```
1   ----------------------------------------------------------------------------------------------------
2   ----------------------------------COMPONENTS OF FLOATING POINT UNIT---------------------------------
3   ----------------------------------------------------------------------------------------------------
4
5   -------------------------------Exponent Subtractor(Comparator)-----------------------------------
6   library IEEE;
7   use IEEE.STD_LOGIC_1164.ALL;
8   use IEEE.NUMERIC_STD.ALL;
9
10  -- Sign Magnitude implementation
11  entity SignMag3BitSub is
12      Port ( Ea : in  UNSIGNED (2 downto 0) := "000";
13             Eb : in  UNSIGNED (2 downto 0) := "000";
14             MagofDiff : out  UNSIGNED (2 downto 0) := "000";
15             Sign : out STD_LOGIC);
16  end SignMag3BitSub;
17
18  architecture Behavioral of SignMag3BitSub is
19
20  begin
21     process(Ea,Eb)
22     variable sum_ans : UNSIGNED (3 downto 0) := "0000";
23     variable carry_ans : UNSIGNED (4 downto 0) := "00000";
24     variable Eanew : UNSIGNED (3 downto 0) := "0000"; --1 bit extra for knowing the sign
25     variable Ebnew : UNSIGNED (3 downto 0) := "0000";
26     variable TempOperand : UNSIGNED (3 downto 0) := "0000";
27     variable TempCarry : UNSIGNED (4 downto 0) := "00001";
28     begin
29        carry_ans(0) := '1';
30        Eanew(2 downto 0) := Ea; --remaining bits as usual
31        Ebnew(2 downto 0) := Eb xor "111"; --2's complement of Eb. (Refer the rule of subtraction using 2's complement)
32
33        for i in 0 to 3 loop
34           sum_ans(i) := (Eanew(i) xor Ebnew(i)) xor carry_ans(i);
35           carry_ans(i+1) := (Eanew(i) and Ebnew(i)) or (Eanew(i) and carry_ans(i)) or (Ebnew(i) and carry_ans(i));
36        end loop;
37
38        if sum_ans(3) = '1' then
39           MagofDiff <= sum_ans(2 downto 0); --Ans is +ve, so no change
40        else
41           sum_ans(2 downto 0) := sum_ans(2 downto 0) xor "111"; -- Ans is -ve, so ans is 2's complement of itself
42           for i in 0 to 3 loop
```

```
43                  TempCarry(i+1) := (sum_ans(i) and TempOperand(i)) or (sum_ans(i) and TempCarry(i)) or (TempOperand(i)
    and TempCarry(i));
44                  sum_ans(i) := (sum_ans(i) xor TempOperand(i)) xor TempCarry(i);
45              end loop; -- This part is to add 1 to sum_ans in order to complete the 2's complement
46              MagofDiff <= sum_ans(2 downto 0);
47          end if;
48          Sign <= not sum_ans(3); -- Zero is +ve, One is -ve, But in program it comes opposite
49
50     end process;
51   end Behavioral;
52
53
```