Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №2

по дисциплине «Проектирование программного обеспечения для устройств на ПЛИС»

на тему «Знакомство с языком VHDL»

Выполнили студенты группы 22ВВП1

Хоссейни Нежад С.А.С.М.

Беляев Д. И.

Демин М.

Приняли:

Федюнин Р.Н.

Гурин Е.И.

Пенза 2024

**Название**

Знакомство с языком VHDL

**Цель работы**

Изучение принципов разработки схем на языке VHDL.

**Ход работы**

1 Моделирование простейших логических схем.

Создать новый проект, при создании задать Family Spartan3.

1.1 Для заданной в табл. 1 функции на языке VHDL разработать схему, проверить правильность ее функционирования с помощью функционального моделирования (Behavioral Simulation). Выполнить моделирование с помощью временного моделирования (Timing Simulation). Сравнить результаты моделирования в различных режимах. Синтезировать схему (см. приложение).



library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity list1 is

Port (x1, x2, x3 : in STD\_LOGIC;

y : out STD\_LOGIC);

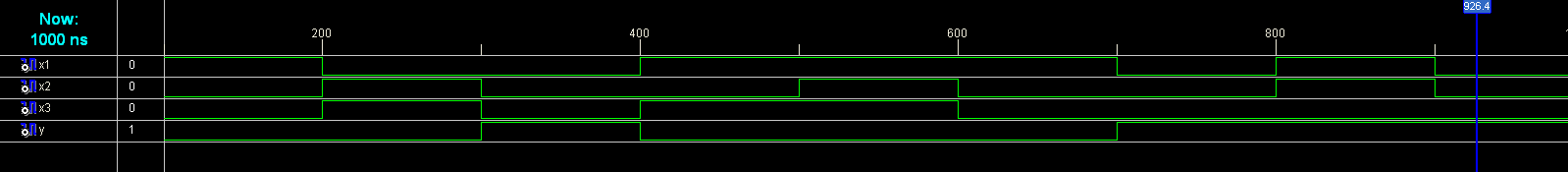
end list1;

architecture Behavioral of list1 is

begin

y <= ((not x1) and (not x2)) or (x1 and x2 and (not x3)) or ((not x1) and x2 and (not x3));

end Behavioral;



1.2 В разработанную в п. 1.1 схему добавить триггер на выходе, проверить работу полученной схемы с помощью функционального моделирования (Behavioral Simulation) и временного моделирования (Timing Simulation). Сравнить результаты п. 1.1 и п. 1.2, а также при различных видах моделирования. Синтезировать схему (см. приложение).

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity list1 is

Port (clk, x1, x2, x3 : in STD\_LOGIC;

y : out STD\_LOGIC);

end list1;

architecture Behavioral of list1 is

begin

process (clk) begin

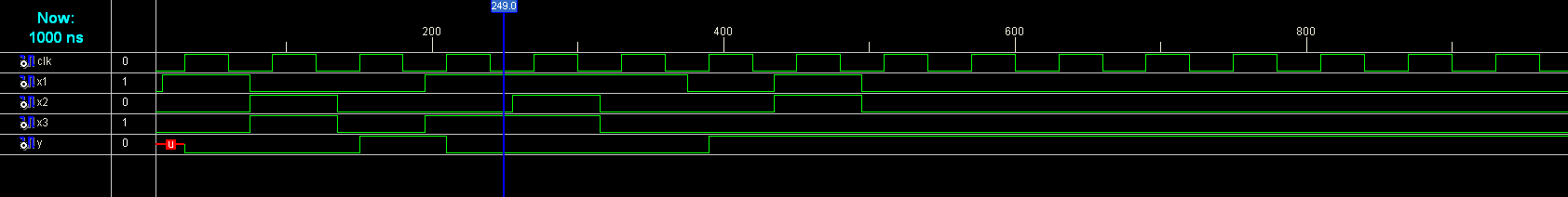
if clk='1' and clk'event then

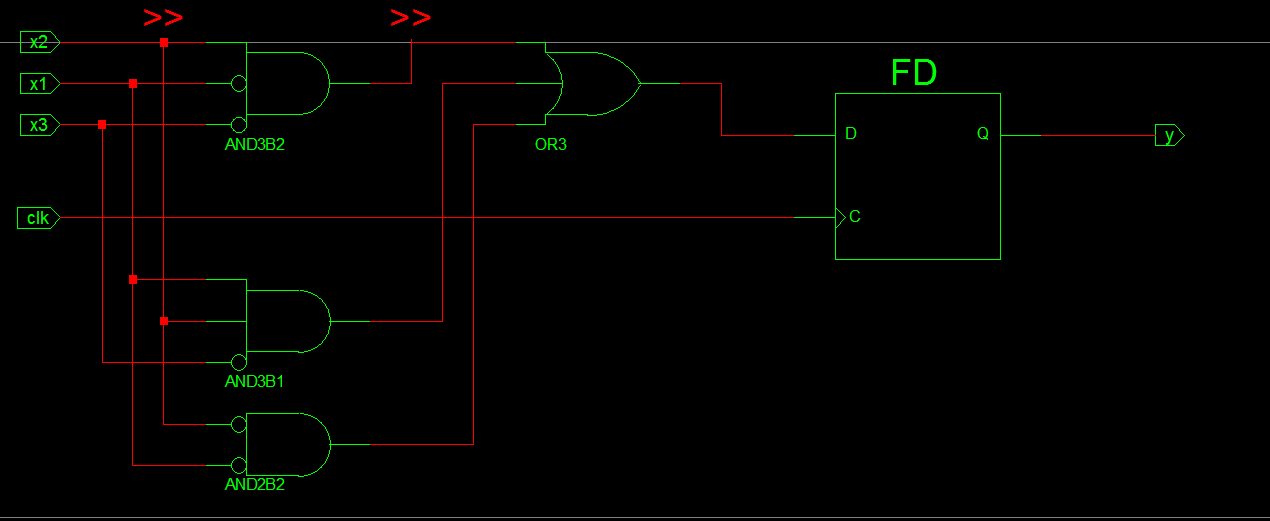
y <= ((not x1) and (not x2)) or (x1 and x2 and (not x3)) or ((not x1) and x2 and (not x3));

end if;

end process;

end Behavioral;





1.3 В разработанную в п. 1.2 схему добавить триггеры на входах. Проверить работу полученной схемы с помощью функционального моделирования (Behavioral Simulation). Сравнить результаты при различных видах моделирования в п. 1.1, п. 1.2 и п. 1.3. Синтезировать схему (см. приложение).

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity list13 is

Port (clk, x1, x2, x3 : in STD\_LOGIC;

y : out STD\_LOGIC);

end list13;

architecture Behavioral of list13 is

signal z1, z2, z3 : STD\_LOGIC;

begin

process (clk) begin

if clk='1' and clk'event then

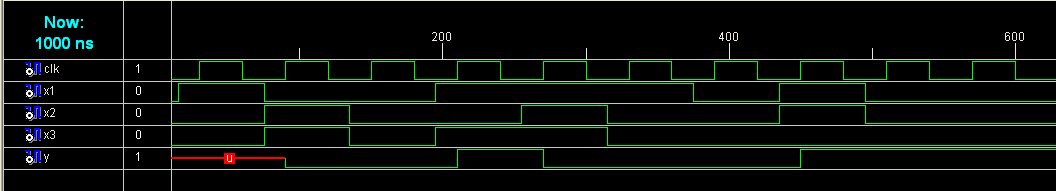
z1 <= x1; z2 <= x2; z3 <= x3;

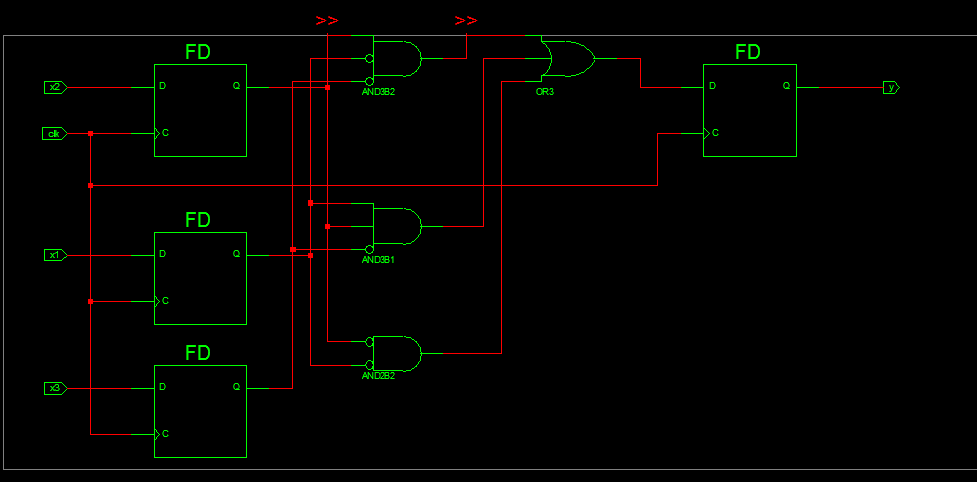
y <= ((not z1) and (not z2)) or (z1 and z2 and (not z3)) or ((not z1) and z2 and (not z3));

end if;

end process;

end Behavioral;

****

****

2 На языке VHDL разработать схему одноразрядного комбинационного сумматора. Проверить работу полученной схемы с помощью функционального моделирования (Behavioral Simulation). Проверить работу полученной схемы с помощью временного моделирования (Timing Simulation). Сравнить результаты моделирования в различных режимах.

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity list2 is

Port (

A : in STD\_LOGIC;

B : in STD\_LOGIC;

Cin : in STD\_LOGIC;

S : out STD\_LOGIC;

Cout : out STD\_LOGIC

);

end list2;

architecture Behavioral of list2 is

begin

process (A, B, Cin)

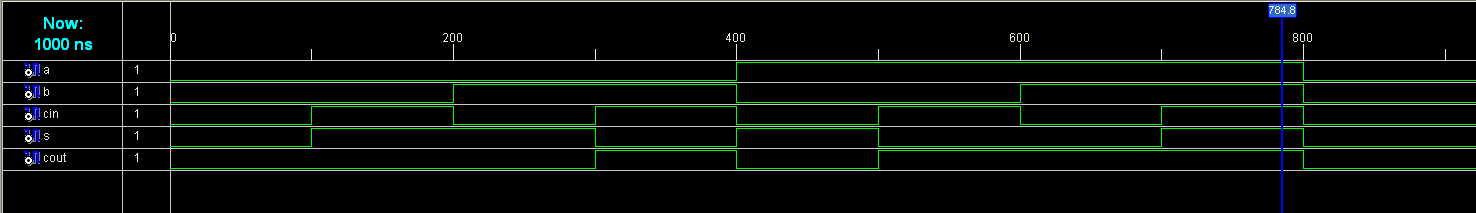
begin

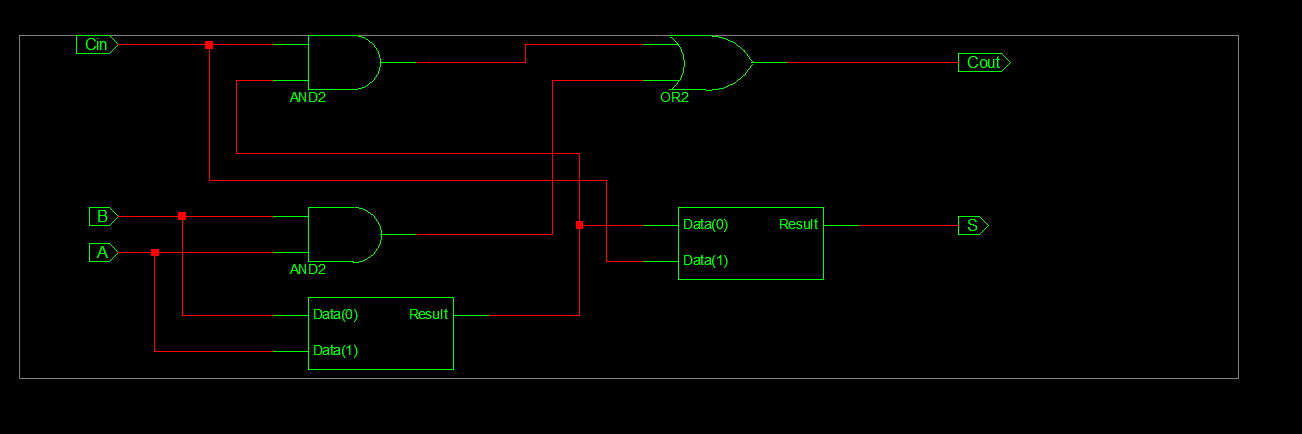
S <= A XOR B XOR Cin;

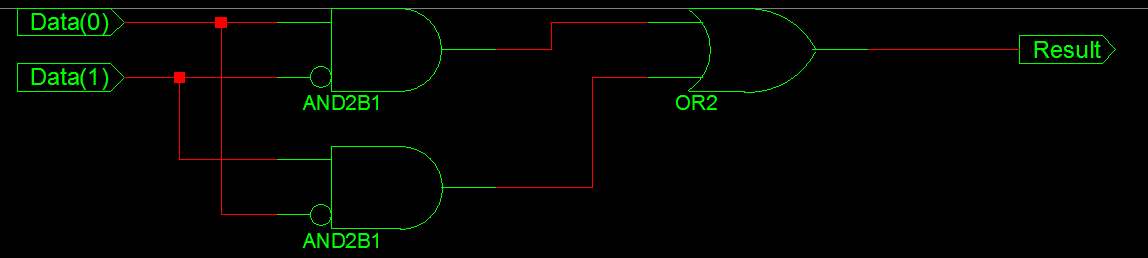
Cout <= (A AND B) OR ((A XOR B) AND Cin);

end process;

end Behavioral;

****

****

****

По согласованию с преподавателем на базе одноразрядного сумматора разработать 2-разрядный сумматор. Проверить работу полученной схемы с помощью функционального и временного моделирования.

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity list22 is

Port (

A : in STD\_LOGIC\_VECTOR(1 downto 0);

B : in STD\_LOGIC\_VECTOR(1 downto 0);

Cin : in STD\_LOGIC;

S : out STD\_LOGIC\_VECTOR(1 downto 0);

Cout : out STD\_LOGIC

);

end list22;

architecture Behavioral of list22 is

signal C0 : STD\_LOGIC;

begin

-- Младший разряд

process (A, B, Cin)

begin

S(0) <= A(0) XOR B(0) XOR Cin;

C0 <= (A(0) AND B(0)) OR (Cin AND (A(0) XOR B(0)));

end process;

-- Старший разряд

process (A, B, C0)

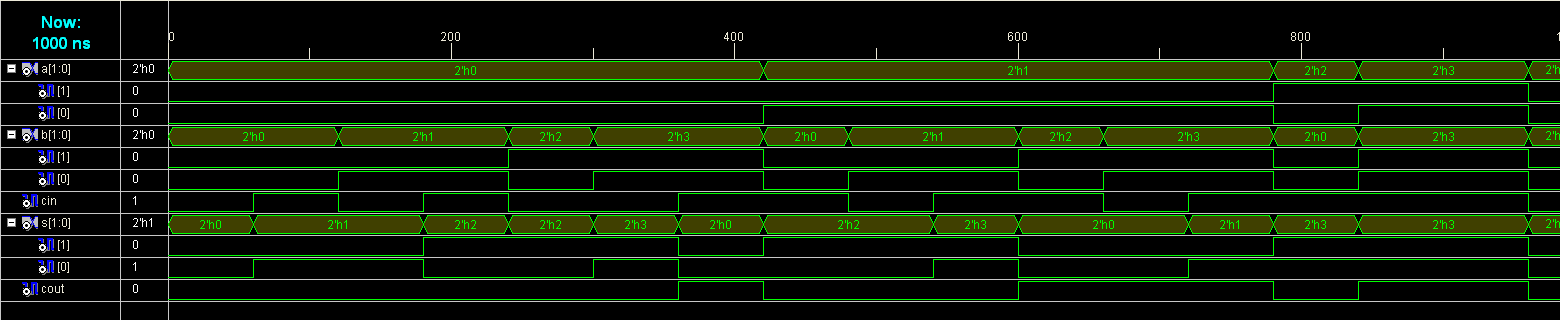
begin

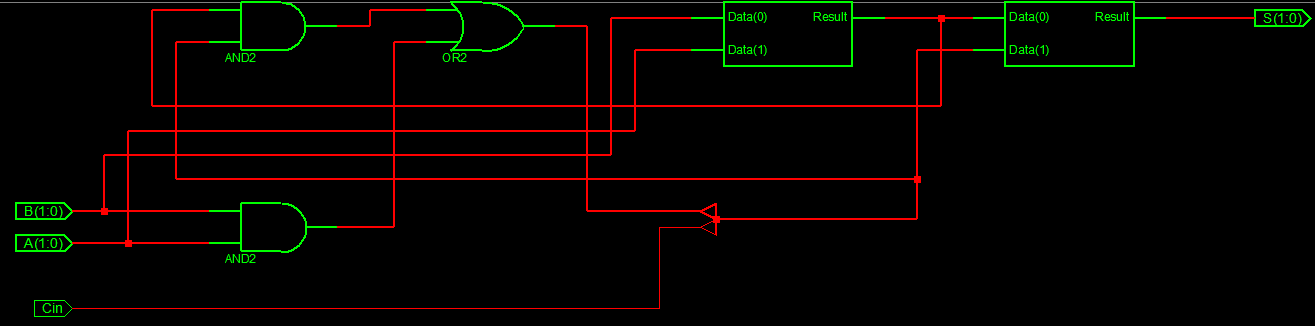
S(1) <= A(1) XOR B(1) XOR C0;

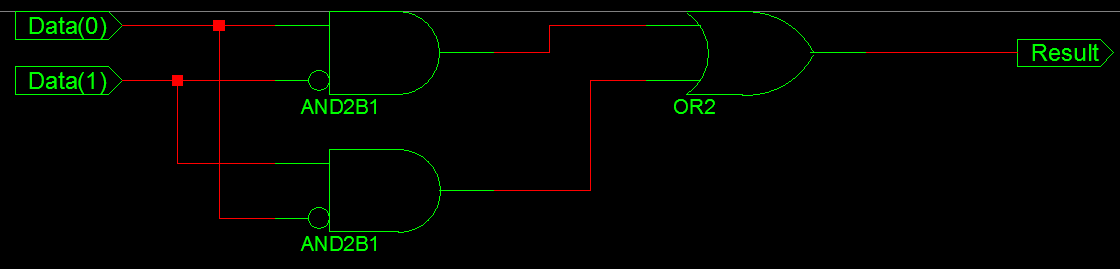
Cout <= (A(1) AND B(1)) OR (C0 AND (A(1) XOR B(1)));

end process;

end Behavioral;

****

****

****

3 На языке VHDL разработать схему разработать схему дешифратора на два входа. Проверить работу полученной схемы с помощью функционального моделирования (Behavioral Simulation).

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity list3 is

Port (

A1 : in STD\_LOGIC;

A0 : in STD\_LOGIC;

D : out STD\_LOGIC\_VECTOR(3 downto 0)

);

end list3;

architecture Behavioral of list3 is

begin

process (A0, A1) begin

D(0) <= (not A1) and (not A0);

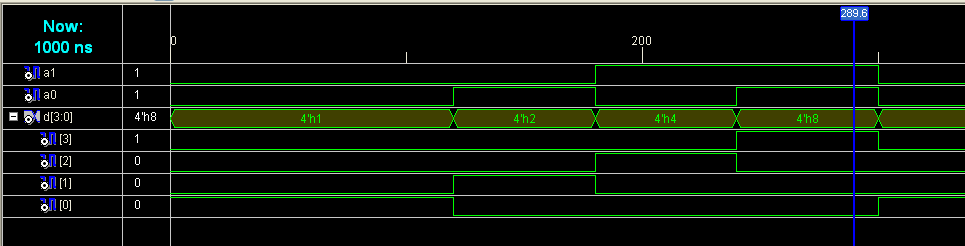
D(1) <= (not A1) and A0;

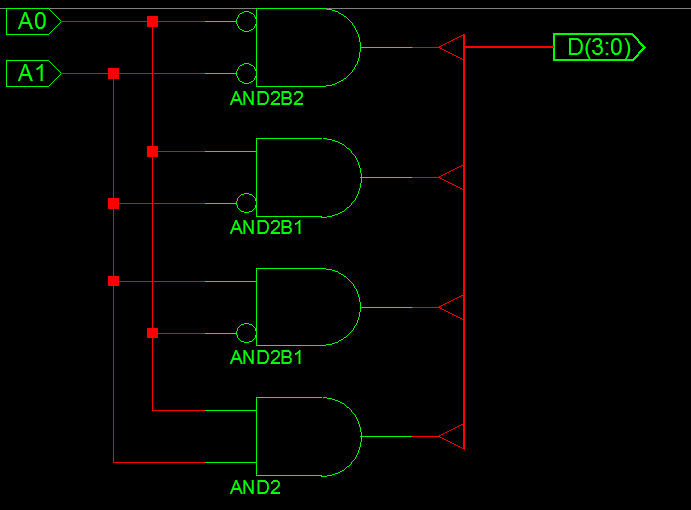
D(2) <= A1 and (not A0);

D(3) <= A1 and A0;

end process;

end Behavioral;

****

****

4 На языке VHDL разработать схему разработать схему одноразрядного мультиплексора на два входа. Проверить работу полученной схемы с помощью функционального моделирования (Behavioral Simulation).

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity list4 is

Port (

A1 : in STD\_LOGIC;

A0 : in STD\_LOGIC;

S : in STD\_LOGIC;

Y : out STD\_LOGIC

);

end list4;

architecture Behavioral of list4 is

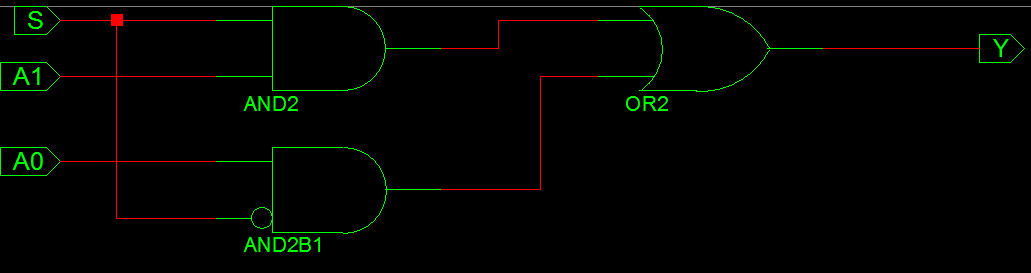
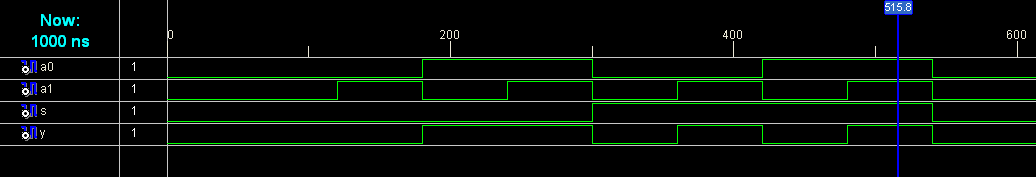
begin

process (A0, A1, S) begin

Y <= (A0 and (not S)) or (A1 and S);

end process;

end Behavioral;



5 На языке VHDL разработать схему дешифратора на четыре входа. Проверить работу полученной схемы с помощью функционального моделирования (Behavioral Simulation). Проверить работу полученной схемы с помощью временного моделирования (Post-Map).

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity list5 is

Port (

A : in STD\_LOGIC\_VECTOR(3 downto 0);

Y : out STD\_LOGIC\_VECTOR(15 downto 0)

);

end list5;

architecture Behavioral of list5 is

begin

process (A)

begin

Y(0) <= not A(3) and not A(2) and not A(1) and not A(0);

Y(1) <= not A(3) and not A(2) and not A(1) and A(0);

Y(2) <= not A(3) and not A(2) and A(1) and not A(0);

Y(3) <= not A(3) and not A(2) and A(1) and A(0);

Y(4) <= not A(3) and A(2) and not A(1) and not A(0);

Y(5) <= not A(3) and A(2) and not A(1) and A(0);

Y(6) <= not A(3) and A(2) and A(1) and not A(0);

Y(7) <= not A(3) and A(2) and A(1) and A(0);

Y(8) <= A(3) and not A(2) and not A(1) and not A(0);

Y(9) <= A(3) and not A(2) and not A(1) and A(0);

Y(10) <= A(3) and not A(2) and A(1) and not A(0);

Y(11) <= A(3) and not A(2) and A(1) and A(0);

Y(12) <= A(3) and A(2) and not A(1) and not A(0);

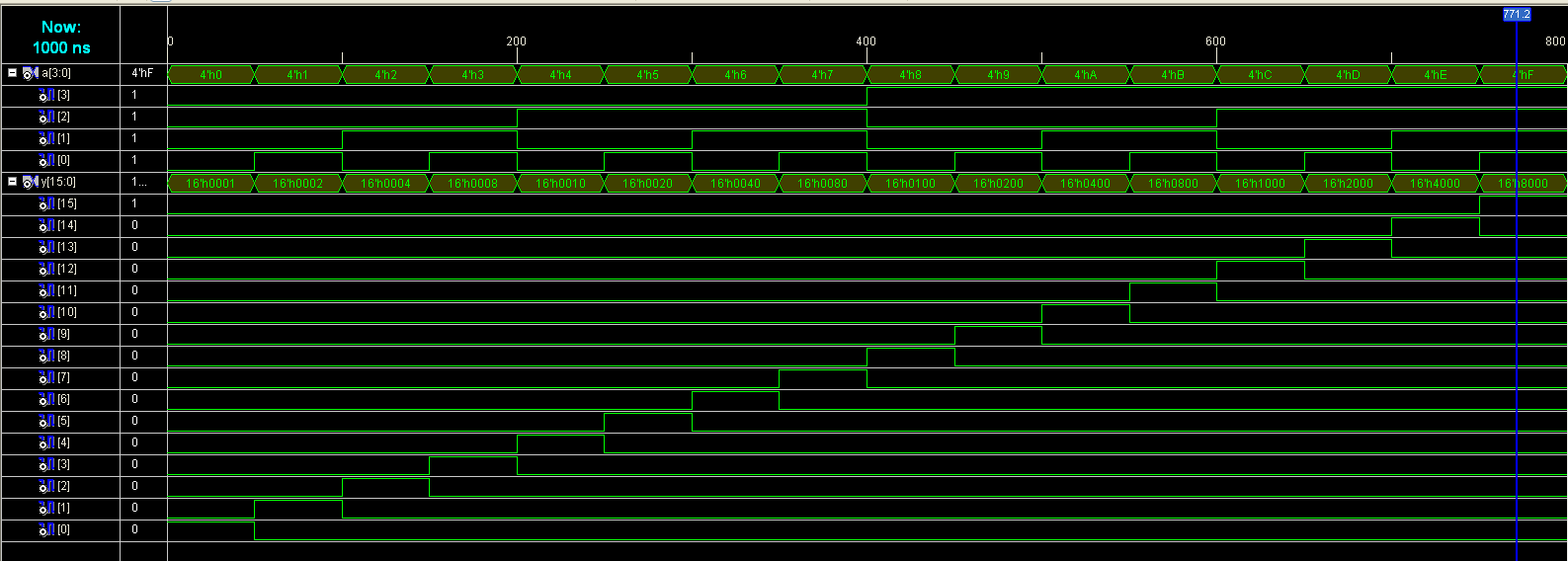
Y(13) <= A(3) and A(2) and not A(1) and A(0);

Y(14) <= A(3) and A(2) and A(1) and not A(0);

Y(15) <= A(3) and A(2) and A(1) and A(0);

end process;

end Behavioral;



На базе дешифратора разработать схему, реализующую функцию алгебры логики по табл. 2 из лабораторной работы №1. Проверить работу полученной схемы с помощью функционального и временного моделирования

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity list52 is

Port (

A : in STD\_LOGIC\_VECTOR(3 downto 0);

Y : out STD\_LOGIC\_VECTOR(15 downto 0);

F : out STD\_LOGIC

);

end list52;

architecture Behavioral of list52 is

signal temp\_Y : STD\_LOGIC\_VECTOR(15 downto 0) := (others => '0');

begin

process (A)

begin

temp\_Y(0) <= not A(3) and not A(2) and not A(1) and not A(0);

temp\_Y(1) <= not A(3) and not A(2) and not A(1) and A(0);

temp\_Y(2) <= not A(3) and not A(2) and A(1) and not A(0);

temp\_Y(3) <= not A(3) and not A(2) and A(1) and A(0);

temp\_Y(4) <= not A(3) and A(2) and not A(1) and not A(0);

temp\_Y(5) <= not A(3) and A(2) and not A(1) and A(0);

temp\_Y(6) <= not A(3) and A(2) and A(1) and not A(0);

temp\_Y(7) <= not A(3) and A(2) and A(1) and A(0);

temp\_Y(8) <= A(3) and not A(2) and not A(1) and not A(0);

temp\_Y(9) <= A(3) and not A(2) and not A(1) and A(0);

temp\_Y(10) <= A(3) and not A(2) and A(1) and not A(0);

temp\_Y(11) <= A(3) and not A(2) and A(1) and A(0);

temp\_Y(12) <= A(3) and A(2) and not A(1) and not A(0);

temp\_Y(13) <= A(3) and A(2) and not A(1) and A(0);

temp\_Y(14) <= A(3) and A(2) and A(1) and not A(0);

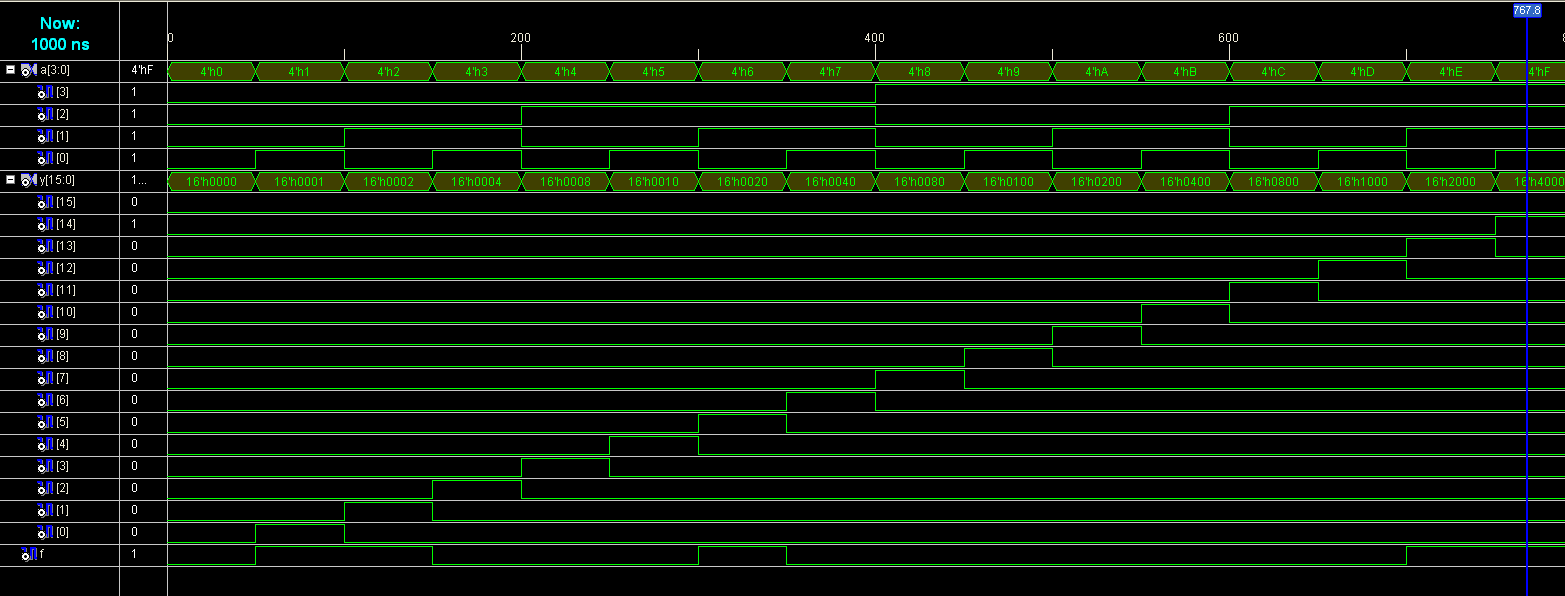
temp\_Y(15) <= A(3) and A(2) and A(1) and A(0);

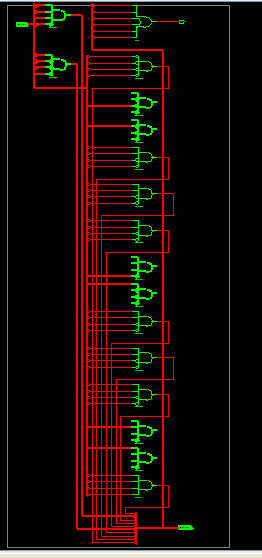
Y <= temp\_Y;

F <= temp\_Y(0) or temp\_Y(1) or temp\_Y(5) or temp\_Y(13) or temp\_Y(14) or temp\_Y(15);

end process;

end Behavioral;



****

**Вывод**

Изучили принципы разработки схем на языке VHDL.