

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 03.05.2024 09:48:42

Уникальный программный ключ:

0b817ca9-0e65f68-4b137e5-4426439-f51e11eabbf77e947df4e4851fda76d089

МИНОБРНАУКИ РОССИИ

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

Кафедра вычислительной техники

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

«12» 08 2020г.



ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ В САПР

XILINX ISE 8.2I

Методические указания к лабораторным работам
для студентов направления подготовки
«Информатика и вычислительная техника»

Курск 2020

УДК 621.3

Составители: В.С. Панищев, М. И. Труфанов, И.Е. Чернецкая

Рецензент

Доктор технических наук, профессор *А.С. Сизов*

Проектирование цифровых устройств в САПР Xilinx ISE 8.2i:
методические указания к лабораторным работам для студентов/ Юго-Зап. гос.
ун-т; сост.: В.С. Панищев, М. И. Труфанов, И.Е. Чернецкая. – Курск, 2020.
– 40 с.: ил. 35. – Библиогр.: с. 40

Методические рекомендации содержат сведения по разработке устройств на базе программируемых логических интегральных схем (ПЛИС) в САПР Xilinx 8.2. Приведены сведения о разработке устройств на базе ПЛИС на схемотехническом уровне, показаны основные этапы разработки, подходы к решению практических задач.

Предназначены для студентов направления подготовки «Информатика и вычислительная техника» очной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать *12.08.20* Формат 60x84 1/16.
Усл. печ. л. 2,33. Уч. – изд.л. 2,11. Тираж 50 экз. Заказ *230*. Бесплатно.
Юго-Западный государственный университет.
305040, Курск, ул. 50 лет Октября, 94.

Содержание

1	Общие сведения	4
2	Схемотехнический редактор пакета Xilinx ISE	6
3	Основные этапы проектирования цифровых устройств в Xilinx ISE	8
4	Пример проектирования	9
5	Варианты заданий	38
6	Контрольные вопросы	39
7	Список литературы	40

1 Общие сведения

Одной из тенденций развития цифровой техники на современном этапе является широкое применение ресурсов программируемой логики не только для реализации отдельных блоков, но и проектирование устройств в целом, вплоть до создания систем на кристалле (System on Chip (SoC)). Программируемые логические интегральные схемы (ПЛИС) и соответствующие средства проектирования позволяют выполнить жесткие требования, предъявляемые ко времени разработки, и в жатые сроки создавать цифровые устройства и системы различного уровня сложности и степени интеграции.

Компания Xilinx (www.xilinx.com) является лидером на мировом рынке микросхем программируемой логики. Перепрограммируемые пользователем базовые матричные кристаллы (Field Programmable Gate Array, или FPGA), а также средства их проектирования и отладки, выпускаемые фирмой Xilinx, используются в устройствах с цифровой обработкой информации – например, в системах телекоммуникации и связи, вычислительной технике, периферийном и тестовом оборудовании, электробытовых приборах.

Основными системами автоматизированного проектирования цифровых устройств на базе ПЛИС фирмы Xilinx являются САПР Xilinx ISE, которые заменили средства разработки предыдущего поколения Foundation Series. Применение программных средств Xilinx ISE позволяет значительно сократить время разработки и повысить уровень эффективности результатов за счет применения усовершенствованных методов проектирования, алгоритмов синтеза, размещения и трассировки проекта в кристалле.

Программные средства серии Xilinx ISE представляют собой систему сквозного проектирования, которая реализует полный цикл разработки цифровых устройств и систем на основе ПЛИС, включающий этапы создания исходных описаний проекта, синтеза, моделирования, размещения и трассировки, а также программирования кристалла.

Отличительными особенностями систем автоматизированного проектирования серии Xilinx ISE являются:

- ✓ поддержка различных методов описания проектируемых устройств (графических, в форме принципиальных схем или диаграмм

состояний, и текстовых, с использованием языков описания аппаратуры HDL (Hardware Description Language);

- ✓ возможность использования проектов, подготовленных в других системах проектирования;

- ✓ наличие схемотехнического редактора, укомплектованного набором обширных библиотек;

- ✓ интеллектуальные средства создания HDL-описаний, формирующие шаблоны на основании информации, предоставляемой пользователем, для языков описания аппаратуры VHDL™, Verilog™ и ABEL™ HDL;

высокоэффективные встроенные средства синтеза HDL-проектов, поддерживающие языки VHDL, Verilog и ABEL HDL;

- ✓ интегрированный интерфейс для средств синтеза третьих фирм, обеспечивающий возможность применения, например, пакетов Synplicity Synplify™/Pro и LeonardoSpectrum™, поддерживающих языки VHDL и Verilog; развитые средства верификации проекта, позволяющие сократить полное время разработки устройства за счет обнаружения возможных ошибок на более ранних стадиях проектирования и сокращения длительности и количества возможных итераций;

- ✓ автоматические средства трассировки проекта в кристаллы различных семейств ПЛИС фирмы Xilinx с учетом оптимизации проекта по различным параметрам; единые средства программирования кристаллов всех семейств ПЛИС фирмы Xilinx, выполненных по различной технологии (CPLD и FPGA), и конфигурационных ППЗУ, поддерживающие несколько типов загрузочных кабелей JTAG-интерфейса;

- ✓ встроенный комплект вспомогательных программных средств, позволяющих повысить эффективность процесса проектирования, включающий анализатор статических временных характеристик Timing Analyzer™, интерактивный графический редактор размещения Floorplanner™, модуль оценки потребляемой мощности XPower™, мастер подготовки описаний блоков синхронизации, выполняемых на основе DCM, Architecture Wizard™, интерактивный графический редактор топологических ограничений PACE™ (Pinout and Area Constraints Editor);

- ✓ доступный для разработчика пользовательский интерфейс и наличие в каждом модуле пакета справочной системы, сокращающие время освоения САПР; наличие интегрированного с пакетом САПР набора инструментов и утилит других фирм, предоставляющих дополнительные удобства в процессе проектирования, включающего утилиту генерации

тестовых сигналов HDL Bencher™, программу моделирования ModelSim XE III Starter™ и редактор диаграмм состояний StateCAD™.

Данные методические указания ориентированы на использование пакета Xilinx ISE версии 8.2i, однако могут быть применены так же для более ранних версий, включительно до версии 5.1.

2 Схемотехнический редактор пакета Xilinx ISE

Схемотехнический редактор Engineering Schematic Capture (ESC) предназначен для ввода принципиальных электрических схем в графическом режиме. Несмотря на то, что программные средства пакета Xilinx ISE ориентированы на применения языков HDL (Hardware Description Language), проектируемое устройство может быть описано полностью с помощью средств схемотехнического редактора, однако целесообразнее использовать смешанный подход описания проектируемого устройства. Данный подход заключается в том, что функциональные блоки проектируемого устройства описываются с помощью одного из языков HDL (ABEL, VHDL или Verilog), а для верхнего уровня иерархии проекта используется схемотехническая форма описания.

Хотя схемотехническая форма представления является более привычной для разработчиков, рекомендуется изучать в дальнейшем один из языков HDL, например VHDL. Этот язык поддерживается большинством САПР, и поэтому разработанный проект легко может быть перенесен из одной системы в другую.

Рассмотрим основные элементы пользовательского интерфейса ECS. Структура основного окна схемотехнического редактора ECS показана на рисунке 1.

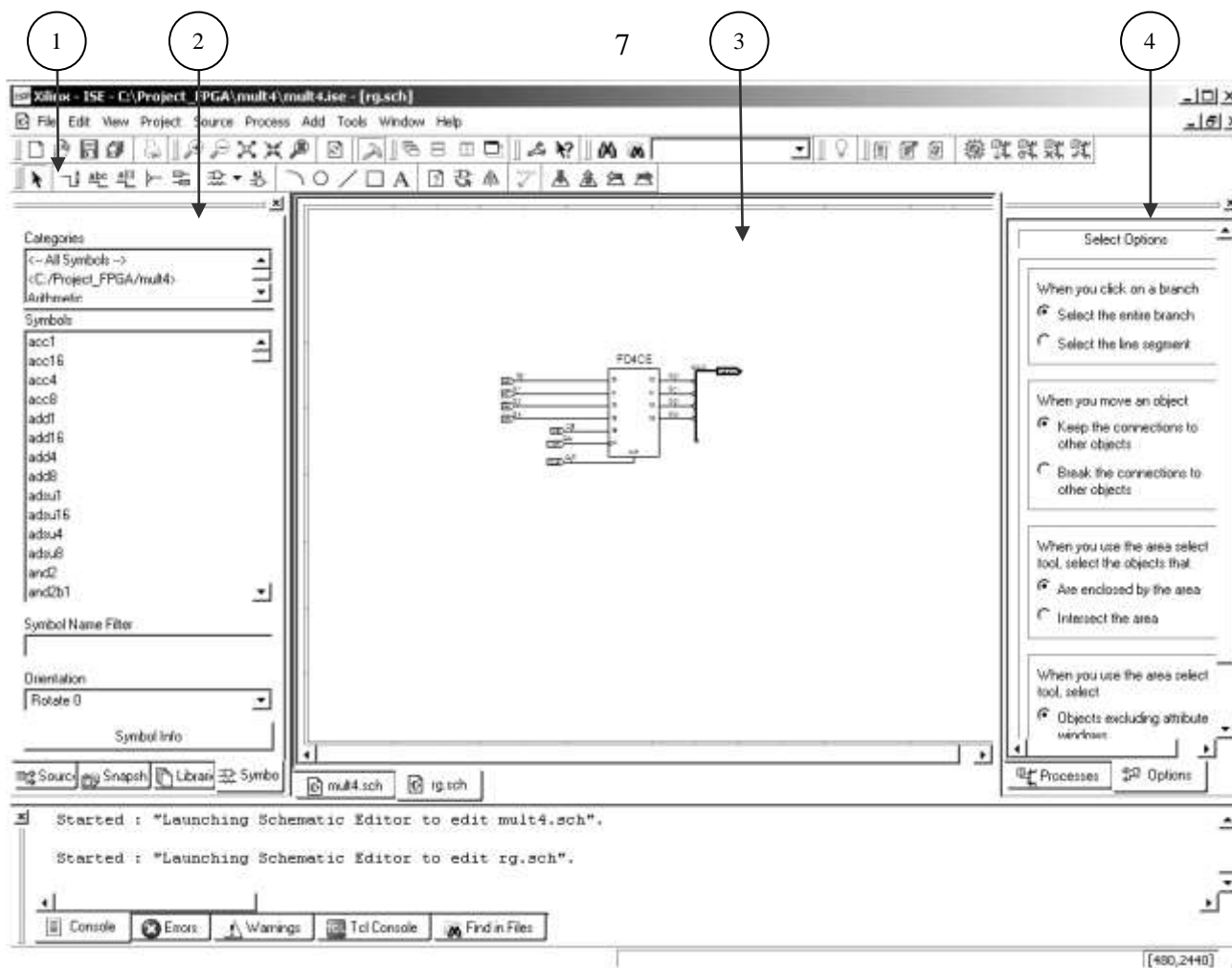





Рисунок 1 – Окно схематического редактора ECS















- 1 – Панель инструментов ECS;
- 2 – Панель библиотек символов;
- 3 – Поле ввода схемы;
- 4 – Панель дополнительных параметров.

Остановимся более подробно на рассмотрении основных кнопок панели инструментов ECS. На рисунке 3 показана панель инструментов ECS.



Рисунок 2 – Панель инструментов ECS

-  - включает режим выбора объектов.
-  - устанавливает режим ввода цепей редактируемой схемы. Аналогично команде подменю Add→Wire (Ctrl+W).
-  - используется для определения названия цепей и шин.

- Add→Net Name (Ctrl + D).
-  - изменяет название выбранной шины и входящих в нее проводников.
 -  - включает режим формирования отводов шины. Аналогично команде подменю Add→Bus Tap(Ctrl + B).
 -  - активирует режим ввода маркеров интерфейсных цепей. Аналогично команде подменю Add→I/O Marker(Ctrl + G).
 -  - используется для ввода библиотечных компонентов схемы. Аналогично команде подменю Add→Symbol (Ctrl+M).
 -  - используется для определения названия компонента схемы. Аналогично команде подменю Add→Instance Name (Ctrl+J).
 -  - позволяет создавать изображение дуги на поле чертежа. Аналогично команде подменю Add→Arc
 -  - позволяет создавать изображение окружности на поле чертежа. Аналогично команде подменю Add→Circle.
 -  - используется для рисования линии на поле чертежа. Аналогично команде подменю Add→Line (Ctrl + L).
 -  - позволяет поместить изображение прямоугольника на поле чертежа. Аналогично команде подменю Add→Rectangle.
 -  - используется для ввода текста на поле чертежа. Аналогично команде подменю Add→Text (Ctrl + T).
 -  - используется для вывода информации об объекте схемы
 -  - позволяет повернуть выбранный объект
 -  - выполняет зеркальное преобразование выбранного объекта
 -  - выполняет проверку корректности ввода схемы. Аналогично команде подменю Tools→Check Schematic.

3 Основные этапы проектирования цифровых устройств в Xilinx ISE

При проектировании цифровых устройств в САПР Xilinx ISE в общем случае можно выделить следующие этапы:

- ✓ создание нового проекта (выбор семейства и типа ПЛИС, а также средств синтеза);
- ✓ подготовка описания проектируемого устройства в схемотехнической, алгоритмической или текстовой форме;
- ✓ синтез устройства;
- ✓ функциональное моделирование;
- ✓ размещение и трассировка проекта в кристалле;
- ✓ временное моделирование;
- ✓ программирование ПЛИС (загрузка проекта в кристалл).

4 Пример проектирования

Порядок использования основных приложений пакета Xilinx ISE рассмотрим на примере разработки простейшего цифрового устройства, реализующего беззнаковое умножение по алгоритму правого сдвига.

Техническое задание на проектируемое устройство можно сформулировать следующим образом: разработать четырехразрядный синхронный умножитель. Для его реализации использовать алгоритм умножения младшими разрядами множителя со сдвигом суммы частичных произведений (СЧП) вправо.

Обратите внимание на то, что представленный способ реализации умножителя не является оптимальным. Цель данного проекта развить навыки использования САПР Xilinx ISE для реализации иерархических проектов.

Кратко опишем основные этапы алгоритма умножения со сдвигом СЧП вправо.

Алгоритм сводится к следующим шагам:

1. Исходное значение СЧП принимается равным 0.
2. Анализируется очередная цифра множителя (анализ начинается с младшей цифры). Если она равна единице, то к СЧП прибавляется множимое, в противном случае (цифра равна нулю) прибавление не производится.
3. Выполняется сдвиг СЧП вправо на один разряд.
4. Пункты 2 и 3 последовательно повторяются для всех цифровых разрядов множителя.

Алгоритм может быть реализован с помощью схемы, показанной на рисунке 3.

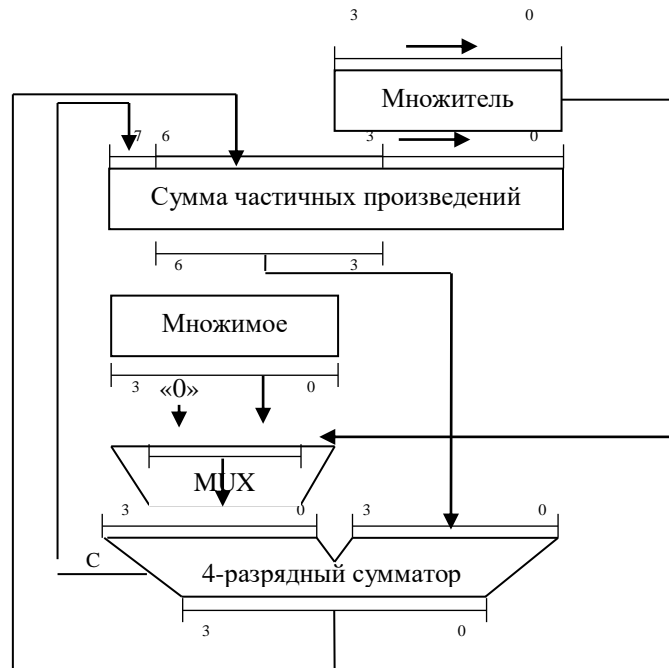


Рисунок 3 – Схема устройства умножения по алгоритму правого сдвига

Множимое и множитель заносятся в 4-разрядные регистры множимого (РМн) и множителя (РМт) соответственно, а все разряды 8-разрядного регистра суммы частичных произведений (РЧП) сбрасываются (устанавливаются в «0»). Умножение происходит за 4 шага. На каждом шаге, в зависимости от состояния младшего разряда множителя, управляющего мультиплексором, на один из входов 4-разрядного сумматора подается либо множимое, либо 0. На второй вход поступает содержимое 4 старших разрядов РЧП. Новое частичное произведение из сумматора пересылается в старшие разряды РЧП. Далее содержимое РЧП сдвигается на один разряд вправо, причем в освободившийся старший разряд регистра заносится значение переноса (С) из старшего разряда сумматора. Поскольку

мультиплексор управляется младшим разрядом РМт, то содержимое этого регистра тоже сдвигается на один разряд вправо. Описанная последовательность выполняется 4 раза.

Создание нового проекта

Прежде чем переходить непосредственно к созданию проекта целесообразно создать отдельную директорию для размещения своих проектов, например C:\Project_FPGA.

Для создания нового проекта необходимо запустить Навигатор проектов и выполнить команду **File**→**New Project**. В результате появится диалоговое окно, показанное на рисунке 4.

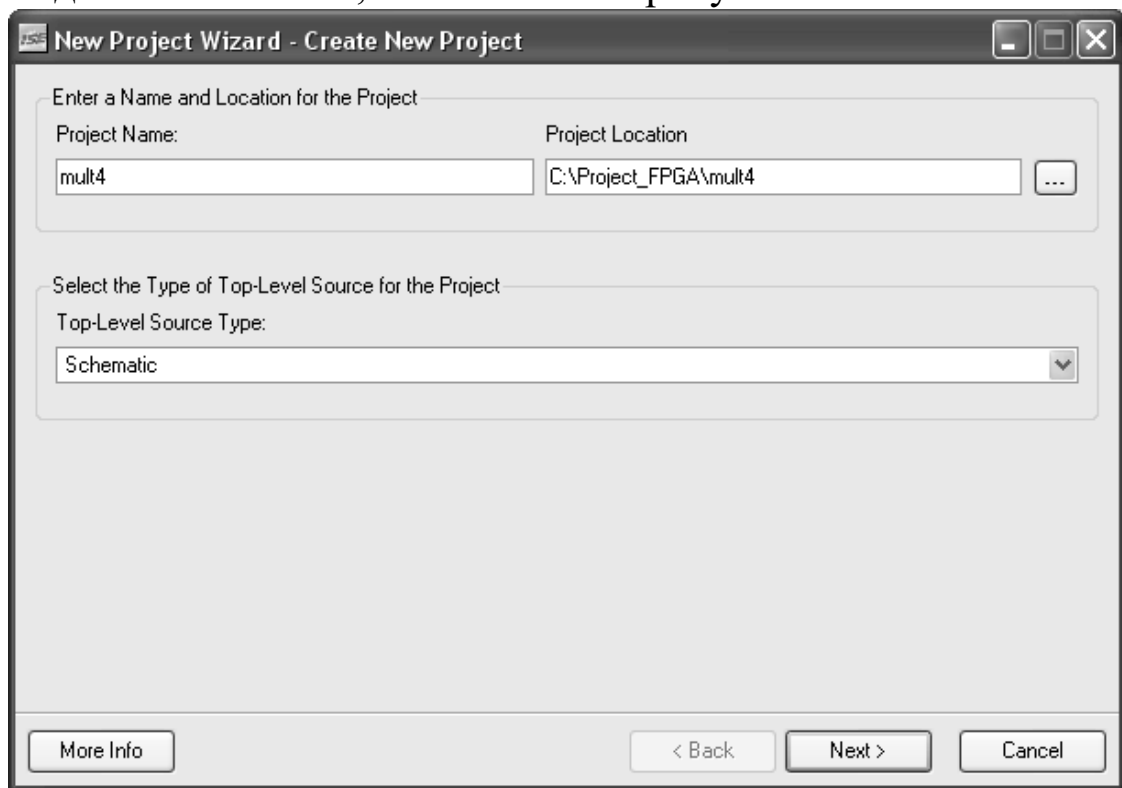


Рисунок 4 –Диалоговое окно создания нового проекта

В данном окне должны быть представлены исходные данные для создания проекта:

1. Название проекта (Project Name);
2. Диск и каталог, в котором предполагается расположить проект (Project Location);
3. Тип представления модуля верхнего уровня (Top-Level Source Type).

В данном примере имя проекта – mult4. В качестве типа модуля верхнего уровня для наглядности представления схемных решений был выбран тип Schematic (тип HDL предполагает применение одного из языков описания аппаратуры, EDIF и NGC/NGO – использование внешних средств логического синтеза).

Нажав Next, появляется окно свойств устройства, показанное на рисунке 5.

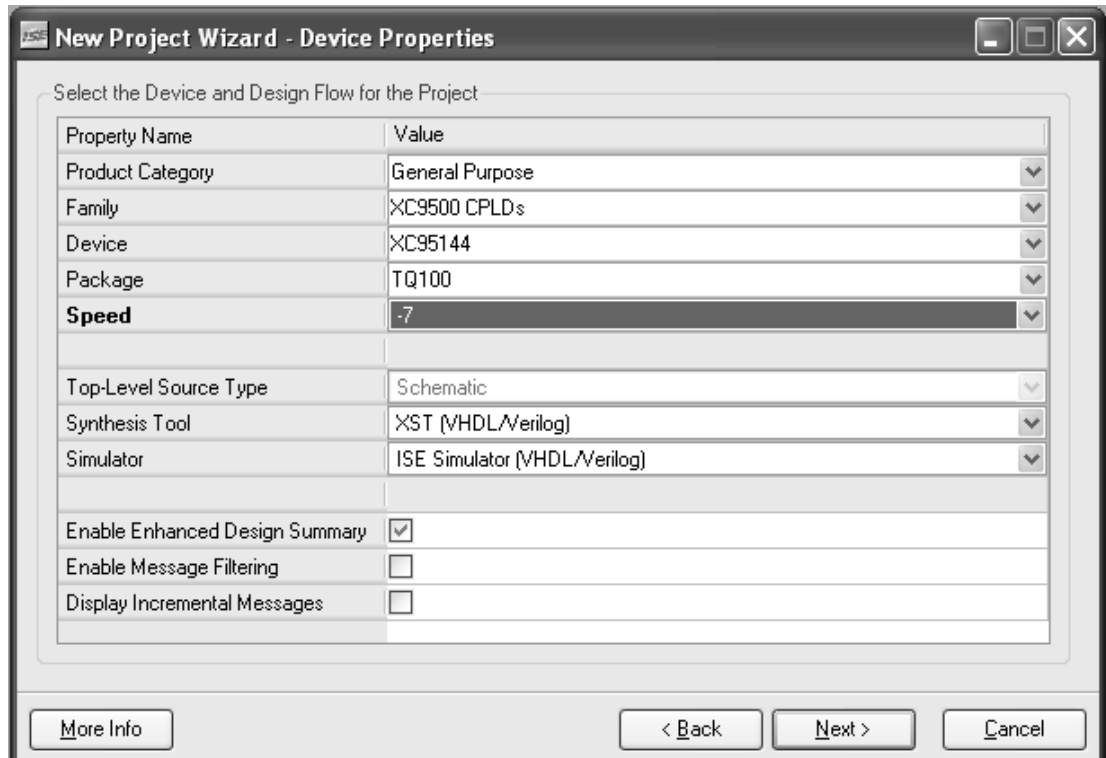


Рисунок 5 – Окно свойств устройства

В данном окне необходимо задать следующие параметры:

1. Категория продукта (Product Category);
2. Семейство ПЛИС (Family);
3. Тип кристалла (Device);
4. Тип корпуса (Package);
5. Быстродействие кристалла (Speed);
6. Средства синтеза устройства (Synthesis Tool);
7. Средства моделирования (Simulator).

В качестве примера для реализации устройства умножения было выбрано семейство ПЛИС XC9500 CPLDs, тип кристалла XC95144 в корпусе TQ100.

Нажав Next, на экране появиться окно создания файла исходного описания, показанное на рисунке 6.

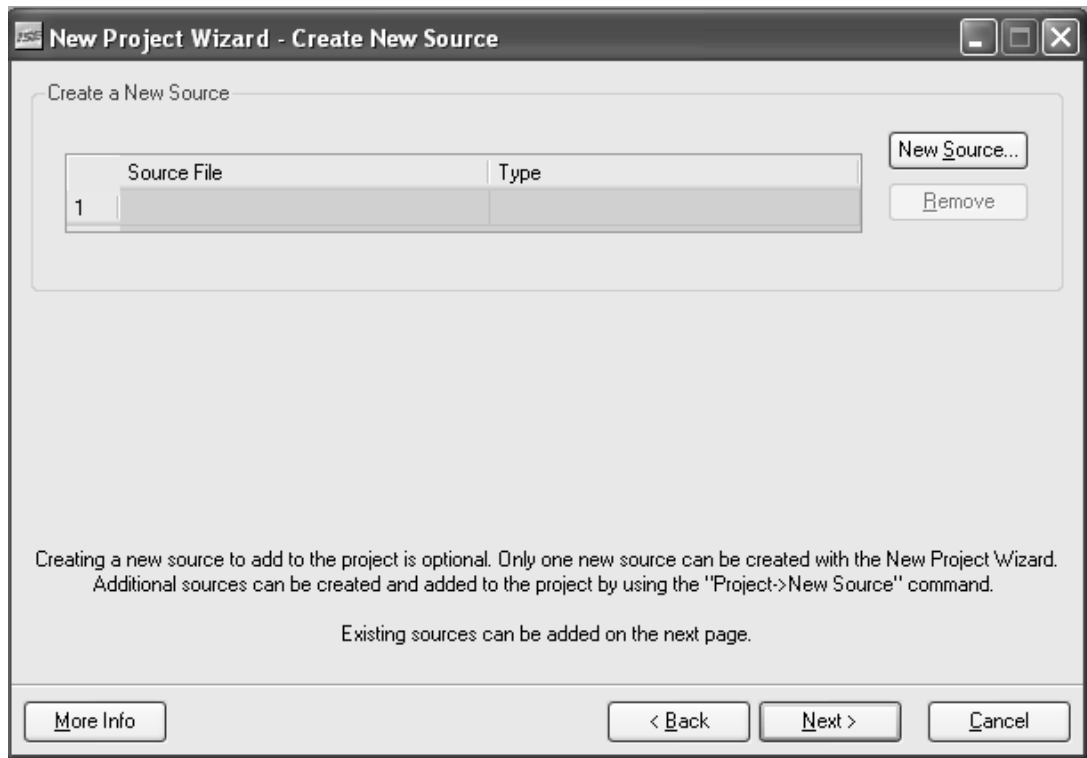


Рисунок 6 – Окно создания файла исходного описания

Нажав New Source, выберите тип файла исходного описания, как показано на рисунке 7.

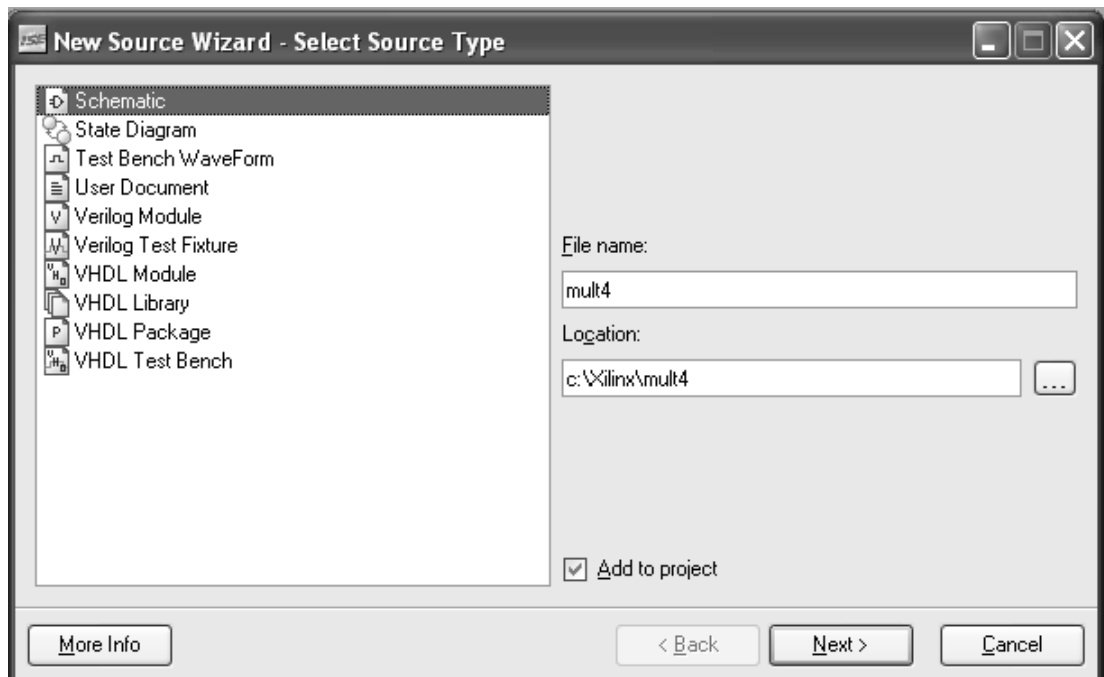


Рисунок 7 – Окно выбора типа файла исходного описания

В данном случае для описания модуля верхнего уровня будет использоваться схемотехнический редактор.

После нажатия Next, на экране будет отображено окно итоговой информации о созданном модуле, показанное на рисунке 8.

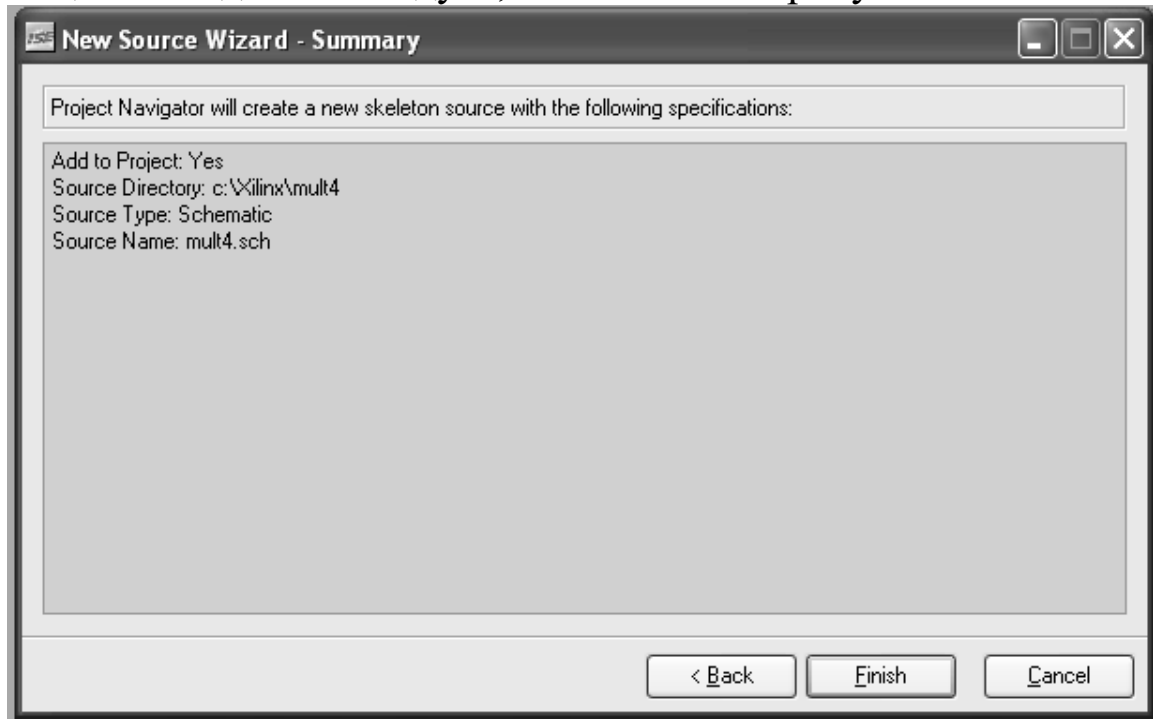


Рисунок 8 – Итоговая информация о созданном модуле

Далее нажмите Finish, а затем на предложение добавить к проекту существующий модуль нажмите Next.

Итоговая информация о созданном проекте показана на рисунке 9.

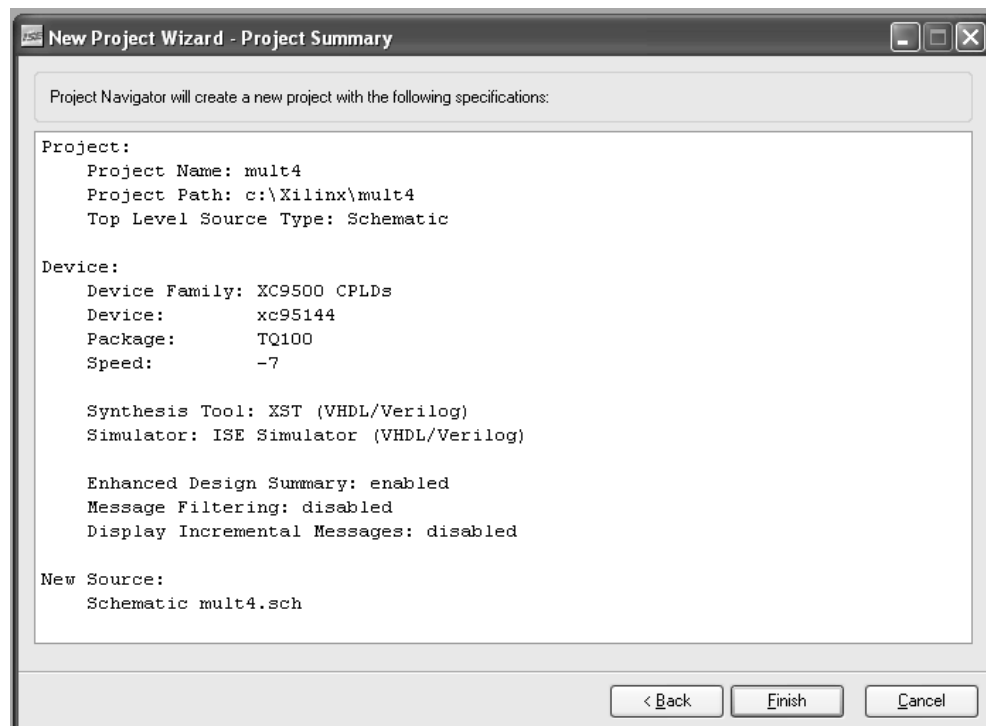


Рисунок 9 – Итог создания проекта

После нажатия кнопки Finish, станет доступно специальное приложение пакета – схмотехнический редактор ECS, окно которого показано на рисунке 10.

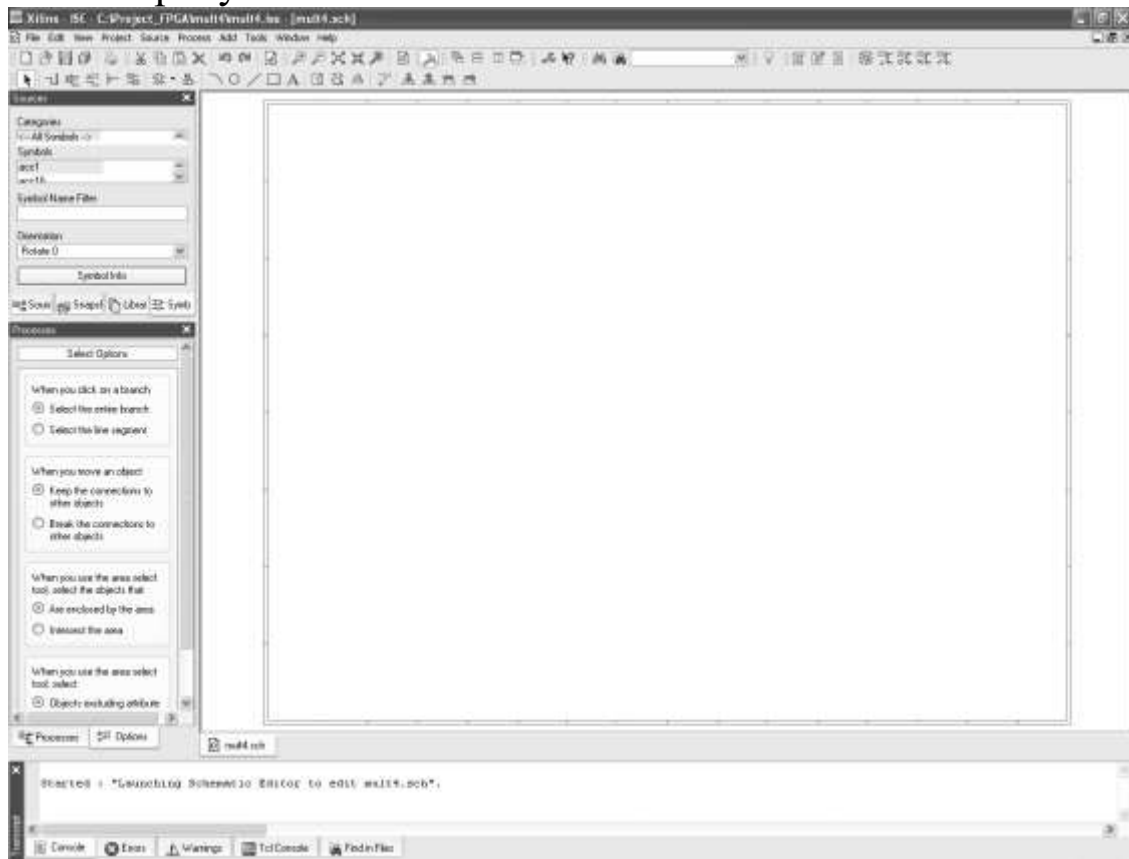


Рисунок 10 – Окно схмотехнического редактора ECS

Создание схемы в редакторе ECS

Процесс создания принципиальной схемы проектируемого устройства в редакторе ECS включает в себя выполнение следующих операций:

- ✓ ввод символов элементов схемы;
- ✓ выполнение необходимых соединения в виде проводников и шин;
- ✓ ввод имен цепей и шин;
- ✓ установка маркеров, определяющих назначение цепей, используемых для подключения «внешних» элементов или выводов ПЛИС;
- ✓ Ввод необходимых атрибутов.

Для разрабатываемого устройства целесообразно включить два уровня иерархии, чтобы не загромождать основной проект.

Первый уровень иерархии – основные функциональные блоки устройства умножения (регистры, сумматор и устройство управления).

Второй уровень иерархии – проект в целом.

Согласно рисунку 4 в состав устройства умножения входят следующие основные блоки: 4-разрядный регистр хранения (РМн), 4-разрядный (РМт) и 8-разрядный (РЧП) регистры сдвига вправо, 4-разрядный сумматор и двухканальный мультиплексор.

Опишем подробно процесс создания схемы на примере 4-разрядного регистра сдвига с параллельной загрузкой. Создадим новый модуль **shr4.sch** и добавим его к основному проекту как показано на рисунке 11.

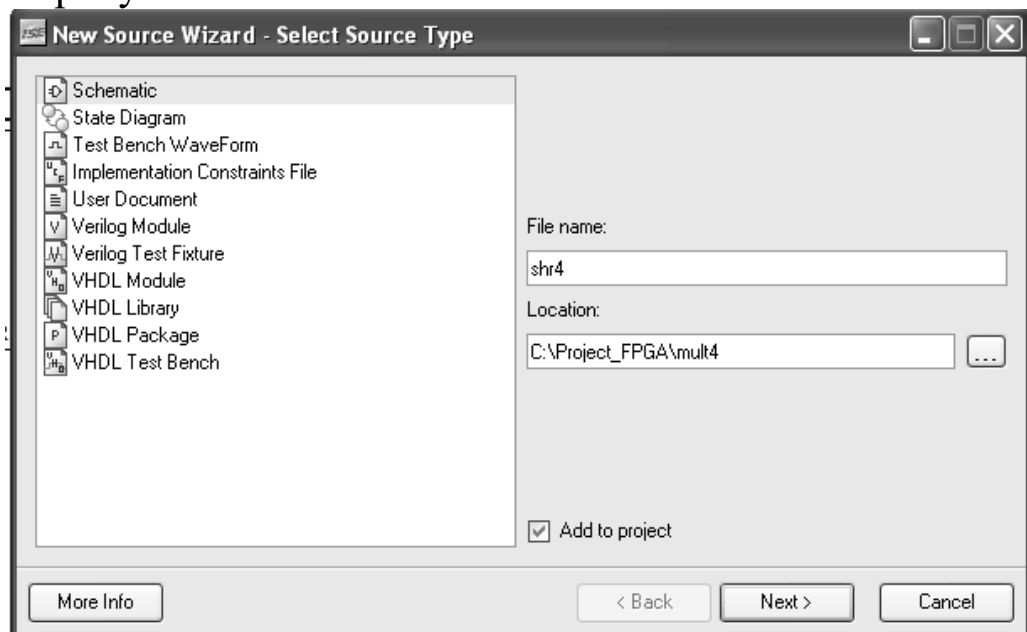


Рисунок 11 – Окно создания модуля shr4.sch

Прежде чем приступить непосредственно к вводу схемы рекомендуется проверить и при необходимости установить параметры страницы. Для этого необходимо активизировать диалоговое окно общих параметров схемы, поместив курсор на поле ввода чертежа и дважды щелкнув левой кнопкой мыши. Диалоговое окно общих параметров схемы показано на рисунке 12.

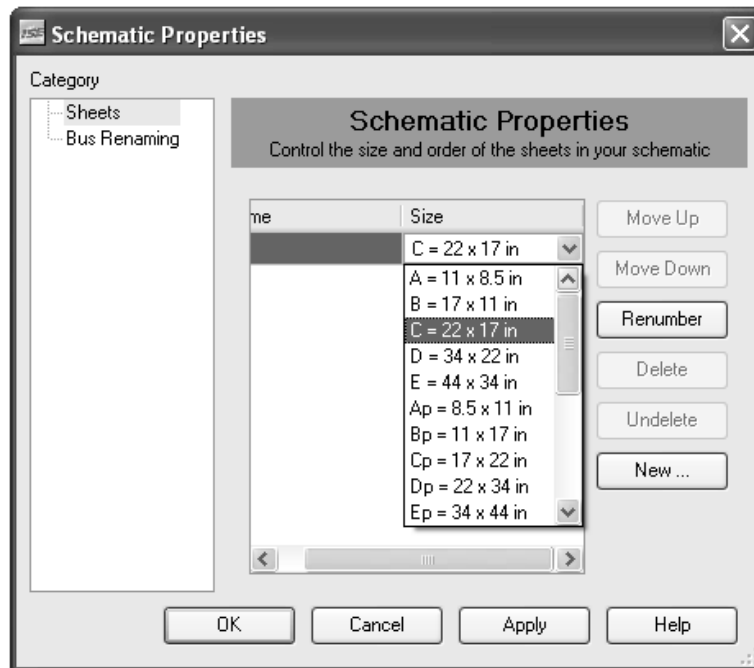


Рисунок 12 – Окно общих параметров схемы

При открытии окна схемотехнического редактора активизируется основной режим – выбора объекта, установленный по умолчанию. В этом режиме осуществляется выделение, перемещение и удаление элементов схемы, а также просмотр и редактирование их параметров.

Для активизации режима ввода символов компонентов создаваемой схемы необходимо выполнить команду Add→Symbol (Ctrl+M), либо нажать соответствующую пиктограмму панели инструментов схемотехнического редактора. Также режим ввода символов автоматически включается при выборе символа на странице библиотек **Symbols**, как показано на рисунке 13.

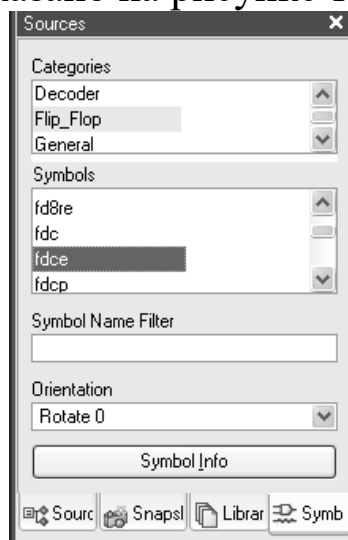


Рисунок 13 – Окно Sources

Прежде всего, необходимо выбрать нужную функциональную группу символов библиотеки компонентов в поле **Categories**. Далее следует выбрать в поле **Symbols** необходимый компонент, после чего курсор мыши следует переместить на поле чертежа. Затем необходимо зафиксировать элемент, щелкнув левой кнопкой мыши. В случае успешного выполнения указанных операций на поле чертежа появляется детальное изображение выбранного компонента.

Для реализации 4-разрядного реестра сдвига с параллельной загрузкой нам понадобится 4 D-триггера и 4 одноразрядных мультиплексора. Размещение элементов на поле чертежа показано на рисунке 14.

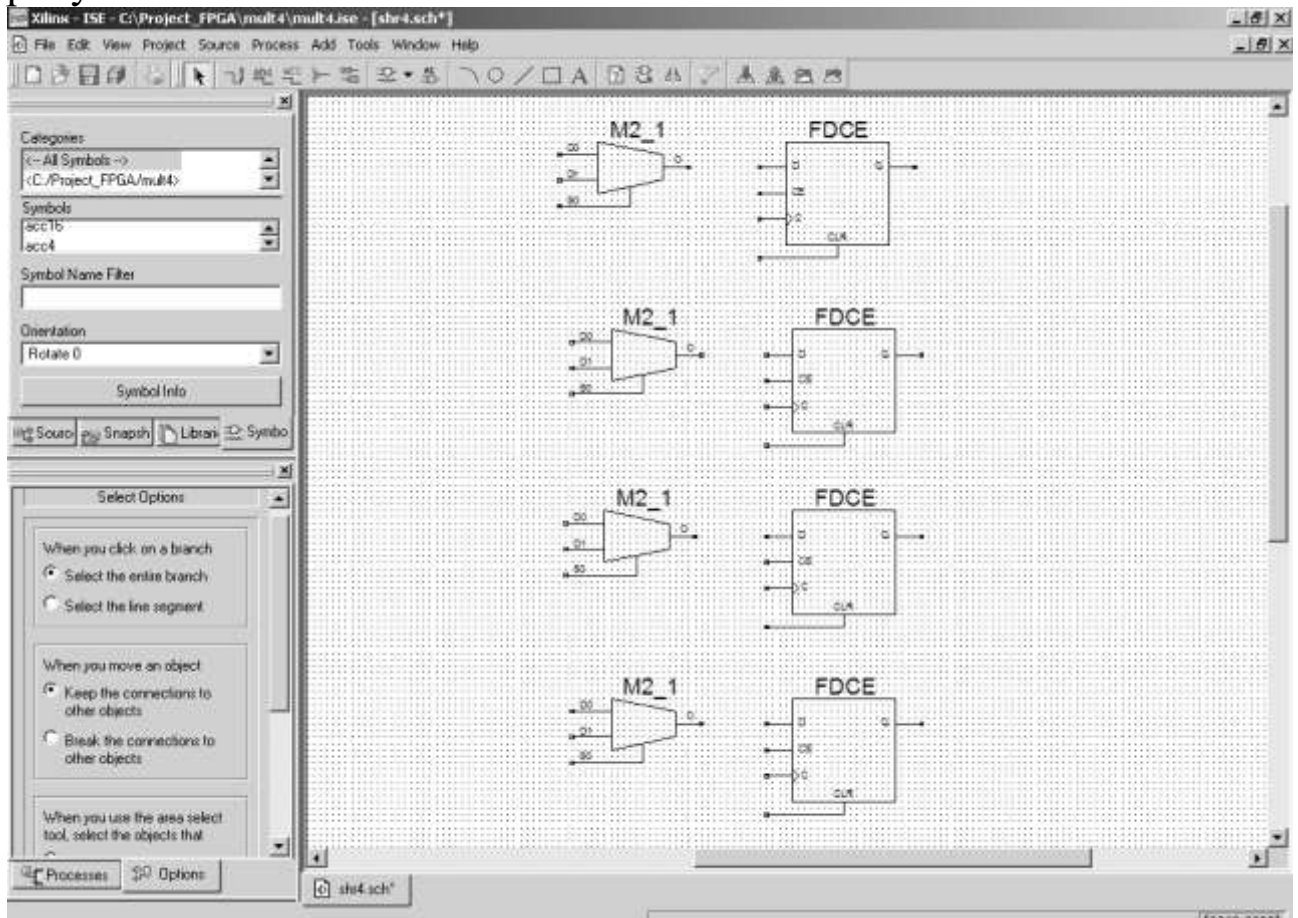


Рисунок 14 – Размещение элементов на поле чертежа

Следующий шаг в процессе создания схемы - выполнение всех необходимых соединений. Для этих целей в редакторе ECS используются проводники (Wire) и шины (Bus). Включение режима ввода проводников производится при выборе команды **Wire** из выпадающего меню **Add** или нажатием соответствующей кнопки на панели инструментов ECS.

Формирование цепи начинается с фиксации стартовой точки, которая может располагаться на свободном месте поля чертежа или совпадать с контактом вывода одного из компонентов. Для этого следует поместить курсор мыши в требуемую точку на поле чертежа и щелкнуть левой кнопкой мыши. Дальнейшая последовательность действий зависит от режима трассировки цепей на изображении схемы, который выбирается с помощью кнопок на странице **Options** в панели дополнительных параметров (рис. 15). Если кнопка **Use the Manual method to add single line segments between the points you indicate** находится в нажатом состоянии, то установлен режим ручной трассировки цепей на изображении схемы. При этом после фиксации начальной точки цепи следует переместить курсор мыши в позицию, соответствующую точке изгиба, соединения с другим проводником или контактом компонента, а также конечной точке цепи. При этом формируемый сегмент цепи отображается штриховой линией. Фиксация сегмента осуществляется щелчком левой кнопкой мыши в конечной точке, после чего новый фрагмент цепи отображается основной пинией (рис. 15).

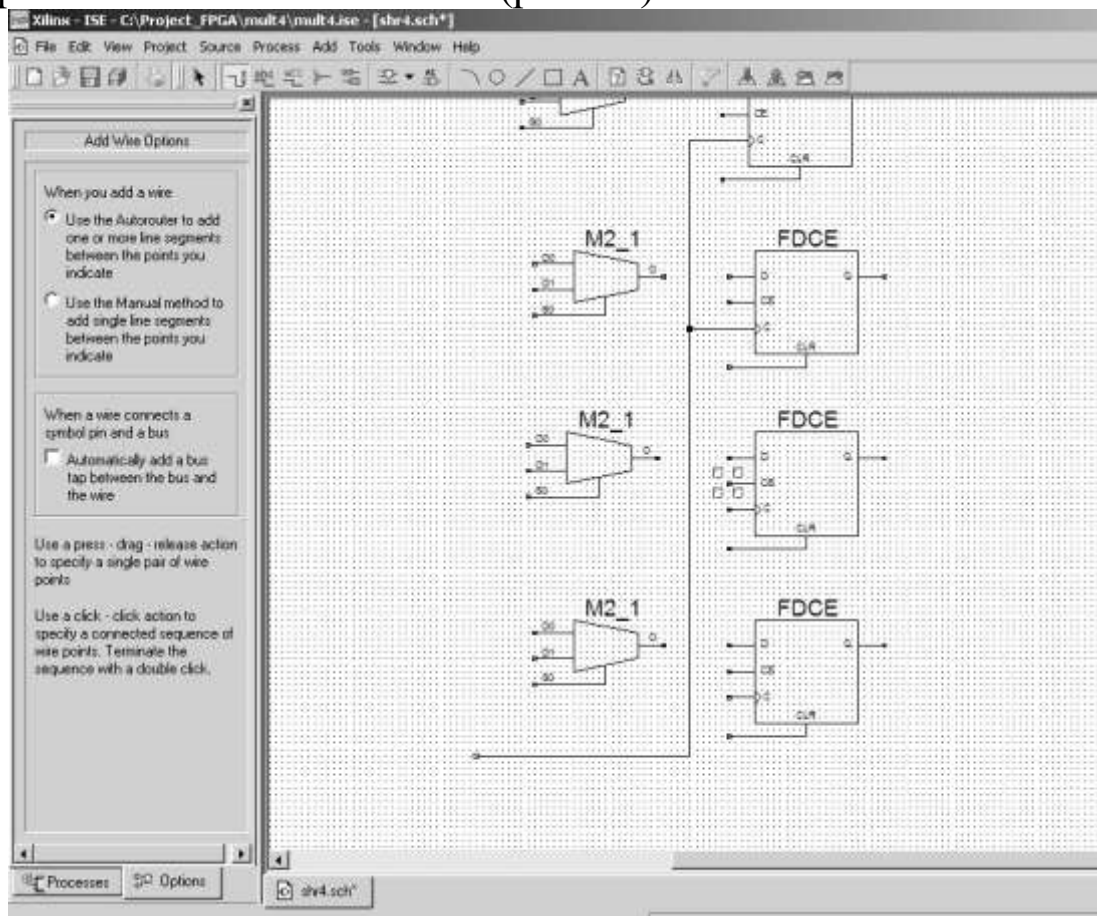


Рисунок 15 – Соединение компонентов схемы с помощью проводников (цепей)

Для формирования соединений элементов схемы в виде шин необходимо выполнить следующую последовательность действий. Вначале в режиме ввода проводников создается графическое изображение шины в виде фрагмента одиночной цепи в соответствии с инструкциями, рассмотренными выше. До тех пор, пока не задано название шины в соответствующем формате (с указанием разрядности или перечислением проводников), она отображается сплошной тонкой линией, как одиночная цепь.

Затем следует перейти в режим формирования отводов шины, нажав соответствующую кнопку на панели инструментов или выполнив команду **Add→Bus Tap (Ctrl+B)** всплывающего подменю **Add**. При этом к курсору присоединяется изображение отвода шины. Символ отвода шины может быть подключен к вертикальному или горизонтальному сегменту шины. Для получения нужной ориентации изображения отвода шины следует воспользоваться группой кнопок, расположенных на странице **Options** в панели дополнительных параметров. Если в нажатом состоянии находится кнопка **Left**, то изображение отвода шины сориентировано так, что контакт для подключения проводника находится справа, а для присоединения к шине - слева. При нажатии кнопки **Right** справа располагается сторона для присоединения к шине, а слева - контакт для подключения цепи. Нажатое состояние кнопки **Top** соответствует ориентации символа отвода шины, при которой контакт для подключения проводника располагается внизу, а сторона, присоединяемая к шине - вверху. При нажатии кнопки **Bottom** устанавливается ориентация противоположная по отношению к задаваемой кнопкой **Top**.

На рисунке 16 показан режим формирования отводов цепи.

Сформировав, таким образом, все отводы шины производится их соединение с соответствующими цепями схемы. После подключения проводников к отводам шины производится присвоение соответствующих названий этих цепей, командой **Add→Net Name (Ctrl + D)**.

Процесс ввода названий цепей и шин представлен на рисунке 17.

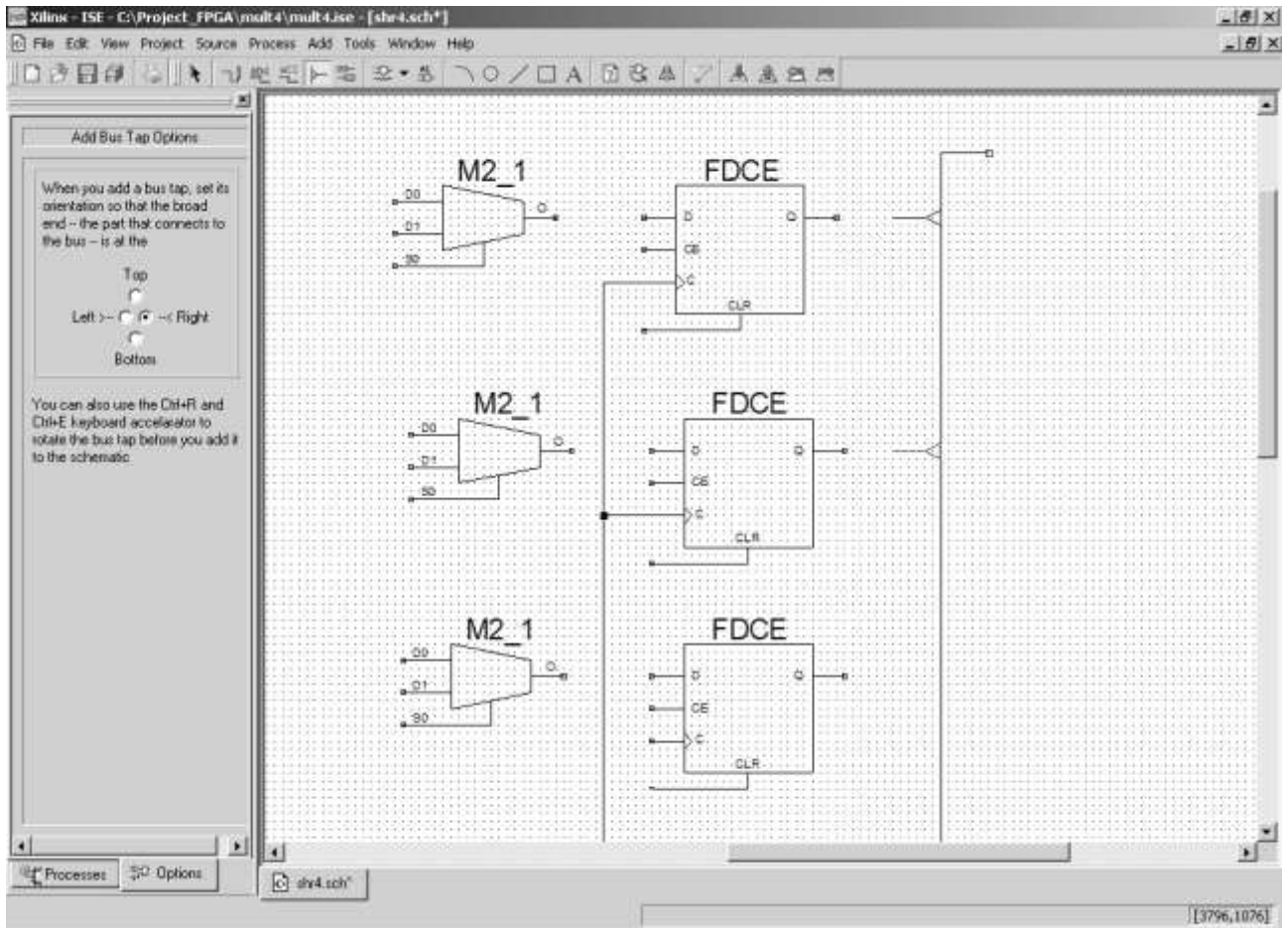


Рисунок 16 – Режим формирования отводов шины

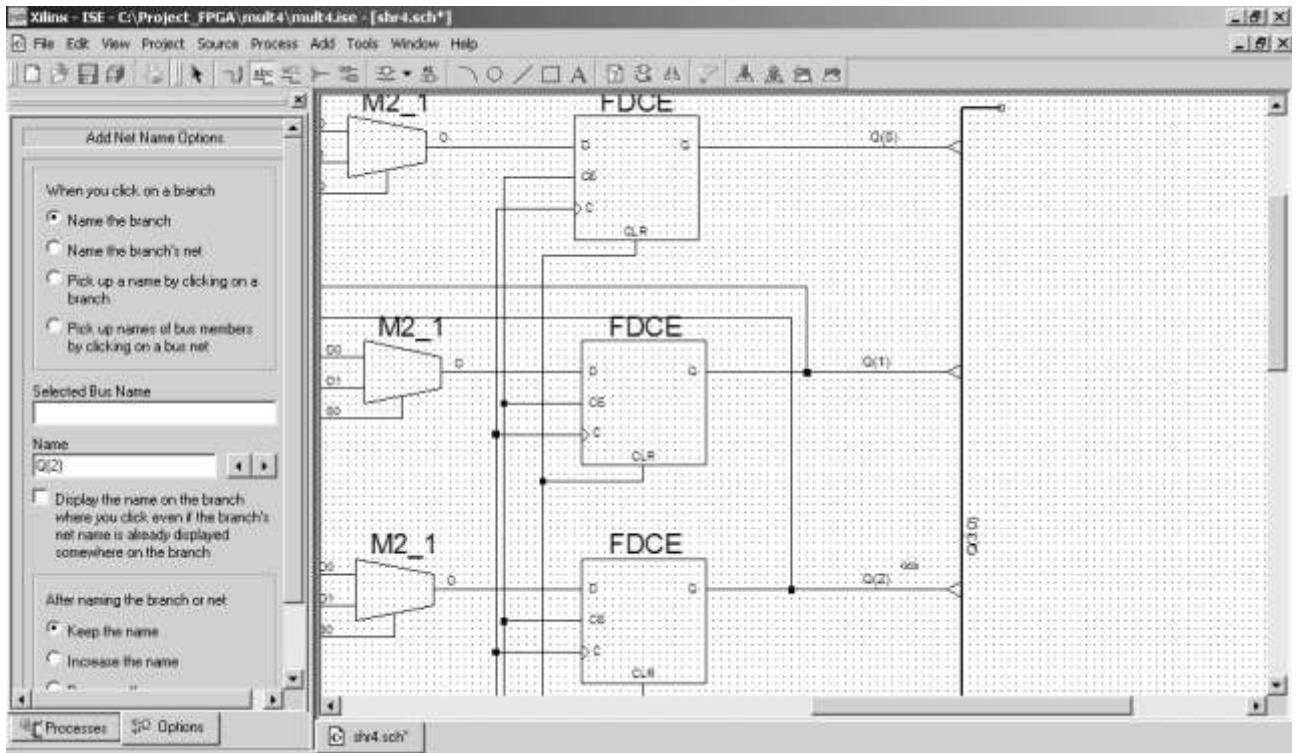


Рисунок 17 – Ввод названий цепей и шин

С помощью кнопок, расположенных на странице Options, следует выбрать способ ввода названия цепей. При этом необходимо учитывать, что если на схеме присутствуют два или более проводника с одинаковыми названиями, но не имеющие точек соединений, то они объединяются электрически в одну цепь. В этом случае считается, что цепь состоит из нескольких ветвей. При нажатой кнопке **Name the branch** устанавливается название одной (выбранной) ветви цепи. Если кнопка **Name the branch's net** находится в нажатом состоянии, то одно название присваивается всем ветвям цепи. Для цепей, состоящих из одной ветви, при фиксации любой из этих двух кнопок получается один и тот же результат. После выбора одной из этих кнопок следует активизировать поле ввода имени цепи, поместив на него курсор и щелкнув левой кнопкой мыши. Название цепи набирается в этом поле с помощью клавиатуры. Кнопки, расположенные справа от поля ввода названия цепи, позволяют уменьшить или увеличить значение индекса, если он используется в имени цепи. Введенный текст названия привязывается к курсору мыши при перемещении последнего на поле чертежа схемы. Далее необходимо поместить указатель на изображение соответствующей цепи и щелкнуть левой кнопкой мыши. При отсутствии ошибок название цепи отображается рядом с ее изображением, в позиции, зафиксированной при щелчке левой кнопки мыши (рис. 17).

Названия шин устанавливаются в том же режиме и той же последовательности, что и имена цепей. При этом нужно обратить внимание на то, что в схемотехническом редакторе ECS различаются два типа шин: упорядоченные и неупорядоченные. В упорядоченной шине все проводники имеют одно и то же имя, совпадающее с названием шины, и соответствующий порядковый номер. В состав неупорядоченной шины входят цепи с произвольными названиями.

Для упорядоченной шины название задается в формате <имя шины> («конечный номер проводника в шине»:<начальный номер проводника в шине>), например, q(3:0) как в нашем примере.

Названия проводников упорядоченной шины указываются в формате <имя шины> (<номер проводника в шине>), например, q(0).

Формат имени неупорядоченной шины имеет следующий вид <имя цепи 1>, <имя цепи 2>[, <имя цепи 3>.....<имя цепи N>], например, CLOCK, RESET, SETUP. Названия отводов

неупорядоченной шины совпадают с соответствующими именами цепей, входящими в состав названия шины.

После формирования цепей, предназначенных для соединений с другими схемами и контактами кристалла, и присвоения им соответствующих названий необходимо установить маркеры, определяющие назначение этих цепей и направление передачи данных по ним. Маркер присоединяется к свободной конечной точке цепи. Для включения режима ввода маркеров цепей следует выполнить команду Add→I/O Marker, после чего к курсору привязывается изображение маркера. Его тип (входной, выходной или двунаправленный) указывается с помощью группы Кнопок с зависимой фиксацией, расположенных на странице **Options** панели дополнительных параметров как показано на рисунке 18.

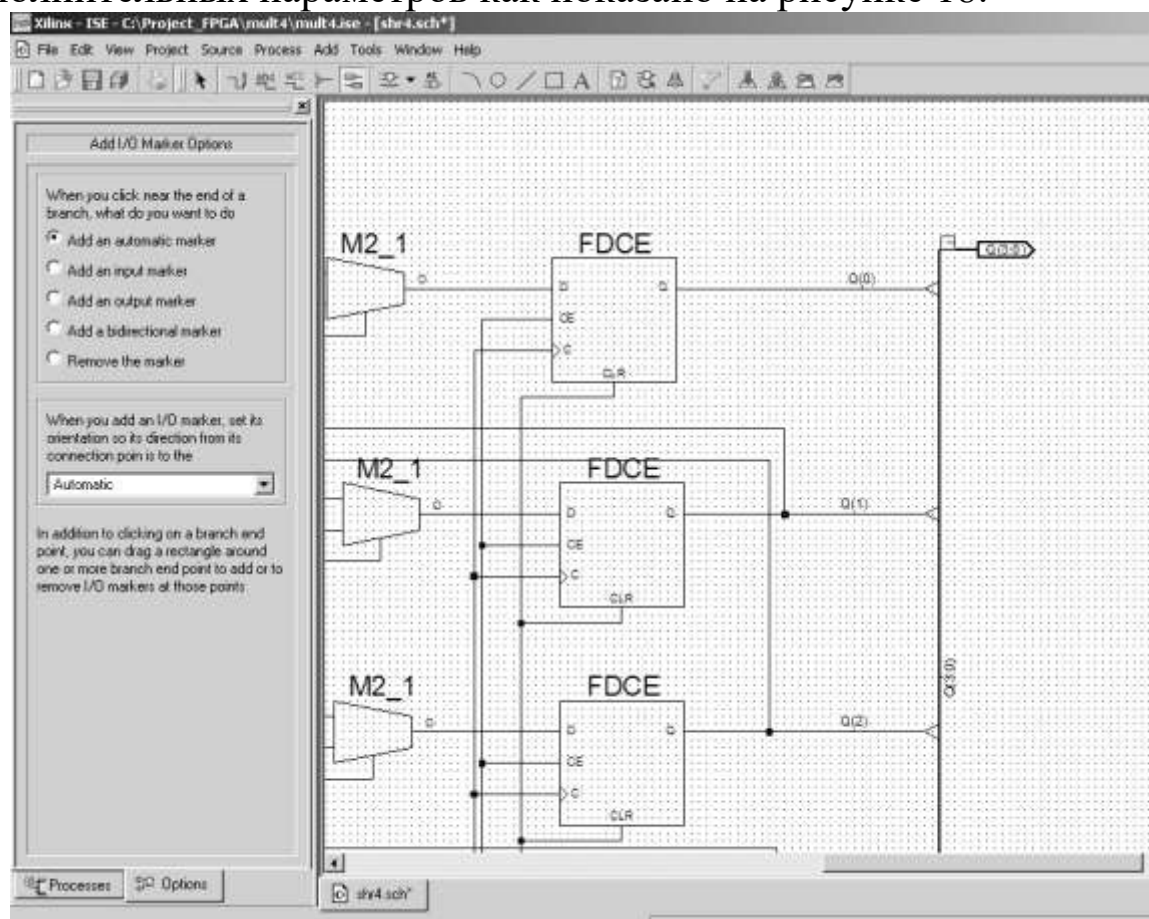


Рисунок 18 – Установка маркеров цепей

Для изменения типа маркера или атрибута какого-либо элемента схемы необходимо перейти в режим выбора объекта, нажав соответствующую кнопку на панели инструментов ECS, поместить курсор на изображения объекта и дважды щелкнуть левой кнопкой мыши.

Пример диалогового окна свойств объекта, в данном случае маркера цепи, показан на рисунке 19.

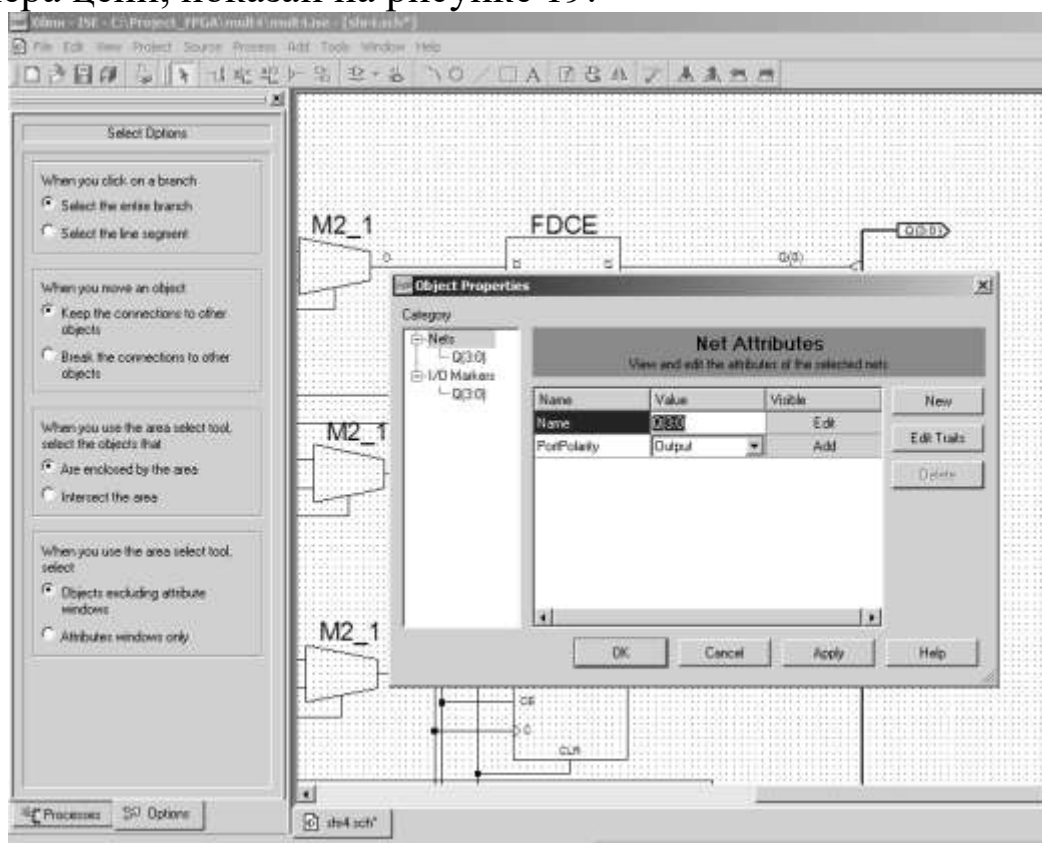


Рисунок 19 – Диалоговое окно свойств маркера цепи

После завершения чертежа схемы необходимо выполнить ее проверку. Часть возможных ошибок обнаруживается уже в процессе создания схемы. Но полный контроль может быть выполнен только для законченной схемы. Для проверки разработанной схемы предназначена команда **Check Schematic**, которая располагается в выпадающем меню **Tools**, а также соответствующей кнопкой на инструментальной панели схемотехнического редактора. В процессе верификации осуществляется контроль целостности схемы и выполнения правил электрических соединений. На этом этапе выявляются такие ошибки, как неподключенные цепи, контакты и отводы шины, а также ошибки, возникающие при соединении выходов нескольких компонентов. Результаты проверки отображаются в окне консольных сообщений, пример которого показан на рисунке 20.



Рисунок 20 – Окно консольных сообщений после выполнения проверки

Законченная схема 4-разрядного регистра сдвига вправо с параллельной загрузкой представлена на рисунке 21.

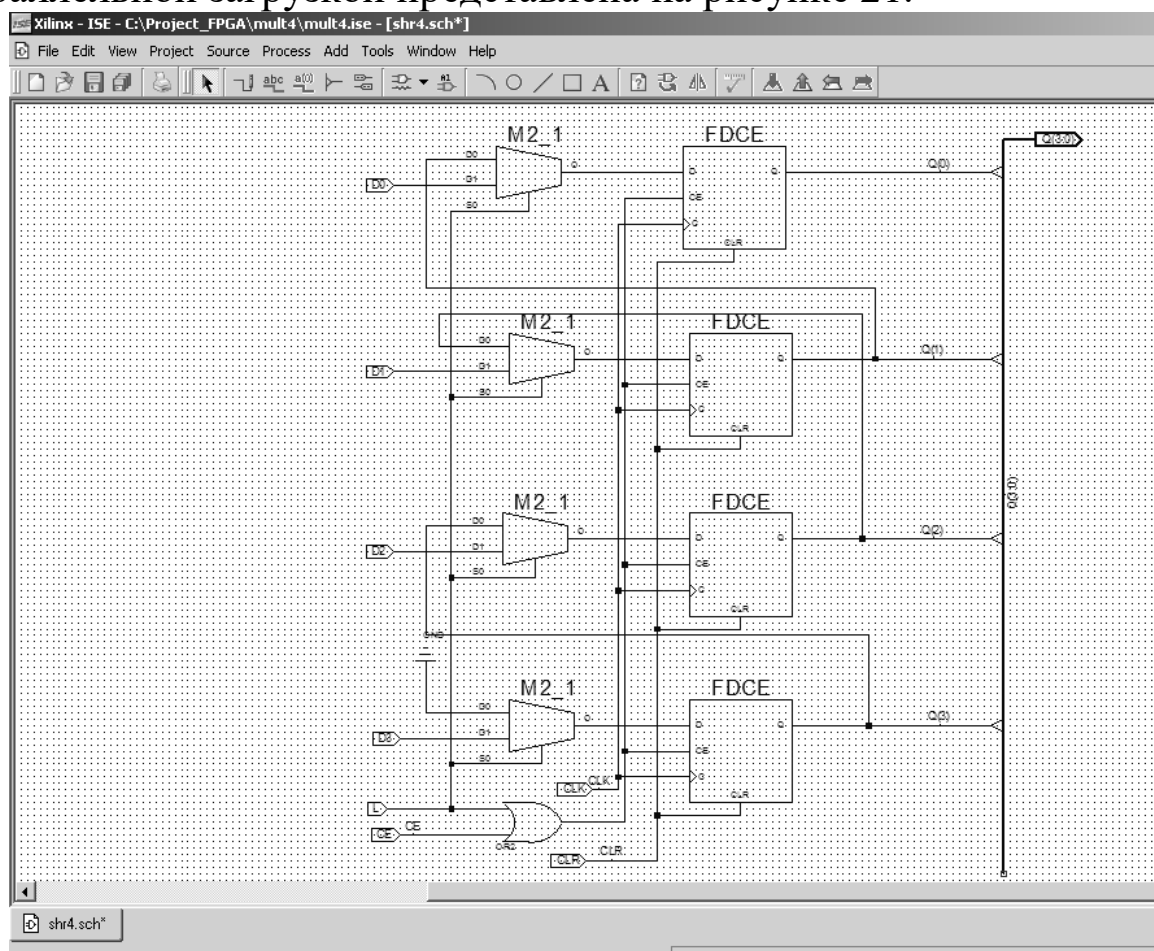


Рисунок 21 – Схема 4-разрядного регистра сдвига вправо с параллельной загрузкой

Выполняя аналогичные действия, создаем модули исходных описаний остальных функциональных блоков устройства умножения, а именно **mux4_2_1.sch**, **shr8.sch**, **rg.sch**. В качестве сумматора можно использовать 4-разрядный сумматор **ADD4** из библиотеки компонентов схемотехнического редактора ECS.

Далее приведем только принципиальные схемы функциональных блоков созданные в схемотехническом редакторе.

На рисунке 22 показана принципиальная схема двухканального мультиплексора, построенная на базе 4-х одноразрядных мультиплексоров.

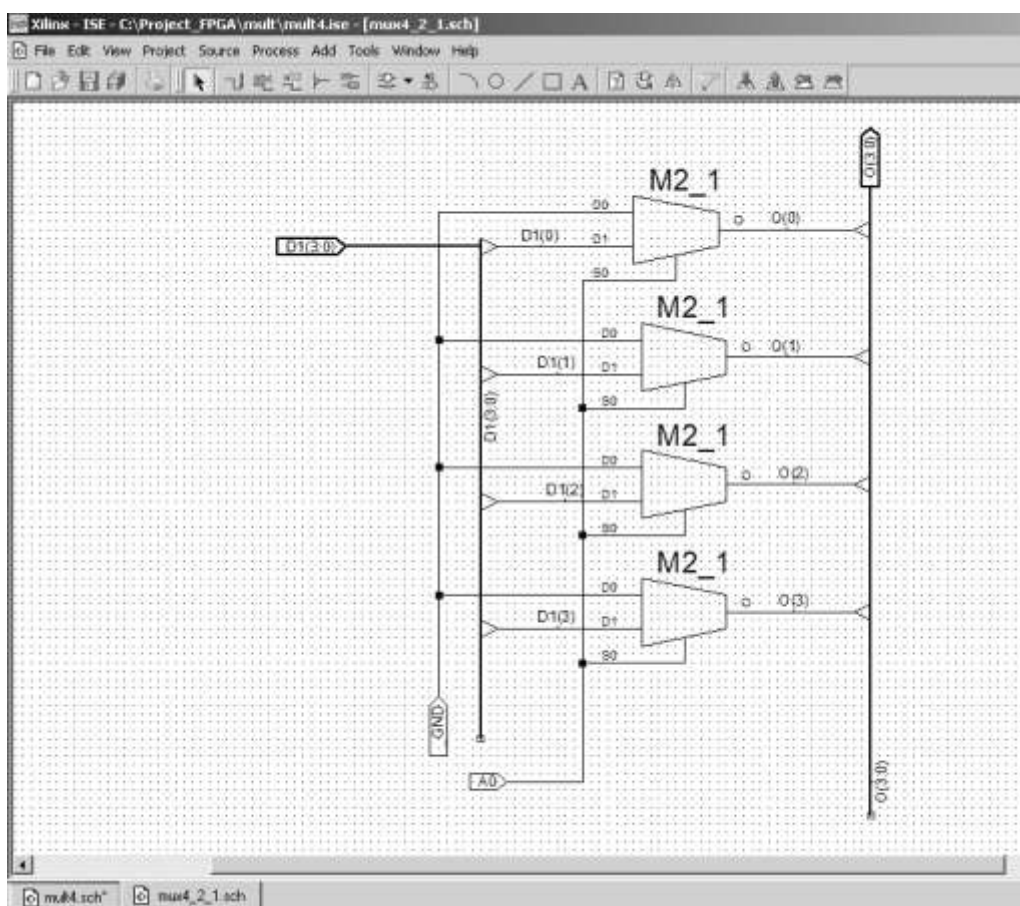


Рисунок 22 – Схема двухканального мультиплексора

На рисунке 23 представлена схема 4-разрядного регистра хранения, созданная на базе библиотечного компонента FD4CE – 4-разрядного регистра. Для удобства выходы FD4CE объединили в 4-разрядную шину.

Схема 8-разрядного регистра сдвига вправо с параллельной загрузкой аналогична изображенной на рисунке 21.

Таким образом, основные функциональные блоки операционного автомата умножителя готовы. Остается разработать устройство управления (УУ) умножителем.

УУ реализует функцию управления ходом вычислительного процесса. Для операционного автомата умножителя УУ должно вырабатывать следующие микрооперации Y_i , i – где номер операции:

Y_0 – загрузка операндов в регистры РМн (rg) и РМт (shr4), сброс регистра РЧП (shr8):

Y_1 – сдвиг содержимого регистра shr4 на один разряд вправо;

Y_2 – запись СЧП в регистр shr8;

Y_3 – сдвиг содержимого регистра shr8 на один разряд вправо;

На рисунке 24 показан один из вариантов реализации УУ.

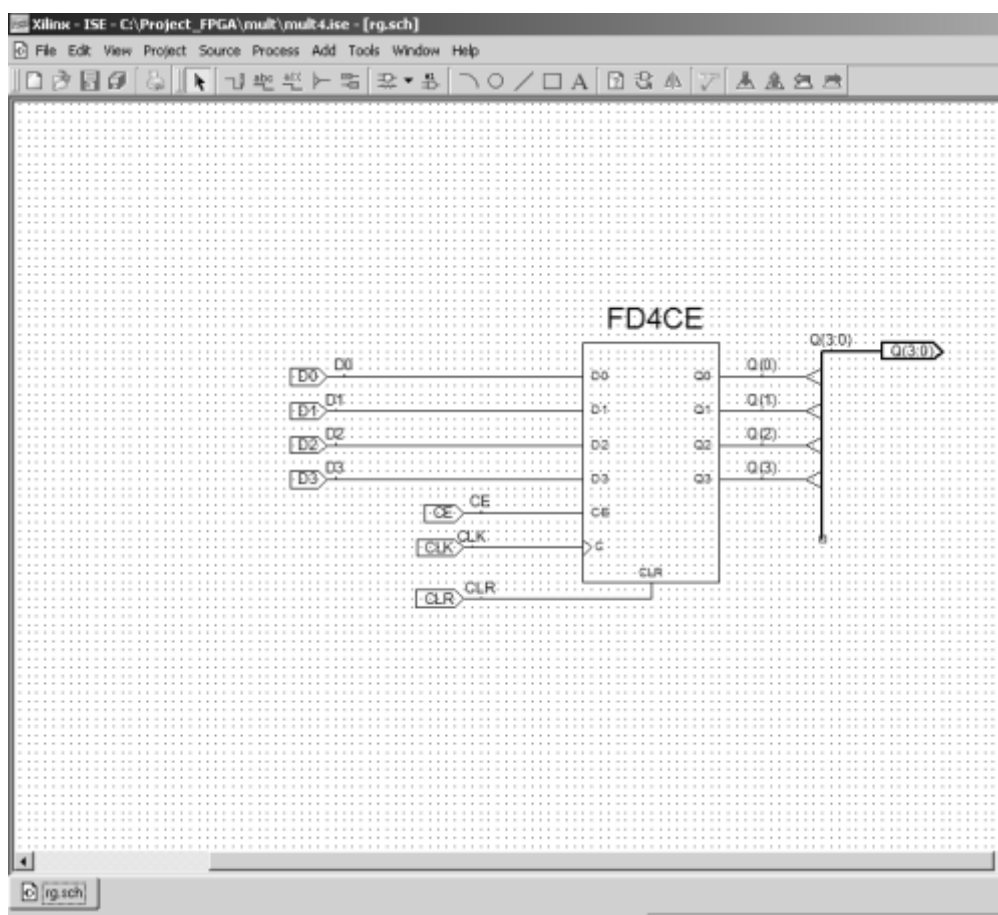


Рисунок 23 – Схема 4-разрядного регистра хранения

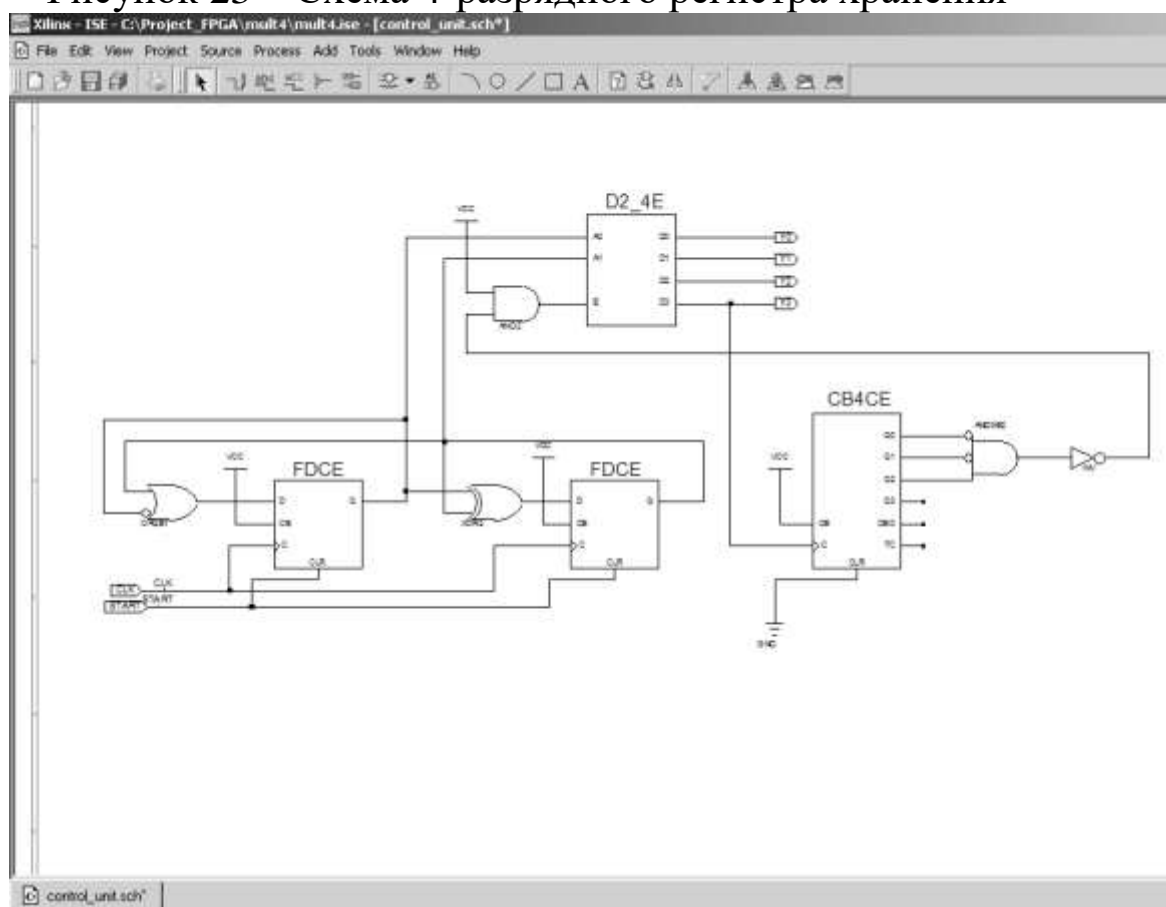


Рисунок 24 – Принципиальная схема УУ

Следующим этапом при проектировании цифрового устройства в схемотехническом редакторе является создание нового символа.

Создание нового символа

Так как проект устройства умножения является иерархическим, то для включения модулей нижнего уровня иерархии в верхний необходимо создать символ.

Схемотехнический редактор ECS предоставляет возможность автоматического формирования символов. «Мастер» автоматического создания символа запускается при выборе команды **Symbol Wizard** из выпадающего меню **Tools**. Рассмотрим процесс создания нового символа для модуля **shr4.sch**.

На рисунке 25 показано окно мастера автоматического создания символа.

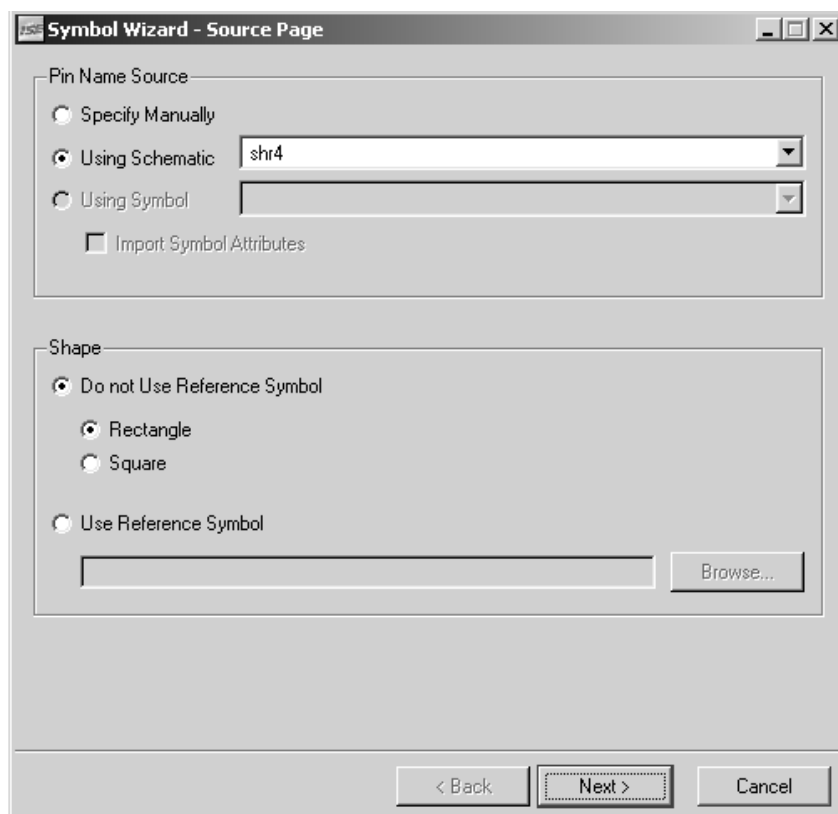


Рисунок 25 – Диалоговое окно «мастера» автоматического создания символа

Прежде всего, нужно выбрать способ описания выводов формируемого символа с помощью группы кнопок **Pin Name Source**, расположенных в диалоговой панели. При нажатии кнопки **Using**

Schematic информация о выводах нового УГО извлекается автоматически из схемы, название которой указано в соответствующем поле редактирования. Для определения формы символа предназначена группа кнопок **Shape**.

После выбора нужных опций необходимо нажать **Next**, в результате чего открывается следующее окно, вид которого представлен на рисунке 26.

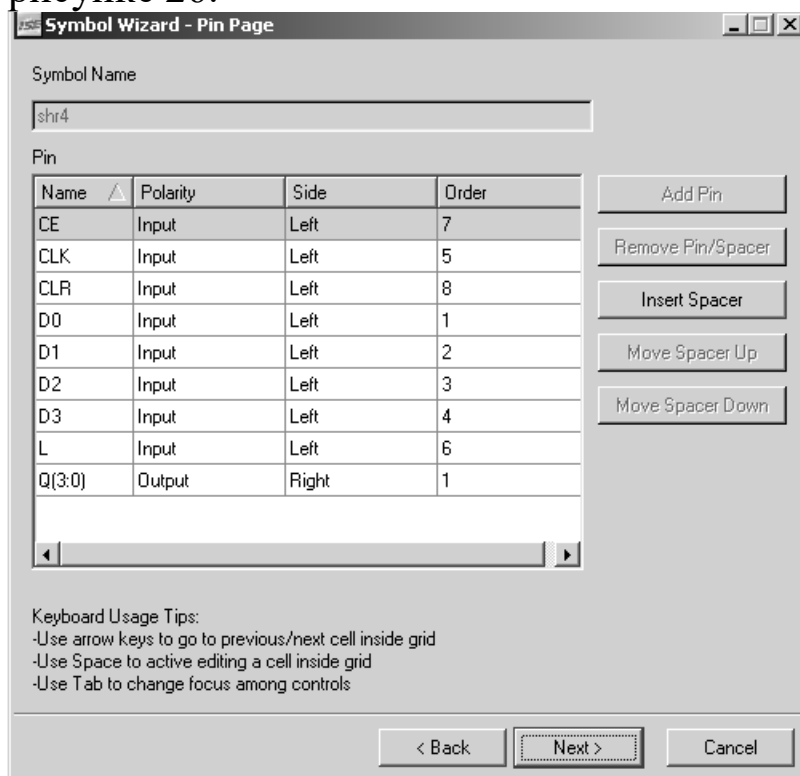


Рисунок 26 – Диалоговое окно ввода и редактирования исходных параметров символа

Это окно содержит поле **Symbol Name**, в котором указывается название формируемого символа, и таблицу описания выводов (контактов) **Pin**.

Таблица описания выводов **Pin** содержит четыре колонки. В первой указывается обозначение вывода Name, во второй - его тип Polarity (вход, выход, двунаправленный), в третьей - сторона символа, с которой он расположен, Side (левая или правая), в четвертой - порядковый номер Order.

В данной таблице только можно изменить расположение вывода Side и его порядковый номер Order. Выполнив все необходимые изменения, следует нажать кнопку **Next**, расположенную в нижней части, после чего открывается очередная диалоговое окно «мастера»,

которое приведено на рисунке 27. В этом окне содержатся основные размеры формируемого символа.

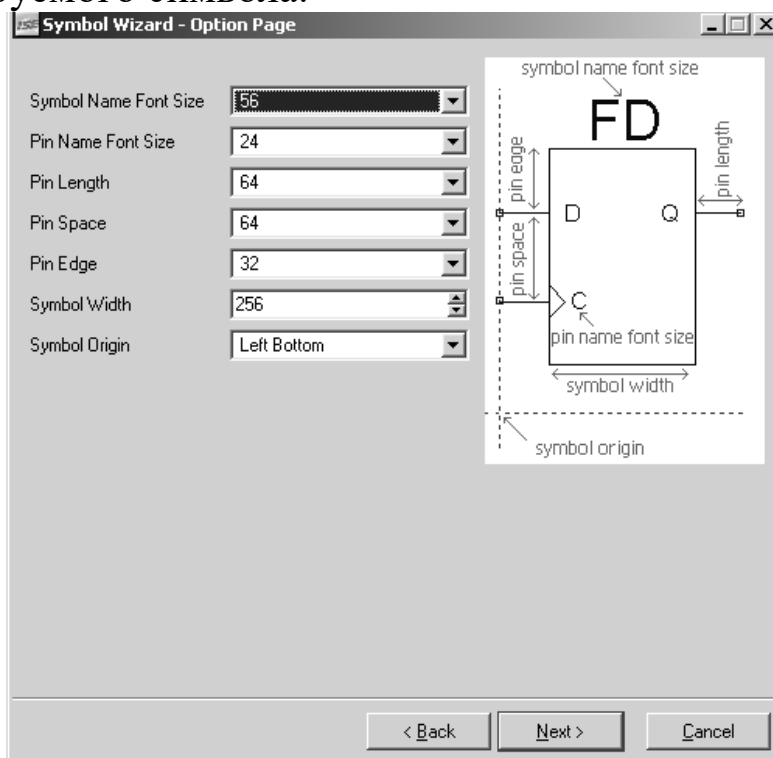


Рисунок 27 – Окно установки геометрических размеров символа

После определения всех необходимых размеров символа необходимо нажать **Next**, в результате чего открывается окно, в котором отображается изображение созданного символа (см. рис. 28).

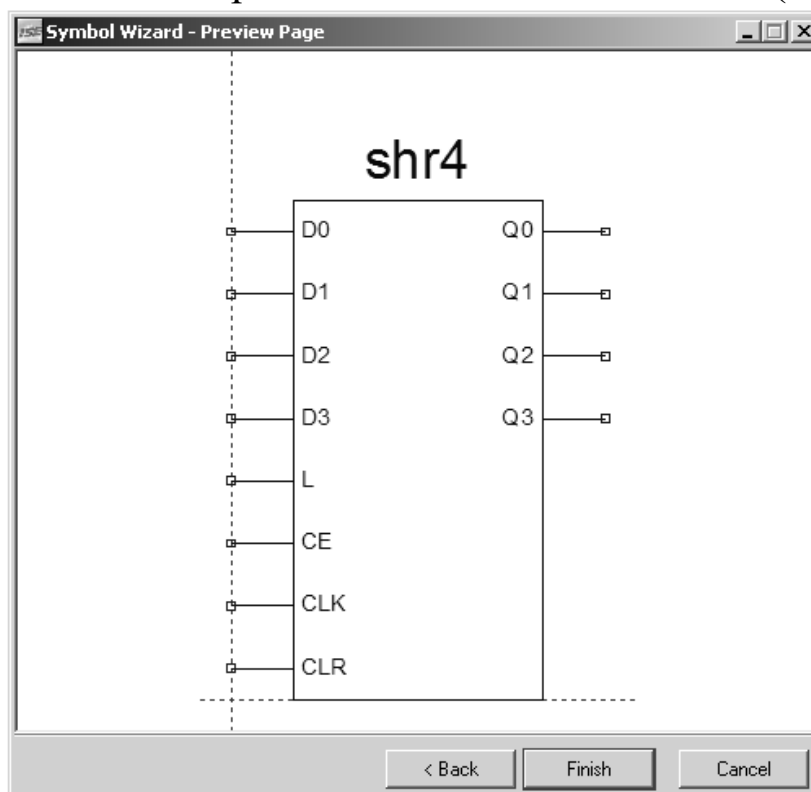


Рисунок 28 – Окно предварительного просмотра

Аналогично создадим символы для модулей mux4_2_1.sch, shr4.sch, shr8.sch, rg.sch и control_unit.sch. Далее полученные символы включаем в модуль верхнего уровня, т.е. в mult4.sch.

Окно иерархии проекта будет выглядеть, как показано на рисунке 29.

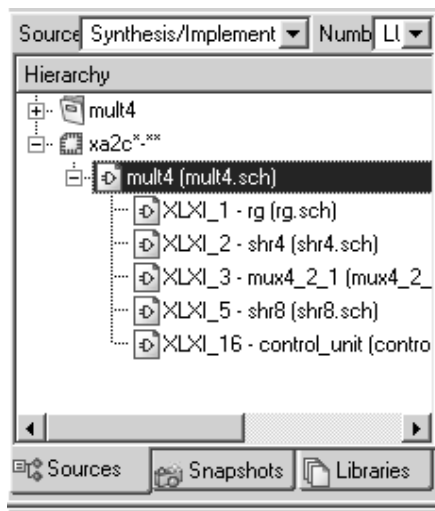


Рисунок 29 – Иерархия проекта mult4

Выполним соединение всех функциональных блоков умножителя в соответствии с алгоритмом функционирования. Подключение входов и выходов разрабатываемого устройства к выводам кристалла осуществляется через входные (IBUF) и выходные (OBUF) буферные элементы соответственно. Если эти элементы не были установлены в схеме, то необходимо при синтезе указать их автоматическое подключение.

Принципиальная схема 4-разрядного устройства умножения представлена на рисунке 30.

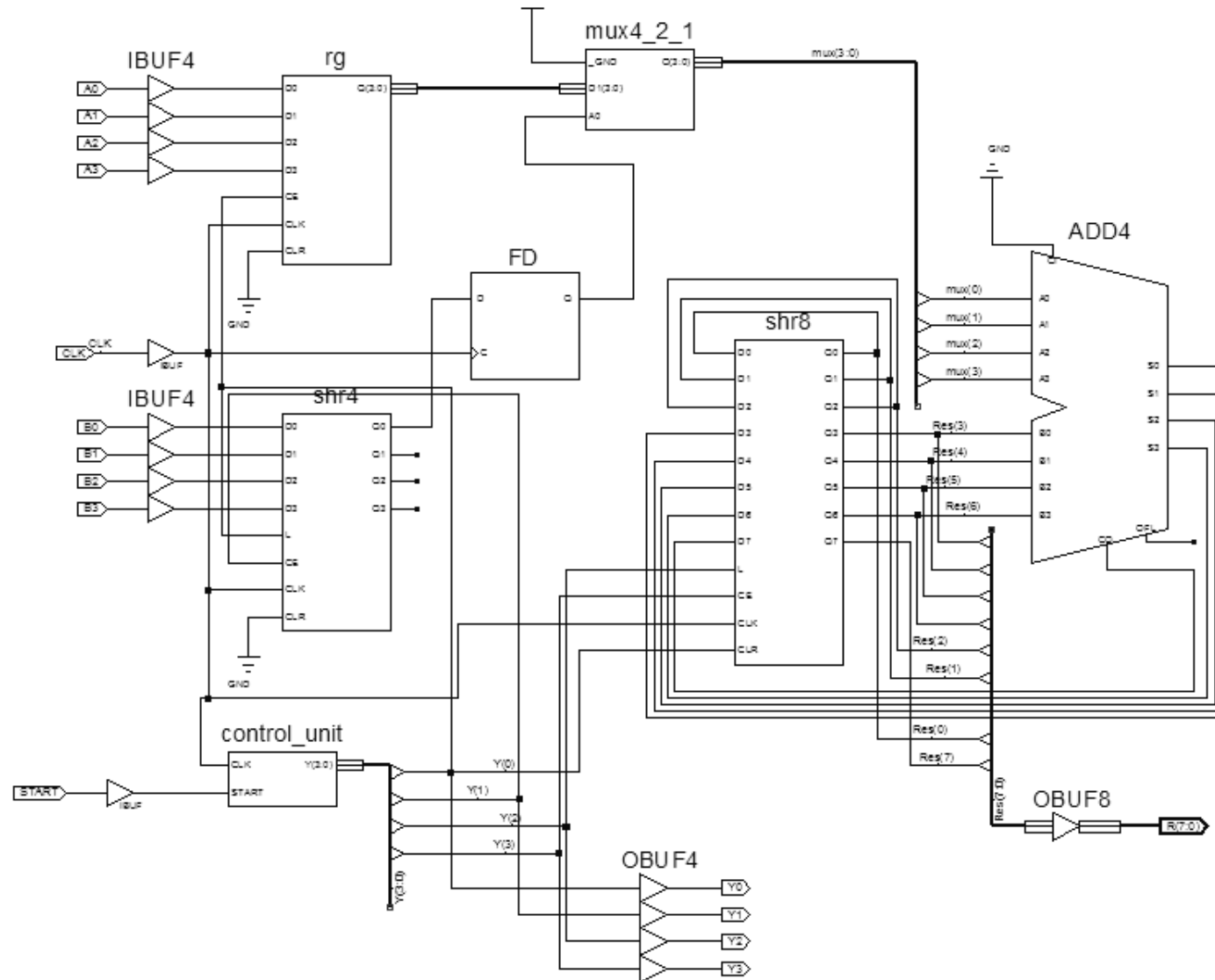


Рисунок 30 – Принципиальная схема устройства умножения на ПЛИС

Создание тестового модуля проекта

Для проверки правильности работы устройства необходимо провести функциональное моделирование. В САПР Xilinx ISE тестовый модуль можно создать двумя способами. Первой способ – это создание текстового описания тестовых воздействий на языке VHDL, второй – создание тестового модуля виде временных сигналов в графическом виде.

В данных методических указаниях рассмотрим только второй способ, так как он является более наглядным.

Для создания тестового модуля в виде временных диаграмм необходимо выполнить команду **Project**→**New Source** и в качестве нового модуля выбрать **Test Bench Waveform**, как показано на рисунке 31.

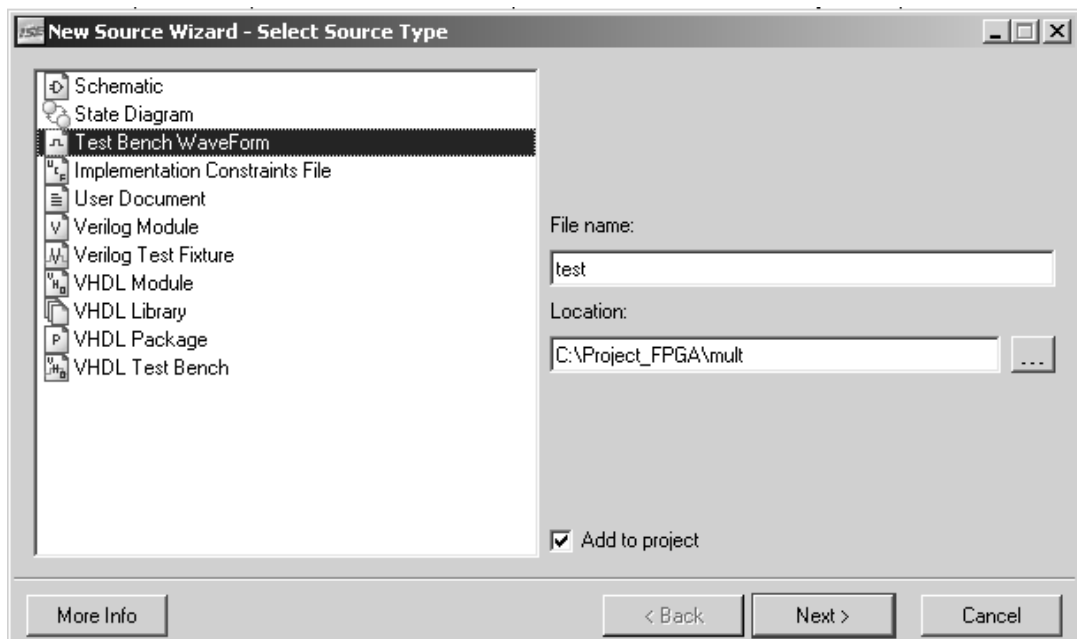


Рисунок 31 – Диалоговое окно создания тестового модуля

Нажав Next, на экране отобразиться диалоговое окно выбора объекта, для которого создается тестовый модуль, как показано на рисунке 32.



Рисунок 32 – Диалоговое окно выбора объекта

Выбор объекта завершается нажатием кнопки Next. Далее на экране появиться окно с параметрами созданного проекта, а затем диалоговое окно инициализации временных параметров, показанное на рисунке 33.

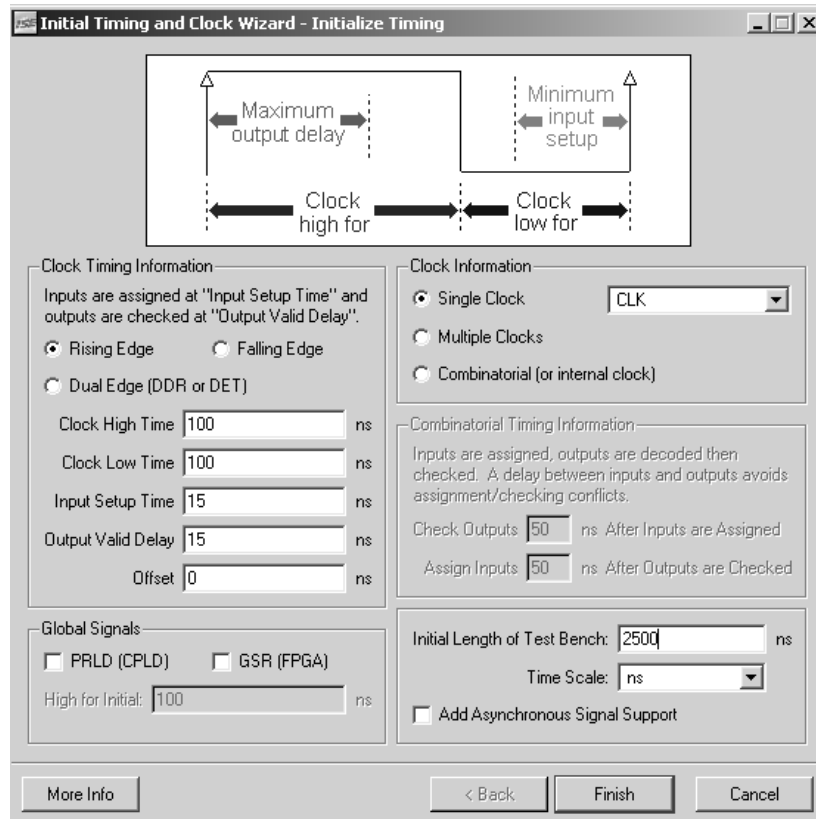


Рисунок 33 – Диалоговое окно инициализации временных параметров

Временные параметры, необходимые для формирования тестовых сигналов, различаются для синхронных и асинхронных устройств. Программа HDL Bencher автоматически определяет тип устройства, для которого разрабатываются тестовые последовательности. Для синхронного устройства необходимо, прежде всего, проконтролировать и при необходимости изменить выбор основного тактового сигнала, в поле **Single clock**, используя выпадающий список интерфейсных сигналов устройства. Затем следует выбрать единицы измерения временных параметров, используя выпадающий список возможных значений поля **Time scale**. Далее нужно определить длительность состояний высокого и низкого логических уровней периода тактового сигнала в поле редактирования **Clock high time** и **Clock low time** соответственно. Кроме того, в поле редактирования **Input setup time** следует указать минимальное значение времени установки входных сигналов, а в поле **Output valid delay** - максимальное значение задержки выходных сигналов.

Программа HDL Bencher автоматически определяет, чем тактируется устройство - фронтом (**Rising edge**), спадом (**Falling edge**) тактового сигнала или тем и другим (**Dual edge**), но разработчик может выбрать это вручную, используя соответствующие кнопки диалоговой панели (рис. 33). При создании тестов для комбинационных устройств нужно указать только значения временных интервалов между моментами подачи входных воздействий и контроля выходных сигналов **Check outputs** и **Assign Inputs**. Эти параметры позволяют исключить возможные конфликты в моменты изменения входных и выходных сигналов.

Процесс установки требуемых параметров заканчивается нажатием кнопки **Finish**, после чего в рабочей области окна программы HDL Bencher отображаются заготовки временных диаграмм (см. рис. 34).

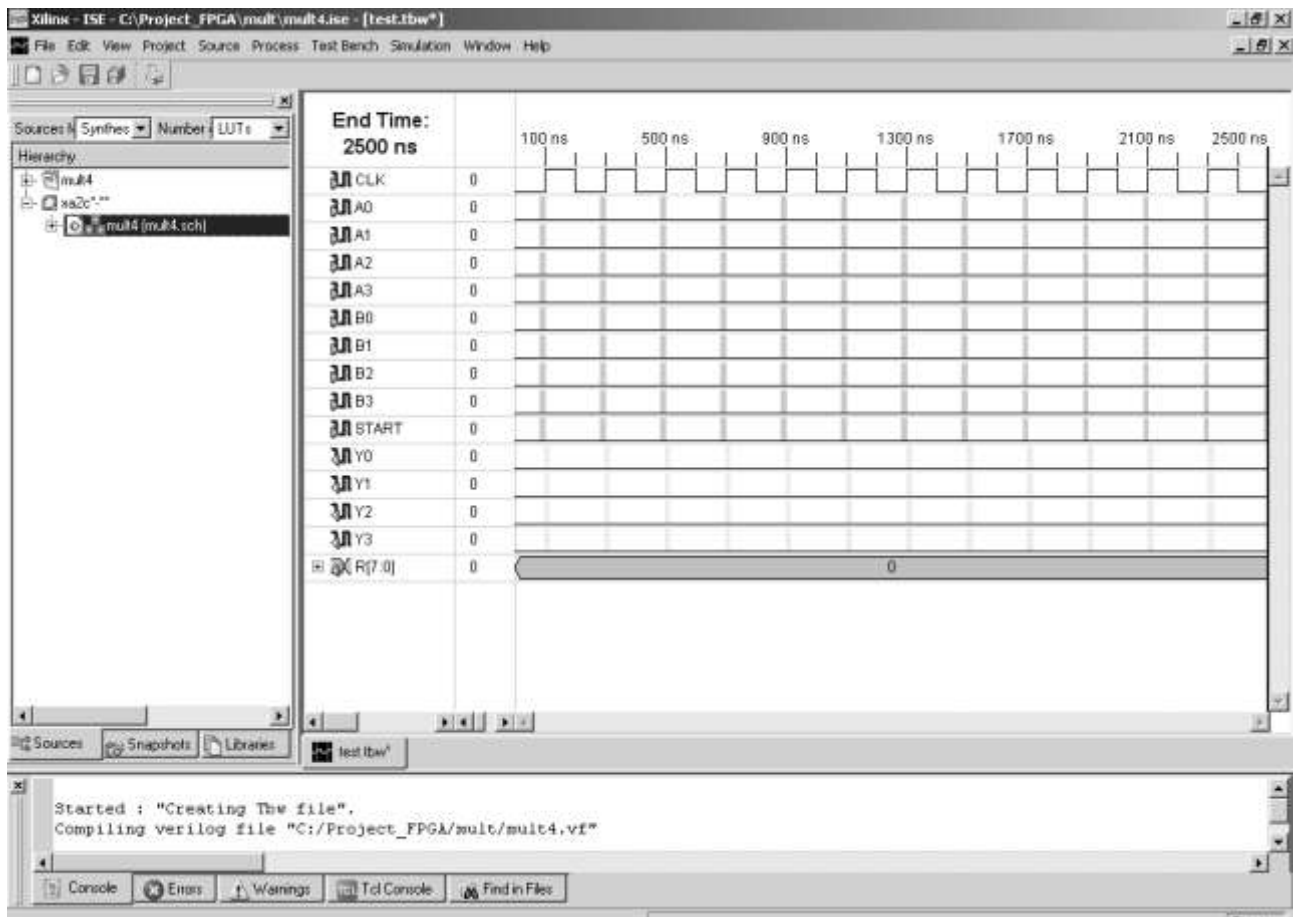


Рисунок 34 – Основное окно программы генерации тестов HDL Benchers

Зададим тестовые значения сигналам как показано на рисунке 35 и выполним моделирование, нажав в окне процессов **Xilinx ISE Simulator** → **Generate Expected Simulation Results**.

Результат выполнения моделирования показан на рисунке 35.

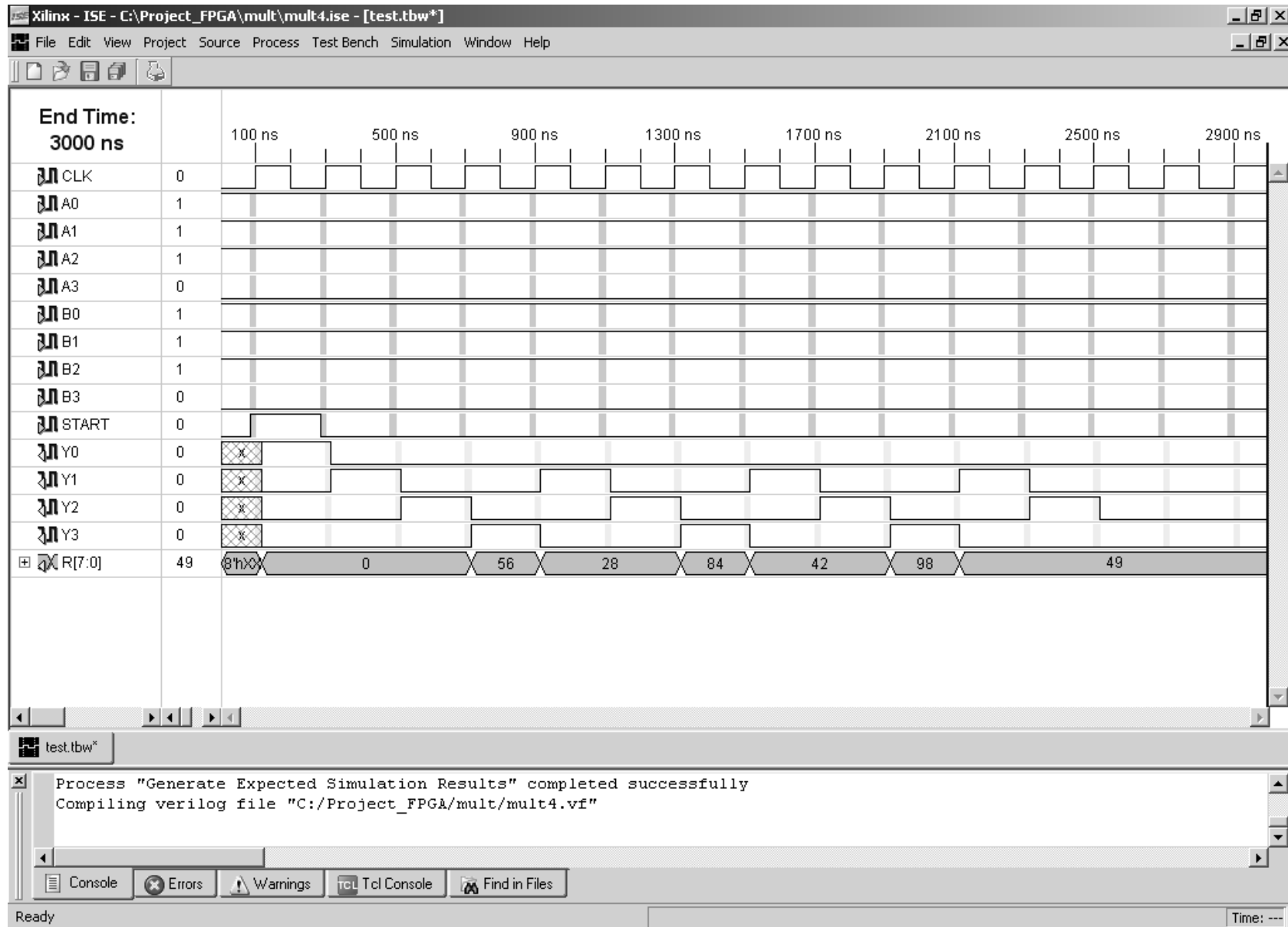


Рисунок 35 – Результат моделирования работы устройства

В примере, показанном на рисунке 35 множимое $A=0111_2=7$, множитель $B=0111_2=7$, результат выполнения операции $R=A*B=00110001_2=49$.

5 Варианты заданий

1. Ввод двух шестнадцатеричных чисел посредством задания ключами соответствующих разрядов двух семисегментных индикаторов. Отображение введенного числа в десятичной системе счисления. (Например, для задания числа 17 необходимо установить разряды a,b,c индикатора, отвечающего за младший разряд числа, и разряды a,b,c индикатора, отвечающего за старший разряд, в единичное состояние).

2. Ввод двух одноразрядных чисел, отображение введенных чисел на индикаторе, выбор операции – сложение, вычитание, вывод результата выбранной операции над числами на индикатор.

3. Ввод двух одноразрядных чисел, отображение введенных чисел на индикаторе, вывод результата умножения чисел на индикатор.

4. Ввод двух целых двухразрядных чисел, отображение введенных чисел на индикаторе, вывод результата деления первого числа на второе на индикатор. Результат округлять до меньшего целого.

5. Ввод двух одноразрядных чисел, отображение введенных чисел на индикаторе, вывод результата деления первого числа на второе на индикатор. Результат округлять до большего целого.

6. Ввод двух одноразрядных чисел, отображение введенных чисел на индикаторе, вывод остатка от деления первого числа на второе на индикатор.

7. Ввод двух двухразрядных чисел m_1, m_2 . Последовательный вывод на индикатор чисел от m_1 до m_2 или от m_2 до m_1 (в зависимости от выбранного режима) с частотой 2 Гц.

8. Генерацию прямоугольных колебаний с различной скважностью, задаваемой пользователем.

9. Генерацию прямоугольных колебаний с различной частотой, задаваемой пользователем.

10. Ввод числа n , последовательный ввод n чисел, сортировку чисел по возрастанию, вывод чисел на индикатор.

11. Ввод числа n , последовательный ввод n чисел, сортировку чисел по убыванию, вывод чисел на индикатор.

12. Генерацию последовательности прямоугольных импульсов различной формы, задаваемой пользователем и отображаемой на индикаторе. Длина последовательности – не более 7.

13. Распознавание во входном сигнале заданной последовательности логических нулей и единиц. Длина последовательности - не более 7.

14. Определение количества нулей и единиц, поступивших на вход устройства, за период времени с предварительно заданной в секундах длительностью, началом которого является момент нажатия на ключ. Для ввода последовательности использовать два ключа, нажатие одного из которых соответствует поступлению на вход «единицы», а второго – «нуля».

15. Ввод десятичного двухразрядного числа, преобразование числа в зависимости от выбора пользователя в двоичную, восьмеричную, шестнадцатеричную системы счисления.

16. Определение частоты входного сигнала и отображение ее на индикаторе с периодом обновления 1 с.

17. Определение временного интервала между двумя нажатиями ключа.

18. Модель светофора.

19. Последовательное считывание трех байт из ОЗУ с 8 битным адресным входом и 8 битным входом/выходом данных, определение среднего арифметического трех считанных значений, запись полученного значения среднего арифметического в ОЗУ по адресу второго считанного байта. Емкость ОЗУ – не более 1024 байт.

6 Контрольные вопросы

7 Список литературы

1. Новожилов, Олег Петрович. Электротехника и электроника [Текст] : учебник для бакалавров / О. П. Новожилов ; МГИУ. - 2-е изд., испр. и доп. - Москва : Юрайт, 2016. - 653 с.

2. Дьяков, И. А. Микропроцессорные системы. Архитектура микроконтроллеров семейства MCS-51 [Электронный ресурс] : учебное пособие / И. А. Дьяков. - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2014. - 79 с. - Режим доступа: biblioclub.ru