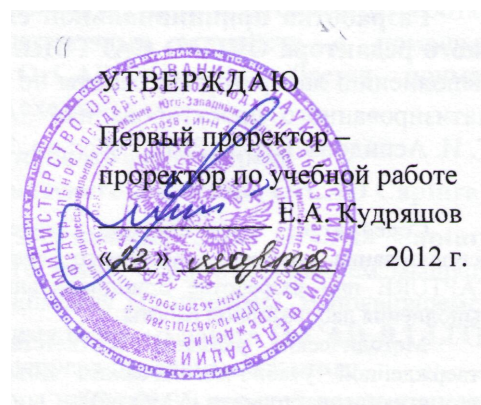


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 28.01.2021 17:36:57
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра конструирования и технологии электронно-
вычислительных средств



РАБОТА В ИНТЕГРИРОВАННОЙ СРЕДЕ ПРОЕКТИРОВАНИЯ ЭЛЕМЕНТОВ ПРОГРАММИРУЕМОЙ ЛОГИКИ

Методические указания по выполнению лабораторной работы
по дисциплине
«Информационные технологии проектирования электронно-
вычислительных средств»
для студентов специальности 210202.65

Курск 2012

УДК 681.325

Составитель: Т. И. Аспидова

Рецензент

Кандидат технических наук, доцент *О. Г. Бондарь*

Работа в интегрированной среде проектирования элементов программируемой логики : методические указания по выполнению лабораторной работы по дисциплине «Информационные технологии проектирования ЭВС/ Юго-Зап. гос. ун-т; сост.: Т. И. Аспидова. Курск, 2012. 27 с.: ил. 9.

Содержатся теоретические сведения, касающиеся реализации типового маршрута проектирования элементов программируемой логики в интегрированной среде. Указывается порядок выполнения лабораторной работы.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям автоматики и электроники (УМО АЭ).

Предназначены для студентов специальности 210202.65.

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 1/16.
Усл. печ. л. 1,57. Уч.-изд. л. 1,42. Тираж 30 экз. Заказ . Бесплатно.

Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94

ОГЛАВЛЕНИЕ

1. ЦЕЛЬ РАБОТЫ.....	4
2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	4
3. ПОДГОТОВКА К РАБОТЕ.....	19
4. РАБОЧЕЕ ЗАДАНИЕ	19
5. КОНТРОЛЬНЫЕ ВОПРОСЫ	26

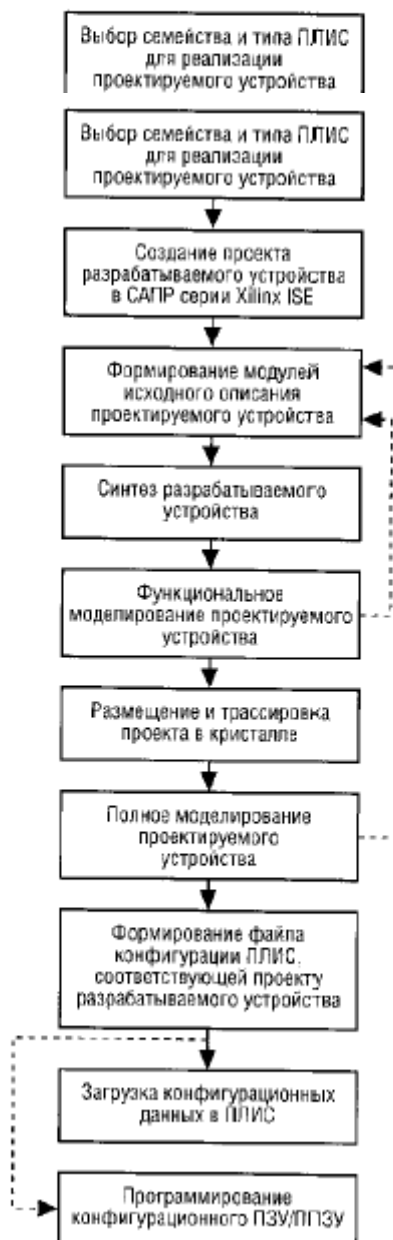
1. ЦЕЛЬ РАБОТЫ

Целью работы является изучение процесса разработки элементов программируемой логики в среде САПР Foundation Series.

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В процессе проектирования цифровых устройств на базе ПЛИС фирмы Xilinx можно выделить следующие этапы:

- выбор семейства и типа кристалла для реализации разрабатываемого устройства;
- создание нового проекта в САПР серии Foundation Series;
- подготовка исходного описания проектируемого устройства в схемотехнической, алгоритмической или текстовой форме;
- синтез проектируемого устройства;
- верификация исходных описаний проектируемого устройства методом функционального моделирования;
- размещение и трассировка проекта разрабатываемого устройства в кристалле;
- полное (временное) моделирование проектируемого устройства, выполняемое с учётом задержек распространения сигналов внутри кристалла;
- формирование конфигурационной последовательности ПЛИС, соответствующей проекту разрабатываемого устройства;



- программирование ПЛИС (загрузка проекта разработанного устройства в кристалл) или конфигурационного ПЗУ/ППЗУ.

Типовой маршрут проектирования цифровых устройств, выполняемых на основе ПЛИС фирмы Xilinx, показан на рисунке.

Структура проекта в САПР Foundation Series

Проектом в САПР Foundation Series называется совокупность модулей (файлов), которые содержат информацию, необходимую для выполнения всех этапов процесса разработки цифрового устройства на базе ПЛИС Xilinx. В его структуре можно выделить следующие группы модулей:

- исходные описания проектируемого устройства в графической или текстовой форме;
- документация, сопровождающая проект;
- промежуточные результаты, используемые в качестве исходных данных для последующих шагов проектирования;
- отчеты о выполнении основных этапов проектирования;
- описания тестовых воздействий, необходимых для моделирования устройства;
- окончательные результаты проектирования, используемые для конфигурирования ПЛИС.

Все модули проекта располагаются в одной папке, название которой совпадает с названием проекта. Изначально проект представлен только заголовком и модулем, в котором указываются параметры проекта. В дальнейшем к нему добавляют модули описания проектируемого устройства, а после выполнения каждого этапа процесса разработки устройства — результаты, полученные на этом этапе, и соответствующий отчет. Кроме того, разработчик может включить в проект необходимую текстовую документацию.

Работа с менеджером проекта пакета Xilinx Foundation Series

Менеджер проекта предназначен для организации эффективного управления разработкой цифрового устройства на

базе ПЛИС Xilinx в среде пакета. Работа с пакетом Xilinx Foundation Series начинается с запуска управляющей оболочки — Навигатора проекта, используя пиктограмму на рабочем столе или строку Foundation Series в меню Программы из панели, открываемой кнопкой Пуск.

После этого на экране монитора отображается основное окно Менеджера проекта, подробная структура которого показана на рис. 1.

Запуск Xilinx Foundation Project Manager

Интерфейс пользователя

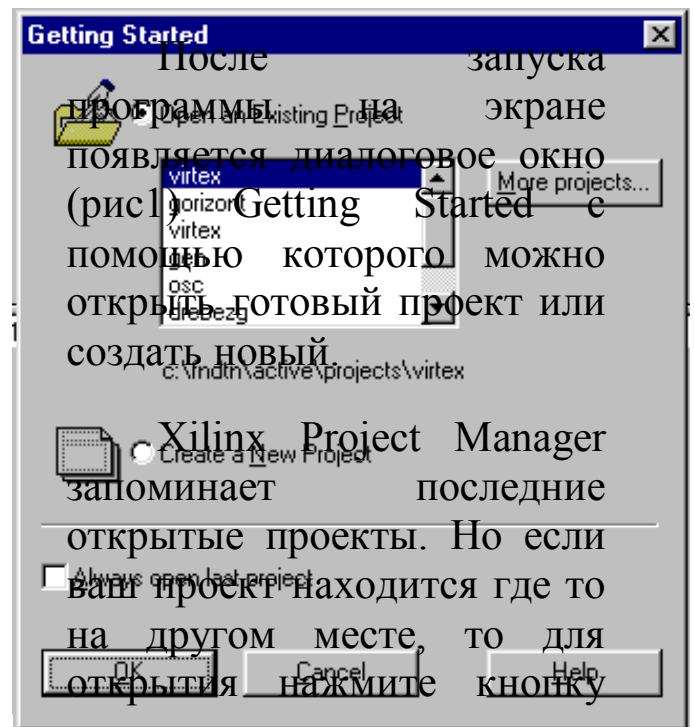


Рисунок 1. Диалоговое окно Getting Started

которой он собственно и расположен и текстового инициализационного файла проекта с расширением **PDF** (таким же как и документ Adobe Acrobat, но сходство лишь в расширении!), имеющего такое же имя как и папка. При открытии проекта система ищет именно этот файл. По умолчанию проекты располагаются в папке `\fndtn\active\project\`.

More projects... Физически проект состоит из двух частей:

папки с именем проекта, в

которой он собственно и расположен и текстового

инициализационного файла проекта с расширением **PDF** (таким же

как и документ Adobe Acrobat, но сходство лишь в расширении!),

имеющего такое же имя как и папка. При открытии проекта

система ищет именно этот файл. По умолчанию проекты

располагаются в папке `\fndtn\active\project\`.

Для создания нового проекта выберите **Create a New Project**. В появившемся диалоговом окне укажите имя проекта **Name** и директорию (выбирается кнопкой **Browse...**) где будет располагаться проект **Directory Type**. В разделе **Flow** укажите вариант ввода электрической схемы **Schematic** или на языке высокого уровня HDL (**Hardware Design Language**). В нижней части диалогового окна выбирается семейство (XC4000E, XC4000L, XC4000EX, XC4000XL, XC4000XLA, XC4000XV, XC5200, XC9500, XC9500XL Spartan, SpartanXL, Spartan II, Virtex), тип кристалла и корпуса внутри семейства и быстродействие (как правило, это число означает задержку сигнала при запрограммированном соединении между двумя любыми выводами в наносекундах). После этого мы попадаем в главное окно Xilinx

В главном окне в виде блок-схемы представлена вся последовательность действий. Начинается все с раздела **Design Entry**, в котором производят описание схемы. Правильность функционирования схемы проводят в разделе **Simulation**. После того как схема сформирована, в разделе **Implementation** с учетом параметров указанных в **.UCF** файле раздела **Files** производится компиляция проекта. По окончании которой, согласно ранее введенной схеме, образуется файл, описывающий внутренние межсоединения внутри кристалла ПЛИС а так же файлы отчетов по проведению операций оптимизации, размещения, трассировки, оценки временных параметров схемы и формирования выходного файла. Теперь, зная все межсоединения, известны временные задержки на каждом сигнале и вентиле. Используя эти данные, в разделе **Verification** производится моделирование функционирования вашей схемы со всеми реальными задержками. В случае если схема работает неправильно, необходимо будет вернуться в раздел **Design Entry** и изменить схему. После того, как схема отлажена с помощью раздела **Programming**, если велась работа с ПЛИС структуры FPGA, программой **Hardware Debugger** можно загрузить проект в ПЛИС или программой **PROM File Formatter** сформировать двоичный файл прошивки ПЗУ. Если же велась работа с ПЛИС структуры CPLD (серии XC9000 и XC9000XL) то с помощью программы **JTAG Programmer** программируется ПЛИС.

Основные команды Xilinx Project Manager:



Создать новый проект



Открыть ранее созданный проект



Добавление/удаление/копирование библиотек в проект или с помощью менеджера библиотек (Tools – Utilites – Schematic Symbol Library Manager)



Информация об выделенном объекте проекта



Поиск объекта в проекте



Создать новую версию компиляции проекта



Скопировать версию компиляции проекта



Просмотреть отчеты по оптимизации, размещению, трассировки, оценки временных параметров схемы и формирования выходного файла



Редактировать комментарий к проекту



Просмотреть командную строку проекта после компиляции



Помощь



Ввод схемы с помощью языка высокого уровня (VHDL, Abel, Verilog)



Ввод схемы с помощью диаграмм состояний (FSM)



Ввод схемы электрической принципиальной (Xilinx Foundation Schematic Editor). В этом случае отдельные элементы схемы можно реализовывать как на языке высокого уровня, так и с помощью диаграмм состояний.



Запуск проекта на компиляцию



Моделирование функционирования вашей схемы со всеми реальными задержками



Программирование ПЛИС



















Ввод описания проекта в виде схемы












Xilinx Foundation Schematic Editor



После запуска программы на экране появляется основное окно редактора схем **Xilinx Foundation Schematic Editor**, в котором производят ввод схемы электрической принципиальной. Ввод схемы осуществляется в многолистовом режиме, т.е. достаточно на одном листе указать имя цепи, чтобы впоследствии к ней на другом листе установить соединение. По умолчанию в нем подключается собственная библиотека компонентов, соответствующая типу выбранной ПЛИС. При изменении типа ПЛИС библиотека автоматически изменяется. Помимо элементарных вентилях, соответствующих выбранной ПЛИС, в библиотеке имеются макросы счетчиков, дешифраторов и прочих функциональных элементов, в том числе и некоторые аналоги микросхем серии X74_***.

Основные команды Xilinx Foundation Schematic Editor:

	Создать новый лист схемы (Ctrl – N)		Отсоединить выводы символа
	Открыть ранее созданную схему (Ctrl – O)		Вызов окна управления эмулятора (F8)
	Сохранить схему (Ctrl – S)		Вызов эмулятора
	Перейти в Xilinx Project Manager		Получить информацию о цепи или символе (F7)
	Распечатать схему (Ctrl – P)		Выбрать объект (F2)
	Вырезать в буфер обмена (Ctrl – X)		Переход по иерархии вверх/вниз
	Вставить в буфер обмена (Ctrl – Ins)		Выбрать элемент из библиотеки (F3)
	Вставить из буфера обмена (Shift – Ins)		Соединить элементы (F4)
	Уменьшить/Увеличить масштаб		Ввод шины (F5)

	Увеличить область (Выделить обл. лев. кн.		Автоматическое подсоединение к шине (F6)
	Страница во весь экран (PgDown)		Добавить цепь или имя шины
	Отменить предыдущее действие		Добавить вывод иерархического блока
	Вернуть отмененное действие		Присоединить цепь к цепи земли или питания
	Вызвать свойства объекта		Ввести рисунок или текст
	Присоединить вывод символа		

Для выбора элемента из библиотеки жмем клавишу F3 и выбираем необходимый элемент из окна или вписываем его название в строке ввода

При работе с элементами часто на экране возникает «мусор» для его устранения будут полезными следующие клавиши:

F9 – поместить точку схемы в центр экрана;

F10 – обновить (перерисовать) экран;

Для соединения элементов используется инструмент соединения элементов (F4) при его выборе курсор превращается в карандаш с проводником (см. выше). Причем можно указать только начало и конец соединения и система **AUTO ROUTE** автоматически проложит проводник на листе схемы, а потом, перейдя в режим выбора (F2), можно изменить прохождение проводника на схеме. Для соединения двух элементов не обязательно заходить в этот режим. Достаточно подвести элемент к элементу так, чтобы элементы касались выводами впритык, а потом отвести в сторону. Элементы можно также соединять с помощью указания имени цепи. Для этого отводим проводник от вывода элемента, жмем правую клавишу мыши в выпавшем окне выбора действий с цепью (рис. 5) выбираем **Add Label**. При выборе **Add Terminal** на схеме появится внешний порт, при выборе **Add PWR** – цепь можно присоединить к логическому «нулю» (GND) или к логической «единице» (VCC). Важным

моментом является последовательность соединения элементов, определяющие выводы ПЛИС. Если посмотреть архитектуру блока ввода/вывода, то вопросов возникнуть не должно, однако, часто возникают ошибки, когда неправильно соединяются элементы. Последовательность соединения для конфигурирования вывода как вход следующая: **IPAD – IBUF** или **IPAD – IBUFG** или **IPAD – BUFG** (в, том случае, если вход используется как вход тактовой частоты, комбинация зависит от серии ПЛИС) для конфигурирования вывода как выход: **OBUF – OPAD**, как выход с третьим состоянием **OBUFT – OPAD** или **OBUEF – OPAD** (рис. 2).

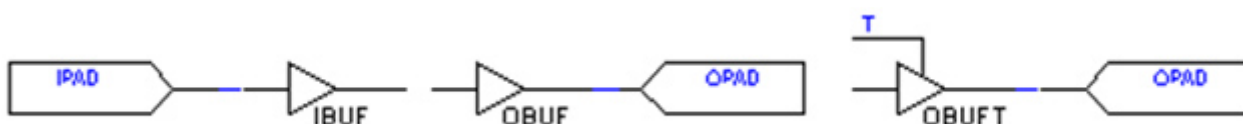


Рис. 2 Конфигурирование вывода как входного и как выходного

Для того чтобы привязать входной или выходной элемент (IPAD, OPAD, IOPAD) к конкретному выводу ПЛИС вызываются параметры элемента «Выбрать объект (F2)» - выбираем элемент - «вызвать свойства объекта» или дважды щелкнуть левой кнопкой мыши по нему. В разделе **Parameters** в графе **Name** указываем имя параметра **LOC** а в графе **Description** P<номер вывода> (**p10**). Далее жмем **Add**, и параметр появляется в окне параметров элемента. Теперь можно нажать **OK** или для перемещения параметра в удобное место сразу нажать **Move**. Этот параметр будет располагаться рядом с символом (рис. 10.) Для удаления параметра снова заходим в это окно, выбираем необходимый параметр и жмем **Delete**.

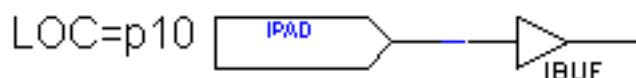


Рис. 3 Элемент, привязанный к выводу №10

Для ввода шины используется компонент ввода шины (F5) при его выборе курсор превращается в карандаш с шиной (см.

выше). Для ввода шины указываю левой кнопкой мыши начало шины, а в конце дважды щелкают левой кнопки, после чего выпадает меню ввода имени шины. Шина может быть одиночной, а может быть составной. В случае одиночной шины выбирается пункт Simple Bus, в графе Name указывается ее имя и указывается ширина шины в графе Range (например, Bus[7:0] это 8-ми разрядная шина). В случае комплексной шины через запятую указывается состав шины (например, A[0:3],B,D[1:4] это 9-ти разрядная шина состоящая из следующих сигналов: A0,A1,A2,A3,B,D1,D2,D3,D4). После чего, если эта шина является портом, в графе Terminal Maker выбирается тип порта: Input, Output, BiDirectional. Если же это просто внутренняя шина, то указывается None.

Если сейчас подводить проводники к шине (F4 или F6), то цепь будет подходить к шине в виде «вилки». Для того, что бы вид соединителя к шине (**Bus Tap**) был другим, перед подсоединением к ней элементов выбираем шину (F2) и жмем иконку

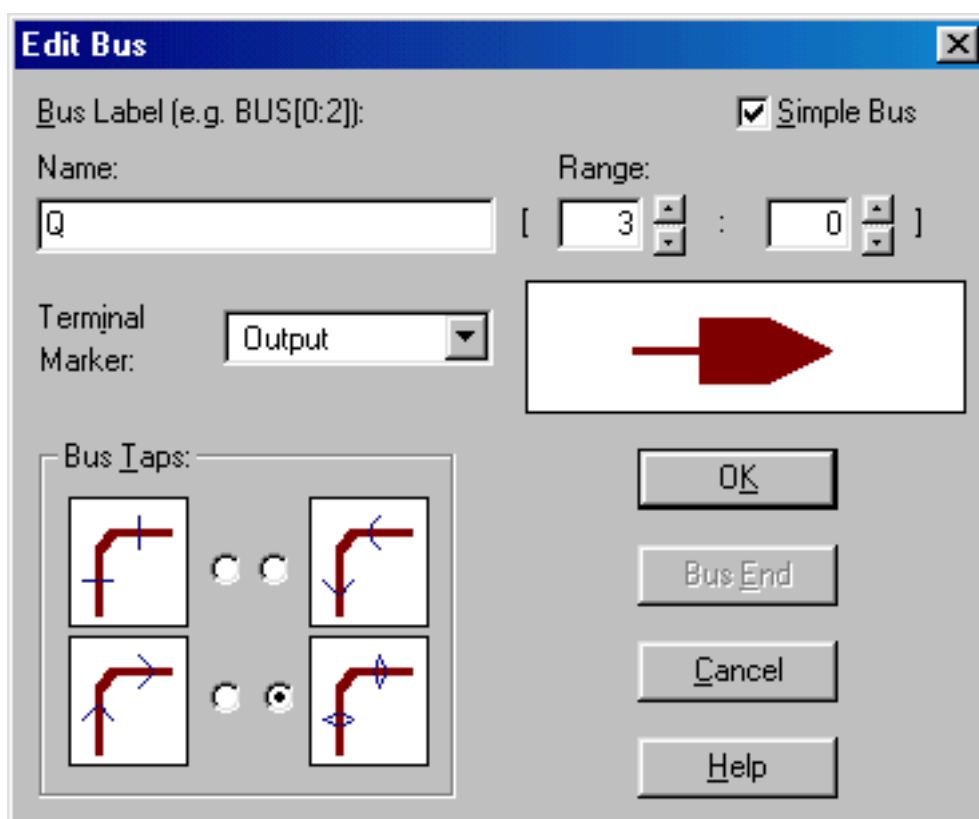


Рис. 4 Меню Edit Bus

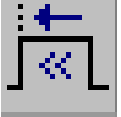
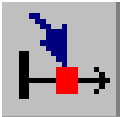
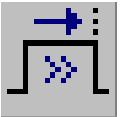
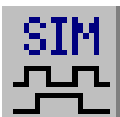

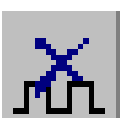

изменения свойства объекта или ждем на шину правую клавишу мыши и выбираем пункт **Edit Bus**. И в появившемся окне (рис. 4) в разделе **Bus Taps** выбираем любой из четырех видов. Функционального значения это не имеет. Это введено для удобного чтения схемы.

Для автоматического соединения выводов элементов к шине используется специальный инструмент (F6). Порядок работы с ним следующий:

1. расставляются элементы, которые должны подсоединяться к шине;
2. рисуется шина;
3. выбирается инструмент (F6);
4. мышкой выбирается шина, к которой будут подсоединяться элементы (при этом у указателя курсора появляется прямоугольник с указанием имени цепи);
5. щелкаем указателем по выводам элементов, которые должны будут присоединены к шине. Причем система автоматически (начиная с младшего индекса) будет добавлять проводники к шине

Для получения сведений о цепи или элементе применяется специальный инструмент получения информации о цепи или символе (F7). При его вызове курсор преобразуется в курсор с вопросительным знаком и если указать им на какую-нибудь цепь, она автоматически подсветится, а в окне высветится информация о выводах (с указанием элементов), подсоединенных к этой цепи. И если указать мышью по какому-нибудь выводу он автоматически расположится в центре экрана, а на самом выводе (для удобства восприятия) появится жирная точка. Для того чтобы посмотреть цепь достаточно только указать ее имя в поле для ввода и нажать ввод.

Расстановка/удаление точек для временного моделирования на схеме производится с помощью специального инструмента (F8). При его вызове появляется дополнительное меню:

	Установка пробы на вход элемента		Вернуться назад на пол такта
	Установка пробы на цепь, сигнал которой формируется с выхода		Сделать шаг вперед на пол такта
	Запустить временной эмулятор		Сделать шаг вперед на время, установленное во временном симуляторе
	Удалить все пробы		Сохранить пробы на диске

Вначале расставляются все точки, в которых нам будет нужно посмотреть состояние, после чего, нажимается кнопка запуска временного симулятора.

Для формирования своего библиотечного элемента выполняется путем вызова опции **Create Macro Symbol From Current Sheet** из меню **Hierarchy**. В выпавшем меню в графе **Symbol Name** указываем название макроэлемента, в графе **Reference** указывается буква обозначающая позицию на схеме (в дальнейшем при выборе элемента из библиотеки на схеме он будет нумероваться как D1, при выборе следующего - D2 и т.д.). В графе **Comment** можно кратко описать его функциональное значение. После нажатия кнопки **ОК** и генерирования библиотечного элемента система спросит, желаете ли вы редактировать этот элемент. Если вы ответите **Yes**, то схема останется в редакторе, в противном случае она закроется. Для ввода/вывода сигналов в/из элемента, схема должна содержать специальные выводы иерархического блока (см. рис. 5, опция **Add Terminal**). После выбора элемента, при желании, можно изменить порядок расположения выводов элемента. Для этого вызываются свойства элемента и выбирается **Symbol Editor**, в котором простым перетаскиванием можно изменить расположение выводов в элементе, размер элемента, нарисовать несложные графические значки и т.д.

Моделирование

Xilinx Logic Simulator

Моделирование производится с помощью встроенного **Logic Simulator**, который вызывается соответствующей кнопкой из меню специального инструмента для временного моделирования (F8). При этом система автоматически генерирует список связей вашей схемы и загружает его в **Logic Simulator**, поэтому точки, в которых нам нужно посмотреть поведение схемы можно расставлять как до вызова **Logic Simulator**, так и после его загрузки.

Для установки какого либо воздействия на сигнал или его состояние сначала выбирается мышью сам сигнал (или шина), а затем нажимается пиктограмма необходимого воздействия. Воздействия на сигнал устанавливаются, используя окно **Stimulator Selector**. Внешний вид **Stimulator Selector** представлен на рис. 5.

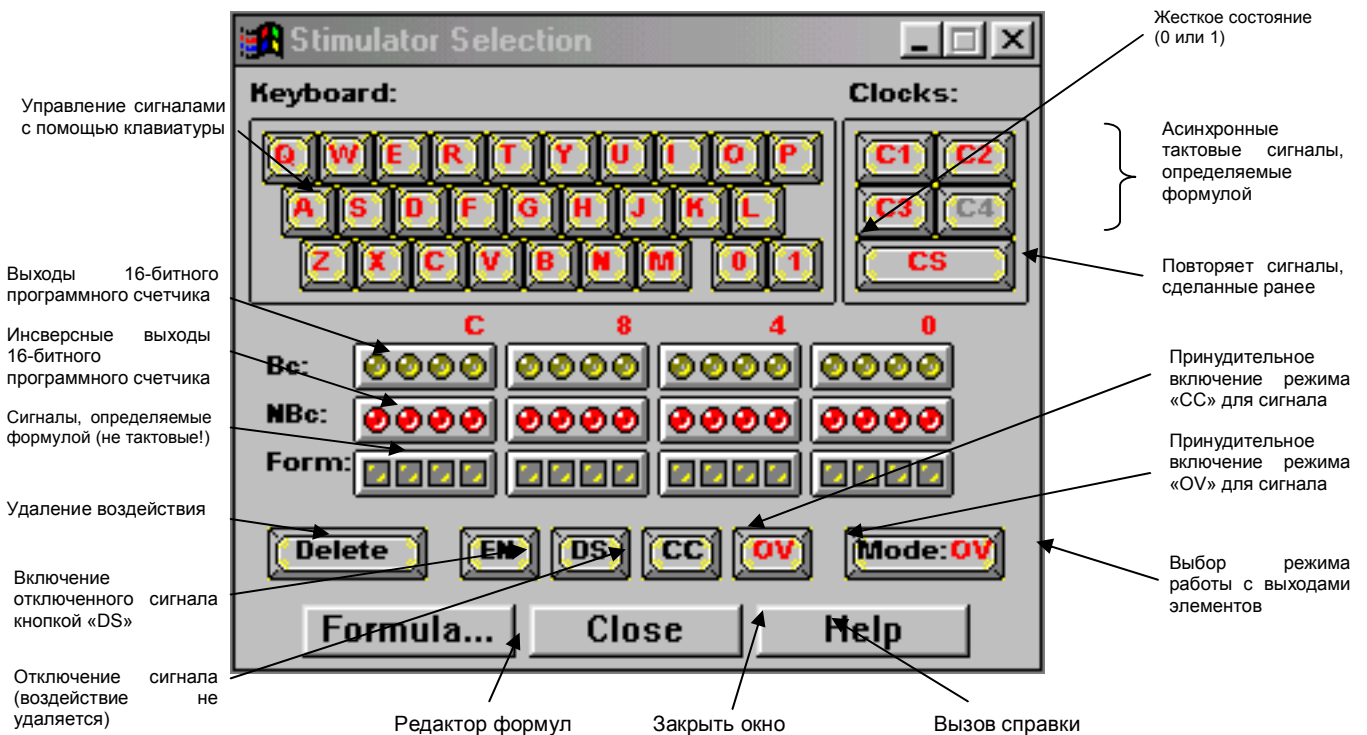


Рис. 5. Внешний вид *Stimulator Selector*

Для установки воздействия на сигнал можно сначала его выбрать, щелкнув на нем левой кнопкой мыши, (строка выделится) а потом указать на метод воздействия в окне **Stimulator Selector**, а можно подвести указатель мыши на необходимое воздействие, нажать левую кнопку мыши (при этом курсор преобразится в перекрестие) и не отпуская кнопку мыши, перетащить его на необходимый сигнал. Для управления сигналами можно использовать клавиатуру, при этом состояние сигнала (из 0 в 1) переключается нажатием клавиши на вашей клавиатуре *только в нижнем латинском режиме!!!*. Для задания тактовых сигналов можно воспользоваться 16-битным программным счетчиком. Его тактовая частота задается в параметрах в разделе **Clock**. Эту частоту можно определить как периодом, так и частотой. Или же тактовые сигналы можно определить формулой. Для этого нажимается кнопка **Formula...** и, выбрав необходимый стимулятор (нужно дважды щелкнуть левой кнопкой мыши) в строке ввода **Edit formula** (рис. ввести формулу).

В редакторе формул используются следующие символы:

Символ:	Значение:
H, L	Высокий и низкий уровень (1 и 0)
X	Неизвестное состояние (высокое или низкое)
Z	Состояние высокого импеданса (3-е состояние)
0..9	Цифры, используемые для определения величин
()	Круглые скобки, используются для выделения подвыражений
ps, ns, us, ms	пс, нс, мкс, мс (по умолчанию – нс)
[]	Квадратные скобки, используются для определения шестнадцатеричного значения шины

Например:

H40L10	1=40нс; 0=10нс
(H40L10)20	20 раз повторяется 1=40нс; 0=10нс
H4usL1us	1=4мкс; 0=1мкс
((H10L10)20x30)10	1=10нс; 0=10нс повторяется 20 раз, потом 30нс длится неизвестное состояние. И все это повторяется 10 раз
([2]40[A0]55)10	10 раз повторяется Шина=02h 40нс; Шина=A0 55нс

Далее определяем режимы моделирования. Всего бывает 4 режима моделирования:

- Functional (математическое моделирование, т.е. все задержки=0)
- Timing (моделирование с учетом реальных задержек)
- Glitch (режим сбоев, машина случайным образом затягивает длительность тактов)
- Unit (режим моделирования с единичными задержками)

После определения режима моделирования определяемся с *режимом работы с выходами элементов.* Существуют 2 режима:

- CC (Chip Controlled Mode) выходы внутренних элементов переписывают входные стимуляторы
- OV (Override Mode) позволяет стимуляторам переписывать присвоенные выходным выводам сигналы.

После того, как определены все воздействия на сигналы жмем кнопку инициализации, при этом эмулятор приходит в начальное состояние (время моделирования=0) и жмем кнопку «сделать шаг», при этом на экране симулятора появятся временные диаграммы входных и выходных состояний схемы. За поведением схемы удобно наблюдать из Xilinx Schematic Editor, для этого переходим в Xilinx Schematic Editor и, нажимая кнопки управления, наблюдаем за схемой. Все единицы на схеме подсвечиваются красным цветом, а нули зеленым, значение шины пишется в числовом виде (рис. 15). Для изменения состояния сигналов, можно нажать клавиши на клавиатуре, а можно щелкнуть левой клавишей мыши на имя назначенной кнопки, либо на текущее значение сигнала и снова жмем шаг. Если необходимо сделать длительное моделирование, то вызываем опцию **Start Long Simulation...** из меню **Options**.

Для измерения временных параметров жмем на поле с временными диаграммами правую клавишу мыши и в выпавшем меню, в графе **Measurements**, включаем опцию **Measurements On**. Для удобства рассмотрения пользуемся кнопками масштабирования или же левой клавишей мыши выделяем необходимый участок на линейке времени, и как только отпустим кнопку мыши - выделенный участок увеличится на все окно. Для

удаления измерений в графе **Measurements**, выбираем опцию **Delete All**, или же выключаем переключателем в основном окне.

Для добавления комментария к сигналу выбираем его и ждем на поле с временными диаграммами правую клавишу мыши и в выпавшем меню, в графе **Comments** выбираем пункт **Add**.

Для удобства (если приходится часто назначать большое количество сигналов) модель желательно сохранять, при этом сохраняется не только назначенные воздействия на сигналы, но и временные диаграммы, присутствующие на момент сохранения.

Компиляция проекта

Компиляция проекта осуществляется в Xilinx Project Manager, путем нажатия кнопки **Implementation**. При ее нажатии система автоматически предложит обновить список связей проекта, после чего выдаст меню настройки компиляции В нем можно установить тип кристалла и параметры компиляции. Для того чтобы не сохранять варианты компиляции ставится «птичка» в графе **Overwrite current version**. Нажав кнопку **Options**, попадаем в настройку компиляции проекта. Для того чтобы можно было произвести временное моделирование с учетом реальных временных задержек необходимо включить опцию **Produce Timing Simulation Data**.

Значения параметров в строках не изменяйте. Все настройки происходят в разделе **Implementation**, для этого напротив этой надписи нажимается кнопка **Edit Template**. В разделе **Optimize and Map** выбирается стратегия оптимизации (генерирование 5-ти входных функций, упаковка по минимуму площади или быстродействию, использование защелок в блоках ввода/вывода, использование глобальных тактовых буферов). В разделе **Place & Route** в графе **Place & Route Effort Level** устанавливается усилие трассировки. Если перевести движок в крайнее левое положение, то машина предпримет меньшее количество итераций компиляции, что дает высокую скорость, но худшие результаты, если движок перевести в крайнее правое положение, процесс компиляции будет происходить медленно, но результаты будут значительно лучше. На процесс компиляции можно повлиять, если внести соответствующие указания в файле констант пользователя **User**

Constraint File (*.ucf). В нем автоматически уже выделенная в комментариях изложена краткая помощь по его использованию

Xilinx Flow Engine

После окончания установки всех настроек жмем кнопку Run и система запускает компилятор **Flow Engine**, которой в автоматическом режиме выполняет всю компиляцию

Успешное завершения компиляции оповещается меню. Теперь откомпилированный проект из Xilinx Project Manager, используя Verification, загружаем во временной эмулятор, где с учетом задержек можно промоделировать поведение схемы. В этом режиме точки задаются путем выбора из списка цепей и модуля проекта *«Выбрать компонент или цепь для моделирования»*, поскольку многие цепи сокращаются в процессе компиляции в результате оптимизации, рекомендуется называть цепи от символа вывода и буфером (например между IPAD и IBUF). Поскольку эта цепь никогда не выбросится, то имя цепи обязательно будет в списке, что значительно облегчит поиск необходимых сигналов.

3. ПОДГОТОВКА К РАБОТЕ

Изучить методические указания к данной работе.

Установить пакет в собственном персональном компьютере или освоить работу с пакетом в лаборатории кафедры.

4. РАБОЧЕЕ ЗАДАНИЕ

В качестве рабочего задания разработаем декадный счетчик импульсов с выводом на семисегментный индикатор.

1. Создадим новый проект NEWPROJ на кристалле Spartan XC4000-PQ208.

2. Создаем необходимые библиотечные элементы (двоично-десятичный счетчик и дешифратор). Начнем с создания счетчика. Счетчик можно реализовать самому либо взять готовый библиотечный компонент. Рассмотрим оба способа. Для построения десятичного счетчика необходимо реализовать четырехразрядный

двоичный счетчик с ограничением счета. Счетчик будем выполнять с параллельным переносом. Для этого берем 4 триггера типа FDC. Жмем и в нижней строке появившегося окна **SC Symbols** вводим FDC. Редактор автоматически перенесет курсор к выбранному элементу. Это является наиболее быстрым способом поиска элементов, хотя его можно было и выбрать перелистывая весь список элементов библиотеки. Для добавления в схему элемента, который уже есть на листе достаточно войти в режим добавления элементов (F3) и щелкнуть мышкой по уже имеющемуся элементу на схеме. Выставляем все необходимые элементы, а затем выполняем межсоединения (рис.1).

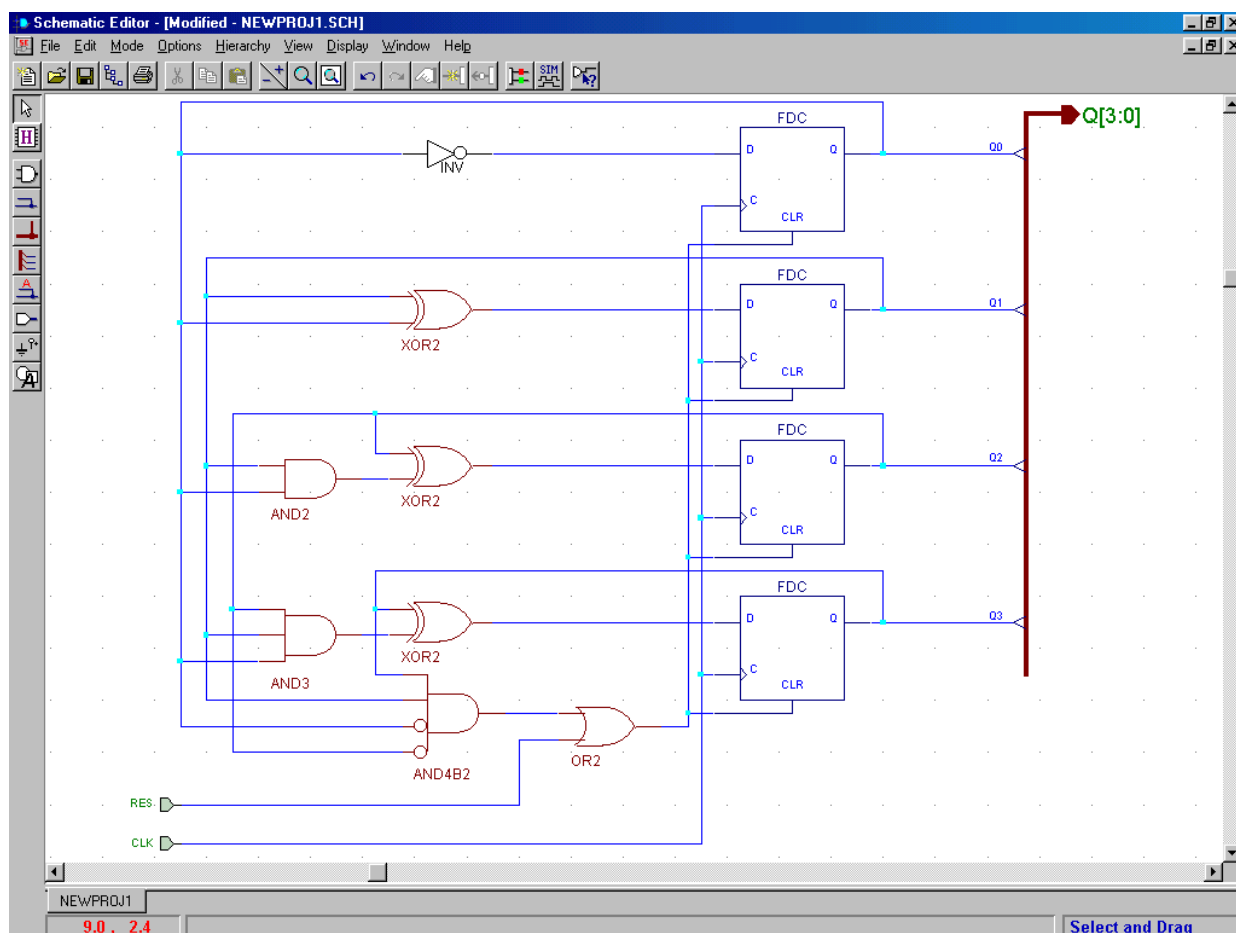


Рис. 6. Двоично-десятичный счетчик

Теперь осталось добавить входные и выходные сигналы. Сначала введем входной сигнал сброса счетчика в ноль. Для этого переходим в режим ввода межсоединений (F4), ведем цепь на

свободное место и жмем правую клавишу мыши, при этом выпадает меню и выбираем **Add Terminal**. После выбора этого пункта меню переходим в окно указания параметров вывода иерархического блока **Hierarchy Connector**. В строке **Name** вводим имя **RES** в строке **Terminal Type**, устанавливаем тип вывода **INPUT** и жмем кнопку **OK**

Вывод можно ввести так же используя иконку в главном окне Xilinx Foundation Schematic Editor. При этом сразу же выпадает меню в котором вводим имя CLK тип вывода INPUT и жмем кнопку OK. И выставляем вывод в любое свободное место на листе. Потом же просто в режиме ввода межсоединений (F4), соединяем с цепью соединяющей все входы С триггеров FDC. Выход счетчика можно сделать с помощью 4-х выводов, но удобнее его сделать в виде шины. Для этого переходим в режим ввода шины (F5) и после того, как ввели шину дважды щелкаем левой клавишей мыши или жмем правую клавишу мыши и выбираем пункт **Add Bus Terminal**. И попадаем в меню добавления вывода шины. в графе **Name** вводим имя шины Q, в графе **Range** (3 : 0), поскольку шина 4-х битная, а счет начинается с нуля), в **Terminal Maker** – тип шины **Output** (на выход).

После чего переходим в режим ввода межсоединений (F4), соединяем шину с выходами триггеров. Теперь необходимо указать, какой сигнал соответствует какому сигналу в шине. Для этого дважды щелкаем по цепи, подходящей к шине от верхнего триггера и в выпавшем меню **Net Name** вводим имя цепи Q0. Таким же называем и остальные цепи Q1, Q2, Q3.

Следует отметить, что имя цепи появится в том месте, где щелкали мышью по цепи. В дальнейшем имя цепи можно передвинуть мышью на любое место. В результате всех этих действий должна получиться следующая схема (рис. 6). Данную схему можно было нарисовать с помощью опции **Add Label** (рис. 2). В этом случае, вместо того, чтобы проводить цепь от элемента к элементу, достаточно просто обозначить имя цепи. И все элементы с подсоединенными одинаковыми именами цепей будут считаться соединенными. В данном случае было бы удобно таким образом соединять схему переноса и схему ограничения счета, т.е. не нужно было бы рисовать петли от выходов триггеров FDC - Q до входов

элементов AND и XOR. Но зато такое описание не наглядно и порой затрудняет поиск цепи. Для этой цели удобно применить специальный инструмент получения информации о цепи или символе (F7). При его вызове курсор преобразуется в курсор с вопросительным знаком и если указать им на какую-нибудь цепь, она автоматически подсветится, а в окне высветится информация о выводах (с указанием элементов), подсоединенных к этой цепи. И если указать мышью по какому-нибудь выводу он автоматически расположится в центре экрана, а на самом выводе (для удобства восприятия) появится жирная точка. Для того чтобы посмотреть цепь достаточно только указать ее имя и нажать ввод.

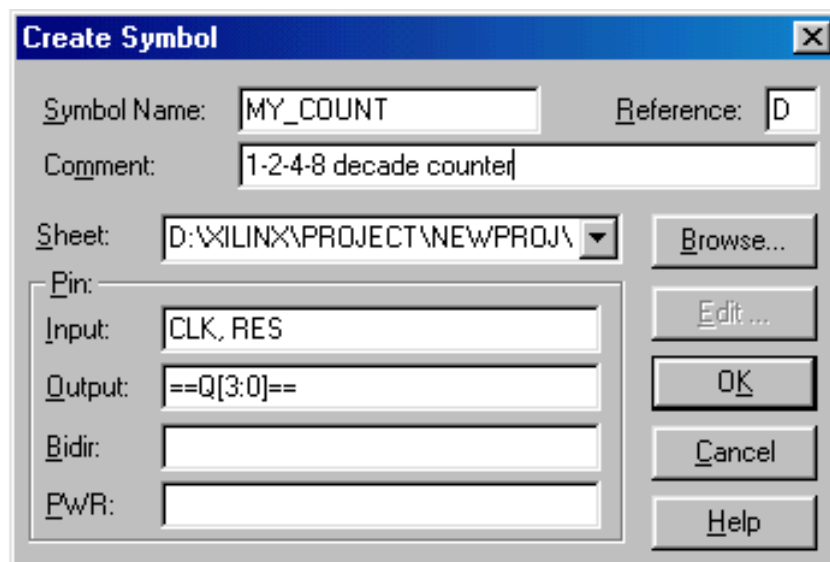


Рис. 7. Меню Create Symbol

Теперь из этой схемы создадим макроэлемент. Для этого выбираем в меню Hierarchy подменю Create Macro Symbol From Current Sheet. В выпавшем меню (рис. 7) в графе Symbol Name указываем название макроэлемента MY_COUNT, в графе Reference - D (здесь указывается буква в позиции на схеме, в дальнейшем при выборе элемента из библиотеки на схеме он будет нумероваться как D1, при выборе следующего - D2 и т.д.). В графе Comment можно кратко описать его функциональное значение. После нажатия кнопки ОК и генерирования библиотечного элемента система спросит, желаете ли вы редактировать этот элемент. Если вы ответите Yes, то схема останется в редакторе, в противном случае она закроется. В любом случае, этот лист уже не является

схемой проекта, и для дальнейшей работы создаем новый лист (Ctrl-N). Теперь заходим в библиотеку элементов (F3) и находим наш счетчик (рис. 8).

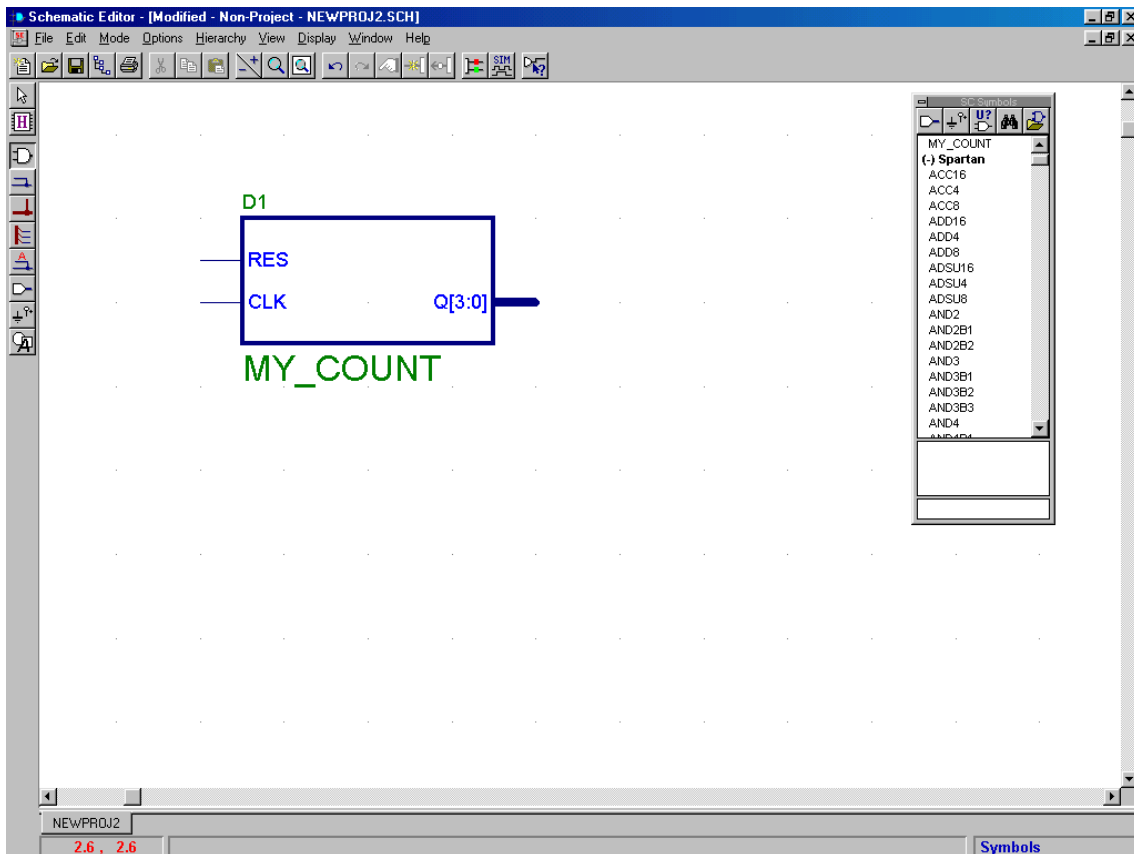


Рис. 8. Макроэлемент пользователя MY_COUNT.

Для создания дешифратора на языке высокого уровня вызываем меню из меню **Hierarchy** опцию **New Symbol Wizard...** в выпавшем окне жмем кнопку «Далее» в следующем окне указываем имя символа **LED_DECODE** в графе **Symbol Name**, в разделе **Contents** выбираем **HDL Code** и язык **VHDL** в разделе **HDL Language**. В следующем меню назначаем выходы (рис. 9). Для добавления вывода жмем кнопку **NEW** и в графе **Name** вводим его имя **HEX** в разделе **Bus** устанавливаем **[3:0]** и в разделе **Direction** выбираем **Input**. В результате получаем входную шину **HEX[3:0]**. Аналогично добавляем выходную шину для семисегментного индикатора **LED[6:0]**, только не забываем, что в

разделе **Direction** выбирается **Output**. Если же необходимо добавить вывод, а не шину то раздел **Bus** не трогается вообще, т.е. [0:0].

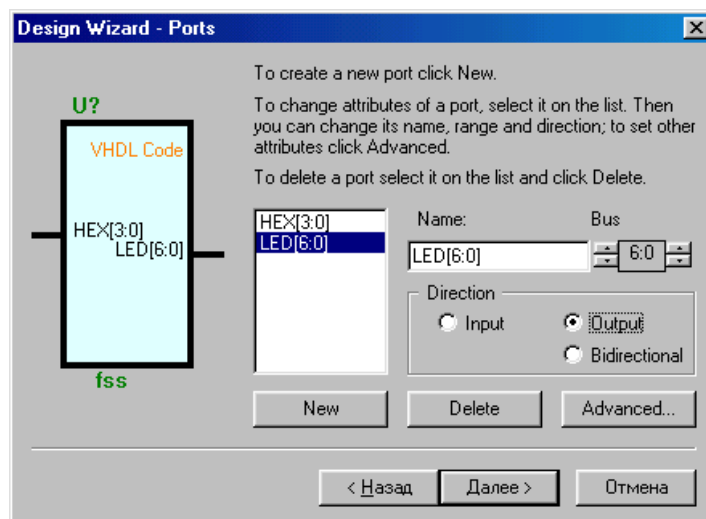


Рис. 9. Определение выводов символа

Добавляем если нужно комментарий в следующем окне и жмем «**Готово**» Выбираем инструмент перехода по иерархии, дважды щелкаем по нему левой кнопкой мыши загрузится **HDL Editor**, в котором вводим текст описания дешифратора. При создании описания можно воспользоваться приложением к данной лабораторной работе и/или использовать режим **TOOLS => LANGUAGE ASSISTANT HDL-редактора**

В шине младший 0-й бит соответствует сегменту «а» (рис. 10), а 6-й сегменту «g». В данном случае выполнен дешифратор для светодиодного семисегментного индикатора с общим анодом.

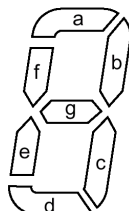


Рис. 10. Семисегментный индикатор

В случае применения индикатора с общим катодом необходимо проинвертировать данные. После ввода описания в меню **Synthesis** выбираем опцию **Synthesize**, по окончании работы которой будет предложено обновить макрос.

После подготовки всех элементов схемы они соедините их между собой и с элементами ввода/вывода. В результате Вы должны получить следующую схему (рис.11)

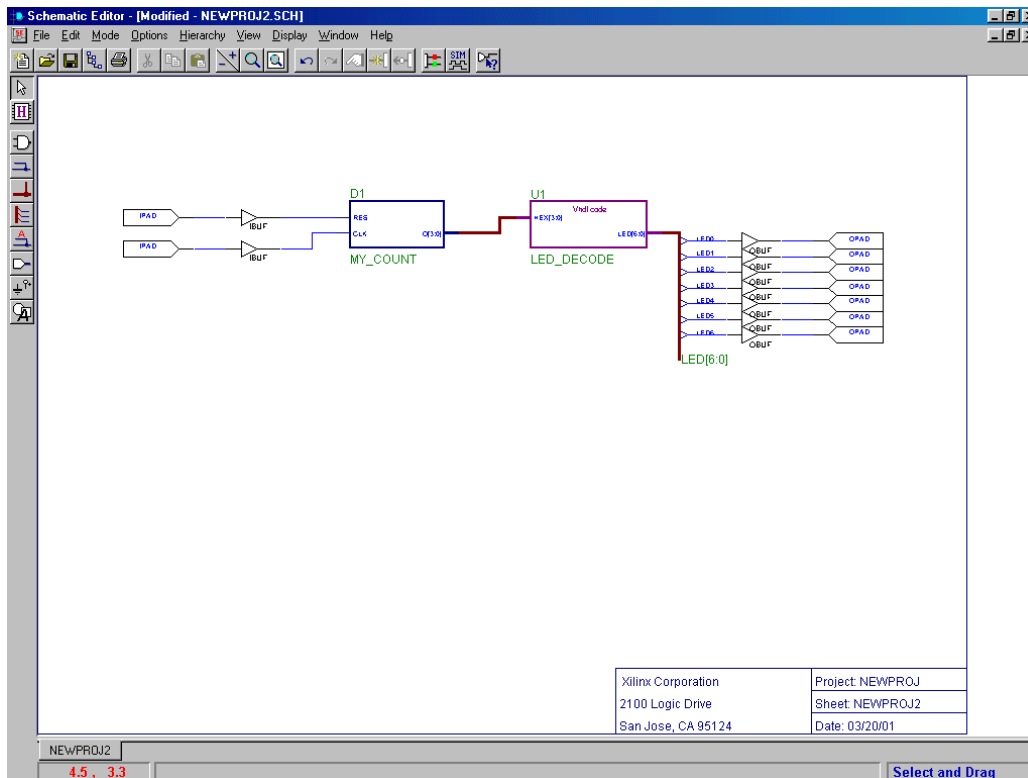


Рис. 11. Законченная схема

- Проведите функциональное моделирование устройства, убедитесь в правильности разработки схемы.
- Сохраните схему, закройте **Xilinx Schematic Editor**, перейдите в **Xilinx Project Manager** и откомпилируйте схему. При трассировке должны быть задействованы правые контакты микросхемы.
- Проведите моделирование устройства с учетом временных характеристик, убедитесь в правильности разработки схемы.
- Загрузите редактор топологии (рис. 11) и проанализируйте

- задействованные для реализации устройства ресурсы ПЛИС.
- Создайте VHDL-описание счетчика, включите его в схему вместо элемента MY_COUNT и воспроизведите все ранее указанные действия. Убедитесь, что оба варианта работают одинаково.
 - Определите, для реализации какого варианта схемы понадобилось больше ресурсов ПЛИС.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Этапы проектирования элементов программируемой логики.
2. Структура проекта в Foundation Series.
3. Какая информация вводится при создании нового проекта?
4. Объясните блок-схему, приведенную в главном окне Foundation Series.
5. Конфигурирование вывода как входного и выходного.
6. Конфигурирование вывода как двунаправленного.
7. Как осуществляется привязка вывода схемы к контакту ПЛИС?
8. Как создаются иерархические элементы?
9. Способы задания стимулирующих воздействий (Add Stimulators).
10. Режимы моделирования в XILINX Foundation Series.
11. Что такое User Constraints File и для чего он используется?
12. Что такое VHDL, для чего он используется и в чем преимущества перед схемным проектированием?
13. Этапы проектирования с использованием VHDL.
14. Что такое EDIF?
15. Что такое синтез VHDL-описания?

16. Что такое несинтезируемые конструкции VHDL? Приведите примеры.
17. Что такое FPGA?
18. Стили описания в VHDL.
19. Различные понятия «времени» в моделировании

