Digital Computer Design

Project 1 ALU Design

Project 1.D **ALU-2**

1- Create the following files fulladder.sv, NbitFulladder.sv, MUX2.sv and MUX4.sv.

```
module fulladder(input logic a, b, cin,
output logic s, cout);

        logic p, g;
        assign p = a ^ b;
        assign g = a & b;
        assign s = p ^ cin;
        assign cout = g | (p & cin);

endmodule
```

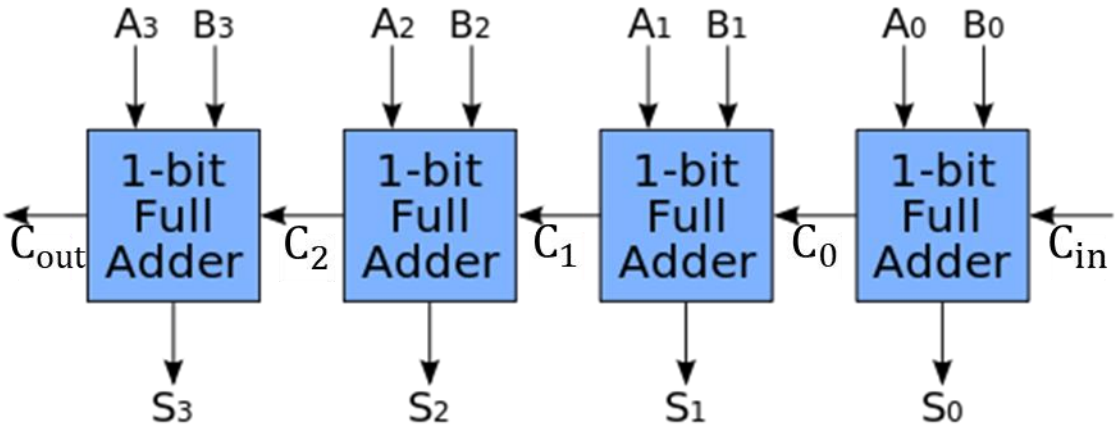
fulladder.sv

```
module NbitFulladder( input logic [3:0] a, b,
input logic cin,
output logic [3:0] s,
output logic cout );

        wire [2:0] c;
        fulladder i_0 (a[0],b[0],cin,s[0],c[0]);
        fulladder i_1 (a[1],b[1],c[0],s[1],c[1]);
        fulladder i_2 (a[2],b[2],c[1],s[2],c[2]);
        fulladder i_3 (a[3],b[3],c[2],s[3],cout);

endmodule
```

NbitFulladder.sv



4-bit full adder using 1-bit full adders

```
module MUX2(input logic [3:0] d0, d1,
            input logic s,
            output logic [3:0] y);

            assign y = s ? d1 : d0;

endmodule
```

MUX2.sv

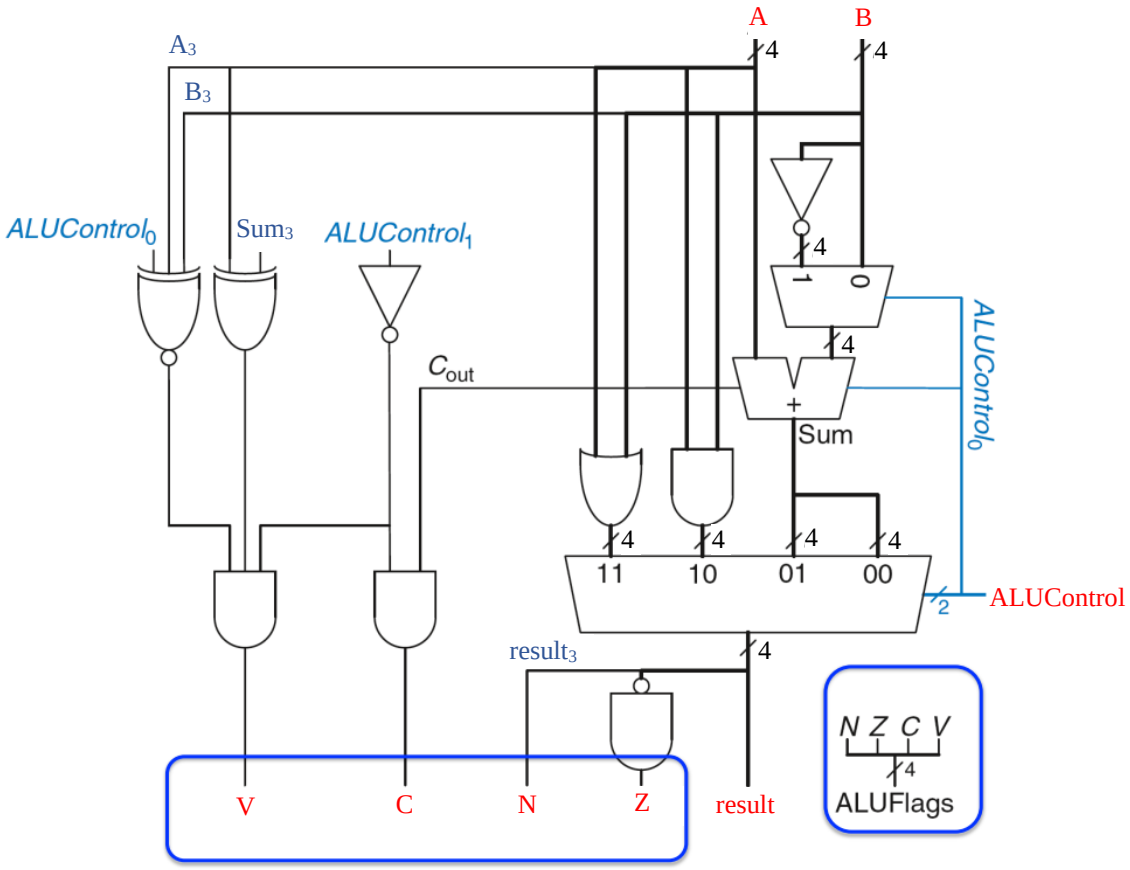
```
Module MUX4(input logic [3:0] d0, d1, d2, d3,
input logic [1:0] s,
output logic [3:0] y);

            assign y = s[1] ?
            (s[0] ? d3 : d2):
            (s[0] ? d1 : d0);

endmodule
```

MUX4.sv

2- Create 4-bit ALU module with flags given below using NbitFulladder, MUX2 and MUX4 modules. The input and output variable names (coloured as red) should be given as in the figure below.



ALU Control $_{1:0}$	Function
00	Add
01	Subtract
10	AND
11	OR

3- Create a **testbench** for the new ALU design, simulate the project and observe the outputs.