```
1
2   ------------------------------------Floating Point Divide Unit----------------------------------
3
4   library IEEE;
5   use IEEE.STD_LOGIC_1164.ALL;
6   use IEEE.numeric_std.all;
7
8   entity division2 is
9       Port ( a  : in  unsigned(7 DOWNTO 0);
10             b : in  unsigned(7 DOWNTO 0);
11             q : out unsigned(7 downto 0) := "00000000";
12             suc_flag: out std_logic);
13  end division2;
14
15  architecture Behave1 of division2 is
16  signal overflow,underflow: std_logic:='0';
17  begin
18  process(a, b)
19  variable m : unsigned(5 downto 0) := "000000";
20  variable n : unsigned(4 downto 0) := "00000";
21  variable g : unsigned(4 downto 0) := "00000";
22  variable exp1 : unsigned(4 downto 0) := "00000";
23  variable exp2 : unsigned(4 downto 0) := "00000";
24  variable u1 : unsigned(4 downto 0) := "00000";
25  variable u2 : unsigned(4 downto 0) := "00000";
26  variable t : unsigned(10 downto 0) := "00000000000";
27  variable q1 : unsigned(7 downto 0) := "00000000";
28  variable check1 : STD_LOGIC;
29  variable check2 : STD_LOGIC;
30  begin
31
32  if (a(6 downto 0) = "0000000" and b(6 downto 0) = "0000000")    ---------------- Always taking positive QNaN
33  then
34     q1 := "01111111";
35  elsif (a(6 downto 0) = "0000000")                 --------- Always taking positive zero
36  then
37     q1 := "00000000";
38  elsif (b(6 downto 0) = "0000000")                 ---------- Always taking positive infinity
39  then
40     q1 := "01110000";
41  else
42     exp1(2 downto 0) := unsigned(a(6 downto 4));
```

```
43       exp2(2 downto 0) := unsigned(b(6 downto 4));
44       u1 := exp1 - exp2;
45       u2 := "00011";
46       u1 := u2 + u1;
47
48       if (u1(4) = '1' or u1 = "00000")
49       then
50          underflow <= '1';
51          q1 := "00000000";
52       elsif ((u1(4) = '0' and u1(3) = '1') or (u1 = "00111"))
53       then
54          overflow <= '1';
55          q1 := "00000000";
56       else
57          if(a(7) = '1')
58          then
59             check1 := '1';
60          else
61             check1 := '0';
62          end if;
63             t(3 downto 0) := unsigned(a(3 downto 0));
64             t(10 downto 5) := "000000";
65             t(4) := '1';
66
67          if(b(7) = '1')
68          then
69             check2 := '1';
70          else
71             check2 := '0';
72          end if;
73             m(3 downto 0) := unsigned(b(3 downto 0));
74             m(5 downto 4) := "01";
75
76          for k in 4 downto 0
77          loop
78          t := t sll 1;
79          t(10 downto 5) := t(10 downto 5) - m;
80          if (t(10) = '1')
81          then
82             t(0) := '0';
83             t(10 downto 5) := t(10 downto 5) + m;
84          else
```

```
85             t(0) := '1';
86          end if;
87          end loop;
88
89          n := t(4 downto 0);
90          g := "00001";
91
92          for i in 0 to 4
93          loop
94             n := n sll 1;
95             u1 := u1 - g;
96             if(n(4) = '1')
97             then
98                exit;
99             end if;
100         end loop;
101
102         if (u1(4) = '1' or u1 = "00000")
103         then
104            underflow <= '1';
105            q1 := "00000000";
106         elsif ((u1(4) = '0' and u1(3) = '1') or (u1 = "00111"))
107         then
108            overflow <= '1';
109            q1 := "00000000";
110         else
111            if ((check1 = '1' and check2 = '0') or (check1 = '0' and check2 = '1'))
112            then
113               q1(7) :=  '1';
114               q1(6 downto 4) := u1(2 downto 0);
115               q1(3 downto 0) := n(3 downto 0);
116            else
117               q1(7) :=  '0';
118               q1(6 downto 4) := u1(2 downto 0);
119               q1(3 downto 0) := n(3 downto 0);
120            end if;
121         end if;
122      end if;
123   end if;
124   q <= q1;
125   suc_flag<=overflow or underflow;
126   end process;
```

```
127   end Behave1;
128
```