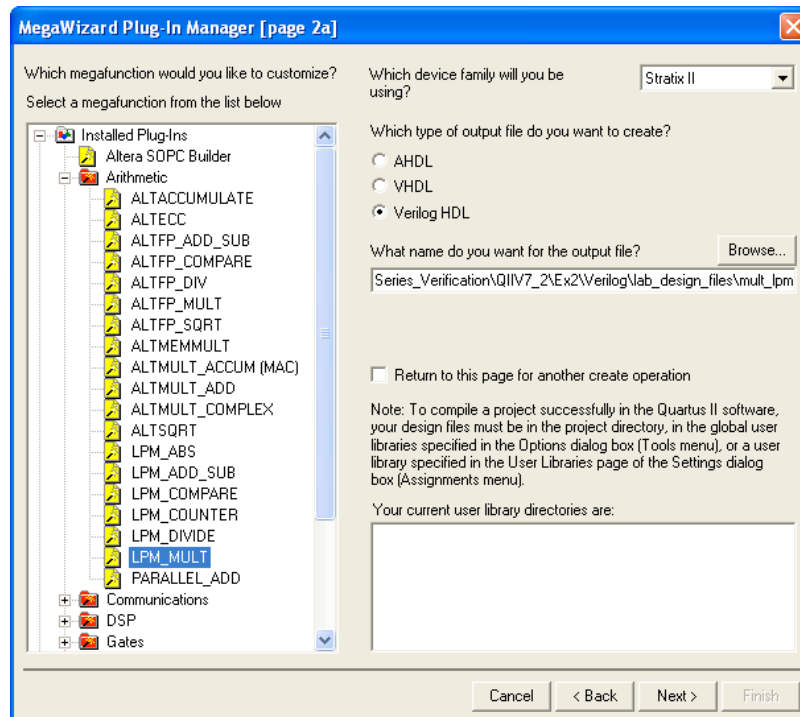


**Цели работы:**

- *Получить навыки и знания необходимые для выполнения всех шагов функциональной симуляции с мегафункциями Altera*
- *Получить навыки создания ссылок на библиотеки*
- *Получить навыки симулирования в ModelSim при помощи тест-бенчей*

## Шаг 1: Создание LPM мегафункций Altera при помощи MegaWizard Plug-In Manager

1. Запустите MegaWizard Plug-In Manager. MegaWizard может быть запущен как из меню Tools пакета Quartus II, так и отдельно из директории установки пакета Quartus II путем исполнения файла `\quartus\bin\qmegawiz.exe`
2. На странице 1 Megawizard выберите **Create a new custom megafunction variation**. Щелкните **Next**.



3. На стринца 2а, выберите функцию **LPM\_MULT**.
4. Установите тип выходного файла - **VHDL**.

5. В разделе указания имени файла, нажмите кнопку **Browse** и выберите соответствующую директорию для сохранения генерируемых файлов, в качестве имени файла используйте **mult\_lpm**:

**VHDL:**

lab2\lab\_design\_files\mult\_lpm

6. Установите тип целевой микросхемы в **Stratix II** или любой другой тип микросхемы, содержащей внутренние аппаратные блоки умножения. Нажмите **Next**.
7. На странице 3, установите параметр **dataa** в значение **8** бит и параметр **datab** в значение **3** бита. Дважды нажмите **Next**.
8. На странице 5, выберите пункт **Yes, I want an output latency of 1 clock cycles**. Нажмите **Next**.
9. На странице 6, обратите внимание на файл имитационной модели, необходимый для моделирования данной мегафункции. Нажмите **Next**.
10. На странице 7 **Summary**, убедитесь в том что отмечены все позиции. Нажмите **Finish**.
11. Заново запустите MegaWizard Plug-In Manager для создания другой мегафункции.
12. На странице 2а, выберите мегафункцию **ROM: 1-PORT**. Установите необходимый тип выходного файла, и выберите тип микросхемы аналогичный выбранному для мегафункции умножителя. В примере - **Stratix II**.
13. Назовите новую мегафункцию **hvalues\_mf**, и установите выходную директорию в ту же папку **lab\_design\_files** что и для мегафункции умножителя. нажмите **Next**.
14. На странице 3, установите разрядность шины **q** в **3** бита.
15. Установите количество трехбитных слов равным **4**. Значение 4 необходимо впечатать вручную в соответствующее поле. Дважды нажмите **Next**.
16. На странице 5, выберите в опцию инициализации памяти файлом и укажите на файл **hvalues\_mf\_init.hex** находящейся в проектной директории lab2. Нажмите **Next**.
17. На странице 6, обратите внимание на файл имитационной модели, необходимый для моделирования данной мегафункции. Нажмите **Next**.
18. На странице 7 **Summary**, убедитесь в том что отмечены все позиции. Нажмите **Finish**.

## Шаг 2: Создание дополнительной библиотеки в ModelSim

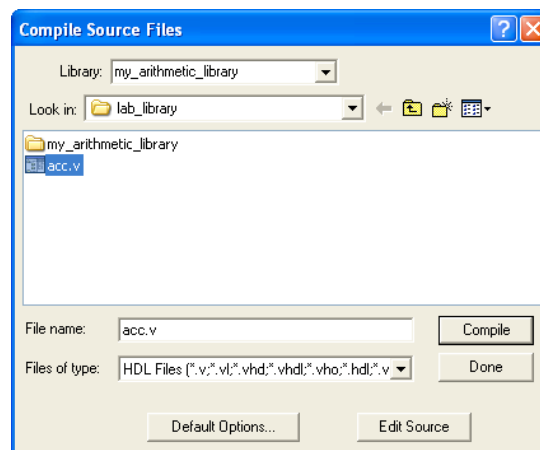
На данном шаге производится создание дополнительной пользовательской библиотеки и компиляция в неё исходных файлов. Библиотека позволяет удобно организовывать хранение прекомпилированных элементов, без необходимости каждый раз добавлять необходимые файлы в рабочий проект. В библиотеку будет откомпилирован элемент **acc.vhd** и библиотека будет подключена к рабочему проекту.

Обычно, во всех сторонних средствах моделирования, необходимо выполнять подобную процедуру для создания библиотек поддержки моделирования мегафункций Altera. Однако, при использовании ModelSim-Altera, данные библиотеки уже созданы и корректным образом подключены к среде моделирования.

1. Запустите ModelSim.
2. Из меню **File** выберите команду **Change Directory**.
3. Выберите директорию **lab\_library** в директории проекта **lab2**. Щелкните **OK**. Эта процедура закроет текущий открытый проект.
4. Из меню **File** выберите пункт **New** и выберите позицию **Library**.
5. Назовите библиотеку **my\_arithmetic\_library**. Пункт **Library Physical Name** обновится автоматически. Щелкните **OK**.

Теперь новая библиотека должна появиться в списке библиотек окна **Workspace**. Следующим шагом в неё будут скомпилированы библиотечные элементы.

6. Из меню **Compile** выберите пункт **Compile**.



7. Выберите, куда должны быть скомпилированы проектные элементы. Пункт **Library** установите в значение **my\_arithmetic\_library**.
8. Выберите элемент **acc.vhd** и нажмите кнопку **Compile**.
9. Нажмите **Done** когда компиляция закончится.

Можно убедиться в том, что файл успешно откомпилирован в нужную библиотеку раскрыв позицию **my\_arithmetic\_library** в списке библиотек в окне **Workspace**.

10. Из меню **File** вновь выполните команду **Change Directory**.

11. Выберите директорию **lab\_design\_files** в которую были сохранены созданные в шаге 1 мегафункции Altera.

### Шаг 3: Создание нового проекта в ModelSim и подключение к нему библиотеки

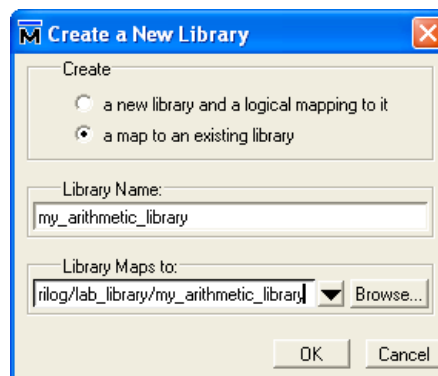
1. Из меню **File** выберите пункт **New** и выберите позицию **Project**.
2. Назовите новый проект(в примере **my\_project**). Так как выполнена команда смены директории, то опция **Project Location** должна указывать на **lab\_design\_files**.
3. Оставьте имя в опции **Default Library Name** так как оно задано по умолчанию - **work**. Щелкните **OK**.

*Все остальные файлы которые будут компилироваться (кроме файла **acc**) будут компилироваться в библиотеку **work library**.*

4. В открывшемся диалоге щелкните **Add Existing File**. Щелкните **Browse** для того, чтобы выбрать и добавить файлы в проект.
5. Удерживая клавишу **shift** или **ctrl**, выберите следующие HDL файлы:

VHDL
<b>filter.vht (testbench)</b>
<b>filter.vhd</b>
<b>hvalues_mf.vhd</b>
<b>mult_lpm.vhd</b>
<b>state_m.vhd</b>
<b>taps.vhd</b>

6. Нажмите **Open** и затем **OK** для того, чтобы добавить файлы.
7. Закройте диалоговое окно **Add items to the Project**.
8. Создание ссылки на дополнительную библиотеку. Из меню **File** выберите пункт **New** и выберите позицию **Library**.



9. Выберите пункт **a map to an existing library**, и введите имя **my\_arithmetic\_library** в поле **Library Name**.
10. Щелкните **Browse** и выберите директорию **my\_arithmetic\_library** которая находится в директории **lab\_library**. Щелкните **OK**.

Для того чтобы проверить корректность подключения библиотеки следует открыть закладку **Library** окна **Workspace**. После подобного подключения библиотеки, в проекте будет доступен прекомпилированный элемент **acc** в библиотеке **my\_arithmetic\_library**. Остальные проектные файлы будут находится в рабочей библиотеке **work**.

#### Шаг 4: Компиляция файлов и запуск симулятора

Рассмотрим возможность организации файлов в проекте по отдельным папкам.

1. Щелкните правой кнопкой мыши в пустом пространстве закладки **Project** окна **Workspace**. Выберите пункт **Add to Project** и выберите позицию **Folder**.
2. Создайте папку с именем **filtertaps** в верхнем уровне **Top Level**.
3. Щелкните правой кнопкой мыши на файле **taps.vhd** ( закладка файлов окна **Workspace** ) выберите пункт **Properties**. В позиции **Place in Folder** выберите вновь созданную папку **filtertaps**. Щелкните **OK**.
4. Прodelайте аналогичную операцию для остальных файлов в проекте в соответствии с таблицей, представленной внизу:

File Name	Folder Name
mult_lpm.vhd	multiplication
state_m.vhd	statemachine
hvalues_mf.vhd	coefficients

5. Раскройте созданные папки в закладке **Project** окна **Workspace** для того, чтобы убедиться что все файлы успешно размещены по соответствующим папкам.
6. Из меню **Compile** выберите пункт **Compile All**.

Откомпилированные проектные элементы появятся в библиотеке **work** на закладке **Library** окна **Workspace**.

7. Из меню **Simulate** выберите пункт **Start Simulation**.
8. Установите параметр **Resolution** в **ns**. Раскройте библиотеку **work** на вкладке **Design** диалогового окна и выберите сущность **filter\_vhd\_vec\_tst (VHDL)**.
9. Перейдите на вкладку **Libraries** и, нажав кнопку **Add**, добавьте созданную вами библиотеку , содержащую проектный элемент **acc** в список библиотек. Щелкните **OK**.

#### Шаг 6: Создание конфигурации моделирования и запуск проекта на моделирование

Для того, чтобы не настраивать каждый раз опции моделирования при запуске некоего проекта можно создать так называемую конфигурацию моделирования, которая будет хранить всю необходимую информацию о настройках моделирования.

1. Выйдете из режима моделирования. Откройте меню **Simulate** и выберите пункт **End Simulation**.
2. Из меню **File** выберите пункт **Add to Project** и выберите позицию **Simulation Configuration**.

*Диалоговое окно **Add Simulation Configuration** аналогично диалоговому окну **Start Simulation** за исключением того, что вы можете создать конфигурацию моделирования и сохранить её в нужную директорию.*

3. Задайте имя конфигурации в поле **Simulation Configuration Name** (в примере **my\_configuration**).
4. Выберите тот же проектный элемент, тот же минимальный квант времени и добавьте необходимые библиотеки также как в шаге 5. Щелкните **OK**.

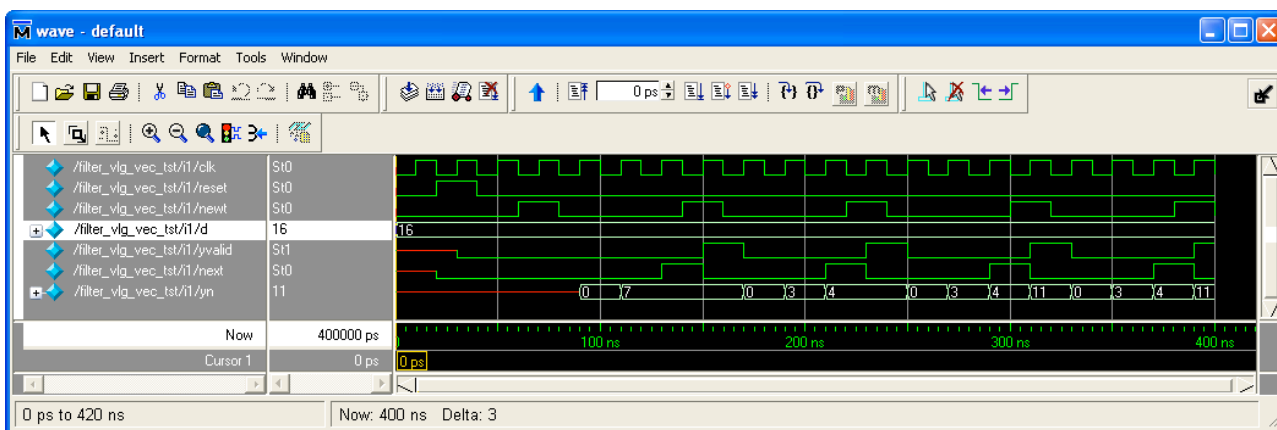
*Новая конфигурация моделирования появится в закладке **Project** окна **Workspace**.*

5. Дважды щёлкните на **my\_configuration** для запуска моделирования.
6. В закладке **sim** окна **Workspace** выберите и раскройте сущность **i1**.

*Тестбенч файл **filter.vht** состоит из трех тестовых модулей (или сущностей). Одна из этих сущностей, которую мы моделируем это **filter\_vhd\_vec\_tst**. Тестбенч вызывает проект **filter** и присваивает ему имя **i1**. Раскрыв сущность **i1** мы видим иерархию проектных модулей сущности **filter**, такую же как мы видели в лабораторной работе №1.*

7. И окна **Objects** перетащите в окно **Wave** следующие сигналы: **clk**, **reset**, **newt**, **d**, **yvalid**, **nxt**, и **yn**.
8. Запустите симуляцию до 400 нс.

*Нет необходимости пользоваться **.do** файлом с командами **force**. Тестбенч сам генерирует необходимые воздействия на сигналы.*



9. Работа завершена