

# Лабораторная работа 1 (конъюнктор)



## Цель

Ознакомиться с программой Quartus Prime Lite, языками VHDL, SystemVerilog и написать на данных языках программу для заданной вариантом схемы.

## Задание

Реализовать на языке VHDL и SystemVerilog элемент AND «конъюнктор».

## Выполнение

1. Запускаем Quartus Prime Lite версии 20.1 от имени администратора и создаем новый проект, выбираем для него название и директорию

New Project Wizard

**Directory, Name, Top-Level Entity**

What is the working directory for this project?

E:/QuartusLab/lab1\_conjunctor

What is the name of this project?

lab1\_conjunctor

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

lab1\_conjunctor

Use Existing Project Settings...

< Back Next > Finish Cancel Help

2. Пропускаем страницы с шаблонами и добавлением файлов
3. Выбираем **Cyclone IV E**, количество ножек **144**,  
Частотное ядро  
**8L** и выбрать следующий  
ПЛИС  
**EP4CE6E22C8L** по указанию методички.

**New Project Wizard**

**Family, Device & Board Settings**

Device: **Board**

Select the family and device you want to target for compilation.  
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family:  
Family: **Cyclone IV E**  
Device: **All**

Target device:  
☐ Auto device selected by the Fitter  
☒ Specific device selected in 'Available devices' list  
☐ Other: n/a

Show in 'Available devices' list:  
 Package: **Any**  
 Pin count: **144**  
 Core speed grade: **BL**  
 Name filter:   
☒ Show advanced devices

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit
EP4CE6E22C8L	1.0V	6272	92	92	276480	30
EP4CE6E22I8L	1.0V	6272	92	92	276480	30
EP4CE10E22C8L	1.0V	10320	92	92	423936	46
EP4CE10E22I8L	1.0V	10320	92	92	423936	46
EP4CE15E22C8L	1.0V	15408	82	82	516096	112
EP4CE15E22I8L	1.0V	15408	82	82	516096	112

< Back Next > Finish Cancel Help

#### 4. Настраиваем симуляцию VHDL, все остальное выключаем

**New Project Wizard**

**EDA Tool Settings**

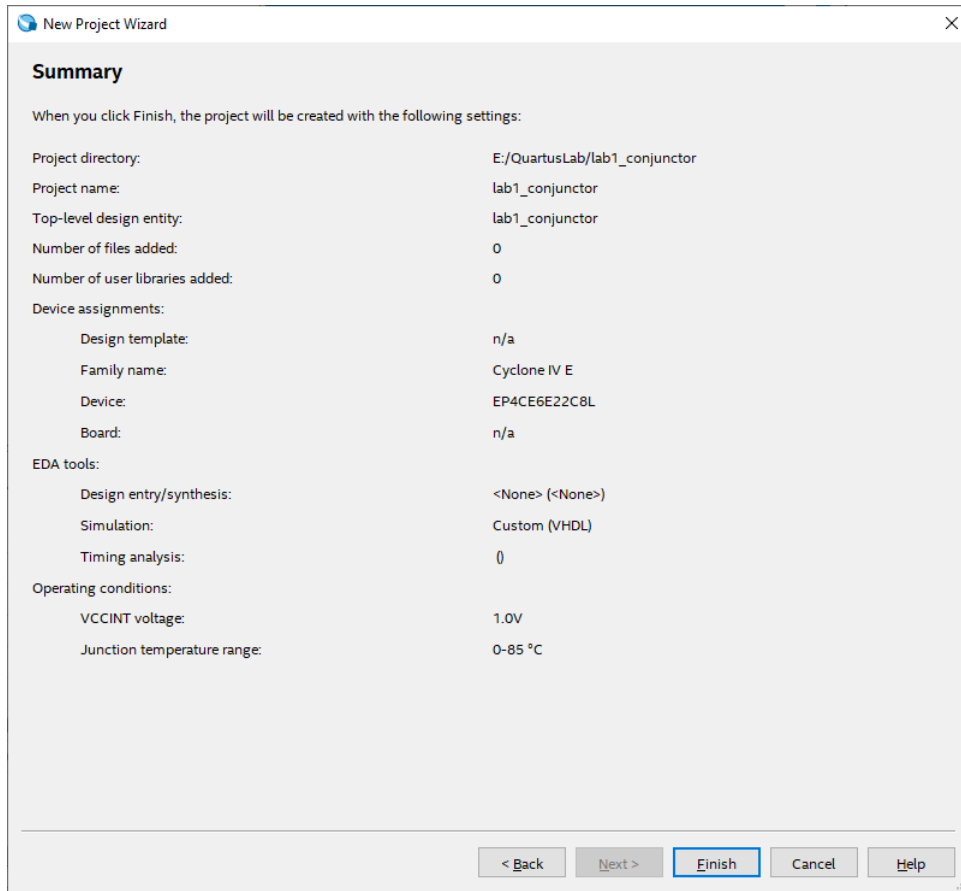
Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

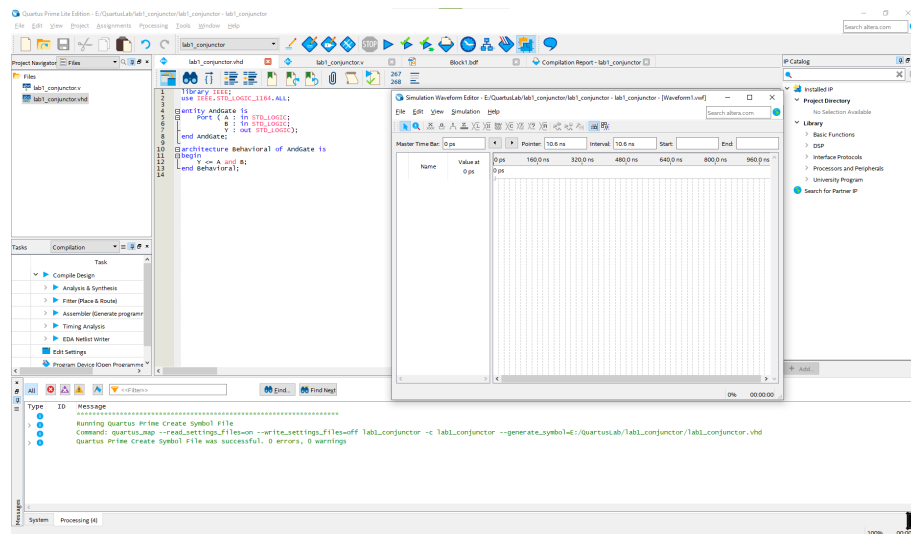
Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synth...	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	Custom	VHDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back Next > Finish Cancel Help

#### 5. Получаем отчет



6. Создаем **Verilog HDL File** , **Block Diagram/Schematic File** , **VHDL File** и в разделе **Verification/Debugging File** выбираем **University Program VWF**

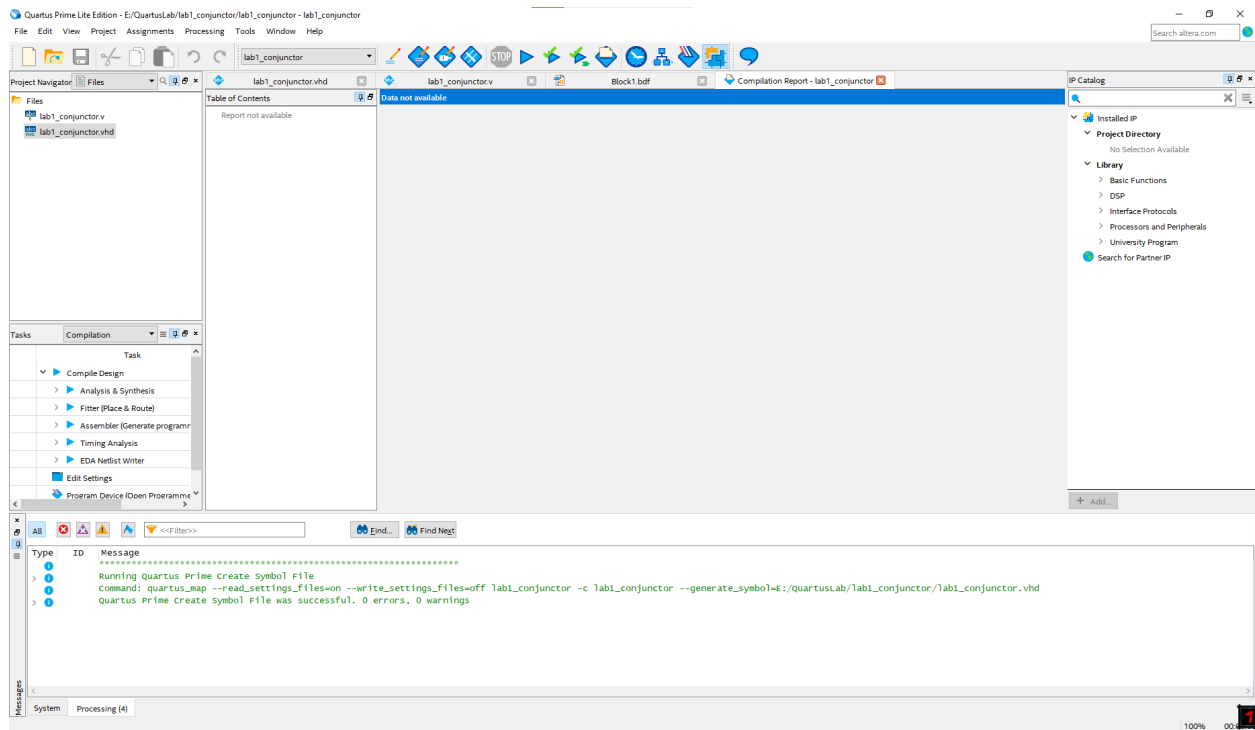


7. Заполняем их кодом

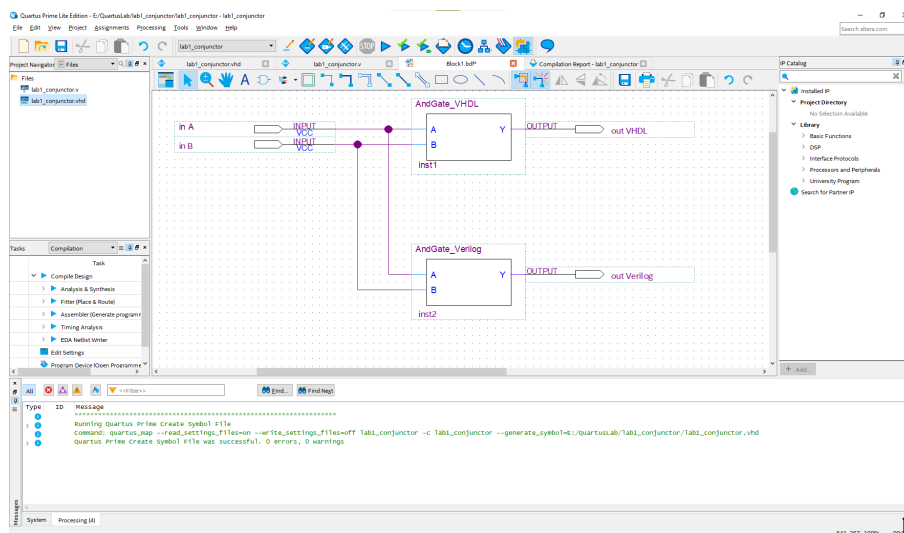
```
module AndGate (  
    input A,  
    input B,  
    output Y  
);  
    assign Y = A & B;  
endmodule
```

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
  
entity AndGate is  
    Port ( A : in STD_LOGIC;  
          B : in STD_LOGIC;  
          Y : out STD_LOGIC);  
end AndGate;  
  
architecture Behavioral of AndGate is  
begin  
    Y <= A and B;  
end Behavioral;
```

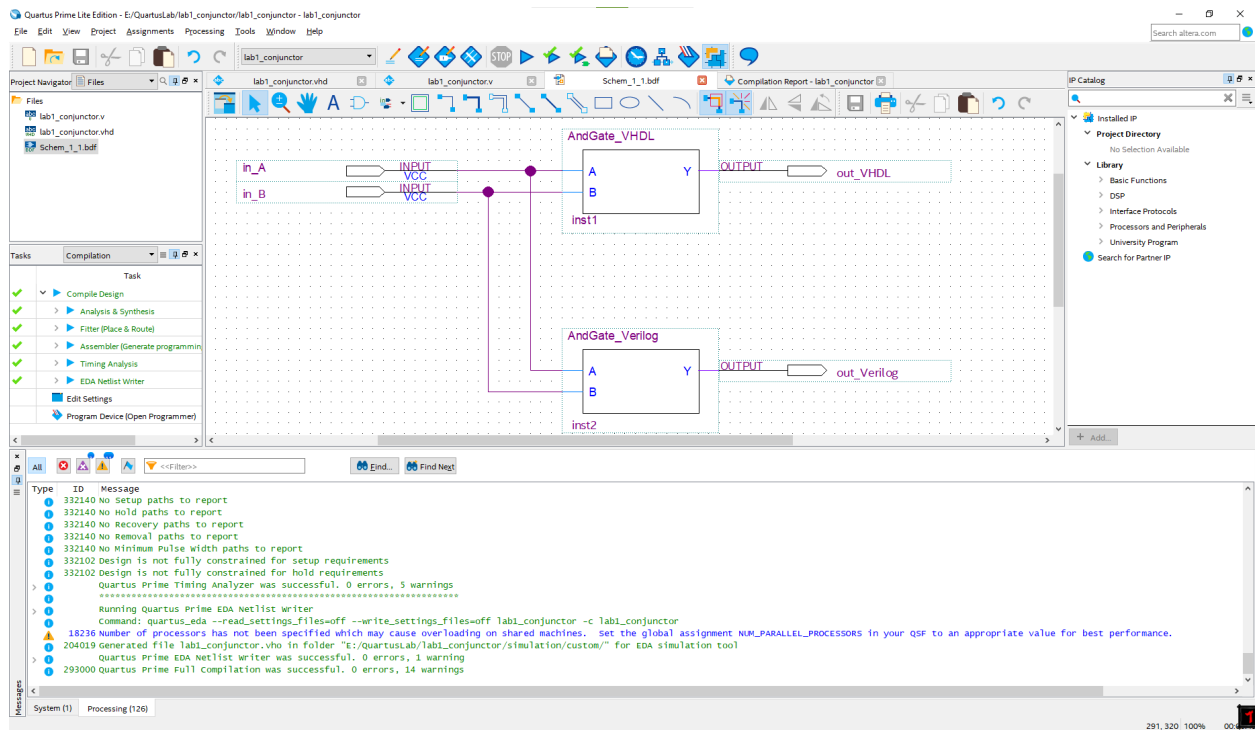
## 8. Компилируем



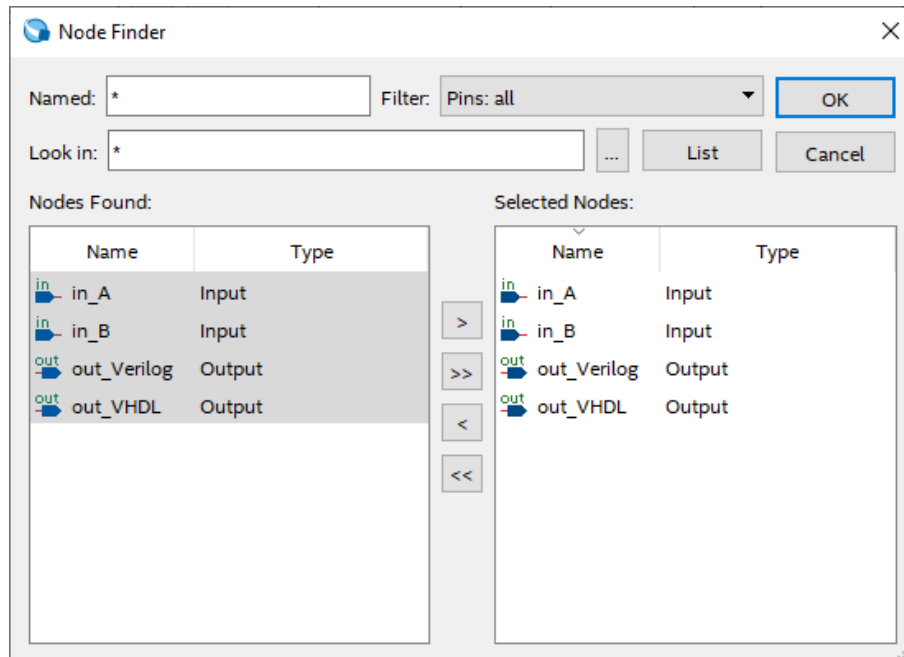
9. Добавляем наши схемы на Block окно и добавляем к ним входы и выходы и сохраняем под именем **Schem\_1\_1.bpf**

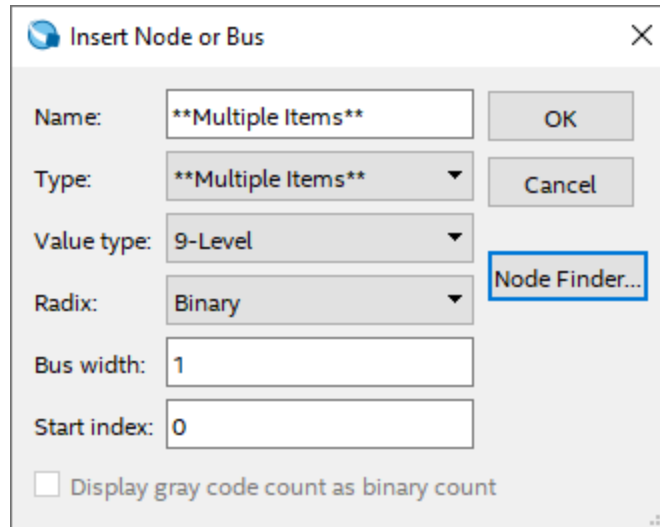


10. Отправляем нашу схему на верхний уровень и запускаем компиляцию проекта

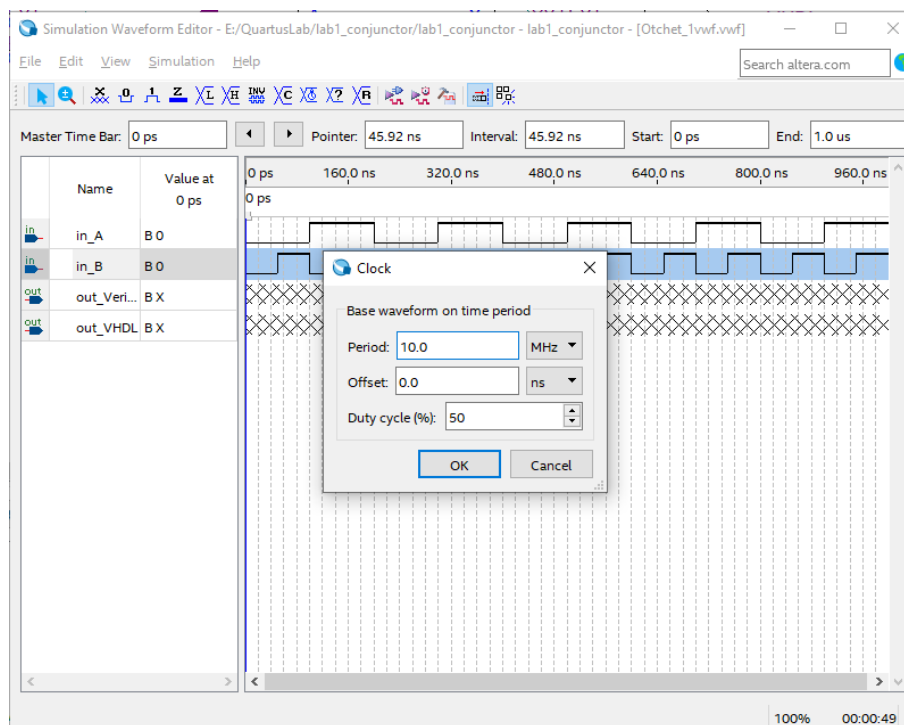


## 11. Выделяем все найденные нами сигналы





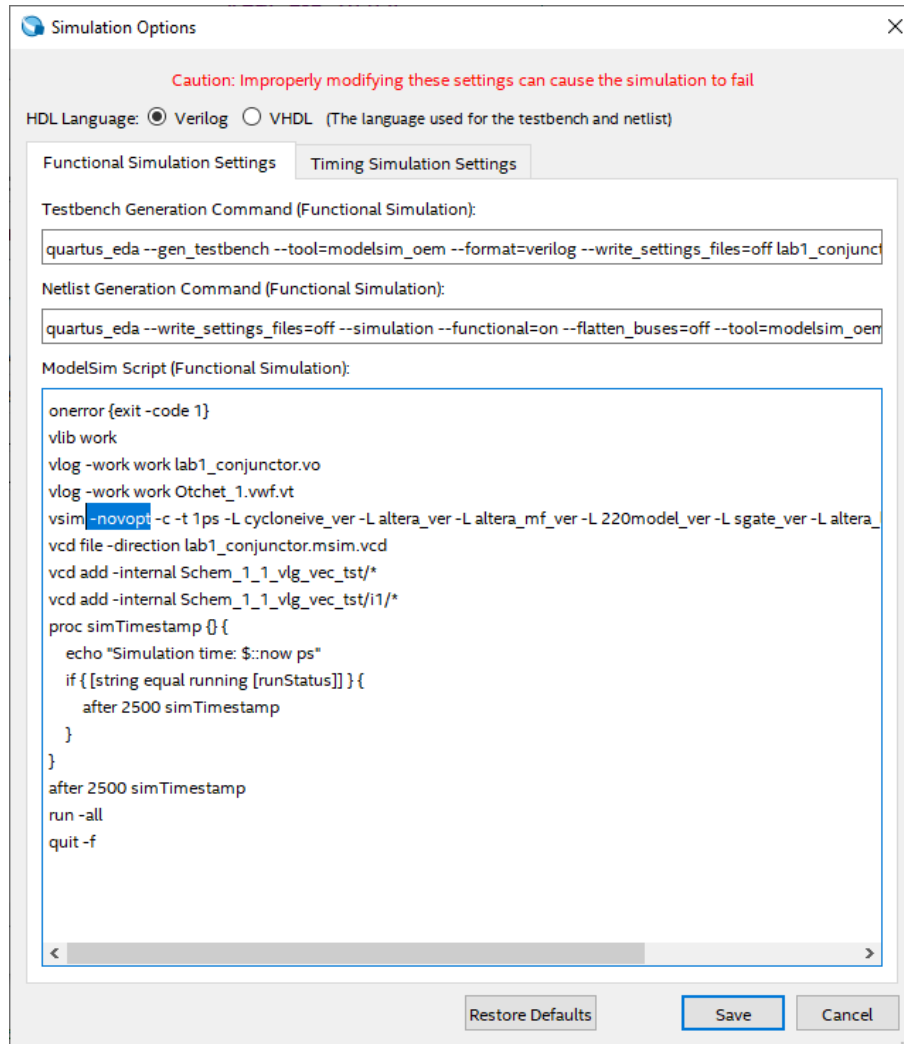
12. Задаем чистоты сигналам 10 MHz для A и 5 MHz для B



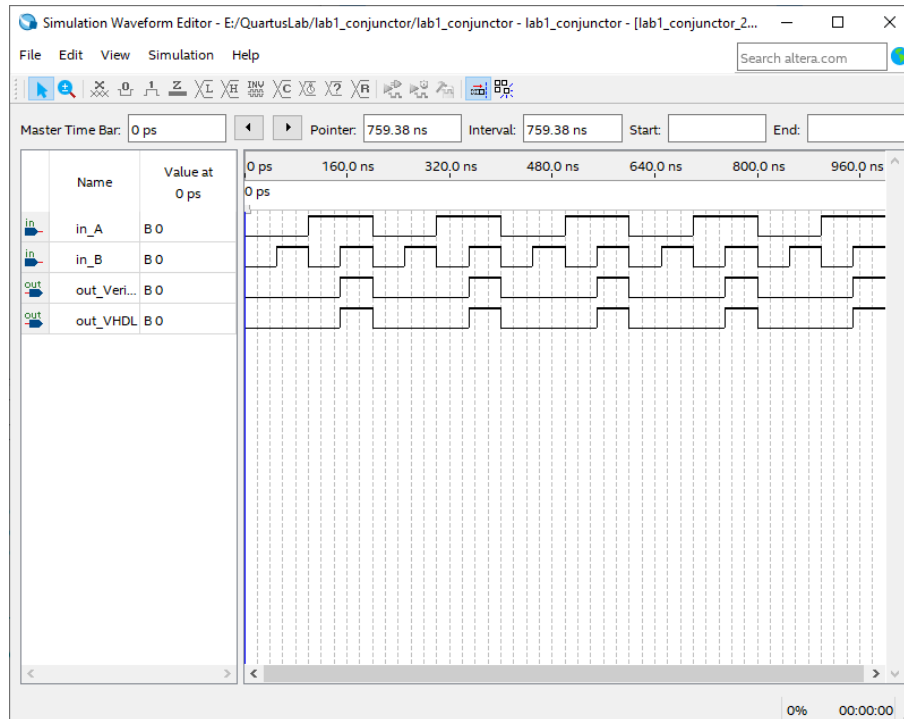
После сохраняем его как `Otchet_1.vwf`

13. Сбрасываем настройки по умолчанию и убираем показатель оптимизации -novopt





## 14. Запускаем модель



## Вывод

В ходе данной работы мы познакомились с программой Quartus Prime Lite и языками VHDL И SystemVerilog, а также запрограммировали логический элемент XOR на данных языках и проверили работу нашего кода с помощью составления схемы и запуска симуляции работы.