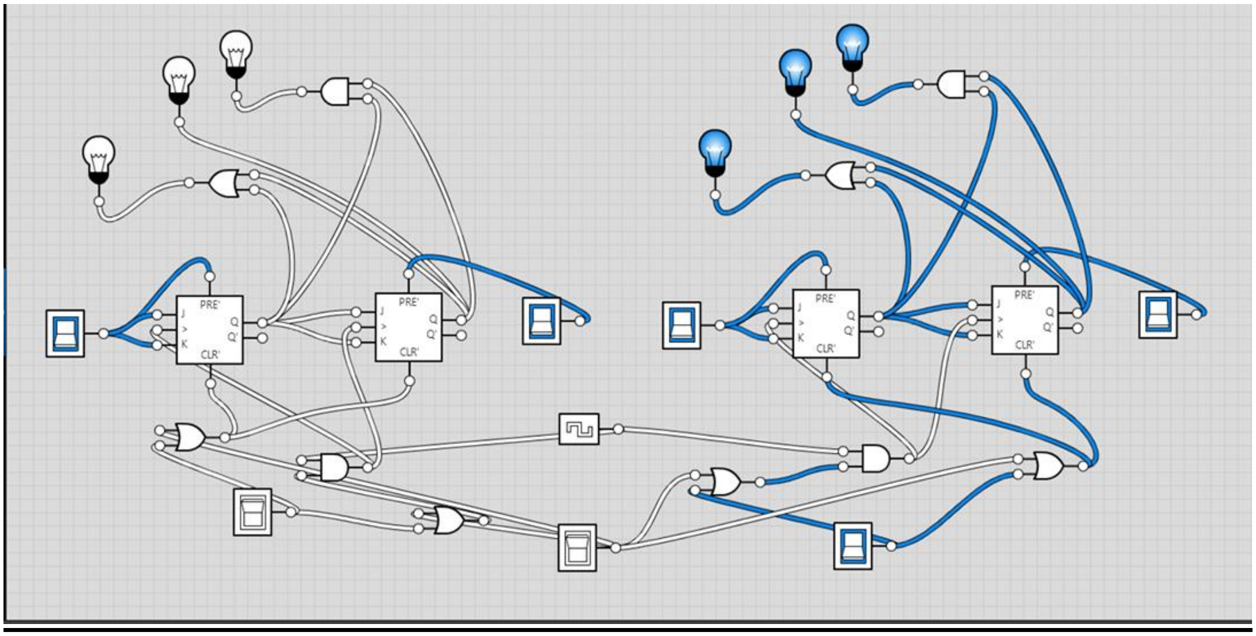
1. Car Light



----- car_EZ.v -----

```
module T_FF(q_out, t, clk, reset);
    output q_out;
    input t,clk, reset;
    reg q_out;

    initial
        begin
            q_out=1'b0;
        end

    always @ (posedge clk or posedge reset)
    if(reset)
        q_out <= 0;
    else
        begin
            q_out <= (t == 1) ? ~q_out : q_out;
        end
endmodule


module car_light(q_out, t, clk, reset);
    output [2:0] q_out;
    input t, clk, reset;

    wire q0, q1;
    T_FF t_ff1(q0, t, clk, reset);
    T_FF t_ff2(q1, q0, clk, reset);

    assign q_out[0] = q0 | q1 ;
    assign q_out[1] = q1;
    assign q_out[2] = q0 & q1;

endmodule

module car(left, right, l , r, E, clk);

    output [2:0] left, right;
    input l, r, E, clk;

    wire or_clk_out_1, re1, clk_left;
    wire or_clk_out_2, re2, clk_right;

    nor or_re_1(re1, left, E);
    or or_clk_1(or_clk_out_1, l, E);
    and and_clk_1(clk_left, or_clk_out_1, clk);
    car_light car_light_left(light_left, 1'b1, clk_left, re1);

    nor or_re_2(re2, r, E);
    or or_clk_2(or_clk_out_2, r, E);
    and and_clk_2(clk_right, or_clk_out_2, clk);
    car_light car_light_right(light_right, 1'b1, clk_right, re2);

endmodule
```

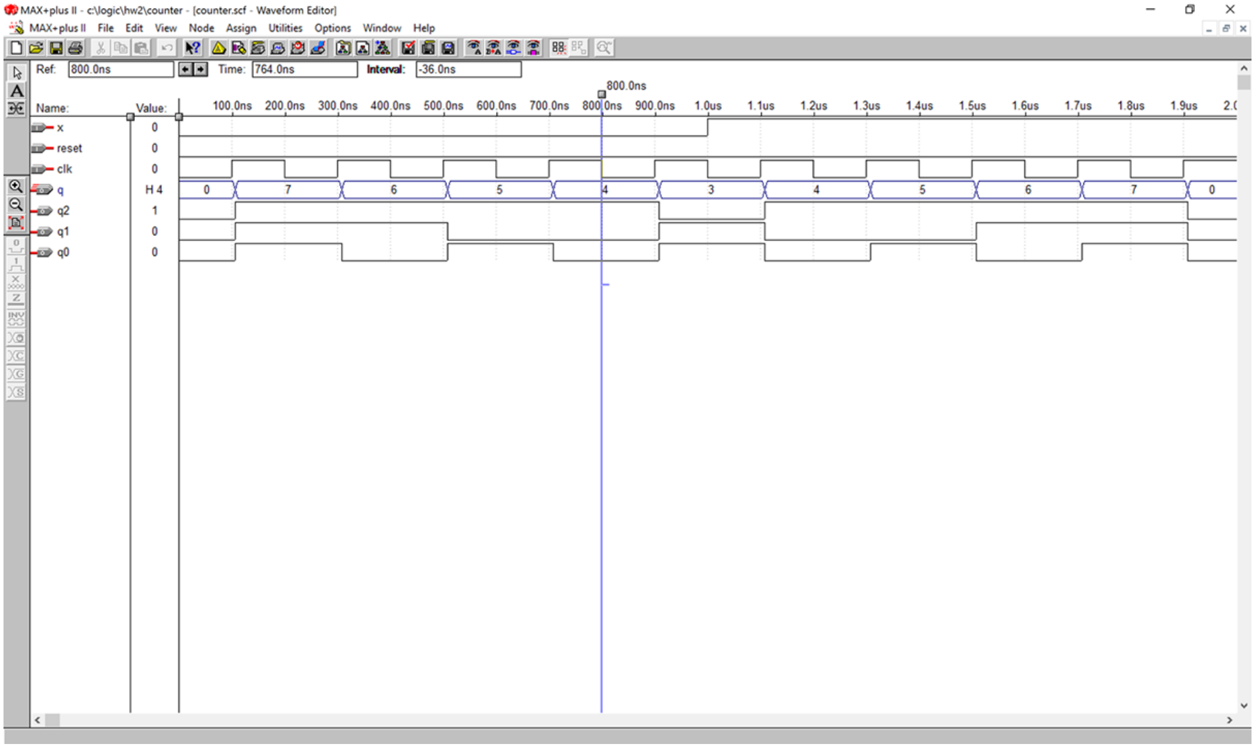
----- car_EZ_stimulus.v -----

```
module car_light_stimulus;
    reg l, r, E, clk;
    reg din_0, din_1;
    wire [2:0] left, right;

    car_light car1(left, right, left , right, E, clk);

    initial
        begin
            $dumpfile("TimeDiagram.vcd");
            $dumpvars(0,car_light_stimulus);
            l = 1'b0;
            r = 1'b0;
            E = 1'b0;
            clk = 1'b0;
        end

    always
        #5 clk = !clk;

    initial
        begin
          #10 l = !l;
          #40 l = !l;
              r = !r;
          #40 l = !l;
          #40 l = !l;
              r = !r;
          #10 E = !E;
        end

    initial
    begin
        #250 $finish;
    end
    initial
        $monitor ($time, clk, left, right, E);
endmodule
```

Count Up and Down

```
module T_FF(q, t, clk, reset);
    output q;
    input t,clk, reset;
    reg q;

    initial
        begin
            q=1'b0;
        end

    always @ (posedge clk)
    if(reset)
        q <= 0;
    else
        begin
            q <= (t == 1) ? ~q : q;
        end
endmodule

module counter(q, x, clk, reset);

output [2:0] q;
input x, clk, reset;
wire and_1,and_2,or_1,and2_1,and2_2,or_2,q1, q2, q3;

T_FF t1(q1, 1'b1, clk, reset);

and and1(and_1, !x, !q1);
and and2(and_2, x, q1);
or or1(or_1, and_1, and_2);

T_FF t2(q2,  or_1, clk, reset);

and and3(and2_1, and_1, !q2);
and and4(and2_2, and_2, q2);
or or2(or_2, and2_1, and2_2);

T_FF t3(q3, or_2, clk, reset);

assign q[0] = q1;
assign q[1] = q2;
assign q[2] = q3;

endmodule
```

Time Diagram



MAX+plus II - c:\logic\hw2\counter - [counter.scf - Waveform Editor]

Ref: 800.0ns   Time: 764.0ns   Interval: -36.0ns

Name:	Value:
x	0
reset	0
clk	0
q	H 4
q2	1
q1	0
q0	0