Course Code: CS223

Course Name: Digital Design **Section:** 4

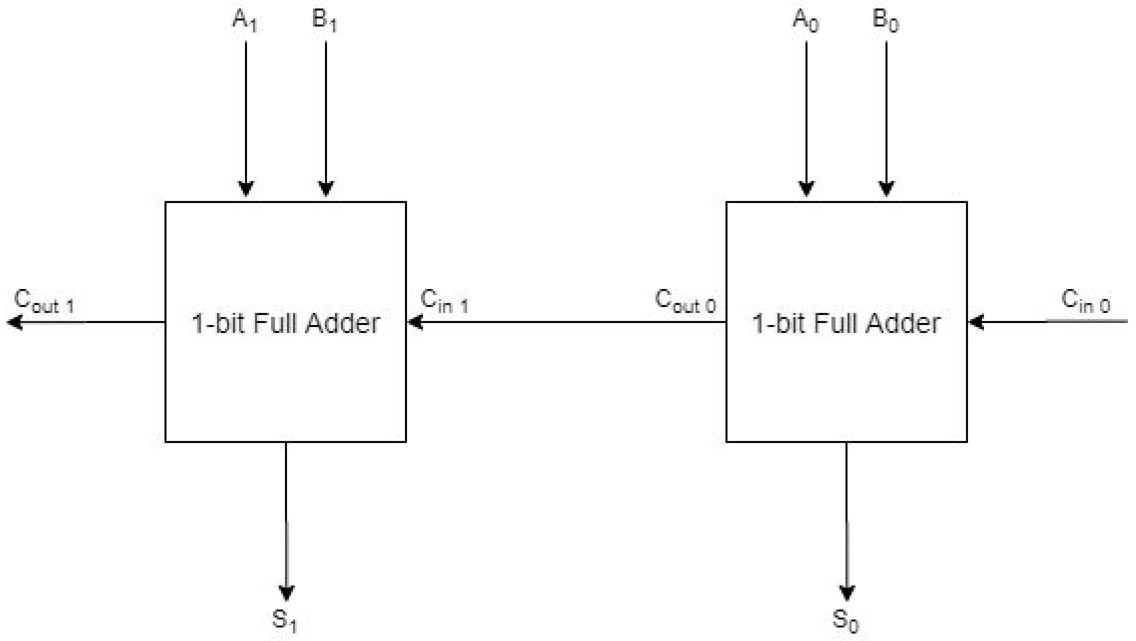
Lab 2

Name: Berk Saltuk Yılmaz **ID:** 21903419

Date: 16.10.2020

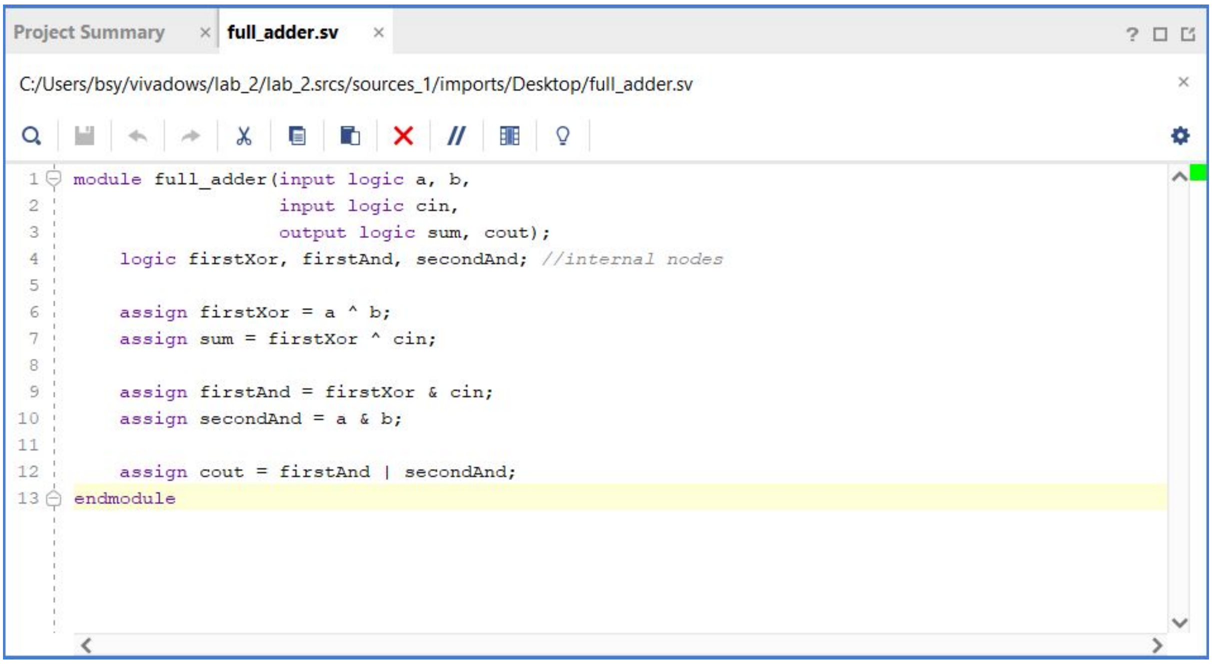
Part (b):

Circuit Schematic for 2-bit adder made from full adders:



Part (c):

Behavioral SystemVerilog module for the full adder:

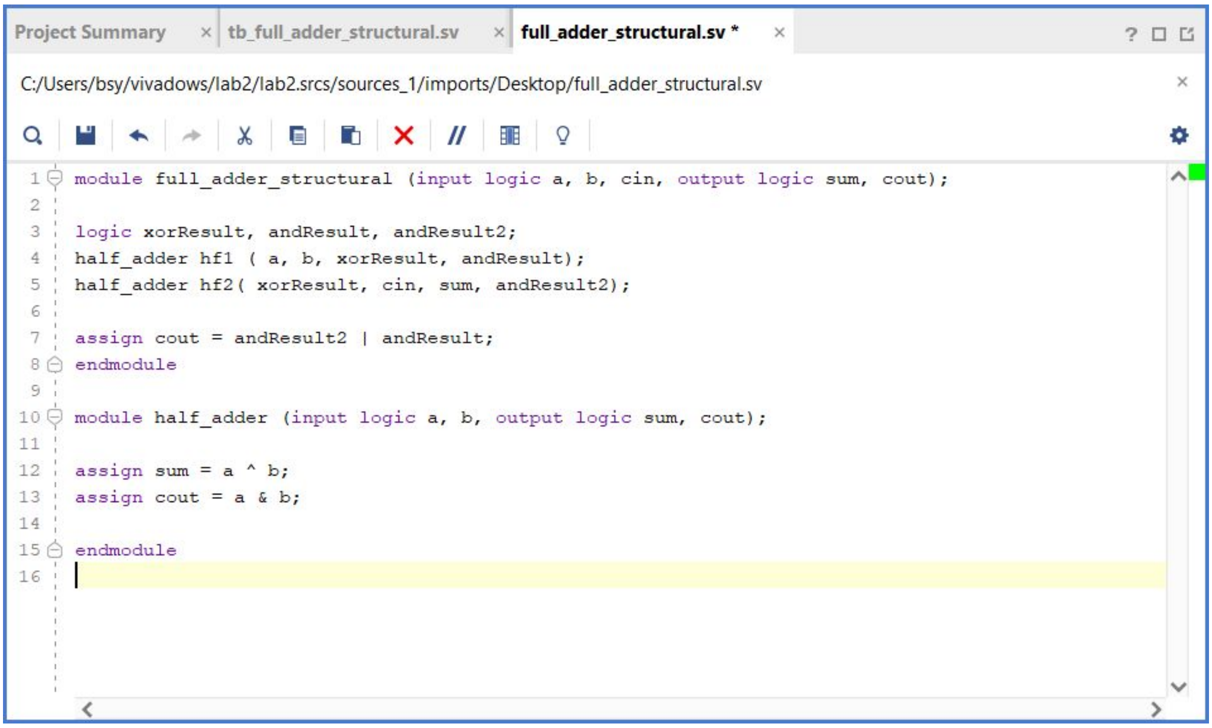


C:/Users/bsy/vivadows/lab_2/lab_2.srcs/sources_1/imports/Desktop/full_adder.sv

```
module full_adder(input logic a, b,
                  input logic cin,
                  output logic sum, cout);
    logic firstXor, firstAnd, secondAnd; //internal nodes

    assign firstXor = a ^ b;
    assign sum = firstXor ^ cin;

    assign firstAnd = firstXor & cin;
    assign secondAnd = a & b;

    assign cout = firstAnd | secondAnd;
endmodule
```

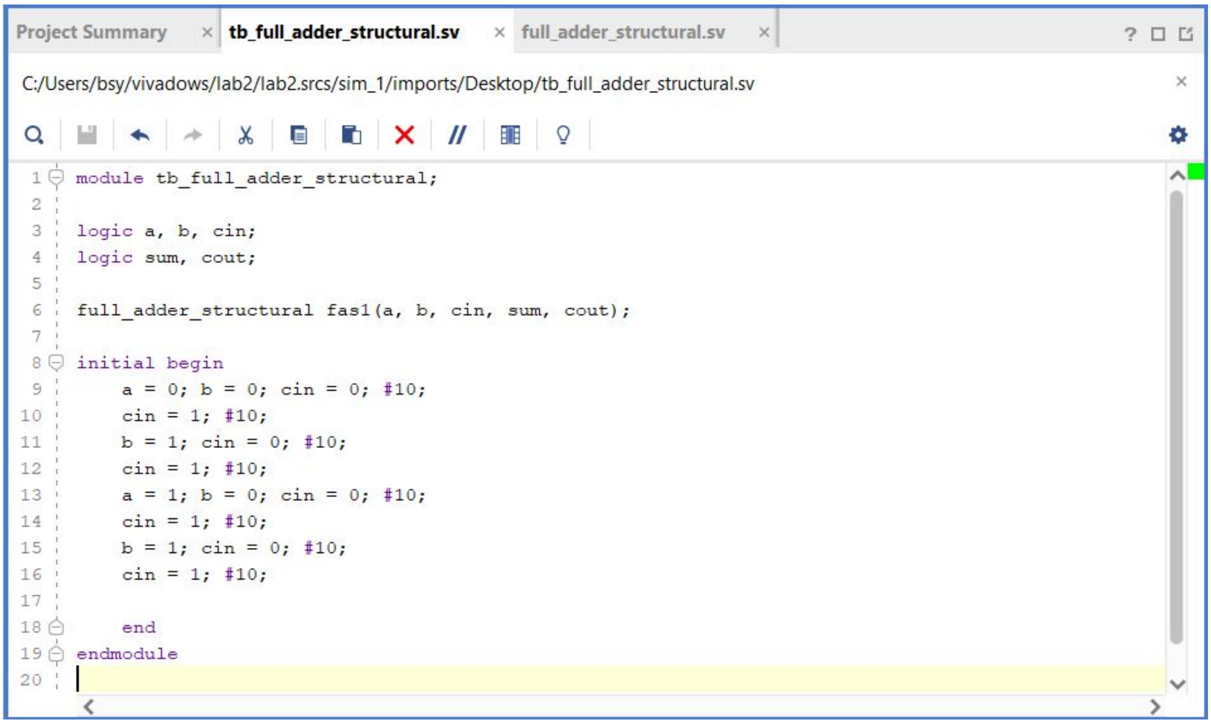
Part (d):

Structural SystemVerilog module for the full adder:


Project Summary
tb_full_adder_structural.sv
full_adder_structural.sv *
C:/Users/bsy/vivadows/lab2/lab2.srcs/sources_1/imports/Desktop/full_adder_structural.sv

```
module full_adder_structural (input logic a, b, cin, output logic sum, cout);

logic xorResult, andResult, andResult2;
half_adder hf1 ( a, b, xorResult, andResult);
half_adder hf2( xorResult, cin, sum, andResult2);

assign cout = andResult2 | andResult;
endmodule

module half_adder (input logic a, b, output logic sum, cout);

assign sum = a ^ b;
assign cout = a & b;

endmodule

```

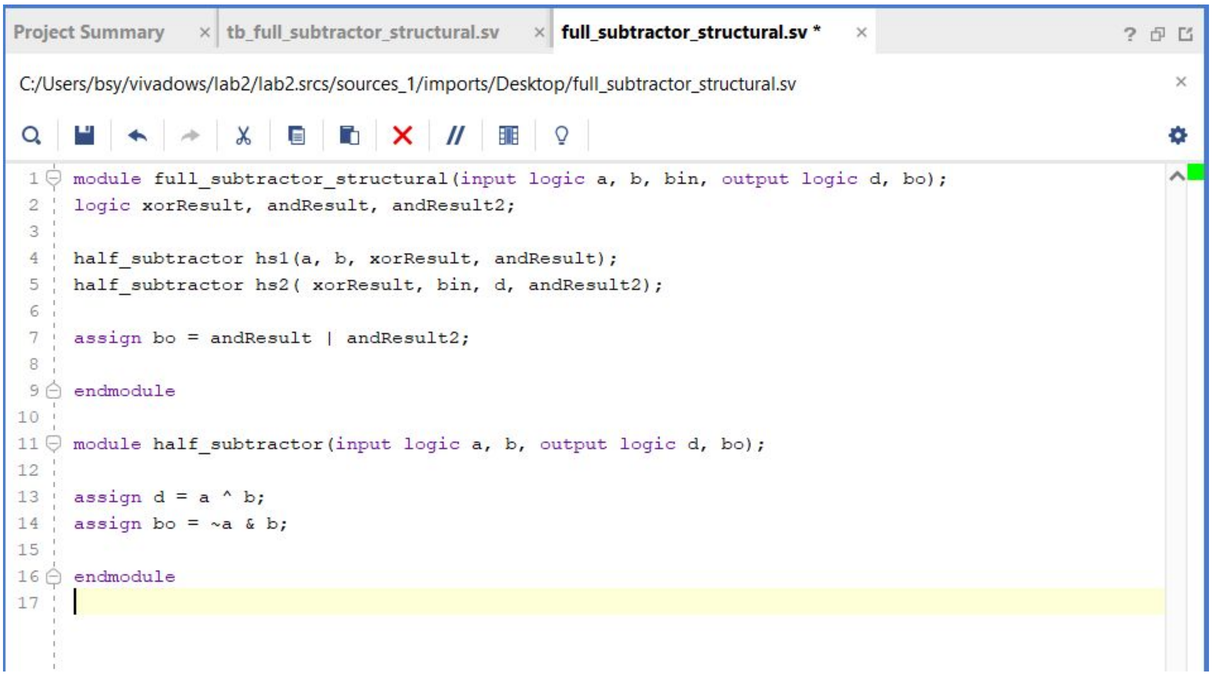
Testbench for structural SystemVerilog module for the full adder:


Project Summary
tb_full_adder_structural.sv
full_adder_structural.sv
C:/Users/bsy/vivadows/lab2/lab2.srcs/sim_1/imports/Desktop/tb_full_adder_structural.sv

```
module tb_full_adder_structural;

logic a, b, cin;
logic sum, cout;

full_adder_structural fas1(a, b, cin, sum, cout);

initial begin
    a = 0; b = 0; cin = 0; #10;
    cin = 1; #10;
    b = 1; cin = 0; #10;
    cin = 1; #10;
    a = 1; b = 0; cin = 0; #10;
    cin = 1; #10;
    b = 1; cin = 0; #10;
    cin = 1; #10;

    end
endmodule

```

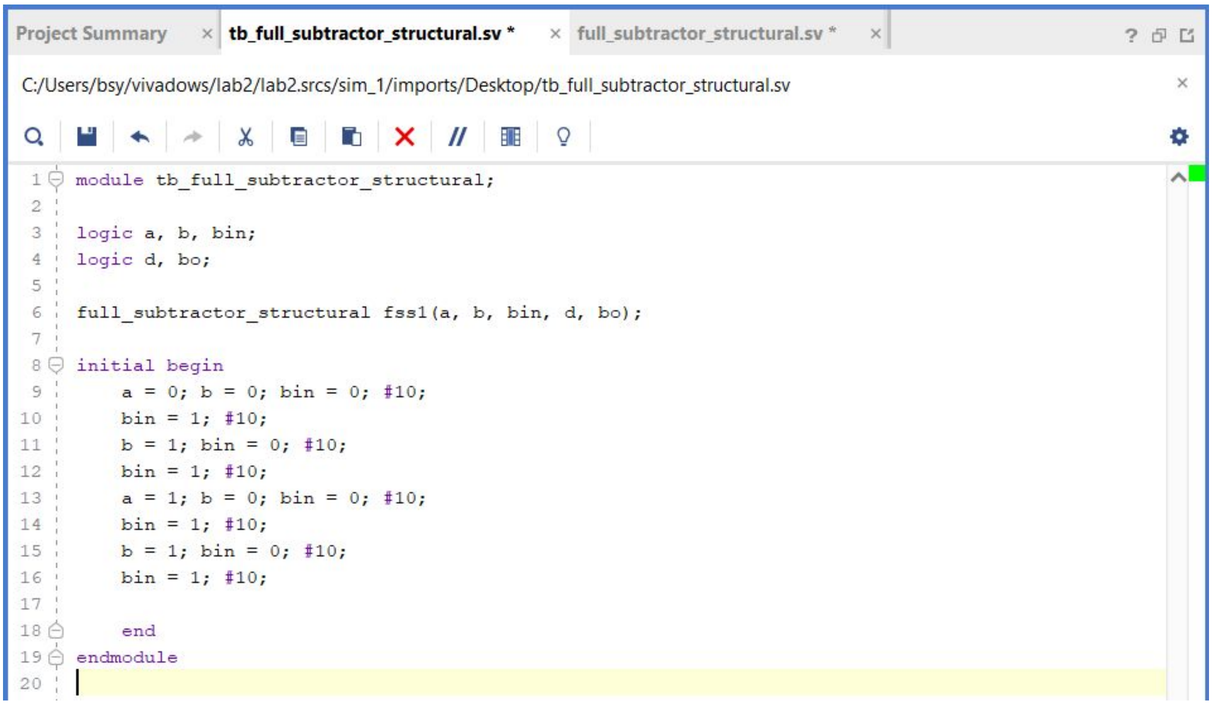
Part (e):

Structural SystemVerilog module for the full subtractor:


Project Summary
tb_full_subtractor_structural.sv
full_subtractor_structural.sv *
C:/Users/bsy/vivadows/lab2/lab2.srcs/sources_1/imports/Desktop/full_subtractor_structural.sv

```
module full_subtractor_structural(input logic a, b, bin, output logic d, bo);
logic xorResult, andResult, andResult2;

half_subtractor hs1(a, b, xorResult, andResult);
half_subtractor hs2( xorResult, bin, d, andResult2);

assign bo = andResult | andResult2;

endmodule

module half_subtractor(input logic a, b, output logic d, bo);

assign d = a ^ b;
assign bo = ~a & b;

endmodule

```

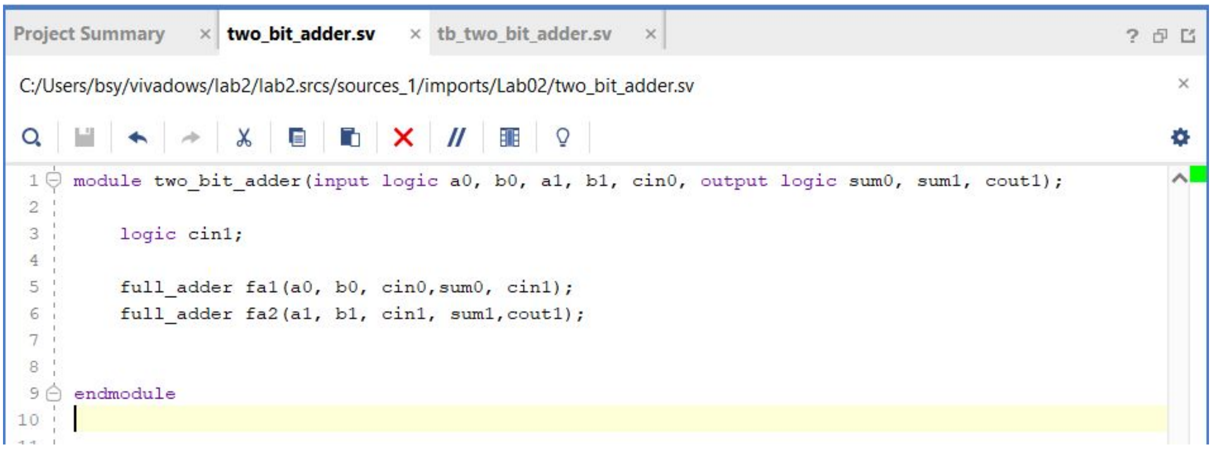
Testbench for Structural SystemVerilog module for the full subtractor:


Project Summary
tb_full_subtractor_structural.sv *
full_subtractor_structural.sv *
C:/Users/bsy/vivadows/lab2/lab2.srcs/sim_1/imports/Desktop/tb_full_subtractor_structural.sv

```
module tb_full_subtractor_structural;

logic a, b, bin;
logic d, bo;

full_subtractor_structural fss1(a, b, bin, d, bo);

initial begin
    a = 0; b = 0; bin = 0; #10;
    bin = 1; #10;
    b = 1; bin = 0; #10;
    bin = 1; #10;
    a = 1; b = 0; bin = 0; #10;
    bin = 1; #10;
    b = 1; bin = 0; #10;
    bin = 1; #10;

    end
endmodule

```

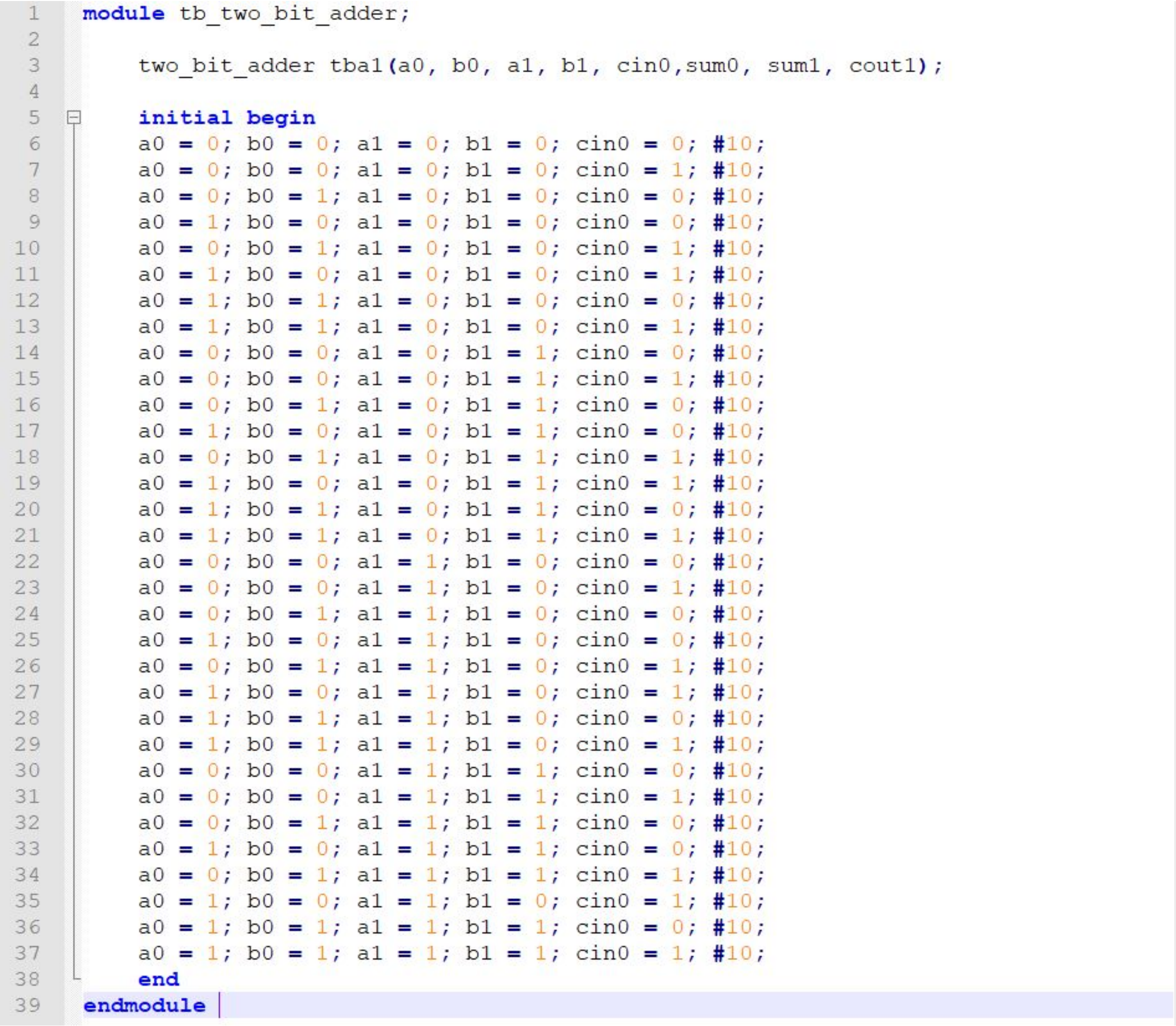
Part (f):

Structural SystemVerilog module for the 2-bit adder:


Project Summary
two_bit_adder.sv
tb_two_bit_adder.sv
C:/Users/bsy/vivadows/lab2/lab2.srcs/sources_1/imports/Lab02/two_bit_adder.sv

```
module two_bit_adder(input logic a0, b0, a1, b1, cin0, output logic sum0, sum1, cout1);

    logic cin1;

    full_adder fa1(a0, b0, cin0,sum0, cin1);
    full_adder fa2(a1, b1, cin1, sum1,cout1);


endmodule

```

Testbench for Structural SystemVerilog module for the 2-bit adder:



```
module tb_two_bit_adder;

    two_bit_adder tba1(a0, b0, a1, b1, cin0,sum0, sum1, cout1);

    initial begin
    a0 = 0; b0 = 0; a1 = 0; b1 = 0; cin0 = 0; #10;
    a0 = 0; b0 = 0; a1 = 0; b1 = 0; cin0 = 1; #10;
    a0 = 0; b0 = 1; a1 = 0; b1 = 0; cin0 = 0; #10;
    a0 = 1; b0 = 0; a1 = 0; b1 = 0; cin0 = 0; #10;
    a0 = 0; b0 = 1; a1 = 0; b1 = 0; cin0 = 1; #10;
    a0 = 1; b0 = 0; a1 = 0; b1 = 0; cin0 = 1; #10;
    a0 = 1; b0 = 1; a1 = 0; b1 = 0; cin0 = 0; #10;
    a0 = 1; b0 = 1; a1 = 0; b1 = 0; cin0 = 1; #10;
    a0 = 0; b0 = 0; a1 = 0; b1 = 1; cin0 = 0; #10;
    a0 = 0; b0 = 0; a1 = 0; b1 = 1; cin0 = 1; #10;
    a0 = 0; b0 = 1; a1 = 0; b1 = 1; cin0 = 0; #10;
    a0 = 1; b0 = 0; a1 = 0; b1 = 1; cin0 = 0; #10;
    a0 = 0; b0 = 1; a1 = 0; b1 = 1; cin0 = 1; #10;
    a0 = 1; b0 = 0; a1 = 0; b1 = 1; cin0 = 1; #10;
    a0 = 1; b0 = 1; a1 = 0; b1 = 1; cin0 = 0; #10;
    a0 = 1; b0 = 1; a1 = 0; b1 = 1; cin0 = 1; #10;
    a0 = 0; b0 = 0; a1 = 1; b1 = 0; cin0 = 0; #10;
    a0 = 0; b0 = 0; a1 = 1; b1 = 0; cin0 = 1; #10;
    a0 = 0; b0 = 1; a1 = 1; b1 = 0; cin0 = 0; #10;
    a0 = 1; b0 = 0; a1 = 1; b1 = 0; cin0 = 0; #10;
    a0 = 0; b0 = 1; a1 = 1; b1 = 0; cin0 = 1; #10;
    a0 = 1; b0 = 0; a1 = 1; b1 = 0; cin0 = 1; #10;
    a0 = 1; b0 = 1; a1 = 1; b1 = 0; cin0 = 0; #10;
    a0 = 1; b0 = 1; a1 = 1; b1 = 0; cin0 = 1; #10;
    a0 = 0; b0 = 0; a1 = 1; b1 = 1; cin0 = 0; #10;
    a0 = 0; b0 = 0; a1 = 1; b1 = 1; cin0 = 1; #10;
    a0 = 0; b0 = 1; a1 = 1; b1 = 1; cin0 = 0; #10;
    a0 = 1; b0 = 0; a1 = 1; b1 = 1; cin0 = 0; #10;
    a0 = 0; b0 = 1; a1 = 1; b1 = 1; cin0 = 1; #10;
    a0 = 1; b0 = 0; a1 = 1; b1 = 0; cin0 = 1; #10;
    a0 = 1; b0 = 1; a1 = 1; b1 = 1; cin0 = 0; #10;
    a0 = 1; b0 = 1; a1 = 1; b1 = 1; cin0 = 1; #10;
    end
endmodule
```