```
`timescale 1ns / 1ns

module alu_testbench();
    reg [31:0] a;
    reg [31:0] b;
    reg [2:0] sel;

    wire[31:0] out_data;
    reg fi = 1'b0;

    alu tu(
        .a(a),
        .b(b),
        .sel(sel),
        .out_data(out_data)
    );

    initial begin
        $dumpfile("alu_testbench.vcd");
        $dumpvars(0, alu_testbench);

        a = 32'd10;
        b = 32'd2;
        b = -b;
        sel = 3'b0;
        #1;
        $display("Signed Add: a = %0d, b = %0d, out = %0d", a, $signed(b), out_data);
        if (out_data != 32'd8) begin
            fi = 1'b1;
        end
        sel = 3'b1;
        a = 32'd1;
        #1;
        $display("Unsiged Add: a = %0d, b = %0d, out = %0d", a, $unsigned(b),
out_data);
        if (out_data != 32'd4294967295) begin
            fi = 1'b1;
        end
        sel = 3'b10;
        a = 32'd10;
        b = 32'd2;
        b = -b;
        #1;
        $display("Signed Subtract: a = %0d, b = %0d, out = %0d", a, $signed(b),
out_data);
        if (out_data != 32'd12) begin
            fi = 1'b1;
        end
        sel = 3'b11;
        a = b;
        b = 32'd1;
        #1;
        $display("Unsigned Subtract: a = %0d, b = %0d, out = %0d", a, $unsigned(b),
out_data);
        if (out_data != 32'd4294967293) begin
            fi = 1'b1;
        end
        sel = 3'b100;
        a = 32'd0;
        b = 32'd1;
        #1;
        $display("And: a = %0d, b = %0d, out = %0d", a, b, out_data);
        if (out_data != 32'd0) begin
            fi = 1'b1;
        end
        sel = 3'b101;
        #1;
        $display("Or: a = %0d, b = %0d, out = %0d", a, b, out_data);
        if (out_data != 32'd1) begin
            fi = 1'b1;
        end
        sel = 3'b110;
        a = 32'd4;
        a = -a;
        b = 32'd1;
        #1;
        $display("Shift Right Arithmetic: a = %0d, b = %0d, out = %0d", $signed(a),
b, out_data);
        if (out_data != 32'd2147483646) begin
            fi = 1'b1;
        end
        sel = 3'b111;
        a = 32'd4;
        b = 32'd1;
        #1;
        $display("Shift Right Logical: a = %0d, b = %0d, out = %0d", $signed(a), b,
out_data);
        if (out_data != 32'd2) begin
            fi = 1'b1;
        end

        if (fi == 1'b0) begin
            $display("Pass");
        end else begin
            $display("Fail");
        end
        $finish();
    end
endmodule
```