```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3   USE ieee.std_logic_arith.ALL;
4   USE ieee.std_logic_signed.all;
5
6   ENTITY ALU IS
7   PORT( a, b          : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
8         op            : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
9         result        : INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
10        carry         : OUT STD_LOGIC;
11        zero          : OUT STD_LOGIC);
12  END ALU;
13
14  ARCHITECTURE description OF ALU IS
15  --Internal
16
17  --AND32
18  COMPONENT and32 IS
19  PORT( Ain, Bin        : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
20        result          : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
21  END COMPONENT;
22
23  --OR32
24  COMPONENT or32 IS
25  PORT( Ain, Bin        : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
26        result          : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
27  END COMPONENT;
28
29  --Adder
30
31  COMPONENT adder32 IS
32  PORT( Ain, Bin        : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
33        Cin             : IN STD_LOGIC;
34        Cout            : OUT STD_LOGIC;
35        result          : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
36  END COMPONENT;
37
38  --left shift
39  COMPONENT leftSH IS
40  PORT( a             : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
41        result          : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
42  END COMPONENT;
43
44  --right shift
45  COMPONENT rightSH IS
46  PORT( a             : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
47        result          : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
48  END COMPONENT;
49
50  --mux8 1
51  COMPONENT mux8_1 IS
52  PORT( x0, x1, x2, x3, x4, x5, x6, x7    : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
53        s                                 : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
54        y                                 : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
55  END COMPONENT;
56
57  --mux2_1
58  COMPONENT mux2_1 IS
59  PORT( x1, x2        : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
60        s             : IN STD_LOGIC;
61        y             : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
62  END COMPONENT;
63
64  --no32
65
66  COMPONENT no32 IS
67  PORT( Xin           : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
68        Yout          : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
```

```
69   END COMPONENT;
70   --Internal wires
71   SIGNAL andR       : STD_LOGIC_VECTOR(31 DOWNTO 0);
72   SIGNAL orR        : STD_LOGIC_VECTOR(31 DOWNTO 0);
73   SIGNAL addR       : STD_LOGIC_VECTOR(31 DOWNTO 0);
74   SIGNAL leftR      : STD_LOGIC_VECTOR(31 DOWNTO 0);
75   SIGNAL rightR     : STD_LOGIC_VECTOR(31 DOWNTO 0);
76   SIGNAL negR       : STD_LOGIC_VECTOR(31 DOWNTO 0);
77   SIGNAL notb       : STD_LOGIC_VECTOR(31 DOWNTO 0);
78
79   BEGIN
80   --connect mux8_1
81   mux1: mux8_1 PORT MAP(x0 => andR,
82                         x1 => orR,
83                         x2 => addR,
84                         x3 => "ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ",
85                         x4 => leftR,
86                         x5 => rightR,
87                         x6 => addR,
88                         x7 => "ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ",
89                         s  => op,
90                         y  => result);
91
92   --connect adder32
93   adder1: adder32 PORT MAP( Ain    => a,
94                             Bin    => negR,
95                             Cin    => op(2),
96                             Cout   => carry,
97                             result => addR);
98   --connect and32
99   and1: and32 PORT MAP (a, b, andR);
100
101  --connect or32
102  or1: or32 PORT MAP(a, b, orR);
103
104  --connect leftSH
105  leftSH1: leftSH PORT MAP(a, leftR);
106
107  --connect rightSH
108  rightSH1: rightSH PORT MAP(a, rightR);
109
110  --connect negative decider mux2_1
111  negMux: mux2_1 PORT MAP( x1  => b,
112                           x2  => notb,
113                           s   => op(2),
114                           y   => negR);
115
116  --connect not gate
117  notber: no32 PORT MAP(b, notb);
118
119  --zero flag
120  zero <= NOT(result(0) OR result(1) OR result(2) OR result(3) OR result(4) OR result(5) OR
     result(6) OR result(7) OR result(8) OR result(9) OR result(10) OR result(11) OR result(12)
     OR result(13) OR result(14) OR result(15) OR result(16) OR result(17) OR result(18) OR
     result(19) OR result(20) OR result(21) OR result(22) OR result(23) OR result(24) OR  result(
     25) OR result(26) OR result(27) OR result(28) OR result(29) OR result(30) OR result(31));
121
122  END description;
```