



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных Технологий

Кафедра инструментального и прикладного программного обеспечения

ЛАБОРАТОРНАЯ РАБОТА №2

Разработка программно-аппаратного сумматора чисел с плавающей точкой

по дисциплине:

Разработка программно-аппаратного обеспечения информационных и
автоматизированных систем

Выполнил: Миронов Д. С.

Группа: ИКМО-05-23

Преподаватель: Унгер А.Ю.

Москва 2023

Цель работы: разработать сумматор чисел с плавающей точкой, использовать 13-битный формат числа с плавающей точкой, где 8 бит мантисса + 4 бита порядок + 1 бит знака = 13 бит. Порядок - величина беззнаковая.

Листинг 1 – код модуля сортировки

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity lab2 is
    Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          c : in  STD_LOGIC_vector(3 downto 0);
          d : in  STD_LOGIC_vector(3 downto 0);
          e : in  STD_LOGIC_vector(7 downto 0);
          f : in  STD_LOGIC_vector(7 downto 0);
          g : out STD_LOGIC;
          h : out STD_LOGIC;
          i : out STD_LOGIC_vector(3 downto 0);
          j : out STD_LOGIC_vector(3 downto 0);
          k : out STD_LOGIC_vector(7 downto 0);
          l : out STD_LOGIC_vector(7 downto 0));
end lab2;
architecture Behavioral of lab2 is
begin
    process (a,b,c,d,e,f)
    begin
        if ((c&e)>(d&f)) then
            g<=a;
            h<=b;
            i<=c;
            j<=d;
            k<=e;
            l<=f;
        else
            g<=b;
            h<=a;
            i<=d;
            j<=c;
            k<=f;
            l<=e;
        end if;
    end process;
end Behavioral;
```

Листинг 2 – код модуля выравнивания порядков

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
```

```

use IEEE.NUMERIC_STD.ALL;
entity lab2_2 is
  Port ( z : in  STD_LOGIC_vector(3 downto 0) ;
        x : in  STD_LOGIC_vector(3 downto 0);
        n : in  STD_LOGIC_vector(7 downto 0);
        v : out STD_LOGIC_vector(7 downto 0));
end lab2_2;
architecture Behavioral of lab2_2 is
  signal p : unsigned(3 downto 0 );
  begin
    p <= unsigned(z) - unsigned(x);
    with p select
      v <= n                                     when "0000",
      "0" & n(7 downto 1 ) when "0001",
      "00" & n(7 downto 2 ) when "0010",
      "000" & n(7 downto 3 ) when "0011",
      "0000" & n(7 downto 4 ) when "0100",
      "00000" & n(7 downto 5 ) when "0101",
      "000000" & n(7 downto 6 ) when "0110",
      "0000000" & n(7) when "0111",
      "00000000" when others;
  end Behavioral;

```

Листинг 3 – код модуля сложения

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity lab2_sum is
  Port ( g : in  STD_LOGIC;
        k : in  STD_LOGIC;
        n : in  STD_LOGIC_vector (7 downto 0);
        v : in  STD_LOGIC_vector (7 downto 0);
        sum : out STD_LOGIC_vector (8 downto 0));
end lab2_sum;
architecture Behavioral of lab2_sum is
  signal ot_k , ot_v, ot_sum: unsigned( 8 downto 0);
begin
  ot_k <= "0" & unsigned(n);
  ot_v <= "0" & unsigned(v);
  process (g,k,ot_k,ot_v)
  begin
    if g = k then
      ot_sum <= ot_k + ot_v;
    else
      ot_sum <= ot_k - ot_v;
    end if;
  end process;
  sum <= std_logic_vector(ot_sum);
end Behavioral;

```

Листинг 4 – код модуля нормализации

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity lab2_normal is
    Port ( sum : in  STD_LOGIC_vector(8 downto 0);
          z : in  STD_LOGIC_vector(3 downto 0);
          e : out STD_LOGIC_vector(3 downto 0);
          fr : out STD_LOGIC_vector(7 downto 0));
end lab2_normal;
architecture Behavioral of lab2_normal is
    signal l_op : unsigned(2 downto 0);
    signal sum_norm : std_logic_vector(7 downto 0);
begin
    l_op <= "000" when (sum(7) = '1')
        else
            "001" when (sum(6) = '1')
        else
            "010" when (sum(5) = '1')
        else
            "011" when (sum(4) = '1')
        else
            "100" when (sum(3) = '1')
        else
            "101" when (sum(2) = '1')
        else
            "110" when (sum(1) = '1')
        else
            "111";

    with l_op select
        sum_norm <= std_logic_vector(sum(7 downto 0)) when "000",
        std_logic_vector(sum(6 downto 0)) & "0" when "001",
        std_logic_vector(sum(5 downto 0)) & "00" when "010",
        std_logic_vector(sum(4 downto 0)) & "000" when "011",
        std_logic_vector(sum(3 downto 0)) & "0000" when "100",
        std_logic_vector(sum(2 downto 0)) & "00000" when "101",
        std_logic_vector(sum(1 downto 0)) & "000000" when "110",
        std_logic_vector(sum(0 downto 0)) & "0000000" when others;

    process (sum, sum_norm, z, l_op)
    begin
        if (sum(8) = '1') then
            e <= std_logic_vector(unsigned(z) + 1);
            fr <= sum(8 downto 1);
        elsif (l_op > unsigned(z)) then
            e <= (others => '0');
            fr <= (others => '0');
        else
            e <= std_logic_vector(unsigned(z) - l_op);
            fr <= std_logic_vector(sum_norm(7 downto 0));
        end if;
    end process;
end Behavioral;
```

Листинг 5 – код тестов модуля сортировки

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY lab2_test_sort IS
END lab2_test_sort;
ARCHITECTURE behavior OF lab2_test_sort IS
    COMPONENT lab2
    PORT(
        a : IN  std_logic;
        b : IN  std_logic;
        c : IN  std_logic_vector(3 downto 0);
        d : IN  std_logic_vector(3 downto 0);
        e : IN  std_logic_vector(7 downto 0);
        f : IN  std_logic_vector(7 downto 0);
        g : OUT std_logic;
        h : OUT std_logic;
        i : OUT std_logic_vector(3 downto 0);
        j : OUT std_logic_vector(3 downto 0);
        k : OUT std_logic_vector(7 downto 0);
        l : OUT std_logic_vector(7 downto 0) );
    END COMPONENT;
    signal a : std_logic := '0';
    signal b : std_logic := '0';
    signal c : std_logic_vector(3 downto 0) := (others => '0');
    signal d : std_logic_vector(3 downto 0) := (others => '0');
    signal e : std_logic_vector(7 downto 0) := (others => '0');
    signal f : std_logic_vector(7 downto 0) := (others => '0');
    signal g : std_logic;
    signal h : std_logic;
    signal i : std_logic_vector(3 downto 0);
    signal j : std_logic_vector(3 downto 0);
    signal k : std_logic_vector(7 downto 0);
    signal l : std_logic_vector(7 downto 0);
BEGIN
    uut: lab2 PORT MAP (
        a => a,
        b => b,
        c => c,
        d => d,
        e => e,
        f => f,
        g => g,
        h => h,
        i => i,
        j => j,
        k => k,
        l => l );
    stim_proc: process
    begin
        -- hold reset state for 10 ns.
        -- wait for 10 ns;
        a <= '0';
        b <= '0';
        c <= "1010";
        d <= "1001";
        e <= "10000100";
        f <= "10111010";
```

```

wait for 10 ns;
a <= '0';
b <= '0';
c <= "1001";
d <= "1010";
e <= "10111010";
f <= "10000100";
wait for 10 ns;
a <= '0';
b <= '0';
c <= "1010";
d <= "1010";
e <= "10111010";
f <= "10000100";
wait for 10 ns;
a <= '0';
b <= '0';
c <= "1010";
d <= "1010";
e <= "10000100";
f <= "10111010";
wait for 10 ns;
end process;
END;

```

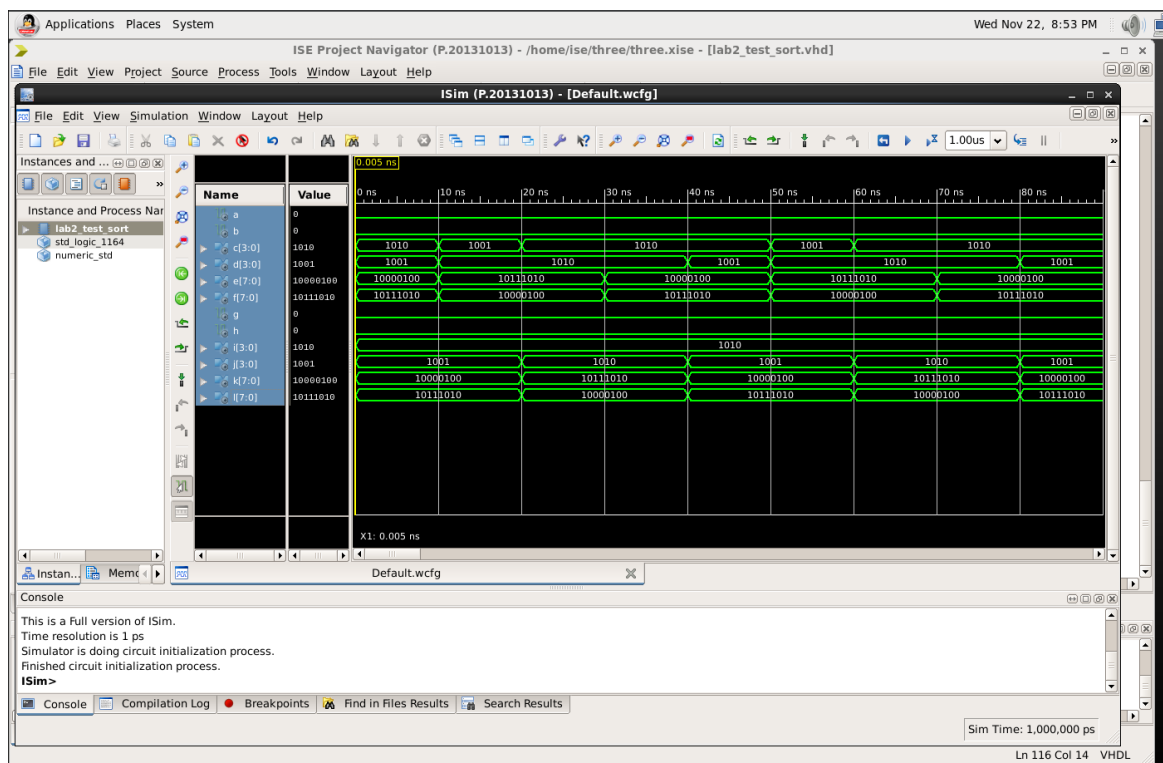


Рисунок 1 –прохождение тестов модуля сортировки

Листинг 6 – код тестов модуля выравнивания порядков

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY lab2_test_equals IS

```

```

END lab2_test_equals;
ARCHITECTURE behavior OF lab2_test_equals IS
  COMPONENT lab2_2
  PORT(
    z : IN  std_logic_vector(3 downto 0);
    x : IN  std_logic_vector(3 downto 0);
    n : IN  std_logic_vector(7 downto 0);
    v : OUT std_logic_vector(7 downto 0)    );
  END COMPONENT;
  signal z : std_logic_vector(3 downto 0) := (others => '0');
  signal x : std_logic_vector(3 downto 0) := (others => '0');
  signal n : std_logic_vector(7 downto 0) := (others => '0');
  signal v : std_logic_vector(7 downto 0);
BEGIN
  uut: lab2_2 PORT MAP (
    z => z,
    x => x,
    n => n,
    v => v    );
  stim_proc: process
  begin
    x <= "1000";
    z <= "1010";
    n <= "10000011";
    wait for 10 ns;
    wait;
    end process;
END;

```

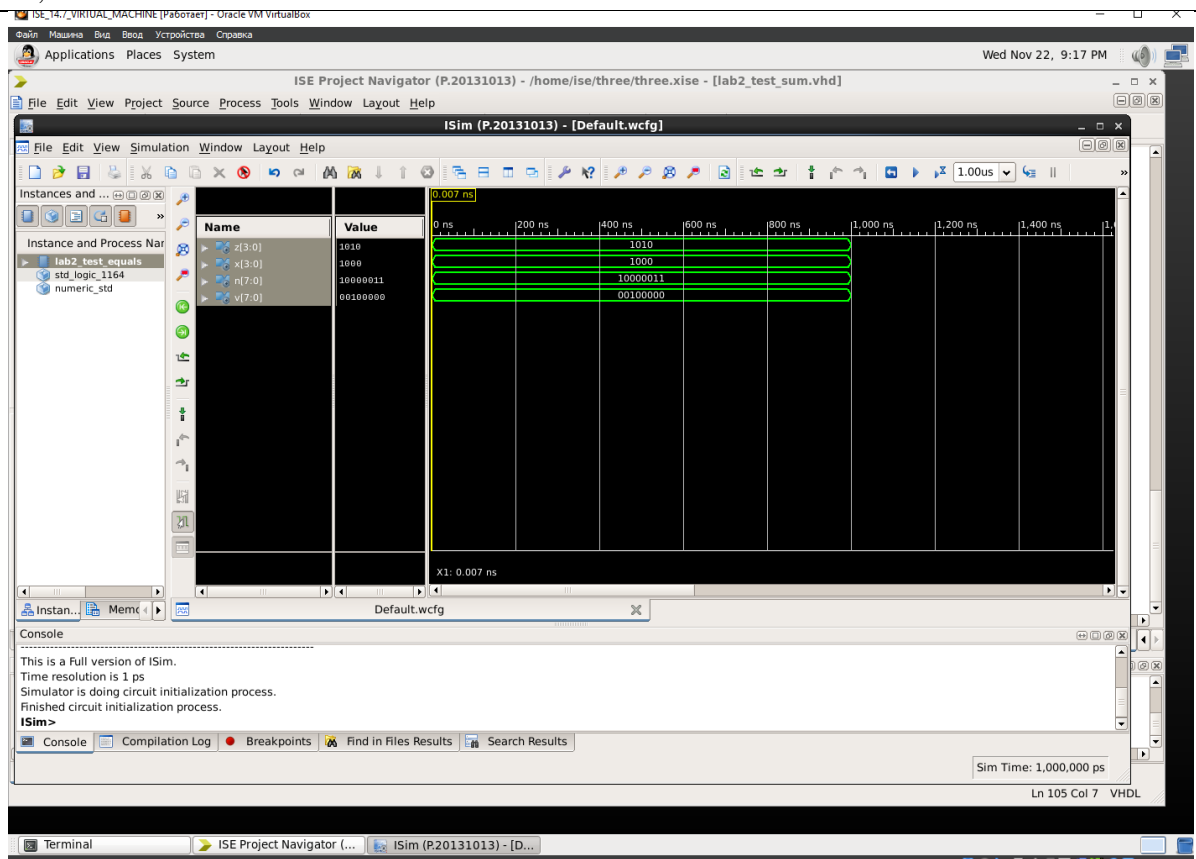


Рисунок 2 –прохождение тестов модуля выравнивания порядков

Листинг 7 – код тестов модуля сложения

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY lab2_test_sum IS
END lab2_test_sum;
ARCHITECTURE behavior OF lab2_test_sum IS
    COMPONENT lab2_sum
    PORT(
        g : IN  std_logic;
        k : IN  std_logic;
        n : IN  std_logic_vector(7 downto 0);
        v : IN  std_logic_vector(7 downto 0);
        sum : OUT std_logic_vector(8 downto 0)    );
    END COMPONENT;
    signal g : std_logic := '0';
    signal k : std_logic := '0';
    signal n : std_logic_vector(7 downto 0) := (others => '0');
    signal v : std_logic_vector(7 downto 0) := (others => '0');
    signal sum : std_logic_vector(8 downto 0);
BEGIN
    uut: lab2_sum PORT MAP (
        g => g,
        k => k,
        n => n,
        v => v,
        sum => sum    );
    stim_proc: process
    begin
        g <= '0';
        k <= '0';
        n <= "10010000";
        v <= "10100111";
        wait for 10 ns;
        g <= '1';
        k <= '1';
        n <= "10010000";
        v <= "10100111";
        wait for 10 ns;
        g <= '0';
        k <= '1';
        n <= "10100110";
        v <= "00000110";
        wait for 10 ns;
        wait;
    end process;
END;
```

Третий скрине

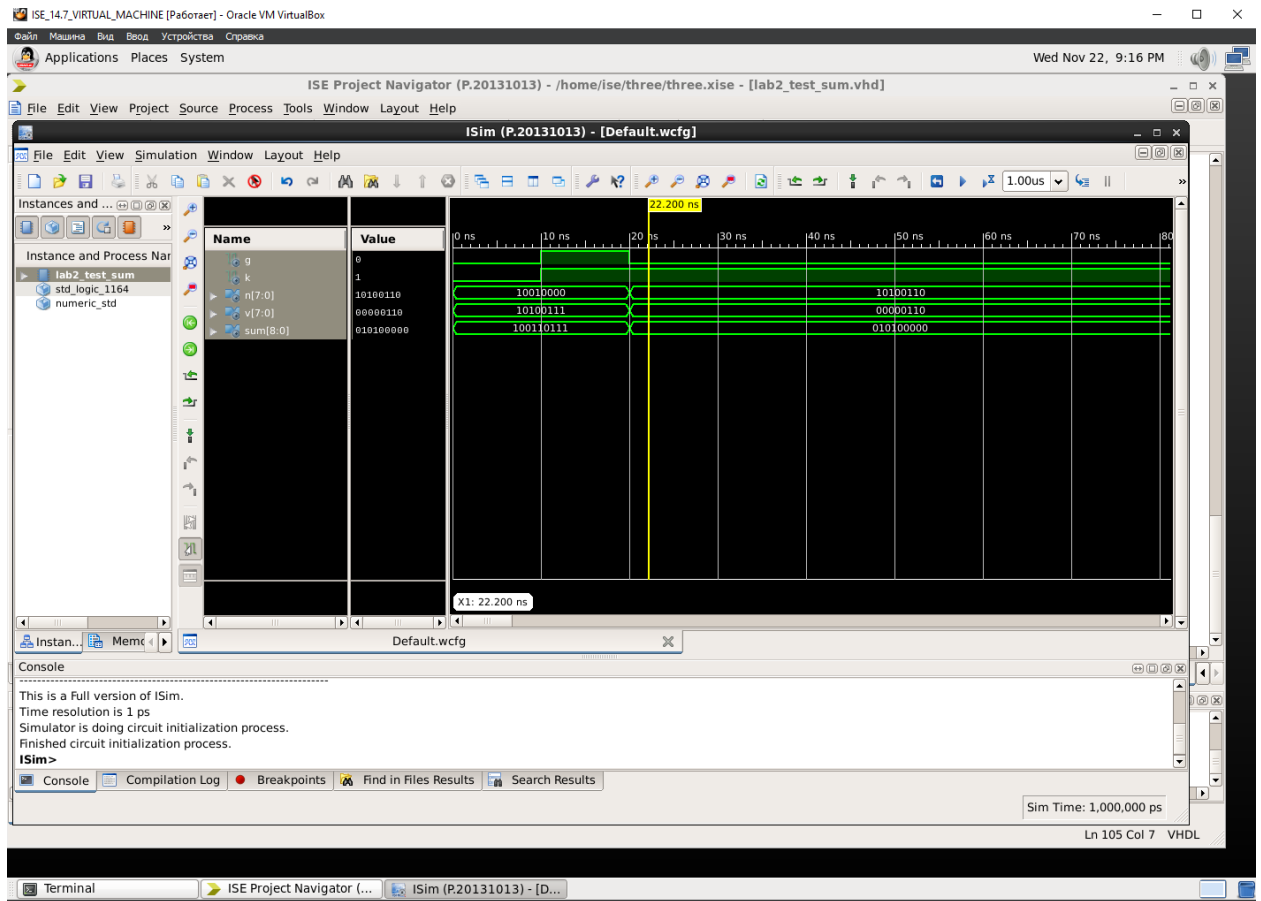


Рисунок 3 – результат прохождения тестов модуля сложения

Листинг 8 – код тестов модуля нормализации

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY lab2_test_sum IS
END lab2_test_sum;
ARCHITECTURE behavior OF lab2_test_sum IS
    COMPONENT lab2_sum
    PORT(
        g : IN  std_logic;
        k : IN  std_logic;
        n : IN  std_logic_vector(7 downto 0);
        v : IN  std_logic_vector(7 downto 0);
        sum : OUT std_logic_vector(8 downto 0) );
    END COMPONENT;
    signal g : std_logic := '0';
    signal k : std_logic := '0';
    signal n : std_logic_vector(7 downto 0) := (others => '0');
    signal v : std_logic_vector(7 downto 0) := (others => '0');
    signal sum : std_logic_vector(8 downto 0);
BEGIN
    uut: lab2_sum PORT MAP (
        g => g,
        k => k,
        n => n,
        v => v,
        sum => sum );

```

```

stim_proc: process
begin
g <= '0';
k <= '0';
n <= "10010000";
v <= "10100111";
wait for 10 ns;
g <= '1';
k <= '1';
n <= "10010000";
v <= "10100111";
wait for 10 ns;
g <= '0';
k <= '1';
n <= "10100110";
v <= "00000110";
wait for 10 ns;
wait;
end process;
END;

```

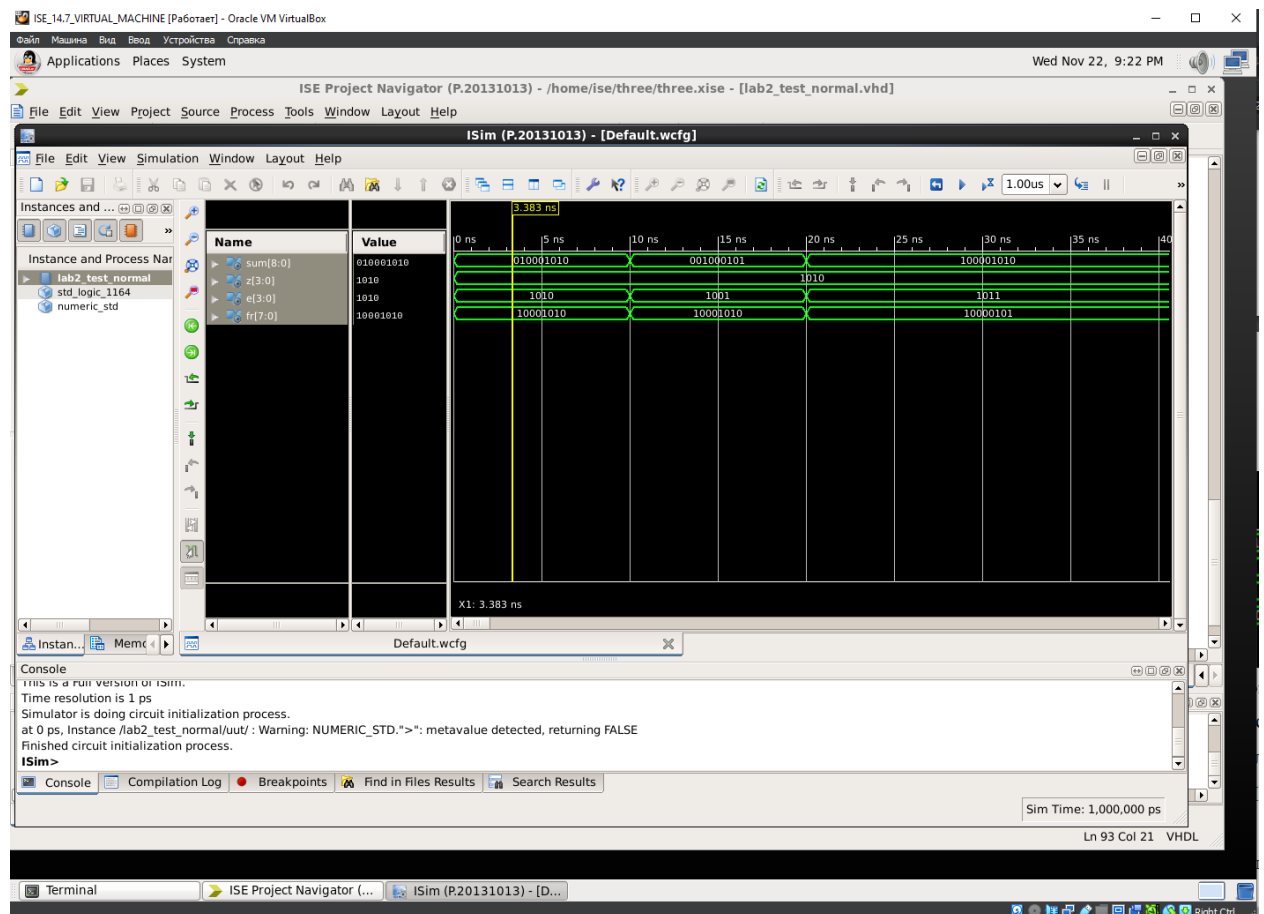


Рисунок 4 –прохождение тестов модуля нормализации

Листинг 9 – код сумматора

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

use IEEE.NUMERIC_STD.ALL;
entity lab2_end is
  Port ( a : in  STD_LOGIC;
        b : in  STD_LOGIC;
        c : in  STD_LOGIC_vector(3 downto 0);
        d : in  STD_LOGIC_vector(3 downto 0);
        e : in  STD_LOGIC_vector(7 downto 0);
        f : in  STD_LOGIC_vector(7 downto 0);
        a_out : out STD_LOGIC;
        c_out : out STD_LOGIC_vector(3 downto 0);
        e_out : out STD_LOGIC_vector(7 downto 0));
end lab2_end;
architecture Behavioral of lab2_end is
  signal g, h : std_logic;
  signal i, j : std_logic_vector(3 downto 0);
  signal m1, m2, m3, m4 : std_logic_vector(7 downto 0);
  signal sum : std_logic_vector(8 downto 0);
  component lab2 is
    Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          c : in  STD_LOGIC_vector(3 downto 0);
          d : in  STD_LOGIC_vector(3 downto 0);
          e : in  STD_LOGIC_vector(7 downto 0);
          f : in  STD_LOGIC_vector(7 downto 0);
          g : out STD_LOGIC;
          h : out STD_LOGIC;
          i : out STD_LOGIC_vector(3 downto 0);
          j : out STD_LOGIC_vector(3 downto 0);
          k : out STD_LOGIC_vector(7 downto 0);
          l : out STD_LOGIC_vector(7 downto 0));
  end component;
  component lab2_2 is
    Port ( z : in  STD_LOGIC_vector(3 downto 0) ;
          x : in  STD_LOGIC_vector(3 downto 0);
          n : in  STD_LOGIC_vector(7 downto 0);
          v : out STD_LOGIC_vector(7 downto 0));
  end component;
  component lab2_sum is
    Port ( g : in  STD_LOGIC;
          k : in  STD_LOGIC;
          n : in  STD_LOGIC_vector (7 downto 0);
          v : in  STD_LOGIC_vector (7 downto 0);
          sum : out STD_LOGIC_vector (8 downto 0));
  end component;
  component lab2_normal is
    Port ( sum : in  STD_LOGIC_vector(8 downto 0);
          z : in  STD_LOGIC_vector(3 downto 0);
          e : out STD_LOGIC_vector(3 downto 0);
          fr : out STD_LOGIC_vector(7 downto 0));
  end component;
begin
  lab2_0: lab2
  port map (

```

```

        c => c,
        d => d,
        e => e,
        f => f,
        a => a,
        b => b,
        g => g,
        h => h,
        i => i,
        i => i,
        k => k,
        k => k );
lab2_2_0: lab2_2
port map (
z => z,
x => x,
n => n,
v => v );
lab2_sum_0: lab2_sum
port map (
g => g,
k => k,
n => n,
v => v,
sum => sum );
lab2_normal_0: lab2_normal
port map (
sum => sum,
z => z,
e => e,
fr => fr );
a_out <= g;
end Behavioral;

```

Написать Test Bench для сумматора, включающий случаи указанные в задании

Было необходимо рассмотреть случаи сложения чисел с одинаковым знаком, чисел с разными знаками, а также случаи в котором пришлось бы выполнять нормализацию (уменьшать или увеличивать порядок после этапа сложения). Выражение, результат в десятичной системе и запись числа в нашем формате для каждого случая представлены в виде смешанного списка списка:

$$711(0.1010.10110001) + 300(0.1001.10010100) = 1011 (0.1010.11111011)$$

$$-39 (1.0110.10011100) + 711(0.1010.10110001) = 672(0.1010.10101000)$$

$666(0.1010.10110100) + 711(0.1010.10110001) = 1377(0.1011.10110010)$

$-711(1.1010.10110001) + 956(0.1010.11101111) = 245(0.1000.11111000)$

Тестовый модуль – Листинг 10

```
-- TestBench Template
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
--USE ieee.numeric_std.ALL;
ENTITY testbench IS
END testbench;
ARCHITECTURE behavior OF testbench IS
-- Component Declaration
    COMPONENT lab2_end
        PORT(
            a_in: in  STD_LOGIC;
            b_in : in  STD_LOGIC;
            c_in : in  STD_LOGIC_vector(3 downto 0);
            d_in : in  STD_LOGIC_vector(3 downto 0);
            e_in : in  STD_LOGIC_vector(7 downto 0);
            f_in : in  STD_LOGIC_vector(7 downto 0);
            a_out : out STD_LOGIC;
            c_out : out STD_LOGIC_vector(3 downto 0);
            e_out : out STD_LOGIC_vector(7 downto 0));

        END COMPONENT;
    signal a_in : std_logic := '0';
        signal b_in : std_logic := '0';
        signal c_in : std_logic_vector(3 downto 0) := (others => '0');
        signal d_in : std_logic_vector(3 downto 0) := (others => '0');
        signal e_in : std_logic_vector(7 downto 0) := (others => '0');
        signal f_in : std_logic_vector(7 downto 0) := (others => '0');

        --Outputs
        signal a_out : std_logic;
        signal c_out : std_logic_vector(3 downto 0);
        signal e_out : std_logic_vector(7 downto 0);
BEGIN
-- Component Instantiation
    uut: lab2_end PORT MAP(
        a_in=>a_in,
        b_in=>b_in,
        c_in=>c_in,
        d_in=>d_in,
        e_in=>e_in,
        f_in=>f_in,
        a_out=>a_out,
        c_out=>c_out,
        e_out=>e_out
    );

-- Test Bench Statements
tb : PROCESS
```

```

BEGIN

    a_in<='0';
    b_in<='0';
    c_in<="1010";
    d_in<="1001";
    e_in<="10110001";
    f_in<="10010100";

wait for 10 ns;

    a_in<='0';
    b_in<='1';
    c_in<="1010";
    d_in<="0110";
    e_in<="10110001";
    f_in<="10011100";

wait for 10 ns;

    a_in<='0';
    b_in<='0';
    c_in<="1010";
    d_in<="1010";
    e_in<="10110001";
    f_in<="10110100";

wait for 10 ns;

    a_in<='1';
    b_in<='0';
    c_in<="1010";
    d_in<="1010";
    e_in<="10110001";
    f_in<="11101111";

wait for 10 ns;
    wait; -- will wait forever
END PROCESS tb;
-- End Test Bench

END;

```

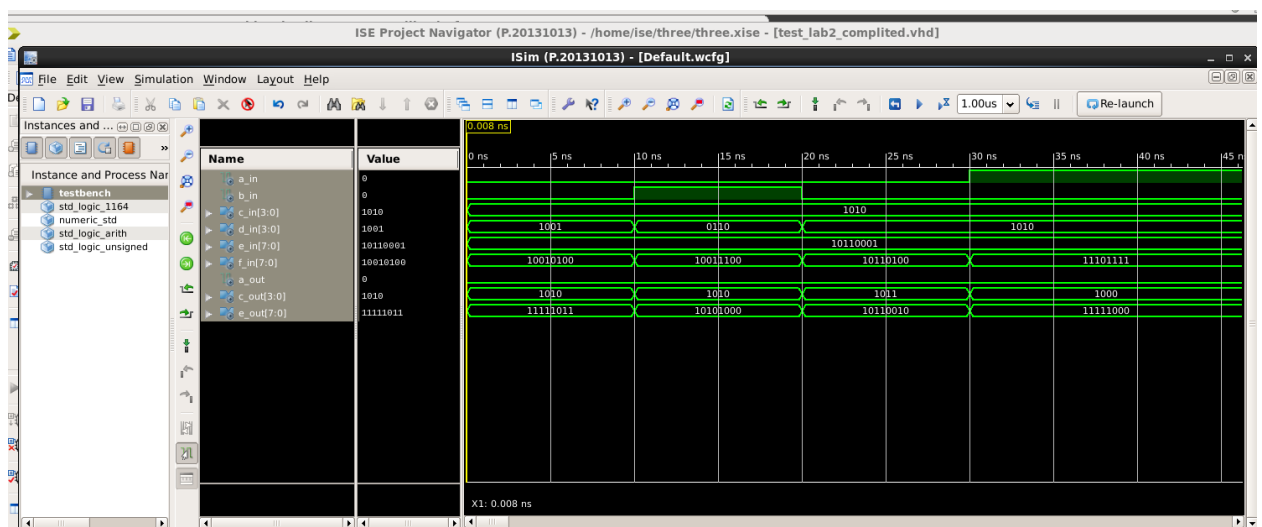


Рисунок 5 –прохождение тестов сумматора

Логически устройство отработало верно, полученные значения совпали с ручными расчетами, необходимо перевести полученные результаты обратно в десятичный вид и сравнить с результатами расчетов в десятичной системе счисления. Результаты представлены в таблице ниже.

Выражение	Результат, в десятичной СС	Результат в формате с плавающей точкой	Результат, переведенный в десятичную СС	Модуль разницы
711(0.1010.10110001))+ 300(0.1001.10010100))-39	1011	0.1010.11111011	1004	7
(1.0110.10011100) + 711(0.1010.10110001))	672	0.1010.10101000	672	0
666(0.1010.10110100))+ 711(0.1010.10110001))	1377	0.1011.10110010	1424	47
-711(1.1010.10110001) + 956(0.1010.11101111))	245	0.1000.11111000	248	3

Вывод

В ходе выполнения лабораторной работы по разработке сумматора чисел с плавающей точкой, были углублены знания языка VHDL и применения инструмента Xilinx ISE. Разработаны необходимые модули для выполнения сложения чисел с плавающей точкой, написаны и успешно пройдены тесты для проверки работоспособности. В конце было выполнено сравнение полученных результатов для оценки точности вычислений с плавающей точкой. Работа выполнена в полном объеме, что подтверждает усвоение принципов построения логических устройств и создания функциональных модулей.