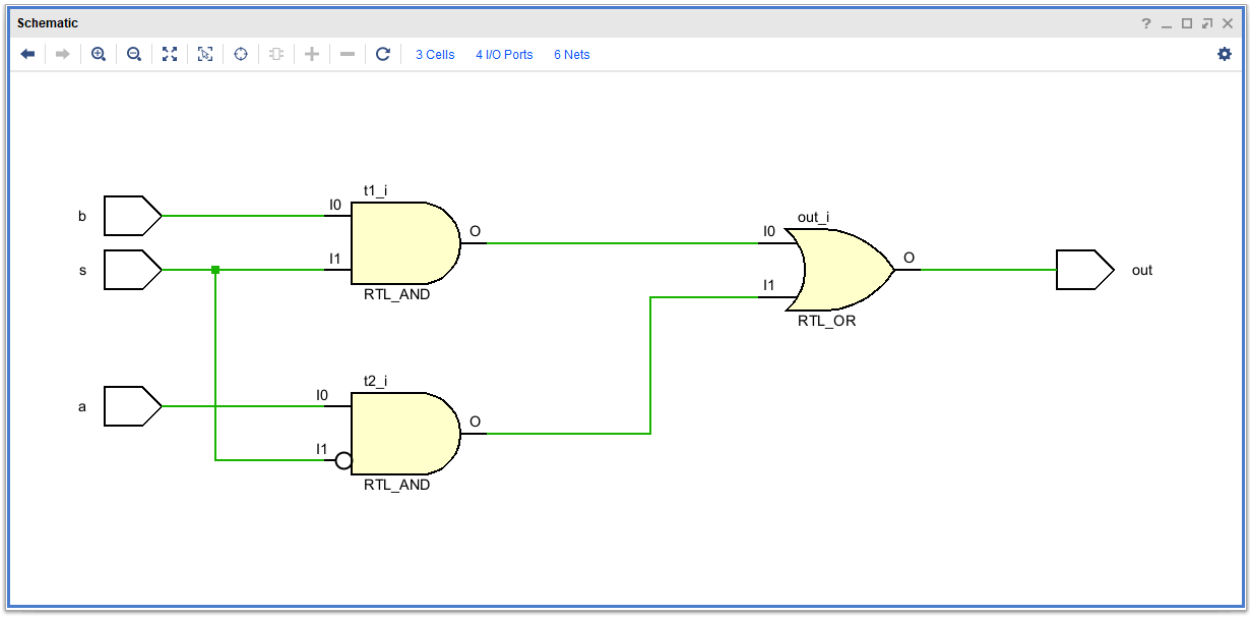
1.

The different methods seem to produce the same RTL schematic.

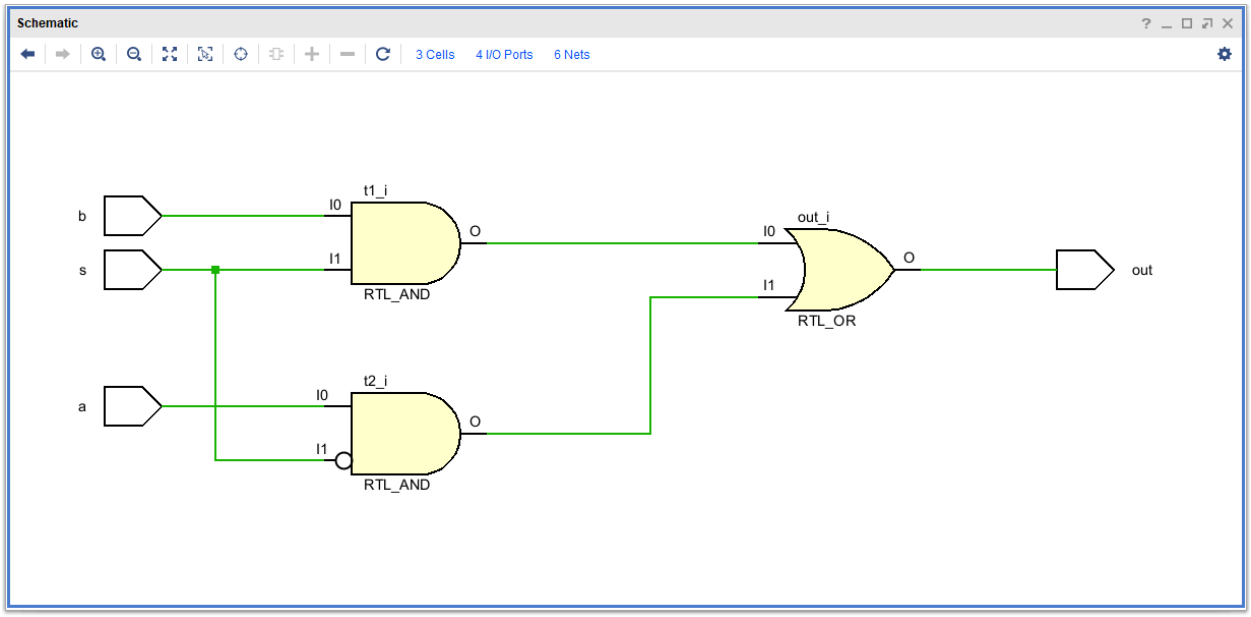
Gate-level:




```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: UNH ECE
// Engineer: Nick Snyder
//
// Create Date: 02/02/2023 11:23:31 PM
// Design Name: 2-1 mux
// Module Name: mux_gate
// Project Name: homework1
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module mux_gate(out, s, a, b);

    output out;
    input s, a, b;

    wire t1, t2, sbar;

    not(sbar, s);
    and(t1, b, s);
    and(t2, a, sbar);
```

```
    or(out, t1, t2);

endmodule
```

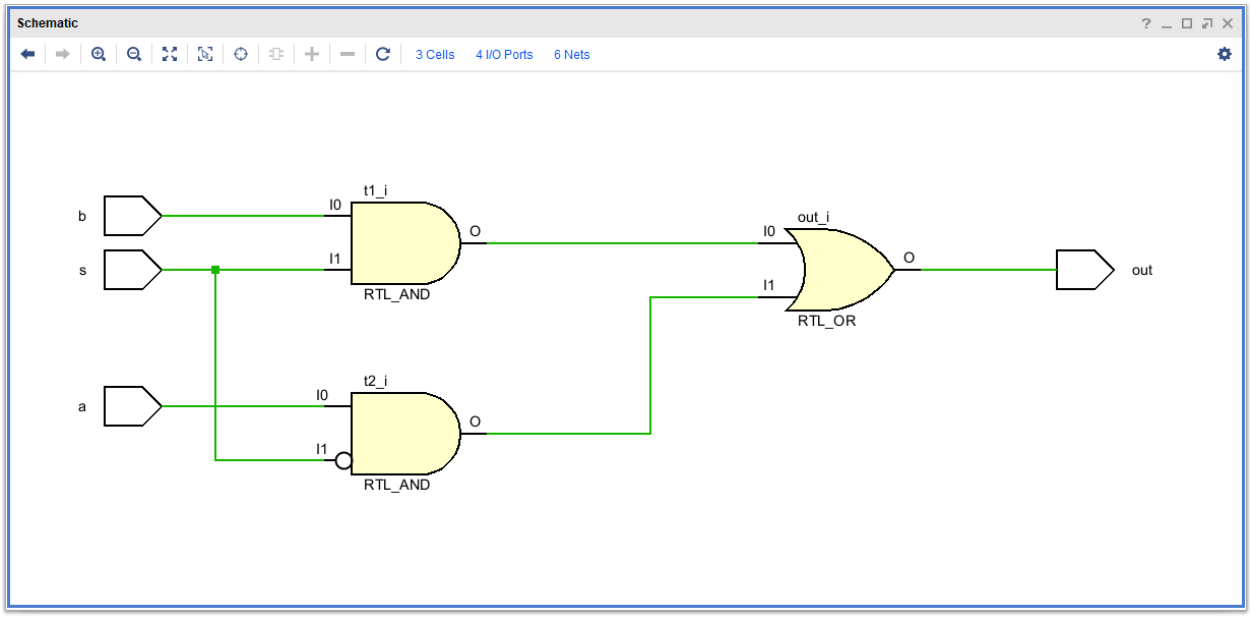
RTL-level:




```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: UNH ECE
// Engineer: Nick Snyder
//
// Create Date: 02/02/2023 11:23:31 PM
// Design Name: 2-1 mux
// Module Name: mux_RTL
// Project Name: homework1
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module mux_RTL(out, s, a, b);

    output out;
    input s, a, b;

    wire t1, t2, sbar;

    assign sbar = ~s;
    assign t1 = b & s;
```

```
    assign t2 = a & sbar;
    assign out = t1 | t2;

endmodule
```

Behavioral-level:

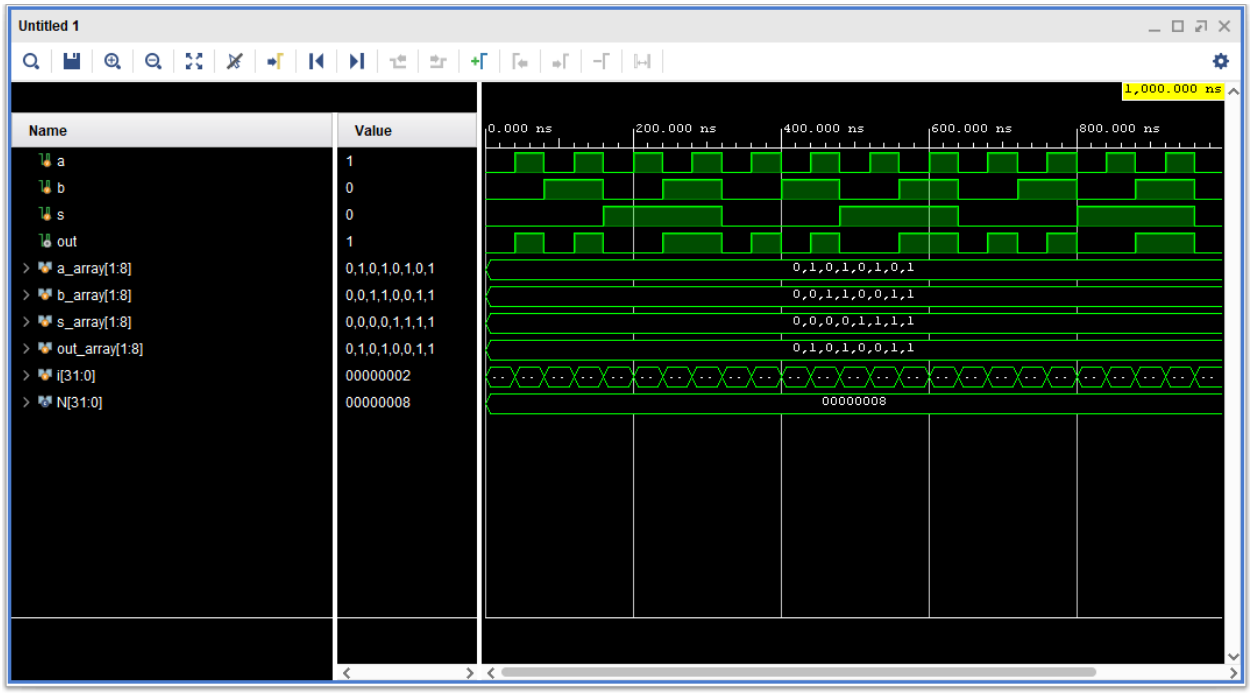



```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: UNH ECE
// Engineer: Nick Snyder
//
// Create Date: 02/02/2023 11:23:31 PM
// Design Name: 2-1 mux
// Module Name: mux_behavioral
// Project Name: homework1
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module mux_behavioral(out, s, a, b);

    output reg out;
    input s, a, b;

    always @(*)begin
    if (s)
        out = b;
```

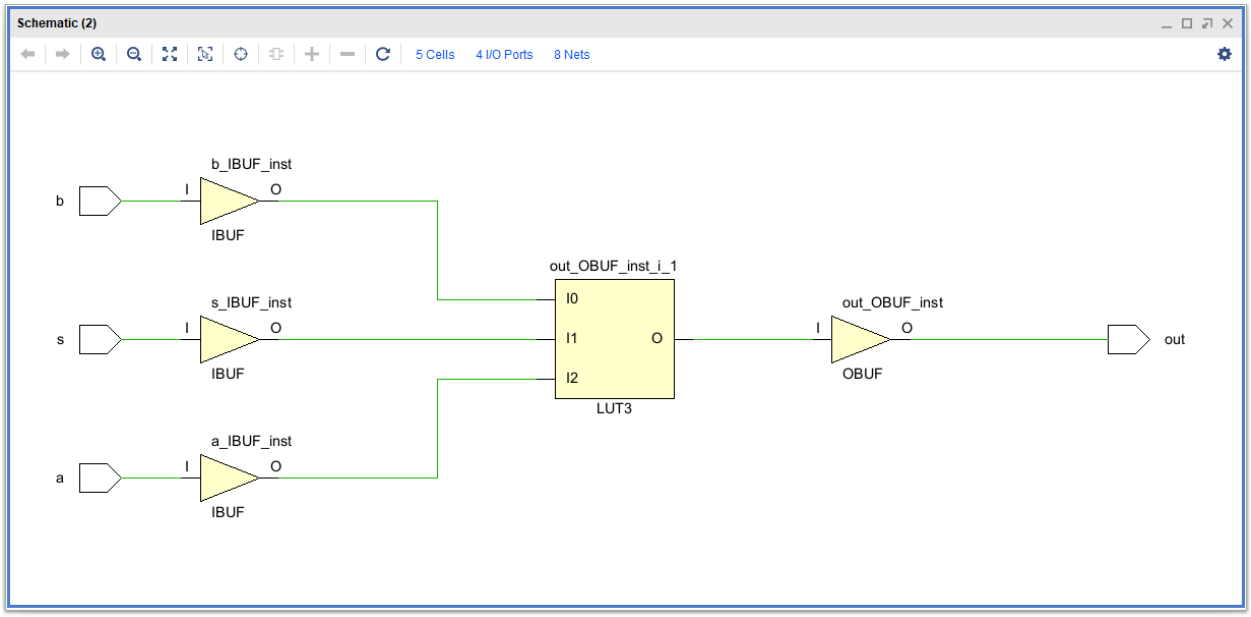
```
        else
            out = a;
        end

    endmodule
```

2.

<1>



<2>

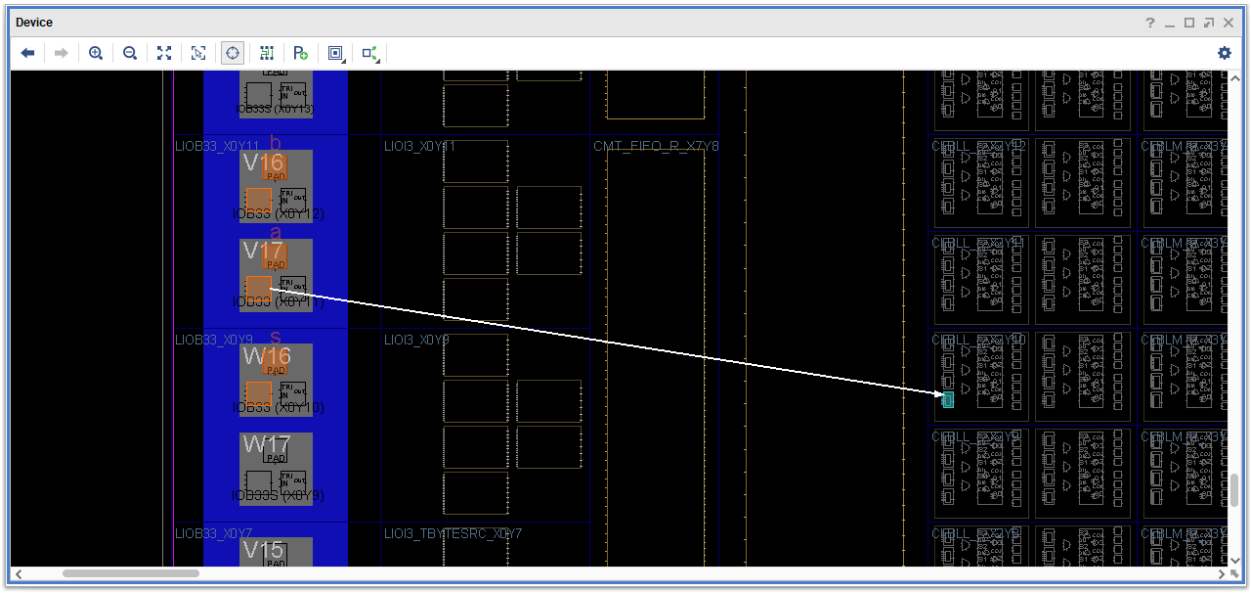


<3>

```
        set_property PACKAGE_PIN U16 [get_ports out]
        set_property PACKAGE_PIN V17 [get_ports a]
        set_property PACKAGE_PIN V16 [get_ports b]
```

```
set_property PACKAGE_PIN W16 [get_ports s]
set_property IOSTANDARD LVCMOS33 [get_ports out]
set_property IOSTANDARD LVCMOS33 [get_ports s]
set_property IOSTANDARD LVCMOS33 [get_ports b]
set_property IOSTANDARD LVCMOS33 [get_ports a]
```

<4>



Design implemented on hardware works as intended.