```
1
2   ---------------------------------Floating Point Multiplier-----------------------------------
3
4   library IEEE;
5   use IEEE.STD_LOGIC_1164.ALL;
6   use IEEE.NUMERIC_STD.ALL;
7
8   entity floating_point_mul_8bit is
9       Port ( a : in  unsigned(7 downto 0);
10             b : in  unsigned(7 downto 0);
11             prod : out  unsigned(7 downto 0);
12             suc_flag : out  std_logic);
13  end floating_point_mul_8bit;
14
15  architecture Behavioral of floating_point_mul_8bit is
16
17  begin
18     process(a,b)
19     --declaring variable for each part of the floating point number
20     variable sign_a,sign_b,sign_z,suc_var: std_logic;
21     variable exp_a,exp_b,exp_z: unsigned(3 downto 0);
22     variable man_a,man_b: unsigned(4 downto 0);
23     variable man_z: unsigned(9 downto 0);
24     begin
25     --checking infinite value in inputs
26     if (a(6 downto 4)="111" or b(6 downto 4)="111") then
27     suc_flag<='0';
28     prod<="01110000";
29     else
30
31        --assigning value to each variable
32           sign_a:=a(7);
33           sign_b:=b(7);
34           exp_a(2 downto 0):=a(6 downto 4);
35           exp_a(3):='0';
36           exp_b(2 downto 0):=b(6 downto 4);
37           exp_b(3):='0';
38           man_a(3 downto 0):=a(3 downto 0);
39           man_b(3 downto 0):=b(3 downto 0);
40           man_a(4):='1';
41           man_b(4):='1';
42        --success if no one lowers success flag
```

```
43             suc_var:='1';
44
45          --difference between exponents
46             exp_z:=exp_a+exp_b;
47             exp_z:=exp_z-"0011";
48          --checking underflow
49             if exp_z(3)='1' then
50                suc_var:='0';
51             end if;
52
53          --multiplying mantissas
54             man_z:=resize(man_a*man_b,10);
55
56          --normalising
57             if (man_z(9)='1') then
58                man_z:=man_z srl 1;
59                exp_z:=exp_z+"0001";
60             end if;
61
62          --raising overflow flag and lowering sucess flag
63             if (exp_z(3)='1') then
64                suc_var:='0';
65             end if;
66
67          -- calculating sign of product
68             sign_z:=sign_a xor sign_b;
69
70          --Assigning final values to prod
71             prod(3 downto 0)<=man_z(7 downto 4);
72             prod(6 downto 4)<=exp_z(2 downto 0);
73             prod(7)<=sign_z;
74             suc_flag<=suc_var;
75          end if;
76          end process;
77    end Behavioral;
78
79
```