

**Course Code: CS223**

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

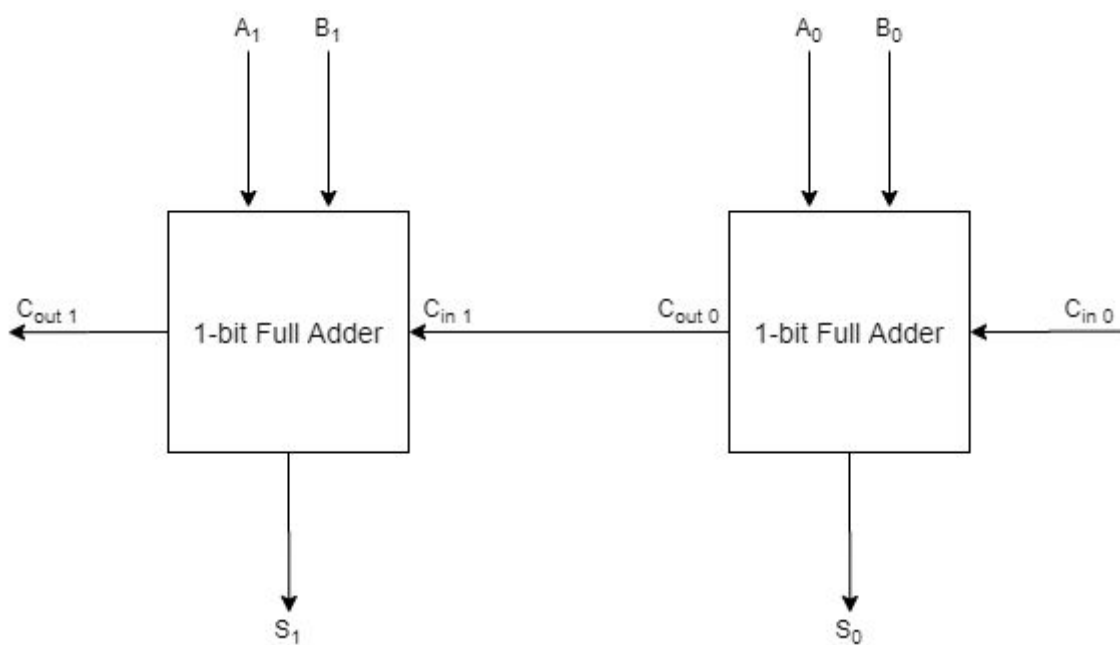
**Lab 2**

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

**Date: 16.10.2020**

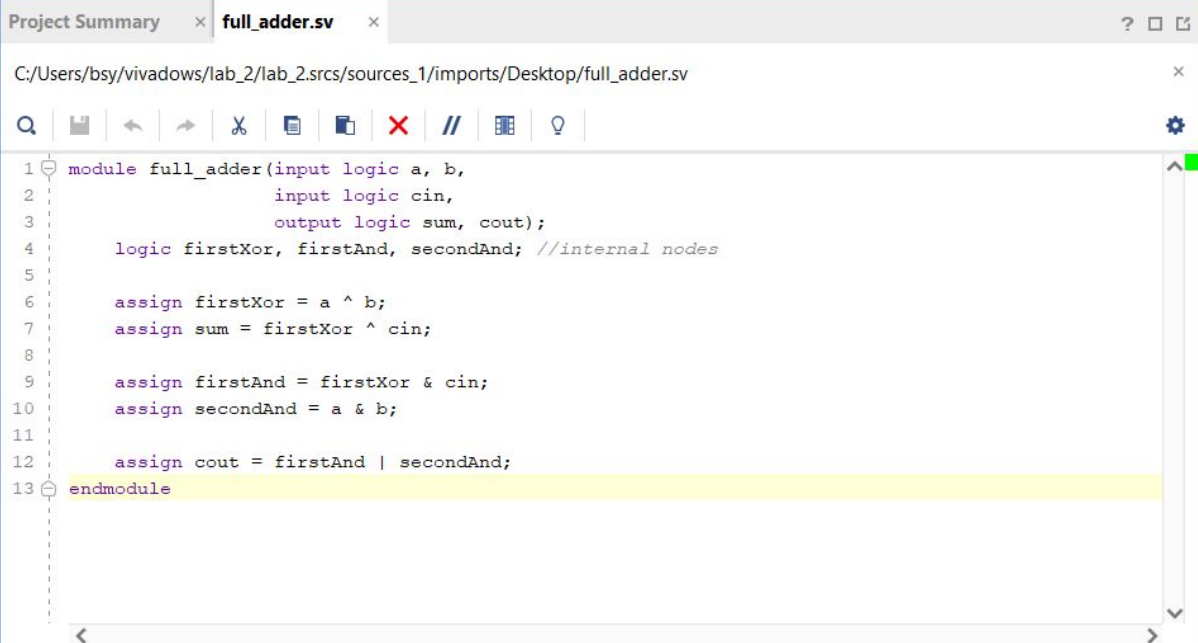
**Part (b):**

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



## Part (c):

Behavioral SystemVerilog module for the full adder:

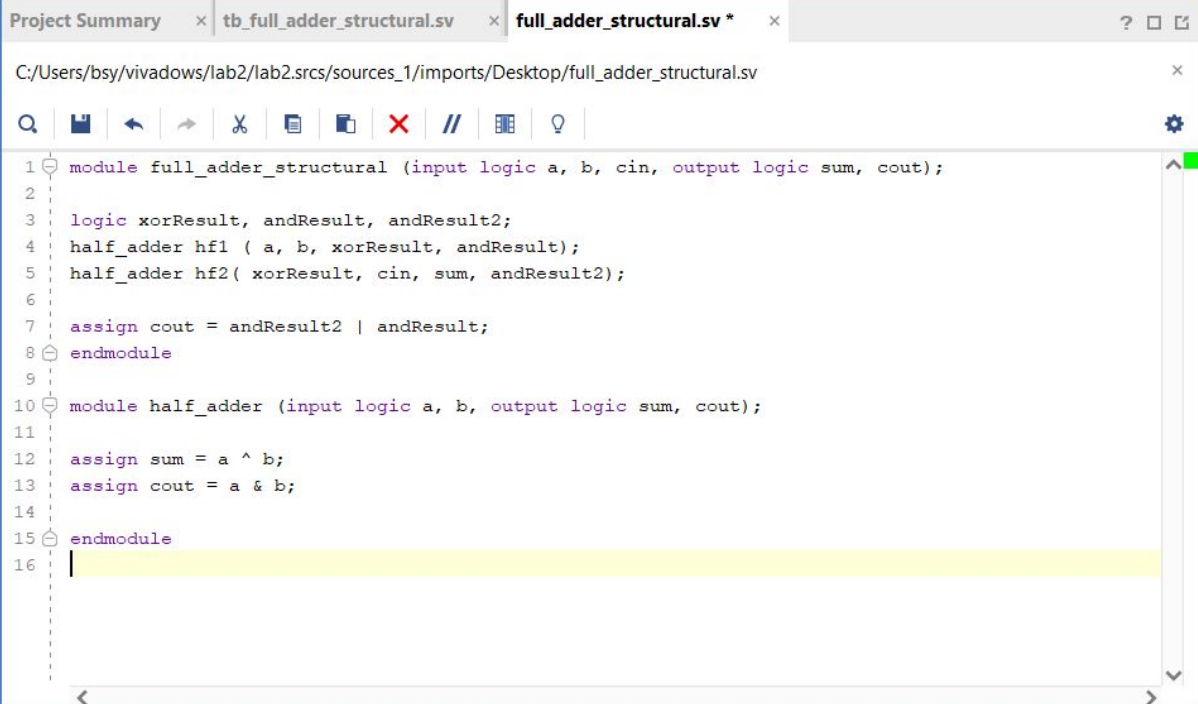


The screenshot shows a code editor window with a tab labeled 'full\_adder.sv'. The file path is 'C:/Users/bsy/vivadovs/lab\_2/lab\_2.srscs/sources\_1/imports/Desktop/full\_adder.sv'. The code is as follows:

```
1 module full_adder(input logic a, b,  
2                   input logic cin,  
3                   output logic sum, cout);  
4     logic firstXor, firstAnd, secondAnd; //internal nodes  
5  
6     assign firstXor = a ^ b;  
7     assign sum = firstXor ^ cin;  
8  
9     assign firstAnd = firstXor & cin;  
10    assign secondAnd = a & b;  
11  
12    assign cout = firstAnd | secondAnd;  
13 endmodule
```

## Part (d):

### Structural SystemVerilog module for the full adder:

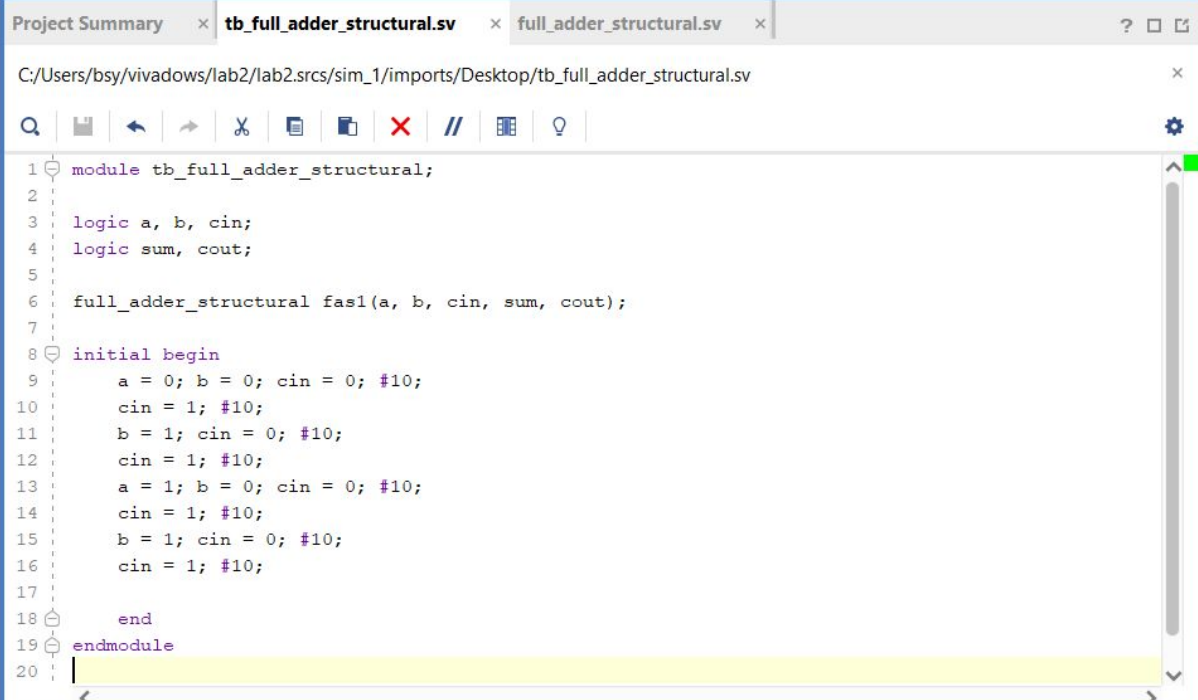


```

1 module full_adder_structural (input logic a, b, cin, output logic sum, cout);
2
3     logic xorResult, andResult, andResult2;
4     half_adder hf1 ( a, b, xorResult, andResult);
5     half_adder hf2( xorResult, cin, sum, andResult2);
6
7     assign cout = andResult2 | andResult;
8 endmodule
9
10 module half_adder (input logic a, b, output logic sum, cout);
11
12     assign sum = a ^ b;
13     assign cout = a & b;
14
15 endmodule
16

```

### Testbench for structural SystemVerilog module for the full adder:



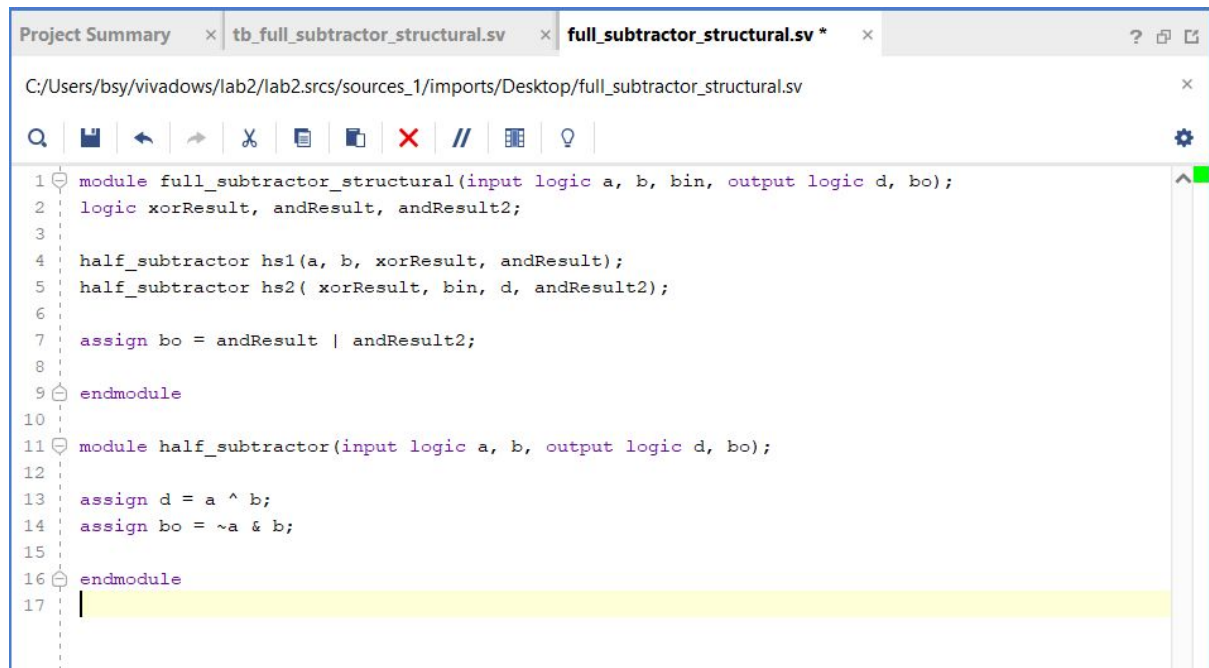
```

1 module tb_full_adder_structural;
2
3     logic a, b, cin;
4     logic sum, cout;
5
6     full_adder_structural fas1(a, b, cin, sum, cout);
7
8     initial begin
9         a = 0; b = 0; cin = 0; #10;
10        cin = 1; #10;
11        b = 1; cin = 0; #10;
12        cin = 1; #10;
13        a = 1; b = 0; cin = 0; #10;
14        cin = 1; #10;
15        b = 1; cin = 0; #10;
16        cin = 1; #10;
17
18    end
19 endmodule
20

```

## Part (e):

### Structural SystemVerilog module for the full subtractor:



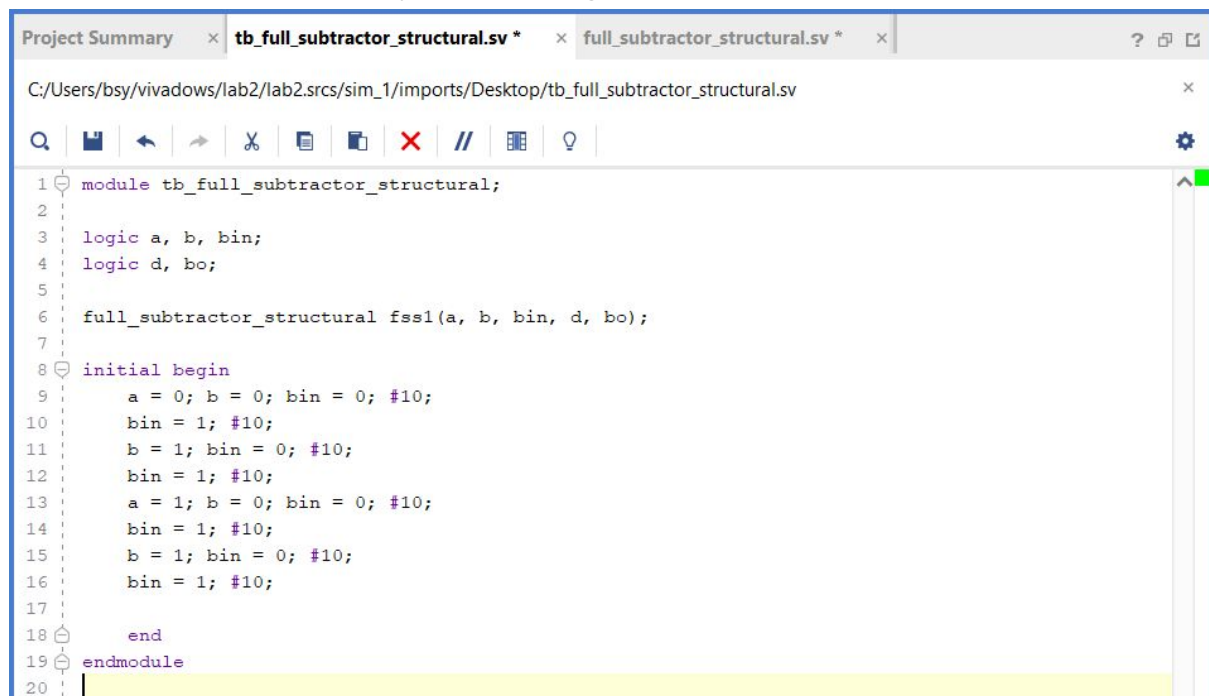
```

Project Summary x tb_full_subtractor_structural.sv x full_subtractor_structural.sv * x
C:/Users/bsy/vivadown/lab2/lab2.srscs/sources_1/imports/Desktop/full_subtractor_structural.sv

1 module full_subtractor_structural(input logic a, b, bin, output logic d, bo);
2   logic xorResult, andResult, andResult2;
3
4   half_subtractor hsl(a, b, xorResult, andResult);
5   half_subtractor hs2(xorResult, bin, d, andResult2);
6
7   assign bo = andResult | andResult2;
8
9 endmodule
10
11 module half_subtractor(input logic a, b, output logic d, bo);
12
13   assign d = a ^ b;
14   assign bo = ~a & b;
15
16 endmodule
17

```

### Testbench for Structural SystemVerilog module for the full subtractor:



```


Project Summary x tb_full_subtractor_structural.sv * x full_subtractor_structural.sv * x
C:/Users/bsy/vivadown/lab2/lab2.srscs/sim_1/imports/Desktop/tb_full_subtractor_structural.sv

1 module tb_full_subtractor_structural;
2
3   logic a, b, bin;
4   logic d, bo;
5
6   full_subtractor_structural fss1(a, b, bin, d, bo);
7
8   initial begin
9     a = 0; b = 0; bin = 0; #10;
10    bin = 1; #10;
11    b = 1; bin = 0; #10;
12    bin = 1; #10;
13    a = 1; b = 0; bin = 0; #10;
14    bin = 1; #10;
15    b = 1; bin = 0; #10;
16    bin = 1; #10;
17
18   end
19 endmodule
20

```

## Part (f):

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

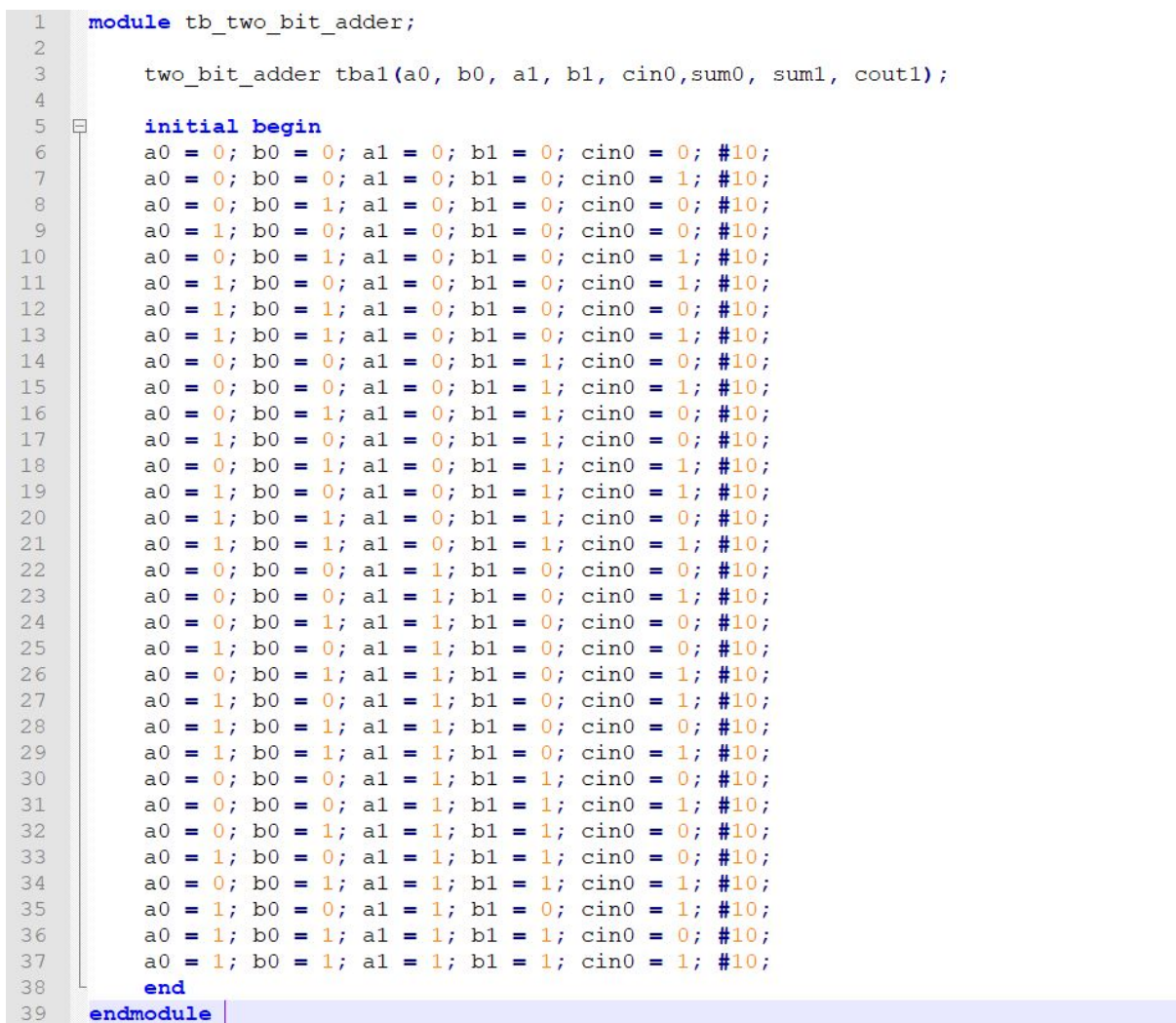


```

1 module two_bit_adder(input logic a0, b0, a1, b1, cin0, output logic sum0, sum1, cout1);
2
3     logic cin1;
4
5     full_adder fa1(a0, b0, cin0, sum0, cin1);
6     full_adder fa2(a1, b1, cin1, sum1, cout1);
7
8
9 endmodule
10

```

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



```

1 module tb_two_bit_adder;
2
3     two_bit_adder tba1(a0, b0, a1, b1, cin0, sum0, sum1, cout1);
4
5     initial begin
6         a0 = 0; b0 = 0; a1 = 0; b1 = 0; cin0 = 0; #10;
7         a0 = 0; b0 = 0; a1 = 0; b1 = 0; cin0 = 1; #10;
8         a0 = 0; b0 = 1; a1 = 0; b1 = 0; cin0 = 0; #10;
9         a0 = 1; b0 = 0; a1 = 0; b1 = 0; cin0 = 0; #10;
10        a0 = 0; b0 = 1; a1 = 0; b1 = 0; cin0 = 1; #10;
11        a0 = 1; b0 = 0; a1 = 0; b1 = 0; cin0 = 1; #10;
12        a0 = 0; b0 = 1; a1 = 0; b1 = 0; cin0 = 0; #10;
13        a0 = 1; b0 = 1; a1 = 0; b1 = 0; cin0 = 1; #10;
14        a0 = 0; b0 = 0; a1 = 0; b1 = 1; cin0 = 0; #10;
15        a0 = 0; b0 = 0; a1 = 0; b1 = 1; cin0 = 1; #10;
16        a0 = 0; b0 = 1; a1 = 0; b1 = 1; cin0 = 0; #10;
17        a0 = 1; b0 = 0; a1 = 0; b1 = 1; cin0 = 0; #10;
18        a0 = 0; b0 = 1; a1 = 0; b1 = 1; cin0 = 1; #10;
19        a0 = 1; b0 = 0; a1 = 0; b1 = 1; cin0 = 1; #10;
20        a0 = 1; b0 = 1; a1 = 0; b1 = 1; cin0 = 0; #10;
21        a0 = 1; b0 = 1; a1 = 0; b1 = 1; cin0 = 1; #10;
22        a0 = 0; b0 = 0; a1 = 1; b1 = 0; cin0 = 0; #10;
23        a0 = 0; b0 = 0; a1 = 1; b1 = 0; cin0 = 1; #10;
24        a0 = 0; b0 = 1; a1 = 1; b1 = 0; cin0 = 0; #10;
25        a0 = 1; b0 = 0; a1 = 1; b1 = 0; cin0 = 0; #10;
26        a0 = 0; b0 = 1; a1 = 1; b1 = 0; cin0 = 1; #10;
27        a0 = 1; b0 = 0; a1 = 1; b1 = 0; cin0 = 1; #10;
28        a0 = 1; b0 = 1; a1 = 1; b1 = 0; cin0 = 0; #10;
29        a0 = 1; b0 = 1; a1 = 1; b1 = 0; cin0 = 1; #10;
30        a0 = 0; b0 = 0; a1 = 1; b1 = 1; cin0 = 0; #10;
31        a0 = 0; b0 = 0; a1 = 1; b1 = 1; cin0 = 1; #10;
32        a0 = 0; b0 = 1; a1 = 1; b1 = 1; cin0 = 0; #10;
33        a0 = 1; b0 = 0; a1 = 1; b1 = 1; cin0 = 0; #10;
34        a0 = 0; b0 = 1; a1 = 1; b1 = 1; cin0 = 1; #10;
35        a0 = 1; b0 = 0; a1 = 1; b1 = 0; cin0 = 1; #10;
36        a0 = 1; b0 = 1; a1 = 1; b1 = 1; cin0 = 0; #10;
37        a0 = 1; b0 = 1; a1 = 1; b1 = 1; cin0 = 1; #10;
38    end
39 endmodule

```