گزارشکار آزمایش ۱۴

: add_sub_floating_point

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.Numeric_Std.all;
use ieee.std_logic_unsigned.all;
entity add_sub_floating_point is
port(
operand_1 : IN std_logic_vector(7 downto 0)
--operand(7):sign  operand(6 downto   4):exponent
 operand_2 : IN std_logic_vector(7 downto 0);--operand(3 downto to 0):fraction
  clk       : IN std_logic;
operator  : IN std_logic;  -- 0:add 1:sub
overflow : OUT std_logic;
result   :  OUT  std_logic_vector(7 downto 0)
);
end add_sub_floating_point;
architecture Behavioral of add_sub_floating_point is
 begin
   process(clk) is
    variable meaning_section_operand_1 : unsigned(4 downto 0);
    variable meaning_section_operand_2 : unsigned(4 downto 0);
   variable maximum_exponent : std_logic_vector(2 downto 0);
    variable meaning_section_result : unsigned(5 downto 0);
    variable sign_of_result : std_logic ;
    variable  j: integer;
    variable ovf : std_logic:='0';
            begin
              if(rising_edge (clk)) then
```

```
meaning_section_operand_1(4) := '1'; meaning_section_operand_1(3)
:=operand_1(3);meaning_section_operand_1(2) :=operand_1(2);
meaning_section_operand_1(1) :=operand_1(1);meaning_section_operand_1(0)
:=operand_1(0);

meaning_section_operand_2(4):= '1';  meaning_section_operand_2(3)
:=operand_2(3);meaning_section_operand_2(2) :=operand_2(2);
meaning_section_operand_2(1) :=operand_2(1);meaning_section_operand_2(0)
:=operand_2(0);

 if(operand_1(6 downto 4) > operand_2(6 downto 4)) then

  maximum_exponent := operand_1(6 downto 4);

 for j in 1 to 10 loop

  meaning_section_operand_2(3 downto 0) := meaning_section_operand_2(4 downto 1);

   meaning_section_operand_2(4) := '0';

   if(j = conv_integer(operand_1(6 downto 4) - operand_2(6 downto 4)))then

            exit;

end if;

end loop;

 end if;

 if(operand_1(6 downto 4) < operand_2(6 downto 4))then

    maximum_exponent := operand_2(6 downto 4);

 for j in 1 to 10 loop

meaning_section_operand_1(3 downto 0) := meaning_section_operand_1(4 downto 1);

 meaning_section_operand_1(4) := '0';

  if(j = conv_integer(operand_2(6 downto 4) - operand_1(6 downto 4)))then

 exit;

 end if;

end loop;

end if;

 if(operator = '0')then

   if (operand_2(7)='0')then

    meaning_section_result :=
resize(meaning_section_operand_1,6)+meaning_section_operand_2;
```

```
sign_of_result :='0';
else
 if(meaning_section_operand_1 < meaning_section_operand_2)then
 sign_of_result := '1';
meaning_section_result := resize(meaning_section_operand_2,6) –
meaning_section_operand_1;
  else
    sign_of_result := '0';
   meaning_section_result := resize(meaning_section_operand_1,6) -
meaning_section_operand_2;
end if;
end if;
 else
  if (operand_2(7)='0')then
     if(meaning_section_operand_1 < meaning_section_operand_2)then
          sign_of_result := '1';
          meaning_section_result := resize(meaning_section_operand_2,6) -
          meaning_section_operand_1;
     else
    sign_of_result := '0';
   meaning_section_result := resize(meaning_section_operand_1,6) -
meaning_section_operand_2;
  end if;
 else
meaning_section_result := resize(meaning_section_operand_1,6) +
meaning_section_operand_2;
sign_of_result := '0';
end if;
end if;
if(meaning_section_result(5 downto 4) = "11" or meaning_section_result(5 downto 4) =
"10")then
```

```
meaning_section_result(4 downto 0) := meaning_section_result(5
downto1);meaning_section_result(5):='0';
maximum_exponent := maximum_exponent + 1;
 end if;
if(meaning_section_result(5 downto 4) = "00")then
  for j in 1 to 4 loop
   meaning_section_result(5 downto 1) := meaning_section_result(4
   downto 0);  meaning_section_result(0):='0';
   maximum_exponent := maximum_exponent - 1;
     if(meaning_section_result(5 downto 4) = "01")then
            exit;
     end if;
    end loop;
   end if;
   end if;
  if(maximum_exponent ="111" )then
              ovf :='1';
 else
              ovf:='0';
end if;
 if(ovf ='0')then
            result(7)<= sign_of_result;
            result(6 downto 4) <= maximum_exponent ;
             result(3) <= meaning_section_result(3);
             result(2) <= meaning_section_result(2);
             result(1) <= meaning_section_result(1);
             result(0) <= meaning_section_result(0);
 else
   result <= "XXXXXXXX";
 end if;
```

```
  overflow <= ovf;
            end process;
end Behavioral;
```

: add_sub_floating_point_tb

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY add_sub_floating_point_tb IS
END add_sub_floating_point_tb;
ARCHITECTURE behavior OF add_sub_floating_point_tb IS
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT add_sub_floating_point
    PORT(
         operand_1 : IN  std_logic_vector(7 downto 0);
         operand_2 : IN  std_logic_vector(7 downto 0);
         clk : IN  std_logic;
         operator : IN  std_logic;
         overflow : OUT  std_logic;
         result : OUT  std_logic_vector(7 downto 0)
        );
    END COMPONENT;
   --Inputs
   signal operand_1 : std_logic_vector(7 downto 0) := (others => '0');
   signal operand_2 : std_logic_vector(7 downto 0) := (others => '0');
   signal clk : std_logic := '0';
   signal operator : std_logic := '0';
  --Outputs
   signal overflow : std_logic;
   signal result : std_logic_vector(7 downto 0);
```

```
BEGIN

        -- Instantiate the Unit Under Test (UUT)

  uut: add_sub_floating_point PORT MAP (

       operand_1 => operand_1,

       operand_2 => operand_2,

       clk => clk,

       operator => operator,

       overflow => overflow,

       result => result

      );

clk <= '1' ,'0' after 100 ns , '1' after 200 ns , '0' after 300 ns ,'1' after 400 ns , '0' after 500 ns ,
'1' after 600 ns, '0' after 700 ns , '1' after 800 ns;

 operand_1 <= "01100000" ,"01101111" after 800 ns;

  operand_2 <= "01000000" ,"11000000" after 400 ns , "01101000" after 800 ns;

   operator <='0' ,'1' after 200 ns ,'0' after 400 ns , '1' after 600 ns , '0' after 800 ns;

END;
```

خروجی شبیه سازی add_sub_floating_point_tb :