```
1   library ieee;
2   use ieee.std_logic_1164.all;
3   use ieee.std_logic_arith.all;
4   use ieee.std_logic_unsigned.all;
5   use ieee.numeric_std.all;
6
7   entity alu is
8      port(
9         a, b: in std_logic_vector(31 downto 0);
10        op: in std_logic_vector(2 downto 0);
11        result: inout std_logic_vector(31 downto 0);
12        cout, zero: out std_logic);
13  end alu;
14
15  architecture description of alu is
16     component adder32 is
17        port(
18           cin, mode: in std_logic;
19           x, y: in std_logic_vector(31 downto 0);
20           s : out std_logic_vector(31 downto 0);
21           cout: out std_logic);
22     end component adder32;
23        signal addd: std_logic_vector(31 downto 0);
24        signal coutplus: std_logic;
25        signal shift: std_logic_vector(31 downto 0);
26           begin
27              muxadd: adder32 port map (op(2), op(2), a, b, addd, coutplus);
28                 process (op)
29                    begin
30                       if op = "000" then
31                          result <= a and b;
32                       elsif op = "001" then
33                          result <= a or b;
34                       elsif op = "010" then
35                          result <= addd;
36                          cout <= coutplus;
37                       elsif op = "110" then
38                          result <= addd;
39                          cout <= coutplus;
40                       elsif op = "100" then
41                          shift(0) <= a(31);
42                          shift(31 downto 1) <= a(30 downto 0);
43                          result <= shift;
44                       elsif op = "101" then
45                          shift(31) <= a(0);
46                          shift(30 downto 0) <= a(31 downto 1);
47                          result <= shift;
48                       else
49                          result <= (others => '0');
50                       end if;
51
52                       if result = "00000000000000000000000000000000"  then
53                          zero <= '1';
54                       else
55                          zero <= '0';
56                       end if;
57
58                end process;
59
60  end description;
61
62
```