Лабораторная работа 2 (Сложные логические функции)

☑ Done ☑

Цель

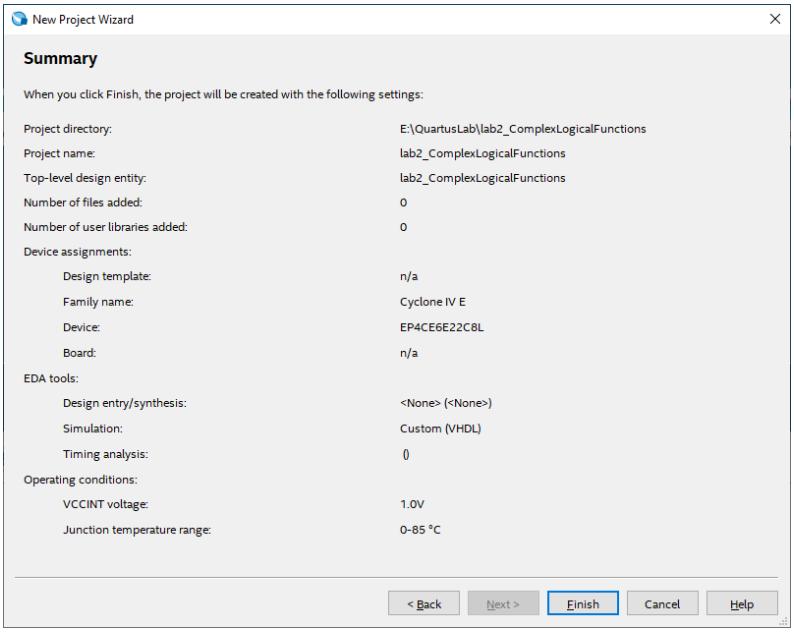
Ознакомиться с примерами сложных логических функций. Написать на языках VHDL и SystemVerylog программу для заданной вариантом логической функции.

Задание

Реализовать на языке VHDL и SystemVerylog сложную логическую функцию на выбор.

Выполнение

1. Создаем пустой проект с такими же параметрами как и лабораторная №1



2. Мой вариант

2. Создаем `Verilog HDL File`, `Block Diagram/Schematic File`, `VHDL File` и в разделе Verification/Debugging File выбираем `University Program VWF` и заполняем их кодом

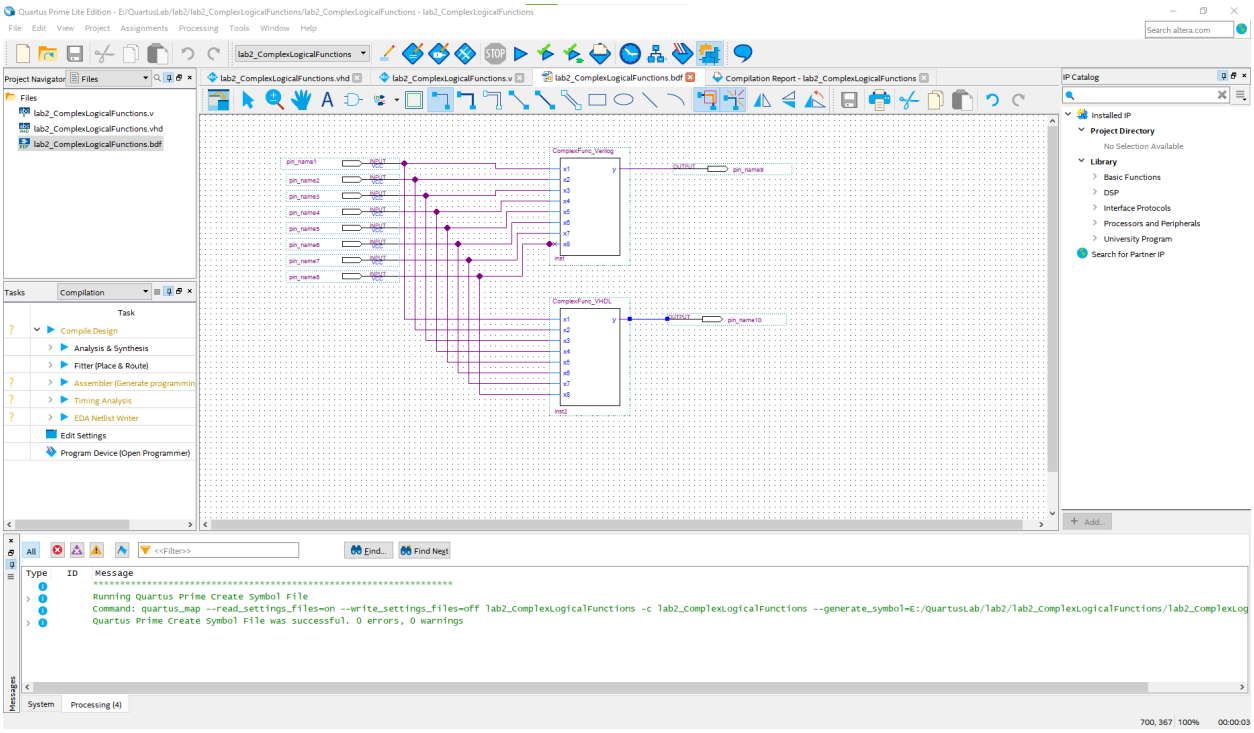
```
Library IEEE;
Use IEEE.STD_LOGIC_1164.all;
entity ComplexFunc_VHDL is
    Port (x1: in STD_LOGIC;
            x2: in STD_LOGIC;
            x3: in STD_LOGIC;
            x4: in STD_LOGIC;
            x5: in STD_LOGIC;
            x6: in STD_LOGIC;
            x7: in STD_LOGIC;
```

```
            x8: in STD_LOGIC;
            y: out STD_LOGIC);
end;
architecture Behavioral of ComplexFunc_VHDL is
begin
    y <= not((x1 and x2 and x3 and x4) or (x5 and x6 and x7 and
end;
```

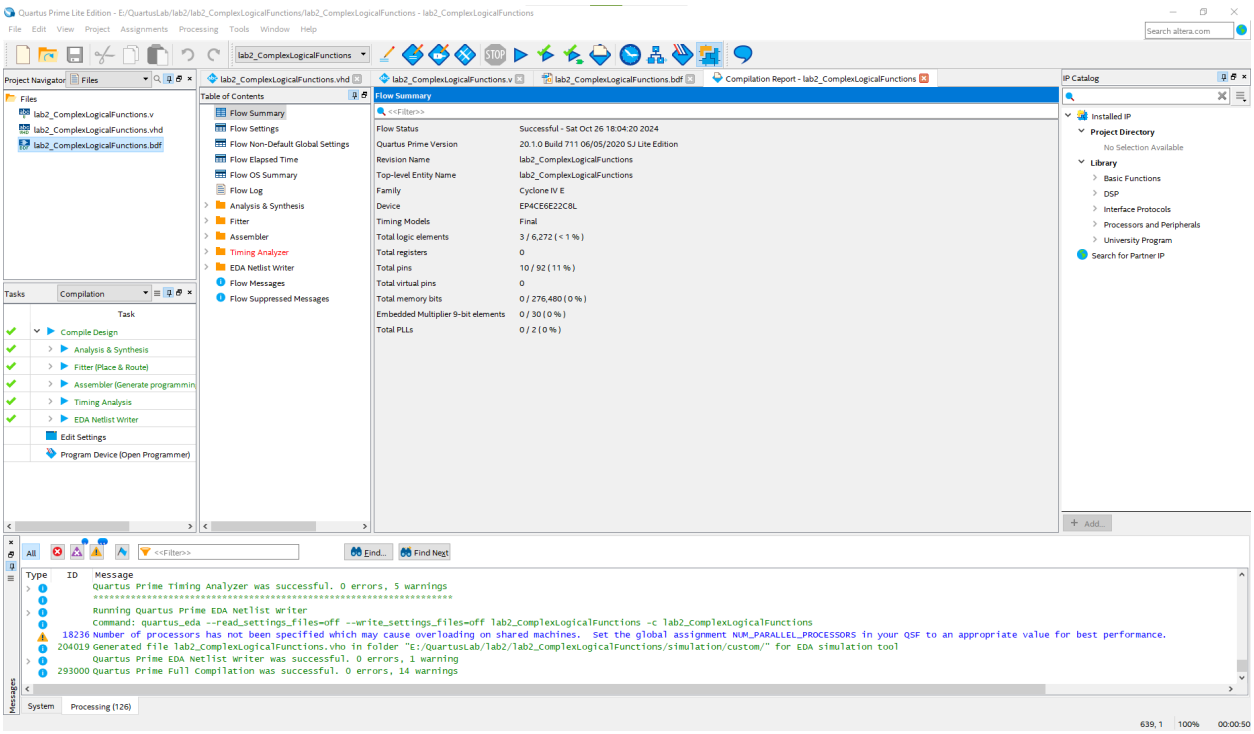
```
module ComplexFunc_Verilog (
    input x1, x2, x3, x4, x5, x6, x7, x8,
    output y);

    assign y = ~((x1 & x2 & x3 & x4) | (x5 & x6 & x7 & x8));
endmodule
```

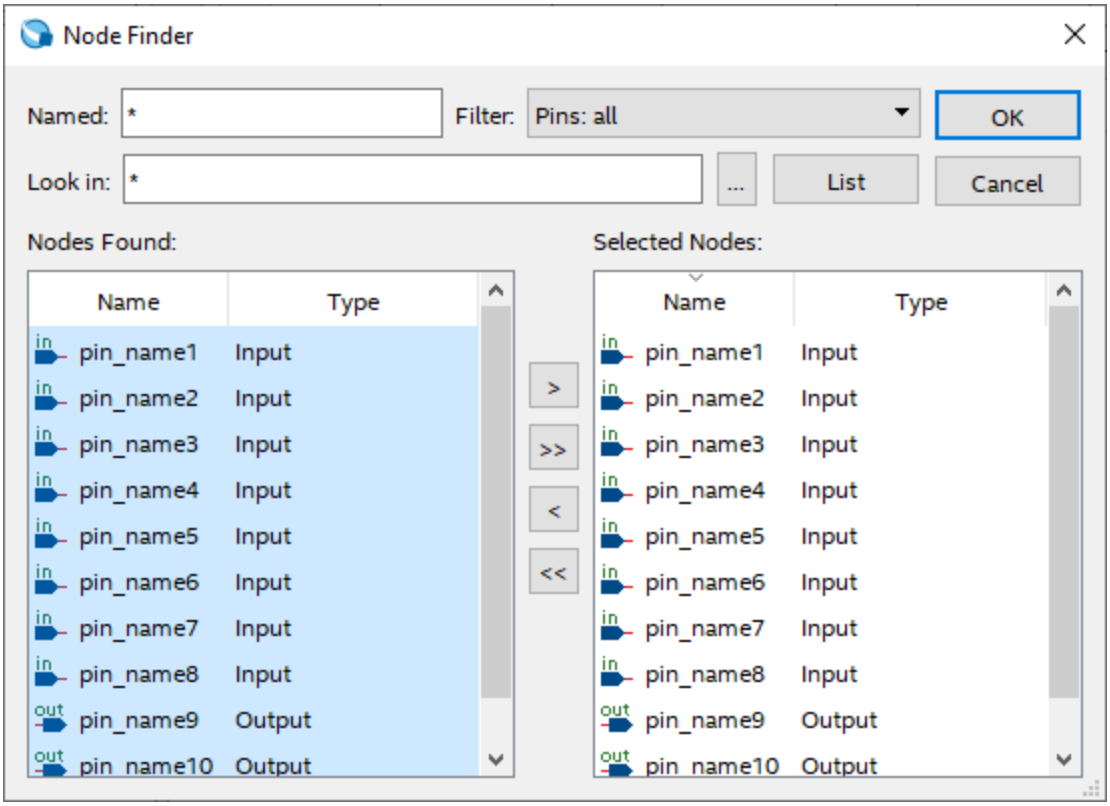
3. Компилируем их и добавляем на Block файл



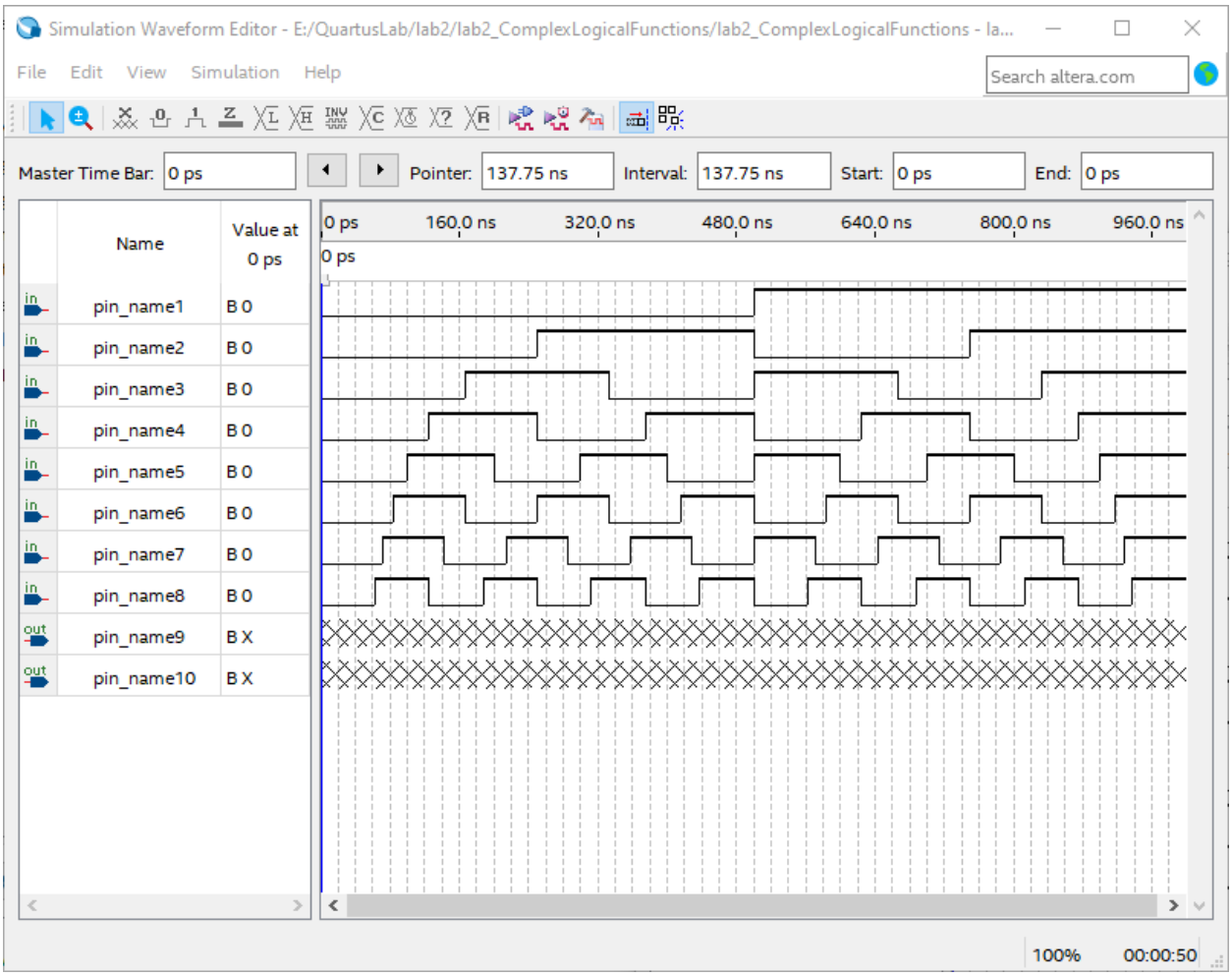
4. Отправляем нашу схему на верхний уровень и запускаем компиляцию проекта, дожидаемся успешного завершения.



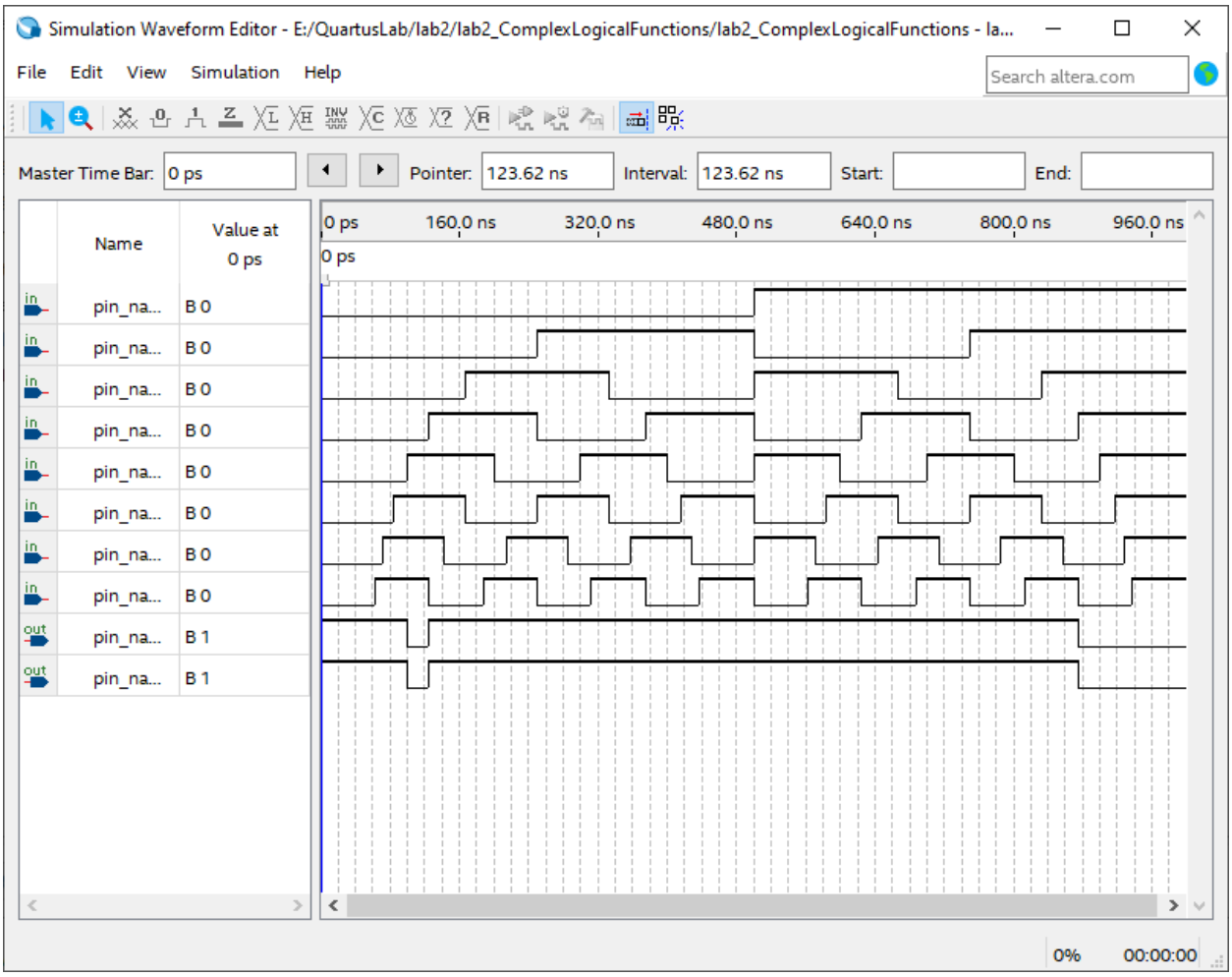
5. На странице добавления узлов в модуляцию ищем все наши узлы и добавляем их (забыл переименовать)



6. Для каждого узла выставляем разную частоту от 1 до 8 MHz



6. Убираем **novopt** из параметра оптимизации и запускаем симуляцию



Вывод

В ходе данной работы мы познакомились с построением сложных логических функций, а также запрограммировали заданную логическую функцию на языках VHDL и SystemVerylog и проверили работу нашего кода с помощью составления схемы и запуска симуляции работы.