```
1   // AlU module
2   module ALU #(parameter W=4)
3         (input [W-1:0] A,
4         input [W-1:0] B,
5         input [2:0] ALUcntrl,
6         output reg [W-1:0] O,
7         output reg CO,
8         output reg OVF,
9         output reg N,
10        output reg Z);
11  // Since switch case block can be implemented only in always block
12  // I created always block with default sensitivity list
13  always @(*)
14  begin
15     OVF=0;// Assign default parameters
16     CO=0;// These parameters are meaningful in only addition and subtraction
17     // Doing operations according to value of control signals
18     case(ALUcntrl)
19        // Addition
20        3'b000:
21           begin
22           {CO,O} = A+B;
23           Z = (O==0) ? 1:0;
24           N=O[W-1];
25           // Overflow occurs when A and B are positive however result is negative
26           if (~A[W-1] & ~B[W-1])
27              begin
28              if(O[W-1])
29                 OVF=1;
30              end
31           // Overflow occurs when A and B are negative however result is positive
32           if (A[W-1] & B[W-1])
33              begin
34              if(~O[W-1])
35                 OVF=1;
36              end
37           end
38        // Subtraction
39        3'b001:
40           begin
41           {CO,O} = A-B;
42           Z = (O==0) ? 1:0;
43           // Overflow occurs when A is Negative and B is positive however result is positiv
44           if (A[W-1] & ~B[W-1])
45              begin
46              if(~O[W-1])
47                 OVF=1;
48              end
49           // Overflow occurs when A is Positive and B is negative however result is negativ
50           if (~A[W-1] & B[W-1])
51              begin
52              if(O[W-1])
53                 OVF=1;
54              end
55           N=O[W-1];
56           end
57        // Subtraction
58        3'b010:
59           begin
60           {CO,O} = B-A;
61           Z = (O==0) ? 1:0;
62           // Overflow occurs when A is positive and B is negative however result is positiv
63           if (~A[W-1] & B[W-1])
64              begin
65              if(~O[W-1])
66                 OVF=1;
```

```
67              end
68            // Overflow occurs when A is negative and B is positive however result is negativ
69            if (A[W-1] & ~B[W-1])
70               begin
71               if(O[W-1])
72                  OVF=1;
73               end
74            N=O[W-1];
75            end
76         //Bit Clear
77         3'b011:
78            begin
79            O = A & ~B;
80            Z = (O==0) ? 1:0;
81            N=O[W-1];
82            end
83         //AND operator
84         3'b100:
85            begin
86            O = A & B;
87            Z = (O==0) ? 1:0;
88            N=O[W-1];
89            end
90         //OR operator
91         3'b101:
92            begin
93            O = A | B;
94            Z = (O==0) ? 1:0;
95            N=O[W-1];
96            end
97         //XOR operator
98         3'b110:
99            begin
100           O = A ^ B;
101           Z = (O==0) ? 1:0;
102           N=O[W-1];
103           end
104        //XNOR operator
105        3'b111:
106           begin
107           O = ~(A ^ B);
108           Z = (O==0) ? 1:0;
109           N=O[W-1];
110           end
111     endcase
112  end
113  endmodule
```