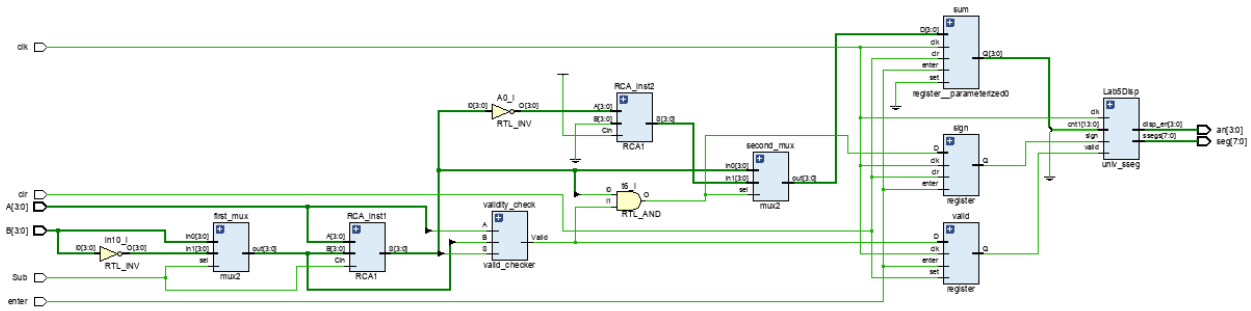
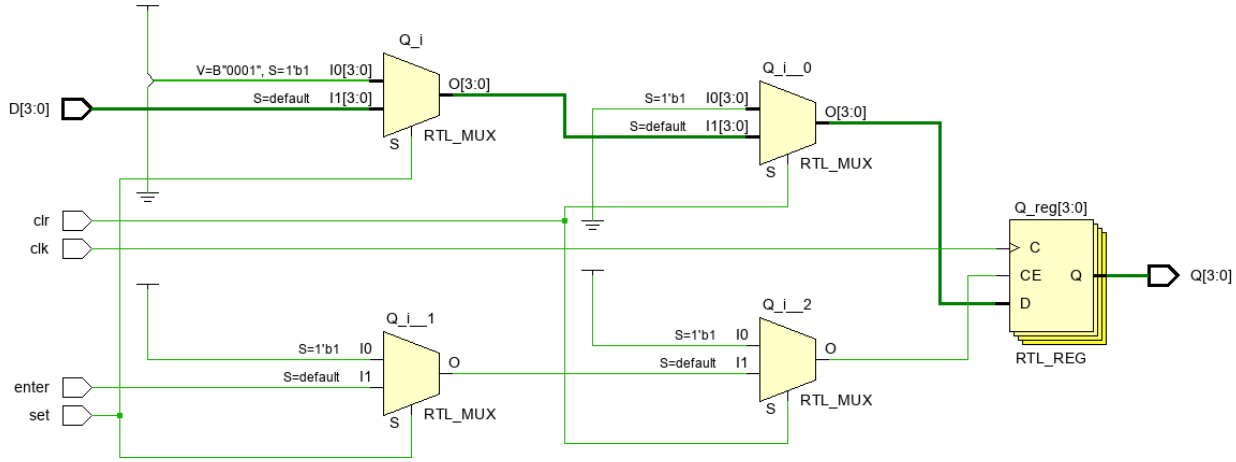
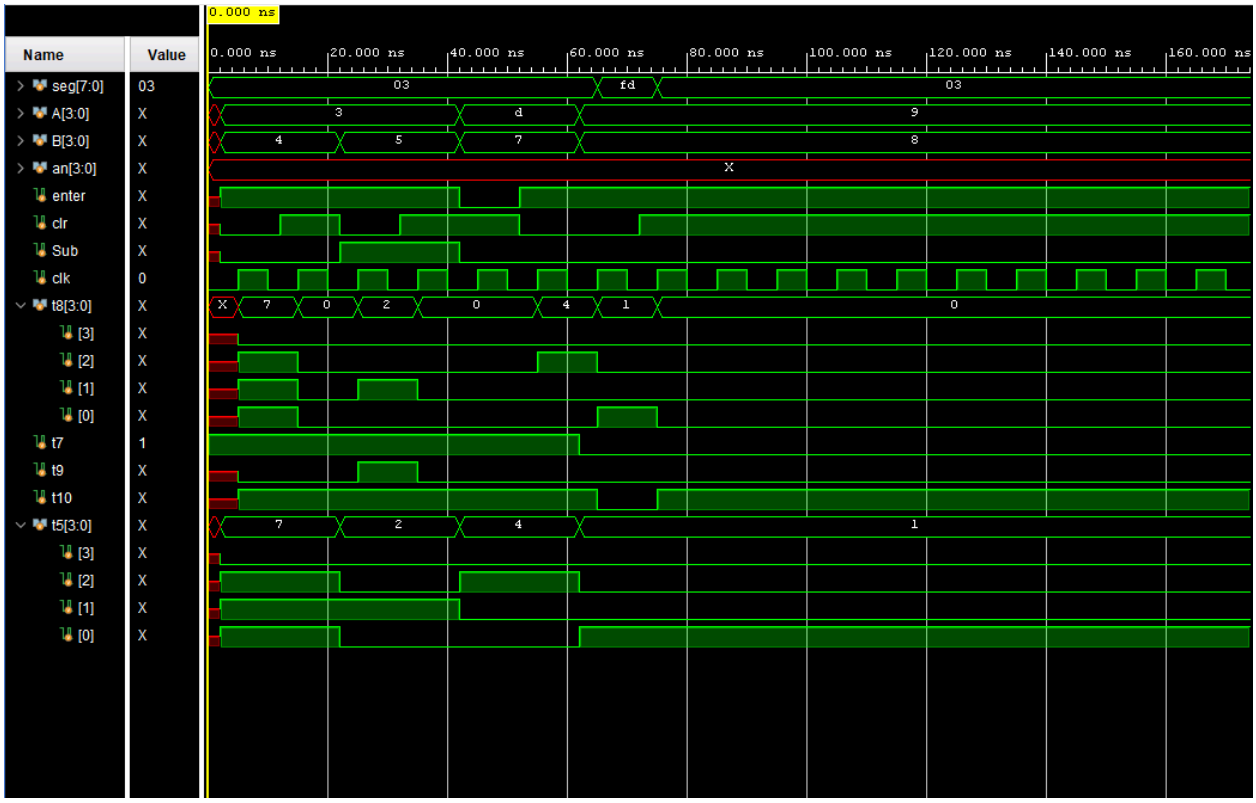
Lab5 Top Level Schematic



Register schematic



Top level schematic



Top Level Design Code:

```
module Lab5(
    input [3:0] A ,
    input [3:0] B ,
    input Sub,
    input enter,
    input clr,
    input clk,
    output [7:0] seg,
    output [3:0] an
    //output neg
    );

    logic [3:0] t1;
    logic [3:0] t3;
    logic [3:0] t4;
    logic [3:0] t5;
    logic t6;
    logic t7;
    logic [3:0] t8;
```

```
    logic t9;
    logic t10;
    assign t6 = t3[3] & t7;
    //assign neg = t6;

    RCA1 RCA_inst1 (.A(A), .B(t1),.Cin(Sub), .S(t3), .C0());
    mux2 first_mux(.in0(B),.in1(~B),.sel(Sub),.out(t1));
    RCA1 RCA_inst2 (.A(~t3),.B(4'b0000),.Cin(1),.S(t4), .C0());
    mux2 second_mux(.in0(t3),.in1(t4),.sel(t6),.out(t5));
    valid_checker validity_check(.A(A[3]),.B(t1[3]),.S(t3[3]),.Valid(t7));
    register #(1) sign(.clk(clk), .enter(enter), .clr(clr), .D(t6), .Q(t9));
    register #(1) valid(.clk(clk), .enter(enter), .D(t7), .set(clr), .Q(t10));
    register #(4) sum(.clk(clk), .enter(enter), .D(t5), .clr(clr), .set(1'b0), .Q(t8));
    //mux2 third_mux(.in0(4'b1111),.in1(4'b1110),.sel(t7),.out(an));
    univ_sseg Lab5Disp (.clk(clk), . cnt1({10'b0,t8}), .sign(t9), .valid(t10), .ssegs(seg),
.disp_en(an));
   // SevenSegDecoder ssegdec1(.binary(t5),.seg(seg));


endmodule
```

Top Level Sim Code:

```
module Lab5sim();
    logic[7:0]seg;
    logic[3:0]A,B,an;
    logic enter, clr, Sub, clk;
    logic neg;
    Lab5 lab5sim(.*);
    always begin
        clk = 0;  #5;
        clk = 1;  #5;
        end
    initial begin
        #2;
        A=3; B=4; enter = 1; clr=0; Sub=0;
        #10 ;
        clr = 1;
        #10;
        A=3; B=5; enter = 1; clr = 0; Sub = 1;
        #10;
        clr = 1;
        #10;
```

```
        A=-3; B=7; enter = 0; clr = 1; Sub =0;
        #10;
        clr=0;
        enter=1;
        #10;
        A=-7; B=-8; enter = 1; clr = 0; Sub = 0;
        #10;
        clr = 1;
        #10;
    end
endmodule
```

Register Design Code:

```
module register # (parameter WIDTH = 4) (
    input clk, enter, clr, set,
    input [WIDTH-1:0] D,
    output logic [WIDTH-1:0] Q );
    always_ff @(posedge clk)
    begin
        if (clr)
        Q <= 0;
        else if (set)
        Q <= 1;
        else if (enter)
        Q <= D;
    end
endmodule
```

Register Sim:

```
module register_sim();
    logic clk,clr,enter,set;
    logic [3:0]D,Q;

    register #(4) regsim(.*);

    always
        begin
            clk = 0;
            #5
            clk = 1;
```

```
        #5;
    end
  initial
    begin
        clr = 0; enter = 0; set = 1; D = 4'b1001;
        #10;
        clr = 1; enter = 0; set = 0; D = 4'b1001;
        #10;
        clr = 0; enter = 1; set = 0; D = 4'b1001;
        #10;
    end


endmodule
```