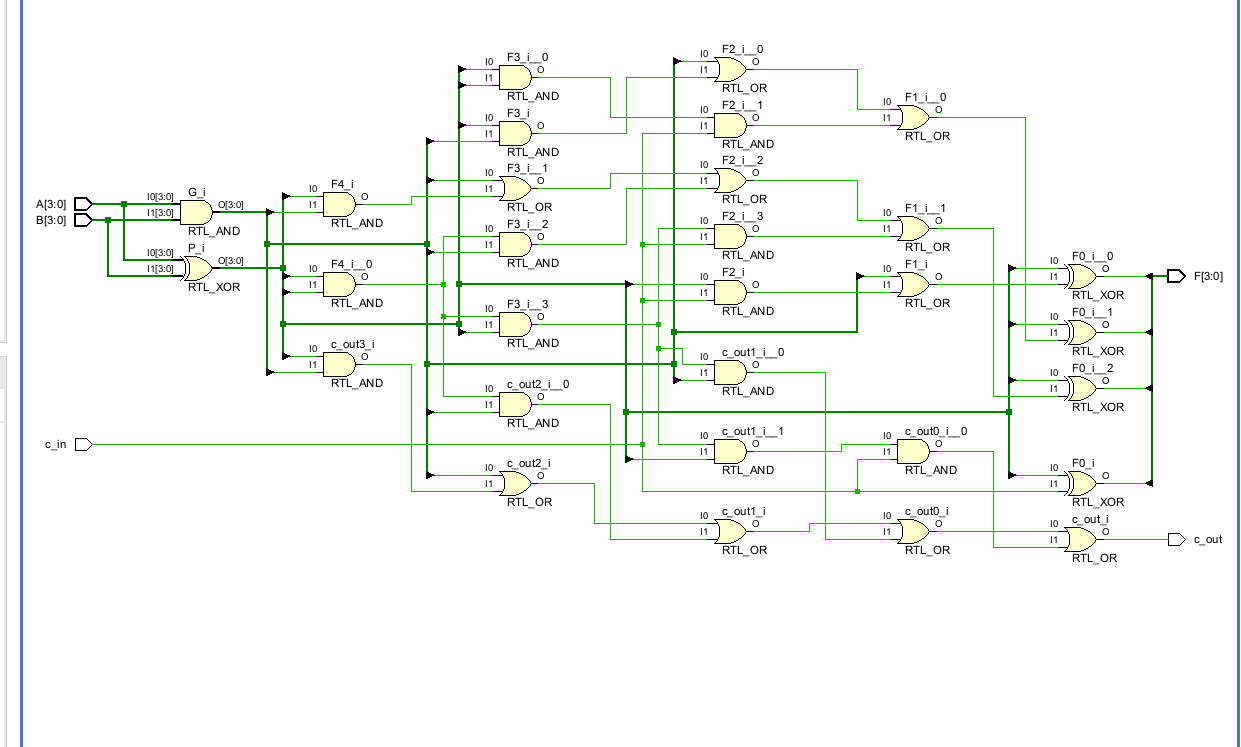
    end

endmodule

仿真结果截图:



电路图:



实验四:

智力抢答器:

激励文件:

`timescale 1ns / 1ps

module Smart\_responder\_tb;

    reg clk,rst\_n;

    reg [3:0]btn;

    wire [3:0] an;

    wire [7:0] seg\_code;

    initial begin

        clk = 1'b0;

        rst\_n = 1'b0;

        btn = 4'd0;

        #5.87 rst\_n = 1'b1;   *//采用学号后三位作为延迟时间*

        #5.87 btn = 4'd1;

        #5.87 btn = 4'd3;

        #5.87 btn = 4'd0;

        #5.87 rst\_n = 1'b0;

        #5.87 rst\_n = 1'b1;

        #5.87 btn = 4'd2;

        #5.87 btn = 4'd6;

        #5.87 btn = 4'd0;

    end

    always #5 clk <= ~clk;

    Smart\_responder Smart\_responder(

        .clk( clk ),

        .rst\_n( rst\_n ),

        .btn( btn ),

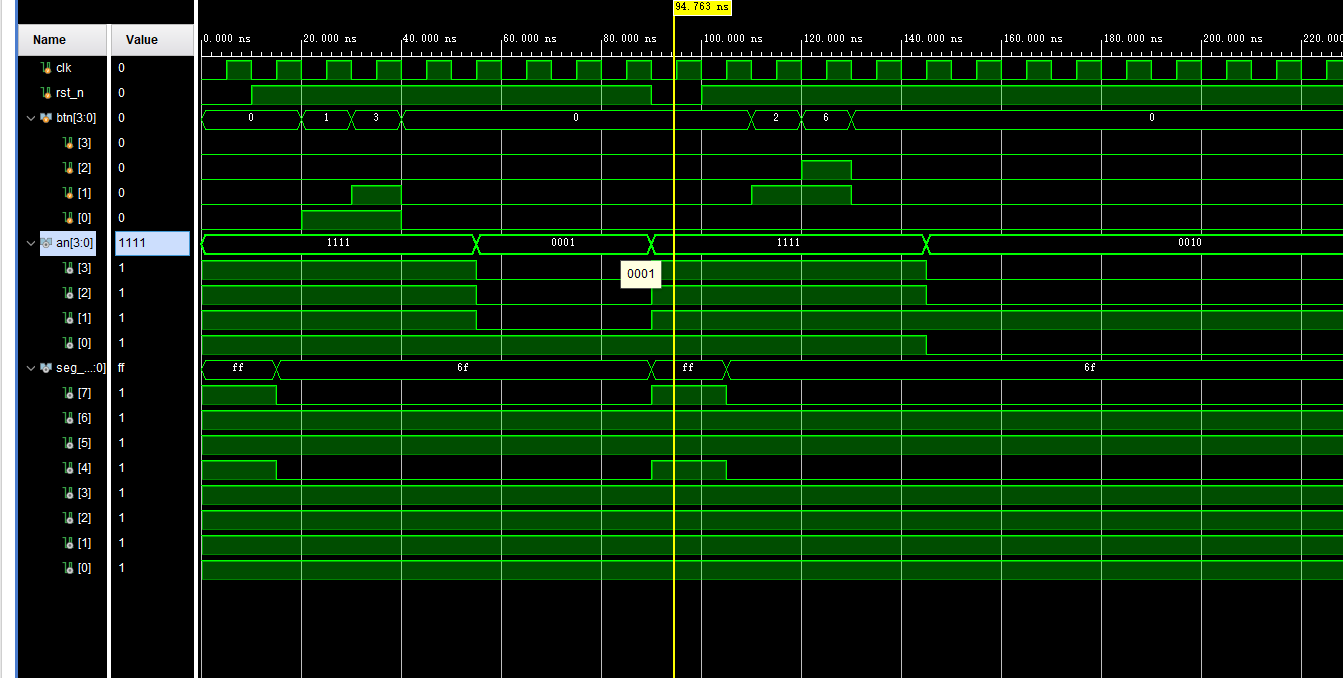
        .an( an ),

        .seg\_code( seg\_code )

    );

endmodule

仿真结果截图:



电路图:  
