Lab 2. ALU

1. Objectives:

(1) Learn how to design an arithmetic logic unit (ALU) with Verilog.

2. Resources:

(1) Read the "ALU Introduction 1-3" in Canvas**.**

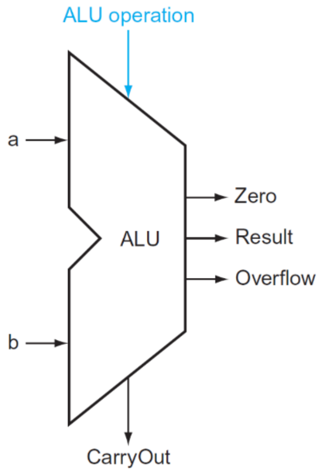
3. Requirements

(1) Design ALU with the **dataflow modeling method**.

(2) For each circuit design, please add your **design code**, **RTL schematic**, and **simulation results** to the report.

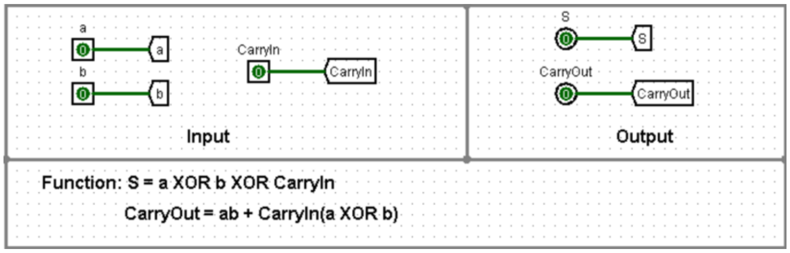
4. Tasks:

Design a 32-bit ALU that can realize the following functions:



ALU control lines	Function
0000	AND
0001	OR
0010	Add
0110	Subtract
0111	Set on less than
1100	NOR

4.1 Design a 1-bit full adder (5 points)



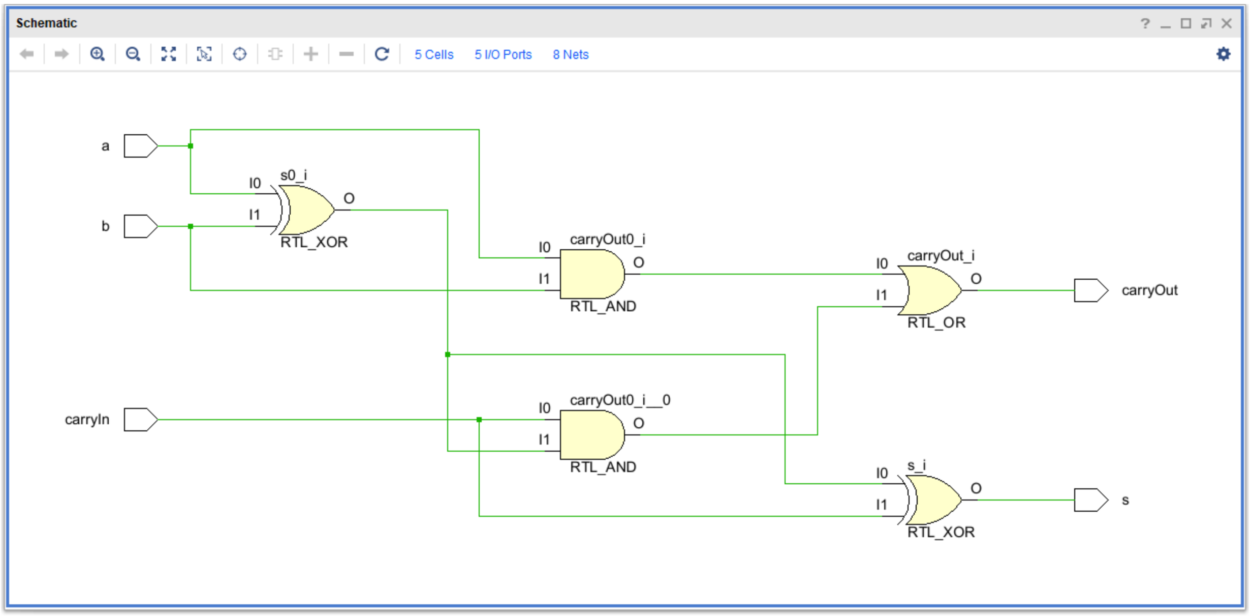
Function: S = a XOR b XOR CarryIn

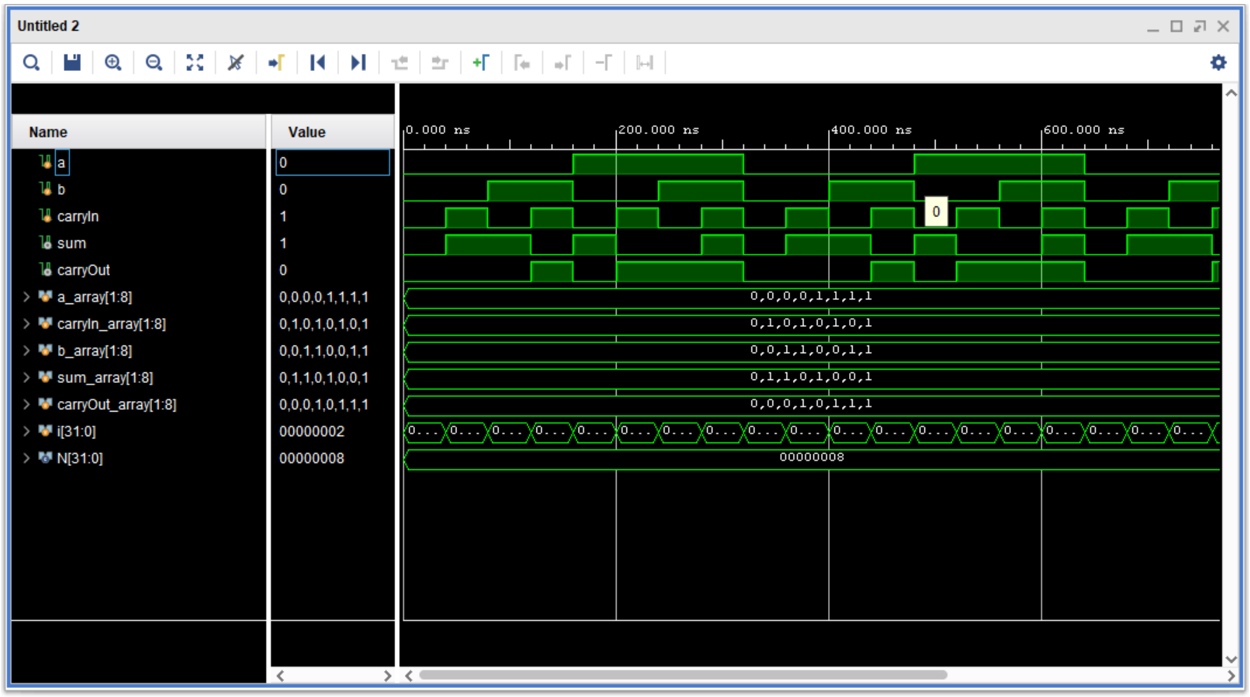
CarryOut = ab + CarryIn(a XOR b)

Verilog code, RTL schematic, and simulation result:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/21/2023 03:55:12 PM
// Design Name:
// Module Name: FullAdder_1b
// Project Name:
```

```
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module FullAdder_1b(s, carryOut, carryIn, a, b);

    output s, carryOut;
    input carryIn, a, b;

    assign s = a ^ b ^ carryIn;
    assign carryOut = a & b | carryIn & (a ^ b);

endmodule
```

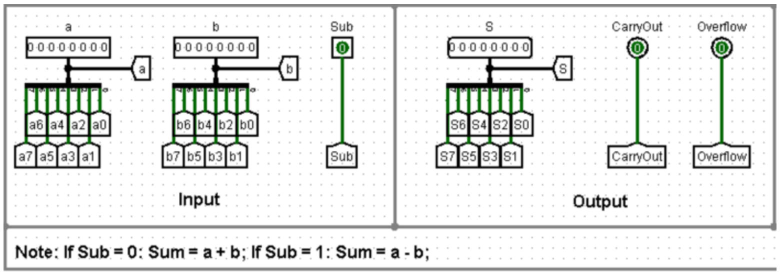




Name	Value
a	0
b	0
carryIn	1
sum	1
carryOut	0
a_array[1:8]	0,0,0,0,1,1,1,1
carryIn_array[1:8]	0,1,0,1,0,1,0,1
b_array[1:8]	0,0,1,1,0,0,1,1
sum_array[1:8]	0,1,1,0,1,0,0,1
carryOut_array[1:8]	0,0,0,1,0,1,1,1
i[31:0]	00000002
N[31:0]	00000008

4.2 Design a 8-bit Ripple Carry Adder and Subtracter (20 points)

Requirement: Only 1-bit full adders designed in 4.1, basic logic gates, and multiplexers (if needed) are allowed.

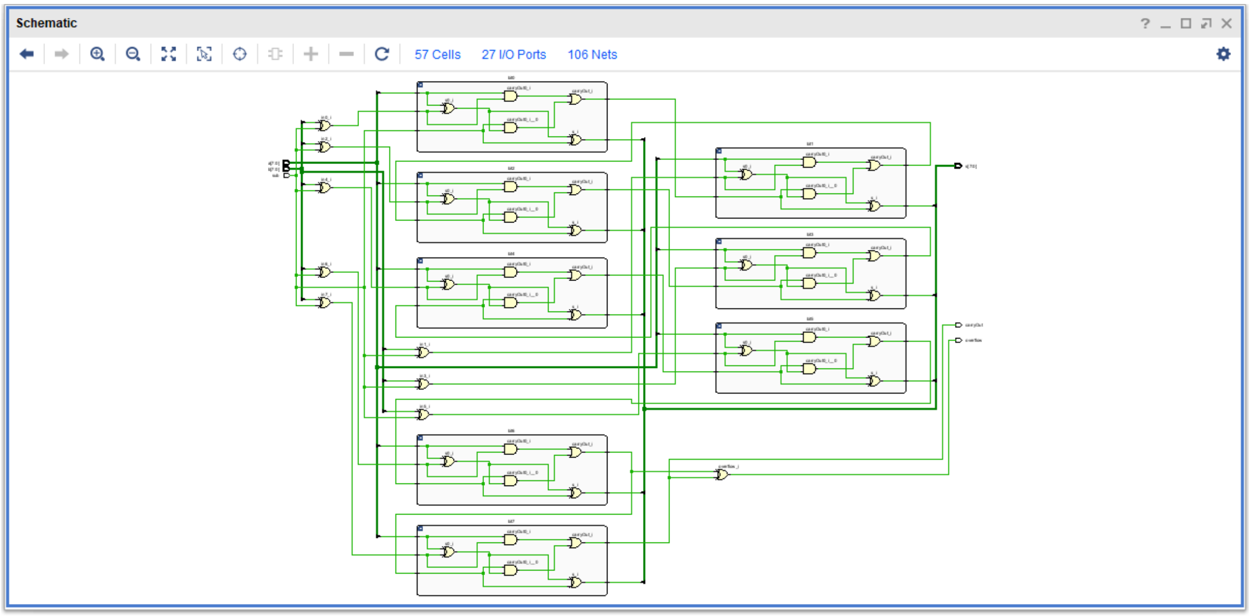


Note: If Sub = 0: Sum = a + b; If Sub = 1: Sum = a - b;

Verilog code, RTL schematic, and simulation result:

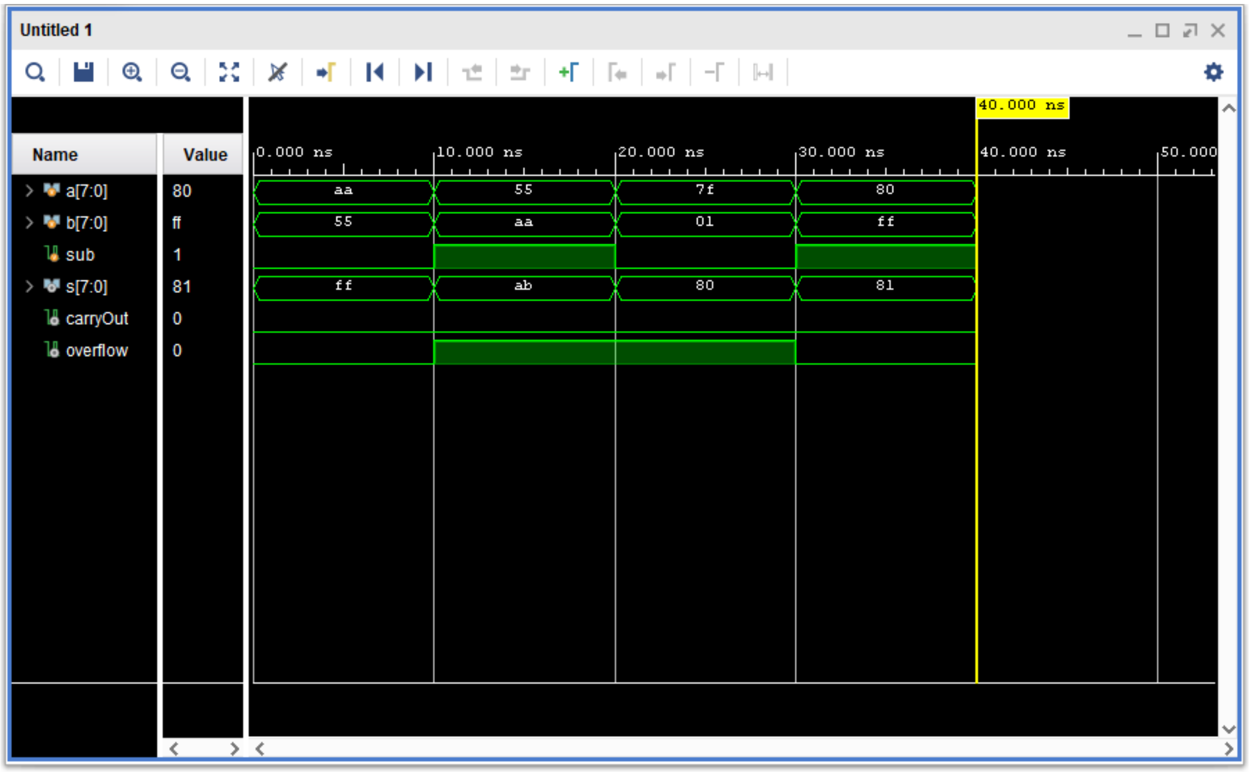
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/21/2023 03:55:12 PM
// Design Name:
// Module Name: RippleCarryAdderSubtractor_8b
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
```

```
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module RippleCarryAdderSubtractor_8b(s, carryOut, overflow, sub, a, b);

    output [7:0]s;
    output carryOut, overflow;
    input [7:0]a, b;
    input sub;

    wire in0, in1, in2, in3, in4, in5, in6, in7;
    wire carry0, carry1, carry2, carry3, carry4, carry5, carry6, carry7;

    assign in0 = b[0] ^ sub;
    assign in1 = b[1] ^ sub;
    assign in2 = b[2] ^ sub;
    assign in3 = b[3] ^ sub;
    assign in4 = b[4] ^ sub;
    assign in5 = b[5] ^ sub;
    assign in6 = b[6] ^ sub;
    assign in7 = b[7] ^ sub;

    assign carry0 = 1'b0 ^ sub;

    FullAdder_1b bit0(s[0], carry1, carry0, a[0], in0);
    FullAdder_1b bit1(s[1], carry2, carry1, a[1], in1);
    FullAdder_1b bit2(s[2], carry3, carry2, a[2], in2);
    FullAdder_1b bit3(s[3], carry4, carry3, a[3], in3);
    FullAdder_1b bit4(s[4], carry5, carry4, a[4], in4);
    FullAdder_1b bit5(s[5], carry6, carry5, a[5], in5);
    FullAdder_1b bit6(s[6], carry7, carry6, a[6], in6);
    FullAdder_1b bit7(s[7], carry8, carry7, a[7], in7);

    assign carryOut = carry8;
    assign overflow = carry7 ^ carry8;

endmodule
```



Schematic

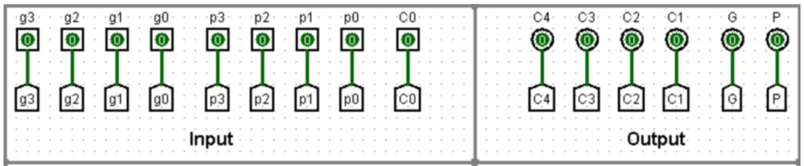
57 Cells 27 I/O Ports 106 Nets



Untitled 1

Name	Value
a[7:0]	80
b[7:0]	ff
sub	1
s[7:0]	81
carryOut	0
overflow	0

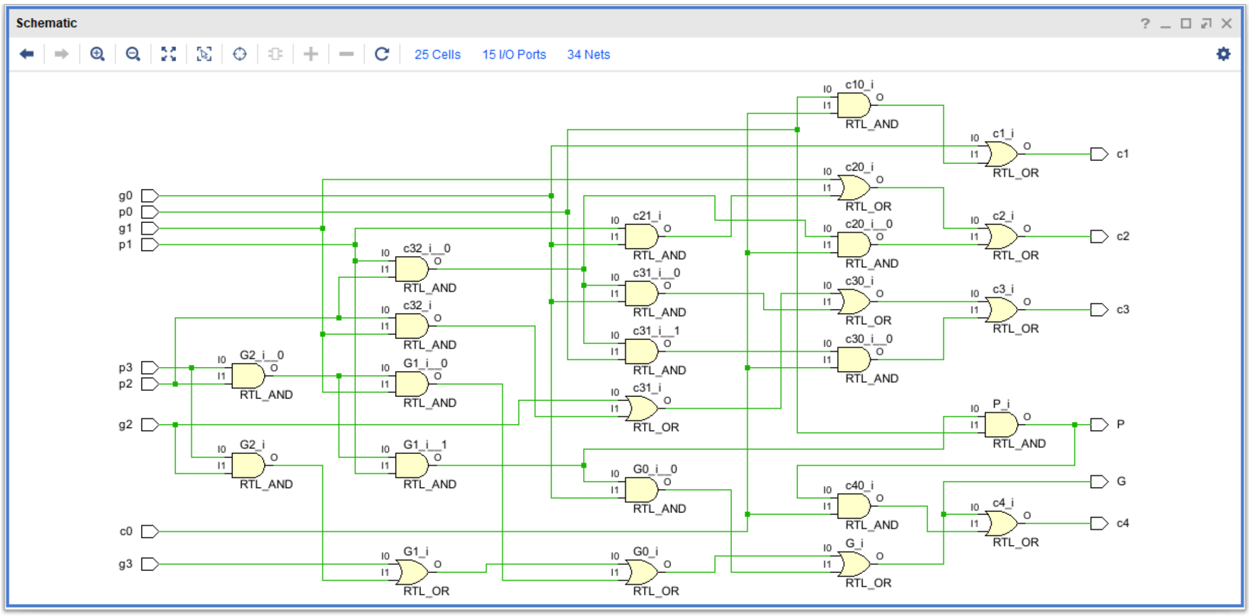
4.3 Design a 4-bit Carry Lookahead Unit (20 points)

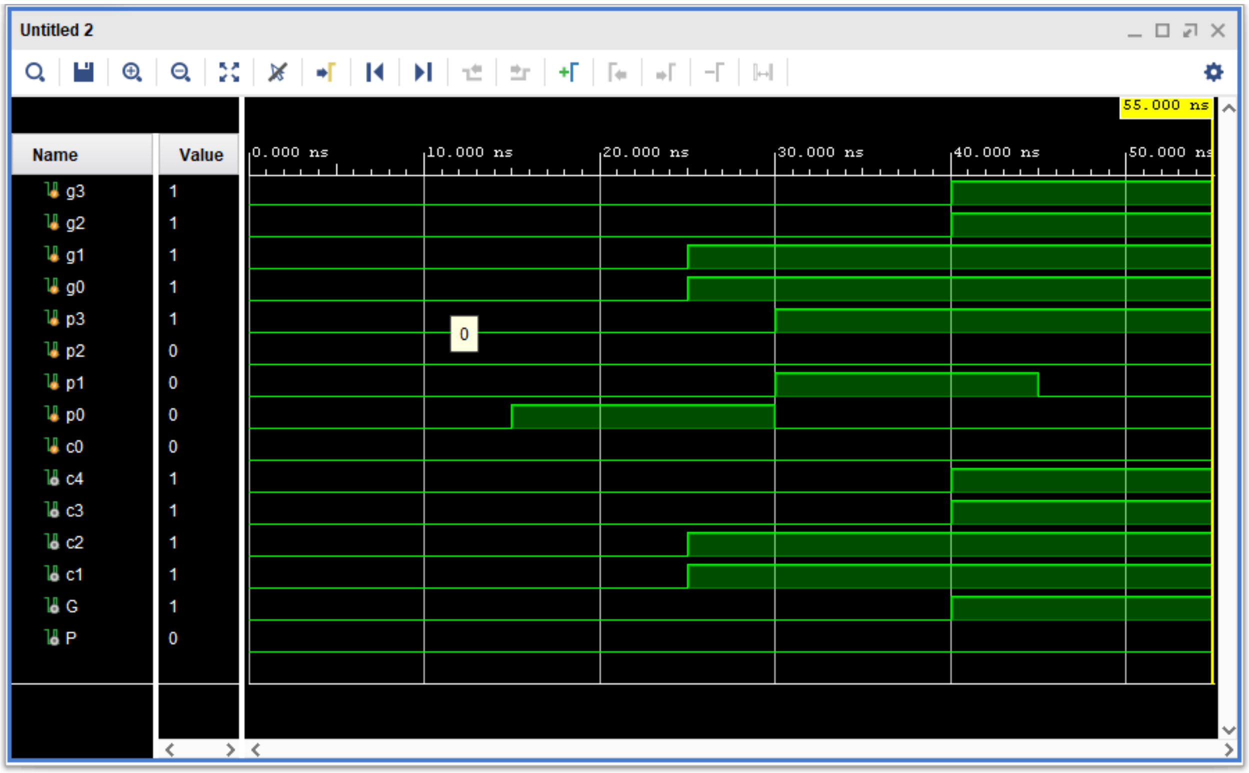


Note: Please reference the lecture slides to understand the meaning of inputs and outputs.

Verilog code, RTL schematic, and simulation result:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: 
// Engineer: 
// 
// Create Date: 02/21/2023 03:55:12 PM
// Design Name: 
// Module Name: CarryLookaheadUnit_4b
// Project Name: 
// Target Devices: 
// Tool Versions: 
// Description: 
// 
// Dependencies: 
// 
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
// 
//////////////////////////////////////////////////////////////////////////////////


module CarryLookaheadUnit_4b(c4, c3, c2, c1, G, P, g3, g2, g1, g0, p3, p2, p1, p0, c0);

    output c4, c3, c2, c1, G, P;
    input g3, g2, g1, g0, p3, p2, p1, p0, c0;
    
    assign P = p3 & p2 & p1 & p0;
    assign G = g3 | (p3 & g2) | (p3 & p2 & g1) | (p3 & p2 & p1 & g0);
    assign c1 = g0 | (p0 & c0);
    assign c2 = g1 | (p1 & g0) | (p1 & p2 & c0);
    assign c3 = g2 | (p2 & g1) | (p2 & p1 & g0) | (p2 & p1 & p0 & c0);
    assign c4 = g3 | (p3 & g2) | (p3 & p2 & g1) | (p3 & p2 & p1 & g0) | (p3 & p2 & p1 & p0 &
c0);

endmodule
```

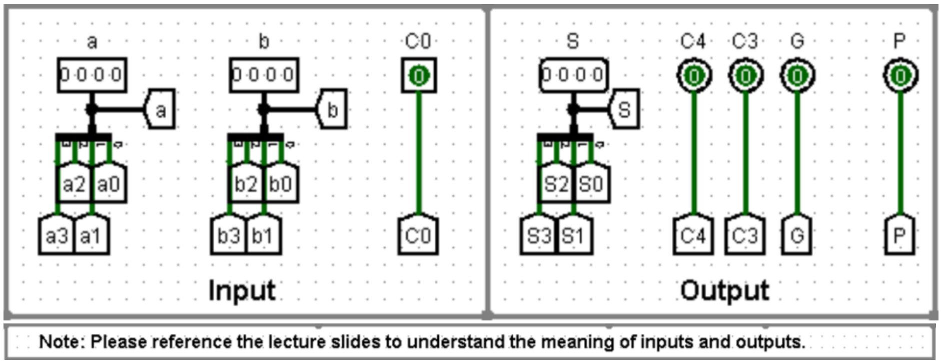




Name	Value
g3	1
g2	1
g1	1
g0	1
p3	1
p2	0
p1	0
p0	0
c0	0
c4	1
c3	1
c2	1
c1	1
G	1
P	0

4.4 Design a 4-bit Carry Lookahead Adder (15 points)

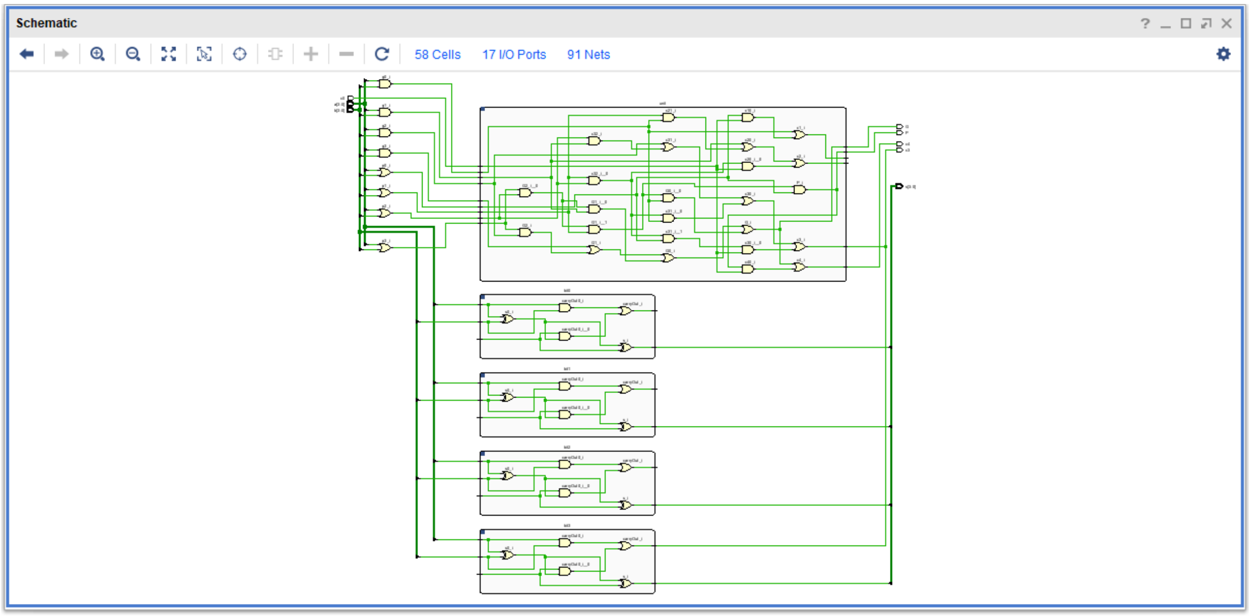
Requirement: only 1-bit full adders designed in 4.1, 4-bit Carry Lookahead Unit designed in 4.3, and basic logic gates are allowed.

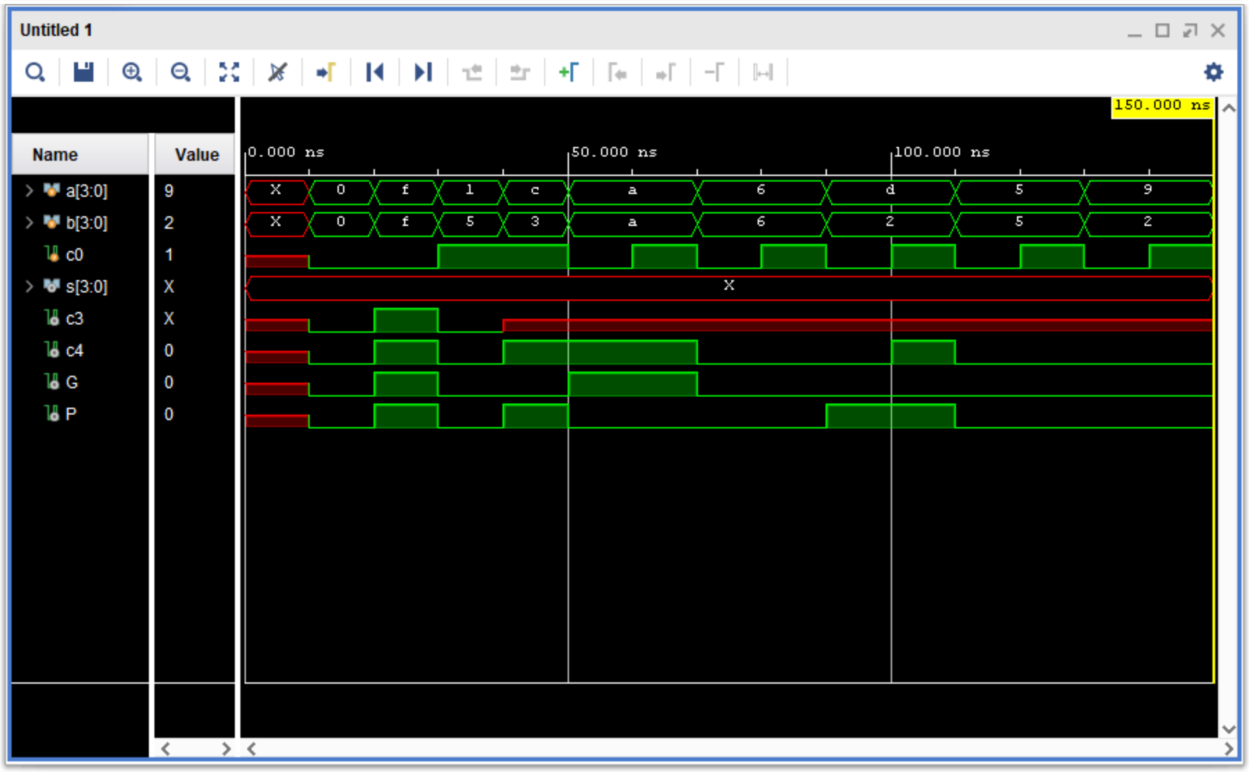


Note: Please reference the lecture slides to understand the meaning of inputs and outputs.

Verilog code, RTL schematic, and simulation result:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: 
// Engineer: 
// 
// Create Date: 02/21/2023 03:55:12 PM
// Design Name: 
// Module Name: CarryLookaheadAdder_4b
// Project Name: 
// Target Devices: 
// Tool Versions: 
// Description: 
// 
// Dependencies: 
// 
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
// 
//////////////////////////////////////////////////////////////////////////////////


module CarryLookaheadAdder_4b(c4, c3, G, P, s, c0, a, b);

    output [3:0]s;
    output c4, c3, G, P;
    input [3:0]a, b;
    input c0;

    wire g3, g2, g1, g0, p3, p2, p1, p0, o3, o2, o1, o0;

    assign g0 = a[0] & b[0];
    assign p0 = a[0] | b[0];
    assign g1 = a[1] & b[1];
    assign p1 = a[1] | b[1];
    assign g2 = a[2] & b[2];
    assign p2 = a[2] | b[2];
    assign g3 = a[3] & b[3];
    assign p3 = a[3] | b[3];

    CarryLookaheadUnit_4b unit(c4, c3, c2, c1, G, P, g3, g2, g1, g0, p3, p2, p1, p0, c0);
    FullAdder_1b bit0(s[0], , o0, a[0], b[0]);
    FullAdder_1b bit1(s[1], , o1, a[1], b[1]);
    FullAdder_1b bit2(s[2], , o2, a[2], b[2]);
    FullAdder_1b bit3(s[3], c3, o3, a[3], b[3]);

endmodule
```

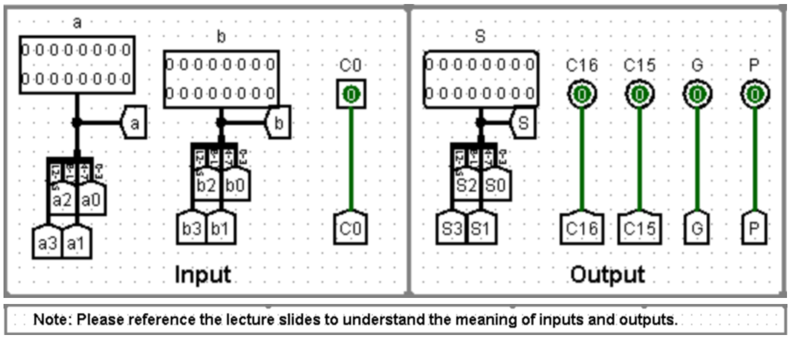




Name	Value
a[3:0]	9
b[3:0]	2
c0	1
s[3:0]	X
c3	X
c4	0
G	0
P	0

4.5 Design a 16-bit Carry Lookahead Adder (15 points)

Requirement: only 4-bit Carry Lookahead Adders designed in 4.4, 4-bit Carry Lookahead Unit designed in 4.3, and basic logic gates are allowed.

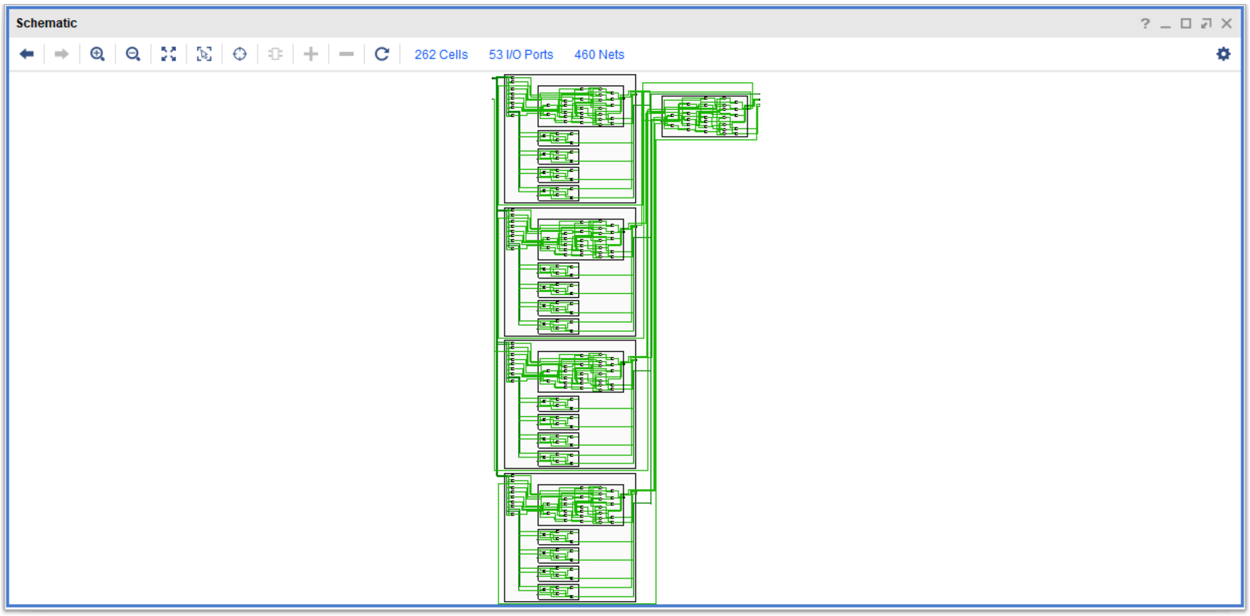


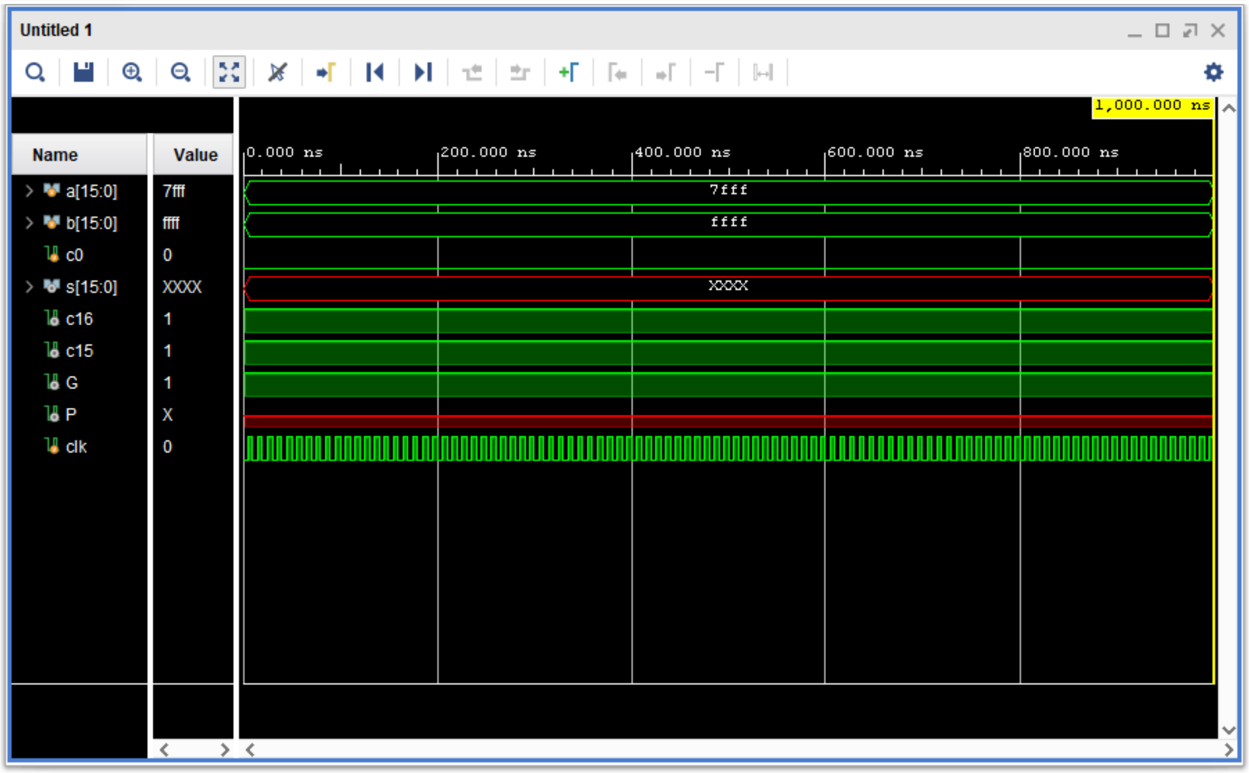
Note: Please reference the lecture slides to understand the meaning of inputs and outputs.

Verilog code, RTL schematic, and simulation result:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: 
// Engineer: 
// 
// Create Date: 02/21/2023 03:55:12 PM
// Design Name: 
// Module Name: CarryLookaheadAdder_16b
// Project Name: 
// Target Devices: 
// Tool Versions: 
// Description: 
// 
// Dependencies: 
// 
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
// 
//////////////////////////////////////////////////////////////////////////////////


module CarryLookaheadAdder_16b(s, c16, c15, G, P, a, b, c0);

    output [15:0]s;
    output c16, c15, G, P;
    input [15:0] a, b;
    input c0;

    wire g3, g2, g1, g0, p3, p2, p1, p0, c3, c2, c1, o0;

    assign c15 = c3;

    CarryLookaheadUnit_4b unit(c16, c3, c2, c1, G, P, g3, g2, g1, g0, p3, p2, p1, p0, c0);

    CarryLookaheadAdder_4b bit0_3( , , g0, p0, s[3:0], c0, a[3:0], b[3:0]);
    CarryLookaheadAdder_4b bit4_7( , , g1, p1, s[7:4], c1, a[7:4], b[7:4]);
    CarryLookaheadAdder_4b bit8_11( , , g2, p1, s[11:8], c2, a[11:8], b[11:8]);
    CarryLookaheadAdder_4b bit12_15( , , g3, p3, s[15:12], c3, a[15:12], b[15:12]);
```

```
endmodule
```

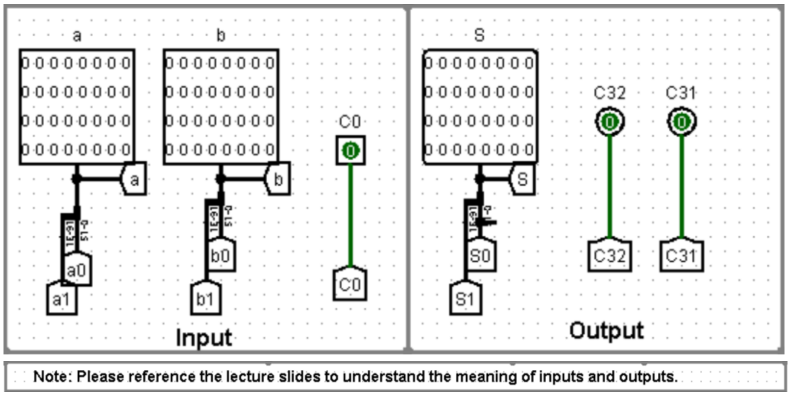




Name	Value
a[15:0]	7fff
b[15:0]	ffff
c0	0
s[15:0]	XXXX
c16	1
c15	1
G	1
P	X
clk	0

4.6 Design a 32-bit Carry Lookahead Adder (5 points)

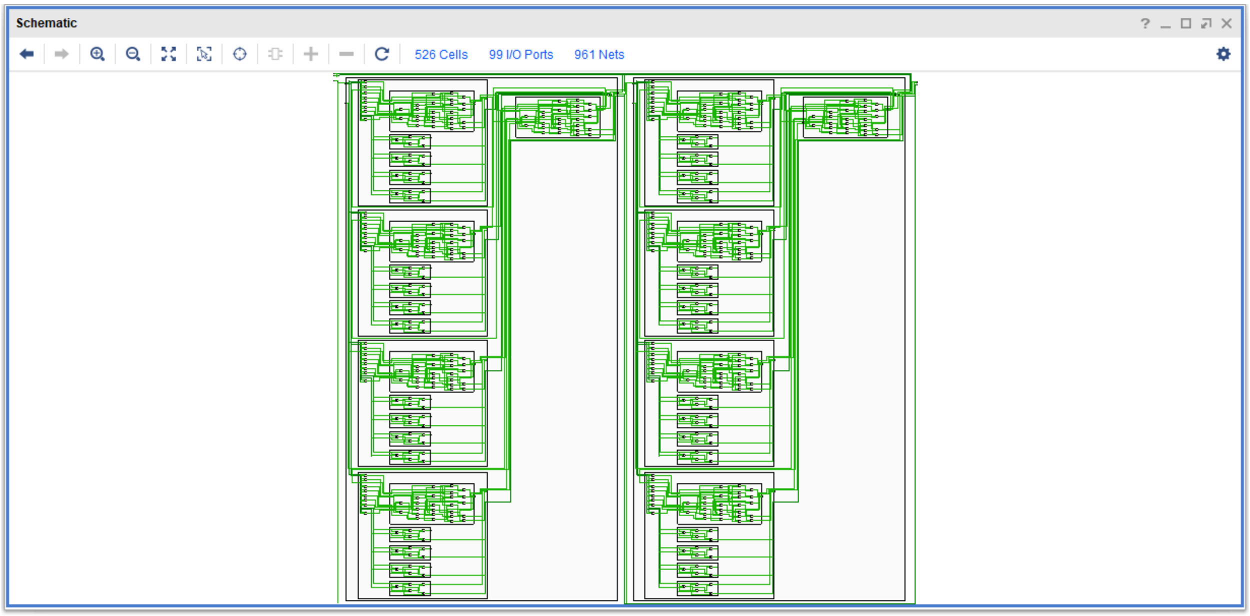
Requirement: only 16-bit Carry Lookahead Adders designed in 4.5 and basic logic gates are allowed.

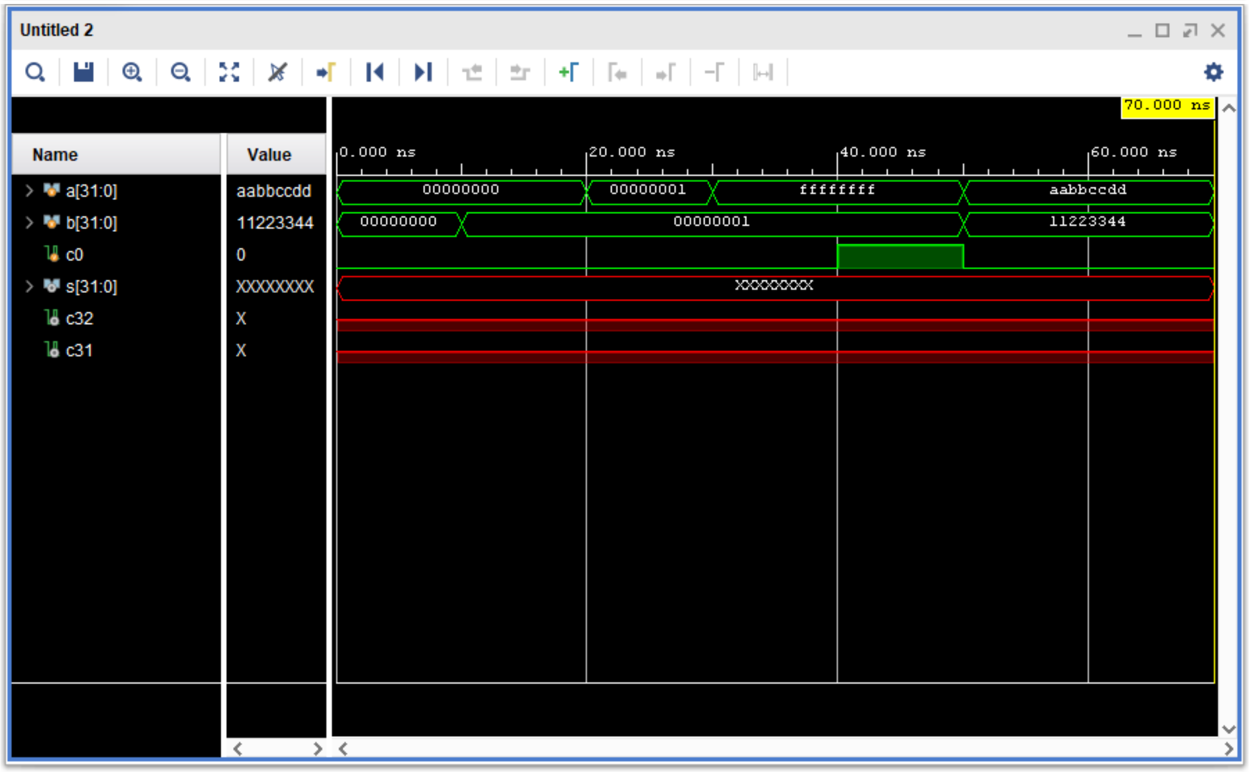


Note: Please reference the lecture slides to understand the meaning of inputs and outputs.

Verilog code, RTL schematic, and simulation result:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: 
// Engineer: 
// 
// Create Date: 02/21/2023 03:55:12 PM
// Design Name: 
// Module Name: CarryLookaheadAdder_32b
// Project Name: 
// Target Devices: 
// Tool Versions: 
// Description: 
// 
// Dependencies: 
// 
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
// 
//////////////////////////////////////////////////////////////////////////////////


module CarryLookaheadAdder_32b(s, c32, c31, a, b, c0);

    output [31:0]s;
    output c32, c31;
    input [31:0]a, b;
    input c0;

    wire c16;

    CarryLookaheadAdder_16b bit0_15(s[15:0], c16, , a[15:0], b[15:0], c0);
    CarryLookaheadAdder_16b bit16_31(s[31:16], c32, c31, a[31:16], b[31:16], c16);

endmodule
```

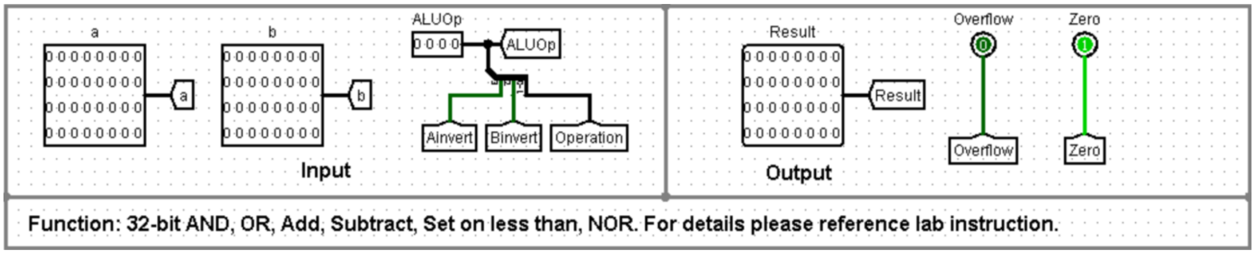




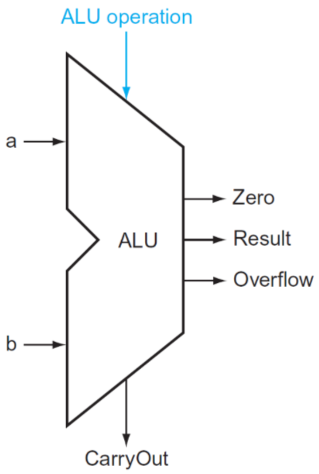
Name	Value
a[31:0]	aabbccdd
b[31:0]	11223344
c0	0
s[31:0]	XXXXXXXX
c32	X
c31	X

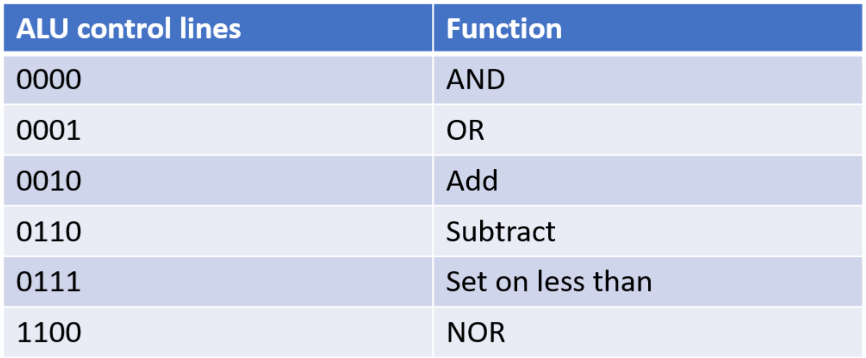
4.7 Design a 32-bit ALU (20 points)

Requirement: only 32-bit Carry Lookahead Adder designed in 4.6, multiplexers, extenders, splitters, and basic logic gates are allowed.


a
b
ALUOp
0000
ALUOp
a
b
Ainvert
Binvert
Operation
Input
Result
Overflow
Zero
Result
Overflow
Zero
Output
Function: 32-bit AND, OR, Add, Subtract, Set on less than, NOR. For details please reference lab instruction.

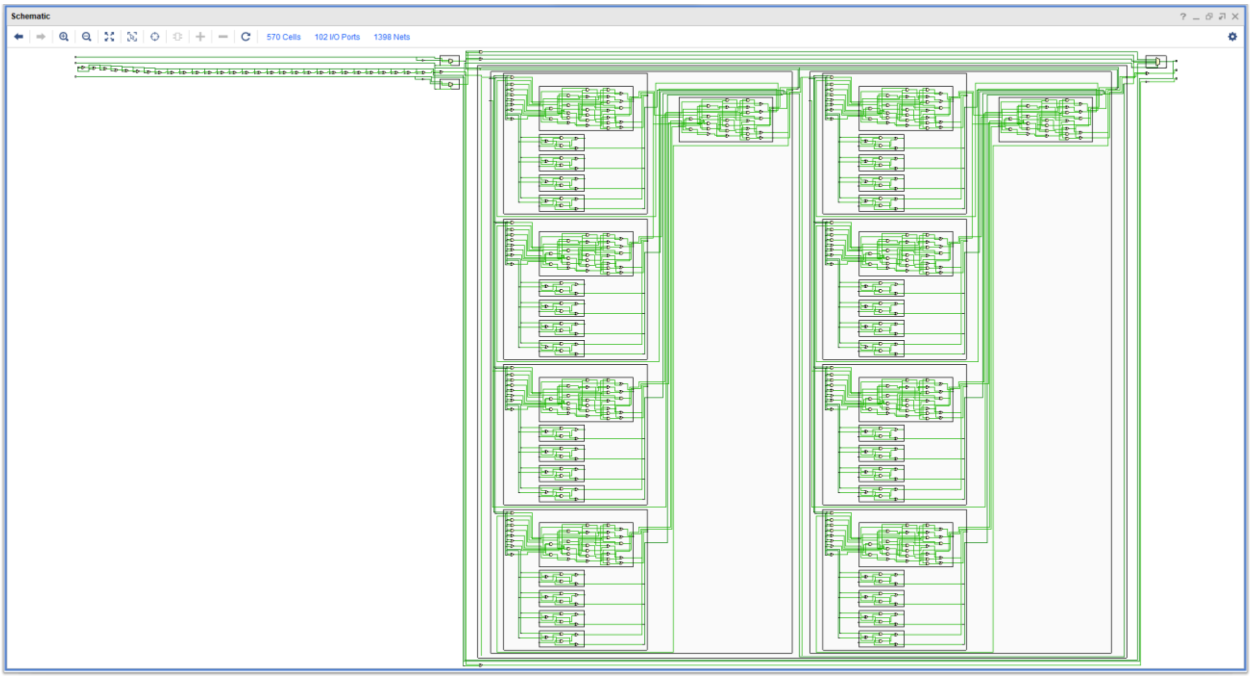
Verilog code, RTL schematic, and simulation result:

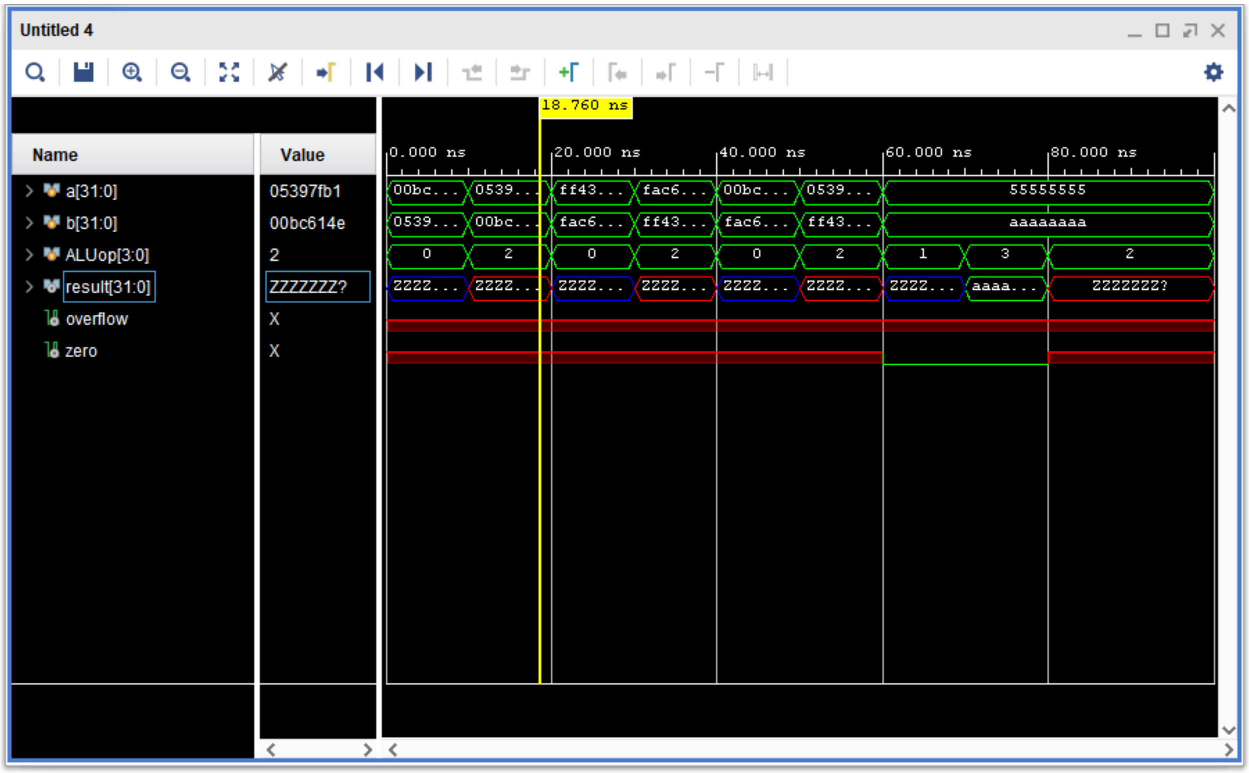




ALU control lines	Function
0000	AND
0001	OR
0010	Add
0110	Subtract
0111	Set on less than
1100	NOR

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: 
// Engineer: 
// 
// Create Date: 02/21/2023 03:55:12 PM
// Design Name: 
// Module Name: ArithmaticLogicUnit_32b
// Project Name: 
// Target Devices: 
// Tool Versions: 
// Description: 
// 
// Dependencies: 
// 
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
// 
//////////////////////////////////////////////////////////////////////////////////


module ArithmaticLogicUnit_32b(result, overflow, zero, a, b, ALUop);

    output [31:0]result;
    output overflow, zero;
    input [31:0]a, b;
    input [3:0]ALUop;

    wire [31:0]out;
    wire c32, c31, o;

    mux_2c_32b mux_a(a_in, ALUop[3], a, ~a);
    mux_2c_32b mux_b(b_in, ALUop[2], b, ~b);

    CarryLookaheadAdder_32b unit(o, c32, c31, a_in, b_in, ALUop[2]);

    mux_4c_32b mux(out, ALUop[1:0], a_in & b_in, a_in | b_in, o, b);

    assign result = out;
    assign overflow = c32 ^ c31;
    assign zero = ~(out[31] | out[30] | out[29] | out[28] |
                    out[27] | out[26] | out[25] | out[24] |
                    out[23] | out[22] | out[21] | out[20] |
                    out[19] | out[18] | out[17] | out[16] |
                    out[15] | out[14] | out[13] | out[12] |
                    out[11] | out[10] | out[9] | out[8] |
                    out[7] | out[6] | out[5] | out[4] |
                    out[3] | out[2] | out[1] | out[0]);

endmodule
```





Name	Value
a[31:0]	05397fb1
b[31:0]	00bc614e
ALUop[3:0]	2
result[31:0]	ZZZZZZZ?
overflow	X
zero	X