```
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;
6      entity alu is
7   port(
8         a : in std_logic_vector(31 downto 0);
9         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 : out std_logic;
13        zero : out std_logic);
14        end alu;
15
16  architecture description of alu is
17
18        COMPONENT adder32
19        port(
20         cin         :IN STD_LOGIC;
21         mode        :IN STD_LOGIC; -- 1 = sub , 0 = add
22         x           : in std_logic_vector(31 downto 0);
23         y           : in std_logic_vector(31 downto 0);
24         s           : out std_logic_vector(31 downto 0);
25         cout        :OUT STD_LOGIC
26        );
27        END COMPONENT;
28        signal    addd         :  std_logic_vector(31 downto 0);
29        signal    coutplus     :  std_logic;
30        signal    sft    :  std_logic_vector(31 downto 0);
31
32
33   begin
34  -- code goes here
35  muxadd: adder32 port map(op(2), op(2), a, b, addd, coutplus);
36  process(op)begin --start op
37
38     if(op = "000")then --op if start
39        --  a and b
40        result <= a and b ;
41
42  elsif (op = "001")then
43        -- a or b
44        result <= a or b ;
45
46  elsif (op = "010")then
47        result <= addd;
48        cout <= coutplus;
49        -- a+b
50
51  elsif (op = "110")then
52          result <= addd;
53          cout <= coutplus;
54        -- a-b
55
56  elsif (op = "100")then
57        sft(0)<= a(31);
58        sft(31 downto 1) <= a(30 downto 0) ;
59        result <= ((sft) );
60        -- a<<1
61
62  elsif (op = "101")then
```

```
63          sft(31)<= a(0);
64          sft(30 downto 0) <= a(31 downto 1) ;
65          result <= ((sft) );
66          --result <= (a srl '1');
67          -- a>>1
68
69          end if; -- end op if
70
71          if(result = "00000000000000000000000000000000" )then -- zero if start
72                zero <= '1';
73                else
74                zero <= '0';
75          end if; -- end zero if
76
77     end process; --end op process
78
79    end description;
80
```

```
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      entity mux2too1 is
6      port(
7         st : in std_logic;
8         w0,w1 : in std_logic;
9         cin : out std_logic
10        );
11        end mux2too1;
12
13  architecture description of mux2too1 is begin
14     --if (st = '0') then -- st if start
15        --cin <= '0';
16        --elsif(st = '1')then
17        --cin <= '1';
18     --end if; --st if end
19  end description;
20
```

```
 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
 6  ENTITY adder32 IS
 7     PORT(cin       :IN STD_LOGIC;
 8          mode      :IN STD_LOGIC; -- 1 = sub , 0 = add
 9          x         : in std_logic_vector(31 downto 0);
10          y         : in std_logic_vector(31 downto 0);
11          s         : out std_logic_vector(31 downto 0);
12          cout      :OUT STD_LOGIC
13     );
14  end adder32;
15
16  ARCHITECTURE description of adder32 IS
17     SIGNAL c31,c30,c29,c28,c27,c26,c25,c24,c23,c22,c21,c20,c19,c18,c17,c16,c15,c14,c13,c12,
    c11,c10,c9,c8,c7,c6,c5,c4,c3,c2,c1,c0 : STD_LOGIC;
18     COMPONENT adder
19        port(
20        x : in std_logic;
21        y : in std_logic;
22        mode : in std_logic;
23        cin : in std_logic;
24        cout : out std_logic;
25        s : out std_logic
26        );
27        END COMPONENT;
28  BEGIN
29        --st0: adder port map (x(0), y(0), mode, cin, c1, s(0));
30        st0: adder port map (x(0), y(0), mode, mode, c1, s(0));
31        st1: adder port map (x(1), y(1), mode, c1, c2, s(1));
32        st2: adder port map (x(2), y(2), mode, c2, c3, s(2));
33        st3: adder port map (x(3), y(3), mode, c3, c4, s(3));
34        st4: adder port map (x(4), y(4), mode, c4, c5, s(4));
35        st5: adder port map (x(5), y(5), mode, c5, c6, s(5));
36        st6: adder port map (x(6), y(6), mode, c6, c7, s(6));
37        st7: adder port map (x(7), y(7), mode, c7, c8, s(7));
38        st8: adder port map (x(8), y(8), mode, c8, c9, s(8));
39        st9: adder port map (x(9), y(9), mode, c9, c10, s(9));
40        st10: adder port map (x(10), y(10), mode, c10, c11, s(10));
41        st11: adder port map (x(11), y(11), mode, c11, c12, s(11));
42        st12: adder port map (x(12), y(12), mode, c12, c13, s(12));
43        st13: adder port map (x(13), y(13), mode, c13, c14, s(13));
44        st14: adder port map (x(14), y(14), mode, c14, c15, s(14));
45        st15: adder port map (x(15), y(15), mode, c15, c16, s(15));
46        st16: adder port map (x(16), y(16), mode, c16, c17, s(16));
47        st17: adder port map (x(17), y(17), mode, c17, c18, s(17));
48        st18: adder port map (x(18), y(18), mode, c18, c19, s(18));
49        st19: adder port map (x(19), y(19), mode, c19, c20, s(19));
50        st20: adder port map (x(20), y(20), mode, c20, c21, s(20));
51        st21: adder port map (x(21), y(21), mode, c21, c22, s(21));
52        st22: adder port map (x(22), y(22), mode, c22, c23, s(22));
53        st23: adder port map (x(23), y(23), mode, c23, c24, s(23));
54        st24: adder port map (x(24), y(24), mode, c24, c25, s(24));
55        st25: adder port map (x(25), y(25), mode, c25, c26, s(25));
56        st26: adder port map (x(26), y(26), mode, c26, c27, s(26));
57        st27: adder port map (x(27), y(27), mode, c27, c28, s(27));
58        st28: adder port map (x(28), y(28), mode, c28, c29, s(28));
59        st29: adder port map (x(29), y(29), mode, c29, c30, s(29));
60        st30: adder port map (x(30), y(30), mode, c30, c31, s(30));
61        st31: adder port map (x(31), y(31), mode, c31, cout, s(31));
```

```
62    end description;
63
```

```
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      entity adder is
6      port(
7         x : in std_logic;
8         y : in std_logic;
9         mode : in std_logic;
10        cin : in std_logic;
11        cout : out std_logic;
12        s : out std_logic
13        );
14        end adder;
15
16  architecture description of adder is
17     signal t :  std_logic;
18  begin
19  process(mode) begin
20        if(mode = '1')then
21        --t<= ( ( not y ) xor mode);-- if 0 add if sub
22        t<= ( (  y ) xor mode);-- if 0 add if sub
23        else
24        t<= y;
25        end if;
26        --s <= x xor t xor cin;
27        s <= (x xor t xor cin) or ( x and t and cin );
28        cout <= (x and t) or (x and cin) or (cin and t) ;
29        end process;
30  end description;
31
```