



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра Вычислительной Техники (ВТ)

**ОТЧЕТ
ПО ЛАБОРАТОНОЙ РАБОТЕ №1
по дисциплине**

**«Разработка программно-аппаратного обеспечения информационных и
автоматизированных систем»**

Выполнил студент группы ИКМО-05-23

Миронов Д.С.

Принял старший преподаватель

Унгер А.Ю.

Лабораторная работа
выполнена

« » _____2023г.

«Зачтено»

« » _____2023 г.

Москва 2023

1. Составить для 2-входового декодера с дополнительным входом разрешения таблицу истинности.
2. По таблице истинности составить карту Карно.
3. По карте Карно записать логическую формулу устройства на основе дизъюнктивной нормальной формы (ДНФ).
4. Зарисовать структурную схему устройства, состоящую из базовых логических вентилей (И, ИЛИ, НЕ).
5. Создать проект в интегрированной среде разработки *Xilinx ISE*. Выбрать в качестве целевого устройства ПЛИС начального уровня семейства *Spartan-3*.
6. Написать исходный код на языке VHDL, описывающий работу устройства.
7. Написать тестовый модуль (*Test Bench*) для проверки функционала устройства.
8. Проверить работу устройства визуально по диаграммам переходов, с помощью симулятора *ISim*.
9. На базе 2-входового декодера собрать 3-входовый декодер с дополнительным входом разрешения. 9. На базе 3-входового декодера собрать 4-входовый декодер с дополнительным входом разрешения.
10. Зарисовать структурную схему.
11. Описать схему на языке VHDL с тремя модулями: 2-входовой декодер (1) и 3-входовый декодер (2) и 4-входовый декодер.
12. Проверить работу устройства визуально с помощью диаграмм переходов в симуляторе *ISim*.
13. Написать вывод о проделанной работе.

1. Составить для 2-входового декодера с дополнительным входом разрешения таблицу истинности.

s	a	b	Y1	Y2	Y3	Y4
0	0	0	0	0	0	0
1	0	0	1	0	0	0
0	0	1	0	0	0	0
1	0	1	0	1	0	0
0	1	0	0	0	0	0
1	1	0	0	0	1	0
0	1	1	0	0	0	0
1	1	1	0	0	0	1

2. По таблице истинности составить карту Карно.

3. По карте Карно записать логическую формулу устройства на основе дизъюнктивной нормальной формы (ДНФ).

$$y_0 = \bar{a} * \bar{b}$$

$$y_1 = \bar{a} * b$$

$$y_2 = a * \bar{b}$$

$$y_3 = a * b$$

4. Зарисовать структурную схему устройства, состоящую из базовых логических вентилей (И, ИЛИ, НЕ).

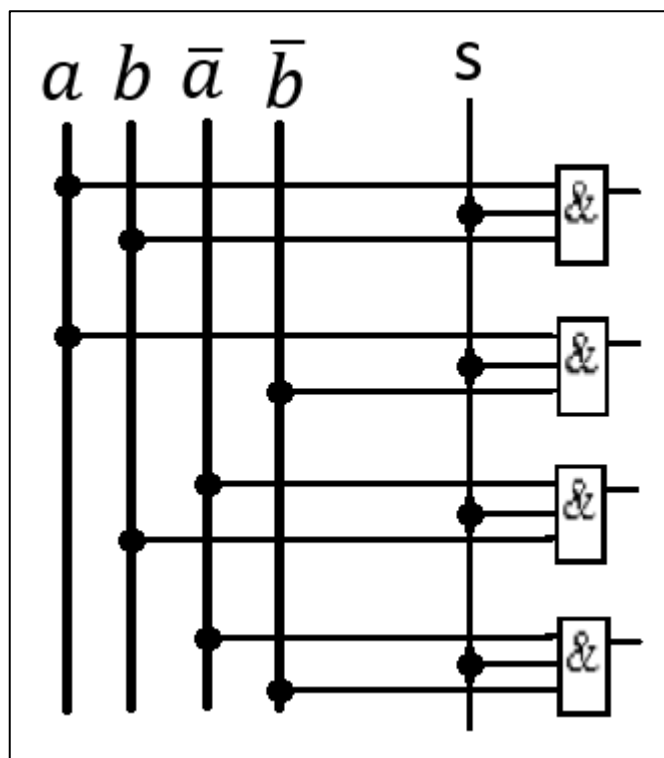


Рисунок 1 – Структурная схема

5. Создать проект в интегрированной среде разработки Xilinx ISE. Выбрать в качестве целевого устройства ПЛИС начального уровня семейства Spartan-3.

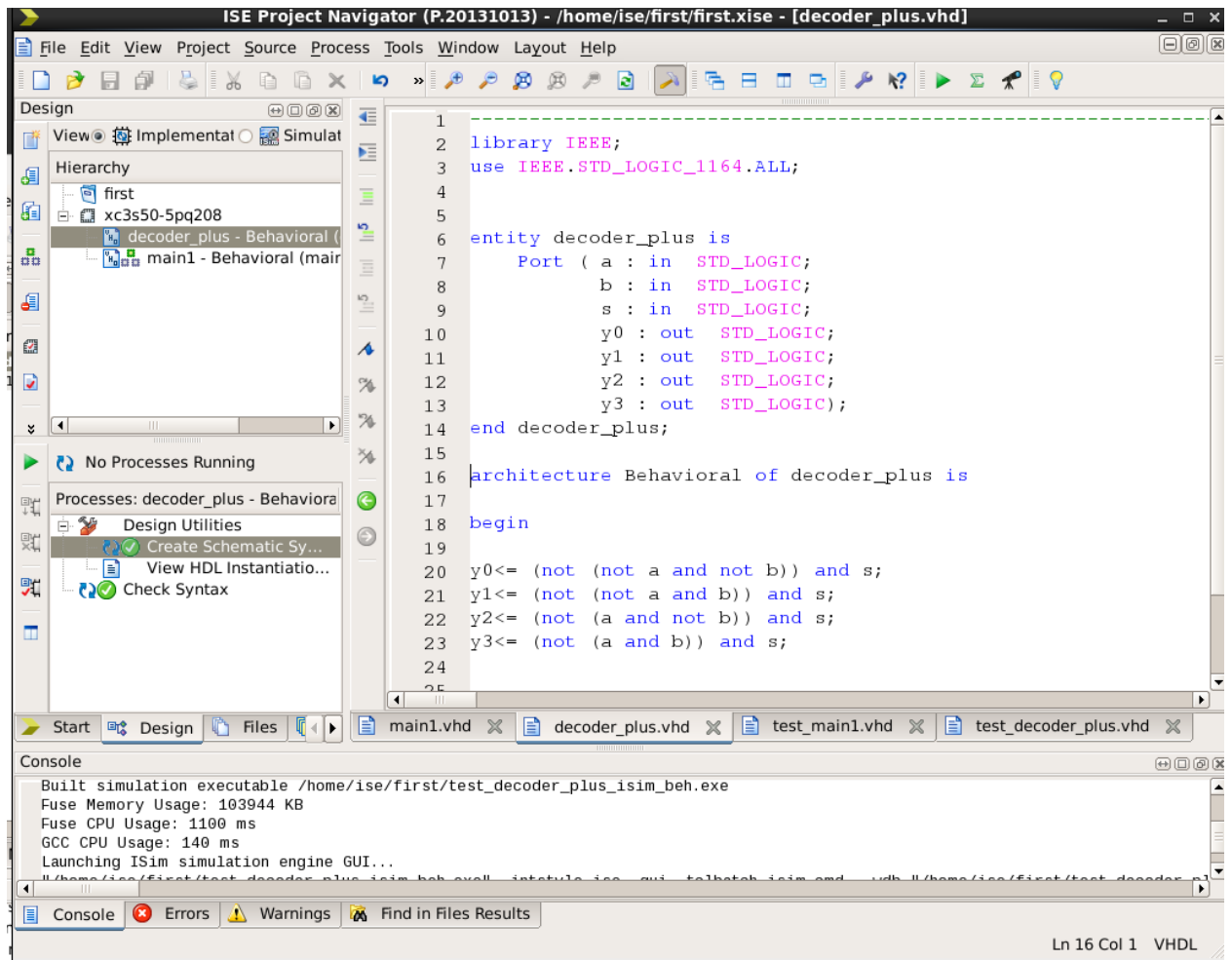


Рисунок 2 – Создан проект в интегрированной среде разработки Xilinx ISE

6. Написать исходный код на языке VHDL, описывающий работу устройства.

```
5
6 entity decoder_plus is
7     Port ( a : in  STD_LOGIC;
8           b : in  STD_LOGIC;
9           s : in  STD_LOGIC;
10          y0 : out STD_LOGIC;
11          y1 : out STD_LOGIC;
12          y2 : out STD_LOGIC;
13          y3 : out STD_LOGIC);
14 end decoder_plus;
15
16 architecture Behavioral of decoder_plus is
17
18 begin
19
20 y0<= (not (not a and not b)) and s;
21 y1<= (not (not a and b)) and s;
22 y2<= (not (a and not b)) and s;
23 y3<= (not (a and b)) and s;
24
25
26 end Behavioral;
27
28
```

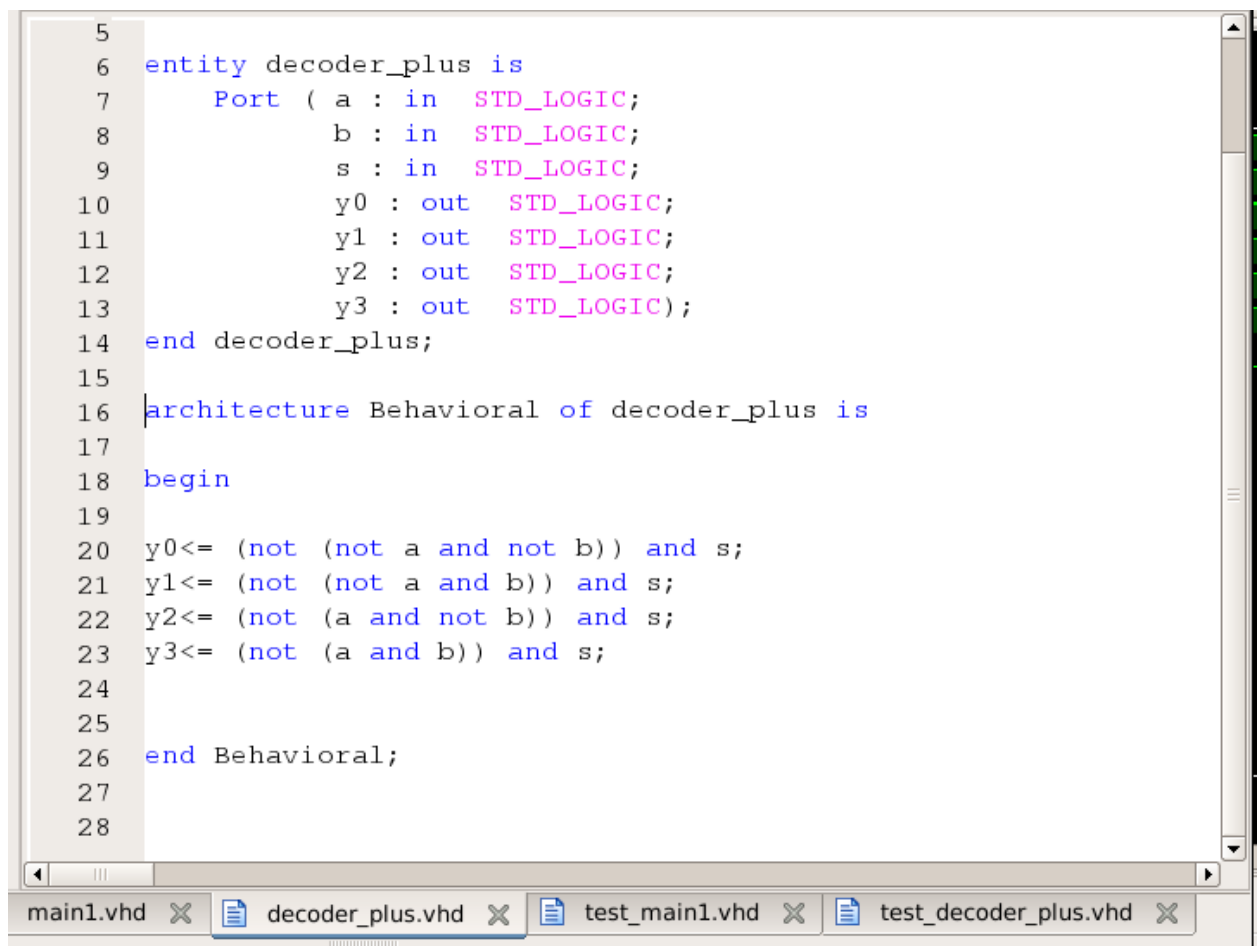


Рисунок 3 – Написан исходный код на языке VHDL

7. Написать тестовый модуль (Test Bench) для проверки функционала устройства.

```
65     begin
66         -- hold reset state for 100 ns.
67         wait for 100 ns;
68
69         a <='0';
70         b <='0';
71         s <='1';
72         wait for 100 ns;
73
74         a <='0';
75         b <='1';
76         s <='1';
77         wait for 100 ns;
78
79         a <='1';
80         b <='0';
81         s <='1';
82         wait for 100 ns;
83
84         a <='1';
85         b <='1';
86         s <='1';
87         wait for 100 ns;
88
89
```

main1.vhd ✕ decoder_plus.vhd ✕ test_main1.vhd ✕ test_decoder_plus.vhd ✕

Рисунок 4 – Написан тестовый модуль

```
89
90  -----
91  a <='0';
92  b <='0';
93  s <='0';
94  wait for 100 ns;
95
96  a <='0';
97  b <='1';
98  s <='0';
99  wait for 100 ns;
100
101  a <='1';
102  b <='0';
103  s <='0';
104  wait for 100 ns;
105
106  a <='1';
107  b <='1';
108  s <='0';
109  wait for 100 ns;
110
111
112  -- insert stimulus here
113
```

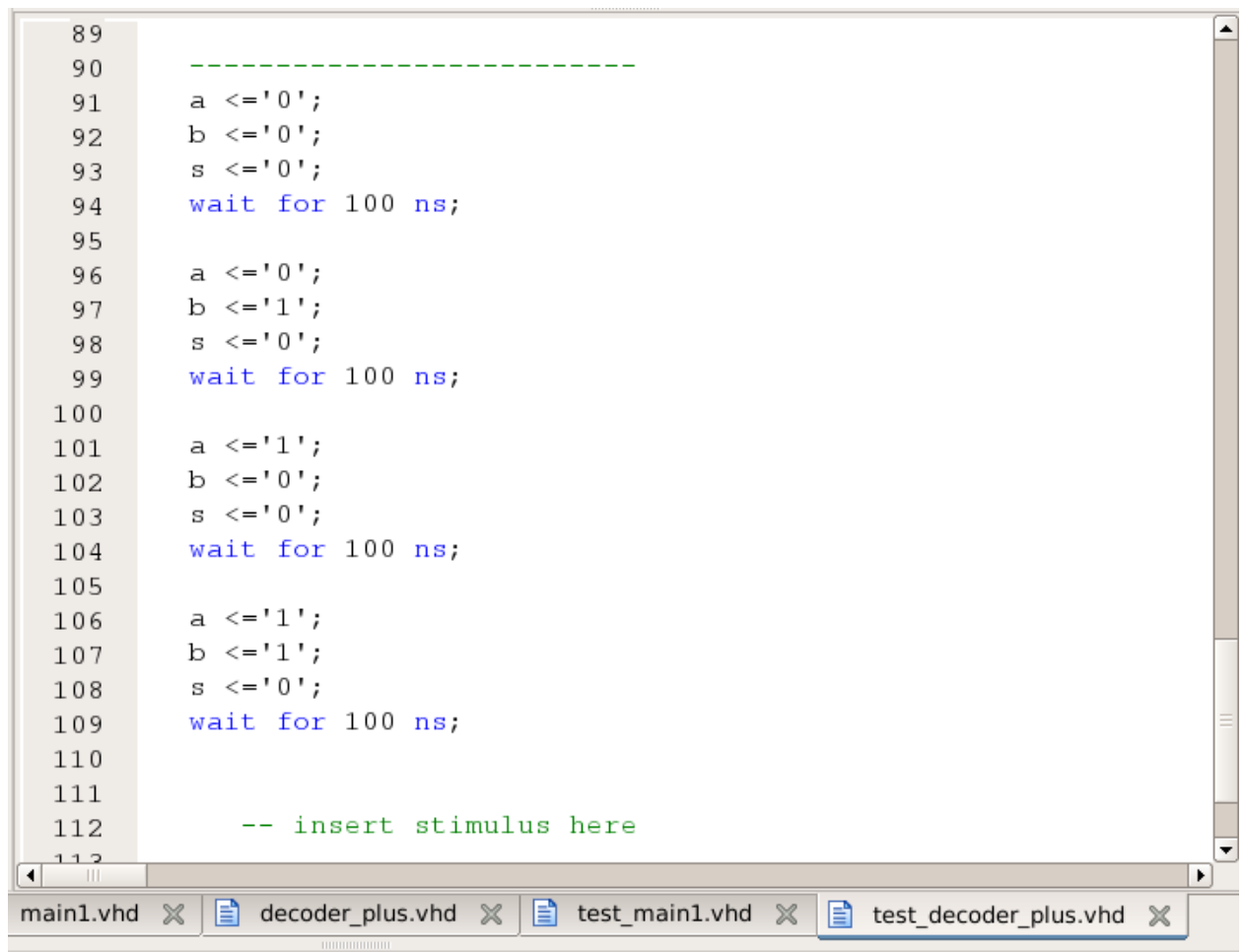


Рисунок 5 – Написан тестовый модуль

8. Проверить работу устройства визуально по диаграммам переходов, с помощью симулятора ISim.

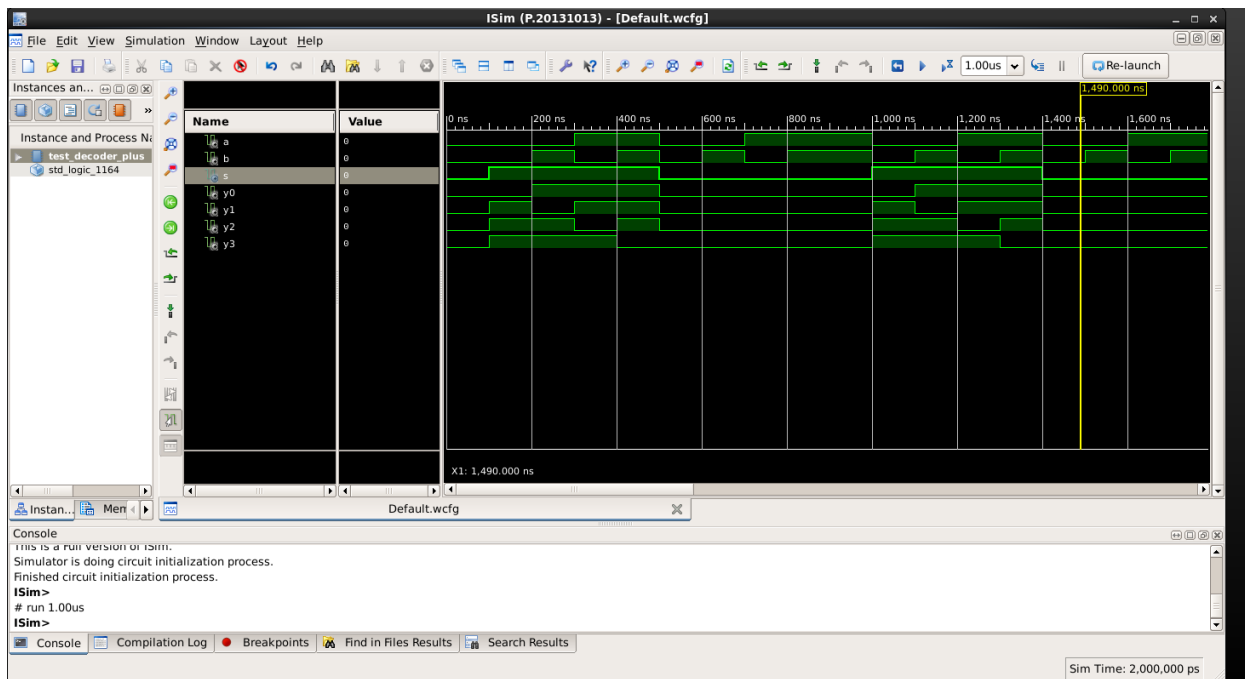


Рисунок 6 – Проверена работа устройства визуально по диаграммам переходов

9. На базе 2-входового декодера собрать 3-входовый декодер с дополнительным входом разрешения. 9. На базе 3-входового декодера собрать 4-входовый декодер с дополнительным входом разрешения. Зарисовать структурную схему.

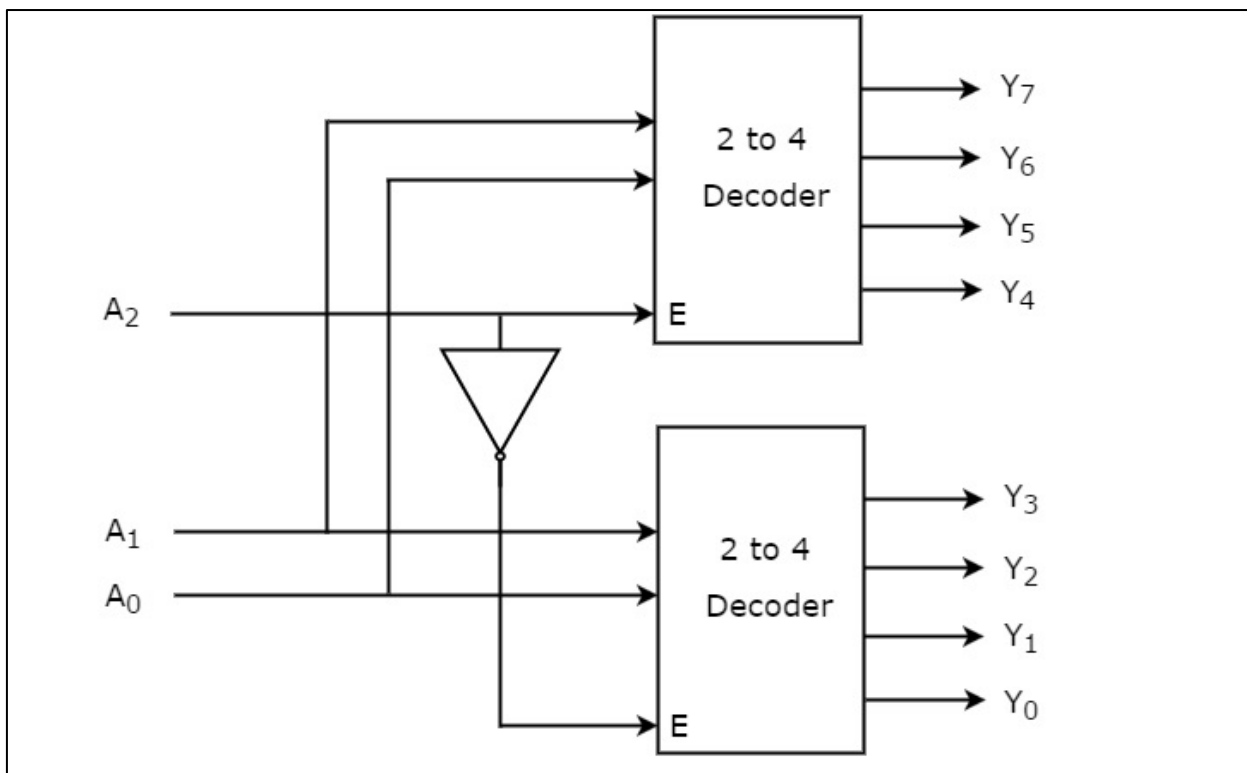


Рисунок 7 – Структурная схема 3-8

11. Описать схему на языке VHDL с тремя модулями: 2-входовой декодер (1) и 3-входовой декодер (2) и 4-входовой декодер.

Листинг описывающий схему декодера 3 на 8

```
entity decoder3 is
    Port ( a3 : in  STD_LOGIC_VECTOR (2 downto 0);
           b3 : out STD_LOGIC_VECTOR (7 downto 0);
           enable : in  STD_LOGIC);
end decoder3;
architecture Behavioral of decoder3 is
    component decoder2 is
        Port ( en : in  STD_LOGIC;
              a   : in  STD_LOGIC_VECTOR (1 downto 0);
              b   : out STD_LOGIC_VECTOR(3 downto 0));
    end component;

    signal m0: STD_LOGIC;
    signal m1: STD_LOGIC;
begin
    U1: decoder2 Port map(m1,a3(1 downto 0),b3(7 downto 4));
    U2: decoder2 Port map(m0,a3(1 downto 0),b3(3 downto 0));
    m0 <= NOT a3(2) AND enable;
    m1 <= a3(2) AND enable;
end Behavioral;
```

Листинг теста декодера 3 на 8

```
BEGIN

    uut: decoder3 PORT MAP (
        a3 => test_seq(2 downto 0),
        b3 => b3,
        enable => test_seq(3)
    );

    -- Stimulus process
    stim_proc: process
    begin
        -- hold reset state for 100 ns.
        test_seq<="0000";

        wait for 100ns;
        test_seq<="0001";
        wait for 100ns;
        test_seq<="0010";
        wait for 100ns;
        test_seq<="0011";
        wait for 100ns;
        test_seq<="1000";
        wait for 100ns;
        test_seq<="1001";
        wait for 100ns;
        test_seq<="1010";
        wait for 100ns;
        test_seq<="1011";
        wait for 100ns;
        test_seq<="1100";
        wait for 100ns;
        test_seq<="1101";
```

Продолжение листинга теста декодера 3 на 8

```
wait for 100ns;  
test_seq<="1110";  
wait for 100ns;  
test_seq<="1111";  
wait;  
    wait;  
end process;  
END;
```

12. Проверить работу устройства визуально с помощью диаграмм переходов в симуляторе ISim.

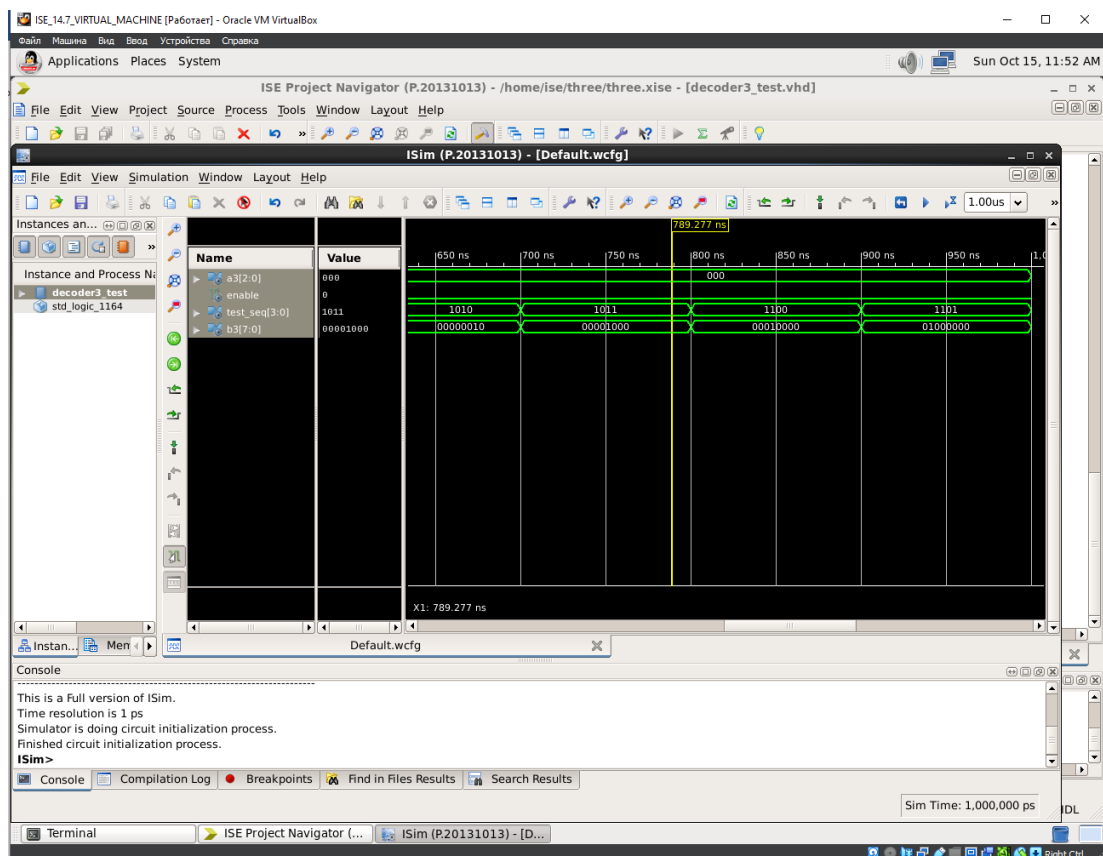


Рисунок 10 – Проверена работу устройства визуально с помощью диаграмм переходов

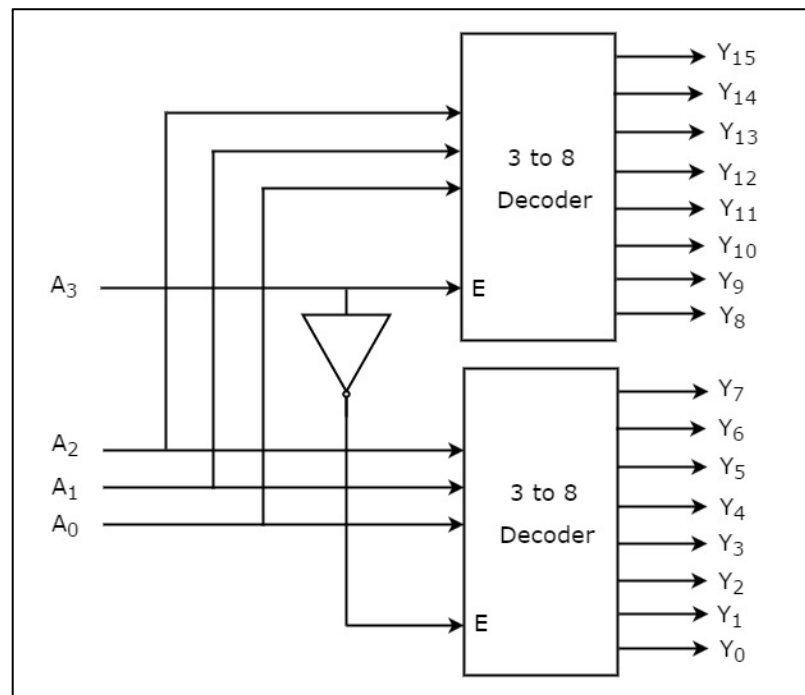


Рисунок 11 – Структурная схема 4-16

Листинг описывающий схему декодера 4 на 16

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

entity decoder4 is
    Port ( a4 : in  STD_LOGIC_VECTOR (3 downto 0);
           b4 : out STD_LOGIC_VECTOR (15 downto 0);
           enable4 : in  STD_LOGIC);
end decoder4;

architecture Behavioral of decoder4 is

    component decoder3 is
        Port ( enable : in  STD_LOGIC;
              a3: in  STD_LOGIC_VECTOR (2 downto 0);
              b3: out STD_LOGIC_VECTOR(7 downto 0));
    end component;

    signal m0: STD_LOGIC;
    signal m1: STD_LOGIC;
begin
```

```

U1: decoder3 Port map(m1,a4(2 downto 0),b4(15 downto 8));
U2: decoder3 Port map(m0,a4(2 downto 0),b4(7 downto 0));

m0 <= NOT a4(2) AND enable4;

m1 <= a4(2) AND enable4;

end Behavioral;

```

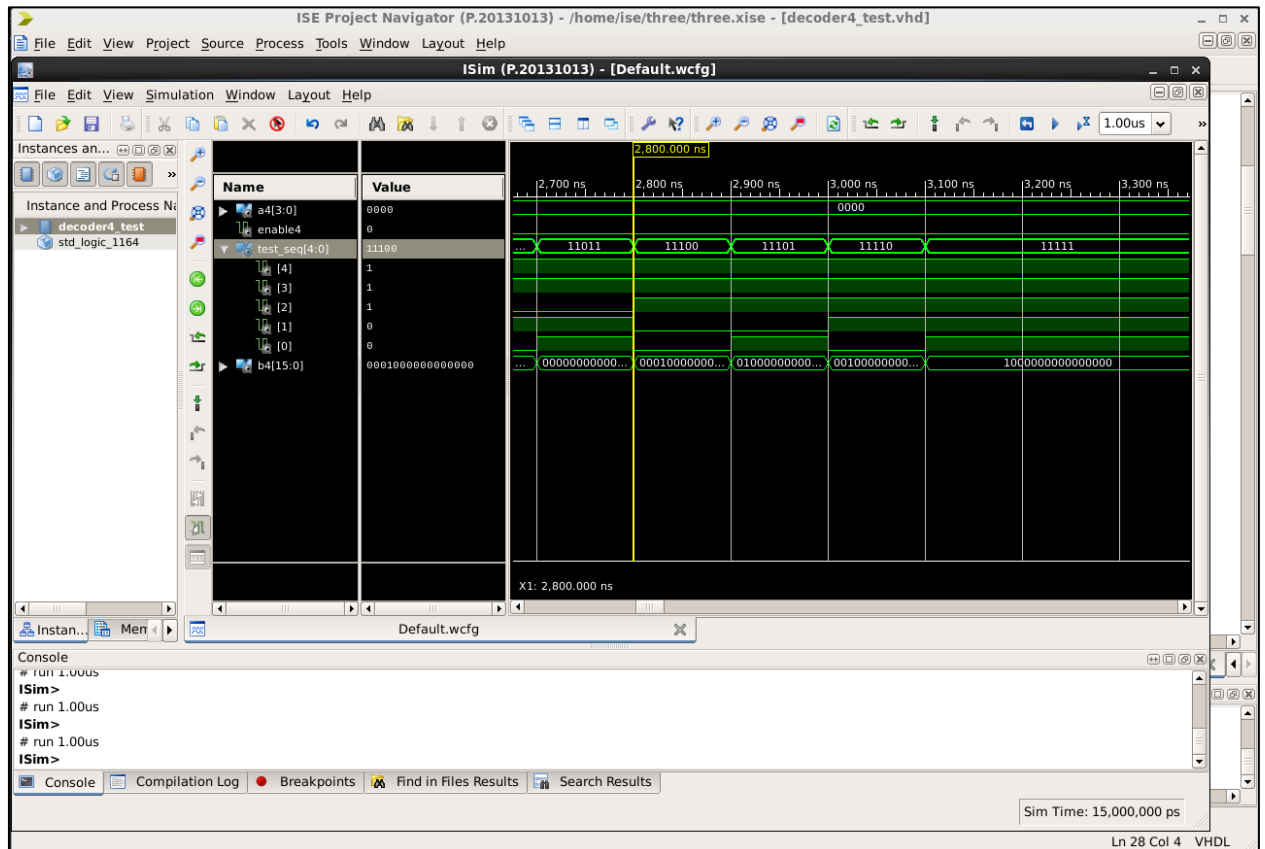


Рисунок – Проверить работу устройства визуально с помощью диаграмм переходов

Вывод: Разработали 4-входовый двоичный декодер с дополнительным входом разрешения, составили таблицу истинности, написали исходный код на языке VHDL, написали тестовый модуль. Проверили работу устройства визуально по диаграммам переходов, с помощью симулятора ISim. На базе 2-входового декодера собрали 3-входовый декодер с дополнительным входом разрешения. На базе 3-входового декодера собрали 4-входовый декодер с дополнительным входом разрешения. Зарисовали структурную схему. Описали схему на языке VHDL с тремя модулями: 2-входовой декодер (1) и 3-входовый декодер

(2) и 4-входовый декодер. Проверили работу устройства визуально с помощью диаграмм переходов в симуляторе ISim.