САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

Пакет ModelSim. Введение. Лабораторная работа 1.

Дисциплина

«Технологии проектирования аппаратных средств компьютерных систем»

Выполнил:

Бараев Д.Р.

Группа: 3540901/02001

Преподаватель: А. П. Антонов

Санкт-Петербург

2021

Оглавление

[1 Lab 1\_1 Основной процесс моделирования 4](#_Toc63689628)

[1.1 Задание 4](#_Toc63689629)

[1.2 Создание рабочей папки 4](#_Toc63689630)

[1.3 Скомпилируйте элементы проекта 4](#_Toc63689631)

[1.4 Запуск моделирования 5](#_Toc63689632)

[1.5 Установите точки останова и перейдите по исходному тексту 6](#_Toc63689633)

[1.6 Анализ сигналов 8](#_Toc63689634)

[1.7 Масштабирование отображения сигнала 8](#_Toc63689635)

[1.8 Использование курсоров в окне волны 9](#_Toc63689636)

[1.9 Выводы 10](#_Toc63689637)

[2 Lab 1\_2 Поток проектов 10](#_Toc63689638)

[2.1 Задание 10](#_Toc63689639)

[2.2 Создание проекта 10](#_Toc63689640)

[2.3 Компиляция 10](#_Toc63689641)

[2.4 Добавление папок 11](#_Toc63689642)

[2.5 Перемещение файлов в папки 11](#_Toc63689643)

[2.6 Использование конфигураций моделирования 11](#_Toc63689644)

[2.7 Выводы 12](#_Toc63689645)

[3 Lab 1\_3 Работа с несколькими библиотеками 12](#_Toc63689646)

[3.1 Задание 12](#_Toc63689647)

[3.2 Создание проекта 14](#_Toc63689648)

[3.3 Загрузка без связывания библиотек 15](#_Toc63689649)

[3.4 Ссылка на библиотеку ресурсов 15](#_Toc63689650)

[4 Lab 1\_4 Автоматизация моделирования 16](#_Toc63689651)

[4.1 Создание простого файла DO 16](#_Toc63689652)

[4.2 Запуск в режиме командной строки 17](#_Toc63689653)

[4.3 Tcl-конструкции 19](#_Toc63689654)

[4.4 Выводы 20](#_Toc63689655)

[5 Lab 1\_5 Просмотр и инициализация памяти 20](#_Toc63689656)

[5.1 Задание 20](#_Toc63689657)

[5.2 Создание библиотеки и файлов памяти 20](#_Toc63689658)

[5.3 Экспорт данных памяти в файл 21](#_Toc63689659)

[5.4 Изменение данных 23](#_Toc63689660)

[5.5 Выводы 23](#_Toc63689661)

# Lab 1\_1 Основной процесс моделирования

## Задание

* Cоздайте рабочую библиотеку проекта
* Скомпилируйте элементы проекта
* Загрузить проект
* Запустите моделирование.

## Создание рабочей папки

Прежде чем смоделировать дизайн, сначала создаем библиотеку и компилируем исходный код в нее.

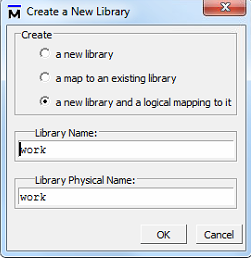


Рисунок 1. 1 - Создание библиотеки

## Скомпилируйте элементы проекта

Создав рабочую библиотеку, мы готовы скомпилировать исходные файлы.

1. Скомпилируйте counter.v и tcounter.v
2. Проверка скомпилированных элементы проекта.

В окне библиотеки нажимаем значок «+» рядом с *рабочей* библиотекой и видим скомпилированные элементы.

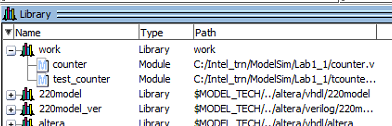


Рисунок 1. 2 - Скомпилированные counter.v и tcounter.v в библиотеке

Теперь произведем загрузку проекта в симулятор.

1. Загрузите модуль *test\_counter* в симулятор.

Когда дизайн загрузится, откроется окно структуры (помечено как **sim** ), Objects, Processes, Wave.

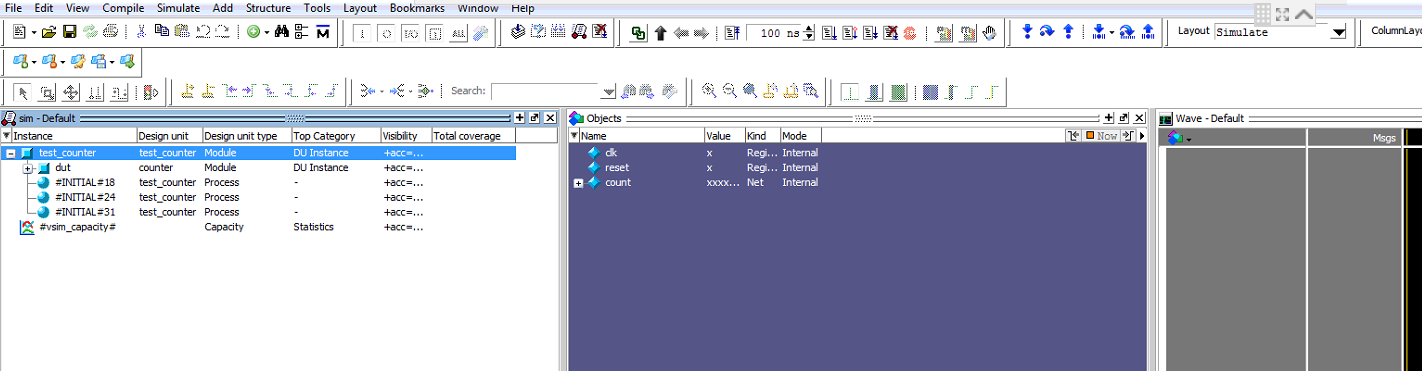


Рисунок 1. 3 - Окно структуры

## Запуск моделирования

1. Добавить сигналы в окно *Wave*.

* В окне структуры (sim) щелкните правой кнопкой мыши *test\_counter,* чтобы открыть всплывающее контекстное меню.
* Нажмите *Add Wave*

Все сигналы в проекте добавляются в окно Wave.

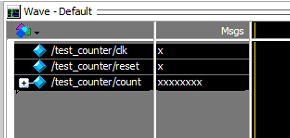


Рисунок 1. 4 - Появившиеся сигналы в окне Wave

1. Запустите симуляцию.

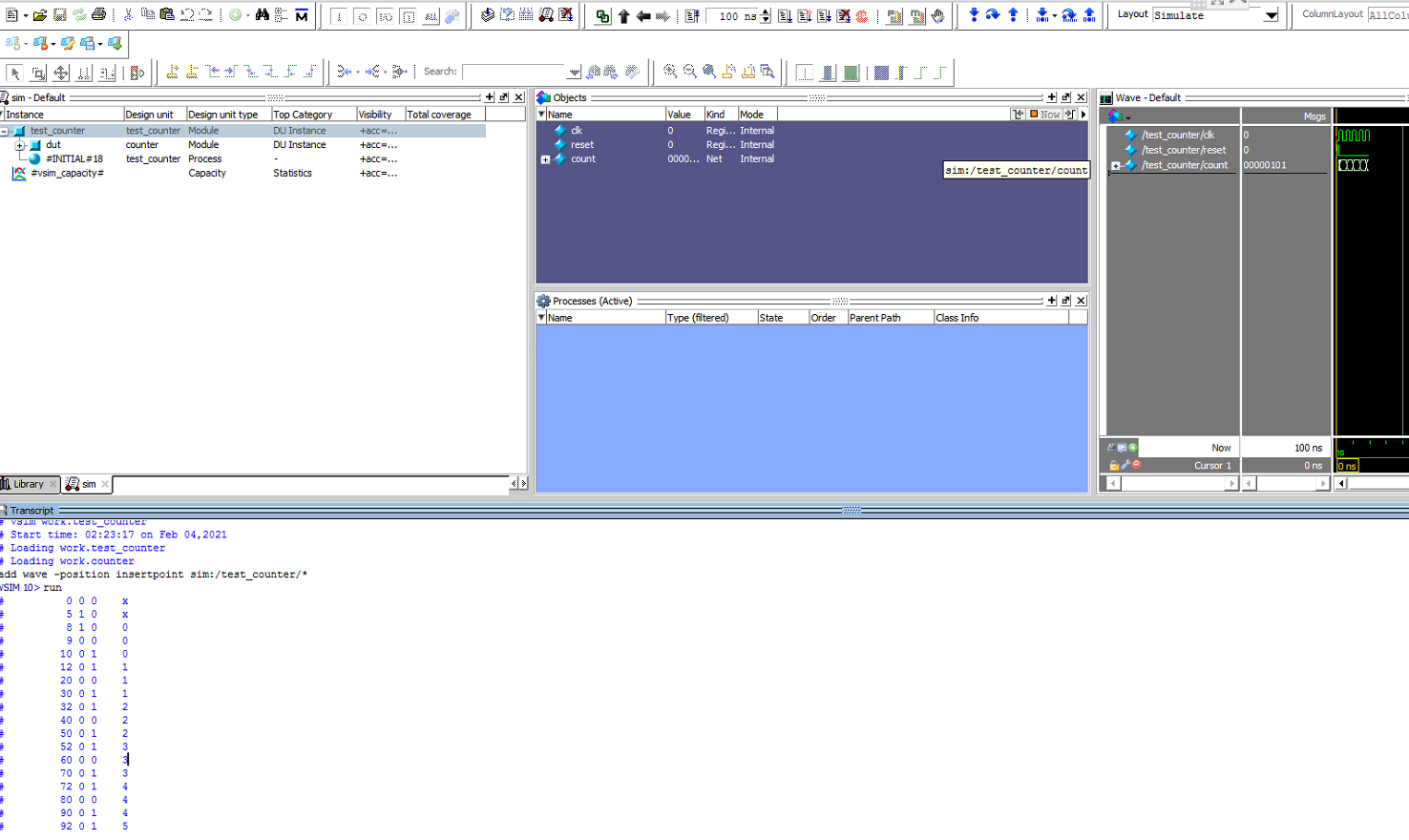


Рисунок 1. 5 - Результаты симуляции при 100ns

1. Введите run 500 в строке *VSIM>* окна *Transcript.*

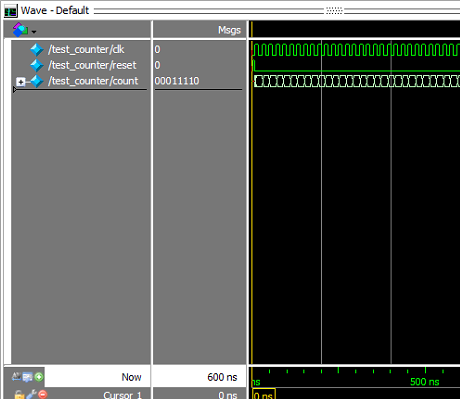


Рисунок 1. 6 - Моделирование при 600нс

Моделирование продвигается еще на 500 нс до 600 нс.

При нажатии кнопки *Run -All* симуляция продолжается до тех пор, пока вы не выполните команду *Break* или не встретите оператор в вашем коде (т.е. оператор Verilog $ stop), который останавливает моделирование.

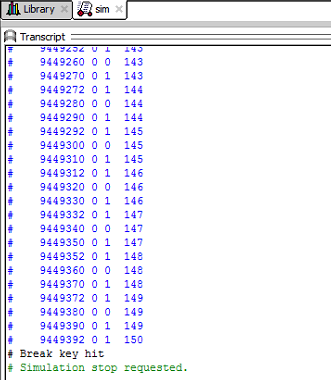


Рисунок 1. 7- Симуляция до Break

## Установите точки останова и перейдите по исходному тексту

Далее кратко рассматриваем одну интерактивную функцию отладки в среде ModelSim. Устанавливаем точку останова в окне исходного кода, запускаем моделирование, а затем выполняется пошаговое выполнение проекта в тестовом задании. Точки останова могут быть установлены только на исполняемых строках, которые обозначены красными номерами строк.

1. Откройте *counter.v* в окне исходного кода.

* Выберите *View > Files***,** чтобы открыть окно «Файлы».
* Щелкните значок + рядом с именем файла sim, чтобы просмотреть содержимое набора данных *vsim.wlf.*

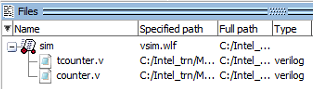


Рисунок 1. 8 - Содержимое vsim.wlf

Дважды щелкните *counter.v* (или *counter.vhd,* если вы имитируете файлы VHDL), чтобы открыть файл в окне исходного кода.

1. Установите точку останова в строке 36 *counter.v*

* Прокрутите до строки 36 и щелкните столбец Ln # (номер строки) рядом с номером строки.

Красная точка появляется в столбце номера строки в строке номер 36, указывая на то [,](https://translate.googleusercontent.com/translate_f#11) что точка останова установлен.

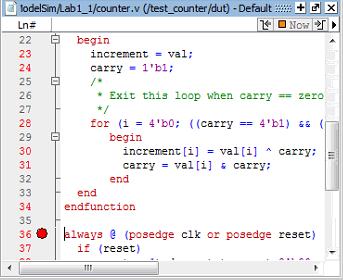


Рисунок 1. 9 - Установка точки останова

1. Отключите, включите и удалите точку останова.

* Щелкните красную точку, чтобы отключить точку останова. Он станет серой точкой.
* Щелкните серую точку еще раз, чтобы снова включить точку останова. Он станет красной точкой.
* Щелкните красную точку правой кнопкой мыши и выберите Удалить точку останова 36.
* Щелкните столбец номера строки рядом с номером строки 36 еще раз, чтобы воссоздать точку останова.

1. Перезапустите симуляцию.

Симуляция выполняется до тех пор, пока не будет достигнута точка останова. Когда симуляция достигает точки останова, она перестает работать, выделяет линию синей стрелкой в коде.

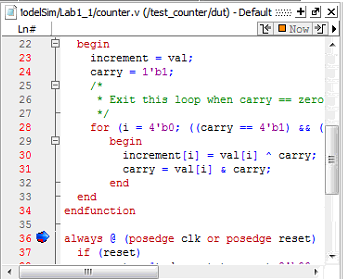


Рисунок 1. 10 - Синяя стрелка указывает место остановки моделирования

1. Перед тем, как продолжить, вам необходимо завершить текущее моделирование.

## Анализ сигналов

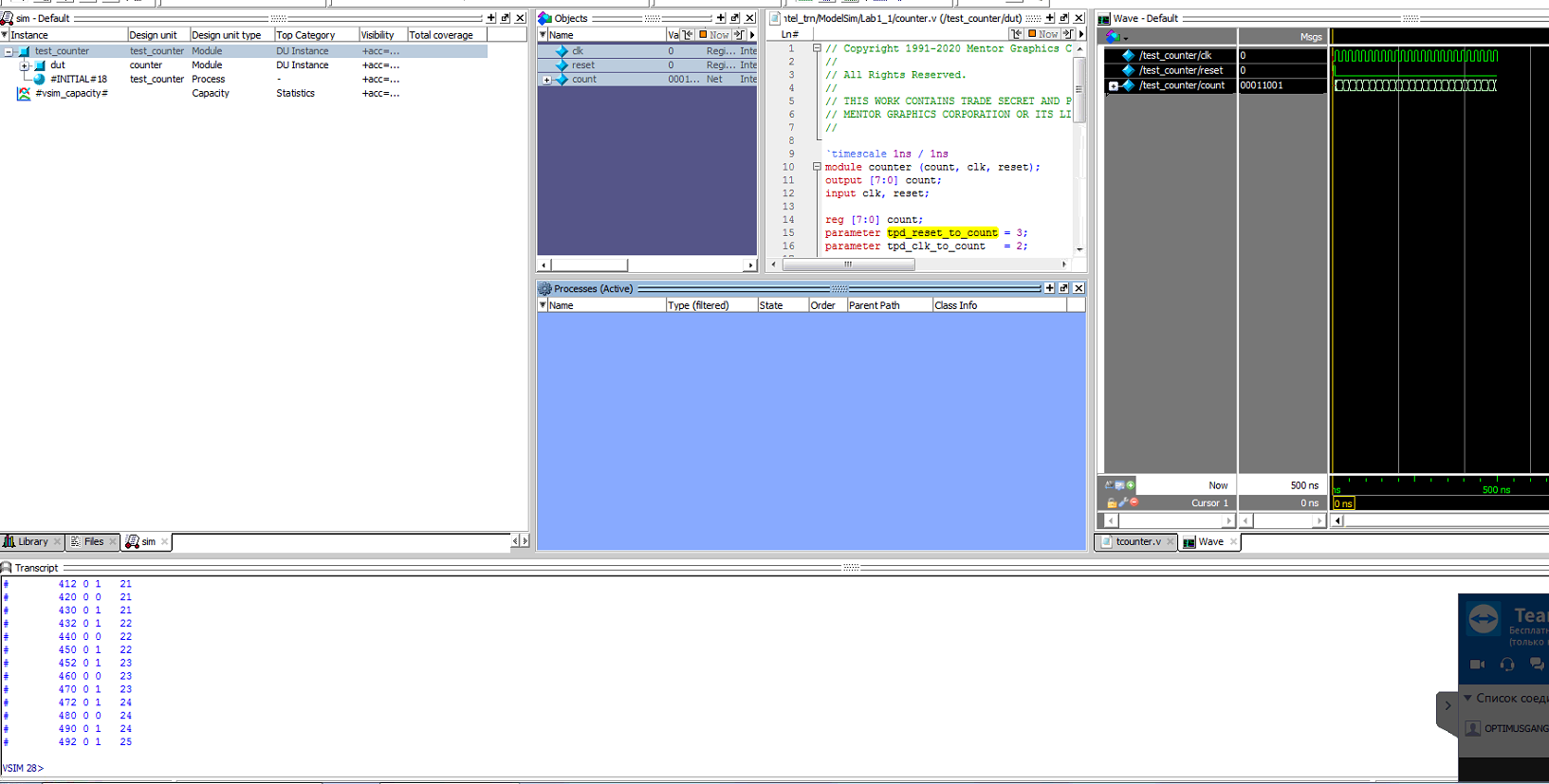


Рисунок 1. 11 - Симуляция на 500 нс для показа формы сигналов

## Масштабирование отображения сигнала

Существует множество способов масштабирования отображения формы сигнала.

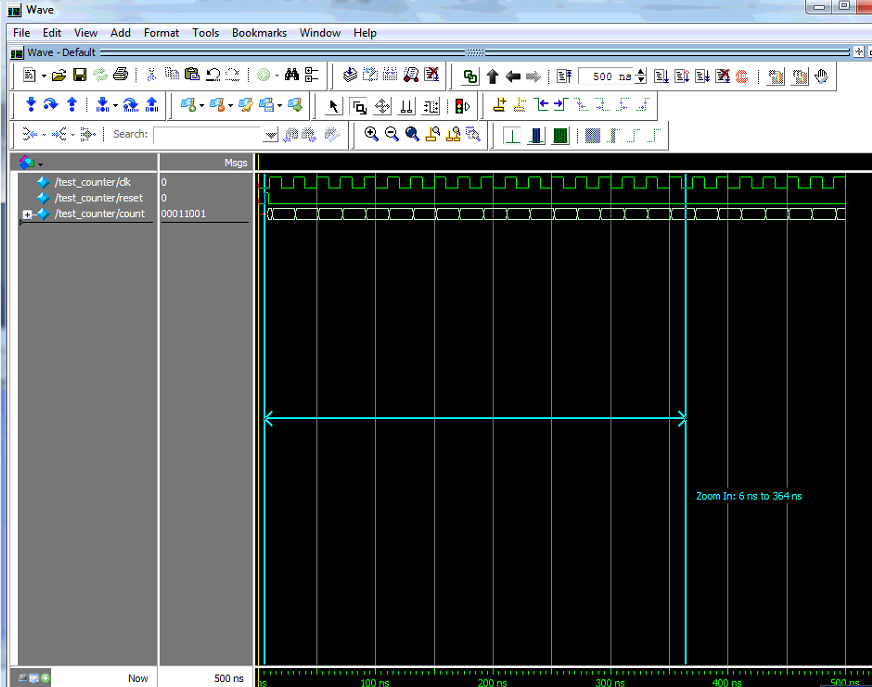


Рисунок 1. 12 - Режим масштабирования

Вы можете изменить Radix и Format для шин в окне Wave.

Значения счетчика шины будут отображаться как десятичные без знака.

## Использование курсоров в окне волны

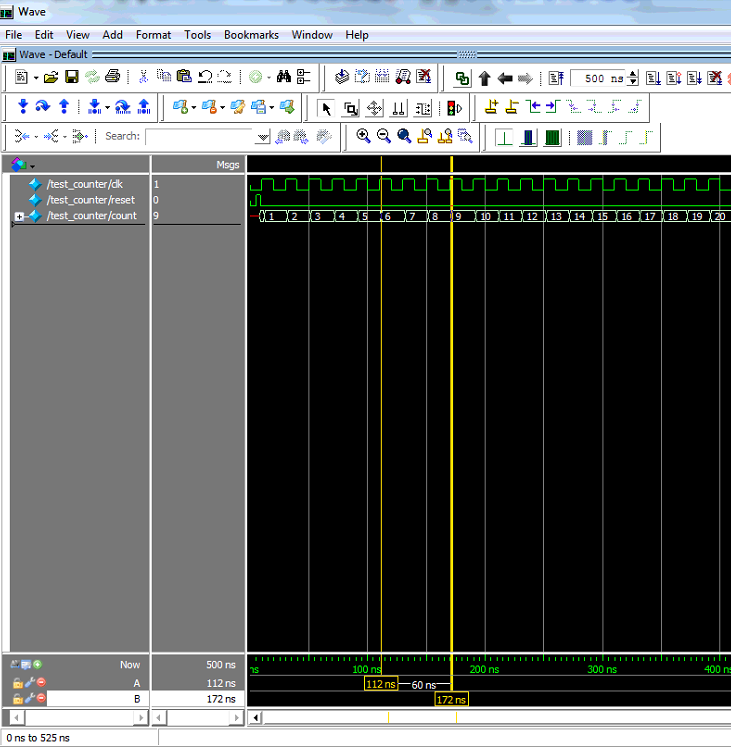


Рисунок 1. 13 - Добавление второго курсора

## Выводы

В ходе данной лабораторной работы ознакомились с пакетом ModelSim. Обучились созданию библиотек в ModelSim. Также была осуществлена загрузка файлов в библиотеку, их компиляция, был получен опыт работы со схемами симуляции (Wave). ModelSim имеет множество функций по работе как с окном симуляции, для удобства пользователя, так и для проведения симуляции в целом.

# Lab 1\_2 Поток проектов

## Задание

В этой лабораторной работе будет осуществлена практика в создании проекта. Как минимум, проекты содержат рабочую библиотеку и состояние сеанса, которое хранится в файле .mpf.

Проект также может состоять из:

* Исходные файлы HDL или ссылки на исходные файлы
* другие файлы, такие как README или другая документация по проекту
* библиотеки
* ссылки на глобальные библиотеки

## Создание проекта

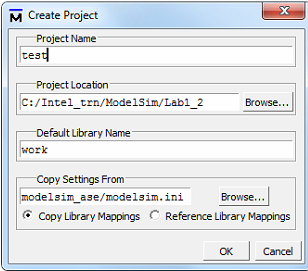


Рисунок 2. 1 - Создание проекта

## Компиляция

Задав настройки проекта и добавив в проект объекты, переходим к компиляции.

ModelSim компилирует оба файла и меняет символ в столбце «Состояние» на зеленую галочку, означающая, что компиляция прошла успешно. В случае сбоя компиляции символом будет красный значок «X», и будет выведено сообщение об ошибке.

## Добавление папок

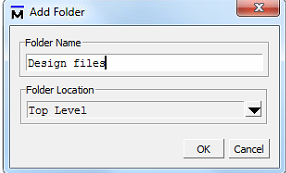


Рисунок 2. 2 - Окно добавления папок

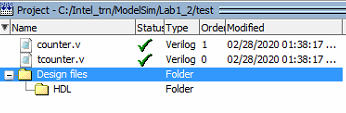


Рисунок 2. 3 - Добавление подпапки

## Перемещение файлов в папки

1. Переместите *tcounter.v* и *counter.v* в папку *HDL.*

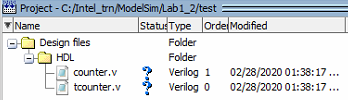


Рисунок 2. 4 - Перемещение файлов в папку HDL

Выбранные файлы перемещаются в папку HDL. Щелкните значок «+» рядом с папкой HDL, чтобы просмотреть файлы.

Теперь файлы отмечены знаком "?" в столбце Статус, потому что были перемещены файлы. Проект теперь не понимает, действительна ли предыдущая компиляция.

## Использование конфигураций моделирования

1. Создайте новую конфигурацию моделирования.

В окне проекта теперь отображается счетчик конфигурации моделирования в папке HDL

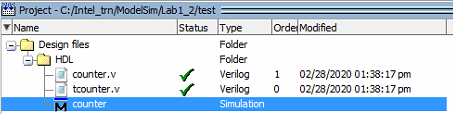


Рисунок 2. 5 - Конфигурации моделирования

1. Загрузите конфигурацию моделирования.

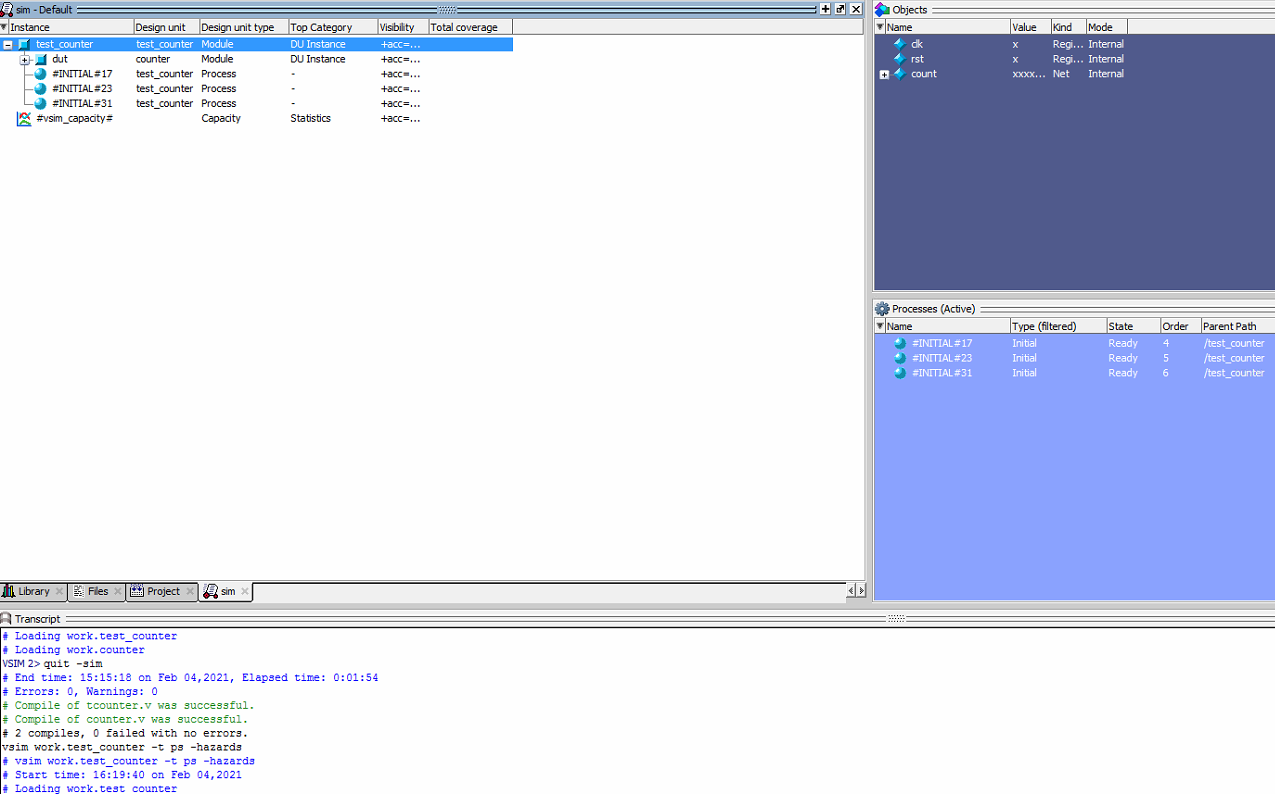


Рисунок 2. 6 - Загрузка конфигурации моделирования

## Выводы

В ходе лабораторной работы ознакомились с созданием проектов в ModelSim. Получен опыт создания проекта, добавления в него файлов и расположения их по папкам. Также была создана специальная конфигурация для файла counter, которая при запуске симуляции будет всегда работать, как было указано ранее.

# Lab 1\_3 Работа с несколькими библиотеками

## Задание

В данной лабораторной работе будет произведена работа с несколькими библиотеками. Возможно несколько библиотек для организации проекта, для доступа к IP из стороннего источника или для совместного использования общих частей между симуляциями.

* Создается библиотека ресурсов, содержащая конструктор счетчика
* Создать проект и скомпилировать в него тестовый стенд
* Связаться с библиотекой, содержащая счетчик
* Запустите моделирование.

Используются файлы Verilog counter.v и tcounter.v . Эти файлы находятся в *C:\Intel\_trn\ModelSim\Lab1\_3*. Это рабочая папка для Lab1\_3.

1. Создайте каталог для библиотеки ресурсов.
2. Создайте каталог для тестового стенда.

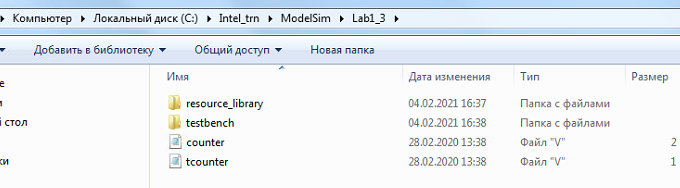


Рисунок 3. 1 - Создание отдельных директорий для counter и tcounter

Убедитесь, что для файла modelsim.ini в каталоге установки установлено значение «Только для чтения». Это предотвращает постоянное отображение библиотек ресурсов в файле modelsim.ini.

1. Создайте библиотеку ресурсов

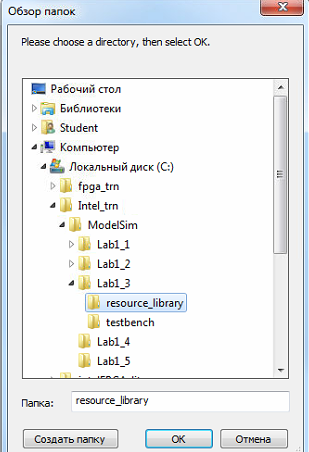


Рисунок 3. 2 – Изменение рабочей директории

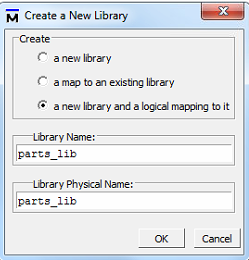


Рисунок 3. 3 - Создание новой библиотеки

1. Скомпилируйте счетчик в библиотеку ресурсов

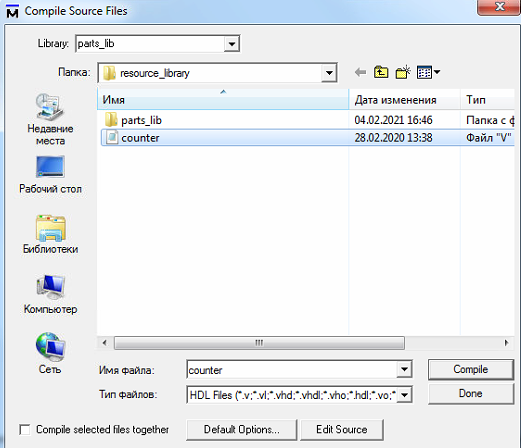


Рисунок 3. 4 - Компиляция счетчика в библиотеку

## Создание проекта

1. Создайте проект
2. Добавьте в проект испытательный стенд

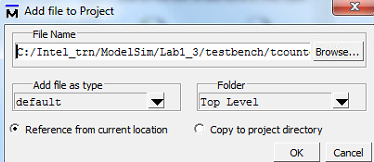


Рисунок 3. 5 - Добавление файлов в проект

1. Скомпилируйте тестовый стенд

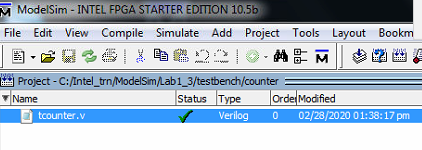


Рисунок 3. 6 - Компиляция

## Загрузка без связывания библиотек

Требуется сделать ссылку на библиотеку parts\_lib, которая была создана ранее.

1. Загрузите проект Verilog без ссылки на библиотекой.

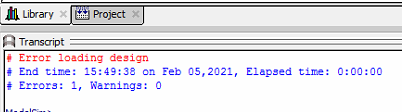


Рисунок 3. 7 - Сообщение об ошибке загрузки

## Ссылка на библиотеку ресурсов

1. Укажите библиотеку поиска во время моделирования.

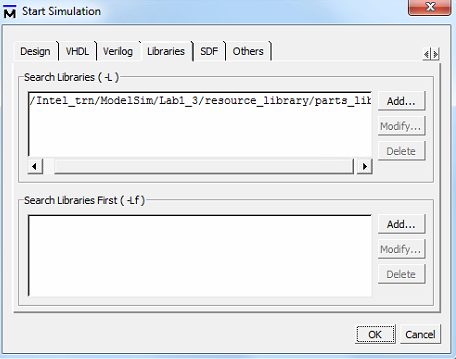


Рисунок 3. 8 - Путь библиотеки

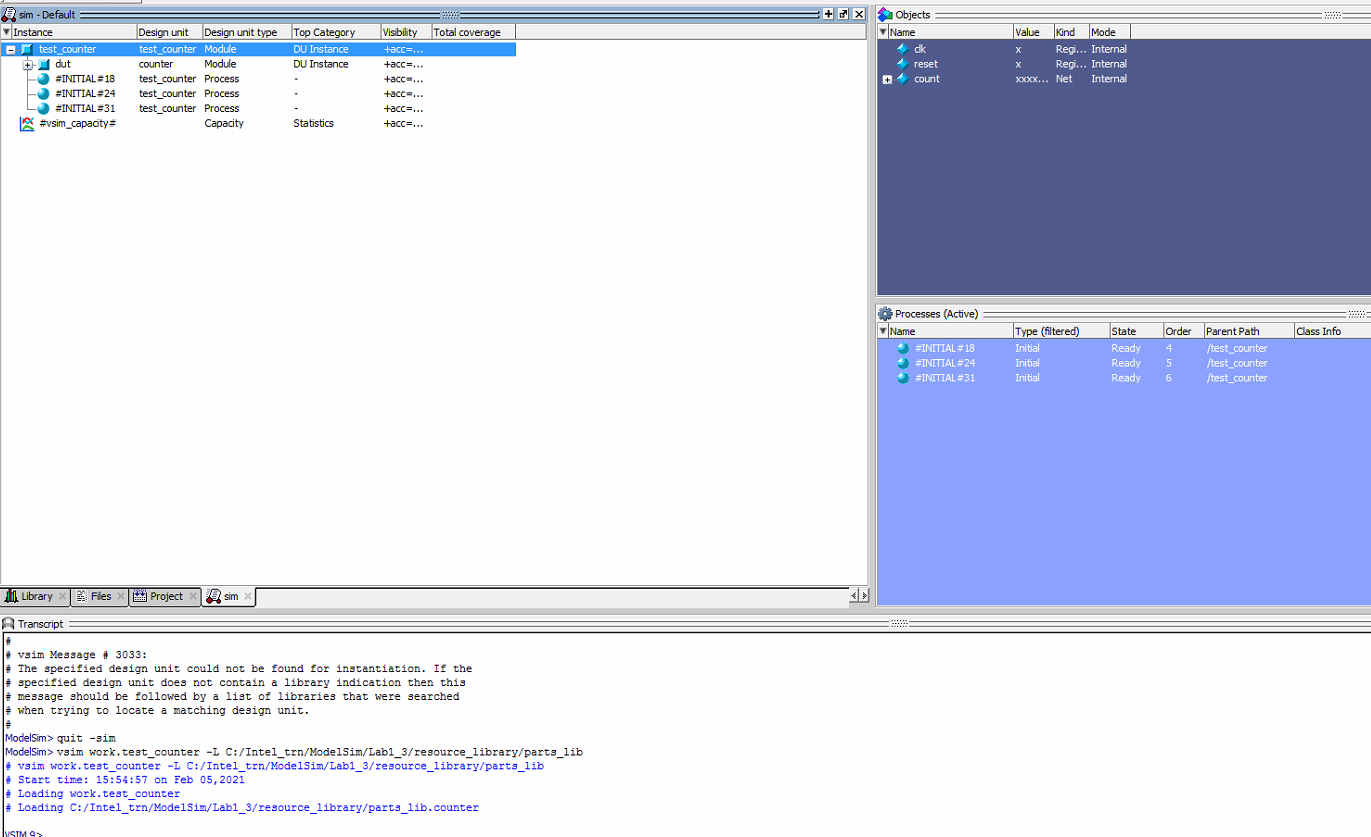


Рисунок 3. 9 - Конструкция загружается без ошибок

# Lab 1\_4 Автоматизация моделирования

## Создание простого файла DO

В данной лабораторной работе создается DO файл, который загружает дизайн, добавляет сигналы в окно Wave, предоставляет стимул для этих сигналов, а затем продвигает моделирование. Также можно создать файл DO из сохраненного файла расшифровки.

1. Создайте файл DO, который будет добавлять сигналы в окно Wave, форсировать сигналы и запускать моделирование.
2. Сохраните файл.
3. Запустите файл DO командой do sim.do.

ModelSim загружает дизайн, выполняет сохраненные команды и рисует волны в окне Wave.

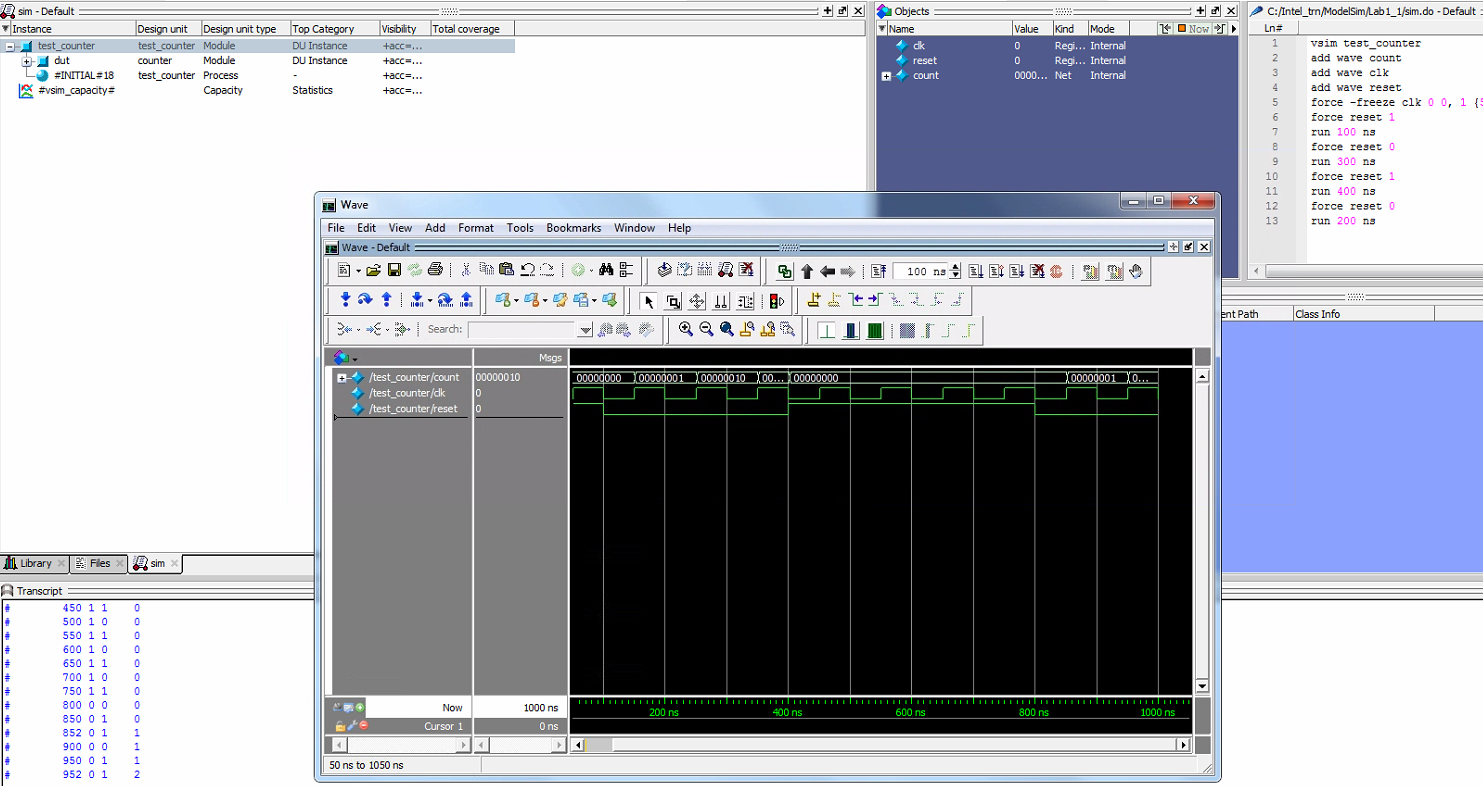


Рисунок 4. 1 - Запуск файла DO

## Запуск в режиме командной строки

1. В поле поиска рядом с кнопкой «Пуск» введите cmd и запустите окно командной строки.
2. В окне командной строки введите команду: cd C:\Intel\_trn\ModelSim\Lab1\_4
3. Создайте новую библиотеку дизайна и скомпилируйте исходный файл.

* Введите **vlib work** в командной строке.
* Для Verilog введите vlog counter.v в командной строке.

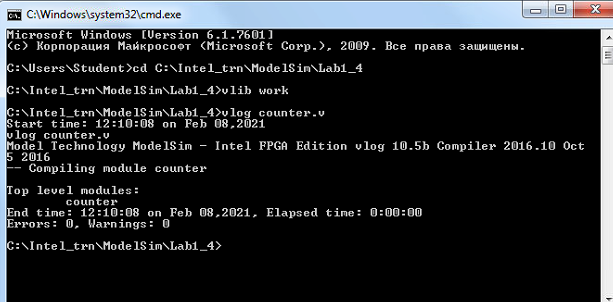


Рисунок 4. 2 - Создание новой библиотеки и компилирование исходного файла

1. Создайте файл DO.

Сохраните файл с именем sim.do и поместите его в C:\Intel\_trn\ModelSim\Lab1\_4

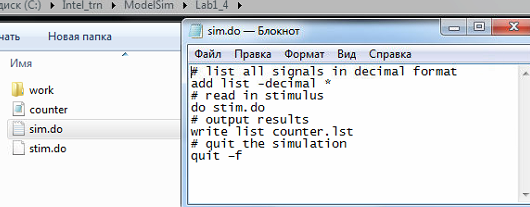


Рисунок 4. 3 - Созданный DO файл

1. Запустите моделирование в режиме командной строки.

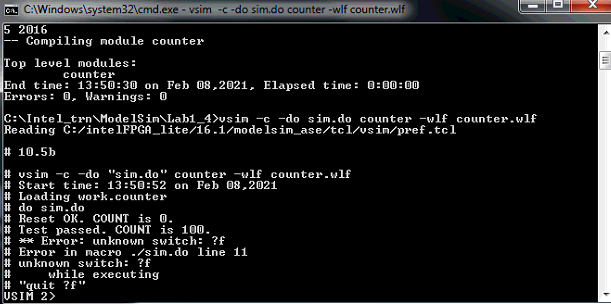


Рисунок 4. 4 – Моделирование

В процессе моделирование возникает ошибка «quit -f».

Файл counter.lst полностью соответствует представленному на изображениях в задании.

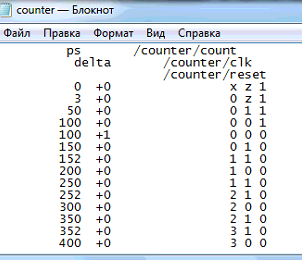


Рисунок 4. 5 - Содержимое файла counter.lst

## Tcl-конструкции

В данном пункте создается Tcl-конструкции, которые проверяют определенные значения сигнала, а затем добавляют закладки, которые увеличивают масштаб окна Wave, когда это значение существует. Закладки позволяют сохранять определенный диапазон масштабирования и положение прокрутки в окне Wave.

1. Создайте конструкцию, где нужно вставить код, указанный в методическом указании.
2. Сохраните скрипт с именем add\_bkmrk.do в C: \ Intel\_trn \ ModelSim \ Lab1\_1
3. Загрузите конструкторский модуль *test\_counter* и убедитесь, что для системы счисления установлено двоичное значение.
4. Запустите файл DO и запустите проект.

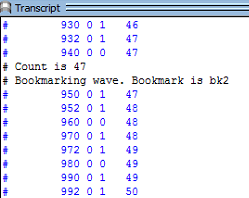


Рисунок 4. 6 - Запуск моделирования

Запуск этого файла в среде ModelSim вызвал симуляцию, в которой уже присутствуют закладки bk1 450->600 и bk2 850->1000.

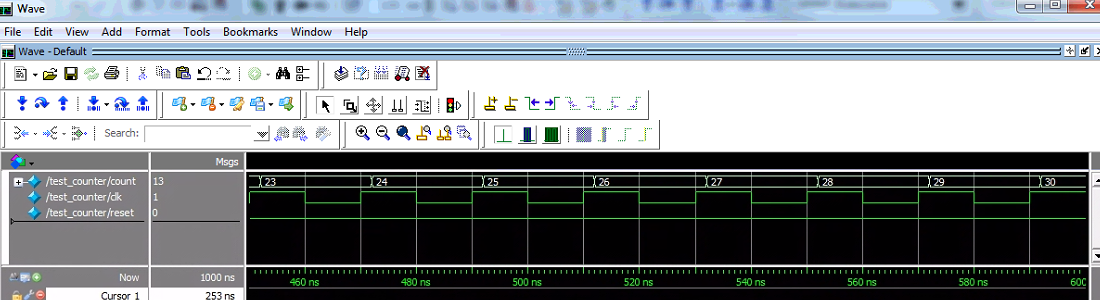


Рисунок 4. 7 - Закладка bk1 450->600

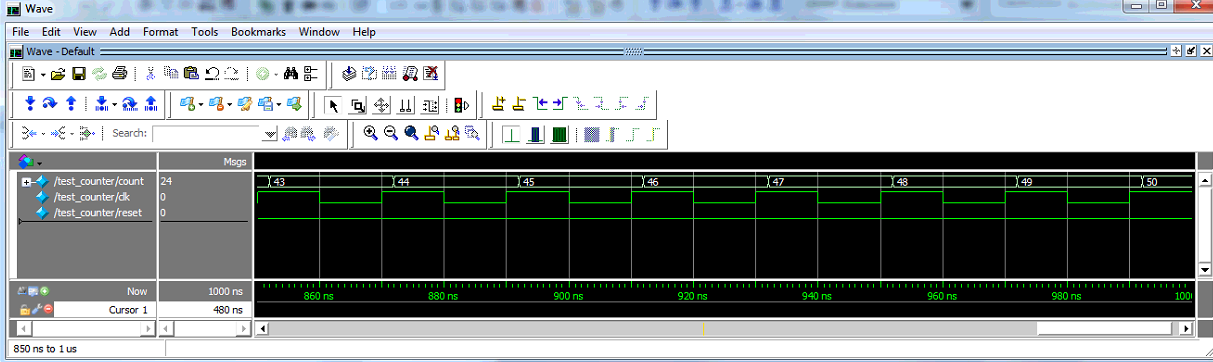


Рисунок 4. 8 - Закладка bk2 850->1000

## Выводы

В ходе данной лабораторной работы ознакомились как с do-файлами, запуском в ModelSim, так и с работой ModelSim через командную строку Windows. Команды позволяют создавать рабочие библиотеки, запускать do-файлы, запускать wlf файлы для проверки их напрямую.

# Lab 1\_5 Просмотр и инициализация памяти

## Задание

В ходе работы нужно научиться смотреть и инициализировать память в пакете ModelSim, который может воспринимать как память массивы целых чисел, а также массивы reg и wire.

## Создание библиотеки и файлов памяти

При помощи команд vlib work и vlog \*.v создана рабочая библиотека, а также файлы в ней. Произвели компиляцию. При открытии файла ram\_tb появилась возможность взаимодействовать с Memory List.

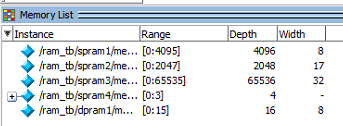


Рисунок 5. 1 - Окно Memory List

Для каждого элемента будет открываться свои данные памяти. Запуск симуляции произведет взаимодействие с данными памяти.

После команды run -all, записываются данные в spram1/mem.

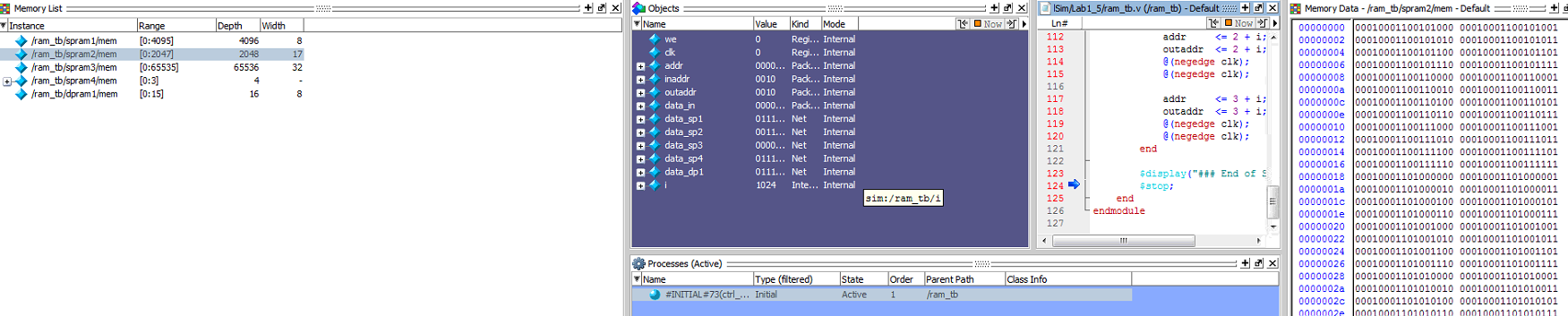


Рисунок 5. 2 - Список данных в spram1/mem

Представление данных изменяются с помощью инструментария пакета. Двоичные значения можно перевести в десятичную систему, а также в строке можно отображать одно значение. С помощью команд Goto и Find можно переходить к нужным строкам, или находить необходимые данные.

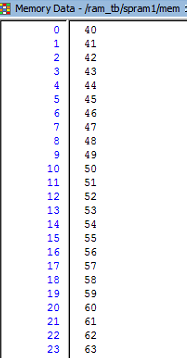


Рисунок 5. 3 - Изменение представления данных

## Экспорт данных памяти в файл

С помощью команды File>Export>Memory Data… можно экспортировать данные с памятью в другой файл. При этом возможно задать свои конфигурации для файла (отображение строк и систему счисления). Для релокации данных нужно поставить галочку на пункте “No addresses”

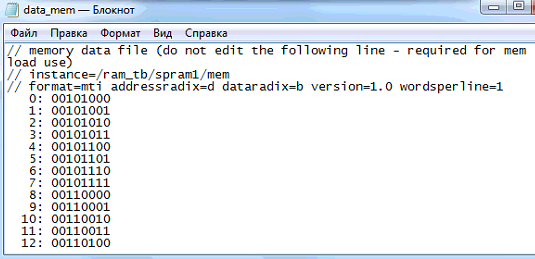


Рисунок 5. 4 - Экспортированный файл data\_mem.mem

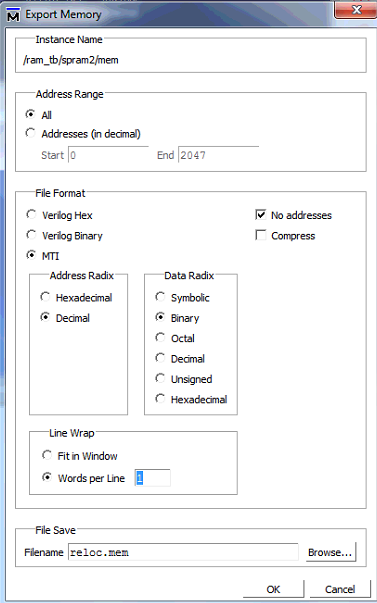


Рисунок 5. 5 - Конфигурация файла памяти без конкретного адреса

На рисунке 5-6 продемонстрированы данные spram3/mem после загрузки в них данных из reloc.mem. Данные соответствуют условиям задания.

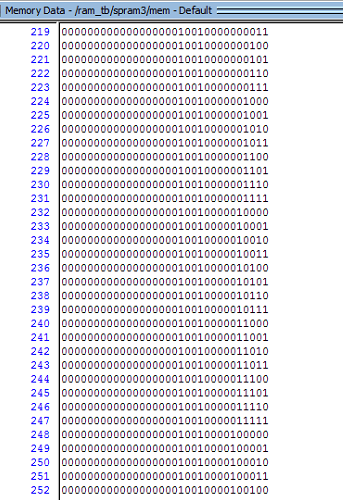


Рисунок 5. 6 - spram3/mem с загруженными данными

## Изменение данных

Существует возможность изменять значения в файлах с данными памяти. К примеру, это можно задать через окно Change Memory.

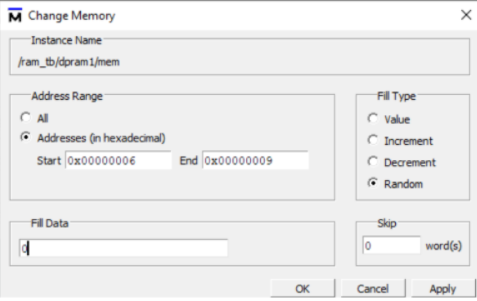


Рисунок 5. 7 - Изменение данных в dpram1/mem

## Выводы

В ходе данной лабораторной работы ознакомились с файлами памяти в ModelSim. Создали «память», осмотрены файлы с данными, а также освоены способы экспорта и импорта данных из одного файла в другой. Также ознакомились с возможностью изменять данные.