```
1
2   ----------------------------------------4) FLOATING POINT ADDITION----------------------------------
3
4
5   library IEEE;
6   use IEEE.STD_LOGIC_1164.ALL;
7   use IEEE.NUMERIC_STD.ALL;
8
9   entity FloatingPointUnit is
10      Port ( Sa : in  STD_LOGIC;
11             Ea : in  UNSIGNED (2 downto 0);
12             Ma : in  UNSIGNED (3 downto 0);
13             Sb : in  STD_LOGIC;
14             Eb : in  UNSIGNED (2 downto 0);
15             Mb : in  UNSIGNED (3 downto 0);
16             AddSubtract : in STD_LOGIC;
17             Sr : out  STD_LOGIC;
18             Er : Buffer  UNSIGNED (2 downto 0);
19             Mr : Buffer  UNSIGNED (3 downto 0);
20             ErrorFlag : out STD_LOGIC);
21  end FloatingPointUnit;
22
23  architecture Structural of FloatingPointUnit is
24
25  ---------------------------------Component Declearations-----------------------------------------
26  component LeadZeroDet is
27      Port ( M : in  UNSIGNED (9 downto 0) :="0000000000";
28             Carry : in STD_LOGIC;
29             X : out  INTEGER := 0;
30             ShiftDirection : out  STD_LOGIC);
31  end component LeadZeroDet;
32
33  component MUX is
34      Port ( Ea : in  UNSIGNED (2 downto 0);
35             Eb : in  UNSIGNED (2 downto 0);
36             Sign : in STD_LOGIC := '0';
37             ErTemp : out  UNSIGNED (2 downto 0));
38  end component MUX;
39
40  component FivetoTenBit is
41      Port ( Min : in  UNSIGNED (4 downto 0);
42             Mout : out  UNSIGNED (9 downto 0));
```

```
43   end component FivetoTenBit;
44
45   component MantissaSubtractor is
46       Port ( Ma : in  UNSIGNED (9 downto 0) := "0000000000"; --No of STD_LOGICs are : 4(Mantissa) + 1(Implied
     STD_LOGIC), sign STD_LOGIC is considered in sum_ans variable
47              Mb : in  UNSIGNED (9 downto 0) := "0000000000"; --Earlier was 3 downto 0
48              AddSub : in STD_LOGIC;
49              Magnitude : out  UNSIGNED (9 downto 0) := "0000000000";
50              Carry : out STD_LOGIC := '0'; -- This can be generated in case of addition of 1.Ma + 1.Mb
51              Sign : out STD_LOGIC);
52   end component MantissaSubtractor;
53
54   component NormaliseAndRound is
55       Port ( M : in  UNSIGNED (9 downto 0);
56              X : in  INTEGER := 0;
57              ShiftDirection : in STD_LOGIC := '0';
58              Mr : Buffer  UNSIGNED (3 downto 0));
59   end component NormaliseAndRound;
60
61   component SHIFTER is
62      -- generic (N : natural := 5); --This is the input size of Mantissa,Including implied bit
63       Port ( Min : in  UNSIGNED (4 downto 0):= "00000"; --1.M means total 1+4 bits (1 of implied bit,and 4 of mantissa)
64              Shift : in  UNSIGNED(2 downto 0) := "000"; -- since exponent are of 3bits, their differenec is also 3 bits
65              Mout : out  UNSIGNED (9 downto 0) := "0000000000"); --Max shift possible is 5, so we have included them
66   end component SHIFTER;
67
68   component SWAP is
69       generic (N : natural := 4); --This is the input size of Mantissa,Excluding implied bit
70       Port ( Ma : in  UNSIGNED (N-1 downto 0); --As per decided convention
71              Mb : in  UNSIGNED (N-1 downto 0);
72              SwapIn : in STD_LOGIC;
73              Mout1 : out  UNSIGNED (N downto 0); --Output size increses as we attach the implied bit here itself
74              Mout2 : out  UNSIGNED (N downto 0));
75   end component SWAP;
76
77   component SignMag3BitSub is
78       Port ( Ea : in  UNSIGNED (2 downto 0) := "000";
79              Eb : in  UNSIGNED (2 downto 0) := "000";
80              MagofDiff : out  UNSIGNED (2 downto 0) := "000";
81              Sign : out STD_LOGIC);
82   end component SignMag3BitSub;
83
```

```
 84   component ControlLogic is
 85       Port ( Sa : in  STD_LOGIC; --Sign of FP no. a
 86              Sb : in  STD_LOGIC; --Sign of FP no. b
 87              AS : in  STD_LOGIC; --The Operation Desired by the user
 88              S4 : in  STD_LOGIC; --Sign of the mantissa difference
 89              S3 : in  STD_LOGIC; --Sign of Exponent Difference
 90              Sr : out  STD_LOGIC; --Sign of Final Result
 91
 92              AddSub : out  STD_LOGIC); --Operation signal to the mantissa adder or subtractor
 93   end component ControlLogic;
 94
 95   component ExpSubtractor is
 96       Port ( X : in  INTEGER := 0;
 97              TempEr : in UNSIGNED(2 downto 0);
 98              Er : Buffer UNSIGNED (2 downto 0);
 99              ShiftDirection : in STD_LOGIC);
100
101   end component ExpSubtractor;
102
103   component Error is
104       Port ( Mr : in  UNSIGNED (3 downto 0);
105              Er : in  UNSIGNED (2 downto 0);
106              E : out  STD_LOGIC);
107   end component Error;
108
109   ------------------------------------------Component Instantiation ----------------------------------------------
110   signal Sign,ShiftDirection,S4,addsub,carry : STD_LOGIC := 'U';
111   Signal MagofDiff,erTemp : UNSIGNED(2 downto 0);
112   signal magnitude,Shifted_SmallerM,TenBitLargerM : UNSIGNED(9 DOWNTO 0);
113   signal M_with_ELarger,M_with_ESmaller : UNSIGNED(4 downto 0);
114   signal X:INTEGER;
115   begin
116
117      ExpSub1 : SignMag3BitSub port map(Ea,Eb,MagofDiff,Sign);
118      Swapper : SWAP port map(Ma,Mb,Sign,M_with_ELarger,M_with_ESmaller);
119      Shiftr : SHIFTER port map(M_with_ESmaller,MagofDiff,Shifted_SmallerM);
120      FivetoTenB : FivetoTenBit port map(M_with_ELarger,TenBitLargerM);
121      Control : ControlLogic port map(Sa,Sb,AddSubtract,S4,Sign,Sr,AddSub);
122      MantAddSub : MantissaSubtractor port map(TenBitLargerM,Shifted_SmallerM,AddSub,Magnitude,Carry,S4);
123      LZD : LeadZeroDet port map(Magnitude,Carry,X,ShiftDirection);
124      NormRnd : NormaliseAndRound port map(Magnitude,X,Shiftdirection,Mr);
125      Mux1 : MUX port map(Ea,Eb,Sign,ErTemp);
```

```
126       ExpSub2 : ExpSubtractor port map(X,ErTemp,Er,ShiftDirection);
127       ErrorBlock : Error port map(Mr,Er,ErrorFlag);
128
129    end Structural;
130
131
```