آزمایش سوم

گزارش آزمایش سوم

محمدپیام تائبی ٤٠٠١٠٤٨٦٧

امیرحسین علمدار ٤٠٠١٠٥١٤٤

علیرضا سلیمیان ٤٠٠١٠٥٠٣٦

ابتدا یک cascade 1 bit comparator میسازیم که در ادامه با کنار هم گذاشتن تعدادی از آنها مقایسه کننده های بزرگتری بسازیم

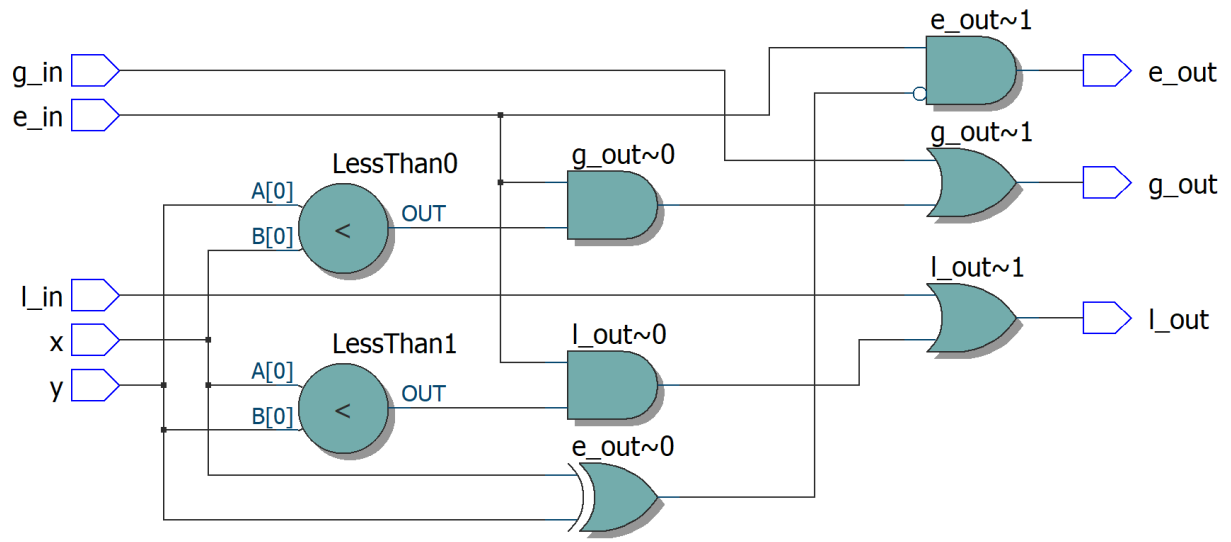
```
Ln#
1  module one_bit_comparator(input g_in, input e_in, input l_in, input x, input y, output g_out, output e_out, output l_out);
2          assign g_out = g_in | (e_in & (x > y));
3          assign e_out = e_in & (x == y);
4          assign l_out = l_in | (e_in & (x < y));
5  endmodule
6
```

در ادامه با کنار هم گذاشتن ۴ تا از مقایسه کننده بالا یک مقایسه کننده ی ۴ بیتی میسازیم:

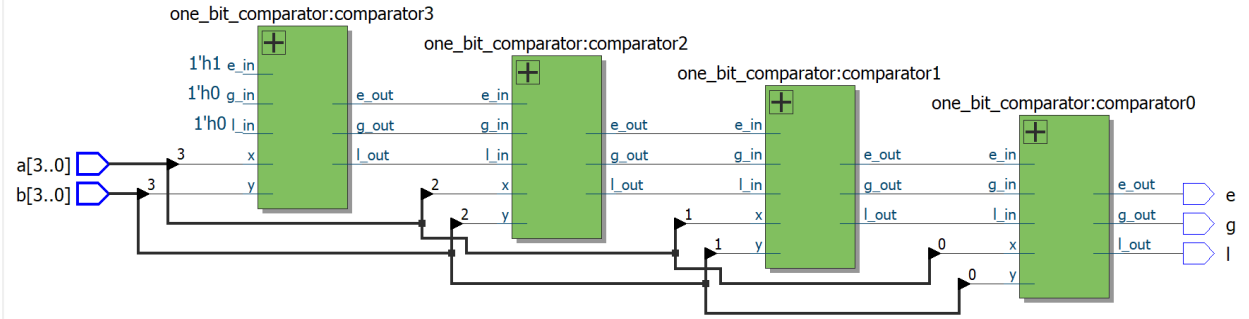
```
1  module four_bit_comparator(input[3:0] a, input[3:0] b,output g, output e, output l);
2
3  wire[8:0] c;
4
5  one_bit_comparator comparator3(.g_in(1'b0), .e_in(1'b1), .l_in(1'b0), .x(a[3]), .y(b[3]), .g_out(c[8]), .e_out(c[7]), .l_out(c[6]));
6
7  one_bit_comparator comparator2(.g_in(c[8]), .e_in(c[7]), .l_in(c[6]), .x(a[2]), .y(b[2]),.g_out(c[5]), .e_out(c[4]), .l_out(c[3]));
8
9  one_bit_comparator comparator1(.g_in(c[5]), .e_in(c[4]), .l_in(c[3]), .x(a[1]), .y(b[1]),.g_out(c[2]), .e_out(c[1]), .l_out(c[0]));
10
11 one_bit_comparator comparator0(.g_in(c[2]), .e_in(c[1]), .l_in(c[0]), .x(a[0]), .y(b[0]),.g_out(g), .e_out(e), .l_out(l));
12
13 endmodule
```

که در ادامه مدار این ماژول ها را نیز میتوانید ببینید :

برای کد اول :



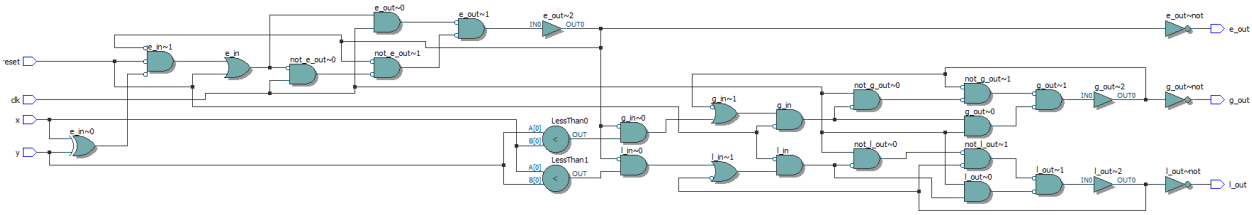
و برای کد دوم :



برای بخش دوم سوال باید یک مقایسه کننده ی سریال بسازیم که کد وریلاگ آن به شکل زیر است:

```
module serial_comparator(input reset, input clk, input x, input y, output g_out, output e_out, output l_out);

    wire not_g_out, not_e_out, not_l_out;
    wire g_in, e_in, l_in;

    assign g_in = ((e_out & (x > y)) | g_out) & (~reset);
    assign e_in = ((e_out & (x == y)) & (~reset)) | (reset);
    assign l_in = ((e_out & (x < y)) | l_out) & (~reset);

    assign g_out = ~(not_g_out & ~(clk & g_in));
    assign not_g_out = ~(g_out & ~(clk & ~(g_in)));

    assign e_out = ~(not_e_out & ~(clk & e_in));
    assign not_e_out = ~(e_out & ~(clk & ~(e_in)));

    assign l_out = ~(not_l_out & ~(clk & l_in));
    assign not_l_out = ~(l_out & ~(clk & ~(l_in)));


endmodule
```

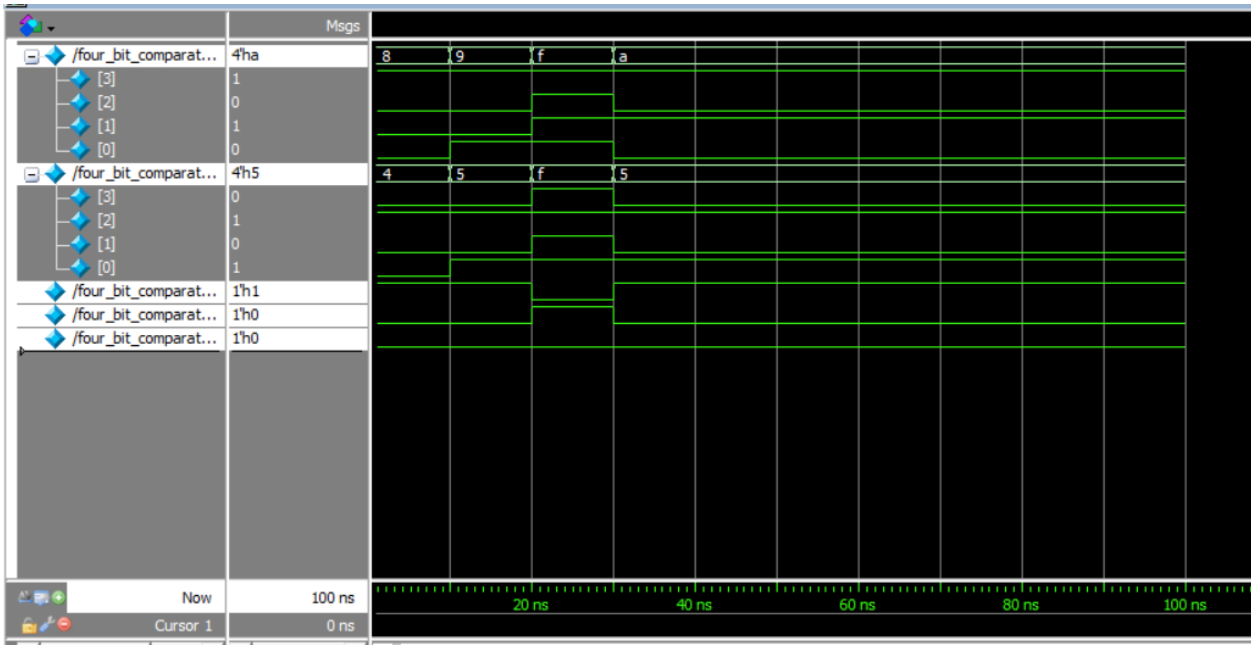
که مدار نهایی ما نیز به این شکل خواهد بود:



در آخر ابتدا برای مقایسه کننده ۴ یک تست بنچ مینویسیم:

```
module four_bit_comparator_test_bench();
        reg[3:0] x;
        reg[3:0] y;
        wire g, e, l;
        four_bit_comparator comparator(.a(x), .b(y), .g(g), .e(e), .l(l));

        initial
begin

                x = 4'b1000;
                y = 4'b0100;
                #10
                x = 4'b1001;
                y = 4'b0101;
                #10
                x = 4'b1111;
                y = 4'b1111;
                #10
                x = 4'b1010;
                y = 4'b0101;
                #10;
end
endmodule
```

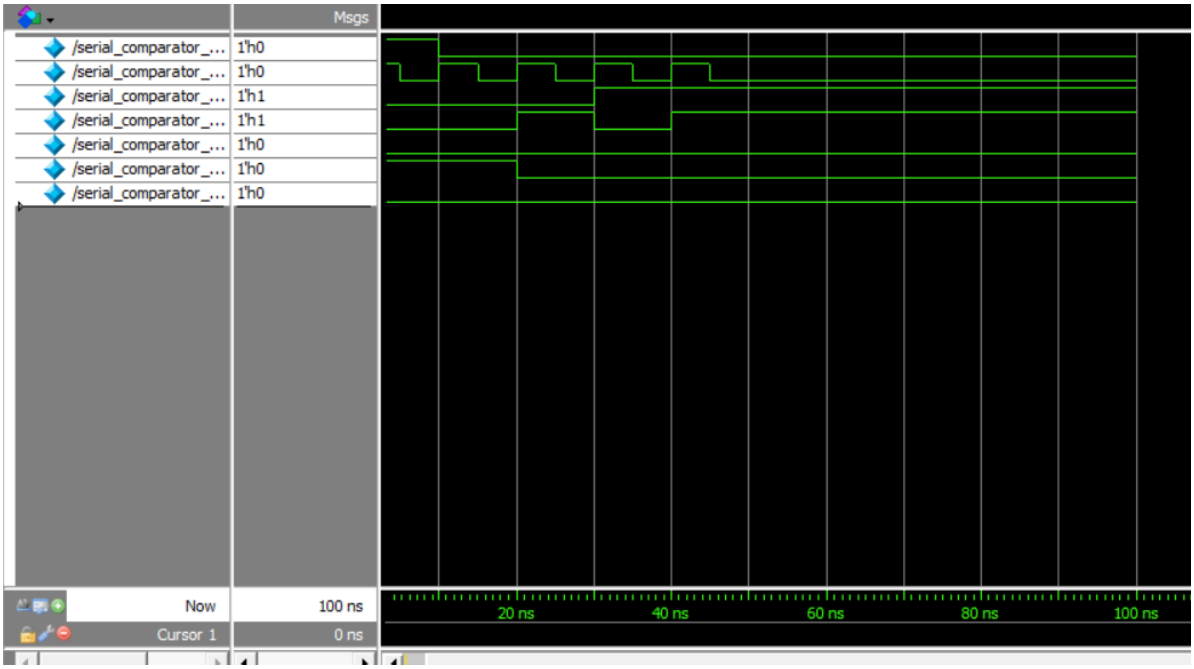
که پس از اجرا :



و در آخر برای مقایسه کننده سریال یک تست بنچ مینویسیم :

```
module serial_comparator_test_bench();
    reg reset, clk, x, y;
    wire g_out, e_out, l_out;
    serial_comparator comparator(.reset(reset), .clk(clk), .x(x), .y(y), .g_out(g_out), .e_out(e_out), .l_out(l_out));
    initial begin
        reset = 1;
        clk = 1;
        x = 0;
        y = 0;
        #5;
       clk = 0;
        #5;

        reset = 0;
        clk = 1;
        #5;
        clk = 0;
        #5;

        x = 0;
        y = 1;
        clk = 1;
        #5;

        clk = 0;
        #5;

        x = 1;
        y = 0;
        clk = 1;
        #5;

        clk = 0;
        #5;

        x = 1;
        y = 1;
        clk = 1;
        #5;

        clk = 0;
        #5;
    end
endmodule
```

که پس از اجرا :



در آخر تمام سه کد را روی برد FPGA سنتر میکنیم و کارکرد آن را بررسی میکنیم