

```

1  module ALU #(parameter W=8) (ALUCtrl,A,B,out,co,ovf,z,n);
2  input [(W-1):0] A,B;
3  input [2:0] ALUCtrl;
4  output [(W-1):0] out;
5  output reg co,ovf,z,n;
6  reg [(W-1):0] out;
7  reg [15:0] temp;
8  reg [7:0] multiplier_copy;
9  reg [15:0] multiplicand_copy;
10 reg [15:0] negative_output;
11 reg [15:0] product_temp;
12
13 reg [5:0] bit;
14
15 initial bit = 0;
16 initial negative_output = 0;
17
18
19 always @(*)
20 begin
21     case(ALUCtrl)
22     0: // Add
23     begin
24         {co,out} = A + B ;
25         z=(out == 0) ? 1 : 0;
26         if ((A[W-1] == 0 && B[W-1] == 0 && out[W-1] == 1) || (A[W-1] == 1 && B[W-1] == 1
&& out[W-1] == 0)) begin
27             ovf = 1;
28             n=0;
29         end else begin
30             ovf = 0;
31             if (out[W-1]==1) begin
32                 n=1;
33             end else begin
34                 n=0;
35             end
36         end
37     end
38     1: // SubAB
39     begin
40         {co,out} = A + (~B + 1) ; // (~B + 1)
41         z=(out == 0) ? 1 : 0;
42         if ((A[W-1] == 1 && B[W-1] == 0 && out[W-1] == 0) || (A[W-1] == 0 && B[W-1] ==
1 && out[W-1] == 1)) begin
43             ovf = 1;
44             n=0;
45         end else begin
46             ovf = 0;
47             if (out[W-1]==1) begin
48                 n=1;
49             end else begin
50                 n=0;
51             end
52         end
53     end
54     2: //and
55     begin
56         out = A & B;
57         co=0;
58         ovf=0;
59         z=(out == 0) ? 1 : 0;
60         if (out[W-1]==1) begin
61             n=1;
62         end else begin
63             n=0;
64         end
65     end
66     3: // or
67     begin
68         out = A | B;
69         co=0;
70         ovf=0;
71         z=(out == 0) ? 1 : 0;
72         if (out[W-1]==1) begin
73             n=1;
74         end else begin
75

```

```
76         n=0;
77     end
78 end
79
80 4: // xor
81 begin
82     out = A ^ B;
83     co=0;
84     ovf=0;
85     z=(out == 0) ? 1 : 0;
86     if (out[w-1]==1) begin
87         n=1;
88     end else begin
89         n=0;
90     end
91 end
92
93 5: // clear
94 begin
95     out = 0;
96     co=0;
97     ovf=0;
98     z=(out == 0) ? 1 : 0;
99     if (out[w-1]==1) begin
100         n=1;
101     end else begin
102         n=0;
103     end
104 end
105
106 6: // shift
107 begin
108     out = A;
109     co=0;
110     ovf=0;
111     z=(out == 0) ? 1 : 0;
112     if (out[w-1]==1) begin
113         n=1;
114     end else begin
115         n=0;
116     end
117 end
118
119     default: out = A + B ;
120 endcase
121 end
122
123 endmodule
124
```