САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

Пакеты Quartus и ModelSim. Лабораторная работа 2.

Дисциплина

«Технологии проектирования аппаратных средств компьютерных систем»

Выполнил:

Бараев Д.Р.

Группа: 3540901/02001

Преподаватель: А. П. Антонов

Санкт-Петербург

2021

Оглавление

[1 Lab 2\_1 Запуск ModelSim из Quartus Prime 3](#_Toc64911336)

[1.1 Цели 3](#_Toc64911337)

[1.2 Схема проекта 3](#_Toc64911338)

[1.3 Добавление IP-компонентов в проект 4](#_Toc64911339)

[1.4 Настройка ModelSim из Quartus 5](#_Toc64911340)

[1.5 Выводы 7](#_Toc64911341)

[2 Lab 2\_2 Запуск ModelSim независимо от Quartus Prime 8](#_Toc64911342)

[2.1 Задание 8](#_Toc64911343)

[2.2 Обзор проекта 8](#_Toc64911344)

[2.3 Запуск ModelSim независимо от Quartus Prime 9](#_Toc64911345)

[2.4 Выводы 12](#_Toc64911346)

[3 Lab 2\_3 Задание для самостоятельного выполнения 12](#_Toc64911347)

[3.1 Цель работы 12](#_Toc64911348)

[3.2 Задание 12](#_Toc64911349)

[3.3 Создание проекта и модулей на основе IP 12](#_Toc64911350)

[3.4 Создание модулей на Verilog 13](#_Toc64911351)

[3.5 Компиляция проекта и RTL Viewer 15](#_Toc64911352)

[3.6 Создание проекта и добавление файлов из Quartus Prime в пакете ModelSim 16](#_Toc64911353)

[3.7 Выводы 17](#_Toc64911354)

# Lab 2\_1 Запуск ModelSim из Quartus Prime

## Цели

В ходе данной лабораторной работы предстоит ознакомиться со следующим:

1. Как запустить ModelSim для моделирования из Quartus Prime.
2. Как использовать IP-функции из IP-каталога Quartus Prime с ModelSim.
3. Некоторые полезные ярлыки для ModelSim.

## Схема проекта

В этой лабораторной работе используется простая схема цифровой логики.

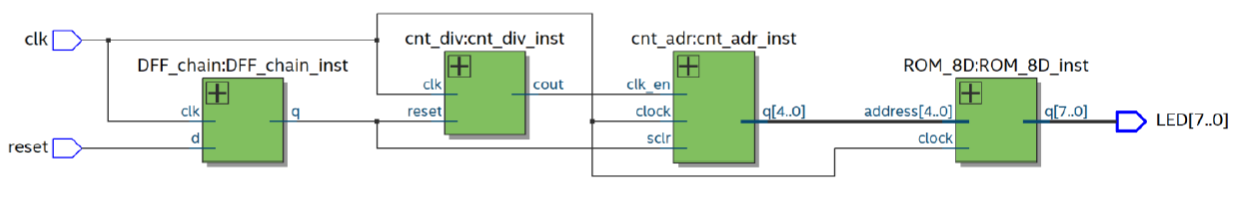


Рисунок 1. 1 – Схема проекта

Входы для модуля верхнего уровня:

* Clk - тактовый сигнал (частота 25 МГц).
* Сброс - сигнал сброса, поступающий от кнопки.

Выходы для модуля верхнего уровня:

* LED [7: 0] - выводы FPGA соединены со светодиодами на макетной плате.

Алгоритм реализации проекта:

* Входные часы делятся (значение деления параметризуется) на единицу cnt\_div .
* Модуль cnt\_adr предоставляет адреса для модуля ROM\_8D .
* Устройство ROM\_8D хранит данные и, в соответствии с текущим адресом, отображает слово (8 бит) данных насветодиод [7..0] выводит.
* Сигнал сброса синхронно сбрасывает все счетчики, имеющие активное значение «1».

Блок *DFF\_chain* синхронизирует внешний сигнал сброса. Он состоит из двух триггеров DFF, соединенных в цепочку.

Модуль *cnt\_div* делит входящий тактовый сигнал. Коэффициент деления — это параметр для единицы.

Модуль *cnt\_adr* предоставляет адреса для модуля ROM\_8D. Необходимо создать модуль *cnt\_adr*, используя LPM\_COUNTER из библиотеки IP.

Модуль *ROM\_8D* хранит данные, которые хранятся в файле *C:\Intel\_trn\Quartus\_ModelSim\Lab2\_1\ROM\_8D.hex*. Необходимо создать модуль *ROM\_8D*, используя ROM: *1-PORT* из библиотеки IP.

Блок *Lab2\_1* — это блок верхнего уровня. Он соединяет все блоки вместе. Исходный код для блока предоставляется.

После того, как убедились наличии всех нужных файлов в рабочей папке, переходим к созданию проекта в Quartus Prime и к добавлению IP-компонентов в проект.

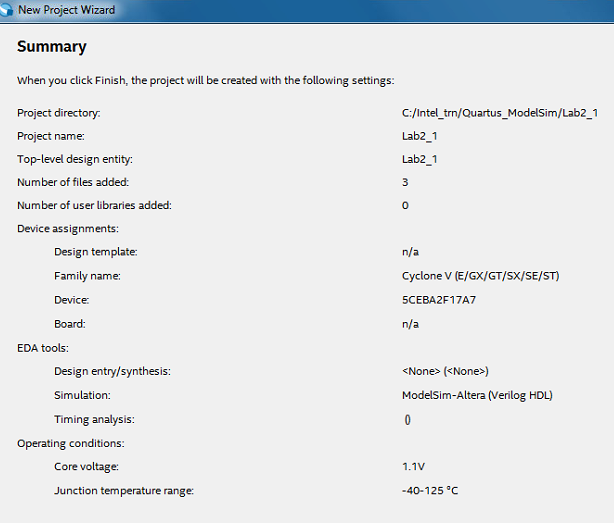


Рисунок 1. 2 - Настройки проекта

## Добавление IP-компонентов в проект

1. Создание модуля cnt\_adr с помощью LPM\_COUNTER.
2. Создание блока ROM\_8D с помощью ROM: 1-PORT.

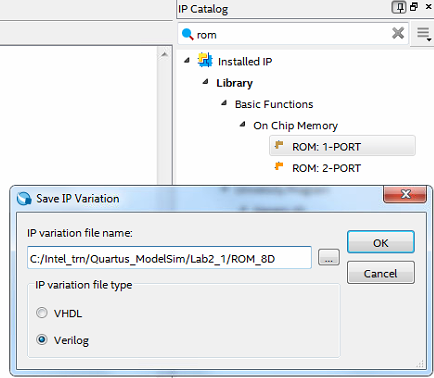


Рисунок 1. 3 - Создание блока ROM\_8D

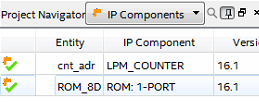


Рисунок 1. 4 - Компоненты IP успешно созданы

При помощи Analysis and Elaboration проверили правильность проекта.

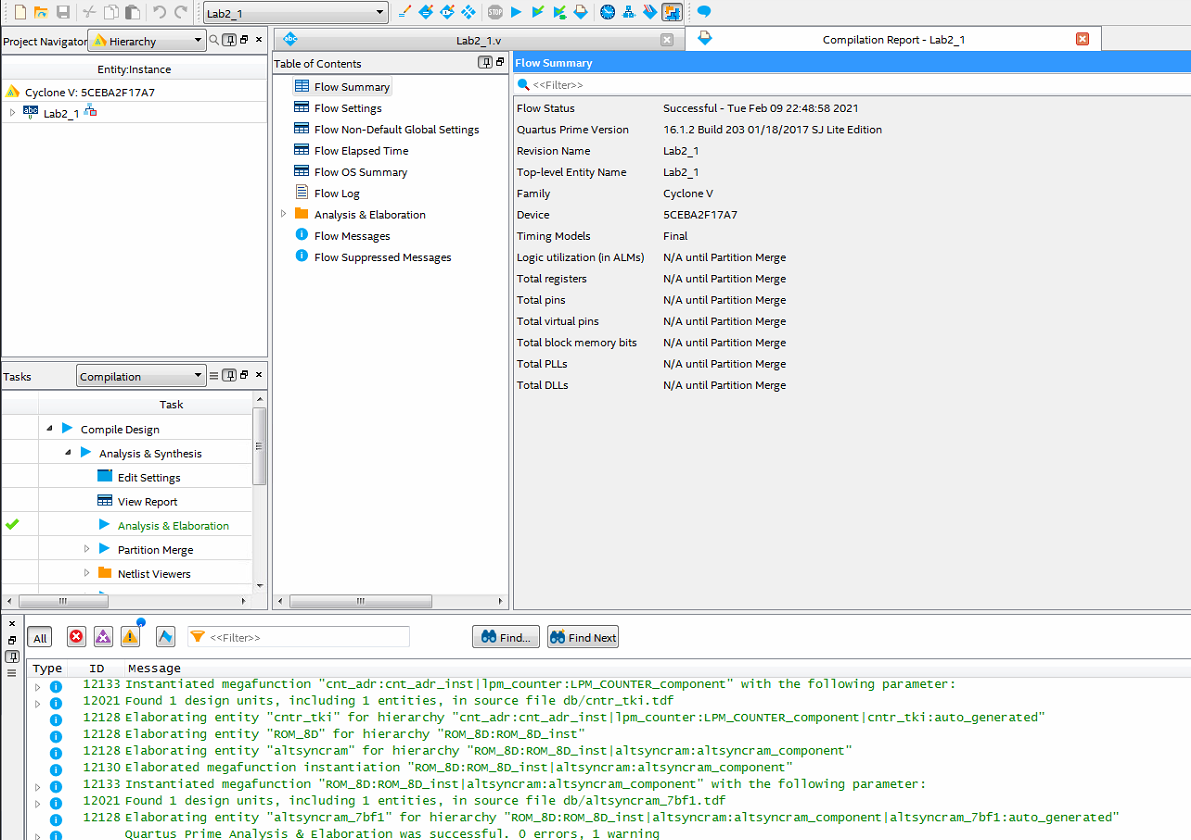


Рисунок 1. 5 - Результаты Analysis and Elaboration

## Настройка ModelSim из Quartus

Необходимо проверить настройки, чтобы Modelsim мог открываться через Quartus Prime Lite.

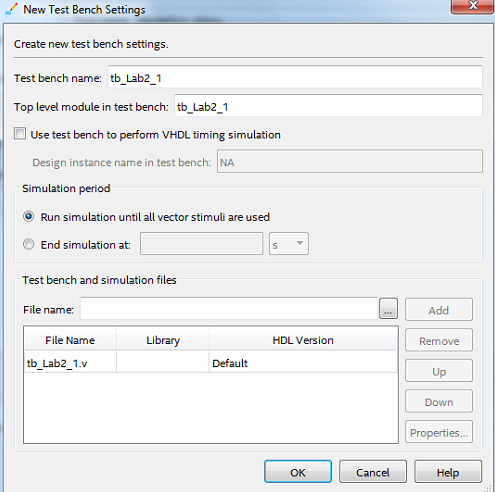


Рисунок 1. 6 - Настройка тестового стенда

После того, как все настройки выполнены, вы можете перейти на панель инструментов для полной компиляции проекта.

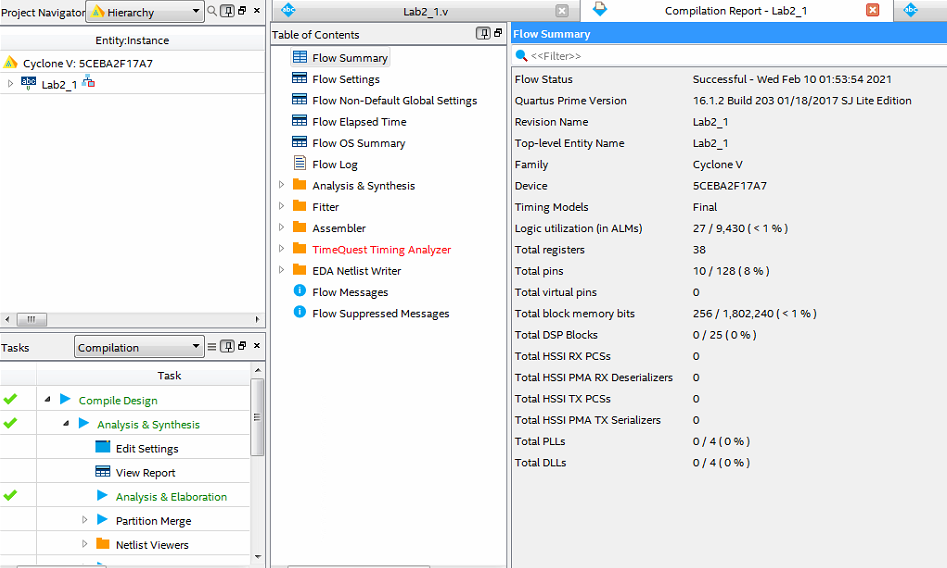


Рисунок 1. 7 -Полная компиляция проекта

Затем из Quartus: *Tools=> Run Simulation=> RTL Simulation*. ModelSim открывается на экране.

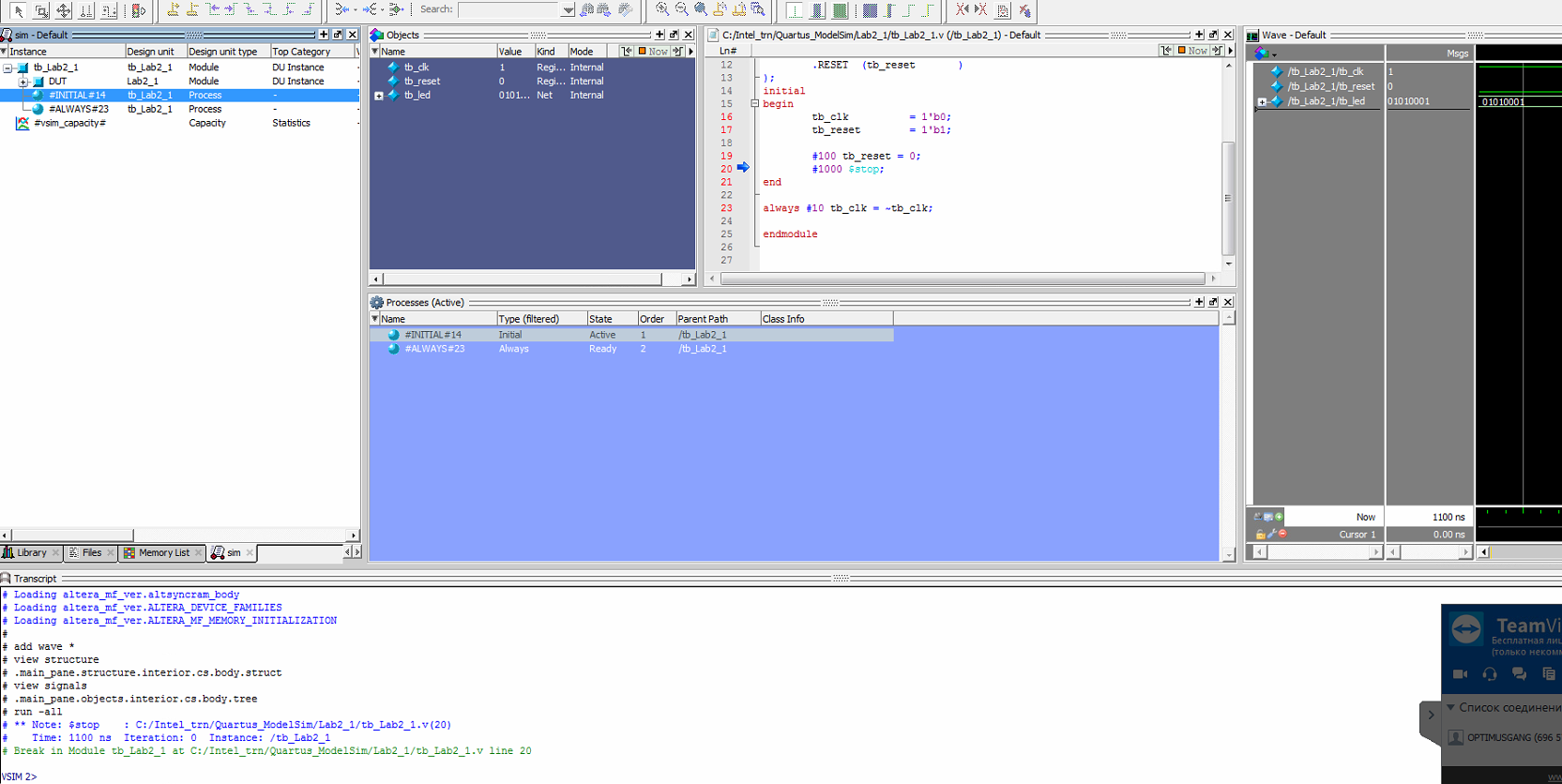


Рисунок 1. 8 - Графический интерфейс ModelSim

Выберите вкладку *Wave*, чтобы увидеть окно волны. Теперь вы можете наблюдать за формой сигнала и проверять работоспособность проекта.

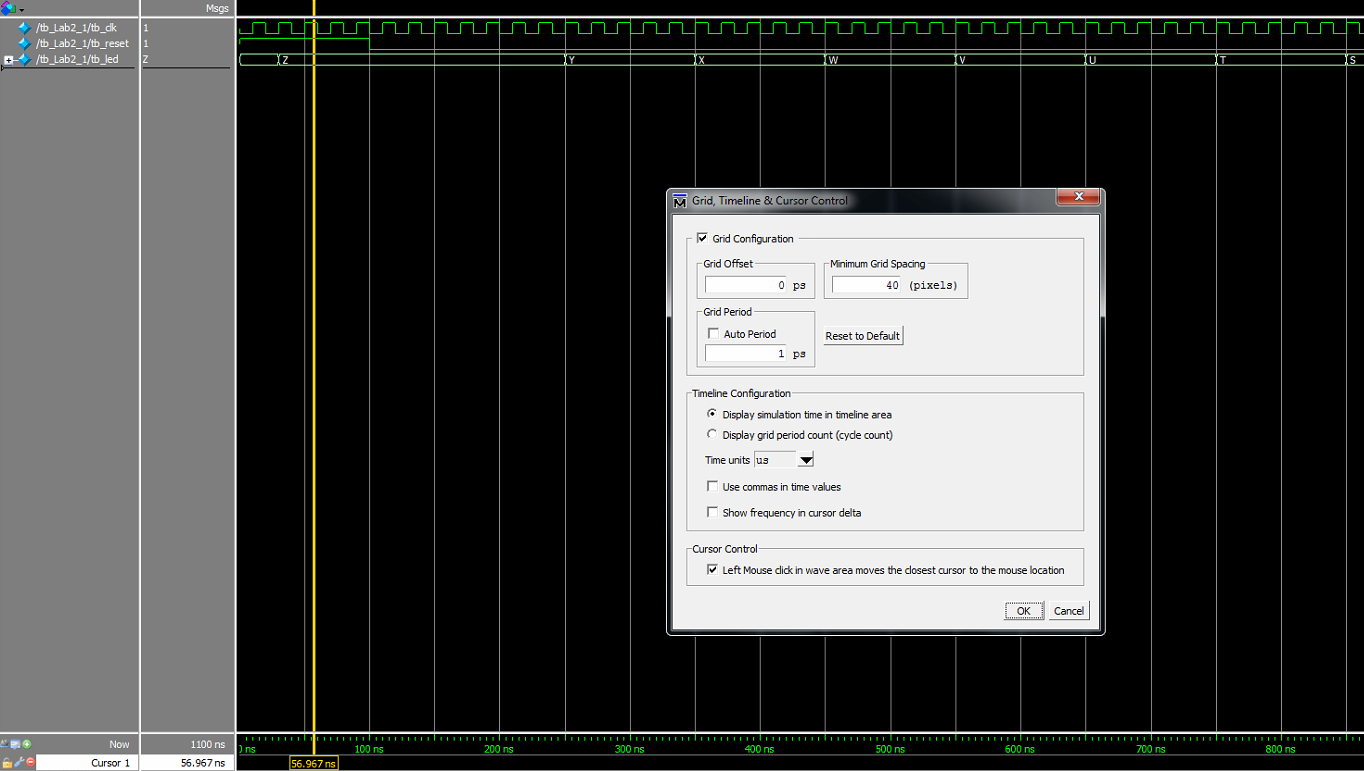


Рисунок 1. 9 - Окно Wave

## Выводы

В ходе данной лабораторной работы узнали, как использовать IP-модули в моделировании и запускать моделирование ModelSim из Quartus Prime с использованием родной ссылки.

Был создан проект в Quartus Prime, состоящий из файлов, описанных на языке Verilog, а также файлов из библиотеки IP компонентов пакета. Симуляция осуществлялась на основе файла-заготовки с тестом проекта.

# Lab 2\_2 Запуск ModelSim независимо от Quartus Prime

## Задание

В ходе выполнения данной лабораторной работы произошло ознакомление с такими вопросами, как:

* Как запустить ModelSim независимо от Quartus Prime
* Как использовать IP-функции из IP-каталога Quartus Prime с ModelSim

## Обзор проекта

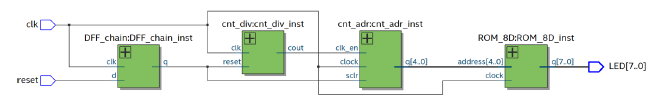


Рисунок 2. 1 - Схема проекта

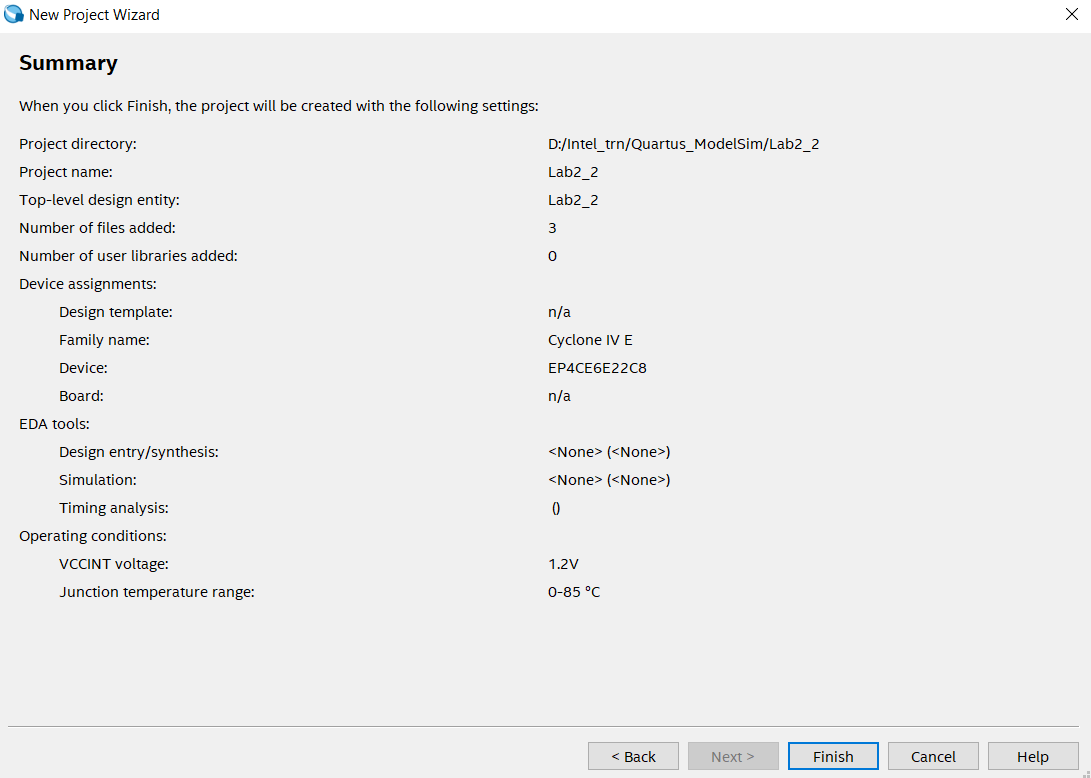


Рисунок 2. 2 - Создание проекта

1. Создание модуля cnt\_adr с помощью LPM\_COUNTER.
2. Создание блока ROM\_8D с помощью ROM: 1-PORT.

Проверьте, добавлены ли IP-адреса в проект: Навигатора проекта => IP Компоненты

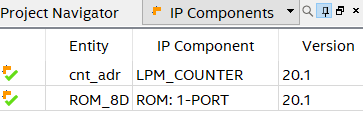


Рисунок 2. 3 - Результат добавления блоков cnt\_adr и ROM\_8D

Проверка правильности проекта через Analysis & Elaboration.

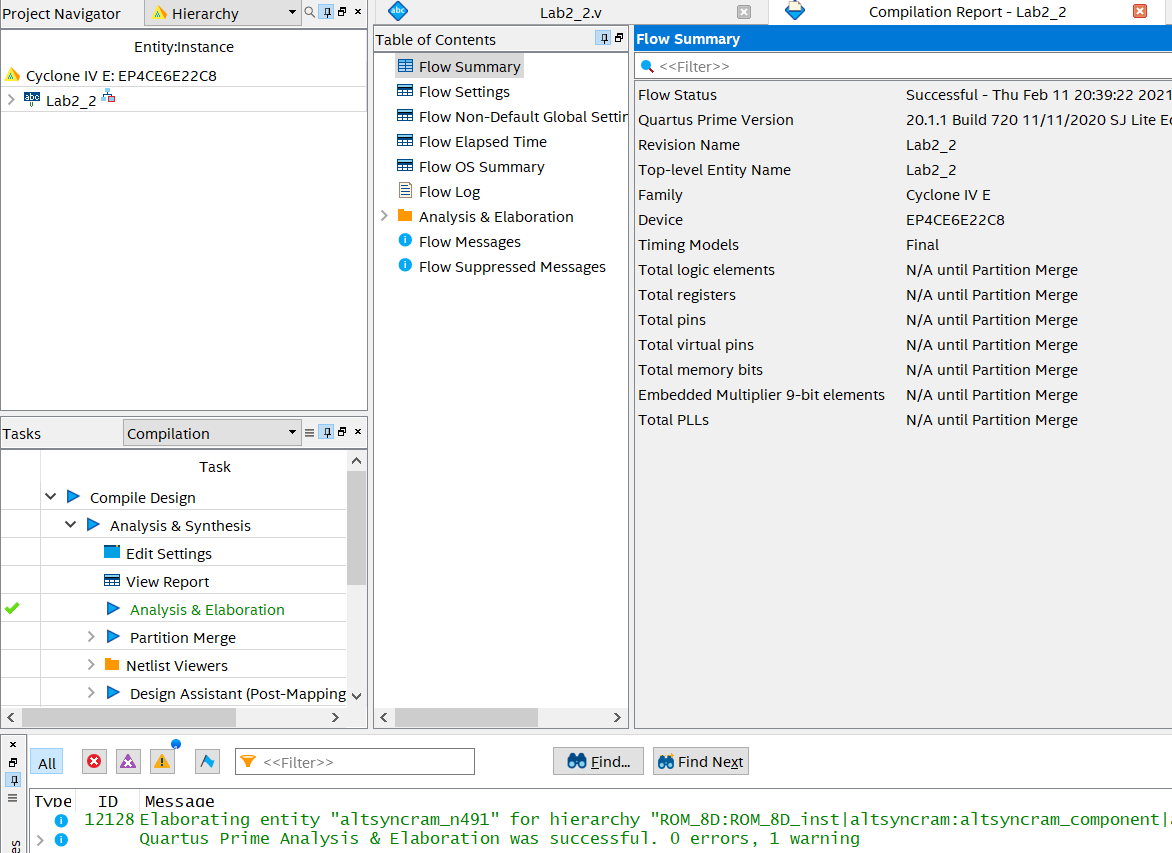


Рисунок 2. 4 - Проверка правильности проекта

## Запуск ModelSim независимо от Quartus Prime

1. Запустите ModelSim Intel FPGA Starter Edition
2. Измените рабочую папку на D:\Intel\_trn\Quartus\_ModelSim\Lab2\_2
3. Создайте рабочую библиотеку

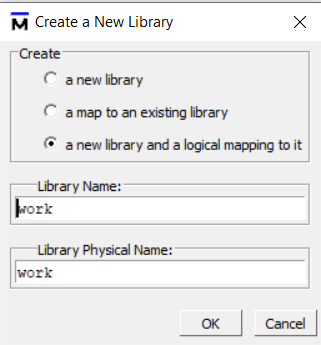


Рисунок 2. 5 - Создание рабочей библиотеки

1. Скомпилируйте исходные файлы

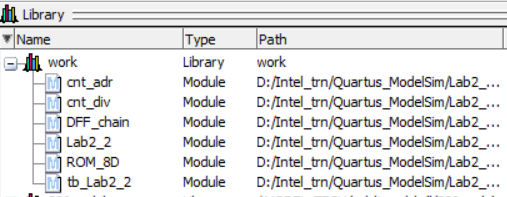


Рисунок 2. 6 - Скомпилированные файлы

1. В окне библиотеки щелкните значок «+» рядом с рабочей библиотекой, чтобы отобразить содержащиеся в ней файлы. После дважды щелкните tb\_Lab2\_2, чтобы загрузить проект.

Выходят две ошибки, о которых сообщается в окне стенограммы.

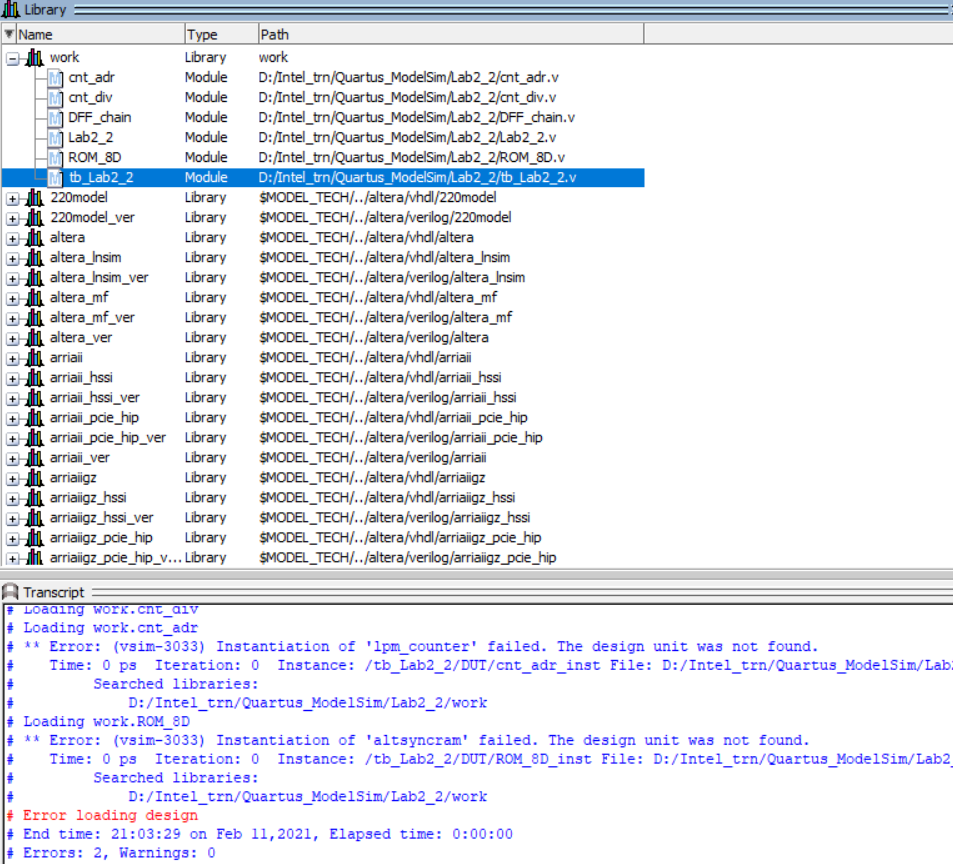


Рисунок 2. 7 - После загрузки проекта выходят ошибки

Для исправления ошибки необходимо обратиться к библиотекам lpm и altera\_mf.

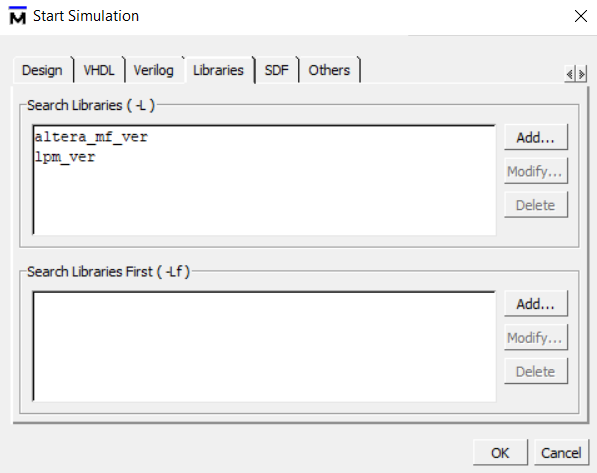


Рисунок 2. 8 - Добавление библиотек

После добавления библиотек симуляция прошла без ошибок и проект запустился.

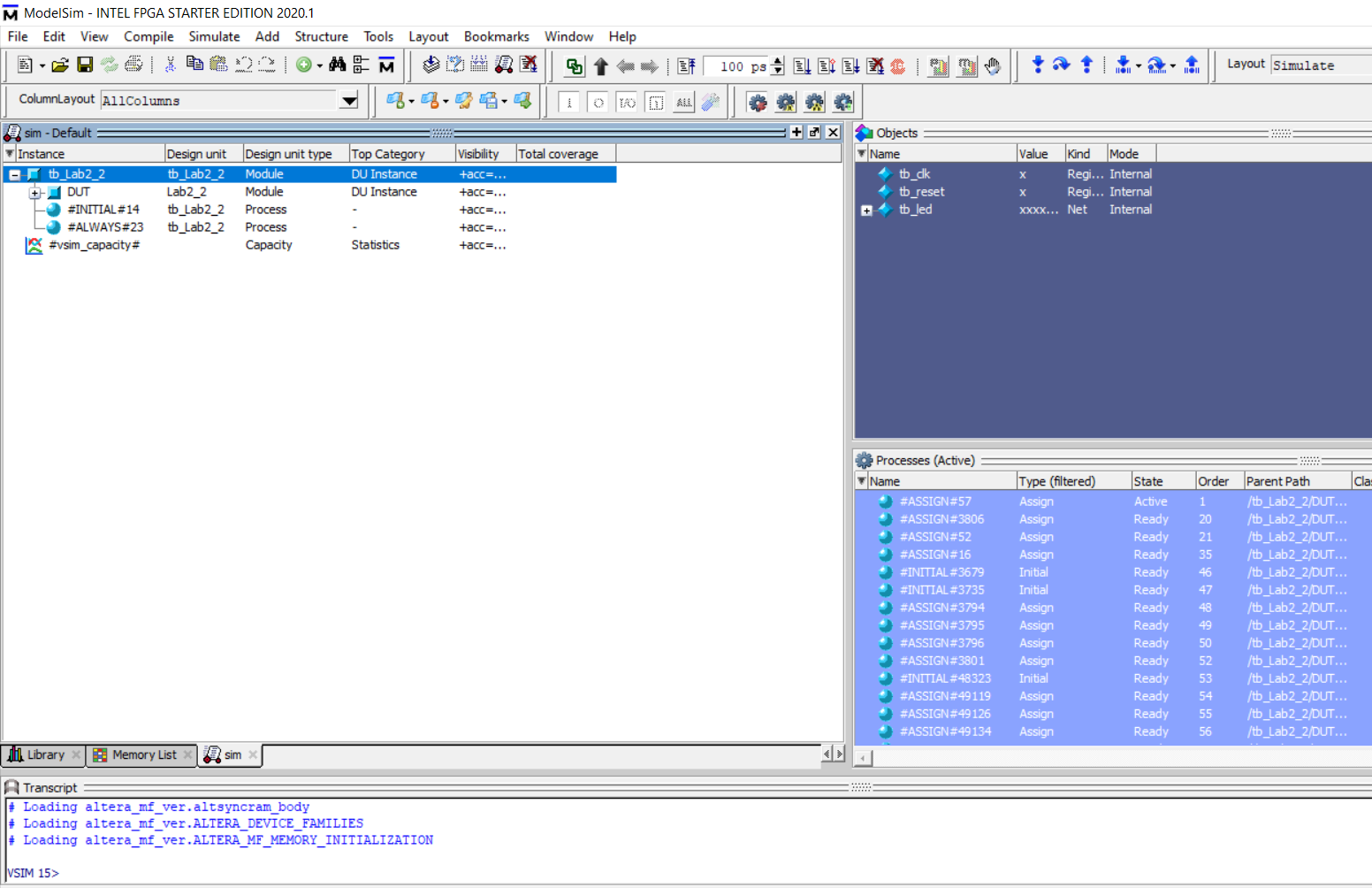


Рисунок 2. 9 - Успешный запуск проекта

Введите в окне стенограммы команду: ***do wave\_my\_lab2\_2.do***

Введите в окошко стенограммы команду: run 1000 ns

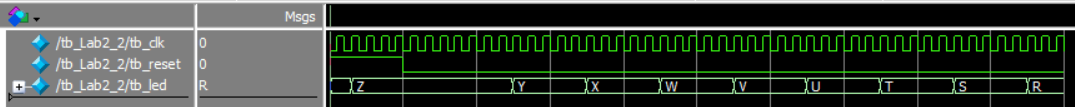


Рисунок 2. 10 - Волновое окно с сигналами верхнего уровня

## Выводы

В ходе лабораторной работы ознакомились с использованием IP-модулей в моделировании, и с моделированием ModelSim независимо от Quartus Prime. Проект был запущен самостоятельно, а необходимые библиотеки были добавлены вручную.

# Lab 2\_3 Задание для самостоятельного выполнения

## Цель работы

Целями лабораторной работы являются:

1. Описание части модулей для проекта на языке Verilog, а другой части – с помощью инструментария в IP библиотеке Quartus Prime.
2. Создание теста и его запуск в ModelSim.

## Задание

* Создать проект в пакете Quartus
* Создать модули на Verilog (имена указаны на структуре)
* Создать модули на основе IP (имена указаны на структуре)
* Создать модуль верхнего уровня иерархии на Verilog.
* Осуществить компиляцию проекта в пакете Quartus, исправить ошибки и проверить с помощью RTL Viewer, что проект собран правильно.
* Разработать тест - простейший тест (за основу можно взять тест из LAB2\_1).
* Создать проект в пакете ModelSim
* Включить исходные файлы в проект
* Осуществить компиляцию исходных файлов, включая тест.
* Создать конфигурацию для моделирования
* Осуществить моделирование для всего цикла счетчика CNT
* Проверить правильность работы устройства, при необходимости отладить его.
* Сохранить do файл с настройками окна временных диаграмм.

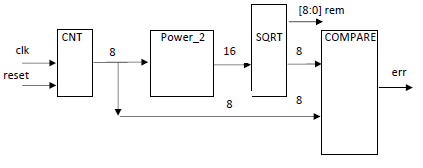


Рисунок 3. 1 - Структура проекта

## Создание проекта и модулей на основе IP

1. Создайте проект
2. Создать модули на Verilog, которые указаны на структуре.

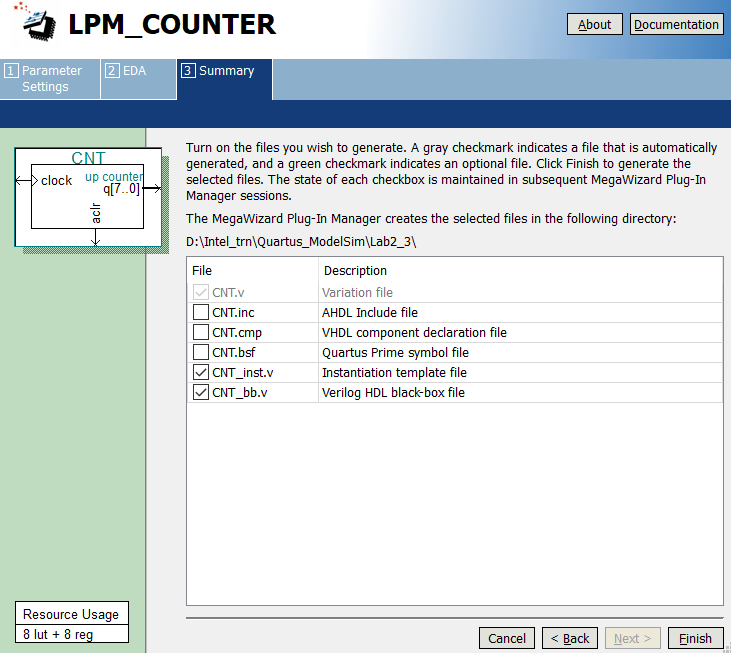


Рисунок 3. 2 - Создание счетчика CNT

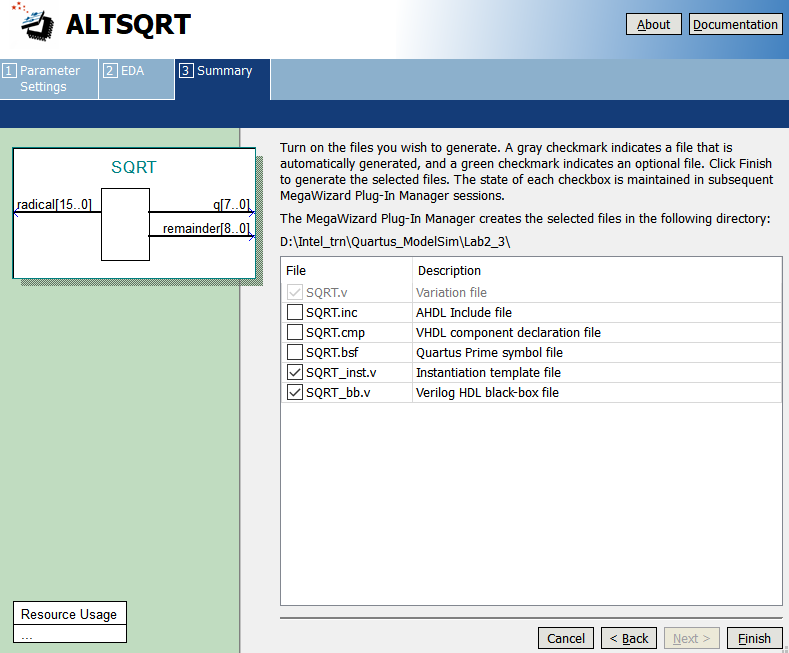


Рисунок 3. 3 - Создание модуля SQRT

## Создание модулей на Verilog

Описываем модули, которые необходимо писать на языке Verilog.

1. Модуль возведения в степень Power\_2.

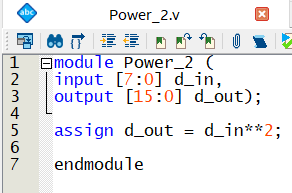


Рисунок 3. 4 - Код описания модуля Power\_2

1. Модуль сравнения COMPARE.

Он выводит 1, если входные данные из счетчика и модуля извлечения корня равны, и выводит 0, если они не равны.

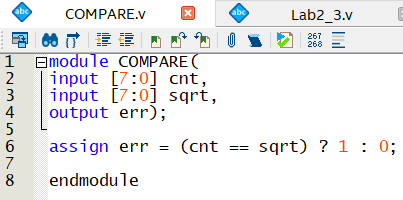


Рисунок 3. 5 - Код описания модуля COMPARE

1. Модуль верхнего уровня Lab2\_3.

Для правильной работы проекта необходимо описать его верхний уровень, задающий работу каждому отдельному модулю.

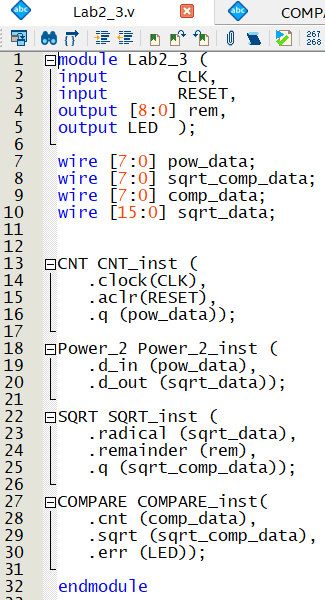


Рисунок 3. 6 - Код описания модуля верхнего уровня

## Компиляция проекта и RTL Viewer

Компиляция проекта была осуществлена без ошибок.

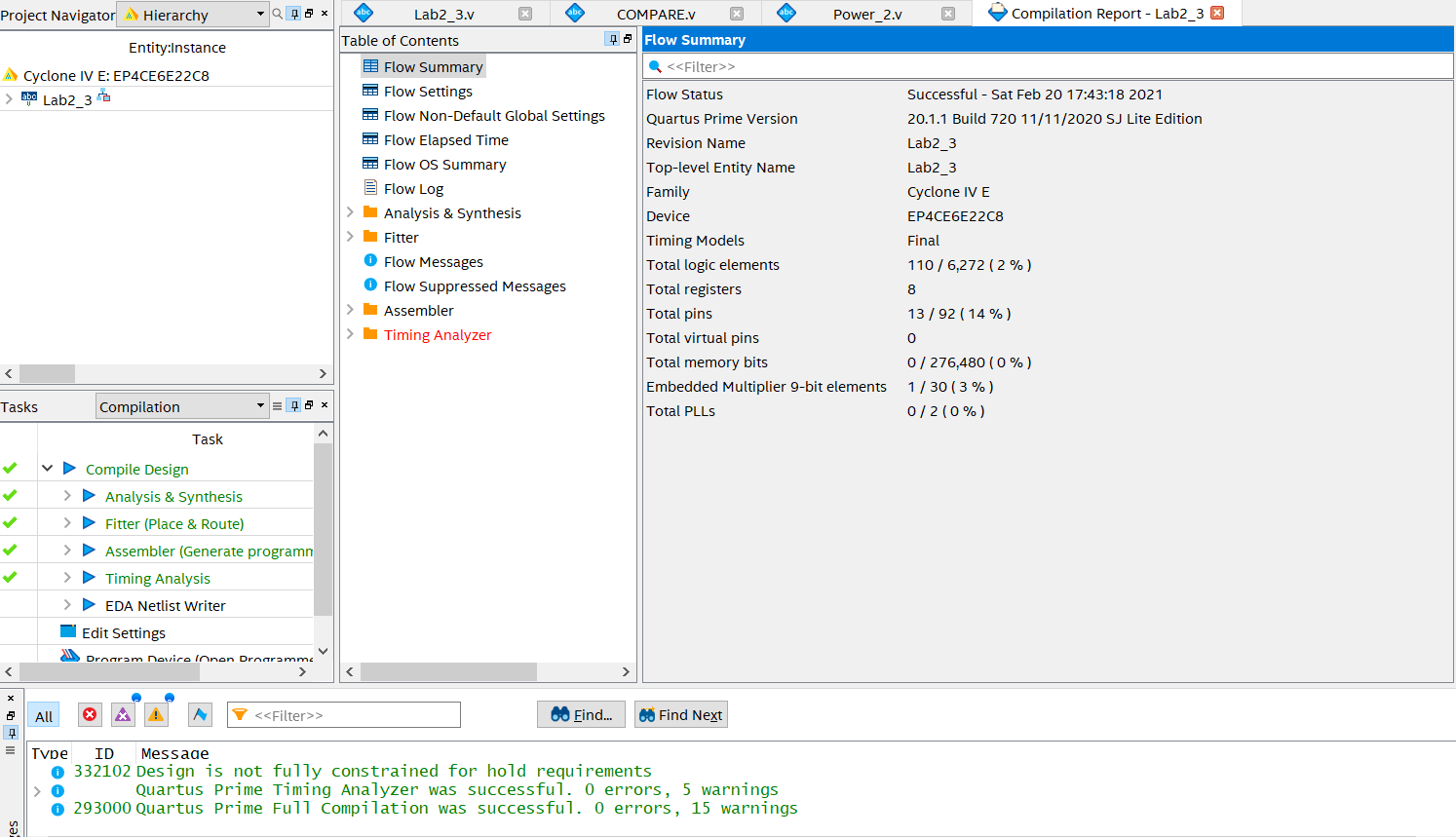


Рисунок 3. 7 - Компиляция проекта

Схема в RTL Viewer показывает, что проект был описан правильно и без ошибок.

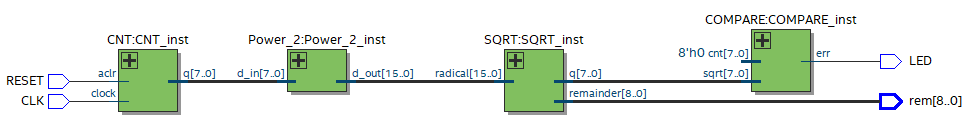


Рисунок 3. 8 - Схема проекта в RTL Viewer

Для симуляции в ModelSim был составлен простейший тест на основе теста из Lab2\_1.

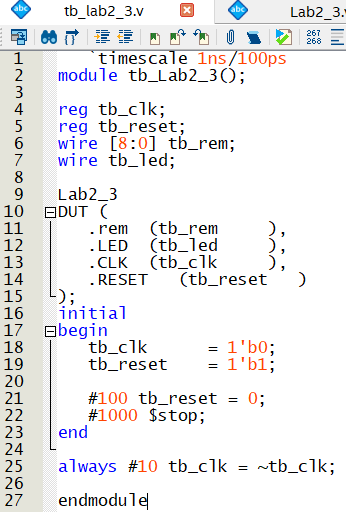


Рисунок 3. 9 - Тест tb\_Lab2\_3

## Создание проекта и добавление файлов из Quartus Prime в пакете ModelSim

Был запущен ModelSim, где был создан новый проект для тестирования. В него были добавлены и успешно скомпилированы файлы описаний из Quartus Prime.

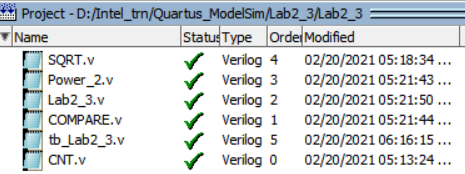


Рисунок 3. 10 - Добавленные и скомпилированные файлы в ModelSim

Для запуска симуляции были добавлены две библиотеки.

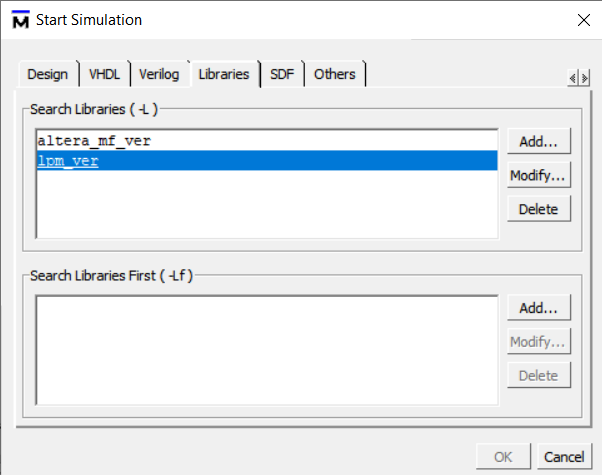


Рисунок 3. 11 - Добавленные библиотеки

Произвели запуск симуляции.

Командой *run -all* остановили симуляцию по строке «#1000 $stop;». Далее снова запустили команду run-all, чтобы счётчик мог проработать до 255.

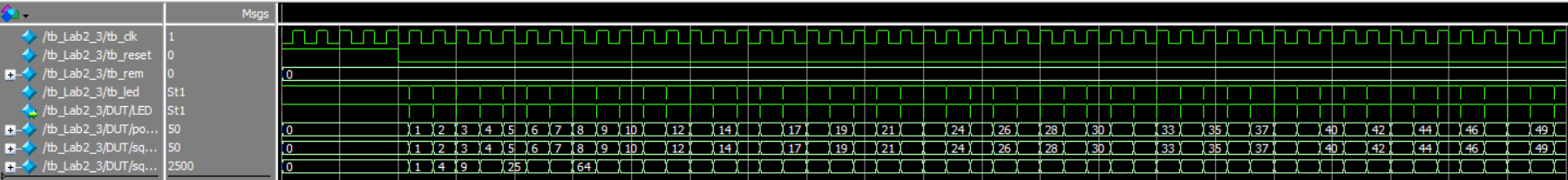


Рисунок 3. 12 - Результаты моделирования

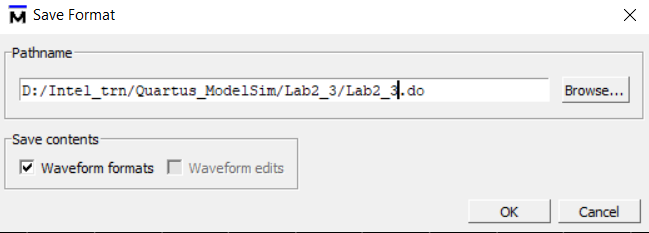


Рисунок 3. 13 - Сохранили, как файл Lab2\_3.do

## Выводы

В данной самостоятельной лабораторной работе были созданы модули с помощью библиотеки IP компонентов в Quartus Prime, при помощи простейшего описания на языке Verilog. Также был создан проект в ModelSim, в котором провели симуляция проекта, при помощи описанного на Verilog теста, взятого из Lab2\_1.