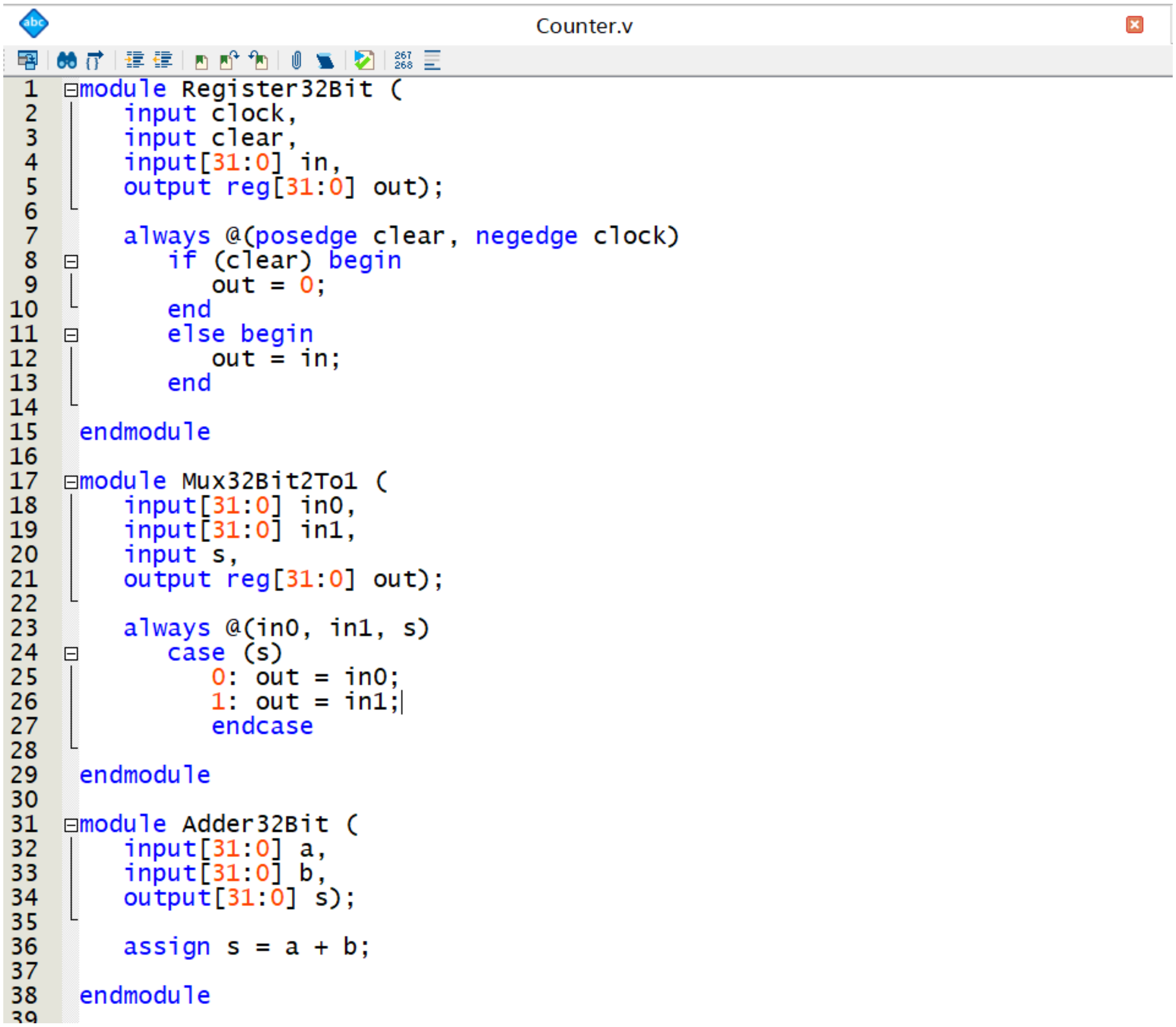
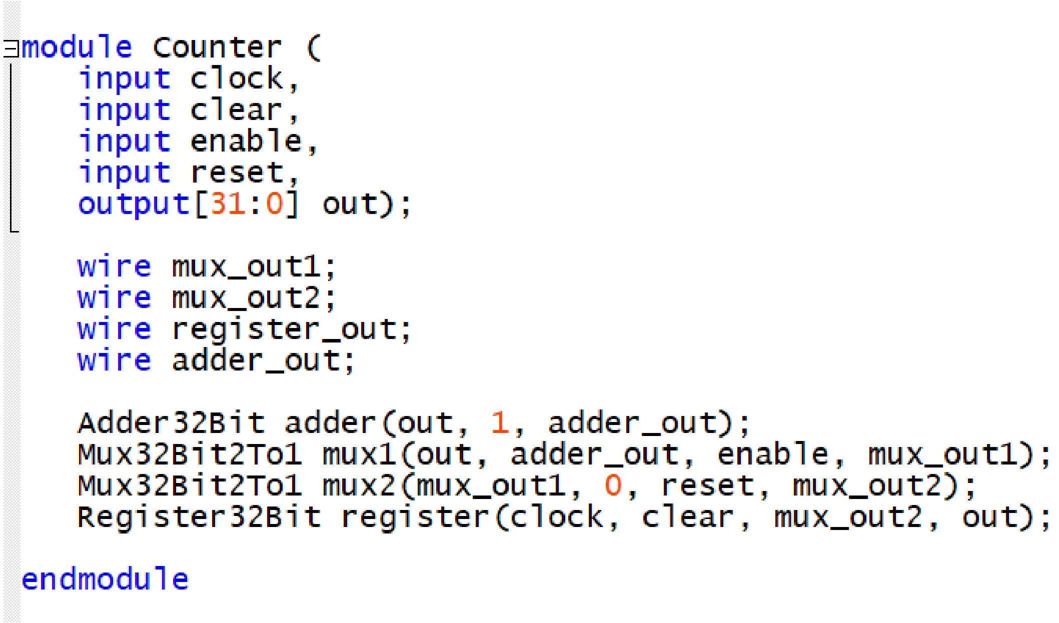


```
module Register32Bit (
    input clock,
    input clear,
    input[31:0] in,
    output reg[31:0] out);

    always @(posedge clear, negedge clock)
        if (clear) begin
            out = 0;
        end
        else begin
            out = in;
        end

endmodule

module Mux32Bit2To1 (
    input[31:0] in0,
    input[31:0] in1,
    input s,
    output reg[31:0] out);

    always @(in0, in1, s)
        case (s)
            0: out = in0;
            1: out = in1;
            endcase

endmodule

module Adder32Bit (
    input[31:0] a,
    input[31:0] b,
    output[31:0] s);

    assign s = a + b;

endmodule
```



```
module Counter (
    input clock,
    input clear,
    input enable,
    input reset,
    output[31:0] out);

    wire mux_out1;
    wire mux_out2;
    wire register_out;
    wire adder_out;

    Adder32Bit adder(out, 1, adder_out);
    Mux32Bit2To1 mux1(out, adder_out, enable, mux_out1);
    Mux32Bit2To1 mux2(mux_out1, 0, reset, mux_out2);
    Register32Bit register(clock, clear, mux_out2, out);

endmodule
```