

Часть 3

Самоучитель САПР Quartus II

Данное учебное пособие познакомит вас с функциональными возможностями и основными характеристиками программного пакета Quartus, представляющего систему автоматизированного проектирования цифровых устройств на СБИС программируемой логики фирмы Altera.

3.1. Обзор учебного пособия

В учебном пособии описывается типовая процедура проектирования на примере разработки цифрового фильтра с конечной импульсной характеристикой (КИХ-фильтра). Проект **fir_filter**.

Представленный проект создается по нисходящей иерархической методологии проектирования. Файлы описания проекта создаются в текстовом и графическом редакторах пакета. В начале создается общая блок-схема верхнего уровня иерархии; затем выполняются описания отдельных составляющих проекта; после этого файлы проекта объединяются.

Затем определяются и анализируются требования к временным характеристикам проекта, и выполняется его компиляция. После этого проводится моделирование проекта (с учетом внутрисхемных задержек) и выполняется анализ временных характеристик. В конечном итоге конфигурация созданного проекта загружается в микросхему семейства АРЕХ™ фирмы Altera®.

Данное учебное пособие и рассматриваемый в нем проект `fir_filter` помогают быстро освоить работу с программным пакетом Quartus. Пособие хорошо структурировано и имеет модульную структуру, ориентированную на поэтапное освоение материала.

Материал представлен следующими разделами:

- В разделе **Ввод описания проекта** вы научитесь создавать файл блок-схемы проекта (**Block Design File**). Файл блок-схемы `*.bdf` создается с помощью графического редактора блок-схем (**Block Editor**). В этом разделе также создаются несколько текстовых файлов описания модулей проекта низших уровней иерархии на языке Verilog HDL. Файлы имеют расширение `*.v` и создаются в текстовом редакторе пакета Quartus с использованием встроенного мастера мегафункций **MegaWizard™**.
- Раздел **Компиляция** научит вас выполнять компиляцию проекта и управлять процессом компиляции с помощью соответствующих параметров. Вы также научитесь просматривать топологию проекта, показывающую созданные компилятором размещение проекта и его разводку на кристалле ПЛИС. Кроме того, будет показано, как выполнять назначения ресурсов реализации отдельных частей проекта.
- Раздел **Анализ временных характеристик проекта** научит вас анализировать характеристики быстродействия проекта, включая режим многотактового анализа.
- Раздел **Моделирование** научит вас создавать файлы `*.vwf` временных диаграмм (**Vector Waveform File**), содержащие векторы входных воздействий для моделирования. Вы также научитесь управлять моделированием и анализом временных характеристик с помощью соответствующих настроек.
- Раздел **Программирование** научит вас использовать программатор Quartus для конфигурирования микросхем Altera.

Файлы учебного пособия

При установке (**Installation**) пакета Quartus все файлы учебного пособия и рассмотренного в нем учебного проекта автоматически копируются на жесткий диск (винчестер).

Файлы учебного пособия, расположены в следующих подкаталогах корневого каталога пакета Quartus:

Название каталога:	Описание:
<code>\qdesigns</code>	Рабочий каталог Quartus.
<code>\qdesigns\tutorial</code>	Содержит все файлы этого учебного пособия. В каталоге находится файл <code>readme.txt</code> , описывающий текущие изменения и дополнения, сделанные в учебном пособии.

Название каталога:	Описание:
<code>\qdesigns\fir_filter</code>	Каталог, в котором вы будете создавать учебный проект fir_filter . Используйте этот каталог для предотвращения каких-либо изменений исходного набора рабочих файлов в каталоге <code>\qdesigns\tutorial</code> .

 На рабочих станциях платформы UNIX каталог `qdesigns` является подкаталогом `/usr`.

Сокращения команд

Многие команды Quartus имеют варианты сокращенного (быстрого) вызова (**shortcuts**). Однако в этом учебном пособии используются исходные команды, без сокращений. Для информации о доступных командах мыши, клавиатуры, панели инструментов обратитесь непосредственно к меню пакета Quartus, команде **Toolbar** (панель инструментов) в меню **Toolsmenu** (меню инструментальных средств) и непосредственно к справочной системе **Help** пакета Quartus.

Вызов справки

В данном учебном пособии ссылки на справки пакета Quartus обозначаются рисунком следов ().

В справочной системе содержится много самой новой и полной информации о пакете Quartus. Для быстрого получения справки удобно пользоваться контекстной справкой (**Context Sensitive Help**), а также поиском информации по ключевым словам (**Search Index**). Для поиска необходимой информации можно также воспользоваться содержанием (**Contents**) справочной системы **Help**.

1. Контекстная справка

Контекстная справка (**Context Sensitive Help**) позволяет моментально получить справочную информацию по необходимым вопросам.

Для получения контекстной справки можно использовать следующие методы:

Метод вызова:	Описание:
Клавиши Shift+F1	Нажмите клавиши Shift+F1 , укажите интересующий вас элемент на экране, ключевое слово в тексте или в командном меню и затем нажмите левую клавишу мыши.
Клавиша F1	Для получения справки по выделенной команде меню, открытому диалоговому окну, открытому или всплывающему окну сообщений, нажмите клавишу F1 . Нажатие клавиши F1 в активном окне подсистемы Quartus позволяет получить общую информацию по данной подсистеме, а также справку по относящимся к ней ключевым словам.

Метод вызова: Описание:

Команда Help Для получения справки по отображаемым в окне **Messages** сообщениям, выделите нужное сообщение (**message**), нажмите правую кнопку мыши и во всплывающем меню выберите команду **Help**.

2. Поиск тематической информации по ключевым словам

В справочной системе Quartus имеется указатель (**Index**) ключевых слов (**Keywords**), который удобно использовать для быстрого поиска тематической информации.

Для поиска тематических разделов (**Help topic**) выполните следующие действия.

1. В меню **Help** (справка) пакета Quartus выберите команду **Index** (ключевые слова). Если окно справочной системы Quartus уже открыто, можно сразу перейти на вкладку **Index** (рис.3.1).

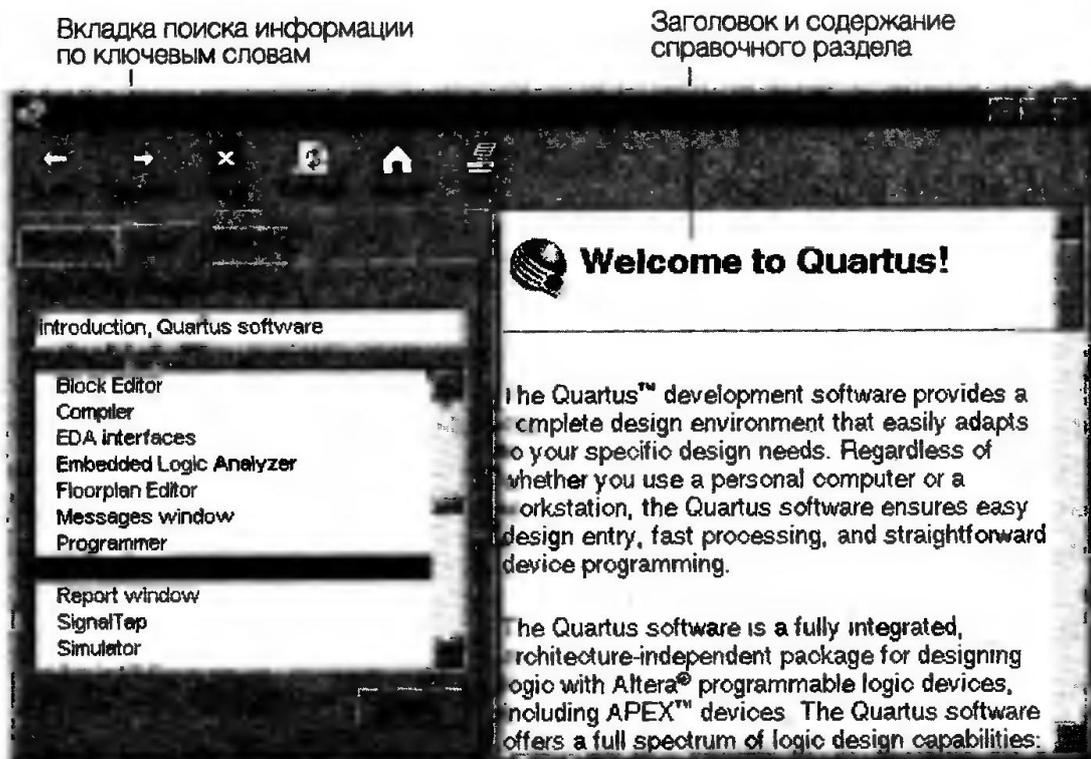


Рис. 3.1

2. В поле **Type in the keyword to find** введите искомое ключевое слово.
3. Для отображения раздела выделите его заголовок и нажмите кнопку **Display** (показать) или дважды щелкните левой кнопкой мыши на выделенном заголовке раздела.

3. Содержание справки

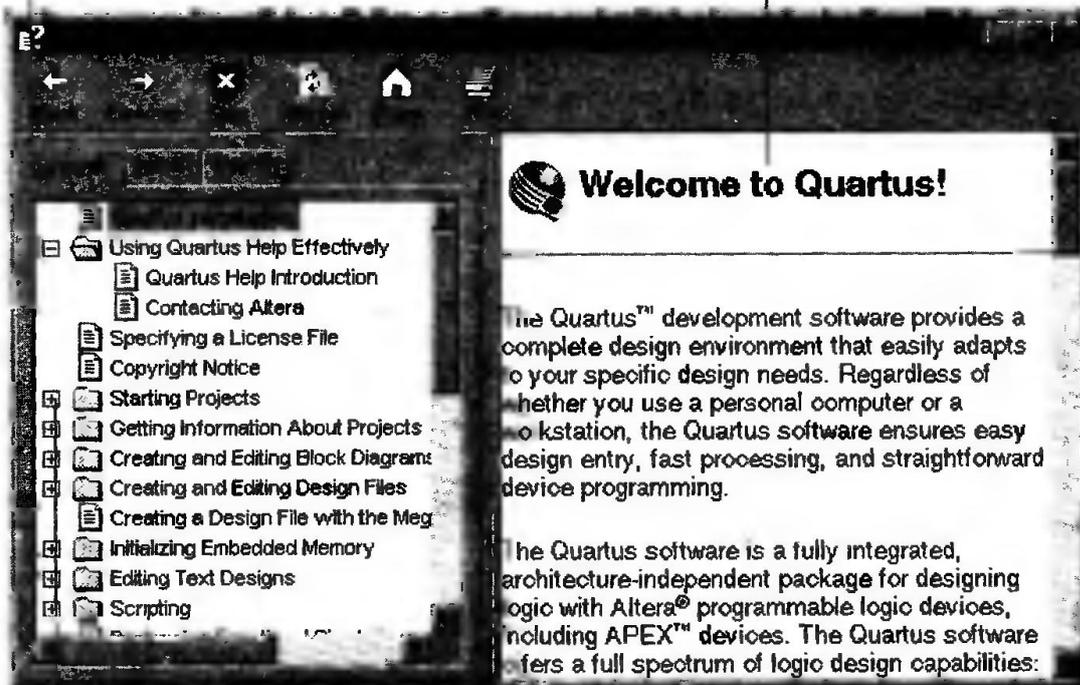
Справочная система пакета Quartus имеет содержание (**Contents List**), обеспечивающее поиск информации по разделам (**category**).

Чтобы просмотреть содержание справочного раздела, выполните следующие действия.

1. В меню **Help** (справка) окна **Quartus** выберите команду **Contents** (содержание). Если окно справочной системы Quartus уже открыто, вы можете сразу перейти на вкладку **Contents** (рис. 3.2).

Вкладка содержания справочной системы

Заголовок и содержание справочного раздела



Щелчок на символе "+" раскрывает содержание справочного раздела

Рис. 3.2

2. Для раскрытия оглавления нужного раздела щелкните значок "+".
3. Для просмотра тематической статьи (**topic**) щелкните на ее заголовке.

4. Поиск справочной информации во всем тексте справки

Справка пакета Quartus предоставляет возможность поиска информации во всем тексте (**Full-Text**) справки.

Для поиска справочной информации во всем тексте выполните следующие действия:

1. В меню **Help** (справка) окна **Quartus** выберите команду **Search** (поиск). Если окно **Help** уже открыто — перейдите на вкладку **Search** (рис. 3.3).
2. В поле **Type in the word(s) to search for** введите искомое слово (слова).
3. Для расширения или сужения границ поиска используются следующие опции:

- **Search previous results** (искать в найденном) — используется для ограничения поиска среди слов, найденных на предыдущем этапе;
 - **Match similar words** (искать похожие слова) — используется для поиска сходных слов;
 - **Search titles only** (искать только среди заголовков) — используется для ограничения области поиска по заголовкам
4. Для начала поиска нажмите **List Topics** (список статей) — в поле **Select topic** появятся заголовки тематических справочных статей, соответствующие критерию поиска.
 5. Для отображения статьи выберите ее имя и нажмите **Display** (показать) или дважды щелкните на выбранном имени правой кнопкой мыши.

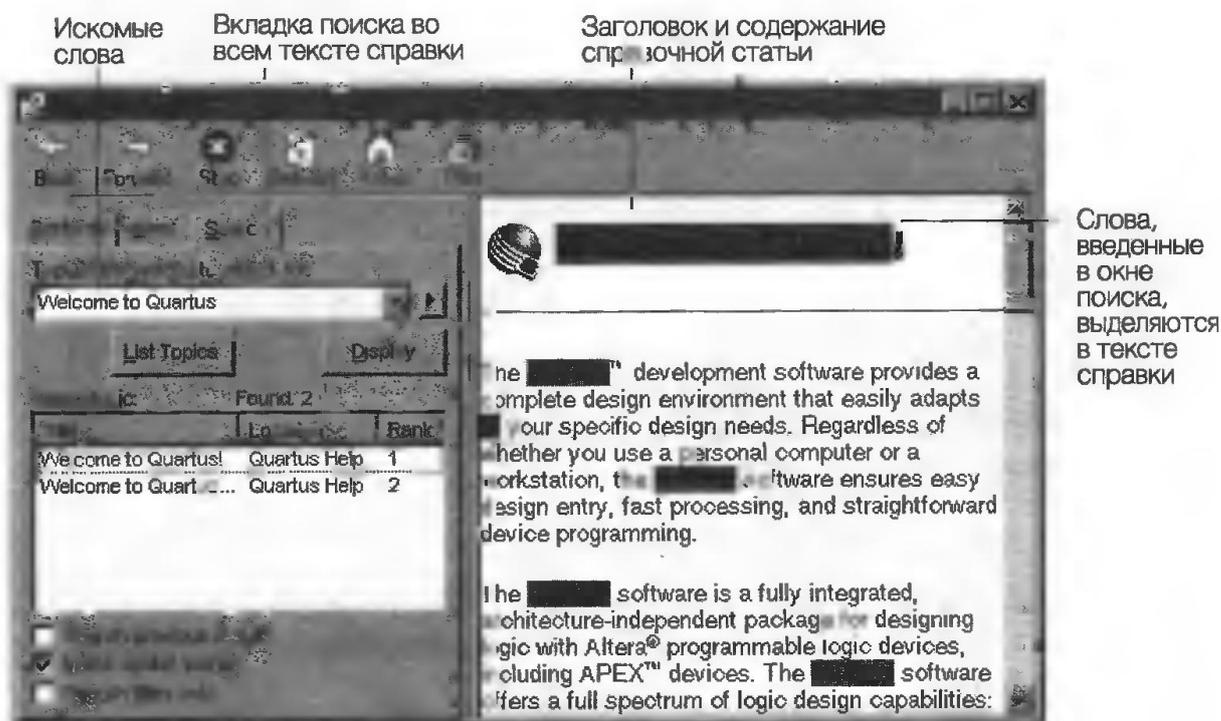


Рис. 3.3

3.2. Ввод описания проекта

Проект в пакете Quartus представляет собой полный набор файлов проекта, файлов назначений, файлов моделирования, системных установок и информации об иерархической структуре проекта. На примере учебного проекта, который вы создадите с помощью этого пособия, демонстрируется методология нисходящего проектирования.

В пяти следующих разделах пособия описываются этапы создания учебного проекта **fir_filter**. В начале создается файл ***.bdf (Block Design File)** верхнего уровня иерархии, содержащий блок-схему проекта. Блок-

схема проекта включает в себя блоки более низких уровней иерархии. Затем создаются текстовые файлы *.v (**Verilog Design Files**), описывающие блоки проекта на языке Verilog HDL. Описание функциональных блоков низших уровней иерархии будет также создано на основе готовых мегафункций (**Custom megafunction variation**).

Урок 1. Запуск пакета Quartus

В первом уроке вы запустите пакет Quartus и приступите к созданию вашего первого проекта.

 В этом учебном пособии предполагается, что рабочий каталог Quartus, имеющий по умолчанию имя *qdesigns*, расположен на диске **D:** вашего компьютера. Если вы установили рабочий каталог Quartus в другое место или/и под другим именем — измените соответствующий путь или имя рабочего каталога.

Для запуска пакета Quartus выполните следующие действия:

1. В меню **Start** операционной системы Windows выберите **Programs > Altera > Quartus**.

Для запуска из командной строки введите `quartus` и нажмите клавишу **Enter** (ввод) — откроется окно пакета Quartus.

2. Разверните окно пакета **Quartus** на весь экран (**Maximize**) нажатием соответствующей стандартной кнопки окна (рис. 3.4).



Рис. 3.4

Урок 2. Определение имени проекта и его основных параметров

Для облегчения создания нового проекта пакет Quartus имеет встроенный мастер создания проекта (**New Project Wizard**). Для создания проекта с помощью мастера **New Project**, выполните следующие действия.

1. В меню **File** (операции с файлами) выберите **New Project Wizard** (мастер создания проекта). Откроется окно **New Project** (новый проект). При первом запуске окна **New Project**, отображается окно заставки. Для перехода к окну первого этапа создания проекта нажмите кнопку **Next** (следующий). В окне первого этапа создания проекта указывается имя рабочего каталога проекта, текущее имя проекта и имя файла верхнего уровня иерархии (рис. 3.5).

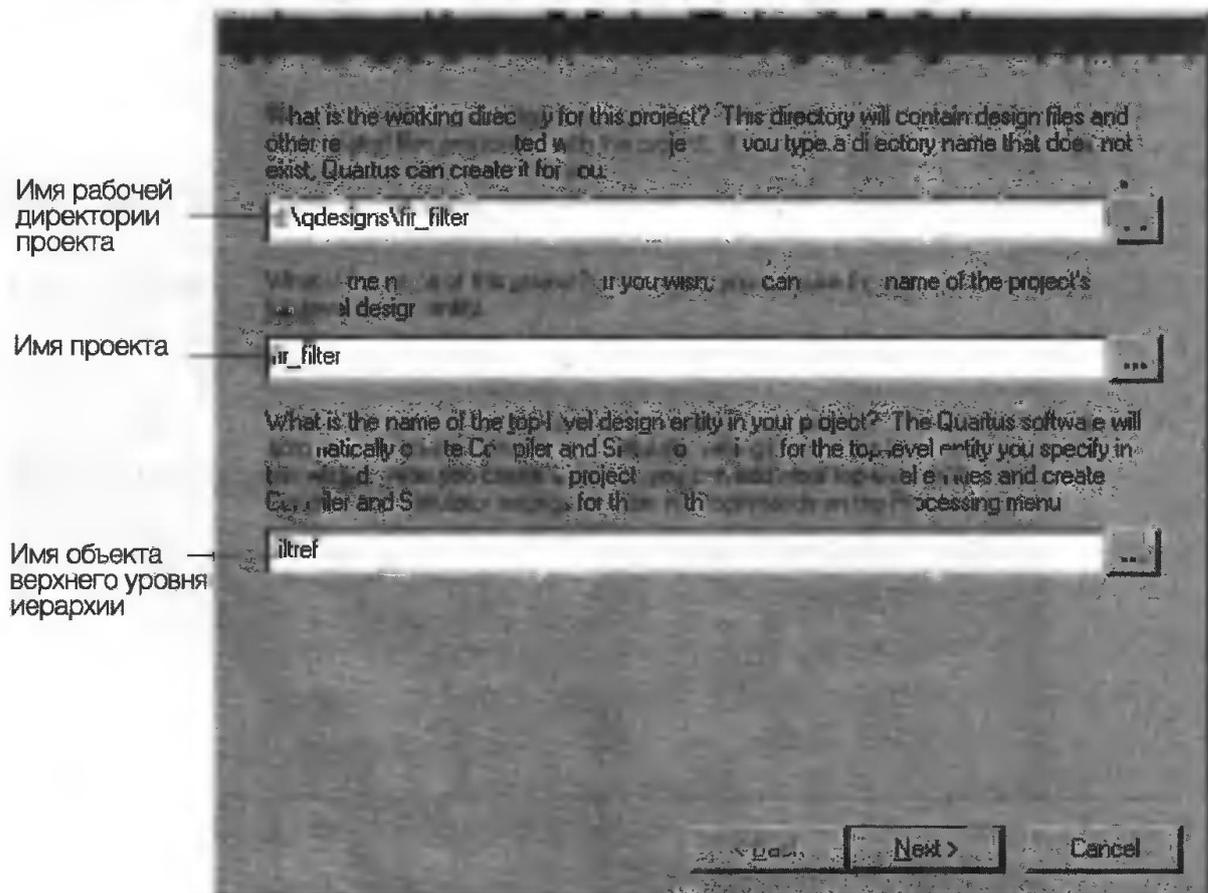


Рис. 3.5

Комментарии:

Фраза над окном каталога

Как называется рабочий каталог проекта? Этот каталог должен содержать файлы проекта и другие, связанные с проектом файлы. Если вы введете имя несуществующего каталога, Quartus создаст его автоматически.

Фраза над окном имени проекта	Какое имя этого проекта? Вы можете использовать имя файла верхнего уровня иерархии.
Фраза над окном имени высшего блока иерархии проекта	Какое имя блока верхнего уровня иерархии проекта? Quartus автоматически создает установки компиляции и моделирования для блока, указанного в этом окне. После создания проекта вы сможете добавить другие объекты верхних уровней иерархии и создать для них установки компиляции и моделирования при помощи меню Processing .

2. В строке ввода рабочего каталога проекта (верхняя строка ввода) введите имя каталога или выберите его, используя кнопку обзора **Browse (...)**. В данном примере вам необходимо задать каталог `d:\qdesigns\fir_filter`.
3. В строке ввода имени проекта (средняя строка ввода) введите имя проекта. В этом примере необходимо задать имя `fir_filter`.
4. В строке ввода имени блока верхнего уровня иерархии (нижняя строка ввода) введите `filtref`.

 *Заданное имя проекта по умолчанию присваивается блоку верхнего уровня иерархии. Однако можно задать и имя, отличное от имени блока верхнего уровня иерархии.*

5. Нажмите кнопку **Next** (следующий) – появится окно второго этапа создания нового проекта (рис. 3.6).

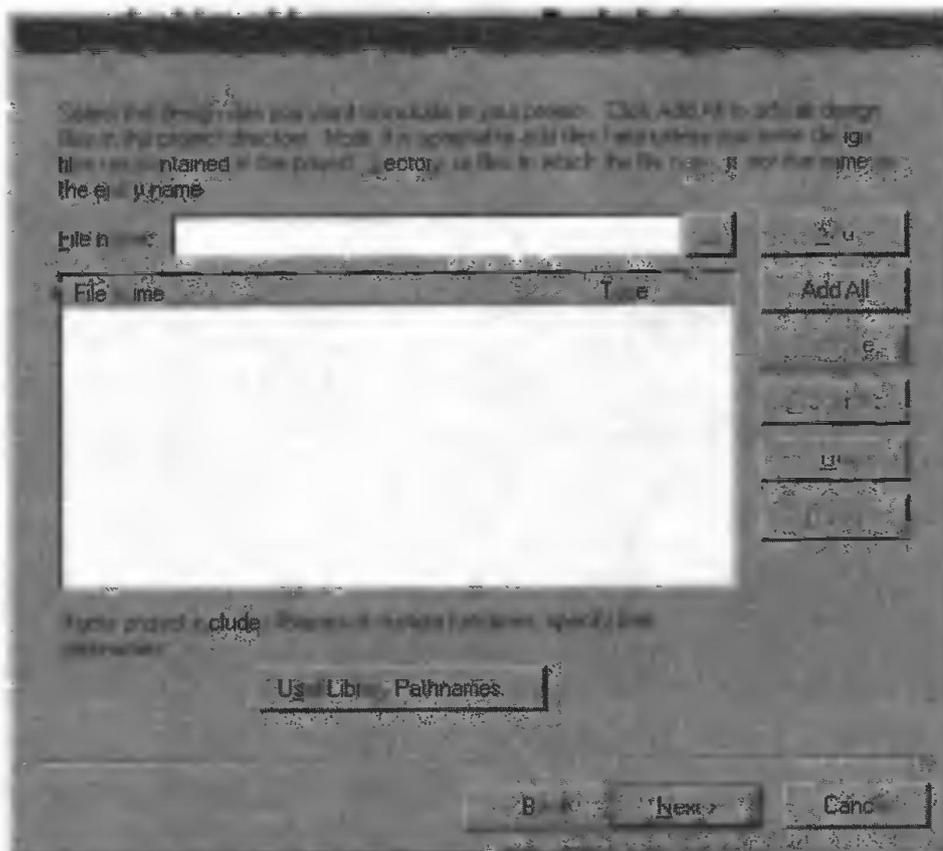


Рис. 3.6

Комментарии:

Надпись над окном	Выберите файлы, которые вы хотите включить в свой проект. Нажмите кнопку Add All , чтобы добавить в рабочий каталог проекта все файлы. Предупреждение: если вы не имеете файлов проекта в других каталогах или файлы, имя которых отлично от имени проекта, то добавлять файлы в этом окне не обязательно.
Подпись под окном	Если ваш проект включает библиотеки специализированных функций, укажите их пути.

6. Для добавления всех файлов проекта в рабочий каталог проекта нажмите **Add All** (добавить все файлы). На данном этапе список файлов пуст, поскольку проект **fir_filter** — новый и для него еще не создано файлов.

 Если вы хотите добавить к новому проекту существующие файлы или другие источники, то для их выбора вы можете нажать кнопку обзора **Browse (...)**, выбрать нужные файлы и добавить их, нажав кнопку **Add** (добавить).

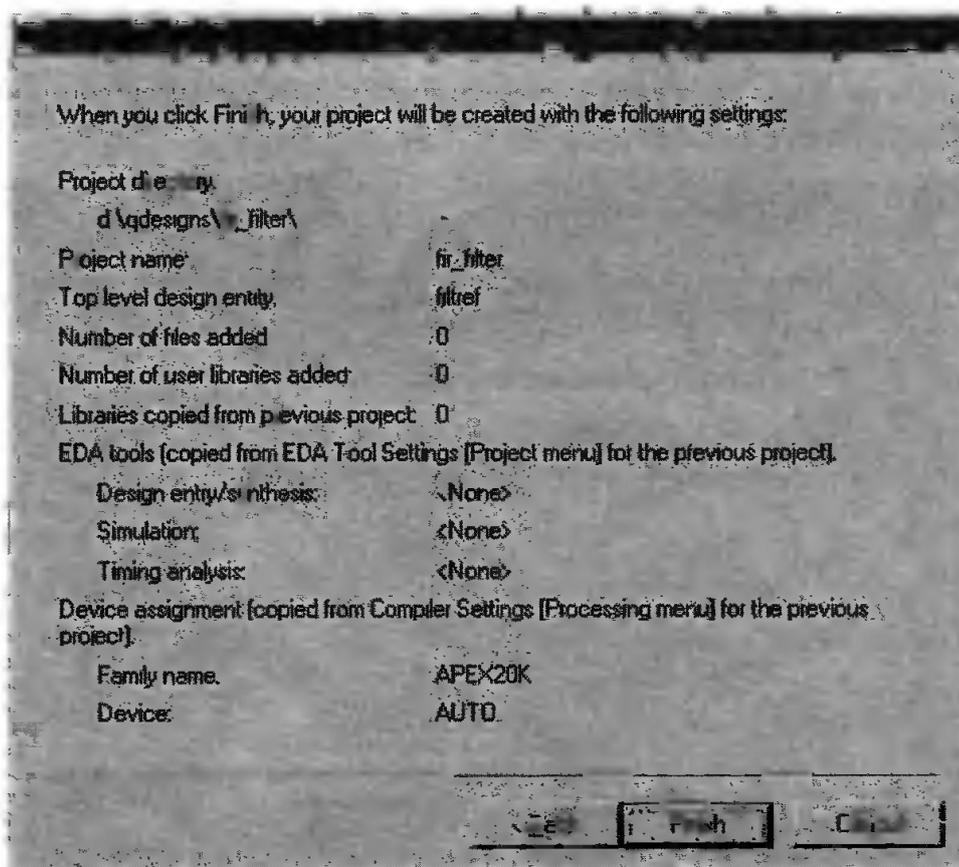


Рис. 3.7

7. Нажмите кнопку **Next** (следующий) — появится сводное окно заданных вами параметров проекта. В этом окне отображаются установки, заданные в мастере **New Project**: рабочий каталог, имя

проекта, блок верхнего уровня иерархии и ряд служебных файлов проекта таких, как информация о пользовательских библиотеках, используемых средствах электронных САПР (EDA), выбранном семействе и типе ПЛИС и др. (рис. 3.7).

8. Нажмите кнопку **Finish** (готово) – новый проект создан. Имя блока верхнего уровня иерархии появится в строке **Hierarchies** (иерархия) окна **Project Navigator** (навигатор проекта) (рис. 3.8).



Рис. 3.8

Урок 3. Создание блок-схемы

В этом разделе описываются этапы создания блок-схемы верхнего уровня иерархии.

Запустите файл **filtref.bdf**.

 Если вами уже освоена работа в редакторе блок-схем (**Block Editor**), вы можете пропустить этот раздел и сэкономить свое время, скопировав следующие готовые файлы из подкаталога `\qdesigns\tutorial` в подкаталог `\qdesigns\fir_filter`.

Имя файла:	Описание:
filtref.bdf	Файл блок-схемы верхнего уровня иерархии (Block Design File).
mult.v	Текстовый файл описания умножителя на языке Verilog HDL, созданный на базе готовой библиотечной мегафункции <code>lpm_mult</code> с помощью встроенного мастера MegaWizard . На блок-схеме filtref.bdf этот файл представлен в виде блока mult .

Рекомендуется копировать файлы учебного пособия следующим образом: открыть файл в пакете Quartus с помощью команды **Open** (открыть) меню **File**, затем выбрать команду **Save As** (сохранить файл как ...) меню **File**, в появившемся окне установить ключ **Add file to current project** (добавить файл к текущему проекту), а затем сохранить файл в подкаталоге `\qdesigns\fir_filter`.

Скопировав предлагаемые готовые файлы, вы можете сразу перейти к уроку 4: "Создание описания на языке Verilog HDL".

В этом уроке описываются следующие этапы проектирования:

- создание графического файла ***.bdf (Block Design File)** блок-схемы проекта;
- создание функционального блока;
- ввод примитивов и мегафункций;
- настройка и изменение свойств экрана;
- программы работы с блоками;
- ввод символов входных/выходных выводов;
- присвоение имен входным/выходным выводам;
- соединение графических обозначений и блоков;
- распределение сигналов между блоками.

1. Создание файла блок-схемы проекта (Block Design File)

В этом подразделе вы создадите новый графический файл ***.bdf (Block Design File)** блок-схемы проекта с именем **filtref.bdf**. Этот файл является файлом верхнего уровня иерархии проекта **fir_filter**.

Для создания нового BDF-файла выполните следующие действия:

1. В меню **File** (операции с файлами) выберите команду **New** (создать новый файл) – автоматически появится вкладка **Design Files** (файлы проекта).
2. На вкладке **Design Files** (файлы проекта) выберите **Block Diagram/Schematic File** (файлы блок-схемы/схемотехнические файлы).
3. Нажмите кнопку **OK** – откроется окно редактора блок-схем (**Block Editor**).
4. В меню **File** (операции с файлами) выберите команду **Save As** (сохранить файл как).
5. Выберите каталог, где вы хотите сохранить BDF-файл. В окне **Save As** автоматически выделен рабочий каталог проекта **d:\qdesigns\fir_filter**.
6. Если необходимо – введите имя **filtref** в поле **File name** (имя файла).
7. Если необходимо – включите опцию **Add file to current project** (добавить файл к текущему проекту).
8. Нажмите кнопку **Save** (сохранить) – теперь файл сохранен и добавлен к проекту.

2. Создание функционального блока

Для создания нового блока (файл **filtref.bdf**) выполните следующие действия:

1. На панели инструментов редактора блок-схем (**Block Editor**) нажмите клавишу **Block Tool** (рисование блока). Кнопки панели инструментов показаны на рис. 3.9.

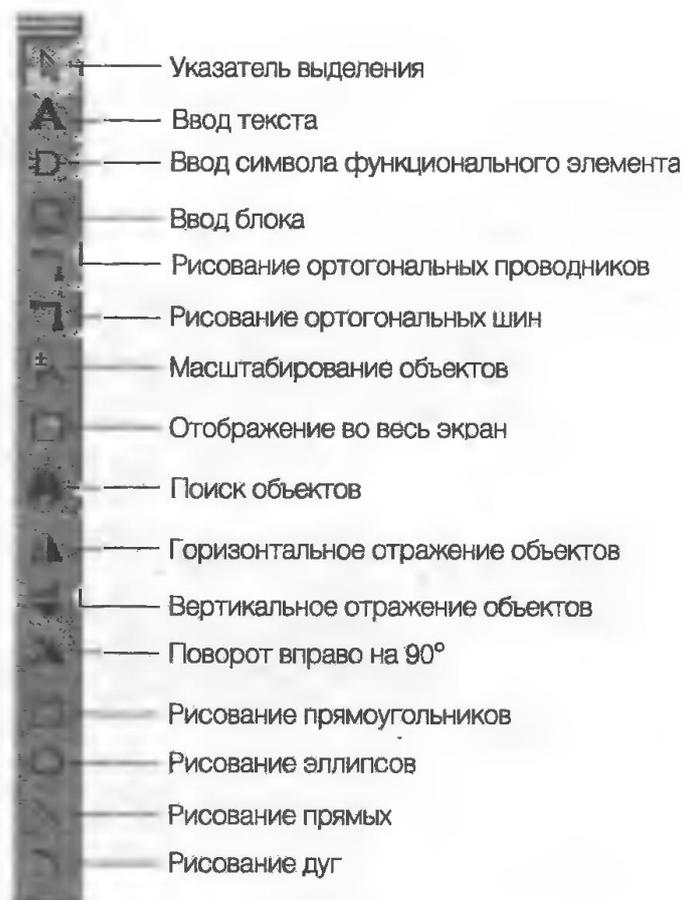


Рис. 3.9

2. Нажмите на белом поле в области редактора блок-схем (**Block Editor**) левую кнопку мыши и, удерживая ее, выделите прямоугольную область, соответствующую требуемому размеру создаваемого блока.

 Для коррекции своих действий вы можете использовать команды **Undo** (отмена действия) и **Redo** (повтор действия) в меню **Edit** (редактирование). Вид созданного блока в редакторе блок-схем (**Block Editor**) показан на рис. 3.10.



Рис. 3.10

3. Нажмите клавишу указателя выбора (**Selection Tool**). На панели инструментов редактора **Block Editor** указатель **Selection Tool** выглядит в виде стрелки.
4. Наведите указатель на созданный блок и дважды щелкните левой кнопкой мыши — откроется диалоговое окно **Block Properties** (свойства блока).
5. Перейдите на вкладку **General** (основные) (рис. 3.11).

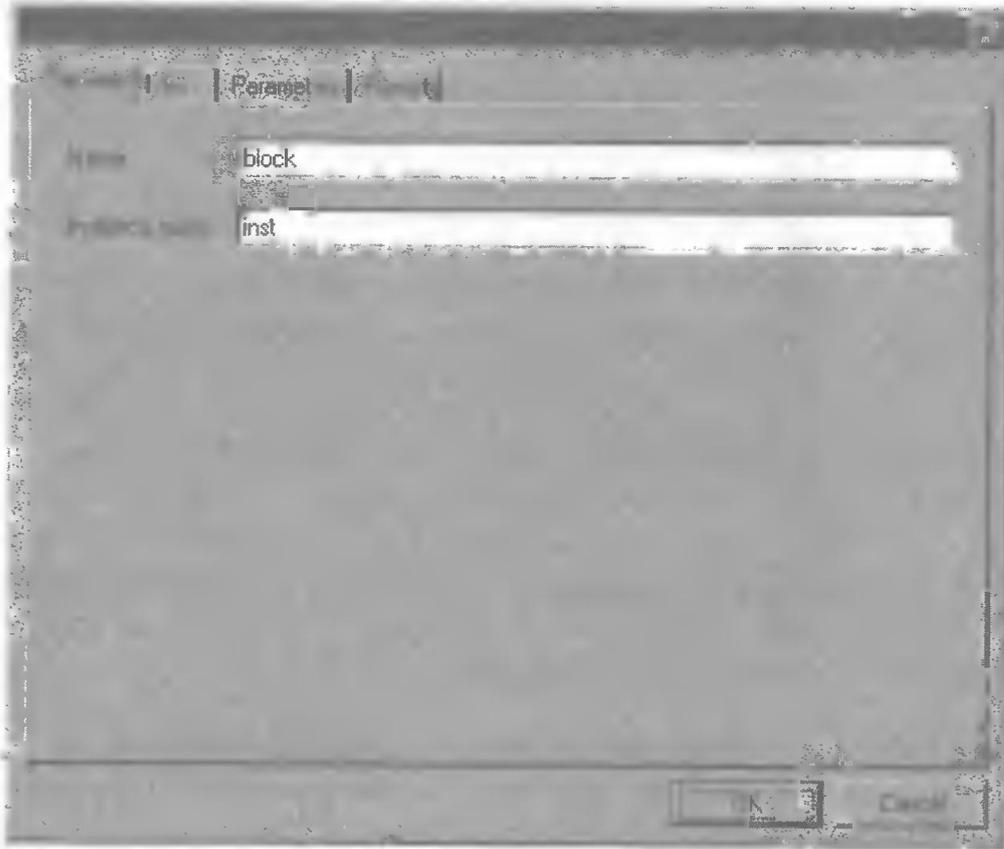


Рис. 3.11

6. В поле **Name** (имя) введите имя блока — taps и оставьте без изменений имя inst.
7. Откройте вкладку **I/Os** (входы/выходы).
8. Под надписью **I/O** (входы/выходы) в поле **Name** (имя) введите имя первого вывода — clk, а в поле **Type** (тип выводов) выберите **INPUT** (вход).
9. Нажмите кнопку **Add** (добавить) — вывод с именем clk появится в списке выводов блока **Existing block I/Os** (существующие входы/выходы блока) (рис. 3.12).
10. Для присвоения имени каждому из выводов, приведенных ниже, повторите действия 8 и 9:



*Вы также можете ввести в строке имена всех входных выводов через запятую, а затем нажать кнопку **Add** (добавить).*

Имя:	Тип:
clk (уже введено)	INPUT
Reset	INPUT
sel[1..0]	INPUT
newt	INPUT
d[7..0]	INPUT
x[7..0]	OUTPUT

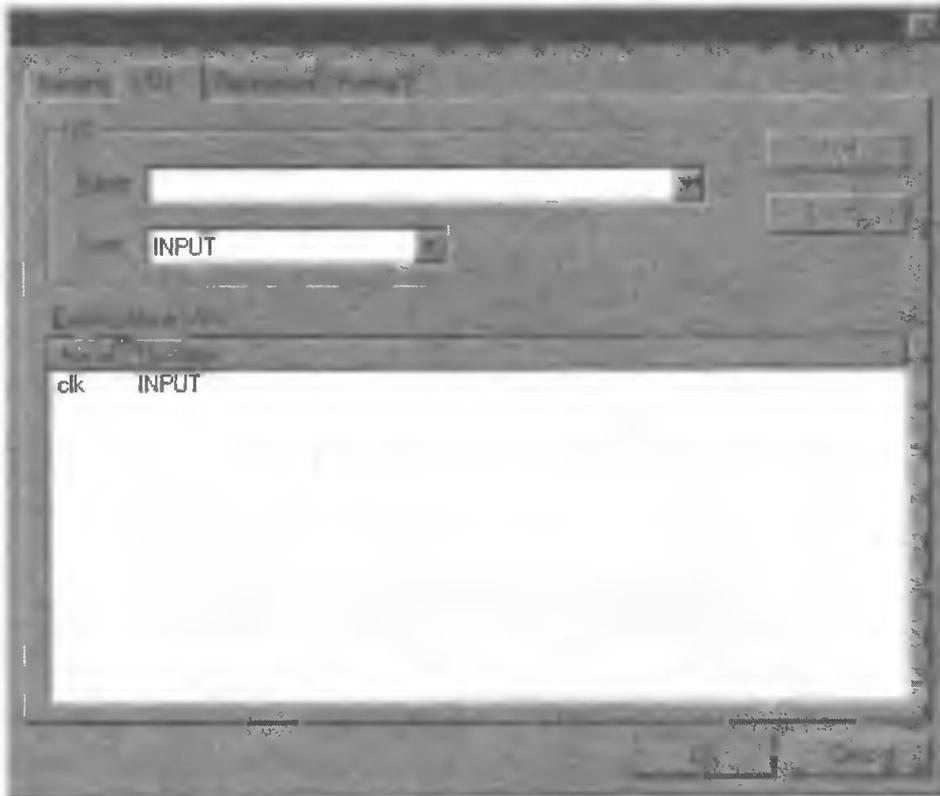


Рис. 3.12

11. Нажмите кнопку **ОК**. Теперь вы определили входные и выходные выводы блока taps. Блок taps входит в блок-схему проекта, создание которой описывается в уроке 4.
12. Выберите блок taps и нажмите правую кнопку мыши.
13. Во всплывающем меню выберите команду **AutoFit**. Эта команда устанавливает размер рамки блока taps по таблице портов входа/выхода, чтобы вся информация о блоке была хорошо видна (рис. 3.13).
14. Для создания трех следующих блоков: state_m, hvalues и ass повторите действия 1–13. Порты блоков приведены ниже.

Порты блока hvalues.

Имя:	Тип:
sel[1..0]	INPUT
h[2..0]	OUTPUT

Порты блока acc.

Имя:	Тип:
xh[10..0]	INPUT
clk	INPUT
first	INPUT
yn[7..0]	OUTPUT

Порты блока state_m.

Имя:	Тип:
clk	INPUT
reset	INPUT
newt	INPUT
sel[1..0]	OUTPUT
next	OUTPUT
First	OUTPUT

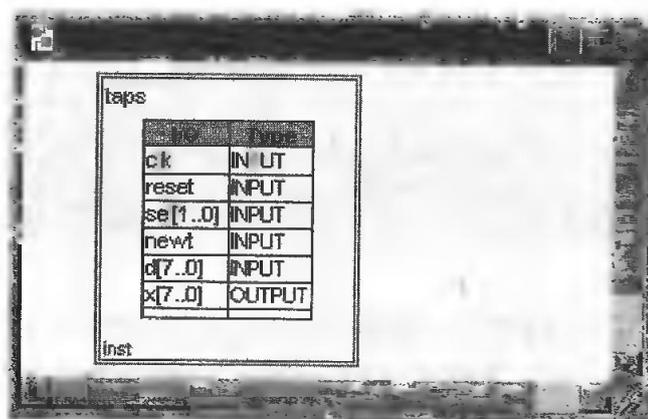


Рис. 3.13

- Для перемещения любого из блоков на новое место на панели инструментов редактора **Block Editor** выберите **Selection Tool** (указатель выделения) или нажмите клавишу **Esc**. Наведите указатель на нужный блок, затем нажмите левую кнопку мыши и, удерживая ее, перетащите блок на новое место. После окончания создания четырех блоков, расположите их примерно так, как показано на рис. 3.14.
- Для сохранения изменений в файле проекта в меню **File** (операции с файлами) выберите команду **Save** (сохранить файл).

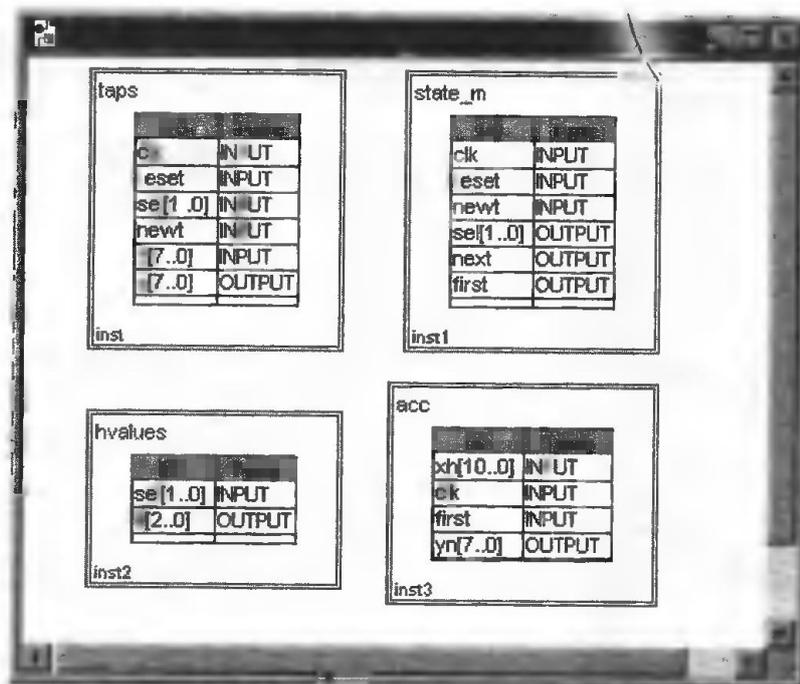


Рис. 3.14

3. Создание графических обозначений примитива и мегафункции

Помимо блоков, созданных в этом примере, файл *.bdf (Block Design File) блок-схемы проекта может содержать графические обозначения других функциональных узлов, включая примитивы, библиотеки параметризованных модулей (Library of Parameterized Modules — LPM) и другие мегафункции (megafunctions). Все эти элементы вы можете использовать при создании блок-схем в редакторе Block Editor.

Для добавления в схему проекта (файл filtref.bdf) D-триггера (примитив DFF) выполните следующие действия:

1. Дважды щелкните левой кнопкой мыши на пустом пространстве в окне редактора Block Editor.
2. В появившемся диалоговом окне Symbol (графические обозначения), в списке Libraries (библиотеки), откройте раздел библиотек d:\Quartus\libraries, щелкнув символ "+". Откройте подразделы primitives (примитивы) и storage (элементы памяти).
3. В подразделе storage выберите примитив dff (D-триггер). графическое обозначение выбранного примитива появится в правом окне (рис. 3.15).

 Вы также можете ввести имя требуемого примитива (dff) непосредственно с клавиатуры в поле Name (имя).

4. Нажмите кнопку ОК — графическое обозначение примитива DFF будет привязано к указателю мыши.

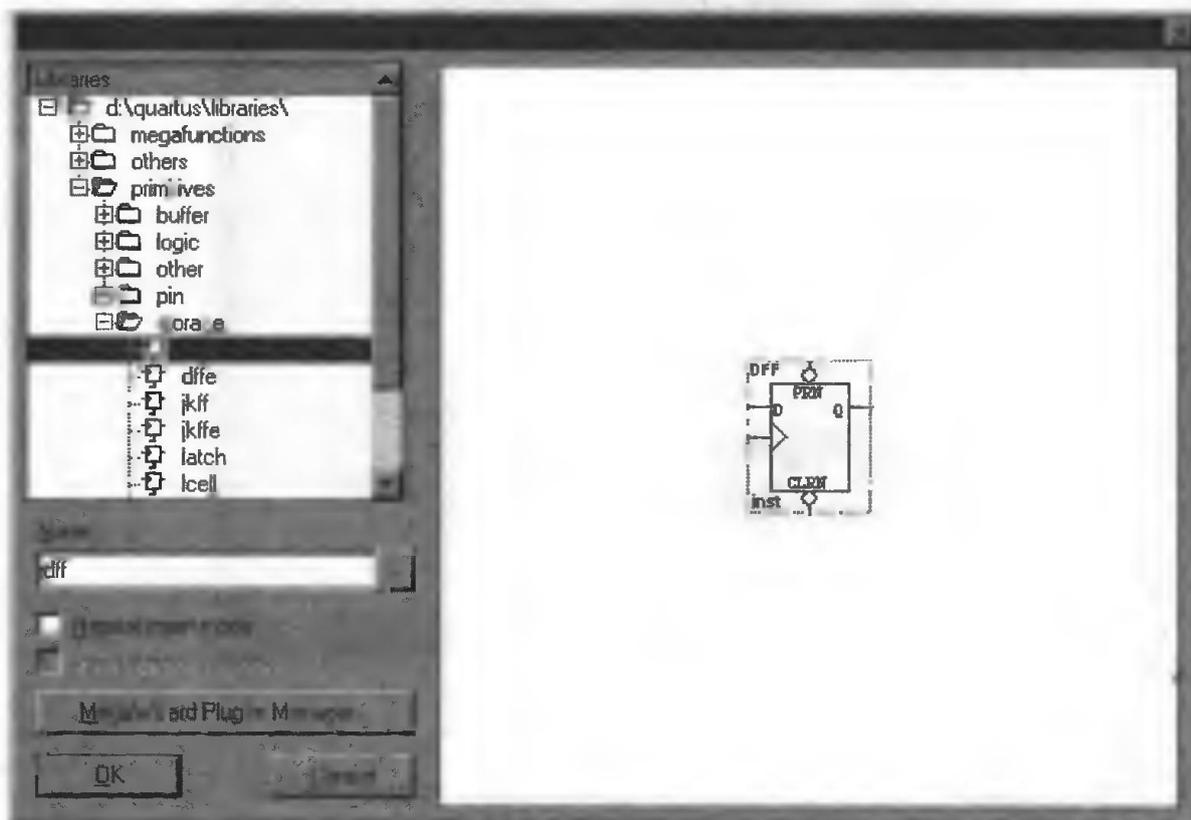


Рис. 3.15

5. Поместите символ DFF на нужное место в окне редактора **Block Editor** и щелкните левой кнопкой мыши. Теперь примитив DFF добавлен к блок-схеме проекта.
6. Чтобы ввести в файл **filtref.bdf** символ DFFE (D-триггер со входом разрешения записи), повторите действия 1–5.

Следующий функциональный элемент, который необходимо ввести, — умножитель. Он будет создан на базе (как экземпляр) мегафункции `lpm_mult`. Для создания умножителя вы можете использовать встроенный мастер мегафункций **MegaWizard**. Мастер **MegaWizard** обеспечивает создание и редактирование файлов функциональных узлов проекта, на основе специализированных готовых мегафункций, включая функции библиотек параметризованных модулей (**LPM**). Мастер **MegaWizard** помогает легко и просто настроить параметры вашего узла. При создании очередного функционального узла мастер **MegaWizard** предлагает выбрать значения его параметров и свойства входных/выходных портов. После создания умножителя с помощью мастера **MegaWizard** вы сможете использовать его в файле проекта.

Для создания описания умножителя `mult` с помощью мастера **MegaWizard** выполните следующие действия:

1. Дважды щелкните левой кнопкой мыши на свободном месте поля редактора **Block Editor** (редактор блок-схем).

- В диалоговом окне **Symbol** щелкните по надписи **MegaWizard Plug-In Manager**. Откроется первая страница мастера **MegaWizard**, показанная на рис. 3.16.

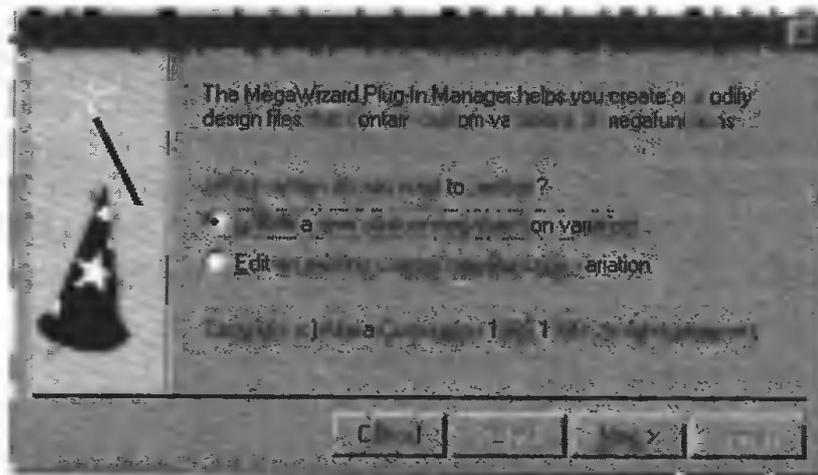


Рис. 3.16

Комментарии:

Верхняя надпись: Мастер MegaWizard помогает вам создавать или дорабатывать файлы проекта на основе мегафункций (файлы, представляющие собой настроенные экземпляры мегафункций).

Надпись над ключами: Какое действие вы хотите выполнить?

Верхний ключ: Создать новый вариант мегафункции.

Нижний ключ: Редактировать существующий вариант мегафункции.

- Выберите **Create a new custom megafunction variation** (создать новый вариант мегафункции) и нажмите кнопку **Next** (следующий шаг).
- В списке **Available Megafunctions** (доступные мегафункции) нажмите значок "+", чтобы раскрыть каталог **arithmetic** (арифметические мегафункции), затем выберите **LPM_MULT** (мегафункция умножителя).
- На вопрос **Which type of output file do you want to create?** (какой тип выходного файла вы хотите создать?) выберите **Verilog HDL** (файл текстового описания на языке Verilog HDL).
- На вопрос **What name do you want for the output file?** (какое имя вы хотите присвоить выходному файлу?) введите **d:\qdesigns\fir_filter\mult.v** и нажмите кнопку **Next**.
- На вопрос **How wide should the 'dataa' input bus be?** (какой разрядности должна быть входная шина **dataa**?) введите 8.
- На вопрос **How wide should the 'datab' input bus be?** (какой разрядности должна быть входная шина **datab**?) введите 3.

9. Для принятия по умолчанию всех остальных вопросов и создания графического обозначения (**symbol**) блока нажмите кнопку **Finish**, графическое обозначение созданного блока показывается в правом окне **Symbol**.
10. Нажмите кнопку **OK**. Графическое обозначение (символ) умножителя **mult** будет прикреплено к указателю мыши.
11. Чтобы разместить графическое обозначение в поле редактора блок-схем **Block Editor**, щелкните на рабочем поле окна редактора.
12. Чтобы сохранить изменения в файле проекта в меню **File** (операции с файлами), выполните команду **Save** (сохранить).

Текущий вид создаваемой в редакторе **Block Editor** блок-схемы проекта показан на рис. 3.17.

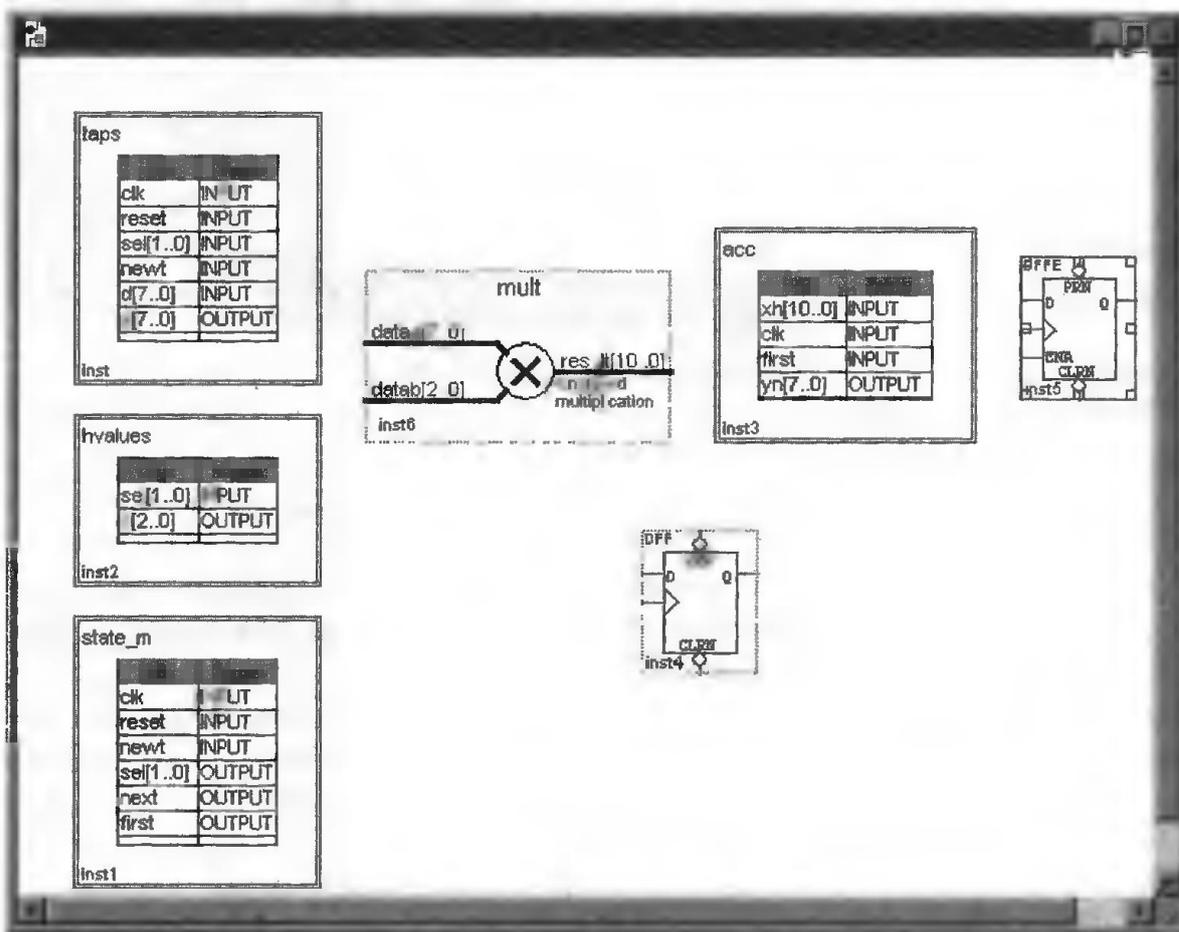


Рис. 3.17

4. Настройка и изменение свойств экрана

При необходимости вы можете изменить параметры окна **Block Editor**. Для изменения настроек экрана выполните следующие действия:

1. В меню **Tools** (инструментальные средства) выберите команду **Options**.

2. В списке **Category** (разделы), под строкой **Block/Symbol Editor** (свойства редактора блок-схем/графических обозначений), выберите раздел **General** (основные) (рис. 3.18).



Рис. 3.18

Комментарии:

Show guidelines

Показывать направляющие

Guideline spacing

Шаг направляющих

Snap to grid

Привязать к сетке

Use rubberbanding

Использовать "резиновые" связи

Show parameter assignments

Показывать значения параметров

Show block I/O tables

Показывать таблицы входов-выходов блока

Show mapper tables

Показывать отображение таблиц

3. В разделе **General** (основные) задайте необходимые вам установки.
4. Для изменения цветов и шрифтов редактора **Block Editor**, выберите **Colors** (цвета) или **Fonts** (шрифты) в списке **Category**.
5. Нажмите кнопку **OK**.

Уменьшение или увеличение изображения осуществляется с помощью инструмента масштабирования **Zoom** (кнопка **Zoom Tool** на панели инструментов **Block Editor**), а также с помощью команд **Zoom**, **Zoom In** (увеличить), **Zoom Out** (уменьшить) и **Fit in Window** (вписать изображение в окно) в меню **View** (вид).

5. Создание обозначений входных/выходных выводов

Для создания обозначений выводов выполните следующие действия:

1. Нажмите клавишу **Symbol Tool** на панели **Block Editor**. Появится диалоговое окно **Symbol**, которое вы использовали для ввода функциональных элементов DFF и DFFE. Заметим, что при открытии этого окна с помощью кнопки панели инструментов будет включен режим **Repeat-insert mode** (режим повторной вставки элементов).



*При использовании режима **Repeat-insert mode** (режим повторной вставки) обозначение выбранного функционального элемента сразу привязывается к указателю мыши. Это упрощает размещение на экране нескольких копий обозначения функционального элемента. Для завершения вставки копий элемента необходимо нажать клавишу **Esc** или нажать правую кнопку мыши и в появившемся всплывающем меню выбрать команду **Cancel** (отмена).*

2. Для раскрытия в списке **Libraries** (библиотеки) диалогового окна **Symbol** раздел **d:\Quartus\libraries**, щелкните значок "+". Раскройте также раздел **primitives** (примитивы), а затем — раздел **pin** (выводы).
3. В разделе **pin** (выводы) выберите примитив **input** (вход).
4. Нажмите кнопку **OK**.
5. Для создания пяти обозначений входов Input щелкните левой кнопкой мыши пять раз подряд на пустом месте в левой части рабочего поля. Выводам автоматически будут присвоены имена вида **pin_name<number>** с номерами в порядке возрастания. Нажмите кнопку **Esc**.
6. Повторите действия 1–5 для создания и расположения обозначений трех выходов (Output) в правой части рабочего поля (рис. 3.19).
7. Выберите команду **Save** (сохранить) в меню **File** (операции с файлами).

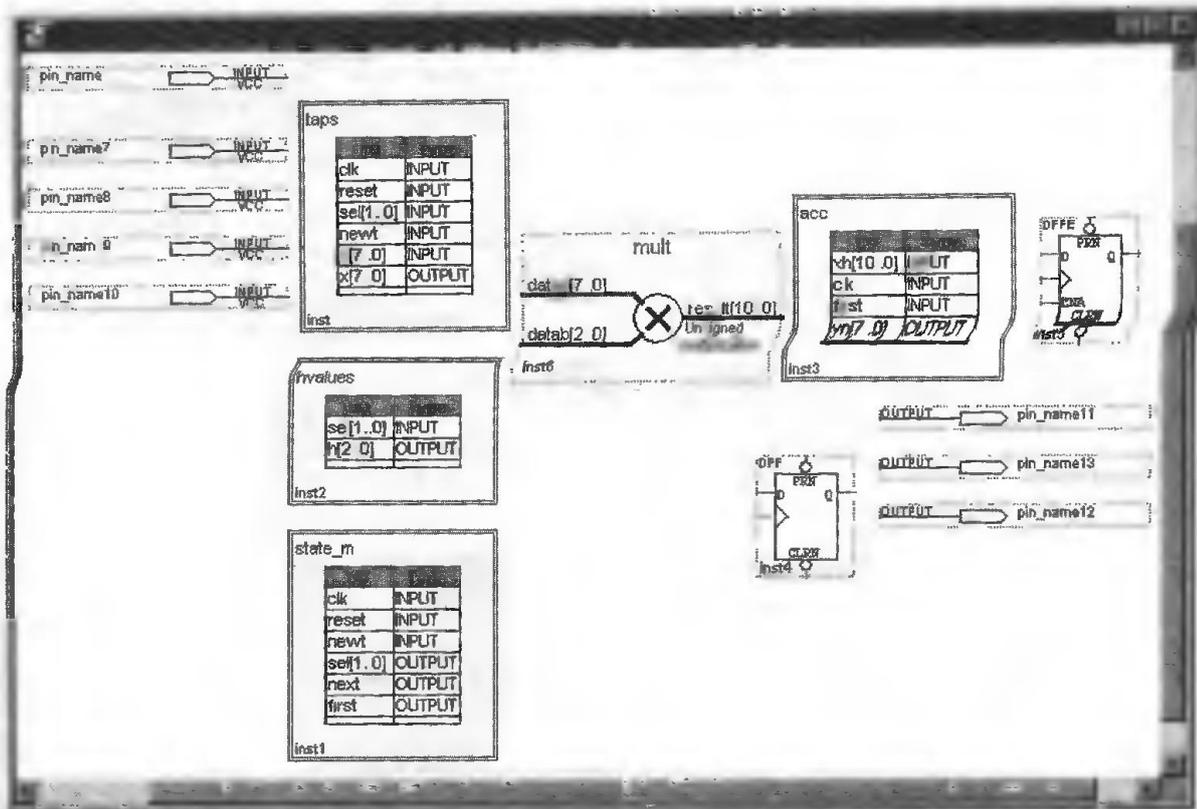


Рис. 3.19

6. Присвоение имен выводам блоков

Теперь созданным выводам необходимо присвоить имена. Для назначения имен выполните следующие действия:

1. Наведите указатель на первый входной вывод и дважды щелкните левой кнопкой мыши. В диалоговом окне **Pin Properties** (свойства выводов) откроется вкладка **General** (основные) (рис. 3.20).
2. Переименуйте выводы, заменив присвоенное по умолчанию имя первого вывода (pin_name), введя в строке **Pin name(s)** (имена вывода(ов)) имя clkx2.
3. Нажмите кнопку **OK**.
4. Для переименования всех выводов согласно таблице 3.1 повторите действия 1–3.

Таблица 3.1

Тип вывода	Назначаемое имя	Описание
INPUT	clkx2 (уже введен)	Вторичный тактовый сигнал проекта FIR filter .
INPUT	clk	Основной (первичный) тактовый сигнал проекта FIR filter .
INPUT	d[7..0]	Входная шина данных проекта FIR filter .

Окончание табл. 3.1

Тип вывода	Назначаемое имя	Описание
INPUT	reset	Сигнал сброса проекта FIR filter .
INPUT	newt	Входной сигнал записи данных $d[7..0]$ в блок taps.
OUTPUT	yn_out[7..0]	Выходная шина данных проекта FIR filter .
OUTPUT	yvalid	Сигнал подтверждения нормального состояния выхода $yn[7..0]$ функции асс.
OUTPUT	next	Сигнал готовности FIR filter к приему следующего байта (8-бит) данных.

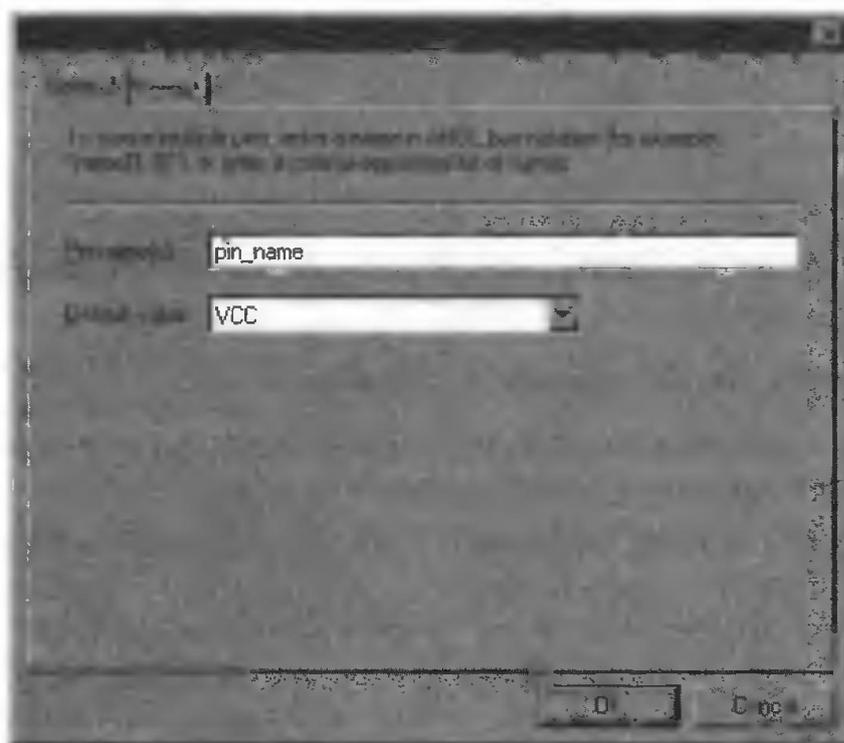


Рис. 3.20

Комментарий

Надпись над окнами

Для создания многоразрядных выводов введите имя в формате шины AHDL, например, $name[3..0]$ или список выводов, разделенных запятой.

5. Разместите графические обозначения (**Symbols**) всех выводов Input (вход) и Output (выход) так, чтобы они совпали с соответствующими обозначениями других функциональных элементов или блоков схемы (рис. 3.21).
6. Сохраните файл блок-схемы, выбрав команду **Save** (сохранить) в меню **File** (операции с файлами).

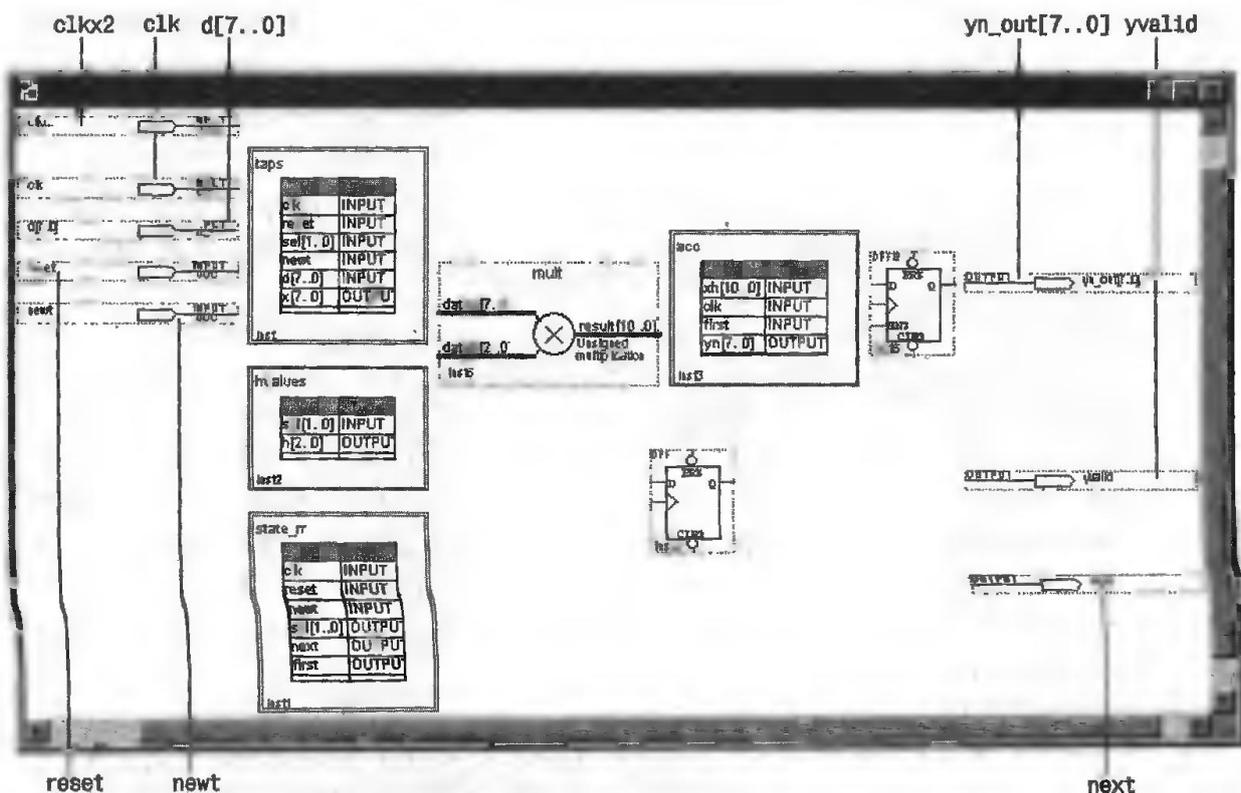


Рис. 3.21

7. Соединение функциональных элементов и блоков

Связи, соединяющие функциональные элементы и блоки, включают **Nodes** (проводники, сигнальные линии), **Buses** (шины) и **Conduits** (каналы). **Conduits** (каналы) – это шины, включающие один или более сигналов.

Большинство необходимых линий соединения функциональных элементов (**Symbols**) и блоков (**Blocks**) на блок-схеме (графическом файле BDF) можно нарисовать с помощью автоматически изменяемого указателя "smart" **Selection Tool**.

Указатель "smart" **Selection Tool** автоматически переключается на указатель инструмента рисования необходимых линий при наведении его на обозначение вывода или на границу блока. Например, при наведении на обозначение вывода, автоматически появляется указатель рисования ортогональных линий (**Orthogonal Node Tool**), а при наведении на границу блока – указатель инструмента рисования ортогональных шин (**Orthogonal Bus Tool**). Вы также можете самостоятельно выбирать нужные инструменты рисования ортогональных шин и линий на соответствующей панели инструментов.

Система Quartus автоматически создает связи между соединенными блоками, а также между сигналами шины и входами/выходами блоков. Таким образом, шина выступает в роли канала для любого числа

сигналов. Для получения дополнительной информации о каналах (**Conduits**) и о распределителях сигналов (**mappers**) см. раздел "Распределение сигналов между блоками".

Для рисования линий шины (**Bus**) или канала (**Conduit**) выполните следующие действия:

1. На панели инструментов (**Toolbar**) редактора блок-схем (**Block Editor**) нажмите кнопку **Orthogonal Bus Tool** (рисование ортогональных шин).
2. Для определения начала шины щелкните левой кнопкой мыши на обозначении входа `clk`, затем, переместите указатель, для рисования линии, до границы блока `taps`. При соединении шины с блоком, на краю блока `taps` автоматически появится символ распределителя (**mapper**). Распределитель позволяет назначать входные/выходные порты блока определенным сигналам шины. Для получения дополнительной информации об использовании распределителей сигналов блоков (**block mappers**) см. раздел "Распределение сигналов между блоками".
3. Для создания других необходимых соединений функциональных элементов (**Symbols**) и блоков (**Blocks**), приведенных в таблице 3.2, повторите действия 1, 2. Для облегчения выполнения соединений вы можете посмотреть рис. 3.22 или использовать готовый файл **filtref.bdf**.

Таблица 3.2

Начало связи	Конец связи
Входной (INPUT) вывод <code>clk</code>	Блок <code>taps</code> (уже введен)
Шина, соединяющая вывод <code>clk</code> (INPUT) с блоком <code>taps</code>	Блок <code>state_m</code>
Входы <code>st</code> (INPUT) выходы <code>d[7..0]</code>	Шина, соединяющая блок <code>taps</code> с блоком <code>state_m</code>
Входной (INPUT) вывод <code>reset</code>	Шина, соединяющая блок <code>taps</code> с блоком <code>state_m</code>
Входной (INPUT) вывод <code>newt</code>	Шина, соединяющая блок <code>taps</code> с блоком <code>state_m</code>
Блок <code>state_m</code>	Выходной (OUTPUT) вывод <code>next</code>
Вывод <code>Q</code> примитива триггера <code>DFFE</code>	Выходные (OUTPUT) выходы <code>yn_out[7..0]</code>



Для проведения линии, соединяющей провод или шину с уже существующими проводом или шиной, вы можете использовать инструменты **Orthogonal Node Tool** (рисование ортогональных проводов) или **Orthogonal Bus Tool** (рисование ортогональных

шин). Эти инструменты расположены на панели инструментов (**Toolbar**) редактора **Block Editor** (см. рис. 3.9, поясняющий назначения кнопок панели инструментов). При рисовании соединений проводов или шин с уже существующими проводами или шинами точка их соединения рисуется автоматически.

4. Нажмите кнопку **Selection Tool** (указатель выбора) на панели инструментов редактора **Block Editor** (редактор блок-схем).
5. Для создания других соединений между функциональными элементами и блоками, приведенными в таблице 3.3, повторите действие 2. Для облегчения выполнения соединений вы можете посмотреть рис. 3.22 или использовать готовый файл **filtref.bdf**.

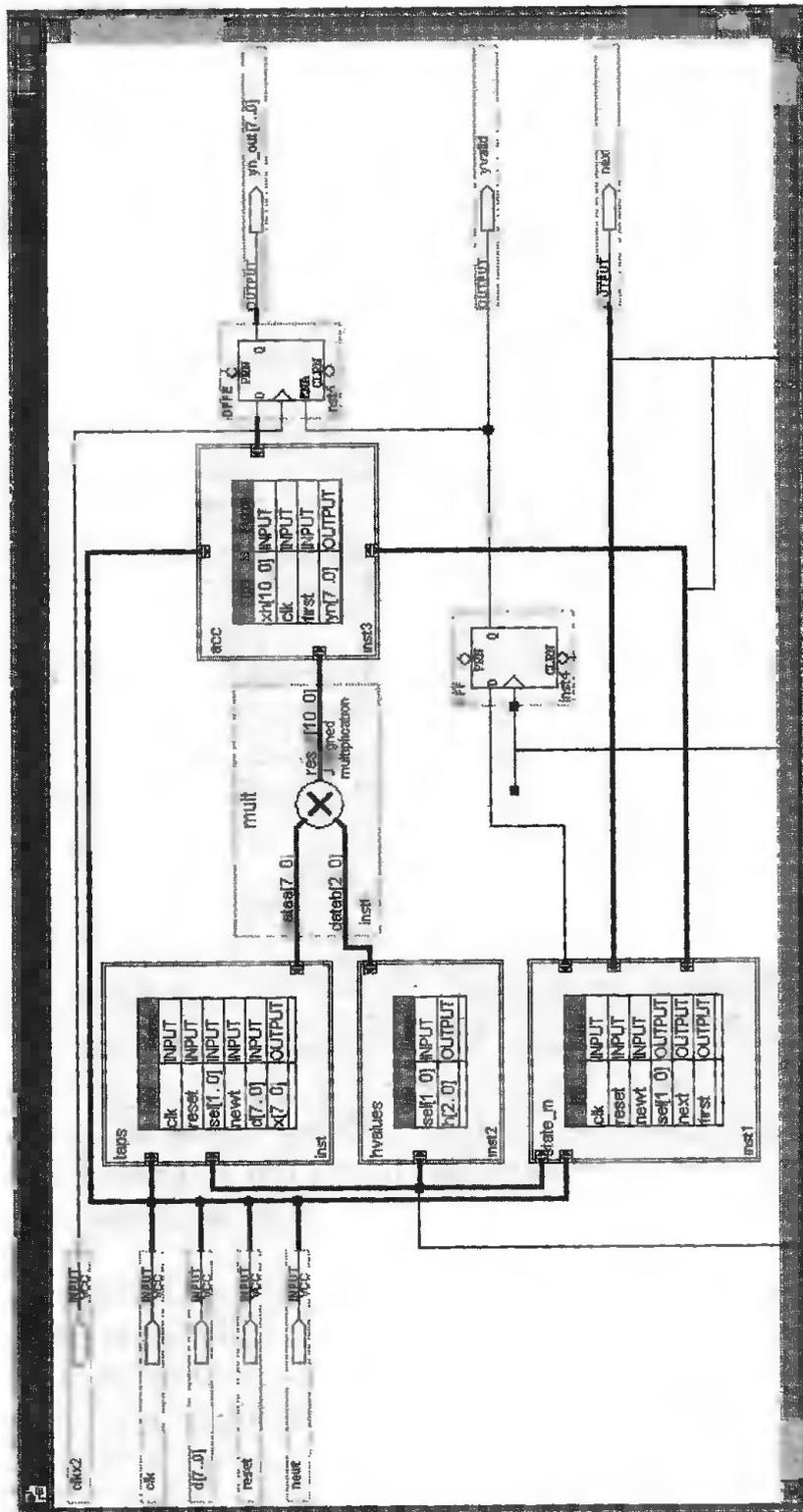
Таблица 3.3

Начало связи	Конец связи
Блок taps	Блок hvalues
Шина, соединяющая блок taps с блоком hvalues	Блок state_m
Шина, соединяющая входной вывод clk (INPUT) с блоком taps	Блок acc
Блок state_m	Блок acc
Блок taps	Вход dataa[7..0] функционального элемента mult
Блок hvalues	Вход datab[7..0] функционального элемента mult
Выход result[10..0] символа mult	Блок acc

6. На панели инструментов редактора **Block Editor** выберите инструмент рисования ортогональных линий (**Orthogonal Node Tool**).
7. Проведите линию от входа D функционального элемента DFF до границы блока state_m.
8. Для создания других соединений блоков и функциональных элементов, показанных в следующей таблице, повторите действие 7. Для облегчения выполнения соединений вы можете посмотреть рис. 3.22 или использовать готовый файл **filtref.bdf**.

Таблица 3.4

Начало связи	Конец связи
Блок state_m	Вход D функционального элемента DFF (уже проведена)
Блок acc	Вход D функционального элемента DFFE



Шина или канал

Проводник, который позже
будет соединен по имени

Точка соединения

Рис. 3.22

 Поскольку вход D символа DFF соединен с блоком state_m эта связь может рассматриваться как канал. Но, поскольку она включает в себя единственный сигнал, правильнее будет обозначить это соединение не толстой линией шины, а тонкой ортогональной линией.

9. Выберите шину (**Bus**) или канал (**Conduit**), соединяющий входной вывод clk с блоком taps, нажмите правую кнопку мыши и во всплывающем меню откройте **Properties** (свойства).
10. В диалоговом окне **Conduit Properties** (свойства канала) перейдите на вкладку **Signals** (сигналы). В списке **Connections** (соединения) показываются все соединения в выделенных каналах (рис. 3.23).

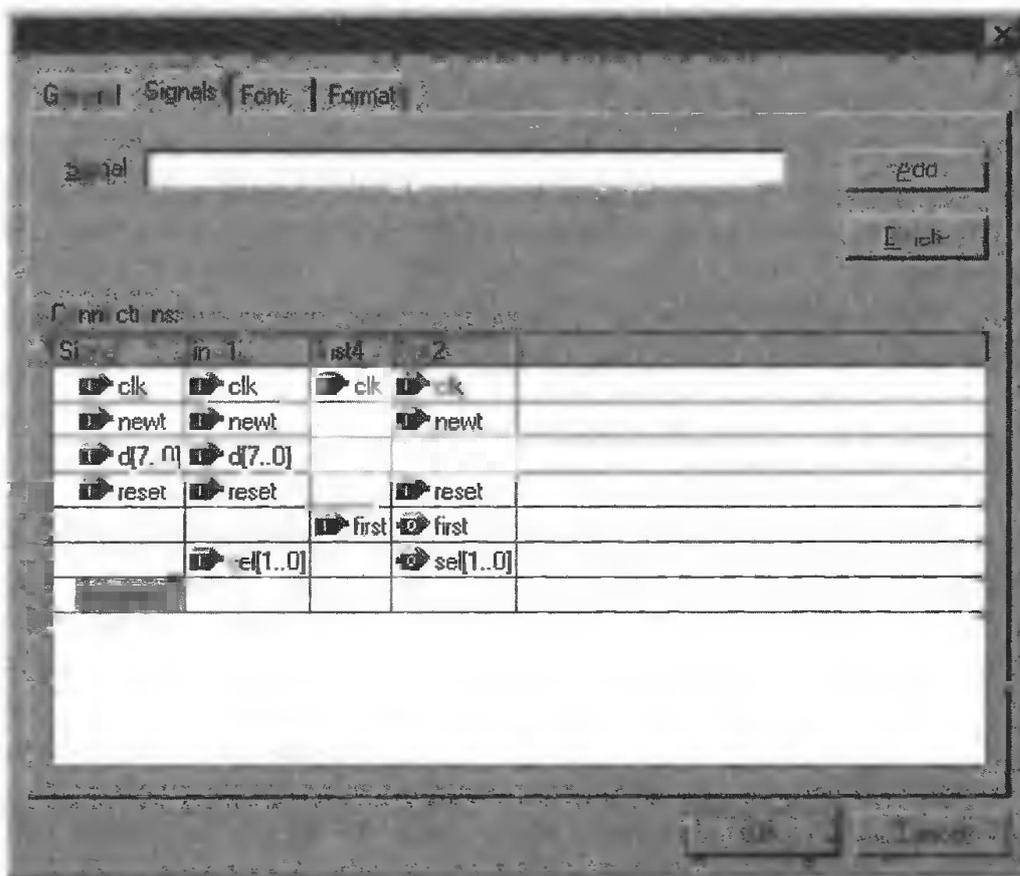


Рис. 3.23

По умолчанию цвета раскраски выходных сигналов отличаются от цветов раскраски входных.

11. Убедитесь, что все сигналы описаны правильно и нажмите кнопку **ОК**.

Для создания проводника (**Node**) выполните следующие действия:

1. На панели инструментов редактора **Block Editor** выберите инструмент **Orthogonal Node Tool** (инструмент рисования ортогональных проводов).

2. Проведите линию от выходного вывода Q функционального элемента DFF к выходному выводу $yvalid$.
3. Для создания других соединений между выводами и примитивами, указанными в таблице 3.5. повторите действия 1, 2.

Таблица 3.5

Начало связи	Конец связи
Выход Q примитива DFF	Выходной контакт $yvalid$ (уже проведена)
Входной контакт $clkx2$	Вход $clock$ примитива DFFE
Соединение выхода Q примитива DFF с выходным выводом $yvalid$	Вход $enable$ (ENA) примитива DFFE

4. Для создания соединения по имени (**Connection by Name**) проведите линию от соединения входа $clock$ с символом DFF на пустое место. Выполнение соединений по имени (**Connections by Name**) подробно описано в разделе "Распределение сигналов между блоками".
5. Сохраните файл проекта, выбрав в меню **File** (операции с файлами) команду **Save** (сохранить файл).

8. Разводка сигналов между блоками

Пакет Quartus предоставляет различные пути разводки сигналов (**map signals**) между блоками (**Blocks**) и функциональными элементами (**Symbols**) (таблица 3.6).

В этом разделе учебного пособия показано, как развести шесть сигналов, которые не были разведены автоматически в режиме **smart**. Также показано, как выполнить соединение двух сигналов по имени и как выполнить соединения в явном виде для четырех других сигналов.

С помощью разводки **smart** выполните следующие соединения в файле **filtref.bdf** (называть эти соединения не надо) (таблица 3.7).

Таблица 3.6

Метод разводки	Описание
Автоматическое соединение Smart	Если имена сигналов входов/выходов (I/Os) одного блока совпадают с именами соответствующих сигналов другого блока, соединения формируются автоматически. Вам не надо рисовать линии этих шин (Buses) или каналов (Conduits). Вам необходимо лишь указать, какие сигналы блоков должны оставаться неприсоединенными. Это необходимо для предотвращения автоматического выполнения их соединений.

Окончание табл. 3.6

Метод разводки	Описание
Connection by name (соединение по именам проводов или шин)	<p>Если имена соединяемых входов/выходов (I/O) блоков различны, то вы можете соединить их с каналом или шиной по имени (by Name). Для этого необходимо назначить каналу имя, соответствующее имени присоединяемого входа/выхода.</p> <p>Этот способ удобен для соединения каналов без отрисовки их линий на схеме. Для создания соединения между такими каналами и блоками сначала необходимо проверить, чтобы каждый такой канал соединялся с блоком одним концом и не имел физического соединения с другим блоком или функциональным элементом.</p> <p>Соединения двух каналов по имени (connection by name) можно выполнить, присвоив им одинаковые имена (полагая, что эти имена соответствуют именам соединяемых входов/выходов блоков). Таким же способом можно соединять по имени и отдельные проводники.</p>
Задание соединений в явном виде с помощью распределителей сигналов (mappers)	<p>Если имена входов/выходов соединяемых блоков различны, но блоки физически соединены — распределение сигналов можно задать в явном виде. Для этого необходимо присвоить имена выводов блоков сигналам шины, а затем распределить сигналы по соединяемым выводам блоков.</p>

Таблица 3.7

Начало связи	Конец связи
Входной (INPUT) вывод clk	Блок входов/выходов (I/Os) с именами clk в блоках, соединенных с выводом clk.
Входные (INPUT) выходы d[7..0]	Блок входов/выходов (I/Os) с именем d[7..0] в блоке taps
Входной (INPUT) вывод reset	Блок входов/выходов (I/Os) с именами reset в блоках taps и state_m
Входной (INPUT) newt	Блок входов/выходов (I/Os) с именами newt в блоках taps и state_m
Блок входов/выходов I/O с именем sel[7..0] блока taps	Блок входов/выходов (I/Os) с именами sel[7..0] в блоках hvalues и state_m
Блок входов/выходов I/O с именем first блока state_m	Блок входов/выходов (I/Os) с именем first в блоке acc
Блок входов/выходов I/O с именем next блока state_m	Выходной (OUTPUT) вывод с именем nex

Чтобы назначить имена ряду сигналов для возможности соединения их по имени (**Connection by Name**) или логического соединения (**Logical connection**), выполните следующие действия:

1. С помощью указателя выбора (**Selection Tool**) выделите проводник, соединяющий блок `state_m` со входом `D` примитива `DFF`.
2. Нажмите правую кнопку мыши и во всплывающем меню выберите **Properties** (свойства).
3. Если необходимо — перейдите на вкладку **General** (основные) в диалоговом окне **Conduit Properties** (свойства каналов).
4. В поле **Conduit name** (имя канала) введите имя `next`. (рис. 3.24).

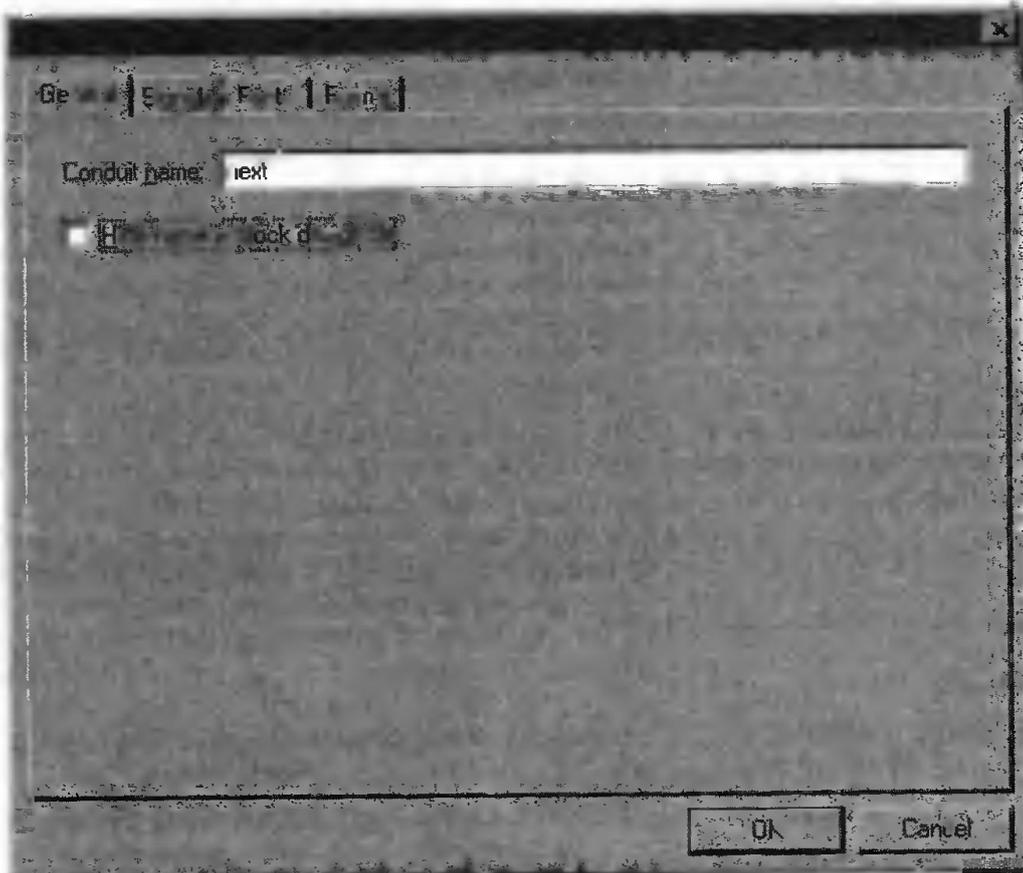


Рис. 3.24

Комментарий:

Hide name in block design file

Скрыть имя в файле блок-схемы проекта

5. Нажмите кнопку **OK** — сигнал `next` автоматически добавится к каналу и его имя появится под линией канала. Добавление этого имени создает логическое соединение между блоком `state_m` и входом `D` примитива `DFF`.
6. Для именованного проводов, соединенных с входом `Clock` примитива `DFF`, выполните действия 1–5. Назовите вывод `clk` так, чтобы вы могли создать логическое соединение или "соединение по имени" (**Connection by Name**) между входным выводом `clk` и входом `Clock` примитива `DFF`.
7. Нажмите **Save** (меню **File**).

Чтобы система Quartus могла правильно соединить функциональный элемент mult с другими блоками проекта необходимо точно развести четыре оставшихся соединения. Для выполнения соединений между блоками, имеющими различные имена, выполните следующие действия:

1. На шине, соединяющей блок taps со входом dataa[7..0] функционального элемента mult, дважды щелкните по обозначению mapper (распределитель сигналов) на конце шины блока taps (рис. 3.25) — откроется диалоговое окно **Mapper Properties** (свойства распределителя), показанное на рис. 3.26.

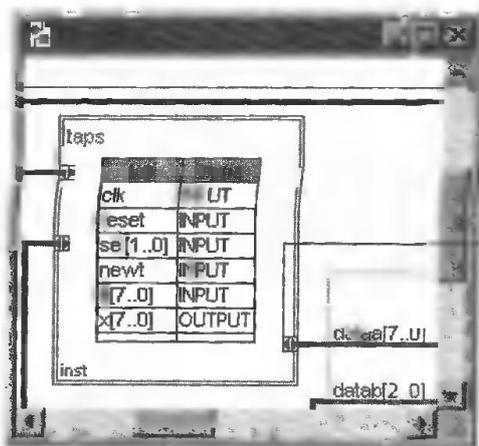


Рис. 3.25

2. Если необходимо — откройте в этом диалоговом окне вкладку **General** (основные) (рис. 3.26).

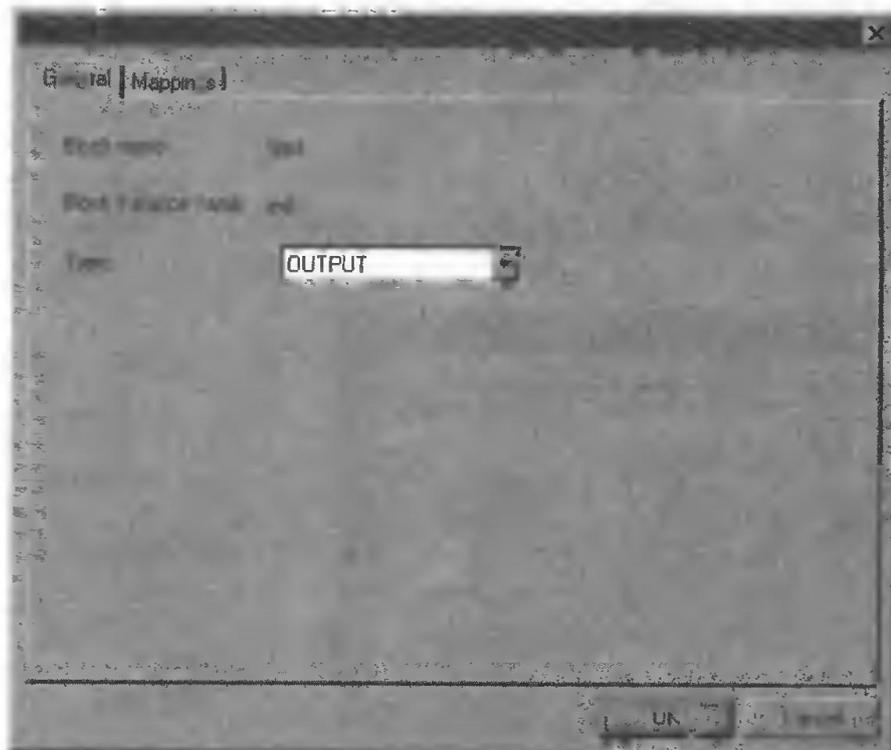


Рис. 3.26

3. В списке **Type** (тип вывода) выберите **OUTPUT** (выход).
4. Откройте вкладку **Mappings**.
5. В списке **I/O on block** (входы/выходы блока) выберите $x[7..0]$.
6. В поле **Signals in conduit** (сигналы в канале) введите $dataa[7..0]$.
7. Для выполнения соединения нажмите кнопку **Add** — соединение появится в списке **Existing mappings** (рис. 3.27).

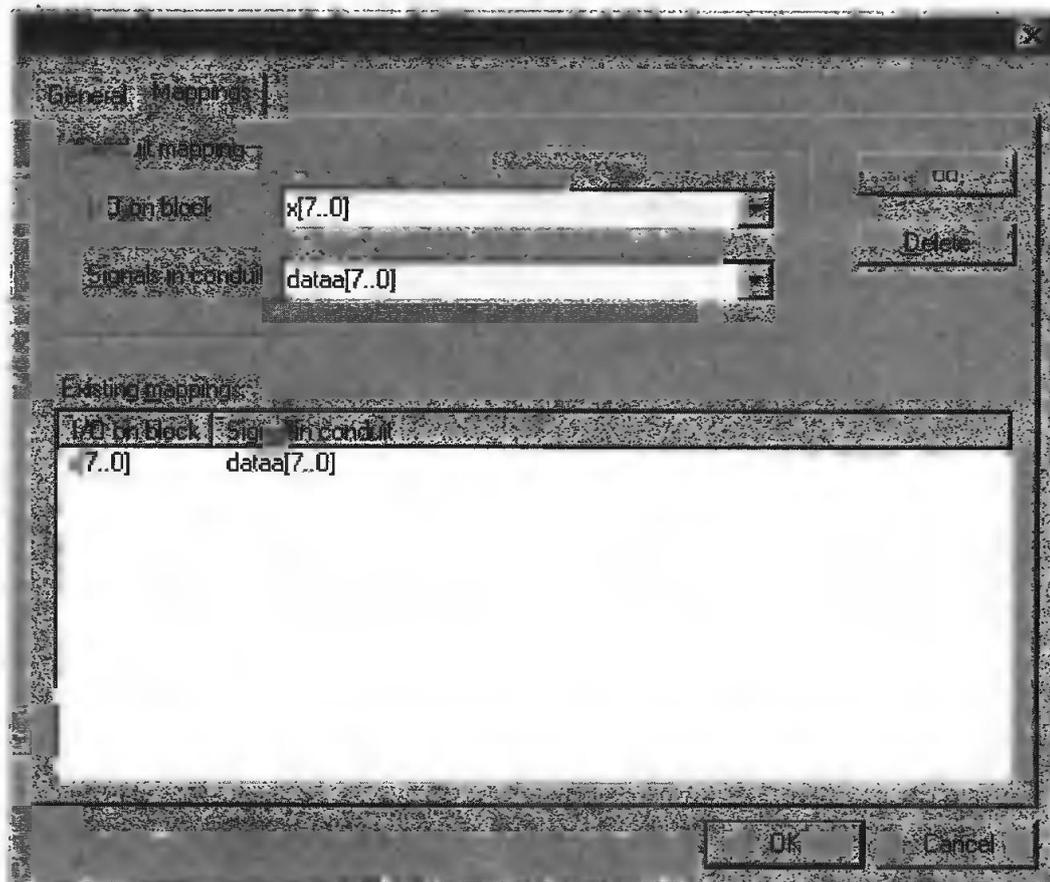


Рис. 3.27

8. Нажмите кнопку **OK** — сигнал $dataa[7..0]$ добавится к шине и появится таблица, в которой будут показаны сигналы, соединенные со входом $dataa[7..0]$ функционального элемента **mult** (рис. 3.28).

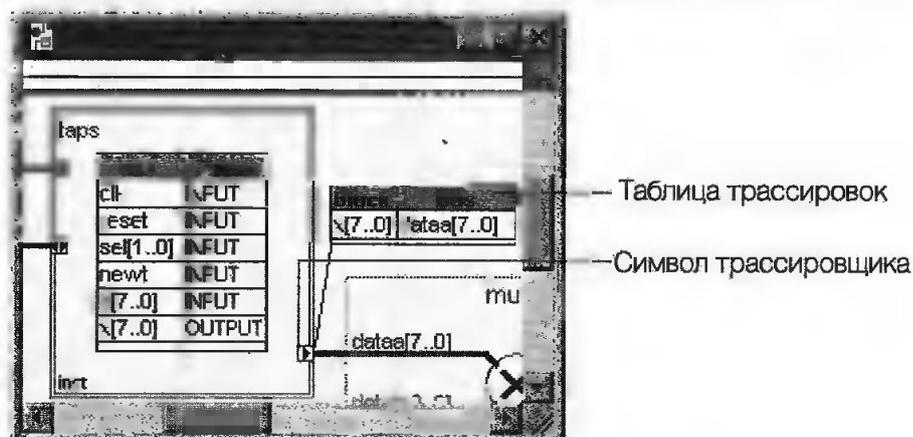


Рис. 3.28

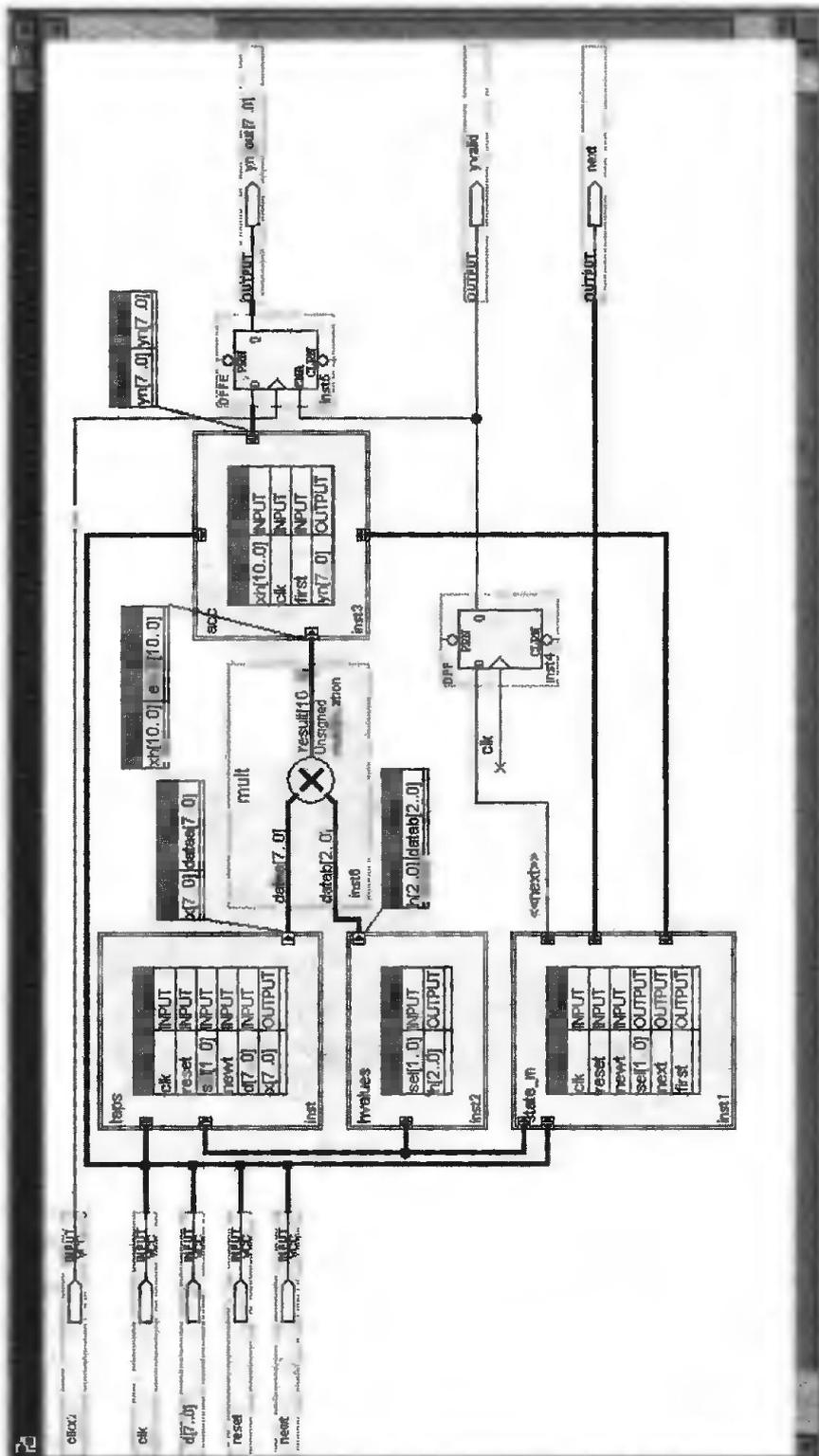


Рис. 3.29

9. Для создания других необходимых соединений повторите действия 1–8 (таблица 3.8). Для облегчения выполнения соединений вы можете посмотреть рис. 3.2 или использовать готовый файл **filtref.bdf**.

Таблица 3.8

Связь	Тип вывода	Входы/выходы (I/O) блока	Сигналы в канале
Шина от блока taps к входу dataa[7..0] функционального элемента mult (уже проведена)	OUTPUT (выход)	x[7..0]	dataa[7..0]
Шина от блока hvalues к входу datab[7..0] mult	OUTPUT (выход)	h[2..0]	datab[2..0]
Шина из выхода result[10..0] функционального элемента mult к блоку асс.	INPUT (вход)	xh[10..0]	result[10..0]
Шина от блока асс к входу D примитива DFFE.	BIDIR (двунаправленный)	yn[7..0]	yn[7..0]

10. Для сохранения файла блок-схемы проекта (**Block Design File**) выполните команду **Save** (сохранить) меню **File**. Текущая блок-схема создаваемого проекта показана на рис. 3.29.

Урок 4. Создание описания на языке Verilog HDL

После создания блока необходимо создать файл описания этого блока (если описание еще не создано). Система Quartus позволяет сначала создать файл с основной (типовой) структурой описываемого блока, а затем наполнить эту структуру конкретным содержанием.

 Если вами уже хорошо изучено создание файлов на языке Verilog HDL в текстовом редакторе (**Text Editor**) пакета Quartus, вы можете скопировать готовые файлы *hvalues.v*, *taps.v*, *state_m.v* и *acc.v* из подкаталога `\qdesigns\tutorial` в подкаталог `\qdesigns\fir_filter`. Рекомендуется копировать файлы учебного пособия следующим образом: открыть файл в пакете Quartus с помощью команды **Open** (открыть) меню **File**, затем выбрать команду **Save As** (сохранить файл как ...) меню **File**, в появившемся окне установить ключ **Add file to current project** (добавить файл к текущему проекту), а затем сохранить файл в подкаталоге `\qdesigns\fir_filter`.

Этот урок учебного пособия описывает следующие действия:

1. Создание текстового файла Verilog Design File для блока hvalues.
2. Создание текстовых файлов Verilog Design File для других блоков.

1. Создание описания блока hvalues на языке Verilog HDL

Для создания структуры текстового файла Verilog Design File (.v) блока hvalues выполните следующие действия:

1. Выделите блок hvalues.
2. Нажмите правую кнопку мыши и во всплывающем меню выберите команду **Create Design File** (создать файл проекта).
3. Под надписью **File type** (тип файла) выберите **Verilog HDL** (файл на языке Verilog HDL).
4. Включите функцию **Add the new design file to the current project** (добавить новый файл к текущему проекту).
5. Проверьте, чтобы в строке **File name** (имя файла) отображался файл **hvalues.v**, расположенный в рабочем каталоге проекта **fir_filter** (рис. 3.30).

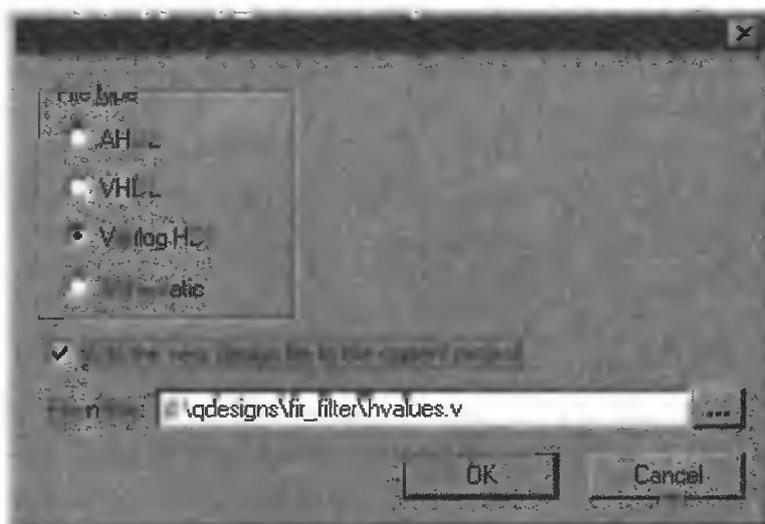


Рис. 3.30

6. Нажмите кнопку **ОК**. Система Quartus сообщит об успешном создании файла и автоматически откроет окно текстового редактора (**Text Editor**) с загруженным в него новым файлом. Фрагмент созданного файла показан ниже. Файл включает шаблон объявления модуля с именами портов соответствующим определенным в блоке данным.



В тексте описания содержится несколько пар созданных системой комментариев, которые начинаются со слов "ALTERA" и заканчиваются словами "DO NOT REMOVE THIS LINE!" Не вводите текст между этими комментариями, поскольку он впоследствии может быть изменен системой Quartus.

Пример файла *hvalues.v*:

```
// Module Declaration
module hvalues
(
    // {{ALTERA_ARGS_BEGIN}} DO NOT REMOVE THIS LINE!
    sel, h
    // {{ALTERA_ARGS_END}} DO NOT REMOVE THIS LINE!
);
// Port Declaration
// {{ALTERA_IO_BEGIN}} DO NOT REMOVE THIS LINE!
input [1:0] sel;
output [2:0] h;
// {{ALTERA_IO_END}} DO NOT REMOVE THIS LINE!

endmodule
```

7. Для выполнения описания блока **hvalues.v** перед выражением `end module` введите следующие строки:

```
reg [2:0]h;
always @(sel)
case (sel)
    2'b 00 : h = 3'b 111;
    2'b 01 : h = 3'b 101;
    2'b 10 : h = 3'b 011;
    2'b 11 : h = 3'b 001;
endcase
```

Теперь описание **hvalues.v** примет следующий вид:

```
// Module Declaration
module hvalues
(
    // {{ALTERA_ARGS_BEGIN}} DO NOT REMOVE THIS LINE!
    sel, h
    // {{ALTERA_ARGS_END}} DO NOT REMOVE THIS LINE!
);
// Port Declaration
// {{ALTERA_IO_BEGIN}} DO NOT REMOVE THIS LINE!
input [1:0] sel;
output [2:0] h;
// {{ALTERA_IO_END}} DO NOT REMOVE THIS LINE!

    reg [2:0]h;

always @(sel)
```

```
case (sel)
    2'b 00 : h = 3'b 111;
    2'b 01 : h = 3'b 101;
    2'b 10 : h = 3'b 011;
    2'b 11 : h = 3'b 001;
endcase

endmodule
```

8. Сохраните файл проекта, выбрав в меню **File** (операции с файлами) команду **Save** (сохранить).

2. Копирование файлов текстового описания (Verilog Design File) других блоков

Для сокращения времени можно воспользоваться готовыми файлами (Verilog Design Files) описания блоков `state_m` и `acc` на языке Verilog HDL.

Скопируйте файлы `taps.v`, `state_m.v` и `acc.v` из каталога `\qdesigns\tutorial` в каталог `\qdesigns\fir_filter`. Файлы учебного пособия рекомендуется копировать следующим образом: открыть файл в пакете Quartus с помощью команды **Open** (открыть) меню **File**, затем выбрать команду **Save As** (сохранить как ...) меню **File**, в появившемся окне установить ключ **Add file to current project** (добавить файл к текущему проекту), а затем сохранить файл в подкаталоге `\qdesigns\fir_filter`.

Урок 5. Создание функционального блока при помощи мастера мегафункций MegaWizard

Для рассматриваемого проекта необходимо создать функцию 12-разрядного арифметического сумматора. Вы можете создать ее на основе стандартной мегафункции `lpm_add_sub`. Например, так создана функция `ассум`, представляемая файлом `acc.v`. Для создания таких функций удобно пользоваться встроенным в пакет Quartus мастером мегафункций **MegaWizard**. Для дополнительной информации о мастере мегафункций **MegaWizard** см. раздел "Создание графических обозначений примитива и мегафункции".

 Если вы уже знакомы с использованием мастера **MegaWizard** — можете скопировать готовые файлы из каталога `\qdesigns\tutorial` в каталог `\qdesigns\fir_filter`. Файлы рекомендуется копировать следующим образом: открыть файл в пакете Quartus с помощью команды **Open** (открыть) меню **File**, затем выбрать команду **Save As** (сохранить как ...) меню **File**, в появившемся окне установить ключ **Add file to current project**

(добавить файл к текущему проекту), а затем сохранить файл в подкаталоге `\qdesigns\fir_filter`. После копирования файлов можно сразу переходить к разделу "Компиляция проекта".

Имя файла:	Описание:
accum.v	Функциональный блок на языке Verilog HDL, созданный на основе мегафункции <code>1pm_add_sub</code> с помощью мастера мегафункций MegaWizard .
accum.inc	Включаемый текстовый файл ASCII, автоматически созданный мастером MegaWizard и используемый при компиляции. Файл содержит объявление имени созданного блока и его входов и выходов на языке Verilog HDL.

В этом уроке рассматриваются следующие действия:

1. Создание текстового Verilog HDL-описания блока на основе мегафункции.
2. Подключение к проекту файлов, созданных мастером **MegaWizard**.

1. Создание текстового Verilog HDL-описания блока на основе мегафункции

Чтобы создать описание Verilog HDL функционального блока на основе готовой мегафункции `1pm_add_sub` с помощью мастера **MegaWizard** выполните следующие действия:

1. В меню **Tools** (инструментальные средства) выберите команду **MegaWizard**. Появится окно мастера **MegaWizard** (рис. 3.31).

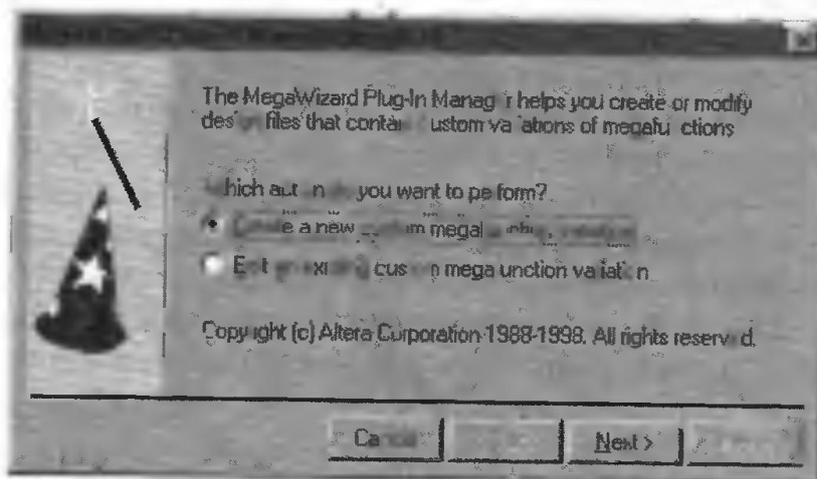


Рис. 3.31

2. На вопрос: "Which action do you want to perform?" (какое действие вы хотите выполнить?) выберите **Create a new custom megafunction variation** (создать новый блок на основе мегафункции), нажмите кнопку **Next** (следующий шаг).

3. В списке **Available Megafunctions** (доступные мегафункции) раскройте папку арифметических мегафункций, щелкнув по значку "+", и затем выберите мегафункцию **LPM_ADD_SUB** (мегафункция сумматора/вычитателя).
4. На вопрос: "**Which type of output file do you want to create?**" (какой тип выходного файла вы хотите создать?) выберите **Verilog HDL**.
5. На вопрос: "**What name do you want for the output file?**" (как вы хотите назвать выходной файл?) введите `d:\qdesigns\fir_filter\accum.v` и нажмите кнопку **Next** (следующий шаг).
6. На вопрос: "**How wide should the input buses be?**" (какова требуемая разрядность входных шин 'dataa' и 'datab'?) введите 8.
7. На вопрос: "**Which operating mode do you want for the adder/subtractor?**" (каков требуемый режим работы сумматора/вычитателя) выберите **Addition only** (только суммирование) и нажмите кнопку **Next** (следующий шаг).
8. На вопрос: "**Is the 'dataa' or 'datab' bus value a constant?**" (является ли шина 'dataa' или 'datab' константой?) выберите **No** (нет) – оба значения переменные, и нажмите кнопку **Next** (следующий шаг).
9. На вопрос: "**Do you want any optional inputs or outputs?**" (хотите использовать какие-либо дополнительные входы или выходы?) удостоверьтесь, что все дополнительные входы/выходы выключены и нажмите кнопку **Next** (следующий шаг).
10. На вопрос: "**Do you want to pipeline the function?**" (хотите реализовать эту функцию с использованием конвейера?) выберите **No** (нет) и нажмите **Next** (следующий шаг). Появится итоговая страница со списком файлов, созданных встроенным мастером *MegaWizard*.
11. Нажмите кнопку **Finish** (готово) – мастер **MegaWizard** создаст файл `accum.v`, готовый к использованию в вашем проекте.

2. Подключение к проекту файлов, созданных мастером MegaWizard

Для того чтобы добавить созданные файлы `accum.v` и `mult.v` к проекту `fir_filter`, выполните следующие действия:

1. В меню **Project** (проект) выберите команду **Add Files to Project** (добавить файлы к проекту). Автоматически появится вкладка **Add Files**.
2. В поле **File Name** (имя добавляемого файла) введите файл `accum.v` и, при необходимости, файл `mult.v` (рис. 3.32).
3. Нажмите кнопку **ОК**. Теперь вы создали все необходимые файлы и добавили их к проекту `fir_filter`.

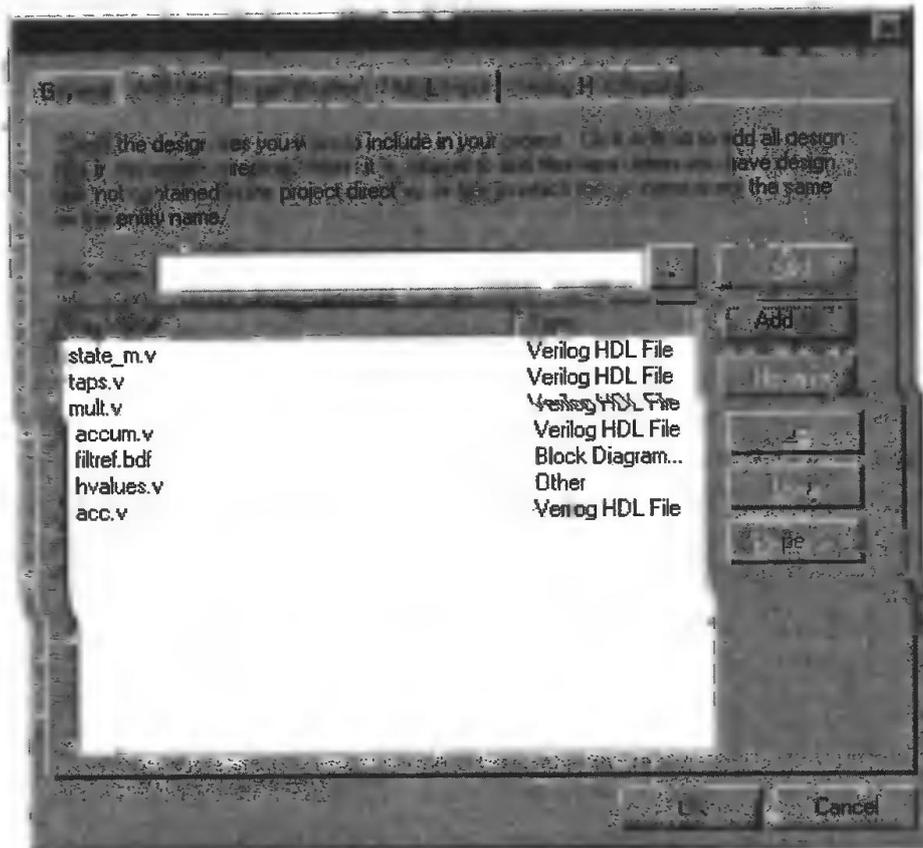


Рис. 3.32

3.3. Компиляция проекта

Компилятор пакета Quartus состоит из ряда модулей, выполняющих следующие функции: проверка проекта на наличие ошибок, логический синтез, размещение и разводка проекта в ПЛИС, генерация выходных файлов для моделирования проекта, анализа временных характеристик и программирования ПЛИС.

В начале компиляции из проекта извлекается информация об иерархических связях между составляющими его файлами и описание проекта проверяется на наличие основных ошибок. Затем создается организационная карта проекта и все файлы преобразуются в единую базу данных, с которой впоследствии будет работать система.

Компиляция может выполняться с учетом заданных требований: обеспечение требуемых временных характеристик проекта, увеличение быстродействия, оптимизация используемых ресурсов ПЛИС. Компилятор создает файлы для программирования или конфигурирования ПЛИС Altera.

Промежуточные и окончательные результаты компиляции в системе Quartus можно просмотреть в окне **Compilation Report** (отчет о компиляции).

Программирование или конфигурирование ПЛИС Altera может быть выполнено как с помощью встроенных средств пакета Quartus (**Programmer**), так и с использованием стандартных промышленных средств программирования.

Следующие уроки учебного пособия посвящены: настройке компилятора; компиляции файла верхнего уровня иерархии проекта; просмотру с помощью редактора физического размещения (**Floorplan**) созданной во время последней компиляции (**Last Compilation Floorplan**) компоновки проекта в ПЛИС; назначению реализации логических функций в блоке встроенной памяти (**ESB**) и повторной компиляции (**recompile**) проекта после внесенных изменений.

Урок 6. Настройка компилятора

Система Quartus позволяет выполнять компиляцию как всего проекта, так и любой его составляющей. Для компиляции может быть выбрана любая часть иерархии проекта.

При настройке компилятора определяются компилируемая часть проекта (**Compilation focus**), тип компиляции, семейство и тип ПЛИС, а также другие управляющие компиляцией параметры. При создании нового проекта система Quartus по умолчанию устанавливает значения всех необходимых параметров. Параметры, заданные по умолчанию, вы можете переопределить в соответствии с вашими требованиями. Кроме того, вы можете выбирать различные параметры настройки при выполнении компиляции.

В данном уроке описываются следующие действия:

1. Просмотр основных параметров компилятора.
2. Определение семейства и типа ПЛИС.
3. Определение режима компиляции.
4. Определение и настройка параметров логического синтеза и разводки.
5. Определение параметров верификации проекта на этапе компиляции.

 Здесь описываются действия по просмотру и редактированию параметров настройки компилятора с помощью меню и диалоговых окон. Однако параметры настройки компилятора можно также определить с использованием мастера пошаговой настройки компилятора **Compiler Settings Wizard**. Запуск мастера **Compiler Settings Wizard** осуществляется в меню **Processing** (обработка).

1. Просмотр и настройка основных свойств компилятора

Вкладка **General** (основные параметры) диалогового окна **Compiler Settings** (параметры компилятора) позволяет выбрать нужную группу

параметров, определить и сохранить новую группу параметров, выбрать объекты компиляции или удалить существующие параметры настройки.

Для просмотра заданных по умолчанию значений основных параметров компиляции текущего проекта выполните следующие действия:

1. Убедитесь, что находитесь в режиме компиляции **Compile**, выбрав команду **Compile Mode** (режим компиляции) в меню **Processing** (обработка).
2. В меню **Processing** (обработка) выберите **Compiler Settings** (параметры компилятора). В появившемся окне автоматически откроется вкладка **General** (основные параметры).

В этом случае на вкладке **General** показаны только основные параметры компилятора, заданные пакетом Quartus по умолчанию при создании проекта **fir_filter** с помощью мастера **New Project**. Эти параметры по умолчанию назначены объекту **filtref** верхнего уровня иерархии проекта (рис. 3.33).

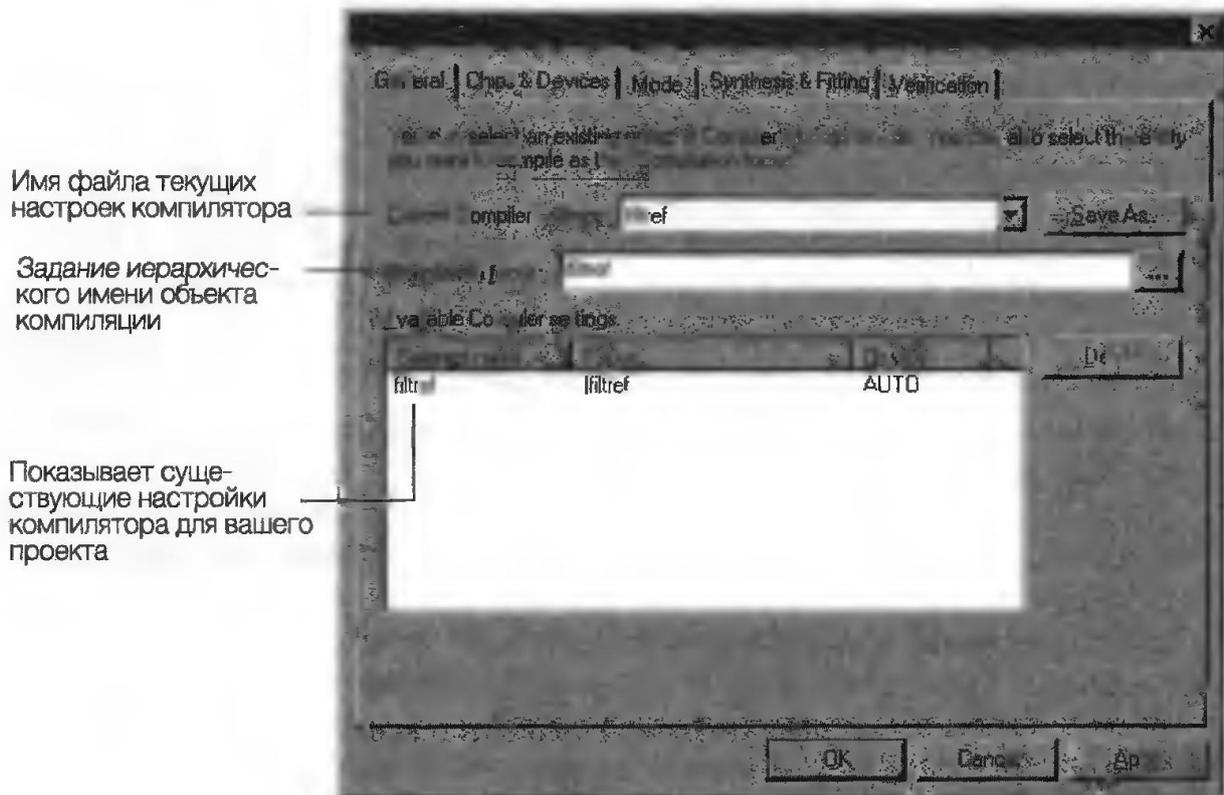


Рис. 3.33

2. Выбор семейства и типа микросхемы

На вкладке **Chips & Devices** диалогового окна **Compiler Settings** (параметры компилятора) можно выбрать семейство ПЛИС (**Device Family**) и тип ПЛИС (**Device**), на которой вы хотите реализовать ваш проект.

Чтобы выбрать семейство и тип микросхемы, выполните следующие действия:

1. Перейдите на вкладку **Chips & Devices** в диалоговом окне **Compiler Settings** (параметры компилятора).
2. В списке **Family** (семейство) выберите **APEX20K**.
3. В поле **Target Device** (ПЛИС для реализации проекта) выберите **Specific device selected in "Available devices" list** (ПЛИС из списка доступных микросхем).
5. Из списка **Available devices** (доступные типы микросхем) выберите тип **EP20K100QC208-1**.
6. В правой части поля **Available devices** выберите следующие параметры:
 - a) из списка **Package** (корпус) выберите корпус **PQFP**;
 - b) из списка **Pin count** (количество выводов) выберите **208**;
 - c) из списка **Speed Grade** (градация быстродействия) выберите **-1** (рис. 3.34).

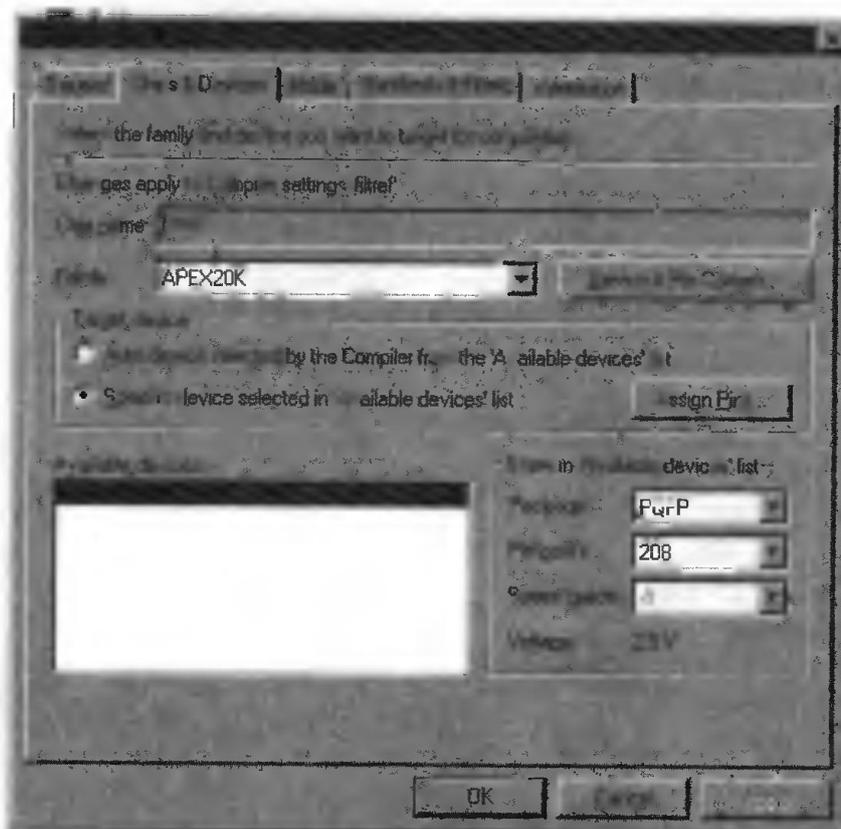


Рис. 3.34

7. Нажмите кнопку **Apply** (применить выбранные параметры).

3. Выбор режима компиляции

Вкладка **Mode** (режим) диалогового окна **Compiler Settings** (параметры компилятора) позволяет вам задать параметры, определяющие тип компиляции, скорость ее выполнения и требуемый для компиляции объем дискового пространства.

Для того чтобы выбрать режим компиляции, выполните следующие действия:

1. В диалоговом окне **Compiler Settings** (параметры компилятора) перейдите на вкладку **Mode** (режим).
2. В поле **Compilation level** (вид компиляции) выберите **Full compilation** (полная компиляция).
3. Для сокращения времени компиляции в поле **Compilation speed/disk usage tradeoff** (выбор между скоростью выполнения компиляции и используемым дисковым пространством) выберите **Smart compilation/more disk space** (ускоренная, частичная компиляция). В режиме **Smart compilation** компилируются только измененные части проекта.
4. Проверьте, включена ли функция **Preserve fewer node names to save disk space** (для экономии дискового пространства запретить использование для узлов нескольких имен) (рис. 3.35).

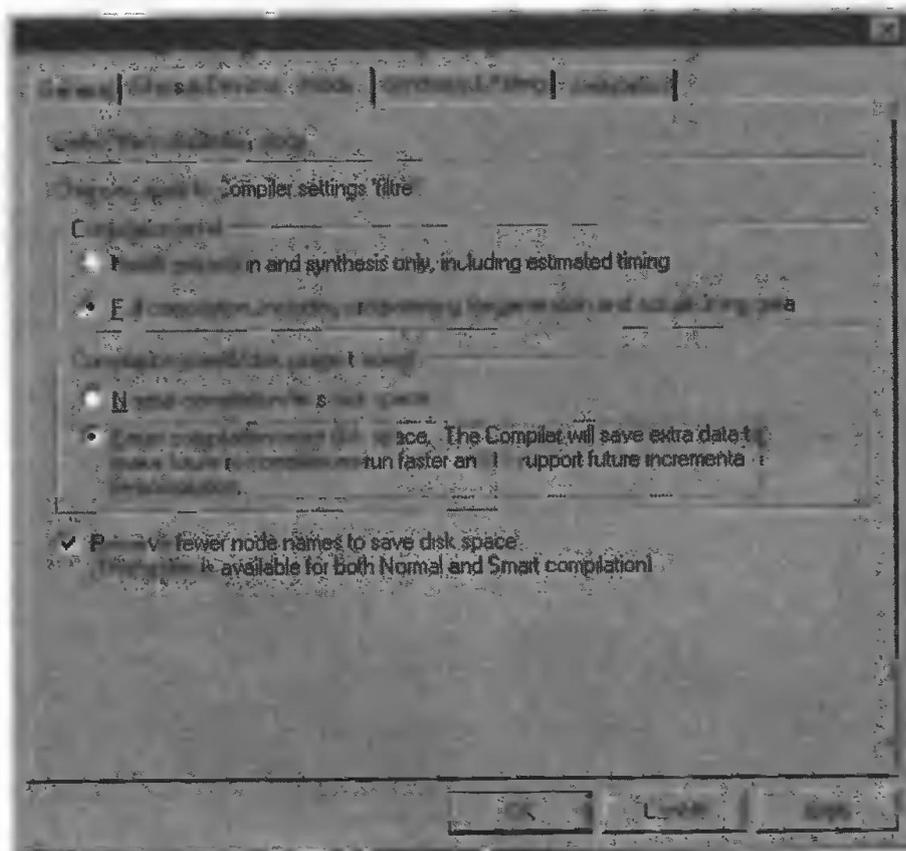


Рис. 3.35

4. Определение параметров логического синтеза и компоновки проекта

Вкладка **Synthesis & Fitting** (синтез и разводка) диалогового окна компилятора позволяет вам определять параметры, определяющие реализацию проекта в ПЛИС. В этом диалоговом окне можно также

определить выполнение компоновки проекта с учетом заданных требований к быстродействию.

Чтобы определить параметры синтеза и разводки, выполните следующие действия:

1. В диалоговом окне **Compiler Settings** (параметры компилятора) перейдите на вкладку **Synthesis & Fitting** (синтез и разводка)
2. Убедитесь, что включена опция **Use timing-driven compilation to achieve performance goals** (компиляция с использованием временных характеристик для достижения заданного быстродействия проекта) (рис. 3.36).

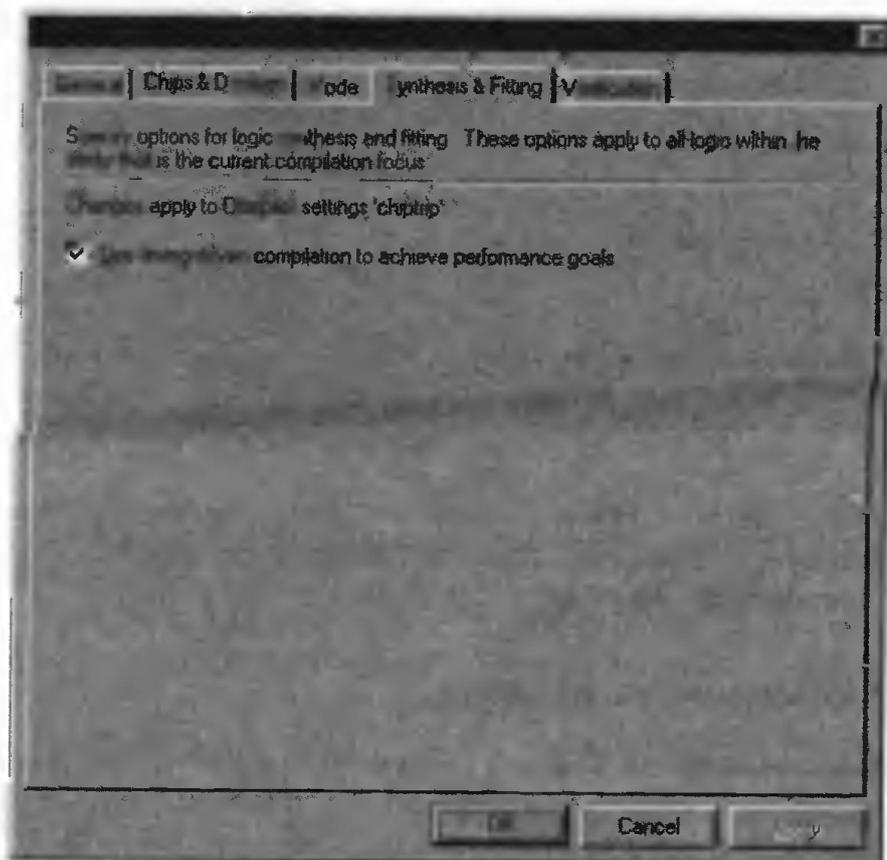


Рис. 3.36

5. Определение параметров верификации

На вкладке **Verification** (верификация) диалогового окна **Compiler Settings** (параметры компилятора) можно задать опции автоматического запуска анализа временных характеристик и/или запуска моделирования (**Simulation**) после окончания компиляции.

Для того чтобы определить параметры верификации проекта, выполните следующие действия:

1. В диалоговом окне **Compiler Settings** (параметры компилятора) откройте вкладку **Verification** (верификация).

- Убедитесь, что опция **Run timing analyses** (запускать анализ временных характеристик) включена (рис. 3.37).

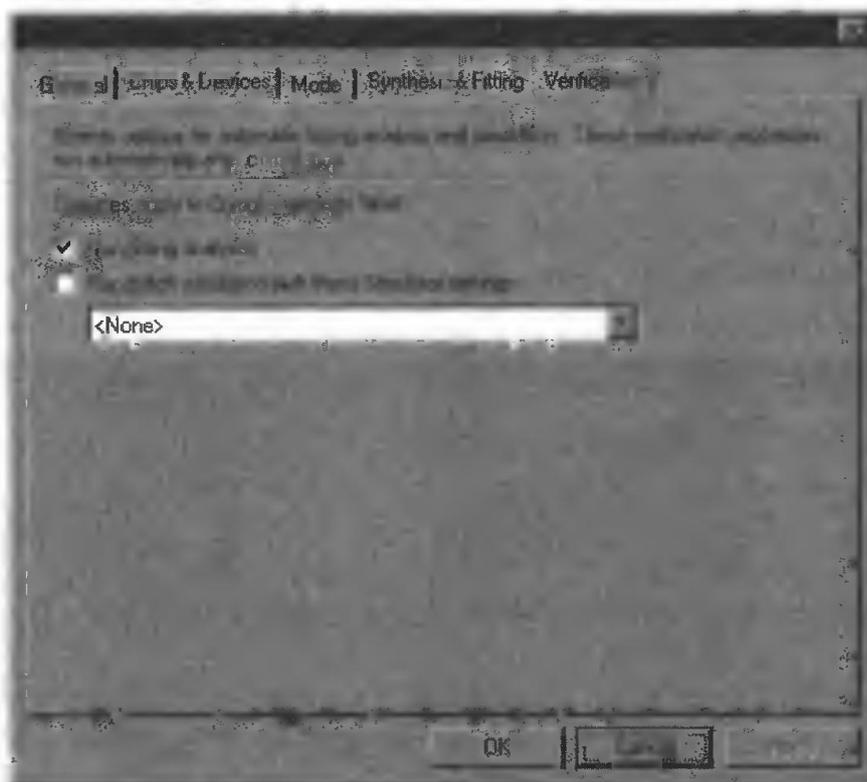


Рис. 3.37

- Нажмите кнопку **OK** — заданные вами параметры компиляции сохранятся в установках проекта **filtref**. После запуска компилятора эти параметры будут использованы по умолчанию.

Урок 7. Компиляция проекта

Во время компиляции проекта используются текущие установленные для него параметры компиляции. Компилятор автоматически распознает и обрабатывает все файлы, относящиеся к объекту компиляции: файлы ***.inc** — включаемые файлы, содержащие описания функций на языке AHDL; файлы ***.mif** — файлы инициализации памяти; файлы ***.hex** — файлы инициализации памяти в шестнадцатиричном формате Intel; файлы ***.psf**, ***.esf**, ***.and**, ***.csf** — содержащие параметры проекта и компилятора.

Генерируемые в процессе компиляции предупреждения (**Warnings**) и сообщения об ошибках (**Error messages**) автоматически появляются в окне **Messages** (сообщения).

В этом уроке описываются следующие действия:

- Запуск компилятора.
- Локализация (определение положения) источников сообщений
- Просмотр сообщений компилятора

1. Запуск компилятора

Для компиляции объекта проекта **filtref** выполните следующие действия:

1. В меню **Processing** (обработка) выберите **Start Compilation** (запуск компиляции). Начнется компиляция объекта проекта **filtref** и всех входящих в него объектов с использованием определенных для проекта **filtref** текущих параметров компилятора. Степень завершенности компиляции и продолжительность каждого ее этапа автоматически отображается в окне **Status** (рис. 3.38).

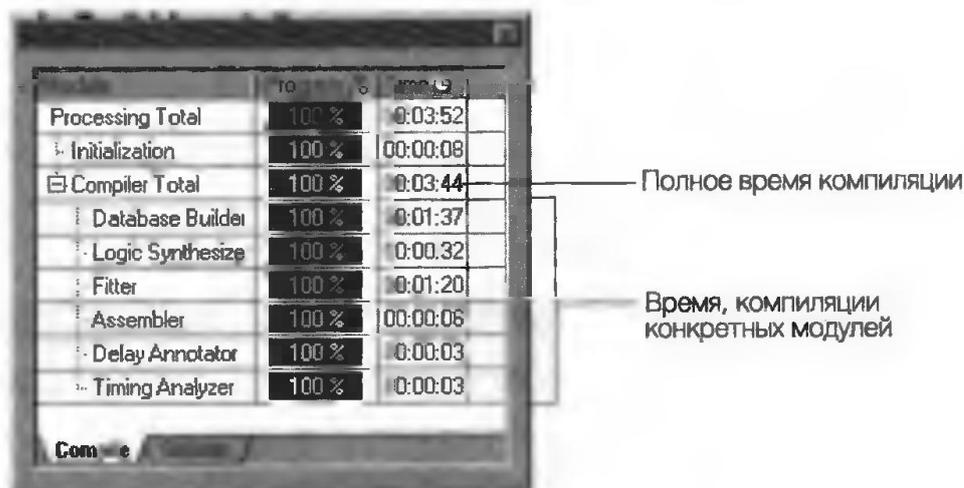


Рис. 3.38

Кроме того, результаты компиляции отображаются в окне **Compilation Report** (отчет о компиляции).

Компиляция выполняется в фоновом режиме. Это позволяет при большом времени компиляции работать с другими окнами пакета Quartus, а также с другими запущенными в операционной системе программными пакетами. Однако компиляция текущего объекта **filtref** не занимает много времени.

2. После успешного завершения компиляции на экран будет выведено соответствующее сообщение. Нажмите кнопку **ОК**.

2. Локализация источников сообщений

В течение компиляции все сообщения появляются во вкладке **Processing** (обработка) окна **Messages** (сообщения). Выводимые в окне **Messages** сообщения могут относиться к определенному месту в файле проекта или в другом исходном файле.

Чтобы найти (локализовать) источник сообщения компилятора, выполните следующие действия:

1. В окне **Messages** (сообщения) разверните подкаталог **d:\qdesigns\fir_filter\acc.v**, нажав значок "+", и просмотрите информационное сообщение (рис. 3.39).

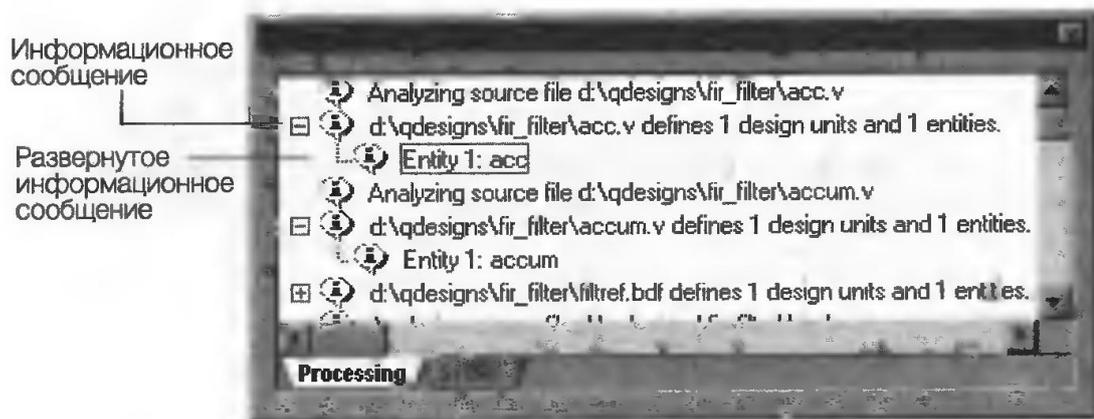


Рис. 3.39

2. Дважды щелкните по развернутому информационному сообщению **Entity 1:acc**. В окне **Text Editor** (текстовый редактор) откроется файл **acc.v**, и в нем цветом будет выделен источник сообщения – раздел **Module Declaration** (объявление модуля) (рис. 3.40).

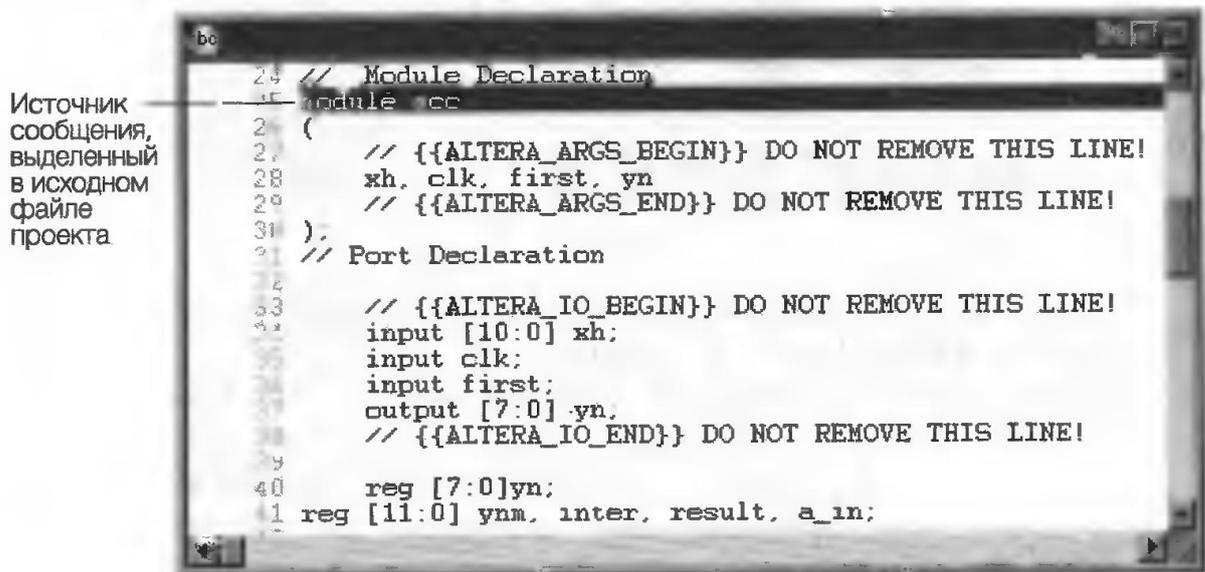


Рис. 3.40

3. После просмотра источника сообщения закройте окно текстового редактора (**Text Editor**).

3. Просмотр отчета о компиляции

Информация о текущей компиляции автоматически выводится в окне **Compilation Report** (отчет о компиляции). При первом отображении этого окна, в правой его области, появляется раздел **Summary** (сводная информация). В этом разделе отображается сводная информация о компиляции: заданные временные характеристики, имена всех откомпилированных объектов проекта, общее количество используемых логических ячеек и выводов ПЛИС, а также общий объем использованной памяти.

Разделы отчета о компиляции (**Compilation Report**) организованы в иерархическую древовидную структуру и предоставляют следующую информацию: в разделе **floorplan** (топология) показывается выполненная компилятором компоновка проекта в ПЛИС: размещение элементов проекта и связи между ними, использование выводов и логических элементов ПЛИС, реализуемые ими логические выражения, использование глобальных синхросигналов и сигналов управления; максимальную рабочую частоту переключений f_{max} ; времена установки и удержания сигналов, задержки распространения сигнала между выводами ПЛИС; время компиляции и др.

Для просмотра нужного раздела отчета о компиляции выполните следующие действия:

1. Чтобы развернуть раздел окна **Compilation Report** (отчет о компиляции) в левой части окна, щелкните по значку "+" у соответствующего раздела (рис. 3.41).

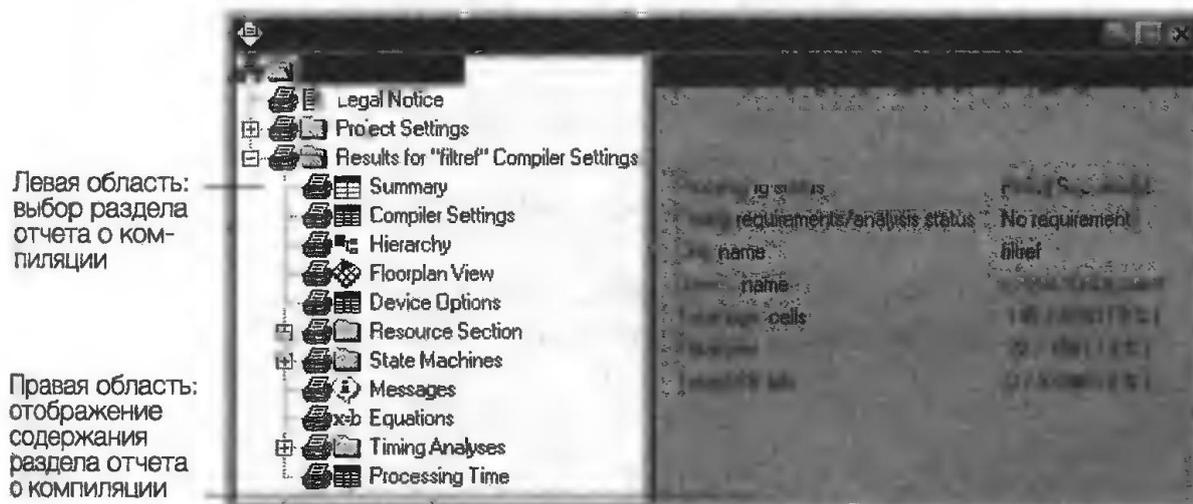


Рис. 3.41

2. Выберите (выделите) нужный раздел отчета — информация данного раздела появится в правой области окна.

Дополнительную информацию о результатах анализа временных характеристик (**Timing Analysis**) можно посмотреть в уроке 10 учебного пособия "Просмотр результатов анализа временных характеристик". Дополнительную информацию можно также найти в справочной системе пакета Quartus в разделе "Overview: Viewing the Results of a Compilation or Simulation in the Report Window" (просмотр информации о компиляции или моделировании в окне отчета).

Урок 8. Просмотр текущей компоновки проекта, созданной компилятором

В системе Quartus возможны два режима просмотра компоновки проекта.

- **Current Assignments floorplan** – размещение, соответствующее текущим пользовательским назначениям ресурсов проекта. В данном режиме размещение доступно для редактирования.
- **Last Compilation floorplan** – текущее размещение проекта, созданное компилятором при последней компиляции. В данном режиме размещение недоступно для редактирования.

Редактор физического размещения (**Floorplan Editor**) позволяет просмотреть распределение проекта по логическим ячейкам (**Logic cells**) ПЛИС, логическим блокам (**LAB**), структурам **MegaLAB**, а также увидеть схематическое изображение корпуса и расположение выводов используемой ПЛИС.

 *Размещение и разводка проекта в ПЛИС могут отличаться от показанных на иллюстрациях из-за усовершенствований в новых версиях пакета Quartus.*

1. Открытие текущей топологии проекта, созданной последней компиляцией

В этом уроке описываются следующие действия:

1. Открытие текущей топологии проекта, созданной последней компиляцией.
2. Отображение информации о разводке внутренних связей в ПЛИС.
3. Отображение логических выражений, реализующих составляющие проекта в ПЛИС.
4. Отображение структур **MegaLAB**.

Чтобы открыть текущую топологию проекта, созданную последней компиляцией, выполните следующие действия:

1. В меню **Processing** (обработка) выберите команду **Open Last Compilation Floorplan** (открыть текущее размещение проекта, созданное компилятором).
2. Для увеличения размера окна редактора **Floorplan Editor** (редактор физического размещения) включите в меню **View** функцию **Full View**. По умолчанию топология показывается в режиме отображения логических ячеек (**Logic Cell**). Показываются логические ячейки каждого массива логических блоков (**LAB – Logic Array**

Block). Неиспользуемые логические ячейки отображаются белым цветом, используемые — определенными цветами.

3. Информация о назначении используемых цветов выводится в окне **Color Legend Window** (окно цветовых обозначений), которое вызывается соответствующей командой в меню **View** редактора **Floorplan Editor** (редактор физического размещения).

2. Отображение информации о разводке связей проекта

Редактор физического размещения позволяет просмотреть входящие связи (**Fan-In**) и выходящие связи (**Fan-out**) каждого узла ПЛИС.

Для просмотра входящих и исходящих связей, выполните следующие действия:

1. В меню **View** выберите команду **Routing > Show Node Fan-In & Fan-Out**.
2. В первом **LAB** (логическом блоке) — **LAB1**, выберите седьмую сверху логическую ячейку (**Logic Cell**) — **LC7**. Входящие связи логической ячейки обозначены темно-сиреневыми линиями, а выходящие — темно-синими. Направление связей обозначается стрелками (рис. 3.42).

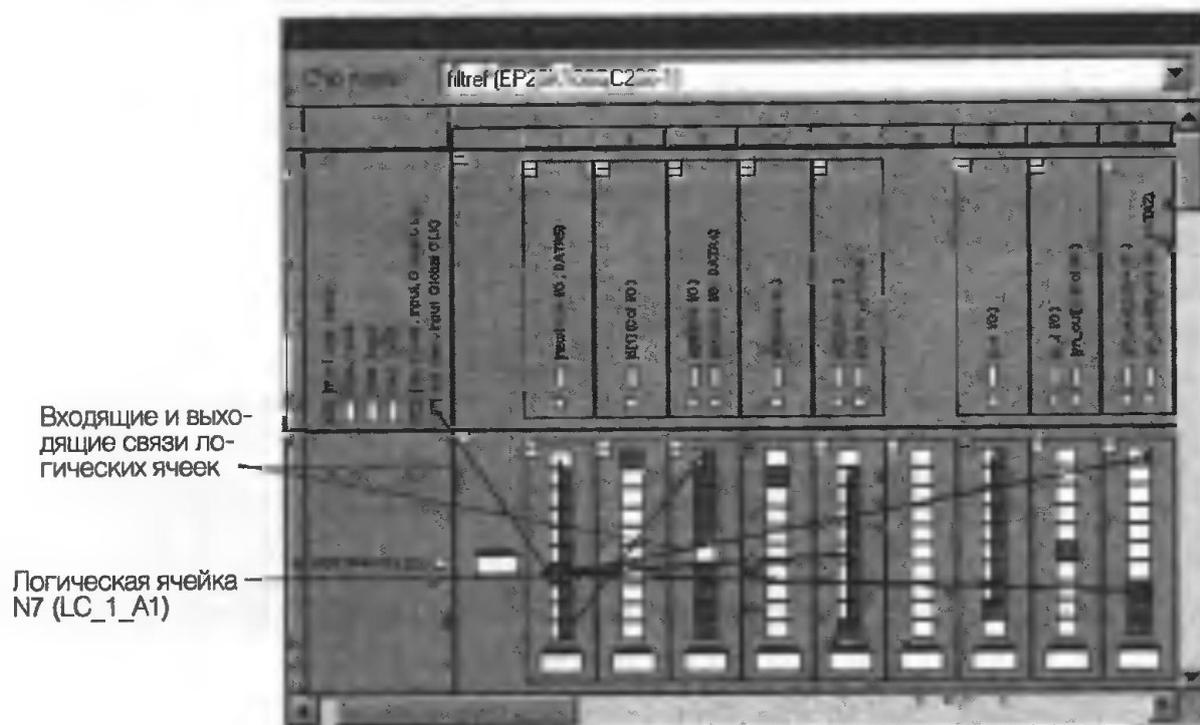


Рис. 3.42

3. Для отображения задержек на путях распространения сигналов, включите **Routing > Show Routing Delays** (меню **View**). Задержки, относящиеся к выбранным логическим ячейкам, показываются на топологии проекта рядом с входящими и выходящими связями.

- После просмотра входящих и выходящих связей, выберите в меню **View** команду **Routing > Hide Routing** (разводка > не показывать) (спрятать) разводку).

3. Информация, отображаемая в окне *Equations*

Окно **Equations** размещается в нижней части окна редактора **Floorplan Editor** (редактор физического размещения).

В окне **Equations** отображаются логические выражения, а также текстовое описание входящих и выходящих связей выбранных логических ячеек либо выводов ПЛИС.

Чтобы посмотреть информацию для входного вывода `reset`, выполните следующие действия:

- Включите отображение окна **Equations**, выбрав в меню **View** команду **Equations**.
- В режиме **Last Compilation Floorplan** (размещение, созданное последней компиляцией), в верхнем левом углу топологии, выберите вывод `reset`. В окне **Equations** будет выведено выражение, а также выходящие связи вывода `reset`. Обратите внимание, что левая часть окна **Fan-In** (входящие связи) остается пустой, поскольку `reset` является входом и у него нет входящих связей, но есть только выходящие (рис. 3.43).



Рис. 3.43

- Для просмотра информации об узле `E1L1Q(|state_m:inst1|filter~28)` выполните следующие действия:
 - вверху списка **Fan-Out** (выходящие связи) выберите имя регистрового узла `E1L1Q(|state_m:inst1|filter~28)`;
 - нажмите **Go To** (перейти) — в списке **Equations** появятся реализуемые узлом логические выражения;
 - для раскрытия содержания термина щелкните левой кнопкой мыши на первом, помеченном синим цветом, терме выражения из списка **Equations** (рис. 3.44).

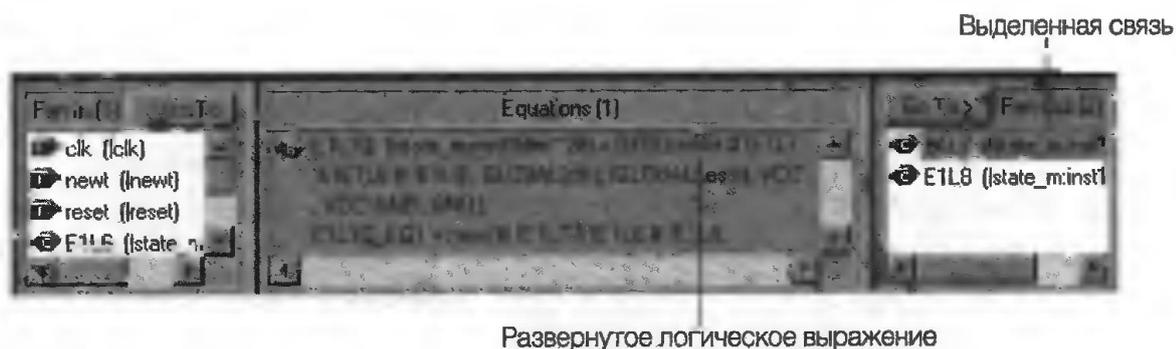


Рис. 3.44

4. Отображение блоков MegaLAB

Блоки MegaLABs реализованы в микросхемах семейства APEX 20K. Каждый блок MegaLAB представляет собой структуру, объединяющую 16 логических блоков LAB (Logic Array Block), один блок встроенной памяти ESB (Embedded System Block) и систему внутренних соединений блока MegaLAB (MegaLAB Interconnect).

Для просмотра блоков MegaLABs в окне редактора физического размещения (Floorplan Editor) выполните следующие действия:

1. В меню **View** выберите **MegaLAB Interior** (режим отображения блоков MegaLAB). Блоки MegaLAB в редакторе топологии отображаются в виде прямоугольников. Неиспользованные ресурсы блоков MegaLAB по умолчанию показываются белым цветом. Используемые ресурсы — цветами, обозначающими использованные для разводки внутренние соединения (рис. 3.45).

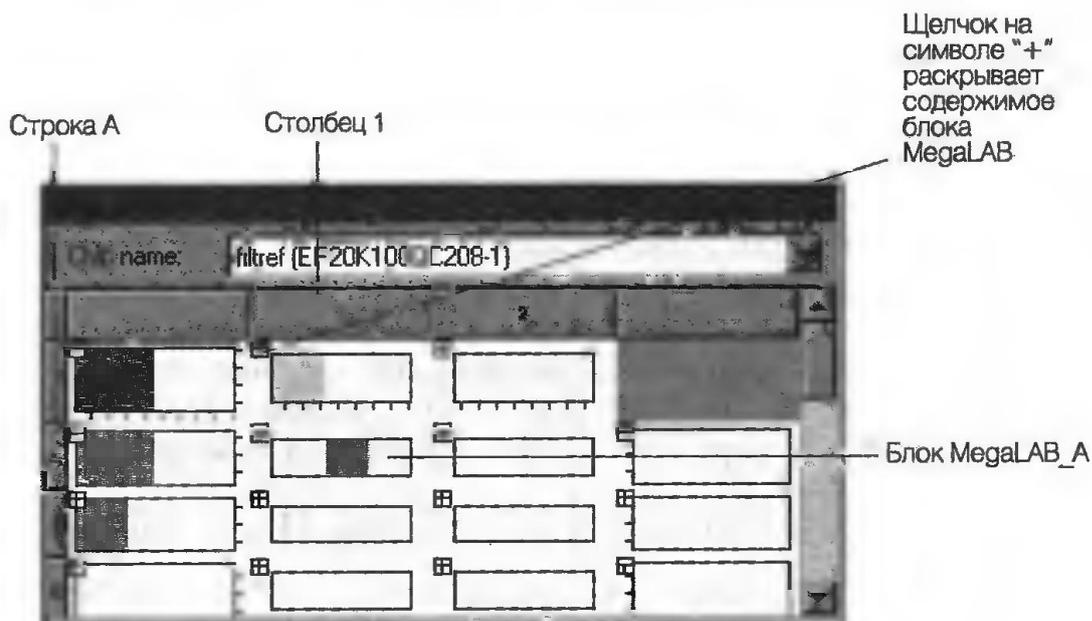


Рис. 3.45

2. Чтобы развернуть структуру **MegaLAB_A1**, щелкните по символу "+" рядом с прямоугольником блока, расположенного в строке 1 столбца А. Блок MegaLAB раскроется, и станут видны его LAB (логические блоки).
3. В меню **View** выключите режим **Full View** (полный вид).
4. В меню **View** выберите **Interior Cells** (внутренние ячейки).
5. Закройте топологию, созданную последней компиляцией (**Last Compilation Floorplan**), выбрав в меню **File** команду **Close**.

Урок 9. Реализация логических функций в блоках ESB

Система Quartus позволяет задать принудительное размещение логики составляющих проекта в блоках встроенной памяти ESB (**Embedded System Block**).

Блоки встроенной памяти ESB реализованы в микросхемах семейства APX20K и могут быть использованы для организации встроенной оперативной (RAM) либо постоянной (ROM) памяти, а также для реализации определенных логических функций в макроячейках матричной логики блоков ESB.

В этом уроке описываются следующие действия:

1. Сохранение (копирование) назначений ресурсов проекта, выполненных компилятором.
2. Изменение сохраненных назначений ресурсов.
3. Выполнение локальных назначений ресурсов проекта.
4. Перекомпиляция проекта после внесенных изменений.
5. Просмотр размещения проекта на ПЛИС, выполненного компилятором.

1. Сохранение назначений ресурсов, выполненных компилятором

Для сохранения назначений ресурсов, выполненных компилятором используется функция **back-annotation**. Эта функция позволяет сохранить (скопировать) выполненные компилятором назначения выбранных ресурсов в качестве текущих пользовательских.

Эту функцию удобно использовать при необходимости изменения назначений ресурсов проекта. При этом сначала выполненные компилятором назначения сохраняются в качестве текущих пользовательских, затем последние подвергаются корректировке в соответствии с требованиями разработчика.

Для сохранения назначений выводов и типа ПЛИС выполните следующие действия:

1. В меню **Processing** выберите пункт **Back-annotate Assignments** (сохранение назначений ресурсов).

2. В диалоговом окне **Back-annotate Assignments** под строкой **Assignment(s) to Back-annotate** (сохраняемые назначения ресурсов) выберите **Pin & device assignments** (назначения выводов и типа ПЛИС) (рис. 3.46).

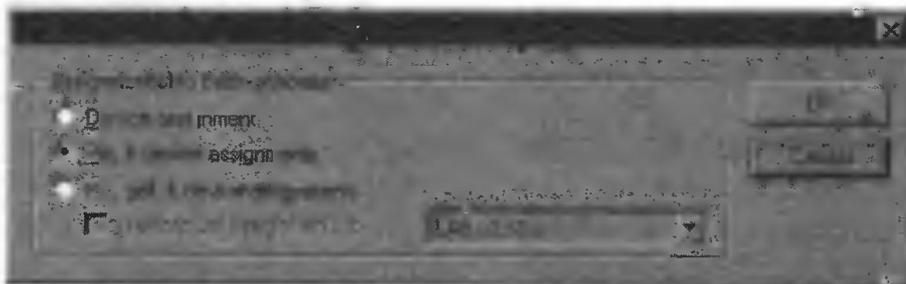


Рис. 3.46

3. Нажмите кнопку **ОК**.

2. Проверка сохранения назначений ресурсов

Чтобы убедиться, что функция **back-annotation** сохранила выполненные компилятором назначения ресурсов, откройте в редакторе **Floorplan Editor** (редактор топологии) топологию **Current Assignments Floorplan** (топологию, заданную назначениями пользователя).

Для проверки сохранения назначений ресурсов выполните следующие действия:

1. В меню **Processing** откройте **Current Assignments** (текущие назначения ресурсов).
2. В меню **View** включите режим **Full Screen** (отображение на весь экран). В верхней части окна отображается список имен (**Chip Name**), присвоенных ПЛИС и заданный вами (пользователем) тип ПЛИС (**Device**) — EP20K100QC208-1. Кроме того, теперь на топологии **Current Assignments floorplan** (размещение, заданное назначениями пользователя) отображаются назначения выводов, выполненные компилятором при последней компиляции и соответствующие назначениям выводов в топологии **Last Compilation floorplan** (текущее размещение, созданное компилятором). Обратите внимание, поскольку при выполнении функции **Back-annotate** не было выбрано сохранение назначений логических ячеек (**Logic Cells**), все логические ячейки на топологии **Current Assignments floorplan** (топологии, соответствующей текущим назначениям ресурсов) пусты (отображаются белым цветом).
3. Выключите в меню **View** режим **Full Screen** (полный экран).
4. Закройте редактор размещения (**Floorplan Editor**), выбрав в меню **File** команду **Close** (закреть).

3. Назначения реализации логических функций в блоках встроенной памяти

Чтобы назначить реализацию логики объекта `state_m:inst1` в блоках ESB, выполните следующие действия:

1. В навигаторе проекта (**Project Navigator**) перейдите на вкладку **Hierarchies** (иерархия).

 *Окно подсистемы **Project Navigator** появляется автоматически при запуске системы **Quartus**. Однако при необходимости **Project Navigator** можно вызвать через меню **View > Auxiliary Windows > Project Navigator** (вид > вспомогательные окна > навигатор проекта).*

2. Раскройте иерархию проекта **filtref**, щелкнув по символу "+" на вкладке **Hierarchies** (рис. 3.47)

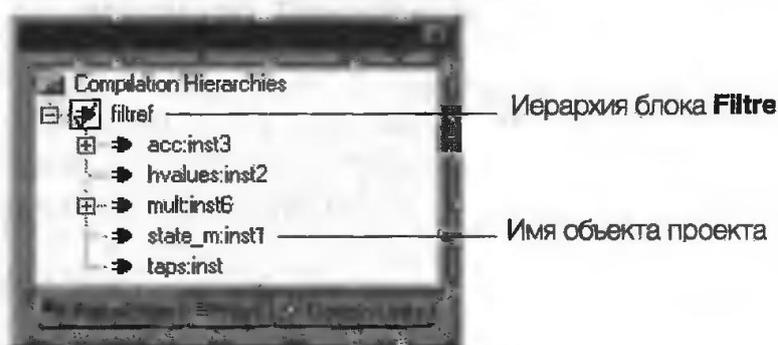


Рис. 3.47

3. Выберите в иерархии проекта **filtref** имя объекта `state_m:inst1`.
4. Нажмите правую кнопку мыши и во всплывающем меню выберите **Assignment Organizer** (организатор назначений ресурсов). Откроется диалоговое окно **Assignment Organizer**. В окне будет выбрана опция **Edit specific entity & node settings for:** (редактировать параметры указанного объекта и узла). В окне **Name** будут показаны иерархический путь и имя объекта `state_m:inst1`.
5. В списке **Assignment Categories** (категории назначений ресурсов), щелкните по символу "+", чтобы развернуть **Options for Entities Only** (параметры только объектов).
6. Щелкните по **Click here to add a new assignment text**.
7. Под **Assignment** (назначения ресурсов), в списке **Name** (имя), выберите **Technology Mapper** (технологический разводчик).
8. В списке **Setting** (установки), выберите **Product term** (матричная логика) (рис. 3.48).
9. Нажмите **Add** (добавить). Назначение появится в списке **Assignment Categories** (категории назначений ресурсов).
10. Нажмите **OK**.

Кнопка обзора (**Browse**), открывающая окно поиска объектов (**Node Finder**)

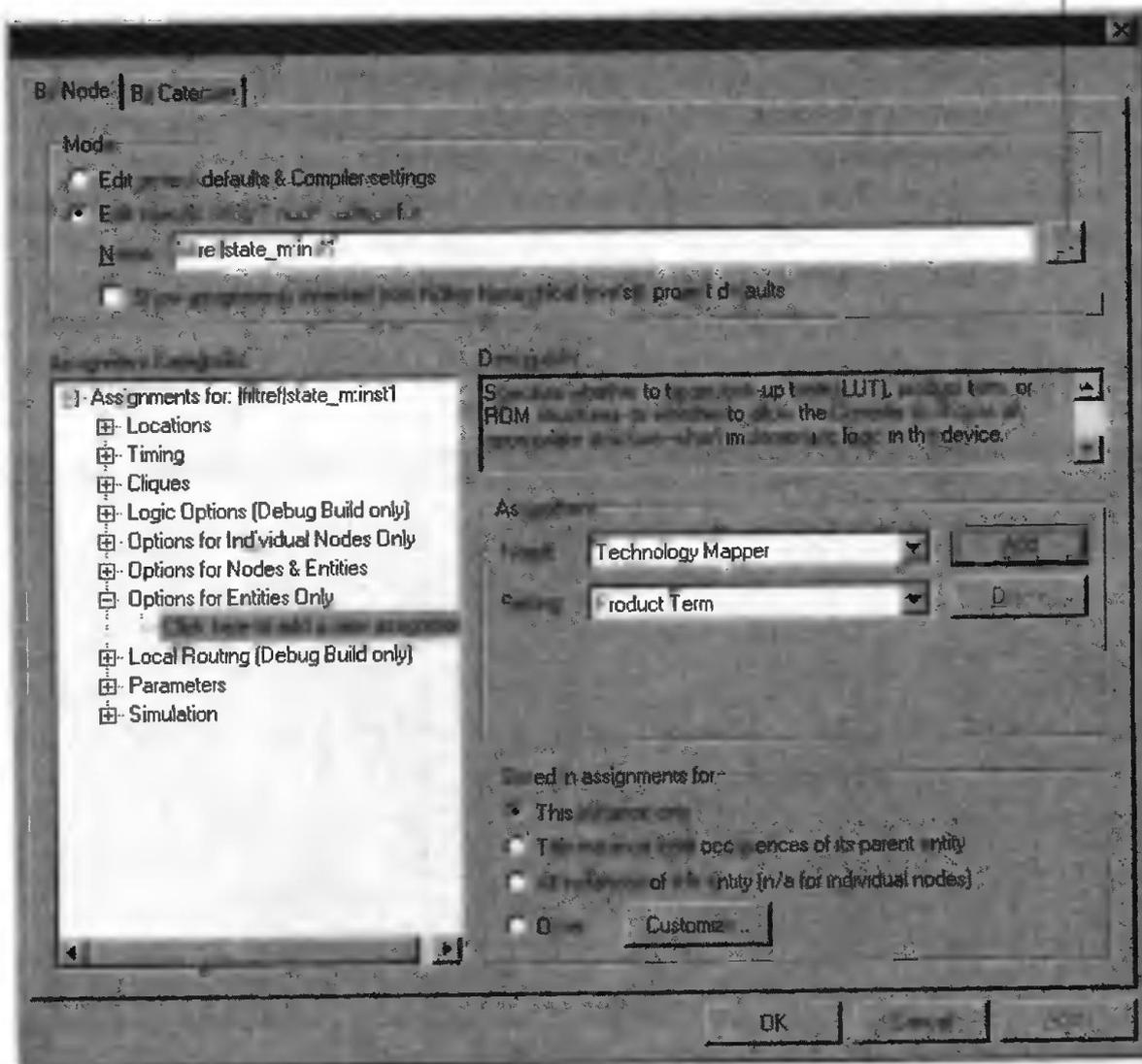


Рис. 3.48

4. Перекомпиляция проекта

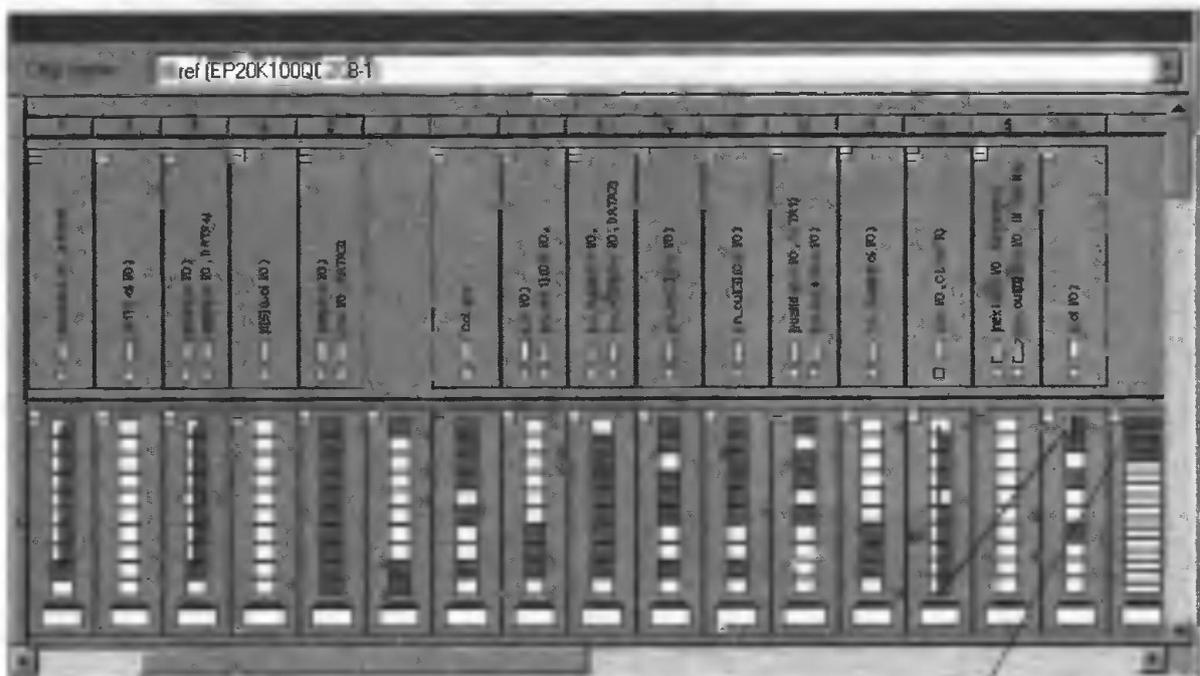
Чтобы перекомпилировать проект с новыми сделанными вами назначениями ресурсов, в меню **Processing** (обработка) выберите **Start Compilation** (запуск компиляции).

5. Просмотр текущей компоновки проекта, созданной компилятором

Для просмотра текущей компоновки проекта, созданной компилятором при последней компиляции, выполните следующие действия:

1. Откройте топологию, выбрав в меню **Processing** (обработка) **Last Compilation Floorplan** (текущее размещение, созданное последней компиляцией).

2. Для увеличения размера окна редактора **Floorplan Editor** (редактора физического размещения) в меню **View** (вид) включите режим **Full Screen** (полный экран). Обратите внимание, что назначения выводов сохранились от предыдущей компоновки, однако теперь логика реализована в блоках встроенной памяти — **ESBs**.
3. Щелкните по кнопке **Selection Tool** (инструмент выбора) на панели инструментов редактора **Floorplan Editor**.
4. Указателем **Selection Tool** укажите на некоторые из используемых встроенных блоков памяти **ESBs**, расположенные в центре строки 1. В окнах всплывающего текста ("**bubble text**") будут показаны относящиеся к этим блокам названия сигналов. Обратите внимание, что поскольку все сигналы относятся к объекту `state_m`, их имена начинаются со "`state_m`" (рис. 3.49).



Блоки встроенной памяти (ESB)

Рис. 3.49

5. Выключите режим **Full Screen** (отображение на весь экран) в меню **View** (вид).

3.4. Анализ временных характеристик проекта

С помощью подсистемы **Timing Analyzer** (анализатор временных характеристик) пакета **Quartus** вы можете анализировать быстродействие всей синтезированной компилятором логики. Можно проследить пути распространения сигналов в редакторе физического размещения (**Floorplan Editor**) и определить критические для быстродействия пути сигнала.

Анализатор временных характеристик (**Timing Analyzer**) пакета Quartus автоматически запускается по окончании компиляции. Кроме того, **Timing Analyzer** можно запустить вручную, перед этим необходимо хотя бы раз выполнить компиляцию проекта.

По умолчанию анализатор временных характеристик вычисляет следующие характеристики: максимальную частоту f_{\max} переключений каждого регистра, критические задержки на путях "регистр-регистр" для наихудшего случая, времена t_{SU} установки входных сигналов, времена t_{H} удержания входных сигналов для каждого входного регистра, задержки t_{CO} на пути "тактовый сигнал — выход" (**clock-to-output**) для каждого выходного регистра в иерархии проекта. Кроме того, анализируются все пути распространения сигналов "вывод-вывод" (**pin-to-pin**) и вычисляются транспортные задержки t_{PD} сигнала между выводами.

После окончания компиляции, информация о результатах анализа временных параметров приводится в соответствующем разделе отчета компилятора (**Compilation Report**) (рис. 3.50).

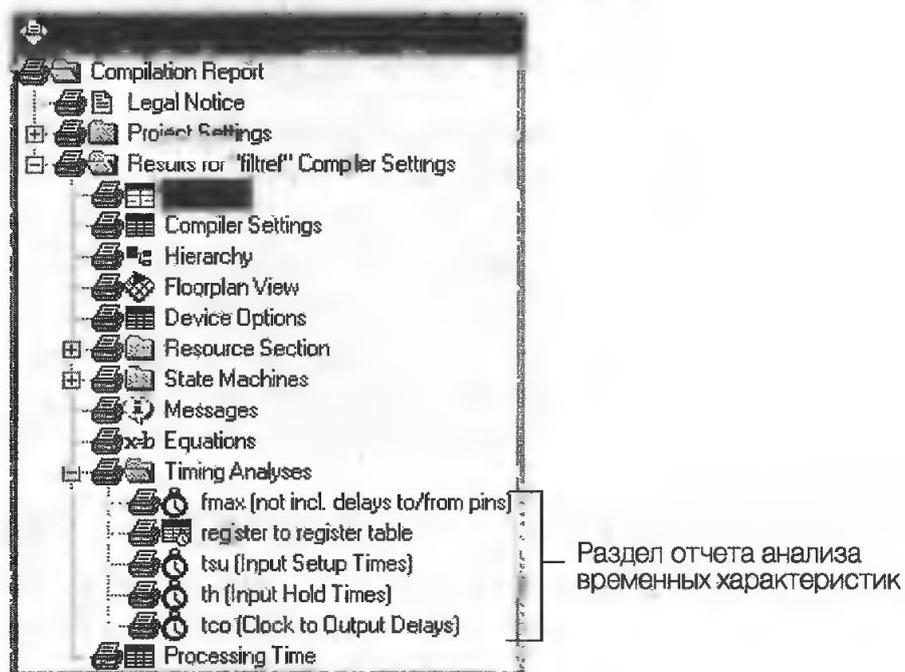


Рис. 3.50

В уроках 10–15 описываются следующие действия:

- просмотр результатов анализа временных характеристик в окне отчета о компиляции;
- определения требований к быстродействию;
- выполнения многотактового анализа временных характеристик;
- назначения ветвей многотактовой синхронизации.

Урок 10. Просмотр результатов анализа временных характеристик

После компиляции проекта результаты анализа временных характеристик (**Timing analysis results**) можно посмотреть в отчете о компиляции (**Compilation report**).

Ниже рассматривается выполнение следующих действий:

1. Просмотр максимальной частоты внутренних переключений f_{\max} .
2. Просмотр списка временных путей f_{\max} .
3. Локализация временных путей на топологии проекта f_{\max} .
4. Просмотр времен установки входных сигналов t_{SU} .

Информацию о просмотре других временных характеристик см. в справочной системе пакета Quartus в разделе "Viewing Timing Analysis Results in the Report Window" (просмотр результатов анализа временных характеристик в окне отчета).

1. Просмотр максимальной частоты внутренних переключений f_{\max}

В разделе f_{\max} отчета **Compilation Report** (отчет о компиляции) показываются ограничения по частоте внутренних переключений и быстродействию проекта для наихудшего случая. Показывается, учитываются ли при вычислении частоты f_{\max} задержки сигнала на входящих и выходящих связях выводов ПЛИС, которые могут быть заданы на вкладке **Clock Settings** (параметры тактового сигнала) диалогового окна **Timing Settings** (задание временных характеристик), вызываемого из меню **Project** (проект).

Чтобы открыть раздел отчета о f_{\max} , выполните следующие действия:

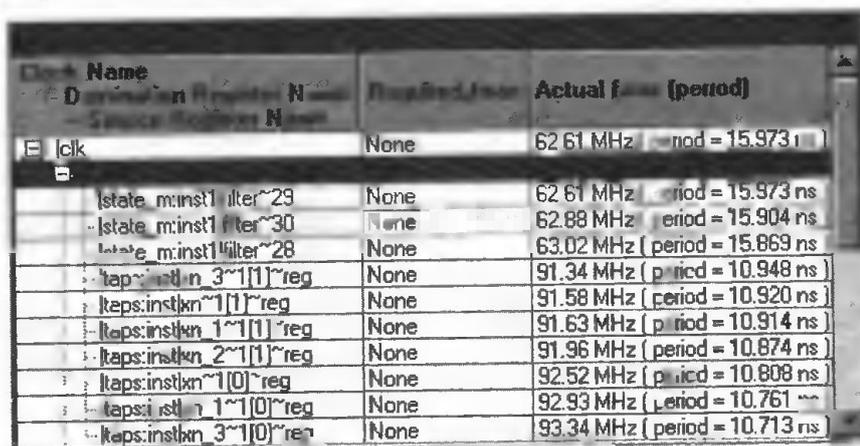
1. Отчет о компиляции можно открыть, выбрав в меню **Processing** (обработка) команду **Open Compilation Report** (открыть отчет о компиляции).
2. Чтобы развернуть содержимое раздела **Timing Analyses** (анализ временных характеристик), щелкните по символу "+" соответствующего раздела в левой области окна **Compilation Report** (отчет о компиляции).
3. В разделе **Timing Analyses** (анализ временных характеристик) выберите подраздел f_{\max} — информация о f_{\max} появится в таблице. Красным цветом отображаются фактические значения временных характеристик, не удовлетворяющие заданным требованиям.
4. Чтобы развернуть список регистров, синхронизируемых тактовым сигналом, щелкните по символу "+". По умолчанию в списке показаны 10 регистров с наименьшими частотами переключений.

Дополнительную информацию о настройке сообщений анализатора временных характеристик см. в справочной системе (Help) пакета Quartus в разделе "Specifying Timing Analysis Reporting Restrictions" (уточнение сообщений анализа временных характеристик).

2. Просмотр списка временных путей f_{\max}

Чтобы просмотреть список временных путей f_{\max} выполните следующие действия:

1. Для раскрытия списка и отображения путей от регистров-источников к регистрам-приемникам сигнала, щелкните по символу "+". При этом в разделе f_{\max} раскроется имя первого регистра-приемника. По умолчанию в списке показывается 10 самых медленных регистров (рис. 3.51).



Name	Register/Source	Actual f_{\max}	(period)
clk	None	62.61 MHz	period = 15.973 ns
state_minst1_filter29	None	62.61 MHz	period = 15.973 ns
state_minst1_filter30	None	62.88 MHz	period = 15.904 ns
state_minst1_filter28	None	63.02 MHz	period = 15.869 ns
taps:instkn_3[1]reg	None	91.34 MHz	period = 10.948 ns
taps:instkn_1[1]reg	None	91.58 MHz	period = 10.920 ns
taps:instkn_1[1]reg	None	91.63 MHz	period = 10.914 ns
taps:instkn_2[1]reg	None	91.96 MHz	period = 10.874 ns
taps:instkn_1[0]reg	None	92.52 MHz	period = 10.808 ns
taps:instkn_1[0]reg	None	92.93 MHz	period = 10.761 ns
taps:instkn_3[1]reg	None	93.34 MHz	period = 10.713 ns

Рис. 3.51

Фактический источник и/или имена регистров-источников могут отличаться от показанных на этом рисунке из-за усовершенствований сделанных в новых версиях пакета Quartus после выхода этого учебного пособия.

2. Выберите имя первого регистра-источника в списке.
3. Во всплывающем меню щелкните по правой кнопке мыши и выберите список путей (List Paths). Во вкладке System (система) окна Messages (сообщения) будут показаны задержки распространения сигналов для регистра-источника, включая среднюю задержку на путях сигнала.
4. Чтобы раскрыть сообщение о частоте f_{\max} в окне Messages (сообщения) щелкните по символу "+". В окне будут показаны наибольшая задержка "регистр-регистр", наименьшая крутизна фронта тактового сигнала и сообщения о задержках установки сигналов.

5. Чтобы растянуть сообщение о максимальной задержке "регистр-регистр", щелкните по символу "+". Для вычисления отображенной в этом окне задержки на всем пути распространения сигнала используются средние приращения времени задержки. На следующем рисунке показан пример типовых сообщений о задержках. Значения для конкретной ситуации могут существенно отличаться от показанных (рис. 3.52).

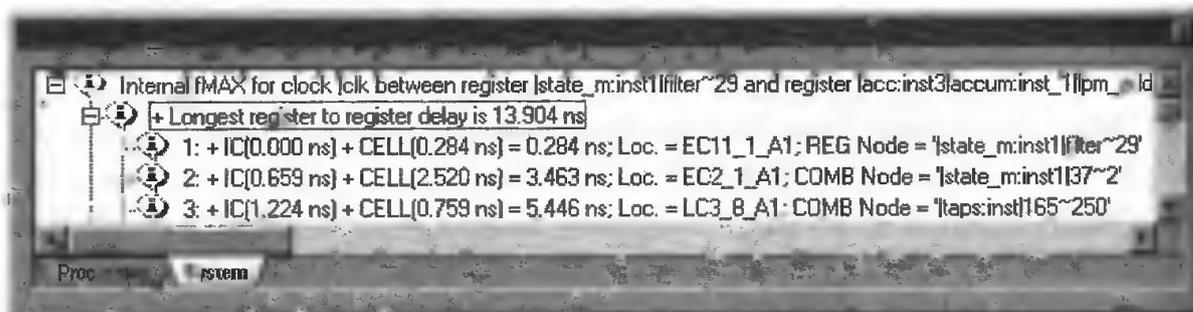


Рис. 3.52

3. Локализация временных путей f_{max} в редакторе топологии

Для локализации источника сообщений о задержках на путях распространения сигналов на топологии **Last Compilation floorplan** (текущее размещение, созданное компилятором) выполните следующие действия:

1. В окне **Messages** (сообщения) выберите сообщение **Longest register to register** (наиболее длинный путь "регистр-регистр").

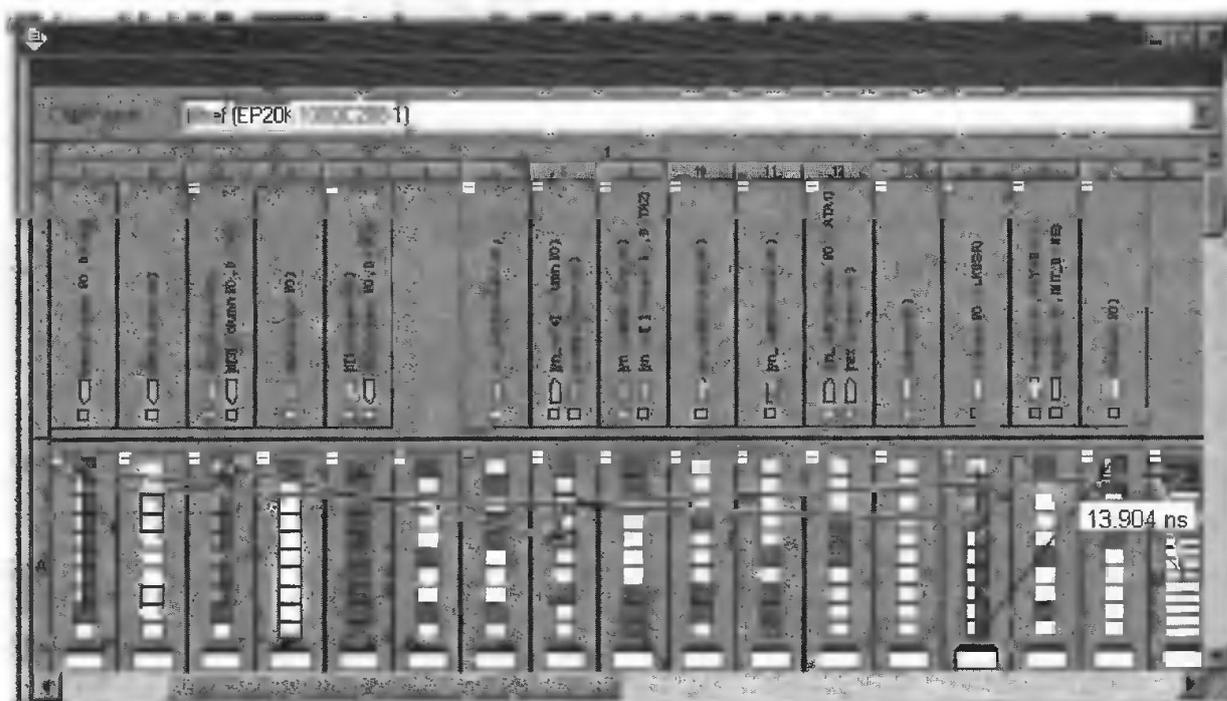


Рис. 3.53

2. Во всплывающем меню щелкните по правой кнопке мыши и выберите **Locate** (локализовать). На топологии проекта выделится путь данных и рядом появится общее время задержки сигнала на этом пути (рис. 3.53).

 Фактический путь и задержка сигнала могут отличаться от показанных на рисунке из-за усовершенствований сделанных в новых версиях пакета Quartus после выхода этого учебного пособия.

3. Для закрытия топологии **Last Compilation floorplan** (текущее размещение, созданное компилятором) в меню **File** (операции с файлами) выберите **Close** (закрыть).

4. Просмотр отчета о временах установки сигналов

В разделе t_{SU} отчета о компиляции (**Compilation Report**) отображаются заданные и фактические значения времени установки t_{SU} сигналов на выводах связанными со входами данных или входами разрешения синхронизации триггеров (регистров)-приемников.

Чтобы открыть раздел отчета о временах t_{SU} , выполните следующие действия:

1. Для раскрытия раздела **Timing Analyses** (анализ временных характеристик) отчета о компиляции (**Compilation Report**) щелкните по символу "+" в левой области соответствующего окна.
2. В разделе **Timing Analyses** (анализ временных характеристик) выберите раздел t_{SU} . В таблице будет отображена информация о временах установки входных сигналов.
3. Чтобы раскрыть список и увидеть имена регистра и связанных с выводами тактовых сигналов, щелкните по символу "+".
4. Чтобы вернуться к блок-схеме проекта **filtref.bdf**, закройте окно **Compilation Report** (отчет о компиляции).

 Выполняя действия, аналогичные описанным выше, вы можете также просматривать в отчете **Compilation Report** пути регистр-регистр для времен t_{1F} , t_{CO} и t_{PD} . Дополнительную информацию см. в справочной системе (**Help**) пакета Quartus в разделе "Viewing Timing Analysis Results in the Report Window" (просмотр результатов анализа временных характеристик в окне отчета).

Урок 11. Задание требуемых временных характеристик

Пакет Quartus позволяет задавать требуемые временные характеристики (**Timing Requirements**) проекта в целом, а также индивидуально для каждого объекта, узла и вывода. Например, вы можете задать требуемые времена: t_{SU} установки сигналов (**setup time**), t_H удержания (**hold time**), задержек t_{CO} "тактовый сигнал — выход" (**clock-to-output**), задержек t_{PD} распространения сигналов между выводами (**pin-to-pin**), а также максимальную частоту внутренних переключений f_{max} .

Требуемые временные характеристики (**Timing Requirements**) проекта в целом определяются в окне **Timing Settings**, вызываемого из меню **Project**. Требования к временным характеристикам индивидуально для объектов, узлов и выводов определяются с помощью **Assignment Organizer** из меню **Tools**.

В этом уроке описываются следующие действия:

1. Определение, требуемой частоты f_{max} .
2. Исключение временных путей (**timing paths**) из анализа.

 Здесь рассматривается определение требуемых временных характеристик, а также других параметров анализа временных характеристик через команды меню и соответствующие им диалоговые окна. Однако для упрощения работы можно использовать встроенный мастер **Timing Wizard** (мастер задания временных характеристик), вызываемый соответствующей командой из меню **Project**.

1. Задание требуемой f_{max}

Вы можете задать требуемую частоту внутренних переключений для реализации проекта на выбранной микросхеме с помощью установки f_{max} . Задание значения f_{max} можно выполнить как для всего проекта, так и для выбранных частей проекта.

Чтобы определить требуемое значение f_{max} , выполните следующие действия:

1. В меню **Project** выберите **Timing Settings** (временные характеристики) — автоматически откроется вкладка **Clock Settings** (параметры тактового сигнала) (рис. 3.54).
2. Под надписью **Specify circuit frequency as** (определить частоту схемы как...) выбрать **Default required fmax** (частота f_{max} , заданная по умолчанию).
3. В поле **Default required fmax** (частота f_{max} , заданная по умолчанию), введите значение 45 и выберите **MHz** (мегагерцы) из списка единиц измерения (рис. 3.54).

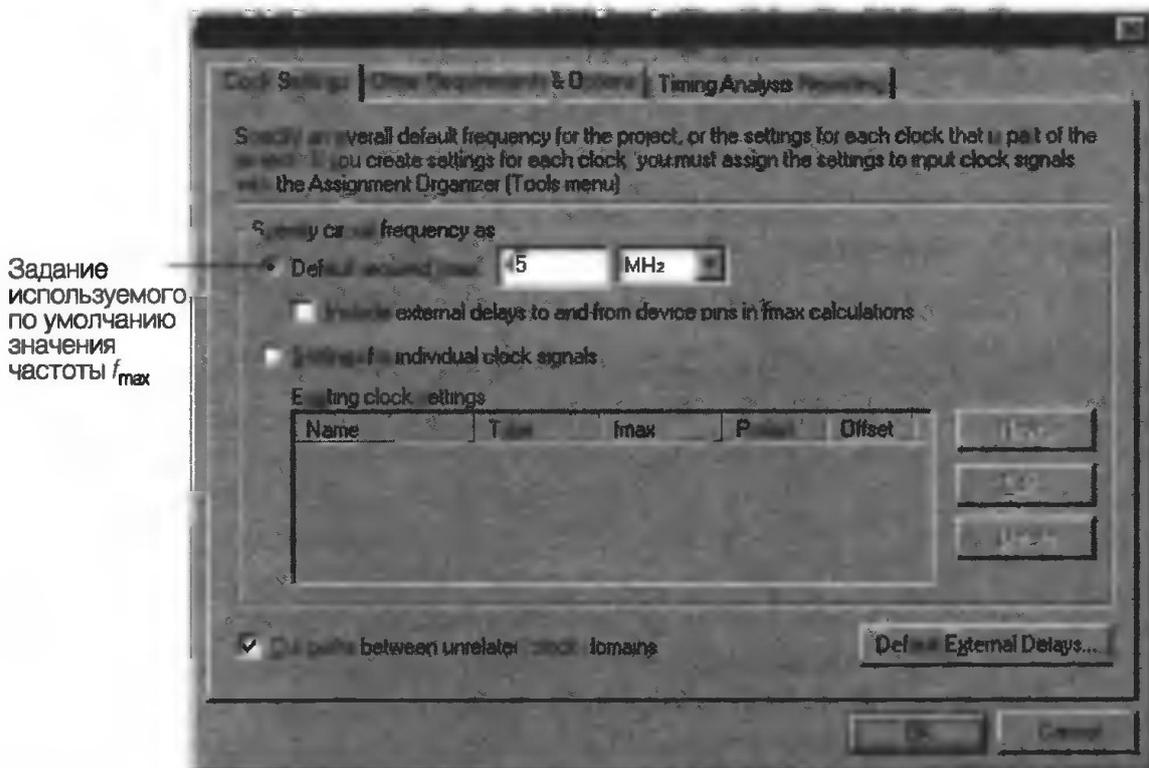


Рис. 3.54

 После определения требуемой f_{\max} для проекта в целом, вы можете изменить эти назначения индивидуально для конкретных тактовых сигналов, как описано в уроке 12 "Анализ временных характеристик при многотактовой синхронизации".

2. Исключение временных путей из анализа

При выполнении анализа временных характеристик вы можете исключать из анализа определенные пути сигнала на вкладке **Other Requirements & Options** (другие требования и параметры).

Чтобы исключить определенные категорий путей или определенные пути по принципу "от точки — к точке", вы можете выбрать опции **Cut Timing Path** в диалоговом окне организатора назначений **Assignment Organizer**.

Чтобы отключить обратную связь через двунаправленные выводы, выполните следующие действия:

1. В диалоговом окне **Timing Settings** (установка временных характеристик) щелкните по вкладке **Other Requirements & Options** (другие требования и установки).
2. Убедитесь, что пункт **Cut off feedback from I/O pins** (отключение обратной связи через двунаправленные выводы) включен.
3. Нажмите кнопку **ОК**. Теперь при анализе временных характеристик все обратные связи (через **I/O pins**) ПЛИС будут проигнорированы.

- Дополнительную информацию по установке значений t_{SU} , t_H , t_{CO} и t_{PD} см. в справочной системе (**Help**) пакета Quartus в разделе "Specifying Project-Wide Timing Requirements" (определение требуемых временных характеристик всего проекта).

Урок 12. Анализ временных характеристик проектов с многотактовой синхронизацией

Подсистема **Timing Analyzer** (анализатор временных характеристик) пакета Quartus в процессе компиляции автоматически выполняет анализ временных характеристик для одного тактового сигнала. Однако возможно также выполнение анализа временных характеристик проектов, использующих сложную многотактовую синхронизацию.

Чтобы выполнить анализ временных характеристик при многотактовой синхронизации, сначала необходимо определить соотношения между всеми используемыми тактовыми последовательностями. Для определения этих соотношений необходимо определить основные (первичные) синхросигналы (**absolute clocks**), не зависящие от других синхросигналов, а также вторичные синхросигналы (**derived clocks**), являющиеся производными от других синхросигналов. Пакет Quartus позволяет определить требования к каждому тактовому сигналу.

В этом уроке описываются следующие действия:

1. Задание параметров тактового сигнала.
2. Привязка параметров тактового сигнала к выводам микросхемы.
3. Повторное выполнение анализа временных характеристик.
4. Просмотр раздела **Clock Requirement** (требования к тактовому сигналу).

1. Задание параметров тактового сигнала

Для определения характеристик первичного тактового сигнала, выполните следующие действия:

1. В меню **Project** выберите команду **Timing Settings** (установка временных характеристик) — автоматически открывается вкладка **Clock Settings** (установки тактовых сигналов).
2. Под надписью **Specify circuit frequency as** (определить частоту схемы как...) выберите **Settings for individual clock signals** (индивидуальная установка параметров тактовых сигналов).
3. Нажмите на кнопку **New** (новые). Откроется новое диалоговое окно **New Clock Settings** (новые установки тактового сигнала).
4. В поле **Clock settings name** (имя группы параметров тактового сигнала) в качестве названия новой группы параметров тактового сигнала введите **clocka**.

- Чтобы определить параметры тактового сигнала как параметры первичного тактового сигнала, убедитесь, что под надписью **Relationship to other clock settings** (отношение к другим тактовым сигналам) выбран тип **Independent of other clock settings** (независимый от параметров других тактовых сигналов).
- Для определения требуемой f_{\max} первичного тактового сигнала, в поле **Required fmax** (требуемая f_{\max}), введите 50 и из списка единиц измерения выберите **MHz** (мегагерц) (рис. 3.55).

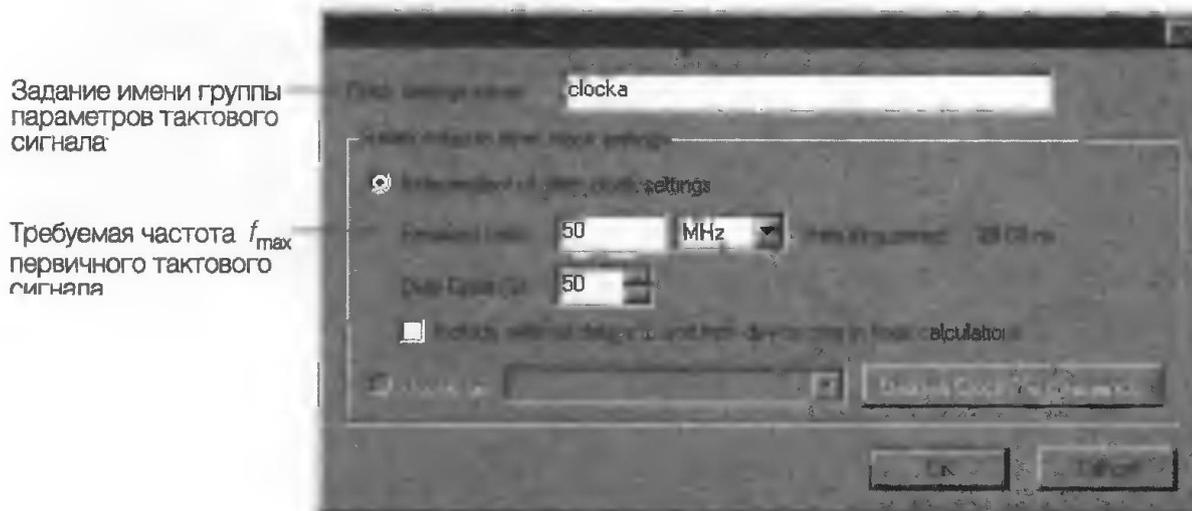


Рис. 3.55

- Нажмите кнопку **OK** — параметры сигнала clocka появятся в списке **Existing clock settings** (существующие параметры тактового сигнала).

Чтобы установить параметры зависимого (вторичного) тактового сигнала, выполните следующие действия:

- Нажмите на кнопку **New**. Откроется новое диалоговое окно **New Clock Settings** (новые параметры тактового сигнала).
- Нажмите на кнопку **New**. Откроется новое диалоговое окно **New Clock Settings** введите clockb.
- Под надписью **Relationship to other clock settings** (отношение к другим тактовым сигналам) выберите **Based on** (основанный на...) и выберите из списка clocka.
- Чтобы определить требуемые характеристики вторичного (зависимого) тактового сигнала, выберите **Derived Clock Requirements** (требования к зависимому (вторичному) тактовому сигналу). Откроется диалоговое окно **Derived clock Requirements**.
- Чтобы задать частоту f_{\max} вторичного тактового сигнала в два раза большую частоты основного тактового сигнала, выберите **2** в окне **Multiply base absolute clock fmax by** (умножить частоту основного тактового сигнала на...).

6. Чтобы задать смещение вторичного тактового сигнала относительно основного тактового сигнала, введите 1.0 и выберите **ns** (наносекунды) в диалоговом окне **Offset from base absolute clock** f_{max} (смещение относительно основного абсолютного тактового сигнала f_{max}) (рис. 3.56).

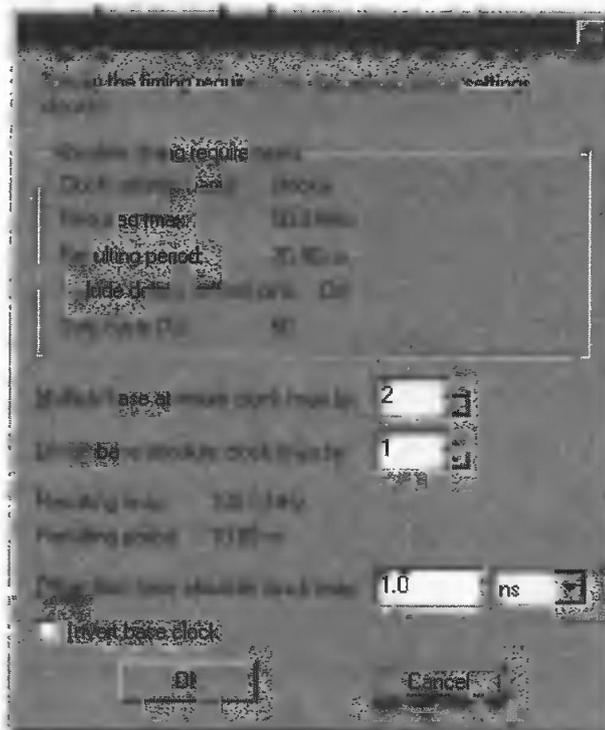


Рис. 3.56

7. Нажмите кнопку **OK**.
8. Чтобы добавить параметры тактового сигнала `clockb` к существующему списку параметров тактовых сигналов в окне **New Clock Settings** нажмите кнопку **OK**.
9. В диалоговом окне **Timing Settings** (временные характеристики) нажмите кнопку **OK**.

2. Привязка параметров тактового сигнала к выводам ПЛИС

После выполнения установок первичного и вторичного тактовых сигналов, необходимо применить эти установки к соответствующему тактовому сигналу(ам) проекта. Для создания, удаления, редактирования и просмотра назначений параметров проекта вы можете использовать команду **Assignment Organizer** (организатор назначений) меню **Tools** (инструментальные средства).

Для привязки параметров сlocks к выводу `clk` выполните следующие действия:

1. В меню **Tools** (инструментальные средства) выберите **Assignment Organizer** (организатор назначений). Автоматически появится вкладка **By Node** (по узлам).

2. Под надписью **Mode** (режим) выберите **Edit specific entity and node settings** (редактирование заданного объекта и параметров узла) (рис. 3.57).

Кнопка открытия окна
поиска узлов (**Node Finder**)

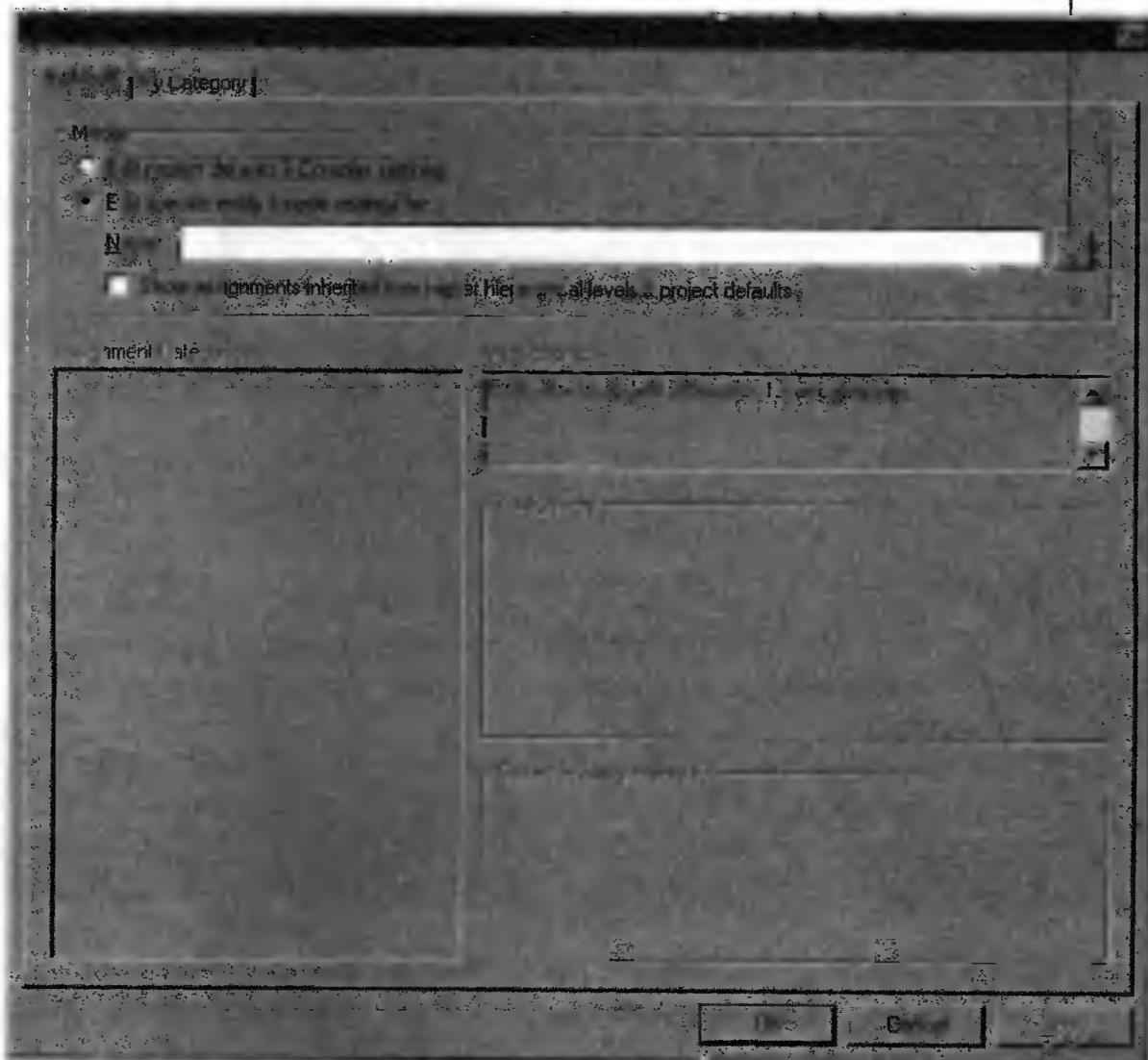


Рис. 3.57

3. Нажмите кнопку **Browse (...)** (обзор) рядом с полем имени — откроется диалоговое окно **Node Finder** (окно поиска узлов).
4. Чтобы найти нужный узел, в окне **Node Finder** (окно поиска узлов) из списка **Filter** выберите **Pins: all** (все выводы) и нажмите кнопку **Start** (рис. 3.58).
5. В списке **Nodes Found** (найденные узлы) дважды щелкните по имени вывода `clk`.
6. Чтобы скопировать имя вывода `clk` из диалогового окна **Node Finder** в диалоговое окно **Assignment Organizer**, нажмите кнопку **OK** — вы возвратитесь к диалоговому окну **Assignment Organizer**.

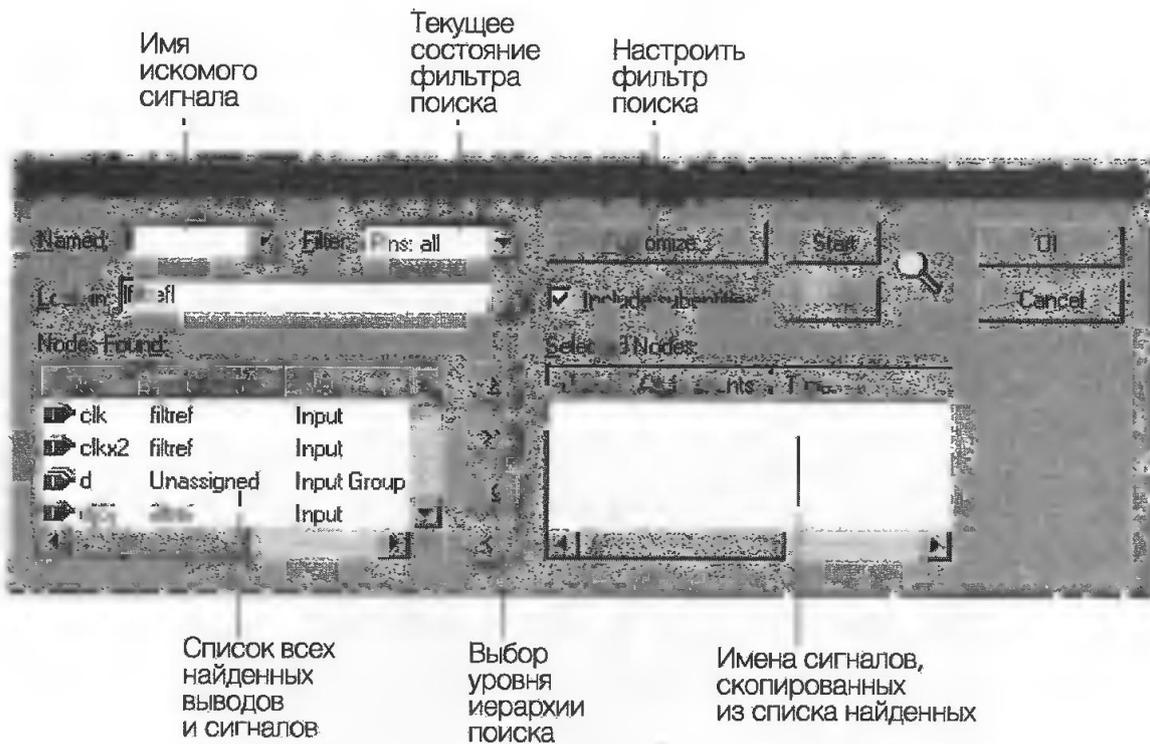


Рис. 3.58

7. Чтобы раскрыть раздел **Timing** (временные характеристики), в списке **Assignment categories** (категории назначений) щелкните по символу "+".
8. Чтобы добавить новый текст, щелкните на надписи **click here** (щелкнуть здесь).
9. Убедитесь, что в списке **Name** (имя) под надписью **Assignment** (назначения) выделено **Clock Settings** (параметры тактового сигнала).
10. Чтобы убедиться, что назначаемые параметры тактового сигнала соответствуют нужному выводу, проверьте, что в списке **Settings** выделен **clock**.
11. Нажмите **Add:** (добавить) — новое назначение появится в списке **Assignment Categories** (рис. 3.59).
12. Нажмите кнопку **OK**.
13. Чтобы назначить параметры **clock** выводу **clkx2**, повторите действия 1–12.

 Вы можете также определить много других типов индивидуальных временных параметров. Дополнительную информацию о назначениях временных параметров см. в справочной системе (**Help**) пакета Quartus в разделе "Making Individual Timing Assignments" (выполнение индивидуальных назначений временных параметров).

Кнопка открытия окна поиска сигналов (Node Finder)

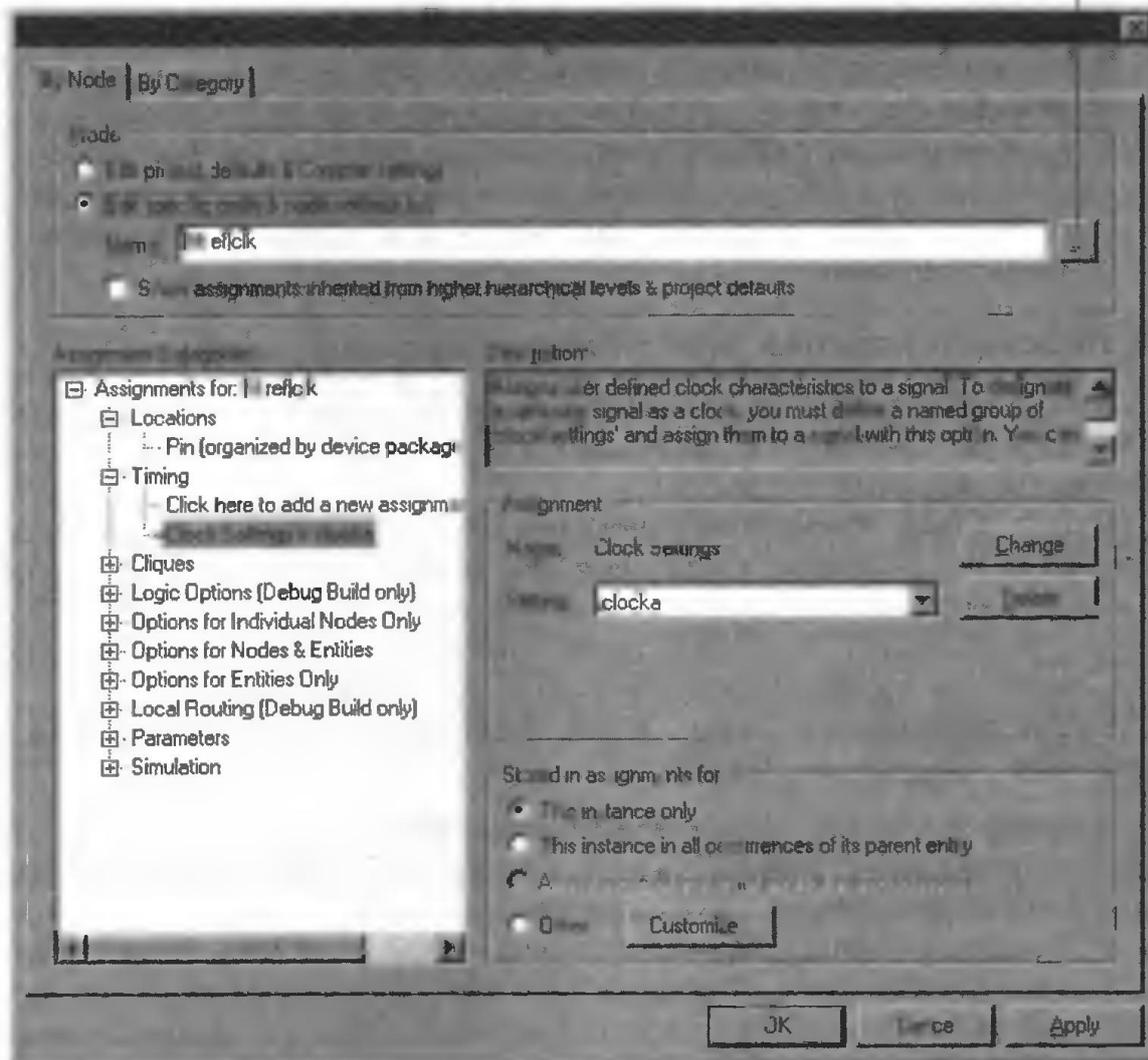


Рис. 3.59

3. Повторный запуск анализа временных характеристик

Для повторного запуска анализа временных характеристик выполните следующие действия:

1. В меню **Processing** выберите **Start Timing Analysis** (запуск анализа временных характеристик).
2. В диалоговом окне появится вопрос **"Want to recompile before starting timing analyses"** (хотите перекомпилировать проект перед запуском анализа временных характеристик), щелкните **No** (нет). Анализ временных характеристик будет выполнен без перекомпиляции проекта.



Вопрос о перекомпиляции проекта задается потому, что перепреопределение требований к быстройдействию затрагивает размещение и разводку проекта в ПЛИС, если используется компиляция

с оптимизацией по быстродействию. Однако для этого учебного проекта перекомпиляция не требуется.

- По окончании анализа в диалоговом окне будет выведено сообщение о том, что анализ временных характеристик был неудачен. Нажмите кнопку **ОК**. Выполненный анализ показал, что заданные требования к быстродействию не были достигнуты. В следующих разделах будет показано, как обеспечить требуемые временные характеристики.

4. Просмотр раздела требований к тактовом сигналам

При анализе проектов со сложной синхронизацией для каждого тактового сигнала автоматически создается раздел **Clock Requirement** (требования к тактовым сигналам).

Раздел **Clock Requirement** показывает критические пути проекта, для которых временные характеристики имеют пограничные значения (т.е. могут удовлетворять заданным требованиям, а могут и не удовлетворять). Черным цветом отображаются критические пути, удовлетворяющие заданным требованиям, красным — не удовлетворяющие.

Чтобы просмотреть раздел **Clock Requirement** для сигнала `clkx2`, выполните следующие действия:

- В левой области окна **Compilation Report** (отчет о компиляции) щелкните по символу "+", чтобы раскрыть раздел **Timing Analyses** (анализ временных характеристик).
- В разделе **Timing Analyses** выберите подраздел **Clock Requirement** для сигнала `clkx2`. Информация о результатах анализа будет показана в таблице. При необходимости размеры столбцов таблицы могут быть изменены путем перетаскивания их границ (рис. 3.60)

Signal Name	Requirement Name	Input Clock Name	Destination Clock Name	Signal Setup (ns)	Signal Hold (ns)	Actual Margin (ns)	Req. Margin (ns)
acc:inst3 accum:inst_1	inst5[0]	clk	clkx2	1.000 ns	0.220 ns	1.553 ns	-1.333 ns
acc:inst3 accum:inst_1	inst5[2]	clk	clkx2	1.000 ns	0.220 ns	1.505 ns	-1.285 ns
acc:inst3 accum:inst_1	inst5[3]	clk	clkx2	1.000 ns	0.220 ns	1.505 ns	-1.285 ns
acc:inst3 ccum:inst_1	inst5[5]	clk	clkx2	1.000 ns	0.220 ns	1.482 ns	-1.262 ns
acc:inst3 ccum:inst_1	inst5[1]	clk	clkx2	1.000 ns	0.220 ns	1.466 ns	-1.246 ns
acc:inst3 ccum:inst_1	inst5[4]	clk	clkx2	1.000 ns	0.220 ns	1.448 ns	-1.226 ns
acc:inst3 accum:inst_1	inst5[6]	clk	clkx2	1.000 ns	0.220 ns	1.445 ns	-1.225 ns
acc:inst3 accum:inst_1	inst5[7]	clk	clkx2	1.000 ns	0.220 ns	1.257 ns	-1.037 ns
inst_4	inst5[0]	clk	clkx2	1.000 ns	0.220 ns	1.102 ns	-0.882 ns
inst_4	inst5[2]	clk	clkx2	1.000 ns	0.220 ns	1.102 ns	-0.882 ns
Timing analysis restricted	To change the limit u						

Требование к установке сигнала — минимальный сдвиг между сигналами в 1.0 нс, определяемый настройками тактового сигнала `clockb`

Красный цвет обозначает, что значения параметров не удовлетворяют заданным требованиям

Рис. 3.60

 Фактические значения параметров могут отличаться от приведенных на рисунке из-за усовершенствований, сделанных в пакете Quartus после выпуска настоящего учебного пособия.

Раздел **Clock Requirement** для сигнала `clkx2` отображает критические времена красным цветом, поскольку требование к f_{max} не было достигнуто. В уроке 13 и последующих описывается решение этой проблемы введением путей **multicycle path** многотактовой синхронизации.

Урок 13. Определение путей Multicycle многотактовой синхронизации

В проекте `fir_filter` сдвиг на 1.0 нс, заданный для тактового сигнала `clockb`, заставляет систему реализовывать невыполнимые требования по быстродействию. Это обусловлено тем, что при анализе временных характеристик учитывается требование надежного защелкивания данных в тактируемых регистрах. При многотактовой синхронизации вводятся самые жесткие условия анализа, обусловленные устанавливаемыми по умолчанию соотношениями фронтов установки и записи сигналов в триггеры.

В проектах с многотактовой синхронизацией задержка на пути сигнала определяется для каждого тактового сигнала, синхронизирующего регистры-источники. При анализе проекта `fir_filter` по умолчанию считается, что запись сигнала в регистры осуществляется первым после момента установки сигнала фронтом тактовой последовательности `clkx2`. В этом случае минимальные отличия временного положения фронтов тактовых последовательностей определяют максимально жесткие требования к времени задержки.

Соотношения фронтов тактовых последовательностей в проекте `fir_filter` показаны на рис. 3.61.

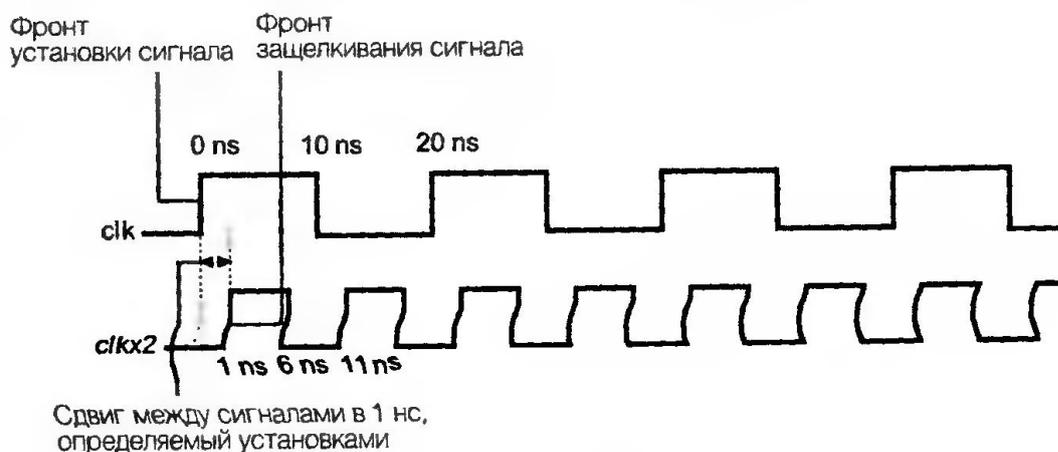


Рис. 3.61

Вы можете задать многотактовую синхронизацию при назначении временных характеристик конкретного пути.

Двум или более регистрам, тактируемым сигналом `clkx2`, можно назначить многотактовую синхронизацию, изменив установки по умолчанию и назначив запись сигнала вторым после момента его установки фронтом тактовой последовательности `clkx2`. Соотношения сигналов после изменения установок показаны на рис. 3.62.

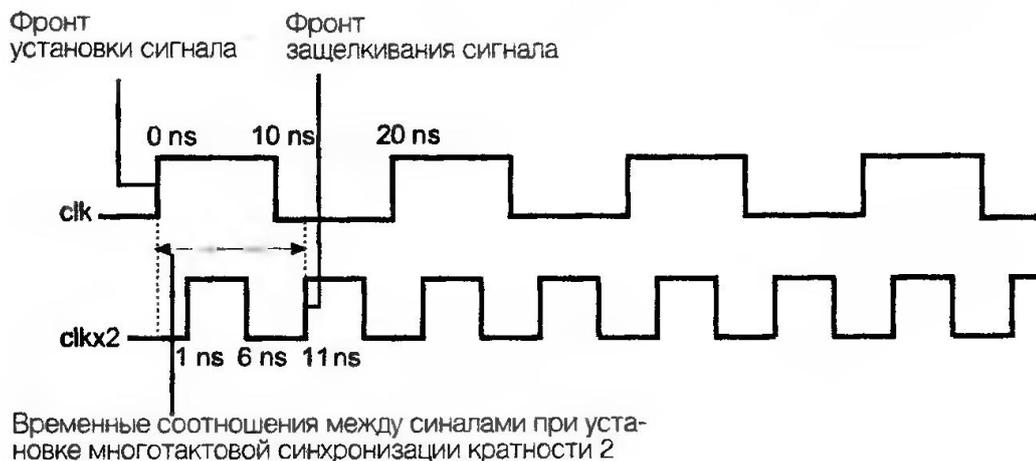


Рис. 3.62

Ниже описываются следующие действия:

1. Задание назначений многотактовой синхронизации.
2. Повторный запуск анализатора временных характеристик.
3. Просмотр раздела требований к тактовому сигналу.

1. Задание назначений многотактовой синхронизации

При создании назначения "точка-точка" (**point-to-point**) соотношений двух тактовых сигналов, заданное назначение автоматически применяется ко всем путям "регистр-регистр".

Чтобы назначить многотактовую синхронизацию всем путям регистр-регистр для сигналов `clk` и `clkx2`, выполните следующие действия:

1. Если необходимо открыть блок-схему **filtref.bdf** — в меню **File** выберите команду **Open** (открыть) и затем выберите файл **filtref.bdf**.
2. В блок-схеме **filtref.bdf** выберите входной вывод `clkx2`.
3. Во всплывающем по правой кнопке мыши меню выберите **Assignment Organizer** (организатор назначений) — откроется диалоговое окно **Assignment Organizer** с выделенной опцией **Edit specific entity & node settings for** (изменить параметры заданного объекта или узла) и с иерархическим путем и именем вывода `clkx2`, показанным в поле **Name**.
4. В списке **Assignment Categories** (категории назначений) щелкните по символу "+" чтобы раскрыть раздел **Timing** (синхронизация).

5. Чтобы добавить новое назначение, щелкните надпись **Click here** (щелкнуть здесь).
6. Под строкой **Assignment** (назначение) из списка **Name** выберите **Multicycle** (многотактная).
7. Чтобы определить путь с многотактной синхронизацией (по второму периоду), в поле **Setting** (установки) введите значение 2.
8. Определите назначение **Point-to-point** (точка-точка). В блоке **Fed by** (вести от) введите имя сигнала `clk` или нажмите **Browse (...)**, чтобы выбрать его с помощью **Node Finder** (окна поиска узла).
9. Нажмите кнопку **Add** (добавить) – теперь назначение появится в списке **Assignment Categories**.
10. Нажмите кнопку **OK**.

2. Повторный запуск анализа временных характеристик

Для повторного запуска анализа выполните следующие действия:

1. В меню **Processing** выберите **Start Timing Analysis** (запуск анализа временных характеристик).
2. Будет выведен вопрос: **"Want to recompile before starting timing analysis"** (хотите перекомпилировать проект перед запуском анализа временных характеристик?). Нажмите кнопку **No** (нет), анализ запустится без перекомпиляции проекта.
3. Будет выведено сообщение: **"timing analysis was successful"** (анализ временных характеристик прошел успешно), нажмите кнопку **OK**. Временной анализ был завершен успешно потому, что временные характеристики проекта удовлетворяют заданным требованиям.

3. Просмотр результатов анализа временных характеристик: требования к тактовому сигналу

Чтобы просмотреть результаты анализа временных характеристик и увидеть, как назначение многотактной синхронизации повлияло на выполнение заданных требований к временным характеристикам, выполните следующие действия:

1. Чтобы раскрыть раздел **Timing Analysis** (анализ временных характеристик), щелкните по символу "+" в левой области окна **Compilation Report** (отчет о компиляции).
2. В разделе **Timing Analysis** выберите подраздел **Clock Requirement** (требования к тактовому сигналу) для сигнала `clkx2`. В разделе **Clock Requirement** критические временные характеристики теперь отображаются черным цветом, это означает, что после отмены заданных по умолчанию параметров и назначения многотактной синхронизации временные характеристики удовлетворяют заданным требованиям (рис. 3.63).

Source Name	Destination Name	Source Clock Name	Destination Clock Name	Required Setup Relationship	Required Maximum P2P Time	Actual Maximum P2P Time	Violation
acc:inst3 accum:inst_1 p	inst5[0]	clk	clkx2	11.000 ns	10.220 ns	1.553 ns	8.667 ns
acc:inst3 accum:inst_1 p	inst5[2]	clk	clkx2	11.000 ns	10.220 ns	1.505 ns	8.715 ns
acc:inst3 accum:inst_1 p	inst5[3]	clk	clkx2	11.000 ns	10.220 ns	1.505 ns	8.715 ns
acc:inst3 accum:inst_1 p	inst5[5]	clk	clkx2	11.000 ns	10.220 ns	1.482 ns	8.738 ns
acc:inst3 accum:inst_1 p	inst5[1]	clk	clkx2	11.000 ns	10.220 ns	1.466 ns	8.754 ns
acc:inst3 accum:inst_1 p	inst5[4]	clk	clkx2	11.000 ns	10.220 ns	1.446 ns	8.774 ns
acc:inst3 accum:inst_1 p	inst5[6]	clk	clkx2	11.000 ns	10.220 ns	1.445 ns	8.775 ns
acc:inst3 accum:inst_1 p	inst5[7]	clk	clkx2	11.000 ns	10.220 ns	1.257 ns	8.963 ns
ir	inst5[0]	clk	clkx2	11.000 ns	10.220 ns	1.102 ns	9.118 ns
ir	inst5[2]	clk	clkx2	11.000 ns	10.220 ns	1.102 ns	9.118 ns
Timing analysis restricted	To change the limit						

Значения минимальных времен установки соответствуют заданному режиму 2-тактовой синхронизации (Multicycle 2)

Выделение черным цветом, означает, что значения параметров удовлетворяют заданным требованиям

Рис. 3.63



Фактические значения синхронизации могут отличаться от приведенных на рисунке из-за усовершенствований, сделанных в пакете Quartus после выхода этого учебного пособия.

3.5. Моделирование проекта

Моделирование (**Simulation**) позволяет полностью проверить проект и убедиться в правильности его функционирования перед программированием или конфигурированием ПЛИС.

Исходными данными для моделирования являются воздействия в виде входных векторов. Подсистема моделирования (**Simulator**) пакета Quartus моделирует выходные сигналы, соответствующие реакции проекта на заданные входные воздействия, которая очень близка к реакции запрограммированной ПЛИС. В типовых задачах разработчик задает наборы входных векторов и анализирует полученные в результате моделирования выходные сигналы.

В зависимости от решаемых задач, подсистема моделирования (**Simulator**) позволяет выполнять функциональное моделирование (**Functional Simulation**) либо моделирование с учетом временных параметров ПЛИС (**Timing Simulation**). При функциональном моделировании проверяется правильность описания и логического функционирования проекта. При моделировании с учетом временных параметров проверяется не только логическое функционирование проекта, но и его функционирование с учетом параметров выбранной ПЛИС в самых жестких временных условиях.

В следующих уроках рассматривается создание векторного файла *.vwf для анализа временных характеристик, определение параметров моделирования, запуск моделирования с учетом временных параметров и анализ результатов моделирования.

Урок 14. Создание временных диаграмм сигналов для моделирования

Векторные файлы ***.vwf (Vector Waveform Files)** можно создавать с помощью редактора временных диаграмм (**Waveform Editor**). Файлы ***.vwf** описывают входные и выходные векторы сигналов в графическом виде с помощью временных диаграмм (**Waveforms**).

Входные воздействия можно задать и в текстовом виде в векторных файлах (**Vector File**) ***.vec**. Система Quartus позволяет работать с файлами обоих указанных типов.

 Если вы уже знакомы с созданием *vwf*-файлов с помощью редактора временных диаграмм (**Waveform Editor**), вы можете сэкономить время и скопировать готовый файл **fir.vwf** из подкаталога **\qdesigns\tutorial** в подкаталог **\qdesigns\fir_filter**. Рекомендуется копировать файлы учебного пособия следующим образом: открыть файл в пакете Quartus с помощью команды **Open** (открыть) меню **File**, затем выбрать команду **Save As** (сохранить как...) меню **File**, в появившемся окне установить ключ **Add file to current project** (добавить файл к текущему проекту), а затем сохранить файл в подкаталоге **\qdesigns\fir_filter**. Если вы скопировали предоставленный файл, можете сразу переходить к разделу: "Определение параметров моделирования".

В текущем уроке рассматриваются следующие действия:

1. Создание файла временных диаграмм.
2. Добавление к временным диаграммам входных и выходных узлов.
3. Редактирование временных диаграмм входных сигналов.

1. Создание временных диаграмм

Для создания файла ***.vwf** временных диаграмм выполните следующие действия:

1. В меню **File** (файл) выберите команду **New** (новый).
2. На вкладке **Other Files** (другие файлы) выберите тип создаваемого файла — **Vector Waveform File** (векторный файл временных диаграмм).
3. Нажмите кнопку **ОК**: откроется редактор временных диаграмм (**Waveform Editor**) с пустым окном временных диаграмм (рис. 3.64).
4. В меню **Time** (время) командой **End Time** (конечное время) задайте конечное время интервала рисования временных диаграмм
5. В появившемся окне в поле **Time** (время) введите значение 700 и выберите в списке единиц измерения **ns** (наносекунды).
6. Нажмите кнопку **ОК**.

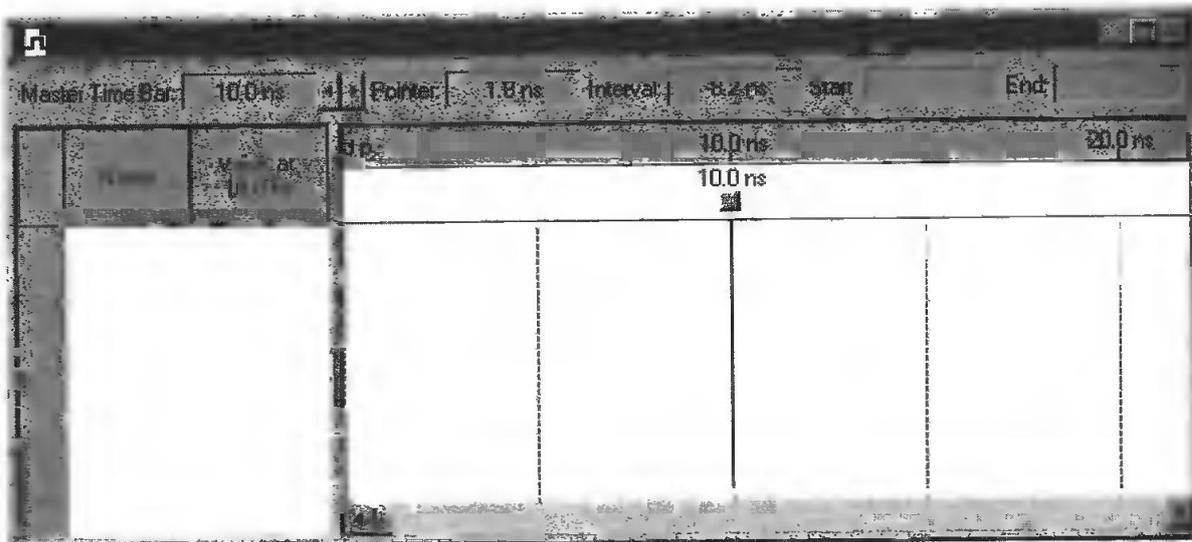


Рис. 3.64

7. Сохраните файл под именем **fir.vwf**, для чего в меню **File** выберите команду **Save As**, в поле **Name**, открывшегося окна введите **fir** и нажмите кнопку **Save**.

2. Добавление входов и выходов к временным диаграммам

После создания файла *.vwf, необходимо ввести в него входы и выходы, а также другие узлы для которых будут создаваться временные диаграммы.

Для добавления узлов к временным диаграммам выполните следующие действия:

1. Для нахождения вводимых узлов удобно пользоваться окном поиска узлов (**Node Finder**). Вызов окна осуществляется в меню **View > Auxiliary Windows > Node Finder** (вид > вспомогательные окна > окно поиска узлов).
2. В окне **Node Finder** (окно поиска узлов), в списке **Filter** (фильтр типов узлов), выберите **Pins all:** (все типы выводов).
3. Для поиска узлов нажмите кнопку **Start** (старт) (рис. 3.65).
4. В списке **Nodes Found** (найденные узлы) выберите выходы **clk**, **clkx2**, **d**, **newt**, **reset**, **yvalid**, **next**, **yn_out** и перетащите их в колонку **Name** (имя) временных диаграмм *.vwf. Большие непрерывные группы имен из нескольких частей можно выбрать левой кнопкой мыши с нажатой клавишей **Shift**, а большие группы имен, состоящие из нескольких отдельных частей, можно выбрать левой кнопкой мыши с нажатой клавишей **Ctrl**.
5. Закройте окно поиска узлов через меню **View > Auxiliary Windows > Node Finder**.

 С помощью окна поиска узлов вы можете также задавать собственные фильтры поиска узлов в проекте. Подробные инст-

рукции по созданию таких фильтров можно получить в справочной системе пакета Quartus в разделе "Creating a Custom Filter" (создание заданного фильтра).

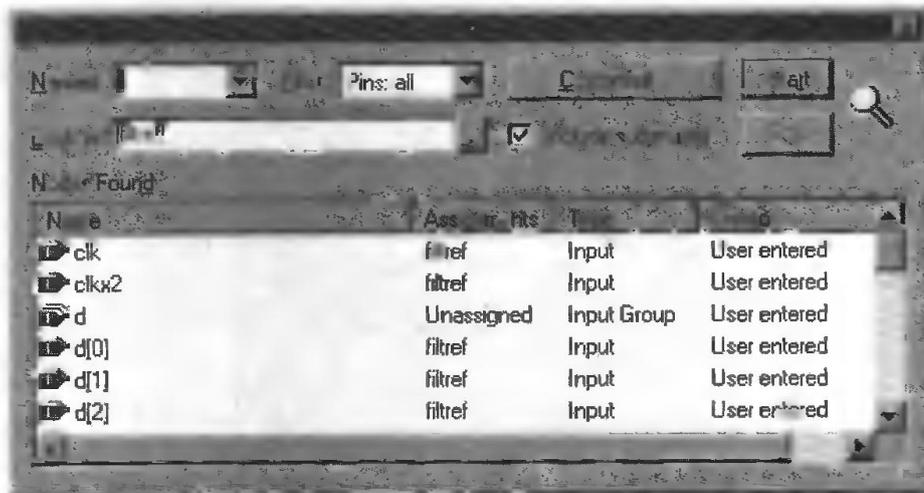


Рис. 3.65

3. Редактирование временных диаграмм входных сигналов

Входные векторы для моделирования создаются путем задания логических уровней на временных диаграммах соответствующих узлов.

Для того, чтобы отредактировать временную диаграмму входа clk, выполните следующие действия:

1. На панели инструментов редактора временных диаграмм (**Waveform Editor**) щелкните по кнопке **Selection Tool** (указатель выбора).
2. Чтобы выбрать и выделить полный временной интервал диаграммы входа clk, щелкните левой кнопкой мыши на маркере clk, расположенном на левом краю окна (рис. 3.66).

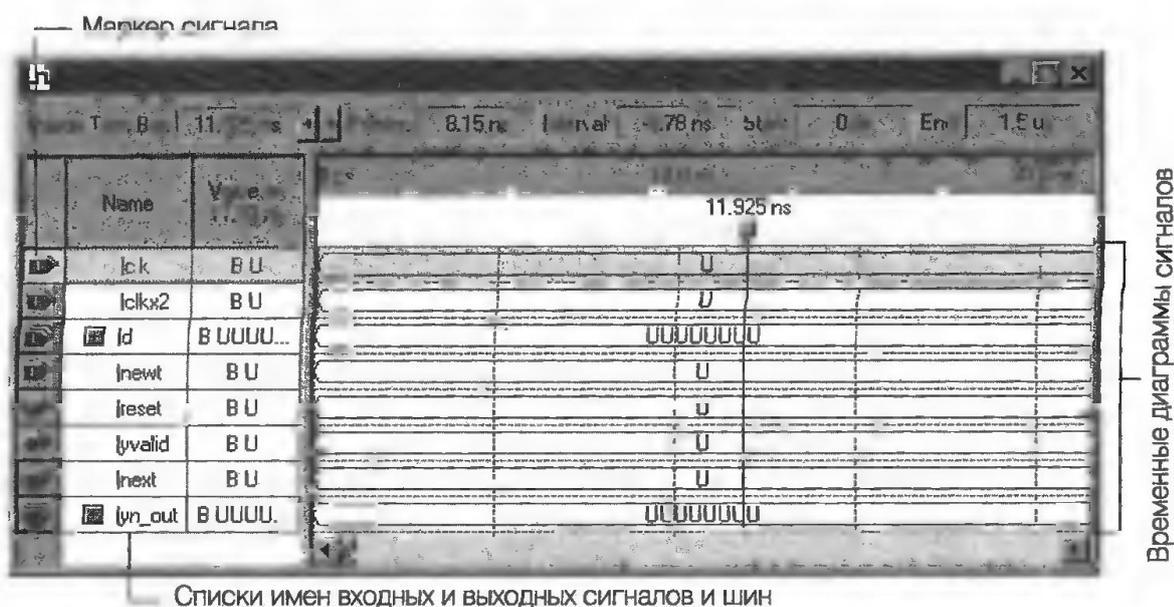


Рис. 3.66

3. В меню **Value** (величина) выберите **Clock** (тактовый сигнал).
4. Под надписью **Base waveform on** (основывать временную диаграмму на...) выберите **Clock Settings** (установки тактового сигнала) и выберите из списка clocka (рис. 3.67).

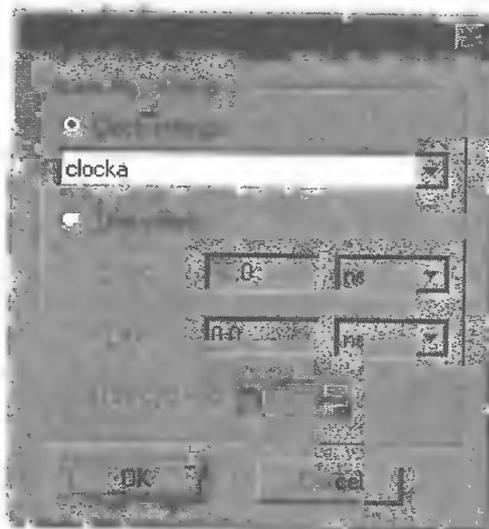


Рис. 3.67

5. Нажмите кнопку **OK** — будет создана временная диаграмма clk с использованием параметров тактового сигнала clocka.

Для того, чтобы отредактировать временную диаграмму входа clkx2, выполните следующие действия:

1. На панели инструментов редактора временных диаграмм (**Waveform Editor**) щелкните по кнопке **Selection Tool** (указатель выбора).
2. Чтобы выбрать и выделить полный временной интервал диаграммы входа clkx2, щелкните левой кнопкой мыши на маркере узла clk, расположенном на левом краю окна (рис. 3.66).
3. В меню **Value** (значение) выберите **Clock** (тактовый сигнал).
4. Под надписью **Base waveform on** (основывать временную диаграмму на...) выберите **Clock Settings** (установки тактового сигнала) и выберите из списка clockb (рис. 3.67).
5. Нажмите кнопку **OK** — теперь временная диаграмма clk будет создана с использованием параметров тактового сигнала clockb.

Чтобы отредактировать временную диаграмму входной шины d, выполните следующие действия:

1. Чтобы выбрать и выделить полный временной интервал диаграммы входной шины d, щелкните левой кнопкой мыши на маркере шины, расположенном на левом краю окна (рис. 3.66).
2. В меню **Value** (величина) выберите **Arbitrary Value** (произвольная величина).

3. В списке **Radix** (тип системы счисления) выберите **Unsigned Decimal** (десятичное положительное).
4. В списке **Numeric or Named value** (числовая или именованная величина) введите 16 (рис. 3.68).

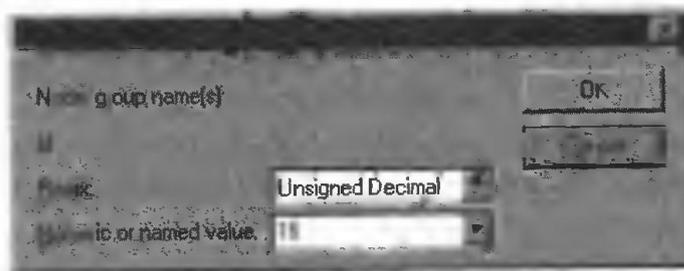


Рис. 3.68

5. Нажмите **ОК** — на временной диаграмме входной шины *d* будет отображено значение **16** в двоичном коде.

Чтобы отредактировать временную диаграмму входа *newt*, выполните следующие действия:

1. Чтобы выбрать и выделить полный временной интервал диаграммы входной шины *newt*, щелкните левой кнопкой мыши на маркере шины, расположенном на левом краю окна (рис. 3.66).
2. В меню **Value** (величина) выберите **Clock** (тактовый сигнал).
3. Под надписью **Base waveform on** (основывать временную диаграмму на...) выберите **Time period** (период времени) и выберите из списка.
4. В поле **Period** (период) введите 80, а из списка единиц измерения выберите **ns** (наносекунды).
5. Из списка **Duty Cycle** (коэффициент заполнения) выберите **25**.
6. Нажмите кнопку **ОК** — на временной диаграмме ввода *newt* будет показана импульсная последовательность с периодом 80 нс и коэффициент заполнения 25%.

Чтобы отредактировать временную диаграмму входа *reset* выполните следующие действия:

1. С помощью указателя выбора (**Selection Tool**) выделите на временной диаграмме входа *reset* интервал 0–20 нс, для чего поместите указатель в позицию **0 ns**, нажмите левую кнопку мыши и, удерживая ее, переместите указатель в позицию **20 ns**.
2. На выделенном интервале нажмите правую кнопку мыши и во всплывающем меню выберите **Value > Forcing Low** (низкий логический уровень).
3. С помощью указателя выбора (**Selection Tool**) выделите на временной диаграмме входа *reset* интервал 20–40 нс, для чего поместите указатель в позицию **20 ns**, нажмите левую кнопку мыши и, удерживая ее, переместите указатель в позицию **40 ns**.

4. На выделенном интервале нажмите правую кнопку мыши и во всплывающем меню выберите **Value > Forcing High** (высокий логический уровень).
5. Чтобы увидеть временную диаграмму на всем временном интервале, в меню **View** выберите **Fit in Window** (вписать в окно).
6. С помощью указателя выбора (**Selection Tool**) на временной диаграмме входа `reset` выделите интервал от 40 нс до конца временной диаграммы, для чего поместите указатель в позицию **40 ns**, нажмите левую кнопку мыши и, удерживая ее, переместите указатель к концу временной диаграммы.
7. Нажмите на выбранном интервале правую кнопку мыши, во всплывающем меню выберите **Value > Forcing Low** (низкий логический уровень).
8. Сохраните файл, выбрав в меню **File** (операции с файлами) команду **Save** (сохранить).

Созданные временные диаграммы показаны на рис. 3.69.

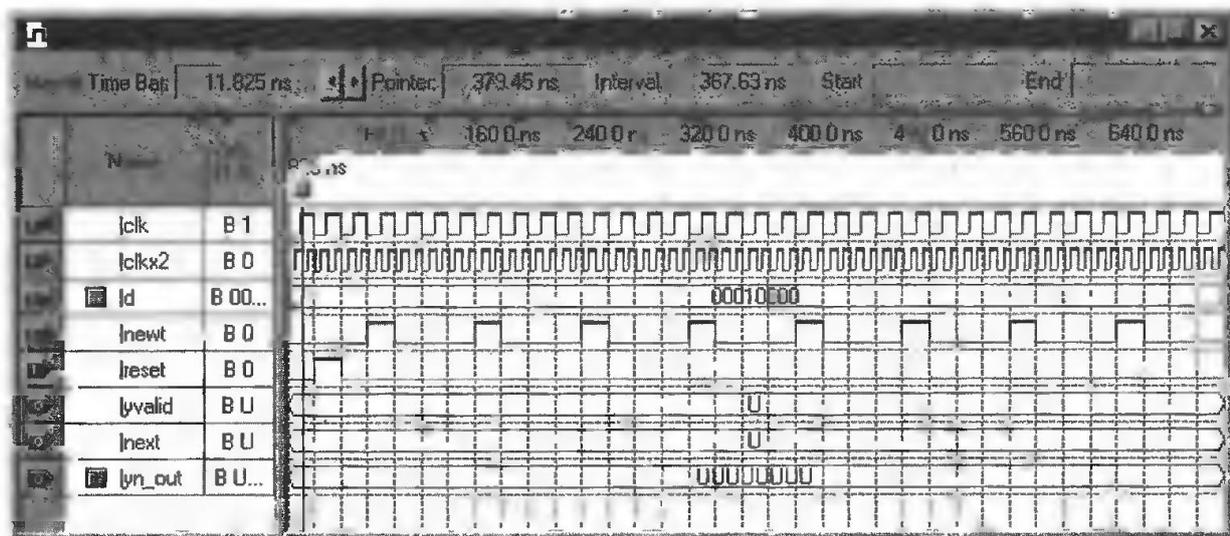


Рис. 3.69

Урок 15. Определение параметров моделирования

Система Quartus позволяет осуществлять моделирование всего проекта или любой его составляющей. Вы можете назначить любой объект верхнего уровня иерархии проекта в качестве моделируемой части проекта (**Simulation Focus**).

Настройки моделирования определяют моделируемую часть проекта (**Simulation Focus**), тип моделирования, интервал времени моделирования, векторы входных воздействий и другие параметры.

Параметры моделирования автоматически назначаются системой Quartus при создании нового проекта, кроме того, вы можете самостоятельно определять группы параметров моделирования.

В этом уроке рассматриваются следующие действия:

1. Просмотр основных параметров подсистемы моделирования (**Simulator**).
2. Задание временных параметров и параметров векторов.
3. Задание режима моделирования.
4. Установка дополнительных параметров моделирования.

 *Ниже поясняется выполнение просмотра и редактирования параметров моделирования с использованием меню и диалоговых окон. Для облегчения процесса настройки моделирования удобно использовать встроенный мастер настройки моделирования (**Simulator Settings Wizard**). Данный мастер вызывается соответствующей командой меню **Processing** (обработка).*

1. Просмотр основных параметров моделирования

На вкладке **General** (основные) диалогового окна **Simulation Settings** (рис. 3.70), вызываемого из меню **Processing** (обработка), осуществляется выбор групп параметров моделирования, определение и сохранение новой группы параметров, определение моделируемой части проекта (**Simulation Focus**), а также удаление существующих параметров настройки.

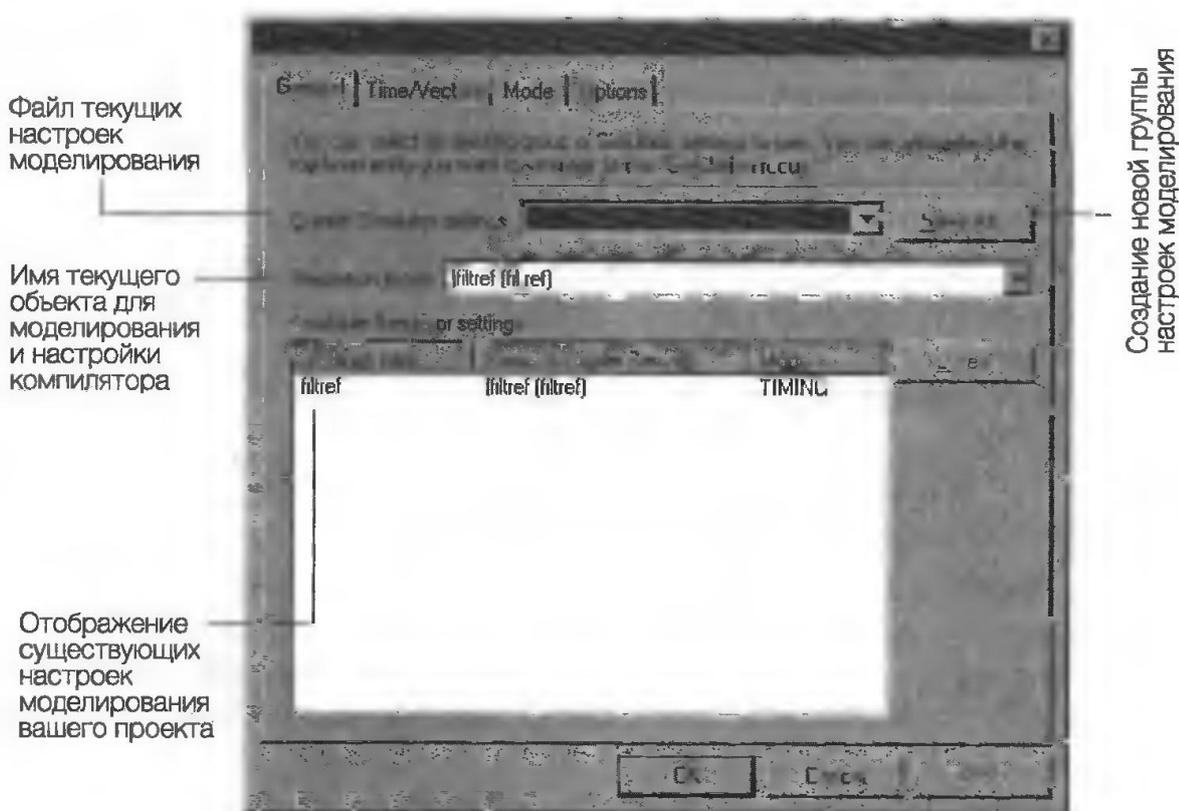


Рис. 3.70

Для просмотра заданных по умолчанию основных параметров моделирования текущего проекта выполните следующие действия:

1. Сначала необходимо перейти в режим моделирования (**Simulate Mode**), выбрав соответствующую команду в меню **Processing** (обработка).
2. В меню **Processing** выберите **Simulator Settings** (установки подсистемы моделирования) — автоматически откроется вкладка **General** (основные, общие).
На вкладке **General** отображаются основные параметры моделирования, установленные пакетом Quartus при создании вашего проекта (рис. 3.70).

2. Задание интервала моделирования и параметров входных векторов

На вкладке **Time/Vectors** диалогового окна **Simulator Settings** подсистемы моделирования (**Simulator**) задается интервал времени моделирования и генерации входных векторов. Входные воздействия для моделирования задаются векторными файлами ***.vwf** временных диаграмм или векторных файлов ***.vec** с текстовым описанием входных векторов, входные воздействия можно также задавать в окне **Tcl Console**.

Чтобы определить интервал времени моделирования и источник входных сигналов, выполните следующие действия:

1. В диалоговом окне **Simulator Settings** подсистемы моделирования (**Simulator**) перейдите на вкладку **Time/Vectors** (время/вектора).
2. Под надписью **Simulation period** (интервал моделирования) в поле **Start time** (начальное время) введите 0 и из списка единиц измерения выберите **ns** (наносекунды).
3. Убедитесь, что под надписью **End time** (конечное время) выбран пункт **Run simulation until all vector stimuli are used** (выполнять моделирование до тех пор, пока не будут использованы все входные вектора).
4. Под надписью **Vectors** (вектора) в поле **Source of Vector Stimuli** (источник векторов входных воздействий) введите **D:\qdesigns\fir_filter\fir.vwf** или выберите файл, нажав кнопку **Browse (...)** (обзор).

 При запуске моделирования система автоматически ищет векторные файлы (***.vwf**, ***.vec**, или ***.tbl**) с именем, указанным в параметрах настройки. Поэтому не надо специально указывать файл, если его имя совпадает с именем, указанным в текущих параметрах настройки,

5. Убедитесь, что под надписью **Vectors** (векторы) включены опции **Automatically add pins to simulation output waveforms** (автоматическое добавление к временным диаграммам выходных выводов) (рис. 3.71).

появится описание моделирования с учетом временных параметров (рис. 3.72).

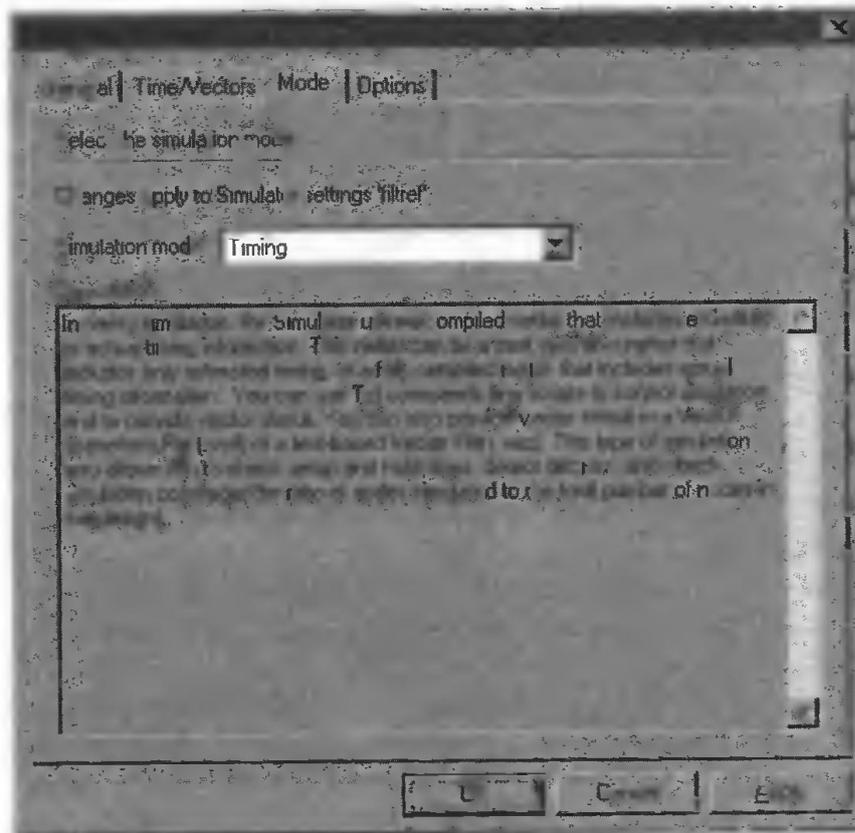


Рис. 3.72

Комментарий:

Текст в окне Description Во время моделирования с учетом временных параметров используется список соединений, включающий информацию о предположительных или реальных задержках внутри ПЛИС. Этот список может быть списком только после синтеза и содержать предполагаемые задержки либо списком соединений после полной компиляции и содержать точно рассчитанные временные характеристики. Вы можете использовать команды **Tcl** и описания на макроязыке для контроля моделирования и обозначения входных векторов. Также вы можете задавать входные вектора из файла векторов временных диаграмм (**.vwf**) или текстовых векторных файлов (**.vec**). Этот тип моделирования также пригоден для расчета времени установки и удержания, выявления неполадок и проверки границ моделирования (отношение моделируемых точек к общему числу точек в проекте).

4. Установка дополнительных параметров моделирования

На вкладке **Options** (дополнительные параметры) диалогового окна **Simulator Settings** устанавливаются дополнительные параметры для отслеживания обнаруженных при моделировании рисков (критических состояний сигналов).

Для установки параметров отслеживания рисков (критических соединений) выполните следующие действия:

1. В диалоговом окне **Simulator Settings** перейдите на вкладку **Options** (параметры).
2. Включите установку **Simulation coverage report** (сообщать об обнаруженных при моделировании рисках) (рис. 3.73).

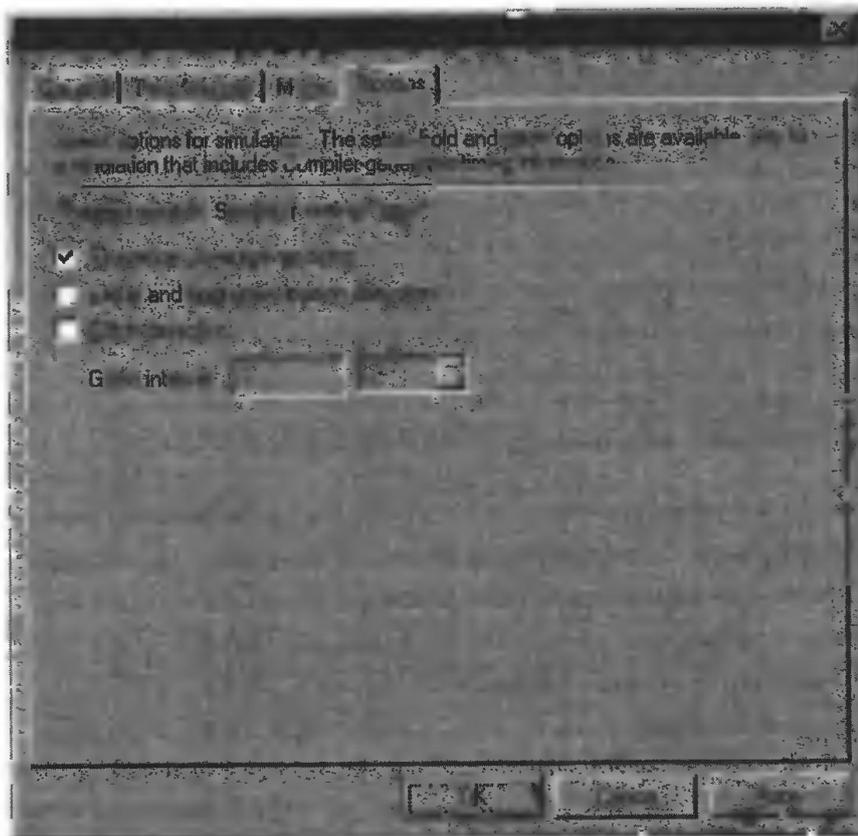


Рис. 3.73

3. При нажатии кнопки ОК все определенные вами для проекта **filtref** параметры и опции моделирования будут сохранены.

Урок 16. Выполнение моделирования

Чтобы запустить моделирование выполните следующие действия:

1. В меню **Processing** (обработка) выберите команду **Run Simulation** (запуск моделирования).

После этого начинается моделирование объекта **filtref** верхнего уровня иерархии проекта, а также всех вложенных объектов проекта, с использованием заданных для **filtref** параметров и векторного файла входных воздействий **fir.vwf**. При моделировании проекта автоматически открывается окно **Status** (состояние), в котором показывается ход моделирования. Кроме того, результаты моделирования показываются в окне **Simulation Report** (отчет о ком-

пиляции), содержимое которого изменяется в соответствии с текущим состоянием моделирования.

Моделирование выполняется в фоновом режиме. Во время моделирования компьютер может быть использован для других задач. Моделирование проекта **filtref** не занимает много времени, поэтому вам не придется долго ждать. При работе со сложными проектами вы можете запустить моделирование, а затем переключиться на работу с другими окнами и программами.

2. После успешного завершения моделирования на экран будет выведено соответствующее сообщение. После этого нажмите кнопку **ОК**.

••• *Дополнительную информацию можно получить в справочной системе (**Help**) пакета **Quartus** в разделах "Creating a Breakpoint" (создание контрольных точек) и "Updating Embedded Memory" (обновление содержимого встроенной памяти).*

Урок 17. Анализ результатов моделирования

В процессе моделирования автоматически появляется окно **Simulation Report**. Сообщения о моделировании отображают необходимую информацию о текущем моделировании. По умолчанию область временного анализа моделирования отображается в окне **Simulation Report**. При выборе раздела **Simulation Waveform** окна **Simulator Report** показываются полученные при моделировании временные диаграммы заданных выходов.

Окно **Simulation Report** содержит также другие необходимые разделы, предоставляющие различную информацию, включая параметры моделирования, генерируемые при моделировании сообщения, а также сообщение о затраченном на моделирование времени.

В этом уроке описываются следующие действия:

1. Просмотр полученных при моделировании временных диаграмм.
2. Создание временных маркеров (**Time Bar**).

1. Просмотр временных диаграмм моделирования

После выполнения моделирования объектов выбранной части проекта вы можете просмотреть и проанализировать временные диаграммы заданных выходных сигналов.

Для просмотра полученных при моделировании временных диаграмм выполните следующие действия:

1. В левой области окна **Simulation Report** выберите раздел **Simulation Waveform** (временные диаграммы моделирования).
2. Чтобы увидеть временную диаграмму на всем временном интервале, в меню **View** выберите **Fit in Window** (вписать в окно).

Временная диаграмма появится в правой области окна **Simulation Report** (рис. 3.74).

Щелчок раскрывает группу (шину) и показывает отдельные битовые сигналы шины

Кнопки перемещения опорного временного маркера

Кнопка перетаскивания опорного временного маркера

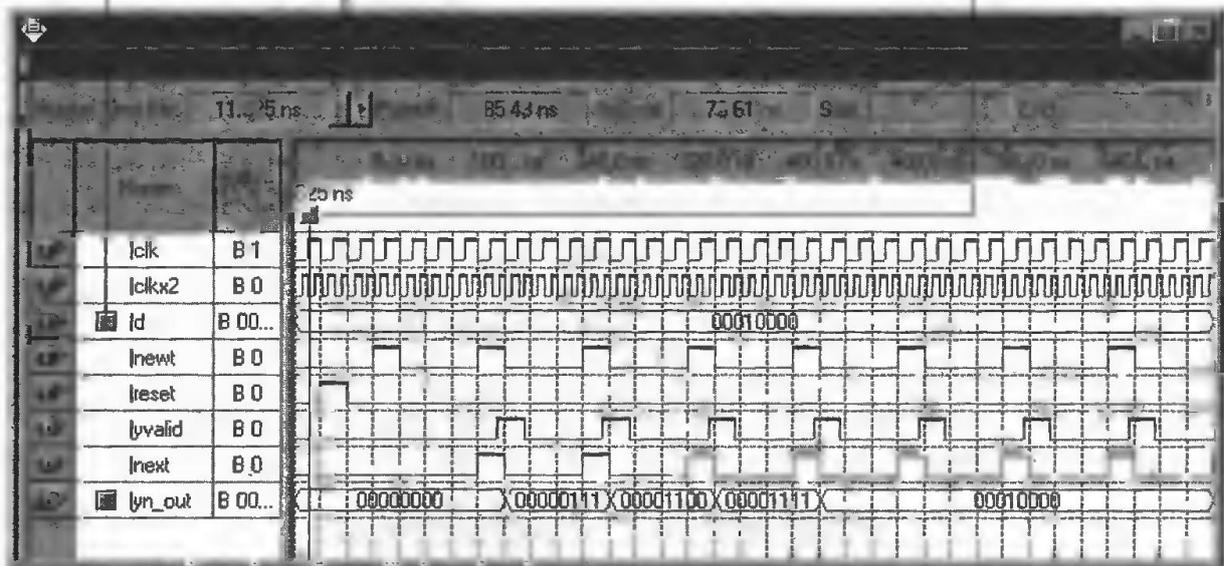


Рис. 3.74

В редакторе **Waveform Editor** (редактор временных диаграмм) можно выполнять следующие действия с отображаемыми данными:

- можно выбирать систему счисления, используемую для отображения значений групп (шин). Для этого выделите нужную группу, нажмите правую кнопку мыши и во всплывающем меню выберите **Properties** (свойства), а затем в списке **Radix** (основание системы счисления) выберите нужную систему счисления;
- вы можете перемещать по событиям на временной оси опорный временной маркер, нажимая на стрелки перемещения, как показано на рисунке. Кроме того, вы можете также перетаскивать временной маркер на нужное место непосредственно мышью за специальную кнопку на самом маркере;
- вы можете изменять масштаб отображения с помощью кнопки **Zoom Mode Tool** панели инструментов редактора **Waveform Editor** (редактор временных диаграмм), а также с помощью команд **Zoom In** (увеличить), **Zoom Out** (уменьшить), **Fit In Window** (вписать в окно) и команды **Zoom** (увеличение) в меню **View**;
- вы можете создавать дополнительные временные маркеры и использовать их для измерения временных интервалов (см. следующий раздел).

2. Создание временных маркеров и измерение интервалов

Для создания временного маркера выполните следующие действия:

1. Наведите указатель на кнопку перемещения опорного временного маркера (**Master Time Bar handle**), нажмите левую кнопку мыши и перетащите маркер до совмещения со вторым фронтом сигнала `clk`. Временное положение маркера отображается вверху окна и составляет 30.0 нс.
2. Переместите указатель на время 160 нс. Соответствующее положению указателя мыши время **160.0 ns** отображается вверху окна в поле **Pointer** (указатель). Интервал между опорным временным маркером и текущим положением указателя мыши **130.0 ns** отображается в поле **Interval** (интервал).
3. Нажмите правую кнопку мыши в области временных диаграмм и во всплывающем меню выберите **Insert Time Bar** (вставить временной маркер).
4. В поле **Time** (время) введите 160 и выберите из списка единиц измерения **ns** (наносекунды) (рис. 3.75).



Рис. 3.75

5. Нажмите кнопку **ОК**. Созданный временной маркер появится на временной диаграмме в точке **160.0 ns**. Над маркером отображается интервал **+130.0 ns** между первым и вторым временными маркерами (рис. 3.76).

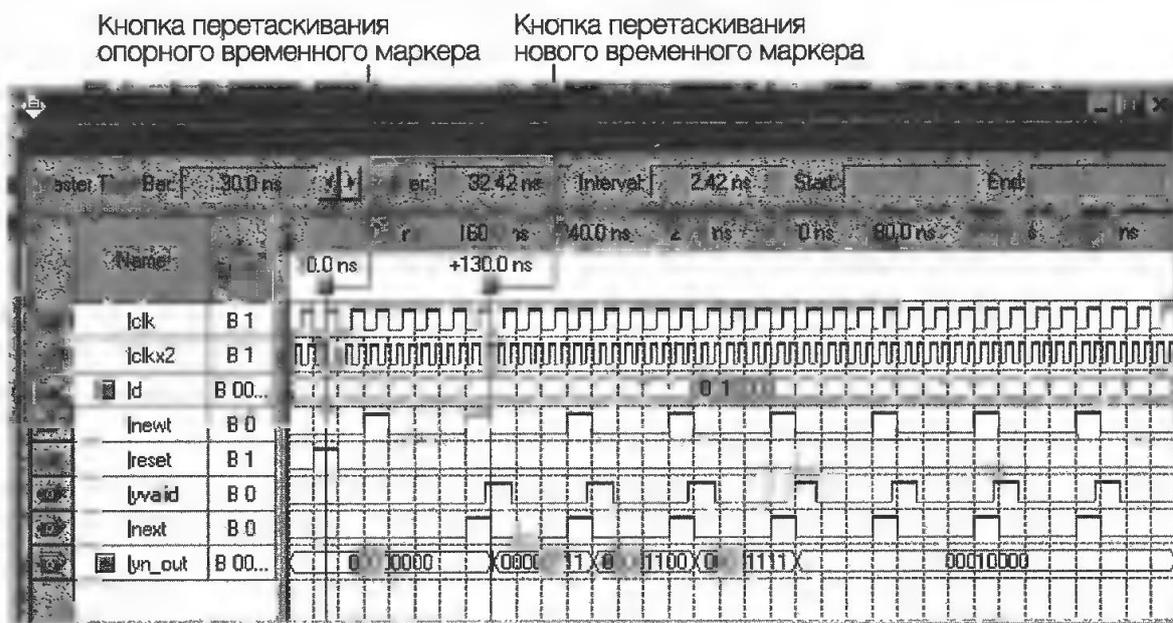


Рис. 3.76

-  При перемещении созданных временных маркеров по временной оси, соответственно будут изменяться отображаемые значения временных интервалов.

3.6. Программирование и конфигурирование ПЛИС

Конфигурирование ПЛИС (например, АРЕХ20К), а также программирование конфигурационных ПЗУ типа ЕРС2 выполняются в пакете Quartus с помощью подсистемы программирования **Programmer** (программатор).

Физически загрузка конфигурационной информации в микросхемы выполняется с помощью аппаратных устройств MasterBlaster или ByteBlasterMV, подключаемых к соответствующим к последовательному или параллельному порту персонального компьютера.

Программатор (**Programmer**) пакета Quartus позволяет также выполнить следующие операции с конфигурационными ПЗУ ЕРС2 типа EEPROM:

- **IBlank-check**: проверить отсутствие в микросхеме записанной конфигурационной информации;
- **IExamine**: считать конфигурационную информацию из микросхемы;
- **IVerify**: сравнить конфигурационную информацию, считанную из запрограммированной микросхемы, и информацию из файла программирования.

Конфигурационная информация генерируется компилятором (**Compiler**) пакета Quartus в виде одного или нескольких конфигурационных файлов. Последние затем используются программатором (**Programmer**) пакета для конфигурирования ПЛИС (например, АРЕХ20К) или программирования конфигурационных ПЗУ ЕРС2.

Программатор может использоваться для конфигурирования или программирования как одной, так и нескольких микросхем.

Конфигурирование микросхем семейства АРЕХ 20К может быть выполнено в следующих двух режимах:

1. Последовательный пассивный режим (**Passive Serial configuration mode**).
2. Режим JTAG (**JTAG mode**).

В режиме JTAG может быть выполнена конфигурация одной либо нескольких микросхем. При этом конфигурация микросхем выполняется в порядке, указанном в окне подсистемы программирования (**Programmer**).

В этом учебном пособии рассматривается программирование одной микросхемы.

Урок 18. Программирование микросхем фирмы Альтера

В этом уроке описывается выполнение следующих действий:

1. Открытие окна подсистемы программирования (**Programmer**).
2. Установка пассивной последовательной цепочки конфигурирования.
3. Конфигурирование микросхемы.
4. Изменение режимов программирования.
5. Добавление микросхемы в цепочку конфигурирования.

Перед выполнением описанных в данном разделе действий необходимо выполнить компиляцию проекта.

1. Открытие подсистемы программирования (**Programmer**)

При подготовке к программированию или конфигурированию микросхем необходимо открыть окно программатора (**Programmer**) и создать файл ***.cdf** (Chain Description File – файл описания схемы (цепочки) конфигурирования), в котором содержится информация о семействе и типе микросхем, а также параметры настройки аппаратных устройств конфигурирования.

Файлы ***.cdf** используются для программирования или конфигурирования одной или нескольких микросхем в цепочке JTAG (**JTAG chain**) или в последовательной пассивной цепочке (**Passive Serial chain**).

Для открытия окна **Programmer** и создания CDF-файла выполните следующие действия:

1. В меню **File** (операции с файлами) выберите команду **New** (создать новый файл).
2. В диалоговом окне **New** (создание нового файла) перейдите на вкладку **Other Files** (другие файлы).
3. Из списка **Other Files** (другие файлы) выберите **Chain Description File** (файл описания схемы конфигурирования) (рис. 3.77).
4. Для создания CDF-файла нажмите кнопку **OK**. В окне **Programmer** откроется новый чистый CDF-файл (рис. 3.78).
5. Сохраните файл, выбрав в меню **File** (операции с файлами) команду **Save As** (сохранить файл как...).
6. В диалоговом окне **Save As** (сохранить файл как...) введите в поле задания имени файла имя **fir_filter.cdf**.
7. Убедитесь в том, что в списке типов сохраняемых файлов выбран файл типа **Chain Description File** (файл описания схемы конфигурирования).
8. Нажмите кнопку **Save** (сохранить).

 Окно **Programmer** можно также открыть через меню **Processing** (обработка) командой **Open Programmer** (открыть подсистему программирования).

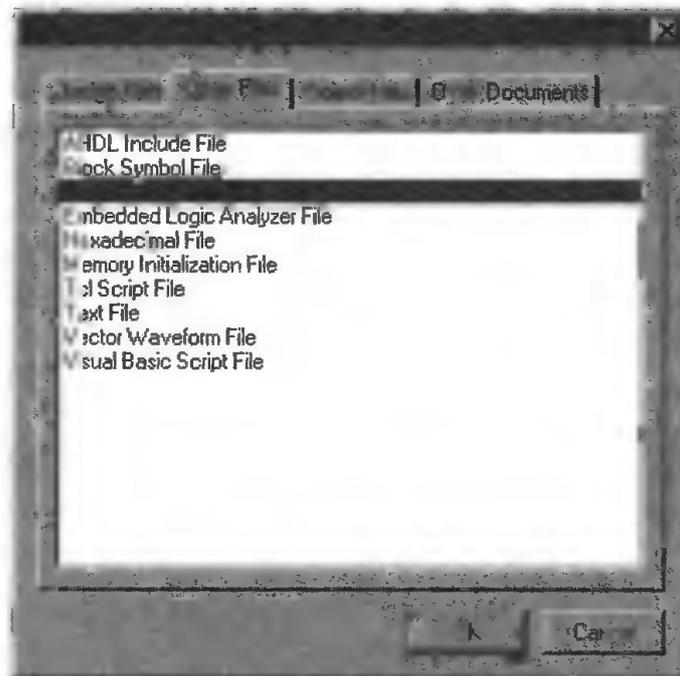


Рис. 3.77



Рис. 3.78

2. Установка последовательной пассивной схемы конфигурирования

Для выполнения конфигурации одной микросхемы в последовательной пассивной цепочке выполните следующие действия:

1. В списке **Mode** (режим (схема) конфигурирования) окна **Programmer** выберите **Passive Serial** (последовательная пассивная схема).

2. Под надписью **Programming Hardware** (аппаратные средства программирования) щелкните **Setup** (установить). Откроется диалоговое окно **Hardware Setup** (установка аппаратных средств).
3. В списке **Hardware Type** (тип аппаратного конфигурационного устройства) выберите **ByteBlasterMV** или **MasterBlaster** и, при необходимости, определите используемый порт компьютера и скорость обмена информацией в бодах (**Bod**). Нажмите кнопку **OK**.
4. Нажмите кнопку **Add File** (добавить файл программирования).
5. В диалоговом окне **Select File** (выбор файла программирования), в поле **File Name** (имя файла) введите `filtref.sof`.
6. Нажмите кнопку **Open** (открыть файл) (рис. 3.79).

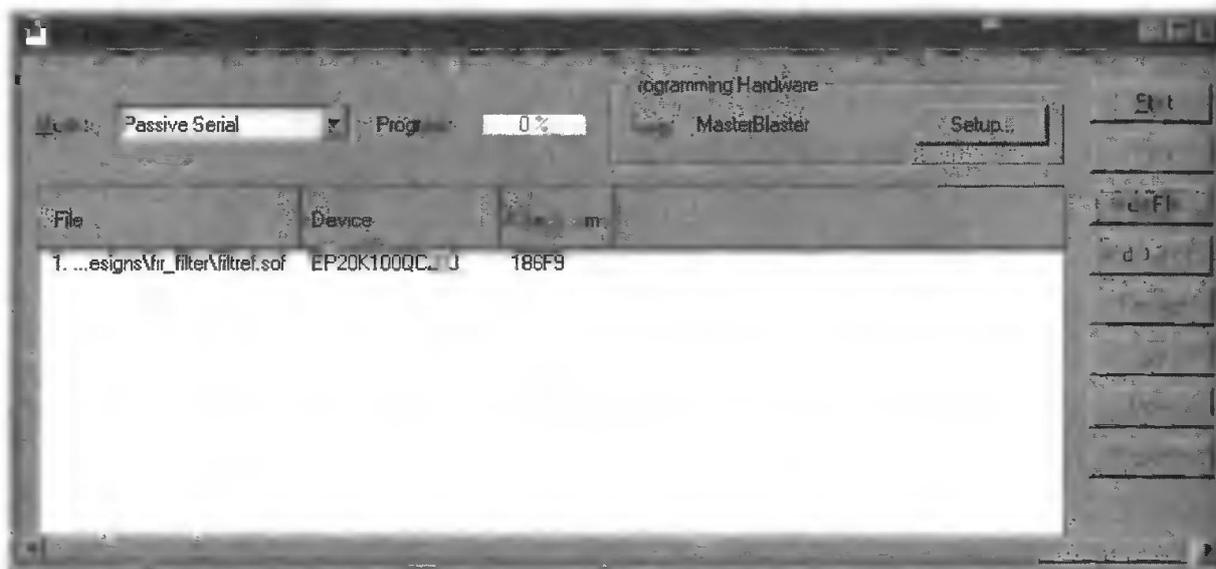


Рис. 3.79

7. В меню **File** (операции с файлами) выберите команду **Save** (сохранить файл).



Для программирования нескольких микросхем необходимо повторить действия 1–5, чтобы добавить конфигурационные файлы для других конфигурируемых микросхем. Однако в нашем случае требуется всего один конфигурационный файл, поскольку в этом учебном пособии описывается программирование одной микросхемы.

3. Выполнение конфигурирования микросхем

Для конфигурирования микросхемы или нескольких микросхем выполните следующие действия:

1. Сначала необходимо подключить соответствующее конфигурационное устройство к вашей рабочей станции платформ PC или UNIX. Для этого выполните одно из следующих действий:

- ✓ Подключите устройство загрузки конфигурации ByteBlasterMV или MasterBlaster к вашей рабочей станции платформы PC. Дополнительную информацию вы можете найти в руководстве "Quartus Installation & Licensing for PCs manual".

или

- ✓ Подключите устройство загрузки конфигурации ByteBlasterMV или MasterBlaster к вашей рабочей станции платформы UNIX. Дополнительную информацию вы можете найти в руководстве "Quartus Installation & Licensing for PCs manual".

2. Запустите конфигурирование, нажав в окне **Programmer** кнопку **Start**. После успешного выполнения конфигурирования появится соответствующее сообщение.

4. Изменение режимов программирования

Микросхемы семейства APEX20K можно конфигурировать в JTAG-режиме или в последовательном пассивном режиме (**Passive Serial**).

При выборе JTAG-режима в окне отображаются функции подсистемы программирования: **Program/Configure**, **Verify**, **Blank-Check**, **Examine**, **Security Bit**. При работе с микросхемами ПЛИС (например, APEX20K) используется только функция **Program/Configure** (программирование/конфигурирование). Другие функции, показанные в окне для JTAG-режима, используются при работе с конфигурационными ПЗУ (например, типа EPС2) (рис. 3.80).

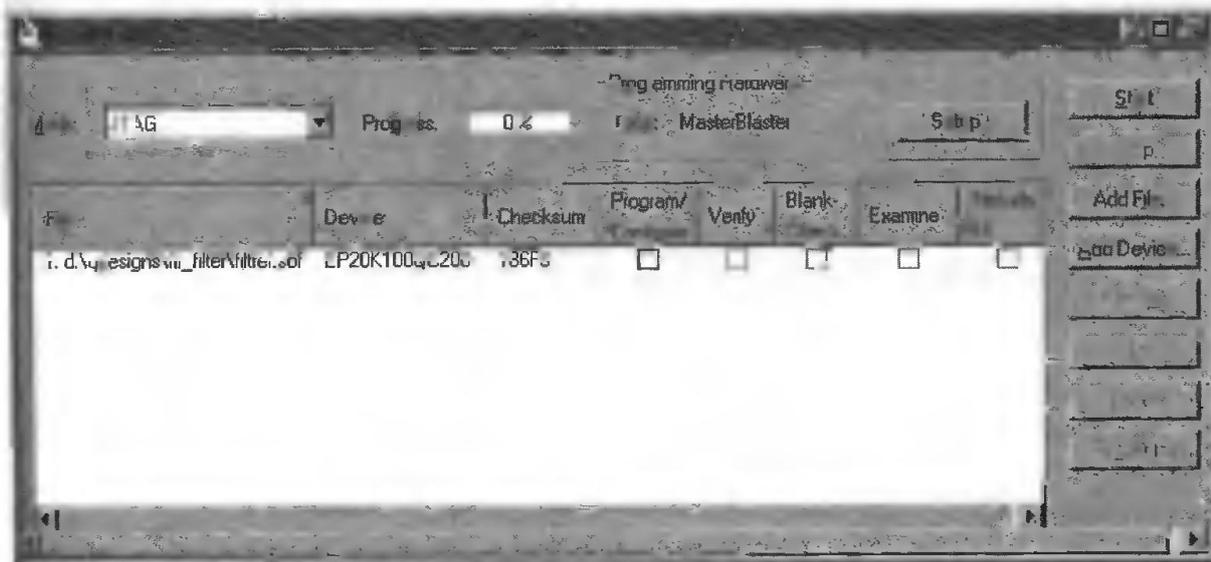


Рис. 3.80

5. Включение микросхемы в цепочку конфигурирования

Включение микросхемы в цепь конфигурирования выполняется в окне **Programmer**. При включении микросхемы в схему JTAG можно

включить функцию **Examine**, которая позволяет считать конфигурационную информацию из микросхемы, поместить ее во временный буфер, а затем сохранить в файле ***.pof** – Programmer Object File (объектный файл программатора).

Если вы не хотите запрограммировать какую-либо микросхему в конфигурационной цепочке – выключите все функции программирования: **Program**, **Verify**, **Blank-Check** и **Examine**. При этом конфигурационные данные будут проходить через такую микросхему, но запрограммирована она не будет.

Чтобы включить микросхему в JTAG-цепочку, выполните следующие действия:

1. В окне **Programmer** выберите **Add Device** (добавить микросхему в JTAG-цепочку).
2. В диалоговом окне **Select Device** (выбор микросхемы) выберите микросхемы, включающиеся в список **Devices** (микросхемы).
3. Нажмите кнопку **OK**.
4. Сохраните созданную цепочку, выбрав в меню **File** (операции с файлами) команду **Save** (сохранить файл).

 При необходимости включения в цепочку дополнительных микросхем необходимо повторить действия 1–3. Однако в нашем случае этого не требуется, поскольку программируется одна микросхема.

 Дополнительную информацию по программированию микросхем можно получить в справочной системе пакета *Quartus*.