```vhdl

------------------------------MAIN CODE----FLOATINGPOINT_ALU.VHD-----------------------------------------

--LIBRARIES TO USE
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

--ENTITY OF THE MICRO PROCESSOR
entity floating_point_alu is
   port( num1,num2 : in unsigned(7 downto 0);
         op_code:in unsigned(2 downto 0);
         clk:in std_logic;
         output: out unsigned(7 downto 0);
         op_flag:out std_logic);
end floating_point_alu;

--ARCHTECTURE OF THE MICRO PROCESSOR
architecture Behavioral of floating_point_alu is
--COMPONENT OF MULTIPLEXER
component mux_main is
   port(in_add, in_sub, in_mul, in_div, fl_add, fl_sub, fl_mul, fl_div : in unsigned(7 downto 0);
         in_add_fg,in_sub_fg,in_mul_fg,in_div_fg,fl_add_fg,fl_sub_fg,fl_mul_fg,fl_div_fg,clk:in std_logic;
      op_code:in unsigned( 2 downto 0 );
      res:out unsigned(7 downto 0);
      suc_flag:out std_logic);
end component;
--COMPONENT OF ADDER-SUBTRACTOR UNIT
component AdderSubtractor is
    Port ( InpA : in  UNSIGNED (7 downto 0);
           InpB : in  UNSIGNED (7 downto 0);
            Opr : in STD_LOGIC := '0'; -- '0' means Add(Default) , '1' means Subtract
           Sum : out  Unsigned (7 downto 0);
           Error : out STD_LOGIC);
end component;

--COMPONENT OF INTEGER MULTIPLICATION
component int_mul is
      port(num1,num2: in unsigned(7 downto 0);
         prod:out unsigned(7 downto 0);
         suc_flag: out std_logic);
end component;
```

```

--COMPONENT OF INTEGER DIVISION
component divison1 is
    Port ( a : in  unsigned(7 downto 0);
           b : in  unsigned(7 downto 0);
           q : out unsigned(7 downto 0);
--         r : out unsigned(7 downto 0));
            suc_flag:out std_logic);
end component;

--COMPONENT OF FLOATING POINT ADDITION AND SUBTRACTION
component FloatingPointUnit is
      Port ( Sa : in  STD_LOGIC;
           Ea : in  UNSIGNED (2 downto 0);
           Ma : in  UNSIGNED (3 downto 0);
           Sb : in  STD_LOGIC;
           Eb : in  UNSIGNED (2 downto 0);
           Mb : in  UNSIGNED (3 downto 0);
           AddSubtract : in STD_LOGIC;
           Sr : out  STD_LOGIC;
           Er : Buffer  UNSIGNED (2 downto 0);
           Mr : Buffer  UNSIGNED (3 downto 0);
           ErrorFlag : out STD_LOGIC);

end component;

--COMPONENT OF FLAOTING POINT MULTIPLICATION
component floating_point_mul_8bit is
    Port ( a : in  unsigned(7 downto 0);
           b : in  unsigned(7 downto 0);
           prod : out  unsigned(7 downto 0);
           suc_flag : out  std_logic);

end component;

--COMPONENT OF FLOATING POINT DIVISION
component division2 is
    Port ( a  : in  unsigned(7 DOWNTO 0);
           b : in  unsigned(7 DOWNTO 0);
           q : out unsigned(7 downto 0) := "00000000";
           suc_flag: out std_logic);
end component;
```

```vhdl



--SIGNAL USED
signal in_add_res, in_sub_res, in_mul_res, in_div_res, fl_add_res,fl_sub_res,fl_mul_res,fl_div_res: unsigned(7 downto 0);
signal  in_add_suc,in_sub_suc,in_mul_suc,in_div_suc,  fl_add_suc,fl_sub_suc,fl_mul_suc,fl_div_suc: std_logic;
--MAIN BODY
begin
--INTEGER ADDER
INT_ADD_INS:   AdderSubtractor port map(num1,num2,'0',in_add_res,in_add_suc);
--INTEGER SUBTRACTOR
INT_SUB_INS:   AdderSubtractor port map(num1,num2,'1',in_sub_res,in_sub_suc);
--INTEGER MULTIPLIER
INT_MUL_INS:   int_mul port map(num1,num2,in_mul_res,in_mul_suc);
--INTEGER DIVISION UNIT
INT_DIV_INS:   divison1 port map(num1,num2,in_div_res,in_div_suc);
--FLOATING POINT ADDER
FLT_ADD_INS:   FloatingPointUnit port map(num1(7),num1(6 downto 4),num1(3 downto 0),num2(7),num2(6 downto 4),num2(3 downto 0),'0',fl_add_res(7),fl_add_res(6 downto 4),fl_add_res(3 downto 0),fl_add_suc);
--FLOATING POINT SUBTRACTOR
FLT_SUB_INS:   FloatingPointUnit port map(num1(7),num1(6 downto 4),num1(3 downto 0),num2(7),num2(6 downto 4),num2(3 downto 0),'0',fl_sub_res(7),fl_sub_res(6 downto 4),fl_sub_res(3 downto 0),fl_sub_suc);
--FLOATING POINT MULTIPLIER
FLT_MUL_INS:   floating_point_mul_8bit port map(num1,num2,fl_mul_res,fl_mul_suc);
--FLOATING POINT DIVISION UNIT
FLT_DIV_INS:   division2 port map(num1,num2,fl_div_res,fl_div_suc);
--MUX TO SELECT THE OUTPUT
MUX_INS:       mux_main port map(in_add_res,  in_sub_res,  in_mul_res,  in_div_res, fl_add_res, fl_sub_res, fl_mul_res, fl_div_res, in_add_suc,in_sub_suc,in_mul_suc,in_div_suc,not(fl_add_suc),not(fl_sub_suc),fl_mul_suc, fl_div_suc,clk,op_code,output,op_flag);
--END OF ARCHITECTURE
end Behavioral;


```