

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Московский физико-технический институт
(государственный университет)

Кафедра радиотехники

Лабораторная работа

**Проектирование цифровых устройств в среде
WebPack ISE**

Лабораторная работа по курсу:
Основы цифровой электроники

Составитель
А. Л. Ларин

Москва 2010

Составитель: А.Л. Ларин

Проектирование цифровых устройств в среде WebPack ISE:
Лабораторная работа по курсу: *Основы цифровой электроники* – М.:
МФТИ, 2010. – 59 с.

© Московский физико-технический институт
(государственный университет), 2010

Содержание

Введение	4
1. Семейство микросхем <i>SPARTAN-II</i>	5
2. Описание лабораторного макета	7
3. Работа с пакетом <i>WebPACK ISE</i>	10
3.1. Создание проекта	11
3.2. Создание принципиальной схемы в среде схемотехнического редактора	14
3.3. Ввод временных и топологических ограничений проекта	22
4. Формирование тестовых сигналов для проверки работы устройства путем моделирования	24
5. Моделирование работы устройства	27
6. Реализация проекта (Implementation)	28
7. Программирование ПЛИС с помощью модуля <i>iMPACT</i> пакета <i>WebPACK ISE</i>	28
8. Задание	34
Приложение	35
Краткий обзор схемного и символьного редакторов	35
Возможности схемотехнического редактора <i>WebPACK ISE</i>	35
Пользовательский интерфейс схемотехнического редактора	36
Схемный редактор	40
Редактор символов	40
Файлы схемного редактора и редактора символов	40
Входные файлы	41
Выходные файлы	41
Рекомендации для схемного проектирования	41
<i>Schematics</i> (схемы)	43
<i>Symbols</i> (символа) и <i>Instances</i> (объекты)	44
<i>Nets</i> (цепи) и <i>Buses</i> (шины)	45
Шины	45
IO Маркеры (Маркеры Ввода/Вывода)	46
<i>Attributes</i> (признаки)	47
Рабочая область редакторов схем и символов	47
Вкладка <i>Symbols</i> (символов)	48
<i>Options Tab</i> (вкладка опций)	49
Панель инструментов схемного редактора	50
Панель инструментов редактора символов	51
Сообщения об ошибках и предупреждения выдаваемые программой проверки схемы (<i>DRC</i>)	53
Литература	59

Введение

Целью работы является ознакомление с интерфейсом и возможностями САПР (системы автоматического проектирования) *WebPack ISE 10.1*, приобретение практических навыков в создании и моделировании простейших комбинационных и последовательностных схем с использованием схемотехнического редактора.

Для создания цифрового устройства на базе ПЛУ (программируемые логические устройства) фирмы *Xilinx* необходимо выполнить следующую последовательность операций:

1. Создать новый проект, указав семейство, тип ПЛУ и средство синтеза.
2. Разработать описание проектируемого устройства в схемотехнической, алгоритмической или текстовой форме.
3. Выполнить синтез устройства.
4. Провести проверку проекта методом функционального моделирования.
5. Выполнить размещение (placement) и трассировку (routing) проекта в кристалле.
6. Провести окончательную верификацию проекта методом временного моделирования.
7. Загрузить конфигурационные данные проекта в кристалл (выполнить программирование ПЛУ).
8. Продемонстрировать работу созданного устройства преподавателю.

Операции функционального и временного моделирования не являются обязательными, но позволяют значительно сократить общее время разработки устройства за счет раннего обнаружения возможных ошибок. Ниже рассматривается выполнение необходимых операций проектирования при использовании схемотехнического описания проекта.

Схемотехнический редактор *Engineering Schematic Capture™ (ECS)* входит в состав САПР *WebPACK ISE*. При описании проектируемого устройства программные средства рассматриваемого пакета ориентированы, прежде всего, на применение языков высокого уровня *HDL (Hardware Description Language)*. Однако схемотехнический редактор *Engineering Schematic Capture™ (ECS)* также поддерживает описание устройств с помощью графического языка *Logic Editor*.

технический редактор целесообразно использовать в случае применения смешанного способа описания проектируемого устройства. При таком способе функциональные блоки проектируемого устройства, как правило, описываются на одном из языков *HDL*, а для верхнего уровня иерархии проекта, где производится их объединение, используется схемотехническая форма описания. Графические символы для представления функциональных блоков и схема верхнего уровня иерархии проекта создаются средствами схемотехнического редактора. Учитывая, что *WebPACK ISE* содержит обширные библиотеки компонентов для всех поддерживаемых пакетом семейств ПЛУ фирмы *Xilinx*, описание проектируемого устройства может быть полностью выполнено в виде схем, созданных в среде редактора. Несмотря на то, что представление устройства в виде схемы часто является более привычным для разработчиков, рекомендуется изучить и в дальнейшем использовать один из языков *VHDL* или *Verilog*. Эти языки поддерживаются большинством САПР, и поэтому созданный проект можно перенести из одной системы в другую.

1. Семейство микросхем SPARTAN-II

Программируемые логические устройства (ПЛУ) в настоящее время стали основным, очень гибким инструментом цифрового проектирования. Микросхемы типа *FPGA* были разработаны фирмой *Xilinx, Inc.* в 1998 году. Для иллюстрации возможностей ИС типа *FPGA* мы воспользуемся одним из популярных семейств этой фирмы – семейством *SPARTAN-II*. ПЛИС (программируемые логические интегральные схемы) семейства *SPARTAN-II* применяются в проектах как альтернатива специализированным интегральным схемам емкостью до 600 000 вентилях и системным быстродействием до 200 МГц. Напряжение питания, подаваемое на ядро кристалла семейства *SPARTAN-II*, составляет 2.5 В, а на блоки ввода-вывода подается напряжение 3.3 В.

Семейство *SPARTAN-II* состоит из 6 типов кристаллов, а семейство *SPARTAN-III* содержит 7 типов кристаллов, отличающихся логической емкостью. Сравнительные параметры некоторых ИС приведены в табл. 1.

Т а б л и ц а 1

Основные характеристики семейства *Spartan-II(E)*

Кристалл	Логические Ячейки (LC)	Системные вентили	Матрица конфигу- рируемых логиче- ских бло- ков <i>CLB</i>	CLB	Блочная ОЗУ, Бит	Пользова- тельские блоки ввода- вывода (IOB), max
XC2S15	432	15000	8 × 12	96	16384	86
XC2S30	972	30000	12 × 18	216	24576	132
....
XC2S600E	15552	600000	48 × 72	3 456	294912	514

Кристаллы семейства *SPARTAN-II* производятся на основе статического ОЗУ (*SRAM*), поэтому их функционирование определяется загружаемыми во внутренние ячейки памяти конфигурационными данными. Конфигурационные данные создаются пользователем при помощи программного обеспечения проектирования *ISE* или *WebPACK ISE*, разработанного фирмой *Xilinx*. Программное обеспечение включает в себя модули схемного и текстового ввода, моделирования, автоматического и ручного размещения и трассировки, создания, загрузки и верификации конфигурационных данных.

САПР для *SPARTAN-II* включает унифицированную библиотеку (*Unified library*) стандартных функций. Эта библиотека содержит свыше 400 примитивов и макросов от двухвходовых вентилей И до 16 битных аккумуляторов и включает арифметические функции, компараторы, счетчики, регистры данных, дешифраторы, шифраторы, устройства ввода-вывода, защелки, мультиплексоры и регистры сдвига.

Программное средство размещения и трассировки (*place-and-route, PAR*) обеспечивает автоматическое выполнение процесса размещения проекта в кристалл. Процедура разбиения на физические блоки получает исходную информацию о проекте в виде перечня связей и осуществляет привязку абстрактных логических элементов к реальным физическим ресурсам архитектуры *FPGA (IOB, CLB)*. Затем процедура размещения определяет для них наилучшее место на кристалле, руководствуясь информацией о

межсоединениях и желаемом быстродействии. Завершается процедура трассировки выполнением соединений между блоками.

Последним этапом создания устройства является конфигурирование кристалла. Конфигурирование – это процесс загрузки битовой последовательности, полученной с помощью программного обеспечения проектирования, во внутреннюю конфигурационную память кристаллов *FPGA*. *SPARTAN-II* может загружаться как побитно (ведущий/подчиненный последовательные режимы и *JTAG*), так и побайтно (подчиненный параллельный режим).

Конфигурационные данные при выключенном питании должны храниться во внешнем энергонезависимом устройстве статической памяти. Обычно для этого применяются микросхемы флэш-памяти *Xilinx* серии *XC1700* или *XC1800*. В лабораторном макете используется память *XC18V256*, в которую и производится запись конфигурационных данных.

2. Описание лабораторного макета

Лабораторная работа выполняется на макете (рис. 1), который содержит:

1. *FPGA XC2S15* фирмы *XILINX*;
2. флэш-память *XC18V256*;
3. 8-разрядный цифро-аналоговый преобразователь (ЦАП) *AD7302*;
4. кварцевый генератор с частотой колебаний 4 МГц;
5. разъем для подключения кабеля загрузки конфигурационных данных;
6. двухразрядный 7-сегментный индикатор;
7. 8 светодиодов;
8. два 20-контактных разъема для подключения внешних сигналов и наблюдения за выходными сигналами;
9. кнопку перезаписи конфигурационных данных из памяти в *FPGA*;
10. кнопку, назначение которой определяется пользователем;
11. набор 8 переключателей.

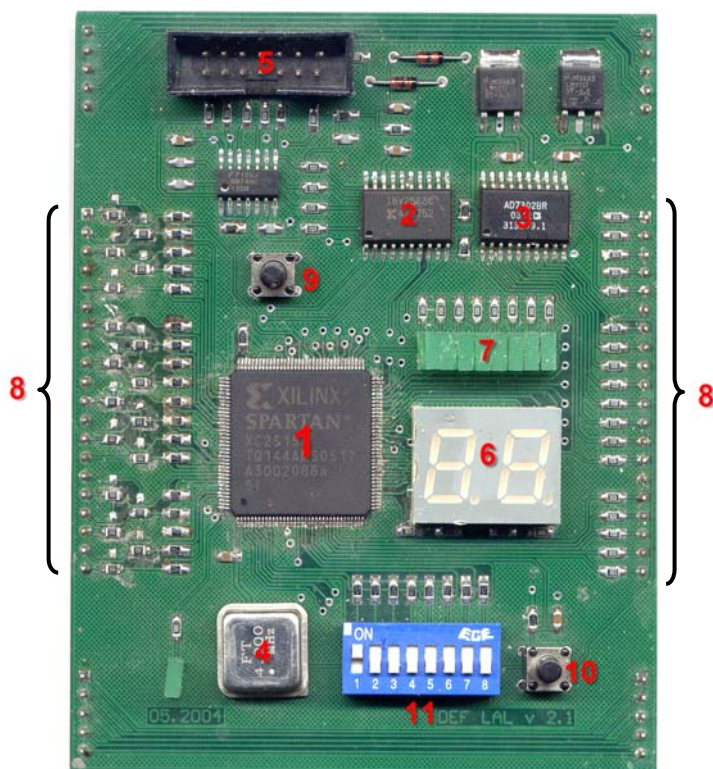


Рис. 1. Лабораторный макет

При назначении номеров выводов микросхемы XC2S15 следует руководствоваться табл. 2, в которой показано соответствие выводов микросхемы с номерами контактов разъемов 8, 7-сегментных индикаторов 6, светодиодов 7, кварцевого генератора 4, кнопки 10 и переключателей 11. Номера контактов левого 20-контактного разъема обозначены в таблице $L1 - L20$ (номера идут сверху вниз), а правого 20-контактного разъема – $R1 - R20$. Контакты с $L7$ по $L14$ на монтажной плате соединены с восемью переключателями, с помощью которых на эти контакты могут быть поданы напряжения, соответствующие логическим нулям или логическим единицам.

Таблица 2

Обознач. контакта разъема	Вывод FPGA	Обознач. контакта разъема	Вывод FPGA	Переключатели 11	Вывод FPGA	7-сегм. индикатор	Вывод FPGA
L1	p91	R1	p94	1	p3	Старший разряд	
L2	p113	R2	p93	2	p4	a	p57
L3	p114	R3	p87	3	p5	b	p54
L4	p117	R4	p86	4	p6	c	p40
L5	p118	R5	Вых ЦАП	5	p7	d	p43
L6		R6		6	p10	e	p41
L7	p120	R7	p74	7	p19	f	p56
L8	p121	R8	p67	8	p20	g	p58
L9	p123	R9	p66	Кв.генер	p18	Младший разряд	
L10	p130	R10	p65	Кнопка	p22	a	p49
L11	p131	R11	p63	Светодиоды		b	p51
L12	p132	R12	p62	7	p75	c	p47
L13	p133	R13	p60	6	p76	d	p44
L14	p134	R14	P59	5	p77	e	p46
L15		R15		4	p79	f	p50
L16	p136	R16	p29	3	p80	g	p48
L17	p137	R17	p28	2	p83		
L18	p15	R18	p27	1	p84		
L19	p140	R19	p26	0	p85		
L20	p141	R20	p23				

В верхней и нижней части макета с каждой стороны расположены по 5 контактов. Каждый контакт с левой стороны соединен с соответствующим контактом, расположенным справа. Это позволяет избежать использования длинных проводников при необходимости соединения контактов L1 – L20 с контактами R1 – R20.

Обозначение 7-сегментных индикаторов приведено на рис. 2. В лабораторном макете использованы индикаторы с общим анодом, то есть для того, чтобы светился нужный сегмент, на него надо подать напряжение, соответствующее НИЗКОМУ логическому уровню. Для свечения светодиодов, обозначенных на рис.1 цифрой 7, на их входы необходимо подавать сигнал ВЫСОКОГО уровня. Кнопка 10 в исходном состоянии обеспечивает сигнал высокого уровня.

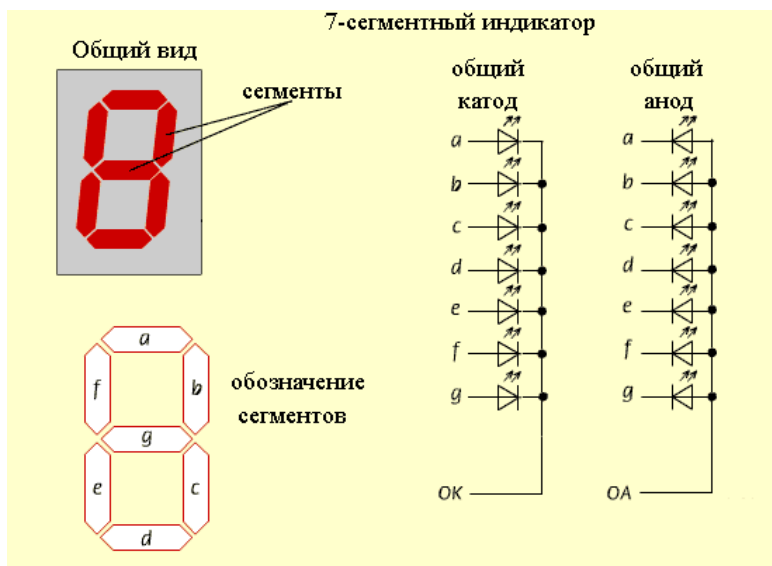



Рис. 2. Общий вид и обозначение сегментов индикатора

Входы цифро-аналогового преобразователя 3 соединены с контактами правого разъема 8. Младший разряд соединен с контактом *R7*, следующий – с контактом *R8* и т. д. Старший разряд ЦАП соединен с контактом *R14*.

3. Работа с пакетом *WebPACK ISE*

Перед началом работы необходимо прочитать Приложение, где даны определения терминов и советы по работе с пакетом *WebPACK ISE*. Затем убедиться, что на диске *C* имеется директория *STUDENTS*.

Для запуска пакета *WebPACK ISE* следует дважды щелкнуть

левой клавишей мыши на пиктограмме , расположенной на рабочем столе ПК. На экране появится окно, показанное на рис. 3. В том случае, если последним пользователем при выходе из программы *WebPACK ISE* проект не был закрыт, то вид окна будет отличаться от приведенного ниже. В этом случае надо выбрать пункт меню *File* \Rightarrow *Close Project*.

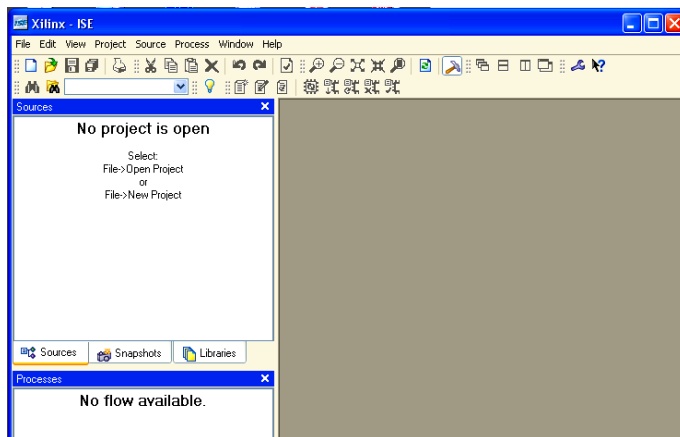


Рис. 3. Вид окна при активизации пакета WebPACK ISE

3.1. Создание проекта

В открывшейся диалоговой панели нужно определить исходные данные, необходимые для создания проекта:

- название проекта;
- диск и директория, в котором предполагается расположить проект;
- семейство ПЛУ, на базе которого разрабатывается устройство;
- тип кристалла;
- тип корпуса;
- быстроедействие;
- средство синтеза.

Для задания значений указанных параметров необходимо:

1. Выбрать пункт меню *File* \Rightarrow *New Project* для создания нового проекта.
2. В появившемся окне (рис. 4) в поле *Project Name* дать имя создаваемому проекту, например, *gr611_IVANOV*, а в поле *Project Location* выбрать место на диске (латинским шрифтом). Имя проекта автоматически определяет название рабочего каталога проекта.
3. В следующем поле *Top-Level Source Type* имеется

возможность выбрать способ описания создаваемого устройства (тип исходного модуля) на одном из языков описания схем *HDL* или путем рисования электрических схем. В настоящей лабораторной работе предлагается воспользоваться схемотехническим редактором (*Schematic*).

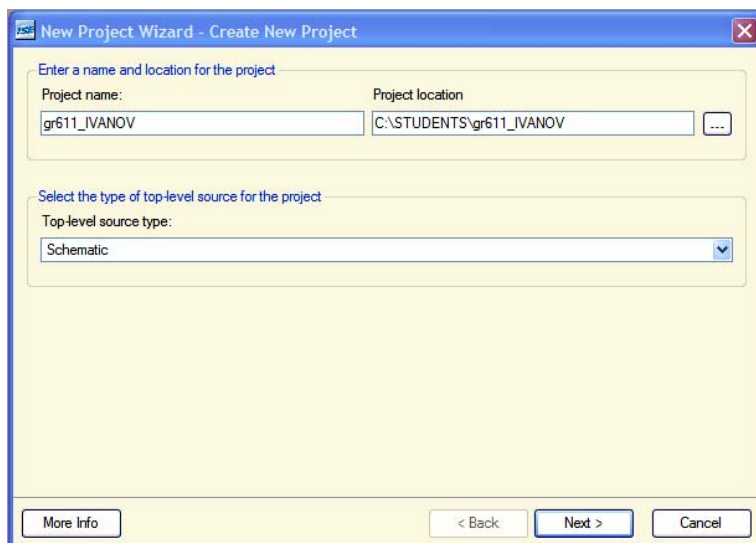


Рис. 4. Окно выбора имени проекта, места расположения на диске и типа исходного модуля

После выполнения команды *Next* появляется окно (рис. 5), в котором выбирается тип устройства. На лабораторном макете установлена микросхема *FPGA* семейства *SPARTAN-II* – *XC2S15* в корпусе *TQ144*. В качестве симулятора выбирается *ISE Simulator (VHDL/Verilog)*, как показано на рис. 5.

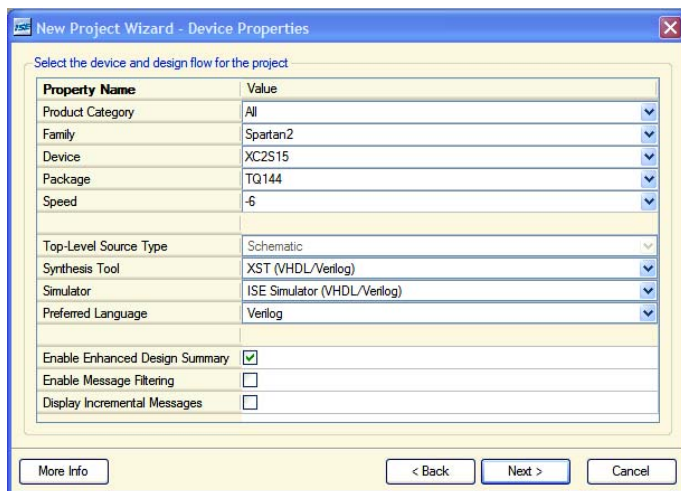


Рис.5. Выбор типа ПЛУ и средств синтеза для создаваемого проекта

Выполнение команды *Next* приводит к появлению окна, показанного на рис. 6, в котором начинается создание модуля исходного описания устройства.

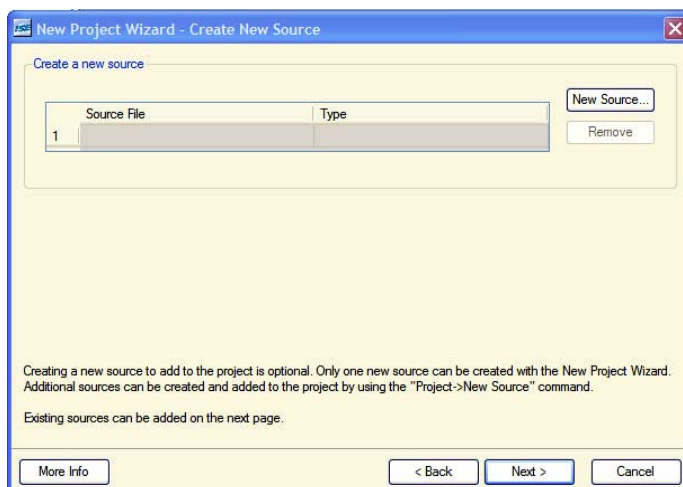


Рис. 6. Окно создания модуля исходного описания проектируемого устройства

3.2. Создание принципиальной схемы в среде схемотехнического редактора

В процессе создания принципиальной схемы проектируемого устройства выполняются следующие операции:

- ввод символов элементов схемы;
- соединение компонентов схемы с помощью проводников и шин;
- ввод названий цепей и шин;
- установка маркеров цепей, используемых для подключения "внешних" элементов или выводов ПЛУ.

Щелкнув по пункту проекта *New Source* (добавить новый источник), открываем новое окно (рис. 7), в котором выбираем вариант описания устройства *Schematic* и даем имя файлу.

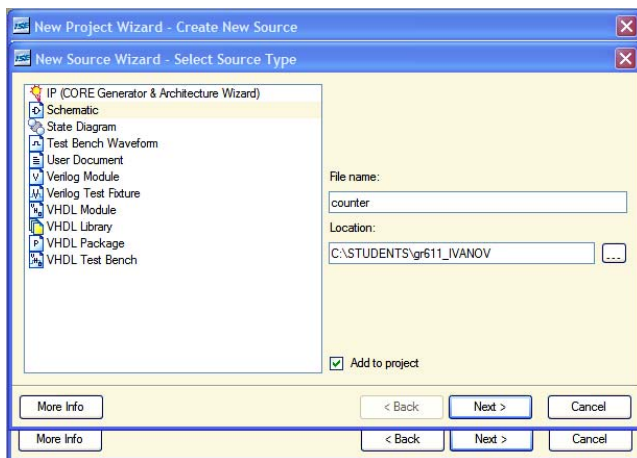


Рис. 7. Диалоговая панель установки параметров для нового
исходного модуля проекта

После этого, выполняя команды *Next* и *Finish*, подтверждаем создание директории (в нашем случае *gr611_IVANOV*). В открывающихся затем окнах последовательно выполняем команды *Next* и *Finish*.

При открытии основного окна схемотехнического

редактора (рис. 8) активизирован режим выбора объекта, установленный по умолчанию. В этом режиме осуществляется выделение, перемещение и удаление элементов схемы, а также просмотр и редактирование их параметров.

Открытое окно содержит поле чертежа (рабочая область) для создания схемы требуемого устройства.

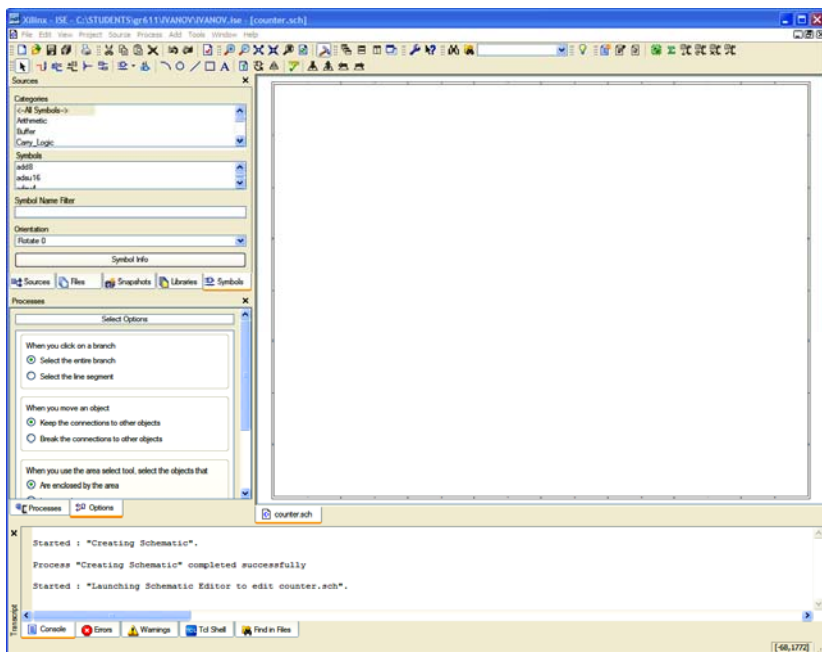


Рис. 8. Основное окно редактора проекта пакета *WebPACK ISE*

Теперь можно приступить к составлению схемы. Предположим, что заданным устройством является двоичный 8-разрядный счетчик с индикацией его содержимого с помощью светодиодов. Предусмотрим возможность выбора входных сигналов: от кварцевого генератора или от кнопки. Во вкладке *Categories* (Категории) выбираем раздел *Counter*, а затем во вкладке *Symbols* (Символы) – *cb8ce* (8-разрядный двоичный счетчик с входом разрешения счета *CE* и асинхронным сбросом *CLR*), и мышью переносим его изображение на поле чертежа

(рис. 9). Информацию о любом элементе, помещенном в поле чертежа, можно получить, наведя на него курсор и нажав на правую клавишу мыши. Во всплывающем меню выбрать *Symbol* \Rightarrow *Symbol Info*.

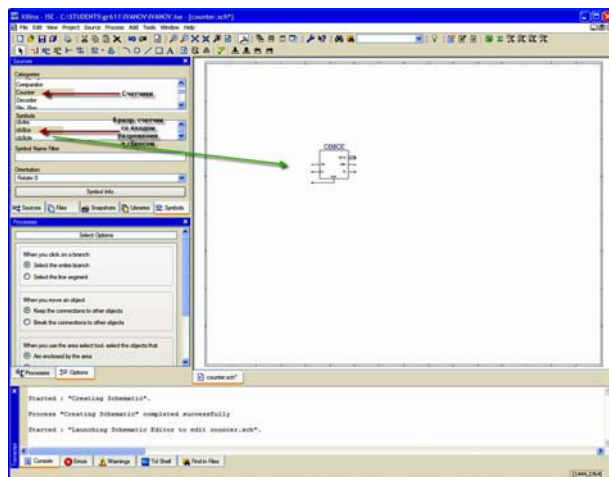




Рис. 9. Выбор элемента и перенос его в поле чертежа

Выходы счетчика, состояния которых необходимо индизировать с помощью светодиодов, изображены в виде 8-разрядной шины $Q(7:0)$. Подключать нагрузку непосредственно к выходам логического элемента нельзя, для этой цели служат выходные буферы. Первым шагом выбираем в категории *IO* символ *obuf* (буфер вывода) и переносим на схему этот элемент 8 раз (рис. 10). Затем в инструментальной панели выбираем команду  *Add IO Marker* (добавить маркер ввода/вывода) и к выходу каждого буфера подключаем маркер (2 шаг). Следующим шагом необходимо присвоить имена цепям, идущим от буферов, и соответствующим им маркерам. Для этого на инструментальной панели выбираем команду  *Add Net Name* и в открывшемся окне *Add Net Name Options* панели *Processes* записываем имя цепи (Рис. 11), в нашем примере это *D7*. Переносим маркер мыши на цепь, которой хотим дать имя, и нажимаем на левую клавишу мыши. Чтобы не набирать каждый раз новое имя, если оно

оканчивается цифрой, можно воспользоваться опцией *Increase the name* или *Decrease the name*. В этом случае при наведении маркера на новую цепь цифра будет увеличена на 1 или уменьшена на 1.

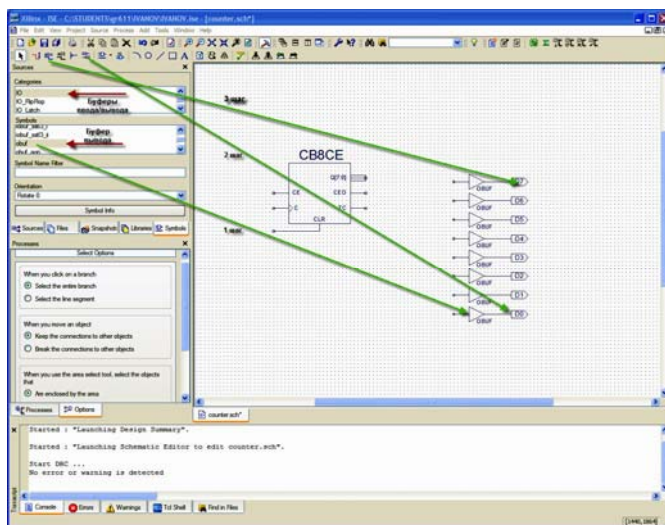


Рис. 10. Добавление выходных IO-маркеров и имен цепей

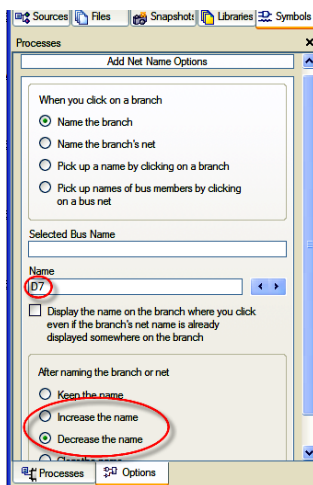




Рис. 11. Окно присвоения имен цепям и шинам

Теперь необходимо соединить выходы счетчика, представленные в виде шины, с входами буферов. Эта операция начинается с рисования шины. Выбираем пиктограмму  *Add Wire* и рисуем шину (1 шаг на рис. 12). Затем выбираем пиктограмму  *Add Bus Tap* и переносим в поле чертежа восемь отводов шины (2-й шаг на рис. 12). Следующей операцией является присвоение имени шине $Q(7:0)$ и цепям на выходах отводов шины $Q(7) \div Q(0)$. Присвоение осуществляется так же, как и в предыдущем случае. На сформированных выходах $D7 \div D0$ можно будет наблюдать, воспользовавшись осциллографом, изменение состояния каждого разряда нашего счетчика. Кроме того, к этим выходам подключены входы 8-разрядного цифроаналогового преобразователя, на выходе которого можно наблюдать изменение напряжения пропорционального числу, находящемуся в счетчике.

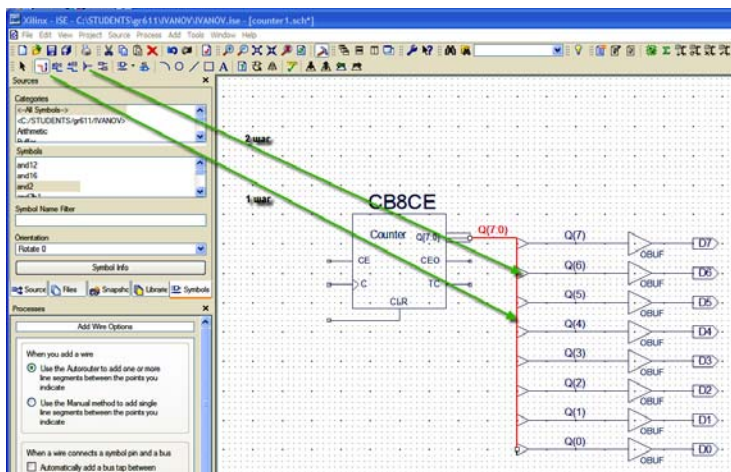


Рис. 12. Добавление отводов шины

Аналогично в схему вводятся шинные отводы и *IO*-маркеры для подключения светодиодов, индицирующих состояние триггеров в счетчике (рис. 13). Поскольку схема уже содержит цепи с именами $Q(7) \div Q(0)$, присвоение тех же имен новым цепям

выполняется следующим образом. Помещаем курсор на цепь, которой хотим присвоить имя, например $Q(7)$, и активизируем ее. В оперативной панели управления во всплывающем меню *Edit* выбираем опцию *Rename* и в ней команду *Rename Selected Net*. В появившемся окне (рис. 14) записываем $Q(7)$. Затем переходим к следующей цепи и так до $Q(0)$. Выходным цепям присвоены имена $LED7 \div LED0$.

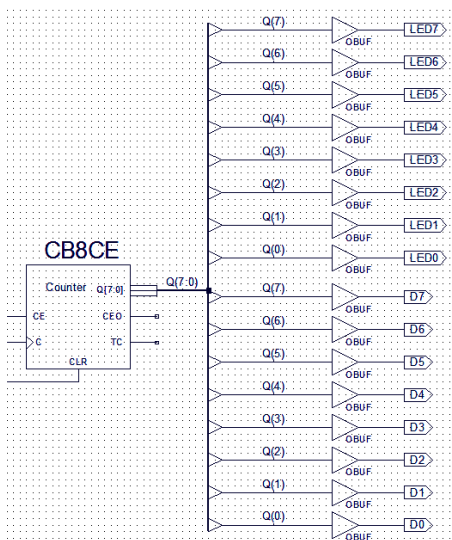


Рис. 13. Подключения выходных I/O-маркеров для светодиодов

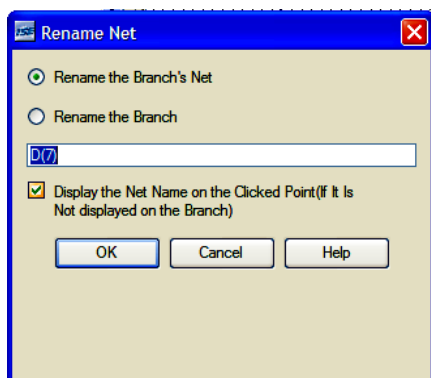


Рис. 14. Изменение имени цепи

Переходим к созданию входных цепей (рис. 15). На вход С счетчика сигналы поступают с выхода мультимплексора, с помощью которого осуществляется выбор источника сигнала. Во вкладке *Categories* выбираем раздел *Mux* (мультимплексоры), а во вкладке *Symbols* выбираем элемент *muxf5* (2-входовой мультимплексор) и переносим его в поле чертежа.

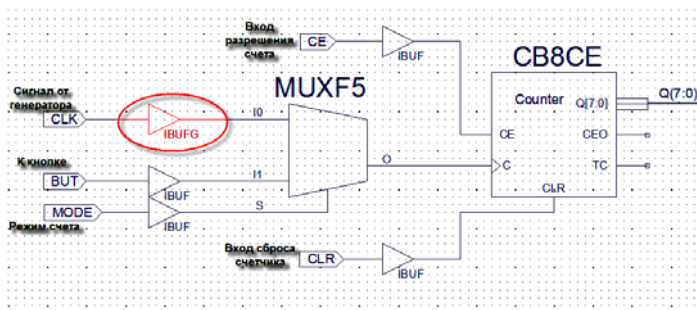



Рис. 15. Организация входных цепей счетчика

Внешние сигналы подаются на входы элементов схемы через входные буферы. Для введения их в схему выбираем в категории *IO* символ *ibuf* (входной буфер) и подключаем этот элемент к каждому входу мультимплексора и счетчика за исключением входа мультимплексора *IO*, на который подается сигнал от кварцевого генератора. Если сигнал поступает на вывод *GCLKIOB* (глобальный *IO*-буфер тактового сигнала) микросхемы, а именно на такой вход подан сигнал кварцевого генератора, то входной буфер должен быть *ibufg*, на что обращено внимание на рис. 15. Затем к входам буферов добавляем *IO*-маркеры и присваиваем имена цепям, связанным с входными буферами. Внутренним цепям, например, цепи соединяющей выход мультимплексора *O* со счетным входом счетчика *C*, имена можно не присваивать. По умолчанию они имеют вид *XLXN_#*.


Для проверки созданной схемы следует воспользоваться командой *Check Schematic*, которая располагается в всплывающем меню *Tools*, или кнопкой  на инструментальной панели схемотехнического редактора. В процессе верификации осуществляется контроль целостности схемы и выполнения правил электрических


соединений. После выполнения проверки в окне состояния появляется отчет, в котором отображаются сообщения о возможных ошибках и предупреждения с указанием цепи или компонента, с которым они связаны. Отчет результата проверки правильно составленной схемы выглядит так:

Start DRC ...

No error or warning is detected



В случае появления сообщения об ошибках, необходимо обратиться к Приложению на страницах с 50 по 55 и устранить указанные ошибки.

Заключительным шагом в процессе разработки схемы является ее сохранение в виде файла на диске. Для этого следует использовать команду *Save* из всплывающего меню *File* или кнопку , расположенную на оперативной панели управления.

В процессе создания схемы часто используются операции удаления, перемещения и копирования элементов схемы. Эти процедуры выполняются в режиме выбора объекта схмотехнического редактора, который автоматически активизируется при отмене большинства операций или включается при нажатии кнопки  на панели инструментов.

Выполнение операций редактирования начинается с указания соответствующего объекта. Для выделения элемента схемы нужно поместить курсор на его изображение и щелкнуть левой кнопкой мыши. Для отмены выделения следует щелкнуть левой клавишей мыши на свободном поле чертежа.

Чтобы удалить выбранный элемент или фрагмент схемы, необходимо нажать клавишу *Del* на клавиатуре или выбрать пункт *Edit* в основном меню, а затем в соответствующем всплывающем меню – строку *Delete*.

Для копирования выделенного элемента или фрагмента схемы в буфер обмена нужно нажать кнопку , расположенную на оперативной панели, или выполнить команду *Copy* из всплывающего меню *Edit*. Чтобы вставить копию содержимого буфера обмена на поле чертежа, следует воспользоваться кнопкой  на оперативной панели или выбрать команду *Paste* из всплывающего меню *Edit*, после чего к курсору привязывается копируемое изображение.

Далее необходимо указать курсором место расположения полученной копии на поле чертежа и зафиксировать ее левой кнопкой мыши.

Для перемещения выбранной части схемы на поле чертежа поместить курсор на изображение одного из выделенных элементов, нажать левую кнопку мыши и, передвигая мышь, выбрать новое положение перемещаемой части схемы. Новое положение фрагмента фиксируется при отпускании левой кнопки мыши.

3.3. Ввод временных и топологических ограничений проекта

Для ввода дополнительной информации, используемой программами синтеза, размещения и трассировки, используют файл *User Constraints File (UCF)*. При этом сохраняется универсальность модулей исходного описания, которые могут использоваться в других проектах.

Чтобы приступить к созданию *UCF*-файла, необходимо в окне *Sources* (рис. 15а) активизировать вставку *Source* (1 шаг), щелчком левой клавиши мыши выделить строку с названием модуля верхнего уровня иерархии (2 шаг), в окне *Processes* активизировать вставку *Processes* (3 шаг), после чего в окне процессов, щелкнув мышью по значку **+** (4 шаг), развернуть *User Constraints*. Затем дважды щелкаем мышью по строке *Floorplan Area / IO / Logic - Post-Synthesis* (рис. 15б).

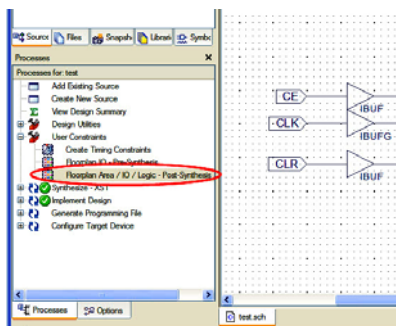
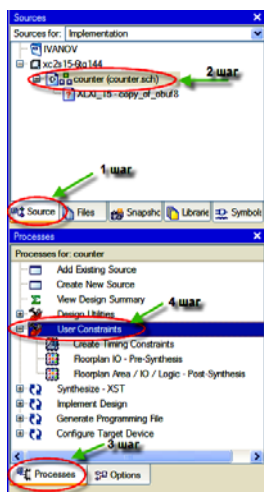


Рис. 15б. Создание *USF*-файла

Рис. 15а. Создание *UCF*-файла

В результате выполнения перечисленных действий на экране появляются три окна: *Design Browser*, *Design Object List* и *Device Architecture* (рис.16). В окне *Design Object List-I/O Pins* перечислены все входные и выходные порты проектируемого устройства с указанием направления сигнала. Третий столбец, обозначенный *Loc*, предназначен для привязки портов к выводам микросхемы. Обозначение выводов имеет вид *p#*. После заполнения всех строк в столбце *Loc*, необходимо сохранить полученный файл.

При выполнении лабораторной работы, на этапе назначения номера вывода каждому из портов согласно табл. 2 (с. 7), надо быть особенно внимательным.

В правом окне затемненными прямоугольниками отмечены выводы ПЛИУ, занесенные в список назначений.

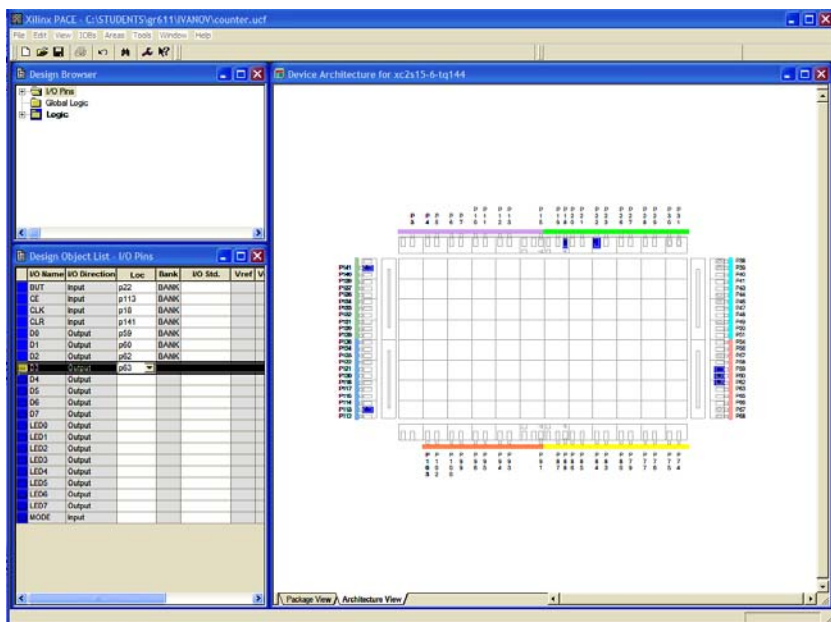


Рис. 16. Назначение выводов микросхемы

4. Формирование тестовых сигналов для проверки работы устройства путем моделирования

Чтобы убедиться в том, что создаваемое устройство работает правильно, необходимо создать тестовые сигналы, используемые в качестве входных воздействий. Тестовые сигналы представляются на испытательном стенде (*Test Bench*) в графическом виде.

Тестовые сигналы создаются следующим образом:

1. Активизируйте файл *counter.sch* в окне *Sources*.
2. Создайте новый испытательный стенд, выбирая *Project* ⇒ *New Source*.
3. В *New Source Wizard* выберите *Test Bench WaveForm* в качестве исходного типа, и в поле *File Name* наберите имя файла, в нашем примере *count.tbw*.
4. Щелкните *Next*.
5. Окно связанных источников (*Associated Source*) показывает, что тестовые сигналы связаны с файлом источника *counter*. Щелкните *Next*.
6. Итоговое (*Summary*) окно (рис. 17) показывает, что источник добавлен к проекту. В этом окне отображается директория источника, его тип и имя. Щелкните *Finish*.

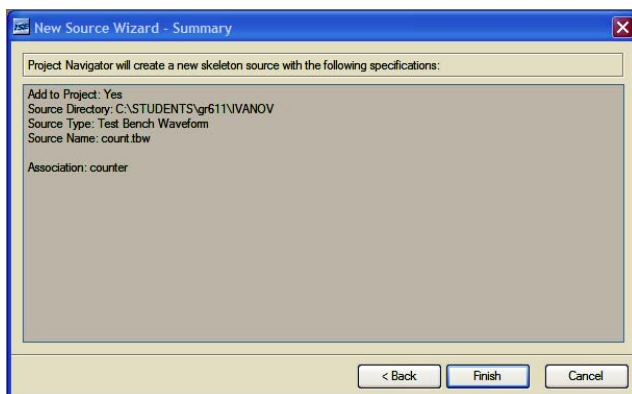


Рис. 17. Итоговое окно

7. В открывшемся диалоговом окне *Initialize Timing* (рис. 18) не-

обходимо установить тактовую частоту, время установления тактового сигнала и время задержки выходного сигнала. Зададимся максимальной частотой тактового сигнала 25 МГц и задержкой выходных сигналов относительно тактового сигнала не более 10 нс. Для выполнения этих требований заполните поля в диалоговом окне *Initialize Timing* следующими данными:

* *Clock High Time*: 20 ns.

* *Clock Low Time*: 20 ns.

* *Input Setup Time*: 10 ns.

* *Output Valid Delay*: 10 ns.

* *Offset*: 0 ns.

* *Global Signals*: GSR (FPGA)

Примечание. При использовании микросхем *FPGA* (на рис. 18 галочкой отмечено, что *GSR (FPGA)* разрешено) по умолчанию к значению *Offset* автоматически добавляется 100 ns.

* *Initial Length of Test Bench*: 10000 ns.

В остальных полях можно оставить значения, присутствующие по умолчанию.

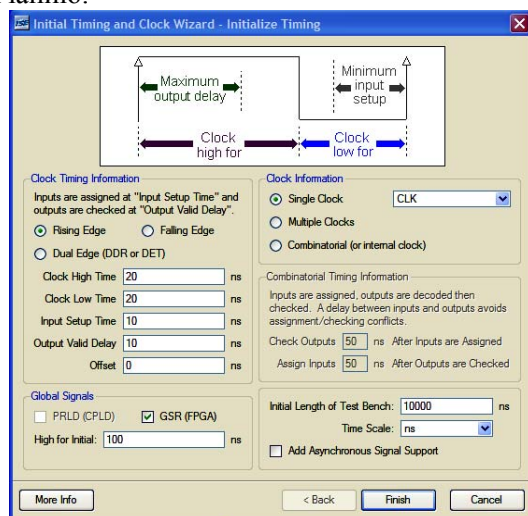


Рис. 18. Инициализация временных параметров

8. Для завершения инициализации временных параметров щелкнуть *Finish*.

9. В открывшемся окне (рис. 19) представлены все заданные ранее входные и выходные сигналы. Затененные области, предшествующие нарастающему фронту *CLK*, соответствуют времени установления входных сигналов (*Input Setup Time*) в диалоговом окне *Initialize Timing*.

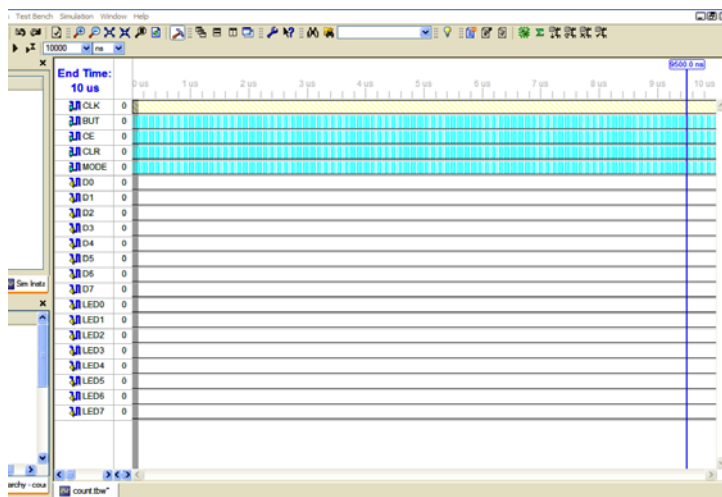


Рис. 19. Окно задания тестовых сигналов

10. Задание тестовых сигналов производится путем нажатия левой клавиши мыши при установке маркера мыши в том месте, где необходимо изменить уровень сигнала на противоположный. Так, при щелчке мышью в точке 1 (рис. 20) значение сигнала *BUT* становится равным 1 до точки конечного времени моделирования. При следующем щелчке в точке 2 значение сигнала *BUT* устанавливается равным 0 и т. д.

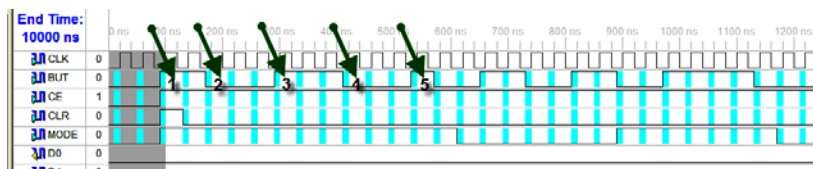


Рис. 20. Задание тестовых сигналов

11. Сохранить файл тестовых сигналов *count_tbw*.
12. В окне *Sources* выберите *Behavioral Simulation* (1), чтобы увидеть, что файл тестовых сигналов автоматически добавлен в проект (2) (рис. 21).

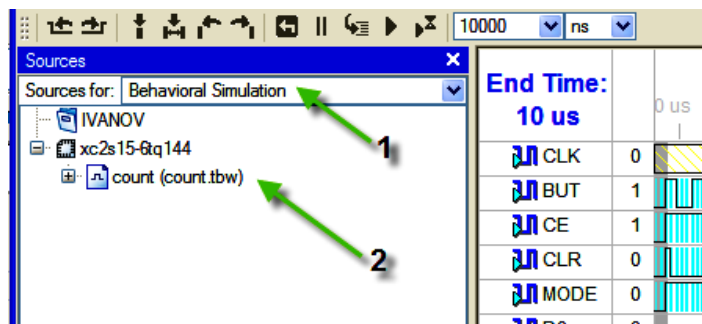


Рис. 21. Выбор *Behavioral Simulation*

13. Закрыть окно тестовых сигналов.

5. Моделирование работы устройства

Убедиться с помощью моделирования в том, что создаваемое устройство будет работать так, как вы ожидаете, можно следующим образом:

1. Убедитесь, что в окне *Sources* выбрано *Behavioral Simulation* и *counter.tbw*.
2. Дважды щелкните по *Add Test Bench To Project*. В окне *Processes*, нажмите "+", чтобы раскрыть *Xilinx ISE Simulator* и дважды щелкните *Simulate Behavioral Model*.
ISE Simulator открывает и выполняет моделирование в интервале времени 1000 ns.
3. Чтобы провести моделирование на установленном ранее интервале времени выберите вкладку *Simulation*, раскройте ее и нажмите на *Run For Specified Time*.
4. Убедитесь, что ваше устройство работает так, как ожидалось

Результаты моделирования имеют вид, показанный на рис. 22.

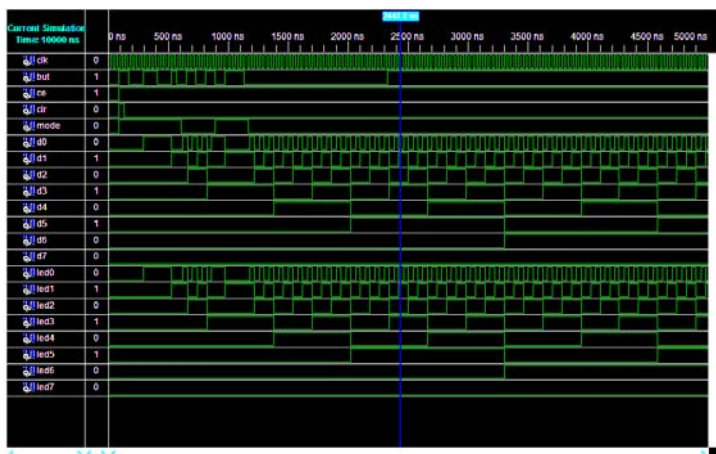


Рис. 22. Результаты моделирования


6. Реализация проекта (Implementation)

1. Активизируйте исходный файл *counter.sch* в окне *Sources*.
2. Дважды щелкните мышью по *Synthesize-XST* для запуска процесса синтеза схемы. Обратите внимание, что после того как выполнение процесса полностью завершено, рядом с его названием появляется галочка в зеленом круге, что указывает на отсутствие ошибок или предупреждений. Наличие предупреждений отмечается появлением восклицательного знака в желтом треугольнике, а ошибки – появлением крестика в красном круге.
3. Следующим шагом запускается процесс реализации проекта, для чего надо дважды щелкнуть мышью по *Implement Design* во вкладке *Processes*. После его завершения переходим к последней операции.

7. Программирование ПЛИС с помощью модуля *iMPACT* пакета *WebPACK ISE*

Для конфигурирования ПЛИС, выпускаемых фирмой *Xilinx*, не требуется специальных аппаратных средств программирования, достаточно загрузочного кабеля.

Прежде чем приступить непосредственно к работе с модулем программирования ПЛИС *iMPACT*, который входит в состав пакета *WebPACK ISE*, надо соединить загрузочный кабель, подключенный к *LPT* порту ПК, с разъемом 5 (рис. 1) макетной платы, предназначенным для конфигурирования кристаллов (*JTAG*-порт). После этого следует подать на макетную плату напряжения питания.

Для программирования ПЛИС необходимо преобразовать результаты, полученные на этапе размещения и трассировки проекта в кристалл, в формат, воспринимаемый средствами программирования. Для создания конфигурационной последовательности (файла программирования) в окне *Sources* активизируем вставку *Source*. Щелчком левой клавиши мыши выделяем строку с названием модуля верхнего уровня иерархии (в данном случае *counter(counter.sch)*), и в окне *Processes* активизируем вставку *Processes*. Затем в окне процессов, щелкнув мышью по значку  у строки *Configure Target Device*, разворачиваем содержимое этого раздела. Далее следует активизировать модуль программирования двойным щелчком левой кнопки мыши на строке *Generate Target PROM/ACE File*. В появившемся окне выбираем ОК, при этом открывается окно (рис. 23), в котором отмечает- ся режим *Prepare a PROM File*.

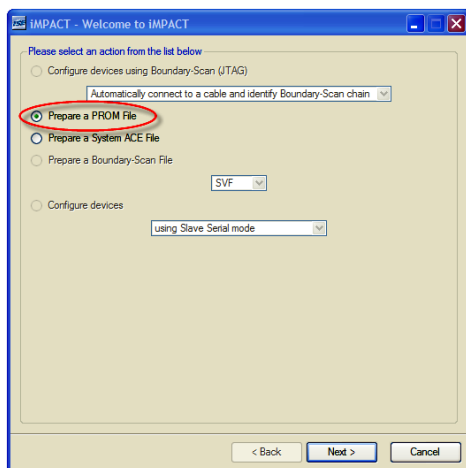
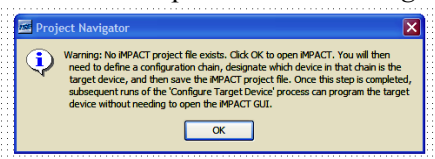


Рис. 23. Выбор режима формирования файла программирования

Следующим шагом выбираем команду *Next* и во всплывающем окне устанавливаем в качестве адресата флэш-память фирмы XILINX и формат файла – *MCS*, как показано на рис. 24.

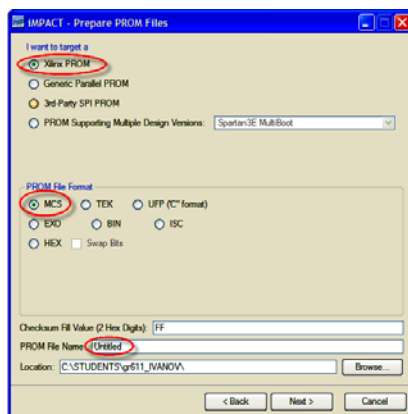


Рис. 24. Выбор адресата и формата загрузочного файла

Имя *PROM* файлу можно присвоить произвольное или оставить имеющееся по умолчанию – *Untitled*. После выполнения команды *Next* открывается окно (рис. 25), в котором выбирается режим записи данных в память. В макете используется режим последовательного ввода данных.

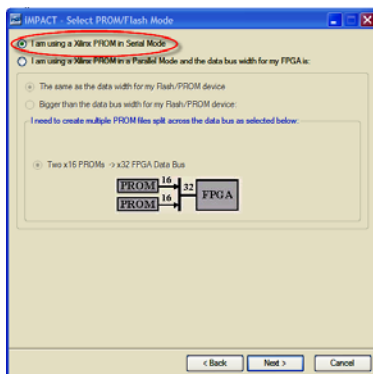


Рис. 25. Окно выбора режима ввода данных

После выбора режима ввода в открывшемся окне (рис. 25) последовательно выбираем тип памяти *xc18v* (1), объем памяти 256K (2) и включаем эту память в проект (3).

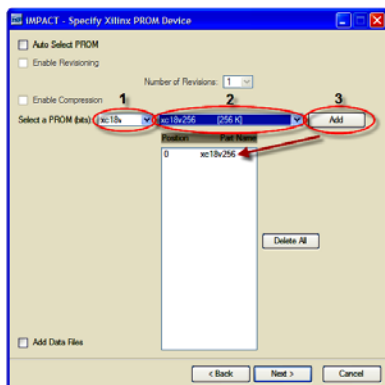


Рис. 26. Окно выбора типа памяти

При переходе к следующей процедуре открывается окно, где предлагается добавить устройство, в которое будут переданы данные из памяти. Соглашаясь на выполнение этой операции, увидим всплывающее окно (рис. 27), в котором выбираем созданный на предыдущих этапах файл с расширением *.bit*.

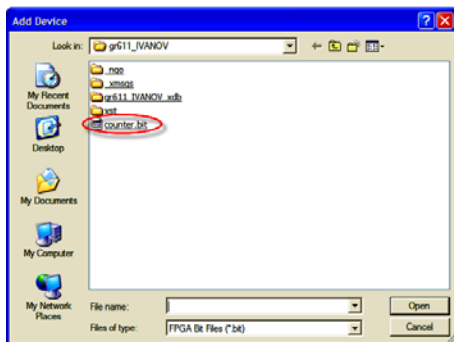
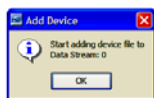
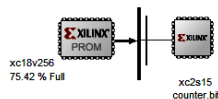


Рис. 27. Окно выбора файла, загружаемого в FPGA

Затем отказываемся от добавления новых устройств и видим на экране изображение цепочки, состоящей из памяти *xc18v256* и FPGA *xc2s15*.



Теперь в окне *Processes* необходимо активизировать появившуюся вставку *Generate File...* После выполнения этой операции

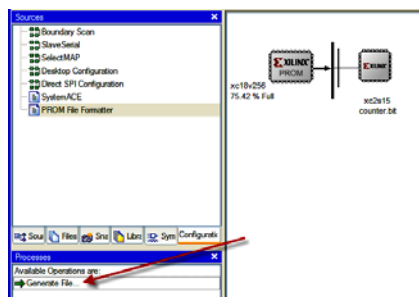


Рис. 28. Создание загрузочного файла

появляется сообщение

PROM File Generation Succeeded

Итак, PROM файл создан, и можно переходить к его загрузке в память. В окне *Sources* активизируем вставку *Boundary Scan* и выполняем команду [Right click to Add Device or Initialize JTAG chain](#). В открывшемся окне *Add Device* выбираем файл с расширением *.mcs* и открываем его (рис. 28).

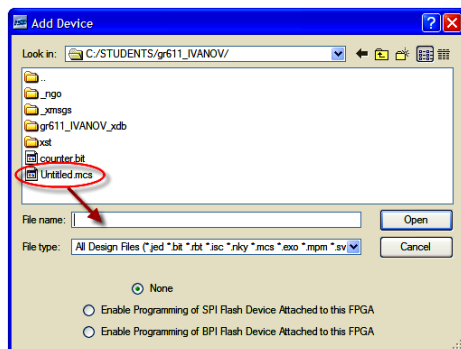
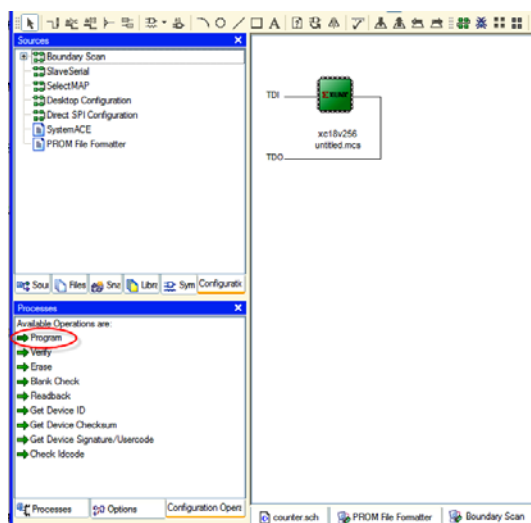
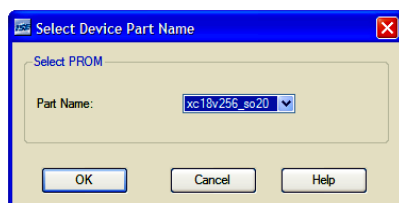


Рис. 28. Выбор загрузочного файла

В окне *Select Device Part Name* можно выбрать тип памяти и корпуса, в котором расположена память. В макете используется память *xc18v256* в корпусе *so20*.



Информация о ходе создания конфигурационной последовательности отображается в окне консольных сообщений (*Comments*) и строке состояния. После успешного завершения этого процесса, отмеченного соответствующей пиктограммой (⚠️ или ✅) в строке *Generate Programming File*, можно приступить к программированию ПЛИС.

Работа программы *iMPACT* в этом случае начинается с обнаружения загрузочного кабеля. Затем, при отсутствии ошибок за-

гружается конфигурационная последовательность, и в окне *Comments* появляется сообщение *Process "Configure Target Device" completed successfully*. Теперь конфигурационная последовательность загружена в флэш-память 2 (рис. 1) и для ее перезаписи в ПЛИС достаточно нажать на кнопку 9 (рис. 1).

8. Задание

1. Получить у преподавателя вариант создаваемого устройства.
2. Нарисовать на бумаге схему заданного устройства и показать ее преподавателю.
3. В схемотехническом редакторе составить принципиальную схему устройства и продемонстрировать ее преподавателю.
4. Ввести временные и топологические ограничения проекта.
5. Проверить работоспособность схемы путем моделирования и показать преподавателю результаты моделирования.
6. Осуществить программирование ПЛИС с помощью модуля *iMPACT*, убедиться в правильной работе устройства и продемонстрировать преподавателю работу устройства.

Приложение

Краткий обзор схемного и символьного редакторов

Схемный и символьный редакторы позволяют создавать и редактировать схемы и символы. Схема (*Schematic*) – графическое представление проекта. Символами (*Symbol*) называют основные элементы схемы, а также ссылки на другие модули, типа модулей HDL.

Возможности схемотехнического редактора *WebPACK ISE*.

Схемотехнический редактор предназначен для создания модулей исходного описания проекта в форме принципиальных схем, а также для формирования условных графических образов (УГО) компонентов и функциональных блоков проектируемого цифрового устройства.

Отличительными особенностями редактора являются:

- возможность одновременного открытия нескольких рабочих окон для создания и редактирования схем и символов;
- использование единой управляющей графической оболочки для создания и редактирования схем и символов;
- удобный механизм создания многостраничных схем;
- эффективные средства разработки многоуровневых иерархических структур;
- двухступенчатый контроль электрических соединений и целостности схемы;
- возможность внесения дополнительной информации, необходимой в процессе проектирования устройства, в форме атрибутов;
- "мастер" автоматического формирования УГО функциональных блоков и компонентов схем;
- возможность размещения на страницах схемы графических и текстовых элементов оформления.

При создании схем в среде редактора используются следующие элементы:

- условные графические образы компонентов из библиотек

- редактора или созданные пользователем;
- элементы соединений (проводники и шины);
- *ИО*-маркеры цепей, предназначенных для подключения "внешних" элементов к выводам интегральных схем;
- атрибуты (параметры компонентов, цепей и контактов);
- текстовая информация;
- декоративные графические элементы.

Два последних типа элементов не являются обязательными. Они не оказывают влияния на процесс проектирования и используются лишь для представления схем (документации) в соответствии с предъявляемыми требованиями.

Пользовательский интерфейс схемотехнического редактора

Запуск схемотехнического редактора производится автоматически из управляющей оболочки *навигатора проекта* при выборе команд создания или редактирования модуля схемотехнического описания проекта. При их успешном выполнении на экране монитора отображается основное окно программы *ECS* в режиме редактирования схем, подробная структура которого показана на рис. П1. Главное окно редактора в режиме создания и модификации символов имеет аналогичную структуру и отличается только составом меню команд оперативной и инструментальной панелей.

Исходное окно схемотехнического редактора включает следующие элементы:

- заголовок окна;
- главное меню;
- оперативную панель управления;
- панель инструментов редактора;
- область расположения рабочих окон;
- панель библиотек символов и дополнительных параметров;
- строку состояния.

Оперативная панель управления и панель инструментов редактора могут быть выключены в текущей конфигурации. Эти элементы пользовательского интерфейса ускоряют процесс создания и редактирования схем и символов, поэтому рекомендуется установить режим отображения этих панелей, используя команды пункта

View главного меню редактора.

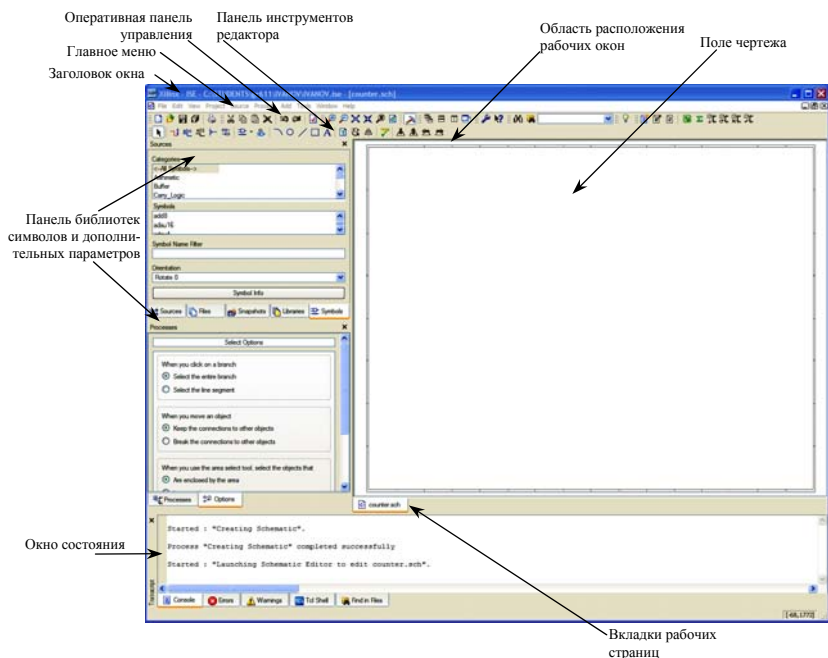


Рис. П1. Основное окно Навигатора проекта пакета *WebPACK ISE*

В заголовке окна схмотехнического редактора отображается название программы и имя редактируемого файла схемы или символа.

Главное меню схмотехнического редактора предоставляет доступ к группам команд, используемых для создания и модификации схем и символов. Каждый пункт главного меню открывает всплывающее меню, в котором находится соответствующая группа команд. Всплывающее меню *File* содержит команды создания, открытия, сохранения и закрытия файлов схмотехнического описания проектируемого устройства или его функциональных блоков. Кроме того, в эту группу включены команды управления печатью, а также завершения работы со схмотехническим редактором. Выпадающее меню *Edit* объединяет стандартные команды редактирования схем, установки и просмотра атрибутов, выбора объектов

схемы для последующих операций, поиска на чертеже схемы символов и цепей. В этой же группе находится команда включения режима редактирования выбранного УГО, используемого в схеме, а также команды переименования шин и цепей. Кроме того, меню *Edit* включает в себя команду контроля и установки параметров конфигурации схемотехнического редактора. Во всплывающем меню *View* сгруппированы команды, управляющие изображением в рабочей области окна. Здесь же находятся команды включения и выключения панелей управления: оперативной, инструментальной и дополнительной. Наличие маркера в строке меню *View* указывает на то, что соответствующая панель представлена в основном окне схемотехнического редактора. Для изменения состояния панелей на противоположное достаточно щелкнуть левой кнопкой мыши на строке всплывающего меню, в которой указано название этой панели. В этой же группе расположена команда перехода на следующий иерархический уровень и обратно. Выпадающее меню *Add* содержит команды, используемые для ввода всех необходимых элементов схемы. Каждая строка этого меню соответствует одному из элементов схемы, рассмотренных в предыдущем разделе. Во всплывающем меню *Tools* находится команда запуска мастера создания символа для представления функционального блока проектируемого устройства, а также команда вывода информации об элементах схемы. Здесь же содержатся команды контроля целостности и наличия ошибок соединений схемы и установки маркеров интерфейсных цепей модуля. Выпадающее меню *Window* содержит стандартные команды управления рабочими окнами редактора: создания, закрытия, переключения и установки их взаимного расположения. Пункт *Help* основного меню открывает доступ к справочной системе схемотехнического редактора.

Оперативная панель содержит кнопки быстрого доступа, которые дублируют наиболее часто используемые команды управления схемотехнического редактора.

В панели инструментов схемотехнического редактора сгруппированы кнопки быстрого доступа, которые предназначены для выполнения операций, часто используемых в процессе создания и редактирования схем. Некоторые кнопки панели инструментов являются недоступными и отображаются серым цветом. Эти кнопки разблокируются только при выполнении операций, для которых

они предназначены. Таким образом, исключается возможность выполнения недопустимых операций.

Встраиваемая панель дополнительных параметров и библиотек символов в режиме создания и редактирования схемы содержит два окна: *Sources* и *Processes*. Окно *Sources* состоит из четырех вкладок *Categories*, *Symbols*, *Symbol Name Filter* и *Orientation*, отображает состав библиотек УГО текущего проекта и обеспечивает механизм быстрого поиска требуемого компонента и размещения его на поле чертежа. В поле выбора *Categories* находится список функциональных групп элементов библиотек текущего проекта. Поле выбора *Symbols* содержит полный список символов функциональной группы компонентов, название которой выделено в поле *Categories*. Поле редактирования *Symbol Name Filter* предназначено для ввода фильтра названий компонентов. Если в поле *Symbol Name Filter* содержится некоторая алфавитно-цифровая последовательность, то в поле *Symbols* будут отображаться только элементы, названия которых начинаются с данной строки текста. В поле выбора *Orientation* указывается ориентация выбранного символа при его размещении на поле чертежа.

Окно *Processes* является расширением инструментальной панели схмотехнического редактора и используется для ввода вспомогательной информации, необходимой в процессе выполнения выбранной операции. Структура этого окна зависит от текущего режима работы (выбранного типа операции), устанавливаемого кнопками панели инструментов.

Область расположения рабочих окон предназначена для размещения окон редактирования схем и символов. Каждое открытое рабочее окно снабжено вкладкой, на которой указано название редактируемого файла, и представляет собой поле чертежа. Закладки располагаются вдоль нижней границы рабочей области основного окна редактора. Расположение окон в рабочей области устанавливается с помощью команд выпадающего меню *Window* или соответствующих кнопок оперативной панели.

Окно состояния, расположенное в нижней части окна схмотехнического редактора, используется для вывода сообщений о текущих выполняемых командах, результатах их выполнения и необходимых дополнительных действиях при этом. Здесь же отображаются координаты текущего положения курсора мыши на поле

чертежа.

Работа с редактором пакета *WebPACK ISE* строится по следующему принципу. Вначале курсором и щелчком левой кнопки мыши указывается объект, над которым необходимо осуществить заданную операцию. Затем с помощью команд меню или кнопок оперативной и инструментальной панелей определяется тип операции, которая должна быть выполнена.

Схемный редактор

Схемный редактор используется для выполнения следующих операций:

- Создание схемы верхнего уровня, которая используется в качестве входных данных для моделирования на поведенческом уровне или при синтезе в *ISE™*.

Замечание. Более низкие уровни схемного проектирования можно определить, используя схемные файлы, файлы *HDL*, *IP* ядра или диаграммы состояния.

- Создание схем низшего уровня для включения в состав схемы верхнего уровня.
- Создание схем низшего уровня для включения в *HDL*-файл.
- Сохранение схемных файлов в качестве архивов или просмотр их как *HDL*-файлов только для чтения.

Редактор символов

Редактор символов используется для выполнения следующих операций:

- Создание нового символа для включения в схему.
- Редактирование существующего символа для последующего включения в схему.

Файлы схемного редактора и редактора символов

Схемный редактор и редактор символов работают со следующими файлами.

Входные файлы

Следующие файлы рассматриваются как входные:

- *SCH*

Файл *SCH* – формат базы данных, в котором сохранены схемы. В *SCH*-файл вводить символы можно используя команды *Add* \Rightarrow *Symbol*.

- *SYM*

Файл *SYM* – формат базы данных, в котором сохранены символы. *SYM*-файлы можно использовать в схемах. Можно создать собственные *SYM*-файлы, используя редактор символов, или создать файл символа для модуля низшего уровня типа *VHDL* или модуля *Verilog*. Символы должны быть созданы для всех модулей низшего уровня, прежде чем они могут быть введены в схему.

Замечание. Может быть использован любой символ из директории вашего проекта или из семейства библиотек устройств.

Выходные файлы

В качестве выходных создаются следующие файлы:

- *SCH*

Файл *SCH* – формат базы данных, в котором сохранены схемы. Сохраняют *SCH*-файл, выполняя команду *File* \Rightarrow *Save*.

Замечание. Чтобы дать возможность другим инструментальным средствам читать или использовать данные проекта, *Project Navigator* автоматически вызывает программу *netlister*, которая преобразует файлы *SCH* в соответствующий *HDL netlist* формат. Схемные данные и данные структур, не поддерживаемые форматами *HDL* (например, неэлектрические объекты), в *netlist* файле не появляются.

- *SYM*

Файл *SYM* – формат базы данных, в котором сохранены символы. Сохраняют *SYM*-файл, выполняя команду *File* \Rightarrow *Save*.

Рекомендации для схемного проектирования

Создавая схему и символы, используйте следующие рекомендации, чтобы гарантировать успешное решение задачи.

Общие рекомендации

- I/O-компоненты типа *IBUF* и *OBUF* используйте только на верхнем уровне проекта.
- Когда несколько сигналов в шине соединяются с портом ввода/вывода, убедитесь, что каждый из них имеет одинаковую полярность (входной, выходной или двунаправленный).
- Убедитесь, что схемы, выводы, символы и цепи имеют разрешенные имена. Разрешенные имена должны начинаться с символа (*A-Z*, *a-z*) и могут содержать только алфавитно-цифровые символы (*A-Z*, *a-z*, 0-9) и символы подчеркивания (*_*).

Замечание. Требование вызвано тем, что *Verilog* и *VHDL* идентификаторы должны придерживаться этого соглашения об именах.

- Убедитесь, что выводы, символы, шины и цепи не используют слова, зарезервированные *VHDL* или *Verilog*.
- Помните, что когда два или больше сигналов формируют один сигнал, он должен быть образован монтажным ИЛИ.
- Если вы помещаете *IO*-компоненты в качестве *IO*-портов, а не как-то иначе, используете только *IO*-примитивы типа *IBUF* и *IFD*. Не используйте в этом случае *IO*-макросы, типа *IBUF4* и *IFD_1*. При использовании *IO*-макросов, выбранный механизм синтеза может вставить в портах дополнительные буферы *IBUF* или *OBUF*, вызывая ошибки при выполнении *MAP*.
- Называя или переименовывая цепи, используйте опции *Add Net Name* или *Rename Bus* вместо того, чтобы дважды щелкать на изображении цепи и изменять свойства объекта. Эти опции дают больше гибкости и возможность выбрать, изменить имя части цепи или всей цепи.

Подсказки HDL

- Помните, что промежуточный *HDL* (*Verilog/VHDL*) определен выбранным *Functional Model Target Language* в диалоговом окне *Functional Model Properties*.
- Убедитесь, что при использовании *VHDL* на все входные выводы поданы сигналы. Это гарантирует определенное значение сигнала для входного вывода. Непод-

ключенные выводы приводят к появлению предупреждения, и если они принадлежат символу унифицированной библиотеки *Xilinx®*, то принимают значение по умолчанию.

- Помните, что *Verilog* чувствителен к регистру клавиатуры, а *VHDL* не чувствителен.

Дополнительно используйте следующие советы при работе со схемами и символами.

- Чтобы обновить все схемы в проекте с текущими символами из ваших директорий символов, выполните процесс *Update All Schematic Files*. Если Вы редактировали символ, то должны запустить этот процесс.
- Чтобы изменить цветовую схему редакторов схем и символов, используйте *Edit ⇒ Preferences*.
- Для увеличения масштаба изображения схемы или символа нажмите *Ctrl* и перемещайте мышь вниз и направо. Для уменьшения масштаба нажмите *Ctrl* и перемещайте мышь вверх и налево.
- Чтобы изменить размер листа, щелкните мышью на пустом месте схемы и выберите *Edit ⇒ Object Properties*.
- Чтобы переименовывать цепь, щелкните правой кнопкой мыши по изображению цепи и выберите *Rename Selected Net*. Вы можете переименовать отдельную ветвь цепи или всю цепь.
- Чтобы быстро добавлять *IO* маркеры и соединительные провода к выводам символа, выбирают *Add ⇒ IO Marker* и затем используют мышь, чтобы перетащить выбранное изображение маркера к входному или выходному выводу символа.
- Чтобы быстро находить все ветви цепи, даже если они явно не связаны, выберите *Tools ⇒ Query*.

***Schematics* (схемы)**

Схема включает следующие компоненты:

- **Instances (объекты)** – ссылки на символы из библиотеки символов устройств *Xilinx®*, на символы, которые вы создаете или на символы из других проектов.
- **Nets (цепи)** – связывают объекты и *IO* маркеры.
- **IO markers** указывают на место, где сигналы входят или выходят из схемы и показывают направление (полярность) сигнала (ввод, вывод или двунаправленный).

Схема может также содержать графические и текстовые аннотации, наряду с элементами блока, которые представляют другие схемы. Схема может содержать более одного листа, их количество практически неограничено.

Symbols (символа) и Instances (объекты)

Symbols являются графическим представлением компонентов. Обычно, символы – электрические компоненты типа вентилях. Символы могут также не иметь никакого электрического значения, как, например, заголовок блока. ***Instances*** - справочная информация на символы, которые появляются в изображении схемы.

Символы включают следующие основные элементы:

- **Графические символы (*Graphics*)** – изображения, которые определяют вид символа и не имеют никакого электрического значения. Электрические параметры определяются выводами (*pin*) символа и его атрибутами. Пояснительный текст, отображаемый на символе, является графической информацией, не имеющей электрического значения.
- **Выводы (*Pins*)** обеспечивают электрические соединения между символом и проводником в схеме. Если символ представляет собой отдельный компонент, то вывод символа представляет физический вывод, к которому может быть подключен проводник. Выводы могут представлять или единственную электрическую точку подключения (скалярный вывод) или многократные электрические точки подключения (шинный вывод).

При редактировании схемного (*SCH*) файла, таблица символов (***Symbols***) обеспечивает доступ к символам из библиотеки символов устройств *Xilinx®*, к символам, которые вы создаете или к

символам из других проектов. Вы можете создать свои собственные символы, используя *Symbol Wizard* (мастер символов).

***Nets* (цепи) и *Buses* (шины) .**

***Net* (цепь)** – совокупность связанных выводов и проводников. Каждый проводник имеет имя цепи, которое идентифицирует его в схемном и символьном редакторах и в программе *netlister*. Два проводника или больше могут иметь одно то же имя цепи. Каждый проводник, который использует общее имя цепи, становится частью этой цепи, и все выводы символов, связанные с этими проводниками, связаны электрически.

По умолчанию цепи автоматически назначается имя *XLXN_number*. Вы можете изменить это имя на более содержательное. Например, можно переименовать цепь так, чтобы имя отражало функцию сигнала, распространяющегося по этой цепи.

Возможны 2 типа цепей:

- **Скалярные цепи**, имеющие единственный сигнал.
- **Шины**, являющиеся группами скалярных цепей, имеют несколько сигналов.

Шины

Шины – удобный способ представления связанной группы сигналов. Однако шины могут быть группой любых сигналов, как связанных, так и нет. Группирование позволяет получить менее громоздженную схему, функционально более наглядный рисунок. При этом становятся более простыми соединения между основной схемой и блоком. Шины особенно полезны в следующих ситуациях:

- Маршрутизация множества сигналов от одной стороны схемы к другой.
- Подключение более одного сигнала к блоку.
- Соединение нескольких сигналов, проходящих через иерархические уровни, с единственным *IO* маркером.

Шинные отводы (*Bus taps*) дают возможность подключить скалярную цепь к соответствующей части шины. Шинные отводы используются для:

- соединения одной из цепей шины к скалярному выводу символа,
- соединения части шины с выводами шины меньшей ширины,
- соединения одной цепи или части шины с *IO* маркером.

Можно также подключить несколько цепей от шины или всю шину к шинному выводу символа, если число сигналов соответствует числу выводов.

***IO* Маркеры (маркеры ввода/вывода)**

IO маркеры присоединены к цепям и определяют направление сигнала (ввод, вывод или двунаправленный). Направление *IO* маркера должно соответствовать направлению вывода, к которому подключена его цепь. Например, если цепь связана с выводом *D*-триггера (ввод), направление *IO* маркера, связанного с цепью, должно быть на ввод. *IO* маркеры показывают также, что цепь доступна извне.

IO маркеры на схеме верхнего уровня соединяются с выводами устройства, в то время как *IO* маркеры на схемах низшего уровня соединяются с выводами символа. В зависимости от их размещения *IO* маркеры указывают на различные типы подключений, как показано в следующей таблице.

Местоположение <i>IO</i> маркера	Тип Подключения
Схема верхнего уровня	Подключение к внешнему выводу
Схема низшего уровня	Подключение к листу на другом иерархическом уровне
Отсутствие <i>IO</i> маркера	Подключение к другим проводам с тем же самым именем, на том же самом иерархическом уровне (или на текущем листе или другом листе)

IO маркер можно поместить только в конце сегмента цепи. Если *IO* маркер помещается непосредственно у вывода символа, то между ними автоматически вставляется проводник.

Attributes (признаки)

Признаки дают информацию об электрических характеристиках и поведении символов и цепей. Инструментальные средства Xilinx® и имитаторы используют признаки, чтобы проанализировать и обработать различные компоненты схемы. Признаки могут содержать следующую информацию:

- о размещении и трассировке,
- о библиотеке символов,
- о директивах соответствия и оптимизации,
- об ограничениях.

Признак состоит из имени и значения. Обычно имя отражает назначение признака (например, *PortPolarity*). Значение дает информацию о признаке. Например, признак *PortPolarity* может иметь значение вход, выход или двунаправленный.

Рабочая область редакторов схем и символов

В рабочей области Вы можете редактировать изображение схемы и символа. Одновременно открытыми можно иметь несколько файлов. Для переключения между различными файлами выберите нужную вкладку внизу рабочей области. Если схема расположена на нескольких листах, вкладки на левой стороне схемы указывают на каждый из листов. Для переключения между различными листами выберите соответствующую вкладку слева.

С объектами в схемном файле или файле символа можно работать следующим образом.

- Чтобы добавить объект, выберите команду из меню *Add*, сделайте соответствующие параметры настройки во вкладке *Options* и затем нажмите левую клавишу мыши в рабочей области схемы или символа.
- Чтобы управлять существующими объектами, выберите объект, а затем действие, которое необходимо выполнить, используя меню команды или кнопки панели.

Обратите внимание, что почти все команды перманентны. Это означает, что они остаются в силе до тех пор, пока не выбрана другая команда.

Вкладка *Symbols* (символов)

Вкладка *Symbols*, называемая также навигатор символов, позволяет выбирать символ, чтобы поместить его в схему. Символы из библиотек символов *Xilinx*® можно выбрать из директории текущего проекта или из библиотеки символов, созданной Вами.

Учтите, что если вы изменяете семейство устройств, используемых в созданном проекте при открытом схемном файле, то необходимо закрыть вашу схему и повторно открыть ее, чтобы обновить символы во вкладке *Symbols*.

Categories (категории)

Определяют тип символа, который вы добавляете в свою схему, что помогает сужать поиск символа.

All Symbols

Содержит унифицированные символы библиотеки *Xilinx*®, символы и любые другие символы, доступные в проекте.

Directory paths (пути каталога)

Показывают директории, содержащие созданные и ставшие доступными для текущей схемы символы.

Symbol types (типы символов)

Показывают символы унифицированной библиотеки *Xilinx*® определенного типа. Например, чтобы увидеть все буферы, имеющиеся в библиотеке, щелкните по вкладке *Buffer*.

Symbols (символы)

Здесь можно выбирать символ библиотеки, чтобы добавить его в схему. Список символов изменяется, когда в поле *Categories* выбираются другие объекты или вводится текст в поле *Symbol Name Filter*. Чтобы выбирать символ, щелкните мышью по имени символа.

Symbol Name Filter (фильтр имени символа)

Позволяет сужать поиск после набора первой буквы или нескольких букв имени символа. Поле *Symbols* показывает только

символы, имена которых начинаются с набранных букв. Вы можете также искать символы путем сопоставления с образцом, используя следующие специальные знаки: звездочка (*) – представляет любое число букв, и вопросительный знак (?) – заменяет одну букву.

Orientation (ориентация)

Определяет, каким образом выбранный символ отображается на схеме. Возможно зеркальное изображение и повороты.

Symbol Info (информация о символе)

Открывает раздел *Xilinx® Libraries Guides*, в котором описан выбранный символ.

Options Tab (вкладка опций)

Вкладка *Options* изменяется в зависимости от используемых инструментальных средств.

Schematic Editor (схемный редактор)






















- Add Bus Tap Options*
- Add Circle Options*
- Add Instance Name Options*
- Add IO Marker Options*
- Add Net Name Options*
- Add Text Options*
- Add Wire Options*
- Select Options*

Symbol Editor (редактор символов)

- Add Attribute Window Options*
- Add Bubble Options*
- Add Circle Options*
- Add Pin Options*
- Add Text Options*
- Select Options*











Панель инструментов схемного редактора

Панель инструментов схемного редактора позволяет вносить изменения в схему. Чтобы сделать видимой или скрыть панель инструментов, выберите *View* \Rightarrow *Toolbars* \Rightarrow *Schematic Editor*.

	<i>Select</i> – переводит редактор в режим выбора, давая возможность выбрать объекты.
	<i>Add Wire</i> – нарисовать провод между выводами символа, <i>IO</i> маркерами или отводами шины.
	<i>Add Net Name</i> – назначить уникальное имя цепи или шины.
	<i>Rename Bus</i> – переименовать выбранные отводы шины и цепи, связанные с шиной.
	<i>Add Bus Tap</i> – добавить отвод шины.
	<i>Add IO Marker</i> – добавить <i>IO</i> маркер к цепи.
	<i>Add Symbol</i> – выбрать и поместить символ в схему. Можно также воспользоваться раскрывающимся списком рядом с кнопкой панели <i>Add Symbol</i> , чтобы выбрать последний используемый символ и добавить его в схему. Список содержит пять последних символов, помещенных в схему.
	<i>Add Instance Name</i> – назначить имя объекта символу.
	<i>Add Arc</i> – добавить неэлектрические дуги к рисунку.
	<i>Add Circle</i> – добавить неэлектрические круги к рисунку.
	<i>Add Line</i> – добавить неэлектрические линии к рисунку.
	<i>Add Rectangle</i> – добавить неэлектрические прямоугольники к рисунку.
	<i>Add Text</i> – добавить текст к рисунку.
	<i>Query</i> – получить информацию о выбранном объекте.
	<i>Rotate</i> – повернуть выбранный объект на 90 градусов против часовой стрелки.
	<i>Mirror</i> – зеркально отразить выбранный объект относительно оси <i>Y</i> .
	<i>Check Schematic</i> – выполнить проверку правил проектирования схемы (DRC).
	<i>Push</i> – показать модуль, лежащий в основе объекта.
	<i>Pop</i> – показать схему, расположенную на один уровень выше схемы, которая видна.
	<i>Previous View</i> – показать предыдущее окно.
	<i>Next View</i> . Если Вы использовали кнопку <i>Previous View</i> , то будет отображено окно, с которого Вы попали в текущее окно

Панель инструментов редактора символов

Панель инструментов схемного редактора позволяет редактировать символы. Чтобы сделать видимой или скрыть панель инструментов, выберите *View ⇒ Toolbars ⇒ Symbol Editor*.

	<i>Select</i> – переводит редактор в режим выбора, давая возможность выбрать объекты.
	<i>Add Pin</i> – определяет местоположения вывода (<i>pin</i>) символа.
	<i>Add Attribute Window</i> – добавляет окно атрибута к символу. Окно атрибута отображает значение признака для символа или вывода.
	<i>Add Arc</i> – добавить неэлектрические дуги к рисунку.
	<i>Add Bubble</i> – добавить изображение отрицания к символу.
	<i>Add Circle</i> – добавить неэлектрические круги к рисунку.
	<i>Add Line</i> – добавить неэлектрические линии к рисунку.
	<i>Add Rectangle</i> – добавить неэлектрические прямоугольники к рисунку.
	<i>Add Text</i> – добавить текст к рисунку.
	<i>Check Symbol</i> – выполнить проверку правил проектирования символа (<i>DRC</i>).

Всплывающие Меню

Правая кнопка мыши обеспечивает доступ к часто используемым командам. Чтобы обратиться к команде, выберите объект или фон окна, нажмите правую кнопку мыши и выберите команду из всплывающего меню. Меню, которое появляется при нажатии на правую кнопку мыши, зависит от редактора, в котором Вы работаете (схемном или символьном), и от места на экране, где была нажата клавиша мыши.

Клавиши быстрого доступа

Клавиша	Команда
<i>Esc</i>	Выбрать объекты
<i>Left Arrow</i>	Прокрутка влево
<i>Right Arrow</i>	Прокрутка вправо
<i>Up Arrow</i>	Прокрутка вверх
<i>Down Arrow</i>	Прокрутка вниз

Клавиша	Команда
<i>Home</i>	Прокрутка вверх листа
<i>End</i>	Прокрутка вниз листа
<i>F5</i>	Обновление
<i>F6</i>	Полномасштабное изображение
<i>F7</i>	Уменьшить масштаб
<i>F8</i>	Увеличить масштаб
<i>F11</i>	Увеличить выделенные объекты
<i>Ctrl+A</i>	Выделить все
<i>Ctrl+B</i>	Добавить вывод шины
<i>Ctrl+D</i>	Добавить имя цепи
<i>Ctrl+E</i>	Зеркальное отображение
<i>Ctrl+G</i>	Добавить IO маркер
<i>Ctrl+J</i>	Добавить имя объекта
<i>Ctrl+L</i>	Добавить линию
<i>Ctrl+M</i>	Добавить символ
<i>Ctrl+R</i>	Повернуть
<i>Ctrl+T</i>	Добавить текст
<i>Ctrl+W</i>	Добавить проводник
<i>Ctrl+Left</i>	Прокрутка к левой стороне листа
<i>Ctrl+Right</i>	Прокрутка к правой стороне листа

Действия мышью

Действия мышью	Результат
<i>Ctrl</i> + рисование прямоугольника вниз и вправо	Увеличение масштаба изображения, выделенного прямоугольником
<i>Ctrl</i> + рисование прямоугольника вверх и влево	Уменьшение масштаба изображения, выделенного прямоугольником
Двойной щелчок по изображению объекта	В схемном или символьном редакторе отображаются свойства объекта
Вращение колеса мыши	Прокрутка вверх/вниз
<i>Shift</i> + вращение колеса мыши	Прокрутка влево/вправо
<i>Ctrl</i> + вращение колеса мыши	Увеличение/уменьшение масштаба

Сообщения об ошибках и предупреждения, выдаваемые программой проверки схемы (DRC).

Сообщение	Описание
Error: Block attribute <i>block_attribute_name</i> is an illegal block attribute name. It is a reserved word for <i>VHDL</i> or <i>Verilog</i> .	Имя признака объекта не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Block <i>block_name</i> is an illegal block name. It is a reserved word for <i>vhdl</i> or <i>verilog</i> .	Имя объекта не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Bus attribute <i>bus_attribute_name</i> is an illegal bus attribute name. It is a reserved word for <i>vhdl</i> or <i>verilog</i> .	Имя признака шины не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Bus <i>bus_name</i> and subbus <i>subbus_name</i> are connected to different <i>IO</i> markers.	Шина и ее часть (например, шина A(7:0) и ее фрагмент A(1:0)) не могут быть соединены с независимыми IO маркерами. Если шина соединена с портом, ее фрагмент не может быть связан с портом.
Error: Bus <i>bus_name</i> and its bus members must be connected to symbol pins or <i>IO</i> markers.	Шина и ее часть (например, шина A(7:0) и ее фрагмент A(1:0)) не могут быть соединены с выводом символа или <i>IO</i> маркером.
Error: Bus <i>Bus_name</i> cannot be connected both to input ports and instance output pins.	Шина не может быть соединена с входными портами и выходными выводами одновременно.
Error: Bus <i>bus_name</i> is connected to load pin(s) and/or <i>IO</i> markers, but not connected to any source pin or <i>IO</i> marker.	Шина должна иметь источник сигнала.
Error: Bus <i>bus_name</i> is connected to source pins and/or <i>IO</i> markers, but not connected to any load pin or <i>IO</i> marker.	Шина должна иметь нагрузку.
Warning: Bus <i>bus_name</i> is connected to too many source pins and/or <i>IO</i> markers. A Wired OR connection will be used.	Шина не может быть соединена с несколькими выходными выводами.

Сообщение	Описание
Error: Members of bus <i>bus_name</i> are connected to <i>IO</i> markers with different polarities. All bus members must be connected to <i>IO</i> markers with the same polarity, either Input, Output, or Bidirectional.	Все цепи шины должны быть соединены с <i>IO</i> маркерами, имеющими соответствующее направление (полярность) сигнала.
Error: Some members of bus <i>bus_name</i> are connected to <i>IO</i> markers while other members are not. Bus members must either all be connected to <i>IO</i> markers or none be connected to <i>IO</i> markers.	С <i>IO</i> маркерами должны быть соединены все цепи шины или не соединена ни одна.
Error: Bus <i>bus_name</i> is an illegal bus name. It is a reserved word for <i>vhdl</i> or <i>verilog</i> .	Имя шины не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Bus tap at (<i>location</i>) has no source or destination.	Вход и выход шинного отвода должны быть соединены с цепью.
Error: Incorrect bus tap at (<i>location</i>); destination branch <i>d_branch_name</i> is not part of the source branch <i>s_branch_name</i> .	Цепь, подключенная к шинному отводу, должна иметь имя, совпадающее с именем одной из цепей в шине.
Error: Schematic attribute <i>schematic_attribute_name</i> is an illegal schematic attribute name. It is a reserved word for <i>vhdl</i> or <i>verilog</i> .	Имя схемного признака не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Schematic name <i>schematic_name</i> has an illegal name. Legal schematic name is ([A-Za-z0-9][A-Za-z_]*).	Схема должна иметь разрешенное имя. Разрешенные имена должны начинаться с буквы (<i>A-Z</i> , <i>a-z</i>) и содержать только алфавитно-цифровые знаки (<i>A-Z</i> , <i>a-z</i> , 0–9) и подчеркивание (<i>_</i>).
Error: Schematic <i>schematic_name</i> is an illegal schematic name. It is a reserved word for <i>vhdl</i> or <i>verilog</i> .	Имя схемы не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Instance <i>instance_name</i> has an illegal name <i>instance_name</i> . Legal instance name is either a scalar name ([A-Za-z_][A-Za-z0-9]*) or an iterative array (For example, <i>five_inst</i> (4:0)).	Объект должен иметь разрешенное имя. Разрешенные имена должны начинаться с буквы (<i>A-Z</i> , <i>a-z</i>) и содержать только алфавитно-цифровые знаки (<i>A-Z</i> , <i>a-z</i> , 0–9) и подчеркивание (<i>_</i>).

Сообщение	Описание
Error: Instance <i>instance_name</i> has the same name as a symbol.	Объект не может иметь то же имя, что и имя, созданное пользователем, или имеющееся в библиотеке <i>Xilinx®</i> .
Error: Instance <i>instance_name</i> referencing symbol <i>symbol_name</i> is out of date.	Символ, отражающий объект, имеет более позднее время модификации, чем объект. Это обычно означает, что символ был изменен, но объект не был обновлен.
Error: Instance <i>instance_name</i> refers to a symbol which has an illegal name <i>symbol_name</i> . Legal symbol name is ([A-Za-z_][A-Za-z0-9_]*).	Символ должен иметь разрешенное имя. Разрешенные имена должны начинаться с буквы (<i>A-Z</i> , <i>a-z</i>) и содержать только алфавитно-цифровые знаки (<i>A-Z</i> , <i>a-z</i> , 0–9) и подчеркивание (<i>_</i>).
Error: I/O marker at (<i>location</i>) is not connected to a net or bus.	<i>IO</i> маркер должен быть соединен с цепью или шиной.
Error: Net attribute <i>net_attribute_name</i> is an illegal net attribute name. It is a reserved word for <i>VHDL</i> or <i>Verilog</i> .	Имя признака цепи не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Net is not named.	Отсутствует имя цепи в схеме.
Error: Net <i>net_name</i> is an illegal name. Legal net name is either a scalar name ([A-Za-z_][A-Za-z0-9_]*), a concatenated name, or a bus name.	Цепь должна иметь разрешенное имя. Разрешенные имена должны начинаться с буквы (<i>A-Z</i> , <i>a-z</i>) и содержать только алфавитно-цифровые знаки (<i>A-Z</i> , <i>a-z</i> , 0–9) и подчеркивание (<i>_</i>).
Error: Net <i>net_name</i> has the same name as the symbol referenced by instance <i>instance_name</i> .	Цепь не может иметь то же имя, что и имя, созданное пользователем или, имеющееся в библиотеке <i>Xilinx®</i> .
Error: Net <i>net_name</i> has the same name as a symbol instance.	Цепь и объект символа не могут иметь одно и то же имя.
Error: Port-Pin mismatch on symbol <i>symbol_name</i> . No symbol pin exists for port <i>port_name</i> .	Символ должен иметь вывод, соответствующий каждой цепи с <i>I/O</i> маркером в основной схеме.
Error: Net <i>net_name</i> is an illegal name, because there is no corresponding bit bus for this bit bus member.	Недопустимо имя цепи <i>A(0)</i> , если нет шины с именем <i>A(0:XX)</i> .

Сообщение	Описание
Error: Net <i>net_name</i> is an illegal net name. It is a reserved word for <i>vhdl_or_verilog</i> .	Имя цепи не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Net <i>Net_name</i> cannot be connected both to an input port and an instance output pin.	Шина не может быть соединена с входными портами и выходными выводами одновременно.
Error: Net <i>net_name</i> is connected to load pin(s) and/or IO markers, but not connected to any source pin or IO marker.	Цепь должна иметь источник сигнала.
Error: Net <i>net_name</i> is connected to source pins and/or I/O markers, but not connected to any load pin or IO marker.	Цепь должна иметь нагрузку.
Warning: Net <i>net_name</i> is connected to too many source pins and/or IO markers. A Wired OR connection will be used.	Цепь не может быть соединена с несколькими выходными выводами.
Warning: Net <i>net_name</i> overlaps with net <i>net_name</i> .	Две цепи в схеме не должны перекрываться.
Error: Net <i>net_name</i> must be connected to a symbol pin or an IO marker.	Цепь должна быть соединена с выводом символа или с <i>IO</i> маркером.
Error: Net <i>net_name</i> is an illegal net name, because it is similar to <i>net_name2</i> .	Если в схеме имеется цепь с именем <i>A</i> , то нельзя назвать другую цепь в схеме <i>A(0:3)</i> .
Warning: <i>Object_name</i> is outside the sheet boundary.	Объект не может быть помещен вне листа.
Error: Pin attribute <i>pin_attribute_name</i> is an illegal pin attribute name. It is a reserved word for <i>vhdl_or_verilog</i> .	Имя атрибута вывода не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Pin is not named.	Отсутствует имя вывода в схеме.
Error: Pin <i>pin_name</i> is an illegal pin name. It is a reserved word for <i>vhdl_or_verilog</i> .	Имя вывода не может быть сохранено. Зарезервировано как ключевое слово <i>VHDL</i> или <i>Verilog</i> .
Error: Pin <i>pin_name</i> is connected to a bus of a different width.	Вывод в виде шины должен быть соединен с шиной той же самой размерности. Например, шинный вывод <i>A(0:3)</i> не может быть соединен с шиной <i>A(0:8)</i> .

Сообщение	Описание
Error: Pin <i>pin_name</i> is connected to an unnamed bus.	Вывод связан с шиной, которая не имеет имени.
Error: Pin <i>pin_name</i> is connected to an unordered bus.	Вывод соединен с неупорядоченной шиной.
Warning: Pin <i>pin_name</i> must be connected to a net.	Объект имеет «висящий» вывод, т. е. не подключенный к цепи.
Error: Source branch of bus tap at (<i>location</i>) is not a bus.	Источником для шинного отвода должна быть шина.
Error: Pin polarity mismatch on symbol <i>symbol_name</i> . Symbol pin <i>pin_name</i> and underlying schematic port have different polarity.	Вывод символа должен иметь соответствующую цепь в основной схеме, и полярность вывода должна быть такой же, как полярность цепи.
Error: Symbol pin <i>pin_name</i> implies an IO marker with the same name in the associated schematic. But <i>schematic_file_name</i> does not contain an IO marker <i>pin_name</i> .	Вывод символа должен иметь соответствующую цепь в основной схеме, а цепь должна быть соединена с <i>IO</i> маркером.
Error: Symbol <i>symbol_name</i> referenced by instance <i>instance_name</i> cannot be found.	Символ объекта не может быть найден. Символ должен быть в одной из директорий библиотеки <i>Xilinx</i> , в директории проекта или в какой-либо другой директории, показанной во вкладке Symbols.
Error: Net <i>net-name</i> is connected to multiple source pins. A Wired OR connection is not allowed for at least one of the connected source pins.	Все выходные выводы должны иметь признак " <i>WiredOrOK</i> " в состоянии <i>TRUE</i> (ИСТИННО), иметь возможность подключить несколько выходных выводов к одной цепи.
Error: Bus <i>bus_name</i> is connected to multiple source pins. A Wired OR connection is not allowed for at least one of the connected source pins.	Все выходные выводы должны иметь признак " <i>WiredOrOK</i> " в состоянии <i>TRUE</i> (ИСТИННО), иметь возможность подключить несколько выходных выводов к шине.
Warning: Net <i>net-name</i> is connected to an input/output I/O marker and an instance bidirectional pin.	Двунаправленный порт в модуле более низкого уровня не должен быть соединен с обычным выходным или входным портом в схеме верхнего уровня.

Сообщение	Описание
Warning: Bus <i>bus-name</i> is connected to input/output I/O markers and instance bidirectional pins.	Двунаправленный порт в модуле более низкого уровня не должен быть соединен с обычным выходным или входным портом в схеме верхнего уровня.
Error: Schematic <i>schematic_name</i> begins with an underscore. Schematic name cannot begin with an underscore.	Имя схемы не может начинаться с подчеркивания.
Error: Schematic attribute <i>schematic_attribute_name</i> begins with an underscore. Schematic attribute name cannot begin with an underscore.	Имя схемного признака не может начинаться с подчеркивания.
Error: Pin/Block/Net/Bus <i>name</i> begins with an underscore. Pin/Block/Net/Bus name cannot begin with an underscore.	Имя вывода, блока, цепи или шины не может начинаться с подчеркивания.
Error: Pin/Block/Net/Bus attribute <i>attribute_name</i> begins with an underscore. Pin/Block/Net/Bus attribute name cannot begin with an underscore.	Имя признака вывода, блока, цепи или шины не может начинаться с подчеркивания.

Литература

1. *Зотов В. Ю.* Проектирование цифровых устройств на основе ПЛИС фирмы XILINX в САПР WebPACK ISE. – М.: Горячая линия-Телеком, 2003.
2. *Зотов В. Ю.* Схемотехнический редактор пакета WebPACK ISE. Журнал «Компоненты и технологии». 2001, № 8.
3. *Ларин А.Л.* Основы цифровой электроники: учебное пособие. – М.: МФТИ, 2008.
4. *Титце У., Шенк К.* Полупроводниковая схемотехника. В 2-х т. – М.: Додэка, 2008.
5. *Уэйкерли Дж.* Проектирование цифровых устройств. В 2-х т. – М.: Постмаркет, 2002.
6. URL: <http://chip.phys.msu.ru/docs/fpga/max.djvu>
7. URL: <http://www.fpga-cpld.ru/saprl.html>
8. URL: http://www.xilinx.com/direct/ise10_tutorials/ise10tut.pdf